2501977172 – Andi Izzat Zaky Ashari
Data Mining & Visalization [LAB Assignment]

Predictive Model – Random Forest

I. Introduction

This dataset (Telco Customer Churn) contain several variables including:

$ **customerID** : chr -> Indicating customer's ID
$ **gender** : chr -> Customer's gender
$ **SeniorCitizen** : int -> Wheter the customer is a new citizen or not in the area

$ Partner : chr -> Whether the customer has a partner or not
$ Dependents : chr -> Whether the customer has a dependents or not
$ **tenure** : int -> For how long they've become the customer (in month)

...

$ PaymentMethod : chr -> Indicate the customer's payment methods
$ **MonthlyCharges** : num -> Indicate the customer's current monthly charge for their plan
$ **TotalCharges** : num -> Indicate the customer's total charges, calculated in the end of every quarter
$ **Churn** : chr -> Wheter the customer still active or not

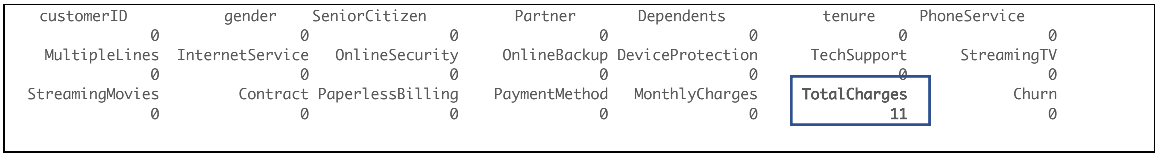
*The **Bold** variable is the variable that I will use for further exploration

II. Eploratory Data Analysis

Here's the data summary

```
── Data Summary ────────────────────────
                           Values
Number of rows             7043
Number of columns          21
________________________________________
Column type frequency:
character                17
numeric                  4
________________________________________
Group variables            None
________________________________________
```

There are 7043 data inside this dataset, and for the column type, there are 17 variables indicated as charcter variables, and 4 as numeric variables. And when running, colSums(is.na()) function, I noticed that there are one variable with 11 missing values



```
   customerID           gender    SeniorCitizen          Partner       Dependents           tenure     PhoneService
            0                0                0                0                0                0                0
  MultipleLines  InternetService   OnlineSecurity     OnlineBackup DeviceProtection      TechSupport      StreamingTV
            0                0                0                0                0                0                0
StreamingMovies         Contract PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges            Churn
            0                0                0                0                0               11                0
```

Its time to clean the dataset!

I assign a new dataframe named "customer_clean" for the cleaned dataset using this command:

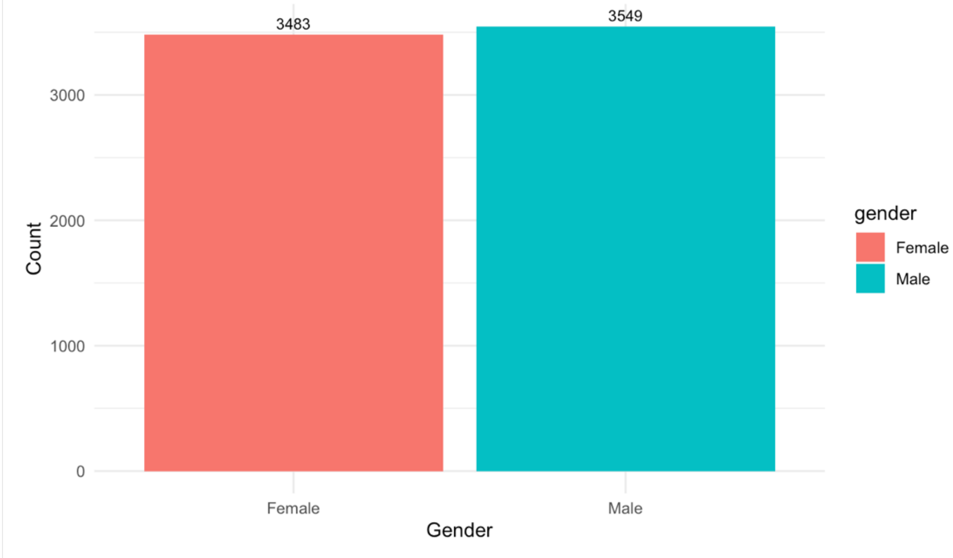
```
customer_clean <- na.omit(customer)
colSums(is.na(customer_clean))
```

And we got the result a cleaned dataset!

