***Learning Statistics with R - University of Adelaide***

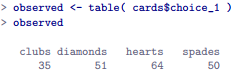
***Part V – Statistical Tools***

**Chapter 12 – Categorical Data Analysis**

* Now that we’ve got the basic theory behind hypothesis testing, it’s time to start looking at specific tests commonly used in psychology.
* Both **χ 2 tests** + **t-tests** are very frequently used in scientific practice, + while they’re not as powerful as **analysis of variance (ANOVA)** + **regression**, they’re much easier to understand.
* The term **categorical data** = another name for **nominal scale data**
* **Categorical Data Analysis** = collection of tools you can use when your data are nominal scale.

**12.1 The χ 2 goodness-of-fit test**

* **The χ 2 goodness-of-fit test** = 1 of the oldest hypothesis tests around invented by Karl Pearson around the turn of the century w/ some corrections made later by Sir Ronald Fisher
* Over the years, there have been a lot of studies showing humans have a lot of difficulties in simulating randomness.
* Try as we might to “act” random, we think in terms of patterns + structure, + so when asked to do something at random, what people actually do is anything but random.
* As a consequence, the study of human randomness (or non-randomness, as the case may be) opens up a lot of deep psychological questions about how we think about the world.
* W/ this in mind, consider a very simple study 🡪 ask people to imagine a shuffled deck of cards + mentally pick 1 card from this imaginary deck at random.
* After they’ve chosen 1 card, ask them to mentally select a 2nd.
* For both choices, look at suit + after asking n = 200 people, look at the data + figure out whether or not the cards people pretended to select were really random.



* Looking at our frequency table, there’s a hint people might be more likely to select hearts than clubs, but it’s not completely obvious just from looking at it whether that’s really true or just due to chance.
* From this point on, treat this table as the data we’re looking to analyze
* Mathematical notation for an element of the observed vector O 🡺 O(i)(j) = j-ith observation in i-th category (where i = 1, 2, 3 or 4)
* To refer to a set of *all* observed frequencies, statisticians group all observed values into a vector:
* **O** = (O1, O2, O3, O4)
* Our research hypothesis is that people DON’T choose cards randomly.
* Want to translate this into + construct a statistical test of some statistical hypotheses
* **Pearson’s χ 2 goodness of fit test**
* 1) Construct our null 🡪 H0: All four suits are chosen w/ equal probability
* To say the same thing in a mathematical way, use the notation **P(j)** = *true* probability the j-th suit is chosen.
* *If the null is True*, each suit has a 25% chance of being selected 🡪 Null claims P1-P4 each = .25
* In the same way we group observed frequencies into a vector O to summarize an entire dataset, use P to refer to the probabilities that correspond to our null.

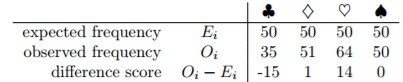


* In this particular instance, our null corresponds to a vector of probabilities, P, in which all probabilities are equal
* 2) Experimental task could be for people to imagine they were drawing from a deck w/ 2X as many clubs as any other suit 🡪 H0: 
* *As long as all probabilities are positive + sum up to 1, it’s a perfectly legitimate choice for the null*
* Most common use of the goodness of fit test = test a null where all categories are equally likely
* 2) Construct Alternative H1
* All we’re really interested in is demonstrating the probabilities involved AREN’T all identical (people’s choices WEREN’T completely random).





* At this point, we have our observed frequencies O + a collection of probabilities P corresponding the null we want to test.
* 3) Construct a test of the null
* If we want to test H0 against H1, we need a test statistic that measures how close the data are to the null.
* If the data don’t resemble what you’d expect to see if the null were true, it probably isn’t true.
* If this null were true, we’d expect to see frequencies (n = 200 observations) of P(i) = .25 🡺 50 of each suite
* If E(i) refers to the # of category i responses we’re expecting if null is true, then **E(i) = n \* P(i)**
* On average we’d expect to see 50 observations in each category
* To translate this into a test statistic, want to compare the expected observations in a category (i) E(i) w/ observed observations for that category O(i)
* Calculate the difference between what the null expected + what we actually did find 🡪 O(i – E(i)



* Clear people chose more hearts + fewer clubs than the null predicted.
* However, a moment’s thought suggests the raw differences aren’t quite what we’re looking for
* Feels like it’s just as bad when the null predicts *too few* observations (like w/ hearts) as it is when it predicts *too many* (as w/ clubs).
* A bit weird we have a negative for clubs + a positive for hearts.
* 1 easy way to fix this is to *square everything*



* Now have a collection of #’s that’re big whenever the null makes a bad prediction but small whenever it makes a good one, no matter the sign
* Also must divide these by the expected frequency E(i)
* Since E(i) = 50 for all categories in our example, it’s not a very interesting calculation



* See 4 different **error scores**, each one telling us how big a mistake the null made when we tried to use it to predict our observed frequencies.
* To convert this into a useful test statistic, could just add these up = **goodness of fit statistic, X2** or **GOF**.



