

MASTER - Notebook 1

Matteo Grazioso 884055

```
In [ ]: # Import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import json
import warnings
warnings.filterwarnings('ignore')
```

```
In [ ]: # Disply all columns and all rows
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

```
In [ ]: # The fileS contain the data of the validation of tickets in the city of public transport of Venice.

# Import the data into a dataframe of a txt file
path = 'data/raw/validazioni.txt'          # Period: 2022-05-13 to 2022-07-15
# path = 'data/raw/esportazioneCompleta.txt'      # Period: 2023-01-23 to 2023-03-14

df = pd.read_csv(path, header=0, sep='\t')
# Save the name of the file in a variable for future use extracting the name of the file from the path
file_name = path.split('/')[1].split('.')[0]
```

```
In [ ]: # Check the first 5 rows of the data
df.head()
```

```
Out [ ]:
```

	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	DESCRIZIONE_TITOLO
0	13/05/2022 00:00	65676291870913797	5089	FERROVIA "D"	11149	7gg-Tpl 43,60-ComVe16,40
1	13/05/2022 00:00	36141384536591364	5032	FERROVIA "B"	11107	48h-Tpl 24,90-ComVe5,10
2	13/05/2022 00:00	36144856606063108	5031	P.le Roma "G"	11108	72h-Tpl 33,40-ComVe6,60
3	13/05/2022 00:00	36144856608393988	506	VENEZIA	12106	Bigl Aer-Venezia TSC
4	13/05/2022 00:00	36144856608393732	506	VENEZIA	12106	Bigl Aer-Venezia TSC

```
In [ ]: # Check the last 5 rows of the data
df.tail()
```

```
Out [ ]:
```

	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	DESCRIZIONE_TITOLO
5056636	15/07/2022 06:46	36141603027798788	3435	Oriago Stazi	14123	Extra tratta 3
5056637	15/07/2022 06:46	36141603027798788	3435	Oriago Stazi	14123	Extra tratta 3
5056638	15/07/2022 06:46	36141603027798788	3436	Oriago Centr	14123	Extra tratta 3
5056639	15/07/2022 07:01	65676630087266309	1306	Spinea Orgna	14123	Extra tratta 3
5056640	15/07/2022 07:07	36428626034318852	1312	Spinea Giorg	14123	Extra tratta 3

```
In [ ]: # Create a subset of the data with the first 10% of the rows and the last 10% of the rows
# df = df.iloc[:int(len(df)*0.1),:]
# df = df.append(df.iloc[-int(len(df)*0.1):,:])
```

Explorative Data Analysis

```
In [ ]: # Dates and hour of the validation of the ticket are in the same column 'DATA_VALIDAZIONE'
# Split the column 'DATA_VALIDAZIONE' into two columns 'DATA' and 'ORA' and convert them to datetime format
df.insert(0, 'DATA', pd.to_datetime(df['DATA_VALIDAZIONE'].str.split(' ').str[0], format='%d/%m/%Y'))
df.insert(1, 'ORA', pd.to_datetime(df['DATA_VALIDAZIONE'].str.split(' ').str[1], format='%H:%M').dt.time)

# Drop the column 'DATA_VALIDAZIONE'
# df.drop('DATA_VALIDAZIONE', axis=1, inplace=True)
```

```
# Display the first 5 rows of the dataframe
df.head()
```

```
Out [ ]:
```

	DATA	ORA	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	DESCRIZIONE_TITOLO
0	2022-05-13	00:00:00	13/05/2022 00:00	65676291870913797	5089	FERROVIA "D"	11149	7gg-Tpl 43,60-ComVe16,40
1	2022-05-13	00:00:00	13/05/2022 00:00	36141384536591364	5032	FERROVIA "B"	11107	48h-Tpl 24,90-ComVe5,10
2	2022-05-13	00:00:00	13/05/2022 00:00	36144856606063108	5031	P.le Roma "G"	11108	72h-Tpl 33,40-ComVe6,60
3	2022-05-13	00:00:00	13/05/2022 00:00	36144856608393988	506	VENEZIA	12106	Bigl Aer-Venezia TSC
4	2022-05-13	00:00:00	13/05/2022 00:00	36144856608393732	506	VENEZIA	12106	Bigl Aer-Venezia TSC

```
In [ ]: # Set the format of the timestamp
df['DATA_VALIDAZIONE'] = pd.to_datetime(df['DATA_VALIDAZIONE'], format='%d/%m/%Y %H:%M')
```

```
In [ ]: # Print the date of the first and last validation using both data and hour
print('First validation: ', df['DATA'].min(), df['ORA'].min())
print('Last validation: ', df['DATA'].max(), df['ORA'].max())

# Print the number of Serial numbers
print('Number of Serial numbers: ', df['SERIALE'].nunique())

# Print the number of validation (rows)
print('Number of validation: ', df.shape[0])

# Print the number of tickets
print('Number of tickets: ', df['DESCRIZIONE_TITOLO'].nunique())
# Print the number of titolo
print('Number of titolo: ', df['TITOLO'].nunique())
# TODO: why the number of unique TITOLO is different from the number of DESCRIZIONE_TITOLO?

# Print the number of FERMATA
print('Number of FERMATA: ', df['FERMATA'].nunique())
# Print the number of DESCRIZIONE
print('Number of DESCRIZIONE: ', df['DESCRIZIONE'].nunique())
# TODO: why the number of unique DESCRIZIONE is different from the number of FERMATA?
```

First validation: 2022-05-13 00:00:00 00:00:00
Last validation: 2022-07-15 00:00:00 23:59:00
Number of Serial numbers: 2038775
Number of validation: 5056641
Number of tickets: 168
Number of titolo: 170
Number of FERMATA: 1672
Number of DESCRIZIONE: 935

```
In [ ]: # Which is the most used ticket?  
df['DESCRIZIONE_TITOLO'].value_counts().head(10)
```

```
Out [ ]: DAILY PASS VENEZIA - AVM          972478  
75'-Tpl 6,64-ComVe0,86          743608  
48h-Tpl 24,90-ComVe5,10         600320  
72h-Tpl 33,40-ComVe6,60         492911  
Bigl.Aut.75'Mestre/Lido-tsc     422668  
7gg-Tpl 43,60-ComVe16,40        342870  
75'-Tpl 13,28-ComVe1,72         232644  
Biglietto 72 ore Roll. Venice   170675  
72ore online no aerobus         108357  
7 days online no aerobus        101869  
Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: # Which is the most frequent validation in date and hour?  
# Date and hour are in two different columns; DATA_VALIDAZIONE does not exist anymore  
df.groupby(['DATA', 'ORA'])['SERIALE'].count().sort_values(ascending=False).head(10)  
# TODO: #4 Re-aswer the question of the most frequent validation after cleaning operations
```

```
Out[ ]: DATA      ORA
2022-05-17 03:38:00    383
2022-05-18 03:38:00    362
2022-06-04 15:27:00    258
2022-05-27 17:25:00    258
2022-06-03 16:28:00    254
2022-05-27 10:26:00    250
           10:54:00    249
2022-06-04 16:26:00    247
2022-05-27 11:14:00    246
           10:10:00    242
Name: SERIALE, dtype: int64
```

```
In [ ]: # Which is the most frequent FERMATA?
df['DESCRIZIONE'].value_counts().head(10)
# TODO: #4 Re-aswer the question of the most frequent FERMATA after cleaning operations
```

```
Out[ ]: San Marco-Sa    583170
Rialto                346023
LIDO S.M.E.          274706
BURANO "C"           248780
VENEZIA              238940
P.le Roma "G"        220171
FERROVIA "B"         194802
San Marco Va         145476
Punta Sabbio         114300
FERROVIA "D"         101044
Name: DESCRIZIONE, dtype: int64
```

Categories

```
In [ ]: # Add a new column with the code profile of the ticket
df.insert(7, "TICKET_CODE", 'TBD')
```

This column will be filled with the code of the ticket profile according to the ticket type and the ticket validity as follows:

1. One-day ticket

2. Two-day ticket
3. Three-day ticket
4. Weekly ticket (Seven-day ticket)
5. Monthly ticket
 - 5-STUD. Monthly ticket for students
 - 5-RET. Monthly ticket for retirees
 - 5-WKRS. Monthly ticket for workers
6. Annual ticket
 - 6-STUD. Annual ticket for students
 - 6-RET. Annual ticket for retirees
 - 6-WKRS. Annual ticket for workers
7. 75 minutes ticket
8. Other ticket (if it is necessary to add other types of tickets)

```
In [ ]: df.head()
```

Out []:	DATA	ORA	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	TICKET_CODE	DESCRIZIONE_TITOLO
0	2022-05-13	00:00:00	2022-05-13	65676291870913797	5089	FERROVIA "D"	11149	TBD	7gg-Tpl 43,60-ComVe16,40
1	2022-05-13	00:00:00	2022-05-13	36141384536591364	5032	FERROVIA "B"	11107	TBD	48h-Tpl 24,90-ComVe5,10
2	2022-05-13	00:00:00	2022-05-13	36144856606063108	5031	P.le Roma "G"	11108	TBD	72h-Tpl 33,40-ComVe6,60
3	2022-05-13	00:00:00	2022-05-13	36144856608393988	506	VENEZIA	12106	TBD	Bigl Aer-Venezia TSC
4	2022-05-13	00:00:00	2022-05-13	36144856608393732	506	VENEZIA	12106	TBD	Bigl Aer-Venezia TSC

```
In [ ]: # Create a dictionary with the ticket code and the ticket profile
dict_tickets = {'1': 'One-day ticket', '2': 'Two-day ticket', '3': 'Three-day ticket',
               '4': 'Seven-day ticket',
               '5': 'Monthly ticket', '5-STUD': 'Monthly ticket for students',
               '5-RET': 'Monthly ticket for retired', '5-WKRS': 'Monthly ticket for workers',
               '6': 'Annual ticket', '6-STUD': 'Annual ticket for students', '6-RET': 'Annual ticket for retired',
               '6-WKRS': 'Annual ticket for workers',
               '7': '75 minutes ticket', '8': 'Other ticket'}

# Export the dictionary to a json file
with open('data/dictionaries/dict_ticket_codes.json', 'w') as fp:
    json.dump(dict_tickets, fp)
```

```
In [ ]: # How many unique values are there in the column 'DESCRIZIONE_TITOLO'?
df['DESCRIZIONE_TITOLO'].nunique()
```

Out []: 168

```
In [ ]: # Which are the unique values of the column 'DESCRIZIONE_TITOLO'?
df['DESCRIZIONE_TITOLO'].unique()
```

```
Out[ ]: array(['7gg-Tpl 43,60-ComVe16,40', '48h-Tpl 24,90-ComVe5,10',  
              '72h-Tpl 33,40-ComVe6,60', 'Bigl Aer-Venezia TSC',  
              'DAILY PASS VENEZIA - AVM', '72H R.Venice+aerop.AR online',  
              '75'-Tpl 6,64-ComVe0,86", 'Biglietto 72 ore Roll. Venice',  
              'Bigl.Aut.75'Mestre/Lido-tsc", 'Aer+boat-Tpl14,50-C.Ve1,50',  
              '72hAerCS-Tpl40,40-CVe6,60', 'Aeroporto-Venezia AR',  
              '72 ore R.Venice online', '7 days online no aerobus',  
              '72hAerAR-Tpl46,40-CVe6,60', '48hAerCS-Tpl31,90-CVe5,10',  
              'Bicicletta "Palmare"', 'L.17-auto "C"da 4,01 a 4,50 mt',  
              '72ore online no aerobus', 'L.17-auto "AeB" fino a 4 metri',  
              '48ore online no aerobus', '75'-Tpl 13,28-ComVe1,72",  
              'Extra tratta 4 ', '48ore online aerobus AR',  
              'Daily Pass Venezia Online', 'Ferry17-carri+35q.rim.',  
              'Ferry17-autocarri+35q.', 'Ferry11-autocarri+35q.',  
              'Extra tratta 5 ', 'L.17-auto "D" oltre metri 4,50',  
              'Extra tratta 2 ', 'Ciclomotore fino 50cc',  
              'Bicicletta "biglietteria"', 'Extra tratta 3 ',  
              'Bigl.Mestre/Lido 75' a bordo', 'Prenotaz OCCASIONALE si barra',  
              'AtvoCanova+Actv 72Hroll.online', '72 ore R.Venice+aeroporto CS',  
              'Extra tratta 1', '7 days online aerobus AR',  
              'Atvo Canova+Navig AR online', 'Atvo Canova+Actv 72H online',  
              'Tragh-Tpl 4,41-C.Ve0,59', 'ARRIVA VENETO AEROPORTO',  
              'ARRIVA VENETO tratta 8-9-10', '24hAerCS-Tpl22,90-CVe5,10',  
              'Extra tratta 7 ', 'biglietto merci C.Semplice',  
              'Bordo 75min CartaVenezia', 'Bigl.urbano CHIOGGIA',  
              '72 ore R.Venice+aeroporto AR', '48hAerAR-Tpl37,90-CVe5,10',  
              'Extra tratta 2 TVM', 'Extra tratta 1 TVM',  
              'Carnet CHIOGGIA 10c. TICKET', '7ggAerAR-Tpl56,60-CVe16,40',  
              "PeopleMover+Bus+Tram 75'", 'Ferry17-Trasporti pericolosi',  
              'NA-24H metropolitano ORD+1', '72ore online aerobus CS',  
              'ARRIVA VENETO tratta 3', 'abbonamento 30 gg.PeopleMover',  
              'ord. navigazione 75' online", 'L.11-auto "D" oltre metri 4,50',  
              'Extra tratta 6 ', '24ore online aerobus AR',  
              'Cav -Trep + Actv 24H', '7ggAerCS-Tpl50,60-CVe16,40',  
              'Ciclomotore oltre 50cc', 'Bagaglio CartaVenezia',  
              'Cav - Trep + Actv 72H', 'Cav-Trep - S.Marco AR',  
              'Mens. cose animali RETE UNICA', 'Tragh-Tpl 8,82-C.Ve1,18',  
              '72H RVenice+aerop.CS online', 'Aeroporto-Venezia CS ONLINE',  
              'NA-Traghetto ordinario', 'ARRIVA VENETO tratta 4',  
              'Extra tratta 4 TVM', 'ARRIVA VENETO tratta 1',
```


'72ore online aerobus AR', '48ore online aerobus CS',
'NA-24h-Tpl 14,90-Com.Ve5,10', 'Jesolo - S.Marco AR',
'Jesolo + Actv 24H', "75'-Tpl 12,60-CVe2,40 online",
'Extra tratta 5 TVM', 'Bicicletta "concessionari"',
'Gruppi e Scuole', '7 days online aerobus CS',
'Extra tratta 3 TVM', 'T.Fusina Ve+ACTV 24 ore',
'T.Fusina Ve+ACTV 72 ore', 'NA-24H metropolitano ORD+2',
'Tariffa carozzina', 'Aeroporto-Venezia AR ONLINE',
'ARRIVA Extra tr.8-9-10 BORDO', '24hAerAR-Tpl28,90-CVe5,10',
'L.11-auto "C"da 4,01 a 4,50 mt', '24H metropolitano ORD online',
'ARRIVA VENETO tratta 2', 'Extra tratte 2-3-4 BORDO',
'Bus+People mover online', '24ore online no aerobus',
'ARRIVAExtra tr.2-3-4 BORDO', 'ARRIVA VENETO tratta 7',
'24ore online aerobus CS', 'Ev12-Tpl 57,00-C.Ve3,00',
'ARRIVA Aeroporto BORDO', 'Extra tratta 8-9-10 ',
'Gruppi e scuole online TVM ar', 'Extra tratta 7 TVM',
'Extra tratte 5-6-7 BORDO', 'AtvoCanova+Navig 1corsa online',
'L.11-auto "AeB" fino a 4 metri', 'Gruppi e Scuole AR-SM',
'Biglietto MOTO FINO 50 cc', 'ARRIVA Extra tr. 5-6-7 BORDO',
'Ferry11-carri+35q.rim.', "PeopleMover+Bus+Tram 75'carnet",
'Extra tratta 1 BORDO', 'Atvo Canova+Navig AR', 'Ferry17-AUTOBUS',
'Atvo Canova+Actv 72H', 'ARRIVA Extra tr.1 BORDO',
'NA-24H metropolitano ORD.', 'ARRIVA VENETO tratta 6',
'ARRIVA VENETO tratta 5', '24H metropolitano ORD+1 online',
'Biglietto Soc. Sportive', '24H metropolitano ORD+2 online',
'Ev5-Tpl 33,50-C.Ve1,50', 'Gruppi Organizzati CS',
'Su e Zo per i ponti 2022', 'ARRIVA Integ.Aerop. BORDO',
'Gruppi e scuole online 2viaggi', 'Extra tratta 8-9-10 TVM',
'Traghetto Gratuito', 'Atvo Canova+Navig 1 corsa',
'Gruppi e scuole online TVM cs', 'Ev8-Tpl 45,00-C.Ve3,00',
'Extra tratta 6 TVM', 'NA-24H metropolitano RES+1',
'Ferry11-AUTOBUS', 'Salone Nautico 2022 A/R',
'Salone Nautico 2022 1 corsa', "VENDITA A BORDO 75' CV",
'SpiaggeAR-Tpl 11,75-ComVe1,25', 'Prenotazione Veicolo ABBONATO',
'NA-24H metropolitano RES.', 'Bibione-S.Marco AR',
'Caorle-P.S.Margh-S.Marco AR', 'Caorle-P.S.Margh. + Actv 24H',
'EracleaMare-S.Marco AR', 'Bibione + Actv 24H',
'Lignano-S.Marco AR', 'EracleaMare + Actv 24H',
'NA-24H metropolitano RES+2', 'Extra tratte 8-9-10 BORDO',
'Bigl.urbano CHIOGGIA bordo', 'NA-12h-Tpl 13,40-ComVe4,60',

```
'Ville Venete+24H actv urb+nav', 'Camp.Marina-S.Marco AR',  
"VENDITA A BORDO 75' ORD.", 'Ev3-Tpl 26,50-C.Ve1,50',  
'Atvo Canova+Actv 72H rolling', 'Lignano + Actv 24H',  
'SOSTITUTIVO Pass Imob'], dtype=object)
```

```
In [ ]: # Get the number of unique values of the column 'DESCRIZIONE_TITOLO'  
num_unique_DESCRIZIONE_TITOLO = len(df['DESCRIZIONE_TITOLO'].unique())  
print('The number of unique values of the column DESCRIZIONE_TITOLO is: ', num_unique_DESCRIZIONE_TITOLO)
```

The number of unique values of the column DESCRIZIONE_TITOLO is: 168

```
In [ ]: # Convert the column 'DESCRIZIONE_TITOLO' into upper case  
df['DESCRIZIONE_TITOLO'] = df['DESCRIZIONE_TITOLO'].str.upper()  
# Count the number of unique values of the column 'DESCRIZIONE_TITOLO'  
df['DESCRIZIONE_TITOLO'].value_counts()
```

Out[]: DAILY PASS VENEZIA – AVM	972478
75'-TPL 6,64-COMVE0,86	743608
48H-TPL 24,90-COMVE5,10	600320
72H-TPL 33,40-COMVE6,60	492911
BIGL.AUT.75'MESTRE/LIDO-TSC	422668
7GG-TPL 43,60-COMVE16,40	342870
75'-TPL 13,28-COMVE1,72	232644
BIGLIETTO 72 ORE ROLL. VENICE	170675
720RE ONLINE NO AEROBUS	108357
7 DAYS ONLINE NO AEROBUS	101869
PEOPLEMOVER+BUS+TRAM 75'	85968
BIGL AER-VENEZIA TSC	71552
480RE ONLINE NO AEROBUS	65389
72 ORE R.VENICE ONLINE	59982
TRAGH-TPL 8,82-C.VE1,18	38351
DAILY PASS VENEZIA ONLINE	36709
EXTRA TRATTA 2	33102
CAV -TREP + ACTV 24H	25071
EXTRA TRATTA 3	22710
BICICLETTA "BIGLIETTERIA"	20041
JESOLO + ACTV 24H	17869
NA-24H METROPOLITANO ORD+2	17203
NA-24H METROPOLITANO ORD+1	15662
L.17-AUTO "D" OLTRE METRI 4,50	14081
CAV-TREP – S.MARCO AR	14034
7GGAERAR-TPL56,60-CVE16,40	13902
EXTRA TRATTA 4	13678
JESOLO – S.MARCO AR	13620
7 DAYS ONLINE AEROBUS AR	13366
BIGL.MESTRE/LIDO 75' A BORDO	12055
72 ORE R.VENICE+AEROPORTO CS	12018
TRAGH-TPL 4,41-C.VE0,59	11599
PRENOTAZ OCCASIONALE SI BARRA	11187
48HAERCS-TPL31,90-CVE5,10	9889
72HAERCS-TPL40,40-CVE6,60	9855
L.17-AUTO "AEB" FINO A 4 METRI	8279
72HAERAR-TPL46,40-CVE6,60	8152
24HAERCS-TPL22,90-CVE5,10	8048
ORD. NAVIGAZIONE 75' ONLINE	6976
L.17-AUTO "C"DA 4,01 A 4,50 MT	6941

EXTRA TRATTA 1	6800
AER+BOAT-TPL14,50-C.VE1,50	6693
AEROPORTO-VENEZIA AR	6520
ARRIVA VENETO TRATTA 8-9-10	6323
7GGAERCS-TPL50,60-CVE16,40	6295
72H RVENICE+AEROP.CS ONLINE	6190
720RE ONLINE AEROBUS AR	6055
EXTRA TRATTA 5	5347
72H R.VENICE+AEROP.AR ONLINE	5087
72 ORE R.VENICE+AEROPORTO AR	4963
720RE ONLINE AEROBUS CS	4565
BICICLETTA "PALMARE"	4307
CARNET CHIOGGIA 10C. TICKET	4241
7 DAYS ONLINE AEROBUS CS	3933
T.FUSINA VE+ACTV 24 ORE	3713
75'-TPL 12,60-CVE2,40 ONLINE	3524
BIGL.URBANO CHIOGGIA	3430
GRUPPI E SCUOLE	3421
BORDO 75MIN CARTAVENEZIA	3345
48HAERAR-TPL37,90-CVE5,10	3312
GRUPPI E SCUOLE ONLINE TVM AR	3097
FERRY17-AUTOCARRI+35Q.	3037
SALONE NAUTICO 2022 A/R	2916
ATVO CANOVA+ACTV 72H ONLINE	2788
T.FUSINA VE+ACTV 72 ORE	2691
TARIFFA CAROZZINA	2573
480RE ONLINE AEROBUS CS	2386
ATVOCANOVA+ACTV 72HROLL.ONLINE	2260
BICICLETTA "CONCESSIONARI"	2141
CICLOMOTORE FINO 50CC	2101
SU E ZO PER I PONTI 2022	1998
GRUPPI E SCUOLE AR-SM	1907
L.11-AUTO "D" OLTRE METRI 4,50	1845
EXTRA TRATTA 6	1815
ATVO CANOVA+ACTV 72H	1769
AEROPORTO-VENEZIA AR ONLINE	1697
480RE ONLINE AEROBUS AR	1532
ABBONAMENTO 30 GG.PEOPLEMOVER	1509
CAORLE-P.S.MARGH-S.MARCO AR	1484
L.11-AUTO "AEB" FINO A 4 METRI	1450

BUS+PEOPLE MOVER ONLINE	1339
CAORLE-P.S.MARGH. + ACTV 24H	1297
24HAERAR-TPL28,90-CVE5,10	1209
BIBIONE-S.MARCO AR	1173
L.11-AUTO "C"DA 4,01 A 4,50 MT	1109
ARRIVA VENETO TRATTA 1	1090
EV3-TPL 26,50-C.VE1,50	1000
240RE ONLINE AEROBUS CS	893
BIBIONE + ACTV 24H	845
FERRY11-AUTOCARRI+35Q.	838
EXTRA TRATTA 1 TVM	801
EXTRA TRATTA 8-9-10	766
AEROPORTO-VENEZIA CS ONLINE	749
EXTRA TRATTA 2 TVM	748
SALONE NAUTICO 2022 1 CORSA	718
FERRY17-CARRI+35Q.RIM.	668
PEOPLEMOVER+BUS+TRAM 75'CARNET	648
EXTRA TRATTE 2-3-4 BORDO	641
ERACLEAMARE-S.MARCO AR	606
ARRIVA VENETO TRATTA 4	580
EV5-TPL 33,50-C.VE1,50	558
CAV - TREP + ACTV 72H	544
NA-24H-TPL 14,90-COM.VE5,10	532
EXTRA TRATTA 7	514
ARRIVA VENETO TRATTA 6	509
240RE ONLINE NO AEROBUS	457
EV12-TPL 57,00-C.VE3,00	455
24H METROPOLITANO ORD ONLINE	435
EXTRA TRATTA 3 TVM	398
EXTRA TRATTA 4 TVM	387
24H METROPOLITANO ORD+2 ONLINE	346
ATVO CANOVA+NAVIG 1 CORSA	345
EV8-TPL 45,00-C.VE3,00	308
GRUPPI ORGANIZZATI CS	303
ARRIVA VENETO AEROPORTO	298
ARRIVA VENETO TRATTA 7	279
BIGLIETTO SOC. SPORTIVE	274
LIGNANO-S.MARCO AR	274
BAGAGLIO CARTAVENEZIA	270
GRUPPI E SCUOLE ONLINE TVM CS	264

ATVO CANOVA+NAVIG AR	257
BIGLIETTO MOTO FINO 50 CC	244
24H METROPOLITANO ORD+1 ONLINE	239
CICLOMOTORE OLTRE 50CC	238
ARRIVA VENETO TRATTA 3	237
ARRIVA EXTRA TR.8-9-10 BORDO	236
240RE ONLINE AEROBUS AR	222
ARRIVA VENETO TRATTA 2	206
EXTRA TRATTA 1 BORDO	205
ERACLEAMARE + ACTV 24H	204
ARRIVA EXTRA TR.1 BORDO	182
GRUPPI E SCUOLE ONLINE 2VIAGGI	180
ARRIVAEXTRA TR.2-3-4 BORDO	171
ARRIVA EXTRA TR. 5-6-7 BORDO	166
EXTRA TRATTA 5 TVM	160
BIGLIETTO MERCI C.SEMPLICE	159
FERRY17-AUTOBUS	134
SPIAGGEAR-TPL 11,75-COMVE1,25	125
NA-TRAGHETTO ORDINARIO	109
ATVO CANOVA+NAVIG AR ONLINE	102
NA-24H METROPOLITANO ORD.	97
ATVOCANOVA+NAVIG 1CORSIA ONLINE	80
ARRIVA VENETO TRATTA 5	74
MENS. COSE ANIMALI RETE UNICA	48
EXTRA TRATTE 5-6-7 BORDO	38
FERRY17-TRASPORTI PERICOLOSI	32
NA-24H METROPOLITANO RES+1	31
LIGNANO + ACTV 24H	25
EXTRA TRATTA 8-9-10 TVM	23
ARRIVA AEROPORTO BORDO	22
PRENOTAZIONE VEICOLO ABBONATO	20
VILLE VENETE+24H ACTV URB+NAV	18
EXTRA TRATTA 7 TVM	18
NA-24H METROPOLITANO RES+2	17
ARRIVA INTEG.AEROP. BORDO	14
NA-24H METROPOLITANO RES.	13
VENDITA A BORDO 75' CV	13
FERRY11-CARRI+35Q.RIM.	13
BIGL.URBANO CHIOGGIA BORDO	12
VENDITA A BORDO 75' ORD.	10

EXTRA TRATTA 6 TVM	9
TRAGHETTO GRATUITO	8
FERRY11-AUTOBUS	3
EXTRA TRATTE 8-9-10 BORDO	3
NA-12H-TPL 13,40-COMVE4,60	3
CAMP.MARINA-S.MARCO AR	2
ATVO CANOVA+ACTV 72H ROLLING	1
SOSTITUTIVO PASS IMOB	1

Name: DESCRIZIONE_TITOLO, dtype: int64

One-day tickets

