Bentley University GR521 in R

Content extracted from the How to Data Website

This PDF generated on 30 November 2021

Contents

GR521 is a graduate Managerial Statistics course at Bentley University. The description from the course catalog can be found here.

Mathematical topics include random variables, discrete and continuous probability distributions, confidence intervals, hypothesis testing, single-variable linear models, and optionally advanced topics such as data mining, time permitting.

Basics

- How to do basic mathematical computations
- How to compute summary statistics

Random variables and probability distributions

- How to generate random values from a distribution
- How to compute probabilities from a distribution
- How to plot continuous probability distributions
- How to plot discrete probability distributions

Confidence intervals and hypothesis testing

- How to compute a confidence interval for a population mean
- How to do a two-sided hypothesis test for a sample mean
- How to do a two-sided hypothesis test for two sample means

Linear modeling

- How to fit a linear model to two columns of data
- How to compute R-squared for a simple linear model

Other end-of-semester topics, time permitting

- How to do a one-way analysis of variance (ANOVA)
- How to perform a chi-squared test on a contingency table

Content last modified on 30 November 2021.

How to do basic mathematical computations

Description

How do we write the most common mathematical operations in a given piece of software? For example, how do we write multiplication, or exponentiation, or logarithms, in Python vs. R vs. Excel, and so on?

Solution in R

For those expressions that need the Python math package, use the code `import math` beforehand to ensure that package is loaded. Alternatively, you can write `from math import *` and thus drop the `math` prefixes in the table below.

Mathematical notation	R code
$x+y$	`x+y`
$x-y$	`x-y`
xy	`x*y`
$\frac{x}{y}$	`x/y`
x^y	`x^y`
$\lvert x\rvert$	`abs(x)`
$\ln x$	`log(x)`
$\log_a b$	`log(b,a)`
e^x	`exp(x)`
π	`pi`
$\sin x$	`sin(x)`
$\sin^{-1} x$	`asin(x)`
$\sqrt{x}$	`sqrt(x)`

Other trigonometric functions are also available besides just `sin`, including `cos`, `tan`, etc.

R naturally applies these functions across vectors. For example, you can square all the entries in a vector as in the example below.

```
example.vector <- c( -3, 2, 0.5, -1, 10, 9.2, -3.3 )
example.vector ^ 2
```

```
[1]   9.00   4.00   0.25   1.00 100.00  84.64  10.89
```

Content last modified on 23 November 2021.

See a problem? Tell us or edit the source.

How to compute summary statistics

Description

The phrase "summary statistics" usually refers to a common set of simple computations that can be done about any dataset, including mean, median, variance, and some of the others shown below.

Related tasks:

- How to summarize a column (on website)
- How to summarize and compare data by groups (on website)

Solution in R

We first load a famous dataset, Fisher's irises, just to have some example data to use in the code that follows. (See how to quickly load some sample data (on website).)

```
library(datasets)
data(iris)
```

How big is the dataset? The output shows number of rows then number of columns.

```
dim(iris)  # Short for "dimensions."
```

```
[1] 150   5
```

What are the columns and their data types? Can I see a sample of each column?

```
str(iris)  # Short for "structure."
```

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

What do the first few rows look like?

```
head(iris) # Gives 5 rows by default.  You can do head(iris,10), etc.
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1 5.1          3.5         1.4          0.2         setosa
2 4.9          3.0         1.4          0.2         setosa
3 4.7          3.2         1.3          0.2         setosa
4 4.6          3.1         1.5          0.2         setosa
5 5.0          3.6         1.4          0.2         setosa
6 5.4          3.9         1.7          0.4         setosa
```

The easiest way to get summary statistics for an R `data.frame` is with the `summary` function.

```r
summary(iris)
```

```
  Sepal.Length    Sepal.Width     Petal.Length    Petal.Width   
 Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100  
 1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300  
 Median :5.800   Median :3.000   Median :4.350   Median :1.300  
 Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199  
 3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800  
 Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500  
       Species  
 setosa    :50  
 versicolor:50  
 virginica :50  
```

The columns from the original dataset are the column headings in the summary output, and the statistics computed for each are listed below those headings.

