

SESSION 3: ANALYSING GENOMIC AND TRANSCRIPTOMIC DATA IN R

A mini-course

Objectives

1. If else and loops
2. Base R plotting
3. Ggplot2

Loops

- Loop functions in R allow you to iterate over elements in data structures
 - `for() {`
`}`
 - `while(){`
`}`

For loop

- Syntax:

```
for (variable in sequence) { code }
```

Try

```
for (i in 1:5) {  
  print(i)  
}
```

While loops

- Syntax:

```
while (condition) { code }
```

Try

Example of a while loop in R

```
i <- 1
```

```
while (i <= 5) {
```

```
  print(i)
```

```
  i <- i + 1
```

```
}
```

Conditional statements

- Conditional statements allow you to execute code based on certain conditions
- Commonly used conditional statements: if, else if, else

If statement

- The if statement allows you to execute code if a condition is TRUE
- Syntax:

```
if (condition) { code }
```

Try

```
x <- 5
```

```
if (x > 0) {
```

```
  print("Positive number")
```

```
}
```

If...else statements

- The if...else statement allows you to execute one block of code if a condition is TRUE, and another block if it is FALSE

- Syntax:

```
if (condition) { code1 }  
  
else { code2 }
```

Try

```
x <- -5  
if (x > 0) {  
  print("Positive number")  
} else {  
  print("Not a positive number")  
}
```


if...else if...else Statement

- The if...else if...else statement allows you to test multiple conditions
- Syntax:

```
if (condition1) { code1 } else if (condition2) { code2 } else { code3 }
```

Try

```
x <- 0  
if (x < 0) {  
  print("Negative number")  
} else if (x > 0) {  
  print("Positive number")  
} else {  
  print("Zero")  
}
```

Exercise 1

Objectives

- ✓ If else and loops
- 2. Base R plotting
- 3. Ggplot2

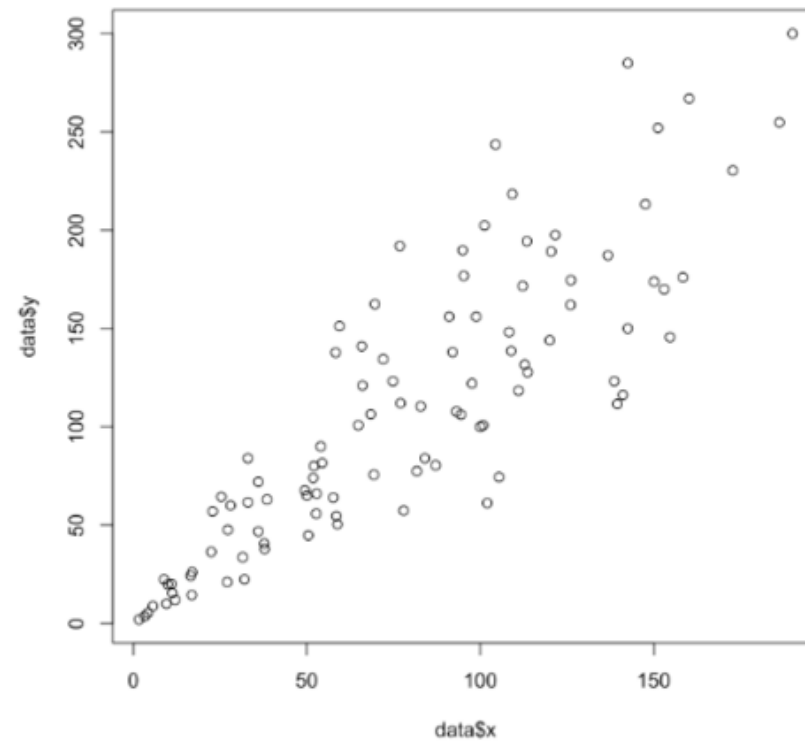
Base R plots

- `plot()`: General purpose plotting function.
- `hist()`: For creating histograms.
- `boxplot()`: For creating boxplots.

General syntax

Argument	
main	Plot title
xlab	x-axis label
ylab	y-axis label
abline()	line of best fit

