Bret Young

DSC 640

Assignment 1.2

10 September 2020

- bar plot
- stacked bar plot
- pie chart
- donut chart

In [3]:
```
# Import required packages
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Bar Chart

In [6]:
```
# Load dataset
url = '~/Desktop/DSC 640/ex1-2/hotdog-contest-winners.xlsm'
data_1 = pd.read_excel(url)
```

In [7]:
```
data_1.head
```

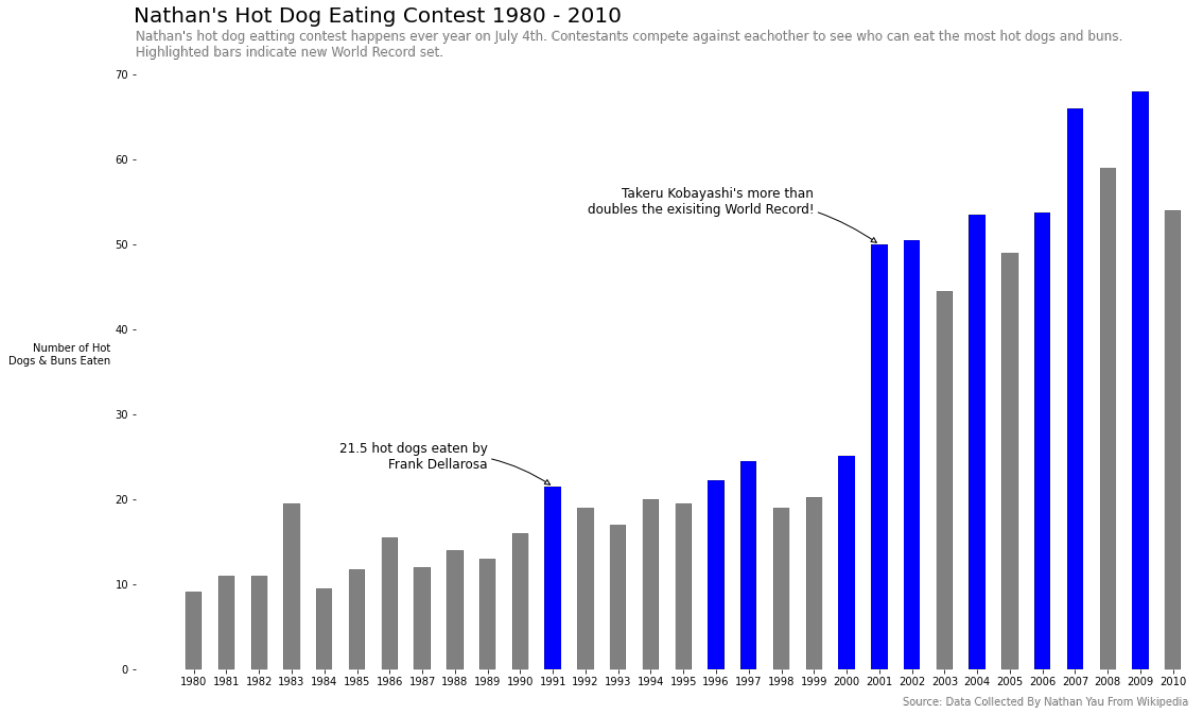
Out[7]:
```
<bound method NDFrame.head of     Year                         Winner
Dogs eaten        Country  New record
0   1980  Paul Siederman & Joe Baldini        9.10  United States
0
1   1981               Thomas DeBerry       11.00  United States
0
2   1982                Steven Abrams       11.00  United States
0
3   1983                  Luis Llamas       19.50         Mexico
0
4   1984                Birgit Felden        9.50        Germany
0
5   1985              Oscar Rodriguez       11.75  United States
0
```