We can also compute these statistics (and others) one at a time for any given set of data points. Here, we let `xs` be one column from the above `data.frame` but you could use any vector or list.

```r
xs <- iris$Sepal.Length

mean( xs )           # mean, or average, or center of mass
median( xs )         # 50th percentile
quantile( xs, 0.25 ) # compute any percentile, such as the 25th
var( xs )            # variance
sd( xs )             # standard deviation, the square root of the variance
sort( xs )           # data in increasing order
sum( xs )            # sum, or total
```

Content last modified on 26 July 2021.

See a problem? Tell us or edit the source.

How to generate random values from a distribution

Description

There are many famous continuous probability distributions, such as the normal and exponential distributions. How can we get access to them in software, to generate random values from a chosen distribution?

Related tasks:

- How to compute probabilities from a distribution
- How to plot continuous probability distributions
- How to plot discrete probability distributions

Solution in R

Because R is designed for use in statistics, it comes with many probability distributions built in. A list of them is online here.

Regardless of whether the distribution is discrete or continuous, prefix the name of the distribution with `r`, which stands for "random values." Here are two examples.

Using a **normal distribution:**

```
# 20 random values from the normal distribution with μ=10 and σ=5
rnorm( 20, mean=10, sd=5 )
```

```
 [1] 12.834868 10.263235  7.319098  4.505638 13.341125  4.707107  1.326426
 [8] 14.601987  5.471431  9.473594 19.464538  6.449898 12.961158  8.912349
[15] 20.832911 13.854628 10.340365 13.120180 11.649607  6.076106
```

Using a **uniform distribution:**

```
# 20 random values from the uniform distribution on the interval [50,60]
runif( 20, min=50, max=60 )
```

```
 [1] 58.53269 57.80936 52.86535 56.72409 57.54950 59.10527 54.18573 54.45018
 [9] 50.83745 52.57525 53.22530 51.38194 58.38474 56.65091 56.13401 59.22261
[17] 55.29325 55.06911 53.83386 51.99634
```

Content last modified on 27 May 2021.

See a problem? Tell us or edit the source.

How to compute probabilities from a distribution

Description

There are many famous continuous probability distributions, such as the normal and exponential distributions. How can we get access to them in software, to compute the probability of a value/values occurring?

Related tasks:

- How to generate random values from a distribution
- How to plot continuous probability distributions
- How to plot discrete probability distributions

Solution in R

Because R is designed for use in statistics, it comes with many probability distributions built in. A list of them is online here.

To compute a probability from a **discrete** distribution, prefix the name of the distribution with `d` (for "density") and call it as a function on the value whose probability you want to know, plus any parameters the distrubtion needs.

```
# For a binomial random variable with 10 trials
# and probability 0.5 of success on each trial,
# what is the probability of exactly 3 successes?
dbinom( 3, size=10, prob=0.5 )
```

```
[1] 0.1171875
```

If you change the prefix to `p`, then R will compute the probability *up to* the parameter you specify, as in the following example.

```
# For a binomial random variable with 10 trials
# and probability 0.5 of success on each trial,
# what is the probability of up to (and including) 3 successes?
pbinom( 3, size=10, prob=0.5 )
```

```
[1] 0.171875
```

To compute a probability from a **continuous** distribution, prefix the name with `d`, just as in the example above. But you can compute only the probability that a random value will fall in an interval $[a, b]$, not the probability that it will equal a specific value.

```
# For a normal random variable with mean μ=10 and standard deviation σ=5,
# what is the probability of the value lying in the interval [12,13]?
pnorm( 13, mean=10, sd=5 ) - pnorm( 12, mean=10, sd=5 )
```

```
[1] 0.07032514
```

Consequently, we can also compute:

```r
pnorm( 13, mean=10, sd=5 )     # the probability of a value < 13
1 - pnorm( 13, mean=10, sd=5 ) # the probability of a value > 13
```

```
[1] 0.7257469



[1] 0.2742531
```

Content last modified on 14 September 2021.

