MPBA G505 - Statistics & Basic Econometrics

Determining The Factors Effecting Final Placement Offers CTC Using Multiple Linear Regression

Submitted to Dr. Achint Nigam - Department of Management

GROUP 06

Chamarthi Sai Saran	2022H1540808P
Sindhe Pandurang Patel	2022H1540810P
Panuganti Sai Praveen	2022H1540824P
Muttevi Sai Yasasvi	2022H1540831P
Sai Aravind Kashyap	2022H1540832P

Table of Contents

Factors Effecting Final Placement Offers

About Data:

Data Source: Data source will be purely based on live data collected on campus and is observational data. Data collection: The data collection process will not include any data set. Surveys, Google forms and ARC (Alumni Relation Cell), PC (Placement Cell) will be our important sources of data collection.

Variables: variables as in our project statement are

1. Independent variables: CGPA, technologies known, past experience,extracurricular activities, technical and non-technical certifications, coding proficiency and internships

2. Dependent variables: Salary package

Scope of project: There is a bigger scope of finding different patterns, insights that can be inferred based on the regression and can be used for also predictive analytics

• Observational data is considered for regression.

Development path:

1. **Data Collection:** We have gathered the data required for the analysis through Google form survey which was circulated among the final year students who got placed as of now. We gathered 122 students data. Obtained dataset has been usef for nnext sequential operation i.e. data cleansing.

2. **Data cleansing:** We look up the data for any duplicate records, irrelevant data, data type mismatch and correcting the missing values with some basic assumptions based on common prediction.

3. **Exploratory data analysis for initial data insights:** We performed correlation between the dependent and independent variables as there are many independent variables captured and find the best suitable variables based on the correlation coefficient. Based on the results from the correlation, multiple regression will be applied to the best dependent variables to the independent variable and find the best fit line and the influence of the dependent variables to give us the results for our problem statement.

4. **Development:** Code development consists of a. Exploratory Data analysis. b. Correlation analysis for variables. c. Multiple regressions on the suited variables. d. Prediction of salary package based on the model trained.

5. **Testing & Prediction:** Test few samples on the model and predict the value of dependent variable(CTC).

6. **Conclusion:** Automated significant variable section by using backward selection process and found the best fit model.

1. Data collection

Our primary source for data collection was through direct and online survey. The tool that we have used is Google form, we have made google form with CTC as a primary dependent variable and technical proficiency, certifications other variables as independent variables one's. Then initialize the required libraries in R-programming

```
a. Initializing the required libraries


```r
library(tidyverse)
```


```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.
3.2 ──
## ✓ ggplot2 3.4.0      ✓ purrr   0.3.4
## ✓ tibble  3.1.8      ✓ dplyr   1.0.10
## ✓ tidyr   1.2.1      ✓ stringr 1.4.1
## ✓ readr   2.1.2      ✓ forcats 0.5.2
```


```
## Warning: package 'ggplot2' was built under R version 4.2.2
```


```
## Warning: package 'dplyr' was built under R version 4.2.2
```


```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflict
s() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```


```r
library(ggplot2)
library(tinytex)
library(Hmisc)
```
```

```
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
##
## Attaching package: 'Hmisc'
##
## The following objects are masked from 'package:dplyr':
##
##     src, summarize
##
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.2
```

```
## corrplot 0.92 loaded
```

```r
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.2.2
```

```r
library(dplyr)
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.2.2
```

```r
# library(caret)
```

````
b. Importing the raw data set into the environment


```r
path <- "D:/Statistics&Econometrics/Data/survey_data.csv"
test_data_path="D:/Statistics&Econometrics/Data/new_test_data.csv"

# read dataset
df <- read.csv(path)
# columns of raw data
colnames(df)
```

```
## [1] "Package.in.LPA..CTC."
## [2] "CGPA.till.last.semester"
## [3] "No..of.internships.done"
## [4] "No..of.coding.languages.known..Java..C...SQL..Python...."
## [5] "Competitive.Coding.Proficiency"
## [6] "No..of.domain.technologies.known..eg..MAT.Lab..Auto.CAD..."
## [7] "No..of.Technical.certifications.done..AWS..Data.Science..Big.Data....
"
## [8] "No..of.NON.Tech.Certifications.done..CFA..Bloomberg..."
## [9] "No..of.club.memberships..competitions.won.Hackathon..Case.study.compe
titions..."
```
````

2. Data Cleansing

After the raw data got collected, the following actions were performed as a part of data cleansing. Renamed the column's IVs (Independent variables) in the sheet for better fit Replaced the null values from technical certifications from zero to one. Changed the null values of non-technical certifications to zero.Replaced the null values of club membership/competitions/hackathons with zero

````
```r
# rename all columns as below
df <- df %>%
  rename("CTC" = "Package.in.LPA..CTC.",
     "CGPA" ="CGPA.till.last.semester",
     "internships" ="No..of.internships.done",
     "domain_technologies" ="No..of.domain.technologies.known..eg..MAT.Lab..A
uto.CAD...",
     "technical_certifications" ="No..of.Technical.certifications.done..AWS..
Data.Science..Big.Data....",
     "non_tech_certifications" ="No..of.NON.Tech.Certifications.done..CFA..Bl
oomberg...",
````
````

```
     "hackathons" ="No..of.club.memberships..competitions.won.Hackathon..Case
.study.competitions...",
     "coding_languages" ="No..of.coding.languages.known..Java..C...SQL..Pytho
n....",
     "competitive_coding_proficiency"="Competitive.Coding.Proficiency")

# columns after rename
colnames(df)
```

```
## [1] "CTC"                            "CGPA"
## [3] "internships"                    "coding_languages"
## [5] "competitive_coding_proficiency" "domain_technologies"
## [7] "technical_certifications"       "non_tech_certifications"
## [9] "hackathons"
```

Null values Removal

```
#remove null values by replacing it with 1 for Technical certifications
df["technical_certifications"][is.na(df["technical_certifications"])] <- 1

#remove null values by replacing it with 0
#df[is.na(df)] <- 0

#Look at the first few lines of the data frame using the 'head' function
head(df)

##     CTC CGPA internships coding_languages competitive_coding_proficiency
## 1 95.00 8.80           2                9                              9
## 2 69.00 8.68           1                4                              8
## 3 62.10 9.10           2                4                              9
## 4 55.00 8.04           2                3                              8
## 5 51.96 8.50           2                2                              7
## 6 50.00 6.50           2                4                              8
##   domain_technologies technical_certifications non_tech_certifications
## 1                   7                        1                       5
## 2                   6                        7                       4
## 3                   4                        9                       3
## 4                   6                        1                       2
## 5                   1                        1                       3
## 6                   1                        7                      NA
##   hackathons
## 1          3
## 2          4
## 3          1
## 4          4
## 5         NA
## 6          3
```

3. Exploratory data analysis –

As part of observational data analysis, we checked the correlation between dependent variable and all the independent variables along with scatterplots for the same. We also plotted all the graphs so that inference can be made using those plots.

```
a. Summary Statistics


```r
#getting the existing data insights
summary(df)
```