```
6   1986                  Mark Heller        15.50  United States
0
7   1987                  Don Wolfman        12.00  United States
0
8   1988                    Jay Green        14.00  United States
0
9   1989                    Jay Green        13.00  United States
0
10  1990                  Mike DeVito        16.00  United States
0
11  1991               Frank Dellarosa        21.50  United States
1
12  1992               Frank Dellarosa        19.00  United States
0
13  1993                  Mike DeVito        17.00  United States
0
14  1994                  Mike DeVito        20.00  United States
0
15  1995                Edward Krachie        19.50  United States
0
16  1996                Edward Krachie        22.25  United States
1
17  1997             Hirofumi Nakajima        24.50          Japan
1
18  1998             Hirofumi Nakajima        19.00          Japan
0
19  1999                 Steve Keiner        20.25  United States
0
20  2000   Kazutoyo "The Rabbit" Arai        25.13          Japan
1
21  2001              Takeru Kobayashi        50.00          Japan
1
22  2002              Takeru Kobayashi        50.50          Japan
1
23  2003              Takeru Kobayashi        44.50          Japan
0
24  2004              Takeru Kobayashi        53.50          Japan
1
25  2005              Takeru Kobayashi        49.00          Japan
0
26  2006   Takeru "Tsunami" Kobayashi        53.75          Japan
1
27  2007                 Joey Chestnut        66.00  United States
1
28  2008                 Joey Chestnut        59.00  United States
0
29  2009                 Joey Chestnut        68.00  United States
1
30  2010                 Joey Chestnut        54.00  United States
0>
```

In [26]:
```
# set colors for bars
col = []
for val in data_1['New record']:
    if val == 1:
        col.append('blue')
    else:
        col.append('gray')
```

In [110]:
```
# Create axes and figure
fig = plt.figure()
ax1 = fig.add_subplot(111)

# Set figure size
fig.set_size_inches(18.5, 10.5)

# Add plot to figure
ax1.bar(np.arange(len(data_1['Year'])), data_1['Dogs eaten'], width =
0.5, color = col)

# Change x-axis values
plt.xticks(np.arange(len(data_1['Year'])), data_1['Year'])

# Set titles, caption and axis labels
fig.suptitle("Nathan's Hot Dog Eating Contest 1980 - 2010", x = 0.296,
y = 0.95, fontsize=20)
fig.text(.87, .08, 'Source: Data Collected By Nathan Yau From Wikipedi
a', ha = 'right', color = 'gray')
ax1.set_title("Nathan's hot dog eatting contest happens ever year on J
uly 4th. Contestants compete against eachother to see who can eat the
most hot dogs and buns.\nHighlighted bars indicate new World Record se
t.", loc='left', color = 'gray')
ax1.set_ylabel("Number of Hot\nDogs & Buns Eaten", rotation = 0, ha='r
ight')

# Remove frame
plt.box(on = None)

# Add annotation
ax1.annotate("21.5 hot dogs eaten by\nFrank Dellarosa",
            xy=(11, 21.5), xycoords='data',
            xytext=(9, 25), textcoords='data',
            size=12, va="center", ha="right",
            arrowprops=dict(arrowstyle="-|>",
                            connectionstyle="arc3,rad=-0.2",
                            fc="w"),
            )
ax1.annotate("Takeru Kobayashi's more than\ndoubles the exisiting Worl
d Record!",
```

```
            xy=(21, 50), xycoords='data',
            xytext=(19, 55), textcoords='data',
            size=12, va="center", ha="right",
            arrowprops=dict(arrowstyle="-|>",
                            connectionstyle="arc3,rad=-0.2",
                            fc="w"),
            )

