**Supervised Learning Regression – Course Project**

**IBM Machine Learning**

Author: Isidro Brevers Gómez

E-mail: ibrevers@gmail.com

## 1. Objective and brief description of the data set and a summary of its attributes

The data set used contains information about traffic accidents in Madrid, Spain in 2018, which is published by Madrid City Council at https://datos.madrid.es/portal/site/egob/ in CSV format.

The objective of the analysis will be developing a model that allows predicting the number of casualties resulting from accidents, based on certain parameters such as the day of the week, the hour, weather and road conditions, districts, type of accident and persons involved, and other.

The data set consists on a register for every person involved in the accident, including (non-useful data disregarded through cleansing work is omitted, full detail of the data set can be found at https://datos.madrid.es/FWProjects/egob/Catalogo/Seguridad/Ficheros/Estructura_DS_Accidentes_trafico_2010_2018.pdf):

- *Fecha* [object]: accident date.
- *Rango horario* [object]: hourly range when the accident took place.
- *Día semana* object [object]: day of the week when the accident took place.
- *Distrito* [object]: district where the accident took place.
- *CPFA Granizo* [object]: weather conditions hail.
- *CPFA Hielo* [object]: weather conditions ice.
- *CPFA Lluvia* [object]: weather conditions rain.
- *CPFA Niebla* [object]: weather conditions fog.
- *CPFA Seco* [object]: weather conditions dry.
- *CPFA Nieve* [object]: weather conditions snow.
- *CPSV Mojada* [object]: road conditions wet.
- *CPSV Aceite* [object]: road conditions oil.
- *CPSV Barro* [object]: road conditions mud.
- *CPSV Grava Suelta* [object]: road conditions stones.
- *CPSV Hielo* [object]: road conditions ice.
- *CPSV Seca Y Limpia* [object]: road conditions dry and clean.
- *Victimas* [int64]: number of casualties in the accident.
- *TIPO ACCIDENTE* [object]: type of accident
- *Tipo Vehiculo* [object]: type of vehicle involved in the accident.
- *TIPO PERSONA* [object]: type of person involved in the accident.
- *SEXO* [object]: sex of person involved in the accident.
- *LESIVIDAD* [object]: damage degree.
- *Tramo Edad* [object]: age range of the person involved in the accident.

## 2. Initial plan for data exploration

As a first step, the CSV file was downloaded and read into a dataframe for further exploration and analysis, resulting into the following (first 5 results):

| | FECHA | RANGO HORARIO | DIA SEMANA | DISTRITO | LUGAR ACCIDENTE | Nº | Nº PARTE | CPFA Granizo | CPFA Hielo | CPFA Lluvia | ... | CPSV Grava Suelta | CPSV Hielo | CPSV Seca Y Limpia | * Nº VICTIMAS | TIPO ACCIDENTE | Tipo Vehiculo | TIPO PERSONA | SEXO | LESIVIDAD | Tramo Edad |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | CALLE DE SAN BASILIO - CALLE DEL CRISTO DE LA ... | 0 | 2018/1 | NO | NO | NO | ... | NO | NO | SI | 1 | ATROPELLO | NO ASIGNADO | PEATON | HOMBRE | HG | DE 15 A 17 AÑOS |
| 1 | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | CALLE DE SAN BASILIO - CALLE DEL CRISTO DE LA ... | 0 | 2018/1 | NO | NO | NO | ... | NO | NO | SI | 1 | ATROPELLO | NO ASIGNADO | TESTIGO | HOMBRE | IL | DE 30 A 34 ANOS |
| 2 | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | CALLE DE SAN BASILIO - CALLE DEL CRISTO DE LA ... | 0 | 2018/1 | NO | NO | NO | ... | NO | NO | SI | 1 | ATROPELLO | TURISMO | CONDUCTOR | HOMBRE | IL | DE 35 A 39 AÑOS |
| 3 | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | AVENIDA DE FRANCISCO PI Y MARGALL - AVENIDA DE... | | 2018/3 | NO | NO | NO | ... | NO | NO | SI | 1 | CHOQUE CON OBJETO FIJO | NO ASIGNADO | TESTIGO | HOMBRE | IL | DE 21 A 24 AÑOS |
| 4 | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | AVENIDA DE FRANCISCO PI Y MARGALL - AVENIDA DE... | | 2018/3 | NO | NO | NO | ... | NO | NO | SI | 1 | CHOQUE CON OBJETO FIJO | NO ASIGNADO | TESTIGO | MUJER | IL | DE 40 A 44 AÑOS |

The above represents al the traffic accidents occurred in Madrid, Spain in 2018, including information about whether and road conditions, when and where the accident took place, the persons and type of vehicles involved and other.

Then, some initial checks were done in order to verify:

- The size of the data: 30,122 rows, 26 columns.
- Data does not include null or inconsistent values.
- Data types.
- Basic statistical attributes of the data.

```
Data columns (total 23 columns):
FECHA                 26463 non-null object
RANGO HORARIO         26463 non-null object
DIA SEMANA            26463 non-null object
DISTRITO              26463 non-null object
CPFA Granizo          26463 non-null object
CPFA Hielo            26463 non-null object
CPFA Lluvia           26463 non-null object
CPFA Niebla           26463 non-null object
CPFA Seco             26463 non-null object
CPFA Nieve            26463 non-null object
CPSV Mojada           26463 non-null object
CPSV Aceite           26463 non-null object
CPSV Barro            26463 non-null object
CPSV Grava Suelta     26463 non-null object
CPSV Hielo            26463 non-null object
CPSV Seca Y Limpia    26463 non-null object
Victimas              26463 non-null int64
TIPO ACCIDENTE        26463 non-null object
Tipo Vehiculo         26463 non-null object
TIPO PERSONA          26463 non-null object
SEXO                  26463 non-null object
LESIVIDAD             26463 non-null object
Tramo Edad            26463 non-null object
dtypes: int64(1), object(22)
```

| | Victimas |
|---|---|
| count | 26463.000000 |
| mean | 1.488115 |
| std | 1.278060 |
| min | 1.000000 |
| 25% | 1.000000 |
| 50% | 1.000000 |
| 75% | 2.000000 |
| max | 19.000000 |

## 3. Actions taken for data cleansing and feature engineering

Once data was read, some data cleaning was necessary in order to ensure a better understanding and further analysis of the information.

- Some columns were dropped and renamed.
- It was check that the data set does not include null or missing values.
- Data where TIPO PERSONA was TESTIGO was dropped, as witness were not involved in the accident thus such information is not useful.

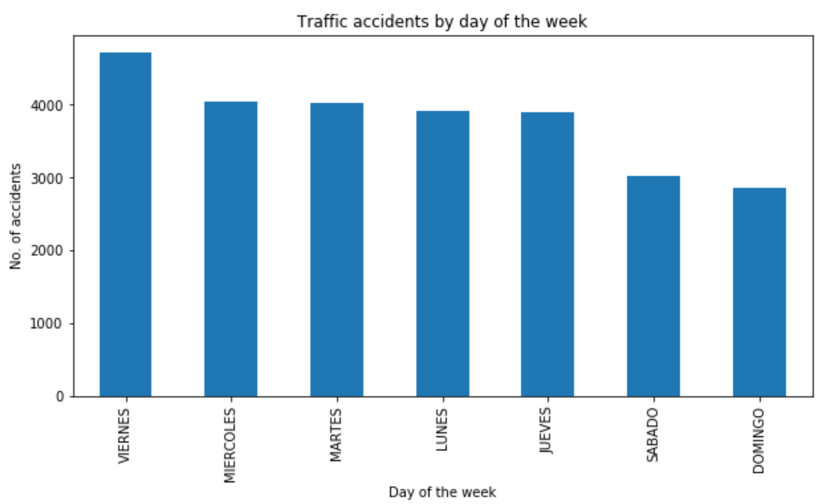
| | FECHA | RANGO HORARIO | DIA SEMANA | DISTRITO | CPFA Granizo | CPFA Hielo | CPFA Lluvia | CPFA Niebla | CPFA Seco | CPFA Nieve | ... | CPSV Grava Suelta | CPSV Hielo | CPSV Seca Y Limpia | Victimas | TIPO ACCIDENTE | Tipo Vehiculo | TIPO PERSONA | SEXO | LESIVIDAD | Tramo Edad |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | NO | NO | NO | NO | SI | NO | ... | NO | NO | SI | 1 | ATROPELLO | NO ASIGNADO | PEATON | HOMBRE | HG | DE 15 A 17 AÑOS |
| 2 | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | NO | NO | NO | NO | SI | NO | ... | NO | NO | SI | 1 | ATROPELLO | TURISMO | CONDUCTOR | HOMBRE | IL | DE 35 A 39 AÑOS |
| 7 | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | NO | NO | NO | NO | SI | NO | ... | NO | NO | SI | 1 | CHOQUE CON OBJETO FIJO | TURISMO | CONDUCTOR | HOMBRE | IL | DE 21 A 24 AÑOS |
| 8 | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | NO | NO | NO | NO | SI | NO | ... | NO | NO | SI | 1 | CHOQUE CON OBJETO FIJO | TURISMO | VIAJERO | HOMBRE | HG | DE 21 A 24 AÑOS |
| 9 | 01/01/2018 | DE 1:00 A 1:59 | LUNES | SAN BLAS | NO | NO | NO | NO | SI | NO | ... | NO | NO | SI | 1 | COLISIÓN DOBLE | TURISMO | CONDUCTOR | HOMBRE | IL | DE 25 A 29 AÑOS |

- For visualization purposes, specific dataframes were created grouping data, delete columns and setting specific indexes.

