



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ ΔΙΑΧΕΙΡΙΣΗ
ΔΕΔΟΜΕΝΩΝ, ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΓΝΩΣΗΣ**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Διαχείριση και Οπτικοποίηση Μεγάλων Δεδομένων για τα
Κρούσματα COVID-19 στην Ελλάδα**

Στυλιανός Ε. Σέρβος

**Επιβλέπουσα: Μαρία Ρούσσου, Επίκουρη Καθηγήτρια
Μέλος Επιτροπής: Αλέξανδρος Ντούλας, Επίκουρος Καθηγητής
Μέλος Επιτροπής: Δημήτριος Γουνόπουλος, Καθηγητής**

ΑΘΗΝΑ

ΜΑΪΟΣ 2021

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Διαχείριση και Οπτικοποίηση Μεγάλων Δεδομένων για τα Κρούσματα COVID-19 στην Ελλάδα

Στυλιανός Ε. Σέρβος

A.M.: CS2.18.0013

ΕΠΙΒΛΕΠΟΥΣΑ: Μαρία Ρούσσου, Επίκουρη Καθηγήτρια
ΜΕΛΟΣ ΕΠΙΤΡΟΠΗΣ: Αλέξανδρος Ντούλας, Επίκουρος Καθηγητής
ΜΕΛΟΣ ΕΠΙΤΡΟΠΗΣ: Δημήτριος Γουνόπουλος, Καθηγητής

Αφιερωμένη στη Μονάδα Προσβασιμότητας για ΦμεΑ του ΕΚΠΑ

ΜΑΪΟΣ 2021

ΠΕΡΙΛΗΨΗ

Η λοίμωξη COVID-19, υπεύθυνος για την οποία είναι ο πρωτοεμφανιζόμενος κορονοϊός SARS-CoV2, εμφανίστηκε στην Κίνα κατά το τέλος του 2019 και έκτοτε έχει λάβει διαστάσεις πανδημίας, σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας (World Health Organization-W.H.O.). Οι σύγχρονες εφαρμογές της πληροφορικής και ειδικότερα ο κλάδος των μεγάλων δεδομένων συνεισφέρουν σημαντικά στη διαχείριση των πληροφοριών σχετικά με τα περιστατικά λοίμωξης από τη νόσο, όπως αυτά καταγράφονται από τις δομές δημόσιας υγείας αλλά και ιδιωτικές ειδησεογραφικές ιστοσελίδες.

Στο πλαίσιο της παρούσας διπλωματικής εργασίας έχει πραγματοποιηθεί η συλλογή μεγάλων δεδομένων για την COVID-19 σε περιοχές της Ελλάδας για το χρονικό διάστημα από 01/01/2020 έως και 27/05/2021, μέσα από ελληνικούς ειδησεογραφικούς ιστοχώρους, έτσι ώστε να είναι δυνατή η εξαγωγή πληροφοριών για τα κρούσματα COVID-19 στην Ελλάδα (με βάση την ημερομηνία, την τοποθεσία και τον αριθμό των κρουσμάτων). Στη συνέχεια δημιουργήθηκαν διαδραστικές οπτικοποιήσεις για να παρουσιαστούν τα παραπάνω δεδομένα με τρόπο όσο γίνεται πιο κατανοητό στο ευρύ κοινό.

Για την υλοποίηση της εργασίας χρησιμοποιήθηκαν η γλώσσα προγραμματισμού Python και το εργαλείο προγραμματισμού Jupyter Notebook. Συνολικά, συλλέχθηκαν 284.688 ιστοσελίδες από ιστοχώρους όπως οι in.gr, tovima.gr, tanea.gr και protothema.gr. Στη διαδραστική οπτικοποίηση των δεδομένων δίνεται η δυνατότητα στους χρήστες να φιλτράρουν τα δεδομένα σύμφωνα με διάφορα κριτήρια καθώς και να συγκρίνουν ακόμα και γεωγραφικά διαμερίσματα της Ελλάδας ως προς τον αριθμό κρουσμάτων από τη λοίμωξη COVID-19.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Διαχείριση Μεγάλων Δεδομένων, Οπτικοποίηση Δεδομένων

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: μεγάλα δεδομένα, οπτικοποίηση δεδομένων, web crawling,

αποθήκευση περιεχομένων σελίδων, εξαγωγή πληροφοριών, γλώσσα

Python, COVID-19

ABSTRACT

COVID-19, responsible for which is the debutant coronavirus SARS-Cov2, appeared in China at the end of 2019 and has since received dimensions of pandemic, according to the World Health Organization – W.H.O. Contemporary applications of informatics, especially in the field of big data, contribute significantly to information management about cases of infection by the disease, as these are recorded by structures of public health and private news webpages.

In the context of this thesis, data was collected from Greek news websites about COVID-19 infection cases in the various areas of Greece, for the period from 01/01/2020 to 27/05/2021. The goal has been to extract information on COVID-19 cases in Greece (based on the date, location and number of cases) and proceed to create interactive visualizations to present the above data in a manner that is as understandable as possible to the general public.

The thesis was implemented using the Python programming language and the Jupyter Notebook programming tool. In total, 284,688 webpages were collected from websites such as in.gr, tovima.gr, tanea.gr and protothema.gr. The interactive visualization of the data provides users with the ability to filter the data according to various criteria as well as to compare geographical regions of Greece in terms of the number of cases of COVID-19 infection.

SUBJECT AREA: Big Data Management, Data Visualization

KEY WORDS: big data, data visualization, web crawling, saving page contents,
information extraction, Python language, COVID-19

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη διεκπεραίωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω την επιβλέπουσα επίκουρη καθηγήτρια κ. Μαρία Ρούσσου για τη συνεργασία μας και την απλόχερη βοήθεια που μου προσέφερε και εξακολουθεί να μου προσφέρει. Πάντοτε συναρπαστική και καθοδηγητική στην περιπλάνησή μου σε δύσβατες περιοχές, όπως η «Επικοινωνία ανθρώπου μηχανής», που μου γνώρισε προπτυχιακά. Πάντοτε γενναιόδωρη στην προσφορά χρόνου, κόπου και γνώσεων, με αμείωτη φροντίδα και κέφι στο καθοριστικό για τις μεταπτυχιακές αναζητήσεις μου μεταπτυχιακό της μάθημα «Οπτικοποίηση Δεδομένων». Ένα μεγάλο ευχαριστώ. Κι ας μην είναι αρκετό για όλα όσα της οφείλω...

Η πρώτη μου επαφή με τη διαχείριση μεγάλων δεδομένων έγινε κατά το εαρινό εξάμηνο 2019-2020, όταν είχα την ευκαιρία να διδαχθώ το μάθημα του επίκουρου καθηγητή κ. Αλέξανδρου Ντούλα «Διαχείριση Μεγάλων Δεδομένων». Στην εξ αποστάσεως εκπαίδευση που έκτοτε μου παρείχε, όπως και στις γενναιόδωρες υποδείξεις του, στην άψογη και «ζεστή» του φροντίδα οφείλω πολλά.

Στον καθηγητή κ. Δημήτρη Γουνόπουλο οφείλω τη μύησή μου στις «Τεχνικές εξόρυξης δεδομένων», όπως και στις «Τεχνικές ανάλυσης δεδομένων υψηλής κλίμακας». Τον ευχαριστώ για όλα όσα μου δίδαξε. Για την προθυμία του συνεχώς να με συμβουλεύει, αλλά και για το ότι δέχθηκε τη συμμετοχή του στην επιτροπή επίβλεψης της διπλωματικής μου εργασίας.

Διπλωματικής εργασίας, που δεν θα μπορούσα να φανταστώ ότι θα ήμουν ικανός να εκπονήσω, όταν -το 2013- εισαγόμουν στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του ΕΚΠΑ. Δίχως τον καθηγητή κ. Γιώργο Κουρουπέτρογλου, τον Πρόεδρο τώρα του Τμήματός μας, δίχως τους ΑΝΘΡΩΠΟΥΣ του στη Μονάδα Προσβασιμότητας για ΦμεΑ, την κ. Αριάδνη Βελισσαροπούλου, την κ. Μαρία Εμμανουήλ, την κ. Βιβή Ρήγα, την κ. Ελένη Πιτσινέλη, τον κ. Γιάννη Πετρούλη, το πιθανότερο θα ήταν να έχω γίνει καταγέλαστος για την αναπηρία μου και να έχω φύγει από τη Σχολή. Δεν θα μπορούσα να αφιερώσω τη διπλωματική μου παρά μόνο στη Μονάδα προσβασιμότητας για ΦμεΑ του ΕΚΠΑ.

Θερμά ευχαριστώ τον καθηγητή κ. Πάνο Ροντογιάννη, που μου άνοιξε ορίζοντες, με καθοδήγησε, μου συμπαραστάθηκε στις προπτυχιακές σπουδές και στην πτυχιακή μου εργασία σαν να ήταν μέλος της οικογένειάς μου. Του οφείλω πολλά, όπως οφείλω και στην οικογένεια μου για τη συνεχή στήριξη και συμπαράστασή της σε όλα τα χρόνια των προπτυχιακών και μεταπτυχιακών σπουδών μου. Μα πιο πολύ ευχαριστώ τη μητέρα μου για το χιούμορ και την κατανόηση που πάντοτε δείχνει στα δύσκολα, για την εγκάρδια παρουσία της σε όλα αυτά τα χρόνια της μαθησιακής μου δοκιμασίας και ενηλικίωσης.

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΡΟΛΟΓΟΣ.....	11
1. ΕΙΣΑΓΩΓΗ	12
1.1 Λοίμωξη COVID-19: Ορισμός και παρούσα κατάσταση ανά τον πλανήτη	12
1.2 Ορισμός των Μεγάλων Δεδομένων	12
1.3 Τύποι των Μεγάλων Δεδομένων	12
1.4 Ορισμός της Οπτικοποίησης Δεδομένων	13
1.5 Πληροφορίες Οπτικοποιήσεων	13
1.6 Διάρθρωση εργασίας	17
2. ΣΤΑΔΙΑ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ	18
2.1 Συλλογή Ιστοσελίδων (Web Crawling)	18
2.2 Αποθήκευση περιεχομένων σελίδων	18
2.3 Εξαγωγή πληροφοριών	18
2.4 Οπτικοποίηση Δεδομένων	18
3. ΥΛΟΠΟΙΗΣΕΙΣ ΤΩΝ ΣΤΑΔΙΩΝ ΤΗΣ ΕΡΓΑΣΙΑΣ	19
3.1 Κώδικας που κάνει συλλογή (crawling) σελίδων	19
3.2 Κώδικας που αποθηκεύει περιεχόμενα σελίδων	20
3.3 Κώδικας που επεξεργάζεται ελληνικές τοποθεσίες	22
3.4 Κώδικας που εκτελεί εξαγωγή πληροφοριών	34
3.5 Κώδικας που επεξεργάζεται αρχεία	37
3.6 Κώδικας που εκτελεί οπτικοποίηση δεδομένων	54
4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΥΛΟΠΟΙΗΣΕΩΝ	57
4.1 Αποτελέσματα του σταδίου Web Crawling	57
4.2 Αποτελέσματα του σταδίου Αποθήκευση περιεχομένων σελίδων	60
4.3 Αποτελέσματα του σταδίου Εξαγωγή Πληροφοριών	63
4.4 Αποτελέσματα του σταδίου Οπτικοποίηση Δεδομένων	64
5. ΣΥΜΠΕΡΑΣΜΑΤΑ	68
ΠΑΡΑΡΤΗΜΑ.....	70
Προγράμματα Python	70
I.1 Προγράμματα με λίστες	70
I.1.1 Πρόγραμμα που βρίσκει τους άρτιους αριθμούς μιας λίστας	70
I.1.2 Πρόγραμμα που βρίσκει τα τετράγωνα των αριθμών μιας λίστας	70

I.2 Προγράμματα με συμβολοσειρές	70
I.2.1 Πρόγραμμα που τυπώνει Hello World	70
I.2.2 Πρόγραμμα που διαβάζει και τυπώνει συμβολοσειρά	70
 ΑΝΑΦΟΡΕΣ	 71

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

Εικόνα 1: Πληροφορίες Οπτικοποιήσεων	13
Εικόνα 2: Κώδικας που κάνει συλλογή (crawl) σελίδων	19
Εικόνα 3: Κώδικας που αποθηκεύει περιεχόμενα σελίδων(1-49)	21
Εικόνα 4: Κώδικας που αποθηκεύει περιεχόμενα σελίδων (50-59)	21
Εικόνα 5: Κώδικας που επεξεργάζεται νομούς (1-49).....	23
Εικόνα 6: Κώδικας που επεξεργάζεται νομούς (50-91).....	23
Εικόνα 7: Κώδικας που επεξεργάζεται νομούς (92-112).....	24
Εικόνα 8: Κώδικας που επεξεργάζεται δήμους (1-47)	24
Εικόνα 9: Κώδικας που επεξεργάζεται δήμους (48-87).....	25
Εικόνα 10: Κώδικας που επεξεργάζεται δήμους (88-129).....	25
Εικόνα 11: Κώδικας που επεξεργάζεται δήμους (130-171).....	26
Εικόνα 12: Κώδικας που επεξεργάζεται δήμους (172-213).....	26
Εικόνα 13: Κώδικας που επεξεργάζεται δήμους (214-251).....	27
Εικόνα 14: Κώδικας που επεξεργάζεται δήμους (252-287).....	27
Εικόνα 15: Κώδικας που επεξεργάζεται δήμους (288-327).....	28
Εικόνα 16: Κώδικας που επεξεργάζεται δήμους (328-365).....	28
Εικόνα 17: Κώδικας που επεξεργάζεται δήμους (366-403).....	29
Εικόνα 18: Κώδικας που επεξεργάζεται δήμους (404-437).....	29
Εικόνα 19: Κώδικας που επεξεργάζεται δήμους (438-473).....	30
Εικόνα 20: Κώδικας που επεξεργάζεται δήμους (474-513).....	30
Εικόνα 21: Κώδικας που επεξεργάζεται δήμους (514-555).....	31
Εικόνα 22: Κώδικας που επεξεργάζεται δήμους (556-595).....	31
Εικόνα 23: Κώδικας που επεξεργάζεται δήμους (596-637).....	32
Εικόνα 24: Κώδικας που επεξεργάζεται δήμους (638-656).....	32
Εικόνα 25: Κώδικας που βρίσκει ελληνικές τοποθεσίες	33
Εικόνα 26: Κώδικας που επεξεργάζεται ελληνικές τοποθεσίες	33
Εικόνα 27: Κώδικας που εκτελεί εξαγωγή πληροφοριών (1-49)	34
Εικόνα 28: Κώδικας που εκτελεί εξαγωγή πληροφοριών (50-91)	35
Εικόνα 29: Κώδικας που εκτελεί εξαγωγή πληροφοριών (92-136)	35
Εικόνα 30: Κώδικας που εκτελεί εξαγωγή πληροφοριών (137-164)	36
Εικόνα 31: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (1-49).....	38
Εικόνα 32: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (50-93).....	39
Εικόνα 33: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (94-139).....	40
Εικόνα 34: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (140-185).....	41
Εικόνα 35: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (186-229)	42
Εικόνα 36: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (230-274).....	43
Εικόνα 37: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (275-319)	44
Εικόνα 38: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (320-365).....	45
Εικόνα 39: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (366-411)	46
Εικόνα 40: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (412-455).....	47
Εικόνα 41: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (456-501)	48
Εικόνα 42: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (502-545).....	49
Εικόνα 43: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (546-591)	50
Εικόνα 44: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (592-637)	51
Εικόνα 45: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (638-681).....	52

Εικόνα 46: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (682-702)	53
Εικόνα 47: Κώδικας που επεξεργάζεται αρχεία	53
Εικόνα 48: Κώδικας που εκτελεί οπτικοποίηση δεδομένων (1-49)	54
Εικόνα 49: Κώδικας που εκτελεί οπτικοποίηση δεδομένων (50-93)	55
Εικόνα 50: Κώδικας που εκτελεί οπτικοποίηση δεδομένων (94-138)	56
Εικόνα 51: Σελίδες που συλλέχθηκαν από τον ιστοχώρο in.gr	57
Εικόνα 52: Σελίδες που συλλέχθηκαν από τον ιστοχώρο tovima.gr	58
Εικόνα 53: Σελίδες που συλλέχθηκαν από τον ιστοχώρο tanea.gr	59
Εικόνα 54: Σελίδες που συλλέχθηκαν από τον ιστοχώρο protothema.gr	60
Εικόνα 55: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο in.gr	61
Εικόνα 56: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο tovima.gr	61
Εικόνα 57: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο tanea.gr	62
Εικόνα 58: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο protothema.gr	63
Εικόνα 59: Αποτελέσματα του σταδίου Εξαγωγή Πληροφοριών	64
Εικόνα 60: Κατακόρυφο γράφημα ράβδων (bar chart) όπου οπτικοποιούνται τα κρούσματα ανά γεωγραφικό διαμέρισμα της Ελλάδας με φίλτρα το μήνα και το έτος	65
Εικόνα 61: Γράφημα γραμμής (Line chart) όπου οπτικοποιούνται τα κρούσματα ανά μήνα με φίλτρα το γεωγραφικό διαμέρισμα της Ελλάδας, τους μήνες και το έτος	65
Εικόνα 62: Γράφημα περιοχής (Area chart) όπου οπτικοποιούνται τα κρούσματα ανά έτος με φίλτρο το γεωγραφικό διαμέρισμα της Ελλάδας	66

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Πλεονεκτήματα και Μειονεκτήματα Οπτικοποιήσεων Μεγάλων Δεδομένων για τη COVID-19 στην Ελλάδα.....	14
--	----

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία αποτελεί μέρος των υποχρεώσεων για τη λήψη μεταπτυχιακού διπλώματος στο Τμήμα Πληροφορικής και Τηλεπικοινωνιών του Εθνικού Καποδιστριακού Πανεπιστημίου Αθηνών. Η εργασία εκπονήθηκε κατά τη διάρκεια του ακαδημαϊκού έτους 2020-2021 με επιβλέπουσα την επίκουρη καθηγήτρια Μαρία Ρούσσου και μέλη επιτροπής τον επίκουρο καθηγητή Αλέξανδρο Ντούλα και τον καθηγητή Δημήτριο Γουνόπουλο.

1. ΕΙΣΑΓΩΓΗ

Σε αυτό το κεφάλαιο θα παρουσιαστούν πληροφορίες για τη λοίμωξη COVID-19, τα μεγάλα δεδομένα (Big Data) καθώς και για την οπτικοποίηση αυτών.

1.1 Λοίμωξη COVID-19: Ορισμός και παρούσα κατάσταση ανά τον πλανήτη

Η λοίμωξη COVID-19 είναι μια οξεία λοίμωξη του αναπνευστικού που προκαλείται από τον πρωτοεμφανιζόμενο κορονοϊό SARS-CoV2. Πρωτοεμφανίστηκε στα τέλη του 2019 στην περιοχή της Wuhan, στην Κίνα και έχει έκτοτε λάβει διαστάσεις πανδημίας [1]. Με βάση τον ιστοχώρο του Παγκόσμιου Οργανισμού Υγείας (World Health Organization-W.H.O.) μέχρι και τις 27 Απριλίου του 2021 είχαν καταγραφεί 146.067.511 κρούσματα σε όλο τον πλανήτη και συνολικά 3.092.497 θάνατοι από τη νόσο [2].

Στόχος της παρούσας εργασίας είναι αφενός η συλλογή μεγάλων δεδομένων για την COVID-19 σε περιοχές της Ελλάδας μέσα από ελληνικούς ειδησεογραφικούς ιστοχώρους, όπως in.gr, tonima.gr, tanea.gr, protothema.gr έτσι ώστε να είναι δυνατή η εξαγωγή πληροφοριών για τα κρούσματα COVID-19 στην Ελλάδα (με βάση την ημερομηνία, την τοποθεσία και τον αριθμό των κρουσμάτων). Αφετέρου, είναι η δημιουργία διαδραστικών οπτικοποιήσεων για να παρουσιαστούν τα παραπάνω δεδομένα με τρόπο όσο γίνεται πιο κατανοητό στο ευρύ κοινό.

1.2 Ορισμός των Μεγάλων Δεδομένων

Τα μεγάλα δεδομένα αναφέρονται σε μεγάλα σύνολα δεδομένων (μεγάλους όγκους δεδομένων). Πάνω σε μεγάλα δεδομένα μπορούμε να εφαρμόσουμε διεργασίες όπως ανάλυση, οπτικοποίηση, αναζήτηση, μετάδοση, αποθήκευση κ.ά. Τα μεγάλα δεδομένα είναι τόσο μεγάλα και σύνθετα που δεν μπορούν να χρησιμοποιηθούν παραδοσιακά λογισμικά επεξεργασίας δεδομένων για τη διαχείρισή τους. [3]

1.3 Τύποι των Μεγάλων Δεδομένων

Οι βασικοί τύποι των Μεγάλων Δεδομένων που έχουμε είναι οι παρακάτω [3]:

- Σχεσιακά δεδομένα (Πίνακες/Συναλλαγές)
- Πολυμεσικά δεδομένα (εικόνα, ήχος, κείμενο, βίντεο)
- Δεδομένα μορφής γραφημάτων (Graph Data) όπως το Web
- Σημασιολογικός Ιστός
- Κοινωνικά δίκτυα
- Streaming Data
- Δομημένα δεδομένα
- Ημιδομημένα δεδομένα(XML)

Πίνακας 1: Πλεονεκτήματα και Μειονεκτήματα Οπτικοποιήσεων Μεγάλων Δεδομένων για τη COVID-19 στην Ελλάδα

ΤΙΤΛΟΣ ΟΠΤΙΚΟΠΟΙΗΣΗΣ	ΟΡΓΑΝΙΣΜΟΣ	ΠΛΕΟΝΕΚΤΗΜΑΤΑ	ΜΕΙΟΝΕΚΤΗΜΑΤΑ
Χάρτης Υγειονομικής Ασφάλειας και Προστασίας από τη λοίμωξη covid-19 [5]	Ελληνική Κυβέρνηση	Δυνατότητα μεγέθυνσης στην οπτικοποίηση και επικέντρωση στο επίπεδο κινδύνου κάθε νομού ξεχωριστά.	Δεν υπάρχουν μειονεκτήματα στην οπτικοποίηση.
Κρούσματα [6]	Ελληνική Κυβέρνηση-Υπουργείο Υγείας	Δυνατότητα τοποθέτησης κέρσora πάνω σε τελεία της οπτικοποίησης για ένδειξη των κρουσμάτων την ημερομηνία που αντιστοιχεί στην εκάστοτε επιλεγθείσα τελεία.	Καταγράφονται αρκετές ημερομηνίες, με αποτέλεσμα να υπάρχει πληθώρα τελειών στην οπτικοποίηση. Ο αριθμός των τελειών είναι πιο μεγάλος από τον αριθμό των ημερομηνιών που είναι γραμμένες στον άξονα των x.
Απώλειες [6]	Ελληνική Κυβέρνηση-Υπουργείο Υγείας	Δυνατότητα τοποθέτησης κέρσora πάνω σε τελεία της οπτικοποίησης για ένδειξη των απωλειών την ημερομηνία που αντιστοιχεί στην εκάστοτε επιλεγθείσα τελεία.	Καταγράφονται αρκετές ημερομηνίες, με αποτέλεσμα να υπάρχει πληθώρα τελειών στην οπτικοποίηση. Ο αριθμός των τελειών είναι πιο μεγάλος από τον αριθμό των ημερομηνιών που είναι γραμμένες στον άξονα των x.
Διασωληνωμένοι [6]	Ελληνική Κυβέρνηση-Υπουργείο Υγείας	Δυνατότητα τοποθέτησης κέρσora πάνω σε τελεία της οπτικοποίησης για ένδειξη των διασωληνωμένων την ημερομηνία που αντιστοιχεί στην	Καταγράφονται αρκετές ημερομηνίες, με αποτέλεσμα να υπάρχει πληθώρα τελειών στην οπτικοποίηση. Ο αριθμός των τελειών είναι πιο

		εκάστοτε επιλεχθείσα τελεία.	μεγάλος από τον αριθμό των ημερομηνιών που είναι γραμμένες στον άξονα των x.
Ηλικιακή κατανομή κρουσμάτων –Απώλειες [6]	Ελληνική Κυβέρνηση-Υπουργείο Υγείας	Δυνατότητα τοποθέτησης κέρσορα πάνω σε τομέα της οπτικοποίησης (διάγραμμα πίτας) για ένδειξη των απωλειών για την ηλικιακή ομάδα που αντιστοιχεί στον επιλεχθέντα τομέα.	Δεν υπάρχουν μειονεκτήματα στην οπτικοποίηση.
Ηλικιακή κατανομή κρουσμάτων –Κρούσματα [6]	Ελληνική Κυβέρνηση-Υπουργείο Υγείας	Δυνατότητα τοποθέτησης κέρσορα πάνω σε τομέα της οπτικοποίησης (διάγραμμα πίτας) για ένδειξη των κρουσμάτων για την ηλικιακή ομάδα που αντιστοιχεί στον επιλεχθέντα τομέα.	Δεν υπάρχουν μειονεκτήματα στην οπτικοποίηση.
Ηλικιακή κατανομή κρουσμάτων-Διασωληνωμένοι [6]	Ελληνική Κυβέρνηση-Υπουργείο Υγείας	Δυνατότητα τοποθέτησης κέρσορα πάνω σε τομέα της οπτικοποίησης (διάγραμμα πίτας) για ένδειξη των διασωληνωμένων για την ηλικιακή ομάδα που αντιστοιχεί στον επιλεχθέντα τομέα.	Δεν υπάρχουν μειονεκτήματα στην οπτικοποίηση.
Κρούσματα ανά Χώρα/Περιοχή/Κυριαρχία (Cases by Country/Region/Sovereignty) [7]	Johns Hopkins Coronavirus Resource Center	Εύκολο προς κατανόηση για τον χρήστη.	Δεν υπάρχουν μειονεκτήματα στην οπτικοποίηση.
Ημερήσια Κρούσματα (Daily Cases) [7]	Johns Hopkins Coronavirus Resource Center	Με την τοποθέτηση του κέρσορα στο εσωτερικό του γραφήματος περιοχής (area	Στον άξονα των x εμφανίζονται μόνον μήνες και έτη. Η τοποθέτηση του κέρσορα καθορίζει

		chart) εμφανίζονται τα κρούσματα ανά ημερομηνία.	την πλήρη ημερομηνία, με αποτέλεσμα να δυσκολεύει τον χρήστη η αναζήτηση ακριβούς ημερομηνίας.
Καθημερινά νέα επιβεβαιωμένα κρούσματα COVID-19 (Daily new confirmed COVID-19 cases) [8]	Oxford Martin School	Με την τοποθέτηση του κέρσορα στο γράφημα γραμμής (line chart) αναφέρεται ο μέσος αριθμός των επιβεβαιωμένων κρουσμάτων ανά ημερομηνία.	Δεν αναγράφονται όλες οι ημερομηνίες στον άξονα των x, μολονότι υπάρχει δυνατότητα εύρεσης του μέσου αριθμού των επιβεβαιωμένων κρουσμάτων ανά συγκεκριμένη ημερομηνία με κύλιση της αντίστοιχης μπάρας από το χρήστη.
Καθημερινά νέα επιβεβαιωμένα κρούσματα COVID-19 ανά εκατομμύριο ανθρώπων (Daily new confirmed COVID-19 cases per million people) [8]	Oxford Martin School	Με την τοποθέτηση του κέρσορα πάνω σε οποιαδήποτε χώρα χώρα, αναφέρεται το μέσος αριθμός των επιβεβαιωμένων κρουσμάτων στη συγκεκριμένη χώρα. Υπάρχει, επίσης, δυνατότητα επιλογής ηπείρου ή όλης της υφελίου.	Δεν υπάρχουν μειονεκτήματα στην οπτικοποίηση.
Σωρευτικά επιβεβαιωμένα κρούσματα COVID-19 (Cumulative confirmed COVID-19 cases) [8]	Oxford Martin School	Με την τοποθέτηση του κέρσορα στο γράφημα γραμμής (line chart) αναφέρεται ο σωρευτικός αριθμός των επιβεβαιωμένων κρουσμάτων ανά ημερομηνία.	Δεν αναγράφονται όλες οι ημερομηνίες στον άξονα των x, μολονότι υπάρχει δυνατότητα εύρεσης του μέσου αριθμού των επιβεβαιωμένων κρουσμάτων ανά συγκεκριμένη ημερομηνία με κύλιση της αντίστοιχης

			μπάρας από το χρήστη.
Σωρευτικά επιβεβαιωμένα κρούσματα COVID-19 ανά εκατομμύριο ανθρώπων (Cumulative confirmed COVID-19 cases per million people) [8]	Oxford Martin School	Με την τοποθέτηση του κέρσορα πάνω σε οποιαδήποτε χώρα χώρα, αναφέρεται το μέσος αριθμός των επιβεβαιωμένων κρουσμάτων στη συγκεκριμένη χώρα. Υπάρχει, επίσης, δυνατότητα επιλογής ηπείρου ή όλης της υφελίου.	Δεν υπάρχουν μειονεκτήματα στην οπτικοποίηση.

1.6 Διάρθρωση εργασίας

Στο κεφάλαιο 1 αναφέρονται ο ορισμός και οι τύποι των μεγάλων δεδομένων, ο ορισμός της οπτικοποίησης δεδομένων και πληροφορίες για ορισμένες οπτικοποιήσεις.

Στο κεφάλαιο 2, καταγράφονται τα στάδια με βάση τα οποία η παρούσα διπλωματική εργασία έχει δομηθεί.

Στο κεφάλαιο 3 παρουσιάζεται η υλοποιήσεις των παραπάνω σταδίων.

Τέλος, στο κεφάλαιο 4 αναφέρονται τα αποτελέσματα των υλοποιήσεων των σταδίων που αναλύθηκαν στο κεφάλαιο 3.

2. ΣΤΑΔΙΑ ΤΗΣ ΔΙΠΛΩΜΑΤΙΚΗΣ ΕΡΓΑΣΙΑΣ

Σε αυτό το κεφάλαιο θα αναφερθούν τα στάδια στα οποία η διπλωματική εργασία έχει δομηθεί.

2.1 Συλλογή Ιστοσελίδων (Web Crawling)

Στο στάδιο αυτό πραγματοποιήθηκε συλλογή (crawling) σελίδων από ευρέως χρησιμοποιούμενους ελληνικούς ειδησεογραφικούς ιστοχώρους (sites) όπως in.gr, tonima.gr, tanea.gr, protothema.gr, οι οποίες και αποθηκεύτηκαν σε αρχεία τύπου .csv.

2.2 Αποθήκευση περιεχομένων σελίδων

Στο στάδιο αυτό τα περιεχόμενα των σελίδων για τις οποίες εκτελέστηκε συλλογή σελίδων (crawling) αποθηκεύτηκαν σε αρχεία html.

2.3 Εξαγωγή πληροφοριών

Στο στάδιο αυτό εκτελέστηκε ο παρακάτω αλγόριθμος:

Για κάθε html αρχείο κάνε:

 ψάξε να βρεις τη λέξη κρούσμα ή κρούσματα.

 αν τη βρεις τότε:

 ψάξε να βρεις την ημερομηνία του άρθρου

 κοντά στη λέξη κρούσμα/κρούσματα βρες έναν αριθμό που θα είναι αυτός από τα κρούσματα

 κοντά στη λέξη κρούσμα/κρούσματα βρες το όνομα μιας περιοχής της Ελλάδας.

Τα αποτελέσματα του παραπάνω αλγορίθμου αποθηκεύτηκαν σε αρχείο .csv της μορφής: ημερομηνία, τοποθεσία, κρούσματα

2.4 Οπτικοποίηση Δεδομένων

Στο στάδιο αυτό τα δεδομένα του αρχείου που δημιουργήθηκε στο στάδιο Εξαγωγή Πληροφοριών οπτικοποιήθηκαν.

3. ΥΛΟΠΟΙΗΣΕΙΣ ΤΩΝ ΣΤΑΔΙΩΝ ΤΗΣ ΕΡΓΑΣΙΑΣ

Στο κεφάλαιο αυτό θα αναφερθούν οι υλοποιήσεις των σταδίων της εργασίας στη γλώσσα προγραμματισμού Python με τη χρήση του εργαλείου προγραμματισμού Jupyter Notebook.

Η γλώσσα Python προτιμήθηκε έναντι άλλων γλωσσών προγραμματισμού, όπως η C και η Java, επειδή είναι μια γλώσσα απλή και κατανοητή, δεν χρειάζεται πολλές γραμμές κώδικα και χρησιμοποιείται κατεξοχήν για τη διαχείριση δεδομένων. Αντίστοιχα, το εργαλείο προγραμματισμού Jupyter Notebook είναι ευρέως διαδεδομένο στην αγορά εργασίας, προσφέρεται ιδιαίτερα για τη δημιουργία διαδραστικών οπτικοποιήσεων και χρησιμοποιείται και αυτό ιδιαίτερα συχνά για τη διαχείριση δεδομένων.

