Lecture 2

## Agenda

Walk through Data Preparation using the “Welcome to the Team!” Minicase.

## Loading packages

Always start by loading the packages you will use. A good practice is to load all packages in your first code chunk.

Note that unlike R scripts, when you run a code chunk R runs all lines on it.

#You only need to install ONCE, comment the next line afterwards.   
#install.packages("tidyverse")   
#You need to call the library every time you want to use it   
library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.4 ✔ readr 2.1.5  
✔ forcats 1.0.0 ✔ stringr 1.5.1  
✔ ggplot2 3.5.2 ✔ tibble 3.3.0  
✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
✔ purrr 1.1.0   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

# Data Prep with Welcome to the Team

## Importing data

We will use read.csv to read the csv files into R. Make sure that the notebook file is saved in the **same folder** as your csv files. Two ways to make sure of it:

#A:   
# You can go to Session -> Set Working Directory -> Set Directory (choose the same folder as csv files) or to Source File Location if it is saved on the same folder as your csv file.   
  
#B:   
#Use getwd to get your current working directory and setwd to set a new one. You have to copy the file path to the folder where you have your csv file. Note it asks for '/' and not for '\'.   
getwd()

[1] "/Users/theresawohlever/git\_repos/BQOM-2578\_DataMining/BQOM-2578\_DataMining\_twohlever/assignments/02"

#setwd("C:/Users/XXXXXX/OneDrive - University of Pittsburgh/XXXXXXX")

Let’s clean up the environment:

# use the rm() function or the "broom" icon button under the Environment tab  
#  
#rm( list <- ls() )

After making sure everything is in the same folder, we can load the datasets that we will use through read.csv:

#read.csv will read the csv into a dataframe, which we can manipulate in R.   
sales <- read.csv("sales.csv")   
features <- read.csv("features.csv")   
stores <- read.csv("stores.csv")

## a. Discovery

#head displays the first rows   
head(sales)

Store Dept Date Weekly\_Sales IsHoliday  
1 1 1 2010-02-05 24924.50 FALSE  
2 1 1 2010-02-12 46039.49 TRUE  
3 1 1 2010-02-19 41595.55 FALSE  
4 1 1 2010-02-26 19403.54 FALSE  
5 1 1 2010-03-05 21827.90 FALSE  
6 1 1 2010-03-12 21043.39 FALSE

#tail displays the last rows   
tail(sales)

Store Dept Date Weekly\_Sales IsHoliday  
421565 45 98 2012-09-21 467.30 FALSE  
421566 45 98 2012-09-28 508.37 FALSE  
421567 45 98 2012-10-05 628.10 FALSE  
421568 45 98 2012-10-12 1061.02 FALSE  
421569 45 98 2012-10-19 760.01 FALSE  
421570 45 98 2012-10-26 1076.80 FALSE

#dim tells you how many rows by how many columns you have  
dim(sales)

[1] 421570 5

#names returns the names of the columns that you have   
names(sales)

[1] "Store" "Dept" "Date" "Weekly\_Sales" "IsHoliday"

#summary will give you relevant summary statistics for each variable depending on its type   
summary(sales)

Store Dept Date Weekly\_Sales   
 Min. : 1.0 Min. : 1.00 Length:421570 Min. : -4989   
 1st Qu.:11.0 1st Qu.:18.00 Class :character 1st Qu.: 2080   
 Median :22.0 Median :37.00 Mode :character Median : 7612   
 Mean :22.2 Mean :44.26 Mean : 15981   
 3rd Qu.:33.0 3rd Qu.:74.00 3rd Qu.: 20206   
 Max. :45.0 Max. :99.00 Max. :693099   
 IsHoliday   
 Mode :logical   
 FALSE:391909   
 TRUE :29661

* What is our level of analysis observation?
* What are our most important variables? Are they of the right type?
* What can we say about them?

**Do the same for features and stores.**

## B. Structuring

Now, the Team’s analysis should be at the Store level, not at the department level. Therefore, we can aggregate to the department level sales. That is, by adding the sales fo all departments in a store, we get sales at the store.

We will use piping %>% to do this. The code below groups the sales data by the store and date, and then returns for each grouping two items: the sum of the weekly sales and the mean of IsHoliday so that is info is not lost.

sales2<-sales %>% group\_by(Store,Date) %>% summarize(Weekly\_Sales=sum(Weekly\_Sales),IsHoliday=mean(IsHoliday))

`summarise()` has grouped output by 'Store'. You can override using the  
`.groups` argument.

head(sales2)

# A tibble: 6 × 4  
# Groups: Store [1]  
 Store Date Weekly\_Sales IsHoliday  
 <int> <chr> <dbl> <dbl>  
1 1 2010-02-05 1643691. 0  
2 1 2010-02-12 1641957. 1  
3 1 2010-02-19 1611968. 0  
4 1 2010-02-26 1409728. 0  
5 1 2010-03-05 1554807. 0  
6 1 2010-03-12 1439542. 0

dim(sales2) # compare this to the other datasets... features?

