Basic Data Wrangling & Exploration, ft. Tidyverse

Kursus R: Pengenalan dan Praktikal

Mohd Azmi Bin Suliman

azmi.suliman@moh.gov.my

Pusat Penyelidikan Penyakit Tak Berjangkit, Institut Kesihatan Umum

Wednesday, 02 October 2024

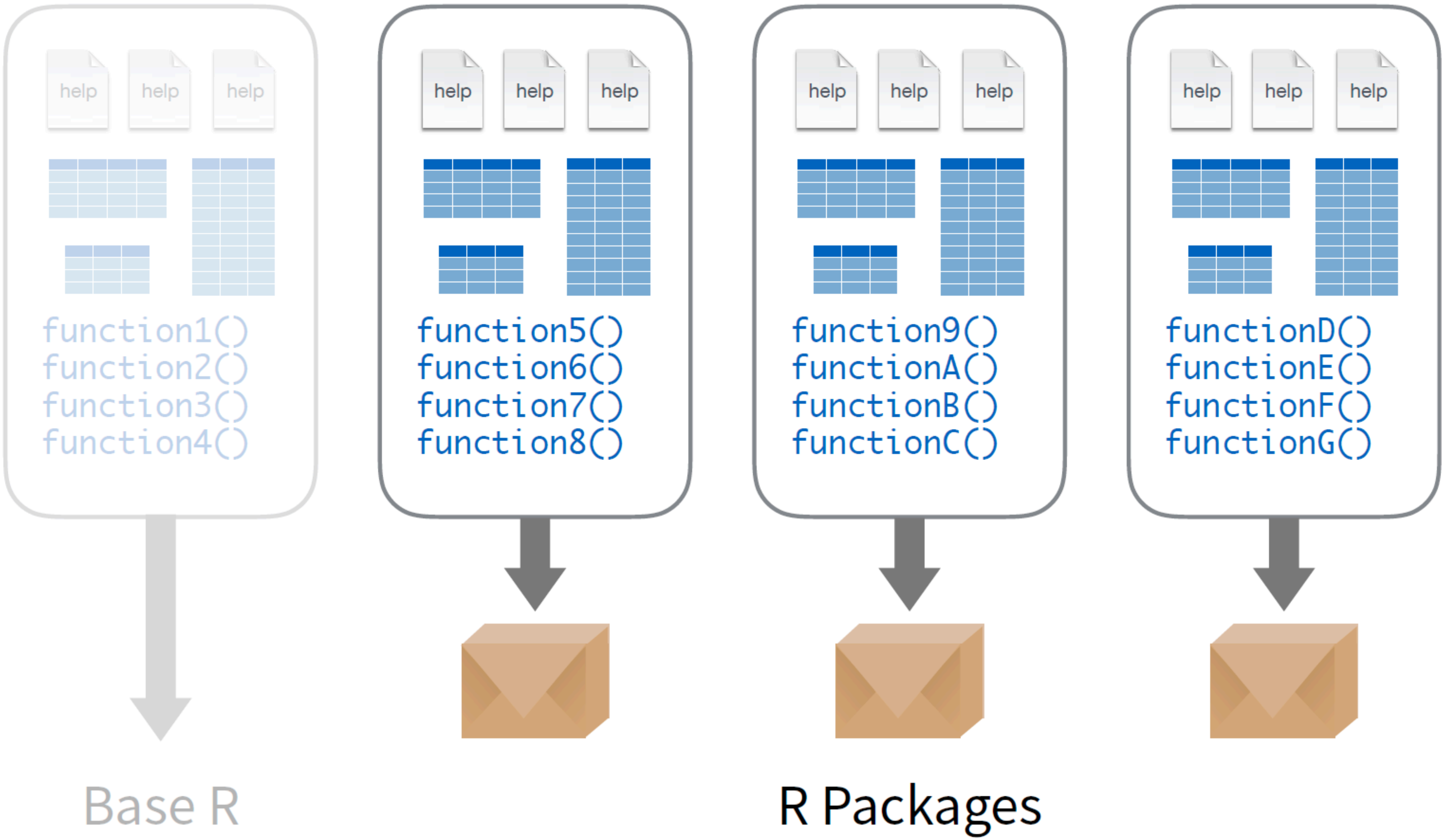
Packages

Packages in R

- The strength of R is it’s open-source philosophy
- When installing R, you get a basic set of packages
- Package in R is a collection of functions, data, and compiled code
- R allow for custom function, and even custom package
- The custom package can be shared with others, e.g., via CRAN
- These custom packages enriched the R ecosystem

Packages in R

Packages in R



Packages in R

- Among common packages in R

Package	Description
tidyverse	collection of packages designed for data science
haven	import/export SPSS, Stata and SAS files
readxl	import/export excel files
lubridate	work with date and time
ggplot2	data visualisation
dplyr	data manipulation
stringr	string manipulation
forcats	factor manipulation

