

# Basic statistics

## 1 Prepare data sets for GUI session

Open the command line

```
my.normal <- data.frame(x=seq(from=-3, to=3, by=0.1))
my.normal
my.draws <- data.frame()
my.draws
library(MASS)
?shrimp
shrimp <- shrimp
shrimp
?cabbages
cabbages <- cabbages
cabbages
?mammals
mammals <- mammals
mammals
```

Close the command line

## 2 GUI probability functions and random sampling

### Probability functions

Data - DistributionFunctions

Data set [my.normal] - Source column [x] - Result type [Density]

Close the data editor

Data - DistributionFunctions

Data set [my.normal] - Source column [x] - Result type [Probability]

Close the data editor

### Random number generator

Data - Random numbers

Data set [my.draws] - Target column [normal.40.5] - Sample size [100] - Mean [40] - Std. dev [5]

Close the data editor

Data - Random numbers

Data set [my.draws] - Target column [uniform.neg3.3] - Sample size [100] - Distribution [uniform] -

Minimum [-3] - Maximum [3]

Close the data editor

### Random sampling from empirical data

Data - Random sample

Data set [mammals\$brain] - Sample size [100] - Replacement [v] - Save in [my.draws\$brain]

Close the data editor

Data - Restructure - Stack  
From data set [my.draws.brains] - To data set [my.draws] - Stack column [brains] - Create group column [untick]  
Close the data editor

Data - Random sample  
Data set [my.normal\$x] - Sample size [100] - Replacement [v] - Save in [my.draws\$x]  
Close the data editor

Data - Restructure - Stack  
From data set [my.draws.x] - To data set [my.draws] - Stack column [x] - Create group column [untick]  
Close the data editor

### 3 GUI descriptive statistics

#### Mean, median, variance

Switch to object explorer and double click the shrimp  
Close the data editor

Statistics - Data summaries - Summary statistics  
Data set [shrimp] - Data [untick all except mean, median, variance]

#### Correlation coefficient

Switch to object explorer and double click the mammals  
Close the data editor

Statistics - Data summaries - Correlations  
Data set [mammals] - Variables [body and brain]

### 4 GUI significance tests

#### Student's *t* test

Statistics - Compare samples - One sample - T test  
Data set [my.draws] - Variable [norm.40.5] - Mean under null hypothesis [39]

Statistics - Compare samples - Two samples - T test  
Data set [blank] - Variable 1 [my.draws\$brain] - Variable 2 [mammals\$brain]

#### F test (one way anova)

Switch to object explorer and double click the cabbages  
Close the data editor

Graph - 2D plot - Box plot  
Data set [cabbages] - X column [Date] - Y column [VitC]

Statistics - Compare samples - K samples - One way anova  
Data set [cabbages] - Variable [VitC] - Grouping variable [Date]

## 5 GUI linear models

### Anova

Graph - 2D plot - Box plot

Data set [cabbages] - X column [Cult] - Y column [VitC]

Statistics - Data summaries - Crosstabulations

Data set [cabbages] - Variables [Cult and Date] - Options [untick all]

Statistics - Anova - Fixed effects

Data set [cabbages] - Create formula - [VitC, response] - [Cult and Date, main and interaction]

### Linear regression

Graph - 2D plot - Fit linear least squares

Data set [cabbages] - X columns [HeadWt] - Y columns [VitC]

Statistics - Regression - Linear

Data set [cabbages] - Formula [VitC~HeadWt]

## 6 Prepare data sets for command line session

```
ls()
rm(list=ls())
#R: data(shrimp, cabbages, mammals)
#S: shrimp <- shrimp
#S: cabbages <- cabbages
#S: mammals <- mammals
```

## 7 Probability functions and random sampling

### Probability functions

```
x <- seq(-3, 3, 0.1)
plot(x, dnorm(x))
plot(x, pnorm(x))
my.normal <- data.frame(x=x, pdf=dnorm(x), cdf=pnorm(x))
rm(x)
ls()
```

### Random number generator

```
y <- rnorm(100, m=40, s=5)
z <- runif(100, -3, 3)
hist(y)
hist(z)
my.draws <- data.frame(normal.40.5=y, uniform.neg3.3=z)
rm(y, z)
ls()
```

### Random sampling from empirical data

```
my.draws$brain <- sample(mammals$brain, 100, replace=T)
my.draws$x <- sample(my.normal$x, 100, replace=T)
my.draws
```

## 8 Descriptive statistics

### Mean, median, variance

```
mean(shrimp)
median(shrimp)
var(shrimp)
```

### Correlation coefficient

```
cor(mammals$body, mammals$brain)
cor(mammals)
```

## 9 Significance tests

### Student's *t* test

```
t.test(my.draws$normal.40.5, mu=39)
```

### F test (one way anova)

```
boxplot(split(cabbages$VitC, cabbages$Date))
aov(VitC~Date, data=cabbages)
summary(aov(VitC~Date, data=cabbages))
```

## 10 Linear models

### Anova

```
boxplot(split(cabbages$VitC, cabbages$Cult))
table(cabbages$Cult, cabbages$Date)
aov(VitC~Cult*Date, data=cabbages)
summary(aov(VitC~Cult*Date, data=cabbages))
interaction.plot(cabbages$Cult, cabbages$Date, cabbages$VitC)
```

### Linear regression

```
plot(cabbages$HeadWt, cabbages$VitC)
abline(lm(VitC~HeadWt, data=cabbages))
summary(lm(VitC~HeadWt, data=cabbages))
```

## 11 Data objects in S

Data are usually stored either in a vector or data frame

Data elements can be numeric (like 9), character (like "nine"), or logical (TRUE/FALSE)

```
shrimp[3]           # extract 3rd element from shrimp vector
mammals$brain       # extract "brain" column from mammals data frame
names(mammals)      # show column names of mammals data frame
```

## 12 Programming

You can paste the following code into the command line:

```

cv <- function(x, return.list=FALSE)
#####
###
### Function: cv
###
### Purpose: Calculate coefficient of variation (CV)
###
### Args:      x is a vector of numbers
###           return.list is whether a list should be returned
###
### Returns:  CV as a number if return.list is FALSE, or
###           a list of mean, stdev, and cv if return.list is TRUE
###
#####
{
  m <- mean(x)
  s <- sqrt(var(x)) # or sd(x) in R
  cv <- s/m

  if(return.list==TRUE)
    output <- list(mean=m, stdev=s, cv=cv)
  else
    output <- cv

  return(output)
}

```

## 13 New functions

Data manipulation	data.frame seq split names table list
Import/export	library #R: data
Basic statistics	dnorm pnorm rnorm runif sample mean median var #R: sd cor t.test
Graphics	hist boxplot interaction.plot
Modelling	aov
Programming	function return