Joel Riondato Final Project MXB262

Joel_Riondato_N11380748

2024-04-23

Executive Sumary

This R-Markdown acts as a guide for individuals within or prospecting for a career in the data field. The report will include present and past positions within the data space & analyse critical factors that will aid to determine the optimal position, the markdown can also be used as a resource to compare data positions.

The criteria to evaluation the position type in this report include the following:
- Distribution of yearly salary by data position. - Proportion of work style relative to location by data position. - Proportion of experience level positions by data position. - Salary by data position internationally.

The perspective of this report is to provide the tools and resources to indicate an optimal data position for an individual, like myself which will enter the data field in the coming years.

```r
#calling the packages that will be used within this markdown.
library(ggplot2)
library(magrittr)
library(ggthemes)
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:magrittr':
##
##     extract
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2

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

```r
library(dplyr)
library(stringr)
```

Data Sources:

The first data used was retrieved from Kaggle, the data set includes international data positions collected via survey from years 2020 to 2024, with month not specified. Although accessed through Kaggle, it originates from a survey website, 'ai-jobs.net' with its respective link to the data source here https://ai-jobs.net/salaries/download/ . It is not clear if the same person has completed the survey multiple times or every survey entry is a new person.

Kaggle Data Link: https://www.kaggle.com/datasets/murilozangari/jobs-and-salaries-in-data-field-2024

The second data set was collected from stackoverflow & its 2023 developer survey. The survey although public, heavily contained entries via the platform's users. As this is a yearly survey, its implied that everyone completed the survey once, however having multiple accounts of a user is a possibility.

Stack Overflow link: https://cdn.stackoverflow.co/files/jo7n4k8s/production/49915bfd46d0902c3564fd9a06b509d08a20488c.zip/stack-overflow-developer-survey-2023.zip

Data Wrangling Section: KaggleDataSet

```r
#reading in the first data set from Kaggle
#ds <- read.csv('/Users/joel/Desktop/Uni/MXB262/Final Report/Data/DataScienceJobDetails2024.csv')
ds <- read.csv('Data/DataScienceJobDetails2024.csv')

#adding a job_type column that is catagorised by the job_category in the data set. More than once respo
KaggleDataSet <- ds %>%
    mutate(job_type = case_when( #using a case method for each job_category
        job_category %in% c("Data Analysis", "BI and Visualization") ~ "Analyst",
        job_category %in% c("Data Engineering", "Cloud and Database","Data Quality and Operations") ~ "
        job_category %in% c("Data Science and Research", "Machine Learning and AI", "Data Architecture a
        job_category %in% c("Leadership and Management", "Data Management and Strategy") ~ "Manager",
        TRUE ~ "Other"  #returns other as default response if no matches occur.
    ))

#changing the work_setting values in the data set so its equal to other data set in future (In-person -
KaggleDataSet <- KaggleDataSet %>%
  mutate(work_setting = case_when( #same structure as method used before in job_category
    str_detect(work_setting, "Hybrid") ~ "Hybrid",
    str_detect(work_setting, "In-person") ~ "In-Person",
    str_detect(work_setting, "Remote") ~ "Remote",
    TRUE ~ "Other" #default case when no matches occur
))

#converting usd to aud (1.55 @ 23-04-2024) within the salary field.
KaggleDataSet$salary_in_aud <- KaggleDataSet$salary_in_usd * 1.55


#creating another version of Kaggle data set which excludes data where the person doesnt live in the sa
KaggleDataSetLivesAndWorksSameCountry <- KaggleDataSet %>%
  mutate(LivesAndWorksCountrySame = case_when(
    employee_residence == company_location ~ "Yes",
    TRUE ~ "No"
  )) %>% filter(!LivesAndWorksCountrySame == "No")



#removing the columns not needed in the data set so its easier to work with.
KaggleDataSet <- KaggleDataSet %>% select(-work_year, -job_title, -employment_type, -company_location, 
#changing employee_residence column to Country
names(KaggleDataSet)[names(KaggleDataSet) == 'employee_residence'] <- 'Country'
```

Data Wrangling Section: Stack Overflow 2023 Survey Data

```
#calling libraries to be used
library(tidyr)
library(tidyverse)
library(dplyr)
library(stringr)

#reading the data set in
#stackoverflowdata2023 <- read.csv("/Users/joel/Desktop/Uni/MXB262/Final Report/Data/stackoverflow2023/
stackoverflowdata2023 <- read.csv("Data/stackoverflow2023/survey_results_public.csv")


#adding a job_type column that is catagorised by the DevType column in the dataset. More than once resp
stackoverflowdata2023V2 <- stackoverflowdata2023 %>%
  mutate(job_type = case_when(
    DevType %in%  "Engineer, data" ~ "Engineer",
    DevType %in% c("Data scientist or machine learning specialist","Research & Development role")  ~ "S
    DevType %in% "Data or business analyst" ~ "Analyst",
    DevType %in% c("Product manager","Project manager") ~ "Manager",
    TRUE ~ "Other" #default case.
  ))

#neglecting rows in data that aren't relevant (NA, 0 , Other)
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% filter(!job_type == "Other")
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% filter(!Country == "NA")
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% filter(!Currency == "NA")
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% filter(!CompTotal == "NA")
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% filter(!CompTotal == 0)