3.1 Κώδικας που κάνει συλλογή (crawling) σελίδων

Ο κώδικας που κάνει συλλογή (crawling) σελίδων από ελληνικές ειδησεογραφικές ιστοσελίδες είναι ο κώδικας της Εικόνας 2.

```

1  from bs4 import BeautifulSoup
2  import requests
3  import re
4  import pandas as pd
5  import csv
6  import sys
7
8  def url_in_file(filename, url):
9      df = pd.read_csv(filename, encoding='cp737', engine='python')
10     return url in list(df['Page URL'])
11
12 def pages_in_file(filename):
13     df = pd.read_csv(filename, encoding='cp737', engine='python')
14     return df.shape[0]
15
16 def getPages(prefix, url, filename):
17     req = requests.get(url)
18     soup = BeautifulSoup(req.content, 'html.parser')
19     reg = "^" + prefix
20     pages = soup.findAll("a", attrs={'href': re.compile(reg)})
21     for page in pages:
22         pageUrl = page.get("href")
23         if (not url_in_file(filename, pageUrl)):
24             print pageUrl
25             with open(filename, 'ab') as newFile:
26                 newFileWriter = csv.writer(newFile)
27                 newFileWriter.writerow([pageUrl])
28             getPages(prefix, pageUrl, filename)
29
30 def main():
31     reload(sys)
32     sys.setdefaultencoding("utf-8")
33     url = "https://www.protothema.gr/"
34     site = url[12:-1]
35     prefix = url
36     filename = site + "_pages.csv"
37     pathname = "Datasets/" + filename
38     #with open(pathname, 'wb') as newFile:
39     #    newFileWriter = csv.writer(newFile)
40     #    newFileWriter.writerow(['Page URL'])
41     getPages(prefix, url, pathname)
42     pages = pages_in_file(pathname)
43     print "Crawled " + str(pages) + " pages from " + site + "."
44
45 main()

```

Εικόνα 2: Κώδικας που κάνει συλλογή (crawl) σελίδων

Οι συναρτήσεις που υλοποιήθηκαν στον παραπάνω κώδικα εξηγούνται κάτωθι:

- `url_in_file`: Η συνάρτηση αυτή ελέγχει αν το `url` που παίρνει σαν όρισμα είναι γραμμένο στο αρχείο με το όνομα που δίνεται σαν όρισμα.
- `pages_in_file`: Η συνάρτηση αυτή επιστρέφει το πλήθος των σελίδων που γράφτηκαν στο αρχείο με το όνομα που δίνεται σαν όρισμα.
- `getPages`: Η συνάρτηση αυτή κάνει συλλογή (*crawling*) σελίδων από το `url` που δίνεται σαν όρισμα και τις γράφει στο αρχείο με το όνομα που δίνεται σαν όρισμα.
- `main`: Η κύρια συνάρτηση του κώδικα. Στη συνάρτηση αυτή γράφεται το `url` του ιστοχώρου από τον οποίο υπάρχει η επιθυμία να πραγματοποιηθεί ανίχνευση σελίδων. Στη συνέχεια, ορίζεται το όνομα του αρχείου στο οποίο τα `urls` των σελίδων που ανιχνεύονται προορίζονται να γραφούν. Καλείται η συνάρτηση με την οποία πραγματοποιείται ανίχνευση σελίδων. Υπάρχουν σχόλια σε κάποια τμήματα του κώδικα για να μη δημιουργείται ξανά αρχείο από την αρχή σε περίπτωση που υπάρχει ανάγκη νέας εκτέλεσης του κώδικα. Με τη συγκεκριμένη αυτή συνάρτηση τυπώνεται ο ακριβής αριθμός των σελίδων οι οποίες ανιχνεύθηκαν από τους ελληνικούς ειδησεογραφικούς ιστοχώρους (*sites*).

3.2 Κώδικας που αποθηκεύει περιεχόμενα σελίδων

Ο κώδικας που αποθηκεύει τα περιεχόμενα των σελίδων που συλλέχθηκαν σε αρχεία `html` είναι ο κώδικας των Εικόνων 3 και 4.

```

1  from bs4 import BeautifulSoup
2  import requests
3  import pandas as pd
4  import sys
5  import csv
6
7  def page_with_textual_data(url):
8      return "wp-content/uploads" not in url
9
10 def result_request_page(url):
11     try:
12         req = requests.get(url)
13         return True
14     except:
15         return False
16
17 def save_contents_to_html_file(filename, filename2, prefix):
18     df1 = pd.read_csv(filename, encoding='cp737', engine='python')
19     df2 = pd.read_csv(filename2, encoding='cp737', engine='python')
20     set1 = set(df1['Page URL'])
21     set2 = set(df2['Page URL'])
22     urls = list(set1 - set2)
23     pages_saved = df2.shape[0]
24     for url in urls:
25         if (page_with_textual_data(url)):
26             if (not result_request_page(url)):
27                 continue
28             else:
29                 pages_saved = pages_saved + 1
30                 req = requests.get(url)
31                 soup = BeautifulSoup(req.content, 'html.parser')
32                 with open(filename2, 'ab') as newFile:
33                     newFileWriter = csv.writer(newFile)
34                     newFileWriter.writerow([url])
35                 html_file = prefix + "content" + str(pages_saved) + ".html"
36                 folder = prefix + "files"
37                 pathname = folder + "/" + html_file
38                 with open(pathname, "w") as file:
39                     file.write(str(soup.body))
40
41 def main():
42     reload(sys)
43     sys.setdefaultencoding("utf-8")
44     filename = "protothema.gr_pages.csv"
45     i = filename.index("_")
46     prefix = filename[0:i + 1]
47     filename2 = prefix + "pages_saved.csv"
48     pathname = "Datasets/" + filename
49     pathname2 = "Datasets/" + filename2

```

Εικόνα 3: Κώδικας που αποθηκεύει περιεχόμενα σελίδων(1-49)

```

50     #with open(pathname2, 'wb') as newFile:
51         #newFileWriter = csv.writer(newFile)
52         #newFileWriter.writerow(['Page URL'])
53     save_contents_to_html_file(pathname, pathname2, prefix)
54     df = pd.read_csv(pathname2, encoding='cp737', engine='python')
55     for i in range(0, df.shape[0]):
56         url = df.iloc[i, 0]
57         print "The content of " + url + " was saved successfully."
58
59 main()

```

Εικόνα 4: Κώδικας που αποθηκεύει περιεχόμενα σελίδων (50-59)

Οι συναρτήσεις που υλοποιήθηκαν στον παραπάνω κώδικα εξηγούνται κάτωθι:

- `page_with_textual_data`: Η συνάρτηση αυτή ελέγχει αν η σελίδα με το url που δίνεται σαν όρισμα περιλαμβάνει δεδομένα μορφής κειμένου.
- `result_request_page`: Η συνάρτηση αυτή χειρίζεται εξαιρέσεις (exceptions) όταν βρίσκει το αποτέλεσμα της αίτησης της σελίδας με το url που δίνεται σαν όρισμα.
- `save_contents_to_html_file`: Με τη συγκεκριμένη συνάρτηση αποθηκεύονται περιεχόμενα σελίδων που περιλαμβάνουν δεδομένα μορφής κειμένου σε αρχεία html και γράφονται στο αρχείο `filename2` που δίνεται σαν όρισμα τα urls των σελίδων των οποίων αποθηκεύτηκαν τα περιεχόμενα.
- `main`: Η κύρια συνάρτηση του κώδικα. Στη συνάρτηση αυτή ορίζεται το όνομα του αρχείου στο οποίο γράφτηκαν τα urls των σελίδων που ανιχνεύθηκαν και το όνομα του αρχείου στο οποίο γράφονται τα urls των σελίδων των οποίων αποθηκεύτηκαν τα περιεχόμενα. Αυτό γίνεται για να μην αποθηκευτούν ξανά περιεχόμενα σελίδων που ήδη αποθηκεύτηκαν σε περίπτωση που ο κώδικας εκτελεστεί εκ νέου. Καλείται η συνάρτηση που αποθηκεύει περιεχόμενα σελίδων. Υπάρχουν σχόλια σε κάποια τμήματα του κώδικα για να μη δημιουργείται ξανά αρχείο από την αρχή σε περίπτωση που ο κώδικας εκτελεστεί εκ νέου. Με τη συνάρτηση αυτή τυπώνονται οι ακριβείς σελίδες των οποίων τα περιεχόμενα αποθηκεύτηκαν.

3.3 Κώδικας που επεξεργάζεται ελληνικές τοποθεσίες

Ανάμεσα στα στάδια Αποθήκευση περιεχομένων σελίδων και Εξαγωγή Πληροφοριών γράφτηκε κώδικας που επεξεργάζεται ελληνικές τοποθεσίες. Αυτό προτιμήθηκε για να μη γίνεται συνέχεια η επεξεργασία των ελληνικών τοποθεσιών όσο τρέχει το πρόγραμμα που εκτελεί εξαγωγή πληροφοριών, το οποίο θα εξηγηθεί στην επόμενη ενότητα. Τα αρχεία κώδικα που υπάρχουν στο πρόγραμμα που επεξεργάζεται ελληνικές τοποθεσίες είναι τα παρακάτω:

`create_dictionary_counties.py`

`create_dictionary_municipalities.py`

`find_locations.py`

`main.py`

Ο κώδικας που είναι γραμμένος στο αρχείο `create_dictionary_counties.py` είναι ο κώδικας των Εικόνων 5, 6 και 7.

```

1  #-*- coding: utf-8 -*-
2
3  import sys
4
5  def create_dictionary_counties(counties):
6      reload(sys)
7      sys.setdefaultencoding("utf-8")
8      dictionary_counties = {}
9      for county in counties:
10         if (county == 'Εβρου'):
11             dictionary_counties.update({'Εβρου': 'Εβρος', 'Εβρος': 'Εβρος', 'Εβρο': 'Εβρος'})
12         elif (county == 'Ροδόπη'):
13             dictionary_counties.update({'Ροδόπη': 'Ροδόπη', 'Ροδόπη': 'Ροδόπη'})
14         elif (county == 'Εάνθη'):
15             dictionary_counties.update({'Εάνθη': 'Εάνθη', 'Εάνθη': 'Εάνθη'})
16         elif (county == 'Καβάλα'):
17             dictionary_counties.update({'Καβάλα': 'Καβάλα', 'Καβάλα': 'Καβάλα'})
18         elif (county == 'Δράμα'):
19             dictionary_counties.update({'Δράμα': 'Δράμα', 'Δράμα': 'Δράμα'})
20         elif (county == 'Σερρών'):
21             dictionary_counties.update({'Σερρών': 'Σέρρες', 'Σέρρες': 'Σέρρες'})
22         elif (county == 'Κιλκίς'):
23             dictionary_counties.update({'Κιλκίς': 'Κιλκίς'})
24         elif (county == 'Θεσσαλονίκη'):
25             dictionary_counties.update({'Θεσσαλονίκη': 'Θεσσαλονίκη', 'Θεσσαλονίκη': 'Θεσσαλονίκη'})
26         elif (county == 'Χαλκιδική'):
27             dictionary_counties.update({'Χαλκιδική': 'Χαλκιδική', 'Χαλκιδική': 'Χαλκιδική'})
28         elif (county == 'Πιερίας'):
29             dictionary_counties.update({'Πιερίας': 'Πιερία', 'Πιερία': 'Πιερία'})
30         elif (county == 'Ημαθίας'):
31             dictionary_counties.update({'Ημαθίας': 'Ημαθία', 'Ημαθία': 'Ημαθία'})
32         elif (county == 'Πέλλα'):
33             dictionary_counties.update({'Πέλλα': 'Πέλλα', 'Πέλλα': 'Πέλλα'})
34         elif (county == 'Φλώρινα'):
35             dictionary_counties.update({'Φλώρινα': 'Φλώρινα', 'Φλώρινα': 'Φλώρινα'})
36         elif (county == 'Καστοριά'):
37             dictionary_counties.update({'Καστοριά': 'Καστοριά', 'Καστοριά': 'Καστοριά'})
38         elif (county == 'Κοζάνη'):
39             dictionary_counties.update({'Κοζάνη': 'Κοζάνη', 'Κοζάνη': 'Κοζάνη'})
40         elif (county == 'Τρεβενά'):
41             dictionary_counties.update({'Τρεβενά': 'Τρεβενά', 'Τρεβενά': 'Τρεβενά'})
42         elif (county == 'Ιωάννινα'):
43             dictionary_counties.update({'Ιωάννινα': 'Ιωάννινα', 'Ιωάννινα': 'Ιωάννινα', 'Τιάννενα': 'Ιωάννινα'})
44         elif (county == 'Θεσπρωτία'):
45             dictionary_counties.update({'Θεσπρωτία': 'Θεσπρωτία', 'Θεσπρωτία': 'Θεσπρωτία'})
46         elif (county == 'Πρέβεζα'):
47             dictionary_counties.update({'Πρέβεζα': 'Πρέβεζα', 'Πρέβεζα': 'Πρέβεζα'})
48         elif (county == 'Αρτα'):
49             dictionary_counties.update({'Αρτα': 'Αρτα', 'Αρτα': 'Αρτα'})

```

Εικόνα 5: Κώδικας που επεξεργάζεται νομούς (1-49)

```

50         elif (county == 'Τρικάλων'):
51             dictionary_counties.update({'Τρικάλων': 'Τρίκοια', 'Τρίκοια': 'Τρίκοια'})
52         elif (county == 'Καρδίτσα'):
53             dictionary_counties.update({'Καρδίτσα': 'Καρδίτσα', 'Καρδίτσα': 'Καρδίτσα'})
54         elif (county == 'Λάρισα'):
55             dictionary_counties.update({'Λάρισα': 'Λάρισα', 'Λάρισα': 'Λάρισα'})
56         elif (county == 'Μαγνησία'):
57             dictionary_counties.update({'Μαγνησία': 'Μαγνησία', 'Μαγνησία': 'Μαγνησία'})
58         elif (county == 'Εύβοια'):
59             dictionary_counties.update({'Εύβοια': 'Εύβοια', 'Εύβοια': 'Εύβοια'})
60         elif (county == 'Βοιωτία'):
61             dictionary_counties.update({'Βοιωτία': 'Βοιωτία', 'Βοιωτία': 'Βοιωτία'})
62         elif (county == 'Φθιώτιδα'):
63             dictionary_counties.update({'Φθιώτιδα': 'Φθιώτιδα', 'Φθιώτιδα': 'Φθιώτιδα'})
64         elif (county == 'Φωκίδα'):
65             dictionary_counties.update({'Φωκίδα': 'Φωκίδα', 'Φωκίδα': 'Φωκίδα'})
66         elif (county == 'Ευρυτανία'):
67             dictionary_counties.update({'Ευρυτανία': 'Ευρυτανία', 'Ευρυτανία': 'Ευρυτανία'})
68         elif (county == 'Αιτωλοακαρνανία'):
69             dictionary_counties.update({'Αιτωλοακαρνανία': 'Αιτωλοακαρνανία', 'Αιτωλοακαρνανία': 'Αιτωλοακαρνανία'})
70         elif (county == 'Αχαΐα'):
71             dictionary_counties.update({'Αχαΐα': 'Αχαΐα', 'Αχαΐα': 'Αχαΐα'})
72         elif (county == 'Ηλεία'):
73             dictionary_counties.update({'Ηλεία': 'Ηλεία', 'Ηλεία': 'Ηλεία'})
74         elif (county == 'Κέρκυρα'):
75             dictionary_counties.update({'Κέρκυρα': 'Κέρκυρα', 'Κέρκυρα': 'Κέρκυρα'})
76         elif (county == 'Λευκάδα'):
77             dictionary_counties.update({'Λευκάδα': 'Λευκάδα', 'Λευκάδα': 'Λευκάδα'})
78         elif (county == 'Κεφαλληνία'):
79             dictionary_counties.update({'Κεφαλληνία': 'Κεφαλλονιά', 'Κεφαλλονιά': 'Κεφαλλονιά', 'Κεφαλλονιά': 'Κεφαλλονιά',
80 'Κεφαλλονιά': 'Κεφαλλονιά'})
81         elif (county == 'Ζακύνθος'):
82             dictionary_counties.update({'Ζακύνθος': 'Ζάκυνθος', 'Ζάκυνθος': 'Ζάκυνθος', 'Ζάκυνθο': 'Ζάκυνθος'})
83         elif (county == 'Αττική'):
84             dictionary_counties.update({'Αττική': 'Αττική', 'Αττική': 'Αττική'})
85         elif (county == 'Κορινθία'):
86             dictionary_counties.update({'Κορινθία': 'Κόρινθος', 'Κορινθία': 'Κόρινθος', 'Κόρινθος': 'Κόρινθος', 'Κορίνθο': 'Κόρινθος',
87 'Κόρινθο': 'Κόρινθος'})
88         elif (county == 'Αργολίδα'):
89             dictionary_counties.update({'Αργολίδα': 'Αργολίδα', 'Αργολίδα': 'Αργολίδα'})
90         elif (county == 'Αρκαδία'):
91             dictionary_counties.update({'Αρκαδία': 'Αρκαδία', 'Αρκαδία': 'Αρκαδία'})
92         elif (county == 'Μεσσηνία'):
93             dictionary_counties.update({'Μεσσηνία': 'Μεσσηνία', 'Μεσσηνία': 'Μεσσηνία'})

```

Εικόνα 6: Κώδικας που επεξεργάζεται νομούς (50-91)

```

92     elif (county == 'Λακωνίας'):
93         dictionary_counties.update({'Λακωνίας': 'Λακωνία', 'Λακωνία': 'Λακωνία'})
94     elif (county == 'Λήμνου'):
95         dictionary_counties.update({'Λήμνου': 'Λήμνος', 'Λήμνος': 'Λήμνος', 'Λήμνο': 'Λήμνος'})
96     elif (county == 'Χίου'):
97         dictionary_counties.update({'Χίου': 'Χίος', 'Χίος': 'Χίος', 'Χίο': 'Χίος'})
98     elif (county == 'Σάμου'):
99         dictionary_counties.update({'Σάμου': 'Σάμος', 'Σάμος': 'Σάμος', 'Σάμο': 'Σάμος'})
100    elif (county == 'Κυκλάδων'):
101        dictionary_counties.update({'Κυκλάδων': 'Κυκλάδες', 'Κυκλάδες': 'Κυκλάδες'})
102    elif (county == 'Δωδεκανήσου'):
103        dictionary_counties.update({'Δωδεκανήσου': 'Δωδεκάνησα', 'Δωδεκανήσων': 'Δωδεκάνησα', 'Δωδεκάνησα': 'Δωδεκάνησα'})
104    elif (county == 'Λασιθίου'):
105        dictionary_counties.update({'Λασιθίου': 'Λασιθί', 'Λασιθί': 'Λασιθί'})
106    elif (county == 'Ηρακλείου'):
107        dictionary_counties.update({'Ηρακλείου': 'Ηράκλειο', 'Ηράκλειο': 'Ηράκλειο'})
108    elif (county == 'Ρεθύμνης'):
109        dictionary_counties.update({'Ρεθύμνης': 'Ρέθυμνο', 'Ρεθύμνου': 'Ρέθυμνο', 'Ρέθυμνο': 'Ρέθυμνο'})
110    elif (county == 'Χανίων'):
111        dictionary_counties.update({'Χανίων': 'Χανιά', 'Χανιά': 'Χανιά'})
112    return dictionary_counties

```

Εικόνα 7: Κώδικας που επεξεργάζεται νομούς (92-112)

Η συνάρτηση `create_dictionary_counties` επεξεργάζεται τους νομούς και τους αποθηκεύει σε ένα dictionary.

Ο κώδικας που είναι γραμμένος στο αρχείο `create_dictionary_municipalities.py` είναι ο κώδικας των Εικόνων 8-24.

```

1  # -*- coding: utf-8 -*-
2
3  import sys
4
5  def create_dictionary_municipalities(municipalities):
6      reload(sys)
7      sys.setdefaultencoding("utf-8")
8      dictionary_municipalities = {}
9      for municipality in municipalities:
10         if (municipality == 'Ορεστιάδας'):
11             dictionary_municipalities.update({'Ορεστιάδας': 'Ορεστιάδα', 'Ορεστιάδα': 'Ορεστιάδα'})
12         elif (municipality == 'Διδυμοτείχου'):
13             dictionary_municipalities.update({'Διδυμοτείχου': 'Διδυμότειχο', 'Διδυμότειχο': 'Διδυμότειχο'})
14         elif (municipality == 'Σουφλίου'):
15             dictionary_municipalities.update({'Σουφλίου': 'Σουφλί', 'Σουφλί': 'Σουφλί'})
16         elif (municipality == 'Αλεξανδρούπολης'):
17             dictionary_municipalities.update({'Αλεξανδρούπολης': 'Αλεξανδρούπολη', 'Αλεξανδρούπολη': 'Αλεξανδρούπολη'})
18         elif (municipality == 'Σομοθράκης'):
19             dictionary_municipalities.update({'Σομοθράκης': 'Σομοθράκη', 'Σομοθράκη': 'Σομοθράκη'})
20         elif (municipality == 'Αρριανών'):
21             dictionary_municipalities.update({'Αρριανών': 'Αρριανά', 'Αρριανά': 'Αρριανά'})
22         elif (municipality == 'Μαρωνείας-Σοπών'):
23             dictionary_municipalities.update({'Μαρωνείας-Σοπών': 'Μαρωνεία-Σάπες', 'Μαρωνεία-Σάπες': 'Μαρωνεία-Σάπες',
24             'Μαρωνεία': 'Μαρωνεία-Σάπες', 'Σάπες': 'Μαρωνεία-Σάπες', 'Μαρωνείας': 'Μαρωνεία-Σάπες', 'Σοπών': 'Μαρωνεία-Σάπες'})
25         elif (municipality == 'Κομοτηνής'):
26             dictionary_municipalities.update({'Κομοτηνής': 'Κομοτηνή', 'Κομοτηνή': 'Κομοτηνή'})
27         elif (municipality == 'Ιάσμου'):
28             dictionary_municipalities.update({'Ιάσμου': 'Ιασμος', 'Ιασμος': 'Ιασμος', 'Ιασμο': 'Ιασμος'})
29         elif (municipality == 'Αβδηρών'):
30             dictionary_municipalities.update({'Αβδηρών': 'Αβδηρα', 'Αβδηρα': 'Αβδηρα'})
31         elif (municipality == 'Μύκης'):
32             dictionary_municipalities.update({'Μύκης': 'Μύκη', 'Μύκη': 'Μύκη'})
33         elif (municipality == 'Βάνθης'):
34             dictionary_municipalities.update({'Βάνθης': 'Βάνθη', 'Βάνθη': 'Βάνθη'})
35         elif (municipality == 'Τοπίρου'):
36             dictionary_municipalities.update({'Τοπίρου': 'Τόπειρος', 'Τόπειρος': 'Τόπειρος', 'Τόπειρο': 'Τόπειρος'})
37         elif (municipality == 'Νέστου'):
38             dictionary_municipalities.update({'Νέστου': 'Νέστος', 'Νέστος': 'Νέστος', 'Νέστο': 'Νέστος'})
39         elif (municipality == 'Καβάλας'):
40             dictionary_municipalities.update({'Καβάλας': 'Καβάλα', 'Καβάλα': 'Καβάλα'})
41         elif (municipality == 'Παγγαίου'):
42             dictionary_municipalities.update({'Παγγαίου': 'Παγγαίο', 'Παγγαίο': 'Παγγαίο'})
43         elif (municipality == 'Θάσου'):
44             dictionary_municipalities.update({'Θάσου': 'Θάσος', 'Θάσος': 'Θάσος', 'Θάσο': 'Θάσος'})
45         elif (municipality == 'Δοξάτου'):
46             dictionary_municipalities.update({'Δοξάτου': 'Δοξάτο', 'Δοξάτο': 'Δοξάτο'})
47         elif (municipality == 'Παρανεστίου'):
48             dictionary_municipalities.update({'Παρανεστίου': 'Παρανέστι', 'Παρανέστι': 'Παρανέστι'})

```

Εικόνα 8: Κώδικας που επεξεργάζεται δήμους (1-47)

```

48 elif (municipality == 'Αράμας'):
49     dictionary_municipalities.update({'Αράμας': 'Αράμα', 'Αράμα': 'Αράμα'})
50 elif (municipality == 'Προσοτσάνης'):
51     dictionary_municipalities.update({'Προσοτσάνης': 'Προσοτσάνη', 'Προσοτσάνη': 'Προσοτσάνη'})
52 elif (municipality == 'Κάτω Νευροκοπίου'):
53     dictionary_municipalities.update({'Κάτω Νευροκοπίου': 'Κάτω Νευροκόπι', 'Κάτω Νευροκόπι': 'Κάτω Νευροκόπι'})
54 elif (municipality == 'Νέας Ζίχνης'):
55     dictionary_municipalities.update({'Νέας Ζίχνης': 'Νέα Ζίχνη', 'Νέα Ζίχνη': 'Νέα Ζίχνη'})
56 elif (municipality == 'Αμφίπολης'):
57     dictionary_municipalities.update({'Αμφίπολης': 'Αμφίπολη', 'Αμφίπολη': 'Αμφίπολη'})
58 elif (municipality == 'Βισαλτίας'):
59     dictionary_municipalities.update({'Βισαλτίας': 'Βισαλτία', 'Βισαλτία': 'Βισαλτία'})
60 elif (municipality == 'Εμμανουήλ Παπάς'):
61     dictionary_municipalities.update({'Εμμανουήλ Παπάς': 'Εμμανουήλ Παπάς', 'Εμμανουήλ Παπάς': 'Εμμανουήλ Παπάς'})
62 elif (municipality == 'Σερρών'):
63     dictionary_municipalities.update({'Σερρών': 'Σέρρες', 'Σέρρες': 'Σέρρες'})
64 elif (municipality == 'Ηράκλειας'):
65     dictionary_municipalities.update({'Ηράκλειας': 'Ηράκλεια', 'Ηράκλεια': 'Ηράκλεια'})
66 elif (municipality == 'Σιντικής'):
67     dictionary_municipalities.update({'Σιντικής': 'Σιντική', 'Σιντική': 'Σιντική'})
68 elif (municipality == 'Κιλκίς'):
69     dictionary_municipalities.update({'Κιλκίς': 'Κιλκίς'})
70 elif (municipality == 'Παιονίας'):
71     dictionary_municipalities.update({'Παιονίας': 'Παιονία', 'Παιονία': 'Παιονία'})
72 elif (municipality == 'Χαλκηδόνας'):
73     dictionary_municipalities.update({'Χαλκηδόνας': 'Χαλκηδόνα', 'Χαλκηδόνα': 'Χαλκηδόνα', 'Χαλκηδόνα': 'Χαλκηδόνα'})
74 elif (municipality == 'Δέλτα'):
75     dictionary_municipalities.update({'Δέλτα': 'Δέλτα'})
76 elif (municipality == 'Φραιοκάστρου'):
77     dictionary_municipalities.update({'Φραιοκάστρου': 'Φραιόκαστρο', 'Φραιόκαστρο': 'Φραιόκαστρο'})
78 elif (municipality == 'Παύλου Μελά'):
79     dictionary_municipalities.update({'Παύλου Μελά': 'Πάυλος Μελάς', 'Πάυλος Μελάς': 'Πάυλος Μελάς', 'Πάυλος Μελά': 'Πάυλος
Μελάς'})
80 elif (municipality == 'Αμπελοκήπων-Μενεμένης'):
81     dictionary_municipalities.update({'Αμπελοκήπων-Μενεμένης': 'Αμπελόκηποι-Μενεμένη', 'Αμπελόκηποι-Μενεμένη': 'Αμπελόκηποι-
Μενεμένη', 'Αμπελόκηποι': 'Αμπελόκηποι-Μενεμένη', 'Μενεμένη': 'Αμπελόκηποι-Μενεμένη', 'Αμπελόκηπων': 'Αμπελόκηποι-Μενεμένη',
'Μενεμένης': 'Αμπελόκηποι-Μενεμένη', 'Αμπελόκηπους-Μενεμένη': 'Αμπελόκηποι-Μενεμένη', 'Αμπελόκηποι': 'Αμπελόκηποι-Μενεμένη'})
82 elif (municipality == 'Κορδελιού-Εύσομου'):
83     dictionary_municipalities.update({'Κορδελιού-Εύσομου': 'Κορδελιό-Εύσομος', 'Κορδελιό-Εύσομος': 'Κορδελιό-Εύσομος', 'Κορδελιό-Εύσομος': 'Κορδελιό-Εύσομος',
'Κορδελιό': 'Κορδελιό-Εύσομος', 'Εύσομος': 'Κορδελιό-Εύσομος', 'Κορδελιού': 'Κορδελιό-Εύσομος', 'Εύσομου': 'Κορδελιό-Εύσομος', 'Κορδελιό-Εύσομο': 'Κορδελιό-Εύσομος',
'Εύσομο': 'Κορδελιό-Εύσομος'})
84 elif (municipality == 'Νεάπολης-Συκεών'):
85     dictionary_municipalities.update({'Νεάπολης-Συκεών': 'Νεάπολη-Συκιές', 'Νεάπολη-Συκιές': 'Νεάπολη-Συκιές',
'Νεάπολη': 'Νεάπολη-Συκιές', 'Συκιές': 'Νεάπολη-Συκιές', 'Νεάπολης': 'Νεάπολη-Συκιές', 'Συκεών': 'Νεάπολη-Συκιές'})
86 elif (municipality == 'Θεσσαλονίκης'):
87     dictionary_municipalities.update({'Θεσσαλονίκης': 'Θεσσαλονίκη', 'Θεσσαλονίκη': 'Θεσσαλονίκη'})

```

Εικόνα 9: Κώδικας που επεξεργάζεται δήμους (48-87)

```

88 elif (municipality == 'Κολομαριάς'):
89     dictionary_municipalities.update({'Κολομαριάς': 'Κολομαριά', 'Κολομαριά': 'Κολομαριά'})
90 elif (municipality == 'Θερμαϊκού'):
91     dictionary_municipalities.update({'Θερμαϊκού': 'Θερμαϊκός', 'Θερμαϊκός': 'Θερμαϊκός', 'Θερμαϊκό': 'Θερμαϊκός'})
92 elif (municipality == 'Θέρμης'):
93     dictionary_municipalities.update({'Θέρμης': 'Θέρμη', 'Θέρμη': 'Θέρμη'})
94 elif (municipality == 'Πυλαίας-Χορτιάτης'):
95     dictionary_municipalities.update({'Πυλαίας-Χορτιάτης': 'Πυλαία-Χορτιάτης', 'Πυλαία-Χορτιάτης': 'Πυλαία-Χορτιάτης',
'Πυλαία': 'Πυλαία-Χορτιάτης', 'Χορτιάτης': 'Πυλαία-Χορτιάτης', 'Πυλαίας': 'Πυλαία-Χορτιάτης', 'Χορτιάτη': 'Πυλαία-Χορτιάτης'})
96 elif (municipality == 'Λαγκαδάς'):
97     dictionary_municipalities.update({'Λαγκαδάς': 'Λαγκαδός', 'Λαγκαδός': 'Λαγκαδός'})
98 elif (municipality == 'Βόλβης'):
99     dictionary_municipalities.update({'Βόλβης': 'Βόλβη', 'Βόλβη': 'Βόλβη'})
100 elif (municipality == 'Αριστοτέλη'):
101     dictionary_municipalities.update({'Αριστοτέλη': 'Αριστοτέλης', 'Αριστοτέλης': 'Αριστοτέλης'})
102 elif (municipality == 'Σιθωνίας'):
103     dictionary_municipalities.update({'Σιθωνίας': 'Σιθωνία', 'Σιθωνία': 'Σιθωνία'})
104 elif (municipality == 'Πολύγυρου'):
105     dictionary_municipalities.update({'Πολύγυρου': 'Πολύγυρος', 'Πολύγυρος': 'Πολύγυρος', 'Πολύγυρο': 'Πολύγυρος'})
106 elif (municipality == 'Κασσάνδρας'):
107     dictionary_municipalities.update({'Κασσάνδρας': 'Κασσάνδρα', 'Κασσάνδρα': 'Κασσάνδρα'})
108 elif (municipality == 'Νέας Προποντίδας'):
109     dictionary_municipalities.update({'Νέας Προποντίδας': 'Νέα Προποντία', 'Νέα Προποντία': 'Νέα Προποντία'})
110 elif (municipality == 'Δίου-Ολύμπου'):
111     dictionary_municipalities.update({'Δίου-Ολύμπου': 'Διόν Ολύμπος', 'Διόν Ολύμπος': 'Διόν Ολύμπος'})
112 elif (municipality == 'Κατερίνης'):
113     dictionary_municipalities.update({'Κατερίνης': 'Κατερίνη', 'Κατερίνη': 'Κατερίνη'})
114 elif (municipality == 'Πύδνας-Κολινδρού'):
115     dictionary_municipalities.update({'Πύδνας-Κολινδρού': 'Πύδνα-Κολινδρός', 'Πύδνα-Κολινδρός': 'Πύδνα-Κολινδρός',
'Πύδνα': 'Πύδνα-Κολινδρός', 'Κολινδρός': 'Πύδνα-Κολινδρός', 'Πύδνας': 'Πύδνα-Κολινδρός', 'Κολινδρού': 'Πύδνα-Κολινδρός',
'Πύδνα-Κολινδρό': 'Πύδνα-Κολινδρός'})
116 elif (municipality == 'Αλεξάνδρειας'):
117     dictionary_municipalities.update({'Αλεξάνδρειας': 'Αλεξάνδρεια', 'Αλεξάνδρεια': 'Αλεξάνδρεια'})
118 elif (municipality == 'Βέροιας'):
119     dictionary_municipalities.update({'Βέροιας': 'Βέροια', 'Βέροια': 'Βέροια'})
120 elif (municipality == 'Ηρωικής Πόλεως Νάουσας'):
121     dictionary_municipalities.update({'Ηρωικής Πόλεως Νάουσας': 'Νάουσα', 'Ηρωικής Πόλης Νάουσας': 'Νάουσα', 'Ηρωική Πόλη
Νάουσας': 'Νάουσα', 'Νάουσα': 'Νάουσα', 'Νάουσα': 'Νάουσα'})
122 elif (municipality == 'Πέλλας'):
123     dictionary_municipalities.update({'Πέλλας': 'Πέλλα', 'Πέλλα': 'Πέλλα'})
124 elif (municipality == 'Σκύδρας'):
125     dictionary_municipalities.update({'Σκύδρας': 'Σκύδρα', 'Σκύδρα': 'Σκύδρα'})
126 elif (municipality == 'Εδεσσας'):
127     dictionary_municipalities.update({'Εδεσσας': 'Εδεσσα', 'Εδεσσα': 'Εδεσσα'})
128 elif (municipality == 'Αλμωπίας'):
129     dictionary_municipalities.update({'Αλμωπίας': 'Αλμωπία', 'Αλμωπία': 'Αλμωπία'})

