```
install.packages("dplyr")
install.packages("readxl")

library(dplyr)
library(readxl)

# Using the Read.table option with a CSV file to bring in all datasets #
#   file.choose() opens the File open dialigue box and allows you to choose where you file is and your file
#
#   header = T - T =TRUE, header = does the first row in your dataset have a header or variable names?  #
#   sep="," - you're now telling R that you have a CSV file, where a , separates your variables  #
jan_csv.data <- read.table(file.choose(), header=T, sep=",")

#  To view the data - just making sure it was read in correctly
jan_csv.data



feb_csv.data <- read.table(file.choose(), header=T, sep=",")
feb_csv.data



march_csv.data


april_csv.data




# Bringing in the datasets using the readxl option  #
#   we do not have to tell R which directory, because we set that out already  #
#   we specify the name of the Excel file, including the file ending  #
#   col_names=T or TRUE - we are telling R that the first row in our file has variable names  #
jan.data <- read_excel("January.xlsx", col_names = TRUE )
jan.data

jan.data <- read_excel("wksp2_data1.xlsx", sheet ="January", col_names = TRUE )
jan.data

#  Bring in February, March, and April



#  To view the data - just making sure it was read in correctly  #
#  Notice that it doesn't show you the entire dataset - just a tibble #
```

```
jan.data
feb.data



################################################################################################
#####################################
#  Merging January and February data - using the Merge function available in Base R  #
#   calling on the function merge()  #
#   listing the names of the datasets we want to merge and separating them with a ,  #
#   telling how we want the datasets to be merged using the by statement - where ID in the Jan file = ID
in the Feb file  #
#   and the same with the Trmt variable  #
janfeb_merge.data <- merge(jan.data, feb.data, by=c("ID"="ID", "Trmt"="Trmt"))
janfeb_merge.data
#  What happened ?



# taking advantage of the Joining functions available in the library DPLYR  #
janfeb.data <- full_join(jan.data, feb.data, by=c("ID"="ID", "Trmt"="Trmt"))
janfeb.data

janmar.data


janapr.data <- full_join(janmar.data, april.data, by=c("ID"="ID",  "Trmt"="Trmt"))
janapr.data


#  R keeps all the variables and we want to remove the extra pieces  #
#  First we create a vector of the variables we want to keep  and then we apply it to our dataset  #
myvars <- c("ID", "Trmt", "Weight_jan", "Weight_feb", "Weight_mar", "Weight_apr")
final_jannapr.data <- janapr.data[myvars]
final_jannapr.data


#  We want to create a new variable for wtgain from January to March and then a second one from
January to April  #

final.data$wtgain.janmar = final.data$Weight_mar - final.data$Weight_jan
final.data$wtgain.janapr

#View the new variables

final.data$wtgain.janmar
final.data$wtgain.janapr
```

```
# View the complete dataset
final.data

#  Another common method of creating a new variable is to recode one.  We will recode the weight taken in January  #
#  into a new variable called wtjancat where we will have 3 weight classes:  Light, middle, and heavy #
#  By using the attach() and detach() functions we are creating the new variables directly into our dataset       #

final.data$wtjancat[final.data$Weight_jan < 17] <- "Light weights"
final.data$wtjancat[final.data$Weight_jan > 16 & final.data$Weight_jan <=20] <- "Middle weights"
# Create one for Heavy Weight


final.data
```