Project Milestone 2

Yuning Wu, Justin Yang, David Young

3/13/2022

Project Description

Through this visualization project, we have come up with a set of seven questions for which we identified potential points of interests and areas of use for our targeted audience. These are:

1. State vs salary: which states have the highest paying jobs
2. Industry vs salary: which industries (media, technology, finance for example) have the highest paying jobs
3. Sector vs salary: which sectors (healthcare, government, education) have the highest paying jobs
4. Job title vs salary: which positions earn the most money
5. Company rating vs average salary by company: do higher rated companies on glassdoor pay their workers more
6. Cost of living vs salary: what is the most optimal job in terms of net earnings
7. Comparisons with other similar jobs

Data

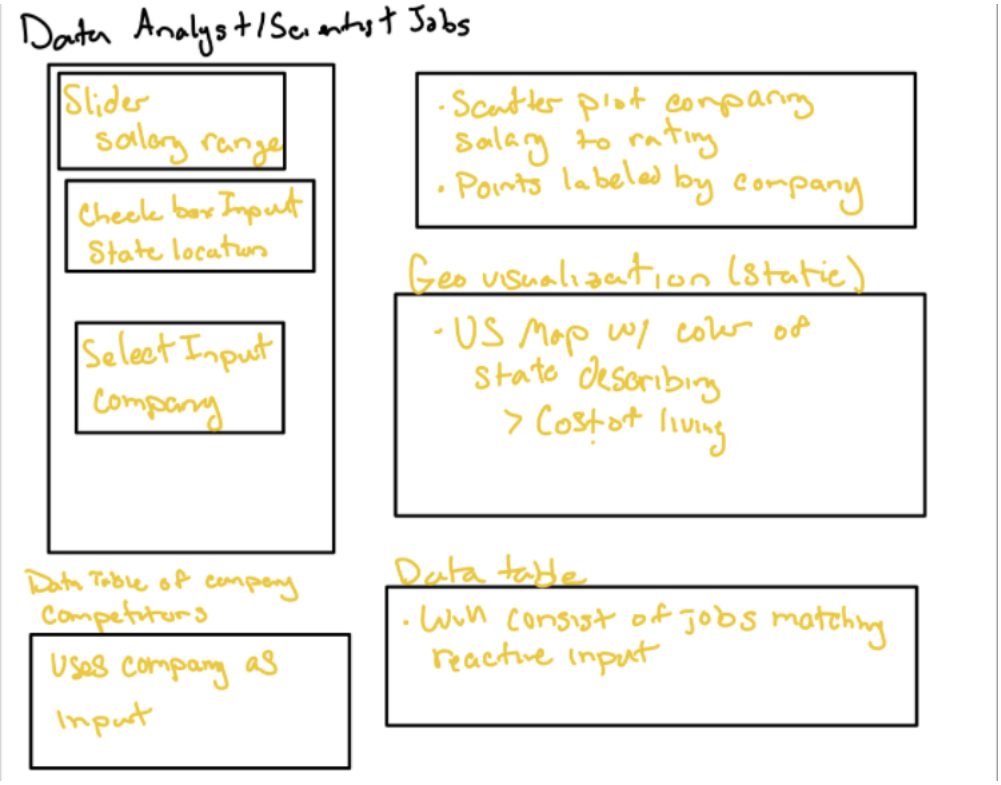
The data we used was scrapped from glassdoor, a review aggregation website of companies. The website allows users to anonymously rate companies, submit salaries and apply for jobs, all important for this project. (Source: wikipedia) The initial kaggle dataset is pre-cleaned and filtered, already containing information about salary, job descriptions and company ratings. After filtering out unwanted variables, removing those with missing data and mutating our columns, our newly processed data contains the Company Name, the

city and state the job is located in, Rating, Industry, Sector, Job Title and two salary related columns: one with Average reported salary ranges and one with the calculated Average Salary based on the Salary Estimate. (the latter is still to be implemented)

