Competitive Data Science Salaries

Amanda Frithsen

Proposed Question: Your CEO has decided that the company needs a full-time scientist, and possibly a team of them in the future. She thinks she needs someone who can help drive data science within the entire organization and could potentially lead a team in the future. She understands that data scientist salaries vary widely across the world and is unsure what to pay them. To complicate matters, salaries are going up due to the great recession and the market is highly competitive. Your CEO has asked you to prepare an analysis on data science salaries and provide them with a range to be competitive and get top talent. The position can work offshore, but the CEO would like to know what the difference is for a person working in the United States. Your company is currently a small company but is expanding rapidly.

Alternative ways to ask proposed question: * What is a competitive data scientist salary range for the company to offer? * How have salaries gone up due to recession in recent years? * How do data scientist salaries vary across the world?
* How do competitive data scientist salaries in the US differ from elsewhere in the world? * What is a typical, competitive salary for a top-talented data scientist? - How should we consider other factors - remote, company size, experience_level, job title? Could these factors be used as negotiation points for salary? * What salary is "top talent" being offered? What job titles would you look for on a resume to ensure "top talent"?

```
library(dplyr)
```

```
##
## 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
```

```
library(readr)
library(ggplot2)
library(sf)
```

```
## Linking to GEOS 3.12.1, GDAL 3.8.4, PROJ 9.3.1; sf_use_s2() is TRUE
```

```
library(rnaturalearth)
library(rnaturalearthdata)
```

```
##
## Attaching package: 'rnaturalearthdata'
```

```
## The following object is masked from 'package:rnaturalearth':
##
##     countries110
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
# Load data into data frame
library(readr)
salaries <- read_csv("r project data.csv")
```

```
## New names:
## Rows: 607 Columns: 12
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (7): experience_level, employment_type, job_title, salary_currency, empl... dbl
## (5): ...1, work_year, salary, salary_in_usd, remote_ratio
## 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.
## * `` -> `...1`
```

```
# examine data frame
head(salaries)
```

```
## # A tibble: 6 x 12
##    ...1 work_year experience_level employment_type job_title              salary
##   <dbl>     <dbl> <chr>            <chr>           <chr>                   <dbl>
## 1     0      2020 MI               FT              Data Scientist          70000
## 2     1      2020 SE               FT              Machine Learning Scie~ 260000
## 3     2      2020 SE               FT              Big Data Engineer       85000
## 4     3      2020 MI               FT              Product Data Analyst    20000
## 5     4      2020 SE               FT              Machine Learning Engi~ 150000
## 6     5      2020 EN               FT              Data Analyst            72000
## # i 6 more variables: salary_currency <chr>, salary_in_usd <dbl>,
## #   employee_residence <chr>, remote_ratio <dbl>, company_location <chr>,
## #   company_size <chr>
```

```
tail(salaries)
```

```
## # A tibble: 6 x 12
##    ...1 work_year experience_level employment_type job_title     salary
##   <dbl>     <dbl> <chr>            <chr>           <chr>          <dbl>
```

```
## 1   601      2022 EN              FT             Data Analyst   52000
## 2   602      2022 SE              FT             Data Engineer 154000
## 3   603      2022 SE              FT             Data Engineer 126000
## 4   604      2022 SE              FT             Data Analyst  129000
## 5   605      2022 SE              FT             Data Analyst  150000
## 6   606      2022 MI              FT             AI Scientist  200000
## # i 6 more variables: salary_currency <chr>, salary_in_usd <dbl>,
## #   employee_residence <chr>, remote_ratio <dbl>, company_location <chr>,
## #   company_size <chr>
```

```r
str(salaries)
```

```
## spc_tbl_ [607 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1              : num [1:607] 0 1 2 3 4 5 6 7 8 9 ...
##  $ work_year         : num [1:607] 2020 2020 2020 2020 2020 2020 2020 2020 2020 2020 ...
##  $ experience_level  : chr [1:607] "MI" "SE" "SE" "MI" ...
##  $ employment_type   : chr [1:607] "FT" "FT" "FT" "FT" ...
##  $ job_title         : chr [1:607] "Data Scientist" "Machine Learning Scientist" "Big Data Engineer"
##  $ salary            : num [1:607] 70000 260000 85000 20000 150000 72000 190000 11000000 135000 1250
##  $ salary_currency   : chr [1:607] "EUR" "USD" "GBP" "USD" ...
##  $ salary_in_usd     : num [1:607] 79833 260000 109024 20000 150000 ...
##  $ employee_residence: chr [1:607] "DE" "JP" "GB" "HN" ...
##  $ remote_ratio      : num [1:607] 0 0 50 0 50 100 100 50 100 50 ...
##  $ company_location  : chr [1:607] "DE" "JP" "GB" "HN" ...
##  $ company_size      : chr [1:607] "L" "S" "M" "S" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   work_year = col_double(),
##   ..   experience_level = col_character(),
##   ..   employment_type = col_character(),
##   ..   job_title = col_character(),
##   ..   salary = col_double(),
##   ..   salary_currency = col_character(),
##   ..   salary_in_usd = col_double(),
##   ..   employee_residence = col_character(),
##   ..   remote_ratio = col_double(),
##   ..   company_location = col_character(),
##   ..   company_size = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
summary(salaries)
```

```
##       ...1         work_year    experience_level   employment_type
##  Min.   :  0.0   Min.   :2020   Length:607         Length:607
##  1st Qu.:151.5   1st Qu.:2021   Class :character   Class :character
##  Median :303.0   Median :2022   Mode  :character   Mode  :character
##  Mean   :303.0   Mean   :2021
##  3rd Qu.:454.5   3rd Qu.:2022
##  Max.   :606.0   Max.   :2022
##   job_title             salary         salary_currency    salary_in_usd
##  Length:607         Min.   :    4000   Length:607         Min.   :  2859
```

```
##  Class :character   1st Qu.:   70000   Class :character   1st Qu.: 62726
##  Mode  :character   Median :  115000   Mode  :character   Median :101570
##                     Mean   :  324000                      Mean   :112298
##                     3rd Qu.:  165000                      3rd Qu.:150000
##                     Max.   :30400000                      Max.   :600000
##  employee_residence  remote_ratio    company_location   company_size
##  Length:607         Min.   :  0.00   Length:607         Length:607
##  Class :character   1st Qu.: 50.00   Class :character   Class :character
##  Mode  :character   Median :100.00   Mode  :character   Mode  :character
##                     Mean   : 70.92
##                     3rd Qu.:100.00
##                     Max.   :100.00
```

```r
colnames(salaries)
```

```
##  [1] "...1"               "work_year"          "experience_level"
##  [4] "employment_type"    "job_title"          "salary"
##  [7] "salary_currency"    "salary_in_usd"      "employee_residence"
## [10] "remote_ratio"       "company_location"   "company_size"
```

Notes from examination of data frame: * 607 observations, 12 variables * 7 character variables: experience_level, employment_type, job_title, salary_currency, employee_residence, company_location, company_size 5 numeric (double) variables: ...1, work_year, salary, salary_in_usd, remote_ratio * ...1 column appears to be identification number for each employee, starting at 0 and ending at 606 - rename this column - start counting at 1 * work_years appear to only include 2020, 2021, 2022 (could be changed to factor) * experience_level, employment_type, salary_currency, company_size, remote_ratio could be possibly changed to factors * IQR of salaries_in_usd seems reasonable (62726 to 150000); potential for outliers on either side of data given values of min (2859) and max (600000) * employee_residence and company_location provide the ISO country codes

```r
# check for NA or missing values
sum(is.na(salaries))
```

```
## [1] 0
```

Based on the sum of the NA values in the salaries data frame, there are no NA values.

```r
# rename first column
colnames(salaries)[colnames(salaries) == "...1"] <- "ID_number"
colnames(salaries)
```

```
##  [1] "ID_number"          "work_year"          "experience_level"
##  [4] "employment_type"    "job_title"          "salary"
##  [7] "salary_currency"    "salary_in_usd"      "employee_residence"
## [10] "remote_ratio"       "company_location"   "company_size"
```

```r
# change ID numbers to go from 1 to 607
salaries$ID_number <- salaries$ID_number + 1

# check that the values now go from 1 to 607
min(salaries$ID_number)
```

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

```r
max(salaries$ID_number)
```

```
## [1] 607
```