III. Data Visualization

I want to see the gender distribution for the customers, hence im using this command below:

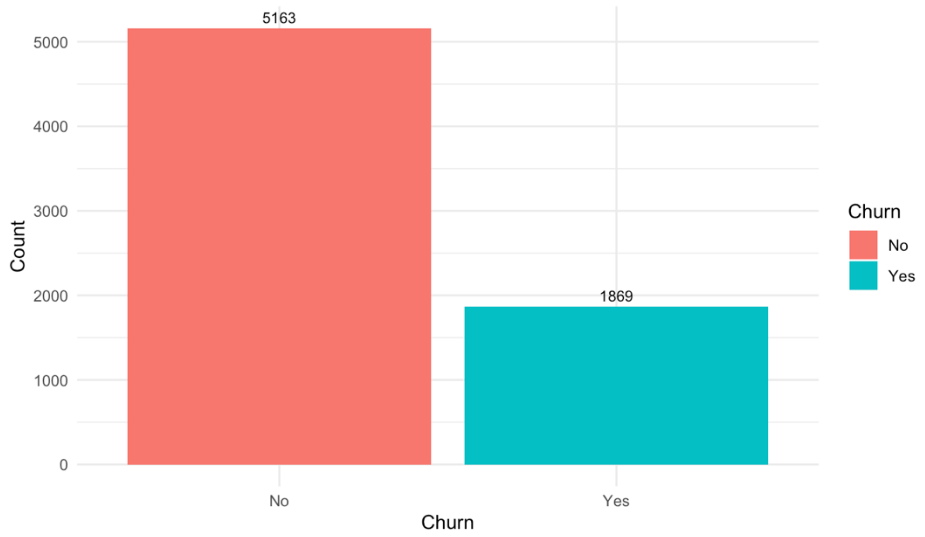
```
gender_counts <- customer_clean %>%
  group_by(gender) %>%
  summarize(count = n()) %>%
  mutate(percentage = count/sum(count) * 100)

gender_plot <- ggplot(gender_counts, aes(x = gender, y = count, fill = gender)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = count), vjust = -0.5, color = "black", size = 3) +
  labs(x = "Gender", y = "Count", title = "Gender Distribution") +
  theme_minimal()
```



From this plot, now we can see that there are 3,549 Male customers and 3,483 female customers.

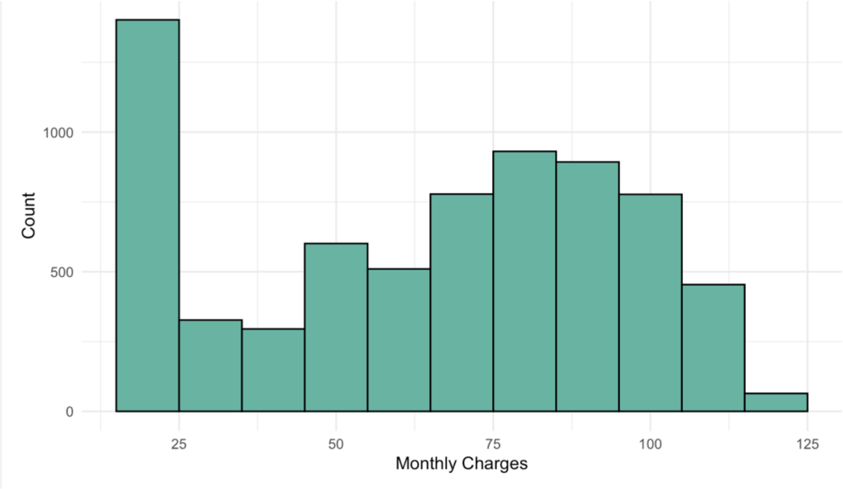
Now I wonder how my churn customers (customer who unsubscribe their plan), lets see the visualization for churn customers usingn this command below:

```
churn_counts <- customer_clean %>%
  group_by(Churn) %>%
  summarize(count = n()) %>%
  mutate(percentage = count/sum(count) * 100)

churn_plot <- ggplot(churn_counts, aes(x = Churn, y = count, fill = Churn)) +
  geom_bar(stat = "identity") +
  geom_text(aes(label = count), vjust = -0.5, color = "black", size = 3) +
  labs(x = "Churn", y = "Count", title = "Churn Distribution") +
  theme_minimal()
```



Good for the company that from 7,032 customers, the number of customer who unsubscribe their plan is just 1,869, which is the percentage of difference of 'churn' and 'no-churn' is 63.8%
I want to see monthly charge distribution fir the customers using this command:

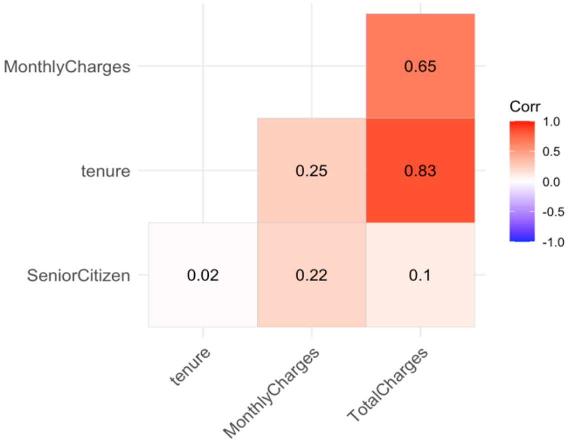
```
monthly_charges_plot <- ggplot(customer_clean, aes(x = MonthlyCharges)) +
  geom_histogram(binwidth = 10, fill = "#69b3a2", color = "black") +
  labs(x = "Monthly Charges", y = "Count", title = "Monthly Charges Distribution") +
  theme_minimal()
```



As we can see here, there are ~1500 customers who get charges 10-15USD monthly, and there are only small group of customers that have to pay 120-125USD monthly.

