Data Programming with R - Autumn Trimester 2022/23

Divya Dwivedi: 22200315

2022-12-18

Project Background:

This project examines the employment projections (growth) in the USA for 2021-2031 analyzing the fastest growing and largest declining occupation trends as well as analyzing the degree supply and demand to fill in the growing number of jobs for that decade.

The file, called Employment-Projections.csv, is available in the US Bureau of Labor Statistics website (**https://data.bls.gov/projections/occupationProj**) .

This data represents the employment statistics in 2021 and the projection for the next 10 years (until 2031) based on multiple occupations. The data includes number of jobs in 2021, the median annual wage, projected number of jobs in 2031, typical entry level education, annual average occupation job openings and some other data which will be used for this analysis.

Problem Statement

This project will analyze the trend of occupations with maximum increase and decrease in job count in th United states. The objective of this study is to help young students choose their field of study as per the current market trend and its requirement.

```
#Loading the packages

library(ggplot2)
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(lattice)
```

```
#Loading the data set
df <- read.csv("Employment_Projections.csv")
```

Part 1: Analysis

```
#Printing Column names
colnames(df)
```

```
##  [1] "Occupation.Title"
##  [2] "Occupation.Code"
##  [3] "Employment.2021"
##  [4] "Employment.2031"
##  [5] "Employment.Change..2021.2031"
##  [6] "Employment.Percent.Change..2021.2031"
##  [7] "Occupational.Openings..2021.2031.Annual.Average"
##  [8] "Median.Annual.Wage.2021"
##  [9] "Typical.Entry.Level.Education"
## [10] "Education.Code"
## [11] "Work.Experience.in.a.Related.Occupation"
## [12] "Workex.Code"
## [13] "Typical.on.the.job.Training"
## [14] "trCode"
```

Cleaning the Data Set

From the column names and dataset we can observe multiple columns that are not going to help us with our analysis such as Occupation code, Education.Code, Workex.Code, trCode.

Hence, removing those columns from our dataset for further analysis.

```
#Dropping unneccessary column names using their index number
df <- df[-c(2,10,12,14)]
```

```
#Renaming column names for convenience our use

df<-df %>%
  rename(
    title = Occupation.Title,
    emp21 = Employment.2021,
    emp31 = Employment.2031,
    change = Employment.Change..2021.2031,
    percent= Employment.Percent.Change..2021.2031,
    opening=Occupational.Openings..2021.2031.Annual.Average,
    annual_wage=Median.Annual.Wage.2021,
    education=Typical.Entry.Level.Education,
    workex= Work.Experience.in.a.Related.Occupation,
    training= Typical.on.the.job.Training
    )
```

The data set contains 10 columns and 832 rows.

It contains of 4 categorical variables, and 6 numerical variables.

Defining each variable:

Occupation title: This contains the different occupations in the United States considered for the employment prediction growth.

Employment 2021: This column contains the employment count associated with each occupation in the year 2021.The National Employment Matrix measures total employment as a count of jobs, not a count of individual workers.

Employment 2031: This column contains the employment count prediction associated with each occupation in the year 2031

Employment Change 2021 2031: This contains the difference in the growth or reduction in the employment change over the decade

Employment percent change 2021 203:1 This contains the percentage change in the employment growth over the decade

Occupational Openings, 2021-2031 Annual Average: This contains the openings in each occupation on an average for the decade 2021 to 2031

Median Annual Wage 2021: This is the median annual wage for each occupation in the United states in 2021 in USD

Typical Entry Level Education This contains eight different levels of education ranging from Doctoral or Professional degree to no formal education giving an idea of the employment scope with respect to the education degree.

Work Experience in a relation Occupation: Work experience has been divided into three factors None, Less than 5 Years, 5 years or more

Typical On the job training: Training or preparation that is typically needed, once employed in an occupation, to attain competency in the skills needed in that occupation. Training is occupation-specific rather than job-specific; skills learned can be transferred to another job in the same occupation.

```
#  Shortening the names of occupations, more specifically in the "Occupation"
#  Column deleting all the strings which come after asterics and deleting
#  the unnecessary spaces resulted from that action. This step is needed to
#  display the occupation names in user-friendly way, especially when
#  displaying those occupations in figures.

df$title <- gsub("\\*.*", "", df$title)
```

Analyzing the Data Set

```
#Counting the no of rows and columns of the dataset
nrow(df)
```

```
## [1] 832
```

```
ncol(df)
```

```
## [1] 10
```

```
#Checking any missing values
names(which(colSums(is.na(df))>0))
```

```
## [1] "annual_wage"
```

```
#Handling Missing values

df[is.na(df)]<-0 #Replace na values with 0 using is.na()
sum(is.na(df)) #Checking for the final count of missing values
```

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

```r
#Numerical summary of the dataset
summary(df)
```

```
##     title               emp21             emp31            change        
##  Length:832         Min.   :   0.30   Min.   :   0.3   Min.   :-335.700  
##  Class :character   1st Qu.:  16.48   1st Qu.:  17.3   1st Qu.:   0.100  
##  Mode  :character   Median :  49.15   Median :  51.4   Median :   1.400  
##                     Mean   : 190.07   Mean   : 200.1   Mean   :   9.997  
##                     3rd Qu.: 156.57   3rd Qu.: 162.1   3rd Qu.:   7.700  
##                     Max.   :3855.20   Max.   :4560.9   Max.   : 924.000  
##     percent           opening          annual_wage      education        
##  Min.   :-38.200   Min.   :  0.000   Min.   :     0   Length:832        
##  1st Qu.:  0.500   1st Qu.:  1.675   1st Qu.: 37770   Class :character  
##  Median :  4.500   Median :  5.150   Median : 49120   Mode  :character  
##  Mean   :  4.377   Mean   : 23.474   Mean   : 61549                     
##  3rd Qu.:  8.300   3rd Qu.: 16.600   3rd Qu.: 77030                     
##  Max.   : 45.700   Max.   :741.400   Max.   :208000                     
##     workex            training        
##  Length:832         Length:832        
##  Class :character   Class :character  
##  Mode  :character   Mode  :character  
##                                       
##                                       
##                                       
```

Summary of the Data Set:

There were a total of 7 missing values in the Median annual Wage column which has been replaced by 0 for analyzing the data further.

We cannot remove the missing value rows as each row in this data set corresponds to a unique profession due to which we simply replace the missing values by 0.

As per the numerical summary data, we can see that the employment in 2021 and 2031 is at a minimum of 0.3 but is predicted to reach a maximum from 3855.20 to 4560.90 in a decade's time.