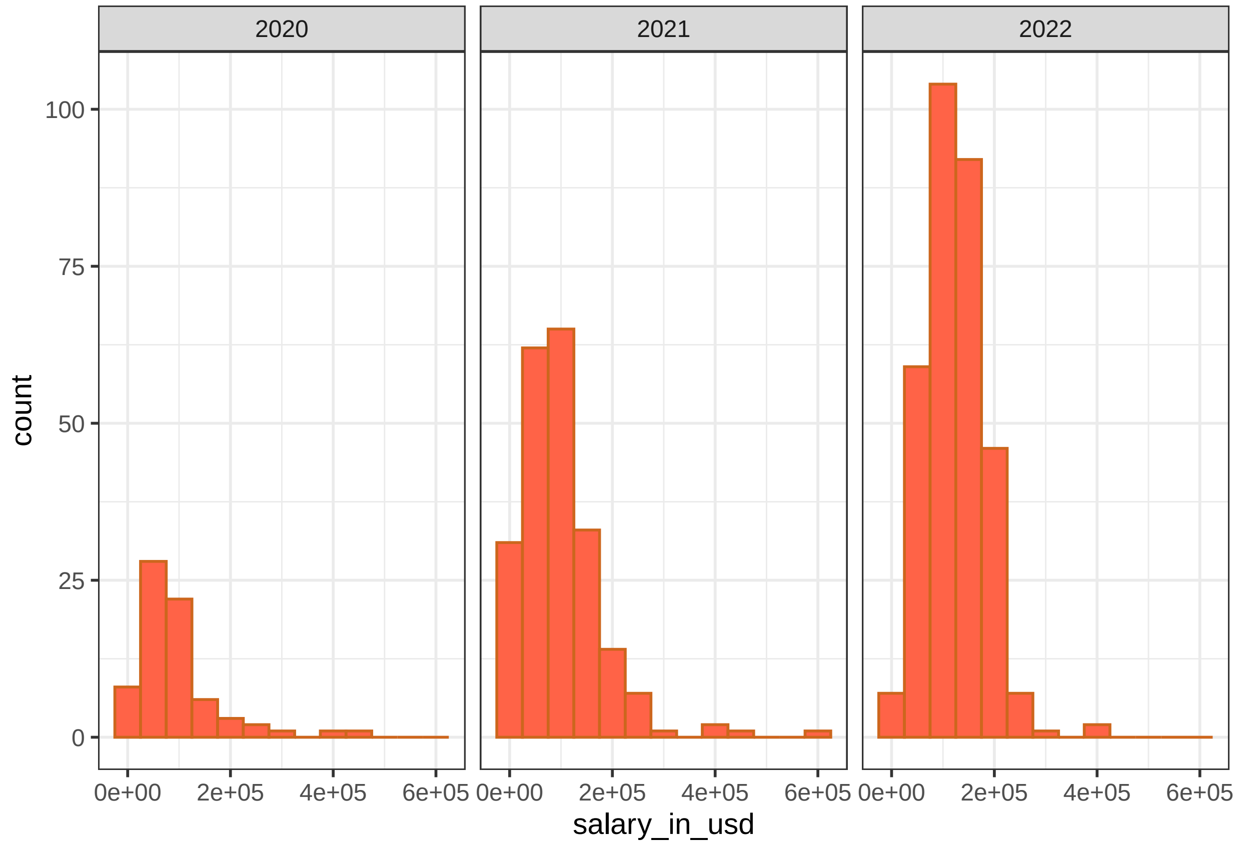
```r
# change variables to factors
salaries <- salaries %>%
  mutate(work_year = as.factor(work_year)) %>%
  mutate(experience_level =
           factor(experience_level, levels = c("EN", "MI", "SE", "EX"), ordered = TRUE)) %>%
  mutate(employment_type = as.factor(employment_type)) %>%
  mutate(salary_currency = as.factor(salary_currency)) %>%
  mutate(company_size = factor(company_size, levels = c("S", "M", "L"), ordered = TRUE)) %>%
  mutate(remote_ratio = as.factor(remote_ratio)) %>%
  mutate(company_location = as.factor(company_location))

str(salaries)
```

```
## tibble [607 x 12] (S3: tbl_df/tbl/data.frame)
##  $ ID_number         : num [1:607] 1 2 3 4 5 6 7 8 9 10 ...
##  $ work_year         : Factor w/ 3 levels "2020","2021",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ experience_level  : Ord.factor w/ 4 levels "EN"<"MI"<"SE"<..: 2 3 3 2 3 1 3 2 2 3 ...
##  $ employment_type   : Factor w/ 4 levels "CT","FL","FT",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ job_title         : chr [1:607] "Data Scientist" "Machine Learning Scientist" "Big Data Engineer"
##  $ salary            : num [1:607] 70000 260000 85000 20000 150000 72000 190000 11000000 135000 1250
##  $ salary_currency   : Factor w/ 17 levels "AUD","BRL","CAD",..: 8 17 9 17 17 17 17 10 17 17 ...
##  $ salary_in_usd     : num [1:607] 79833 260000 109024 20000 150000 ...
##  $ employee_residence: chr [1:607] "DE" "JP" "GB" "HN" ...
##  $ remote_ratio      : Factor w/ 3 levels "0","50","100": 1 1 2 1 2 3 3 2 3 2 ...
##  $ company_location  : Factor w/ 50 levels "AE","AS","AT",..: 13 30 19 21 49 49 49 23 49 39 ...
##  $ company_size      : Ord.factor w/ 3 levels "S"<"M"<"L": 3 1 2 1 3 3 1 3 3 1 ...
```

Additional thoughts about variables… * work_year could provide insight about how salaries are changing in recent years (histogram of salaries faceted by work_year could be a good way to see this) * Is there a relationship between job title and work_experience? employment_type? salary? * How do work_experience and employment_type impact salary? Perhaps look at summary tables to compare these variables * salary, salary_currency, and salary_in_usd all related; given our company is in the us, most helpful to consider salary_in_usd * interested to see if remote_ratio impacts salary…could ability to work remotely be a "perk" to offset salary? consider looking at box plots faceted by remote_ratio * how do salaries vary based on location? look at company_location…use world map to visualize * how does company_size impact other variables (salary, experience_level, employment_type, etc.)? look at the company_size of more experienced employees perhaps by using bar chart

First, I would like to examine salaries during each value of work_year (2020, 2021, 2022), to see how significant the change is over the course of the three years.

```r
# Create a histogram of the salaries faceted by work year
ggplot(salaries,
       aes(x = salary_in_usd))+
  geom_histogram(binwidth = 50000, fill = "Tomato", color = "Chocolate 3") +
  # after experimenting with different values of binwidth, $50,000 seemed to show the most detail witho
  theme_bw()+
  facet_grid(~work_year)
```




```
# upon examination, the histogram appears to show that 2022 has the most data values
salaries %>% count(work_year)
```

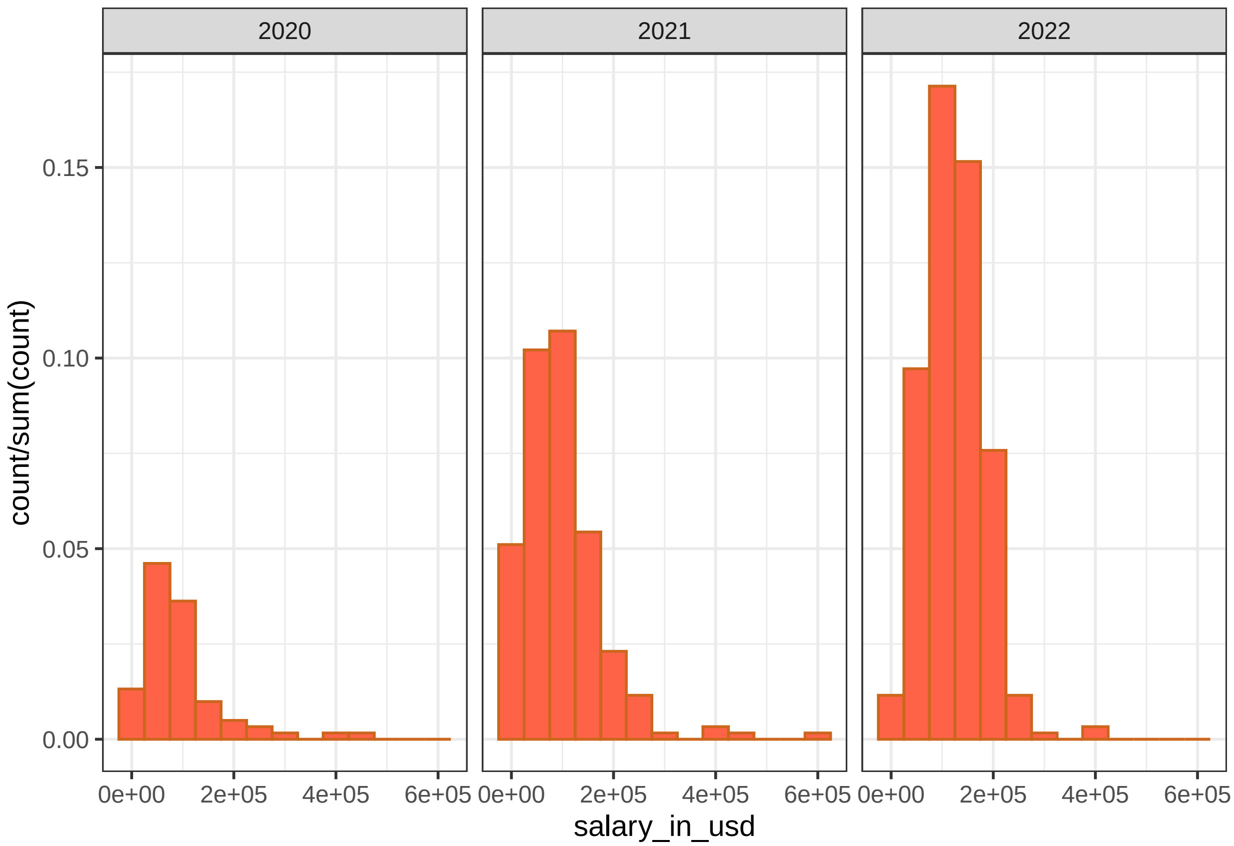
```
## # A tibble: 3 x 2
##   work_year     n
##   <fct>     <int>
## 1 2020         72
## 2 2021        217
## 3 2022        318
```

All three years show data that is unimodal and right skewed. The center of the data does appear to increase over the three years. After looking at the counts and noticing how much smaller the sample space from 2020 is compared to the other two years (with only 72 observations), it would be more helpful to look at a plot comparing the relative frequencies of the values.

```
# Create a relative frequency histogram of the salaries faceted by work year
ggplot(salaries,
       aes(x = salary_in_usd))+
  geom_histogram(aes(y = ..count../sum(..count..)), binwidth = 50000, fill = "Tomato", color = "Chocola
  theme_bw()+
  facet_grid(~work_year)
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
## This warning is displayed once every 8 hours.
```

```
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



When trying to make a relative frequency histogram, I hit a wall when the relative frequencies were calculated using the overall total (607) instead of the total for each facet. The best way I could find to rectify this was to group the data by year and bin in order to calculate the percents and then create a bar chart.

```r
# find the number of bins
# round used to ensure 23.88..rounded up to the nearest whole number
number_bins <- round((max(salaries$salary_in_usd) - min(salaries$salary_in_usd))/25000, digits = 0)
number_bins
```

```
## [1] 24
```

```r
min(salaries$salary_in_usd)
```

```
## [1] 2859
```

```r
max(salaries$salary_in_usd)
```

```
## [1] 6e+05
```

```r
salaries_by_year <- salaries %>%
  # group to find the counts per year
  group_by(work_year) %>%
  mutate(count_by_year = n()) %>%
  ungroup()

head(salaries_by_year)
```

```
## # A tibble: 6 x 13
##   ID_number work_year experience_level employment_type job_title          salary
##       <dbl> <fct>     <ord>            <fct>           <chr>               <dbl>
## 1         1 2020      MI               FT              Data Scientist      70000
## 2         2 2020      SE               FT              Machine Learning ~ 260000
## 3         3 2020      SE               FT              Big Data Engineer   85000
## 4         4 2020      MI               FT              Product Data Anal~  20000
## 5         5 2020      SE               FT              Machine Learning ~ 150000
## 6         6 2020      EN               FT              Data Analyst        72000
## # i 7 more variables: salary_currency <fct>, salary_in_usd <dbl>,
## #   employee_residence <chr>, remote_ratio <fct>, company_location <fct>,
## #   company_size <ord>, count_by_year <int>
```