# Show plot
plt.show

# save file
fig.savefig("python_bar.png")
```



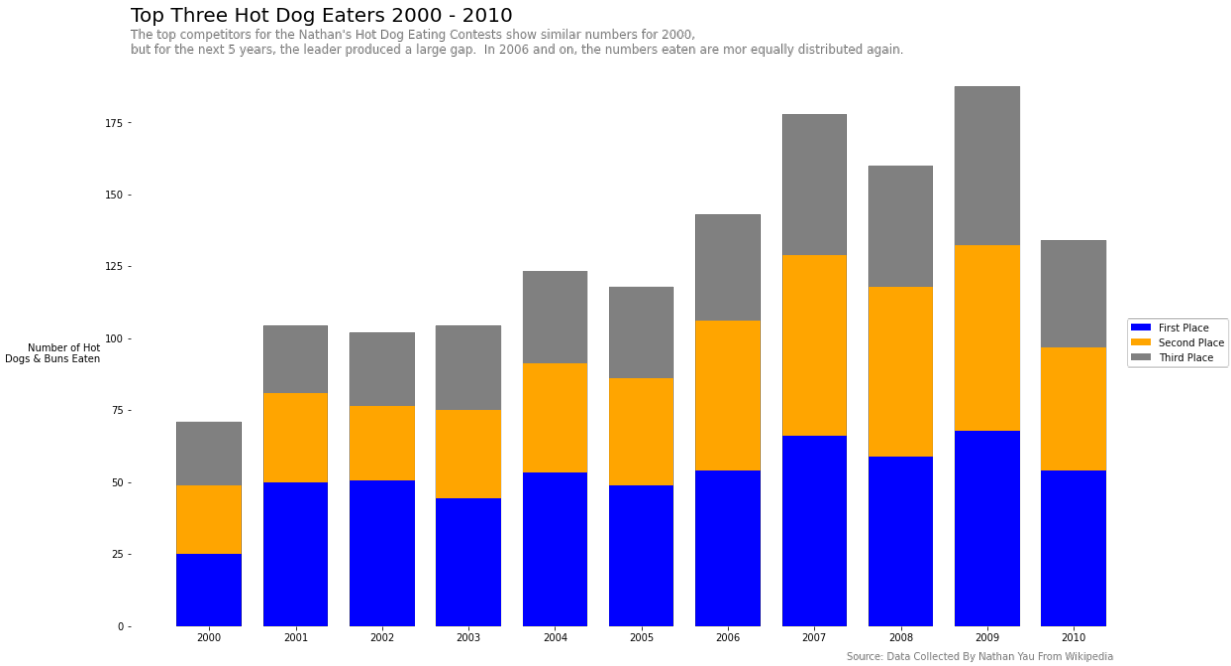
Stacked Bar Chart

In [81]:
```
# Load dataset
url_2 = url = '~/Desktop/DSC 640/ex1-2/hotdog-places.xlsm'
data_2 = pd.read_excel(url_2)
```

In [83]:
```
data_2.head
```

Out[83]:
```
<bound method NDFrame.head of    2000  2001  2002  2003  2004  2005  2006  2007  2008  2009  2010
0    25  50.0  50.5  44.5  53.5    49    54    66    59  68.0    54
1    24  31.0  26.0  30.5  38.0    37    52    63    59  64.5    43
2    22  23.5  25.5  29.5  32.0    32    37    49    42  55.0    37>
```

In [127]:
```
# Create axes and figure
fig = plt.figure()
ax1 = fig.add_subplot(111)

# Set figure size
fig.set_size_inches(18.5, 10.5)

# Add plot to figure
ax1.bar(np.arange(len(data_2.columns)), data_2.iloc[0], width = 0.75,
color = 'blue')
ax1.bar(np.arange(len(data_2.columns)), data_2.iloc[1], bottom = data_
2.iloc[0], width = 0.75, color = 'orange')
ax1.bar(np.arange(len(data_2.columns)), data_2.iloc[2], bottom = (data
_2.iloc[0] + data_2.iloc[1]), width = 0.75, color = 'gray')