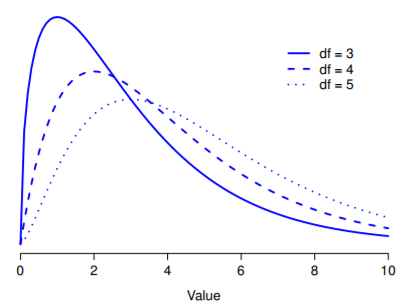
* If we let k refer to total # of categories, the X2 statistic is given by



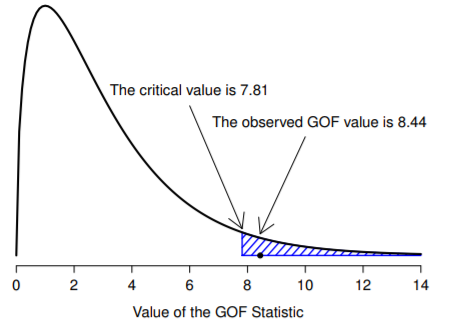
* It’s clear that if X2 is small, the observed data are very close to what the null predicted 🡪 we need a large X2 statistic in order to reject the null.
* Is 8.44 a big enough value to reject the null?
* To determine whether or not a particular value of X2 is large enough to justify rejecting the null, figure out what the *sampling distribution* for X2 would be if the null were true.
* Suppose the null is *actually true* + if so, the *true probability* an observation falls in the i-th category is P(i)
* Think about what this actually means: Kind of like saying *nature* makes the decision about whether or not the observation ends up in category i by flipping a *weighted* coin (probability of getting a head = P(j))
* Therefore can think of our observed frequency O(i) by imagining nature flipped n of these coins (1 for each observation in the dataset) + exactly O(i) of them came up heads.
* This is a pretty weird way to think about the experiment but what it does is remind you we’ve seen this scenario before = *exact same set up that gave rise to the binomial distribution*
* In other words, if the null is true, it follows that our observed frequencies were generated by sampling from a binomial distribution:



* Now, if you remember from the CLT discussion, the binomial distribution starts to look identical to the normal distribution, especially when N is large + when P(i) isn’t too close to 0 or 1.
* In other words, as long as N \* P(i) is large enough (or “when the expected frequency E(i) is large enough), the theoretical distribution of O(i) is approximately normal.
* Better yet, if O(i) is normally distributed, then so is 
* Since E(i) is a *fixed value*, subtracting off E(i) + dividing by the square root of E(i) changes the mean + SD of the normal distribution; but that’s all it does.
* Goodness of fit statistic 🡪 taking a bunch of things that’re normally-distributed, squaring them, + adding them up.
* Reminder: when you take a bunch of things that have a standard normal distribution (i.e., µ = 0 + SD = 1), square them, + add them up, the resulting quantity has a **chi-square distribution**.
* So now we know the null predicts that the sampling distribution of the goodness of fit statistic is a chi-square distribution.
* Remember if the # of things you’re adding up is k, the **degrees of freedom** for the resulting chi-square distribution is k, but the ACTUAL dF for the chi-square goodness of fit test is k - 1.
* Supposed to be looking at the # of *genuinely independent* things getting added together
* Even though we’re adding k things, only k - 1 are truly independent, so dF is actually only k - 1.
* If we change the dF, the chi-square distribution changes shape quite substantially.



