Εξόρυξη Δεδομένων και Αλγόριθμοι Μάθησης

Υλοποιητικό Project 2022 – 23

Μαρία Σίμωση – 1072659
Αλέξανδρος Τσαπάρας – 1072824

Βιβλιοθήκες

Οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι **pandas, numpy, seaborn, spicy, matplotlib, yellowbrick, tensorflow, sklearn, keras, geopandas.**

Οδηγίες Εγκατάστασης

Για την εγκατάσταση τους, αρχική προϋπόθεση είναι η εγκατάσταση της έκδοσης 3.11 της Python και του pip. Έπειτα ο χρήστης πρέπει να τρέξει τις εξής εντολές στο terminal/command prompt, που ισχύουν για Windows/Mac/Linux:

- pip install pandas
- pip install numpy
- pip install seaborn
- pip install spicy
- pip install matplotlib
- pip install tensorflow
- pip install yellowbrick
- pip install scikit-learn
- pip install keras
- pip install geopandas

Ερώτημα 1ο

- **Προεπεξεργασία**

Για την αρχική ανάλυση των δεδομένων, φορτώνουμε το data frame και τυπώνουμε τα βασικά στατιστικά μεγέθη κάθε στήλης.

	Latitude	Longitude	Average temperature per year	Hospital beds per 1000 people	...	Population aged 65 and over (%)	Daily tests	Cases	Deaths
count	38472.000000	38472.000000	38472.000000	38472.000000	...	38472.000000	3.057700e+04	3.821800e+04	34862.000000
mean	23.737060	20.210847	17.717145	3.165305	...	10.663392	3.944059e+04	2.879027e+05	8090.504733
std	26.060413	61.074922	8.134926	2.561767	...	6.770417	1.501847e+05	1.405243e+06	29548.753221
min	-40.900000	-106.350000	-2.000000	0.200000	...	1.000000	-2.391720e+05	1.000000e+00	1.000000
25%	8.620000	-3.440000	11.000000	1.400000	...	5.000000	1.505000e+03	2.074000e+03	77.000000
50%	27.510000	21.820000	20.000000	2.500000	...	8.000000	5.520000e+03	2.143100e+04	527.000000
75%	45.940000	47.480000	25.000000	4.490000	...	16.000000	2.038200e+04	1.373770e+05	3480.500000
max	64.960000	179.410000	29.000000	13.050000	...	28.000000	2.945871e+06	2.860567e+07	513091.000000

Παρατηρούμε ότι υπάρχουν αρνητικές τιμές στη στήλη Daily tests. Οι τιμές αυτές δεν έχουν νόημα, οπότε αφαιρούμε τις αντίστοιχες γραμμές. Έπειτα, εντοπίζουμε τις στήλες με missing values.

```
Daily tests                          7895
Deaths                               3610
Cases                                 254
Entity                                  0
Continent                               0
Latitude                                0
Longitude                               0
Average temperature per year            0
Hospital beds per 1000 people           0
Medical doctors per 1000 people         0
GDP/Capita                              0
Population                              0
Median age                              0
Population aged 65 and over (%)         0
Date                                    0
dtype: int64
```

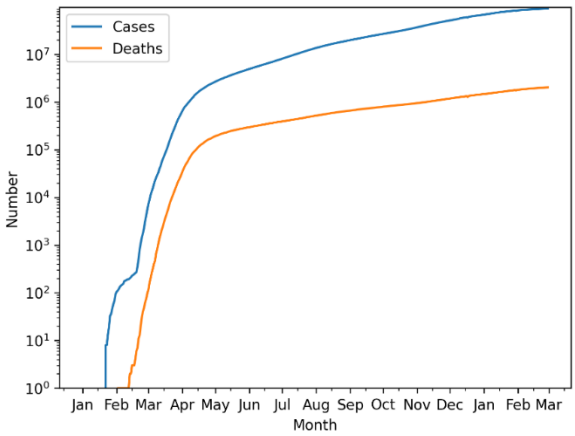
Υπάρχουν χώρες με ελάχιστες καταχωρήσεις στη στήλη Daily tests που θα δυσκολέψουν την ανάλυσή μας. Επιλέγουμε ως κατώφλι το 75% των συνολικών γραμμών της χώρας και αφαιρούμε από τα δεδομένα μας όσες χώρες έχουν ακόμα μεγαλύτερο ποσοστό missing values στη στήλη αυτή, δηλαδή τις εξής 7:

```
Index(['Algeria', 'Armenia', 'Belarus', 'Madagascar', 'Mauritania', 'Oman',
       'Vietnam'],
      dtype='object', name='Entity')
```

