Teste 1

Schmidt, Sophie C.

29 Mai, 2019

knitr::opts\_chunk$set(  
 collapse = TRUE,  
 warning = FALSE,  
 message = FALSE,  
 echo = TRUE,  
 comment = "#>",  
 fig.path = "../figures/",  
 fig.width=2,   
 fig.height=2  
)

Pakete für heute:

library(dplyr)  
library(tidyr)  
library(archdata)

## Binomialtest

Den ersten Test, den wir heute kennenlernen, ist der Binomialtest. Er testet die Häufigkeit eines Auftretens zweier **nominalen** Werte gegenüber einer angenommenen theoretischen Verteilung. Unser Beispiel sieht so aus:

In einem Gräberfeld wurden 200 Bestattungen ausgegraben, davon konnten alle exakt als männlich oder weiblich bestimmt werden und es sind 75 Frauen und 125 Männer.

Wir stellen deshalb eine Hypothese auf: Der Frauen- und Männeranteil der Gemeinschaft war genau wie bei unserer 50:50 (biologisch gut begründet vermutbar). Strukturell werden Frauen also weniger häufig bestattet als Männer.

Daraus entwickeln wir die statistischen Hypothesen:

* H0 = Die Stichprobe lässt keinen Rückschluss auf die Grundgesamtheit zu und es kann keine strukturelle Benachteilung festgestellt werden
* H1 = ungerichtet: Es gibt Unterschiede in der Häufigkeit der Bestattung nach Geschlecht; gerichtet: Männer werden häufiger bestattet als Frauen

Der Binomialtest ist in R base umgesetzt und sehr leicht ausführbar:

# Das ist die allgemeine Syntax binom.test(nsuccesses, ntrials, p, alternative="greater")  
# unser Bsp sieht so aus:  
  
binom.test(125, 200, 0.5, alternative="greater")  
#>   
#> Exact binomial test  
#>   
#> data: 125 and 200  
#> number of successes = 125, number of trials = 200, p-value =  
#> 0.0002497  
#> alternative hypothesis: true probability of success is greater than 0.5  
#> 95 percent confidence interval:  
#> 0.5650678 1.0000000  
#> sample estimates:  
#> probability of success   
#> 0.625  
  
# es lässt sich auch anders herum formulieren:  
  
binom.test(75, 200, 0.5, alternative = "less") # jetzt zählen wir die 75 Frauengräber als "successes" und müssen folglich auch die Richtung des Tests mit alternative "umdrehen"  
#>   
#> Exact binomial test  
#>   
#> data: 75 and 200  
#> number of successes = 75, number of trials = 200, p-value =  
#> 0.0002497  
#> alternative hypothesis: true probability of success is less than 0.5  
#> 95 percent confidence interval:  
#> 0.0000000 0.4349322  
#> sample estimates:  
#> probability of success   
#> 0.375  
  
# oder wir sagen: keine Ahnung, in welche Richtung das Ungleichgewicht geht:  
  
binom.test(75, 200, 0.5, alternative = "two.sided")  
#>   
#> Exact binomial test  
#>   
#> data: 75 and 200  
#> number of successes = 75, number of trials = 200, p-value =  
#> 0.0004994  
#> alternative hypothesis: true probability of success is not equal to 0.5  
#> 95 percent confidence interval:  
#> 0.3077138 0.4460514  
#> sample estimates:  
#> probability of success   
#> 0.375

Easy no?

Dann setzen wir das doch einmal mit einem echten archäolgischen Datensatz um. Am einfachsten ist es, wir nehmen genau die gleiche Fragestellung mit dem Gräberfeld Ernest Witte in Texas:

data(EWBurials)  
# am besten ihr schaut ihn euch einmal an. Wie wär es mit View oder ?

So, jetzt wissen wir, wie die Spalte heißt, auf die wir uns beziehen möchten. Was wir jetzt machen können, ist eine Verschachtelung von Anfragen. Zuerst fragen wir danach, wie viele Einträge es denn in der Spalte Sex des Datensatzes EWBurials gibt, die “Male” heißen. Das zählen übernimmt die praktische kleine Funktion “nrow” (n bedeutet die Größe eine Stichprobe und wir zählen rows, also Zeilen).

