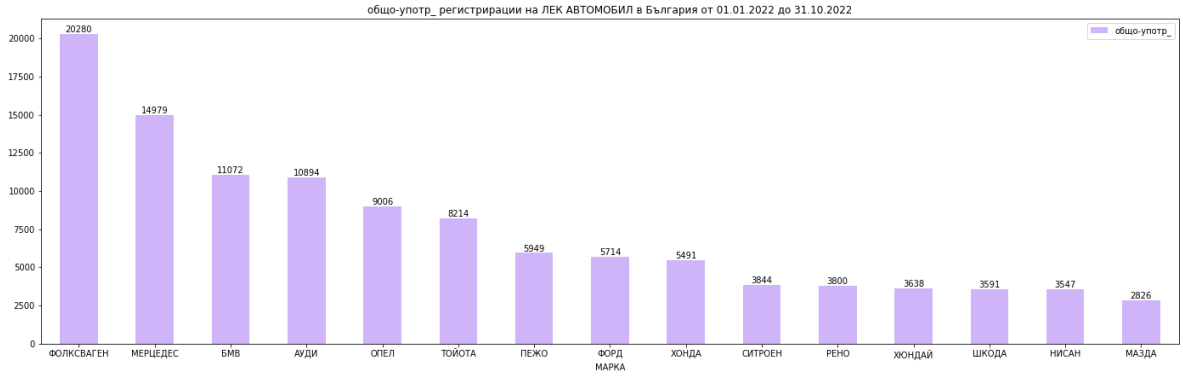
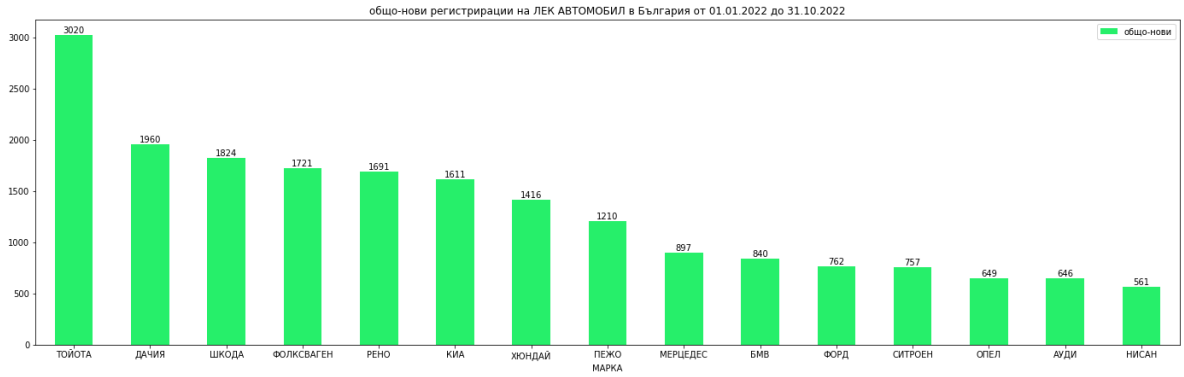
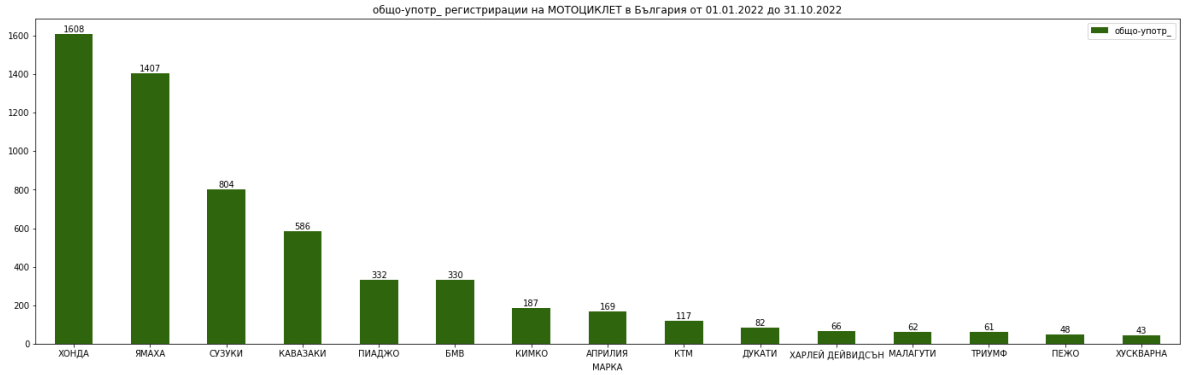
```
"""
This notebook is demo ETL job that:
extraxts data from Bulgarian government data for registered
vehicles(01.01.2022 to 39.10.2022),
transforms da data and saves/visualizes desired vehicle type and age total
number of registrations.
"""

from pyspark.sql.functions import col, sum
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np

def extract_from_blob(path):
    """
    Extracting data from mounted blob storage in Azure

    args: path to csv file

    returns: DataFrame
    """

    df = spark.read.csv(path, inferSchema=True, header=True)
    return df


def transform_replace_dots_in_schema(df):
    """
    This transformation function is replacing "." with "_" in col names

    args: DataFrame with "."(dot) in column names

    returns: DataFrame with "."(dot) replaced with "_"(dash) in column names
    """

    clean_df = df.toDF(*(c.replace('.', '_') for c in df.columns))
    return clean_df


def transform_get_top_15_registrations_data(df, col_type, col_age):
    """
    This transformation function is filtering DF for desired vehicles, and
desired age (brand-new or used)
    and it is ranking the 15 most registered used or brand_new models in
Bulgaria

    args: DataFrame with all registered car brands
    args: col_type type of desired vehicles
    args: col_age age of desired vehicles brand-new or used
```

