RStudio Cheat Sheet

by Adela Vrtkova and Martina Litschmannova, Department of Applied Mathematics, FEECS, VŠB-TUO
via cheatography.com

Workspace, Using libraries

?boxplot
- getting help documentation for function *boxplot*

getwd()
- returning the current working directory

setwd("C:/Users/RStudio")
- setting the working directory to specified file

install.packages("packageZ")
- downloading and installing a package called *packageZ*

library(packageZ)
- activating already installed package called *packageZ*

packageZ::functionF(x)
- calling function *functionF* from specified package *packageZ*

moments, *EnvStats*, *dunn.test*, *lsr*, *openxlsx*, *car*, *epiR*
- important packages

After the hash, I can write whatever.
- writing notes into the script

Importing data

data = read.csv2("C:/Users/RStudio/data.csv")
- importing data in **csv** from specified file and saving as *data*

data = read.csv2("http://am-nas.vsb.cz/DATA/dataset.csv")
- importing data in **csv** from the internet and saving as *data*

data = readWorkbook("C:/USER/DATA/dataset.xlsx", sheet=1, startRow=4, colNames=TRUE, cols=2:9) # openxlsx package
- importing data in **xlsx**

Working with data

data = as.data.frame(data)
- saving imported data as an object of class *data.frame*

data.S = stack(data)
- transferring data table into the standard data matrix

data.S.omit = na.omit(data.S)
- omitting entire rows with missing values (NAs)

Probability distribution - Prefixes

r-	generating random numbers from the distribution
d-	probability density function $f(x)$ or probability mass function $P(X=x)$
p-	$P(X \leq x)$
q-	quantile function

Probability distribution - Discrete

-binom	Binomial distribution $Bi(n,\pi)$
-hyper	Hypergeometric distribution $H(N,M,n)$! R code requires - $H(M,N-M,n)$
-nbinom	Negative binomial distribution $NB(k,\pi)$! definition in JASP/R - number of unsuccessful trials
-pois	Poission distribution $Po(\lambda t)$

Probability distribution - Continuous

-unif	Uniform distribution $U(a,b)$
-exp	Exponential distribution $Exp(\lambda)$
-norm	Normal distribution $N(\mu,\sigma^2)$! JASP applet Distributions requires $N(\mu,\sigma^2)$! R code requires - $N(\mu,\sigma)$

EDA for a Qualitative Variable

data$group = as.factor(data$group)
- redefining group variable as *factor*

table(data$group)
- frequency table

barplot(table(data$group))
- creating a bar plot

pie(table(data$group))
- creating a pie chart

EDA for a Quantitative Variable

summary(data$values)	summary statistics
length(data$values)	sample size (attention if NAs present)
min(data$values)	minimum
mean(data$values)	arithmetic mean
quantile(data$values,probs=0.3)	30% quantile
max(data$values)	maximum
sd(data$values)	standard deviation
var(data$values)	variance
moments::skewness(data$values)	skewness
moments::kurtosis(data$values)-3	kurtosis
boxplot(data$values)	boxplot
hist(data$values)	histogram
plot(density(data$values))	plotting kernel density estimation
qqnorm(data$values); qqline(data$values)	QQ-plot

Function tapply()

tapply(dataS$values, dataS$group, mean)
- calculates the mean for *values* by *group* in *data*

tapply(dataS$values, dataS$group, quantile, probs=0.4)
- calculates the 40% quantile for *values* by *group* in *data*

tapply(dataS$values, dataS$group, moments::kurtosis)-3
- calculates the kurtosis for *values* by *group* in *data*

