Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης
Εργαστηριακή Άσκηση 2022-2023

Λένος Χρίστου (ΑΜ: 1063014)
Γρηγόρης Καπαδούκας (ΑΜ: 1072484)

29 Απριλίου 2023

0 Αναλυτική Καταγραφή του Περιβάλλοντος Υλοποίησης

0.1 Καταγραφή Βιβλιοθηκών που Χρησιμοποιήθηκαν

Για να υλοποιήσουμε την εργασία χρησιμοποιήσαμε γλώσσα προγραμματισμού Python, όπως ζητείται στην εκφώνηση, με τις εξής κύριες βιβλιοθήκες:

- Matplotlib
- Numpy
- Pandas
- Scipy
- Seaborn
- Scikit-learn
- Jupytext (για τη προαιρετική χρήση Jupyter Notebook)

0.2 Αναλυτικά Βήματα για την Δημιουργία Πανομοιότυπου Περιβάλλοντος Υλοποίησης

Παρακάτω δίνουμε αναλυτικά βήματα για την εγκατάσταση των βιβλιοθηκών σε ένα Python virtual environment, έτσι ώστε το περιβάλλον υλοποίησης να είναι πανομοιότυπο με αυτό που χρησιμοποιήσαμε εμείς:

1. Εγκατάσταση του Miniconda μέσω του installer στη σελίδα:

   https://docs.conda.io/en/latest/miniconda.html

   Το Miniconda είναι μια δωρεάν μινιμαλιστική πλατφόρμα με cross-platform υποστήριξη που περιέχει το εργαλείο conda, με σκοπό την εύκολη δημιουργία και

διαχείριση των Python virtual environments.

Τα virtual environments αποτελούν ένα ”απομονωμένο χώρο” όπου μπορούμε να εγκαταστήσουμε και να χρησιμοποιήσουμε κάποια συγκεκριμένη έκδοση της Python και βιβλιοθήκες της, χωρίς να επηρεάσουμε τυχόν εγκατάσταση της Python που βρίσκεται ήδη στο σύστημα.

Εναλλακτική επιλογή που μπορεί να χρησιμοποιηθεί στη θέση του Miniconda είναι το Anaconda. Το Miniconda αναφέρεται επειδή το προτιμήσαμε εμείς στην χρήση μας.

2. Δημιουργία του conda virtual environment με αυτόματη εγκατάσταση των βιβλιοθηκών που επιθυμούμε μέσω της εκτέλεση της εξής εντολής στον φάκελο της εργασίας σε τερματικό (ή command prompt αντίστοιχα σε πλατφόρμα Windows):

```
conda env create -f environment.yml
```

Ή άμα επιθυμείται εγκατάσταση του Jupyter Notebook ταυτόχρονα, μέσω της εντολής:

```
conda env create -f environment-jupyter-notebook.yml
```

3. Η εγκατάσταση των βιβλιοθηκών στο virtual environment έχει ολοκληρωθεί, οπότε τώρα θα φορτώσουμε το environment με την εξής εντολή:

```
conda activate tf
```

4. Τώρα πλέον είμαστε έτοιμοι και μπορούμε να εκτελέσουμε τον κώδικα απευθείας στο τερματικό ή μέσω του Jupyter Notebook:

Για να εκτελέσουμε απευθείας τον κώδικα στο environment εκτελούμε απλά την εξής εντολή:

```
python Code/<filename>.py
```

Για να εκτελέσουμε το Jupyter Notebook στο environment εκτελούμε την εξής εντολή στο τερματικό:

```
jupyter notebook
```

1 Υλοποίηση και Αποτελέσματα Ερωτήματος 1

1.1 Σύντομη Περιγραφή της Διαδικασίας Υλοποίησης

Για την υλοποίηση του ερωτήματος αυτού, αναφέρουμε αρχικά ότι χρησιμοποιούμε την βιβλιοθήκη Pandas για το διάβασμα και την χρήστη του data.csv αρχείου με τα δεδομένα μας.

Έπειτα χρησιμοποιώντας την μέθοδο describe() του Pandas τυπώνουμε σχετικά στατιστικά στοιχεία για τα δεδομένα κάθε στήλης, πιο συγκεκριμένα τιμές για το συνολικό άθροισμα ανά στήλη (count), μέση τιμή (mean), τυπική απόκλιση (std), ελάχιστη τιμή

(min), τιμές 25%, 50%, 75%, και μέγιστη τιμή (max). Στις τιμές αυτές στρογγυλοποιούμε στα δύο δεκαδικά ψηφία για το τύπωμα, ώστε να είναι πιο αναγνώσιμα.

Ακόμα τυπώνουμε μέσω for λούπας ιστογράμματα για κάθε Series του Pandas DataFrame που περιέχει τα δεδομένα του dataset, με εξαίρεση τα Series 'Country' και 'Date', για τα οποία η προβολή ιστογράμματος δεν θα μας δώσει χρήσιμη πληροφορία. Με αυτόν τον τρόπο μπορούμε να καταλάβουμε την κατανομή των δεδομένων. Για τα ιστογράμματα χρησιμοποιούμε την παράμετρο "bins = 20" με σκοπόν να κάνουμε grouping των τιμών του άξονα x σε 20 ισομεγέθη bins στο εύρος τιμών που παίρνουν οι τιμές των εισαγωγών στο Series κάθε φορά. Αυτό είναι αναγκαίο επειδή πολύ συχνά τιμές εμφανίζονται μια μόνο φορά στα δεδομένα, οπότε αν δείξουμε αυτές τις τιμές χωρίς κανένα grouping δεν θα μπορούσαμε να λάβουμε αποτελέσματα σχετικά με την κατανομή της πιθανότητας των τιμών.

