R Notebook

```
library(tidyverse)
library(ggstatsplot)
```

Principles of Data Visualization and Introduction to ggplot2

I have provided you with data about the 5,000 fastest growing companies in the US, as compiled by Inc. magazine. lets read this in:

```
# import data from github
inc <- read.csv("https://raw.githubusercontent.com/charleyferrari/CUNY_DATA_608/master/module1/Data/inc5
```

And lets preview this data:

```
# preview data
head(inc)
```

```
##   Rank                           Name Growth_Rate   Revenue
## 1    1                           Fuhu      421.48 1.179e+08
## 2    2        FederalConference.com      248.31 4.960e+07
## 3    3                The HCI Group      245.45 2.550e+07
## 4    4                      Bridger      233.08 1.900e+09
## 5    5                       DataXu      213.37 8.700e+07
## 6    6 MileStone Community Builders      179.38 4.570e+07
##                       Industry Employees         City State
## 1 Consumer Products & Services       104   El Segundo    CA
## 2          Government Services        51     Dumfries    VA
## 3                       Health       132 Jacksonville    FL
## 4                       Energy        50      Addison    TX
## 5      Advertising & Marketing       220       Boston    MA
## 6                  Real Estate        63       Austin    TX
```

```
# view data summary
summary(inc)
```

```
##       Rank          Name             Growth_Rate         Revenue
##  Min.   :   1   Length:5001        Min.   :  0.340   Min.   :2.000e+06
##  1st Qu.:1252   Class :character   1st Qu.:  0.770   1st Qu.:5.100e+06
##  Median :2502   Mode  :character   Median :  1.420   Median :1.090e+07
##  Mean   :2502                      Mean   :  4.612   Mean   :4.822e+07
##  3rd Qu.:3751                      3rd Qu.:  3.290   3rd Qu.:2.860e+07
##  Max.   :5000                      Max.   :421.480   Max.   :1.010e+10
##
##    Industry           Employees           City              State
##  Length:5001        Min.   :    1.0   Length:5001        Length:5001
```

```
##  Class :character   1st Qu.:   25.0   Class :character   Class :character
##  Mode  :character   Median :   53.0   Mode  :character   Mode  :character
##                     Mean   :  232.7
##                     3rd Qu.:  132.0
##                     Max.   :66803.0
##                     NA's   :12
```

Think a bit on what these summaries mean. Use the space below to add some more relevant non-visual exploratory information you think helps you understand this data:

Ok, for me its easier to write out and define what they mean:

- `Rank` is the rank of the company in regard to `Growth Rate` (ordinal)
- `Name` is the name of the company
- `Growth Rate` is the rate at which the company has grown over a period of time (continuous).
- `Revenue` is the amount of money each company has made (continuous)
- `Industry` is a categorical variable stating the industry type of the company
- `Employees` are the number of employees (continuous)
- `City` is the city where the company is located
- `State` is the state where the company is located

So for any object data types the `summary()` function is simply counting the records and defining the class type as `character`. For numeric data types we are getting some summary statistics including: - Min - 1st Quartile - Median - Mean - 3rd Quartile - Max - How many NA's

Some grouping by categorical data would help me better understand where companies that have high `Growth Rate` tend to be located.

```
# Insert your code here, create more chunks as necessary
(top_5_states <- inc %>%
  group_by(State) %>%
  tally(sort = T) %>%
  head())
```

```
## # A tibble: 6 x 2
##   State     n
##   <chr> <int>
## 1 CA      701
## 2 TX      387
## 3 NY      311
## 4 VA      283
## 5 FL      282
## 6 IL      273
```

```
# top 5 states with most companies and average growth rate
inc %>%
  group_by(State) %>%
  summarise(n = n(), mean_growth_rate = round(mean(Growth_Rate),2)) %>%
  arrange(desc(n)) %>%
  head() %>%
  arrange(desc(mean_growth_rate))
```