```r
salaries_by_year <- salaries_by_year %>%
  # count number of observations within each bin
  group_by(work_year, count_by_year, bin = cut(salary_in_usd, seq(0, 600000, by = 50000))) %>%
  # find the number of observations in each bin
  summarize(count_salaries = n()) %>%
  # percent of observations in each bin
  mutate(percent = count_salaries/count_by_year *100)
```

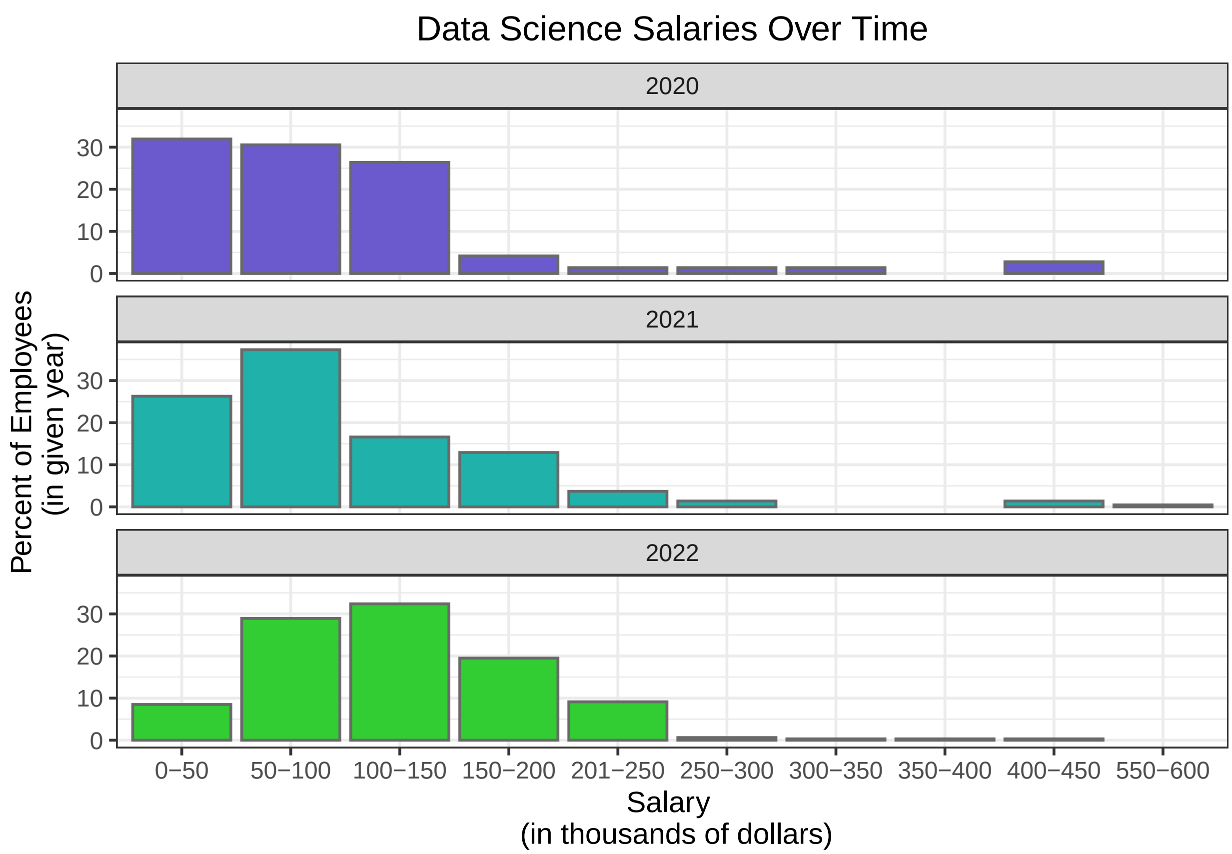
```
## `summarise()` has grouped output by 'work_year', 'count_by_year'. You can
## override using the `.groups` argument.
```

```r
unique(salaries_by_year$bin)
```

```
##  [1] (0,5e+04]       (5e+04,1e+05]   (1e+05,1.5e+05] (1.5e+05,2e+05]
##  [5] (2e+05,2.5e+05] (2.5e+05,3e+05] (3e+05,3.5e+05] (4e+05,4.5e+05]
##  [9] (5.5e+05,6e+05] (3.5e+05,4e+05]
## 12 Levels: (0,5e+04] (5e+04,1e+05] (1e+05,1.5e+05] ... (5.5e+05,6e+05]
```

```r
# vector of labels for x-axis to make values readable
x_labels <- c("(0,5e+04]" = "0-50",
              "(5e+04,1e+05]" = "50-100",
              "(1e+05,1.5e+05]" = "100-150",
              "(1.5e+05,2e+05]" = "150-200",
              "(2e+05,2.5e+05]" = "201-250",
              "(2.5e+05,3e+05]" = "250-300",
              "(3e+05,3.5e+05]" = "300-350",
              "(3.5e+05,4e+05]" = "350-400",
              "(4e+05,4.5e+05]" = "400-450",
              "(4.5e+05,5e+05]" = "450-500",
              "(5e+05,5.5e+05]" = "500-550",
```

```r
              "(5.5e+05,6e+05]" = "550-600")

# create bar chart using percents calculated with in each year
ggplot(salaries_by_year,
       aes(x = bin, y = percent, fill = work_year))+
  geom_col(color = 'Dim Gray')+
  facet_wrap(~work_year, ncol = 1) +
  theme_bw() +
  scale_fill_manual(values = c("2020" = "Slate Blue",
                               "2021" = "Light Sea Green",
                               "2022" = "Lime Green")) +
  scale_x_discrete(labels = x_labels) +
  labs(title = 'Data Science Salaries Over Time',
         x = "Salary \n (in thousands of dollars)",
         y = "Percent of Employees \n (in given year)") +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
```



After looking at the graphical representations, I would also like to look at some summary statistics within each year.

```r
# because I want to preserve the data frame created previously, I am going to use a new data frame to b
salaries_by_year_2 <- salaries %>%
  group_by(work_year) %>%
  summarize(Q1_salary_in_usd_by_year = quantile(salary_in_usd, 0.25),
            median_salary_in_usd_by_year = median(salary_in_usd),
            Q3_salary_in_usd_by_year = quantile(salary_in_usd, 0.75))
```

```r
salaries_by_year_2
```

```
## # A tibble: 3 x 4
##   work_year Q1_salary_in_usd_by_~1 median_salary_in_usd~2 Q3_salary_in_usd_by_~3
##   <fct>                      <dbl>                  <dbl>                  <dbl>
## 1 2020                      45724.                  75544                 115526
## 2 2021                      50000                   82528                 135000
## 3 2022                      81666                  120000                 160000
## # i abbreviated names: 1: Q1_salary_in_usd_by_year,
## #   2: median_salary_in_usd_by_year, 3: Q3_salary_in_usd_by_year
```

Next, I would like to examine average salaries compared to job_title, experience_level and employment_type. I will use summary tables to look at the mean and median.

```r
# average salary by job title
salaries_by_job_title <- salaries %>%
  group_by(job_title) %>%
  summarize(mean_salary_in_usd = mean(salary_in_usd),
            median_salary_in_usd = median(salary_in_usd))
salaries_by_job_title
```

```
## # A tibble: 50 x 3
##    job_title                          mean_salary_in_usd median_salary_in_usd
##    <chr>                                           <dbl>                <dbl>
##  1 3D Computer Vision Researcher                    5409                 5409
##  2 AI Scientist                                    66136.               45896
##  3 Analytics Engineer                             175000               179850
##  4 Applied Data Scientist                         175655               157000
##  5 Applied Machine Learning Scientist             142069.               56700
##  6 BI Data Analyst                                 74755.               76500
##  7 Big Data Architect                              99703                99703
##  8 Big Data Engineer                               51974                41306.
##  9 Business Data Analyst                           76691.               70912
## 10 Cloud Data Engineer                            124647               124647
## # i 40 more rows
```

```r
# filter to identify the job titles that reported the maximum and minimum salary
# I looked at both mean and median to ensure they were the same
filter(salaries_by_job_title, salaries_by_job_title$mean_salary_in_usd == max(salaries_by_job_title$mean
```

```
## # A tibble: 1 x 3
##   job_title           mean_salary_in_usd median_salary_in_usd
##   <chr>                            <dbl>                <dbl>
## 1 Data Analytics Lead             405000               405000
```

```r
filter(salaries_by_job_title, salaries_by_job_title$median_salary_in_usd == max(salaries_by_job_title$me
```

```
## # A tibble: 1 x 3
##   job_title           mean_salary_in_usd median_salary_in_usd
##   <chr>                            <dbl>                <dbl>
## 1 Data Analytics Lead             405000               405000
```

```r
filter(salaries_by_job_title, salaries_by_job_title$mean_salary_in_usd == min(salaries_by_job_title$mean
```

```
## # A tibble: 1 x 3
##   job_title                     mean_salary_in_usd median_salary_in_usd
##   <chr>                                      <dbl>                <dbl>
## 1 3D Computer Vision Researcher               5409                 5409
```

```r
filter(salaries_by_job_title, salaries_by_job_title$median_salary_in_usd == min(salaries_by_job_title$me
```

```
## # A tibble: 1 x 3
##   job_title                     mean_salary_in_usd median_salary_in_usd
##   <chr>                                      <dbl>                <dbl>
## 1 3D Computer Vision Researcher               5409                 5409
```

```r
# sort by median salary and create a table with the job titles that reported the top 3 salaries and  cr
# I thought looking at these jobs with descriptions may give an idea of the skills the company is "buyi
top_bottom_job_salaries <- arrange(salaries_by_job_title, median_salary_in_usd)

top_bottom_job_salaries <- top_bottom_job_salaries[c(1:3, 48:50),]
top_bottom_job_salaries
```

```
## # A tibble: 6 x 3
##   job_title                     mean_salary_in_usd median_salary_in_usd
##   <chr>                                      <dbl>                <dbl>
## 1 3D Computer Vision Researcher              5409                 5409
## 2 Product Data Analyst                      13036                13036
## 3 Computer Vision Engineer                  44419.               26304.
## 4 Principal Data Engineer                  328333.              200000
## 5 Financial Data Analyst                   275000               275000
## 6 Data Analytics Lead                      405000               405000
```

I wonder how much data each of these average salaries is based on?
Which data science jobs occur the most in our data set? How does experience vary depending on job title?

