



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Ανάλυση Δεδομένων με χρήση της Python. Μελέτη Περίπτωσης

Κωνσταντίνος Γκόγκας

Επιβλέπουσα: Γεωργία Φουτσιτζή

Άρτα, Ιούλιος 2024

Data Analysis with Python. A Case Study

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Τόπος, Ημερομηνία

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπουσα καθηγήτρια

Γεωργία Φουτσιτζή, Καθηγήτρια

2. Μέλος επιτροπής

Νικόλαος Αντωνιάδης, Καθηγητής

3. Μέλος επιτροπής

Χρήστος Γκόγκος, Καθηγητής

© Γκόγκας, Κωνσταντίνος, 2024.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εξ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Γκόγκας Κωνσταντίνος

Υπογραφή

ΕΥΧΑΡΙΣΤΙΕΣ

Είναι σημαντική η αναγνώριση της βοήθειας που έλαβα από την καθηγήτρια μου κα. Γεωργία Φουτσιτζή για την πολύτιμη βοήθεια της στην διεκπεραίωση της πτυχιακής εργασίας.

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία αναλύεται η Python, μια γλώσσα η οποία χρησιμοποιείται πάρα πολύ στην ανάλυση δεδομένων. Γίνεται αναφορά στο τί είναι τα δεδομένα και πως κατηγοριοποιούνται. Στην συνέχεια βασικές εντολές της Python και βιβλιοθήκες απαραίτητες για την ανάλυση δεδομένων αναλύονται στην εργασία. Στην συνέχεια εστιάζουμε σε μία στατιστική ανάλυση δεδομένων σε δεδομένα που έχουν περιεχόμενο την ποιότητα οξυγόνου της Νέας Υόρκης για το 2022 και πιο συγκεκριμένα τον δείκτη PM 2.5. Η ανάλυση αποτελείται από απλούς στατιστικούς δείκτες όπως είναι η μέση τιμή και τυπική απόκλιση επεκτείνεται σε έλεγχο υποθέσεων ANOVA και συσχέτιση Pearson.

Λέξεις-κλειδιά: Ανάλυση δεδομένων, Python, δείκτης ποιότητας του αέρα .

ABSTRACT

In this thesis Python, a language which is very much used in data analysis, is analysed. It is mentioned what data is and how it is categorized. Then basic Python commands and libraries necessary for data analysis are analyzed in the paper. We then focus on a statistical data analysis on a data set with the context of New York City's oxygen quality for 2022 and more specifically the PM 2.5 index. The analysis consists of simple statistical indicators such as mean and standard deviation and is developed in hypothesis testing with ANOVA and Pearson correlation.

Keywords: Data analysis, Python, air quality index.

Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ	6
ΠΕΡΙΛΗΨΗ	7
ABSTRACT	8
Κατάλογος Εικόνων	11
ΚΕΦΑΛΑΙΟ 1 Δεδομένα	14
1.2 Τύποι δεδομένων	14
1.3 Δομές Δεδομένων	17
ΚΕΦΑΛΑΙΟ 2 Python	18
2.1 Γλώσσες Σεναρίων	18
2.2 Ονόματα και διακριτικά	18
2.3 Σχόλια στην Python	19
2.4 Τύποι Δεδομένων στην Python	19
2.4.1 Αριθμητικός τύπος δεδομένων	19
2.4.2 Χαρακτήρες	20
2.4.3 Λίστα	21
2.4.4 Τύπος Δεδομένων Boolean	21
2.4.5 Μετατροπή τύπου δεδομένων	22
2.5 Δομή Επιλογής IF	24
2.6 Δομή επανάληψης	26
2.6.1 while Loop	26
2.6.2 for loop	27
2.6.3 Break και Continue	27
2.7 Συναρτήσεις	29
2.7.1 def	29
2.7.2 Global μεταβλητές	30
ΚΕΦΑΛΑΙΟ 3 Βιβλιοθήκες της Python	32
3.1 Pandas	32
3.2 Numpy	33
3.3 Matplotlib	34
3.4 Seaborn	35
3.5 Statistics	35

3.6 warnings	36
ΚΕΦΑΛΑΙΟ 4 Εφαρμογή- Δείκτης ποιότητας αέρα PM2.5	37
4.1 Αναγνώριση του προβλήματος	37
4.1.1 AQI	38
4.1.2 PM2.5.....	39
4.2 Καθαρισμός Δεδομένων	39
4.2.1 Αφαίρεση διπλών εγγραφών.....	40
4.2.2 Αντιμετώπιση τιμών NULL	40
4.2.3 Ακραίες τιμές.....	42
4.3 Περιγραφική στατιστική	44
4.3.1 Διαγράμματα.....	44
4.3.2 Κεντρική Τάση	49
4.3.3 Διασπορά	52
4.4 Επαγωγική στατιστική	61
4.4.1 Έλεγχος υποθέσεων	61
4.4.2 Συσχέτιση Pearson	64
ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα.....	66
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	68

Κατάλογος Εικόνων

Εικόνα 1: Δενδροειδής γραφική απεικόνιση για τους τύπους δεδομένων.

Εικόνα 2: Δενδροειδής γραφική απεικόνιση για τις δομές δεδομένων.

Εικόνα 3: Τρόποι εγγραφής σχολίων στην Python.

Εικόνα 4: Δήλωση και εμφάνιση μεταβλητή τύπου int.

Εικόνα 5: Δήλωση και εμφάνιση μεταβλητή τύπου float.

Εικόνα 6: Δήλωση και εμφάνιση μεταβλητής τύπου String.

Εικόνα 7: Δήλωση και εμφάνιση μεταβλητής τύπου List.

Εικόνα 8: Δήλωση και εμφάνιση μεταβλητής τύπου Boolean.

Εικόνα 9: Πράξεις μεταβλητών τύπου Boolean.

Εικόνα 10: Σιωπηρή μετατροπή τύπου.

Εικόνα 11: Παραδείγματα μετατροπών δεδομένων με την χρήση μεθόδων του παραπάνω πίνακα.

Εικόνα 12: Παράδειγμα απλής if.

Εικόνα 13: Παράδειγμα με if και else.

Εικόνα 14: Παράδειγμα με if, elif και else.

Εικόνα 15: Παράδειγμα if με string και εμφωλευμένη if.

Εικόνα 16: Απλό παράδειγμα while loop.

Εικόνα 17: Απλό παράδειγμα for loop.

Εικόνα 18: Παράδειγμα break.

Εικόνα 19: Παράδειγμα continue.

Εικόνα 20: Παράδειγμα συνάρτησης σε Python.

Εικόνα 21: Συνάρτηση με return.

Εικόνα 22: Δήλωση global μεταβλητής.

Εικόνα 23: Global μεταβλητή μέσα στην συνάρτηση.

Εικόνα 24: Πίνακας κατηγοριών AQI.

Εικόνα 25: Αφαίρεση διπλών εγγραφών με την χρήση της drop_duplicate.

Εικόνα 26: Έλεγχος τιμών null με την χρήση της is_null.

Εικόνα 27: Θερμικός χάρτης για τον εντοπισμό ελλειπών στοιχείων.

Εικόνα 28: Αφαίρεση γραμμών που περιέχουν NULL τιμές με την εντολή dropna().

Εικόνα 29: Θερμικός χάρτης μετά από την αφαίρεση γραμμών με NULL τιμές.

Εικόνα 30: Δημιουργία διαγράμματος box plot για να δείξουμε ποσό των ακραίων τιμών.

Εικόνα 31: Δημιουργία διαγράμματος box plot για να δείξουμε ότι αφαιρέθηκαν οι ακραίες τιμές.

Εικόνα 32: Ραβδογράμμο που απεικονίζει την μέτρηση Daily Mean PM2.5 Concentration για κάθε κομητεία της Νέας Υόρκης με την χρήση Python.

Εικόνα 33: Διάγραμμα γραμμής που απεικονίζει την μέτρηση Daily Mean PM2.5 Concentration για κάθε μήνα του 2022 για την Νέα Υόρκη με την χρήση Python.

Εικόνα 34: Διάγραμμα ντόνατ που απεικονίζει την το ποσοστό επί του συνόλου για το AQI Category για την Νέα Υόρκη το 2022 με την χρήση Python.

Εικόνα 35: Θερμικός χάρτης που απεικονίζει τον αριθμό μετρήσεων σε κάθε AQI Category για τις κομητείες της Νέας Υόρκης το 2022 με την χρήση Python.

Εικόνα 36: Συνάρτηση mean() στην Python.

Εικόνα 37: Μέση τιμή σε ιστόγραμμα.

Εικόνα 38: Χρήση της συνάρτησης median() για την διάμεσο.

Εικόνα 39: Χρήση της συνάρτησης mode() για να βρούμε την μεγαλύτερη συχνότητα στα δεδομένα.

Εικόνα 40: Επικρατούσα τιμή σε ιστόγραμμα.

Εικόνα 41: Εύρος στην Python.

Εικόνα 42: Συνάρτηση describe() στην Python.

Εικόνα 43: Ενδοτεταρτημοριακή απόκλιση στην Python.

Εικόνα 44: Διακύμανση στην Python

Εικόνα 45: Διάγραμμα χαμηλής τυπικής απόκλισης.

Εικόνα 46: Διάγραμμα υψηλής τυπικής απόκλισης

Εικόνα 47: Πρώτη, δεύτερη και τρίτη τυπική απόκλιση στην Python

Εικόνα 48: Πρώτη, δεύτερη και τρίτη τυπική απόκλιση σε ιστόγραμμα.

Εικόνα 49: Καμπύλη συμμετρικής κατανομής

Εικόνα 50: Καμπύλη θετική ασυμμετρίας

Εικόνα 51: Καμπύλη αρνητικής ασυμμετρίας

Εικόνα 52: Συνάρτηση skew() στην Python

Εικόνα 53: Ιστόγραμμα για το PM 2.5.

Εικόνα 54: Συνάρτηση kurt() στην Python.

Εικόνα 55: Κομμάτι κώδικα για την δημιουργία της στήλης AQI_ Category.

Εικόνα 56: Κομμάτι κώδικα για την χρήση της `f_oneway()`.

Εικόνα 57: Διαγράμματα αρνητικής, θετικής και χωρίς συσχέτιση.

Εικόνα 58: Κομμάτι κώδικα για την εκτέλεση της συσχέτισης Pearson με την `pearson()`.

ΚΕΦΑΛΑΙΟ 1 Δεδομένα

Πολλές φορές ο όρος δεδομένα και πληροφορία χρησιμοποιούνται για τον ίδιο σκοπό. Αυτό είναι λάθος καθώς οι δύο αυτές έννοιες είναι διαφορετικές μεταξύ τους. Τα δεδομένα είναι ένα σύνολο από στοιχεία τυποποιημένα σε καθορισμένη μορφή και είναι έτοιμα για επεξεργασία από ανθρώπους ή μηχανή. Τα δεδομένα μεταφέρουν πληροφορία, περιγράφουν ποσότητα, ποιότητα, γεγονότα και στατιστικές. Η πληροφορία είναι το αποτέλεσμα μετά την επεξεργασία των δεδομένων. Τα ακατέργαστα δεδομένα, συχνά αναφέρονται και ως πηγαία, είναι τα δεδομένα που δεν έχουν υποστεί κάποια επεξεργασία. Τα ακατέργαστα δεδομένα είναι εξαιρετικά δύσκολα να κατανοηθούν καθώς είναι οπτικά ακατάστατα και δεν έχουν συνοχή. Κωδικοποίηση, μορφοποίηση και ανάλυση είναι τρόποι επεξεργασίας αυτών το δεδομένων καθώς αποτελούνε πολύτιμο πόρο πληροφορίας. [1], [2], [3].

1.2 Τύποι δεδομένων

Τα δεδομένα μπορούν να περιγράφουν χρησιμοποιώντας διαφορετικές κλίμακες μέτρησης. Οι διαφορετικοί τύποι των δεδομένων κατηγοριοποιούνται σύμφωνα με τον τύπο κλίμακας μέτρησης. Όταν επεξεργαζόμαστε δεδομένα είναι καλό να γνωρίζουμε τον τύπο τους. Για παράδειγμα σε μία λίστα εργαζομένων τα δεδομένα μπορούν να περιέχουν τον κωδικό του εργαζόμενου και την ηλικία του. Δεν είναι όμως όλα τα δεδομένα αριθμοί, στην ίδια λίστα μπορεί να συμπεριλαμβάνεται και το όνομα του εργαζόμενου ή/και η ημερομηνία πρόσληψής του. Δύο βασικές κατηγορίες δεδομένων είναι τα κατηγορικά (ποιοτικά) και τα αριθμητικά (ποσοτικά) δεδομένα.

Τύποι δεδομένων

Τα **κατηγορικά** (ή ποιοτικά) **δεδομένα** αναφέρονται σε χαρακτηριστικά. Μπορούν να αντιπροσωπεύουν πράγματα όπως φύλο, εθνικότητα, γλώσσα ομιλίας, χώρα. Τα ποιοτικά δεδομένα μπορούν επίσης να αναπαριστάνονται από αριθμούς, για παράδειγμα από ακεραίους, πρέπει να τα βλέπουμε ως σύμβολα και να στερούνται τις περισσότερες από τις ιδιότητες των αριθμών. Ακόμα και αν αναπαριστάνονται από αριθμούς, π.χ. από ακεραίους, πρέπει να αντιμετωπίζονται ως σύμβολα. Για παράδειγμα το αναγνωριστικό υπάλληλου σε μία λίστα συμβολίζεται με μια ακέραια μοναδική τιμή για κάθε υπάλληλο. Η τιμή αυτή θα χρησιμοποιείται μόνο για να αρχειοθετήσει τους υπαλλήλους. Τα κατηγορικά δεδομένα χωρίζονται στα **ονομαστικά δεδομένα** και τα **δεδομένα διάταξης** [4].

Οι τιμές στα ονομαστικά δεδομένα είναι απλώς διαφορετικά ονόματα. Αντιπροσωπεύουν διακριτές μονάδες που δεν έχουν ποσοτική αξία. Σε αυτήν την κατηγορία τα

δεδομένα δεν έχουν σειρά, οπότε άμα αλλάξει η σειρά των τιμών τους, το νόημα δεν αλλάζει.[4]
Παραδείγματα ονομαστικών δεδομένων είναι:

Το φύλο Πχ. Κυκλώστε την επιλογή σας:

Ποιο είναι το φύλο σας:

- Αρσενικό
- Θηλυκό

Το χρώμα των ματιών. Πχ. Ποιο είναι το χρώμα ματιών σας:

- Καστανό
- Μελί
- Γαλάζια
- Πράσινα

Στα δεδομένα διάταξης μπορούμε να ορίσουμε μια σχέση διάταξης. Και εδώ μπορούν να πάρουν αριθμητικές τιμές [5]. Παραδείγματα αυτής της κατηγορίας:

Τα αθλητικά βραβεία, η σειρά κατάταξης σε ένα διαγωνισμό, η στάση απέναντι σε ένα θέμα,

Πχ.

Ποιότητα παρεχόμενων υπηρεσιών:

Χαμηλό	Μέτριο	Καλό	Πολύ καλό	Εξαιρετικό
1	2	3	4	5

Η επόμενη βασική κατηγορία είναι τα **αριθμητικά** ή **ποσοτικά** δεδομένα. Εδώ τα δεδομένα αναπαριστούνται από πραγματικούς αριθμούς και κρατάνε τις πιο πολλές από τις ιδιότητες των αριθμών. Τα δεδομένα σε αυτόν τον τύπο χωρίζονται σε δύο υποκατηγορίες: τα **διακριτά** και τα **συνεχή** δεδομένα. Τα διακριτά δεδομένα έχουν πεπερασμένο ή αριθμήσιμο σύνολο τιμών. Παραδείγματα διακριτών δεδομένων είναι ο αριθμός υπαλλήλων σε μία επιχείρηση, ο αριθμός των μελών μιας οικογένειας, τα αποτελέσματα από 100 ρίψεις ζαριού. Τα συνεχή δεδομένα, θεωρητικά μπορούν να πάρουν οποιαδήποτε τιμή σε ένα διάστημα. Παραδείγματα συνεχών δεδομένων είναι η θερμοκρασία, το βάρος, το ύψος. Τα ποσοτικά δεδομένα χωρίζονται δύο υποκατηγορίες: τα **δεδομένα διαστήματος** και τα **δεδομένα αναλογίας** [4].

Τα δεδομένα διαστήματος μπορούν να συγκριθούν (διαταχθούν) μόνο ως προς τις μεταβολές τους (αφαίρεση) και όχι μεταξύ τους (διαίρεση). Δηλαδή η διαφορά μεταξύ των τιμών τους έχει σημασία. Το πρόβλημα με αυτόν τον τύπο είναι ότι δεν έχουν “πραγματικό μηδέν”, δηλ. το “μηδέν” δε δηλώνει την απουσία του μετρούμενου χαρακτηριστικού. Τα δεδομένα αυτά μπορούμε να προστεθούν και να αφαιρεθούν αλλά δεν μπορούμε να

εκτελέσουμε με αυτά τις πράξεις του πολλαπλασιασμού και της διαίρεσης ή να υπολογίσουμε αναλογίες [2][4]. Καλά παραδείγματα των δεδομένων διαστήματος είναι το βάρος, το ύψος, η θερμοκρασία σε βαθμούς Κελσίου ή Φαρενάιτ. Για παράδειγμα:

Πχ. Η σημερινή θερμοκρασία σε ποια κλίμακα αντιστοιχεί:

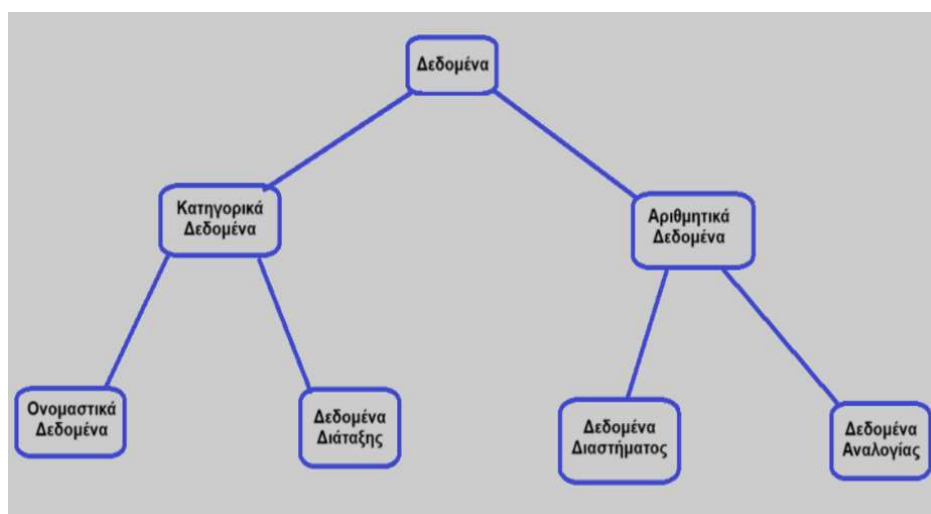
- -5°C
- 0°C
- $+5^{\circ}\text{C}$
- $+10^{\circ}\text{C}$
- $+15^{\circ}\text{C}$

Τέλος, υπάρχουν τα δεδομένα αναλογίας που οι τιμές τους αντιστοιχούν και αναλογικά στην ποσότητα του χαρακτηριστικού που μετρούν. Τα δεδομένα αναλογίας έχουν τις ιδιότητες της διάταξης και της απόστασης του προηγούμενου τύπου και ταυτόχρονα έχουν ένα πραγματικό μηδέν. Η διαφορά είναι όμως ότι εδώ το “πραγματικό μηδέν” δηλώνει την απουσία του χαρακτηριστικού που μελετάται. Οι αριθμητικές πράξεις που γίνονται με αυτά τα δεδομένα είναι ο πολλαπλασιασμός και η διαίρεση. Καλά παραδείγματα είναι η θερμοκρασία σε βαθμούς Κέλβιν, νομισματικές ποσότητες, η ηλικία, το ηλεκτρικό ρεύμα [2][4]. Για παράδειγμα:

Πχ. Μήκος (ίντσες)

- 0
- 5
- 10
- 15

Η παρακάτω εικόνα αποτυπώνει την δενδροειδή μορφή των τύπων δεδομένων που αναφέρθηκαν παραπάνω.



