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

Spring 2024 | Thursdays 4-7pm, Hart 3008

Data Analytics for Managerial Leadership

WK 10: Data wrangling I

PLAN FOR TODAY

Overview of core concepts

- Data combining
- Subsetting
- Adding & sorting variable

CASE IN THE NEWS

Wing, J. M. (Oct 4, 2019). The data life cycle. Harvard Data Science Review.

Williamson, R. (Sep 30, 2020). Process and purpose, not thing and technique: How to pose data science research challenges. Harvard Data Science Review.

READINGS

Matloff, N. (2011).

- Ch 10. Input/output (10.1 through 10.2)

Long, J.D., & Teeter, P. (2019).

- Ch 4. Input and output

Baumer, B., Kaplan, D., & Horton, N

- Data wrangling (Ch 4.1 - 4.3)
- Data joins (Ch 5.1 - 5.2)

DATA WRANGLING: WHY?

To **reorganize** variable types

To **reshape** data

To **rearrange (expand or reduce)** data

To **add/delete** variables

To **summarize** data

DATA WRANGLING PACKAGES

Among many packages in the ***tidyverse***,

we mostly use:

- **dplyr** – to extract & summarize data

RECAP: dplyr


```
#install.packages("dplyr")
library(dplyr)
```

- filter() or select(): to subset the data
- mutate(): to create new variables
- arrange(): to reorder the rows (e.g., ascending, descending)
- group_by(): to split the data
- summarize(): to summarize information
- pipe operator %>% defines sequence of operations; much easier way of manipulating table

Data wrangling: Combining & subsetting

Data combining: join()

DATA MERGE: **join()**

join() functions in **dplyr** package

Caveat:

- Use **key variable(s)** that appear in both tables

COMBINING TWO TABLES: FOUR JOIN TYPES

Inner join – keep only *matched* records

Full join – keep *all* records

Left join – keep all *left* table records

Right join – keep all *right* table records

Exercise: join()

EXAMPLE DATA FRAMES

```
d1 <- tibble(col1=1:5, col2=c(1,2,3,2,1), col3=c(18,4,6,1,2))
d2 <- tibble(col2=c(2,5,1), col4=c(4,22,45))
```

```
d1
```

```
## # A tibble: 5 × 3
##    col1  col2  col3
##   <int> <dbl> <dbl>
## 1     1     1    18
## 2     2     2     4
## 3     3     3     6
## 4     4     2     1
## 5     5     1     2
```

```
d2
```

```
## # A tibble: 3 × 2
##    col2  col4
##   <dbl> <dbl>
## 1     2     4
## 2     5    22
## 3     1    45
```

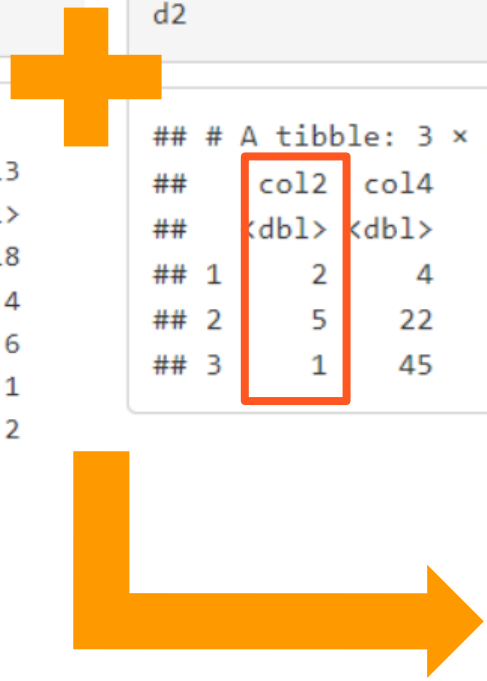
inner_join()

```
d1
```

```
## # A tibble: 5 × 3
##    col1  col2  col3
##   <int> <dbl> <dbl>
## 1     1     1    18
## 2     2     2     4
## 3     3     3     6
## 4     4     2     1
## 5     5     1     2
```

```
d2
```

```
## # A tibble: 3 × 2
##    col2  col4
##   <dbl> <dbl>
## 1     2     4
## 2     5    22
## 3     1    45
```



