Swimming With The Sharks

December 15, 2022

Hunter Webb and Christopher White

Table of Contents

Introduction and Motivation:

Entrepreneurship is the lifeblood of the American economy. According to the CEO of the US Chamber of Commerce, for the last 3 decades companies under 1 year of development have generated 1.5 million new jobs annually in the US. Understanding this fact gives a clear reason why Shark Tank has had such tremendous success grabbing and holding the attention of the US people. Starting in 2009, Shark Tank is a show where entrepreneurs are given a large stage to display their company to the world and also seek guidance and financial support from some of America's best and brightest CEO's.

Our motivation for choosing this data set stems from our own personal interest in Shark Tank as well as a situation Chris White on the team has found himself in. Chris has created a SAAS (Software as a Service) demo of an indoor navigation tool that his girlfriend has used in her own entrepreneurship projects at Towson University. While Chris originally created this tool to help give her a competitive edge in the classroom to impress professors, he did not expect that she would enter competitions with this tool as her product. She won the event and has received $1500 to use to further build out her idea. With those events in mind, doing a deep dive into the world of Shark Tank's pitches may hopefully provide some helpful insight to Chris as he is exploring the world of entrepreneurship himself.

Our team came across a dataset on Kaggle that has comprised pitch information for the first 6 seasons of the American version of Shark Tank. Information regarding company valuation, the industry they are trying to enter, asking price, if a deal was met, what sharks where present that episode, and many more columns are present. There are crucial bits of information that would make the dataset much more informative such as which shark was selected for those who got an offer, and what the agreed upon price was for said deal, only the asking price is present. However, as a team we decided there was enough information to perform a substantial analysis.

In this tutorial we aim to give insight on how to effectively follow the data science pipeline given any context, which will allow the reader to apply this process into any dataset they wish to analyze and generate significant findings.

The first step to doing so is to look at our data and see what we have to work with.

Data Collection and Curation

Looking ahead we are going to want to use a few different libraries to help create useful visuals or run informative testing.

[ ]:
```
#The various imports we we be using
import warnings
warnings.filterwarnings("ignore")
from matplotlib import pyplot as plt
import pandas as pd
import statsmodels.api as sm
from statsmodels.stats.weightstats import ztest
```

Here we grab our data from the csv that was downloaded off Kaggle

[ ]:
```
#Read the data
data = pd.read_csv("shark_tank.csv")
#Print a few entries to see how the dataset is setup
data.head()
```

[ ]:
```
    deal                                        description  episode  \
0  False               Bluetooth device implant for your ear.        1
1   True  Retail and wholesale pie factory with two reta…       1
2   True  Ava the Elephant is a godsend for frazzled par…       1
3  False  Organizing, packing, and moving services deliv…       1
4  False  Interactive media centers for healthcare waiti…       1

             category               entrepreneurs      location  \
0            Novelties              Darrin Johnson  St. Paul, MN
1       Specialty Food                  Tod Wilson  Somerset, NJ
2  Baby and Child Care             Tiffany Krumins   Atlanta, GA
3    Consumer Services  Nick Friedman, Omar Soliman     Tampa, FL
4    Consumer Services              Kevin Flannery      Cary, NC

                                website  askedFor  exchangeForStake  valuation  \
0                                   NaN   1000000                15    6666667
1                  http://whybake.com/    460000                10    4600000
2         http://www.avatheelephant.com/     50000                15     333333
3  http://collegehunkshaulingjunk.com/    250000                25    1000000
4               http://www.wispots.com/   1200000                10   12000000

   season            shark1           shark2         shark3        shark4  \
0       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
1       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
2       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
```

```
3       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
4       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John

              shark5                           title episode-season  \
0  Kevin Harrington                       Ionic Ear            1-1
1  Kevin Harrington         Mr. Tod's Pie Factory            1-1
2  Kevin Harrington              Ava the Elephant            1-1
3  Kevin Harrington  College Foxes Packing Boxes            1-1
4  Kevin Harrington                        Wispots            1-1

   Multiple Entreprenuers
0                   False
1                   False
2                   False
3                   False
4                   False
```

The dataset contains plenty of information regarding the product and the companies pitching the respective products. However we are missing the ability to individually sort the pitches. We could go off episode and season number, but there are multiple pitches in any given episode, so here we add an extra column to individually index through the pitches

[ ]:
```
#Create a new column for the pitch number starting from the shows very first␣
 ↪pitch to the very last pitch in our dataset
data['pitchNumber'] =  (range(1, 1 + len(data)))
#printing entries to see the new column
data.head()
```

[ ]:
```
    deal                                       description  episode  \
0  False             Bluetooth device implant for your ear.        1
1   True  Retail and wholesale pie factory with two reta…        1
2   True  Ava the Elephant is a godsend for frazzled par…        1
3  False  Organizing, packing, and moving services deliv…        1
4  False  Interactive media centers for healthcare waiti…        1

               category                entrepreneurs      location  \
0             Novelties               Darrin Johnson  St. Paul, MN
1        Specialty Food                   Tod Wilson  Somerset, NJ
2  Baby and Child Care              Tiffany Krumins   Atlanta, GA
3    Consumer Services  Nick Friedman, Omar Soliman     Tampa, FL
4    Consumer Services               Kevin Flannery      Cary, NC

                             website  askedFor  exchangeForStake  valuation  \
0                                NaN   1000000                15    6666667
1                  http://whybake.com/    460000                10    4600000
2        http://www.avatheelephant.com/     50000                15     333333
3  http://collegehunkshaulingjunk.com/    250000                25    1000000
```

```
4             http://www.wispots.com/   1200000                 10   12000000

   season             shark1            shark2         shark3        shark4  \
0       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
1       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
2       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
3       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
4       1  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John

             shark5                        title episode-season  \
0  Kevin Harrington                    Ionic Ear            1-1
1  Kevin Harrington        Mr. Tod's Pie Factory            1-1
2  Kevin Harrington             Ava the Elephant            1-1
3  Kevin Harrington  College Foxes Packing Boxes            1-1
4  Kevin Harrington                      Wispots            1-1

   Multiple Entreprenuers  pitchNumber
0                   False            1
1                   False            2
2                   False            3
3                   False            4
4                   False            5
```