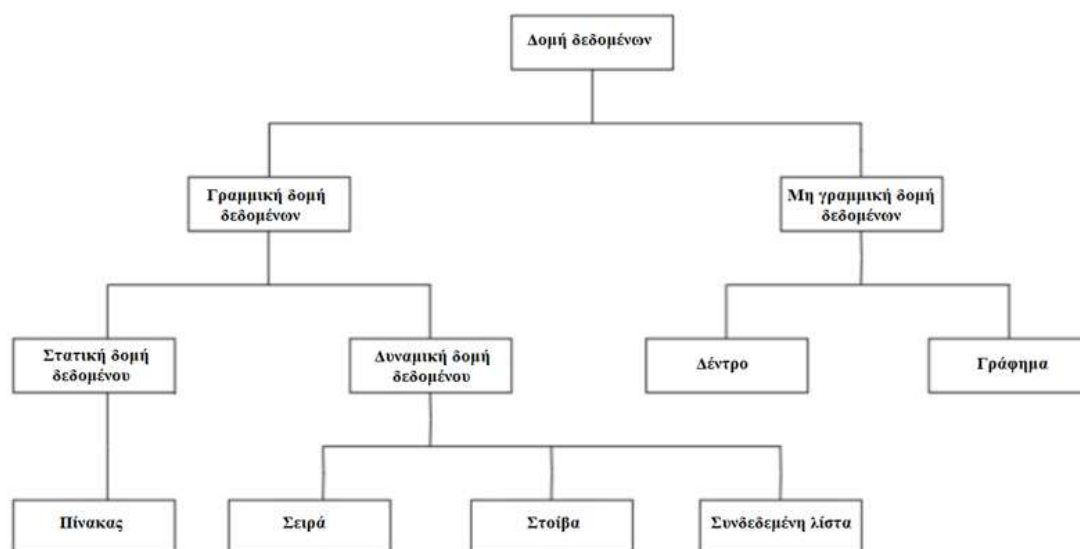
Εικόνα 1: Δενδροειδής γραφική απεικόνιση για τους τύπους δεδομένων

1.3 Δομές Δεδομένων

Μια δομή δεδομένων δεν χρησιμοποιείται μόνο για την οργάνωση των δεδομένων. Χρησιμοποιείται επίσης για την επεξεργασία, την ανάκτηση και την αποθήκευση δεδομένων. Υπάρχουν διάφοροι βασικοί και προηγμένοι τύποι δομών δεδομένων που χρησιμοποιούνται σχεδόν σε κάθε πρόγραμμα ή σύστημα λογισμικού που έχει αναπτυχθεί. Ο διαχωρισμός γίνεται σε δύο βασικές κατηγορίες, στις γραμμικές δομές και στις μη-γραμμικές δομές δεδομένων. [6]

Στις γραμμικές δομές δεδομένων τα στοιχεία διατάσσονται διαδοχικά ή γραμμικά, όπου το κάθε στοιχείο συνδέεται με το προηγούμενο αλλά και με το επόμενο γειτονικό του. Παραδείγματα γραμμικών δομών δεδομένων είναι ο πίνακας, η στοίβα, η ουρά, η συνδεδεμένη λίστα. Η γραμμική δομή διακρίνεται στις στατικές και στις δυναμικές δομές δεδομένων. Στατικές είναι οι δομές στις οποίες το μέγεθος μνήμης δεν μεταβάλλεται και παραμένει σταθερό. Παράδειγμα αυτής της υποκατηγορίας είναι ο πίνακας. Σε αντίθεση με την στατική δομή, στην δυναμική δομή το μέγεθος της μνήμης μεταβάλλεται. Μπορεί να ενημερώνεται τυχαία κατά την εκτέλεση, που έχει ως αποτέλεσμα να θεωρηθεί αποδοτικότερη όσο αναφορά τον χώρο μνήμης του κώδικα. Παραδείγματα αυτής της υποκατηγορίας είναι η ουρά και η στοίβα. [6]

Στις μη-γραμμικές δομές δεδομένων τα στοιχεία δεν τοποθετούνται διαδοχικά ή γραμμικά. Σε μια μη-γραμμική δομή δεν μπορούμε να διατρέξουμε όλα τα στοιχεία σε μια μόνο εκτέλεση. Παραδείγματα για αυτήν την κατηγορία είναι τα δέντρα και τα γραφήματα. [6]



Εικόνα 2: Δενδροειδής γραφική απεικόνιση για τις δομές δεδομένων

ΚΕΦΑΛΑΙΟ 2 Python

Η Python είναι μια προηγμένη γλώσσα σεναρίων (scripting language) που χρησιμοποιείται με επιτυχία για να ενώνει κομμάτια λογισμικών μεταξύ τους. Εκτείνεται σε πολλαπλές πλατφόρμες, προϊόντα ενδιάμεσου λογισμικού και τομείς εφαρμογών. Η Python είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου με ενσωματωμένες δομές δεδομένων. Η Python κυκλοφορεί από το 1991 και έχει μια πολύ ενεργή κοινότητα χρηστών. Συντακτικά, ο κώδικας Python μοιάζει με εκτελέσιμο ψευδοκώδικα. Η ανάπτυξη προγραμμάτων με τη χρήση της Python είναι 5-10 φορές ταχύτερη από τη χρήση της C/C++ και 3-5 φορές ταχύτερη από τη χρήση της Java. Σε πολλές περιπτώσεις, το αρχικό στάδιο μιας εφαρμογής μπορεί να γραφτεί σε Python χωρίς να γραφτεί καθόλου κώδικας C/C++/Java. Συχνά, το πρωτότυπο είναι επαρκώς λειτουργικό και αποδίδει αρκετά καλά ώστε να παραδοθεί ως τελικό προϊόν, εξοικονομώντας σημαντικό χρόνο ανάπτυξης. Άλλες φορές, το πρωτότυπο μπορεί να μεταφραστεί εν μέρει ή στο σύνολό του σε C++ ή Java -- η αντικειμενοστραφής φύση της Python καθιστά τη μετάφραση μια απλή διαδικασία. Η Python μπορεί να εκπληρώσει σημαντικό ρόλο ολοκλήρωσης στο σχεδιασμό μεγάλων εφαρμογών με μεγάλη διάρκεια ζωής. Επιτρέπει την ταχεία ανταπόκριση σε αλλαγές στις απαιτήσεις των χρηστών που απαιτούν την προσαρμογή της λογικής της εφαρμογής υψηλότερου επιπέδου χωρίς να αλλάζουν τα θεμελιώδη υποκείμενα στοιχεία. Επιτρέπει επίσης τη γρήγορη προσαρμογή της εφαρμογής σε αλλαγές στα υποκείμενα συστατικά. [7][8]

2.1 Γλώσσες Σεναρίων

Μια γλώσσα σεναρίων είναι ένα είδους γλώσσας προγραμματισμού που χρησιμοποιεί μια δομή υψηλού επιπέδου όχι μόνο για να ερμηνεύσει αλλά και να εκτελέσει κάθε γραμμή. Οι γλώσσες σεναρίων, είναι ποιο εύκολες στην εκμάθηση και ταχύτερες στον προγραμματισμό σε σχέση από τις μεταγλωττισμένες γλώσσες όπως είναι η C και C++. [20]

Η γλώσσες σεναρίων συνήθως δεν μεταγλωττίζονται, το οποίο είναι και το κύριο στοίχοι που διαφέρει από άλλες γλώσσες προγραμματισμού. Τη θέση του μεταγλωττιστεί την έχει πάρει ένας διερμηνέας, ο οποίος διαβάσει και εκτελεί το πρόγραμμα την ίδια χρονική στιγμή. Η λίστα από γλώσσες σεναρίων συμπεριλαμβάνει την Python, R, PHP, JavaScript, Ruby. [21]

2.2 Ονόματα και διακριτικά

Τα ονόματα στην Python μπορούν να περιλαμβάνουν χαρακτήρες που αποτελούνται από όλα τα κεφαλαία και πεζά γράμματα του Αγγλικού αλφάβητου (A-Z, a-z), όλους τους αριθμούς στο δεκαδικό σύστημα αρίθμησης (0-9) και την κάτω παύλα ή υπογράμμιση (_). Τα ονόματα όμως δεν μπορούν να ξεκινάνε με αριθμούς. Η Python είναι μια γλώσσα με ευαισθησία στη χρήση πεζών και κεφαλαίων γραμμάτων, δηλαδή αντιμετωπίζει διαφορετικά τους κεφαλαίους και τους πεζούς χαρακτήρες. Αυτό ισχύει και για τα αναγνωριστικά. Συνήθως στις γλώσσες προγραμματισμού ο προγραμματιστής ορίζει τον τύπο της μεταβλητής. Στην Python αυτό δεν γίνεται η μεταβλητή θα πάρει αυτόματα τύπο ανάλογα με το τι τιμή θα λάβει. Για παράδειγμα άμα η μεταβλητή *x* λάβει την 5 τότε η μεταβλητή *x* θα έχει τύπο ακέραιας τιμής. [7] [8]

2.3 Σχόλια στην Python

Τα σχόλια στην Python αναγνωρίζονται με ένα σύμβολο hash (#) και επεκτείνονται μέχρι το τέλος της γραμμής. Ωστόσο, οι χαρακτήρες hash σε μια συμβολοσειρά δεν θεωρούνται σχόλια. Υπάρχουν τρεις τρόποι για να γραφεί ένα σχόλιο - ως ξεχωριστή γραμμή, δίπλα στην αντίστοιχη εντολή κώδικα ή ως μπλοκ σχολίων πολλών γραμμών. [9]

```
# σχόλιο σε ξεχωριστή γραμμή
x = 5 # σχόλιο δίπλα στην αντίστοιχη εντολή κώδικα

# μπλοκ
# σχολίων
# πολλών
# γραμμών
```

Εικόνα 3: Τρόποι εγγραφής σχολίων στην Python.

2.4 Τύποι Δεδομένων στην Python

Οι τύποι δεδομένων της Python χρησιμοποιούνται για τον καθορισμό του τύπου μιας μεταβλητής. Ορίζει τι είδους δεδομένα πρόκειται να αποθηκεύσουμε σε μια μεταβλητή. Τα δεδομένα που αποθηκεύονται στη μνήμη μπορεί να είναι πολλών τύπων. Για παράδειγμα, η ηλικία ενός ατόμου αποθηκεύεται ως μια αριθμητική τιμή και το όνομα του αποθηκεύεται ως αλφαριθμητική τιμή. Η ενσωματωμένη συνάρτηση **type()** εμφανίζει τον τύπο των δομών δεδομένων που χρησιμοποιούνται για την αποθήκευση των στοιχείων δεδομένων [10][11].

2.4.1 Αριθμητικός τύπος δεδομένων

Ένας τύπος μεταβλητών στην Python είναι οι αριθμητικές τιμές. Οι τιμές αυτές μπορούν να πάρουν διάφορες μορφές και μέγεθος. Οι ακέραιες τιμές, οι οποίες στον προγραμματισμό ονομάζονται **integers** και συμβολίζονται με **int**, αποτελούνται από όλους του φυσικούς αριθμούς (1,2,3,...,N) μαζί με τους αντίθετους (-1,-2,-3,...,-N) τους αλλά και με το μηδέν (0) δηλαδή τους ακεραίους.[10]

```
# Δήλωση μιας μεταβλητής με όνομα x και τιμή 5
x = 5

print("Ο τύπος της μεταβλητής x είναι: ",type(x)) #εμφανίζει τον τύπο της μεταβλητής με όνομα x

Ο τύπος της μεταβλητής x είναι: <class 'int'>
```

Εικόνα 4: Δήλωση και εμφάνιση μεταβλητή τύπου **int**.

Η επόμενη μορφή που μπορεί να πάρει μια αριθμητική τιμή είναι οι δεκαδικές τιμές ή κλασματικοί αριθμοί (δηλαδή οι ρητοί αριθμοί). Στον προγραμματισμό ονομάζεται και συμβολίζεται με **float**. Η υποδιαστολή συμβολίζεται με μια τελεία (.) και μπορεί να ακολουθήσει και ο μηδενικός χαρακτήρας. Οι αριθμοί όπως 5.2 , -32.467 , 152/43 και το 10.0 συμπεριλαμβάνονται στο σύνολο των **float** αριθμών. [10]

```
a = 5.2 # Δήλωση μιας μεταβλητής με όνομα a και δεκαδική τιμή 5.2
b = -32.467 # Δήλωση μιας μεταβλητής με όνομα b και δεκαδική τιμή -32.467
c = 152/43 # Δήλωση μιας μεταβλητής με όνομα c και τιμή το κλάσμα 152/43
d = 10.0 # Δήλωση μιας μεταβλητής με όνομα d και δεκαδική τιμή 10.0

print("Ο τύπος της μεταβλητής x είναι: ",type(a)) #εμφανίζει τον τύπο της μεταβλητής με όνομα a
print("Ο τύπος της μεταβλητής x είναι: ",type(b)) #εμφανίζει τον τύπο της μεταβλητής με όνομα b
print("Ο τύπος της μεταβλητής x είναι: ",type(c)) #εμφανίζει τον τύπο της μεταβλητής με όνομα c
print("Ο τύπος της μεταβλητής x είναι: ",type(d)) #εμφανίζει τον τύπο της μεταβλητής με όνομα d

Ο τύπος της μεταβλητής x είναι: <class 'float'>
Ο τύπος της μεταβλητής x είναι: <class 'float'>
Ο τύπος της μεταβλητής x είναι: <class 'float'>
Ο τύπος της μεταβλητής x είναι: <class 'float'>
```

Εικόνα 5: Δήλωση και εμφάνιση μεταβλητή τύπου **float**.

2.4.2 Χαρακτήρες

Εκτός από αριθμητικά δεδομένα, μπορεί να δοθεί και ένα σύνολο χαρακτήρων. Οι χαρακτήρες πρέπει να είναι μέσα σε μονά εισαγωγικά. Αυτός ο τύπος δεδομένων ονομάζεται **String**, αλφαριθμητικά στα ελληνικά, και συμβολίζεται με **str**. Υποσύνολα συμβολοσειρών μπορούν να ληφθούν χρησιμοποιώντας τον τελεστή **slice** ([] και [:]) με δείκτες που ξεκινούν από το 0 στην αρχή της συμβολοσειράς και προχωρούν από το -1 στο τέλος. Το σύμβολο της πρόσθεσης (+) είναι ο τελεστής συνένωσης συμβολοσειρών και ο αστερίσκος (*) είναι ο τελεστής επανάληψης στην Python.[10]

```
x = 'Hello World!' #Δήλωση ενός String με όνομα x

print ("Ο τύπος της μεταβλητής x είναι:", type(x)) #Εμφανίζει τον τύπο μεταβλητής x
print (x) # Εμφανίζει ολόκληρο το String
print (x[0]) # Εμφανίζει τον πρώτο χαρακτήρα
print (x[2:5]) # Εμφανίζει τον 3ο έως και τον 5ο χαρακτήρα
print (x[2:]) # Εμφανίζει ολόκληρο το String ξεκινώντας από τον 3ο χαρακτήρα
print (x * 2) # Εμφανίζει το String δύο φορές
print (x + "TEST") # Εμφανίζει συνδεδεμένο String

Ο τύπος της μεταβλητής x είναι: <class 'str'>
Hello World!
H
llo
llo World!
Hello World!Hello World!
Hello World!TEST
```

Εικόνα 6: Δήλωση και εμφάνιση μεταβλητής τύπου String.

2.4.3 Λίστα

Οι λίστες της Python είναι οι πιο ευέλικτοι σύνθετοι τύποι δεδομένων. Μια λίστα Python (List) περιέχει στοιχεία που χωρίζονται με κόμμα και περικλείονται μέσα σε τετράγωνα αγκύλες ([]). Τα στοιχεία που ανήκουν σε μια λίστα της Python μπορούν να είναι διαφορετικού τύπου δεδομένων. Οι τιμές που είναι αποθηκευμένες σε μια λίστα της Python μπορούν να προσπερασθούν χρησιμοποιώντας τον τελεστή slice ([] και [:]) με δείκτες που ξεκινούν από το 0 στην αρχή της λίστας και φτάνουν μέχρι το τέλος -1. Το σύμβολο συν (+) είναι ο τελεστής συνένωσης λιστών και ο αστερίσκος (*) είναι ο τελεστής επανάληψης [10].

```
x = [ 'abcd', 786 , 2.23, 'john', 70.2 ] #Δήλωση μιας λίστας με όνομα x
y = [123, 'john'] #Δήλωση μιας λίστας με όνομα y
print ("Ο τύπος της μεταβλητής x είναι:", type(x)) #Εμφανίζει τον τύπο μεταβλητής x
print ("Ο τύπος της μεταβλητής y είναι:", type(y)) #Εμφανίζει τον τύπο μεταβλητής y
print (x) #Εμφανίζει ολόκληρη την λίστα x
print (x[0]) #Εμφανίζει το πρώτοστοιχείο της λίστας x
print (x[1:3]) #Εμφανίζει τα στοιχεία της λίστας x από το 2ο έως και το 3ο
print (x[2:]) #Εμφανίζει την λίστα x ξεκινώντας από το 3ο στοιχείο
print (y * 2) #Εμφανίζει την λίστα y δύο φορές
print (x + y) #Εμφανίζει την συνένωση της λίστας x και y

Ο τύπος της μεταβλητής x είναι: <class 'list'>
Ο τύπος της μεταβλητής y είναι: <class 'list'>
['abcd', 786, 2.23, 'john', 70.2]
abcd
[786, 2.23]
[2.23, 'john', 70.2]
[123, 'john', 123, 'john']
['abcd', 786, 2.23, 'john', 70.2, 123, 'john']
```

Εικόνα 7: Δήλωση και εμφάνιση μεταβλητής τύπου List.

2.4.4 Τύπος Δεδομένων Boolean

Ο τύπος δεδομένων Boolean χρησιμοποιείται για να αντιπροσωπεύει δύο από τις δύο τιμές Σωστό (True) και Λάθος (False). Πιο συγκεκριμένα, είναι για να δηλώνει αν μια έκφραση είναι αληθής ή όχι. Για παράδειγμα το $1==1$ ισχύει ενώ το $2>5$ είναι λανθασμένη έκφραση. [15]

```
x = True #Δήλωση μιας μεταβλητής με όνομα x και τιμή True
print ("Ο τύπος της μεταβλητής x είναι:", type(x)) #Εμφανίζει τον τύπο μεταβλητής x

y = False #Δήλωση μιας μεταβλητής με όνομα y και τιμή False
print ("Ο τύπος της μεταβλητής y είναι:", type(y)) #Εμφανίζει τον τύπο μεταβλητής y

Ο τύπος της μεταβλητής x είναι: <class 'bool'>
Ο τύπος της μεταβλητής y είναι: <class 'bool'>
```

Εικόνα 8: Δήλωση και εμφάνιση μεταβλητής τύπου Boolean.

Με την εντολή **bool()** μπορούμε να αξιολογήσουμε τις μεταβλητές. Η συγκεκριμένη εντολή χρησιμοποιείται για να επιστρέψει ή να μετατρέψει μία τιμή σε τιμή Boolean, δηλαδή True ή False, χρησιμοποιώντας την τυπική διαδικασία ελέγχου αλήθειας. Όλοι οι αριθμοί, ακέραιοι ή δεκαδικοί, με την παραπάνω μέθοδο μπορούν να οριστούν ως αληθής ή λανθασμένοι. Σε περίπτωση που η μεταβλητή έχει μηδενική τιμή (0 ή 0.0) το αποτέλεσμα θα είναι λανθασμένο. Το ίδιο αποτέλεσμα θα έχει και μια κενή ακολουθία.[15]

```
x = 2 #Δήλωση μιας μεταβλητής με όνομα x και τιμή 2
y = 4 #Δήλωση μιας μεταβλητής με όνομα x και τιμή 4
# Σύγκριση μεταξύ των τιμών x και y
print(bool(x==y))
print(bool(x>y))
print(bool(x<y))

z = 0.0 #Δήλωση μιας μεταβλητής με όνομα z και τιμή 0.0
print(bool(z))

a = () #Δήλωση μιας μεταβλητής με όνομα a και τιμή μια κενή ακολουθία
print(bool(a))

False
False
True
False
False
```

Εικόνα 9: Πράξεις μεταβλητών τύπου Boolean.

2.4.5 Μετατροπή τύπου δεδομένων

Υπάρχουν κυρίως δύο τύποι μεθόδων μετατροπής τύπου στην Python, δηλαδή η έμμεση μετατροπή τύπου (implicit type conversion) και η ρητή μετατροπή τύπου (explicit type conversion). [16]

Όταν η μετατροπή τύπου δεδομένων πραγματοποιείται κατά τη μεταγλώττιση ή κατά τη διάρκεια της εκτέλεσης, τότε ονομάζεται έμμεση μετατροπή τύπου δεδομένων. Η έμμεση

μετατροπή του τύπου δεδομένων γίνεται από την Python, οπότε δεν χρειάζεται να μετατρέψουμε ρητά τον τύπο δεδομένων σε έναν άλλο τύπο δεδομένων. Στην παρακάτω εικόνα απεικονίζεται ένα παράδειγμα. Έχουν δηλωθεί δύο μεταβλητές, η μία είναι τύπου integer ($x = 5$) και η δεύτερη είναι τύπου float ($y = 5.5$). Στην συνέχεια δηλώθηκε μια μεταβλητή με όνομα 'sum' η οποία χρησιμοποιείται για να αποθηκεύσει το άθροισμα των δύο παραπάνω τιμών. Με τον έλεγχο του τύπου της μεταβλητής 'sum', έχει πάρει αυτόματα τον τύπο float. Ο λόγος που μετατρέπεται σε float και όχι σε integer, είναι διότι αν ο μεταγλωττιστής την είχε μετατρέψει σε τύπο δεδομένων integer, τότε θα έπρεπε να αφαιρέσει το κλασματικό μέρος και αυτό θα είχε ως αποτέλεσμα την απώλεια δεδομένων. Έτσι, η Python μετατρέπει πάντα το μικρότερο τύπο δεδομένων σε μεγαλύτερο για να αποτρέψει την απώλεια δεδομένων [16].

```
x = 5 # integer
y = 5.5 # float
sum = x + y
print (sum)
print ("Ο τύπος της μεταβλητής sum είναι:", type(sum))
```

10.5
Ο τύπος της μεταβλητής sum είναι: <class 'float'>

Εικόνα 10: Σιωπηρή μετατροπή τύπου.

