

Stats with Sparrows - 12

Julia Schroeder

12 ANOVA and repeatability

This handout is long, and there are two exercises, with group submissions each. It is recommended to do the first exercise before you begin with the repeatability part. I'll add a note at that bit in the handout to remind you.

Again: housekeeping!

```
rm(list=ls())
setwd("H:/StatsWithSparrows")

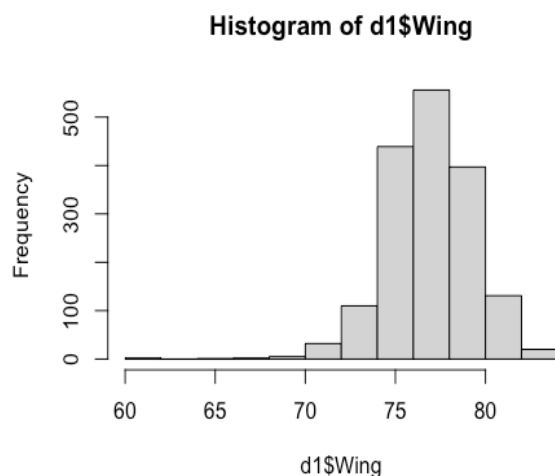
d<-read.table("SparrowSize.txt", header=TRUE)
```

We now examine ANOVA - an analysis of variance that requires a continuous response, and a categorical predictor (explanatory) variable. ANOVAs test for differences in variances, between and within groups (the groups are determined by the levels of the factor). They are a certain kind of linear models. To make things more exciting, we use a new variable: wing length. Let's check it out first:

```
d1<-subset(d, d$Wing!="NA")
summary(d1$Wing)
```

##	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
##	60.0	76.0	77.0	77.4	79.0	84.0

```
hist(d1$Wing)
```

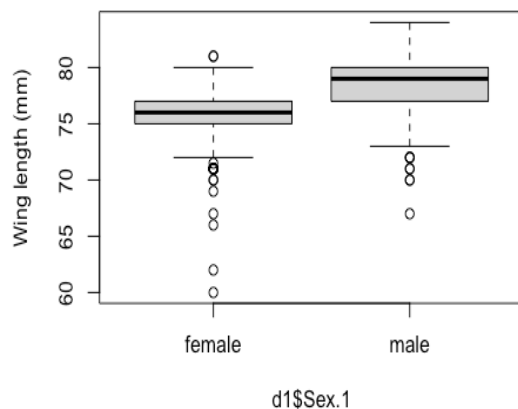


Ok, there are some missing values, and some “outliers”. These likely come from young birds, or from birds that are moulting. We leave them in for now, but keep this in mind for when we interpret our results.

```
model1<-lm(Wing~Sex.1,data=d1)
summary(model1)

##
## Call:
## lm(formula = Wing ~ Sex.1, data = d1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.0961  -1.0961  -0.0961   1.3683   5.3683
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  76.09611    0.07175 1060.50  <2e-16 ***
## Sex.1male     2.53562    0.09998  25.36  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.057 on 1693 degrees of freedom
## Multiple R-squared:  0.2753, Adjusted R-squared:  0.2749
## F-statistic: 643.1 on 1 and 1693 DF, p-value: < 2.2e-16

boxplot(d1$Wing~d1$Sex.1, ylab="Wing length (mm)")
```



Good, we understand most of these output. The F-statistic comes from an ANOVA.

```
anova(model1)

## Analysis of Variance Table
##
## Response: Wing
```

```
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Sex.1      1 2722.0 2721.98  643.15 < 2.2e-16 ***
## Residuals 1693 7165.3    4.23
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

All the sums of squares. Let's have a closer look. There are the sums of squares of the between-group variance (Sex.1), and those for the within-group variances (the residual variance). Then, there are the Mean squares for each of these two (within- and between-groups). The F-value, that is calculated by dividing the mean squares of the between-group estimate by the residual mean squares. Then we ask R to look up the p-value of the F-distribution. Thank you Ronald Fisher!

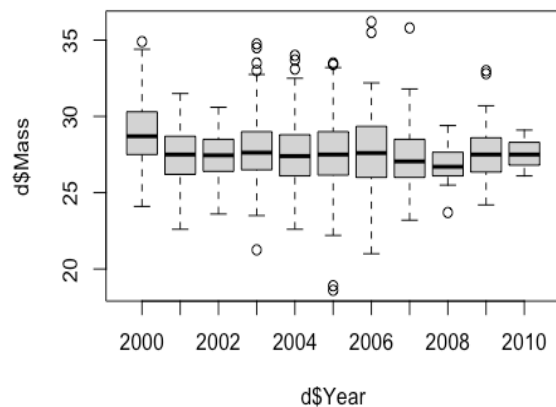
Ok, now we know that wing length differs between the two sexes. But, how much does it differ? To find that out, we actually have to do a t-test. Such a test that one runs after the main analysis, to see which group differs, and how much it differs, from another group, are called "post-hoc" tests.

