```
library(tidyverse)
ff_steam_df <- read_csv("ff_steam_expt.csv")

# Filter for only forced, not planned or reserve shutdowns, outages and derates
type_filter <- c("D1", "D2", "D3", "D4", "SF", "U1", "U2", "U3")
ff_steam_df <- ff_steam_df %>%
  filter(ff_steam_df$Type %in% type_filter)

# Calculate mean Time_To_Repair for each outage Type
ff_steam_df %>% group_by(Type) %>%
  summarise(Mean_TTR_Type = mean(Time_To_Repair, na.rm = TRUE)) %>%
  arrange(desc(Mean_TTR_Type))

# Create a Box Plot graph, x = Type, y = Time_To_Repair
ggplot(data = ff_steam_df, mapping = aes(x = Type, y = Time_To_Repair)) +
  geom_boxplot()

# create a stat summary plot, x = year(start_st), y = time to repair
ggplot(data = ff_steam_df) +
  stat_summary(
    mapping = aes(x = year(start_dt), y = Time_To_Repair),
    fun.ymin = min,
    fun.ymax = max,
    fun.y = mean
  ) +
  facet_wrap(~ Type, nrow = 2)

#Convert Cause_Code to numeric
Cause_Code <- as.numeric(ff_steam_df$Cause_Code)

#Use conditions described in Appendix B
Cause_Code <- ifelse(Cause_Code>=10 & Cause_Code<=1999,"Boiler",
              ifelse(Cause_Code>=3110 & Cause_Code<=3999, "Balance of Plant",
              ifelse(Cause_Code>=4000 & Cause_Code<=4499, "Steam Turbbine",
              ifelse(Cause_Code>=4500 & Cause_Code<=5298, "Generator",
              ifelse(Cause_Code>=8000 & Cause_Code<=8845, "PCR",
              ifelse((Cause_Code>=9000 & Cause_Code<=9340 | Cause_Code==0), "External",
              ifelse(Cause_Code>=9504 & Cause_Code<=9720, "Regulatory",
              ifelse(Cause_Code>=9900 & Cause_Code<=9960, "Personnel",
              ifelse((Cause_Code==2 | Cause_Code>=9990 & Cause_Code<=9991), "InactiveState","Performance")))))))))

#Add it back to the dataframe
ff_steam_df$System <- Cause_Code

##########################################
#Clustering

#First select the variables of importance
ClData <- ff_steam_df %>% select(System, Time_To_Repair:X_Derate)
ClData <- ClData[complete.cases(ClData),]
df <- scale(ClData[-1])

wssplot <- function(data, nc=15, seed=1234){
  wss <- (nrow(data)-1)*sum(apply(data,2,var))
  for (i in 2:nc){
    set.seed(seed)
    wss[i] <- sum(kmeans(data, centers=i)$withinss)}
  plot(1:nc, wss, type="b", xlab="Number of Clusters",
       ylab="Within groups sum of squares")}

wssplot(df)

library(NbClust)
nc <- NbClust(df, min.nc=2, max.nc=15, method="kmeans") #Memory exhaustion
fit.km <- kmeans(df, 6, nstart=25)
fit.km$size
fit.km$centers
aggregate(ClData[-1], by=list(cluster=fit.km$cluster), mean)
ct.km <- table(ClData$System, fit.km$cluster)

library(flexclust)
randIndex(ct.km)

#Using PAM
library(cluster)
set.seed(666)
fit.pam <- pam(ClData[-1], k=6, stand=TRUE)# Memory exhaustion
```