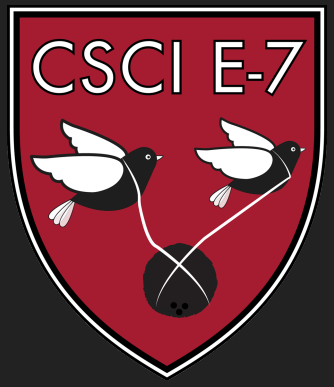


STEVE DELANO

DRIVE FOR SHOW, PUTT FOR DOUGH?

CSCI E-7 SPRING 2017 STAFF

Nenad Svrzikapa
Instructor

Lena Hajjar
Teaching Fellow

Jose Herran
Teaching Fellow

Kaleigh Douglas
Teaching Fellow

Alan Xie
Teaching Fellow

Joe Palin
Teaching Fellow

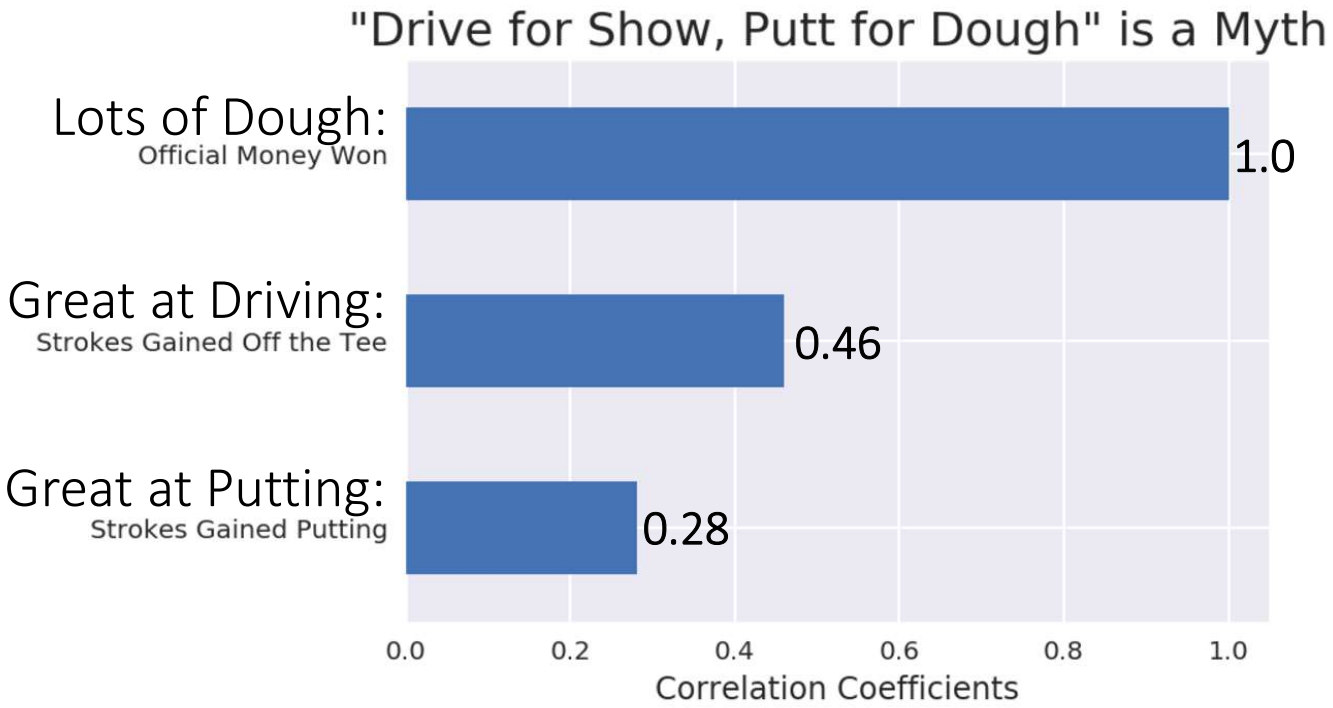
ABSTRACT

- DRIVE FOR SHOW, PUTT FOR DOUGH?
- WHAT OTHER ASPECT OF GOLF CORRELATES WITH WINNING?
- DISPLAY SUMMARY CHARTS TO ANSWER THE QUESTIONS
- REVIEW PROCESS, CODE AND DATA