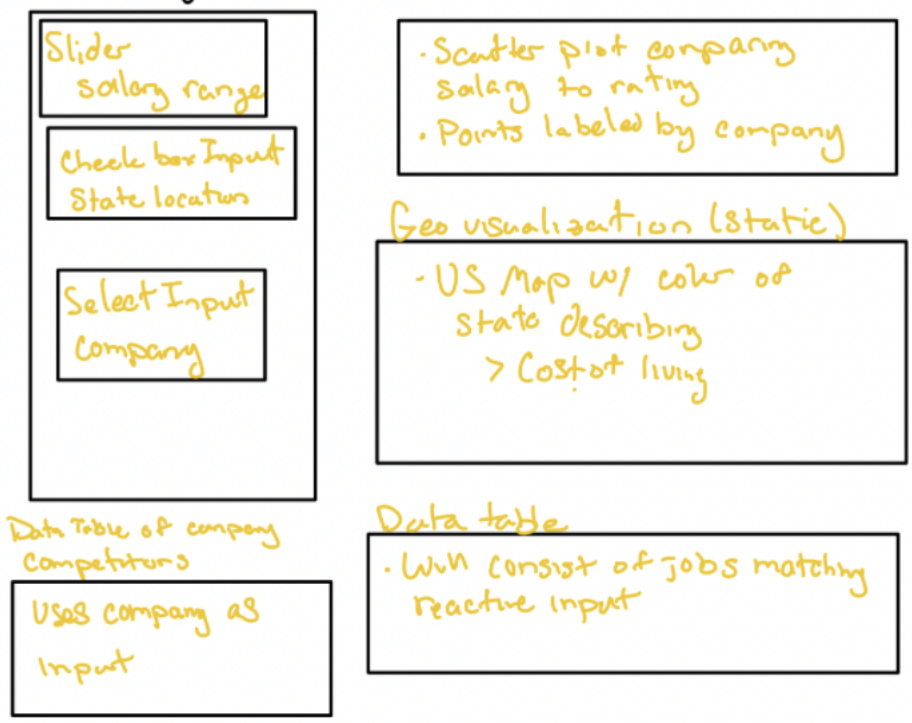
Additionally, we appended a column containing the cost of living index, obtaining the numbers from a secondary calculated dataset from AdvisorSmith. While appending, a major hurdle occurs as AdvisorSmith only accounts for the major cities, not the suburbs found within their respective metropolitan areas. This is most prominently seen for jobs located in the San Francisco Bay Area, as we had to discard jobs located in cities such as Palo Alto, Foster City and Santa Clara. Ultimately, this resulted in around 44% of the initial downloaded dataset being discarded, something that needs to be rectified in the near future.

Sketches of data visualization

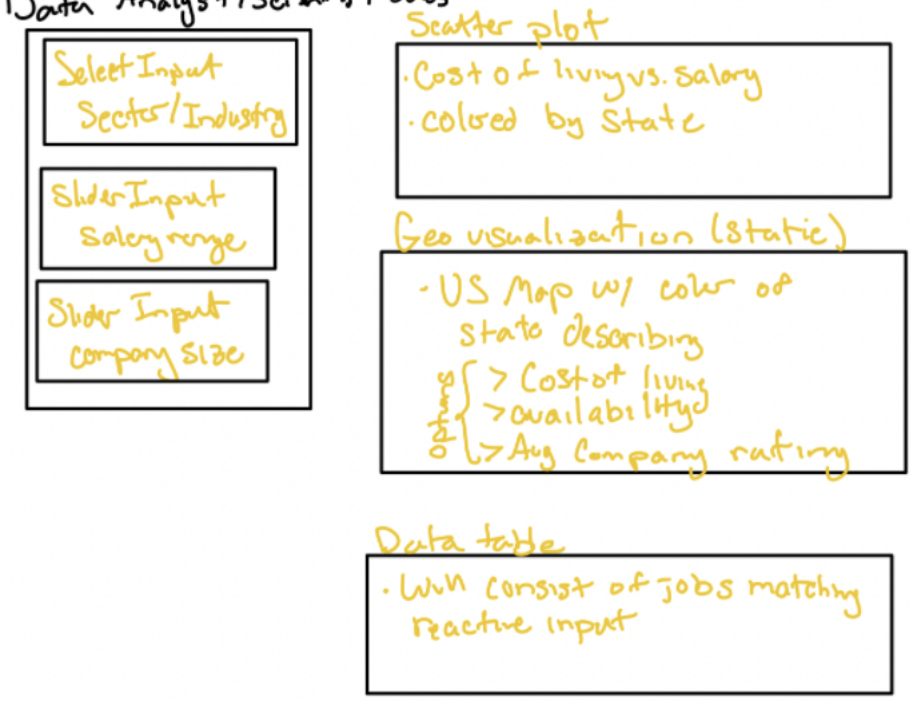
Below are the 3 sketches for the App. The main factors that we are looking to visualize are salary and cost of living.