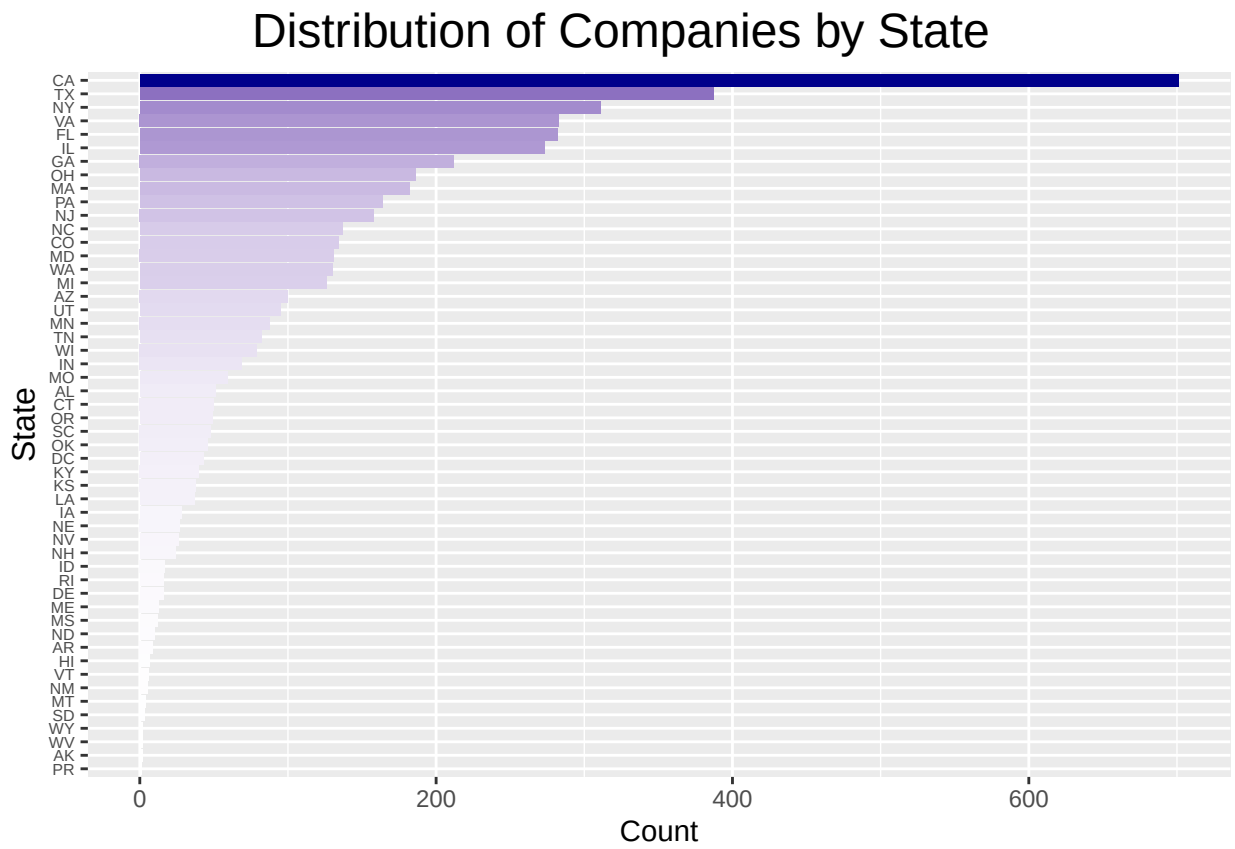
```
## # A tibble: 6 x 3
```

```
##   State     n mean_growth_rate
##   <chr> <int>            <dbl>
## 1 TX      387             6.02
## 2 CA      701             5.9
## 3 FL      282             5.85
## 4 VA      283             4.88
## 5 NY      311             4.37
## 6 IL      273             3.74
```

Question 1

Create a graph that shows the distribution of companies in the dataset by State (ie how many are in each state). There are a lot of States, so consider which axis you should use. This visualization is ultimately going to be consumed on a 'portrait' oriented screen (ie taller than wide), which should further guide your layout choices.

```
# Answer Question 1 here
inc %>%
  group_by(State) %>%
  summarise(count = n()) %>%
  ggplot() +
    geom_bar(aes(count, reorder(State, count), fill = count),
             stat = "identity") +
    scale_fill_gradient(low = "white", high = "darkblue") +
    labs(x="Count", y="State",
         title = "Distribution of Companies by State") +
    theme(plot.title.position = "plot",
          plot.title = element_text(hjust = 0.5,
                                    size = 18),
          axis.text.y = element_text(size = 6),
          axis.title.y = element_text(size = 12),
          legend.position = "none")
```



Question 2

Lets dig in on the state with the 3rd most companies in the data set. Imagine you work for the state and are interested in how many people are employed by companies in different industries. Create a plot that shows the average and/or median employment by industry for companies in this state (only use cases with full data, use R's `complete.cases()` function.) In addition to this, your graph should show how variable the ranges are, and you should deal with outliers.

