Communication

1. In data cleansing what are some of the best practices?

- Identify the sources of the data and before cleaning making sure that data was handled securely and there are no legal issues
- Identify format and the way data was collected to check for data pollution - error propagated by collection systems, mixing up of the default value and missing values, human biases - data entry problems, errors while transferring from one data source to another
- Gather Domain knowledge - special cases, default values, biases, etc.
- Try out data on certain models to get a better understanding of the data e.g: to validate for co-relation between attributes trying them out with predictive models with or without the given attribute data and compare the performance results
- Remove or replace redundant data - duplicates, insufficient data for attributes
- Make data have a consistent meaning for different types of values e.g: -1 for missing data for numeric and 'None or null for missing string types
- Understand the requirements of the future use of the data - analytics/prediction/distribution and decide to keep the dataset as whole or break it down based on requirements

2. How would you qualify a Data Science problem?

A problem which requires knowledge beyond domain knowledge, a problem which requires further evidence and research with the use of data. Many problems can have answers just with the use of facts and domain knowledge; it is when we need to go beyond those facts and require extra knowledge, knowledge, which helps the stakeholders take accurate decisions and it involves have data. There can be no data science without the data. Also, data needs to satisfy quantitative and qualitative requirements for a given problem.

3. What are the key concepts of Hypothesis Testing?

In Hypothesis problem testing the analyst has a conception/claim/notion about the data which is tested for being true with sufficient statistical evidence for it. **Null Hypothesis** is the statement that the value of the parameter is equal to the claimed value. It is the favored assumption. We assume that the null hypothesis is true until we prove otherwise. The antithesis of that is the **Alternative hypothesis**.

We might commit errors while testing the null hypothesis, the types can be understood from the table below

	Null Hypothesis is actually True	Null Hypothesis is actually False
We rejected Null Hypothesis	Type-I Error (False Negative)	True Negative
We Accepted the Null Hypothesis	True Positive	Type - II (error False Positive)

- A **Type I Error** is incorrectly rejecting a true null hypothesis (false negative).
- A **Type II Error** is incorrectly failing to reject an untrue null hypothesis (false positive).

In statistical testing null hypothesis is denoted by H_0 and alternative by H_1

BUSINESS PROBLEM:

The following is an example of a very typical conversation with prospective clients. A multi-national silicon chip manufacturing company is eager to implement an new IoT initiative - specifically predictive maintenance for the machinery in the manufacturing plants. What questions would you need to ask and factors would you need to consider to help them architect the solutions? Consider the following:

1. How would you convey the value of Data Science and Advanced Analytics to the organisation?

Traditional maintenance is a trade-off between minimizing the downtime and using the parts to its maximum. An experienced human can monitor parts and processes to suggest maintenance needs.

- Human capacity has its limits in the context of large manufacturing plants.
  - It's not available 24x7,
  - It can monitor only certain numbers of systems and can take only limited aspects into consideration.
- A machine can not only monitor the system in real-time but with help of Data Science and advance analytics leverage the human expert knowledge combined with data and help optimize the schedule to achieve best results.
- Company can move from OEM manufacturer maintenance schedule to continuous monitoring. Also, presenting the company with previous success stories of similar systems can help us convey the value of our work.

2. What considerations would you have to take into account with regards to the data?

- What kind of sensors are installed, do they generate required information and are they validated by domain expert?
- How does the sensor data convert into digital signal data?
- Does the data from different sensors need to be in sync? if yes, is it?
- What are different kinds of sensor noise present in the system?
- How does data convey different states of the sensors?
- What are the limitations while collecting the data - What is the velocity, volume, variety, and format of the data, is network bandwidth sufficient? and what kind of systems are required to stream and store the data?
- Which the steps of data cleaning and preparation do we need to apply?

3. What kind of statistical analysis would you carry out for the company?

- Initial analysis (Descriptive): Analysis of history of the plant. Determination of amount of data required for the system to work by showing the industry analysis - if present, and if not, then, generating domain experts perspective of data required

- Developing visualizations: After requirements gathering, we will be working with experts to provide with real-time visualizations of the performance of the plant - production numbers, the velocity of manufacturing, critical sensor information with general statistics: hourly, daily stats with an indication of ideal required stats
- Exploratory & diagnostic analysis: generating answers through diagnostic analysis for causes of failures/ problems

4. What kind of Machine learning problem(s) and which algorithms would you consider?

- Predicting the maintenance of the parts
  - Classification - predicts failure in future steps
  - Regression - predicts how much time is left before the next failure
- Finding the best course of assembling steps to increase the speed using Reinforcement learning
- Identification of Noise using neural networks
- Identification of Anamoly using neural networks

5. How would you validate your results and defend your decisions to the business owner?

- For predictive models perform cross-validation and present the success and failure rates of the model
- Check performance of the the models in the wild with silent runs with monitoring by experts
- Calculate the false accept and false rejects of actual runs, if they are in considerable margins then it can be conveyed to the business owner if not then we need to find a backup plan (improving our models) and presenting the expected future performance results in positive light, explaining that models are learning.

PROGRAMMING AND CODING TASKS:

1. Find the number of unique primes factors for the range [1,1000]. Try using a “for loop” and without

In [141]:

```
import math
import scipy.stats as st

def is_prime_for_loop(n):
    # prime number has be > 1
    if n < 2:
        return False
    remainders = [n % i for i in range(2, n)]
    # if no zero remainders then it's a prime
    if 0 in remainders:
        return False
    return True


def is_prime_recursive(n, i):
    if n < 2:
        return False

    if i >= n:
        return True

    if n == 2:
        return True

    if n % i == 0:
        return False
    else:
        return is_prime_recursive(n, i + 1)


def get_prime_for_range(a, b):
    primes = {}
    non_primes = {}
    primes_recursive = {}
    non_primes_recursive = {}

    for i in range(a, b + 1):
        if is_prime_for_loop(i):
            primes[i] = 1
        else:
            non_primes[i] = 1

        if is_prime_recursive(i,2):
            primes_recursive[i] = 1
        else:
            non_primes_recursive[i] = 1

    print ("Number of prime factors using for loop: ", len(primes.keys()), '
    print ("Number of prime factors using recursive : ", len(primes_recursiv


get_prime_for_range(1,1000)

# print(is_prime_for_loop(2),is_prime_for_loop(51), is_prime(2,2), is_prime(
```

```
Number of prime factors using for loop:  168
Number of prime factors using recursive :  168
```

What challenges do you seen in deploying this code into an environment when it would be need to be called frequently?

