Statistics for AI and CS

SUMMARY

by Maaike Lijnzaad

m.f.lijnzaad@student.rug.nl

Date: October 29, 2018

Contents

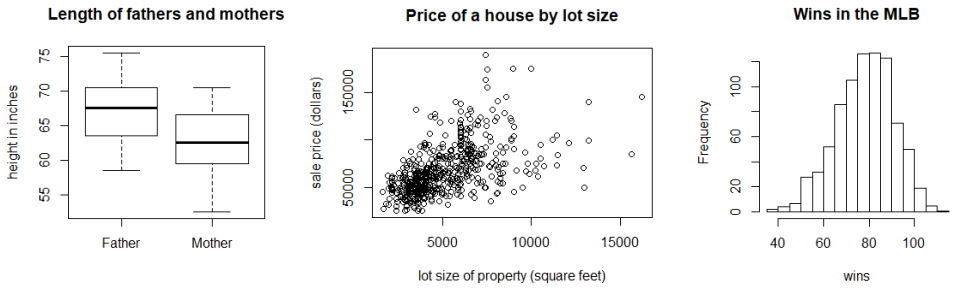
A WORD OF ENCOURAGEMENT

Hi! Thank you for reading my summary. I really hope it helps you understand the material a little bit better. If it doesn't, I encourage you to watch some good YouTube videos on statistics, they really helped me understand the material. Keep your head up, you're going to nail this exam! Good luck :)

1 The basics

1.1 EXPLORATORY DATA ANALYSIS

You use exploratory data analysis to get an insight into the data before performing formal tests. The way you're visualizing your data depends on the type of data (categorical (*factor*) or numerical? how many variables?). Most used are boxplots (Figure 1a), scatter plots (Figure 1b), histograms (Figure 1c), and frequency tables.



(a) Boxplot (b) Scatter plot (c) Histogram

Figure 1: Different plots for exploratory data analysis

They say more than just a measure of central tendency (mean, median, mode). Other things to look out for are the variance (how spread out the data are) and standard deviation (average squared distance from the mean). Some data are skewed: a longer right tail means the data are right-skewed – then the mean and median are very different.

1.2 Parts of a statistical analysis

1. *Introduction.* Give some info on the study and how the data were measure and the context of the research, if you have information on that. First explain what the study aims to discover and then pose the research question.

2. *Exploratory analysis.* Choose a way to visualize the data like mentioned before, and possibly investigate the quality of the data: sample size, skewed or not, et cetera.

3. *Formal analysis.* The main part of your statistical analysis. Choose a statistical test that is relevant to the research question.

4. *Conclusion.* Just answer the research question, basically.

5. *Discussion.* Touch on the critical aspects of the data (is the quality good?), your analysis (do the data meet the assumptions you did?) and your conclusion (how certain are you that this conclusion holds?).

1.3 Hypothesis testing

When you perform a statistical test for your formal analysis, you are always testing a null hypothesis H_0. This H_0 can be something like $\mu_A = \mu_B$ or $\beta_1 = 0$.

Then your alternative hypothesis H_A is $\mu_A \neq \mu_B$ or $\beta_1 \neq 0$, if you're doing a two-sided test. A one-sided test would mean that you're only testing for the alternative that one of the parameters is larger than the other, and not for the opposite. Deciding to do a one-sided test needs good justification, and you need to decide this *before* seeing the data. E.g., you can choose to do a one-sided test if you are only interested in whether A is larger than B and the other way around would not be interesting to the research.

The result of a test is the test statistic (T-value, Z-value, etc.). This test statistic then corresponds to a p-value. If the p-value is smaller than the significance level α (usually 0.05) you can reject the null hypothesis and accept the alternative hypothesis. If the p-value is larger than α, you can't reject H_0 and cannot really prove anything.

2 Confidence intervals

2.1 For a population proportion p (through $\hat{p}$)

Suppose you take a sample of 1,000 Groningen citizens and measure the amount of AI students that are in this group. You have then taken a sample from a bigger population and measured a certain proportion within that sample. You are not sure if this proportion in your sample corresponds with the actual proportion in the real population, and you can never really be since you're not actually testing the entire population. This is where the confidence interval comes in.

If you have a confidence interval of 95% around a proportion, it means that you are 95% sure the actual population proportion falls somewhere in that interval. A confidence interval of 95% is most often used, since it's $1 - \alpha$.

