DARWIN EU(c) R programming with the OMOP CDM

Edward Burn, Adam Black, Marti Catala

2022-09-02T00:00:00+01:00

Table of contents

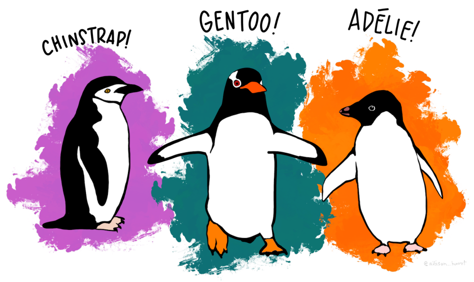
Preface

This book is written for analysts writing analytic code with R to run against the OMOP CDM. This source code for the book can be found at this Github repository Please open an issue there if you have a question or suggestion. Pull requests with suggested changes and additions are also most welcome.

1 Getting started with R

1.1 Installing R and R Studio

1.2 A first data analysis



Artwork by @allison_horst

For a quick example of a data analysis with R, let´s use the data from palmerpenguins package (https://allisonhorst.github.io/palmerpenguins/), which contains data on penguins collected from the Palmer Station in Antarctica.

Because we´ll be using a few packages not included in base R, first we need to install these if we don´t already have them.

```
install.packages("dplyr")
install.packages("ggplot2")
install.packages("palmerpenguins")
```

Once installed, we can load them like so.

```
library(dplyr)
library(ggplot2)
library(palmerpenguins)
```

We can get an overview of the data using the `glimpse()` command.

```r
glimpse(penguins)
```

```
Rows: 344
Columns: 8
$ 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, NA, 36.7, 39.3, 38.9, 39.2, 34.1, ~
$ bill_depth_mm     <dbl> 18.7, 17.4, 18.0, NA, 19.3, 20.6, 17.8, 19.6, 18.1, ~
$ flipper_length_mm <int> 181, 186, 195, NA, 193, 190, 181, 195, 193, 190, 186~
$ body_mass_g       <int> 3750, 3800, 3250, NA, 3450, 3650, 3625, 4675, 3475, ~
$ sex               <fct> male, female, female, NA, female, male, female, male~
$ year              <int> 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007, 2007~
```

Let´s get a count by species

```r
penguins %>%
  group_by(species) %>%
  count()
```

```
# A tibble: 3 x 2
# Groups:   species [3]
  species       n
  <fct>     <int>
1 Adelie      152
2 Chinstrap    68
3 Gentoo      124
```

Now suppose we are particularly interested in the body mass variable. We can first notice that there are a couple of missing records for this.

```r
penguins %>%
  group_by(species) %>%
  summarise(not_missing_body_mass_g=sum(!is.na(body_mass_g)==TRUE),
            missing_body_mass_g=sum(is.na(body_mass_g)==TRUE))
```

```
# A tibble: 3 x 3
  species   not_missing_body_mass_g missing_body_mass_g
  <fct>                       <int>               <int>
1 Adelie                        151                   1
2 Chinstrap                      68                   0
3 Gentoo                        123                   1
```

