Practical 2: Correlation, regression, t-test and ANOVA

BIOM4025 - Statistical Modelling

Erik Postma - e.postma@exeter.ac.uk

Weeks 3 and 4: 7-18 October 2024

Contents

1 Introduction

There is a lot of variation among people in the pitch of their voice, measured by the fundamental frequency (F_0). But how does this variation arise? Earlier studies have found that F_0 remains relatively constant after puberty, and that up to 64% of the variance in pitch in adulthood F_0 is explained by voice pitch at age 7. This suggests that inter-individual differences in F_0 arise very early in life, and maybe even before birth.

In this study, the authors build on these results and test if the F_0 of pre-verbal 4-month old babies' cries predicts the F_0 of their speech as verbal pre-pubertal 5-year old children.

> **HOMEWORK**
> To familiarise yourself with the topic, have a look at this paper: Levrero *et al.*, 2018. The pitch of babies' cries predicts their voice pitch at age 5. *Biology Letters* **14**: 20180065. http://dx.doi.org/10.1098/rsbl.2018.0065.

Many journals, including *Biology Letters*, have made it compulsory to make the data on which an article is based publicly available. This is good news for us, as it allows us to re-analyse their data and compare our results to those presented in the paper. The data can be downloaded here or from the ELE page.

> **NOTE**
> Only have a look at the optional **HARD EXERCISES** once you are finished with all the others, and don't worry if you can't do all of them: They are *hard*!

2 Getting started

2.1 Importing the data

Copy the *Excel* file containing the data into a folder on your computer and have a look at it using *Excel*. Have a look at the column names. Although they are very informative, you will have problems getting `R` to read them correctly. You will therefore have to change the column names to something `R` likes (so no spaces and special symbols, and remember that `R` is case-sensitive). Also check for any missing values. Because `R` cannot import *Excel* files, we will save our data as a `Tab delimited Text (.txt)` file.

Before you can import this file into `R`, you will have to tell `R` where to find the file using `setwd()`:

```
setwd("/This/Is/Path/To/My/Working/Directory")
cry.data <- read.table("rsbl20180065_si_001.txt", header=TRUE,
                       stringsAsFactors = FALSE)
```

2.2 Checking the data

This is a pretty small data set, so to see if our data have imported correctly, we can simply type `cry.data`:

```
cry.data
```

```
   subject sex baby.age baby.weight baby.f0 child.age child.f0 child.weight
1        1   1      108         7.3     384      2034      246         23.0
2        2   1      103         6.7     429      1535      292         15.0
3        3   2      124         7.1     437      2072      280         26.0
4        4   1      118         6.2     412      1601      257         16.5
5        5   2      109         5.6     388      1574      278         15.0
6        6   1      100         5.3     533      1558      312         15.8
7        7   1      153         7.0     447      2154      270         21.0
8        8   1      131         6.3     435      2128      271         20.0
9        9   1      121         6.7     399      2112      260         20.0
10      10   2      130         5.5     543      1628      305         14.0
11      11   2       91         4.3     431      2058      273         19.0
12      12   1       99         5.5     395      1961      216         15.0
13      13   2      107         5.6     393      1700      254         19.0
14      14   1      132         5.0     442      1711      250         17.0
15      15   2       83         4.5     503      2106      254         20.1
   child.height right.ratio left.ratio
1         123.0        0.87       0.96
2         102.0        0.96       1.02
3         120.0        1.02       0.98
4         104.0        0.98       0.89
5         105.0        0.95       0.88
6          96.5        1.00       0.96
7         130.0        0.92       0.92
8         116.0        0.95       0.90
9         112.0        0.94       0.90
10        102.0        0.99       0.91
11        114.0        0.95       0.95
12        103.0        0.93       0.89
13        111.0        0.90       0.92
14        105.0        0.93       0.89
15        114.0        0.94       0.94
```

