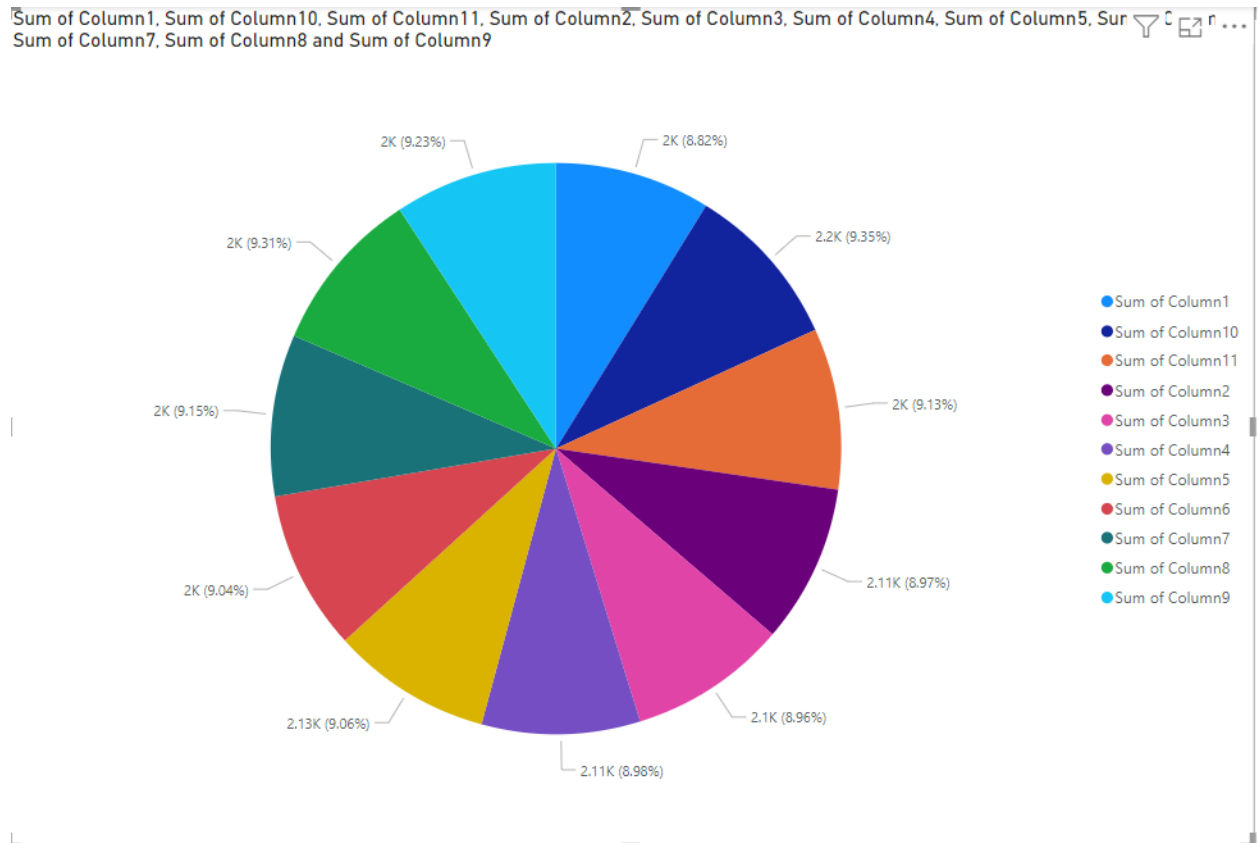


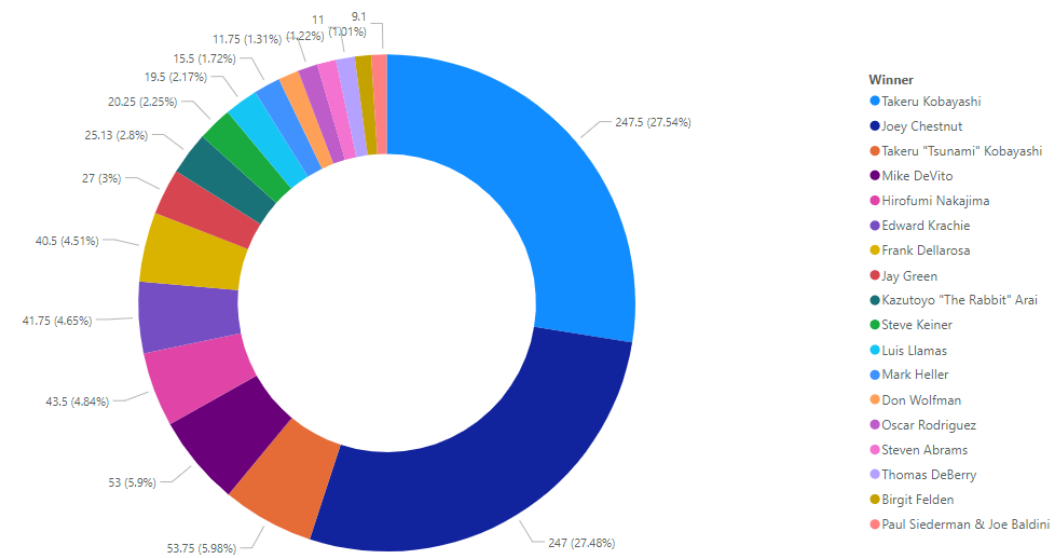
#googled if possible to change the title of powerbi and it said no

