simon-5505-02-slides

2025-03-04

Topics to be covered

```
# A small section of R code to kick start the params.
```

- What you will learn
  - Using select, slice, filter
  - Using mutate, summarize
  - Missing values
  - Histograms
  - Scatterplots
  - Your homework

Some definitions, 1

- Categorical = small number of possible values
- Examples
  - Sex (Male or Female),
  - Race/ethnicity (Caucasian, African American, Hispanic, etc.),
  - Cancer stage (I, II, III, or IV),
  - Birth delivery type (Vaginal, C-section).

Speaker notes

A **categorical variable** is a variable that can only take on a small number of values. Each value is usually associated with a particular category.

Some definitions, 2

- Continuous variable = large number of possible values
- Examples of continuous variables are
  - Birth weight in grams,
  - Gestational age,
  - Fasting LDL level.

Speaker notes

A **continuous variable** is a variable that can take on a large number of possible values, potentially any value in some interval.

There are some variables that are on the boundary between categorical and continuous, but it is not worth quibbling about here.

The point to remember is that the types of graphs that you use and the types of statistics that you compute are dependent on many things, but first and foremost on whether the variables are categorical, continuous, or a mixture.

Today, you will see examples involving mostly continuous variables.

Variable names

- Short but descriptive
- Mix of letters and numbers
  - Must start with a letter
  - Avoid most symbols
- No blanks
  - CamelCase
  - dot.delimited.names
  - underscore_delimited_names

Speaker notes

This data set did not have a header, a line at the very top of the file that lists variable names. R uses the default names V1, V2, etc. As a general rule, you should use brief (but descriptive) names for every variable in your data set. The names should be around 8 characters long. Longer variable names make your typing tedious and much shorter variable names makes your code terse and cryptic.

You should avoid special symbols in your variable names especially symbols that are likely to cause confusion (the dash symbol, for example, which is also the symbol for subtraction). You should also avoid blanks. These rules are pretty much universal across most statistical software packages. If you violate these rules, you will find out that, at a minimum, you will always have to surround your variable name by quotes to avoid problems.

There are times when you'd like to have a blank in your variable name and you can use two special symbols that you can use in R (and most other statistical packages), the underscore symbol (above the minus key on most keyboards) and the dot (period). These symbols create some artificial spacing that mimics the blanks. Another approach is "CamelCase" which is the mixture of upper and lower case within a variable name with each uppercase designating the beginning of a new "word".

Data dictionary for burger-calories

The next few examples will use a data file, burger-calories.

Refer to the data dictionary for details about this file.

Read burger-calories

```r
library(tidyverse)

burgers <- read_tsv(
  file="../data/burger-calories.txt",
  col_types="nnn")
burgers
```

Select, 1

```r
burgers |>
  select(Calories)
```

Select, 2

```r
burgers |>
  select(-Fat)
```

Alternatives to select, 1

```
burgers$Calories
```

Alternatives to select, 2

```
burgers[ , 1]
```

Alternatives to select, 3

```r
burgers[ , "Calories"]
```

Slice, 1

```r
burgers |>
  slice(1)
```

Slice, 2

```r
burgers |>
  slice_max(Calories)
```

Filter, 1

```r
burgers |>
  filter(Sodium == 1500)
```

Filter, 2

```r
burgers |>
  filter(Calories >= 600)
```

Alternatives to slice/filter, 1

```r
burgers[1, ]
```

Alternatives to slice/filter, 2

```r
burgers[burgers$Sodium == 1500, ]
```

Live demo, part 1

Break #1

- What you have learned
  - Using select, slice, filter
- What’s coming next
  - Using mutate, summarize

Data dictionary for sleep

The next few examples will use a different data file, sleep. Refer to the data dictionary for details about this file.

Read text file

```r
library(tidyverse)
```

```r
sleep <- read_tsv(
  file="../data/sleep.txt",
  col_types="cnnnnnnnnnn")
glimpse(sleep)
```

Convert brain weight from grams to kilograms

```r
sleep |>
  mutate(brain_wt_kg=BrainWt/1000) -> sleep
sleep |>
  slice(1:10) |>
  select(Species, BrainWt, brain_wt_kg)
```

Calculate brain to body weight ratio

```r
sleep |>
  mutate(brain_pct=100*brain_wt_kg/BodyWt) |>
  mutate(brain_pct=round(brain_pct, 2)) |>
  mutate(brain_pct=paste0(brain_pct, "%")) -> sleep

sleep |>
  slice(1:10) |>
  select(Species, brain_wt_kg, BodyWt, brain_pct)
```

Summarize body weight

```r
sleep |>
  summarize(
    body_wt_mean=round(mean(BodyWt), 0),
    body_wt_sd=round(sd(BodyWt), 0),
    body_wt_min=min(BodyWt),
    body_wt_max=max(BodyWt))
```

Interpretation of body weight statistics

The mean body weight is just a bit below 200 kilograms with a standard devation of about 900. The range of weights is extremely broad. The smallest animal is 0.005 kilograms (or 5 grams) and the largest is over 6,600 kilograms.

