FACULTAT DE CIÈNCIES, TECNOLOGIA I ENGINYERIES
UVIC | UVIC·UCC

STATISTICS with R

ANALYSIS OF ONE VARIABLE

Continuous variable	*x continuous variable* ○ **Summary statistics** *summary(x) # most important summary statistics* *min(x) # minimum* *max(x) # maximum* *mean(x) # mean, average* *median(x) # median* *sd(x) # standard deviation* *IQR(x) # interquartile rang* *quantile(,) # Ex. 95% percentile: quantile(x, 0.95)* ○ **Dot plot** *plot(x)* ○ **Histogram** *hist(x)* ○ **Box plot or Box-and-whisker plot** *boxplot(x)* ○ **Density function** *plot(density(x))* ○ **Empirical cumulative distribution** *plot(ecdf(x))*
Categorical variable	*x categorical variable* ○ **Frequency table** *table(x)* *prop.table(table(x)) # Table of relative frequencies* *100*prop.table(table(x)) # Table of percentages* ○ **Bar plot** *barplot(table(x))* ○ **Pie chart** *pie(table(x))*

RELATION BETWEEN TWO VARIABLES

	Relation between two variables
Continuous & continuous	*x and y continuous variables* ○ **Correlation coefficient** *cor(x,y) # Pearson correlation coefficient* *cor(x,y, method="spearman") # Spearman correlation coefficient* *cor(M, use="pairwise.complete.obs") # M is a matrix* ○ **Regression line equation** *lm(y~x)* ○ **Scatter plot and regression line** *plot(x,y) # independent before dependent (x,y)* *abline(lm(y~x)) # dependent before independent (y,x)*
Continuous & categorical	*y continuous, x categorical* ○ **Numerical summaries of the continuous variable by each category of the categorical variable** *tapply(<continuous>, <categorical>, <function>)* *# Example:* *tapply(y, x, mean) # mean of y for each category of x* *tapply(y, x, summary) # summary of y for each category of x* ○ **Multiple box plot** *boxplot(<continuous> ~<categorical>)* *# Example:* *boxplot(y~x)*
Categorical & categorical	*x and y categorical variables* ○ **2 by 2 table / Contingency table** *table(x,y) # absolute frequencies* *prop.table(table(x,y)) # total proportions* *prop.table(table(x,y),1) # row proportions* *prop.table(table(x,y),2) # column proportions* *100*prop.table(table(x,y),1) # row percentages* ○ **Bar plot** *barplot(table(x,y))* *barplot(prop.table(table(x,y)))*

RANDOM VARIABLES WITH R

$f(x)$ *or* $P(X = x)$ $\quad$ $P(X \leq x)$ $\quad$ $P(X \leq q) = \alpha$

Table 3.2: Built-in-functions for random variables used in this chapter.

Distribution	para-meters	density	distribution	quantiles	random sampling
Bin	n, p	dbinom(x, n, p)	pbinom(x, n, p)	qbinom(α, n, p)	rbinom$(10, n, p)$
Normal	μ, σ	dnorm(x, μ, σ)	pnorm(x, μ, σ)	qnorm (α, μ, σ)	rnorm$(10, \mu, \sigma)$
Chi-squared	m	dchisq(x, m)	pchisq(x, m)	qchisq(α, m)	rchisq$(10, m)$
T	m	dt(x, m)	pt(x, m)	qt(α, m)	rt$(10, m)$
F	m,n	df(x, m, n)	pf(x, m, n)	qf(α, m, n)	rf$(10, m, n)$

- **Other distributions:**
  Geometric: `dgeom()`
  Negative Binomial: `dnbinom()`
  Poisson: `dpois()`
  Hipergeometric: `dhyper()`
  Exponential: `dexp()`

- **Examples Binomial distribution**

  X Binomial with parameters $n = 8$ i $p = 0.35$
  $P(X = 4)$: `dbinom(4, 8, 0.35)`
  $P(X \leq 4)$: `pbinom(4, 8, 0.35)`
  95% Percentile: `qbinom(0.95, 8, 0.35)`
  Random sample of 25 values of X: `rbinom(25, 8, 0.35)`

- **Examples Normal distribution**

  X Normal of parameters $\mu = 10$ i $\sigma = 3$
  $P(X \leq 15)$: `pnorm(15, 10, 3)`
  $P(X > 20)$: `1-pnorm(20, 10, 3)`
  $P(12 \leq X \leq 20)$: `pnorm(20, 10, 3)- pnorm(12, 10, 3)`
  95% Percentile: `qnorm(0.95, 10, 3)`
  Random sample of 25 values of X: `rnorm(25, 10, 3)`