#I've created this exchange rates matrix for global countries to AUD, rates are from: 5/5/24
exchange_rates <- c(
"Germany" = 1.6270574,
"Canada" = 1.1045459,
"India" = 0.018114717,
"Finland" = 1.6270046,
"Romania" = 0.32713762,
"Australia" = 1,
"Greece" = 1.6269712,
"Cyprus" = 1.6269712,
"Lithuania" = 1.6269712,
"Turkey" = 0.046811468,
"Netherlands" = 1.6269826,
"Italy" = 1.6269826,
"Serbia" = 0.013893213,
"Morocco" = 0.15016055,
"Egypt" = 0.031530894,
"Switzerland" = 1.6683233,
"Spain" = 1.6268748,
"Brazil" = 0.29692736,
"Luxembourg" = 1.6270183,
"Belgium" = 1.6270183,
"United Kingdom of Great Britain and Northern Ireland"= 1.8960272,
"Austria" = 1.6267218,
"China" = 0.20943957,
"Iran,Islamic Republic of..." = 0.000035894582,
"Russian Federation" = 0.016601841,
"France" = 1.6266683,
"Portugal" = 1.6266683,
"Oman" = 3.9258034,
"Poland" = 0.37605658,
"Sweden" = 0.13964465,
"Nomadic" = 0,#null
"Israel" = 0.40661643,
"Kenya" = 0.011193026,
"Singapore" = 1.1179524,
"Pakistan" = 0.0054257294,
"Congo,Republic of the..." = 0.00054272168,
"Dominican Republic" = 0.02606343,
"Malaysia" = 0.31877023,
"Hungary" = 0.0041761514,
"Tunisia" = 0.4834971,
"Argentina" = 0.0017237657,
"Denmark" = 0.2181094,
"Armenia" = 0.0039163154,
"ElSalvador" = 0.1727092, # in usd
"VietNam" = 0.000059892818,
"Colombia" = 0.00038718059,
"Slovakia" = 1.6272561,
"Guatemala" = 0.19466628,
"Ireland" = 1.6272561,
"Mexico" = 0.089088401,
"Bolivia" = 0.21873045,
"Slovenia" = 1.6273862,
"Estonia" = 1.6273862,
"Ukraine" = 0.038324275,
"Thailand" = 0.041125401,
"Iraq" = 0.0011550657,
"Norway"= 0.13914947,
"NewZealand" = 0.90744843,
"Guinea" = 0.00017622576,
"Japan"= 0.0098318899,
"Chile" = 0.001604157,
"Philippines" = 0.026411535,
"HongKong(S.A.R.)" = 0.19346604,
"Republic of Korea" = 0.0016798802,
"Taiwan" = 0.04673362,
"Mongolia" = 0.00044509306,
"Nepal" = 0.011319328,
"Indonesia" = 0.000094318918,
"Malta" = 1.6272907,
"Uruguay" = 0.039530965,
"Bangladesh" = 0.01380191,
"Nigeria" = 0.0010955954,
"SouthKorea" = 0.0011122162,
"SriLanka" = 0.0050977806,
"Bulgaria" = 0.83203241,
"SouthAfrica" = 0.081551592,
"United Arab Emirates" = 0.41168501,
"Czech Republic" = 0.064904686,
"Georgia" = 0.56526821,
"Belarus" = 1.5117735, #USD
"Latvia" = 1.6271336,
"The former Yugoslav Republic of Macedonia" = 0.026443535, #euro
"Croatia" = 1.6270901,  #euro
"Madagascar" = 0.00034022676,
"Zambia" = 0.000056164361, #null
"Albania" = 0.016182416,
"Fiji" = 0.67269559,
"Mozambique" = 0.023715146,
"Jordan" = 2.1323567,
"Côted'Ivoire" = 1, #null,
"Iceland" = 0.010827246,
"Lebanon" = 0.000016537032,
"Angola" = 0.0017935863,
"Algeria" = 0.011247144,
"CostaRica" = 0.0029785091,
"Venezuela, Bolivarian Republic of..." = 0.041427671,
"Paraguay" = 0.0002018787,
"Uzbekistan" = 0.00011916097,
"Kazakhstan" = 0.0034069026,
"Afghanistan" = 0.020911868,
"Honduras" = 0.061271725,
"Jamaica" = 0.0096732824,
"Ghana" = 0.1103904,
"Ecuador" = 1.5121785, #usd
"Barbados" = 0.75608926,
"Bosnia and Herzegovina" = 0.83218568,
"Uganda" = 0.00040032793,
"Maldives" = 0.098134316,
"Qatar" = 0.41546019,
"Republic of Moldova"= 0.085555418,
"Montenegro" = 1.6275645,
"United Republic of Tanzania" = 0.00058339329,
"Azerbaijan" = 0.88895605,
"Zimbabwe" = 0.11184303,
"SaudiArabia" = 0.40323268,
"Namibia" = 0.08158869,
"IsleofMan" = 1.8970879,
"Malawi" = 0.00086889451,
"Togo" = 0.002481253,
"Peru" = 0.40599358,
"Syrian Arab Republic" = 0.00011630497,
"Mauritius" = 0.032684923,
"Ethiopia" = 0.026385283,
"Panama" = 1.5120457,
"Kyrgyzstan" = 0.017050059,
"Suriname" = 0.045545647,
"Somalia" = 0.0026553499,
"Kosovo" = 1, #null
"Nicaragua" = 0.041162645,
"SierraLeone" = 0.065343406,
"Rwanda" = 0.001173303,
"Cuba" = 0.06305892,
"Kuwait" = 4.9183541,
"Swaziland" = 0.081575163,
"Cambodia" = 0.00037209166,
"Yemen" = 0.0060394599,
"United States of America" = 1.5115134
)

#adding the exchange_rate column in the dataset, the case checks to see if the value in the 'Currency'
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>%
    mutate(
        exchange_rate = case_when(
            Currency == "EUR European Euro" ~ 1.6273697,
            Currency == "USD\tUnited States dollar" ~ 1.5115134,
            Country %in% names(exchange_rates) ~ exchange_rates[Country],
            TRUE ~ 0
        ))
#removing data that has echange_rate as 0, therefore indicating that is doesnt match any exchange rates
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% filter(!exchange_rate == 0)

#adding column 'salary_in_aud' in data that uses CompTotal * exchange_rate, this finds the annual AUD t
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>%
    mutate(
        salary_in_aud = CompTotal * exchange_rate
    )

#adding the work_setting column to this data set which categorises the RemoteWork column into work sett
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>%
  mutate(work_setting = case_when(
    str_detect(RemoteWork, "Hybrid (some remote, some in-person)") ~ "Hybrid",
    str_detect(RemoteWork, "In-person") ~ "In-Person",
    str_detect(RemoteWork, "Remote") ~ "Remote",
    TRUE ~ "Other" #default value if no matches occur.
  ))
```

```r
#removing the data where the work_setting is "Other"
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% filter(!work_setting == "Other")

#selecting only relevant columns and making data easier to work with & so it will merge with other data
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>% select(Country, job_type, salary_in_aud, work_se