Live demo, part 2

Break #2

- What you have learned
  - Using mutate, summarize
- What’s coming next
  - Missing values

Variable with missing values

```r
sleep$TotalSleep
```

```
 [1]  3.3  8.3 12.5 16.5  3.9  9.8 19.7  6.2 14.5  9.7 12.5  3.9 10.3  3.1
8.4
[16]  8.6 10.7 10.7  6.1 18.1   NA  3.8 14.4 12.0  6.2 13.0 13.8  8.2  2.9
10.8
[31]   NA  9.1 19.9  8.0 10.6 11.2 13.2 12.8 19.4 17.4   NA 17.0 10.9 13.7
8.4
[46]  8.4 12.5 13.2  9.8  9.6  6.6  5.4  2.6  3.8 11.0 10.3 13.3  5.4 15.8
10.3
[61] 19.4   NA
```

Speaker notes

Several variables in the sleep file have missing values. There are many reasons why a variable might be missing. Maybe someone dropped a test tube in a lab. Maybe someone dropped out of a study. Maybe they skipped a question on your questionnaire.

Always find out why a variable is missing. How you handle the missing value during data analysis depends on the reason it is missing. This is well beyond the scope of the class, but you do need to work with missing values during the data management phase.

R uses the code NA for missing values. Other programs (SAS, SPSS) use a single dot. Other possibilities exist. Use the na argument inside of the read function if the missing value code is anything other than NA.

Statistics involving missing values

```r
sleep |>
  summarize(
    sleep_mean=mean(TotalSleep),
    sleep_sd=sd(TotalSleep))
```

```
# A tibble: 1 × 2
  sleep_mean sleep_sd
       <dbl>    <dbl>
1         NA       NA
```

Speaker notes

If you try to compute a summary statistic on a variable with missing data, the result will be missing. This is different than SAS or SPSS, which by default will ignore missing values when computing most statistics.

The na.rm argument

```r
sleep |>
  summarize(
    sleep_mean=mean(TotalSleep, na.rm=TRUE),
    sleep_sd=sd(TotalSleep, na.rm=TRUE))
```

```
# A tibble: 1 × 2
  sleep_mean sleep_sd
       <dbl>    <dbl>
1       10.5     4.61
```

Speaker notes

Think carefully about this and if you want R to ignore missing values when computing statistics, use the na.rm=TRUE argument.

You may wish to use some sort of imputation model instead of ignoring missing values. Imputation is beyond the scope of this class.

The na.action and use arguments, 1

- More options for complex settings
    - na.action or use arguments
    - Example, cor function
        - use="everything" produces NA
        - use="complete.obs" produces casewise deletion
        - use="pairwise.complete.obs" produces pairwise deletion

Speaker notes

For univariate functions, there are only two realistic ways to handle missing values, but for bivariate and multivariate function, there are a multitude of approaches, such as pairwise deletion, listwise deletion, last observation carried forward, single imputation, and multiple imputation. There is a lot of controversy over various methods for handling missing values.

Correlations with use="everything"

```r
options(width=120)
# The greater output length allows the correlation matrix to fit on one slide.
sleep |>
  select(-1) |>
  cor(use="everything") |>
  round(2)
```

Correlations with use="complete.obs"

```r
sleep |>
  select(-1) |>
  cor(use="complete.obs") |>
  round(2)
options(width=80)
# Reset the width to the default value of 80
```

You cannot test missingness directly

```r
sleep |>
  filter(TotalSleep == NA) |>
  select(Species, TotalSleep)
```

```
# A tibble: 0 × 2
# ℹ 2 variables: Species <chr>, TotalSleep <dbl>
```

Speaker notes

Logic involving missing values is tricky. If you checking for equality and one of the things in the comparison is missing, then the result is neither TRUE, not FALSE, but rather missing.

Fair enough, but R takes it a bit further, and if both sides when you are checking for equality are missing, then they might both be 5 is they weren’t missing or maybe one might be 5 and the other one 10. So it might be TRUE or it might be FALSE, so we’re better off calling the logical result as missing.