## 4. Key Findings and Insights, which synthesizes the results of Exploratory Data Analysis in an insightful and actionable manner

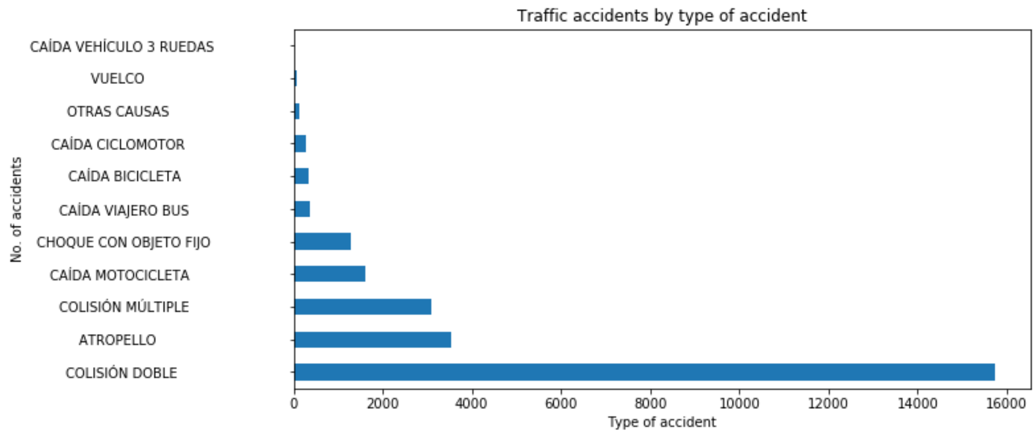
Some visualizations were plotted for the purposes of a better understanding of the information.

4.1. Traffic accidents by day of the week



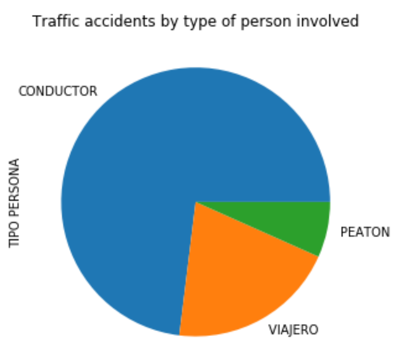

As per the above bar chart, we can extract that Friday is the day of the week when more accidents take place, while on the other hand weekends (Saturday and Sunday) are the days when less accidents take place.

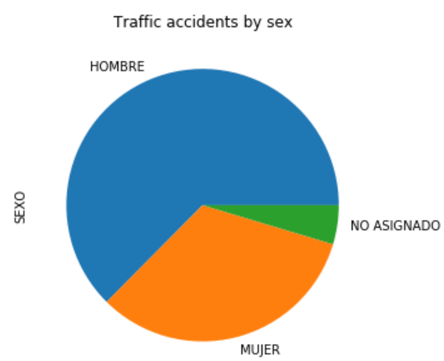
4.2. Traffic accidents by type of accident




As per the above bar chart, we can extract that most of the accidents are double collisions between vehicles, followed by pedestrian impacts and multiple collisions.

4.3. Traffic accidents by type of person involved and sex



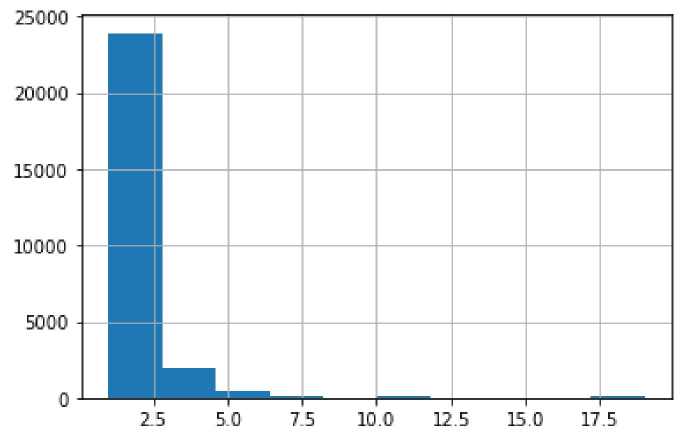

As per the above pie chart, we can extract that in most of the accidents the type of person involved was the driver, as well as that in most accidents the person involved was male.

4.4. Casualties in traffic accidents by date

| | FECHA | Victimas |
|---|---|---|
| 0 | 07/03/2018 | 514 |
| 1 | 20/07/2018 | 487 |
| 2 | 17/07/2018 | 345 |
| 3 | 23/05/2018 | 286 |
| 4 | 15/04/2018 | 271 |
| 5 | 06/04/2018 | 238 |
| 6 | 09/05/2018 | 236 |
| 7 | 27/04/2018 | 226 |
| 8 | 06/01/2018 | 216 |
| 9 | 06/10/2018 | 212 |

## 5. Determining normality

| | Victimas |
|---|---|
| count | 26463.000000 |
| mean | 1.488115 |
| std | 1.278060 |
| min | 1.000000 |
| 25% | 1.000000 |
| 50% | 1.000000 |
| 75% | 2.000000 |
| max | 19.000000 |

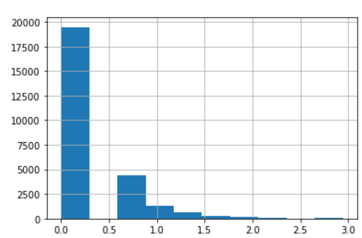



Accidents occurred in Madrid in 2018 had a minimum number of casualties of 1 and a maximum of 19, being the mean 1.49 and the standard deviation 1.28.

Our target, the number of casualties (Victimas) does not look normal. If we run D'Agostino test, we obtain that P-value is close to 0, so our variable y is not normally distributed.

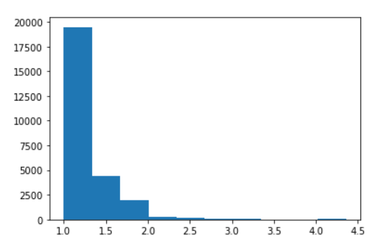
```
NormaltestResult(statistic=31902.082170149784, pvalue=0.0)
```

<u>Testing log / Square root / Box cox</u>



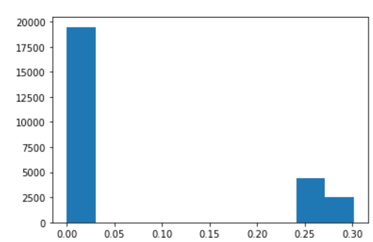
```
normaltest(log_casualties)
```

```
NormaltestResult(statistic=10340.212264191083, pvalue=0.0)
```



```
normaltest(sqrt_casualties)
```

```
NormaltestResult(statistic=19078.37606717139, pvalue=0.0)
```



```
normaltest(boxcox_casualties)
```

```
NormaltestResult(statistic=6011.159265145325, pvalue=0.0)
```

As per the results above, it has not been possible to obtain a normally distributed y-variable.