To open the data in a separate tab, we can also run:

```
View(cry.data)
```

We can also get the column names:

```
names(cry.data)
```

```
 [1] "subject"      "sex"          "baby.age"     "baby.weight"  "baby.f0"
 [6] "child.age"    "child.f0"     "child.weight" "child.height" "right.ratio"
[11] "left.ratio"
```

or some summary statistics:

```
summary(cry.data)
```

```
    subject          sex         baby.age      baby.weight       baby.f0
 Min.   : 1.0   Min.   :1.0   Min.   : 83.0   Min.   :4.300   Min.   :384.0
 1st Qu.: 4.5   1st Qu.:1.0   1st Qu.:101.5   1st Qu.:5.400   1st Qu.:397.0
 Median : 8.0   Median :1.0   Median :109.0   Median :5.600   Median :431.0
 Mean   : 8.0   Mean   :1.4   Mean   :113.9   Mean   :5.907   Mean   :438.1
 3rd Qu.:11.5   3rd Qu.:2.0   3rd Qu.:127.0   3rd Qu.:6.700   3rd Qu.:444.5
 Max.   :15.0   Max.   :2.0   Max.   :153.0   Max.   :7.300   Max.   :543.0
   child.age       child.f0      child.weight    child.height
 Min.   :1535   Min.   :216.0   Min.   :14.00   Min.   : 96.5
 1st Qu.:1614   1st Qu.:254.0   1st Qu.:15.40   1st Qu.:103.5
 Median :1961   Median :270.0   Median :19.00   Median :111.0
 Mean   :1862   Mean   :267.9   Mean   :18.43   Mean   :110.5
 3rd Qu.:2089   3rd Qu.:279.0   3rd Qu.:20.05   3rd Qu.:115.0
 Max.   :2154   Max.   :312.0   Max.   :26.00   Max.   :130.0
  right.ratio       left.ratio
 Min.   :0.8700   Min.   :0.8800
 1st Qu.:0.9300   1st Qu.:0.8950
 Median :0.9500   Median :0.9200
 Mean   :0.9487   Mean   :0.9273
 3rd Qu.:0.9700   3rd Qu.:0.9550
 Max.   :1.0200   Max.   :1.0200
```

We can now have a closer look at the column that contains the gender of the participants (here `sex`) using `summary()`:

```
summary(cry.data$sex)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    1.0     1.0     1.0     1.4     2.0     2.0
```

Note that we need to specify the name of the data frame that contains the variable of interest (`cry.data`) and the name of the variable (`sex`), separated by a `$`. Maybe somewhat surprisingly, this provides us with a mean value for `sex`. We can confirm this using `mean()`:

```
mean(cry.data$sex)
```

```
[1] 1.4
```

We can confirm that currently `R` treats sex as a numeric variable, and not as a character variable or a factor, using `is.numeric()`, `is.character()` and `is.factor()`:

```
is.numeric(cry.data$sex)
```

```
[1] TRUE
```

```
is.character(cry.data$sex)
```

```
[1] FALSE
```

```
is.factor(cry.data$sex)
```

```
[1] FALSE
```

EXERCISE 2.1
Use `as.character()` to coerce sex into a character variable called `sex.as.character` and add this as a new column to `cry.data`. How does this alter the output provided by `summary()`? And if you turn it into a factor? What do you think is the most appropriate type for this variable?

3 Descriptive statistics

Before we delve into the statistical analyses, let's first have a careful look at our data set.

Although in this case we could probably just count the number of rows to get our sample size, we can also let `R` do the counting:

```
nrow(cry.data)
```

```
[1] 15
```

We can calculate the mean, variance and standard deviation of, for example, `child.age`:

```
mean(cry.data$child.age)
```

```
[1] 1862.133
```

```
var(cry.data$child.age)
```

```
[1] 61011.98
```

```
sd(cry.data$child.age)
```

```
[1] 247.006
```

And if we wanted to, we can also get the range by getting the minimum and maximum age:

```
min(cry.data$child.age)
```

```
[1] 1535
```

```
max(cry.data$child.age)
```

```
[1] 2154
```

EXERCISE 2.2
Confirm that the standard deviation is equal to the square-root of the variance.

