Analysis of Categorical Data (cont.)

NRES 710

Last compiled: 2024-09-26

Exercise 2: no big deal on the graphs, if you struggled with that. This class is focused on running analyses, and not making graphs. But if you are still having a hard time running ‘linear models’ and doing the associated assumption checks, then check-in with me and we can work together to help you.

Reading for next week: Ruxton & Beauchamp 2008, skim this. Focus on learning about what post-hoc tests are.

## Summary

Last class we discussed how to analyze data when we have a categorical X-variable data.

Traditionally, this is called a *t-test* – but I am going to avoid calling it that going forward. Instead, we are always going to be using the **linear model** in this class, which is more *parsimonious* (simple) and will create continuity with the approach we are going to take toward understanding statistics in this class

Today we will discuss how to **present results** from analysis of categorical data. We have a ‘canned’ sentence for presenting results when we have two continuous variables, but we need a slightly different approach for when we have categorical X-data.

The good news is that our X-variable data can only be continuous or categorical! So once we learn this second approach, we will only have to know these **two approaches**. We can use them repeatedly during our various analytical activities, perhaps with slight modifications.

## Presenting Results

With continuous X data, we start out by saying: “For each 1 unit increase in X, we observed a unit change in Y.”

There is no such thing as a unit change in X when we have a categorical X variable! So we won’t say that. Instead, we will say something like:

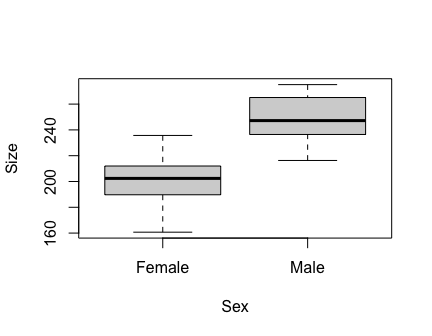
**“We found that [group 1] was [] [Y-units] ([+/-CI]; +/-95% CI) [descriptor word] than [group 2] (p = [p-value]).”**

Don’t need an because it doesn’t really make sense with a categorical X-variable.

* \***if p > 0.05, then add: “; however, our results were not statistically significant.”**

### Example

Let’s take a peak in R at the elephant [seal body size data](lecture_7_seal_data.csv) from last class. We will analyze it using ‘lm()’ with a categorical X-variable:

# Load the data  
datum <- read.csv("lecture\_7\_seal\_data.csv")  
  
# Examine it  
head(datum)  
tail(datum)  
  
# Examine structure of data  
str(datum)

## 'data.frame': 40 obs. of 2 variables:  
## $ Sex : chr "Female" "Female" "Female" "Female" ...  
## $ Size: num 189 195 231 201 203 ...

# Force 'Sex' to be a factor  
datum$Sex <- factor(datum$Sex)  
  
# Scatterplot  
plot(Size ~ Sex, data = datum)

# Fit a linear model  
results <- lm(Size ~ Sex, data = datum)  
summary(results)

##   
## Call:  
## lm(formula = Size ~ Sex, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -42.14 -12.65 -0.43 12.80 32.86   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 202.83 4.04 50.16 < 2e-16 \*\*\*  
## SexMale 46.15 5.72 8.07 9.2e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.1 on 38 degrees of freedom  
## Multiple R-squared: 0.632, Adjusted R-squared: 0.622   
## F-statistic: 65.1 on 1 and 38 DF, p-value: 9.23e-10

Let’s talk about this a bit to make sure we are all on the same page.

The analysis output from linear regression with **categorical X-data** is the same output that we got when we ran linear regression using **two continuous variables**.

