PACCHETTI	
install.package("...")	Installa pacchetto
library(...)	Usa il pacchetto

MATEMATICA	
Is.numeric()	TRUE se numero
Is.finite(...)	TRUE se è finito
Sqrt(x)	Radice
x%/%y	Divisione intera
x%%y	Resto
Abs(x)	Valore assoluto
Exp(x)	Esponenziale
Log(z,y), log10(x), log2(x)	Logaritmo
Round(... ,2)	Approssima a 2 decimali

DIRECTORY	
getwd()	Cartella in cui sono
setwd("...")	Va nella cartella
~	Directory di default
./	Cartella attuale
../	Cartella precedente

OPERATORI LOGICI	
x & y	And
x | y	Or
xor(x,y)	Xor
!x	not

SEQUENZE	
10:20	10,11,...,20
Seq(10,20)	10:20
Seq(10,20,2)	10,12,14,...,20
Seq(to=10,by=0.5,from=1)	Seq(1,10,0.5)
Rep(1,5)	1 1 1 1 1

VETTORI E OPERAZIONI			
x<-8, 8->x	Assegnazione	z<-c(1,2,...)	Vettore
z[3], z[3:4], z[c(1,4)]=z[1] z[4]		Lettura vettore	
z[-c(1,2,5)]		Mostra il vettore tranne il 1°,2°,5° elemento	
length(z)	Lunghezza vettore	t(z)	Trasposto di z

OPERAZIONI			
x==y	Controllo	z%*%w	Moltiplicazione tra vettori (NON: z*w)

OPERAZIONI LOGICHE CON VETTORI	
which(...)	Indice in cui vale una condizione
min(z) / max(z)	Minimo/massimo di z
which.min(z)	Indice del minimo (prendo il 1°)

OPERAZIONI SU VETTORI PER RAPPRESENTAZIONE DEI DATI				
Mean(DAT $ COL)	Media colonna			
Sd(DAT $ COL)	Deviazione standard colonna			
Median(DAT $ COL)	mediana			
Var(DAT$COL)	Varianza			
IQR(DAT $ COL)	IQR - $\boldsymbol{Q_3 - Q_1}$			
Range(DAT $ COL)	Range colonna			
Quantile(DAT $ COL)	Quantile colonna			
	0%	25%	...	100%
	...	...	...	...
Quantile(DAT $ COL, a)	Quantile: q_a ($a = 0.25$ sono quartili)			
Quantile(DAT $ COL, seq(0,1,0.1))	0%	10%	...	100%
	...	...	...	...

Sys.Date() Fornisce la data di oggi
Sys.Date()-1 Fornisce la data di ieri

GESTIONE ERRORI NA

Nella gestione dei NA il comando "na.omit" può rimuovere molte righe e può essere meglio togliere ad esempio la colonna con più errori, oppure sostituire i NA con la mediana (lasciando inalterata la mediana) oppure con la media (lasciando inalterata la media)

Summary(DAT)	Conta i Nas e dice dove stanno
COMM(DAT $COL, na.rm=T)	Rimuove nel comando tutti gli errori NA. (Mean, median,...)
Is.na(DAT $ COL)	Vettore T, F con T dove si trova un NA
DAT $ COL[is.ma(DAT$COL)]	Vettore: NA NA ...
Na.omit(DAT$COL)	Tabella senza le righe con i NA
D<-na.omit(DAT)	Toglie tutte le righe con NA
x<-x[!is.na(x)]	Rimuove NA da un vettore x

TEST DI STUDENT

t.test(a, mu=x, conf.level=0.95, alternative="less")	Test per vedere se la media di *a* è minore di x con confidenza del 95% ($\alpha = 5\%$) Se $p \ll 0$ allora vale l'hp alternativa: la media è minore di x
t.test(VET1, VET2)	Test per vedere se le medie dei due vettori sono statisticamente uguali. Se l'intervallo di confidenza è lontano da 0 (non contiene 0) $\Rightarrow p \ll 0 \Rightarrow$vale l'HP alternativa: statisticamente la differenza tra le medie è $\neq 0$, è lontana, sono diverse Me lo aspettavo dai boxplot? (Con livello di confidenza più alto l'intervallo è più ampio)
t.test(a,b,alternative="less")	Test oer vedere se la media di *a* è minore della media di *b*. Se $p \ll 0$ allora vale l'HP alternativa: la media di *a* è più piccola di *b*.
t.test(a)	Intervallo di confidenza bilatero