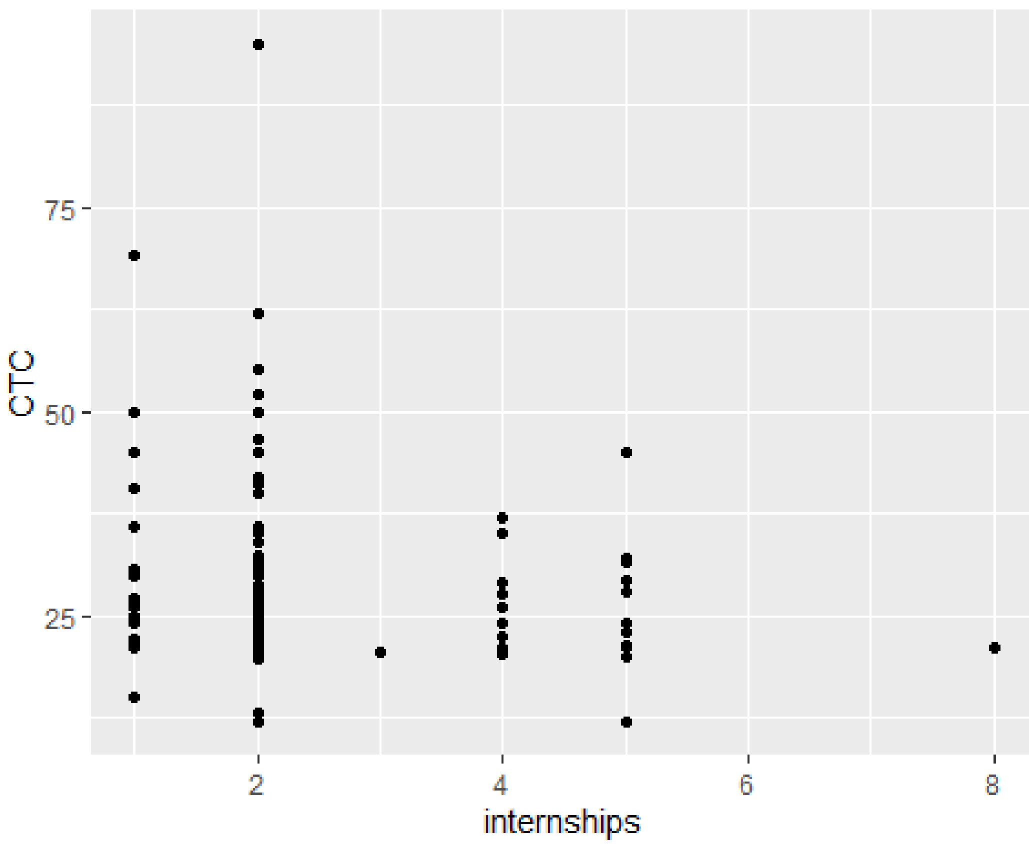
```
##       CTC             CGPA        internships    coding_languages
##  Min.   :12.00   Min.   :5.80   Min.   :1.000   Min.   :0.00    
##  1st Qu.:24.00   1st Qu.:6.70   1st Qu.:2.000   1st Qu.:3.00    
##  Median :27.00   Median :7.22   Median :2.000   Median :3.00    
##  Mean   :30.19   Mean   :7.33   Mean   :2.314   Mean   :3.24    
##  3rd Qu.:34.00   3rd Qu.:8.00   3rd Qu.:2.000   3rd Qu.:4.00    
##  Max.   :95.00   Max.   :9.20   Max.   :8.000   Max.   :9.00    
##                                                                 
##  competitive_coding_proficiency domain_technologies technical_certificatio
ns
##  Min.   :0.000                  Min.   :1.000       Min.   : 1.000        
##  1st Qu.:6.000                  1st Qu.:2.000       1st Qu.: 3.000        
##  Median :6.000                  Median :4.000       Median : 4.000        
##  Mean   :6.331                  Mean   :3.619       Mean   : 4.198        
##  3rd Qu.:7.000                  3rd Qu.:5.000       3rd Qu.: 6.000        
##  Max.   :9.000                  Max.   :8.000       Max.   :10.000        
##                                 NA's   :3                                 
##  non_tech_certifications   hackathons  
##  Min.   :1.000           Min.   :1.00  
##  1st Qu.:1.000           1st Qu.:2.00  
##  Median :2.000           Median :3.00  
##  Mean   :2.451           Mean   :3.06  
##  3rd Qu.:3.000           3rd Qu.:4.00  
##  Max.   :6.000           Max.   :9.00  
##  NA's   :39              NA's   :37    
```

b. Correlation between independent and dependant variables

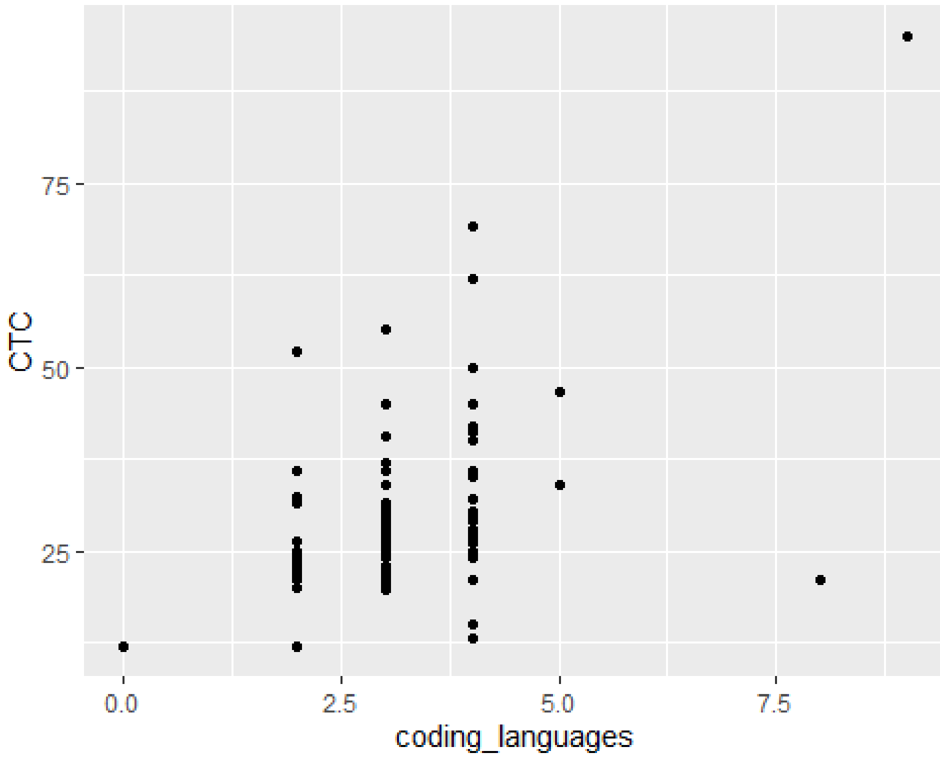
    plot1<-ggplot(data = df, aes(x = internships, y = CTC)) +
  geom_point()
plot2<-ggplot(data = df, aes(x = coding_languages, y = CTC)) +
  geom_point()
```

```
plot3<-ggplot(data = df, aes(x = domain_technologies, y = CTC)) +
  geom_point()
plot4<-ggplot(data = df, aes(x = technical_certifications, y = CTC)) +
  geom_point()
plot5<-ggplot(data = df, aes(x = non_tech_certifications, y = CTC)) +
  geom_point()
plot6<-ggplot(data = df, aes(x = hackathons, y = CTC)) +
  geom_point()
plot7<-ggplot(data = df, aes(x = coding_languages, y = CTC)) +
  geom_point()
plot8<-ggplot(data = df, aes(x = competitive_coding_proficiency, y = CTC)) +
  geom_point()
plot9<-ggplot(data = df, aes(x = CGPA, y = CTC)) +
  geom_point()