The confidence interval is given by:
$\hat{p} - z^\star \cdot \text{SE}(\hat{p}) < p < \hat{p} + z^\star \cdot \text{SE}(\hat{p})$

Your $\hat{p}$ is given by x/n with x = number of "successes" (in the example, the number of AI students in the sample) and n = sample size. The $z^\star$-value directly corresponds with your confidence interval, e.g. in the case of an interval of 95% it's 1.96. Then lastly the standard error SE of your $\hat{p}$ is given by:
$\text{SE}(\hat{p}) = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

You don't really need to remember any of this, actually. But it's good to understand the inner workings of the confidence interval. All of these calculations are easily done in R with `prop.test(x, n, conf.level = 0.95)`. `x` and `n` are single numerical values here.

2.2 For a difference of proportions

Say you want to do a second measuring of the proportion AI students in Groningen a year later, and you want to know the confidence interval for this difference of proportions. This again is done through the `prop.test` function: `prop.test(x = c(x1, x2), n = c(n1, n2), conf.level = 0.95)`. If the number 0 is not included in the confidence interval, you can conclude that the difference between these population proportions is significant.

2.3 For a population mean μ (through $\bar{x}$)

Suppose you've got 50 grades that students got for an exam. The mean of this sample, $\bar{x}$, is not the same as the mean grade for all of the students. You can, like before, find a confidence interval around this mean. This time it's given by:
$\bar{x} - t^\star \cdot \text{SE}(\bar{x}) < \mu < \bar{x} + t^\star \cdot \text{SE}(\bar{x})$

You'll notice that this is almost the same as the equation for a confidence interval for a proportion. Since the test statistic is t here, you use the following function to calculate the confidence level: `t.test(x, conf.level=0.95)`. `x` is a data vector here, of which the mean will be calculated automatically.

2.4 For a difference of means

To find the confidence interval for a difference of means, use the `t.test` function like before: `t.test(x, y, var.equal=FALSE, conf.level=0.95)`. `x` and `y` are both data vectors. `var.equal=FALSE` is the default setting for this function. Matched samples require `TRUE` for the equal variance. Again, if 0 is not included in the resulting confidence interval, the difference between the population means is significant.

2.5 Significance tests for proportions and means

When you want to know whether a population proportion or mean is significant but do not need to know the confidence interval, you can just use the same `prop.test` and `t.test` as before. This time you are only interested in the resulting p-value, so you don't have to specify a confidence level. What you do have to specify though, is the alternative hypothesis. For a proportion, this H_A can either be $p < p_0$ or $p > p_0$ or $p \neq p_0$, and it's the same for a mean except it's with a μ. This is specified in R with the argument `alternative="less"` or `"greater"` or `"two-sided"`.

Some more information on the T-test: it's only useful if the data are somewhat normally distributed. If they are not, you should do the Wilcoxon Rank-Sum test instead (`wilcox.test(x, y)` with `x` and `y` as data vectors). If you want the confidence intervals for this you should specify it with `conf.int =`

`TRUE`.

3 χ^2-squared tests

3.1 Goodness-of-fit test (single categorical variable)

Sometimes you want to know whether given proportions are statistically different from the sample proportions. For example, if you want to know whether a die is fair, the given proportions are all 1/6. If you then measure the proportions of each rolled number for, say, 100 die rolls, you can test if these proportions are significantly different from 1/6 as follows: `chisq.test(x, p = rep(1/6, times=6))` with `x` being a data vector with the measured proportions.

3.2 Test of independence (multiple categorical variables)

If you want to know whether two variables are independent, you put them together in a frequency table to then perform a χ^2-squared test. This table is obtained through `xtabs(~ var1 + var2)`. For the χ^2-squared test of independence, your H_0 is that the two variables are independent of each other and are not related. The H_A is that the variables are dependent. The output of `chisq.test(table)` function will be a p-value that allows you to decide whether to reject the null hypothesis or not.

4 Linear regression

4.1 Simple linear regression

When you have a set of data that are linearly distributed, you could draw a line through the data and predict some new values. This prediction is called either interpolation (predicting a value within the range of the data set) or extrapolation (predicting a value outside that range, so larger or smaller than the data points). Drawing this "prediction line" through the data is the essence of simple linear regression. The simple linear regression model is given by:
$Y_i = \beta_0 + \beta_1 \cdot x_i$
Here, Y_i is the expected value for the ith observation. It's the response variable, the actual outcome, the dependent variable of this equation. β_0 is the intercept and β_1 is the slope of the model. The x_i's are the independent variables.
The distances from the actual data points to the best-of-fit line (given by the equation above) are called residuals or errors. A good, significant regression model seeks to minimize the sum of squared errors (SSE) – in other words, the residuals need to be small, which is obviously a sign of a model that fits the data well.
Figure 2 shows the plot from Figure 1 with a regression line. The line does not seem to fit the data too well.