* – the Y-intercept; previously this was the value when Y = 0; now it is slightly different, it is still the Y-value when X = 0, but we can more technically define this as the average Y-value for the ‘reference group’.
  + We get to choose what the reference group is. By default, R chose ‘female’ as the reference group, and we can see that because ‘SexMale’ was the effect.
* – the difference in size between males and females. The **p-value** is testing the null hypothesis that there is no difference in size between males and females.
* Residual standard error – noise around the average
* Everything else is still the same

We can now use this output to populate the our ‘presenting resultings’ sentence. We will need the +/- 95% CI:

# Confidence intervals  
confint(results)

## 2.5 % 97.5 %  
## (Intercept) 194.65 211.02  
## SexMale 34.57 57.72

# Calculate confidence intervals  
57.7 - 46.1

## [1] 11.6

# simple way: upper limit minus mean  
confint(results)["SexMale", 2] - results$coefficients["SexMale"]

## SexMale   
## 11.58

# extracting estimates from objects to do math

**“We found that males were 46.1 kg (+/-11.6; +/-95% CI) heavier than females (p = 9.24e-10).”**

You could also be more descriptive and perhaps say: *“had 46.1 kg (…) greater mass than females”*. It’s nice to be descriptive – but also good to keep things concise.

What if you wanted to put females first? That’s fine – just swap things around, word smith it a bit, and put females first: “We found that females had 46.1 kg (…) less mass than males”.

If you give R categorical data, R will automatically create the ‘reference’ group using the alphabetical order of the groups. For example, if you gave R data with two experimental treatments in the X-variable, “control” and “burned”, it will automatically make “burned” the reference group – but you might want to change that to visualize the ‘effect’ of burn treatment compared to control, untreated areas.

**Takehome message:** Running a t-test is the same as running a regression, but with a categorical X-variable – but it doesn’t matter with coding in R. The only difference is you have to understand that you have a categorical X-variable, and write a slightly different sentence.

### Changing the reference group

We can change what the reference group is in R using the ‘factor()’ function. For example, using our seal data:

# Examine the categorical data  
str(datum$Sex)

## Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...

# The 'levels' are the groups in the variable, and they are ordered alphabetically by default.  
  
# Re-order the levels  
datum$Sex <- factor(datum$Sex, levels = c("Male", "Female")) # switch the order  
  
# Re-run model  
results2 <- lm(Size ~ Sex, data = datum)  
summary(results2)

##   
## Call:  
## lm(formula = Size ~ Sex, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -42.14 -12.65 -0.44 12.80 32.86   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 248.98 4.04 61.58 < 2e-16 \*\*\*  
## SexFemale -46.14 5.72 -8.07 9.2e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 18.1 on 38 degrees of freedom  
## Multiple R-squared: 0.632, Adjusted R-squared: 0.622   
## F-statistic: 65.1 on 1 and 38 DF, p-value: 9.23e-10

Now ‘SexFemale’ is the effect.

## Multiple groups

Let’s extend this model a little bit. Previously we had considered a situation where we had a ‘binomial’ X-variable (*bi-* and *nomial* = *two* *names* = a categorical variable with two categories). Let’s now consider a situation where we have a multinomial (*multi-* = more than two) X-variable with more than two categories within it.

**Continuous Y; Multinomial X**

* **Group 1 - Juveniles**
* **Group 2 - Subadults**
* **Group 3 - Adults**
* **Y: mass**

The first thing I want to show you is how this works in the general linear modeling framework. Up to this point, we have used our trusty equation for the linear model as:

We previously introduced the concept of ‘dummy coding’, which will be particularly useful for us to explain how to extend our simple linear model to accommodate multiple groups.

We will take our categorical X-variable and ‘dummy code’ it by creating a **column for each category** that describes whether each observation in our data is **within that category (1) or not (0)**. For example:

| Age Category | Juvenile | Subadult | Adult |
| --- | --- | --- | --- |
| Juvenile | 1 | 0 | 0 |
| Juvenile | 1 | 0 | 0 |
| Juvenile | 1 | 0 | 0 |
| Subadult | 0 | 1 | 0 |
| Subadult | 0 | 1 | 0 |
| Subadult | 0 | 1 | 0 |
| Adult | 0 | 0 | 1 |
| Adult | 0 | 0 | 1 |
| Adult | 0 | 0 | 1 |

When we create our general linear model to accommodate these multiple groups, we now need to have an X for every one of these dummy-coded columns. But, we are going to leave one out.

