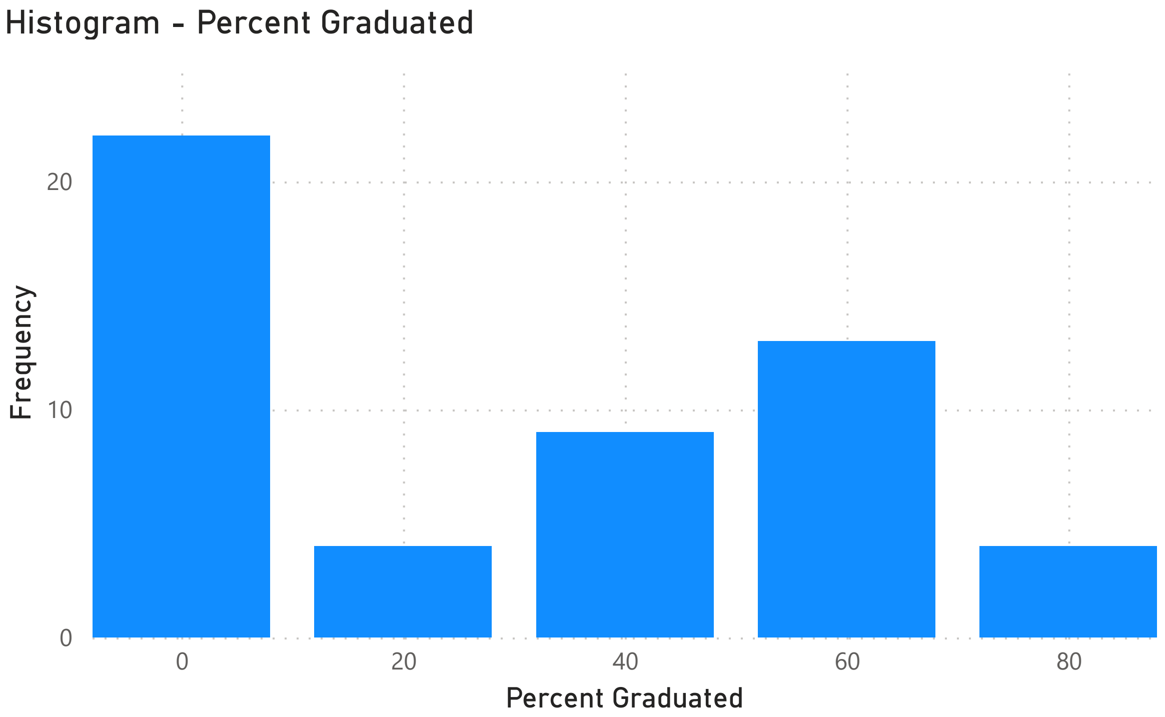
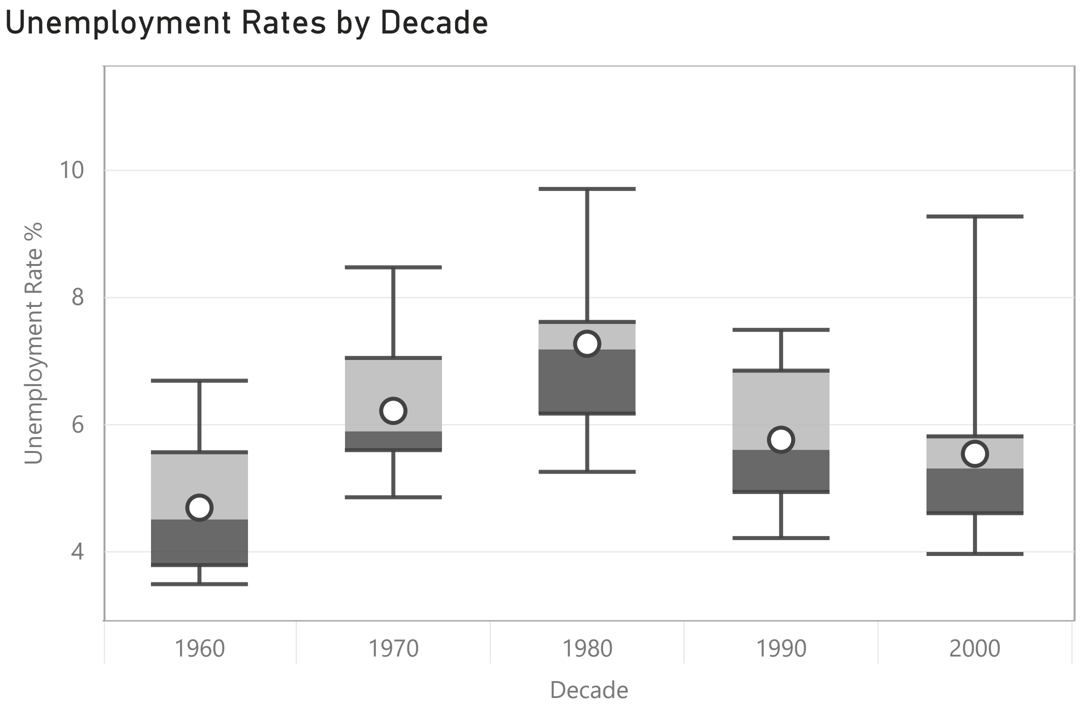
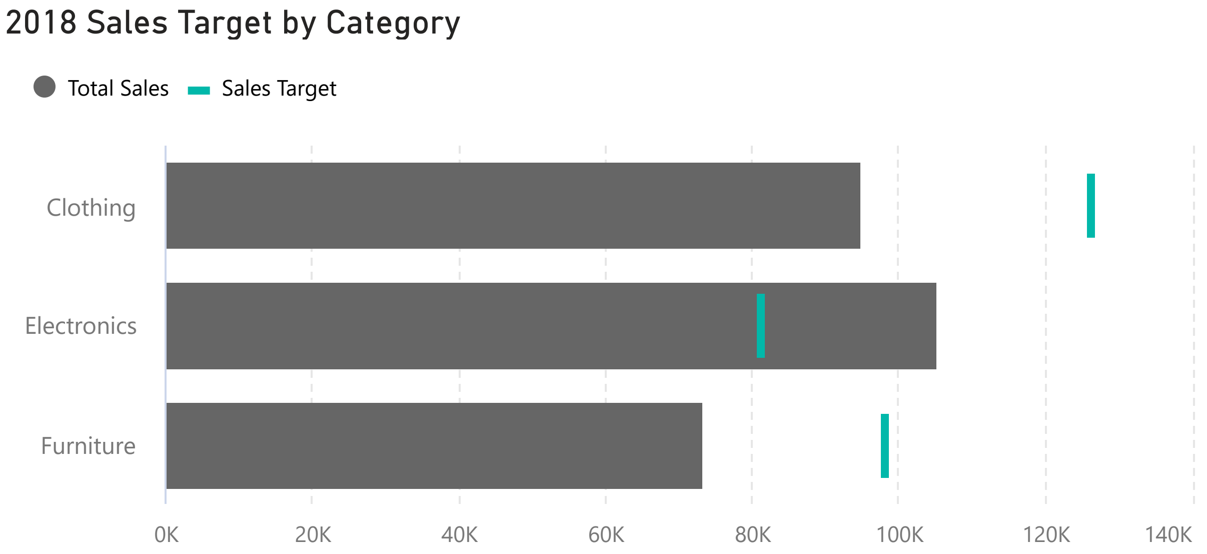
Power BI - Histogram



Power BI - Box Plot



Power BI - Bullet Chart



Campbell640Week11-12

November 6, 2023

[114]:

```
# Load libraries
import pandas as pd
import numpy as np
# import data visualization libraries
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.graph_objects as go
# Date
from datetime import datetime
```

[2]:

```
education = pd.read_csv('data sources/ex6-2/education.csv')
education.shape
```

```
[2]: (52, 7)
```

[9]:

```
unemployment = pd.read_csv('data sources/unemployement-rate-1948-2010.csv')
unemployment.shape
```

```
[9]: (746, 4)
```

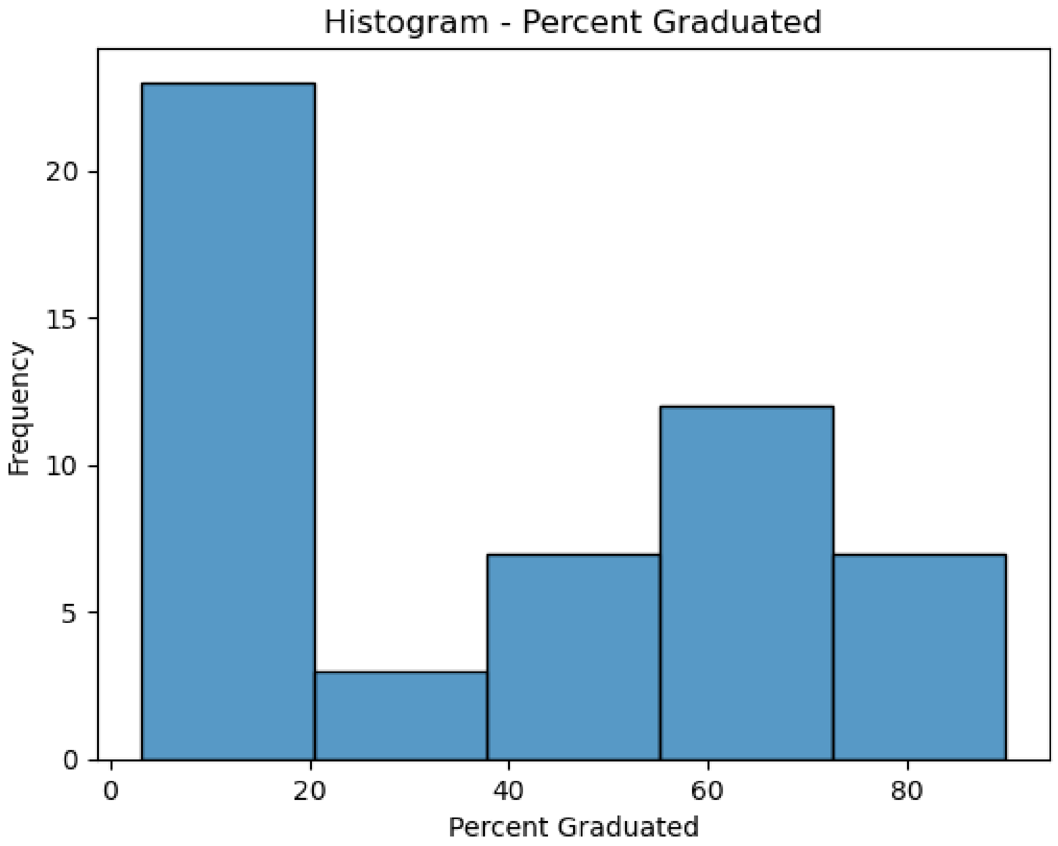
[75]:

```
orders = pd.read_csv('data sources/Order Details.csv')
target = pd.read_csv('data sources/Sales target.csv')
details = pd.read_csv('data sources/List of Orders.csv')
```

0.1 Python - Histogram

[6]:

```
sns.histplot(data=education, x="percent_graduates_sat", bins=5)
plt.title('Histogram - Percent Graduated')
plt.xlabel('Percent Graduated')
plt.ylabel('Frequency')
plt.show()
```

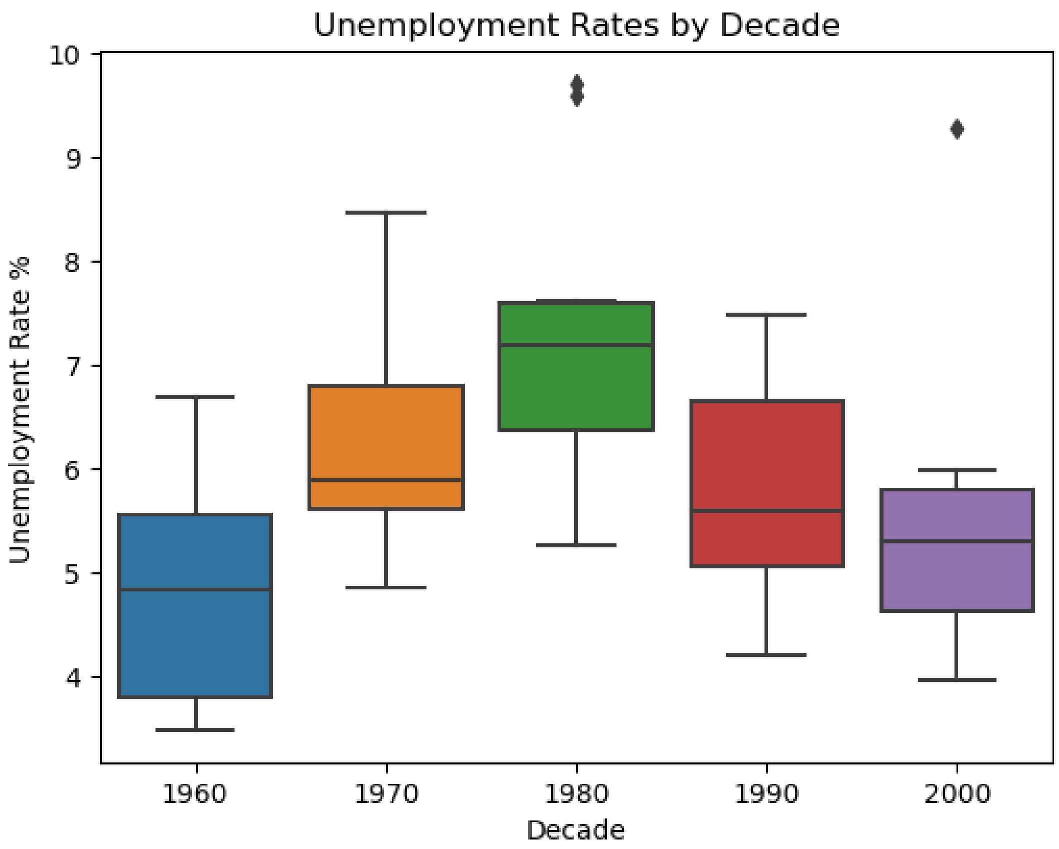


0.2 Python - Box Plot

[12]:
```
# Find decade for year
unemployment['decade'] = np.floor(unemployment['Year']/10)*10
# Convert to int
unemployment['decade'] = unemployment['decade'].astype(int)
```

[32]:
```
# Group by decade and filter
unemployment_decade = unemployment.groupby(['decade', 'Year'])['Value'].mean().
  ↪reset_index(name='value')
unemp_year = unemployment_decade[(unemployment_decade.Year >= 1960) &␣
  ↪(unemployment_decade.Year <= 2009)]
```

[43]:
```
sns.boxplot(data=unemp_year, x="decade", y="value")
plt.title('Unemployment Rates by Decade')
plt.xlabel('Decade')
plt.ylabel('Unemployment Rate %')
plt.show()
```



0.3 Python - Bullet Chart

```
[76]: # Convert string to dates
      target['Month of Order Date'] = pd.to_datetime(target['Month of Order Date'],␣
       ↪format='%b-%y')
      details['Order Date'] = pd.to_datetime(details['Order Date'], format='%d-%m-%Y')
```

```
[77]: # Filter to 2018 and group by category
      target = target[(target['Month of Order Date'] >= '2018-01-01') &␣
       ↪(target['Month of Order Date'] <= '2018-12-01')]
      target = target.groupby(['Category'])['Target'].sum().reset_index(name='target')
```

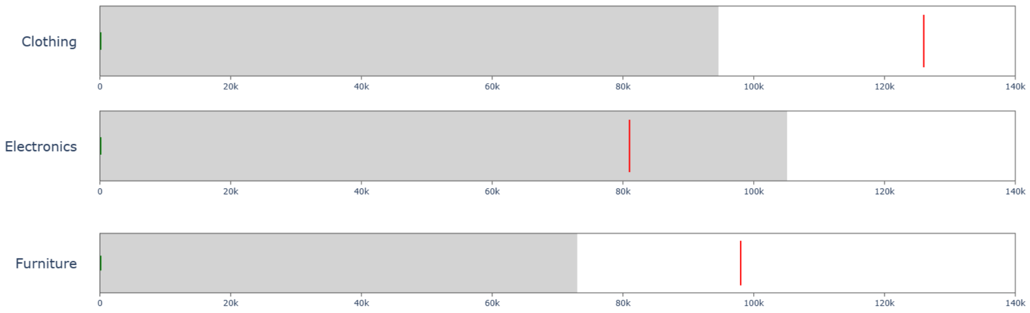
```
[80]: # Filter to 2018 and group by category
      details = details[(details['Order Date'] >= '2018-01-01') & (details['Order␣
       ↪Date'] <= '2018-12-31')]
```

```
[86]: # Group order amount by category
      details = details[(details['Order Date'] >= '2018-01-01') & (details['Order␣
       ↪Date'] <= '2018-12-31')]
```

```
orders = orders[orders['Order ID'].isin(details['Order ID'])]
orders = orders.groupby(['Category'])['Amount'].sum().reset_index(name='amount')
```

[142]:

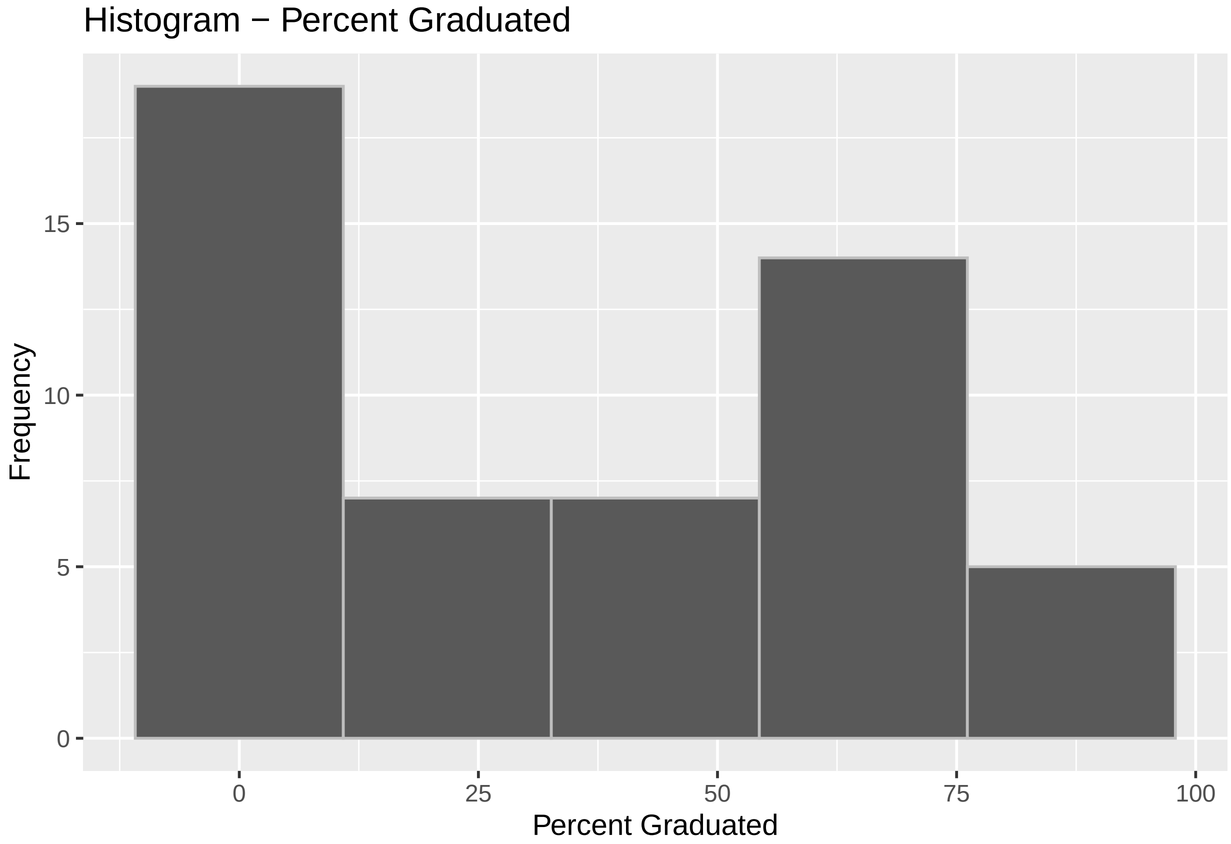
```
# Create bullet chart
fig = go.Figure()

fig.add_trace(go.Indicator(
        mode = "gauge", value = 220,
        domain = {'x': [0.1, 1], 'y': [0.08, 0.25]},
        title = {'text' : orders['Category'][2]},
        delta = {'reference': 200},
        gauge = {
            'shape': "bullet",
            'axis': {'range': [None, 140000]},
            'threshold': {
                'line': {'color': "red", 'width': 2},
                'thickness': 0.75,
                'value': target['target'][2]},
            'steps': [
                {'range': [0, orders['amount'][2]], 'color': "lightgray"}]}))

fig.add_trace(go.Indicator(
        mode = "gauge", value = 220,
        domain = {'x': [0.1, 1], 'y': [0.4, 0.6]},
        title = {'text' : orders['Category'][1]},
        delta = {'reference': 200},
        gauge = {
            'shape': "bullet",
            'axis': {'range': [None, 140000]},
            'threshold': {
                'line': {'color': "red", 'width': 2},
                'thickness': 0.75,
                'value': target['target'][1]},
            'steps': [
                {'range': [0, orders['amount'][1]], 'color': "lightgray"}]}))

fig.add_trace(go.Indicator(
        mode = "gauge", value = 220,
        domain = {'x': [0.1, 1], 'y': [0.7, 0.9]},
        title = {'text' : orders['Category'][0]},
        delta = {'reference': 200},
        gauge = {
            'shape': "bullet",
            'axis': {'range': [None, 140000]},
            'threshold': {
                'line': {'color': "red", 'width': 2},
                'thickness': 0.75,
                'value': target['target'][0]},
            'steps': [
                {'range': [0, orders['amount'][0]], 'color': "lightgray"}]}))

fig.update_layout(height = 1200)
fig.show()
```



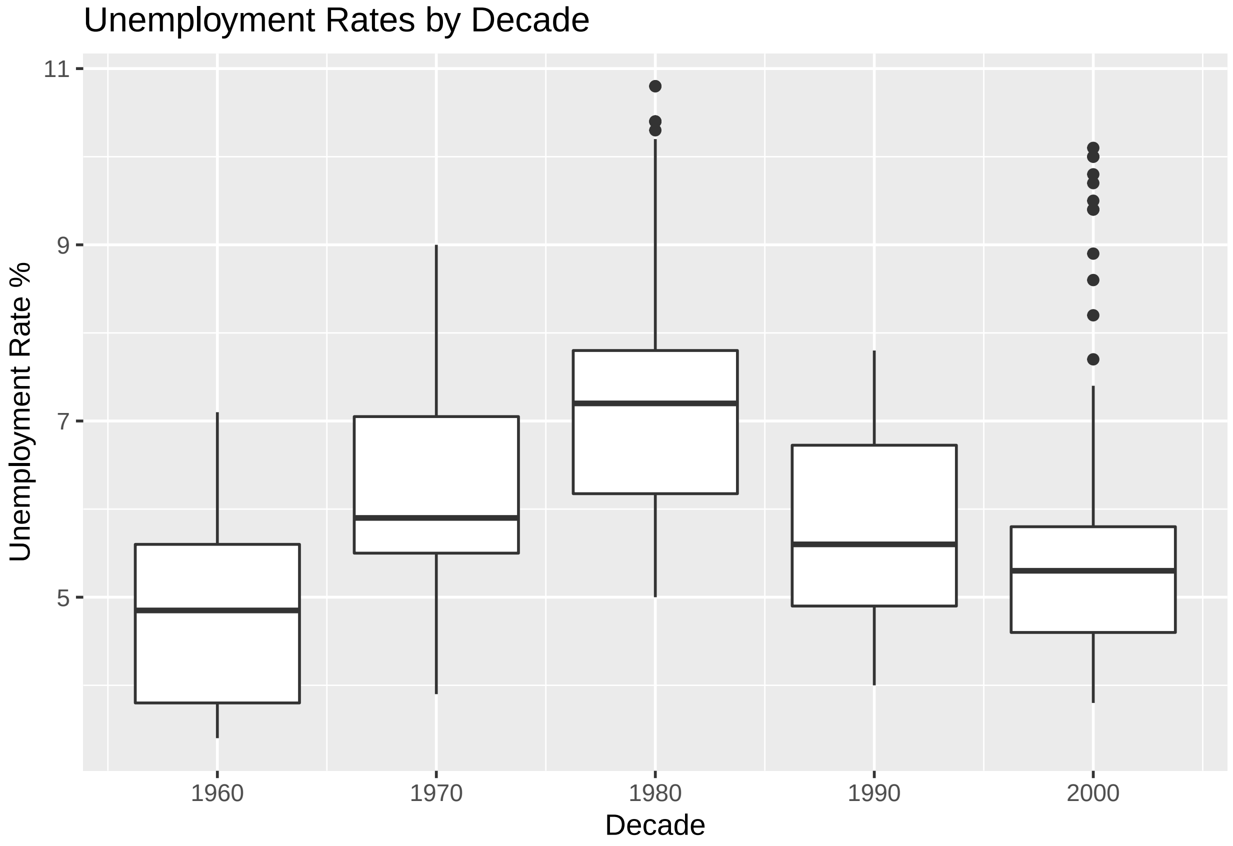
Campbell640Week11-12

2023-11-06

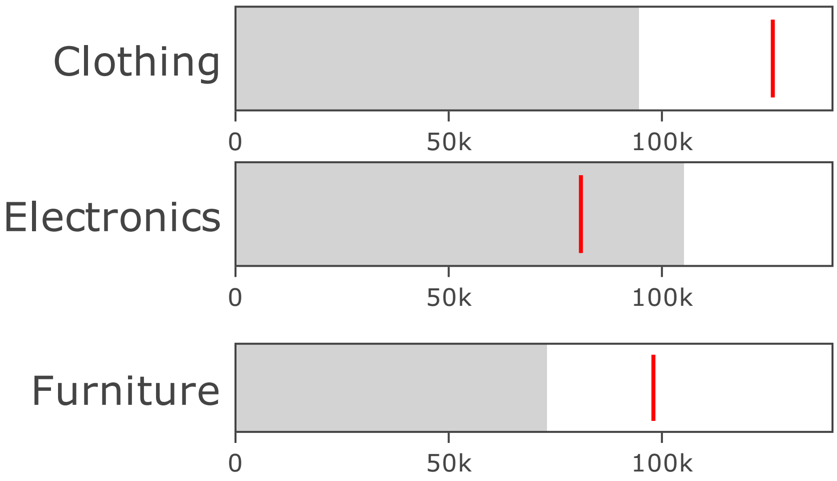
R - Histogram



R - Box Plot


Unemployment Rates by Decade
Unemployment Rate %
11
9
7
5
1960
1970
1980
1990
2000
Decade

R - Bullet Chart



## Code Repository

```
setwd("C:/Users/jcamp/Documents/DSC640/Assignments/data sources")

# Load libraries
library(ggplot2)
library(dplyr)
library(plotly)

# Load file
education <- read.csv("ex6-2/education.csv")
unemployment <- read.csv("unemployement-rate-1948-2010.csv")

# Kaggle: https://www.kaggle.com/datasets/benroshan/ecommerce-data?select=Sales+target.csv
orders <- read.csv("Order Details.csv")
target <- read.csv("Sales target.csv")
list <- read.csv("List of Orders.csv")
ggplot(education, aes(x=percent_graduates_sat)) + geom_histogram(bins=5, col="grey")+
  ggtitle("Histogram - Percent Graduated") +
  xlab("Percent Graduated") +
  ylab("Frequency")
# Summarize number of winners by country
rates_df = unemployment %>%
  mutate(decade = floor(Year/10)*10) %>%
  group_by(decade) %>%
  filter(decade >= 1960 & decade  < 2010)
```

```
ggplot(data=rates_df, mapping=aes(x=decade, y=Value, group=decade))+geom_boxplot()+
  ggtitle("Unemployment Rates by Decade") +
  xlab("Decade") +
  ylab("Unemployment Rate %")
target$Month.of.Order.Date = as.Date(paste("01-", target$Month.of.Order.Date, sep = ""), format = "%d-%

target <- target %>%
            filter(Month.of.Order.Date >= '2018-01-01' & Month.of.Order.Date  <= '2018-12-01') %>%
            group_by(Category) %>%
            summarise(Target = sum(Target))
list$Order.Date = as.Date(list$Order.Date, format = "%d-%m-%Y")

list <- list %>%
  filter(Order.Date>= '2018-01-01' & Order.Date <= '2018-12-31')


orders <- orders %>%
            filter(orders$Order.ID %in% list$Order.ID) %>%
            group_by(Category) %>%
            summarise(Amount = sum(Amount))
fig <- plot_ly()

  fig <- fig %>%
    add_trace(
    type = "indicator",
    mode = "gauge",
    domain = list(x = c(0.25, 1), y = c(0.08, 0.25)),
    title = list(text = orders$Category[3]),
    gauge = list(
      shape = "bullet",
      axis = list(range = list(NULL, 140000)),
      threshold = list(
        line = list(color = "red", width = 2),
        thickness = 0.75,
        value = target$Target[3]),
      steps = list(
      list(range = c(0, orders$Amount[3]), color = "lightgray"))),
  height = 150, width = 600)

  fig <- fig %>%
    add_trace(
    type = "indicator",
    mode = "gauge",
    domain = list(x = c(0.25, 1), y = c(0.4, 0.6)),
    title = list(text = orders$Category[2]),
    gauge = list(
      shape = "bullet",
      axis = list(range = list(NULL, 140000)),
      threshold = list(
        line = list(color = "red", width = 2),
        thickness = 0.75,
        value = target$Target[2]),
      steps = list(
      list(range = c(0, orders$Amount[2]), color = "lightgray"))),
  height = 150, width = 600)

  fig <- fig %>%
    add_trace(
    type = "indicator",
    mode = "gauge",
    domain = list(x = c(0.25, 1), y = c(0.7, 0.9)),
    title = list(text = orders$Category[1]),
    gauge = list(
      shape = "bullet",
      axis = list(range = list(NULL, 140000)),
      threshold = list(
        line = list(color = "red", width = 2),
        thickness = 0.75,
        value = target$Target[1]),
      steps = list(
      list(range = c(0, orders$Amount[1]), color = "lightgray"))),
  height = 150, width = 600)


fig
```