Application of Shiny: College Majors and Income and Employment

terry min

12/03/2017

1.Intro

Last time on Post01, I used ggplot2 and dplyr packages in order to confirm the several arguments about the median salaries of different college majors. Some of the major arguments I confirmed and I conclued:

- Engineering majors receive more early career median salaries.
- STEM majors received more payments in general, except for the majors under the Biology & Life Science major category. (However, we should not conclude that all majors under that category receive less payments: It's about the average!)
- Among non-STEM majors, Pharmacy Pharmaceutical Sciences and Administration under the Health major category received the most. However, in general, majors under the Business major category took the majority of the top 10 non-STEM majors in terms of early career median salaries.
- No relationship could be found between being a STEM major and average salaries.

Median Income, however, is not the only thing that's important when considering majors. Other factors, such as employment rates, are also important. Some people might also be interested in the 25th or 75th percentiles. Therefore, this time, I would like to create my own measure of good college majors considering and rescaling factors such as median salaries, employment rates, 25th and 75th percentile of the salaries. Rather than focusing on individual college majors, I would like to focus on the distribution by turning the measure into grades. (This is a subjective measure.) I would use the Shiny app to visualize the data.

2. Data Preparation

```r
# Required packages
# I assumed that I've already installed the necessary packages.
library(readr)
library(shiny)
```

```
## Warning: package 'shiny' was built under R version 3.4.1
```

```r
library(ggvis)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
# download RData file into your working directory
github <- "https://github.com/fivethirtyeight/data/tree/master/college-majors/"
csv <- "all-ages.csv"
download.file(url = paste0(github, csv), destfile = 'all-ages.csv')
```

```r
# Read in the CSV file using the 'read.csv()' function
income_1 <- read.csv('../data/all-ages.csv')
income <- income_1
```

```r
# Define rescale function to compute any rescaled vector with a potential scale from 0 to 100
rescale100 <- function(x, na.rm = TRUE, xmin, xmax) {
  if(!is.numeric(x)) {
    stop("non-numeric argument")
  }
  return(100 * ((x - xmin)/(xmax - xmin)))
}
```

```r
# Add a variable emp_rate to the data frame
income <- mutate(income, emp_rate = 1 - Unemployment_rate)

# Use rescale100() to add a variable rescaled_median by rescaling Total: 0 is the minimum, and 100 is the max.
#max(income$Median)
income <- mutate(income, rescaled_median = rescale100(income$Median, xmin = 0, xmax = 125000))

# Use rescale100() to add a variable rescaled_emp_rate by rescaling emp_rate: 0 is the minimum, and 1 is the max.
income <- mutate(income, rescaled_emp_rate = rescale100(income$emp_rate, xmin = 0, xmax = 1))

# Use rescale100() to add a variable rescaled_25 by rescaling P25th: 0 is the minimum, and 78000 is the max.
#max(income$P25th) #78000
income <- mutate(income, rescaled_25 = rescale100(income$P25th, xmin = 0, xmax = 78000))

# Use rescale100() to add a variable rescaled_75 by rescaling P75th: 0 is the minimum, and 100 is the max.
#max(income$P75th) #210000
income <- mutate(income, rescaled_75 = rescale100(income$P75th, xmin = 0, xmax = 210000))
```

```r
# Add a variable Overall to the data frame of scores
income <- mutate(income, Overall = 0)
for(i in 1:nrow(income)){
  prop_median <- as.vector(unlist(income[i, which(colnames(income)=="rescaled_median")]))
  prop_unemp_rate <- as.vector(unlist(income[i, which(colnames(income)=="rescaled_emp_rate")]))
  prop_p25th <- as.vector(unlist(income[i, which(colnames(income)=="rescaled_25")]))
  prop_p75th <- as.vector(unlist(income[i, which(colnames(income)=="rescaled_75")]))
  income[i, which(colnames(income)=="Overall")] <- ((0.5*prop_median) + (0.3*prop_unemp_rate) + (0.1*prop_p25th) +
(0.1*prop_p75th))
}

# Use rescale100() to add a variable rescaled_Overall by rescaling Overall
# max(income$Overall) #98.34922
income <- mutate(income, rescaled_Overall = rescale100(income$Overall, xmin = 0, xmax = 98.34922))
```

```r
# Calculate a variable Grade
income <- mutate(income, Grade = "")
for(i in 1:nrow(income)){
  if(income$rescaled_Overall[i] < 50) {income$Grade[i] <- "F"}
  if(income$rescaled_Overall[i] >= 50 & income$rescaled_Overall[i] < 60) {income$Grade[i] <- "D"}
  if(income$rescaled_Overall[i] >= 60 & income$rescaled_Overall[i] < 70) {income$Grade[i] <- "C-"}
  if(income$rescaled_Overall[i] >= 70 & income$rescaled_Overall[i] < 77.5) {income$Grade[i] <- "C"}
  if(income$rescaled_Overall[i] >= 77.5 & income$rescaled_Overall[i] < 79.5) {income$Grade[i] <- "C+"}