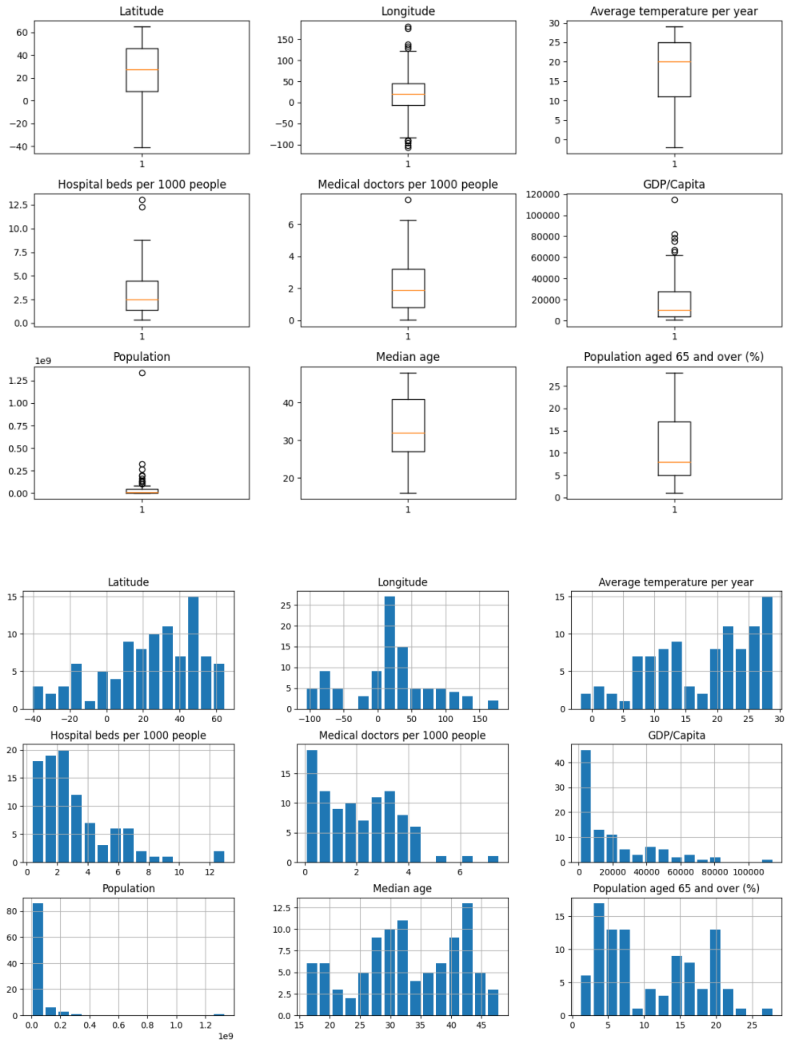
Τα υπόλοιπα Daily tests τα γεμίζουμε με τις μεθόδους ffill και bfill, που συμπληρώνουν το κενό με την προηγούμενη και την επόμενη, αντίστοιχα, γνωστή τιμή της χώρας. Με μία σύντομη έρευνα, βρήκαμε ότι οι πρώτες υπαρκτές τιμές των Cases και Deaths αντιστοιχούν στις ημερομηνίες των πρώτων κρουσμάτων και πρώτων θανάτων της χώρας, οπότε γεμίσαμε τα κενά με 0. Αποθηκεύσαμε τα επεξεργασμένα δεδομένα ως modified_dataset για τη συνέχεια και τα χρησιμοποιήσαμε για τα διάφορα διαγράμματα.

- **Διαγράμματα**

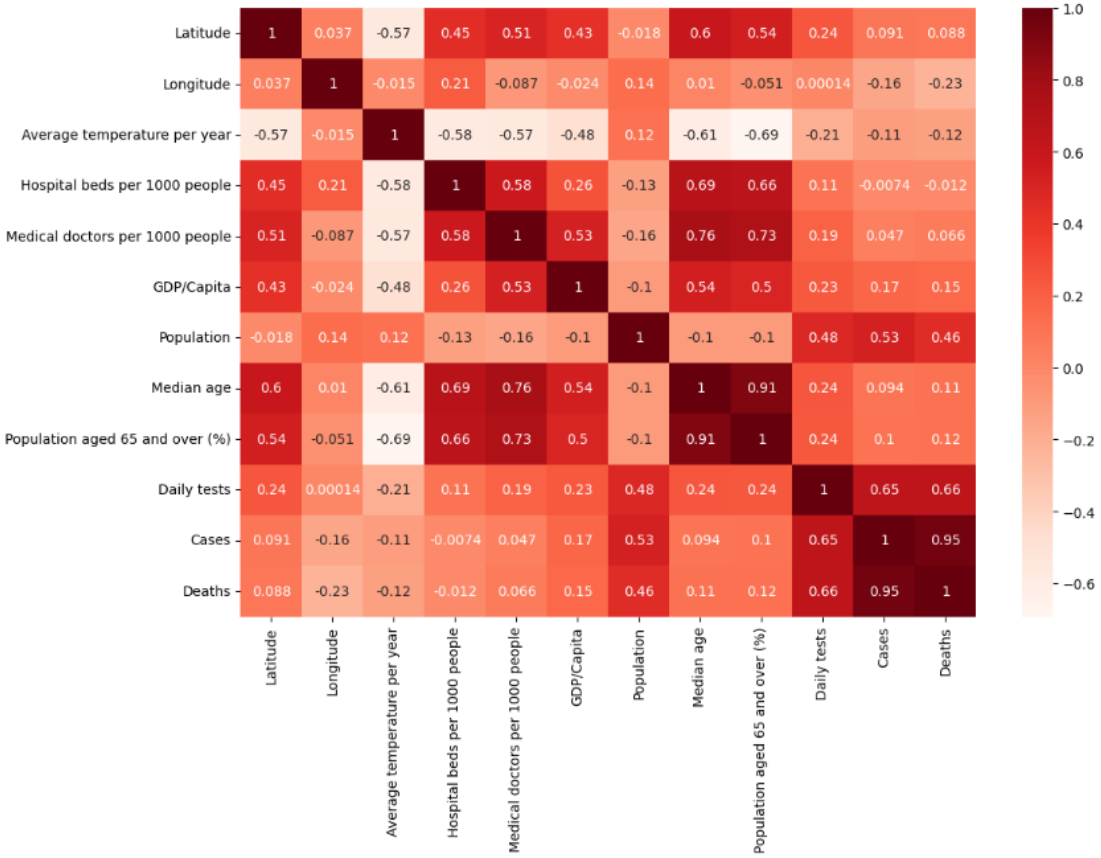
Σχεδιάζουμε τις καμπύλες των Cases και Deaths στον χρόνο και ομοιότητά τους είναι προφανής. Ο ρυθμός αύξησης των δύο μεγεθών μειώνεται αισθητά μετά τους πρώτους μήνες.



