An introduction to R

Data visualisation and graphics

Noémie Becker & Benedikt Holtmann

Winter Semester 16/17
Course outline

Review – Rearranging and manipulating data

Graphics with base R
  • Histograms
  • Scatterplots
  • Boxplots

Saving plots

Graphics with ggplot2
Review – Rearranging and manipulating data

Reshaping data

Package

tidyrr

reshape2

gather() melt()

spread() dcast()
### Combining data sets

**Functions to combine data sets in dplyr**

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>left_join(a, b, by = &quot;x1&quot;)</code></td>
<td>Joins matching rows from b to a</td>
</tr>
<tr>
<td><code>right_join(a, b, by = &quot;x1&quot;)</code></td>
<td>Joins matching rows from a to b</td>
</tr>
<tr>
<td><code>inner_join(a, b, by = &quot;x1&quot;)</code></td>
<td>Returns all rows from a where there are matching values in b</td>
</tr>
<tr>
<td><code>full_join(a, b, by = &quot;x1&quot;)</code></td>
<td>Joins data and returns all rows and columns</td>
</tr>
</tbody>
</table>

### Fish survey

<table>
<thead>
<tr>
<th>Site</th>
<th>Month</th>
<th>Transect</th>
<th>Species</th>
</tr>
</thead>
</table>

### Water characteristics

<table>
<thead>
<tr>
<th>Site</th>
<th>Month</th>
<th>Water temp.</th>
<th>O₂ - content</th>
</tr>
</thead>
</table>

### GPS

<table>
<thead>
<tr>
<th>Site</th>
<th>Transect</th>
<th>Latitude</th>
<th>Longitude</th>
</tr>
</thead>
</table>
Adding new variables

Three ways adding a new variable (log of FID)

Using $\$
Bird_Behaviour$log_FID <- log(Bird_Behaviour$FID)

Using [ ] - operator
Bird_Behaviour[ , "log_FID" ] <- log(Bird_Behaviour$FID)

Using mutate() from dplyr package
Bird_Behaviour <- mutate(Bird_Behaviour, log_FID = log(FID))
Adding new variables

Split one column into two using `separate()` from `dplyr` package

Combine two columns using `unite()` from `tidyR` package

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1_1</td>
<td>1_1</td>
</tr>
<tr>
<td>B</td>
<td>1_2</td>
<td>1_2</td>
</tr>
<tr>
<td>A</td>
<td>2_1</td>
<td>2_1</td>
</tr>
<tr>
<td>B</td>
<td>2_2</td>
<td>2_2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2.1</th>
<th>X2.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1_1</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>1_2</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>2_1</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>2_2</td>
<td></td>
</tr>
</tbody>
</table>
**Subsetting data**

- **Using `[ ]` – operator**

- **Using `subset()`**

```r
subset(Bird_Behaviour, FID < 10)  # selects all rows with FID smaller than 10m

subset(Bird_Behaviour, FID < 10 & Sex == "male")  # selects all rows for males with FID smaller than 10m

subset(Bird_Behaviour, FID > 10 | FID < 15, select = c(Ind, Sex, Year))  # selects all rows that have a value of FID greater than 10 or less than 15. We keep only the IND, Sex and Year column
```

---

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
</tr>
<tr>
<td>==</td>
<td>equal to</td>
</tr>
<tr>
<td>!=</td>
<td>not equal to</td>
</tr>
<tr>
<td>x &amp; y</td>
<td>x and y</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
</tr>
</tbody>
</table>
Course outline

Review – Rearranging and manipulating data

Graphics with base R
- Histograms
- Scatterplots
- Boxplots

Saving plots

Graphics with ggplot2
Graphics with base R

Simple graphics using plotting functions in the \texttt{graphics package}

- Base R, installed by default
- Easy and quick to type
- Wide variety of functions
Data visualisation and graphics

Graphics with base R

Simple graphics using plotting functions in the graphics package

- Base R, installed by default
- Easy and quick to type
- Wide variety of functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hist()</td>
<td>Histograms</td>
</tr>
<tr>
<td>plot()</td>
<td>Scatterplots, etc.</td>
</tr>
<tr>
<td>boxplot()</td>
<td>Box- and whisker plots</td>
</tr>
<tr>
<td>barplot()</td>
<td>Bar- and column charts</td>
</tr>
<tr>
<td>dotchart()</td>
<td>Cleveland dot plots</td>
</tr>
<tr>
<td>contour</td>
<td>Contour of a surface (2D)</td>
</tr>
<tr>
<td>pie()</td>
<td>Circular pie chart</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Creating a **histogram** with `hist()`

**Example 1:**

```r
hist(Sparrows$Tarsus)
```

![Histogram of Sparrows$Tarsus](image)
Creating a **histogram** with `hist()`

**Example 2:** Alter **colour** and the **number of bins**

```r
hist(Sparrows$Tarsus, col = "grey", breaks = 50)
```
Creating a **histogram** with `hist()`

