R Notebook

Code

Data Summary

Worked on the data from the 2017 American Community Survey (ACS) 5-year Public Use Microdata Samples (PUMS), which is a sample of the actual responses collected by the American Community Survey between 2013-2017, and split into population and household characteristics.

However, I will be using the Population characteristics data as I am more interested in determining the demographics of population. For my analysis I will be working on a subset of the PUMS population dataset, specifically I will be working with the following variables namely

- REGION - Defines the regions across the USA
- ST - Defines the different states
- AGEP - Defines the age of the individuals data recorded
- COW - Defines the class of worker an individual is working as
- JWMNP - Time it takes to travel to work
- JWRIP - Defines how an individual travels to Work
- JWTR - Defines the type of transportation an individual uses for work
- MAR - Maritial status
- SCH - School enrollment
- SCHG - Grade level attended
- SCHL - Educational attained
- SEX - Gender
- WKHP - Hours worked per week last year
- WKL - Last worked date
- WKW - Weeks worked during last year
- FOD1P - First Field of Degree
- FOD2P - Second Field of Degree
- INDP - Job industry working in
- JWAP - Arrival time at job
- JWDP - Departure time for Job
- POVPIP - Poverty Index

After loading the required libraries, and the data, I have done the following pre-processing of the data, a rough overview would include the following, with further details in the Methodology section later.

- Gathering the data of different regions from four files to one
- Creating the mappings from the coded values of the variables to their actual categorical values
- Type manipulations and factoring for further anlaysis and plotting

Methodology

To load the data together into one data frame, I used the *dplyr* package's *bind_rows* instead of the base R's due to advantages such as dealing with tibbles and effecient computation. But to use the above we need the columns of combining rows dataframes to have the same data type, the dataset used had different data types, such as the *pums_a* had *char* type for the States column, however others namely *pums_b*, *pums_c* and *pums_d* had *numeric* type for the same column.

Dealing with *NA's* was also segmented, in the sense that I didn't remove the incomplete cases altogether as that then only gave me a data of just 4700 rows, which was barely helpful for analysis. So didn't remove incomplete cases as we can remove when working on subset, if we remove now then there is a possibility that working subset may have none and we loose lot of data

Once the above was obtained, next major task was to map the coded values to their categorical variables for easier understanding and analysis, for example States in US had to be mapped to their values like Alabama for AL etc. To carry this out I have used the *dplyr* and the *tidyverse* packages for piping and tidying the mapping from the excel file. To get the values of each variables, I used the excel file of the pums_data_dictionary for 2013-2017, and used tidyverse constructs such as gathering, spreading and separating the to clean the data and store as dataframe of the mapping for each variable. For example, our dataset has 52 states coded as 1 - 52, tidying the above we got a df that maps these coded 1 - 52 to their correspondong state names and values. The mapping dataframe can be used later on in analysis via *joins* to get the coded variables corresponding values

Hide

```
#load libraries
library(tidyr)
library(tidyverse)
library(ggplot2)
library(dplyr)
load("df_a.RData")
load("df_b.RData")
load("df_c.RData")
load("df_d.RData")
```

Hide

```
#the State columns in pumsa is char type and double in rest, therefore type casting
to adjust that, else error while binding
df_a_mod$ST <- as.numeric(df_a_mod$ST)

#bind the 4 dataframes into 1, don't remove incomplete cases as we can remove when
working on subset, if we remove now then there is a possibility that working subset
may have none and we loose lot of data
df_abcd <- bind_rows(df_a_mod, df_b_mod, df_c_mod, df_d_mod)

#save the above dataframe
save(df_abcd, file = "pums_abcd.RData")

#remove the 4 dataframes
rm(df_a_mod, df_b_mod, df_c_mod, df_d_mod)

#gathering the data dictionaries for mapping
data_dict <- read_csv("csv_pus/PUMS_Data_Dictionary_2013-2017.csv", col_names = FAL
SE, skip_empty_rows = TRUE, progress = show_progress())

#taking unique as else key error based on tidy data rule as ST and REGION are repea
ted, one for Housing and other for person, key error check github issue
data_dict_filtered <- unique(data_dict) %>%
  na.omit(data_dict) %>%
  select(X2, X5, X7) %>%
  filter(X2 %in% colnames(df_abcd))

colnames(data_dict_filtered) <- c("VAR", "CODE", "VAL")

#storing the dictionary for each of our selected columns
for(name in colnames(df_abcd)) {
  dict_temp <- data_dict_filtered %>%
    filter(VAR == name) %>%
    spread(VAR, CODE)


  assign(paste0("dict_", name), dict_temp)
}
```

