

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ · ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

# ΕΞΟΥΞΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΑΛΓΟΡΙΘΜΟΙ ΜΑΘΗΣΗΣ

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ · 2023–2024

**ΠΕΡΙΕΧΟΜΕΝΑ**

<b>1</b>	<b>ΕΡΩΤΗΜΑ 1</b>	<b>2</b>
1.1	ΑΝΑΛΥΣΗ ΣΥΝΟΛΟΥ ΔΕΔΟΜΕΝΩΝ . . . . .	2
1.2	ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ . . . . .	3
<b>2</b>	<b>ΕΡΩΤΗΜΑ 2</b>	<b>6</b>
2.1	ΟΡΙΣΜΟΣ ΤΑΞΙΝΟΜΗΤΩΝ . . . . .	6
2.2	ΑΠΟΤΕΛΣΜΑΤΑ . . . . .	6
2.2.1	NEURAL NETWORKS . . . . .	7
2.3	ΣΥΜΠΕΡΑΣΜΑΤΑ . . . . .	7

## 1 ΕΡΩΤΗΜΑ 1

Το σύνολο δεδομένων περιλαμβάνει 22 .csv αρχεία που αντιστοιχούν σε 22 συμμετέχοντες. Σύμφωνα με την περιγραφή του dataset, περιλαμβάνεται η στήλη `timestamp`, με την ημερομηνία και ώρα, οι στήλες `backx,y,z` και `thighx,y,z` με τις τιμές του κάθε αισθητήρα για κάθε διάσταση, και η στήλη `label`, η οποία προσδιορίζει τη δραστηριότητα του συμμετέχοντα τη δεδομένη στιγμή.

Η στήλη `label` παίρνει τις εξής τιμές:

1 - Walking	8: lying
2 - Running	13 - Cycling (sit)
3 - Shuffling	14 - Cycling (stand)
4 - Stairs (ascending)	130 - Cycling (sit, inactive)
5 - Stairs (descending)	140 - Cycling (stand, inactive)
6 - Standing	

Για την εισαγωγή και την προεπεξεργασία των αρχείων, θα χρησιμοποιήσουμε τη βιβλιοθήκη `pandas` ενώ για την οπτικοποίηση τις βιβλιοθήκες `matplotlib` και `seaborn` της Python.

### 1.1 ΑΝΑΛΥΣΗ ΣΥΝΟΛΟΥ ΔΕΔΟΜΕΝΩΝ

Εισάγουμε τα .csv αρχεία μέσω της `os` βιβλιοθήκης και της `read_csv()` συνάρτησης. Καταρχάς, χρησιμοποιώντας τη `head()` μπορούμε να δούμε τις πρώτες εγγραφές του dataset μας. Για παράδειγμα για το πρώτο αρχείο του συνόλου δεδομένων `S006.csv`:

	timestamp	back_x	back_y	back_z	thigh_x	thigh_y	thigh_z	label
0	2019-01-12 00:00:00.000	-0.760242	0.299570	0.468570	-5.092732	-0.298644	0.709439	6
1	2019-01-12 00:00:00.010	-0.530138	0.281880	0.319987	0.900547	0.286944	0.340309	6
2	2019-01-12 00:00:00.020	-1.170922	0.186353	-0.167010	-0.035442	-0.078423	-0.515212	6
3	2019-01-12 00:00:00.030	-0.648772	0.016579	-0.054284	-1.554248	-0.950978	-0.221140	6
4	2019-01-12 00:00:00.040	-0.355071	-0.051831	-0.113419	-0.547471	0.140903	-0.653782	6

Μέσω της `info()` εξάγουμε το συμπέρασμα πώς για κάθε χρονική στιγμή δίνονται οι τιμές των αισθητήρων, αποθηκευμένες ως `float24`, στις τρεις διαστάσεις (x, y, z) για τις περιοχές της πλάτης και του μηρού, καθώς και ένα `int64` για το `label`. Η ίδια μορφολογία παρατηρείται σε όλα τα .csv του συνόλου δεδομένων, με κάποιες διαφοροποιήσεις που θα αναλυθούν στη συνέχεια.