#adding the experience_level column to the dataset and setting as NA, lets data merge with first datase
stackoverflowdata2023V2 <- stackoverflowdata2023V2 %>%
  mutate(experience_level = NA)
```

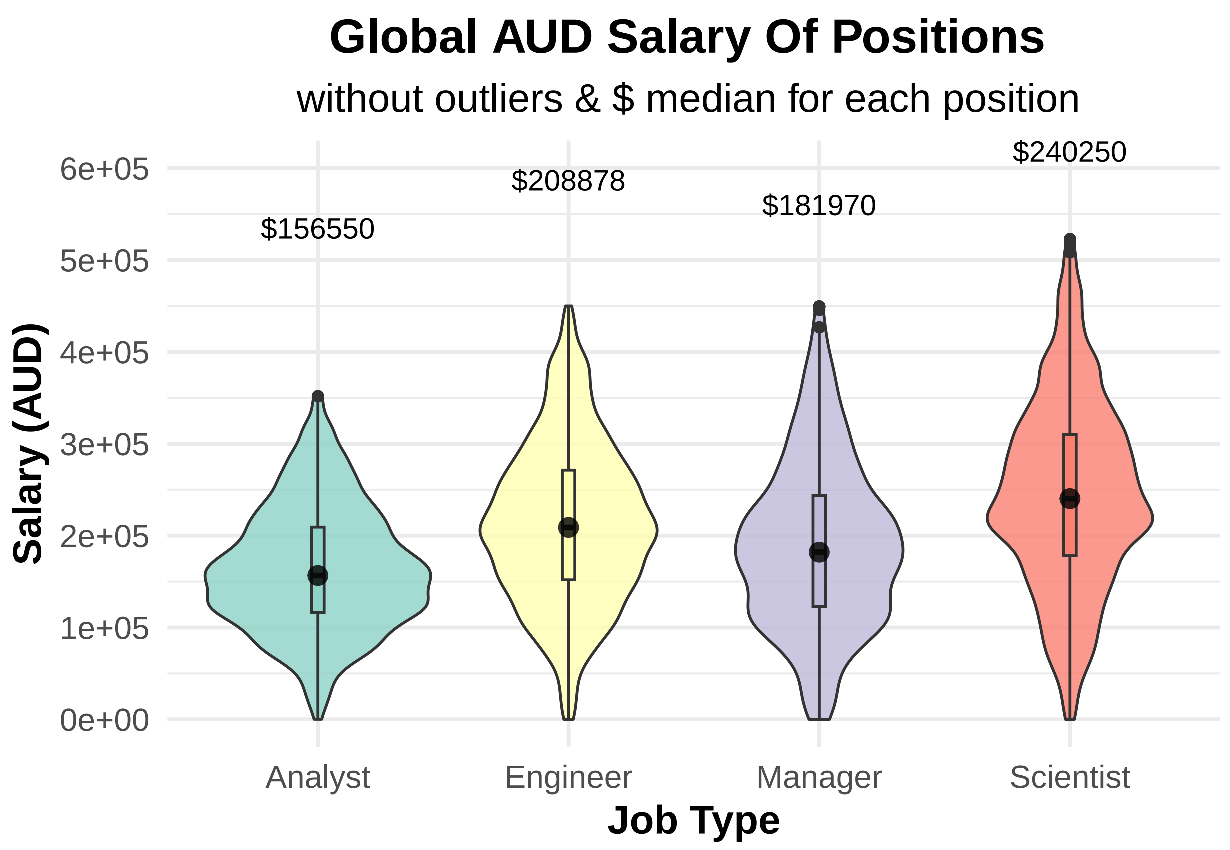
Data Wrangling Section: Joining Kaggle DataSet & Stack Overflow Dataset

```r
#calling the libraries to be used.
library(dplyr)
library(tidyr)
library(dplyr)
library(tidyverse)

#organising the column order in both data sets so they are the same, in order: country, jobtype, experi
KaggleDataSet <- KaggleDataSet[,c(2,4,1,3,5)]
stackoverflowdata2023V2 <- stackoverflowdata2023V2[,c(1,2,5,4,3)]

#joining the rows, this is like a full join, not joining on column as there are no primary keys.
JoinedDataSets <- bind_rows(KaggleDataSet,stackoverflowdata2023V2)

#changing the names of specific countries so it will correctly join with the map data in the future. If
JoinedDataSets <- JoinedDataSets %>%
  mutate(Country = case_when(
    Country == "Bolivia, Plurinational State of" ~ "Bolivia",
    Country == "Czechia" ~ "Czech Republic",
    Country == "Hong Kong" ~ "China",
    Country == "Hong Kong (S.A.R.)" ~ "China",
    Country == "Iran, Islamic Republic of" ~ "Iran",
    Country == "Iran, Islamic Republic of..." ~ "Iran",
    Country == "Korea, Republic of" ~ "North Korea",
    Country == "Republic of Korea" ~ "North Korea",
    Country == "Moldova, Republic of" ~ "Moldova",
    Country == "Russian Federation" ~ "Russia",
    Country == "The former Yugoslav Republic of Macedonia" ~ "North Macedonia" ,
    Country == "Türkiye" ~ "Turkey" ,
    Country == "United Kingdom" ~ "UK",
    Country == "United Kingdom of Great Britain and Northern Ireland" ~ "UK",
    Country == "United States" ~ "USA",
    Country == "United States of America" ~ "USA",
    Country == "Viet Nam" ~ "Vietnam",
    TRUE ~ Country
  ))
```

Violin Boxplot For AUD Salary For Data Positions

```r
#calling library to be used
library(ggthemes)

#creating a function that calculated the outliers
remove_outliers <- function(x) {
    # Calculate the first and third quartiles
    Q1 <- quantile(x, 0.25)
    Q3 <- quantile(x, 0.75)
    # Calculate the interquartile range (IQR)
    IQR <- Q3 - Q1
    # Define the lower and upper bounds for outliers
    lower_bound <- Q1 - 1.5 * IQR
    upper_bound <- Q3 + 1.5 * IQR
    # Return the data points within the bounds
    x[x >= lower_bound & x <= upper_bound]
}
# Remove outliers in each group
JoinedDataSetsNoOutliers <- JoinedDataSets %>% group_by(job_type) %>%
    filter(salary_in_aud %in% remove_outliers(salary_in_aud))
# Ungroup the data frame
JoinedDataSetsNoOutliers <- ungroup(JoinedDataSetsNoOutliers)