plot1
```

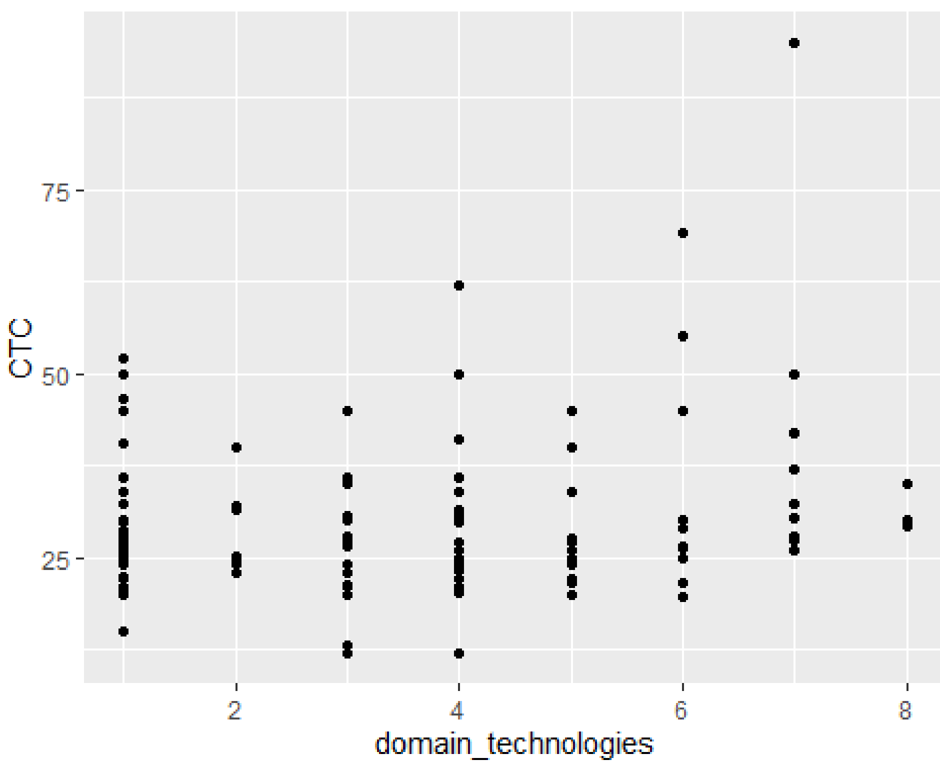



```
plot2
```

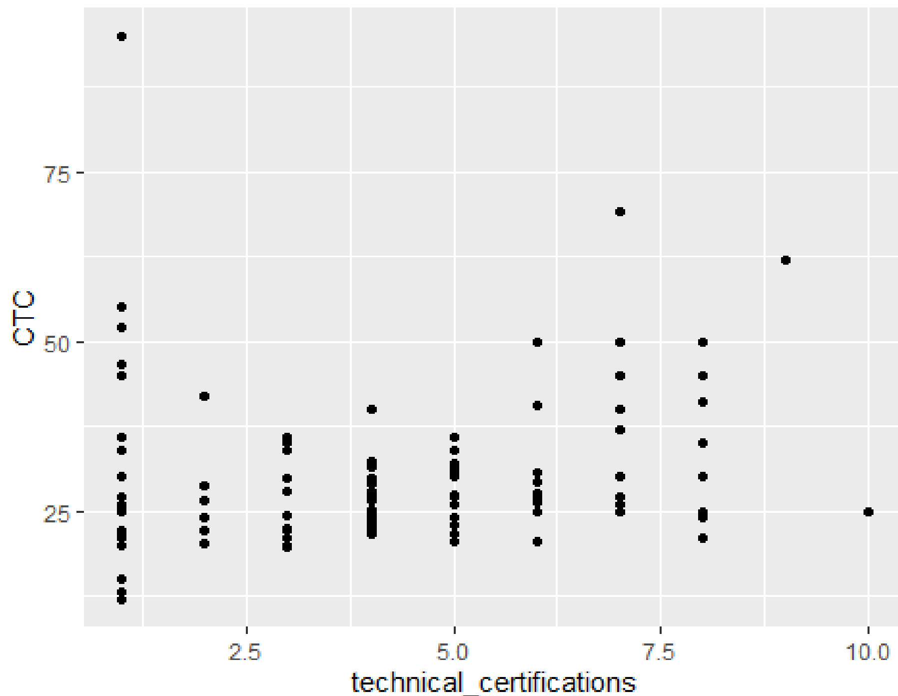



```
plot3

## Warning: Removed 3 rows containing missing values (`geom_point()`).
```

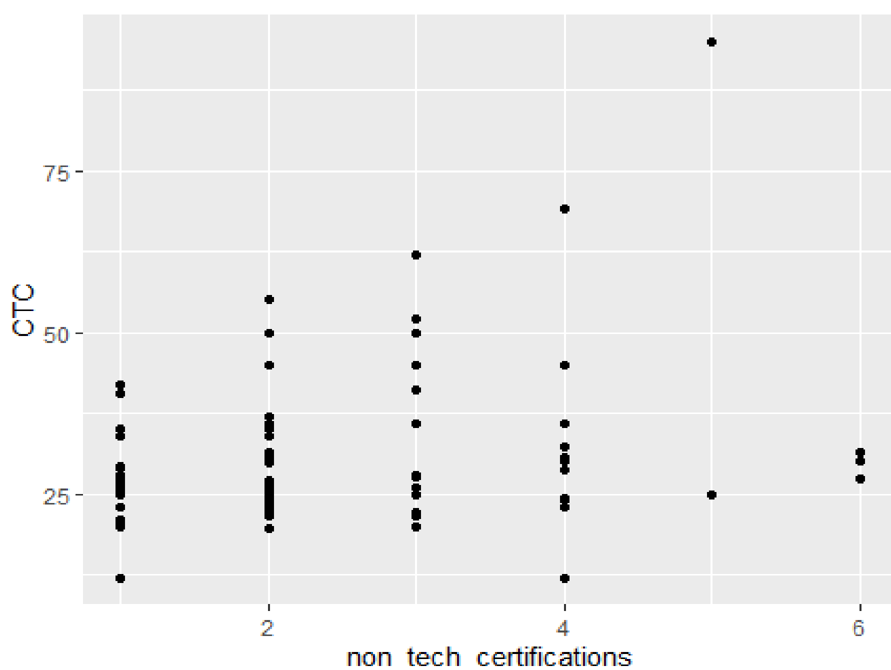


```
plot4
```



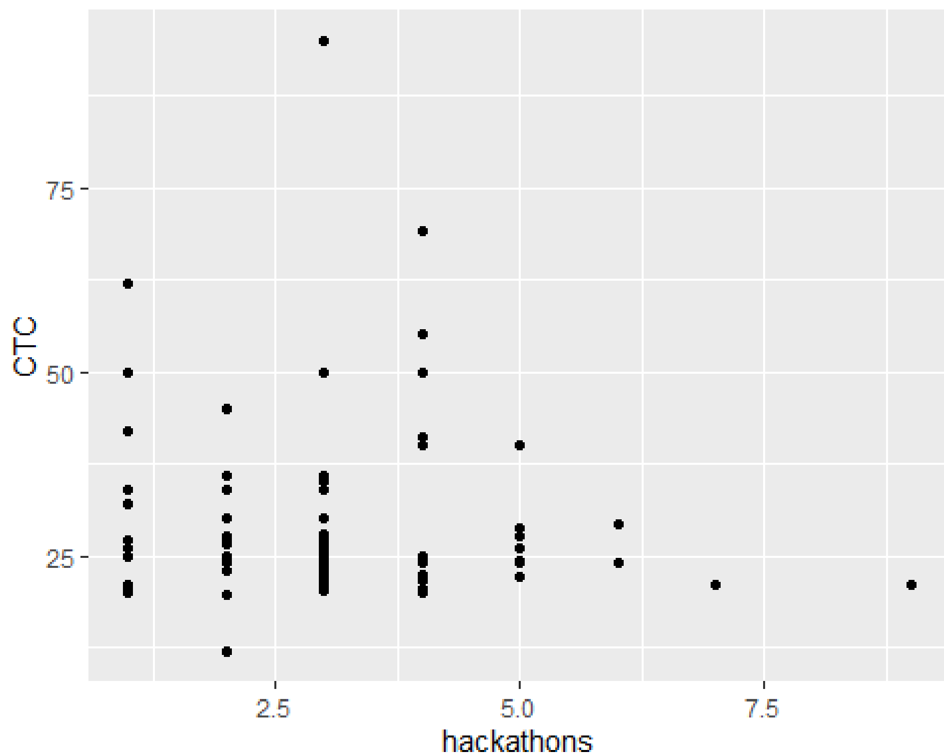

```
plot5

## Warning: Removed 39 rows containing missing values (`geom_point()`).
```

