Mibella Villafana
PST107_22021
21/10/22
Data Science Salaries Analysis

Purpose/Intro:

We are interested in finding out what the mean salary in data science careers is. We intend to educate future Data Analysts and Scientists whether this career will provide them with a large enough salary to provide an excellent standard of living within the geographic location they are based. These findings will guide the hopeful scientists either towards or away from the career based on their personal motivations.

Dataset Information:

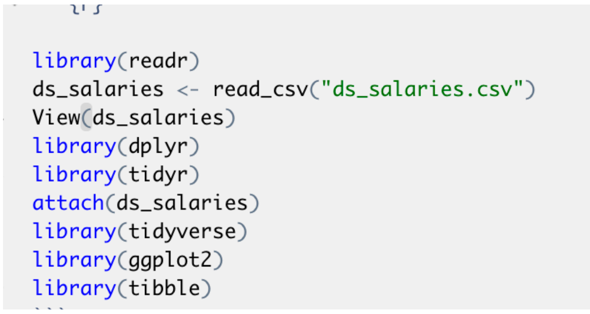
We sourced a csv file from Kaggle (kaggle.com) called “Data Science Job Salaries.” This data has been posted on Kaggle by“ai-jobs.net" which provides job openings and AI related insights. The dataset’s relevant data includes work year paid between 2020-2022, experience level with the values “EN” for entry, “MI” for mid level, “SE” for senior, and “EX” for executive level. It includes employment type whether they are full time (“FT”), part time (“PT”), contractors (“CT”) or freelance (“FL”). We also look at salary in USD, company location, remote ratio (0- less than 20% remote work, 50- partially remote and 100- more than 80% remote), and company size. The data also includes salary (in country’s currency), country’s salary currency, job title (including data scientist, data analyst, data engineer, machine learning engineer, etc.

Tools and libraries

We uploaded the data into R studio via the import function and call the following libraries: “readr,” “dplyr,” “tidyr,” “tidyverse,” “ggplot2,” “tibble.” Dplyr and tidyr allow us to clean and begin processing the data.

Pipeline/Process/Algorithms:

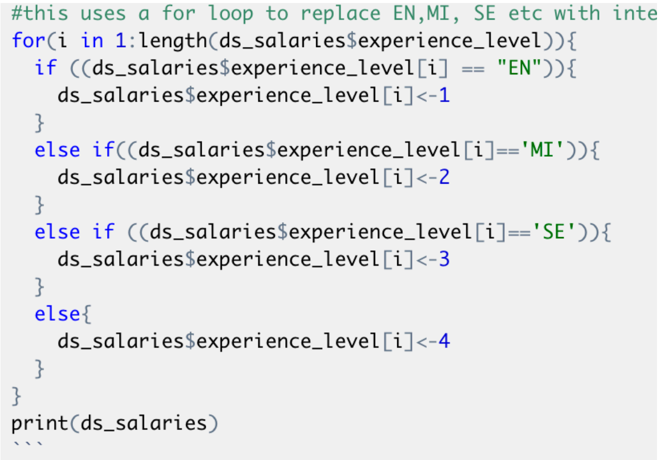
Our most relevant data in this set are “salaries in USD” and “experience level.” We take a few measures to clean and process before visualising the data and returning conclusions. First we



```
library(readr)
ds_salaries <- read_csv("ds_salaries.csv")
View(ds_salaries)
library(dplyr)
library(tidyr)
attach(ds_salaries)
library(tidyverse)
library(ggplot2)
library(tibble)
```

Caption

import the data set and run the libraries listed above. Then we ran a summary(ds_salaries) to identify mean and medians of all the double type values. Then we use the mutate function to modify the variables.
We implemented the “for loop” to switch the experience level character values from “EN,” etc., to “1,2,3,4” so that visualisation would be made easer to see.

