TERADATA University Network DATA Challenge

Topic: Data Analysis on Hire Heroes USA 2017 Report

Table of Contents

Acronyms and Abbreviations

Sl no	Acronym	Description
1	HHUSA	Hire heroes USA
2	PaCT	Partnered Career Transition

1. Executive Summary

More than 250,000 U.S. military members leave the service each year. Without effective transition assistance, many could join the ranks of more than 500,000 veterans already unemployed or underemployed in the United States. Military spouses also encounter unique job search challenges, leading to an unemployment rate five times the national average. The 2017 Hire Heroes Report is an in-depth analysis of data collected from more than 19,000 US military members, veterans and military spouses who signed up for Hire Heroes USA services in 2017, including nearly 12,000 job seekers who became clients in Hire Heroes' Partnered Career Transition (PaCT) program.

Our research on this data and the valuable insights that are developed from various demographics can be used by other veteran service organizations, think tanks, and federal entities to get a better understanding of the transitioning military, veteran and military spouse community, the challenges and experiences of job seeking veterans and military spouses.

https://www.hireheroesusa.org/about-us/

2. Statement of Scope

The main goal of our project is to gain insights from the data of Hire heroes 2017 report and help Hire heroes understand the transitioning of military community in their career. The findings from our analysis are to be presented for the 2019 Teradata challenge. We have taken Hire Hero's 2017 data as a sample and extending our analysis to entire military population in coming years.

Project Objectives:

To understand and determine the important relationships between the HHUSA client's demographic profile and when they registered for services, who will be a confirmed hire, and finally who will complete the survey.

Target Variable:

Alumni_Survey_Completed__c,

Our goal is to predict Alumni survey Completion based on demographic profiles

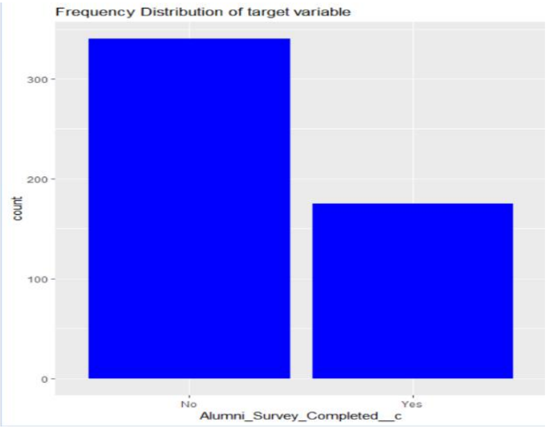
Predictor Variables:

Status__c,Service_Branch__c, MailingState, MailingPostalCode, Service_Rank__c, Military_Spouse_Caregiver__c, Alumni__c, Gender__c,Race__c, Dat_Initial_Assessment_was_Completed__c,

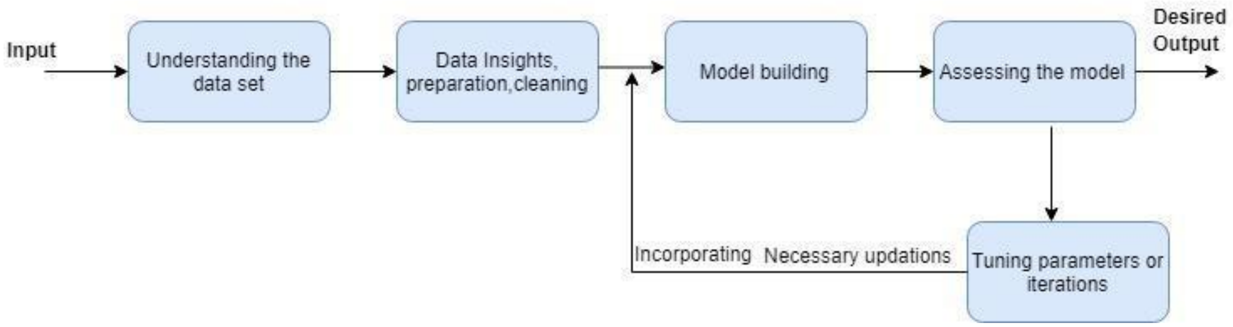
Date_of_Service_EntryNew__c, Date_of_SeparationNew__c, CreatedDate,

Confirmed_Hired_Date__c, DaysTakenToHire, PayGrade, TimeInService ,

YearsInService. One of the target variable in our study is "Alumni_Survey_Completed__c" , which has categorical values of 0's and 1's. We have decoded them as Yes, No for easy understanding. Below is graph showing frequency Distribution of variable Alumni_Survey_Completed__c.



3. Project Flow

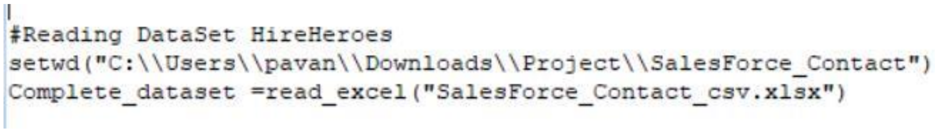

Input
Understanding the data set
Data Insights, preparation,cleaning
Model building
Assessing the model
Desired Output
Tuning parameters or iterations
Incorporating Necessary updations

4.1 Data Preparation

4.1 *Data Access*

Initially, we opted to go for the Teradata university network Data challenge. Once we got the approval, and credentials, we downloaded all the Hire Heroes data sets and supporting documents, which were about 9 excel files in .csv format and 11 MS-word documents.

Reviewing these documents helped us understand the background and workflow of Hire Heroes USA. Their documents also contain information about the operations, flow charts and color codes used in the process. The total size of the data set we have choosen is 209MB, and It has 132446 rows with 391 variables. The reason for choosing this dataset is that it contained rich data with all the information about veterans and their hiring patterns. We can use this data to detect relationships between the variables and build models. In addition to these a data dictionary is also given for reference. There were a set of business questions that are predefined. Below code shows the loading of our data set.



```
#Reading DataSet HireHeroes
setwd("C:\\Users\\pavan\\Downloads\\Project\\SalesForce_Contact")
Complete_dataset =read_excel("SalesForce_Contact_csv.xlsx")
```

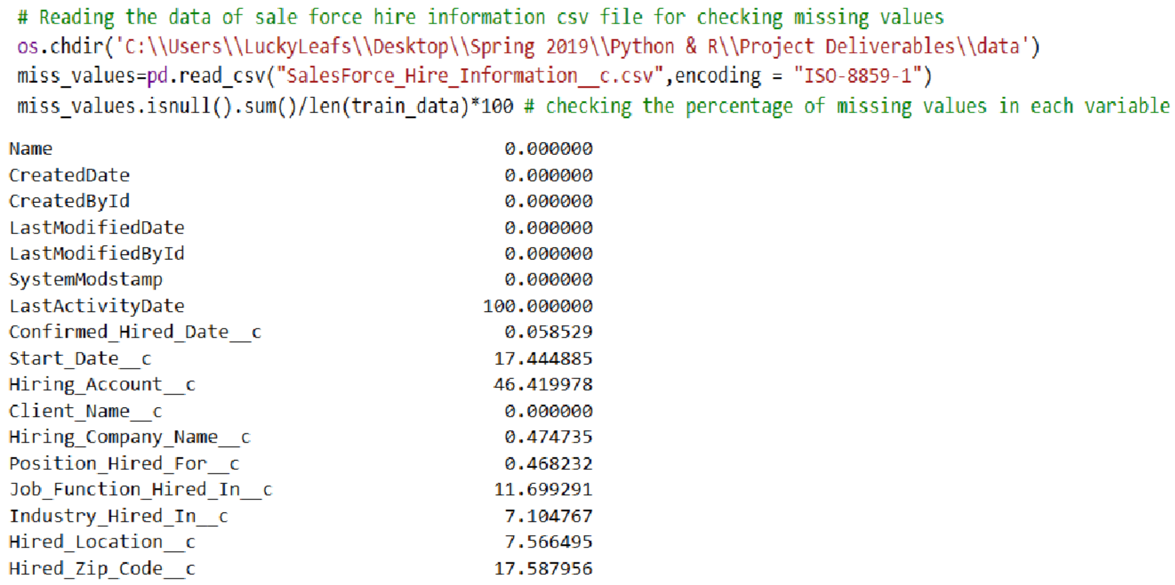
4.2 Data Consolidation

Since the dataset we have chosen has all information about the variables that we need to do analysis. There was no necessity for us to do data consolidation.

4.3 *Data Cleaning*

According to our project scope, we have selected some specific variables which are related to target variable logically. Using the below code all the variables are checked for missing values. The column on the right side gives the percentage of missing values in our data for hire information file. We have excluded columns whose missing values are above 30% from our analysis.

columns with missing values:

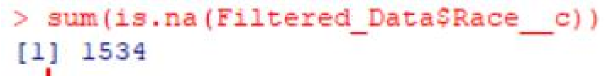


```
# Reading the data of sale force hire information csv file for checking missing values
os.chdir('C:\\Users\\LuckyLeafs\\Desktop\\Spring 2019\\Python & R\\Project Deliverables\\data')
miss_values=pd.read_csv("SalesForce_Hire_Information__c.csv",encoding = "ISO-8859-1")
miss_values.isnull().sum()/len(train_data)*100 # checking the percentage of missing values in each variable
```

```
Name                                0.000000
CreatedDate                         0.000000
CreatedById                         0.000000
LastModifiedDate                    0.000000
LastModifiedById                    0.000000
SystemModstamp                      0.000000
LastActivityDate                  100.000000
Confirmed_Hired_Date__c             0.058529
Start_Date__c                      17.444885
Hiring_Account__c                  46.419978
Client_Name__c                      0.000000
Hiring_Company_Name__c              0.474735
Position_Hired_For__c               0.468232
Job_Function_Hired_In__c           11.699291
Industry_Hired_In__c                7.104767
Hired_Location__c                   7.566495
Hired_Zip_Code__c                  17.587956
```

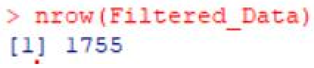
Counting NA Values:

Below is the code to count the NA values for Race_c Variable



```
> sum(is.na(Filtered_Data$Race__c))
[1] 1534
```

Total no of values



```
> nrow(Filtered_Data)
[1] 1755
```

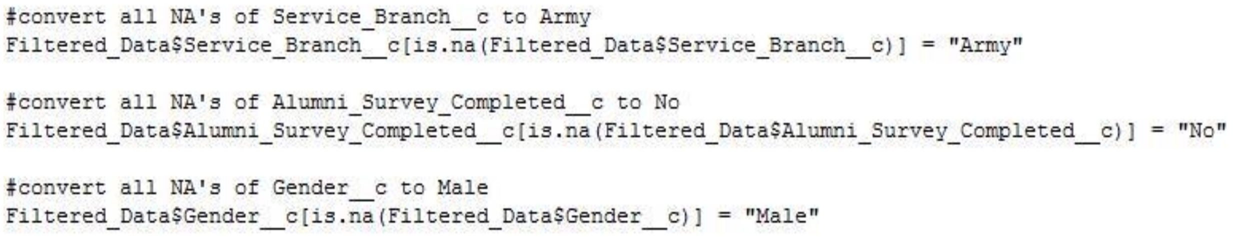
As per our analysis from data set, Race_c variable has more than 80% of NA values, Hence we are ignoring this from further analysis.

The categorical attributes like Service_Branch__c , Alumni_Survey_Completed__c , Gender__c has NA values.

We are replacing the all NA of Categorical variables with their respective Mode Value.

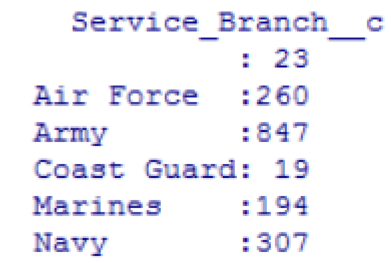
Replacing NA with Mode values

Below is the code for replacing NA’s with variables mode value.

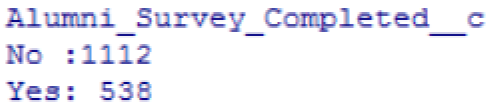


```
#convert all NA's of Service_Branch__c to Army
Filtered_Data$Service_Branch__c[is.na(Filtered_Data$Service_Branch__c)] = "Army"

#convert all NA's of Alumni_Survey_Completed__c to No
Filtered_Data$Alumni_Survey_Completed__c[is.na(Filtered_Data$Alumni_Survey_Completed__c)] = "No"

#convert all NA's of Gender__c to Male
Filtered_Data$Gender__c[is.na(Filtered_Data$Gender__c)] = "Male"
```

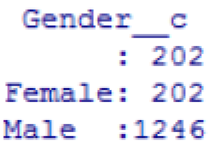
The output is as shown below



```
    Service_Branch__c
              : 23
  Air Force   :260
  Army        :847
  Coast Guard: 19
  Marines     :194
  Navy        :307
```



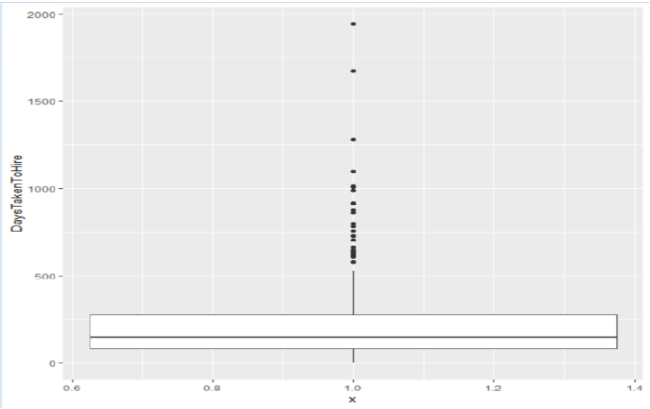
```
Alumni_Survey_Completed__c
No :1112
Yes: 538
```

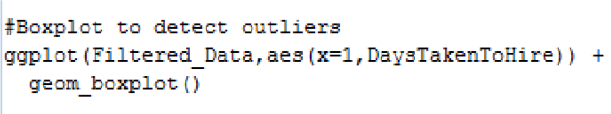


```
 Gender__c
      : 202
Female: 202
Male  :1246
```

Detection of outliers:

As per our observation, “DaysTakenToHire” variable has outliers as shown below.