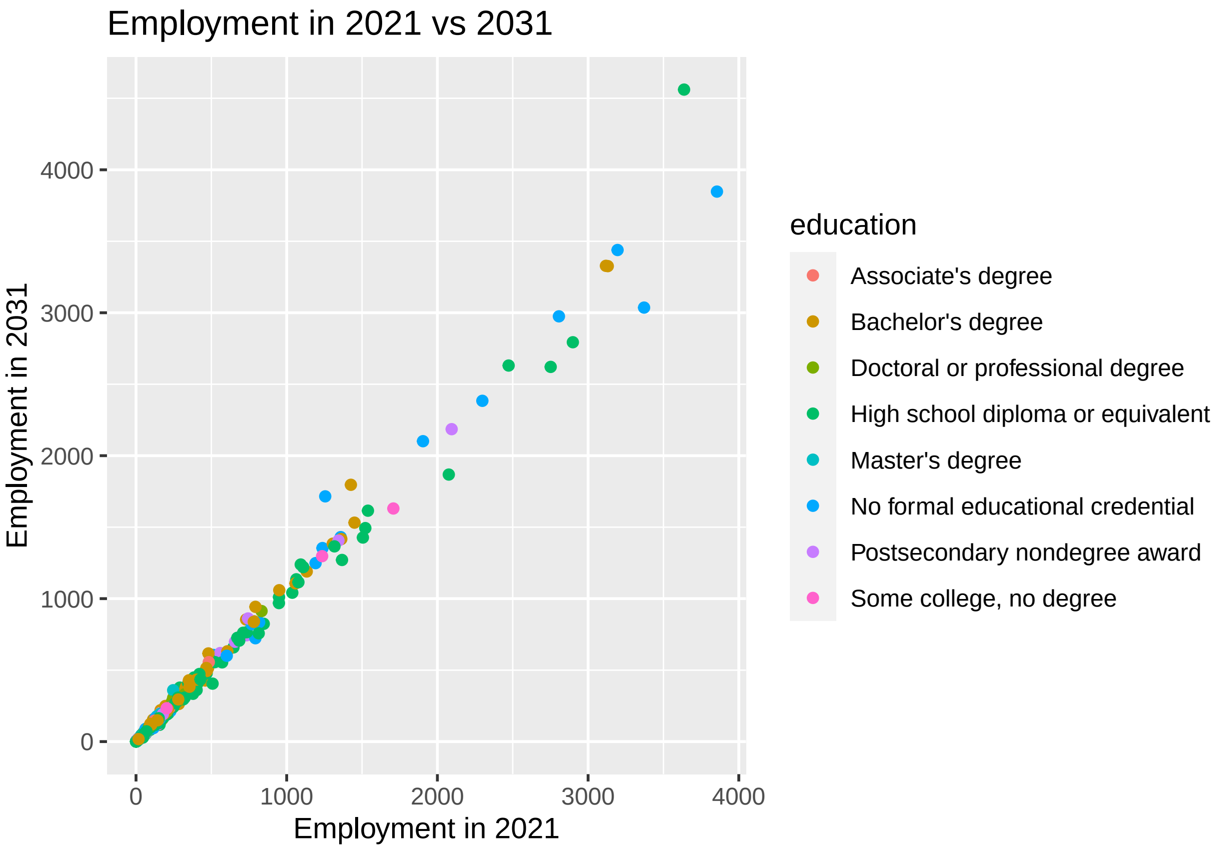
The employment percent change is in the range of -38.20 to 45.700 across all occupations which means that the employment percentage is predicted to increase at a maximum of 45.700. The average increase in employment is 9.997 units for the decade 2021-2031 with its median at 1.400.

The Occupational openings range from 0 to 741.400 with its median at 5.150 and mean at 23.474. The IQR for the same is 14.925.

The median annual wage for the year 2021 for all occupation lies in the range of 0 to 208000 with its mean at 61549 and median at 49120 hence we observe right skewness in the distribution of the of the data points for this entity.

Plot 1:

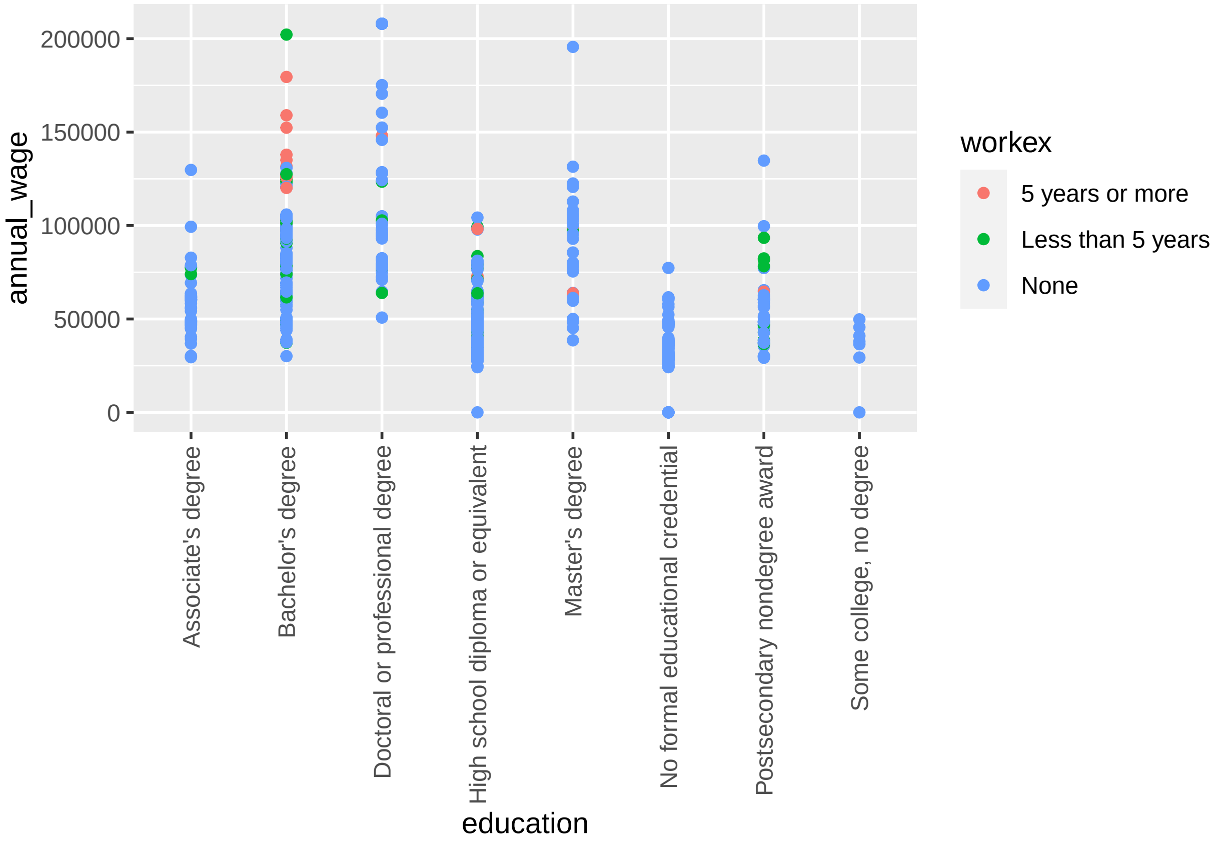
```
qplot(
  x = emp21,
  y = emp31,
  data = df,
  color = education,
  xlab = 'Employment in 2021',
  ylab = 'Employment in 2031',
  main = "Employment in 2021 vs 2031"
)
```



The above scatter plot indicates a strong positive correlation between the two variables: Employment in 2021 vs Employment in 2031. We can see that the correlation best line has an upward 45 degree tilt meaning that most occupations can be seen to have a gradual upward rise in the coming decade. The variance in the plot indicates that there is a significant change in the employment count of jobs in the year 2031 as compared to 2021. A few points are seen to be located below the best line with a negative change which indicates that a few occupations are predicted to have a decrease in the job count in the coming years.