Η ρητή μετατροπή τύπου δεδομένων είναι επίσης γνωστή ως typecasting. Η ρητή μετατροπή τύπου δεδομένων λαμβάνει χώρα όταν ο προγραμματιστής ορίζει σαφώς και ρητά τον ίδιο τύπο στο πρόγραμμα. Για τη ρητή μετατροπή τύπου, υπάρχουν ορισμένες ενσωματωμένες συναρτήσεις της Python. Ο παρακάτω πίνακας περιέχει μερικές από τις ενσωματωμένες συναρτήσεις για τη μετατροπή τύπου δεδομένων, μαζί με τις περιγραφές τους. [16]

Function	Description
int(x)	Μετατρέπει το x σε ακέραιο αριθμό.
float(x)	Μετατρέπει το x σε αριθμό κινητής υποδιαστολής.
str(x)	Μετατρέπει το x σε αλφαριθμητικό.
list(x)	Μετατρέπει το x σε λίστα.
ord(x)	Μετατρέπει έναν χαρακτήρα σε ακέραιο αριθμό.
hex(x)	Μετατρέπει έναν ακέραιο αριθμό σε δεκαεξαδικό αλφαριθμητικό.
oct(x)	Μετατρέπει έναν ακέραιο αριθμό σε οκταδικό αλφαριθμητικό.

```

x = 52
y = 23.7
z = "abcd"
a = "k"

print(int(y)) #Μετατρέπει το y σε ακέραιο αριθμό.
print(float(x)) #Μετατρέπει το x σε αριθμό κινητής υποδιαστολής.
print(str(x)) #Μετατρέπει το x σε αλφαριθμητικό.
print(list(z)) #Μετατρέπει το z σε λίστα.
print(ord(a)) #Μετατρέπει τον χαρακτήρα a σε ακέραιο αριθμό.
print(hex(x)) #Μετατρέπει τον ακέραιο αριθμό x σε δεκαεξαδικό αλφαριθμητικό.
print(oct(x)) #Μετατρέπει τον ακέραιο αριθμό x σε οκταδικό αλφαριθμητικό.

```

```

23
52.0
52
['a', 'b', 'c', 'd']
107
0x34
0o64

```

Εικόνα 11: Παραδείγματα μετατροπών δεδομένων με την χρήση μεθόδων του παραπάνω πίνακα.

2.5 Δομή Επιλογής IF

Η εντολή **if** στην Python είναι η κύρια εντολή που χρησιμοποιείται για επιλογή από εναλλακτικές ενέργειες. Καθώς είναι πρωταρχικό εργαλείο της Python, αντιπροσωπεύει μεγάλο μέρος της λογικής που διαθέτει ένα πρόγραμμα Python. Στην πραγματικότητα, η Python επιτρέπει τον συνδυασμό δηλώσεων σε ένα πρόγραμμα διαδοχικά έτσι ώστε να εκτελούνται η μία μετά την άλλη, και η μία μέσα στην άλλη, έτσι ώστε να εκτελούνται μόνο υπό ορισμένες συνθήκες. [17]

Η δομή της **if** έχει μια καθορισμένη μορφή. Αρχικά πρέπει να γίνει η δήλωση της **if** με την συνθήκη της ακολουθούμενη από ένα **elif** (else if), το οποίο είναι προαιρετικό. Τέλος και εξίσου προαιρετική είναι η εντολή **else** [17]

if συνθήκη 1:
 # code block 1

elif συνθήκη 2:
 # code block 2

else:
 # code block 3

```

x = 20
y = 10

if x > y:
    print("x is greater than y")

```

```

x is greater than y

```

Εικόνα 12: Παράδειγμα απλής if.

```
x = 20
y = 30

if x > y:
    print("x is greater than y")
else:
    print("y is greater than x")
```

y is greater than x

Εικόνα 13: Παράδειγμα με if και else.

```
x = 20
y = 20

if x > y:
    print("x is greater than y")
elif x == y:
    print("x is equal to y")
else:
    print("y is greater than x")
```

x is equal to y

Εικόνα 14: Παράδειγμα με if, elif και else.

Στα παρακάτω παραδείγματα απεικονίζονται απλά κομμάτια κώδικα με αριθμητικές μεταβλητές. Η συνθήκη if δεν είναι μόνο για αριθμητικές μεταβλητές, μπορεί να πάρει διάφορους τύπους μεταβλητών. Στο παρακάτω παράδειγμα είναι ένας κώδικας με Strings και με μια εμφωλευμένη if. [17]

```
name = "John"
lastname = "Terry"
if name == "John":
    if lastname == "Smith":
        print("Same name")
    else:
        print("Same name but different lastname ")
else:
    print("Different name ")
```

Same name but different lastname

Εικόνα 15: Παράδειγμα if με string και ενσωματωμένη if.

2.6 Δομή επανάληψης

Στην γραφή ενός κώδικα η διαδικασία της επανάληψης είναι ιδιαίτερα συχνή. Σε τέτοιες περιπτώσεις είναι πολύ πιο εύκολο να μάθουμε πώς επαναλαμβάνονται στον αλγόριθμο οι εντολές αυτές παρά να τις γράψουμε πολλές φορές. Στην Python οι δύο κύριες δομές επανάληψης που χρησιμοποιούνται είναι η **while** και η **for**. Η **while** παρέχει έναν τρόπο κωδικοποίησης γενικών βρόχων. Η **for** έχει σχεδιαστεί για να διατρέχει τα στοιχεία μιας ακολουθίας αντικείμενο και την εκτέλεση ενός μπλοκ κώδικα για κάθε ένα από αυτά [17] [18].

2.6.1 while Loop

Η εντολή **while** της Python είναι η πιο γενική κατασκευή επανάληψης στη γλώσσα. Σε απλούς όρους, εκτελεί επανειλημμένα ένα μπλοκ εντολών για όσο χρονικό διάστημα η αρχική συνθήκη συνεχίζει να ισχύει. Ονομάζεται "βρόχος" (Loop) επειδή ο έλεγχος συνεχίζει να επιστρέφει στην αρχή της δήλωσης μέχρι η αρχική συνθήκη να μην ισχύει πλέον. Όταν η δοκιμή γίνεται ψευδής, ο έλεγχος περνάει στη δήλωση που ακολουθεί το μπλοκ **while**. [17]

[Εντολές εκτός while loop]

while συνθήκη:

 Εντολή-1

 ...

 Εντολή- N

[Εντολές εκτός while loop]

```
i=1
while i<=8:
    print(i)
    i+=1
```

```
1
2
3
4
5
6
7
8
```

Εικόνα 16: Απλό παράδειγμα while loop.

2.6.2 for loop

Ο βρόχος επανάληψης **for** είναι ένας γενικός βρόχος επανάληψης στην Python, μπορεί να περάσει από τα στοιχεία του οποιοδήποτε αντικείμενο διατεταγμένης ακολουθίας. Η εντολή `for` λειτουργεί σε συμβολοσειρές, λίστες, πλειάδες, άλλες ενσωματωμένους επαναληπτικούς πίνακες, και αντικείμενα κλάσεων [17].

Ο βρόχος `for` της Python αρχίζει με μια γραμμή κεφαλίδας που καθορίζει έναν στόχο ανάθεσης (ή στόχους). Στην κεφαλίδα δίνουμε το στοιχείο και μία ακολουθία, ακολουθούμενη από το μπλοκ εντολών του βρόχου. [17]

[Εντολές εκτός `while` loop]

for στοιχείο **in** ακολουθία:

Εντολή-1

...

Εντολή- N

[Εντολές εκτός `while` loop]

```
names = ["John", "Maria", "Arya", "Ned"]  
  
for name in names:  
    print(name)
```

```
John  
Maria  
Arya  
Ned
```

Εικόνα 17: Απλό παράδειγμα `for` loop.

2.6.3 Break και Continue

Η εντολές **break** και **continue** μπορούν να χρησιμοποιηθούν μέσα στο μπλοκ κώδικα μίας δομής επανάληψης. Συχνά γράφονται σε μία δομή `if` ενσωματωμένη μέσα σε μια δομής επανάληψης, ώστε να εκτελείτε υπό συγκεκριμένες συνθήκες [17].

Η εντολή `break` προκαλεί την άμεση έξοδο από έναν βρόχο. Επειδή ο κώδικας που ακολουθεί τον βρόχο δεν θα εκτελεστεί όταν εκτελεστεί η `break`. Στο παράδειγμα της παρακάτω

εικόνας απεικονίζεται ένα παράδειγμα break σε έναν βρόχο while. Στο παράδειγμα αυτό, ο αριθμός i θα εμφανίζεται μέχρι να είναι ίδιος με τον αριθμό x, τότε εκτελείται η εντολή break μέσα στην if και η ροή του κώδικα συνεχίζει έξω από την επανάληψη [17].

```
x = 10
i = 1

while i <= x:
    if i > x:
        break
    print(i)
    i += 1
print("END")
```

```
1
2
3
4
5
6
7
8
9
10
END
```

Εικόνα 18: Παράδειγμα break.

Η εντολή continue προκαλεί την άμεση μετάβαση στην αρχή ενός βρόχου. Το επόμενο παράδειγμα θέλει να εμφανίσει όλους τους αριθμούς από το 1 έως το 20 που όταν διαιρούνται με τον αριθμό 3 έχει υπόλοιπο 0. [17]

```
for i in range(1,21):
    if i%3==0:
        continue
    print(i)
print("END")
```

```
1
2
4
5
7
8
10
11
13
14
16
17
19
20
END
```

Εικόνα 19: Παράδειγμα continue.

2.7 Συναρτήσεις

Με απλά λόγια, μια **συνάρτηση** είναι μια δομή που ομαδοποιεί ένα σύνολο εντολών ώστε να μπορούν να εκτελούνται περισσότερες από μία φορές σε ένα πρόγραμμα. Οι συναρτήσεις μπορούν επίσης να υπολογίσουν μια τιμή αποτελέσματος και να μας επιτρέψουν τον καθορισμό παραμέτρων που χρησιμεύουν ως είσοδοι της συνάρτησης, οι οποίες μπορεί να διαφέρουν κάθε φορά που ο κώδικας εκτελείται. Η κωδικοποίηση μιας λειτουργίας ως συνάρτηση την καθιστά ένα γενικά χρήσιμο εργαλείο, το οποίο μπορεί να χρησιμοποιηθεί σε διάφορα πλαίσια [17].

Οι συναρτήσεις βοηθάνε με την μεγιστοποίηση επαναχρησιμοποίησης κώδικα και ελαχιστοποίησης του πλεονασμού. Οι συναρτήσεις επιτρέπουν την ομαδοποίηση και την γενίκευση του κώδικα ώστε να χρησιμοποιηθεί αυθαίρετα πολλές φορές αργότερα. Επιτρέπουν την κωδικοποίηση μιας λειτουργία σε ένα μόνο μέρος και δίνεται η δυνατότητα να χρησιμοποιηθεί σε πολλά σημεία. [17]

2.7.1 def

Οι συναρτήσεις στην Python γράφονται με μια νέα δήλωση με τίτλο **def**. Οι συναρτήσεις πολλές φορές δηλώνονται μέσα σε εντολές **if**, βρόχους επανάληψης, ακόμα και σε άλλες συναρτήσεις. Σε αντίθεση με άλλες γλώσσες προγραμματισμού, η δήλωση **def** είναι εκτελέσιμος κώδικας και η συνάρτηση δεν υπάρχει μέχρι να εκτελεστεί η εντολή **def**. Όταν εκτελεστεί η εντολή **def**, δημιουργεί ένα νέο αντικείμενο συνάρτησης και θα το αναθέσει στην ονομαζόμενη συνάρτηση. Όπως συμβαίνει με όλες τις αναθέσεις, το όνομα της συνάρτησης γίνεται αναφορά στο function object. Το αντικείμενο της συνάρτησης μπορεί να εκχωρηθεί σε άλλα ονόματα, να αποθηκευτεί σε μια λίστα κ.ο.κ. Τα αντικείμενα συνάρτησης μπορούν επίσης να έχουν αυθαίρετα χαρακτηριστικά που καθορίζονται από τον χρήστη και συνδέονται με αυτά για την καταγραφή δεδομένων [17].

```
def hello():  
    print("Hello World")
```

```
hello()
```

```
Hello World
```

Εικόνα 20: Παράδειγμα συνάρτησης σε Python.

Όταν καλείται μια συνάρτηση, η εκτέλεση του κώδικα σταματάει μέχρι η συνάρτηση να ολοκληρώσει το έργο της και να επιστρέψει το αποτέλεσμα της. Οι συναρτήσεις που υπολογίζουν μια τιμή την αποστέλλουν πίσω στον κύριο κώδικα με μια δήλωση επιστροφής `return`, η επιστρεφόμενη τιμή γίνεται το αποτέλεσμα της κλήσης της συνάρτησης [17].

```
def is_odd(number):  
    if number % 2 == 1:  
        return True  
    return False  
  
for i in range(0,11,1):  
    print("The number ",i, "is odd:",is_odd(i))
```

```
The number 0 is odd: False  
The number 1 is odd: True  
The number 2 is odd: False  
The number 3 is odd: True  
The number 4 is odd: False  
The number 5 is odd: True  
The number 6 is odd: False  
The number 7 is odd: True  
The number 8 is odd: False  
The number 9 is odd: True  
The number 10 is odd: False
```

Εικόνα 21: Συνάρτηση με `return`.

2.7.2 Global μεταβλητές

Οι μεταβλητές που δημιουργούνται εκτός μιας συνάρτησης είναι γνωστές ως παγκόσμιες μεταβλητές **global**. Οι παγκόσμιες μεταβλητές μπορούν να χρησιμοποιηθούν από όλους, τόσο μέσα σε συναρτήσεις όσο και έξω από αυτές [19].

```
x = 5  
def f():  
    print(x+2)  
  
f()
```

7

Εικόνα 22: Δήλωση global μεταβλητής.

Κανονικά, όταν δημιουργείται μια μεταβλητή μέσα σε μια συνάρτηση, αυτή η μεταβλητή είναι τοπική και μπορεί να χρησιμοποιηθεί μόνο μέσα σε αυτή τη συνάρτηση. Για να δημιουργηθεί μια παγκόσμια μεταβλητή μέσα σε μια συνάρτηση, μπορείτε να χρησιμοποιήσετε τη λέξη-κλειδί `global`. Με την χρήση της λέξης-κλειδί `global`, μπορούν να δημιουργηθούν παγκόσμιες μεταβλητές μέσα στην συνάρτηση [19].

```
def f():  
    global x  
    x = 5  
    print(x % 2)  
  
f()
```

1

Εικόνα 23: Global μεταβλητή μέσα στην συνάρτηση.

ΚΕΦΑΛΑΙΟ 3 Βιβλιοθήκες της Python

Μία βιβλιοθήκη είναι ένας χώρος για αποθήκευση βιβλίων για μελλοντική χρήση. Στον προγραμματισμό ο όρος βιβλιοθήκη (Library) δεν διαφέρει και πολύ από την καθημερινή έννοιά της. Μια βιβλιοθήκη στην Python, αλλά και σε όλες τις γλώσσες προγραμματισμού, περιέχει δέσμες κώδικα που μπορούν να χρησιμοποιηθούν επανειλημμένα σε διαφορετικά προγράμματα. Οι βιβλιοθήκες κάνουν τον προγραμματισμό απλούστερο και γρηγορότερο για τον προγραμματιστή. Γνωστές βιβλιοθήκες για ανάλυση δεδομένων στην Python είναι τα Pandas, Numpy, Scipy, Matplot και StatsModels [22].

3.1 Pandas

Η βιβλιοθήκη Pandas είναι μια από τις πιο σημαντικές βιβλιοθήκες στην επιστήμη των δεδομένων. Πρόκειται για βιβλιοθήκη ανοικτού κώδικα (open-source) που διευκολύνει την ανάλυση, την επεξεργασία και τον καθαρισμό των δεδομένων. Η Pandas είναι προσβάσιμη σε όλους, ευέλικτη αλλά ταυτόχρονα εύκολη και γρήγορη στην χρήση τους. Η Pandas χρησιμοποιείται σε ένα ευρύ φάσμα ακαδημαϊκών και εμπορικών τομέων [22][23].

Κύρια σημεία της βιβλιοθήκης:

- Γρήγορο και αποδοτικό πλαίσιο δεδομένων (DataFrame) που βοηθάει στον χειρισμό δεδομένων,
- Εργαλεία για την ανάγνωση και εγγραφή δεδομένων μεταξύ δομών δεδομένων στη μνήμη και διαφορετικών μορφών (CSV, Excel files, SQL databases),
- Ολοκληρωμένος χειρισμός ελλείπων δεδομένων,
- Ευέλικτη αναδιαμόρφωση δεδομένων,
- Συγκέντρωση ή μετασχηματισμός δεδομένων με μια ισχυρή μηχανή ομαδοποίησης που επιτρέπει λειτουργίες διαχωρισμού-εφαρμογής-συνδυασμού σε σύνολα δεδομένων,
- Συγχώνευση και ένωση συνόλων δεδομένων

[23]

<code>import pandas as pd</code>	Εισαγωγή της βιβλιοθήκης Pandas με τον όνομα <code>pd</code>
<code>pd.read_csv()</code>	Είναι για την εισαγωγή ενός αρχείου CSV
<code>df.head()</code>	Εμφανίζει τις 5 πρώτες γραμμές των δεδομένων

df.tail()	Εμφανίζει τις 5 τελευταίες γραμμές των δεδομένων
df.info	Αυτή η συνάρτηση επιστρέφει μια γρήγορη σύνοψη του συνόλου δεδομένων
df.isnull()	Εμφανίζει σε ποιες γραμμές και στήλες υπάρχουν τιμές που λείπουν.
df.dropna()	Αφαίρει τις γραμμές στις οποίες λείπουν δεδομένα από το σύνολο δεδομένων
df.describe()	Παρουσίαση μια περιγραφική στατιστική του συνόλου δεδομένων
df.sample()	Επιστρέφει ένα τυχαίο δείγμα στοιχείων από έναν άξονα αντικειμένων.

Πίνακας 3.1 : Μερικές σημαντικές συναρτήσεις της βιβλιοθήκης Pandas [23]

3.2 Numpy

Είναι μια δημοφιλής βιβλιοθήκη μηχανικής μάθησης που υποστηρίζει μεγάλους πίνακες και πολυδιάστατα δεδομένα. Αποτελείται από ενσωματωμένες μαθηματικές συναρτήσεις για εύκολους υπολογισμούς. Η διεπαφή συστοιχιών είναι ένα από τα βασικά χαρακτηριστικά αυτής της βιβλιοθήκης [22][24].

Κύρια σημεία της βιβλιοθήκης:

- Παρέχει γρήγορες, προμεταγλωττισμένες συναρτήσεις για αριθμητικές ρουτίνες,
- Υπολογισμός με προσανατολισμό σε συστοιχίες για μεγαλύτερη αποδοτικότητα,
- Υποστηρίζει αντικειμενοστραφή προσέγγιση,
- Συμπαγείς και ταχύτεροι υπολογισμοί με διανυσματοποίηση

[24]

import numpy as np	Εισαγωγή της βιβλιοθήκης Numpy με τον όνομα np
ndarray.min()	Επιστρέφει την ελάχιστη τιμή κατά μήκος ενός συγκεκριμένου άξονα
ndarray.max()	Επιστρέφει την μέγιστη τιμή κατά μήκος ενός συγκεκριμένου άξονα
numpy.unique()	Επιστρέφει τα ταξινομημένα μοναδικά στοιχεία ενός πίνακα
numpy.mean()	Υπολογίζει τον αριθμητικό μέσο όρο
numpy.median()	Υπολογίζει τη διάμεσο
numpy.std()	Υπολογίζει την τυπική απόκλιση

Πίνακας 3.2 : Μερικές σημαντικές συνάρτησης της βιβλιοθήκης Numpy [25]

3.3 Matplotlib

Η Matplotlib είναι μια βιβλιοθήκη γραφικών παραστάσεων για Python. Οι απεικονίσεις των δεδομένων μπορεί να γίνει σε στατικά, κινούμενα αλλά και διαδραστικά διαγράμματα. Είναι μια βιβλιοθήκη που χρησιμοποιείται ως δωρεάν αντικαταστάτης της MATLAB [26].

Κύρια σημεία της βιβλιοθήκης:

- Δημιουργία διαδραστικών γραφικών σχημάτων που μπορούν να μεγεθυνθούν και πανοραμιστούν,
- Το στυλ και η διάταξη του γραφήματος προσαρμόζεται
- Εξαγωγή σε πολλές μορφές αρχείων.