See a problem? Tell us or edit the source.

How to plot continuous probability distributions

Description

There are many famous continuous probability distributions, such as the normal and exponential distributions. How can we get access to them in software, to plot the distribution as a curve?

Related tasks:

- How to generate random values from a distribution
- How to compute probabilities from a distribution
- How to plot discrete probability distributions

Solution in R

Because R is designed for use in statistics, it comes with many probability distributions built in. A list of them is online here.

The challenge with plotting a random variable is knowing the appropriate sample space, because some random variables have sample spaces of infinite width, which cannot be plotted.

But we can just ask R to show us the central 99.98% of a continuous distribution, which is almost always indistinguishable to the human eye from the entire distribution.

We will use a normal distribution with $\mu = 10$ and $\sigma = 5$, but if you wanted to use a different distribution, you could replace `qnorm` and `dnorm` with, for example, `qchisq` and `dchisq` (for the χ^2 distribution), adjusting the named parameters as appropriate. (For a list of supported distributions, see the link above.)

We style the plot below so that it is clear the sample space is continuous.

```
xmin <- qnorm( 0.0001, mean=10, sd=5 )  # compute min x as the 0.0001 quantile
xmax <- qnorm( 0.9999, mean=10, sd=5 )  # compute max x as the 0.9999 quantile
xs <- seq( xmin, xmax, length.out=100 ) # create 100 values in that range
ys <- dnorm( xs, mean=10, sd=5 )        # compute the shape of the distribution
plot( xs, ys, type='l' )                # plot that shape as a smooth line
```