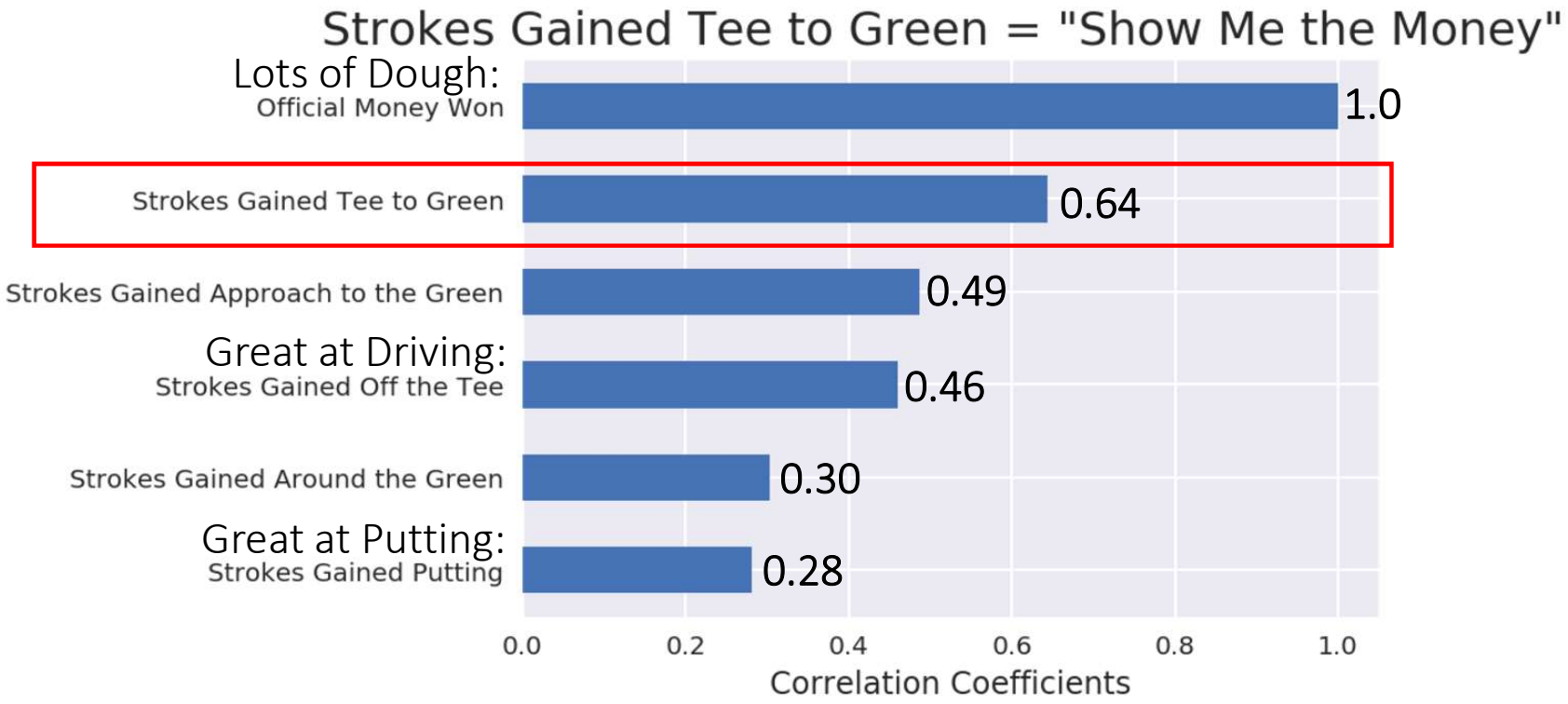
REQUIRED LIBRARIES AND FILES

- PANDAS
- NUMPY
- SEABORN
- MATPLOTLIB.PYPLOT
- SCIPY.STATS
- WARNINGS
- BOKEH
- golf.csv contains data
- FILES AT: https://github.com/stevedelano/CSCI-E-7

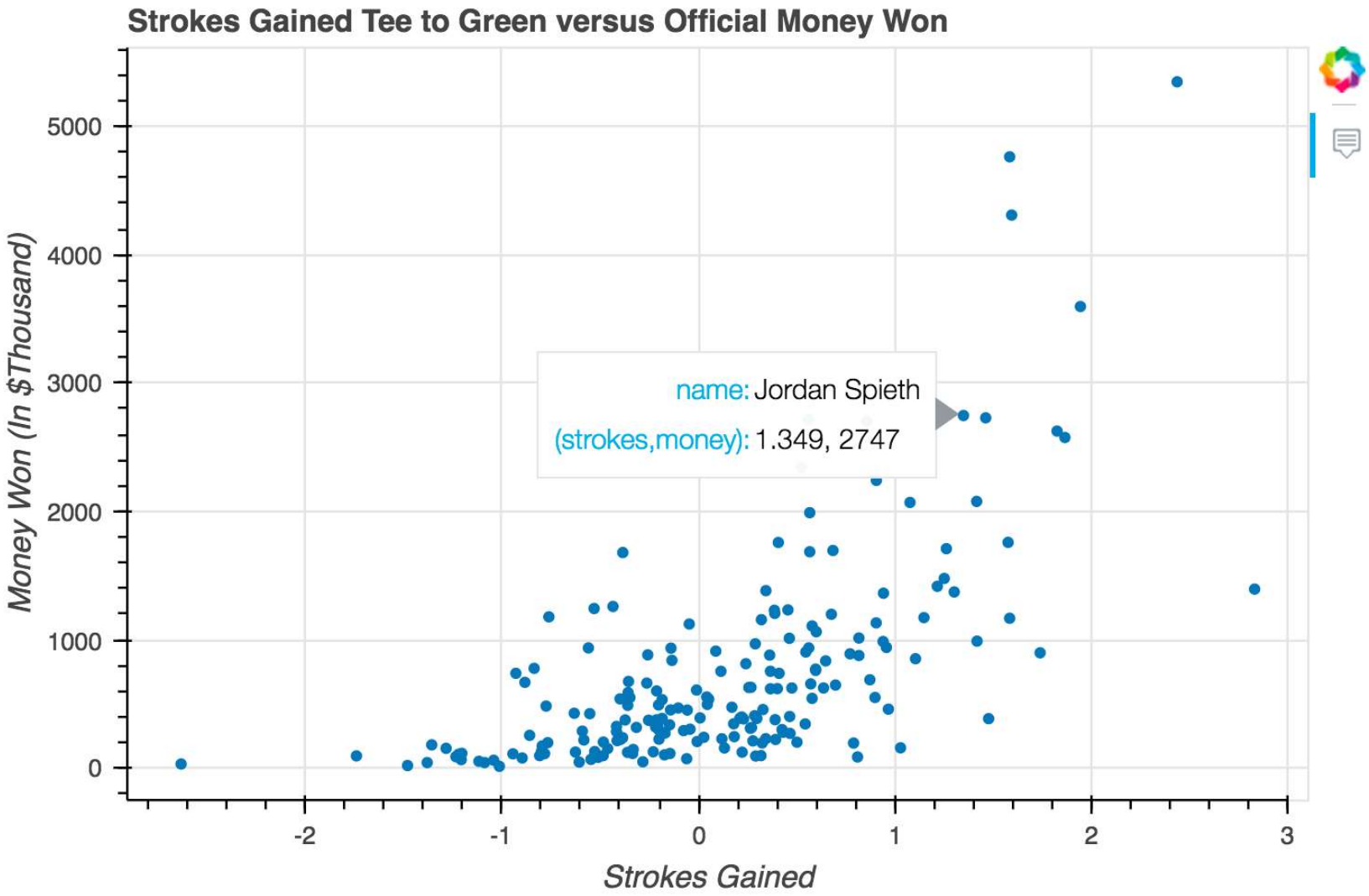
Data Does Not Support the Expression



So, What Matters?