The reason we are going to leave one out is because we don’t need that extra column to capture how that works. Let me show you how this works:

You might be scratching your head why ‘Juvenile’ wasn’t included. We don’t need to include it because it will be automatically be captured by the intercept!

* Subadults will be captured by the term.
* Adults will be captured by the term.
* So then Juveniles will be all the individuals that weren’t in Subadult or Adults categories and thus will captured by the intercept as the ‘reference’ group.

The reference group is the group by which all others are compared.

Another way to look at this is to examine our dummy-coded columns. If we look at the ‘Subadult’ and ‘Adult’ columns, we actually don’t need the ‘Juvenile’ column to know which observations are of juveniles:

|  | Age Category | Juvenile | Subadult | Adult |
| --- | --- | --- | --- | --- |
| 1 | Juvenile | 1 | 0 | 0 |
| 4 | Subadult | 0 | 1 | 0 |
| 7 | Adult | 0 | 0 | 1 |

The individuals that were 0 for both Subadults and Adults must be Juveniles, by the process of elimination.

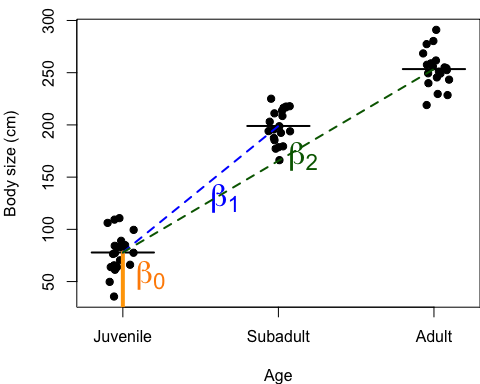
**Let’s see how this works mathematically:**

**Cancel out beta1 and beta2 terms**

The meaning of has not changed – the average Y (size) of our reference group.

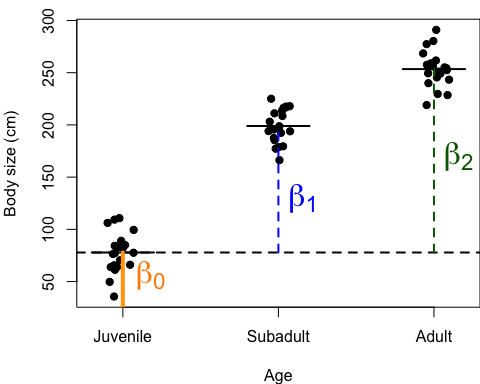
**Cancel out beta2 term**

The meaning of has not changed either – the difference between juveniles and subadults.

**Cancel out beta1 term**

Our new variable, , has a similar meaning to – the difference between adults and juveniles.

I like to visualize these things **graphically**. So I am going to draw something that might help us visualize this on the board:

Note: our **error** is normally distributed with a mean = 0, centered on the average, with a standard deviation of .

* The average size of juveniles is
* The diff between juvs and subads is
* The diff between juvs and adults is

These look like:

**Q:** Does anyone have any questions?

You might be wondering… how would we estimate the difference between subadults and adults?

* tells us about the difference between juveniles and subadults, and whether it’s statistically significant.
* tells us about the difference between juveniles and adults, and whether it’s statistically significant.
* But neither tells us about the difference between juveniles and adults.

**Q:** Anybody know how we might learn about the difference between juveniles and adults? **Changing the reference.**

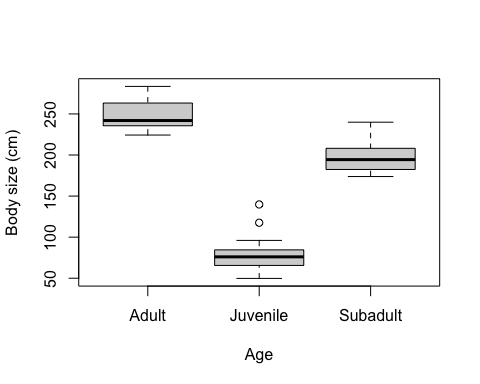
### Comparison of multiple groups in R

Let’s see what this would look like in R. Download some simulated data [here](lecture_8_dataset1.csv).

