Assignment 4

1 Distribution of a continuous variable

```
stock_data <- read.csv("./constituents-financials_csv.csv",
                       stringsAsFactors = FALSE)
glimpse(stock_data)
```

```
## Rows: 505
## Columns: 14
## $ Symbol         <chr> "MMM", "AOS", "ABT", "ABBV", "ACN", "ATVI", "AYI", "ADB~
## $ Name           <chr> "3M Company", "A.O. Smith Corp", "Abbott Laboratories",~
## $ Sector         <chr> "Industrials", "Industrials", "Health Care", "Health Ca~
## $ Price          <dbl> 222.89, 60.24, 56.27, 108.48, 150.51, 65.83, 145.41, 18~
## $ Price.Earnings <dbl> 24.31, 27.76, 22.51, 19.41, 25.47, 31.80, 18.22, 52.31,~
## $ Dividend.Yield <dbl> 2.3328617, 1.1479592, 1.9089824, 2.4995599, 1.7144699, ~
## $ Earnings.Share <dbl> 7.92, 1.70, 0.26, 3.29, 5.44, 1.28, 7.43, 3.39, 6.19, 0~
## $ X52.Week.Low   <dbl> 259.770, 68.390, 64.600, 125.860, 162.600, 74.945, 225.~
## $ X52.Week.High  <dbl> 175.4900, 48.9250, 42.2800, 60.0500, 114.8200, 38.9300,~
## $ Market.Cap     <dbl> 138721055226, 10783419933, 102121042306, 181386347059, ~
## $ EBITDA         <dbl> 9048000000, 601000000, 5744000000, 10310000000, 5643228~
## $ Price.Sales    <dbl> 4.3902707, 3.5754826, 3.7404804, 6.2915710, 2.6041170, ~
## $ Price.Book     <dbl> 11.34, 6.35, 3.19, 26.14, 10.62, 5.16, 3.55, 11.06, 2.5~
## $ SEC.Filings    <chr> "http://www.sec.gov/cgi-bin/browse-edgar?action=getcomp~
```

We are interested in analyzing S&P500 data (https://github.com/datasets/s-and-p-500-companies-financials/) circa 2018. We'll want to examine various attributes by Sector, but they are too granular, so we'll create some sector rollups.

```
stock_data <- stock_data %>% mutate(
  SectorRollup =
    case_when(
      Sector %in% c("Consumer Discretionary", "Consumer Staples") ~ "Consumer",
      Sector %in% c("Industrials", "Materials") ~ "Industry",
      Sector %in% c("Telecommunication Services", "Information Technology") ~ "Tech",
      Sector %in% c("Energy", "Utilities") ~ "Power",
      TRUE ~ Sector
    )
) %>% drop_na()
```

1.1 Histogram

A commonly-used metric for assessing a stock's value is the "P/E" or Price to Earnings ratio ("Price.Earnings" in `stock_data`). A low P/E can indicate that a stock is undervalued, while a very high P/E can indicate that the market has high expectations for a stock's future earnings. If the company is losing money, then P/E can be negative.

Create a histogram which indicates the distribution of Price/Earnings for the bulk of observations in our dataset. Note that there are a good number of outlying points, so you will have to pick some appropriate limits for the graph, using either the `scale_x_continuous` function or the "limits" function. Alternatively, you could use dplyr to `filter` out points beyond the range you wish to display.

Additionally, make sure you choose an appropriate binwidth for your histogram! **Answer:**

```
filtered_data <- stock_data %>%
  filter(Price.Earnings >= 0 & Price.Earnings <= 100)

# Create the histogram
ggplot(filtered_data, aes(x = Price.Earnings)) +
  geom_histogram(binwidth = 5, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Price to Earnings", x = "Price to Earnings", y = "Frequency")
```

