```
##---------------------------------------##
## required libraries
##

install.packages("tidyverse") #loads packages from the book
install.packages("lubridate") #functions useful for working with dates
install.packages("readxl")

library(tidyverse)
library(lubridate)
library(readxl)



##---------------------------------------##
## data import
##
## before you start set your working directory to "source file location"
##
## import the "early" worksheet

early <- read_excel("mass_data.xlsx",sheet="early",skip=2) #skip first two lines

obs.date <- read_excel("mass_data.xlsx",
                       sheet="early",
                       col_names=FALSE,n_max=1)[[2]] #n_max> hor many lines we're reading

obs.date[[1]] #obs.date would give the same thing because it has only one element. If i had more i would need to specify
```

```
obs <- read_excel("mass_data.xlsx",
                  sheet="early",
                  col_names=FALSE,skip=1,n_max=1)[[2]] #skip one, read one (me da un elemento con la segunda linea)
obs[[1]]



## add the date and the observation label and date as columns to the data frame


early$obs.date <- ymd(obs.date[[1]]) ## ymd() strips the time-stamp from this entry (removes time from dates)
# compare obs.date[[1]]  to ymd(obs.date[[1]])


early$obs <- obs[[1]]


early ## print the early data.frame



## import the "late" worksheet


late <- read_excel("mass_data.xlsx",sheet="late",skip=2) ## read in the 1st sheet of .xlsx
obs.date <- read_excel("mass_data.xlsx",sheet="late",col_names=FALSE,n_max=1)[2]
obs <- read_excel("mass_data.xlsx",sheet="late",col_names=FALSE,skip=1,n_max=1)[2]


late$obs.date <- ymd(obs.date[[1]])
late$obs <- obs[[1]]


late ## print the late data.frame
```

```
##---------------------------------------##
## Lets put these two data frames together!

mass_data <- bind_rows(early,late) ## to use bind_rows, all column names must be the same

mass_data ## open from the global enviroment and examine



## bind_rows will only line up rows which share column names

colnames(early)
colnames(early)[1] ## print the frist element of the colnames vector
colnames(early)[2] ## the second element



colnames(early)[1]<-"plant_number" #rename column
colnames(early)

data2<-bind_rows(early,late) ## open from the global enviroment or run the line below
data2[116:124,]



##---------------------------------------##
## library(tidyverse)
## library(dplyr) - it is part of tidyverse
##---------------------------------------##
## filter() - select variables by a value
```

```
##

## logical operators

##

## >, <, >=, <=

## & and

## | or

## ! not

##



filter(mass_data, treatment == "control")



control_data <- filter(mass_data, treatment == "control") ## pulls out the control data

control_data



control_high_data <- filter(mass_data, treatment == "control" | treatment == "high" ) ## pulls data which treatment = control or high

control_high_data



early_control_block_1and3 <- filter(mass_data, treatment == "control" & obs == "early" & block != 2 ) ## pulls, control and early data from block 1 and 3

early_control_block_1and3



mean(early_control_block_1and3$mass) ## mean of mass from early observation control data, in blocks 1 and 3



filter(early_control_block_1and3, mass >= 20.17) ## pulls all data greater than the mean
```

```
##---------------------------------------##

## library(tidyverse)

## library(dplyr)

##---------------------------------------##

## select() - pick variables by their names

##


mass_data ## has 8 variables (plant_id, treatment, block, rep, notes, obs.date, obs)


select(mass_data, plant_id,treatment,mass) ## selects 3 variables, plant_id, treatment, mass



mass_data

select(mass_data, obs, plant_id, everything()) ## selects obs and date, then puts all other varables after it, useful for organizing data

#pone todo el resto al final


mass_data$notes ## useless column, since it has no data lets get rid of it

mass_data <- select(mass_data, -notes) ## selects everything but the notes columbn

mass_data



##---------------------------------------##

## library(tidyverse)

## library(dplyr)

##---------------------------------------##

## summarize() - summarize many values

##
```

summarize(mass_data, mean_mass = mean(mass)) ## computes the mean of the mass, same as mean(data$mass) - y la llama mean_mass

mean(mass_data$mass)

not very useful by itself, but when combined with group_by() it really shines

grouped_data <- group_by(mass_data, treatment, block, obs) ## creates a groups based on treatment,block,observation

grouped_data ## view this output and see -el output dice "groups"

summarize(grouped_data, mean_mass = mean(mass)) ## calculates the mean of each group

summarize(grouped_data, se=sd(mass)/sqrt(length(mass))) ## calculates the SE of each group

#SE esta calculado en lugar de ser una funcion

summarize(grouped_data, mean_mass = mean(mass), se=sd(mass)/sqrt(length(mass))) ## or put both statements together!

##--##

library(tidyverse)

library(tidyr)

##--##

spread() - two columns to multiple columns

##

mass_data ## suppose we want to calculate the difference in mass between the early and late observation

```
filter(mass_data,plant_id==1)

spread(mass_data, key = obs, value = mass) ## something here isn't right... we need to drop date or obs

spread_data <- select(mass_data,-obs.date)

spread_data <- spread(spread_data, key = obs, value = mass)

spread_data



filter(mass_data,plant_id==1) ## check these values against the output of spread_data

filter(mass_data,plant_id==2) ## check these values against the output of spread_data



spread_data$diff <- spread_data$late - spread_data$early ## calculate the difference between the late and early observation

spread_data

ggplot(spread_data,aes(treatment,diff)) + geom_boxplot() ## lets visiual the differences between early and late observations with a boxplot



## spread() is also useful for producing readable tables

grouped_data ## recall the grouped_data

means <- summarize(grouped_data, mean_mass = mean(mass), se=sd(mass)/sqrt(length(mass)))

means$mean_mass <- round(means$mean_mass,2) ## rounds the number to 2 sig digits

means$se <- round(means$se ,2) ## rounds the number to 2 sig digits
```

means

table.data <- unite(means, mean_mass, se, sep=" + ",col="mean + SE")

table.data

spread(table.data, key= obs, value = 'mean + SE') ##almost ready for the thesis

table <- spread(table.data, key= obs, value = 'mean + SE')

table

write_csv(table,"table_one.csv")

##--##

library(tidyverse)

library(tidyr)

##--##

gather() - column names are values not variables

##

spread_data ## is this data.frame tidy? ... no because early and late are values not a variable

tidy_data <- gather(spread_data, "early", "late", key = "observation", value = "mass")

tidy_data

nrow(tidy_data) ## 240 rows.. as expected