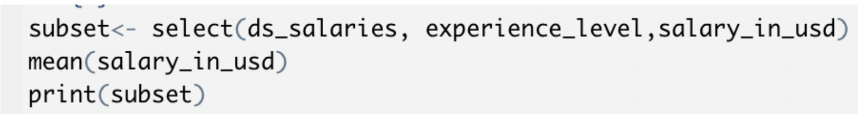


~~~
#this uses a for loop to replace EN,MI, SE etc with inte
for(i in 1:length(ds_salaries$experience_level)){
  if ((ds_salaries$experience_level[i] == "EN")){
    ds_salaries$experience_level[i]<-1
  }
  else if((ds_salaries$experience_level[i]=='MI')){
    ds_salaries$experience_level[i]<-2
  }
  else if ((ds_salaries$experience_level[i]=='SE')){
    ds_salaries$experience_level[i]<-3
  }
  else{
    ds_salaries$experience_level[i]<-4
  }
}
print(ds_salaries)
```
~~~

Caption

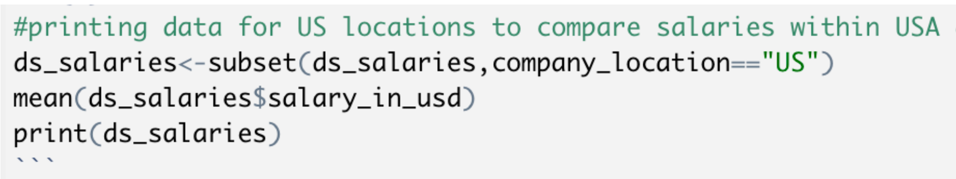
Then we changed the class type to numeric with this function : ds_salaries$experience_level<-as.numeric(ds_salaries$experience_level)
print(ds_salaries)
This organised the experience levels as a double.

We want to isolate the full time workers from the rest so we dropped all others using the subset function and produced a table with only “FT” in the “experience_level” column.
Function used: ds_salaries<-subset(ds_salaries,employment_type==“FT”)
Also, to better visualise this process we arranged the experience level from top to bottom with this function:
 ds_salaries %>% arrange(desc(experience_level)).
This simply organises the data in descending order from experience level 4-1.
We then calculated the mean global salaries in USD totalling to $112297.9.



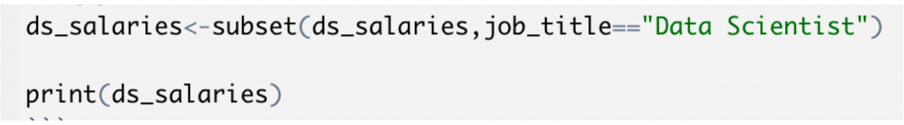
~~~
subset<- select(ds_salaries, experience_level,salary_in_usd)
mean(salary_in_usd)
print(subset)
~~~

We ran another subset to find out average salaries in USD for companies set in US only using the subset below. This printed the salary within US companies to $144637.6



~~~
#printing data for US locations to compare salaries within USA
ds_salaries<-subset(ds_salaries,company_location=="US")
mean(ds_salaries$salary_in_usd)
print(ds_salaries)
```
~~~

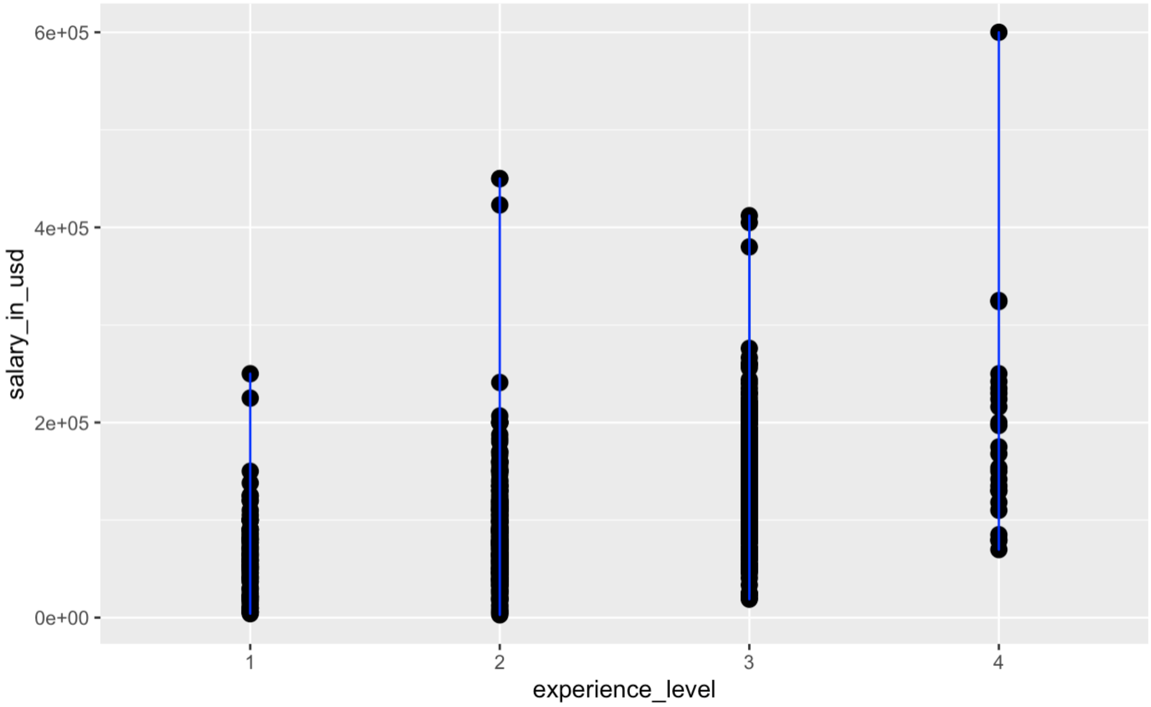
We are curious about Data Scientist jobs isolated from the other variations listed in this table to we ran the whole program from the beginning again without isolating US salaries, and instead isolating “Data Science” jobs using another subset function giving us the mean of $108922.8