1.2 Dividend Yield by Sector Histogram

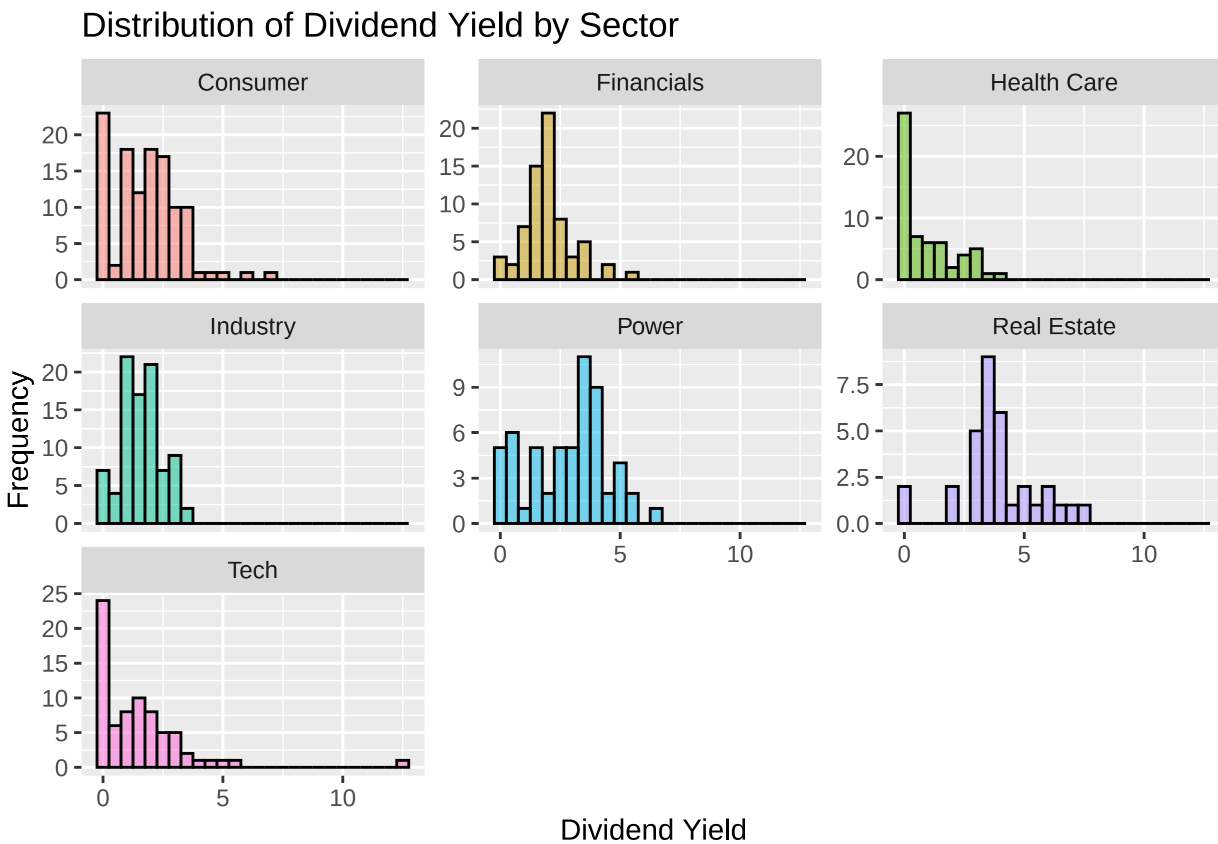
Next, we want to understand how dividend yields vary by sector. Dividend yield refers to the amount of money that the company pays to its shareholders per year (a "dividend"), divided by the stock price. In our dataset, it is in the `Dividend.Yield` variable.

Please create a small-multiples plot of histograms of Dividend yield, with one multiple for each Sector Rollup. Think about ways to make the resulting plot more aesthetically-pleasing, such as including color, changing the binwidth, or altering the number of columns in small-multiples plot. Implement them in your final plot and mention why you did so.

Lastly: which three sectors have the largest number of stocks which don't pay dividends? **Answer:**

```
ggplot(stock_data, aes(x = Dividend.Yield, fill = SectorRollup)) +
  geom_histogram(binwidth = 0.5, color = "black", alpha = 0.5) +
  facet_wrap(~SectorRollup, scales = "free_y", ncol = 3) +
```

```
  labs(title = "Distribution of Dividend Yield by Sector", x = "Dividend Yield", y = "Frequency") + the
```