Τέλος με χρήση των βιβλιοθηκών Pandas, Seaborn και Matplotlib, υπολογίζουμε αρχικά το Correlation Matrix και έπειτα κάνουμε plot το Correlation Matrix Heatmap που προκύπτει από αυτό, έτσι ώστε να εμφανίζονται οι τίτλοι των Series στους άξονες x και y και να εμφανίζονται οι τιμές του correlation που προέκυψαν στο Correlation Matrix ως κελιά στο σημείο τομής οποιονδήποτε δύο κατηγοριών. Οι τιμές για το correlation ανήκουν στο εύρος [-1,1] με 1 να σημαίνει πλήρη συσχέτιση, 0 να σημαίνει καμία συσχέτιση και -1 να σημαίνει πλήρη αρνητική συσχέτιση (πχ αντιστρόφως ανάλογες τιμές).

1.2 Τελικά Αποτελέσματα και Σχολιασμός τους

1.2.1 Τελικά Αποτελέσματα

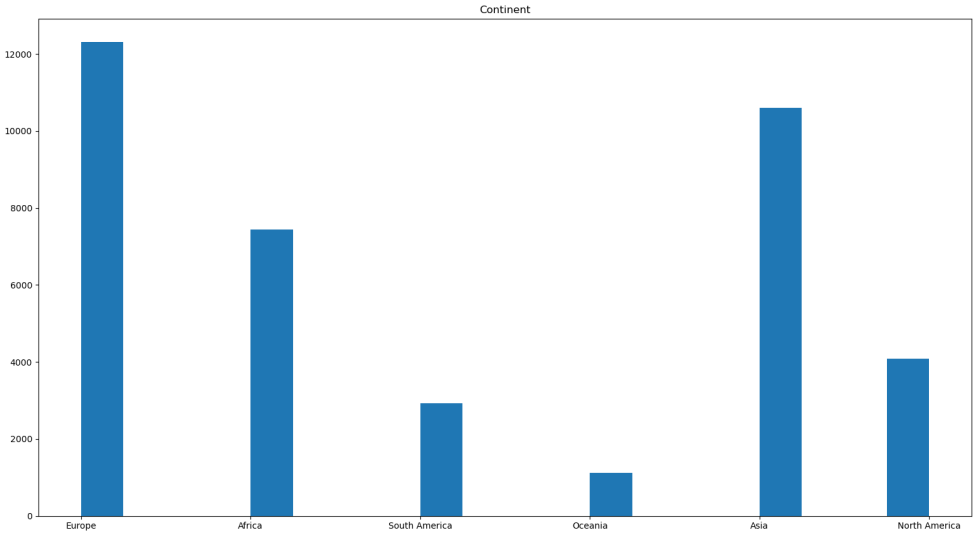
Παρακάτω παρουσιάζουμε ένα πίνακα με τα αποτελέσματα που προέκυψαν από την εκτέλεση της μεθόδου describe() του Pandas στο DataFrame του dataset:

Value	Latitude	Longitude	Average temperature per year	Hospital beds per 1000 people	Medical doctors per 1000 people	GDP / Capita	Population	Median age	Population aged 65 and over (%)	Daily tests	Cases	Deaths
count	38472.0	38472.0	38472.0	38472.0	38472.0	38472.0	38472.0	38472.0	38472.0	30577.0	38218.0	34862.0
mean	23.74	20.21	17.72	3.17	2.09	19002.33	48969829.03	32.75	10.66	39440.59	287902.66	8090.5
std	26.06	61.07	8.13	2.56	1.52	22271.11	142725118.68	8.47	6.77	150184.66	1405242.87	29548.75
min	-40.9	-106.35	-2.0	0.2	0.02	411.6	341284.0	16.0	1.0	-239172.0	1.0	1.0
25%	8.62	-3.44	11.0	1.4	0.82	3659.0	4793900.0	27.0	5.0	1505.0	2074.0	77.0
50%	27.51	21.82	20.0	2.5	1.89	8821.8	11484636.0	32.0	8.0	5520.0	21431.0	527.0
75%	45.94	47.48	25.0	4.49	3.21	25946.2	42862958.0	41.0	16.0	20382.0	137377.0	3480.5
max	64.96	179.41	29.0	13.05	7.52	114704.6	1339180127.0	48.0	28.0	2945871.0	28605669.0	513091.0