```
t.test(d1$Wing~d1$Sex.1, var.equal=TRUE)

##
##  Two Sample t-test
##
## data:  d1$Wing by d1$Sex.1
## t = -25.36, df = 1693, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.731727 -2.339518
## sample estimates:
## mean in group female    mean in group male
##           76.09611           78.63173
```

This is ok for two groups. But what if we have lots of groups? Let's see how that works by testing for differences between years:

```
boxplot(d$Mass~d$Year)
```



```
m2<-lm(Mass~as.factor(Year),data=d)
anova(m2)

## Analysis of Variance Table
##
## Response: Mass
##              Df Sum Sq Mean Sq F value    Pr(>F)
## as.factor(Year)  10  340.2   34.020    7.8866 1.721e-12 ***
## Residuals      1693  7303.0     4.314
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

There's many years, and while we could do a summary -

```
summary(m2)

##
## Call:
## lm(formula = Mass ~ as.factor(Year), data = d)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -9.0051 -1.4051 -0.1089  1.2645  8.4874
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    28.8537     0.1375  209.772  < 2e-16 ***
## as.factor(Year)2001  -1.3620     0.2518   -5.410  7.22e-08 ***
## as.factor(Year)2002  -1.3871     0.2903   -4.778  1.92e-06 ***
## as.factor(Year)2003  -1.0596     0.2105   -5.035  5.30e-07 ***
## as.factor(Year)2004  -1.3182     0.1686   -7.816  9.49e-15 ***
## as.factor(Year)2005  -1.2486     0.1695   -7.367  2.71e-13 ***
## as.factor(Year)2006  -1.1411     0.2349   -4.858  1.29e-06 ***
## as.factor(Year)2007  -1.4176     0.2546   -5.568  2.99e-08 ***
```

```
## as.factor(Year)2008 -2.0810      0.6411 -3.246  0.00119 **
## as.factor(Year)2009 -0.9842      0.4544 -2.166  0.03046 *
## as.factor(Year)2010 -1.2871      1.2070 -1.066  0.28642
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.077 on 1693 degrees of freedom
## (66 observations deleted due to missingness)
## Multiple R-squared:  0.04451,    Adjusted R-squared:  0.03887
## F-statistic: 7.887 on 10 and 1693 DF,  p-value: 1.721e-12
```

we'd only get differences between a year and the reference level (2000), but we struggle to understand the differences between other years, say, the difference between 2009 and 2006. That's something we could work out by just adding the year's level to the reference, and comparing these, but it's quite a lot of work. What we really want to do is t-tests between each and every combination to work out what is going on here. That's practically a Tukey's post-hoc test:

```
?TukeyHSD
```

This says it needs a fitted model object of the `aov()` fit.

```
?aov
```

So there's another way to fit an ANOVA. Let's see.

```
am2<-aov(Mass~as.factor(Year),data=d)
summary(am2)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## as.factor(Year)  10      340   34.02    7.887 1.72e-12 ***
## Residuals      1693     7303    4.31
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 66 observations deleted due to missingness
```

This returns the same information than from the `lm()` call.

```
TukeyHSD(am2)
```

```
##    Tukey multiple comparisons of means
##      95% family-wise confidence level
##
## Fit: aov(formula = Mass ~ as.factor(Year), data = d)
##
## $`as.factor(Year)`
##              diff              lwr              upr              p adj
## 2001-2000 -1.36197549 -2.1734639 -0.55048712 0.0000039
## 2002-2000 -1.38706137 -2.3227432 -0.45137957 0.0001019
## 2003-2000 -1.05961042 -1.7379444 -0.38127646 0.0000284
## 2004-2000 -1.31818721 -1.8617515 -0.77462289 0.0000000
## 2005-2000 -1.24861442 -1.7948606 -0.70236828 0.0000000
```