Hide

```
# modifying the dict_ST for better analysis and cleaner output as bigger names migh
tnbot fit onto display and converting the ST to numeric as it is in numeric in df_A
bcd, compartible join operation
dict_ST <- dict_ST %>%
  separate(VAL, into = c("State", "Code"), sep = "/")

dict_ST$ST <- as.numeric(dict_ST$ST)

head(dict_ST)
```

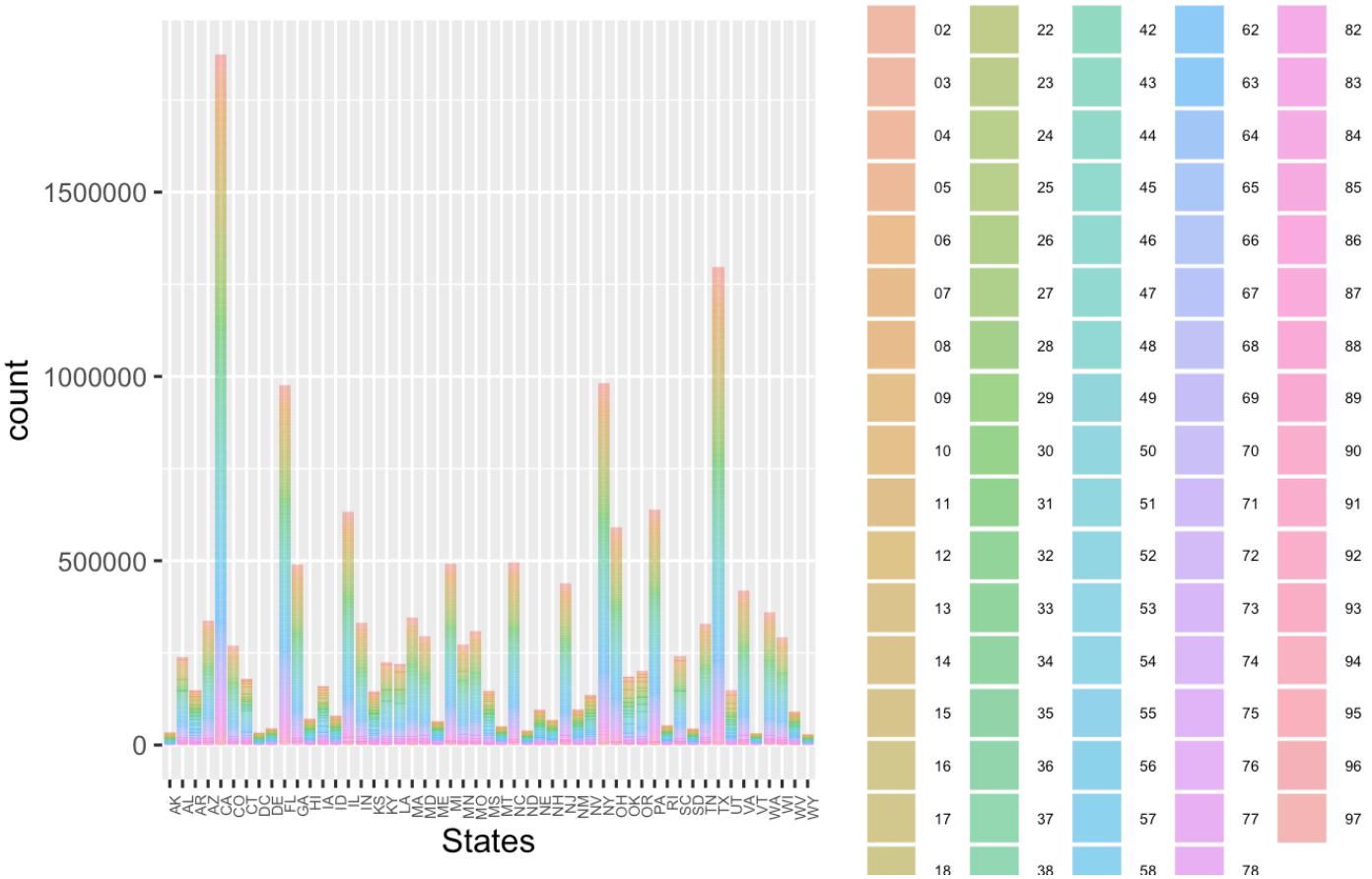
Findings

- different population age groups across states As is expected in reality the development and modernisation of California has resulted in a growth in population it is the most populous state following is Texas. Being at the central and western side they have best weather over all and hence is the population and preferenc of all the age groups.