```
In [ ]: # Which type of ticket are one-day tickets and how many are there?
df[df['DESCRIZIONE_TITOLO'].str.contains('GIORNALIERO|24H|24ORE|24 ORE|DAILY')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: DAILY PASS VENEZIA - AVM          972478
        DAILY PASS VENEZIA ONLINE        36709
        CAV -TREP + ACTV 24H              25071
        JESOLO + ACTV 24H                 17869
        NA-24H METROPOLITANO ORD+2        17203
        NA-24H METROPOLITANO ORD+1        15662
        24HAERCS-TPL22,90-CVE5,10        8048
        T.FUSINA VE+ACTV 24 ORE           3713
        CAORLE-P.S.MARGH. + ACTV 24H      1297
        24HAERAR-TPL28,90-CVE5,10        1209
        24ORE ONLINE AEROBUS CS           893
        BIBIONE + ACTV 24H                845
        NA-24H-TPL 14,90-COM.VE5,10       532
        24ORE ONLINE NO AEROBUS           457
        24H METROPOLITANO ORD ONLINE      435
        24H METROPOLITANO ORD+2 ONLINE    346
        24H METROPOLITANO ORD+1 ONLINE    239
        24ORE ONLINE AEROBUS AR           222
        ERACLEAMARE + ACTV 24H            204
        NA-24H METROPOLITANO ORD.          97
        NA-24H METROPOLITANO RES+1         31
        LIGNANO + ACTV 24H                 25
        VILLE VENETE+24H ACTV URB+NAV     18
        NA-24H METROPOLITANO RES+2        17
        NA-24H METROPOLITANO RES.         13
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('GIORNALIERO|24H|24ORE|24 ORE|DAILY'), 'TICKET_CODE'] = '1'
```

```
In [ ]: # TICKET_CODE = 1: Information about one-day tickets
print("The number of one-day tickets is: ", df[df['TICKET_CODE'] == '1'].shape[0])
print("The number of tickets for each type of one-day ticket is: ")
df[df['TICKET_CODE'] == '1']['DESCRIZIONE_TITOLO'].value_counts()
```

The number of one-day tickets is: 1103633

The number of tickets for each type of one-day ticket is:


```
Out[ ]: DAILY PASS VENEZIA - AVM          972478
        DAILY PASS VENEZIA ONLINE        36709
        CAV -TREP + ACTV 24H              25071
        JESOLO + ACTV 24H                  17869
        NA-24H METROPOLITANO ORD+2         17203
        NA-24H METROPOLITANO ORD+1         15662
        24HAERCS-TPL22,90-CVE5,10         8048
        T.FUSINA VE+ACTV 24 ORE            3713
        CAORLE-P.S.MARGH. + ACTV 24H       1297
        24HAERAR-TPL28,90-CVE5,10        1209
        240RE ONLINE AEROBUS CS            893
        BIBIONE + ACTV 24H                 845
        NA-24H-TPL 14,90-COM.VE5,10        532
        240RE ONLINE NO AEROBUS            457
        24H METROPOLITANO ORD ONLINE        435
        24H METROPOLITANO ORD+2 ONLINE      346
        24H METROPOLITANO ORD+1 ONLINE      239
        240RE ONLINE AEROBUS AR            222
        ERACLEAMARE + ACTV 24H             204
        NA-24H METROPOLITANO ORD.           97
        NA-24H METROPOLITANO RES+1          31
        LIGNANO + ACTV 24H                  25
        VILLE VENETE+24H ACTV URB+NAV       18
        NA-24H METROPOLITANO RES+2          17
        NA-24H METROPOLITANO RES.           13
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: print("Information about the tickets with code 1 related to the serial number: ")
df[df['TICKET_CODE'] == '1'].groupby('DESCRIZIONE_TITOLO')['SERIALE'].value_counts().groupby('DESCRIZIONE_TITOLO').
```

Information about the tickets with code 1 related to the serial number:

Out[]:

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
24H METROPOLITANO ORD ONLINE	147.0	2.959184	2.093295	1.0	2.00	2.0	3.50	11.0
24H METROPOLITANO ORD+1 ONLINE	62.0	3.854839	1.998744	1.0	3.00	4.0	5.00	13.0
24H METROPOLITANO ORD+2 ONLINE	75.0	4.613333	3.533214	1.0	2.00	4.0	5.50	20.0
24HAERAR-TPL28,90-CVE5,10	299.0	4.043478	2.642220	1.0	2.00	4.0	5.00	24.0
24HAERCS-TPL22,90-CVE5,10	2092.0	3.847036	2.333202	1.0	2.00	4.0	5.00	16.0
24ORE ONLINE AEROBUS AR	49.0	4.530612	2.102598	1.0	3.00	5.0	6.00	9.0
24ORE ONLINE AEROBUS CS	187.0	4.775401	2.563775	1.0	3.00	5.0	6.00	15.0
24ORE ONLINE NO AEROBUS	122.0	3.745902	2.283833	1.0	2.00	3.0	5.00	12.0
BIBIONE + ACTV 24H	312.0	2.708333	1.315818	1.0	2.00	3.0	4.00	8.0
CAORLE-P.S.MARGH. + ACTV 24H	464.0	2.795259	1.370429	1.0	2.00	3.0	3.00	9.0
CAV -TREP + ACTV 24H	7370.0	3.401764	1.640898	1.0	2.00	3.0	4.00	17.0
DAILY PASS VENEZIA - AVM	260958.0	3.726569	2.136298	1.0	2.00	3.0	5.00	119.0
DAILY PASS VENEZIA ONLINE	10147.0	3.617720	2.090693	1.0	2.00	3.0	5.00	23.0
ERACLEAMARE + ACTV 24H	79.0	2.582278	1.297014	1.0	1.50	2.0	3.00	6.0
JESOLO + ACTV 24H	5534.0	3.228948	1.523413	1.0	2.00	3.0	4.00	12.0
LIGNANO + ACTV 24H	9.0	2.777778	1.394433	1.0	2.00	3.0	4.00	5.0
NA-24H METROPOLITANO ORD+1	3893.0	4.023118	2.531097	1.0	2.00	3.0	5.00	30.0
NA-24H METROPOLITANO ORD+2	4633.0	3.713145	2.747969	1.0	2.00	3.0	5.00	33.0
NA-24H METROPOLITANO ORD.	31.0	3.129032	2.186960	1.0	2.00	3.0	4.00	12.0
NA-24H METROPOLITANO RES+1	9.0	3.444444	1.740051	2.0	2.00	3.0	4.00	7.0
NA-24H METROPOLITANO RES+2	2.0	8.500000	4.949747	5.0	6.75	8.5	10.25	12.0
NA-24H METROPOLITANO RES.	5.0	2.600000	1.140175	1.0	2.00	3.0	3.00	4.0
NA-24H-TPL 14,90-COM.VE5,10	138.0	3.855072	5.098376	1.0	2.00	3.0	4.00	48.0

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
T.FUSINA VE+ACTV 24 ORE	1144.0	3.245629	2.154565	1.0	2.00	3.0	4.00	44.0
VILLE VENETE+24H ACTV URB+NAV	6.0	3.000000	1.095445	2.0	2.25	3.0	3.00	5.0

Two days tickets

```
In [ ]: # Which type of ticket are two-day tickets and how many are there?
df[df['DESCRIZIONE_TITOLO'].str.contains('48H|48ORE|48 ORE')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: 48H-TPL 24,90-COMVE5,10      600320
48ORE ONLINE NO AEROBUS          65389
48HAERCS-TPL31,90-CVE5,10        9889
48HAERAR-TPL37,90-CVE5,10        3312
48ORE ONLINE AEROBUS CS          2386
48ORE ONLINE AEROBUS AR          1532
Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('48H|48ORE|48 ORE'), 'TICKET_CODE'] = '2'
```

```
In [ ]: # TICKET_CODE = 2: Information about two-day tickets
print("The number of two-day tickets is: ", df[df['TICKET_CODE'] == '2'].shape[0])
print("The number of tickets for each type of two-day ticket is: ")
df[df['TICKET_CODE'] == '2']['DESCRIZIONE_TITOLO'].value_counts()
```

The number of two-day tickets is: 682828
The number of tickets for each type of two-day ticket is:

```
Out[ ]: 48H-TPL 24,90-COMVE5,10      600320
48ORE ONLINE NO AEROBUS          65389
48HAERCS-TPL31,90-CVE5,10        9889
48HAERAR-TPL37,90-CVE5,10        3312
48ORE ONLINE AEROBUS CS          2386
48ORE ONLINE AEROBUS AR          1532
Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: print("Information about the tickets with code 2 related to the serial number: ")
df[df['TICKET_CODE'] == '2'].groupby('DESCRIZIONE_TITOLO')['SERIALE'].value_counts().groupby('DESCRIZIONE_TITOLO').
```

Information about the tickets with code 2 related to the serial number:

```
Out[ ]:
```

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
48H-TPL 24,90-COMVE5,10	94201.0	6.372756	3.520071	1.0	4.0	6.0	8.00	68.0
48HAERAR-TPL37,90-CVE5,10	405.0	8.177778	4.356968	1.0	5.0	7.0	11.00	28.0
48HAERCS-TPL31,90-CVE5,10	1378.0	7.176343	3.897204	1.0	4.0	7.0	9.00	23.0
48ORE ONLINE AEROBUS AR	200.0	7.660000	3.281500	2.0	5.0	7.0	9.25	19.0
48ORE ONLINE AEROBUS CS	344.0	6.936047	3.937038	1.0	4.0	7.0	9.00	25.0
48ORE ONLINE NO AEROBUS	10733.0	6.092332	3.450538	1.0	4.0	6.0	8.00	37.0

Three days tickets

```
In [ ]: # Which type of ticket are three-day tickets and how many are there?
# Do not consider the ticket that contains also 75
df[df['DESCRIZIONE_TITOLO'].str.contains('72H|72ORE|72 ORE')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: 72H-TPL 33,40-COMVE6,60      492911
        BIGLIETTO 72 ORE ROLL. VENICE 170675
        72ORE ONLINE NO AEROBUS      108357
        72 ORE R.VENICE ONLINE       59982
        72 ORE R.VENICE+AEROPORTO CS  12018
        72HAERCS-TPL40,40-CVE6,60    9855
        72HAERAR-TPL46,40-CVE6,60    8152
        72H RVENICE+AEROP.CS ONLINE   6190
        72ORE ONLINE AEROBUS AR       6055
        72H R.VENICE+AEROP.AR ONLINE  5087
        72 ORE R.VENICE+AEROPORTO AR  4963
        72ORE ONLINE AEROBUS CS       4565
        ATVO CANOVA+ACTV 72H ONLINE    2788
        T.FUSINA VE+ACTV 72 ORE       2691
        ATVOCANOVA+ACTV 72HROLL.ONLINE 2260
        ATVO CANOVA+ACTV 72H          1769
        CAV - TREP + ACTV 72H         544
        ATVO CANOVA+ACTV 72H ROLLING   1
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('72H|72ORE|72 ORE'), 'TICKET_CODE'] = '3'
```

```
In [ ]: # TICKET_CODE = 3: Information about three-day tickets
print("The number of three-day tickets is: ", df[df['TICKET_CODE'] == '3'].shape[0])
print("The number of tickets for each type of three-day ticket is: ")
df[df['TICKET_CODE'] == '3']['DESCRIZIONE_TITOLO'].value_counts()
```

The number of three-day tickets is: 898863

The number of tickets for each type of three-day ticket is:

```
Out[ ]: 72H-TPL 33,40-COMVE6,60      492911
        BIGLIETTO 72 ORE ROLL. VENICE 170675
        720RE ONLINE NO AEROBUS      108357
        72 ORE R.VENICE ONLINE       59982
        72 ORE R.VENICE+AEROPORTO CS 12018
        72HAERCS-TPL40,40-CVE6,60    9855
        72HAERAR-TPL46,40-CVE6,60    8152
        72H RVENICE+AEROP.CS ONLINE   6190
        720RE ONLINE AEROBUS AR       6055
        72H R.VENICE+AEROP.AR ONLINE  5087
        72 ORE R.VENICE+AEROPORTO AR   4963
        720RE ONLINE AEROBUS CS       4565
        ATV0 CANOVA+ACTV 72H ONLINE    2788
        T.FUSINA VE+ACTV 72 ORE       2691
        ATVOCANOVA+ACTV 72HROLL.ONLINE 2260
        ATV0 CANOVA+ACTV 72H          1769
        CAV - TREP + ACTV 72H         544
        ATV0 CANOVA+ACTV 72H ROLLING   1
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: print("Information about the tickets with code 3 related to the serial number: ")
df[df['TICKET_CODE'] == '3'].groupby('DESCRIZIONE_TITOLO')['SERIALE'].value_counts().groupby('DESCRIZIONE_TITOLO').
```

Information about the tickets with code 3 related to the serial number:

Out[]:

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
72 ORE R.VENICE ONLINE	7418.0	8.086007	4.727938	1.0	5.0	7.0	11.0	113.0
72 ORE R.VENICE+AEROPORTO AR	515.0	9.636893	4.598913	1.0	7.0	9.0	13.0	26.0
72 ORE R.VENICE+AEROPORTO CS	1338.0	8.982063	4.823866	1.0	5.0	8.0	12.0	28.0
72H R.VENICE+AEROP.AR ONLINE	496.0	10.256048	5.270938	1.0	7.0	10.0	13.0	29.0
72H RVENICE+AEROP.CS ONLINE	670.0	9.238806	4.770016	1.0	6.0	9.0	12.0	35.0
72H-TPL 33,40-COMVE6,60	59213.0	8.324371	4.569172	1.0	5.0	8.0	11.0	109.0
72HAERAR-TPL46,40-CVE6,60	809.0	10.076638	4.653494	1.0	7.0	9.0	13.0	27.0
72HAERCS-TPL40,40-CVE6,60	1048.0	9.403626	4.767394	1.0	6.0	9.0	12.0	27.0
72ORE ONLINE AEROBUS AR	610.0	9.926230	4.843305	1.0	7.0	9.0	13.0	25.0
72ORE ONLINE AEROBUS CS	471.0	9.692144	4.993690	1.0	6.0	9.0	12.0	35.0
72ORE ONLINE NO AEROBUS	13423.0	8.072488	4.433573	1.0	5.0	7.0	11.0	35.0
ATVO CANOVA+ACTV 72H	213.0	8.305164	4.610879	1.0	5.0	7.0	11.0	24.0
ATVO CANOVA+ACTV 72H ONLINE	298.0	9.355705	5.335320	1.0	6.0	8.0	13.0	33.0
ATVO CANOVA+ACTV 72H ROLLING	1.0	1.000000	NaN	1.0	1.0	1.0	1.0	1.0
ATVOCANOVA+ACTV 72HROLL.ONLINE	232.0	9.741379	5.201766	1.0	6.0	9.0	13.0	25.0
BIGLIETTO 72 ORE ROLL. VENICE	21866.0	7.805497	4.399997	1.0	5.0	7.0	10.0	68.0
CAV - TREP + ACTV 72H	56.0	9.714286	2.852204	1.0	8.0	10.0	12.0	15.0
T.FUSINA VE+ACTV 72 ORE	360.0	7.475000	5.271402	1.0	4.0	7.0	9.0	75.0

Seven days tickets

```
In [ ]: # Which type of ticket are weekly tickets and how many are there?
# Exclude the tickets that contains also 72, 75 that are three-day tickets, 17, 48h, 57 that are other types of tic
# 'tratt*' and 'tr' that are reserved to specific routes
df[df['DESCRIZIONE_TITOLO'].str.contains('7GG|7DAYS|7 DAYS')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: 7GG-TPL 43,60-COMVE16,40      342870
        7 DAYS ONLINE NO AEROBUS      101869
        7GGAERAR-TPL56,60-CVE16,40    13902
        7 DAYS ONLINE AEROBUS AR       13366
        7GGAERCS-TPL50,60-CVE16,40     6295
        7 DAYS ONLINE AEROBUS CS        3933
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('7GG|7DAYS|7 DAYS'), 'TICKET_CODE'] = '4'
```

```
In [ ]: # TICKET_CODE = 4: Information about weekly tickets
print("The number of weekly tickets is: ", df[df['TICKET_CODE'] == '4'].shape[0])
print("The number of tickets for each type of weekly ticket is: ")
df[df['TICKET_CODE'] == '4']['DESCRIZIONE_TITOLO'].value_counts()
```

```
The number of weekly tickets is: 482235
The number of tickets for each type of weekly ticket is:
```

```
Out[ ]: 7GG-TPL 43,60-COMVE16,40      342870
        7 DAYS ONLINE NO AEROBUS      101869
        7GGAERAR-TPL56,60-CVE16,40    13902
        7 DAYS ONLINE AEROBUS AR       13366
        7GGAERCS-TPL50,60-CVE16,40     6295
        7 DAYS ONLINE AEROBUS CS        3933
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: print("Information about the tickets with code 4 related to the serial number: ")
df[df['TICKET_CODE'] == '4'].groupby('DESCRIZIONE_TITOLO')['SERIALE'].value_counts().groupby('DESCRIZIONE_TITOLO').
```

```
Information about the tickets with code 4 related to the serial number:
```


Out []:

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
7 DAYS ONLINE AEROBUS AR	910.0	14.687912	8.249103	1.0	9.0	14.0	19.0	53.0
7 DAYS ONLINE AEROBUS CS	265.0	14.841509	8.184275	1.0	10.0	14.0	19.0	47.0
7 DAYS ONLINE NO AEROBUS	7815.0	13.035061	7.271917	1.0	8.0	12.0	17.0	57.0
7GG-TPL 43,60-COMVE16,40	24879.0	13.781502	7.986574	1.0	8.0	13.0	18.0	121.0
7GGAERAR-TPL56,60-CVE16,40	881.0	15.779796	7.995402	1.0	10.0	15.0	21.0	46.0
7GGAERCS-TPL50,60-CVE16,40	420.0	14.988095	8.377567	1.0	9.0	14.5	20.0	57.0

Monthly tickets

```
In [ ]: # Whick type of ticket are monthly tickets and how many are there?
df[df['DESCRIZIONE_TITOLO'].str.contains('MENSILE|30GG|30 GG|MENS')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out [ ]: ABBONAMENTO 30 GG.PEOPLEMOVER      1509
MENS. COSE ANIMALI RETE UNICA           48
Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('MENSILE|30GG|30 GG|MENS'), 'TICKET_CODE'] = '5'
```

```
In [ ]: # If DESCRIZIONE_TITOLO contains 'STUDENTE' or 'STUD' update the column 'TICKET_CODE' with '5-STUD' only for the ti
df.loc[(df['TICKET_CODE'] == '5') & (df['DESCRIZIONE_TITOLO'].str.contains('STUDENTE|STUD|STUD')), 'TICKET_CODE'] =
# If DESCRIZIONE_TITOLO contains 'LAVORATORE' or 'LAV' update the column 'TICKET_CODE' with '5-WKRS' only for the t
df.loc[(df['TICKET_CODE'] == '5') & (df['DESCRIZIONE_TITOLO'].str.contains('LAVORATORE|LAV|LAV')), 'TICKET_CODE'] =
# If DESCRIZIONE_TITOLO contains 'OVER 65' or '65+' or 'PENSIONATI' update the column 'TICKET_CODE' with '5-RET' on
df.loc[(df['TICKET_CODE'] == '5') & (df['DESCRIZIONE_TITOLO'].str.contains('OVER 65|65+|PENSIONATI')), 'TICKET_CODE']
```