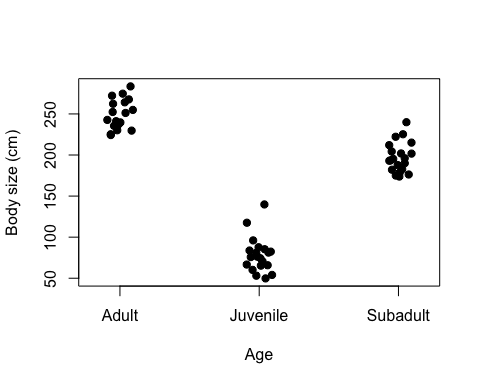
‘Truth’ for these data are:

* Juveniles: 75 kg, so = 75 kg
* Subadults: 200 kg, so = 125 kg
* Adults: 250 kg, so = 175 kg

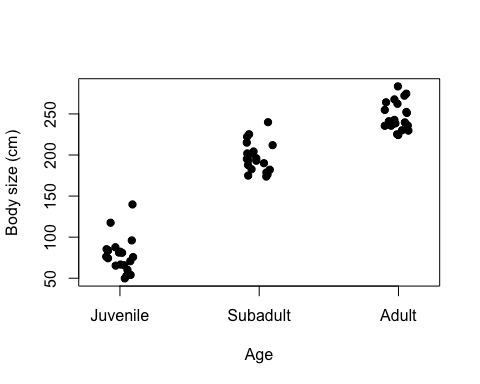
Load in the data:

### Code to test for differences among three groups  
  
# Read in the data  
datum <- read.csv("lecture\_8\_dataset1.csv")  
  
# Examine the data  
head(datum)  
  
# Make datum a factor  
datum$Age <- factor(datum$Age)  
  
# Plot  
plot(Size ~ Age, data = datum, xlab = "Age", ylab = "Body size (cm)")

# Plot w/ jitter  
stripchart(Size ~ Age, data = datum, vertical = TRUE, method = "jitter",  
 pch = 19, xlab = "Age", ylab = "Body size (cm)")



# Re-order categorical X-data  
datum$Age <- factor(datum$Age, levels = c("Juvenile", "Subadult", "Adult"))  
  
# Plot  
stripchart(Size ~ Age, data = datum, vertical = TRUE, method = "jitter",  
 pch = 19, xlab = "Age", ylab = "Body size (cm)")

# Add in dummy-coding  
dummy <- model.matrix(~ Age -1, data = datum)  
colnames(dummy) <- c("Juvenile", "Subadult", "Adult") # rename columns  
# be careful; make sure name order reflects order of categorical variable 'Age'  
datum <- cbind(datum, dummy) # bind to dataframe  
  
# Re-examine data  
head(datum)

Before we use ‘lm()’ to test for these differences in R, let’s quickly look at the ANOVA function:

## Analysis of variance; 'aov()'  
help(aov)  
# Fit an analysis of variance... by calling 'lm()'!!  
  
# Run an ANOVA  
results <- aov(Size ~ Age, data = datum)  
summary(results)

## Df Sum Sq Mean Sq F value Pr(>F)   
## Age 2 303653 151826 413 <2e-16 \*\*\*  
## Residuals 57 20967 368   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

What does it give us…? An ANOVA table!

**Q:** What’s missing here..? Estimates of effects – and whether those individual effects are significant or not. So, this is **not very useful**.

## Multiple comparison using 'lm()'  
# Run analysis  
results2 <- lm(Size ~ Age, data = datum)  
summary(results2)

##   
## Call:  
## lm(formula = Size ~ Age, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.5 -12.7 -3.2 7.6 61.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 78.34 4.29 18.3 <2e-16 \*\*\*  
## AgeSubadult 118.99 6.06 19.6 <2e-16 \*\*\*  
## AgeAdult 169.74 6.06 28.0 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.2 on 57 degrees of freedom  
## Multiple R-squared: 0.935, Adjusted R-squared: 0.933   
## F-statistic: 413 on 2 and 57 DF, p-value: <2e-16

Here are the results, and by now hopefully this is starting to look familiar!