#creating violin plot of the data positions against the AUD salary, with a data point stating the media
ggplot(data = JoinedDataSetsNoOutliers, aes(x = factor(job_type), y = salary_in_aud, fill = job_type)) 
  geom_violin(alpha = 0.8, trim = T) + #calling violin and setting trim to true, makes data look cleane
  geom_boxplot(width = 0.05) +
  geom_point(stat = "summary", fun = "median", size = 3, colour ="black", alpha = 0.8) + #adding a poin
  geom_text(stat = "summary", fun = median, aes(label = paste0("$", round(..y.., 0))), vjust = -16, col
  scale_fill_brewer(palette = "Set3") + #using a colour palette that is aesthetic
  scale_y_continuous(limits = c(0,600000), n.breaks = 6) + #setting the limits & breaks to the y axis s
  labs(
    title = "Global AUD Salary Of Positions ", #adding labels and titles to the axis
    x = "Job Type",
    y = "Salary (AUD)",
    subtitle = "without outliers & $ median for each position "
  ) +
 theme_minimal(base_size = 15) + #adding a theme which increases the font size to make it easier to rea
  theme( #making all the axis and titles bold so its clear
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.position = "none"
  )
```

```
## Warning: The dot-dot notation (`..y..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(y)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Who

The audience for this specific visualisation is the same as mentioned in the executive summary, individuals that are currently within or aspiring to break into the data field. This plot specifically is catered to financial transparency and therefore gives an indication of what a median salary looks like per position. Therefore, the audience would be for individuals that value monetary as an indication for value in the data position.

What

The action for this plot is to recommend and empower the audience with information to recommend a data position type based off median salary and steer their preference to a desired data position type. From the results, its clear that the "scientist" position has the highest median salary & also the highest maximum recorded salary, with the data analyst having the lowest median salary income.

When

The plot is considered static and therefore reveals all data at once.

How

The parameters relating to the action is the y-axis that shows the salary in AUD currency & the geom_point which has the median for each data type. The exploratory characteristics of this plot that could be considered superfluous is the colour filling each of the violin plots, however this aids in the visual presentation of the distribution of the salaries. An explanatory visualisation that could be considered superfluous is perhaps a bar graph that is made up of blocks, where each block represents a value like $10,000. Therefore each bar for a data position would have the applicable number of blocks to represent the maximum pay.

Justification

A violin plot was selected as the graph type for representing AUD salary for the positions. The violin plot provides the distribution of the salary via its thickness, thus adding another dimension to the graph and giving insight to the median and tail ends of the pay spectrum. In addition to the violin plot, a geom_point label was added that identifies each data positions median value. An overlap of the boxplot graph was used to provide familiarity, and also clearly display the IQR of each data position. Pre-attentive attributes that were used include colour hue that is within each of the violin plots as this enables an easier perception of the distribution of values in each data position. The data position plots were ordered in ascending order from the maximum value in their violin plots. The graph also uses a theme which positions and has typeface characteristics (font, bold) which give a clean aesthetic appeal that is simple and concise.

Recommendations:

1. To have data that represents further monetary aspects about each data positions. Such aspects could include average or median bonus each year, questions regarding if they got any other additional compensation such as stock options, sign on bonus, re-location bonus or any other monetary contribution that's within their total yearly compensation.
2. To have information about the average working week hours for each data position, this will then allow a representation of the hourly rate per position type. This would provide value as although a salary may be 30% more, the working hours may be 40% more each week, these aspects should be considered when presenting pay of the positions.

Road Blocks:

There were various roadblocks when preparing this visualisation. They are as follows:

1. Mean vs Median: It was decided to use median over mean as it isn't affected by outliers to the extent averages are.

2. Outliers: Outliers were present within the data set, to resolve this issue the data was grouped by "job_type" and an IQR was calculated for each position using a function found online, there looks to be no outlier function in R. Initially grouping the data by job_type wasn't implemented and the outliers were calculated for the data set as a whole, this still worked but destroyed a lot of individual trends within each data position. The method of grouping by job type enables a greater scope of the position trends to give more valuable data and thus insights from the graph.

3. Job Type Classification: This road block will be present in all visualisations. The classification of the job_type category was not present in either data set, with each containing their own variation of title labels under distinct column names, e.g. DevType, Job_Category. For example, within the Kaggle dataset, Data Analyst was determined if the job_category included "Data Analysis" & "BI and visualisations". The argument could be made a Business Intelligence position is more suited towards "BI and visualisations". This trend is coherent when classifying the job types for both data sets, therefore the classification of jobs is subjective and open ended to bias as positions within data although may have distinct names, they can overlap in many duties and thus can be mis-labelled or appear as two position types.To overcome this road block, assumptions had to be made in classifying the job categories, but I agree with the decisions made.

4. Salary Conversion: The salary conversion within this report is static and does not adjust real-time to conversions. This must be considered as major fluctuations in currency, like Yen at the moment, can influence the distribution within the graphs. Furthermore, some of the data sets (Kaggle) was carried though multiple years & therefore doesn't account for inflation or trends in the currency, e.g. AUD increases 5% year on year. These aspects arent revealed in the end graph, therefore should be acknowledged within the interpretation of results.