##	2006-2000	-1.14112306	-1.8981612	-0.38408495	0.0000688
##	2007-2000	-1.41755784	-2.2380803	-0.59703533	0.0000016
##	2008-2000	-2.08100080	-4.1474707	-0.01453085	0.0466609
##	2009-2000	-0.98416285	-2.4486968	0.48037105	0.5296791
##	2010-2000	-1.28706115	-5.1772604	2.60313813	0.9929548
##	2002-2001	-0.02508588	-1.0932301	1.04305830	1.0000000
##	2003-2001	0.30236507	-0.5494380	1.15416816	0.9878264
##	2004-2001	0.04378829	-0.7051401	0.79271666	1.0000000
##	2005-2001	0.11336108	-0.6375160	0.86423815	0.9999935
##	2006-2001	0.22085243	-0.6948649	1.13656977	0.9995087
##	2007-2001	-0.05558234	-1.0244418	0.91327715	1.0000000
##	2008-2001	-0.71902530	-2.8487505	1.41069994	0.9917374
##	2009-2001	0.37781264	-1.1746986	1.93032386	0.9994681
##	2010-2001	0.07491434	-3.8492521	3.99908075	1.0000000
##	2003-2002	0.32745095	-0.6434021	1.29830396	0.9918009
##	2004-2002	0.06887416	-0.8131013	0.95084966	1.0000000
##	2005-2002	0.13844695	-0.7451839	1.02207779	0.9999908
##	2006-2002	0.24593831	-0.7814493	1.27332592	0.9995402
##	2007-2002	-0.03049647	-1.1055201	1.04452715	1.0000000
##	2008-2002	-0.69393943	-2.8740121	1.48613320	0.9948248
##	2009-2002	0.40289852	-1.2179896	2.02378659	0.9993590
##	2010-2002	0.10000022	-3.8517170	4.05171745	1.0000000
##	2004-2003	-0.25857679	-0.8606709	0.34351731	0.9523031
##	2005-2003	-0.18900400	-0.7935203	0.41551232	0.9955235
##	2006-2003	-0.08151264	-0.8816138	0.71858850	0.9999999
##	2007-2003	-0.35794741	-1.2183614	0.50246662	0.9615269
##	2008-2003	-1.02139037	-3.1040217	1.06124094	0.8905172
##	2009-2003	0.07544757	-1.4118031	1.56269828	1.0000000
##	2010-2003	-0.22745073	-4.1262589	3.67135750	1.0000000
##	2005-2004	0.06957279	-0.3784955	0.51764107	0.9999915
##	2006-2004	0.17706414	-0.5124915	0.86661983	0.9991483
##	2007-2004	-0.09937063	-0.8580784	0.65933717	0.9999983
##	2008-2004	-0.76281359	-2.8055271	1.27989990	0.9821835
##	2009-2004	0.33402436	-1.0967936	1.76484226	0.9996322
##	2010-2004	0.03112606	-3.8465060	3.90875817	1.0000000
##	2006-2005	0.10749135	-0.5841803	0.79916303	0.9999915
##	2007-2005	-0.16894342	-0.9295749	0.59168802	0.9997647
##	2008-2005	-0.83238638	-2.8758151	1.21104237	0.9667328
##	2009-2005	0.26445157	-1.1673873	1.69629044	0.9999567
##	2010-2005	-0.03844673	-3.9164557	3.83956222	1.0000000
##	2007-2006	-0.27643477	-1.2001674	0.64729790	0.9968732
##	2008-2006	-0.93987773	-3.0494566	1.16970112	0.9393125
##	2009-2006	0.15696021	-1.3677970	1.68171738	0.9999998
##	2010-2006	-0.14593808	-4.0592072	3.76733105	1.0000000
##	2008-2007	-0.66344296	-2.7966268	1.46974090	0.9957111
##	2009-2007	0.43339498	-1.1238574	1.99064734	0.9982981
##	2010-2007	0.13049669	-3.7955479	4.05654123	1.0000000
##	2009-2008	1.09683794	-1.3571539	3.55082975	0.9380454
##	2010-2008	0.79393965	-3.5662056	5.15408490	0.9999621
##	2010-2009	-0.30289830	-4.4120866	3.80628996	1.0000000