* **dF** = the # of normally distributed variables we’re squaring + adding together
* The basic idea behind dF is quite simple: calculate by counting up the # of distinct quantities used to describe your data + then subtracting off all the constraints those data must satisfy.
* This is an over-simplification + works nicely for quite a few situations
* But every now + then we’ll come across dF values that aren’t *whole* numbers
* When you come across this, remind yourself dF is actually a bit of a messy concept,
* For an introductory class, it’s best to stick to the simple story
* This is a bit vague, so let’s use our cards data as a concrete example.
* We describe our data using 4 numbers, O1, O2, O3 + O4, corresponding to the observed frequencies of the 4 different categories (hearts, clubs, diamonds, spades).
* These 4 numbers = the random outcomes of our experiment.
* But, my experiment actually has a fixed constraint built into it: the *sample size*, n
* In practice, sample size isn’t always fixed (might run the experiment over a fixed period of time, + the # of people participating depends on how many people show up)
* That doesn’t matter for the current purposes.
* That is, *if we know how many people chose hearts, how many chose diamonds + how many chose clubs; we’d be able to figure out exactly how many chose spades.*
* Although our data are described using 4 numbers, they only actually correspond to 4 - 1 = 3 dF
* There are 4 probabilities we’re interested in (corresponding to the 4 different categories), but these probabilities must sum to 1, which imposes a constraint.
* Therefore, the dF is 4 -1 = 3.
* In general, in chi-square goodness of fit tests for an experiment involving k groups, dF = k - 1.
* The final step in the process of constructing our hypothesis test is to figure out what the **rejection/critical region** is 🡪 values of X2 that lead us to reject the null.
* As we saw earlier, large values of X2 imply the null has done a poor job of predicting the data from our experiment, whereas small values of X2 imply it’s actually done pretty well.
* Therefore, a pretty sensible strategy is to say there’s some critical value such that if X2 is bigger than it, we reject the null, but if X2 is smaller than it, we retain the null.
* The chi-squared goodness of fit test is always a one-sided test.
* Have to figure out what this critical value is
* If we want our test to have significance level of α = .05 (willing to tolerate a Type I error rate of 5%), choose a critical value so that there’s only a 5% chance X2 could get to be that big if the null is true.
* i.e., we want the 95th percentile of the sampling distribution.



* To calculate the 95th percentile of a chi-squared distribution w/ k - 1 dF use **qchisq()** in R



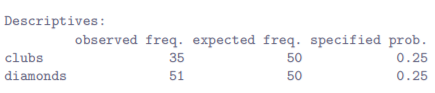
* So if X2 is bigger than 7.81, we can reject the null + we calculated X2 = 8.44, so we reject the null.
* If we want an *exact* p-value, calculate it w/ pchisq() function

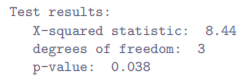


* Recall that the **“p” form** of the probability distribution functions in R always calculate the probability of getting a value of *less than the value you entered* (8.44).
* We want the *opposite* 🡪 probability of getting a value of 8.44 *or more* 🡪 that’s why we told R to use the *upper tail*, NOT the lower tail
* Usually easier to calculate the p-value w/ **1 - pchisq(8.44, 3 )**
* So, in this case we reject the null, since p < .05
* R provides a function that will do all of these calculations 🡪 most people use **chisq.test()** + can also use **goodnessOfFitTest()** in the **lsr**





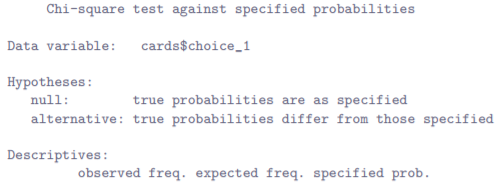


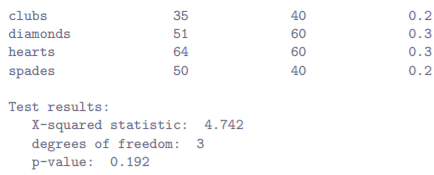


* Notice these are the same numbers we came up with when doing the calculations the long way.
* What if you want to run a goodness of fit test, but the null is now the theoretical prediction = people choose red cards 60% of the time + black cards 40%
* If that were the case, null = expect 30% of choices = hearts, 30% = diamonds, 20% = spades + clubs.
* Specify the probabilities associated with the new null + include it in the command