Everything that has to do with the linear regression model is done in R through `model <- lm(dependent.variable ~ independent.variable)`. Then `model` just gives you the coefficients β_0 (intercept) and β_1. `summary(model)` gives some more information: especially the adjusted R-squared and p-values are interesting. The adjusted R-squared (between 0 and 1) tells us how well

Figure 2: Scatter plot with regression line

the data fits the model (for reference, the adjusted R-squared for the model in Figure 2 is 0.29, which is not very good). The p-values tell us which variables are significant in predicting the response variable, which is even more relevant for multiple linear regression.

Other relevant functions are `abline(model)`, which creates the regression line in the scatter plot, and `plot(model)`, which shows multiple plots regarding the residuals and other model stuff. They are important if you want to check whether the model assumptions hold in your model.

These assumptions are the following:

1. The relationship between the independent and the dependent variable needs to be linear, i.e. the scatter plot of the dependent variable by the independent variable needs to show some linearity.

2. The variance of the residuals needs to be constant, i.e. the red line in

the Residuals vs Fitted plot (obtained through `plot(model)`) needs to be rather straight.

3. The residuals need to be normally distributed, i.e. the points in the Q-Q plot (`plot(model)`) need to fall on a straight line.

It's always good to touch on these aspects of the model in your statistical analysis.

4.2 Multiple linear regression

The difference between multiple and simple linear regression is that there are more than 1 independent variable for multiple regression, so the model equation looks like this:

$Y_i = \beta_0 + \beta_1 \cdot x_{1i} + \beta_2 \cdot x_{2i} + ... + \beta_n \cdot x_{ni}$

Otherwise it works very similarly to simple linear regression. Use `model <- lm(dependent.variable ~ independent.variable1 + independent.variable2 + ...)`. This is where `summary` gets a lot more interesting, since it tells us which variables are significant in predicting the dependent variable and which are not. Taking insignificant variables out of the model can increase the adjusted R-squared which is a sign that the model fits the data better. Sometimes it's also necessary to add polynomials to the model – that's where you add a quadratic term, then a cubic term, etc. and see which of the models has the highest adjusted R-squared value.

4.3 Model selection

There are two main ways to compare models: by looking at the p-values, like described in the previous section, or using a model selector such as AIC

(Akaike Information Criterion).

Looking at the p-values and removing the ones that are not significant from the model can be done when there are not that many variables, but when the model has a lot of independent variables, using AIC might be easier.

AIC is a score that a model gets, and the lower the score, the better the fit. The `stepAIC` function calculates the AIC scores of the different models and chooses the model that has the lowest score. You can easily see the results of the `stepAIC` function through `summary(stepAIC(model))`. It will show the variables that are included in the selected model and their p-value. It's important to note that the model that is selected may have variables in it that are not significant! That's just because the AIC scores are less strict.

5 Analysis of variance

5.1 ANOVA

We've seen before that we can test the difference between population means through a T-test. However, if you want to test the difference between more than one population mean, a T-test would be way too tedious. You'd have to test for the difference between A and B, between B and C, A and C, etcetera. A test that is perfect for this occasion is an analysis of variance (ANOVA). It works slightly differently than the T-test, but yields the same result. (If you'd do a T-test for the difference in means between just two variables A and B, ANOVA would give you the same result. In that case a T-test is preferred, since it's just easier.)

So, for ANOVA, your H_0 is $\mu_1 = \mu_2 = ... = \mu_n$, and the H_A is very simply $\mu_i \neq \mu_j$ for one or more pairs of i and j. The test statistic that is the result of

ANOVA is F. That test statistic is (as always) related to a p-value, which tells you the significance of the difference in means.

5.2 Performing it

There are multiple ways to get this p-value: through the `oneway.test` function, the `aov` function and even the `lm` function. But first sometimes you need to "massage"[1] the data in order to use those functions.

The data need to be stored in a single data vector. This data vector has the value of a data point (`values`) and a factor that indicates what variable that data point is from (`ind`). The following code will create this data vector: `vector <- stack(list(var1 = var1, var2 = var2, var3 = var3, ...))`. You then use it in the `oneway.test` function like this: `oneway.test(values ~ ind, data = vector)`. The only thing that's really interesting from the output of this function is the p-value. You can also get this through the `aov(values ~ ind, data = vector)` function, but you do need to take the `summary` of that first.

In 2.5, we saw that we need to perform a different kind of test for the difference in means if the data are not normally distributed (or skewed; you can check this through a boxplot or a histogram). This is no different in case of ANOVA. The "ANOVA equivalent" of the Wilcoxon Rank-Sum test is the Kruskal-Wallis test. In R, this test is done through `kruskal.test (values ~ ind)`, which again yields a p-value. You can also use `lm` to perform ANOVA.

