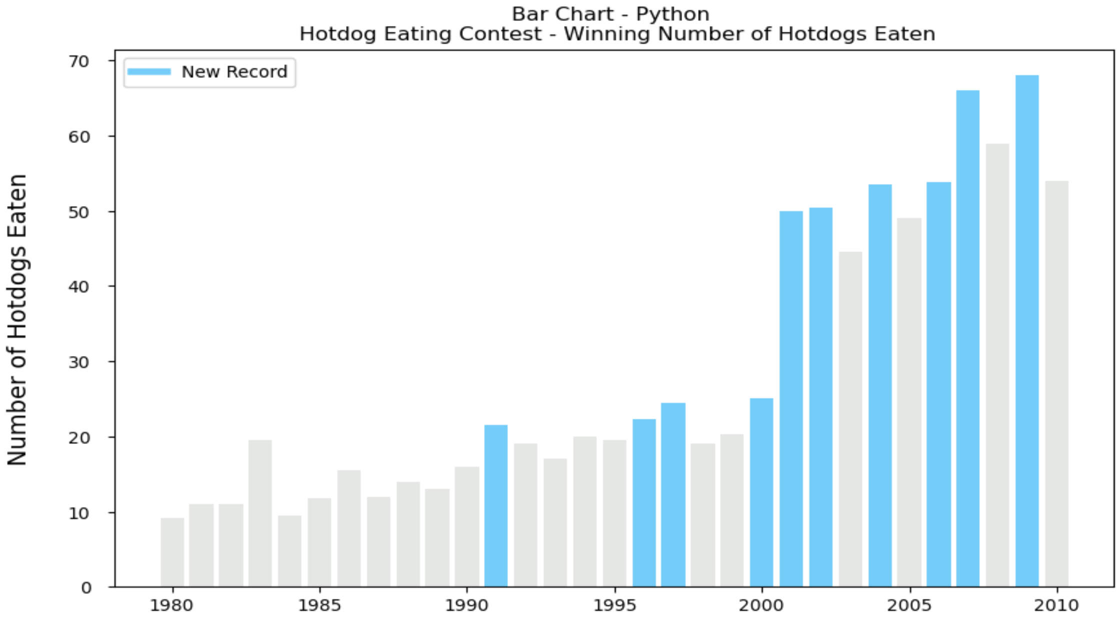
DSC 640: Weeks 1 – 2
Author: Kimberly Cable
Date: Sept 10, 2022
Exercise 1.2 Charts

Bar Charts

Python

```
plt.rcdefaults()
fig, ax = plt.subplots(figsize=(10, 6))

# Change color if new record
colors = ["#74cdf9" if record == 1 else "#e5e7e4" for record in hotdog_winners_df['New record']]

# plot bar chart
ax.bar(hotdog_winners_df['Year'], hotdog_winners_df['Dogs eaten'], color = colors)

# set labels
ax.set_ylabel('Number of Hotdogs Eaten', fontsize = 14)
ax.tick_params(axis='y', which='major', pad = 10)
ax.legend([Line2D([0], [0], color = '#74cdf9', lw = 4)], ['New Record'])
ax.set_title('Bar Chart - Python \n Hotdog Eating Contest - Winning Number of Hotdogs Eaten')
ax.yaxis.labelpad = 20.0

plt.show()

# Save figure
ax.get_figure().savefig('images/barchart-python.png',
        bbox_inches = 'tight',
        transparent = True)
```



R

~~~
```{r}
#| label: barchart
#| fig-width: 6.5

# Highlight record years with color
fill_colors <- c()

for ( i in 1:length(hotdog_winners_df$New_record) ) {
        if (hotdog_winners_df$New_record[i] == 1) {
                fill_colors <- c(fill_colors, "#74cdf9")
        } else {
                fill_colors <- c(fill_colors, "#e5e7e4")
        }
}

fig <- plot_ly(hotdog_winners_df, x = ~Year, y = ~Dogs_eaten, type = 'bar',
    marker = list(color = fill_colors, hoverinfo = 'none'))

fig <- fig %>%
  layout(
    title = "Bar Chart - R \nHotdog Eating Contest \nWinning Number of Hotdogs Eaten",
    xaxis = list(showgrid = FALSE,
          title = ""),
    yaxis = list(showgrid = FALSE,
          title = "Hotdogs Eaten",
          titlefont = list(size = 22)),
    margin = list(l = 5, r = 5, b = 10, t = 30, pad = 10)
  )
```
~~~