Bokeh HoverTool is Cool



Drive for Show, Putt for Dough? Review Process, Code and Data

1. Locate dataset of golf statistics

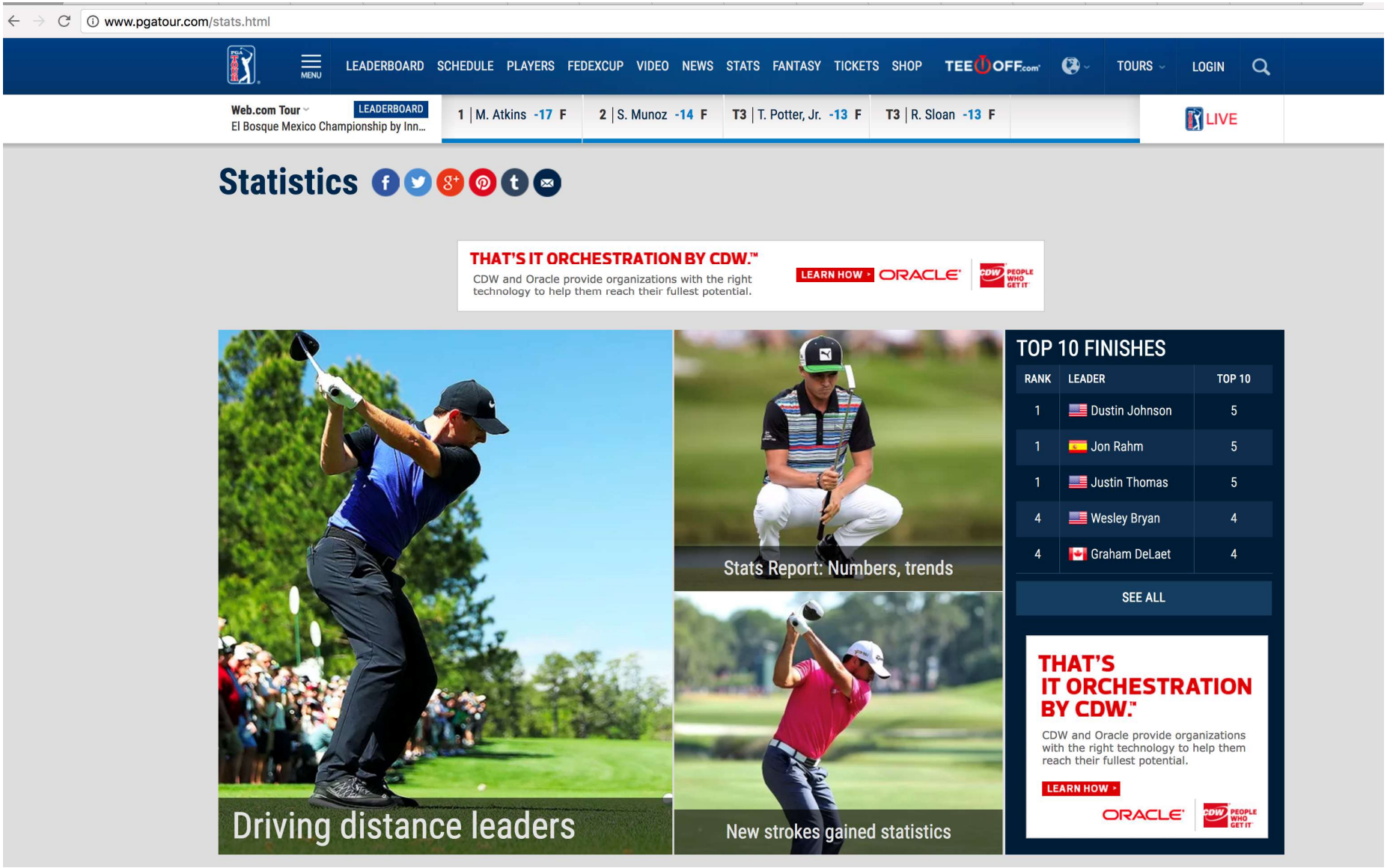
2. Identify key libraries and import them

3. Read and explore data; clean if necessary

4. Analyze the data

5. Create key data visualizations to tell a story

1. Locate Dataset of Golf Statistics



1. Locate Dataset of Golf Statistics

Thanks, Jimmy!

2. Identify Key Libraries and Import Them

2. Identify Key Libraries and Import Them

```
import solution
```

2. Identify Key Libraries and Import Them

```
import solution
```

```
---------------------------------------------------------------------------
--
ImportError                               Traceback (most recent call las
t)
<ipython-input-1-5fc6bd43c1ae> in <module>()
----> 1 import solution

ImportError: No module named 'solution'
```

Nenad would say: "Come on guys. You have to try harder than that!"

2. Identify Key Libraries and Import Them

Ok, back to the drawing board.