# Change x-axis values
plt.xticks(np.arange(len(data_2.columns)), data_2.columns)

# Set titles, caption and axis labels
fig.suptitle("Top Three Hot Dog Eaters 2000 - 2010", x = 0.27, y = 0.9
5, fontsize=20)
fig.text(.87, .08, 'Source: Data Collected By Nathan Yau From Wikipedi
a', ha = 'right', color = 'gray')
ax1.set_title("The top competitors for the Nathan's Hot Dog Eating Con
tests show similar numbers for 2000,\nbut for the next 5 years, the le
ader produced a large gap.  In 2006 and on, the numbers eaten are mor
equally distributed again.", loc='left', color = 'gray')
ax1.set_ylabel("Number of Hot\nDogs & Buns Eaten", rotation = 0, ha='r
ight', va = "top")

# Remove frame
plt.box(on = None)

# show legend
plt.legend(['First Place','Second Place','Third Place'], bbox_to_ancho
r = (0.97, 0.5), loc = "center left")

# Show plot
plt.show

# save file
fig.savefig("python_stacked_bar.png")
```



Pie Chart

In [128]:
```
# Load dataset
url_3 = url = '~/Desktop/DSC 640/ex1-2/obama-approval-ratings.xls'
data_3 = pd.read_excel(url_3)
```

In [129]:
```
data_3.head
```

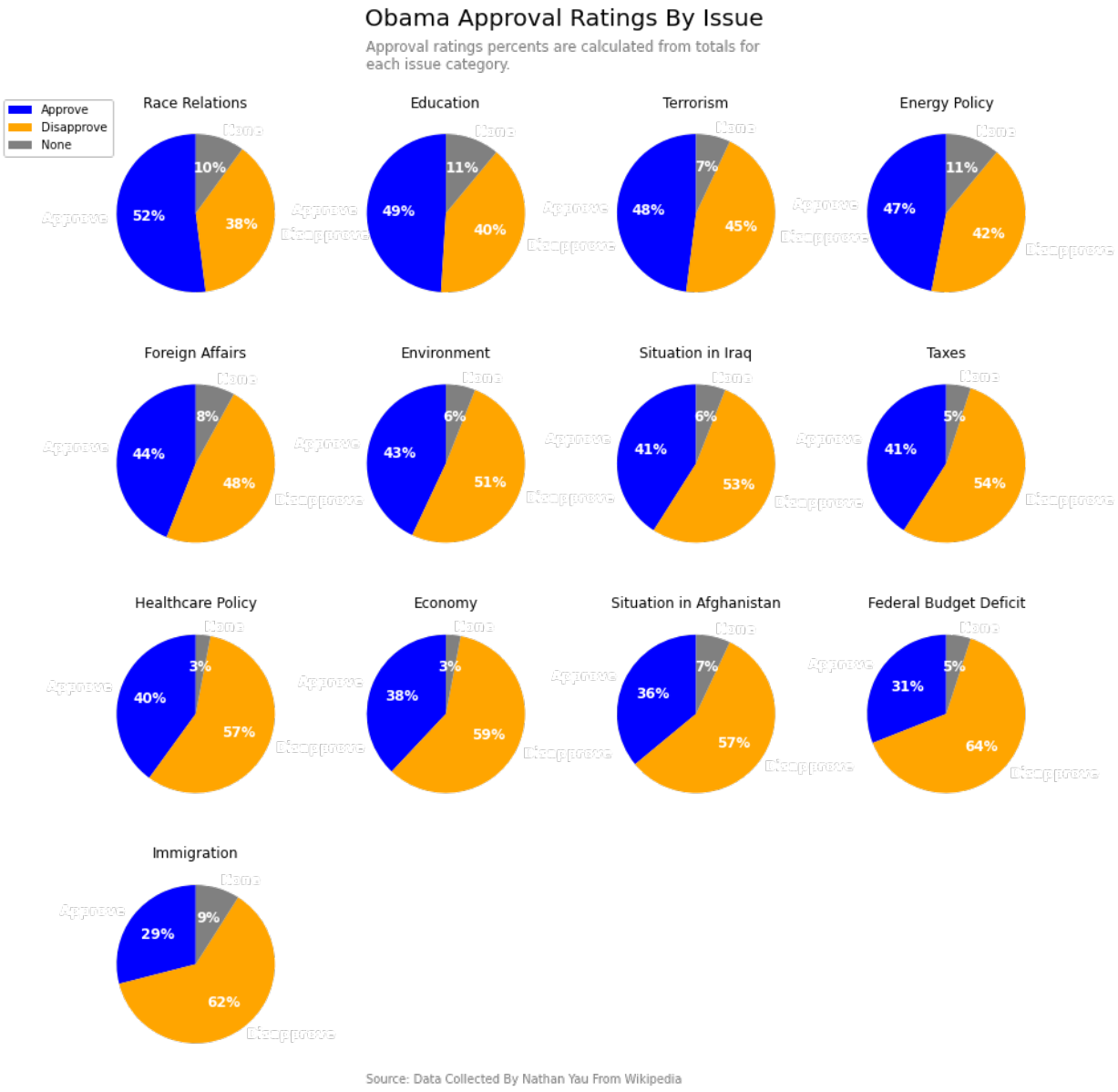
Out[129]:
```
<bound method NDFrame.head of                                 Issue  Approve
Disapprove  None
0                Race Relations       52          38     10
1                     Education       49          40     11
2                     Terrorism       48          45      7
3                  Energy Policy       47          42     11
4               Foreign Affairs       44          48      8
5                   Environment       43          51      6
6            Situation in Iraq       41          53      6
7                         Taxes       41          54      5
8            Healthcare Policy       40          57      3
9                       Economy       38          59      3
10  Situation in Afghanistan       36          57      7
11    Federal Budget Deficit       31          64      5
12                Immigration       29          62      9>
```

In [314]:
```
# Create axes and figure
fig, ax1 = plt.subplots(4,4)
```

