

R Cheatsheet

Random data:

<code>sample(a:b,n,prob=c(),replace=TRUE)</code>	од а до б, да се генерираат n случајно избрани броеви од нив и опционално дали со повторување и со кои тежини
<code>rbinom(n,size,prob)</code>	биномна распределба, # на појавувања - n пати го извршува експер. од m фрлања каде m е параметарот size. исходи се броевите во интервалот [0,size]
<code>rnbinom(n,size,prob,mu)</code>	негативна биномна распределба, параметарот mu мора да биде специфициран (mu = mean/EX)
<code>rgeom(n,prob)</code>	геометриска распределба, # на појавувања до првиот спротивен исход
<code>rpois(n,lambda)</code>	поасонова распределба, параметар lambda
<code>rnorm(n,mean = a, sd = b)</code>	нормална распределба, пр. за mean = 0, sd = 1
<code>rexp(n,rate)</code>	експоненцијална распределба,
<code>runif(iteracii,min=a,max=b)</code>	рамномерна распределба, во интервалот a,b

Probability:

<code>pbinom(a)</code>	За $P(X \leq x)$, се користат р функциите
<code>pbinom(a,lower.tail=FALSE)</code>	За $P(X > x) == 1 - P(X \leq x)$
<code>dbinom()</code>	За $P(X = x) == P(X \leq x) - P(X \leq x - 1)$ кај дискретни сл. пр. и со d можеме вер. да прес.

Analytics:

<code>summary(data)</code>	сумираме податоци
<code>summary(as.factor(data))/100</code>	
<code>mean(data)</code>	просек (mu, EX)
<code>var(data)</code>	дисперзија (DX)
<code>sd(data)</code>	стандардна девијација (sigma, koren od DX)
<code>median(data)</code>	медијана
<code>range(data)[2]-range(data)[1]</code>	ранг/распон
<code>max(data)-min(data)</code>	ранг/распон (на друг начин само пресметан)
<code>min(data), max(data)</code>	елементот со најмала и најголема вредност од тие во data
<code>quantile(data)</code>	за data ги пресметува квантилите
<code>IQR<-quantile(data)[4]-quantile(data)[2]</code>	IRQ = Q3-Q1
<code>levaGranica<-Q1-1.5*IQR</code>	вредностите надвор од границите се сметаат за екстремни вредности
<code>desnaGranica<-Q3+1.5*IQR</code>	

Misc:

<code>sort(data)</code>	ги сортира податоците во низата
<code>unique(data)</code>	ги трга дупликатите во низата
<code>length(data)</code>	пресметува која и е должината
<code>data<-rep(0,times=n)</code>	создава празна низа со n нули
<code>data<-c(0,0,0,0,0)</code>	рачно иницијализираме низа
<code>data<-vector()</code>	создава празен вектор

Console:

<code>print(val)</code>	или принтаме само променлива, или само стринг
<code>cat("something",val,"something\n")</code>	ова е слично на <code>println()</code> во прог. јазици

Data visualisation:

<code>plot(data,type="o", col="blue")</code>	само ќе ги нацрта реализациите поврзани со бела линија
<code>hist(data,col=topo.colors(n), breaks = n)</code>	хистограм, опционално во боја и бо колку интервали
<code>plot(density(data), col="blue")</code>	слично на хистограмот, т.н. денсити плот
<code>boxplot(data)</code>	боксплот дијаграм, информативен за квартали и сл.
<code>print(stem(data,scale=2))</code>	дијаграм на "лисја и гранки", опционално е за скалата
<code>get(getOption("device"))()</code> <code>par(mfrow = c(1, 3))</code>	овој код е за да нацрта три графици и тоа еден ред со трите. пр. с (2,2) би било два реда со по два
<code>plot(data1,data2)</code> <code>res<-lm(data2~data1)</code> <code>abline(res)</code>	овој код црта график од две низи броеви (со иста должина) и пресметува потоа линеарна регресија и ја претставува нејзината линија во графикот