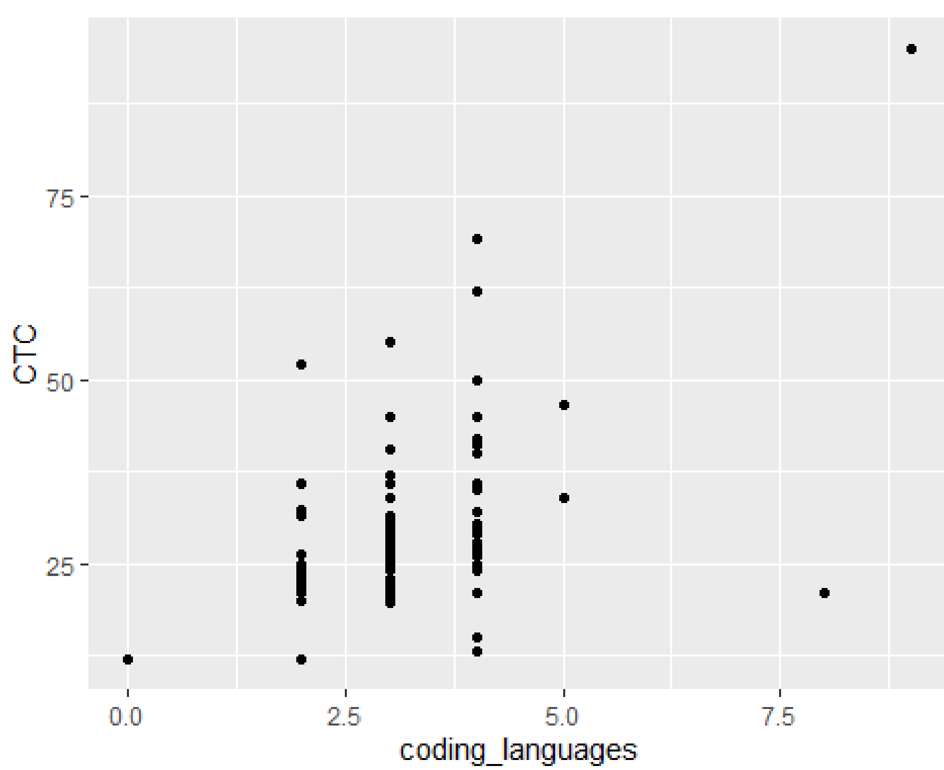



```
plot6

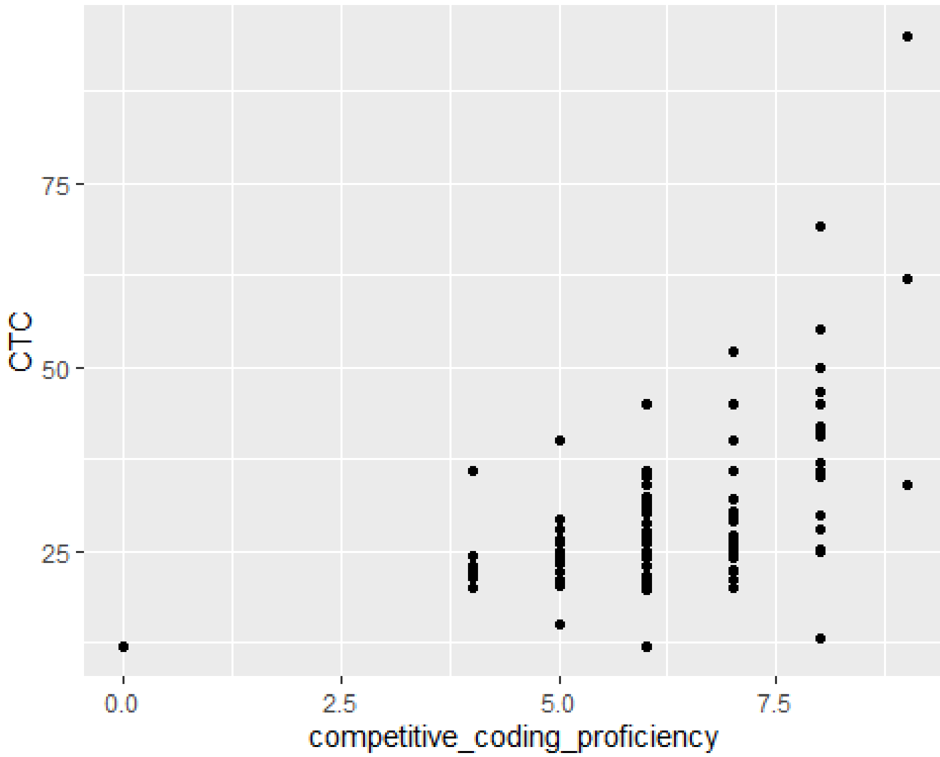
## Warning: Removed 37 rows containing missing values (`geom_point()`).
```


CTC
75
50
25
2.5
5.0
7.5
hackathons

```
plot7
```

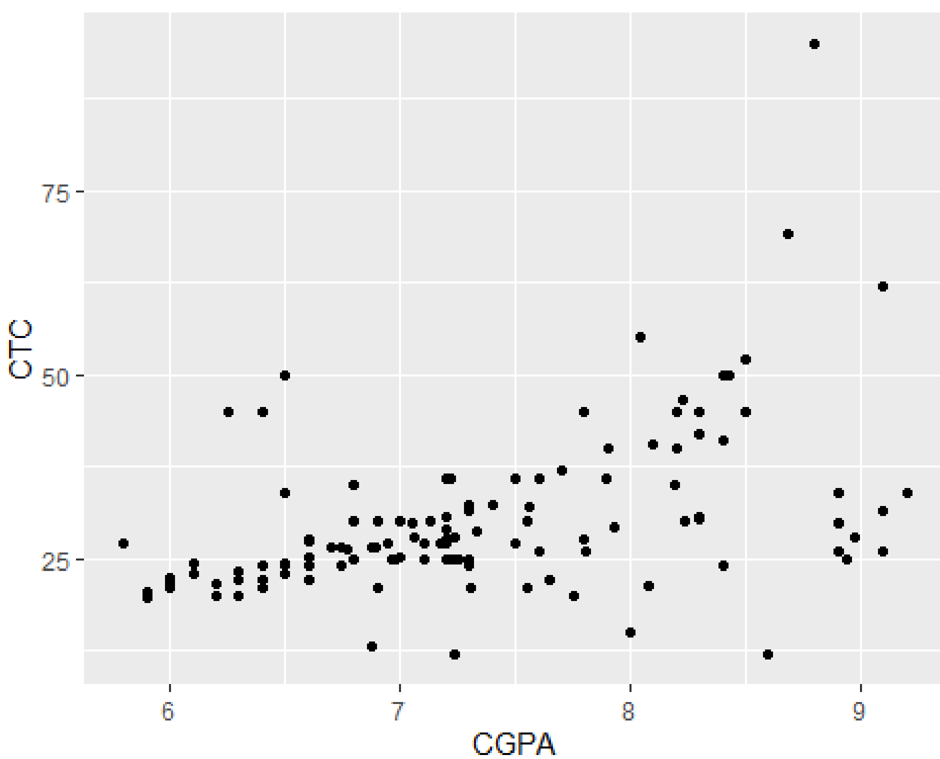

CTC
75
50
25
0.0
2.5
5.0
7.5
coding_languages

```
plot8
```



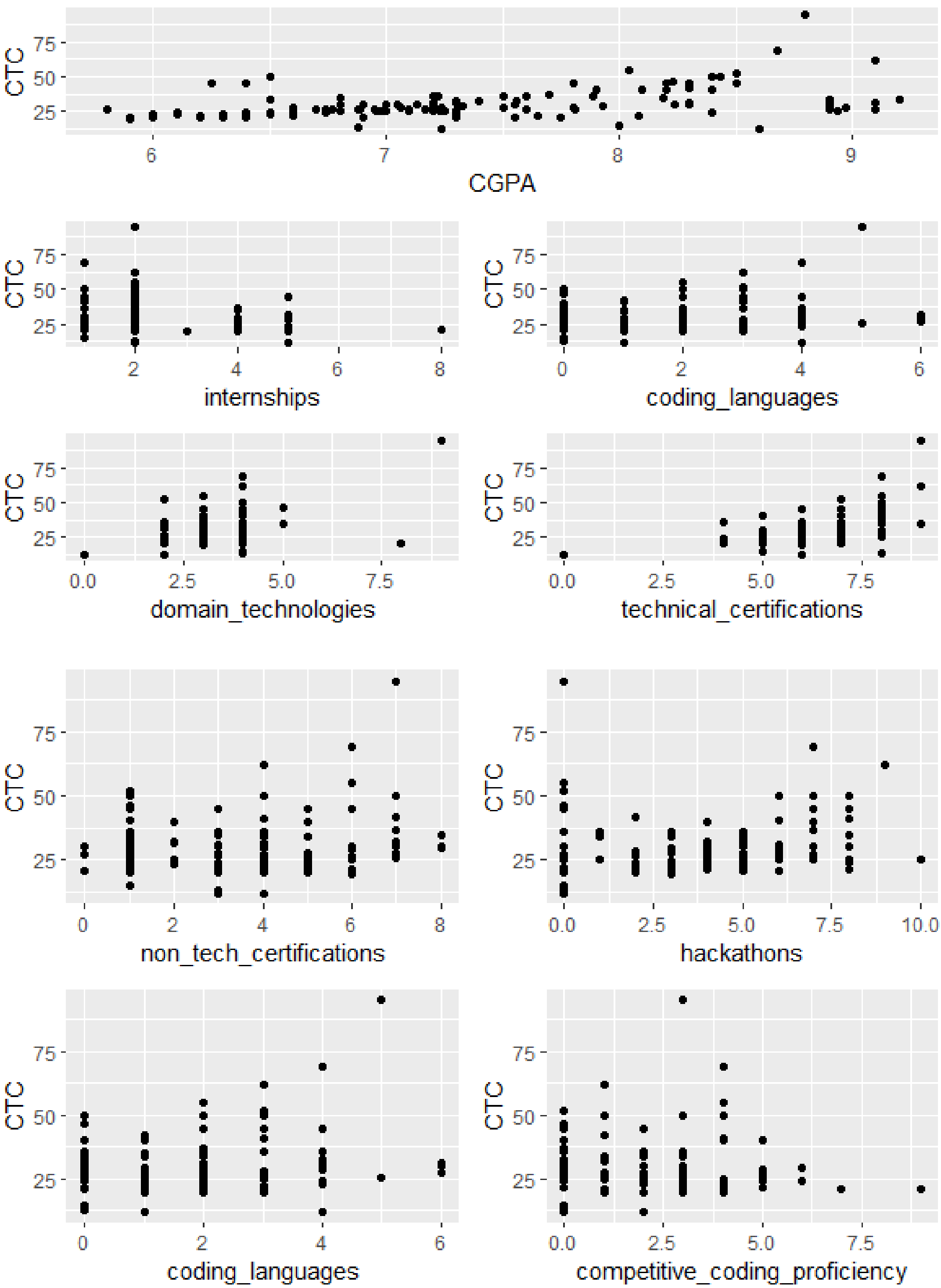
75
50
25
CTC
0.0
2.5
5.0
7.5
competitive_coding_proficiency


