

# BST02: Using R for Statistics in Medical Research

## Part D: Statistics with R

Nicole Erler

Department of Biostatistics, Erasmus Medical Center

✉ [n.erler@erasmusmc.nl](mailto:n.erler@erasmusmc.nl)

24 - 28 February 2020

## In this Section

---

- ▶ Common statistical tests
  - ▶ for continuous data
  - ▶ for categorical data
- ▶ (Generalized) linear regression
- ▶ Useful functions for regression models

## t-test: `t.test()`

---

### One-sample t-test

- compares the **mean of one sample** with a fixed value  $\mu$

## t-test: `t.test()`

---

### One-sample t-test

- ▶ compares the **mean of one sample** with a fixed value  $\mu$

### Two sample / independent samples t-test

- ▶ compares the **difference between the means** of two samples with a fixed value  $\mu$

## t-test: `t.test()`

---

### One-sample t-test

- ▶ compares the **mean of one sample** with a fixed value  $\mu$

### Two sample / independent samples t-test

- ▶ compares the **difference between the means** of two samples with a fixed value  $\mu$

### Related samples t-test

- ▶ compares the **mean of the difference** between related observations with a fixed value  $\mu$  (same as one-sample t-test)

## Wilcoxon Test: `wilcox.test()`

---

### Wilcoxon Signed Rank Test

- tests if **one sample** (or the differences between two paired samples) is/are **symmetric about**  $\mu$

## Wilcoxon Test: `wilcox.test()`

---

### Wilcoxon Signed Rank Test

- ▶ tests if **one sample** (or the differences between two paired samples) is/are **symmetric about  $\mu$**

### Wilcoxon Rank Sum Test / Mann-Whitney test

- ▶ test for a **location shift between the distributions** of two independent samples

See also BBR Sections 7.2 & 7.3 (<http://hbiostat.org/doc/bbr.pdf>)

## Kruskal-Wallis Rank Sum Test: `kruskal.test()`

---

- ▶ **extension** of the Wilcoxon rank sum test for **more than two groups**
- ▶ test for a **difference in location** of a continuous variable between multiple groups
- ▶ the **Wilcoxon rank sum test is a special case** of the Kruskal-Wallis rank sum test



## Other tests for continuous data

---

- ▶ **Kolmogoriv-Smirnov Test:** `'ks.test()'`  
tests if two samples are drawn from the same continuous distribution
- ▶ **Shapiro-Wilk Normality Test:** `'shapiro.test()'`
- ▶ **Friedman Rank Sum Test:** `'friedman.test()'`  
non-parametric test for two or more related samples
- ▶ ...

# Tests for Continuous Data

---

## Demo

- ▶ Tests for Continuous Data

**R** **html**

# Tests for Categorical Data / Proportions

---

## One-sample Proportion Test

- ▶ tests if the **proportion in one sample** is equal to a fixed value  $p$
- ▶ `prop.test()` and `binom.test()`

# Tests for Categorical Data / Proportions

---

## One-sample Proportion Test

- ▶ tests if the **proportion in one sample** is equal to a fixed value  $p$
- ▶ `prop.test()` and `binom.test()`

## Tests for Proportions in Multiple (independent) Groups

- ▶ tests if the **proportions in several samples** are equal
- ▶ `chisq.test()` and `fisher.test()` (when there are cells with 0)

See also BBR Sections 5.7 & 6 (<http://hbiostat.org/doc/bbr.pdf>)

# Tests for Categorical Data / Proportions

---

## Related Samples: McNemar Test

- ▶ Tests for **symmetry** in a  $2 \times 2$  table
- ▶ `mcnemar.test()`

# Tests for Categorical Data / Proportions

---

## Related Samples: McNemar Test

- ▶ Tests for **symmetry** in a  $2 \times 2$  table
- ▶ `mcnemar.test()`

## 3-Dimensional Contingency Table

- ▶ Cochran-Mantel-Haenszel Test
- ▶  $\chi^2$  test for **independence** of two nominal variables **within each stratum**
- ▶ `mantelhaen.test()`

# Tests for Categorical Data

---

## Demo

- ▶ Tests for Categorical Data **R**  
**html**

# Statistical Tests

---

## Continuous Outcomes

- ▶ `t.test()`
- ▶ `wilcox.test()`
- ▶ `kruskal.test()`
- ▶ `ks.test()`
- ▶ `friedman.test()`
- ▶ `shapiro.test()`

## Categorical Outcomes

- ▶ `prop.test()`
- ▶ `binom.test()`
- ▶ `chisq.test()`
- ▶ `fisher.test()`
- ▶ `mcnemar.test()`
- ▶ `mantelhaen.test()`

## Pairwise tests

- ▶ `pairwise.prop.test()`
- ▶ `pairwise.t.test()`
- ▶ `pairwise.wilcox.test()`

## Variance and Correlation

- ▶ `cor.test()`
- ▶ `bartlett.test()`
- ▶ `var.test()`

## Multiple Testing Adjustment

- ▶ `p.adjust()`



# Linear Regression

---

A standard linear regression model has the form

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p + \varepsilon \quad \text{with } \varepsilon \sim N(0, \sigma^2)$$

where

- ▶  $y$  is the **outcome** variable (“dependent variable”)
- ▶  $x_1, \dots, x_p$  are the **covariates** (“independent variables”)
- ▶  $\beta_0, \dots, \beta_p$  are the **regression coefficients**
  - ▶  $\beta_0$  is the intercept
  - ▶  $\beta_1, \dots, \beta_p$  estimate the effects of the covariates
- ▶  $\varepsilon$  is a vector of **residuals**, which we assume to be (approximately) normally distributed.