Now we are getting ready to start producing visuals. To start, we add a column to specify the colors used in our plots. This will help make the visual more informative

[ ]:
```
#Creating new column in dataset to represent if a deal was made or not for␣
↪graphing purposes
#Setting a default value of one to confirm no rows are missed
data['madeDeal'] = 1

#Assign each column based on the desired color for the visual
j = 0
for i in data['deal']:
  if i == True:
    #assigning the column green to indicate a deal was made
    data['madeDeal'][j] = 'green'
  else:
    #assigning the column red to indicate a deal was not made
    data['madeDeal'][j] = 'red'
  j = j + 1

#Visualize the new column in the dataset and confirm there are no 1's, which␣
↪indicate a missed row
data.head()
```

```
[ ]:     deal                                        description  episode  \
    0  False             Bluetooth device implant for your ear.        1
    1   True  Retail and wholesale pie factory with two reta…        1
    2   True  Ava the Elephant is a godsend for frazzled par…        1
    3  False  Organizing, packing, and moving services deliv…        1
    4  False  Interactive media centers for healthcare waiti…        1

                  category                entrepreneurs      location  \
    0            Novelties                Darrin Johnson  St. Paul, MN
    1       Specialty Food                    Tod Wilson  Somerset, NJ
    2  Baby and Child Care              Tiffany Krumins   Atlanta, GA
    3    Consumer Services  Nick Friedman, Omar Soliman     Tampa, FL
    4    Consumer Services               Kevin Flannery      Cary, NC

                                   website  askedFor  exchangeForStake  valuation  \
    0                                  NaN   1000000                15    6666667
    1                 http://whybake.com/    460000                10    4600000
    2      http://www.avatheelephant.com/     50000                15     333333
    3  http://collegehunkshaulingjunk.com/    250000                25    1000000
    4               http://www.wispots.com/   1200000                10   12000000

       …           shark1           shark2         shark3        shark4  \
    0  …  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
    1  …  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
    2  …  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
    3  …  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John
    4  …  Barbara Corcoran  Robert Herjavec  Kevin O'Leary  Daymond John

                shark5                        title episode-season  \
    0  Kevin Harrington                     Ionic Ear            1-1
    1  Kevin Harrington         Mr. Tod's Pie Factory            1-1
    2  Kevin Harrington              Ava the Elephant            1-1
    3  Kevin Harrington  College Foxes Packing Boxes            1-1
    4  Kevin Harrington                       Wispots            1-1

      Multiple Entreprenuers  pitchNumber  madeDeal
    0                  False            1       red
    1                  False            2     green
    2                  False            3     green
    3                  False            4       red
    4                  False            5       red

    [5 rows x 21 columns]
```

Here we are looking ahead towards testing a hypothesis. We would like to eventually see the effect of inflation on the increasing valuation of companies over the course of the show, and so we need to find the years that these 6 seasons took place, to adjust for inflation. The first step of this process was manually looking up the episodes and seeing when they took place.

```
[ ]: #Populating new column with default of 1 to see if any values are missed in the␣
     ↪conditional below
     data['year'] = 1

     #Assigning each row with the proper year based on the episode number
     #We used our pitch number column to determine which episodes belonged to what␣
     ↪year as some seasons occured across multiple years
     j = 0
     for curr_pitch in data['pitchNumber']:
       if curr_pitch <= 51:
         data['year'][j] = '2009'
       elif curr_pitch <= 64 and curr_pitch > 51:
         data['year'][j] = '2010'
       elif curr_pitch <= 100 and curr_pitch > 64:
         data['year'][j] = '2011'
       elif curr_pitch <= 203 and curr_pitch > 100:
         data['year'][j] = '2012'
       elif curr_pitch <= 311 and curr_pitch > 203:
         data['year'][j] = '2013'
       elif curr_pitch <= 427 and curr_pitch > 311:
         data['year'][j] = '2014'
       elif curr_pitch <= 495 and curr_pitch > 427:
         data['year'][j] = '2015'
       j = j + 1

     #View new addition to dataset to confirm it worked
     data.head()
```

```
[ ]:     deal                                        description  episode  \
     0  False              Bluetooth device implant for your ear.        1
     1   True  Retail and wholesale pie factory with two reta…        1
     2   True  Ava the Elephant is a godsend for frazzled par…        1
     3  False  Organizing, packing, and moving services deliv…        1
     4  False  Interactive media centers for healthcare waiti…        1

                  category                entrepreneurs      location  \
     0           Novelties               Darrin Johnson  St. Paul, MN
     1      Specialty Food                   Tod Wilson  Somerset, NJ
     2  Baby and Child Care              Tiffany Krumins   Atlanta, GA
     3    Consumer Services  Nick Friedman, Omar Soliman     Tampa, FL
     4    Consumer Services               Kevin Flannery      Cary, NC

                            website  askedFor  exchangeForStake  valuation  \
     0                          NaN   1000000                15    6666667
     1          http://whybake.com/    460000                10    4600000
```

```
2       http://www.avatheelephant.com/     50000                15     333333
3  http://collegehunkshaulingjunk.com/    250000                25    1000000
4              http://www.wispots.com/   1200000                10   12000000

   …           shark2          shark3        shark4            shark5  \
0  …  Robert Herjavec  Kevin O'Leary  Daymond John  Kevin Harrington
1  …  Robert Herjavec  Kevin O'Leary  Daymond John  Kevin Harrington
2  …  Robert Herjavec  Kevin O'Leary  Daymond John  Kevin Harrington
3  …  Robert Herjavec  Kevin O'Leary  Daymond John  Kevin Harrington
4  …  Robert Herjavec  Kevin O'Leary  Daymond John  Kevin Harrington

                         title episode-season Multiple Entreprenuers  \
0                    Ionic Ear            1-1                  False
1        Mr. Tod's Pie Factory            1-1                  False
2             Ava the Elephant            1-1                  False
3  College Foxes Packing Boxes            1-1                  False
4                      Wispots            1-1                  False

  pitchNumber  madeDeal  year
0           1       red  2009
1           2     green  2009
2           3     green  2009
3           4       red  2009
4           5       red  2009

[5 rows x 22 columns]
```