```

Εικόνα 10: Κώδικας που επεξεργάζεται δήμους (88-129)

```

130 elif (municipality == 'Αμυνταίου'):
131     dictionary_municipalities.update({'Αμυνταίου': 'Αμύνταιο', 'Αμύνταιο': 'Αμύνταιο'})
132 elif (municipality == 'Φλώρινας'):
133     dictionary_municipalities.update({'Φλώρινας': 'Φλώρινα', 'Φλώρινα': 'Φλώρινα'})
134 elif (municipality == 'Πρεσπών'):
135     dictionary_municipalities.update({'Πρεσπών': 'Πρέσπες', 'Πρέσπες': 'Πρέσπες'})
136 elif (municipality == 'Καστοριάς'):
137     dictionary_municipalities.update({'Καστοριάς': 'Καστοριά', 'Καστοριά': 'Καστοριά'})
138 elif (municipality == 'Νεστόριου'):
139     dictionary_municipalities.update({'Νεστόριου': 'Νεστόριο', 'Νεστόριο': 'Νεστόριο'})
140 elif (municipality == 'Ορεστίδος'):
141     dictionary_municipalities.update({'Ορεστίδος': 'Ορεστίδα', 'Ορεστίδα': 'Ορεστίδα'})
142 elif (municipality == 'Εορδαίας'):
143     dictionary_municipalities.update({'Εορδαίας': 'Εορδαία', 'Εορδαία': 'Εορδαία'})
144 elif (municipality == 'Κοζάνης'):
145     dictionary_municipalities.update({'Κοζάνης': 'Κοζάνη', 'Κοζάνη': 'Κοζάνη'})
146 elif (municipality == 'Βοΐου'):
147     dictionary_municipalities.update({'Βοΐου': 'Βόιο', 'Βόιο': 'Βόιο'})
148 elif (municipality == 'Σερβίων-Βελεντού'):
149     dictionary_municipalities.update({'Σερβίων-Βελεντού': 'Σέρβια-Βελεντός', 'Σέρβια-Βελεντός': 'Σέρβια-Βελεντός',
150     'Σέρβια': 'Σέρβια-Βελεντός', 'Βελεντός': 'Σέρβια-Βελεντός', 'Σερβίων': 'Σέρβια-Βελεντός', 'Βελεντού': 'Σέρβια-Βελεντός', 'Σέρβια-Βελεντός': 'Σέρβια-Βελεντός'})
151 elif (municipality == 'Δεσκάτης'):
152     dictionary_municipalities.update({'Δεσκάτης': 'Δεσκάτη', 'Δεσκάτη': 'Δεσκάτη'})
153 elif (municipality == 'Τρεβενών'):
154     dictionary_municipalities.update({'Τρεβενών': 'Τρεβενά', 'Τρεβενά': 'Τρεβενά'})
155 elif (municipality == 'Κόνιτσας'):
156     dictionary_municipalities.update({'Κόνιτσας': 'Κόνιτσα', 'Κόνιτσα': 'Κόνιτσα'})
157 elif (municipality == 'Παγωνίου'):
158     dictionary_municipalities.update({'Παγωνίου': 'Παγώνι', 'Παγώνι': 'Παγώνι'})
159 elif (municipality == 'Ζαγορίου'):
160     dictionary_municipalities.update({'Ζαγορίου': 'Ζαγόρι', 'Ζαγόρι': 'Ζαγόρι'})
161 elif (municipality == 'Ζίτσας'):
162     dictionary_municipalities.update({'Ζίτσας': 'Ζίτσα', 'Ζίτσα': 'Ζίτσα'})
163 elif (municipality == 'Ιωαννινών'):
164     dictionary_municipalities.update({'Ιωαννινών': 'Ιωάννινα', 'Ιωάννιναν': 'Ιωάννινα', 'Ιωάννινα': 'Ιωάννινα',
165     'Τιάννενα': 'Ιωάννινα'})
166 elif (municipality == 'Μετσόβου'):
167     dictionary_municipalities.update({'Μετσόβου': 'Μέτσοβο', 'Μέτσοβο': 'Μέτσοβο'})
168 elif (municipality == 'Βορείων Τζουμέρκων'):
169     dictionary_municipalities.update({'Βορείων Τζουμέρκων': 'Βόρεια Τζουμέρκα', 'Βόρεια Τζουμέρκα': 'Βόρεια Τζουμέρκα'})
170 elif (municipality == 'Δωδώνης'):
171     dictionary_municipalities.update({'Δωδώνης': 'Δωδώνη', 'Δωδώνη': 'Δωδώνη'})
172 elif (municipality == 'Σουλίου'):
173     dictionary_municipalities.update({'Σουλίου': 'Σούλι', 'Σούλι': 'Σούλι'})

```

Εικόνα 11: Κώδικας που επεξεργάζεται δήμους (130-171)

```

172 elif (municipality == 'Φιλιατών'):
173     dictionary_municipalities.update({'Φιλιατών': 'Φιλιάτες', 'Φιλιάτες': 'Φιλιάτες'})
174 elif (municipality == 'Ηγουμενίτσας'):
175     dictionary_municipalities.update({'Ηγουμενίτσας': 'Ηγουμενίτσα', 'Ηγουμενίτσα': 'Ηγουμενίτσα'})
176 elif (municipality == 'Πάργας'):
177     dictionary_municipalities.update({'Πάργας': 'Πάργα', 'Πάργα': 'Πάργα'})
178 elif (municipality == 'Πρέβεζας'):
179     dictionary_municipalities.update({'Πρέβεζας': 'Πρέβεζα', 'Πρέβεζα': 'Πρέβεζα'})
180 elif (municipality == 'Ζηροῦ'):
181     dictionary_municipalities.update({'Ζηροῦ': 'Ζήρος', 'Ζήρος': 'Ζήρος', 'Ζήρο': 'Ζήρος'})
182 elif (municipality == 'Άρτας'):
183     dictionary_municipalities.update({'Άρτας': 'Άρτα', 'Άρτα': 'Άρτα'})
184 elif (municipality == 'Νικολάου Εκουφά'):
185     dictionary_municipalities.update({'Νικολάου Εκουφά': 'Νικόλαος Εκουφάς', 'Νικόλαος Εκουφάς': 'Νικόλαος Εκουφάς', 'Νικόλαο
186     Εκουφά': 'Νικόλαος Εκουφάς'})
187 elif (municipality == 'Γεωργίου Καραϊσκάκη'):
188     dictionary_municipalities.update({'Γεωργίου Καραϊσκάκη': 'Γεώργιος Καραϊσκάκης', 'Γεώργιος Καραϊσκάκης': 'Γεώργιος
189     Καραϊσκάκης', 'Γεώργιο Καραϊσκάκη': 'Γεώργιος Καραϊσκάκης'})
190 elif (municipality == 'Κεντρικών Τζουμέρκων'):
191     dictionary_municipalities.update({'Κεντρικών Τζουμέρκων': 'Κεντρικά Τζουμέρκα', 'Κεντρικά Τζουμέρκα': 'Κεντρικά Τζουμέρκα'})
192 elif (municipality == 'Πύλης'):
193     dictionary_municipalities.update({'Πύλης': 'Πύλη', 'Πύλη': 'Πύλη'})
194 elif (municipality == 'Μετεώρων'):
195     dictionary_municipalities.update({'Μετεώρων': 'Μετέωρα', 'Μετέωρα': 'Μετέωρα'})
196 elif (municipality == 'Τρικκίων'):
197     dictionary_municipalities.update({'Τρικκίων': 'Τρίκοια', 'Τρίκοια': 'Τρίκοια'})
198 elif (municipality == 'Φαρκαδόνος'):
199     dictionary_municipalities.update({'Φαρκαδόνος': 'Φαρκαδόνα', 'Φαρκαδόνας': 'Φαρκαδόνα', 'Φαρκαδόνα': 'Φαρκαδόνα'})
200 elif (municipality == 'Πολυαμά'):
201     dictionary_municipalities.update({'Πολυαμά': 'Πολυμάς', 'Πολυμάς': 'Πολυμάς'})
202 elif (municipality == 'Μουζακίου'):
203     dictionary_municipalities.update({'Μουζακίου': 'Μουζάκι', 'Μουζάκι': 'Μουζάκι'})
204 elif (municipality == 'Σοφάδων'):
205     dictionary_municipalities.update({'Σοφάδων': 'Σοφάδες', 'Σοφάδες': 'Σοφάδες'})
206 elif (municipality == 'Αργιθέας'):
207     dictionary_municipalities.update({'Αργιθέας': 'Αργιθέα', 'Αργιθέα': 'Αργιθέα'})
208 elif (municipality == 'Λίμνης Πλαστήρα'):
209     dictionary_municipalities.update({'Λίμνης Πλαστήρα': 'Λίμνη Πλαστήρα', 'Λίμνη Πλαστήρα': 'Λίμνη Πλαστήρα'})
210 elif (municipality == 'Καρδίτσας'):
211     dictionary_municipalities.update({'Καρδίτσας': 'Καρδίτσα', 'Καρδίτσα': 'Καρδίτσα'})
212 elif (municipality == 'Ελασσόνας'):
213     dictionary_municipalities.update({'Ελασσόνας': 'Ελασσόνα', 'Ελασσόνα': 'Ελασσόνα'})
214 elif (municipality == 'Τυρνάβου'):
215     dictionary_municipalities.update({'Τυρνάβου': 'Τύρναβος', 'Τύρναβος': 'Τύρναβος', 'Τύρναβο': 'Τύρναβος'})

```

Εικόνα 12: Κώδικας που επεξεργάζεται δήμους (172-213)

Διαχείριση και Οπτικοποίηση Μεγάλων Δεδομένων για τα Κρούσματα COVID-19 στην Ελλάδα

```
214         elif (municipality == 'Τεμπών'):  
215             dictionary_municipalities.update({'Τεμπών': 'Τέμπη', 'Τέμπη': 'Τέμπη'})  
216         elif (municipality == 'Λάρισας'):  
217             dictionary_municipalities.update({'Λάρισας': 'Λάρισα', 'Λάρισα': 'Λάρισα'})  
218         elif (municipality == 'Αγιάς'):  
219             dictionary_municipalities.update({'Αγιάς': 'Αγιά', 'Αγιά': 'Αγιά'})  
220         elif (municipality == 'Κιλελέρ'):  
221             dictionary_municipalities.update({'Κιλελέρ': 'Κιλελέρ'})  
222         elif (municipality == 'Φαρσάλων'):  
223             dictionary_municipalities.update({'Φαρσάλων': 'Φάρσαλα', 'Φάρσαλα': 'Φάρσαλα'})  
224         elif (municipality == 'Ερήγα Φεραίου'):  
225             dictionary_municipalities.update({'Ερήγα Φεραίου': 'Ερήγας Φεραίος', 'Ερήγας Φεραίος': 'Ερήγας Φεραίος', 'Ερήγα Φεραίο': 'Ερήγας  
Φεραίος'})  
226         elif (municipality == 'Άλμυρού'):  
227             dictionary_municipalities.update({'Άλμυρού': 'Άλμυρός', 'Άλμυρός': 'Άλμυρός', 'Άλμυρό': 'Άλμυρός'})  
228         elif (municipality == 'Βόλου'):  
229             dictionary_municipalities.update({'Βόλου': 'Βόλος', 'Βόλος': 'Βόλος', 'Βόλο': 'Βόλος'})  
230         elif (municipality == 'Ζαγοράς-Μουρεσίου'):  
231             dictionary_municipalities.update({'Ζαγοράς-Μουρεσίου': 'Ζαγορά-Μούρεσι', 'Ζαγορά-Μούρεσι': 'Ζαγορά-Μούρεσι', 'Ζαγορά-Μούρεσι': 'Ζαγορά-Μούρεσι',  
'Ζαγορά': 'Ζαγορά-Μούρεσι', 'Μούρεσι': 'Ζαγορά-Μούρεσι', 'Ζαγοράς': 'Ζαγορά-Μούρεσι', 'Μουρεσίου': 'Ζαγορά-Μούρεσι'})  
232         elif (municipality == 'Νοτίου Πηλίου'):  
233             dictionary_municipalities.update({'Νοτίου Πηλίου': 'Νότιο Πήλιο', 'Νότιο Πήλιο': 'Νότιο Πήλιο'})  
234         elif (municipality == 'Σκιάθου'):  
235             dictionary_municipalities.update({'Σκιάθου': 'Σκιάθος', 'Σκιάθος': 'Σκιάθος', 'Σκιάθο': 'Σκιάθος'})  
236         elif (municipality == 'Σκοπέλου'):  
237             dictionary_municipalities.update({'Σκοπέλου': 'Σκόπελος', 'Σκόπελος': 'Σκόπελος', 'Σκόπελο': 'Σκόπελος'})  
238         elif (municipality == 'Άλωναίου'):  
239             dictionary_municipalities.update({'Άλωναίου': 'Αλώνησος', 'Αλώνησος': 'Αλώνησος', 'Αλώνησο': 'Αλώνησος'})  
240         elif (municipality == 'Σκύρου'):  
241             dictionary_municipalities.update({'Σκύρου': 'Σκύρος', 'Σκύρος': 'Σκύρος', 'Σκύρο': 'Σκύρος'})  
242         elif (municipality == 'Ιστιαίας - Αιδηψού'):  
243             dictionary_municipalities.update({'Ιστιαίας - Αιδηψού': 'Ιστιαία - Αιδηψός', 'Ιστιαία - Αιδηψός': 'Ιστιαία - Αιδηψός', 'Ιστιαία - Αιδηψό': 'Ιστιαία - Αιδηψός',  
'Ιστιαία': 'Ιστιαία - Αιδηψός', 'Αιδηψός': 'Ιστιαία - Αιδηψός', 'Ιστιαίας': 'Ιστιαία - Αιδηψός', 'Αιδηψού': 'Ιστιαία - Αιδηψός', 'Ιστιαία - Αιδηψό': 'Ιστιαία - Αιδηψός'})  
244         elif (municipality == 'Μαντουδίου-Λίμνης-Αγίας Άννας'):  
245             dictionary_municipalities.update({'Μαντουδίου-Λίμνης-Αγίας Άννας': 'Μαντούδι-Λίμνη-Αγία Άννα', 'Μαντούδι-Λίμνη-Αγία Άννα': 'Μαντούδι-Λίμνη-Αγία Άννα',  
'Μαντούδι-Λίμνη-Αγία Άννα', 'Μαντούδι': 'Μαντούδι-Λίμνη-Αγία Άννα', 'Λίμνη': 'Μαντούδι-Λίμνη-Αγία Άννα', 'Αγία Άννα': 'Μαντούδι-Λίμνη-Αγία Άννα', 'Μαντουδίου': 'Μαντούδι-Λίμνη-Αγία Άννα',  
'Λίμνης': 'Μαντούδι-Λίμνη-Αγία Άννα', 'Αγίας Άννας': 'Μαντούδι-Λίμνη-Αγία Άννα'})  
246         elif (municipality == 'Διρφύων-Μεσσαπίων'):  
247             dictionary_municipalities.update({'Διρφύων-Μεσσαπίων': 'Δίρφυες-Μεσσαπίες', 'Δίρφυες-Μεσσαπίες': 'Δίρφυες-Μεσσαπίες', 'Δίρφυες-Μεσσαπίες': 'Δίρφυες-Μεσσαπίες',  
'Δίρφυες': 'Δίρφυες-Μεσσαπίες', 'Μεσσαπίες': 'Δίρφυες-Μεσσαπίες', 'Διρφύων': 'Δίρφυες-Μεσσαπίες', 'Μεσσαπίων': 'Δίρφυες-Μεσσαπίες'})  
248         elif (municipality == 'Χαλκιδέων'):  
249             dictionary_municipalities.update({'Χαλκιδέων': 'Χαλκίδα', 'Χαλκίδα': 'Χαλκίδα'})  
250         elif (municipality == 'Ερέτριας'):  
251             dictionary_municipalities.update({'Ερέτριας': 'Ερέτρια', 'Ερέτρια': 'Ερέτρια'})
```

Εικόνα 13: Κώδικας που επεξεργάζεται δήμους (214-251)

```
252         elif (municipality == 'Κύμης-Αλιβερίου'):  
253             dictionary_municipalities.update({'Κύμης-Αλιβερίου': 'Κύμη-Αλιβέρι', 'Κύμη-Αλιβέρι': 'Κύμη-Αλιβέρι', 'Κύμη': 'Κύμη-Αλιβέρι',  
'Αλιβέρι': 'Κύμη-Αλιβέρι', 'Κύμης': 'Κύμη-Αλιβέρι', 'Αλιβερίου': 'Κύμη-Αλιβέρι'})  
254         elif (municipality == 'Καρύστου'):  
255             dictionary_municipalities.update({'Καρύστου': 'Κάρυστος', 'Κάρυστος': 'Κάρυστος', 'Κάρυστο': 'Κάρυστος'})  
256         elif (municipality == 'Τανάγρας'):  
257             dictionary_municipalities.update({'Τανάγρας': 'Τανάγρα', 'Τανάγρα': 'Τανάγρα'})  
258         elif (municipality == 'Θηβαίων'):  
259             dictionary_municipalities.update({'Θηβαίων': 'Θήβα', 'Θήβα': 'Θήβα'})  
260         elif (municipality == 'Αλιάρτου'):  
261             dictionary_municipalities.update({'Αλιάρτου': 'Αλιάρτος', 'Αλιάρτος': 'Αλιάρτος', 'Αλιάρτο': 'Αλιάρτος'})  
262         elif (municipality == 'Ορχομενού'):  
263             dictionary_municipalities.update({'Ορχομενού': 'Ορχομενός', 'Ορχομενός': 'Ορχομενός', 'Ορχομενό': 'Ορχομενός'})  
264         elif (municipality == 'Λεβαδέων'):  
265             dictionary_municipalities.update({'Λεβαδέων': 'Λειβαδιά', 'Λειβαδιά': 'Λειβαδιά'})  
266         elif (municipality == 'Διστόμου-Αράχοβα-Αντίκυρας'):  
267             dictionary_municipalities.update({'Διστόμου-Αράχοβα-Αντίκυρας': 'Δίστομο-Αράχοβα-Αντίκυρα', 'Δίστομο-Αράχοβα-Αντίκυρα': 'Δίστομο-Αράχοβα-Αντίκυρα',  
'Αντίκυρα': 'Δίστομο-Αράχοβα-Αντίκυρα', 'Δίστομο': 'Δίστομο-Αράχοβα-Αντίκυρα', 'Αράχοβα': 'Δίστομο-Αράχοβα-Αντίκυρα', 'Αντίκυρας': 'Δίστομο-Αράχοβα-Αντίκυρα',  
'Αντίκυρα': 'Δίστομο-Αράχοβα-Αντίκυρα'})  
268         elif (municipality == 'Λοκρών'):  
269             dictionary_municipalities.update({'Λοκρών': 'Λοκρές', 'Λοκρές': 'Λοκρές'})  
270         elif (municipality == 'Αμφίκλεια-Ελάτεια'):  
271             dictionary_municipalities.update({'Αμφίκλεια-Ελάτεια': 'Αμφίκλεια-Ελάτεια', 'Αμφίκλεια-Ελάτεια': 'Αμφίκλεια-Ελάτεια', 'Αμφίκλεια-Ελάτεια': 'Αμφίκλεια-Ελάτεια',  
'Αμφίκλεια': 'Αμφίκλεια-Ελάτεια', 'Ελάτεια': 'Αμφίκλεια-Ελάτεια', 'Αμφίκλειας': 'Αμφίκλεια-Ελάτεια', 'Ελάτειας': 'Αμφίκλεια-Ελάτεια'})  
272         elif (municipality == 'Μόλου-Αγίου Κωνσταντίνου'):  
273             dictionary_municipalities.update({'Μόλου-Αγίου Κωνσταντίνου': 'Μόλος-Άγιος Κωνσταντίνος', 'Μόλος-Άγιος Κωνσταντίνος': 'Μόλος-Άγιος Κωνσταντίνος',  
'Μόλου': 'Μόλος-Άγιος Κωνσταντίνος', 'Άγιος Κωνσταντίνος': 'Μόλος-Άγιος Κωνσταντίνος', 'Μόλο-Άγιος Κωνσταντίνος': 'Μόλος-Άγιος Κωνσταντίνος',  
'Μόλο': 'Μόλος-Άγιος Κωνσταντίνος', 'Άγιος Κωνσταντίνος': 'Μόλος-Άγιος Κωνσταντίνος'})  
274         elif (municipality == 'Λαμιέων'):  
275             dictionary_municipalities.update({'Λαμιέων': 'Λαμία', 'Λαμία': 'Λαμία'})  
276         elif (municipality == 'Στυλίδας'):  
277             dictionary_municipalities.update({'Στυλίδας': 'Στυλίδα', 'Στυλίδα': 'Στυλίδα'})  
278         elif (municipality == 'Δομοκού'):  
279             dictionary_municipalities.update({'Δομοκού': 'Δομοκός', 'Δομοκός': 'Δομοκός', 'Δομοκό': 'Δομοκός'})  
280         elif (municipality == 'Δελφών'):  
281             dictionary_municipalities.update({'Δελφών': 'Δελφοί', 'Δελφοί': 'Δελφοί', 'Δελφούς': 'Δελφοί'})  
282         elif (municipality == 'Δωρίδας'):  
283             dictionary_municipalities.update({'Δωρίδας': 'Δωρίδα', 'Δωρίδας': 'Δωρίδα', 'Δωρίδα': 'Δωρίδα'})  
284         elif (municipality == 'Καρπενησίου'):  
285             dictionary_municipalities.update({'Καρπενησίου': 'Καρπενήσι', 'Καρπενήσι': 'Καρπενήσι'})  
286         elif (municipality == 'Αγιάρας'):  
287             dictionary_municipalities.update({'Αγιάρας': 'Αγίαρα', 'Αγίαρα': 'Αγίαρα'})
```

Εικόνα 14: Κώδικας που επεξεργάζεται δήμους (252-287)

```

288 elif (municipality == 'Αμφιλοχίας'):
289     dictionary_municipalities.update({'Αμφιλοχίας': 'Αμφιλοχία', 'Αμφιλοχία': 'Αμφιλοχία'})
290 elif (municipality == 'Ακτίου-Βόνιτσας'):
291     dictionary_municipalities.update({'Ακτίου-Βόνιτσας': 'Ακτίο-Βόνιτσα', 'Ακτίο-Βόνιτσα': 'Ακτίο-Βόνιτσα', 'Ακτίο-Βόνιτσα': 'Ακτίο-Βόνιτσα', 'Ακτίο-Βόνιτσα': 'Ακτίο-Βόνιτσα'})
292 elif (municipality == 'Εηρομέρου'):
293     dictionary_municipalities.update({'Εηρομέρου': 'Εηρόμερο', 'Εηρόμερο': 'Εηρόμερο'})
294 elif (municipality == 'Ιεράς Πόλεως Μεσολογγίου'):
295     dictionary_municipalities.update({'Ιεράς Πόλεως Μεσολογγίου': 'Μεσολόγγι', 'Ιερά Πόλη Μεσολογγίου': 'Μεσολόγγι', 'Ιεράς
Πόλης Μεσολογγίου': 'Μεσολόγγι', 'Μεσολογγίου': 'Μεσολόγγι', 'Μεσολόγγι': 'Μεσολόγγι'})
296 elif (municipality == 'Αγρινίου'):
297     dictionary_municipalities.update({'Αγρινίου': 'Αγρίνιο', 'Αγρίνιο': 'Αγρίνιο'})
298 elif (municipality == 'Θέρμου'):
299     dictionary_municipalities.update({'Θέρμου': 'Θέρμο', 'Θέρμο': 'Θέρμο'})
300 elif (municipality == 'Νουποκτίας'):
301     dictionary_municipalities.update({'Νουποκτίας': 'Νούπακτος', 'Νουπακτία': 'Νούπακτος', 'Νουπάκτου': 'Νούπακτος',
'Νούπακτος': 'Νούπακτος', 'Νούπακτο': 'Νούπακτος'})
302 elif (municipality == 'Πατρέων'):
303     dictionary_municipalities.update({'Πατρέων': 'Πάτρα', 'Πάτρα': 'Πάτρα'})
304 elif (municipality == 'Αιγιάλειας'):
305     dictionary_municipalities.update({'Αιγιάλειας': 'Αιγιάλεια', 'Αιγιάλεια': 'Αιγιάλεια'})
306 elif (municipality == 'Καλαβρύτων'):
307     dictionary_municipalities.update({'Καλαβρύτων': 'Καλάβρυτα', 'Καλάβρυτα': 'Καλάβρυτα'})
308 elif (municipality == 'Ερυμάνθου'):
309     dictionary_municipalities.update({'Ερυμάνθου': 'Ερύμανθος', 'Ερύμανθος': 'Ερύμανθος', 'Ερύμανθο': 'Ερύμανθος'})
310 elif (municipality == 'Δυτικής Αχαΐας'):
311     dictionary_municipalities.update({'Δυτικής Αχαΐας': 'Δυτική Αχαΐα', 'Δυτική Αχαΐα': 'Δυτική Αχαΐα'})
312 elif (municipality == 'Ανδραβίδας - Κυλλήνης'):
313     dictionary_municipalities.update({'Ανδραβίδας - Κυλλήνης': 'Ανδραβίδα - Κυλλήνη', 'Ανδραβίδα - Κυλλήνη': 'Ανδραβίδα - Κυλλήνη', 'Ανδραβίδα - Κυλλήνη': 'Ανδραβίδα -
Κυλλήνη', 'Ανδραβίδα': 'Ανδραβίδα - Κυλλήνη'})
314 elif (municipality == 'Πηνειού'):
315     dictionary_municipalities.update({'Πηνειού': 'Πηνειός', 'Πηνειός': 'Πηνειός', 'Πηνειό': 'Πηνειός'})
316 elif (municipality == 'Ηλιδας'):
317     dictionary_municipalities.update({'Ηλιδας': 'Ηλιδα', 'Ηλιδα': 'Ηλιδα'})
318 elif (municipality == 'Πύργου'):
319     dictionary_municipalities.update({'Πύργου': 'Πύργος', 'Πύργος': 'Πύργος', 'Πύργο': 'Πύργος'})
320 elif (municipality == 'Αρχαίας Ολυμπίας'):
321     dictionary_municipalities.update({'Αρχαίας Ολυμπίας': 'Αρχαία Ολυμπία', 'Αρχαία Ολυμπία': 'Αρχαία Ολυμπία'})
322 elif (municipality == 'Ανδρίτσαινας - Κρεστένων'):
323     dictionary_municipalities.update({'Ανδρίτσαινας - Κρεστένων': 'Ανδρίτσαινα - Κρέστενα', 'Ανδρίτσαινα -
Κρέστενα': 'Ανδρίτσαινα - Κρέστενα', 'Ανδρίτσαινα': 'Ανδρίτσαινα - Κρέστενα', 'Κρέστενα': 'Ανδρίτσαινα - Κρέστενα',
'Ανδρίτσαινας': 'Ανδρίτσαινα - Κρέστενα', 'Κρεστένων': 'Ανδρίτσαινα - Κρέστενα'})
324 elif (municipality == 'Ζαχάρως'):
325     dictionary_municipalities.update({'Ζαχάρως': 'Ζαχάρω', 'Ζαχάρω': 'Ζαχάρω'})
326 elif (municipality == 'Κέρκυρας'):
327     dictionary_municipalities.update({'Κέρκυρας': 'Κέρκυρα', 'Κέρκυρα': 'Κέρκυρα'})

```

Εικόνα 15: Κώδικας που επεξεργάζεται δήμους (288-327)

```

328 elif (municipality == 'Παξών'):
329     dictionary_municipalities.update({'Παξών': 'Παξοί', 'Παξοί': 'Παξοί', 'Παξούς': 'Παξοί'})
330 elif (municipality == 'Λευκάδας'):
331     dictionary_municipalities.update({'Λευκάδας': 'Λευκάδα', 'Λευκάδα': 'Λευκάδα'})
332 elif (municipality == 'Μεγανησίου'):
333     dictionary_municipalities.update({'Μεγανησίου': 'Μεγανήσι', 'Μεγανήσι': 'Μεγανήσι'})
334 elif (municipality == 'Ιθάκης'):
335     dictionary_municipalities.update({'Ιθάκης': 'Ιθάκη', 'Ιθάκη': 'Ιθάκη'})
336 elif (municipality == 'Κεφαλονιάς'):
337     dictionary_municipalities.update({'Κεφαλονιάς': 'Κεφαλλονιά', 'Κεφαλλονιά': 'Κεφαλλονιά', 'Κεφαλληνίας': 'Κεφαλλονιά',
'Κεφαλληνία': 'Κεφαλλονιά'})
338 elif (municipality == 'Ζακύνθου'):
339     dictionary_municipalities.update({'Ζακύνθου': 'Ζάκυνθος', 'Ζάκυνθος': 'Ζάκυνθος', 'Ζάκυνθο': 'Ζάκυνθος'})
340 elif (municipality == 'Κυθήρων'):
341     dictionary_municipalities.update({'Κυθήρων': 'Κύθηρα', 'Κύθηρα': 'Κύθηρα'})
342 elif (municipality == 'Σπετσών'):
343     dictionary_municipalities.update({'Σπετσών': 'Σπέτσες', 'Σπέτσες': 'Σπέτσες'})
344 elif (municipality == 'Υδρας'):
345     dictionary_municipalities.update({'Υδρας': 'Υδρα', 'Υδρα': 'Υδρα'})
346 elif (municipality == 'Τροιζηνίας'):
347     dictionary_municipalities.update({'Τροιζηνίας': 'Τροιζηνία', 'Τροιζηνία': 'Τροιζηνία'})
348 elif (municipality == 'Πόρου'):
349     dictionary_municipalities.update({'Πόρου': 'Πόρος', 'Πόρος': 'Πόρος', 'Πόρο': 'Πόρος'})
350 elif (municipality == 'Αγκιστρίου'):
351     dictionary_municipalities.update({'Αγκιστρίου': 'Αγκίστρι', 'Αγκίστρι': 'Αγκίστρι'})
352 elif (municipality == 'Αίγινας'):
353     dictionary_municipalities.update({'Αίγινας': 'Αίγινα', 'Αίγινα': 'Αίγινα'})
354 elif (municipality == 'Σολομίνας'):
355     dictionary_municipalities.update({'Σολομίνας': 'Σολομίνα', 'Σολομίνα': 'Σολομίνα'})
356 elif (municipality == 'Περόματος'):
357     dictionary_municipalities.update({'Περόματος': 'Πέρομα', 'Πέρομα': 'Πέρομα'})
358 elif (municipality == 'Κερατσίνου - Δραπετσώνας'):
359     dictionary_municipalities.update({'Κερατσίνου - Δραπετσώνας': 'Κερατσίνι - Δραπετσώνα', 'Κερατσίνι -
Δραπετσώνα': 'Κερατσίνι - Δραπετσώνα', 'Κερατσίνι': 'Κερατσίνι - Δραπετσώνα', 'Δραπετσώνα': 'Κερατσίνι - Δραπετσώνα',
'Κερατσίνου': 'Κερατσίνι - Δραπετσώνα', 'Κερατσώνας': 'Κερατσίνι - Δραπετσώνα'})
360 elif (municipality == 'Πειραιά'):
361     dictionary_municipalities.update({'Πειραιά': 'Πειραιάς', 'Πειραιάς': 'Πειραιάς', 'Πειραιός': 'Πειραιάς'})
362 elif (municipality == 'Νίκαιας - Αγίου Ιωάννη Ρέντης'):
363     dictionary_municipalities.update({'Νίκαιας - Αγίου Ιωάννη Ρέντη': 'Νίκαια - Άγιος Ιωάννης Ρέντης', 'Νίκαια - Άγιος Ιωάννης
Ρέντης': 'Νίκαια - Άγιος Ιωάννης Ρέντης', 'Άγιος Ιωάννης Ρέντης': 'Νίκαια - Άγιος Ιωάννης Ρέντης', 'Νίκαια - Άγιος Ιωάννης
Ρέντης': 'Νίκαια - Άγιος Ιωάννης Ρέντης', 'Άγιος Ιωάννη Ρέντη': 'Νίκαια - Άγιος Ιωάννης Ρέντης', 'Νίκαια - Άγιος Ιωάννης
Ρέντης'})
364 elif (municipality == 'Κορυδαλλού'):
365     dictionary_municipalities.update({'Κορυδαλλού': 'Κορυδαλλός', 'Κορυδαλλός': 'Κορυδαλλός', 'Κορυδαλλό': 'Κορυδαλλός'})