Data Analyst/Scientist Jobs



Data Analyst/Scientist Jobs



Appendix

```
library(readr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(shiny)
library(shinyWidgets)
```

Data reading and cleaning

```
# Read in data
ds_sal = read_csv("https://uwmadison.box.com/shared/static/l2n9u9d97yxzibvd71y7a370kt0pj1pu.csv")
```

```
## Rows: 742 Columns: 42
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (17): Job Title, Salary Estimate, Job Description, Company Name, Locatio...
## dbl (25): index, Rating, Founded, Hourly, Employer provided, Lower Salary, U...
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
costoliv = read_csv("https://uwmadison.box.com/shared/static/6xphn35fqqq8svwr3y9iis1cpxjqfo8b.csv")
```

```
## Rows: 510 Columns: 3
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): City, State
## dbl (1): Cost of Living Index
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Sneak Preview
head(ds_sal)
```

```
## # A tibble: 6 x 42
##   index `Job Title`  `Salary Estimate` `Job Description`  Rating `Company Name`
##   <dbl> <chr>        <chr>             <chr>               <dbl> <chr>
## 1     0 Data Scient~ $53K-$91K (Glass~ "Data Scientist\n~    3.8 "Tecolote Rese~
## 2     1 Healthcare ~ $63K-$112K (Glas~ "What You Will Do~    3.4 "University of~
## 3     2 Data Scient~ $80K-$90K (Glass~ "KnowBe4, Inc. is~    4.8 "KnowBe4\n4.8"
## 4     3 Data Scient~ $56K-$97K (Glass~ "*Organization an~    3.8 "PNNL\n3.8"
## 5     4 Data Scient~ $86K-$143K (Glas~ "Data Scientist\n~    2.9 "Affinity Solu~
## 6     5 Data Scient~ $71K-$119K (Glas~ "CyrusOne is seek~    3.4 "CyrusOne\n3.4"
## # ... with 36 more variables: Location <chr>, Headquarters <chr>, Size <chr>,
## #   Founded <dbl>, Type of ownership <chr>, Industry <chr>, Sector <chr>,
## #   Revenue <chr>, Competitors <chr>, Hourly <dbl>, Employer provided <dbl>,
## #   Lower Salary <dbl>, Upper Salary <dbl>, Avg Salary(K) <dbl>,
## #   company_txt <chr>, Job Location <chr>, Age <dbl>, Python <dbl>,
## #   spark <dbl>, aws <dbl>, excel <dbl>, sql <dbl>, sas <dbl>, keras <dbl>,
## #   pytorch <dbl>, scikit <dbl>, tensor <dbl>, hadoop <dbl>, tableau <dbl>, ...
```

```
names(ds_sal)
```

```
##  [1] "index"              "Job Title"          "Salary Estimate"
##  [4] "Job Description"    "Rating"             "Company Name"
##  [7] "Location"           "Headquarters"       "Size"
## [10] "Founded"            "Type of ownership"  "Industry"
## [13] "Sector"             "Revenue"            "Competitors"
## [16] "Hourly"             "Employer provided"  "Lower Salary"
## [19] "Upper Salary"       "Avg Salary(K)"      "company_txt"
## [22] "Job Location"       "Age"                "Python"
## [25] "spark"              "aws"                "excel"
## [28] "sql"                "sas"                "keras"
## [31] "pytorch"            "scikit"             "tensor"
## [34] "hadoop"             "tableau"            "bi"
## [37] "flink"              "mongo"              "google_an"
## [40] "job_title_sim"      "seniority_by_title" "Degree"
```

```
head(costoliv)
```

```
## # A tibble: 6 x 3
##   City       State `Cost of Living Index`
##   <chr>      <chr>                  <dbl>
## 1 Abilene    TX                      89.1
## 2 Adrian     MI                      90.5
## 3 Akron      OH                      89.4
## 4 Alamogordo NM                      85.8
## 5 Albany     GA                      87.3
## 6 Albany     OR                     105.
```

```
names(costoliv)
```

```
## [1] "City"                 "State"                "Cost of Living Index"
```

```r
# Separating Location into City and State
ds_salfix = ds_sal %>%
  separate(Location, c('City','State'),sep = ', ') #ignore warning, will be fixed in next line
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 1 rows [127].
```