Now we have the proper years and are closing in on being able to test our first question about inflation. In order to do this, we looked up the USD inflation rate from 2009, and calculate each companies new adjusted valuation.

```
[ ]: #2015 10.5%
#2014 10.3%
#2013 8.6%
#2012 7%
#2011 4.9%
#2010 1.6%
#Above are the inflation rates from 2009-Target Year

#Creates new column with 1 as default value
data['adjustedValuation'] = 1

#Calculating new adjusted valuation
j = 0
for price in data[data['year'] == 2009]['valuation']:
  data['adjustedValuation'][j] = price
  j = j + 1
```

```python
for price in data[data['year'] == 2010]['valuation']:
  data['adjustedValuation'][j] = price * .984
  j = j + 1

for price in data[data['year'] == 2011]['valuation']:
  data['adjustedValuation'][j] = price * .951
  j = j + 1

for price in data[data['year'] == 2012]['valuation']:
  data['adjustedValuation'][j] = price * .93
  j = j + 1

for price in data[data['year'] == 2013]['valuation']:
  data['adjustedValuation'][j] = price * .914
  j = j + 1

for price in data[data['year'] == 2014]['valuation']:
  data['adjustedValuation'][j] = price * .897
  j = j + 1

for price in data[data['year'] == 2015]['valuation']:
  data['adjustedValuation'][j] = price * .895
  j = j + 1

#Visualize new column
data.head()
```

[ ]:

```
    deal                                        description  episode  \
0  False               Bluetooth device implant for your ear.        1
1   True  Retail and wholesale pie factory with two reta…        1
2   True  Ava the Elephant is a godsend for frazzled par…        1
3  False  Organizing, packing, and moving services deliv…        1
4  False  Interactive media centers for healthcare waiti…        1

             category                entrepreneurs      location  \
0           Novelties               Darrin Johnson  St. Paul, MN
1      Specialty Food                   Tod Wilson  Somerset, NJ
2  Baby and Child Care             Tiffany Krumins   Atlanta, GA
3    Consumer Services  Nick Friedman, Omar Soliman     Tampa, FL
4    Consumer Services               Kevin Flannery      Cary, NC

                            website  askedFor  exchangeForStake  valuation  \
0                               NaN   1000000                15    6666667
1                http://whybake.com/    460000                10    4600000
2       http://www.avatheelephant.com/     50000                15     333333
3  http://collegehunkshaulingjunk.com/    250000                25    1000000
```

```
4             http://www.wispots.com/   1200000                10   12000000

   …         shark3        shark4            shark5  \
0  …  Kevin O'Leary  Daymond John  Kevin Harrington
1  …  Kevin O'Leary  Daymond John  Kevin Harrington
2  …  Kevin O'Leary  Daymond John  Kevin Harrington
3  …  Kevin O'Leary  Daymond John  Kevin Harrington
4  …  Kevin O'Leary  Daymond John  Kevin Harrington

                       title episode-season Multiple Entreprenuers  \
0                  Ionic Ear            1-1                  False
1        Mr. Tod's Pie Factory          1-1                  False
2             Ava the Elephant          1-1                  False
3  College Foxes Packing Boxes          1-1                  False
4                    Wispots            1-1                  False

  pitchNumber madeDeal  year  adjustedValuation
0           1      red  2009            6666667
1           2    green  2009            4600000
2           3    green  2009             333333
3           4      red  2009            1000000
4           5      red  2009           12000000

[5 rows x 23 columns]
```