* 
* As you can see the null + expected frequencies are different to last time + as a consequence our X2 test statistic + p-value are different too 🡪 p-value = .192, so we can’t reject the null = these data don’t provide enough evidence against it.
* There’s no point in designing + running an experiment + analyzing the data if you don’t tell anyone about it
* If I wanted to write this result up for a paper or something, the conventional way to report this would be to write something like this:
* **“Of the 200 participants in the experiment, 64 selected hearts for their first choice, 51 selected diamonds, 50 selected spades, and 35 selected clubs. A chi-square goodness of fit test was conducted to test whether the choice probabilities were identical for all four suits. The results were significant (χ2(3) = 8.44, p < .05), suggesting that people did not select suits purely at random.”**
* This is pretty straightforward + hopefully it seems pretty unremarkable, but there’s a few things you should note about this description:
* The **statistical test** is *preceded by the* ***descriptive statistics*** 🡪 told readers something about what the data look like before going on to do the test
* Always remember readers don’t know your data anywhere near as well as you do.
* Unless you describe it to them properly, the statistical tests won’t make any sense to them
* The description tells you what the null being tested is.
* Writers don’t always do this: often a good idea in situations where some ambiguity exists or you can’t rely on readership being intimately familiar w/ statistical tools you’re using
* Quite often readers might not know (or remember) all details of a test your using, so it’s a politeness to “remind” them
* As far as goodness of fit test goes, you can usually rely on a scientific audience knowing how it works
* However, it’s still a good idea to be explicit about stating the null (briefly) b/c the null can be different depending on what you’re using the test for.
* For instance, in the cards example, null was that all the 4 suit probabilities were identical, but there’s nothing special about that hypothesis.
* Could just as easily have tested the null that P1 = 0.7, P2 = P3 = P4 = 0.1
* So it’s helpful to the reader if you explain to them what your null was.
* Also, notice the null was described hypothesis in words, not in math.
* Can describe it in math, but since most readers find words easier to read than symbols, most writers tend to describe the null using words if they can.
* A **stat block** is included.
* When reporting results of the test itself, it didn’t just say “the result was significant”, it included a **stat block** = reports all the raw statistical data.
* For the chi-square goodness of fit test, the info reported = test statistic (8.44) + info about the distribution used in the test (χ2 w/ 3 degrees of freedom 🡪 shortened to χ2(3), + then info about whether the result was significant (in this case p < .05).
* *Particular info that needs to go into the stat block is different for every test*
* General principle = should always provide enough info so that readers could check the test results themselves if they really wanted to.
* The results are interpreted.
* In addition to indicating the result was significant, an interpretation of the result was provided (i.e “*people didn’t choose randoml*y”).
* A kindness to readers b/c it tells them something about what they should believe about what’s going on in your data.
* If you don’t include something like this, it’s can be hard for readers to understand what’s going on.
* As with everything else, overriding concern should be to *explain things to readers*.
* Always remember the point of reporting results = *to communicate to another human being.*
* Writing χ2(3) = 8.44 is essentially a highly condensed way of writing “the sampling distribution of the test statistic is χ2(3), + the value of the test statistic is 8.44”.
* In 1 sense, this is kind of stupid as there are lots of different test statistics that have a chi-square sampling distribution
* The X2 statistic used for a goodness of fit test is only 1 of many (albeit 1 of the most encountered ones).
* In a perfectly-organized world, we’d always have a separate name for the test statistic + the sampling distribution so that the stat block itself tells you exactly what a researcher had calculated
* Sometimes this happens: test statistic used in Pearson goodness of fit test is written X2, but there’s a closely related test known as the G-test in which the test statistic is written = **G**.
* As it happens, the Pearson goodness of fit test + G-test both test *the same null hypothesis +* *the sampling distribution is exactly the same* (i.e., chi-square w/ k - 1 dF).
* If I’d done a G-test for the cards data rather than a goodness of fit test, I’d have ended up w/ a test statistic of G = 8.65, slightly different from the X2 = 8.44 value
* It also produces a slightly smaller p-value of p = .034.
* Suppose the convention was to report the test statistic, then the sampling distribution, + *then* the p-value.
* If that were true, these 2 situations produce different stat blocks:

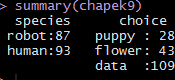
* However, using the condensed reporting standards we get

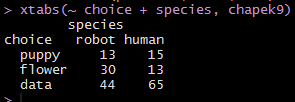
* Now it’s actually unclear which test I actually ran.
* We don’t live in a world in which the contents of a stat block uniquely specifies what tests were ran b/c life is messy.
* Statistics is an intellectual discipline just as much as any other one, + as such it’s a massively distributed, partly-collaborative/partly-competitive project no-one really understands completely.
* Data analysis tools were invented by lots of different people, published as papers in academic journals, implemented, corrected, + modified by lots of other people, + then explained to students in textbooks by someone else.
* As a consequence, a lot of test statistics don’t even have names + as a consequence they’re just given the same name as the corresponding sampling distribution.
* Any test statistic that follows a χ2 distribution is commonly called a **chi-square statistic**, anything that follows a t-distribution is called a **t-statistic** + so on.
* But, as X2 vs. G illustrates, 2 different things w/ the same sampling distribution are still different.
* As a consequence, it’s sometimes a good idea to be clear about *what the actual test you ran was, especially if doing something unusual.*
* If you just say “chi-square test”, it’s not actually clear what test you’re talking about.
* Although, since the 2 most common chi-square tests = goodness of fit test + independence tests, most readers with stats training can probably guess.
* Nevertheless, it’s something to be aware of.