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as stats
import warnings
from bokeh.plotting import figure, output_file, show, ColumnDataSource
from bokeh.models import HoverTool
from bokeh.io import output_notebook, show

# this line tells jupyter notebook to put the plots in the notebook.
%matplotlib inline

# this line makes plots prettier on mac retina screens.
%config InlineBackend.figure_format = 'retina'
```

3. Read and Explore Data; Clean if Necessary

```
# read the csv file into a pandas dataframe
golf = pd.read_csv("golf.csv")
```

```
# get some basic info on the dataframe
golf.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 194 entries, 0 to 193
Data columns (total 21 columns):
PLAYER NAME                                194 non-null object
FEDEX CUP POINTS                           194 non-null int64
AVERAGE DRIVING DISTANCE                   194 non-null float64
Strokes Gained Off the Tee                 194 non-null float64
Strokes gained tee to green                194 non-null float64
Strokes gained approach to the green       194 non-null float64
Strokes gained around the green            194 non-null float64
Strokes gained putting                     194 non-null float64
Strokes gained total                       194 non-null float64
BIRDIE TO BOGEY RATIO                      194 non-null float64
birdie or better % from the rough          194 non-null float64
Scrambling average distance to the hole    194 non-null float64
World Ranking                              194 non-null int64
Scoring Average                            194 non-null float64
Percent of Available Purse Won             194 non-null float64
Official Money Won                         194 non-null int64
Unnamed: 16                                0 non-null float64
Unnamed: 17                                0 non-null float64
Unnamed: 18                                0 non-null float64
Unnamed: 19                                0 non-null float64
Unnamed: 20                                0 non-null float64
dtypes: float64(17), int64(3), object(1)
memory usage: 31.9+ KB
```

3. Read and Explore Data; Clean if Necessary

```
# rename columns for consistent format
golf = golf.rename(columns = {'FEDEX CUP POINTS':"FedEx Cup Points",
                              'AVERAGE DRIVING DISTANCE':'Average Driving Distance',
                              'Strokes gained tee to green':'Strokes Gained Tee to Green',
                              'Strokes gained approach to the green':'Strokes Gained Approach to the Green',
                              'Strokes gained around the green':'Strokes Gained Around the Green',
                              'Strokes gained putting':'Strokes Gained Putting',
                               'Strokes gained total':'Strokes Gained Total',
                              'BIRDIE TO BOGEY RATIO':'Birdie to Bogey Ratio',
                              'biride or better % from the rough':'Birdie or Better % From the Rough',
                              'Scrambling average distance to the hole':'Scrambling Average Distance to the Hole'})


#remove blank columns
golf = golf.drop('Unnamed: 16',axis=1)
golf = golf.drop('Unnamed: 17',axis=1)
golf = golf.drop('Unnamed: 18',axis=1)
golf = golf.drop('Unnamed: 19',axis=1)
golf = golf.drop('Unnamed: 20',axis=1)
```

```
golf.head(2)
```

	PLAYER NAME	FedEx Cup Points	Average Driving Distance	Strokes Gained Off the Tee	Strokes Gained Tee to Green	Strokes Gained Approach to the Green	Strokes Gained Around the Green	Strokes Gained Putting	Strokes Gained Total	Birdie to Bogey Ratio	birdie or better % from the rough	Scrambling Average Distance to the Hole	World Ranking	Scoring Average	Percent of Available Purse Won	Offi Mo Wo
0	Aaron Baddeley	167	291.6	-0.618	-0.184	0.021	0.424	-0.141	-0.315	1.28	24.58	7.08	136	71.940	0.48	388

4. Analyze the Data

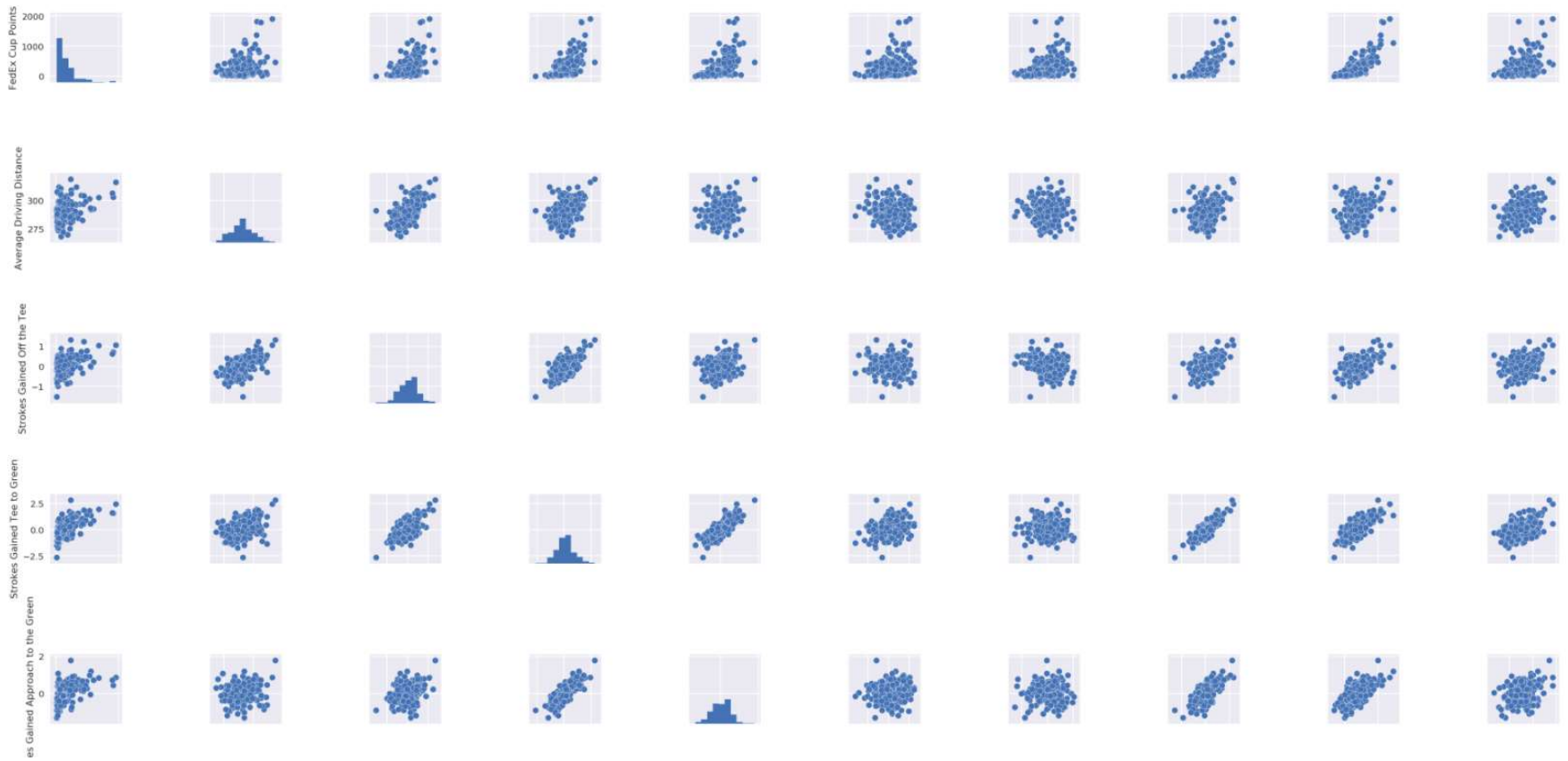
```
# start with key summary statistics
golf.describe()
```

	FedEx Cup Points	Average Driving Distance	Strokes Gained Off the Tee	Strokes Gained Tee to Green	Strokes Gained Approach to the Green	Strokes Gained Around the Green	Strokes Gained Putting	Strokes Gained Total
count	194.000000	194.000000	194.000000	194.000000	194.000000	194.000000	194.000000	194.000000
mean	310.365979	290.468041	0.056495	0.120603	0.045392	0.019933	-0.000907	0.120959
std	320.255476	9.500581	0.435994	0.799866	0.485720	0.274501	0.424551	0.900425
min	6.000000	268.300000	-1.525000	-2.631000	-1.298000	-0.818000	-1.268000	-3.271000
25%	103.250000	284.525000	-0.205250	-0.384750	-0.292000	-0.162750	-0.273750	-0.425750
50%	204.500000	290.550000	0.093000	0.126500	0.033000	0.022000	0.008500	0.079500
75%	408.250000	296.875000	0.347750	0.567000	0.386750	0.198000	0.288000	0.650000
max	1903.000000	318.800000	1.342000	2.830000	1.783000	0.622000	1.023000	2.953000

4. Analyze the Data