Need to implement the cache by saving the key value pair with key being the number and value being either 1 or 0. if the number is present in the cache then it has been calculated for and we can check if it's value is 1(prime) or 0(non-prime) without the need to recalculate.

2. X is a normally distributed variable with mean μ = 30 and standard deviation σ = 4. Find

a. $P(x < 40)$

b. $P(x > 21)$

c. $P(30 < x < 35)$

In [23]:

```
def get_p_value(z_value):

    if z_value < 0:
        return 1 - st.norm.cdf(z_value)
    return st.norm.cdf(z_value)

def get_z_value(x,mean, standard_deviation):
    return (x - mean)/standard_deviation


def p_calculations():

    p_value_less_than_40 = get_p_value(get_z_value(40, 30, 4))
    p_value_greater_than21 = 1 - get_p_value(get_z_value(21, 30, 4))
    p_value_between_30_35 = get_p_value(get_z_value(35, 30, 4)) - get_p_valu

    print("a.  P(x < 40): ", p_value_less_than_40, "\nb.  P(x > 21): ", p_va

p_calculations()
```

```
a.  P(x < 40):  0.9937903346742238
b.  P(x > 21):  0.012224472655044671
c.  P(30 < x < 35) : 0.39435022633314465
```

3. What is your favourite language for Data Analytics/Data Science e.g. R or Python, discuss three limitations and how you overcame or dealt with

them.

Python - Jack of all trades, Master of Data Science. Limitation - while implementing prototype in my current company I couldn't do multithreading, had to manually call different copy of same prototype from tmux to run the mutltiple copies of the program with different parts of the data

Data Analysis of U.K road accidents dataset - year 2016

In [24]:
```
import numpy as np
import pandas as pd

import matplotlib
import cufflinks as cf
import plotly
import plotly.offline as py
import plotly.graph_objs as go

cf.go_offline() # required to use plotly offline (no account required).
py.init_notebook_mode() # graphs charts inline (IPython).
```

In [25]:
```
# *
combined_data=pd.read_csv("/Users/lkhubnani/Downloads/RoadSafety_Casualties.
accidents=pd.read_csv("/Users/lkhubnani/Desktop/challenge/2016/Accidents_201
casualties=pd.read_csv("/Users/lkhubnani/Desktop/challenge/2016/Casualties_2
make_model=pd.read_csv("/Users/lkhubnani/Desktop/challenge/2016/MakeModel201
vehicle=pd.read_csv("/Users/lkhubnani/Desktop/challenge/2016/Vehicle_2016.cs
```

Please provide correct path to the data if you're running it in your enviroment*

In [26]:
```
accidents.head()
```

Out[26]:

	Accident_Index	Location_Easting_OSGR	Location_Northing_OSGR	Longitude	Latitude	Police_F
0	2016010000005	519310.0	188730.0	-0.279323	51.584754	
1	2016010000006	551920.0	174560.0	0.184928	51.449595	
2	2016010000008	505930.0	183850.0	-0.473837	51.543563	
3	2016010000016	527770.0	168930.0	-0.164442	51.404958	
4	2016010000018	510740.0	177230.0	-0.406580	51.483139	

5 rows × 32 columns

In [27]:
```
#Need to replace -1 with NaN to indicate missing values
combined_data.replace(-1, np.nan, inplace=True)
```

In [28]:
```
accidents.columns
```

Out[28]:
```
Index(['Accident_Index', 'Location_Easting_OSGR', 'Location_Northing_OSG
R',
       'Longitude', 'Latitude', 'Police_Force', 'Accident_Severity',
       'Number_of_Vehicles', 'Number_of_Casualties', 'Date', 'Day_of_Wee
k',
       'Time', 'Local_Authority_(District)', 'Local_Authority_(Highway)',
       '1st_Road_Class', '1st_Road_Number', 'Road_Type', 'Speed_limit',
       'Junction_Detail', 'Junction_Control', '2nd_Road_Class',
       '2nd_Road_Number', 'Pedestrian_Crossing-Human_Control',
       'Pedestrian_Crossing-Physical_Facilities', 'Light_Conditions',
       'Weather_Conditions', 'Road_Surface_Conditions',
       'Special_Conditions_at_Site', 'Carriageway_Hazards',
       'Urban_or_Rural_Area', 'Did_Police_Officer_Attend_Scene_of_Acciden
t',
       'LSOA_of_Accident_Location'],
      dtype='object')
```

In [29]:
```
casualties.columns
```

Out[29]:
```
Index(['Accident_Index', 'Vehicle_Reference', 'Casualty_Reference',
       'Casualty_Class', 'Sex_of_Casualty', 'Age_of_Casualty',
       'Age_Band_of_Casualty', 'Casualty_Severity', 'Pedestrian_Locatio
n',
       'Pedestrian_Movement', 'Car_Passenger', 'Bus_or_Coach_Passenger',
       'Pedestrian_Road_Maintenance_Worker', 'Casualty_Type',
       'Casualty_Home_Area_Type', 'Casualty_IMD_Decile'],
      dtype='object')
```