And here we get the complete table, with each of the combinations, the difference in body mass between those, the upper and lower confidence intervals, and a p-value that's adjusted for multiple comparisons. This is because - as you've learned earlier, our definition of "statistical significance" means that 1 in 20 tests will come up as a false positive. So, if you run, as here, 55 tests, 55/20 or 2.75 tests on average will wrongly come up as statistically significant. That's problematic, of course. To account for that, we can adjust the p-values by that percentage. That's what this function does automatically. It is a bit difficult to keep the overview, though, with so many levels.

This is a good time to complete exercise 1. After that, continue below.

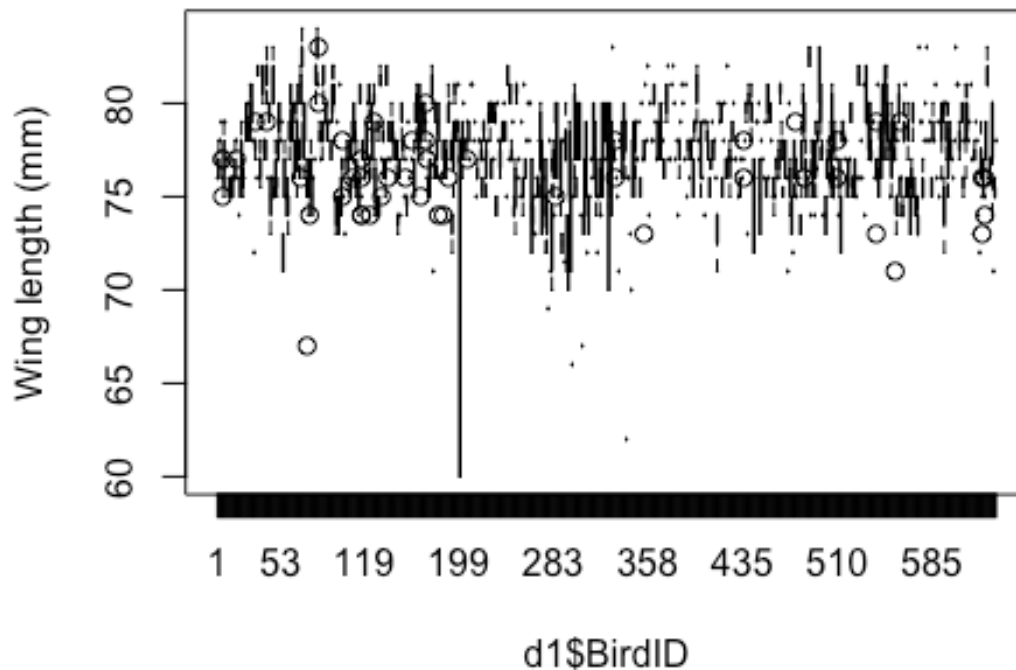
- 1) A researcher wants to know how the magic properties of the plant Aconite change with the environmental temperature the dried flower or leaves are stored in. She collected 100 leaves, and 100 flowers, and stored them in closed containers in a freezer (A), at room temperature (B), and in a heated room (C). She is now interested in the change in their magic properties with respect to their treatment. You are tasked with the statistical analysis. Per group, prepare two separate analyses - one using linear models and their results, and one using ANOVA. Conduct your analyses, then write a methods and results section for this experiments. Make sure to include a descriptive statistics section in addition to the results section. Per group, submit two documents (1 A4 each), both with methods and results.

What if there are many more levels?

What can we do if there's even more levels? Let's use BirdID as factorial for now.