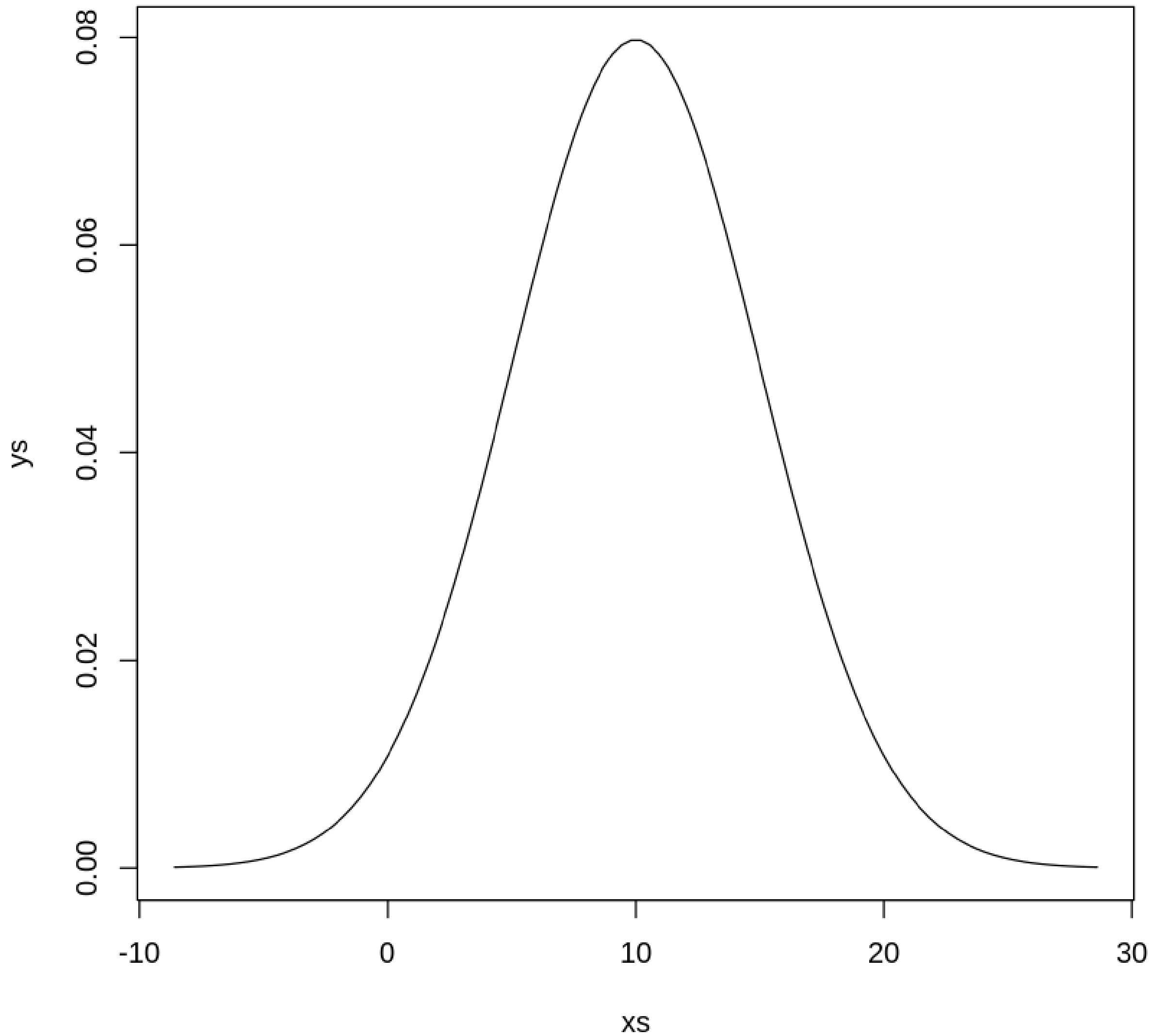

Figure 1: png

Content last modified on 28 May 2021.

See a problem? Tell us or edit the source.

How to plot discrete probability distributions

Description

There are many famous discrete probability distributions, such as the binomial and geometric distributions. How can we get access to them in software, to plot the distribution as a series of points?

Related tasks:

- How to generate random values from a distribution
- How to compute probabilities from a distribution
- How to plot continuous probability distributions

Solution in R

Because R is designed for use in statistics, it comes with many probability distributions built in. A list of them is online here.

The challenge with plotting a random variable is knowing the appropriate sample space, because some random variables have sample spaces of infinite width, which cannot be plotted.

The example below uses a geometric distribution (with $p = 0.5$), whose sample space is $\{0, 1, 2, 3, \ldots\}$. We specify that we just want to use x values in the set $\{0, 1, 2, \ldots, 10\}$. (In some software, the geometric distribution's sample space begins at 1, but not in R.)

If you wanted to use a different distribution, you could replace `dgeom` with, for example, `dbinom`, adjusting the named parameters as appropriate.

We style the plot below so that it is clear the sample space is discrete.

```
xs = 0:8                     # choose the sample space (here, it's 0,1,2,...,10)
ys = dgeom( xs, prob=0.5 )   # compute the shape of the distribution
plot( xs, ys, type='p',      # plot circles...
      xlab='sample space', ylab='probability' )
segments( xs, 0, xs, ys )    # ...and lines
```

