

Tutorial 6: Refactoring R Code

Introduction

In this tutorial, you will refactor the code into separate scripts corresponding to each section. The dataset we will use comes from the `palmerpenguins` package, which contains measurements of penguins from three species.

Load Libraries and Data

```
```{r}
library(tidyverse)
```
```

```
-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr      1.1.4      v readr      2.1.5
v forcats    1.0.0      v stringr    1.5.1
v ggplot2    3.5.1      v tibble     3.2.1
v lubridate  1.9.4      v tidyr      1.3.1
v purrr      1.0.4
-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()     masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
```

```
```{r}
library(palmerpenguins)
library(tidymodels)
```
```

```
-- Attaching packages ----- tidymodels 1.3.0 --
v broom      1.0.7      v rsample      1.2.1
v dials      1.4.0      v tune        1.3.0
v infer      1.0.7      v workflows   1.2.0
v modeldata  1.4.0      v workflowsets 1.1.0
v parsnip    1.3.0      v yardstick   1.3.2
v recipes    1.1.1

-- Conflicts ----- tidymodels_conflicts() --
x scales::discard() masks purrr::discard()
x dplyr::filter()   masks stats::filter()
x recipes::fixed()  masks stringr::fixed()
x dplyr::lag()       masks stats::lag()
x yardstick::spec() masks readr::spec()
x recipes::step()    masks stats::step()
```

```
```{r}
data <- penguins

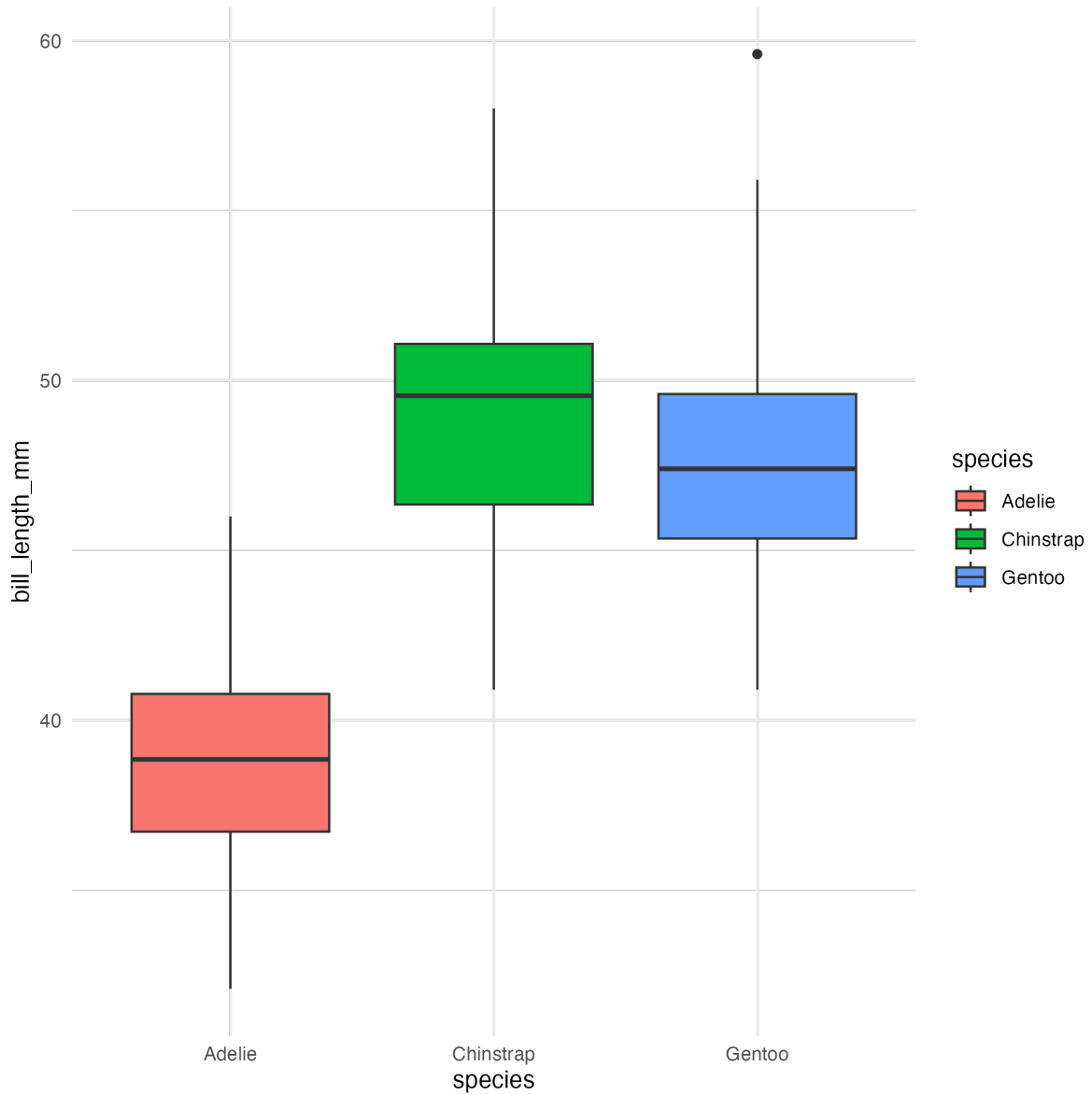
Initial cleaning: Remove missing values
data <- data %>% drop_na()
```
```

Methods

In this section, we perform exploratory data analysis (EDA) and prepare the data for modeling.

```
# Summary statistics
glimpse(data)
```

```
Rows: 333
Columns: 7
$ species      <fct> Adelie, Adelie, Adelie, Adelie, Adelie, Adelie, Adel~
$ island       <fct> Torgersen, Torgersen, Torgersen, Torgersen, Torgerse~
$ bill_length_mm <dbl> 39.1, 39.5, 40.3, 36.7, 39.3, 38.9, 39.2, 41.1, 38.6~
$ bill_depth_mm <dbl> 18.7, 17.4, 18.0, 19.3, 20.6, 17.8, 19.6, 17.6, 21.2~
$ flipper_length_mm <int> 181, 186, 195, 193, 190, 181, 195, 182, 191, 198, 18~
$ body_mass_g   <int> 3750, 3800, 3250, 3450, 3650, 3625, 4675, 3200, 3800~
$ sex          <fct> male, female, female, female, male, female, male, fe~
```



Model

We will fit a classification model using `tidymodels` to predict the species of a penguin based on its physical characteristics.

Table 1: Classification model.

| | Length | Class | Mode |
|-----|--------|-----------|------|
| pre | 3 | stage_pre | list |
| fit | 2 | stage_fit | list |

| | Length | Class | Mode |
|---------|--------|------------|---------|
| post | 1 | stage_post | list |
| trained | 1 | -none- | logical |

Results

We evaluate the performance of the model using the test dataset.

Table 2: Confusion Matrix.

| | Adelie | Chinstrap | Gentoo |
|-----------|--------|-----------|--------|
| Adelie | 36 | 0 | 0 |
| Chinstrap | 1 | 17 | 0 |
| Gentoo | 0 | 0 | 30 |

Conclusion

In this tutorial, we:

- Loaded and cleaned the `palmerpenguins` dataset.
- Performed exploratory data analysis.
- Built a k-Nearest Neighbors classification model using `tidymodels`.
- Evaluated the model's performance.