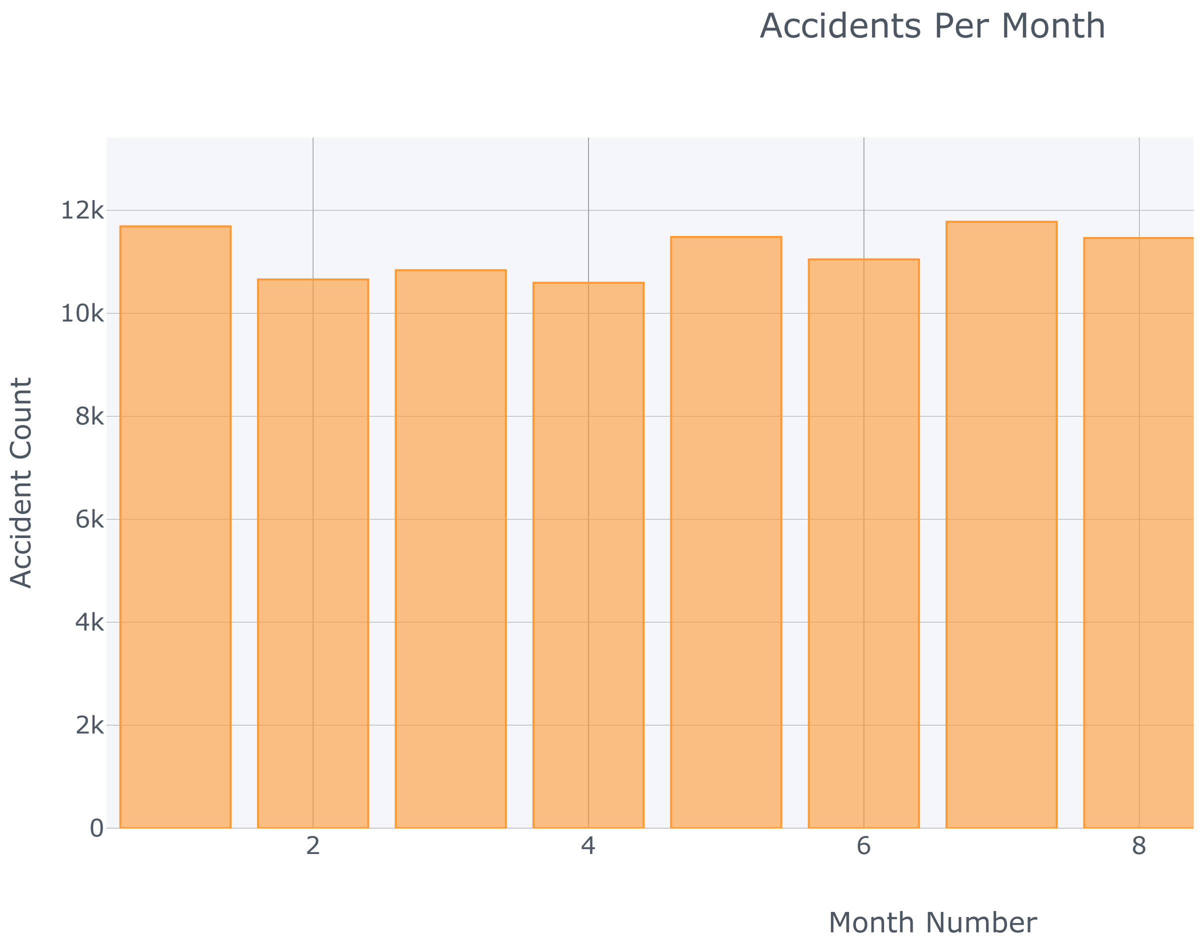
In [30]:

```
combined_data.columns
```

Out[30]:

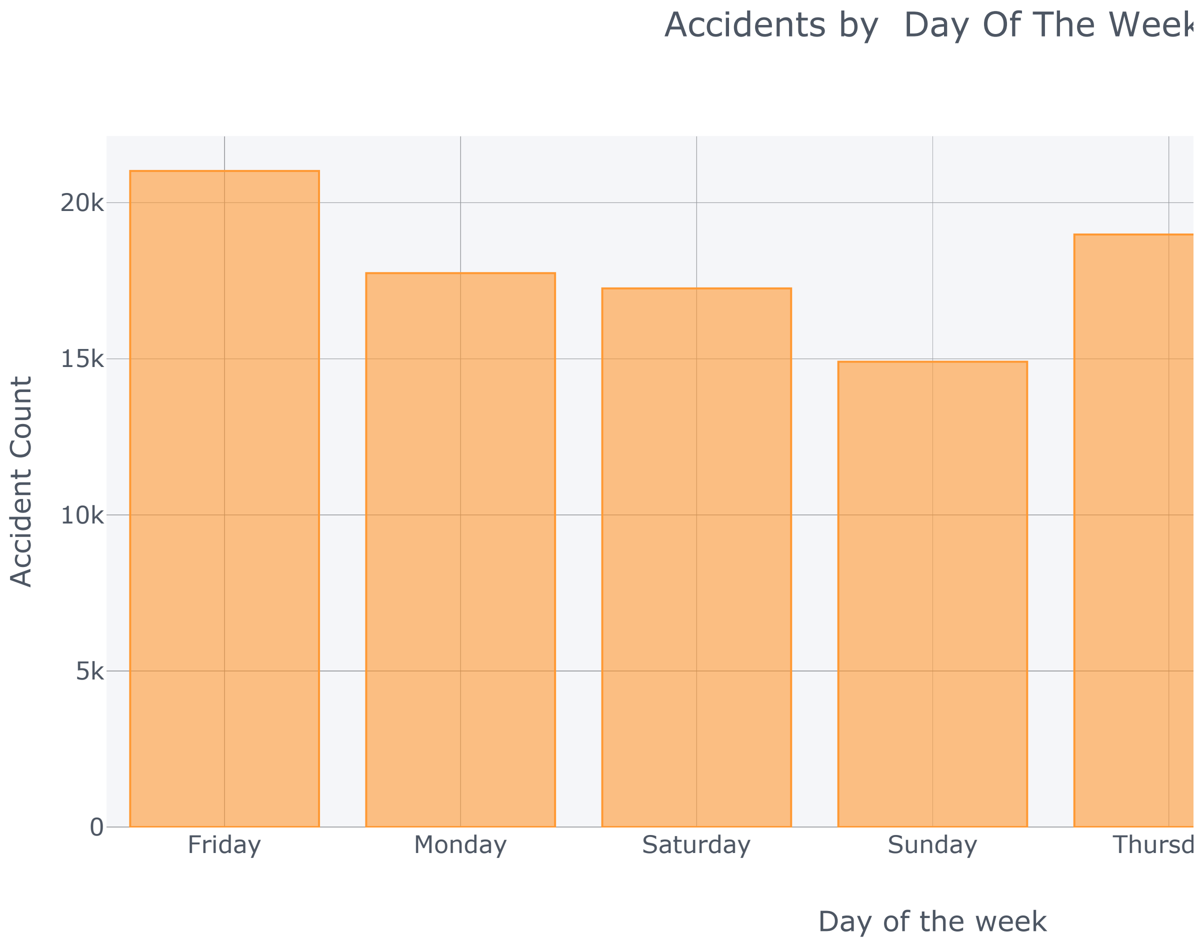
```
Index(['Accident_Index', 'Location_Easting_OSGR', 'Location_Northing_OSG
R',
       'Longitude', 'Latitude', 'Police_Force', 'Accident_Severity',
       'Number_of_Vehicles', 'Number_of_Casualties', 'Date', 'Day_of_Wee
k',
       'Time', 'Local_Authority_(District)', 'Local_Authority_(Highway)',
       '1st_Road_Class', '1st_Road_Number', 'Road_Type', 'Speed_limit',
       'Junction_Detail', 'Junction_Control', '2nd_Road_Class',
       '2nd_Road_Number', 'Pedestrian_Crossing-Human_Control',
       'Pedestrian_Crossing-Physical_Facilities', 'Light_Conditions',
       'Weather_Conditions', 'Road_Surface_Conditions',
       'Special_Conditions_at_Site', 'Carriageway_Hazards',
       'Urban_or_Rural_Area', 'Did_Police_Officer_Attend_Scene_of_Acciden
t',
       'LSOA_of_Accident_Location', 'Vehicle_Reference', 'Casualty_Refere
nce',
       'Casualty_Class', 'Sex_of_Casualty', 'Age_of_Casualty',
       'Age_Band_of_Casualty', 'Casualty_Severity', 'Pedestrian_Locatio
n',
       'Pedestrian_Movement', 'Car_Passenger', 'Bus_or_Coach_Passenger',
       'Pedestrian_Road_Maintenance_Worker', 'Casualty_Type',
       'Casualty_Home_Area_Type', 'Casualty_IMD_Decile', 'Vehicle_Type',
       'Towing_and_Articulation', 'Vehicle_Manoeuvre',
       'Vehicle_Location-Restricted_Lane', 'Junction_Location',
       'Skidding_and_Overturning', 'Hit_Object_in_Carriageway',
       'Vehicle_Leaving_Carriageway', 'Hit_Object_off_Carriageway',
       '1st_Point_of_Impact', 'Was_Vehicle_Left_Hand_Drive?',
       'Journey_Purpose_of_Driver', 'Sex_of_Driver', 'Age_of_Driver',
       'Age_Band_of_Driver', 'Engine_Capacity_(CC)', 'Propulsion_Code',
       'Age_of_Vehicle', 'Driver_IMD_Decile', 'Driver_Home_Area_Type',
       'Vehicle_IMD_Decile', 'accyr', 'Was_Vehicle_Left_Hand_Drive', 'mak
e',
       'model'],
      dtype='object')
```

In [31]:
```
accidents_by_month = accidents.loc[:, 'Date'].groupby(accidents['Date'].map
accidents_by_month = accidents_by_month.to_frame()
accidents_by_month['month'] = accidents_by_month.index
accidents_by_month.columns = ['Count', 'Month']
accidents_by_month
accidents_by_month.iplot(kind='bar', title='Accidents Per Month', x='Month',
```



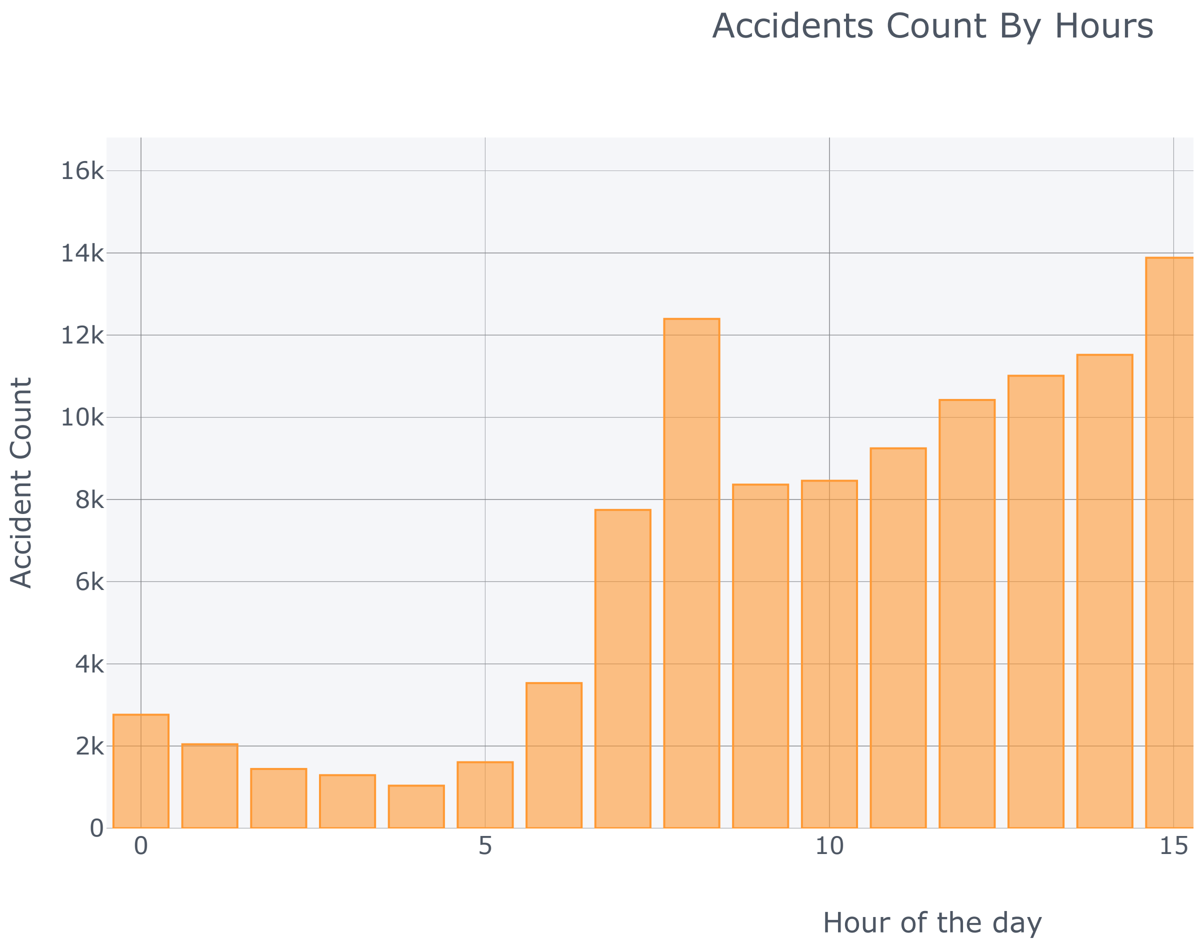
* November has the highest number of accidents and April has the least, There is no clear trend from month to month in the number of accidents

```
In [32]: combined_data.groupby('Day_of_Week').count()['Accident_Index'].iplot(kind='b
```