```
# this generates the pairplots
sns.pairplot(golf)
```

```
<seaborn.axisgrid.PairGrid at 0x7f4b2c90ee48>
```

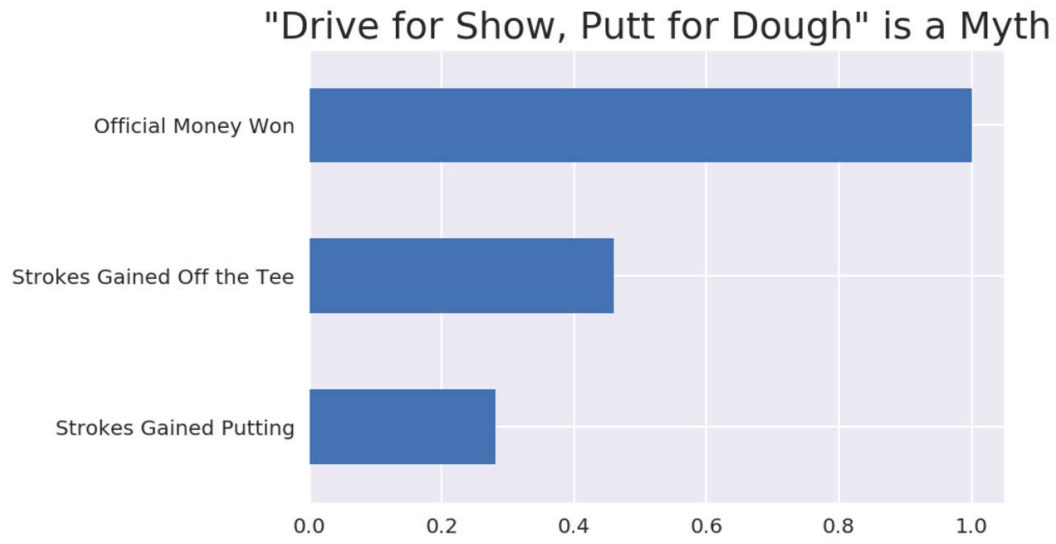


5. Create Key Data Visualizations to Tell a Story

"Drive for Show, Putt for Dough" is a Myth

```
# create a new DataFrame with just Money Won, Driving, Putting
golf_chart = golf
golf_chart = golf_chart.drop('Strokes Gained Approach to the Green',axis=1)
golf_chart = golf_chart.drop('Strokes Gained Around the Green',axis=1)
golf_chart = golf_chart.drop('Strokes Gained Tee to Green',axis=1)