## 6. One-hot encoding

Since all the variables except y-variable are categorical, we need first to apply one-hot encoding to the data set. For these purposes, 107 extra columns will be created, giving raise to a one-hot encoded dataframe with 129 columns:

| | Victimas | RANGO HORARIO_0 | RANGO HORARIO_1 | RANGO HORARIO_2 | RANGO HORARIO_3 | RANGO HORARIO_4 | RANGO HORARIO_5 | RANGO HORARIO_6 | RANGO HORARIO_7 | RANGO HORARIO_8 | ... | CPSV Barro_0 | CPSV Barro_1 | CPSV Grava Suelta_0 | CPSV Grava Suelta_1 | CPSV Hielo_0 | CPSV Hielo_1 | CPSV Seca Y Limpia_0 | CPSV Seca Y Limpia_1 | CPSV Aceite_0 | CPSV Aceite_1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 |
| 2 | 1 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 |
| 7 | 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 |
| 8 | 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 |
| 9 | 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | ... | 1.0 | 0.0 | 1.0 | 0.0 | 1.0 | 0.0 | 0.0 | 1.0 | 1.0 | 0.0 |

5 rows × 129 columns

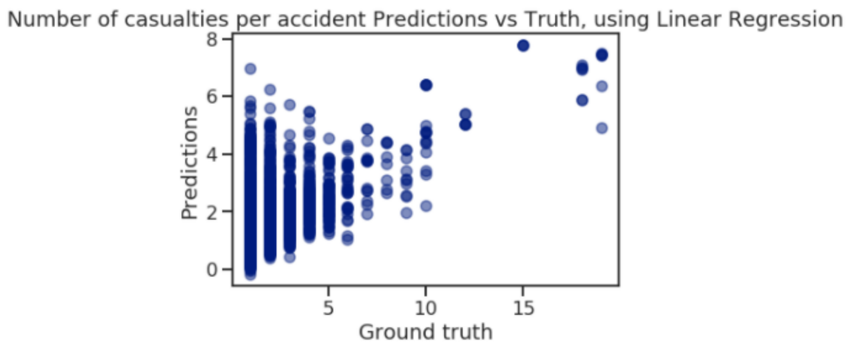
## 7. Train and test

Once the data set is ready, we will proceed to train and test, fit a basic linear regression model on the training data and calculate mean squared error on both the train and test sets.

```
[train    1.057112
 test     1.099653
 Name: one-hot enc, dtype: float64]
```

Error values are very different for the train and test data. In particular, the errors on the test data are much higher, which could be due to the one-hot encoded model being overfitting the data.

Finally, we plot predictions vs actual for the one-hot encoded model:




## 8. Cross validation

8.1. <u>K-Folds</u>

We will first split the data set into three folds using KFold object and proceed to train and test the basic linear regression model again, and calculate mean squared error:

```
[0.25680219387013503, 0.22645297494653172, 0.25643158986857195]
```

Mean squared error has significantly reduced, so our model is performing better.

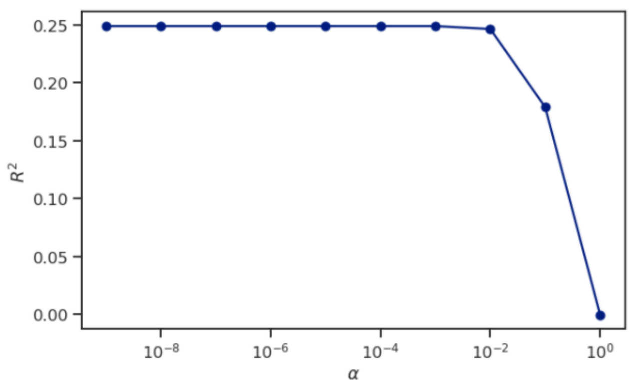
If we do the same but appropriately scaling our data as we go through the folds, we obtain:

```
[0.25666920400037796, 0.22649752484204733, 0.2564449533587234]
```

Results are almost the same, because for vanilla linear regression with no regularization, scaling does not matter for performance.

8.2. Hyperparameter tuning

We first generate an exponentially spaces range of values using the numpy geomspace function, and then tune the alpha hyperparameter for Lasso regression:




```
[(1e-09, 0.24887118802440888),
 (1e-08, 0.24887118886983872),
 (1e-07, 0.24887119732212748),
 (1e-06, 0.2488712816618286),
 (1e-05, 0.2488723038377859),
 (0.0001, 0.24887619113517923),
 (0.001, 0.24887174362252773),
 (0.01, 0.24633063661194332),
 (0.1, 0.178869330665105),
 (1.0, -0.00017640843345367863)]
```

As per the above, we have found that alpha = 0.01 would be optimum, as a higher degree would not significantly improve the model.

## 9. Suggestions for next steps in analyzing the data

In order to improve and go deeper into the analysis done, the following is proposed:

- The current dataset to be updated and complemented with additional data.
- As per the results of section 5, it has not been possible to obtain a normally distributed y-variable through Testing log / Square root / Box cox, so it is suggested to further analyze this and look for other reasons why such results are obtained and additional tools.
- Increase the number of iterations for Lasso.
- Keep training and testing the model using multiple and non-linear regression, as well as other regularization and cross validation techniques, polynominal features, ridge regression…

In [3]:

```
# Import libraries
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
```

In [4]:

```
# Download information (source: Madrid City Council https://datos.madrid.es)
!wget -q -O 'accidents.csv' https://datos.madrid.es/egob/catalogo/300228-18-accidentes-tra
fico-detalle.csv
```

In [5]:

```
# Check the encoding of the csv
#!pip install chardet
#import chardet

#with open("accidents.csv", 'rb') as file:
#    print(chardet.detect(file.read()))
```

In [6]:

```
df_accidents = pd.read_csv('accidents.csv', sep=';', encoding='ISO-8859-1')
df_accidents.head(5)
```

Out[6]:

| | FECHA | RANGO HORARIO | DIA SEMANA | DISTRITO | LUGAR ACCIDENTE | Nº | Nº PARTE | CPFA Granizo | CPFA Hielo | CPF Lluvi |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | CALLE DE SAN BASILIO - CALLE DEL CRISTO DE LA ... | 0 | 2018/1 | NO | NO | N |
| **1** | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | CALLE DE SAN BASILIO - CALLE DEL CRISTO DE LA ... | 0 | 2018/1 | NO | NO | N |
| **2** | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | CALLE DE SAN BASILIO - CALLE DEL CRISTO DE LA ... | 0 | 2018/1 | NO | NO | N |
| **3** | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | AVENIDA DE FRANCISCO PI Y MARGALL - AVENIDA DE... | | 2018/3 | NO | NO | N |
| **4** | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | AVENIDA DE FRANCISCO PI Y MARGALL - AVENIDA DE... | | 2018/3 | NO | NO | N |

5 rows × 26 columns

In [7]:

```
df_accidents.shape
```

Out[7]:

```
(30122, 26)
```

In [8]:

```
df_accidents.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 30122 entries, 0 to 30121
Data columns (total 26 columns):
FECHA                 30122 non-null object
RANGO HORARIO         30122 non-null object
DIA SEMANA            30122 non-null object
DISTRITO              30122 non-null object
LUGAR ACCIDENTE       30122 non-null object
Nº                    30122 non-null object
Nº PARTE              30122 non-null object
CPFA Granizo          30122 non-null object
CPFA Hielo            30122 non-null object
CPFA Lluvia           30122 non-null object
CPFA Niebla           30122 non-null object
CPFA Seco             30122 non-null object
CPFA Nieve            30122 non-null object
CPSV Mojada           30122 non-null object
CPSV Aceite           30122 non-null object
CPSV Barro            30122 non-null object
CPSV Grava Suelta     30122 non-null object
CPSV Hielo            30122 non-null object
CPSV Seca Y Limpia    30122 non-null object
* Nº VICTIMAS         30122 non-null int64
TIPO ACCIDENTE        30122 non-null object
Tipo Vehiculo         30122 non-null object
TIPO PERSONA          30122 non-null object
SEXO                  30122 non-null object
LESIVIDAD             30122 non-null object
Tramo Edad            30122 non-null object
dtypes: int64(1), object(25)
memory usage: 6.0+ MB
```

In [9]:

```
df_accidents.describe()
```

Out[9]:

| | * Nº VICTIMAS |
|---|---|
| **count** | 30122.000000 |
| **mean** | 1.459232 |
| **std** | 1.232216 |
| **min** | 1.000000 |
| **25%** | 1.000000 |
| **50%** | 1.000000 |
| **75%** | 2.000000 |
| **max** | 19.000000 |

# Data cleasing

In [10]:

```
# Delete columns
df_accidents = df_accidents.drop(df_accidents.columns[[4, 5, 6]], axis=1)

# Rename columns
df_accidents.rename(columns={'* Nº VICTIMAS':'Victimas'}, inplace=True)
```

In [11]:

```
df_accidents.shape
```

Out[11]:

```
(30122, 23)
```

In [12]:

```
# Check if there is any missing value
df_accidents.isna().sum()
```