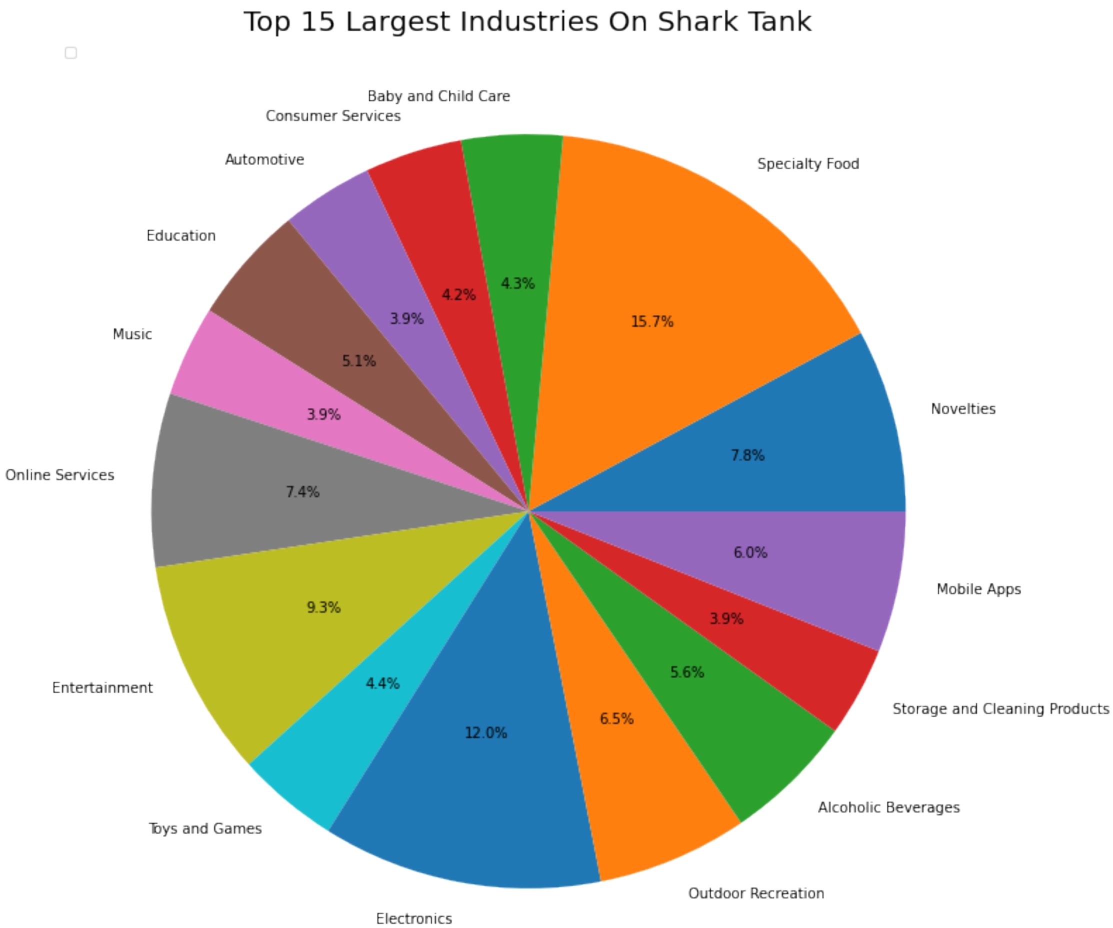
So now we have all of the information we wanted to perform our desired visuals and hypothesis testing on. It's time to start looking into our dataset to pick out helpful information to give to those trying to understand their odds and increase their likelihood of getting a deal with one of the sharks.

Exploratory Data Analysis

The sharks see a wide variety of products during the show. Obviously, everyone is there to make a deal so what is something that will instantly get the sharks attention? Those familiar with the sharks know that they each have their own thing, Lori being the QVC queen and Daymond being the fashion guy. The question becomes does this mean your company is more likely to be on the show if the sharks have interest in the category your company falls in. Let's look at the top 15 categories from our dataset.

[ ]:
```python
#Obtains the unique categories in our dataset
category = data.category.unique()

#Creating a dictionary to hold each category and its corresponding total␣
 ↪valuation
category_total = {key: 0 for key in category}
trim_graph = 0
min_num = 0
```

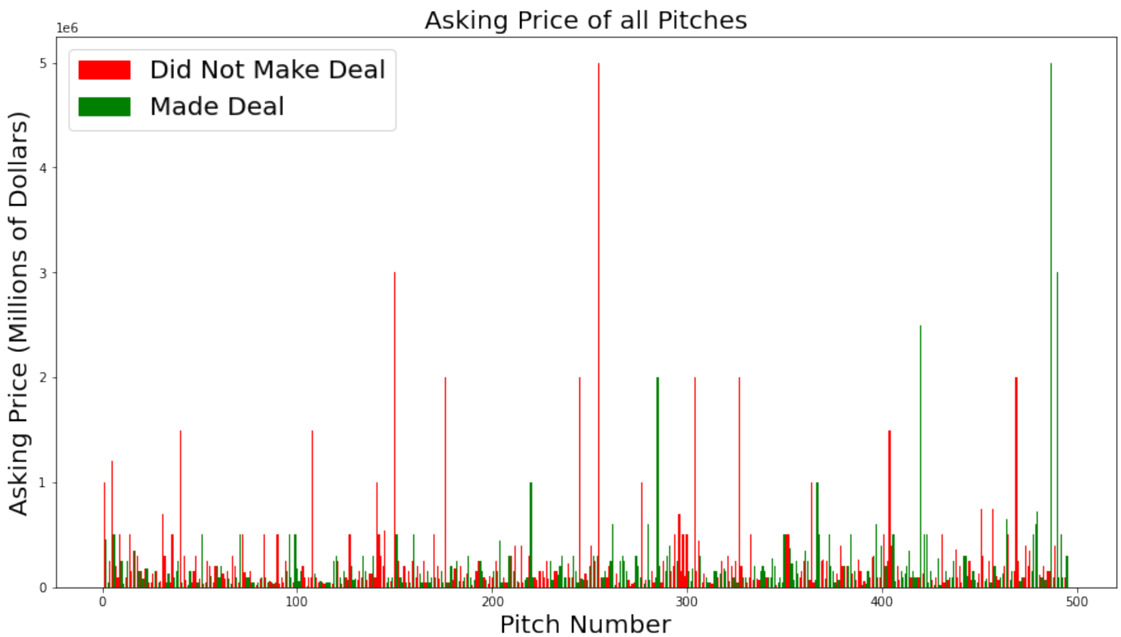
```
#Calculating the
for curr_category in category:
  category_total[curr_category] += (sum(data[data["category"] ==␣
 ↪curr_category]['valuation']))

#Eliminating the bottom 38 entries from category_total based on there valuation␣
 ↪so we are left with the 15 largest valuations based on category
while trim_graph <= 38:
  for k,v in category_total.items():
    min_num = min(category_total.values())
    if v == min_num:
      category_total.pop(k)
      trim_graph += 1
      break

#The following sets our plots size and adds the appropriate labels
plt.figure(figsize=(12,15))
plt.title("Top 15 Largest Industries On Shark Tank", fontsize=20)

#Adding a legend to the graph
plt.legend([category_total.keys()], loc="upper left")

#Creating the pie chart with the data and appropriate labels and chunk sizes
plt.pie(category_total.values(),labels=category_total.keys(),autopct='%1.1f%%')

plt.show()
```