```
# Set figure size & spacing
fig.set_size_inches(12, 12)
fig.tight_layout()

# Set wedge colors
colour = ['blue', 'orange', 'gray']

# Add plot to figure
ax1[0, 0].pie(data_3.iloc[0][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[0, 0].set_title(data_3['Issue'][0], loc = 'center')

# Add plot to figure
ax1[0, 1].pie(data_3.iloc[1][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[0, 1].set_title(data_3['Issue'][1], loc = 'center')

# Add plot to figure
ax1[0, 2].pie(data_3.iloc[2][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[0, 2].set_title(data_3['Issue'][2], loc = 'center')

# Add plot to figure
ax1[0, 3].pie(data_3.iloc[3][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[0, 3].set_title(data_3['Issue'][3], loc = 'center')

# Add plot to figure
ax1[1, 0].pie(data_3.iloc[4][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[1, 0].set_title(data_3['Issue'][4], loc = 'center')

# Add plot to figure
ax1[1, 1].pie(data_3.iloc[5][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
```

```
", size = 12, weight = 'bold'))

# Add axes title
ax1[1, 1].set_title(data_3['Issue'][5], loc = 'center')

# Add plot to figure
ax1[1, 2].pie(data_3.iloc[6][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[1, 2].set_title(data_3['Issue'][6], loc = 'center')

# Add plot to figure
ax1[1, 3].pie(data_3.iloc[7][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[1, 3].set_title(data_3['Issue'][7], loc = 'center')

# Add plot to figure
ax1[2, 0].pie(data_3.iloc[8][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[2, 0].set_title(data_3['Issue'][8], loc = 'center')

# Add plot to figure
ax1[2, 1].pie(data_3.iloc[9][1:], labels = data_3.columns[1:], startan
gle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="w
", size = 12, weight = 'bold'))

# Add axes title
ax1[2, 1].set_title(data_3['Issue'][9], loc = 'center')

# Add plot to figure
ax1[2, 2].pie(data_3.iloc[10][1:], labels = data_3.columns[1:], starta
ngle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="
w", size = 12, weight = 'bold'))