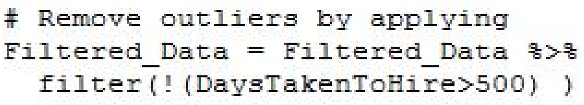




```
#Boxplot to detect outliers
ggplot(Filtered_Data,aes(x=1,DaysTakenToHire)) +
  geom_boxplot()
```

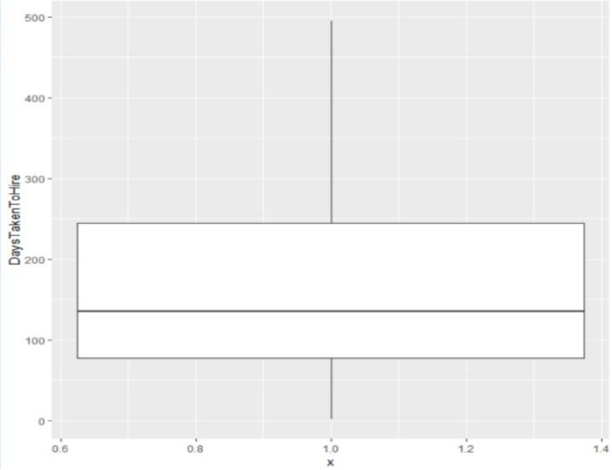
We removed outliers by filtering “DaysTakenToHire” variable

Below is the code for removing the outliers.



```
# Remove outliers by applying
Filtered_Data = Filtered_Data %>%
  filter(!(DaysTakenToHire>500) )
```

The filtered variable has no outliers as shown below in boxplot.

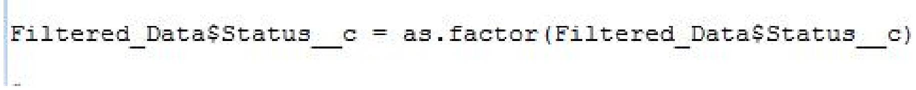


we followed the same procedure to remove the outliers for “DaysTakenToRegester" variable.

Adjustments to data types:

We converted below mentioned character/numeric variables into factor type.

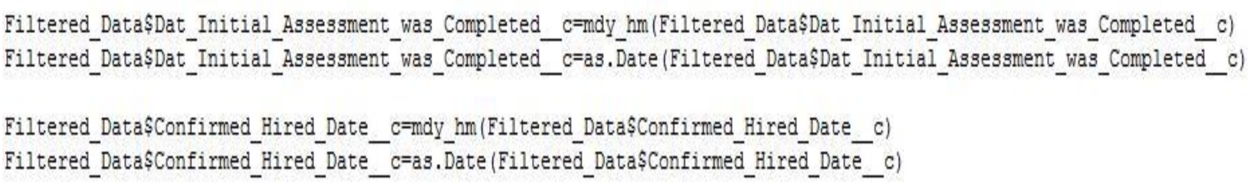
For “Status__c” variable



```
Filtered_Data$Status__c = as.factor(Filtered_Data$Status__c)
```

Similarly, we have converted "Service_Branch__c, MailingState, MailingPostalCode, Service_Rank__c, Military_Spouse_Caregiver__c, Hire_Heroes_USA_Confirmed_Hire__c " variables as factors.

We have converted Initial assessment completed date to Date data type.



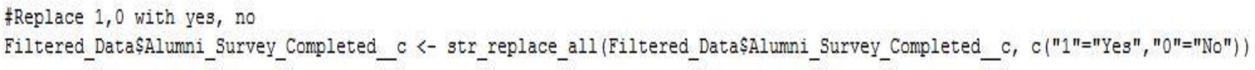
```
Filtered_Data$Dat_Initial_Assessment_was_Completed__c=mdy_hm(Filtered_Data$Dat_Initial_Assessment_was_Completed__c)
Filtered_Data$Dat_Initial_Assessment_was_Completed__c=as.Date(Filtered_Data$Dat_Initial_Assessment_was_Completed__c)

Filtered_Data$Confirmed_Hired_Date__c=mdy_hm(Filtered_Data$Confirmed_Hired_Date__c)
Filtered_Data$Confirmed_Hired_Date__c=as.Date(Filtered_Data$Confirmed_Hired_Date__c)
```

4.3 *Data Transformation*

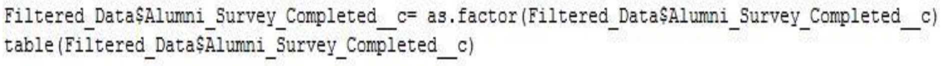
The target variable is decoded as:

Alumni_Survey_Completed__c = Yes; If clients responded to / completed alumni program survey. Alumni_Survey_Completed__c = No; If clients doesn’t respond to / completed alumni program survey

Below is the code for replacing 1,0 with Yes, No.



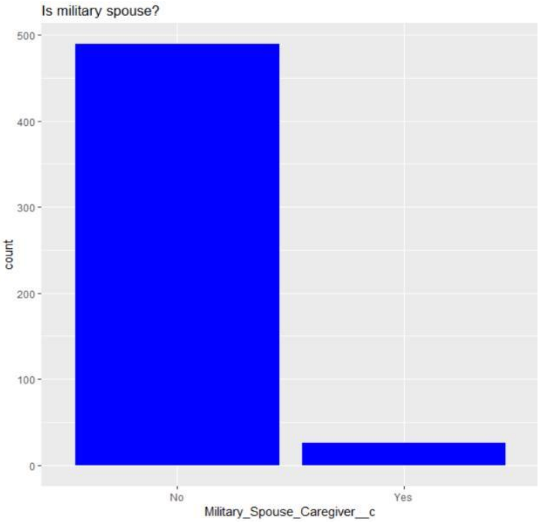
```
#Replace 1,0 with yes, no
Filtered_Data$Alumni_Survey_Completed__c <- str_replace_all(Filtered_Data$Alumni_Survey_Completed__c, c("1"="Yes","0"="No"))
```



```
Filtered_Data$Alumni_Survey_Completed__c= as.factor(Filtered_Data$Alumni_Survey_Completed__c)
table(Filtered_Data$Alumni_Survey_Completed__c)
```

Similarly, we decoded one of the Predictor variable “Military_Spouse_Caregiver__c "

as categorical variable consisting Yes, and No for easy visualization.



Constructing new attributes:

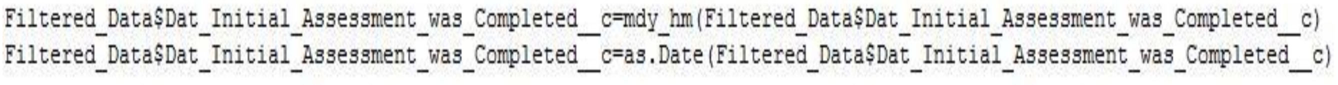
New variable “ DaysTakenToHire ” has been created in R which indicates days taken for clients

to get hired in job.

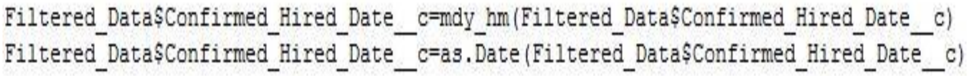
This variable has been created by calculating days difference between

"Confirmed_Hired_Date__c" and “Dat_Initial_Assessment_was_Completed__c " variables.

Below is the code to generate new variable.



```
Filtered_Data$Dat_Initial_Assessment_was_Completed__c=mdy_hm(Filtered_Data$Dat_Initial_Assessment_was_Completed__c)
Filtered_Data$Dat_Initial_Assessment_was_Completed__c=as.Date(Filtered_Data$Dat_Initial_Assessment_was_Completed__c)
```

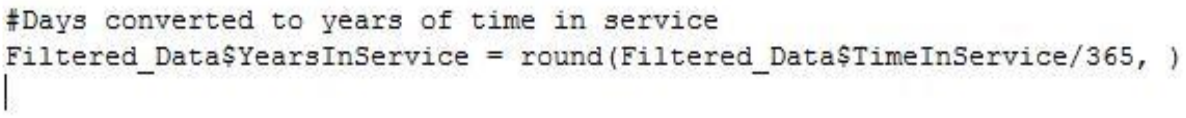


```
Filtered_Data$Confirmed_Hired_Date__c=mdy_hm(Filtered_Data$Confirmed_Hired_Date__c)
Filtered_Data$Confirmed_Hired_Date__c=as.Date(Filtered_Data$Confirmed_Hired_Date__c)
```

Similarly, generated “DaysTakenToRegester " variable by using “CreatedDate , and Date_of_SeparationNew__c " variables, which indicates how many days before or after client’s separation date to register for services.

We also generated “TimeInService " variable by using “Date_of_SeparationNew__c, Date_of_Service_EntryNew__c " variables, which indicates service period of client in military. Later, we converted TimeInServices variables days into years.

Below is the code for converting days to years.

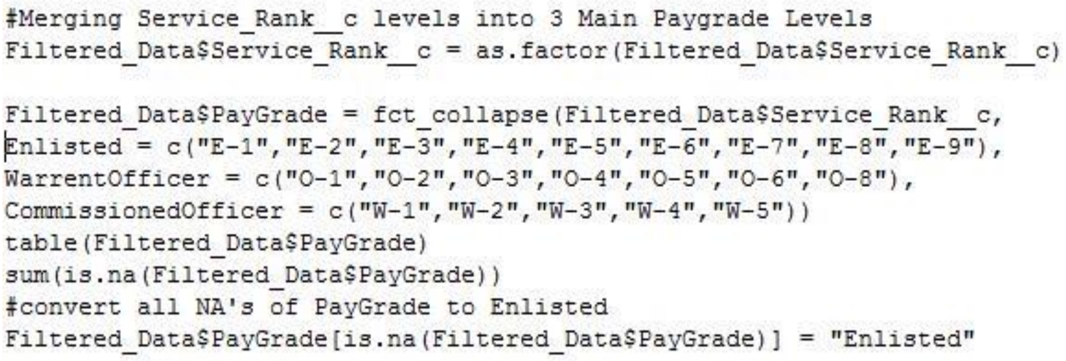


```
#Days converted to years of time in service
Filtered_Data$YearsInService = round(Filtered_Data$TimeInService/365, )
```

We Merged the “Service_Rank__c” variable containing 22 levels into 3 Main Paygrade Levels based on the data from external source to simplify the analysis. Below is official source we took for creating a variable PayGrade.

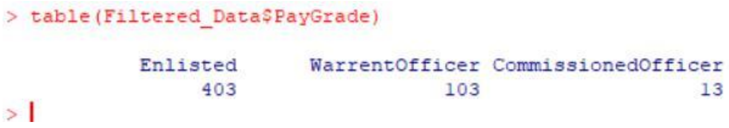
https://en.wikiversity.org/wiki/US_Army_Ranks

The code for merging:



```
#Merging Service_Rank__c levels into 3 Main Paygrade Levels
Filtered_Data$Service_Rank__c = as.factor(Filtered_Data$Service_Rank__c)

Filtered_Data$PayGrade = fct_collapse(Filtered_Data$Service_Rank__c,
Enlisted = c("E-1","E-2","E-3","E-4","E-5","E-6","E-7","E-8","E-9"),
WarrentOfficer = c("O-1","O-2","O-3","O-4","O-5","O-6","O-8"),
CommissionedOfficer = c("W-1","W-2","W-3","W-4","W-5"))
table(Filtered_Data$PayGrade)
sum(is.na(Filtered_Data$PayGrade))
#convert all NA's of PayGrade to Enlisted
Filtered_Data$PayGrade[is.na(Filtered_Data$PayGrade)] = "Enlisted"
```

Below is the distribution of data among Enlisted, WarrentOfficer, CommissionedOfficer



```
> table(Filtered_Data$PayGrade)

           Enlisted      WarrentOfficer CommissionedOfficer
                403                 103                  13
>
```

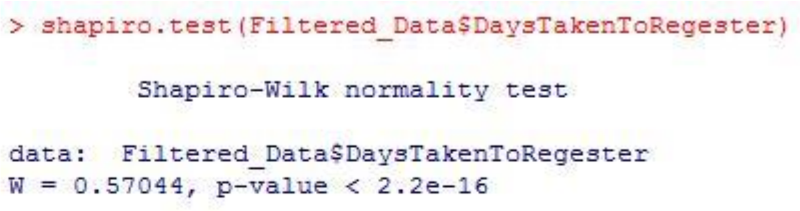
The paygrade variable contains these categories

Normalizing the data:

Normality is a very important assumption that has to be taken care of for the target variable to build the model accordingly.