[1] 6435 4

summary(sales2)

Store Date Weekly\_Sales IsHoliday   
 Min. : 1 Length:6435 Min. : 209986 Min. :0.00000   
 1st Qu.:12 Class :character 1st Qu.: 553350 1st Qu.:0.00000   
 Median :23 Mode :character Median : 960746 Median :0.00000   
 Mean :23 Mean :1046965 Mean :0.06993   
 3rd Qu.:34 3rd Qu.:1420159 3rd Qu.:0.00000   
 Max. :45 Max. :3818686 Max. :1.00000

#To understand what we did, lets look at Sales for Store 1 on a specific date using our first dataset   
head(sales%>%filter(Store==1 & Date=='2010-02-12'))

Store Dept Date Weekly\_Sales IsHoliday  
1 1 1 2010-02-12 46039.49 TRUE  
2 1 2 2010-02-12 44682.74 TRUE  
3 1 3 2010-02-12 10887.84 TRUE  
4 1 4 2010-02-12 35351.21 TRUE  
5 1 5 2010-02-12 29620.81 TRUE  
6 1 6 2010-02-12 9135.00 TRUE

tail(sales%>%filter(Store==1 & Date=='2010-02-12'))

Store Dept Date Weekly\_Sales IsHoliday  
67 1 92 2010-02-12 143081.42 TRUE  
68 1 93 2010-02-12 70202.02 TRUE  
69 1 94 2010-02-12 62329.48 TRUE  
70 1 95 2010-02-12 111390.36 TRUE  
71 1 97 2010-02-12 33874.47 TRUE  
72 1 98 2010-02-12 16309.73 TRUE

#and our second dataset.   
sales2%>%filter(Store==1 & Date=='2010-02-12')

# A tibble: 1 × 4  
# Groups: Store [1]  
 Store Date Weekly\_Sales IsHoliday  
 <int> <chr> <dbl> <dbl>  
1 1 2010-02-12 1641957. 1

#What is the difference?

## C. Cleaning: Data Types and Missing Values

### Changing data type

Dates can be problematic. Notice that right now R did not recognize Date as a date, but just as text. We should tell R it is a date. To do that, we’ll benefit from another package: **lubridate**. https://lubridate.tidyverse.org/

#install.packages("lubridate")   
head(sales2$Date)

[1] "2010-02-05" "2010-02-12" "2010-02-19" "2010-02-26" "2010-03-05"  
[6] "2010-03-12"

library(lubridate)   
sales2$Date<-ymd(sales2$Date)   
#ymd is telling R the date format is year, month and day.   
#Be mindful: the order of month and day can vary in different regions and Excel tends to recognize dates and rearrange according to its region.   
#Make sure to check where are months and days placed when working with dates.   
#ydm would be the function for a date like 2010-26-02   
  
#We can also change IsHoliday type just to show you how type can be changed (it isn't really neccesary)   
#The str() function displays the internal structure of an object such as an array, list, matrix, factor, ...  
str(sales$IsHoliday)

logi [1:421570] FALSE TRUE FALSE FALSE FALSE FALSE ...

sales$IsHoliday<-as.factor(sales$IsHoliday)   
str(sales$IsHoliday)

Factor w/ 2 levels "FALSE","TRUE": 1 2 1 1 1 1 1 1 1 1 ...

# if you need it to be a logical variable, then use "as.logical(sales$IsHoliday)"  
  
summary(sales2) # what has changed?

Store Date Weekly\_Sales IsHoliday   
 Min. : 1 Min. :2010-02-05 Min. : 209986 Min. :0.00000   
 1st Qu.:12 1st Qu.:2010-10-08 1st Qu.: 553350 1st Qu.:0.00000   
 Median :23 Median :2011-06-17 Median : 960746 Median :0.00000   
 Mean :23 Mean :2011-06-17 Mean :1046965 Mean :0.06993   
 3rd Qu.:34 3rd Qu.:2012-02-24 3rd Qu.:1420159 3rd Qu.:0.00000   
 Max. :45 Max. :2012-10-26 Max. :3818686 Max. :1.00000

Now you can do the same for the Date variable in features and stores (insert a code block here).

**Features and Stores**

head(features)

Store Date Temperature Fuel\_Price MarkDown1 MarkDown2 MarkDown3  
1 1 2010-02-05 42.31 2.572 NA NA NA  
2 1 2010-02-12 38.51 2.548 NA NA NA  
3 1 2010-02-19 39.93 2.514 NA NA NA  
4 1 2010-02-26 46.63 2.561 NA NA NA  
5 1 2010-03-05 46.50 2.625 NA NA NA  
6 1 2010-03-12 57.79 2.667 NA NA NA  
 MarkDown4 MarkDown5 CPI Unemployment IsHoliday  
1 NA NA 211.0964 8.106 FALSE  
2 NA NA 211.2422 8.106 TRUE  
3 NA NA 211.2891 8.106 FALSE  
4 NA NA 211.3196 8.106 FALSE  
5 NA NA 211.3501 8.106 FALSE  
6 NA NA 211.3806 8.106 FALSE

tail(features)

Store Date Temperature Fuel\_Price MarkDown1 MarkDown2 MarkDown3  
8185 45 2013-06-21 70.13 3.626 4989.34 385.31 178.56  
8186 45 2013-06-28 76.05 3.639 4842.29 975.03 3.00  
8187 45 2013-07-05 77.50 3.614 9090.48 2268.58 582.74  
8188 45 2013-07-12 79.37 3.614 3789.94 1827.31 85.72  
8189 45 2013-07-19 82.84 3.737 2961.49 1047.07 204.19  
8190 45 2013-07-26 76.06 3.804 212.02 851.73 2.06  
 MarkDown4 MarkDown5 CPI Unemployment IsHoliday  
8185 2463.42 3117.94 NA NA FALSE  
8186 2449.97 3169.69 NA NA FALSE  
8187 5797.47 1514.93 NA NA FALSE  
8188 744.84 2150.36 NA NA FALSE  
8189 363.00 1059.46 NA NA FALSE  
8190 10.88 1864.57 NA NA FALSE

dim(features)