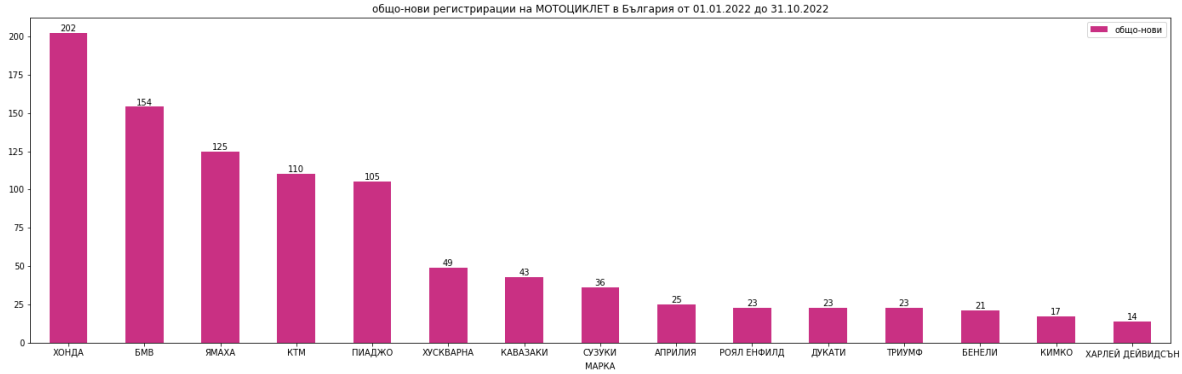
```
    returns: DataFrame 15 most registered models in Bulgaria
    """

    df_new_auto = (df
                   .filter(col("ВИД МПС") == col_type)
                   .select(col("ВИД МПС"), col("МАРКА"), col(col_age))
                   .groupby(col("МАРКА"))
                   .agg(sum(col(col_age)).alias(col_age))
                   .orderBy(col(col_age).desc())
                   .limit(15)
                   )

    return df_new_auto

def visualize(df, col_type, col_age):

    """
    This visualization function displaying desired vehicle type and age
registrations

    args: DataFrame transformed for visualization top 15 most registered
vehicles in Bulgaria
    args: col_type type of desired vehicles
    args: col_age age of desired vehicles brand-new or used
    """

    df1 = df.toPandas()
    color = (np.random.random(), np.random.random(), np.random.random())
    ax = df1.plot(kind='bar',x='МАРКА',y=col_age, color = color, figsize=
(24, 7), rot=0)
    plt.title(f'{col_age} регистрации на {col_type} в България от
01.01.2022 до 31.10.2022')
    for container in ax.containers:
        ax.bar_label(container)



class VehicleSales():
    """
    Vehicle calss etl class that directs the etl process

    args: path to csv table
    args: col_type type of desired vehicles
    args: col_age age of desired vehicles brand-new or used

    returns: visualization of top 15 most registered vehicles of desired
type and desired age
    """

    def __init__(self, path, col_type, col_age):
        self.path = path
```