Wir fragen also gleich wie viele gibt es im Datensatz EWBurials, wo die Spalte des Datensatzes EWBurials namens “Sex” den Eintrag “Male” hat. Die Einschränkung, welche Zeilen gezählt werden sollen (die mit “male”) kommt in eckigen Klammern hinter den Datensatz. Das KOMMA dahinter darf nicht vergessen werden! Weil die Ansprache in eckigen Klammern ja immer mit [Zeile,Spalte] organisiert ist, müssen wir den Spaltenhinweisplatz in der Formel kreiieren, auch wenn wir ihn frei lassen:

# erst einmal Trockenübung:  
# wir fragen, wie viele Zeilen gibt es denn überhaupt?  
nrow(EWBurials)  
#> [1] 49  
# cool! Und jetzt nur die, die männlich sind?  
nrow(EWBurials[EWBurials$Sex == "Male",])  
#> [1] 25  
# klappt oder?  
  
  
#Dann fügen wir das doch in den binom-Test ein:  
  
binom.test(nrow(EWBurials[EWBurials$Sex == "Male",]), nrow(EWBurials), 0.5, alternative = "two.sided")  
#>   
#> Exact binomial test  
#>   
#> data: nrow(EWBurials[EWBurials$Sex == "Male", ]) and nrow(EWBurials)  
#> number of successes = 25, number of trials = 49, p-value = 1  
#> alternative hypothesis: true probability of success is not equal to 0.5  
#> 95 percent confidence interval:  
#> 0.3633783 0.6557526  
#> sample estimates:  
#> probability of success   
#> 0.5102041

Wie können wir das Ergebnis interpretieren?

Der p-Wert ist 1!

Es gibt also eine 100%ige Wahrscheinlichkeit, dass wir uns irren, wenn wir die Alternativhypothese akzeptieren… oh… ups…

Dann müssen wir wohl davon ausgehen, dass es hier eine Gleichverteilung von Männern und Frauen gibt.

Keine große Überraschung für die, die sich die Zahlen vorher angeschaut haben. ;-)

Herzlichen Glückwunsch!

So einfach ist ein Binomialtest!

### Chi-Quadrat-Test

Die erste Anwendung für den Chi-Quadrat-Test, die wir benutzen wollen, ist ähnlich wieder Binomialtest. ABER ich kann sie für deutlich mehr Werte anwenden als nur zwei. Aus Einfachheitsgründen machen wir das aber hier nochmal mit zwei Werten und einer einfachen Verteilung.

# die neolithischen Gräber  
nl\_G <- c(10,50)  
# die Wahrscheinlichkeit, mit der sie sich eigentlich auf die zwei Gruppen verteilen sollten  
Vert\_fl <- c(0.404,0.596)  
# voll simpel der Test  
chisq.test(nl\_G, p= Vert\_fl)  
#>   
#> Chi-squared test for given probabilities  
#>   
#> data: nl\_G  
#> X-squared = 14.036, df = 1, p-value = 0.0001794

Allerdings sind die Ergebnisse, die er uns gibt, nicht ganz so spannend, wie wenn wir es selber ausrechnen. Ich will zB unbedingt herausfinden, wie die Erwartungswerte sind. Dann kann ich nämlich auch erst abschätzen, ob ich den Test überhaupt nehmen sollte (Erinnerung: nicht mehr als 20% der Erwartunswerte sollten unter 5 fallen).

Deswegen weise ich *den ganzen Test* einer Variablen zu. Und dann kann ich nachschauen, wie die Erwartungswerte aussehen und was mich noch alles interessiert.

# ich nenne die Variable chi  
chi <- chisq.test(nl\_G, p = Vert\_fl)  
chi$expected  
#> [1] 24.24 35.76  
chi$p.value  
#> [1] 0.000179351

Ich kann v.a. auch coolerweise innerhalb meines R-markdown-Dokumentes den P-Wert aufrufen. Wenn ich ` r chi$p.value ` schreibe, bekomme ich: 1.793509710^{-4}.

Machen wir weiter mit etwas komplizierterem. Wie wär es wenn es mal mehr als nur zwei Ausprägungen der Variable geben dürfte? Ist doch langweilig, immer nur nach der Verteilung einer Variable zu schauen…

### Chi Quadrat mit zwei oder mehr Variablen!

Also wie wär es, wenn wir schauen, ob Grabbeigaben und Geschlecht zusammenhängen? In der Tabelle von EWBurials wurde nur das Vorhandensein oder die Abwesenheit von Grabbeigaben aufgenommen.

Der Chi-Quadrat-Test von R arbeitet am liebsten mit tables oder matrices (Matrizen, pl. von Matrix).