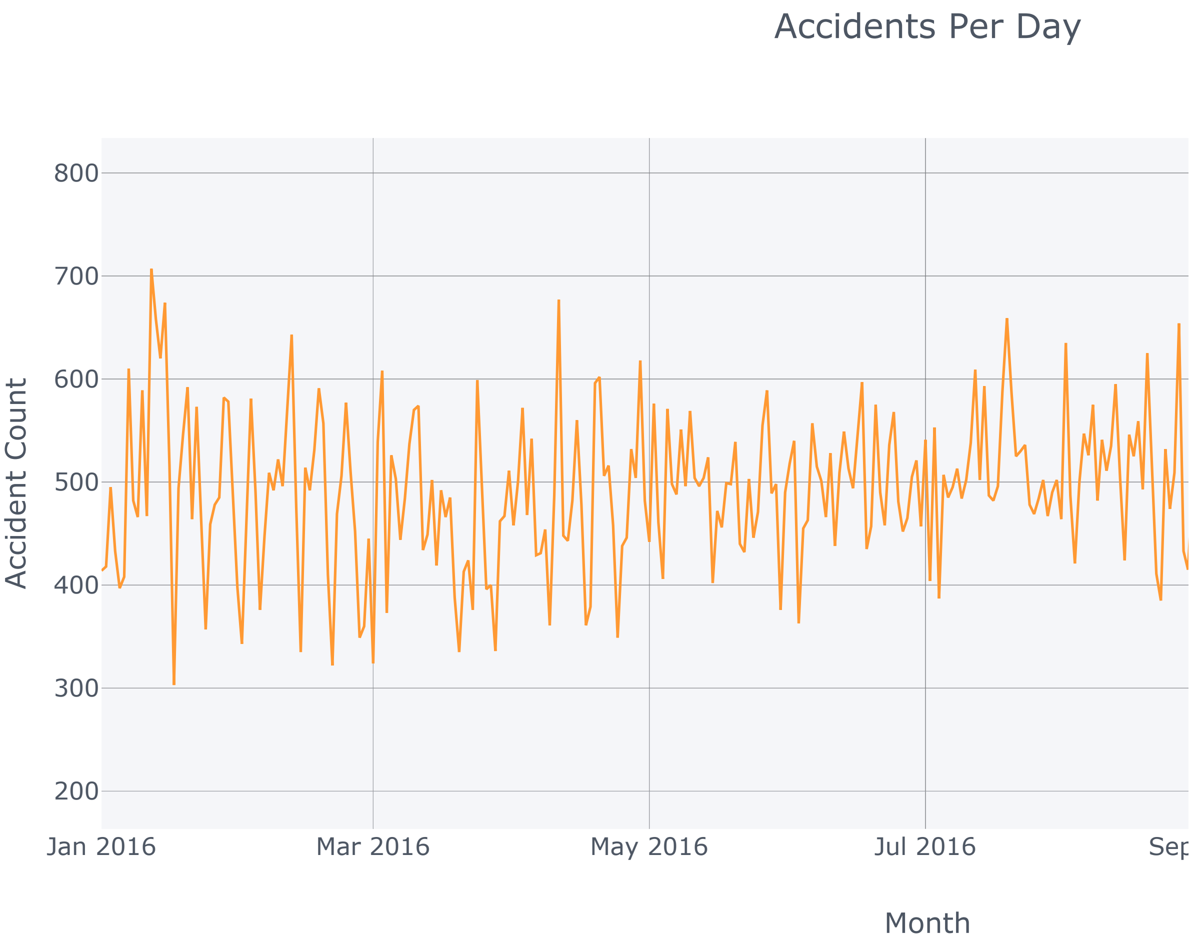
* Friday has the highest number of accidents at 21.017k

In [142]:
```
count_by_hour = combined_data.loc[:, 'Time'].groupby(combined_data['Time'].r
count_by_hour = count_by_hour.to_frame()
count_by_hour['Hour'] = count_by_hour.index
count_by_hour.columns = ['Count', 'Hour']
count_by_hour
count_by_hour.iplot(kind='bar', title='Accidents Count By Hours', x='Hour',
```