Now I want to see the correlations for several variables using Heatmap using this command:

```
correlation_matrix <- cor(customer_clean[, c("SeniorCitizen", "tenure", "MonthlyCharges",     "TotalCharges")])
correlation_heatmap <- ggcorrplot(correlation_matrix, type = "lower", lab = TRUE)
```



Based on the Heatmap graph, we can see that the correlation between 'tenure' and 'Total Charge' has almost perfect positive correlation. It means that customers with longer tenure tend to have a higher Total Charge.

And the 'Senior Citizen' and 'Tenure' has almost-zero correlation, it means that if the citizen is a 'senior citizen' it doesn't mean that they are a 'senior customer' too.

IV. PREDICTIVE MODELLING – RANDOM FOREST

I've finished the exploratory data, now lets doing the predictive modelling to predict the churn for customers.

First, lets convert the churn variable into factor:

```
customer_clean$Churn <- as.factor(customer_clean$Churn)
```

By converting it to a factor, we can later use it as the target variable in a predictive model for customer churn. Factors provide a convenient way to represent and handle categorical data in R, allowing the model to understand and make predictions based on the different categories of the "Churn" variable.

Now we continue to data partitioning process to split the train and test set:

```
set.seed(123)
train_index <- createDataPartition(customer_clean$Churn, p = 0.7, list = FALSE)
train_data <- customer_clean[train_index, ]
test_data <- customer_clean[-train_index, ]
```

By setting the seed to '123' I want to make sure the random process wil produce the same result everytime the code is executed. And the I assign a train_index variable which contain an index from churn customer data that later will be splitted into train data and test data, the ratio is 70/30, which mean there will be 70% for training set and 30% for test set.

After splitting data, now assigning the predictor and target variable. This separation between predictors and the target variable allows us to clearly define the problem and train the model accordingly. It'll helps the model learn the relationship between the predictors and the target variable, enabling it to make accurate predictions on unseen data.

Creating the model:

```
model <- randomForest(formula = Churn ~ ., data = train_data[, c(predictors, target)])
```

I assign a 'model' variable and using randomForest function to set the values inside the this variable, in this case, I want to predict the Churn variable. And assign the prredictors and target to the data train.

Assign the prediction variable:

```
predictions <- predict(model, newdata = test_data[, predictors])
```

we use predictions variable to store the test data that contain predictors variable.
Now lets evaluate the performance using confussion matrix and see the accuracy value.

```
confusion_matrix <- table(predictions, test_data$Churn)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
recall <- confusion_matrix["Yes", "Yes"] / sum(confusion_matrix["Yes", ])
```

I assign a confusion matrix to see the confussion matrix table, we can also get the value of the accuracy from using values from confusion matrx as a couple varibles to calculate the accuracy. I also assign a recall to measure how this model will correctly predicted the test set based on their actual value

```
print(confusion_matrix)
print(paste0("Accuracy: ", round(accuracy * 100, 2), "%"))
```

	Predicted Labels		
Actual Label		NO	YES
	NO	1393	256
	YES	155	304

```
"Accuracy: 80.5%"
"Recall: 66.23%"
```

From the result we got;

- True Negative

  Represent values that are correctly predicted as "NO" for the predicted and actual labels. So the model is correctly predicted that from the test data, there are 1393 customers that 'No-churn'.

- False Positive:

  Represent values that are incorrectly predicted as 'YES' when the actual label is 'NO', The model predicted there are 256 customerrs that churn(unsubscribe), but the actual case is No-churn.

- False Negative:

  Represents values that are incorrectly predicted as "NO" churn when the actual label is "YES" churn, the model incorrectly predicted 155 customers as "NO" churn when the actual label was "YES" churn.

- True Posivie:

  Represents values that are correctly predicted as "YES" churn when the actual label is also "YES" churn. In the given result, the model correctly predicted 304 instances as "Yes" churn.

After receiving the confusion matrix table, I already get the Accurracy of 80.5%, it means this model can effectively predict the data with accuracy of corredted prediction is 80.5%.

And for recall I got 66.23% which means that the model correctly identified 66% of the churned customers in the test set. In other words, out of all the customers who actually churned, the model correctly predicted 66.23% of them.

2501977172 – Andi Izzat Zaky Ashari
Data Mining & Visalization [LAB Assignment]

EDA – Tableu Visualization

I. Data Understanding

We've given 3 dataset which include:

- US Covid-19 – Weather dataset
- US SocioHealth dataset
- US Geometry dataset

All those 3 dataset has a bunch of variables in it.
The target of this EDA & Visualization is to check does covid-19 really effect the citizen's Mental healthy and their economy.