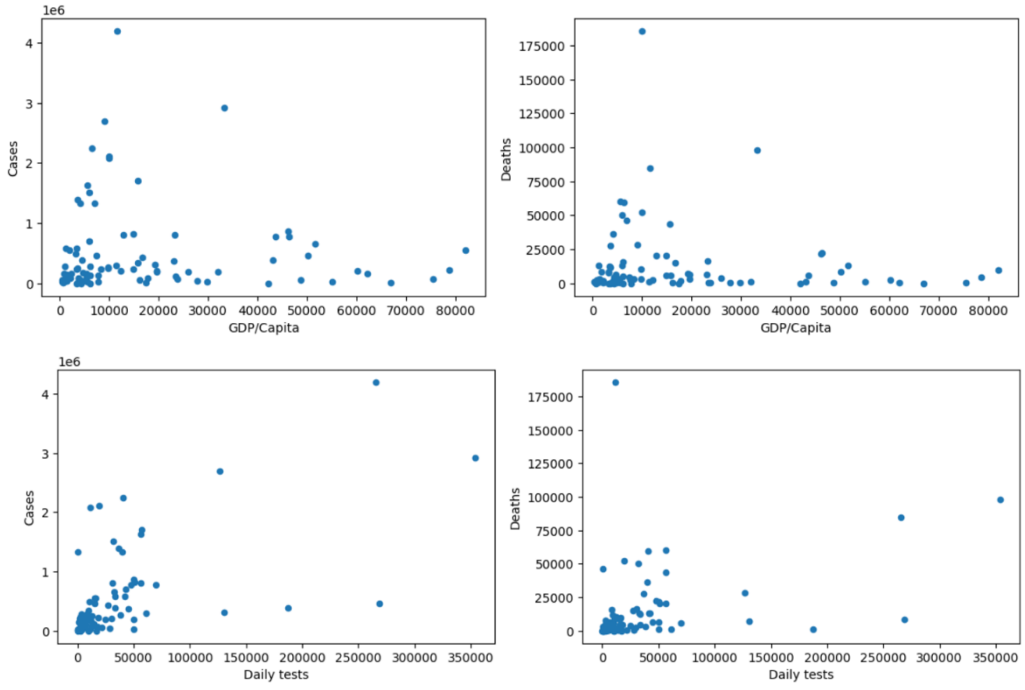
Με έναν συνδυασμό boxplot και ιστογράμματος για κάθε στήλη των δεδομένων που περιγράφει τα γενικά χαρακτηριστικά των χωρών βλέπουμε γρήγορα την κατανομή τους.

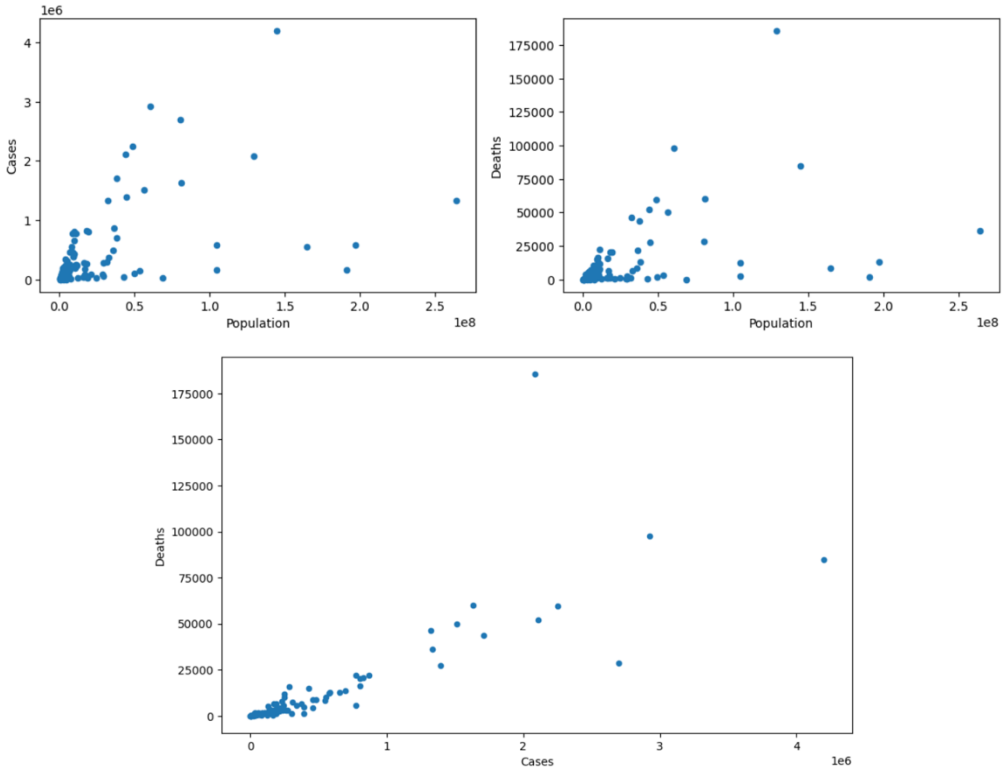


Χρησιμοποιώντας την τελευταία γραμμή (πιο πρόσφατη ημερομηνία) κάθε χώρας φτιάχνουμε ένα heatmap για να δούμε τη συσχέτιση μεταξύ των στηλών. Όπως ήταν αναμενόμενο, τα κρούσματα και οι θάνατοι παρουσιάζουν ισχυρή συσχέτιση. Το ίδιο ισχύει και για τις στήλες Population aged 65 and over(%) και Median age, πολύ λογικά. Αυτές οι δύο στήλες, μάλιστα, εμφανίζουν μεγάλη θετική συσχέτιση με την Hospital beds per 1000 people και ακόμα μεγαλύτερη με την Medical doctors per 1000 people. Το προσδόκιμο ζωής μιας χώρας, δηλαδή, εξαρτάται από το υγειονομικό της σύστημα. Τέλος, βλέπουμε αρκετά μεγάλες τιμές στα Daily tests-Cases και Daily tests-Deaths.



Τέλος, δημιουργούμε scatter plots των Cases και Deaths συναρτήσει κάθε στήλης. Για πιο ευανάγνωστα διαγράμματα, διαγράφουμε τις γραμμές όπου κάποιο μέγεθος έχει z-score > 3, δηλαδή περισσότερες από 3 standard αποκλίσεις από τη μέση τιμή. Βλέποντας μερικά από τα scatter plots, παρατηρούμε ότι τα 2 διαγράμματα για κάθε στήλη μοιάζουν αρκετά μεταξύ τους. Αυτό είναι λογικό, αφού έχουμε ήδη δει πως οι θάνατοι ακολουθούν τα κρούσματα. Σχεδιάζουμε, ακόμη, ένα scatter plot Deaths-Cases που επιβεβαιώνει άλλη μια φορά αυτή τη σχέση. Αξίζει, επίσης, να σημειωθεί ότι δεν υπάρχουν μεγάλες τιμές στα κρούσματα και τους θανάτους, όταν το GDP/Capita αυξάνεται σημαντικά.