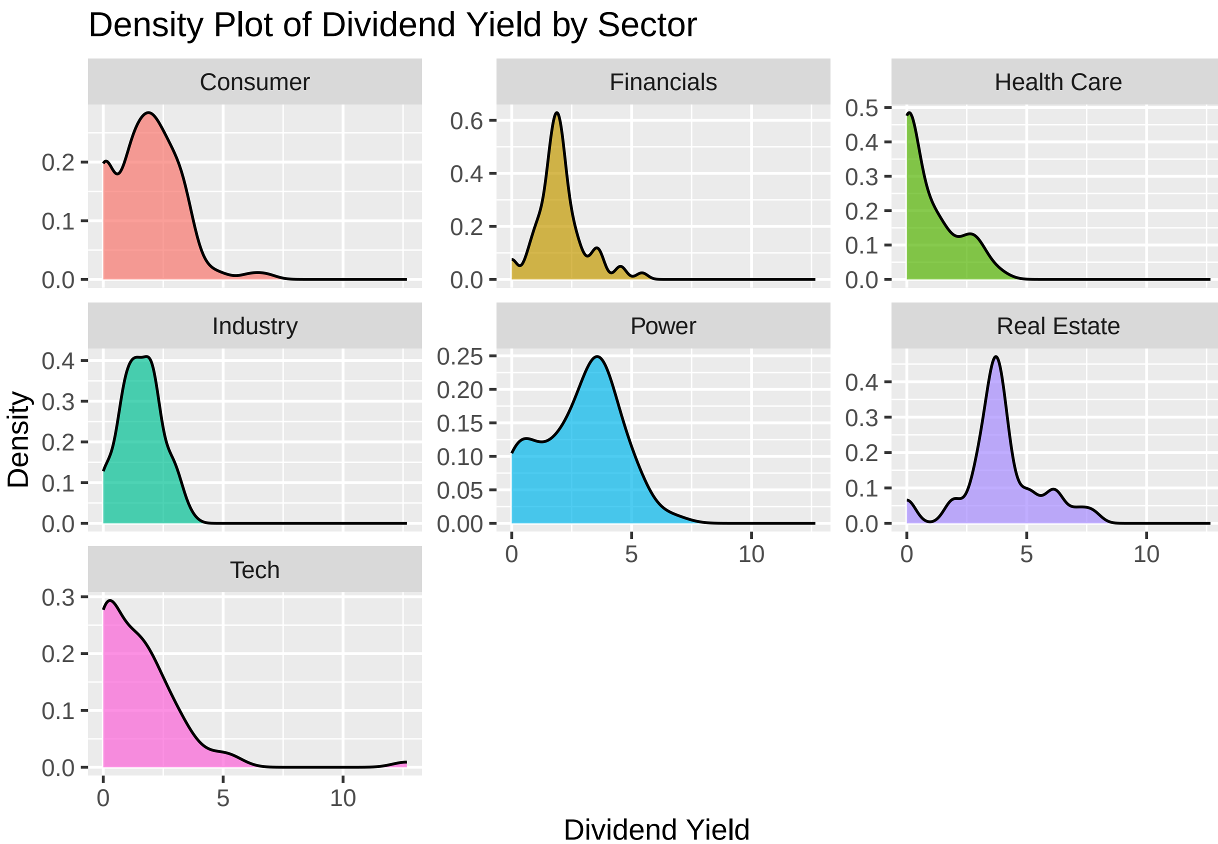


1.3 Kernel Density Estimates

The last plot has the flaw that each sector has differing numbers of stocks in it, and our histogram counts the number of stocks in each bin. Therefore, some large sectors like Tech are much more visually dominant than other sectors. A kernel density estimate would not have this problem.

Please create a density plot of dividend yield by sector similar to the histogram above. Make sure that the bandwidth of the kernel density estimate is appropriate, as we don't want to undersmooth or oversmooth! Examine the '?geom_density' help to understand how to tune the bandwidth parameter.

```
ggplot(stock_data, aes(x = Dividend.Yield, fill = SectorRollup)) +
  geom_density(alpha = 0.7, color = "black") +
  facet_wrap(~SectorRollup, scales = "free_y", ncol = 3) +
  labs(title = "Density Plot of Dividend Yield by Sector", x = "Dividend Yield", y = "Density") + theme
```



