Mounting drive and import statements

- Mount google drive and import libraries

```
## To load up drive

%cd drive/MyDrive/CSE_519_assignment/hw3
```

```
/content/drive/MyDrive/CSE_519_assignment/hw3
```

```
## Import statements

import numpy as np
import pandas as pd
import seaborn as sns
import os
from sklearn import metrics
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import datetime
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor, RandomForestClassifier
from sklearn.model_selection import permutation_test_score
from scipy import stats

root_dir = os.getcwd()
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

Q1Reading store data and train data, visualization and data combination

- Read store data, train data, test data.
- Check missing elements and handle them(Drop, fill with mean).
- Merge store data with train data into a single dataframe.

```
store_df = pd.read_csv(root_dir + "/store.csv")
```

```
train_df = pd.read_csv(root_dir + "/train.csv")
```

```
/usr/local/lib/python3.7/dist-packages/IPython/core/interactiveshell.py:2718:
  interactivity=interactivity, compiler=compiler, result=result)
```

✓ 0s completed at 1:59 AM

```
train_df.shape
```

```
(1017209, 9)
```

```
store_df.head()
```

	Store	StoreType	Assortment	CompetitionDistance	CompetitionOpenSinceMont
0	1	c	a	1270.0	9.
1	2	a	a	570.0	11.
2	3	a	a	14130.0	12.
3	4	c	c	620.0	9.
4	5	a	a	29910.0	4.

```
train_df.head()
```

	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	Sc
0	1	5	2015-07-31	5263	555	1	1	0	
1	2	5	2015-07-31	6064	625	1	1	0	
2	3	5	2015-07-31	8314	821	1	1	0	
3	4	5	2015-07-31	13995	1498	1	1	0	
4	5	5	2015-07-31	4822	559	1	1	0	

```
train_df.describe()
```

	Store	DayOfWeek	Sales	Customers	Open
count	1.017209e+06	1.017209e+06	1.017209e+06	1.017209e+06	1.017209e+06
mean	5.584297e+02	3.998341e+00	5.773819e+03	6.331459e+02	8.301067e-01
std	3.219087e+02	1.997391e+00	3.849926e+03	4.644117e+02	3.755392e-01
min	1.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00

count	1115.00000	1112.000000	761.000000
mean	558.00000	5404.901079	7.224704
std	322.01708	7663.174720	3.212348
min	1.00000	20.000000	1.000000
25%	279.50000	717.500000	4.000000
50%	558.00000	2325.000000	8.000000
75%	836.50000	6882.500000	10.000000
max	1115.00000	75860.000000	12.000000

```
missing = train_df.isnull().sum()
missing.sort_values(ascending=False)
```

```
SchoolHoliday    0
StateHoliday     0
Promo            0
Open             0
Customers        0
Sales            0
Date             0
DayOfWeek        0
Store            0
dtype: int64
```

```
train_df['SalesPerCustomer'] = train_df['Sales']/train_df['Customers']
```

```
train_df.dropna(inplace=True)
```

```
store_df.isnull().sum()
```

```
Store                        0
StoreType                    0
Assortment                   0
```

0	1	c	a	1270.0	9.
1	2	a	a	570.0	11.
2	3	a	a	14130.0	12.
3	4	c	c	620.0	9.
4	5	a	a	29910.0	4.

```
train_df = train_df.merge(right=store_df, on='Store', how='left')
```

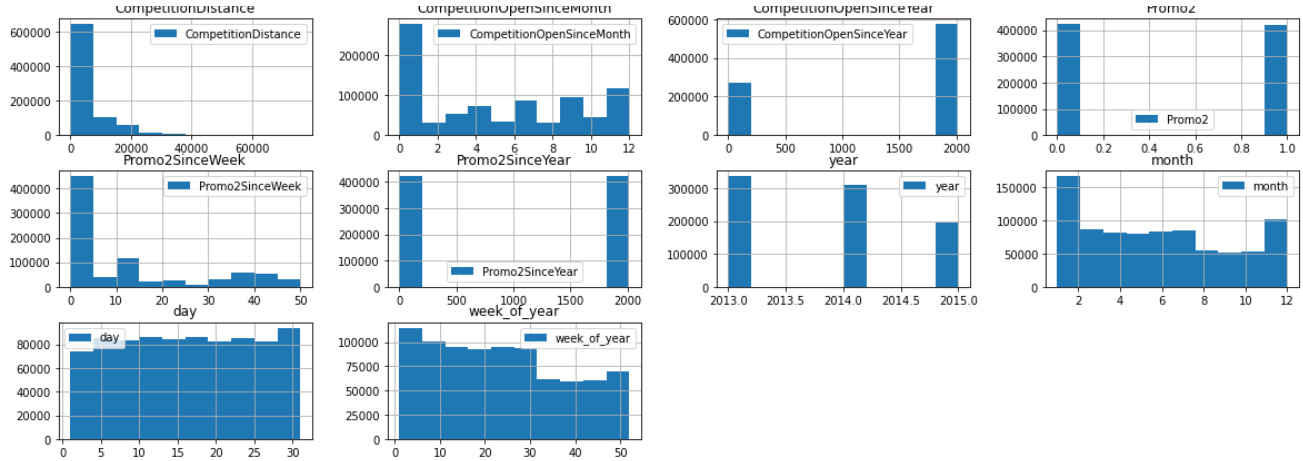
```
train_df.shape
```

