

```

# Table 1

#~~~~~
# Libraries
#~~~~~
library(dplyr)
library(ggplot2)
library(tidyr)

#~~~~~
# Functions
#~~~~~

rescale_01 <-function(x) (x-min(x))/(max(x)-min(x)) -1/2
z_stand<-function(x) (x-mean(x))/sd(x)
ExpectedBrix <- function(x) (x*0.21084778699754 + 4.28455310831511)

#~~~~~
# Thresholds
#~~~~~
Thresh.Brix.min <- 15
Thresh.Brix.max <- 30

Thresh.Pol.min <- 50
Thresh.Pol.max <- 105
ExpectedBrix.delta <- 1

Thresh.Fibre.min <- 4
Thresh.Fibre.max <- 25
Thresh.Fibre.delta <- .25

Thresh.Ash.min <- 0
Thresh.Ash.max <- 8

#Fibre = 100 * (InitialSampleCanWeight - FinalSampleCanWeight) / SampleWeight

# Table 2
Lab_Fibre_Data = read.csv(file = "Sugar_Cane_Input_Files/Lab_Fibre_Weights.csv",
                          header = T, sep = ",", dec = ".")

# Table 3
Lab_Fibre_Data$Fibre1 = 100 * (Lab_Fibre_Data$InitialSampleCanWeight_1 -
                               Lab_Fibre_Data$FinalSampleCanWeight_1) / Lab_Fibre_Data$SampleWeight_1

# Table 4
Lab_Fibre_Data = Lab_Fibre_Data %>% mutate(Fibre2 = 100 *
                                             (InitialSampleCanWeight_2 - FinalSampleCanWeight_2) / SampleWeight_2)

# Table 5
Lab_Fibre_Filtered = Lab_Fibre_Data %>% filter(InitialSampleCanWeight_1 > 0 & FinalSampleCanWeight_1 > 0
                                              & SampleWeight_1 > 0 & InitialSampleCanWeight_2 > 0
                                              & FinalSampleCanWeight_2 > 0 & SampleWeight_2 > 0)

# Table 6 Update codes, so still using the Lab_Fibre_Data
Lab_Fibre_Filtered = Lab_Fibre_Data %>% filter(InitialSampleCanWeight_1 > 0 & FinalSampleCanWeight_1 > 0
                                              & SampleWeight_1 > 0 & InitialSampleCanWeight_2 > 0
                                              & FinalSampleCanWeight_2 > 0 & SampleWeight_2 > 0) %>% filter(abs(Fibre1 - Fibre2) < 0.25)

# Table 7
Lab_Fibre_Filtered = Lab_Fibre_Filtered %>% mutate(Fibre = (Fibre1 + Fibre2)/2)

# Table 8
Lab_Fibre_Filtered = Lab_Fibre_Filtered %>% filter(Fibre > 4) %>% filter(Fibre < 25)

```

```

# Table 9
Lab_Fibre = Lab_Fibre_Filtered %>% select(LabID, Fibre)

# Ash = 100 * FinalWeight / InitialWeight
# where :
#   InitialWeight = InitialSampleInTinWeight - TinWeight
#   FinalWeight   = FinalSampleInTinWeight - TinWeight

# Table 10
Lab_Ash_Data = read.csv(file = "Sugar_Cane_Input_Files/Lab_Ash_Weights.csv",
                        header = T, sep = ",", dec = ".")

# Table 11
Lab_Ash_Calculated = Lab_Ash_Data %>% filter(TinWeight > 0 & InitialSampleInTinWeight > 0
                                             & FinalSampleInTinWeight > 0) %>%
  mutate(Ash = 100 * (FinalSampleInTinWeight - TinWeight) / (InitialSampleInTinWeight - TinWeight) )

# Table 12
Lab_Ash_Filtered = Lab_Ash_Calculated %>% filter(Ash > 0) %>% filter(Ash < 8)

# Table 13
Lab_Ash = Lab_Ash_Filtered %>% group_by(LabID) %>% summarise(Ash = mean(Ash))

# Table 14
Lab_PB_Data = read.csv(file = "Sugar_Cane_Input_Files/Lab_Pol_Brix.csv",
                       header = T, sep = ",", dec = ".")

# Table 15 ExpectedBrix
Lab_PB_Data = Lab_PB_Data %>% mutate(PredBrix = ExpectedBrix(Pol))

# Table 16
z = factor(ifelse(abs(Lab_PB_Data$Brix - Lab_PB_Data$PredBrix) > 1, 1, 0))
plot(Lab_PB_Data$Brix, Lab_PB_Data$PredBrix, col = z, pch = 16, main = "Relationship between measured Brix and predicted Brix",
     xlab = "Measured Brix", ylab = "Predicted Brix")
legend("bottomright", legend = c("z=0", "z=1"), col = c("red", "black"), pch = rep(16, 2))

# Table 17
Lab_PB = Lab_PB_Data %>% filter(abs(Lab_PB_Data$Brix - Lab_PB_Data$PredBrix) < 1) %>%
  filter(Brix > 15 & Brix < 30 & Pol > 50 & Pol < 105) %>%
  select(LabID, Pol, Brix )

# Table 18
Lab = full_join(Lab_Ash, Lab_Fibre, by=c("LabID" = "LabID"))

# Table 19
Lab = full_join(Lab, Lab_PB, by=c("LabID" = "LabID"))

# Table 20
write.table(Lab, file = "Lab_Out.csv", append = FALSE, quote = TRUE, sep = ",",
           eol = "\n", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE,
           qmethod = c("escape", "double"), fileEncoding = "")

# Table 21
Lab_Fibre = transform(Lab_Fibre, Fibre = z_stand(Fibre))
write.table(Lab_Fibre, file = "Lab_Fibre_Out.csv", append = FALSE, quote = TRUE, sep = ",",
           eol = "\n", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE,
           qmethod = c("escape", "double"), fileEncoding = "")

# Table 22
Lab_Ash = Lab_Ash %>% mutate(Ash = log10(Ash)) %>% mutate(Ash = z_stand(Ash))