[1] 8190 12

names(features)

[1] "Store" "Date" "Temperature" "Fuel\_Price" "MarkDown1"   
 [6] "MarkDown2" "MarkDown3" "MarkDown4" "MarkDown5" "CPI"   
[11] "Unemployment" "IsHoliday"

summary(features)

Store Date Temperature Fuel\_Price   
 Min. : 1 Length:8190 Min. : -7.29 Min. :2.472   
 1st Qu.:12 Class :character 1st Qu.: 45.90 1st Qu.:3.041   
 Median :23 Mode :character Median : 60.71 Median :3.513   
 Mean :23 Mean : 59.36 Mean :3.406   
 3rd Qu.:34 3rd Qu.: 73.88 3rd Qu.:3.743   
 Max. :45 Max. :101.95 Max. :4.468   
   
 MarkDown1 MarkDown2 MarkDown3 MarkDown4   
 Min. : -2781 Min. : -265.76 Min. : -179.26 Min. : 0.22   
 1st Qu.: 1578 1st Qu.: 68.88 1st Qu.: 6.60 1st Qu.: 304.69   
 Median : 4744 Median : 364.57 Median : 36.26 Median : 1176.42   
 Mean : 7032 Mean : 3384.18 Mean : 1760.10 Mean : 3292.94   
 3rd Qu.: 8923 3rd Qu.: 2153.35 3rd Qu.: 163.15 3rd Qu.: 3310.01   
 Max. :103185 Max. :104519.54 Max. :149483.31 Max. :67474.85   
 NA's :4158 NA's :5269 NA's :4577 NA's :4726   
 MarkDown5 CPI Unemployment IsHoliday   
 Min. : -185.2 Min. :126.1 Min. : 3.684 Mode :logical   
 1st Qu.: 1440.8 1st Qu.:132.4 1st Qu.: 6.634 FALSE:7605   
 Median : 2727.1 Median :182.8 Median : 7.806 TRUE :585   
 Mean : 4132.2 Mean :172.5 Mean : 7.827   
 3rd Qu.: 4832.6 3rd Qu.:213.9 3rd Qu.: 8.567   
 Max. :771448.1 Max. :229.0 Max. :14.313   
 NA's :4140 NA's :585 NA's :585

Features has a different problem; several missing values.

How relevant are they and how will we handle them?

#If summary has too many variables you can select specific variables using select(var1:var5) OR de-selecting specific variables select(-c(var1:var5)) for a specific dataset.   
#c() indicates a group of variables, we are applying the - to all the group   
#arguments are select(data,vars)  
  
  
summary(select(features,MarkDown1:MarkDown5))

MarkDown1 MarkDown2 MarkDown3 MarkDown4   
 Min. : -2781 Min. : -265.76 Min. : -179.26 Min. : 0.22   
 1st Qu.: 1578 1st Qu.: 68.88 1st Qu.: 6.60 1st Qu.: 304.69   
 Median : 4744 Median : 364.57 Median : 36.26 Median : 1176.42   
 Mean : 7032 Mean : 3384.18 Mean : 1760.10 Mean : 3292.94   
 3rd Qu.: 8923 3rd Qu.: 2153.35 3rd Qu.: 163.15 3rd Qu.: 3310.01   
 Max. :103185 Max. :104519.54 Max. :149483.31 Max. :67474.85   
 NA's :4158 NA's :5269 NA's :4577 NA's :4726   
 MarkDown5   
 Min. : -185.2   
 1st Qu.: 1440.8   
 Median : 2727.1   
 Mean : 4132.2   
 3rd Qu.: 4832.6   
 Max. :771448.1   
 NA's :4140

summary(select(features,-c(MarkDown1:MarkDown5)))

Store Date Temperature Fuel\_Price   
 Min. : 1 Length:8190 Min. : -7.29 Min. :2.472   
 1st Qu.:12 Class :character 1st Qu.: 45.90 1st Qu.:3.041   
 Median :23 Mode :character Median : 60.71 Median :3.513   
 Mean :23 Mean : 59.36 Mean :3.406   
 3rd Qu.:34 3rd Qu.: 73.88 3rd Qu.:3.743   
 Max. :45 Max. :101.95 Max. :4.468   
   
 CPI Unemployment IsHoliday   
 Min. :126.1 Min. : 3.684 Mode :logical   
 1st Qu.:132.4 1st Qu.: 6.634 FALSE:7605   
 Median :182.8 Median : 7.806 TRUE :585   
 Mean :172.5 Mean : 7.827   
 3rd Qu.:213.9 3rd Qu.: 8.567   
 Max. :229.0 Max. :14.313   
 NA's :585 NA's :585

# %>% creates a pipe. It applies a command to what was there before. Can read it as 'then pass it to'  
#It makes your code easier to read  
  
#features%>%select(MarkDown1:MarkDown5)%>%summary()  
#features%>%select(-c(MarkDown1:MarkDown5))%>%summary()

### **Handling Missing Values**

Note how CPI and Unemployment have a few NA values, but Markdown on the other hand has a lot.

