Weekly Summary Week 3

Miranda Goodman

Table of contents

Tuesday, Jan 24

> **!** TIL
>
> Include a *very brief* summary of what you learnt in this class here.
> Today, I learnt the following concepts in class:
>
> 1. What dplyr does
> 2. Hard Coding a data frame into R
> 3. What does it mean to clean a data set

Provide more concrete details here. You can also use footenotes[1] if you like

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

1. In class we learned about what dplyr does: Objective is to provide a set of *"verbs"* for manipulating data pivot_longer, mutate, summary, filter, pivot_wider, reshape, gather, left_join, right_join, inner_join, outer_join
2. In class we learned how to hard code a data frame into R

[1]You can include some footnotes here

```r
data_hard_code <- data.frame(
  Name = c("Alice", "Bob", "Charlie"),
  Age = c(21, 25, 35),
  Height = c(5.5, 6.2, 5.9)
)
data_hard_code
```

```
     Name Age Height
1   Alice  21    5.5
2     Bob  25    6.2
3 Charlie  35    5.9
```

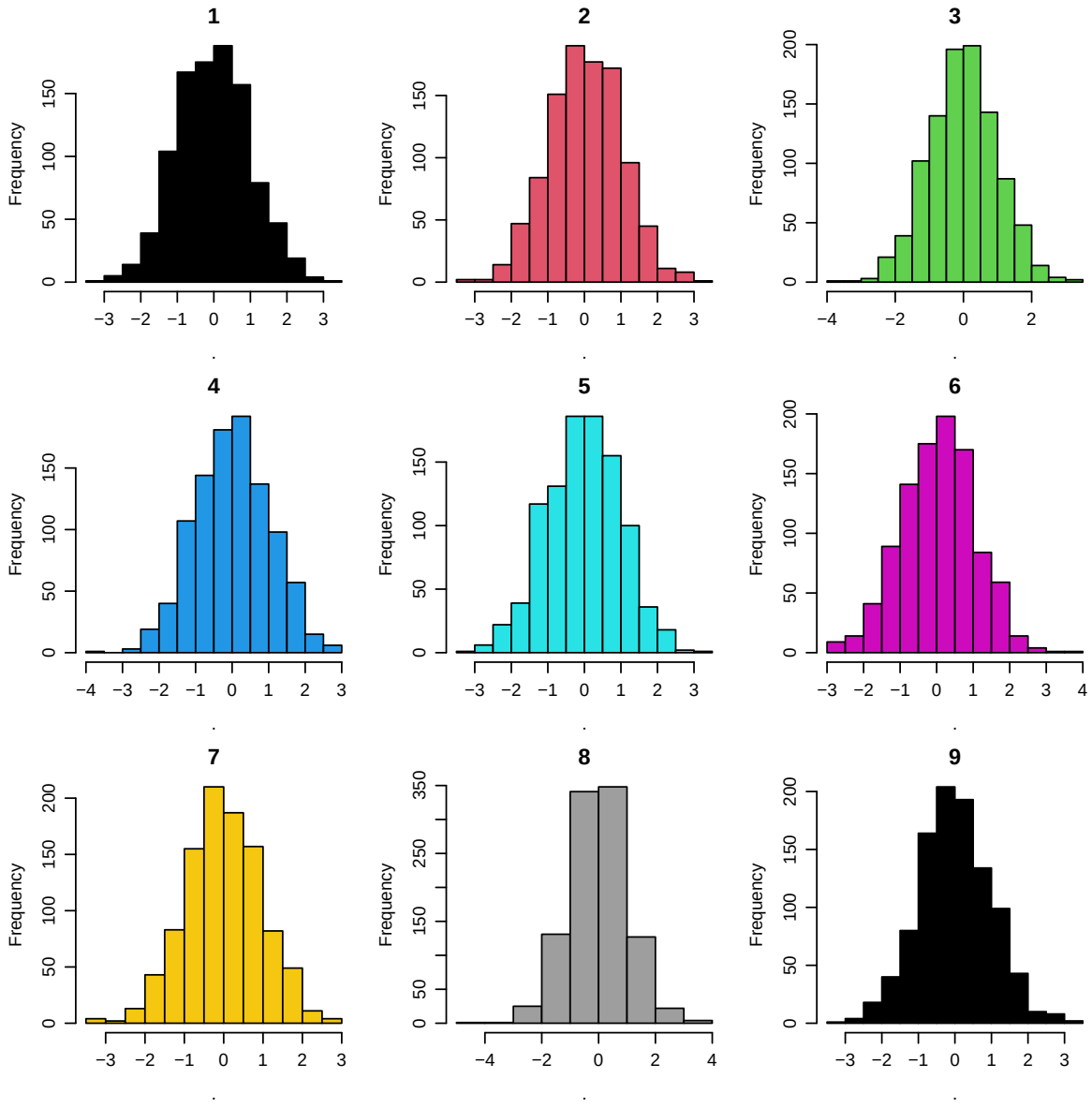
To read a csv file from memory we would do this:

```r
file_location <- "./data/data.csv"
data_from_csv <- read.csv()
```

1. In class we learned how to clean a data set. Cleaning data: what makes a data set clean? -missing data points are removed/NULL Data is Removed/NA Removed -Each row and column should correspond to the same value **Every variable should have its own column a variable is something which holds measurements** Every observation should have its own row **Every cell should have a unique value -Make sure all values for a particular variable are of the same 'data type' e.g., double or character or logical

For example: in class we learnt we learnt about the `map` function from the `purrr` package.

```r
par(mfrow=c(3, 3), mar=c(3.5, 3.5, 2, 1), mgp=c(2.4, 0.8, 0))
map(1:9, function(i)rnorm(1000) %>% hist(., main=i, col=i))
```


1
2
3
4
5
6
7
8
9
Frequency

Thursday, Jan 26

> **!** TIL
>
> Include a *very brief* summary of what you learnt in this class here.
> Today, I learnt the following concepts in class:
>
> 1. We went over the four data types.
> 2. We went over gg plot and learned how to add trend lines.
> 3. We learned about ggplot theme assist and ggplot2.