Tables entstehen, wenn man die table-Funktion benutzt. Sie sind auch eine Arte Datenformat in R, das im Hintergrund anders aussieht als ein Dataframe. Muss uns nicht weiter beschäftigen, wir brauchen eh die Gegenüberstellung von Werten, wie wir sie mit table bekommen:

# zuerst schauen wir einmal, wie diese Table aussieht:  
table(EWBurials$Sex, EWBurials$Goods)  
#>   
#> Absent Present  
#> Female 13 11  
#> Male 10 15  
  
# und dann bauen wir sie einfach in den Befehl für den Chi-Quadrat-Test ein (chisq.test)  
chisq.test(table(EWBurials$Sex, EWBurials$Goods))  
#>   
#> Pearson's Chi-squared test with Yates' continuity correction  
#>   
#> data: table(EWBurials$Sex, EWBurials$Goods)  
#> X-squared = 0.49987, df = 1, p-value = 0.4796

Oje! Ein p-Wert von fast 0,5… was heißt das?

Eine Chance von 50% dass wir uns irren, wenn wir die Alternativhypothese annehmen wollen?

Na toll.

Das ist ja richtig hilfreich.

Aber durchaus gar nicht so selten. Was können wir also sagen?

“Es gibt keine statistischen Hinweis darauf, dass die An- oder Abwesenheit von Grabbeigaben etwas mit dem Geschlecht zu tun hat.”

Die Abweichungen sind ja auch wirklich gering.

Jetzt nehmen wir dochmal zwei Variablen, die mehr als nur eine Ausprägung haben. Zum Bsp die Altersgruppen “Age”, wer in die Tabelle schaut, sieht, dass es da “Child”, “Adolescent”, “young Adult” und noch mehr Kategorien gibt. Haben diese evtl. etwas mit dem Auftreten der Grabbeigaben zu tun? Ist ja durchaus eine gängige archäologische These.

tab\_A\_G <- table(EWBurials$Age, EWBurials$Goods)  
  
chi\_tab\_A\_G <- chisq.test(tab\_A\_G)

Wie muss dieser p-Wert interpretiert werden?

Schauen wir doch vorher nochmal kurz, ob wir den Test überhaupt machen können.

chi\_tab\_A\_G$expected  
#>   
#> Absent Present  
#> Child 0.9387755 1.061224  
#> Adolescent 1.4081633 1.591837  
#> Young Adult 8.9183673 10.081633  
#> Adult 1.4081633 1.591837  
#> Middle Adult 4.6938776 5.306122  
#> Old Adult 5.6326531 6.367347

Najaaaa… eigentlich nicht.

Warum nicht?

Was könnte man tun, um das Problem zu umgehen?

#### Noch mal ein anderes Beispiel..

Nehmen wir dafür Daten von den Neanderthaler-Fundstellen Ferrassie und Castanet. Da sind so schön viele nominale Variablen drin, die wir gegeneinander testen können. Schauen wir sie uns einmal an:

data("EndScrapers") # Daten laden  
View(EndScrapers) # anschauen

Hmmmh, aber da tauchen die Häufigkeitsangaben schon in einer extra Spalte auf. Wie kann ich damit umgehen?

Wer sich an das letzte Skript erinnert, erinnert sich jetzt vllt auch an “aggregate”. Wir wollen vllt einfach mal untersuchen, ob Fundstelle und Retuschierungsgrad zusammenhängen. Dafür kommen jetzt die folgenden Schritte:

1. ich summiere die Freq (also die Häufigkeiten) wobei Site und Retouched die beiden Variablen sind, die behalten werden
2. ich schau sie mir an, hat alles geklappt?
3. ich muss jetzt die Tabelle so umformen, dass die Häufigkeiten unter den Fundstellennamen stehen. Dafür brauche ich spread.
4. Anschauen: Alles klar?
5. Leider ist jetzt immer noch alles ein dataframe. D.h. die Angabe unter “Retouched” ist immer noch ein Vektor und keine direkte Zuweisung der daneben stehenden Werte… was wir brauchen ist eine Matrix. Matrizen zeichnen sich dadurch aus, dass sie Spalten- und Zeilennamen haben. Dafür brauchen wir also folgende Schritte:

5.a) Ich mache aus Retouched Namen der Zeilen

* 1. Ich lösche die Spalte Retouched

1. Jetzt habe ich eine Matrix. Damit kann chisq.test wunderbar umgehen.

Ooookeee.. Los geht’s:

# 1.  
end\_site\_ret <- aggregate(EndScrapers$Freq, by = list(Site = EndScrapers$Site, Retouched = EndScrapers$Retouched), FUN = sum)  
# dran denken, die Freq-Spalte wird jetzt x genannt  
  