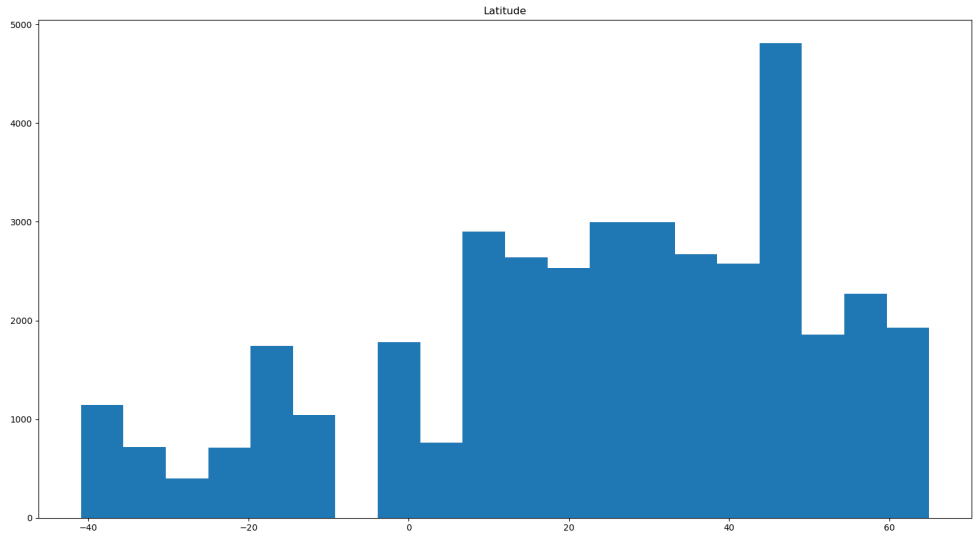
Πίνακας 1: Στατιστικά Δεδομένα για κάθε Στήλη του Dataset

Οι τιμές αυτές φαίνονται και στο αρχείο "describe.csv" που συμπεριλαμβάνεται στον φάκελο "Report".

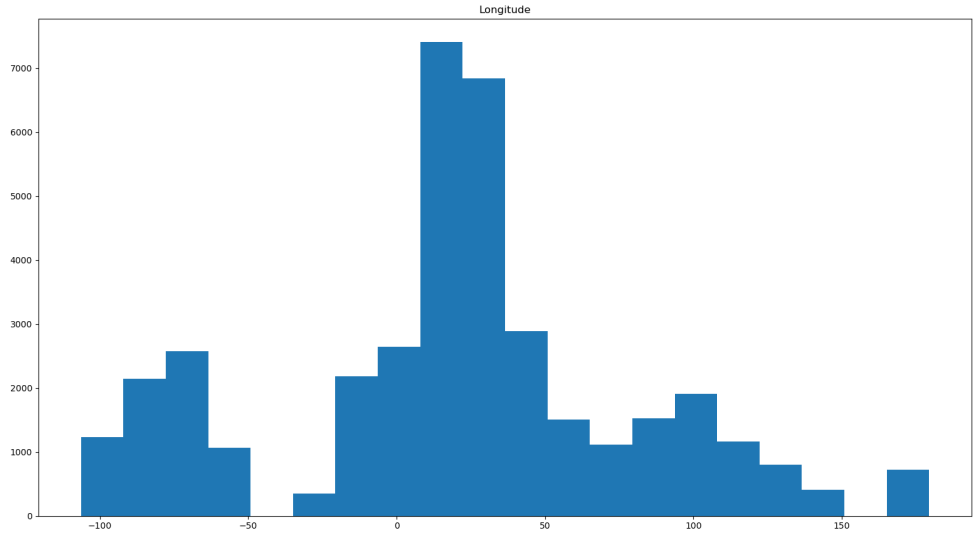
Παρακάτω φαίνονται επίσης τα ιστογράμματα της κάθε στήλης των δεδομένων του dataset που φτιάξαμε:



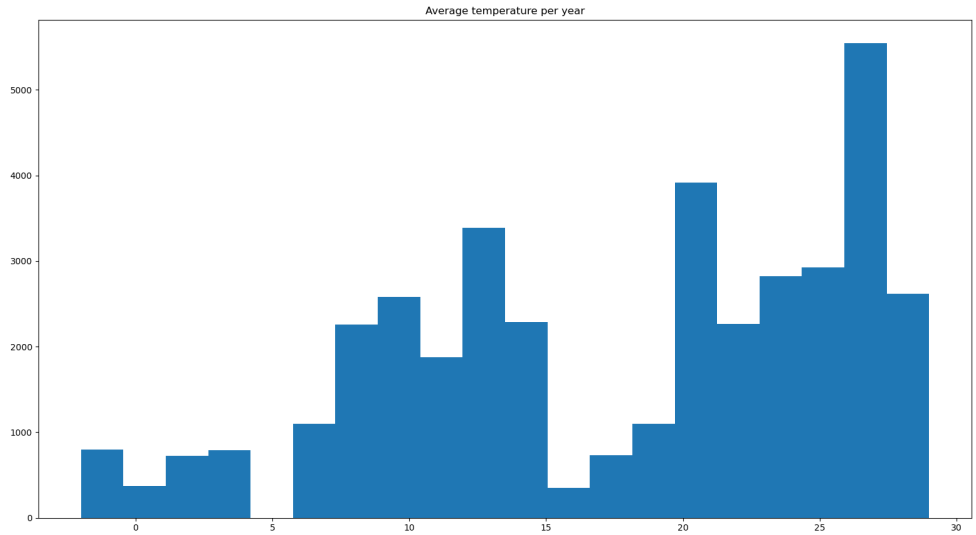
Σχήμα 1: Ιστόγραμμα για στήλη ’Continent’



