COCHIN Oriane, FISSON Jeanne, FOUCHER Tessa

R SOFTWARE - OUR CODE

#charge the table modified by us
R <- read.csv("~/Downloads/RICA_isafe_students_vf.csv", sep=";", skip = 2, header = TRUE
library (stats)

 # A) MAKE A BARPLOT

#create a sous ensemble of data for the region 76
R76 <- subset(R, NREG == 76)
#create a sous ensemble of data for the region 84
R84 <- subset(R, NREG == 84)

#count the number of observation for each production type of each region
observations_76 <- table(R76$OTEXE)
observations_84<- table(R84$OTEXE)

#create a vector of regions name
regions <- c("Region 76", "Region 84")

#créer une matrice avec les données pour les deux régions
matrice_donnees <- rbind(observations_76, observations_84)

#barplot
barplot(matrice_donnees, beside = TRUE,
    main = "Répartition des types de productions pour les régions 76 et 84",
    xlab = "types de production", ylab = "Nombre d'observations",
    col = c("blue", "red"), legend = regions)

 # B) SHAPIRO WILK TEST (for a computer where the table is called R76)

 # shapiro test for each production type for only the region 76 (we can do it also for the region 84 if we want)
shapiro.test(R76$PRODV)
shapiro.test(R76$PRODA)
shapiro.test(R76$PRODT)
shapiro.test(R76$PRODH)

#representation plot for each
qqnorm(R76$PRODV)
 # to compare with normal distribution
qqline(rnorm(1000))

qqnorm(R76$PRODA)
 # to compare with normal distribution
qqline(rnorm(1000))

qqnorm(R76$PRODT)
 # to compare with normal distribution
qqline(rnorm(1000))

qqnorm(R76$PRODH)
 # to compare with normal distribution
qqline(rnorm(1000))

 # then to try to make with filtrating the values of PRODV, PRODA, PRODT, and PRODH different than zero
PRODV_non_zero <- R76$PRODV[R76$PRODV != 0]
PRODA_non_zero <- R76$PRODA[R76$PRODA != 0]
PRODT_non_zero <- R76$PRODT[R76$PRODT != 0]
PRODH_non_zero <- R76$PRODH[R76$PRODH != 0]

#to create a boxplot of the values of PRODV, PRODA, PRODT, and PRODH not egal to 0
boxplot(PRODV_non_zero,
+ main = "Boxplot of the PRODV variable (non zero values)",
+ ylab = "Value of PRODV")

boxplot(PRODA_non_zero,
+ main = "Boxplot of the PRODA variable (non zero values)",
+ ylab = "Value of PRODA")

boxplot(PRODT_non_zero,
+ main = "Boxplot of the PRODT variable (non zero values)",
+ ylab = "Value of PRODT")

boxplot(PRODH_non_zero,
+ main = "Boxplot of the PRODH variable (non zero values)",
+ ylab = "Value of PRODH")