Hide

```
#demographics of population across different states
#different age groups across states
df_abcd %>%
  select(ST, AGEP) %>%
  group_by(ST, AGEP) %>%
  summarize(
    count = n()
  ) %>%
  left_join(dict_ST, by = "ST") %>%
  select(Code, AGEP, count) %>%
  ggplot(aes(x = Code, y = count, fill = AGEP)) +
  geom_bar(alpha = 0.5, stat = "identity") +
  labs(x = "States", fill = "Age") +
  theme(
    axis.text.x = element_text(angle=90, hjust = 1, size = 5),
    legend.text = element_text(size = 5),
  )
```

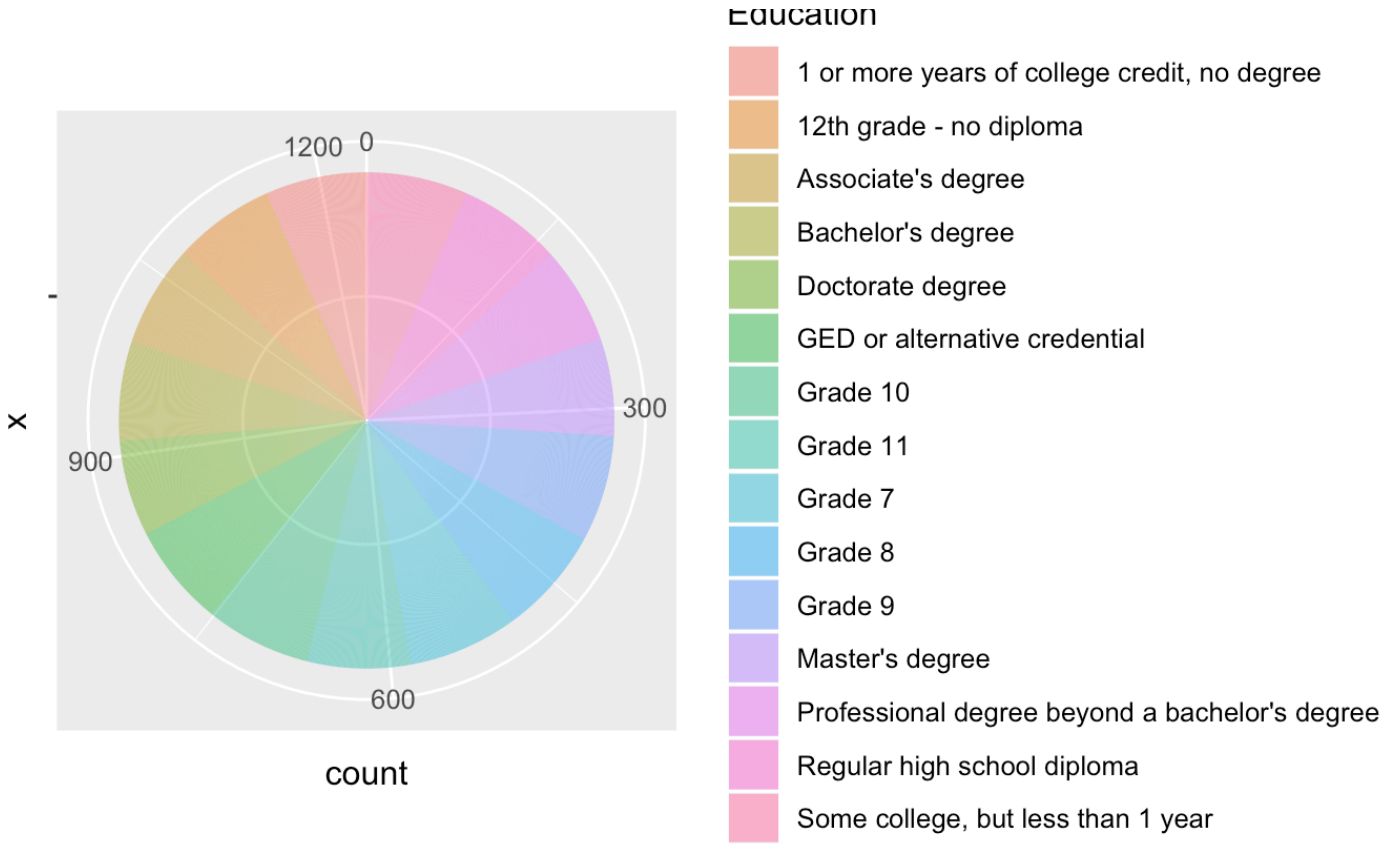
```
Adding missing grouping variables: `ST`
```