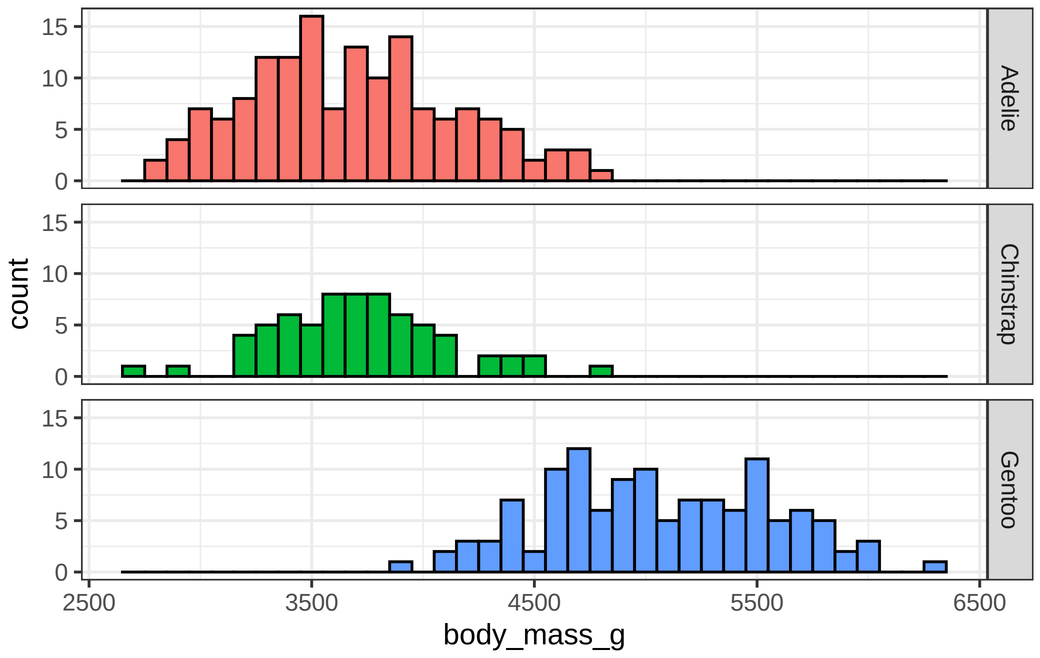
We can get the mean for each of the species (dropping those two missing records).

```
penguins %>%
  group_by(species) %>%
  summarise(mean_body_mass_g=round(mean(body_mass_g, na.rm=TRUE)))
```

```
# A tibble: 3 x 2
  species   mean_body_mass_g
  <fct>                <dbl>
1 Adelie                3701
2 Chinstrap             3733
3 Gentoo                5076
```

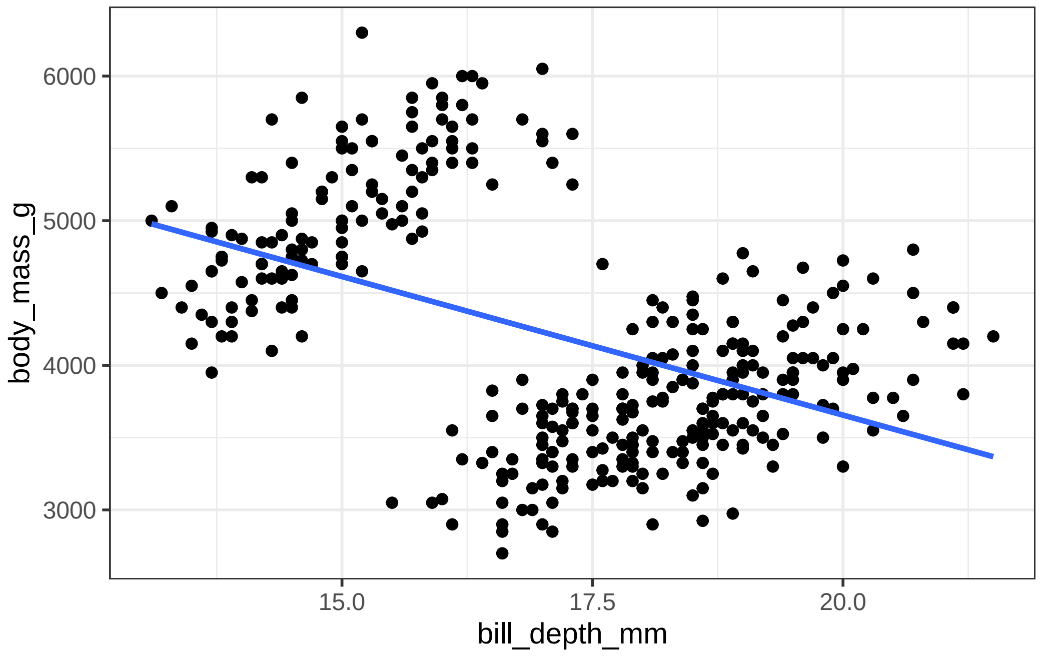
We can then also do a histogram for each of the species.

```
penguins %>%
  ggplot(aes(group=species, fill=species))+
  facet_grid(species~ .) +
  geom_histogram(aes(body_mass_g), colour="black", binwidth = 100)+
  theme_bw()+
  theme(legend.position = "none")
```