**Example 3**: Add **density curve**

```r
hist(Sparrows$Tarsus, 
    col="grey", 
    breaks = 50, 
    freq = FALSE)
```
Creating a **histogram** with **hist()**

**Example 3: Add density curve**

```r
hist(Sparrows$Tarsus, 
    col="grey", 
    breaks = 50, 
    freq = FALSE)
lines(density(Sparrows$Tarsus), 
    col = "blue", 
    lwd = 2)
```
Creating a **histogram** with `hist()`

**Example 4:** Plot only males

```r
hist(Sparrows[Sparrows$Sex == "Male",]$Tarsus, col = "grey", breaks = 50)
```

![Histogram of Sparrows[Sparrows$Sex == "Male",]$Tarsus](image)
Creating a **scatterplot** with **plot()**

**Example 1:**
plot(Sparrows$Wing, Sparrows$Tarsus)

- Relationship between two continuous variables
Creating a **scatterplot** with **plot()**

**Example 2: Alter axis limits and shape of symbols**

```r
defplot(Sparrows$Tarsus, Sparrows$Wing,
        xlim = c(50, 70),
        pch = 15,
        col = "blue")
```

![Symbol shapes](image)
Creating a **scatterplot** with **plot()**

**Example 2:**

```r
plot(Sparrows$Tarsus, Sparrows$Wing, xlim = c(50, 70), pch = 15, col = "blue")
```
Creating a scatterplot with plot()

Example 3: Alter the size of plotting symbols

```r
plot(Sparrows$Wing, Sparrows$Tarsus, xlim = c(50, 70), cex = 1.5)
```
Creating a **line graphs** with `plot()`

**Example 1:**
```
plot(pressure$temperature, pressure$pressure)
```
```
plot(pressure$temperature, pressure$pressure, type = "l")
```
Use the **type** argument to specify the type of plot

<table>
<thead>
<tr>
<th>Possible types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;p&quot;</td>
<td>points</td>
</tr>
<tr>
<td>&quot;l&quot;</td>
<td>lines</td>
</tr>
<tr>
<td>&quot;b&quot;</td>
<td>points connected by lines</td>
</tr>
<tr>
<td>&quot;o&quot;</td>
<td>points overlaid by lines</td>
</tr>
<tr>
<td>&quot;h&quot;</td>
<td>vertical lines from points to the zero axis</td>
</tr>
<tr>
<td>&quot;s&quot;</td>
<td>steps</td>
</tr>
<tr>
<td>&quot;n&quot;</td>
<td>nothing, only the axes</td>
</tr>
</tbody>
</table>
Creating a **boxplot** with **boxplot()**

**Example 1:**

```r
boxplot(Wing ~ Sex, data = Sparrows)
```

- Relationship between continuous and categorical variables
Creating a boxplot with `boxplot()`

**Example 1:**

```r
boxplot(Wing ~ Sex, data = Sparrows)
```

- median
- 25\textsuperscript{th} and 75\textsuperscript{th} percentiles
- maximum and minimum values
- outliers
Creating a boxplot with `boxplot()`

**Example 2:**

```r
boxplot(Wing ~ Sex, data = Sparrows,
       xlab = 'Sex', # Adds label to x-axis
       ylab = 'Wing length (mm)', # Adds label to y-axis
col=c("red", "blue"), # Adds colour
ylim = c(50,70), # Changes axis limits
       main = "Boxplot") # Adds title
```
Creating a **boxplot** with `boxplot()`

Example 2:
Creating a **boxplot** with **boxplot()**

**Example 3: Multiple grouping variables**

```r
boxplot(Wing ~ Sex + Species, data = Sparrows,
       xlab = 'Species and Sex',
       ylab = 'Wing length (mm)',
       col=c("red", "blue"),
       ylim = c(50,70),
       main = "Boxplot")
```
Graphics with base R

Creating a **boxplot** with `boxplot()`

Example 3:
# Quickly exploring data

## Common parameters in graphics

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>main</td>
<td>title of the plot</td>
</tr>
<tr>
<td>xlab</td>
<td>label of x-axis</td>
</tr>
<tr>
<td>ylab</td>
<td>label of y-axis</td>
</tr>
<tr>
<td>xlim</td>
<td>range/limits of x-axis</td>
</tr>
<tr>
<td>ylim</td>
<td>range/limits of y-axis</td>
</tr>
</tbody>
</table>
| col       | colour of the points, bars, etc.  
            | can be character string or hexadecimal colour (e.g. #RRGGBB) |
| breaks    | number of bins |
| pch       | shape of symbol |
| cex       | size of symbols |
| lty       | line type |
| lwd       | line width |
Multiple plots on one page

The `par()` function

- comes with an extensive list of graphical parameters you can change (see `?par`)
- Some options are helpful; others you may never use

To plot multiple charts within the same window, you can use the `mfcol` or `mfrow` parameter