```

Εικόνα 16: Κώδικας που επεξεργάζεται δήμους (328-365)

Διαχείριση και Οπτικοποίηση Μεγάλων Δεδομένων για τα Κρούσματα COVID-19 στην Ελλάδα

```

366 elif (municipality == 'Αγίας Βαρβάρας'):
367     dictionary_municipalities.update({'Αγίας Βαρβάρας': 'Αγία Βαρβάρα', 'Αγία Βαρβάρα': 'Αγία Βαρβάρα'})
368 elif (municipality == 'Χαϊδαρίου'):
369     dictionary_municipalities.update({'Χαϊδαρίου': 'Χαϊδάρι', 'Χαϊδάρι': 'Χαϊδάρι'})
370 elif (municipality == 'Αιγάλεω'):
371     dictionary_municipalities.update({'Αιγάλεω': 'Αιγάλεω'})
372 elif (municipality == 'Περιστερίου'):
373     dictionary_municipalities.update({'Περιστερίου': 'Περιστέρι', 'Περιστέρι': 'Περιστέρι'})
374 elif (municipality == 'Πετρούπολης'):
375     dictionary_municipalities.update({'Πετρούπολης': 'Πετρούπολη', 'Πετρούπολη': 'Πετρούπολη'})
376 elif (municipality == 'Ιλίου'):
377     dictionary_municipalities.update({'Ιλίου': 'Ιλιον', 'Ιλιον': 'Ιλιον'})
378 elif (municipality == 'Αγίων Αναργύρων - Καματερού'):
379     dictionary_municipalities.update({'Αγίων Αναργύρων - Καματερού': 'Άγιοι Ανάργυροι - Καματερό', 'Άγιοι Ανάργυροι -
Καματερό': 'Άγιοι Ανάργυροι - Καματερό', 'Άγιοι Ανάργυροι': 'Άγιοι Ανάργυροι - Καματερό', 'Καματερό': 'Άγιοι Ανάργυροι - Καματερό',
'Άγιον Αναργύρων': 'Άγιοι Ανάργυροι - Καματερό', 'Καματερού': 'Άγιοι Ανάργυροι - Καματερό', 'Άγιους Αναργύρους - Καματερό': 'Άγιοι
Ανάργυροι - Καματερό', 'Άγιους Αναργύρους': 'Άγιοι Ανάργυροι - Καματερό'})
380 elif (municipality == 'Φιλαδέλφειας - Χολκιδόνας'):
381     dictionary_municipalities.update({'Φιλαδέλφειας - Χολκιδόνας': 'Φιλαδέλφεια - Χολκιδόνα', 'Φιλαδέλφεια -
Χολκιδόνα': 'Φιλαδέλφεια - Χολκιδόνα', 'Φιλαδέλφεια': 'Φιλαδέλφεια - Χολκιδόνα', 'Φιλαδέλφειας': 'Φιλαδέλφεια - Χολκιδόνα'})
382 elif (municipality == 'Γαλατσίτου'):
383     dictionary_municipalities.update({'Γαλατσίτου': 'Γαλάτσι', 'Γαλάτσι': 'Γαλάτσι'})
384 elif (municipality == 'Αθηναίων'):
385     dictionary_municipalities.update({'Αθηναίων': 'Αθήνα', 'Αθήνα': 'Αθήνα'})
386 elif (municipality == 'Δάφνης - Ψηττού'):
387     dictionary_municipalities.update({'Δάφνης - Ψηττού': 'Δάφνη - Ψηττός', 'Δάφνη - Ψηττός': 'Δάφνη - Ψηττός',
'Δάφνη': 'Δάφνη - Ψηττός', 'Ψηττός': 'Δάφνη - Ψηττός', 'Δάφνης': 'Δάφνη - Ψηττός', 'Ψηττού': 'Δάφνη - Ψηττός', 'Δάφνη -
Ψηττός': 'Δάφνη - Ψηττός'})
388 elif (municipality == 'Ηλιούπολης'):
389     dictionary_municipalities.update({'Ηλιούπολης': 'Ηλιούπολη', 'Ηλιούπολη': 'Ηλιούπολη'})
390 elif (municipality == 'Βύρωνος'):
391     dictionary_municipalities.update({'Βύρωνος': 'Βύρωνα', 'Βύρωνα': 'Βύρωνα', 'Βύρωνα': 'Βύρωνα'})
392 elif (municipality == 'Καισαριανής'):
393     dictionary_municipalities.update({'Καισαριανής': 'Καισαριανή', 'Καισαριανή': 'Καισαριανή'})
394 elif (municipality == 'Ζωγράφου'):
395     dictionary_municipalities.update({'Ζωγράφου': 'Ζωγράφου'})
396 elif (municipality == 'Παπάγου-Χολαργού'):
397     dictionary_municipalities.update({'Παπάγου-Χολαργού': 'Παπάγου-Χολαργός', 'Παπάγου-Χολαργός': 'Παπάγου-Χολαργός',
'Παπάγου': 'Παπάγου-Χολαργός', 'Χολαργός': 'Παπάγου-Χολαργός', 'Χολαργού': 'Παπάγου-Χολαργός', 'Παπάγου-Χολαργός': 'Παπάγου-Χολαργός',
'Χολαργός': 'Παπάγου-Χολαργός'})
398 elif (municipality == 'Αγίας Παρασκευής'):
399     dictionary_municipalities.update({'Αγίας Παρασκευής': 'Αγία Παρασκευή', 'Αγία Παρασκευή': 'Αγία Παρασκευή'})
400 elif (municipality == 'Χαλανδρίου'):
401     dictionary_municipalities.update({'Χαλανδρίου': 'Χαλάνδρι', 'Χαλάνδρι': 'Χαλάνδρι'})
402 elif (municipality == 'Βριλησίων'):
403     dictionary_municipalities.update({'Βριλησίων': 'Βριλήσσια', 'Βριλήσσια': 'Βριλήσσια'})

```

Εικόνα 17: Κώδικας που επεξεργάζεται δήμους (366-403)

```

404 elif (municipality == 'Αμαρουσίου'):
405     dictionary_municipalities.update({'Αμαρουσίου': 'Μαρούσι', 'Μαρούσι': 'Μαρούσι'})
406 elif (municipality == 'Φιλοθέης-Ψυχικού'):
407     dictionary_municipalities.update({'Φιλοθέης-Ψυχικού': 'Φιλοθέη-Ψυχικό', 'Φιλοθέη-Ψυχικό': 'Φιλοθέη-Ψυχικό',
'Φιλοθέη': 'Φιλοθέη-Ψυχικό', 'Ψυχικό': 'Φιλοθέη-Ψυχικό', 'Φιλοθέης': 'Φιλοθέη-Ψυχικό', 'Ψυχικού': 'Φιλοθέη-Ψυχικό'})
408 elif (municipality == 'Νέας Ιωνίας'):
409     dictionary_municipalities.update({'Νέας Ιωνίας': 'Νέα Ιωνία', 'Νέα Ιωνία': 'Νέα Ιωνία'})
410 elif (municipality == 'Μεταμορφώσεως'):
411     dictionary_municipalities.update({'Μεταμορφώσεως': 'Μεταμόρφωση', 'Μεταμόρφωσης': 'Μεταμόρφωση',
'Μεταμόρφωση': 'Μεταμόρφωση'})
412 elif (municipality == 'Ηρακλείου'):
413     dictionary_municipalities.update({'Ηρακλείου': 'Ηράκλειο', 'Ηράκλειο': 'Ηράκλειο'})
414 elif (municipality == 'Λυκόβруση-Πεύκης'):
415     dictionary_municipalities.update({'Λυκόβруση-Πεύκης': 'Λυκόβруση-Πεύκη', 'Λυκόβруση-Πεύκη': 'Λυκόβруση-Πεύκη',
'Λυκόβруση': 'Λυκόβруση-Πεύκη', 'Πεύκη': 'Λυκόβруση-Πεύκη', 'Λυκόβруσης': 'Λυκόβруση-Πεύκη', 'Πεύκης': 'Λυκόβруση-Πεύκη'})
416 elif (municipality == 'Κηφισιάς'):
417     dictionary_municipalities.update({'Κηφισιάς': 'Κηφισιά', 'Κηφισιά': 'Κηφισιά'})
418 elif (municipality == 'Πεντέλης'):
419     dictionary_municipalities.update({'Πεντέλης': 'Πεντέλη', 'Πεντέλη': 'Πεντέλη'})
420 elif (municipality == 'Μοσχάτου-Ταύρου'):
421     dictionary_municipalities.update({'Μοσχάτου-Ταύρου': 'Μοσχάτο-Ταύρος', 'Μοσχάτο-Ταύρος': 'Μοσχάτο-Ταύρος',
'Μοσχάτο': 'Μοσχάτο-Ταύρος', 'Ταύρος': 'Μοσχάτο-Ταύρος', 'Μοσχάτου': 'Μοσχάτο-Ταύρος', 'Ταύρου': 'Μοσχάτο-Ταύρος', 'Μοσχάτο-
Ταύρο': 'Μοσχάτο-Ταύρος'})
422 elif (municipality == 'Καλλιθέας'):
423     dictionary_municipalities.update({'Καλλιθέας': 'Καλλιθέα', 'Καλλιθέα': 'Καλλιθέα'})
424 elif (municipality == 'Νέας Σμύρνης'):
425     dictionary_municipalities.update({'Νέας Σμύρνης': 'Νέα Σμύρνη', 'Νέα Σμύρνη': 'Νέα Σμύρνη'})
426 elif (municipality == 'Παλαιού Φαλήρου'):
427     dictionary_municipalities.update({'Παλαιού Φαλήρου': 'Παλαιό Φάληρο', 'Παλαιό Φάληρο': 'Παλαιό Φάληρο'})
428 elif (municipality == 'Αγίου Δημητρίου'):
429     dictionary_municipalities.update({'Αγίου Δημητρίου': 'Άγιος Δημήτριος', 'Άγιος Δημήτριος': 'Άγιος Δημήτριος', 'Άγιο
Δημήτριος': 'Άγιος Δημήτριος'})
430 elif (municipality == 'Αλίμου'):
431     dictionary_municipalities.update({'Αλίμου': 'Άλιμος', 'Άλιμος': 'Άλιμος', 'Άλιμο': 'Άλιμος'})
432 elif (municipality == 'Ελληνικού-Αργυρούπολης'):
433     dictionary_municipalities.update({'Ελληνικού-Αργυρούπολης': 'Ελληνικό-Αργυρούπολη', 'Ελληνικό-Αργυρούπολη': 'Ελληνικό-
Αργυρούπολη', 'Ελληνικό': 'Ελληνικό-Αργυρούπολη', 'Αργυρούπολη': 'Ελληνικό-Αργυρούπολη', 'Ελληνικού': 'Ελληνικό-Αργυρούπολη',
'Αργυρούπολης': 'Ελληνικό-Αργυρούπολη'})
434 elif (municipality == 'Γλυφάδας'):
435     dictionary_municipalities.update({'Γλυφάδας': 'Γλυφάδα', 'Γλυφάδα': 'Γλυφάδα'})
436 elif (municipality == 'Βάρης - Βούλας - Βουλιαγμένης'):
437     dictionary_municipalities.update({'Βάρης - Βούλας - Βουλιαγμένης': 'Βάρη - Βούλα - Βουλιαγμένη', 'Βάρη - Βούλα -
Βουλιαγμένη': 'Βάρη - Βούλα - Βουλιαγμένη', 'Βάρη': 'Βάρη - Βούλα - Βουλιαγμένη', 'Βούλα': 'Βάρη - Βούλα - Βουλιαγμένη',
'Βουλιαγμένη': 'Βάρη - Βούλα - Βουλιαγμένη', 'Βάρης': 'Βάρη - Βούλα - Βουλιαγμένη', 'Βούλας': 'Βάρη - Βούλα - Βουλιαγμένη',
'Βουλιαγμένης': 'Βάρη - Βούλα - Βουλιαγμένη'})

```

Εικόνα 18: Κώδικας που επεξεργάζεται δήμους (404-437)

```

438 elif (municipality == 'Κρωπίας'):
439     dictionary_municipalities.update({'Κρωπίας': 'Κρωπία', 'Κρωπία': 'Κρωπία'})
440 elif (municipality == 'Σαρωνικού'):
441     dictionary_municipalities.update({'Σαρωνικού': 'Σαρωνικός', 'Σαρωνικός': 'Σαρωνικός', 'Σαρωνικό': 'Σαρωνικός'})
442 elif (municipality == 'Λαυρεωτικής'):
443     dictionary_municipalities.update({'Λαυρεωτικής': 'Λαυρεωτική', 'Λαυρεωτική': 'Λαυρεωτική'})
444 elif (municipality == 'Μαρκόπουλου Μεσαγιάς'):
445     dictionary_municipalities.update({'Μαρκόπουλου Μεσαγιάς': 'Μαρκόπουλο Μεσαγιάς', 'Μαρκόπουλο Μεσαγιάς': 'Μαρκόπουλο
Μεσαγιάς'})
446 elif (municipality == 'Παιονίας'):
447     dictionary_municipalities.update({'Παιονίας': 'Παιονία', 'Παιονία': 'Παιονία'})
448 elif (municipality == 'Σπάτων-Αρτέμιδος'):
449     dictionary_municipalities.update({'Σπάτων-Αρτέμιδος': 'Σπάτα-Αρτέμιδα', 'Σπάτα-Αρτέμιδα': 'Σπάτα-Αρτέμιδα', 'Σπάτα': 'Σπάτα-
Αρτέμιδα', 'Αρτέμιδα': 'Σπάτα-Αρτέμιδα', 'Σπάτων': 'Σπάτα-Αρτέμιδα', 'Αρτέμιδος': 'Σπάτα-Αρτέμιδα'})
450 elif (municipality == 'Πολλήνης'):
451     dictionary_municipalities.update({'Πολλήνης': 'Πολλήνη', 'Πολλήνη': 'Πολλήνη'})
452 elif (municipality == 'Ραφήνας-Πικερμίου'):
453     dictionary_municipalities.update({'Ραφήνας-Πικερμίου': 'Ραφήνα-Πικέρμι', 'Ραφήνα-Πικέρμι': 'Ραφήνα-Πικέρμι',
'Ραφήνα': 'Ραφήνα-Πικέρμι', 'Πικέρμι': 'Ραφήνα-Πικέρμι', 'Ραφήνας': 'Ραφήνα-Πικέρμι', 'Πικερμίου': 'Ραφήνα-Πικέρμι'})
454 elif (municipality == 'Μαραθώνος'):
455     dictionary_municipalities.update({'Μαραθώνος': 'Μαραθόνας', 'Μαραθόνας': 'Μαραθόνας', 'Μαραθόνα': 'Μαραθόνας'})
456 elif (municipality == 'Διονύσου'):
457     dictionary_municipalities.update({'Διονύσου': 'Διδόνυσος', 'Διδόνυσος': 'Διδόνυσος', 'Διδόνυσο': 'Διδόνυσος'})
458 elif (municipality == 'Θρωπού'):
459     dictionary_municipalities.update({'Θρωπού': 'Θρωπός', 'Θρωπός': 'Θρωπός', 'Θρωπό': 'Θρωπός'})
460 elif (municipality == 'Αχαρνών'):
461     dictionary_municipalities.update({'Αχαρνών': 'Αχαρνές', 'Αχαρνές': 'Αχαρνές'})
462 elif (municipality == 'Φυλής'):
463     dictionary_municipalities.update({'Φυλής': 'Φυλή', 'Φυλή': 'Φυλή'})
464 elif (municipality == 'Ασπροπύργου'):
465     dictionary_municipalities.update({'Ασπροπύργου': 'Ασπρόπυργος', 'Ασπρόπυργος': 'Ασπρόπυργος', 'Ασπρόπυργο': 'Ασπρόπυργος'})
466 elif (municipality == 'Ελευσινίας'):
467     dictionary_municipalities.update({'Ελευσινίας': 'Ελευσίνα', 'Ελευσίνα': 'Ελευσίνα'})
468 elif (municipality == 'Μάνδρας-Ειδυλλίας'):
469     dictionary_municipalities.update({'Μάνδρας-Ειδυλλίας': 'Μάνδρα-Ειδυλλία', 'Μάνδρα-Ειδυλλία': 'Μάνδρα-Ειδυλλία', 'Μάνδρα': 'Μάνδρα-Ειδυλλία',
'Ειδυλλία': 'Μάνδρα-Ειδυλλία', 'Ειδυλλίας': 'Μάνδρα-Ειδυλλία'})
470 elif (municipality == 'Μεγαρέων'):
471     dictionary_municipalities.update({'Μεγαρέων': 'Μέγαρα', 'Μεγάρων': 'Μέγαρα', 'Μέγαρα': 'Μέγαρα'})
472 elif (municipality == 'Λουτρακίου-Αγίων Θεοδώρων'):
473     dictionary_municipalities.update({'Λουτρακίου-Αγίων Θεοδώρων': 'Λουτράκι-Άγιοι Θεόδωροι', 'Λουτράκι-Άγιοι
Θεόδωροι': 'Λουτράκι-Άγιοι Θεόδωροι', 'Λουτράκι': 'Λουτράκι-Άγιοι Θεόδωροι', 'Άγιοι Θεόδωροι': 'Λουτράκι-Άγιοι Θεόδωροι',
'Λουτρακίου': 'Λουτράκι-Άγιοι Θεόδωροι', 'Άγιων Θεοδώρων': 'Λουτράκι-Άγιοι Θεόδωροι', 'Λουτράκι-Άγίους Θεοδώρους': 'Λουτράκι-Άγιοι
Θεόδωροι', 'Άγίους Θεοδώρους': 'Λουτράκι-Άγιοι Θεόδωροι'})

```

Εικόνα 19: Κώδικας που επεξεργάζεται δήμους (438-473)

```

474 elif (municipality == 'Κορινθίων'):
475     dictionary_municipalities.update({'Κορινθίων': 'Κόρινθος', 'Κορινθίας': 'Κόρινθος', 'Κορινθία': 'Κόρινθος',
'Κόρινθος': 'Κόρινθος', 'Κόρινθο': 'Κόρινθος'})
476 elif (municipality == 'Βέλου-Βόχας'):
477     dictionary_municipalities.update({'Βέλου-Βόχας': 'Βέλο-Βόχα', 'Βέλο-Βόχα': 'Βέλο-Βόχα', 'Βέλο': 'Βέλο-Βόχα', 'Βόχα': 'Βέλο-
Βόχα', 'Βέλου': 'Βέλο-Βόχα', 'Βόχας': 'Βέλο-Βόχα'})
478 elif (municipality == 'Σικωνίων'):
479     dictionary_municipalities.update({'Σικωνίων': 'Σικωνίες', 'Σικωνίες': 'Σικωνίες'})
480 elif (municipality == 'Ευλοκάστρου-Ευρωστίνης'):
481     dictionary_municipalities.update({'Ευλοκάστρου-Ευρωστίνης': 'Ευλόκαστρο-Ευρωστίνη', 'Ευλόκαστρο-Ευρωστίνη': 'Ευλόκαστρο-
Ευρωστίνη', 'Ευρωστίνη': 'Ευλόκαστρο-Ευρωστίνη', 'Ευρωστίνη': 'Ευλόκαστρο-Ευρωστίνη', 'Ευλοκάστρου': 'Ευλόκαστρο-Ευρωστίνη',
'Ευρωστίνης': 'Ευλόκαστρο-Ευρωστίνη'})
482 elif (municipality == 'Νεμέας'):
483     dictionary_municipalities.update({'Νεμέας': 'Νεμέα', 'Νεμέα': 'Νεμέα'})
484 elif (municipality == 'Δήμος Άργους - Μυκηνών'):
485     dictionary_municipalities.update({'Άργους - Μυκηνών': 'Άργος - Μυκήνες', 'Άργος - Μυκήνες': 'Άργος - Μυκήνες',
'Άργος': 'Άργος - Μυκήνες', 'Μυκηνών': 'Άργος - Μυκήνες', 'Άργους': 'Άργος - Μυκήνες', 'Μυκηνών': 'Άργος - Μυκήνες'})
486 elif (municipality == 'Ναυπλίων'):
487     dictionary_municipalities.update({'Ναυπλίων': 'Ναύπλιο', 'Ναυπλίου': 'Ναύπλιο', 'Ναύπλιο': 'Ναύπλιο'})
488 elif (municipality == 'Επιδαύρου'):
489     dictionary_municipalities.update({'Επιδαύρου': 'Επίδαυρος', 'Επίδαυρος': 'Επίδαυρος', 'Επίδαυρο': 'Επίδαυρος'})
490 elif (municipality == 'Ερμιονίδας'):
491     dictionary_municipalities.update({'Ερμιονίδας': 'Ερμιονίδα', 'Ερμιονίδα': 'Ερμιονίδα'})
492 elif (municipality == 'Νότιας Κυνουρίας'):
493     dictionary_municipalities.update({'Νότιας Κυνουρίας': 'Νότια Κυνουρία', 'Νότια Κυνουρία': 'Νότια Κυνουρία'})
494 elif (municipality == 'Βόρειας Κυνουρίας'):
495     dictionary_municipalities.update({'Βόρειας Κυνουρίας': 'Βόρεια Κυνουρία', 'Βόρεια Κυνουρία': 'Βόρεια Κυνουρία'})
496 elif (municipality == 'Τρίπολης'):
497     dictionary_municipalities.update({'Τρίπολης': 'Τρίπολη', 'Τρίπολη': 'Τρίπολη'})
498 elif (municipality == 'Γορτυνίας'):
499     dictionary_municipalities.update({'Γορτυνίας': 'Γορτυνία', 'Γορτυνία': 'Γορτυνία'})
500 elif (municipality == 'Μεγαλόπολης'):
501     dictionary_municipalities.update({'Μεγαλόπολης': 'Μεγαλόπολη', 'Μεγαλόπολη': 'Μεγαλόπολη'})
502 elif (municipality == 'Δήμος Οιχαλίας'):
503     dictionary_municipalities.update({'Οιχαλίας': 'Οιχαλία', 'Οιχαλία': 'Οιχαλία'})
504 elif (municipality == 'Τριφυλίας'):
505     dictionary_municipalities.update({'Τριφυλίας': 'Τριφυλία', 'Τριφυλία': 'Τριφυλία'})
506 elif (municipality == 'Πύλου - Νέστορος'):
507     dictionary_municipalities.update({'Πύλου - Νέστορος': 'Πύλος Νέστωρ', 'Πύλος Νέστωρ': 'Πύλος Νέστωρ', 'Πύλος Νέστωρ': 'Πύλος
Νέστωρ', 'Πύλου': 'Πύλος Νέστωρ', 'Πύλο': 'Πύλος Νέστωρ'})
508 elif (municipality == 'Μεσσήνης'):
509     dictionary_municipalities.update({'Μεσσήνης': 'Μεσσήνη', 'Μεσσήνη': 'Μεσσήνη'})
510 elif (municipality == 'Καλαμάτας'):
511     dictionary_municipalities.update({'Καλαμάτας': 'Καλαμάτα', 'Καλαμάτα': 'Καλαμάτα'})
512 elif (municipality == 'Δυτικής Μάνης'):
513     dictionary_municipalities.update({'Δυτικής Μάνης': 'Δυτική Μάνη', 'Δυτική Μάνη': 'Δυτική Μάνη'})

```

Εικόνα 20: Κώδικας που επεξεργάζεται δήμους (474-513)

```

514 elif (municipality == 'Ανατολικής Μάνης'):
515     dictionary_municipalities.update({'Ανατολικής Μάνης': 'Ανατολική Μάνη', 'Ανατολική Μάνη': 'Ανατολική Μάνη'})
516 elif (municipality == 'Σπάρτης'):
517     dictionary_municipalities.update({'Σπάρτης': 'Σπάρτη', 'Σπάρτη': 'Σπάρτη'})
518 elif (municipality == 'Ευρώτας'):
519     dictionary_municipalities.update({'Ευρώτας': 'Ευρώτας', 'Ευρώτας': 'Ευρώτας'})
520 elif (municipality == 'Μονεμβασιάς'):
521     dictionary_municipalities.update({'Μονεμβασιάς': 'Μονεμβασία', 'Μονεμβασία': 'Μονεμβασία'})
522 elif (municipality == 'Ελαφονήσου'):
523     dictionary_municipalities.update({'Ελαφονήσου': 'Ελαφονήσος', 'Ελαφονήσος': 'Ελαφονήσος', 'Ελαφονήσο': 'Ελαφονήσος'})
524 elif (municipality == 'Λήμνου'):
525     dictionary_municipalities.update({'Λήμνου': 'Λήμνος', 'Λήμνος': 'Λήμνος', 'Λήμνο': 'Λήμνος'})
526 elif (municipality == 'Αγίου Ευστρατίου'):
527     dictionary_municipalities.update({'Αγίου Ευστρατίου': 'Άγιος Ευστράτιος', 'Άγιος Ευστράτιος': 'Άγιος Ευστράτιος', 'Άγιο
Ευστράτιο': 'Άγιος Ευστράτιος'})
528 elif (municipality == 'Λέσβου'):
529     dictionary_municipalities.update({'Λέσβου': 'Λέσβος', 'Λέσβος': 'Λέσβος', 'Λέσβο': 'Λέσβος'})
530 elif (municipality == 'Ηρωικής Νήσου Ψαρών'):
531     dictionary_municipalities.update({'Ηρωικής Νήσου Ψαρών': 'Ψαρά', 'Ηρωική Νήσος Ψαρών': 'Ψαρά', 'Ψαρά': 'Ψαρά',
'Ψαρών': 'Ψαρά', 'Ηρωική Νήσο Ψαρών': 'Ψαρά'})
532 elif (municipality == 'Οινουσσών'):
533     dictionary_municipalities.update({'Οινουσσών': 'Οινούσσεζ', 'Οινούσσεζ': 'Οινούσσεζ'})
534 elif (municipality == 'Χίου'):
535     dictionary_municipalities.update({'Χίου': 'Χίος', 'Χίος': 'Χίος', 'Χίο': 'Χίος'})
536 elif (municipality == 'Σάμου'):
537     dictionary_municipalities.update({'Σάμου': 'Σάμος', 'Σάμος': 'Σάμος', 'Σάμο': 'Σάμος'})
538 elif (municipality == 'Φούρνων Κορσεών'):
539     dictionary_municipalities.update({'Φούρνων Κορσεών': 'Φούρνοι Κορσεών', 'Φούρνοι Κορσεών': 'Φούρνοι Κορσεών', 'Φούρνοι
Κορσεών': 'Φούρνοι Κορσεών'})
540 elif (municipality == 'Ικαρίας'):
541     dictionary_municipalities.update({'Ικαρίας': 'Ικαρία', 'Ικαρία': 'Ικαρία'})
542 elif (municipality == 'Ανδρού'):
543     dictionary_municipalities.update({'Ανδρού': 'Ανδρος', 'Ανδρος': 'Ανδρος', 'Ανδρο': 'Ανδρος'})
544 elif (municipality == 'Τήνου'):
545     dictionary_municipalities.update({'Τήνου': 'Τήνος', 'Τήνος': 'Τήνος', 'Τήνο': 'Τήνος'})
546 elif (municipality == 'Κέα'):
547     dictionary_municipalities.update({'Κέα': 'Κέα', 'Κέα': 'Κέα'})
548 elif (municipality == 'Κύθνου'):
549     dictionary_municipalities.update({'Κύθνου': 'Κύθνος', 'Κύθνος': 'Κύθνος', 'Κύθνο': 'Κύθνος'})
550 elif (municipality == 'Σύρου - Ερμούπολης'):
551     dictionary_municipalities.update({'Σύρου - Ερμούπολης': 'Σύρος - Ερμούπολη', 'Σύρος - Ερμούπολη': 'Σύρος - Ερμούπολη',
'Σύρος': 'Σύρος - Ερμούπολη', 'Ερμούπολη': 'Σύρος - Ερμούπολη', 'Ερμούπολη': 'Σύρος - Ερμούπολη', 'Σύρο - Ερμούπολη': 'Σύρος - Ερμούπολη'})
552 elif (municipality == 'Μυκόνου'):
553     dictionary_municipalities.update({'Μυκόνου': 'Μύκονος', 'Μύκονος': 'Μύκονος', 'Μύκονο': 'Μύκονος'})
554 elif (municipality == 'Σεριφου'):
555     dictionary_municipalities.update({'Σεριφου': 'Σέριφος', 'Σέριφος': 'Σέριφος', 'Σέριφο': 'Σέριφος'})

```

Εικόνα 21: Κώδικας που επεξεργάζεται δήμους (514-555)

```

556 elif (municipality == 'Σίφνου'):
557     dictionary_municipalities.update({'Σίφνου': 'Σίφνος', 'Σίφνος': 'Σίφνος', 'Σίφνο': 'Σίφνος'})
558 elif (municipality == 'Κιμώλου'):
559     dictionary_municipalities.update({'Κιμώλου': 'Κίμωλος', 'Κίμωλος': 'Κίμωλος', 'Κίμωλο': 'Κίμωλος'})
560 elif (municipality == 'Μήλου'):
561     dictionary_municipalities.update({'Μήλου': 'Μήλος', 'Μήλος': 'Μήλος', 'Μήλο': 'Μήλος'})
562 elif (municipality == 'Αντιπάρου'):
563     dictionary_municipalities.update({'Αντιπάρου': 'Αντιπάρος', 'Αντιπάρος': 'Αντιπάρος', 'Αντιπαρο': 'Αντιπάρος'})
564 elif (municipality == 'Πάρου'):
565     dictionary_municipalities.update({'Πάρου': 'Πάρος', 'Πάρος': 'Πάρος', 'Πάρο': 'Πάρος'})
566 elif (municipality == 'Νάξου & Μικρών Κυκλάδων'):
567     dictionary_municipalities.update({'Νάξου & Μικρών Κυκλάδων': 'Νάξος & Μικρές Κυκλάδες', 'Νάξος & Μικρές Κυκλάδες': 'Νάξος &
Μικρές Κυκλάδες', 'Νάξος': 'Νάξος & Μικρές Κυκλάδες', 'Μικρές Κυκλάδες': 'Νάξος & Μικρές Κυκλάδες', 'Νάξου': 'Νάξος & Μικρές Κυκλάδες',
'Μικρών Κυκλάδων': 'Νάξος & Μικρές Κυκλάδες', 'Νάξο & Μικρές Κυκλάδες': 'Νάξος & Μικρές Κυκλάδες', 'Νάξο': 'Νάξος & Μικρές Κυκλάδες'})
568 elif (municipality == 'Άμοργού'):
569     dictionary_municipalities.update({'Άμοργού': 'Άμοργός', 'Άμοργός': 'Άμοργός', 'Άμοργό': 'Άμοργός'})
570 elif (municipality == 'Ιητών'):
571     dictionary_municipalities.update({'Ιητών': 'Ιος', 'Ιος': 'Ιος', 'Ιο': 'Ιος'})
572 elif (municipality == 'Σικίνου'):
573     dictionary_municipalities.update({'Σικίνου': 'Σίκινος', 'Σίκινος': 'Σίκινος', 'Σίκινο': 'Σίκινος'})
574 elif (municipality == 'Φολεγάνδρου'):
575     dictionary_municipalities.update({'Φολεγάνδρου': 'Φολεγάνδρος', 'Φολεγάνδρος': 'Φολεγάνδρος', 'Φολεγανδρο': 'Φολεγάνδρος'})
576 elif (municipality == 'Θήρας'):
577     dictionary_municipalities.update({'Θήρας': 'Σαντορίνη', 'Θήρα': 'Σαντορίνη', 'Σαντορίνης': 'Σαντορίνη',
'Σαντορίνη': 'Σαντορίνη'})
578 elif (municipality == 'Ανάφης'):
579     dictionary_municipalities.update({'Ανάφης': 'Ανάφη', 'Ανάφη': 'Ανάφη'})
580 elif (municipality == 'Αγαθονησιού'):
581     dictionary_municipalities.update({'Αγαθονησιού': 'Αγαθονήσι', 'Αγαθονήσι': 'Αγαθονήσι'})
582 elif (municipality == 'Πάτμου'):
583     dictionary_municipalities.update({'Πάτμου': 'Πάτμος', 'Πάτμος': 'Πάτμος', 'Πάτμο': 'Πάτμος'})
584 elif (municipality == 'Λειψών'):
585     dictionary_municipalities.update({'Λειψών': 'Λειψοί', 'Λειψοί': 'Λειψοί', 'Λειψούς': 'Λειψοί'})
586 elif (municipality == 'Λέρου'):
587     dictionary_municipalities.update({'Λέρου': 'Λέρος', 'Λέρος': 'Λέρος', 'Λέρο': 'Λέρος'})
588 elif (municipality == 'Καλύμνου'):
589     dictionary_municipalities.update({'Καλύμνου': 'Κάλυμνος', 'Κάλυμνος': 'Κάλυμνος', 'Κάλυμνο': 'Κάλυμνος'})
590 elif (municipality == 'Αστυπάλαιας'):
591     dictionary_municipalities.update({'Αστυπάλαιας': 'Αστυπάλαια', 'Αστυπάλαια': 'Αστυπάλαια'})
592 elif (municipality == 'Κω'):
593     dictionary_municipalities.update({'Κω': 'Κως', 'Κως': 'Κως'})
594 elif (municipality == 'Νισύρου'):
595     dictionary_municipalities.update({'Νισύρου': 'Νίσυρος', 'Νίσυρος': 'Νίσυρος', 'Νίσυρο': 'Νίσυρος'})