* Most Numbers of accidents happened between 3 pm - 6 pm during the evening and 8 am - 9 am during the morning

```
In [34]: # Convert the date string to date object and sort it based on date
         combined_data.Date = pd.to_datetime(combined_data.Date)
         # create temp dataframe with sorted values
         df_by_date = combined_data.iloc[combined_data.Date.sort_values().index]
         # group by date and count, then plot
         collisions_by_date = df_by_date.groupby('Date').Date.count()

         annotations={'2016-12-21':'Holidays Start','2016-12-25':'Christmas'}
         collisions_by_date.iplot(kind='scatter', title='Accidents Per Day', yTitle='
```



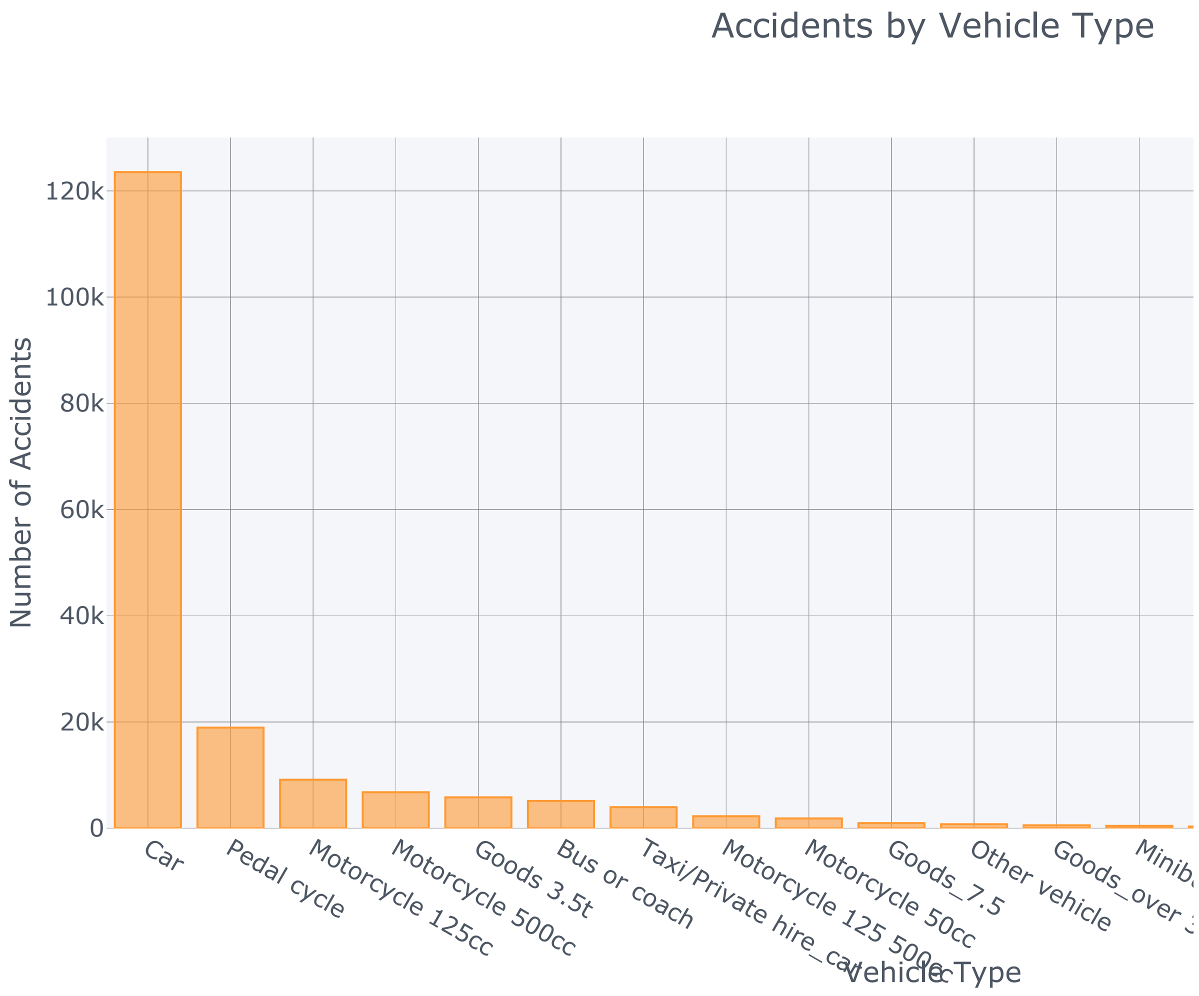
*** One thing we can see is there is a steep decrease in the number of accidents around Dec 21- 25 which might be the result of Christmas Holidays where people usually stay indoors with their families**

In [35]:

```
Vehicle_Type = {1:"Pedal cycle",
2:"Motorcycle 50cc",
3:"Motorcycle 125cc",
4:"Motorcycle 125 500cc",
5:"Motorcycle 500cc",
8:"Taxi/Private hire_car",
9:"Car",
10:"Minibus",
11:"Bus or coach",
16:"Ridden horse",
17:"Agricultural vehicle",
18:"Tram",
19:"Goods 3.5t",
20:"Goods_over 3.5t 7.5t",
21:"Goods_7.5",
22:"Mobility scooter",
23:"Electric motorcycle",
90:"Other vehicle",
97:"Motorcycle - unknown cc",
98:"Goods vehicle - unknown weight",
-1:None}
```

In [36]:

```
combined_data =combined_data.replace({"Vehicle_Type":Vehicle_Type})
# temp_df = combined_data.groupby(Vehicle_Type).Date.count()
series = combined_data['Vehicle_Type'].value_counts()
series.head(3)
series.iplot(kind='bar', yTitle='Number of Accidents', title='Accidents by V
             filename='cufflinks/categorical-bar-chart', xTitle='Vehicle Typ
```