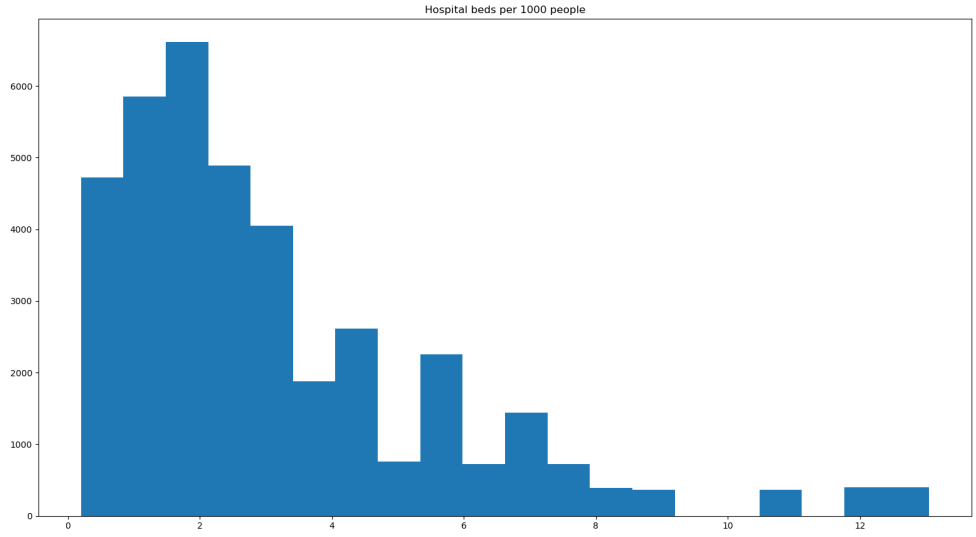
Σχήμα 2: Ιστόγραμμα για στήλη ’Latitude’



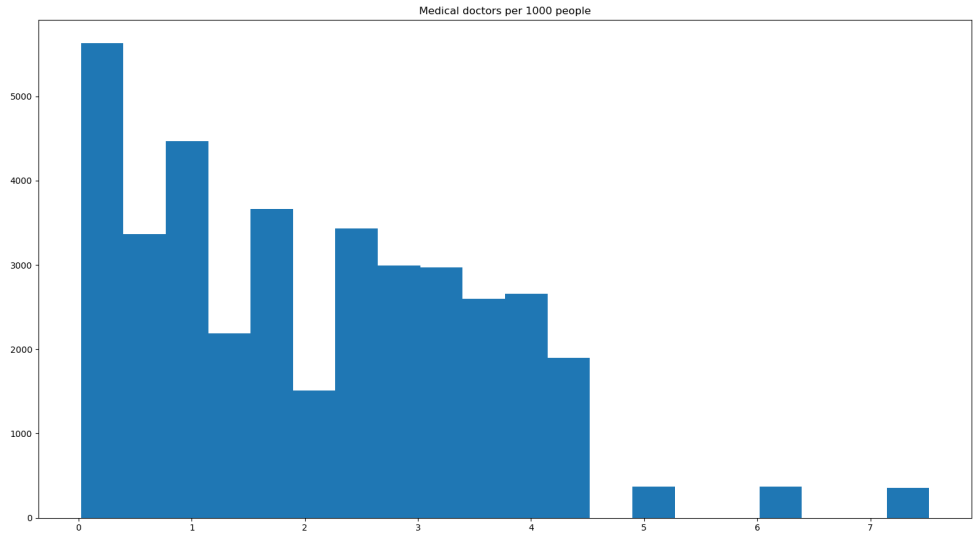
Σχήμα 3: Ιστόγραμμα για στήλη ’Longtitude’



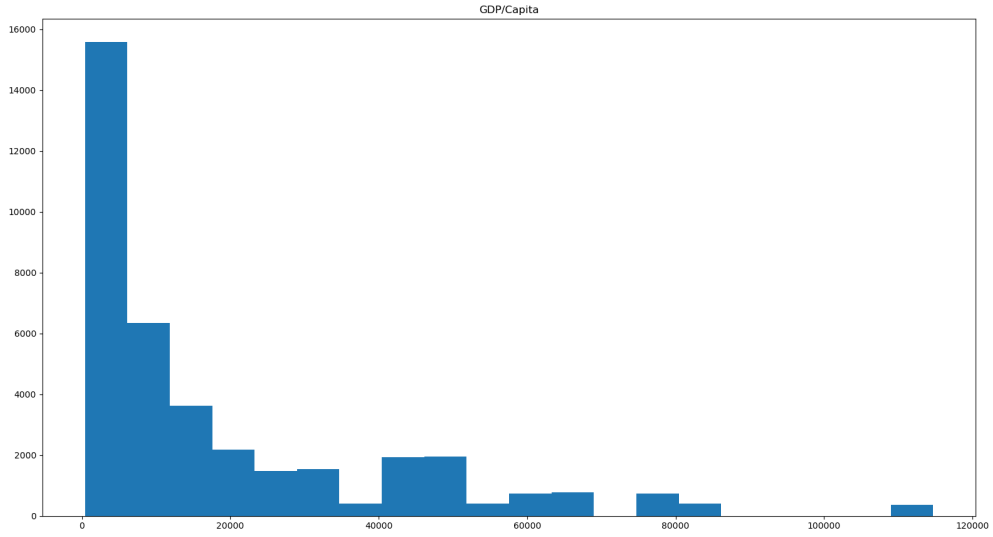
Σχήμα 4: Ιστόγραμμα για στήλη ’Temperature’