```

Εικόνα 22: Κώδικας που επεξεργάζεται δήμους (556-595)

```

596 elif (municipality == 'Τήλου'):
597     dictionary_municipalities.update({'Τήλου': 'Τήλος', 'Τήλος': 'Τήλος', 'Τήλο': 'Τήλος'})
598 elif (municipality == 'Σύμης'):
599     dictionary_municipalities.update({'Σύμης': 'Σύμη', 'Σύμη': 'Σύμη'})
600 elif (municipality == 'Ρόδου'):
601     dictionary_municipalities.update({'Ρόδου': 'Ρόδος', 'Ρόδος': 'Ρόδος', 'Ρόδο': 'Ρόδος'})
602 elif (municipality == 'Χάλκης'):
603     dictionary_municipalities.update({'Χάλκης': 'Χάλκη', 'Χάλκη': 'Χάλκη'})
604 elif (municipality == 'Μεγίστης'):
605     dictionary_municipalities.update({'Μεγίστης': 'Μεγίστη', 'Μεγίστη': 'Μεγίστη'})
606 elif (municipality == 'Καρπάθου'):
607     dictionary_municipalities.update({'Καρπάθου': 'Κάρπαθος', 'Κάρπαθος': 'Κάρπαθος', 'Κάρπαθο': 'Κάρπαθος'})
608 elif (municipality == 'Ηρωικής Νήσου Κάσου'):
609     dictionary_municipalities.update({'Ηρωικής Νήσου Κάσου': 'Κάσος', 'Ηρωική Νήσου Κάσου': 'Κάσος', 'Κάσος': 'Κάσος',
610 'Κάσου': 'Κάσος', 'Ηρωική Νήσο Κάσου': 'Κάσος', 'Κάσο': 'Κάσος'})
611 elif (municipality == 'Σητείας'):
612     dictionary_municipalities.update({'Σητείας': 'Σητεία', 'Σητεία': 'Σητεία'})
613 elif (municipality == 'Ιεράπετρας'):
614     dictionary_municipalities.update({'Ιεράπετρας': 'Ιεράπετρα', 'Ιεράπετρα': 'Ιεράπετρα'})
615 elif (municipality == 'Αγίου Νικολάου'):
616     dictionary_municipalities.update({'Αγίου Νικολάου': 'Άγιος Νικόλαος', 'Άγιος Νικόλαος': 'Άγιος Νικόλαος', 'Άγιο
617 Νικόλαο': 'Άγιος Νικόλαος'})
618 elif (municipality == 'Οροπέδιου Λασιθίου'):
619     dictionary_municipalities.update({'Οροπέδιου Λασιθίου': 'Οροπέδιο Λασιθίου', 'Οροπέδιο Λασιθίου': 'Οροπέδιο Λασιθίου'})
620 elif (municipality == 'Βιάννου'):
621     dictionary_municipalities.update({'Βιάννου': 'Βιάννος', 'Βιάννος': 'Βιάννος', 'Βιάννο': 'Βιάννος'})
622 elif (municipality == 'Δήμος Μινώα Πεδιάδας'):
623     dictionary_municipalities.update({'Μινώα Πεδιάδας': 'Μινώα Πεδιάδα', 'Μινώα Πεδιάδα': 'Μινώα Πεδιάδα'})
624 elif (municipality == 'Αρχανών - Αστερουσίων'):
625     dictionary_municipalities.update({'Αρχανών - Αστερουσίων': 'Αρχανές - Αστερούσια', 'Αρχανές - Αστερούσια': 'Αρχανές -
626 Αστερούσια', 'Αρχανές': 'Αρχανές - Αστερούσια', 'Αστερούσια': 'Αρχανές - Αστερούσια', 'Αρχανών': 'Αρχανές - Αστερούσια',
627 'Αστερουσίων': 'Αρχανές - Αστερούσια'})
628 elif (municipality == 'Χερσονήσου'):
629     dictionary_municipalities.update({'Χερσονήσου': 'Χερσόνησος', 'Χερσόνησος': 'Χερσόνησος', 'Χερσόνησο': 'Χερσόνησος'})
630 elif (municipality == 'Μαλεβιζίου'):
631     dictionary_municipalities.update({'Μαλεβιζίου': 'Μαλεβίτζι', 'Μαλεβίτζι': 'Μαλεβίτζι'})
632 elif (municipality == 'Τόρτινας'):
633     dictionary_municipalities.update({'Τόρτινας': 'Τόρτινα', 'Τόρτινα': 'Τόρτινα'})
634 elif (municipality == 'Φαιστού'):
635     dictionary_municipalities.update({'Φαιστού': 'Φαιστός', 'Φαιστός': 'Φαιστός', 'Φαιστό': 'Φαιστός'})
636 elif (municipality == 'Ανωγείων'):
637     dictionary_municipalities.update({'Ανωγείων': 'Ανώγεια', 'Ανώγεια': 'Ανώγεια'})
638 elif (municipality == 'Μυλοποτάμου'):
639     dictionary_municipalities.update({'Μυλοποτάμου': 'Μυλοπόταμος', 'Μυλοπόταμος': 'Μυλοπόταμος', 'Μυλοπόταμο': 'Μυλοπόταμος'})
640 elif (municipality == 'Αμαρίου'):
641     dictionary_municipalities.update({'Αμαρίου': 'Αμάρι', 'Αμάρι': 'Αμάρι'})

```

Εικόνα 23: Κώδικας που επεξεργάζεται δήμους (596-637)

```

638 elif (municipality == 'Αγίου Βασιλείου'):
639     dictionary_municipalities.update({'Αγίου Βασιλείου': 'Άγιος Βασίλειος', 'Άγιος Βασίλειος': 'Άγιος Βασίλειος', 'Άγιο
640 Βασίλειο': 'Άγιος Βασίλειος'})
641 elif (municipality == 'Ρεθύμνης'):
642     dictionary_municipalities.update({'Ρεθύμνης': 'Ρέθυμνο', 'Ρεθύμνου': 'Ρέθυμνο', 'Ρέθυμνο': 'Ρέθυμνο'})
643 elif (municipality == 'Αποκορώνου'):
644     dictionary_municipalities.update({'Αποκορώνου': 'Αποκόρωνος', 'Αποκόρωνος': 'Αποκόρωνος', 'Αποκόρωνο': 'Αποκόρωνος'})
645 elif (municipality == 'Χανίων'):
646     dictionary_municipalities.update({'Χανίων': 'Χανιά', 'Χανιά': 'Χανιά'})
647 elif (municipality == 'Πλατανιάς'):
648     dictionary_municipalities.update({'Πλατανιάς': 'Πλατανιάς', 'Πλατανιάς': 'Πλατανιάς'})
649 elif (municipality == 'Κισσάμου'):
650     dictionary_municipalities.update({'Κισσάμου': 'Κίσαμος', 'Κίσαμος': 'Κίσαμος', 'Κίσαμο': 'Κίσαμος'})
651 elif (municipality == 'Καντάνου - Σελίνου'):
652     dictionary_municipalities.update({'Καντάνου - Σελίνου': 'Κάντανος Σέλινο', 'Κάντανος Σέλινο': 'Κάντανος Σέλινο',
653 'Κάντανος': 'Κάντανος Σέλινο', 'Καντάνου': 'Κάντανος Σέλινο', 'Κάντανο Σέλινο': 'Κάντανος Σέλινο', 'Κάντανο': 'Κάντανος Σέλινο'})
654 elif (municipality == 'Σφακίων'):
655     dictionary_municipalities.update({'Σφακίων': 'Σφακιά', 'Σφακιά': 'Σφακιά'})
656 elif (municipality == 'Γαυδίου'):
657     dictionary_municipalities.update({'Γαυδίου': 'Γαύδος', 'Γαύδος': 'Γαύδος', 'Γαύδο': 'Γαύδος'})
658 return dictionary_municipalities

```

Εικόνα 24: Κώδικας που επεξεργάζεται δήμους (638-656)

Η συνάρτηση `create_dictionary_municipalities` επεξεργάζεται τους δήμους και τους αποθηκεύει σε ένα dictionary.

Ο κώδικας που είναι γραμμένος στο αρχείο `find_locations.py`

είναι ο κώδικας της Εικόνας 25.

```

1 from bs4 import BeautifulSoup
2 import requests
3
4 def find_locations(type_location):
5     if (type_location == "municipalities"):
6         municipalities = []
7         req = requests.get("https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%AC%CE%BB%CE%BF%CE%B3%CE%BF%CF%82_%CE%B4%CE%AE%CE%BC%CF%89%CE%BD_%CF%84%CE%B7%CF%82_%CE%95%CE%BB%CE%BB%CE%AC%CE%B4%CE%B1%CF%82")
8         soup = BeautifulSoup(req.content, 'html.parser')
9         table = soup.find("table")
10        for row in table.tbody.findAll("tr")[1:]:
11            col = row.findAll("td")
12            municipality = col[0].text
13            if (municipality not in municipalities):
14                municipalities.append(municipality)
15        return municipalities
16    elif (type_location == "counties"):
17        counties = []
18        req = requests.get("https://el.wikipedia.org/wiki/%CE%9A%CE%B1%CF%84%CE%AC%CE%BB%CE%BF%CE%B3%CE%BF%CF%82_%CE%B4%CE%AE%CE%BC%CF%89%CE%BD_%CF%84%CE%B7%CF%82_%CE%95%CE%BB%CE%BB%CE%AC%CE%B4%CE%B1%CF%82")
19        soup = BeautifulSoup(req.content, 'html.parser')
20        table = soup.find("table")
21        for row in table.tbody.findAll("tr")[1:]:
22            col = row.findAll("td")
23            county = col[2].text
24            if (county not in counties):
25                counties.append(county)
26        return counties

```

Εικόνα 25: Κώδικας που βρίσκει ελληνικές τοποθεσίες

Η συνάρτηση `find_locations` παίρνει σαν όρισμα είδος τοποθεσίας (δήμοι, νομοί). Αν το όρισμα που δίνεται είναι 'municipalities' τότε αποθηκεύει σε μία λίστα τους δήμους. Αν όμως το όρισμα που δίνεται είναι 'counties' τότε αποθηκεύει σε μία λίστα τους νομούς.

Ο κώδικας που είναι γραμμένος στο αρχείο `main.py` είναι ο κώδικας της Εικόνας 26.

```

1 import json
2 import find_locations
3 import create_dictionary_municipalities
4 import create_dictionary_counties
5
6 def save_dictionary_to_json_file(data, filename):
7     with open(filename, 'w') as file:
8         json.dump(data, file, ensure_ascii=False)
9
10 def main():
11     municipalities = find_locations.find_locations("municipalities")
12     counties = find_locations.find_locations("counties")
13     dictionary_municipalities = create_dictionary_municipalities.create_dictionary_municipalities(municipalities)
14     dictionary_counties = create_dictionary_counties.create_dictionary_counties(counties)
15     save_dictionary_to_json_file(dictionary_municipalities, "Datasets/municipalities.json")
16     save_dictionary_to_json_file(dictionary_counties, "Datasets/counties.json")
17     print "The dictionaries were saved successfully."
18
19 main()

```

Εικόνα 26: Κώδικας που επεξεργάζεται ελληνικές τοποθεσίες

Οι συναρτήσεις που υλοποιήθηκαν στον παραπάνω κώδικα εξηγούνται κάτωθι:

- `save_dictionary_to_json_file`: Η συνάρτηση αυτή γράφει στο αρχείο με το όνομα που δίνεται σαν όρισμα το dictionary που δίνεται σαν όρισμα.
- `main`: Η κύρια συνάρτηση του κώδικα. Καλεί όλες τις συναρτήσεις που εξηγήθηκαν παραπάνω και αποθηκεύει σε ένα αρχείο τύπου .json το dictionary με τους δήμους και σε ένα άλλο αρχείο τύπου .json το dictionary με τους νομούς. Τυπώνει και ένα μήνυμα που λέει ότι τα dictionaries αποθηκεύτηκαν επιτυχώς. Τα dictionaries αποθηκεύτηκαν

σε αρχεία .json για να είναι έτοιμα για το πρόγραμμα που εκτελεί εξαγωγή πληροφοριών όπως θα αναλυθεί παρακάτω.

3.4 Κώδικας που εκτελεί εξαγωγή πληροφοριών

Ο κώδικας που εκτελεί εξαγωγή πληροφοριών είναι ο κώδικας των Εικόνων 27, 28, 29 και 30.

```

1  # -*- coding: utf-8 -*-
2
3  from bs4 import BeautifulSoup
4  import requests
5  import pandas as pd
6  import sys
7  import csv
8  import os
9  import json
10
11 def get_digits(str1):
12     c = ""
13     for i in str1:
14         if i.isdigit():
15             c += i
16     return c
17
18 def load_dictionary(filename):
19     with open(filename) as file:
20         data = json.load(file)
21     return data
22
23 def find_numbers_distances(words, i):
24     numbers = []
25     distances = []
26     for j in range(0, len(words)):
27         if (j < i):
28             mystr = get_digits(words[j])
29             if (words[j] == mystr):
30                 numbers.append(int(mystr))
31                 distances.append(abs(j - i))
32         elif (j > i):
33             mystr = get_digits(words[j])
34             if (mystr.isdigit()):
35                 numbers.append(int(mystr))
36                 distances.append(abs(j - i))
37     return (numbers, distances)
38
39 def find_specific_type(words, word, type_name):
40     if (type_name == 'int'):
41         for i in range(0, len(words)):
42             if (word in words[i]):
43                 numbers, distances = find_numbers_distances(words, i)
44                 if (len(numbers) == 0):
45                     return -1
46                 numbers_distances = zip(numbers, distances)
47                 sorted_numbers_distances = sorted(numbers_distances, key=lambda x: x[1])
48                 new_numbers = [x[0] for x in sorted_numbers_distances]
49                 return new_numbers[0]

```

Εικόνα 27: Κώδικας που εκτελεί εξαγωγή πληροφοριών (1-49)

```

50 elif (type_name == 'greek_location'):
51     dictionary_municipalities = load_dictionary("JSON Datasets/municipalities.json")
52     dictionary_counties = load_dictionary("JSON Datasets/counties.json")
53     keys_municipalities = dictionary_municipalities.keys()
54     keys_counties = dictionary_counties.keys()
55     text = " ".join(words)
56     for key in keys_municipalities:
57         if (key in text):
58             return dictionary_municipalities[key]
59     for key in keys_counties:
60         if (key in text):
61             return dictionary_counties[key]
62     return ''
63
64 def processing_file(filename):
65     df = pd.read_csv(filename, sep=',', encoding="utf-8", engine='python')
66     df = df.groupby(['Date', 'Location'])['Cases'].max().reset_index(name='Cases')
67     df = df.sort_values('Date')
68     df['Cases'] = df['Cases'].fillna(0).astype(int)
69     df.to_csv(filename, index=False, encoding="utf-8")
70
71 def information_extraction(date, filename, word, words):
72     location = find_specific_type(words, word, 'greek_location')
73     cases = find_specific_type(words, word, 'int')
74     if (date != '' and location != '' and cases >= 0):
75         data = [date, location, cases]
76         with open(filename, 'ab') as newFile:
77             newFileWriter = csv.writer(newFile, delimiter=',')
78             newFileWriter.writerow(data)
79     processing_file(filename)
80
81 def find_neighbouring_words(word, list_words, i):
82     if (i == 0):
83         neighbouring_words = list_words[i:]
84         neighbouring_words = [neighbouring_words[0]] + [neighbouring_words[j] for j in range(1, len(neighbouring_words)) if (word not
in neighbouring_words[j])]
85         if (len(neighbouring_words) > 6):
86             neighbouring_words = neighbouring_words[0:6]
87     elif (i == len(list_words) - 1):
88         neighbouring_words = list_words
89         neighbouring_words = [neighbouring_words[j] for j in range(0, len(neighbouring_words) - 1) if (word not in
neighbouring_words[j])] + [neighbouring_words[len(neighbouring_words) - 1]]
90         if (len(neighbouring_words) > 6):
91             neighbouring_words = neighbouring_words[-6:]

```

Εικόνα 28: Κώδικας που εκτελεί εξαγωγή πληροφοριών (50-91)

```

92 else:
93     neighbouring_words1 = list_words[i:]
94     neighbouring_words1 = [neighbouring_words1[j] for j in range(0, len(neighbouring_words1)) if (word not in
neighbouring_words1[j])]
95     if (len(neighbouring_words1) > 5):
96         neighbouring_words1 = neighbouring_words1[-5:]
97     neighbouring_words2 = list_words[i:]
98     neighbouring_words2 = [neighbouring_words2[0]] + [neighbouring_words2[j] for j in range(1, len(neighbouring_words2)) if (word
not in neighbouring_words2[j])]
99     if (len(neighbouring_words2) > 6):
100         neighbouring_words2 = neighbouring_words2[0:6]
101     neighbouring_words = neighbouring_words1 + neighbouring_words2
102     return neighbouring_words
103
104 def find_date(file, filename):
105     df = pd.read_csv(filename, encoding='cp737', engine='python')
106     number = int(get_digits(file))
107     url = df.iloc[number - 1, 0]
108     req = requests.get(url)
109     soup = BeautifulSoup(req.content, 'html.parser')
110     date_article = soup.find("meta", property="article:published_time")
111     date_article = date_article["content"] if date_article else None
112     if (date_article == None):
113         return ''
114     date = date_article.split("T")
115     return date[0]
116
117 def find_words_in_html_file(words, folder, filename, filename2, prefix):
118     df = pd.read_csv(filename2, encoding='cp737', engine='python')
119     set1 = set(os.listdir(folder))
120     set2 = set(df['HTML File'])
121     html_files = list(set1 - set2)
122     for file in html_files:
123         pathname = folder + "/" + file
124         with open(pathname, "r") as f:
125             contents = f.read()
126         soup = BeautifulSoup(contents, 'html.parser')
127         soup2 = soup.prettify()
128         soup3 = BeautifulSoup(str(soup2), 'html.parser')
129         text = soup3.get_text()
130         texts = text.split('\n')
131         for word in words:
132             results = [t for t in texts if (word in t)]
133             for result in results:
134                 list_words = result.split()
135                 for i in range(0, len(list_words)):
136                     if (word in list_words[i]):

```

Εικόνα 29: Κώδικας που εκτελεί εξαγωγή πληροφοριών (92-136)

```

137         date = find_date(pathname, "CSV Datasets/" + prefix + "pages_saved.csv")
138         neighbouring_words = find_neighbouring_words(word, list_words, i)
139         information_extraction(date, filename, word, neighbouring_words)
140     with open(filename2, 'ab') as newFile:
141         newFileWriter = csv.writer(newFile)
142         newFileWriter.writerow([file])
143
144 def main():
145     reload(sys)
146     sys.setdefaultencoding("utf-8")
147     folder = "tanea.gr_files"
148     filename = "COVID-19_Cases_Greece.csv"
149     pathname = "CSV Datasets/" + filename
150     #with open(pathname, 'wb') as newFile:
151         #newFileWriter = csv.writer(newFile, delimiter=',')
152         #newFileWriter.writerow(['Date', 'Location', 'Cases'])
153     i = folder.index("_")
154     prefix = folder[0:i + 1]
155     filename2 = folder + "_read.csv"
156     pathname2 = "CSV Datasets/" + filename2
157     #with open(pathname2, 'wb') as newFile2:
158         #newFileWriter = csv.writer(newFile2)
159         #newFileWriter.writerow(['HTML File'])
160     words = ['κρούσμα', 'κρούσματα', 'Κρούσμα', 'Κρούσματα']
161     find_words_in_html_file(words, folder, pathname, pathname2, prefix)
162     print "Information extraction successful."
163
164 main()

```

Εικόνα 30: Κώδικας που εκτελεί εξαγωγή πληροφοριών (137-164)

Οι συναρτήσεις που υλοποιήθηκαν στον παραπάνω κώδικα εξηγούνται κάτωθι:

- **get_digits**: Η συνάρτηση αυτή βρίσκει τα ψηφία της συμβολοσειράς που δίνεται σαν όρισμα. Παράδειγμα: Είσοδος: '20,' Έξοδος: '20'
- **load_dictionary**: Η συνάρτηση αυτή φορτώνει στην κύρια μνήμη το dictionary που είναι αποθηκευμένο στο αρχείο με το όνομα που δίνεται σαν όρισμα.
- **find_numbers_distances**: Η συνάρτηση βρίσκει από τη λίστα λέξεων (words) που δίνεται σαν όρισμα τους αριθμούς και τις αποστάσεις τους από τη λέξη στη θέση i που δίνεται σαν όρισμα. Έστω ότι πριν τη θέση i στην οποία βρίσκεται η λέξη κρούσμα/κρούσματα υπάρχει κάποιο από τα εξής: '20,', '30.', '50,', '65.'. Στην περίπτωση αυτή δεν πρέπει κανένας από τους αριθμούς αυτούς να είναι πιο κοντά στη λέξη κρούσμα/κρούσματα. Παράδειγμα: 'στη Θεσσαλονίκη είναι 20. Τα κρούσματα στην Αθήνα είναι 30.'. Σε αυτό το κείμενο ο αριθμός που είναι πιο κοντά στη λέξη κρούσματα πρέπει να είναι το 30 και όχι το 20 γιατί στο παραπάνω κείμενο ο αριθμός 30 αναφέρεται στη λέξη κρούσματα.
- **find_specific_type**: Η συνάρτηση αυτή παίρνει σαν όρισμα τη λίστα λέξεων (words) με τις συμβολοσειρές, μία λέξη και το όνομα του τύπου που ζητείται να βρούμε. Αν το όρισμα που δίνεται είναι 'int' τότε η συνάρτηση αυτή βρίσκει από τη λίστα words που δίνεται σαν όρισμα τον αριθμό που είναι πιο κοντά στη λέξη word που δίνεται σαν όρισμα. Ο αριθμός αυτός είναι τα κρούσματα κορονοϊού. Αν όμως το όρισμα που δίνεται είναι 'greek_location' τότε η συνάρτηση αυτή βρίσκει την περιοχή της Ελλάδας στο κείμενο που αποτελεί τις συμβολοσειρές της λίστας words που δίνεται σαν όρισμα. Για να μην γίνεται συνεχώς επεξεργασία των ελληνικών τοποθεσιών ενώ εκτελείται, το πρόγραμμα φορτώνει στη μνήμη το dictionary με τους δήμους και το dictionary με τους νομούς, τα οποία δημιουργήθηκαν με τον τρόπο που εξηγήθηκε στην προηγούμενη ενότητα.
- **processing_file**: Η συνάρτηση αυτή επεξεργάζεται το αρχείο με το όνομα που δίνεται σαν όρισμα. Για κάθε ζεύγος ημερομηνίας και τοποθεσίας βρίσκει το μεγαλύτερο αριθμό κρουσμάτων.

- `information_extraction`: Η συνάρτηση αυτή βρίσκει την ελληνική τοποθεσία και τον αριθμό των κρουσμάτων από τη λίστα λέξεων (`words`) που δίνεται σαν όρισμα και γράφει στο αρχείο τύπου `.csv` με το όνομα που δίνεται σαν όρισμα την ημερομηνία που δίνεται σαν όρισμα, την ελληνική τοποθεσία που ανευρέθη και τον αριθμό των κρουσμάτων που ανευρέθη. Γράφει τριάδα της μορφής ημερομηνία,τοποθεσία,κρούσματα. Στη συνέχεια καλεί τη συνάρτηση που επεξεργάζεται το αρχείο τύπου `.csv` που είπαμε πριν.
- `find_neighbouring_words`: Η συνάρτηση αυτή βρίσκει από τη λίστα `list_words` που δίνονται σαν όρισμα τις λέξεις που είναι γειτονικές με τη λέξη που βρίσκεται στη θέση `i` που δίνεται σαν όρισμα. Από τις γειτονικές λέξεις αποκλείει αυτές που περιέχουν τη συμβολοσειρά `word` που δίνεται σαν όρισμα γιατί η λέξη `word` βρίσκεται στη θέση `i`.
- `find_date`: Η συνάρτηση αυτή βρίσκει την ημερομηνία του άρθρου με περιεχόμενο που σώθηκε στο αρχείο `file` που δίνουμε σαν όρισμα.
- `find_words_in_html_file`: Η συνάρτηση αυτή βρίσκει λέξεις σε αρχείο `html`. Μόλις βρει λέξη από τη λίστα `words` που δίνεται σαν όρισμα βρίσκει την ημερομηνία, τις γειτονικές λέξεις της λέξης που βρήκε και εκτελεί εξαγωγή πληροφοριών. Στη συνέχεια καταγράφει στο αρχείο με όνομα `filename2` που δίνεται σαν όρισμα τα ονόματα των `html` αρχείων που διαβάστηκαν.
- `main`: Η κύρια συνάρτηση του κώδικα. Στη συνάρτηση αυτή ορίζεται το όνομα του φακέλου που περιλαμβάνει τα αρχεία `html` που πρέπει να διαβαστούν, το όνομα του αρχείου που περιλαμβάνει τριάδες της μορφής ημερομηνία,τοποθεσία,κρούσματα και το όνομα του αρχείου στο οποίο γράφονται τα ονόματα των αρχείων `html` που διαβάστηκαν. Το τελευταίο γίνεται για να μη διαβαστούν ξανά αρχεία `html` που ήδη διαβάστηκαν σε περίπτωση που ο κώδικας εκτελεστεί εκ νέου. Υπάρχουν σχόλια σε κάποια τμήματα του κώδικα για να μη δημιουργούνται νέα αρχεία σε περίπτωση που ο κώδικας εκτελεστεί εκ νέου. Καλείται η συνάρτηση που βρίσκει λέξεις σε αρχείο `html`. Στη συνέχεια τυπώνεται το μήνυμα ότι η εξαγωγή πληροφοριών είναι επιτυχής.

3.5 Κώδικας που επεξεργάζεται αρχεία

Ανάμεσα στα στάδια Εξαγωγή Πληροφοριών και Οπτικοποίηση Δεδομένων γράφτηκε κώδικας που επεξεργάζεται αρχεία. Αυτό έγινε για να δημιουργηθούν καλύτερες οπτικοποιήσεις, οι οποίες θα εξηγηθούν στην επόμενη ενότητα. Τα αρχεία κώδικα που υπάρχουν στο πρόγραμμα που επεξεργάζεται αρχεία είναι τα παρακάτω:

`create_list_areas.py`

`main.py`

Ο κώδικας που είναι γραμμένος στο αρχείο `create_list_areas.py`

είναι ο κώδικας των Εικόνων 31-46.

```

1  # -*- coding: utf-8 -*-
2
3  import sys
4
5  def create_list_areas(locations):
6      reload(sys)
7      sys.setdefaultencoding("utf-8")
8      areas = []
9      for location in locations:
10         if (location == 'Εβρος'):
11             areas.append('Θράκη')
12         elif (location == 'Ροδόπη'):
13             areas.append('Θράκη')
14         elif (location == 'Εάνθη'):
15             areas.append('Θράκη')
16         elif (location == 'Καβάλα'):
17             areas.append('Μακεδονία')
18         elif (location == 'Δράμα'):
19             areas.append('Μακεδονία')
20         elif (location == 'Σέρρες'):
21             areas.append('Μακεδονία')
22         elif (location == 'Κιλκίς'):
23             areas.append('Μακεδονία')
24         elif (location == 'Θεσσαλονίκη'):
25             areas.append('Μακεδονία')
26         elif (location == 'Χαλκιδική'):
27             areas.append('Μακεδονία')
28         elif (location == 'Πιερία'):
29             areas.append('Μακεδονία')
30         elif (location == 'Ημαθία'):
31             areas.append('Μακεδονία')
32         elif (location == 'Πέλλα'):
33             areas.append('Μακεδονία')
34         elif (location == 'Φλώρινα'):
35             areas.append('Μακεδονία')
36         elif (location == 'Καστοριά'):
37             areas.append('Μακεδονία')
38         elif (location == 'Κοζάνη'):
39             areas.append('Μακεδονία')
40         elif (location == 'Γρεβενά'):
41             areas.append('Μακεδονία')
42         elif (location == 'Ιωάννινα'):
43             areas.append('Ηπειρος')
44         elif (location == 'Θεσπρωτία'):
45             areas.append('Ηπειρος')
46         elif (location == 'Πρέβεζα'):
47             areas.append('Ηπειρος')
48         elif (location == 'Αρτα'):
49             areas.append('Ηπειρος')

```

Εικόνα 31: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (1-49)

```

50     elif (location == 'Τρίκαλα'):
51         areas.append('Θεσσαλία')
52     elif (location == 'Καρδίτσα'):
53         areas.append('Θεσσαλία')
54     elif (location == 'Λάρισα'):
55         areas.append('Θεσσαλία')
56     elif (location == 'Μαγνησία'):
57         areas.append('Θεσσαλία')
58     elif (location == 'Εύβοια'):
59         areas.append('Στερεά Ελλάδα')
60     elif (location == 'Βοιωτία'):
61         areas.append('Στερεά Ελλάδα')
62     elif (location == 'Φθιώτιδα'):
63         areas.append('Στερεά Ελλάδα')
64     elif (location == 'Φωκίδα'):
65         areas.append('Στερεά Ελλάδα')
66     elif (location == 'Ευρυτανία'):
67         areas.append('Στερεά Ελλάδα')
68     elif (location == 'Αιτωλοακαρνανία'):
69         areas.append('Στερεά Ελλάδα')
70     elif (location == 'Αχαΐα'):
71         areas.append('Πελοπόννησος')
72     elif (location == 'Ηλεία'):
73         areas.append('Πελοπόννησος')
74     elif (location == 'Κέρκυρα'):
75         areas.append('Ιόνιο Πέλαγος')
76     elif (location == 'Λευκάδα'):
77         areas.append('Ιόνιο Πέλαγος')
78     elif (location == 'Κεφαλλονιά'):
79         areas.append('Ιόνιο Πέλαγος')
80     elif (location == 'Ζάκυνθος'):
81         areas.append('Ιόνιο Πέλαγος')
82     elif (location == 'Αττική'):
83         areas.append('Στερεά Ελλάδα')
84     elif (location == 'Κόρινθος'):
85         areas.append('Πελοπόννησος')
86     elif (location == 'Αργολίδα'):
87         areas.append('Πελοπόννησος')
88     elif (location == 'Αρκαδία'):
89         areas.append('Πελοπόννησος')
90     elif (location == 'Μεσσηνία'):
91         areas.append('Πελοπόννησος')
92     elif (location == 'Λακωνία'):
93         areas.append('Πελοπόννησος')