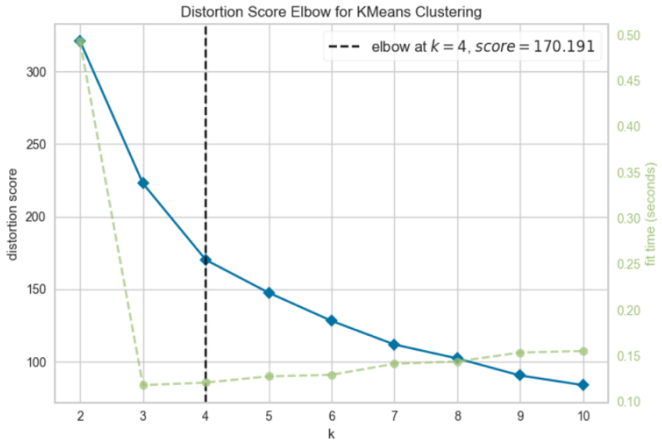




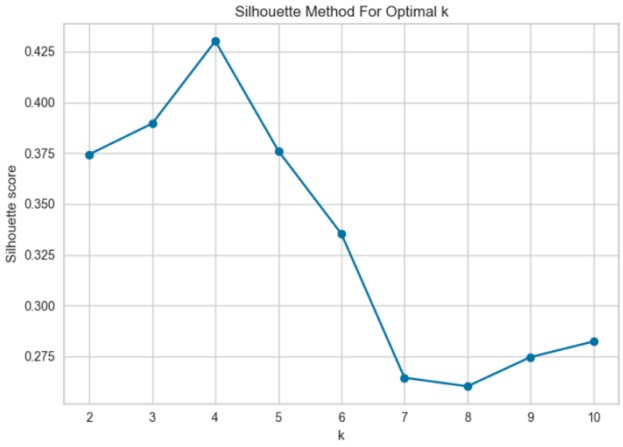
Ερώτημα 2ο

- **Clustering**

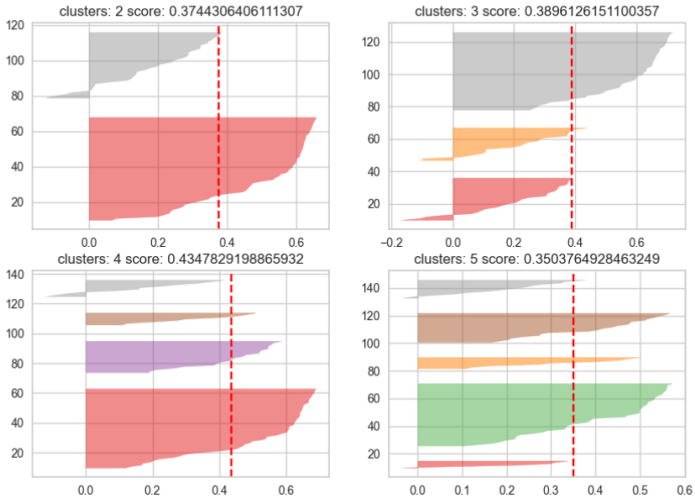
Για αυτή την ανάλυση φορτώνουμε απευθείας τα επεξεργασμένα δεδομένα. Από αυτά δημιουργούμε ένα νέο data frame που έχει μία γραμμή ανά χώρα και τις στήλες: Cases/tests, Deaths/cases, Tests/population, Cases/population, Deaths/population που είναι αντιπροσωπευτικές για την αντιμετώπιση του ιού. Κανονικοποιούμε τα δεδομένα και επιλέγουμε τον αλγόριθμο k means. Πρώτα, όμως, πρέπει να βρεθεί ο ιδανικός αριθμός συστάδων. Δοκιμάζουμε με Elbow method, που υπολογίζει για δεδομένο αριθμό συστάδων το άθροισμα των τετραγωνικών αποστάσεων όλων των σημείων μίας συστάδας από το κεντροειδές της.



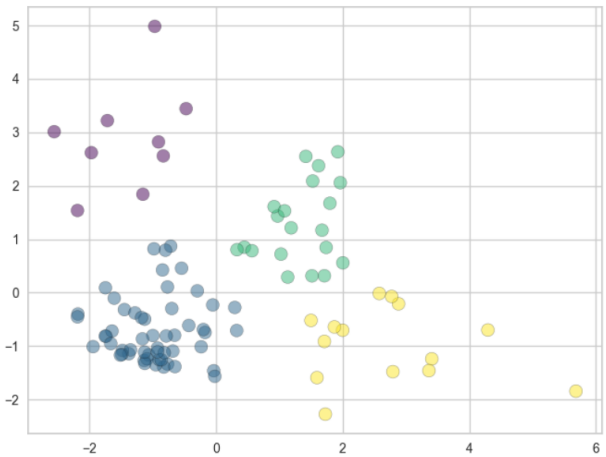
Οι βέλτιστες συστάδες είναι 4, αλλά το elbow point δεν είναι ιδιαίτερα σαφές. Επιβεβαιώνουμε με Silhouette method.



Το silhouette score ισούται με $\frac{b-a}{max(a,b)}$, όπου a η μέση απόσταση των σημείων της ίδιας συστάδας και b η μέση απόσταση όλων των συστάδων. Βλέπουμε και αναλυτικά τους συντελεστές από 2 ως 5 συστάδες.



Εφαρμόζουμε τον k means για 4 συστάδες. Έπειτα, με principal component analysis (PCA) απεικονίζουμε τα δεδομένα σε δύο διαστάσεις, διατηρώντας τη μέγιστη δυνατή πληροφορία, ώστε να δούμε τις συστάδες στον χώρο.