 # Optional : example for PRODH :
 # to add precise value on the box plot

```
text(x = 1, y = min(PRODH_non_zero) - 2, labels = paste("Min =", min(PRODH_non_zero)), pos
= 1)
text(x = 1, y = quantile(PRODH_non_zero, 0.25), labels = paste("Q1 =",
quantile(PRODH_non_zero, 0.25)), pos = 1)
text(x = 1, y = median(PRODH_non_zero), labels = paste("Med =",
median(PRODH_non_zero)), pos = 1)
text(x = 1, y = quantile(PRODH_non_zero, 0.75), labels = paste("Q3 =",
quantile(PRODH_non_zero, 0.75)), pos = 1)
text(x = 1, y = max(PRODH_non_zero) + 2, labels = paste("Max =", max(PRODH_non_zero)),
pos = 1)


# to have a pie plot of the production types distribution in the region 76
# to select the columns corresponding to the type of production :
types_production <- R76[, c("PRODV", "PRODH", "PRODA", "PRODT")]
# to count the number of values that are higher than zero, for each production type
type_counts <- colSums(types_production > 0)
# to calculate the percentages
type_percentages <- type_counts / sum(type_counts) * 100

# to create the pie plot
pie(type_percentages,
+     main = "Percentage of production types with values greater than zero",
+     labels = names(type_counts),
+     col = rainbow(length(type_percentages)),
+     cex = 0.8)


# C) CHI SQUARE

# a) Chisquare of surface and operating income
R$SUTOT_cat <- cut(R$SUTOT, breaks = c(0, 100, 500, 1000, Inf),
+                                   labels = c(“Petite”, “Moyenne”, “Grande”, “Très grande”))
#create contingency table
tableau_contigence <- table(R$RESEX, R$SUTOT_cat)
#execute Chi-Square test of independence
resultat_test <- chisq.test(tableau_contigence)
#print the test results
print(resultat_chitest)

plot(R$SUTOT, R$RESEX,
+        xlab = “Surface”, ylab = “Résultats d’exploitation”,
+        main = “Nuage de points entre Surface et Résultat d’exploitation”)
```

 # b) Charges correlation with surface
R <- apply(R, 2, as.numeric)
R_numeric <- R[, c("SUTOT","CHRAM", "CHRSE", "CHRPH", "CHRAC", "CHRAG", "CHRPV", "CHRCA", "ELECTQUA", "ENTBT", "ENTMT", "LFERM", "LMATE", "ASSRE", "ASSAU", "TXPRO", "TAXES", "AIMTX", "FPERS","CHSOC","CFINL","CAGR4")]
 # verify if conversion success :
str(R_numeric)

correlation_matrixRnum <- cor(R_numeric)
print(correlation_matrixRnum)

library(corrplot)
corrplot(correlation_matrixRnum, method = "color", type = "upper", order = "hclust", addrect = 2, rect.col = "red")

 # D) VARIANCE (possible to do it for both regions to compare them but we only did it for the region 76 by adding ",R84$RESEX", after "R76$RSESEX" for example**)**

 # To make sure we have numerical values for OTEXE and RESEX :
class(R76$OTEXE)
class(R76$RESEX)

 # to convert R76$OTEXE and R76$RESEX into numerical values
R76$OTEXE <- as.numeric(R76$OTEXE)
R76$RESEX <- as.numeric(R76$RESEX)

 # to perform the variance test
var_test_result <- var.test(R76$OTEXE, R76$RESEX)
var_test_result

 # to calculate the mean of RESEX for each unique value of OTEXE
result <- aggregate(RESEX ~ OTEXE, data = R76, FUN = mean)
 # to calculate the mean of RESEX for each unique value of OTEXE
result <- tapply(R76$RESEX, R76$OTEXE, mean)
result

#to plot a barplot
barplot(result, xlab = "OTEXE", ylab = "Mean of RESEX",
+ main = "Mean of RESEX for each unique value of OTEXE")

 # E) T-test

#operating income of a farm according to the regions 76 and 84:

```
resultat-test<- t.test(R76$RSESEX, R84$RESEX)
print(resultat_test)
```

 # F) T-test

#We will make several test depending to the surface but also if the region is situated into a protected region or not :

```
filtered_data_small_zenvi <- R[R$SUTOT < 11 & R$ZENVI == 1, ]
filtered_data_small_nozenvi <- R[R$SUTOT < 11 & R$ZENVI == 0, ]
filtered_data_medium_zenvi <- R[R$SUTOT > 10 & R$SUTOT < 21 & R$ZENVI == 1, ]
filtered_data_medium_nozenvi <- R[R$SUTOT > 10 & R$SUTOT < 21 & R$ZENVI == 0, ]
filtered_data_big_zenvi <- R[R$SUTOT > 20 & R$ZENVI == 1, ]
filtered_data_big_nozenvi <- R[R$SUTOT > 20 & R$ZENVI == 0, ]
```