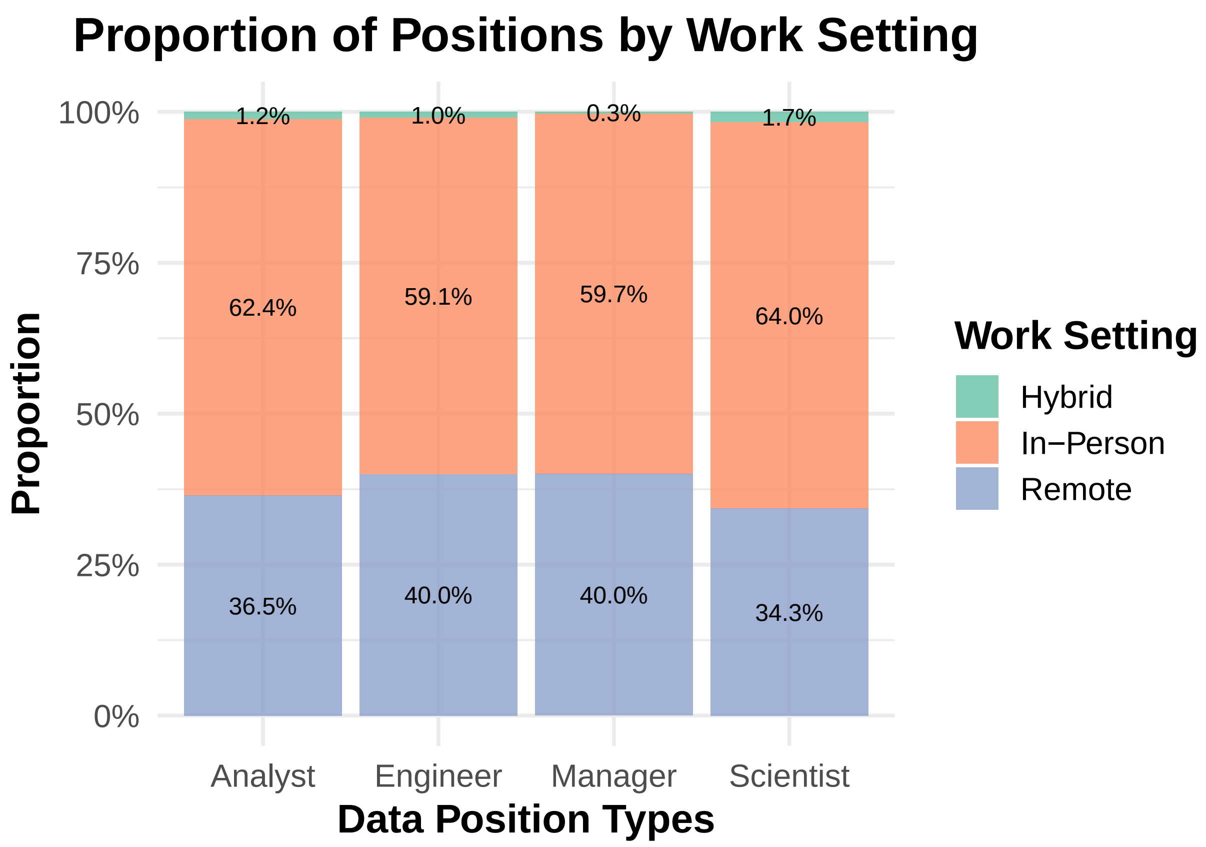
5. Salary Representation: Within both data sets there is a salary representation for each participant, however the employment type for each participant is not equal. The employment type within the Kaggle dataset includes: freelance, full-time, contract & part time. The salary states to participants if not earning a set value per year (salary), then to give best estimate of their earning enabling subjectivity and therefore bias. Furthermore, within the stack overflow data, not being employed is an option as the survey caters toward students or participants actively seeking jobs. However, in these cases the totalcomp, (salary field) is NA in those cases as they have no formal job. Based on these factors, the interpretation and reliability of the salaries should considered as imperfect and perceptions from these values should be mindful. The outliers function would remove alot of the values that arent considered normal, but however this can mean that alot of the data is artificially generated by the particpants.

Bar Graph For Data Positions: Proportion of Location Setting & Proportion of Experience Level

```r
# Removing data where the work setting is NA
JoinedDataSets.Location <- JoinedDataSetsNoOutliers %>% filter(!is.na(work_setting))
# Removing data where the experience level is NA
JoinedDataSets.Experience <- JoinedDataSetsNoOutliers %>% filter(!is.na(experience_level))

# Calculate the percentages for work setting
JoinedDataSets.Location <- JoinedDataSets.Location %>%
  group_by(job_type, work_setting) %>%
  summarise(count = n()) %>%
  mutate(percentage = count / sum(count))
```

```
## ‘summarise()‘ has grouped output by ’job_type’. You can override using the
## ‘.groups‘ argument.
```

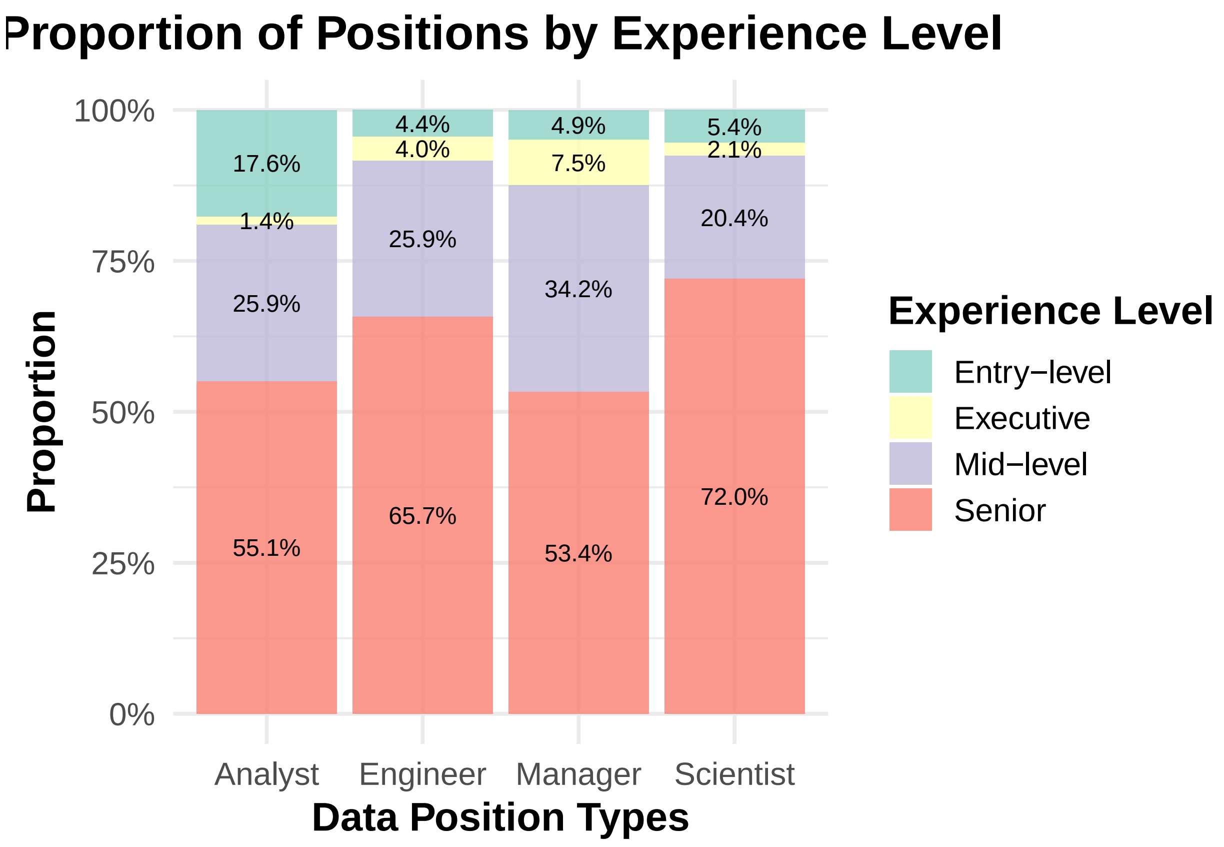
```r
# Create the bar graph with percentage labels
ggplot(data = JoinedDataSets.Location, aes(x = job_type, y = percentage, fill = work_setting)) +
  geom_bar(stat = "identity", position = "fill", alpha = 0.8) +
  geom_text(aes(label = scales::percent(percentage, accuracy = 0.1)),
            position = position_fill(vjust = 0.5), size = 3) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Proportion of Positions by Work Setting",
    x = "Data Position Types",
    y = "Proportion",
    fill = "Work Setting"
  ) +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.title = element_text(face = "bold"),
    legend.position = "right"
  )
```



```
# Calculate the percentages for experience level
JoinedDataSets.Experience <- JoinedDataSets.Experience %>%
  group_by(job_type, experience_level) %>%
  summarise(count = n()) %>%
  mutate(percentage = count / sum(count))
```