```

Εικόνα 32: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (50-93)

```

94     elif (location == 'Λήμνος'):
95         areas.append('Αιγαίο Πέλαγος')
96     elif (location == 'Χίος'):
97         areas.append('Αιγαίο Πέλαγος')
98     elif (location == 'Σάμος'):
99         areas.append('Αιγαίο Πέλαγος')
100    elif (location == 'Κυκλάδες'):
101        areas.append('Αιγαίο Πέλαγος')
102    elif (location == 'Δωδεκάνησα'):
103        areas.append('Αιγαίο Πέλαγος')
104    elif (location == 'Λασιθί'):
105        areas.append('Κρήτη')
106    elif (location == 'Ηράκλειο'):
107        areas.append('Κρήτη')
108    elif (location == 'Ρέθυμνο'):
109        areas.append('Κρήτη')
110    elif (location == 'Χανιά'):
111        areas.append('Κρήτη')
112    elif (location == 'Ορεστιάδα'):
113        areas.append('Θράκη')
114    elif (location == 'Διδυμότειχο'):
115        areas.append('Θράκη')
116    elif (location == 'Σουφλί'):
117        areas.append('Θράκη')
118    elif (location == 'Αλεξανδρούπολη'):
119        areas.append('Θράκη')
120    elif (location == 'Σαμοθράκη'):
121        areas.append('Θράκη')
122    elif (location == 'Αρριανά'):
123        areas.append('Θράκη')
124    elif (location == 'Μαρώνεια-Σάπες'):
125        areas.append('Θράκη')
126    elif (location == 'Κομοτηνή'):
127        areas.append('Θράκη')
128    elif (location == 'Ιασμος'):
129        areas.append('Θράκη')
130    elif (location == 'Αβδηρα'):
131        areas.append('Θράκη')
132    elif (location == 'Μύκη'):
133        areas.append('Θράκη')
134    elif (location == 'Τόπειρος'):
135        areas.append('Θράκη')
136    elif (location == 'Νέστος'):
137        areas.append('Μακεδονία')
138    elif (location == 'Παγγαίο'):
139        areas.append('Μακεδονία')

```

Εικόνα 33: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (94-139)

```

140     elif (location == 'Θάσος'):
141         areas.append('Μακεδονία')
142     elif (location == 'Δοξάτο'):
143         areas.append('Μακεδονία')
144     elif (location == 'Παρανέστι'):
145         areas.append('Μακεδονία')
146     elif (location == 'Προσοτσάνη'):
147         areas.append('Μακεδονία')
148     elif (location == 'Κάτω Νευροκόπι'):
149         areas.append('Μακεδονία')
150     elif (location == 'Νέα Ζίχνη'):
151         areas.append('Μακεδονία')
152     elif (location == 'Αμφίπολη'):
153         areas.append('Μακεδονία')
154     elif (location == 'Βισαλτία'):
155         areas.append('Μακεδονία')
156     elif (location == 'Εμμανουήλ Παπιάς'):
157         areas.append('Μακεδονία')
158     elif (location == 'Ηράκλεια'):
159         areas.append('Αιγαίο Πέλαγος')
160     elif (location == 'Σιντική'):
161         areas.append('Μακεδονία')
162     elif (location == 'Παιονία'):
163         areas.append('Μακεδονία')
164     elif (location == 'Χαλκηδόνα'):
165         areas.append('Μακεδονία')
166     elif (location == 'Δέλτα'):
167         areas.append('Μακεδονία')
168     elif (location == 'Ωραιόκαστρο'):
169         areas.append('Μακεδονία')
170     elif (location == 'Παύλος Μελάς'):
171         areas.append('Μακεδονία')
172     elif (location == 'Αμπελόκηποι-Μενεμένη'):
173         areas.append('Μακεδονία')
174     elif (location == 'Κορδελιό-Εύοσμος'):
175         areas.append('Μακεδονία')
176     elif (location == 'Νεάπολη-Συκιές'):
177         areas.append('Μακεδονία')
178     elif (location == 'Καλαμαριά'):
179         areas.append('Μακεδονία')
180     elif (location == 'Θερμαϊκός'):
181         areas.append('Μακεδονία')
182     elif (location == 'Θέρμη'):
183         areas.append('Μακεδονία')
184     elif (location == 'Πυλαία-Χορτιάτης'):
185         areas.append('Μακεδονία')

```

Εικόνα 34: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (140-185)

```

186         elif (location == 'Λαγκαδάς'):
187             areas.append('Μακεδονία')
188         elif (location == 'Βόλβη'):
189             areas.append('Μακεδονία')
190         elif (location == 'Αριστοτέλης'):
191             areas.append('Μακεδονία')
192         elif (location == 'Σιθωνία'):
193             areas.append('Μακεδονία')
194         elif (location == 'Πολύγυρος'):
195             areas.append('Μακεδονία')
196         elif (location == 'Κασσάνδρα'):
197             areas.append('Μακεδονία')
198         elif (location == 'Νέα Προποντίδα'):
199             areas.append('Μακεδονία')
200         elif (location == 'Διόν Όλυμπος'):
201             areas.append('Μακεδονία')
202         elif (location == 'Κατερίνη'):
203             areas.append('Μακεδονία')
204         elif (location == 'Πύδνα-Κολινδρός'):
205             areas.append('Μακεδονία')
206         elif (location == 'Αλεξάνδρεια'):
207             areas.append('Μακεδονία')
208         elif (location == 'Βέροια'):
209             areas.append('Μακεδονία')
210         elif (location == 'Νάουσα'):
211             areas.append('Μακεδονία')
212         elif (location == 'Σκύδρα'):
213             areas.append('Μακεδονία')
214         elif (location == 'Έδεσσα'):
215             areas.append('Μακεδονία')
216         elif (location == 'Αλμωπία'):
217             areas.append('Μακεδονία')
218         elif (location == 'Αμύνταιο'):
219             areas.append('Μακεδονία')
220         elif (location == 'Πρέσπες'):
221             areas.append('Μακεδονία')
222         elif (location == 'Νεστόριο'):
223             areas.append('Μακεδονία')
224         elif (location == 'Ορεστίδα'):
225             areas.append('Μακεδονία')
226         elif (location == 'Εορδαία'):
227             areas.append('Μακεδονία')
228         elif (location == 'Βόιο'):
229             areas.append('Μακεδονία')

```

Εικόνα 35: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (186-229)

```

230     elif (location == 'Σέρβια-Βελβεντός'):
231         areas.append('Μακεδονία')
232     elif (location == 'Δεσκάτη'):
233         areas.append('Μακεδονία')
234     elif (location == 'Κόνιτσα'):
235         areas.append('Ηπειρος')
236     elif (location == 'Πωγώνι'):
237         areas.append('Ηπειρος')
238     elif (location == 'Ζαγόρι'):
239         areas.append('Ηπειρος')
240     elif (location == 'Ζίτσα'):
241         areas.append('Ηπειρος')
242     elif (location == 'Μέτσοβο'):
243         areas.append('Ηπειρος')
244     elif (location == 'Βόρεια Τζουμέρκα'):
245         areas.append('Ηπειρος')
246     elif (location == 'Δωδώνη'):
247         areas.append('Ηπειρος')
248     elif (location == 'Σούλι'):
249         areas.append('Ηπειρος')
250     elif (location == 'Φιλιάτες'):
251         areas.append('Ηπειρος')
252     elif (location == 'Ηγουμενίτσα'):
253         areas.append('Ηπειρος')
254     elif (location == 'Πάργα'):
255         areas.append('Ηπειρος')
256     elif (location == 'Ζήρος'):
257         areas.append('Ηπειρος')
258     elif (location == 'Νικόλαος Σκουφάς'):
259         areas.append('Ηπειρος')
260     elif (location == 'Γεώργιος Καραϊσκάκης'):
261         areas.append('Ηπειρος')
262     elif (location == 'Κεντρικά Τζουμέρκα'):
263         areas.append('Ηπειρος')
264     elif (location == 'Πύλη'):
265         areas.append('Θεσσαλία')
266     elif (location == 'Μετέωρα'):
267         areas.append('Θεσσαλία')
268     elif (location == 'Φαρκαδόνα'):
269         areas.append('Θεσσαλία')
270     elif (location == 'Παλαμάς'):
271         areas.append('Θεσσαλία')
272     elif (location == 'Μουζάκι'):
273         areas.append('Θεσσαλία')
274     elif (location == 'Σοφάδες'):

```

Εικόνα 36: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (230-274)

```

275         areas.append('Θεσσαλία')
276     elif (location == 'Αργιθέα'):
277         areas.append('Θεσσαλία')
278     elif (location == 'Λίμνη Πλαστήρα'):
279         areas.append('Θεσσαλία')
280     elif (location == 'Ελασσόνα'):
281         areas.append('Θεσσαλία')
282     elif (location == 'Τύρναβος'):
283         areas.append('Θεσσαλία')
284     elif (location == 'Τέμπη'):
285         areas.append('Θεσσαλία')
286     elif (location == 'Αγιά'):
287         areas.append('Θεσσαλία')
288     elif (location == 'Κιλελέρ'):
289         areas.append('Θεσσαλία')
290     elif (location == 'Φάρσαλα'):
291         areas.append('Θεσσαλία')
292     elif (location == 'Ρήγας Φεραίος'):
293         areas.append('Θεσσαλία')
294     elif (location == 'Αλμυρός'):
295         areas.append('Θεσσαλία')
296     elif (location == 'Βόλος'):
297         areas.append('Θεσσαλία')
298     elif (location == 'Ζαγορά-Μούρεσι'):
299         areas.append('Θεσσαλία')
300     elif (location == 'Νότιο Πήλιο'):
301         areas.append('Θεσσαλία')
302     elif (location == 'Σκιάθος'):
303         areas.append('Θεσσαλία')
304     elif (location == 'Σκόπελος'):
305         areas.append('Θεσσαλία')
306     elif (location == 'Αλόνησος'):
307         areas.append('Θεσσαλία')
308     elif (location == 'Σκύρος'):
309         areas.append('Στερεά Ελλάδα')
310     elif (location == 'Ιστιαία - Αιδηψός'):
311         areas.append('Στερεά Ελλάδα')
312     elif (location == 'Μοντιούδι-Λίμνη-Αγία Άννα'):
313         areas.append('Στερεά Ελλάδα')
314     elif (location == 'Δίρφυες-Μεσσάπια'):
315         areas.append('Στερεά Ελλάδα')
316     elif (location == 'Χαλκίδα'):
317         areas.append('Στερεά Ελλάδα')
318     elif (location == 'Ερέτρια'):
319         areas.append('Στερεά Ελλάδα')

```

Εικόνα 37: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (275-319)

```

320         elif (location == 'Κύμη-Αλιβέρι'):
321             areas.append('Στερεά Ελλάδα')
322         elif (location == 'Κάρυστος'):
323             areas.append('Στερεά Ελλάδα')
324         elif (location == 'Τανάγρα'):
325             areas.append('Στερεά Ελλάδα')
326         elif (location == 'Θήβα'):
327             areas.append('Στερεά Ελλάδα')
328         elif (location == 'Αλίσαρτος'):
329             areas.append('Στερεά Ελλάδα')
330         elif (location == 'Ορχομενός'):
331             areas.append('Στερεά Ελλάδα')
332         elif (location == 'Λειβαδιά'):
333             areas.append('Στερεά Ελλάδα')
334         elif (location == 'Δίστομο-Αράχοβα-Αντίκυρα'):
335             areas.append('Στερεά Ελλάδα')
336         elif (location == 'Λοκρές'):
337             areas.append('Στερεά Ελλάδα')
338         elif (location == 'Αμφίκλεια-Ελάτεια'):
339             areas.append('Στερεά Ελλάδα')
340         elif (location == 'Μώλος-Άγιος Κωνσταντίνος'):
341             areas.append('Στερεά Ελλάδα')
342         elif (location == 'Λαμία'):
343             areas.append('Στερεά Ελλάδα')
344         elif (location == 'Στυλίδα'):
345             areas.append('Στερεά Ελλάδα')
346         elif (location == 'Δομοκός'):
347             areas.append('Στερεά Ελλάδα')
348         elif (location == 'Δελφοί'):
349             areas.append('Στερεά Ελλάδα')
350         elif (location == 'Δωρίδα'):
351             areas.append('Στερεά Ελλάδα')
352         elif (location == 'Καρπενήσι'):
353             areas.append('Στερεά Ελλάδα')
354         elif (location == 'Άγραφα'):
355             areas.append('Στερεά Ελλάδα')
356         elif (location == 'Αμφιλοχία'):
357             areas.append('Στερεά Ελλάδα')
358         elif (location == 'Ακτιο-Βόνιτσα'):
359             areas.append('Στερεά Ελλάδα')
360         elif (location == 'Ξηρόμερο'):
361             areas.append('Στερεά Ελλάδα')
362         elif (location == 'Μεσολόγγι'):
363             areas.append('Στερεά Ελλάδα')
364         elif (location == 'Αγρίνιο'):
365             areas.append('Στερεά Ελλάδα')

```

Εικόνα 38: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (320-365)

```

366         elif (location == 'Θέρμο'):
367             areas.append('Στερεά Ελλάδα')
368         elif (location == 'Ναύπακτος'):
369             areas.append('Στερεά Ελλάδα')
370         elif (location == 'Πάτρα'):
371             areas.append('Πελοπόννησος')
372         elif (location == 'Αιγιάλεια'):
373             areas.append('Πελοπόννησος')
374         elif (location == 'Καλάβρυτα'):
375             areas.append('Πελοπόννησος')
376         elif (location == 'Ερύμανθος'):
377             areas.append('Πελοπόννησος')
378         elif (location == 'Δυτική Αχαΐα'):
379             areas.append('Πελοπόννησος')
380         elif (location == 'Ανδραβίδα - Κυλλήνη'):
381             areas.append('Πελοπόννησος')
382         elif (location == 'Πηνειός'):
383             areas.append('Πελοπόννησος')
384         elif (location == 'Ηλιδα'):
385             areas.append('Πελοπόννησος')
386         elif (location == 'Πύργος'):
387             areas.append('Πελοπόννησος')
388         elif (location == 'Αρχαία Ολυμπία'):
389             areas.append('Πελοπόννησος')
390         elif (location == 'Ανδρίτσαινα - Κρέστενα'):
391             areas.append('Πελοπόννησος')
392         elif (location == 'Ζαχάρω'):
393             areas.append('Πελοπόννησος')
394         elif (location == 'Παξοί'):
395             areas.append('Ιόνιο Πέλαγος')
396         elif (location == 'Μεγανήσι'):
397             areas.append('Ιόνιο Πέλαγος')
398         elif (location == 'Ιθάκη'):
399             areas.append('Ιόνιο Πέλαγος')
400         elif (location == 'Κύθηρα'):
401             areas.append('Ιόνιο Πέλαγος')
402         elif (location == 'Σπέτσες'):
403             areas.append('Στερεά Ελλάδα')
404         elif (location == 'Υδρα'):
405             areas.append('Στερεά Ελλάδα')
406         elif (location == 'Τροιζηνία'):
407             areas.append('Στερεά Ελλάδα')
408         elif (location == 'Πόρος'):
409             areas.append('Στερεά Ελλάδα')
410         elif (location == 'Αγκίστρι'):
411             areas.append('Στερεά Ελλάδα')

```

Εικόνα 39: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (366-411)

```

412     elif (location == 'Αίγινα'):
413         areas.append('Στερεά Ελλάδα')
414     elif (location == 'Σαλαμίνα'):
415         areas.append('Στερεά Ελλάδα')
416     elif (location == 'Πέραμα'):
417         areas.append('Στερεά Ελλάδα')
418     elif (location == 'Κερατσίνι - Δραπετσώνα'):
419         areas.append('Στερεά Ελλάδα')
420     elif (location == 'Πειραιάς'):
421         areas.append('Στερεά Ελλάδα')
422     elif (location == 'Νίκαια - Άγιος Ιωάννης Ρέντης'):
423         areas.append('Στερεά Ελλάδα')
424     elif (location == 'Κορυδαλλός'):
425         areas.append('Στερεά Ελλάδα')
426     elif (location == 'Αγία Βαρβάρα'):
427         areas.append('Στερεά Ελλάδα')
428     elif (location == 'Χαϊδάρι'):
429         areas.append('Στερεά Ελλάδα')
430     elif (location == 'Αιγάλεω'):
431         areas.append('Στερεά Ελλάδα')
432     elif (location == 'Περιστέρι'):
433         areas.append('Στερεά Ελλάδα')
434     elif (location == 'Πειρούπολη'):
435         areas.append('Στερεά Ελλάδα')
436     elif (location == 'Ιλίου'):
437         areas.append('Στερεά Ελλάδα')
438     elif (location == 'Άγιοι Ανάργυροι - Καματερό'):
439         areas.append('Στερεά Ελλάδα')
440     elif (location == 'Φιλαδέλφεια - Χαλκηδόνα'):
441         areas.append('Στερεά Ελλάδα')
442     elif (location == 'Γαλάτσι'):
443         areas.append('Στερεά Ελλάδα')
444     elif (location == 'Αθήνα'):
445         areas.append('Στερεά Ελλάδα')
446     elif (location == 'Δάφνη - Υμηττός'):
447         areas.append('Στερεά Ελλάδα')
448     elif (location == 'Ηλιούπολη'):
449         areas.append('Στερεά Ελλάδα')
450     elif (location == 'Βύρωνα'):
451         areas.append('Στερεά Ελλάδα')
452     elif (location == 'Καισαριανή'):
453         areas.append('Στερεά Ελλάδα')
454     elif (location == 'Ζωγράφου'):
455         areas.append('Στερεά Ελλάδα')

```

Εικόνα 40: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (412-455)

```

456 elif (location == 'Παπάγου-Χολαργός'):
457     areas.append('Στερεά Ελλάδα')
458 elif (location == 'Αγία Παρασκευή'):
459     areas.append('Στερεά Ελλάδα')
460 elif (location == 'Χαλάνδρι'):
461     areas.append('Στερεά Ελλάδα')
462 elif (location == 'Βριλήσσια'):
463     areas.append('Στερεά Ελλάδα')
464 elif (location == 'Μαρούσι'):
465     areas.append('Στερεά Ελλάδα')
466 elif (location == 'Φιλοθέη-Ψυχικό'):
467     areas.append('Στερεά Ελλάδα')
468 elif (location == 'Νέα Ιωνία'):
469     areas.append('Στερεά Ελλάδα')
470 elif (location == 'Μεταμόρφωση'):
471     areas.append('Στερεά Ελλάδα')
472 elif (location == 'Λυκόβруση-Πεύκη'):
473     areas.append('Στερεά Ελλάδα')
474 elif (location == 'Κηφισιά'):
475     areas.append('Στερεά Ελλάδα')
476 elif (location == 'Πεντέλη'):
477     areas.append('Στερεά Ελλάδα')
478 elif (location == 'Μοσχάτο-Ταύρος'):
479     areas.append('Στερεά Ελλάδα')
480 elif (location == 'Καλλιθέα'):
481     areas.append('Στερεά Ελλάδα')
482 elif (location == 'Νέα Σμύρνη'):
483     areas.append('Στερεά Ελλάδα')
484 elif (location == 'Παλαιό Φάληρο'):
485     areas.append('Στερεά Ελλάδα')
486 elif (location == 'Άγιος Δημήτριος'):
487     areas.append('Στερεά Ελλάδα')
488 elif (location == 'Άλιμος'):
489     areas.append('Στερεά Ελλάδα')
490 elif (location == 'Ελληνικό-Αργυρούπολη'):
491     areas.append('Στερεά Ελλάδα')
492 elif (location == 'Γλυφάδα'):
493     areas.append('Στερεά Ελλάδα')
494 elif (location == 'Βάρη - Βούλα - Βουλιαγμένη'):
495     areas.append('Στερεά Ελλάδα')
496 elif (location == 'Κρωπία'):
497     areas.append('Στερεά Ελλάδα')
498 elif (location == 'Σαρωνικός'):
499     areas.append('Στερεά Ελλάδα')
500 elif (location == 'Λαυρεωτική'):
501     areas.append('Στερεά Ελλάδα')

```

Εικόνα 41: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (456-501)

```

502     elif (location == 'Μαρκόπουλο Μεσογαίας'):
503         areas.append('Στερεά Ελλάδα')
504     elif (location == 'Παιονία'):
505         areas.append('Στερεά Ελλάδα')
506     elif (location == 'Σπάτα-Αρτέμιδα'):
507         areas.append('Στερεά Ελλάδα')
508     elif (location == 'Παλλήνη'):
509         areas.append('Στερεά Ελλάδα')
510     elif (location == 'Ραφήνα-Πικέρμι'):
511         areas.append('Στερεά Ελλάδα')
512     elif (location == 'Μαραθώνας'):
513         areas.append('Στερεά Ελλάδα')
514     elif (location == 'Διόνυσος'):
515         areas.append('Στερεά Ελλάδα')
516     elif (location == 'Ωρωπός'):
517         areas.append('Στερεά Ελλάδα')
518     elif (location == 'Αχαρνές'):
519         areas.append('Στερεά Ελλάδα')
520     elif (location == 'Φυλή'):
521         areas.append('Στερεά Ελλάδα')
522     elif (location == 'Ασπρόπυργος'):
523         areas.append('Στερεά Ελλάδα')
524     elif (location == 'Ελευσίνα'):
525         areas.append('Στερεά Ελλάδα')
526     elif (location == 'Μάνδρα-Ειδυλλία'):
527         areas.append('Στερεά Ελλάδα')
528     elif (location == 'Μέγαρα'):
529         areas.append('Στερεά Ελλάδα')
530     elif (location == 'Λουτράκι-Άγιοι Θεόδωροι'):
531         areas.append('Στερεά Ελλάδα')
532     elif (location == 'Βέλο-Βόχα'):
533         areas.append('Πελοπόννησος')
534     elif (location == 'Σικωνίες'):
535         areas.append('Πελοπόννησος')
536     elif (location == 'Ευλόκαστρο-Ευρωστίνη'):
537         areas.append('Πελοπόννησος')
538     elif (location == 'Νεμέα'):
539         areas.append('Πελοπόννησος')
540     elif (location == 'Άργος - Μυκήνες'):
541         areas.append('Πελοπόννησος')
542     elif (location == 'Ναύπλιο'):
543         areas.append('Πελοπόννησος')
544     elif (location == 'Επίδαυρος'):
545         areas.append('Πελοπόννησος')

```

Εικόνα 42: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (502-545)

```

546 elif (location == 'Ερμιονίδα'):
547     areas.append('Πελοπόννησος')
548 elif (location == 'Νότια Κυνουρία'):
549     areas.append('Πελοπόννησος')
550 elif (location == 'Βόρεια Κυνουρία'):
551     areas.append('Πελοπόννησος')
552 elif (location == 'Τρίπολη'):
553     areas.append('Πελοπόννησος')
554 elif (location == 'Γορτυνία'):
555     areas.append('Πελοπόννησος')
556 elif (location == 'Μεγαλόπολη'):
557     areas.append('Πελοπόννησος')
558 elif (location == 'Οιχαλία'):
559     areas.append('Θεσσαλία')
560 elif (location == 'Τριφυλία'):
561     areas.append('Πελοπόννησος')
562 elif (location == 'Πύλος Νέστωρ'):
563     areas.append('Πελοπόννησος')
564 elif (location == 'Μεσσήνη'):
565     areas.append('Πελοπόννησος')
566 elif (location == 'Καλαμάτα'):
567     areas.append('Πελοπόννησος')
568 elif (location == 'Δυτική Μάνη'):
569     areas.append('Πελοπόννησος')
570 elif (location == 'Ανατολική Μάνη'):
571     areas.append('Πελοπόννησος')
572 elif (location == 'Σπάρτη'):
573     areas.append('Πελοπόννησος')
574 elif (location == 'Ευρώτας'):
575     areas.append('Πελοπόννησος')
576 elif (location == 'Μονεμβασία'):
577     areas.append('Πελοπόννησος')
578 elif (location == 'Ελαφώνησος'):
579     areas.append('Πελοπόννησος')
580 elif (location == 'Άγιος Ευστράτιος'):
581     areas.append('Αιγαίο Πέλαγος')
582 elif (location == 'Λέσβος'):
583     areas.append('Αιγαίο Πέλαγος')
584 elif (location == 'Ψαρά'):
585     areas.append('Αιγαίο Πέλαγος')
586 elif (location == 'Οινούσσες'):
587     areas.append('Αιγαίο Πέλαγος')
588 elif (location == 'Φοῦρνοι Κορσεών'):
589     areas.append('Αιγαίο Πέλαγος')
590 elif (location == 'Ικαρία'):
591     areas.append('Αιγαίο Πέλαγος')

```

Εικόνα 43: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (546-591)

```

592 elif (location == 'Άνδρος'):
593     areas.append('Αιγαίο Πέλαγος')
594 elif (location == 'Τήνος'):
595     areas.append('Αιγαίο Πέλαγος')
596 elif (location == 'Κέα'):
597     areas.append('Αιγαίο Πέλαγος')
598 elif (location == 'Κύθνος'):
599     areas.append('Αιγαίο Πέλαγος')
600 elif (location == 'Σύρος - Ερμούπολη'):
601     areas.append('Αιγαίο Πέλαγος')
602 elif (location == 'Μύκονος'):
603     areas.append('Αιγαίο Πέλαγος')
604 elif (location == 'Σέριφος'):
605     areas.append('Αιγαίο Πέλαγος')
606 elif (location == 'Σίφνος'):
607     areas.append('Αιγαίο Πέλαγος')
608 elif (location == 'Κίμωλος'):
609     areas.append('Αιγαίο Πέλαγος')
610 elif (location == 'Μήλος'):
611     areas.append('Αιγαίο Πέλαγος')
612 elif (location == 'Αντίπαρος'):
613     areas.append('Αιγαίο Πέλαγος')
614 elif (location == 'Πάρος'):
615     areas.append('Αιγαίο Πέλαγος')
616 elif (location == 'Νάξος & Μικρές Κυκλάδες'):
617     areas.append('Αιγαίο Πέλαγος')
618 elif (location == 'Αμοργός'):
619     areas.append('Αιγαίο Πέλαγος')
620 elif (location == 'Τορ'):
621     areas.append('Αιγαίο Πέλαγος')
622 elif (location == 'Σίκινο'):
623     areas.append('Αιγαίο Πέλαγος')
624 elif (location == 'Φολέγανδρος'):
625     areas.append('Αιγαίο Πέλαγος')
626 elif (location == 'Σαντορίνη'):
627     areas.append('Αιγαίο Πέλαγος')
628 elif (location == 'Ανάφη'):
629     areas.append('Αιγαίο Πέλαγος')
630 elif (location == 'Αγαθονήσι'):
631     areas.append('Αιγαίο Πέλαγος')
632 elif (location == 'Πάτμος'):
633     areas.append('Αιγαίο Πέλαγος')
634 elif (location == 'Λειψοί'):
635     areas.append('Αιγαίο Πέλαγος')
636 elif (location == 'Λέρος'):
637     areas.append('Αιγαίο Πέλαγος')

```

Εικόνα 44: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (592-637)

```

638         elif (location == 'Κάλυμνος'):
639             areas.append('Αιγαίο Πέλαγος')
640         elif (location == 'Αστυπάλαια'):
641             areas.append('Αιγαίο Πέλαγος')
642         elif (location == 'Κως'):
643             areas.append('Αιγαίο Πέλαγος')
644         elif (location == 'Νίσυρος'):
645             areas.append('Αιγαίο Πέλαγος')
646         elif (location == 'Τήλος'):
647             areas.append('Αιγαίο Πέλαγος')
648         elif (location == 'Σύμη'):
649             areas.append('Αιγαίο Πέλαγος')
650         elif (location == 'Ρόδος'):
651             areas.append('Αιγαίο Πέλαγος')
652         elif (location == 'Χάλκη'):
653             areas.append('Αιγαίο Πέλαγος')
654         elif (location == 'Μεγίστη'):
655             areas.append('Αιγαίο Πέλαγος')
656         elif (location == 'Κάρπαθος'):
657             areas.append('Αιγαίο Πέλαγος')
658         elif (location == 'Κάσος'):
659             areas.append('Αιγαίο Πέλαγος')
660         elif (location == 'Σητεία'):
661             areas.append('Κρήτη')
662         elif (location == 'Ιεράπετρα'):
663             areas.append('Κρήτη')
664         elif (location == 'Άγιος Νικόλαος'):
665             areas.append('Κρήτη')
666         elif (location == 'Οροπέδιο Λασιθίου'):
667             areas.append('Κρήτη')
668         elif (location == 'Βιάννος'):
669             areas.append('Κρήτη')
670         elif (location == 'Μινώα Πεδιάδα'):
671             areas.append('Κρήτη')
672         elif (location == 'Αρχανές - Αστερούσια'):
673             areas.append('Κρήτη')
674         elif (location == 'Χερσόνησος'):
675             areas.append('Κρήτη')
676         elif (location == 'Μαλεβίζι'):
677             areas.append('Κρήτη')
678         elif (location == 'Γόρτυνα'):
679             areas.append('Κρήτη')
680         elif (location == 'Φαιστός'):
681             areas.append('Κρήτη')

```

Εικόνα 45: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (638-681)

```

682         elif (location == 'Ανώγεια'):
683             areas.append('Κρήτη')
684         elif (location == 'Μυλοποτάμος'):
685             areas.append('Κρήτη')
686         elif (location == 'Αμάρι'):
687             areas.append('Κρήτη')
688         elif (location == 'Άγιος Βασίλειος'):
689             areas.append('Κρήτη')
690         elif (location == 'Αποκόρωνος'):
691             areas.append('Κρήτη')
692         elif (location == 'Πλατανιάς'):
693             areas.append('Κρήτη')
694         elif (location == 'Κίσσαμος'):
695             areas.append('Κρήτη')
696         elif (location == 'Κάντιανος Σέλινος'):
697             areas.append('Κρήτη')
698         elif (location == 'Σφακιά'):
699             areas.append('Κρήτη')
700         elif (location == 'Γαύδος'):
701             areas.append('Κρήτη')
702     return areas

```

Εικόνα 46: Κώδικας που δημιουργεί λίστα γεωγραφικών διαμερισμάτων (682-702)

Η συνάρτηση `create_list_areas` δημιουργεί λίστα γεωγραφικών διαμερισμάτων. Ομαδοποιεί τις περιοχές της Ελλάδας σε γεωγραφικά διαμερίσματα.

Ο κώδικας που είναι γραμμένος στο αρχείο `main.py`

είναι ο κώδικας της Εικόνας 47.

```

1  import pandas as pd
2  import create_list_areas
3  import sys
4
5  def main():
6      reload(sys)
7      sys.setdefaultencoding("utf-8")
8      pathname = "Datasets/COVID-19_Cases_Greece.csv"
9      df = pd.read_csv(pathname, sep=',', encoding="utf-8", engine='python')
10     df['Area'] = create_list_areas.create_list_areas(list(df['Location']))
11     pathname2 = "Datasets/new_COVID-19_Cases_Greece.csv"
12     df.to_csv(pathname2, index=False, encoding="utf-8")
13     print "The field Area was added successfully."
14
15 main()

```

Εικόνα 47: Κώδικας που επεξεργάζεται αρχεία

Η συνάρτηση `main` επεξεργάζεται αρχεία. Ομαδοποιεί τις περιοχές της Ελλάδας σε γεωγραφικά διαμερίσματα της Ελλάδας. Διαβάζει το αρχείο με τις τριάδες της μορφής ημερομηνία, τοποθεσία, κρούσματα και βάζει τις τετράδες της μορφής ημερομηνία, τοποθεσία,

κρούσματα, διαμέρισμα σε καινούριο αρχείο. Στη συνέχεια τυπώνει μήνυμα ότι το πεδίο διαμέρισμα προστέθηκε επιτυχώς.

3.6 Κώδικας που εκτελεί οπτικοποίηση δεδομένων

Ο κώδικας που εκτελεί οπτικοποίηση δεδομένων είναι ο κώδικας των Εικόνων 48, 49 και 50.

