

PT

Yvo Hu s2962802


February 2021

1

1.1

This script only takes the license plates from the data set and filters it on "Personenauto" then all numbers are changed to a 9, and all letters are changed to an x. Next the output is sorted, and all duplicates are removed.

The output will be a list of license plate configurations



```
9999XX
99XX99
99XXX9
99XXXX
9XXX99
XX9999
XX999X
XX99XX
XXXX99
```

1.2

This script only takes the car brand and model from the data set and filters it on "Personenauto", then its output is sorted and every unique entry will be counted. Then the top 10 with the most entries are kept and the rest are discarded in the output. Finally we remove the count numbers and only display the car brand and model.

The output will be a list of the top 10 most popular car brand and model combinations.

```
VOLKSWAGEN GOLF
VOLKSWAGEN POLO
FORD FOCUS
RENAULT CLIO
FORD FIESTA
TOYOTA TOYOTA
RENAULT TWINGO
PEUGEOT 206
KIA PICANTO
RENAULT MEGANE
```

1.3

This script does the same thing as script 2, except only car brand, model combinations which are registered as taxi's are taken into consideration.

The output will be a list of the top 10 most popular car brand and model combinations which are registered as taxis.

```
MERCEDES-BENZ,SPRINTER  
VOLKSWAGEN,KOMBI  
MERCEDES-BENZ,E 200 CDI  
MERCEDES-BENZ,VITO TOURER  
VOLKSWAGEN,CRAFTER  
VOLKSWAGEN,CADDY  
MERCEDES-BENZ,E 220 CDI  
FORD,TRANSIT/TOURNEO  
MERCEDES-BENZ,VITO  
VOLKSWAGEN,PASSAT
```

1.4

This script takes the car brand from the data set and filters it on "Personenauto". It then counts the entries and removes the duplicates, and filters it on brands with more than 100000 occurrences. These brands are then string matched with the original data set to get all lines which contain those strings in column 3 (the brand column) and are of type "Personenauto". Then it is filtered again on brand, but now too with the date of admission. The date is filtered on numbers higher than 0 to account for bad data, and shortened to only contain the year. The average age of each entry is calculated using an awk program as seen in the source code.

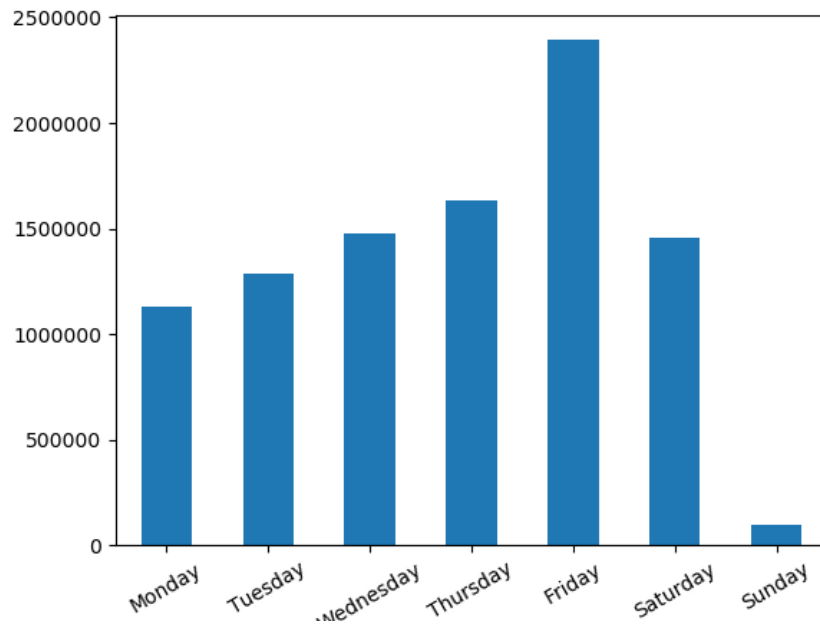
The output will be a list of car brands with more than 100000 entries who are on average the youngest.

```
9 KIA
9 SKODA
11 AUDI
11 HYUNDAI
12 MITSUBISHI
12 NISSAN
12 RENAULT
12 SEAT
13 BMW
13 FORD
13 PEUGEOT
13 SUZUKI
13 TOYOTA
14 CITROEN
14 FIAT
14 MAZDA
14 OPEL
14 VOLKSWAGEN
14 VOLVO
16 MERCEDES-BENZ
```

1.5

This script takes the date of ascription from the data set and filters it on "Personenauto". It then only takes the date of ascription and calculates it using the Zeller's congruence algorithm written in an awk program. The output of the algorithm is in decimal so it is then converted into weekdays and counted. The weekdays and corresponding count are then plotted using matplotlib and pandas.

The output will be a bar chart of weekdays plotted against the number of ascriptions. The least common day is sunday because perhaps the places to get ascribed to are open for a limited amount of hours on sunday or not at all.



1.6

This script checks if the mass per kg and price are valid numbers, and takes the brand from each entry which satisfy those conditions. It then sorts and counts each unique brand entry and filters them on brands with more than 10000 occurrences. These brands are then string matched with the original data set to get all lines which contain those strings in column 3 (the brand column). It then takes the mass per kg and price, and calculates the average for each unique brand entry using an awk program as seen in the source code. Finally it outputs the brand, average mass per kg and price and plots it using matplotlib and pandas.

The output will be a scatterplot of mass per kg plotted against the price. There seems to be a logarithmic correlation between those two variables. Which likely corresponds to luxury cars having less utility that justifies their price than average cars.