- education pattern among different age groups, Quite a 40% are less than 3 years old and age group < 15 aren't working, removing this outlier, gives a better result of the average count of individuals across USA having varying educational backgrounds. Majority of them have a Regular high school diploma atleast and some college

Hide

```
#education pattern among different age groups, quite a 40% are less than 3 years ol
d
df_abcd %>%
  select(AGEP, SCHL) %>%
  left_join(dict_SCHL, by = "SCHL") %>%
  mutate(
    Age = AGEP,
    Education = VAL
  ) %>%
  group_by(Age, Education) %>%
  summarize(
    count = n() / sum(n())
  ) %>%
  na.omit() %>%
  ggplot(aes(x="", y = count, fill = Education)) +
  geom_bar(stat="identity", width = 1, alpha = 0.5) +
  coord_polar("y", start = 0)
```

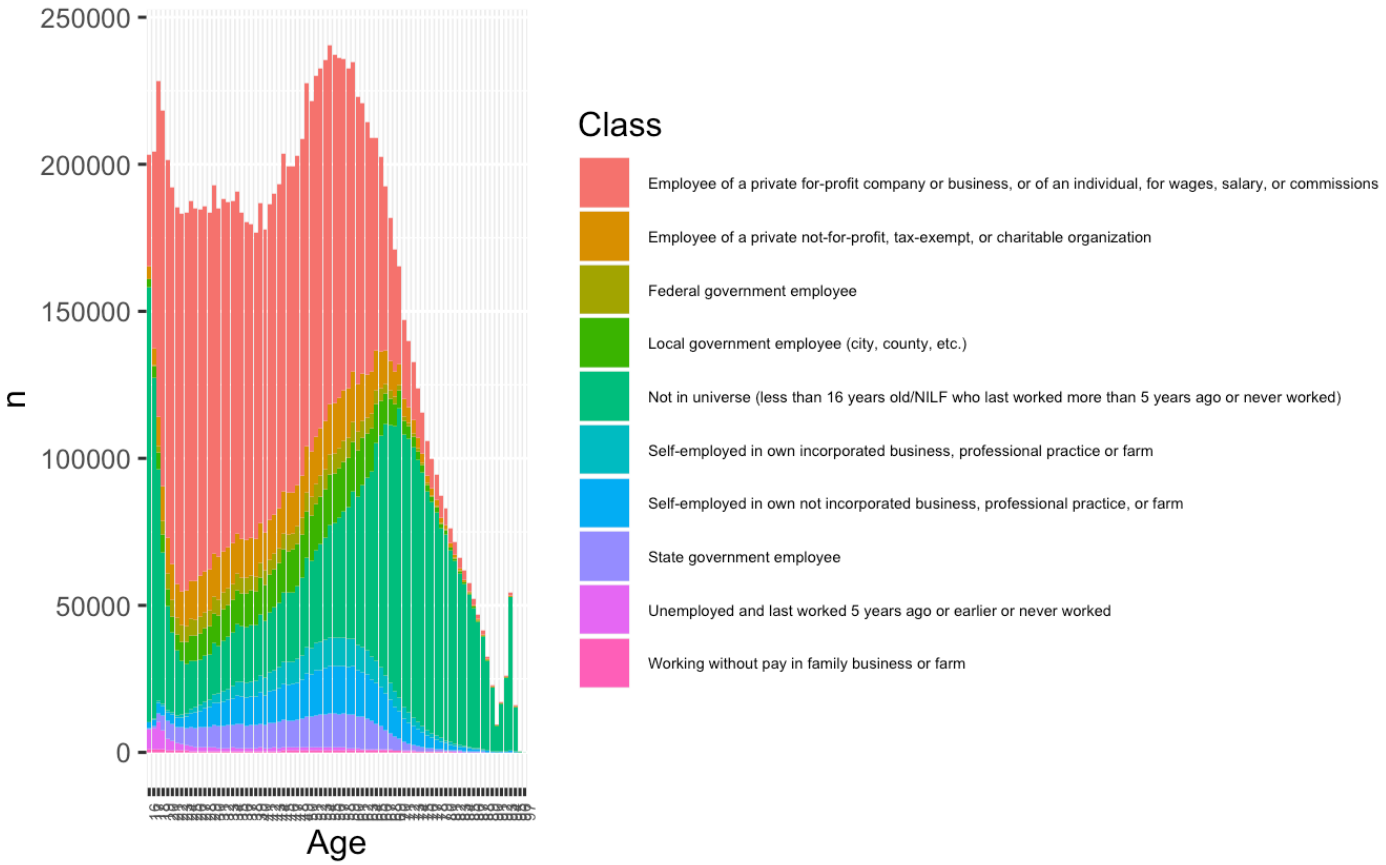


- Class of workers based on age Generally, the aged above 40, are in the government/public sector and the younger geenrarion prefers working in private sector