DATA SET

Il dataset è matrice (array*array)⇒per accedere alla colonna n scrivo df[0,n]=df[,n] (alcuni comandi vogliono solo df[n])

TABLE DI VETTORI

Table(*vettore*)	Genera una tabella	a_1	a_2	a_3
		$\#a_1$	$\#a_2$	$\#a_3$

NOTA: table(dati(A$B)) la fa solo della colonna

Prop.table(table(*vettore*))	Genera una tabella con le proporzioni

DATA SET

(PACKAGE)::(DATASET)	Carica il dataset del package
View(DATASET)	Mostra la tabella interattiva del dataset
Str(DATASET)	Tipi di dato e informazioni
Df[,"COL"] <-NULL	Rimuove la colonna "COL"
DAT<- DAT2[,c("COL1","COL2",...)]	Crea database con solo quelle colonne di DAT2
df <- df [- which(df$COL == 0 \| df$COL == 1),]	Rimuove righe dove COL=1 o COL=0
Diff(df$COL)	Colonna di differenza tra un valore e il precedente (omette il primo valore)
Colnames(df[i])	Nome colonna i°
(DATASET)$(COL)	Considero colonna (COL) nel dataset. Così aggiungo anche una colonna al dataset
As.factor(DATASET$COL)	Trasforma i valori della colonna in variabili categoriche (è da **assegnare**)
As.Date(df$COL)	Trasforma in data (è da **assegnare**)
Weekdays((data))	Trasforma in giorno della settimana una data
Levels(DATASET$COL)<-c('A','B',...)	Sovrascrive nomi dei livelli (solo se sono **FACTOR**), ma i valori sono inalterati
Tail(df, 5) Head(df,5)	Mostra gli ultimi 5, mostra i primi 5

CARICARE DA FONTI ESTERNE

Read.csv	Carico il dataset con , come separatori (HEADER=TRUE)
Read.csv2	Carico il dataset con ; come separatori
Read.table('.../file.txt', header=T, sep=':', stringAsFactors=T)	HEADER: se prima riga è nomi di colonna | SEP: il separatore dei dati StringAsFactors: se le stringhe sono categoriche

LAVORARE SU PIU DATASET

Cbind(dfCOL, dfCOL2,...)	Incolla più colonne. NON è un oggetto DATASET →as.data.frame(cbind(...))
Rbind(df[1,],df[2,])	Incolla la 1° e la 2° riga
Merge(x,y,by="COL") Merge(x,y,by.x="COL1",by.y="COL2")	Unisce dataset sulla colonna COL Unisce dataset su x colla colonna COL1, su y colla colonna COL2
Aggregate(df, by=list(df$gen), FUN=mean)	Aggrega dati e fa tabella con media di ogni colonna in base a gender. Per togliere media su gender faccio df[-1]

	GENDER	AGE
F	NA	37.4
M	NA	37.8

OPERAZIONI SUL DATASET

Lapply(DAT, COM)	Applica il comando COM a tutte le colonne del dataset DAT
Summary(DAT)	Applica a tutte le colonne del dataset (riconoscendo quelle numeriche): MIN, Q_{25}, MEDIAN, MEDIA, Q_{75}, MAX. Se una colonna è non numerica fa il conto delle classi
Lapply(DAT[-c(1,3)],COM)	Esegue il comando su tutto il database tranne la 1° e 3° colonna

REGRESSIONE LINEARE

Per migliorare uso: plot(lm(...)) e nel 4) vedo se i punti estremi influiscono tanto (oltre rosso), provo a toglierli

