```
#   M A T H E M A T I C A L   F O U N D A T I O N
#-------------------------------------------------
# Members :- A K A N K S H A   G O E L ( 02 ) ;  A S H I S H   ( 11 ), DEEPTI ( 18 )
# Topic :- F O R E I G N   S T U D E  N T  E N R O L L M E N T
#-------------------------------------------------------------------------------------
#install.packages('readr')
library(readr)
#READING CSV
df_student=read_csv(file="C:/Users/arpit/Desktop/student_enrollment.csv")
View(df_student)
ncol(df_student)
nrow(df_student)

summary(df_student)
summary(df_student['Country'])
unique((df_student['Country']))
summary(df_student['Grand Total'])
#column specifications
attributes(df_student)
#Data type of each column
spec(df_student)

# 33156 Students are enrolled
sum(df_student['Grand_Total'])
# P R E P R O C E S S I N G


#RENAMING COLUMN NAMES i.e `UNIVERSITY - Male` ----> UNIVERSITY_Male
names(df_student) <- gsub(" - ", "_", names(df_student))
names(df_student) <- gsub(" ", "_", names(df_student))
names(df_student)

#AGGREGATING MALE COLUMNS
male_StandAlone=colnames(df_student)[grepl("Male",colnames(df_student))]
male_StandAlone=male_StandAlone[grepl("STAND_ALONE",male_StandAlone)]

#AGGREGATING FEMALE COLUMNS
female_StandAlone=colnames(df_student)[grepl("Female",colnames(df_student))]
female_StandAlone=female_StandAlone[grepl("STAND_ALONE",female_StandAlone)]
#Creating 2 new columns of Stand Alone Male and female
df_student$STAND_ALONE_Male_Total= rowSums(df_student[, male_StandAlone])
df_student$STAND_ALONE_Female_Total= rowSums(df_student[, female_StandAlone])

summary(df_student$STAND_ALONE_Female_Total)
summary(df_student$STAND_ALONE_Male_Total)

colnames(df_student)
#NOW TOTAL COLUMNS ARE 19
#Dropping the stand alone redundant male columns
col_indices_male <- match(male_StandAlone, names(df_student))
df_student <- df_student[, -col_indices_male]
#Dropping the stand alone redundant female columns
col_indices_female <- match(female_StandAlone, names(df_student))
df_student <- df_student[, -col_indices_female]

colnames(df_student)
#Now 9 columns
# Rearranging the columns
cols <- names(df_student)
cols <- cols[cols != "Grand_Total"]
df_student <- df_student[, c(cols, "Grand_Total")]
colnames(df_student)
```

```
# NOW OUR DATA SET IS READY FOR VISUALISATION
# DIMENSIONS 1905 rows * 9 columns
#--------------------
# PLOTTING          |
#--------------------
# BARPLOT           |
# DOUBLE BAR PLOT   |
# PIE CHART         |
# 3 D PIE CHART     |
# CHOROPLETH       |
#--------------------

#install.packages("ggplot2")
library("ggplot2")
library("reshape2")
library(dplyr)


# P L O T - 1 ( B A R P L O T )
#-----------------------------------------
# G R O U P   B Y   C O U N T R Y
#---------------------------------
student_country = df_student %>% group_by(Country) %>% summarise(Total_pop=
sum(Grand_Total))
student_country <- student_country[order(student_country$Total_pop, decreasing = TRUE), ]
#student_country<-student_country[-1,]
head(student_country,45)
View(student_country)
#PLOTTING BARPLOT
ggplot(head(student_country,45), aes(x = reorder(Country,-Total_pop),y=Total_pop)) +
  geom_bar(stat = "identity",color='#330066',fill='#cc0066') +
  geom_text(aes(label = Total_pop), hjust = 0.5, vjust = -0.3) +
  ggtitle("\t\t\t\t\t\t\tCOUNTRY WISE POPULATION DISTRIBUTION" ,subtitle = "Top 35
Countries")+