Plot 2:

```
q<-qplot(
  x = education,
  y = annual_wage,
  data = df,
  color = workex
)
q + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```

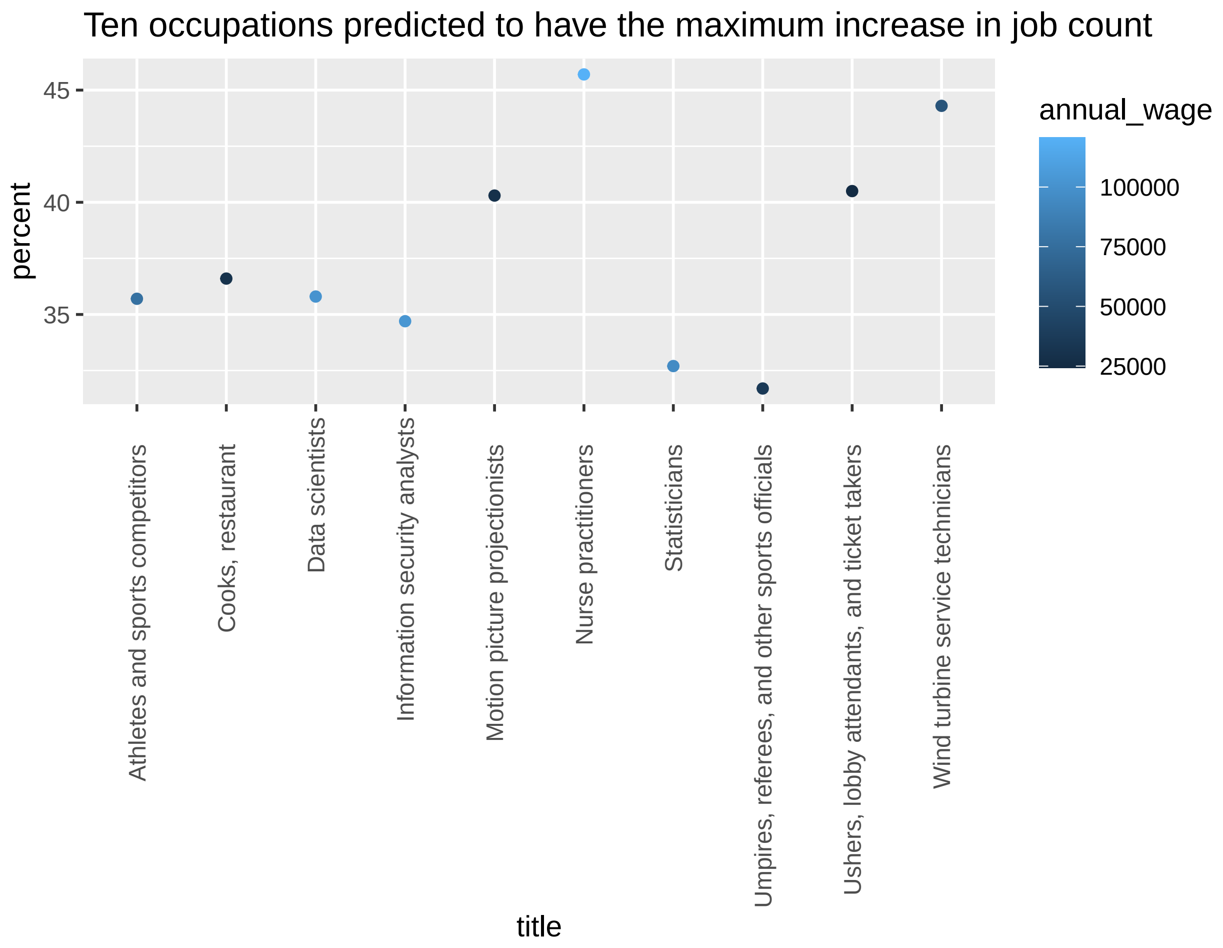


The plot above shows the Median Annual Wage in 2021 for each level of education segregated by their Work experience in their relevant field.It can be observed that the highest wage is for Doctoral or professional degree holders followed by those with a Bachelors and Masters degree in the year 2021. The lowest annual median wage is zero for individuals with high school diploma, no formal educational credential or no degree holders from some college. It can also be mentioned that Work experience does not portray any accurate or fixed trend of result from this graph as individuals with 5 or more years of experience have wage less than those who have none. Hence, there is no fixed trend observed for its correlation with the annual wage.

```
df1<-df %>%
    arrange(desc(df$percent)) %>%
    slice(1:10)
```

Plot 3:

```
q1<-qplot(
  y = percent,
  x = title,
  data = df1,
  color = annual_wage,
  main = 'Ten occupations predicted to have the maximum increase in job count'
)
q1 + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