```r
# average salary by experience level
salaries_by_experience_level <- salaries %>%
  group_by(experience_level) %>%
  summarize(mean_salary_in_usd = mean(salary_in_usd),
            median_salary_in_usd = median(salary_in_usd))
salaries_by_experience_level
```

```
## # A tibble: 4 x 3
##   experience_level mean_salary_in_usd median_salary_in_usd
##   <ord>                         <dbl>                <dbl>
## 1 EN                           61643.                56500
## 2 MI                           87996.                76940
## 3 SE                          138617.               135500
## 4 EX                          199392.               171438.
```

```r
# After examining the differences in the mean and median salaries based on experience level, I would li

salaries_by_experience_level <- salaries %>%
  group_by(experience_level) %>%
  summarize(Q1_salary_in_usd = quantile(salary_in_usd, 0.25),
            median_salary_in_usd = median(salary_in_usd),
            Q3_salary_in_usd = quantile(salary_in_usd, 0.75))
salaries_by_experience_level
```

```
## # A tibble: 4 x 4
##   experience_level Q1_salary_in_usd median_salary_in_usd Q3_salary_in_usd
##   <ord>                       <dbl>                <dbl>            <dbl>
## 1 EN                          27505                56500           85426.
## 2 MI                          48000                76940          112000
## 3 SE                         100000               135500          170000
## 4 EX                         130006.              171438.         233750
```

```r
# average salary by employment type

salaries_by_employment_type <- salaries %>%
  group_by(employment_type) %>%
  summarize(mean_salary_in_usd = mean(salary_in_usd),
            median_salary_in_usd = median(salary_in_usd))
salaries_by_employment_type
```

```
## # A tibble: 4 x 3
##   employment_type mean_salary_in_usd median_salary_in_usd
##   <fct>                        <dbl>                <dbl>
## 1 CT                          184575               105000
## 2 FL                           48000                40000
## 3 FT                          113468.              104196.
## 4 PT                           33070.               18818.
```

```r
# experience within each job title
job_title_by_experience_level <- salaries %>%
  group_by(job_title, experience_level) %>%
  summarize(count = n())
```

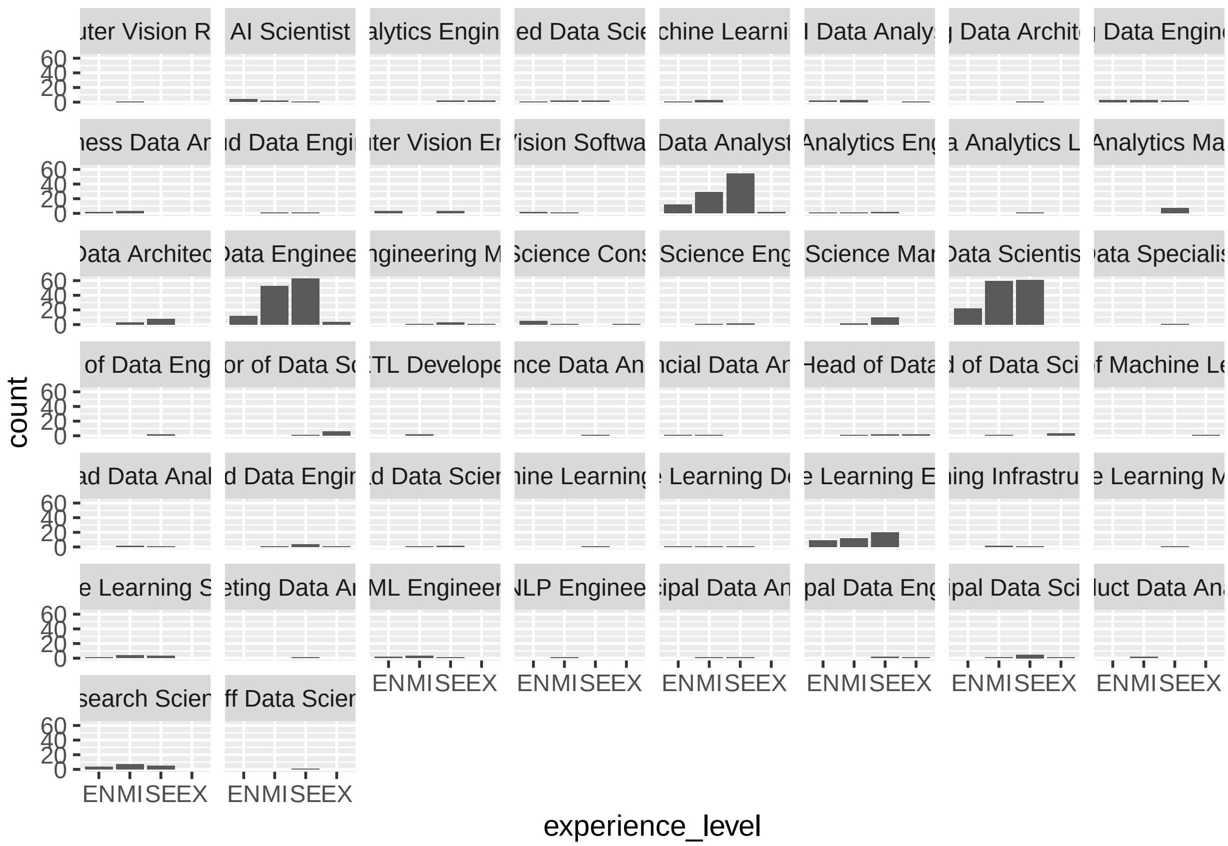
```
## `summarise()` has grouped output by 'job_title'. You can override using the
## `.groups` argument.
```

```r
job_title_by_experience_level
```

```
## # A tibble: 105 x 3
## # Groups:   job_title [50]
##    job_title                          experience_level count
##    <chr>                              <ord>            <int>
##  1 3D Computer Vision Researcher      MI                   1
##  2 AI Scientist                       EN                   4
##  3 AI Scientist                       MI                   2
##  4 AI Scientist                       SE                   1
```

```
##   5 Analytics Engineer                 SE                   2
##   6 Analytics Engineer                 EX                   2
##   7 Applied Data Scientist             EN                   1
##   8 Applied Data Scientist             MI                   2
##   9 Applied Data Scientist             SE                   2
##  10 Applied Machine Learning Scientist EN                   1
## # i 95 more rows
```

```
job_title_by_experience <- filter(job_title_by_experience_level, count > 1)

ggplot(job_title_by_experience_level, aes(x = experience_level, y = count))+
  geom_col() +
  facet_wrap(~job_title, ncol = 8)
```




```
# employment type within each job title
job_title_by_employment_type <- salaries %>%
  group_by(job_title, employment_type) %>%
  summarize(count = n())
```

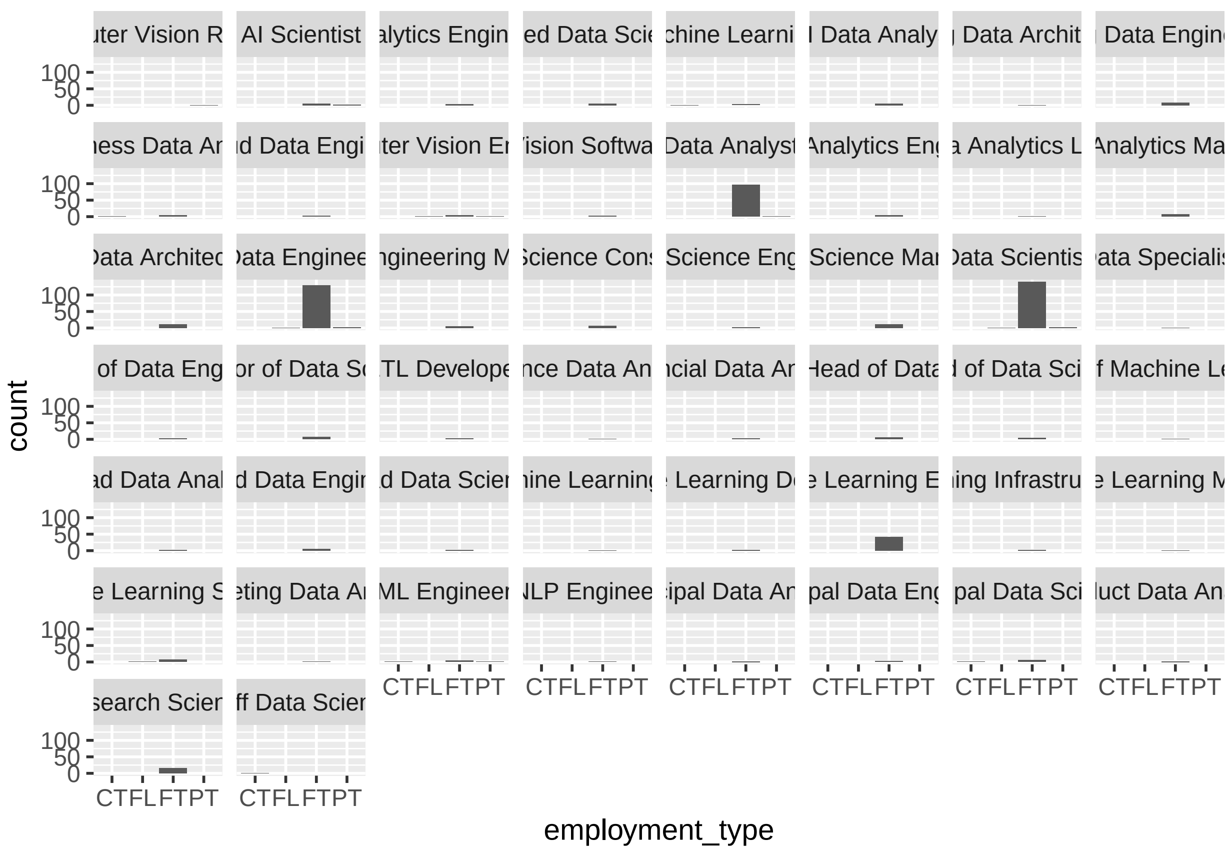
```
## `summarise()` has grouped output by 'job_title'. You can override using the
## `.groups` argument.
```

```
job_title_by_employment_type
```

```
## # A tibble: 64 x 3
```

```
## # Groups:   job_title [50]
##    job_title                          employment_type count
##    <chr>                              <fct>           <int>
##  1 3D Computer Vision Researcher      PT                  1
##  2 AI Scientist                       FT                  5
##  3 AI Scientist                       PT                  2
##  4 Analytics Engineer                 FT                  4
##  5 Applied Data Scientist             FT                  5
##  6 Applied Machine Learning Scientist CT                  1
##  7 Applied Machine Learning Scientist FT                  3
##  8 BI Data Analyst                    FT                  6
##  9 Big Data Architect                 FT                  1
## 10 Big Data Engineer                  FT                  8
## # i 54 more rows
```