   xlab("COUNTRY")+ ylab("POPULATION") + theme_minimal()+
  theme(axis.text.x = element_text(face = "italic",family = "sans",
color="#330000",size=8, angle=90) ,
        axis.text.y = element_text(family = "sans",color = "#330000",face =
"italic",size=8),
        plot.title = element_text(family = "mono",color ="#330033" ,face =
"bold.italic",size = 18),
        plot.subtitle = element_text(family = "mono",color ="#660066" ,face =
"italic",size = 10))


#  G R O U P   B Y  P R O G R A M M  E
#-------------------------------------
student_Programme = df_student %>% group_by(Programme) %>% summarise(Total_pop=
sum(Grand_Total))
student_Programme <- student_Programme[order(student_Programme$Total_pop, decreasing =
TRUE), ]
View(student_Programme)
#plotting bar plot
ggplot(head(student_Programme,35), aes(x = reorder(Programme,-
Total_pop),y=Total_pop,fill=Total_pop)) +
  geom_bar(stat = "identity" ,color='#000066',fill='#0099ff') +
  geom_text(aes(label = Total_pop), hjust = 0.5, vjust = -0.2) +
  ggtitle("\t\t\tPROGRAMME WISE POPULATION DISTRIBUTION" ,subtitle = "Top 35 programs") +
  xlab("PROGRAMME")+ ylab("POPULATION") + theme_minimal()+
  theme(axis.text.x = element_text(face = "italic",family = "sans", color="#0000cc",size
=8, angle=90) ,
      axis.text.y = element_text(family = "sans",color = "#0000cc",face = "bold",size=8),
      plot.title = element_text(family = "mono",color ="#660033" ,face =
"bold.italic",size = 16),
```

```
      plot.subtitle = element_text(family = "mono",color ="#660066" ,face = "italic",size
= 12))

# P L O T - 2 ( D O U B L E    B A R P L O T )
#----------------------------------------
# S T A N D   A L O N E   M A L E   A N D   F E M A L E
#-------------------------------------------------------

# C O U N T R Y
#---------------
library(dplyr)
# G R O U P   BY  C O U N T R Y
#------------------------------

# F E M A L E
df1 = df_student %>% group_by(Country) %>% summarise(S_Female=
sum(STAND_ALONE_Female_Total))
df1 <- df1[order(df1$S_Female, decreasing = TRUE), ]

# M A L E
df2 = df_student %>% group_by(Country) %>% summarise(S_male= sum(STAND_ALONE_Male_Total))
df2 <- df2[order(df2$S_male, decreasing = TRUE), ]

# A G G R E G A T I N G   S T A N D   A L O N E  C O L U M N S
df3 = merge(x=df1,y=df2,by="Country")
df3$total=df3$S_Female+df3$S_male
df3 <- df3[order(df3$total, decreasing = TRUE), ]
df3 = head(df3,30)
df3
data_stand <- melt(df3[,-4], id.vars = "Country", variable.name = "Gender", value.name =
"Count")

# create a double bar plot
ggplot(data_stand, aes(x = Country, y = Count, fill = Gender)) +
geom_bar(stat = "identity", position = "dodge") +
labs(title = "Stand Alone Male and Female Students   ---- C O U N T R Y ", x = "Country",
y = "Number of Students",fill = "Gender")+
theme(axis.text = element_text(face="bold", color="#993333",size=10, angle=90))


# P R O G R A M M E
#------------------
library(dplyr)
# G R O U P   BY  P R O G R A M M E
df1 = df_student %>% group_by(Programme) %>% summarise(S_Female=
sum(STAND_ALONE_Female_Total))
df1 <- df1[order(df1$S_Female, decreasing = TRUE), ]

# G R O U P   BY
df2 = df_student %>% group_by(Programme) %>% summarise(S_male=
sum(STAND_ALONE_Male_Total))
df2 <- df2[order(df2$S_male, decreasing = TRUE), ]

# A G G R E G A T I N G   S T A N D   A L O N E  C O L U M N S
df3 = merge(x=df1,y=df2,by="Programme")