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 from ipywidgets import *
4 import sys
5
6 def create_visualizations(filename):
7     df = pd.read_csv(filename, sep=',', encoding="utf-8", engine='python')
8     df["Year"] = df.Date.str.split('-', expand = True)[0]
9     df["Month"] = df.Date.str.split('-', expand = True)[1]
10    df["Day"] = df.Date.str.split('-', expand = True)[2]
11    df['Year_Int'] = df['Year'].fillna(0).astype(int)
12    df = df.loc[df['Year_Int'] >= 2020]
13    df = df.groupby(['Month', 'Year', 'Area'])['Cases'].sum().reset_index(name='Cases')
14    df['Month_Int'] = df['Month'].fillna(0).astype(int)
15    min_month = df['Month_Int'].min()
16    max_month = df['Month_Int'].max()
17    months = [min_month, max_month]
18    areas = [u'Όλα'] + list(df['Area'].unique())
19    years = [u'Όλα'] + list(df['Year'].unique())
20    df2 = df.groupby(['Month', 'Year'])['Cases'].sum().reset_index(name='Cases')
21    df2['Month_Int'] = df2['Month'].fillna(0).astype(int)
22    df3 = df.groupby(['Year', 'Area'])['Cases'].sum().reset_index(name='Cases')
23    df4 = df.groupby(['Year'])['Cases'].sum().reset_index(name='Cases')
24    df5 = df.groupby(['Month', 'Area'])['Cases'].sum().reset_index(name='Cases')
25    df5['Month_Int'] = df5['Month'].fillna(0).astype(int)
26    df6 = df.groupby(['Month'])['Cases'].sum().reset_index(name='Cases')
27    df6['Month_Int'] = df6['Month'].fillna(0).astype(int)
28
29    def plot_bar(month, year):
30        data = df
31        data2 = df5
32        if (year != u'Όλα'):
33            df_filtered = data.loc[(data['Month_Int'] == month) & (data['Year'] == year)]
34            plt.figure(1, figsize=(14, 6))
35            plt.bar(df_filtered['Area'], df_filtered['Cases'])
36            plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Γεωγραφικό Διαμέρισμα", fontsize=16)
37            plt.xlabel(u"Γεωγραφικό Διαμέρισμα", fontsize=16)
38            plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)
39        else:
40            df_filtered = data2.loc[data2['Month_Int'] == month]
41            plt.figure(1, figsize=(14, 6))
42            plt.bar(df_filtered['Area'], df_filtered['Cases'])
43            plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Γεωγραφικό Διαμέρισμα", fontsize=16)
44            plt.xlabel(u"Γεωγραφικό Διαμέρισμα", fontsize=16)
45            plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)
46
47    def plot_line(area, months, year):
48        data = df
49        data2 = df2

```

Εικόνα 48: Κώδικας που εκτελεί οπτικοποίηση δεδομένων (1-49)

```

50     data3 = df5
51     data4 = df6
52     if (area != u'Όλα' and year != u'Όλα'):
53         df_filtered = data.loc[(data['Area'] == area) & (data['Year'] == year) & (data['Month_Int'] >= months[0]) &
54         (data['Month_Int'] <= months[1])]
55         plt.figure(1, figsize=(14, 6))
56         plt.plot(df_filtered['Month'], df_filtered['Cases'])
57         plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Μήνα", fontsize=16)
58         plt.xlabel(u"Μήνας", fontsize=16)
59         plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)
60     elif (area == u'Όλα' and year != u'Όλα'):
61         df_filtered = data2.loc[(data2['Year'] == year) & (data2['Month_Int'] >= months[0]) & (data2['Month_Int'] <= months[1])]
62         plt.figure(1, figsize=(14, 6))
63         plt.plot(df_filtered['Month'], df_filtered['Cases'])
64         plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Μήνα", fontsize=16)
65         plt.xlabel(u"Μήνας", fontsize=16)
66         plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)
67     elif (area != u'Όλα' and year == u'Όλα'):
68         df_filtered = data3.loc[(data3['Area'] == area) & (data3['Month_Int'] >= months[0]) & (data3['Month_Int'] <= months[1])]
69         plt.figure(1, figsize=(14, 6))
70         plt.plot(df_filtered['Month'], df_filtered['Cases'])
71         plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Μήνα", fontsize=16)
72         plt.xlabel(u"Μήνας", fontsize=16)
73         plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)
74     elif (area == u'Όλα' and year == u'Όλα'):
75         df_filtered = data4.loc[(data4['Month_Int'] >= months[0]) & (data4['Month_Int'] <= months[1])]
76         plt.figure(1, figsize=(14, 6))
77         plt.plot(df_filtered['Month'], df_filtered['Cases'])
78         plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Μήνα", fontsize=16)
79         plt.xlabel(u"Μήνας", fontsize=16)
80         plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)
81
82 def plot_area(area):
83     data = df3
84     data2 = df4
85     if (area != u'Όλα'):
86         df_filtered = data.loc[data['Area'] == area]
87         plt.figure(1, figsize=(14, 6))
88         plt.fill_between(list(df_filtered['Year']), list(df_filtered['Cases']),
89                         color="skyblue", alpha=0.4)
90         plt.plot(list(df_filtered['Year']), list(df_filtered['Cases']), color="Slateblue",
91                 alpha=0.6, linewidth=2)
92         plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Έτος", fontsize=16)
93         plt.xlabel(u"Έτος", fontsize=16)
94         plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)

```

Εικόνα 49: Κώδικας που εκτελεί οπτικοποίηση δεδομένων (50-93)

```

94     else:
95         df_filtered = data2
96         plt.figure(1, figsize=(14, 6))
97         plt.fill_between(list(df_filtered['Year']), list(df_filtered['Cases']),
98                         color="skyblue", alpha=0.4)
99         plt.plot(list(df_filtered['Year']), list(df_filtered['Cases']), color="Slateblue",
100                alpha=0.6, linewidth=2)
101         plt.title(u"Αθροισμα κρουσμάτων COVID-19 στην Ελλάδα ανά Έτος", fontsize=16)
102         plt.xlabel(u"Έτος", fontsize=16)
103         plt.ylabel(u"Κρούσματα COVID-19", fontsize=16)
104         month = widgets.IntSlider(value=min_month,min=min_month,max=max_month,step=1,description=u'Μήνας')
105         year = widgets.Dropdown(options=years, value=u'Όλα', description=u'Έτος')
106         area = widgets.Dropdown(options=areas,value=u'Όλα',description=u'Διαμέρισμα')
107         months = widgets.IntRangeSlider(value=months,min=min_month,max=max_month,step=1,description=u'Μήνες')
108
109         month.set_title = 'month'
110         year.set_title = 'year'
111         area.set_title = 'area'
112         months.set_title = 'months'
113
114         #assign widgets to tabs
115         tab1 = widgets.HBox([month, year])
116         tab2 = widgets.HBox([area, months, year])
117         tab3 = widgets.HBox([area])
118
119         #create tabs
120         tab_nest = widgets.Tab()
121         tab_nest.set_title(0, u'Κρούσματα ανά Διαμέρισμα')
122         tab_nest.set_title(1, u'Κρούσματα ανά Μήνα')
123         tab_nest.set_title(2, u'Κρούσματα ανά Έτος')
124
125         #interact function in isolation
126         f = interactive(plot_bar, month = month, year = year);
127         f2 = interactive(plot_line, area = area, months = months, year = year);
128         f3 = interactive(plot_area, area = area);
129         tab_nest.children = [VBox(children = f.children), VBox(children = f2.children), VBox(children = f3.children)]
130         display(tab_nest)
131
132     def main():
133         reload(sys)
134         sys.setdefaultencoding("utf-8")
135         pathname = "Datasets/new_COVID-19_Cases_Greece.csv"
136         create_visualizations(pathname)
137
138     main()

```

Εικόνα 50: Κώδικας που εκτελεί οπτικοποίηση δεδομένων (94-138)

Οι συναρτήσεις που υλοποιήθηκαν στον παραπάνω κώδικα εξηγούνται κάτωθι:

- **create_visualizations:** Η συνάρτηση αυτή επιτελεί τις ακόλουθες διεργασίες:
 1. Οπτικοποιεί τα δεδομένα του αρχείου με το όνομα που δίνεται σαν όρισμα με γράφημα ράβδων (bar chart).
 2. Οπτικοποιεί τα δεδομένα του αρχείου με το όνομα που δίνεται σαν όρισμα με γράφημα γραμμής (line chart).
 3. Οπτικοποιεί τα δεδομένα του αρχείου με το όνομα που δίνουμε σαν όρισμα με γράφημα περιοχής (area chart).
 4. Βάζει την κάθε οπτικοποίηση που δημιουργήσαμε σε καρτέλα (tab) για να κατανοούν οι χρήστες με ποια οπτικοποίηση αλληλεπιδρούν αν σύρουν μπάρα ή επιλέξουν από dropdown.
- **main:** Η κύρια συνάρτηση του κώδικα. Οπτικοποιεί τα δεδομένα του αρχείου με τις τετράδες της μορφής ημερομηνία, τοποθεσία, κρούσματα, διαμέρισμα. Το αρχείο αυτό δημιουργήθηκε με τον τρόπο που εξηγήθηκε στην προηγούμενη ενότητα για να δημιουργούνται καλύτερες οπτικοποιήσεις.

4. ΑΠΟΤΕΛΕΣΜΑΤΑ ΥΛΟΠΟΙΗΣΕΩΝ

Σε αυτό το κεφάλαιο θα αναφερθούν τα αποτελέσματα των υλοποιήσεων που αναλύθηκαν στο κεφάλαιο 3.

4.1 Αποτελέσματα του σταδίου Web Crawling

- in.gr:

Οι σελίδες που συλλέχθηκαν από τον ιστοχώρο in.gr είναι οι σελίδες της Εικόνας 51.



Εικόνα 51: Σελίδες που συλλέχθηκαν από τον ιστοχώρο in.gr

Από τον ιστοχώρο in.gr συλλέχθηκαν 101.102 σελίδες.

- tonima.gr:

Οι σελίδες που συλλέχθηκαν από τον ιστοχώρο tonima.gr είναι οι σελίδες της Εικόνας 52.

