Analysis 1: UNC Salaries

Sunghwu Song

February 21, 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 file.

Submission: Submit via an electronic document on Gradescope. Must be submitted as an 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"*. If you scan the spreadsheet, you will notice that Dr. Mario is not listed. 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")
```

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"               "dept"
##  [4] "position"             "PRIMARY_WORKING_TITLE" "hiredate"
##  [7] "exempt"               "fte"                   "employed"
## [10] "statesal"             "nonstsal"              "totalsal"
## [13] "stservyr"
```

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

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

```
## tibble [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)
```

```
## Warning: `as.tibble()` is deprecated as of tibble 2.0.0.
## Please use `as_tibble()` instead.
## The signature and semantics have changed, see `?as_tibble`.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

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 employed
##   <chr> <chr>   <chr> <chr>    <chr>               <dbl> <chr>  <dbl>    <dbl>
## 1 AACH~ UNC-CH~ Micr~ Researc~ Research Associ~ 20111010 Exemp~   1         12
## 2 AARN~ UNC-CH~ SW-R~ Functio~ Graphic Designer 20130114 Subje~   0.8       12
```

```
## 3 ABAJ~ UNC-CH~ Peds~ Assista~ NODESCR          20150701 Exemp~   1         12
## 4 ABAR~ UNC-CH~ Kena~ Associa~ Associate Profe~ 19990101 Exemp~   1          9
## 5 ABAR~ UNC-CH~ Inst~ Researc~ Research Techni~ 20110912 Subje~   1         12
## # ... with 4 more variables: 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, JEF
##  $ 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 fr
##  $ 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`.

```
salary5 <- mutate(salary4, YrsEmployed = max(HireYear)-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/
##  $ Job        : chr [1:12646] "Research Professional, Medical" "Functional Paraprofessional" "Assist
##  $ 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] 6 4 2 18 6 8 5 1 12 1 ...
##  - 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(-hiredate, -HireYear) %>%
  arrange(YrsEmployed) %>%
  arrange(Department) %>%
  select(YrsEmployed, Salary, Name, Department, Job, Exempt)
head(salary.final, 10)
```

```
## # A tibble: 10 x 6
##    YrsEmployed Salary Name        Department       Job            Exempt
##          <dbl>  <dbl> <chr>       <chr>            <chr>          <chr>
##  1           0  39646 DALEY, JOS~ A and S - Busin~ Fiscal Affair~ Subject to St~
##  2           0  48814 WEBSTER, C~ A and S - Busin~ HR Coordinator Subject to St~
##  3           0  48814 WOODSON, K~ A and S - Busin~ HR Coordinator Subject to St~
##  4           0  48814 WORTHEN, T~ A and S - Busin~ HR Coordinator Subject to St~
##  5           1  47164 CHESTER, A~ A and S - Busin~ HR Coordinator Subject to St~
##  6           1  47983 GIBSON, JE~ A and S - Busin~ Fiscal Affair~ Subject to St~
##  7           1  39646 RAUSCHER, ~ A and S - Busin~ Fiscal Affair~ Subject to St~
##  8           1  39646 STRINGFELL~ A and S - Busin~ Fiscal Affair~ Subject to St~
##  9           2  48814 WATSON, ST~ A and S - Busin~ HR Coordinator Subject to St~
## 10           2  47983 YOUSEF, HE~ A and S - Busin~ Fiscal Affair~ Subject to St~
```

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)*:

```
salary.final %>%
  filter(Department=="Law") %>%
  select(Salary) -> salaries
  mean(salaries$Salary)
```

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

Answer *(1 Point)*: The average salary of employees in the Law Department is $112567.10

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)*:

```
salary.final %>%
  filter(Department == "Family Medicine") %>%
  filter(between(YrsEmployed, 5, 8)) %>%
  filter(Exempt == "Exempt from Personnel Act")
```

```
## # A tibble: 10 x 6
##    YrsEmployed  Salary Name       Department   Job                   Exempt
##          <dbl>   <dbl> <chr>      <chr>        <chr>                 <chr>
##  1           5 128550. FARAHI, N~ Family Medi~ Assistant Professor   Exempt fro~
##  2           5 162235  RAYALA, B~ Family Medi~ Associate Professor   Exempt fro~
##  3           5  44822. SILVER, B~ Family Medi~ Student Counseling /~ Exempt fro~
##  4           6 139527  HOUSE, LA~ Family Medi~ Assistant Professor   Exempt fro~
##  5           7 160200  CRITES, S~ Family Medi~ Assistant Professor   Exempt fro~
##  6           7 153837  FEDORIW, ~ Family Medi~ Associate Professor   Exempt fro~
##  7           7 142633. KISTLER, ~ Family Medi~ Assistant Professor   Exempt fro~
##  8           7 197325  STEINER, ~ Family Medi~ Professor             Exempt fro~
##  9           7  62940  TROUT, SU~ Family Medi~ Student Counseling /~ Exempt fro~
## 10           8 155976. BECKER-DR~ Family Medi~ Associate Professor   Exempt fro~
```

Answer *(2 Points)*: There are 10 employees that have worked in Family Medicine between 5 and 8 years (inclusive) 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)*:

```
salary.final %>%
  filter(Department == "Linguistics") %>%
  filter(Job == "Professor" | Job == "Associate Professor" | Job == "Assitant Professor") %>%
  summarise(meanSalary = mean(Salary))
```

```
## # A tibble: 1 x 1
##   meanSalary
##        <dbl>
## 1     79935.
```

Answer *(2 Points)*: 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.

```
salary.final %>%
  group_by(Department, YrsEmployed) %>%
  summarize("n" = n(),
            "mean" = mean(Salary),
            "sd" = sd(Salary),
            "min" = min(Salary),
            "max" = max(Salary)
            ) -> deptyear_summary
```

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

```
deptyear_summary %>%
  ungroup() %>%
  head(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           0     4 46522   4584  39646 48814
##  2 A and S - Business Center           1     4 43610.  4589. 39646 47983
##  3 A and S - Business Center           2     2 48398.   588. 47983 48814
##  4 A and S - Business Center           3     2 52190.  2703. 50278 54101
```

```
##  5 A and S - Business Center           4     2 54488   9199. 47983 60993
##  6 Acad Initiatives-UBC                1     1 23250     NA  23250 23250
##  7 Acad Initiatives-UBC                3     1 48782     NA  48782 48782
##  8 Acad Initiatives-UBC                6     1 60341     NA  60341 60341
##  9 Acad Initiatives-UBC                7     1 54851     NA  54851 54851
## 10 Acad Initiatives-UBC               14     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 deviation for one of the 3 departments in your list has a salary standard deviation of "NaN". What does this mean and how did it occur?

Code *(2 Points)*:

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

```
## # A tibble: 3 x 7
## # Groups:   Department [3]
##   Department                     YrsEmployed     n   mean    sd    min    max
##   <chr>                                <dbl> <int>  <dbl> <dbl>  <dbl>  <dbl>
## 1 TEACCH - Supported Employment            3     2 26041. 5261. 22321  29761.
## 2 DPS Security                             3     3 26569. 6896. 18627. 31047.
## 3 FS-Housekeeping Svcs-Zone 13             3     1 26680.   NA  26680. 26680.
```

Answer *(2 Points)*: The 3 departments that award the lowerst average salary for employees who have been employed for 3 years are: TEACCH - Supported Employment, DPS Security, FS-Housekeeping Svcs-Zone 13. There cannot be any standard deviation for FS-Housekeeping Svcs-Zone 13 because there is only one instance. n needs to be greater than 1 for standard deviation to exist.

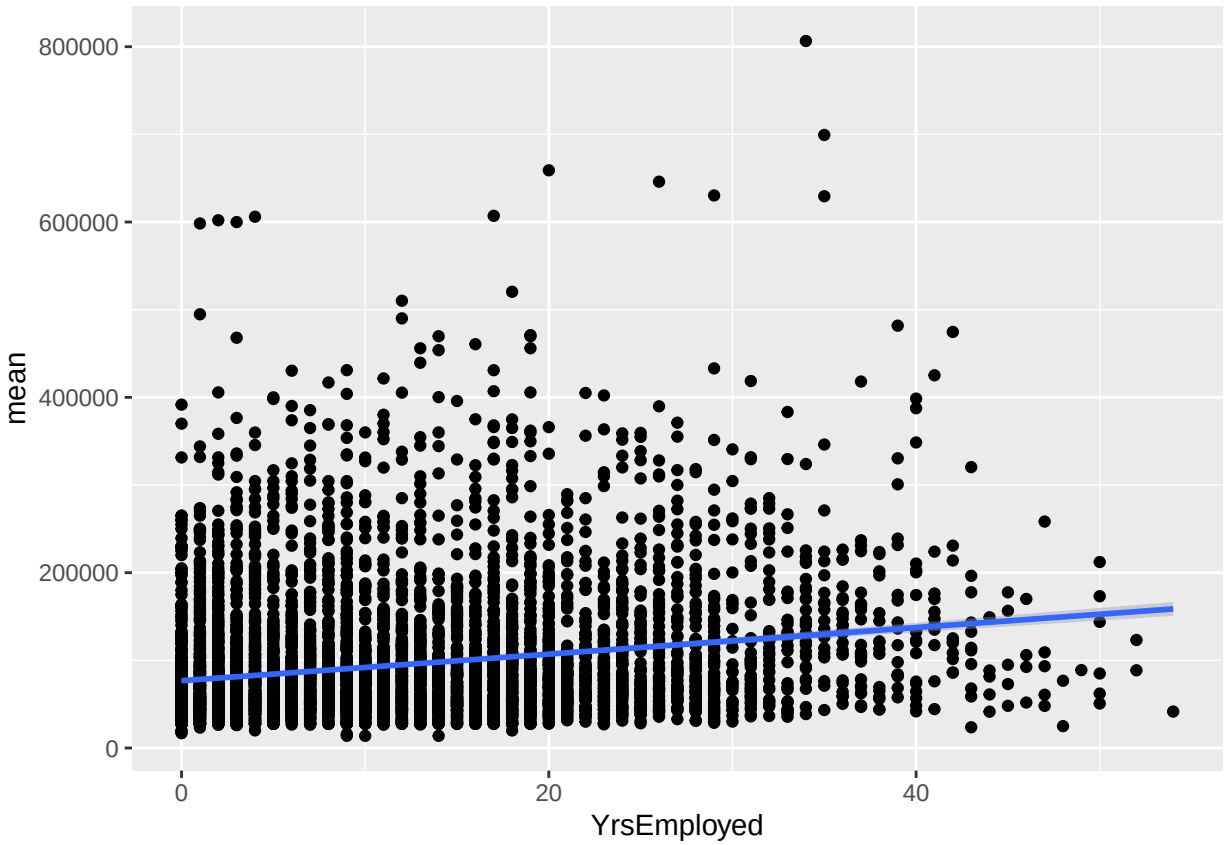
Q11 ***(4 points)***

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) + geom_point(mapping = aes(x=YrsEmployed, y=mean)) + geom_smooth(mapping
```

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



Answer *(2 Points)*: The graph shows a strong and positive relationship between years employeed and mean salary because of the small confidence interval and positive slope.

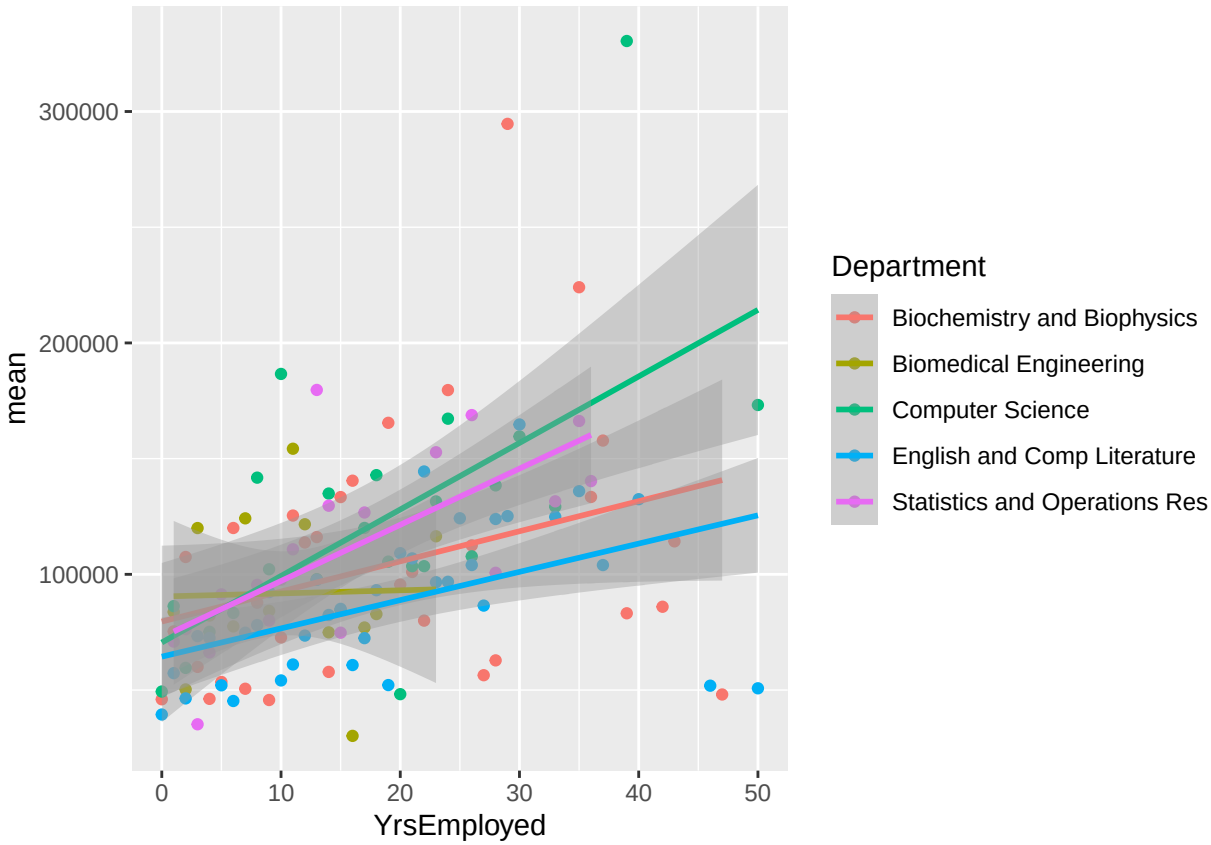
Q12 ***(6 Points)***

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)*:

```
deptyear_summary %>%
  filter(Department %in% c("Computer Science", "Biomedical Engineering", "Biochemistry and Biophysics",
  ggplot(aes(x = YrsEmployed, y = mean, color = Department)) +
  geom_point() +
  geom_smooth(method = 'lm')
```

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



Answer *(3 Points)*: I chose departments by what majors my friends and I were. My friends and I are biochemistry, biomedical engineering, computer science, and statistics majors, and I decided to add the English department to see how a humanities major might be different from various STEM majors in the salary over years employed. Interestingly enough, biomedical engineering does not have any representation for after 25 years employed, and statistics does not have any representation for after 40 years employed. Additionally, biomedical engineering salaries do not seem to increase or decrease over years employed. In terms of starting salary, biomedical engineering is first, then follows biochemistry, statistics, computer science, then english. In terms of rate of increase in salary, computer science is the highest posible salary after 10 years for all departments, and statistics, biomedical engineering, biochemistry, and english follow after, respectively. For years after 10 years, the growth stays similar and the ranks do not change except for English eventually surpassing biomedical engineering after 20 years. But again, this is probably because of the lack of data in BMME departments who have been employed for longer than 25 years.