df3$total=df3$S_Female+df3$S_male
df3 <- df3[order(df3$total, decreasing = TRUE), ]
df3 = head(df3,12)
df3
data_stand <- melt(df3[,-4], id.vars = "Programme", variable.name = "Gender", value.name =
"Count")
# create a double bar plot
ggplot(data_stand, aes(x = Programme, y = Count, fill = Gender)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Stand Alone Male and Female Students   ---- P R O G R A M M E ", x =
```

```
"Programme", y = "Number of Students",fill = "Gender")+
  theme(axis.text = element_text(face="bold", color="#993333",size=10, angle=90))



# P L O T - 3  ( P I E  C H A R T  )
#----------------------------------------
#install.packages("tidyr")
library(tidyr)
library(ggplot2)
student_Programme$pie_program<-sapply(strsplit(student_Programme$Programme, "-"), '[', -2)
#calculating percentages for each programme
lab <- paste0(student_Programme$pie_program," ",

round(head(student_Programme$Total_pop,15)/sum(head(student_Programme$Total_pop,15)) *
100, 2), "%")
lab=head(lab,10)
ggplot(head(student_Programme,10), aes(x = Programme, y = Total_pop, fill = Total_pop)) +
  geom_col() +
  geom_text(aes(label = lab), color = "#000000")+
  coord_polar()+ xlab("")+ ylab("") +
  ggtitle("                        PROGRAMME WISE ")+
  theme(axis.text.x=element_blank(),axis.text.y=element_blank())



# P L O T - 4   ( 3 - D   P I E   C H A R T  )
#----------------------------------------

#install.packages('plotrix',dependencies=TRUE)
library(plotrix)
# P R O G R A M M E     W I S E
#--------------------------------
#Extracting Programme names
student_Programme$pie_program<-sapply(strsplit(student_Programme$Programme, "-"), '[', -2)
#calculating percentages for each programme
lab <- paste0(student_Programme$pie_program," ",

round(head(student_Programme$Total_pop,15)/sum(head(student_Programme$Total_pop,15)) *
100, 2), "%")
lab
plt<-pie3D(head(student_Programme$Total_pop,15),radius = 0.9 , mar = rep(3, 4) ,
           labels = head(lab,15),labelcex = 0.9,
           col = hcl.colors(24, "Spectral"),
           explode = 0.1,main="PROGRAMME WISE DISTRIBUTION",theta
=1.2,border="white",height = 0.1)

# C O U N T R Y   W I S E
#--------------------------
student_country$pie_country<-student_country$Country
#calculating percentages for each country
lab1 <- paste0(student_country$pie_country," ",

round(head(student_country$Total_pop,10)/sum(head(student_country$Total_pop,10)) * 100,
2), "%")
lab1
head(lab1,10)
plt<-pie3D(head(student_country$Total_pop,10),radius = 0.85 , mar = rep(1.5, 4) ,
           labels = head(lab1,10),labelcex = 0.8,
           col = hcl.colors(18, "Spectral"),
           explode = 0.1,main="COUNTRY WISE DISTRIBUTION\t",theta
=1.2,border="white",height = 0.1)


# P L O T - 5  ( C H O R O P L E T H  )
```

```
#-----------------------------------------
#install.packages("countrycode")
library(countrycode)
#Extracting Country codes
cname=countryname(student_country$Country, destination = 'iso3c')
cname
student_country$Codes=cname
colnames(student_country)

#install.packages("plotly",dependencies = "True")
# B A S I C
library(plotly)
fig <- plot_ly(student_country, type='choropleth',locations=student_country$Codes,
z=student_country$Total_pop,colorscale="Spectral")
fig <- fig %>% colorbar(title = "Population" , tickprefix = '~')
fig <- fig %>% layout(title = "FOREIGN STUDENT ENROLLMENT" )
fig
```