Out[12]:

```
FECHA                  0
RANGO HORARIO          0
DIA SEMANA             0
DISTRITO               0
CPFA Granizo           0
CPFA Hielo             0
CPFA Lluvia            0
CPFA Niebla            0
CPFA Seco              0
CPFA Nieve             0
CPSV Mojada            0
CPSV Aceite            0
CPSV Barro             0
CPSV Grava Suelta      0
CPSV Hielo             0
CPSV Seca Y Limpia     0
Victimas               0
TIPO ACCIDENTE         0
Tipo Vehiculo          0
TIPO PERSONA           0
SEXO                   0
LESIVIDAD              0
Tramo Edad             0
dtype: int64
```

In [13]:

```
# Check if there is any null value
df_accidents.isnull().sum()
```

Out[13]:

```
FECHA                 0
RANGO HORARIO         0
DIA SEMANA            0
DISTRITO              0
CPFA Granizo          0
CPFA Hielo            0
CPFA Lluvia           0
CPFA Niebla           0
CPFA Seco             0
CPFA Nieve            0
CPSV Mojada           0
CPSV Aceite           0
CPSV Barro            0
CPSV Grava Suelta     0
CPSV Hielo            0
CPSV Seca Y Limpia    0
Victimas              0
TIPO ACCIDENTE        0
Tipo Vehiculo         0
TIPO PERSONA          0
SEXO                  0
LESIVIDAD             0
Tramo Edad            0
dtype: int64
```

In [14]:

```
# Drop rows with value TIPO PERSONA = TESTIGO, as witness were not directly involved in th
e accident
df_accidents = df_accidents[df_accidents['TIPO PERSONA']!='TESTIGO
']
```

In [15]:

```
df_accidents.head()
```

Out[15]:

| | FECHA | RANGO HORARIO | DIA SEMANA | DISTRITO | CPFA Granizo | CPFA Hielo | CPFA Lluvia | CPFA Niebla | CPFA Seco | CPFA Nieve |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | NO | NO | NO | NO | SI | NO |
| **2** | 01/01/2018 | DE 00:00 A 00:59 | LUNES | USERA | NO | NO | NO | NO | SI | NO |
| **7** | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | NO | NO | NO | NO | SI | NO |
| **8** | 01/01/2018 | DE 1:00 A 1:59 | LUNES | HORTALEZA | NO | NO | NO | NO | SI | NO |
| **9** | 01/01/2018 | DE 1:00 A 1:59 | LUNES | SAN BLAS | NO | NO | NO | NO | SI | NO |

5 rows × 23 columns

# EDA

In [16]:

```
# Traffic accidents by day of the week
df_day = df_accidents['DIA SEMANA'].value_counts()
df_day
```

Out[16]:

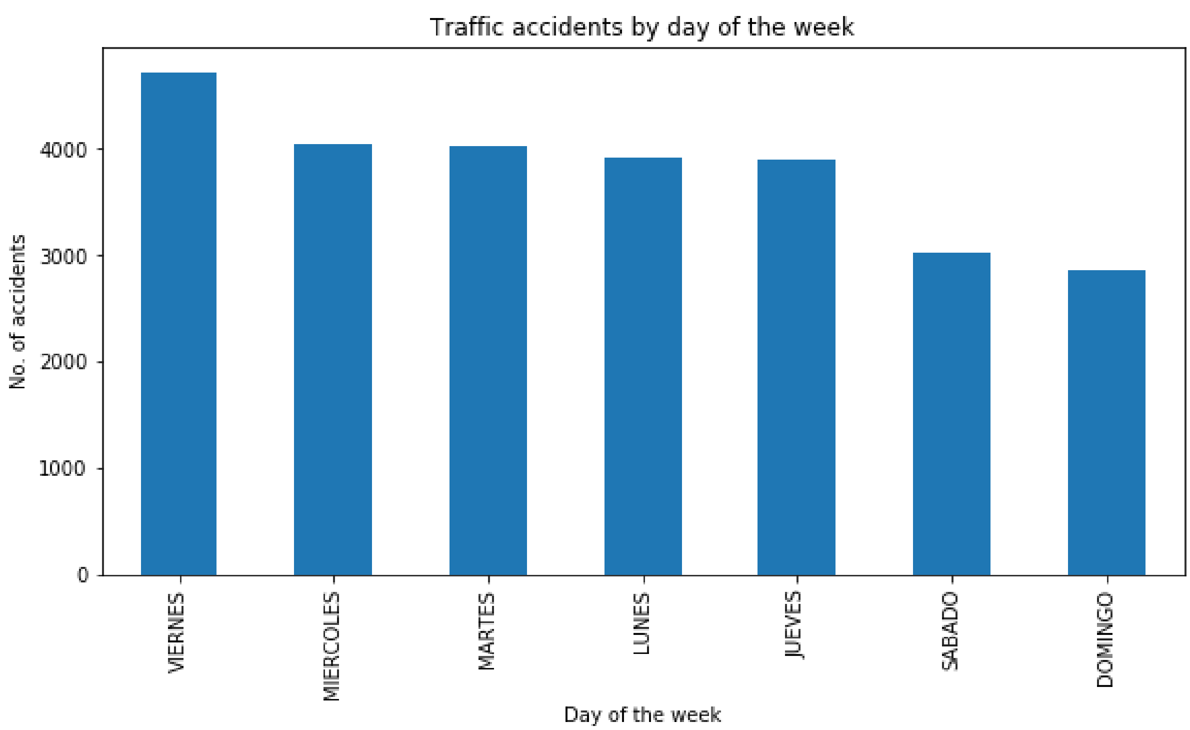
```
VIERNES      4721
MIERCOLES    4042
MARTES       4028
LUNES        3906
JUEVES       3899
SABADO       3018
DOMINGO      2849
Name: DIA SEMANA, dtype: int64
```

In [17]:

```
df_day.plot(kind='bar',
             stacked=False,
             figsize=(10, 5))

plt.title('Traffic accidents by day of the week')
plt.ylabel('No. of accidents')
plt.xlabel('Day of the week')

plt.show()
```




In [18]:

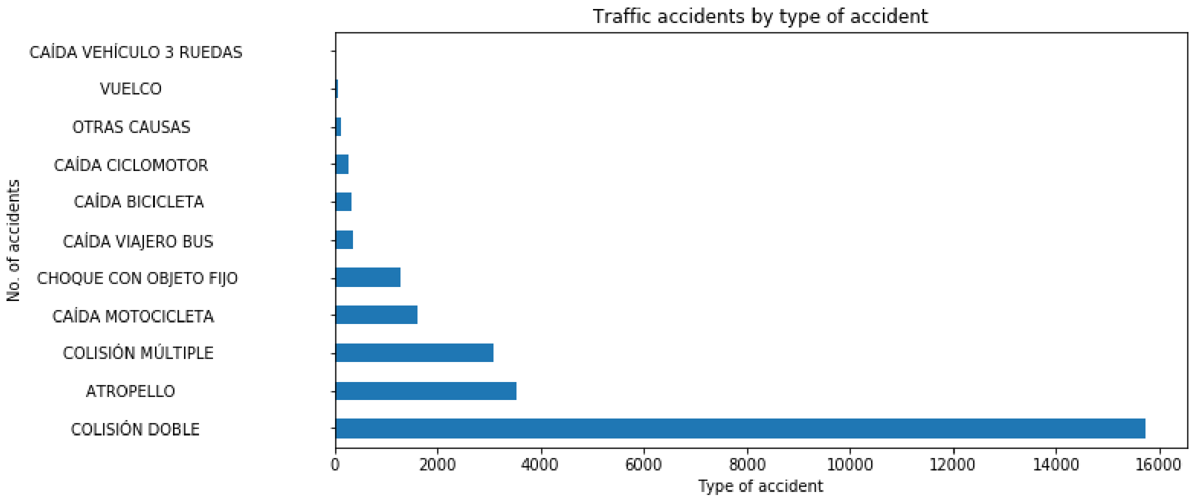
```
# Traffic accidents by type of accident
df_tipo = df_accidents['TIPO ACCIDENTE'].value_counts()
```

In [19]:

```
df_tipo.plot(kind='barh',
             stacked=True,
             figsize=(10, 5))

plt.title('Traffic accidents by type of accident')
plt.ylabel('No. of accidents')
plt.xlabel('Type of accident')

plt.show()
```




In [20]:

```
# Traffic accidents by type of person involved
df_person = df_accidents['TIPO PERSONA'].value_counts()
df_person
```

Out[20]:

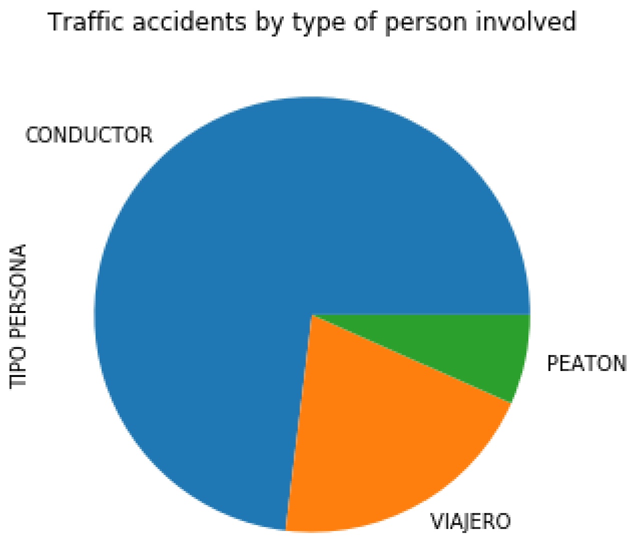
```
CONDUCTOR                                           19333
VIAJERO                                              5365
PEATON                                               1765
Name: TIPO PERSONA, dtype: int64
```

In [21]:

```
df_person.plot(kind='pie',
             stacked=True,
             figsize=(10, 5))

plt.title('Traffic accidents by type of person involved')

plt.show()
```




In [22]:

```
# Traffic accidents by sex
df_sex = df_accidents['SEXO'].value_counts()
df_sex
```

Out[22]:

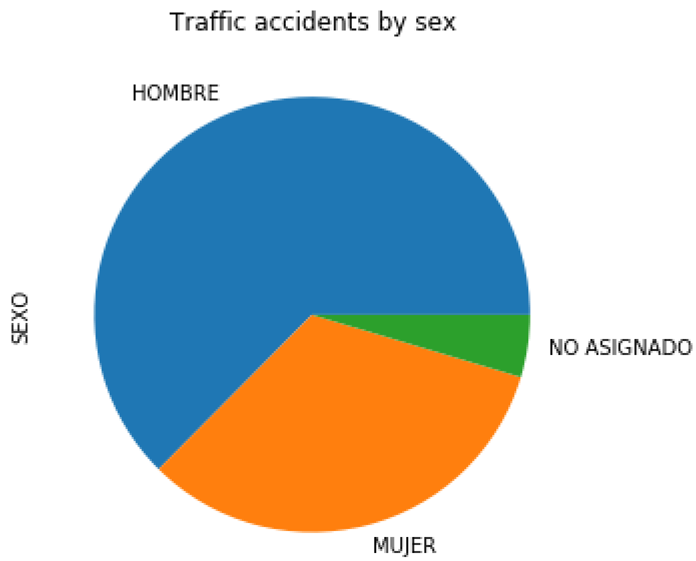
```
HOMBRE         16556
MUJER           8682
NO ASIGNADO     1225
Name: SEXO, dtype: int64
```

In [23]:

```
df_sex.plot(kind='pie',
             stacked=True,
             figsize=(10, 5))

plt.title('Traffic accidents by sex')

plt.show()
```




In [24]:

```
# Casualties by date
df_date = df_accidents.groupby('FECHA').sum()
df_date.sort_values(['Victimas'], ascending=False, axis=0, inplace=True)
df_date.head(10).reset_index()
```

Out[24]:

| | FECHA | Victimas |
|---|---|---|
| 0 | 07/03/2018 | 514 |
| 1 | 20/07/2018 | 487 |
| 2 | 17/07/2018 | 345 |
| 3 | 23/05/2018 | 286 |
| 4 | 15/04/2018 | 271 |
| 5 | 06/04/2018 | 238 |
| 6 | 09/05/2018 | 236 |
| 7 | 27/04/2018 | 226 |
| 8 | 06/01/2018 | 216 |
| 9 | 06/10/2018 | 212 |

In [25]:

```
df_accidents.describe()
```

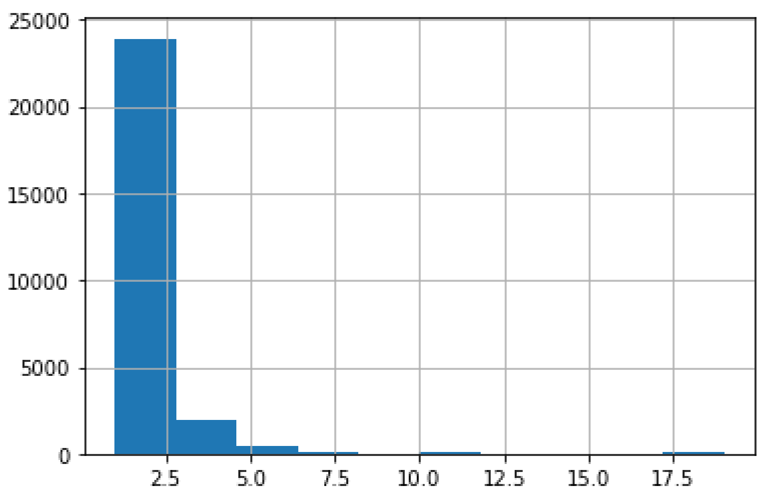
Out[25]:

| | Victimas |
|---|---|
| **count** | 26463.000000 |
| **mean** | 1.488115 |
| **std** | 1.278060 |
| **min** | 1.000000 |
| **25%** | 1.000000 |
| **50%** | 1.000000 |
| **75%** | 2.000000 |
| **max** | 19.000000 |

# Determining normality

In [26]:

```
df_accidents.Victimas.hist();
```




In [27]:

```
from scipy.stats.mstats import normaltest # D'Agostino K^2 Test
normaltest(df_accidents.Victimas.values)
```

Out[27]:

```
NormaltestResult(statistic=31902.082170149784, pvalue=0.0)
```

In [28]:

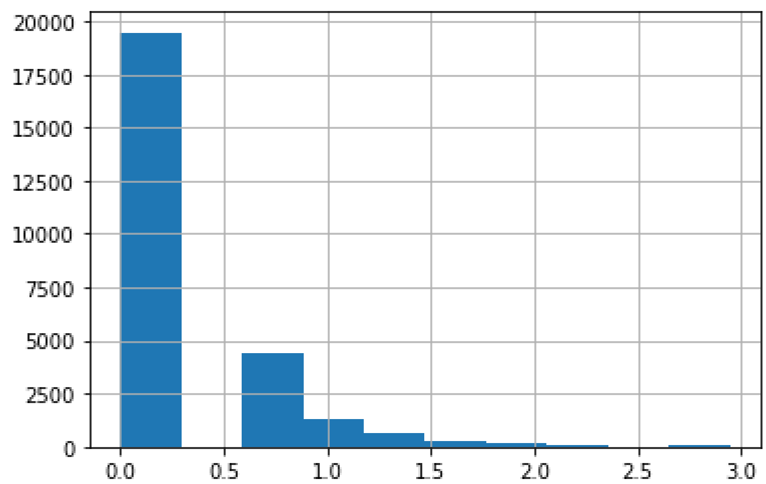
```
# Testing log
log_casualties = np.log(df_accidents.Victimas)
```

In [29]:

```
log_casualties.hist()
```

Out[29]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7cdd7a6350>
```




In [30]:

```
normaltest(log_casualties)
```

Out[30]:

```
NormaltestResult(statistic=10340.212264191083, pvalue=0.0)
```

In [31]:

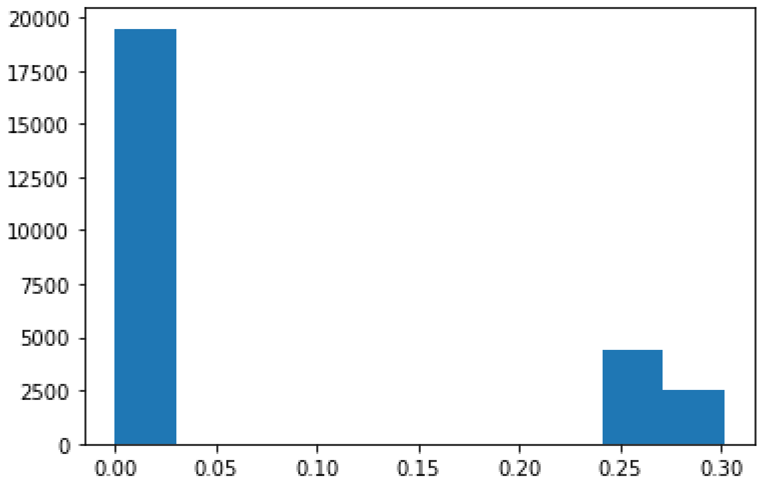
```
# Box cox
from scipy.stats import boxcox
bc_result = boxcox(df_accidents.Victimas)
boxcox_casualties = bc_result[0]
lam = bc_result[1]
```

In [32]:

```
plt.hist(boxcox_casualties)
```

Out[32]:

```
(array([19470.,     0.,     0.,     0.,     0.,     0.,     0.,     0.,
         4445.,  2548.]),
 array([0.        , 0.03015594, 0.06031189, 0.09046783, 0.12062378,
        0.15077972, 0.18093567, 0.21109161, 0.24124756, 0.2714035 ,
        0.30155945]),
 <a list of 10 Patch objects>)
```




In [33]:

```
normaltest(boxcox_casualties)
```

Out[33]:

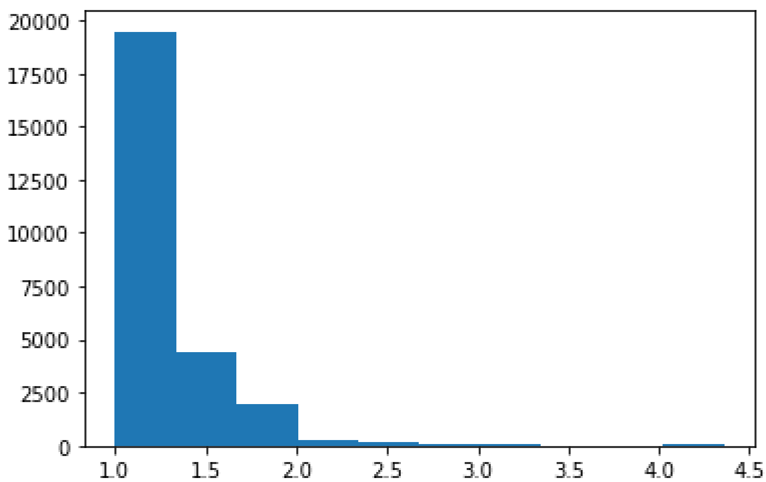
```
NormaltestResult(statistic=6011.159265145325, pvalue=0.0)
```

In [34]:

```
# Square root
sqrt_casualties = np.sqrt(df_accidents.Victimas)
plt.hist(sqrt_casualties)
```

Out[34]:

```
(array([1.947e+04, 4.445e+03, 1.917e+03, 2.790e+02, 1.680e+02, 6.500e+01,
        4.600e+01, 1.500e+01, 1.700e+01, 4.100e+01]),
 array([1.        , 1.33588989, 1.67177979, 2.00766968, 2.34355958,
        2.67944947, 3.01533937, 3.35122926, 3.68711915, 4.02300905,
        4.35889894]),
 <a list of 10 Patch objects>)
```




In [35]:

```
normaltest(sqrt_casualties)
```

Out[35]:

```
NormaltestResult(statistic=19078.37606717139, pvalue=0.0)
```

# One-hot encoding

In [36]:

```
df_accidents = df_accidents.drop(df_accidents.columns[[0]], axis=1)
df_accidents.head()
```

Out[36]:

| | RANGO HORARIO | DIA SEMANA | DISTRITO | CPFA Granizo | CPFA Hielo | CPFA Lluvia | CPFA Niebla | CPFA Seco | CPFA Nieve | CPSV Mojada | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | DE 00:00 A 00:59 | LUNES | USERA | NO | NO | NO | NO | SI | NO | NO | ... |
| **2** | DE 00:00 A 00:59 | LUNES | USERA | NO | NO | NO | NO | SI | NO | NO | ... |
| **7** | DE 1:00 A 1:59 | LUNES | HORTALEZA | NO | NO | NO | NO | SI | NO | NO | ... |
| **8** | DE 1:00 A 1:59 | LUNES | HORTALEZA | NO | NO | NO | NO | SI | NO | NO | ... |
| **9** | DE 1:00 A 1:59 | LUNES | SAN BLAS | NO | NO | NO | NO | SI | NO | NO | ... |

5 rows × 22 columns

In [37]:

```
# Select the object (string) columns
mask = df_accidents.dtypes == np.object
categorical_cols = df_accidents.columns[mask]
```

In [38]:

```
# Determine how many extra columns would be created
num_ohc_cols = (df_accidents[categorical_cols]
                .apply(lambda x: x.nunique())
                .sort_values(ascending=False))


# No need to encode if there is only one value
small_num_ohc_cols = num_ohc_cols.loc[num_ohc_cols>1]

# Number of one-hot columns is one less than the number of categories
small_num_ohc_cols -= 1

# This is 107 columns, assuming the original ones are dropped.
small_num_ohc_cols.sum()
```

Out[38]:

```
107
```

In [39]:

```
# Create a new data set where all of the above categorical features will be one-hot encode
d
from sklearn.preprocessing import OneHotEncoder, LabelEncoder

# Copy of the data
data_ohc = df_accidents.copy()

# The encoders
le = LabelEncoder()
ohc = OneHotEncoder()

for col in num_ohc_cols.index:

    # Integer encode the string categories
    dat = le.fit_transform(data_ohc[col]).astype(np.int)

    # Remove the original column from the dataframe
    data_ohc = data_ohc.drop(col, axis=1)

    # One hot encode the data--this returns a sparse array
    new_dat = ohc.fit_transform(dat.reshape(-1,1))

    # Create unique column names
    n_cols = new_dat.shape[1]
    col_names = ['_'.join([col, str(x)]) for x in range(n_cols)]

    # Create the new dataframe
    new_df = pd.DataFrame(new_dat.toarray(),
                          index=data_ohc.index,
                          columns=col_names)

    # Append the new data to the dataframe
    data_ohc = pd.concat([data_ohc, new_df], axis=1)
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
```

```
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
```

```
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
```

```
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/preprocessing/_e
ncoders.py:415: FutureWarning: The handling of integer data will change in ve
rsion 0.22. Currently, the categories are determined based on the range [0, m
ax(values)], while in the future they will be determined based on the unique
values.
If you want the future behaviour and silence this warning, you can specify "c
ategories='auto'".
In case you used a LabelEncoder before this OneHotEncoder to convert the cate
gories to integers, then you can now use the OneHotEncoder directly.
  warnings.warn(msg, FutureWarning)
```

In [40]:

```
# Column difference is as calculated above
data_ohc.shape[1] - df_accidents.shape[1]
```

Out[40]:

```
107
```

In [41]:

```
print(df_accidents.shape[1])

# Remove the string columns from the dataframe
df_accidents = df_accidents.drop(num_ohc_cols.index, axis=1)

print(df_accidents.shape[1])
```

```
22
1
```

In [42]:

```
# Create train and test splits of the data set
from sklearn.model_selection import train_test_split

y_col = 'Victimas'

# Split the data that is one-hot encoded
feature_cols = [x for x in data_ohc.columns if x != y_col]
X_data_ohc = data_ohc[feature_cols]
y_data_ohc = data_ohc[y_col]

X_train_ohc, X_test_ohc, y_train_ohc, y_test_ohc = train_test_split(X_data_ohc, y_data_ohc
,
                                                                    test_size=0.3, random_state=42)
```

In [43]:

```
data_ohc.head(5)
```

Out[43]:

| | Victimas | RANGO HORARIO_0 | RANGO HORARIO_1 | RANGO HORARIO_2 | RANGO HORARIO_3 | RANGO HORARIO_4 | RANGO HORARIO_5 | HO |
|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **2** | 1 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **7** | 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **8** | 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |
| **9** | 1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | |

5 rows × 129 columns

In [44]:

```
data_ohc.shape
```

Out[44]:

```
(26463, 129)
```

In [45]:

```
data_ohc.dtypes
```

Out[45]:

```
Victimas                  int64
RANGO HORARIO_0         float64
RANGO HORARIO_1         float64
RANGO HORARIO_2         float64
RANGO HORARIO_3         float64
RANGO HORARIO_4         float64
RANGO HORARIO_5         float64
RANGO HORARIO_6         float64
RANGO HORARIO_7         float64
RANGO HORARIO_8         float64
RANGO HORARIO_9         float64
RANGO HORARIO_10        float64
RANGO HORARIO_11        float64
RANGO HORARIO_12        float64
RANGO HORARIO_13        float64
RANGO HORARIO_14        float64
RANGO HORARIO_15        float64
RANGO HORARIO_16        float64
RANGO HORARIO_17        float64
RANGO HORARIO_18        float64
RANGO HORARIO_19        float64
RANGO HORARIO_20        float64
RANGO HORARIO_21        float64
RANGO HORARIO_22        float64
RANGO HORARIO_23        float64
DISTRITO_0              float64
DISTRITO_1              float64
DISTRITO_2              float64
DISTRITO_3              float64
DISTRITO_4              float64
                         ...
TIPO PERSONA_0          float64
TIPO PERSONA_1          float64
TIPO PERSONA_2          float64
SEXO_0                  float64
SEXO_1                  float64
SEXO_2                  float64
CPFA Niebla_0           float64
CPFA Niebla_1           float64
CPFA Granizo_0          float64
CPFA Granizo_1          float64
CPFA Hielo_0            float64
CPFA Hielo_1            float64
CPFA Lluvia_0           float64
CPFA Lluvia_1           float64
CPFA Nieve_0            float64
CPFA Nieve_1            float64
CPFA Seco_0             float64
CPFA Seco_1             float64
CPSV Mojada_0           float64
CPSV Mojada_1           float64
CPSV Barro_0            float64
CPSV Barro_1            float64
CPSV Grava Suelta_0     float64
CPSV Grava Suelta_1     float64
```