Packages in R

- Install once
  - → using `install.packages("package_name")`
  - → can also click on `install` button at `Packages` pane
- Load every time you use R
  - → using `library(package_name)`
  - → can also tick at respective library at `Packages` pane

Packages in R

- Lets try install `tibble` package using code

```
install.packages("tibble")
```

- Then load the package using code

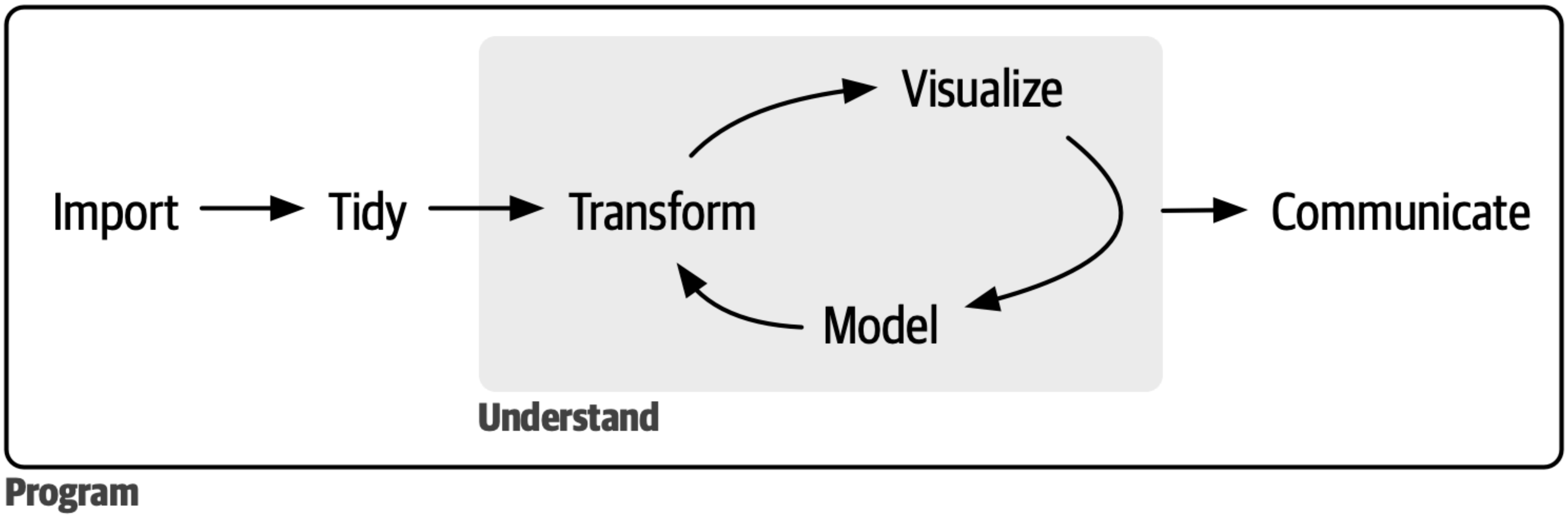
```
library(tibble)
```

Can you try install `tidyverse` package?

Data Wrangling

What is data wrangling?

- Common data analysis look like this



source: r4ds.hadley.nz

What is data wrangling?