You've got a p-value from the ANOVA and know that there is a significant difference – what's next? You could stop there, but of course it's interesting to

[1] I know, I'm sorry, I hate the expression too.

know *which* differences between which means are significant and which are not. The Tukey HSD (Honestly Significant Difference) test can tell you this. Its input is an `aov` model, and its output is a table with the exact differences between means, their lower and upper boundaries and most importantly, the p-value. It basically does what I've described at the start of this chapter: it tests the difference between A and B, between B and C, A and C, etcetera.

5.3 Analysis of covariance: ANCOVA

Let's go back to linear models a little bit: if you've got both numerical and categorical variables as predictors (independent variables) in your model, the significance test you perform is called ANCOVA. To perform it, you just use `lm` like before, except you consider at least one of the variables to be a factor: `factor(var1)`. Remember at least one other variable needs to be numerical to be able to call what you are doing ANCOVA, so you can't have *just* categorical variables. ANCOVA works just the same as a normal linear regression model. It's definitely not the same as ANOVA though, keep that in mind!

5.4 Two-way ANOVA and interaction

Suppose you want to know whether popcorn and good seats make a movie in the cinema more enjoyable. The enjoyment of the movie is expressed in a number and the popcorn and good seats are two categorical variables with values "yes" and "no". The test to see whether the categorical variables are significant for the numerical value of enjoyment is a two-way ANOVA. It's simply performed by `summary(lm(enjoy ~ factor(popcorn) + factor(seats)))`[2].

[2] In this case you don't have to specify the `factor` bit since these variables were factors in the first place. If they are numerical (see `str(your_data)`), you should include the `factor`!

It shows you the p-values for the two factors.

It's also interesting to check whether there is interaction between the two variables, so if there is a difference in mean enjoyment between [good seating and popcorn] and [bad seating and popcorn]. If the seating has an effect on whether the popcorn makes the movie more enjoyable, that is an interaction. In the study that was actually done on the seating and the popcorn, there turned out to be an interaction: popcorn only makes the movie more enjoyable if the seats are comfortable too.

You can check this interaction through an interaction plot (`interaction.plot(factor1, factor2, indep.variable)`). If the lines are parallel, that suggests there is no interaction. In Figure 3, the lines are definitely not parallel, which hints at an interaction.

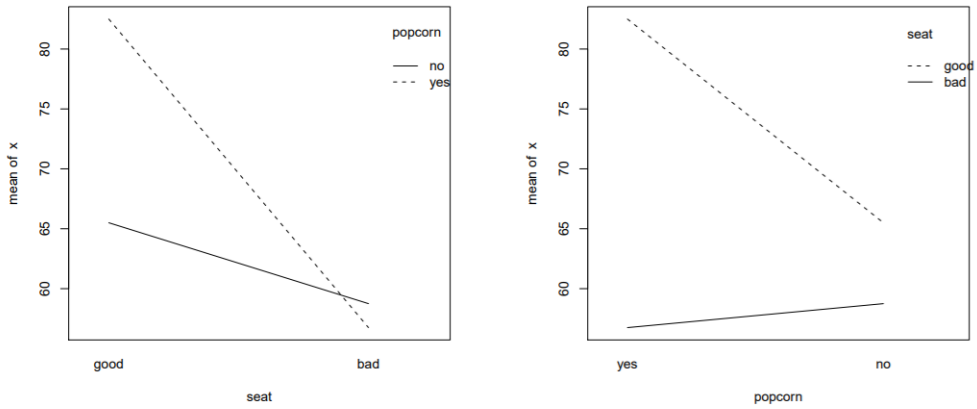


Figure 3: Interaction plot of movie enjoyment by seat quality and popcorn

The formal way to check this interaction is: `summary(lm(enjoy ~ popcorn * seats))`. This will not only give you the p-values for the categorical variables like before, but it will also give you a p-value for the interaction between those variables.

6 Logistic regression

In all of our previous regression models, we have been fitting numerical and/or categorical values to predict a numerical value. But what if you want to predict a categorical value, or rather, the probability of a categorical value? So, say you want to know which factors are an influence on whether someone has lung cancer. Then your response variable is a categorical, binary one: it's either yes or no. You want to know which factors are an influence on the probability of the variable "lung cancer" being "yes". In this case you use (multiple) logistic regression, which is done in R like this: `glm(dependent.categorical.variable ~ var1 + var2 + ..., family = "binomial"`. You can also use AIC model selection on this model, and basically everything else you did with the linear models.