Preliminaries: data processing

Hyeongseong (Sean) Lee

Department of Statistics and Applied Probability

University of California, Santa Barbara

Summer 2022

Data processing in R: `tidyverse`

From PSTAT 10...

```
> data = read.csv('database.csv', header=TRUE)
> data = data[!is.na(data[,1]),]
> data = data[,c(2,3,5)]
> ...
```

We want to process basic steps all at once...

Data processing in R: tidyverse

Let us use the package tidyverse

```
> install.packages(c('tidyverse', 'dplyr'))
> library(tidyverse)
> library(dplyr)
```

- We can use pipelines to combine all the processing steps together

Data processing in R: `tidyverse`

Basic functions for pipelines

- `drop_na`: to delete all the observation with missing values
- `filter`: to extract rows under certain condition
- `mutate`: to create new columns
- `select`: to extract specific columns
- `within`: to change the property of columns as you desire
- `group_by` and `summarise`: to get group-wise summary

Data processing in R: `tidyverse`

Terminologies

- data matrix $X_{n \times p}$
- observations, records: rows
- features, variables: columns
- y (target variable, dependent variable)
  $\longleftrightarrow X_1, X_2, \cdots, X_p$ (predictors, independent variables, features)

Data processing in R: tidyverse

Example

- Given data: data with dimension 1,000×5
- Column names: var1 (numeric); var2 (character; coded as A, B, C); var3 (numeric); var4 (numeric); and var5 (trinomial; 0 1 2 but coded as numeric)
- We want to 1) drop all the rows with missing values; 2) extract observations with var5 being only 0 or 1; 3) create a new column var6=var1+var4; 4) change the type of var2 and var5 to factors; and 5) select var4, var5, and var6

```
> data = read.csv('database.csv', header=TRUE)
> data = data %>% drop_na() %>% filter(var5!=2)
%>% mutate(var6=var1+var4) %>% within({var2 =
factor(var2)
var5 = factor(var5)}) %>% select(var2, var5, var6)
```

Data processing in R: tidyverse

Example

- Given data: same as previous one
- We want to calculate the mean value of var1 for each category of var2

```
> data = read.csv('database.csv', header=TRUE)
> data = data %>% drop_na() %>%
within(var2=factor(var2)) %>% group_by(var2) %>%
summarise(mean_var1 = mean(var1))
```

Note: We can also use tapply function

Data processing in R: loop

- `for`
- `repeat`
- `while`
- How to update the result of each loop? `append`, `cbind`, `rbind`