```
plot9
```



75
50
25
CTC
6
7
8
9
CGPA

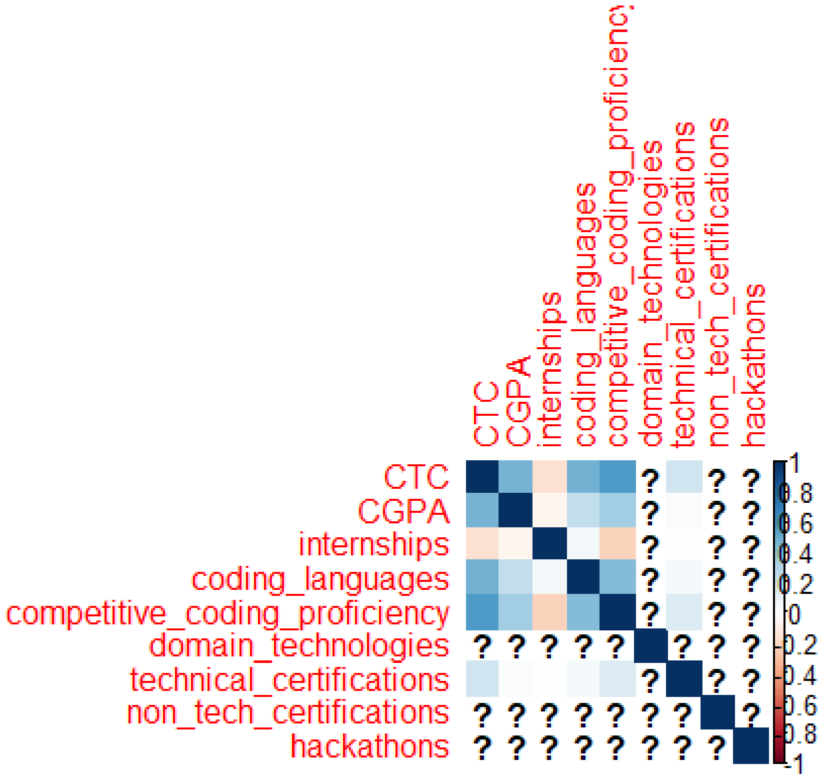
```
# final_plot<-plot9/(plot1|plot2)/(plot3|plot4)
# final_plot1<-(plot5|plot6)/(plot7|plot8)
# view(final_plot)
# view(final_plot1)
```



```r
M<-cor(df)
head(round(M,2))
```

```
##                                 CTC  CGPA internships coding_languages
## CTC                            1.00  0.47       -0.16             0.47
## CGPA                           0.47  1.00       -0.05             0.25
## internships                   -0.16 -0.05        1.00             0.05
## coding_languages               0.47  0.25        0.05             1.00
## competitive_coding_proficiency  0.57  0.34       -0.22             0.44
## domain_technologies              NA    NA          NA               NA
##                                competitive_coding_proficiency
## CTC                                                      0.57
## CGPA                                                     0.34
## internships                                             -0.22
## coding_languages                                         0.44
## competitive_coding_proficiency                           1.00
## domain_technologies                                        NA
##                                domain_technologies technical_certification
s
## CTC                                             NA                     0.2
0
## CGPA                                            NA                     0.0
2
## internships                                     NA                     0.0
1
## coding_languages                                NA                     0.0
5
## competitive_coding_proficiency                  NA                     0.1
6
## domain_technologies                              1                       N
A
##                                non_tech_certifications hackathons
## CTC                                                 NA         NA
## CGPA                                                NA         NA
## internships                                         NA         NA
## coding_languages                                    NA         NA
## competitive_coding_proficiency                      NA         NA
## domain_technologies                                 NA         NA
```

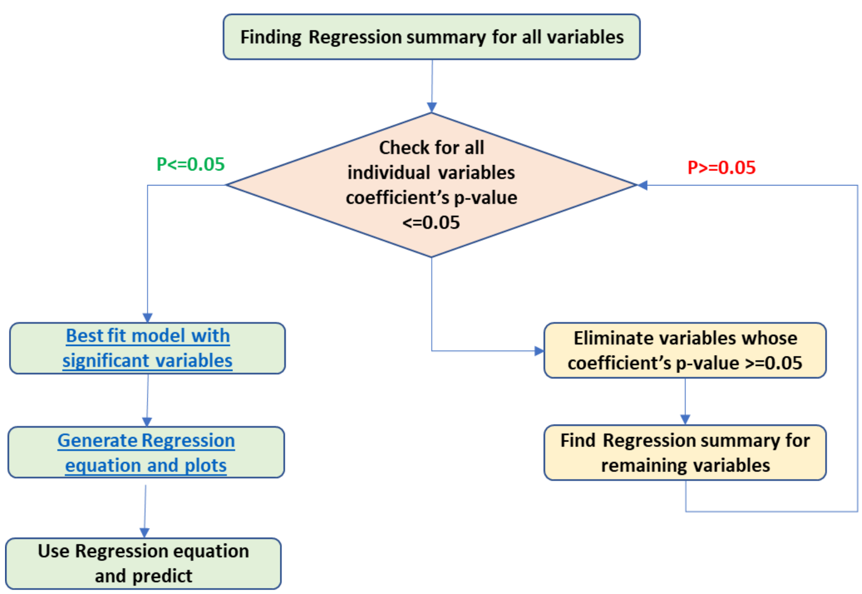
```r
view(M)
corrplot(M, method="color")
```



4. Development

Here, we have mostly automated the entire process of selection of models and independent variables based on backward selection process. This process involves the iterative cycle of removing the unwanted variables based on the constraint of P-value (p>0.05) and repeating until the model contains only the variables with P values less than 0.05.

** Analyzing Complete data Model **



```
# varialbe initialization
data_imported <- df
vars<-names(data_imported)
dependent_var<-combn(vars,1)
dependent_var<-dependent_var[1:dim(dependent_var)[1],1]
dependent_var<-paste(dependent_var,"~")
inpependent_var<-""
vars<-vars[-1]
pvalues<-c()
pvalues1<-c()
dependent_pvars<-c()
satisfied<-TRUE
dependent_pvars1<-c()

# Analysing full datamodel and removing insignificant variables based on back
ward selection with p values < 0.05
for(i in 1:length(vars))
  {
    xx<-combn(vars,i)
    independent_var<-paste("", paste(xx, collapse="+"))
  }

  model_string<- paste(dependent_var,independent_var,sep="")
  model_string

## [1] "CTC ~ CGPA+internships+coding_languages+competitive_coding_proficienc
y+domain_technologies+technical_certifications+non_tech_certifications+hackat
hons"

  lin_mod_1 <- lm(as.formula(model_string), data = data_imported) # Linear Re
gression Model
  summary(lin_mod_1)

##
## Call:
## lm(formula = as.formula(model_string), data = data_imported)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.761  -5.896   0.621   4.504  21.886
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -17.4734     9.8607  -1.772 0.082738 .
## CGPA                           0.2389     1.6482   0.145 0.885349
## internships                   -1.3625     1.1866  -1.148 0.256568
## coding_languages               4.7476     1.2553   3.782 0.000431 ***
## competitive_coding_proficiency 3.9384     1.2384   3.180 0.002579 **
## domain_technologies            0.7775     0.6549   1.187 0.241015
## technical_certifications       0.7027     0.5231   1.343 0.185484
## non_tech_certifications        1.6355     1.1214   1.458 0.151227
```