# We'll drop all Markdown values with select( - ...)   
# and replace NA values with the mean of Unemployment and CPI.  
features2<-features%>%select(-c(MarkDown1:MarkDown5))%>%   
 mutate(Unemployment = replace\_na(Unemployment,mean(Unemployment, na.rm = TRUE)),  
 CPI = replace\_na(CPI,mean(CPI, na.rm = TRUE)))  
  
summary(features2)

Store Date Temperature Fuel\_Price   
 Min. : 1 Length:8190 Min. : -7.29 Min. :2.472   
 1st Qu.:12 Class :character 1st Qu.: 45.90 1st Qu.:3.041   
 Median :23 Mode :character Median : 60.71 Median :3.513   
 Mean :23 Mean : 59.36 Mean :3.406   
 3rd Qu.:34 3rd Qu.: 73.88 3rd Qu.:3.743   
 Max. :45 Max. :101.95 Max. :4.468   
 CPI Unemployment IsHoliday   
 Min. :126.1 Min. : 3.684 Mode :logical   
 1st Qu.:132.7 1st Qu.: 6.833 FALSE:7605   
 Median :172.5 Median : 7.827 TRUE :585   
 Mean :172.5 Mean : 7.827   
 3rd Qu.:212.8 3rd Qu.: 8.519   
 Max. :229.0 Max. :14.313

We have 5 variables: Store, Department, Date, Weekly Sales and IsHoliday. Notice they have different data types. Integers are numbers, Characters are text, dbl are numbers with decimal points and lgl stands for logical; can be False or True.

## D. Enriching

To enrich our sales dataset, we can merge it with the other two; features and stores.

Features provide additional information on the date and location for each store.

Stores provide additional information (Type and Size) about the Store.

Now let’s merge all three datasets:

head(stores)

Store Type Size  
1 1 A 151315  
2 2 A 202307  
3 3 B 37392  
4 4 A 205863  
5 5 B 34875  
6 6 A 202505

head(features2)

Store Date Temperature Fuel\_Price CPI Unemployment IsHoliday  
1 1 2010-02-05 42.31 2.572 211.0964 8.106 FALSE  
2 1 2010-02-12 38.51 2.548 211.2422 8.106 TRUE  
3 1 2010-02-19 39.93 2.514 211.2891 8.106 FALSE  
4 1 2010-02-26 46.63 2.561 211.3196 8.106 FALSE  
5 1 2010-03-05 46.50 2.625 211.3501 8.106 FALSE  
6 1 2010-03-12 57.79 2.667 211.3806 8.106 FALSE

#merging as a command is easy, you just select the datasets to merge and by which variables  
#what requires some thought is identifying the key variables you want to merge the dataset by.  
#Those should be common values between both datasets that make for unique combinations  
storeandfeatures<-merge(stores,features2,by="Store")  
head(storeandfeatures)

Store Type Size Date Temperature Fuel\_Price CPI Unemployment  
1 1 A 151315 2010-02-05 42.31 2.572 211.0964 8.106  
2 1 A 151315 2010-02-12 38.51 2.548 211.2422 8.106  
3 1 A 151315 2010-02-19 39.93 2.514 211.2891 8.106  
4 1 A 151315 2010-02-26 46.63 2.561 211.3196 8.106  
5 1 A 151315 2010-03-05 46.50 2.625 211.3501 8.106  
6 1 A 151315 2010-03-12 57.79 2.667 211.3806 8.106  
 IsHoliday  
1 FALSE  
2 TRUE  
3 FALSE  
4 FALSE  
5 FALSE  
6 FALSE

head(sales2)

# A tibble: 6 × 4  
# Groups: Store [1]  
 Store Date Weekly\_Sales IsHoliday  
 <int> <date> <dbl> <dbl>  
1 1 2010-02-05 1643691. 0  
2 1 2010-02-12 1641957. 1  
3 1 2010-02-19 1611968. 0  
4 1 2010-02-26 1409728. 0  
5 1 2010-03-05 1554807. 0  
6 1 2010-03-12 1439542. 0

df0<-merge(sales2,storeandfeatures,by=c("Date","Store"))  
  
summary(df0)

Date Store Weekly\_Sales IsHoliday.x   
 Min. :2010-02-05 Min. : 1 Min. : 209986 Min. :0.00000   
 1st Qu.:2010-10-08 1st Qu.:12 1st Qu.: 553350 1st Qu.:0.00000   
 Median :2011-06-17 Median :23 Median : 960746 Median :0.00000   
 Mean :2011-06-17 Mean :23 Mean :1046965 Mean :0.06993   
 3rd Qu.:2012-02-24 3rd Qu.:34 3rd Qu.:1420159 3rd Qu.:0.00000   
 Max. :2012-10-26 Max. :45 Max. :3818686 Max. :1.00000   
 Type Size Temperature Fuel\_Price   
 Length:6435 Min. : 34875 Min. : -2.06 Min. :2.472   
 Class :character 1st Qu.: 70713 1st Qu.: 47.46 1st Qu.:2.933   
 Mode :character Median :126512 Median : 62.67 Median :3.445   
 Mean :130288 Mean : 60.66 Mean :3.359   
 3rd Qu.:202307 3rd Qu.: 74.94 3rd Qu.:3.735   
 Max. :219622 Max. :100.14 Max. :4.468   
 CPI Unemployment IsHoliday.y   
 Min. :126.1 Min. : 3.879 Mode :logical   
 1st Qu.:131.7 1st Qu.: 6.891 FALSE:5985   
 Median :182.6 Median : 7.874 TRUE :450   
 Mean :171.6 Mean : 7.999   
 3rd Qu.:212.7 3rd Qu.: 8.622   
 Max. :227.2 Max. :14.313

#Note how IsHoliday is duplicate, because it appeared in both datasets and was not identified as a key variable. To correct, we can remove one and rename the other.   
  