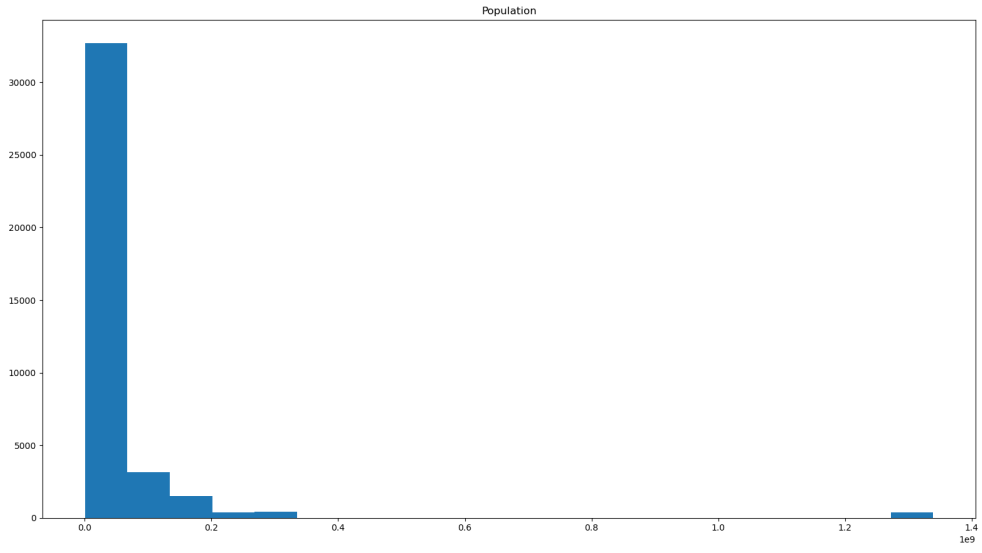
Σχήμα 5: Ιστόγραμμα για στήλη ’Hospital beds per 1000 people’



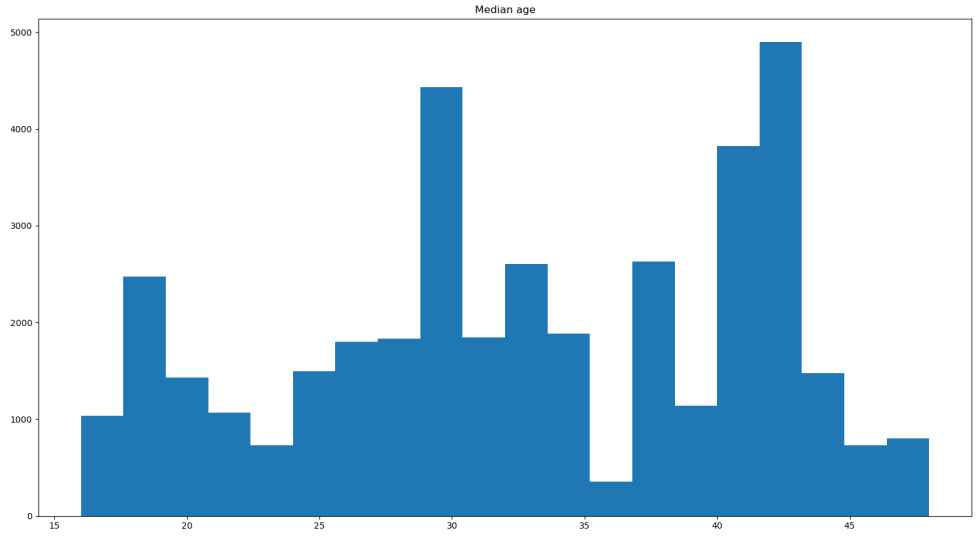
Σχήμα 6: Ιστόγραμμα για στήλη ’Medical doctors per 1000 people’



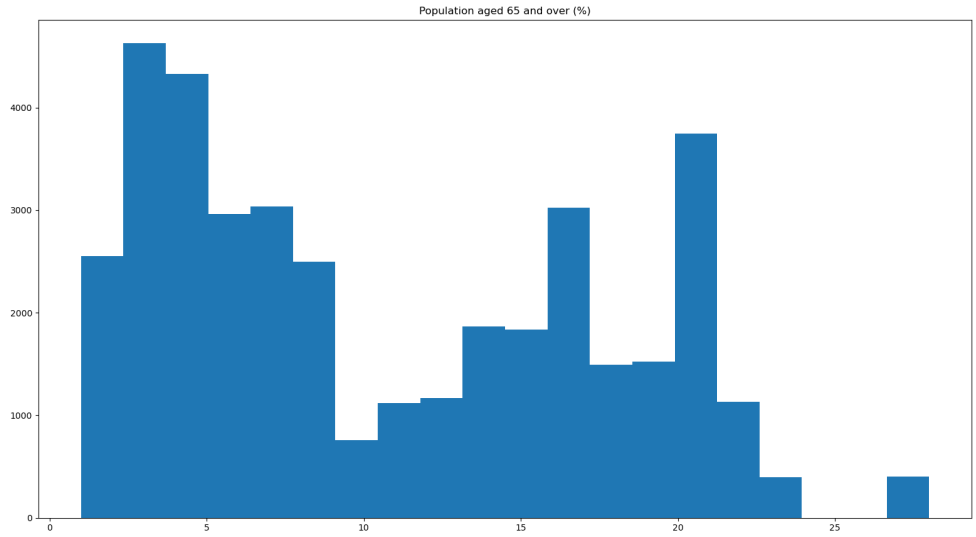
Σχήμα 7: Ιστόγραμμα για στήλη ’GDP/Capita’