# Add axes title
ax1[2, 2].set_title(data_3['Issue'][10], loc = 'center')

# Add plot to figure
ax1[2, 3].pie(data_3.iloc[11][1:], labels = data_3.columns[1:], starta
ngle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="
w", size = 12, weight = 'bold'))
```

```
# Add axes title
ax1[2, 3].set_title(data_3['Issue'][11], loc = 'center')

# Add plot to figure
ax1[3, 0].pie(data_3.iloc[12][1:], labels = data_3.columns[1:], starta
ngle = 90, colors = colour, autopct = '%1.f%%', textprops=dict(color="
w", size = 12, weight = 'bold'))

# Add axes title
ax1[3, 0].set_title(data_3['Issue'][12], loc = 'center')

# Remove axes frame
ax1[3, 1].axis("off")
ax1[3, 2].axis("off")
ax1[3, 3].axis("off")

# Change x-axis values
plt.xticks(None)

# Set titles, caption and axis labels
fig.suptitle("Obama Approval Ratings By Issue", y = 1.08, fontsize = 2
0)
fig.text(0.305 , 1.02, "Approval ratings percents are calculated from
totals for\neach issue category.", ha = 'left', fontsize = 12, color =
'gray')
fig.text(0.305, .02, 'Source: Data Collected By Nathan Yau From Wikipe
dia', ha = 'left', color = 'gray')

# Remove frame
plt.box(on = None)

# show legend
ax1[0,0].legend(['Approve','Disapprove','None'], bbox_to_anchor = (-0.
5, 0.93), loc = "center left")

# Show plot
plt.show

# save file
fig.savefig("python_pie.png", bbox_inches="tight")
```


Obama Approval Ratings By Issue
Approval ratings percents are calculated from totals for each issue category.
Approve
Disapprove
None
Race Relations
52%
38%
10%
Education
49%
40%
11%
Terrorism
48%
45%
7%
Energy Policy
47%
42%
11%
Foreign Affairs
44%
48%
8%
Environment
43%
51%
6%
Situation in Iraq
41%
53%
6%
Taxes
41%
54%
5%
Healthcare Policy
40%
57%
3%
Economy
38%
59%
3%
Situation in Afghanistan
36%
57%
7%
Federal Budget Deficit
31%
64%
5%
Immigration
29%
62%
9%
Source: Data Collected By Nathan Yau From Wikipedia

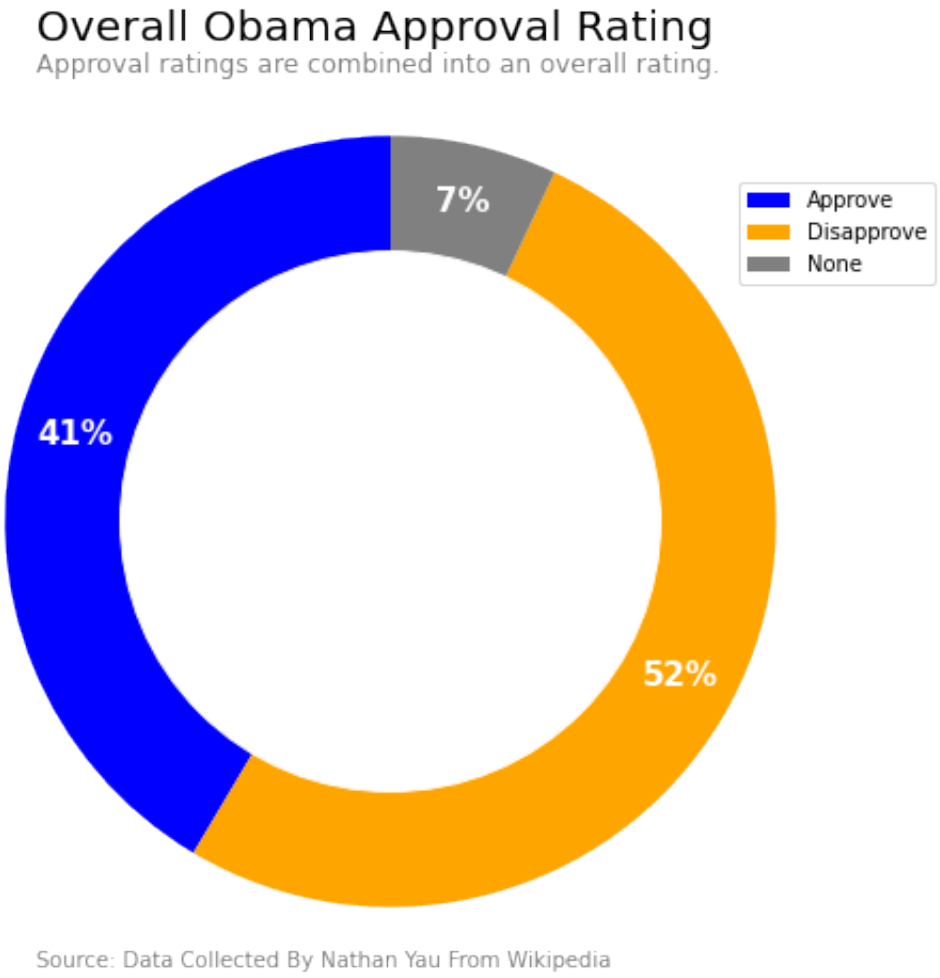
Donut Plot

In [291]:
```
# Use same data as pie chart "obama-approval-ratings.xls"