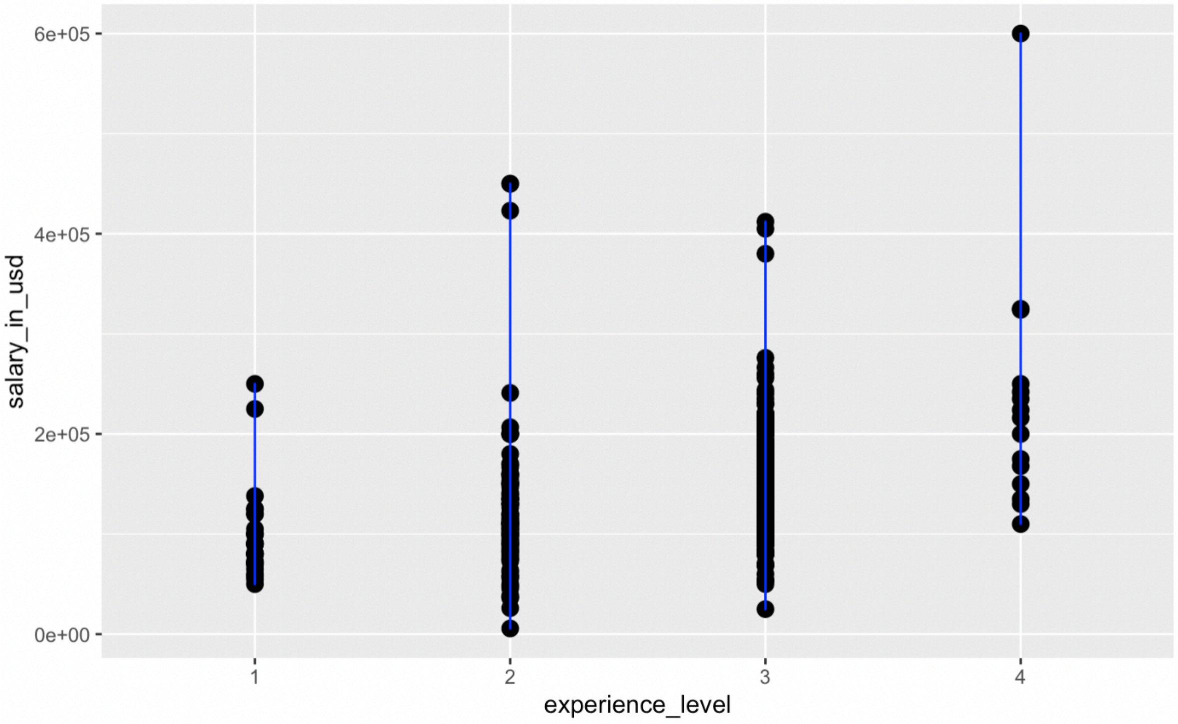
~~~
ds_salaries<-subset(ds_salaries,job_title=="Data Scientist")

print(ds_salaries)
~~~

Then we ran the same program including the isolation of "US" company location and "Data Science" jobs and returned a mean salary value of $143635.1.

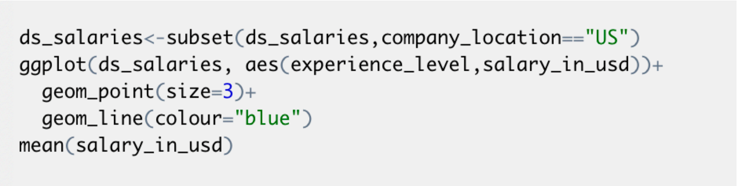
We visualised the data by going back and running the program to first show us the average salaries in USD for companies in the US only for all data science related jobs and got this graph.