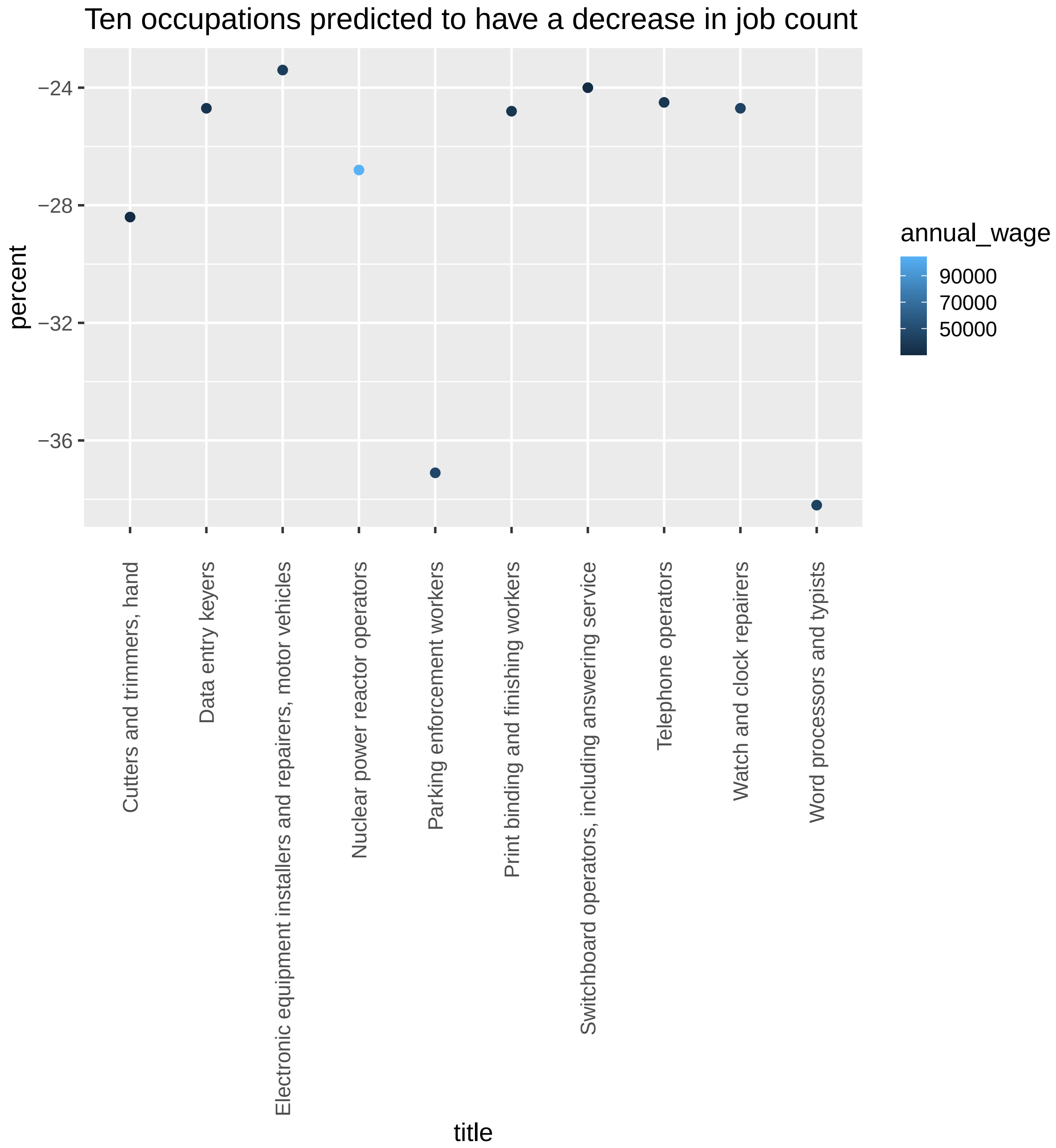
From the plot above, we can see the top ten occupations which have predicted the maximum growth in employment percent change for the decade 2021-2031. The scatter plot is plotted as per the median annual wage scale and it can be seen that the Occupation 29-1171 (Nurse Practitioner) has the maximum Percent change i.e 45.7 with median annual wage 2021 of 120680 USD. This is followed by 49-9081 (Wind turbine service technicians) with a percent change of 44.3 and annual median wage of 56260 USD.

This plot gives us an idea of the field of study that is predicted to increase in growth in the coming decade as per the dataset obtained by the US Bureau of Labor Statistics website.

```
df2<-df %>%
    arrange(df$percent) %>%
    slice(1:10)
```

Plot 4:

```
q2<-qplot(
  y = percent,
  x = title,
  data = df2,
  color = annual_wage,
  main = 'Ten occupations predicted to have a decrease in job count',
)

q2 + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1),
          legend.key.size = unit(0.3, "cm"), legend.key.width = unit(0.4,"cm"))
```



From the plot above, we can see the least ten occupations which have predicted the most negative growth in employment percent change for the decade 2021-2031. The scatter plot is plotted as per the median annual wage scale and it can be seen that the Occupation 43-9022 (Word processors and typists) has the most negative Percent change i.e -38.2 with median annual wage 2021 of 44030 USD. This is followed by 33-3041 (Parking enforcement workers) with a percent decrease in job count of 37.1 and annual median wage of 46590 USD.

This plot gives us an idea of the occupations that are predicted to show most decline in the job counts in the coming decade as per the dataset obtained by the US Bureau of Labor Statistics website.

```
#Checking the sum of occupations which have no increase in employment for the decade
no_change <-  sum(df$percent == 0)
print(no_change)
```

```
## [1] 4
```

```
#Checking the sum of occupations which have an increase in employment for the decade
increase <- sum(df$percent > 0)
print(increase)
```

```
## [1] 635
```

```
#Checking the sum of occupations which have a decrease in employment for the decade
decrease <- sum(df$percent < 0)
print(decrease)
```

```
## [1] 193
```

```
inc <- sum(df$change[which(df$change>0)]*1000)
print(inc)
```

```
## [1] 10476800
```

```
dec <- sum(df$change[which(df$change<0)]*1000)
print(dec)
```

```
## [1] -2159200
```

We can see that out of 832 occupations listed in the data set, 4 occupations are predicted to have no change in employment, 635 have an increase in job employment count and 193 are predicted to have a decrease in job count.

Part 2: R Package

Package selected: Caret

```
library(caret)
```

A caret package is a short form of Classification And Regression Training used for predictive modeling where it provides the tools for the following process.

- Pre-Processing: Where data is pre-processed and also the missing data is checked.preprocess() is provided by caret for doing such task.
- Data splitting: Splitting the training data into two similar categorical data sets is done. Feature selection: Techniques which is most suitable like Recursive Feature selection can be used.
- Training Model: caret provides many packages for machine learning algorithms.
- Resampling for model tuning: The model can be tuned using repeated k-fold, k-fold, etc. Also, the parameter can be tuned using 'tuneLength.'
- Variable importance estimation: vlamp() can be used for any model to access the variable importance estimation.

You can install the caret package by the following command.

```
    __install.packages('caret')__
```

Creating a simple model

We're gonna do that by using the train() function. The function train() is a core function of caret. As its name suggests, it is used to train a model, that is, to apply an algorithm to a set of data and create a model which represents that dataset.

The train() function has three basic parameters:

- Formula : where you specify what is your dependent variable (what you want to predict) and independent variables (features).
- Dataset : is the data.
- Method (or algorithm) : is a string specifying which classification or regression model to use.

```
#Copying the dataset to a new dataset named data
#Dropping title, change and percent variables as they are not needed for
#the linear regression model

data<- df[-c(1,4,5)]

#Converting categorical columns into factors

data$education<- as.factor(data$education)
data$workex<- as.factor(data$workex)
data$training<- as.factor(data$training)
```

```
# Building multiple linear regression model
#Taking emp31 as the response variable and using method as lm
#Writing emp31~. to consider all attributes as predictor variables

model <- train(emp31~.,
               data = data,
               method = "lm" )
```

K-fold cross-validation

The function train() has other optional parameters. Adding re-sampling to our model by adding the parameter trControl (train control) to our train() function.

The re-sampling process can be done by using K-fold cross-validation, leave-one-out cross-validation or bootstrapping. We are going to use 10-fold cross-validation in this example. To achieve that, we need to use another Caret function, trainControl(). Check the code below.

```
## 10-fold CV
# possible values: boot", "boot632", "cv", "repeatedcv", "LOOCV", "LGOCV"

fitControl <- trainControl(method = "repeatedcv",
                           number = 10,    #number of folds
                           repeats = 10)   #repeated ten times
```

Adding preprocessing

The train() function has another optional parameter called **preProcess**. It's used to add some pre-processing to your data.