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

```
# Create the bar graph with percentage labels
ggplot(data = JoinedDataSets.Experience, aes(x = job_type, y = percentage, fill = experience_level)) +
  geom_bar(stat = "identity", position = "fill", alpha = 0.8) +
  geom_text(aes(label = scales::percent(percentage, accuracy = 0.1)),
            position = position_fill(vjust = 0.5), size = 3) +
  scale_y_continuous(labels = scales::percent) +
  scale_fill_brewer(palette = "Set3") +
  labs(
    title = "Proportion of Positions by Experience Level",
    x = "Data Position Types",
    y = "Proportion",
    fill = "Experience Level"
  ) +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.title = element_text(face = "bold"),
    legend.position = "right"
  )
```



Who:

The intended audience is still the general theme mentioned prior, but is catered towards individuals that are: - Wanting to know the difficulty of breaking into a data position type, therefore the frequency of entry level positions in each data position. - Wanting to know the proportion of work experience within each data position, a larger proportion of higher experience levels can suggest this field favours experts. - Proportion of work setting for each of the data positions, therefore if an individual wants to work remote, he/ she may be inclined towards a position more than others.

What:

Alike the first plot, its action is to educate, empower and ultimately recommend a data position type that is best suited to the individuals preferences. From the results, its reveals that all have relatively similar work settings but a data manager doesnt have at all or extremely small potential to work hybrid, but otherwise the trends are similar across all positions. Trends for work experience show that the data analyst have higher proportions of entry level positions out of all positions, therefore would indicate the best suited position for a graduate or individual new to the field. Otherwise, for the position with the other experience levels its fairly equal, with data scientist and engineer having a greater proportion of senior positions from the sample.

When:

Alike the first plot, the data is static and presents all data at once.

How:

The parameters within the data set related to the action is the frequency of the employment type for each position type and therefore displayed as an identity within the bar graph. For the work setting stack bar graph, the related parameter for the action is the frequency of the work setting and therefore in proportion to the total recorded work settings types for that data position. A superfluous visualisation could perhaps be a pie chart for each data position, as this can skew the interpretation of the proportions and cause confusion.

Justification:

Pre-attentive attributes used is alike the first plot, colour was used that provides clear distinction between the work setting in each of the position types. The colour set used also enables clear distinction in colour, therefore the same colour of shades was not used. The addition of geom_text for the percentage of each portion within the bar graph provides a clear interpretation of results, this also aids in the visual clarity of the graph.

Recommendations:

1. Having data that visualises or finds out if the work setting is fixed or dynamic. For example, an employee may be able to work hybrid but also have the option to work remote for a portion of the year. This possibility is not able to be presented as it only assumed fixed work settings, which in the tech industry is uncommon. A new metric could be developed that scores a position type for work setting flexibility, for example a manager would have less flexibility than a worker as they may be required to available in-person more.
2. To have data that classifies the experience levels by the years within a data position type. This would be valuable as its common for an individual within the tech industry to pivot to many roles, therefore a software engineer may believe they have been a data engineer as they have been exposed to data bases. This classification would provide clear and un-biased distinction between levels.

Road Blocks:

There was multiple road blocks for this visualisation and also those inherited from the first visualisation:

1. Mentioned prior, the classification of the position type was influenced by subjectivity, therefore the true trends between positions can be argued and degraded in its accuracy.

2. The classification of experience level is based off survey results and therefore subjective, there appears to be no indication level to get the labels, such as the years within the field. In addition, the stack overflow data set doesn't have information relating to experience level, therefore this data set provided no value in the plot. The plot used a dataset that removed where experience_level is NA, therefore it doesn't affect the proportions, but the plot doesn't utilise all data collected and must be considered.

3. Due to the Kaggle data set being collected over multiple years, its highly possible a person changed roles through and completed the survey with a different experience level & even work setting. The data doesn't account for this and doesnt have the ability to show progression of a position over time. This road block couldn't be resolved as there is no way to identify if a person completed the survey more than once, but rather due to the collection method.

World Map for Data Position Salaries

```r
#calling libraries used
library(ggplot2)
library(tidyverse)
library(magrittr)
library(ggthemes)
library(mapproj)
```

```
## Loading required package: maps

##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```

```r
library(viridis)
```

```
## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:maps':
##
##     unemp
```