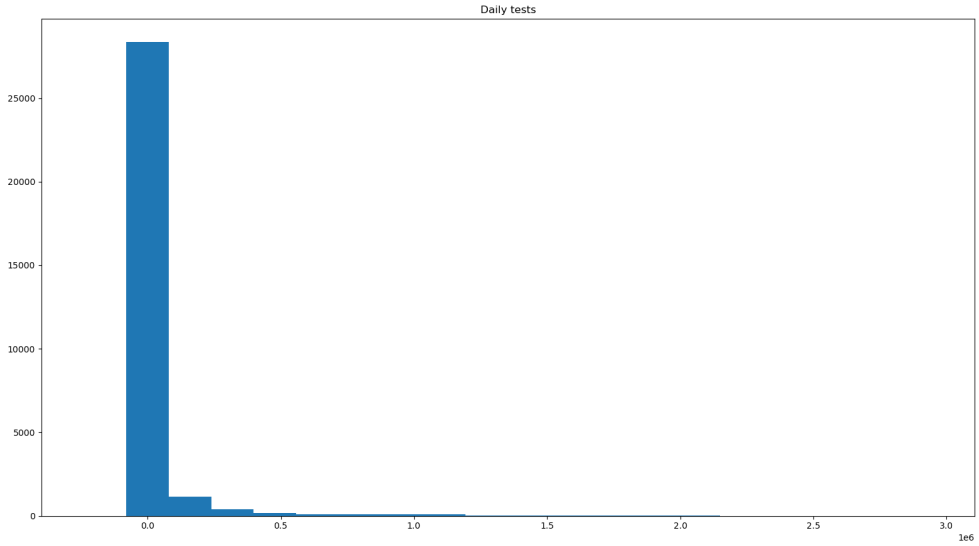
Σχήμα 8: Ιστόγραμμα για στήλη ’Population’



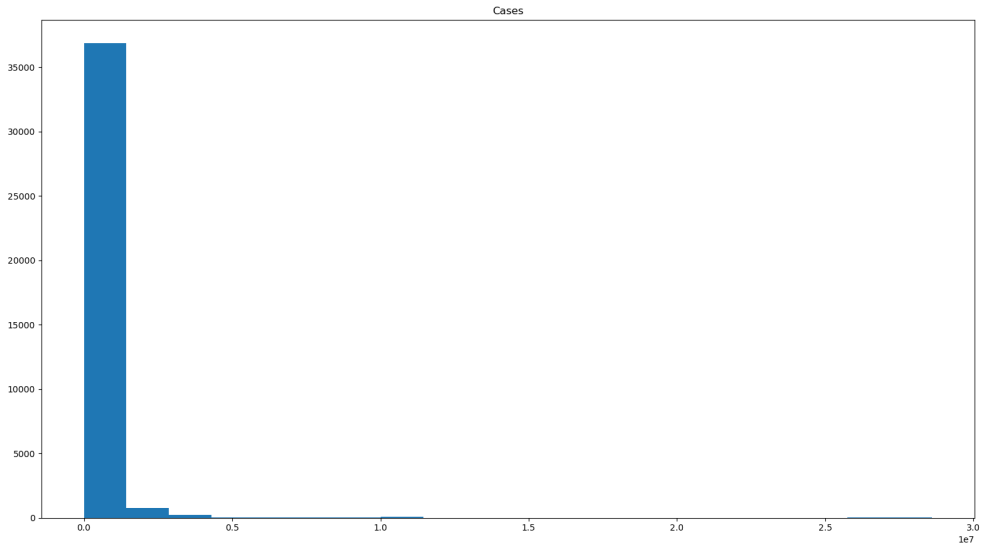
Σχήμα 9: Ιστόγραμμα για στήλη ’Median age’



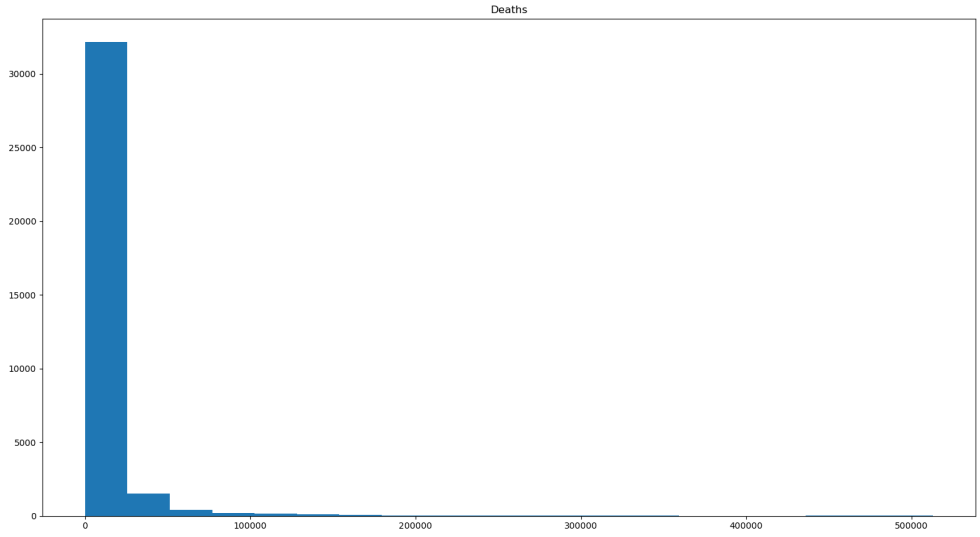
Σχήμα 10: Ιστόγραμμα για στήλη ’Population aged 65 and over (%)’