We have performed Shapiro-wilk test for determining normality for “DaysTakenToRegester "

Below is the code to check the normality:

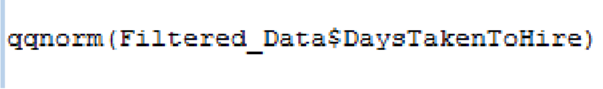


```
> shapiro.test(Filtered_Data$DaysTakenToRegester)

        Shapiro-Wilk normality test

data:  Filtered_Data$DaysTakenToRegester
W = 0.57044, p-value < 2.2e-16
```

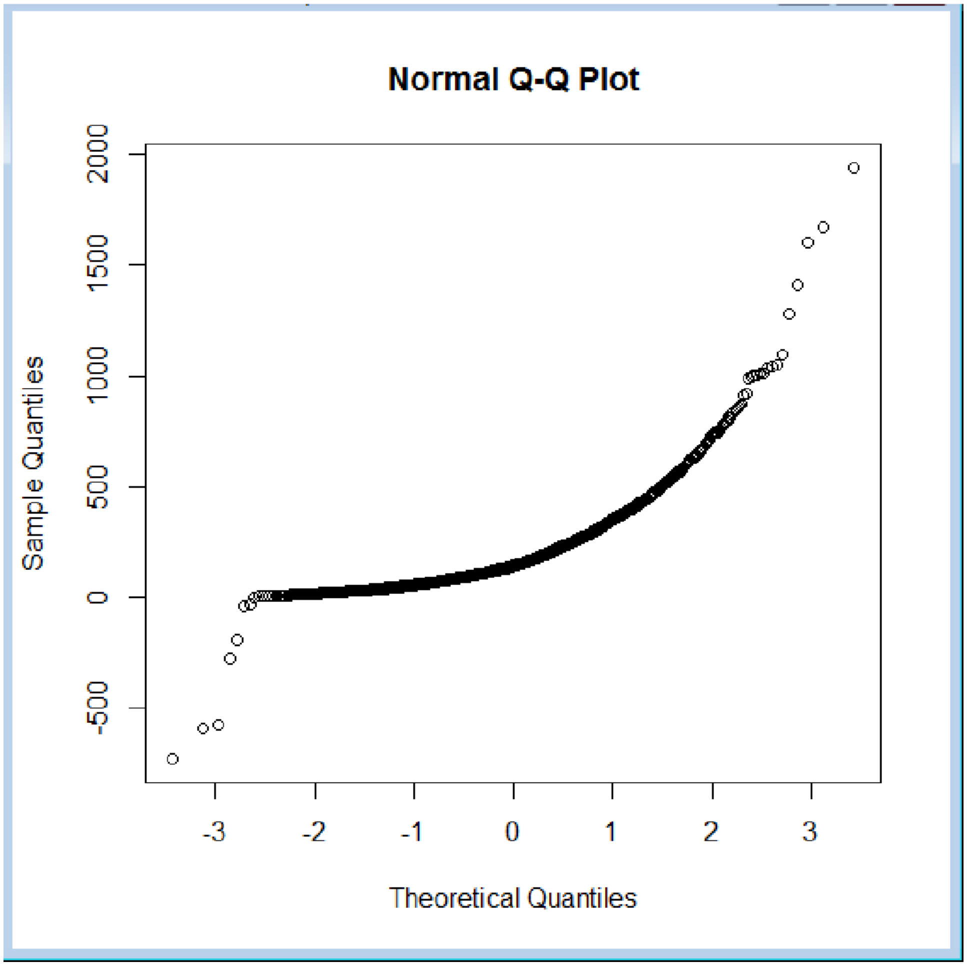
Similarly, we verified the normality for “DaysTakenToHire " ,

For “DaysTakenToHire " variable we used qqplot to determine normality.

Below is the Q-Q plot for “DayTakenToHire”.



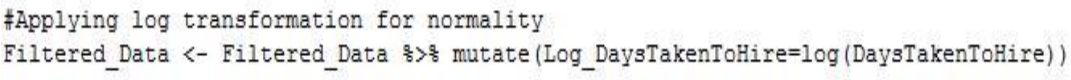
```
qqnorm(Filtered_Data$DaysTakenToHire)
```



Log transformation

We applied Log transformation on “DayTakenToHire” to normalize the variable.

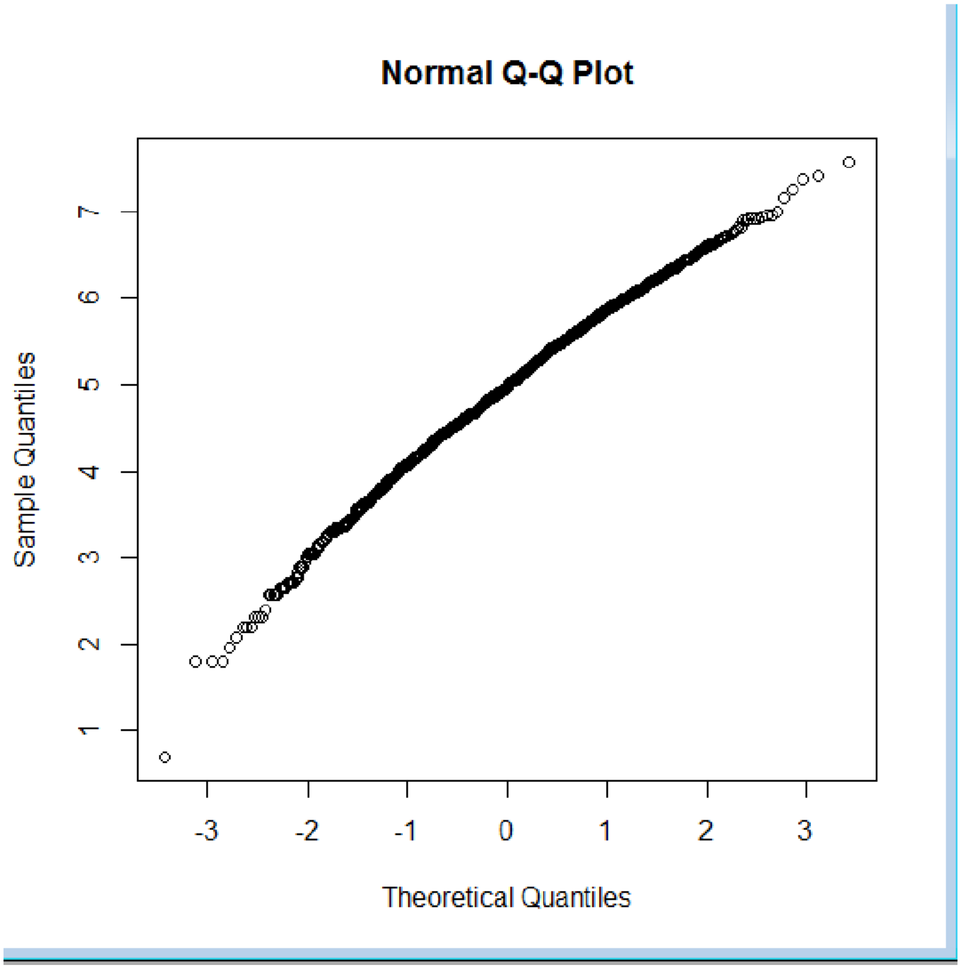
Below is the code for Log Transformation.



```
#Applying log transformation for normality
Filtered_Data <- Filtered_Data %>% mutate(Log_DaysTakenToHire=log(DaysTakenToHire))
```

After Log transformation

```
qqnorm(Filtered_Data$Log_DaysTakenToHire)
```



From the above qqplot we can see that the log transformed variable is perfectly normal.

Similarly, we verified the Normality for “DaysTakenToRegester " variable and applied Log Transformation to normalize the variable.

4.5 *Data Reduction*

As all our independent variables are categorical, we can't use PCA, Factor analysis or clustering for data reduction. We have studied documentation about data and researched internet to find out variables that are logically related to Target variable.

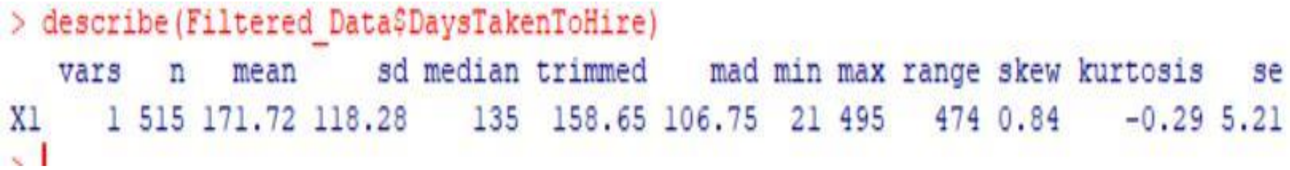
In the selected variables based on missing values percentage and variance of variables we have removed some columns(i.e Race__c, Military_Spouse_Caregiver__c)

4.6 *Descriptive Statistics*

Before we start with our predictive analytics and run the model, let us study few of the variables in detail

For numerical variables

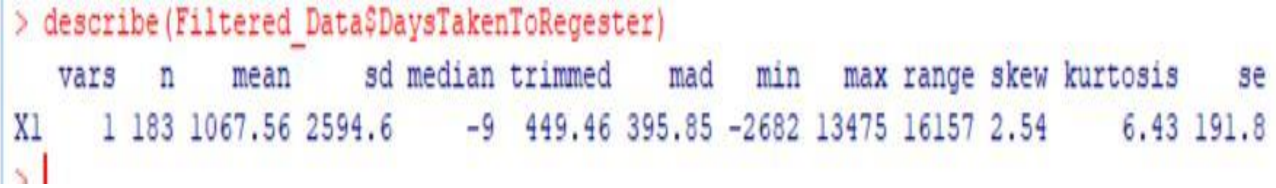
1.**DaysTakenToHire**



```
> describe(Filtered_Data$DaysTakenToHire)
   vars   n   mean     sd median trimmed    mad min max range skew kurtosis   se
X1    1 515 171.72 118.28    135  158.65 106.75  21 495   474 0.84    -0.29 5.21
```

The range of the data is (21,495). We have a mean value of 171.72. Kurtosis and skewness are mostly in limits.

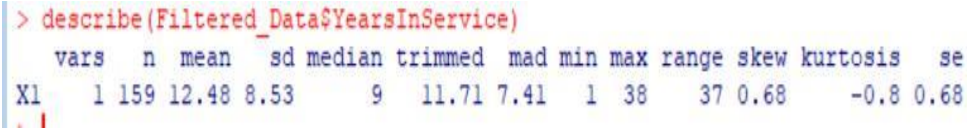
2. **DaysTakenToRegister**



```
> describe(Filtered_Data$DaysTakenToRegester)
   vars   n    mean     sd median trimmed    mad   min   max range skew kurtosis    se
X1    1 183 1067.56 2594.6     -9  449.46 395.85 -2682 13475 16157 2.54     6.43 191.8
```

The range of the data is (-2682, 13475) and mean value is 183 days. Here we got negative no of days because some people are registering in the Hire Heroes even before leaving the military service. Skewness is off the charts, so we need apply some transformations before model building.

1. **YearsInService**

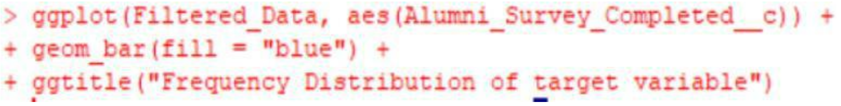


```
> describe(Filtered_Data$YearsInService)
   vars   n  mean   sd median trimmed  mad min max range skew kurtosis   se
X1    1 159 12.48 8.53      9   11.71 7.41   1  38    37 0.68     -0.8 0.68
```

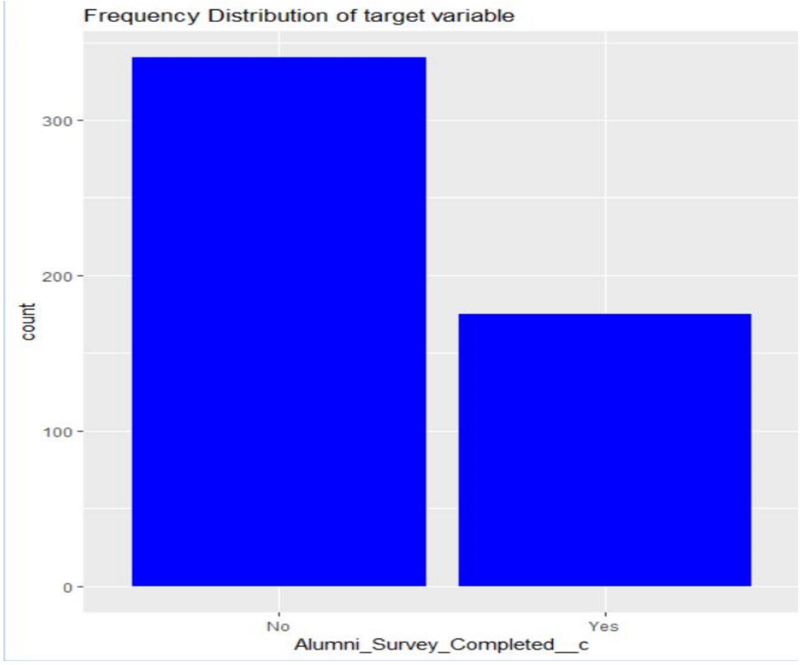
The range of the years in service is (1,38) and the mean years are 12.48. Skewness and kurtosis and skewness are in limits which is good for model building.