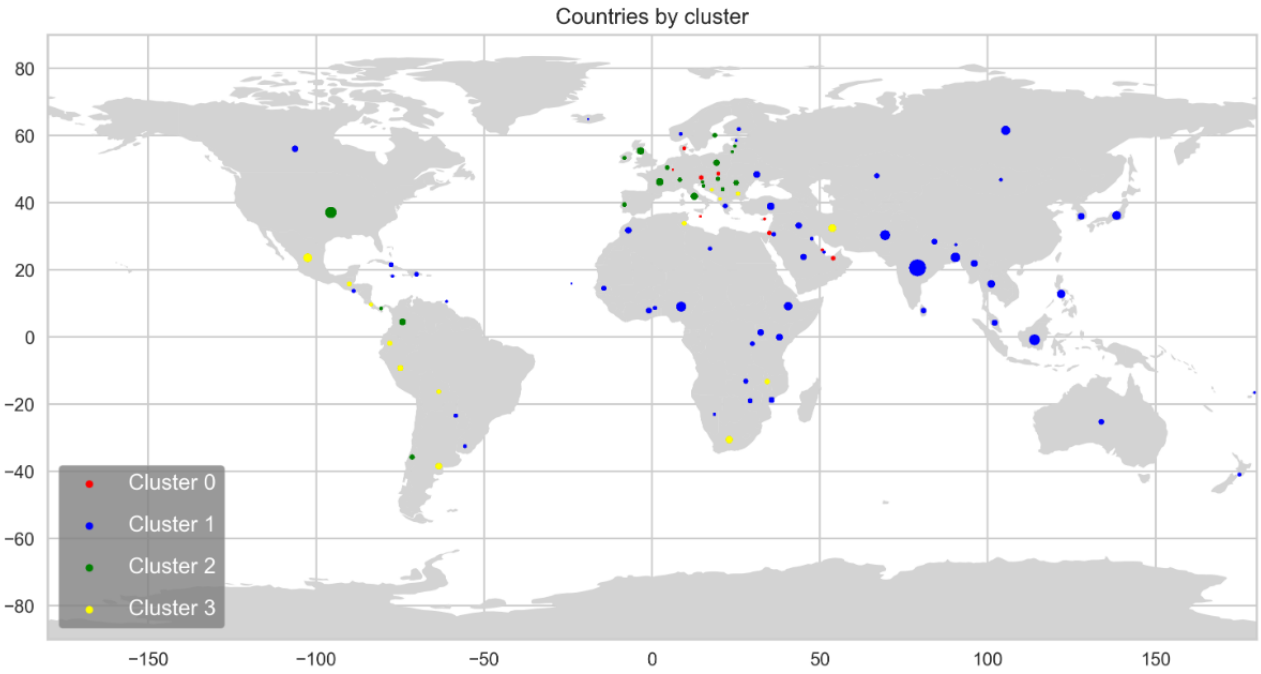


- **Ανάλυση αποτελεσμάτων**

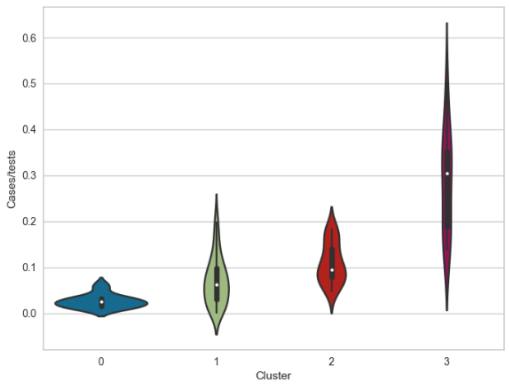
Τυπώνουμε τα ονόματα των χωρών που ανήκουν σε κάθε cluster.

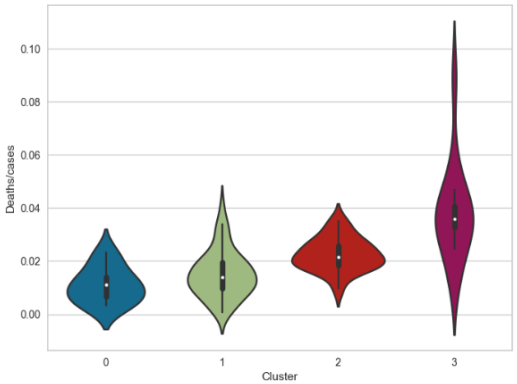
```
Cluster 0 countries:
['Austria' 'Bahrain' 'Cyprus' 'Denmark' 'Israel' 'Luxembourg' 'Malta'
 'Slovakia' 'United Arab Emirates']

Cluster 1 countries:
['Australia' 'Bangladesh' 'Bhutan' 'Canada' 'Cape Verde' 'Cuba'
 'Dominican Republic' 'El Salvador' 'Estonia' 'Ethiopia' 'Fiji' 'Finland'
 'Ghana' 'Greece' 'Iceland' 'India' 'Indonesia' 'Iraq' 'Jamaica' 'Japan'
 'Jordan' 'Kazakhstan' 'Kenya' 'Kuwait' 'Libya' 'Malaysia' 'Mongolia'
 'Morocco' 'Mozambique' 'Myanmar' 'Namibia' 'Nepal' 'New Zealand'
 'Nigeria' 'Norway' 'Pakistan' 'Paraguay' 'Philippines' 'Qatar' 'Russia'
 'Rwanda' 'Saudi Arabia' 'Senegal' 'South Korea' 'Sri Lanka' 'Thailand'
 'Togo' 'Trinidad and Tobago' 'Turkey' 'Uganda' 'Ukraine' 'Uruguay'
 'Zambia' 'Zimbabwe']

Cluster 2 countries:
['Belgium' 'Chile' 'Colombia' 'Croatia' 'France' 'Hungary' 'Ireland'
 'Italy' 'Latvia' 'Lithuania' 'Panama' 'Poland' 'Portugal' 'Romania'
 'Serbia' 'Slovenia' 'Sweden' 'Switzerland' 'United Kingdom'
 'United States']

Cluster 3 countries:
['Albania' 'Argentina' 'Bolivia' 'Bosnia and Herzegovina' 'Bulgaria'
 'Costa Rica' 'Ecuador' 'Guatemala' 'Iran' 'Malawi' 'Mexico' 'Peru'
 'South Africa' 'Tunisia']
```