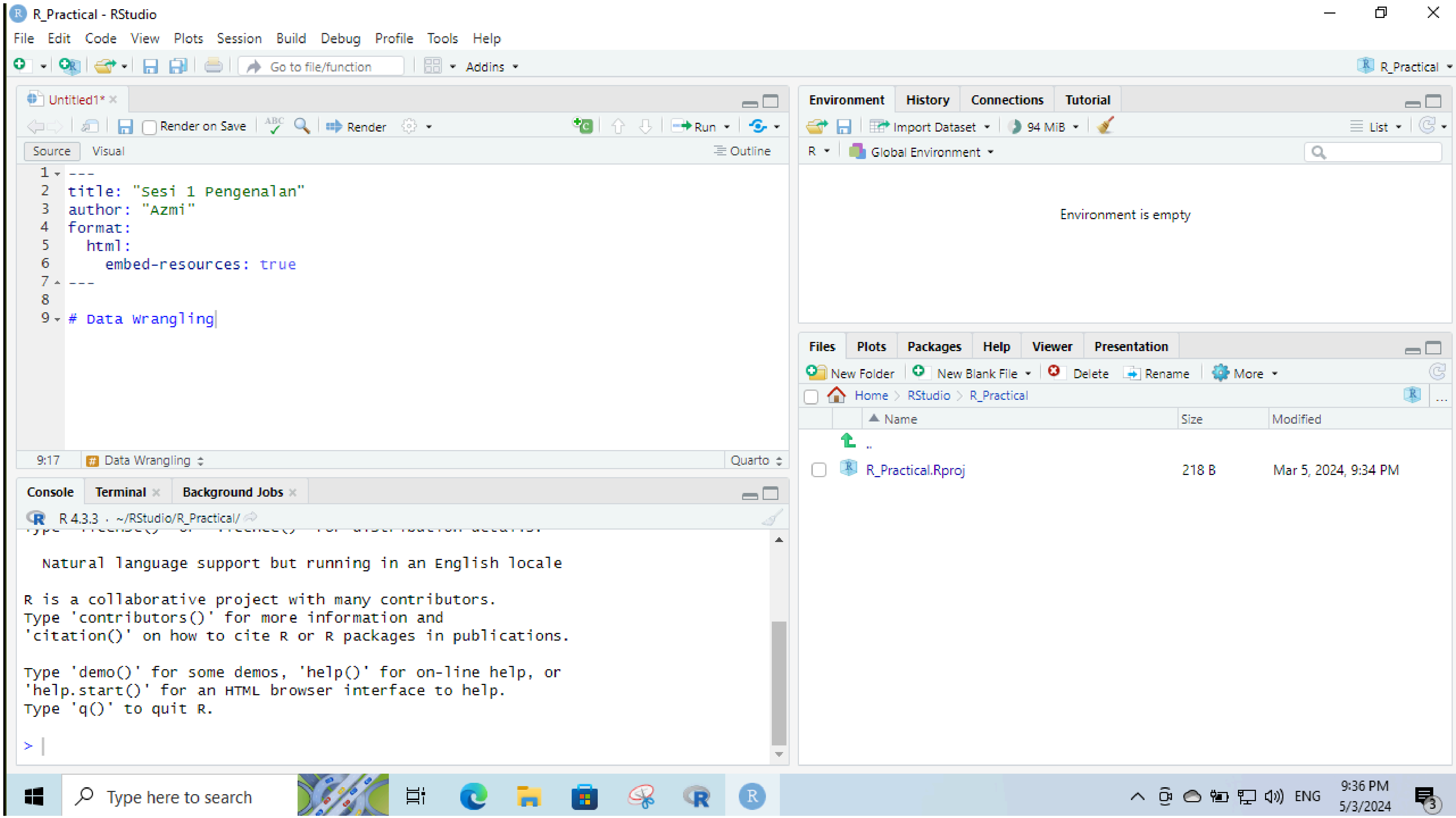
- real world data commonly messy!
- 80% of time taken spend on data cleaning
- improving data quality > improving the accuracy & efficiency
- data wrangling involving **tidying** and **transforming** data, from raw form to analysis-ready data.
- common data wrangling action
    - → label data
    - → recategorise categorical variable - usually collapsing groups
    - → binning continuous variable

Lets try some data wrangling

Setup your project

- We will use
  - → current project `R_Practical`
  - → current Quarto document `Sesi 1 Pengenalan`
- Add new level 1 header
  - → Single # symbol
  - → Followed by the title `Data Wrangling`