Statistical inference - One variable

shapiro.test(data$values)
- Shapiro-Wilk test

varTest(data$values, sigma.squared=400, alternative="two.sided", conf.level=0.95) # EnvStats package
- confidence interval for variance and one-sample Chi-squared test on variance ($H_0: \sigma^2 = 400, H_A: \sigma^2 \neq 400$)

t.test(data$values, mu=5, alternative="less", conf.level=0.95)
- confidence interval for mean and one-sample Student's t-test ($H_0: \mu = 5, H_A: \mu < 5$)

wilcox.test(data$values, mu=8, alternative="greater", conf.level=0.95, conf.int=TRUE)
- confidence interval for median and one-sample Wilcoxon test ($H_0: x_{0,5} = 8, H_A: x_{0,5} > 8$)

binom.test(x,n,p=0.18,alternative="two.sided",conf.level=0.95)
- confidence interval for probability and one-sample Binomial test (Clooper-Pearson method) ($H_0: \pi = 0.18, H_A: \pi \neq 0.18$)

RStudio Cheat Sheet

by Adela Vrtkova and Martina Litschmannova, Department of Applied Mathematics, FEECS, VŠB-TUO

via cheatography.com

Statistical inference - Two variables

`var.test(data$valuesA, data$valuesB)`

- confidence interval for the ratio of variances, F-test of equality of variances $(H_0: \sigma_A^2 = \sigma_B^2, H_A: \sigma_A^2 \neq \sigma_B^2)$

```
t.test(data$valuesA, data$valuesB, alternative="two.sided",
       var.equal=TRUE, conf.level=0.95)
```

- confidence interval for the difference of means and two-sample Student's t-test $(H_0: \mu_A = \mu_B, H_A: \mu_A \neq \mu_B)$

```
t.test(data$valuesA, data$valuesB, alternative="greater",
       var.equal=FALSE, conf.level=0.95)
```

- confidence interval for the difference of means and Aspin-Welch test $(H_0: \mu_A = \mu_B, H_A: \mu_A > \mu_B)$

```
wilcox.test(data$valuesA, data$valuesB, alternative="less",
            conf.level=0.95, conf.int=TRUE)
```

- confidence interval for the difference of medians and Mann-Whitney test $(H_0: x_{0,5}^A = x_{0,5}^B, H_A: x_{0,5}^A < x_{0,5}^B)$

`prop.test(c(x1,x2),c(n1,n2), alternative="two.sided",conf.level=0.95)`

- confidence interval for the difference of probabilities and Test of equality of probabilities $(H_0: \pi_A = \pi_B, H_A: \pi_A \neq \pi_B)$

Statistical inference - Three and more variables

`bartlett.test(dataS$values~dataS$group)`

- Bartlett's test of homogeneity of variances

`leveneTest(dataS$values~dataS$group) # car package`

- Levene's test of homogeneity of variances

`results = aov(dataS$values~dataS$group); summary(results)`

- ANOVA

`TukeyHSD(results)`

- post-hoc analysis after ANOVA (if necessary)

`kruskal.test(dataS$values~dataS$group)`

- Kruskall-Wallis test

`dunn.test(dataS$values~dataS$group, altp=TRUE) # dunn.test package`

- post-hoc analysis after Kruskal-Wallis test (if necessary)

Contingency tables

`tab = table(data$factor1, data$factor2)`

- contingency table of two categorical variables *factor1* and *factor2*

`tab = matrix(c(12,45,23,54), ncol=2, byrow=TRUE)`

- building a contingency table with *matrix* function (could be improved with *rownames* and *colnames* functions)

`mosaicplot(tab)`

- Mosaic plot

`cramersV(tab) # lsr package`

- Cramér's V measure of association

`results = chisq.test(tab); results$expected; results$p.value`

- Chi-squared test of independence in contingency tables, expected counts and p-value

`epi.2by2(tab) # epiR package`

- Chi-squared test of independence, OR, RR and their confidence intervals (dependent on the structure of the table)

Goodness-of-fit test

```
observed = c(979, 1002, 1015, 980, 1040, 984)
expected = c(1/6, 1/6, 1/6, 1/6, 1/6, 1/6)
chisq.test(observed, p=expected, rescale.p=TRUE)
```

- saving observed counts and expected probabilities, performing the test