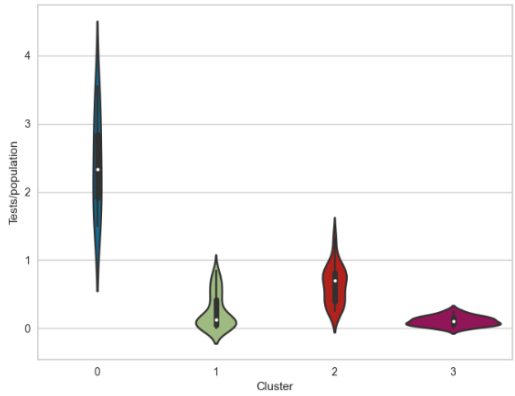
Και απεικονίζουμε τα clusters στον παγκόσμιο χάρτη.



Κοιτάζοντας τα violin plots, παρατηρούμε ότι το cluster 0 παρουσιάζει τα χαμηλότερα ποσοστά θετικότητας και θνησιμότητας, ενώ γίνονται πολλά Tests/Population. Ακριβώς το αντίθετο ισχύει για το cluster 3. Αυτές οι χώρες, δηλαδή, μπορούν να θεωρηθούν οι περισσότερο και λιγότερο πετυχημένες, αντίστοιχα, στην αντιμετώπιση της πανδημίας.





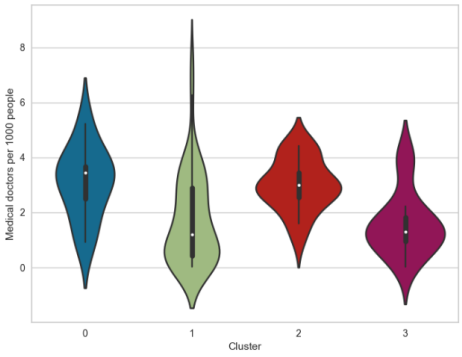


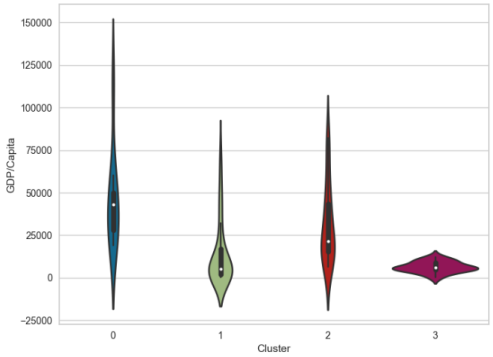
Στο διάγραμμα Deaths/cases παρατηρούμε το cluster 3 να φτάνει πολύ υψηλές τιμές. Τυπώνουμε τις στήλες του cluster και εντοπίζουμε αυτήν την τιμή στο Μεξικό, όπου το ποσοστό θετικότητας είναι 42%, το ποσοστό θνησιμότητας πλησιάζει το 9%, ενώ ο αριθμός των τεστ δεν φτάνει καν το 4% του πληθυσμού της χώρας.

```
Mexico      0.417882      0.088989          0.038665
```

Βλέπουμε στον χάρτη πως το cluster 3 αποτελείται κυρίως από χώρες της Νότιας Αμερικής, της Αφρικής και της Ανατολικής Ευρώπης, ενώ το cluster 0 περιορίζεται σε χώρες της Δυτικής Ευρώπης και Ασίας.

Επίσης, από violin plots για τα γενικά χαρακτηριστικά των χωρών, προκύπτει πως το cluster 0 υπερέχει σε ιατρικό προσωπικό και GDP/Capita, ενώ το cluster 3 υστερεί.





Τέλος, το cluster 0 έχει σημαντικά μικρότερο μέσο πληθυσμό σε σχέση με τις υπόλοιπες ομάδες.

```
Population mean value for cluster 0: 4652014.222222222
Population mean value for cluster 1: 65323146.88888889
Population mean value for cluster 2: 36048514.1
Population mean value for cluster 3: 31184726.57142857
```

Ερώτημα 3ο

Πρόβλεψη ποσοστού θετικότητας στην Ελλάδα 3 ημέρες μετά

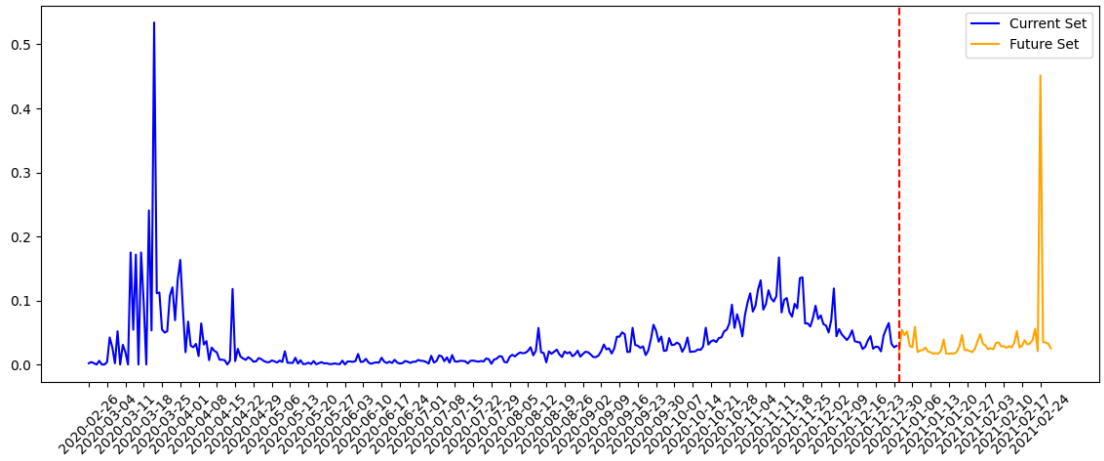
Αρχικά φορτώνουμε στο πρόγραμμα μας όλες τις συναρτήσεις που θα χρησιμοποιήσουμε από τις βιβλιοθήκες που αναφέραμε και ξεκινάμε με την υλοποίηση.

