BIO781P Basic statistical tests in R cribsheet

Rob Knell

6 November 2016

Chi square test for independence

Here are some data from an experiment looking at the relationship between population of origin of zebrafish on their predator avoidance behaviour. Population 1 is heavily predated by birds whereas population 2 comes from a nearby region where all the predatory birds were hunted out some 50 years ago. Each individual fish was placed in a separate tank and tested by "flying" a silhouette of a predatory bird on a wire over the tank, and the fish were scored as to whether they showed avoidance behaviour or not. The table below shows the number of fish that were scored in each category.

	Population 1	Population 2
Avoidance behaviour	73	54
No avoidance	27	46

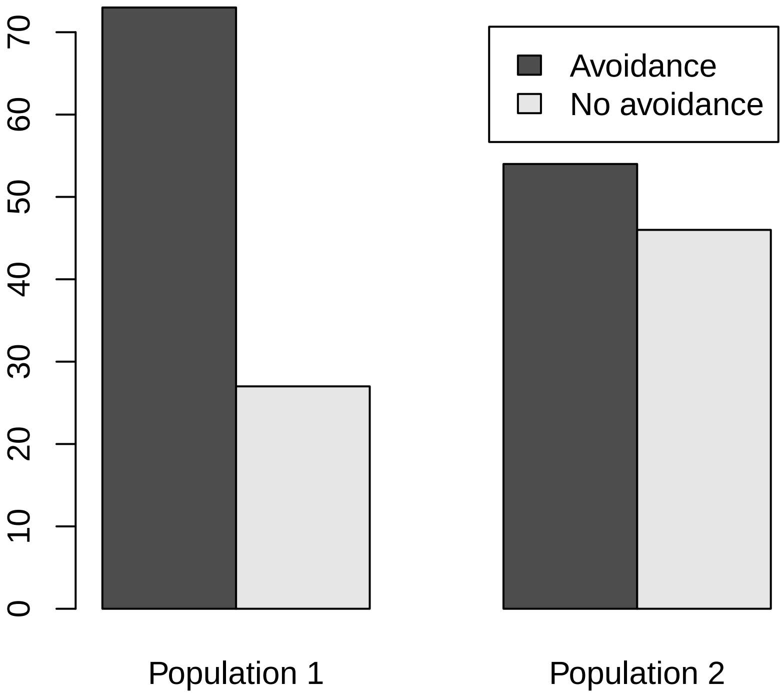
1. Input the data into R as a 2x2 matrix. Use the rownames() and colnames() functions to name the columns and rows.

```
avoidance <- matrix(c(73,27,54,46), nrow=2)
rownames(avoidance) <- c("Avoidance","No avoidance")
colnames(avoidance) <- c("Population 1", "Population 2")

avoidance
```

```
##              Population 1 Population 2
## Avoidance              73           54
## No avoidance           27           46
```

2. Draw a barplot of the data using the barplot() function. Include a legend.

```
barplot(avoidance, beside=TRUE, legend=TRUE)
```



3. Carry out a chi square test for independence of data using the chisq.test() function.

```r
chisq.test(avoidance)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  avoidance
## X-squared = 6.9895, df = 1, p-value = 0.008199
```

- What is the null hypothesis?

  No association between population and the frequency of avoidance behaviour

- What are the assumptions?

  Data are frequencies, data are independent, counts are higher than 5 per cell.

- What does it tell you?

  Fish from population 1 are signficantly more likely to show avoidance behaviour.

Chi square test for goodness of fit

Chi square test for goodness of fit In a 20 km2 area of the Etosha Pan National Park in Namibia, there are five dominant habitat types: salt pan, pan-edge shrubland, 'sweetveld' grassland, mopane-dominated savannah woodland and tall savannah woodland with other tree species such as Terminalia prunoides. The areas occupied by the various habitats are as follows (in km2).

Salt pan 7

Pan-edge shrubland 1

Sweetveld grassland 5

Mopane savannah 4

Tall savannah 3

A survey of Gemsbok (***Oryx gazella***) recorded the locations of 62 sightings of individuals or groups of animals over a one week period. The numbers in each habitat were:

Salt pan 11

Pan-edge shrubland 5

Sweetveld grassland 21

Mopane savannah 14

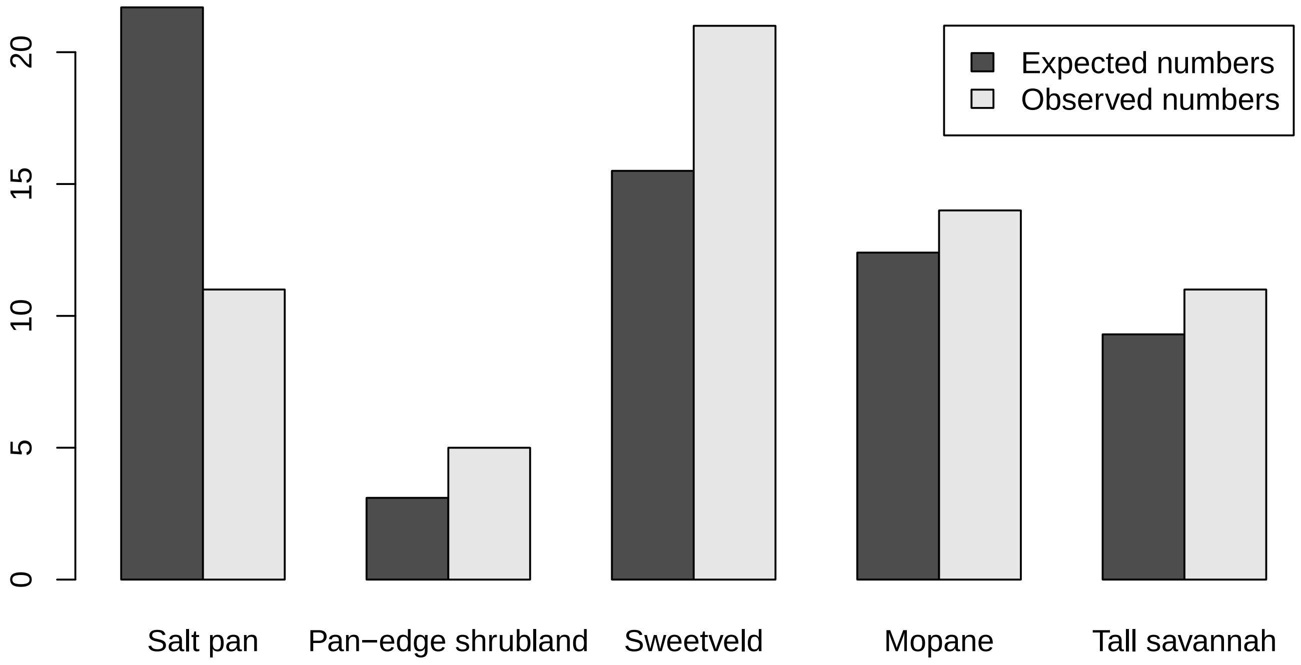
Tall savannah 11

1. Calculate the expected numbers of Gemsbok in each habitat area, assuming that they are evenly distributed according to the area of each.

```
areas <- c(7,1,5,4,3)
proportions <- areas/sum(areas)
gemsbok<-c(11,5,21,14,11)

expected.gemsbok<-sum(gemsbok) * proportions
```

2. Input your expected and observed numbers into R as a 5x2 matrix. Name the rows and columns as before and plot a barplot.

```
gemsbok.mat <- matrix(c(expected.gemsbok,gemsbok),byrow=TRUE, nrow=2)

rownames(gemsbok.mat) <- c("Expected numbers", "Observed numbers")
colnames(gemsbok.mat) <- c("Salt pan", "Pan-edge shrubland", "Sweetveld", "Mopane", "Tall savannah")

barplot(gemsbok.mat, beside=TRUE, legend=TRUE)
```



3. Carry out a chi-squared test for goodness of fit using the chisq.test() function. For a goodness of fit chisquare test you need to specify the expected values using the p= argument and if they are not probabilities you need to set the rescale.p argument to TRUE.

```
chisq.test(gemsbok.mat[2,], p=gemsbok.mat[1,], rescale.p=TRUE)
```

```
## Warning in chisq.test(gemsbok.mat[2, ], p = gemsbok.mat[1, ], rescale.p =
## TRUE): Chi-squared approximation may be incorrect

##
##  Chi-squared test for given probabilities
##
## data:  gemsbok.mat[2, ]
## X-squared = 8.9094, df = 4, p-value = 0.06341
```

4. Repeat the test but this time with a count of 10 Gemsbok in the salt pan and 22 in the grassland. How does the result change? What does this tell you about the value of statistical significance testing?

```
gemsbok2<-c(10,5,22,14,11)

expected.gemsbok2<-sum(gemsbok2) * proportions

gemsbok.mat2 <- matrix(c(expected.gemsbok2,gemsbok2),byrow=TRUE, nrow=2)

chisq.test(gemsbok.mat2[2,], p=gemsbok.mat2[1,], rescale.p=TRUE)
```

```
## Warning in chisq.test(gemsbok.mat2[2, ], p = gemsbok.mat2[1, ], rescale.p =
## TRUE): Chi-squared approximation may be incorrect

##
##  Chi-squared test for given probabilities
##
## data:  gemsbok.mat2[2, ]
## X-squared = 10.716, df = 4, p-value = 0.02995
```

