Simple statistical tests

This page demonstrates how to conduct simple statistical tests using **base** R, **rstatix**, and **gtsummary**.

- T-test
- Shapiro-Wilk test
- Wilcoxon rank sum test
- Kruskal-Wallis test
- Chi-squared test
- Correlations between numeric variables

…many other tests can be performed, but we showcase just these common ones and link to further documentation.

Each of the above packages bring certain advantages and disadvantages:

- Use **base** R functions to print a statistical outputs to the R Console
- Use **rstatix** functions to return results in a data frame, or if you want tests to run by group
- Use **gtsummary** if you want to quickly print publication-ready tables

Preparation

Load packages

This code chunk shows the loading of packages required for the analyses. In this handbook we emphasize `p_load()` from **pacman**, which installs the package if necessary *and* loads it for use. You can also load installed packages with `library()` from **base** R. See the page on [R basics] for more information on R packages.

```
pacman::p_load(
  rio,          # File import
  here,         # File locator
  skimr,        # get overview of data
  tidyverse,     # data management + ggplot2 graphics,
  gtsummary,     # summary statistics and tests
  rstatix,       # statistics
  corrr,         # correlation analayis for numeric variables
  janitor,       # adding totals and percents to tables
  flextable      # converting tables to HTML
  )
```

Import data

We import the dataset of cases from a simulated Ebola epidemic. If you want to follow along, click to download the "clean" linelist (as .rds file). Import your data with the `import()` function from the **rio** package (it accepts many file types like .xlsx, .rds, .csv - see the [Import and export] page for details).

```
# import the linelist
linelist <- import("linelist_cleaned.rds")
```

base R

You can use **base** R functions to conduct statistical tests. The commands are relatively simple and results will print to the R Console for simple viewing. However, the outputs are usually lists and so are harder to manipulate if you want to use the results in subsequent operations.

T-tests

A t-test, also called "Student's t-Test", is typically used to determine if there is a significant difference between the means of some numeric variable between two groups. Here we'll show the syntax to do this test depending on whether the columns are in the same data frame.

Syntax 1: This is the syntax when your numeric and categorical columns are in the same data frame. Provide the numeric column on the left side of the equation and the categorical column on the right side. Specify the dataset to `data =`. Optionally, set `paired = TRUE`, and `conf.level =` (0.95 default), and `alternative =` (either "two.sided", "less", or "greater"). Enter `?t.test` for more details.

```
## compare mean age by outcome group with a t-test
t.test(age_years ~ gender, data = linelist)
```

```
    Welch Two Sample t-test

data:  age_years by gender
t = -21.344, df = 4902.3, p-value < 2.2e-16
alternative hypothesis: true difference in means between group f and group m is not equal to
95 percent confidence interval:
 -7.571920 -6.297975
sample estimates:
mean in group f mean in group m
       12.60207        19.53701
```

Syntax 2: You can compare two separate numeric vectors using this alternative syntax. For example, if the two columns are in different data sets.

```
t.test(df1$age_years, df2$age_years)
```

You can also use a t-test to determine whether a sample mean is significantly different from some specific value. Here we conduct a one-sample t-test with the known/hypothesized population mean as `mu =`:

```
t.test(linelist$age_years, mu = 45)
```

Shapiro-Wilk test

The Shapiro-Wilk test can be used to determine whether a sample came from a normally-distributed population (an assumption of many other tests and analysis, such as the t-test). However, this can only be used on a sample between 3 and 5000 observations. For larger samples a quantile-quantile plot may be helpful.

```
shapiro.test(linelist$age_years)
```

Wilcoxon rank sum test

The Wilcoxon rank sum test, also called the Mann–Whitney U test, is often used to help determine if two numeric samples are from the same distribution when their populations are not normally distributed or have unequal variance.

```
## compare age distribution by outcome group with a wilcox test
wilcox.test(age_years ~ outcome, data = linelist)
```

```
    Wilcoxon rank sum test with continuity correction

data:  age_years by outcome
W = 2501868, p-value = 0.8308
alternative hypothesis: true location shift is not equal to 0
```

Kruskal-Wallis test

The Kruskal-Wallis test is an extension of the Wilcoxon rank sum test that can be used to test for differences in the distribution of more than two samples. When only two samples are used it gives identical results to the Wilcoxon rank sum test.

```
## compare age distribution by outcome group with a kruskal-wallis test
kruskal.test(age_years ~ outcome, linelist)
```

```
    Kruskal-Wallis rank sum test

data:  age_years by outcome
Kruskal-Wallis chi-squared = 0.045675, df = 1, p-value = 0.8308
```