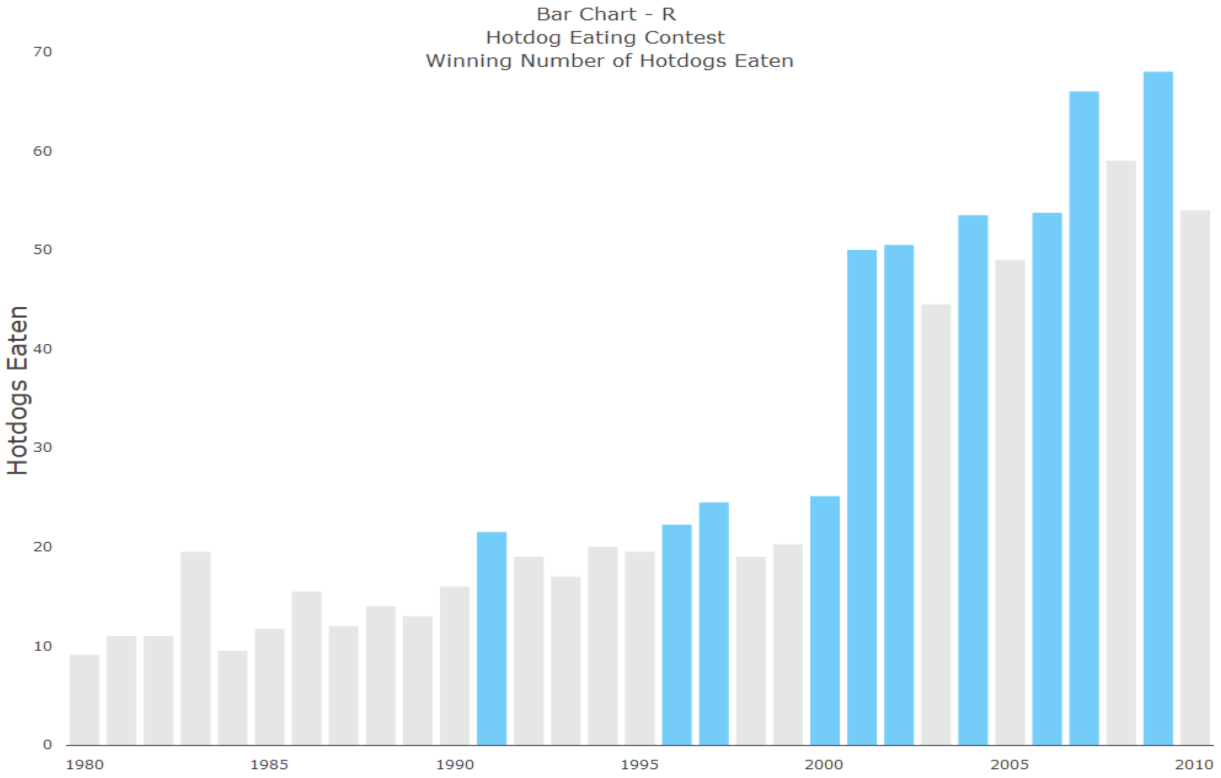
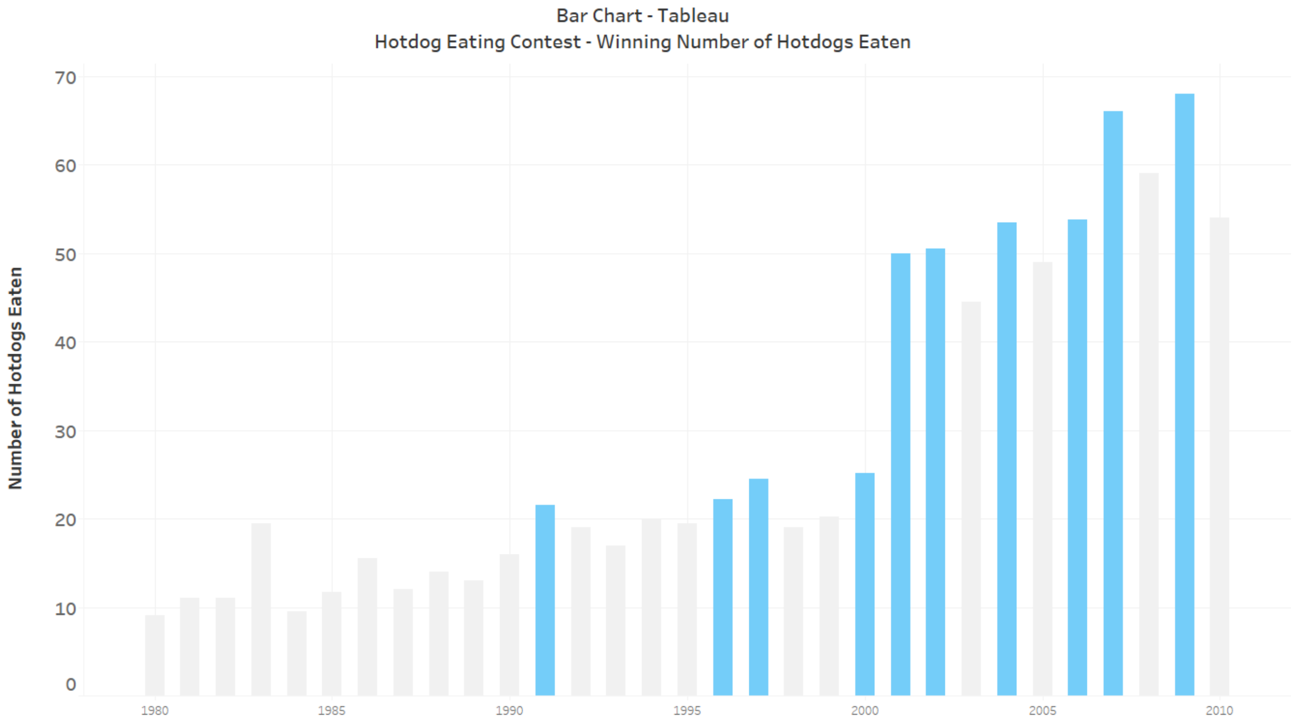


Tableau
(Please see Weeks1-2_Tableau.twb for code)

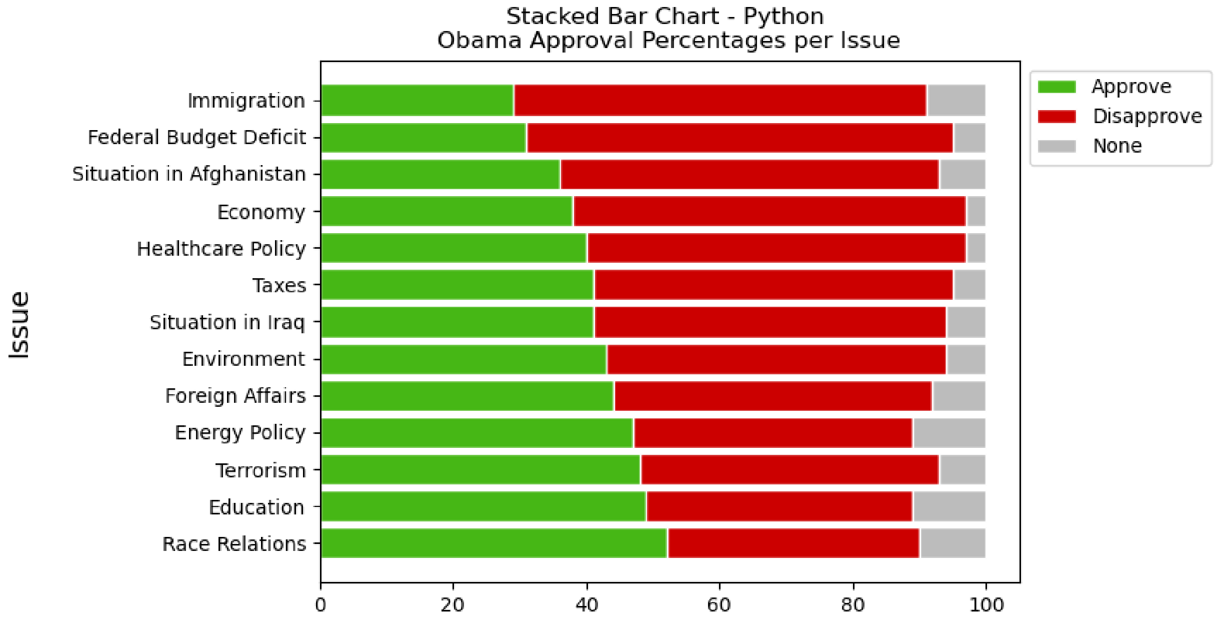


Stacked Bar Chart:

Python

```
# Get percentages of Approve/Disapprove/None
totals = [i + j + k for i, j, k in zip(obama_df['Approve'], obama_df['Disapprove'], obama_df['None'])]
approve = [i / j * 100 for i, j in zip(obama_df['Approve'], totals)]
disapprove = [i / j * 100 for i, j in zip(obama_df['Disapprove'], totals)]
none = [i / j * 100 for i, j in zip(obama_df['None'], totals)]
```

```
plt.rcdefaults()

fig, ax = plt.subplots()

# plot
barHeight = 0.85
names = obama_df['Issue']

# Create Approve
ax.barh(names, approve, color='#46b715', edgecolor='white', height=barHeight, label = 'Approve')

# Create Disapprove
ax.barh(names, disapprove, left=approve, color='#cc0000', edgecolor='white', height=barHeight, label = 'Disapprove')

# Create None
ax.barh(names, none, left=[i+j for i,j in zip(approve, disapprove)], color='#bcbcbc', edgecolor='white',
    height=barHeight, label = 'None')

# Set labels
plt.ylabel("Issue", fontsize = 14)
plt.legend(loc='upper left', bbox_to_anchor=(1,1), ncol=1)
plt.title('Stacked Bar Chart - Python \nObama Approval Percentages per Issue')
ax.yaxis.labelpad = 20.0

plt.show()

# Save figure
ax.get_figure().savefig('images/stackedbarchart-python.png',
     bbox_inches = 'tight',
     transparent = True)
```



R

~~~
```{r}
#| label: stackedchart
#| fig-width: 4.5

names <-  obama_df$Issue

top_labels <- c('Approve', 'Disapprove', 'None')

fig <- plot_ly(obama_df, x = ~Approve, y = ~names, type = 'bar',
        orientation = 'h', marker = list(color = '#46b715',
                        line = list(color = '#ffffff',
                              width = 1.5)))
fig <- fig %>% add_trace(x = ~Disapprove, marker = list(color = '#cc0000'))
fig <- fig %>% add_trace(x = ~None, marker = list(color = '#bcbcbc'))

fig <- fig %>%
  layout(xaxis = list(title = "",
         showgrid = FALSE,
         showline = FALSE,
         showticklabels = TRUE,
         zeroline = FALSE,
         domain = c(0.15, 1)),
  yaxis = list(title = "",
         showgrid = FALSE,
         showline = FALSE,
         showticklabels = FALSE,
         zeroline = FALSE,
         categoryorder = 'category descending'),
  barmode = 'stack',
  showlegend = FALSE,
  title = "Stacked Bar Chart - R \nObama Approval Percentages per Issue",
  autosize = F,
  margin = list(l = 70, r = 5, b = 5, t = 120, pad = 10)
  )

# labeling the y-axis
fig <- fig %>%
  add_annotations(xref = 'paper',
          yref = names, x = 0.14,
          y = names,
          xanchor = 'right',
          text = names,
          font = list(family = 'Arial', size = 14,
               color = '#000000'),
          showarrow = FALSE, align = 'right')

# label top titles
fig <- fig %>%
  add_annotations(xref = 'x', yref = 'paper',
          x = c(43 / 2, 45 + 42 / 2, 43 + 42 + 21 / 2),
          y = 1.07,
          text = top_labels,
          font = list(family = 'Arial', size = 12,
                color = '#000000'),
          showarrow = FALSE)

export(fig, file = "images/stackedbarchart-r.png")
```
~~~

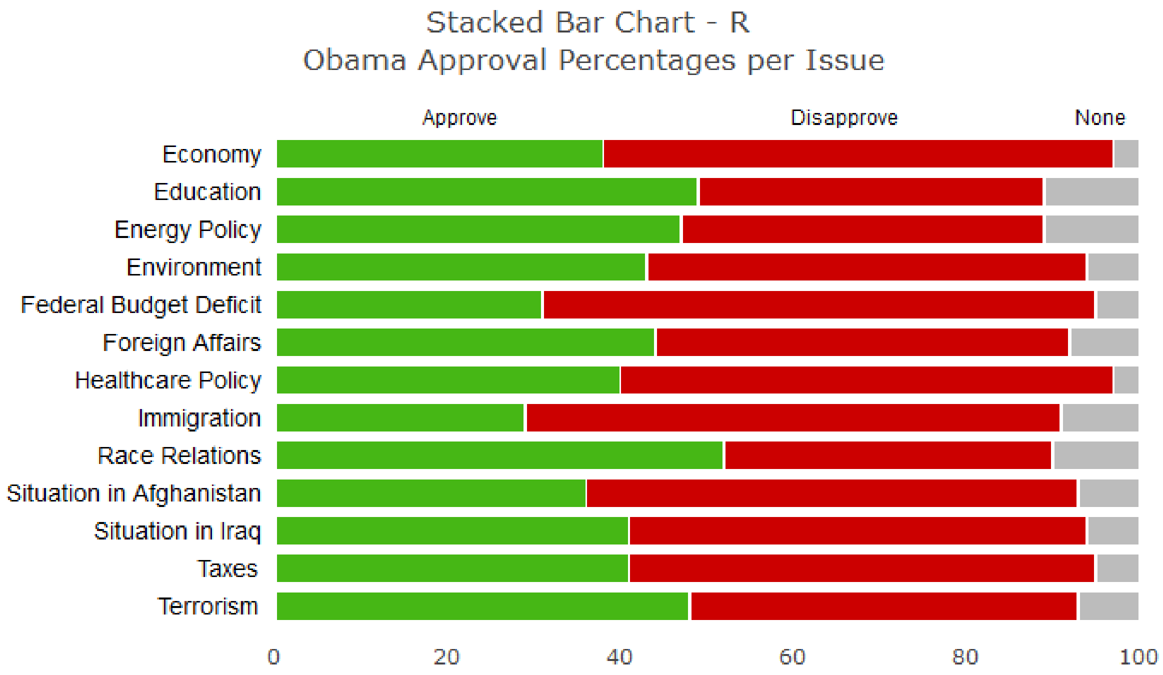
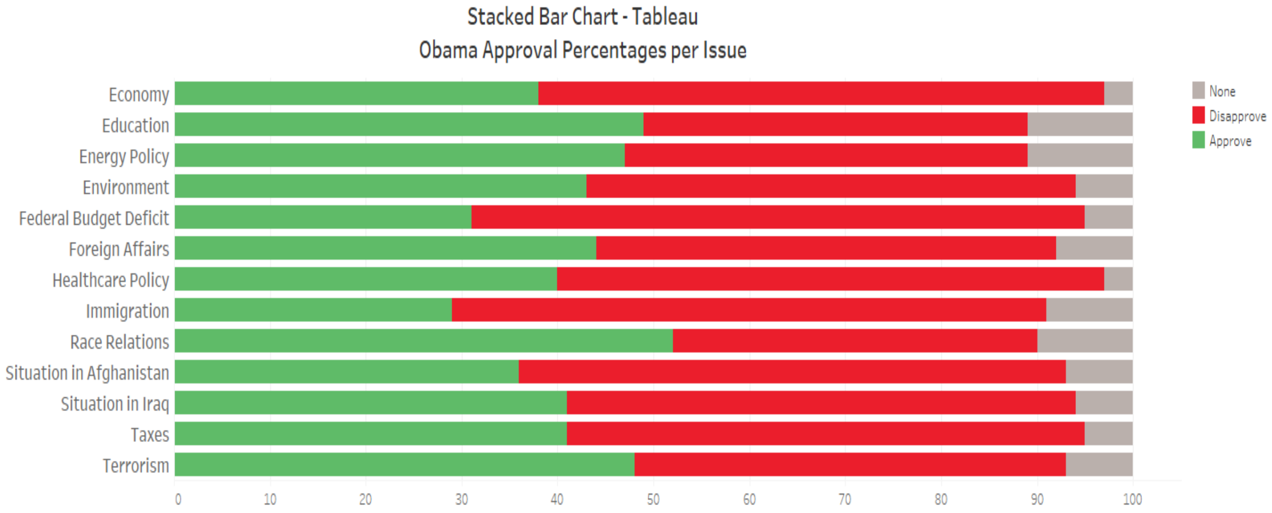


Tableau
(Please see Weeks1-2_Tableau.twb for code)

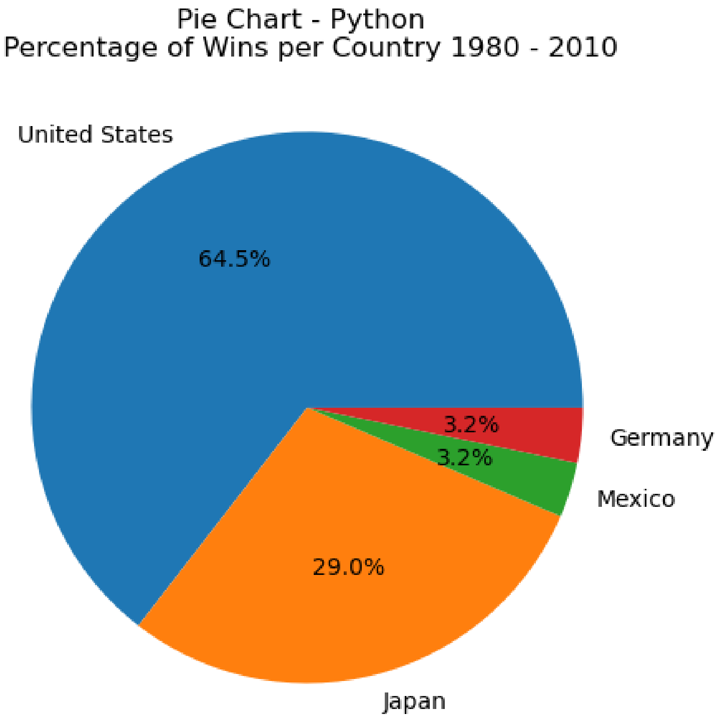


Pie Chart:

Python

```
countries = hotdog_winners_df['Country'].value_counts()
```

```
plt.rcdefaults()

fig, ax = plt.subplots()

ax.pie(countries.values, labels = countries.index, autopct='%1.1f%%',)
ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

# Title
plt.title('Pie Chart - Python \n Percentage of Wins per Country 1980 - 2010', pad = 20)

plt.show()

# Save figure
ax.get_figure().savefig('images\piechart-python.png',
        bbox_inches = 'tight',
        transparent = True)
```



R

~~~
```{r}
#| label: piechart
#| fig-width: 6.5

# Get the counts for each country
countries = dplyr::count(hotdog_winners_df, Country, sort = FALSE)

# plot pie chart
fig <- plot_ly(countries, labels = ~Country, values = ~n, type = 'pie',
        textinfo = 'percent', insidetextorientation = 'radial')

fig <- fig %>%
  layout(title = 'Pie Chart - R \n Percentage of Wins per Country 1980 - 2010',
    xaxis = list(showgrid = FALSE,
         zeroline = FALSE,
         showticklabels = FALSE),
    yaxis = list(showgrid = FALSE,
         zeroline = FALSE,
         showticklabels = FALSE),
    autosize = F,
    margin = list(l = 5, r = 5, b = 5, t = 120, pad = 10)
  )

export(fig, "images/piechart-r.png")
```
~~~

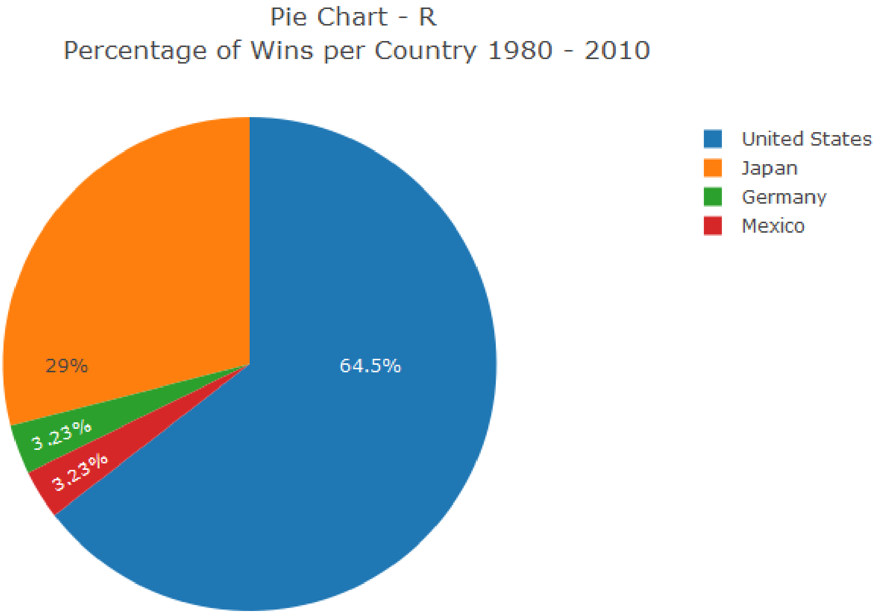
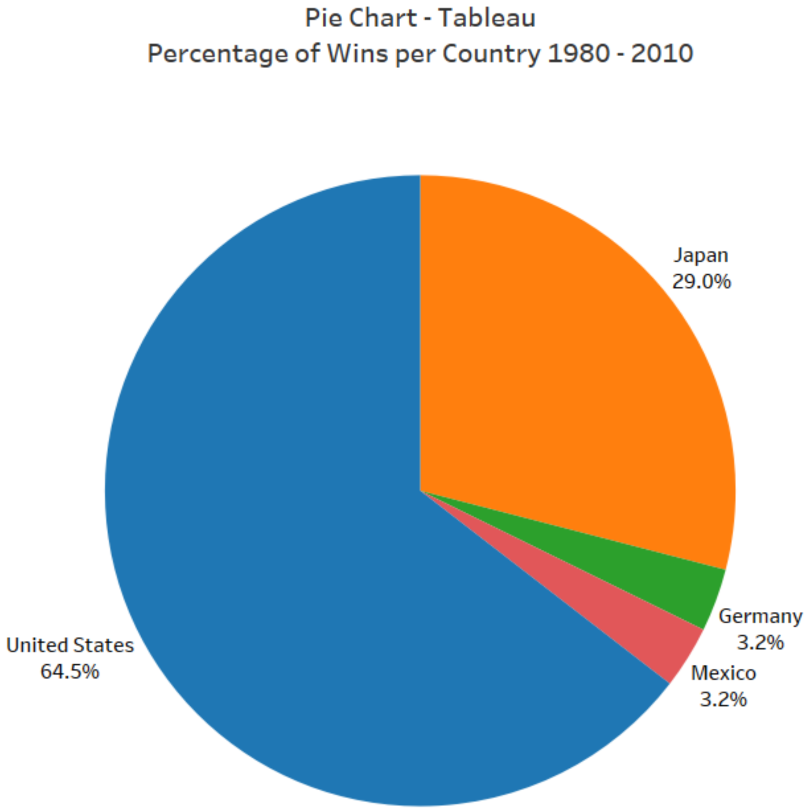


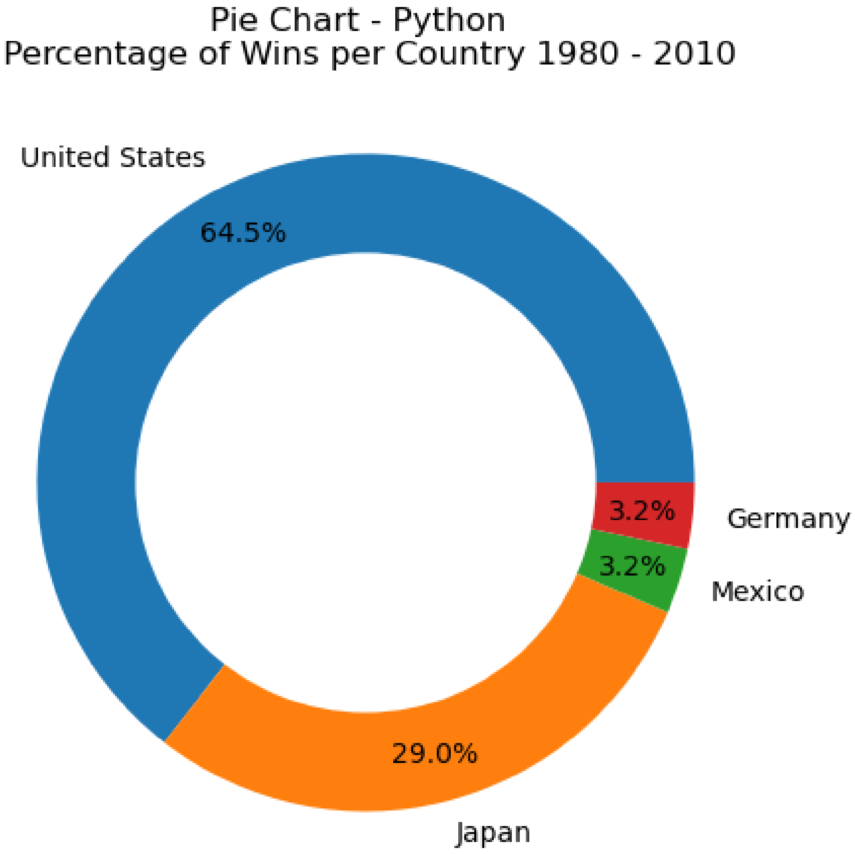
Tableau
(Please see Weeks1-2_Tableau.twb for code)



Donut Chart:

Python

```
plt.rcdefaults()

fig, ax = plt.subplots()

# Create a pie chart
ax.pie(countries.values, labels = countries.index, autopct='%1.1f%%', pctdistance = 0.85)
ax.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.

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

# Adding Circle in Pie chart
fig.gca().add_artist(centre_circle)

# Title
plt.title('Pie Chart - Python \n Percentage of Wins per Country 1980 - 2010', pad = 20)

plt.show()

# Save figure
ax.get_figure().savefig('images\donutchart-python.png',
        bbox_inches = 'tight',
        transparent = True)
```