```
## hackathons                           -0.4289     0.8732  -0.491 0.625550
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.847 on 48 degrees of freedom
##   (64 observations deleted due to missingness)
## Multiple R-squared:  0.6528, Adjusted R-squared:  0.595
## F-statistic: 11.28 on 8 and 48 DF,  p-value: 8.149e-09
```

```
  testing<-summary(lin_mod_1)$coefficients[,4]

  anova_res<-anova(lin_mod_1)
  anova_res
```

```
## Analysis of Variance Table
##
## Response: CTC
##                                Df Sum Sq Mean Sq F value    Pr(>F)
## CGPA                            1 3144.3 3144.30 40.1717 7.617e-08 ***
## internships                     1  141.5  141.50  1.8078  0.185097
## coding_languages                1 2526.8 2526.81 32.2826 7.650e-07 ***
## competitive_coding_proficiency  1  677.3  677.26  8.6527  0.005015 **
## domain_technologies             1  111.6  111.57  1.4254  0.238387
## technical_certifications        1  278.6  278.58  3.5592  0.065275 .
## non_tech_certifications         1  166.4  166.40  2.1259  0.151337
## hackathons                      1   18.9   18.88  0.2412  0.625550
## Residuals                      48 3757.0   78.27
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
  for(i in 1:length(vars))
  {
    pvalue<-anova_res$'Pr(>F)'[i]
    #pvalue<-as.numeric(as.character(testing[i]))
    if(pvalue<0.06)
    {
      pvalues[(length(pvalues) + 1)] <-pvalue
      dependent_pvars[(length(dependent_pvars) + 1)] <-vars[i]

    }
  }
```

Iterating through desired variables returned from the above function and checking for p values and removing insignificant variables to give the best fit model

```
while(satisfied)
{
  for(i in 1:length(dependent_pvars))
  {
    xx<-combn(dependent_pvars,i)
    independent_var<-paste("", paste(xx, collapse="+"))
  }
  model_check_string<-paste(dependent_var,independent_var,sep="")
  lin_mod_2 <- lm(as.formula(model_check_string), data = data_imported) # Lin
ear Regression Model
  testing<-summary(lin_mod_2)$coefficients[,4]
  as.character(testing[3])
  anova_res1<-anova(lin_mod_2)
  anova_res1
  for(i in 1:length(dependent_pvars))
  {
    pvalue1<-anova_res1$'Pr(>F)'[i]
    if(pvalue1<0.06)
    {
      pvalues1[(length(pvalues1) + 1)] <-pvalue1
      dependent_pvars1[(length(dependent_pvars1) + 1)] <-dependent_pvars[i]

    }
  }
  if(length(dependent_pvars) == length(dependent_pvars1))
  {
    satisfied=FALSE
  }
  else
  {
    dependent_pvars<-dependent_pvars1

  }
}

# displaying the final set of p values and independent variables for the best
fit model

df1<-do.call(rbind, Map(data.frame, Dependent_variables=dependent_pvars1,pval
uesss=pvalues1))
df1

## Dependent_variables             pvaluesss
## CGPA                          3.955051e-10
## coding_languages              3.264684e-07
## competitive_coding_proficiency 8.307336e-06
```

Dynamic building of best fit model

```
for(i in 1:length(dependent_pvars1))
{
  xx<-combn(dependent_pvars1,i)
  independent_var<-paste("", paste(xx, collapse="+"))
}

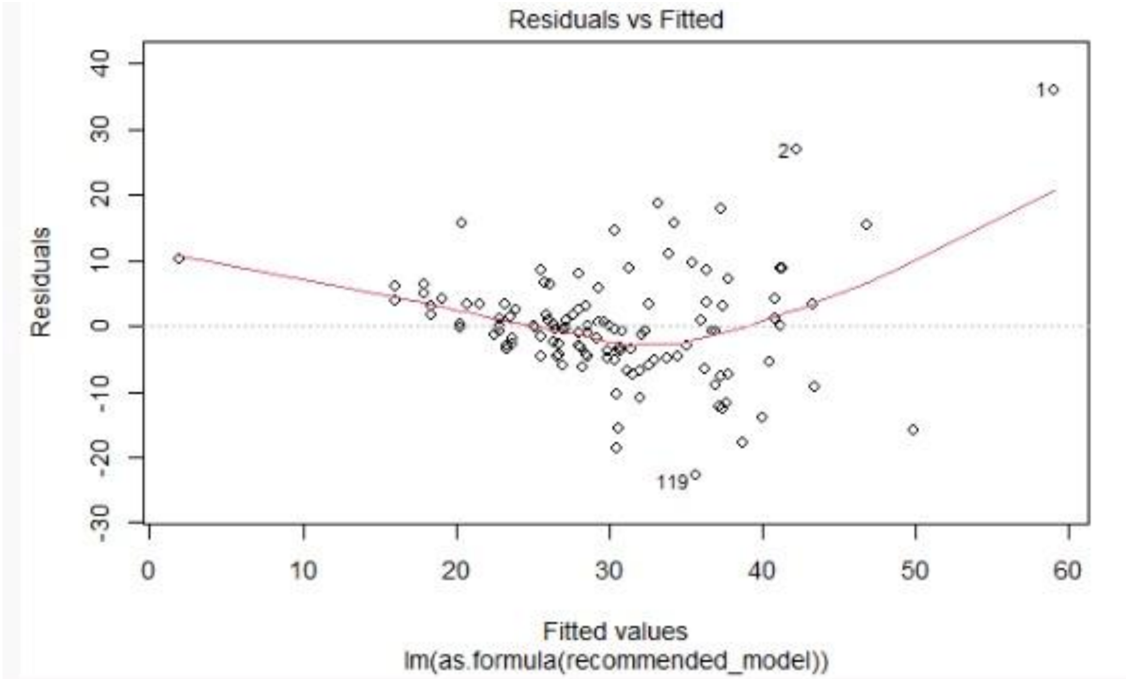

recommended_model<-paste(dependent_var,independent_var,sep="")