Each sparrow is measured more than once. Now, it is interesting to know if each wing length measurement is the same for the same sparrow everytime it is caught, or whether the wing length changes over time. For instance, one could imagine that sparrows grow longer wing feathers when they get older. We can test that with an ANOVA - it tests, if we remember from earlier, whether the *variation within-groups is smaller than the variation among-groups*. In this case, group is birdID. Note that in english, the term *between* is only used for two groups, not for more, then we use *among* instead.

```
boxplot(d1$Wing~d1$BirdID, ylab="Wing length (mm)")
```



Here, we have loads and loads of groups. A couple hundred groups, actually. This is when an ANOVA can be really helpful. Let's see where this gets us:

That's a lot! of individual birds. Before we even start with any model, we need to get a grip on the data structure. Can we get an idea of how many individual birds there are, and how often they are measured? There are several ways to do this, but here, we'll use dplyr

```
install.packages("dplyr")
require(dplyr)
## Loading required package: dplyr
## Warning: package 'dplyr' was built under R version 4.0.2
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##   filter, lag
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```



```
tbl_df(d1)
```

```
## Warning: `tbl_df()` is deprecated as of dplyr 1.0.0.  
## Please use `tibble::as_tibble()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
## # A tibble: 1,695 x 8  
##   BirdID Year Tarsus Bill Wing Mass Sex Sex.1  
##   <int> <int> <dbl> <dbl> <dbl> <dbl> <int> <chr>  
## 1     1   2002  16.9  NA    76  23.6     0 female  
## 2     2   2001  16.8  NA    76  27.5     1 male  
## 3     2   2002  17.2  NA    76  28.1     1 male  
## 4     2   2003  17.5 13.5    76  27.8     1 male  
## 5     2   2004  17.8 13.4    77  26.5     1 male  
## 6     2   2004  17.7 13.1    78  26     1 male  
## 7     2   2004  17.5 13.3    77  27.7     1 male  
## 8     2   2004  17.8 13.1    77  26.6     1 male  
## 9     2   2004  17.8 13.1    77  26.6     1 male  
## 10    2   2005  17.7 13.3    77  26.6     1 male  
## # ... with 1,685 more rows
```

```
glimpse(d1)
```

```
## Rows: 1,695  
## Columns: 8  
## $ BirdID <int> 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 4, 5, 5, 5, 5, 5, 5, 5,  
## 5,...  
## $ Year <int> 2002, 2001, 2002, 2003, 2004, 2004, 2004, 2004, 2004, 2005,  
## 20...  
## $ Tarsus <dbl> 16.9, 16.8, 17.2, 17.5, 17.8, 17.7, 17.5, 17.8, 17.8, 17.7,  
## 18...  
## $ Bill <dbl> NA, NA, NA, 13.5, 13.4, 13.1, 13.3, 13.1, 13.1, 13.3, 13.2,  
## NA...  
## $ Wing <dbl> 76, 76, 76, 76, 77, 78, 77, 77, 77, 77, 78, 78, 79, 76, 76,  
## 76...  
## $ Mass <dbl> 23.60, 27.50, 28.10, 27.75, 26.50, 26.00, 27.70, 26.60, 26.  
## 60,...  
## $ Sex <int> 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
## 1,...  
## $ Sex.1 <chr> "female", "male", "male", "male", "male", "male", "male", "male", "  
## mal...
```

Well, those are two helpful functions we commit to our memory for future use. Now let's get to the main task: we want to do some serious sub-totalling. What we need to do is group the data. Now, each row has one data entry. That means, some information in the data is repeated. For instance, when a bird is caught more than once, it gets more than one lines. We want to count how often birds were caught only once, twice, thrice ect. To do this, the best thing to do would be first to sort all data by BirdID, the identifier of individual birds. Then, we'd count how many rows each BirdID occupies. Then, we'd group them again

by how many double, triple ect. observations we have, until we have a good summary that fits our needs. That sounds complicated. I'm sure you could do this with your magical skills you learned last week. But we'll try another way:

With dplyr, we can "pipe" objects to another function. That is super helpful when we don't want to write lengthy code with if's and for's:

```
d$Mass %>% cor.test(d$Tarsus, na.rm=TRUE)

##
## Pearson's product-moment correlation
##
## data: . and d$Tarsus
## t = 22.374, df = 1642, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4454229 0.5195637
## sample estimates:
##      cor
## 0.4833596
```

It takes the first object before the percentage sign (here d

Mass) and inserts it as the first argument into the following function. It even gets inserted * before * dTarsus.

Let's see what we can do with this:

```
d1 %>%
  group_by(BirdID) %>%
  summarise (count=length(BirdID))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 618 x 2
##   BirdID count
##   <int> <int>
## 1      1      1
## 2      2     10
## 3      3      1
## 4      4      1
## 5      5      9
## 6      6      4
## 7      7      2
## 8      8      1
## 9      9      1
## 10     10      3
## # ... with 608 more rows
```

Ok - let's slow down. Here, we take our data d, and then group it by BirdID (the identifier for each individual bird). Then we take this data and pass it on to the function summarise.

This is a super helpful function, it makes one row of many rows of data. Here, it takes each group of data on the same BirdID, and summarises it by counting how many lines are in this group.

The output is not very helpful. We see that well, bird 4 shows up 1 time in the dataset, bird 11 10 times, bird 13 only 1 time, and so forth. But it is not really helpful yet because we need to summarise this *again*, this time by count. Let's do that. I have to make a confession first: dplyr actually has a shorter way of doing this. I just didn't tell you right away because I wanted you to understand the group_by and summarise functions first. The shortcut to the above is this:

```
d1 %>%
  group_by(BirdID) %>%
  summarise (count=length(BirdID))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 618 x 2
##   BirdID count
##   <int> <int>
## 1     1     1
## 2     2    10
## 3     3     1
## 4     4     1
## 5     5     9
## 6     6     4
## 7     7     2
## 8     8     1
## 9     9     1
## 10    10     3
## # ... with 608 more rows
```

On to further summarizing:

```
d1 %>%
  group_by(BirdID) %>%
  summarise (count=length(BirdID)) %>%
  count(count)

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 12 x 2
##   count      n
##   <int> <int>
## 1     1   222
## 2     2   147
## 3     3    88
## 4     4    48
## 5     5    47
## 6     6    26
## 7     7    15
```

```
## 8      8      10
## 9      9       6
## 10     10       7
## 11     11       1
## 12     12       1
```

You see, this gives us exactly the information we need. In our dataset of 1695 observations of wing length, 222 birds have been measured only once. So they are not very good data for getting within-group estimates. But 147 have been measured 2, and even more have been measured 3, 4, 5, up to 12 times! That is an amazing dataset of *repeated measures*. Let's run the ANOVA now, and let's not forget to tell R that we use BirdID as a factor!

```
model3<-lm(Wing~as.factor(BirdID), data=d1)
anova(model3)

## Analysis of Variance Table
##
## Response: Wing
##              Df Sum Sq Mean Sq F value    Pr(>F)
## as.factor(BirdID)  617 8147.3  13.2047   8.1734 < 2.2e-16 ***
## Residuals        1077 1740.0   1.6156
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ok. Here we see clearly that there is statistically significantly more variation among groups (BirdID) than within-groups (residuals). The mean squares are much larger among groups than within. So here the difference between an ANOVA and a t.test becomes really clear: a t.test is about parameter estimates, the means, about the difference in size. An ANOVA is about the difference in variance among groups vs within groups. Because, clearly, any post-hoc test here would be still way too many comparisons for us to even consider interpreting. So instead of looking at means, we look at the variation within and among groups - that's a different way of describing data - instead of looking at centrality, we now look at spread. That's the important concept here to get clear with.

Now, when we look at the F statistic - 8 means the among variance divided by the within variance results in 8, that means, the amount of variance explained by differences among Birds (levels in the factorial variable) is 8 times more than the residual variance (the variance that is not explained by differences among birds). That means, individual birds are more likely to have a more similar wing length on re-measuring, than if you compare among birds. This makes sense - if you measure the same individual twice, the value will be more similar than if you measure two different individuals.

What else can we do with this sort of analysis?

We will examine the whole ANOVA bits again, from a bit of a different perspective. We will look at *repeatability*.

Another way of saying what we found here would be that individual bird have consistent wing length. They differ less between multiple measures than they differ from each other.

Another way of saying that is saying bird's wing length is *repeatable*. The statistical term is called *repeatability*, or r , or "intraclass correlation coefficient", and it is important in several ways. It can be used to describe biological things, as we did here. It can however, also, be use to assess the quality of a method - to test for individual observer repeatability, as we did in the lecture.

The repeatability can be calculated in different ways. The simplest way is using the SS and MS of an ANOVA. A very good biologist and statistician, Kate Lessells, published a paper on this, it is called "Unrepeatable repeatabilities: a common mistake". This paper was rejected in many journals, and ultimately was published in a bird journal, "The Auk". It was published in 1987, and since has been cited about 3000 times. It has been very successful in teaching generations of biologists how to do the correct statistics. I strongly urge you to read it, maybe best today, as now your knowledge on ANOVA is fresh, and you will understand it much easier!