Setup your project

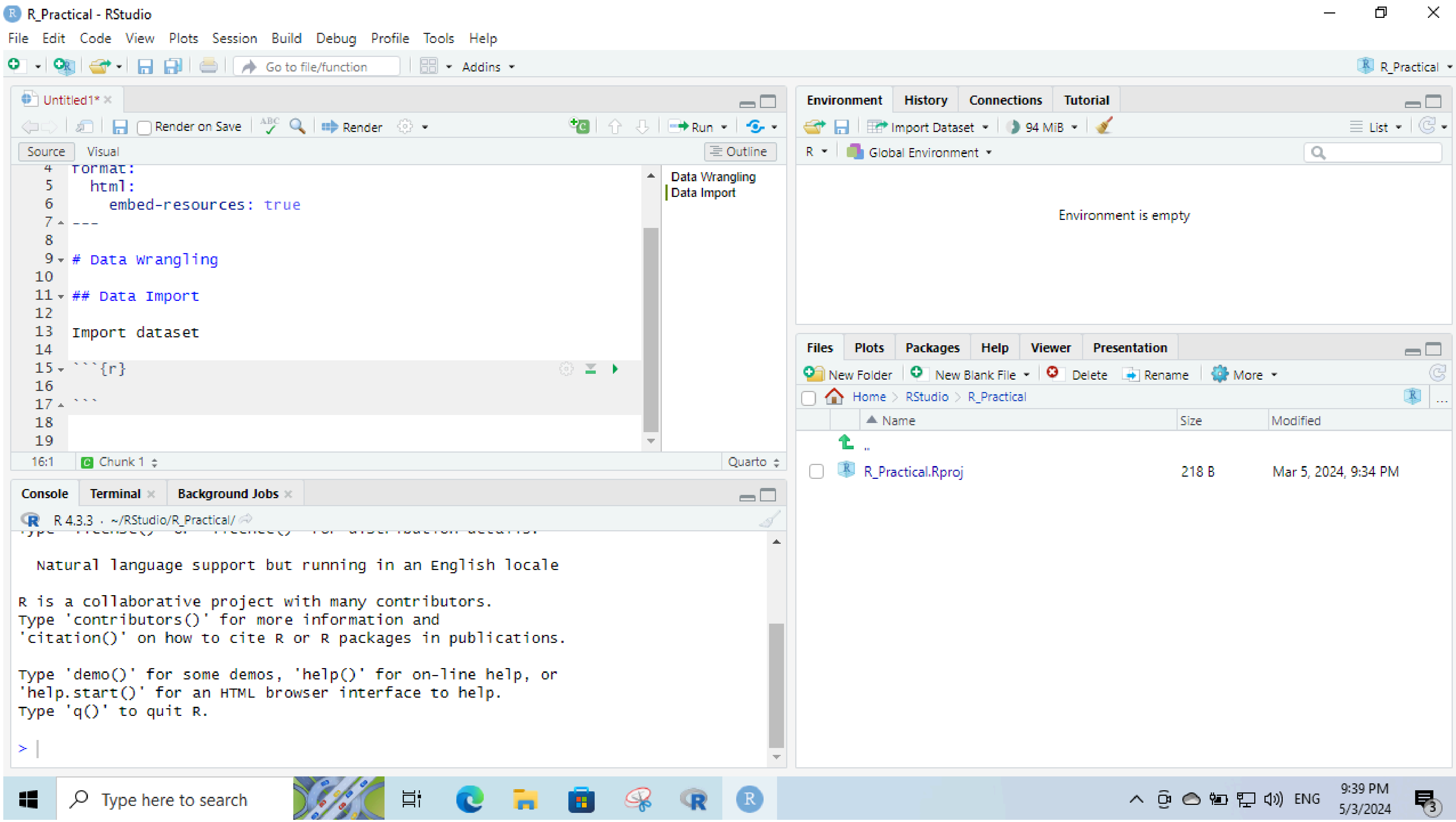


Import Dataset

Import Dataset

- We will use the `asthma_ds.sav` dataset (SPSS file)
- Copy the dataset to the `R_Practical` project folder (working directory)
- Add new level 2 header
  - → Double ## symbol
  - → Followed by the title `Import Dataset`
- Insert new R code chunk

Import Dataset



Import Dataset

- We will use the `read_sav` `haven` package to import the dataset
- The dataset is stored in R object `asthma_ds0`

```
library(tidyverse)
library(haven)

asthma_ds0 <- read_sav("Dataset/asthma_ds.sav")
```

Import Dataset

- We can view the dataset by writing the object name

```
asthma_ds0
```

```
# A tibble: 150 × 16
      id idR   Gender       Age WorkStatus   Height Weight_Pre PA_HW Weight_Post
   <dbl> <chr> <dbl+lbl>  <dbl> <dbl+lbl>     <dbl>      <dbl> <dbl>       <dbl>
 1     1 nXSw  2 [Female]    34 2 [Unemploy…    179       84.2     5        76.7
 2     2 yg2t  1 [Male]      31 2 [Unemploy…    169       81.8     1        75.7
 3     3 QBW4  1 [Male]      25 1 [Employed]    164       88.5     1        84.1
 4     4 2x2S  2 [Female]    33 2 [Unemploy…    136       53.2     4        46.9
 5     5 mOnn  1 [Male]      28 2 [Unemploy…    172       71.3     2        63.3
 6     6 D3sl  1 [Male]      33 2 [Unemploy…    178       87.3     0        82.6
 7     7 le6j  2 [Female]    31 2 [Unemploy…    140       48.8     5        41.9
 8     8 r3gC  2 [Female]    34 1 [Employed]    140       49.1     2        43.6
 9     9 3Tyt  1 [Male]      31 1 [Employed]    171       60.1     1        54.7
10    10 cmKF  1 [Male]      28 1 [Employed]    163       93.1     1        86.2
# ℹ 140 more rows
# ℹ 7 more variables: Tx2 <dbl+lbl>, PEFR_Pre <dbl>, PEFR_Post <dbl>,
#   SxWheeze_Pre <dbl+lbl>, SxWheeze_Post <dbl+lbl>, PS_Pre <dbl>,
#   PS_Post <dbl>
```

Import Dataset

- We can also use `View` function (capital V)

```
View(asthma_ds0)
```

Import Dataset

- In this dataset, we can notice that the `Gender` variable is coded as `1` and `2`, with label `Male` and `Female` respectively.
- Similar to other factor (categorical) variables in this dataset.
- This is common in SPSS dataset, where the label is stored separately from the data.
- We can use `as_factor` function to convert the variable to factor, and apply the label to the factor.

```
asthma_ds <- as_factor(asthma_ds0)
```