```r
#subsetting the data by country and position to determine the mean and median
JoinedDataSets.Worldbycountryandjobtype <- JoinedDataSetsNoOutliers %>%
  group_by(Country, job_type) %>%
  summarize(
    mean_salary = mean(salary_in_aud, na.rm = TRUE),
    median_salary = median(salary_in_aud, na.rm = TRUE))
```

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

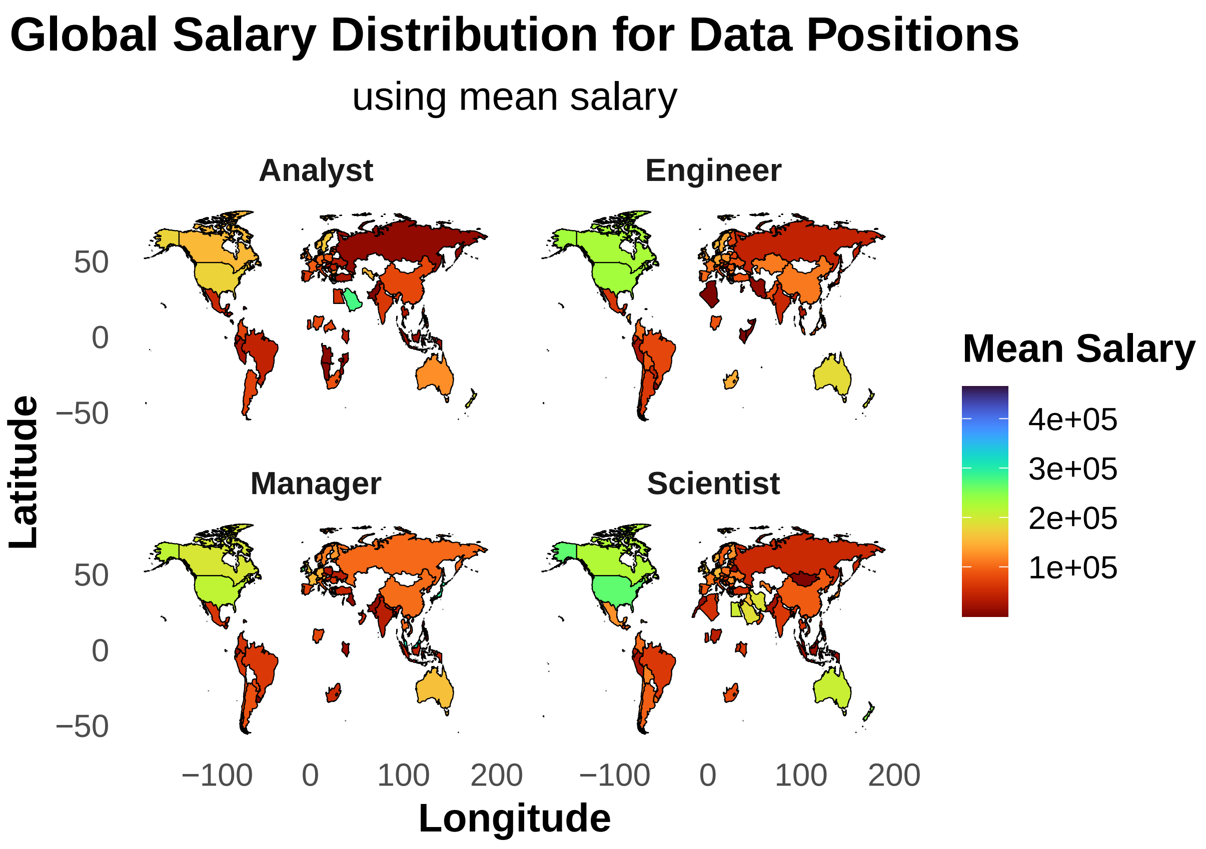
```
#changing name of company_location to region so it can merge with mapdata
names(JoinedDataSets.Worldbycountryandjobtype)[names(JoinedDataSets.Worldbycountryandjobtype) == 'Count

#calling built in map_data tibble
mapdata <- map_data("world")
#merging both tibbles on region column
mapdatatest2 <- left_join(JoinedDataSets.Worldbycountryandjobtype, mapdata, by="region")
```

```
## Warning in left_join(JoinedDataSets.Worldbycountryandjobtype, mapdata, by = "region"): Detected an u
## i Row 1 of ‘x‘ matches multiple rows in ‘y‘.
## i Row 35137 of ‘y‘ matches multiple rows in ‘x‘.
## i If a many-to-many relationship is expected, set ‘relationship =
##   "many-to-many"‘ to silence this warning.
```

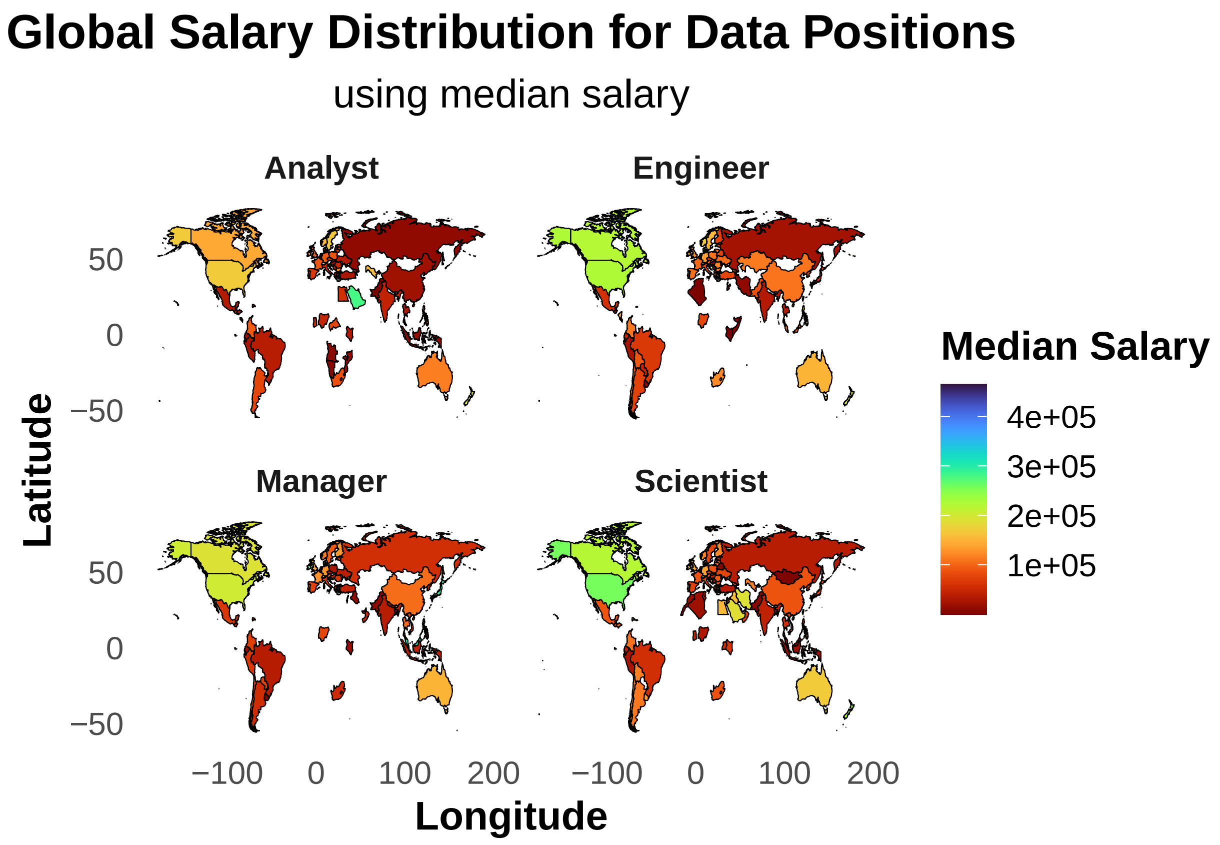
```
#excludes all countries on map that have no data for mean_salary
# testmap <- mapdata %>% filter(!is.na(mapdata$mean_salary))