```r
ggplot(job_title_by_employment_type, aes(x = employment_type, y = count))+
  geom_col() +
  facet_wrap(~job_title, ncol = 8)
```



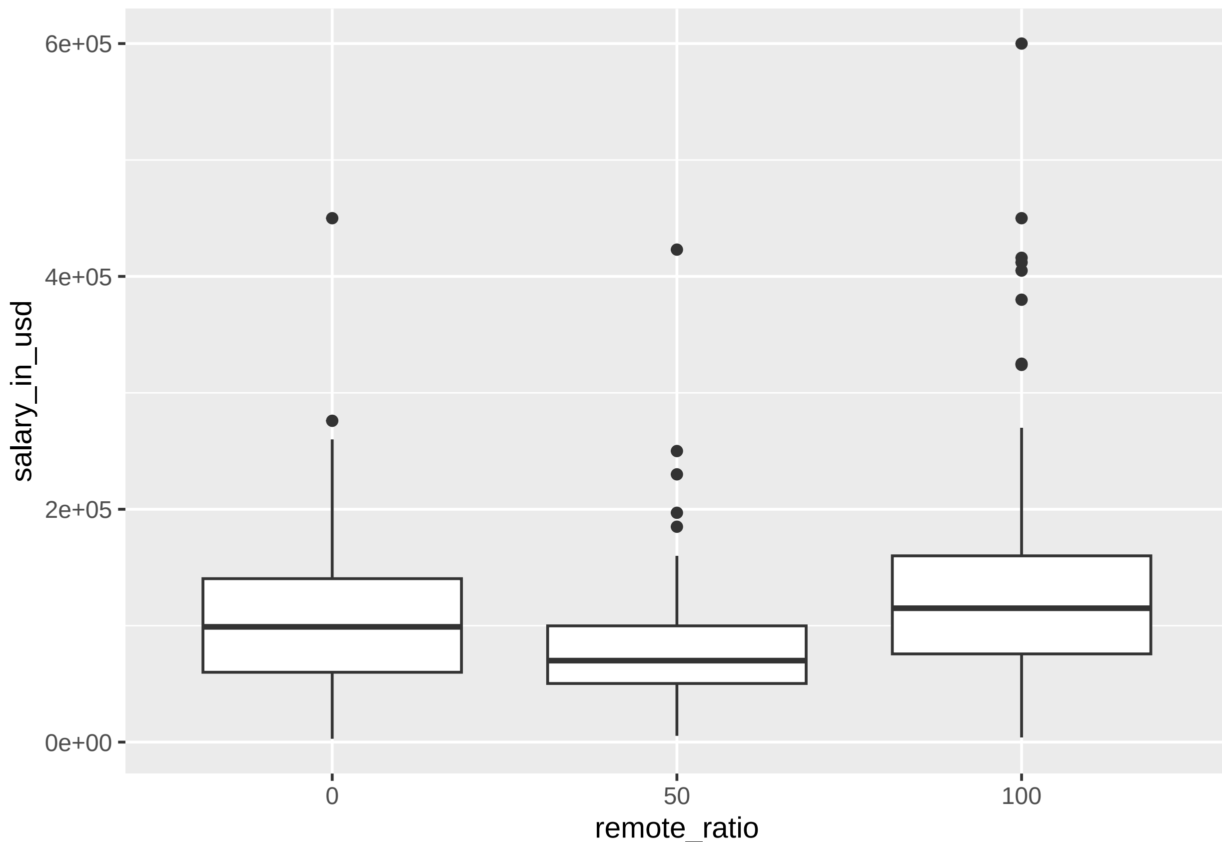

```r
# this plot is a mess and difficult to read
# I am interested in seeing the job titles that appear the most in this data set

job_title_by_employment_type <- arrange(job_title_by_employment_type, desc(count))
head(job_title_by_employment_type, 5)
```

```
## # A tibble: 5 x 3
```

```
## # Groups:   job_title [5]
##   job_title                employment_type count
##   <chr>                    <fct>           <int>
## 1 Data Scientist           FT                140
## 2 Data Engineer            FT                129
## 3 Data Analyst             FT                 96
## 4 Machine Learning Engineer FT                41
## 5 Research Scientist       FT                 16
```

The box plots below are created to look at the remote ratio versus salary.

```r
ggplot(salaries, aes(x = remote_ratio, y = salary_in_usd)) +
  geom_boxplot()
```



The shape of the three box plots was surprising to me. The 0% remote and 100% remote are almost identical, with 100% remote being slightly higher.

I was really excited to go through the article about maps in R. After experimenting with mapping in R, I thought it would be cool to make a color coded with map based on salary amounts in order to begin by comparing salaries in the United States to salaries elsewhere.

```r
# look at the unique values in company_location in alphabetical order
levels(unique(salaries$company_location))
```

```
##  [1] "AE" "AS" "AT" "AU" "BE" "BR" "CA" "CH" "CL" "CN" "CO" "CZ" "DE" "DK" "DZ"
## [16] "EE" "ES" "FR" "GB" "GR" "HN" "HR" "HU" "IE" "IL" "IN" "IQ" "IR" "IT" "JP"
```

```
## [31] "KE" "LU" "MD" "MT" "MX" "MY" "NG" "NL" "NZ" "PK" "PL" "PT" "RO" "RU" "SG"
## [46] "SI" "TR" "UA" "US" "VN"
```

```
# curious to see how many observations there are in each company_location
company_locations_count <- salaries %>%
  group_by(company_location) %>%
  summarize(count = n())
company_locations_count
```

```
## # A tibble: 50 x 2
##    company_location count
##    <fct>            <int>
##  1 AE                   3
##  2 AS                   1
##  3 AT                   4
##  4 AU                   3
##  5 BE                   2
##  6 BR                   3
##  7 CA                  30
##  8 CH                   2
##  9 CL                   1
## 10 CN                   2
## # i 40 more rows
```

```
arrange(company_locations_count, count)
```

```
## # A tibble: 50 x 2
##    company_location count
##    <fct>            <int>
##  1 AS                   1
##  2 CL                   1
##  3 CO                   1
##  4 DZ                   1
##  5 EE                   1
##  6 HN                   1
##  7 HR                   1
##  8 HU                   1
##  9 IE                   1
## 10 IL                   1
## # i 40 more rows
```

```
# the majority of data observations is from the United States (355); other countries with more than 5 o

# create a data frame with locations of each country that appear in "company_location"
# start by setting up a data frame with company locations as a column
location_coords <- data.frame(company_location = unique(salaries$company_location))

# import csv file with latitude and longitude values for each country
country_locations <- read.csv("country_locations.csv")

# left join location_coords with country_locations to include latitude and longitude for each of the co
```

```r
location_coords <- left_join(location_coords, country_locations,
                             join_by("company_location" == "country"))

head(location_coords)
```

```
##   company_location latitude  longitude           name
## 1               DE 51.16569  10.451526        Germany
## 2               JP 36.20482 138.252924          Japan
## 3               GB 55.37805  -3.435973 United Kingdom
## 4               HN 15.20000 -86.241905       Honduras
## 5               US 37.09024 -95.712891  United States
## 6               HU 47.16249  19.503304        Hungary
```

```r
# aggregate the salaries data by company_location and find the average salary_in_usd for each company_l
avg_salary_by_location <- salaries %>%
  group_by(company_location) %>%
  summarize(avg_salary_in_usd = mean(salary_in_usd))

# which countries have the highest average salaries?
head(avg_salary_by_location)
```

```
## # A tibble: 6 x 2
##   company_location avg_salary_in_usd
##   <fct>                        <dbl>
## 1 AE                          100000
## 2 AS                           18053
## 3 AT                           72921.
## 4 AU                          108043.
## 5 BE                           85699
## 6 BR                           18603.
```

```r
arrange(avg_salary_by_location, desc(avg_salary_in_usd))
```

```
## # A tibble: 50 x 2
##    company_location avg_salary_in_usd
##    <fct>                        <dbl>
##  1 RU                          157500
##  2 US                          144055.
##  3 NZ                          125000
##  4 IL                          119059
##  5 JP                          114127.
##  6 AU                          108043.
##  7 AE                          100000
##  8 DZ                          100000
##  9 IQ                          100000
## 10 CA                           99824.
## # i 40 more rows
```

```r
# merge avg_salary_by_location with location_coords
salaries_with_locations <- full_join(avg_salary_by_location, location_coords, join_by(company_location)

# check data frame to ensure it looks as expected and does not have any NA values
head(salaries_with_locations)
```

```
## # A tibble: 6 x 5
##   company_location avg_salary_in_usd latitude longitude name
##   <chr>                        <dbl>    <dbl>     <dbl> <chr>
## 1 AE                          100000     23.4     53.8  United Arab Emirates
## 2 AS                           18053    -14.3   -170.   American Samoa
## 3 AT                           72921.    47.5     14.6  Austria
## 4 AU                          108043.   -25.3    134.   Australia
## 5 BE                           85699     50.5      4.47 Belgium
## 6 BR                           18603.   -14.2    -51.9  Brazil
```

```r
is.na(salaries_with_locations)
```

```
##       company_location avg_salary_in_usd latitude longitude  name
##  [1,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [2,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [3,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [4,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [5,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [6,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [7,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [8,]            FALSE             FALSE    FALSE     FALSE FALSE
##  [9,]            FALSE             FALSE    FALSE     FALSE FALSE
## [10,]            FALSE             FALSE    FALSE     FALSE FALSE
## [11,]            FALSE             FALSE    FALSE     FALSE FALSE
## [12,]            FALSE             FALSE    FALSE     FALSE FALSE
## [13,]            FALSE             FALSE    FALSE     FALSE FALSE
## [14,]            FALSE             FALSE    FALSE     FALSE FALSE
## [15,]            FALSE             FALSE    FALSE     FALSE FALSE
## [16,]            FALSE             FALSE    FALSE     FALSE FALSE
## [17,]            FALSE             FALSE    FALSE     FALSE FALSE
## [18,]            FALSE             FALSE    FALSE     FALSE FALSE
## [19,]            FALSE             FALSE    FALSE     FALSE FALSE
## [20,]            FALSE             FALSE    FALSE     FALSE FALSE
## [21,]            FALSE             FALSE    FALSE     FALSE FALSE
## [22,]            FALSE             FALSE    FALSE     FALSE FALSE
## [23,]            FALSE             FALSE    FALSE     FALSE FALSE
## [24,]            FALSE             FALSE    FALSE     FALSE FALSE
## [25,]            FALSE             FALSE    FALSE     FALSE FALSE
## [26,]            FALSE             FALSE    FALSE     FALSE FALSE
## [27,]            FALSE             FALSE    FALSE     FALSE FALSE
## [28,]            FALSE             FALSE    FALSE     FALSE FALSE
## [29,]            FALSE             FALSE    FALSE     FALSE FALSE
## [30,]            FALSE             FALSE    FALSE     FALSE FALSE
## [31,]            FALSE             FALSE    FALSE     FALSE FALSE
## [32,]            FALSE             FALSE    FALSE     FALSE FALSE
## [33,]            FALSE             FALSE    FALSE     FALSE FALSE
## [34,]            FALSE             FALSE    FALSE     FALSE FALSE
## [35,]            FALSE             FALSE    FALSE     FALSE FALSE
## [36,]            FALSE             FALSE    FALSE     FALSE FALSE
## [37,]            FALSE             FALSE    FALSE     FALSE FALSE
## [38,]            FALSE             FALSE    FALSE     FALSE FALSE
## [39,]            FALSE             FALSE    FALSE     FALSE FALSE
## [40,]            FALSE             FALSE    FALSE     FALSE FALSE
## [41,]            FALSE             FALSE    FALSE     FALSE FALSE
```

```
## [42,]           FALSE            FALSE    FALSE     FALSE FALSE
## [43,]           FALSE            FALSE    FALSE     FALSE FALSE
## [44,]           FALSE            FALSE    FALSE     FALSE FALSE
## [45,]           FALSE            FALSE    FALSE     FALSE FALSE
## [46,]           FALSE            FALSE    FALSE     FALSE FALSE
## [47,]           FALSE            FALSE    FALSE     FALSE FALSE
## [48,]           FALSE            FALSE    FALSE     FALSE FALSE
## [49,]           FALSE            FALSE    FALSE     FALSE FALSE
## [50,]           FALSE            FALSE    FALSE     FALSE FALSE
```

```r
# change "name" values for Czech Republic and United States to ensure included when joining with "world

salaries_with_locations$name[salaries_with_locations$name == "United States"] <- "United States of Amer

salaries_with_locations$name[salaries_with_locations$name == "Czech Republic"] <- "Czechia"
```