  if(income$rescaled_Overall[i] >= 79.5 & income$rescaled_Overall[i] < 82) {income$Grade[i] <- "B-"}
  if(income$rescaled_Overall[i] >= 82 & income$rescaled_Overall[i] < 86) {income$Grade[i] <- "B"}
  if(income$rescaled_Overall[i] >= 86 & income$rescaled_Overall[i] < 88) {income$Grade[i] <- "B+"}
  if(income$rescaled_Overall[i] >= 88 & income$rescaled_Overall[i] < 90) {income$Grade[i] <- "A-"}
  if(income$rescaled_Overall[i] >= 90 & income$rescaled_Overall[i] < 95) {income$Grade[i] <- "A"}
  if(income$rescaled_Overall[i] >= 95) {income$Grade[i] <- "A+"}
}
```

```r
# Convert some variables as factors for barcharts
income$Grade <- factor(income$Grade,
                          levels = c('A+', 'A', 'A-', 'B+', 'B', 'B-', 'C+', 'C', 'C-', 'D', 'F'))
```

```r
# Variable names for barcharts
categorical <- 'Grade'
```

```r
# Variable names for histograms
continuous <- c('rescaled_emp_rate','rescaled_median','rescaled_25', 'rescaled_75', 'rescaled_Overall')
```

```r
# Frequency table
grade_categories <- c('A+', 'A', 'A-', 'B+', 'B', 'B-', 'C+', 'C', 'C-', 'D', 'F')
grade_frequency <- rep(0, 11)
for (i in 1:11) {
  grade_frequency[i] <- sum(income$Grade == grade_categories[i])
}
grade_proportion <- rep(0, 11)
for (i in 1:11) {
  grade_proportion[i] <- grade_frequency[i] / sum(grade_frequency)
}
grade_df <- data.frame(
  Grade = grade_categories,
  Freq = as.integer(grade_frequency),
  Prop = round(grade_proportion, 2))
```

```
## Define UI for application that draws a histogram
ui <- fluidPage(
  titlePanel("Grade Visualizer"),
  sidebarLayout(
    sidebarPanel(
      conditionalPanel(condition = "input.tabselected==1",
                       h4("Grade Distribution"),
                       tableOutput("table")),
      conditionalPanel(condition = "input.tabselected==2",
                       selectInput("var2", "X-axis variable", continuous,
                                   selected = "rescaled_median"),
                       sliderInput("width", "Bin Width",
                                   min = 1, max = 10, value = 10)),
      conditionalPanel(condition = "input.tabselected==3",
                       selectInput("var3", "X-axis variable", continuous,
                                   selected = "rescaled_median"),
                       radioButtons("show", "Show line",
                                    choices = list("none" = 1, "lm" = 2, "loess" = 3),
                                    selected = 1),
                       selectInput("var4", "Y-axis variable", continuous,
                                   selected = "emp_rate"),
                       sliderInput("opac", "Opacity",
                                   min = 0, max = 1, value = 0.5))),
    mainPanel(
      tabsetPanel(type = "tabs",
                  tabPanel("Barchart", value = 1,
                           ggvisOutput("barchart")),
                  tabPanel("Histogram", value = 2,
                           ggvisOutput("histogram"),
                           verbatimTextOutput("stats")),
                  tabPanel("Scatterplot", value = 3,
                           ggvisOutput("scatterplot"),
                           h5("Correlation:"),
                           verbatimTextOutput("cor")),
                  id = "tabselected")
    ))
)
```

```
# Server logic
server <- function(input, output) {
  output$table <- renderTable({grade_df})
  output$cor <- renderPrint(cat(cor(as.vector(unlist(income[,eval(input$var3)])), as.vector(unlist(income[,eval(in
put$var4)])))))
    vis_barchart <- reactive({
    var1 <- prop("x", as.symbol("Grade"))
    income %>%
      ggvis(x = var1, fill := "#75AADB", stroke := "#75AADB", opacity := 0.8) %>%
      add_axis("y", title = "frequency")
  })
  vis_barchart %>% bind_shiny("barchart")

  vis_histogram <- reactive({
    var2 <- prop("x", as.symbol(input$var2))
    income %>%
      ggvis(x = var2, fill := "grey") %>%
      layer_histograms(stroke := 'white',
                       width = input$width)
  })
  vis_histogram %>% bind_shiny("histogram")

  vis_scatterplot <- reactive({
    var3 <- prop("x", as.symbol(input$var3))
    var4 <- prop("y", as.symbol(input$var4))
    if (input$show == 1) {
      income %>%
        ggvis(x = var3, y = var4) %>%
        layer_points(fillOpacity := input$opac, fillOpacity.hover := 1)
    } else if (input$show == 2) {
      income %>%
        ggvis(x = var3, y = var4) %>%
        layer_points(fillOpacity := input$opac, fillOpacity.hover := 1) %>%
        layer_model_predictions(model = "lm")
    } else if (input$show == 3) {
      income %>%
        ggvis(x = var3, y = var4) %>%
        layer_points(fillOpacity := input$opac, fillOpacity.hover := 1) %>%
        layer_model_predictions(model = "loess")
    }
  })
  vis_scatterplot %>% bind_shiny("scatterplot")
}
```

```r
# Application
shinyApp(ui = ui, server = server)
```