**12.2 - The χ 2 test of independence (or association)**

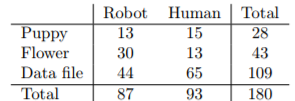
* To determine whether or not visitor is human, guard bots ask whether a visitor prefers puppies, flowers or large, properly formatted data
* But what if humans + robots have the same preferences? That probably wouldn’t be a very good test then, would it?
* Have a single data frame called chapek9, which contains 2 factors, species + choice
* In total = 180 entries in the data frame, 1 for each “person” (both robots + humans) asked
* Specifically, 93 humans, 87 robots + overwhelmingly the preferred choice = data file.



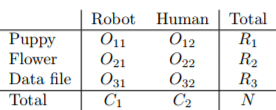
* However, these summaries don’t address the question we’re interested in.
* To do that, we need a *more detailed* description of the data 🡪 want to do is look at the choices *broken down by species* 🡪 need to **cross-tabulate**
* Since data are stored in a data frame, it’s convenient to use **xtabs()**



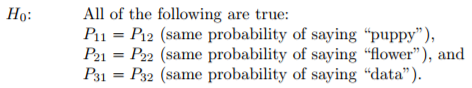
* Contingency table:



* This is a nice way to report the descriptive statistics for this data set.
* Quite clear the vast majority of the humans chose the data file, whereas robots tended to be a lot more *even* in preferences.
* Leaving aside the question of why humans might be more likely to choose the data file for the moment, our 1st order of business = to determine if discrepancy between human choices + robot choices in the data set is statistically significant.
* How do we analyze this data?
* Specifically, since research hypothesis = “humans + robots answer the question in different ways”, how can I construct a test of the null that “humans + robots answer the question the same way”



* O(i, j) = an observed frequency of the # of respondents of species j who gave answer i
* Total number of observations = N + R(i) = the row totals + C(j) = the column totals
* A technical note: this test description pretends column totals are fixed (i.e., researcher intended to survey 87 robots + 93 humans) + row totals are random (i.e. just turned out 28 people chose “puppy”).
* Should technically refer to this situation as a **chi-square test of homogeneity +** reserve the term **chi-square test of independence** for the situation where BOTH row + column totals are *random outcomes* of the experiment.
* Think about the null: If robots + humans respond the same way, it means the probability a robot says “puppy” = the same probability a human says “puppy”, + so on for the other 2 possibilities
* So, if we use P(i,j) = the probability a member of species j gives response i, our null is that:



* Since the null is claiming the true choice probabilities *don’t* depend on species, we can let P(i) refer to this probability (e.g., P1 = the true probability of choosing “puppy”
* In much the same way we did w/ goodness of fit, we need to calculate the expected frequencies.
* For each of the observed counts O(i, j), need to figure out what the null would tell us to expect.
* This time, it’s a little bit trickier.
* If there are a total of C(j) people in species j, + the true probability of *anyone* (regardless of species) choosing option i = P(i), the expected frequency is just:



* This is all very well + good, but we have a problem.
* Unlike in the goodness of fit test, *the null doesn’t actually specify a particular value for P(i) 🡪* It’s something *we have to estimate* from the data
* If 28/180 people selected flowers, a natural estimate for the probability of choosing flowers is 28/180 = ~.16.
* What we’re saying is our estimate for the probability of choosing option i = the row total divided by the total sample size:



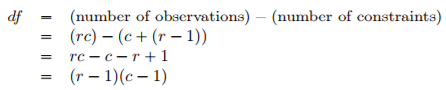
* Therefore, our expected frequency can be written as the product of the row total + column total, divided by the total number of observations:



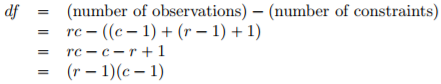
* Now that we’ve figured out how to calculate the expected frequencies, it’s straightforward to define a test statistic following the exact same strategy used in the goodness of fit test (pretty much the same statistic)
* For a contingency table w/ r rows + c columns, the equation that defines our X2 statistic is:



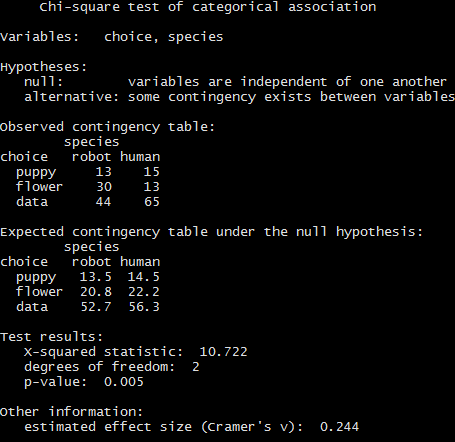
* Only difference = have to include 2 summations to indicate we’re summing over both rows + cols.
* As before, large values of X2 indicate the null provides a poor description of the data, whereas small values of X2 suggest it does a good job of accounting for the data.
* Therefore, like last time, reject the null if X2 is too large + not surprisingly, this statistic is χ 2 distributed
* All we need to do is figure out how many dF are involved, which actually isn’t too hard.
* Can (usually) think of dF as being = # of data points you’re analyzing - # of constraints.
* A contingency table w/ r rows + c columns contains a total of r \* c observed frequencies = total # of observations.
* Constraints are slightly trickier 🡪 The answer is *always the same* 🡪 **dF = (r -1)(p -1)**
* The explanation for why dF takes this value is different depending on the experimental design.
* For the sake of argument, suppose we honestly intended to survey exactly 87 robots + 93 humans (column totals fixed by experimenter), but left row totals free to vary (random variables).
* Think about the constraints that apply here.
* Since we deliberately fixed column totals, we have c constraints right there, but there’s more to it than that.
* Remember our null had some **free parameters** (i.e., we had to estimate the P(i) values) + those matter too
* Every free parameter in the null is like an additional constraint.
* So, how many of those are there?
* Well, since these probabilities have to sum to 1, there’s only r - 1 of these.
* So our total dF is:

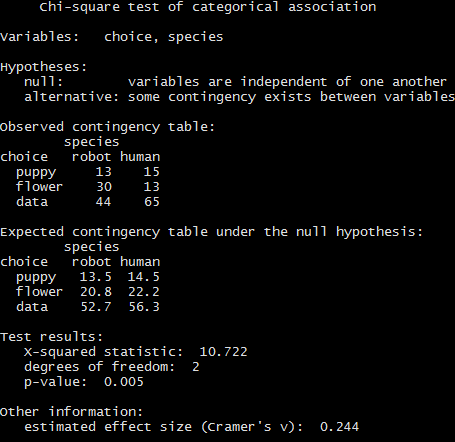


* Alternatively, suppose the only thing the experimenter fixed was total sample size N 🡪 we quizzed the 1st 180 people we saw + it just turned out 87 were robots + 93 were humans.
* This time around our reasoning would be slightly different, but still leads to the same answer.
* Our null still has r - 1 free parameters corresponding to the choice probabilities
* But now also has c - 1 free parameters corresponding to the species probabilities, b/c we’d also have to estimate the probability a randomly sampled person turns out to be a robot.
* Finally, since we actually did fix the total number of observations N, that’s *one more constraint*.
* So now we have r\*c observations, + [ (c – 1) + (r – 1) ] constraints.
* This gives:



* R lets you do it multiple ways 🡪 **chisq.test()** or **associationTest()** fromlsr package
* associationTest() works in the exact same way as xtabs()





