Week 3 Summary

Carson

Table of contents

Tuesday, Jan 24

❗ TIL

Today, I learnt the following concepts in class:

1. Git
2. Weekly Summary Template
3. read.csv
4. dplyr
5. ggplot2

```
library(dplyr)
library(purrr)
library(ggplot2)
```

```r
data_hard_code <- data.frame(
  Name = c("Denise", "Hamm", "Kronkite"),
  Age = c(9, 33, 67),
  Height = c(5.6, 3.5, 9.9)
)
data_hard_code
```

```
      Name Age Height
1   Denise   9    5.6
2     Hamm  33    3.5
3 Kronkite  67    9.9
```

CSV from memory

```r
file_location <- "C:\\Users\\carso\\stat380week3csv.txt"
```

```r
csv_data <- read.csv(file_location)
```

```
Warning in read.table(file = file, header = header, sep = sep, quote = quote, :
incomplete final line found by readTableHeader on
'C:\Users\carso\stat380week3csv.txt'
```

```r
csv_data
```

```
                        album year              band
1 I'm Wide Awake It's Morning 2005       Bright Eyes
2                           Z 2005 My Morning Jacket
3                       Grace 1994      Jeff Buckley
```

read_csv from Tidyverse

data.table from R package

Begin analysis:

dplyr and ggplot to facilitate Exploratory Data Analysis

dplyr

Provides set of verbs for manipulating data

We'll take the Cars(mpg) dataset

```
head(mpg, 10) %>% knitr::kable()
```

manufacturer	model	displ	year	cyl	trans	drv	cty	hwy	fl	class
audi	a4	1.8	1999	4	auto(l5)	f	18	29	p	compact
audi	a4	1.8	1999	4	manual(m5)	f	21	29	p	compact
audi	a4	2.0	2008	4	manual(m6)	f	20	31	p	compact
audi	a4	2.0	2008	4	auto(av)	f	21	30	p	compact
audi	a4	2.8	1999	6	auto(l5)	f	16	26	p	compact
audi	a4	2.8	1999	6	manual(m5)	f	18	26	p	compact
audi	a4	3.1	2008	6	auto(av)	f	18	27	p	compact
audi	a4 quattro	1.8	1999	4	manual(m5)	4	18	26	p	compact
audi	a4 quattro	1.8	1999	4	auto(l5)	4	16	25	p	compact
audi	a4 quattro	2.0	2008	4	manual(m6)	4	20	28	p	compact

What makes a data set clean?

Eliminating 'NULL' and 'NA' and 'NaN' and 'missing' entries

Unifying data types for var values

Tidy, each row corresponds to one case (every var has its own column, every obs has its own row, every cell has unique value)

Select: Selects specified existing columns to present

```
mpg %>%
  select(c(manufacturer, model))
```

```
# A tibble: 234 x 2
   manufacturer model
   <chr>        <chr>
 1 audi         a4
 2 audi         a4
 3 audi         a4
 4 audi         a4
 5 audi         a4
 6 audi         a4
 7 audi         a4
 8 audi         a4 quattro
 9 audi         a4 quattro
10 audi         a4 quattro
# ... with 224 more rows
```

Mutate: Creates new columns

```
mpg%>%
  mutate(cty * hwy)
```

```
# A tibble: 234 x 12
   manufac~1 model displ  year   cyl trans drv     cty   hwy fl    class cty *~2
   <chr>     <chr> <dbl> <int> <int> <chr> <chr> <int> <int> <chr> <chr>   <int>
 1 audi      a4      1.8  1999     4 auto~ f        18    29 p     comp~     522
 2 audi      a4      1.8  1999     4 manu~ f        21    29 p     comp~     609
 3 audi      a4      2    2008     4 manu~ f        20    31 p     comp~     620
 4 audi      a4      2    2008     4 auto~ f        21    30 p     comp~     630
 5 audi      a4      2.8  1999     6 auto~ f        16    26 p     comp~     416
 6 audi      a4      2.8  1999     6 manu~ f        18    26 p     comp~     468
 7 audi      a4      3.1  2008     6 auto~ f        18    27 p     comp~     486
 8 audi      a4 q~   1.8  1999     4 manu~ 4        18    26 p     comp~     468
 9 audi      a4 q~   1.8  1999     4 auto~ 4        16    25 p     comp~     400
10 audi      a4 q~   2    2008     4 manu~ 4        20    28 p     comp~     560
# ... with 224 more rows, and abbreviated variable names 1: manufacturer,
#   2: `cty * hwy`
```