#operating income a surface between 0 to 10 ha according to if there is a ZENVI

```
resultat_test1 <- t.test(filtered_data_small_zenvi, filtered_data_small_nozenvi)
print(resultat_test1)
```

#operating income a surface between 11 to 20 ha according to if there is a ZENVI

```
resultat_test2 <- t.test(filtered_data_medium_zenvi, filtered_data_medium_nozenvi)
print(resultat_test2)
```

#operating income a surface between 21 to 30 ha according to if there is a ZENVI

```
resultat_test3 <- t.test(filtered_data_big_zenvi, filtered_data_big_nozenvi)
print(resultat_test3)
```

 # G) CHARGES

=> Correlation : for each PROD (for the region 76, but can also do it for the region 84 to compare), we know that the better would be with the OTEXE but it was longer to do it (because we have 17 different OTEXE)

 # to define 4 numerical lists

```
R76_numericV <- R76[, c("PRODV","CHRAM", "CHRSE", "CHRPH", "CHRAC", "CHRAG",
"CHRPV", "CHRCA", "ELECTQUA", "ENTBT", "ENTMT", "LFERM", "LMATE", "ASSRE",
"ASSAU", "TXPRO", "TAXES", "AIMTX", "FPERS","CHSOC","CFINL","CAGR4")]
R76_numericV <- sapply(R76_numericV, as.numeric, na.rm = TRUE)

R76_numericA <- R76[, c("PRODA","CHRAM", "CHRSE", "CHRPH", "CHRAC", "CHRAG",
"CHRPV", "CHRCA", "ELECTQUA", "ENTBT", "ENTMT", "LFERM", "LMATE", "ASSRE",
"ASSAU", "TXPRO", "TAXES", "AIMTX", "FPERS","CHSOC","CFINL","CAGR4")]
R76_numericA <- sapply(R76_numericA, as.numeric, na.rm = TRUE)
```

```r
R76_numericT <- R76[, c("PRODT","CHRAM", "CHRSE", "CHRPH", "CHRAC", "CHRAG", "CHRPV", "CHRCA", "ELECTQUA", "ENTBT", "ENTMT", "LFERM", "LMATE", "ASSRE", "ASSAU", "TXPRO", "TAXES", "AIMTX", "FPERS","CHSOC","CFINL","CAGR4")]
R76_numericT <- sapply(R76_numericT, as.numeric, na.rm = TRUE)

R76_numericH <- R76[, c("PRODH","CHRAM", "CHRSE", "CHRPH", "CHRAC", "CHRAG", "CHRPV", "CHRCA", "ELECTQUA", "ENTBT", "ENTMT", "LFERM", "LMATE", "ASSRE", "ASSAU", "TXPRO", "TAXES", "AIMTX", "FPERS","CHSOC","CFINL","CAGR4")]
R76_numericH <- sapply(R76_numericH, as.numeric, na.rm = TRUE)


# to verify if the conversion succeed
str(R76_numericV)
str(R76_numericA)
str(R76_numericT)
str(R76_numericH)

# to calculate the 4 correlation
correlation_matrixV <- cor(R76_numericV)
correlation_matrixA <- cor(R76_numericA)
correlation_matrixT <- cor(R76_numericT)
correlation_matrixH <- cor(R76_numericH)

# to see the result if we want :
print(correlation_matrixV)
print(correlation_matrixA)
print(correlation_matrixT)
print(correlation_matrixH)

#to plot a correlation plot
corrplot(correlation_matrixV, method = "color", type = "upper", order = "hclust", addrect = 2, rect.col = "red")
corrplot(correlation_matrixA, method = "color", type = "upper", order = "hclust", addrect = 2, rect.col = "red")
corrplot(correlation_matrixT, method = "color", type = "upper", order = "hclust", addrect = 2, rect.col = "red")
corrplot(correlation_matrixH, method = "color", type = "upper", order = "hclust", addrect = 2, rect.col = "red")
```