R

~~~
```{r}
#| label: donutchart
#| fig-width: 6.5

# Get the counts for each country
countries = dplyr::count(hotdog_winners_df, Country, sort = FALSE)

# plot pie chart
fig <- plot_ly(countries, labels = ~Country, values = ~n)

fig <- fig %>%
  add_pie(hole = 0.6)

fig <- fig %>%
  layout(title = 'Donut Chart - R \n Percentage of Wins per Country 1980 - 2010',
    xaxis = list(showgrid = FALSE,
          zeroline = FALSE,
          showticklabels = FALSE),
    yaxis = list(showgrid = FALSE,
          zeroline = FALSE,
          showticklabels = FALSE),
    autosize = F,
    margin = list(l = 5, r = 5, b = 5, t = 120, pad = 10)
  )

export(fig, "images/donutchart-r.png")

```
~~~

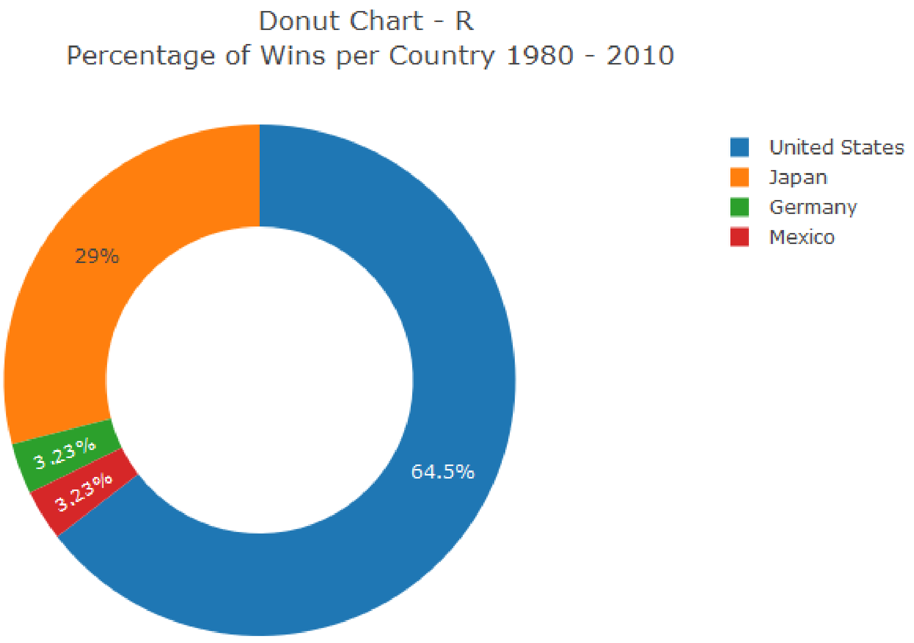
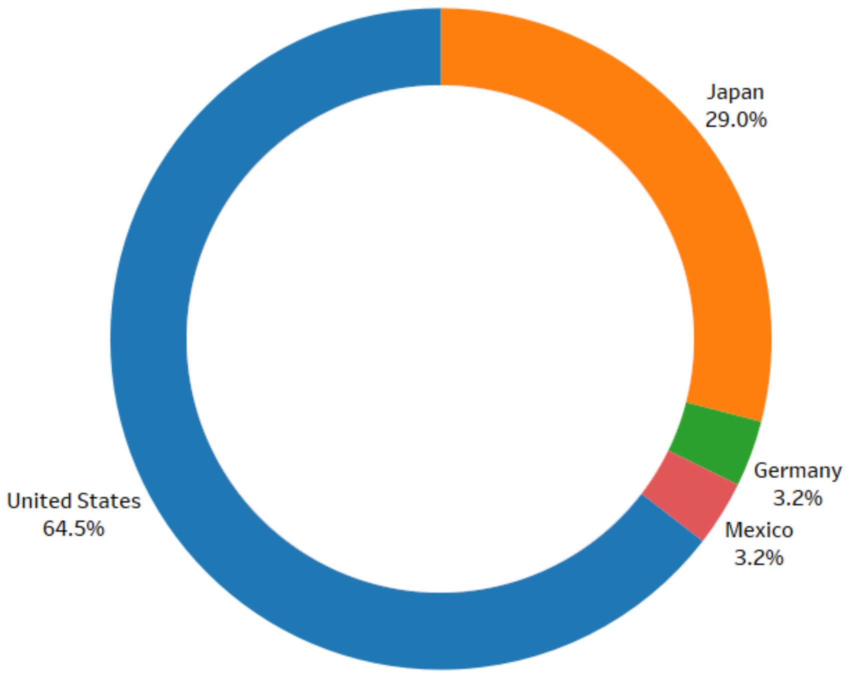


Tableau
(Please see Weeks1-2_Tableau.twb for code)



Donut Chart - Tableau
Percentage of Wins per Country 1980 - 2010