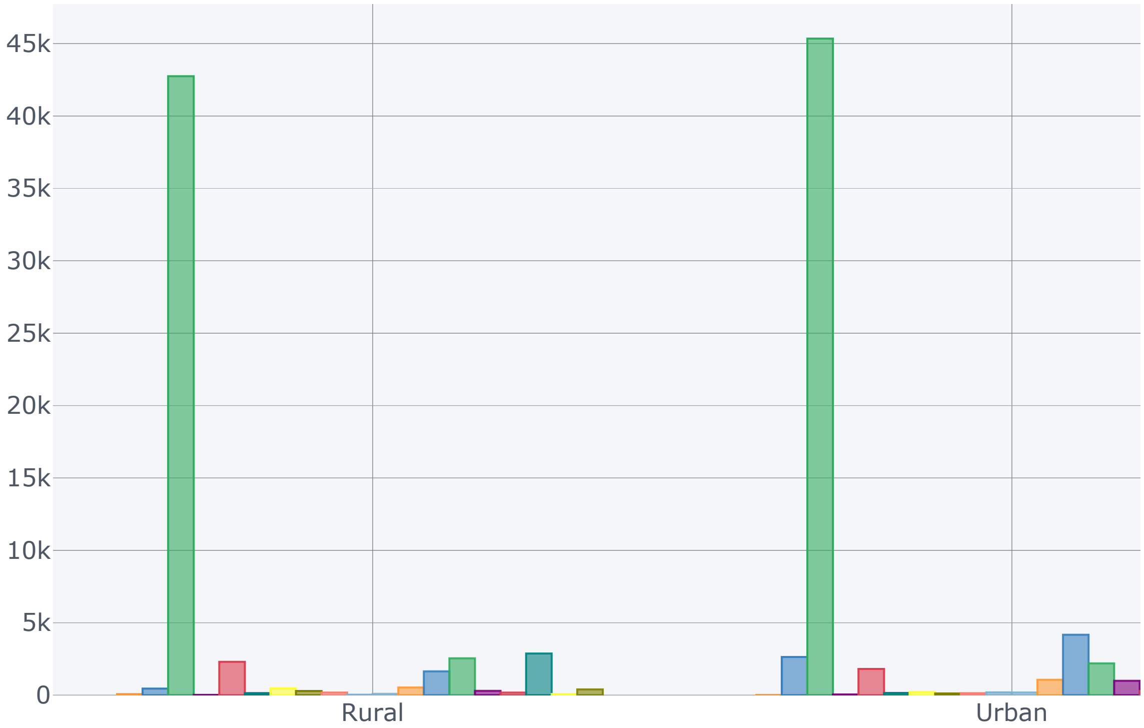


* Car accidents have the highest count (which was be expected), Interestingly pedal cycle constitute the significant amount of accidents as well which wasn't suspected as much.

In [37]:

```
# maping the values
Urban_or_Rural_Area = {1: "Urban", 2:"Rural", 3:"Unallocated"}
combined_data =combined_data.replace({"Urban_or_Rural_Area":Urban_or_Rural_A
```

In [38]:
```
new = combined_data[combined_data['Urban_or_Rural_Area'] != 'Unallocated'].g
new = new.loc[:, ['Accident_Index']]
plot_df = new.unstack('Vehicle_Type').loc[:, 'Accident_Index']
plot_df.iplot(kind='bar', barmode='stacked')
```



* Difference in trends in Urban and Rural areas

- There are more accidents in rural areas related to Motorcycle in 500cc and goods vehicle under 3.5-tonne category
- Motorcycle in under 125cc has more (almost double) accidents in urban areas than rural areas
- Taxis have higher (6 times as much) accident counts in urban areas

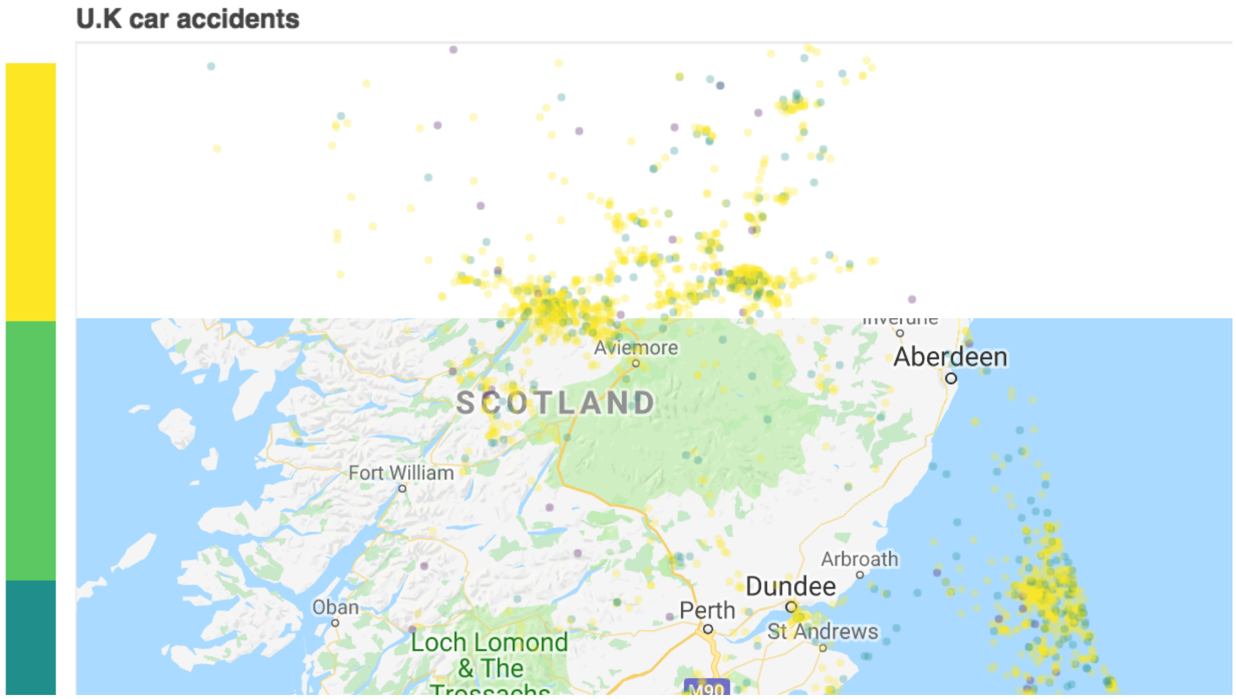
This may be attributed to the distribution of the vehicles in these areas due to their functionality. Further evidence can be collected about the distribution of vehicles in the said areas.