Παρόλο που στην περιγραφή αναφέρεται πως δεν υπάρχουν `missing values`, για να ελέγξουμε την ακεραιότητα του dataset, μέσω της συνάρτησης `concat()` ενώνουμε όλα τα 22 αρχεία σε ένα ενιαίο dataframe. Τρέχοντας την `isnull().sum()`, έχουμε:

	sum
timestamp	0
back <sub>x</sub>	0
back <sub>y</sub>	0
back <sub>z</sub>	0
thigh <sub>x</sub>	0
thigh <sub>y</sub>	0
thigh <sub>z</sub>	0
label	0
index	5740689
Unnamed: 0	6323682

Παρατηρούμε πως στις στήλες `backx,y,z` και `thighx,y,z`, οι οποίες είναι και αυτές που μας ενδιαφέρουν, όντως δεν παρατηρούνται `missing values`. Όμως, έχουν εμφανιστεί `NaN` τιμές στις στήλες "index" και "Unnamed: 0", οι οποίες στήλες μάλιστα θα εμφανίζονται επιπλέον σε κάποια αρχεία.

Ελέγχοντας όλα τα αρχεία, η στήλη "index" εμφανίζεται στα αρχεία `S015.csv` και `S021.csv` και η στήλη "Unnamed: 0" στο αρχείο `S023.csv`. Έπειτα από έλεγχο, φαίνεται πως πρόκειται για δείκτες αύξουσας αρίθμησης που δεν προσφέρουν κάποια επιπλέον πληροφορία. Επομένως, μπορούμε να τις αφαιρέσουμε χρησιμοποιώντας τη συνάρτηση `drop('όνομα', axis=1)`. Τα επεξεργασμένα αρχεία αποθηκεύονται με το επίθεμα `fix`.

Χρησιμοποιώντας τη συνάρτηση `describe()` μπορούμε να υπολογίσουμε βασικές στατιστικές μετρικές για τα δεδομένα μας. Η συνάρτηση επιστρέφει ένα `dataframe` με τις ακόλουθες στήλες:

- **count**: ο συνολικός αριθμός των μη-μηδενικών τιμών για κάθε στήλη.
- **mean**: ο μέσος όρος των τιμών για κάθε στήλη.
- **min**: η ελάχιστη τιμή για κάθε στήλη.
- **25%**: η τιμή του 25ου εκατοστημορίου για κάθε στήλη.
- **50%**: η τιμή του 50ου εκατοστημορίου για κάθε στήλη.
- **75%**: η τιμή του 75ου εκατοστημορίου για κάθε στήλη.
- **max**: η μέγιστη τιμή για κάθε στήλη.

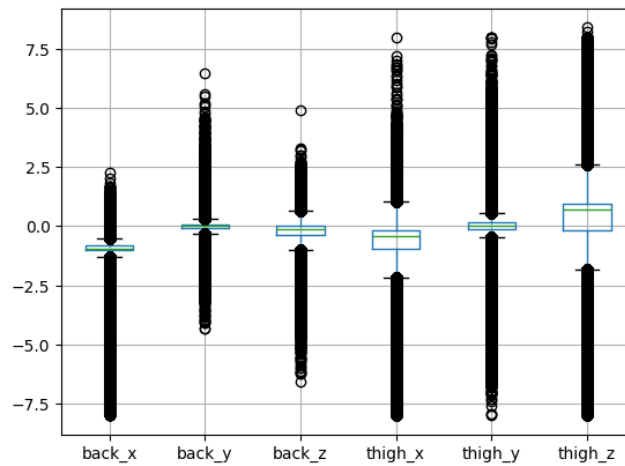
Ενώνοντας συγκεντρωτικά τις μετρήσεις όλων των συμμετεχόντων στο `df_combined` μέσω της `concat()`, αυτά είναι τα **βασικά συγκεντρωτικά στατιστικά μεγέθη** όπως προκύπτουν από το `describe()` για όλες τις μετρήσεις από τους συμμετέχοντες, έχοντας αφαιρέσει την ετικέτα `label` μιας και αποτελείται από κατηγορικές τιμές:

	<code>back<sub>x</sub></code>	<code>back<sub>y</sub></code>	<code>back<sub>z</sub></code>	<code>thigh<sub>x</sub></code>	<code>thigh<sub>y</sub></code>	<code>thigh<sub>z</sub></code>
count	6461328	6461328	6461328	6461328	6461328	6461328
mean	-0.884957	-0.013261	-0.169378	-0.594888	0.020877	0.374916
std	0.377592	0.231171	0.364738	0.626347	0.388451	0.736098
min	-8.000000	-4.307617	-6.574463	-8.000000	-7.997314	-8.000000
25%	-1.002393	-0.083129	-0.372070	-0.974211	-0.100087	-0.155714
50%	-0.974900	0.002594	-0.137451	-0.421731	0.032629	0.700439
75%	-0.812303	0.072510	0.046473	-0.167876	0.154951	0.948675
max	2.291708	6.491943	4.909483	7.999756	7.999756	8.406235

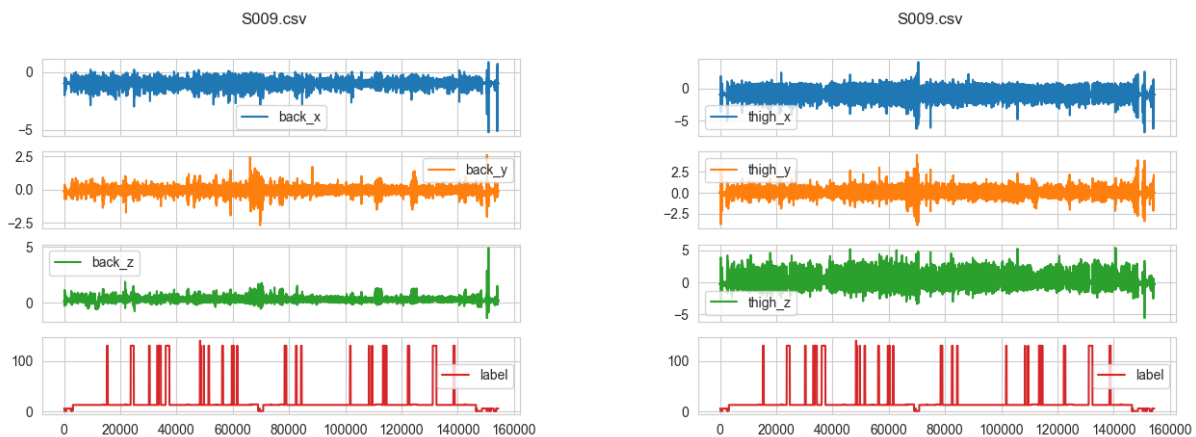
Ως αρχικές παρατηρήσεις, βλέπουμε πως οι τιμές βρίσκονται στο διάστημα  $[-8, 8]$ , ενώ η τυπική τους απόκλιση είναι μικρή, το οποίο δείχνει ότι οι μετρήσεις είναι αρκετά συμπυκνωμένες γύρω από τον μέσο όρο που είναι κοντά στο μηδέν. Προφανώς ελέγχοντας κάθε συμμετέχοντα ξεχωριστά, μπορεί να διεξαχθούν αντίστοιχα συμπεράσματα.

## 1.2 ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ

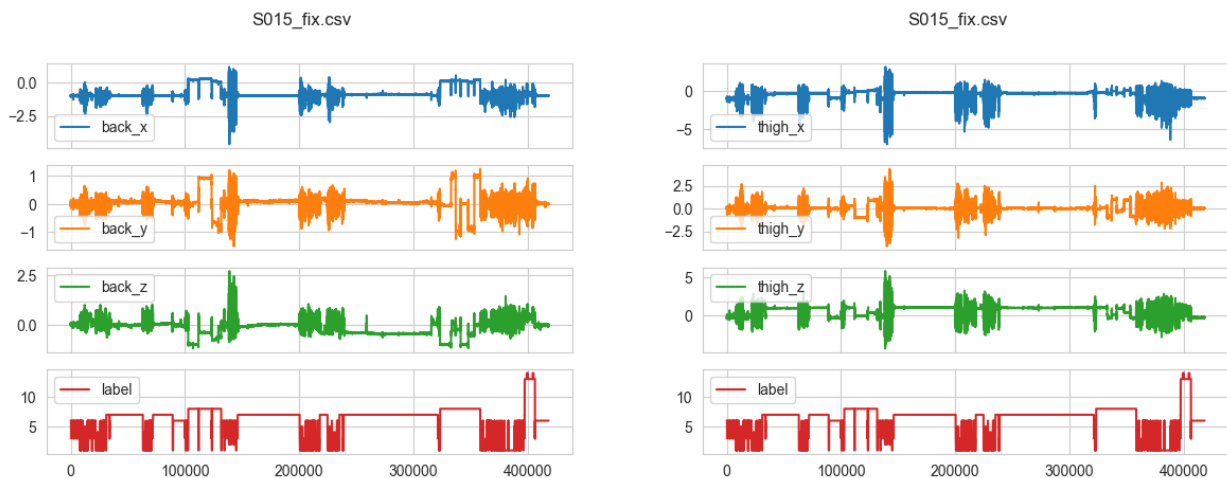
Μέσω της `plotbox()` της `Matplotlib`, μπορούμε να δημιουργήσουμε το διάγραμμα των τιμών της `df_combined` για μια πρώτη οπτικοποίηση των δεδομένων:



Πέρα από τις προηγούμενες παρατηρήσεις που επιβεβαιώνονται, επιπλέον παρατηρούμε μια συμμετρικότητα κοντά στο μηδέν για κάθε διάσταση. Επίσης, χρησιμοποιώντας την `displot()`, μπορούμε να οπτικοποιήσουμε το πώς κατανέμονται οι τιμές. Ενδεικτικά για τον S009:



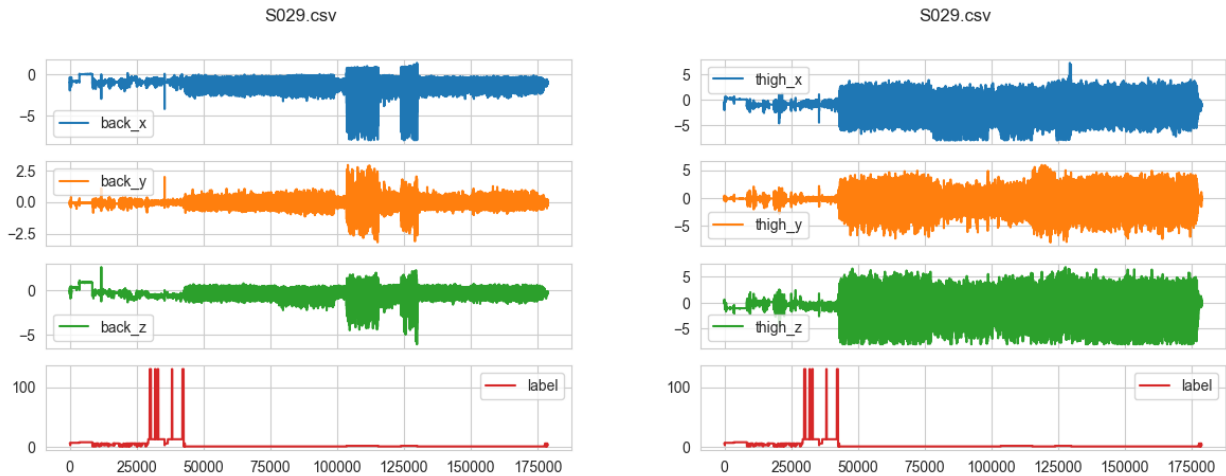
Χρησιμοποιώντας την `plot()`, μπορούμε να δημιουργήσουμε subplots για τις στήλες  $back_{x,y,z}$  και  $thigh_{x,y,z}$ . Αυτά, για παράδειγμα, είναι τα subplots για τον συμμετέχοντα S015:



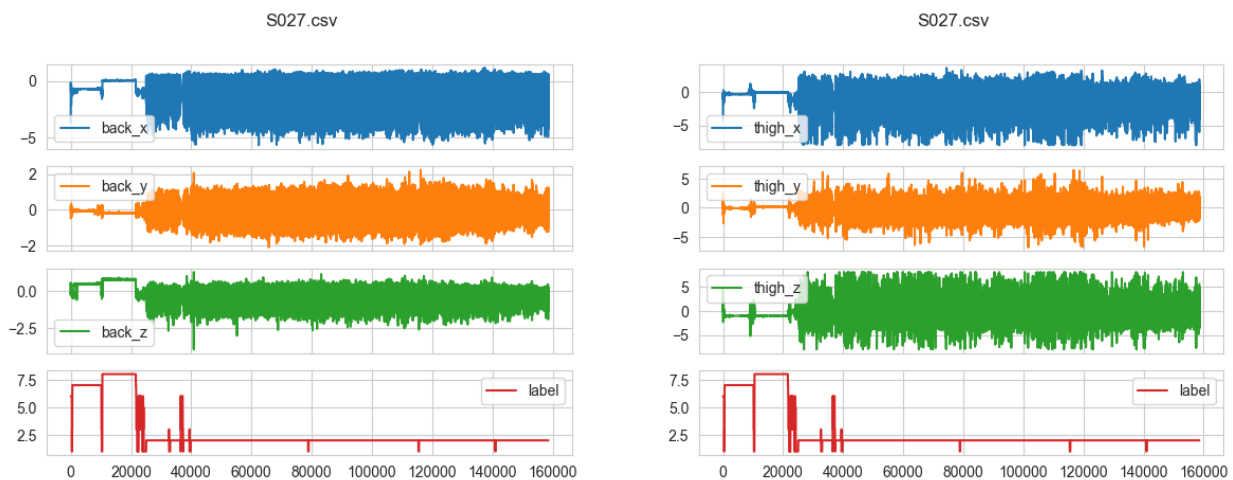
Φαίνεται ότι ο συμμετέχοντας κατά τη διάρκεια της μέτρησης μεταβάλλει τη φυσική του δραστηριότητα (μάλιστα με παρόμοιο τρόπο σε πλάτη και μηρό), καθώς υπάρχουν στιγμές που δεν υπάρχουν έντονες διακυμάνσεις των τιμών των μετρήσεων των αισθητήρων και άλλες όπου είναι πιο ενεργός, με την τιμή του

label να αλληιάζει και αυτή. Στις στιγμές που ο συμμετέχοντας δεν κινείται, το label φαίνεται να παίρνει την τιμή 8 – Standing που επιβεβαιώνει τη στασιμότητά του.

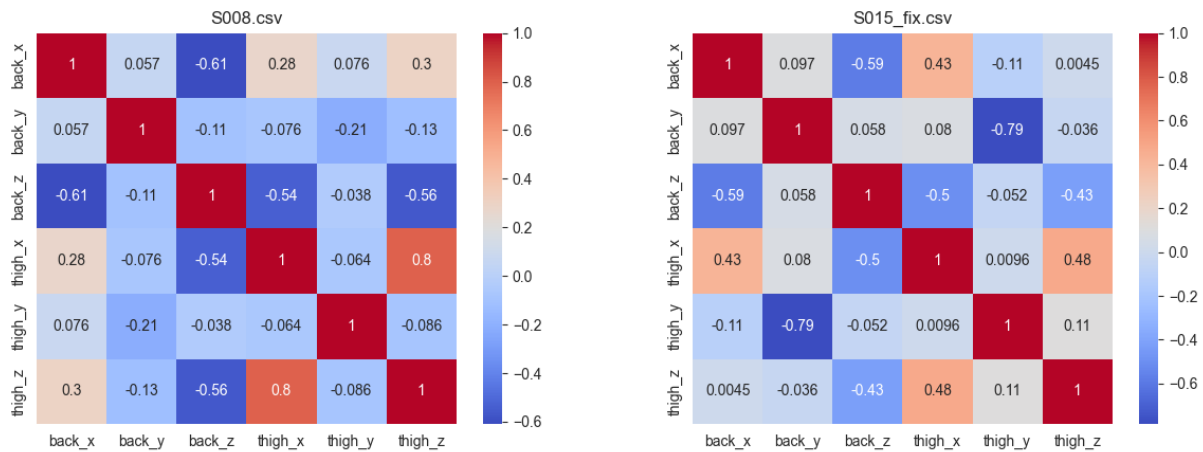
Από την άλλη, στον συμμετέχοντα S029 παρατηρούμε πως η κίνηση της πλάτης δεν ταυτίζεται με την (έντονη) κίνηση των μηρών, κάτι που μας οδηγεί στο συμπέρασμα πως ο συμμετέχοντας κάνει ποδήλατο. Το γεγονός αυτό επιβεβαιώνεται και από τα spikes του label στις τιμές 100.