Lm(COL1~COL2+..., data=df)	Cerco se COL1 dipende da COL2+... [COL2 è il predittore]
Summary(lm(...))	- Residui: distanzz tra quanto predico e i dati. Se troppo grandi non c'è relazione. Cerco medio=0 - Coefficients: β_0, β_1. Il p-value dice se il coefficiente è statisticamente $\neq 0$. Voglio p-value ~ 0 - R^2: più è vicino a 1 meglio è, significa che il modello statistico lavora bene - (ultimo p-value): mostr se $(\beta_1, \beta_2) \neq (0,0)$

SUMMARY GRAFICO

Plot(bmi ~ glu, data=Pima.tr) Abline(lm(...)$coefficients, col='red')	Punti sperimentali e retta di regressione. Devono essere simili
Qqnorm(lm(...)$residuals) Qqline(lm(...)$residuals)	Distribuzione in quantili dei residui, confronto tra quantili teorici e pratici (voglio simili). Punti⇒ distrib in quantili dei residui trovati Retta⇒distrib quantili di una normale
Hist(lm(...)$residuals) Plot(lm(...)$residuals)	Visualizzazione residui. L'istogramma dovrebbe essere forma di una normale. Se residui non casuali (seguono pattern)⇒non indipendenti⇒errore in modello o ignoro variabile
Plot(ln(...))	Stampa 4 plot: 1) Residual vs Fitted: mostra se i residui hanno qualche andamento. Voglio la linea rossa costante 0 2) Qqnorm + qqlines 3) ... 4) Mostra gli outlier (male se oltre le linee rosse)
Plot(D$C, lm(C~...)$fitted_values) Abline(0,1)	Confronto i valori che volevo predire e lm(C~...). La x corrisponde al dato vero, la y al dato predetto. Idealmente devono stare lungo la bisettrice (abline(0,1))

PROGRAMMAZIONE

Print(i) / print('testo')	
Paste(...,...)	Stampa (assegna) i due oggetti Negli if / for devo usare il print per farlo stampare
If(i==1){...}else{...} NB: ii<-c(0,1,0,1) if(ii=1) →F T F T esegue solo il 1°	
ii<-c(0,1,0,1) Ifelse(ii==1, 1,2)=(2,1,2,1)	
For(i in seq(1,10,1))	
While(i<10){...}	
Repeat{...}	ripete all'infinito, serve break

FUNZIONI

My_fun<-function(a,b){ if(a==b) c<-a return(c)}

GRAFICI

Distribuzione di … ⇔ istogramma / boxplot

Par(mfrow=c(1,2), mar=c(2,1,1,1))	Mostra due grafici sulla stessa riga (1 - righe, 2 - colonne), sistema i grafici

ISTOGRAMMA

Hist(DAT$COL, freq=F, breaks=10, xlab='temp', ylab='Fr', main='title', ylim=c(0,20),xlim=c(35,39),col='pink')

Breaks: quanti rettangolo (se a R non piace fa come vuole)
Xlab, ylab: label agli assi
Ylim, xlim: scala degli assi
Freq=F: impongo frequenza relativa invece che la frequenza (da usare quando ad esempio voglio fare due istogrammi tra M, F e non sono in uguale quantità)
Col: colore

Hist(DAT$COL[DAT$COL2=='A'],...)	Istogramma per quelli con Col2='A'

BOXPLOT

Valgono gli stessi flag di hist per i label, titolo, ylim,…

Boxplot(DAT$COL)	Boxplot della colonna COL
Boxplot('C', data=DAT)=Boxplot(D$C)	
Boxplot(C~V,data=D)	V - variabile categorica Fa un boxplot per ogni variabile categorica V ES: Boxplot(Temp~Gender, data=D) fa 2 boxplot, uno per M e uno per F, in cui la mediana e gli outlier sono calcolati singolarmente per M e per F

PLOT

Xx<-seq(0,10,0.1) yy<-... plot(xx,yy) plot(xx,yy,'l')	Fa il grafico con punti, nel secondo con linea

