Analysis 1: UNC Salaries

Anthony Hu

6/3/2021

Instructions

Overview: For each question, show your R code that you used to answer each question in the provided chunks. When a written response is required, be sure to answer the entire question in complete sentences outside the code chunks. When figures are required, be sure to follow all requirements to receive full credit. Point values are assigned for every part of this analysis.

Helpful: Make sure you knit the document as you go through the assignment. Check all your results in the created PDF or HTML file.

Submission: Submit via an electronic document on Sakai. Must be submitted as an HTML or a PDF file generated in RStudio.

Introduction

Universities are typically opaque, bureaucratic institutions. To be transparent to tax payers, many public schools, such as the University of North Carolina, openly report **salary information**. In this assignment, we will analyze this information to answer pivotal questions that have endured over the course of time. The most recent salary data for UNC-Chapel Hill faculty and staff has already been downloaded in CSV format and titled *"UNC_System_Salaries Search and Report.csv"*. People get depressed when they see that many digits after the decimal.

To answer all the questions, you will need the R package `tidyverse` to make figures and utilize `dplyr` functions.

Data Information

Make sure the CSV data file is contained in the folder of your RMarkdown file. First, we start by using the `read_csv` function from the `readr` package found within the tidyverse. The code below executes this process by creating a tibble in your R environment named "salary".

```
salary=read_csv("UNC_System_Salaries Search and Report.csv")
salary
```

```
## # A tibble: 12,646 x 13
##    Name   campus2  dept   position   PRIMARY_WORKING~ hiredate exempt   fte
##    <chr>  <chr>    <chr>  <chr>      <chr>            <chr>    <chr>  <dbl>
##  1 AACHO~ UNC-CHA~ Micro~ Research ~ Research Associ~ 10/10/2~ Exemp~   1
##  2 AARNI~ UNC-CHA~ SW-Re~ Functiona~ Graphic Designer 1/14/20~ Subje~   0.8
##  3 ABAJA~ UNC-CHA~ Peds-~ Assistant~ NODESCR          7/1/2015 Exemp~   1
##  4 ABARB~ UNC-CHA~ Kenan~ Associate~ Associate Profe~ 1/1/1999 Exemp~   1
##  5 ABARE~ UNC-CHA~ Insti~ Research ~ Research Techni~ 9/12/20~ Subje~   1
##  6 ABATE~ UNC-CHA~ Med A~ Fiscal Af~ Accounting Tech~ 4/20/20~ Subje~   1
##  7 ABATE~ UNC-CHA~ Schoo~ Administr~ Student Service~ 1/3/2012 Subje~   1
```

```
##  8 ABBEN~ UNC-CHA~ Drama~ Lecturer   Master Electric~ 6/20/20~ Exemp~   1
##  9 ABBOT~ UNC-CHA~ Med A~ Human Res~ HR Consultant    10/3/20~ Subje~   1
## 10 ABD-E~ UNC-CHA~ Schoo~ Dean Educ~ Dean & Professor 7/1/2016 Exemp~   1
## # ... with 12,636 more rows, and 5 more variables: employed <dbl>,
## #   statesal <lgl>, nonstsal <lgl>, totalsal <dbl>, stservyr <dbl>
```

Now, we will explore the information that is contained in this dataset. The code below provides the names of the variables contained in the dataset.

```
names(salary)
```

```
##  [1] "Name"                  "campus2"
##  [3] "dept"                  "position"
##  [5] "PRIMARY_WORKING_TITLE" "hiredate"
##  [7] "exempt"                "fte"
##  [9] "employed"              "statesal"
## [11] "nonstsal"              "totalsal"
## [13] "stservyr"
```

Next, we will examine the type of data contains in these different variables.

```
str(salary,give.attr=F)
```

```
## spec_tbl_df [12,646 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ Name                  : chr [1:12646] "AACHOUI, YOUSSEF" "AARNIO, REA T" "ABAJAS, YASMINA L" "ABARI
##  $ campus2               : chr [1:12646] "UNC-CHAPEL HILL" "UNC-CHAPEL HILL" "UNC-CHAPEL HILL" "UNC-CI
##  $ dept                  : chr [1:12646] "Microbiology and Immunology" "SW-Research Projects" "Peds-He
##  $ position              : chr [1:12646] "Research Professional, Medical" "Functional Paraprofessional
##  $ PRIMARY_WORKING_TITLE: chr [1:12646] "Research Associate" "Graphic Designer" "NODESCR" "Associate
##  $ hiredate              : chr [1:12646] "10/10/2011" "1/14/2013" "7/1/2015" "1/1/1999" ...
##  $ exempt                : chr [1:12646] "Exempt from Personnel Act" "Subject to State Personnel Act"
##  $ fte                   : num [1:12646] 1 0.8 1 1 1 1 1 1 1 1 ...
##  $ employed              : num [1:12646] 12 12 12 9 12 12 12 9 12 9 ...
##  $ statesal              : logi [1:12646] NA NA NA NA NA NA ...
##  $ nonstsal              : logi [1:12646] NA NA NA NA NA NA ...
##  $ totalsal              : num [1:12646] 49128 33257 139405 181000 41098 ...
##  $ stservyr              : num [1:12646] 1 5 2 20 6 8 6 1 19 1 ...
```

You will notice that the variable "hiredate" is recorded as a character. The following code will first modify the original dataset to change this to a date variable with the format *mm/dd/yyyy*. Then, we will remove the hyphens to create a numeric variable as *yyyymmdd*. Finally, in the spirit of tidyverse, we will convert this data frame to a tibble.

```
salary$hiredate=as.Date(salary$hiredate, format="%m/%d/%Y")
salary$hiredate=as.numeric(gsub("-","",salary$hiredate))
salary=as_tibble(salary)
```

Now, we will use `head()` to view of first five rows and the modifications made to the original data. The rest of the assignment will extend off this modified dataset named `salary` which by now should be in your global environment.

```
head(salary,5)
```

```
## # A tibble: 5 x 13
##   Name   campus2  dept    position  PRIMARY_WORKING~ hiredate exempt    fte
##   <chr>  <chr>    <chr>   <chr>     <chr>               <dbl> <chr>   <dbl>
## 1 AACHO~ UNC-CHA~ Microb~ Research~ Research Associ~ 20111010 Exempt~   1
## 2 AARNI~ UNC-CHA~ SW-Res~ Function~ Graphic Designer 20130114 Subjec~   0.8
## 3 ABAJA~ UNC-CHA~ Peds-H~ Assistan~ NODESCR          20150701 Exempt~   1
## 4 ABARB~ UNC-CHA~ Kenan-~ Associat~ Associate Profe~ 19990101 Exempt~   1
## 5 ABARE~ UNC-CHA~ Instit~ Research~ Research Techni~ 20110912 Subjec~   1
## # ... with 5 more variables: employed <dbl>, statesal <lgl>,
## #   nonstsal <lgl>, totalsal <dbl>, stservyr <dbl>
```

Assignment

Part 1: Reducing the Data to a Smaller Set of Interest

Q1 *(2 Points)*

Create a new dataset named `salary2` that only contains the following variables:

- "Name"
- "dept"
- "position"
- "hiredate"
- "exempt"
- "totalsal"

Then, use the `names()` function to display the variable names of `salary2`.

```
#
salary2 <- select(salary, Name, dept, position, hiredate, exempt, totalsal)
names(salary2)
```

```
## [1] "Name"     "dept"     "position" "hiredate" "exempt"   "totalsal"
```

Q2 *(2 Points)*

Now, we modify `salary2`. Rename the variables "dept","position","exempt","totalsal" to "Department","Job","Exempt", and "Salary", respectively. Do this for a new dataset called `salary3` and use `names()` to display the variable names of `salary3`.

```
#
salary3 <- rename(salary2, Department = dept, Job = position, Exempt = exempt, Salary = totalsal)
names(salary3)
```

```
## [1] "Name"       "Department" "Job"        "hiredate"   "Exempt"
## [6] "Salary"
```

Q3 ***(2 Points)***

Now, we modify `salary3`. Create a new variable called "HireYear" that only contains the first four digits of the variable "hiredate" in a new dataset named `salary4`. *Hint: Use the concept seen in the conversion of flight times to minutes since midnight.* Use the function `str()` to ensure that your new variable "HireYear" reports the year of the date that the employee was hired.

```
#
salary4 <- mutate(salary3, HireYear = hiredate %/% 10000)
str(salary4)
```

```
## tibble [12,646 x 7] (S3: tbl_df/tbl/data.frame)
##  $ Name      : chr [1:12646] "AACHOUI, YOUSSEF" "AARNIO, REA T" "ABAJAS, YASMINA L" "ABARBANELL, JEFI
##  $ Department: chr [1:12646] "Microbiology and Immunology" "SW-Research Projects" "Peds-Hematology/On
##  $ Job       : chr [1:12646] "Research Professional, Medical" "Functional Paraprofessional" "Assistar
##  $ hiredate  : num [1:12646] 20111010 20130114 20150701 19990101 20110912 ...
##  $ Exempt    : chr [1:12646] "Exempt from Personnel Act" "Subject to State Personnel Act" "Exempt fro
##  $ Salary    : num [1:12646] 49128 33257 139405 181000 41098 ...
##  $ HireYear  : num [1:12646] 2011 2013 2015 1999 2011 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Name = col_character(),
##   ..   campus2 = col_character(),
##   ..   dept = col_character(),
##   ..   position = col_character(),
##   ..   PRIMARY_WORKING_TITLE = col_character(),
##   ..   hiredate = col_character(),
##   ..   exempt = col_character(),
##   ..   fte = col_double(),
##   ..   employed = col_double(),
##   ..   statesal = col_logical(),
##   ..   nonstsal = col_logical(),
##   ..   totalsal = col_double(),
##   ..   stservyr = col_double()
##   .. )
```

Q4 ***(2 points)***

Now, we modify `salary4`. Create a new variable called "YrsEmployed" which reports the number of full years the employee has worked at UNC. Assume that all employees are hired January 1. Create a new dataset named `salary5` and again use `str()` to display the variables in `salary5`. (Use 2020 to create `YrsEmployed`)

```
#
salary5 <- mutate(salary4, YrsEmployed = 2020 - HireYear)
str(salary5)
```

```
## tibble [12,646 x 8] (S3: tbl_df/tbl/data.frame)
##  $ Name       : chr [1:12646] "AACHOUI, YOUSSEF" "AARNIO, REA T" "ABAJAS, YASMINA L" "ABARBANELL, JE
##  $ Department : chr [1:12646] "Microbiology and Immunology" "SW-Research Projects" "Peds-Hematology/O
##  $ Job        : chr [1:12646] "Research Professional, Medical" "Functional Paraprofessional" "Assista
##  $ hiredate   : num [1:12646] 20111010 20130114 20150701 19990101 20110912 ...
##  $ Exempt     : chr [1:12646] "Exempt from Personnel Act" "Subject to State Personnel Act" "Exempt f
```

```
##  $ Salary     : num [1:12646] 49128 33257 139405 181000 41098 ...
##  $ HireYear   : num [1:12646] 2011 2013 2015 1999 2011 ...
##  $ YrsEmployed: num [1:12646] 9 7 5 21 9 11 8 4 15 4 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   Name = col_character(),
##   ..   campus2 = col_character(),
##   ..   dept = col_character(),
##   ..   position = col_character(),
##   ..   PRIMARY_WORKING_TITLE = col_character(),
##   ..   hiredate = col_character(),
##   ..   exempt = col_character(),
##   ..   fte = col_double(),
##   ..   employed = col_double(),
##   ..   statesal = col_logical(),
##   ..   nonstsal = col_logical(),
##   ..   totalsal = col_double(),
##   ..   stservyr = col_double()
##   .. )
```

Q5 ***(4 points)***

Now, we modify `salary5` to create our final dataset named `salary.final`. Use the pipe `%>%` to make the following changes:

- Drop the variables "hiredate" and "HireYear".
- Sort the observations first by "Department" and then by "YrsEmployed".
- Rearrange the variables so that "YrsEmployed" and "Salary" are the first two variables in the dataset, in that order, without removing any of the other variables.

After you have used the `%>%` to make these changes, use the function `head()` to display the first 10 rows of `salary.final`.

```
#
salary.final = salary5 %>%
  select(-c(hiredate, HireYear)) %>%
  arrange(Department, YrsEmployed) %>%
  select(YrsEmployed, Salary, everything())
head(salary.final, 10)
```

```
## # A tibble: 10 x 6
##    YrsEmployed Salary Name       Department      Job          Exempt
##          <dbl>  <dbl> <chr>      <chr>           <chr>        <chr>
##  1           3  39646 DALEY, JO~ A and S - Busi~ Fiscal Affa~ Subject to S~
##  2           3  48814 WEBSTER, ~ A and S - Busi~ HR Coordina~ Subject to S~
##  3           3  48814 WOODSON, ~ A and S - Busi~ HR Coordina~ Subject to S~
##  4           3  48814 WORTHEN, ~ A and S - Busi~ HR Coordina~ Subject to S~
##  5           4  47164 CHESTER, ~ A and S - Busi~ HR Coordina~ Subject to S~
##  6           4  47983 GIBSON, J~ A and S - Busi~ Fiscal Affa~ Subject to S~
##  7           4  39646 RAUSCHER,~ A and S - Busi~ Fiscal Affa~ Subject to S~
##  8           4  39646 STRINGFEL~ A and S - Busi~ Fiscal Affa~ Subject to S~
##  9           5  48814 WATSON, S~ A and S - Busi~ HR Coordina~ Subject to S~
## 10           5  47983 YOUSEF, H~ A and S - Busi~ Fiscal Affa~ Subject to S~
```

Part 2: Answering Questions Based on All Data

Q6 *(2 Points)*

What is the average salary of employees in the Law Department?

Code *(1 Point)*:

```
#
Law_dept <- filter(salary.final, Department == "Law")
Avg_salary_law <- mean(Law_dept$Salary)
Avg_salary_law
```

```
## [1] 112567.1
```

Answer *(1 Point)*: (Place Answer Here Using Complete Sentences)

The average salary of employees in the Law Department is 112567.1 USD.

Q7 *(4 Points)*

How many employees have worked in Family Medicine between 5 and 8 years (inclusive) and are exempt from personnel act?

Code *(2 Points)*:

```
#
Family_Med <- filter(salary.final, Department == "Family Medicine",
  YrsEmployed %in% 5:8, Exempt == "Exempt from Personnel Act")
dim(Family_Med)
```

```
## [1] 16  6
```

Answer *(2 Points)*: (Place Answer Here Using Complete Sentences)

16 employees work in Family Medicine between 5 and 8 years, and are exempt from personnel act.

Q8 *(4 Points)*

What is the mean salary of employees from the Linguistics department who are professors, associate professors, or assistant professors?

Code *(2 Points)*:

```
#
Linguistics <- filter(salary.final, Department == "Linguistics",
  Job %in% c("Professor", "Associate Professor", "Assistant Professor"))
mean_salary_linguistics <- mean(Linguistics$Salary)
mean_salary_linguistics
```

```
## [1] 79935.17
```

Answer *(2 Points)*: (Place Answer Here Using Complete Sentences)

The mean salary of employees from the Linguistics department who are professors associate professors, or assistant professors is 79935.17.

Part 3: Answering Questions Based on Summarized Data

Q9 *(4 Points)*

Based off the data in `salary.final`, create a grouped summary based off combinations of "Department" and "YrsEmployed". Call the new tibble `deptyear_summary`. Your summarized tibble, `deptyear_summary`, should report all of the following statistics with corresponding variable names in the following order.

- "n" = number of employees for each combination
- "mean" = average salary for each combination
- "sd" = standard deviation of salary for each combination.
- "min" = minimum salary for each combination.
- "max" = maximum salary for each combination

In the process, make sure you use `ungroup()` with the pipe `%>%` to release the grouping so future work is no longer group specific. Following the creation of `deptyear_summary`, prove that your code worked by using `head()` to view the first 10 rows.

```
#
deptyear_summary <- salary.final %>%
  group_by(Department, YrsEmployed) %>%
  select(Department, YrsEmployed, Salary, everything()) %>%
  summarise(n=n(), mean = mean(Salary), sd = sd(Salary),
            min = min(Salary), max = max(Salary), .groups = 'drop')
head(deptyear_summary, 10)
```

```
## # A tibble: 10 x 7
##    Department                YrsEmployed     n   mean     sd   min   max
##    <chr>                           <dbl> <int>  <dbl>  <dbl> <dbl> <dbl>
##  1 A and S - Business Center           3     4 46522   4584  39646 48814
##  2 A and S - Business Center           4     4 43610.  4589. 39646 47983
##  3 A and S - Business Center           5     2 48398.   588. 47983 48814
##  4 A and S - Business Center           6     2 52190.  2703. 50278 54101
##  5 A and S - Business Center           7     2 54488   9199. 47983 60993
##  6 Acad Initiatives-UBC                4     1 23250     NA  23250 23250
##  7 Acad Initiatives-UBC                6     1 48782     NA  48782 48782
##  8 Acad Initiatives-UBC                9     1 60341     NA  60341 60341
##  9 Acad Initiatives-UBC               10     1 54851     NA  54851 54851
## 10 Acad Initiatives-UBC               17     2 64916  12875. 55812 74020
```

Q10 *(4 Points)*

Using the summarized data in `deptyear_summary`, use the `dplyr` functions to identify the 3 departments that award the lowest average salary for employees who have been employed for 3 years. The output should only show the 3 departments along with the corresponding years employeed, which should all be 3, and the four summarizing statistics created.

Furthermore, explain why the standard deviations for the 3 departments in your list have salary standard deviations of `NA`. What does this mean and how did it occur?

Code *(2 Points)*:

```
#
a = deptyear_summary %>%
  filter(YrsEmployed == 3) %>%
  arrange(mean)
head(a, 3)
```

```
## # A tibble: 3 x 7
##   Department                      YrsEmployed     n  mean    sd   min   max
##   <chr>                                 <dbl> <int> <dbl> <dbl> <dbl> <dbl>
## 1 Religious Studies                         3     1 16852    NA 16852 16852
## 2 Ath Olympic Sport Administratn            3     1 19276    NA 19276 19276
## 3 Jewish Studies                            3     1 19750    NA 19750 19750
```

Answer *(2 Points)*: (Place Answer Here Using Complete Sentences)

The 3 departments with the lowest average salary for employees who have been employed for 3 years are Religious Studies, Athletic Olympic Sport Adminstration, and Jewish studies. The standard deviation for salary is NA because there is only one entry – n equals 1. Therefore, the standard deviation does not exist for these departments.

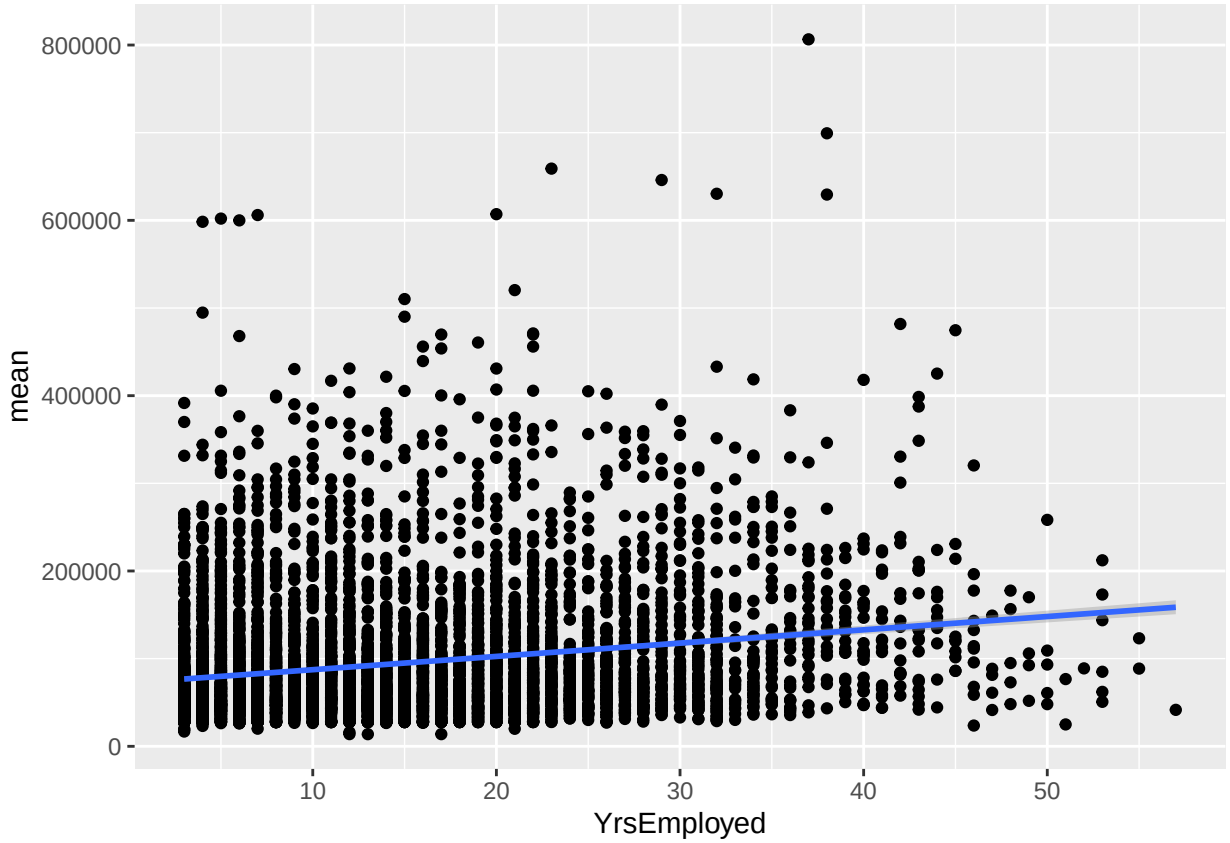
Q11 *(4 points)*

Create a scatter plot using `geom_point()` along with fitted lines using `geom_smooth` with the argument `method="lm"` showing the linear relationship between average salary and the years employeed. For this plot, use the summarized data in `deptyear_summary`. Following the plot, please explain what this plot suggests about the relationship between the salary a UNC employee makes and how many years that employee has served. Make reference to the figure and use descriptive adjectives (i.e. "strong", "weak", etc.) and terms (i.e. "positive", "negative", etc.) that are appropriate for discussing linear relationships.

Code and Figure *(2 Points)*:

```
#
ggplot(data = deptyear_summary, mapping = aes(x = YrsEmployed, y = mean)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Answer *(2 Points)*: (Place Answer Here Using Complete Sentences)

There is a weak positive correlation between years employed, and the mean salary of a department. Though the linear regression line shows a positively sloped line, there are many datapoints both above and lower than the linear regression line. Hence, the correlation is extremely weak, borderline useless.

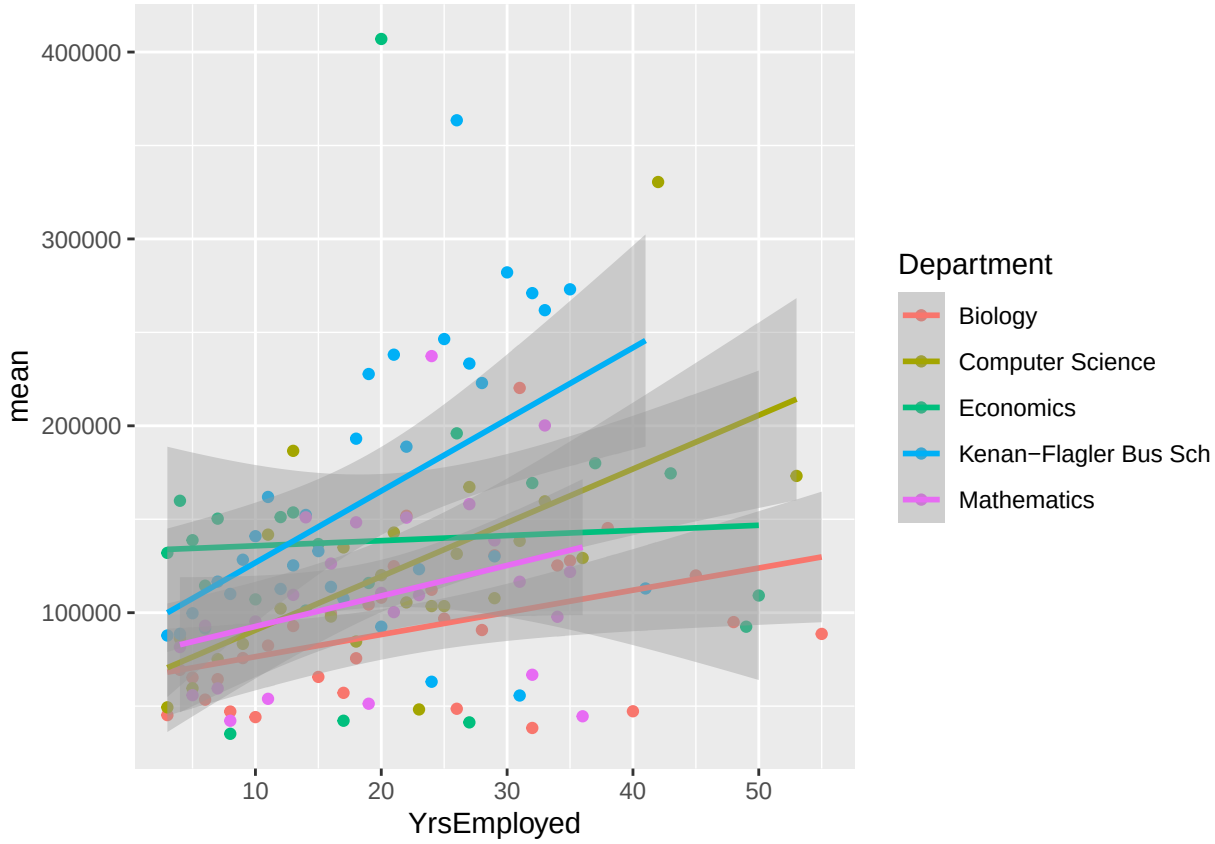
Q12 *(6 Points)*

The purpose of summarizing the data was to analyze the previously discussed linear relationship by group. In `deptyear_summary`, there are 702 unique departments represented. You can verify this by using `length(unique(deptyear_summary$Department))`. In this part, I want you to select 5 academic departments, not previously discussed, and in one figure, display the scatter plots and fitted regression lines representing the relationship between average salary and years employed in 5 different colors. Then, in complete sentences, I want you to state what departments you chose and explain the differences and/or similarities between the groups regarding the previously mentioned relationship. Compare departments on the starting salary and the rate of increase in salary based on the fitted lines.

Code and Figure: *(3 Points)*:

```
# I will use the Computer Science, Math, Biology,
# Kenan Flagler Business School, and Economics departments.
a = deptyear_summary %>%
  filter(Department %in% c("Kenan-Flagler Bus Sch", "Biology",
"Computer Science", "Mathematics", "Economics"))
ggplot(data = a, mapping = aes(x = YrsEmployed, y = mean, color = Department)) +
  geom_point() +
  geom_smooth(method = lm)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Answer *(3 Points)*: (Place Answer Here Using Complete Sentences)

In general, there is a positive correlation between years employed and the average salary of a department. What differes from each deparment is the strenght of the correlation. For instance, departments like economics and biology tend to have flatter linear regression lines, signalling a weaker but positive correlation between years employed and average salary. In contrast, the business school, mathematics, and computer science departments have shown stronger increases of average income as the number of employed years increases.

The starting incomes also significantly vary. The highest average starting salary is for economics, followed by business, mathematics, computer science, and biology. However, as noted above, average salary for some departments experience more significant increases over time.

One thing to note is that the graph left out various outliers, such as various datapoints which mean income exceed the limit shown on the y axis. It is unknown how many datapoints are left off the graph. However, since the outliers are mostly above the range of the confidence interval of the income, we can assume that the correlation is stronger than shown on the graph.