Then we found the same graph but including all countries globally and returned this graph.

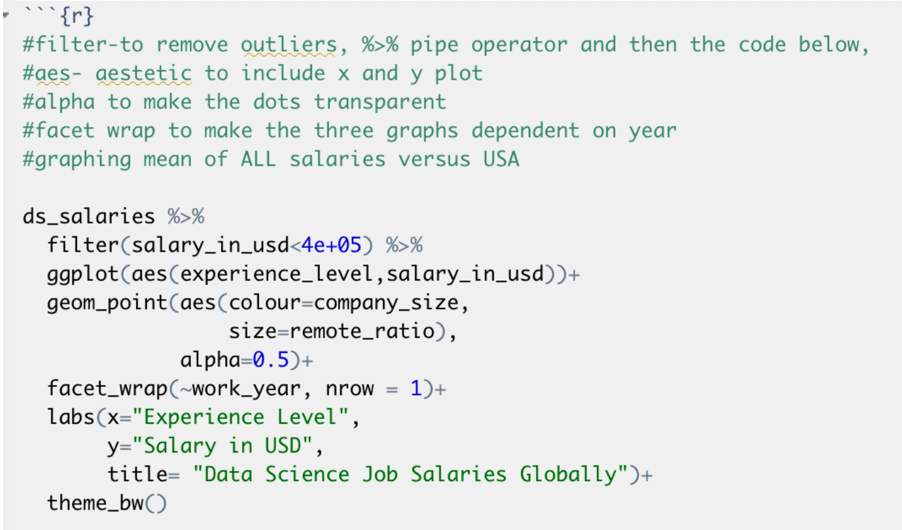


We used this algorithm to plot



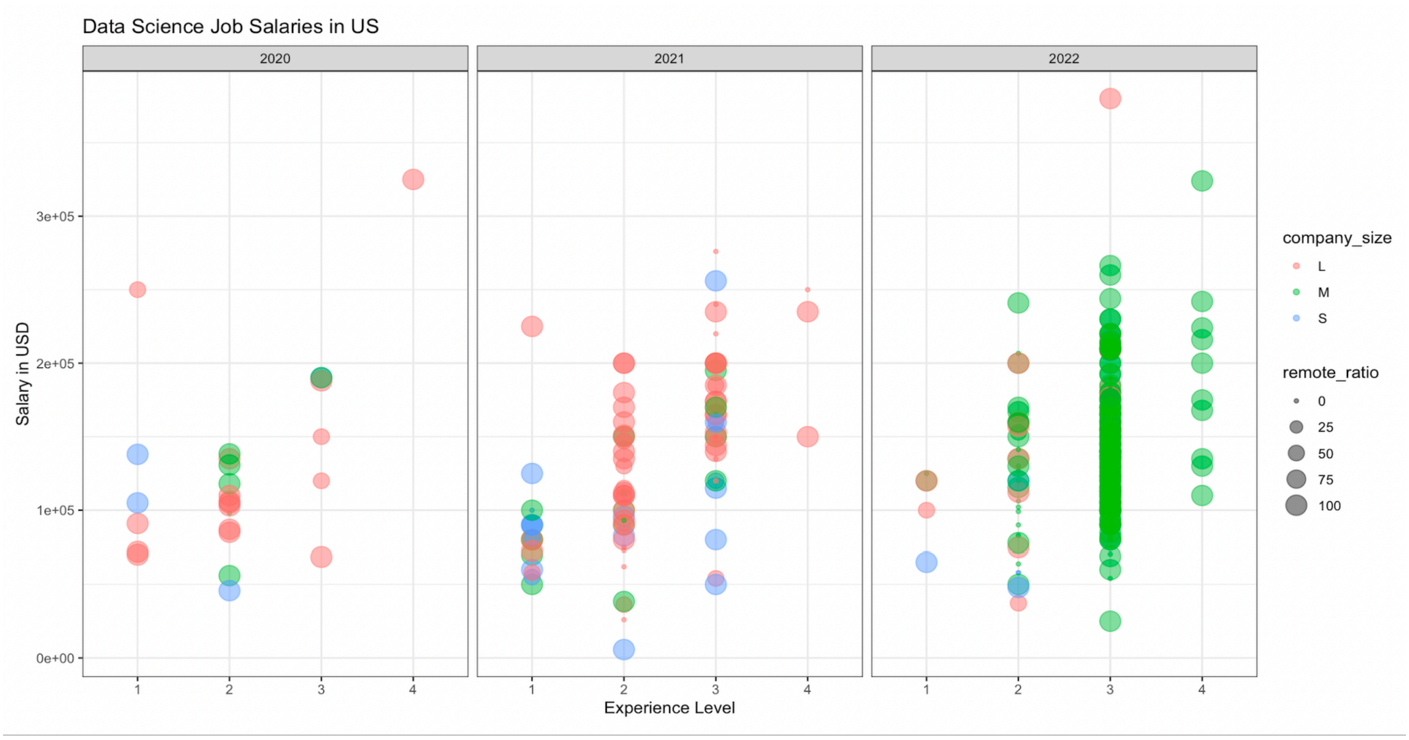
```
ds_salaries<-subset(ds_salaries,company_location=="US")
ggplot(ds_salaries, aes(experience_level,salary_in_usd))+
  geom_point(size=3)+
  geom_line(colour="blue")
mean(salary_in_usd)
```

