Instructor: Nara Yoon, Ph.D. (yoonnx@jmu.edu)

Spring 2024 | Thursdays 4-7pm, Hart 3008

Data Analytics for Managerial Leadership

WK 9: Data structure II

PLAN FOR TODAY

Overview of core concepts

- Handling data quality issues
- Recoding/renaming variables

READINGS

Matloff, N. (2011). The art of R programming a tour of statistical software design. No Starch Press.

- Ch 6. Factors and tables (6.1 through 6.4)

Data quality issues

DATA QUALITY ISSUES

1. Outliers
2. Missing values
3. Duplicates

CHECK DATASET DESCRIPTION

New York Air Quality Measurements

Description

Daily air quality measurements in New York, May to September 1973.

Usage

```
airquality
```

Format

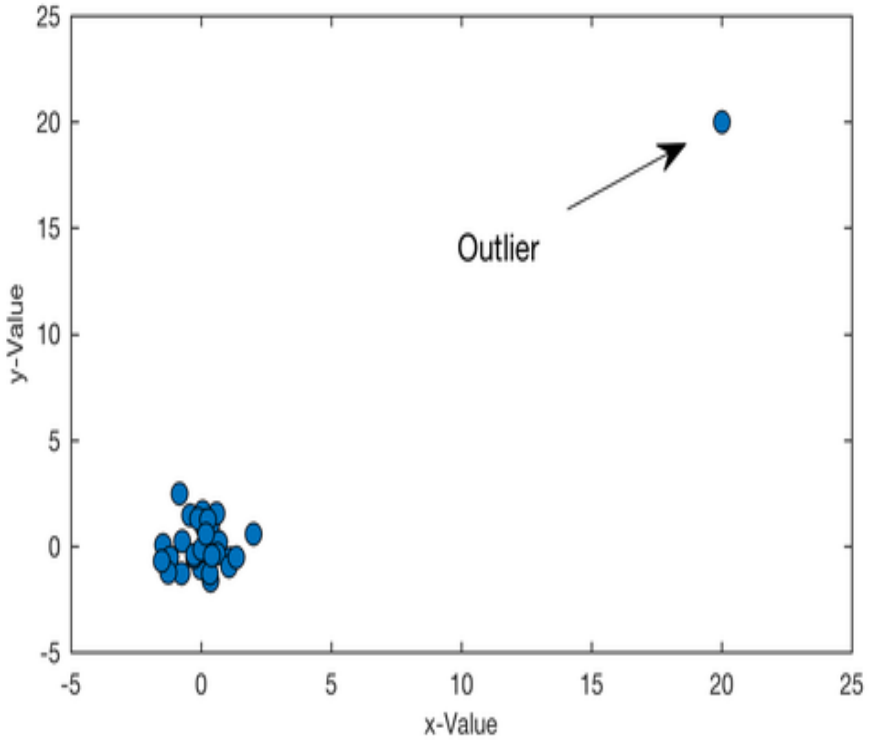
A data frame with 153 observations on 6 variables.

[,1]	Ozone	numeric Ozone (ppb)
[,2]	Solar.R	numeric Solar R (lang)
[,3]	Wind	numeric Wind (mph)
[,4]	Temp	numeric Temperature (degrees F)
[,5]	Month	numeric Month (1--12)
[,6]	Day	numeric Day of month (1--31)

1. Outliers

1. OUTLIERS

Observations whose characteristics are *considerably different* from most of the others in the dataset



2. Missing values

2. MISSING VALUES

Coded as “NA”

How do we know whether our dataset has missing cases?

is.na()

CHECKING MISSING CASES: is.na()

Where is missing record?

```
airquality_missing <- as.data.frame(airquality)
is.na(airquality_missing)
```

```
##       Ozone Solar.R  Wind  Temp Month   Day
##  [1,] FALSE   FALSE FALSE FALSE FALSE FALSE
##  [2,] FALSE   FALSE FALSE FALSE FALSE FALSE
##  [3,] FALSE   FALSE FALSE FALSE FALSE FALSE
##  [4,] FALSE   FALSE FALSE FALSE FALSE FALSE
##  [5,]  TRUE    TRUE FALSE FALSE FALSE FALSE
##  [6,] FALSE    TRUE FALSE FALSE FALSE FALSE
```

CHECKING MISSING CASES: is.na()

Number of missing records in whole dataset

```
colSums(is.na(airquality_df))
```

```
##   Ozone Solar.R    Wind    Temp   Month     Day
##      37       7       0       0       0       0
```

COUNTING # OF MISSING CASES

Number of missing cases in specific variable

```
length(which(is.na(airquality_missing$Solar.R)))
```

```
## [1] 7
```

So far, we looked at locating “missing” cases…

is.na()

Then, what if we want to focus on “complete” cases?

complete.cases()

CHECKING COMPLETE CASES: complete.cases()

```
airquality_df[complete.cases(airquality_df),]
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 7      23     299  8.6   65     5   7
```

COUNTING # OF COMPLETE CASES

Displays number of rows that are incomplete cases in airquality dataset

```
nrow(airquality_missing[!complete.cases(airquality_missing),])
```

```
## [1] 42
```

COUNTING # OF COMPLETE CASES

Displays number of rows that are complete cases in airquality dataset

```
nrow(airquality_missing[complete.cases(airquality_missing),])
```

```
## [1] 111
```