Πρώτο μας βήμα είναι να θέσουμε «σπόρους» (seeds) για να έχουμε το ίδιο αποτέλεσμα κάθε φορά που τρέχουμε το πρόγραμμα. Έπειτα ανοίγουμε το προ-επεξεργασμένο csv αρχείο με τα δεδομένα μας

(modified_dataframe.csv) και τα αποθηκεύουμε σε ένα dataframe, από το οποίο κρατάμε μόνο τις στήλες/σειρές για την Ελλάδα. Δημιουργούμε μια νέα στήλη "Positivity Rate" και υπολογίζουμε το ποσοστό θετικότητας ανά μέρα.

- **Διαχωρισμός τωρινών/μελλοντικών δεδομένων**

Αρχικά πρέπει να χωρίσουμε τα δεδομένα σε current_set και future_set, ανάλογα με αν βρίσκονται πριν ή μετά τη 1/1/2021, για να τα χρησιμοποιήσουμε και στα δύο μοντέλα μας. Στο παρακάτω σχήμα φαίνεται το ποσοστό θετικότητας της Ελλάδας.



Έπειτα θα χρησιμοποιήσουμε το current_set για την εκπαίδευση του μοντέλου και το future_set για την αξιολόγηση του.

- **Train/Test Split**

Αρχικά βρίσκουμε το ποσοστό θετικότητας τρεις μέρες μετά και το προσθέτουμε σε μια νέα στήλη, αφαιρώντας όποιες γραμμές έχουν NaN τιμές. Έπειτα χωρίζουμε σε X και y τα δεδομένα, για τα χαρακτηριστικά και τις τιμές-στόχους αντίστοιχα. Αμέσως μετά κάνουμε κλιμάκωση των δεδομένων χρησιμοποιώντας τον MinMaxScaler, ο οποίος μέσω της συνάρτησης fit_transform() υπολογίζει τις ελάχιστες και μέγιστες τιμές της εισόδου και, στη συνέχεια, κλιμακώνει τις τιμές σε ένα εύρος μεταξύ 0 και 1. Τέλος χρησιμοποιούμε την συνάρτηση train_test_split(), για να χωρίσουμε τα δεδομένα σε trainX και testX για τα χαρακτηριστικά και trainY και testY για τις τιμές-στόχους. Το σετ εκπαίδευσης (train) θα περιέχει το 80% των αρχικών δεδομένων, ενώ το σετ δοκιμών (test) θα περιέχει το 20%. Οι συγκεκριμένες περιπτώσεις σε κάθε σετ καθορίζονται τυχαία, αλλά ο διαχωρισμός θα είναι συνεπής λόγω της ρύθμισης της κατάστασης random_state=0.

- **Εκπαίδευση**

– Για το RNN:

Στα τεχνητά νευρωνικά δίκτυα, οι νευρώνες είναι θεμελιώδεις μονάδες που λαμβάνουν σήματα εισόδου, τα επεξεργάζονται και παράγουν ένα σήμα εξόδου. Είναι οργανωμένα σε επίπεδα, σχηματίζοντας μια αρχιτεκτονική δικτύου.

Στο πλαίσιο του LSTM, το οποίο είναι ένας τύπος επαναλαμβανόμενου νευρωνικού δικτύου (RNN), οι νευρώνες είναι εξοπλισμένοι με πρόσθετους μηχανισμούς για την καταγραφή διαδοχικών ή χρονικών εξαρτήσεων στα δεδομένα εισόδου. Τα LSTM έχουν μια πιο περίπλοκη εσωτερική δομή, συμπεριλαμβανομένων κυψελών μνήμης και πυλών, που τους επιτρέπουν να διατηρούν και να επεξεργάζονται πληροφορίες σε μεγαλύτερες ακολουθίες

Υλοποίηση:

Χρησιμοποιούμε την συνάρτηση Sequential, για να φτιάξουμε ένα στιγμιότυπο του Διαδοχικού μοντέλου, το οποίο είναι μια γραμμική στοίβα στρωμάτων στο Keras. Έπειτα ορίζουμε ένα επίπεδο κάνοντας χρήση του μοντέλου LSTM, στο οποίο αναφέρουμε το μέγεθος της εισόδου (input_shape) και τον αριθμό των νευρώνων (10), από τους οποίους αποτελείται το επίπεδο. Παρατηρήσαμε επίσης πως η προσθήκη ενός ακόμα επιπέδου αυξάνει το μέσο τετραγωνικό σφάλμα. Το τελευταίο χαρακτηριστικό του μοντέλου είναι ο ένας νευρώνας που δημιουργούμε μέσω της Dense(), για να ορίσουμε την τιμή εξόδου. Έπειτα συντάσσουμε το μοντέλο κάνοντας χρήση ενός δημοφιλούς αλγορίθμου βελτιστοποίησης για την εκπαίδευση νευρωνικών δικτύων, τον "adam" και θέτοντας την παράμετρο απώλεια σε 'mean_squared_error', υποδεικνύοντας ότι το μέσο τετραγωνικό σφάλμα θα χρησιμοποιηθεί ως η συνάρτηση απώλειας για τη μέτρηση της διαφοράς μεταξύ των προβλεπόμενων και των πραγματικών τιμών. Τέλος εκπαιδεύουμε το μοντέλο χρησιμοποιώντας τα trainX και trainY, με παραμέτρους epochs=10 και batch_size=1. Η παράμετρος epochs καθορίζει πόσες φορές θα επαναληφθούν τα δεδομένα εκπαίδευσης κατά τη διάρκεια της εκπαίδευσης, όπου στην δικιά μας περίπτωση παρατηρούμε ότι για περισσότερες εποχές γίνεται overfitting. Η παράμετρος batch_size καθορίζει τον αριθμό των δειγμάτων που θα χρησιμοποιηθούν σε κάθε ενημέρωση των βαρών του μοντέλου. Η επιλογή των παραμέτρων έγινε βασιζόμενη στο μικρότερο δυνατό μέσο τετραγωνικό σφάλμα.