Import Dataset

```r
head(asthma_ds)
```

```
# A tibble: 6 × 16
     id idR   Gender   Age WorkStatus Height Weight_Pre PA_HW Weight_Post Tx2
  <dbl> <chr> <fct>  <dbl> <fct>       <dbl>      <dbl> <dbl>       <dbl> <fct>
1     1 nXSw  Female    34 Unemployed    179       84.2     5        76.7 Drug B
2     2 yg2t  Male      31 Unemployed    169       81.8     1        75.7 Place…
3     3 QBW4  Male      25 Employed      164       88.5     1        84.1 Drug B
4     4 2x2S  Female    33 Unemployed    136       53.2     4        46.9 Drug A
5     5 mOnn  Male      28 Unemployed    172       71.3     2        63.3 Place…
6     6 D3sl  Male      33 Unemployed    178       87.3     0        82.6 Drug A
# ℹ 6 more variables: PEFR_Pre <dbl>, PEFR_Post <dbl>, SxWheeze_Pre <fct>,
#   SxWheeze_Post <fct>, PS_Pre <dbl>, PS_Post <dbl>
```

Simple Data Wrangling

Select Variable/Column

- We can select variable/column using `select` function
- We can use `:` to select range of variable

```r
asthma_ds1 <- select(asthma_ds, idR:Weight_Post)

head(asthma_ds1)
```

```
# A tibble: 6 × 8
  idR   Gender   Age WorkStatus Height Weight_Pre PA_HW Weight_Post
  <chr> <fct>  <dbl> <fct>       <dbl>      <dbl> <dbl>       <dbl>
1 nXSw  Female    34 Unemployed    179       84.2     5        76.7
2 yg2t  Male      31 Unemployed    169       81.8     1        75.7
3 QBW4  Male      25 Employed      164       88.5     1        84.1
4 2x2S  Female    33 Unemployed    136       53.2     4        46.9
5 mOnn  Male      28 Unemployed    172       71.3     2        63.3
6 D3sl  Male      33 Unemployed    178       87.3     0        82.6
```

Arithmetic Transformation

- We can perform arithmetic transformation using `mutate` function
  - → For example, we can calculate BMI using `Weight_Pre` and `Height` variable

```r
asthma_ds2 <- asthma_ds1
asthma_ds2$Ht_m <- asthma_ds2$Height/100
asthma_ds2$BMI_Pre <- asthma_ds2$Weight_Pre/(asthma_ds2$Ht_m^2)

head(asthma_ds2)
```

```
# A tibble: 6 × 10
  idR   Gender   Age WorkStatus Height Weight_Pre PA_HW Weight_Post  Ht_m
  <chr> <fct>  <dbl> <fct>       <dbl>      <dbl> <dbl>       <dbl> <dbl>
1 nXSw  Female    34 Unemployed    179       84.2     5        76.7  1.79
2 yg2t  Male      31 Unemployed    169       81.8     1        75.7  1.69
3 QBW4  Male      25 Employed      164       88.5     1        84.1  1.64
4 2x2S  Female    33 Unemployed    136       53.2     4        46.9  1.36
5 mOnn  Male      28 Unemployed    172       71.3     2        63.3  1.72
6 D3sl  Male      33 Unemployed    178       87.3     0        82.6  1.78
# ℹ 1 more variable: BMI_Pre <dbl>
```

Binning Continuous Variable

- We can bin continuous variable using `cut` function
  → For example, we can bin `BMI_Pre` variable into 4 categories

BMI Category	BMI Range
Underweight	<= 18.49
Normal	18.50 - 22.99
Overweight	23.00 - 24.99
Obese	>= 25.00

Binning Continuous Variable

- For example, we can bin `BMI_Pre` variable into 4 categories

```
asthma_ds3 <- asthma_ds2
asthma_ds3$BMI_PreCat <- cut(asthma_ds3$BMI_Pre,
                             breaks = c(0, 18.49, 22.99, 24.99, 100),
                             labels = c("Underweight", "Normal",
                                        "Overweight", "Obese"))

head(asthma_ds3)
```

```
# A tibble: 6 × 11
  idR   Gender   Age WorkStatus Height Weight_Pre PA_HW Weight_Post  Ht_m
  <chr> <fct>  <dbl> <fct>       <dbl>      <dbl> <dbl>       <dbl> <dbl>
1 nXSw  Female    34 Unemployed    179       84.2     5        76.7  1.79
2 yg2t  Male      31 Unemployed    169       81.8     1        75.7  1.69
3 QBW4  Male      25 Employed      164       88.5     1        84.1  1.64
4 2x2S  Female    33 Unemployed    136       53.2     4        46.9  1.36
5 mOnn  Male      28 Unemployed    172       71.3     2        63.3  1.72
6 D3sl  Male      33 Unemployed    178       87.3     0        82.6  1.78
# ℹ 2 more variables: BMI_Pre <dbl>, BMI_PreCat <fct>
```

Tidyverse

What is Tidyverse?