Power Bi

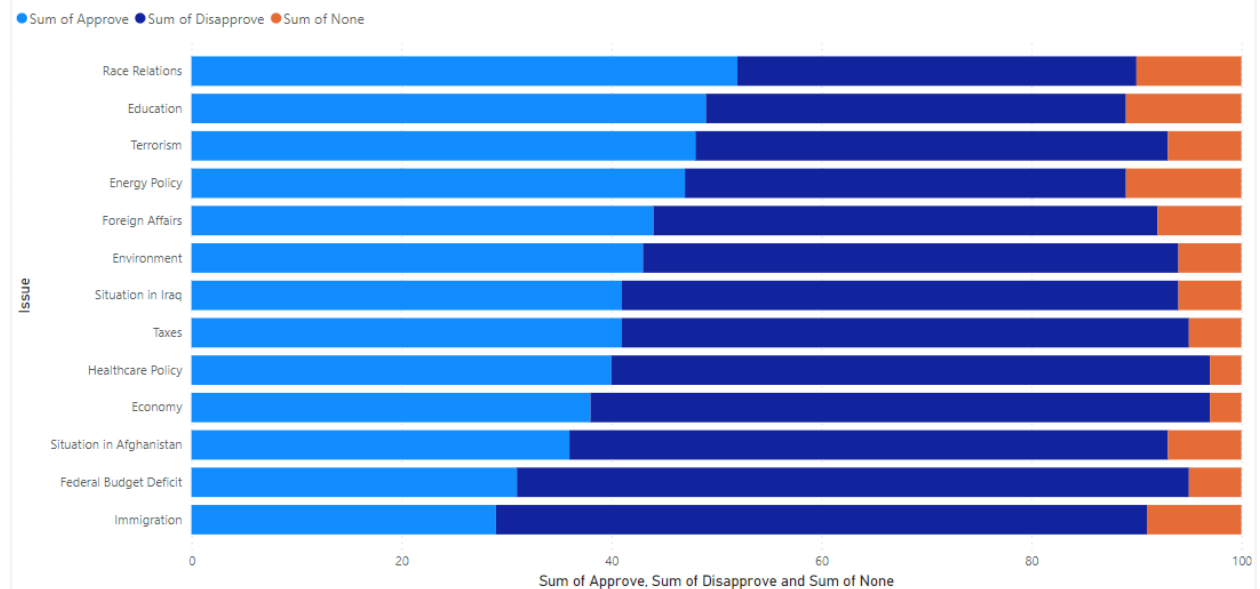
This is from the hotdog places dataset. It automatically summed up every column.