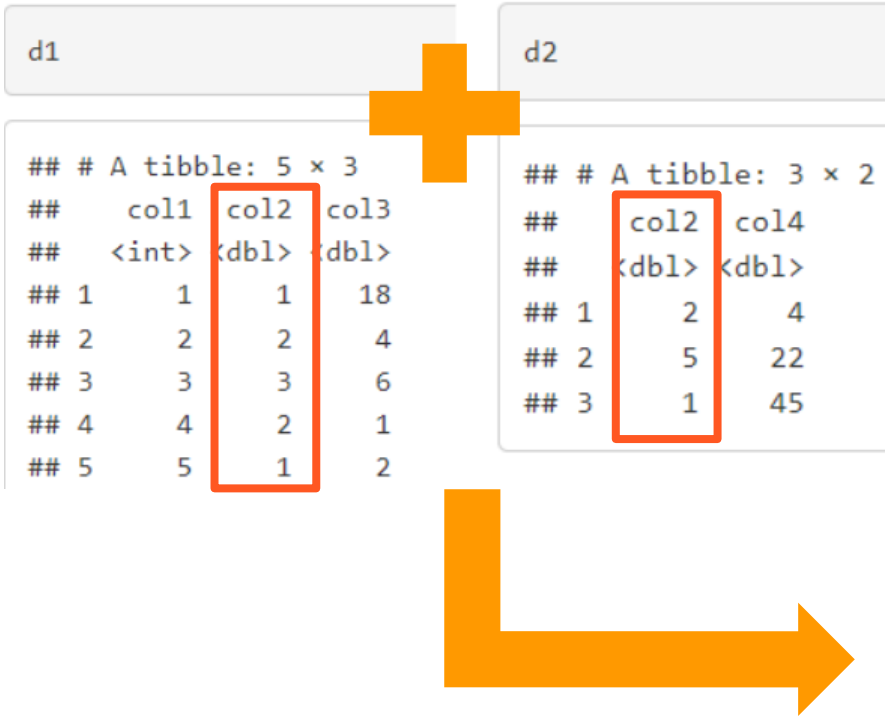
```
d1_d2 <- inner_join(d1,d2)
```

```
## Joining, by = "col2"
```

```
d1_d2
```

```
## # A tibble: 4 × 4
##    col1  col2  col3  col4
##   <int> <dbl> <dbl> <dbl>
## 1     1     1    18    45
## 2     2     2     4     4
## 3     4     2     1     4
## 4     5     1     2    45
```

full_join()



```
d1
```

```
## # A tibble: 5 × 3
##    col1  col2  col3
##   <int> <dbl> <dbl>
## 1     1     1    18
## 2     2     2     4
## 3     3     3     6
## 4     4     2     1
## 5     5     1     2
```

```
d2
```

```
## # A tibble: 3 × 2
##    col2  col4
##   <dbl> <dbl>
## 1     2     4
## 2     5    22
## 3     1    45
```

```
d1_d2 <- full_join(d1,d2)
```

```
## Joining, by = "col2"
```

```
d1_d2
```

```
## # A tibble: 6 × 4
##    col1  col2  col3  col4
##   <int> <dbl> <dbl> <dbl>
## 1     1     1    18    45
## 2     2     2     4     4
## 3     3     3     6    NA
## 4     4     2     1     4
## 5     5     1     2    45
## 6    NA     5    NA    22
```

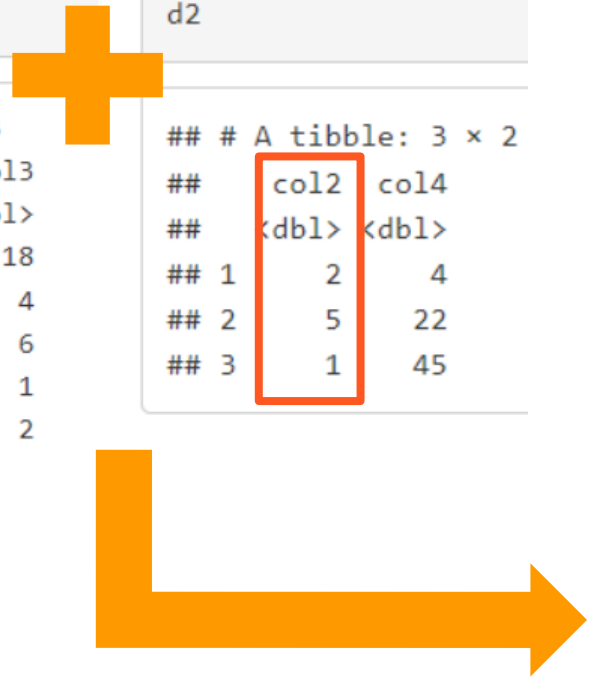
left_join()

```
d1
```

```
## # A tibble: 5 × 3
##    col1  col2  col3
##   <int> <dbl> <dbl>
## 1     1     1    18
## 2     2     2     4
## 3     3     3     6
## 4     4     2     1
## 5     5     1     2
```

```
d2
```

```
## # A tibble: 3 × 2
##    col2  col4
##   <dbl> <dbl>
## 1     2     4
## 2     5    22
## 3     1    45
```



```
d1_new <- left_join(d1,d2)
```

```
## Joining, by = "col2"
```

```
d1_new
```

```
## # A tibble: 5 × 4
##    col1  col2  col3  col4
##   <int> <dbl> <dbl> <dbl>
## 1     1     1    18    45
## 2     2     2     4     4
## 3     3     3     6    NA
## 4     4     2     1     4
## 5     5     1     2    45
```