#there are two options - un-comment one only:  
#df<-df0%>%select(-(IsHoliday.y))%>%rename(IsHoliday=IsHoliday.x) # keep as an number  
df<-df0%>%select(-(IsHoliday.x))%>%rename(IsHoliday=IsHoliday.y) #keep as a logical  
  
summary(df)

Date Store Weekly\_Sales Type   
 Min. :2010-02-05 Min. : 1 Min. : 209986 Length:6435   
 1st Qu.:2010-10-08 1st Qu.:12 1st Qu.: 553350 Class :character   
 Median :2011-06-17 Median :23 Median : 960746 Mode :character   
 Mean :2011-06-17 Mean :23 Mean :1046965   
 3rd Qu.:2012-02-24 3rd Qu.:34 3rd Qu.:1420159   
 Max. :2012-10-26 Max. :45 Max. :3818686   
 Size Temperature Fuel\_Price CPI   
 Min. : 34875 Min. : -2.06 Min. :2.472 Min. :126.1   
 1st Qu.: 70713 1st Qu.: 47.46 1st Qu.:2.933 1st Qu.:131.7   
 Median :126512 Median : 62.67 Median :3.445 Median :182.6   
 Mean :130288 Mean : 60.66 Mean :3.359 Mean :171.6   
 3rd Qu.:202307 3rd Qu.: 74.94 3rd Qu.:3.735 3rd Qu.:212.7   
 Max. :219622 Max. :100.14 Max. :4.468 Max. :227.2   
 Unemployment IsHoliday   
 Min. : 3.879 Mode :logical   
 1st Qu.: 6.891 FALSE:5985   
 Median : 7.874 TRUE :450   
 Mean : 7.999   
 3rd Qu.: 8.622   
 Max. :14.313

Let’s look at our new nice dataset, both for store10 and for the other stores:

# Use == which is a logical operator, it will only return values for when Store is equal to 10  
print("Store 10 Weekly Sales Summary Statistics")

[1] "Store 10 Weekly Sales Summary Statistics"

df%>%filter(Store==10)%>%summary()

Date Store Weekly\_Sales Type   
 Min. :2010-02-05 Min. :10 Min. :1627707 Length:143   
 1st Qu.:2010-10-11 1st Qu.:10 1st Qu.:1740771 Class :character   
 Median :2011-06-17 Median :10 Median :1827522 Mode :character   
 Mean :2011-06-17 Mean :10 Mean :1899425   
 3rd Qu.:2012-02-20 3rd Qu.:10 3rd Qu.:1944111   
 Max. :2012-10-26 Max. :10 Max. :3749058   
 Size Temperature Fuel\_Price CPI   
 Min. :126512 Min. :42.58 Min. :2.825 Min. :126.1   
 1st Qu.:126512 1st Qu.:59.06 1st Qu.:3.107 1st Qu.:126.6   
 Median :126512 Median :73.56 Median :3.675 Median :129.1   
 Mean :126512 Mean :72.24 Mean :3.576 Mean :128.7   
 3rd Qu.:126512 3rd Qu.:85.11 3rd Qu.:3.905 3rd Qu.:130.5   
 Max. :126512 Max. :95.36 Max. :4.468 Max. :131.2   
 Unemployment IsHoliday   
 Min. :6.943 Mode :logical   
 1st Qu.:7.545 FALSE:133   
 Median :8.494 TRUE :10   
 Mean :8.363   
 3rd Qu.:9.003   
 Max. :9.765

print("") # vertical spacing :-)

[1] ""

cat("\n") # vertical spacing, 2nd approach :-)

writeLines("\n") # vertical spacing, third approach :-)

print("All other stores Weekly Sales Summary Statistics")

[1] "All other stores Weekly Sales Summary Statistics"

# now use != which is a different logical operator, it will return all results for Store when Store is NOT equal to 10  
df%>%filter(Store!=10)%>%summary()

Date Store Weekly\_Sales Type   
 Min. :2010-02-05 Min. : 1.00 Min. : 209986 Length:6292   
 1st Qu.:2010-10-08 1st Qu.:12.75 1st Qu.: 546687 Class :character   
 Median :2011-06-17 Median :23.50 Median : 948637 Mode :character   
 Mean :2011-06-17 Mean :23.30 Mean :1027591   
 3rd Qu.:2012-02-24 3rd Qu.:34.25 3rd Qu.:1391634   
 Max. :2012-10-26 Max. :45.00 Max. :3818686   
 Size Temperature Fuel\_Price CPI   
 Min. : 34875 Min. : -2.06 Min. :2.472 Min. :126.1   
 1st Qu.: 67334 1st Qu.: 47.11 1st Qu.:2.921 1st Qu.:132.4   
 Median :126970 Median : 62.51 Median :3.439 Median :185.9   
 Mean :130373 Mean : 60.40 Mean :3.354 Mean :172.6   
 3rd Qu.:202356 3rd Qu.: 74.66 3rd Qu.:3.732 3rd Qu.:213.0   
 Max. :219622 Max. :100.14 Max. :4.468 Max. :227.2   
 Unemployment IsHoliday   
 Min. : 3.879 Mode :logical   
 1st Qu.: 6.891 FALSE:5852   
 Median : 7.872 TRUE :440   
 Mean : 7.991   
 3rd Qu.: 8.595   
 Max. :14.313