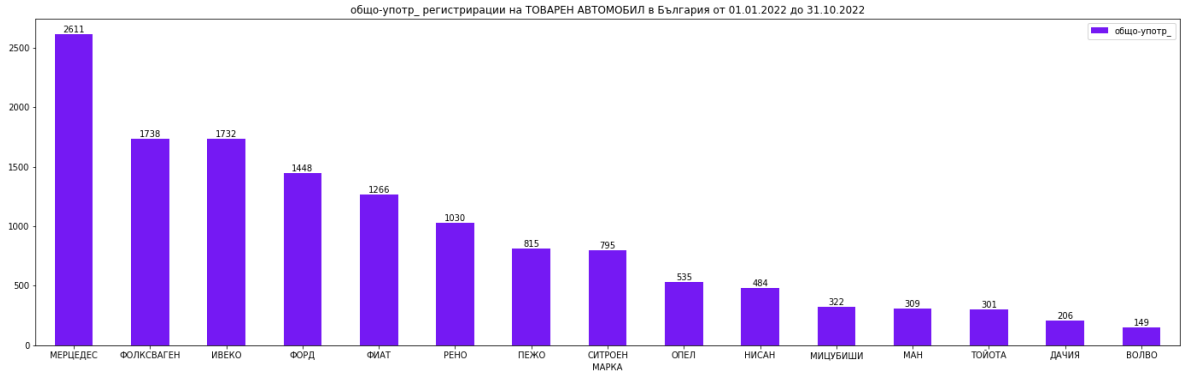
```
        self.col_type = col_type
        self.col_age = col_age

    def run(self):
        """
        This function runs all etl functions in the class
        """

        df1 = self.extract_data()
        df_updated = self.transform_data(df1)
        df_load = self.load_visualizations(df_updated)

    def extract_data(self):
        """
        Callint the extraxt function with provided path

        returns: DataFrame
        """

        df = extract_from_blob(self.path)
        return df

    def transform_data(self, df):
        """
        Calling the transformation function

        args: DataFrame

        returns: DataFrame transfored and ready for visualization witf
provided desired parameters
        """

        valid_df = transform_replace_dots_in_schema(df)
        df_updated = transform_get_top_15_registrations_data(valid_df,
self.col_type, self.col_age)
        return df_updated

    def load_visualizations(self, df):
        """
        Calling the Visualization function with desired parameters
        """

        visualize(df, self.col_type, self.col_age)
```