STATISTICAL TESTS WITH R

y continuous variable *x categorical variable*	**Normality Test: Shapiro-Wilk** H0: Data follow a normal distribution H1: Data do not follow a normal distribution *tapply(<continuous>,<categorical>,function(x) shapiro.test(x))*	
	If Shapiro p-value > alpha **Data follow a normal distribution**	If Shapiro p-value < alpha **Data DO NOT follow a normal distribution**
Test for the mean H0: mean=prespecified value H1: mean≠ prespecified value	T-test t for one sample *t.test(y, mu=value)*	Wilcoxon test for one sample *wilcox.test(y, mu=value)*
Test for the equality of two means H0: mean1=mean2 H1: mean1≠ mean2	T-test for independent samples (previously, you should test for the equality of variances) *t.test(y~x, var.equal=T) # if variances are equal* *t.test(y~x,var.equal=F) # if variances are different*	Wilcoxon test for independent samples (also known as Wilcoxon–Mann–Whitney test) *wilcox.test(y~x)*
Test for the equality of two means with paired samples H0: mean1=mean2 H1: mean1≠ mean2	T-test for paired samples *d<-y1-y2* *t.test(d,mu=0)*	Wilcoxon test for paired samples *wilcox.test(y1,y2,paired=TRUE)*
Test for the equality of more than two means H0: mean1 = mean2 = ... = meank H1: at least one of the means is different	one-factor ANOVA (Requires normality and homoscedasticity) *aov(y~x)* Normality: *shapiro.test(residuals(lm(y~x)))* Post-hoc analysis: *TukeyHSD(aov)* Robust ANOVA (if homoscedasticity is not fulfilled): *oneway.test(y~x)* two-factor ANOVA *aov(y~x1*x2)*	Kruskal-Wallis test *kruskal.test(y~x)*
Test for the equality of two variances H0: variance1= variance2 H1: variance1≠ variance2	F test for the equality of variances *var.test(y~x)*	
Test for the equality of several variances H0: var1 = var2 = ... = vark H1: at least one of the means is different	Homoscedasticity test *install.packages("lmtest")* *library(lmtest)* *bptest(lm(y ~ x),studentize = F)*	

Test for one proportion H0: p= prespecified value p0 H1: p≠ p0	Binomial test for one proportion *binom.test(k,n,p0)*
Test for equality of proportions H0: proportion1= proportion2 H1: proportion1≠ proportion2	Test for the equality of two proportions *prop.test(table(x1,x2)) # x1 i x2 are factors with 2 categories*
Multinomial test $H_0: (\pi_{1,} \dots, \pi_m) = (p_{1,} \dots, p_m)$ $H_1: (\pi_{1,} \dots, \pi_m) \neq (p_{1,} \dots, p_m)$	Multinomial test for proportions *prop.test(x=c(n_1,..., n_m),p=c(p_1, ..., p_m))*
Test for independence of 2 categorical variables H0: X and Y are independent H1: X and Y are related	Chi-squared test for independence of 2 factors *chisq.test(table(x1,x2)) # x1 and x2 are categorical variables*
Test for independence of 2 categorical variables with 2 categories H0: X and Y are independent H1: X and Y are related	Fisher test for independence of 2 factors (2x2 tables) *fisher.test(table(x1,x2)) # x1 and x2 are categorical variables*
Test for odds ratio H0: OR=1 H1: OR ≠ 1	Odds ratio test for 2 factors (2x2 tables) *install.packages("epitools")* *library("epitools")* *oddsratio(table(x1, x2))* *oddsratio(table, rev="c") # reverse columns* *oddsratio(table, rev="both") #reverse both, columns and rows*

Test for independence of two continuous variables H0: X and Y are not correlated H1: X and Y are correlated	Correlation test *cor.test(x,y) # Pearson correlation* *cor.test(x,y, method=c("spearman")) # Spearman correlation*

Outliers test H0: No outliers H1: data contain outliers	Outliers test *library(outliers)* *grubbs.test(x)*

Correction for multiple testing	Benjamini and Hochberg FDR control *p.adjust(p, method = "fdr", n = length(p))*

Regression models with R

Linear regression *Y continuous* *X1, X2 explanatory*	*model<-lm(y~x1+x2, data = data)* *summary(model)* *Check normality of residuals:* *shapiro.test(residuals(model))*
Logistic regression *Y binary* *X1, X2 explanatory*	*model<-glm(y~x1+x2, data = data, family = "binomial")* *summary(model)*
Seleccion of variables in regression *(step-wise regression)*	*step(model)*
Diagnostics in regression: Residuals vs predictions Plots	*plot(predict(model), residuals(model))* *abline(a=0, b=0)*