HANDLING MISSING DATA:
DROP OBSERVATIONS WITH MISSING VALUES

```
airquality_df_new <- na.omit(airquality_df)
airquality_df_new
```

```
##     Ozone Solar.R Wind Temp Month Day
## 1      41     190  7.4   67     5   1
## 2      36     118  8.0   72     5   2
## 3      12     149 12.6   74     5   3
## 4      18     313 11.5   62     5   4
## 7      23     299  8.6   65     5   7
```

3. Duplicates

3. DUPLICATES

Records that are redundant

duplicated() function

3. DUPLICATES

```
length(which(duplicated(airquality_missing)))
```

```
## [1] 0
```

```
airquality_missing_new4 <- airquality_missing[!duplicated(airquality_missing),]
```

Recoding variables: dplyr package

dplyr: COLLECTION OF USEFUL FUNCTIONS

mutate(): add new variables that are based upon existing variables

select(): select variables based on “variable names”

filter(): select observations based on “values”

summarise(): produces summary statistics

arrange(): reorders the rows by values of selected columns

Pipe operator %>%: structures sequence of operations

Recoding variables:
From “categorical” to “numerical”

CHECK DATASET DESCRIPTION

Edgar Anderson's Iris Data

Description

This famous (Fisher's or Anderson's) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

Usage

```
iris
iris3
```

Format

`iris` is a data frame with 150 cases (rows) and 5 variables (columns) named `Sepal.Length`, `Sepal.Width`, `Petal.Length`, `Petal.Width`, and `Species`.

LOAD DATASET (iris)

```
iris_new <- iris

str(iris_new)
```

```
## 'data.frame':    150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ..
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..:
```

COUNTING #OBSERVATIONS BY EACH CATEGORY

```
iris_new %>% group_by(Species) %>% summarise(count=n()) %>% arrange()
```

```
## # A tibble: 3 × 2
##   Species    count
##   <fct>      <int>
## 1 setosa        50
## 2 versicolor    50
## 3 virginica     50
```

RECODING VARIABLES: FROM “CATEGORICAL” TO “NUMERICAL”

```
iris_new %>% mutate(New_Cat = case_when(Species == "setosa" ~ 1,
                                        Species == "versicolor" ~ 2,
                                        Species == "virginica" ~ 3)) %>%
            select(Species,New_Cat ) %>%
            unique()
```

```
##         Species New_Cat
## 1        setosa       1
## 51   versicolor       2
## 101   virginica       3
```

Recoding variables:

From “numerical” to “categorical”

CHECK DATASET DESCRIPTION

Motor Trend Car Road Tests

Description

The data was extracted from the 1974 *Motor Trend* US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

Usage

```
mtcars
```

Format

A data frame with 32 observations on 11 (numeric) variables.

[, 1] mpg	Miles/(US) gallon
[, 2] cyl	Number of cylinders
[, 3] disp	Displacement (cu.in.)
[, 4] hp	Gross horsepower
[, 5] drat	Rear axle ratio
[, 6] wt	Weight (1000 lbs)
[, 7] qsec	1/4 mile time
[, 8] vs	Engine (0 = V-shaped, 1 = straight)
[, 9] am	Transmission (0 = automatic, 1 = manual)
[,10] gear	Number of forward gears
[,11] carb	Number of carburetors

LOAD DATASET (mtcars)

```
mtcars <- mtcars
```

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
```

DATA FRAME STRUCTURE (mtcars)

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1

COUNTING #OBS BY EACH GROUP IN THE VARIABLE

First, we want to arrange a variable that needs to be recoded ("cyl" variable)

```
mtcars %>% group_by(cyl) %>% summarise(count=n()) %>% arrange()
```

```
## # A tibble: 3 × 2
##     cyl count
##   <dbl> <int>
## 1     4    11
## 2     6     7
## 3     8    14
```

RECODING VARIABLE:
FROM “NUMERICAL” TO “CATEGORICAL”

```
mtcars %>% mutate(New_Cat = case_when(cyl == 6 ~ "6 Cylinders",
                                      cyl == 4 ~ "4 Cylinders",
                                      cyl == 8 ~ "8 Cylinders")) %>%
           rownames_to_column(var="Model_Name") %>%
           select(cyl, New_Cat) %>%
           unique()
```

```
##   cyl     New_Cat
## 1   6 6 Cylinders
## 3   4 4 Cylinders
## 5   8 8 Cylinders
```

Renaming & deleting variables

RENAMING VARIABLES: rename()

```
iris_rename <- iris
head(iris_rename)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

RENAMING VARIABLES: rename()

```
iris %>%
    rename(sep_length = Sepal.Length,
           sep_width = Sepal.Width,
           petal_length = Petal.Length,
           petal_width = Petal.Width)
```

```
##     sep_length sep_width petal_length petal_width    Species
## 1          5.1       3.5          1.4         0.2     setosa
## 2          4.9       3.0          1.4         0.2     setosa
## 3          4.7       3.2          1.3         0.2     setosa
## 4          4.6       3.1          1.5         0.2     setosa
## 5          5.0       3.6          1.4         0.2     setosa
```

DELETING VARIABLES: NULL

```
iris_null$Sepal.Width <- NULL
head(iris_null)
```

```
##   Sepal.Length Petal.Length Petal.Width Species
## 1          5.1          1.4         0.2  setosa
## 2          4.9          1.4         0.2  setosa
## 3          4.7          1.3         0.2  setosa
## 4          4.6          1.5         0.2  setosa
## 5          5.0          1.4         0.2  setosa
## 6          5.4          1.7         0.4  setosa
```