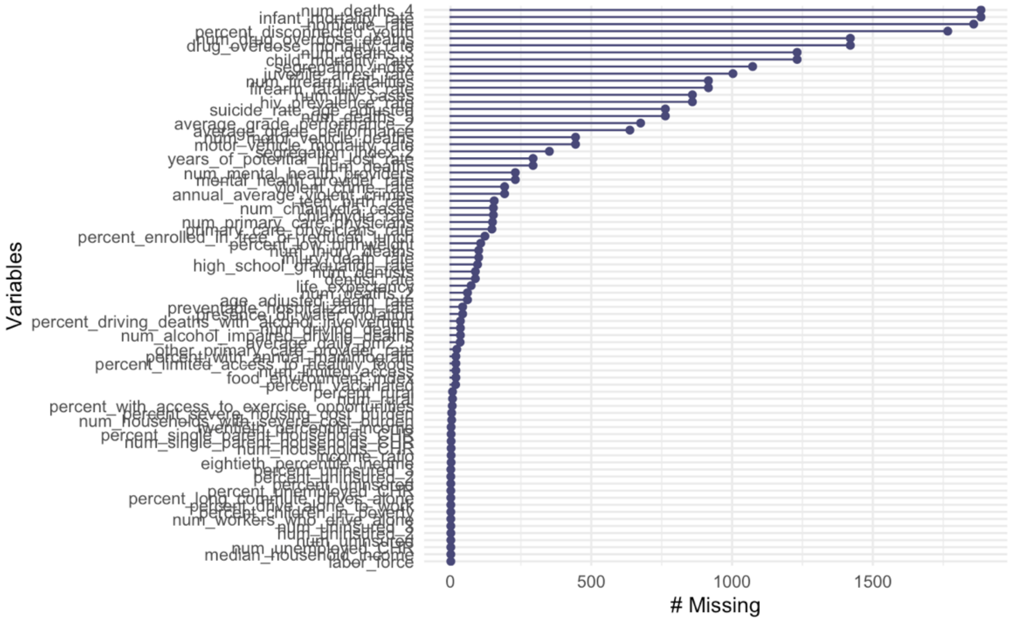
Checking the Missing Value for each dataset:
For checking missing values for all dataset, im using visualization to see all missing values, because all datasets has so many variables. Not that for visualize the missing value, im using Rstudio, not Tableu, I will use Tableu to visualize the report result.

- US SocioHealth
  Let's see how much the missing values!

```
sum(is.na(SocioHealth))

> 25871
```

There are 25,871 missing values in this dataset! Let's see the visualization

```
  # Identify columns with missing values
  cols_with_missing <- colnames(SocioHealth)[apply(SocioHealth, 2, anyNA)]

  # Filter the data frame to include only columns with missing values
  SocioHealth_filtered <- SocioHealth[ , cols_with_missing]

  # Visualize missing values
  gg_miss_var(SocioHealth_filtered)
```



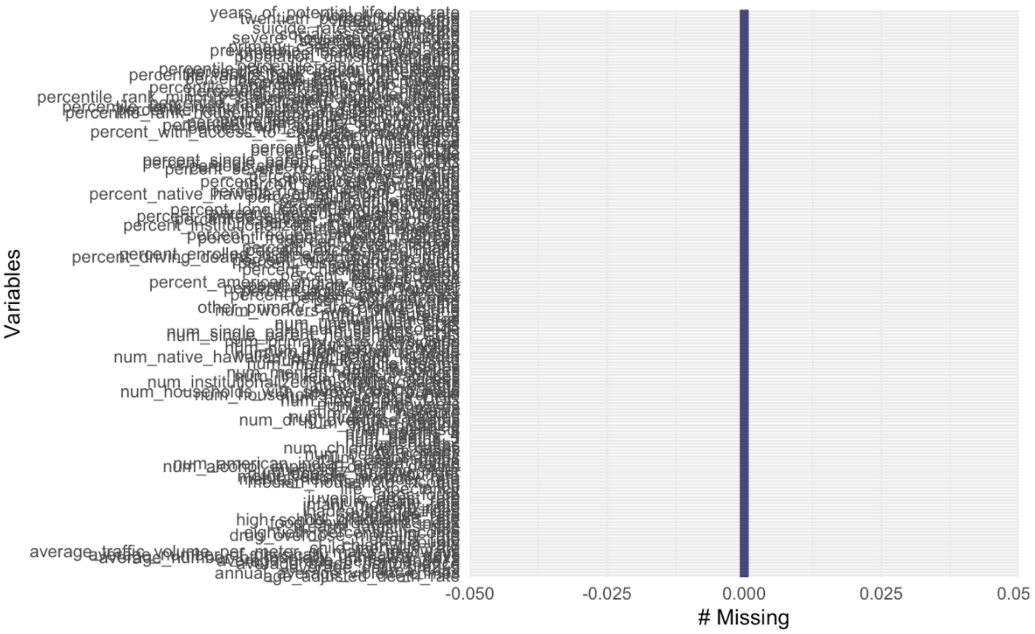
Even we can't see the exact variables that contain missing values, from the visualization, we can see that there's a lot of variables that has a missing values.

Now lets clean it!

```
SocioHealth <- na.omit(SocioHealth)
```

And visualize again

```
gg_miss_var(SocioHealth)
```



The dataset already free from missing values.

- US Covid-19 - Weather

I noticed that this dataset is the largest one, now lets see how much missing values from this dataset!