```
In [ ]: # According to the page of agevolation of specific categories of people available at the site web of ACTV
# (https://actv.avmspa.it/it/content/categorie-agevolate-0), the DDRG 1201-1297/2022 tickets are monthly tickets fo

# Which type of ticket are yearly tickets for blind people and how many are there?
df[df['DESCRIZIONE_TITOLO'].str.contains('DDGR1201-1297/2022')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: Series([], Name: DESCRIZIONE_TITOLO, dtype: int64)
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('DDGR1201-1297/2022'), 'TICKET_CODE'] = '5'
```

```
In [ ]: # TICKET_CODE = 5: Information about monthly tickets
print("The number of monthly tickets is: ", df[df['TICKET_CODE'] == '5'].shape[0])
print("The number of monthly tickets for students is: ", df[df['TICKET_CODE'] == '5-STUD'].shape[0])
print("The number of monthly tickets for workers is: ", df[df['TICKET_CODE'] == '5-WKRS'].shape[0])
print("The number of monthly tickets for retired people is: ", df[df['TICKET_CODE'] == '5-RET'].shape[0])

print("The number of tickets for each type of monthly ticket (including the subtypes) is: ")
df[df['TICKET_CODE'].isin(['5', '5-STUD', '5-WKRS', '5-RET'])].groupby('TICKET_CODE')['DESCRIZIONE_TITOLO'].value_c
```

```
The number of monthly tickets is: 1557
The number of monthly tickets for students is: 0
The number of monthly tickets for workers is: 0
The number of monthly tickets for retired people is: 0
The number of tickets for each type of monthly ticket (including the subtypes) is:
```

```
Out[ ]: TICKET_CODE  DESCRIZIONE_TITOLO
5                ABBONAMENTO 30 GG.PEOPLEMOVER    1509
                MENS. COSE ANIMALI RETE UNICA      48
Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: print("Information about the tickets with code 5 (including the subtypes) related to the serial number: ")
df[df['TICKET_CODE'].isin(['5', '5-STUD', '5-WKRS', '5-RET'])].groupby(['TICKET_CODE', 'DESCRIZIONE_TITOLO'])['SERI
```

```
Information about the tickets with code 5 (including the subtypes) related to the serial number:
```

```
Out[ ]:
```

		count	mean	std	min	25%	50%	75%	max
TICKET_CODE	DESCRIZIONE_TITOLO								
5	ABBONAMENTO 30 GG.PEOPLEMOVER	223.0	6.766816	11.755366	1.0	1.0	1.0	4.0	58.0
	MENS. COSE ANIMALI RETE UNICA	5.0	9.600000	8.848729	1.0	1.0	12.0	12.0	22.0

Yearly tickets

```
In [ ]: # Which type of ticket are yearly tickets and how many are there?
df[df['DESCRIZIONE_TITOLO'].str.contains('ANNUALE|ANN|12MESI|12 MESI')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: Series([], Name: DESCRIZIONE_TITOLO, dtype: int64)
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('ANNUALE|ANN|12MESI|12 MESI'), 'TICKET_CODE'] = '6'
```

```
In [ ]: # If DESCRIZIONE_TITOLO contains 'STUDENTE' or 'STUD' update the column 'TICKET_CODE' with '6-STUD' only for the ti
df.loc[(df['TICKET_CODE'] == '6') & (df['DESCRIZIONE_TITOLO'].str.contains('STUDENTE|STUD|STUD')), 'TICKET_CODE'] =
# If DESCRIZIONE_TITOLO contains 'LAVORATORE' or 'LAV' update the column 'TICKET_CODE' with '6-WKRS' only for the t
df.loc[(df['TICKET_CODE'] == '6') & (df['DESCRIZIONE_TITOLO'].str.contains('LAVORATORE|LAV|LAV')), 'TICKET_CODE'] =
# If DESCRIZIONE_TITOLO contains 'OVER 65' or '65+' or 'PENSIONATI' update the column 'TICKET_CODE' with '6-RET' on
df.loc[(df['TICKET_CODE'] == '6') & (df['DESCRIZIONE_TITOLO'].str.contains('OVER 65|65+|PENSIONATI')), 'TICKET_CODE']
```

```
In [ ]: # According to the page of agevolation of specific categories of people available at the site web of ACTV
# (https://actv.avmspa.it/it/content/categorie-agevolate-0), the for OVER 75 are yearly tickets for free

# Which type of ticket are yearly tickets for OVER 75 and how many are there?
df[df['DESCRIZIONE_TITOLO'].str.contains('OVER 75|OVER75|PENSIONATI')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: Series([], Name: DESCRIZIONE_TITOLO, dtype: int64)
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('OVER 75|OVER75|PENSIONATI'), 'TICKET_CODE'] = '6-RET'
```

```
In [ ]: # According to the page of yearly tickets available at the site web of ACTV
# (https://actv.avmspa.it/it/content/abbonamento-annuale-0), the bus pass for students has a validity of 12 months

# Which type of ticket are yearly tickets for students and how many are there?
# Exclude the tickets that have already the field TICKET_CODE populated with 5-STUD or 6-STUD
df[(df['DESCRIZIONE_TITOLO'].str.contains('STUDENTE|STUD|STUD')) & ~ (df['TICKET_CODE'].isin(['5-STUD', '6-STUD']))]
```

```
Out[ ]: Series([], Name: DESCRIZIONE_TITOLO, dtype: int64)
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[(df['DESCRIZIONE_TITOLO'].str.contains('STUDENTE|STUD|STUD')) & ~ (df['TICKET_CODE'].isin(['5-STUD', '6-STUD'])), 'TICKET_CODE'] = '6-STUD'
```

```
In [ ]: # TICKET_CODE = 6: Information about annual tickets
print("The number of annual tickets is: ", df[df['TICKET_CODE'] == '6'].shape[0])
print("The number of annual tickets for students is: ", df[df['TICKET_CODE'] == '6-STUD'].shape[0])
print("The number of annual tickets for workers is: ", df[df['TICKET_CODE'] == '6-WKRS'].shape[0])
print("The number of annual tickets for retired people is: ", df[df['TICKET_CODE'] == '6-RET'].shape[0])

print("The number of tickets for each type of annual ticket (including the subtypes) is: ")
df[df['TICKET_CODE'].isin(['6', '6-STUD', '6-WKRS', '6-RET'])].groupby('TICKET_CODE')['DESCRIZIONE_TITOLO'].value_c

The number of annual tickets is: 0
The number of annual tickets for students is: 0
The number of annual tickets for workers is: 0
The number of annual tickets for retired people is: 0
The number of tickets for each type of annual ticket (including the subtypes) is:
Out [ ]: Series([], Name: DESCRIZIONE_TITOLO, dtype: int64)
```

```
In [ ]: print("Information about the tickets with code 6 (including the subtypes) related to the serial number: ")
df[df['TICKET_CODE'].isin(['6', '6-STUD', '6-WKRS', '6-RET'])].groupby(['TICKET_CODE', 'DESCRIZIONE_TITOLO'])['SERI

Information about the tickets with code 6 (including the subtypes) related to the serial number:
Out [ ]:  count  mean  std  min  25%  50%  75%  max
          _____
```

75 minutes tickets

```
In [ ]: # Which type of ticket are 75' (75 minutes) tickets and how many are there?
df[df['DESCRIZIONE_TITOLO'].str.contains('75\'|75MIN|75 MIN')]['DESCRIZIONE_TITOLO'].value_counts()
```

```
Out[ ]: 75'-TPL 6,64-COMVE0,86          743608
        BIGL.AUT.75'MESTRE/LIDO-TSC      422668
        75'-TPL 13,28-COMVE1,72         232644
        PEOPLEMOVER+BUS+TRAM 75'        85968
        BIGL.MESTRE/LIDO 75' A BORDO     12055
        ORD. NAVIGAZIONE 75' ONLINE      6976
        75'-TPL 12,60-CVE2,40 ONLINE     3524
        BORDO 75MIN CARTAVENEZIA         3345
        PEOPLEMOVER+BUS+TRAM 75'CARNET    648
        VENDITA A BORDO 75' CV            13
        VENDITA A BORDO 75' ORD.          10
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[df['DESCRIZIONE_TITOLO'].str.contains('75\'|75MIN|75 MIN'), 'TICKET_CODE'] = '7'
```

```
In [ ]: # TICKET_CODE = 7: Information about 75' (75 minutes) tickets
print("The number of 75' (75 minutes) tickets is: ", df[df['TICKET_CODE'] == '7'].shape[0])
print("The number of tickets for each type of 75' (75 minutes) ticket is: ")
df[df['TICKET_CODE'] == '7'].groupby('TICKET_CODE')['DESCRIZIONE_TITOLO'].value_counts()

The number of 75' (75 minutes) tickets is: 1511459
The number of tickets for each type of 75' (75 minutes) ticket is:
```

```
Out[ ]: TICKET_CODE DESCRIZIONE_TITOLO
        7          75'-TPL 6,64-COMVE0,86          743608
          BIGL.AUT.75'MESTRE/LIDO-TSC      422668
          75'-TPL 13,28-COMVE1,72         232644
          PEOPLEMOVER+BUS+TRAM 75'        85968
          BIGL.MESTRE/LIDO 75' A BORDO     12055
          ORD. NAVIGAZIONE 75' ONLINE      6976
          75'-TPL 12,60-CVE2,40 ONLINE     3524
          BORDO 75MIN CARTAVENEZIA         3345
          PEOPLEMOVER+BUS+TRAM 75'CARNET    648
          VENDITA A BORDO 75' CV            13
          VENDITA A BORDO 75' ORD.          10
        Name: DESCRIZIONE_TITOLO, dtype: int64
```

```
In [ ]: print("Information about the tickets with code 7 related to the serial number: ")
df[df['TICKET_CODE'] == '7'].groupby('DESCRIZIONE_TITOLO')['SERIALE'].value_counts().groupby('DESCRIZIONE_TITOLO').
```

Information about the tickets with code 7 related to the serial number:

Out[]:

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
75'-TPL 12,60-CVE2,40 ONLINE	2034.0	1.732547	0.605153	1.0	1.0	2.0	2.00	4.0
75'-TPL 13,28-COMVE1,72	127257.0	1.828143	0.572819	1.0	2.0	2.0	2.00	7.0
75'-TPL 6,64-COMVE0,86	669753.0	1.110272	0.370967	1.0	1.0	1.0	1.00	26.0
BIGL.AUT.75'MESTRE/LIDO-TSC	295281.0	1.431409	0.980805	1.0	1.0	1.0	2.00	20.0
BIGL.MESTRE/LIDO 75' A BORDO	11852.0	1.017128	0.132965	1.0	1.0	1.0	1.00	3.0
BORDO 75MIN CARTAVENEZIA	3196.0	1.046621	0.297119	1.0	1.0	1.0	1.00	10.0
ORD. NAVIGAZIONE 75' ONLINE	6175.0	1.129717	0.384991	1.0	1.0	1.0	1.00	4.0
PEOPLEMOVER+BUS+TRAM 75'	84609.0	1.016062	0.136532	1.0	1.0	1.0	1.00	4.0
PEOPLEMOVER+BUS+TRAM 75'CARNET	292.0	2.219178	1.825716	1.0	1.0	2.0	2.00	10.0
VENDITA A BORDO 75' CV	12.0	1.083333	0.288675	1.0	1.0	1.0	1.00	2.0
VENDITA A BORDO 75' ORD.	8.0	1.250000	0.462910	1.0	1.0	1.0	1.25	2.0

Other types of tickets

```
In [ ]: # Which type of ticket are other tickets and how many are there?
# The other tickets are the tickets that are not already classified in the previous categories
df[~df['TICKET_CODE'].isin(['1', '2', '3', '4', '5', '5-STUD', '5-WKRS', '5-RET', '6', '6-STUD', '6-WKRS', '6-RET',
```

Out[]:	BIGL AER-VENEZIA TSC	71552
	TRAGH-TPL 8,82-C.VE1,18	38351
	EXTRA TRATTA 2	33102
	EXTRA TRATTA 3	22710
	BICICLETTA "BIGLIETTERIA"	20041
	L.17-AUTO "D" OLTRE METRI 4,50	14081
	CAV-TREP - S.MARCO AR	14034
	EXTRA TRATTA 4	13678
	JESOLO - S.MARCO AR	13620
	TRAGH-TPL 4,41-C.VE0,59	11599
	PRENOTAZ OCCASIONALE SI BARRA	11187
	L.17-AUTO "AEB" FINO A 4 METRI	8279
	L.17-AUTO "C"DA 4,01 A 4,50 MT	6941
	EXTRA TRATTA 1	6800
	AER+BOAT-TPL14,50-C.VE1,50	6693
	AEROPORTO-VENEZIA AR	6520
	ARRIVA VENETO TRATTA 8-9-10	6323
	EXTRA TRATTA 5	5347
	BICICLETTA "PALMARE"	4307
	CARNET CHIOGGIA 10C. TICKET	4241
	BIGL.URBANO CHIOGGIA	3430
	GRUPPI E SCUOLE	3421
	GRUPPI E SCUOLE ONLINE TVM AR	3097
	FERRY17-AUTOCARRI+35Q.	3037
	SALONE NAUTICO 2022 A/R	2916
	TARIFFA CAROZZINA	2573
	BICICLETTA "CONCESSIONARI"	2141
	CICLOMOTORE FINO 50CC	2101
	SU E ZO PER I PONTI 2022	1998
	GRUPPI E SCUOLE AR-SM	1907
	L.11-AUTO "D" OLTRE METRI 4,50	1845
	EXTRA TRATTA 6	1815
	AEROPORTO-VENEZIA AR ONLINE	1697
	CAORLE-P.S.MARGH-S.MARCO AR	1484
	L.11-AUTO "AEB" FINO A 4 METRI	1450
	BUS+PEOPLE MOVER ONLINE	1339
	BIBIONE-S.MARCO AR	1173
	L.11-AUTO "C"DA 4,01 A 4,50 MT	1109
	ARRIVA VENETO TRATTA 1	1090
	EV3-TPL 26,50-C.VE1,50	1000

FERRY11-AUTOCARRI+35Q.	838
EXTRA TRATTA 1 TVM	801
EXTRA TRATTA 8-9-10	766
AEROPORTO-VENEZIA CS ONLINE	749
EXTRA TRATTA 2 TVM	748
SALONE NAUTICO 2022 1 CORSA	718
FERRY17-CARRI+35Q.RIM.	668
EXTRA TRATTE 2-3-4 BORDO	641
ERACLEAMARE-S.MARCO AR	606
ARRIVA VENETO TRATTA 4	580
EV5-TPL 33,50-C.VE1,50	558
EXTRA TRATTA 7	514
ARRIVA VENETO TRATTA 6	509
EV12-TPL 57,00-C.VE3,00	455
EXTRA TRATTA 3 TVM	398
EXTRA TRATTA 4 TVM	387
ATVO CANOVA+NAVIG 1 CORSA	345
EV8-TPL 45,00-C.VE3,00	308
GRUPPI ORGANIZZATI CS	303
ARRIVA VENETO AEROPORTO	298
ARRIVA VENETO TRATTA 7	279
LIGNANO-S.MARCO AR	274
BIGLIETTO SOC. SPORTIVE	274
BAGAGLIO CARTAVENEZIA	270
GRUPPI E SCUOLE ONLINE TVM CS	264
ATVO CANOVA+NAVIG AR	257
BIGLIETTO MOTO FINO 50 CC	244
CICLOMOTORE OLTRE 50CC	238
ARRIVA VENETO TRATTA 3	237
ARRIVA EXTRA TR.8-9-10 BORDO	236
ARRIVA VENETO TRATTA 2	206
EXTRA TRATTA 1 BORDO	205
ARRIVA EXTRA TR.1 BORDO	182
GRUPPI E SCUOLE ONLINE 2VIAGGI	180
ARRIVAEXTRA TR.2-3-4 BORDO	171
ARRIVA EXTRA TR. 5-6-7 BORDO	166
EXTRA TRATTA 5 TVM	160
BIGLIETTO MERCI C.SEMPLICE	159
FERRY17-AUTOBUS	134
SPIAGGEAR-TPL 11,75-COMVE1,25	125

NA-TRAGHETTO ORDINARIO	109
ATVO CANOVA+NAVIG AR ONLINE	102
ATVOCANOVA+NAVIG 1CORSA ONLINE	80
ARRIVA VENETO TRATTA 5	74
EXTRA TRATTE 5-6-7 BORDO	38
FERRY17-TRASPORTI PERICOLOSI	32
EXTRA TRATTA 8-9-10 TVM	23
ARRIVA AEROPORTO BORDO	22
PRENOTAZIONE VEICOLO ABBONATO	20
EXTRA TRATTA 7 TVM	18
ARRIVA INTEG.AEROP. BORDO	14
FERRY11-CARRI+35Q.RIM.	13
BIGL.URBANO CHIOGGIA BORDO	12
EXTRA TRATTA 6 TVM	9
TRAGHETTO GRATUITO	8
NA-12H-TPL 13,40-COMVE4,60	3
FERRY11-AUTOBUS	3
EXTRA TRATTE 8-9-10 BORDO	3
CAMP.MARINA-S.MARCO AR	2
SOSTITUTIVO PASS IMOB	1

Name: DESCRIZIONE_TITOLO, dtype: int64

```
In [ ]: # Populate the column 'TICKET_CODE' with the code of the ticket profile according to the ticket type and the ticket
df.loc[~df['TICKET_CODE'].isin(['1','2','3','4','5','5-STUD','5-WKRS','5-RET','6','6-STUD','6-WKRS','6-RET','7']),
```

```
In [ ]: # TICKET_CODE = 7b: Information about other tickets
print("The number of other tickets is: ", df[df['TICKET_CODE'] == '8'].shape[0])
```

The number of other tickets is: 376066