# #creating map visualisation with fill as mean_salary, each job type has its own world map.
# KaggleDataSet.Map.Mean <- ggplot(data = mapdatatest2, aes(x= long, y = lat, group = group)) +
#   geom_polygon(aes(fill = mean_salary), color = "black") + scale_fill_viridis(option = "turbo") + the
# KaggleDataSet.Map.Mean
#
# #creating map visualisation with fill as median_salary, each job type has its own world map.
# KaggleDataSet.Map.Median <- ggplot(data = mapdatatest2, aes(x= long, y = lat, group = group)) +
#   geom_polygon(aes(fill = median_salary), color = "black") + scale_fill_viridis(option = "turbo") + t
# KaggleDataSet.Map.Median
#
# #data set which states which values dont merge with the mapdata as the long & lat variables are NA.
# dummydata <- mapdatatest2 %>% filter(is.na(long))



#creating world map of mean salary for each country
KaggleDataSet.Map.Mean <- ggplot(data = mapdatatest2, aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = mean_salary), color = "black", size = 0.2) +
  scale_fill_viridis(option = "turbo", name = "Mean Salary", direction = -1) + #switching the turbo col
  labs(
    title = "Global Salary Distribution for Data Positions",
    subtitle = "using mean salary",
    x = "Longitude",
    y = "Latitude"
  ) +
  facet_wrap(~ job_type) + #facet wrap against the data type so a map for each position is created
  theme_minimal(base_size = 15) +
  theme( #adding themes to its more visually pleasing
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.title = element_text(face = "bold"),
    strip.text = element_text(face = "bold"),
    panel.grid = element_blank()
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
KaggleDataSet.Map.Mean
```



```
#creating world map of median salary for each country
KaggleDataSet.Map.Median <- ggplot(data = mapdatatest2, aes(x = long, y = lat, group = group)) +
  geom_polygon(aes(fill = median_salary), color = "black", size = 0.2) +
  scale_fill_viridis(option = "turbo", name = "Median Salary", direction = -1) +
  labs(
    title = "Global Salary Distribution for Data Positions",
    subtitle = "using median salary",
    x = "Longitude",
    y = "Latitude"
  ) +
  facet_wrap(~ job_type) +
  theme_minimal(base_size = 15) +
  theme(
    plot.title = element_text(hjust = 0.5, face = "bold"),
    plot.subtitle = element_text(hjust = 0.5),
    axis.title.x = element_text(face = "bold"),
    axis.title.y = element_text(face = "bold"),
    legend.title = element_text(face = "bold"),
    strip.text = element_text(face = "bold"),
    panel.grid = element_blank()
  )
KaggleDataSet.Map.Median
```



Who:

The general audience is the same as prior but is catered again towards individuals that value position income and also country. It gives the viewers a perspective to move to a different country that meets their income expectations/aspirations for the data position. For example, if a person lived in Australia, they might be inclined to move to a country in South East Asia as it has a higher median/ mean annual income for a data position.

What:

The intended action alike the other plots is to educate, empower & ultimately recommend a data position for the audience that account for annual pay. It also educates pay trends for each position globally, therefore if a position has a higher mean/ median for a aspiring data type, then the individual should consider moving there.

When:

The data is static and shows all information at once.

How:

The parameters related to the action is the pay determined through median and mean and the location the employee works. Exploratory visualisations that are superfluous to the communication could be perhaps rather than a world map, a plot that has the flag instead. This would be considered excessive as a simple label would be clearer and it doesn't give a sense of scope for neighboring countries and therefore positioning.

Justification:

Pre-attentive attributes used include a colour scale that has a clear distinction between levels, also using a colour set that is more vibrant opposed to pastel aids in this distinction. The turbo colour set was used and inversed, this enables an easier interpretation as red is considered negative (low income) & blue is positive r the opposite (higher income). Removing countries that had no calculated mean or median was chosen, as this reduces the clutter of the map. Labeling was considered but it caused cluttering of information, using an interaction tool like plotly could be used and when hovering over a country, the name and exact value could be displayed.

Recommendation:

1. To have further information regarding states for relevant countries, such as USA & Australia. This would only be applicable to certain countries. Another option could to have a representation of continent, but I doubt this would give more information than take away.
2. To create another set of world maps which has the country of residence and the company location equal, then join this to the map tibble of the world. This would give indication of countries where the employees reside and work are the same, opposed to working for a company based in the USA but living in India for example. The justification of this is item 2 in the road blocks section.

Road Blocks:

- Due to the world maps data set, it doesn't account for specific countries and has some merged together, such as the neglect of Scotland, England and Wales as the United Kingdom. This therefore loses some of the data's value. However the data has a relatively good representation of countries, e.g. it doesn't only show continents.
- To determine the median/ mean income for the country, the value of employee residence in the Kaggle data set was used. This is important to consider as it also included the country in which the company was located. Therefore, if a person worked remote for a US company, but resided in India, the income for that data position would be attributed to India. This must be considered within the interpretation, as it perhaps can inflate a countries mean/ median income for a data position. This finding is supported as South East Asia is popular for digital nomads which work remote, therefore there may be an incorrect interpretation of more fruitful data position availability and income in that country as they all originated from other countries, but workers live in such places due to other factors, like weather, expenses ect. This issue is only applicable for the Kaggle dataset as the stack overflow only lists the country the person lives in. This finding was tested by creating KaggleDataSetLivesAndWorksSameCountry data frame, which excluded rows where the employee residence was different to the company country location, this however only removed 120 observations and therefore wasnt considered influential & therefore wasnt further persued.

Conslusion

From the variety of visualisations, a range of trends and insights are displayed. Regarding the aim into trying to determine an optimal data position, it can be argued that data analyst is the best data position due to the highest proportion of entry levels positions recorded. A data position that is more relevant to a senior then a data scientist is viewed as the best option. Regarding country, USA consistently has the highest or high incomes for the data positions and could perhaps be the best country to work in. However, all these findings can be argued due to personal opinions and wants. Th primary aim of this report has been fullfilled and has given you the viewer the tools to look and analyse various data positions to help find a data career that is suited to you. Thankyou.