For categorical variables

The Frequency Distribution of Target variable Alumni_Survey_Completed__c is

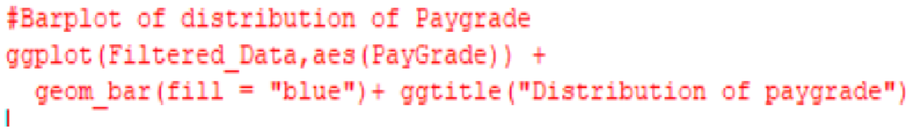


```
> ggplot(Filtered_Data, aes(Alumni_Survey_Completed__c)) +
+ geom_bar(fill = "blue") +
+ ggtitle("Frequency Distribution of target variable")
```

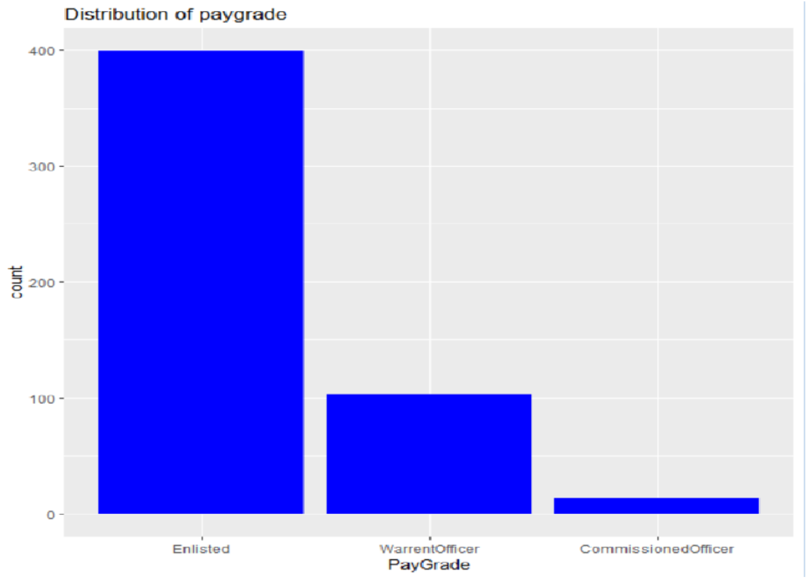


From the above plot we can observe that most of the alumni have not completed the survey.

2. The distribution of paygrade

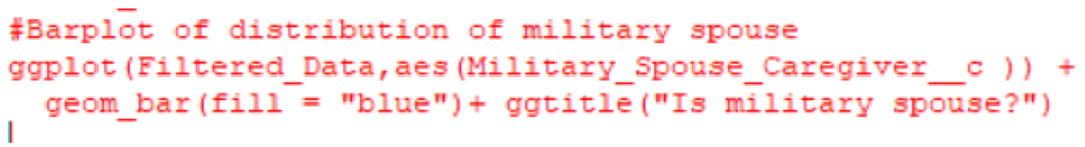


```
#Barplot of distribution of Paygrade
ggplot(Filtered_Data,aes(PayGrade)) +
  geom_bar(fill = "blue")+ ggtitle("Distribution of paygrade")
```

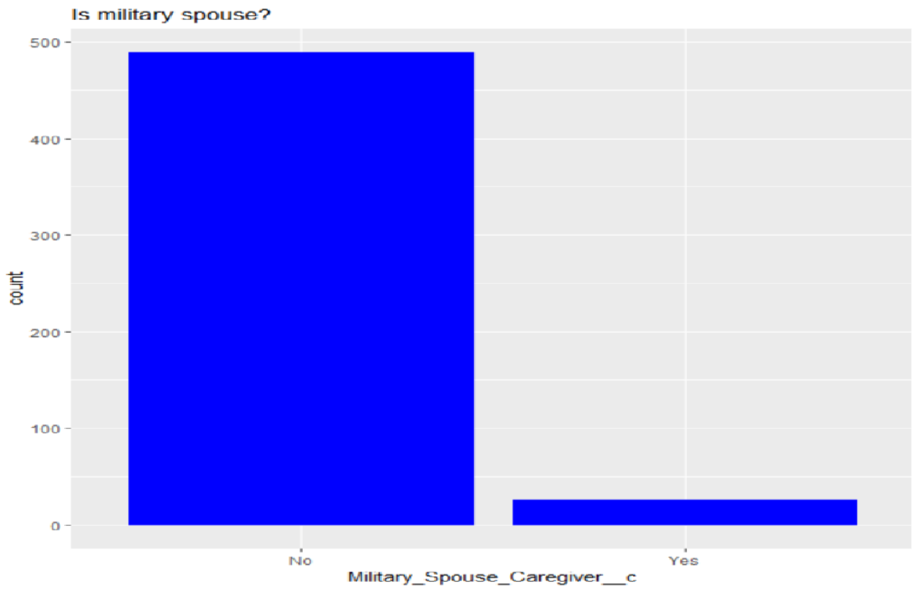


PayGrade is dominated by people with Enlisted status.

Distribution of military spouse.

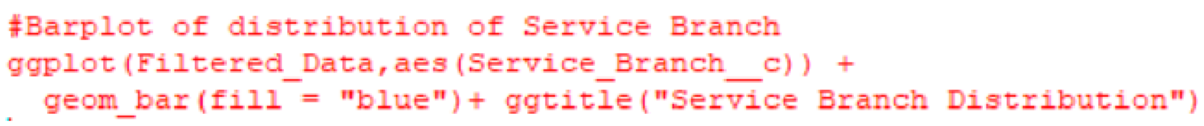


```
#Barplot of distribution of military spouse
ggplot(Filtered_Data,aes(Military_Spouse_Caregiver__c )) +
  geom_bar(fill = "blue")+ ggtitle("Is military spouse?")
```

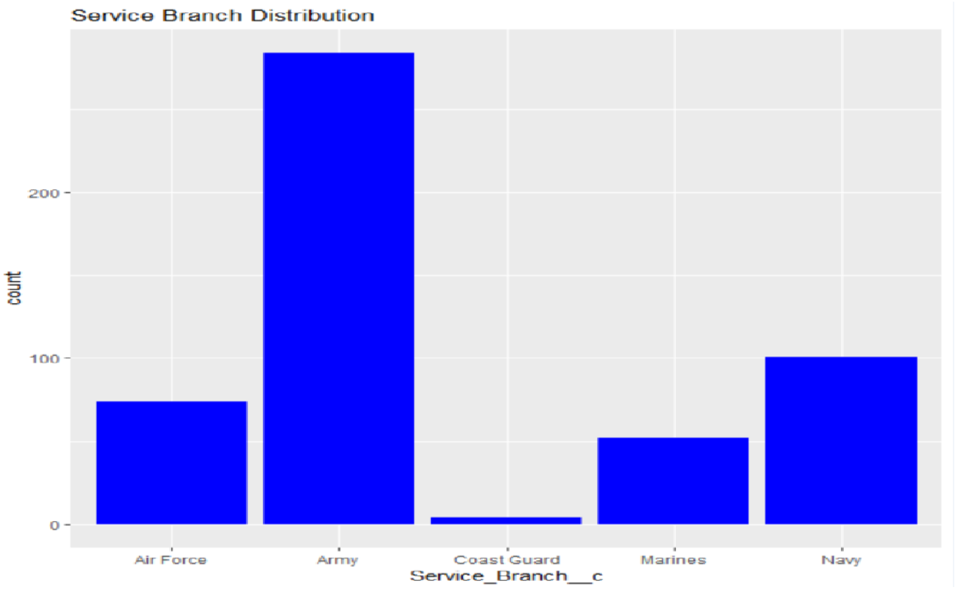


As people with spouse status are very few in our analysis. We can ignore this variable while building models.

Distribution of Service Branch.



```
#Barplot of distribution of Service Branch
ggplot(Filtered_Data,aes(Service_Branch__c)) +
  geom_bar(fill = "blue")+ ggtitle("Service Branch Distribution")
```

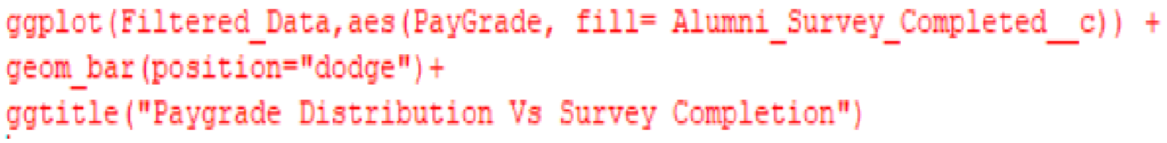


From the above figure most of the alumni we analyzing are from Army.

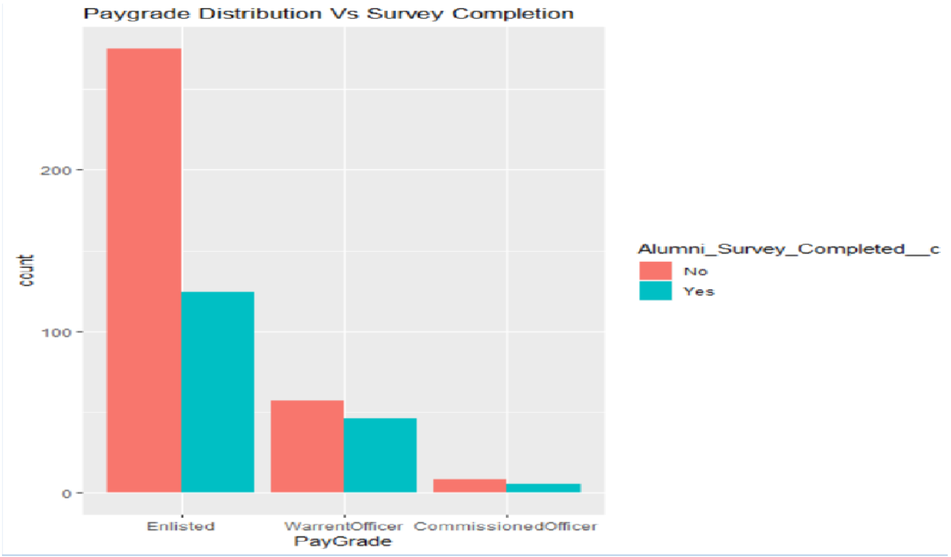
Analysis for both categorical and numerical variables

Now we will look into the demographics of the alumni and their relationship with the survey completion status.

Bar chart showing survey completion stats of people with different pay grades.

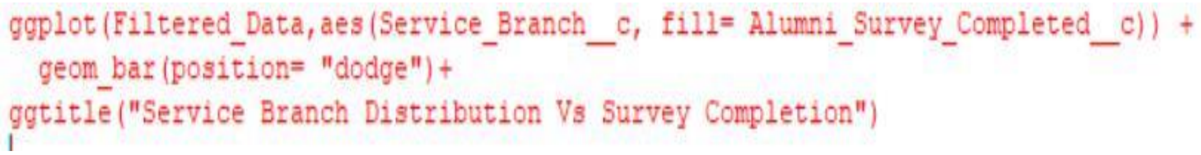


```
ggplot(Filtered_Data,aes(PayGrade, fill= Alumni_Survey_Completed__c)) +
geom_bar(position="dodge")+
ggtitle("Paygrade Distribution Vs Survey Completion")
```