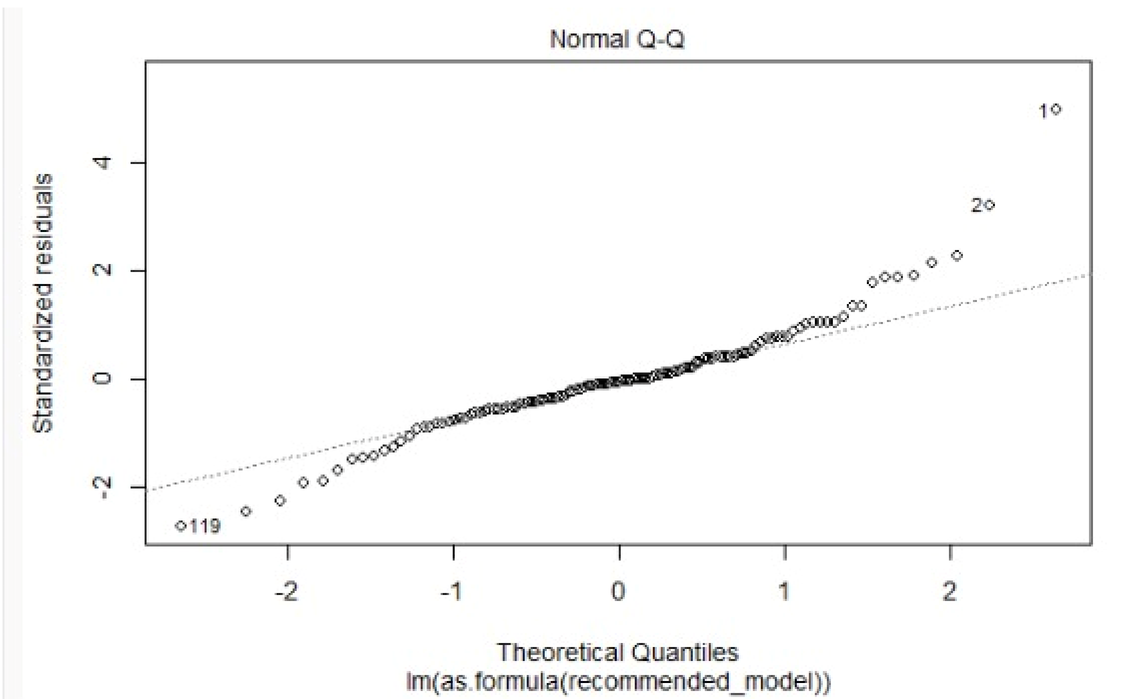
print(paste("The recommended best fit model from the code is : ",recommended_
model))
```

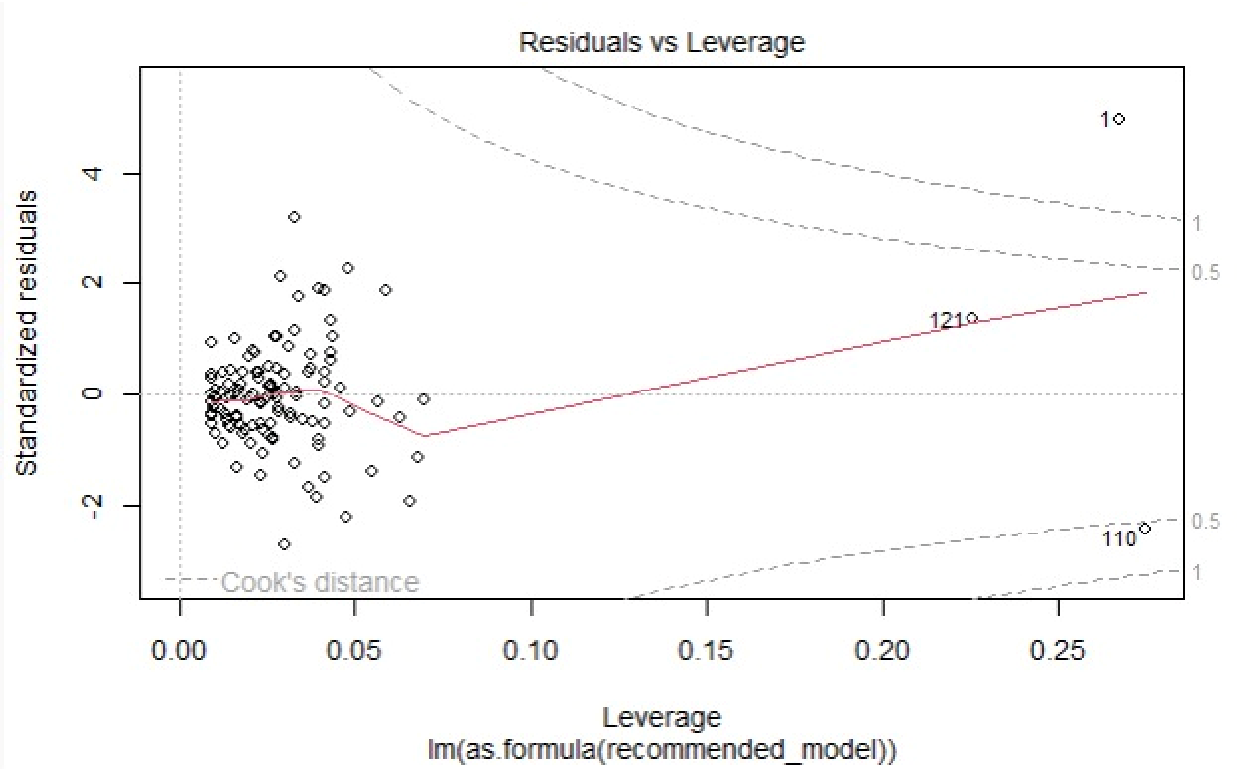
```
## [1] "The recommended best fit model from the code is :  CTC ~ CGPA+coding_
languages+competitive_coding_proficiency"
```

```
best_model<-lm(as.formula(recommended_model),data_imported)
summary(best_model)
```

```
##
## Call:
## lm(formula = as.formula(recommended_model), data = data_imported)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.677  -4.513  -0.402   3.416  36.005
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -24.3996     6.7218  -3.630 0.000422 ***
## CGPA                             3.6361     0.9530   3.815 0.000219 ***
## coding_languages                 2.6566     0.8313   3.196 0.001794 **
## competitive_coding_proficiency   3.0542     0.6549   4.664 8.31e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.467 on 117 degrees of freedom
## Multiple R-squared:  0.4553, Adjusted R-squared:  0.4414
## F-statistic:  32.6 on 3 and 117 DF,  p-value: 2.165e-15
```


Residuals vs Fitted
Residuals
Fitted values
lm(as.formula(recommended_model))
1
2
119


Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(as.formula(recommended_model))
1
2
119



** Generating regression equation for best fit model **

```
regression_eqn<-""

regEq <- function(lmObj, dig) {
  gsub(":", "*",
       paste0(
         names(lmObj$model)[1]," = ",
         paste0(
           c(round(lmObj$coef[1], dig), round(sign(lmObj$coef[-1])*lmObj$coef
[-1], dig)),
           c("", rep("*", length(lmObj$coef)-1)),
           paste0(c("", names(lmObj$coef)[-1]), c(ifelse(sign(lmObj$coef)[-1]
==1," + "," - "), "")),
           collapse=""
         )
       )
  )
}
regression_equation<- regEq(best_model,length(dependent_pvars1))

regression_equation

## [1] "CTC = -24.4 + 3.636*CGPA + 2.657*coding_languages + 3.054*competitive
_coding_proficiency"
```

** Plotting the best fit line **

```
plot1<-ggplot(data = model.frame(best_model), aes(x = CGPA, y = CTC)) +
 geom_point() +
 geom_smooth(aes(y = predict(best_model)))

 #plot2<-ggplot(data = model.frame(best_model), aes(x = internships, y = CTC)
) +
#  geom_point() +
#  geom_smooth(aes(y = predict(best_model)))

plot3<-ggplot(data = model.frame(best_model), aes(x = coding_languages, y = C
TC)) +
  geom_point() +
  geom_smooth(aes(y = predict(best_model)))

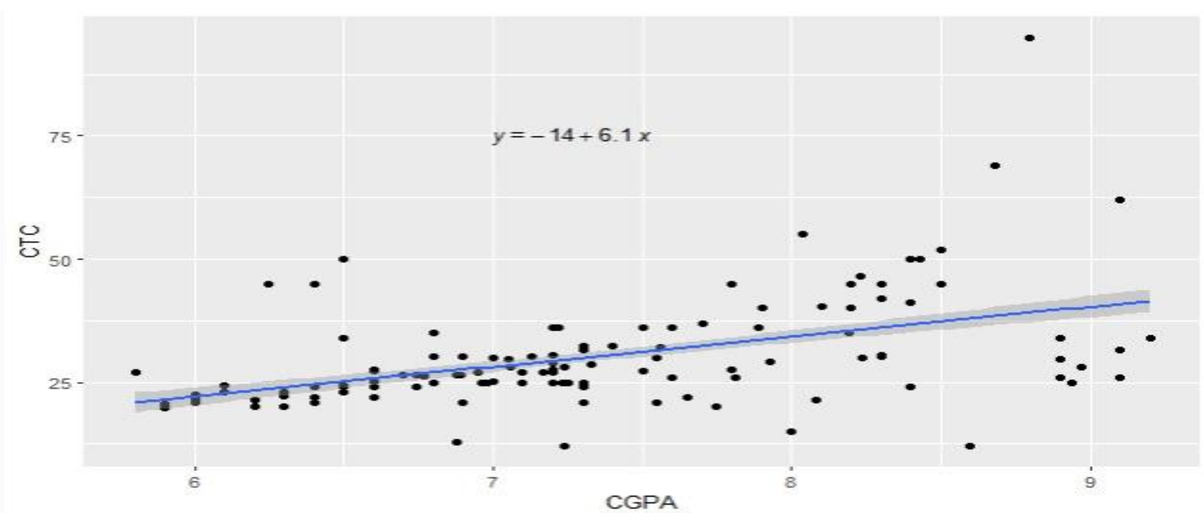
plotsss<-ggplot(data = model.frame(best_model), aes(x = competitive_coding_pr
oficiency, y = CTC)) +
  geom_point() +
  geom_smooth(aes(y = predict(best_model)))

