Script

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v readr   2.1.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(skimr)
library(naniar)
```

```
##
## Attaching package: 'naniar'
```

```
## The following object is masked from 'package:skimr':
##
##     n_complete
```

```
library(dlookr)
```

```
##
## Attaching package: 'dlookr'
```

```
## The following object is masked from 'package:Hmisc':
##
##     describe
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
## The following object is masked from 'package:base':
##
##     transform
```

```
library(visdat)
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:Hmisc':
##
##     subplot
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(reticulate)
library(dataPreparation)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loading required package: progress
```

```
## dataPreparation 1.0.4
```

```
## Type data_preparation_news() to see new features/changes/bug fixes.
```

```
# Import data
data<- read_csv(file.choose(),quote = "\"",
                col_types = cols(
                  .default = col_character()))

# Observations with wrong columns
error_cases <- data[grep("Angebot", data$seller),1:(ncol(data)-1)]

colnames(error_cases)[3:ncol(error_cases)] <- colnames(data)[4:ncol(data)]
error_cases$seller <- "privat"

# Corrected data frame
data <- rbind.data.frame(data[-(grep("Angebot", data$seller)),], error_cases)

char_vars <- c("name", "seller", "offertype",   "abtest", "vehicletype", "gearbox", "model", "fueltype", "brand",
               "notrepaireddamage", "nrofpictures", "postalcode")

num_vars <- c("price", "yearofregistration","powerps","kilometer","monthofregistration")
data[num_vars] <- lapply(data[num_vars], function(x) as.numeric(as.character(x)))

date_vars <- c("datecrawled", "datecreated","lastseen")
data[date_vars] <- lapply(data[date_vars], function(x) as.Date(as.character(x)))

# Clean extra vars
rm(error_cases, char_vars, date_vars, num_vars)

# Replacing 0 to NA
data[data == 0] <- NA
```

```
# load packages
import pandas as pd
from pandas import read_csv
from sklearn.model_selection import train_test_split
from datetime import datetime
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.neighbors import LocalOutlierFactor
# load the dataset
data = pd.DataFrame(r.data)
# Categorical var
data["seller"].value_counts()
```

```
## privat        199997
## gewerblich         3
## Name: seller, dtype: int64
```

```
data["seller"] = pd.Categorical(data["seller"])
data["seller"].dtype
# string var
```

```
## CategoricalDtype(categories=['gewerblich', 'privat'], ordered=False)
```

```
for col in ["name", "seller", "offertype",  "abtest", "vehicletype", "gearbox", "model", "fueltype", "brand", "notrepairedda
mage", "nrofpictures", "postalcode"]: data[col] = data[col].astype('str')
# datetime var
data[["datecrawled", "datecreated","lastseen"]] = data[["datecrawled", "datecreated","lastseen"]].apply(pd.to_datetime, form
at='%Y-%m-%d')
# second df for fill NA
df = data[["price", "yearofregistration","powerps","kilometer","monthofregistration"]]
df = df.replace(np.nan, 0)
df = df.values
# Fill NA
imp = SimpleImputer(missing_values=0, strategy='mean')
imp.fit(df)
```

▾ SimpleImputer

SimpleImputer(missing_values=0)

```
imp.statistics_
```

```
## array([2.33437313e+04, 2.00456407e+03, 1.29925034e+02, 1.25530300e+05,
##        6.37749627e+00])
```

```
df = imp.transform(df)
df = pd.DataFrame(df)
df.columns = ["price", "yearofregistration","powerps","kilometer","monthofregistration"]
for col in ["price", "powerps", "monthofregistration"]: df[col] = df[col].astype('int')
# Replace columns in original df
columns=["price", "yearofregistration","powerps","kilometer","monthofregistration"]
data[columns] = df[columns]

del(df)
```

```r
# load data
data <- py$data

# defining vars
char_vars <- c("name", "seller", "offertype",   "abtest", "vehicletype", "gearbox", "model", "fueltype", "brand",
               "notrepaireddamage", "nrofpictures", "postalcode")

num_vars <- c("price", "yearofregistration","powerps","kilometer","monthofregistration")
data[num_vars] <- lapply(data[num_vars], function(x) as.numeric(as.character(x)))

date_vars <- c("datecrawled", "datecreated","lastseen")
data[date_vars] <- lapply(data[date_vars], function(x) as.Date(as.character(x)))

# Replacing NA
data <- data %>%
  mutate_if(.predicate=is.character, .funs=~na_if(.,"NA"))

# Clean vars in environment
rm(char_vars, date_vars, num_vars)

# desribe data
skim(data)
```

Data summary

Name	data
Number of rows	200000
Number of columns	20
_____	
Column type frequency:	
character	12
Date	3
numeric	5

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
name	0	1.00	4	327	45	0	135028
seller	0	1.00	6	10	0	2	0
offertype	0	1.00	6	7	0	2	0
abtest	0	1.00	4	7	0	2	0
vehicletype	20378	0.90	3	10	0	8	0
gearbox	10849	0.95	6	9	0	2	0
model	11117	0.94	2	18	0	249	0
fueltype	18083	0.91	3	7	0	7	0
brand	0	1.00	3	14	0	40	0
notrepaireddamage	38851	0.81	2	3	0	2	0
nrofpictures	0	1.00	5	5	0	1	0
postalcode	0	1.00	4	5	0	8025	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
datecrawled	0	1	2016-03-04	2016-04-06	2016-03-20	34
datecreated	0	1	2014-03-09	2016-04-06	2016-03-20	106
lastseen	0	1	2016-03-04	2016-04-06	2016-04-03	34

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
price	0	1	23343.71	485320	8.88	1	1300	3299	80002147483647	<U+2587><U+2581><U+2581><U+2581><U+2581><U+2581>
yearofregistration	0	1	2004.56	93.60	1000	1999	2003	2008	9999	<U+2587><U+2581><U+2581><U+2581><U+2581><U+2581>
powerps	0	1	129.82	189.03	1	87	122	150	19211	<U+2587><U+2581><U+2581><U+2581><U+2581><U+2581>
kilometer	0	1	125530.30	40151.15	5000	100000	150000	150000	150000	<U+2581><U+2581><U+2581><U+2581><U+2581><U+2587>
monthofregistration	0	1	6.34	3.17	1	4	6	9	12	<U+2586><U+2585><U+2587><U+2583><U+2586>