**Q:** What is the reference group? How do we know?

* The intercept is our estimate of mean size of juveniles.
* ‘AgeSubadult’ is the effect of being subadult compared to juveniles.
* ‘AgeAdult’ is the effect of being an adult compared to juveniles.

Notice: these estimates are pretty good estimates of truth! Which is what statistics should do. It’s not perfect because there’s randomness in our data (process error, sampling error, etc.).

And then all of our usual metrics at the bottom.

### Difference between ANOVA and t-test

Let’s review the difference between ANOVA and t-test. We just did an ANOVA, but we could have done a bunch of t-tests. Why not just do a bunch of t-tests to do all of these individual comparisons? Why might I not want to do that?

**Q:** Can anyone remember why we might not want to a bunch of t-tests? Inflating our risk of committing Type I error!

Remember: Type I error is when we reject the null when in fact it is True; what we **most want to avoid**. We want to keep our risk of committing Type I error to be no greater than 0.05.

* When we calculate a single p-value, we have a 5% chance (0.05) of getting things wrong.
* When we do more and more tests, we increase the chance of committing Type I error.

**If we did 3 t-tests**:

* **P(No committing error) = (1 - )** – we take this to the power of the number of tests we will do
* **P(No errors) = (1 - 0.05)^3**
* **P(>1 errors) = 1 - (1 - 0.05)^3**

# Probability of committing no errors in three t-tests  
(1 - 0.05)^3

## [1] 0.8574

# Probability of committing >=1 Type I errors in three t-tests  
1 - (1 - 0.05)^3

## [1] 0.1426

14% is a lot more than 5%! We need to be careful if we start doing multiple tests.

This is why they teach us to use analyses like ANOVA in statistics. It is meant to be one single test that can evaluate multiple comparisons and thus minimizes our risk of committing Type I error.

* Concept: first, run ANOVA, and get a single p-value. That p-value tells you whether **at least two groups are different from another group**; but it does not mean that all groups are different.
* **ANOVA w/ p < 0.05 – at least 2 groups are different**
* This is a single p-value testing the significance of that categorical variable as a whole.
* After getting the significant p-value, you then do a **‘post-hoc’ test** to evaluate those differences. (We’ll talk about this next week.)

The important part is that our ‘lm()’ summary output gives us the single ANOVA p-value, which is at the bottom of the summary.

Another way to get this is to use:

# ANOVA test  
anova(results2)

## Analysis of Variance Table  
##   
## Response: Size  
## Df Sum Sq Mean Sq F value Pr(>F)   
## Age 2 303653 151826 413 <2e-16 \*\*\*  
## Residuals 57 20967 368   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

So now we know that there is at least two groups that are different from each other.

To know **which ones**, we can use a ‘post-hoc test’… or we can just look up to the results of the linear model! The p-values with our effects tell us that:

* Subadults are significantly larger than juveniles
* Adults are also significantly larger than juveniles

And now we also have **estimates**! The effect sizes for each of the values. We get all of this information from a single annalysis and a single model output.

### Reporting results

How do we report the results?

summary(results2)

##   
## Call:  
## lm(formula = Size ~ Age, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.5 -12.7 -3.2 7.6 61.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 78.34 4.29 18.3 <2e-16 \*\*\*  
## AgeSubadult 118.99 6.06 19.6 <2e-16 \*\*\*  
## AgeAdult 169.74 6.06 28.0 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.2 on 57 degrees of freedom  
## Multiple R-squared: 0.935, Adjusted R-squared: 0.933   
## F-statistic: 413 on 2 and 57 DF, p-value: <2e-16

confint(results2)

## 2.5 % 97.5 %  
## (Intercept) 69.75 86.93  
## AgeSubadult 106.85 131.14  
## AgeAdult 157.60 181.89

“We found that subadults were 133.6 kg (+/-13.7; +/-95% CI) heavier than juveniles (p < 2e-16).”