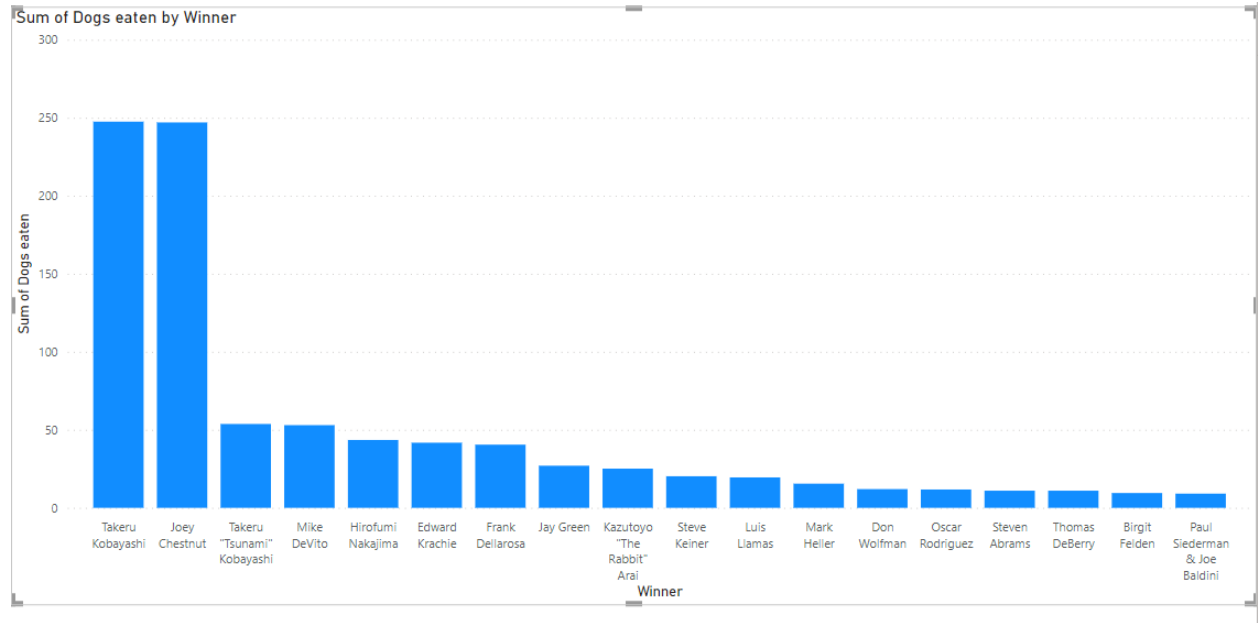
Sum of Dogs eaten by Winner



Sum of Approve, Sum of Disapprove and Sum of None by Issue

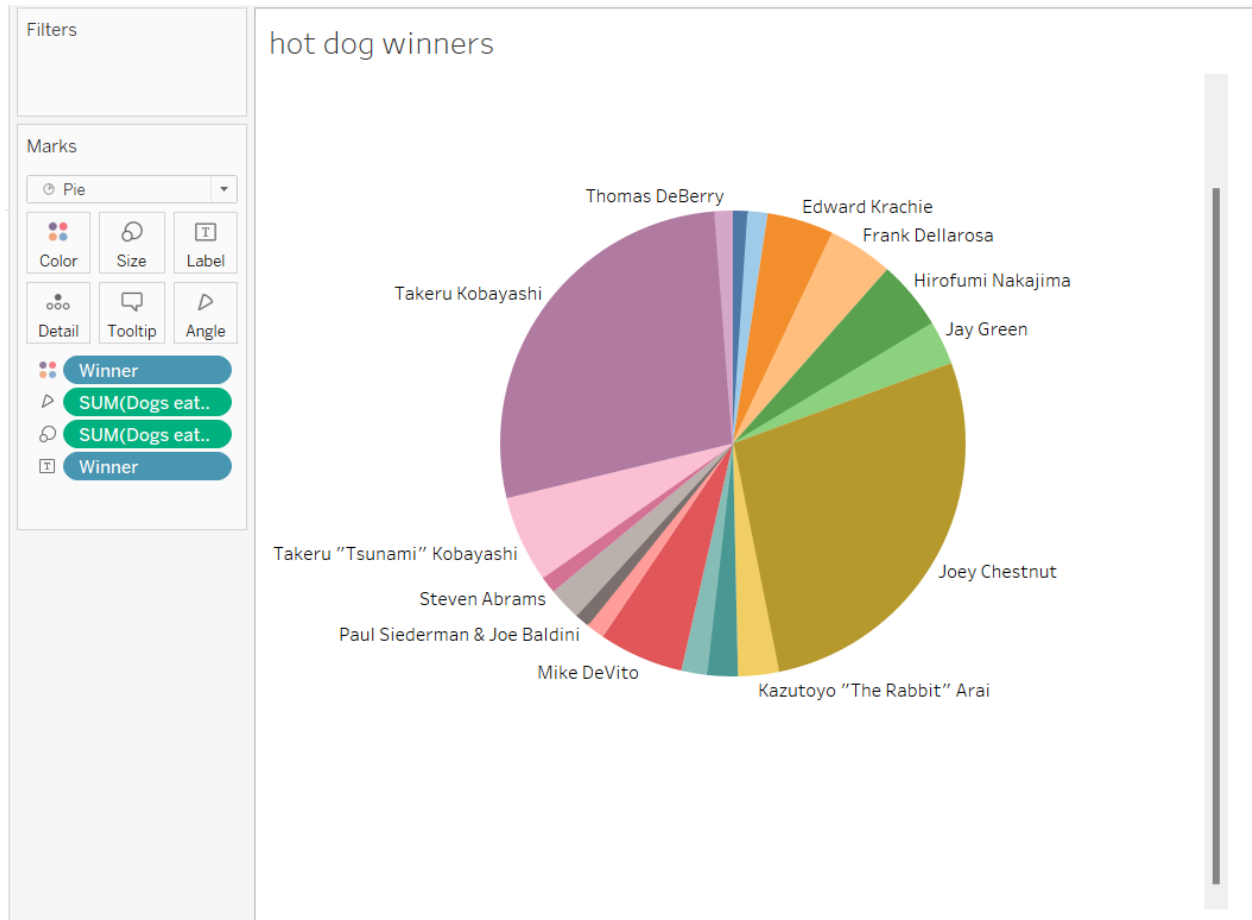


#hot dog winners bar chart



Tableau

#hot dog winner pie chart

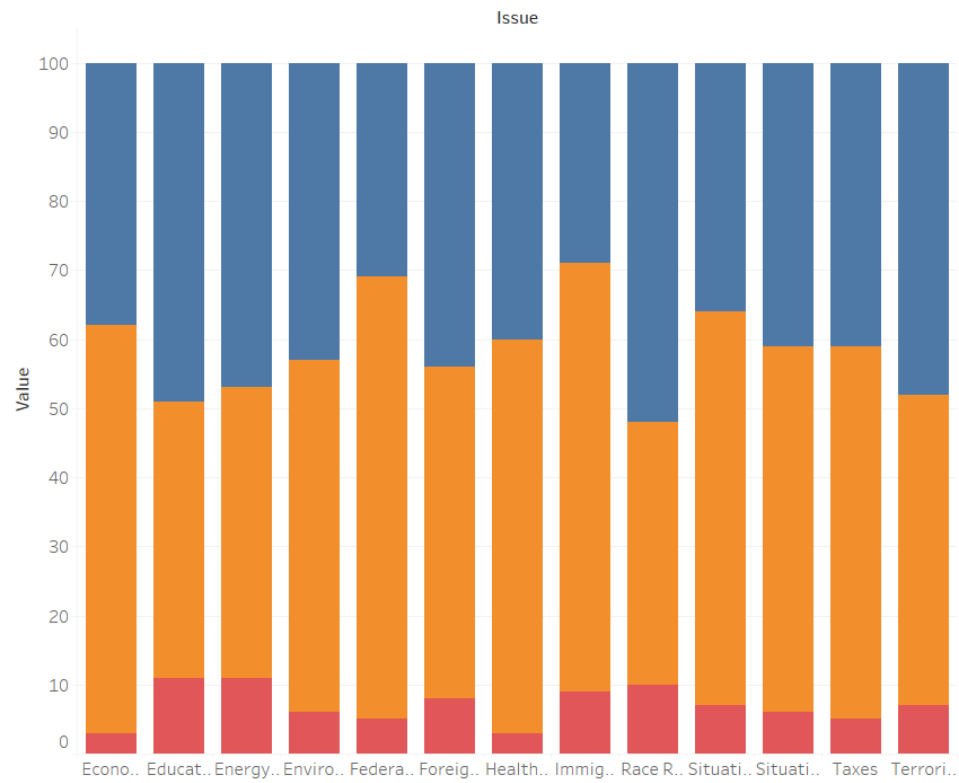


#could not figure out donut chart in tableau but i got it in power bi! Was just trying to do some extra work.