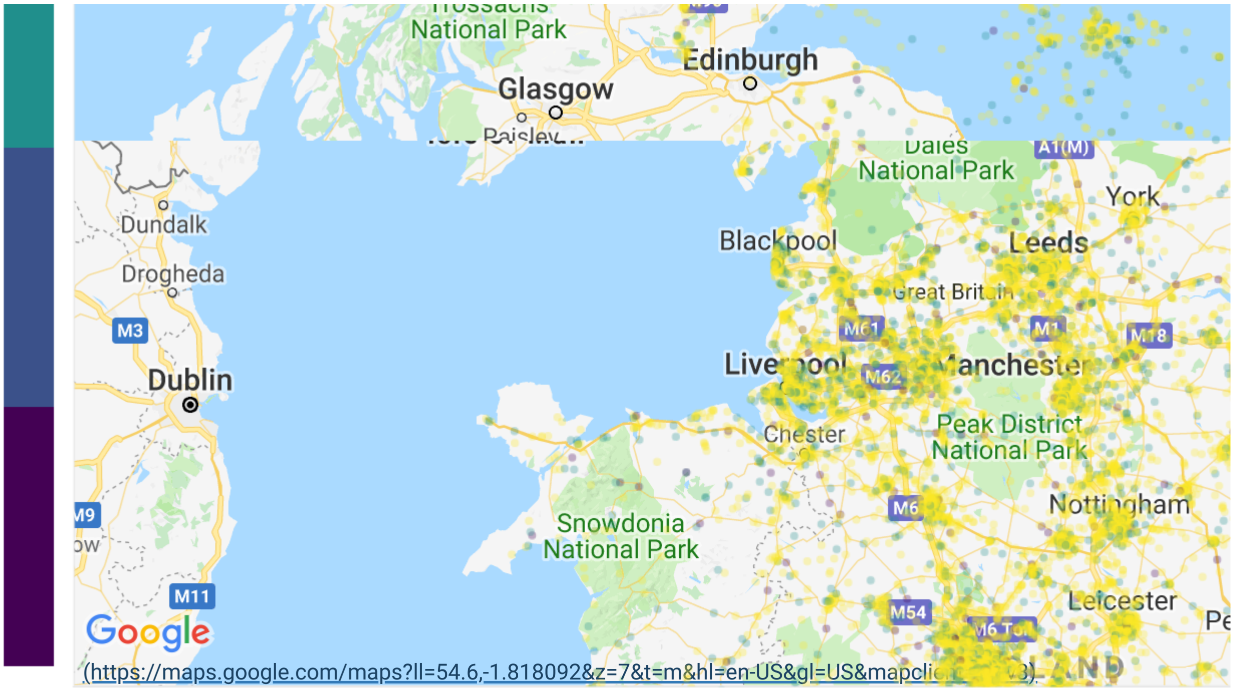
In [126]:

```
from bokeh.io import output_file, output_notebook, show
from bokeh.models import (
  GMapPlot, GMapOptions, ColumnDataSource, Circle, LogColorMapper, BasicTick
    DataRange1d, PanTool, WheelZoomTool, BoxSelectTool
)
from bokeh.models.mappers import ColorMapper, LinearColorMapper
from bokeh.palettes import Viridis5
from bokeh.palettes import YlGn3
```

In [139]:
```
map_options = GMapOptions(lat=54.60, lng=-1.818092, map_type="roadmap", zoor
plot = GMapPlot(
    x_range=DataRange1d(), y_range=DataRange1d(), map_options=map_options, p
)
plot.title.text = "U.K car accidents"
plot.api_key = "AIzaSyD2wt7fzHO5m44C2EjtxuCO7h8XWIabzFQ"
acc_sample = accidents.sample(frac=0.2)
source = ColumnDataSource(
    data=dict(
        lat=acc_sample.Latitude.tolist(),
        lon=acc_sample.Longitude.tolist(),
        size=[x for x in acc_sample.Number_of_Casualties.tolist()],
        color=acc_sample.Accident_Severity.tolist()
    )
)
color_mapper = LinearColorMapper(palette=Viridis5)
circle = Circle(x="lon", y="lat", fill_color={'field': 'color', 'transform':
plot.add_glyph(source, circle)

color_bar = ColorBar(color_mapper=color_mapper, ticker=BasicTicker(),
                     label_standoff=12, border_line_color=None, location=(0,
plot.add_layout(color_bar, 'left')

plot.add_tools(PanTool(), WheelZoomTool(), BoxSelectTool(), HoverTool())
#output_file("gmap_plot.html")
hover = plot.select_one(HoverTool)
hover.point_policy = "follow_mouse"
hover.tooltips = [
    ("Number of Casualties", "@size")
]
output_notebook()

show(plot)
```

(http://bokeh.pydata.org) BokehJS 0.12.5 successfully loaded.




Trossachs National Park
Edinburgh
Glasgow
Paisley
Dales National Park
A1(M)
York
Dundalk
Drogheda
Blackpool
Leeds
Great Britain
M3
M61
M1
M18
Dublin
Liverpool
M62
Manchester
Chester
Peak District National Park
M9
M6
Nottingham
Snowdonia National Park
M11
M54
M6 Toll
Leicester
Google
ENGLAND
(https://maps.google.com/maps?ll=54.6,-1.818092&z=7&t=m&hl=en-US&gl=US&mapclie...).