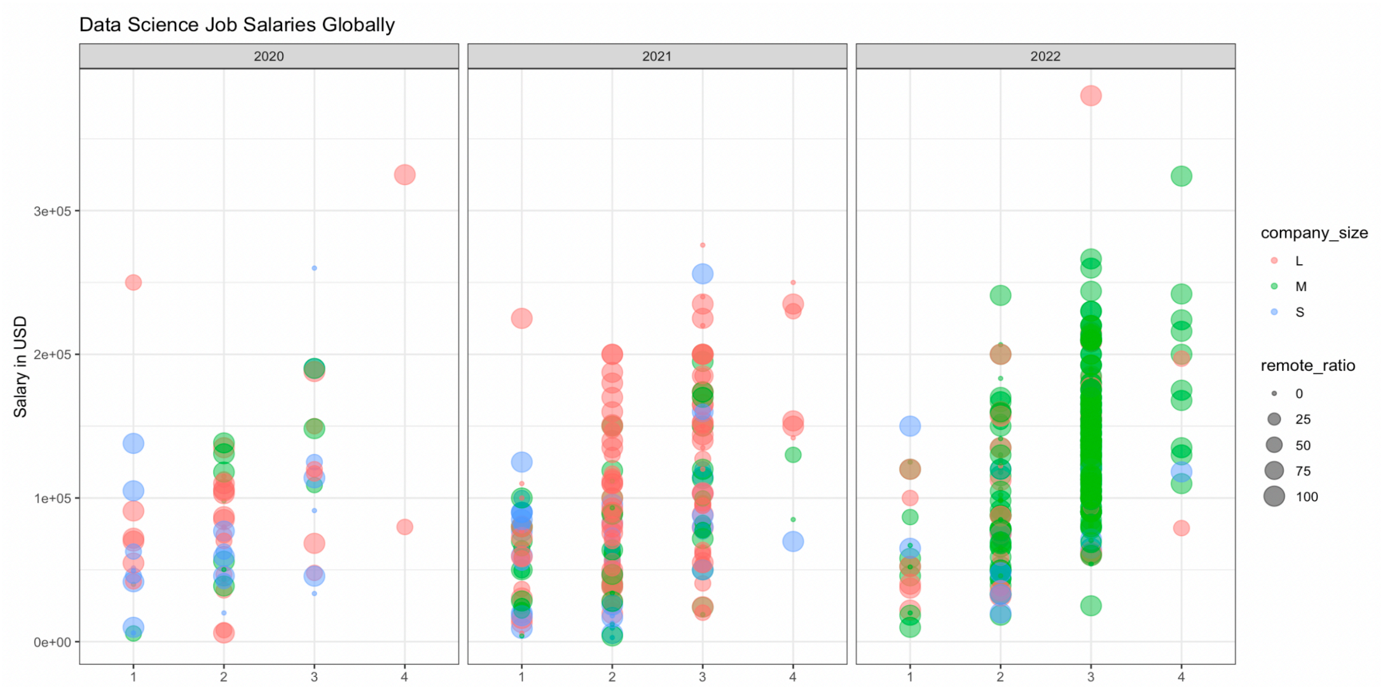
We tried a few different graphs as this one did not show a big difference visually so we used a different algorithm to plot the same values and added company size and remote ratio within the colour and size of the dots.