```
In [ ]: print("Information about the tickets with code 8 related to the serial number: ")
df[df['TICKET_CODE'] == '8'].groupby('DESCRIZIONE_TITOLO')['SERIALE'].value_counts().groupby('DESCRIZIONE_TITOLO').
```

Information about the tickets with code 8 related to the serial number:

Out[]:

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
AER+BOAT-TPL14,50-C.VE1,50	4038.0	1.657504	0.630205	1.0	1.00	2.0	2.00	5.0
AEROPORTO-VENEZIA AR	4205.0	1.550535	0.512106	1.0	1.00	2.0	2.00	3.0
AEROPORTO-VENEZIA AR ONLINE	1242.0	1.366345	0.490287	1.0	1.00	1.0	2.00	3.0
AEROPORTO-VENEZIA CS ONLINE	745.0	1.005369	0.073126	1.0	1.00	1.0	1.00	2.0
ARRIVA AEROPORTO BORDO	22.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
ARRIVA EXTRA TR. 5-6-7 BORDO	159.0	1.044025	0.343952	1.0	1.00	1.0	1.00	5.0
ARRIVA EXTRA TR.1 BORDO	172.0	1.058140	0.400428	1.0	1.00	1.0	1.00	5.0
ARRIVA EXTRA TR.8-9-10 BORDO	224.0	1.053571	0.278989	1.0	1.00	1.0	1.00	3.0
ARRIVA INTEG.AEROP. BORDO	14.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
ARRIVA VENETO AEROPORTO	294.0	1.013605	0.116044	1.0	1.00	1.0	1.00	2.0
ARRIVA VENETO TRATTA 1	818.0	1.332518	0.797452	1.0	1.00	1.0	1.00	8.0
ARRIVA VENETO TRATTA 2	186.0	1.107527	0.427750	1.0	1.00	1.0	1.00	4.0
ARRIVA VENETO TRATTA 3	199.0	1.190955	0.553883	1.0	1.00	1.0	1.00	6.0
ARRIVA VENETO TRATTA 4	490.0	1.183673	0.708507	1.0	1.00	1.0	1.00	9.0
ARRIVA VENETO TRATTA 5	63.0	1.174603	0.459301	1.0	1.00	1.0	1.00	3.0
ARRIVA VENETO TRATTA 6	409.0	1.244499	0.633134	1.0	1.00	1.0	1.00	8.0
ARRIVA VENETO TRATTA 7	199.0	1.402010	0.898481	1.0	1.00	1.0	2.00	9.0
ARRIVA VENETO TRATTA 8-9-10	4670.0	1.353961	1.009263	1.0	1.00	1.0	2.00	40.0
ARRIVAEXTRA TR.2-3-4 BORDO	160.0	1.068750	0.277497	1.0	1.00	1.0	1.00	3.0
ATVO CANOVA+NAVIG 1 CORSA	250.0	1.380000	0.793801	1.0	1.00	1.0	1.00	6.0
ATVO CANOVA+NAVIG AR	124.0	2.072581	1.176832	1.0	1.00	2.0	2.00	8.0
ATVO CANOVA+NAVIG AR ONLINE	56.0	1.821429	0.896603	1.0	1.00	2.0	2.00	5.0
ATVOCANOVA+NAVIG 1CORSIA ONLINE	64.0	1.250000	0.534522	1.0	1.00	1.0	1.00	3.0

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
BAGAGLIO CARTAVENEZIA	250.0	1.080000	0.271837	1.0	1.00	1.0	1.00	2.0
BIBIONE-S.MARCO AR	657.0	1.785388	0.610802	1.0	1.00	2.0	2.00	6.0
BICICLETTA "BIGLIETTERIA"	17398.0	1.151914	0.537253	1.0	1.00	1.0	1.00	18.0
BICICLETTA "CONCESSIONARI"	1972.0	1.085700	0.332965	1.0	1.00	1.0	1.00	7.0
BICICLETTA "PALMARE"	4282.0	1.005838	0.084895	1.0	1.00	1.0	1.00	3.0
BIGL AER-VENEZIA TSC	70645.0	1.012839	0.118102	1.0	1.00	1.0	1.00	4.0
BIGL.URBANO CHIOGGIA	2953.0	1.161531	0.433189	1.0	1.00	1.0	1.00	4.0
BIGL.URBANO CHIOGGIA BORDO	12.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
BIGLIETTO MERCI C.SEMPLICE	142.0	1.119718	0.346868	1.0	1.00	1.0	1.00	3.0
BIGLIETTO MOTO FINO 50 CC	241.0	1.012448	0.111105	1.0	1.00	1.0	1.00	2.0
BIGLIETTO SOC. SPORTIVE	63.0	4.349206	7.112367	1.0	1.00	1.0	2.50	30.0
BUS+PEOPLE MOVER ONLINE	1193.0	1.122381	0.532605	1.0	1.00	1.0	1.00	11.0
CAMP.MARINA-S.MARCO AR	1.0	2.000000	NaN	2.0	2.00	2.0	2.00	2.0
CAORLE-P.S.MARGH-S.MARCO AR	825.0	1.798788	0.697254	1.0	1.00	2.0	2.00	10.0
CARNET CHIOGGIA 10C. TICKET	1460.0	2.904795	2.213698	1.0	1.00	2.0	4.00	19.0
CAV-TREP - S.MARCO AR	7422.0	1.890865	0.654980	1.0	2.00	2.0	2.00	10.0
CICLOMOTORE FINO 50CC	2093.0	1.003822	0.061721	1.0	1.00	1.0	1.00	2.0
CICLOMOTORE OLTRE 50CC	238.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
ERACLEAMARE-S.MARCO AR	341.0	1.777126	0.692526	1.0	1.00	2.0	2.00	6.0
EV12-TPL 57,00-C.VE3,00	50.0	9.100000	7.611082	1.0	4.00	6.5	11.00	35.0
EV3-TPL 26,50-C.VE1,50	143.0	6.993007	4.793623	1.0	4.00	6.0	9.00	23.0
EV5-TPL 33,50-C.VE1,50	71.0	7.859155	5.270662	1.0	3.00	7.0	11.00	21.0
EV8-TPL 45,00-C.VE3,00	42.0	7.333333	5.435505	1.0	3.00	7.0	9.75	22.0

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
EXTRA TRATTA 1	4538.0	1.498457	1.208894	1.0	1.00	1.0	2.00	14.0
EXTRA TRATTA 1 BORDO	186.0	1.102151	0.337392	1.0	1.00	1.0	1.00	3.0
EXTRA TRATTA 1 TVM	711.0	1.126582	0.851352	1.0	1.00	1.0	1.00	21.0
EXTRA TRATTA 2	20687.0	1.600135	1.326412	1.0	1.00	1.0	2.00	34.0
EXTRA TRATTA 2 TVM	687.0	1.088792	0.414011	1.0	1.00	1.0	1.00	5.0
EXTRA TRATTA 3	14976.0	1.516426	1.116038	1.0	1.00	1.0	2.00	23.0
EXTRA TRATTA 3 TVM	364.0	1.093407	0.572136	1.0	1.00	1.0	1.00	9.0
EXTRA TRATTA 4	9197.0	1.487224	1.098644	1.0	1.00	1.0	2.00	27.0
EXTRA TRATTA 4 TVM	365.0	1.060274	0.280668	1.0	1.00	1.0	1.00	4.0
EXTRA TRATTA 5	3532.0	1.513873	1.162074	1.0	1.00	1.0	2.00	22.0
EXTRA TRATTA 5 TVM	132.0	1.212121	0.579949	1.0	1.00	1.0	1.00	5.0
EXTRA TRATTA 6	1253.0	1.448524	1.281451	1.0	1.00	1.0	2.00	33.0
EXTRA TRATTA 6 TVM	9.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
EXTRA TRATTA 7	367.0	1.400545	0.771982	1.0	1.00	1.0	2.00	7.0
EXTRA TRATTA 7 TVM	18.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
EXTRA TRATTA 8-9-10	583.0	1.313894	0.766073	1.0	1.00	1.0	1.00	10.0
EXTRA TRATTA 8-9-10 TVM	23.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
EXTRA TRATTE 2-3-4 BORDO	550.0	1.165455	0.508478	1.0	1.00	1.0	1.00	6.0
EXTRA TRATTE 5-6-7 BORDO	32.0	1.187500	0.535061	1.0	1.00	1.0	1.00	3.0
EXTRA TRATTE 8-9-10 BORDO	3.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
FERRY11-AUTOBUS	3.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
FERRY11-AUTOCARRI+35Q.	836.0	1.002392	0.048882	1.0	1.00	1.0	1.00	2.0
FERRY11-CARRI+35Q.RIM.	13.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
FERRY17-AUTOBUS	133.0	1.007519	0.086711	1.0	1.00	1.0	1.00	2.0
FERRY17-AUTOCARRI+35Q.	2962.0	1.025321	0.633176	1.0	1.00	1.0	1.00	29.0
FERRY17-CARRI+35Q.RIM.	654.0	1.021407	0.227175	1.0	1.00	1.0	1.00	4.0
FERRY17-TRASPORTI PERICOLOSI	32.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
GRUPPI E SCUOLE	478.0	7.156904	10.609046	1.0	1.00	2.0	10.00	51.0
GRUPPI E SCUOLE AR-SM	1123.0	1.698130	0.600554	1.0	1.00	2.0	2.00	4.0
GRUPPI E SCUOLE ONLINE 2VIAGGI	14.0	12.857143	9.882830	1.0	5.00	13.0	16.00	37.0
GRUPPI E SCUOLE ONLINE TVM AR	1781.0	1.738911	0.625126	1.0	1.00	2.0	2.00	4.0
GRUPPI E SCUOLE ONLINE TVM CS	206.0	1.281553	0.848729	1.0	1.00	1.0	1.00	8.0
GRUPPI ORGANIZZATI CS	123.0	2.463415	5.446149	1.0	1.00	1.0	1.00	38.0
JESOLO - S.MARCO AR	7354.0	1.852053	0.615436	1.0	2.00	2.0	2.00	8.0
L.11-AUTO "AEB" FINO A 4 METRI	1405.0	1.032028	0.231984	1.0	1.00	1.0	1.00	7.0
L.11-AUTO "C"DA 4,01 A 4,50 MT	1089.0	1.018365	0.153491	1.0	1.00	1.0	1.00	4.0
L.11-AUTO "D" OLTRE METRI 4,50	1804.0	1.022727	0.329766	1.0	1.00	1.0	1.00	11.0
L.17-AUTO "AEB" FINO A 4 METRI	8112.0	1.020587	0.212877	1.0	1.00	1.0	1.00	9.0
L.17-AUTO "C"DA 4,01 A 4,50 MT	6821.0	1.017593	0.164205	1.0	1.00	1.0	1.00	7.0
L.17-AUTO "D" OLTRE METRI 4,50	13814.0	1.019328	0.226372	1.0	1.00	1.0	1.00	17.0
LIGNANO-S.MARCO AR	159.0	1.723270	0.786854	1.0	1.00	2.0	2.00	7.0
NA-12H-TPL 13,40-COMVE4,60	2.0	1.500000	0.707107	1.0	1.25	1.5	1.75	2.0
NA-TRAGHETTO ORDINARIO	2.0	54.500000	75.660426	1.0	27.75	54.5	81.25	108.0
PRENOTAZ OCCASIONALE SI BARRA	10420.0	1.073608	0.401856	1.0	1.00	1.0	1.00	14.0
PRENOTAZIONE VEICOLO ABBONATO	20.0	1.000000	0.000000	1.0	1.00	1.0	1.00	1.0
SALONE NAUTICO 2022 1 CORSA	690.0	1.040580	0.211648	1.0	1.00	1.0	1.00	3.0

	count	mean	std	min	25%	50%	75%	max
DESCRIZIONE_TITOLO								
SALONE NAUTICO 2022 A/R	1894.0	1.539599	0.581694	1.0	1.00	2.0	2.00	5.0
SOSTITUTIVO PASS IMOB	1.0	1.000000	NaN	1.0	1.00	1.0	1.00	1.0
SPIAGGEAR-TPL 11,75-COMVE1,25	67.0	1.865672	0.385161	1.0	2.00	2.0	2.00	3.0
SU E ZO PER I PONTI 2022	1253.0	1.594573	0.674764	1.0	1.00	2.0	2.00	6.0
TARIFFA CAROZZINA	2333.0	1.102872	0.337298	1.0	1.00	1.0	1.00	4.0
TRAGH-TPL 4,41-C.VE0,59	10792.0	1.074778	0.381027	1.0	1.00	1.0	1.00	10.0
TRAGH-TPL 8,82-C.VE1,18	22955.0	1.670704	0.652067	1.0	1.00	2.0	2.00	19.0
TRAGHETTO GRATUITO	7.0	1.142857	0.377964	1.0	1.00	1.0	1.00	2.0

Summary of the ticket profiles

```
In [ ]: # Print the number of tickets for each ticket profile code ordered by the code of the ticket profile; print the name
df['TICKET_CODE'].value_counts().sort_index().rename(dict_tickets).reindex(dict_tickets.values(), fill_value=0)
```

```
Out[ ]: One-day ticket          1103633
Two-day ticket              682828
Three-day ticket           898863
Seven-day ticket           482235
Monthly ticket              1557
Monthly ticket for students      0
Monthly ticket for retired      0
Monthly ticket for workers      0
Annual ticket                0
Annual ticket for students      0
Annual ticket for retired      0
Annual ticket for workers      0
75 minutes ticket           1511459
Other ticket                 376066
Name: TICKET_CODE, dtype: int64
```

```
In [ ]: print("The number of tickets for each ticket profile code ordered by the code of the ticket profile, with a reference to the name of the serial number, is:")
df.groupby('TICKET_CODE')['SERIALE'].value_counts().groupby('TICKET_CODE').describe()
```

The number of tickets for each ticket profile code ordered by the code of the ticket profile, with a reference to the name of the serial number, is:

```
Out [ ]:
```

	count	mean	std	min	25%	50%	75%	max
TICKET_CODE								
1	297767.0	3.706364	2.137101	1.0	2.0	3.0	5.0	119.0
2	107261.0	6.366042	3.527327	1.0	4.0	6.0	8.0	68.0
3	109037.0	8.243651	4.567479	1.0	5.0	8.0	11.0	113.0
4	35170.0	13.711544	7.863425	1.0	8.0	13.0	18.0	121.0
5	228.0	6.828947	11.691770	1.0	1.0	1.0	4.0	58.0
7	1200462.0	1.259064	0.641590	1.0	1.0	1.0	1.0	26.0
8	288857.0	1.301911	0.971396	1.0	1.0	1.0	1.0	108.0

```
In [ ]: # Countplot of the column 'TICKET_CODE'
fig, ax = plt.subplots(figsize=(15,8))
# Countplot of the column 'TICKET_CODE'
sns.countplot(x='TICKET_CODE', data=df, order=df['TICKET_CODE'].value_counts().sort_index().index)
plt.title('Countplot of the column TICKET_CODE', fontsize=20)
plt.xlabel('Ticket code', fontsize=15)
plt.ylabel('Count (in millions)', fontsize=15)

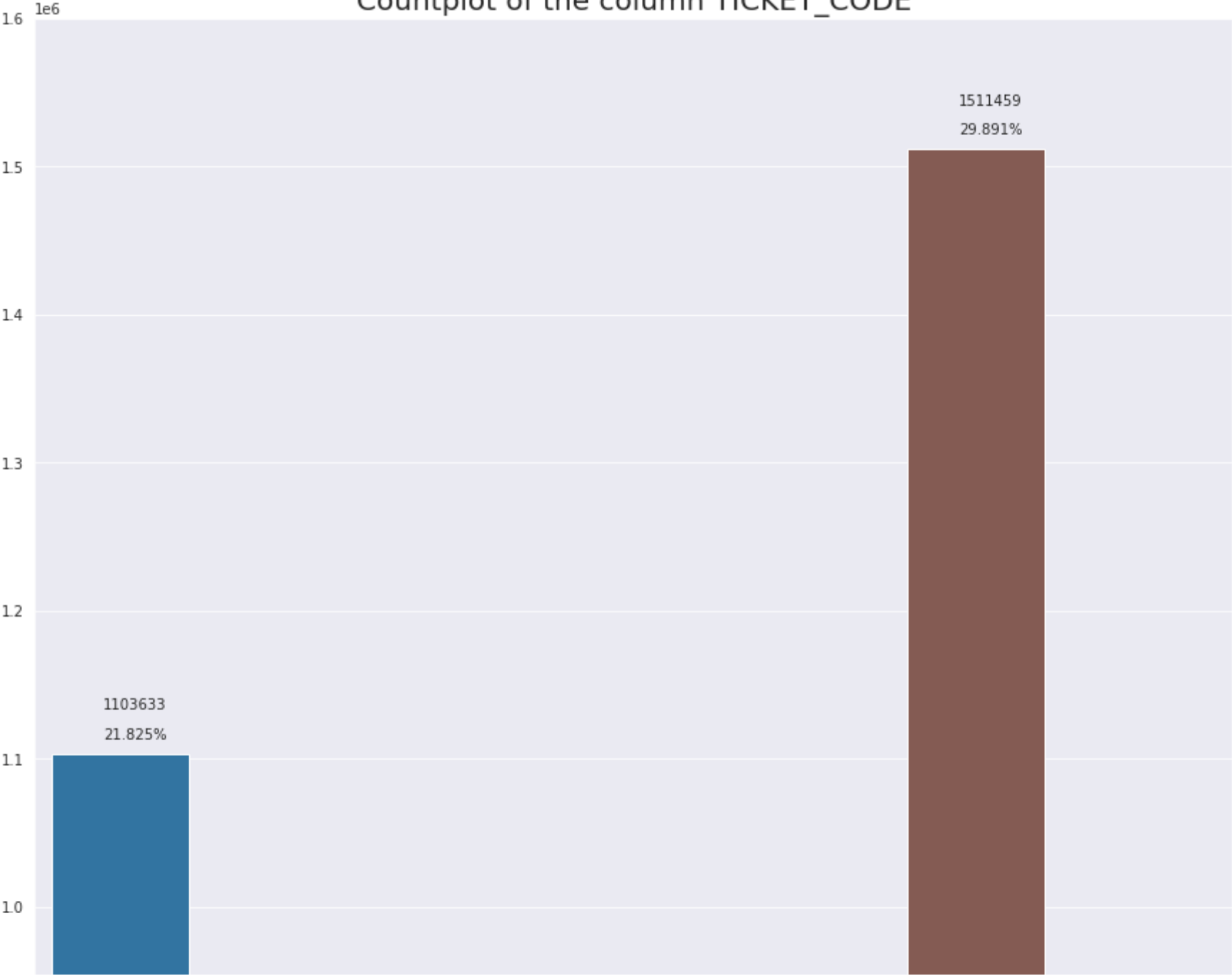
# Change yticks to have a better visualization
scale = np.arange(0, max(df['TICKET_CODE'].value_counts())+100000, 100000)
plt.yticks(scale)

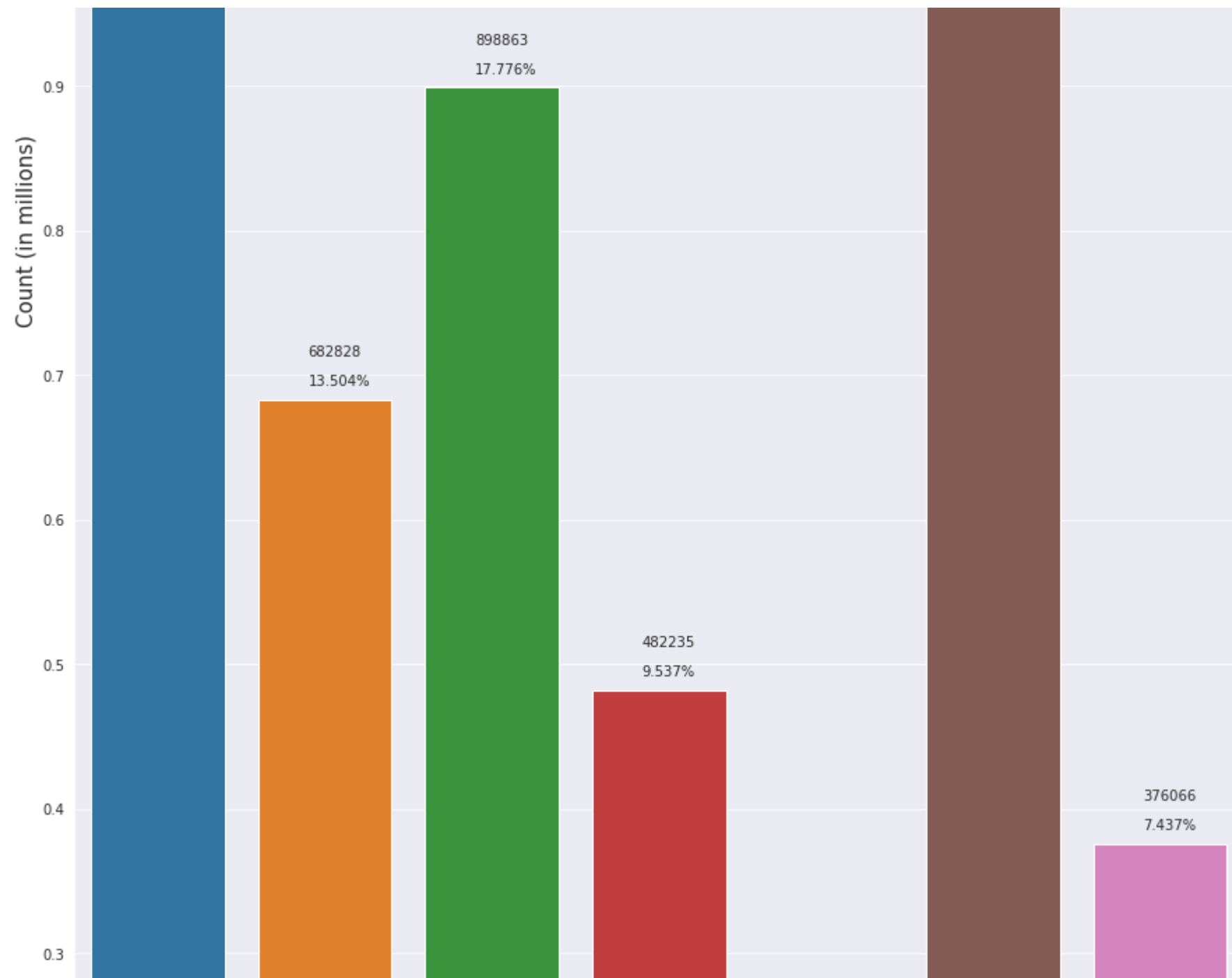
# Add the percentage of each category on top of the bars
for p in ax.patches:
    ax.annotate('{:.3f}%'.format(100*p.get_height()/len(df)), (p.get_x()+0.3, p.get_height()+10000))

# Add the count of each category on top of the bars
for p in ax.patches:
    ax.annotate('{:.0f}'.format(p.get_height()), (p.get_x()+0.3, p.get_height()+30000))
```

```
# Add a padding on the top of the plot  
plt.subplots_adjust(top=3)
```


Countplot of the column TICKET_CODE





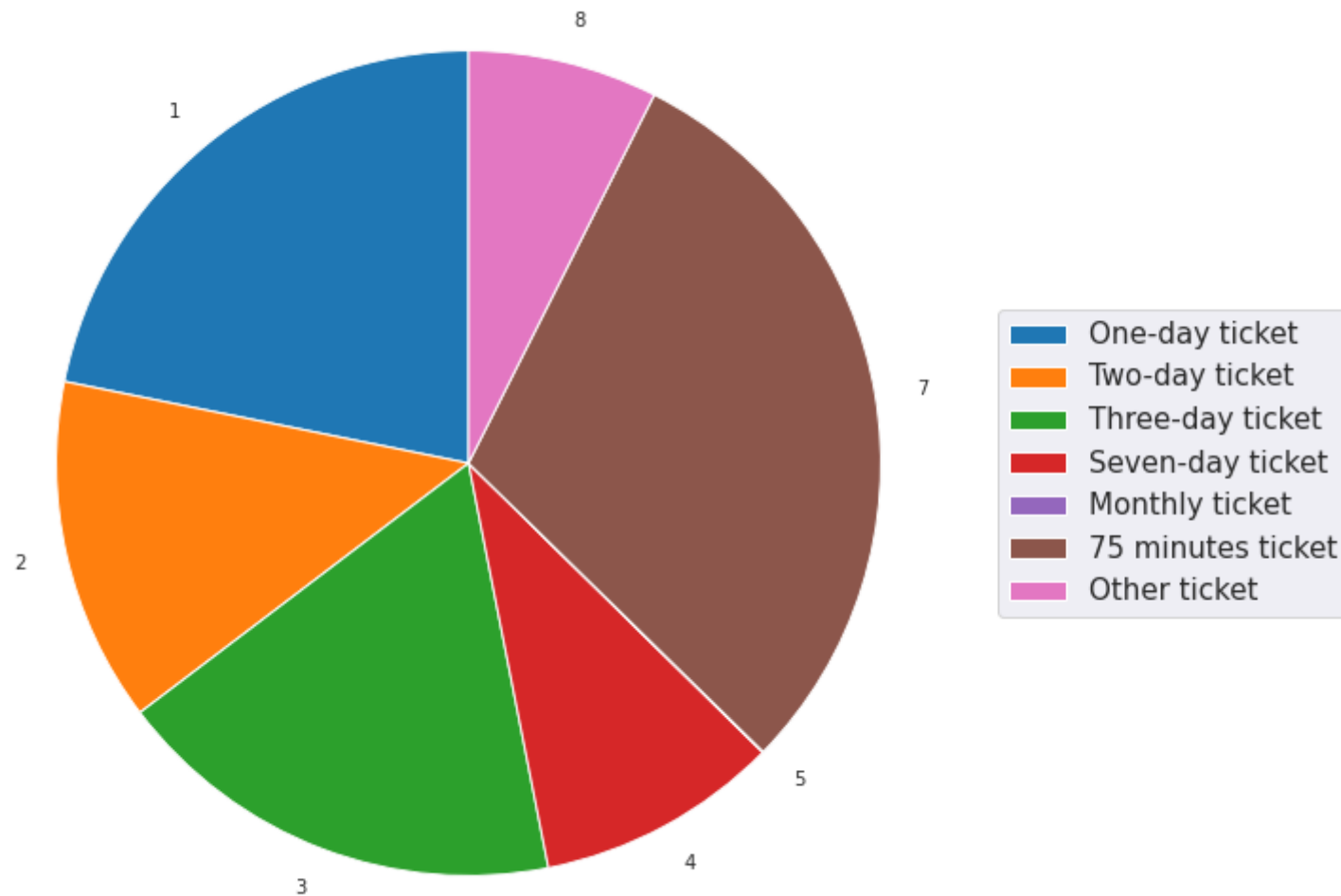


```
In [ ]: # Plot a pie chart of the column 'TICKET_CODE'
fig, ax = plt.subplots(figsize=(20,10))
df['TICKET_CODE'].value_counts().sort_index().plot.pie(startangle=90)

# Add the name of the ticket profile on the pie chart
plt.legend(labels=df['TICKET_CODE'].value_counts().sort_index().rename(dict_tickets).index, loc='center left', bbox

plt.title('Pie chart of the column TICKET_CODE', fontsize=20)
plt.ylabel('')
fig.patch.set_facecolor('white')
plt.show()
```

Pie chart of the column TICKET_CODE



Delete the validation that are with TICKET_CODE = 8 (other tickets)

```
In [ ]: shape_before = df.shape[0]

# Delete 8 tickets because they are not useful for the analysis
```

```
df = df[df['TICKET_CODE'] != '8']

# Print the number of rows before and after the deletion of the 8 tickets and the difference
print('The number of rows before the deletion of the 8 tickets is: {}'.format(shape_before))
print('The number of rows after the deletion of the 8 tickets is: {}'.format(df.shape[0]))
print('The difference is: {}'.format(shape_before - df.shape[0]))

# TODO: to de-comment later
```

The number of rows before the deletion of the 8 tickets is: 5056641
The number of rows after the deletion of the 8 tickets is: 4680575
The difference is: 376066

Data Cleaning

Stops similar

```
In [ ]: # Print the number of unique values in the column 'DESCRIZIONE' that are the names of the stops
print('The number of unique values in the column DESCRIZIONE is: {}'.format(df['DESCRIZIONE'].nunique()))
```

The number of unique values in the column DESCRIZIONE is: 600

```
In [ ]: def get_common_prefix(string_list):
        """
        This function returns the common prefix of a list of strings.
        If there is no common prefix, it returns an empty string.
        :param string_list: list of strings
        :return: string that is the common prefix of the list of strings
        """
        first_prefix = string_list[0].split(" ")[0]
        # Create and empty dictionary
        prefix_dict = {}

        # Iterate over the list of strings
        for string in string_list[1:]:
            # Check if the string starts with the first prefix
            if not string.startswith(first_prefix):
                # If the string does not start with the first prefix, split the string and take the first word
                first_prefix = string.split(" ")[0]
```

```

        if string.startswith(first_prefix):
            # In the dictionary add the new prefix as key and the list of strings that have this prefix as value
            prefix_dict[first_prefix] = [string for string in string_list if string.startswith(first_prefix)]
        else:
            # In the dictionary add the new prefix as key and the list of strings that have this prefix as value
            prefix_dict[first_prefix] = [string for string in string_list if string.startswith(first_prefix)]
    return prefix_dict

```