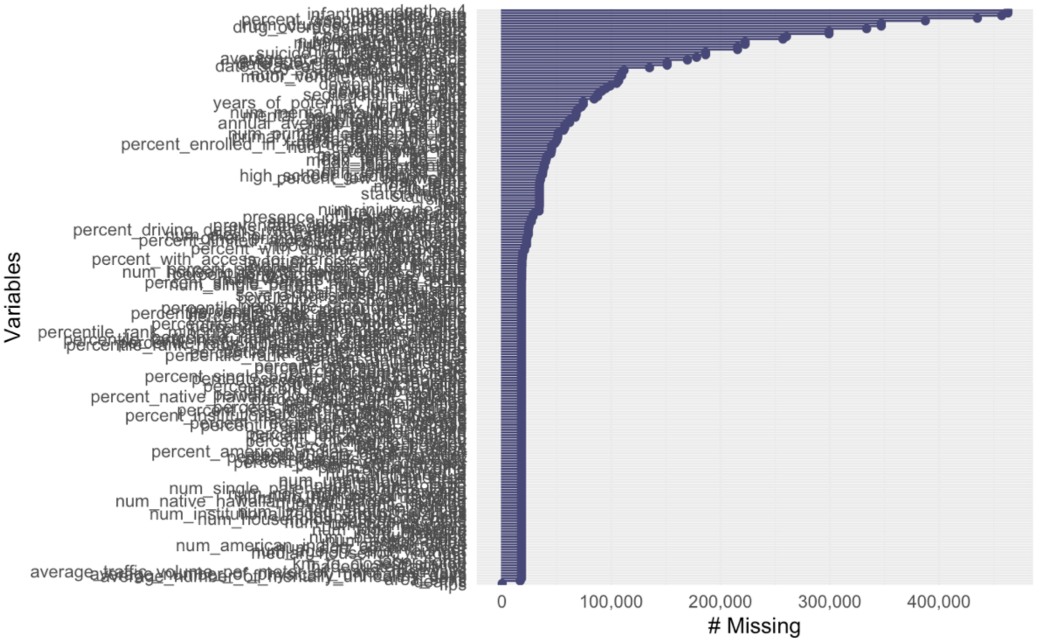
```
sum(is.na(covid))

> 12225014
```

From this dataset only, there are 1.2+ million missing values!

Lets visualize it

```
# Identify columns with missing values
cols_with_missingCovid <- colnames(covid)[apply(covid, 2, anyNA)]

# Filter the data frame to include only columns with missing values
covid_filtered <- covid[ , cols_with_missingCovid]



# Visualize missing values
```

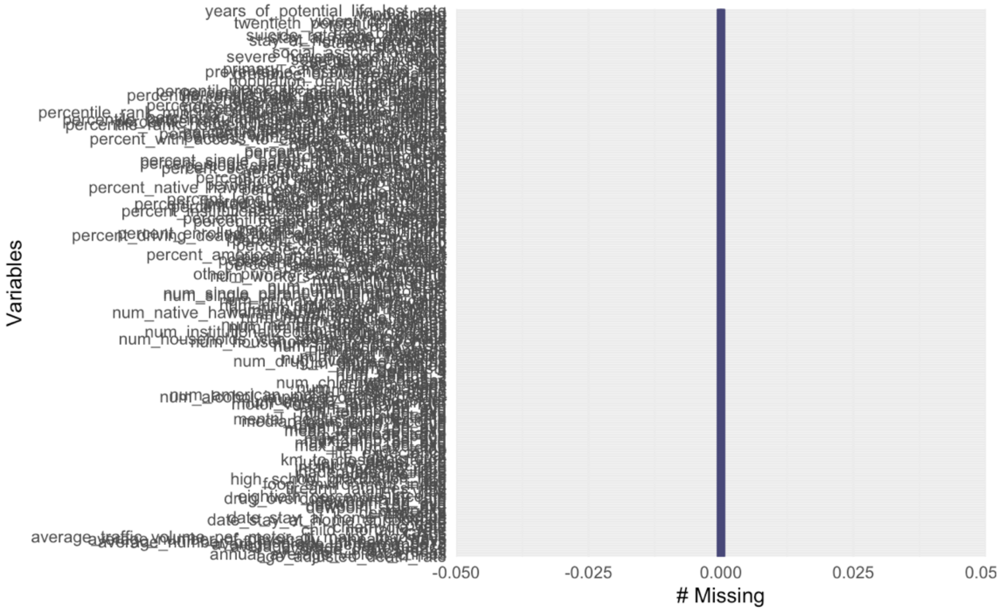
```
gg_miss_var(covid_filtered) + scale_y_continuous(labels = scales::number_format(scale
= 1, accuracy = 1, big.mark = ",", decimal.mark = "."))
```



1 .2 million is not a small number, as we can see here, there are a lot of missing values from a lot of variables, now lets clean it!

```
covid <- na.omit(covid)

> 0
```

And visualize it!



The dataset already free from missing values!

- US Geometry

  This dataset is the smallest (in size) dataset.

  Let’s see if theres missing values in it!

```
sum(is.na(geometryUS))

> 0
```

  There is no missing values from this dataset, which means we don’t need to visualize the missing values.