#stacked bar graph for obama ratings

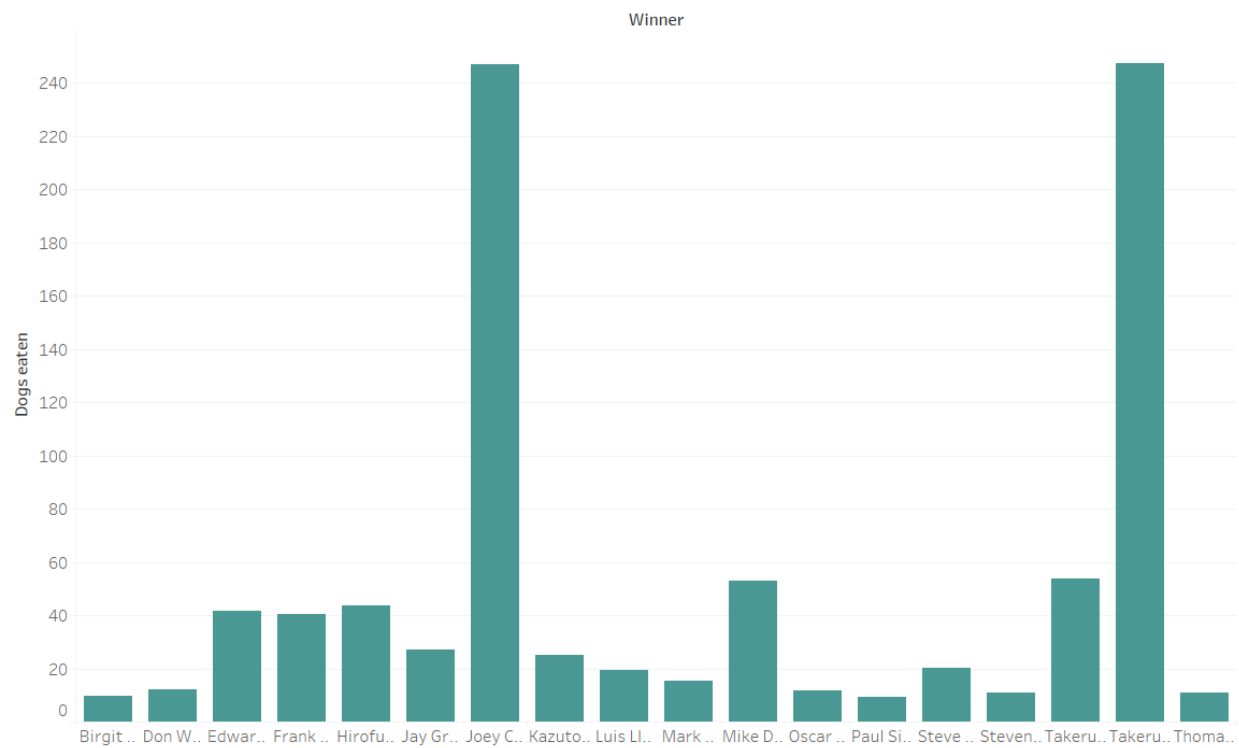
Columns	Issue
Rows	Measure Values

Sheet 1



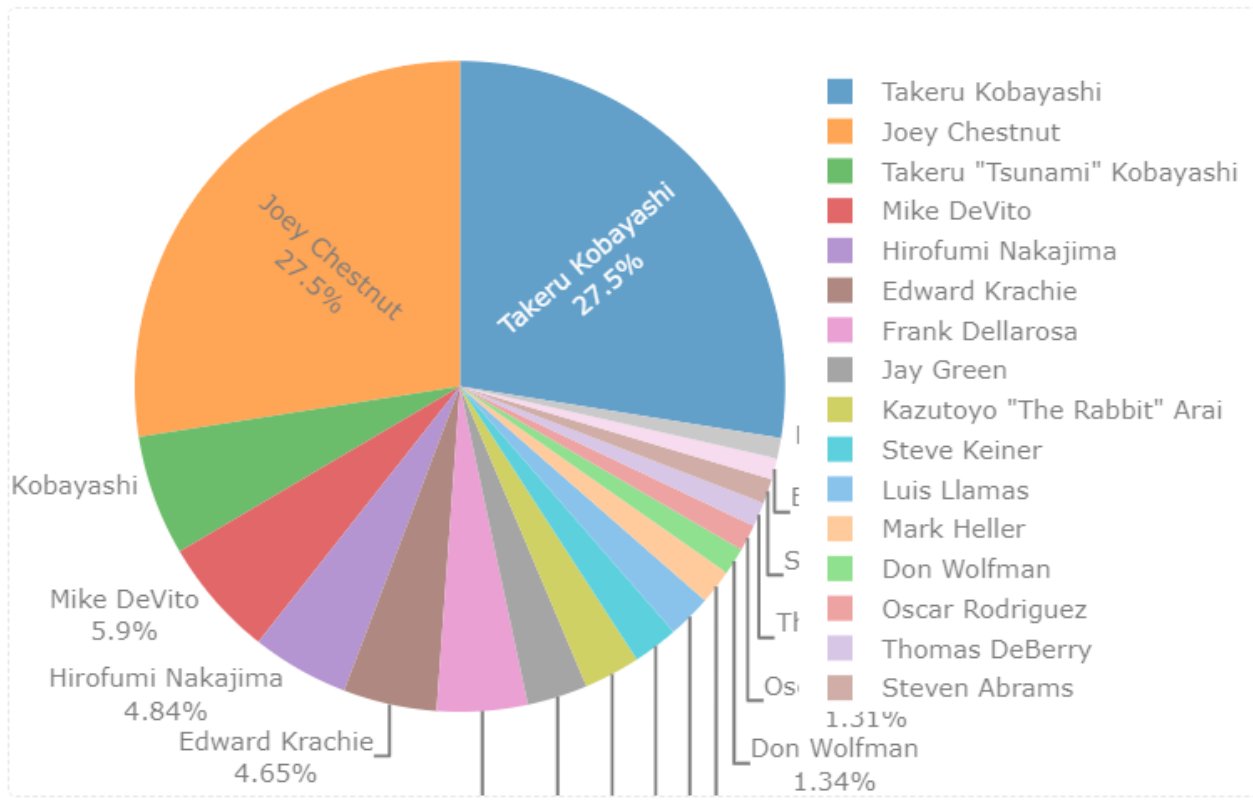
#hot dog winners bar chart

Hot Dog Contest Winners Dog Eaten

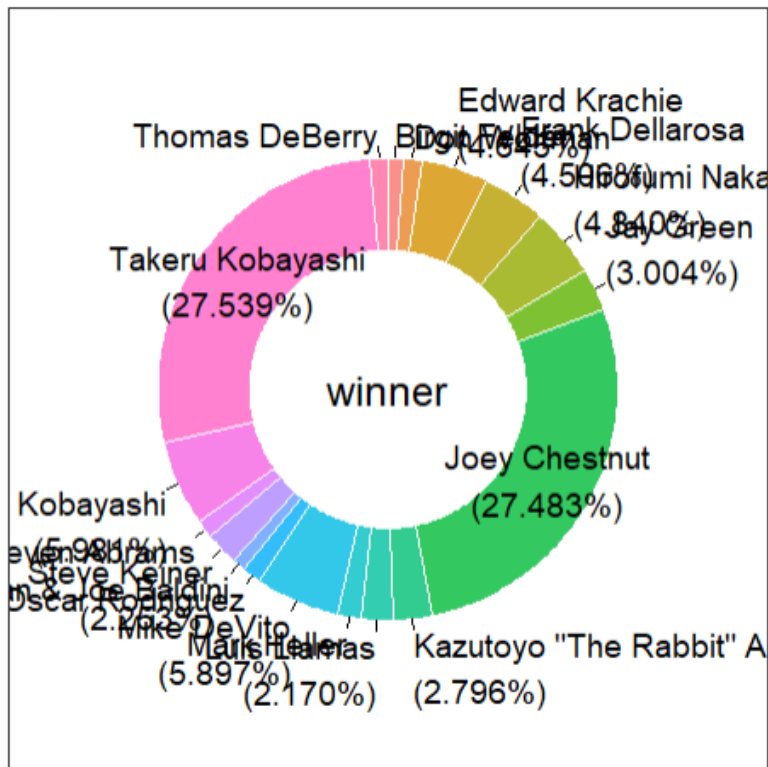


RSTUDIO

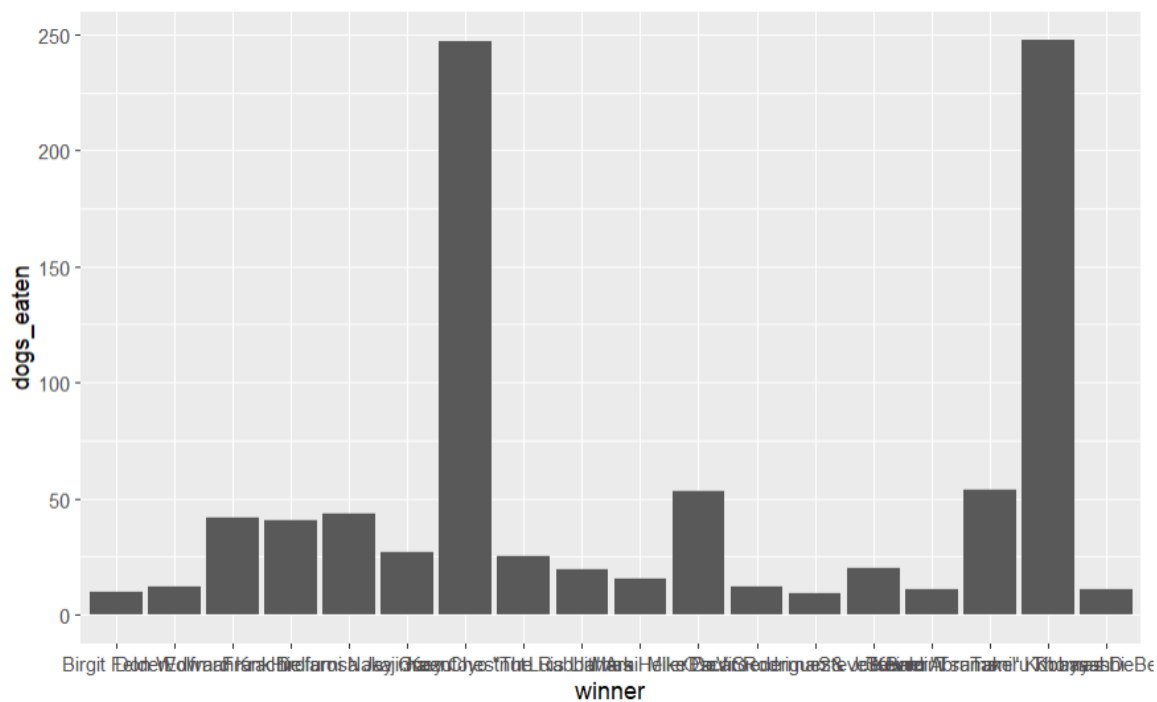