In this example we're going to use the following pre-processing:

- center data (i.e. compute the mean for each column and subtracts it from each respective value);
- scale data (i.e. put all data on the same scale, e.g. a scale from 0 up to 1)

However, there are more pre-processing possibilities such as "BoxCox", "YeoJohnson", "expoTrans", "range", "knnImpute", "bagImpute", "medianImpute", "pca", "ica" and "spatialSign".

```
model.cv <- train(emp31 ~ .,
               data = data,
               method = "lasso",  # now we're using the lasso method
               trControl = fitControl,
               preProcess = c('scale', 'center'))

model.cv
```

```
## The lasso
##
## 832 samples
##   6 predictor
##
## Pre-processing: scaled (17), centered (17)
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 750, 748, 750, 748, 750, 749, ...
## Resampling results across tuning parameters:
##
##   fraction  RMSE       Rsquared   MAE
##   0.1       391.08686  0.9936635  209.81121
##   0.5       208.61900  0.9936635  112.09661
##   0.9        46.47342  0.9937683   20.83802
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.9.
```

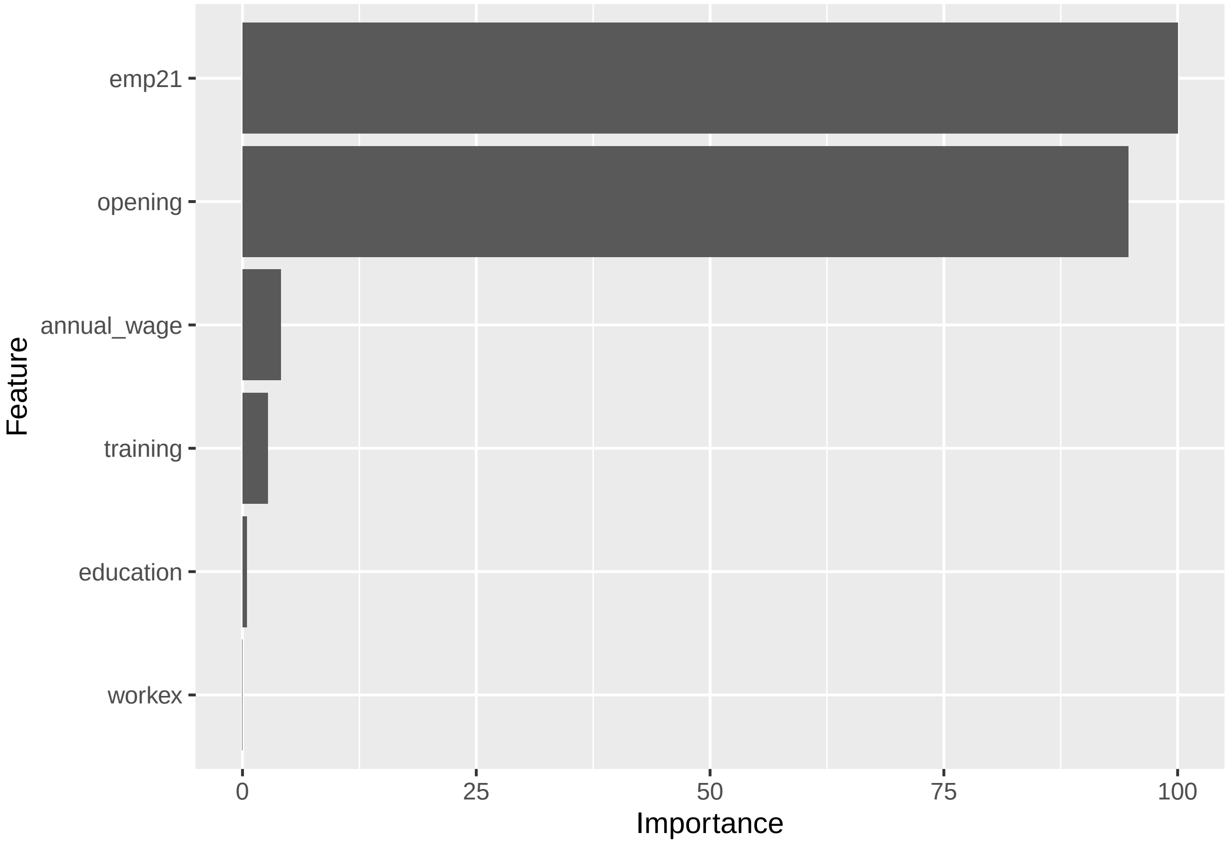
Root Mean Square Error (RMSE) is the standard deviation of the residuals (prediction errors). Residuals are a measure of how far from the regression line data points are. We select the RMSE with the smallest value giving an R-squared of 0.99. This indicates how well the regression model explains observed data. In this case, 99% of the variability observed in the target variable is explained by the regression model.

Variable Importance

We can use the Caret function varImp to estimate the importance of each predictor variable from the linear model. The return of varImp can be passed to the function ggplot to generate a visualization.

Plot 5

```
ggplot(varImp(model.cv))
```



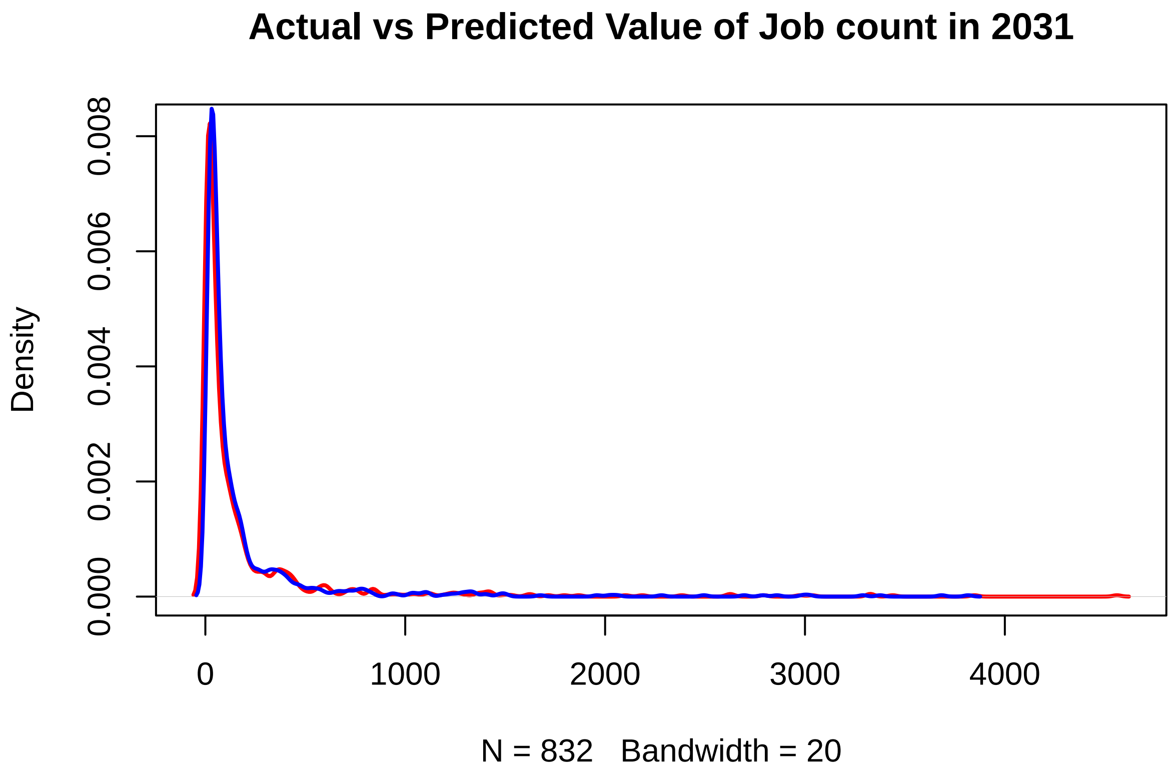
From the linear model we can see that amongst all predictor variables, emp21 variable has the most infuence and is the most important predictor variable in the model followed by opening, annual_wage, training, education and workex.

```
predictions <- predict(model.cv, data)
head(predictions)
```

```
##          1          2          3          4          5          6 
## 1411.22511   64.19680   41.59240   37.04101   26.86176   28.07252 
```