[26]

<code>import matplotlib.pyplot as plt</code>	Εισαγωγή της βιβλιοθήκης matplotlib με τον όνομα <code>plt</code>
<code>plt.scatter()</code>	Σχεδιάζει ένα διάγραμμα διασποράς
<code>plt.Boxplot()</code>	Σχεδιάζει ένα θηκόγραμμα
<code>plt.plot()</code>	Σχεδιάζει τον άξονα y σε σχέση με τον άξονα x ως γραμμές ή/και δείκτες.
<code>plt.rcParams[]</code>	Αλλάζει δυναμικά τις προεπιλεγμένες ρυθμίσεις rc
<code>plt.figure</code>	Δημιουργεί ένα νέο σχήμα ή ενεργοποιεί ένα υπάρχον
<code>plt.axvline</code>	Προσθέτει μία κάθετη γραμμή στους άξονες
<code>plt.legend()</code>	Τοποθέτηση ενός υπομνήματος στο σχήμα
<code>plt.title()</code>	Ορίζει έναν τίτλο για το σχήμα

Πίνακας 3.3 : Μερικές σημαντικές συναρτήσεις της βιβλιοθήκης Matplotlib [26]

3.4 Seaborn

Η Seaborn είναι μια βιβλιοθήκη για οπτικοποίηση δεδομένων, βασισμένη στην Matplotlib. Προσφέρει υψηλού επιπέδου διεπαφές για σχεδίαση όμορφων και κατατοπιστικών γραφημάτων [27].

Κύρια σημεία της βιβλιοθήκης:

- Η Seaborn καθιστά την δημιουργία διαγραμμάτων πολύ εύκολή. Πολλές από τις οπτικοποιήσεις δημιουργούνται με μία γραμμή κώδικα.
- Έχει διάφορα ενσωματωμένα στυλ που μπορούν να ρυθμιστούν για να αλλάξουν την προεπιλεγμένη αισθητική σχεδίασης.
- Ίσως και το πιο σημαντικό χαρακτηριστικό της συγκεκριμένης βιβλιοθήκης. Ομαδοποιεί αυτόματα και συγκεντρώνει τα δεδομένα σε τακτοποιημένη μορφή. [28]

<code>import seaborn as sns</code>	Εισαγωγή της βιβλιοθήκης seaborn με το όνομα sns
<code>Sns.distplot()</code>	Δημιουργία διαγράμματος κατανομής
<code>Sns.boxplot()</code>	Δημιουργία θηκογράμματος
<code>Sns.heatmap()</code>	Δημιουργία διαγράμματος heatmap
<code>Sns.Set_style()</code>	Ορίζει αλλαγές στο στυλ του διαγράμματος .
<code>Sns.swampplot()</code>	Δημιουργία διαγράμματος swampplot
<code>Sns.Relplot()</code>	Used for creating relational plots
<code>Sns.Catplot()</code>	Used for creating categorical plots

Πίνακας 3.4 : Μερικές σημαντικές συναρτήσεις της βιβλιοθήκης Seaborn [27]

3.5 Statistics

Η Statistics είναι μια έτυμη μονάδα της Python, (python module) η οποία παρέχει έτυμα σχηματισμένες συναρτήσεις για τον εύκολο υπολογισμό μαθηματικών στατιστικών αριθμητικών δεδομένων. Απευθύνεται κύριως στο επίπεδο των γραφικών και επιστημονικών υπολογιστών. [30]

<code>mean()</code>	Υπολογίζει τον αριθμητικό μέσο
<code>median()</code>	Υπολογίζει την διάμεσο
<code>mode()</code>	Υπολογίζει την ποιο κοινή τιμή διακριτών ή ονομαστικών δεδομένων

quantiles()	Χωρίζει τα δεδομένα σε διαστήματα με ίσες πιθανότητες.
pstdev()	Υπολογίζει την πληθυσμιακή τυπική απόκλιση των δεδομένων.
stdev()	Υπολογίζει την δειγματική τυπική απόκλιση των δεδομένων.
variance()	Υπολογίζει την δειγματική διακύμανση των δεδομένων.
pvariance()	Υπολογίζει την πληθυσμιακή διακύμανση των δεδομένων.

Πίνακας 3.5 : Μερικές σημαντικές συναρτήσεις της βιβλιοθήκης Statistics[30]

3.6 warnings

Τα μηνύματα ειδοποίησης (warnings) εμφανίζονται συνήθως σε περιπτώσεις όπου είναι χρήσιμο να ειδοποιείται ο χρήστης για την κατάσταση στο πρόγραμμα. Οι καταστάσεις αυτές δεν δικαιολογεί την εμφάνιση μίας εξαίρεσης και τον τερματισμό του προγράμματος. [30]

Υπάρχει ένας αριθμός ενσωματωμένων εξαιρέσεων που αντιπροσωπεύουν κατηγορίες προειδοποιήσεων. Αυτή η κατηγοριοποίηση είναι χρήσιμη για να μπορείτε να φιλτράρετε ομάδες προειδοποιήσεων. [30]

Το φίλτρο προειδοποιήσεων ελέγχει αν οι προειδοποιήσεις αγνοούνται, εμφανίζονται ή μετατρέπονται σε σφάλματα δημιουργώντας μια εξαίρεση. [30]

Η εντολή `warnings.filterwarnings("ignore")` που χρησιμοποιείτε στον κώδικα, δεν επιτρέπει στα μηνύματα ειδοποιήσεων να εμφανίζονται. [30]

ΚΕΦΑΛΑΙΟ 4 Εφαρμογή- Δείκτης ποιότητας αέρα PM2.5

Στο κεφάλαιο 3, διερευνήσαμε τις μεθοδολογίες και τα εργαλεία που χρησιμοποιούνται για τη διεξαγωγή ολοκληρωμένης στατιστικής ανάλυσης δεδομένων στην Python. Από τη μία πλευρά, έχουμε βιβλιοθήκες όπως η Pandas και η Statistics που διευκολύνουν την υλοποίηση ενός μεγάλου εύρους ήδη κατασκευασμένων μαθηματικών εξισώσεων. Από την άλλη πλευρά, έχουμε βιβλιοθήκες που διευκολύνουν την απεικόνιση δεδομένων, όπως η Matplotlib και η Seaborn.

Η στατιστική ανάλυση δεδομένων είναι ένας σημαντικός παράγοντας σε μία έρευνα. Παρέχει ένα πλαίσιο για την εξαγωγή ουσιαστικών συμπερασμάτων από τα ακατέργαστα δεδομένα. Στόχος του συγκεκριμένου κεφαλαίου είναι να δείξει ότι η χρήση της Python καθιστά το βήμα της ανάλυσης δεδομένων σε μια έρευνα πολύ εύκολο. Η συγκεκριμένη ανάλυση επικεντρώνεται στη συγκέντρωση λεπτών σωματιδίων PM 2.5, ενός βασικού δείκτη για την ατμοσφαιρική ρύπανση που είναι γνωστό ότι έχει αρνητικές επιπτώσεις στην ανθρώπινη υγεία και το περιβάλλον.

4.1 Αναγνώριση του προβλήματος

Τις ημέρες που η ποιότητα του αέρα είναι καλή, ο αέρας είναι καθαρός και περιέχει μόνο μικρές ποσότητες στερεών σωματιδίων και χημικών ρύπων. Η κακή ποιότητα του αέρα, η οποία περιέχει υψηλά επίπεδα ρύπων, είναι συχνά θολός και επικινδύνως για την υγεία και το περιβάλλον. Η ποιότητα του αέρα περιγράφεται σύμφωνα με τον Δείκτη Ποιότητας του Αέρα (AQI Index), ο οποίος βασίζεται στη συγκέντρωση των ρύπων που υπάρχουν στον αέρα σε μια συγκεκριμένη τοποθεσία. Για τον λόγο ότι ο αέρας βρίσκεται συνεχόμενα σε κίνηση, η ποιότητά του μπορεί να μεταβληθεί από μέρα σε μέρα, ακόμα και σε μικρές χρονικές στιγμές όπως η ώρα.[31]

Ο ανθρώπινος παράγοντας παίζει καταλυτικό ρόλο σε αυτήν την μεταβολή του αέρα. Γεωγραφικά στοιχεία όπως τα βουνά, ακτογραμμές, και πεδιάδες που έχουν τροποποιηθεί από τον άνθρωπο, μπορεί να έχει ως αποτέλεσμα την συγκέντρωση ή και την διασπορά ατμοσφαιρικών ρύπων σε μια περιοχή. Ωστόσο, οι τύποι και οι ποσότητες των ρύπων που εισέρχονται στον αέρα έχουν πολύ μεγαλύτερο αντίκτυπο στην ποιότητα του αέρα. Φυσικές πηγές, όπως η ηφαιστειακή δραστηριότητα και οι καταιγίδες σκόνης, προσθέτουν ορισμένους ρύπους στον αέρα, αλλά οι περισσότεροι ρύποι προέρχονται από την ανθρώπινη δραστηριότητα. [31]

Μαζί με τον άνθρωπο, μεταβολή στην ποιότητα του αέρα δημιουργεί και ο ίδιος ο αέρας με την συμπεριφορά του. Ο αέρας με την συνεχόμενη κίνηση του μετακινεί ταυτόχρονα την ατμοσφαιρική κίνηση. Σε μια ορεινή ενδοχώρα μπορεί να έχει μεγαλύτερη ατμοσφαιρική

ρύπανση κατά τη διάρκεια της ημέρας, όταν η θαλάσσια αύρα ωθεί τους ρύπους πάνω από τη στεριά, και μικρότερη ατμοσφαιρική ρύπανση τα βράδια, επειδή η κατεύθυνση της αύρας αντιστρέφεται και ωθεί την ατμοσφαιρική ρύπανση πάνω από τον ωκεανό. [31]

Οι θερμοκρασίες μπορούν επίσης να επηρεάσουν τον αέρα. Όταν η θερμοκρασία του αέρα είναι χαμηλότερη, οι ρύποι των καυσαερίων μπορούν να παγιδευτούν κοντά στην επιφάνεια κάτω από ένα στρώμα πυκνού, ψυχρού αέρα. Τους καλοκαιρινούς μήνες, ο θερμαινόμενος αέρας ανεβαίνει και διασκορπίζει τους ρύπους από την επιφάνεια της Γης μέσω της ανώτερης τροπόσφαιρας. Ωστόσο, η αυξημένη ηλιακή ακτινοβολία έχει ως αποτέλεσμα πιο επιβλαβές όζον σε επίπεδο εδάφους. [31]

Η επιλογή της πόλης της Νέας Υόρκης για την ανάλυση της συγκέντρωσης των PM 2.5 και του δείκτη ποιότητας του αέρα (AQI) το 2022 δεν ήταν τυχαία. Η πόλη της Νέας Υόρκης αποτελεί ένα από τα σημαντικότερα αστικά περιβάλλοντα στον κόσμο, το οποίο χαρακτηρίζεται από τον πυκνό πληθυσμό, τις ποικίλες οικονομικές δραστηριότητες και τα εκτεταμένα δίκτυα μεταφορών. Αυτοί οι παράγοντες συμβάλλουν εγγενώς σε αυξημένα επίπεδα σωματιδιακής ύλης στην ατμόσφαιρα, καθιστώντας την ενδιαφέρουσα μελέτη περίπτωσης.

4.1.1 AQI

Η ποιότητα του αέρα αναφέρεται με τη χρήση του Δείκτη Ποιότητας Αέρα (AQI). Οι ρύποι που συνδέονται με επιπτώσεις στην ανθρώπινη και περιβαλλοντική υγεία περιλαμβάνουν τους δείκτες PM2.5, PM10, το όζον σε επίπεδο εδάφους, το διοξείδιο του αζώτου και το διοξείδιο του θείου. Τα παραπάνω στοιχεία χρησιμοποιούνται για τον υπολογισμό του δείκτη AQI. [32] [33]

Ο δείκτης AQI έχει έξι κατηγορίες που επικοινωνούν το επίπεδο ανησυχίας για την υγεία χρησιμοποιώντας συγκεκριμένα χρώματα. Ο κωδικός Πράσινο και Κίτρινο σημαίνει ότι ο αέρας είναι γενικά ασφαλής για όλους. Ο κωδικός πορτοκαλί είναι ανθυγιεινός για ευαίσθητες ομάδες, συμπεριλαμβανομένων των παιδιών, των ηλικιωμένων και των ατόμων με καρδιακές και πνευμονικές παθήσεις. Ο κωδικός Κόκκινο και Μωβ σημαίνει ότι ο αέρας είναι ανθυγιεινός για όλους, και ο κωδικός Καστανό είναι μια προειδοποίηση υγείας για καταστάσεις έκτακτης ανάγκης. [32], [33]

Daily AQI Color	Levels of Concern	Values of Index	Description of Air Quality
Green	Good	0 to 50	Air quality is satisfactory, and air pollution poses little or no risk.
Yellow	Moderate	51 to 100	Air quality is acceptable. However, there may be a risk for some people, particularly those who are unusually sensitive to air pollution.
Orange	Unhealthy for Sensitive Groups	101 to 150	Members of sensitive groups may experience health effects. The general public is less likely to be affected.
Red	Unhealthy	151 to 200	Some members of the general public may experience health effects; members of sensitive groups may experience more serious health effects.
Purple	Very Unhealthy	201 to 300	Health alert: The risk of health effects is increased for everyone.
Maroon	Hazardous	301 and higher	Health warning of emergency conditions: everyone is more likely to be affected.

Εικόνα 24: Πίνακας κατηγοριών AQI

4.1.2 PM2.5

Τα αιωρούμενα σωματίδια (PM) δεν είναι ένας μόνο ρύπος, αλλά μάλλον ένα μείγμα πολλών χημικών ειδών. Πρόκειται για ένα πολύπλοκο μείγμα στερεών και αερολυμάτων που αποτελείται από μικρά σταγονίδια υγρού, ξηρά στερεά θραύσματα και στερεούς πυρήνες με υγρή επικάλυψη. Τα σωματίδια ποικίλλουν ευρέως ως προς το μέγεθος, το σχήμα και τη χημική σύνθεση και μπορεί να περιέχουν ανόργανα ιόντα, μεταλλικές ενώσεις, στοιχειακό άνθρακα, οργανικές ενώσεις και ενώσεις από τον φλοιό της γης. Τα σωματίδια ορίζονται με βάση τη διάμετρό τους για τους ρυθμιστικούς σκοπούς της ποιότητας του αέρα. Αυτά με διάμετρο 10 μικρόμετρα (Microns) ή μικρότερη (PM10) εισπνέονται στους πνεύμονες και μπορούν να προκαλέσουν δυσμενείς επιπτώσεις στην υγεία. Ως λεπτόκοκκα σωματίδια ορίζονται τα σωματίδια με διάμετρο 2,5 μικρά (Microns) ή μικρότερη (PM2,5). Επομένως, τα PM2.5 αποτελούν μέρος των PM10. [34]

Η συνεχή έκθεση στο PM2.5 έχει συνδεθεί με πρόωρη θνησιμότητα, αυξημένες εισαγωγές σε νοσοκομεία για καρδιακά ή πνευμονικά αίτια, οξεία και χρόνια βρογχίτιδα, κρίσεις άσθματος, επισκέψεις στα επείγοντα περιστατικά, αναπνευστικά συμπτώματα και περιορισμένες ημέρες δραστηριότητας. Αυτές οι δυσμενείς επιπτώσεις στην υγεία έχουν αναφερθεί κυρίως σε βρέφη, παιδιά και ηλικιωμένους με προϋπάρχουσες καρδιακές ή πνευμονικές παθήσεις. [34]

4.2 Καθαρισμός Δεδομένων

Στο προηγούμενο κεφάλαιο δόθηκε έμφαση στην θεμελιώδη σημασία της κατανόησης του προβλήματος πριν από την έναρξη οποιασδήποτε στατιστικής ανάλυσης δεδομένων. Για ποιους λόγους είναι σημαντικό να παρακολουθούνται οι μεταβολές στις μετρήσεις του AQI και ποιο συγκεκριμένα για το PM 2.5. Αυτό το κρίσιμο αρχικό βήμα θέτει τις βάσεις για ουσιαστικές γνώσεις και τεκμηριωμένη

λήψη αποφάσεων. Ωστόσο, η έναρξη της ανάλυσης δεδομένων χωρίς να αντιμετωπιστεί η ακεραιότητα και η ποιότητα των υποκείμενων δεδομένων μπορεί να δημιουργήσει σημαντικές προκλήσεις και να οδηγήσει σε παραπλανητικές συμπεράσματα.

Ο καθαρισμός δεδομένων είναι ένα σημαντικό πρώιμο βήμα στη διαδικασία ανάλυσης δεδομένων. Αυτό το βήμα, που περιλαμβάνει την προετοιμασία και την επικύρωση των δεδομένων, πραγματοποιείται πριν το στάδιο της ανάλυσης. Ο καθαρισμός των δεδομένων είναι η διαδικασία διόρθωσης εσφαλμένων, ελλιπών, διπλών ή άλλων λανθασμένων δεδομένων σε ένα σύνολο δεδομένων. Αποτελεί ένα από τα βασικά στοιχεία των εργασιών προετοιμασίας δεδομένων που προετοιμάζουν τα σύνολα δεδομένων για χρήση σε εφαρμογές επιχειρηματικής ευφυΐας (BI) και επιστήμης δεδομένων. [35]

Προκειμένου να έχουμε ένα σωστό αποτέλεσμα στην ανάλυση, τα δεδομένα πρέπει να είναι καθαρά και αυτό είναι ο κύριος λόγος που καθιστά αυτό το βήμα πολύ σημαντικό. Ο καθαρισμός των δεδομένων βοηθάει ακόμα και στην παραγωγικότητα, οργάνωση και επιλύσει προβλημάτων σε επόμενα βήματα της ανάλυσης δεδομένων. [35]

4.2.1 Αφαίρεση διπλών εγγραφών

Τα διπλά δεδομένα είναι ένα από τα πιο συνηθισμένα προβλήματα στην ποιότητα των δεδομένων και συχνά παραβλέπονται. Τα διπλά δεδομένα (duplicate data) είναι εγγραφές που μοιράζονται τις ίδιες τιμές για ορισμένες ή όλες τις μεταβλητές ενός συνόλου δεδομένων. Οι εγγραφές αυτές οφείλονται σε σφάλματα κατά την καταχώρηση τους, από την συγχώνευση ή προσθήκη δεδομένων από διαφορετικές πηγές ασυνεπείς συμβάσεις μορφοποίησης ή ονοματοδοσίας, ελλιπή ή διφορούμενα αναγνωριστικά, ή προβλήματα απόξεσης ή ανίχνευσης δεδομένων. Τα δεδομένα αυτά μπορούν να επηρεάσουν την ποιότητα, την ακρίβεια και την αξιοπιστία των δεδομένων και να οδηγήσουν σε ανακριβή αποτελέσματα στην ανάλυση ή και την μοντελοποίηση. [36] [38]

```
#Removing Duplicates  
df=df.drop_duplicates()  
df
```

Εικόνα 25: Αφαίρεση διπλών εγγραφών με την χρήση της drop_duplicate.

Η drop_duplicate() είναι μια από τις πολλές συναρτήσεις που ανήκουν στην βιβλιοθήκη Panda. Η παραπάνω συνάρτηση χρησιμοποιείτε για να επιστρέψει το σύνολο δεδομένων αφού αφαιρεθούν τα διπλά δεδομένα. Υπάρχει η δυνατότητα μέσω παραμέτρων να εξευρεθούν κάποιες στήλες από αυτών των έλεγχο. Σε περίπτωση που στα δεδομένα υπάρχουν δείκτες, όπως είναι οι χρονικοί δείκτες, τους αγνοεί και δεν τους περιλαμβάνει στον έλεγχο. [38]

4.2.2 Αντιμετώπιση τιμών NULL

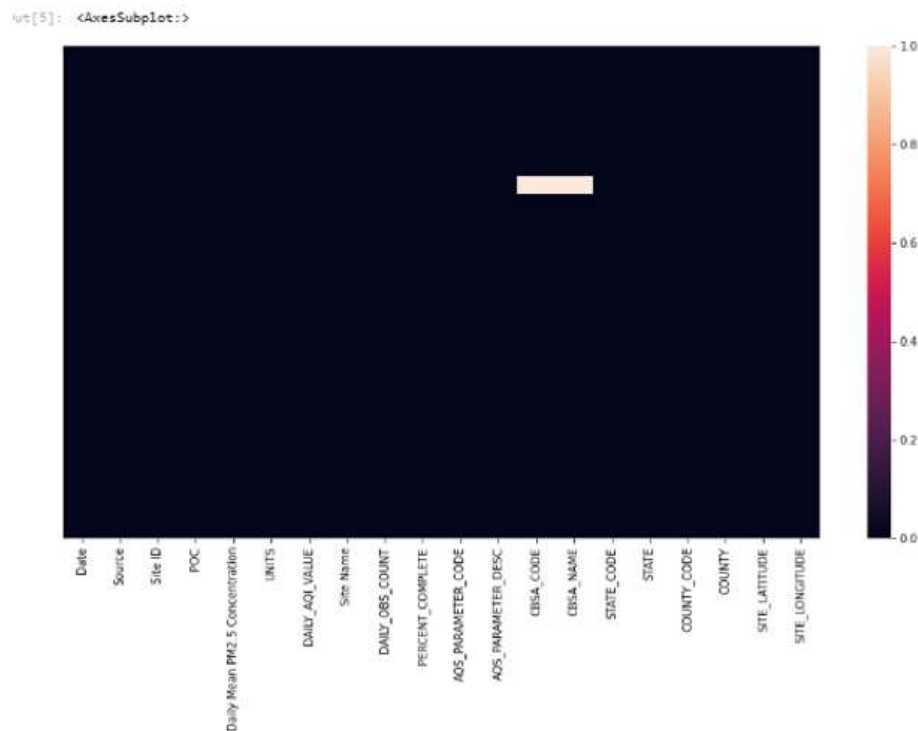
Η τιμή NULL είναι ένας ειδικός δείκτης που χρησιμοποιείται στην SQL για να δηλώσει ότι μια τιμή δεδομένων δεν υπάρχει στη βάση δεδομένων. Με άλλα λόγια, είναι απλώς ένας χώρος τοποθέτησης για να δηλωθούν τιμές που λείπουν ή που είναι άγνωστες. Λόγω αυτής της δομής των τιμών NULL, δεν είναι δυνατή η χρήση των παραδοσιακών τελεστών σύγκρισης (=, <, > και <>).