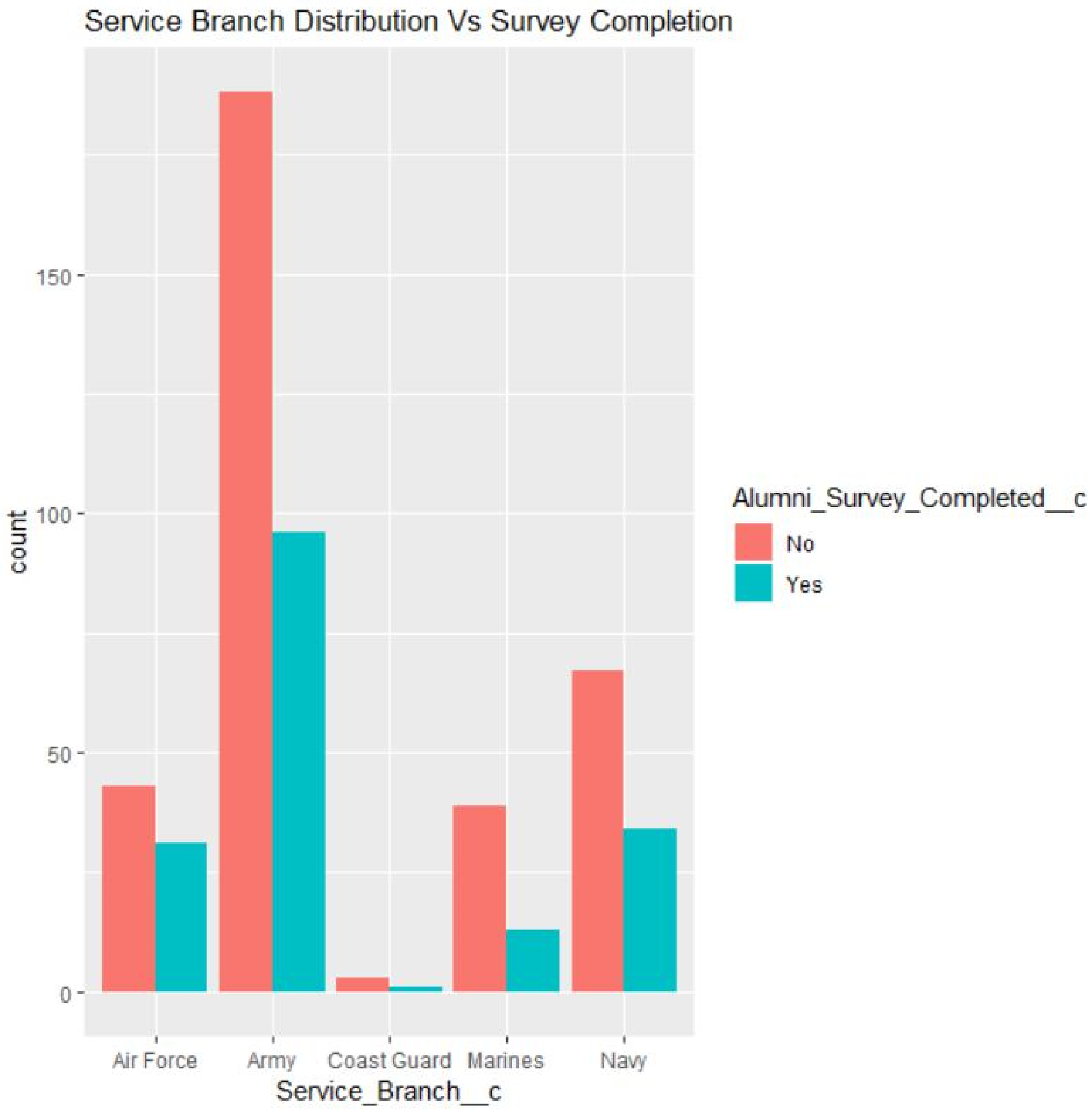


The people are distributed equally

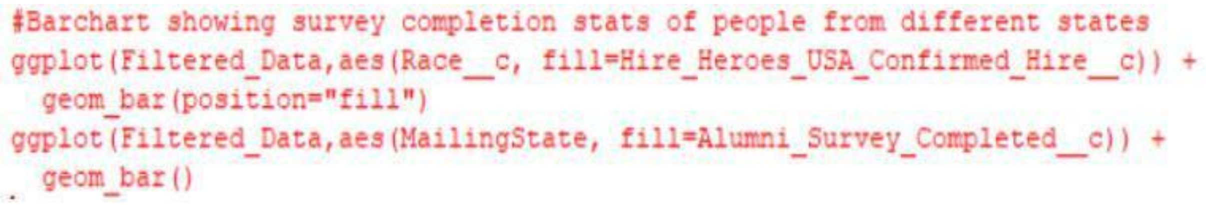
Barchart showing survey completion stats of people from different service branches



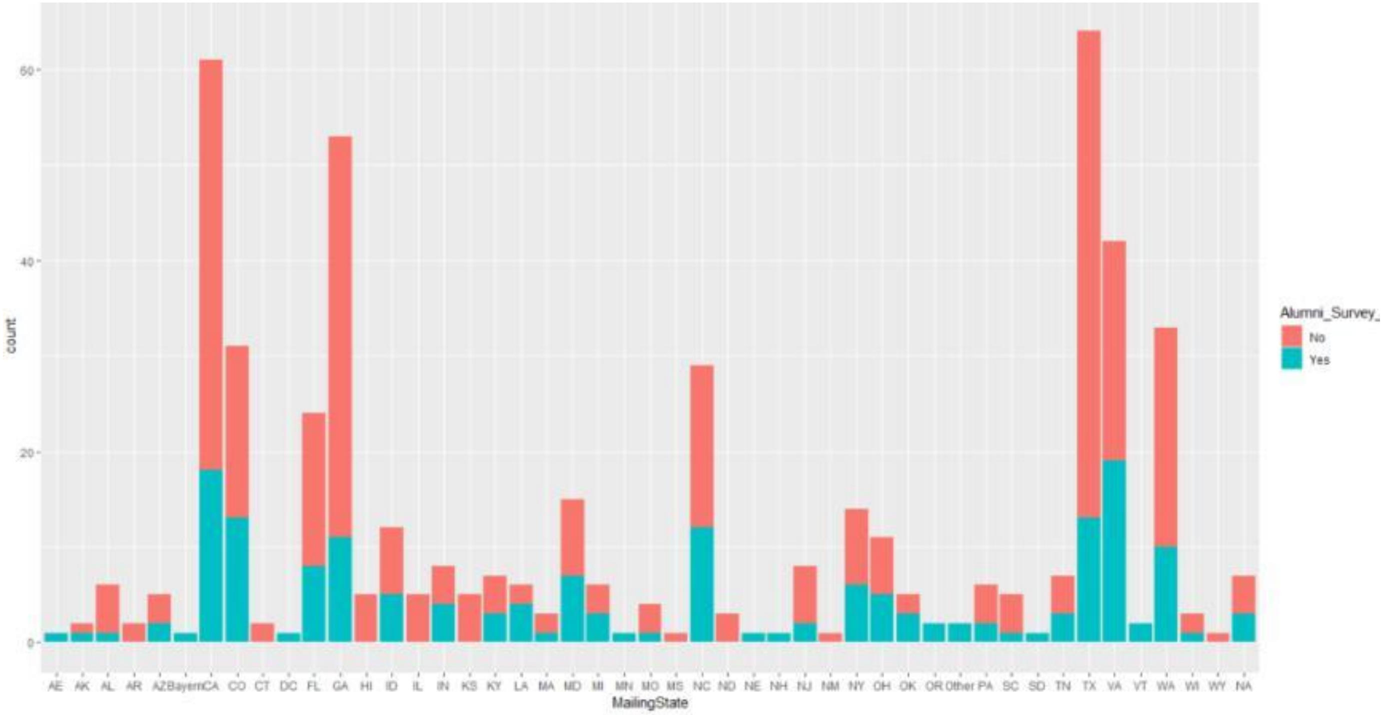
```
ggplot(Filtered_Data,aes(Service_Branch__c, fill= Alumni_Survey_Completed__c)) +
  geom_bar(position= "dodge")+
ggtitle("Service Branch Distribution Vs Survey Completion")
```



Barchart showing survey completion stats of people from different states

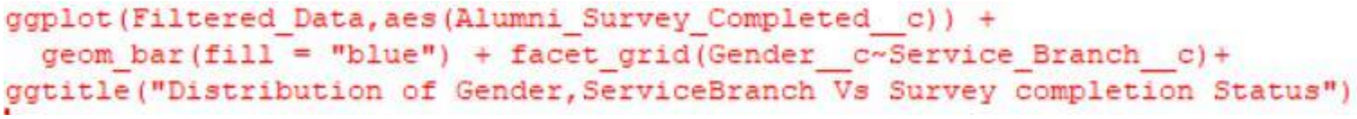


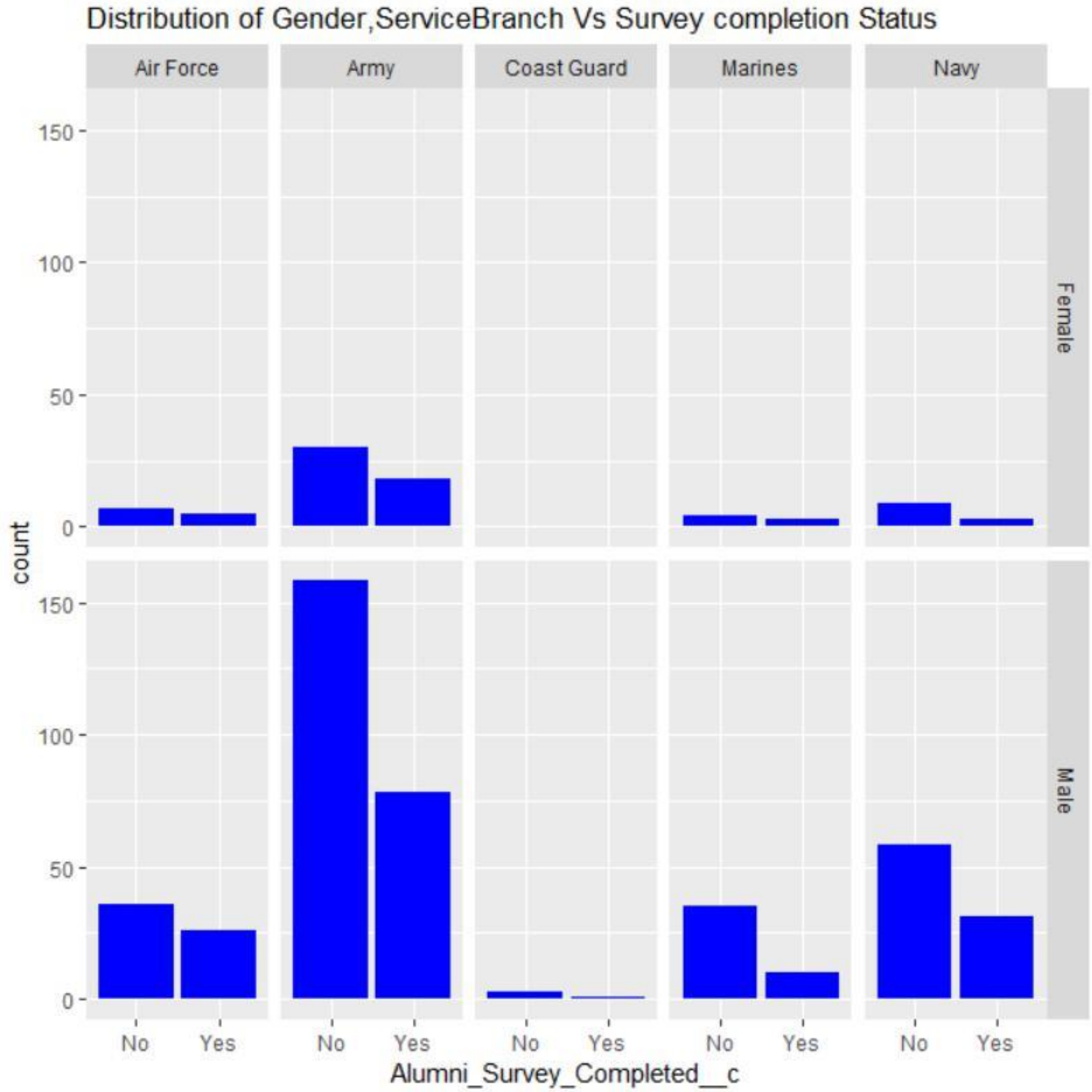
```
#Barchart showing survey completion stats of people from different states
ggplot(Filtered_Data,aes(Race__c, fill=Hire_Heroes_USA_Confirmed_Hire__c)) +
  geom_bar(position="fill")
ggplot(Filtered_Data,aes(MailingState, fill=Alumni_Survey_Completed__c)) +
  geom_bar()
```



Interestingly Virginia state has highest no of people completing survey despite being less in population.