Filter = Only presents rows with given

```
mpg %>%
  filter(model == 'a4')
```

```
# A tibble: 7 x 11
  manufacturer model displ  year   cyl trans      drv     cty   hwy fl    class
  <chr>        <chr> <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
1 audi         a4      1.8  1999     4 auto(l5)   f        18    29 p     compa~
2 audi         a4      1.8  1999     4 manual(m5) f        21    29 p     compa~
```

```
3 audi         a4      2    2008     4 manual(m6) f        20    31 p     compa~
4 audi         a4      2    2008     4 auto(av)   f        21    30 p     compa~
5 audi         a4      2.8  1999     6 auto(l5)   f        16    26 p     compa~
6 audi         a4      2.8  1999     6 manual(m5) f        18    26 p     compa~
7 audi         a4      3.1  2008     6 auto(av)   f        18    27 p     compa~
```

Others:

- summary
- mutate
- melt
- reshape

ggplot2

gg stands for Grammar of Graphics

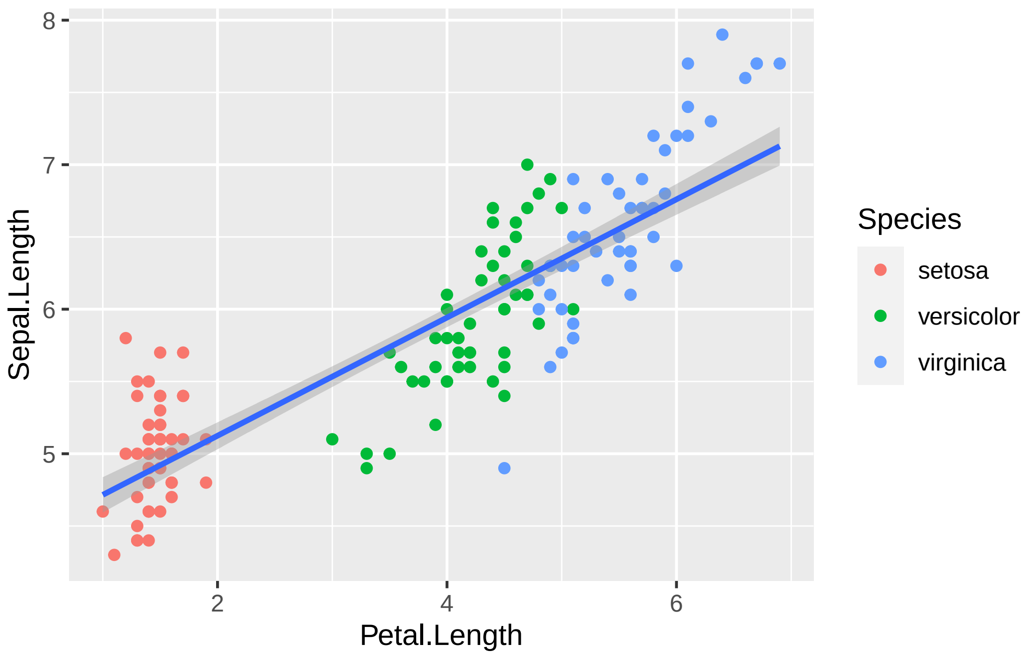
There is no ggplot1

```r
plt <- ggplot(iris)
```

Add points

```r
plt + geom_point(
  aes(x=Petal.Length, y=Sepal.Length, colour=Species)
) +
  #Trend line
geom_smooth(
  aes(x=Petal.Length, y=Sepal.Length),
  method = lm
)
```

```
`geom_smooth()` using formula = 'y ~ x'
```



Thursday, Jan 19

❗ TIL

Today, I learnt the following concepts in class:

1. Project Overview/Course Info
2. ggthemeassist
3. purrr

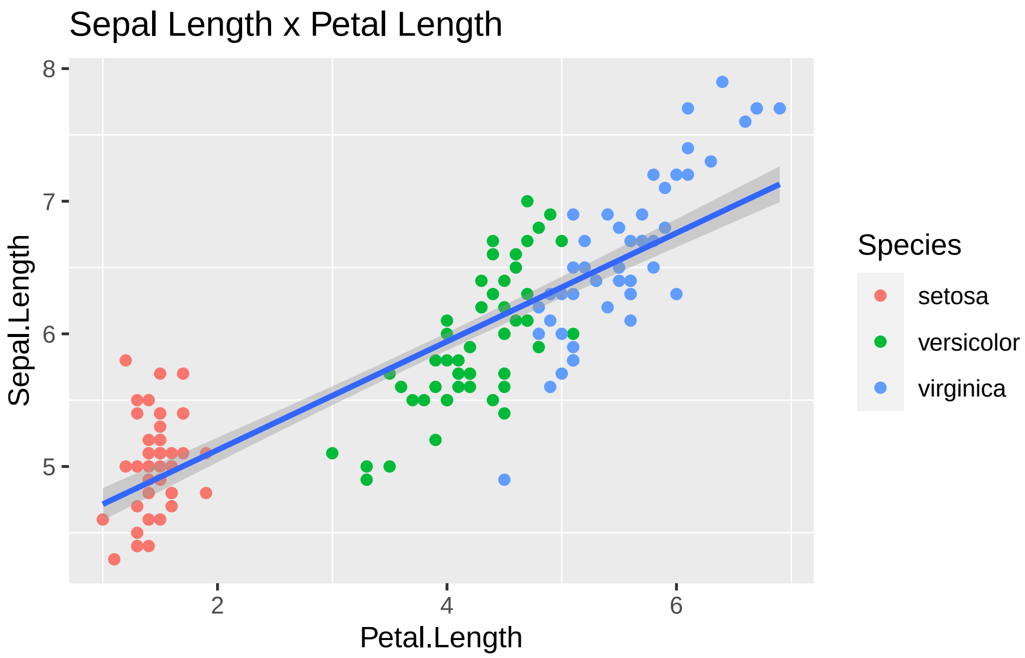
ggThemeAssist

```r
library(ggThemeAssist)
```

```
Warning: package 'ggThemeAssist' was built under R version 4.1.3
```

```r
plt + geom_point(
  aes(x=Petal.Length, y=Sepal.Length, colour=Species)
) +
  #Trend line
geom_smooth(
  aes(x=Petal.Length, y=Sepal.Length),
  method = lm
) + theme(panel.grid.major = element_line(colour = "ivory",
    linetype = "blank"), panel.background = element_rect(linetype = "solid")) +labs(title
```

```
`geom_smooth()` using formula = 'y ~ x'
```



More on data types

1. Char ("This is our last goodbye")
2. Int (45)
3. Dbl (89.4)
4. Booleans ('TRUE', 'FALSE')

Every column in dataframe has same data type

What are factors?

Factors are categorical variables

‘var’ contains the country code for people in North America:

```
var <- c(
  'USA',
  'USA',
  'CAN',
  'CAN',
  'CHI',
  'JPN',
  'MEX',
  'ARG',
  'SPN'
)
var
```

```
[1] "USA" "USA" "CAN" "CAN" "CHI" "JPN" "MEX" "ARG" "SPN"
```