Με το παρακάτω κομμάτι κώδικα γίνεται ο έλεγχος για αυτές τις τιμές NULL με δύο τρόπους. Με την χρήση της συνάρτησης isnull() των Pandas, η αναζήτηση των τιμών null.

```
# examining missing values
print("Missing values distribution: ")
print(df.isnull().mean())
print("")
sns.heatmap(df.isnull(),yticklabels=False)
```

Εικόνα 26: Έλεγχος τιμών null με την χρήση της is_null.

Με την χρήση ενός θερμικού χάρτη (heat map) είναι πολύ εύκολος ο εντοπισμός των ελλιπών τιμών και σε ποιες στήλες βρίσκονται. Ακόμα και σε περιπτώσεις που πολλαπλές στήλες έχουν κάποιον συσχετισμό σε τιμές που λείπουν, η συσχέτιση θα οπτικοποιηθεί.



Εικόνα 27: Θερμικός χάρτης για τον εντοπισμό ελλιπών στοιχείων.

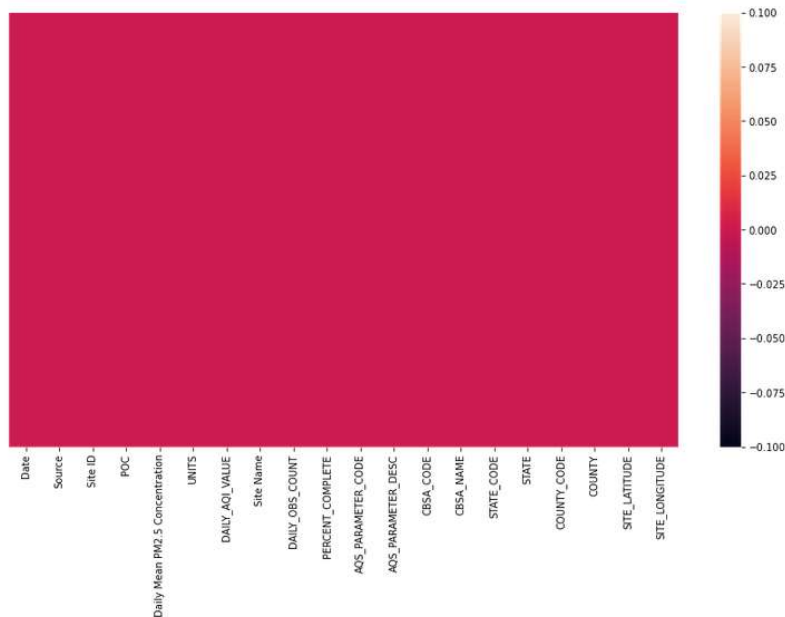
Με την χρήση της συνάρτησης dropna(), της Pandas, οι γραμμές με τα ελλιπή στοιχεία διαγράφονται.

```
df=df.dropna()  
df
```

Εικόνα 28: Αφαίρεση γραμμών που περιέχουν NULL τιμές με την εντολή dropna().

```
sns.heatmap(df.isnull(),yticklabels=False)
```

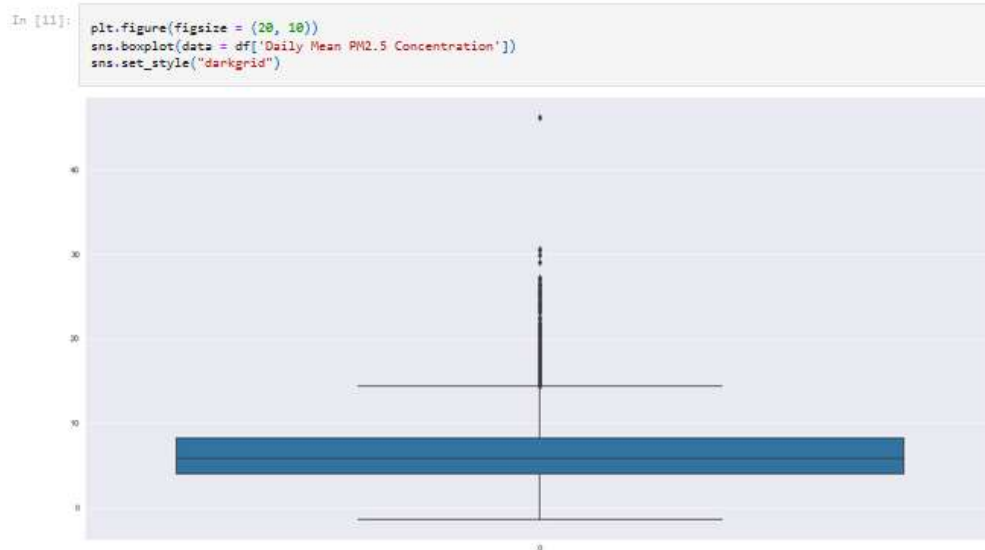
<AxesSubplot:>



Εικόνα 29: Θερμικός χάρτης μετά από την αφαίρεση γραμμών με NULL τιμές.

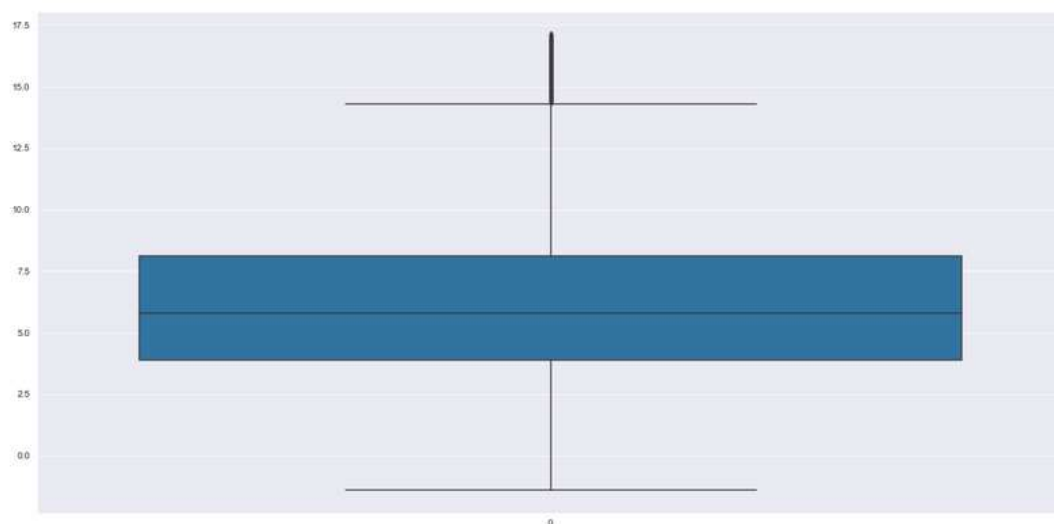
4.2.3 Ακραίες τιμές

Ακραίες τιμές είναι τα δεδομένα που βρίσκεται σε μη φυσιολογική απόσταση από άλλες τιμές σε ένα τυχαίο δείγμα από ένα πλήθος. Με ποιο απλά λόγια, μια ακραία τιμή είναι ένα εξαιρετικά υψηλό ή εξαιρετικά χαμηλό σημείο δεδομένων σε σχέση με το πλησιέστερο σημείο δεδομένων και τις υπόλοιπες γειτονικές τιμές που συνυπάρχουν σε ένα γράφημα δεδομένων ή ένα σύνολο δεδομένων.



Εικόνα 30: Δημιουργία θηκογράμματος (boxplot) για να δείξουμε ότι υπάρχουν ακραίες τιμές.

Για τον εντοπισμό των ακραίων τιμών χρησιμοποιήθηκε η κανονική κατανομή. Η κανονική κατανομή υπολογίζει την διασπορά των δεδομένων γύρω από την μέση τιμή. Για δεδομένα τα οποία βρίσκονται στο 99.7%, θα βρίσκονται μέσα στα όρια της τρίτης κανονικής κατανομή, δηλαδή στο διάστημα $(\bar{x} - 3s, \bar{x} + 3s)$. Τα δεδομένα που βρίσκονται εκτός των ορίων θεωρούνται ακραίες τιμές και αφαιρούνται.



Εικόνα 31: Δημιουργία θηκογράμματος (boxplot) για να δείξουμε ότι αφαιρέθηκαν οι ακραίες τιμές.

4.3 Περιγραφική στατιστική

Η περιγραφική στατιστική, εκτός από τους πίνακες συχνοτήτων και τα διαγράμματα, είναι και τα λεγόμενα *στατιστικά μέτρα* που συνοψίζουν ένα σύνολο δεδομένων, το οποίο μπορεί να είναι είτε μια αναπαράσταση ολόκληρου του συνόλου είτε ενός δείγματός του. Η περιγραφική στατιστική αναλύεται σε μέτρα κεντρικής τάσης και μέτρα μεταβλητότητας (διασποράς). Τα μέτρα κεντρικής τάσης περιλαμβάνουν τη μέση τιμή, τη διάμεσο και την επικρατούσα τιμή, ενώ τα μέτρα μεταβλητότητας περιλαμβάνουν την τυπική απόκλιση, τη διακύμανση, την ελάχιστη και τη μέγιστη μεταβλητή, την κύρτωση και τη ασυμμετρία. [39]

Η περιγραφική στατιστική χρησιμοποιείται για να επαναπροσδιοριστούν οι δυσνόητες ποσοτικές πληροφορίες από ένα πολύ μεγάλο μέγεθος σύνολο δεδομένων, σε ποιο μικρά κομμάτια ευκολότερα στην κατανόηση. Ο μέσος όρος μαθημάτων (M.O.) ενός φοιτητή, για παράδειγμα, παρέχει μια καλή κατανόηση της περιγραφικής στατιστικής. Η ιδέα στο συγκεκριμένο παράδειγμα είναι ότι παίρνει σημεία δεδομένων από ένα ευρύ φάσμα εξετάσεων μαθημάτων και βαθμών και τα υπολογίζει κατά μέσο όρο για να παρέχει μια γενική κατανόηση της συνολικής ακαδημαϊκής επίδοσης ενός φοιτητή. Ο προσωπικός M.O. ενός φοιτητή αντικατοπτρίζει τη μέση ακαδημαϊκή του επίδοση. [39]

4.3.1 Διαγράμματα

Ένα από τα πιο αποτελεσματικά εργαλεία για την οπτική αναπαράσταση δεδομένων είναι το διάγραμμα. Υπάρχει μια ποικιλία διαφορετικών τύπων διαγραμμάτων, καθένας από τους οποίους είναι κατάλληλος για συγκεκριμένο σκοπό.

4.3.1.1 Ραβδογράμματα

Τα ραβδογράμματα είναι το πιο συνηθισμένο οπτικό υλικό που χρησιμοποιείται και αυτό σημαίνει μικρότερη καμπύλη εκμάθησης για το κοινό. Τα μάτια συγκρίνουν τα τελικά σημεία των ράβδων, και είναι εύκολο να δουν γρήγορα ποια κατηγορία είναι η μεγαλύτερη, ποια η μικρότερη και επίσης τη σημαντική διαφορά μεταξύ των κατηγοριών. Στο παρακάτω παράδειγμα εξετάζεται ο μέσος όρος των μετρήσεων του PM 2.5 για το έτος 2022 για όλες τις κομητείες της Νέας Υόρκης. Συγκρίνοντας τα ύψη των ράβδων διακρίνεται ότι το Κινγκς είχε τις υψηλότερες μετρήσεις σε αντίθεση με την κομητεία Ρόκλαντ που είχε τις χαμηλότερες.

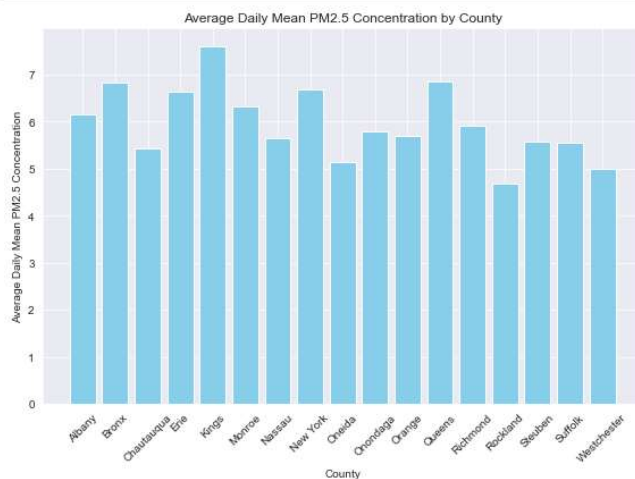
```

county_mean_concentration = df.groupby('COUNTY')['Daily Mean PM2.5 Concentration'].mean()

plt.figure(figsize=(8, 6))
bars = plt.bar(county_mean_concentration.index, county_mean_concentration.values, color='skyblue')

plt.xlabel('County')
plt.ylabel('Average Daily Mean PM2.5 Concentration')
plt.title('Average Daily Mean PM2.5 Concentration by County')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

```



Εικόνα 32: Ραβδογράμμο που απεικονίζει την μέτρηση Daily Mean PM2.5 Concentration για κάθε κομητεία της Νέας Υόρκης με την χρήση Python.

4.3.1.2 Διάγραμμα Χρονοσειράς

Τα γραφήματα χρονοσειράς χρησιμοποιούνται συνήθως για την απεικόνιση συνεχών δεδομένων. Επειδή τα σημεία συνδέονται μέσω γραμμής, υπονοείται μια σύνδεση μεταξύ των σημείων που μπορεί να μην έχει νόημα για κατηγορικά δεδομένα. Συχνά, τα συνεχή δεδομένα είναι σε κάποια μονάδα χρόνου: ημέρες, ώρες, χρόνια ή μήνες. Στο παρακάτω παράδειγμα παρουσιάζεται μια γραμμή που δείχνει την μεταβολή του πεδίο Daily Mean PM 2.5 για κάθε μήνα για το έτος 2022. Είναι εμφανές ότι ο Ιανουάριος ήταν ο μήνας με τον υψηλότερο δείκτη για το PM 2.5, που πιθανώς επηρεάζεται από τις αυξημένες απαιτήσεις θέρμανσης κατά τη διάρκεια του χειμώνα, ενώ ο Απρίλιος είχε τις χαμηλότερες μετρήσεις. Αξίζει να σημειωθεί ότι, οι εποχιακές διακυμάνσεις έδειξαν χαμηλότερα επίπεδα PM 2.5 κατά τη διάρκεια της άνοιξης και του φθινοπώρου σε σύγκριση με το καλοκαίρι και το χειμώνα, με το χειμώνα να παρουσιάζει τα υψηλότερες συγκεντρώσεις.

```

import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.dates import MonthLocator, DateFormatter

df['Date'] = pd.to_datetime(df['Date'])

monthly_data = df.groupby(df['Date'].dt.to_period('M')).mean()

monthly_df = pd.DataFrame({
    'Month': monthly_data.index.to_timestamp(),
    'Mean PM2.5 Concentration': monthly_data['Daily Mean PM2.5 Concentration']
})

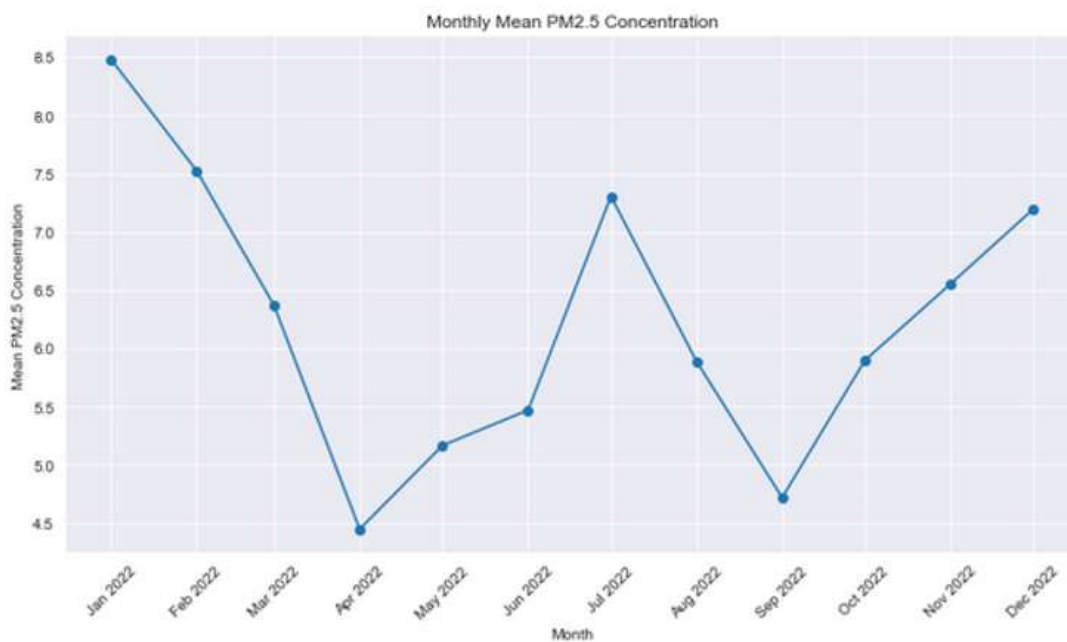
plt.figure(figsize=(10, 6))
plt.plot(monthly_df['Month'], monthly_df['Mean PM2.5 Concentration'], marker='o', linestyle='-')

plt.title('Monthly Mean PM2.5 Concentration')
plt.xlabel('Month')
plt.ylabel('Mean PM2.5 Concentration')

plt.gca().xaxis.set_major_locator(MonthLocator())
plt.gca().xaxis.set_major_formatter(DateFormatter("%b %Y"))

plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()

```



Εικόνα 33: Διάγραμμα γραμμής που απεικονίζει την μέτρηση Daily Mean PM2.5 Concentration για κάθε μήνα του 2022 για την Νέα Υόρκη με την χρήση Python.

4.3.1.3 Κυκλικό Διάγραμμα

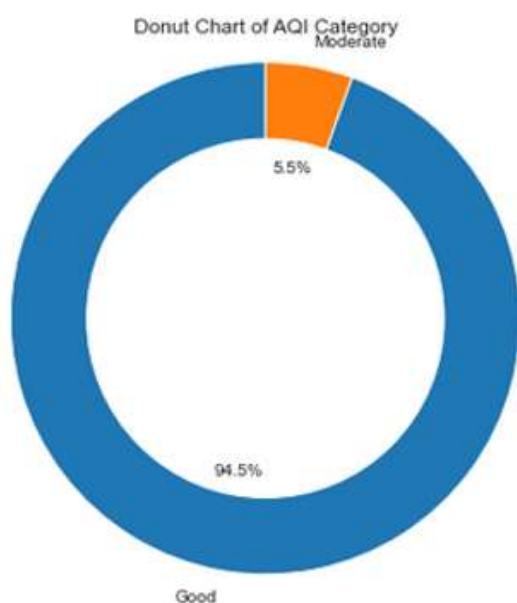
Ένα κυκλικό διάγραμμα δείχνει συνήθως τις αναλογίες των κατηγορικών δεδομένων. Ο ακροατής καλείται να συγκρίνει το μήκος τόξου με ένα άλλο μήκος τόξου, για να συγκρίνει τα ποσοστά κάθε κατηγορίας. Για παράδειγμα, ο δείκτης AQI χωρίζεται σε έξι κύριες κατηγορίες. Καλή (0-50), Μέτρια (51-100), Ανθυγιεινή για ευαίσθητες ομάδες (101-150), Ανθυγιεινή (150-200), Πολύ ανθυγιεινή (201-300) και Επικίνδυνη (300-500). Στα δεδομένα μας μπορούμε να δούμε από το κυκλικό διάγραμμα ότι η Νέα Υόρκη το 2022 είχε μόνο για τις κατηγορίες Καλή και Μέτρια, με την πλειοψηφία να είναι στην καλή κατηγορία. Με 94,5% των μέτρων στην κατηγορία «καλό» και μόνο 5,5% στην κατηγορία «μέτριο», αυτό δείχνει ότι τα μέτρα που έχει λάβει η Νέα Υόρκη για τη βελτίωση της ποιότητας του αέρα είναι καλά.

```
category_counts = df['AQI_Category'].value_counts()

# Plotting
plt.figure(figsize=(5, 5))
plt.pie(category_counts, labels=category_counts.index, autopct='%1.1f%%', startangle=90)

centre_circle = plt.Circle((0,0),0.70,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)

plt.title('Donut Chart of AQI Category')
plt.axis('equal')
plt.tight_layout()
plt.show()
```



Εικόνα 34: Κυκλικό διάγραμμα που απεικονίζει την το ποσοστό επί του συνόλου για το AQI Category για την Νέα Υόρκη το 2022 με την χρήση Python.