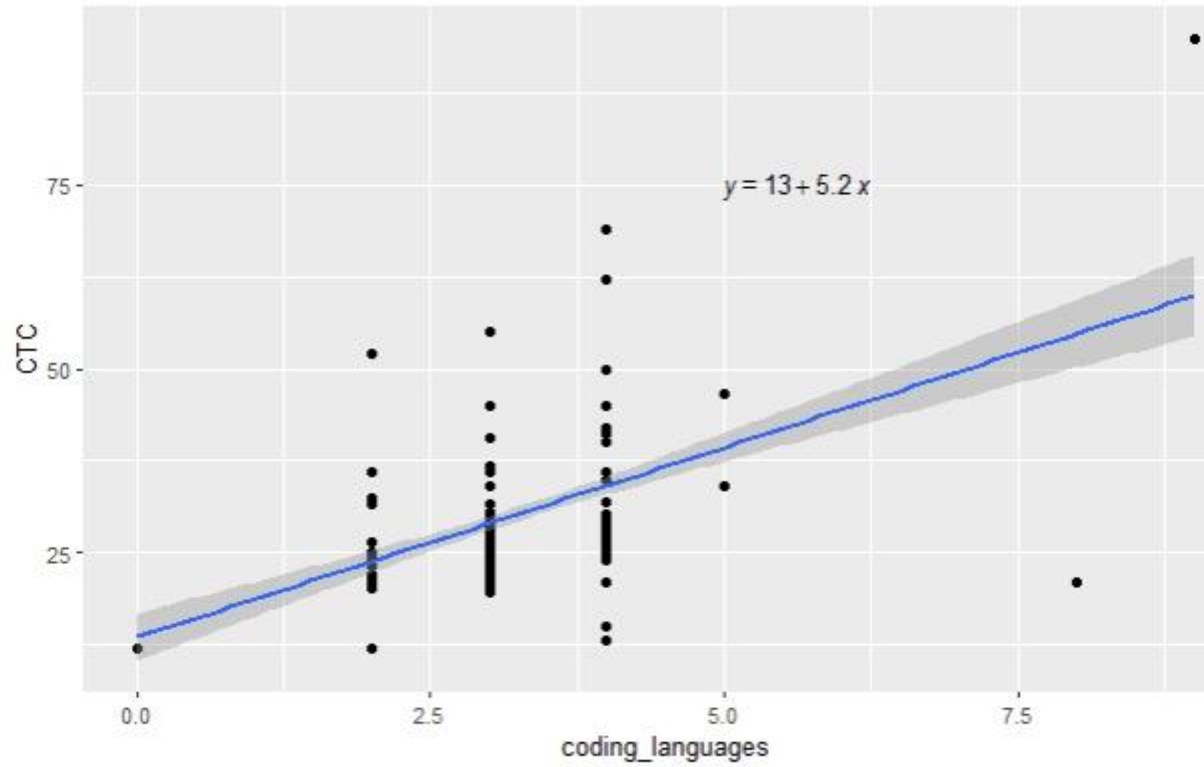
#plot5<-ggplot(data = model.frame(best_model), aes(x = technical_certificatio
ns, y = CTC)) +
  #geom_point() +
  #geom_smooth(aes(y = predict(best_model)))

plot1

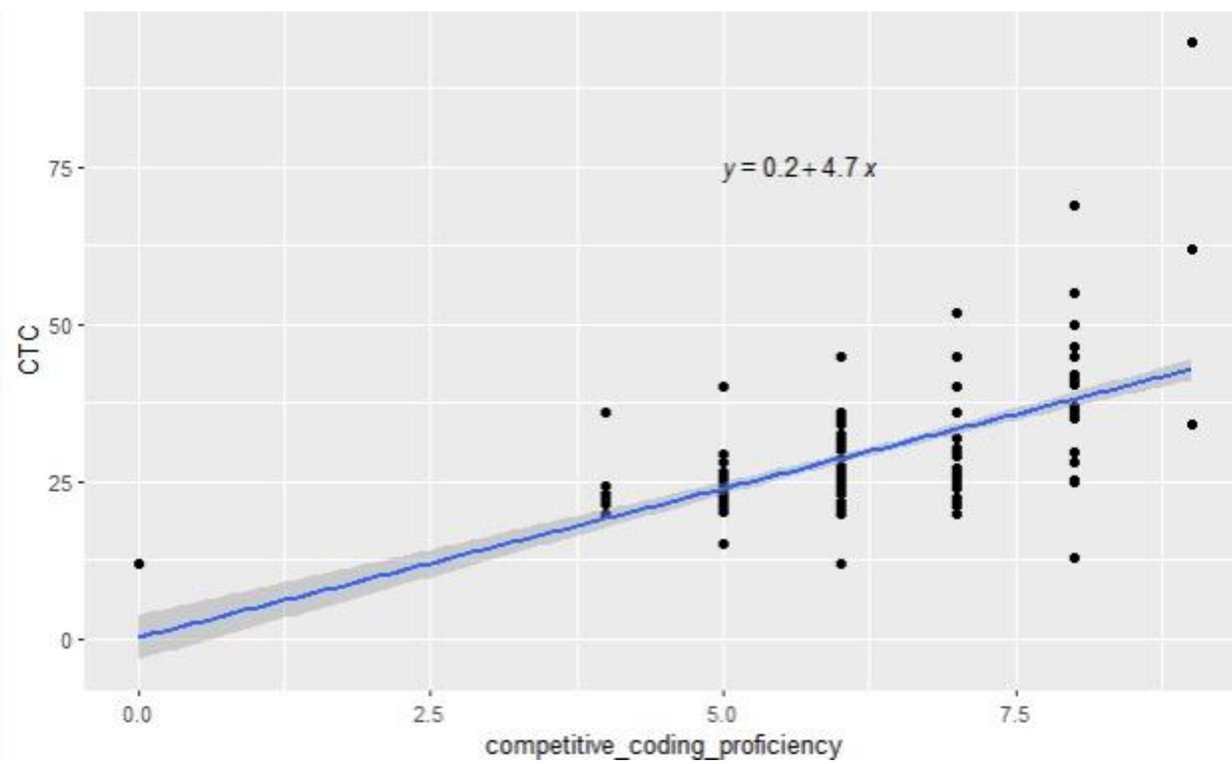
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

Plots -





y = 13 + 5.2 x
CTC
coding_languages



y = 0.2 + 4.7 x
CTC
competitive_coding_proficiency

Insights

CGPA is positively correlated with the best fit line and is a major factor in the prediction of CTC

Coding languages and Coding Proficiency are strongly correlated and plays a major role in prediction of CTC

Prediction

We used 60% of our data to test our model and predict CTC we can find the fitted values and standard error in the result

```
library(prediction)

## Warning: package 'prediction' was built under R version 4.2.2

test_data<-read.csv(test_data_path)
view(test_data)
fitted<-predict(best_model,test_data)
pred <- prediction(best_model, test_data)
pred %>%
  summarise(pred_mean = mean(fitted) ,
            se = mean(se.fitted),
            ci_low = pred_mean - (1.96 * se),
            ci_high = pred_mean + (1.96 * se),
            total_n = n()) %>%
  as.data.frame()

##   pred_mean       se   ci_low  ci_high total_n
## 1  34.21933 1.399234 31.47683 36.96183      60

view(pred)
```

```
Predicted values Table

##    CGPA coding_languages competitive_coding_proficiency   fitted se.fitted
## 1  8.80                9                              9 58.99505 4.3787179
## 2  8.68                4                              8 42.22153 1.5363877
## 3  9.10                4                              9 46.80286 2.0499574
## 4  8.04                3                              8 37.23782 1.4448758
## 5  8.50                2                              7 33.19965 1.8569058
## 6  6.50                4                              8 34.29483 1.6915441
## 7  8.43                4                              8 41.31250 1.4070134
## 8  8.40                4                              8 41.20342 1.3934203
## 9  8.23                5                              8 43.24189 1.6211540
## 10 6.40                4                              8 33.93122 1.7582318
## 11 8.20                3                              8 37.81960 1.4949471
## 12 7.80                3                              8 36.36516 1.3978763
```

```
## 13 8.30                4                              8 40.83981 1.3514950
## 14 6.25                4                              7 30.33164 1.5572237
## 15 8.50                4                              6 35.45869 1.5336453
## 16 8.30                4                              8 40.83981 1.3514950
## 17 8.40                4                              8 41.20342 1.3934203
## 18 8.10                3                              8 37.45599 1.4619907
## 19 7.90                4                              7 36.33120 1.0277151
## 20 8.20                4                              5 31.31369 1.7691303
```

Conclusion

The developed model is significant and can be used to predict but it has less predictive power and standard error was also a bit high. So, by training the model with more and more data and using the subset method to select the significant variables we can increase the efficiency and predictive power of the model. We can also increase R squared value ,reduce Standard Error value and increase F statistic value by using subset method of significant variable selection and by using more data. So there is further development scope in the project.