Chi-squared test

Pearson's Chi-squared test is used in testing for significant differences between categorical croups.

```
## compare the proportions in each group with a chi-squared test
chisq.test(linelist$gender, linelist$outcome)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  linelist$gender and linelist$outcome
X-squared = 0.0011841, df = 1, p-value = 0.9725
```

Correlations

Correlation between numeric variables can be investigated using the **tidyverse corrr** package. It allows you to compute correlations using Pearson, Kendall tau or Spearman rho. The package creates a table and also has a function to automatically plot the values.

```
correlation_tab <- linelist %>%
  select(generation, age, ct_blood, days_onset_hosp, wt_kg, ht_cm) %>%   # keep numeric va
  correlate()      # create correlation table (using default pearson)

correlation_tab    # print
```

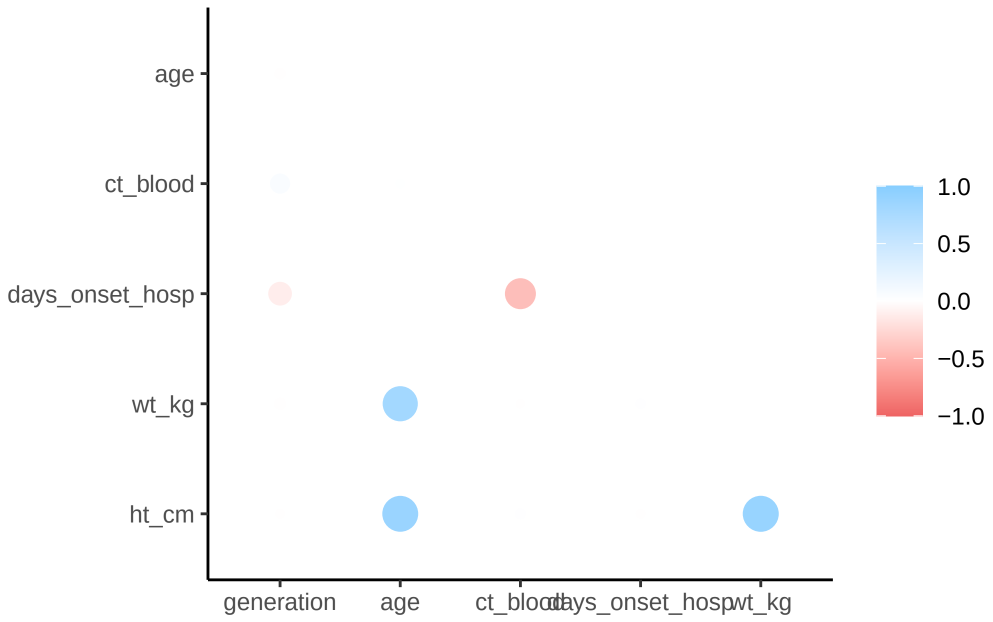
```
# A tibble: 6 x 7
  term            generation      age ct_blood days_onset_hosp    wt_kg    ht_cm
  <chr>                <dbl>    <dbl>    <dbl>           <dbl>    <dbl>    <dbl>
1 generation        NA       -2.22e-2  0.179         -0.288    -0.0302  -0.00942
2 age               -0.0222  NA        0.00849       -0.000635  0.833    0.877
3 ct_blood           0.179    8.49e-3 NA             -0.600    -0.00636  0.0181
4 days_onset_hosp   -0.288   -6.35e-4 -0.600         NA         0.0153  -0.00953
5 wt_kg             -0.0302   8.33e-1 -0.00636        0.0153   NA        0.884
6 ht_cm             -0.00942  8.77e-1  0.0181        -0.00953   0.884   NA
```

```
## remove duplicate entries (the table above is mirrored)
correlation_tab <- correlation_tab %>%
  shave()

## view correlation table
correlation_tab
```

```
# A tibble: 6 x 7
  term            generation       age ct_blood days_onset_hosp  wt_kg ht_cm
  <chr>                <dbl>     <dbl>    <dbl>           <dbl>  <dbl> <dbl>
1 generation        NA       NA         NA               NA      NA       NA
2 age               -0.0222  NA         NA               NA      NA       NA
3 ct_blood           0.179    0.00849   NA               NA      NA       NA
4 days_onset_hosp   -0.288   -0.000635 -0.600            NA      NA       NA
5 wt_kg             -0.0302   0.833    -0.00636           0.0153 NA       NA
6 ht_cm             -0.00942  0.877     0.0181          -0.00953  0.884   NA
```

```
## plot correlations
rplot(correlation_tab)
```



Resources

Much of the information in this page is adapted from these resources and vignettes online:

gtsummary dplyr corrr sthda correlation