Although `base-R` doesn't contain a function to calculate the standard error of the mean, this is easy to calculate ourselves because we know that it is equal to the standard deviation divided by the square root of the sample size:

```
sd(cry.data$child.age)/sqrt(length(cry.data$child.age))
```

```
[1] 63.77668
```

EXERCISE 2.3
Compare your estimates to those reported by the authors in Section 2a. Are they the same?

EXERCISE 2.4 (HARD)
Write your own function to calculate the variance.

Often we want to calculate a mean and standard error for a subset of our data. For example, what is the mean F_0 of male and female babies? We can do this by selecting only the rows for boys or girls. Remember that in this dataset, boys are coded as `1` and girls as `2`.

```
mean(cry.data$baby.f0[cry.data$sex==1])
```

```
[1] 430.6667
```

```
mean(cry.data$baby.f0[cry.data$sex.as.character=="1"])
```

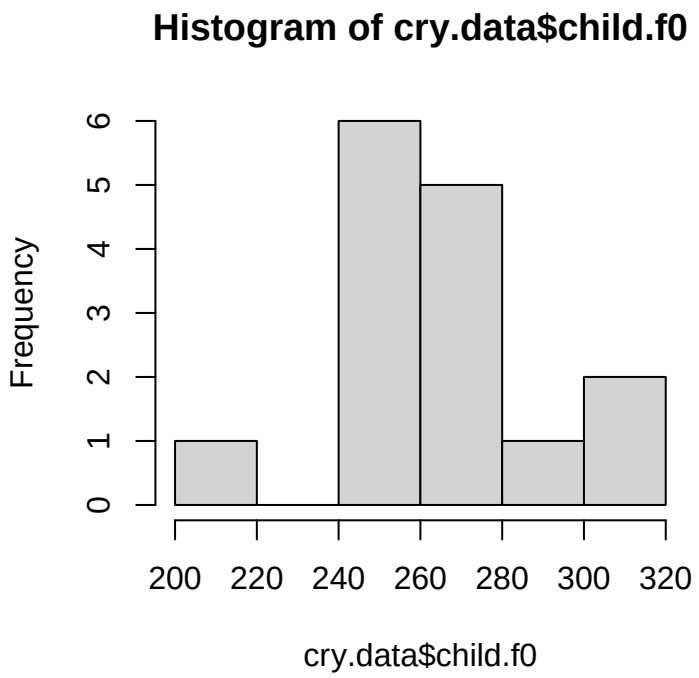
```
[1] 430.6667
```

EXERCISE 2.5
Calculate the mean and standard error for F_0 of female children, and compare this to the values reported in Table 1.

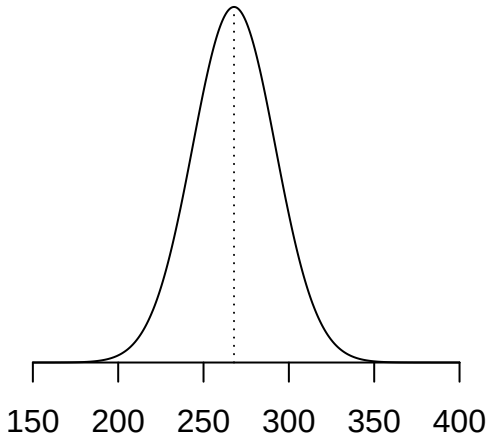
4 Probabilities

Let's have a closer look at the variation in F_0 among children: To visualise this variation, we can plot a histogram:

```
hist(cry.data$child.f0)
```



This distribution looks reasonably normal, so based on our sample we can expect the *true* distribution of f_0 to look something like this:

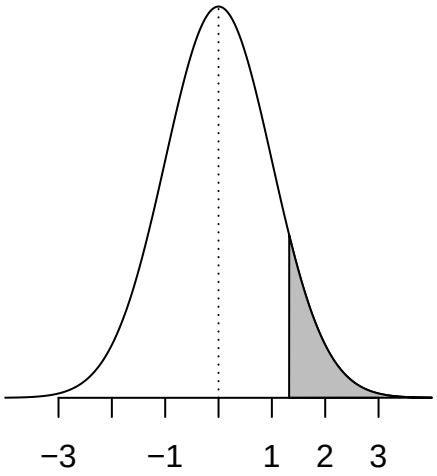


Now that we know the mean and standard deviation of f_0 (mean $\pm$ s.d.=267.87 $\pm$ 24.28 Hz), we can calculate the probability that a (randomly selected) child has an f_0 of at least 300 Hz. To this end we calculate the difference between 300 Hz and the population mean, and express this difference in standard deviations:

```
d <- (300-mean(cry.data$child.f0))/sd(cry.data$child.f0)
d
```

```
[1] 1.323571
```

So in other words, a child with a pitch of 300 Hz has a pitch that is 1.32 standard deviations higher than that of the of the 'average' child. What is the probability of sampling a child that has a pitch that is *at least* this high? To answer this question, we can use the standard normal distribution and calculate the area under the curve that is located to the right of 1.32, *i.e.* the shaded area in the figure below:



We can calculate this area using the `pnorm()` function, but because this function by default gives us the area to the *left* of 1.32, we have to do:

```
1-pnorm(d)
```

```
[1] 0.0928228
```

In other words, the probability that the voice of a randomly selected child has a pitch that is at least 300 Hz is 0.0928228.

EXERCISE 2.6

What is the probability that a randomly selected child has a voice with a pitch between 250 and 300 Hz?

`pnorm()` is part of a set of very useful functions, all related to the normal distribution. In the previous practical you have encountered `rnorm()`, which allows you to sample from a normal distribution. As you have seen above, `pnorm()` provides you with the area under the curve of a standard normal distribution that is to the left of a certain value of `x` (measured in standard deviations). Furthermore, there is `dnorm()` which provides you with the probability density for a given value of `x`. This function allows you to plot curves such as those above. Finally, there is `qnorm()`, which does the opposite of `pnorm()` in that it allows you to specify the area under the curve (*i.e.* a probability) and it will return the value of `x` that goes with this.

TIP
Similar functions exist for other distributions, including the t distribution (`dt()`, `pt()`, `qt()` and `rt()`) and the F-distribution (`df()`, `pf()`, *etc.*).

EXERCISE 2.7 (HARD) Calculate the range of f_0 that is expected to contain the f_0 of 75% of all children.

5 Differences between boys and girls, and between babies and children

Although there are obvious differences in the pitch of male and female voices at adulthood, do these differences already exist in babies and children? To test if there is a significant difference between male and female babies, we can use `t.test()` to perform a two-sample t-test:

```
t.test(cry.data$baby.f0 ~ cry.data$sex, var.equal=TRUE)
```

```
    Two Sample t-test

data:  cry.data$baby.f0 by cry.data$sex
t = -0.67875, df = 13, p-value = 0.5092
alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
95 percent confidence interval:
 -77.38267  40.38267
sample estimates:
mean in group 1 mean in group 2
       430.6667        449.1667
```

How do these results compare to those reported by the authors in Table 1?

Here, `baby.f0` is our dependent variable (we are trying to explain variation in f_0), and `sex` is the predictor or independent variable. In other words, we are modelling `baby.f0` *as a function of* `sex`. The output provided by `t.test()` provides us with a t-value, degrees of freedom and a p-value. Furthermore, it provides a 95% confidence interval for the difference between the sexes. Neither of these provide strong evidence for there being a difference between male and female babies and we therefore do not reject our null-hypothesis.

EXERCISE 2.8
Above we have made the assumption that the variances in both groups are equal (`var.equal=TRUE`). Is this assumption justified? If we don't want to make this assumption, we can specify `var.equal=FALSE` instead. How does this alter our results?