This pie chart was made looking at each unique categories total valuation acorss all of the pitches and then only looking at the top 15 categories. It can be seen that Specialty Food is the top category on the show. On the other hand, being in a certain category could mean nothing in terms of getting on the show or making a deal. From watching the show one of the sharks main concerns are how much entreprener(s) are asking for and why they are asking for that much. A decent amount of the time the sharks are surprised by how much the entrepreneurs are asking for. Whether it is because they feel they are asking for too much based on the product or they are not asking for enough based on what they say they plan to do with the money. This leads us to our first question, are entrepreneurs more likely to make a deal with they go in with a low asking price? We would expect to see more deals made with lower asking prices as that becomes less of a risk for the sharks.

[ ]:
```
#The following sets our plots size and adds the appropriate labels
plt.figure(figsize=(15,8))
plt.xlabel("Pitch Number", fontsize = 20)
plt.title("Asking Price of all Pitches", fontsize = 20)
plt.ylabel("Asking Price (Millions of Dollars)", fontsize = 20)
```

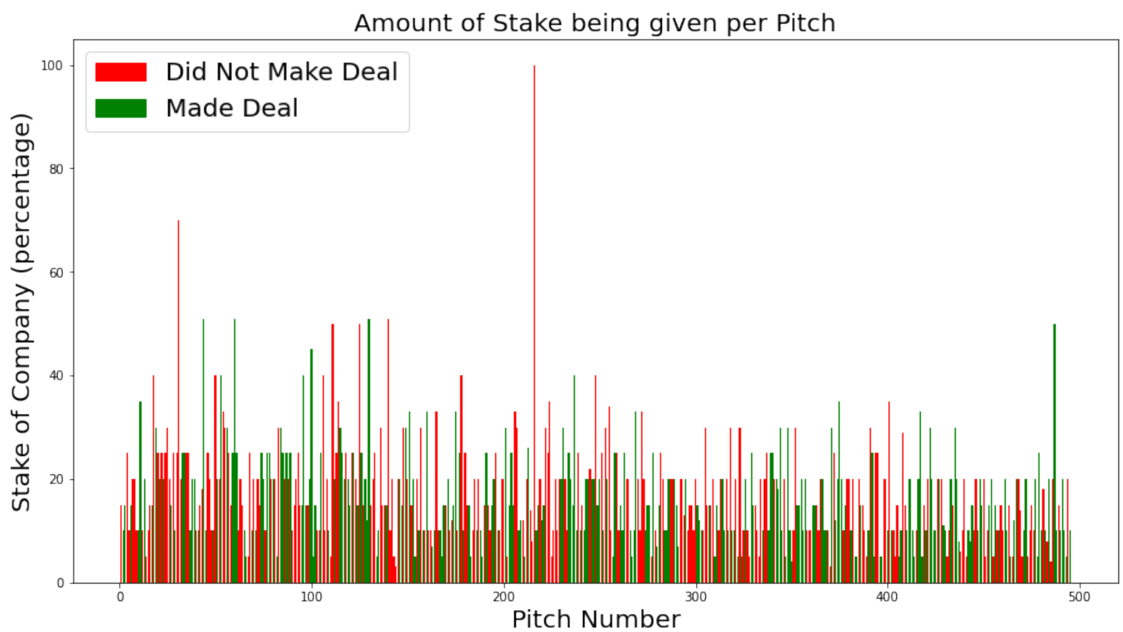
```
#Here you can see we used the 'madeDeal' column from earlier to indicate on the graph whether the pitch was successful in making a deal or not
plt.bar(data['pitchNumber'], data['askedFor'], color = data['madeDeal'])

#The following creates and adds the legend to the graph
colors = {'Did Not Make Deal':'red', 'Made Deal':'green'}
labels = list(colors.keys())
handles = [plt.Rectangle((0,0),1,1, color=colors[label]) for label in labels]
plt.legend(handles, labels, loc = "upper left",  fontsize = 20)

plt.show()
```



The above graph represents every pitch in our dataset. We graphed the asking price from each pitch in millions of dollars. We then used the colors, green and red, to represent whether or not that pitch was sucessful in making a deal. From the graph it is hard to see any specific trends or draw a conclusion as to whether or not how much entreprenuers ask for has an effect on if they are able to make a deal or not. It can be seen across all pitches that all ranges of asking prices sometimes do and do not result in deals. The other side of asking price is how much stake the company is willing to give to the sharks. Now we will be looking to see if there are any trends in how much stake a company is giving in terms of making a deal or not. We would expect to see more deals when the company is giving up a majority stake of the company as that becomes less of a risk for the shark to take.

[ ]:
```
#The following sets our plots size and adds the appropriate labels
plt.figure(figsize=(15,8))
```