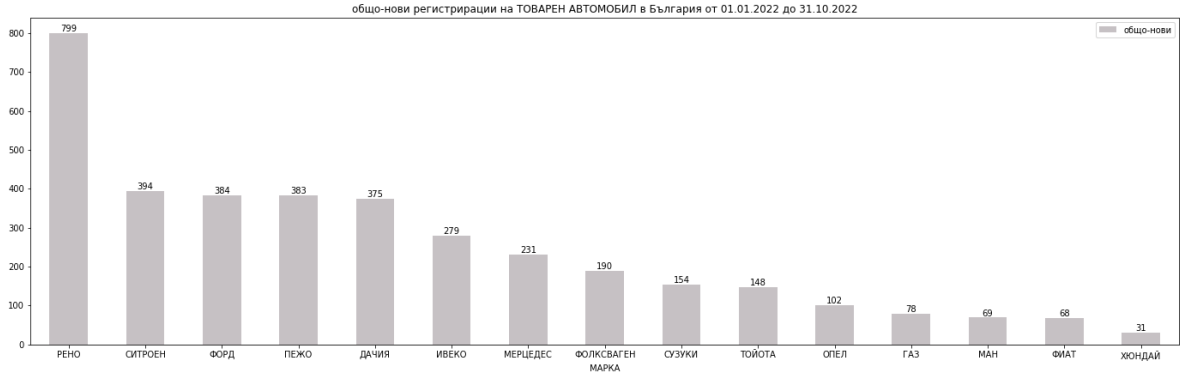

общо-употр_ регистрации на ЛЕК АВТОМОБИЛ в България от 01.01.2022 до 31.10.2022
общо-употр_
20280
14979
11072
10894
9006
8214
5949
5714
5491
3844
3800
3638
3591
3547
2826
ФОЛКСВАГЕН
МЕРЦЕДЕС
БМВ
АУДИ
ОПЕЛ
ТОЙОТА
ПЕЖО
ФОРД
ХОНДА
СИТРОЕН
РЕНО
ХЮНДАЙ
ШКОДА
НИСАН
МАЗДА
МАРКА


общо-нови регистрации на ЛЕК АВТОМОБИЛ в България от 01.01.2022 до 31.10.2022
общо-нови
3020
1960
1824
1721
1691
1611
1416
1210
897
840
762
757
649
646
561
ТОЙОТА
ДАЧИЯ
ШКОДА
ФОЛКСВАГЕН
РЕНО
КИА
ХЮНДАЙ
ПЕЖО
МЕРЦЕДЕС
БМВ
ФОРД
СИТРОЕН
ОПЕЛ
АУДИ
НИСАН
МАРКА


общо-употр_ регистрации на МОТОЦИКЛЕТ в България от 01.01.2022 до 31.10.2022
общо-употр_
1608
1407
804
586
332
330
187
169
117
82
66
62
61
48
43
ХОНДА
ЯМАХА
СУЗУКИ
КАВАЗАКИ
ПИАДЖО
БМВ
КИМКО
АПРИЛИЯ
КТМ
ДУКАТИ
ХАРЛЕЙ ДЕЙВИДСЪН
МАЛАГУТИ
ТРИУМФ
ПЕЖО
ХУСКВАРНА
МАРКА


общо-нови регистрации на МОТОЦИКЛЕТ в България от 01.01.2022 до 31.10.2022
общо-нови
202
154
125
110
105
49
43
36
25
23
23
23
21
17
14
ХОНДА
БМВ
ЯМАХА
КТМ
ПИАДЖО
ХУСКВАРНА
КАВАЗАКИ
СУЗУКИ
АПРИЛИЯ
РОЯЛ ЕНФИЛД
ДУКАТИ
ТРИУМФ
БЕНЕЛИ
КИМКО
ХАРЛЕЙ ДЕЙВИДСЪН
МАРКА


общо-употр_ регистрирации на ТОВАРЕН АВТОМОБИЛ в България от 01.01.2022 до 31.10.2022
общо-употр_
2611
1738
1732
1448
1266
1030
815
795
535
484
322
309
301
206
149
МЕРЦЕДЕС
ФОЛКСВАГЕН
ИВЕКО
ФОРД
ФИАТ
РЕНО
ПЕЖО
СИТРОЕН
ОПЕЛ
НИСАН
МИЦУБИШИ
МАН
ТОЙОТА
ДАЧИЯ
ВОЛВО
МАРКА


общо-нови регистрирации на ТОВАРЕН АВТОМОБИЛ в България от 01.01.2022 до 31.10.2022
общо-нови
799
394
384
383
375
279
231
190
154
148
102
78
69
68
31
РЕНО
СИТРОЕН
ФОРД
ПЕЖО
ДАЧИЯ
ИВЕКО
МЕРЦЕДЕС
ФОЛКСВАГЕН
СУЗУКИ
ТОЙОТА
ОПЕЛ
ГАЗ
МАН
ФИАТ
ХЮНДАЙ
МАРКА