* This output gives us enough info to write up the result:
* **Pearson’s χ2 revealed a significant association between species and choice (χ2(2) = 10.7, p = .01): robots appeared to be more likely to say that they prefer flowers, but the humans were more likely to say they prefer data”**
* Notice that, once again, we provide a little bit of interpretation to help readers understand what’s going on w/ the data.
* Later on in a discussion section, we’d provide a bit more context such as:
* **“The fact that humans appeared to have a stronger preference for raw data files than robots is somewhat counterintuitive. However, in context it makes some sense: the civil authority on Chapek 9 has an unfortunate tendency to kill and dissect humans when they are identified. As such it seems most likely that the human participants did not respond honestly to the question, so as to avoid potentially undesirable consequences. This should be considered to be a substantial methodological weakness”**
* This could be classified as a rather extreme example of a reactivity effect
* Obviously, in this case the problem is severe enough that the study is more or less worthless as a tool for understanding the difference preferences among humans and robots.
* However, this illustrates the difference between getting a statistically significant result (null is rejected in favor of the alternative) + finding something of scientific value (the data tell us nothing of interest about our research hypothesis due to a big methodological flaw)
  1. **The continuity correction**
* There’s a tiny change you need to make to your calculations whenever you only have 1 dF, the **continuity correction**, or sometimes the **Yates correction**.
* Remember the χ2 test is based on an approximation, specifically on the assumption that the binomial distribution starts to look like a normal distribution for large values of n.
* 1 problem w/ this is it often doesn’t quite work, especially when you’ve only got 1 dF (e.g., when you’re doing a test of independence on a 2x2 contingency table).
* The main reason for this is that the *true* sampling distribution for the X2 statistic is *actually discrete* (b/c you’re dealing w/ categorical data), but the *χ2 distribution is continuous*.
* This can introduce systematic problems.
* Specifically, when n is small + df = 1, the goodness of fit statistic tends to be “too big”, meaning you actually have a bigger α value than you think (or, equivalently, the p values are a bit too small).
* Yates suggested a simple fix in which you redefine the goodness of fit statistic as:



* Basically we just subtract off 0.5 everywhere
* The correction is basically a hack + is not derived from any principled theory, but rather is based on an examination of behavior of the test + observing that the corrected version seems to work better
* Sometimes you’ll see R (or any other software) introduce this correction, so it’s kind of useful to know what they’re about.
* You’ll know when it happens, because R’s output will explicitly say it has used a continuity or Yates’ correction

**12.4 Effect size**

* Remember it’s becoming commonplace to ask researchers to report some measure of **effect size**.
* Suppose you’ve run a chi-square test that turns out to be significant = you now know there’s some association between your variables (independence test) or some deviation from the specified probabilities (goodness of fit test).
* Now to report a measure of effect size 🡪 given there is an association/deviation, how strong is it?
* There are several different measures one can report, + several different tools one can use to calculate them.
* 2 most commonly reported measures of effect size = the **φ statistic** + a superior version **Cramer’s V**
* Mathematically, they’re very simple.
* To calculate φ, just divide X2 value by the sample size + take the square root



* The idea here is that the φ statistic is supposed to range between 0 (no at all association) + 1 (perfect association), *but it doesn’t always do this when your contingency table is bigger than 2x2,*
* For bigger tables, it’s actually possible to obtain φ > 1, which is unsatisfactory.
* So, to correct for this, people usually prefer to report Cramer’s V, a pretty simple adjustment to φ.
* If you’ve got a contingency table w/ r rows + c columns, then define k = min(r, c) to be the smaller of these 2 values.
* If so, Cramer’s V statistic is:



* This seems to be a fairly popular measure, presumably b/c it’s easy to calculate + gives answers that aren’t completely silly
* V really does range from 0 (no at all association) to 1 (perfect association).
* Calculating V or φ is obviously pretty straightforward, so much so core packages in R don’t have functions to do it, though other packages do.
* **lsr** package 🡪 **cramersV()**takes a contingency table as input + prints out effect size



* However, if using **associationTest()** to do analysis, you won’t actually need to use this at all, b/c it reports the Cramer’s V statistic as part of the output.
* It’s worth mentioning the **assocstats()** function in the **vcd** package.
* A command like **assocstats( chapekFrequencies )** will run the χ2 test as well as the **likelihood ratio test +** report 3 different measures of effect size 🡺 φ, , Cramer’s V , + the **contingency coefficient**