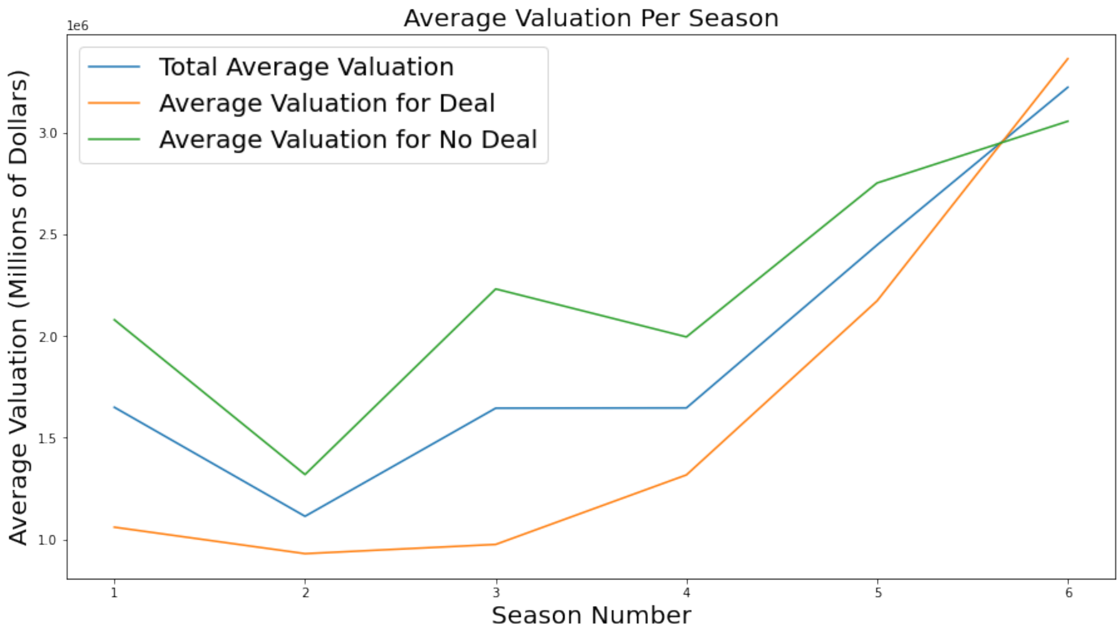
```
plt.xlabel("Pitch Number", fontsize = 20)
plt.title("Amount of Stake being given per Pitch", fontsize = 20)
plt.ylabel("Stake of Company (percentage)", fontsize = 20)

#Here you can see we used the 'madeDeal' column from earlier to indicate on the␣
↪graph whether the pitch was successful in making a deal or not
plt.bar(data['pitchNumber'], data['exchangeForStake'], color = data['madeDeal'])

#The following creates and adds the legend to the graph
colors = {'Did Not Make Deal':'red', 'Made Deal':'green'}
labels = list(colors.keys())
handles = [plt.Rectangle((0,0),1,1, color=colors[label]) for label in labels]
plt.legend(handles, labels, loc = "upper left",  fontsize = 20)

plt.show()
```



The above graph represents every pitch in our dataset. We graphed the percentage of stake the entreprenuers are offering. We then used the colors, green and red, to represent whether or not that pitch was sucessful in making a deal. From the graph it is hard to see any specific trends or draw a conclusion as to whether or not how much of the company entreprenuers give up has an effect on if they are able to make a deal or not. It can be seen across all pitches that all ranges of stake being given sometimes do and do not result in deals. The two pitches that offered the majority of the company were actually unsucessful in making a deal, the opposite of what we expected. Upon further analysis this can be determined to be logical as the stake of the company is not the only factor, these companies could have had extremely high asking prices. With this in mind we will now look at the combination of the two previous graphs, graphing every pitches valuation. As looking at all pitches can be overwhelming we will now break it down per season. We do this to not only make the graphs clearer but also because the earlier pitches and seasons can be considered a learning process for the sharks as well in terms of how and what they will invest in. Breaking it down per season will allow us to also see if as the years go on if inflation has an affect on companies valuations. We will still be breaking it down in terms of whether a deal was made or not.

```
[ ]: #Creating various arrays to store appropriate data
seasons_avg = []
acceptance_avg = []
decline_avg = []

#Getting each unique season number
seasons = data.season.unique()

#Calculates each seasons average valuation for every pitch and stores it in the␣
 ↪seasons_avg array
for curr_season in seasons:
  seasons_avg.append(sum(data[data["season"] == curr_season]['valuation'])/
 ↪len(data[data["season"] == curr_season]))

#calculates each seasons average valuation for every pitch in which a deal was␣
 ↪made and stores it in the acceptance_avg array
for curr_season in seasons:
      #Using the 'deal' column to determine if a deal was made and or not
      tmp = data[data["deal"] == True]
      acceptance_avg.append(sum(tmp[data["season"] ==␣
 ↪curr_season]['valuation'])/len(tmp[data["season"] == curr_season]))

#calculates each seasons average valuation for every pitch in which a deal was␣
 ↪not made and stores it in the acceptance_avg array
for curr_season in seasons:
      #Using the 'deal' column to determine if a deal was made and or not
      tmp = data[data["deal"] == False]
      decline_avg.append(sum(tmp[data["season"] == curr_season]['valuation'])/
 ↪len(tmp[data["season"] == curr_season]))

#The following sets our plots size and adds the appropriate labels
plt.figure(figsize=(15,8))
plt.xlabel("Season Number", fontsize = 20)
plt.title("Average Valuation Per Season", fontsize = 20)
plt.ylabel("Average Valuation (Millions of Dollars)", fontsize = 20)

#Plotting the various arrays made above
plt.plot(seasons, seasons_avg, label="Total Average Valuation")
plt.plot(seasons, acceptance_avg, label="Average Valuation for Deal")
plt.plot(seasons, decline_avg, label="Average Valuation for No Deal")

#Adding a legend to the graph
```

```
plt.legend(fontsize=20)

plt.show()
```



