```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
sns.set()
```

```
df = pd.read_csv('processed_data.csv', index_col=0, parse_dates=True)
```

```
<ipython-input-7-1cc9c1ee5d13>:1: UserWarning: Could not infer format, so each element will be parsed individually, falling back to
  df = pd.read_csv('processed_data.csv', index_col=0, parse_dates=True)
```

```
df.head(5)
```

	Date	TotalBikesCount	Mon	Tue	Wed	Thu	Fri	Sat	Sun	holiday	DayLightHrs	AvgTempInC	PRCP_IN	DryDay	YearsCount
0	2012-10-03	3521.0	0	0	1	0	0	0	0	0	11.277359	5.60	0.0	1	0.000000
1	2012-10-04	3475.0	0	0	0	1	0	0	0	0	11.219142	5.65	0.0	1	0.002740
2	2012-10-05	3148.0	0	0	0	0	1	0	0	0	11.161038	5.95	0.0	1	0.005479
3	2012-10-	2006.0	0	0	0	0	0	1	0	0	11.103056	6.05	0.0	1	0.008219

Next steps: Generate code with df View recommended plots

```
df['Date'] = pd.to_datetime(df['Date'])
df.dtypes
```

```
Date              datetime64[ns]
TotalBikesCount          float64
Mon                        int64
Tue                        int64
Wed                        int64
Thu                        int64
Fri                        int64
Sat                        int64
Sun                        int64
holiday                    int64
DayLightHrs              float64
AvgTempInC               float64
PRCP_IN                  float64
DryDay                     int64
YearsCount               float64
dtype: object
```

```
df = df.set_index('Date')
df.head(5)
```

Date	TotalBikesCount	Mon	Tue	Wed	Thu	Fri	Sat	Sun	holiday	DayLightHrs	AvgTempInC	PRCP_IN	DryDay	YearsCount
2012-10-03	3521.0	0	0	1	0	0	0	0	0	11.277359	5.60	0.0	1	0.000000
2012-10-04	3475.0	0	0	0	1	0	0	0	0	11.219142	5.65	0.0	1	0.002740
2012-10-05	3148.0	0	0	0	0	1	0	0	0	11.161038	5.95	0.0	1	0.005479
2012-10-06	2006.0	0	0	0	0	0	1	0	0	11.103056	6.05	0.0	1	0.008219
2012-10-07	2142.0	0	0	0	0	0	0	1	0	11.045208	6.05	0.0	1	0.010959

Next steps: Generate code with df View recommended plots

```
df.shape
```

```
(2097, 14)
```

```
df.columns
```

```
Index(['TotalBikesCount', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun',
       'holiday', 'DayLightHrs', 'AvgTempInC', 'PRCP_IN', 'DryDay',
       'YearsCount'],
      dtype='object')
```

```
indep_cols = ['Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun',
       'holiday', 'DayLightHrs', 'AvgTempInC', 'PRCP_IN', 'DryDay','YearsCount']
x = df[indep_cols]
y = df['TotalBikesCount']


from sklearn.linear_model import LinearRegression

model = LinearRegression(fit_intercept=False)
model.fit(x, y)
```

```
LinearRegression
LinearRegression(fit_intercept=False)
```

```
predictions = model.predict(x)
predictions
```

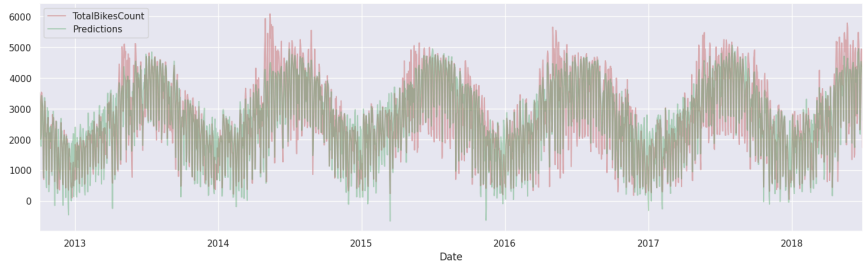
```
array([3509.10208841, 3375.37680672, 3171.1559462 , ..., 4294.78165878,
       4075.72089562, 2626.74466652])
```

```
df1= df[['TotalBikesCount']].copy()
df1['Predictions'] = predictions
df1.head(5)
```