To delineate categorical, we must specify:

```
as.factor(var)
```

```
[1] USA USA CAN CAN CHI JPN MEX ARG SPN
Levels: ARG CAN CHI JPN MEX SPN USA
```

```
iris$Species
```

```
  [1] setosa     setosa     setosa     setosa     setosa     setosa    
  [7] setosa     setosa     setosa     setosa     setosa     setosa    
 [13] setosa     setosa     setosa     setosa     setosa     setosa    
 [19] setosa     setosa     setosa     setosa     setosa     setosa    
 [25] setosa     setosa     setosa     setosa     setosa     setosa    
 [31] setosa     setosa     setosa     setosa     setosa     setosa    
 [37] setosa     setosa     setosa     setosa     setosa     setosa    
 [43] setosa     setosa     setosa     setosa     setosa     setosa    
 [49] setosa     setosa     versicolor versicolor versicolor versicolor
 [55] versicolor versicolor versicolor versicolor versicolor versicolor
 [61] versicolor versicolor versicolor versicolor versicolor versicolor
 [67] versicolor versicolor versicolor versicolor versicolor versicolor
 [73] versicolor versicolor versicolor versicolor versicolor versicolor
 [79] versicolor versicolor versicolor versicolor versicolor versicolor
 [85] versicolor versicolor versicolor versicolor versicolor versicolor
 [91] versicolor versicolor versicolor versicolor versicolor versicolor
 [97] versicolor versicolor versicolor versicolor virginica  virginica 
[103] virginica  virginica  virginica  virginica  virginica  virginica 
[109] virginica  virginica  virginica  virginica  virginica  virginica 
[115] virginica  virginica  virginica  virginica  virginica  virginica 
[121] virginica  virginica  virginica  virginica  virginica  virginica 
[127] virginica  virginica  virginica  virginica  virginica  virginica 
[133] virginica  virginica  virginica  virginica  virginica  virginica 
[139] virginica  virginica  virginica  virginica  virginica  virginica 
[145] virginica  virginica  virginica  virginica  virginica  virginica 
Levels: setosa versicolor virginica
```

We'll do 'class' for the mpg table

```
as.factor(mpg$class)
```

```
  [1] compact    compact    compact    compact    compact    compact   
  [7] compact    compact    compact    compact    compact    compact   
 [13] compact    compact    compact    midsize    midsize    midsize   
 [19] suv        suv        suv        suv        suv        2seater   
 [25] 2seater    2seater    2seater    2seater    suv        suv       
 [31] suv        suv        midsize    midsize    midsize    midsize   
 [37] midsize    minivan    minivan    minivan    minivan    minivan   
 [43] minivan    minivan    minivan    minivan    minivan    minivan   
 [49] pickup     pickup     pickup     pickup     pickup     pickup    
 [55] pickup     pickup     pickup     suv        suv        suv       
 [61] suv        suv        suv        suv        pickup     pickup    
 [67] pickup     pickup     pickup     pickup     pickup     pickup    
 [73] pickup     pickup     suv        suv        suv        suv       
 [79] suv        suv        suv        suv        suv        pickup    
 [85] pickup     pickup     pickup     pickup     pickup     pickup    
 [91] subcompact subcompact subcompact subcompact subcompact subcompact
 [97] subcompact subcompact subcompact subcompact subcompact subcompact
[103] subcompact subcompact subcompact subcompact subcompact subcompact
[109] midsize    midsize    midsize    midsize    midsize    midsize   
[115] midsize    subcompact subcompact subcompact subcompact subcompact
[121] subcompact subcompact suv        suv        suv        suv       
[127] suv        suv        suv        suv        suv        suv       
[133] suv        suv        suv        suv        suv        suv       
[139] suv        suv        suv        compact    compact    midsize   
[145] midsize    midsize    midsize    midsize    midsize    midsize   
```

```
[151] suv        suv        suv        suv        midsize    midsize   
[157] midsize    midsize    midsize    suv        suv        suv       
[163] suv        suv        suv        subcompact subcompact subcompact
[169] subcompact compact    compact    compact    compact    suv       
[175] suv        suv        suv        suv        suv        midsize   
[181] midsize    midsize    midsize    midsize    midsize    midsize   
[187] compact    compact    compact    compact    compact    compact   
[193] compact    compact    compact    compact    compact    compact   
[199] suv        suv        pickup     pickup     pickup     pickup    
[205] pickup     pickup     pickup     compact    compact    compact   
[211] compact    compact    compact    compact    compact    compact   
[217] compact    compact    compact    compact    compact    subcompact
[223] subcompact subcompact subcompact subcompact subcompact midsize   
[229] midsize    midsize    midsize    midsize    midsize    midsize   
Levels: 2seater compact midsize minivan pickup subcompact suv
```