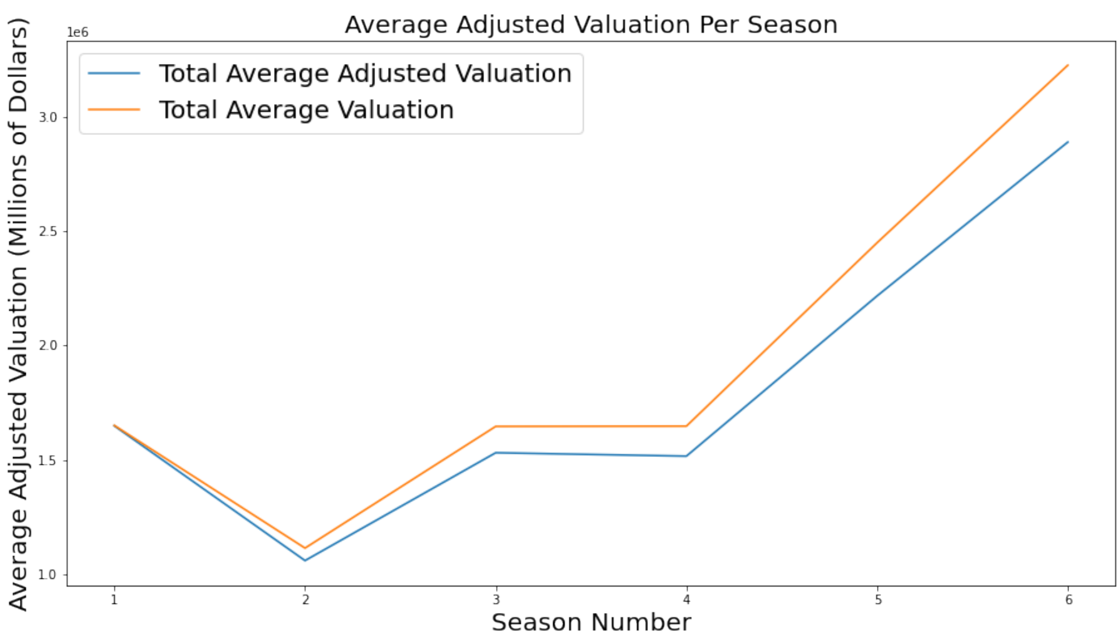
On this plot we have the total average valuation per season, representing all pitches, in blue. The total average valuation per season for picthes that did not make a deal in green. Lastly, in orange we have the total average valuation per season for pitches that did make a deal. From the plot it can be seen that until seasons 5 companies with lower valutaions than the average were more likely to make a deal. It can also be seen that after season two the average valuation per season only increased along with the average valuation for pitches that made a deal. The question is what caused these sizeable increases in valuation after season two? This leads us to our hypothesis testing.

Hypothesis Testing

The first question we want to tackle regards the valuations across the season and if there really is a positive trend, which would indicate that the show is bringing on bigger and bigger companies overtime, which wouldn't leave any space for smaller companies to get their chance.

[ ]:
```
#Creating an array to store each seasons average adjusted valuation based on
    ↪inflation rates
seasons_avg_adj = []

#Getting each unique season number
seasons = data.season.unique()
```

```
#Calculating each seasons average adjusted valuation and storing it in the␣
↪seasons_avg_adj array
for curr_season in seasons:
  seasons_avg_adj.append(sum(data[data["season"] ==␣
↪curr_season]['adjustedValuation'])/len(data[data["season"] == curr_season]))

#The following sets our plots size and adds the appropriate labels
plt.figure(figsize=(15,8))
plt.xlabel("Season Number", fontsize = 20)
plt.title("Average Adjusted Valuation Per Season", fontsize = 20)
plt.ylabel("Average Adjusted Valuation (Millions of Dollars)", fontsize = 20)

#Plotting the newly created average adjusted valuation and the original average␣
↪valuation per season
plt.plot(seasons, seasons_avg_adj, label="Total Average Adjusted Valuation")
plt.plot(seasons, seasons_avg, label="Total Average Valuation")

#Creating the plots legend
plt.legend(fontsize=20)

plt.show()
```



Here we are just viewing the differences between the average valuations to see if the adjusted valuations reveals the same trends we observered earlier, and it clearly does still reveal the same positive trend, but we cannot make that claim until our testing results backs up our claim.

We will run a least squares regression test to evaluate the trend between the adjusted average valuation of companies across the season and see if there is a strong positive relationship between the two variables. We will conclude the strength of the relationship based on the R squared value the test returns

[ ]:
```python
#Running a least squares regression test
ind = seasons
dep = seasons_avg_adj
ind_ = sm.add_constant(ind)
lm = sm.OLS(dep, ind_).fit()

lm.summary()
```

```
/usr/local/lib/python3.8/dist-packages/statsmodels/stats/stattools.py:74:
ValueWarning: omni_normtest is not valid with less than 8 observations; 6
samples were given.
  warn("omni_normtest is not valid with less than 8 observations; %i "
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.640
Model:                            OLS   Adj. R-squared:                  0.550
Method:                 Least Squares   F-statistic:                     7.105
Date:                Thu, 15 Dec 2022   Prob (F-statistic):             0.0561
Time:                        21:49:57   Log-Likelihood:                -85.167
No. Observations:                   6   AIC:                             174.3
Df Residuals:                       4   BIC:                             173.9
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        8.44e+05   4.03e+05      2.094      0.104   -2.75e+05    1.96e+06
x1          2.758e+05   1.03e+05      2.666      0.056   -1.15e+04    5.63e+05
==============================================================================
Omnibus:                          nan   Durbin-Watson:                   1.610
Prob(Omnibus):                    nan   Jarque-Bera (JB):                0.602
Skew:                           0.319   Prob(JB):                        0.740
Kurtosis:                       1.586   Cond. No.                         9.36
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

Two things happen when we run this test. We see an R-Squared value of 0.64, which indicates a moderate strength of relation between these two values. However, the bigger problem is that we do not have enough samples to base this test on, only using the 6 averages throws a warning as the testing set is not big enough. We will have to alter our approach to testing this relationship, and address this weakness of the dataset in our conclusion below.

Luckily we have another similar set of columns that will allow us to run this test. The pitch numbers will serve as a very similar independant variable (x-axis) to the dependent variable of adjusted valuations (y-axis). This way we will have a significantly larger testing set of 495 observations. The reason these values are still similar is because we are still capturing the change in valuations over the course of the show, just this time not in an aggregate manner.

[ ]:
```
#Running a least squares regression test
ind = data['pitchNumber']
dep = data['adjustedValuation']
ind_ = sm.add_constant(ind)
lm = sm.OLS(dep, ind_).fit()