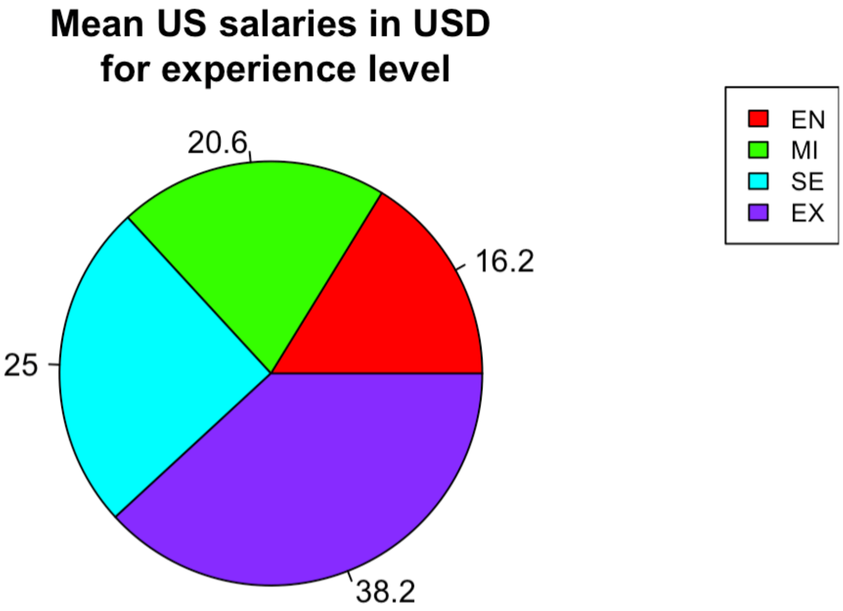
~~~
```{r}
#filter-to remove outliers, %>% pipe operator and then the code below,
#aes- aestetic to include x and y plot
#alpha to make the dots transparent
#facet wrap to make the three graphs dependent on year
#graphing mean of ALL salaries versus USA