```r
# load world map
world <- ne_countries(returnclass = "sf")


# merge world map data with salaries data
world_salaries <- full_join(salaries_with_locations, world, join_by("name" == "admin"))
head(world_salaries)
```

```
## # A tibble: 6 x 173
##   company_location avg_salary_in_usd latitude longitude name          featurecla
##   <chr>                        <dbl>    <dbl>     <dbl> <chr>         <chr>
## 1 AE                          100000     23.4      53.8 United Arab ~ Admin-0 c~
## 2 AS                           18053    -14.3    -170.  American Sam~ <NA>
## 3 AT                           72921.    47.5      14.6 Austria       Admin-0 c~
## 4 AU                          108043.   -25.3     134.  Australia     Admin-0 c~
## 5 BE                           85699     50.5       4.47 Belgium      Admin-0 c~
## 6 BR                           18603.   -14.2     -51.9 Brazil        Admin-0 c~
## # i 167 more variables: scalerank <int>, labelrank <int>, sovereignt <chr>,
## #   sov_a3 <chr>, adm0_dif <int>, level <int>, type <chr>, tlc <chr>,
## #   adm0_a3 <chr>, geou_dif <int>, geounit <chr>, gu_a3 <chr>, su_dif <int>,
## #   subunit <chr>, su_a3 <chr>, brk_diff <int>, name.y <chr>, name_long <chr>,
## #   brk_a3 <chr>, brk_name <chr>, brk_group <chr>, abbrev <chr>, postal <chr>,
## #   formal_en <chr>, formal_fr <chr>, name_ciawf <chr>, note_adm0 <chr>,
## #   note_brk <chr>, name_sort <chr>, name_alt <chr>, mapcolor7 <int>, ...
```

```r
# check the data frame for rows with empty geometries
filter(world_salaries, st_is_empty(geometry))
```

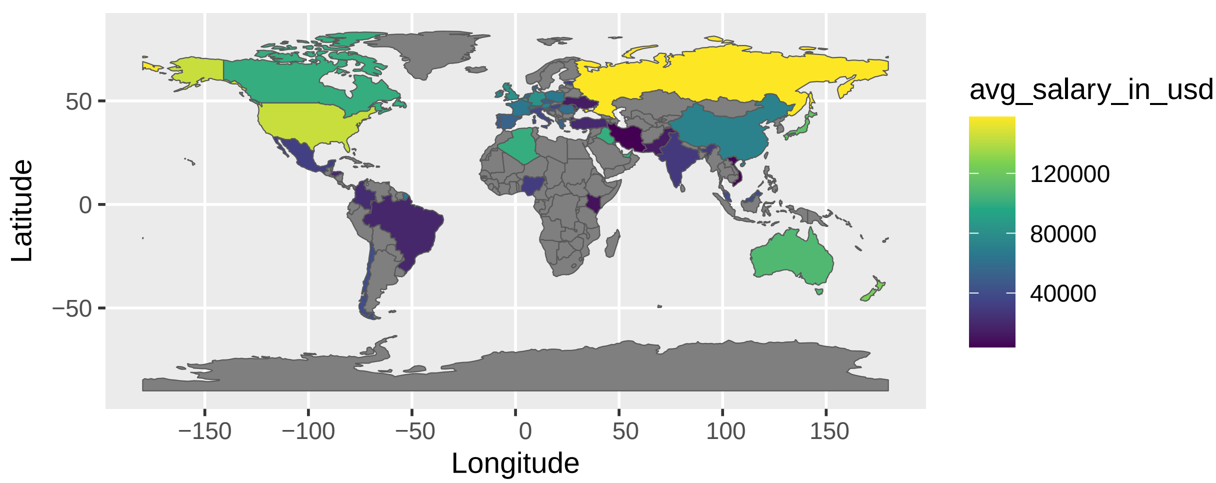
```
## # A tibble: 3 x 173
##   company_location avg_salary_in_usd latitude longitude name          featurecla
##   <chr>                        <dbl>    <dbl>     <dbl> <chr>         <chr>
## 1 AS                           18053   -14.3     -170.  American Sam~ <NA>
## 2 MT                           28369    35.9       14.4 Malta         <NA>
## 3 SG                           89294     1.35     104.  Singapore     <NA>
## # i 167 more variables: scalerank <int>, labelrank <int>, sovereignt <chr>,
## #   sov_a3 <chr>, adm0_dif <int>, level <int>, type <chr>, tlc <chr>,
```

```
## #   adm0_a3 <chr>, geou_dif <int>, geounit <chr>, gu_a3 <chr>, su_dif <int>,
## #   subunit <chr>, su_a3 <chr>, brk_diff <int>, name.y <chr>, name_long <chr>,
## #   brk_a3 <chr>, brk_name <chr>, brk_group <chr>, abbrev <chr>, postal <chr>,
## #   formal_en <chr>, formal_fr <chr>, name_ciawf <chr>, note_adm0 <chr>,
## #   note_brk <chr>, name_sort <chr>, name_alt <chr>, mapcolor7 <int>, ...
```

Missing data includes: * American Samoa - not listed in the “world” data frame, likely because it is a US territory * Czech Republic - appears in “world” data frame as Czechia; can go back and rename in the “salaries_with_locations” data frame * Malta -??? * Singapore - ??? * United States - appears in “world” data frame as United States of America; can go back and rename in “salaries_with_locations” data frame

After renaming the United States and Czech Republic, there are no more rows with empty geomtry…

```
# plot map
ggplot(data = world_salaries, aes(geometry = geometry)) +
  geom_sf(aes(fill = avg_salary_in_usd), position = "identity") +
  scale_fill_viridis_c(option = "viridis") +
  xlab("Longitude") + ylab("Latitude")
```



Finally, below I look at company_size compared to experience_level, employment_type, salary_in_usd, and remote_ratio using summary tables and appropriate plots.

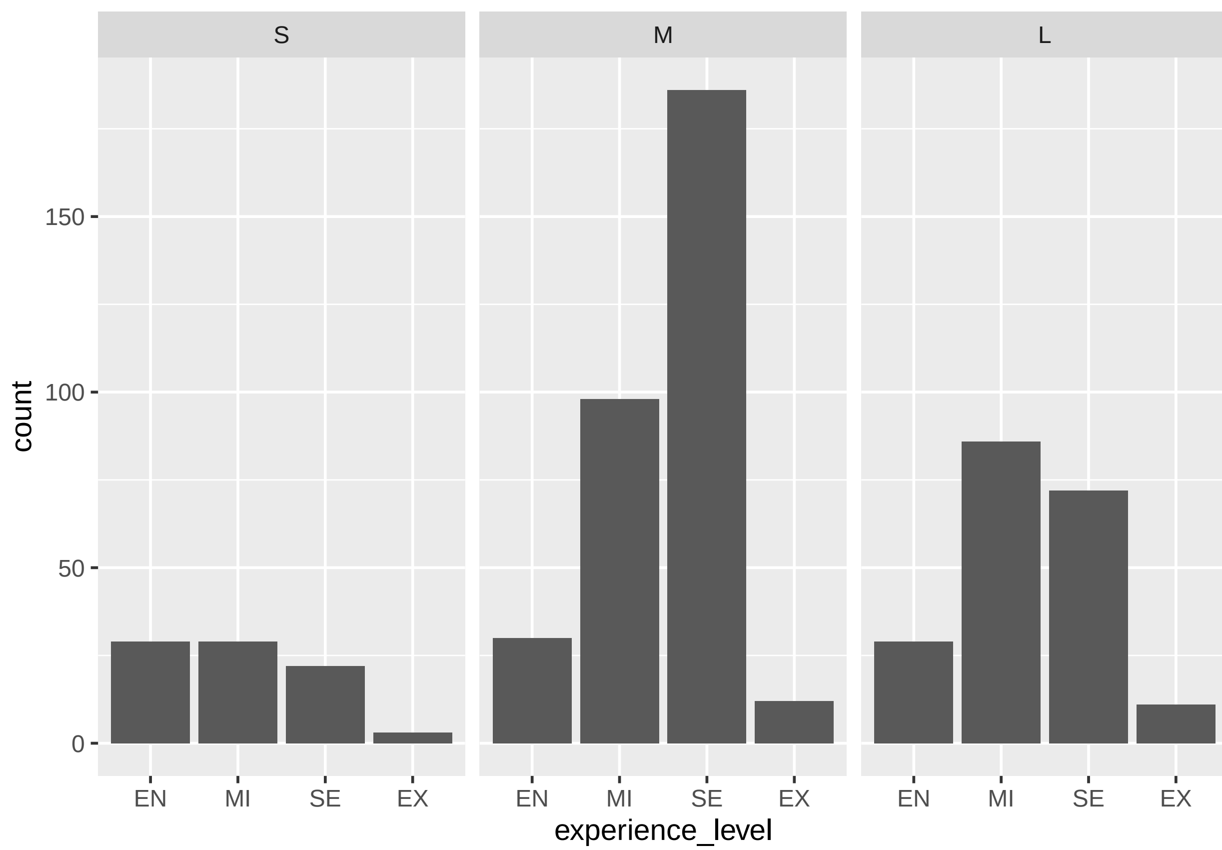
```
# create a summary table with the counts for each company size grouped by experience_level
company_size_by_experience <- salaries %>%
  group_by(company_size, experience_level) %>%
  summarize(count = n())
```

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

```r
company_size_by_experience
```

```
## # A tibble: 12 x 3
## # Groups:   company_size [3]
##    company_size experience_level count
##    <ord>        <ord>            <int>
##  1 S            EN                  29
##  2 S            MI                  29
##  3 S            SE                  22
##  4 S            EX                   3
##  5 M            EN                  30
##  6 M            MI                  98
##  7 M            SE                 186
##  8 M            EX                  12
##  9 L            EN                  29
## 10 L            MI                  86
## 11 L            SE                  72
## 12 L            EX                  11
```

```r
# create bar plot with counts faceted by company_size
ggplot(company_size_by_experience, aes(x = experience_level, y = count)) +
  geom_col() +
  facet_grid(~company_size)
```



```r
# create bar chart using percents calculated with in each year
ggplot(salaries_by_year,
       aes(x = bin, y = percent, fill = work_year))+
  geom_col(color = 'Dim Gray')+
  facet_wrap(~work_year, ncol = 1) +
  theme_bw() +
  scale_fill_manual(values = c("2020" = "Slate Blue",
                               "2021" = "Light Sea Green",
                               "2022" = "Lime Green")) +
  scale_x_discrete(labels = x_labels) +
  labs(title = 'Data Science Salaries Over Time',
       x = "Salary \n (in thousands of dollars)",
       y = "Percent of Employees \n (in given year)") +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
```

Based on this plot, there are clearly varying numbers of employees included in each company size. It would be more useful to look at the percent instead of the counts.