Σχήμα 11: Ιστόγραμμα για στήλη ’Daily tests’

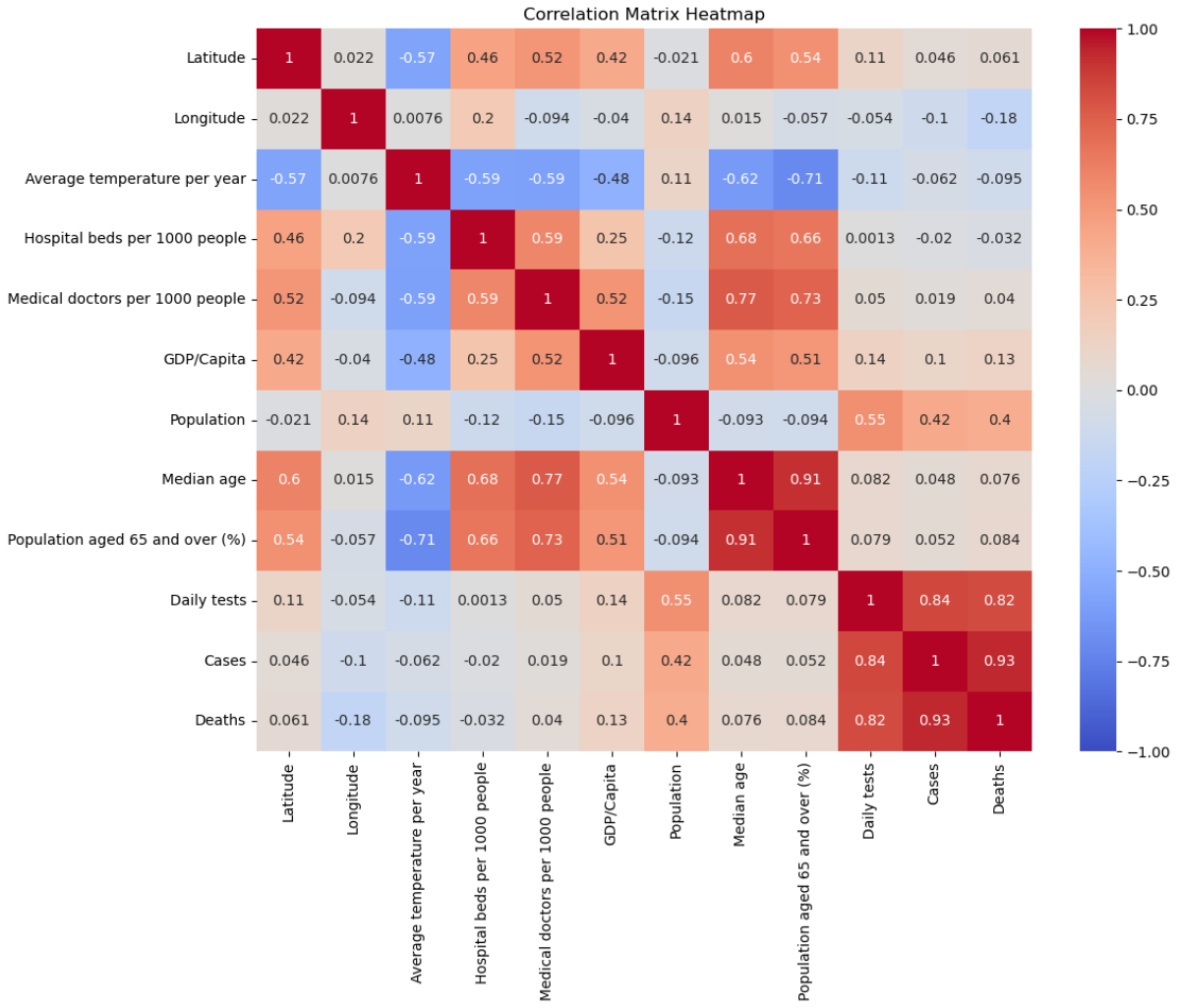


Σχήμα 12: Ιστόγραμμα για στήλη ’Cases’



Σχήμα 13: Ιστόγραμμα για στήλη ’Deaths’

Τέλος παρακάτω παρουσιάζουμε το Correlation Matrix Heatmap που φτιάξαμε για τις στήλες του dataset.



Σχήμα 14: Correlation Matrix Heatmap για τις στήλες

1.2.2 Συμπεράσματα

Από τα στατιστικά στοιχεία και τα ιστογράμματα συμπεραίνουμε πως πολλές από τις στήλες περιέχουν στοιχεία που φαίνεται να αποτελούν half-normal κατανομές, με διαφορετικές τιμές μέσης τιμής και διασποράς. Οι στήλες αυτές είναι οι 'Medical doctors per 1000 people', 'GDP/Capita', 'Population', 'Daily tests', 'Cases', 'Deaths'. Επίσης η στήλη 'Hospital beds per 1000 people' φαίνεται να ακολουθεί log-normal κατανομή. Για τις υπόλοιπες στήλες δεν μπορούμε να συμπεράνουμε ότι ανήκουν σε κάποια κατανομή.