This is called a three valued logic system and it has advantages and disadvantages. I won’t get into any technical details, except to say that you should never make assumptions. Check what you do when you are working with missing values to make sure that the three valued logic system doesn’t produce results that you didn’t expect.

What is happening

```r
sleep$TotalSleep == NA
```

```
 [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
NA
[26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
NA
[51] NA NA NA NA NA NA NA NA NA NA NA NA
```

The is.na function

```r
sleep |>
  filter(is.na(TotalSleep)) |>
  select(Species, TotalSleep)
```

Counting missing values

```r
sleep |>
  select(where(is.numeric)) |>
  mutate(across(everything(), is.na)) |>
  summarize(across(everything(), sum)) |>
  pivot_longer(
    everything(),
    names_to="variable",
    values_to="missing_count") |>
  filter(missing_count > 0)
```

Live demo, part 3

Break #3

- What you have learned
  - Missing values
- What’s coming next
  - Histograms

Basic histogram

```r
sleep |>
  ggplot() +
  aes(x=BodyWt) +
  geom_histogram(
    bins=8,
    color="black",
    fill="white") -> better_histogram
```

Labeled histogram

```r
better_histogram +
  ggtitle("Graph drawn by Steve Simon on 2025-01-02") +
  xlab("Body weight (kg)") -> labeled_histogram
```

Extra tick marks

```r
xticks <- seq(0, 6500, 500)
yticks <- seq(0, 50, 5)
labeled_histogram +
  scale_x_continuous(
    breaks=xticks,
    labels=format(xticks, big.mark=",")) +
  scale_y_continuous(breaks=yticks) -> extra_ticks
```

Log transformed histogram

```r
ticks <- 10^(-2:3)
labeled_histogram +
  scale_x_log10(
    breaks=ticks,
    minor_breaks=NULL,
    labels=ticks) -> histogram_with_log10
```

Live demo, part 4

Break #4

- What you have learned
  - Histograms
- What's coming next
  - Scatterplots

Labeled scatterplot

```r
sleep |>
  ggplot() +
  aes(x=BodyWt, y=BrainWt) +
  geom_point() +
  ggtitle("This graph was drawn by Steve Simon on 2025-01-02") +
  xlab("Body weight (kg)") +
  ylab("Brain weight (g)") -> labeled_scatterplot
```

Log transformed scatterplot

```r
xticks <- 10^(-2:3)
yticks <- 10^(-2:3)
labeled_scatterplot +
  scale_x_log10(
    breaks=xticks,
    minor_breaks=NULL,
    labels=xticks) +
  scale_y_log10(
    breaks=yticks,
    minor_breaks=NULL,
    labels=yticks)-> log_scatterplot
```

Add a trend line

```r
log_scatterplot +
  geom_smooth(method="gam") -> trend_line_graph
```

Live demo, part 5

Break #5

- What you have learned
  - Scatterplots
- What's coming next
  - Your homework

Program

- Download the template program
    - Store it in your src folder
- Modify the file names
    - Use your last name instead of “simon”
- Modify the documentation headers
    - Add your name
    - Optional: change the copyright statement

Data

- Download the data file
  - Store it in your data folder
- Refer to the data dictionary, if needed.

Question 1

There is an unusually low value for ht, 29.5 inches. Display information about the case, age, wt, ht, and bmi for this person (but no other variables). Do you think this is a legitimate value?

Question 2

Calculate the mean, standard deviation, minimum, and maximum values for age and bmi. Round the values appropriately. Interpret these numbers.

Question 3

The data dictionary implies that there are no missing values in the dataset. Write some code that checks whether this is true or not. Display the missing value counts in a vertical format.

Question 4

Display a scatterplot with age on the x-axis and bmi on the y-axis. Be sure to include units of measurement on each axis. Include a smooth trend line. Interpret this graph.

Grading rubric

You will be evaluated using the general grading rubric for programming assignments.

Your submission

- Save the output in html format
- Convert it to pdf format.
- Make sure that the pdf file includes
  - Your last name
  - The number of this course
  - The number of this module
- Upload the file

If it doesn’t work

Please review the suggestions if you encounter an error page.

File details

This programming assignment was written by Steve Simon on 2024-12-18 and is placed in the public domain.

Summary

- What you have learned
  - Using select, slice, filter
  - Using mutate, summarize
  - Missing values
  - Histograms
  - Scatterplots
  - Your homework