Moving one animal from salt pan to grassland has changed the p-value from non-significant to apparently reasonably signficant…

T-tests and type I and II errors

NB Some of this is quite advanced. I've included the code for the first two parts of the exercise - try to work out what's happening and if you can, do the rest

The rnorm() function will generate random normally distributed numbers. It takes three arguments: the first tells the function how many numbers are required, the second is the mean of the distribution from which they're to be drawn and the third is the standard deviation of the distribution. t.test() is the function that carries out a t-test

1. Using rnorm write a script that will create two objects and then carry out a t-test to compare their means. The first should consist of 8 random normally distributed numbers with mean 10 and standard deviation 3, and the second should consist of 8 random normally distributed numbers with mean 12 and standard deviation 3.

```
X1 <- rnorm(8, 10, 3)
X2 <- rnorm(8, 12, 3)

t.test(X1,X2)
```

```
##
##  Welch Two Sample t-test
##
## data:  X1 and X2
## t = -1.1182, df = 11.455, p-value = 0.2864
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -5.960693  1.931732
## sample estimates:
## mean of x mean of y
##  9.143329 11.157809
```

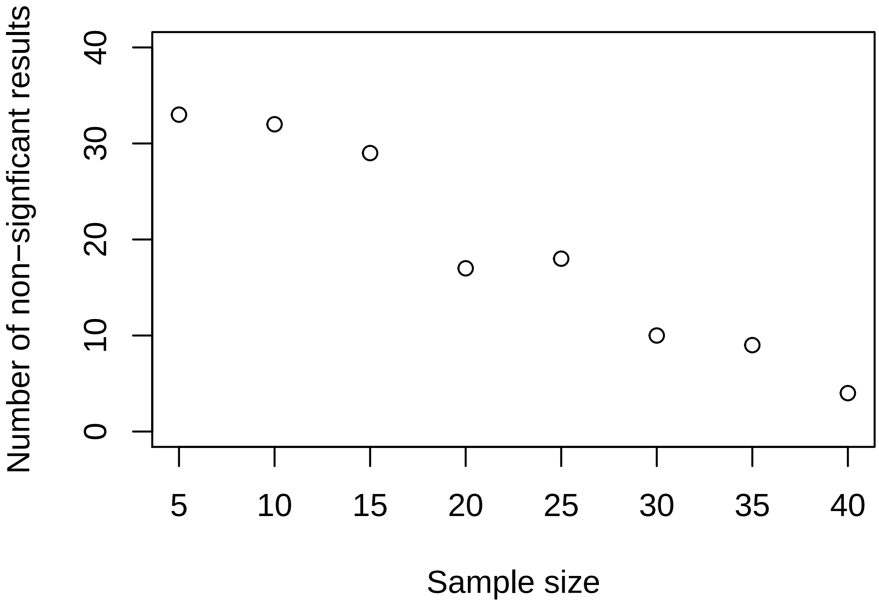
2. Run the script 40 times and count the number of times the p-value is below 0.05. You can include loops in R using the for (i in 1:x) {} code, where your instructions go between the curly brackets and the loop will be run x times. We can use this to look at simulated output from analysis in a much easier way than repeatedly running a script. Here's an example script that will do what you've just done but with each sample 15 numbers in length.

```
output<-matrix(nrow=40,ncol=2,data=NA) ##Set up matrix for output
colnames(output)<-c("Probability","Significant?")   ###Names for the matrix columns

for (i in 1:40) {   ###set up loop

  X1<-rnorm(8,10,3)   ###first vector of random numbers
  X2<-rnorm(12,12,3)   ###second vector
  test<-t.test(X1,X2)   ###Do the t-test
  output[i,1]<-test$p.value
  ###take the p-value and put it in the appropriate place in the output matrix
  output[i,2]<-ifelse(output[i,1]<0.05,1,0)
  ###If it's significant put a 1 in the second column, otherwise put a zero

}   ### close loop

paste("The number of significant results is:",sum(output[,2])) ###give the number of significant t-tests
paste("The number of non-significant results is:",40-sum(output[,2]))   ###and the non-significant ones
```

3. By making a few small changes you can use this script to see how the proportion of significant results changes with sample size. Run the script for sample sizes of 5,10,15,20,25,30,35 and 40 and plot a graph showing how the number of non-significant results changes as the sample size gets bigger.

```
output<-matrix(nrow=40,ncol=2,data=NA) ##Set up matrix for output
colnames(output)<-c("Probability","Significant?")   ###Names for the matrix columns
number.sig <- numeric(8)  ###vector for total number of significant results
```