- Tidyverse is a collection of packages designed for data science
- The strength of Tidyverse is the **tidy** data philosophy
- Pipe operator %>% is the main feature of Tidyverse
- The packages are:

Package	Description
ggplot2	data visualisation
dplyr	data manipulation
tidyr	data tidying
readr	data import
purrr	functional programming

Package	Description
tibble	data structure
stringr	string manipulation
forcats	factor manipulation
broom	tidy statistical output
modelr	modelling functions

How does tidyverse make code more readable?

- Tidyverse is designed to work together
- `mutate` function is used to create new variable
- While pipe operator `%>%` is used to chain multiple function
- These two were commonly used in Tidyverse

`mutate` function

- `mutate` function is used to
  - → create new variable
  - → modify existing variable

```r
asthma_ds4 <- asthma_ds1
asthma_ds4 <- mutate(asthma_ds4, Ht_m = Height/100)
head(asthma_ds4)
```

```
# A tibble: 6 × 9
  idR   Gender   Age WorkStatus Height Weight_Pre PA_HW Weight_Post  Ht_m
  <chr> <fct>  <dbl> <fct>       <dbl>      <dbl> <dbl>       <dbl> <dbl>
1 nXSw  Female    34 Unemployed    179       84.2     5        76.7  1.79
2 yg2t  Male      31 Unemployed    169       81.8     1        75.7  1.69
3 QBW4  Male      25 Employed      164       88.5     1        84.1  1.64
4 2x2S  Female    33 Unemployed    136       53.2     4        46.9  1.36
5 mOnn  Male      28 Unemployed    172       71.3     2        63.3  1.72
6 D3sl  Male      33 Unemployed    178       87.3     0        82.6  1.78
```

`mutate` function + Pipe Operator

- We can use pipe operator `%>%` to chain multiple function

Example of Data Wrangling **without** Pipe Operator

```r
asthma_ds0 <- read_sav("Dataset/asthma_ds.sav")
asthma_ds1 <- as_factor(asthma_ds0)
asthma_ds2 <- select(asthma_ds1, idR:Weight_Post)
asthma_ds3 <- asthma_ds2
asthma_ds3$Ht_m <- asthma_ds3$Height/100
asthma_ds3$BMI_Pre <- asthma_ds3$Weight_Pre/(asthma_ds3$Ht_m^2)
asthma_ds3$BMI_PreCat <- cut(asthma_ds3$BMI_Pre,
                                      breaks = c(0, 18.49, 22.99, 24.99, 100),
                                      labels = c("Underweight", "Normal",
                                                 "Overweight", "Obese"))
head(asthma_ds3)
```

```
# A tibble: 6 × 11
  idR   Gender   Age WorkStatus Height Weight_Pre PA_HW Weight_Post  Ht_m
  <chr> <fct>  <dbl> <fct>       <dbl>      <dbl> <dbl>       <dbl> <dbl>
1 nXSw  Female    34 Unemployed    179       84.2     5        76.7  1.79
2 yg2t  Male      31 Unemployed    169       81.8     1        75.7  1.69
3 QBW4  Male      25 Employed      164       88.5     1        84.1  1.64
4 2x2S  Female    33 Unemployed    136       53.2     4        46.9  1.36
5 mOnn  Male      28 Unemployed    172       71.3     2        63.3  1.72
6 D3sl  Male      33 Unemployed    178       87.3     0        82.6  1.78
# ℹ 2 more variables: BMI_Pre <dbl>, BMI_PreCat <fct>
```

Example of Data Wrangling **with** Pipe Operator

```
asthma_ds5 <- read_sav("Dataset/asthma_ds.sav") %>%
  as_factor() %>%
  select(idR:Weight_Post) %>%
  mutate(Ht_m = Height/100,
         BMI_Pre = Weight_Pre/(Ht_m^2),
         BMI_PreCat = cut(BMI_Pre, breaks = c(0, 18.49, 22.99, 24.99, 10
                          labels = c("Underweight", "Normal",
                                     "Overweight", "Obese")))

head(asthma_ds5)
```

```
# A tibble: 6 × 11
  idR   Gender   Age WorkStatus Height Weight_Pre PA_HW Weight_Post  Ht_m
  <chr> <fct>  <dbl> <fct>       <dbl>      <dbl> <dbl>       <dbl> <dbl>
1 nXSw  Female    34 Unemployed    179       84.2     5        76.7  1.79
2 yg2t  Male      31 Unemployed    169       81.8     1        75.7  1.69
3 QBW4  Male      25 Employed      164       88.5     1        84.1  1.64
4 2x2S  Female    33 Unemployed    136       53.2     4        46.9  1.36
5 mOnn  Male      28 Unemployed    172       71.3     2        63.3  1.72
6 D3sl  Male      33 Unemployed    178       87.3     0        82.6  1.78
# ℹ 2 more variables: BMI_Pre <dbl>, BMI_PreCat <fct>
```