We can also create a dataset with the three merged datasets, to avoid having to load and merge and clean again next time:

## F. Publishing

summary(df)

Date Store Weekly\_Sales Type   
 Min. :2010-02-05 Min. : 1 Min. : 209986 Length:6435   
 1st Qu.:2010-10-08 1st Qu.:12 1st Qu.: 553350 Class :character   
 Median :2011-06-17 Median :23 Median : 960746 Mode :character   
 Mean :2011-06-17 Mean :23 Mean :1046965   
 3rd Qu.:2012-02-24 3rd Qu.:34 3rd Qu.:1420159   
 Max. :2012-10-26 Max. :45 Max. :3818686   
 Size Temperature Fuel\_Price CPI   
 Min. : 34875 Min. : -2.06 Min. :2.472 Min. :126.1   
 1st Qu.: 70713 1st Qu.: 47.46 1st Qu.:2.933 1st Qu.:131.7   
 Median :126512 Median : 62.67 Median :3.445 Median :182.6   
 Mean :130288 Mean : 60.66 Mean :3.359 Mean :171.6   
 3rd Qu.:202307 3rd Qu.: 74.94 3rd Qu.:3.735 3rd Qu.:212.7   
 Max. :219622 Max. :100.14 Max. :4.468 Max. :227.2   
 Unemployment IsHoliday   
 Min. : 3.879 Mode :logical   
 1st Qu.: 6.891 FALSE:5985   
 Median : 7.874 TRUE :450   
 Mean : 7.999   
 3rd Qu.: 8.622   
 Max. :14.313

df%>%write.csv("Supermarketdata.csv",row.names = FALSE)

## **E. Verifying / Exploring**

### First Graph

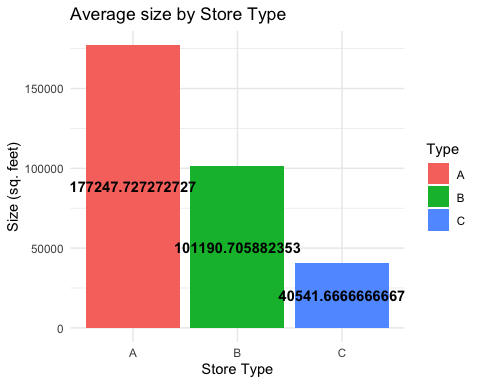
# recall the store types of A, B, and C and we can find the mean Size by Type  
table(df$Type)

A B C   
3146 2431 858

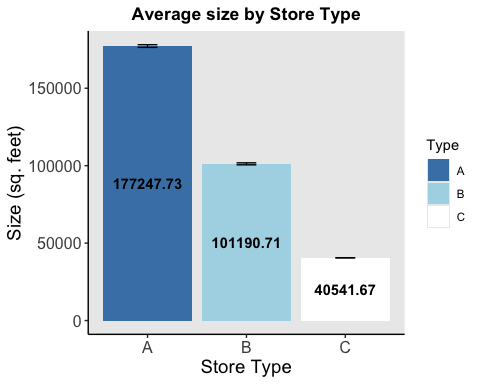
a0<-df%>%group\_by(Type)%>%summarize(SIZE=mean(Size))  
a0

# A tibble: 3 × 2  
 Type SIZE  
 <chr> <dbl>  
1 A 177248.  
2 B 101191.  
3 C 40542.

# This same coding is helpful for doing graphs:  
  
ggplot(a0) +  
 aes(x = Type, y = SIZE, fill = Type) +  
 geom\_bar(stat = "identity") +  
 ggtitle("Average size by Store Type") +  
 xlab("Store Type") +  
 ylab("Size (sq. feet)") +  
 theme\_minimal() +  
 geom\_text(aes(label = SIZE), position = position\_stack(vjust = 0.5),  
 fontface = "bold") +  
 scale\_color\_brewer(palette = 'Set1')



#Same graph but nicer. look for online sources (e.g., chatGPT ) for graphing format code   
  
ggplot(df%>%group\_by(Type)%>%summarize(SIZE=round(mean(Size),2),Obs=n(),SE\_SIZE=sd(Size)/sqrt(Obs))) +  
 aes(x = Type, y = SIZE, fill = Type) +  
 geom\_bar(stat = "identity") +  
 geom\_errorbar(aes(ymin = SIZE - SE\_SIZE, ymax = SIZE + SE\_SIZE),  
 width = 0.2, position = position\_dodge(0.9)) +  
 scale\_fill\_manual(values = c("steelblue", "lightblue", "white"))+  
 ggtitle("Average size by Store Type") +  
 xlab("Store Type") +  
 ylab("Size (sq. feet)") +  
 theme(panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 axis.line = element\_line(color = "black"),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 14)) +  
 geom\_text(aes(label = SIZE), position = position\_stack(vjust = 0.5),  
 fontface = "bold") +  
 scale\_color\_brewer(palette = 'Set1')