```

In [ ]: # To avoid problem regarding the letters in uppercase and lowercase, convert all the letters in uppercase
df['DESCRIZIONE'] = df['DESCRIZIONE'].str.upper()

```

```

In [ ]: # Use the function get_common_prefix to find the common prefix of the strings in the column 'DESCRIZIONE' and print

# Create a string list with the unique values of the column 'DESCRIZIONE'
string_list = df['DESCRIZIONE'].unique().tolist()

dict_prefix = get_common_prefix(string_list)
for key, value in dict_prefix.items():
    print('{}: {}'.format(key, value))

# Print the number of keys in the dictionary
print('The number of keys in the dictionary is: {}'.format(len(dict_prefix.keys())))

```

FERROVIA: ['FERROVIA "D"', 'FERROVIA "B"', 'FERROVIA "E"', 'FERROVIA "C"', 'FERROVIA "A"']
P.LE: ['P.LE ROMA "G"', 'P.LE ROMA "E"', 'P.LE ROMA "B"', 'P.LE ROMA "F"', 'P.LE ROMA "C"', 'P.LE ROMA "D"', 'P.LE ROMA "A"']
VENEZIA: ['VENEZIA', 'VENEZIA CORS']
GIUDECCA: ['GIUDECCA PAL']
S.: ['S. MARCUOLA-', 'S.ERASMO CHI', 'S.ERASMO CAP', 'S. ERASMO PU', 'S. MARCO (GI', 'S. PIETRO IN']
SAN: ['SAN MARCO-SA', 'SAN MARCO VA', 'SAN STAE', 'SAN MARCO MA', 'SANTA MARIA ', 'SAN TOMA"', 'SAN ROCCO BR', 'SAN GIORGIO', 'SANT' ELENA', 'SANTA MARTA', 'SAN DONA' RI', 'SAN DONA' PA', 'SAN MARCO CA', 'SANT' ANGELO', 'SANT' ANTONIO', 'SAN DONA' PI', 'SAN SAMUELE', 'SAN DONA' MA', 'SAN MARCO MO', 'SAN SILVESTR', 'SAN MARCO BO', 'SAN MARCO FO', 'SAN MARCO MU', 'SAN MARCO SA', 'SAN NICOLO"', 'SAN DONA' CE', 'SANT' ALVISE', 'SAN SERVULO', 'SAN BASILIO', 'SAN DONA' VA', 'SAN DONA' FA', 'SAN PIETRO D', 'SAN NICOLO' ', 'SAN LAZZARO', 'SAN LIBERALE', 'SAN PIETRO P', 'SAN PIETRO CA', 'SAN DONA' PE', 'SAN PIETRO C', 'SAN PIETRO B', 'SANSOVINO VE', 'SAN TROVASO']
PIAVE: ['PIAVE FIUME', 'PIAVE PUCCIN', 'PIAVE PODGOR']
F.TE: ['F.TE NOVE "C"', 'F.TE NOVE "A"', 'F.TE NOVE "D"', 'F.TE NOVE']
AEROPORTO: ['AEROPORTO MA']
RIALTO: ['RIALTO', 'RIALTO MERCA', 'RIALTO "A"']
PELLESTRINA: ['PELLESTRINA', 'PELLESTRINA ']
ACCADEMIA: ['ACCADEMIA "B"', 'ACCADEMIA "A"']
LIDO: ['LIDO S.M.E. ', 'LIDO SAN NIC']
MESTRE: ['MESTRE CENTR']
SPINEA: ['SPINEA MARTI', 'SPINEA GIORG', 'SPINEA ALFIE', 'SPINEA PIAZZ', 'SPINEA POZZU', 'SPINEA ROSSI', 'SPINEA D ESEN', 'SPINEA VILLA', 'SPINEA SANRE', 'SPINEA ORGNA', 'SPINEA REPUB']
ZATTERE: ['ZATTERE']
GIGLIO: ['GIGLIO']
LIBERTA': ["LIBERTA' SAN", "LIBERTA' RIG", "LIBERTA' FIN", "LIBERTA' POR", "LIBERTA' COM"]
DON: ['DON STURZO V', 'DON STURZO P']
SANTA: ['SANTA MARIA ', 'SANTA MARTA']
CA': ["CA' D'ORO", "CA' REZZONIC", "CA' ROSSA OB", "CA' ROSSA VO", "CA' ROSSA SE", "CA' SABBIONI", "CA' ROSSA BI", "CA' MARCELLO", "CA' LIN GATT", "CA' LIN ERAC", "CA' BIANCA L", "CA' LIN CAST", "CA' SOLARO C"]
TRONCHETTO: ['TRONCHETTO M', 'TRONCHETTO F', 'TRONCHETTO', 'TRONCHETTO T', 'TRONCHETTO V']
ZITELLE: ['ZITELLE']
GIARDINI: ['GIARDINI "B"', 'GIARDINI BIE', 'GIARDINI "A"']
SANT': ["SANT' ELENA", "SANT' ANGELO", "SANT'ANTONIO", "SANT' ALVISE"]
ARSENALE: ['ARSENALE "B"', 'ARSENALE "A"']
GALLO: ['GALLO BARBAR', 'GALLO GIOLIT', 'GALLO LOREDA', 'GALLO TORTA', "GALLO SANT'A", 'GALLO MOROSI', 'GALLO BRAGAD', 'GALLO CONTAR', 'GALLO DEI KI', 'GALLO DEI GI', 'GALLO BIBLIO', 'GALLO GALOPP']
ORTO: ['ORTO']
BANDIERA: ['BANDIERA GHE', 'BANDIERA CRU', 'BANDIERA TOM', 'BANDIERA CAN', 'BANDIERA DUR', 'BANDIERA PIL']
GARIBALDI: ['GARIBALDI MU', 'GARIBALDI OG', 'GARIBALDI GI', 'GARIBALDI C']
PUNTA: ['PUNTA SABBIO']

BISSUOLA: ['BISSUOLA COL', 'BISSUOLA CAD', 'BISSUOLA ESI', 'BISSUOLA PIS', 'BISSUOLA TEV', 'BISSUOLA VAR']
CANAL: ['CANAL LEONE']
LAVELLI: ['LAVELLI PAOL']
RIZZARDI: ['RIZZARDI CAR']
STAZIONE: ['STAZIONE MES', 'STAZIONE PAD', 'STAZIONE MAR']
CARDUCCI: ['CARDUCCI PAS', 'CARDUCCI FEL']
PERON: ['PERON SARAGA', 'PERON BASEGG']
REDETORE: ['REDETORE']
ALTINIA: ['ALTINIA INDR', 'ALTINIA CA', 'ALTINIA SAN', 'ALTINIA MUNI', 'ALTINIA BERI', 'ALTINIA FAVA', 'ALTINIA 181', 'ALTINIA FORT']
BELFREDO: ['BELFREDO TER']
ORLANDA: ['ORLANDA APPI', 'ORLANDA BAGA', 'ORLANDA PINE', 'ORLANDA SABB', 'ORLANDA SAN', 'ORLANDA CAVE', 'ORLANDA PIOV', 'ORLANDA CASI', 'ORLANDA 200', 'ORLANDA TIBU', 'ORLANDA CENT', 'ORLANDA DON']
MIRANESE: ['MIRANESE SAN', 'MIRANESE PIE', 'MIRANESE SEL', 'MIRANESE GIU', 'MIRANESE CAL', 'MIRANESE IVA', 'MIRANESE PER', 'MIRANESE LAZ', 'MIRANESE PIA', 'MIRANESE MON', 'MIRANESE VIV', 'MIRANESE AVA']
TRIESTE: ['TRIESTE CATE', 'TRIESTE ROBI', 'TRIESTE ERAC', 'TRIESTE BOSC', 'TRIESTE MIRA', 'TRIESTE MAZZ', 'TRIESTE PARC']
27: ['27 OTTOBRE D']
FISICA: ['FISICA DEPOS']
TREPORTI: ['TREPORTI']
CORSO: ['CORSO DEL PO']
RAMPA: ['RAMPA CAVALC']
CELESTIA: ['CELESTIA']
SALUTE: ['SALUTE']
TORCELLO: ['TORCELLO']
SABBADINO: ['SABBADINO L', 'SABBADINO BA', 'SABBADINO PA', 'SABBADINO LA']
CASONA: ['CASONA MARZI', 'CASONA BISSU', 'CASONA VALLE']
TRIESTINA: ['TRIESTINA MO', 'TRIESTINA TE', 'TRIESTINA ZU', 'TRIESTINA LA', 'TRIESTINA UL', 'TRIESTINA AL', 'TRIESTINA PR', 'TRIESTINA SC', 'TRIESTINA AE', 'TRIESTINA PA', 'TRIESTINA FO', 'TRIESTINA ZO', 'TRIESTINA TO', 'TRIESTINA PI']
ALBERONI: ['ALBERONI CA', 'ALBERONI DEL', 'ALBERONI FAR', 'ALBERONI STE', 'ALBERONI COL', 'ALBERONI GOL', 'ALBERONI SAN', 'ALBERONI OTT', 'ALBERONI SPI']
PADANA: ['PADANA CIVIC', 'PADANA AVENA', 'PADANA STAZI', 'PADANA COLOM', 'PADANA DELLE', 'PADANA DEL L']
MARTELLAGO: ['MARTELLAGO', 'MARTELLAGO V', 'MARTELLAGO F', 'MARTELLAGO D', 'MARTELLAGOTR']
MARGHERA: ['MARGHERA ARD', 'MARGHERA CIM', 'MARGHERA SAL']
PEOPLE: ['PEOPLE MOVER']
CAPPUCCINA: ['CAPPUCCINA V', 'CAPPUCCINA S', 'CAPPUCCINA B']
GUGLIE: ['GUGLIE', 'GUGLIE B']
TERRAGLIO: ['TERRAGLIO VI', 'TERRAGLIO PE', 'TERRAGLIO TE', 'TERRAGLIO GA', 'TERRAGLIO BO', 'TERRAGLIO CA', 'TERRAGLIO NI', 'TERRAGLIO FA']

PALEOCAPA: ['PALEOCAPA PA']
CHIOGGIA: ['CHIOGGIA PIA', 'CHIOGGIA SAN', 'CHIOGGIA CAM', 'CHIOGGIA ZAR', 'CHIOGGIA DE', 'CHIOGGIA DEN']
S.ERASMO: ['S.ERASMO CHI', 'S.ERASMO CAP']
SACCA: ['SACCA FISOLA']
PASQUALIGO: ['PASQUALIGO M']
RIVA: ['RIVA DE BIAS']
TOSATTO: ['TOSATTO IMPA', 'TOSATTO PACC']
PERTINI: ['PERTINI CHIE', 'PERTINI FOSC', 'PERTINI VESP', 'PERTINI CARR', 'PERTINI GAVA']
CAFASSO: ['CAFASSO BOTT']
CASTELLANA: ['CASTELLANA S', 'CASTELLANA C', 'CASTELLANA P', 'CASTELLANA B', 'CASTELLANA M']
PIAZZA: ['PIAZZA MERCA', 'PIAZZALE GIO', 'PIAZZALE RAV']
CAVALCAVIA: ['CAVALCAVIA V']
CALABRIA: ['CALABRIA CAM']
MONTE: ['MONTE CELO F', 'MONTE CERVIN']
TRE: ['TREPORTI', 'TRE ARCHI', 'TRENTO PODGO', 'TREZZO TERRA', 'TREVISO', 'TREZZO BATTU', 'TRENTO FAGAR', 'TREVISO
SAN ', 'TREVISO FS', 'TRENTO GAZZE']
PARK: ['PARK PETROLI']
ILARIA: ['ILARIA ALPI ']
MALCONTENTA: ['MALCONTENTA ']
FORTE: ['FORTE MARGHE']
MURANO: ['MURANO VENIE', 'MURANO NAVAG', 'MURANO FARO', 'MURANO MUSEO', 'MURANO COLON', 'MURANO DA MU', 'MURANO SE
REN']
QUARNARO: ['QUARNARO CAL']
MOGLIANO: ['MOGLIANO BUR', 'MOGLIANO BEL', 'MOGLIANO MAR', 'MOGLIANO FS', 'MOGLIANO MUN', 'MOGLIANO CEN']
DESE: ['DESE CENTRO', 'DESE CICOGNE', 'DESE FS', 'DESE LITOMAR']
SPIRITO: ['SPIRITO SANT']
OLIVI: ['OLIVI']
GOBBI: ['GOBBI ORLAND', 'GOBBI SAN DO', 'GOBBI VALLEN', 'GOBBI MIRTIL', 'GOBBI DON BO', "GOBBI CA' DO", 'GOBBI MAN
DAR']
GAZZERA: ['GAZZERA ALTA']
MALAMOCCO: ['MALAMOCCO PA', 'MALAMOCCO AL', 'MALAMOCCO ST', 'MALAMOCCO CE', 'MALAMOCCO BA', 'MALAMOCCO GA', 'MALAM
OCCO BE', 'MALAMOCCO OC', 'MALAMOCCO SC']
RINASCITA: ['RINASCITA BE', 'RINASCITA EM']
FAVRETTI: ['FAVRETTI MES']
PIAZZALE: ['PIAZZALE GIO', 'PIAZZALE RAV']
PAGANELLO: ['PAGANELLO TI']
ASSEGGIANO: ['ASSEGGIANO V', 'ASSEGGIANO M', 'ASSEGGIANO C', 'ASSEGGIANO E', 'ASSEGGIANO P', 'ASSEGGIANO D']
OSPEDALE: ['OSPEDALE', 'OSPEDALE DEL', 'OSPEDALE MIR']
BURANO: ['BURANO "C"', 'BURANO "B"', 'BURANO "A"']
CIRCONVALLAZ: ['CIRCONVALLAZ']

ANCONA: ['ANCONA TORIN', 'ANCONA CARBO']
CORRENTI: ['CORRENTI CAP']
TRIVIGNANO: ['TRIVIGNANO L', 'TRIVIGNANO B', 'TRIVIGNANO C', 'TRIVIGNANO P', 'TRIVIGNANO G']
PASSO: ['PASSO CAMPAL']
ZELARINO: ['ZELARINO CHI', 'ZELARINO CAR', 'ZELARINO MUN', 'ZELARINO PAR']
CIRCONVALAZI: ['CIRCONVALAZI']
MATTUGLIE: ['MATTUGLIE DI', 'MATTUGLIE PE']
BECCARIA: ['BECCARIA CAN', 'BECCARIA CHI', 'BECCARIA COR', 'BECCARIA ORO', 'BECCARIA PAR']
MARCONI: ['MARCONI LORE', 'MARCONI MARC', 'MARCONI FIUM', 'MARCONI DES ', 'MARCONI CORA']
MARCON: ['MARCONI LORE', 'MARCONI MARC', 'MARCON COOPE', 'MARCONI FIUM', 'MARCON CENTR', 'MARCONI DES ', 'MARCONI CORA', 'MARCON LOMAB', 'MARCON ALTIN', 'MARCON CULT', 'MARCON MUNIC', 'MARCON VITTO']
VESPUCCI: ['VESPUCCI GRI', 'VESPUCCI BOE', 'VESPUCCI SAN', "VESPUCCI CA'", 'VESPUCCI CAT', 'VESPUCCI PIG', 'VESPUCCI GAR']
TORINO: ['TORINO UNIVE', 'TORINO ROSSE', 'TORINO']
SALAMONIO: ['SALAMONIO MA']
CIMITERO: ['CIMITERO']
DURANDO: ['DURANDO BELL']
CREA: ['CREA']
CAPOLINEA: ['CAPOLINEA CA', 'CAPOLINEA FU']
RIELTA: ["RIELTA CA' R", 'RIELTA PARCO']
MAZZORBO: ['MAZZORBO']
GALILEI: ['GALILEI DARS']
CERTOSA: ['CERTOSA', 'CERTOSA (A ']
FARO: ['FARO ROCCHET']
D'ANNUNZIO: ["D'ANNUNZIO D", "D'ANNUNZIO P", "D'ANNUNZIO S"]
CAVERGNAGO: ['CAVERGNAGO M', 'CAVERGNAGO T']
MIRANO: ['MIRANO SCUOL', 'MIRANO MATTE', 'MIRANO CENTR', 'MIRANO CARDU', 'MIRANO GRAMS', 'MIRANO PESTR', 'MIRANO T REVI', 'MIRANO SPORT', 'MIRANO PERUG', 'MIRANO GRIMA', 'MIRANO DANTE', 'MIRANO BATTI', 'MIRANO FOSSA']
TRENTO: ['TRENTO PODGO', 'TRENTO FAGAR', 'TRENTO GAZZE']
PALAZZO: ['PALAZZO DEL ']
VIGNOLE: ['VIGNOLE']
SELVANESE: ['SELVANESE PL']
AGENZIA: ['AGENZIA ENT']
MORANDI: ['MORANDI NICE']
TEVERE: ['TEVERE BAGLI', 'TEVERE PARCO']
BACINI: ['BACINI - ARS']
EINAUDI: ['EINAUDI CAST']
PALIAGA: ["PALIAGA CA' "]
NOALE: ['NOALE', 'NOALE OSPEDA', 'NOALE LANCER']
MARSALA: ['MARSALA CENT']

DEI: ['DEI MURAZZI ']
MADONNA: ['MADONNA DELL']
CALUCCI: ['CALUCCI QUAR']
CAMPORESE: ['CAMPORESE GR']
DOLO: ['DOLO MANZONI', 'DOLO CENTRO', "DOLO CA' TRO", 'DOLO BANDIER', 'DOLO DORIA', 'DOLO CIVICO ']
PAOLUCCI: ['PAOLUCCI LON']
COLOMBO: ['COLOMBO']
CHIMICA: ['CHIMICA INGR', 'CHIMICA MECC']
GIOVANNACCI: ['GIOVANNACCI ']
PREGANZIOL: ['PREGANZIOL F', 'PREGANZIOL G', 'PREGANZIOL ', 'PREGANZIOL M']
PASINI: ['PASINI FRATE', 'PASINI LAVOR']
TREZZO: ['TREZZO TERRA', 'TREZZO BATTU']
VIA: ['VIA VILLABON', 'VIA DEI CANT']
TITO: ['TITO CASTELL']
GOZZI: ['GOZZI CAPPUC']
CALVI: ['CALVI PARMES']
GATTA: ['GATTA SANTA ', 'GATTA SCARAN', 'GATTA Verci', 'GATTA 90', 'GATTA PROTAG', 'GATTA SCARAM']
SALICI: ['SALICI VILLA']
FAVIGNANA: ['FAVIGNANA']
GRAN: ['GRAN VIALE']
BOTTENIGO: ['BOTTENIGO PI', 'BOTTENIGO CA', 'BOTTENIGO BO', 'BOTTENIGO MA']
PORTOSECCO: ['PORTOSECCO', 'PORTOSECCO L', 'PORTOSECCO C']
RISORGIMENTO: ['RISORGIMENTO']
SANPIETRO: ['SANPIETRO CA']
LAZZARETTO: ['LAZZARETTO N']
CAROMAN: ['CAROMAN']
TREVISO: ['TREVISO', 'TREVISO SAN ', 'TREVISO FS']
BRENDOLE: ['BRENDOLE STI', 'BRENDOLE FAV', 'BRENDOLE ARS', 'BRENDOLE']
OSPIZIO: ['OSPIZIO MARI']
ROBEGANO: ['ROBEGANO MON', 'ROBEGANO 25 ']
CAVANIS: ['CAVANIS CAPO']
AZOTO: ['AZOTO SOTTAN']
MARTIRI: ['MARTIRI DELL']
VILLABONA: ['VILLABONA PI', 'VILLABONA BO', 'VILLABONA 87', 'VILLABONA MO', 'VILLABONA 8']
ZERO: ['ZERO BRANCO']
VALLENARI: ['VALLENARI ST']
SALZANO: ['SALZANO TOSC']
VISINONI: ['VISINONI COM', 'VISINONI POL']
SCORZE': ["SCORZE' MOGL", "SCORZE' CAPO", "SCORZE' SAN ", "SCORZE' MUNI", "SCORZE' ROMA", "SCORZE' FERM", "SCORZE' CECC"]

SCUOLA: ['SCUOLA MEDIA', 'SCUOLA ZENDR']
MORANZANI: ['MORANZANI EL', 'MORANZANI 32', 'MORANZANI CO']
GAGGIO: ['GAGGIO FERMI']
SOTTOMARINA: ['SOTTOMARINA ', 'SOTTOMARINA']
FIESSO: ["FIESSO D'ART", 'FIESSO GEMIT']
OLMO: ['OLMO VITTORI', 'OLMO PELLICO', 'OLMO PAPA LU', 'OLMO CALVI']
DE: ['DESE CENTRO', 'DEI MURAZZI ', 'DESE CICOONE', 'DESE FS', 'DESE LITOMAR', 'DE NICOLA CH']
MAERNE: ['MAERNE FS', 'MAERNE CENTR', 'MAERNE CHIES', 'MAERNE CIMIT']
SANSOVINO: ['SANSOVINO VE']
ZENDRINI: ['ZENDRINI VIL']
PESEGGIA: ['PESEGGIA VI']
CORTIVO: ['CORTIVO TOMB']
MIRA: ['MIRANESE SAN', 'MIRANESE PIE', 'MIRANESE SEL', 'MIRANESE GIU', 'MIRANESE CAL', 'MIRANESE IVA', 'MIRANESE P
ER', 'MIRANO SCUOL', 'MIRANESE LAZ', 'MIRANESE PIA', 'MIRANO MATTE', 'MIRANESE MON', 'MIRANESE VIV', 'MIRANESE AV
A', 'MIRANO CENTR', 'MIRANO CARDU', 'MIRANO GRAMS', 'MIRANO PESTR', 'MIRANO TREVI', 'MIRANO SPORT', 'MIRA RISCOS
S', 'MIRANO PERUG', 'MIRANO GRIMA', 'MIRANO DANTE', 'MIRA PORTE', 'MIRA ALIGHIE', 'MIRANO BATTI', 'MIRANO FOSSA',
'MIRA ALBRIZZ', 'MIRA CENTRO']
ROMEA: ['ROMEA MALCAN']
ORIAGO: ['ORIAGO STAZI', 'ORIAGO SOMMO', 'ORIAGO ROMAG', 'ORIAGO CENTR']
ISOLA: ['ISOLA UNIONE']
MARTELLAGOTR: ['MARTELLAGOTR']
STRA: ['STRA PIAZZA ']
CAMPOCROCE: ['CAMPOCROCE']
FERM.SERV.: ['FERM.SERV. D']
KLINGER: ['KLINGER SAN ', 'KLINGER RAVA', 'KLINGER PRES']
AREOPORTO: ['AREOPORTO MA']
PADOVA: ['PADOVA FIERA', 'PADOVA SAN L']
VALLON: ['VALLON BORGO']
SFMR: ['SFMR SPINEA']
CIVE': ["CIVE' PALTAN", "CIVE' MIRA"]
ROSOLINA: ['ROSOLINA']
ROSARA: ['ROSARA']
RIO: ['RIO SAN MART']
FRESCADA: ['FRESCADA']
APPRODO: ['APPRODO NAVE']
CALTANA: ['CALTANA']
PIOVE: ['PIOVE DI SAC']
INCR.: ['INCR. VIA DA', 'INCR. VIA GR', 'INCR. VIA VE']
BOJON: ['BOJON']
TESSERA: ['TESSERA SCUO']