**12.5 Assumptions of the test(s)**

* **All statistical tests make assumptions**, + it’s usually a good idea to check that assumptions are met.
* For the chi-square test, assumptions are:
* Expected frequencies are sufficiently large.
* Remember the χ2 sampling distribution emerges b/c the binomial distribution is pretty similar to a normal distribution
* But, this is only true when the # of observations is sufficiently large.
* In practice, all expected frequencies need to be reasonably big.
* How big is reasonably big? Opinions differ, but the default assumption seems to be you’d like to see all expected frequencies > 5, though for larger tables you’d probably be okay if at least 80% of expected frequencies are > 5 + none are < 1.
* These are rough guidelines, not hard + fast rules, + seem to be somewhat conservative
* Data are independent of one another.
* 1 somewhat hidden assumption of a chi-square test is you have to genuinely believe the observations are independent.
* Suppose we’re interested in proportion of babies born at a particular hospital that are boy + observed 20 girls + only 10 boys.
* Seems like a pretty convincing difference, but later on, it turns out we’d actually walked into the same ward 10 times + only seen 2 girls + 1 boy
* Original 30 observations were massively non-independent + were only in fact equivalent to 3 independent observations.
* This is an extreme example, but it illustrates the basic issue.
* **Nonindependence** messes things up + sometimes causes you to falsely reject the null
* It can go the other way too.
* Consider what would happen if we’d done the cards experiment slightly differently + instead of asking 200 people to try to imagine sampling 1 card at random, we asked 50 people to select 4 cards.
* 1 possibility = everyone selects 1 heart, club, diamond, + spade (the **representativeness heuristic**).
* This is highly non-random behavior from people, but in this case, we’d get an observed frequency of 50 for all 4 suits.
* For this example, the fact that the observations are non-independent (b/c the 4 cards picked will be related to each other) actually leads to the opposite effect = falsely retaining the null.
* If you happen to find yourself in a situation where independence is violated, it may be possible to use the **McNemar test** or the **Cochran test**
* Similarly, if expected cell counts are too small, check out the **Fisher exact test**.

12.6 The most typical way to do chi-square tests in R When discussing how to do a chi-square goodness of fit test (Section 12.1.7) and the chi-square test of independence (Section 12.2.2), I introduced you to two separate functions in the lsr package. We ran our goodness of fit tests using the goodnessOfFitTest() function, and our tests of independence (or association) using the associationTest() function. And both of those functions produced quite - 371 - detailed output, showing you the relevant descriptive statistics, printing out explicit reminders of what the hypotheses are, and so on. When you’re first starting out, it can be very handy to be given this sort of guidance. However, once you start becoming a bit more proficient in statistics and in R it can start to get very tiresome. A real statistician hardly needs to be told what the null and alternative hypotheses for a chi-square test are, and if an advanced R user wants the descriptive statistics to be printed out, they know how to produce them! For this reason, the basic chisq.test() function in R is a lot more terse in its output, and because the mathematics that underpin the goodness of fit test and the test of independence is basically the same in each case, it can run either test depending on what kind of input it is given. First, here’s the goodness of fit test. Suppose you have the frequency table observed that we used earlier, > observed clubs diamonds hearts spades 35 51 64 50 If you want to run the goodness of fit test against the hypothesis that all four suits are equally likely to appear, then all you need to do is input this frequenct table to the chisq.test() function: > chisq.test( x = observed ) Chi-squared test for given probabilities data: observed X-squared = 8.44, df = 3, p-value = 0.03774 Notice that the output is very compressed in comparison to the goodnessOfFitTest() function. It doesn’t bother to give you any descriptive statistics, it doesn’t tell you what null hypothesis is being tested, and so on. And as long as you already understand the test, that’s not a problem. Once you start getting familiar with R and with statistics, you’ll probably find that you prefer this simple output rather than the rather lengthy output that goodnessOfFitTest() produces. Anyway, if you want to change the null hypothesis, it’s exactly the same as before, just specify the probabilities using the p argument. For instance: > chisq.test( x = observed, p = c(.2, .3, .3, .2) ) Chi-squared test for given probabilities data: observed X-squared = 4.7417, df = 3, p-value = 0.1917 Again, these are the same numbers that the goodnessOfFitTest() function reports at the end of the output. It just hasn’t included any of the other details. What about a test of independence? As it turns out, the chisq.test() function is pretty clever.11 If you input a cross-tabulation rather than a simple frequency table, it realises that you’re asking for a test of independence and not a goodness of fit test. Recall that we already have this cross-tabulation stored as the chapekFrequencies variable: > chapekFrequencies species choice robot human puppy 13 15 flower 30 13 data 44 65 11Not really. - 372 - To get the test of independence, all we have to do is feed this frequency table into the chisq.test() function like so: > chisq.test( chapekFrequencies ) Pearson’s Chi-squared test data: chapekFrequencies X-squared = 10.7216, df = 2, p-value = 0.004697 Again, the numbers are the same as last time, it’s just that the output is very terse and doesn’t really explain what’s going on in the rather tedious way that associationTest() does. As before, my intuition is that when you’re just getting started it’s easier to use something like associationTest() because it shows you more detail about what’s going on, but later on you’ll probably find that chisq.test() is more convenient.