R Cheatsheet

STAT218

|  |
| --- |
| Note |
| This document is a work in progress and will be updated prior to each test. |

### Basics

Loading data:

* [.RData file] load('<FILEPATH>')
* [from package] data(<DATASET NAME>, package = '<PACKAGE NAME>')
* [reading a CSV file] read.csv('<FILEPATH>')

Viewing dataframes:

* [preview] head(<NAME>)
* [data viewer] view(<NAME>)
* [check structure] str(<NAME>)

Extracting a variable from a dataframe:

* DATAFRAME$VARIABLE

### Summary statistics

If x is a vector of values of a numeric variable…

* mean(x) computes the average
* median(x) computes the median
* min(x) and max(x) compute the minimum and maximum
* quantile(x, probs = <PERCENTILE>) computes the percentile
* summary(x) computes the five-number summary, plus the mean
* range(x) computes the range (min, max)
* IQR(x) computes the interquartile range
* var(x) computes the variance
* sd(x) computes the standard deviation

If df is a dataframe with a numeric variable y and a categorical variable x…

* df |> group\_by(x) |> summarize(<OUTPUT.NAME> = <FUNCTION>(y)) computes the statistic specified by <FUNCTION> separately for each category of the variable x (requires tidyverse package)

*See especially* [*Lab 2: Descriptive statistics*](lab2-descriptive.qmd).

### Tables

If x and y are a vectors of values of two categorical variables…

* table(x) computes the frequency distribution (counts)
* table(x) |> proportions() computes the frequency distribution (proportions)
* table(x, y) computes a contingency table
* table(x, y) |> proportions(margin = NULL) computes proportions using grand total
* table(x, y) |> proportions(margin = 1) computes proportions using row total (group by x)
* table(x, y) |> proportions(margin = 2) computes proportions using column total (group by y)

*See especially* [*Lab 1: R basics*](lab1-rbasics.qmd) *and* [*Lab 3: Bivariate summaries*](lab3-bivariate.qmd).

### Graphics

If x and y are vectors of values of two numeric variables…

* hist(x, breaks = <NUMBER OF BINS>) generates a histogram
* boxplot(x) generates a boxplot
* plot(x, y) generates a scatterplot

If x and y are vectors of values of two categorical variables…

* table(x) |> barplot() generates a bar plot (counts)
* table(x) |> proportions() |> barplot() generates a bar plot (proportions)
* table(x, y) |> proportions(margin = 2) |> barplot(legend = T) generates a stacked bar plot grouped by y
* table(y, x) |> proportions(margin = 2) |> barplot(legend = T) generates a stacked bar plot grouped by x

If x is a vector of values of a categorical variable and y is a vector of values of a numeric variable…

* boxplot(y ~ x) generates a boxplot with x on the x axis (vertical orientation)
* boxplot(y ~ x, horizontal = T) generates a boxplot with y on the x axis (horizontal orientation)

*See especially* [*Lab 2: Descriptive statistics*](lab2-descriptive.qmd) *and* [*Lab 3: Bivariate summaries*](lab3-bivariate.qmd)*.*

### One- and two-sample inference

#### Direct calculations for one-sample inference

If x is a vector of n values of a numeric variable…

* mean(x) + c(-1, 1)\*qt(1 - alpha/2, df = n - 1)\*sd(x)/sqrt(n) computes a % confidence interval
  + sd(x)/sqrt(n) is the standard error for the estimate
  + qt(1 - alpha/2, df = n - 1) is the critical value
  + for a 99% interval, use : qt(0.995, df = n - 1)
  + for a 95% interval, use : qt(0.975, df = n - 1)
  + for a 90% interval, use : qt(0.95, df = n - 1)
* tstat <- (mean(x) - mu\_0)/(sd(x)/sqrt(n)) computes the *T* statistic for a hypothesis test of against any of the three alternatives
  + for : 2\*pt(abs(tstat), lower.tail = F) computes a two-sided *p*-value
  + for : qt(1 - alpha/2, df = n - 1) computes the critical value for a level two-sided test
  + for : pt(tstat, lower.tail = F) computes an upper-sided *p*-value
  + for : qt(1 - alpha, df = n - 1) computes the critical value for a level upper-sided test
  + for : pt(tstat, lower.tail = T) computes a lower-sided *p*-value
  + for : qt(alpha, df = n - 1) computes the critical value for a level lower-sided test
* to compute quantiles or frequencies directly using the model:
  + pt(q = <QUANTILE>, df = <DEGREES OF FREEDOM>) computes the frequency of values less than q for the model
  + pt(q = <QUANTILE>, df = <DEGREES OF FREEDOM>, lower.tail = F) computes the frequency of values greater than q for the model
  + qt(p = <PROPORTION>, df = <DEGREES OF FREEDOM>) computes the th quantile for the model

*See especially* [*Lab 4: Point estimation*](lab4-estimation.qmd)*,* [*Lab 5: Intervals*](lab5-intervals.qmd)*, and* [*Lab 6: Hypothesis testing basics*](lab6-hypothesis.qmd)*.*

#### Using the t.test(...) function for one-sample inference

If x is a vector of values of a numeric variable then

t.test(x, mu = mu\_0, alternative = <DIRECTION>, conf.level = <COVERAGE>)

performs a one-sample *t* test at significance level 1 - <COVERAGE> where:

* mu\_0 is the hypothetical value for the mean
* direction can be 'less', 'greater', or 'two.sided' (in quotes)
* coverage should be : the ‘complement’ of the intended significance level for the test

Outputs are the test statistic, *p*-value, confidence interval, and point estimate.

*See especially* [*Lab 7: Directional tests*](lab7-directional.qmd)*.*

#### Using the t.test(...) function for two-sample inference

If DATA is a dataframe with variables VARIABLE (numeric) and GROUP (categorical with two categories), where GROUP distinguishes two independent samples then

t.test(VARIABLE ~ GROUP, data = DATA, mu = delta\_0,   
 alternative = <DIRECTION>, conf.level = <COVERAGE>)

performs a two-sample *t* test at significance level 1 - <COVERAGE> where:

* delta\_0 is the hypothetical difference in means (often 0 but not always)
* direction can be 'less', 'greater', or 'two.sided' (in quotes)
* coverage should be : the ‘complement’ of the intended significance level for the test

Returns test statistic, *p*-value, confidence interval for the difference, and point estimates.

*See especially* [*Lab 8: Two sample inference*](lab8-twosample.qmd)*.*