```
CPSV Hielo_0           float64
CPSV Hielo_1           float64
CPSV Seca Y Limpia_0   float64
CPSV Seca Y Limpia_1   float64
CPSV Aceite_0          float64
CPSV Aceite_1          float64
Length: 129, dtype: object
```

# Train and test

In [46]:

```
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

LR = LinearRegression()

# Storage for error values
error_df = list()

LR = LR.fit(X_train_ohc, y_train_ohc)
y_train_ohc_pred = LR.predict(X_train_ohc)
y_test_ohc_pred = LR.predict(X_test_ohc)

error_df.append(pd.Series({'train': mean_squared_error(y_train_ohc, y_train_ohc_pred),
                           'test' : mean_squared_error(y_test_ohc,  y_test_ohc_pred)},
                          name='one-hot enc'))


error_df
```

Out[46]:

```
[train    1.221186
 test     1.194478
 Name: one-hot enc, dtype: float64]
```

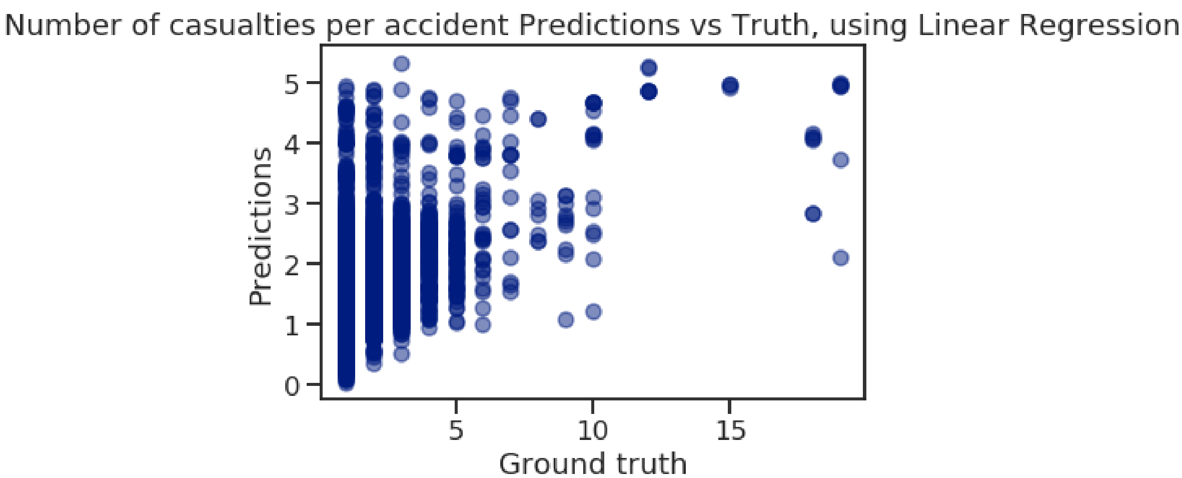
In [47]:

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline


sns.set_context('talk')
sns.set_style('ticks')
sns.set_palette('dark')

ax = plt.axes()
# we are going to use y_test, y_test_pred
ax.scatter(y_test_ohc, y_test_ohc_pred, alpha=.5)

ax.set(xlabel='Ground truth',
       ylabel='Predictions',
       title='Number of casualties per accident Predictions vs Truth, using Linear Regress
ion');
```




# Cross validation

In [48]:

```
from sklearn.preprocessing import StandardScaler, PolynomialFeatures
from sklearn.model_selection import KFold, cross_val_predict
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.metrics import r2_score
from sklearn.pipeline import Pipeline
```

In [49]:

```
data_ohc = data_ohc.reset_index()
X = data_ohc.drop('Victimas', axis=1)
y = data_ohc.Victimas
```

In [50]:

```
kf = KFold(shuffle=True, random_state=72018, n_splits=3)
```

In [51]:

```
for train_index, test_index in kf.split(X):
    print("Train index:", train_index[:10], len(train_index))
    print("Test index:",test_index[:10], len(test_index))
    print('')
```

```
Train index: [ 0  2  4  5  6  8 10 11 12 13] 17642
Test index: [ 1  3  7  9 14 17 18 19 20 26] 8821

Train index: [ 0  1  2  3  4  7  8  9 10 11] 17642
Test index: [ 5  6 16 22 23 24 29 38 41 43] 8821

Train index: [ 1  3  5  6  7  9 14 16 17 18] 17642
Test index: [ 0  2  4  8 10 11 12 13 15 21] 8821
```

In [52]:

```
scores = []
lr = LinearRegression()

for train_index, test_index in kf.split(X):
    X_train, X_test, y_train, y_test = (X.iloc[train_index, :],
                                        X.iloc[test_index, :],
                                        y[train_index],
                                        y[test_index])

    lr.fit(X_train, y_train)

    y_pred = lr.predict(X_test)

    score = r2_score(y_test.values, y_pred)

    scores.append(score)

scores
```

Out[52]:

```
[0.25680219387013503, 0.22645297494653172, 0.25643158986857195]
```

In [53]:

```
# Scaling
scores = []

lr = LinearRegression()
s = StandardScaler()

for train_index, test_index in kf.split(X):
    X_train, X_test, y_train, y_test = (X.iloc[train_index, :],
                                        X.iloc[test_index, :],
                                        y[train_index],
                                        y[test_index])

    X_train_s = s.fit_transform(X_train)

    lr.fit(X_train_s, y_train)

    X_test_s = s.transform(X_test)

    y_pred = lr.predict(X_test_s)

    score = r2_score(y_test.values, y_pred)

    scores.append(score)

scores
```

Out[53]:

```
[0.25666920400037796, 0.22649752484204733, 0.2564449533587234]
```

In [54]:

```
# Hyperparameter tuning
alphas = np.geomspace(1e-9, 1e0, num=10)
alphas
```

Out[54]:

```
array([1.e-09, 1.e-08, 1.e-07, 1.e-06, 1.e-05, 1.e-04, 1.e-03, 1.e-02,
       1.e-01, 1.e+00])
```

In [55]:

```
scores = []
coefs = []
for alpha in alphas:
    las = Lasso(alpha=alpha, max_iter=100)

    estimator = Pipeline([
        ("scaler", s),
        ("lasso_regression", las)])

    predictions = cross_val_predict(estimator, X, y, cv = kf)

    score = r2_score(y, predictions)

    scores.append(score)
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 10446.1010953826
55, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 11724.1566686666
03, tolerance: 3.1978390261875114
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 9784.53904647412
7, tolerance: 2.6332655991384213
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 10445.8183195387
85, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 11723.7337941942
24, tolerance: 3.1978390261875114
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 9784.0540256503
9, tolerance: 2.6332655991384213
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 10442.9907485589
57, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 11719.5053978885
12, tolerance: 3.1978390261875114
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 9779.2046326504
8, tolerance: 2.6332655991384213
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 10414.7337085554
85, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 11677.2561621075
24, tolerance: 3.1978390261875114
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
```

```
ight want to increase the number of iterations. Duality gap: 9730.79217907072
6, tolerance: 2.6332655991384213
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 10134.0450643680
2, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 11258.2439434733
86, tolerance: 3.1978390261875114
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 9254.69691068688
7, tolerance: 2.6332655991384213
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 7511.08337224725
6, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 7417.12774263600
4, tolerance: 3.1978390261875114
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 5249.16307517615
6, tolerance: 2.6332655991384213
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 196.613746719460
32, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 216.713160530391
18, tolerance: 3.1978390261875114
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 54.6670984690281
3, tolerance: 2.6332655991384213
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 2.81987916535217
66, tolerance: 2.8133928352794486
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 15.9600125718534
4, tolerance: 3.1978390261875114
```

```
  positive)
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 5.8024500819774
4, tolerance: 2.6332655991384213
  positive)
```