```r
salaries_by_company_size <- salaries %>%
  # group to find the counts per size
  group_by(company_size) %>%
  mutate(count_by_size = n()) %>%
  ungroup() %>%
  # group to find counts per experience level within each size
  group_by(company_size, experience_level, count_by_size) %>%
  summarize(count_employees = n()) %>%
```

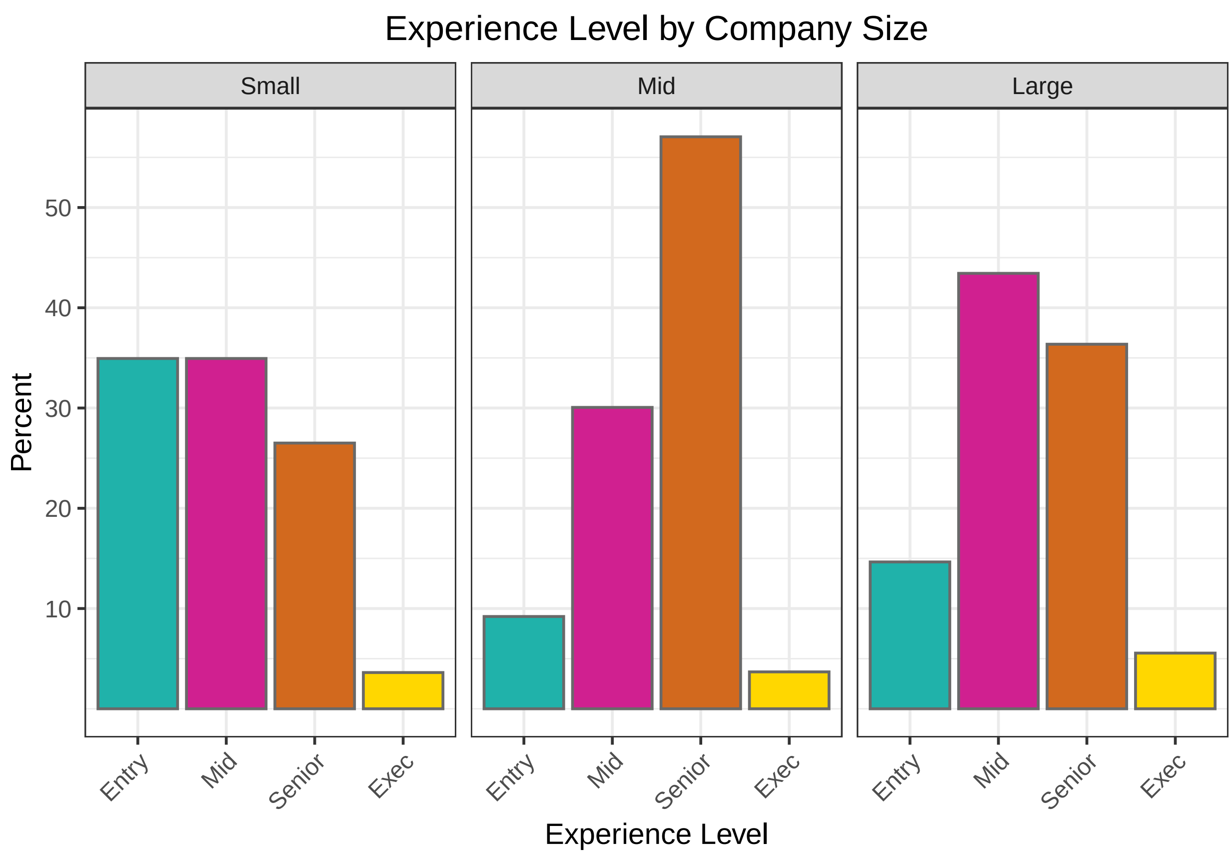
```r
  # add a column with percents
  mutate(percent = count_employees/count_by_size *100)
```

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

```r
salaries_by_company_size
```

```
## # A tibble: 12 x 5
## # Groups:   company_size, experience_level [12]
##    company_size experience_level count_by_size count_employees percent
##    <ord>        <ord>                    <int>           <int>   <dbl>
##  1 S            EN                          83              29   34.9
##  2 S            MI                          83              29   34.9
##  3 S            SE                          83              22   26.5
##  4 S            EX                          83               3    3.61
##  5 M            EN                         326              30    9.20
##  6 M            MI                         326              98   30.1
##  7 M            SE                         326             186   57.1
##  8 M            EX                         326              12    3.68
##  9 L            EN                         198              29   14.6
## 10 L            MI                         198              86   43.4
## 11 L            SE                         198              72   36.4
## 12 L            EX                         198              11    5.56
```

```r
# create bar plot with counts faceted by company_size with percents instead of counts
ggplot(salaries_by_company_size,
       aes(x = experience_level, y = percent, fill = experience_level)) +
  geom_col(color = 'Dim Gray') +
  facet_grid(~company_size,
             labeller = labeller(company_size = c(S = "Small", M = "Mid", L = "Large"))) +
  theme_bw() +
  scale_fill_manual(values = c("EN" = "Light Sea Green", "MI" = "Violet Red", "SE" = "Chocolate", "EX" =
  scale_x_discrete(labels = c("EN" = "Entry", "MI" = "Mid", "SE" = "Senior", "EX" = "Exec")) +
  scale_y_continuous(breaks = c(10, 20, 30, 40, 50)) +
  labs(title = "Experience Level by Company Size",
       x = "Experience Level",
       y = "Percent") +
  theme(legend.position = "none", plot.title = element_text(hjust = 0.5), axis.text.x = element_text(an
```



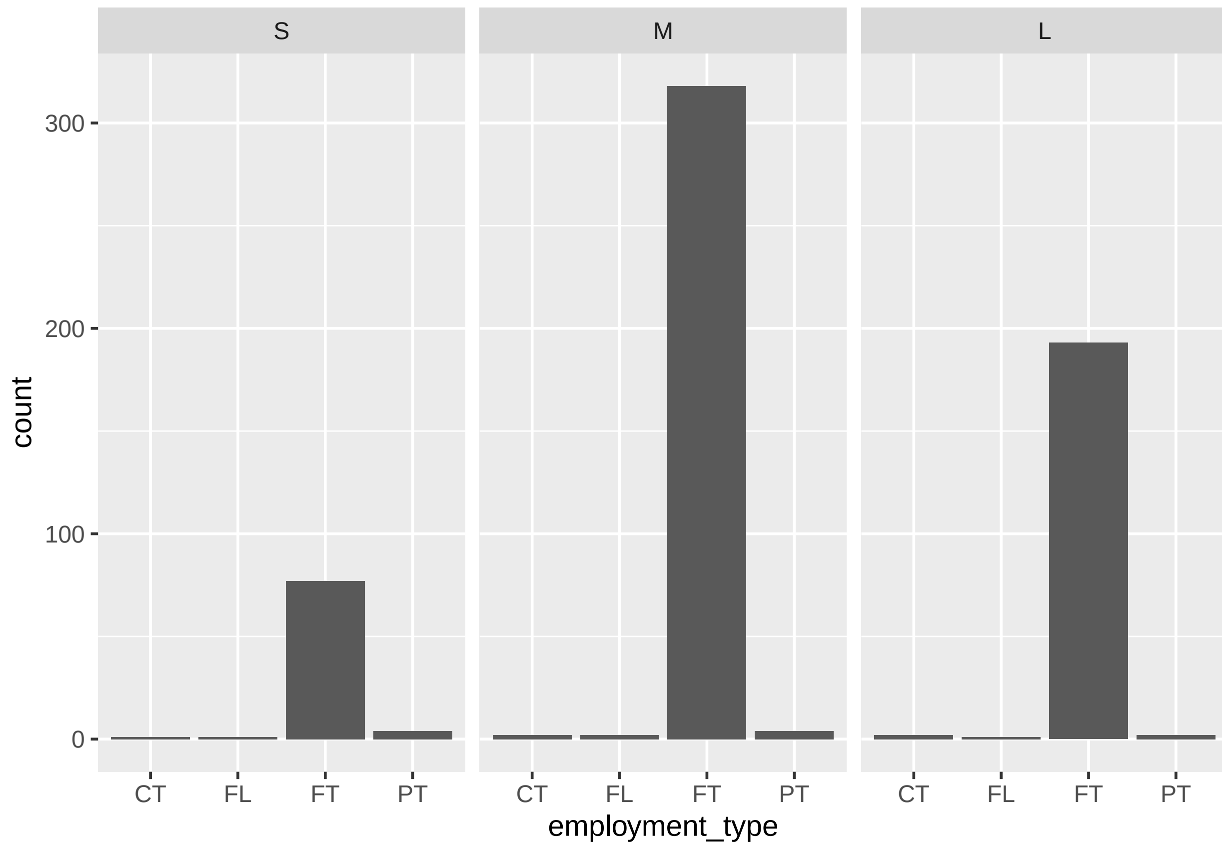

```r
# create a summary table with the counts for each company size grouped by employment type
company_size_by_employment_type <- salaries %>%
  group_by(company_size, employment_type) %>%
  summarize(count = n())
```

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

```r
company_size_by_employment_type
```

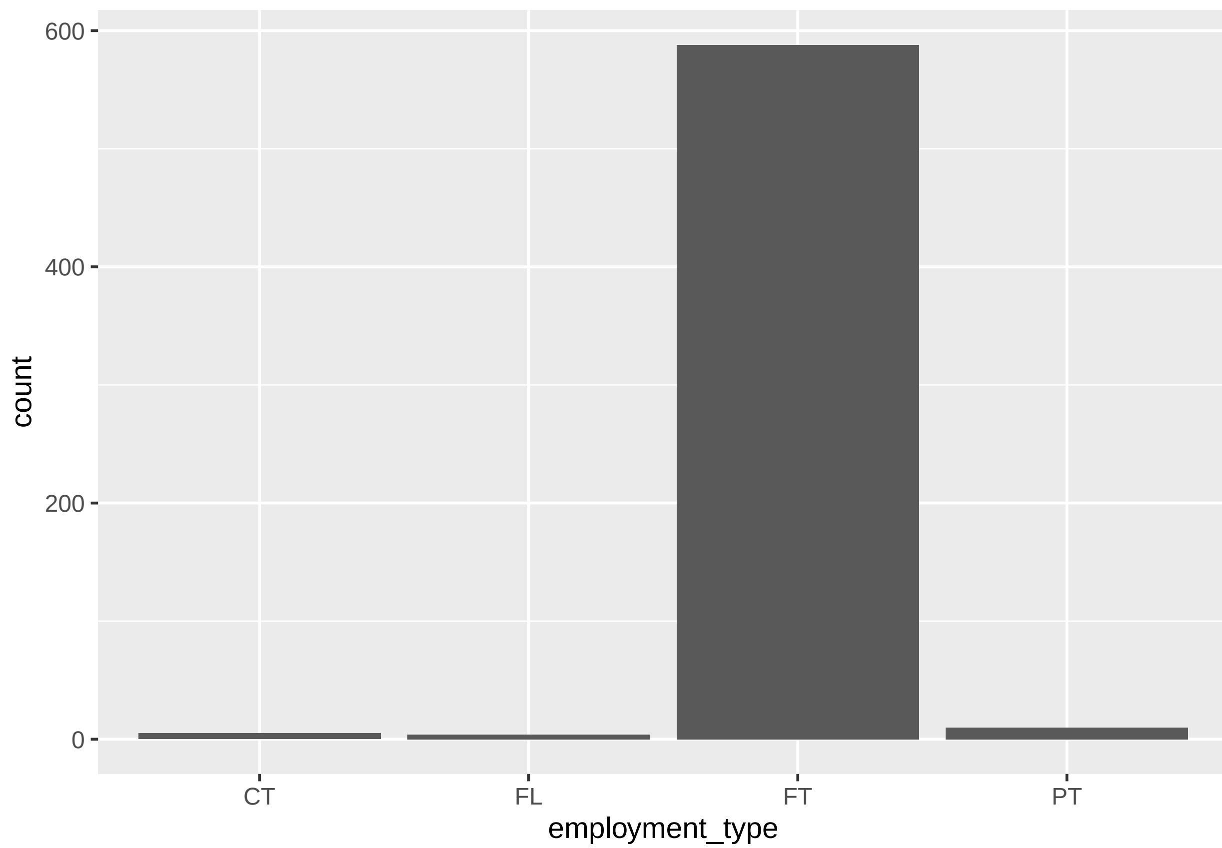
```
## # A tibble: 12 x 3
## # Groups:   company_size [3]
##    company_size employment_type count
##    <ord>        <fct>           <int>
##  1 S            CT                  1
##  2 S            FL                  1
##  3 S            FT                 77
##  4 S            PT                  4
##  5 M            CT                  2
##  6 M            FL                  2
##  7 M            FT                318
##  8 M            PT                  4
##  9 L            CT                  2
## 10 L            FL                  1
## 11 L            FT                193
## 12 L            PT                  2
```