```
REBOSOLA: ['REBOSOLA CIV']
CAMPAGNA: ['CAMPAGNA LUP']
VIGONZA: ['VIGONZA PERA']
CAPRICCIO: ['CAPRICCIO CE']
The number of keys in the dictionary is: 215
```

Update some keys in the dictionary

```
In [ ]: # Rename the key 'P.le' with 'P.le Roma'
dict_prefix['P.LE ROMA'] = dict_prefix.pop('P.LE')
# Rename the key 'F.TE' with 'F.TE NOVE'
dict_prefix['F.TE NOVE'] = dict_prefix.pop('F.TE')
```

```
In [ ]: # Print the values of the dictionary with the keys 'S.' and 'San'
print('The values of the dictionary with the key S. are: {}'.format(dict_prefix['S.']))
print('The values of the dictionary with the key San are: {}'.format(dict_prefix['SAN']))
```

The values of the dictionary with the key S. are: ['S. MARCUOLA-', 'S.ERASMO CHI', 'S.ERASMO CAP', 'S. ERASMO PU', 'S. MARCO (GI', 'S. PIETRO IN']

The values of the dictionary with the key San are: ['SAN MARCO-SA', 'SAN MARCO VA', 'SAN STAE', 'SAN MARCO MA', 'SANTA MARIA ', 'SAN TOMA'', 'SAN ROCCO BR', 'SAN GIORGIO', 'SANT' ELENA', 'SANTA MARTA', 'SAN DONA' RI', 'SAN DONA' PA', 'SAN MARCO CA', 'SANT' ANGELO', 'SANT'ANTONIO', 'SAN DONA' PI', 'SAN SAMUELE', 'SAN DONA' MA', 'SAN MARCO M O', 'SAN SILVESTR', 'SAN MARCO BO', 'SAN MARCO FO', 'SAN MARCO MU', 'SAN MARCO SA', 'SAN NICOLO'', 'SAN DONA' CE', 'SANT' ALVISE', 'SAN SERVULO', 'SAN BASILIO', 'SAN DONA' VA', 'SAN DONA' FA', 'SAN PIETRO D', 'SAN NICOLO' ', 'SAN LAZZARO', 'SAN LIBERALE', 'SAN PIETRO P', 'SANPIETRO CA', 'SAN DONA' PE', 'SAN PIETRO C', 'SAN PIETRO B', 'SANSOVI NO VE', 'SAN TROVASO']

S.Erasmo

```
In [ ]: # Create a new key in the dictionary with the key S.ERASMO; insert as value the list of strings that have the prefix
dict_prefix['S.ERASMO'] = [string for string in dict_prefix['S.'] if string.startswith('S.ERASMO')]

# Add the value 'S. Erasmo Pu' originally in the key 'San' to the key 'S.ERASMO'
dict_prefix['S.ERASMO'].append('S. ERASMO PU')

# Remove the strings that have the prefix 'S.ERASMO' from the keys 'S.' and 'San'
dict_prefix['S.'] = [string for string in dict_prefix['S.'] if not string.startswith('S.ERASMO')]
dict_prefix['S.'] = [string for string in dict_prefix['S.'] if not string.startswith('S. ERASMO PU')]
```

```
# Print the values of the dictionary with the key 'S.ERASMO'
print('The values of the dictionary with the key S.ERASMO are: {}'.format(dict_prefix['S.ERASMO']))
```

The values of the dictionary with the key S.ERASMO are: ['S.ERASMO CHI', 'S.ERASMO CAP', 'S. ERASMO PU']

San Marco

```
In [ ]: # Create a new key in the dictionary with the key 'San Marco'; insert as value the list of strings that have the pr
dict_prefix['SAN MARCO'] = [string for string in dict_prefix['SAN'] if string.startswith('SAN MARCO')]

# Add the value S. MARCO (Gi', 'S. Pietro in Gu') originally in the key 'S.' to the key 'San Marco'
dict_prefix['SAN MARCO'].append('S. MARCO (GI')

# Remove the strings that have the prefix 'San Marco' from the keys 'S.' and 'San'
dict_prefix['SAN'] = [string for string in dict_prefix['SAN'] if not string.startswith('SAN MARCO')]
dict_prefix['S.'] = [string for string in dict_prefix['S.'] if not string.startswith('S. MARCO (GI')]

# Print the values of the dictionary with the key 'San Marco'
print('The values of the dictionary with the key San Marco are: {}'.format(dict_prefix['SAN MARCO']))
```

The values of the dictionary with the key San Marco are: ['SAN MARCO-SA', 'SAN MARCO VA', 'SAN MARCO MA', 'SAN MARCO CA', 'SAN MARCO MO', 'SAN MARCO BO', 'SAN MARCO FO', 'SAN MARCO MU', 'SAN MARCO SA', 'S. MARCO (GI']

San Dona'

```
In [ ]: # Create a new key in the dictionary with the key 'San Dona'; insert as value the list of strings that have the pre
dict_prefix['SAN DONA'] = [string for string in dict_prefix['SAN'] if string.startswith('SAN DONA')]

# Remove the strings that have the prefix 'San Dona' from the keys 'S.' and 'San'
dict_prefix['SAN'] = [string for string in dict_prefix['SAN'] if not string.startswith('SAN DONA')]

# Print the values of the dictionary with the key 'San Dona'
print('The values of the dictionary with the key San Dona are: {}'.format(dict_prefix['SAN DONA']))
```

The values of the dictionary with the key San Dona are: ["SAN DONA' RI", "SAN DONA' PA", "SAN DONA' PI", "SAN DONA' MA", "SAN DONA' CE", "SAN DONA' VA", "SAN DONA' FA", "SAN DONA' PE"]

San Pietro

```
In [ ]: # Create a new key in the dictionary with the key 'San Pietro'; insert as value the list of strings that have the w
dict_prefix['SAN PIETRO'] = [string for string in dict_prefix['SAN'] if 'PIETRO' in string] + [string for string in

# Remove the strings that have the word 'Pietro' from the keys 'S.' and 'San'
dict_prefix['SAN'] = [string for string in dict_prefix['SAN'] if 'PIETRO' not in string]
dict_prefix['S.'] = [string for string in dict_prefix['S.'] if 'PIETRO' not in string]

# Print the values of the dictionary with the key 'San Pietro'
print('The values of the dictionary with the key San Pietro are: {}'.format(dict_prefix['SAN PIETRO']))
```

The values of the dictionary with the key San Pietro are: ['SAN PIETRO D', 'SAN PIETRO P', 'SANPIETRO CA', 'SAN PIETRO C', 'SAN PIETRO B', 'S. PIETRO IN']

Ca' Rossa

```
In [ ]: # Create a new key in the dictionary with the key 'Ca' Rossa'; insert as value the list of strings that have the wo
dict_prefix['CA\' ROSSA'] = [string for string in dict_prefix['CA\''] if 'CA' in string and 'ROSSA' in string]

# Remove the strings that have the word 'Ca' Rossa' from the keys 'Ca'
dict_prefix['CA\''] = [string for string in dict_prefix['CA\''] if 'CA' not in string or 'ROSSA' not in string]

# Print the values of the dictionary with the key 'Ca Rossa'
print('The values of the dictionary with the key Ca\' Rossa are: {}'.format(dict_prefix['CA\' ROSSA']))
```

The values of the dictionary with the key Ca' Rossa are: ["CA' ROSSA OB", "CA' ROSSA VO", "CA' ROSSA SE", "CA' ROSSA BI"]

Manage the remaining values in the keys 'S.' and 'San' and others

```
In [ ]: # Manage the remaining values in the keys 'S.', 'San', 'Santa', 'Sant'', 'Ca'', 'Piazza', 'Piazzale', 'Stazione', '
# Create a new key for each value in the keys as above and assign the value as value of the new key
# Remove the values from the keys as above

if 'S.' in dict_prefix:
    for value in dict_prefix['S.']:
        dict_prefix[value] = [value]
    dict_prefix.pop('S.')

if 'SAN' in dict_prefix:
    for value in dict_prefix['SAN']:
```

```
    dict_prefix[value] = [value]
    dict_prefix.pop('SAN')

if 'SANTA' in dict_prefix:
    for value in dict_prefix['SANTA']:
        dict_prefix[value] = [value]
    dict_prefix.pop('SANTA')

if 'SANT\' ' in dict_prefix:
    for value in dict_prefix['SANT\' ']:
        dict_prefix[value] = [value]
    dict_prefix.pop('SANT\' ')

if 'CA\' ' in dict_prefix:
    for value in dict_prefix['CA\' ']:
        dict_prefix[value] = [value]
    dict_prefix.pop('CA\' ')

if 'PIAZZA' in dict_prefix:
    for value in dict_prefix['PIAZZA']:
        dict_prefix[value] = [value]
    dict_prefix.pop('PIAZZA')

if 'PIAZZALE' in dict_prefix:
    for value in dict_prefix['PIAZZALE']:
        dict_prefix[value] = [value]
    dict_prefix.pop('PIAZZALE')

if 'VIA' in dict_prefix:
    for value in dict_prefix['VIA']:
        dict_prefix[value] = [value]
    dict_prefix.pop('VIA')

if 'STAZIONE' in dict_prefix:
    for value in dict_prefix['STAZIONE']:
        dict_prefix[value] = [value]
    dict_prefix.pop('STAZIONE')

if 'TREVISO' in dict_prefix:
    for value in dict_prefix['TREVISO']:
```



```

        dict_prefix[value] = [value]
    dict_prefix.pop('TREVIS0')

    if 'TRENTO' in dict_prefix:
        for value in dict_prefix['TRENTO']:
            dict_prefix[value] = [value]
        dict_prefix.pop('TRENTO')

    if 'INCR.' in dict_prefix:
        for value in dict_prefix['INCR.']:
            dict_prefix[value] = [value]
        dict_prefix.pop('INCR.')

    if 'DE' in dict_prefix:
        for value in dict_prefix['DE']:
            dict_prefix[value] = [value]
        dict_prefix.pop('DE')

```

Treviso and Trento

```

In [ ]: # Remove the values Treviso, Trento, Trezzo and Treporti from the key 'Tre'
dict_prefix['TRE'] = [string for string in dict_prefix['TRE'] if 'TREVIS0' not in string and 'TRENTO' not in string]

# Print the values of the dictionary with the key 'Tre'
print('The values of the dictionary with the key Tre are: {}'.format(dict_prefix['TRE']))

# TODO: Correct the values of the keys 'Treviso' and 'Trento' with the correct values

```

The values of the dictionary with the key Tre are: ['TRE ARCHI']

Keys with only an item

```

In [ ]: # If a key has only one value, then rename the key with the value
# Use copy() to avoid RuntimeError: dictionary changed size during iteration
for key, value in dict_prefix.copy().items():
    if len(value) == 1:
        dict_prefix[value[0]] = dict_prefix.pop(key)

```

Finally, the update dictionary is

```
In [ ]: # Print the dictionary in the new format  
for key, value in dict_prefix.items():  
    print('{}: {}'.format(key, value))
```

FERROVIA: ['FERROVIA "D"', 'FERROVIA "B"', 'FERROVIA "E"', 'FERROVIA "C"', 'FERROVIA "A"']
VENEZIA: ['VENEZIA', 'VENEZIA CORS']
PIAVE: ['PIAVE FIUME', 'PIAVE PUCCIN', 'PIAVE PODGOR']
RIALTO: ['RIALTO', 'RIALTO MERCA', 'RIALTO "A"']
PELLESTRINA: ['PELLESTRINA', 'PELLESTRINA ']
ACCADEMIA: ['ACCADEMIA "B"', 'ACCADEMIA "A"']
LIDO: ['LIDO S.M.E. ', 'LIDO SAN NIC']
SPINEA: ['SPINEA MARTI', 'SPINEA GIORG', 'SPINEA ALFIE', 'SPINEA PIAZZ', 'SPINEA POZZU', 'SPINEA ROSSI', 'SPINEA D
ESEN', 'SPINEA VILLA', 'SPINEA SANRE', 'SPINEA ORGNA', 'SPINEA REPUB']
LIBERTA': ["LIBERTA' SAN", "LIBERTA' RIG", "LIBERTA' FIN", "LIBERTA' POR", "LIBERTA' COM"]
DON: ['DON STURZO V', 'DON STURZO P']
TRONCHETTO: ['TRONCHETTO M', 'TRONCHETTO F', 'TRONCHETTO', 'TRONCHETTO T', 'TRONCHETTO V']
GIARDINI: ['GIARDINI "B"', 'GIARDINI BIE', 'GIARDINI "A"']
ARSENALE: ['ARSENALE "B"', 'ARSENALE "A"']
GALLO: ['GALLO BARBAR', 'GALLO GIOLIT', 'GALLO LOREDA', 'GALLO TORTA', "GALLO SANT'A", 'GALLO MOROSI', 'GALLO BRAG
AD', 'GALLO CONTAR', 'GALLO DEI KI', 'GALLO DEI GI', 'GALLO BIBLIO', 'GALLO GALOPP']
BANDIERA: ['BANDIERA GHE', 'BANDIERA CRU', 'BANDIERA TOM', 'BANDIERA CAN', 'BANDIERA DUR', 'BANDIERA PIL']
GARIBALDI: ['GARIBALDI MU', 'GARIBALDI OG', 'GARIBALDI GI', 'GARIBALDI C']
BISSUOLA: ['BISSUOLA COL', 'BISSUOLA CAD', 'BISSUOLA ESI', 'BISSUOLA PIS', 'BISSUOLA TEV', 'BISSUOLA VAR']
CARDUCCI: ['CARDUCCI PAS', 'CARDUCCI FEL']
PERON: ['PERON SARAGA', 'PERON BASEGG']
ALTINIA: ['ALTINIA INDR', "ALTINIA CA' ", 'ALTINIA SAN ', 'ALTINIA MUNI', 'ALTINIA BERI', 'ALTINIA FAVA', 'ALTINIA
181', 'ALTINIA FORT']
ORLANDA: ['ORLANDA APPI', 'ORLANDA BAGA', 'ORLANDA PINE', 'ORLANDA SABB', 'ORLANDA SAN ', 'ORLANDA CAVE', 'ORLANDA
PIOV', 'ORLANDA CASI', 'ORLANDA 200', 'ORLANDA TIBU', 'ORLANDA CENT', 'ORLANDA DON ']
MIRANESE: ['MIRANESE SAN', 'MIRANESE PIE', 'MIRANESE SEL', 'MIRANESE GIU', 'MIRANESE CAL', 'MIRANESE IVA', 'MIRANE
SE PER', 'MIRANESE LAZ', 'MIRANESE PIA', 'MIRANESE MON', 'MIRANESE VIV', 'MIRANESE AVA']
TRIESTE: ['TRIESTE CATE', 'TRIESTE ROBI', 'TRIESTE ERAC', 'TRIESTE BOSC', 'TRIESTE MIRA', 'TRIESTE MAZZ', 'TRIESTE
PARC']
SABBADINO: ['SABBADINO L', 'SABBADINO BA', 'SABBADINO PA', 'SABBADINO LA']
CASONA: ['CASONA MARZI', 'CASONA BISSU', 'CASONA VALLE']
TRIESTINA: ['TRIESTINA MO', 'TRIESTINA TE', 'TRIESTINA ZU', 'TRIESTINA LA', 'TRIESTINA UL', 'TRIESTINA AL', 'TRIES
TINA PR', 'TRIESTINA SC', 'TRIESTINA AE', 'TRIESTINA PA', 'TRIESTINA FO', 'TRIESTINA ZO', 'TRIESTINA TO', 'TRIESTI
NA PI']
ALBERONI: ["ALBERONI CA'", 'ALBERONI DEL', 'ALBERONI FAR', 'ALBERONI STE', 'ALBERONI COL', 'ALBERONI GOL', 'ALBERO
NI SAN', 'ALBERONI OTT', 'ALBERONI SPI']
PADANA: ['PADANA CIVIC', 'PADANA AVENA', 'PADANA STAZI', 'PADANA COLOM', 'PADANA DELLE', 'PADANA DEL L']
MARTELLAGO: ['MARTELLAGO', 'MARTELLAGO V', 'MARTELLAGO F', 'MARTELLAGO D', 'MARTELLAGOTR']
MARGHERA: ['MARGHERA ARD', 'MARGHERA CIM', 'MARGHERA SAL']
CAPPUCCINA: ['CAPPUCCINA V', 'CAPPUCCINA S', 'CAPPUCCINA B']

GUGLIE: ['GUGLIE', 'GUGLIE "B"']
TERRAGLIO: ['TERRAGLIO VI', 'TERRAGLIO PE', 'TERRAGLIO TE', 'TERRAGLIO GA', 'TERRAGLIO BO', 'TERRAGLIO CA', 'TERRAGLIO NI', 'TERRAGLIO FA']
CHIOGGIA: ['CHIOGGIA PIA', 'CHIOGGIA SAN', 'CHIOGGIA CAM', 'CHIOGGIA ZAR', 'CHIOGGIA DE', 'CHIOGGIA DEN']
S.ERASMO: ['S.ERASMO CHI', 'S.ERASMO CAP', 'S. ERASMO PU']
TOSATTO: ['TOSATTO IMPA', 'TOSATTO PACC']
PERTINI: ['PERTINI CHIE', 'PERTINI FOSC', 'PERTINI VESP', 'PERTINI CARR', 'PERTINI GAVA']
CASTELLANA: ['CASTELLANA S', 'CASTELLANA C', 'CASTELLANA P', 'CASTELLANA B', 'CASTELLANA M']
MONTE: ['MONTE CELO F', 'MONTE CERVIN']
MURANO: ['MURANO VENIE', 'MURANO NAVAG', 'MURANO FARO', 'MURANO MUSEO', 'MURANO COLON', 'MURANO DA MU', 'MURANO SE REN']
MOGLIANO: ['MOGLIANO BUR', 'MOGLIANO BEL', 'MOGLIANO MAR', 'MOGLIANO FS', 'MOGLIANO MUN', 'MOGLIANO CEN']
DESE: ['DESE CENTRO', 'DESE CICOGNE', 'DESE FS', 'DESE LITOMAR']
GOBBI: ['GOBBI ORLAND', 'GOBBI SAN DO', 'GOBBI VALLEN', 'GOBBI MIRTIL', 'GOBBI DON BO', 'GOBBI CA' DO', 'GOBBI MAN DAR']
MALAMOCCO: ['MALAMOCCO PA', 'MALAMOCCO AL', 'MALAMOCCO ST', 'MALAMOCCO CE', 'MALAMOCCO BA', 'MALAMOCCO GA', 'MALAMOCCO BE', 'MALAMOCCO OC', 'MALAMOCCO SC']
RINASCITA: ['RINASCITA BE', 'RINASCITA EM']
ASSEGGIANO: ['ASSEGGIANO V', 'ASSEGGIANO M', 'ASSEGGIANO C', 'ASSEGGIANO E', 'ASSEGGIANO P', 'ASSEGGIANO D']
OSPEDALE: ['OSPEDALE', 'OSPEDALE DEL', 'OSPEDALE MIR']
BURANO: ['BURANO "C"', 'BURANO "B"', 'BURANO "A"']
ANCONA: ['ANCONA TORIN', 'ANCONA CARBO']
TRIVIGNANO: ['TRIVIGNANO L', 'TRIVIGNANO B', 'TRIVIGNANO C', 'TRIVIGNANO P', 'TRIVIGNANO G']
ZELARINO: ['ZELARINO CHI', 'ZELARINO CAR', 'ZELARINO MUN', 'ZELARINO PAR']
MATTUGLIE: ['MATTUGLIE DI', 'MATTUGLIE PE']
BECCARIA: ['BECCARIA CAN', 'BECCARIA CHI', 'BECCARIA COR', 'BECCARIA ORO', 'BECCARIA PAR']
MARCONI: ['MARCONI LORE', 'MARCONI MARC', 'MARCONI FIUM', 'MARCONI DES ', 'MARCONI CORA']
MARCON: ['MARCONI LORE', 'MARCONI MARC', 'MARCON COOPE', 'MARCON FIUM', 'MARCON CENTR', 'MARCONI DES ', 'MARCONI CORA', 'MARCON LOMAB', 'MARCON ALTIN', 'MARCON CULT', 'MARCON MUNIC', 'MARCON VITTO']
VESPUCCI: ['VESPUCCI GRI', 'VESPUCCI BOE', 'VESPUCCI SAN', 'VESPUCCI CA', 'VESPUCCI CAT', 'VESPUCCI PIG', 'VESPUCCI GAR']
TORINO: ['TORINO UNIVE', 'TORINO ROSSE', 'TORINO']
CAPOLINEA: ['CAPOLINEA CA', 'CAPOLINEA FU']
RIELTA: ['RIELTA CA' R', 'RIELTA PARCO']
CERTOSA: ['CERTOSA', 'CERTOSA (A ']
D'ANNUNZIO: ['D'ANNUNZIO D', 'D'ANNUNZIO P', 'D'ANNUNZIO S']
CAVERGNAGO: ['CAVERGNAGO M', 'CAVERGNAGO T']
MIRANO: ['MIRANO SCUOL', 'MIRANO MATTE', 'MIRANO CENTR', 'MIRANO CARDU', 'MIRANO GRAMS', 'MIRANO PESTR', 'MIRANO T REVI', 'MIRANO SPORT', 'MIRANO PERUG', 'MIRANO GRIMA', 'MIRANO DANTE', 'MIRANO BATTI', 'MIRANO FOSSA']
TEVERE: ['TEVERE BAGLI', 'TEVERE PARCO']

NOALE: ['NOALE', 'NOALE OSPEDA', 'NOALE LANCER']
 DOLO: ['DOLO MANZONI', 'DOLO CENTRO', "DOLO CA' TRO", 'DOLO BANDIER', 'DOLO DORIA', 'DOLO CIVICO ']
 CHIMICA: ['CHIMICA INGR', 'CHIMICA MECC']
 PREGANZIOL: ['PREGANZIOL F', 'PREGANZIOL G', 'PREGANZIOL ', 'PREGANZIOL M']
 PASINI: ['PASINI FRATE', 'PASINI LAVOR']
 TREZZO: ['TREZZO TERRA', 'TREZZO BATTU']
 GATTA: ['GATTA SANTA ', 'GATTA SCARAN', 'GATTA Verci', 'GATTA 90', 'GATTA PROTAG', 'GATTA SCARAM']
 BOTTENIGO: ['BOTTENIGO PI', 'BOTTENIGO CA', 'BOTTENIGO BO', 'BOTTENIGO MA']
 PORTOSECCE: ['PORTOSECCE', 'PORTOSECCE L', 'PORTOSECCE C']
 BRENDOL: ['BRENDOL STI', 'BRENDOL FAV', 'BRENDOL ARS', 'BRENDOL']
 ROBEGANO: ['ROBEGANO MON', 'ROBEGANO 25 ']
 VILLABONA: ['VILLABONA PI', 'VILLABONA BO', 'VILLABONA 87', 'VILLABONA MO', 'VILLABONA 8']
 VISINONI: ['VISINONI COM', 'VISINONI POL']
 SCORZE': ["SCORZE' MOGL", "SCORZE' CAPO", "SCORZE' SAN ", "SCORZE' MUNI", "SCORZE' ROMA", "SCORZE' FERM", "SCORZE' CECC"]
 SCUOLA: ['SCUOLA MEDIA', 'SCUOLA ZENDR']
 MORANZANI: ['MORANZANI EL', 'MORANZANI 32', 'MORANZANI CO']
 SOTTOMARINA: ['SOTTOMARINA ', 'SOTTOMARINA']
 FIESSO: ["FIESSO D'ART", 'FIESSO GEMIT']
 OLMO: ['OLMO VITTORI', 'OLMO PELLICO', 'OLMO PAPA LU', 'OLMO CALVI']
 MAERNE: ['MAERNE FS', 'MAERNE CENTR', 'MAERNE CHIES', 'MAERNE CIMIT']
 MIRA: ['MIRANESE SAN', 'MIRANESE PIE', 'MIRANESE SEL', 'MIRANESE GIU', 'MIRANESE CAL', 'MIRANESE IVA', 'MIRANESE P
ER', 'MIRANO SCUOL', 'MIRANESE LAZ', 'MIRANESE PIA', 'MIRANO MATTE', 'MIRANESE MON', 'MIRANESE VIV', 'MIRANESE AV
A', 'MIRANO CENTR', 'MIRANO CARDU', 'MIRANO GRAMS', 'MIRANO PESTR', 'MIRANO TREVI', 'MIRANO SPORT', 'MIRA RISCOS
S', 'MIRANO PERUG', 'MIRANO GRIMA', 'MIRANO DANTE', 'MIRA PORTE', 'MIRA ALIGHIE', 'MIRANO BATTI', 'MIRANO FOSSA',
'MIRA ALBRIZZ', 'MIRA CENTRO']
 ORIAGO: ['ORIAGO STAZI', 'ORIAGO SOMMO', 'ORIAGO ROMAG', 'ORIAGO CENTR']
 KLINGER: ['KLINGER SAN ', 'KLINGER RAVA', 'KLINGER PRES']
 PADOVA: ['PADOVA FIERA', 'PADOVA SAN L']
 CIVE': ["CIVE' PALTAN", "CIVE' MIRA"]
 P.LE ROMA: ['P.LE ROMA "G', 'P.LE ROMA "E', 'P.LE ROMA "B', 'P.LE ROMA "F', 'P.LE ROMA "C', 'P.LE ROMA "D', 'P.LE
ROMA "A']
 F.TE NOVE: ['F.TE NOVE "C', 'F.TE NOVE "A', 'F.TE NOVE "D', 'F.TE NOVE']
 SAN MARCO: ['SAN MARCO-SA', 'SAN MARCO VA', 'SAN MARCO MA', 'SAN MARCO CA', 'SAN MARCO MO', 'SAN MARCO BO', 'SAN M
ARCO FO', 'SAN MARCO MU', 'SAN MARCO SA', 'S. MARCO (GI']
 SAN DONA: ["SAN DONA' RI", "SAN DONA' PA", "SAN DONA' PI", "SAN DONA' MA", "SAN DONA' CE", "SAN DONA' VA", "SAN DO
NA' FA", "SAN DONA' PE"]
 SAN PIETRO: ['SAN PIETRO D', 'SAN PIETRO P', 'SANPIETRO CA', 'SAN PIETRO C', 'SAN PIETRO B', 'S. PIETRO IN']
 CA' ROSSA: ["CA' ROSSA OB", "CA' ROSSA VO", "CA' ROSSA SE", "CA' ROSSA BI"]
 GIUDECCA PAL: ['GIUDECCA PAL']

AEROPORTO MA: ['AEROPORTO MA']
MESTRE CENTR: ['MESTRE CENTR']
ZATTERE: ['ZATTERE']
GIGLIO: ['GIGLIO']
ZITELLE: ['ZITELLE']
ORTO: ['ORTO']
PUNTA SABBIO: ['PUNTA SABBIO']
CANAL LEONE: ['CANAL LEONE']
LAVELLI PAOL: ['LAVELLI PAOL']
RIZZARDI CAR: ['RIZZARDI CAR']
REDENTORE: ['REDENTORE']
BELFREDO TER: ['BELFREDO TER']
27 OTTOBRE D: ['27 OTTOBRE D']
FISICA DEPOS: ['FISICA DEPOS']
TREPORTI: ['TREPORTI']
CORSO DEL PO: ['CORSO DEL PO']
RAMPA CAVALC: ['RAMPA CAVALC']
CELESTIA: ['CELESTIA']
SALUTE: ['SALUTE']
TORCELLO: ['TORCELLO']
PEOPLE MOVER: ['PEOPLE MOVER']
PALEOCAPA PA: ['PALEOCAPA PA']
SACCA FISOLA: ['SACCA FISOLA']
PASQUALIGO M: ['PASQUALIGO M']
RIVA DE BIAS: ['RIVA DE BIAS']
CAFASSO BOTT: ['CAFASSO BOTT']
CAVALCAVIA V: ['CAVALCAVIA V']
CALABRIA CAM: ['CALABRIA CAM']
TRE ARCHI: ['TRE ARCHI']
PARK PETROLI: ['PARK PETROLI']
ILARIA ALPI : ['ILARIA ALPI ']
MALCONTENTA : ['MALCONTENTA ']
FORTE MARGHE: ['FORTE MARGHE']
QUARNARO CAL: ['QUARNARO CAL']
SPIRITO SANT: ['SPIRITO SANT']
OLIVI: ['OLIVI']
GAZZERA ALTA: ['GAZZERA ALTA']
FAVRETTI MES: ['FAVRETTI MES']
PAGANELLO TI: ['PAGANELLO TI']
CIRCONVALLAZ: ['CIRCONVALLAZ']

CORRENTI CAP: ['CORRENTI CAP']
PASSO CAMPAL: ['PASSO CAMPAL']
CIRCONVALAZI: ['CIRCONVALAZI']
SALAMONIO MA: ['SALAMONIO MA']
CIMITERO: ['CIMITERO']
DURANDO BELL: ['DURANDO BELL']
CREA: ['CREA']
MAZZORBO: ['MAZZORBO']
GALILEI DARS: ['GALILEI DARS']
FARO ROCCHET: ['FARO ROCCHET']
PALAZZO DEL : ['PALAZZO DEL ']
VIGNOLE: ['VIGNOLE']
SELVANESE PL: ['SELVANESE PL']
AGENZIA ENT: ['AGENZIA ENT']
MORANDI NICE: ['MORANDI NICE']
BACINI - ARS: ['BACINI - ARS']
EINAUDI CAST: ['EINAUDI CAST']
PALIAGA CA' : ["PALIAGA CA' "]
MARSALA CENT: ['MARSALA CENT']
MADONNA DELL: ['MADONNA DELL']
CALUCCI QUAR: ['CALUCCI QUAR']
CAMPORESE GR: ['CAMPORESE GR']
PAOLUCCI LON: ['PAOLUCCI LON']
COLOMBO: ['COLOMBO']
GIOVANNACCI : ['GIOVANNACCI ']
TITO CASTELL: ['TITO CASTELL']
GOZZI CAPPUC: ['GOZZI CAPPUC']
CALVI PARMES: ['CALVI PARMES']
SALICI VILLA: ['SALICI VILLA']
FAVIGNANA: ['FAVIGNANA']
GRAN VIALE: ['GRAN VIALE']
RISORGIMENTO: ['RISORGIMENTO']
SANPIETRO CA: ['SANPIETRO CA']
LAZZARETTO N: ['LAZZARETTO N']
CAROMAN: ['CAROMAN']
OSPIZIO MARI: ['OSPIZIO MARI']
CAVANIS CAPO: ['CAVANIS CAPO']
AZOTO SOTTAN: ['AZOTO SOTTAN']
MARTIRI DELL: ['MARTIRI DELL']
ZERO BRANCO: ['ZERO BRANCO']

VALLENARI ST: ['VALLENARI ST']
SALZANO TOSC: ['SALZANO TOSC']
GAGGIO FERMI: ['GAGGIO FERMI']
ZENDRINI VIL: ['ZENDRINI VIL']
PESEGGIA VI: ['PESEGGIA VI']
CORTIVO TOMB: ['CORTIVO TOMB']
ROMEA MALCAN: ['ROMEA MALCAN']
ISOLA UNIONE: ['ISOLA UNIONE']
MARTELLAGOTR: ['MARTELLAGOTR']
STRA PIAZZA : ['STRA PIAZZA ']
CAMPOCROCE: ['CAMPOCROCE']
FERM.SERV. D: ['FERM.SERV. D']
AREOPORTO MA: ['AREOPORTO MA']
VALLON BORGO: ['VALLON BORGO']
SFMR SPINEA: ['SFMR SPINEA']
ROSOLINA: ['ROSOLINA']
ROSARA: ['ROSARA']
RIO SAN MART: ['RIO SAN MART']
FRESCADA: ['FRESCADA']
APPRODO NAVE: ['APPRODO NAVE']
CALTANA: ['CALTANA']
PIOVE DI SAC: ['PIOVE DI SAC']
BOJON: ['BOJON']
TESSERA SCUO: ['TESSERA SCUO']
REBOSOLA CIV: ['REBOSOLA CIV']
CAMPAGNA LUP: ['CAMPAGNA LUP']
VIGONZA PERA: ['VIGONZA PERA']
CAPRICCIO CE: ['CAPRICCIO CE']
S. MARCUOLA-: ['S. MARCUOLA-']
SAN STAE: ['SAN STAE']
SANTA MARIA : ['SANTA MARIA ']
SAN TOMA': ["SAN TOMA'"]
SAN ROCCO BR: ['SAN ROCCO BR']
SAN GIORGIO: ['SAN GIORGIO']
SANT' ELENA: ["SANT' ELENA"]
SANTA MARTA: ['SANTA MARTA']
SANT' ANGELO: ["SANT' ANGELO"]
SANT'ANTONIO: ["SANT'ANTONIO"]
SAN SAMUELE: ['SAN SAMUELE']
SAN SILVESTR: ['SAN SILVESTR']

SAN NICOLO': ["SAN NICOLO'"]
SANT' ALVISE: ["SANT' ALVISE"]
SAN SERVULO: ['SAN SERVULO']
SAN BASILIO: ['SAN BASILIO']
SAN NICOLO' : ["SAN NICOLO' "]
SAN LAZZARO: ['SAN LAZZARO']
SAN LIBERALE: ['SAN LIBERALE']
SAN SOVINO VE: ['SAN SOVINO VE']
SAN TROVASO: ['SAN TROVASO']
CA' D'ORO: ["CA' D'ORO"]
CA' REZZONIC: ["CA' REZZONIC"]
CA' SABBIONI: ["CA' SABBIONI"]
CA' MARCELLO: ["CA' MARCELLO"]
CA' LIN GATT: ["CA' LIN GATT"]
CA' LIN ERAC: ["CA' LIN ERAC"]
CA' BIANCA L: ["CA' BIANCA L"]
CA' LIN CAST: ["CA' LIN CAST"]
CA' SOLARO C: ["CA' SOLARO C"]
PIAZZA MERCA: ['PIAZZA MERCA']
PIAZZALE GIO: ['PIAZZALE GIO']
PIAZZALE RAV: ['PIAZZALE RAV']
VIA VILLABON: ['VIA VILLABON']
VIA DEI CANT: ['VIA DEI CANT']
STAZIONE MES: ['STAZIONE MES']
STAZIONE PAD: ['STAZIONE PAD']
STAZIONE MAR: ['STAZIONE MAR']
TREVISO SAN : ['TREVISO SAN ']
TREVISO FS: ['TREVISO FS']
TRENTO PODGO: ['TRENTO PODGO']
TRENTO FAGAR: ['TRENTO FAGAR']
TRENTO GAZZE: ['TRENTO GAZZE']
INCR. VIA DA: ['INCR. VIA DA']
INCR. VIA GR: ['INCR. VIA GR']
INCR. VIA VE: ['INCR. VIA VE']
DESE CENTRO: ['DESE CENTRO']
DEI MURAZZI : ['DEI MURAZZI ']
DESE CICOGNE: ['DESE CICOGNE']
DESE FS: ['DESE FS']
DESE LITOMAR: ['DESE LITOMAR']
DE NICOLA CH: ['DE NICOLA CH']