Hide

```
#class of workers in different age groups as mumeric
dict_COW$COW <- as.numeric(dict_COW$COW)

#agegroup < 15 aren't working
df_abcd %>%
  select(AGEP, COW) %>%
  left_join(dict_COW, by = "COW") %>%
  mutate(
    Age = AGEP,
    Class = VAL
  )%>%
  count(Age, Class) %>%
  filter(Age > 15) %>%
  ggplot(aes(x = Age, y = n, fill = Class)) +
  geom_bar(stat = "identity") +
  theme(
    axis.text.x = element_text(angle=90, hjust = 1, size = 5),
    legend.text = element_text(size = 5),
  )
```

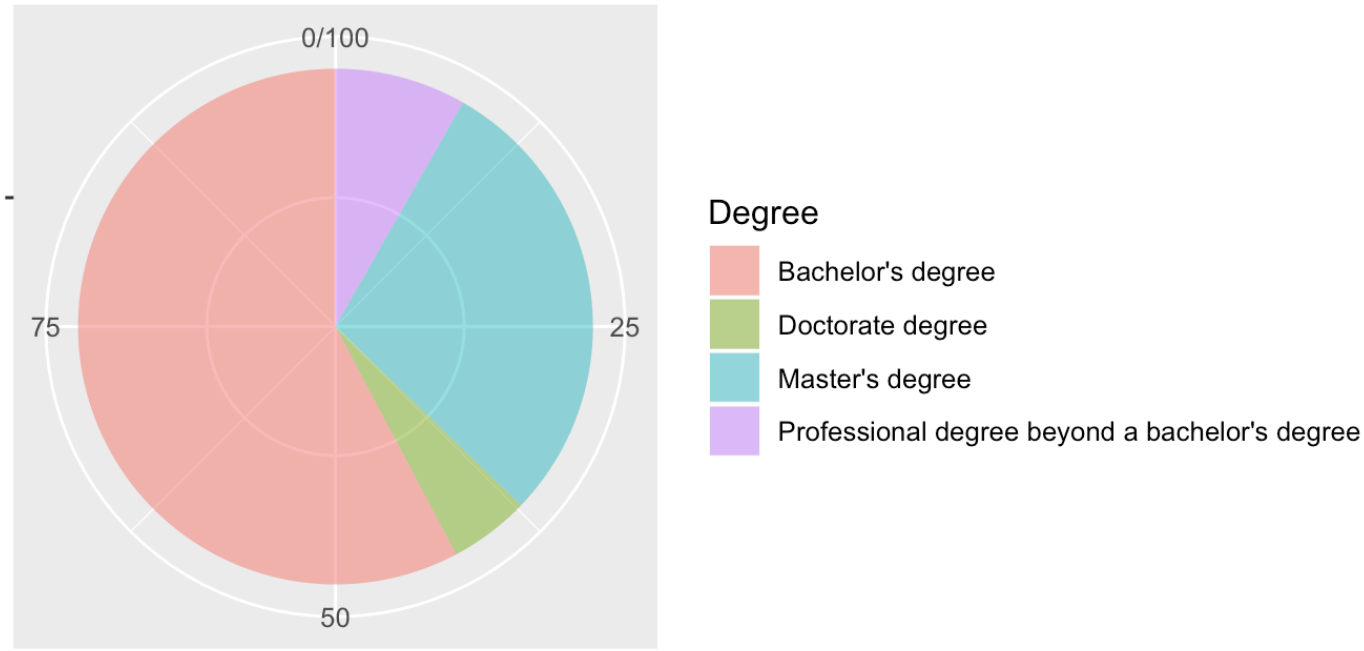


- Percentage of individuals with degree obtained A majority of them have atleast Bachelors degree, and very few go for doctrate and culminate at masters degree.