Alternatively, we could have tested for a difference in pitch between male and female babies using an analysis of variance (ANOVA) using `aov()`. To this end we first have to use `aov()` to create a new object containing the results (which I have called `my.anova`), and then ask for a summary of `my.anova` using `summary()`:

```r
my.anova <- aov(cry.data$baby.f0 ~ cry.data$sex)
summary(my.anova)
```

```
             Df Sum Sq Mean Sq F value Pr(>F)
cry.data$sex  1   1232    1232   0.461  0.509
Residuals    13  34767    2674
```

EXERCISE 2.9
Compare the output from this ANOVA to the output from the t-test. What are the similarities? And what are the differences?

In the case above, our dependent variable was found in one column, and the explanatory variable was in another. But what if we want to compare the voice pitch of babies and children, which are in two different columns? This is done in a very similar manner:

```r
t.test(cry.data$baby.f0, cry.data$child.f0, var.equal = TRUE)
```

```
    Two Sample t-test

data:  cry.data$baby.f0 and cry.data$child.f0
t = 11.725, df = 28, p-value = 2.57e-12
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 140.4651 199.9349
sample estimates:
mean of x mean of y
 438.0667  267.8667
```

If we were to report this result in a paper, we could write something like: *The mean pitch of babies (438 Hz) is significantly higher than the mean pitch of children at age 5 (268 Hz) ($t = 11.7$, d.f. $= 28$, $p < 0.001$).*

This two-sample t-test ignores the fact that `baby.f0` and `child.f0` were measured for the same individuals. To take advantage of this *paired* design, we can use a paired t-test instead:

```r
t.test(cry.data$baby.f0, cry.data$child.f0, paired = TRUE)
```

```
    Paired t-test

data:  cry.data$baby.f0 and cry.data$child.f0
t = 16.521, df = 14, p-value = 1.409e-10
alternative hypothesis: true mean difference is not equal to 0
95 percent confidence interval:
 148.104 192.296
sample estimates:
mean difference
          170.2
```

EXERCISE 2.10
How did the degrees of freedom change compared to the two-sample t-test? Why?

6 Is there a relationship between pitch of babies and children?

Having established that there is no significant difference in pitch between baby boys and girls, but a highly significant difference between babies and children, we are ready to tackle our main question: Does pitch as a baby predict pitch as a child?

To test if `baby.f0` and `child.f0` are correlated, we can use `cor.test()`:

```
cor.test(cry.data$baby.f0, cry.data$child.f0)
```

```
    Pearson's product-moment correlation

data:  cry.data$baby.f0 and cry.data$child.f0
t = 2.9805, df = 13, p-value = 0.01063
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.1853709 0.8665661
sample estimates:
      cor
0.6371325
```

This reveals a significant correlation between `baby.f0` and `child.f0` (r=0.637, t=2.98, d.f.=13, p=0.011).

Although this tells us that there is an association between `baby.f0` and `child.f0`, we would also like to know how exactly `child.f0` depends on `baby.f0`. To this end, we can regress `child.f0` (the dependent, or y variable) against `baby.f0` (the independent, predictor, or x variable) with `lm()`. Similar to the ANOVA we did above, we first create an object containing the output, and then ask for a `summary()` of the results:

```
my.regression <- lm(child.f0 ~ baby.f0, data=cry.data)
summary(my.regression)
```

```
Call:
lm(formula = child.f0 ~ baby.f0, data = cry.data)

Residuals:
    Min      1Q  Median      3Q     Max
-38.730  -4.145   4.050   9.874  26.899

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 134.2386    45.1140   2.976   0.0107 *
baby.f0       0.3050     0.1023   2.980   0.0106 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.42 on 13 degrees of freedom
Multiple R-squared:  0.4059,    Adjusted R-squared:  0.3602
F-statistic: 8.883 on 1 and 13 DF,  p-value: 0.01063
```

Again, there is a significant relationship between the pitch of a baby's cry and the pitch of its voice as a child ($b \pm s.e.$ = 0.305 Hz^{-1} $\pm$ 0.102, t=2.98, d.f.=13, p = 0.0106). Alternatively, we could report the results from the ANOVA: $F_{1,13} = 8.88$, p = 0.0106.