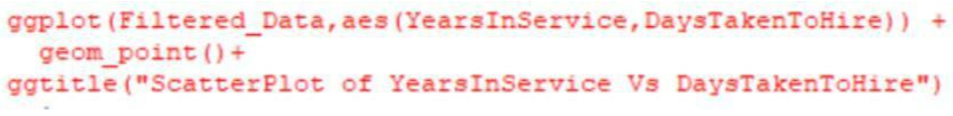
Distribution of Gender,ServiceBranch Vs Survey completion Status


```
ggplot(Filtered_Data,aes(Alumni_Survey_Completed__c)) +
  geom_bar(fill = "blue") + facet_grid(Gender__c~Service_Branch__c)+
ggtitle("Distribution of Gender,ServiceBranch Vs Survey completion Status")
```

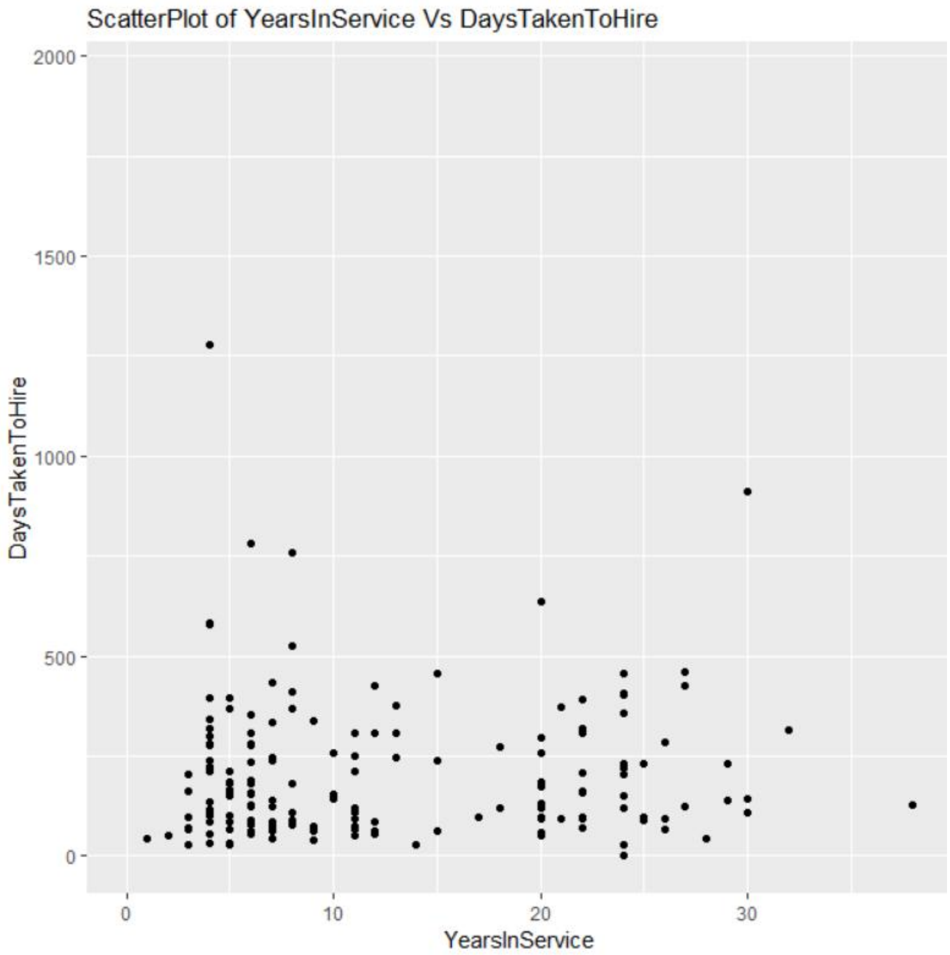


In both genders army has the highest no of people who filled survey, navy follows next.

ScatterPlot of YearsInService Vs DaysTakenToHire

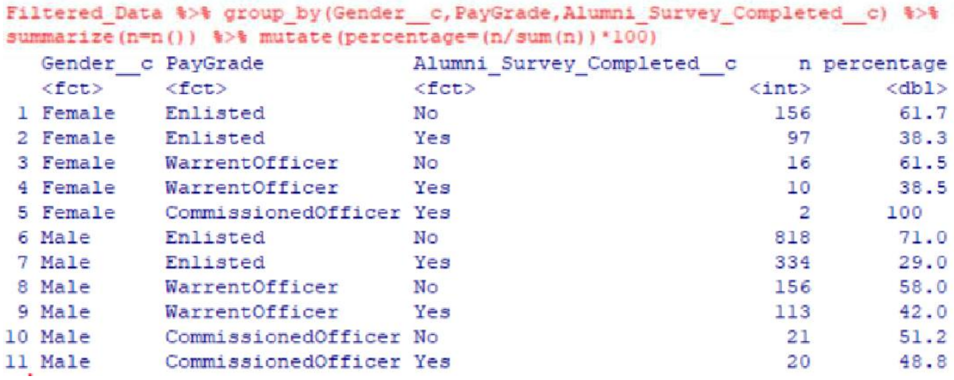


```
ggplot(Filtered_Data,aes(YearsInService,DaysTakenToHire)) +
  geom_point()+
ggtitle("ScatterPlot of YearsInService Vs DaysTakenToHire")
```



There seems to be no great relationship between Years in service and Days taken to hire.

Table showing relative percentages of people grouped to their gender, paygrade and Survey completion status.



```
Filtered_Data %>% group_by(Gender__c,PayGrade,Alumni_Survey_Completed__c) %>%
summarize(n=n()) %>% mutate(percentage=(n/sum(n))*100)
   Gender__c PayGrade            Alumni_Survey_Completed__c     n percentage
   <fct>     <fct>               <fct>                      <int>      <dbl>
 1 Female    Enlisted            No                           156       61.7
 2 Female    Enlisted            Yes                           97       38.3
 3 Female    WarrentOfficer      No                            16       61.5
 4 Female    WarrentOfficer      Yes                           10       38.5
 5 Female    CommissionedOfficer Yes                            2      100
 6 Male      Enlisted            No                           818       71.0
 7 Male      Enlisted            Yes                          334       29.0
 8 Male      WarrentOfficer      No                           156       58.0
 9 Male      WarrentOfficer      Yes                          113       42.0
10 Male      CommissionedOfficer No                            21       51.2
11 Male      CommissionedOfficer Yes                           20       48.8
```

4.7 Data Dictionary

Sl no	Variable	Description	Data Type	Source
1	Status__c	Indicates whether client was unemployed, underemployed, active duty, etc. at time of registration	Factor	Dataset
2	Service_Branch__c	Branch of military service (Army, Navy, etc.)	Factor	Dataset
3	MailingState	State of residence for contact	Factor	Dataset
4	MailingPostalCode	Zip code of residence for contact	Factor	Dataset
5	Service_Rank__c	Most recent pay grade of job seeking client (if military service / E-4, O-7, etc.)	Factor	Dataset
6	Alumni_Survey_Completed__c	True / False (if true, client responded to / completed alumni program survey)	Factor	Dataset
7	Military_Spouse_Caregiver__c	True / False (if true, indicates job seeking client is a spouse of veteran / servicemember and does not have military service)	Factor	Dataset
8	Alumni__c	True / False (has the client opted IN to the HHUSA alumni program)	num	Dataset
9	Gender__c	Gender (M/F)	Factor	Dataset
10	Race__c	Indicated race (white, black or african american, prefer not to answer, etc.)	Factor	Dataset
11	Confirmed_Hired_Date__c	Date Transition Specialist was able to confirm details of the job seeker's new role	Date	Dataset
12	Dat_Initial_Assessment_was_Completed__c	Date of first conversation with Transition Specialist	Date	Dataset
13	Date_of_Service_EntryNew__c	First day of client's military service	Date	Dataset
14	Date_of_SeparationNew__c	Last day of client's military service (actual or anticipated)	Date	Dataset
15	CreatedDate	Date unique record was created within Salesforce	Date	Dataset
16	Hire_Heroes_USA_Confirmed_Hire__c	True / False (indicates the client's new role has been confirmed by an HHUSA staff member)	Date	Dataset
17	DaysTakenToHire	This variable has been created by calculating days difference between "Confirmed_Hired_Date__c" and “Dat_Initial_Assessment_was_Completed__c " variables.	Int	Created variable

18	TimeInService	Created by using difference in Date_of_SeparationNew__c, Date_of_Service_EntryNew__c variables	Int	Created variable
19	YearsInService	TimeInService /365 gives value in years	num	Created variable
20	DaysTakenToRegester	Created by using difference in CreatedDate , and Date_of_SeparationNew__c variables, which indicates how many days before or after client’s separation date to register for services.	Int	Created variable
21	PayGrade	“Service_Rank__c” variable containing 22 levels into 3 Main Paygrade Levels based on the data from external source to simplify the analysis.	Factor	Created variable

5.Modeling Techniques:

5.1 Logistic Regression Model:

Objective:

The objective of this model is to comprehend the effect of selected independent variables (PayGrade,Hire_Heroes_USA_Confirmed_Hire__c,Gender__c,Military_Spouse_Caregiver__c,Service_Branch__c and Status__c) on the target variable “Alumni_Survey_Completed__c. The target variable “Alumni_Survey_Completed__c” is binary variable with values Yes/No.

Concept:

Logistic regression is similar to that of simple linear regression except that logistic regression has binary response variable and its result explains about the impact of each variable on the odds ratio of the observed event of interest, here our event of interest is to determine whether the Alumni completed the survey or not. The log odds ratio is the ratio of two odds and it is a summary measure of the relationship between two variables. The use of log odds ratio in logistic regression provides a more simplistic description of the probabilistic relationship of the variables and the outcome in comparison to linear regression by which linear relationships and more information can be drawn.

5.2 Neural Network:

Neural network can learn the dataset patterns easily and it is skilled enough to deliver much better classification in the case of non-linear boundaries i.e. especially if there more categorical variables. This driven us to move forward with neural network as our second algorithm for data mining as our dataset has more no of Categorical predictors.

5.3 Assumptions

5.3.1 Logistic Regression Model

- Logistic Regression assumes linear relationship between the logit of independent variables and dependent variables, but however there need not be the assumption of linear relationship between the actual dependent and independent variables.

- To obtain fruitful results from logistic regression, sample size must be large as our sample has nearly 1700 observations, so this condition has not been completely satisfied by our data. If there are only few observations reliability of estimation decreases.

- Independent variables should not be linear functions of each other, there are no such variables in our dataset so even this condition holds well with our data.

- There should be no outliers in data, which can be assessed by converting the continuous predictors to standardized, or Z scores and remove the irrelevant values that are out of variable domain.

- The outcome must be discrete or otherwise the dependent variable must be dichotomous in nature (i.e., Yes/No in our dataset)

- Normal distribution is not necessary for the target variable and homoscedasticity is not necessary for each level of independent variables.

5.3.2 Neural Network Model

- Before running this model missing values in the dataset must be imputed. Initially missing values were present in the dataset but during the Data Cleaning phase they were imputed in appropriate way i.e. by mean/median. The neural network is also known as black box model and interpreting results of this model is very difficult. Chances of overfitting of this model are high if algorithm get over trained on dataset.

5.4 Model Goals

Logistic regression

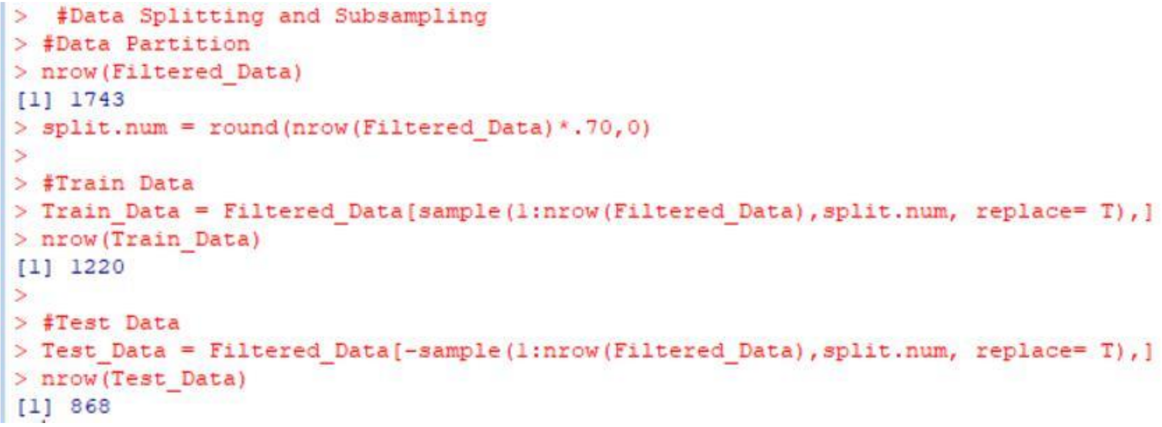
The goal of the logistic regression is to find survey completion stats of Hire heroes alumni that is how logits of odds ratio of Alumni_Survey_Completed__c will be effected for every unit raise in the independent variable (such as status, gender, paygrade, spouse and etc.) that helps in predicting the target variable 'Alumni_Survey_Completed__c'.

Neural Network

Neural Network is powerful computational data model technique that can capture and represent complex input/output relationships. Neural Network acquires knowledge through self-learning. Neural Network is formed by the inter connection of neurons and each neuron has specific synaptic weights assigned to it, which in turn these weights are multiplied with values of independent variables that helps to determine the value of target variable

‘Alumni_Survey_Completed__c’.

6. Data Splitting and Subsampling



```
>  #Data Splitting and Subsampling
> #Data Partition
> nrow(Filtered_Data)
[1] 1743
> split.num = round(nrow(Filtered_Data)*.70,0)
>
> #Train Data
> Train_Data = Filtered_Data[sample(1:nrow(Filtered_Data),split.num, replace= T),]
> nrow(Train_Data)
[1] 1220
>
> #Test Data
> Test_Data = Filtered_Data[-sample(1:nrow(Filtered_Data),split.num, replace= T),]
> nrow(Test_Data)
[1] 868
```

Data splitting is generally the act of partitioning the available data into two portions, usually for cross validation purposes. Cross validation techniques belong to the conventional approaches where we ensure good generalization and to avoid over training. The basic idea is to divide the dataset into two subsets, one is training and the other is testing. Cross-validation techniques can also be used when evaluating and mutually comparing more models, various training algorithms, or when seeking for optimal model parameters

Train set

Training portion of the data is used to build a predictive model as the model sees this set of data while determining the best data transformation and to determine which predictors to include in the model and which one to eliminate.

Test set

The testing set is used only after the model is being build and it is used to compare predictive capabilities across different models and estimate evaluate the model’s performance. Here in our project we are performing the data splitting and the proportion chosen is 70% for the training set and 30% for the test. The idea is that more training data is a good thing because it makes the classification model better while more test data makes the error estimate more accurate.