Provide more concrete details here, e.g.,

1. In class we went over the four data types. -string this is a character -Integers 1,2,3,4 -Double 2.2 -Booleans true/false

var contains the country code for people in north america

```
var <- c(
  "USA",
  "MEX"
)
var
```

```
[1] "USA" "MEX"
```

In order to specifically tell R this is categorical data use: as.factor(var)

1. In class we also reviewed ggplot and learned how to add trend lines

```
plt  <- ggplot(iris)
plt <- plt + geom_points(
  aes(x=Petal.Length, y=Sepal.Length,
      colour=Species)
)+
geom_smooth(
  aes(x=Petal.Length, y=Sepal.Length),
  method = lm
)
```

1. In class we learned about ggThemeAssist:

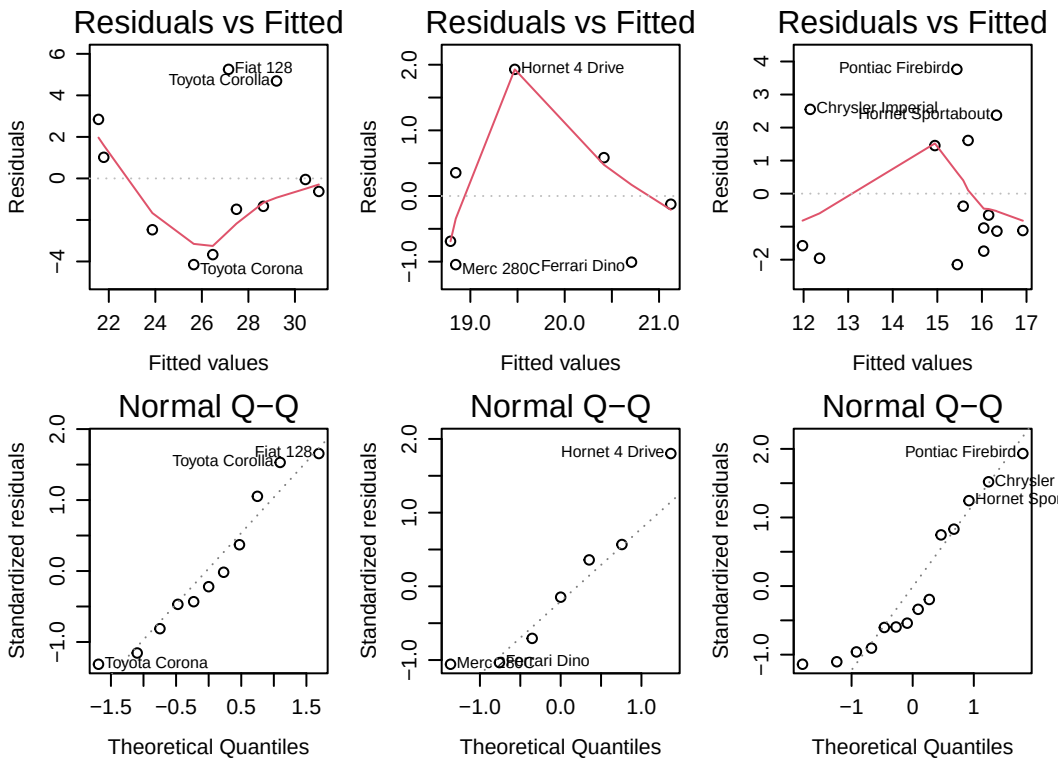
```
# library(ggThemeAssist)
library(ggplot2)
```

```
Warning: package 'ggplot2' was built under R version 4.1.3
```

```
ggplot(mpg) +
  geom_point(aes(x=displ, y=hwy)) +
  theme(panel.grid.major = element_line(linetype = "dotted"),
    plot.title = element_text(family = mono),
    plot.background = element_rect(fill = "white",
      linetype = "dashed")) +
  labs(title = "Plot Title",
       x = "x axis title", y = "y axis title")
```

In class we learnt how to use the `map` function to create multiple regression diagnostic plots

```
par(mfcol=c(2, 3), mar=c(3.5, 3.5, 2, 1), mgp=c(2.4, 0.8, 0))
mtcars %>%
  split(.$cyl) %>%
  map(~ lm(mpg ~ wt, data = .x)) %>%
  map(function(x)plot(x, which=c(1, 2)))
```


Residuals vs Fitted
Residuals
Fitted values
Fiat 128
Toyota Corolla
Toyota Corona
Hornet 4 Drive
Merc 280C
Ferrari Dino
Pontiac Firebird
Chrysler Imperial
Hornet Sportabout
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Fiat 128
Toyota Corolla
Toyota Corona
Hornet 4 Drive
Merc 280C
Ferrari Dino
Pontiac Firebird
Chrysler
Hornet Spor