2 Transforming and Summarizing Before Plotting

Next, we're going to use the General Social Survey to understand the relationship between educational attainment (`degree`) and political views (`polviews`) in 2016.

```
glimpse(gss_sm)
```

```
## Rows: 2,867
## Columns: 32
## $ year        <dbl> 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016~
## $ id          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,~
## $ ballot      <labelled> 1, 2, 3, 1, 3, 2, 1, 3, 1, 3, 2, 1, 2, 3, 2, 3, 3, 2,~
## $ age         <dbl> 47, 61, 72, 43, 55, 53, 50, 23, 45, 71, 33, 86, 32, 60, 76~
## $ childs      <dbl> 3, 0, 2, 4, 2, 2, 2, 3, 3, 4, 5, 4, 3, 5, 7, 2, 6, 5, 0, 2~
## $ sibs        <labelled> 2, 3, 3, 3, 2, 2, 2, 6, 5, 1, 4, 4, 3, 6, 0, 1, 3, 8,~
## $ degree      <fct> Bachelor, High School, Bachelor, High School, Graduate, Ju~
## $ race        <fct> White, White, White, White, White, White, White, Other, Bl~
## $ sex         <fct> Male, Male, Male, Female, Female, Female, Male, Female, Ma~
## $ region      <fct> New England, New England, New England, New England, New En~
## $ income16    <fct> $170000 or over, $50000 to 59999, $75000 to $89999, $17000~
## $ relig       <fct> None, None, Catholic, Catholic, None, None, None, Catholic~
## $ marital     <fct> Married, Never Married, Married, Married, Married, Married~
## $ padeg       <fct> Graduate, Lt High School, High School, NA, Bachelor, NA, H~
## $ madeg       <fct> High School, High School, Lt High School, High School, Hig~
## $ partyid     <fct> "Independent", "Ind,near Dem", "Not Str Republican", "Not ~
## $ polviews    <fct> Moderate, Liberal, Conservative, Moderate, Slightly Libera~
## $ happy       <fct> Pretty Happy, Pretty Happy, Very Happy, Pretty Happy, Very~
```

```
## $ partners    <fct> NA, "1 Partner", "1 Partner", NA, "1 Partner", "1 Partner"~
## $ grass       <fct> NA, Legal, Not Legal, NA, Legal, Legal, NA, Not Legal, NA,~
## $ zodiac      <fct> Aquarius, Scorpio, Pisces, Cancer, Scorpio, Scorpio, Capri~
## $ pres12      <labelled> 3, 1, 2, 2, 1, 1, NA, NA, NA, 2, NA, NA, 1, 1, 2, 1, ~
## $ wtssall     <dbl> 0.9569935, 0.4784968, 0.9569935, 1.9139870, 1.4354903, 0.9~
## $ income_rc   <fct> Gt $170000, Gt $50000, Gt $75000, Gt $170000, Gt $170000, ~
## $ agegrp      <fct> Age 45-55, Age 55-65, Age 65+, Age 35-45, Age 45-55, Age 4~
## $ ageq        <fct> Age 34-49, Age 49-62, Age 62+, Age 34-49, Age 49-62, Age 4~
## $ siblings    <fct> 2, 3, 3, 3, 2, 2, 2, 6+, 5, 1, 4, 4, 3, 6+, 0, 1, 3, 6+, 2~
## $ kids        <fct> 3, 0, 2, 4+, 2, 2, 2, 3, 3, 4+, 4+, 4+, 3, 4+, 4+, 2, 4+, ~
## $ religion    <fct> None, None, Catholic, Catholic, None, None, None, Catholic~
## $ bigregion   <fct> Northeast, Northeast, Northeast, Northeast, Northeast, Nor~
## $ partners_rc <fct> NA, 1, 1, NA, 1, 1, NA, 1, NA, 3, 1, NA, 1, NA, 0, 1, 0, N~
## $ obama       <dbl> 0, 1, 0, 0, 1, 1, NA, NA, NA, 0, NA, NA, 1, 1, 0, 1, 0, 1,~
```

```
gss_sm <- gss_sm %>% mutate(degree =as.character(degree))
```

2.1 Filter and Mutate

If we examine the categories in `degree`, we see they make a distinction between “Junior College” and “Bachelor”. Create a roll up variable `degreeRollup` which combines the two using `mutate` and `case_when`.