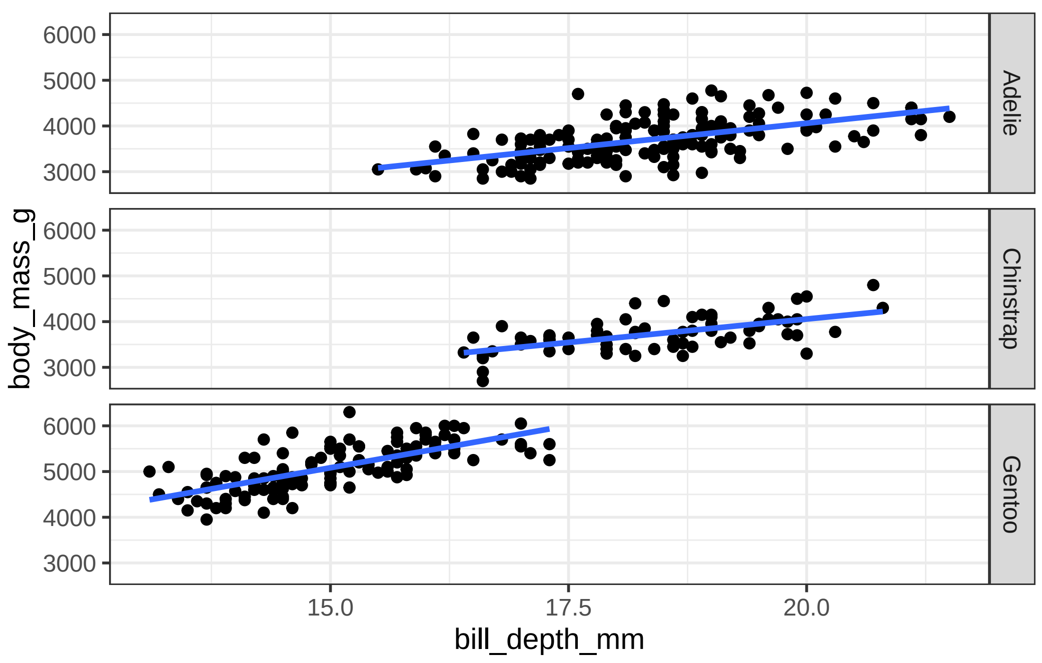
How about the relationship between body mass and bill depth?

```
penguins %>%
  ggplot(aes(x=bill_depth_mm,y=body_mass_g))+
  geom_point()+
  geom_smooth(method="lm",se=FALSE )+
  theme_bw()+
  theme(legend.position = "none")
```



But what about by species?

```
penguins %>%
  ggplot(aes(x=bill_depth_mm,y=body_mass_g))+
  facet_grid(species~ .) +
  geom_point()+
  geom_smooth(method="lm",se=FALSE )+
  theme_bw()+
  theme(legend.position = "none")
```



Oh, your first data analysis and you have already found an example of Simpson´s paradox!

2 Structure of a project

2.1 Scripts, projects, and packages

2.2 Structure of an R project

2.3 Adding renv

3 Working with databases from R

3.1 CDMConnector

3.2 Dbplyr

3.2.1 show_query()

3.2.2 filter(), select(), mutate()

3.2.3 right_join(), left_join(), inner_join(), and anti_join()

3.2.4 summarise()

3.2.5 collect() and compute()

3.2.6 working with dates

3.2.7 working with strings

3.3 sqlRender

4 Working with Apache Arrow from R

The Apache Arrow project defines two data formats for tabular data. The Arrow Dataset is an in-memory format for columnar data that can be accessed and used from multiple programming languages including R, python, and Java, and more. The feather file format is an on-disk format that can be used to efficently store and manipulate larger than memory dataframes. The arrow R package provides tools for working with both of these from R.

The Andromeda R package provides a way to manipulate larger than memory dataframes in R. There is an open issue to convert Andromeda to arrow and much of that work has been completed but not yet released in OHDSI. Andromeda objects are simply a list references to on-disk feather files that can be manipulated from R using `dplyr`. Andromeda should only be used if data cannot be constrained to available RAM.

5 Analysis in R

References