EXERCISE 2.11
How much of the variation `child.f0` is explained by variation in `baby.f0`? How does this number compare to that reported by the authors?

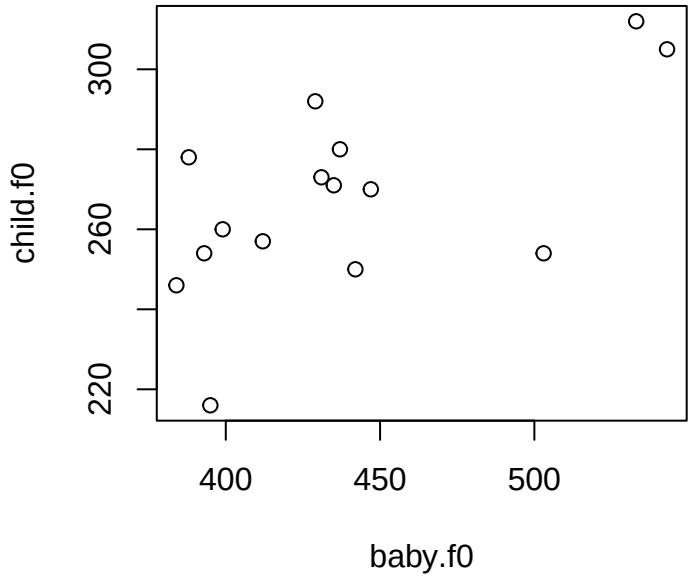
EXERCISE 2.12 (HARD) Rather than using `cor.test()` and `lm()`, use `var()` and `cov()` to calculate the correlation coefficient and slope.

7 Visualising a regression

Finally, it would be nice to visualise the relationship between `child.f0` and `baby.f0`. For now we will not try to plot the data for boys and girls separately, like they do in Figure 1 in the paper. How to do this, we will cover in later practicals.

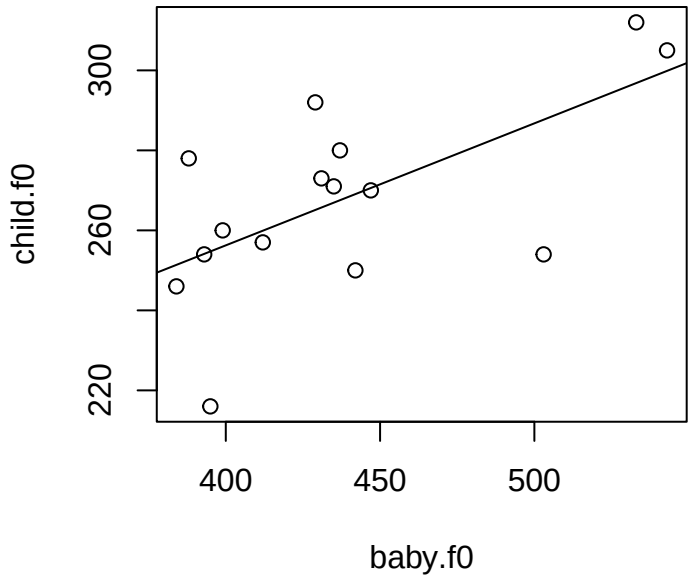
As we have seen last week, we can create a simple plot using `plot()`:

```
plot(child.f0 ~ baby.f0, data=cry.data)
```



Although this plots the data points, we would also like to add a regression line. This we can do with `abline()`, which uses the estimate of the intercept and slope stored in `my.regression`:

```
plot(child.f0 ~ baby.f0, data=cry.data)
abline(my.regression)
```



EXERCISE 2.13 Although this figure doesn't look too bad, it still needs a bit of work to make it ready for publication. Use `xlim`, `ylim`, `pch`, `col`, `xlab` and `ylab` to make it look more like Figure 1.