Additionally, we see that there are some missing observations for both `degree` and `polviews`. Filter them out using the `is.na` function.

Answer:

```
gss_sm %>% mutate(
  degreeRollup  = case_when(
    degree %in% c("Junior College", "Bachelor") ~ "College",
    TRUE ~ degree
  )
)
```

```
## # A tibble: 2,867 x 33
##     year    id ballot       age childs sibs   degree race  sex   region income16
##    <dbl> <dbl> <labelled> <dbl>  <dbl> <labe> <chr>  <fct> <fct> <fct>  <fct>
##  1  2016     1 1             47      3 2      Bache~ White Male  New E~ $170000~
##  2  2016     2 2             61      0 3      High ~ White Male  New E~ $50000 ~
##  3  2016     3 3             72      2 3      Bache~ White Male  New E~ $75000 ~
##  4  2016     4 1             43      4 3      High ~ White Fema~ New E~ $170000~
##  5  2016     5 3             55      2 2      Gradu~ White Fema~ New E~ $170000~
##  6  2016     6 2             53      2 2      Junio~ White Fema~ New E~ $60000 ~
##  7  2016     7 1             50      2 2      High ~ White Male  New E~ $170000~
##  8  2016     8 3             23      3 6      High ~ Other Fema~ Middl~ $30000 ~
##  9  2016     9 1             45      3 5      High ~ Black Male  Middl~ $60000 ~
## 10  2016    10 3             71      4 1      Junio~ White Male  Middl~ $60000 ~
## # i 2,857 more rows
## # i 22 more variables: relig <fct>, marital <fct>, padeg <fct>, madeg <fct>,
## #   partyid <fct>, polviews <fct>, happy <fct>, partners <fct>, grass <fct>,
## #   zodiac <fct>, pres12 <labelled>, wtssall <dbl>, income_rc <fct>,
## #   agegrp <fct>, ageq <fct>, siblings <fct>, kids <fct>, religion <fct>,
## #   bigregion <fct>, partners_rc <fct>, obama <dbl>, degreeRollup <chr>
```

2.2 Grouping and Summarizing

Next, create a new dataset group the transformed `gss_sm` dataset by degreeRollup and polviews , summarize the number of observations in each group, and use mutate to compute the percentage of within each degree2 level that agrees with each of the political views.

Note that if you pass in two variables to `group_by`, the first variable dictates the first level in the grouping hierachy, and the second variable is then nested within the first.

Answer:

```r
grouped_data <- gss_sm %>%
  group_by(degree, polviews) %>%
  summarize(count = n()) %>%
  mutate(percentage = count / sum(count) * 100)
```

```
## `summarise()` has grouped output by 'degree'. You can override using the
## `.groups` argument.
```

```r
grouped_data
```

```
## # A tibble: 43 x 4
## # Groups:   degree [6]
##    degree   polviews               count percentage
##    <chr>    <fct>                  <int>      <dbl>
##  1 Bachelor Extremely Liberal         25      4.66
##  2 Bachelor Liberal                   94     17.5
##  3 Bachelor Slightly Liberal          68     12.7
##  4 Bachelor Moderate                 146     27.2
##  5 Bachelor Slightly Conservative     84     15.7
##  6 Bachelor Conservative              95     17.7
##  7 Bachelor Extremely Conservative    21      3.92
##  8 Bachelor <NA>                       3      0.560
##  9 Graduate Extremely Liberal         29      9.12
## 10 Graduate Liberal                   78     24.5
## # i 33 more rows
```