Now lets see the data distribution for variables that we want to visualize later using Tableu

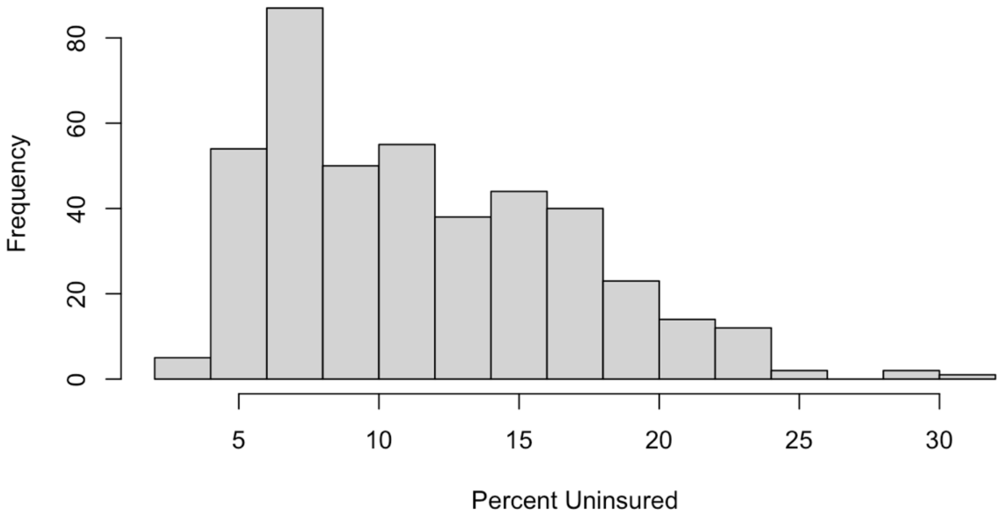
- SocioHealth

  Percent Uninsured

  Before see the distribution for percentage of uninsured citizen, lets see the summary data

```
summary(SocioHealth$percent_uninsured)

>
Min.    1st Qu.  Median   Mean    3rd Qu.   Max.
3.334   6.941    10.555   11.540  15.423    31.208
```

  From all US citizen in this dataset, the average percentage of uninsured citizen is 11.54%, which means there is 1 out of 10 people is uninsured. To make it more understandable, lets visualize it



  From the histogtam, we can see that there are a small frequency of states that has 31.2% of citizen that uninsured.

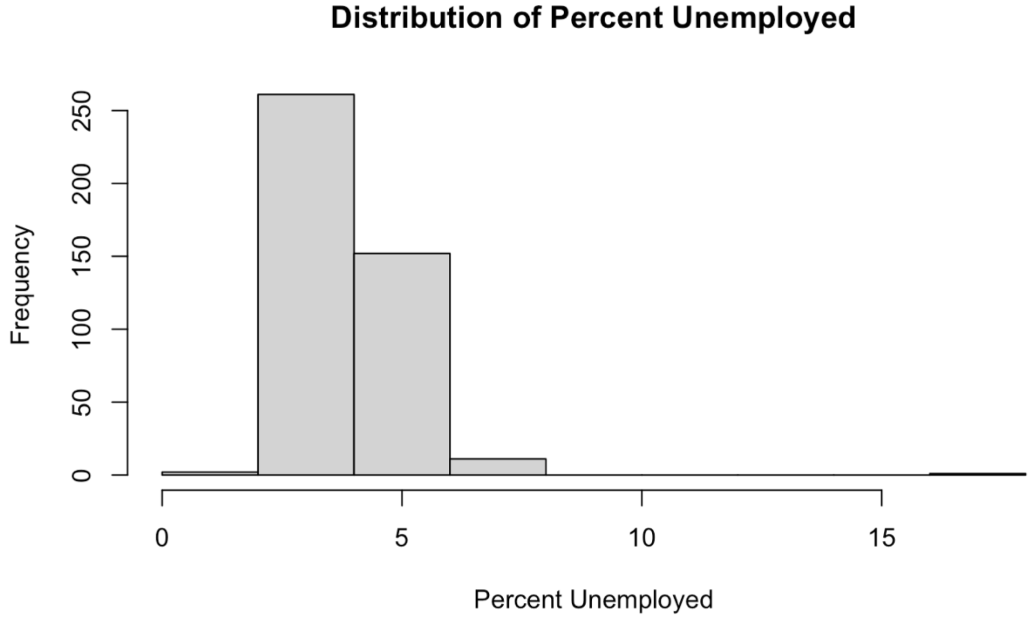
Percent Unemployed

Before see the distribution for percentage of unemployed citizen, lets see the summary data

```
summary(SocioHealth$percent_unemployed_CHR)

>
Min.    1st Qu.  Median    Mean    3rd Qu.    Max.
1.919   3.182    3.687     3.888   4.391      17.042
```

From all US citizen in this dataset, the average percentage of unemployed citizen is 3..88%, which is good, and To make it more understandable, lets visualize it

```
hist(SocioHealth$percent_unemployed_CHR, xlab = "Percent Unemployed", main =
"Distribution of Percent Unemployed")
```



Based on the visualization, there are ~250 county with percentage og unemployed in average is 3.88%.