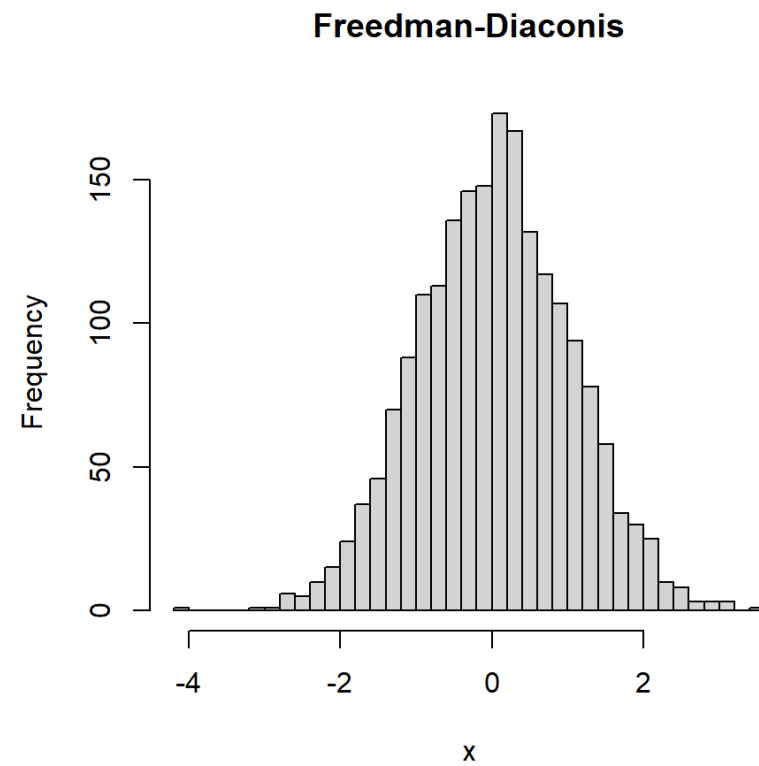
Scatterplot



Changing plot aesthetics in base R

Feature	Function
col=	Adds colour
pch=	Changes point type e.g. to a solid circle
abline()	adds line to the scatter plot

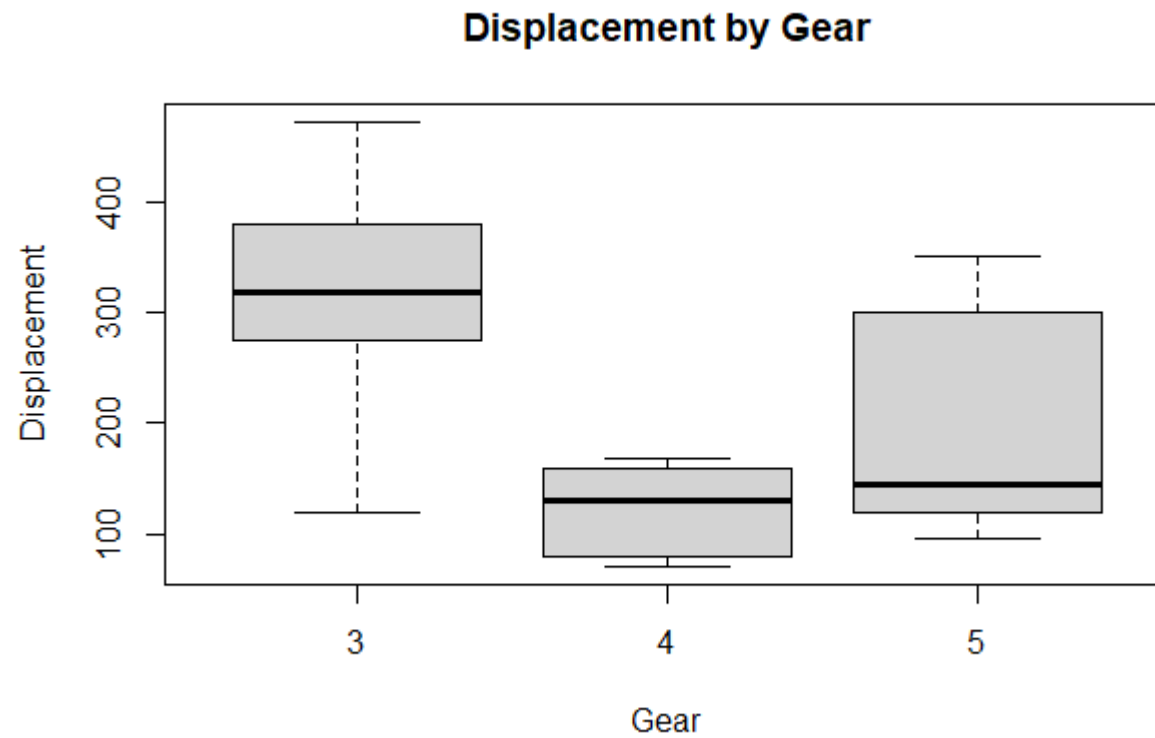
Histogram



Changing histogram aesthetics in base R

Argument	Function
col=	Adds colour
breaks=	Controls the number of bars in a histogram

Boxplots



Exercise 2

Objectives

- ✓ If else and loops
- ✓ Base R plotting
- 3. Ggplot2

Ggplot2 package

- `ggplot()` creates the global plot
- Use `'geom_'` with `ggplot()` to specify which type of plot. E.g.
 - `geom_point()`
 - `geom_bar()`
 - `geom_boxplot()`
- `ggsave` to save file

ggplot2 syntax

```
ggplot(data, mapping) +
```

Global mapping
– applies to the
entire plot

```
  geom_point(mapping, stat, position, colour, fill, shape, size, alpha) +
```

Geom-specific

```
  coord_cartesian() +
```

```
  scale_colour_discrete() +
```

```
  facet_wrap() +
```

```
  theme_bw() +
```

```
  ggtitle() +
```

```
  labs()
```