Plot 6

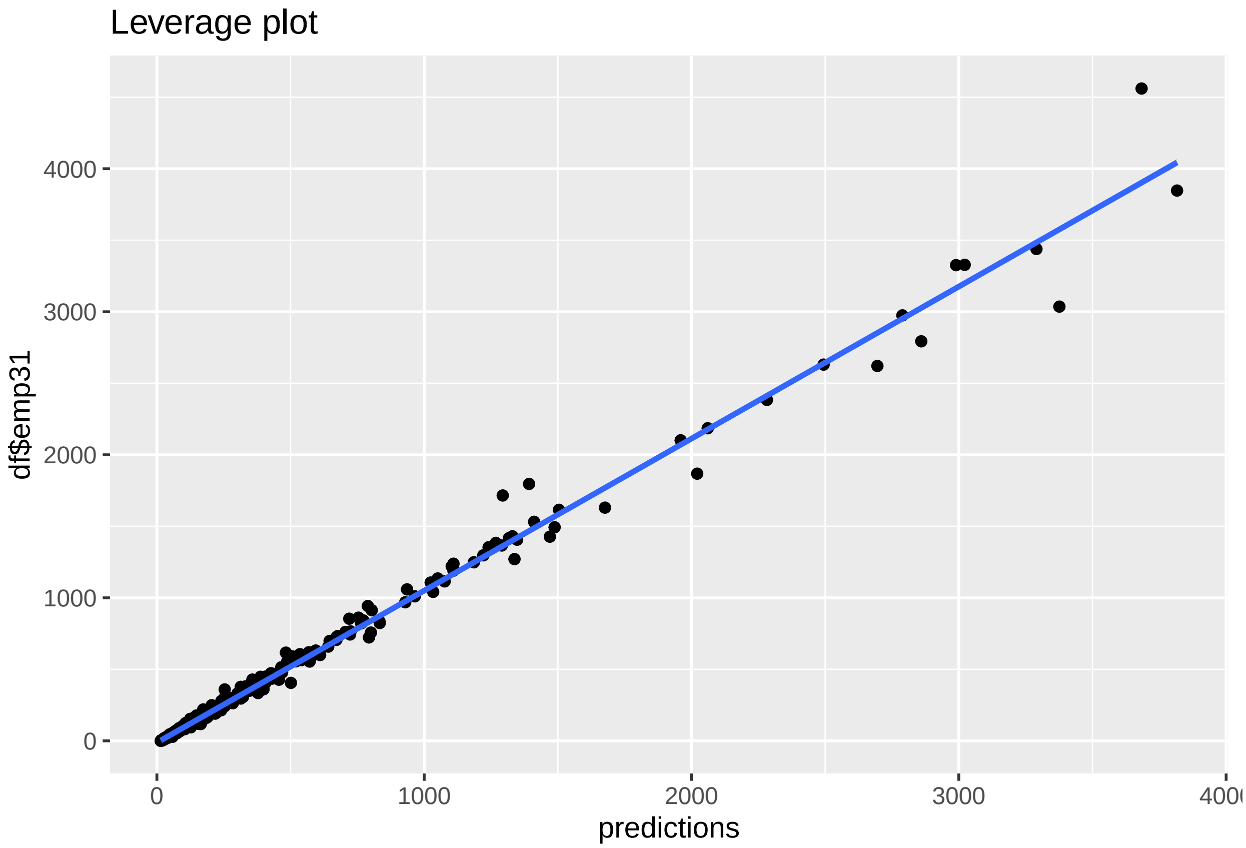
```
#Density Plot

plot(density(df$emp31, bw = 20), lwd = 2,
     col = "red", main = "Actual vs Predicted Value of Job count in 2031")
lines(density(predictions, bw = 20), lwd = 2,
     col = "blue",main="")
```