```

```

write.table(Lab_Ash, file = "Lab_Ash_Out.csv", append = FALSE, quote = TRUE, sep = ",",
            eol = "\n", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE,
            qmethod = c("escape", "double"), fileEncoding = "")

# Table 23
Lab_PB$Bbin = cut(Lab_PB$Brix, 40, labels = FALSE)

# Table 24
Lab_PB$Bbin = as.factor(Lab_PB$Bbin)

# Table 25
Lab_B_Stratified_Balanced = Lab_PB %>% group_by(Bbin) %>% summarise(sample_n(Lab_PB, size = 50, replace = T))

# Table 26
Lab_B_sampled = transform(Lab_B_Stratified_Balanced, Brix = rescale_01(Brix))
Lab_B_sampled = Lab_B_sampled %>% select(LabID, Brix)
write.table(Lab_B_sampled, file = "Lab_Brix_Out.csv", append = FALSE, quote = TRUE, sep = ",",
            eol = "\n", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE,
            qmethod = c("escape", "double"), fileEncoding = "")

# Table 27
Lab_P_Stratified_Balanced = Lab_PB %>% group_by(Bbin) %>% summarise(sample_n(Lab_PB, size = 50, replace = T))
Lab_P_sampled = Lab_P_Stratified_Balanced %>% mutate(Pol = rescale_01(Pol)) %>% select(LabID, Pol)
Lab_P_sampled = subset(Lab_P_sampled, select = -Bbin)
write.table(Lab_P_sampled, file = "Lab_Pol_Out.csv", append = FALSE, quote = TRUE, sep = ",",
            eol = "\n", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE,
            qmethod = c("escape", "double"), fileEncoding = "")

# Table 28
#~~~~~
# Libraries
#~~~~~
library(dplyr)
library(ggplot2)
library(tidyr)

#~~~~~
# Thresholds
#~~~~~
Thresh.Brix.min <- 15
Thresh.Brix.max <- 30

Thresh.Pol.min <- 50
Thresh.Pol.max <- 105

Thresh.Fibre.min <- 4
Thresh.Fibre.max <- 25

Thresh.Ash.min <- 0
Thresh.Ash.max <- 8

# Table 29
NIRData = read.csv("Sugar_Cane_Input_Files/NIRPred.csv",
                  header = T, sep = ",", dec = ".")
# Table 30 /// The time already is a character when I import NIRData does it
#               still need to change to POSIXct type? Cause RStudio already recognise it
NIRData$DateTime = as.POSIXct(NIRData$DateTime, format = "%Y-%m-%d %H:%M:%S")

# Table 31
NIRData$LabID = floor(NIRData$ScanID)

# Table 32

```

```

NIRData_Filtered = NIRData %>% filter(GH < 3.5 & NH < 2) %>%
  filter(NIR_Brix > 15 & NIR_Brix < 30 & NIR_Pol > 50 & NIR_Pol < 105 &
    NIR_Fibre > 4 & NIR_Fibre < 25 & NIR_Ash > 0 & NIR_Ash < 8) %>%
  filter(ScanID > 0)

# Table 33
NIR_Final = NIRData_Filtered %>% group_by(LabID) %>%
  summarise(DateTime = min(DateTime), NIR_Pol = mean(NIR_Pol), NIR_Brix = mean(NIR_Brix),
    NIR_Fibre = mean(NIR_Fibre), NIR_Ash = mean(NIR_Ash))

# Table 34
write.table(NIR_Final, file = "NIR_Final_Out.csv", append = FALSE, quote = TRUE, sep = ",",
  eol = "\n", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE,
  qmethod = c("escape", "double"), fileEncoding = "")

# Table 35
his1 = ggplot(data = NIR_Final) +
  geom_histogram(mapping = aes(NIR_Brix), fill = "red", bins = 100, alpha = 0.5) +
  theme_bw() + labs(x = "Pol", y = "Frequency")
his2 = ggplot(data = NIR_Final) +
  geom_histogram(mapping = aes(NIR_Fibre), fill = "blue", bins = 100, alpha = 0.5) +
  theme_bw() + labs(x = "Fibre", y = "Frequency")
gridExtra::grid.arrange(his1, his2, ncol = 2)

# Table 36
summary(NIR_Final$NIR_Brix)
summary(NIR_Final$NIR_Fibre)

# Table 37
line1 = ggplot(data = NIR_Final, mapping = aes(DateTime, NIR_Brix)) + geom_smooth(mapping = aes(DateTime, NIR_Brix)) + labs(x = "Month", y = "Brix")
line2 = ggplot(data = NIR_Final, mapping = aes(DateTime, NIR_Fibre)) + geom_smooth(mapping = aes(DateTime, NIR_Fibre)) + labs(x = "Month", y = "Fibre")
gridExtra::grid.arrange(line1, line2, ncol = 2)

```