4.3.1.4 Χάρτης θερμότητας

Μια μέθοδος για την ενσωμάτωση των λεπτομερειών ενός πίνακα με ταυτόχρονη αξιοποίηση των οπτικών ενδείξεων είναι η χρήση ενός χάρτη θερμότητας. Διαβάζοντας το παρακάτω πίνακα, είναι εμφανές δύο πράγματα. Αρχικά, οι μετρήσεις για την κατηγορία Καλή είναι εμφανές περισσότερες σε σχέση με την κατηγορία Μέτρια. Επιπλέον, πρωτοπόρα κοινότητα στις μετρήσεις για την καλή κατηγορία είναι η κοινότητα Έρι με 1237 μετρήσεις και την ακολουθεί η Κουίνς με 1069 μετρήσεις. Παρότι συνολικά οι μετρήσεις ήταν πολύ ικανοποιητικές, παρατηρήθηκαν κάποιες χωρικές ιδιαιτερότητες. Περιοχές όπως το Τσαουτάκουα και το Ρόκλαντ παρουσιάζουν μικρό αριθμό μετρήσεων στις δύο αυτές κατηγορίες. Μεγάλη σημασία έχουν οι περιοχές των Κουίνς και των Κίνγκς, οι οποίες σύμφωνα με τις ποσότητες των μετρήσεων που ανήκουν στην κατηγορία «καλή», παραμένουν επίσης προγονικές περιοχές για την κατηγορία «μέτρια».

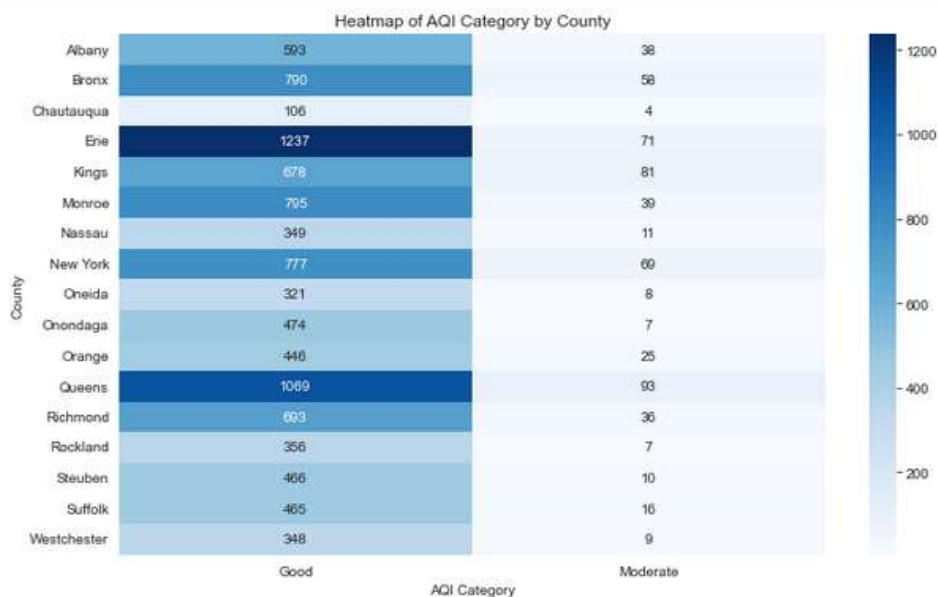
```

pivot_df = df.pivot_table(index='COUNTY', columns='AQI_Category', aggfunc='size', fill_value=0)

# Plotting
plt.figure(figsize=(10, 6))
sns.heatmap(pivot_df, annot=True, cmap='Blues', fmt='g')

plt.title('Heatmap of AQI Category by County')
plt.xlabel('AQI Category')
plt.ylabel('County')
plt.tight_layout()
plt.show()

```



Εικόνα 35: Θερμικός χάρτης που απεικονίζει τον αριθμό μετρήσεων σε κάθε AQI Category για τις κομητείες της Νέας Υόρκης το 2022 με την χρήση Python.

4.3.2 Κεντρική Τάση

Η κεντρική τάση είναι μια περιγραφική σύνοψη ενός συνόλου δεδομένων μέσω μιας ενιαίας τιμής που αντικατοπτρίζει το κέντρο της κατανομής των δεδομένων. Η κεντρική τάση είναι μία από τις πιο βασικές έννοιες της στατιστικής. Παρόλο που δεν παρέχει πληροφορίες σχετικά με τις μεμονωμένες τιμές στο σύνολο δεδομένων, παρέχει μια ολοκληρωμένη σύνοψη του συνόλου των δεδομένων.[5]

4.3.2.1 Μέση τιμή (Mean Value)

Η μέση τιμή (mean value) ορίζεται με το άθροισμα όλων των μετρήσεων δια του πλήθους των μετρήσεων. [5]

$$\bar{x} = \frac{1}{n} \left(\sum_{i=1}^n x_i \right) = \frac{x_1 + x_2 + \dots + x_n}{n}$$

Όπου : $\bar{x}$ = δειγματικός μέσος

x_i = η i-τιμή της μεταβλητής

n = μέγεθος δείγματος

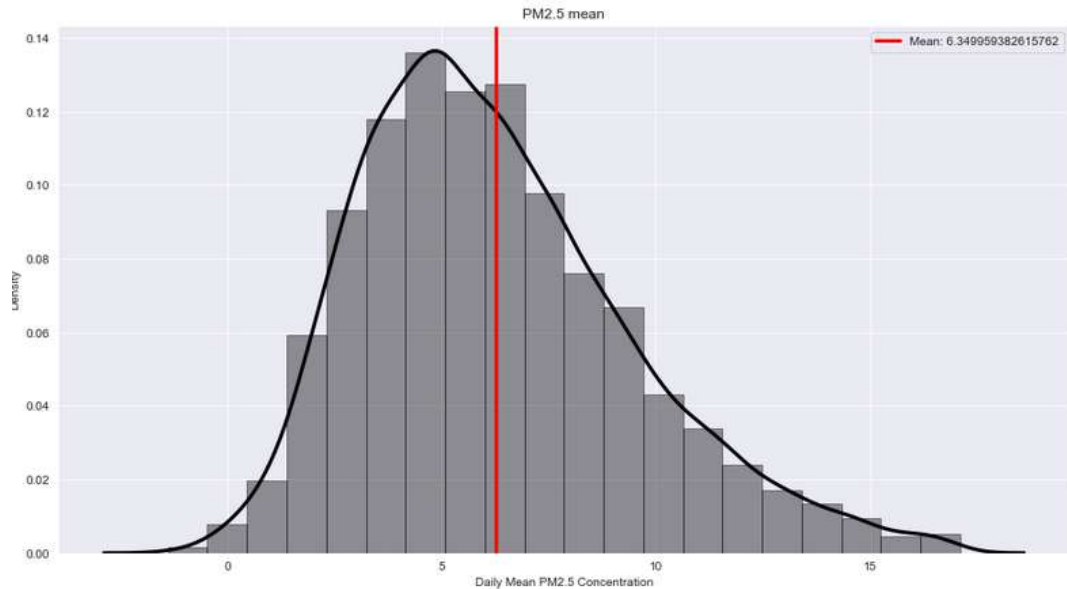
Για τον στατιστικό πληθυσμό το n θα αντιστοιχεί στο N , το οποίο απεικονίζει το πλήθος στοιχείων στατιστικού πληθυσμού.

Η βιβλιοθήκη Pandas διαθέτει μία έτοιμη υλοποιημένη συνάρτηση για την εύρεση της μέσης τιμής, την **mean()**. Στο παρακάτω παράδειγμα χρησιμοποιείται η συνάρτηση για να υπολογιστεί η μέση τιμή από την στήλη 'Daily Mean PM2.5 Concentration' του συνόλου των δεδομένων.

```
df['Daily Mean PM2.5 Concentration'].mean()
6.250896159317207
```

Εικόνα 36: Συνάρτηση mean() στην Python.

Από το αποτέλεσμα αυτό καταλαβαίνουμε ότι η μέση τιμή του PM2.5 για την Νέα Υόρκη για το 2022 είναι 6.250 $\mu\text{g}/\text{m}^3$ LC.



Εικόνα 37: Μέση τιμή σε ιστόγραμμα.

4.3.2.2 Διάμεσος (Median)

Η διάμεσος (median) μιας πεπερασμένης λίστας αριθμών είναι ο "μεσαίος" αριθμός, όταν οι αριθμοί αυτοί παρατίθενται κατά σειρά από τον μικρότερο προς τον μεγαλύτερο. Εάν το σύνολο δεδομένων έχει περιττό αριθμό παρατηρήσεων, επιλέγεται η μεσαία. Για παράδειγμα, ο ακόλουθος κατάλογος επτά αριθμών, 1, 3, 3, 6, 7, 8, 9 έχει διάμεσο το 6, που είναι η τέταρτη τιμή. Εάν το σύνολο δεδομένων έχει ζυγό αριθμό παρατηρήσεων, δεν υπάρχει διακριτή μεσαία τιμή και η διάμεσος ορίζεται συνήθως ως ο αριθμητικός μέσος των δύο μεσαίων τιμών. Για παράδειγμα, αυτό το σύνολο δεδομένων με 8 αριθμούς 1, 2, 3, 4, 5, 6, 8, 9 έχει διάμεση τιμή 4,5, δηλαδή $(4 + 5) / 2$. [5]

Αν το μέγεθος δείγματος (n) είναι περιττός αριθμός τότε ο μαθηματικός υπολογισμός είναι (εικόνα)

$$median(x) = x_{\frac{(n+1)}{2}}$$

Αν μέγεθος δείγματος (n) είναι άρτιος αριθμός τότε ο τύπος για τον υπολογισμό μετατρέπεται σε

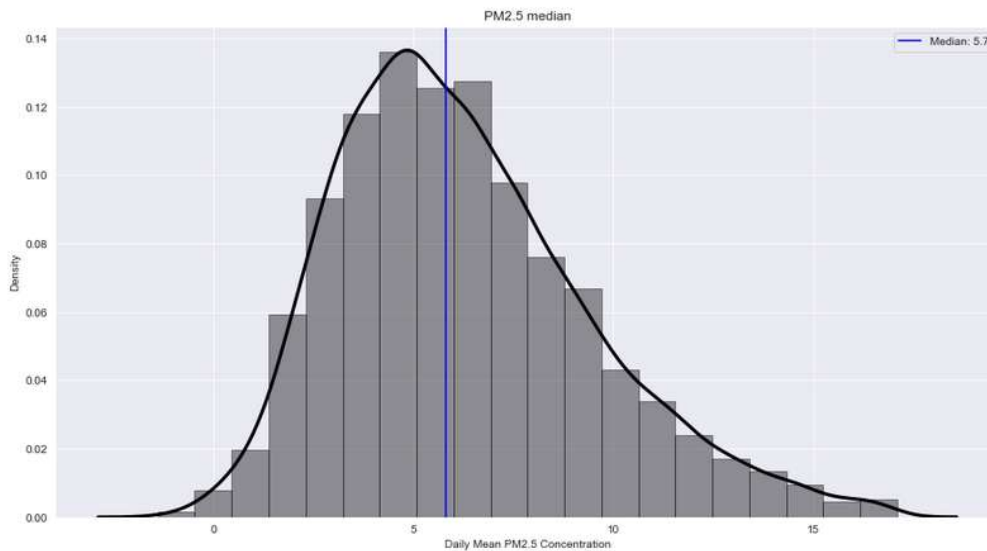
$$\text{median}(x) = \frac{x_{(\frac{n}{2})} + x_{((\frac{n}{2})+1)}}{2}$$

Η βιβλιοθήκη Pandas διαθέτει την συνάρτηση **median()** για την εύρεση της διάμεσου. Εδώ η διάμεσος είναι της στήλης ‘Daily Mean PM2.5 Concentration’ ισούται με 5.8.

```
df['Daily Mean PM2.5 Concentration'].median()
```

5.8

Εικόνα 38: Χρήση της συνάρτησης median() για την διάμεσο.



Εικόνα 38: Διάμεσος σε ιστόγραμμα.

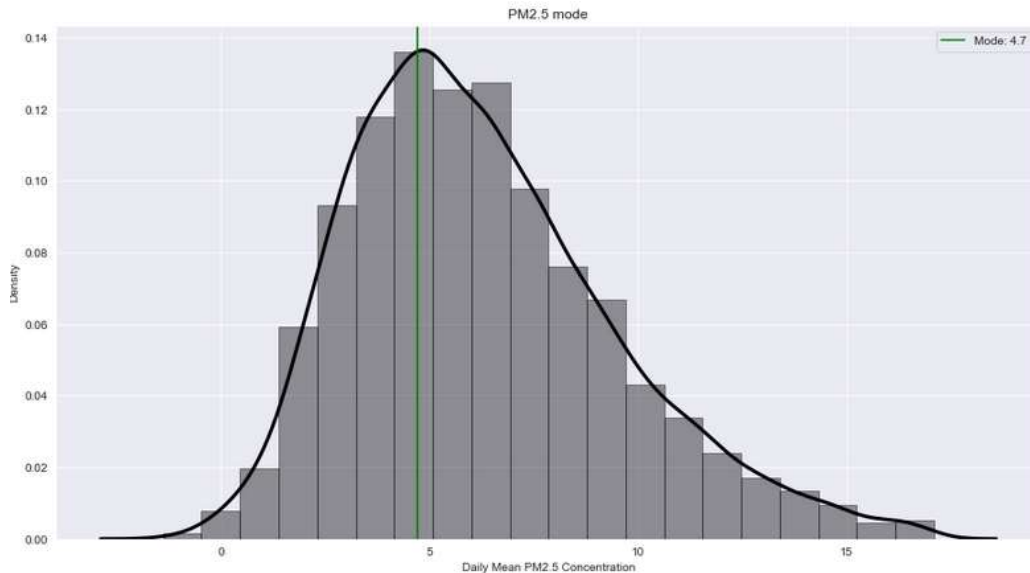
4.3.2.3 Επικρατούσα τιμή (Mode)

Η επικρατούσα τιμή (mode) είναι η τιμή που εμφανίζεται με την μεγαλύτερη συχνότητα σε ένα σύνολο δεδομένων. Ένα σύνολο δεδομένων μπορεί να έχει μία επικρατούσα τιμή, καμία ή περισσότερες από μία επικρατούσες τιμές. Στην συγκεκριμένη περίπτωση η επικρατούσα τιμή είναι η 4,7.[5]

```
df['Daily Mean PM2.5 Concentration'].mode()[0]
```

4.7

Εικόνα 39: Χρήση της συνάρτησης `mode()` για να βρούμε την μεγαλύτερη συχνότητα στα δεδομένα.



Εικόνα 40: Επικρατούσα τιμή σε ιστόγραμμα.

4.3.3 Διασπορά

Η μεταβλητότητα, σχεδόν εξ ορισμού, είναι ο βαθμός στον οποίο τα σημεία δεδομένων σε μια στατιστική κατανομή ή σε ένα σύνολο δεδομένων αποκλίνουν -από τη μέση τιμή, καθώς και ο βαθμός στον οποίο αυτά τα σημεία δεδομένων διαφέρουν μεταξύ τους. Όταν οι τιμές μιας μεταβλητής δεν διαφέρουν πολύ από την μέση τιμή αποκαλούνται ως ομοιογενείς, και έχει ως αποτέλεσμα την μικρή διασπορά. Στην άλλη πλευρά, όσο οι τιμές απομακρύνονται από την μέση τιμή τους τόσο αυξάνεται και η διασπορά. [5]

4.3.3.1 Εύρος (Range)

Εύρος είναι η διαφορά μεταξύ μέγιστης και ελάχιστης τιμής. Εξαρτάται μόνο από δύο τιμές, την μέγιστη και την ελάχιστη μεταβλητή, για την κατανομή των μετρήσεων.

$$Range(x) = Max(x) - Min(x)$$

Για τον υπολογισμό της μέγιστης τιμής σε μία στήλη, χρησιμοποιείτε η συνάρτηση `max()` και αντίστοιχα για την ελάχιστη τιμή η `min()`.

```
df['Daily Mean PM2.5 Concentration'].max() - df['Daily Mean PM2.5 Concentration'].min()
18.5
```

Εικόνα 41: Εύρος στην Python.

Στην στήλη Daily Mean PM 2.5 Concentration που υπάρχει στο σύνολο των δεδομένων εμφανίζει ένα εύρος ίσο με 18.5.

4.3.3.2 Ενδοτεταρτημοριακό εύρος Q (Interquartile range)

Το ενδοτεταρτημοριακό εύρος είναι η πιο αξιόπιστη μέτρηση διασποράς από το εύρος. Το ενδοτεταρτημοριακό εύρος υπολογίζεται από τις μισές μετρήσεις της κατανομής και όχι μόνο από δύο τιμές. [5]

$$IQR = Q_3 - Q_1$$

Η Pandas με την συνάρτηση describe δίνει την δυνατότητα στην γρήγορη περιγραφή ενός συνόλου δεδομένων ή σε συγκεκριμένες στήλες.

```
df['Daily Mean PM2.5 Concentration'].describe()
count    10545.000000
mean      6.250896
std       3.171304
min      -1.400000
25%       3.900000
50%       5.800000
75%       8.100000
max      17.100000
Name: Daily Mean PM2.5 Concentration, dtype: float64
```

Εικόνα 42: Συνάρτηση describe() στην Python.

Με την αφαίρεση του τρίτου και του πρώτου τεταρτημρίου έχουμε ως αποτέλεσμα του IQR να είναι 4.19.

```
df['Daily Mean PM2.5 Concentration'].quantile(0.75) - df['Daily Mean PM2.5 Concentration'].quantile(0.25)
4.199999999999999
```

Εικόνα 43: Ενδοτεταρτημοριακό εύρος στην Python.

4.3.3.3 Διακύμανση (Variance)

Η διακύμανση ή μέση τετραγωνική απόκλιση είναι μία στατιστική μέτρηση μεταξύ των αριθμών σε ένα σύνολο δεδομένων. Η διακύμανση υπολογίζει το πόσο απέχει κάθε αριθμός ενός συνόλου δεδομένων από την μέση τιμή. Υπολογίζεται με τη λήψη των διαφορών μεταξύ κάθε αριθμού στο σύνολο δεδομένων και του μέσου όρου, στη συνέχεια με τον τετραγωνισμό των διαφορών για να γίνουν θετικές. Τέλος, με τη διαίρεση του αθροίσματος των τετραγώνων με τον αριθμό των τιμών στο σύνολο δεδομένων. Η διακύμανση συμβολίζεται με το σ^2 .

$$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{N}$$

Όπου : x_i = Κάθε τιμή στο σύνολο δεδομένων

$\bar{x}$ = Η μέση τιμή ολόκληρου του συνόλου δεδομένων

N = Ο αριθμός των τιμών στο σύνολο δεδομένων

Όταν ο υπολογισμός γίνεται σε ένα δείγμα του συνόλου δεδομένων, ο παρανομαστής παίρνει την μορφή n-1 που ονομάζεται και ως "βαθμοί ελευθερίας".

$$s^2 = \frac{\sum_{i=2}^n (x_i - \bar{X})^2}{n - 1}$$

Η διακύμανση χρησιμοποιείται για να δείξει πως οι μεμονωμένοι αριθμοί σχετίζονται μεταξύ τους σε ένα σύνολο δεδομένων, αντί να χρησιμοποιούνται ευρύτερες μαθηματικές τεχνικές, όπως η ταξινόμηση αριθμών σε τεταρτημόρια. Το πλεονέκτημα της διακύμανσης είναι ότι αντιμετωπίζει όλες τις αποκλίσεις από την μέση τιμή ως ίδιες, ανεξάρτητα από την κατεύθυνσή τους. Οι τετραγωνικές αποκλίσεις δεν μπορούν να αθροίσουν στο μηδέν και να δώσουν την εντύπωση ότι δεν υπάρχει καθόλου μεταβλητότητα στα δεδομένα.

Στην Python για τον υπολογισμό της διακύμανσης χρησιμοποιείται η συνάρτηση **variance()** της βιβλιοθήκης statistics. Η διακύμανση του PM 2.5 στο σύνολο δεδομένων είναι 10.06.