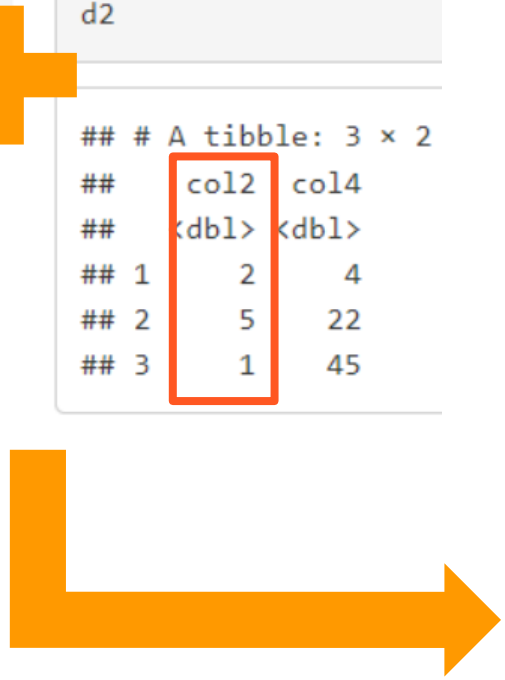
right_join()

```
d1
```

```
## # A tibble: 5 × 3
##    col1  col2  col3
##   <int> <dbl> <dbl>
## 1     1     1    18
## 2     2     2     4
## 3     3     3     6
## 4     4     2     1
## 5     5     1     2
```

```
d2
```

```
## # A tibble: 3 × 2
##    col2  col4
##   <dbl> <dbl>
## 1     2     4
## 2     5    22
## 3     1    45
```



```
d1_d2 <- right_join(d1,d2)
```

```
## Joining, by = "col2"
```

```
d1_d2
```

```
## # A tibble: 5 × 4
##    col1  col2  col3  col4
##   <int> <dbl> <dbl> <dbl>
## 1     1     1    18    45
## 2     2     2     4     4
## 3     4     2     1     4
## 4     5     1     2    45
## 5    NA     5    NA    22
```

Data subsetting: Feature selection

SUBSET WITH CONDITIONAL STATEMENT: dplyr package

filter() – select data that meet certain conditions

select() – select data that contain certain columns

subset() – convenient way of selecting desired data

LOAD DATA (iris)

```
iris <- iris
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
```

1. SUBSET WITH: subset()

```
iris <- iris
iris.new <- subset(iris, Species=="setosa" & Sepal.Width > 3)
head(iris.new)
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1           5.1         3.5          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
## 7           4.6         3.4          1.4         0.3  setosa
```

2. SUBSET WITH: filter() & select()

```
temp_df <- filter(iris, Species=="setosa" & Sepal.Width > 3 )
```

← First, get the desired rows

```
temp_df <- select(temp_df, Species, Sepal.Width)

head(temp_df)
```

← Next, get the desired columns

```
##   Species Sepal.Width
## 1  setosa         3.5
## 2  setosa         3.2
## 3  setosa         3.1
## 4  setosa         3.6
## 5  setosa         3.9
## 6  setosa         3.4
```

← This is how the new data looks like

3. SUBSET WITH: filter(), select(), & PIPE %>%

```
temp_df <- iris %>%
            filter(Species=="setosa" & Sepal.Width > 3) %>%
            select(Species, Sepal.Width)

head(temp_df)
```

```
##   Species Sepal.Width
## 1  setosa         3.5
## 2  setosa         3.2
## 3  setosa         3.1
## 4  setosa         3.6
## 5  setosa         3.9
## 6  setosa         3.4
```

Adding/renaming variable & sorting observations

LOAD DATA (iris)

```
iris <- iris
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
```

ADD NEW COLUMN: **mutate()**

```
iris4 <- iris %>%
          mutate(length_avg=Sepal.Length/Petal.Length)
head(iris4)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species length_avg
## 1          5.1         3.5          1.4         0.2  setosa   3.642857
## 2          4.9         3.0          1.4         0.2  setosa   3.500000
## 3          4.7         3.2          1.3         0.2  setosa   3.615385
## 4          4.6         3.1          1.5         0.2  setosa   3.066667
## 5          5.0         3.6          1.4         0.2  setosa   3.571429
## 6          5.4         3.9          1.7         0.4  setosa   3.176471
```