Από το Correlation Matrix Heatmap παρατηρούμε ότι υπάρχει μεγάλη συσχέτιση μεταξύ 'Daily tests', 'Cases' και 'Deaths', καθώς και μεταξύ 'Median age', 'People aged 65 and over (%)', 'Hospital beds per 1000 people' και 'Medical doctors per 1000 people'. Υπάρχει επίσης μεγάλη αρνητική συσχέτιση μεταξύ του 'Average Temperature per year' σε σχέση με τα 'Population aged 65 and over (%)', 'Median age', 'Medical doctors per 1000 people', 'Hospital beds per 1000 people' και 'Latitude'. Σημειώνουμε ότι υπάρχουν και άλλες συσχετίσεις εκτός από αυτές που αναφέρονται, αλλά δεν είναι

τόσο ισχυρές όσο αυτές που αναφέρουμε εδώ, οπότε τις παραλείπουμε για λόγους συντομίας, εφόσον φαίνονται και στο heatmap.

2 Υλοποίηση και Αποτελέσματα Ερωτήματος 2

2.1 Σύντομη Περιγραφή της Διαδικασίας Υλοποίησης

2.1.1 Χειρισμός Τιμών που Λείπουν

Για την υλοποίηση του ερωτήματος αυτού, αρχικά αφού διαβάσουμε το αρχείο του dataset θα πρέπει να αντιμετωπίσουμε τις τιμές που λείπουν από αυτό. Αυτό το επιτυγχάνουμε με τη χρήση των εντολών:

```
df = df.groupby('Entity',group_keys=False).apply(lambda x: x.fillna(method='ffill'))
df = df.groupby('Entity',group_keys=False).apply(lambda x: x.fillna(method='bfill'))
```

Αυτές οι εντολές επιτυγχάνουν αρχικά το grouping του DataFrame με βάση το 'Entity' (χώρα) και μετά την εφαρμογή της fillna μεθόδου, η οποία συμπληρώνει τις τιμές που λείπουν αρχικά με χρήση forward fill στη πρώτη εντολή και έπειτα με backward fill στη δεύτερη εντολή.

Το forward fill επιτυγχάνει την αντικατάσταση τιμών που λείπουν με την τελευταία προηγούμενη καταγεγραμμένη τιμή, αλλά επειδή γίνεται να υπάρχουν ακόμα κενά (πχ αν λείπουν τιμές στην αρχή των καταγεγραμμένων στοιχείων), κάνουμε έπειτα το backward fill που συμπληρώνει τιμές που λείπουν με την επόμενη καταγεγραμμένη τιμή.

Ο λόγος που προτιμούμε πρώτα να κάνουμε forward fill και μετά backward fill είναι επειδή με αυτόν τον τρόπο ελαχιστοποιούμε την συμπλήρωση με βάση τις μελλοντικές ως προς τη χρονική στιγμή που συμπληρώνουμε τιμές.

Έπειτα αφαιρούμε όλα τα duplicates που τυχόν προέκυψαν με χρήση της μεθόδου drop_duplicates() του Pandas.

2.1.2 Δημιουργία Νέων Πεδίων για το Clustering

Αποφασίσαμε για το πιο αποτελεσματικό clustering να προσθέσουμε ορισμένα πεδία στα δεδομένα τα οποία εξάγονται από τα υπόλοιπα δεδομένα. Πιο συγκεκριμένα τα νέα πεδία αυτά τα ορίζουμε ως εξής:

- Positive Ratio = Today's New Cases / Daily Tests
- Death Ratio = Today's Deaths / New Cases
- Tested Ratio = Daily Tests / Population

2.1.3 Προεπεξεργασία Δεδομένων

Το πρώτο βήμα της προεπεξεργασίας θα είναι να κάνουμε aggregation των δεδομένων. Έτσι αντί να αποθηκεύουμε για κάθε χώρα τα δεδομένα κάθε μέρας ξεχωριστά, θα αποθηκεύσουμε είτε τον μέσο όρο είτε την τελευταία τιμή μιας στήλης δεδομένων κάθε φορά. Αυτό μας διευκολύνει στο clustering, επειδή μειώνει τον όγκο των δεδομένων εισόδου και ταυτόχρονα απομονώνει την σημαντική πληροφορία από τα δεδομένα για όλες τις ημέρες. Έτσι θα αποθηκεύσουμε τη μέση τιμή των 'Positive Ratio', 'Death Ratio', και 'Tested Ratio' και θα αποθηκεύσουμε και την τελευταία τιμή που εμφανίζεται για όλα τα υπόλοιπα Series.

Επειδή το 'Continent' περιέχει κατηγορικά δεδομένα, αποφασίζουμε να τα μετατρέψουμε σε αριθμητικά δεδομένα ώστε να μπορέσουμε να κάνουμε κανονικοποίηση αργότερα πάνω σε αυτά. Άρα κάνουμε one-hot encoding για όλο το Series 'Continent' πάνω στο DataFrame που προέκυψε από το προηγούμενο βήμα.

Έπειτα αφαιρούμε το Series 'Date' από το DataFrame των προηγούμενων βημάτων, επειδή αυτό είναι ίδιο σε κάθε περίπτωση (ίσο με την τελευταία ημερομηνία του dataset).

Τέλος κάνουμε κανονικοποίηση στα δεδομένα (με χρήση αντικειμένου StandardScaler της βιβλιοθήκης Scikit-learn) και αποθηκεύουμε τα τελικά δεδομένα στη μεταβλητή normalized_data.

2.1.4 Δημιουργία Δενδρογράμματος και Συσταδοποίηση