‘Forcats’ is really useful here:

```
library(forcats)
```

```
Warning: package 'forcats' was built under R version 4.1.3
```

```
manufacturer <- as.factor(mpg$manufacturer)
#Minimum mileage as base level
fct_reorder(manufacturer, mpg$hwy, max)
```

```
  [1] audi       audi       audi       audi       audi       audi      
  [7] audi       audi       audi       audi       audi       audi      
 [13] audi       audi       audi       audi       audi       audi      
 [19] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet 
 [25] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet 
 [31] chevrolet  chevrolet  chevrolet  chevrolet  chevrolet  chevrolet 
 [37] chevrolet  dodge      dodge      dodge      dodge      dodge     
 [43] dodge      dodge      dodge      dodge      dodge      dodge     
 [49] dodge      dodge      dodge      dodge      dodge      dodge     
 [55] dodge      dodge      dodge      dodge      dodge      dodge     
 [61] dodge      dodge      dodge      dodge      dodge      dodge     
 [67] dodge      dodge      dodge      dodge      dodge      dodge     
 [73] dodge      dodge      ford       ford       ford       ford      
 [79] ford       ford       ford       ford       ford       ford      
 [85] ford       ford       ford       ford       ford       ford      
 [91] ford       ford       ford       ford       ford       ford      
 [97] ford       ford       ford       honda      honda      honda     
[103] honda      honda      honda      honda      honda      honda     
[109] hyundai    hyundai    hyundai    hyundai    hyundai    hyundai   
[115] hyundai    hyundai    hyundai    hyundai    hyundai    hyundai   
[121] hyundai    hyundai    jeep       jeep       jeep       jeep      
[127] jeep       jeep       jeep       jeep       land rover land rover
[133] land rover land rover lincoln    lincoln    lincoln    mercury   
[139] mercury    mercury    mercury    nissan     nissan     nissan    
[145] nissan     nissan     nissan     nissan     nissan     nissan    
[151] nissan     nissan     nissan     nissan     pontiac    pontiac   
[157] pontiac    pontiac    pontiac    subaru     subaru     subaru    
[163] subaru     subaru     subaru     subaru     subaru     subaru    
[169] subaru     subaru     subaru     subaru     subaru     toyota    
[175] toyota     toyota     toyota     toyota     toyota     toyota    
[181] toyota     toyota     toyota     toyota     toyota     toyota    
[187] toyota     toyota     toyota     toyota     toyota     toyota    
[193] toyota     toyota     toyota     toyota     toyota     toyota    
[199] toyota     toyota     toyota     toyota     toyota     toyota    
[205] toyota     toyota     toyota     volkswagen volkswagen volkswagen
[211] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
[217] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
[223] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
[229] volkswagen volkswagen volkswagen volkswagen volkswagen volkswagen
15 Levels: land rover lincoln mercury jeep dodge ford subaru ... volkswagen
```

'purrr'

Provides a set of functional programming tools

Consider the following procedure:

1. Filter 'iris' by species
2. Compute the 'Sepal.Area' as 'Sepal.Length' × 'Sepal.Width'
3. Find avg of 'Sepal.Area' for every flower in species

dplyr method:

```
iris %>%
  mutate(Area = Sepal.Length * Sepal.Length) %>%
  group_by(Species)%>%
    summarize(mean(Area))
```

```
# A tibble: 3 x 2
  Species    `mean(Area)`
  <fct>             <dbl>
1 setosa             25.2
2 versicolor         35.5
3 virginica          43.8
```

Consider the task: 1. Take number 'i' from 1 … 10 1. Create matrix with random entries of dimension 'i' x 'i' 1. Compute average of elements in matrix 1. Print

```
results <- c()
for (i in 1:10){
  M <- matrix(
    runif(i * i), nrow=i
  )
  results[i] <- mean(M)
}
results
```

```
 [1] 0.4043255 0.5341401 0.4970057 0.5838070 0.4618511 0.5013570 0.5342580
 [8] 0.4669383 0.5695291 0.5008722
```

$$i \to M_{i \times i} \to mean(M)$$

```
library(purrr)
map(
  1:10,
  function(i){
    mean(
      matrix(
        c(1:i*i), nrow=i
      )
    )
  }
)
```

```
[[1]]
[1] 1

[[2]]
[1] 3

[[3]]
[1] 6

[[4]]
[1] 10

[[5]]
[1] 15

[[6]]
[1] 21

[[7]]
[1] 28

[[8]]
[1] 36

[[9]]
[1] 45

[[10]]
[1] 55
```