Plot 7

```
#Leverage plot
ggplot(data,
       aes(x = predictions,
           y = df$emp31)) +
  geom_point() + geom_smooth(method = "lm", se = TRUE)+ labs(title = "Leverage plot")
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



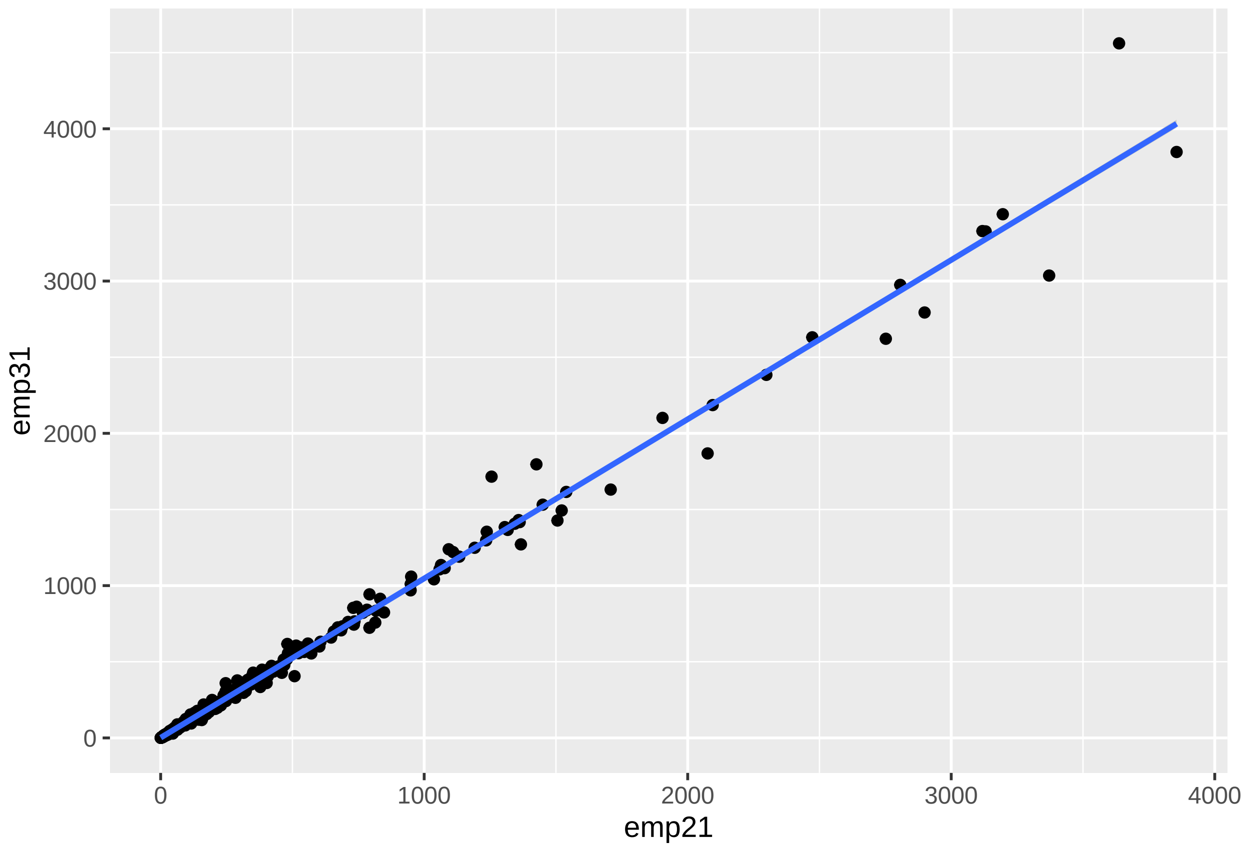
The above two Actual vs Predicted value plots depict the model accuracy. The data points are closer to the best fit line in the Leverage plot stating that the error margin and variance is extremely low and our estimate of employment in 2031 is 99% accurate as confirmed by the R squared value.

We do observe a few outlier’s in the model consisting of a few leverage and a few influential points affecting the data’s explanatory summary.

Plot 8

```
#Plotting correlation graph between the most significant predictor variable
# and response variable to see the best fit line and its variation
ggplot(data, aes(emp21, emp31), conf.int = TRUE,cor.coef = TRUE)+
  geom_point() + geom_smooth(method = 'lm')
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



Setup the resampling approach

Here we will use 10-fold cross validation 5 times (repeated CV).

```
# Set seed for reproducibility
set.seed(100)

# Set up the resampling, here repeated CV
tr <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
```

Do the modeling

Stepwise regression with caret Here we run automated stepwise regression and look at the resampling results. Since there is no tuning parameter so the final results table has just one row.

```
# Note that trace is a parameter sent to the underlying modeling function
step_model <- train(emp31~.,data=data,
                    method = "lmStepAIC", trControl = tr, trace = FALSE)
step_model$results
```

```
##   parameter     RMSE  Rsquared     MAE   RMSESD  RsquaredSD    MAESD
## 1      none 40.30757 0.9941156 14.9443 25.22995 0.004343309 5.088536
```

```
step_model$finalModel
```

```
##
## Call:
## lm(formula = .outcome ~ emp21 + opening + annual_wage + ‘educationBachelor’s degree‘ +
##     ‘trainingShort-term on-the-job training‘, data = dat)
##
## Coefficients:
##                              (Intercept)
##                               -1.376e+00
##                                    emp21
##                                9.418e-01
##                                  opening
##                                7.540e-01
##                              annual_wage
##                                6.592e-05
##             ‘educationBachelor’s degree‘
##                                1.144e+01
## ‘trainingShort-term on-the-job training‘
##                               -8.519e+00
```

From the Resampling approach we get the best model summary as listed above with RMSE 40.3 and Rsquared as 0.99.

Even modeling using linear regression, caret can improve your workflow by simplifying data splitting, automating your resampling and providing a vehicle for comparing models. If you need to compare different model types and particularly if you run models with tuning parameters caret will save you an incredible amount of time by automating resampling on different settings of your tuning parameters and allowing you to use a consistent syntax across hundreds of different model types.

Part 3: Functions/Programming

Values are not the only place to store information in R, and functions are not the only way to create unique behavior. You can also do both of these things with R's S3 system. The S3 system provides a simple way to create object-specific behavior in R. In other words, it is R's version of object-oriented programming (OOP). The system is implemented by generic functions. These functions examine the class attribute of their input and call a class-specific method to generate output. Many S3 methods will look for and use additional information that is stored in an object's attributes. Many common R functions are S3 generics.

Summary function

```
#Creating a list of first ten rows to analyze the data

data2 = list(emp21= df$emp21[1:10], emp31= df$emp31[1:10],
             change=df$change[1:10], open=df$opening[1:10],
             annual_wage=df$annual_wage[1:10])

#Creating a S3 class named emp
class(data2)='emp'

#Creating a summary correlation function

summary.emp = function(x) {
  v<-cor(df$emp21, df$emp31)
  return(v)
}

summary(data2)
```

```
## [1] 0.9952181
```

We can see that the correlation between the two entities (emp21 and emp31) is strong and positive and at 0.99

```
#create a  function that returns the numerical summary

summary.emp <- function(x) {
    cat('min' = min(x),'max' = max(x), 'mean'=mean(x), 'median'= median(x))
}

#Using sapply to obtain a matrix of values using the list and the functiom
sapply(data2, summary)
```

```
##             emp21   emp31 change    open annual_wage
## Min.       12.700   12.90  -8.00   0.600           0
## 1st Qu.    24.775   25.95  -0.25   1.875       54185
## Median     36.600   38.15   1.45   3.850       68910
## Mean      199.220  209.20   9.98  19.030       72328
## 3rd Qu.    88.175   83.20   5.45   9.925      101955
## Max.     1449.800 1531.60  81.80 136.400      127150
```

The numstat function gives us an idea of the numerical summary of an entity. For example, calling the opening variable from the dataset we see that it ranges from 0 to 741.400 with its mean at 23.4744 and median at 5.1500.