Hide

```
# Educational qualifications, degrees type Master, bachelors etc among individuals

df_abcd %>%
  select(SCHL, FOD1P, FOD2P) %>%
  na.omit() %>%
  inner_join(dict_FOD1P, by = "FOD1P") %>%
  inner_join(dict_FOD1P, by = c("FOD2P" = "FOD1P")) %>%
  inner_join(dict_SCHL, by = "SCHL") %>%
  mutate(
    Domain1 = VAL.x,
    Domain2 = VAL.y,
    Degree = VAL
  ) %>%
  select(Degree, Domain1, Domain2) %>%
  count(Degree) %>%
  mutate(
    percent = n / sum(n) * 100
  ) %>%
  ggplot(aes(x="", y = percent, fill = Degree)) +
  geom_bar(stat="identity", width = 1, alpha = 0.5) +
  coord_polar("y", start = 0) +
  labs(x = element_blank(), y = element_blank())
```



Hide

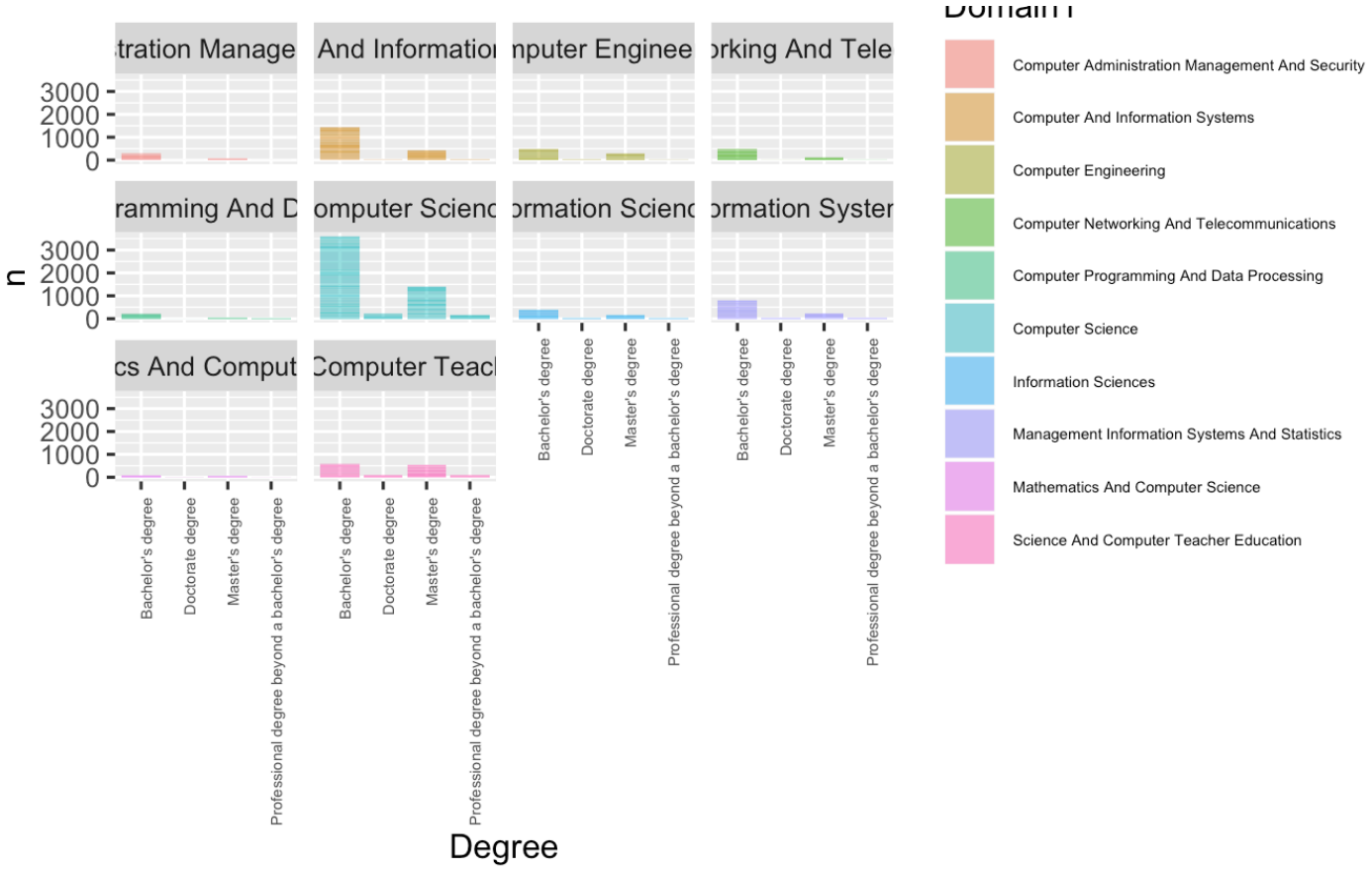
```
NA
```

- Individuals with Computer Science degree and there further degrees Facet wrapping, we can figuee out that in computer science and allied domains, people who had bachelors, also pursued Masters and very few pursuing Doctrate across all the allied domains.