# dataframe with correlation data
golf_chart_corr = golf_chart.corr()
```

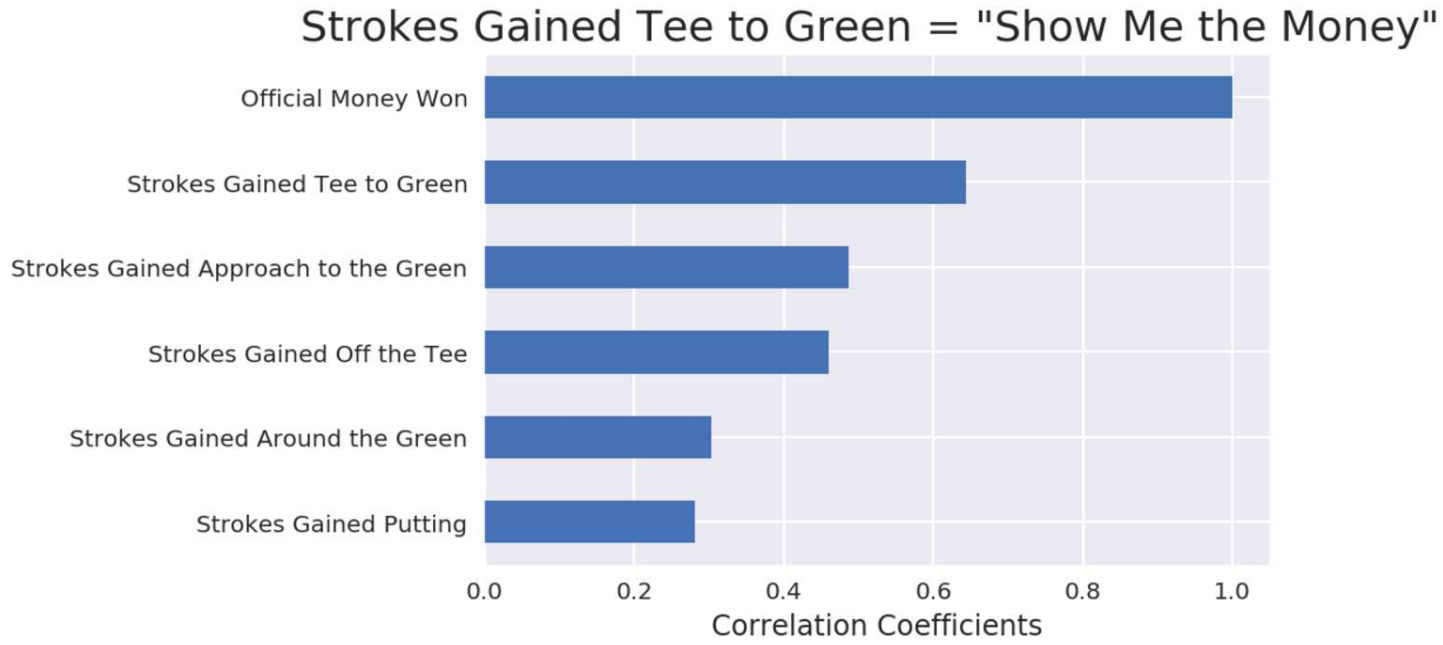
```
# output a simple bar chart of correlations with catchy title

ax = golf_chart_corr['Official Money Won'].sort_values(ascending=True).plot.barh()

ax.set_xlabel('Correlation Coefficients', fontsize=12)
ax.set_title('"Drive for Show, Putt for Dough" is a Myth', fontsize=18)

plt.show()
```



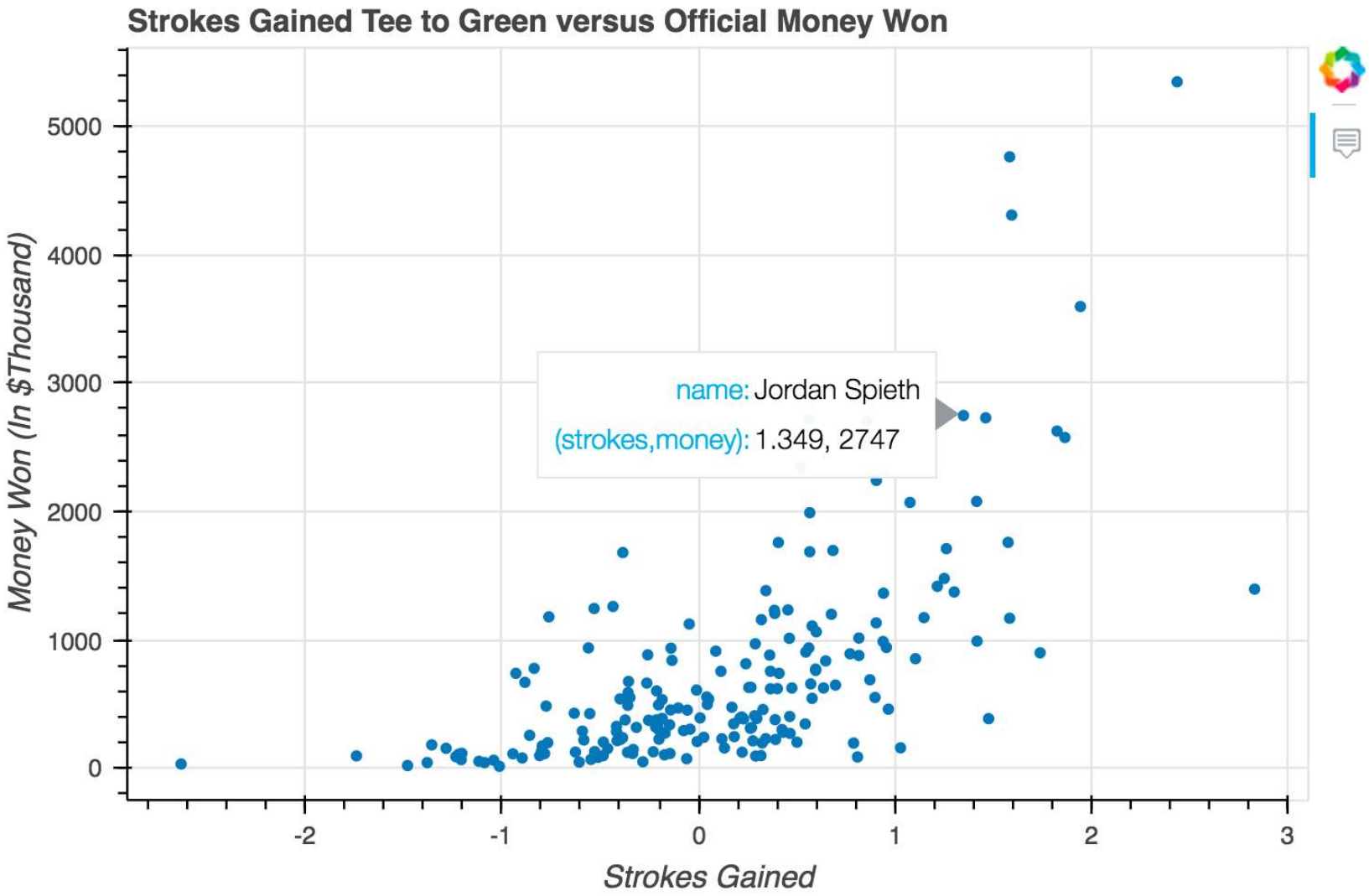
5. Create Key Data Visualizations to Tell a Story

Strokes Gained Tee to Green = "Show Me the Money"