ds_salaries %>%
  filter(salary_in_usd<4e+05) %>%
  ggplot(aes(experience_level,salary_in_usd))+
  geom_point(aes(colour=company_size,
                 size=remote_ratio),
             alpha=0.5)+
  facet_wrap(~work_year, nrow = 1)+
  labs(x="Experience Level",
       y="Salary in USD",
       title= "Data Science Job Salaries Globally")+
  theme_bw()
~~~

This produced these two graphs:

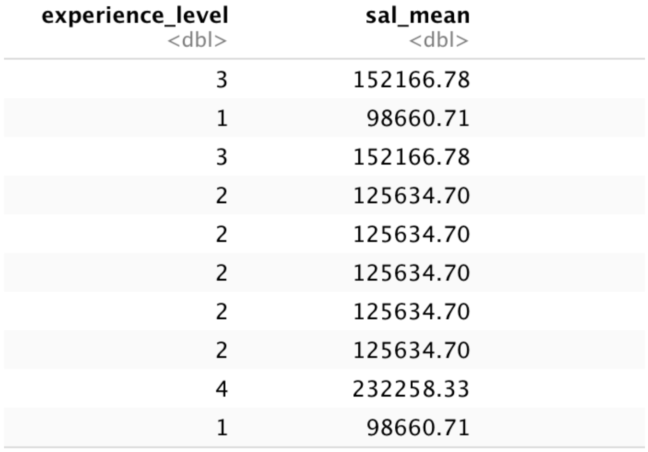




There are only subtle differences in these graphs visually as well, so we made a pie chart. As you can see the largest (executive level) group in the chart earns 38% of the salary amount, followed by senior level with 25%, followed by mid level career with 20.6%, and leaving entry level earners with the lowest at 16.2%. They also have too many different variables that we could leave out and visualise in different graphs to make things more clear.

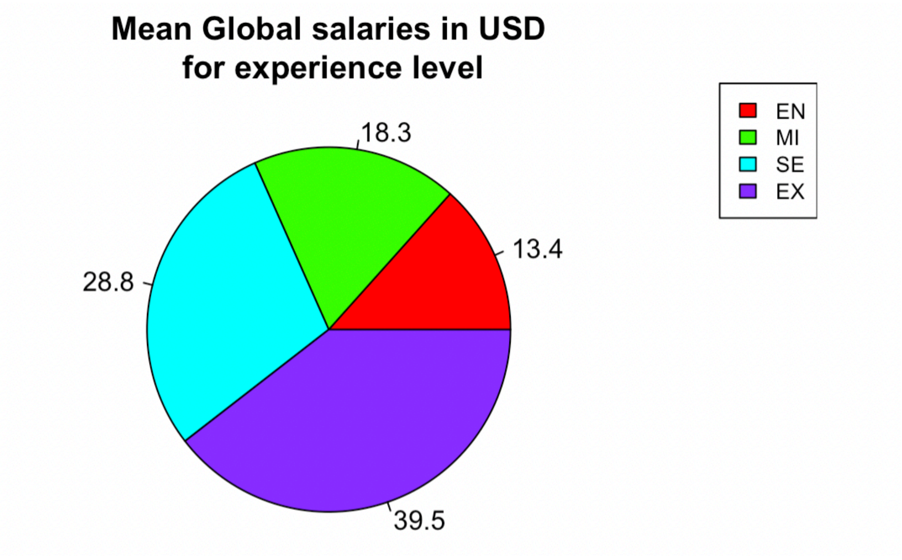


These are reflected by the values from this table which takes the percentages from the highest salary being $232258.33 and lowest being $98660.71.

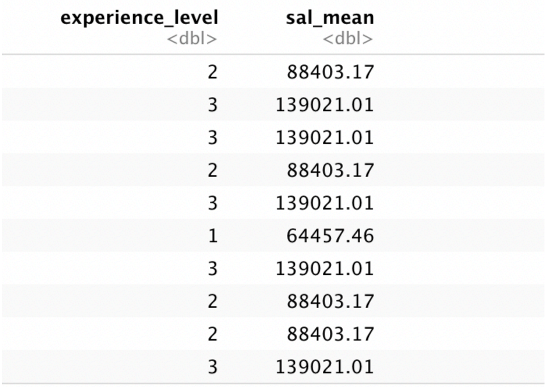


experience_level <dbl>	sal_mean <dbl>
3	152166.78
1	98660.71
3	152166.78
2	125634.70
2	125634.70
2	125634.70
2	125634.70
2	125634.70
4	232258.33
1	98660.71

We have produced another pie chart that shows the global salaries in USD based divided by experience level.



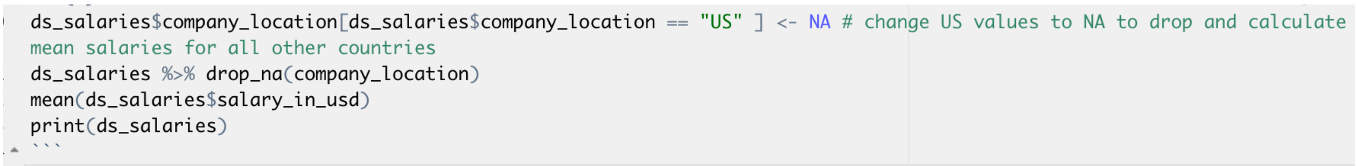
Compared to the US values, the Global percentage values do not vary significantly because they are in relation to each other. However, the actual values do vary for example the highest salary average which is in executive level is $190727.72 and lowest being in entry level is $64457.46. So the global salary proportions are similar, but values are not.



experience_level <dbl>	sal_mean <dbl>
2	88403.17
3	139021.01
3	139021.01
2	88403.17
3	139021.01
1	64457.46
3	139021.01
2	88403.17
2	88403.17
3	139021.01

This can lead us to some interesting questions regarding salaries that reflect standard of living within certain geographical locations and what is essential for living a good life.

To further our curiosity, we relabelled the “US” values in the company location argument to “NA” so that we could drop their values and keep all the other countries’. This returned a mean salary in USD for all other countries without the US included.



```
ds_salaries$company_location[ds_salaries$company_location == "US" ] <- NA # change US values to NA to drop and calculate
mean salaries for all other countries
ds_salaries %>% drop_na(company_location)
mean(ds_salaries$salary_in_usd)
print(ds_salaries)
```
```

This is new data the we could graph in the future which would give a more accurate visualisation of the difference between US salaries and that of all other countries. In any case we did return the mean value of $113468.1 which is not very different from the global average including the US, $112297.9.

Hypothesis

We predict in our hypothesis that there will be a significant difference in salaries within the US and globally.
H0- There is no difference in salaries.
H1- There is a difference in salaries.

We use a t-test to provide a significance values. Though there is difficulty in providing a two sided T- test that used the values of the mean of US companies and the global mean. The reason is because in order to find the mean of US values and global values we would have to make another table within the dataset separating the two different values. This would result in a different sample size for each set that we are comparing. This may be something we cold do in the future once more skills are obtained to do so.