lm.summary()
```

[ ]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:      adjustedValuation   R-squared:                       0.024
Model:                            OLS   Adj. R-squared:                  0.022
Method:                 Least Squares   F-statistic:                     12.08
Date:                Thu, 15 Dec 2022   Prob (F-statistic):           0.000554
Time:                        21:49:57   Log-Likelihood:                -8143.9
No. Observations:                 495   AIC:                         1.629e+04
Df Residuals:                     493   BIC:                         1.630e+04
Df Model:                           1
Covariance Type:            nonrobust
===============================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------
const        1.069e+06   3.05e+05      3.505      0.000     4.7e+05    1.67e+06
pitchNumber  3703.1257   1065.329      3.476      0.001    1609.981    5796.270
==============================================================================
Omnibus:                      425.515   Durbin-Watson:                   2.080
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7805.281
Skew:                           3.825   Prob(JB):                         0.00
Kurtosis:                      20.886   Cond. No.                         573.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

It may be difficult to interpret why the R squared value has dropped so much more from .6 to .024 when this test was supposed to be similar to the previous one. The variance of this new independent variable is much higher so R squared values are going to be much lower. However in this case we still cannot claim there is a strong positive relationship between valuations across time.

Now for our next test we are going to evaluate if having a website gives any indication towards the likelihood of securing a deal. Intuitivley it would make sense that those who have a website are more prepared and further invested in their companies, but we cannot be sure until the data reveals this fact. We will perform a two sided z-test and only confirm the claim that having a website indicates a higher likelihood of getting a deal if the p value is < 0.05

[ ]:
```
#Creating an array to represent pitches with no website
noWebsite = []

#Going through the datas 'website' column to determine if there is a website␣
 ↪present
for result in data[data['website'].isnull() == True]['deal']:
  #appending 1 if a deal was made and 0 if no deal was made
  if result == True:
    noWebsite.append(1)
  else:
    noWebsite.append(0)

#Creating an array to represent pitches with a website
hasWebsite = []
for result in data[data['website'].isnull() == False]['deal']:
  #appending 1 if a deal was made and 0 if no deal was made
  if result == True:
    hasWebsite.append(1)
  else:
    hasWebsite.append(0)
```

Now we have assigned getting a deal as a 1 and not getting a deal as a 0 to create a poisson like distribution of our data, which will allow us to perform our desired test

[ ]:
```
#Two sided ztest based upon the company having or not having a website
ztest(noWebsite, hasWebsite, value=0, alternative='two-sided')
```

[ ]:
```
(-2.464689767495524, 0.013713192618461075)
```

Great! The test returns a P-value of .0137, which is well below our threshold of .05. This allows us to confirm the fact that having a website offers a good insight to the likelihood of securing a deal from one of the sharks.

Communications of Insights Attained

After conducting an analysis of the data, it has become clear that there are several factors that can improve the chances of securing a deal on Shark Tank. These include having a website ready, understanding that there is still space for smaller startups, and tailoring the product towards one of the main industries featured on the show.

First, the finding that there is no significant relationship between the average valuation of companies on Shark Tank and the year suggests that even smaller companies have a chance of appearing on the program and potentially securing a deal. This means that entrepreneurs should not be discouraged from appearing on the show, even if their company is not as developed as others.

Secondly, the significant relationship between having a website and securing a deal on Shark Tank indicates that it would be beneficial to have a website ready before appearing on the program. This could potentially increase the chances of success and help secure a deal with one of the Sharks.

Thirdly, tailoring the product towards one of the main industries featured on the show, such as health and wellness, food and beverage, or technology, can also improve the chances of securing a deal. By focusing on these industries and making the product fit within one of these categories, entrepreneurs can make their product more appealing to the Sharks and increase the chances of securing a deal.

Overall, these findings highlight the importance of being prepared with a website, understanding that even smaller companies still have a chance, and tailoring the product towards one of the main industries on the show when making a decision about appearing on the program. By taking these factors into account, entrepreneurs can increase their chances of success and secure a deal on Shark Tank.

Although our dataset had many columns to start and we were able to make various useful columns the dataset did have some weaknesses. The biggest one being we did not know which shark made the deal if a deal was made. A lot of meaningful conclusions could have been discovered with this information such as seeing which shark makes the most deals. Based off of this information we could then determine the top five sharks to be pitching to inorder to have a higher chance of making a deal. On that same note, we also did not have any data indicating if deals were made by one shark or multiple sharks. Knowing this we could have determined the best shark lineup for not just making a deal but making a deal with multiple sharks. Another weakness of our dataset was not knowing anything about potential negotiations that occured. With this information we could have furthered analyzed asking price and stake by seeing if starting out by asking low resulted in the sharks investing more as they start to potentially compete with one another. With shark tank currently being on its 14th season it would have been great to have an up to date dataset or had at least up to season 13. This would have allowed us to enhance our analysis on inflation rates as well as speak to the sharks own learning process in the earlier seasons. It also would have allowed us to do our original hypothesis testing since we would have then had more than 8 observations for the least squares regression test.

Overall, this tutorial while offering insights into the world of Shark Tank should have also given a good intro to the pipeline of data science. Collecting data, currating it, and running analysis through visuals and hypotheis testing is a wildly powerful tool that anyone can use for their own benefit. For example, Chris will likely use this journal to understand that he should create a website for his company and mutate his product to fit into a use case for one of the main industries covered if his girlfriend was to hypothetically go on Shark Tank and they wished to maximize their odds of securing a deal.

[ ]:

```

```