```

1 Page URL
2 https://www.tovima.gr/2020/10/22/opinions/arxi-tou-deyterou-kymatos-me-rekor-krousmaton/
3 https://www.tovima.gr/category/opinions/
4 https://www.tovima.gr/editor/a.Bulla
5 https://www.tovima.gr/2020/10/20/opinions/enas-akrivo-pliromenos-chef-stin-kouzina-tous/
6 https://www.tovima.gr/wp-content/uploads/2020/10/19/admin-ajax-1.jpg
7 https://www.tovima.gr/2020/10/22/politics/oruc-reis-anti-navtex-i-elliniki-apantisi-stin-neia-paranomi-tourkiki-navtex/
8 https://www.tovima.gr/category/politics/
9 https://www.tovima.gr/2020/10/22/politics/koronoios-pros-diaggelma-mitsotaki-kai-sto-vathos-neia-metra/
10 https://www.tovima.gr/wp-content/uploads/2020/10/22/1-5.jpg
11 https://www.tovima.gr/2020/10/22/society/o-koronoios-saronei-tin-ellada-neia-metra-pro-ton-pylon/
12 https://www.tovima.gr/category/society/
13 https://www.tovima.gr/2020/10/22/society/koronoios-speydi-sti-thessaloniki-o-xardalias-ektakti-syskepsi/
14 https://www.tovima.gr/wp-content/uploads/2020/10/22/XAPAAHIAE-MAKKA-e1600408829487.jpg
15 https://www.tovima.gr/tag/%e6%b8%e6%b5%cf%83%cf%83%e6%b1%ce%bb%ce%bf%ce%bd%ce%af%ce%ba%e6%b7/
16 https://www.tovima.gr/2020/10/21/society/koronoios-etasi-ksfygan-ta-kroumata-sti-thessaloniki/
17 https://www.tovima.gr/wp-content/uploads/2020/10/21/party.jpeg
18 https://www.tovima.gr/tag/koronoios/
19 https://www.tovima.gr/2020/10/22/society/dimitris-kampanaros-thrinou-gia-ton-idiotiti-tou-girokomeiou-to-teleytaio-minya/
20 https://www.tovima.gr/wp-content/uploads/2020/10/22/1-571.jpg
21 https://www.tovima.gr/tag/%e6%ac%e6%b3%ce%b9%ce%bf%cf%82-%cf%83%cf%84%ce%ad%cf%86%ce%bl%ce%bd%ce%bf%cf%82/
22 https://www.tovima.gr/2020/10/22/society/agios-stefanos-sygklonizei-i-istoria-tou-idiotiti-tou-girokomeiou-to-simeiona-aytoktonias/
23 https://www.tovima.gr/wp-content/uploads/2020/10/21/1-571.jpg
24 https://www.tovima.gr/tag/%e6%b1%cf%85%cf%84%ce%bf%ce%ba%cf%84%ce%bf%ce%bd%ce%af%ce%b1/
25 https://www.tovima.gr/2020/10/21/vimatv/agios-stefanos-to-sparaktiko-gramma-tou-idiotiti-tou-girokomeiou-pou-aytoktonise/
26 https://www.tovima.gr/category/vimatv/
27 https://www.tovima.gr/2020/10/22/vimatv/xrysi-aygi-en-anamoni-tis-apofasi-gia-tis-anastoles-ton-poinon/
28 https://www.tovima.gr/wp-content/uploads/2020/10/22/EVTL_DIKH_XRYSH_AYGH_APOFASH_7H_MERA_ANASTOLES-10_med-1-1-600x400-1.jpg
29 https://www.tovima.gr/tag/anastoles/
30 https://www.tovima.gr/2020/10/16/society/diki-xrysis-aygis-ti-deytera-i-apofasi-gia-tis-anastoles-ti-zitisan-oi-katadikastentes/
31 https://www.tovima.gr/wp-content/uploads/2020/10/14/diki_xrisiavgi.jpg
32 https://www.tovima.gr/tag/%e6%b1%cf%80%cf%8c%cf%86%ce%b1%cf%83%ce%b7/
33 https://www.tovima.gr/2020/10/15/society/dimoskopisi-oi-polites-gia-tin-katadiki-tis-xrysis-aygis/
34 https://www.tovima.gr/wp-content/uploads/2020/10/15/1-18.jpg
35 https://www.tovima.gr/tag/apofasi-x-a/
36 https://www.tovima.gr/2020/10/07/society/diki-xrysis-aygis-to-twitter-leei-me-mia-foni-tsakiste_tous_nazi/
37 https://www.tovima.gr/wp-content/uploads/2020/10/07/pavlosi-el602068174904.jpeg
38 https://www.tovima.gr/tag/twitter/
39 https://www.tovima.gr/2020/10/16/world/pos-to-facebook-kai-to-twitter-apeiloun-ti-dimokratia/
40 https://www.tovima.gr/category/world/
41 https://www.tovima.gr/2020/10/22/world/prokalei-o-oktai-tha-kanoume-ereynes-akoma-kai-notia-tis-kritis/
42 https://www.tovima.gr/wp-content/uploads/2020/10/22/Fuat-Oktay.jpg
43 https://www.tovima.gr/2020/10/21/politics/oruc-reis-neia-tourkiki-navtex-parateinetai-i-agonia-stin-an-mesogeio/
44 https://www.tovima.gr/wp-content/uploads/2020/10/12/Oruc-Reis-1.jpg
45 https://www.tovima.gr/tag/oruc-reis/
46 https://www.tovima.gr/2020/10/21/politics/ellinoturkika-pos-to-elliniko-naytiko-ksetrypose-ta-tourkika-upovryxia-kai-estise-teikos-sto-oruc-reis/
47 https://www.tovima.gr/wp-content/uploads/2020/10/21/5-600x450-1.jpg
48 https://www.tovima.gr/tag/ellinoturkika/
49 https://www.tovima.gr/2020/10/21/oktai-tha-kanoume-ereynes-akoma-kai-notia-tis-kritis/

```

Εικόνα 52: Σελίδες που συλλέχθηκαν από τον ιστοχώρο tonima.gr

Από τον ιστοχώρο τονima.gr συλλέχθηκαν 81.100 σελίδες.

- tanea.gr:
Οι σελίδες που συλλέχθηκαν από τον ιστοχώρο tanea.gr είναι οι σελίδες της Εικόνας 53.

1	Page URL
2	https://www.tanea.gr/2020/12/18/greece/koronaioi-agonas-dromou-gia-teixos-anosias-eos-ton-icounio/
3	https://www.tanea.gr/
4	https://www.tanea.gr/2020/12/18/greece/koronaioi-poi-es-perifereies-vrazoun-i-eikona-sta-nosokomeia/
5	https://www.tanea.gr/category/greece/
6	https://www.tanea.gr/2020/12/18/greece/koronaioi-katholiko-lockdown-se-elysina-ti-leei-o-dimarkos-tis-perioxis-sto-mega/
7	https://www.tanea.gr/wp-content/uploads/2020/12/DIM.jpg
8	https://www.tanea.gr/2020/12/18/people/o-ellinas-protestantis-vasilias/
9	https://www.tanea.gr/category/people/
10	https://www.tanea.gr/2020/12/15/people/aytos-einai-o-26xronos-ellinas-pou-vraveyei-o-oie-gia-to-perivallon/
11	https://www.tanea.gr/wp-content/uploads/2020/12/Lefteris_Arapakis_2020_UNEF_Young_Champions_of_the_Earth_winner_credit_UNEF.jpg
12	https://www.tanea.gr/2020/12/18/greece/ta-atoma-me-provlimata-orasis-vlepoun-ta-asteria/
13	https://www.tanea.gr/wp-content/uploads/2020/12/cofunded-full-1.jpg
14	https://www.tanea.gr/2020/12/18/market/ependyseis-stin-ygeia-enas-thlivera-anektalloytos-moxlos-oikonomikis-anaptyksis/
15	https://www.tanea.gr/category/market/
16	https://www.tanea.gr/2020/12/17/market/to-betshop-gr-stirizei-os-megalos-xorigos-ton-giannis-fejzullai-sta-ringk/
17	https://www.tanea.gr/wp-content/uploads/2020/12/1200x800_Giannis_Fejzullai.jpg
18	https://www.tanea.gr/2020/12/18/greece/koronaioi-proeidopoiisi-apo-kathigiti-mikroviologias-maskai-apostaseis-3-metron/
19	https://www.tanea.gr/wp-content/uploads/2020/12/Cristmas-1.jpg
20	https://www.tanea.gr/2020/12/18/politics/government/epixeirisi-eleytheria-syskepsi-ypo-ton-prothypourgo-gia-to-sxedio-envoliasmou-se-ekseliksi-askiseis-prosomioisis/
21	https://www.tanea.gr/category/politics/
22	https://www.tanea.gr/2020/12/17/politics/mitsotakis-i-pandimia-apaitei-synergasia-politikis-kai-epistimis/
23	https://www.tanea.gr/wp-content/uploads/2020/12/kmelus-3.jpeg
24	https://www.tanea.gr/print/2020/12/18/politics/i-germania-mprosta-crsti-meta-merkel-epoxi/
25	https://www.tanea.gr/printed/
26	https://www.tanea.gr/2020/12/18/inbox/to-skitso-tou-arka-gia-to-petrelaio-thermansis/
27	https://www.tanea.gr/category/inbox/
28	https://www.tanea.gr/2020/12/17/inbox/oaka-ksekinoun-ergasies-anakataskeyis-allagi-edras-gia-tin-aek-sto-mats-me-volo/
29	https://www.tanea.gr/wp-content/uploads/2020/12/5206737.jpg
30	https://www.tanea.gr/2020/12/17/sports/football/olympiakos-den-xanei-kontra-stin-aek-o-martins/
31	https://www.tanea.gr/category/sports/
32	https://www.tanea.gr/2020/12/17/sports/i-parakamera-tou-aek-olympiakos/
33	https://www.tanea.gr/wp-content/uploads/2020/12/3-186.jpg
34	https://www.tanea.gr/print/2020/12/18/sports/oristika-ektos-mountial-kai-olympiakon-agonon-oi-rosoi/
35	https://www.tanea.gr/print/2020/12/18/sports/ixiro-xastouki-stous-skeyorous/
36	https://www.tanea.gr/wp-content/uploads/2019/07/2178443-2.jpg
37	https://www.tanea.gr/print/2020/12/18/sports/tmikola-sti-meivourntmiki/
38	https://www.tanea.gr/print/2020/12/18/sports/oxi-alli-apoleia/
39	https://www.tanea.gr/print/2020/12/18/sports/i-lysi-gia-ti-rosia-den-apotelei-lysi-sto-provlima-tou-ntopingk/
40	https://www.tanea.gr/print/2020/12/18/sports/i-vradia-tou-leva/
41	https://www.tanea.gr/wp-content/uploads/2020/12/LEWA.jpg
42	https://www.tanea.gr/print/2020/12/18/sports/o-anthropos-ton-ntermpi/
43	https://www.tanea.gr/wp-content/uploads/2020/12/MAKENTA.jpg
44	https://www.tanea.gr/print/2020/12/18/sports/ekloges-stin-epo-apo-to-neo-etos/
45	https://www.tanea.gr/wp-content/uploads/2020/12/2020-12-17T145842Z_769176298_RC22FK9NWF8V_RTRMADP_5_SOCCER-ENGLAND-SUNDERLAND.jpg
46	https://www.tanea.gr/print/2020/12/18/sports/alli-mia-niki-gia-ti-giounaitent/
47	https://www.tanea.gr/wp-content/uploads/2020/12/viki.jpg
48	https://www.tanea.gr/print/2020/12/18/sports/nea-arnisi-tis-premier-ligk/
49	https://www.tanea.gr/print/2020/12/18/sports/kseapase-stin-almna/

Εικόνα 53: Σελίδες που συλλέχθηκαν από τον ιστοχώρο tanea.gr

Από τον ιστοχώρο tanea.gr συλλέχθηκαν 80.483 σελίδες.

- protothema.gr:

Οι σελίδες που συλλέχθηκαν από τον ιστοχώρο protothema.gr είναι οι σελίδες της Εικόνας 54.

1	Page URL
2	https://www.protothema.gr/oles-oi-eidiseis/
3	https://www.protothema.gr/sunday-editions/
4	https://www.protothema.gr/frontpages/
5	https://www.protothema.gr/rss
6	https://www.protothema.gr/greece/
7	https://www.protothema.gr/tag/metanastes/
8	https://www.protothema.gr/tag/rouvikonas/
9	https://www.protothema.gr/economy/
10	https://www.protothema.gr/tag/sudaxeis/
11	https://www.protothema.gr/politics/
12	https://www.protothema.gr/world/
13	https://www.protothema.gr/tag/donald-trab/
14	https://www.protothema.gr/sports/
15	https://www.protothema.gr/tag/champions-league/
16	https://www.protothema.gr/tag/europa-league/
17	https://www.protothema.gr/tag/kompi-mpraiant/
18	https://www.protothema.gr/car-and-speed/
19	https://www.protothema.gr/car-and-speed/news/
20	https://www.protothema.gr/car-and-speed/preview/
21	https://www.protothema.gr/car-and-speed/video/
22	https://www.protothema.gr/car-and-speed/tech/
23	https://www.protothema.gr/car-and-speed/racing/
24	https://www.protothema.gr/car-and-speed/formula-1/
25	https://www.protothema.gr/car-and-speed/pisokinisi/
26	https://www.protothema.gr/car-and-speed/newsauto-times-technika-charaktiristika/
27	https://www.protothema.gr/car-and-speed/moto/
28	https://www.protothema.gr/life-style/
29	https://www.protothema.gr/life-style/Hollywood/
30	https://www.protothema.gr/life-style/fashion/
31	https://www.protothema.gr/life-style/Gossip/
32	https://www.protothema.gr/sxeseis/
33	https://www.protothema.gr/tag/eurovision/
34	https://www.protothema.gr/culture/
35	https://www.protothema.gr/culture/theater/
36	https://www.protothema.gr/culture/cinema/
37	https://www.protothema.gr/culture/arts/
38	https://www.protothema.gr/culture/music/
39	https://www.protothema.gr/culture/books/
40	https://www.protothema.gr/technology/
41	https://www.protothema.gr/tag/facebook/
42	https://www.protothema.gr/environment/
43	https://www.protothema.gr/thema-insights/
44	https://www.protothema.gr/stories/
45	https://www.protothema.gr/travelling/
46	https://www.protothema.gr/recipes/
47	https://www.protothema.gr/world/afieromata/
48	https://www.protothema.gr/ugeia/
49	https://www.protothema.gr/zoi/
50	https://www.protothema.gr/naked-city/

Εικόνα 54: Σελίδες που συλλέχθηκαν από τον ιστοχώρο protothema.gr

Από τον ιστοχώρο protothema.gr συλλέχθηκαν 22.003 σελίδες.

4.2 Αποτελέσματα του σταδίου Αποθήκευση περιεχομένων σελίδων

- in.gr:
Τα αρχεία html στα οποία αποθηκεύτηκαν τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο in.gr είναι τα αρχεία html της Εικόνας 55.

Όνομα	Ημερομηνία τροποποι...	Τύπος	Μέγεθος
in.gr_content1	11/1/2021 16:06	Firefox HTML Doc...	114 KB
in.gr_content2	11/1/2021 16:06	Firefox HTML Doc...	113 KB
in.gr_content3	11/1/2021 16:06	Firefox HTML Doc...	109 KB
in.gr_content4	11/1/2021 16:06	Firefox HTML Doc...	64 KB
in.gr_content5	11/1/2021 16:06	Firefox HTML Doc...	104 KB
in.gr_content6	11/1/2021 16:06	Firefox HTML Doc...	104 KB
in.gr_content7	11/1/2021 16:06	Firefox HTML Doc...	105 KB
in.gr_content8	11/1/2021 16:06	Firefox HTML Doc...	52 KB
in.gr_content9	11/1/2021 16:06	Firefox HTML Doc...	79 KB
in.gr_content10	11/1/2021 16:06	Firefox HTML Doc...	114 KB
in.gr_content11	11/1/2021 16:06	Firefox HTML Doc...	53 KB
in.gr_content12	11/1/2021 16:06	Firefox HTML Doc...	101 KB
in.gr_content13	11/1/2021 16:06	Firefox HTML Doc...	105 KB
in.gr_content14	11/1/2021 16:06	Firefox HTML Doc...	101 KB
in.gr_content15	11/1/2021 16:06	Firefox HTML Doc...	111 KB
in.gr_content16	11/1/2021 16:06	Firefox HTML Doc...	112 KB
in.gr_content17	11/1/2021 16:06	Firefox HTML Doc...	99 KB
in.gr_content18	11/1/2021 16:06	Firefox HTML Doc...	121 KB
in.gr_content19	11/1/2021 16:06	Firefox HTML Doc...	31 KB
in.gr_content20	11/1/2021 16:06	Firefox HTML Doc...	133 KB
in.gr_content21	11/1/2021 16:06	Firefox HTML Doc...	108 KB
in.gr_content22	11/1/2021 16:06	Firefox HTML Doc...	109 KB
in.gr_content23	11/1/2021 16:06	Firefox HTML Doc...	51 KB
in.gr_content24	11/1/2021 16:06	Firefox HTML Doc...	128 KB
in.gr_content25	11/1/2021 16:06	Firefox HTML Doc...	103 KB

Εικόνα 55: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο in.gr

Τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο in.gr αποθηκεύτηκαν σε 77.244 αρχεία html.

- tovima.gr: Τα αρχεία html στα οποία αποθηκεύτηκαν τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο tovim.gr είναι τα αρχεία html της Εικόνας 56.

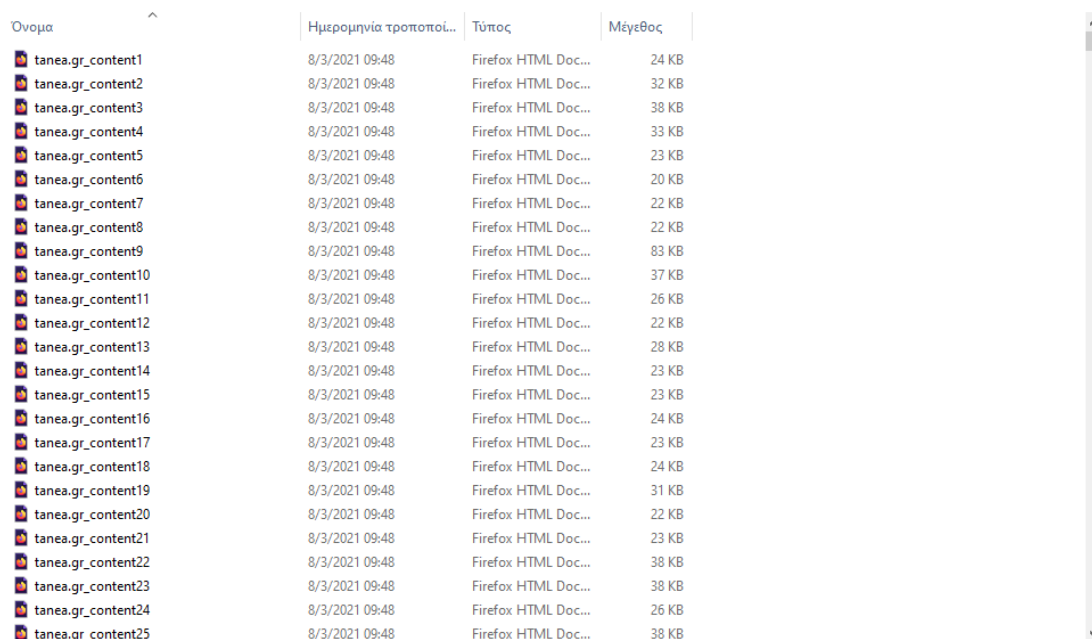
Όνομα	Ημερομηνία τροποποι...	Τύπος	Μέγεθος
tovima.gr_content1	14/1/2021 16:18	Firefox HTML Doc...	59 KB
tovima.gr_content2	14/1/2021 16:18	Firefox HTML Doc...	73 KB
tovima.gr_content3	14/1/2021 16:18	Firefox HTML Doc...	74 KB
tovima.gr_content4	14/1/2021 16:18	Firefox HTML Doc...	75 KB
tovima.gr_content5	14/1/2021 16:18	Firefox HTML Doc...	69 KB
tovima.gr_content6	14/1/2021 16:18	Firefox HTML Doc...	77 KB
tovima.gr_content7	14/1/2021 16:18	Firefox HTML Doc...	34 KB
tovima.gr_content8	14/1/2021 16:18	Firefox HTML Doc...	44 KB
tovima.gr_content9	14/1/2021 16:18	Firefox HTML Doc...	78 KB
tovima.gr_content10	14/1/2021 16:18	Firefox HTML Doc...	36 KB
tovima.gr_content11	14/1/2021 16:18	Firefox HTML Doc...	34 KB
tovima.gr_content12	14/1/2021 16:18	Firefox HTML Doc...	75 KB
tovima.gr_content13	14/1/2021 16:18	Firefox HTML Doc...	72 KB
tovima.gr_content14	14/1/2021 16:18	Firefox HTML Doc...	76 KB
tovima.gr_content15	14/1/2021 16:18	Firefox HTML Doc...	70 KB
tovima.gr_content16	14/1/2021 16:18	Firefox HTML Doc...	34 KB
tovima.gr_content17	14/1/2021 16:18	Firefox HTML Doc...	72 KB
tovima.gr_content18	14/1/2021 16:18	Firefox HTML Doc...	79 KB
tovima.gr_content19	14/1/2021 16:18	Firefox HTML Doc...	77 KB
tovima.gr_content20	14/1/2021 16:18	Firefox HTML Doc...	37 KB
tovima.gr_content21	14/1/2021 16:18	Firefox HTML Doc...	59 KB
tovima.gr_content22	14/1/2021 16:18	Firefox HTML Doc...	80 KB
tovima.gr_content23	14/1/2021 16:18	Firefox HTML Doc...	83 KB
tovima.gr_content24	14/1/2021 16:18	Firefox HTML Doc...	71 KB
tovima.gr_content25	14/1/2021 16:18	Firefox HTML Doc...	58 KB

Εικόνα 56: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο tovim.gr

Τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο tonima.gr αποθηκεύτηκαν σε 51.116 αρχεία html.

- tanea.gr:

Τα αρχεία html στα οποία αποθηκεύτηκαν τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο tanea.gr είναι τα αρχεία html της Εικόνας 57.



Όνομα	Ημερομηνία τροποποι...	Τύπος	Μέγεθος
tanea.gr_content1	8/3/2021 09:48	Firefox HTML Doc...	24 KB
tanea.gr_content2	8/3/2021 09:48	Firefox HTML Doc...	32 KB
tanea.gr_content3	8/3/2021 09:48	Firefox HTML Doc...	38 KB
tanea.gr_content4	8/3/2021 09:48	Firefox HTML Doc...	33 KB
tanea.gr_content5	8/3/2021 09:48	Firefox HTML Doc...	23 KB
tanea.gr_content6	8/3/2021 09:48	Firefox HTML Doc...	20 KB
tanea.gr_content7	8/3/2021 09:48	Firefox HTML Doc...	22 KB
tanea.gr_content8	8/3/2021 09:48	Firefox HTML Doc...	22 KB
tanea.gr_content9	8/3/2021 09:48	Firefox HTML Doc...	83 KB
tanea.gr_content10	8/3/2021 09:48	Firefox HTML Doc...	37 KB
tanea.gr_content11	8/3/2021 09:48	Firefox HTML Doc...	26 KB
tanea.gr_content12	8/3/2021 09:48	Firefox HTML Doc...	22 KB
tanea.gr_content13	8/3/2021 09:48	Firefox HTML Doc...	28 KB
tanea.gr_content14	8/3/2021 09:48	Firefox HTML Doc...	23 KB
tanea.gr_content15	8/3/2021 09:48	Firefox HTML Doc...	23 KB
tanea.gr_content16	8/3/2021 09:48	Firefox HTML Doc...	24 KB
tanea.gr_content17	8/3/2021 09:48	Firefox HTML Doc...	23 KB
tanea.gr_content18	8/3/2021 09:48	Firefox HTML Doc...	24 KB
tanea.gr_content19	8/3/2021 09:48	Firefox HTML Doc...	31 KB
tanea.gr_content20	8/3/2021 09:48	Firefox HTML Doc...	22 KB
tanea.gr_content21	8/3/2021 09:48	Firefox HTML Doc...	23 KB
tanea.gr_content22	8/3/2021 09:48	Firefox HTML Doc...	38 KB
tanea.gr_content23	8/3/2021 09:48	Firefox HTML Doc...	38 KB
tanea.gr_content24	8/3/2021 09:48	Firefox HTML Doc...	26 KB
tanea.gr_content25	8/3/2021 09:48	Firefox HTML Doc...	38 KB

Εικόνα 57: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο tanea.gr

Τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο tanea.gr αποθηκεύτηκαν σε 46.387 αρχεία html.

- protothema.gr:

Τα αρχεία html στα οποία αποθηκεύτηκαν τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο protothema.gr είναι τα αρχεία html της Εικόνας 58.

Όνομα	Ημερομηνία τροποποι...	Τύπος	Μέγεθος
protothema.gr_content1	26/4/2021 17:00	Firefox HTML Doc...	95 KB
protothema.gr_content2	26/4/2021 17:00	Firefox HTML Doc...	117 KB
protothema.gr_content3	26/4/2021 17:00	Firefox HTML Doc...	90 KB
protothema.gr_content4	26/4/2021 17:00	Firefox HTML Doc...	99 KB
protothema.gr_content5	26/4/2021 17:00	Firefox HTML Doc...	92 KB
protothema.gr_content6	26/4/2021 17:00	Firefox HTML Doc...	92 KB
protothema.gr_content7	26/4/2021 17:00	Firefox HTML Doc...	94 KB
protothema.gr_content8	26/4/2021 17:00	Firefox HTML Doc...	77 KB
protothema.gr_content9	26/4/2021 17:00	Firefox HTML Doc...	82 KB
protothema.gr_content10	26/4/2021 17:00	Firefox HTML Doc...	92 KB
protothema.gr_content11	26/4/2021 17:00	Firefox HTML Doc...	103 KB
protothema.gr_content12	26/4/2021 17:00	Firefox HTML Doc...	170 KB
protothema.gr_content13	26/4/2021 17:00	Firefox HTML Doc...	104 KB
protothema.gr_content14	26/4/2021 17:00	Firefox HTML Doc...	93 KB
protothema.gr_content15	26/4/2021 17:00	Firefox HTML Doc...	80 KB
protothema.gr_content16	26/4/2021 17:00	Firefox HTML Doc...	30 KB
protothema.gr_content17	26/4/2021 17:00	Firefox HTML Doc...	99 KB
protothema.gr_content18	26/4/2021 17:00	Firefox HTML Doc...	46 KB
protothema.gr_content19	26/4/2021 17:00	Firefox HTML Doc...	93 KB
protothema.gr_content20	26/4/2021 17:00	Firefox HTML Doc...	83 KB
protothema.gr_content21	26/4/2021 17:00	Firefox HTML Doc...	98 KB
protothema.gr_content22	26/4/2021 17:00	Firefox HTML Doc...	82 KB
protothema.gr_content23	26/4/2021 17:00	Firefox HTML Doc...	89 KB
protothema.gr_content24	26/4/2021 17:00	Firefox HTML Doc...	90 KB
protothema.gr_content25	26/4/2021 17:00	Firefox HTML Doc...	94 KB

Εικόνα 58: Αρχεία html στα οποία αποθηκεύτηκαν περιεχόμενα σελίδων που συλλέχθηκαν από τον ιστοχώρο protothema.gr

Τα περιεχόμενα των σελίδων που συλλέχθηκαν από τον ιστοχώρο protothema.gr αποθηκεύτηκαν σε 21.531 αρχεία html.

4.3 Αποτελέσματα του σταδίου Εξαγωγή Πληροφοριών

Τα αποτελέσματα του σταδίου Εξαγωγή Πληροφοριών είναι τα αποτελέσματα της Εικόνας 59.

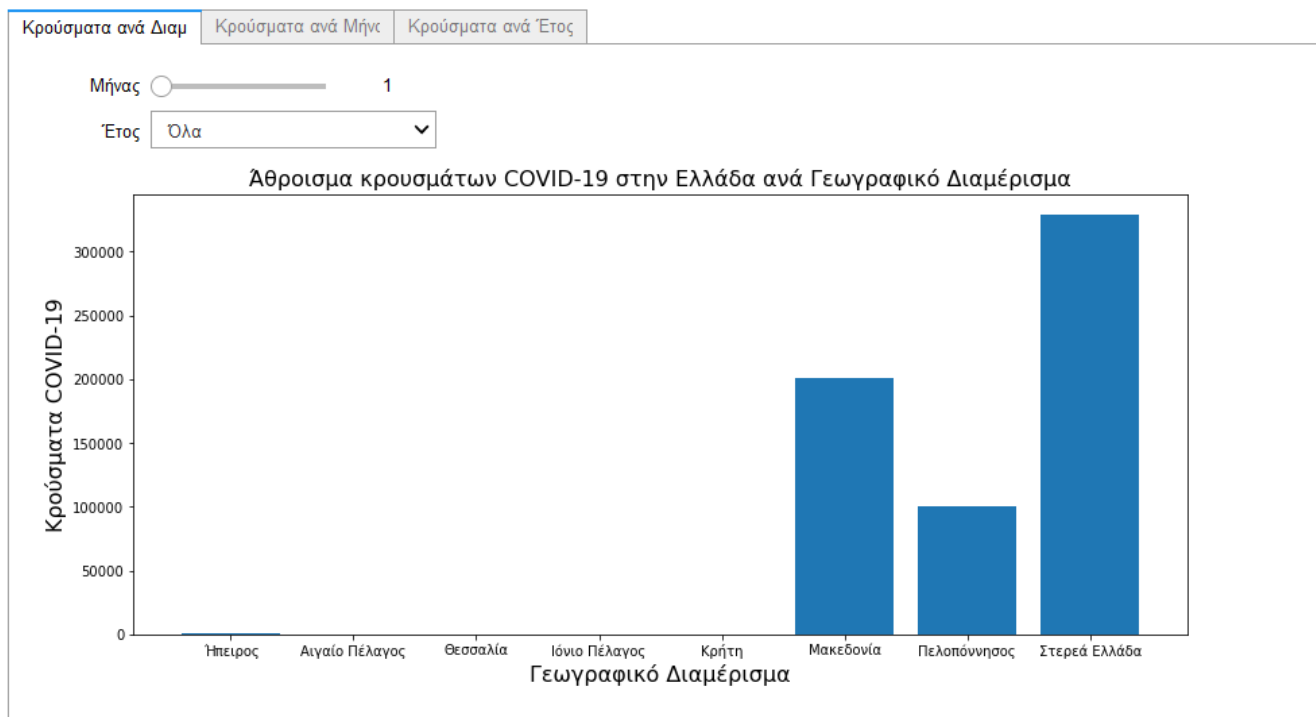
1	Date, Location, Cases
2	2000-11-21, Αττική, 271
3	2005-05-20, Αττική, 364
4	2013-08-23, Πρέβεζα, 17
5	2014-03-01, Αττική, 364
6	2017-09-16, Αττική, 676
7	2018-04-12, Αττική, 676
8	2018-04-13, Αττική, 676
9	2018-04-16, Αττική, 676
10	2018-11-09, Αττική, 676
11	2018-11-18, Αττική, 676
12	2018-11-30, Αττική, 676
13	2018-12-09, Αττική, 676
14	2018-12-19, Αττική, 364
15	2018-12-25, Πρέβεζα, 17
16	2018-12-28, Αττική, 364
17	2018-12-29, Αττική, 271
18	2019-01-01, Αττική, 271
19	2019-01-05, Αττική, 364
20	2019-01-11, Αττική, 364
21	2019-01-14, Πρέβεζα, 17
22	2019-01-21, Πρέβεζα, 17
23	2019-01-29, Άργος - Μυκήνες, 400
24	2019-01-29, Αττική, 364
25	2019-01-29, Πρέβεζα, 17
26	2019-01-31, Αττική, 364
27	2019-02-01, Αττική, 364
28	2019-02-04, Αττική, 364
29	2019-02-04, Πρέβεζα, 17
30	2019-02-05, Πρέβεζα, 17
31	2019-02-06, Αττική, 364
32	2019-02-07, Λάρισα, 11
33	2019-02-15, Πρέβεζα, 17
34	2019-02-16, Αττική, 364
35	2019-02-20, Αττική, 271
36	2019-02-21, Αττική, 364
37	2019-02-22, Πρέβεζα, 17
38	2019-02-23, Αττική, 364
39	2019-02-23, Πρέβεζα, 17
40	2019-02-25, Πρέβεζα, 17
41	2019-03-04, Πρέβεζα, 17
42	2019-03-06, Αττική, 271
43	2019-03-08, Αττική, 364
44	2019-03-09, Αττική, 271
45	2019-03-09, Πρέβεζα, 17
46	2019-03-11, Αττική, 364
47	2019-03-12, Αττική, 364
48	2019-03-15, Αττική, 271
49	2019-03-17, Αττική, 364
50	2019-03-20, Πρέβεζα, 17

Εικόνα 59: Αποτελέσματα του σταδίου Εξαγωγή Πληροφοριών

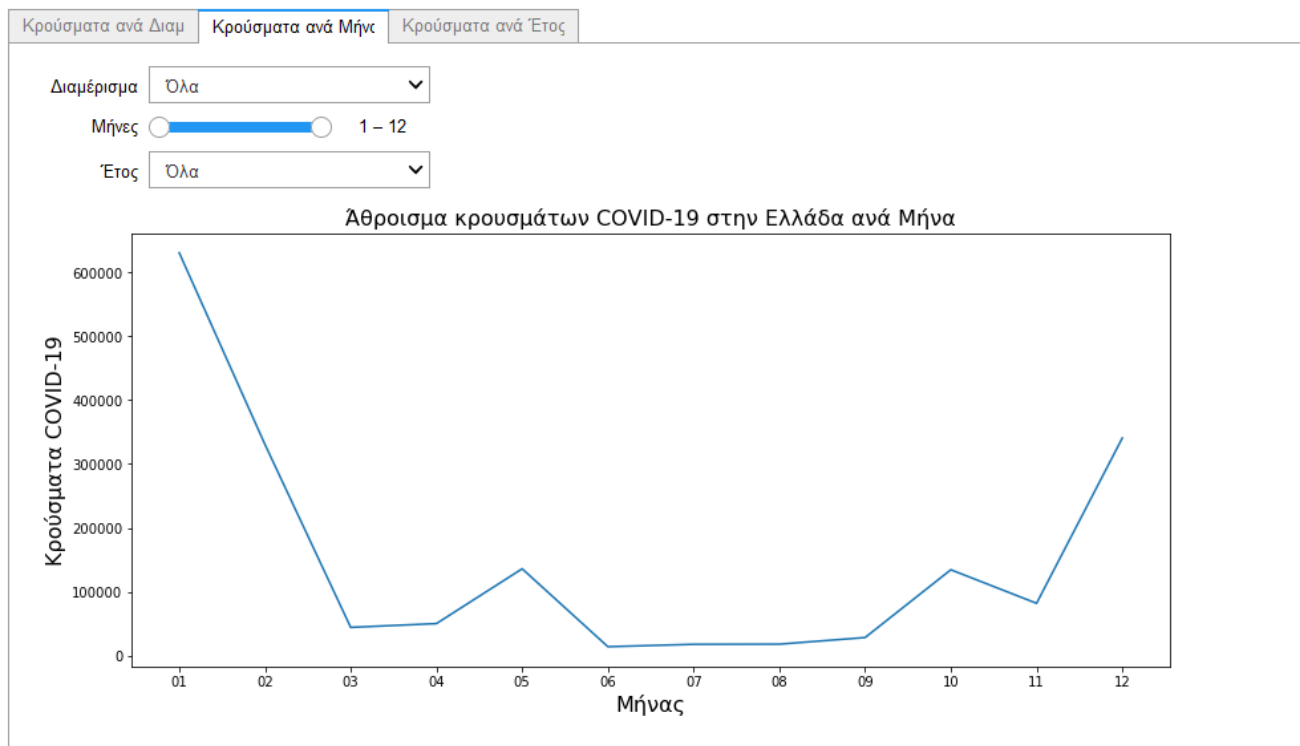
Οι τριάδες της μορφής ημερομηνία, τοποθεσία, κρούσματα από την Εικόνα 59 είναι 7.793.

4.4 Αποτελέσματα του σταδίου Οπτικοποίηση Δεδομένων

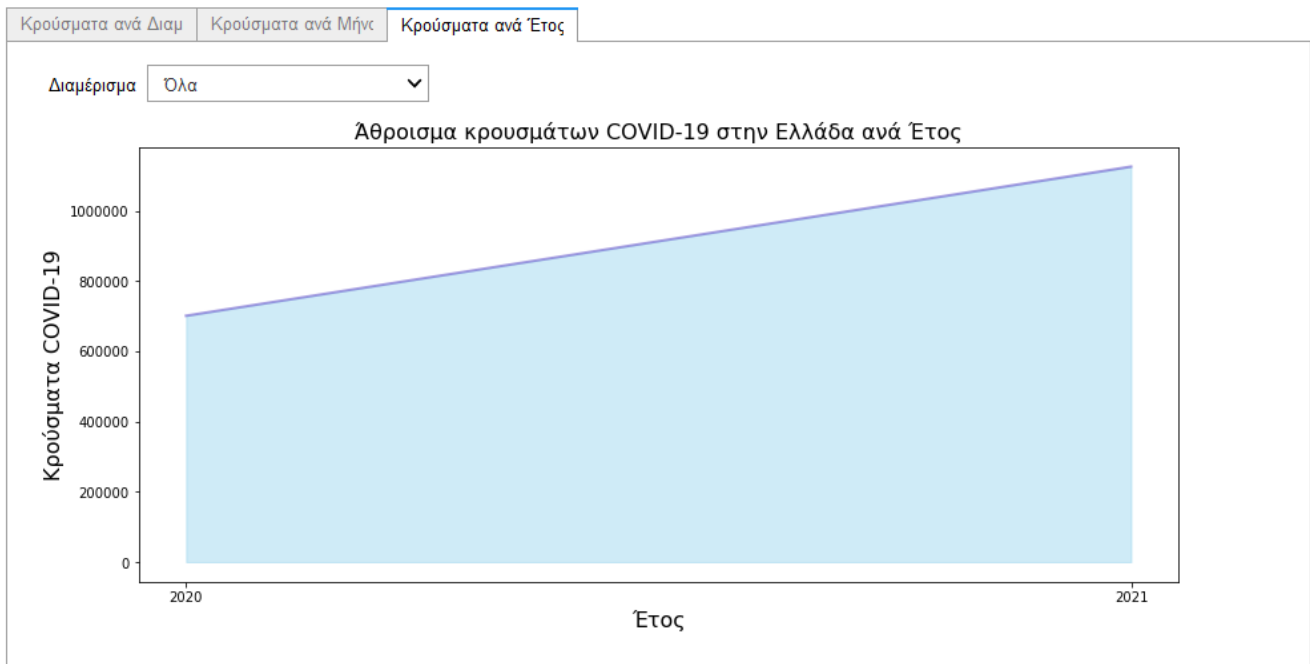
Τα αποτελέσματα του σταδίου Οπτικοποίηση Δεδομένων είναι οι οπτικοποιήσεις των Εικόνων 60, 61 και 62.



Εικόνα 60: Κατακόρυφο γράφημα ράβδων (bar chart) όπου οπτικοποιούνται τα κρούσματα ανά γεωγραφικό διαμέρισμα της Ελλάδας με φίλτρα το μήνα και το έτος



Εικόνα 61: Γράφημα γραμμής (Line chart) όπου οπτικοποιούνται τα κρούσματα ανά μήνα με φίλτρα το γεωγραφικό διαμέρισμα της Ελλάδας, τους μήνες και το έτος



Εικόνα 62: Γράφημα περιοχής (Area chart) όπου οπτικοποιούνται τα κρούσματα ανά έτος με φίλτρο το γεωγραφικό διαμέρισμα της Ελλάδας

Οι οπτικοποιήσεις των Εικόνων 60, 61 και 62 εξηγούνται κάτωθι:

- **Εικόνα 60:**
Στην οπτικοποίηση της εικόνας αυτής ο χρήστης έχει τη δυνατότητα να επιλέξει από την μπάρα μήνα και από το dropdown έτος. Με άλλα λόγια ο χρήστης επιλέγει το μήνα και το έτος στα οποία υπάρχουν κρούσματα της λοίμωξης COVID-19 ανά γεωγραφικό διαμέρισμα της Ελλάδας. Ο συγκεκριμένος τύπος οπτικοποίησης προτιμήθηκε επειδή τα γεωγραφικά διαμερίσματα λαμβάνουν κατηγορικά δεδομένα (categorical data) τα οποία λαμβάνουν τιμές που δεν μπορούν να μπουν σε μια διάταξη. Παραδείγματα κατηγορικών δεδομένων είναι τα ακόλουθα:
Παράδειγμα 1: Ήπειρος, Θεσσαλία, Κρήτη δεν μπαίνουν σε μια διάταξη.
Παράδειγμα 2: Σωστό, Λάθος δεν μπαίνουν σε μια διάταξη.
- **Εικόνα 61:**
Στην οπτικοποίηση της εικόνας αυτής ο χρήστης έχει τη δυνατότητα να επιλέξει από το πρώτο dropdown γεωγραφικό διαμέρισμα της Ελλάδας, από την μπάρα εύρος μηνών και από το δεύτερο dropdown έτος. Με άλλα λόγια ο χρήστης επιλέγει το εύρος μηνών, το έτος και το γεωγραφικό διαμέρισμα της Ελλάδας στα οποία υπάρχουν κρούσματα της λοίμωξης COVID-19. Αυτός ο τύπος οπτικοποίησης προτιμήθηκε επειδή οι μήνες είναι διατεταγμένα δεδομένα (ordinal data) και άρα λαμβάνουν τιμές που μπορούν να μπουν σε μια διάταξη. Παραδείγματα διατεταγμένων δεδομένων είναι τα ακόλουθα:
Παράδειγμα 1: Δευτέρα, Τρίτη, Τετάρτη μπαίνουν σε μια διάταξη
Παράδειγμα 2: Ιανουάριος, Φεβρουάριος, Μάρτιος μπαίνουν σε μια διάταξη

- Εικόνα 62:

Στην οπτικοποίηση της εικόνας αυτής ο χρήστης έχει τη δυνατότητα να επιλέξει από το dropdown γεωγραφικό διαμέρισμα της Ελλάδας. Με άλλα λόγια ο χρήστης επιλέγει το γεωγραφικό διαμέρισμα της Ελλάδας στο οποίο υπάρχουν κρούσματα της λοίμωξης COVID-19. Όπως και στην περίπτωση της εικόνας 59, αυτός ο τύπος οπτικοποίησης προτιμήθηκε επειδή τα έτη λαμβάνουν διατεταγμένα δεδομένα (ordinal data) τα οποία μπορούν να μπουν σε μια διάταξη. Εξηγήθηκαν προηγουμένως παραδείγματα διατεταγμένων δεδομένων.

Από το link <https://github.com/stylservos/COVID-19-Cases-in-Greece> ο χρήστης θα μεταφερθεί στο link <https://hub.gke2.mybinder.org/user/stylservos-covi-cases-in-greece-km7k17nd/notebooks/COVID-19%20Cases%20in%20Greece.ipynb> πατώντας το κουμπί *launch binder*.

Στο δεύτερο link υπάρχουν οι οπτικοποιήσεις που δημιουργήθηκαν. Στην περίπτωση που δεν εμφανιστούν αμέσως, ο χρήστης πατάει το κουμπί *Run*.

5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η παρούσα διπλωματική εργασία, μετά τη θεωρητική τους υποστήριξη, παρουσίασε τα στάδια υλοποίησης για την οπτικοποίηση μεγάλων δεδομένων που σχετίζονται με τη λοίμωξη COVID-19 στην Ελλάδα από 4 ελληνικούς ειδησεογραφικούς ιστοχώρους. Για την υλοποίηση αυτή έγινε χρήση της γλώσσας προγραμματισμού Python και του εργαλείου προγραμματισμού Jupyter Notebook.

Κατά την επιτέλεση του σταδίου «Συλλογή σελίδων» αποκτήθηκε στην πράξη γνώση της μεγαλύτερης ευκολίας στη χρήση των βιβλιοθηκών «beautifulsoup4», «requests», «re», οι οποίες είναι οι βιβλιοθήκες που χρησιμοποιήθηκαν κατά το στάδιο αυτό, έναντι βιβλιοθηκών όπως η scrapy, η οποία είναι αρκετά πιο δύσκολη. Έγινε, επίσης, κατανοητή, για άλλη μια φορά, η ευκολία στη χρήση των ευρέως διαδεδομένων στην πληροφορική αρχείων τύπου csv, στα οποία συγκεντρώθηκαν τα urls των σελίδων.

Κατά την εκτέλεση του επόμενου σταδίου, «Αποθήκευση περιεχομένων σελίδων» έγινε επίσης χρήση των βιβλιοθηκών «beautifulsoup4» και «requests», για να διαβαστούν τα περιεχόμενα των ιστοσελίδων και να αποθηκευτούν σε αρχεία html αν είχαν περιεχόμενα κειμένου. Η χρήση των συγκεκριμένων βιβλιοθηκών παρουσιάζει τα πλεονεκτήματα που αναφέρθηκαν στο προηγούμενο στάδιο, «Συλλογή σελίδων». Παράλληλα, κατά το στάδιο αυτό αποκτήθηκε γνώση του ποιος είναι ο βέλτιστος τρόπος να αποφευχθεί η πιθανότητα αποθήκευσης σε αρχείο html περιεχομένων σελίδων που ήδη έχουν αποθηκευτεί. Συγκεκριμένα, ως υποθέσουμε ότι S1 είναι το σύνολο των σελίδων που συλλέχθηκαν στο προηγούμενο στάδιο και S2 το σύνολο των σελίδων των οποίων αποθηκεύτηκαν τα περιεχόμενα. Αφού υπολογιστεί η διαφορά των συνόλων S1-S2, επιτελείται αποθήκευση των περιεχομένων σελίδων που δεν είχαν προηγουμένως αποθηκευτεί. Ο τρόπος αυτός είναι ιδιαίτερα αποτελεσματικός ως προς το χρόνο εκτέλεσης του κώδικα, καθώς αποφεύγεται το ενδεχόμενο να διαβαστεί από τον κώδικα το σύνολο των περιεχομένων των σελίδων που συλλέχθηκαν, διεργασία που θα απαιτούσε πολύ περισσότερο χρόνο.

Κατά το στάδιο «Εξαγωγή πληροφοριών», η υπεροχή στη χρήση της βιβλιοθήκης «beautifulsoup4» έγινε αντιληπτή για άλλη μια φορά, κατά την ανάγνωση των αρχείων html τα οποία δημιουργήθηκαν στο προηγούμενο στάδιο. Ο καλύτερος τρόπος για να διαβαστούν τα αρχεία html και να βρεθούν για κάθε αρχείο οι λέξεις «κρούσμα» και «κρούσματα», η περιοχή της Ελλάδας, η ημερομηνία του άρθρου και ο αριθμός των κρουσμάτων έγινε γνωστός. Η διαδικασία ήταν ανάλογη της διαδικασίας που υλοποιήθηκε στο στάδιο αποθήκευσης περιεχομένων σελίδων σε html, με αντίστοιχα πλεονεκτήματα. Αξίζει να αναφερθεί ότι για την ανάγνωση περιεχομένου ιστοσελίδων ως προς την ημερομηνία του εκάστοτε άρθρου χρησιμοποιήθηκε, πέρα από τη «beautifulsoup4», και η βιβλιοθήκη «requests». Βρέθηκε επίσης αποτελεσματικός τρόπος να αποφευχθεί το ενδεχόμενο κάθε φορά που εκτελείται το πρόγραμμα να γίνεται επεξεργασία ελληνικών τοποθεσιών, γεγονός που θα ήταν ιδιαίτερα χρονοβόρο. Αυτό κατέστη εφικτό με τη χρήση ενός βοηθητικού προγράμματος που επεξεργάζεται ελληνικές τοποθεσίες. Το πρόγραμμα δημιούργησε ένα λεξικό (dictionary) για νομούς και ένα λεξικό (dictionary) για δήμους και κάθε ένα εξ' αυτών αποθηκεύτηκε σε ένα αρχείο τύπου json. Τα αρχεία αυτά αξιοποιήθηκαν με τη σειρά τους στο στάδιο που υλοποιήθηκε η εξαγωγή των πληροφοριών. Κατά τη διαδικασία αυτή έγινε κατανοητή η υπεροχή των αρχείων τύπου json έναντι των αρχείων csv για την αποθήκευση δεδομένων τύπου dictionary.

Η οπτικοποίηση δεδομένων έγινε με τη χρήση της βιβλιοθήκης «matplotlib», ενώ για τη διαδραστικότητα των οπτικοποιήσεων χρησιμοποιήθηκε η «ipywidgets». Έγινε αντιληπτή η ευκολία στη χρήση της βιβλιοθήκης matplotlib, έναντι άλλων βιβλιοθηκών όπως η seaborn. Αντίστοιχα, αποκτήθηκε γνώση της ευκολίας στη χρήση των διαδραστικών εργαλείων που παρέχει η βιβλιοθήκη «ipywidgets», καθώς και η ευκολία στη χρήση της ίδιας της βιβλιοθήκης.

Αναφορικά με τις οπτικοποιήσεις που υλοποιήθηκαν στην παρούσα εργασία, πλεονεκτήματά τους σε σχέση με άλλες οπτικοποιήσεις δεδομένων γύρω από το ίδιο θέμα που μπορούν να ανευρεθούν στο διαδίκτυο, είναι ότι είναι ιδιαίτερα εύκολες στην κατανόησή τους, περιλαμβάνουν τόσο τιμές που μπορούν να μπουν σε μια διάταξη (διατεταγμένα δεδομένα) όσο και τιμές που δεν μπορούν να μπουν σε μια διάταξη (κατηγορικά δεδομένα). Μπορούν επίσης μέσω αυτών να συγκριθούν γεωγραφικά διαμερίσματα της Ελλάδας ως προς τον αριθμό κρουσμάτων από τη λοίμωξη Covid19 (έναντι νομών ή δήμων, που συναντώνται στις άλλες έως σήμερα ευρέως διαθέσιμες στο κοινό οπτικοποιήσεις).

ΠΑΡΑΡΤΗΜΑ

Προγράμματα Python

I.1 Προγράμματα με λίστες

I.1.1 Πρόγραμμα που βρίσκει τους άρτιους αριθμούς μιας λίστας

```
numbers = [1,2,3,4,5,6,7,8,9,10]
evens = [number for number in numbers if (number % 2 == 0)]
print evens
```

I.1.2 Πρόγραμμα που βρίσκει τα τετράγωνα των αριθμών μιας λίστας

```
numbers = [1,2,3,4,5,6,7,8,9,10]
squares = [number ** 2 for number in numbers]
print squares
```

I.2 Προγράμματα με συμβολοσειρές

I.2.1 Πρόγραμμα που τυπώνει Hello World

```
print "Hello World"
```

I.2.2 Πρόγραμμα που διαβάζει και τυπώνει συμβολοσειρά

```
my_string = raw_input("Give a string: ")
print my_string
```

ΑΝΑΦΟΡΕΣ

- [1] B.L. Tesini, "Respiratory Syndromes (COVID-19, MERS, and SARS)," στο *Merck Manual Consumer Version*. (2021). [Online]. Διαθέσιμο: <https://www.merckmanuals.com/home/infections/respiratory-viruses/coronaviruses-and-acute-respiratory-syndromes-covid-19-mers-and-sars?query=coronavirus>. [Προσπελάστηκε: Απρ. 29, 2021].
- [2] World Health Organization, "Weekly epidemiological update on COVID-19," Απρ. 27, 2021. [Online]. Διαθέσιμο: <https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19---27-april-2021>. [Προσπελάστηκε: Απρ. 27, 2021].
- [3] Α. Ντούλας. (2021). Διαχείριση Μεγάλων Δεδομένων.[Διαφάνειες PowerPoint]. Διαθέσιμο: <https://eclass.uoa.gr/modules/auth/opencourses.php?fc=648>
- [4] R. Mitchell, *Web Scraping with Python: Collecting Data from the Modern Web*, 1st. ed, O'Reilly Media, Inc., 2015. [Online]. Διαθέσιμο: <https://yanfei.site/docs/dpsa/references/PyWebScrapingBook.pdf>. [Προσπελάστηκε: Απρ. 19, 2021].
- [5] "Χάρτης Υγειονομικής Ασφάλειας και Προστασίας από τη λοίμωξη covid-19," *covid19.gov.gr*, Οκτ. 23, 2020. [Online]. Διαθέσιμο: <https://covid19.gov.gr/covid-map>. [Προσπελάστηκε Απρ. 20, 2021].
- [6] "Ημερήσια Επισκόπηση: Ενημέρωση για την πορεία του COVID-19 στην Ελλάδα," *covid19.gov.gr*, Απρ. 04, 2020. [Online]. Διαθέσιμο: <https://covid19.gov.gr/covid19-live-analytics/>. [Προσπελάστηκε Απρ. 20, 2021].
- [7] Johns Hopkins University, "Johns Hopkins Coronavirus Resource Center," *Johns Hopkins Coronavirus Resource Center*, 2021. [Online]. Διαθέσιμο: <https://coronavirus.jhu.edu/map.html>. [Προσπελάστηκε Απρ. 20, 2021].
- [8] M. Roser, H. Ritchie, E. Ortiz-Ospina, and J. Hasell, "Greece: Coronavirus Pandemic Country Profile," *Our World in Data*, May 2020, [Online]. Available: <https://ourworldindata.org/coronavirus/country/greece?country=-GRC>. [Προσπελάστηκε Απρ. 20, 2021].
- [9] "Beautiful Soup Documentation — Beautiful Soup 4.9.0 documentation," *Crummy: The Site*. [Online]. Διαθέσιμο: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. [Προσπελάστηκε: Απρ. 03, 2021].
- [10] "Extract links from webpage (BeautifulSoup)," *Python Tutorials*. [Online]. Διαθέσιμο: <https://pythonspot.com/extract-links-from-webpage-beautifulsoup/>. [Προσπελάστηκε: Απρ. 03, 2021].
- [11] M. Breuss, "Beautiful Soup: Build a Web Scraper with Python," *Real Python*. [Online]. Διαθέσιμο: <https://realpython.com/beautiful-soup-web-scraper-python/>. [Προσπελάστηκε: Απρ. 10, 2021].
- [12] J. Fincher, "Reading and Writing CSV Files in Python," *Real Python*. [Online] Διαθέσιμο: <https://realpython.com/python-csv/>. [Προσπελάστηκε: Απρ. 10, 2021].
- [13] "Interacting with CSV files using Python," *beancoder.com*. [Online]. Διαθέσιμο: <http://beancoder.com/csv-files-using-python/>. [Προσπελάστηκε: Απρ. 11, 2021].
- [14] "Python BeautifulSoup," *ZetCode*, Ιούλ. 27, 2020. [Online]. Διαθέσιμο: <https://zetcode.com/python/beautifulsoup/>. [Προσπελάστηκε: Απρ. 03, 2021]
- [15] "Python os.listdir() Method," *tutorialspoint.com*. [Online] Διαθέσιμο: https://www.tutorialspoint.com/python/os_listdir.htm. [Προσπελάστηκε: Απρ. 10, 2021].
- [16] C. Tyagi, "Adding new column to existing DataFrame in Pandas," *GeeksforGeeks*, Δεκ. 11, 2018. [Online]. Διαθέσιμο: <https://www.geeksforgeeks.org/adding-new-column-to-existing-dataframe-in-pandas/>. [Προσπελάστηκε: Απρ. 10, 2021].
- [17] "How to plot a Line Chart in Python using Matplotlib," *datatofish.com*, Απρ. 12, 2020. [Online]. Διαθέσιμο: <https://datatofish.com/line-chart-python-matplotlib/>. [Προσπελάστηκε: Απρ. 11, 2021].
- [18] "Matplotlib Bars," στο *Python Tutorial*, *w3schools.com*. [Online]. Διαθέσιμο: https://www.w3schools.com/python/matplotlib_bars.asp. [Προσπελάστηκε: Απρ. 12, 2021].
- [19] J. Zheng, "Matplotlib Series 7: Area chart," *Jingwen Zheng*, Νοέμ. 14, 2018. [Online]. Διαθέσιμο: <https://jingwen-z.github.io/data-viz-with-matplotlib-series7-area-chart/>. [Προσπελάστηκε: Απρ. 12, 2021].
- [20] "ipywidgets — Jupyter Widgets 8.0.0a4 documentation," *ipywidgets.readthedocs.io*. [Online]. Διαθέσιμο: <https://ipywidgets.readthedocs.io/en/latest/index.html>. [Προσπελάστηκε: Ιούν. 01, 2021].
- [21] L. Tagliaferri, "How To Work with Web Data Using Requests and BeautifulSoup with Python 3," *DigitalOcean*, Ιούλ. 14, 2017. [Online]. Διαθέσιμο: <https://www.digitalocean.com/community/tutorials/how-to-work-with-web-data-using-requests-and-beautiful-soup-with-python-3>. [Προσπελάστηκε: Ιούν. 01, 2021]