```
# output a simple bar chart of correlations with catchy title
# use the golf_corr dataframe from earlier that has all the Strokes Gained Variables

ax = golf_corr['Official Money Won'].sort_values(ascending=True).plot.barh()

ax.set_xlabel('Correlation Coefficients', fontsize=12)
ax.set_title('Strokes Gained Tee to Green = "Show Me the Money" ', fontsize=18)

plt.show()
```



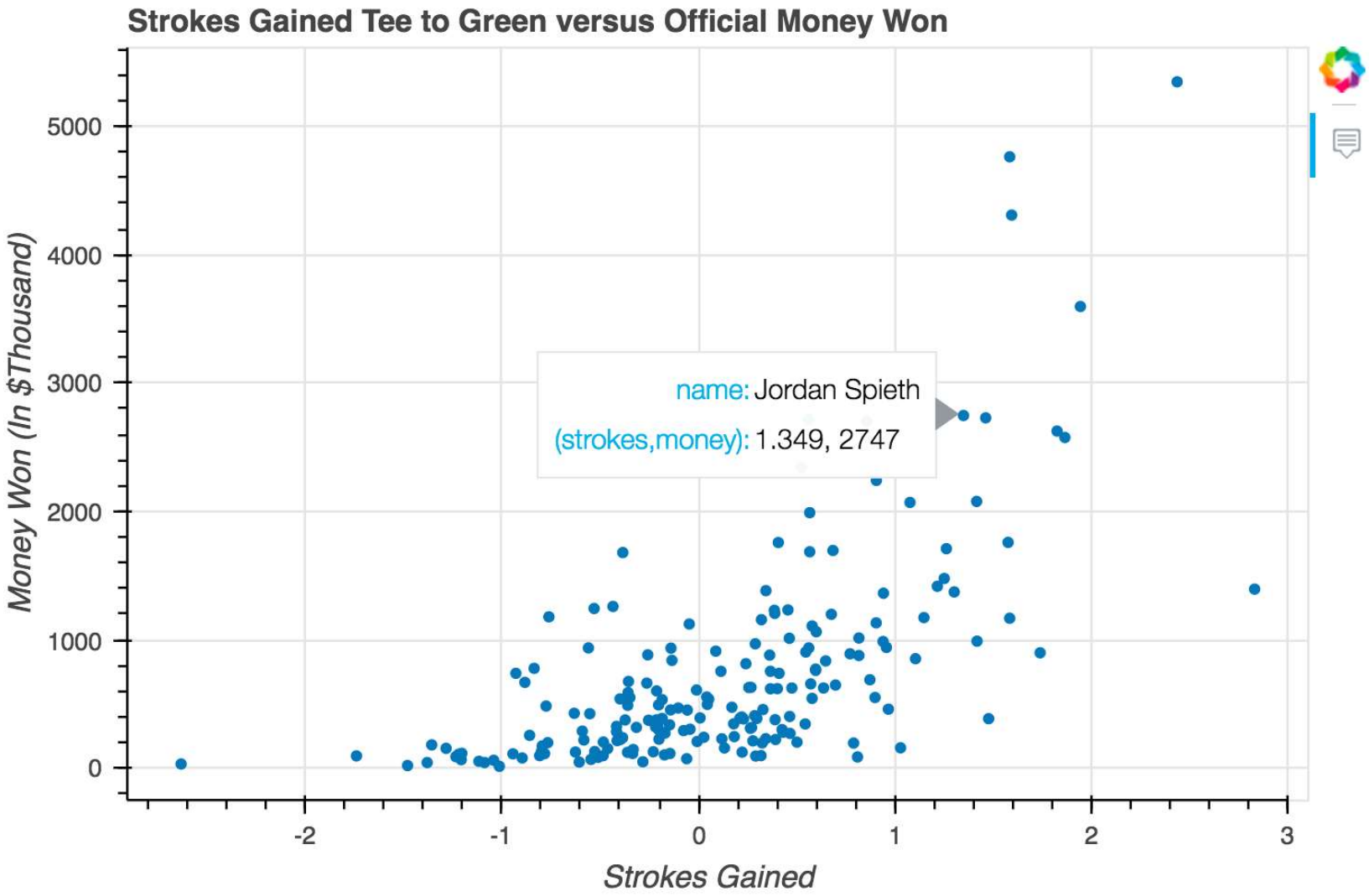
Bokeh HoverTool is Cool



5. Create Key Data Visualizations to Tell a Story

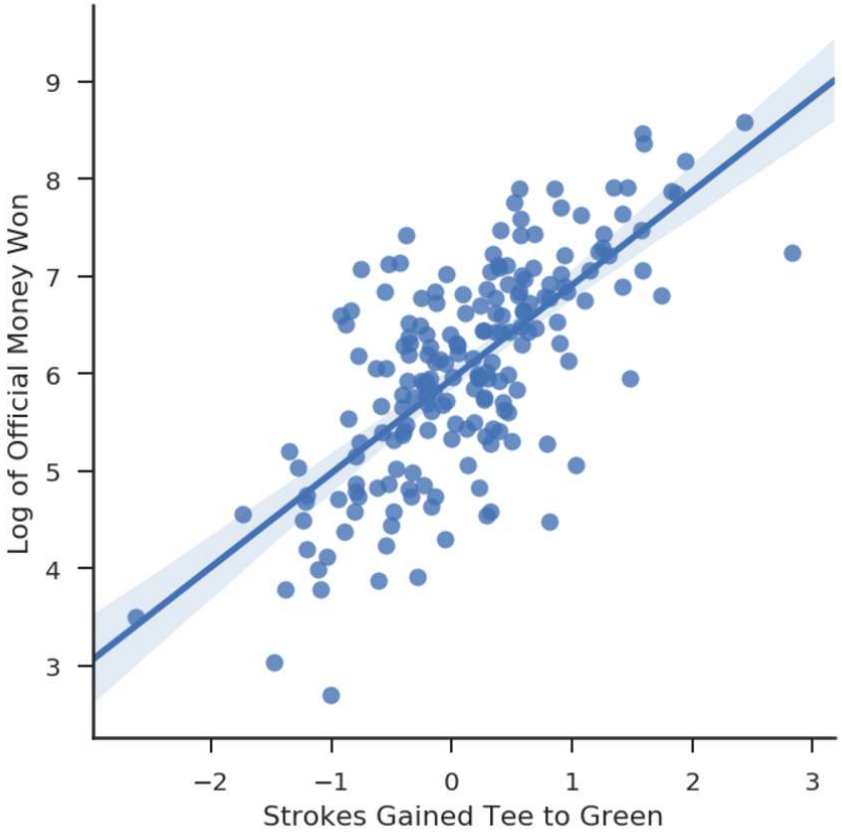
```
source = ColumnDataSource(
        data=dict(
            strokes = golf['Strokes Gained Tee to Green'],
            money = golf['Official Money Won'],
            name= golf['PLAYER NAME'],
        )
    )