Print function

```
#Creating a print function to print job count in 2021, 2031 and the change
#in the decade

print.emp <- function(wkr) {
cat('Jobs in 2021',wkr$emp21, '\n')
cat('Jobs in 2031', wkr$emp31, '\n')
cat('change', wkr$change, '\n')
}

print(data2)
```

```
## Jobs in 2021 1449.8 50.6 28.3 23.7 12.7 14.5 239 44.9 28 100.7
## Jobs in 2031 1531.6 54.7 34.2 24.6 12.9 14.1 255.1 42.1 30 92.7
## change 81.8 4.1 5.9 0.9 0.2 -0.4 16.1 -2.8 2 -8
```

```
# Create a linear model using S3 Class

mod <- lm(emp31 ~ emp21, data = data)
class(mod)
```

```
## [1] "lm"
```

```
print(mod)
```

```
##
## Call:
## lm(formula = emp31 ~ emp21, data = data)
##
## Coefficients:
## (Intercept)        emp21
##       1.310        1.046
```

```
class(mod) <- "data.frame"
mod$coefficients
```

```
## (Intercept)       emp21
##    1.310230    1.045706
```

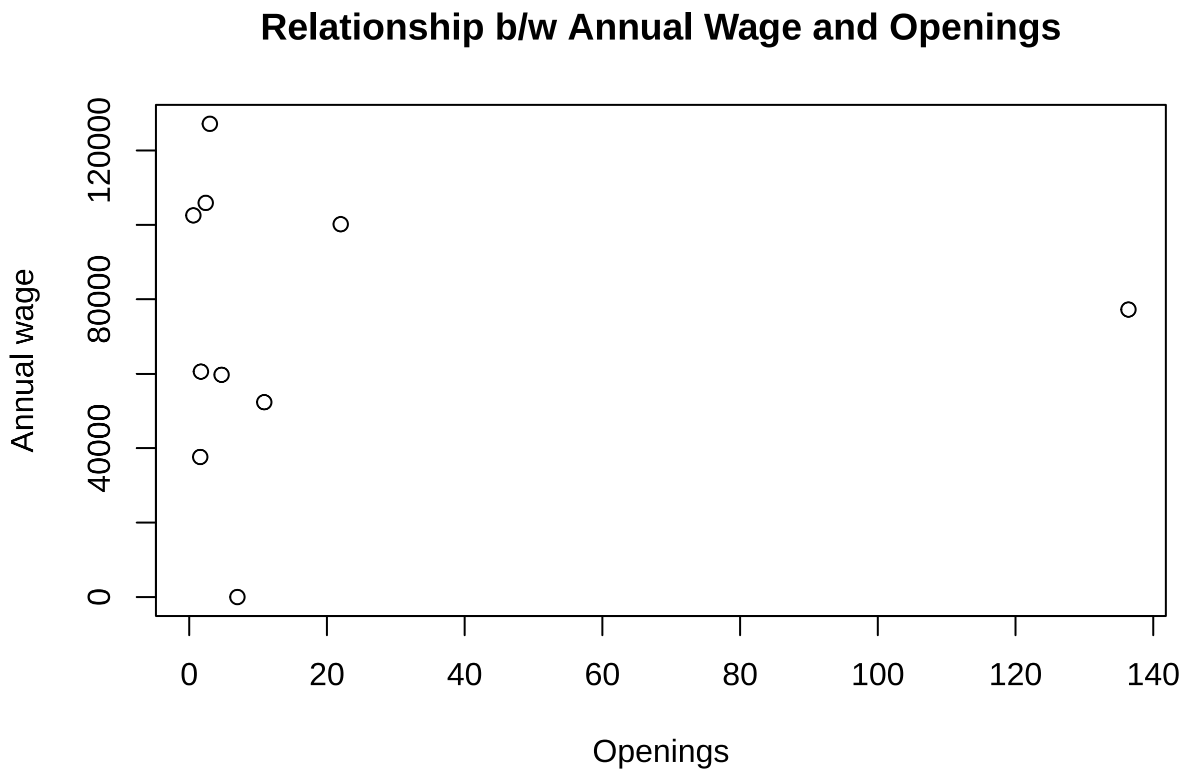
This linear model gives us an the linear coefficients of the model with only emp21 and emp31 as its variables.

The final linear equation is:

(emp31= 1.310 + 1.046 * emp21)

Plot function

```
#Creating a Plot method

plot.emp <- function(p) {

plot(p$open, p$annual_wage, xlab = "Openings", ylab = "Annual wage",
main = "Relationship b/w Annual Wage and Openings")

}

plot(data2)
```



From the plot function graph above we can see there is no correlation between the no of job openings and Annual Wage of an Occupation and hence we cannot make an estimate or conclusion from the same. Similarly, we can plot graphs for all other entities we wish to see the relation for.

Conclusion

The project reveals that the number of job growth from 2021-2031 will be 8.37 million which is the difference between the number of jobs created (10.47 million) and the number of jobs reduced (2.1 million). The growth will be 7.7 percent. From 823 occupations 4 will have no change in job quantity, 635 occupations will face job increase and 193 job decrease.

From the top ten occupations which will face largest increase in job quantity (in absolute numbers) the home health and personal care aides will have around than 1.1 million job increase. From the ten largest declining occupation (cashiers, tellers, bookkeepers, secretaries) list it is possible to assume that the major factor in that decline is automation.

When we look into the percentage numbers, the top and bottom ten occupations are totally different because the total numbers of those occupations are less. No single occupation has double or multiple time increase in job quantity: the maximum share of increase is 45.7% for Nurse practitioner. The largest decline is 36% which word processors and typists will face.

Another interesting result is the number of jobs created based on wage quintiles. It is interesting to note that almost half of newly created jobs (more specifically 5.3 million jobs or 45% of all newly created jobs) will be created in the first quantile where the salary threshold is a little bit more than $35,000.

The project also analyzes the relationship between the educational attainment and employment. Based on that analysis the result shows that the largest demand for the new jobs will be for the Bachelor's degree requiring 3.6 million degrees to fill the new jobs. But in percentage terms the highest demand per individual occupation job increase will be for Master's degree: 13.8% on average or 10.7 based on median calculation, and the lowest decline will be for Master's degree comprising 1.7%.

As a conclusion it is possible to assume that to cover the Bachelor's degree gap the employers will higher the people with Master's degrees (which has a surplus) increasing the underemployment rate in the US or they will reduce their requirement of educational attainment level to Associate's degree and hire the excess amount of Associate degree holders for those job openings reducing the level of quality of those jobs. Another assumption for this result is the possible trend that Associate's and Master's degree holders (who will be out of job openings) will be better of proceeding to the next level of educational attainment to fill the gap in the job openings requiring Bachelor's and Doctoral or professional degrees.

Bibliography

[1] @Book{, author = {Hadley Wickham}, title = {ggplot2: Elegant Graphics for Data Analysis}, publisher = {Springer-Verlag New York}, year = {2016}, isbn = {978-3-319-24277-4}, url = {https://ggplot2.tidyverse.org}, }

[2] @Manual{, title = {caret: Classification and Regression Training}, author = {Max Kuhn}, year = {2022}, note = {R package version 6.0-93}, url = {https://CRAN.R-project.org/package=caret}, }

[3] @Manual{, title = {dplyr: A Grammar of Data Manipulation}, author = {Hadley Wickham and Romain François and Lionel Henry and Kirill Müller}, year = {2022}, note = {R package version 1.0.10}, url = {https://CRAN.R-project.org/package=dplyr}, }