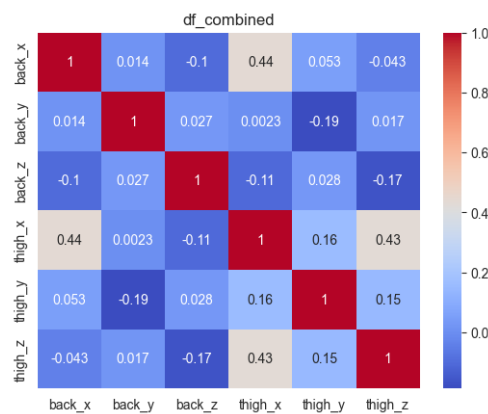
Τέλος, για τον συμμετέχοντα S027, φαίνεται να έχει μια πολύ έντονη φυσική δραστηριότητα με τη label να παραμένει σταθερή με τιμή κοντά στο 2.5, κάτι από το οποίο μπορούμε να συμπεράνουμε πως ο συμμετέχοντας τρέχει:



Τέλος, για τον εντοπισμό συσχετίσεων, μπορούμε να δημιουργήσουμε ένα heatmap() μέσω της seaborn. Για παράδειγμα, για τον συμμετέχοντα S008 φαίνεται πως υπάρχει μια κάποια συσχέτιση ανάμεσα στις στήλες back\_x + back\_z, thigh\_x + thigh\_z και back\_z + thigh\_x ενώ στον S015 ανάμεσα στις στήλες back\_x + back\_z και back\_y + thigh\_y.



Δημιουργώντας heatmap και για το συγκεντρωτικό dataframe `df_combined`, παρατηρούμε μια αμυδρή συσχέτιση ανάμεσα στα `thigh_x + back_x` και `thigh_x + thigh_z`.



Στο παράρτημα παρατίθενται τα plots και τα heatmaps για όλους τους συμμετέχοντες.

## 2 ΕΡΩΤΗΜΑ 2

### 2.1 ΟΡΙΣΜΟΣ ΤΑΞΙΝΟΜΗΤΩΝ

Στην συνάρτηση `get_classifier(option)` ορίζονται και μπορούν να επιλεγθούν οι ταξινομητές που θα χρησιμοποιηθούν στη συνέχεια.

Σε κάθε περίπτωση ταξινομητή, σε καθένα από τα 22 dataframes του dataset αφαιρείται η στήλη `timestamp`, και το dataframe διαχωρίζεται από τη στήλη `label` στα `X` και `Y`. Όταν γίνει ο διαχωρισμός, γίνονται `split` στα dataframes `X_train`, `X_test`, `Y_train`, `Y_test` με `test_size = 0.3`.

Αφού γίνει η επιλογή του classifier, αυτός γίνεται `train` μέσω της `fit(X_train, Y_train)` και αποθηκεύονται τα `predictions` του μέσω της `predict(X_test)`.

### 2.2 ΑΠΟΤΕΛΑΣΜΑΤΑ

Τρέχουμε κάθε classifier για όλους τους συμμετέχοντες:

## 2.2.1 NEURAL NETWORKS

file	training accuracy	testing accuracy
S006.csv	0.9141337173536156	0.9122442155399509
S008.csv	0.9329507794280103	0.9315337677112421
S009.csv	0.8956013466020495	0.8916271040138111
S010.csv	0.8519300925436921	0.8499360159249254
S012.csv	0.9738502515979364	0.9730834604488996
S013.csv	0.9007985198546176	0.8996423539612008
S014.csv	0.8991970063148047	0.8973778274987039
S015_fix.csv	0.9135976563300259	0.9136856865150815
S016.csv	0.9141337173536156	0.9447883255491156
S017.csv	0.9329507794280103	0.9109135048143804
S018.csv	0.8956013466020495	0.8708032518979748
S019.csv	0.8519300925436921	0.9578000537008861
S019.csv	0.9604528171614061	0.9572518571556431
S020.csv	0.9542851869085204	0.9545083401376414
S021_fix.csv	0.9347834306997145	0.9334105321202095
S022.csv	0.9020188641720371	0.9002873194379992
S023_fix.csv	0.9318747924277648	0.9256308422531119
S024.csv	0.9218583766848449	0.9191376243623073
S025.csv	0.8687010665187103	0.8650728577798875
S026.csv	0.8737227345922998	0.8699446645716628
S027.csv	0.9876495387719804	0.9867580292584497
S028.csv	0.9772106137134159	0.9761270533155749
S029.csv	0.9475463825229214	0.9443625850974541

## 2.3 ΣΥΜΠΕΡΑΣΜΑΤΑ