Reason for choosing 70-30 instead of 50-50

- When we take data set into consideration, we have huge dataset, so we prefer to choose 70-30 split instead of 50-50 split.
- To improve the predictive ability, we choose the 70-30 division for training and testing dataset instead of 50-50.
- Also, higher percent for training data as we want to assure that we have enough data so that we can identify properly the trends for our models as lower percent for training may not recognize the larger events for classification.

Comparison of Categorical Variables:

After analysis of the variables we found there is no much difference in percentage of values between training and testing dataset.

Training Dataset:

Variable	**N**	**Missing**	**No. Levels**	**Mode**	**Mode Percentage**	**Mode Frequency**
Status__c Service_Branch__c	1220 1220	0 0	7 5	Employed Army	42.7% 57.9	522 707
Gender__c	1220	0	2	Male	82%	1011
Military_Spouse_Caregiver__c	1220	0	2	No	94%	1148
PayGrade	1220	0	3	Enlisted	87%	994

Testing Dataset:

Variable	N	Missing	No. of Levels	Mode	Mode Percentage	Mode Frequency
Status__c	868	0	7	Employed	43%	375
Service_Branch__c	868	0	5	Army	55.5%	482
Gender__c	868	0	2	Male	83.9%	729
Military_Spouse_Caregiver__c	868	0	2	No	94%	822
PayGrade	868	0	3	Enlisted	78.5%	682

7.Model Building

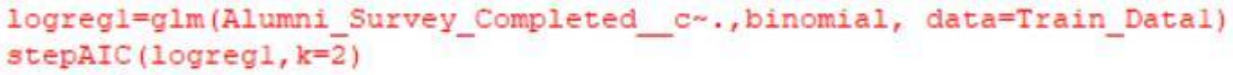
7.1 Logistic Regression

Our dataset consists of categorical variables with more number of levels, so before proceeding with the building logistic regression we need to reduce the total number of variables if this variable reduction has not been performed our model becomes more complex and difficult to interpret, so there is need of performing variable reduction.

Variable Reduction (selecting the variables)

Variable reduction in logistic regression can be performed by either forward selection or backward selection method.

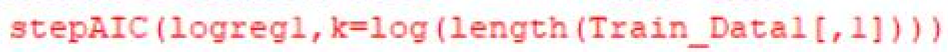
Below shown is the code in R to perform forward selection



```
logregl=glm(Alumni_Survey_Completed__c~.,binomial, data=Train_Datal)
stepAIC(logregl,k=2)
```

By forward selection we got AIC Value as 1462

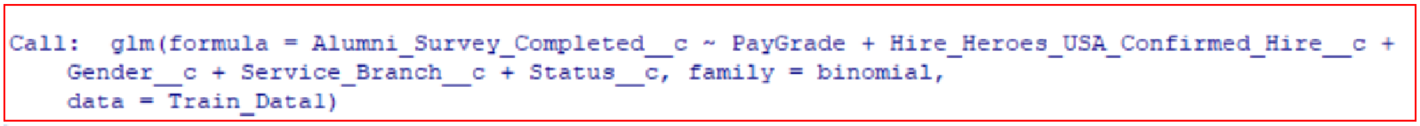
By backward selection we got AIC Value as 1463



```
stepAIC(logregl,k=log(length(Train_Datal[,1])))
```

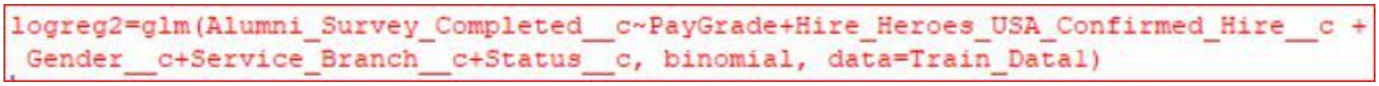
We will take variables selected from forward selection as it has low AIC value.

The variables selected from forward selection are as shown below.



```
Call:  glm(formula = Alumni_Survey_Completed__c ~ PayGrade + Hire_Heroes_USA_Confirmed_Hire__c + 
    Gender__c + Service_Branch__c + Status__c, family = binomial, 
    data = Train_Data1)
```

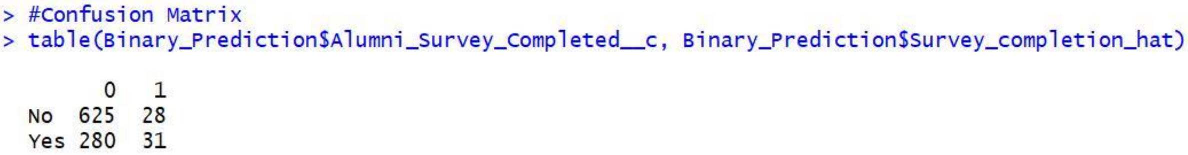
So by considering the variables selected from the forward selection we are going to build our logistic regression and the code to build logistic regression in R is as shown below



```
logreg2=glm(Alumni_Survey_Completed__c~PayGrade+Hire_Heroes_USA_Confirmed_Hire__c +
 Gender__c+Service_Branch__c+Status__c, binomial, data=Train_Data1)
```

#Using augment from broom package to predict probabilities and create confusion matrix Binary_Prediction <- augment(logreg2, type.predict = "response") %>%
 mutate(Survey_completion_hat = round(.fitted))

# Confusion Matrix for train data

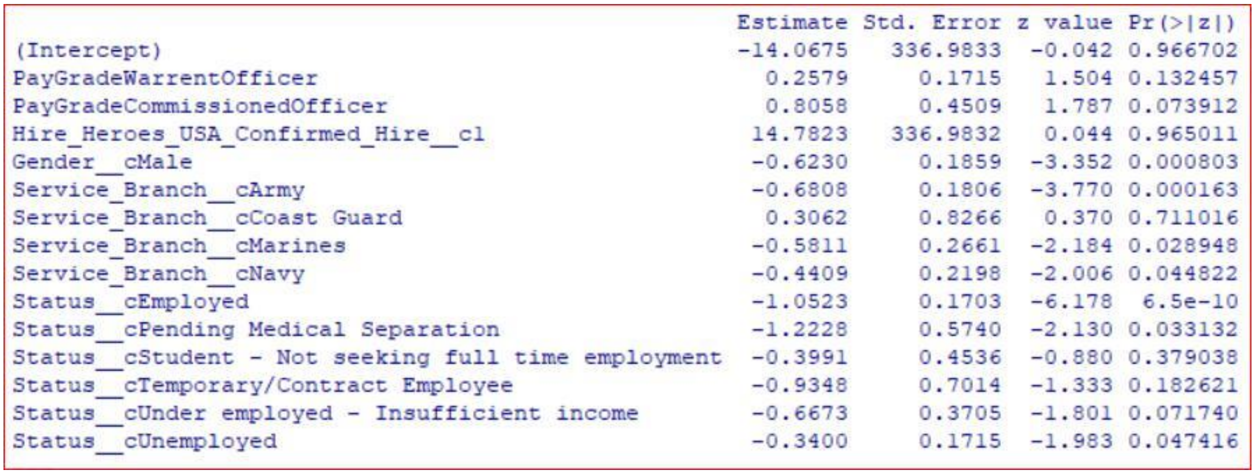


```
> #Confusion Matrix
> table(Binary_Prediction$Alumni_Survey_Completed__c, Binary_Prediction$Survey_completion_hat)
     
        0   1
  No  625  28
  Yes 280  31
```

32.05% is the Misclassification Error

7.1.1 Results and Interpretation

Coefficients:



```
                                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                                   -14.0675   336.9833  -0.042 0.966702
PayGradeWarrentOfficer                          0.2579     0.1715   1.504 0.132457
PayGradeCommissionedOfficer                     0.8058     0.4509   1.787 0.073912
Hire_Heroes_USA_Confirmed_Hire__c1             14.7823   336.9832   0.044 0.965011
Gender__cMale                                  -0.6230     0.1859  -3.352 0.000803
Service_Branch__cArmy                          -0.6808     0.1806  -3.770 0.000163
Service_Branch__cCoast Guard                    0.3062     0.8266   0.370 0.711016
Service_Branch__cMarines                       -0.5811     0.2661  -2.184 0.028948
Service_Branch__cNavy                          -0.4409     0.2198  -2.006 0.044822
Status__cEmployed                              -1.0523     0.1703  -6.178  6.5e-10
Status__cPending Medical Separation            -1.2228     0.5740  -2.130 0.033132
Status__cStudent - Not seeking full time employment -0.3991     0.4536  -0.880 0.379038
Status__cTemporary/Contract Employee           -0.9348     0.7014  -1.333 0.182621
Status__cUnder employed - Insufficient income  -0.6673     0.3705  -1.801 0.071740
Status__cUnemployed                            -0.3400     0.1715  -1.983 0.047416
```

Logistic Regression Equation:

Log (odds ratio of Alumni completes Survey) = -14.0675 + 0.2579*(PayGradeWarrentOfficer)+0.8058*(PayGradeCommissionedOfficer)+14.78*(Hire_Heroes_USA_Confirmed_Hire__c1)-0.6230*(Gender__cMale)-0.6808* (Service_Branch__cArmy)+0.3062*(Service_Branch__Coast Guard)-0.5811*(Service_Branch__cMarines)-0.4409*(Service_Branch__cNavy)-1.0523*(Status__cEmployed)-1.222*(Status__cPending Medical Separation)-0.3991*(Status__cStudent - Not seeking full time employment)-0.9348*(Status__cTemporary/Contract Employee)-0.6673*(Status__cUnder employed - Insufficient income)-0.3400*(Status__cUnemployed)

The regression equation with positive coefficients indicates that there is a raise in the odds of the Alumni completes Survey with subsequent increment in the explanatory variables listed in the equation. If the Explanatory variables are categorical like in our case.

Interpretation:

We can interpret it as for example, if Service_Branch is Coast Guard then log odds of survey completion increases by 0.3062 units. Similarly, regression with negative coefficients indicates that there is fall in the Alumni who completes the survey with subsequent decrease in the corresponding independent variables. For example, if Status__c is Unemployed then log odds of completing the survey decreases by 0.34 units.

Null deviance for the above model is 1388 on 1099 degrees of freedom, this is the deviance value that shows how dependent variable is predicted given the model only includes the intercept value and doesn't include any explanatory variable, similarly residual deviance for the above model is

1301 on 1301 degrees of freedom, this is the deviance value that includes both intercept and explanatory variables in the model.

7.2 Neural networks

Neural network model is built by taking computational effort, training difficulty, dimensionality and comprehensibility into consideration. From PCA analysis, all the variables are selected as input independent variables and the two hidden layers with five hidden units are chosen for model building as model is performing better in terms of misclassification rate. Following variables are used as inputs in the input layer:

PayGrade,

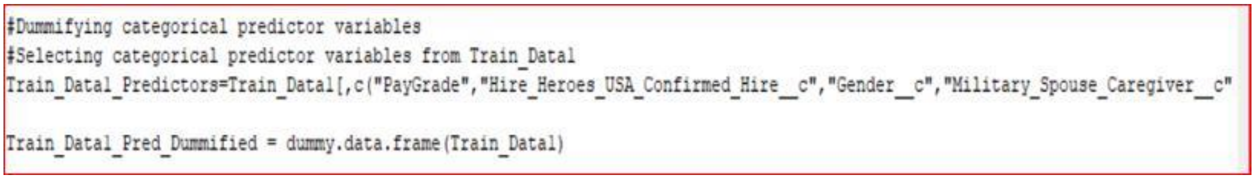
Hire_Heroes_USA_Confirmed_Hire__c

Gender__c

Military_Spouse_Caregiver__c

Service_Branch__c","Status__c

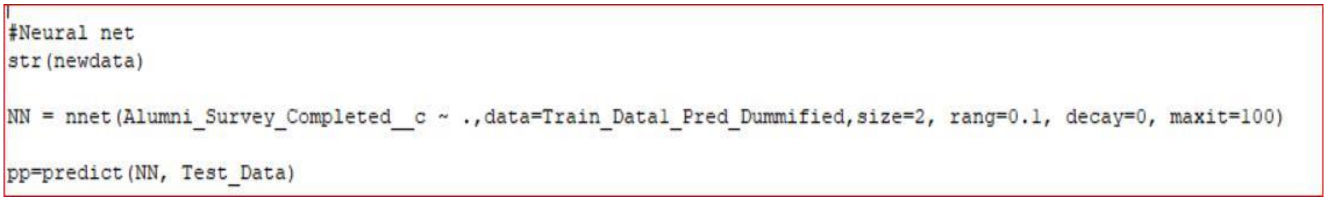
Since the data is completely in categorical form it was required to convert into numerical form to apply the neural network concept. So dummifying the variables is done.



```
#Dummifying categorical predictor variables
#Selecting categorical predictor variables from Train_Datal
Train_Datal_Predictors=Train_Datal[,c("PayGrade","Hire_Heroes_USA_Confirmed_Hire__c","Gender__c","Military_Spouse_Caregiver__c"

Train_Datal_Pred_Dummified = dummy.data.frame(Train_Datal)
```