In [56]:

```
list(zip(alphas,scores))
```

Out[56]:

```
[(1e-09, 0.24887118802440888),
 (1e-08, 0.24887118886983872),
 (1e-07, 0.24887119732212748),
 (1e-06, 0.2488712816618286),
 (1e-05, 0.2488723038377859),
 (0.0001, 0.24887619113517923),
 (0.001, 0.24887174362252773),
 (0.01, 0.24633063661194332),
 (0.1, 0.1788869330665105),
 (1.0, -0.00017640843345367863)]
```

In [57]:

```
Lasso(alpha=1e-6).fit(X, y).coef_
```

```
/srv/conda/envs/notebook/lib/python3.7/site-packages/sklearn/linear_model/coo
rdinate_descent.py:475: ConvergenceWarning: Objective did not converge. You m
ight want to increase the number of iterations. Duality gap: 5449.19880610749
35, tolerance: 4.322401231908705
  positive)
```

Out[57]:

```
array([ 1.36341033e-06,  1.05666359e-01,  1.89363047e-03, -1.19331324e-01,
       -3.90787029e-02, -1.20170343e-01,  9.33688623e-02,  4.56393768e-02,
       -9.64661739e-02,  6.05335791e-02, -5.52274578e-02, -4.87164983e-02,
        8.61944334e-02,  1.23820276e-02,  1.58445942e-01,  4.76525392e-03,
       -4.15339221e-02,  6.78115921e-01, -1.01043396e-01,  4.58220940e-03,
        1.56892609e-01,  2.25412728e-01, -1.11824921e-01, -7.72288935e-02,
       -1.58455522e-01,  1.09692472e-02, -7.08815978e-02,  1.01549916e-01,
        3.99628916e-02, -7.75539642e-02, -3.40424699e-03, -6.36147745e-02,
       -7.96436127e-02, -8.19901828e-02, -1.25652587e-02,  2.16759315e-02,
       -2.87228629e-02,  1.82118481e-01, -1.34283539e-01,  6.09983976e-02,
        1.42108063e-02,  6.71501452e-02,  1.36944544e-02, -4.84995060e-02,
       -3.71243868e-02, -1.98084833e-02,  1.23822641e-01,  1.07644366e-01,
        1.45941880e-01,  1.22605201e-01,  3.60671312e-02,  1.37842624e-02,
        4.90262060e-04, -5.37269708e-03, -9.64365647e-03,  1.86330345e-03,
        1.66149595e-02, -9.92548779e-04,  4.53747199e-01,  5.22706956e-03,
       -2.79796855e-02,  5.77146711e-03, -1.52774779e-01, -1.98721321e-01,
        3.70356813e-01,  1.25506173e-01,  1.65767781e+00, -4.53421490e-01,
        1.87770595e-01, -6.27939910e-01,  7.23441863e-02, -5.71800896e-01,
       -2.72989511e-01,  8.96586476e-02,  5.93589809e-02, -3.78795270e-01,
       -2.05679760e-01, -3.81182377e-01, -2.09134518e-01, -1.97566948e-01,
       -1.12014425e-01,  7.74212182e-01, -3.38581687e-01, -8.34724717e-02,
        2.77935955e-01, -4.39481832e-01, -5.27817015e-01,  1.13705155e-01,
       -1.03237781e-01, -4.71440561e-02, -2.33480722e-02,  0.00000000e+00,
        3.66315582e-02,  1.81012968e-02,  6.91560234e-01,  6.53860836e-01,
       -9.68312032e-02,  9.45224735e-01,  1.81559981e-01, -3.28899727e-01,
       -4.60143661e-01,  1.44902762e-01, -1.24731028e-02, -3.26815897e-02,
        1.74596314e-01,  2.77190973e-01, -8.92972255e-15,  4.07421817e-02,
       -1.42622396e-14,  5.94159282e-01, -5.66233171e-14,  5.87236634e-01,
       -0.00000000e+00,  3.96909696e-01, -2.93573144e-15,  3.97649520e-01,
       -0.00000000e+00, -1.11109756e-01,  0.00000000e+00, -2.02544841e-01,
        1.00690060e-16, -1.88302099e-01,  9.65636074e-16, -1.82787680e-01,
        0.00000000e+00, -5.65930170e-03,  8.75726217e-19, -1.36780829e-01,
        1.65170311e-14])
```

In [58]:

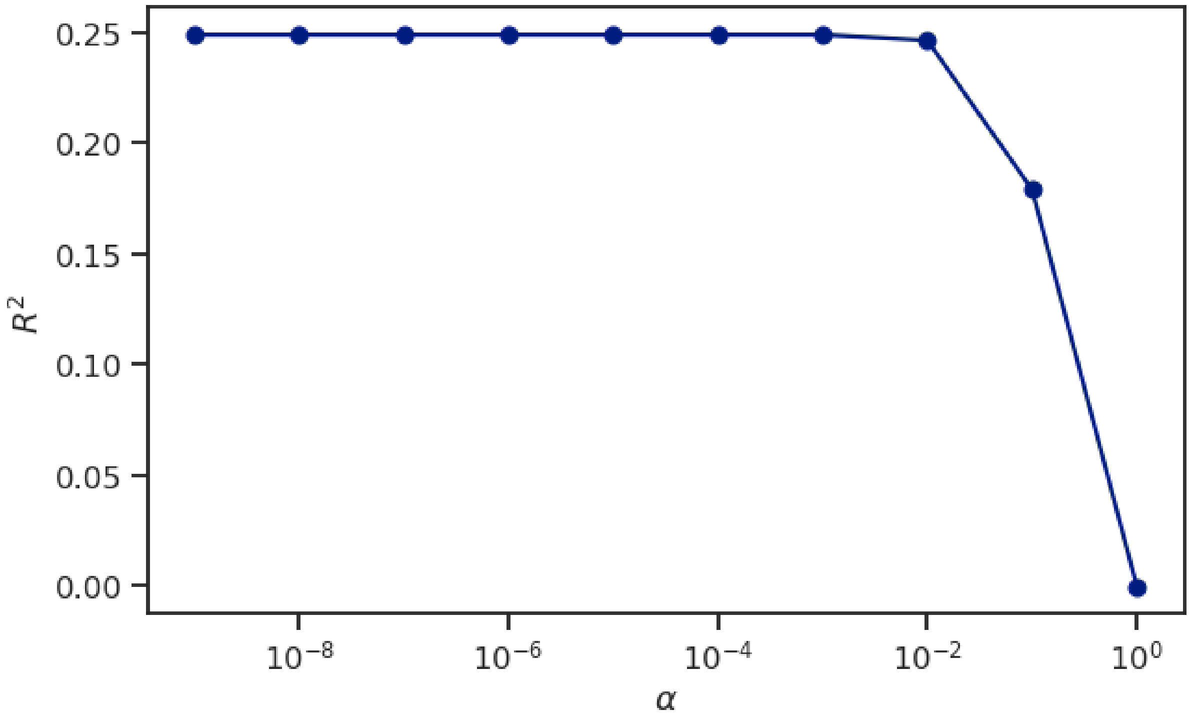
```
Lasso(alpha=1.0).fit(X, y).coef_
```

Out[58]:

```
array([ 8.05734695e-07,  0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
        0.00000000e+00, -0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
        0.00000000e+00,  0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
        0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
        0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00,  0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
        0.00000000e+00, -0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00, -0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00,  0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
        0.00000000e+00,  0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
        0.00000000e+00, -0.00000000e+00,  0.00000000e+00,  0.00000000e+00,
       -0.00000000e+00])
```

In [59]:

```
plt.figure(figsize=(10,6))
plt.semilogx(alphas, scores, '-o')
plt.xlabel('$\\alpha$')
plt.ylabel('$R^2$');
```



In [ ]:

```
# PolynomialFeatures

pf = PolynomialFeatures(degree=3)

scores = []
alphas = np.geomspace(0.06, 6.0, 20)
for alpha in alphas:
    las = Lasso(alpha=alpha, max_iter=100)

    estimator = Pipeline([
        ("scaler", s),
        ("make_higher_degree", pf),
        ("lasso_regression", las)])

    predictions = cross_val_predict(estimator, X, y, cv = kf)

    score = r2_score(y, predictions)

    scores.append(score)
```

In [ ]:

```
plt.semilogx(alphas, scores);
```

In [ ]:

```
# Once we have found the hyperparameter (alpha~1e-2=0.01)
# make the model and train it on ALL the data
# Then release it into the wild .....
best_estimator = Pipeline([
                    ("scaler", s),
                    ("make_higher_degree", PolynomialFeatures(degree=2)),
                    ("lasso_regression", Lasso(alpha=0.03))])

best_estimator.fit(X, y)
best_estimator.score(X, y)
```

In [ ]:

```

```