GENERICHE

Points(0,1,col='red')	Aggiunge un punto rosso al plot/istogramma/boxplot già fatto
Lines(xx,yy)	Aggiunge linea al plot/istogramma/boxplot già fatto
Abline(a,b,h=...,v=...)	Aggiunge linea dritta al plot/istogramma/boxplot già fatto. a: intercetto asse y b: coeff ang **oppure** h=x :linea orizzontale in x **oppure** v=x : linea verticale in x

PROBABILITA'

Distribuzioni: norm, pois, gamma, exp... Le funzioni che seguono valgono con tutte ma cambiano i parametri da inserire (vedi help)

Dnorm(…)	Ritorna il valore che assume la distribuzione in base ai parametri
Pnorm(…)	Ritorna il valore di funzione di distribuzione ($\mathbb{P}(X \leq t)$)
Qnorm(0.85,…)	Trova l'85° percentile, cioè $\mathbb{P}\big(X < qnorm(0.85, \dots)\big) = 0.85$
Rnorm(…)	Genera numeri random che seguono la distribuzione (bello da vedere con istogramma)

ESEMPIO VETTORI E INDICI

```
zz <-  c(19 , 30, 7, 3, 21, 5, 3)
#Trovo elementi <10
zz<10 #Vettore F F T T F T T
zz[zz<10] #7 3 5 3
#Trovo gli elementi che sono min(zz)
min(zz)
which.min(zz) #indice 1° el uguale al min, non tutti!
zz == min(zz) #Controllo: F F F T F F T
#salvo tutti gli indici del minimo
indici_minimo <- which(zz == min(zz))
indici_minimo #4 7
#trova tutti i valori del min
zz[indici_minimo] #3 3
zz[zz == min(zz)] #3 3
```

STAMPA NOME COLONNA SE NUMERICA

```
for(i in seq(1,8,1)){
 if(is.numeric(my_data[,i])){
  print(paste(colnames(my_data[i]), 'è numerica'))
  paste('a',2)
 }
}
```

CREO CLASSI DI OBESITA

```
dati_pima<- MASS::Pima.te

#BMI
#[16 - 18.5)  SOTTOPESO  -> 1
#[18.5 - 25)  NORMOPESO  -> 2
#[25 - 30)   SOVRAPPESO -> 3
#[30 - 35)  OBESITA' CLASSE I -> 4
#[35- )  OBESITA GRAVE  -> 5

dati_pima$classif_peso<-0
dati_pima$classif_peso[dati_pima$bmi <18.5] <-1
dati_pima$classif_peso[dati_pima$bmi >=18.5 & dati_pima$bmi <25]
<- 2
dati_pima$classif_peso[dati_pima$bmi >=25 & dati_pima$bmi <30] <-
3
dati_pima$classif_peso[dati_pima$bmi >=30 & dati_pima$bmi <35] <-
4
dati_pima$classif_peso[dati_pima$bmi >=35] <-5

dati_pima$classif_peso <- as.factor(dati_pima$classif_peso)
levels(dati_pima$classif_peso)<-c('Normopeso', 'sovrappeso','obesità',
'obesità grave')
```

DATASET DI BOLZANO + MEDIA DI MARZO

```
d_Bz <- d [d$regione == 'P.A. Bolzano',]
m_MARZO <- mean(d_Bz$nuovi_positivi[d_Bz$data >= '2021-03-01' & d_Bz$data < '2021-04-01'])
```

RINOMINO COLONNE

```
colnames(agg_DF) <- c("Nuovo Nome", ...)
```

CONTO QUANTE RIGHE DA RIMUOVERE + RIMUOVO RIGHE

```
#conto quante colonne hanno ...==0, ...==1
dim(df_bad1[df_bad1$Intelligence == 0 | df_bad1$Intelligence == 1,  ])
#rimuovo quelle colonne
df_bad2 <- df_bad1[ - which(df_bad1$Intelligence == 0 | df_bad1$Intelligence == 1),  ]
```