Data Exploration with Tidyverse

Data Exploration

- Data exploration is the first step in data analysis
- But data exploration also commonly use at various stage of data analysis

Summary Statistics with `summarise` function

- Tidyverse provide various function to get summary statistics
  - → `summarise` function

```
asthma_ds5 %>%
  summarise(Mean_Height = mean(Height),
            SD_Height = sd(Height),
            Min_Height = min(Height),
            Max_Height = max(Height))
```

```
# A tibble: 1 × 4
  Mean_Height SD_Height Min_Height Max_Height
        <dbl>     <dbl>      <dbl>      <dbl>
1        164.      15.3        129        195
```

Summary Statistics with `count` function

- `count` function is used to count the frequency of each level of a factor variable

```
asthma_ds5 %>%
  count(Gender)
```

```
# A tibble: 2 × 2
  Gender      n
  <fct>   <int>
1 Male       86
2 Female     64
```

Summary Statistics with `group_by` function

- `group_by` function is used to group the data by a factor variable

```
asthma_ds5 %>%
  group_by(Gender) %>%
  summarise(Mean_Height = mean(Height),
            SD_Height = sd(Height),
            Min_Height = min(Height),
            Max_Height = max(Height))
```

```
# A tibble: 2 × 5
  Gender Mean_Height SD_Height Min_Height Max_Height
  <fct>        <dbl>     <dbl>      <dbl>      <dbl>
1 Male          174.      8.98        151        195
2 Female        150.     11.0         129        179
```

Cross Tabulation with `table` function

- `table` function is used for crosstabulation

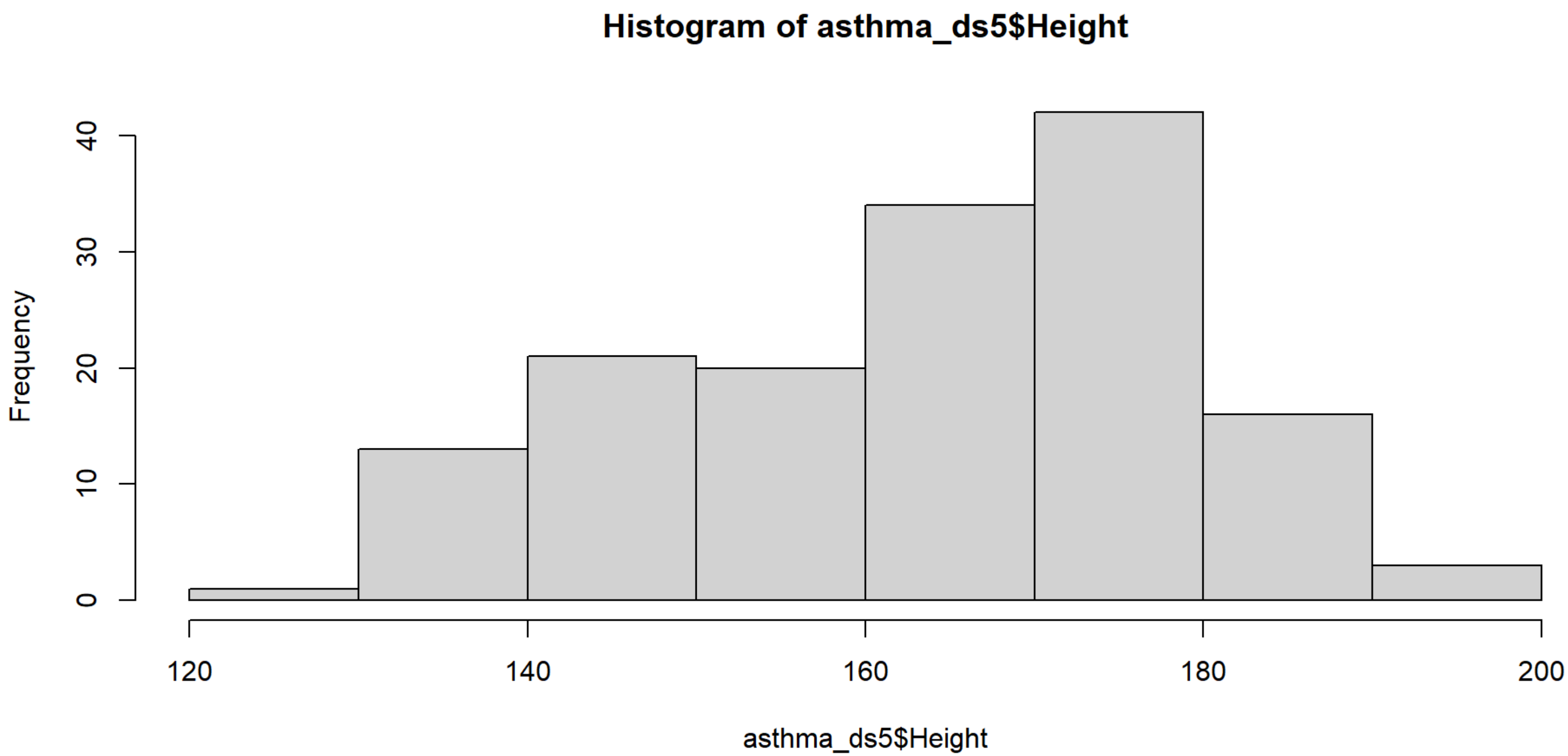
```
with(asthma_ds5, table(WorkStatus, Gender))
```

```
            Gender
WorkStatus   Male Female
  Employed     53     17
  Unemployed   33     47
```