Hide

```
# Facet wrapping the degree across the domain1 or Field of degree 1 among individua
ls

df_abcd %>%
  select(SCHL, FOD1P, FOD2P) %>%
  na.omit() %>%
  inner_join(dict_FOD1P, by = "FOD1P") %>%
  inner_join(dict_FOD1P, by = c("FOD2P" = "FOD1P")) %>%
  inner_join(dict_SCHL, by = "SCHL") %>%
  mutate(
    Domain1 = VAL.x,
    Domain2 = VAL.y,
    Degree = VAL
  ) %>%
  select(Degree, Domain1, Domain2) %>%
  filter(str_detect(Domain1, "Computer|Information" )) %>%
  count(Degree, Domain1, Domain2) %>%
  ggplot(aes(x = Degree, y = n, fill = Domain1)) +
  geom_bar(alpha = 0.5, stat = "identity") +
  facet_wrap(vars(Domain1)) +
  theme(
    axis.text.x = element_text(angle=90, hjust = 1, size = 5),
    legend.text = element_text(size = 5)
  )
```



- ◦ CLass of workers across states California being an IT hub as well hs a mmajority of them

working in Private industry. Not being close in resources for Private sector, and Alaska lags much behind Alabama. Also being a hub , California has more federal government class workers than local government.

State <chr>
Alabama
Alabama
Alabama
Alabama
Alabama
Alabama
Alabama
Alabama
Alabama
Alaska

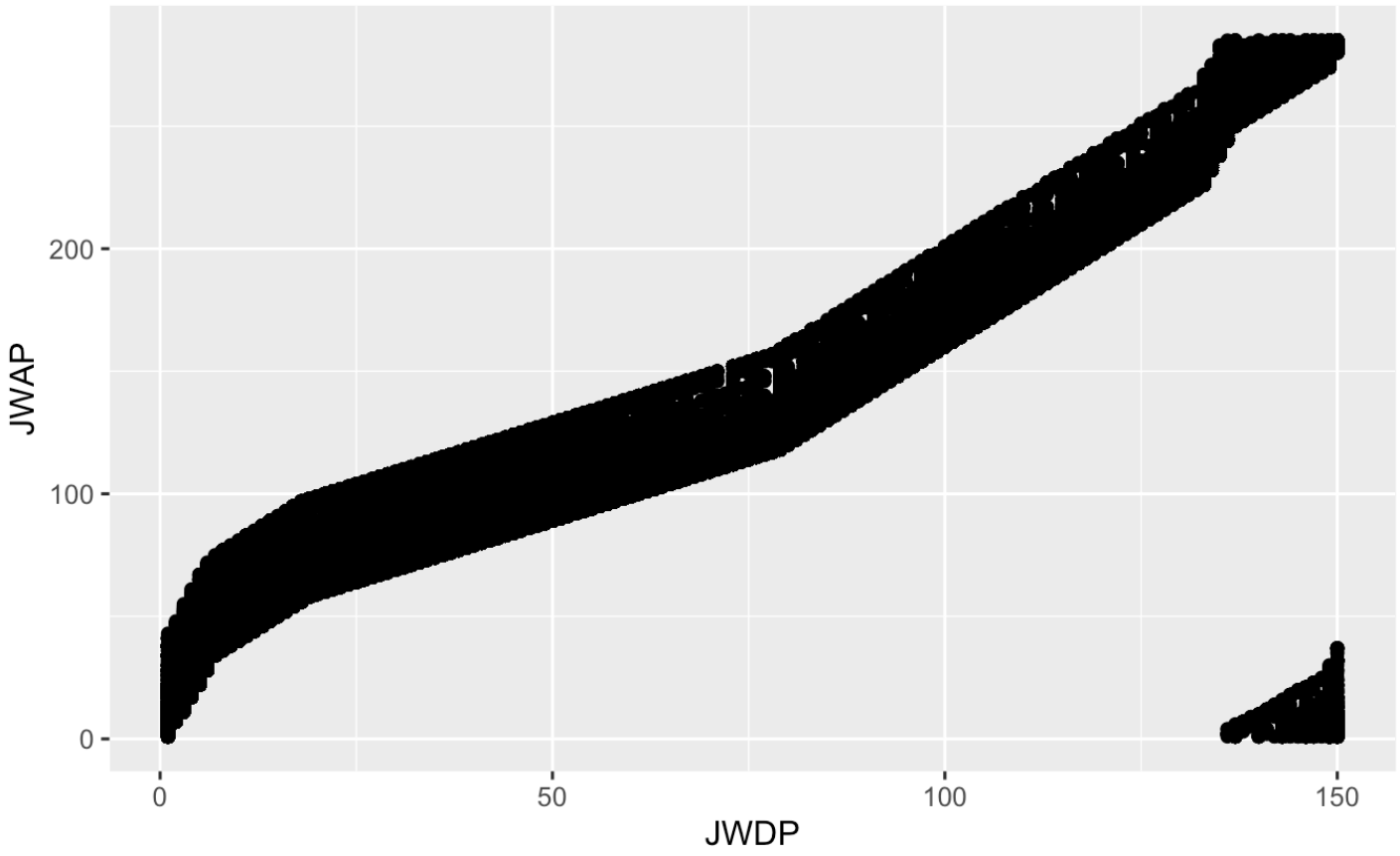
1-10 of 459 rows | 1-1 of 3 columns Previous **1** 2 3 4 5 6 ... 46 Next