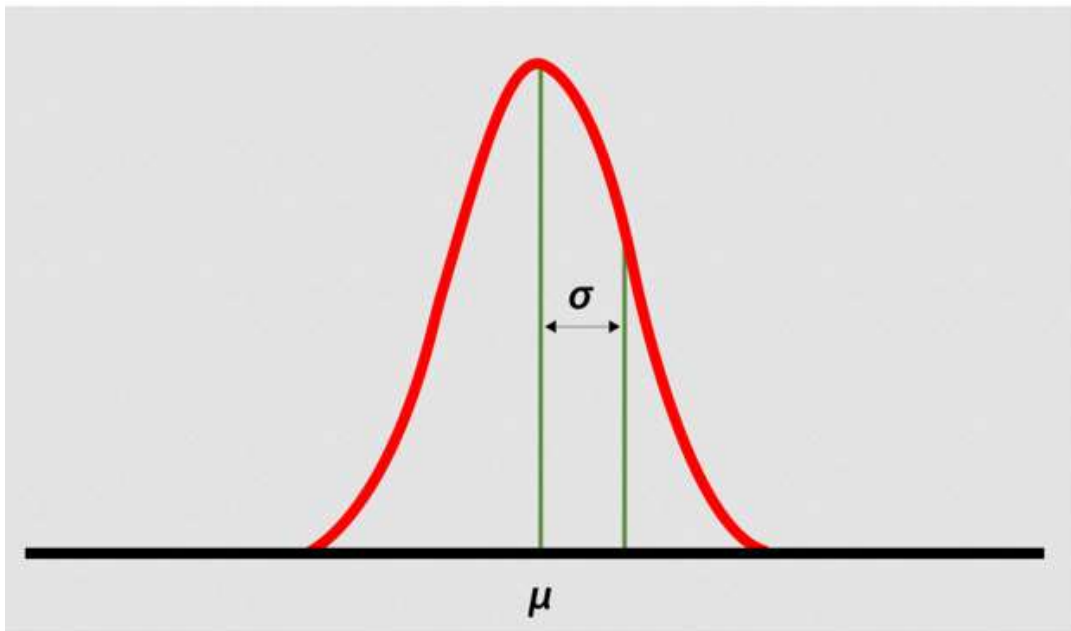
```
statistics.variance(df['Daily Mean PM2.5 Concentration'])
```

```
10.057167491587107
```

Εικόνα 44: Διακύμανση στην Python

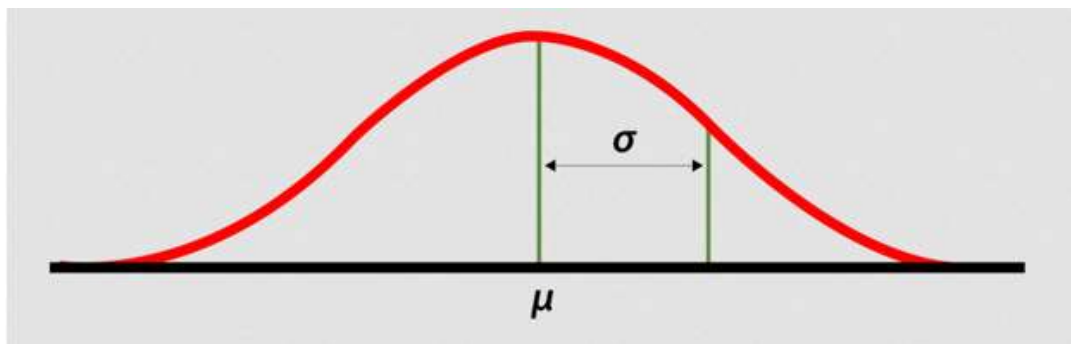
4.3.3.4 Τυπική απόκλιση (standard deviation)

Η τυπική απόκλιση, η οποία συμβολίζεται με σ ή s , είναι ένα μέτρο που δείχνει πόσο διασκορπισμένα είναι τα δεδομένα στο σύνολό τους. Η χαμηλή ή μικρή τυπική απόκλιση υποδηλώνει ότι τα δεδομένα είναι συγκεντρωμένα κοντά στη μέση τιμή.



Εικόνα 45: Διάγραμμα χαμηλής τυπικής απόκλισης.

Σε αντίθεση η υψηλή ή μεγάλη τυπική απόκλιση υποδηλώνει ότι τα δεδομένα είναι περισσότερο διασκορπισμένα.



Εικόνα 46: Διάγραμμα υψηλής τυπικής απόκλισης

Ο υπολογισμός της τυπικής απόκλισης γίνεται από το τύπο

$$\sigma = \sqrt{\sigma^2}$$

Στην περίπτωση που ο υπολογισμός γίνεται από ένα δείγμα και όχι σε ολόκληρο το πλήθος δεδομένων, ο υπολογισμός μετατρέπεται και στη θέση του N μπαίνει το N-1

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Με την χρήση της συνάρτησης `std()`, η Python, υπολογίζει την κανονική κατανομή της επιλεγμένης στήλης. Για να προκύψει η δεύτερη και η τρίτη τυπική απόκλιση, το αποτέλεσμα πρέπει να πολλαπλασιαστεί με τους αριθμούς δύο και τρία, αντίστοιχα.

```
df['Daily Mean PM2.5 Concentration'].std()
```

```
3.171303752652369
```

```
2*df['Daily Mean PM2.5 Concentration'].std()
```

```
6.342607505304738
```

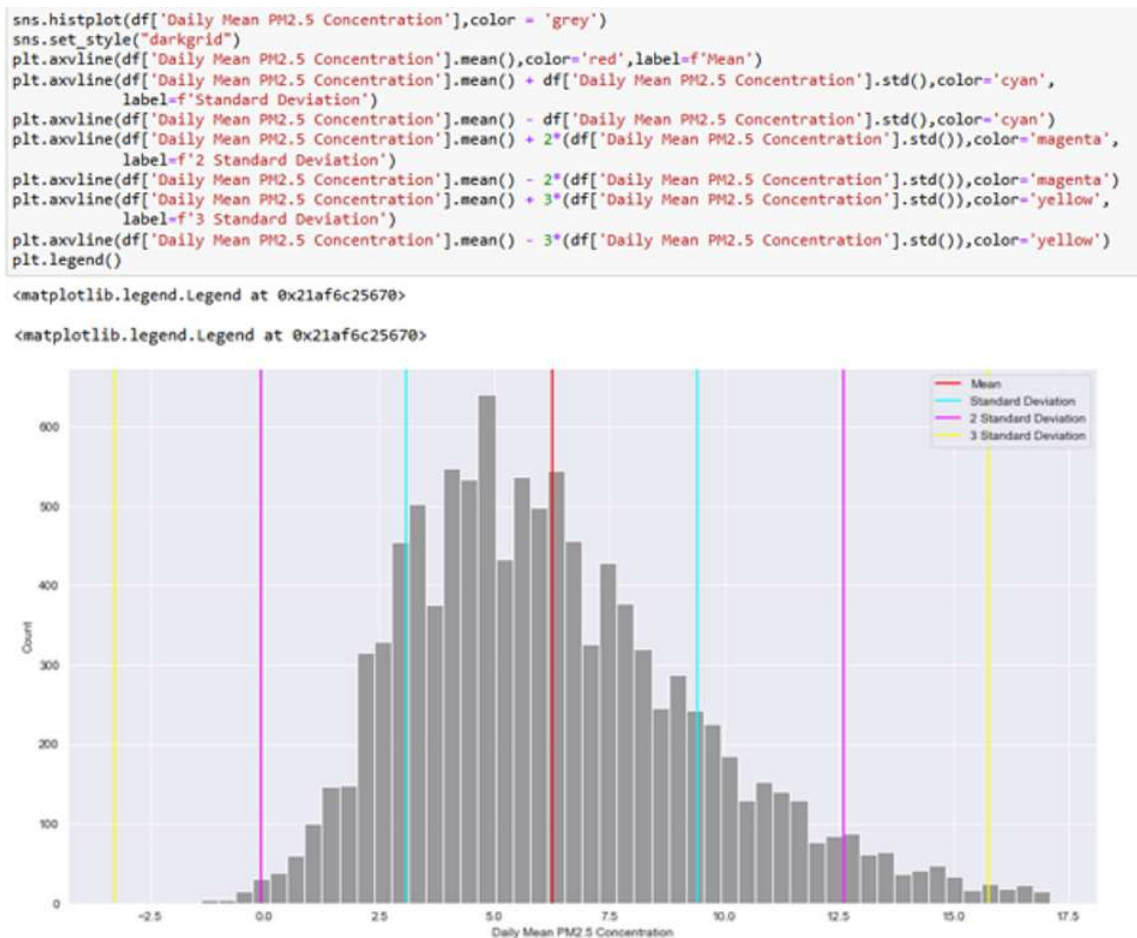
```
3*df['Daily Mean PM2.5 Concentration'].std()
```

```
9.513911257957107
```

Εικόνα 47: Πρώτη, δεύτερη και τρίτη τυπική απόκλιση στην Python

Για να γίνει ποιο κατανοητή η σημασία των κανονικών κατανομών, η οπτικοποίηση τους σε ένα ιστόγραμμα είναι απαραίτητη. Καθώς η κανονική κατανομή είναι η συγκέντρωση των τιμών γύρω από την μέση τιμή, τα διαστήματα που πρέπει να υπολογιστούν για την οπτικοποίηση των κατανομών είναι

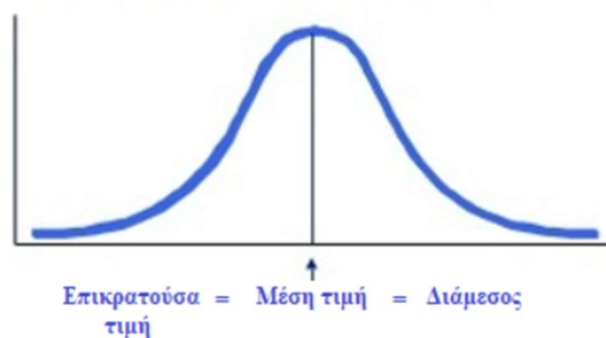
$(\bar{x} - s, \bar{x} + s)$ για την πρώτη κανονική κατανομή, $(\bar{x} - 2s, \bar{x} + 2s)$ για την δεύτερη κανονική κατανομή και $(\bar{x} - 3s, \bar{x} + 3s)$ για την τρίτη κατανομή.



Εικόνα 48: Πρώτη, δεύτερη και τρίτη τυπική απόκλιση σε ιστόγραμμα.

4.3.3.5 Ασυμμετρία (Skewness)

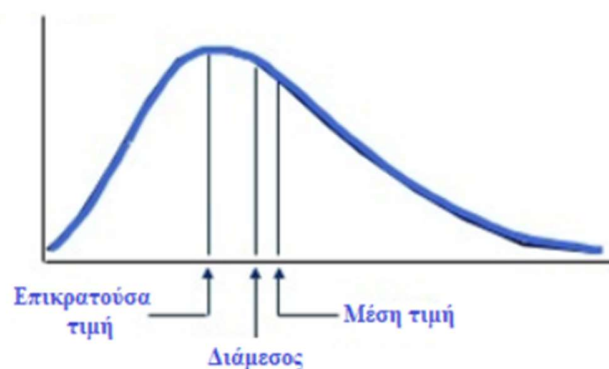
Επιπλέον ιδιότητα ενός συνόλου δεδομένων είναι ο τρόπος με τον οποίον κατανέμονται τα δεδομένα. Για να υπάρχει μια συμμετρική κατανομή η διάμεσος και η μέση τιμή πρέπει να είναι ίσες ($\bar{x} = M$). Στην περίπτωση που αυτό δεν ισχύει, τότε οι η κατανομή χαρακτηρίζεται ως θετικά ή αρνητικά ασύμμετρη.



ΣΥΜΜΕΤΡΙΚΗ ΚΑΤΑΝΟΜΗ

Εικόνα 49: Καμπύλη συμμετρικής κατανομής

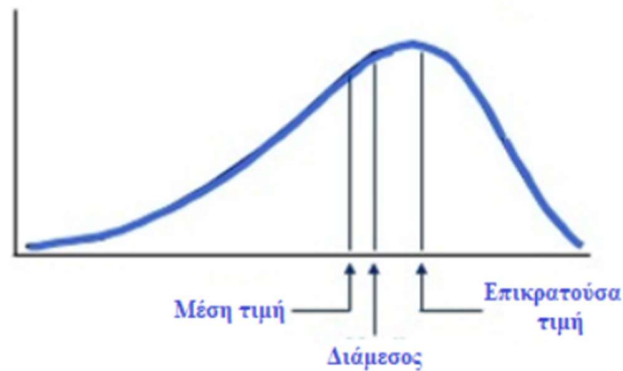
Στην θετική ασυμμετρία μιας κατανομής, οι περισσότερες τιμές του συνόλου είναι συγκεντρωμένες στο αριστερό άκρο της κατανομής. Σε μια θετική ασυμμετρία η επικρατούσα τιμή έχει μικρότερη τιμή από την διάμεσο, με την μέση τιμή να ξεπερνάει τις υπόλοιπες ($M_o < M < \bar{x}$).



ΘΕΤΙΚΗ ΑΣΣΥΜΕΤΡΙΑ

Εικόνα 50: Καμπύλη θετικής ασυμμετρίας

Στην αρνητική ασυμμετρία οι τιμές συσσωρεύονται στο δεξιό μέρος της κορυφής. Στην αρνητική ασυμμετρία η επικρατούσα τιμή έχει την υψηλότερη τιμή. Αμέσως μετά ακολουθεί η διάμεσος και με την μικρότερη τιμή είναι η μέση τιμή ($\bar{x} < M < M_o$).



Εικόνα 51: Καμπύλη αρνητικής ασσυμετρίας

Με την χρήση της συνάρτησης **skew()**, παρατηρείται ότι οι τιμές του PM 2.5 έχει τον συντελεστή β_1 ίσο με 0.68.

```
df['Daily Mean PM2.5 Concentration'].skew()
0.6816696038861196
```

Εικόνα 52: Συνάρτηση skew() στην Python

Ο συντελεστής β_1 καθορίζει άμα η στήλη παρουσιάζει συμμετρία ή κάποιο είδος ασσυμετρίας.

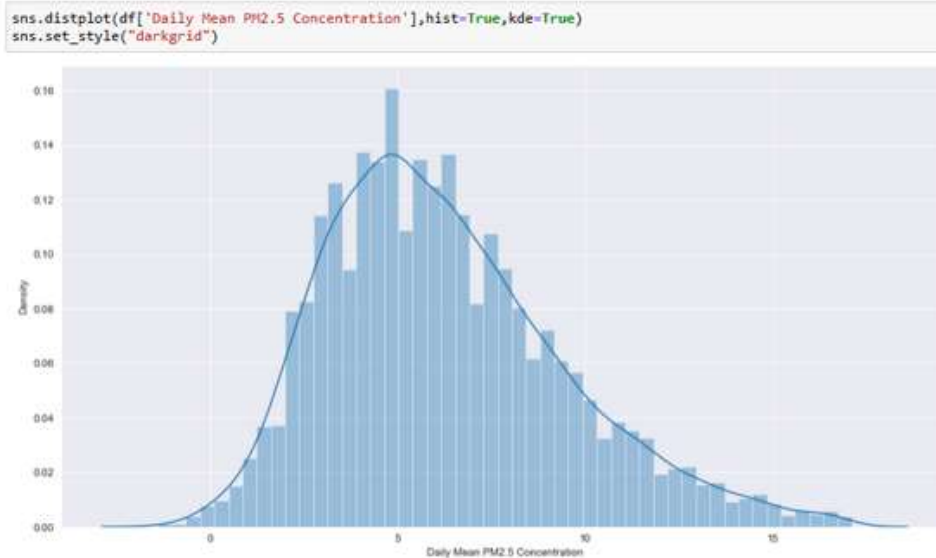
$$\beta_1 = \frac{\mu_3^2}{s^2}$$

Όπου μ_3 είναι η 3^η κεντρική ροπή και υπολογίζεται από τον τύπο

$$\mu_3 = \frac{\sum (x_i - \bar{x})^3}{n}$$

- $\beta_1 = 0$ η κατανομή θα είναι συμμετρική
- $\beta_1 > 0$ η κατανομή θα παρουσιάζει θετική ασσυμετρία
- $\beta_1 < 0$ η κατανομή θα παρουσιάζει αρνητική ασσυμετρία

Η στήλη daily Mean PM 2.5 Concentration έχει μεγαλύτερη μέση τιμή από ότι διάμεσο και επικρατούσα τιμή οπότε μπορούμε να διαπιστώσουμε ότι τα δεδομένα μας έχουν θετική ασυμμετρία.



Εικόνα

53: Ιστόγραμμα για το PM 2.5.

4.3.3.6 Κύρτωση (Kurtosis)

Κύρτωση είναι το επίπεδο συγκέντρωσης των τιμών γύρω από την μέση τιμή και προς τα άκρα της κατανομής. Οι κατανομές με μεγάλη συγκέντρωση των τιμών κοντά στην μέση τιμή ονομάζονται λεπτόκυρτες. Πλατύκυρτες είναι οι κατανομές που οι τιμές είναι απλωμένες εκατέρωθεν της μέσης τιμής. Τέλος, οι κανονικές κατανομές χαρακτηρίζονται ως μεσόκυρτες.

Ο συντελεστής κυρτότητας Pearson υπολογίζει την κυρτότητα β_4

$$\beta_4 = \frac{\mu_4}{S}$$

Όπου μ_4 είναι η 4^η κεντρική ροπή

$$\mu_4 = \frac{\sum (x_i - \bar{x})^4}{n}$$

```
df['Daily Mean PM2.5 Concentration'].kurt()  
8.2886239943354787
```

Εικόνα 54: Συνάρτηση kurt() στην Python.

4.4 Επαγωγική στατιστική

Η **επαγωγική στατιστική** ή **στατιστική συμπερασματολογία** ασχολείται με τη μελέτη των χαρακτηριστικών και την ανάλυση των δεδομένων ενός δείγματος, με στόχο την εξαγωγή συμπερασμάτων για τον πληθυσμό της έρευνας. Η επαγωγική στατιστική δίνει την δυνατότητα στον εντοπισμό τάσεων και μοτίβων σε μεγάλα σύνολα δεδομένων. Η παροχή πληθυσμιακών εκτιμήσεων και ο έλεγχος υποθέσεων για την εξαγωγή συμπερασμάτων σχετικά με το πλήθος, είναι οι κύριοι λόγοι που χρησιμοποιείται αυτή η μορφή στατιστικής.

4.4.1 Έλεγχος υποθέσεων

Στην στατιστική συμπερασματολογία ανήκει ο έλεγχος υποθέσεων. Χρησιμοποιεί τα δεδομένα ενός δείγματος για να εξάγει συμπεράσματα σχετικά με μια παράμετρο του πληθυσμού ή μια κατανομή πιθανότητας του πληθυσμού. Δημιουργούνται δύο υποθέσεις, η μηδενική υπόθεση (H_0) και η εναλλακτική υπόθεση (H_a). Ο έλεγχος περιλαμβάνει δειγματικά δεδομένα για να προσδιορίσει-για να ελέγξει αν μπορεί η H_0 να απορριφθεί ή όχι. Εάν η H_0 απορριφθεί, το στατιστικό συμπέρασμα είναι ότι η εναλλακτική υπόθεση H_a είναι αληθής. Ο έλεγχος υποθέσεων χρησιμοποιείται συνήθως στην επιστημονική έρευνα, στον ποιοτικό έλεγχο και στις διαδικασίες λήψης αποφάσεων για τη λήψη τεκμηριωμένων κρίσεων και την εξαγωγή στατιστικών συμπερασμάτων [40].

Σε ένα στατιστικό έλεγχο τα βήματα είναι καθορισμένα:

1. **Διατύπωση των υποθέσεων:** Η μηδενική υπόθεση H_0 και η εναλλακτική υπόθεση H_a . Η υπόθεση του ελέγχου είναι η μηδενική υπόθεση και οποιαδήποτε άλλη υπόθεση ονομάζεται εναλλακτική υπόθεση.
2. **Υπολογισμός του στατιστικού ελέγχου (Test Statistic):** Το test statistic είναι μία αριθμητική τιμή η οποία υπολογίζεται από τα δεδομένα. Η τιμή του στατιστικού ελέγχου μας λέει αν αποδεχόμαστε τη μηδενική υπόθεση ή την απορρίπτουμε και αποδεχόμαστε την εναλλακτική υπόθεση ως ορθή. Ορισμένοι από τους ελέγχους που χρησιμοποιούνται για τον υπολογισμό αυτής της τιμής είναι η ANOVA, το t-test, το chi-square test. Το κριτήριο που χρησιμοποιείται εξαρτάται από τα ερωτήματα που πρέπει να απαντηθούν και τον τύπο των δεδομένων που είναι διαθέσιμα.
3. **Ορισμός του επιπέδου σημαντικότητας (α):** Το επίπεδο σημαντικότητας είναι η πιθανότητα να απορριφθεί η μηδενική υπόθεση ενώ είναι σωστή. Συνήθως

συμβολίζεται σε ποσοστιαία τιμή ή σε δεκαδική αριθμητική τιμή, όπως 5% (0.05) ή 1% (0.01).

4. **Απόφαση:** Αφού ολοκληρωθεί ο στατιστικός έλεγχος, ως αποτέλεσμα θα προκύψουν δύο τιμές, η τιμή test statistic και η τιμή p-value. Τέλος, θα πρέπει να συγκριθεί η τιμή p-value με το επίπεδο σημαντικότητας για να απορριφθεί ή όχι η μηδενική υπόθεση.