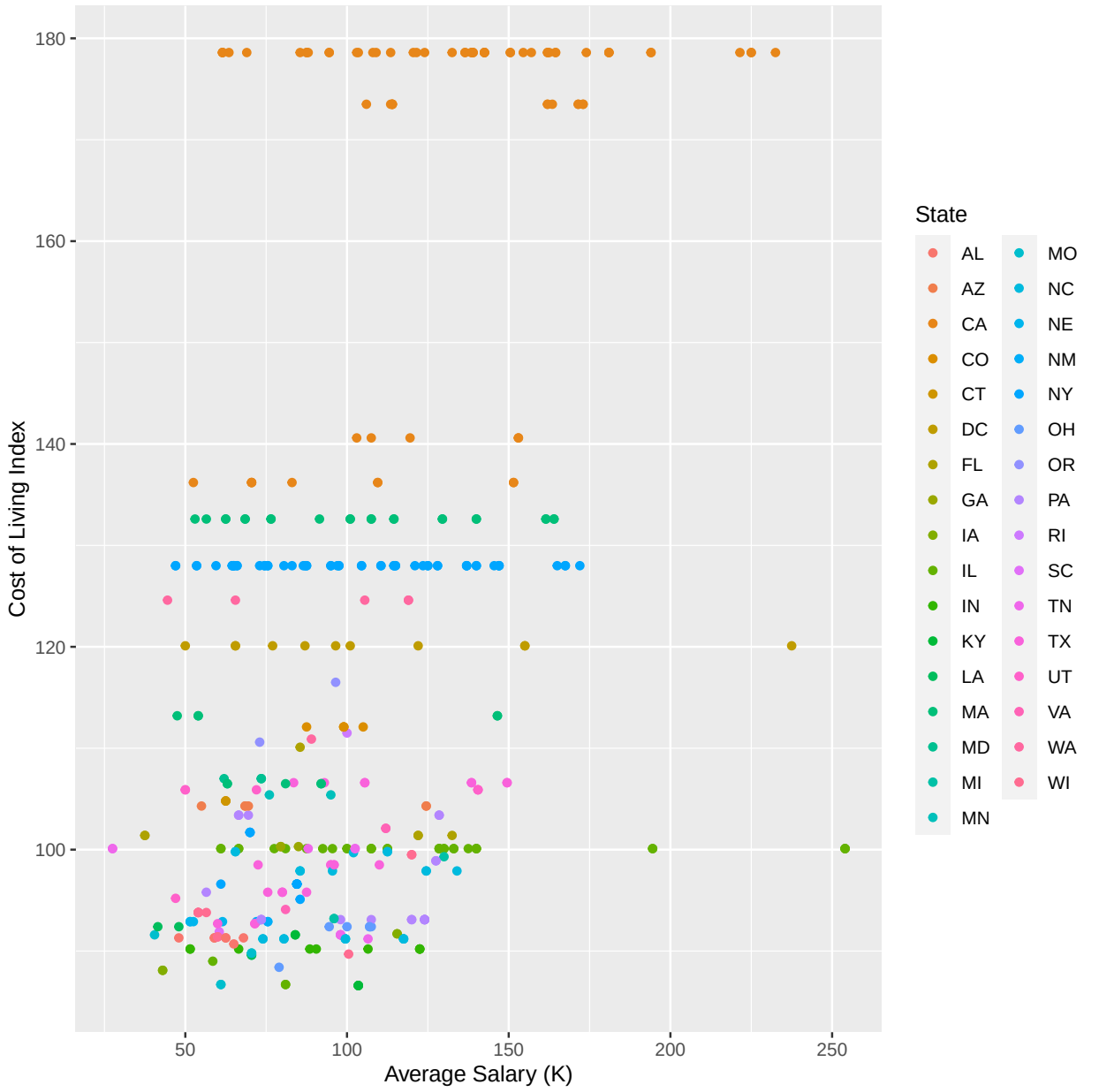
```r
ds_salfix[127,]$City = 'Los Angeles'
ds_salfix[127,]$State = 'CA'

# Filtering columns to our desired variables
ds_salfix = ds_salfix[,c(1:3,5:8,13:14,19:21)]
ds_salfix$cost.of.living = NA #empty column

# Appending Cost of Living onto dataframe
for(i in seq(length(ds_salfix$City))){
  if (ds_salfix[i,]$City %in% costoliv$City) {
    ix = which(costoliv$City == ds_salfix[i,]$City)
    for (j in ix){
      if(ds_salfix[i,]$State == costoliv[j,]$State){
        # print(i)
        # print(ds_salfix[i,]$City)
        # print(j)
        # print(costoliv[j,]$State)
        # print(ds_salfix[i,]$cost.of.living)
        ds_salfix[i,]$cost.of.living = costoliv[j,]$`Cost of Living Index`
      }
    }
  }
}
```