“We found that adults were 176.7 kg (+/-13.7; +/-95% CI) heavier than juveniles (p < 2e-16).”

But, there are three groups, so we will need to write three sentences to report all of those comparisons. To do this, we have to recreate our results object. It’s pretty easy!

We will use the ‘relevel()’ function. To do this, we will replace the whole ‘Sex variable’ with a ‘relevel()’ function call. The relevel() function requires two arguments: the variable you want to change the reference for, and what you want the new reference to be.

# Re-run analysis with different reference  
results3 <- lm(Size ~ relevel(Age, ref = "Subadult"), data = datum)  
summary(results3)

##   
## Call:  
## lm(formula = Size ~ relevel(Age, ref = "Subadult"), data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.5 -12.7 -3.2 7.6 61.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 197.33 4.29 46.01 < 2e-16 \*\*\*  
## relevel(Age, ref = "Subadult")Juvenile -118.99 6.06 -19.62 < 2e-16 \*\*\*  
## relevel(Age, ref = "Subadult")Adult 50.75 6.06 8.37 1.7e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.2 on 57 degrees of freedom  
## Multiple R-squared: 0.935, Adjusted R-squared: 0.933   
## F-statistic: 413 on 2 and 57 DF, p-value: <2e-16

This new summary looks a little uglier… but it gives us the information that we needed.

“We found that adults were 43.2 kg (+/-13.7; +/-95% CI) heavier than subadults (p = 4.53e-08).”

Note: the juvenile-subadult estimate remained the same, but became negative.

**Questions?**

Next week we will learn about ‘post-hoc tests’. But, I am going to encourage us to build models that can evaluate global difference and within-group differences within one single model, such as we have done using ‘lm()’, **for two reasons**.

* First, this is more simple and minimize extra steps of having to do ‘post-hoc tests’. As we get more advanced with our approaches, post-hoc tests will not exist for all of your needs, so building models that can test for differences within the model is the best approach, in my opinion.
* Second, do we really care about p-values? Post-hoc tests are used to generate p-values, but we don’t think p-values are the goal of statistics. We want to measure effects, and the modeling approach we are using here is good at estimating effects.

### Using the dummy-coded variables

Dummy coded variables can be pretty useful!

* You can combine different dummy-coded variables
* And you can avoid the ugly ‘relevel()’ results
* It may be more intuitive

Turns out we can really easily fit these models using the ‘dummy-coded’ variables. Here’s how that works.

# Using the dummy-coded variables to run this analysis  
results4 <- lm(Size ~ Subadult + Adult, data = datum)  
summary(results4)

##   
## Call:  
## lm(formula = Size ~ Subadult + Adult, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.5 -12.7 -3.2 7.6 61.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 78.34 4.29 18.3 <2e-16 \*\*\*  
## Subadult 118.99 6.06 19.6 <2e-16 \*\*\*  
## Adult 169.74 6.06 28.0 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.2 on 57 degrees of freedom  
## Multiple R-squared: 0.935, Adjusted R-squared: 0.933   
## F-statistic: 413 on 2 and 57 DF, p-value: <2e-16

# Changing the reference  
results5 <- lm(Size ~ Juvenile + Adult, data = datum)  
summary(results5)

##   
## Call:  
## lm(formula = Size ~ Juvenile + Adult, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.5 -12.7 -3.2 7.6 61.5   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 197.33 4.29 46.01 < 2e-16 \*\*\*  
## Juvenile -118.99 6.06 -19.62 < 2e-16 \*\*\*  
## Adult 50.75 6.06 8.37 1.7e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.2 on 57 degrees of freedom  
## Multiple R-squared: 0.935, Adjusted R-squared: 0.933   
## F-statistic: 413 on 2 and 57 DF, p-value: <2e-16

We purposefully left-out ‘Juvenile’, and it is forced to the reference.

These results are the same as when we fit the model using the ‘Age’ categorical variable.

I like these results better because the effect description is really accurate: “Effect of being subadult” is more intuitive than the “Effect of being”AgeSubadult”, in my opinion.