# 2.  
head(end\_site\_ret) # head ist ein kleiner Trick, der mir nur die ersten paar Einträge eines data frames zeigt   
#> Site Retouched x  
#> 1 Castenet A Retouched 868  
#> 2 Ferrassie H Retouched 592  
#> 3 Castenet A Unretouched 330  
#> 4 Ferrassie H Unretouched 1210  
  
# 3. die Angaben unter x werden meine Werte in der Tabelle, die Site-Angaben werden meine Spaltenköpfe:  
end\_sr <- end\_site\_ret %>% spread(value = "x", key = "Site")  
  
# 4. alles klar?  
head(end\_sr) # anschauen  
#> Retouched Castenet A Ferrassie H  
#> 1 Retouched 868 592  
#> 2 Unretouched 330 1210  
  
# 5. a) rownames ist der Befehl: der data frame end\_sr bekommt die rownames, die unter end\_sr$Retouched liegen   
rownames(end\_sr) <- end\_sr$Retouched  
head(end\_sr) # einmal checken  
#> Retouched Castenet A Ferrassie H  
#> Retouched Retouched 868 592  
#> Unretouched Unretouched 330 1210  
  
# 5. b) jetzt brauch ich die Spalte end\_sr$Retouched nicht mehr. weg damit:  
end\_sr <- end\_sr[,-1]  
head(end\_sr)  
#> Castenet A Ferrassie H  
#> Retouched 868 592  
#> Unretouched 330 1210

Puuuh, aber jetzt wird es spannend: Klappt der Chi-Quadrat-Test?

chi <- chisq.test(end\_sr)  
  
chi  
#>   
#> Pearson's Chi-squared test with Yates' continuity correction  
#>   
#> data: end\_sr  
#> X-squared = 450.15, df = 1, p-value < 2.2e-16  
  
chi$expected  
#> Castenet A Ferrassie H  
#> Retouched 583.0267 876.9733  
#> Unretouched 614.9733 925.0267

Yaay! Erfolg!

Aber was ist das für ein komischer p-Wert ? “2.2e-16” ?

Das ist die wissenschaftliche Notation von 2,2 mal e hoch -16.

* e ist die eulersche Zahl, rund 2,718
* hoch minus etwas ist ein Bruch: 2⁻² = 2/2² = 2/4 = 1/2
* 2,2 \* 2,718⁻¹⁶ = 2,2/2.781¹⁶ = winzig klein

p ist also winzig klein! Yeah!

Wie interpretier ich das?

## Stärke eines Zusammenhangs

Um die **Stärke** eines Zusammenhangs zu messen, kann ich nicht den p-Wert des Chi-Quadrat-Tests nehmen.

Dafür nehme ich am besten Cramér’s V… oder Cramérs Index (CI) oder K… manchmal bennenen Statistiker Sachen unterschiedlich und das ist voll nervig. Aber egal. Alles das gleiche.

Es gibt mehrere Pakete, die man installieren kann, die diese Berechnungen anstellen, in base r ist das leider nicht umgesetzt. Wir installieren “lsr”, das Paket for *Learning Statitics with R*.

install.packages("lsr", dependencies = TRUE)  
library(lsr)

Im Paket lsr gibt es die Funktion “cramersV”. Lasst uns die mal ausprobieren. Wir geben ihr einfach die gleiche Matrix wie der Chi-Quadrat-Funktion:

cramersV(end\_sr)   
#> [1] 0.3873649

Und was sagt uns das Ergebnis nun?

Cramérs V kann Werte zwischen 0 und 1 annehmen. Ist V = 0, dann besteht kein Zusammenhang, ist V = 1, besteht ein vollständiger Zusammenhang. Der Wert von 0,39 ist gar nicht so übel. Es ist ein moderater Zusammenhang feststellbar.

## Herzlichen Glückwunsch!

Wir haben mit dem binomial-Test geschaut, ob sich eine Variable einer bestimmten Verteilung entsprechend verhält und haben dafür einen Wahrscheinlichkeitswert errechnet.

Wir haben den Chi-Quadrat-Test genutzt, um a) das gleiche wie mit dem binomial-Test zu überprüfen und b) die Abhängigkeit zweier Variablen zueinander zu untersuchen und auch dafür immer Warhscheinlichkeitswerte errechnet.

Und wir haben jetzt die Stärke eines Zusammenhangs berechnet.

Super!