- Covid-19

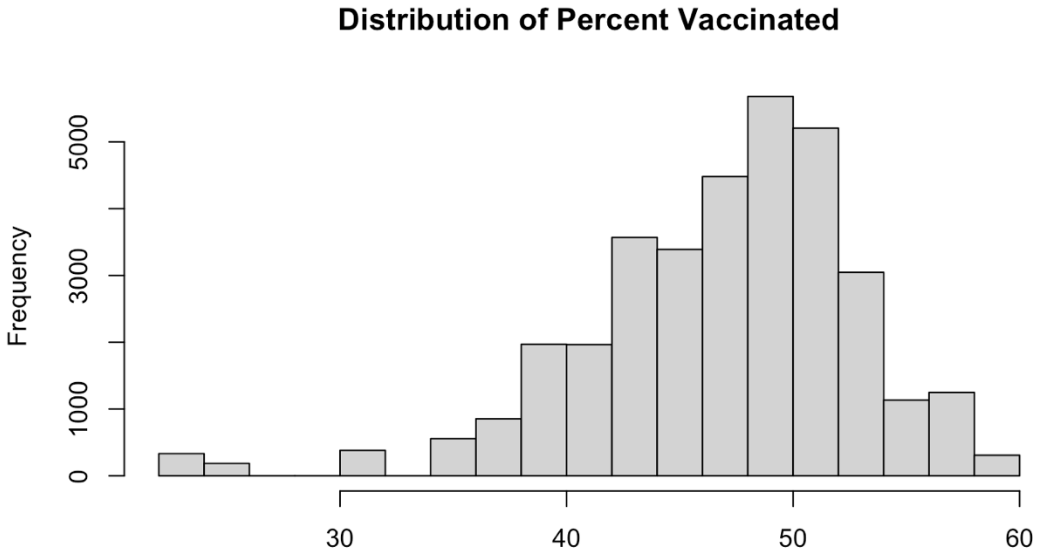
Percent vaccinated

Lets see the percentage of vaccinated citizen in all across US

```
summary(covid$percent_vaccinated)

>
Min.    1st Qu.  Median    Mean    3rd Qu.    Max.
22.0    44.0     48.0      47.3    52.0       59.0
```

Based on the data, the average percentage of vaccinated citizen in all state is 47.3%, which is not good to anticipate the virus, and the max value per percentage vaccinated citizen is still below 70%.

```
hist(covid$percent_vaccinated, xlab = "Percent Vaccinated", main = "Distribution
of Percent Vaccinated")
```

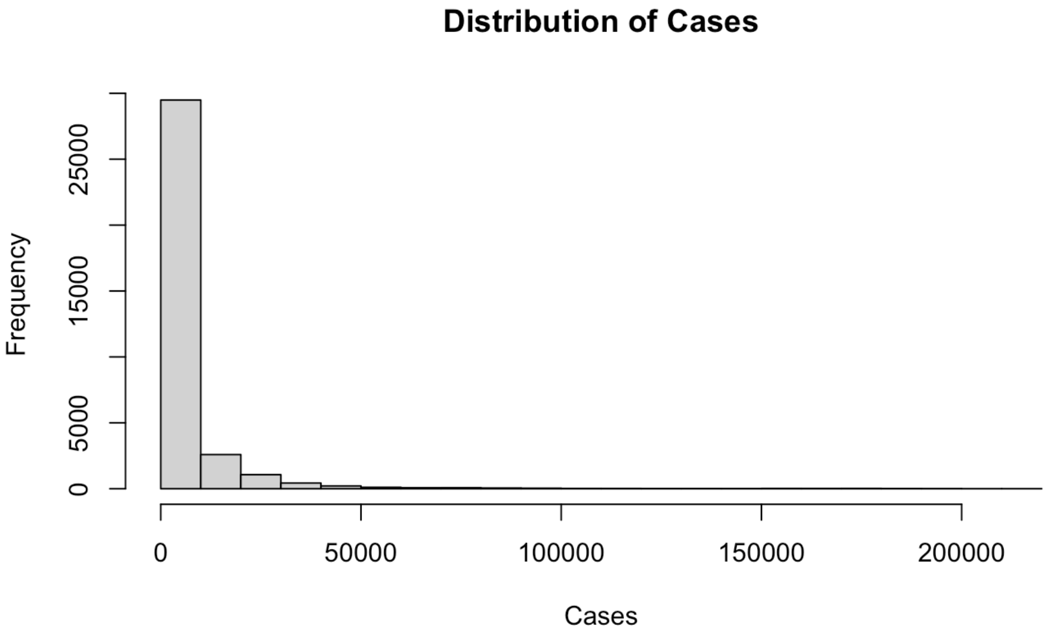


Cases

Lets see how many cases of covid-19 in US (2020 Data)

```
summary(covid$cases)

>
Min.  1st Qu.  Median    Mean    3rd Qu.    Max.
1     259      1528      5704    5212       216441
```

```
hist(covid$cases, xlab = "Cases", main = "Distribution of Cases")
```



Average cases in US states is 5,704 cases, and the highest cases is 216,441cases.