Neural net model building in R

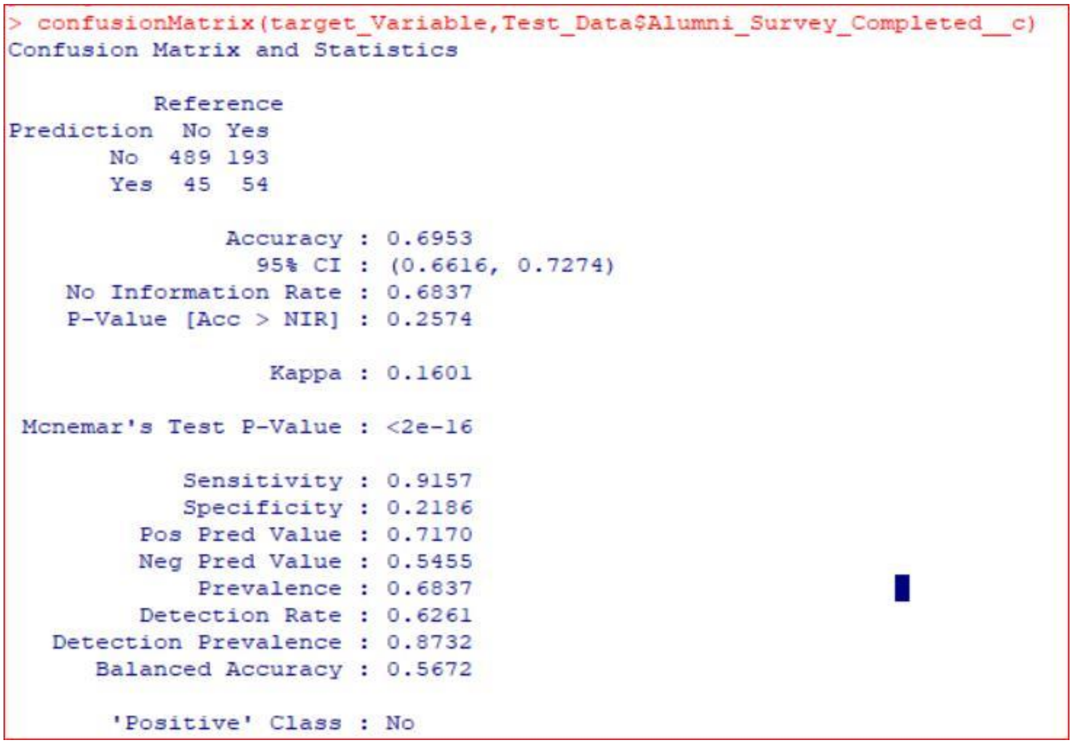
The parameter size is number of hidden layers, Maxit is number of iterations and all other parameters are default values for nnet model



```
#Neural net
str(newdata)

NN = nnet(Alumni_Survey_Completed__c ~ .,data=Train_Datal_Pred_Dummified,size=2, rang=0.1, decay=0, maxit=100)

pp=predict(NN, Test_Data)
```

Results and Interpretations

From the Confusion matrix below we can see that, accuracy of our Neural network model is 69.5%. Predicting is done on test data and then the predicted target variables are compared with true variables in test data using confusion matrix.



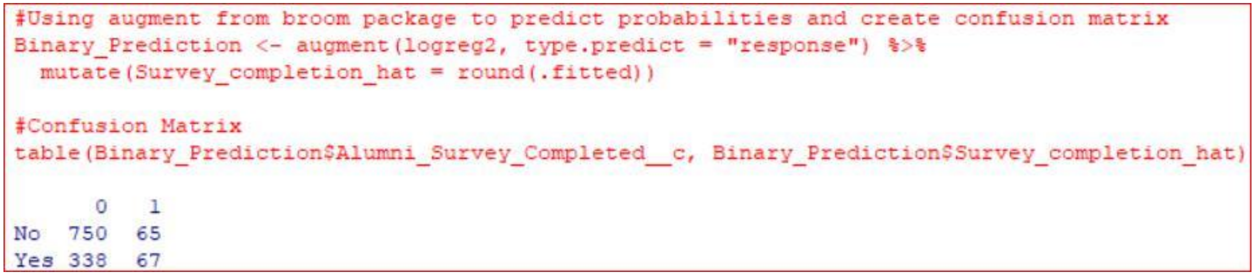
```
> confusionMatrix(target_Variable,Test_Data$Alumni_Survey_Completed__c)
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  489 193
       Yes  45  54

               Accuracy : 0.6953
                 95% CI : (0.6616, 0.7274)
    No Information Rate : 0.6837
    P-Value [Acc > NIR] : 0.2574

                  Kappa : 0.1601

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.9157
            Specificity : 0.2186
         Pos Pred Value : 0.7170
         Neg Pred Value : 0.5455
             Prevalence : 0.6837
         Detection Rate : 0.6261
   Detection Prevalence : 0.8732
      Balanced Accuracy : 0.5672

       'Positive' Class : No
```

8. Model assessments

8.1 Logistic Regression model:

- **Accuracy Measurement:**

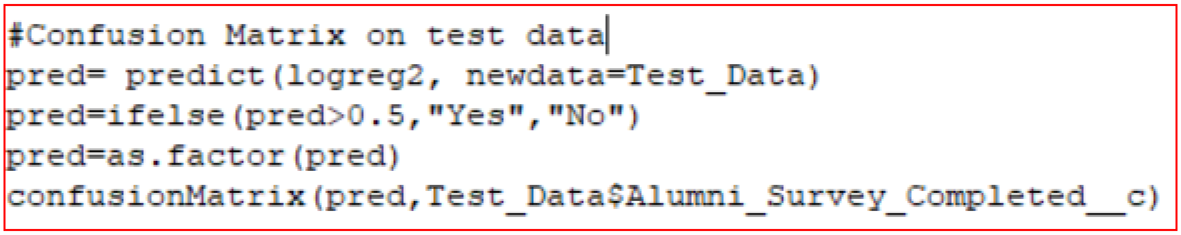
Accuracy of the model is a key metric which determines how well is the classification done by the model. Higher the accuracy means the model has higher predictive ability. The below code shows the code for determining the accuracy of the model on both training dataset and testing dataset.



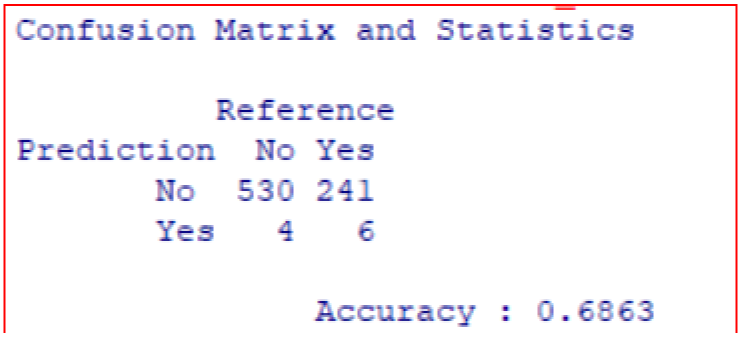
```
#Using augment from broom package to predict probabilities and create confusion matrix
Binary_Prediction <- augment(logreg2, type.predict = "response") %>%
  mutate(Survey_completion_hat = round(.fitted))

#Confusion Matrix
table(Binary_Prediction$Alumni_Survey_Completed__c, Binary_Prediction$Survey_completion_hat)

      0   1
No   750  65
Yes 338  67
```

Model has accuracy of 67% on the train data and 69% on test data.

Confusion matrix provides the tabular summary of predicted class labels v/s actual class labels. It is simpler to understand and can determine the Sensitivity and Specificity of the model. Below is the code for creating the confusion matrix.



```
#Confusion Matrix on test data
pred= predict(logreg2, newdata=Test_Data)
pred=ifelse(pred>0.5,"Yes","No")
pred=as.factor(pred)
confusionMatrix(pred,Test_Data$Alumni_Survey_Completed__c)
```



```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  530 241
       Yes   4   6

               Accuracy : 0.6863
```

The above table shows the values for true positive (TP), true negatives (TN), false positives (FP), false negatives (FN).

Strengths

- Logistic regression doesn’t require any assumption of direct relationship between predictor and target variables.

- The foremost important advantage of logistic regression is to calculate the odds of the dependent variable based on the weights of independent variable.
- Logistic regression can utilize feature of variable reduction/selection while creating the regression model by defining the entry/exit level while executing the model.
- Logistic regression model has high predictive power.

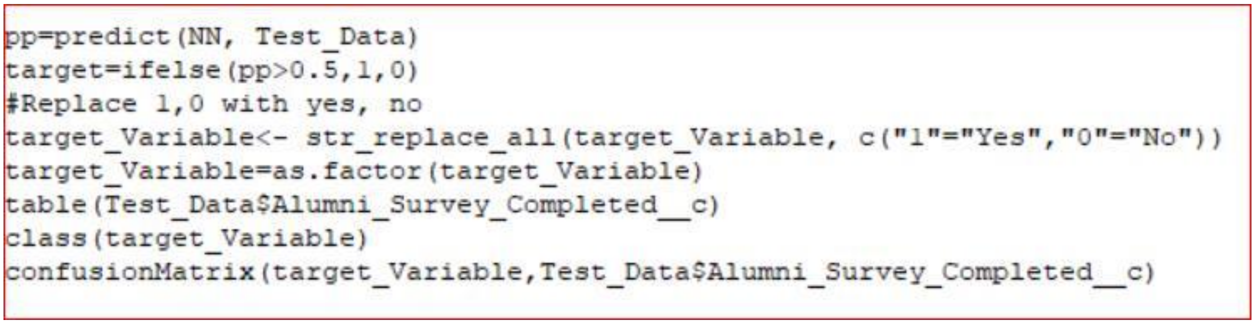
Weaknesses

- Logistic regression is not appropriate for small sample sizes as it produces inaccurate parameter estimates.

8.2 Neural Network Model

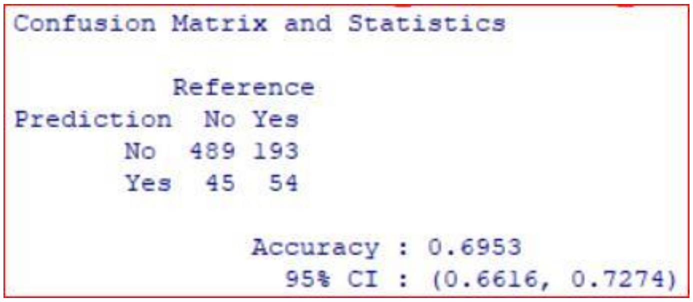
- **Accuracy measurement**

The below R code is to determine accuracy of the model. Higher the accuracy higher is the classification ability of the model.



```
pp=predict(NN, Test_Data)
target=ifelse(pp>0.5,1,0)
#Replace 1,0 with yes, no
target_Variable<- str_replace_all(target_Variable, c("1"="Yes","0"="No"))
target_Variable=as.factor(target_Variable)
table(Test_Data$Alumni_Survey_Completed__c)
class(target_Variable)
confusionMatrix(target_Variable,Test_Data$Alumni_Survey_Completed__c)
```

The accuracy of the model is 70% on test data as shown below from confusion matrix.



```
Confusion Matrix and Statistics

          Reference
Prediction  No Yes
       No  489 193
       Yes  45  54

               Accuracy : 0.6953
                 95% CI : (0.6616, 0.7274)
```

Weakness

- Extracting knowledge from Neural Network is very difficult and it is very hard to explain.
- Neural Network is back box method and it is not flexible with the addition of new data in the model, this can be huge drawback to our model as we keep adding on new customers to our database

Neural Network has less accuracy in comparison with the other models mentioned in the documentation.

Strengths

- The major advantage of neural network is it doesn't require any prior model that is model doesn't assume any structure for model before starting with the Neural Network model
- Neural Network has the ability to train itself to understand the pattern alumni survey completion with independent variables consisting of continuous variables.
- Neural Networks can be used to include independent variables that are not linearly related to the target variables like some of the categorical variables.
- Neural Network is non-parametric method, so chances of creating error in parameter estimation is very less.

9. Final Model Conclusions

To summarize, we initial started off with the descriptive statistics that helps us in finding the summary statistics such as Number of levels, mode and etc. for categorical variables. Then we started with Logistic Regression with both forward and backward variable selection that helps us to eliminate some of the irrelevant variables from the model, Logistic Regression provides the fruitful results such as which variable is significant in predicting survey completion and which variable is affecting survey completion positively and negatively. From this model we got an Accuracy of 67% on testing dataset in predicting survey completion.

Then we proceeded with Neural Network that produced the results that are quite satisfactory where Accuracy is 69%, Misclassification Rate is 31. Even it is quite hard to explain the algorithm that exist behind the working of Neural Network the results are accurate than logistic regression.

So from the above results we can say that **Neural Network can be obtained as final model** as it has high accuracy. We can also consider Logistic regression model as it provides the

equation in most simplified form and it even has significant values that explains most of the variance in the target variable Alumni_Survey_Completed. Considering all these factors we choose Neural Network as best model.

References

https://www.udemy.com/r-programming/

https://support.rstudio.com/hc/en-us

http://monashbioinformaticsplatform.github.io/2015-09-28-rbioinformatics-intro-r/reference.html