4.4.1.1 ANOVA

Το τεστ ANOVA, ή ανάλυση διακύμανσης, είναι μια μέθοδος που χρησιμοποιείται στην στατιστική που βασίζεται στην ανάλυση της συνολικής διακύμανσης που παρατηρείται σε ένα σύνολο δεδομένων. Κρίνει κατά πόσον οι παρατηρούμενες διαφορές μεταξύ πολλαπλών μέσων τιμών από τις ομάδες είναι στατιστικά σημαντικές. Το τεστ ANOVA θα δείξει αν η διακύμανση είναι διαφορετική, αλλά όχι ποια είναι η διαφορά. Στο συγκεκριμένο παράδειγμα θέλει να συγκρίνει τις κατηγορίες του πεδίου AQI_Category και το μετρήσιμο θα είναι το πεδίο Daily Mean PM 2.5 Concentration.

```
aqi_ranges = [(0, 50), (51, 100), (101, 150), (151, 200), (201, 300), (301, 500)]
aqi_categories = ['Good', 'Moderate', 'Unhealthy for Sensitive Groups', 'Unhealthy', 'Very Unhealthy', 'Hazardous']

def categorize_aqi(aqi_value):
    for i, (lower, upper) in enumerate(aqi_ranges):
        if lower <= aqi_value <= upper:
            return aqi_categories[i]

df['AQI_Category'] = df['DAILY_AQI_VALUE'].apply(categorize_aqi)

print(df[['DAILY_AQI_VALUE', 'AQI_Category']])
```

	DAILY_AQI_VALUE	AQI_Category
0	13	Good
1	33	Good
2	16	Good
3	25	Good
4	52	Moderate
...	...	...
11073	35	Good
11074	39	Good
11075	53	Moderate
11076	58	Moderate
11078	54	Moderate

[10545 rows x 2 columns]

Εικόνα 55: Κομμάτι κώδικα για την δημιουργία της στήλης AQI_Category.

Η μηδενική υπόθεση (H_0) είναι η υπόθεση ότι όλες οι κατηγορίες στα δεδομένα να έχουν ίδια μέση τιμή.

$$H_0: \mu_1 = \mu_2 = \mu_3 = \dots = \mu_n$$

Η εναλλακτική υπόθεση (H_a) είναι το αντίθετο, δηλαδή ότι υπάρχει τουλάχιστον μια κατηγορία με διαφορετική μέση τιμή από τις υπόλοιπες.

$$H_a : \text{Not all } \mu_i \text{ are equal}$$

Το στατιστικό τεστ που χρησιμοποιεί ο έλεγχος ANOVA είναι κατανομή F.

$$F = \frac{\text{Διακύμανση εντός της ομάδας}}{\text{Διακύμανση μεταξύ των ομάδων}}$$

Χρησιμοποιήθηκε η συνάρτηση **f_oneway()**, της βιβλιοθήκης `scipy.stats`, για να γίνει το τεστ ONE WAY ANOVA. Στην μηδενική υπόθεση (H_0) ορίζεται η μέση τιμή των κατηγοριών που διακρίνεται το πεδίο `AQI_Category`, το οποίο δημιουργήθηκε παραπάνω, να είναι ίδιες.

Μετά την εκτέλεση της ANOVA με τη χρήση της `f_oneway()`, το αποτέλεσμα περιλαμβάνει συνήθως το στατιστικό F και την τιμή p. Η τιμή p-τιμής συγκρίνεται με ένα επιλεγμένο επίπεδο σημαντικότητας (π.χ. 0,05) για να καθοριστεί αν πρέπει να απορριφθεί η μηδενική υπόθεση. Εάν η τιμή p-value είναι μικρότερη από το επίπεδο σημαντικότητας, απορρίπτουμε τη μηδενική υπόθεση και συμπεραίνουμε ότι υπάρχουν στατιστικά σημαντικές διαφορές μεταξύ των μέσων όρων των ομάδων. Για να χρησιμοποιηθεί στην Python η συνάρτηση `f_oneway()` χρειάζεται να δοθούν και τα δύο πεδία για να γίνει η σύγκριση. Συγκρίνει την συγκέντρωση των τιμών της στήλης `Daily Mean PM 2.5 Concentration` με τις κατηγορίες της στήλης `AQI_Category`.

Με το παρακάτω κομμάτι κώδικα χωρίζουμε την στήλη `DAILY_AQI_VALUE` από το σύνολο δεδομένων σε μία νέα στήλη `AQI_Category`. Ο διαχωρισμός που επιλέχθηκε έγινε από την κατηγοριοποίηση που ήδη υπάρχει για το AQI.

```
from scipy.stats import f_oneway

grouped_states = [df[df['AQI_Category'] == state]['Daily Mean PM2.5 Concentration'] for state in df['AQI_Category'].unique()]

f_stat, p_value_anova = f_oneway(*grouped_states)

print(f'F-Statistic: {f_stat}')
print(f'P-Value (ANOVA): {p_value_anova}')

if p_value_anova < 0.05:
    print('There is a significant difference in mean concentrations among states.')
else:
    print('There is no significant difference in mean concentrations among states.')

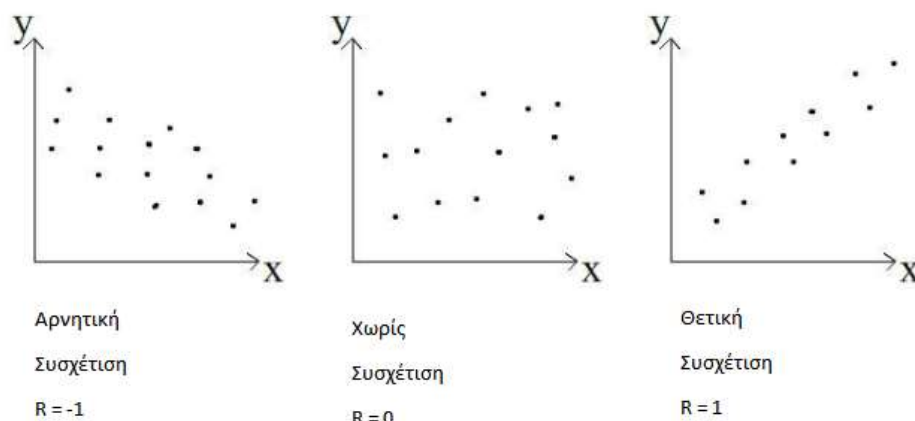
F-Statistic: 5268.09955996962
P-Value (ANOVA): 0.0
There is a significant difference in mean concentrations among states.
```

Εικόνα 56: Κομμάτι κώδικα για την χρήση της `f_oneway()`.

Στο αποτέλεσμα διακρίνεται η τιμή με όνομα p-value. Αυτή είναι η τιμή που δείχνει εάν μπορούμε να απορρίψουμε την H_0 ή όχι. Η τιμή α πρέπει να οριστεί από την αρχή, συνήθως η τιμή που της δίνεται είναι $\alpha=0.05$. Στην περίπτωση που το p-value είναι μικρότερο από την τιμή α τότε μπορούμε να πούμε ότι η μηδενική υπόθεση απορρίπτεται. Στην περίπτωση που η τιμή p-value είναι μεγαλύτερη από την τιμή α τότε η H_0 δεν απορρίπτεται. Στο συγκεκριμένο παράδειγμα η τιμή είναι p-value= 0.0 που είναι μικρότερο από το $\alpha=0.05$. Το συμπέρασμα είναι ότι, σε επίπεδο σημαντικότητας 5%, δεν μπορούμε να ισχυριστούμε ότι όλοι οι μέσοι όροι είναι ίσοι.

4.4.2 Συσχέτιση Pearson

Η συσχέτιση Pearson είναι ένας από τους πιο συνηθισμένους τρόπους για να μετρηθεί μία γραμμική συσχέτιση. Η γραμμική συσχέτιση μετράει την ισχύ και την κατεύθυνση της σχέσης μεταξύ των μεταβλητών, υποδεικνύοντας πόσο στενά τείνουν να κινούνται μαζί οι τιμές τους: οι θετικά συσχετιζόμενες μεταβλητές αυξάνονται ή μειώνονται μαζί, ενώ οι αρνητικά συσχετιζόμενες μεταβλητές κινούνται προς αντίθετες κατευθύνσεις. Μια τιμή ανάμεσα στο 1 και το -1 μετράει το πόσο ισχυρή είναι η σχέση αλλά και την κατεύθυνση της συσχέτισης μεταξύ δύο μεταβλητών.



Εικόνα 57: Διαγράμματα αρνητικής, θετικής και χωρίς συσχέτιση.

Η συσχέτιση Pearson ανέδειξε την τιμή να είναι ίση με -0.042. Καθώς δεν υπάρχει μία ευθεία που να μπορεί να διαπεράσει από όλα τα δεδομένα και την τιμή της συσχέτισης να είναι -0.043, δηλαδή πολύ κοντά στο 0, τότε μπορούμε να πούμε πως το γεωγραφικό πλάτος και

καθημερινές μετρήσεις του δείκτη PM 2.5 δεν έχουν κάποια σχέση μεταξύ τους ή ότι δεν αποτυπώνεται αυτή η σχέση στην έρευνα.

```
import pandas as pd
from scipy.stats import pearsonr
import numpy as np
import matplotlib.pyplot as plt

pm25_concentration = df['Daily Mean PM2.5 Concentration']
site_latitude = df['SITE_LATITUDE']

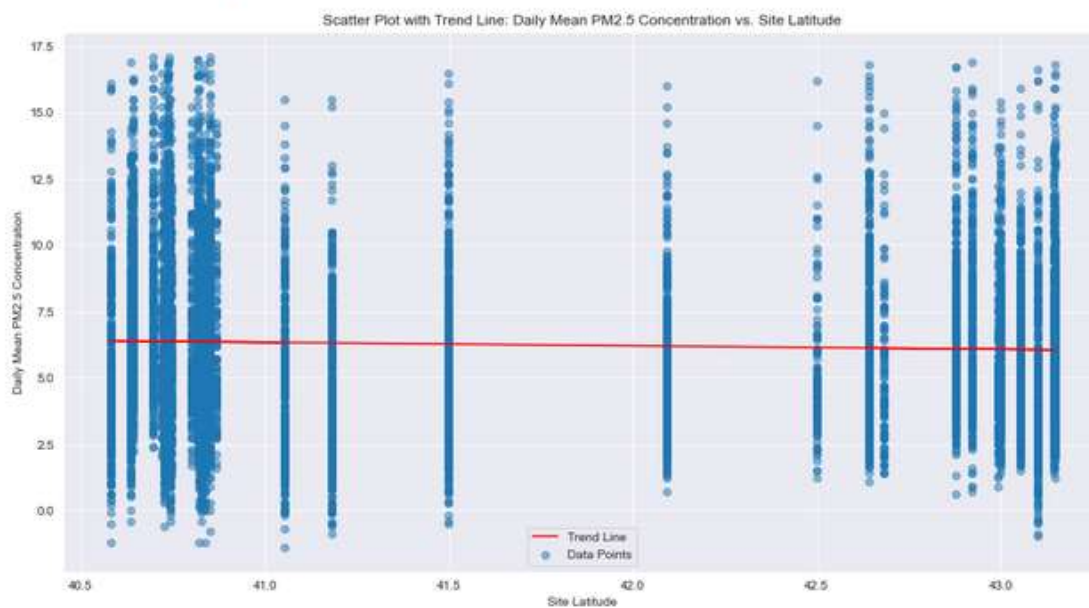
correlation_coefficient, p_value = pearsonr(pm25_concentration, site_latitude)

print(f'Pearson Correlation Coefficient: {correlation_coefficient}')
print(f'P-value: {p_value}')

slope, intercept = np.polyfit(site_latitude, pm25_concentration, 1)
trend_line = np.polyval([slope, intercept], site_latitude)

plt.scatter(site_latitude, pm25_concentration, alpha=0.5, label='Data Points')
plt.plot(site_latitude, trend_line, color='red', label='Trend Line')
plt.title('Scatter Plot with Trend Line: Daily Mean PM2.5 Concentration vs. Site Latitude')
plt.xlabel('Site Latitude')
plt.ylabel('Daily Mean PM2.5 Concentration')
plt.legend()
plt.show()
```

Pearson Correlation Coefficient: -0.04216793958442408
P-value: 1.4803609812924441e-05



Εικόνα 58: Κομμάτι κώδικα για την εκτέλεση της συσχέτισης Pearson με την εντολή pearson().

Μια τυχαία τοποθέτηση μιας τιμής στον άξονα X ή Y (βλ. Εικόνα 58) είναι πολύ πιθανό να είναι λανθασμένη καθώς δεν υπάρχει κάποια ευθεία να μπορεί να διαπεράσει όλα τα δεδομένα.

ΚΕΦΑΛΑΙΟ 5 Συμπεράσματα

Η Python συνέβαλε δραστικά στο να γίνει η διεξαγωγή της ανάλυσης της ποιότητας του αέρα για την Νέα Υόρκη το 2022 σημαντικά πιο εύκολη. Οι λεπτομερείς βιβλιοθήκες της Python, όπως pandas, NumPy και Matplotlib, αποδείχτηκαν σημαντικά εργαλεία καθώς περιείχαν όλες τις απαραίτητες συναρτήσεις για τον καθαρισμό, επεξεργασία, ανάλυση και οπτικοποίηση των δεδομένων. Αξιοποιώντας αυτές τις βιβλιοθήκες, εκτελέστηκαν απρόσκοπτα εργασίες όπως ο υπολογισμός περιγραφικών στατιστικών στοιχείων (μέσος όρος, διάμεσος, τρόπος, διακύμανση, τυπική απόκλιση), η εκτέλεση ανάλυσης συσχέτισης και η δημιουργία οπτικοποιήσεων. Επιπλέον, η ευελιξία της Python διευκόλυνε την ενσωμάτωση στατιστικών δοκιμών όπως η ANOVA και η συσχέτιση Pearson, επιτρέποντας τον έλεγχο υποθέσεων. Συνολικά, η Python αναδείχτηκε ως ένα πολύτιμο εργαλείο για την άντληση χρήσιμων πληροφοριών σχετικά με τις χωρικές και χρονικές τάσεις για την ποιότητα οξυγόνου για την Νέα Υόρκη το 2022.

Η ανάλυση κατανομής του Δείκτη Ποιότητας Αέρα στη Νέα Υόρκη για το 2022 έδειξε κατά βάση μια καλή κατάσταση της ποιότητας του αέρα, με το 94,5% των μετρήσεων να τοποθετούνται στην κατηγορία «καλές» και μόνο 5,5% να εμπίπτουν στην κατηγορία «μέτρια». Η κατανομή αυτή υποδηλώνει συνολικά υγιείς συνθήκες ποιότητας του αέρα σε όλη τη διάρκεια του έτους, γεγονός που αποδεικνύει τις αποτελεσματικές προσπάθειες διαχείρισης της ποιότητας του αέρα και ελέγχου της ρύπανσης στην περιοχή. Η επικράτηση των «καλών» μετρήσεων του δείκτη AQI δείχνει ότι η πλειονότητα των ημερών ήταν εντός ορίων στα πρότυπα ποιότητας του αέρα που έχουν καθοριστεί από τις αρμόδιες αρχές, παρέχοντας θετικές προοπτικές για τη δημόσια υγεία. Παρότι συνολικά οι μετρήσεις ήταν πολύ ικανοποιητικές, παρατηρήθηκαν κάποιες χωρικές ιδιαιτερότητες, με ορισμένες περιοχές όπως το Έρι και το Κουίνς να παρουσιάζουν σταθερά υψηλότερα ποσοστά «καλών» μετρήσεων AQI σε σύγκριση με το Τσαουτάκουα και το Ρόκλαντ. Στην ίδια λογική, περιοχές όπως το Queens και το Kings παρουσίασαν συχνότερα «μέτριες» ενδείξεις AQI, αναδεικνύοντας τοπικές προκλήσεις για την ποιότητα του αέρα.

Εκτός από κάποιες εξαιρέσεις στην χωρική ανάλυση, εξαιρέσεις υπήρχαν και στην χρονική ανάλυση. Οι μηνιαίες τάσεις υπογράμμισαν ότι τον Ιανουάριο και τον Φεβρουάριο παρατηρήθηκαν οι υψηλότερες τιμές PM 2.5 επίπεδα, που πιθανώς επηρεάζονται από τις αυξημένες απαιτήσεις θέρμανσης κατά τη διάρκεια του χειμώνα, ενώ ο Απρίλιος και ο Σεπτέμβριος παρουσίασαν τα χαμηλότερα επίπεδα. Αξίζει να σημειωθεί ότι, οι εποχιακές διακυμάνσεις έδειξαν χαμηλότερα επίπεδα PM 2.5 κατά τη διάρκεια της άνοιξης και του φθινοπώρου σε σύγκριση με το καλοκαίρι και το χειμώνα, με το χειμώνα να παρουσιάζει τα υψηλότερες συγκεντρώσεις.

Η ανάλυση συσχέτισης που έγινε μεταξύ των μέσων ημερήσιων μετρήσεων του δείκτη PM 2.5 και του γεωγραφικού πλάτους στη Νέα Υόρκη για το 2022 παρόλο δεν αποκάλυψε

συγκεκριμένη σχέση, αλλά υποδεικνύουν μια πιθανή επίδραση γεωγραφικού παράγοντα στα επίπεδα του PM 2.5. Ο συντελεστής Pearson βοηθάει στο να αναδείξει κατά πόσο υπάρχει η τάση των επιπέδων του PM 2.5 να μεταβάλλονται συστηματικά με την μεταβολή του γεωγραφικού πλάτους. Αν και η έρευνα δεν εντοπίζει συγκεκριμένους λόγους που οδηγούν σε αυτές τις αντιστοιχίσεις, υποδηλώνουν ότι το γεωγραφικό πλάτος μπορεί να επηρεάσει την ποιότητα του αέρα.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Βασίλειος Τ. Ταμπακάς. Εισαγωγή Στις Βάσεις Δεδομένων, Εκδόσεις Gotsis , 2017
- [2] <https://towardsdatascience.com/data-types-in-statistics-347e152e8bee>
- [3] <https://en.wikipedia.org/wiki/Data>
- [4] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar. Εισαγωγή στην Εξόρυξη Δεδομένων, Εκδόσεις Τζιόλας, ISBN:9789604188130
- [5] Αικατερίνη Μπακούρα. Εισαγωγή στην Στατιστική, Εκδόσεις Δίσιγμα, 2013
- [6] <https://www.geeksforgeeks.org/data-structures/>
- [7] <https://www.python.org/doc/essays/omg-darpa-mcc-position/>
- [8] https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html#part-1-beginning-python
- [9] <https://www.simplilearn.com/tutorials/python-tutorial/comments-in-python>
- [10] <https://www.digitalocean.com/community/tutorials/python-data-types>
- [11] https://www.tutorialspoint.com/python/python_data_types.htm#
- [12] <https://www.toppr.com/guides/python-guide/references/methods-and-functions/methods/built-in/type/python-type/>
- [13] https://el.wikipedia.org/wiki/%CE%91%CE%BA%CE%AD%CF%81%CE%B1%CE%B9%CE%BF%CF%82_%CE%B1%CF%81%CE%B9%CE%B8%CE%BC%CF%8C%CF%82
- [14] <https://emeritus.org/blog/coding-learn-what-is-a-float-in-python/>
- [15] <https://www.geeksforgeeks.org/boolean-data-type-in-python/>
- [16] <https://intellipaat.com/blog/tutorial/python-tutorial/type-conversion-in-python/>
- [17] https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf
- [18] <http://www.algorithmos.gr/domi-epanalipsis.html>
- [19] https://www.w3schools.com/python/python_variables_global.asp
- [20] <https://www.techtarget.com/whatis/definition/scripting-language>
- [21] <https://www.fermyon.com/blog/scripts-vs-compiled-wasm>
- [22] <https://www.geeksforgeeks.org/libraries-in-python/>

- [23] <https://pandas.pydata.org/about/>
- [24] <https://www.simplilearn.com/top-python-libraries-for-data-science-article>
- [25] <https://numpy.org/doc/stable/reference/routines.statistics.html>
- [26] <https://matplotlib.org/stable/>
- [27] <https://seaborn.pydata.org/>
- [28] <https://www.dunderdata.com/blog/top-5-reasons-to-use-seaborn-for-data-visualizations>
- [29] <https://docs.python.org/3/library/statistics.html>
- [30] <https://docs.python.org/3/library/warnings.html>
- [31] <https://scied.ucar.edu/learning-zone/air-quality/what-is-air-quality>
- [32] <https://www.airnow.gov/aqi/aqi-basics/>
- [33] <https://scijinks.gov/air-quality/>
- [34] <https://ww2.arb.ca.gov/resources/inhalable-particulate-matter-and-health>
- [35] <https://www.techtarget.com/searchdatamanagement/definition/data-scrubbing>
- [36] <https://www.sagacitysolutions.co.uk/about/news-and-blog/data-deduplication/>
- [37] <https://www.linkedin.com/advice/0/what-potential-consequences-leaving-duplicates-your>
- [38] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop_duplicates.html
- [39] https://www.investopedia.com/terms/d/descriptive_statistics.asp
- [40] <https://www.britannica.com/science/statistics/Hypothesis-testing>