### Feature Engineering:

#Feature Engineering sounds complex but just consists of creating new variables that could be useful for analysis. Below two examples:  
df$Store10<-ifelse(df$Store==10,1,0)  
df$Sales\_by\_sqfeet<-df$Weekly\_Sales/df$Size

**Relationship between variables:**

### **Correlation Matrix:**

cor() only uses numeric variables, so the pipe has a step to include only numeric data

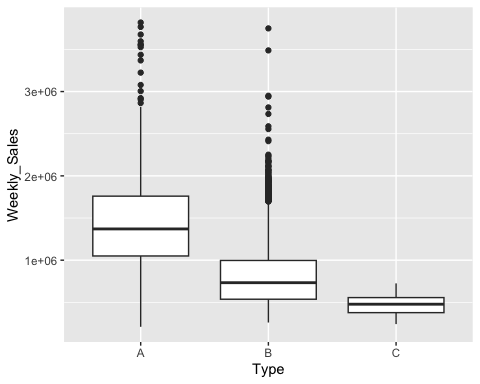
df%>%select\_if(is.numeric)%>%cor()

Store Weekly\_Sales Size Temperature Fuel\_Price  
Store 1.000000000 -0.335332015 -0.244269842 -0.02265908 0.060022955  
Weekly\_Sales -0.335332015 1.000000000 0.810468495 -0.06381001 0.009463786  
Size -0.244269842 0.810468495 1.000000000 -0.09248017 0.009053225  
Temperature -0.022659077 -0.063810013 -0.092480172 1.00000000 0.144981806  
Fuel\_Price 0.060022955 0.009463786 0.009053225 0.14498181 1.000000000  
CPI -0.209491930 -0.072634162 -0.009737309 0.17688768 -0.170641795  
Unemployment 0.223531274 -0.106176090 -0.089170140 0.10115786 -0.034683745  
Store10 -0.150904567 0.227729841 -0.009018761 0.09463278 0.071378638  
Sales\_by\_sqfeet -0.008094087 0.221365116 -0.333762936 0.10054739 -0.009613673  
 CPI Unemployment Store10 Sales\_by\_sqfeet  
Store -0.209491930 0.22353127 -0.150904567 -0.008094087  
Weekly\_Sales -0.072634162 -0.10617609 0.227729841 0.221365116  
Size -0.009737309 -0.08917014 -0.009018761 -0.333762936  
Temperature 0.176887676 0.10115786 0.094632782 0.100547389  
Fuel\_Price -0.170641795 -0.03468374 0.071378638 -0.009613673  
CPI 1.000000000 -0.30202006 -0.164336100 -0.023399676  
Unemployment -0.302020064 1.00000000 0.029240220 0.027711773  
Store10 -0.164336100 0.02924022 1.000000000 0.349380948  
Sales\_by\_sqfeet -0.023399676 0.02771177 0.349380948 1.000000000

### **Boxplot:**

A boxplot to get a sense of the distribution in weekly sales:

ggplot(df)+geom\_boxplot(mapping =aes(group=Type, x=Type, y=Weekly\_Sales))



### **A line graph for sales by date:**

#First, we have to group sales by store. Let's compare Store 10 vs all the other stores.  
  
  
a0<- df%>%group\_by(Store10,Date)%>%summarize(Weekly\_Sales=mean(Weekly\_Sales))

`summarise()` has grouped output by 'Store10'. You can override using the  
`.groups` argument.

a0

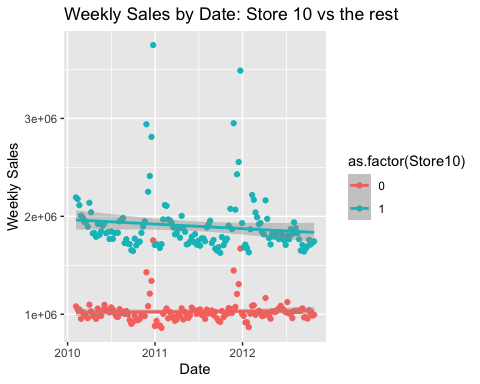
# A tibble: 286 × 3  
# Groups: Store10 [2]  
 Store10 Date Weekly\_Sales  
 <dbl> <date> <dbl>  
 1 0 2010-02-05 1080857.  
 2 0 2010-02-12 1049106.  
 3 0 2010-02-19 1049172.  
 4 0 2010-02-26 953677.  
 5 0 2010-03-05 1020100.  
 6 0 2010-03-12 999638.  
 7 0 2010-03-19 978230.  
 8 0 2010-03-26 960010.  
 9 0 2010-04-02 1097390.  
10 0 2010-04-09 1030096.  
# ℹ 276 more rows

a0%>%filter(Date=='2010-02-05')

# A tibble: 2 × 3  
# Groups: Store10 [2]  
 Store10 Date Weekly\_Sales  
 <dbl> <date> <dbl>  
1 0 2010-02-05 1080857.  
2 1 2010-02-05 2193049.

ggplot(a0)+ aes(x = Date, y = Weekly\_Sales, color= as.factor(Store10)) +  
 geom\_point() + # Add data points with transparency  
 geom\_smooth(method = 'lm') + # Increase line width and add transparency to trendline  
 labs(x = "Date", y = "Weekly Sales") +  
 ggtitle("Weekly Sales by Date: Store 10 vs the rest")

