Project 1

PJ Grant

2024-12-01

```
#Import Libraries
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

```
library(readxl)
library(writexl)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
#File path for excel outputs to create charts in PPT
file_path = '~/MM/DSE5002/Week_5/Project 1/proj_1_summary_stats.xlsx'
```

Load and prepare data for use

```
#Load Data
raw_data = read.csv('~/MM/DSE5002/Week_5/Project 1/r project data.csv')

#Update name and begin data cleaning
data = raw_data
```

```r
#dropping unnecessary columns
drop_cols = c('X', 'salary', 'salary_currency')

data = data |>
  select(-all_of(drop_cols))

#Rename salary column
data = data |>
  rename(
    salary = salary_in_usd
  )

#Select only FT roles
data = data |>
  filter(
    employment_type == "FT"
  )
```

#EDA

```r
#Determine how many submissions we have for each role
summarystats = data |>
  group_by(job_title) |>
  summarise(
    count = n(),
    .groups = 'drop'
  )
```

We will analyze the following roles: Data Scientist, Data Engineer, Data Analyst, and Machine Learning Engineer. These account for 70% of full-time submissions, with the rest being variations of these roles.

```r
#Isolating the 4 roles above for the rest of the analysis
kept_roles = c('Data Scientist', 'Data Engineer', 'Data Analyst', 'Machine
Learning Engineer')
ds_data = data |>
  filter(
    job_title %in% kept_roles
  )

#Distinguish which submissions are US based vs Offshore
ds_data = ds_data |>
   mutate(
    off_onshore = ifelse(employee_residence == 'US','On','Off')
  )
```

Aggregations for Slide 1

```
# Determine the average onshore vs offshore salary for each role in 2020,
2022 and the growth rate
avg_salary = ds_data |>
  filter(
    work_year != '2021'
  ) |>
  group_by(job_title, off_onshore, work_year) |>
  summarize(
   avg_salary = round(mean(salary),0),
   .groups = 'drop'
  ) |>
  pivot_wider(
    names_from = work_year,
    values_from = avg_salary,
    names_prefix = 'year_'
  ) |>
  mutate(
    cagr = round(((year_2022 / year_2020)^(1/2)) - 1,2)
  )

#write data to excel to create chart in ppt
write_xlsx(avg_salary, path = '~/MM/DSE5002/Week_5/Project 1/slide1.xlsx')
```

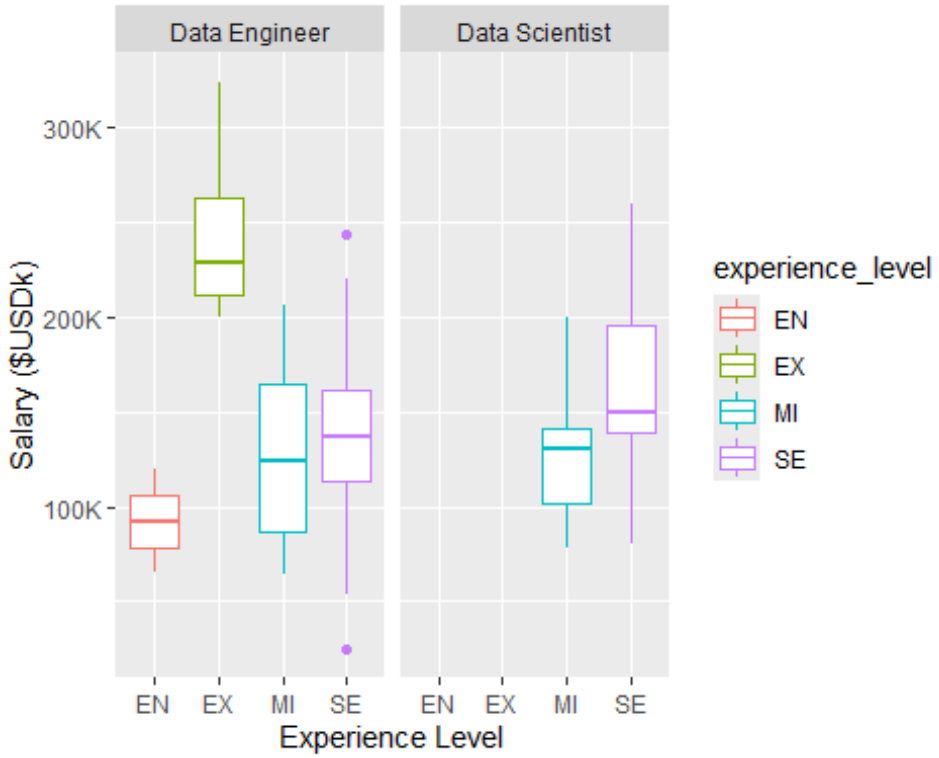
Aggregations for Slide 2

```
#Filter for only 2022 Data for DS and DE roles in the US for small and medium
sized companies
slide2 = ds_data |>
  filter(
    work_year == '2022',
    job_title %in% c('Data Scientist', 'Data Engineer'),
    employee_residence == 'US',
    company_size %in% c('S', 'M')
  )

# Calculate the mean and median salaries for each of these roles, these will
serve as the salary ranges we expect to pay these two roles + 20% for
overhead. The recommendation on the page will be to hire one SE Data
scientist and a MI Data Engineer
salary_ranges = slide2 |>
  group_by(job_title, experience_level) |>
  summarize(
    median_salary = median(salary),
    avg_salary = mean(salary),
    .groups = 'drop'
  )

#Create a histogram, faceted by job type showing the salaries for each of
```

```r
these roles
ggplot(slide2, aes(x = experience_level, y = salary, colour =
experience_level)) +
  geom_boxplot() +
  scale_y_continuous(labels = label_number(scale = 1e-3, suffix = "K")) +
  labs(y = "Salary ($USDk)", x = 'Experience Level') +
  facet_wrap(~job_title)
```




```r
#Work from home % by job title and experience level
wfh = slide2 |>
  group_by(job_title, experience_level, remote_ratio) |>
  summarize(
    count = n(),
    .groups = 'drop'
  )

#write data to excel to create chart in ppt
write_xlsx(salary_ranges, path = '~/MM/DSE5002/Week_5/Project 1/slide2.xlsx')
```

Aggregations for Slide 3

```r
#Looking to fill out the rest of the team with a mid level analyst and data
engineer who lives offshore
slide3 = ds_data |>
  filter(
    job_title %in% c('Data Analyst', 'Data Engineer', 'Machine Learning
```

```
Engineer'),
    company_size %in% c('S', 'M'),
    experience_level == "MI",
    work_year != '2020',
    employee_residence != 'US'
  ) |>
  group_by(job_title, employee_residence) |>
  summarise(
    avg_salary = mean(salary),
    median_salary = median(salary),
    .groups = 'drop'
  )

#write data to excel to create chart in ppt
write_xlsx(slide3, path = '~/MM/DSE5002/Week_5/Project 1/slide3.xlsx')
```