RENAMING COLUMN: **rename()**

```
iris4 <- iris4 %>%
         rename("Length.Avg"= "length_avg")
head(iris4)
```

←RENAME (“NEW” = “OLD”)

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species Length.Avg
## 1          5.1         3.5          1.4         0.2  setosa   3.642857
## 2          4.9         3.0          1.4         0.2  setosa   3.500000
## 3          4.7         3.2          1.3         0.2  setosa   3.615385
## 4          4.6         3.1          1.5         0.2  setosa   3.066667
## 5          5.0         3.6          1.4         0.2  setosa   3.571429
## 6          5.4         3.9          1.7         0.4  setosa   3.176471
```

SORT OBSERVATIONS: **arrange()**

We want to sort observations in certain variable in “ascending” order

```
iris5 <- iris %>%
          arrange(Sepal.Length) # sort in ascending order

head(iris5)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          4.3         3.0          1.1         0.1  setosa
## 2          4.4         2.9          1.4         0.2  setosa
## 3          4.4         3.0          1.3         0.2  setosa
## 4          4.4         3.2          1.3         0.2  setosa
## 5          4.5         2.3          1.3         0.3  setosa
## 6          4.6         3.1          1.5         0.2  setosa
```

SORT OBSERVATIONS: **arrange()**

Sort observations in certain variable in “descending” order

```
iris6 <- iris %>%
        arrange(desc(Sepal.Length))  # desc(Sepal.Length) for descending order

head(iris6)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1          7.9         3.8          6.4         2.0 virginica
## 2          7.7         3.8          6.7         2.2 virginica
## 3          7.7         2.6          6.9         2.3 virginica
## 4          7.7         2.8          6.7         2.0 virginica
## 5          7.7         3.0          6.1         2.3 virginica
## 6          7.6         3.0          6.6         2.1 virginica
```

Exercise subsetting (example dataset)

CREATE DATA

First, create 5 columns like this

```
Name <- c("Charlie", "Cindy", "David", "Flower")
Age <- c(10, 15, 20, 25)
Gender <- c("Male", "Female", "Male", "Female")
BloodType <- c("AB", "A", "B", "O")
HighestDegree  <- c("High school", "College", "Grad school", "High school")
```

CREATE DATA

Then, make them into one data frame

```
g.dat <- data.frame(Name, Age, Gender, BloodType, HighestDegree)
```

CHECKING YOUR DATA FRAME

How does your data frame look like?

```
g.dat
```

```
##      Name Age Gender BloodType HighestDegree
## 1 Charlie  10   Male        AB   High school
## 2   Cindy  15 Female         A       College
## 3   David  20   Male         B   Grad school
## 4  Flower  25 Female         O   High school
```

SUBSETTING DATA: TASK 1

people whose age fall between 15 and 20

SUBSETTING DATA

One way to do it – use subset() function

Subset data so that it contain characteristics of people whose age fall between 15 and 20

```
g.dat1 <- subset(g.dat, Age>=15 & Age<=20)
head(g.dat1)
```

```
##    Name Age Gender BloodType HighestDegree
## 2 Cindy  15 Female         A       College
## 3 David  20   Male         B  Grad school
```

SUBSETTING DATA

Other way to do it – use filter(), select(), and pipe %>% function

Subset data so that it contain characteristics of people whose age fall between 15 and 20

```
g.dat3 <- g.dat %>%
          filter(Age>=15 & Age<=20)
head(g.dat3)
```

```
##     Name Age Gender BloodType HighestDegree
## 1 Cindy  15 Female         A        College
## 2 David  20   Male         B   Grad school
```

SUBSETTING DATA: TASK 2

female whose age>=15

SUBSETTING DATA

One way to do it – use subset() function

Subset data so that it contain characteristics of female whose age>=15

```
g.dat4 <- subset(g.dat, Age>=15 & Gender=="Female")
head(g.dat4)
```

```
##     Name Age Gender BloodType HighestDegree
## 2  Cindy  15 Female         A       College
## 4 Flower  25 Female         O   High school
```

SUBSETTING DATA

Other way to do it – use filter(), select(), and pipe %>% function

Subset data so that it contain characteristics of female whose age>=15

```
g.dat5 <- g.dat %>%
          filter(Age>=15 & Gender=="Female")
head(g.dat5)
```

```
##      Name Age Gender BloodType HighestDegree
## 1   Cindy  15 Female         A        College
## 2 Flower  25 Female         O    High school
```