However, for now, I will also explain how to calculate the repeatability. It is given as

$$r = \frac{s_A^2}{(s_W^2 + s_A^2)}$$

Where s_A is the among-group variance and s_W is the within-group variance. So what we really do here is we calculate the ratio of variance of the total variance ($s_W^2 + s_A^2$) that is explained by among-group differences. Among over total - that's the difference to F - which is the ratio of among variance explained over residual variance. And total variance explained is among + residual. Now, what this does, really is taking out the F statistic, and replacing it with something that is directly, biologically, interpretable.

We look at the % of variance explained by among-group differences. How can we do that? Remember how the ANOVA really only describes the variance partitions. We only used the sums of squares because the denominator canceled out. However, remember how calculating SS of among-groups was complicated, because we had to weigh for the sample size of each group? So far we only had balanced datasets, where each group had the same number of observations. So what can we do if that differs, and how does that influence the ANOVA results, and the resulting ratios?

That's what Kates paper is about - this group sample size is not as easy as one might think. It's easy for balanced group sizes. But if they are not balanced, you're in trouble. You can read this up in the paper, but for the record:

$$s_W^2 = MS_W$$

and

$$s_A^2 = \frac{MS_A - MS_W}{n_0}$$

The paper points out that many people used to just assume that we could just ignore n_0 . That is wrong, because the sums of squares squares are not the variance unless we account properly for sample sizes of the groups. So we won't make this mistake. We know where

the MS in the function come from, but what exactly is n_0 then? It is most often, actually not the sample size. It would be the sample size n if all group sizes were equal, say, we'd have 10 birds observed once, 10 twice, 10 thrice ect. That would be called a balanced dataset. However, much to our distress, in ecology and evolution, most datasets are far from balanced. Our datasets are often “dirty” and heterogeneous. As we saw earlier, we don't have balanced sample sizes for our BirdID ANOVA here. So we need to calculate n_0 . It is calculated by a complicated function:

$$n_0 = \left[\frac{1}{a-1} \left[\sum_{i=1}^a n_i - \left(\frac{\sum_{i=1}^a n_i^2}{\sum_{i=1}^a n_i} \right) \right] \right]$$

This looks horribly complicated. But, as most things in life, it is not. Check this out: a is the number of groups. n_i is the sample size in each group. In our BirdID example, it's 618. 618 individual birds, and n_1 would be 1, n_2 is 10, and so forth. We have just done this when we explored the data structure:

```
d1 %>%
  group_by(BirdID) %>%
  summarise (count=length(BirdID))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 618 x 2
##   BirdID count
##   <int> <int>
## 1      1      1
## 2      2     10
## 3      3      1
## 4      4      1
## 5      5      9
## 6      6      4
## 7      7      2
## 8      8      1
## 9      9      1
## 10     10      3
## # ... with 608 more rows
```

We could also find it out a in a more elegant way, using this code:

```
d1 %>%
  group_by(BirdID) %>%
  summarise (count=length(BirdID)) %>%
  count(count)

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 12 x 2
##   count      n
##   <int> <int>
## 1      1    222
```

```
## 2      2    147
## 3      3     88
## 4      4     48
## 5      5     47
## 6      6     26
## 7      7     15
## 8      8     10
## 9      9      6
## 10     10      7
## 11     11      1
## 12     12      1
```

Cool. So we know what a and n_i are. Next, we are told to do lots of sums. First, let's look at the inner most bracket, the fracture. We sum the square values of each group's n , and divide this sum by the unsquared sum of the n s of each group. That's easy enough, don't you think? Dplyr to the rescue:

```
d1 %>%
  group_by(BirdID) %>%
  summarise (count=length(BirdID)) %>%
  summarise (sum(count))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 1 x 1
##   `sum(count)`
##       <int>
## 1         1695
```

Uhh. That's the denominator = 1695. The numerator is similar, but we square the count (n) of each group first:

```
d1 %>%
  group_by(BirdID) %>%
  summarise (count=length(BirdID)) %>%
  summarise (sum(count^2))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 1 x 1
##   `sum(count^2)`
##       <dbl>
## 1          7307
```

Cool. That's 7307. Ok, R, this is easy, the fraction is:

```
7307/1695
```

```
## [1] 4.310914
```