Έπειτα χρησιμοποιούμε το μοντέλο, για να κάνουμε τις "προβλέψεις", μέσω της συνάρτησης predict(), σχετικά με τα χαρακτηριστικά του συνόλου δοκιμών (testX). Τέλος υπολογίζουμε το μέσο τετραγωνικό σφάλμα μεταξύ των προβλέψεων και των πραγματικών τιμών ($mse \approx 0.01994$).

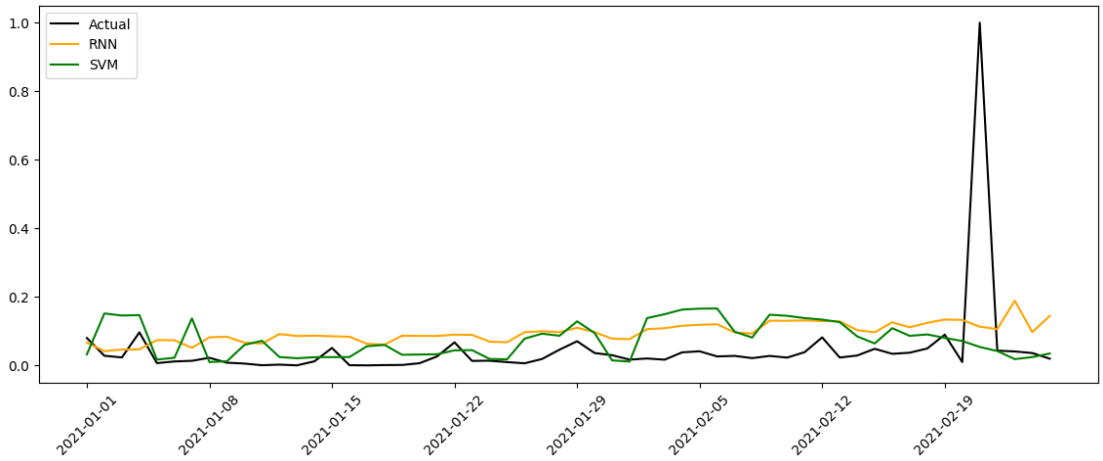
– Για το SVM:

Χρησιμοποιούμε την συνάρτηση svm.SVR(), για να φτιάξουμε ένα στιγμιότυπο του SVR μοντέλου. Το μοντέλο SVR είναι μια παραλλαγή του SVM που χρησιμοποιείται για παλινδρόμηση. Η παράμετρος του kernel έχει οριστεί σε "rbf", που σημαίνει συνάρτηση ακτινικής βάσης, υποδεικνύοντας ότι το μοντέλο θα χρησιμοποιήσει τον πυρήνα RBF (Radial Basis Function). Η παράμετρος C έχει οριστεί σε 400, η οποία ελέγχει την αντιστάθμιση μεταξύ της μεγιστοποίησης του περιθωρίου και της ελαχιστοποίησης του σφάλματος. Η παράμετρος γάμμα (gamma) έχει οριστεί σε «κλίμακα», η οποία προσδιορίζει ότι η κλίμακα του πυρήνα RBF θα υπολογιστεί αυτόματα με βάση τα δεδομένα εισόδου. Η παράμετρος epsilon έχει οριστεί στο 0.15, η οποία ελέγχει το πλάτος του epsilon-tube γύρω από τις προβλεπόμενες τιμές, η οποία αν έχει μεγαλύτερη τιμή του 0.15 αυτό σημαίνει ότι επιτρέπεται μεγαλύτερο περιθώριο για να θεωρηθούν αποδεκτά τα σφάλματα. Η επιλογή των παραμέτρων έγινε βασιζόμενη στο μικρότερο δυνατό μέσο τετραγωνικό σφάλμα.

Έπειτα ακολουθούμε την ίδια διαδικασία με το RNN και βρίσκουμε σφάλμα ($mse \approx 0.02014$).

- **Πρόβλεψη τιμών μετά τη 1/1/2021**

Ακολουθούμε την ίδια διαδικασία, αλλά αντί για current_set τώρα έχουμε το future_set. Οπότε υπολογίζουμε το ποσοστό θετικότητας 3 μέρες μετά, αφαιρούμε τις γραμμές με NaN τιμές, ορίζουμε τα σύνολα X και y και κλιμακώνουμε τα δεδομένα. Τέλος προβλέπουμε τα χαρακτηριστικά του συνόλου X μέσω της predict() κάνοντας χρήση των δύο μοντέλων, τις οποίες προβλέψεις περνάμε σε ένα νέο dataframe (predictions) μαζί με τις πραγματικές τιμές. Υπολογίζουμε τα μέσα τετραγωνικά σφάλματα, $mse_{RNN} \approx 0.0195$ *και* $mse_{SVM} \approx 0.02048$ και προβάλουμε τα δεδομένα μας.



Παρατηρούμε ότι οι προβλέψεις των μοντέλων μας δεν απέχουν πολύ από τις κανονικές τιμές, αρά η εκπαίδευση τους ήταν επιτυχημένη. Ένας άλλος τρόπος για να αξιολογήσουμε τα μοντέλα μας είναι μέσω των μετρικών.

	RNN	**SVM**
Mean Absolute Error	0.084285	0.069196
Mean Squared Error	0.019502	0.020477
Root Mean Squared Error	0.13965	0.143098
Mean Absolute Percentage Error	5066402262488.44	4497913996245.437
R2 Score	-0.139998	-0.196987

Συμπερασματικά, και τα δύο μοντέλα παρουσιάζουν σχετικά παρόμοια απόδοση όσον αφορά τα MAE και MSE, με το μοντέλο SVM να έχει ελαφρώς καλύτερη απόδοση.