Shiny applications not supported in static R Markdown documents

3. Questions

(1) How can we interpret the grade frequency?

Answer: Looking at the “grade” distribution, only two majors are considered to be in the A range and most college majors are considered to be in the C- and D range. This can be interpreted that those two majors with “A” might be outliers and the measure of grade might need some adjustments. But also, this might be an indication that most majors receive similar amount of salaries and have similar employment rate.

(2) What can be said about the employment rate?

Answer: Looking at the histogram, and changing the Bid Width, we can see that there are only few majors with employment rate lower than 90%.

(3) What can be said about the relationship between the median salary and the employment rate?

Answer: The correlation between the rescaled median value and the rescaled employment rate are 0.3, indicating that there is a moderate positive linear relationship between the two.

4. Conclusion

Although I used a subjective measure for considering “good majors,” the fact that most firms are in the same range in the distribution might indicate that most majors, except for some outliers, receive similar amount of salaires and have similar employment rate.

Still, the measure seems to need some adjustments. By looking into the individual data on the histogram using the Shiny app, it is obvious that there are outliers in the measures.

Finally, there is a moderate correlation between the median salary and the employment rate. This might be indicating that the majors with higher salaries are more likely to be highered. This also might be an indication that those majors are in high demand in job market. However, further analysis is nedeeded to justify this argument.

5. References

1. [GlassDoor: 50 Highest Paying College Majors][https://www.glassdoor.com/blog/50-highest-paying-college-majors/]
2. [BusinessInsider: The college majors with the highest starting salaries][https://www.youtube.com/watch?v=_oRlrcoy4xw]
3. [PayScale: Highest Paying Bachelor Degrees by Salary Potential][https://www.payscale.com/college-salary-report/majors-that-pay-you-back/bachelors]
4. [Github Data][https://github.com/fivethirtyeight/data/tree/master/college-majors]
5. [American Community Survey 2010-2012 Public Use Microdata Series][http://www.census.gov/programs-surveys/acs/data/pums.html]
6. [The Economic Guide To Picking A College Major][https://fivethirtyeight.com/features/the-economic-guide-to-picking-a-college-major/]
7. [Carnevale et al, “What’s It Worth?: The Economic Value of College Majors.”][http://cew.georgetown.edu/whatsitworth]
8. [US News: Best Undergraduate Petroleum Engineering Programs (Doctorate)][https://www.usnews.com/best-colleges/rankings/engineering-doctorate-petroleum]
9. [Shiny App Tutorials][https://shiny.rstudio.com/tutorial/]