Pie chart of hotdog winners



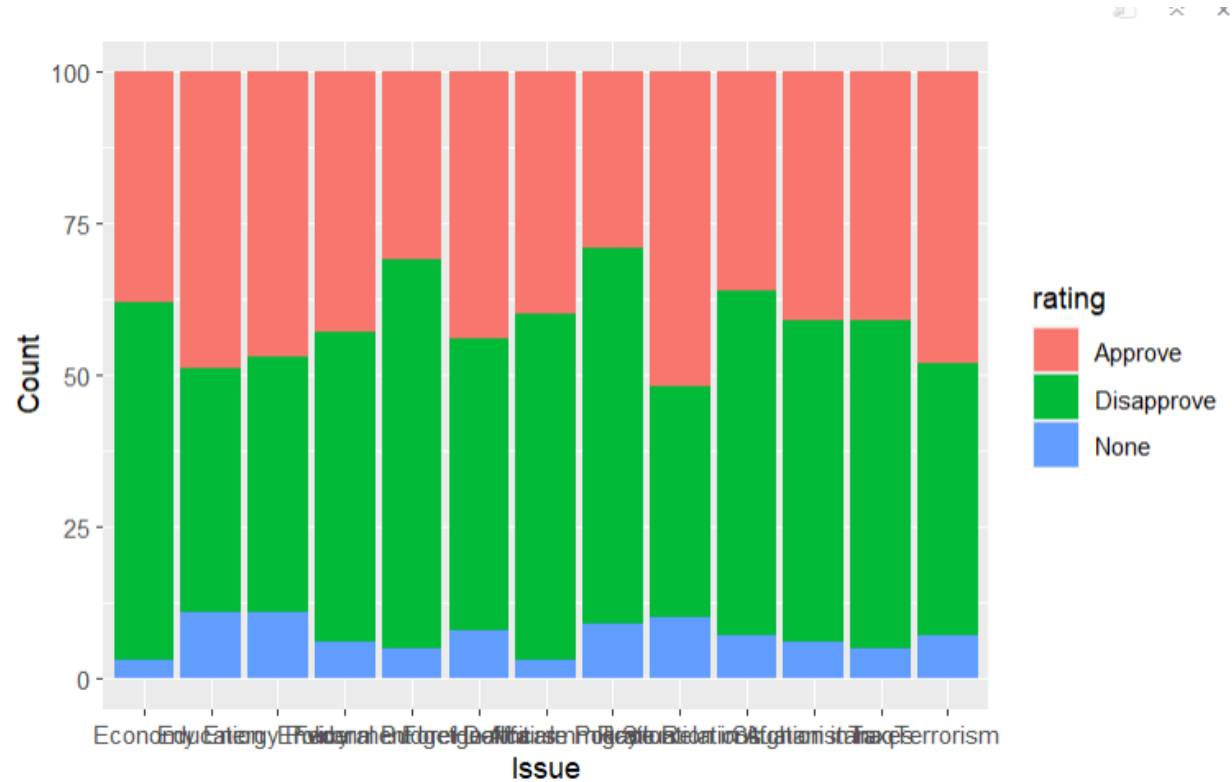
Donut chart of hotdog winners



Bar chart of hot dog winners



Stacked bar chart of obama ratings



```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
In [2]: #import data to dataframe
df_hcw=pd.read_excel('hotdog-contest-winners.xlsx')
df_hcw
```

Out[2]:

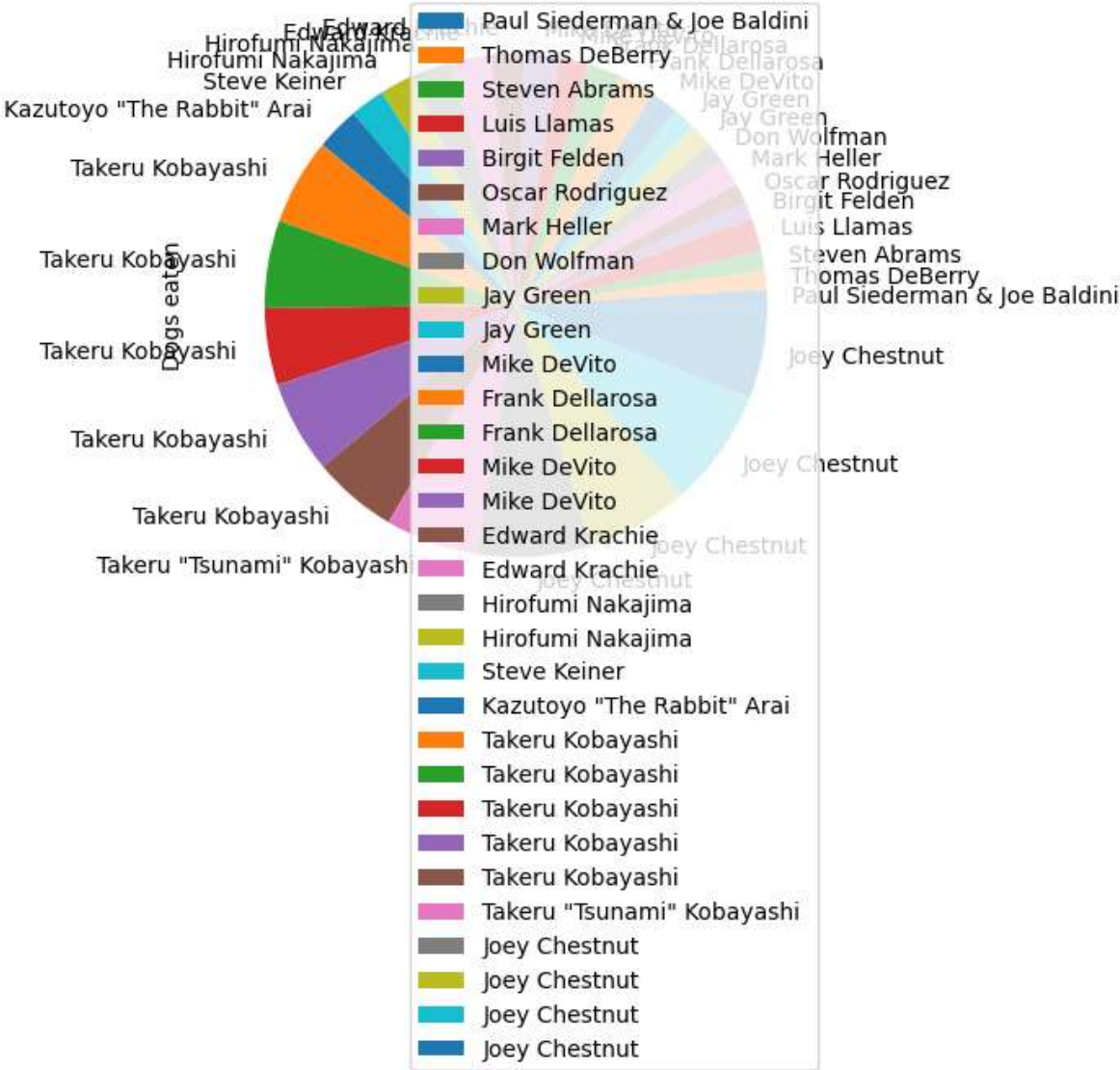
	Year	Winner	Dogs eaten	Country	New record
0	1980	Paul Siederman & Joe Baldini	9.10	United States	0
1	1981	Thomas DeBerry	11.00	United States	0
2	1982	Steven Abrams	11.00	United States	0
3	1983	Luis Llamas	19.50	Mexico	0
4	1984	Birgit Felden	9.50	Germany	0
5	1985	Oscar Rodriguez	11.75	United States	0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

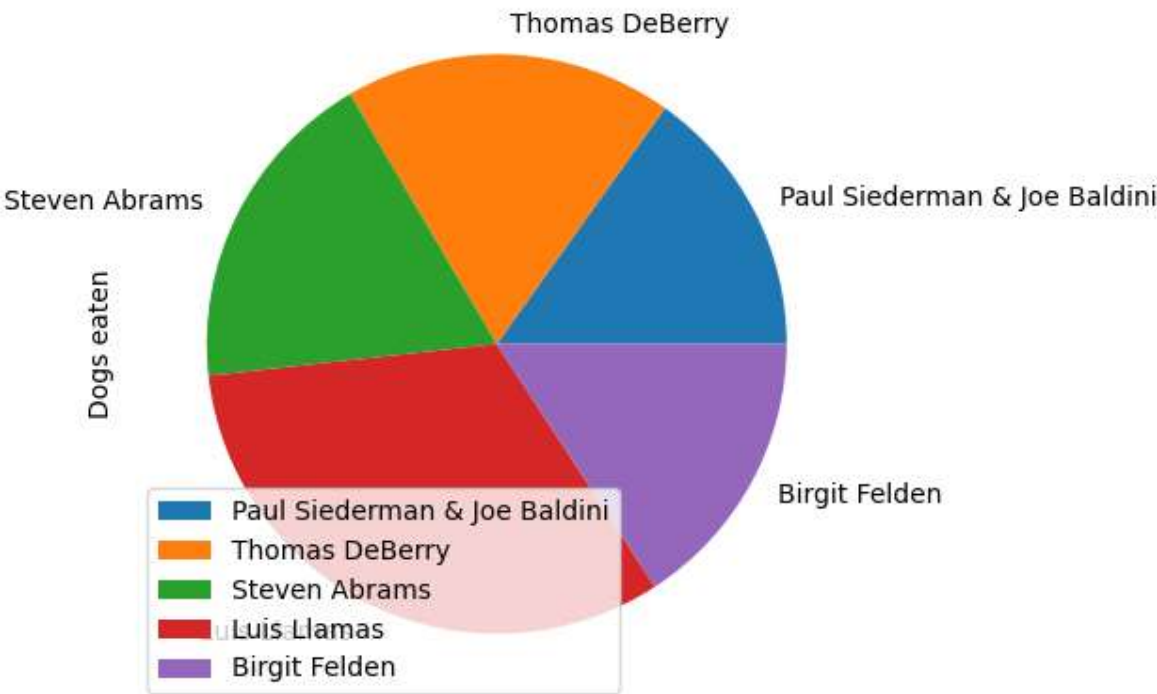
Out[3]:

	Year	Winner	Dogs eaten	Country	New record
0	1980	Paul Siederman & Joe Baldini	9.1	United States	0
1	1981	Thomas DeBerry	11.0	United States	0
2	1982	Steven Abrams	11.0	United States	0
3	1983	Luis Llamas	19.5	Mexico	0
4	1984	Birgit Felden	9.5	Germany	0

```
In [4]: #pie chart
plot = df_hcw.plot.pie(y='Dogs eaten', labels = df_hcw['Winner'], figsize=(5,
```

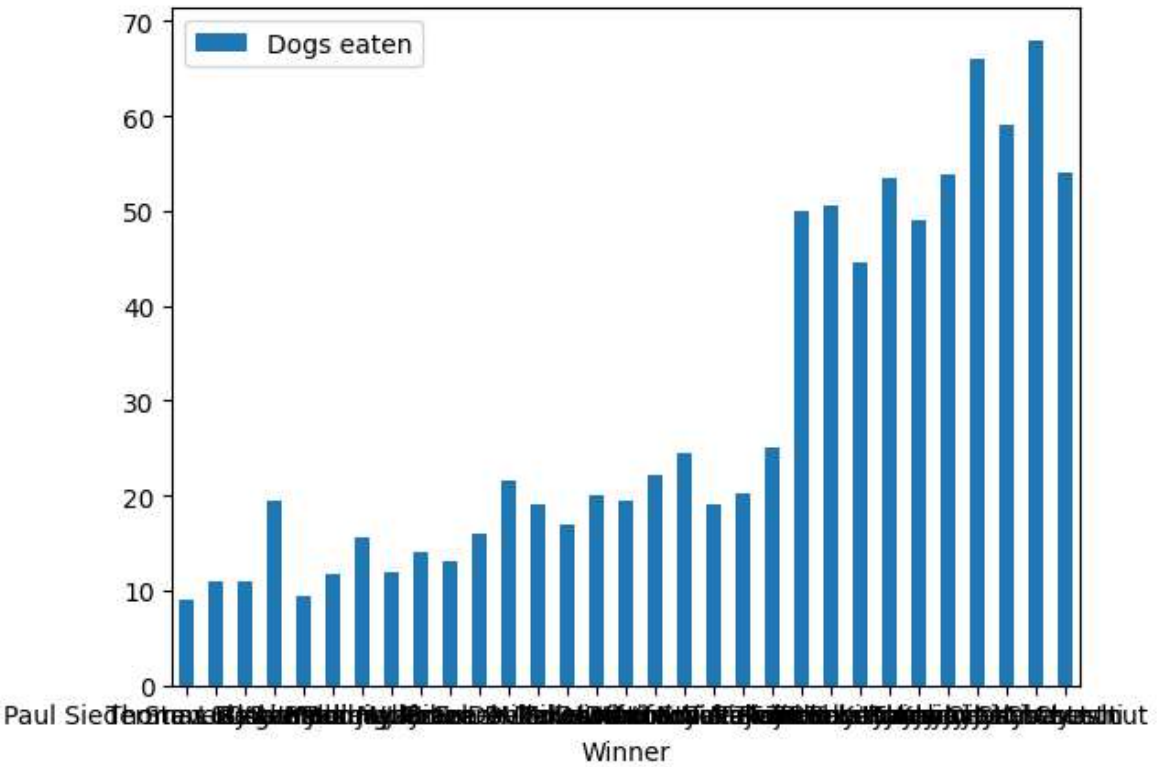


```
In [5]: #just used the head since there's a lot of data.
plot = df_hcw_h.plot.pie(y='Dogs eaten', labels = df_hcw['Winner'], figsize=(5
```



```
In [11]: #bar chart of hot dog winners with count of dogs eaten
ax = df_hcw.plot.bar(x='Winner', y='Dogs eaten', rot=0)
ax
```

Out[11]: <AxesSubplot: xlabel='Winner'>



```
In [6]: #import data to dataframe
df_hp=pd.read_excel('hotdog-places.xlsm')
df_hp.index += 1
df_hp['index'] = df_hp.index
df_hp
```

Out[6]:

	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	index
1	25	50.0	50.5	44.5	53.5	49	54	66	59	68.0	54	1
2	24	31.0	26.0	30.5	38.0	37	52	63	59	64.5	43	2
3	22	23.5	25.5	29.5	32.0	32	37	49	42	55.0	37	3

```
In [7]: #donut chart
labels = ["1st place", "2nd place", "3rd place"]

colors = ['#FF0000', '#0000FF', '#FFFF00',
          '#ADFF2F', '#FFA500']

explode = (0.05, 0.05, 0.05)

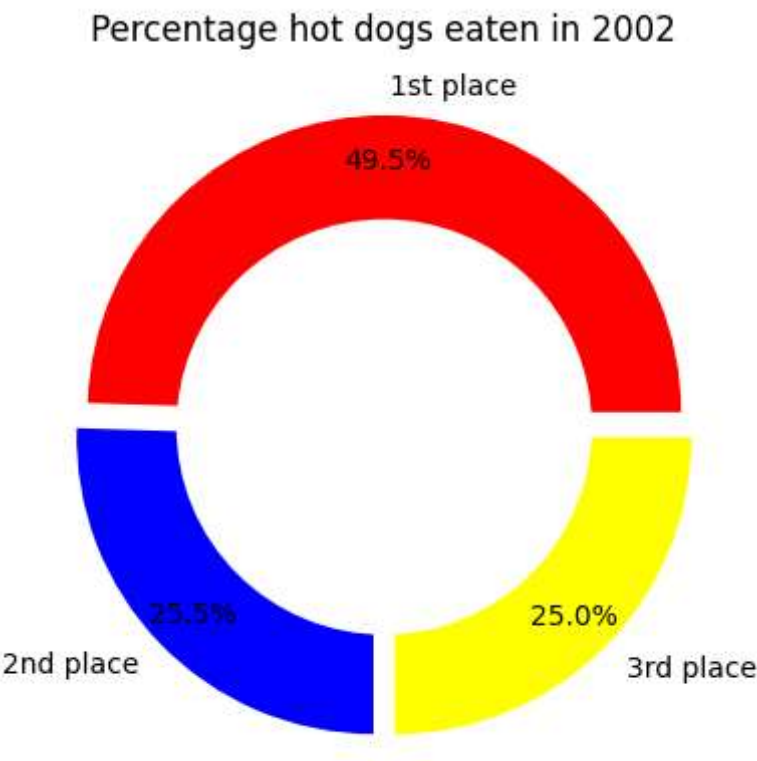
plt.pie(df_hp[2002], colors=colors, labels=labels,
        autopct='%1.1f%%', pctdistance=0.85,
        explode=explode)

centre_circle = plt.Circle((0, 0), 0.70, fc='white')
fig = plt.gcf()

fig.gca().add_artist(centre_circle)

plt.title('Percentage hot dogs eaten in 2002')

plt.show()
```