```r
# create bar plot with counts faceted by company_size
ggplot(company_size_by_employment_type, aes(x = employment_type, y = count)) +
  geom_col() +
  facet_grid(~company_size)
```



While it may be helpful to look at the percents instead of the counts for employment_type, there is clearly mostly full time employees, regardless of the size of the company. I am interested in looking at the bar plot for employment_type overall…

```r
ggplot(salaries, aes(x=employment_type)) +
  geom_bar()
```



This looks comparable to each of the company size bar plots in shape. After looking at other variable comparisons, I may want to examine percents instead of counts.

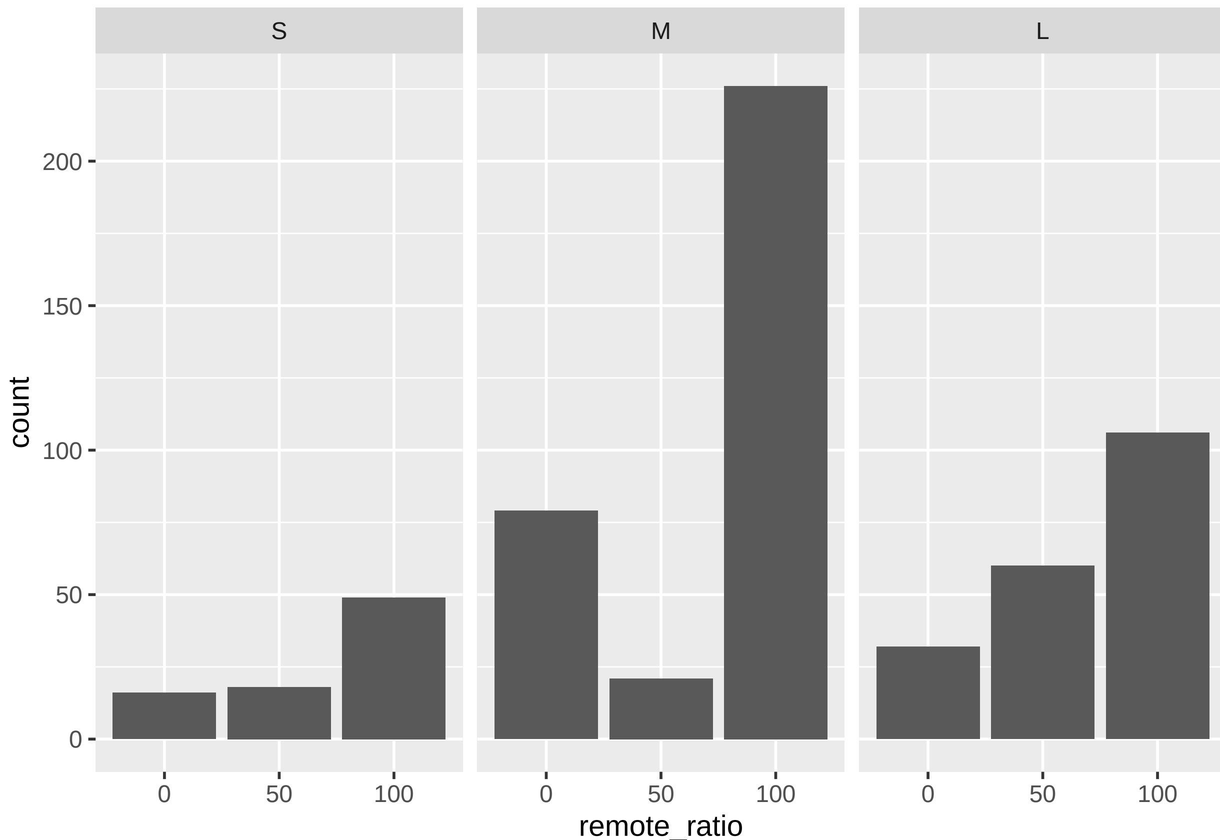
```
# create a summary table with the counts for each company size grouped by remote_ratio
company_size_by_remote_ratio <- salaries %>%
  group_by(company_size, remote_ratio) %>%
  summarize(count = n())
```

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

```
company_size_by_remote_ratio
```

```
## # A tibble: 9 x 3
## # Groups:   company_size [3]
##   company_size remote_ratio count
##   <ord>        <fct>        <int>
## 1 S            0               16
## 2 S            50              18
## 3 S            100             49
## 4 M            0               79
## 5 M            50              21
## 6 M            100            226
## 7 L            0               32
## 8 L            50              60
## 9 L            100            106
```

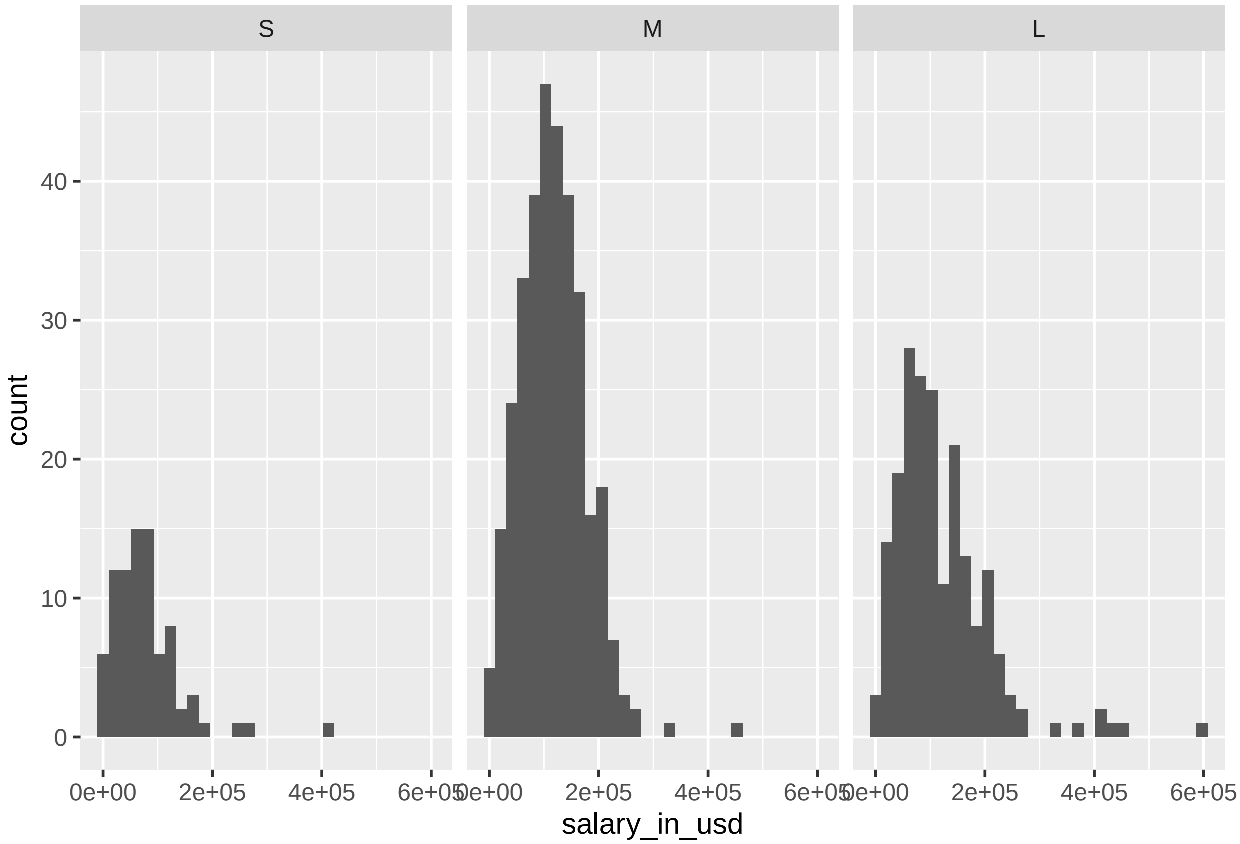
```r
# create bar plot with counts faceted by company_size
ggplot(company_size_by_remote_ratio, aes(x = remote_ratio, y = count)) +
  geom_col() +
  facet_grid(~company_size)
```



The distribution of 0%, 50%, and 100% remote workers varies based on company size. Again, looking at percents may be more useful given the varying total for each company size.

```r
ggplot(salaries, aes(x = salary_in_usd)) +
  geom_histogram() +
  facet_grid(~company_size)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



All three of the histograms are right skewed with the mode around $100,000. Large companies appear to have more high outliers.

MY DATA STORY My presentation takes the viewer from a broader view to a more detailed view. My first slide looks at the increase of data science salaries in recent years. Where are data scientists being paid these amounts we see in recent years? That is answered in the second slide, which compares data salaries based on location, highlighting the countries where data science jobs pay the most, including the US. Now that we have seen what the "typical" salary may be in the US in 2024, we need to consider what our company is willing to pay based on our priorities. If we aspire to grow from a small company to a mid-size company (and eventually a large company), how much will we have to pay and who will we need to hire? This is addressed by looking at a breakdown of company sizes and employee experience.