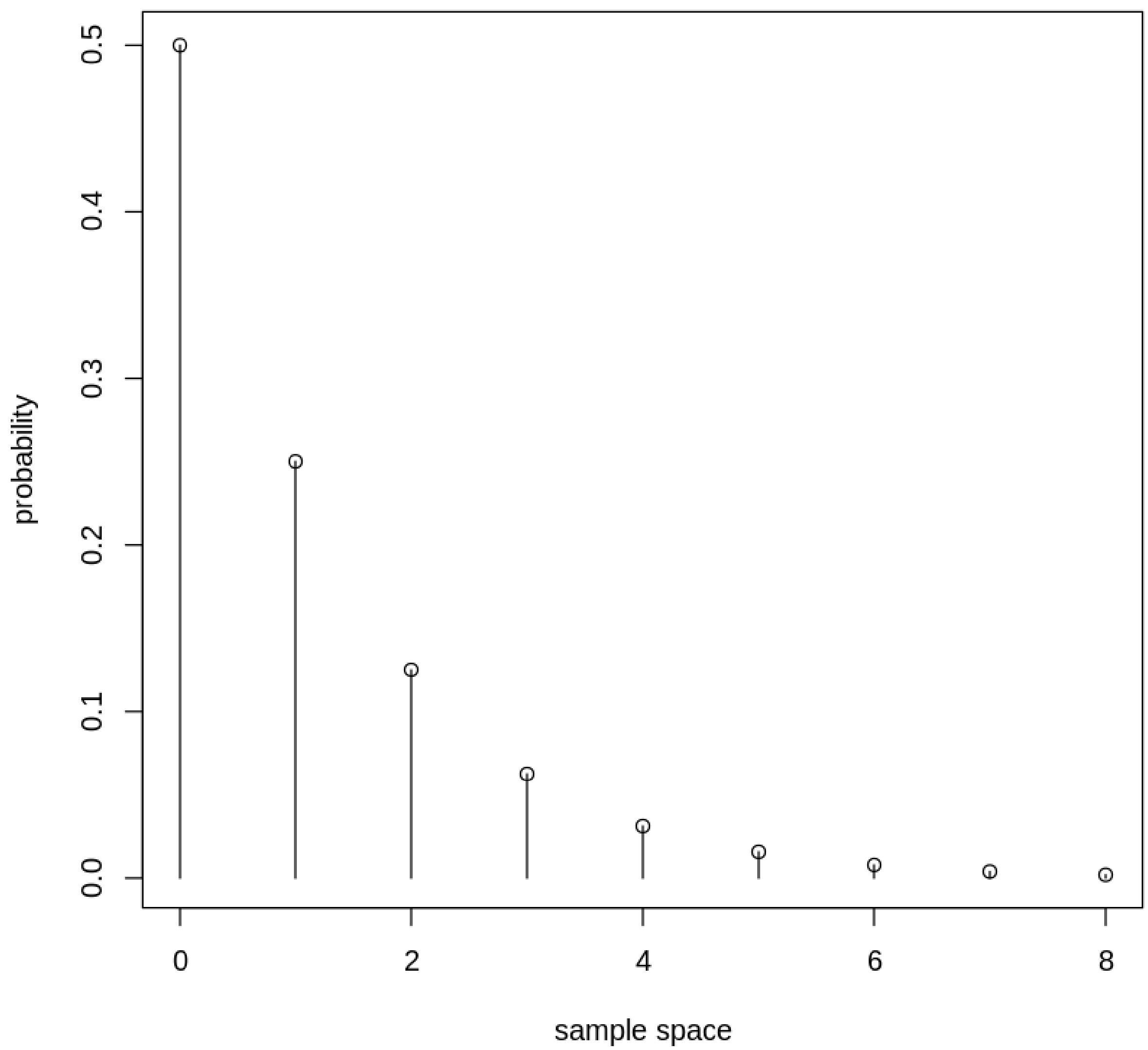

Figure 2: png

Content last modified on 28 May 2021.

See a problem? Tell us or edit the source.

How to compute a confidence interval for a population mean

Description

If we have a set of data that seems normally distributed, how can we compute a confidence interval for the mean? Assume we have some confidence level already chosen, such as $\alpha = 0.05$.

We will use the t-distribution because we have not assumed that we know the population standard deviation, and we have not assumed anything about our sample size. If you know the population standard deviation or have a large sample size (typically at least 30), then you can use z-scores instead; see how to compute a confidence interval for a population mean using z-scores (on website).

Related tasks:

- How to compute a confidence interval for a population mean using z-scores (on website)
- How to do a two-sided hypothesis test for a sample mean
- How to do a two-sided hypothesis test for two sample means
- How to compute a confidence interval for a mean difference (matched pairs) (on website)
- How to compute a confidence interval for a regression coefficient (on website)
- How to compute a confidence interval for a single population variance (on website)
- How to compute a confidence interval for the difference between two means when both population variances are known (on website)
- How to compute a confidence interval for the difference between two means when population variances are unknown (on website)
- How to compute a confidence interval for the difference between two proportions (on website)
- How to compute a confidence interval for the expected value of a response variable (on website)
- How to compute a confidence interval for the population proportion (on website)
- How to compute a confidence interval for the ratio of two population variances (on website)