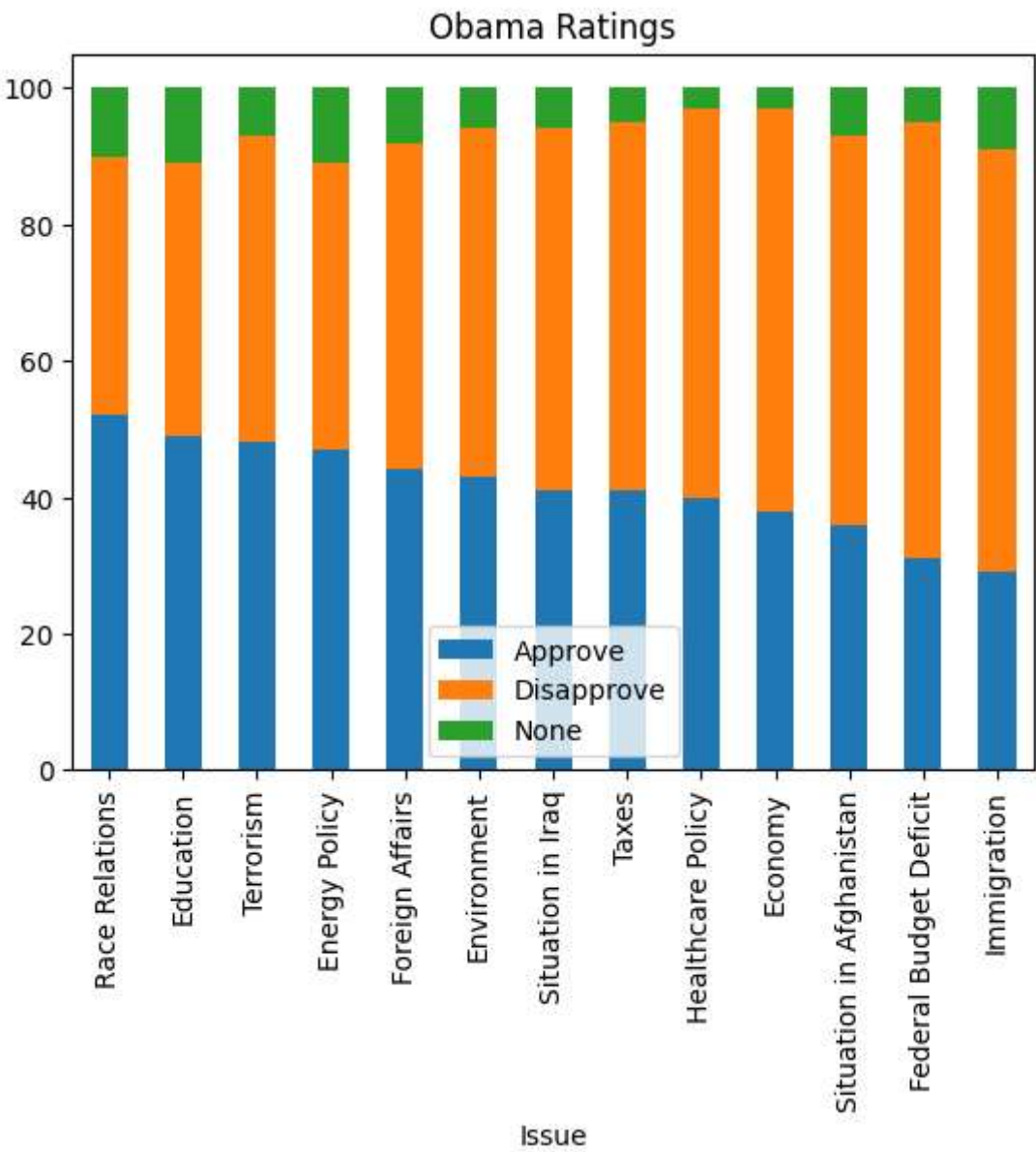
```
In [8]: #import data to dataframe
df_oar=pd.read_excel('obama-approval-ratings.xls')
df_oar
```

Out[8]:

	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 [9]: #stacked bar chart
df_oar.plot.bar(x='Issue', stacked=True, title='Obama Ratings')
```

Out[9]: <AxesSubplot: title={'center': 'Obama Ratings'}, xlabel='Issue'>



```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```

---
title: "CHARTS"
output: html_document
date: "2023-11-29"
---

```{r setup, include=FALSE}
setwd("C:/Users/brean/OneDrive/Desktop/640/week1/breannaparker640week1")
knitr::opts_chunk$set(echo = TRUE)
```

## R Markdown

```{r}
library(readxl)
library(tidyr)
library(ggplot2)
library(gcookbook)

data_hcw <- read_excel("hotdog-contest-winners.xlsx")
data_hcw

```

```{r}
colnames(data_hcw) <- c('year', 'winner', 'dogs_eaten', 'country', 'new_record')

data_hcw

```

```{r}

library(plotly)

top_descend <- data_hcw[order(-data_hcw$dogs_eaten),]

fig <- plot_ly(type='pie', labels=top_descend$winner, values=top_descend$dogs_eaten,
 textinfo='label+percent',
 insidetextorientation='radial')
fig

```

```{r}
library(readxl)

data_hp <- read_excel("hotdog-places.xlsx")
data_hp

```

```{r}
rownames(data_hp) <- 1 : 3
data_hp

```

```{r}

```

```

#could not figure out pie chart with hotdog places so had to use hotdog winners again
library(plotly)
library(webr)

PieDonut(data_hcw, aes(winner, count = dogs_eaten),
 r0=0.6)

...

```{r}
library(readxl)
data_oar <- read_excel("obama-approval-ratings.xls")
data_oar
...

```{r}

data_pivot <-pivot_longer(data_oar, cols = c("Approve", "Disapprove", "None"), names_to =
"rating", values_to = "Count")

...

```{r}
ggplot(data_pivot, aes(x = Issue, y = Count, fill = rating)) +
  geom_col()
...

```