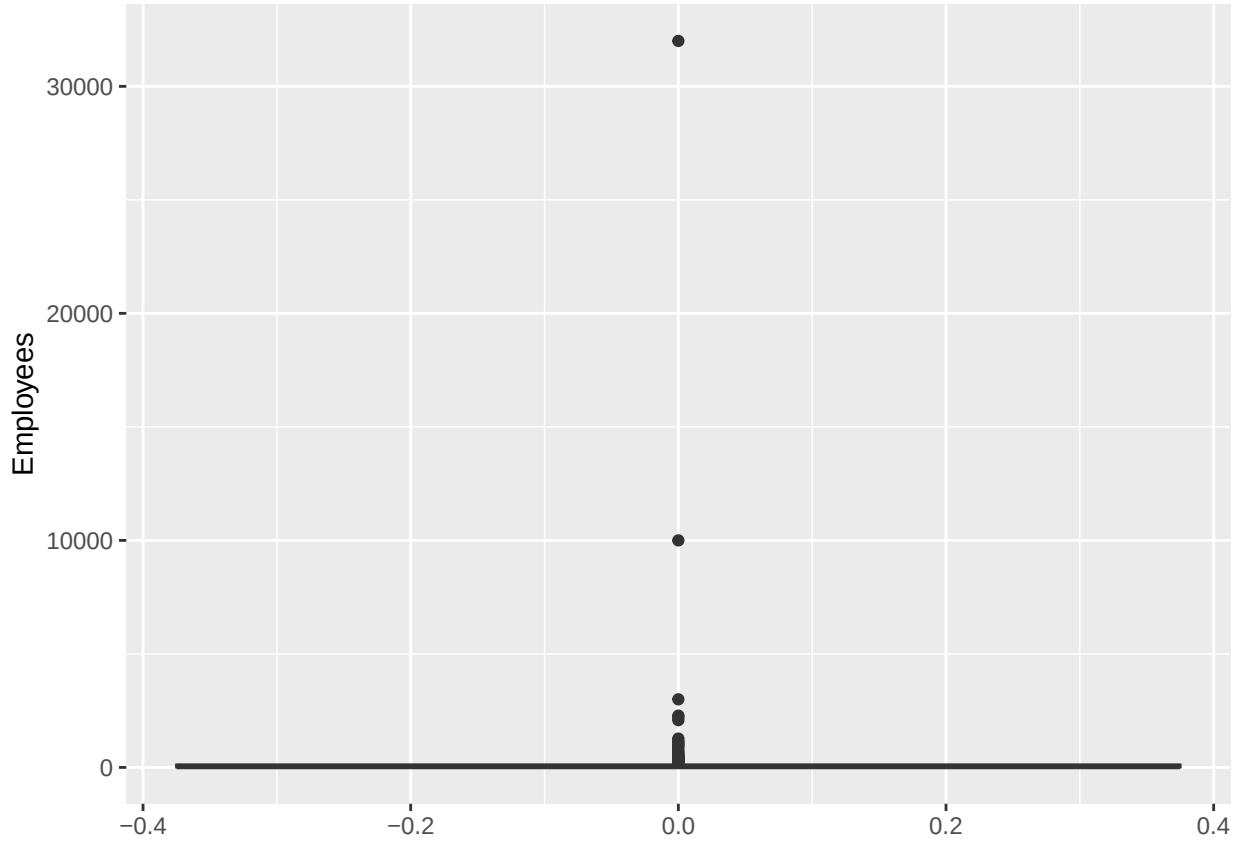
```
# Answer Question 2 here
# isloate state with 3rd most highest companies (NY)
inc_ny <- inc[inc$State == "NY",]

# look at summary for NY employees column
summary(inc_ny$Employees)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    21.0    45.0   271.3   105.5 32000.0
```

Let's visualize `Employees` in a boxplot

```
inc_ny %>%
  ggplot() +
    geom_boxplot(aes(Employees)) +
    coord_flip()
```



Looks like we have some extreme variation, wonder what industries this is in.

Let's group by industry and create min, max, mean and median columns sorted decreasing by max.