Date	TotalBikesCount	Predictions
2012-10-03	3521.0	3509.102088
2012-10-04	3475.0	3375.376807
2012-10-05	3148.0	3171.155946
2012-10-06	2006.0	1859.059961
2012-10-07	2142.0	1776.258124

Next steps: Generate code with df1 View recommended plots

```
import datetime as dt
df1.plot(alpha=0.4, figsize=(18,5), color=['r','g'])
plt.legend(loc='upper left')
plt.show()
```




```
model.coef_
```

```
array([-6.54065823e+02, -5.26114967e+02, -5.40396337e+02, -6.88055804e+02,
       -1.00778818e+03, -2.35415766e+03, -2.43062192e+03, -1.20007497e+03,
        1.11462695e+02,  4.06258710e+02, -1.75737653e+05,  5.17444792e+02,
        4.02758119e+01])
```

```
coeffs = pd.Series(model.coef_, index=x.columns)
coeffs
```

```
Mon              -654.065823
Tue              -526.114967
Wed              -540.396337
Thu              -688.055804
Fri             -1007.788184
Sat             -2354.157663
```

```
Sun             -2430.621920
holiday         -1200.074973
DayLightHrs       111.462695
AvgTempInC        406.258710
PRCP_IN       -175737.652995
DryDay            517.444792
YearsCount         40.275812
dtype: float64
```

```
from sklearn.utils import resample
model.fit(x, y)
print('model.coef_',model.coef_)
model.fit(*resample(x, y))
print('model.coef_',model.coef_)
model.fit(*resample(x, y))
print('model.coef_',model.coef_)

temp = np.asarray([
                [1,1,2,3,4], # 1st row
                [2,6,7,8,9], # 2nd row
                [3,6,7,8,9], # 3rd row
                [4,6,7,8,9], # 4th row
                [5,6,7,8,9]  # 5th row
              ])
print('temp :',temp)
print('temp resampled :',resample(temp))
```

```
model.coef_ [-6.54065823e+02 -5.26114967e+02 -5.40396337e+02 -6.88055804e+02
 -1.00778818e+03 -2.35415766e+03 -2.43062192e+03 -1.20007497e+03
  1.11462695e+02  4.06258710e+02 -1.75737653e+05  5.17444792e+02
  4.02758119e+01]
model.coef_ [-6.99570531e+02 -5.49344506e+02 -5.83341821e+02 -7.22021089e+02
 -1.01762675e+03 -2.36544042e+03 -2.43334573e+03 -1.07874228e+03
  1.17115541e+02  3.93574558e+02 -1.74988796e+05  5.29516756e+02
  4.30482149e+01]
model.coef_ [-7.24812775e+02 -6.29592600e+02 -5.92059710e+02 -7.30519042e+02
 -1.10752008e+03 -2.44077290e+03 -2.49590039e+03 -1.18347509e+03
  1.05351519e+02  4.26924463e+02 -1.66634668e+05  5.42060170e+02
  4.91725625e+01]
temp : [[1 1 2 3 4]
 [2 6 7 8 9]
 [3 6 7 8 9]
 [4 6 7 8 9]
 [5 6 7 8 9]]
temp resampled : [[4 6 7 8 9]
 [3 6 7 8 9]
 [1 1 2 3 4]
 [4 6 7 8 9]
 [2 6 7 8 9]]
```

```
from sklearn.utils import resample
np.random.seed(1)
coeffs_tmp = [model.fit(*resample(x, y)).coef_ for i in range(1000)]
err = np.std(coeffs_tmp,0) #Columnar STD


# With these errors estimated, let's again look at the results:
print(pd.DataFrame({'effect': coeffs.round(0),
                    'error': err.round(0)}))
```

```
                effect    error
Mon             -654.0     66.0
Tue             -526.0     69.0
Wed             -540.0     67.0
Thu             -688.0     67.0
Fri            -1008.0     66.0
Sat            -2354.0     64.0
Sun            -2431.0     64.0
holiday        -1200.0    114.0
DayLightHrs      111.0      7.0
AvgTempInC       406.0     16.0
PRCP_IN     -175738.0  11018.0
DryDay           517.0     25.0
YearsCount        40.0      6.0
```