Plot: Histogram

- We can quickly plot histogram using base R function `hist`

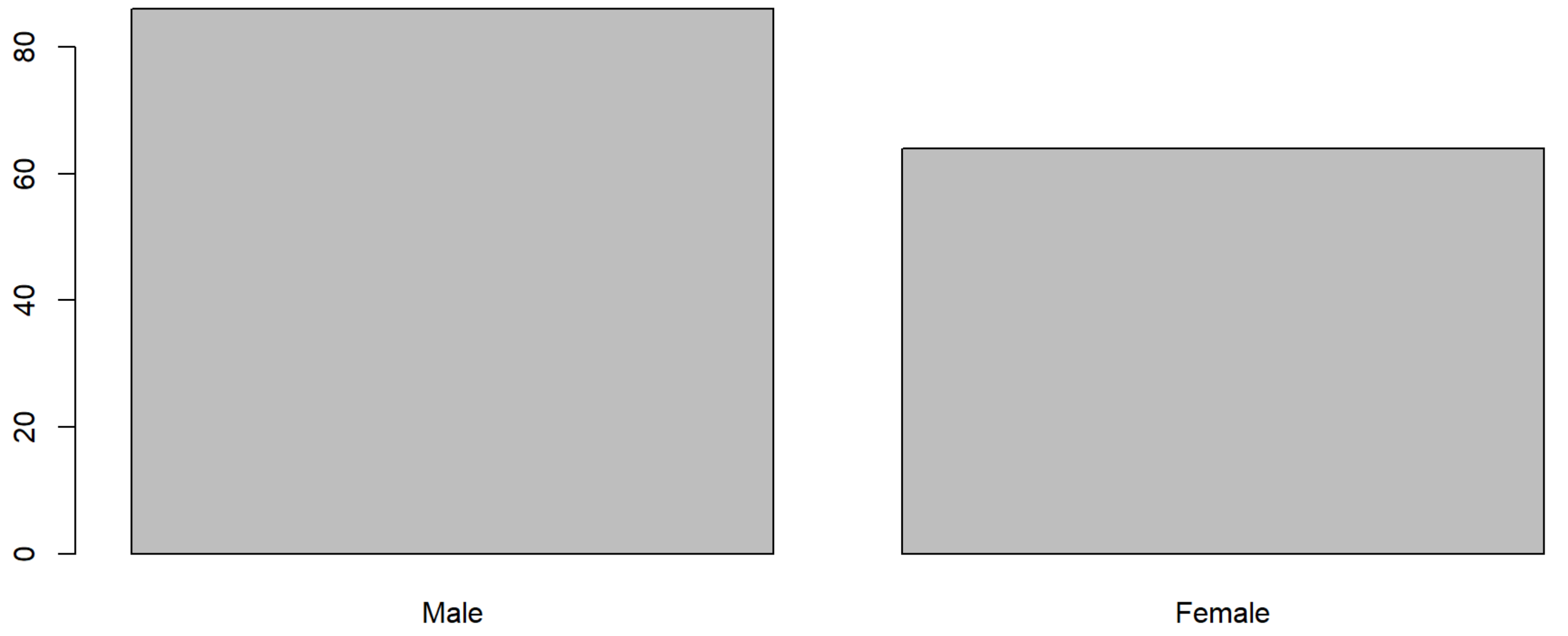
```
hist(asthma_ds5$Height)
```



Plot: Bar Chart

- We can quickly plot bar chart using base R function `barplot`

```
barplot(table(asthma_ds5$Gender))
```



Finish! See you tomorrow~