# sum each rating category, then convert to percentage for overall app
roval rating
tot_approve = sum(data_3['Approve'])
tot_disapprove = sum(data_3['Disapprove'])
tot_none = sum(data_3['None'])
tot_all = tot_approve + tot_disapprove + tot_none

perc_approve = (tot_approve/tot_all)
perc_disapprove = (tot_disapprove/tot_all)
perc_none = (tot_none/tot_all)

# create dataframe of totals
d = {'Approve': [perc_approve],
     'Disapprove': [perc_disapprove],
     'None': [perc_none],
}

data_3_donut = pd.DataFrame(d, columns = ['Approve', 'Disapprove', 'No
ne'])
print(data_3_donut)
```

```
    Approve  Disapprove  None
0  0.414615    0.515385  0.07
```

In [341]:
```
# Create axes and figure
fig, ax1 = plt.subplots()

# Set figure size & spacing
fig.set_size_inches(7, 7)
fig.tight_layout()

# Set wedge colors
colour = ['blue', 'orange', 'gray']

# Add plot to figure
ax1.pie(data_3_donut,  startangle = 90, colors = colour, autopct = '%1
.f%%', textprops = dict(color = "w", size = 15, weight = 'bold'), pctd
istance = 0.85)

# add a circle at the center
my_circle=plt.Circle( (0,0), 0.7, color='white')
p=plt.gcf()
p.gca().add_artist(my_circle)

# Change x-axis values
plt.xticks(None)

# Set titles, caption and axis labels
fig.suptitle("Overall Obama Approval Rating", y = 0.99, fontsize = 20)
fig.text(0.185 , 0.93, "Approval ratings are combined into an overall
rating.", ha = 'left', fontsize = 12, color = 'gray')
fig.text(0.185, .1, 'Source: Data Collected By Nathan Yau From Wikiped
ia', ha = 'left', color = 'gray')

# Remove frame
plt.box(on = None)
ax1.axis("off")

# show legend
ax1.legend(['Approve','Disapprove','None'], bbox_to_anchor = (0.85, 0.
8), loc = "center left")

# Show plot
plt.show

# save file
fig.savefig("python_donut.png", bbox_inches="tight")
```

```
/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site
-packages/ipykernel_launcher.py:12: MatplotlibDeprecationWarning: No
n-1D inputs to pie() are currently squeeze()d, but this behavior is
deprecated since 3.1 and will be removed in 3.3; pass a 1D array ins
tead.
  if sys.path[0] == '':
```



In [ ]: