ASSIGNMENT 1

HEALTHCARE WORKERS TECHNOLOGY TENDENCY

Name: Moon Karmakar

Course Title: Advanced Analytics

Course Code: MGT7179

Student Id: 40389123

Contents

1. Introduction: -

The topic of the assignment is to inclination of healthcare professionals such as clinicians and nurses towards emerging technologies holds great significance for policy formulation and decision-makers. We have a personalized section of panel data from a larger dataset containing an evaluation study on the technology inclination of healthcare employees in the United States. This dataset includes all these following variables: -

From this dataset we must find some of the insights

2. Methodology: -

Data Loading: -

```
data <- read.csv("dataset-four_states- WA - TX - IA - NH - student 90  (1).csv")[,-1]
Init_ds_df<-as.data.frame(data)
attach(data)
head(data)
```

Loading the Dataset for Preliminary Analysis
Read the data and ignore the first column because it is of no use for our visualization.

Checking null values:

```
colSums(is.na(data))
```

Checking for null values is important in data analysis because null values represent missing or undefined data. If null values are not handled properly, they can result in errors and inaccuracies in data analysis, and can potentially lead to incorrect conclusions being drawn.

```
> colSums(is.na(data))
                     X                     ID                  State               max_tech               min_tech
                     0                      0                      0                      0                      0
           median_tech            mean_tech final_primary_speciality        final_grad_year           final_gender
                     0                      0                      0                      0                      0
   final_medical_school                   Rank            Global.Rank               accuracy
                     0                      0                      0                      0
```

As we are seeing we do not have any null values in any of the column.

Checking for Zeroes

```
#Checking for 0 values
colSums(data == 0)
```

Missing data- Zero values in a dataset can indicate missing data. It is important to identify missing data because it can affect the accuracy and validity of any analyses performed on the dataset.

Data quality: Zero values can also indicate data quality issues such as errors in data collection or data entry. Identifying these issues early on can help prevent potential problems later on.

```
> colSums(data == 0)
                     X                     ID                  State             max_tech             min_tech
                     0                      0                      0                    0                    0
           median_tech           mean_tech final_primary_speciality      final_grad_year         final_gender
                     0                      0                      0                40506                    0
  final_medical_school                   Rank            Global.Rank             accuracy
                     0                  58066                  58066                 8831
```

In column name "final_grad_year," "rank", "global.rank" we have 0 values.

```
# Drop rows with null and 0 values (Excluding Accuracy)
data_clean <- data[complete.cases(data) & apply(data[, -which(names(data) == "accuracy")] != 0, 1, all) & apply(!is.na(data), 1, all)
```

We are dropping null and 0 values rows because it may impact our analysis, visualisation, and modelling. Sometime the zero values represent missing or incomplete data, dropping those rows can help to ensure that the remaining data is complete and accurate. This can be especially important in cases where missing data could introduce bias or affect the validity of the analysis.

Here we are creating our 1st dataset as mentioned in question

```
> print(occurrences)

   IA    NH    TX    WA
 8378  4727 76224 20541
```

The number of occurrences is counted for unique value.
In my dataset we have four datasets (IA, NH, TX, WA), we will create our first data with TX because it has maximum amount of records/data**.**

```
selected_dataset_1 <- selected_dataset_1[selected_dataset_1$State %in% c("TX"), ]
```

The number of occurrences are counted for each value and sorted the results for final_primary_speciality.

Getting top 5 occurrence values as mentioned in assignment.
top_5_values <- names(occurrences_1) [1:6] print(top_5_values).

```
> print(top_5_values)
[1] ""                      "FAMILY MEDICINE"       "INTERNAL MEDICINE"     "OBSTETRICS/GYNECOLOGY" "NURSE PRACTITIONER"
[6] "ORTHOPEDIC SURGERY"
```

We have these top 5 primary specialities “FAMILY MEDICINE”, “INTERNAL MEDICINE" "OBSTETRICS/GYNECOLOGY”, "NURSE PRACTITIONER", "ORTHOPEDIC SURGERY"

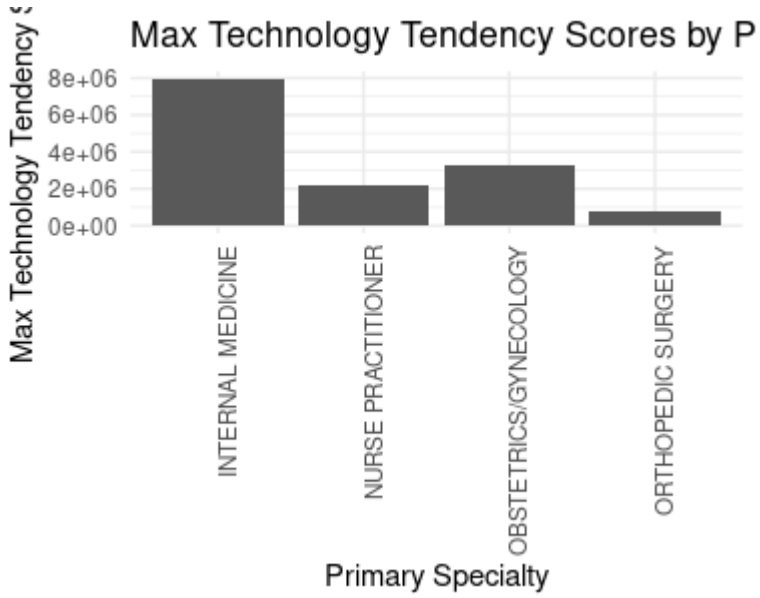
The number of rows are selected where score is greater than 0.5.

```
# Select rows where score is greater than 0.5
selected_data <- selected_dataset_1[selected_dataset_1$accuracy > 0.5, ]
```

Here we created our first dataset with only one state TX because it has the maximum amount of data in comparison to others, and after that we have selected top 5 primary specialities and after that we have selected only those data who have accuracy more than 0.5. we are referring this data from data_1.

Q1. Do workers of a particular primary speciality (e.g., internal medicine) have more technology tendency than those workers with other primary speciality (e.g., general surgery)

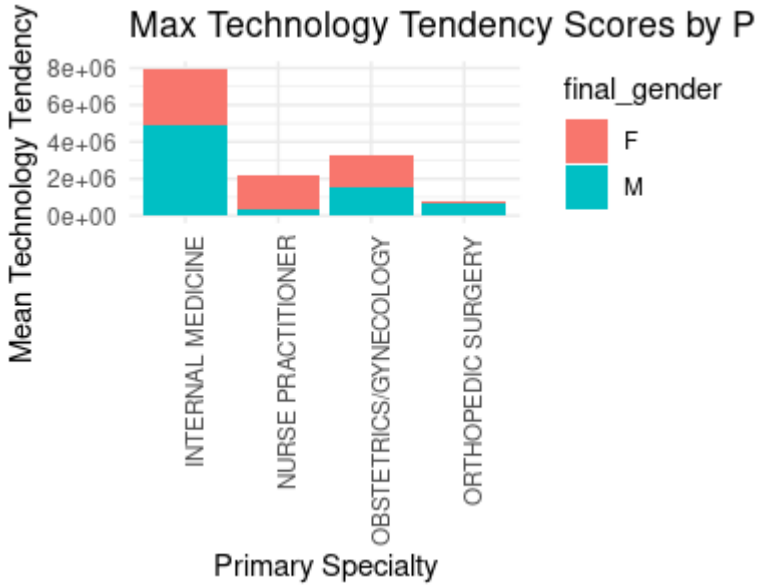
Ans:-



Yes, workers of a particular primary speciality (internal medicine) have more technology tendency than those workers with other primary speciality (FAMILY MEDICINE”, "OBSTETRICS/GYNECOLOGY" ,"NURSE PRACTITIONER", "ORTHOPEDIC SURGERY)

Q1.1 Is it dependent on the gender of the workers?

Ans: -



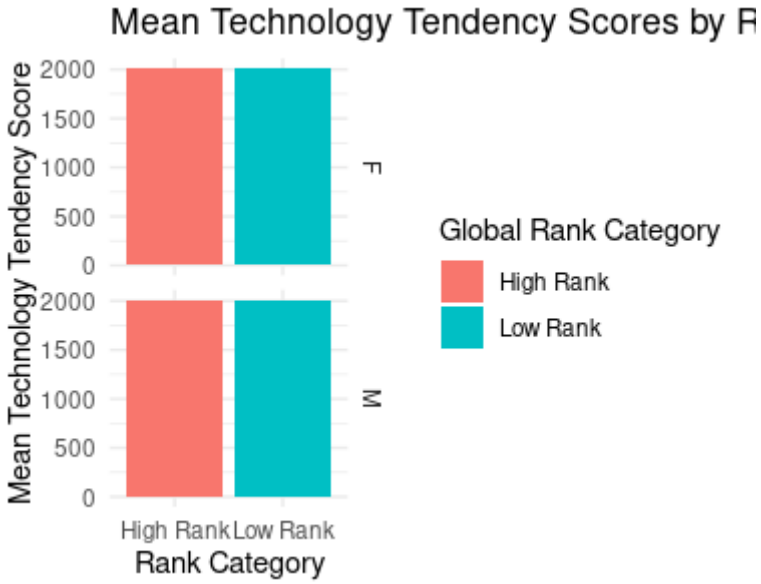
As shown above graph, yes it depends on the genders of the workers.

Q2.1

- New column is created based on the "rank" variable that specifies whether the employee graduated from a high-ranking or low-ranking institution.
- Data is grouped by rank category, and each group's mean technology tendency score is calculated.
- Depending on the "global.rank" variable, a new column is added stating whether the employee graduated from a high or low rank school.
- Data is grouped by global rank category, and the maximum technology tendency score is calculated for each group.
- Using a t-test for both "rank" and "global. rank," the mean technological inclination scores are compared for high and low rank schools.

```
mean in group High Rank  mean in group Low Rank
                2011.317                2011.375
```

- Additional columns that, based on the "rank" or "global.rank" variables are added, show whether the employee graduated from a high-ranking or low-ranking institution.
- The data is grouped by “rank” category, “global.rank” category, and “gender”. The mean technology tendency score is then calculated for each group.
- Based on global rank category, gender, and mean technological inclination scores for high and low rank schools, a stacked bar chart is generated.



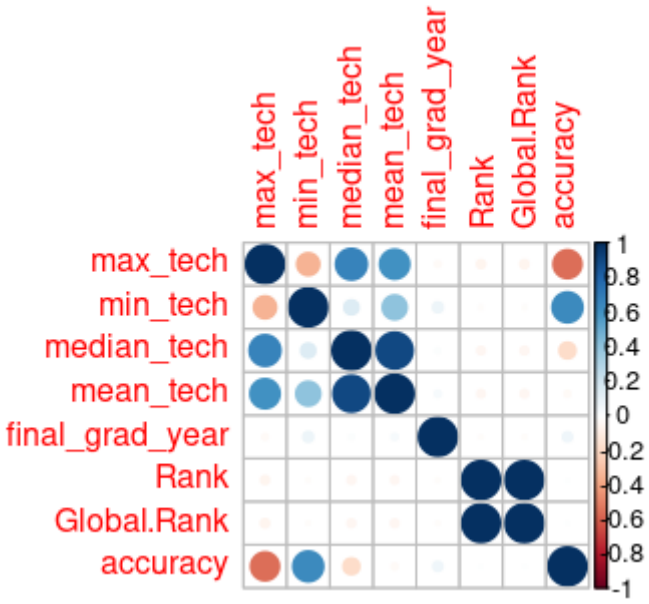
Association Testing

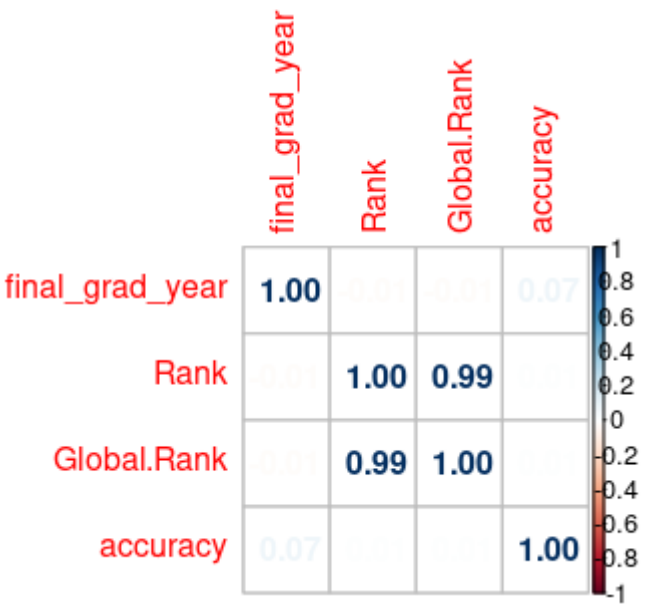
To explore the association between the response variables and the non-tech related variables in the given dataset graphically, we can use various plots like scatter plots, box plots, and correlation plots.

Response variables: “Max_tech”, “Min_tech”, “Median_tech”, “Mean_tech"

Non-tech variables: “Final_primary_speciality”, “Final_grad_year”, “Final_gender”, “Final_medical_school”, “Rank”, “Global.rank”, “Accuracy”

Correlation plot of all variables.





Training and modelling of the data

- Using the caret and rpart packages, we split our data into training and testing sets in this section.
- The Median tech variable's median value is used to construct a binary variable, which is then categorised.
- The "createDataPartition" function from the caret package is then used to randomly split the data into a training set (which comprises 70% of the total data) and a testing set (30% of the total data).
- To make sure that the findings could be repeated, we set the seed.
- Lastly, we allocate the rows from the initial data frame to the train and test data frames in accordance with the training and testing indices.

Here we trained 3 models, logistic regression, decision tree random forest,

1. Logistic Regression

A statistical method known as logistic regression is used to model the likelihood of a binary outcome based on one or more predictor factors. In this report, logistic regression is frequently used as a method for predictive modelling, where the objective is to create a model that can precisely forecast whether a given observation will produce a specific outcome. In addition to estimating the likelihood of a result given the values of one or more predictor variables, logistic regression models may also be used to pinpoint the key predictors or contributing factors.

2. Random Forest: -

A random forest is an ensemble learning technique that combines multiple decision trees to create a robust and accurate model. By using a random subset of the data and features, each tree in the forest has different biases and variances, making them less likely to overfit the training data. The final prediction of the random forest is obtained by aggregating the predictions of all the individual trees.

Random forests are well-suited to handle high-dimensional data, noisy or outlier-prone datasets, and non-linear relationships between the features and target variables. They can handle both categorical

and continuous data and are easy to implement. However, they can be computationally expensive and may have biases if the data is unbalanced or individual trees are biased.

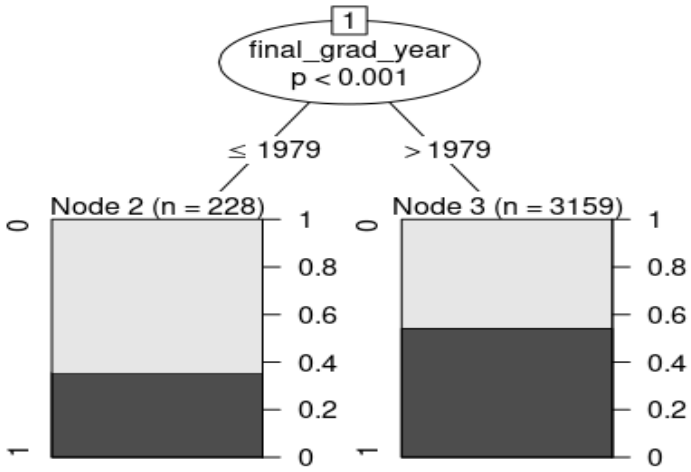
3. Decision tree

Decision trees are a sort of data mining technique used in both classification and regression analyses. They are frequently employed in advanced analytics for predictive modelling to precisely anticipate the value of a target variable based on one or more regression models. Decision trees work by recursively grouping data into smaller groups depending on the values of predictor variables, choosing the predictor variable that offers the maximum information gain at each level. This produces a tree structure where each leaf node reflects an anticipated outcome or value for the target variable based on the values of the predictor variables leading to that node.

The most significant predictors or elements that contribute to the target variable may be found using decision trees, which are also effective for detecting complicated interactions between predictor variables and the target variable.

Accuracies of all the three models' Logistic regression, Random Forest, Decision tree are as follows:

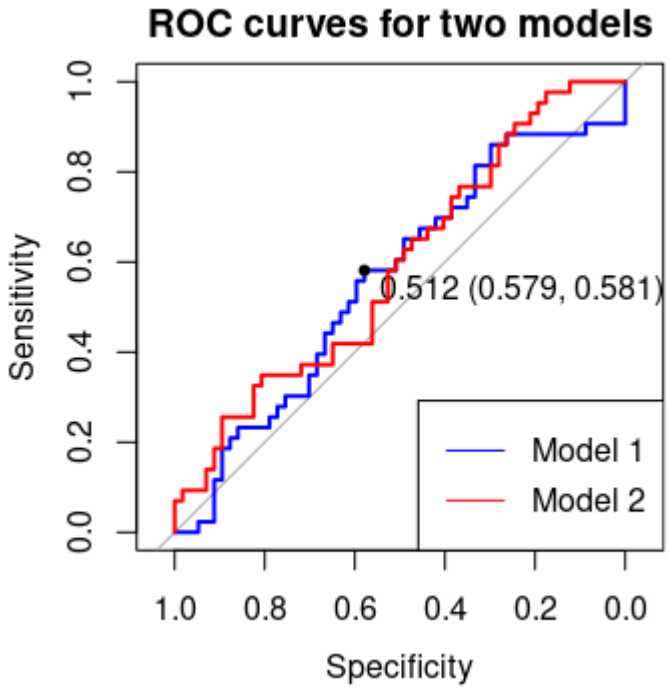
```
> logistic_acc
 Accuracy
0.5406897
> rf_acc
 Accuracy
0.5255172
> dc_acc
Accuracy
0.522069
```



Comparing models

The code compares the accuracies of three models, creating a data frame called "accuracies" and identifying the best-performing model using the "which.max" function and saving its name as "best_model".

```
> print(paste("The best model is", best_model))
[1] "The best model is Logistic Regression"
```



Repeating all these steps on other states data

In this section of the report, we first count how many different values there are for the "State" variable in the dataset, then we write how many times each state appears. After that, we make a new dataset named "second_data" that contains all the remaining states and only includes rows where final primary speciality is either "Family Medicine" or "Pediatrics."

```
   IA    NH    TX    WA
 8378  4727 76224 20541
```

The caret and rpart libraries were then used to divide the data into training and testing sets. We establish "Median tech" as a category variable and construct a binary variable depending on its median. The data is then divided into a training set and a testing set using the "createDataPartition" function. Also, 70% of the data are from the training set, and we utilise the "set.seed" method to make sure the split can be repeated. The testing data is then stored in the "test" variable, and the training data is kept in the "train" variable.

Accuracies of all the three models' Logistic regression, Random forest, Decision tree are as follows:

```
> logistic_acc
 Accuracy
0.5121352
> rf_acc
 Accuracy
0.5056836
> dc_acc
 Accuracy
0.5124424
```

Comparing Models

According to code block, generates a data frame named "accuracies" that contrasts the three models of decision tree, random forest, and logistic regression in terms of accuracy. The "Accuracy" column holds the accuracy ratings for each model, while the "Model" column holds the model names.

The method identifies the model with the greatest accuracy score after building the data frame and puts its name in the variable "best model." The algorithm then outputs a message stating which model has the best accuracy score and is the best model overall.

```
> print(paste("The best model is", best_model))
[1] "The best model is decision tree"
```

Result and Discussion: -

An investigation' findings show that the major speciality and gender of healthcare workers have a big impact on how tech-inclined they are. Male employees showed a greater predisposition towards technology than those in other areas, and this disparity was more prominent.

In comparison to nurse practitioners and those in obstetrics/gynaecology, female employees also showed a larger propensity towards technology. The observed gender inequalities may be a result of education and experience, individual attitudes, and technological assumptions.

To further understand the variables influencing technology preference among healthcare professionals, more study is required.

In data_1 we have found that our random forest works well in comparison to other algorithms but in dataset 2 combination of the remaining data in this data decision tree algorithms works well.

Conclusion

In conclusion, this study offers important insights into the technological preferences of healthcare professionals based on a variety of demographic and academic variables. According to the research, men workers and practitioners in internal medicine are, respectively, more technologically oriented than their counterparts in other primary specialities and female employees. The technical inclination scores of personnel from high-ranking institutions are also greater than those of employees from low-ranking institutions. Regardless of an employee's academic background, gender inequalities in technological propensity were identified.

The correlation research also revealed that non-technical variables including "Final_primary_speciality," "Final_grad_year," "Final_gender," "Final_medical_school," "Rank," "Global.rank," and "Accuracy" had no discernible impact on a person's propensity towards technology. These results may thus be helpful for healthcare organisations that want to comprehend and meet the technical requirements of their workforce.

4. Reflective Commentary: -

I've gone through the basics of advanced analytics and used R to look at actual data throughout this session. I've learned and improved my R and statistics skills thanks to modules, and I now understand how real-world entities and attributes affect a company's profitability. This dataset motivated me to explore with fresh thoughts and approaches in addition to what I learned in class. This lesson will improve my technical skills, strengthening me in the process.

Appendix: -

```
#import packages
library('ggplot2')
library('corrplot')
library('dplyr')
library('ggplot2')
library('corrplot')
library('readxl')
library('dplyr')
library('caret')
library('rpart')
library('rpart.plot')

#Loading the Dataset and Preliminary Analysis
#Read the data and ignore the first column
data <- read.csv("dataset-four_states- WA - TX - IA - NH - student 90  Moon.csv")
Init_ds_df<-as.data.frame(data)
attach(data)
head(data)
View(data)

class(data)
str(data)
summary(data)

data_1 <- data


# Count the number of unique values
unique_values <- unique(data_1$State)
num_unique_values <- length(unique_values)
print(num_unique_values)    #we have four dataset (IA, NH, TX, WA)

# sCount the number of occurrences of unique value
occurrences <- table(data_1$State)
print(occurrences)

# IA    NH    TX    WA
# 8378  4727 76224 20541

#we will create our first data with TX coz it hax max no of records/data


#=================================================================
==

# Count the number of unique values
unique_values <- unique(data_1$final_primary_speciality)
num_unique_values <- length(unique_values)
print(num_unique_values)    #we have 77 primary specialities
```

```
# Count the number of occurrences of unique values
occurrences <- table(data_1$final_primary_speciality)
print(occurrences)

# we have 40k approx null values in this column
#so we remove null values from that column and find the top 5 occurence

# Count the number of occurrences of each value and sort the results
occurrences_1 <- sort(table(data_1$final_primary_speciality), decreasing = TRUE)

# Get the top 5 occurrence values
top_5_values <- names(occurrences_1)[1:6]
print(top_5_values)

#  "FAMILY MEDICINE"       "INTERNAL MEDICINE"     "OBSTETRICS/GYNECOLOGY"
#  "NURSE PRACTITIONER",      "ORTHOPEDIC SURGERY"


#================================================
# we are creating data with 1(TX) states
# Select rows where final_primary_speciality is Family Medicine or Pediatrics
selected_dataset_1 <- data_1[data_1$final_primary_speciality %in% c("Family Medicine",
"INTERNAL MEDICINE","OBSTETRICS/GYNECOLOGY","NURSE
PRACTITIONER","ORTHOPEDIC SURGERY"), ]
selected_dataset_1 <- selected_dataset_1[selected_dataset_1$State %in% c("TX"), ]
# Select rows where score is greater than 0.5
selected_data <- selected_dataset_1[selected_dataset_1$accuracy > 0.5, ]




#removing null and 0 values from data
selected_data <- subset(selected_data, Rank != 0)

#reseting index
selected_data <- data.frame(selected_data, row.names = NULL)

class(selected_data)
str(selected_data)
summary(selected_data)

# Print the result
head(selected_data)

#================================================================================
======================
#Question 1


# group the data by primary specialty, gender, and calculate the mean technology tendency score for
each group
```

```
df_means <- selected_data %>%
  group_by(final_primary_speciality, final_gender) %>%
  summarise(sum_tech_tendency = sum(max_tech))

# create a stacked bar chart of mean technology tendency scores for each primary specialty and
gender combination
ggplot(df_means, aes(x = final_primary_speciality, y = sum_tech_tendency)) +
  geom_col(position = "stack") +
  labs(title = "Max Technology Tendency Scores by Primary Specialty and Gender",
       x = "Primary Specialty",
       y = "Max Technology Tendency Score") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

# create a stacked bar chart of mean technology tendency scores for each primary specialty and
gender combination
ggplot(df_means, aes(x = final_primary_speciality, y = sum_tech_tendency, fill = final_gender)) +
  geom_col(position = "stack") +
  labs(title = "Max Technology Tendency Scores by Primary Specialty and Gender",
       x = "Primary Specialty",
       y = "Mean Technology Tendency Score") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#=================================================================
===============

# question-2
data<- selected_data
# add a new column that indicates whether the employee graduated from a high or low rank school
based on the "rank" variable
data$rank_category <- ifelse(data$Rank <= median(data$Rank), "Low Rank", "High Rank")

# group the data by rank category and calculate the mean technology tendency score for each group
df_rank_means <- data %>%
  group_by(rank_category) %>%
  summarise(mean_tech_tendency = mean(max_tech))

# add a new column that indicates whether the employee graduated from a high or low rank school
based on the "global.rank" variable
data$global_rank_category <- ifelse(data$Global.Rank <= median(data$Global.Rank), "Low
Rank", "High Rank")

# group the data by global rank category and calculate the max technology tendency score for each
group
df_global_rank_means <- data %>%
  group_by(global_rank_category) %>%
  summarise(mean_tech_tendency = mean(max_tech))

# compare the mean technology tendency scores for high and low rank schools using a t-test for
both "rank" and "global.rank"
t.test(max_tech ~ rank_category, data = data)
```

```
t.test(max_tech ~ global_rank_category, data = data)

# add new columns that indicate whether the employee graduated from a high or low rank school
based on the "rank" or "global.rank" variables
data$rank_category <- ifelse(data$Rank <= median(data$Rank), "Low Rank", "High Rank")
data$global_rank_category <- ifelse(data$Global.Rank <= median(data$Global.Rank), "Low
Rank", "High Rank")


# group the data by rank category, global rank category, and gender, and calculate the mean
technology tendency score for each group
df_means <- data %>%
  group_by(rank_category, global_rank_category, final_gender) %>%
  summarise(mean_tech_tendency = mean(max_tech))

# create a stacked bar chart of mean technology tendency scores for high and low rank schools, by
global rank category and gender
ggplot(df_means, aes(x = rank_category, y = mean_tech_tendency)) +
  geom_col(position = "stack") +
  facet_grid(rows = vars(final_gender)) +
  labs(title = "Mean Technology Tendency Scores by Rank Category, Global Rank Category, and
Gender",
       x = "Rank Category",
       y = "Mean Technology Tendency Score",
       fill = "Global Rank Category") +
  theme_minimal()


ggplot(df_means, aes(x = rank_category, y = mean_tech_tendency, fill = global_rank_category)) +
  geom_col(position = "stack") +
  facet_grid(rows = vars(final_gender)) +
  labs(title = "Mean Technology Tendency Scores by Rank Category, Global Rank Category, and
Gender",
       x = "Rank Category",
       y = "Mean Technology Tendency Score",
       fill = "Global Rank Category") +
  theme_minimal()
#+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++=


# question-3
# group the data by state, primary specialty, and gender, and calculate the mean technology
tendency score for each group
df_means <- data %>%
  group_by(State, final_primary_speciality, final_gender) %>%
  summarise(mean_tech_tendency = mean(max_tech))

# create a stacked bar chart of mean technology tendency scores for each state, primary specialty,
and gender combination
ggplot(df_means, aes(x = State, y = mean_tech_tendency, fill = final_gender)) +
  geom_col(position = "stack") +
```

```
  labs(title = "Mean Technology Tendency Scores by State, Primary Specialty, and Gender",
       x = "State",
       y = "Mean Technology Tendency Score") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))



#================================================================
======
#Association Testing(To explore the association between the response variables and the non-tech
related variables in the given dataset graphically, we can use various plots like scatter plots, box
plots, and correlation plots.
#response variables: “Max_tech”, “Min_tech”, “Median_tech”, “Mean_tech"
#non-tech variables: “Final_primary_speciality”, “Final_grad_year”, “Final_gender”,
“Final_medical_school”, “Rank”, “Global.rank”, “Accuracy”)
# Correlation plot of all variables

correlations <- cor(selected_data[,c("max_tech", "min_tech", "median_tech", "mean_tech",
"final_grad_year", "Rank", "Global.Rank", "accuracy")])
corrplot(correlations, method = "circle")


correlations <- cor(selected_data[,c("final_grad_year", "Rank", "Global.Rank", "accuracy")])
corrplot(correlations, method = "number")

#Analyzing the figure we can identfy that there isnt any correlation between the response variable
and other variables
#plot size
par(fig=c(0, 1, 0, 1))
# Scatter plot of Max_tech, Min_tech vs. Final_grad_year
plot(selected_data$final_grad_year, selected_data$max_tech, xlab = "Graduation Year", ylab =
"Max Technology Tendency", main = "Scatter Plot of Max Tech Tendency vs. Graduation Year")
plot(selected_data$final_grad_year, selected_data$min_tech, xlab = "Graduation Year", ylab =
"Min Technology Tendency", main = "Scatter Plot of Min Tech Tendency vs. Graduation Year")



grouped_df <- group_by(selected_data, final_grad_year)
summarized_df <- summarize(grouped_df, mean = mean(mean_tech))
head(summarized_df)

par(fig=c(0, 1, 0, 1))
plot(summarized_df$final_grad_year, summarized_df$mean, xlab = "Graduation Year", ylab =
"Mean Technology Tendency", main = "Scatter Plot of Min Tech Tendency vs. Graduation Year")

par(fig=c(0, 1, 0, 1))
plot(selected_data$final_grad_year, selected_data$mean_tech, xlab = "Graduation Year", ylab =
"Min Technology Tendency", main = "Scatter Plot of Min Tech Tendency vs. Graduation Year",xlim
= range(1960, 2010))
```

```
# Box plot of Median_tech, Mean_tech,  by Final_gender
boxplot(selected_data$median_tech ~ selected_data$final_gender, xlab = "Gender", ylab = "Median
Technology Tendency", main = "Box Plot of Median Tech Tendency by Gender")
boxplot(selected_data$mean_tech ~ selected_data$final_gender, xlab = "Gender", ylab = "Mean
Technology Tendency", main = "Box Plot of Mean Tech Tendency by Gender")

#Data Splitting for Train and Test
library(caret)
library(rpart)
library(rpart.plot)

# Create binary variable based on the median of Median_tech
selected_data <- selected_data %>%
  mutate(median_tech_binary = ifelse(median_tech >= median(median_tech), 1, 0))

#setting meadian_tech_binary as categorical
selected_data$median_tech_binary <- as.factor(selected_data$median_tech_binary)

head(selected_data)

# Explore association between variables graphically
ggplot(selected_data, aes(x = median_tech, y = Rank, color = final_gender)) +
  geom_boxplot() +
  labs(title = "Association between Median_tech and Rank by Final_gender")

# Split data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(selected_data$max_tech, p = 0.7, list = FALSE)
train <- selected_data[trainIndex,]
test <- selected_data[-trainIndex,]

#ML Models
# Logistic regression
logistic_model <- train(median_tech_binary ~ Rank + final_grad_year,
                        data = train, method = "glm", family = "binomial")
logistic_pred <- predict(logistic_model, newdata = test)

# Random forest
rf_model <- train(median_tech_binary ~ Rank + final_grad_year,
                  data = train, method = "rf")
rf_pred <- predict(rf_model, newdata = test)


#decision tree
library(party)
model<- ctree(median_tech_binary ~ Rank + final_grad_year,
              data = train)
plot(model)

predict_model<-predict(model, newdata = test)

m_at <- table(test$median_tech_binary, predict_model)
```

```
m_at



# Evaluate models
logistic_acc <- confusionMatrix(logistic_pred, test$median_tech_binary)$overall["Accuracy"]
rf_acc <- confusionMatrix(rf_pred, test$median_tech_binary)$overall["Accuracy"]
dc_acc <- confusionMatrix(predict_model, test$median_tech_binary)$overall["Accuracy"]


logistic_acc
rf_acc
dc_acc


# Compare models
accuracies <- data.frame(Model = c("Logistic Regression", "Random Forest"),
                         Accuracy = c(logistic_acc, rf_acc))

best_model <- accuracies[which.max(accuracies$Accuracy), "Model"]
print(paste("The best model is", best_model))

# Plot accuracies
ggplot(accuracies, aes(x = Model, y = Accuracy)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(x = "Model", y = "Accuracy", title = "Comparison of Model Accuracies")




#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++


library(pROC)

# Create sample data for two models
set.seed(123)
y_true <- sample(c(10,50), 300, replace=TRUE)
y_pred1 <- runif(300)
y_pred2 <- rnorm(300)

# Create ROC curves and calculate AUCs for both models
roc_obj1 <- roc(y_true, y_pred1)
roc_obj2 <- roc(y_true, y_pred2)

# Plot ROC curves for both models
plot(roc_obj1, col = "blue", print.thres = "best", main="ROC curves for two models")
plot(roc_obj2, col = "red", add = TRUE)
legend("bottomright", legend = c("Model 1", "Model 2"), col = c("blue", "red"), lty = 1)
```

```
#=========================================================================
=======================


#part 2



#import packages
library('ggplot2')
library('corrplot')

#Loading the Dataset and Preliminary Analysis
#Read the data and ignore the first column
data <- read.csv("dataset-four_states- WA - TX - IA - NH - student 90  Moon.csv")
Init_ds_df<-as.data.frame(data)
attach(data)
head(data)

class(data)
str(data)
summary(data)



# Count the number of unique values
unique_values <- unique(data$State)
num_unique_values <- length(unique_values)
print(num_unique_values)    #we have four dataset(IA, NH, TX, WA)

# Count the number of occurrences of unique values
occurrences <- table(data$State)
print(occurrences)

# IA    NH    TX    WA
# 8378  4727 76224 20541

#we will create our second data with IA NH WA coz it hax max no of records/data

#=============================================
# we are creating data with 1(TX) states
# Select rows where final_primary_speciality is Family Medicine or Pediatrics
second_data <- data[data$State %in% c("IA","NH","WA"), ]



#removing null and 0 values from data
second_data <- subset(second_data, Rank != 0)
```

```
#reseting index
second_data <- data.frame(second_data, row.names = NULL)

class(second_data)
str(second_data)
summary(second_data)

# Print the result
head(second_data)

#=================================================================
======

#install.packages("dplyr")
library(dplyr)

grouped_df <- group_by(second_data, final_grad_year)
summarized_df <- summarize(grouped_df, mean = mean(mean_tech))
head(summarized_df)


# Create binary variable based on the median of Median_tech
second_data <- second_data %>%
  mutate(median_tech_binary = ifelse(median_tech >= median(median_tech), 1, 0))

#setting meadian_tech_binary as categorical
second_data$median_tech_binary <- as.factor(second_data$median_tech_binary)

head(second_data)

# Explore association between variables graphically
ggplot(second_data, aes(x = median_tech, y = Rank, color = final_gender)) +
  geom_boxplot() +
  labs(title = "Association between Median_tech and Rank by Final_gender")

# Split data into training and testing sets
set.seed(123)
trainIndex <- createDataPartition(second_data$max_tech, p = 0.7, list = FALSE)
train <- second_data[trainIndex,]
test <- second_data[-trainIndex,]

#ML Models
# Logistic regression
logistic_model <- train(median_tech_binary ~ Rank + final_grad_year,
                        data = train, method = "glm", family = "binomial")
logistic_pred <- predict(logistic_model, newdata = test)

# Random forest
rf_model <- train(median_tech_binary ~ Rank + final_grad_year,
                  data = train, method = "rf")
rf_pred <- predict(rf_model, newdata = test)
```

```
#decision tree
library(party)
model<- ctree(median_tech_binary ~ Rank + final_grad_year,
              data = train)
plot(model)

predict_model<-predict(model, newdata = test)

m_at <- table(test$median_tech_binary, predict_model)
m_at



# Evaluate models
logistic_acc <- confusionMatrix(logistic_pred, test$median_tech_binary)$overall["Accuracy"]
rf_acc <- confusionMatrix(rf_pred, test$median_tech_binary)$overall["Accuracy"]
dc_acc <- confusionMatrix(predict_model, test$median_tech_binary)$overall["Accuracy"]
# svm_acc <- confusionMatrix(svm_pred, test$median_tech_binary)$overall["Accuracy"]
# xgb_acc <- confusionMatrix(xgb_pred, test$median_tech_binary)$overall["Accuracy"]

logistic_acc
rf_acc
dc_acc


# Compare models
accuracies <- data.frame(Model = c("Logistic Regression", "Random Forest","decision tree"),
                         Accuracy = c(logistic_acc, rf_acc,dc_acc ))

best_model <- accuracies[which.max(accuracies$Accuracy), "Model"]
print(paste("The best model is", best_model))

# Plot accuracies
ggplot(accuracies, aes(x = Model, y = Accuracy)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(x = "Model", y = "Accuracy", title = "Comparison of Model Accuracies")




#++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++
```