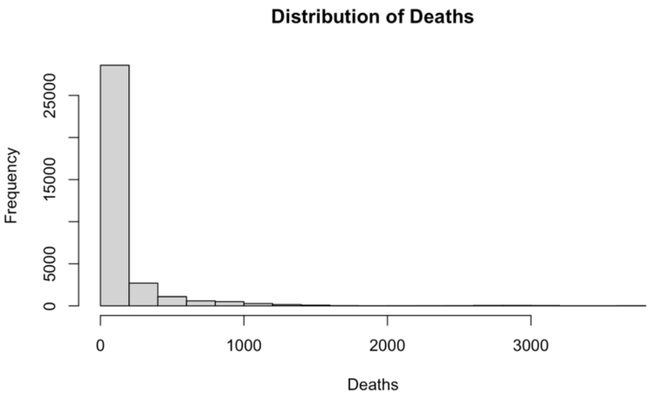
Deaths

Aftes checkin on the number of cases, now lets see the number of death due to covid-19

```
summary(covid$deaths)

>
Min.    1st Qu.  Median    Mean    3rd Qu.    Max.
0.0     6.0      37.0      140.5   120.0      3782.0
```

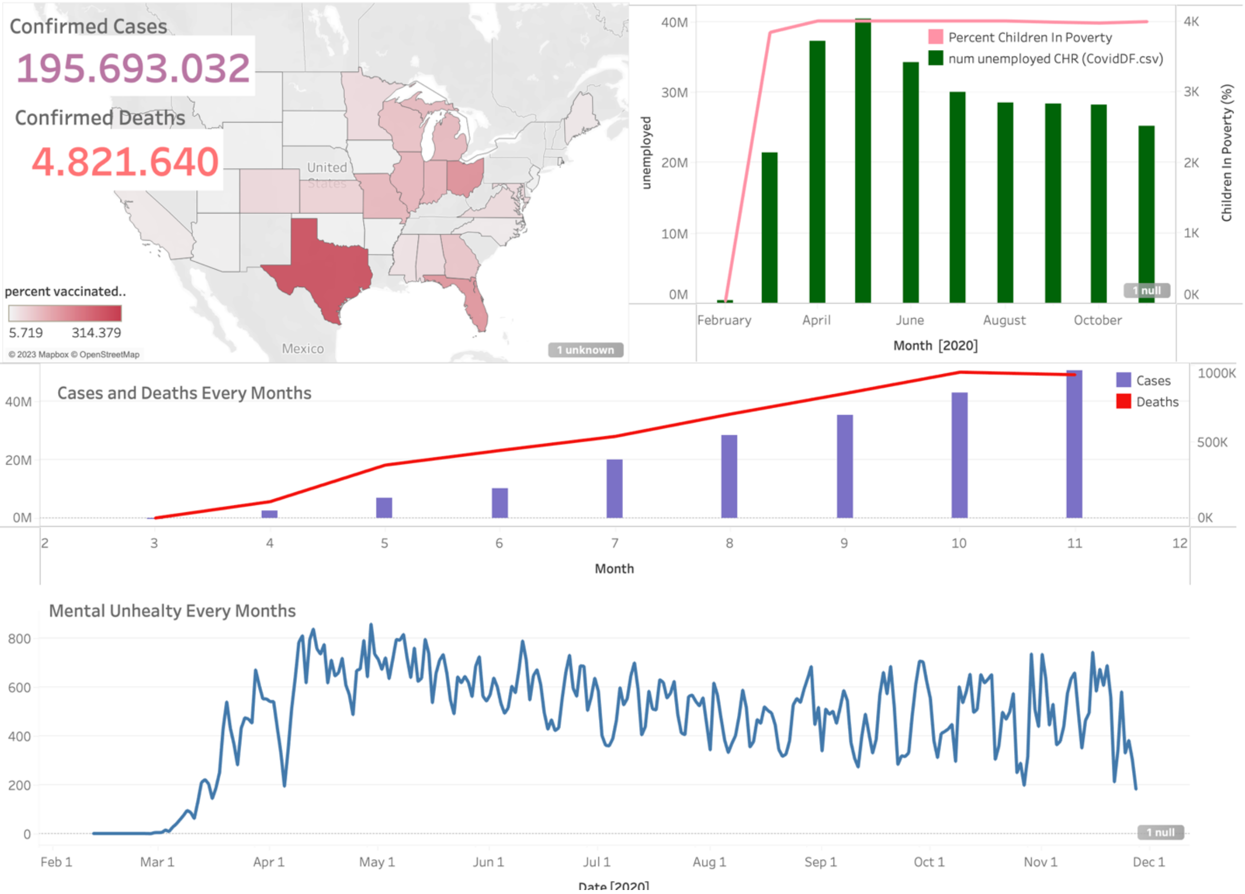
The minimum number of death is 0, which means there are county that has 0 death case due to covid, and the highest death case is 3782 cases.

```
hist(covid$deaths, xlab = "Deaths", main = "Distribution of Deaths")
```



2501977172 – Andi Izzat Zaky Ashari
Data Mining & Visalization [LAB Assignment]

TABLEU DASHBOARD



Based on this visualization, when covid is started to raise, the number of mental unhealthy, unemployed citizen, and childrenn in poverty is also rise following the covid-19 case. Based on data inn 2020, the confirmed cases is 195.6 million cases, and for death cases is 4.8 million.

Covid-19 really have a huge impact globally, from economic side to human mental health.