First let's create custom function.

```
# create group by function for particular column
# aggregates by another and sorts descending by another (default max)
groupby_column <- function(df, col_grp, agg_col, sort_col=max_grp) {

  # quote columns
  col_grp <- dplyr::enquo(col_grp)
  agg_col <- dplyr::enquo(agg_col)
  sort_col <- dplyr::enquo(sort_col)

  # groupby and add aggreations
  result <- df %>%
    dplyr::group_by(!!col_grp) %>%
    dplyr::summarise(n_grp = n(),
              min_grp = min(!!agg_col),
              max_grp = max(!!agg_col),
              mean_grp = round(mean(!!agg_col),2),
              med_grp = round(median(!!agg_col),2)) %>%
    arrange(desc(!!sort_col))

  return(result)
}
```

Let's apply the custom function.

```
# apply function to dataset groupby Industry, aggregated by Employees
(inc_ny_grouped <- groupby_column(df = inc_ny,
                                  col_grp = Industry,
                                  agg_col = Employees))
```

```
## # A tibble: 25 x 6
##    Industry                     n_grp min_grp max_grp mean_grp med_grp
##    <chr>                        <int>   <int>   <int>    <dbl>   <dbl>
##  1 Business Products & Services    26       4   32000   1492.     70.5
##  2 Consumer Products & Services    17       5   10000    626.     25
##  3 IT Services                     43       8    3000    204.     54
##  4 Travel & Hospitality             7       6    2280    548.     61
##  5 Human Resources                 11       7    2081    438.     56
##  6 Software                        13      15    1271    246.     80
##  7 Media                           11       4     602    108      45
##  8 Financial Services              13      14     483    144.     81
##  9 Security                         4      25     450    135      32.5
## 10 Food & Beverage                  9       5     383     76.4    41
## # ... with 15 more rows
```

Extreme outliers in `Business Products & Services` and `Consumer Products & Services` will skew any visualizations. Try as is first.

Try with `geom_crossbar()`

First create custom function.

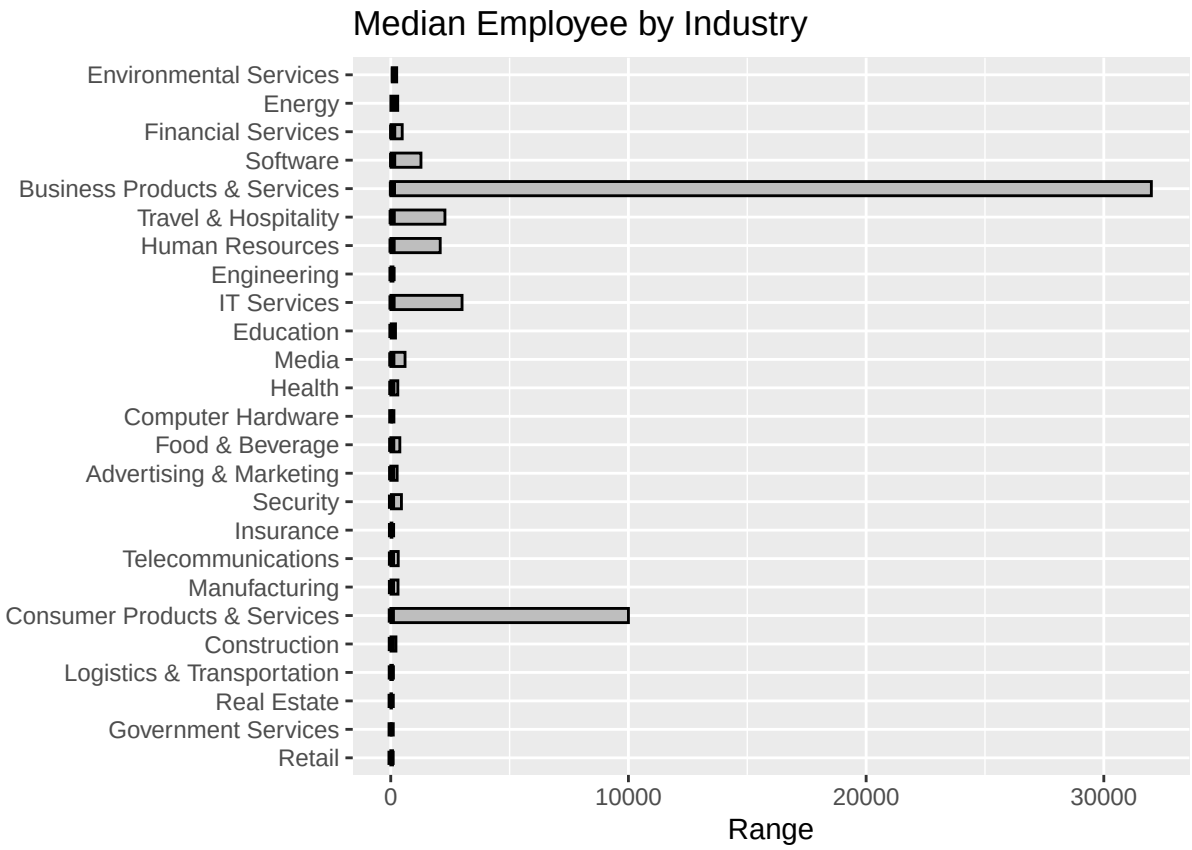
```
# create custom function to plot with geom_crossbar
# enter df, x, y, sort order for x, ymin, ymax and fill (default grey)
crossbar_plot <- function(df, x, y, sort_x, ymin, ymax,
                          fill ="grey", xlab="", ylab="Range",
                          title = "", width = 0.5) {
  # quote columns
  x <- dplyr::enquo(x)
  y <- dplyr::enquo(y)
  sort_x <- dplyr::enquo(sort_x)
  ymin <- dplyr::enquo(ymin)
  ymax <- dplyr::enquo(ymax)

  # plot
  result <- df %>%
    ggplot(aes(x = reorder(as.factor(!!x), !!sort_x), y = !!y)) +
      geom_crossbar(aes(ymin = !!ymin, ymax = !!ymax), width = width,
                fill = fill) + coord_flip() +
      labs(x = xlab, y=ylab, title = title)

  return(result)
}
```