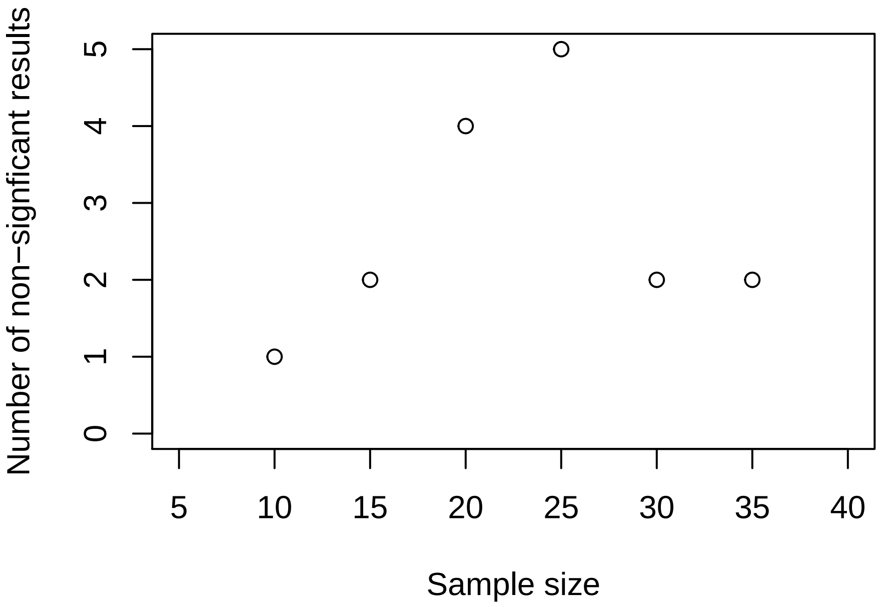
```
sample.sizes <- seq(from = 5, to = 40, by = 5)  # Use seq() function to make sequence of numbers


for (j in 1:8) {    ###set up first loop

    Z1<-sample.sizes[j]   ###Get sample size for this loop from vector of sample sizes

for (i in 1:40) {   ###set up second loop

  X1<-rnorm(Z1,10,3)   ###first vector of random numbers
  X2<-rnorm(Z1,12,3)   ###second vector
  test<-t.test(X1,X2)   ###Do the t-test
  output[i,1]<-test$p.value
  ###take the p-value and put it in the appropriate place in the output matrix
  output[i,2]<-ifelse(output[i,1]<0.05,1,0)
  ###If it's significant put a 1 in the second column, otherwise put a zero

}   ### close loop i loop

  number.sig[j]<-sum(output[,2])  ###write the number of significant tests for this iteration into the number.sig vec

} ### close j loop

plot(sample.sizes,40-number.sig, xlab="Sample size", ylab = "Number of non-signficant results", ylim=c(0,40))
```



4. Now change the script so that both random samples are drawn from the same population (i.e. the mean value that you put into rnorm is the same for both X1 and X2). Plot a graph showing how the number of significant results varies with sample size.

```
#######Script is exactly the same as the last chunk except the random samples are both drawn from
#######populations with a mean of 10 and we're plotting the number of signficant results, not the
#######number of non-signficant results
#######In this case any signficant result is a type 1 error - a false positive

output<-matrix(nrow=40,ncol=2,data=NA) ##Set up matrix for output
colnames(output)<-c("Probability","Significant?")   ###Names for the matrix columns
number.sig <- numeric(8)  ###vector for total number of significant results

sample.sizes <- seq(from = 5, to = 40, by = 5)  # Use seq() function to make sequence of numbers
```

```
for (j in 1:8) {    ###set up first loop

    Z1<-sample.sizes[j]   ###Get sample size for this loop from vector of sample sizes

for (i in 1:40) {   ###set up second loop

  X1<-rnorm(Z1,10,3)   ###first vector of random numbers
  X2<-rnorm(Z1,10,3)   ###second vector - note that the mean is now 10 not 12
  test<-t.test(X1,X2)   ###Do the t-test
  output[i,1]<-test$p.value
  ###take the p-value and put it in the appropriate place in the output matrix
  output[i,2]<-ifelse(output[i,1]<0.05,1,0)
  ###If it's significant put a 1 in the second column, otherwise put a zero

}   ### close loop i loop

  number.sig[j]<-sum(output[,2])  ###write the number of significant tests for this
  ###iteration into the number.sig vector

} ### close j loop

plot(sample.sizes,number.sig, xlab="Sample size", ylab = "Number of non-signficant results",
     ylim=c(0,5))
```



5. Your first graph shows an estimate of the Type II error rate – the chance of not finding a significant result when the two samples are drawn from different populations. Your second graph shows an estimate of the Type I error rate – the chance of detecting a significant difference when the two samples are drawn from the same population. Is there a difference between the two graphs? What is the difference and why?

I'll leave that to you to work out.