Solution in R

When applying this technique, you would have a series of data values for which you needed to compute a confidence interval for the mean. But in order to provide code that runs independently, we create some fake data below. When using this code, replace our fake data with your real data.

```r
alpha <- 0.05       # replace with your chosen alpha (here, a 95% confidence level)
data <- c( 435,542,435,4,54,43,5,43,543,5,432,43,36,7,876,65,5 ) # fake

# If you need the two values stored in variables for later use, do:
answer <- t.test( data, conf.level=1-alpha )
lower_bound <- answer$conf.int[1]
upper_bound <- answer$conf.int[2]

# If you just need to see the results in a report, do this alone:
t.test( data, conf.level=1-alpha )
```

```
	One Sample t-test

data:  data
t = 3.1853, df = 16, p-value = 0.005753
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
  70.29848 350.05446
sample estimates:
mean of x
 210.1765
```

Note: The solution above assumes that the population is normally distributed, which is a common assumption in introductory statistics courses, but we have not verified that assumption here.

Content last modified on 30 November 2021.

See a problem? Tell us or edit the source.

How to do a two-sided hypothesis test for a sample mean

Description

Say we have a population whose mean μ is known. We take a sample $x_1, \ldots, x_n$ and compute its mean, $\bar{x}$. We then ask whether this sample is significantly different from the population at large, that is, is $\mu = \bar{x}$?

Related tasks:

- How to compute a confidence interval for a population mean
- How to do a two-sided hypothesis test for two sample means
- How to do a one-sided hypothesis test for two sample means (on website)
- How to do a hypothesis test for a mean difference (matched pairs) (on website)
- How to do a hypothesis test for a population proportion (on website)

Solution in R

This is a two-sided test with the null hypothesis $H_0 : \mu = \bar{x}$. We choose a value $0 \leq \alpha \leq 1$ as the probability of a Type I error (false positive, finding we should reject H_0 when it's actually true).

```
# Replace these first three lines with the values from your situation.
alpha <- 0.05
pop.mean <- 10
sample <- c( 9, 12, 14, 8, 13 )

# Run a one-sample t-test and print out alpha, the p value,
# and whether the comparison says to reject the null hypothesis.
t.test( sample, mu=pop.mean, conf.level=1-alpha )
```

```
    One Sample t-test

data:  sample
t = 1.0366, df = 4, p-value = 0.3585
alternative hypothesis: true mean is not equal to 10
95 percent confidence interval:
  7.986032 14.413968
sample estimates:
mean of x
     11.2
```

Although we can deduce the answer to our question from the above output, by comparing the p value with α manually, we can also ask R to do it.

```
# Is there enough evidence to reject the null hypothesis?
result <- t.test( sample, mu=pop.mean, conf.level=1-alpha )
result$p.value < alpha
```

```
[1] FALSE
```

In this case, the sample does not give us enough information to reject the null hypothesis. We would continue to assume that the sample is like the population, $\mu = \bar{x}$.

Content last modified on 05 October 2021.

See a problem? Tell us or edit the source.

How to do a two-sided hypothesis test for two sample means

Description

If we have two samples, $x_1, \ldots, x_n$ and $x'_1, \ldots, x'_m$, and we compute the mean of each one, we might want to ask whether the two means seem approximately equal. Or more precisely, is their difference statistically significant at a given level?

Related tasks:

- How to compute a confidence interval for a sample mean
- How to do a two-sided hypothesis test for a sample mean
- How to do a one-way analysis of variance (ANOVA)
- How to do a one-sided hypothesis test for two sample means (on website)
- How to do a hypothesis test for a mean difference (matched pairs) (on website)
- How to do a hypothesis test for a population proportion (on website)