Good. What's next? We're supposed to subtract this fraction from the denominator. Clever, huh, we don't have to calculate that monster again. We certainly can do that:

```
1695-7307/1695
```

```
## [1] 1690.689
```

Even better. The last part is easy. a is 618, so it's really 1 over 617. We can do that:

```
(1/617)*(1695-7307/1695)
```

```
## [1] 2.740177
```

And that's our n_0 , excellent. It is actually pretty close to 3, which you can see as a sort of a centrality measure of how many observations we have for each group. Because the differences are so extreme, we can't use a mean or so, we've got to use this monster of n_0 . But since you've become so handy with sums of squares, you probably have some understanding where this comes from, and what it corrects for.

Let's finally calculate the repeatability! I already forgot the equation - I have really bad memory, I can only remember we needed to divide the part of variation that's explained by among group differences by the total variation, corrected for n_0 . Here is the equation again:

$$r = \frac{s_A^2}{(s_W^2 + s_A^2)}$$

with

$$s_W^2 = MS_W$$

and

$$s_A^2 = \frac{MS_A - MS_W}{n_0}$$

and (monster)

$$n_0 = \left[\frac{1}{a-1} \left[\sum_{i=1}^a n_i - \left(\frac{\sum_{i=1}^a n_i^2}{\sum_{i=1}^a n_i} \right) \right] \right]$$

which is 2.74.

Here is out ANOVA result:

```
model3<-lm(Wing~as.factor(BirdID), data=d1)
anova(model3)
```

```
## Analysis of Variance Table
```

```
##
```

```
## Response: Wing
```

```
##
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
## as.factor(BirdID)	617	8147.3	13.2047	8.1734	< 2.2e-16 ***
## Residuals	1077	1740.0	1.6156		


```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

So, really, the repeatability is:

```
((13.20-1.62)/2.74)/((1.62+((13.20-1.62)/2.74))
## [1] 0.7229006
```

The result of r is a fraction. We can express this as a percentage. Now we know that 72% of the variation in wing length is explained by among-individual differences. That equally means that individuals are relatively consistent in their wing length - most of the variation comes from differences among individuals, and only 100-72 = 18% of residual variance remains - within birds. Very cool. The concept of repeatability is an important one, and you should remember this for the rest of your life, as everything, really in this course. Over the time you spend here in Silwood, you will come across other methods of calculating the repeatability, but the principles stay the same: we look at the ratio of variance explained among vs between groups.

Excercises:

- 1) (you should have done this one already! But I'll repeat it here – don't do it a second time)

A researcher wants to know how the magic properties of the plant Aconite change with the environmental temperature the dried flower or leaves are stored in. She collected 100 leaves, and 100 flowers, and stored them in closed containers in a freezer (A), at room temperature (B), and in a heated room (C). She is now interested in the change in their magic properties with respect to their treatment. You are tasked with the statistical analysis. Per group, prepare two separate analyses - one using linear models and their results, and one using ANOVA. Conduct your analyses, then write a methods and results section for this experiment. Make sure to include a descriptive statistics section in addition to the results section. Per group, submit two documents (1 A4 each), both with methods and results.

- 2) We often use repeatability to assess how accurately an we can measure a trait. In that case, an reseracher measures the same trait twice or more often, on a number of individuals. Then we can calculate the repeatability of these measures. This is important to understand how much “noise” or randomness one can expect in ones data. You will find some data from someone conducting such an analysis on their measurements of fly morphometrics. The dataset¹ “Wylde_single.mounted.txt” consists of 22 morphological trait measurements in the neriid fly *Telostylinus angusticollis*, each of which has been measured twice. The fly ID is denoted in the column “ID”. There is also a column “Sex” - this is important because certain traits are only measurable in females, and others only in males. For now, we ignore the columns that denote “Condition”, and “Larval diet” - we can come back to these later on. In this exercise, you will calculate the repeatability for each of the 22 traits, and report them in a table. If you can, produce a code that prevents you from re-typing the same code - can you write a short function that loops through it? Submit one table with the

repeatabilities (in word or pdf) per group, make the table look professional, like in a paper.

¹This data is from a paper on a preprint server, and courtesy of the author, Zac Wylde. If you want to know what these measurements are, check it out: DOI: 10.1101/2020.05.07.083428