What if we specified all three of the dummy-coded variables, would that screw it up?

# Using the dummy-coded variables to run this analysis  
results6 <- lm(Size ~ Subadult + Adult + Juvenile, data = datum)  
summary(results6)

##   
## Call:  
## lm(formula = Size ~ Subadult + Adult + Juvenile, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.5 -12.7 -3.2 7.6 61.5   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 78.34 4.29 18.3 <2e-16 \*\*\*  
## Subadult 118.99 6.06 19.6 <2e-16 \*\*\*  
## Adult 169.74 6.06 28.0 <2e-16 \*\*\*  
## Juvenile NA NA NA NA   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 19.2 on 57 degrees of freedom  
## Multiple R-squared: 0.935, Adjusted R-squared: 0.933   
## F-statistic: 413 on 2 and 57 DF, p-value: <2e-16

Juvenile was forced to be “NA”, and instead Juvenile was forced to be reference because – it was the last one included.

Let’s try a little critical thinking exercise. Consider this model:

# Using the dummy-coded variables to run this analysis  
results7 <- lm(Size ~ Adult, data = datum)  
summary(results7)

##   
## Call:  
## lm(formula = Size ~ Adult, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -88.04 -52.90 -1.68 44.41 102.11   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 137.83 8.37 16.5 < 2e-16 \*\*\*  
## Adult 110.25 14.50 7.6 2.8e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 52.9 on 58 degrees of freedom  
## Multiple R-squared: 0.499, Adjusted R-squared: 0.491   
## F-statistic: 57.8 on 1 and 58 DF, p-value: 2.83e-10

**Q:** What does this model do…?

It compares the size of adults compared to the size of all other observations combined into a single group. Juveniles and subadults are lumped.

“We found that adults were 115.0 kg (…) heavier than juveniles and subadults combined as a group (p = 7.69e-09).”

Here’s another way to do it:

# Using the dummy-coded variables to run this analysis  
results8 <- lm(Size ~ I(Juvenile + Subadult), data = datum)  
# I() is a function to do whatever is inside the parentheses  
summary(results8)

##   
## Call:  
## lm(formula = Size ~ I(Juvenile + Subadult), data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -88.04 -52.90 -1.68 44.41 102.11   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 248.1 11.8 21.0 < 2e-16 \*\*\*  
## I(Juvenile + Subadult) -110.2 14.5 -7.6 2.8e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 52.9 on 58 degrees of freedom  
## Multiple R-squared: 0.499, Adjusted R-squared: 0.491   
## F-statistic: 57.8 on 1 and 58 DF, p-value: 2.83e-10

These are advantages of using ‘dummy-coded’ variables! It gives us *flexibility.*

**Questions?**

## Summary

* It doesn’t matter how many groups your categorical variable has. It’s still ‘lm(Y ~ X)’.
* It will create a number of effects (). The number of s will be the number of groups, , minus 1 (the reference).
* We have a different results sentence now, that we will use for any categorical variables.
* For categorical variables, you will need a sentence for all pairwise comparisons.

## Truth

Code to simulate the three-age data and save it. Just here in case you are interested in how it was made.

############################### 'Truth' ################################   
### Code for simulating data to be analyzed body size data for two sexes  
  
# Simulate the binomial X-variable (sex)  
n <- 60  
x <- c(rep("Juvenile", n/3), rep("Subadult", n/3), rep("Adult", n/3))  
x <- factor(x, levels = c("Juvenile", "Subadult", "Adult"))  
  
# Simulate continuous y-variable data  
y <- ifelse(x == "Juvenile", rnorm(n/3, mean = 75, sd = 20), #juveniles  
 ifelse(x == "Subadult", rnorm(n/3, mean = 200, sd = 20), #subad  
 rnorm(n/2, mean = 250, sd = 20))) #adults  
  
# Create dataframe  
datum <- data.frame(Age = x, Size = y)  
  
# Save these data for future use  
write.csv(datum, "lecture\_8\_dataset1.csv")

[–go to next lecture–](lecture_9.html)