`geom\_smooth()` using formula = 'y ~ x'



**And a nicer line-graph:**

a0<-df%>%filter(Type=='B')%>%group\_by(Store10,Date)%>%summarize(Weekly\_Sales=mean(Weekly\_Sales))

`summarise()` has grouped output by 'Store10'. You can override using the  
`.groups` argument.

a0

# A tibble: 286 × 3  
# Groups: Store10 [2]  
 Store10 Date Weekly\_Sales  
 <dbl> <date> <dbl>  
 1 0 2010-02-05 786403.  
 2 0 2010-02-12 766447.  
 3 0 2010-02-19 776663.  
 4 0 2010-02-26 703084.  
 5 0 2010-03-05 750098.  
 6 0 2010-03-12 736081.  
 7 0 2010-03-19 724819.  
 8 0 2010-03-26 731002.  
 9 0 2010-04-02 822564.  
10 0 2010-04-09 760332.  
# ℹ 276 more rows

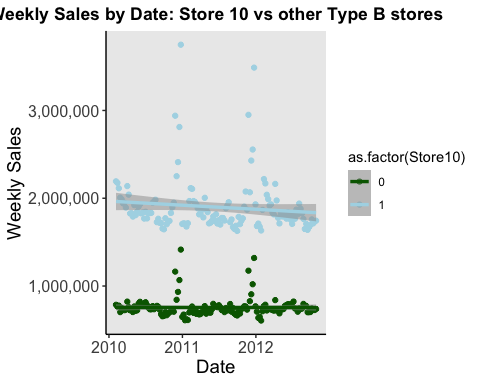
a0%>%filter(Date=='2010-02-05')

# A tibble: 2 × 3  
# Groups: Store10 [2]  
 Store10 Date Weekly\_Sales  
 <dbl> <date> <dbl>  
1 0 2010-02-05 786403.  
2 1 2010-02-05 2193049.

ggplot(a0)+ aes(x = Date, y = Weekly\_Sales, color= as.factor(Store10)) +  
 geom\_point() + # Add data points with transparency  
 geom\_smooth(method = 'lm', size = 1.2, alpha = 0.5,se=TRUE) + # Increase line width and add transparency to trendline  
 labs(x = "Date", y = "Weekly Sales") +  
 ggtitle("Weekly Sales by Date: Store 10 vs other Type B stores") +  
 scale\_color\_manual(values = c("darkgreen", "lightblue", "steelblue")) +  
 theme(panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.title = element\_text(hjust = 0.5, face = "bold"),  
 axis.line = element\_line(color = "black"),  
 axis.text = element\_text(size = 12),  
 axis.title = element\_text(size = 14))+  
 scale\_y\_continuous(labels = scales::comma) # Format y-axis labels with commas

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.

`geom\_smooth()` using formula = 'y ~ x'



What impressions do you get from Store 10 performance compared with the other Stores from these graphs?

What would be your ‘killer figure’ for your team’s presentation?

Which variables would you consider the most important when assessing Store 10 performance?

After all the data wrangling, do you think modeling will be easier and more effective or not? Why?

# maybe explicitly make Type a categorical variable with the as.factor() function  
#  
df2 <- df  
head(df2)

Date Store Weekly\_Sales Type Size Temperature Fuel\_Price CPI  
1 2010-02-05 1 1643691 A 151315 42.31 2.572 211.0964  
2 2010-02-05 10 2193049 B 126512 54.34 2.962 126.4421  
3 2010-02-05 11 1528009 A 207499 46.04 2.572 214.4249  
4 2010-02-05 12 1100046 B 112238 49.47 2.962 126.4421  
5 2010-02-05 13 1967221 A 219622 31.53 2.666 126.4421  
6 2010-02-05 14 2623470 A 200898 27.31 2.784 181.8712  
 Unemployment IsHoliday Store10 Sales\_by\_sqfeet  
1 8.106 FALSE 0 10.862710  
2 9.765 FALSE 1 17.334709  
3 7.368 FALSE 0 7.363933  
4 13.975 FALSE 0 9.801015  
5 8.316 FALSE 0 8.957302  
6 8.992 FALSE 0 13.058716

df2$Type = as.factor( df2$Type)  
head(df2)

Date Store Weekly\_Sales Type Size Temperature Fuel\_Price CPI  
1 2010-02-05 1 1643691 A 151315 42.31 2.572 211.0964  
2 2010-02-05 10 2193049 B 126512 54.34 2.962 126.4421  
3 2010-02-05 11 1528009 A 207499 46.04 2.572 214.4249  
4 2010-02-05 12 1100046 B 112238 49.47 2.962 126.4421  
5 2010-02-05 13 1967221 A 219622 31.53 2.666 126.4421  
6 2010-02-05 14 2623470 A 200898 27.31 2.784 181.8712  
 Unemployment IsHoliday Store10 Sales\_by\_sqfeet  
1 8.106 FALSE 0 10.862710  
2 9.765 FALSE 1 17.334709  
3 7.368 FALSE 0 7.363933  
4 13.975 FALSE 0 9.801015  
5 8.316 FALSE 0 8.957302  
6 8.992 FALSE 0 13.058716

str(df2$Type)

Factor w/ 3 levels "A","B","C": 1 2 1 2 1 1 2 2 2 2 ...