Solution in R

If we call the mean of the first sample $\bar{x}_1$ and the mean of the second sample $\bar{x}_2$, then this is a two-sided test with the null hypothesis $H_0 : \bar{x}_1 = \bar{x}_2$. We choose a value $0 \le \alpha \le 1$ as the probability of a Type I error (false positive, finding we should reject H_0 when it's actually true).

```r
# Replace these first three lines with the values from your situation.
alpha <- 0.10
sample1 <- c( 6, 9, 7, 10, 10, 9 )
sample2 <- c( 12, 14, 10, 17, 9 )

# Run a one-sample t-test and print out alpha, the p value,
# and whether the comparison says to reject the null hypothesis.
t.test( sample1, sample2, conf.level=1-alpha )
```

```
    Welch Two Sample t-test

data:  sample1 and sample2
t = -2.4617, df = 5.7201, p-value = 0.05097
alternative hypothesis: true difference in means is not equal to 0
90 percent confidence interval:
 -7.0057683 -0.7942317
sample estimates:
mean of x mean of y 
      8.5      12.4 
```

Although we can deduce the answer to our question from the above output, by comparing the p value with α manually, we can also ask R to do it.

```r
# Is there enough evidence to reject the null hypothesis?
result <- t.test( sample1, sample2, conf.level=1-alpha )
result$p.value < alpha
```

```
[1] TRUE
```

In this case, the samples give us enough evidence to reject the null hypothesis at the $\alpha = 0.10$ level. The data suggest that $\bar{x}_1 \neq \bar{x}_2$.

Here we did not assume that the two samples had equal variance. If in your case they do, you can pass the parameter `var.equal=TRUE` to `t.test`.

Content last modified on 28 May 2021.

See a problem? Tell us or edit the source.

How to fit a linear model to two columns of data

Description

Let's say we have two columns of data, one for a single independent variable x and the other for a single dependent variable y. How can I find the best fit linear model that predicts y based on x?

In other words, what are the model coefficients β_0 and β_1 that give me the best linear model $\hat{y} = \beta_0 + \beta_1 x$ based on my data?

Related tasks:

- How to compute R-squared for a simple linear model
- How to fit a multiple linear regression model (on website)
- How to predict the response variable in a linear model (on website)

Solution in R

This solution uses fake example data. When using this code, replace our fake data with your real data.

```
# Here is the fake data you should replace with your real data.
xs <- c( 393, 453, 553, 679, 729, 748, 817 )
ys <- c(  24,  25,  27,  36,  55,  68,  84 )

# If you need the model coefficients stored in variables for later use, do:
model <- lm( ys ~ xs )
beta0 = model$coefficients[1]
beta1 = model$coefficients[2]

# If you just need to see the coefficients, do this alone:
lm( ys ~ xs )
```

```
Call:
lm(formula = ys ~ xs)

Coefficients:
(Intercept)           xs
   -37.3214       0.1327
```

The linear model in this example is approximately $y = 0.133x - 37.32$.

Content last modified on 28 May 2021.

See a problem? Tell us or edit the source.

How to compute R-squared for a simple linear model

Description

Let's say we have fit a linear model to two columns of data, one for a single independent variable x and the other for a single dependent variable y. How can we compute R^2 for that model, to measure its goodness of fit?

Related tasks:

- How to fit a linear model to two columns of data
- How to compute adjusted R-squared (on website)

Solution in R

We assume you have already fit a linear model to the data, as in the code below, which is explained fully in a separate task, how to fit a linear model to two columns of data.

```
xs <- c( 393, 453, 553, 679, 729, 748, 817 )
ys <- c(  24,  25,  27,  36,  55,  68,  84 )
model <- lm( ys ~ xs )
```

You can get a lot of information about your model from its summary.

```
summary( model )
```

```
Call:
lm(formula = ys ~ xs)

Residuals:
      1       2       3       4       5       6       7 
  9.163   2.199  -9.072 -16.795  -4.431   6.047  12.890 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept) -37.32142   18.99544  -1.965  0.10664   
xs            0.13272    0.02959   4.485  0.00649 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 11.62 on 5 degrees of freedom
Multiple R-squared:  0.8009,	Adjusted R-squared:  0.7611 
F-statistic: 20.12 on 1 and 5 DF,  p-value: 0.006486
```

In particular, it contains the R^2 value.

```
summary( model )$r.squared
```

```
[1] 0.8009488
```

Content last modified on 01 June 2021.