Where,

```
mapping = aes(x, y, colour, fill, shape, size, alpha)
```

Note: Colour, shape, alpha, etc. can be passed

- 1) to the **mapping** argument - changes attributes based on some variable in the dataset itself (e.g. colour = location, or shape = hdi_class)
- 2) directly as specific arguments in the geom function - changes the attributes to fixed values that are provided separately from the data (e.g. colour = "red" or size = 10)

Example

```
# Colour points by Species, shape points by Species, and add a smooth regression line
ggplot(data = iris, aes(x = Sepal.Length, y = Sepal.Width, color = Species, shape = Species)) +
  geom_point(size = 3, alpha = 0.6) + # Adjust the size and transparency of the points
labs(
  title = "Sepal Length vs Sepal Width",
  x = "Sepal Length (cm)",
  y = "Sepal Width (cm)",
  caption = "Data source: iris dataset"
) +
theme_minimal() + # Use a minimal theme
theme(
  plot.title = element_text(hjust = 0.5), # Center the plot title
  legend.position = "bottom" # Place the legend at the bottom
)
```



<https://ggplot2.tidyverse.org/reference/index.html>

Faceting plots

- Faceting allows you to create multiple subplots based on a factor variable, enabling comparison across categories.

Functions:

- `facet_wrap(~ variable)`: Creates a series of plots wrapped into a 2D layout.
- `facet_grid(rows ~ cols)`: Creates a grid of plots based on row and column factors.

Faceting plots

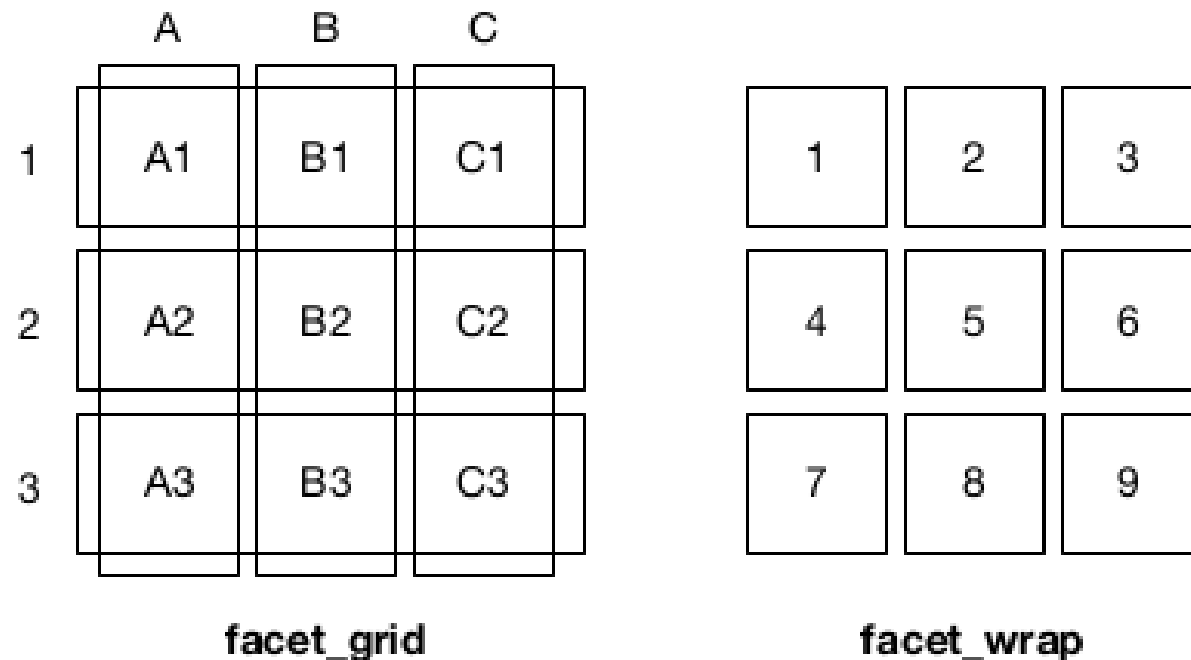


Fig. 7.1 A sketch illustrating the difference between the two facetting systems. `facet_grid()` (left) is fundamentally 2d, being made up of two independent components. `facet_wrap()` (right) is 1d, but wrapped into 2d to save space.

Theming plots

- Theming allows you to customize the appearance of your plots for better presentation and readability.

Functions:

- `theme()`: Base function for customizing themes.
- `theme_*()`: Predefined themes (e.g., `theme_bw()`, `theme_minimal()`, `theme_classic()`).
- Customize text, axes, legend, grid lines, and more.

Saving plots

`ggsave()`

Parameters:

- `filename`: Name of the file to save the plot.
- `plot`: The plot to save (if not the last created plot).
- `device`: File format (e.g., "png", "pdf", "svg").
- `width`, `height`, `units`: Dimensions and units of the saved plot.

Exercise 3

Objectives

- ✓ If else and loops
- ✓ Base R plotting
- ✓ Ggplot2