Findings/conclusions

It appears that there is a difference in salaries based on location. Compared to the mean salary from the Bureau of Labor Statistics which states that, “the median wage for workers in the United States in the second quarter of 2022 was about $1,041 per week or **$54,132 per year** (assuming 52 weeks of work per year).” This shows that altogether the findings we have analysed all prove a much higher salary than the national average and can further hypothesise that a data scientist related salary will prove to be sufficient to create a high standard of living. There are definitely future comparisons that can be made to determine relative earning salary to cost of living in various geographic locations.
In addition, the two sided t- test can certainly be improved to be functional. It will be an interesting test to effectively implement as it will provide information on whether these comparisons in mean salaries that we have access to are statistically significant.
So far we only found with one sample t- test that we have 95% confidence that the mean US salary in USD is between $137488 and $151787.3.

Personal reflection

In my opinion, this data set provided information to us that is very hopeful in terms of salary amounts around the world including the US. I believe it may be one category to consider while entering the field, however because the salary box is sufficiently ticked, it will now depend on other aspects of the career that would make it attractive or not.
```

Overall, the project was a huge learning curve and simply cleaning the data and producing visualisations were a fun challenge to work through.

**Future**

As mentioned previously, in the future we could compare different costs of living values with the salaries within this data set. We would produce graphs to reflect that. We would do more research as to whether there is a true difference within the way the different variations of “data scientist” careers are labelled and whether it is important to divide them up into different groups given that they are different. We can spend more time finessing the graph values and trying out bar graphs. I hope this project gives clients insight as to whether this is a great career to pursue and if it’s based on salary, then it sure seems a safe direction to go.

In addition, improvements in using R Studio can be made as running the code to plot the proper values in this project needs to be done in order, and when you want to run different values for example including only “US salaries,” you must re run the code from the top to only include the desired values. This may be confusing for the client to run on their own, so an error that can be fixed in the future.

## References

Bhalla, D. (n.d.). *R : Keep / drop columns from data frame*. ListenData. Retrieved October 21, 2022, from https://www.listendata.com/2015/06/r-keep-drop-columns-from-data-frame.html

Bhatia, R. (2022, June 15). *Data Science Job Salaries*. Kaggle. Retrieved October 21, 2022, from https://www.kaggle.com/datasets/ruchi798/data-science-job-salaries

*Calculate Group Mean & add as new column to data frame (R example)*. Statistics Globe. (2022, May 9). Retrieved October 21, 2022, from https://statisticsglobe.com/calculate-group-mean-add-as-new-column-data-frame-r

CN, P. (2022, August 3). *The sub() and GSUB() function in R*. DigitalOcean. Retrieved October 21, 2022, from https://www.digitalocean.com/community/tutorials/sub-and-gsub-function-r

*Convert data frame column to numeric in R (2 example codes)*. Statistics Globe. (2022, March 16). Retrieved October 21, 2022, from https://statisticsglobe.com/convert-data-frame-column-to-numeric-in-r

*How to to replace values in a DataFrame in R*. Data to Fish. (2022, March 12). Retrieved October 21, 2022, from https://datatofish.com/replace-values-dataframe-r/#:~:text=Here%20is%20the%20syntax%20to%20replace%20values%20in,Replace%20a%20value%20under%20a%20single%20DataFrame%20column%3A

Kumar, G. S. (2022, July 20). *How to delete rows in R? explained with examples*. Spark by {Examples}. Retrieved October 21, 2022, from https://sparkbyexamples.com/r-programming/drop-dataframe-rows-in-r/#:~:text=R%20provides%20a%20subset(),notation%20%5B%5D%20and%20%2Dc().

Mike DewarMike Dewar 10.8k1414 gold badges4848 silver badges6464 bronze badges. (1957, August 1). *Convert data.frame columns from factors to characters*. Stack Overflow. Retrieved October 21, 2022, from https://stackoverflow.com/questions/2851015/convert-data-frame-columns-from-factors-to-characters

R - pie charts. (n.d.). Retrieved October 21, 2022, from https://sceweb.sce.uhcl.edu/helm/WEBPAGE-R/my_files/ChartsGraphs/Module-1/r__pie_charts.html

*Reorder data frame rows in R*. Datanovia. (2018, October 19). Retrieved October 21, 2022, from https://www.datanovia.com/en/lessons/reorder-data-frame-rows-in-r/

YouTube. (2021, February 2). *Ggplot for plots and graphs. an introduction to data visualization using R programming*. YouTube. Retrieved October 21, 2022, from https://www.youtube.com/watch?v=HPJn1CMvtmI&t=1s

YouTube. (2022, March 30). *T-test and interpreting P values using R programming*. YouTube. Retrieved October 21, 2022, from https://www.youtube.com/watch?v=fO2X-8FXY6k