Short Visualization 1 of Average Salary vs Cost of living:

```
ds_salfix %>%
  drop_na(cost.of.living) %>%
  ggplot() +
  geom_point(aes(`Avg Salary(K)`, cost.of.living, col = State)) +
  labs(
    x = "Average Salary (K)",
    y = "Cost of Living Index",
    color = "State"
  )
```



This short, non-dynamic visualization plots the relationship between average salary and cost of living. A simple observation is that most of the points a plopped under an average salary of $150,000 and cost of living index of 140, many California points are located near the cost of living index of 180 and our desired jobs that maximizes pay and minimizes the cost of living index are found within the state of Illinois (presumably all in Chicago)

Missing Data exploration

```
missingcities = c()
`%!in%` <- Negate(`%in%`)
for(i in seq(length(ds_salfix$City))){
  if (ds_salfix[i,]$City %!in% costoliv$City) {
    missingcities = c(missingcities, ds_sal[i,]$Location)
  }
}
missingcities %>%
  unique()
```

```
##   [1] "Linthicum, MD"           "Clearwater, FL"
##   [3] "Richland, WA"            "Chantilly, VA"
##   [5] "Plano, TX"               "Cambridge, MA"
##   [7] "Newark, NJ"              "Mountain View, CA"
##   [9] "Herndon, VA"             "Hillsboro, OR"
##  [11] "Groton, CT"              "Sunnyvale, CA"
##  [13] "Ipswich, MA"             "Redlands, CA"
##  [15] "Woburn, MA"              "Fremont, CA"
##  [17] "Long Beach, NY"          "Marlborough, MA"
##  [19] "Allendale, NJ"           "Bellevue, WA"
##  [21] "Longmont, CO"            "Beavercreek, OH"
##  [23] "Fort Lauderdale, FL"     "Armonk, NY"
##  [25] "Saint Louis, MO"         "Palo Alto, CA"
##  [27] "Coraopolis, PA"          "Framingham, MA"
##  [29] "Vancouver, WA"           "Lake Forest, IL"
##  [31] "Maryland Heights, MO"    "Arlington, VA"
##  [33] "Tacoma, WA"              "Landover, MD"
##  [35] "Patuxent River, MD"      "Suitland, MD"
##  [37] "McLean, VA"              "Fort Belvoir, VA"
##  [39] "Silver Spring, MD"       "Southfield, MI"
##  [41] "Matawan, NJ"             "Lyndhurst, NJ"
```

```
##  [43] "Rockville, MD"          "Alabaster, AL"
##  [45] "Ashburn, VA"            "Fort Worth, TX"
##  [47] "Valencia, CA"           "Novato, CA"
##  [49] "Aurora, CO"             "Riverton, UT"
##  [51] "Ewing, NJ"              "South San Francisco, CA"
##  [53] "Cupertino, CA"          "Frederick, MD"
##  [55] "West Reading, PA"       "Dearborn, MI"
##  [57] "Winter Park, FL"        "San Rafael, CA"
##  [59] "Hamilton, NJ"           "Woodbridge, NJ"
##  [61] "Lewes, DE"              "Burbank, CA"
##  [63] "Newton, MA"             "Annapolis Junction, MD"
##  [65] "Highland, CA"           "Burleson, TX"
##  [67] "Hoopeston, IL"          "Scotts Valley, CA"
##  [69] "Millville, DE"          "San Mateo, CA"
##  [71] "Parlier, CA"            "Cherry Hill, NJ"
##  [73] "Port Washington, NY"    "Phila, PA"
##  [75] "Oakland, CA"            "Boise, ID"
##  [77] "Oak Ridge, TN"          "Agoura Hills, CA"
##  [79] "Pella, IA"              "San Ramon, CA"
##  [81] "Red Bank, NJ"           "West Palm Beach, FL"
##  [83] "Exton, PA"              "Orange, CA"
##  [85] "Lenexa, KS"             "Vail, CO"
##  [87] "Natick, MA"             "Richfield, OH"
##  [89] "Hampton, VA"            "Marietta, GA"
##  [91] "Clovis, CA"             "Chandler, AZ"
##  [93] "Westlake, OH"           "Fort Lee, NJ"
##  [95] "Blue Bell, PA"          "Jersey City, NJ"
##  [97] "Emeryville, CA"         "Santa Barbara, CA"
##  [99] "Carle Place, NY"        "King of Prussia, PA"
## [101] "Santa Clara, CA"        "Brisbane, CA"
## [103] "Foster City, CA"        "Holyoke, MA"
## [105] "Waltham, MA"            "Gaithersburg, MD"
## [107] "Bedford, MA"            "Aliso Viejo, CA"
```

```
## [109] "Dublin, CA"              "Arvada, CO"
## [111] "Franklin, TN"            "Plymouth Meeting, PA"
## [113] "Reston, VA"              "Scottsdale, AZ"
## [115] "Alameda, CA"             "Glen Burnie, MD"
## [117] "Milpitas, CA"            "Cambridge, MD"
## [119] "Irvine, CA"
```

```
cutdata = length(missingcities %>%
  unique())/length(ds_salfix$City)


cutdata*100
```

```
## [1] 16.03774
```

As mentioned in the Data section, a lot of suburbs of metropolitan areas are missing from the cost of living data set, with the large ones being from the SF Bay Area, as well as in DC, New York, Philadelphia, Boston and Seattle, or areas involving big tech, resulting in our cut data of ~45%. Thus it provides a need way to account for these and rectify this issue.

Short Visualization 2

```
ds_salfix <- ds_salfix %>%
  rename(Avg.Salary.K = 'Avg Salary(K)',
         Job.Title = 'Job Title') %>%
  mutate(Avg.Salary.K = as.numeric(Avg.Salary.K))


# Boxplot
box = function(salary, x){
  # salary = filter(salary, selected == TRUE)
  ggplot(salary) +
    geom_boxplot(aes(x = "Avg Salary(K)",
                     y = x,
                     fill=x)) +
```

```
    theme(legend.position="bottom") +
    labs(title= "Boxplot of Average Salaries",
         x= "Average Salary (K)",
         y= x)
}

ui = fluidPage(
  titlePanel(h1("Average Salaries for Data Science Jobs", align = "center")),
  inputPanel(
    selectInput("x_var", "Select a variable that you want to compare average salaries",
                c("State" = "State", "Job Title" = "Job.Title"), selected = "State")
  ),
  mainPanel(plotOutput("boxplot"))
)

server = function(input, output){
  output$boxplot = renderPlot({box(ds_salfix, input$x_var)})
}

app <- shinyApp(ui, server)

app
```

```
## PhantomJS not found. You can install it with webshot::install_phantomjs(). If it is installed, pleas
```

For this visualization, we created a shiny app which allows the user to select an x variable to compare average salaries. At this stage, we offer two choices: State and Sector. After choosing the desired x variable, the user will be able to view a box plot that shows the average salary of data science related jobs in thousands being compared across that variable.