- performing a linear regression on arrival and departure time of work

Hide

```
# modifying the type of time
df_abcd$JWDP <- as.numeric(df_abcd$JWDP)
df_abcd$JWAP <- as.numeric(df_abcd$JWAP)
# performing a linear regression on arrival and departure

df_abcd %>%
  select(JWAP, JWDP) %>%
  na.omit() %>%
  ggplot(aes(x = JWDP, y = JWAP)) +
  geom_point()
```



Hide

```
relation <- lm(df_abcd$JWDP ~ df_abcd$JWAP)
relation
```

```

Call:
lm(formula = df_abcd$JWDP ~ df_abcd$JWAP)

Coefficients:
 (Intercept)  df_abcd$JWAP
    -13.5957        0.6583
```

Hide

```
#su mary of relation
options(scipen=999)
summary(relation)
```

```
Call:
lm(formula = df_abcd$JWDP ~ df_abcd$JWAP)

Residuals:
    Min      1Q  Median      3Q     Max
-38.021  -4.043   0.640   4.374 162.937

Coefficients:
                 Estimate   Std. Error t value            Pr(>|t|)
(Intercept)  -13.59572158   0.00777058   -1750 <0.0000000000000002 ***
df_abcd$JWAP   0.65830278   0.00006944    9480 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.617 on 6714755 degrees of freedom
  (9044180 observations deleted due to missingness)
Multiple R-squared:  0.9305,    Adjusted R-squared:  0.9305
F-statistic: 8.987e+07 on 1 and 6714755 DF,  p-value: < 0.00000000000000022
```

Based on the summary above, the equation is y = mx + c, where c = -13.597 and m = 0.6543 are coefficents, and hence the model is y = 0.65483x - 13.597. The intercept represents, the time we don’t leave the office and stay overnight in the office itself, and the slope term in our model says that if we depart 1 minute late, the arrival time also changes, in our case if we deprt 1 unit (30 minute) late, we will reach 1 unit (5 minutes) late.

The required above timelines, can vary from actual value by 0.00007 unit. The residual error of 7.61, denotes that the time of arrival ccan deviate from the true regression line by 7.61 (Approx 8 units) (i.e) 40 minutes maximum. Since the adjusted R-square value is close to 1, the regression explains the observed variance in the response variable. Since the F-statistic rvalue < 0.05 at 95% confidence interval, we reject the Null hypothesis and there exists a a relation between amount of claims and number of claims. Here for F-statistic we have a value >>> 1, which is required to be large for smaller number of data points in our data and it staisfies that condition.

Discussions

As for the above analysis detailed so far, I am confident at an interval of 65% - 75%. This is because I feel couple of problems could have been addressed much better such as analysing the departure and arrival time relation. This could have been taken as a paired t-test to check alter. Furthermore this followed by, using factors and ordering them could have given us much more insights into the data, while I did use factors as default categorical variables, I didn’t explicitly mention and leveraged them.

The key takeaway I am confident is the analysis done, from developing the data dictionary for mapping the codes via the tidyverse and dplyr packages followed up with using joins to gain some insights.

Although the analysis is prety clear, however I had also tried others such as facet wrapping with Degree, Domain1 and Domain2, but that resulted in a very big grif, not readable.

As for an industrial or policy maker perspective, yes it can be used but it has some cons such as it might fail at outliers as some edge cases such as checking and ensuring for arrival time > departure time is not taken into consideration. Similarly other such edge cases can be incorporated to improve.