# Linear Regression

---

To fit a **linear regression** in R we use the function `lm()`.

The most important arguments are

- ▶ **formula:**  
a formula object
- ▶ **data:**  
a `data.frame` (optional, but usually needed)
- ▶ **subset:**  
a vector specifying which observations should be used (optional)

# Model Formula

---

A formula object has the form

```
outcome ~ linear predictor
```

for example

```
y ~ x1 + x2 + x3
```

- ▶ Variables are separated by “+” signs.
- ▶ An intercept is automatically included.
- ▶ one-sided formulas (omitting the outcome) are possible (used for random effects specification)

## Model Formula: Interactions

Interaction terms are written using “:” or “\*”.

“\*” includes the main effects and interaction terms, i.e.,

```
y ~ x1 * x2
```

is equivalent to

```
y ~ x1 + x2 + x1:x2
```

Interactions between multiple variables can be written using “()”, i.e.,

```
y ~ x1 * (x2 + x3)
```

is equivalent to

```
y ~ x1 * x2 + x1 * x3
```

## Model Formula: Interactions

To specify a **higher level interaction** (for example a three-way interaction) “^” is used, i.e.,

```
y ~ (x1 + x2 + x3)^3
```

will create all interactions up to 3-way and is equivalent to

```
y ~ x1 * x2 * x3
```

and equivalent to

```
y ~ x1 + x2 + x3 + x1:x2 + x1:x3 + x2:x3 + x1:x2:x3
```

and

```
y ~ (x1 + x2 + x3)^2
```

will create all two-way interactions and is equivalent to

```
y ~ x1 + x2 + x3 + x1:x2 + x1:x3 + x2:x3
```

## Model Formula: Removing terms

---

The “-” sign can be used to remove terms from a model formula, for example

```
y ~ x1 * x2 * x3 - x2 - x1:x3
```

is equivalent to

```
y ~ x1 + x3 + x1:x2 + x2:x3 + x1:x2:x3
```

The **intercept** can be removed from a formula by using “-1” or “+0”, i.e.

```
y ~ x1 + x2 - 1
```

```
y ~ x1 + x2 + 0
```

# Generalized Linear Regression (GLM)

---

A **generalized linear regression** model has the form

$$g(\mathbb{E}(y)) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

where  $g(\cdot)$  is a link function and  $y$  is from the exponential family.

For example **logistic regression** for binary  $y$ :

$$\log \left( \frac{P(y = 1)}{1 - P(y = 1)} \right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p$$

$\log \left( \frac{p}{1-p} \right)$  is the **logit** link.

# Generalized Linear Regression (GLM)

---

To fit a **GLM** in R we use the function `glm()`.

The most important arguments are

- ▶ **formula:**  
a formula object
- ▶ **family:**  
a family object or name of the family function, describing the error distribution and link function
- ▶ **data:**  
a `data.frame` (optional, but usually needed)
- ▶ **subset:**  
a vector specifying which observations should be used (optional)



# Families and Link Functions

Common families & available links in R:

(see also ?family)

family	link
binomial	logit, probit, cauchit, log, cloglog
gaussian	identity, log, inverse
Gamma	inverse, identity, log
poisson	log, identity, sqrt

The `family` argument in `glm()` can be specified in the following ways:

- ▶ `binomial(link = "logit")`
- ▶ `binomial()`
- ▶ `binomial`
- ▶ `"binomial"`

## Note:

When the link is not explicitly specified (i.e. option 1), the default link is used.

# Regression

---

## Demo

- ▶ Regression Basics [R](#) [html](#)

## Practical

- ▶ Linear Regression [html](#)
- ▶ Name of practical [html](#)

# Regression

---

## Regression Models

- ▶ `lm()`
- ▶ `glm()`

## Regression Results

- ▶ `summary()`
- ▶ `coef()`, `confint()`
- ▶ `fitted()`, `residuals()`
- ▶ `AIC()`, `BIC()`
- ▶ `anova()`

## Plots

- ▶ `plot()`
- ▶ `qqnorm()`, `qqline()`, `qqplot()`

## Topic

- ▶ `ns()`, `bs()`, `I()`
- ▶ `p.adjust()`
- ▶ `all.vars()`
- ▶ `update()`
- ▶ `as.formula()`