For example, `par(mfrow=c(2, 2))` divides the graphic window into four panels (two rows and two columns)
Data visualisation and graphics

Multiple plots on one page

```r
par(mfrow = c(2,2))
```

```r
par(mfcol = c(2,2))
```
Saving plots

There are several possibilities saving a plot

1. `dev.print()`

Example:
```
plot(x, y, ....)  # Make a plot
```

# After you are finished with the plot use:
```
dev.print(device = pdf, file = "filename.pdf")
```

Important:
**When you are done, you have to close the printing device!**
```
dev.off()    # shuts down current device
```
Saving plots

There are several possibilities saving a plot

2. `savePlot()`

Example:

```r
plot(x, y, ....)  # Make a plot

savePlot(filename = "Figure1.pdf", type = "pdf")
```

Important:

It is possible that it does not work for your system!
(uses X11 device, most Unix systems)
There are several possibilities saving a plot

3. Plot directly into a file

Example:
```r
pdf("Figure2.pdf", width = 4, height = 4)  # width and height are in inches

hist(x)  # You can execute multiple graphing commands
plot(x, y, ...)  # The result of each will go into the pdf file

dev.off()
```

But file is not printed on screen!
There are several possibilities saving a plot.

3. **Plot directly into a file**

<table>
<thead>
<tr>
<th>Functions to save plots</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdf()</td>
<td>Opens a pdf-file as device</td>
</tr>
<tr>
<td>postscript()</td>
<td>Opens a postscript-file as device</td>
</tr>
<tr>
<td>png()</td>
<td>Opens a png-file as device</td>
</tr>
<tr>
<td>jpeg()</td>
<td>Opens a jpeg-file as device</td>
</tr>
<tr>
<td>tiff()</td>
<td>Opens a tiff-file as device</td>
</tr>
<tr>
<td>bmp()</td>
<td>Opens a bmp-file as device</td>
</tr>
</tbody>
</table>
Why use \texttt{ggplot2}?

- Many users, a lot of support
- Check out the \texttt{ggplot2} documentation at \url{http://docs.ggplot2.org/}
- Very flexible and powerful
- Sophisticated plots for publication
Graphics with ggplot2

To create a plot you use the `ggplot()` function

**Basic structure:**

```r
ggplot(data, 
aes(x variable, y variable)) + 
geom_object()  
```

# data frame with variables to plot
# specifies which variables to plot
# specifies the geometric objects

**Commonly used geometric objects:**

- **Histogram:** `+ geom_histogram()`
- **Scatterplot:** `+ geom_point()`
- **Boxplot:** `+ geom_boxplot()`
Creating a histogram with ggplot()

Example 1:
```r
ggplot(Sparrows, aes(Tarsus)) +
  geom_histogram(col = "grey", binwidth = 0.1) +
  xlab("Tarsus length (mm)") +
  ylab("Frequency")
```
Creating a **scatterplot** with *ggplot()*

**Example 1:**

```r
ggplot(Sparrows,
      aes(x = Wing,
          y = Tarsus)) +
      geom_point()
```
Creating a **scatterplot** with `ggplot()`

**Example 2:** Avoid overplotting of symbols

```r
ggplot(Sparrows, 
aes(x = Wing, y = Tarsus)) + geom_point(position=position_jitter(width=0.5, height=0))
```
Creating a **scatterplot** with **ggplot()**

**Example 2:** Avoid overplotting of symbols
Creating a **scatterplot** with **ggplot()**

**Example 3:** Alter **colour**, **shape**, and **size** of symbols

```r
ggplot(Sparrows, aes(x = Wing, y = Tarsus, colour = Sex, shape = Species)) + geom_point(size = 2)
```
Creating a **scatterplot** with **ggplot()**

**Example 3:**

- **Species:** SESP, SSTS
- **Sex:** Female, Male

Data visualisation and graphics with ggplot2
Creating a **boxplot** with **ggplot()**

**Example 1:**

```r
ggplot(Sparrows, aes(Sex, Wing, fill=Sex)) + geom_boxplot()
```
Preparing plots for publication

- Title and axis labels
- Range of axes
- Colours
- Overall appearance (themes)
- Text size
- Legend
Preparing plots for publication

• Title and axis labels
• Range of axes
• Colours
• Overall appearance (themes)
• Text size
• Legend

Sparrow morphology

Data visualisation and graphics

Graphics with ggplot2

Wing length (mm)

Female  Male
Further reading

Books

- *R in a Nutshell* by Joseph Adler
- *R Graphics Cookbook* by Winston Chang
- *A Beginner's Guide to R* by Main E. Yee, Elena N. Seijo, and Erik M. L. Meentens

Internet

- [http://docs.ggplot2.org/](http://docs.ggplot2.org/)
- [http://www.cookbook-r.com/](http://www.cookbook-r.com/)
- [marianattestad.com/blog](http://marianattestad.com/blog)