Let's try out custom function.

```
# plot by Median Employee by Industry
crossbar_plot(inc_ny_grouped, Industry, med_grp, med_grp, min_grp, max_grp,
              title = "Median Employee by Industry")
```



Doesn't deal with outliers well, impossible to tell what actual median is. This is NOT a good visualation. Try Log transform and re-plot.

Transformation

```
# log10 transform Employee column
inc_ny_emp_log <- inc_ny %>%
  mutate(emp_log = round(log10(Employees)),2) %>%
  select(Rank, Name, Employees, emp_log, everything()) %>%
  arrange(desc(emp_log))

# preview
head(inc_ny_emp_log)
```

```
##   Rank                       Name Employees emp_log Growth_Rate   Revenue
## 1 4577 Sutherland Global Services     32000       5        0.48 5.976e+08
## 2 4936                      Coty     10000       4        0.36 4.600e+09
## 3 1069       Systems Made Simple       382       3        3.94 1.671e+08
## 4 1499      Sterling Infosystems      2081       3        2.66 2.149e+08
```

```
## 5 1640                   BlueWolf       500       3        2.38 9.040e+07
## 6 2218               Globo Mobile       320       3        1.67 4.500e+06
##                         Industry      City State 2
## 1 Business Products & Services Pittsford    NY 2
## 2 Consumer Products & Services  New York    NY 2
## 3                  IT Services  Syracuse    NY 2
## 4              Human Resources  New York    NY 2
## 5                  IT Services  New York    NY 2
## 6                     Software  New York    NY 2
```