```
In [ ]: # Export the dictionary in a json file
name_file = 'dict_prefix_' + file_name.split('.')[0] + '.json'
with open('data/dictionaries/' + name_file, 'w') as fp:
    json.dump(dict_prefix, fp)
```

Useless stamps

```
In [ ]: # TODO: #1 Remove useless rows that have a minimum temporal gap for the same serial and fermata
# DE-COMMENT THE FOLLOWING LINES OF CODE
```

```
In [ ]: # Find the serial with the highest number of validations, and the same for each ticket profile, save the results i
dict_serial = {}
for ticket in df['TICKET_CODE'].unique():
    dict_serial[ticket] = df[df['TICKET_CODE'] == ticket]['SERIALE'].value_counts().index[0]

# Print the serial with the highest number of validations, and the same for each ticket profile
for ticket in df['TICKET_CODE'].unique():
    print('The serial with the highest number of validations for the ticket profile {} is: {}'.format(ticket, dict_s
```

```
The serial with the highest number of validations for the ticket profile 4 is: 36142613562646276
The serial with the highest number of validations for the ticket profile 2 is: 41675643963691780
The serial with the highest number of validations for the ticket profile 3 is: 40549743932450308
The serial with the highest number of validations for the ticket profile 1 is: 40831218909282052
The serial with the highest number of validations for the ticket profile 7 is: 36405993298989316
The serial with the highest number of validations for the ticket profile 5 is: 36144856468188676
```

```
In [ ]: # Group by the serial and the fermata
# df = df.groupby(['SERIALE', 'FERMATA']).apply(lambda x: x.sort_values(by='DATA_VALIDAZIONE', ascending=True))

# Print the first 5 rows of the df
# df.head()
# DO NOT DE-COMMENT THIS CELL
```

```
In [ ]: # Reset the index of the df and drop the old index in order to have a new index starting from 0 to the number of ro
# It is necessary to have a new index because the groupby function has created a multi-index
df.reset_index(drop=True, inplace=True)
```

```
In [ ]: # Create a new column 'MIN_TEMPORAL_GAP' that contains the minimum temporal gap between two validations for the same  
df = df.groupby(['SERIALE', 'DATA', 'DESCRIZIONE']).apply(lambda x: x.assign(MIN_TEMPORAL_GAP = x['DATA_VALIDAZIONE']
```

```
In [ ]: df.head(20)
```

Out[]:

	DATA	ORA	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	TICKET_CODE	DESCRIZIONE_TITOLO	MIN
0	2022-05-13	00:00:00	2022-05-13 00:00:00	65676291870913797	5089	FERROVIA "D"	11149	4	7GG-TPL 43,60- COMVE16,40	
1	2022-05-13	00:00:00	2022-05-13 00:00:00	36141384536591364	5032	FERROVIA "B"	11107	2	48H-TPL 24,90- COMVE5,10	
2	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856606063108	5031	P.LE ROMA "G"	11108	3	72H-TPL 33,40- COMVE6,60	
3	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856474364932	506	VENEZIA	11261	1	DAILY PASS VENEZIA - AVM	
4	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856606062852	5031	P.LE ROMA "G"	11108	3	72H-TPL 33,40- COMVE6,60	
5	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856474364676	506	VENEZIA	11261	1	DAILY PASS VENEZIA - AVM	
6	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856474361092	506	VENEZIA	11261	1	DAILY PASS VENEZIA - AVM	
7	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856605835780	507	VENEZIA	11107	2	48H-TPL 24,90- COMVE5,10	
8	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856605836036	507	VENEZIA	11107	2	48H-TPL 24,90- COMVE5,10	
9	2022-05-13	00:00:00	2022-05-13 00:00:00	36144856474361348	506	VENEZIA	11261	1	DAILY PASS VENEZIA - AVM	
10	2022-05-13	00:01:00	2022-05-13 00:01:00	36423061903008260	5022	GIUDECCA PAL	11149	4	7GG-TPL 43,60- COMVE16,40	
11	2022-05-13	00:01:00	2022-05-13 00:01:00	36423061903008260	5022	GIUDECCA PAL	11149	4	7GG-TPL 43,60- COMVE16,40	
12	2022-05-13	00:01:00	2022-05-13 00:01:00	36426331591572228	506	VENEZIA	11553	3	72H R.VENICE+AEROP.AR ONLINE	
13	2022-05-13	00:01:00	2022-05-13 00:01:00	36426331591501316	506	VENEZIA	11553	3	72H R.VENICE+AEROP.AR ONLINE	
14	2022-05-13	00:01:00	2022-05-13 00:01:00	36426331591504644	506	VENEZIA	11553	3	72H R.VENICE+AEROP.AR	

	DATA	ORA	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	TICKET_CODE	DESCRIZIONE_TITOLO	MIN
									ONLINE	
15	2022-05-13	00:01:00	2022-05-13 00:01:00	65676290387638021	5022	GIUDECCA PAL	11149	4	7GG-TPL 43,60- COMVE16,40	
16	2022-05-13	00:01:00	2022-05-13 00:01:00	37270756644007428	5132	S. MARCUIOLA-	11101	7	75'-TPL 6,64- COMVE0,86	
17	2022-05-13	00:01:00	2022-05-13 00:01:00	37270756644007684	5132	S. MARCUIOLA-	11101	7	75'-TPL 6,64- COMVE0,86	
18	2022-05-13	00:01:00	2022-05-13 00:01:00	36144856464934660	5032	FERROVIA "B"	11101	7	75'-TPL 6,64- COMVE0,86	
19	2022-05-13	00:01:00	2022-05-13 00:01:00	36144856464934404	5032	FERROVIA "B"	11101	7	75'-TPL 6,64- COMVE0,86	

In []: `df.tail(20)`

Out[]:

	DATA	ORA	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	TICKET_CODE	DESCRIZIONE_TI
4680555	2022-07-15	01:02:00	2022-07-15 01:02:00	36146081881259268	5013	SAN MARCO-SA	11261	1	DAILY PASS VENI
4680556	2022-07-15	01:10:00	2022-07-15 01:10:00	40832955277083652	509	VENEZIA	12101	7	BIGL.AUT.75'MESTRE/
4680557	2022-07-15	01:13:00	2022-07-15 01:13:00	41958855181075972	5013	SAN MARCO-SA	11109	3	BIGLIETTO 72 ORE V
4680558	2022-07-15	01:13:00	2022-07-15 01:13:00	37553456891176708	5013	SAN MARCO-SA	11107	2	48H-TPL 2 COMV
4680559	2022-07-15	01:28:00	2022-07-15 01:28:00	40551480566825732	5013	SAN MARCO-SA	11149	4	7GG-TPL 4 COMVE
4680560	2022-07-15	01:28:00	2022-07-15 01:28:00	40551480566829060	5013	SAN MARCO-SA	11149	4	7GG-TPL 4 COMVE
4680561	2022-07-15	01:48:00	2022-07-15 01:48:00	40551480575979012	5043	SAN TOMA'	11107	2	48H-TPL 2 COMV
4680562	2022-07-15	01:51:00	2022-07-15 01:51:00	40551480298275076	5036	SAN STAE	11261	1	DAILY PASS VENI
4680563	2022-07-15	01:51:00	2022-07-15 01:51:00	40551480298275076	5036	SAN STAE	11261	1	DAILY PASS VENI
4680564	2022-07-15	02:00:00	2022-07-15 02:00:00	41958855447884548	510	VENEZIA	12101	7	BIGL.AUT.75'MESTRE/
4680565	2022-07-15	02:00:00	2022-07-15 02:00:00	37271982183274756	5013	SAN MARCO-SA	11261	1	DAILY PASS VENI
4680566	2022-07-15	02:01:00	2022-07-15 02:01:00	37271982183271940	5013	SAN MARCO-SA	11261	1	DAILY PASS VENI
4680567	2022-07-15	02:01:00	2022-07-15 02:01:00	36146082284251652	510	VENEZIA	12101	7	BIGL.AUT.75'MESTRE/
4680568	2022-07-15	02:01:00	2022-07-15 02:01:00	36146082284251908	510	VENEZIA	12101	7	BIGL.AUT.75'MESTRE/
4680569	2022-07-15	02:01:00	2022-07-15 02:01:00	40832955277145348	510	VENEZIA	12101	7	BIGL.AUT.75'MESTRE/
4680570	2022-	02:27:00	2022-07-15	37271982183271940	4525	SANTA MARIA	11261	1	DAILY PASS VENI

	DATA	ORA	DATA_VALIDAZIONE	SERIALE	FERMATA	DESCRIZIONE	TITOLO	TICKET_CODE	DESCRIZIONE_TI
	07-15		02:27:00						
4680571	2022-07-15	02:27:00	2022-07-15 02:27:00	37271982183274756	4525	SANTA MARIA	11261	1	DAILY PASS VENI
4680572	2022-07-15	04:33:00	2022-07-15 04:33:00	36088514819663876	5030	P.LE ROMA "F	5	7	75'-TPL 6,64-COMV
4680573	2022-07-15	05:06:00	2022-07-15 05:06:00	40832955551087108	509	VENEZIA	12101	7	BIGL.AUT.75'MESTRE/
4680574	2022-07-15	05:13:00	2022-07-15 05:13:00	40832947760207876	509	VENEZIA	12101	7	BIGL.AUT.75'MESTRE/

In []: `df['MIN_TEMPORAL_GAP'].value_counts()`