hover = HoverTool(
        tooltips=[
            ("name", "@name"),
            ("(strokes,money)", "@strokes, @money"),
        ]
    )

p = figure(plot_width=600, plot_height=400, tools=[hover],
           title="Strokes Gained Tee to Green versus Official Money Won")

p.xaxis.axis_label = "Strokes Gained"
p.yaxis.axis_label = "Money Won (In $Thousand)"
p.xaxis.bounds = (-3,3)

p.circle('strokes', 'money', size=4, source=source)

output_notebook()

show(p)
```

Bokeh HoverTool is Cool



5. Create Key Data Visualizations to Tell a Story

```
sns.lmplot(x='Strokes Gained Tee to Green', y='Log of Official Money Won', data=golf)
plt.show()
```



Conclusions

Conclusions:

"Drive for Show, Putt for Dough" is a Myth

Strokes Gained Tee to Green = "Show Me the Money"

Pairplots Allow us to Visualize Relationships Between Variables

Bokeh makes really cool interactive visualizations

The visualizations lead us to look at the log of Official Money Won which has a 0.7 correlation with Strokes Gained Tee to Green. The last row of the pairplots gives a nice visual of the strong correlation.

Python and Golf are a match made in heaven

Thank you, Joe and Nenad, for a great course.

I learned a ton and had lots of fun!!!

REFERENCES

- http://www.pgatour.com/stats.html
- http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.rename.html
- http://pandas.pydata.org/pandas-docs/version/0.17.0/generated/pandas.DataFrame.drop.html
- http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.describe.html
- http://pandas.pydata.org/pandas-docs/stable/generated/pandas.DataFrame.corr.html
- http://pandas.pydata.org/pandas-docs/stable/visualization.html
- https://docs.python.org/3.1/library/warnings.html
- http://seaborn.pydata.org/generated/seaborn.heatmap.html
- http://seaborn.pydata.org/generated/seaborn.lmplot.html
- http://seaborn.pydata.org/generated/seaborn.pairplot.html
- http://bokeh.pydata.org/en/latest/docs/user_guide.html
- http://bokeh.pydata.org/en/latest/docs/user_guide/plotting.html
- http://bokeh.pydata.org/en/latest/docs/user_guide/tools.html#hover-tool
- https://docs.scipy.org/doc/numpy-1.10.1/reference/generated/numpy.log.html

Acknowledgments

- Jimmy DeLano. Son. Webscraping, creating csv file, providing feedback on analysis and initial video.
- Cori DeLano. Daughter. Providing feedback on video and video script.
- Tim Hogan and Joe Kambourakis. Teachers in Data Science Immersive (DSI) at General Assembly. Training in exploratory data analysis and data visualization.
- Erik Ellis, Geoff Counihan, Dimitri Linde, DK Kim, Wai Kin Shing, Phil Webb, Shawn Sherrell. Classmates in DSI. Providing ideas for overall analysis and specific code (cited in Jupyter notebook).