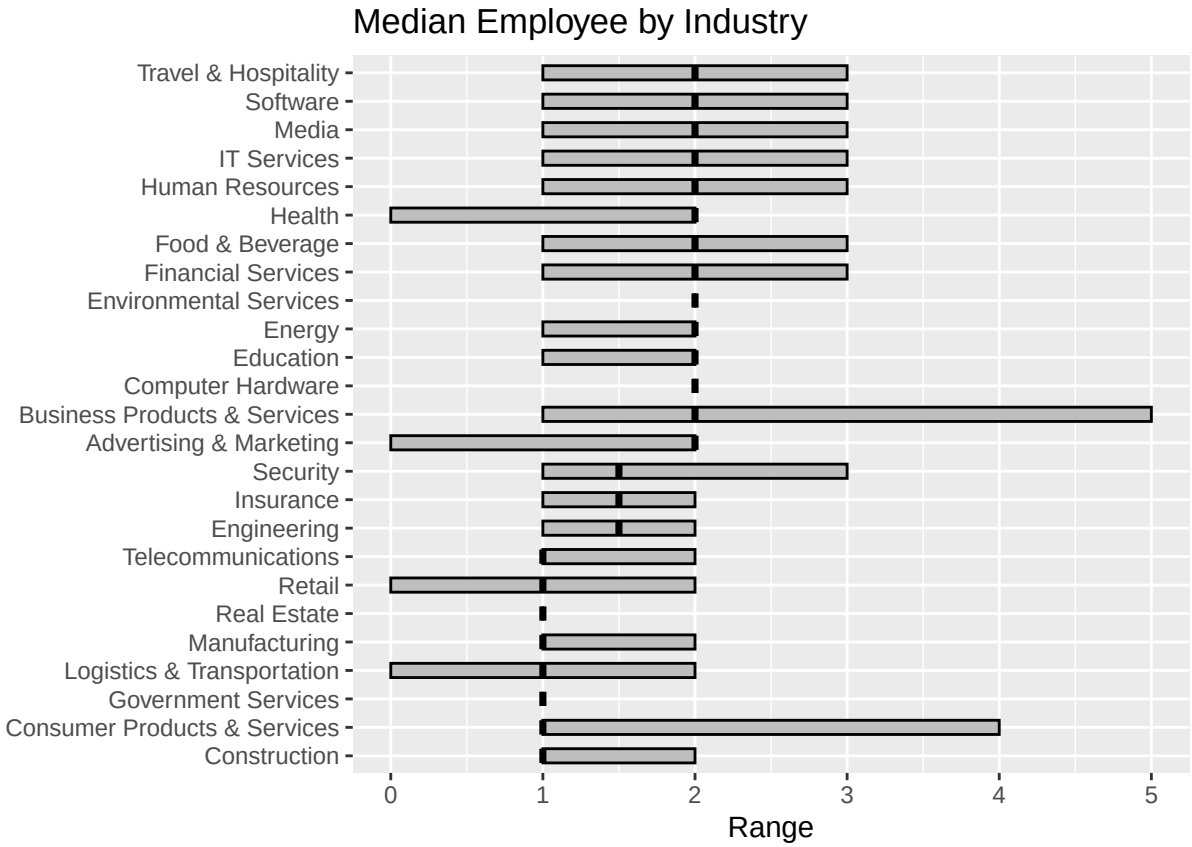
Apply customer `groupby_column()` function

```
(inc_ny_emp_log_grp <- groupby_column(inc_ny_emp_log, Industry, emp_log))
```

```
## # A tibble: 25 x 6
##    Industry                     n_grp min_grp max_grp mean_grp med_grp
##    <chr>                        <int>   <dbl>   <dbl>    <dbl>   <dbl>
##  1 Business Products & Services    26       1       5     2.08     2
##  2 Consumer Products & Services    17       1       4     1.59     1
##  3 Financial Services              13       1       3     2.08     2
##  4 Food & Beverage                  9       1       3     1.67     2
##  5 Human Resources                 11       1       3     2        2
##  6 IT Services                     43       1       3     1.84     2
##  7 Media                           11       1       3     1.64     2
##  8 Security                         4       1       3     1.75     1.5
##  9 Software                        13       1       3     1.92     2
## 10 Travel & Hospitality             7       1       3     2.14     2
## # ... with 15 more rows
```

Let's re-plot with custom `crossbar_plot()` function.

```
# plot by Median Employee by Industry
crossbar_plot(inc_ny_emp_log_grp,
              Industry, med_grp, med_grp, min_grp, max_grp,
              title = "Median Employee by Industry")
```



This is too generalized, also difficult to communicate range and median in log format to stakeholders.

Let's try dropping outliers

Drop Outliers

Let's classify outliers in the typical manner +- 1.5 * interquartile range

```
# find quartiles
quartiles <- stats::quantile(inc_ny$Employees, probs=c(.01,.99), na.rm=F)

# get IQR
IQR <- stats::IQR(inc_ny$Employees)

# define lower and upper outliers
lower <- quartiles[1] - 1.5*IQR
upper <- quartiles[2] + 1.5*IQR

# subset data set removing outliers
inc_ny_no_outlier <-
  subset(inc_ny, inc_ny$Employees > lower & inc_ny$Employees < upper)

# dimensions after
dim_outlier <-as.numeric(dim(inc_ny)[1])
dim_no_outlier <- as.numeric(dim(inc_ny_no_outlier)[1])
```

```
# compare records of dataset
print(paste0("Rows prior to outlier reduction ", dim_outlier))
```

```
## [1] "Rows prior to outlier reduction 311"
```

```
print(paste0("Rows after 1% outlier reduction ",dim_no_outlier))
```

```
## [1] "Rows after 1% outlier reduction 308"
```

```
# preview
head(inc_ny_no_outlier)
```