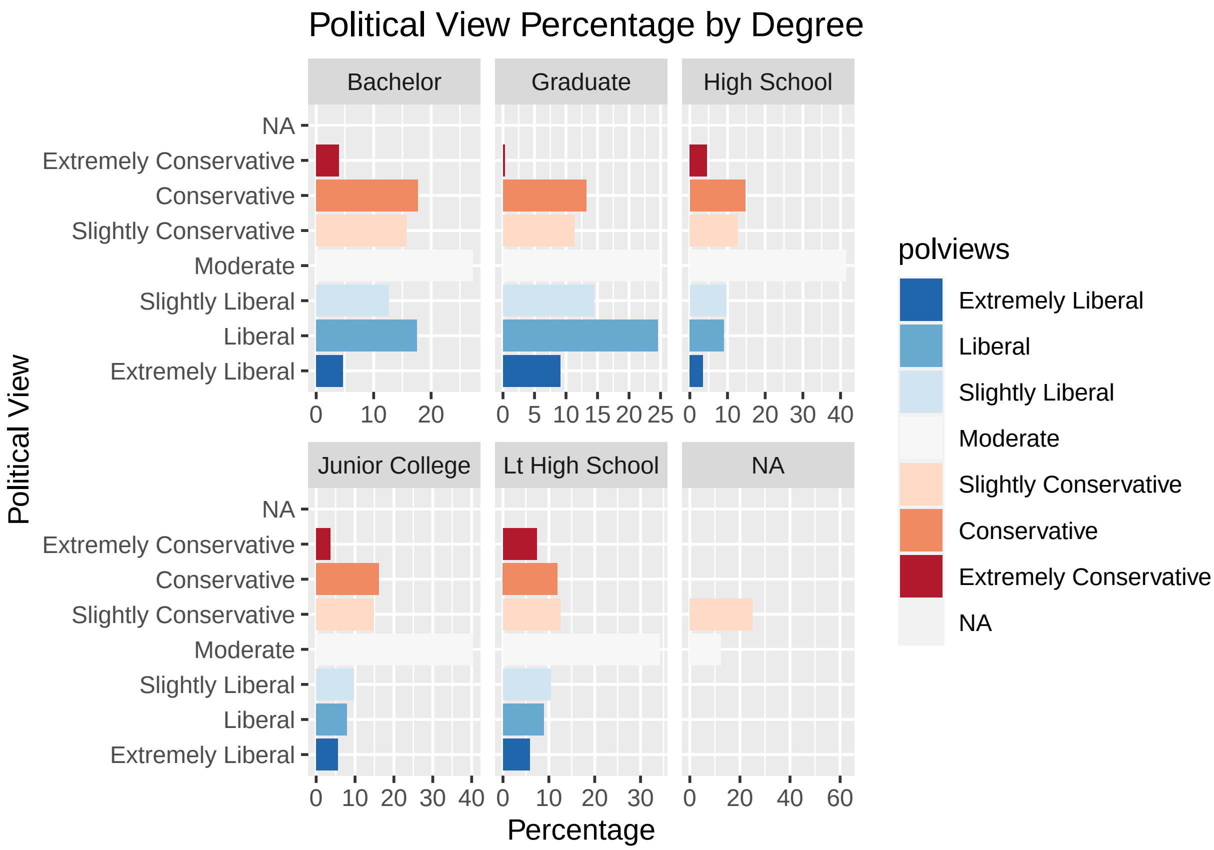
2.3 Plot a summary table

Last, create a faceted bar chart of this summary table using `geom_col`. We want to plot political view percentage, split up by degree type. Recall that in`geom_col`, we map our categorical variable to the "x" aesthetic, and the height of the bar is controlled by the "y" aesthetic.

Lastly, to make the bar chart aesthetically pleasing, make sure to prevent the labels from overlapping! This can be done by flipping the coordinate system via `coord_flip`. If you want, you can color the bars according to the standard Liberal-Conservative color scheme via adding `scale_fill_brewer(type="div", palette = "RdBu", direction=-1)` to your plot to create a diverging Red-Blue color palette.

Answer:

```r
plot <- ggplot(grouped_data, aes(x = polviews, y = percentage, fill = polviews)) +
  geom_col() + facet_wrap(~degree, scales = "free_x") +  coord_flip() +
  scale_fill_brewer(type = "div", palette = "RdBu", direction = -1) +
  labs(x = "Political View", y = "Percentage", title = "Political View Percentage by Degree")
plot
```


Political View Percentage by Degree
Bachelor
Graduate
High School
Junior College
Lt High School
NA
NA
Extremely Conservative
Conservative
Slightly Conservative
Moderate
Slightly Liberal
Liberal
Extremely Liberal
Political View
Percentage
polviews
Extremely Liberal
Liberal
Slightly Liberal
Moderate
Slightly Conservative
Conservative
Extremely Conservative
NA