```
Out[ ]: 0.0      130011
        1.0      38129
        2.0      31106
        3.0      23096
        4.0      16897
        5.0      13775
        6.0      11365
        7.0       9447
        8.0       8110
        9.0       6814
       10.0       5831
       11.0       4932
       12.0       4690
       13.0       3844
       14.0       3345
       15.0       3013
       16.0       2498
       17.0       2291
       18.0       2172
       19.0       1834
       20.0       1655
       21.0       1365
       22.0       1223
       23.0       1169
       24.0       1022
       25.0        984
       26.0        840
       27.0        719
       28.0        678
       30.0        640
       29.0        620
       32.0        536
       31.0        492
       33.0        469
       35.0        446
       38.0        431
       34.0        412
       37.0        379
       36.0        374
       39.0        343
```


247.0	330
44.0	329
41.0	325
260.0	323
274.0	323
239.0	319
215.0	318
248.0	318
303.0	314
353.0	314
224.0	309
267.0	309
40.0	308
262.0	308
235.0	307
249.0	306
42.0	305
222.0	304
332.0	304
218.0	303
227.0	303
204.0	303
253.0	303
326.0	302
302.0	302
305.0	301
223.0	301
290.0	300
383.0	299
234.0	299
276.0	299
265.0	297
313.0	297
273.0	296
340.0	296
325.0	296
382.0	295
297.0	295
360.0	295
205.0	294

352.0	293
60.0	293
300.0	293
309.0	292
45.0	292
217.0	290
458.0	290
196.0	290
46.0	290
312.0	290
289.0	289
337.0	289
203.0	288
187.0	287
207.0	287
311.0	285
272.0	285
198.0	284
307.0	284
320.0	284
232.0	284
229.0	284
351.0	283
426.0	283
245.0	283
368.0	282
323.0	282
238.0	282
243.0	282
173.0	281
165.0	281
346.0	281
365.0	281
318.0	280
240.0	280
201.0	280
277.0	280
226.0	280
43.0	280
283.0	279

162.0	279
450.0	278
221.0	278
202.0	278
264.0	277
408.0	277
435.0	277
284.0	276
362.0	276
280.0	276
281.0	276
246.0	276
233.0	275
361.0	275
197.0	275
451.0	275
167.0	274
350.0	273
269.0	273
322.0	273
321.0	273
406.0	273
338.0	272
401.0	272
216.0	272
385.0	272
319.0	271
161.0	271
354.0	271
189.0	271
355.0	271
183.0	271
333.0	271
299.0	271
191.0	270
210.0	270
56.0	270
178.0	270
356.0	270
372.0	270

381.0	269
287.0	269
286.0	269
212.0	269
449.0	269
268.0	269
308.0	268
231.0	268
384.0	267
250.0	267
241.0	267
254.0	267
278.0	267
275.0	266
328.0	266
418.0	266
314.0	265
363.0	265
395.0	265
491.0	265
373.0	264
390.0	264
190.0	264
317.0	263
54.0	263
208.0	263
257.0	263
481.0	262
61.0	262
181.0	262
47.0	261
316.0	261
330.0	261
306.0	261
347.0	261
296.0	260
440.0	260
271.0	260
171.0	260
259.0	260

255.0	260
282.0	260
403.0	260
51.0	259
344.0	259
327.0	259
376.0	258
200.0	258
206.0	258
422.0	258
160.0	258
366.0	258
345.0	258
194.0	258
236.0	258
279.0	258
180.0	257
230.0	257
392.0	257
310.0	257
301.0	257
412.0	257
59.0	257
251.0	257
339.0	257
342.0	257
446.0	256
228.0	256
371.0	256
432.0	256
430.0	255
404.0	255
411.0	255
49.0	255
419.0	255
359.0	255
398.0	254
417.0	254
349.0	254
367.0	254

214.0	253
209.0	253
357.0	253
252.0	253
456.0	253
388.0	253
163.0	253
270.0	253
410.0	253
288.0	253
294.0	253
341.0	252
170.0	252
256.0	252
211.0	252
427.0	251
396.0	251
169.0	251
431.0	251
199.0	251
444.0	250
120.0	250
143.0	250
52.0	250
387.0	250
391.0	249
402.0	249
462.0	249
476.0	249
329.0	249
364.0	248
489.0	248
55.0	248
414.0	248
177.0	248
136.0	247
433.0	247
63.0	247
472.0	247
441.0	247

266.0	247
242.0	246
477.0	246
517.0	246
467.0	246
48.0	246
478.0	246
389.0	246
343.0	246
292.0	246
416.0	246
380.0	246
369.0	246
442.0	245
57.0	245
176.0	245
291.0	244
348.0	244
334.0	244
453.0	244
474.0	244
393.0	243
455.0	243
53.0	243
304.0	243
184.0	243
421.0	243
151.0	242
399.0	242
293.0	242
479.0	242
182.0	242
150.0	242
377.0	242
386.0	241
188.0	241
394.0	241
315.0	241
336.0	241
335.0	240

473.0	240
464.0	240
358.0	240
438.0	240
374.0	240
263.0	240
152.0	240
219.0	239
172.0	239
195.0	239
465.0	239
469.0	238
378.0	238
186.0	238
185.0	238
429.0	238
324.0	238
460.0	238
448.0	237
425.0	237
400.0	236
70.0	236
409.0	236
144.0	235
475.0	234
370.0	234
50.0	234
220.0	233
67.0	233
103.0	232
436.0	232
379.0	232
237.0	232
244.0	232
434.0	231
166.0	231
164.0	231
128.0	230
529.0	230
482.0	230

495.0	230
498.0	230
428.0	229
375.0	229
457.0	229
492.0	229
298.0	229
461.0	228
135.0	228
295.0	228
454.0	228
261.0	227
471.0	227
331.0	227
192.0	227
468.0	226
102.0	226
58.0	226
507.0	226
124.0	225
213.0	225
168.0	225
420.0	225
452.0	224
504.0	224
105.0	223
158.0	223
437.0	223
407.0	223
225.0	222
146.0	222
127.0	222
107.0	222
447.0	221
285.0	221
513.0	221
506.0	221
153.0	220
490.0	220
485.0	220

116.0	220
175.0	220
509.0	220
523.0	219
480.0	219
466.0	219
130.0	219
512.0	219
459.0	218
65.0	218
505.0	218
179.0	217
159.0	217
62.0	217
511.0	217
142.0	217
258.0	217
508.0	217
415.0	216
93.0	216
413.0	216
484.0	216
494.0	216
121.0	215
109.0	215
423.0	215
193.0	214
524.0	214
397.0	214
541.0	213
87.0	213
522.0	213
502.0	213
470.0	212
510.0	212
443.0	212
91.0	212
540.0	211
405.0	211
149.0	211

134.0	211
542.0	211
99.0	210
424.0	210
69.0	210
104.0	209
487.0	209
106.0	209
129.0	207
500.0	207
132.0	207
515.0	207
174.0	206
526.0	206
145.0	206
157.0	206
101.0	206
544.0	206
501.0	205
119.0	205
155.0	205
528.0	205
126.0	205
561.0	204
521.0	204
493.0	204
503.0	204
108.0	204
82.0	204
148.0	203
536.0	202
531.0	202
497.0	202
559.0	202
74.0	201
95.0	200
486.0	200
533.0	200
488.0	199
499.0	199

115.0	198
75.0	198
546.0	198
123.0	198
439.0	197
137.0	197
514.0	197
156.0	197
111.0	196
72.0	196
445.0	196
463.0	195
100.0	195
539.0	195
76.0	195
140.0	194
88.0	194
483.0	193
138.0	193
71.0	192
68.0	192
154.0	191
519.0	191
122.0	191
554.0	190
525.0	190
133.0	190
543.0	190
73.0	189
139.0	189
90.0	189
564.0	189
518.0	189
520.0	188
496.0	188
131.0	187
141.0	187
535.0	186
92.0	186
64.0	186

538.0	185
81.0	185
110.0	185
97.0	185
66.0	184
551.0	183
580.0	183
552.0	183
83.0	183
530.0	182
112.0	182
547.0	182
570.0	181
527.0	180
79.0	180
125.0	180
114.0	179
85.0	179
78.0	179
86.0	179
550.0	178
557.0	178
556.0	178
569.0	178
565.0	176
147.0	176
89.0	175
562.0	175
537.0	174
117.0	174
579.0	172
80.0	171
555.0	170
545.0	169
77.0	169
553.0	168
548.0	167
576.0	167
118.0	166
532.0	165

534.0	165
84.0	164
94.0	164
549.0	164
605.0	163
558.0	162
583.0	161
113.0	160
98.0	160
600.0	159
516.0	159
589.0	159
573.0	158
568.0	158
96.0	157
560.0	157
575.0	155
567.0	154
588.0	152
577.0	150
571.0	150
587.0	149
593.0	148
578.0	146
599.0	146
566.0	145
607.0	145
590.0	144
574.0	143
614.0	141
601.0	140
603.0	140
596.0	139
581.0	138
644.0	136
598.0	136
604.0	136
615.0	135
616.0	134
635.0	134

592.0	133
609.0	133
591.0	133
611.0	133
563.0	133
619.0	133
572.0	130
613.0	130
606.0	129
612.0	129
610.0	128
626.0	128
586.0	126
594.0	126
585.0	124
632.0	124
595.0	123
582.0	122
584.0	122
628.0	122
618.0	122
627.0	122
597.0	121
602.0	120
636.0	117
621.0	117
617.0	111
643.0	110
608.0	107
641.0	107
692.0	107
620.0	107
622.0	106
638.0	106
640.0	105
660.0	105
648.0	104
624.0	104
659.0	103
634.0	102

658.0	100
665.0	100
647.0	97
657.0	96
639.0	96
662.0	94
625.0	93
676.0	93
655.0	92
652.0	92
637.0	91
673.0	91
623.0	91
642.0	91
629.0	91
646.0	90
666.0	90
656.0	87
630.0	86
672.0	86
649.0	85
654.0	83
633.0	83
645.0	82
650.0	82
669.0	81
670.0	80
674.0	79
689.0	76
653.0	74
651.0	73
690.0	72
663.0	70
667.0	70
671.0	70
664.0	70
677.0	69
661.0	68
695.0	68
678.0	67

699.0	67
702.0	66
700.0	66
683.0	65
631.0	64
701.0	64
691.0	64
707.0	63
703.0	61
681.0	60
687.0	60
726.0	59
696.0	58
698.0	58
721.0	57
693.0	57
694.0	57
668.0	57
679.0	55
685.0	54
684.0	54
675.0	54
727.0	54
720.0	50
709.0	50
738.0	48
688.0	48
725.0	47
710.0	47
724.0	46
682.0	46
731.0	46
697.0	46
680.0	44
737.0	43
708.0	43
706.0	42
734.0	42
719.0	42
714.0	41

713.0	41
716.0	41
718.0	40
715.0	40
705.0	39
736.0	37
704.0	36
728.0	36
732.0	36
750.0	35
686.0	35
739.0	35
735.0	34
745.0	34
733.0	34
717.0	33
723.0	33
741.0	33
762.0	32
744.0	32
729.0	32
712.0	32
767.0	31
749.0	31
753.0	31
802.0	31
746.0	30
711.0	29
759.0	29
777.0	29
748.0	29
752.0	29
758.0	28
780.0	27
730.0	27
761.0	27
788.0	26
760.0	26
751.0	25
805.0	25

757.0	25
722.0	25
756.0	25
787.0	25
769.0	24
742.0	23
755.0	23
770.0	23
786.0	23
740.0	23
747.0	23
784.0	23
795.0	21
794.0	21
806.0	21
768.0	21
775.0	21
772.0	20
801.0	20
754.0	20
831.0	20
807.0	19
776.0	19
796.0	19
773.0	18
782.0	18
797.0	18
763.0	18
743.0	17
778.0	17
823.0	16
785.0	16
808.0	16
810.0	16
771.0	16
790.0	16
765.0	16
766.0	16
1418.0	15
822.0	15

1124.0	15
798.0	15
1423.0	15
781.0	15
764.0	15
779.0	15
793.0	14
817.0	14
908.0	13
1433.0	13
1013.0	13
1080.0	13
891.0	13
789.0	12
883.0	12
804.0	12
840.0	12
830.0	11
814.0	11
1008.0	11
1434.0	11
1198.0	11
1435.0	11
999.0	10
783.0	10
893.0	10
1415.0	10
827.0	10
818.0	10
811.0	10
843.0	10
1109.0	10
1417.0	10
1042.0	10
841.0	10
914.0	9
900.0	9
856.0	9
1047.0	9
819.0	9

815.0	9
1216.0	9
1429.0	9
1061.0	9
1100.0	9
833.0	9
826.0	9
1144.0	9
1029.0	9
820.0	9
871.0	9
1427.0	9
853.0	9
1076.0	8
829.0	8
858.0	8
877.0	8
851.0	8
934.0	8
921.0	8
809.0	8
844.0	8
828.0	8
918.0	8
850.0	7
980.0	7
1437.0	7
1426.0	7
832.0	7
1052.0	7
1185.0	7
1424.0	7
1084.0	7
867.0	7
824.0	7
928.0	7
927.0	7
825.0	7
791.0	7
835.0	7

1099.0	7
1436.0	7
1095.0	7
1412.0	7
803.0	7
1017.0	7
886.0	7
998.0	7
874.0	6
897.0	6
857.0	6
1053.0	6
892.0	6
1002.0	6
859.0	6
774.0	6
1121.0	6
1065.0	6
993.0	6
971.0	6
839.0	6
1118.0	6
942.0	6
915.0	6
1015.0	6
848.0	6
1122.0	6
1035.0	6
1038.0	6
961.0	6
1188.0	6
1168.0	6
1032.0	6
1125.0	6
911.0	6
1041.0	6
910.0	6
1425.0	6
987.0	6
1007.0	6

1126.0	6
821.0	6
1101.0	6
1103.0	6
920.0	6
902.0	5
992.0	5
1000.0	5
1404.0	5
1043.0	5
1191.0	5
972.0	5
929.0	5
1422.0	5
792.0	5
1181.0	5
1187.0	5
1421.0	5
1165.0	5
1351.0	5
1069.0	5
1025.0	5
1036.0	5
876.0	5
836.0	5
849.0	5
884.0	5
1150.0	5
975.0	5
1039.0	5
885.0	5
889.0	5
1016.0	5
890.0	5
1074.0	5
852.0	5
1066.0	5
837.0	5
887.0	5
1060.0	5

966.0	5
838.0	5
1001.0	5
1054.0	5
947.0	5
1303.0	5
1057.0	4
1020.0	4
1123.0	4
1048.0	4
994.0	4
870.0	4
1192.0	4
1402.0	4
1098.0	4
865.0	4
997.0	4
881.0	4
935.0	4
973.0	4
1199.0	4
1083.0	4
1027.0	4
1169.0	4
986.0	4
909.0	4
959.0	4
958.0	4
901.0	4
834.0	4
860.0	4
1407.0	4
1151.0	4
923.0	4
899.0	4
955.0	4
1009.0	4
1004.0	4
1428.0	4
1297.0	4

905.0	4
847.0	4
1128.0	4
1283.0	4
1416.0	4
943.0	4
1360.0	4
1405.0	4
962.0	4
888.0	4
1390.0	4
1411.0	4
1062.0	4
1166.0	4
960.0	4
799.0	4
912.0	4
898.0	4
907.0	4
948.0	4
1387.0	4
1323.0	4
868.0	4
1114.0	4
1141.0	4
1228.0	4
916.0	4
945.0	4
1310.0	4
996.0	4
1182.0	4
1079.0	4
1324.0	4
1073.0	4
1028.0	3
906.0	3
1274.0	3
1137.0	3
1250.0	3
1155.0	3

1244.0	3
1203.0	3
1026.0	3
949.0	3
816.0	3
1413.0	3
866.0	3
846.0	3
1130.0	3
1096.0	3
800.0	3
1012.0	3
1264.0	3
1334.0	3
880.0	3
1111.0	3
813.0	3
1406.0	3
1420.0	3
1269.0	3
1223.0	3
1033.0	3
1070.0	3
1209.0	3
930.0	3
1106.0	3
1072.0	3
1102.0	3
924.0	3
894.0	3
1113.0	3
872.0	3
1394.0	3
913.0	3
1003.0	3
1200.0	3
1133.0	3
1108.0	3
855.0	3
1034.0	3

963.0	3
1409.0	3
1302.0	3
1156.0	3
882.0	3
1024.0	3
1190.0	3
1381.0	3
1147.0	3
1030.0	3
950.0	3
878.0	3
845.0	3
1131.0	3
1037.0	3
941.0	3
869.0	3
1367.0	3
1419.0	3
903.0	3
965.0	3
862.0	3
875.0	3
953.0	3
1347.0	3
1120.0	3
854.0	3
1271.0	2
1071.0	2
946.0	2
1161.0	2
1225.0	2
1059.0	2
1255.0	2
1148.0	2
1389.0	2
939.0	2
1197.0	2
933.0	2
1376.0	2

1340.0	2
863.0	2
1193.0	2
974.0	2
956.0	2
1378.0	2
984.0	2
1116.0	2
1064.0	2
1068.0	2
1401.0	2
964.0	2
1050.0	2
1438.0	2
926.0	2
1430.0	2
1329.0	2
1091.0	2
1204.0	2
1307.0	2
1230.0	2
1259.0	2
922.0	2
1046.0	2
1300.0	2
1006.0	2
1229.0	2
1152.0	2
1031.0	2
1093.0	2
1400.0	2
1388.0	2
1179.0	2
1138.0	2
937.0	2
1132.0	2
970.0	2
1294.0	2
1333.0	2
1439.0	2

1308.0	2
895.0	2
1089.0	2
1385.0	2
978.0	2
1221.0	2
1075.0	2
1011.0	2
1330.0	2
1377.0	2
1063.0	2
879.0	2
1077.0	2
1167.0	2
967.0	2
1112.0	2
1149.0	2
1251.0	2
1396.0	2
1258.0	2
979.0	2
1432.0	2
1010.0	2
1056.0	2
1005.0	2
1178.0	2
1206.0	2
1313.0	2
1085.0	2
904.0	2
931.0	2
977.0	2
1301.0	2
925.0	2
1170.0	2
1040.0	2
896.0	2
954.0	2
1276.0	2
1014.0	2

1403.0	2
1296.0	2
1222.0	2
1049.0	1
1210.0	1
1142.0	1
1350.0	1
1215.0	1
1267.0	1
1127.0	1
990.0	1
1311.0	1
940.0	1
1088.0	1
1356.0	1
1263.0	1
1117.0	1
1224.0	1
1058.0	1
1344.0	1
1180.0	1
1326.0	1
1153.0	1
1392.0	1
864.0	1
1284.0	1
1134.0	1
1242.0	1
1163.0	1
1341.0	1
1366.0	1
1145.0	1
1299.0	1
944.0	1
812.0	1
988.0	1
1321.0	1
1194.0	1
1261.0	1
1174.0	1

981.0	1
936.0	1
1045.0	1
1051.0	1
1306.0	1
982.0	1
1245.0	1
1022.0	1
1372.0	1
861.0	1
932.0	1
1195.0	1
1335.0	1
957.0	1
1136.0	1
1247.0	1
1023.0	1
1082.0	1
1211.0	1
1183.0	1
1184.0	1
1172.0	1
1391.0	1
1373.0	1
1055.0	1
1277.0	1
1097.0	1
1233.0	1
969.0	1
1021.0	1
1160.0	1
985.0	1
1154.0	1
976.0	1
1352.0	1
1090.0	1
1393.0	1
1399.0	1
1176.0	1
1395.0	1

1290.0	1
919.0	1
1235.0	1
1226.0	1
1278.0	1
842.0	1
1129.0	1
1196.0	1
1282.0	1
1382.0	1
1234.0	1
1249.0	1
1078.0	1
1143.0	1
1266.0	1
1332.0	1
1281.0	1
1018.0	1
1346.0	1
1252.0	1
873.0	1
1092.0	1
1207.0	1
1186.0	1
1345.0	1
1164.0	1
951.0	1
1260.0	1
1275.0	1
1339.0	1
1386.0	1
1159.0	1
1232.0	1
1343.0	1

Name: MIN_TEMPORAL_GAP, dtype: int64

```
In [ ]: # How many rows have a minimum temporal gap equal to NaN?  
df[df['MIN_TEMPORAL_GAP'].isna()].shape[0]
```

```
Out[ ]: 4196926
```



```
In [ ]: # Cleaning operation: remove the rows using the minimum temporal gap

# Find a reasonable delta of MIN_TEMPORAL_GAP to remove the rows that have a minimum temporal gap for the same series
# Print the minimum value of the column MIN_TEMPORAL_GAP
print('The minimum value of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].min()))

# Print the maximum value of the column MIN_TEMPORAL_GAP
print('The maximum value of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].max()))

# Print the mean value of the column MIN_TEMPORAL_GAP
print('The mean value of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].mean()))

# Print the median value of the column MIN_TEMPORAL_GAP
print('The median value of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].median()))

# Print the standard deviation of the column MIN_TEMPORAL_GAP
print('The standard deviation of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].std()))

# Print the 0.05th percentile of the column MIN_TEMPORAL_GAP
print('The 0.05th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.05)))

# Print the 0.10th percentile of the column MIN_TEMPORAL_GAP
print('The 0.10th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.10)))

# Print the 25th percentile of the column MIN_TEMPORAL_GAP
print('The 25th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.25)))

# Print the 75th percentile of the column MIN_TEMPORAL_GAP
print('The 75th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.75)))

# Print the 90th percentile of the column MIN_TEMPORAL_GAP
print('The 90th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.90)))

# Print the 95th percentile of the column MIN_TEMPORAL_GAP
print('The 95th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.95)))

# Print the 99th percentile of the column MIN_TEMPORAL_GAP
print('The 99th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.99)))

# Print the 99.9th percentile of the column MIN_TEMPORAL_GAP
```

```

print('The 99.9th percentile of the column MIN_TEMPORAL_GAP is: {}'.format(df['MIN_TEMPORAL_GAP'].quantile(0.999)))

# Decide the delta of MIN_TEMPORAL_GAP using the 25th percentile of the column MIN_TEMPORAL_G
delta = df['MIN_TEMPORAL_GAP'].quantile(0.1)
if delta == 0:
    delta = df['MIN_TEMPORAL_GAP'].quantile(0.25)
if delta == 0:
    delta = df['MIN_TEMPORAL_GAP'].median()
print('The delta of MIN_TEMPORAL_GAP is: {}'.format(delta))

```

```

The minimum value of the column MIN_TEMPORAL_GAP is: 0.0
The maximum value of the column MIN_TEMPORAL_GAP is: 1439.0
The mean value of the column MIN_TEMPORAL_GAP is: 108.73571743144305
The median value of the column MIN_TEMPORAL_GAP is: 5.0
The standard deviation of the column MIN_TEMPORAL_GAP is: 192.22288839487143
The 0.05th percentile of the column MIN_TEMPORAL_GAP is: 0.0
The 0.10th percentile of the column MIN_TEMPORAL_GAP is: 0.0
The 25th percentile of the column MIN_TEMPORAL_GAP is: 0.0
The 75th percentile of the column MIN_TEMPORAL_GAP is: 155.0
The 90th percentile of the column MIN_TEMPORAL_GAP is: 431.0
The 95th percentile of the column MIN_TEMPORAL_GAP is: 540.0
The 99th percentile of the column MIN_TEMPORAL_GAP is: 706.0
The 99.9th percentile of the column MIN_TEMPORAL_GAP is: 1182.0
The delta of MIN_TEMPORAL_GAP is: 5.0

```

```

In [ ]: # Cleaning operation: remove the rows using the minimum temporal gap

# Save the number of rows before the cleaning operation
shape_before = df.shape[0]

# Delete the rows that have a minimum temporal gap for the same serial and fermata more than the delta calculated b
# Do not remove the rows with NaN values because they are the first validations of the day of a specific serial and
df = df[(df['MIN_TEMPORAL_GAP'] > delta) | (df['MIN_TEMPORAL_GAP'].isna())]

# Print the number of rows before and after the cleaning operation and the difference
print('The number of rows before the cleaning operation is: {}'.format(shape_before))
print('The number of rows after the cleaning operation is: {}'.format(df.shape[0]))
print('The difference is: {}'.format(shape_before - df.shape[0]))
# Calculate the percentage of rows that has just been deleted
print('The percentage of rows that has just been deleted is: {}'.format(round((shape_before - df.shape[0])/shape_b

```

The number of rows before the cleaning operation is: 4680575
The number of rows after the cleaning operation is: 4427561
The difference is: 253014
The percentage of rows that has just been deleted is: 5.41%

```
In [ ]: # Delete the column MIN_TEMPORAL_GAP because it is not useful anymore
df.drop('MIN_TEMPORAL_GAP', axis=1, inplace=True)
```

```
In [ ]: # Create a new dataframe, copied from the original one
df_new = df.copy()

# Update the column 'DESCRIZIONE' of the new df with the new values of the dictionary:
# the value that are present in the dataframe are the values of the dictionary; you have to substitute with the key
for key, value in dict_prefix.items():
    df_new['DESCRIZIONE'] = df_new['DESCRIZIONE'].replace(value, key)

# Print the head of the new dataframe
print(df_new.head())

# Export the new dataframe in a txt file
# The name of the file is dataset_cleaned followed by the name (file_name variable) of the file that has been cleaned
name_file = 'dataset_cleaned_' + file_name.split('.')[0] + '.txt'
df_new.to_csv('data/processed/' + name_file, sep='\t', index=False)

print('The script has finished')
```

	DATA	ORA	DATA_VALIDAZIONE	SERIALE	FERMATA	\
0	2022-05-13	00:00:00	2022-05-13	65676291870913797	5089	
1	2022-05-13	00:00:00	2022-05-13	36141384536591364	5032	
2	2022-05-13	00:00:00	2022-05-13	36144856606063108	5031	
3	2022-05-13	00:00:00	2022-05-13	36144856474364932	506	
4	2022-05-13	00:00:00	2022-05-13	36144856606062852	5031	

	DESCRIZIONE	TITOLO	TICKET_CODE	DESCRIZIONE_TITOLO
0	FERROVIA	11149	4	7GG-TPL 43,60-COMVE16,40
1	FERROVIA	11107	2	48H-TPL 24,90-COMVE5,10
2	P.LE ROMA	11108	3	72H-TPL 33,40-COMVE6,60
3	VENEZIA	11261	1	DAILY PASS VENEZIA - AVM
4	P.LE ROMA	11108	3	72H-TPL 33,40-COMVE6,60

The script has finished