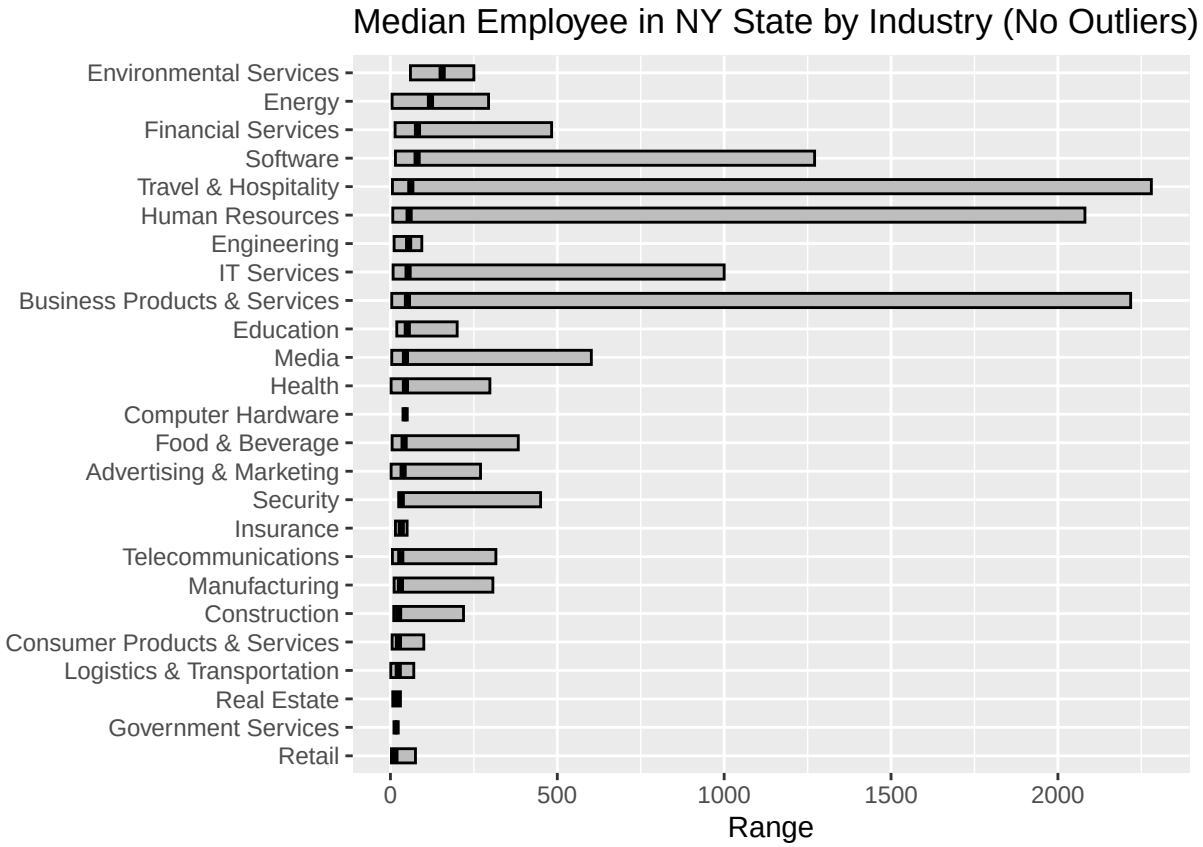
```
##    Rank                      Name Growth_Rate  Revenue
## 26   26              BeenVerified       84.43 13700000
## 30   30                  Sailthru       73.22  8100000
## 37   37              YellowHammer       67.40 18000000
## 38   38                 Conductor       67.02  7100000
## 48   48 Cinium Financial Services       53.65  5900000
## 70   70                  33Across       44.99 27900000
##                         Industry Employees      City State
## 26 Consumer Products & Services        17  New York    NY
## 30      Advertising & Marketing        79  New York    NY
## 37      Advertising & Marketing        27  New York    NY
## 38      Advertising & Marketing        89  New York    NY
## 48           Financial Services        32 Rock Hill    NY
## 70      Advertising & Marketing        75  New York    NY
```

Defining outliers with the classic

$$Lower = Q1 - (1.5 * IQR) Upper = Q1 + (1.5 * IQR)$$

removes 45 rows which is too drastic, so I chose to remove only 1% from top and bottom which only removes the top 3 rows. Different quantiles could be chosen as well.

```
# groupby and plot without outliers
groupby_column(inc_ny_no_outlier, Industry, Employees) %>%
  crossbar_plot(Industry, med_grp, med_grp, min_grp, max_grp,
                title = "Median Employee in NY State by Industry (No Outliers)")
```



This showcases the median and range of `Employees` by `Industry` in NY state. That said, it is important to understand which records were removed. If I had more time I would compute that and experiment with more quantiles.