```
(844340, 19)
```

```
train_df.head()
```

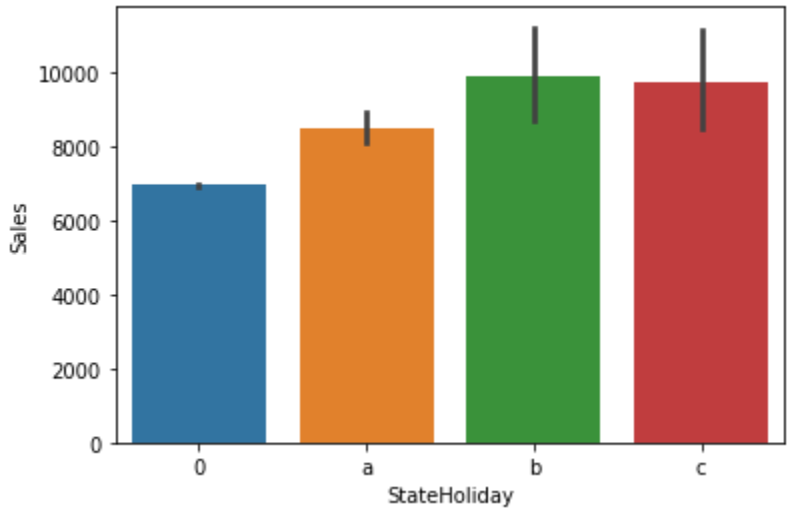
	Store	DayOfWeek	Date	Sales	Customers	Open	Promo	StateHoliday	Sc
0	1	5	2015-07-31	5263	555	1	1	0	
1	2	5	2015-07-31	6064	625	1	1	0	
2	3	5	2015-07-31	8314	821	1	1	0	
3	4	5	2015-07-31	13995	1498	1	1	0	
4	5	5	2015-07-31	4822	559	1	1	0	

Converting string of date to datetime object and segregating to


CompetitionDistance
CompetitionDistance
CompetitionOpenSinceMonth
CompetitionOpenSinceMonth
CompetitionOpenSinceYear
CompetitionOpenSinceYear
Promo2
Promo2
Promo2SinceWeek
Promo2SinceWeek
Promo2SinceYear
Promo2SinceYear
year
year
month
month
day
day
week_of_year
week_of_year

```
## Holidays + no_holidays
sns.barplot(x='StateHoliday', y='Sales', data=train_df)
```

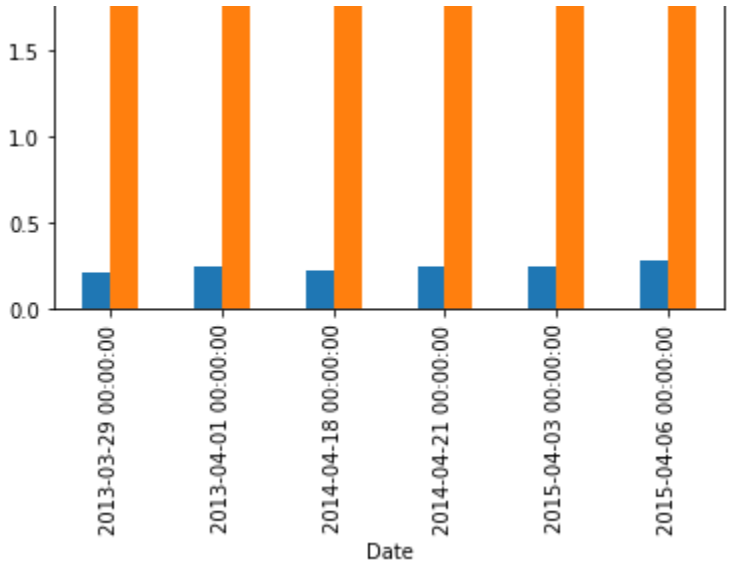
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff752246ad0>
```



4	2013-01-05	0	5951593

```
holidays_df = sales_per_day[sales_per_day['StateHoliday'] != "0"]
holiday_a = holidays_df[holidays_df['StateHoliday']=='a']
holiday_b = holidays_df[holidays_df['StateHoliday']=='b']
holiday_c = holidays_df[holidays_df['StateHoliday']=='c']
non_holidays = sales_per_day[sales_per_day['StateHoliday'] == "0"]
```

Helper function to check 7 days before holiday and plot the sales against the sales of the holiday


1.5
1.0
0.5
0.0
2013-03-29 00:00:00
2013-04-01 00:00:00
2014-04-18 00:00:00
2014-04-21 00:00:00
2015-04-03 00:00:00
2015-04-06 00:00:00
Date

3	1	2013-01-05	4997