See a problem? Tell us or edit the source.

How to do a one-way analysis of variance (ANOVA)

Description

If we have multiple independent samples of the same quantity (such as students' SAT scores from several different schools), we may want to test whether the means of each of the samples are the same. Analysis of Variance (ANOVA) can determine whether any two of the sample means differ significantly. How can we do an ANOVA?

Related tasks:

- How to do a two-sided hypothesis test for two sample means (which is just an ANOVA with only two samples)
- How to do a two-way ANOVA test with interaction (on website)
- How to do a two-way ANOVA test without interaction (on website)
- How to compare two nested linear models (on website)
- How to conduct a mixed designs ANOVA (on website)
- How to conduct a repeated measures ANOVA (on website)
- How to perform an analysis of covariance (ANCOVA) (on website)
- How to do a Kruskal-Wallis test (on website)

Solution in R

R expects you to have all the samples in one vector, and the groups they came from in a separate, categorical vector. So, for example, if we had SAT scores from four different schools (named A, B, C, and D), then our data might be arranged like this.

```
SAT.scores <- c(
    1100, 1250, 1390, 970, 1510, 1010, 1050, 1090, 1110,
    900, 1550, 1300, 1270, 1210, 900, 850, 1110, 1070, 910, 920
)
school.names <- c(
    'A', 'A', 'A', 'A', 'A', 'B', 'B', 'B', 'B',
    'C', 'C', 'C', 'C', 'C', 'D', 'D', 'D', 'D', 'D', 'D'
)
```

ANOVA tests the null hypothesis that all group means are equal. You choose α, the probability of Type I error (false positive, finding we should reject H_0 when it's actually true). I will use $\alpha = 0.05$ in this example.

```
# Run a one-way ANOVA and print a summary of all the output
result <- aov( SAT.scores ~ school.names )
summary( result )
```

```
             Df Sum Sq Mean Sq F value Pr(>F)
school.names  3 321715  107238   3.689 0.0342 *
Residuals    16 465140   29071
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The p-value reported in that output is 0.0433. You could manually check whether $p < \alpha$. Since it is, we would reject H_0, and therefore conclude that at least one pair of means is statistically significantly different.

Or you could ask R to do the comparison for you, but getting the p-value from the ANOVA summary is fiddly:

```
alpha <- 0.05
p.value <- unname( unlist( summary( result ) ) )[9]
p.value < alpha
```

```
[1] TRUE
```

Content last modified on 28 May 2021.

See a problem? Tell us or edit the source.

How to perform a chi-squared test on a contingency table

Description

If we have a contingency table showing the frequencies observed in two categorical variables, how can we run a χ^2 test to see if the two variables are independent?

Solution in R

Here we will use a 2×4 matrix to store a contingency table of education vs. gender, taken from Penn State University's online stats review website. You should use your own data. (Note: R's `table` function is useful for creating contingency tables from data.)

```
data <- matrix( c( 60, 54, 46, 41, 40, 44, 53, 57 ), ncol = 4,
                dimnames=list( c('F','M'), c('HS','BS','MS','PhD') ),
                byrow =TRUE)
data
```

```
  HS BS MS PhD
F 60 54 46 41
M 40 44 53 57
```

The χ^2 test's null hypothesis is that the two variables are independent. We choose a value $0 \leq \alpha \leq 1$ as the probability of a Type I error (false positive, finding we should reject H_0 when it's actually true).

R provides a `chisq.test` function that does exactly what we need.

```
results <- chisq.test( data )
results
```

```
    Pearson's Chi-squared test

data:  data
X-squared = 8.0061, df = 3, p-value = 0.04589
```

We can manually compare the p-value to an α we've chosen, or ask R to do it.

```
alpha <- 0.05            # or choose your own alpha here
results$p.value < alpha  # reject the null hypothesis?
```

```
[1] TRUE
```

Content last modified on 16 September 2021.

See a problem? Tell us or edit the source.