Question 3

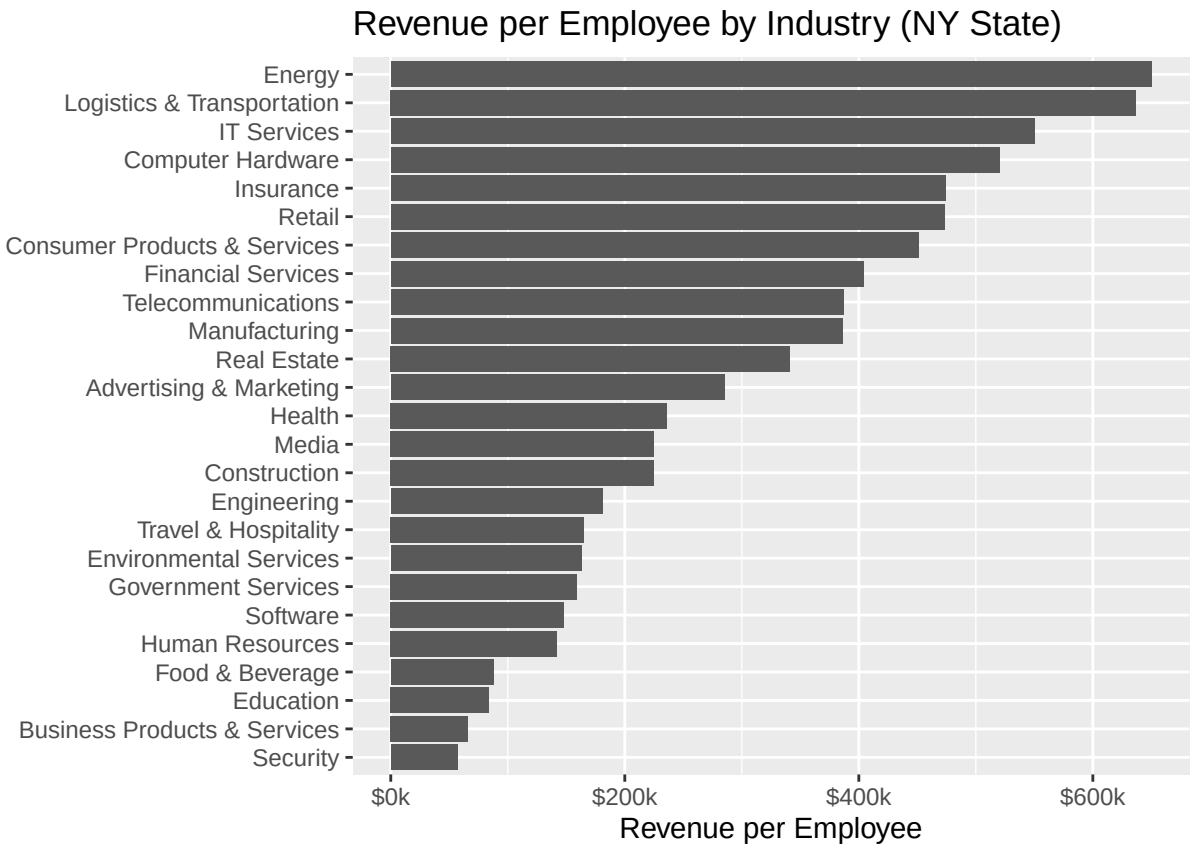
Now imagine you work for an investor and want to see which industries generate the most revenue per employee. Create a chart that makes this information clear. Once again, the distribution per industry should be shown.

```
# group by Industry then get revenue per emp
(inc_ny_rev_emp <- inc_ny %>%
  group_by(Industry) %>%
  summarise(n = n(),
            num_emp = sum(Employees),
            rev_sum = sum(Revenue)) %>%
  mutate(rev_emp_per_1k = round((rev_sum / num_emp) / 1000)) %>%
  select(Industry, n, rev_emp_per_1k) %>%
  arrange(desc(rev_emp_per_1k)))
```

```
## # A tibble: 25 x 3
##    Industry                        n rev_emp_per_1k
##    <chr>                       <int>          <dbl>
##  1 Energy                          5            650
##  2 Logistics & Transportation      4            637
```

```
##  3 IT Services                       43           550
##  4 Computer Hardware                  1           520
##  5 Insurance                          2           474
##  6 Retail                            14           473
##  7 Consumer Products & Services      17           451
##  8 Financial Services                13           404
##  9 Telecommunications                17           387
## 10 Manufacturing                     13           386
## # ... with 15 more rows
```

```
inc_ny_rev_emp %>%
  ggplot() +
    geom_bar(aes(rev_emp_per_1k, reorder(Industry, rev_emp_per_1k)),
             stat = "Identity") +
    scale_x_continuous(labels = function(x) scales::dollar(x, suffix = "k")) +
    labs(y = "", x = "Revenue per Employee",
         title = "Revenue per Employee by Industry (NY State)")
```



This doesn't show the distribution, just the revenue per employee, will need to try a different chart.