Statistical methods for linguistic research: Foundational Ideas

Audrey Bürki and Shravan Vasishth

Universität Potsdam
buerki@uni-potsdam.de

August 23, 2017

Summary and Introduction

1. We learnt about the principles behind statistical inference
2. Statistical inference is used in the context of statistical models, i.e., mathematical formulations of the hypothesized data generation process

Summary and Introduction

1. We learnt about the principles behind statistical inference
2. Statistical inference is used in the context of statistical models, i.e., mathematical formulations of the hypothesized data generation process
3. Many statistical models to choose from
4. Choice of model depends on:
   - Data (e.g.,type of variables, distribution)
   - Design (e.g., within or between)
   - Research question
   - Hypotheses about the process(es) underlying the generation of the data

Summary and Introduction

1. We saw two of these models yesterday: one sample t-test and paired t-test.
   - Useful when DV is continuous
   - Useful to compare two means or the mean of a sample with a reference value
   - Requires datapoints to be independent

Summary and Introduction

1. We saw two of these models yesterday: one sample t-test and paired t-test.
   - Useful when DV is continuous
   - Useful to compare two means or the mean of a sample with a reference value
   - Requires datapoints to be independent
2. This morning we will look at another (but not completely unrelated) statistical model : linear regression
3. We will then extend this model to linear mixed-effects models

Linear regression: context and assumptions

1. Linear regression is useful when:
   - We want to model a continuous variable (e.g., RT, acoustic duration)
   - We hypothesize that this continuous variable is influenced by one or a combination of several other variables (categorical or continuous)
   - The relationship between the DV and the other variables is linear

Linear regression: context and assumptions

1. Linear regression is useful when:
   - We want to model a continuous variable (e.g., RT, acoustic duration)
   - We hypothesize that this continuous variable is influenced by one or a combination of several other variables (categorical or continuous)
   - The relationship between the DV and the other variables is linear
2. Linear regression can be used when following assumptions are met:
   - Datapoints are independent
   - Errors are normally distributed (more on this later)

A very simple example: Stress and symptoms

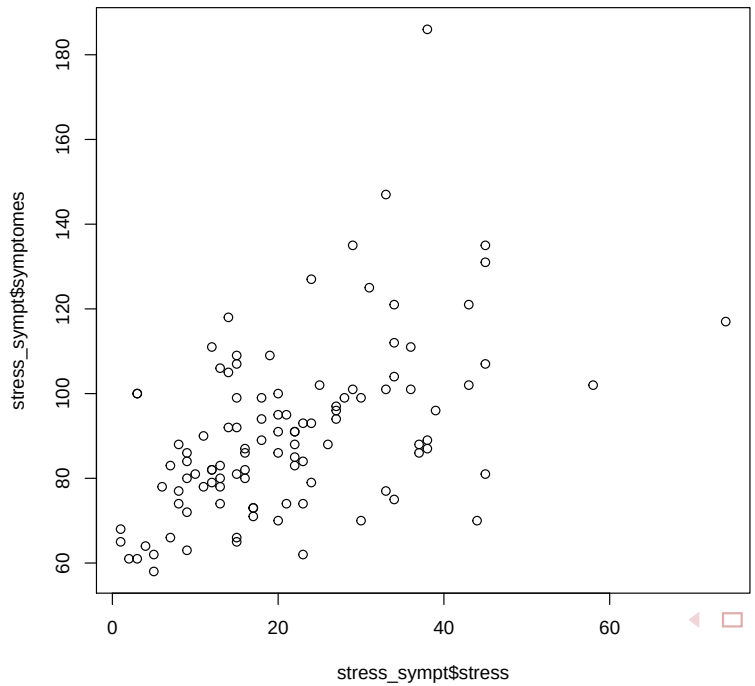
A researcher asked whether the level of stress influences the number of psychological symptoms (data from Howell, 2011)
Students were asked about their level of stress and psychological symptoms
The data: one value per variable for each student

```
stress_sympt<-read.table("data/StressSymptomes.txt",
                         header=TRUE)
head(stress_sympt, n=5)

##   sujet stress symptomes
## 1    s1     30        99
## 2    s2     27        94
## 3    s3      9        80
## 4    s4     20        70
## 5    s5      3       100
```

Plot the data

Always useful to start with visual representation of the data (how are the variables distributed? outliers? is there a relationship between the variables? is the relationship linear?)

```
plot(stress_sympt$symptomes~stress_sympt$stress)
```



(ask R to) Run the linear model

Amounts to finding the straight line that best fits the data
For this, we have to estimate (R does it for us)

1. the intercept
2. the slope

(ask R to) Run the linear model

Amounts to finding the straight line that best fits the data
For this, we have to estimate (R does it for us)

1. the intercept
2. the slope

```
m=lm(symptomes~stress, data=stress_sympt)
```

Check assumptions

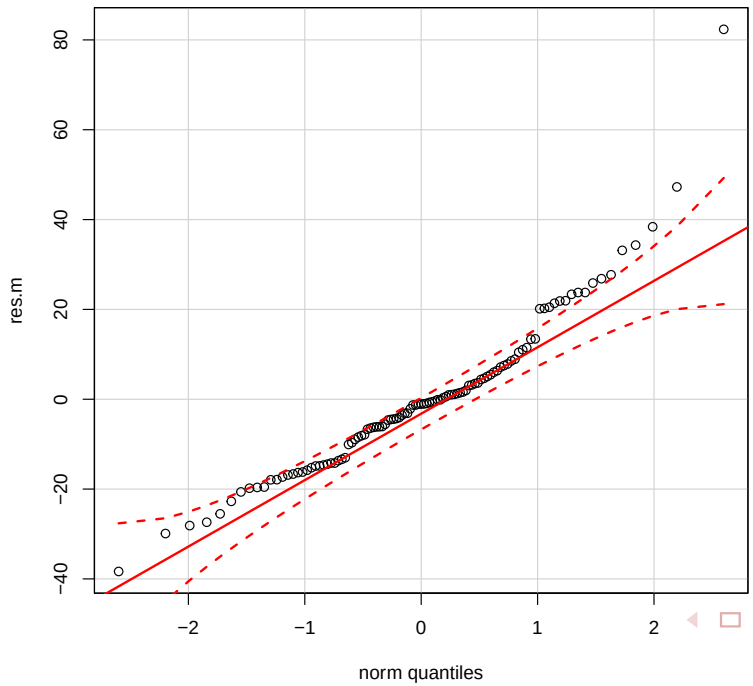
Before we look at the results, we check whether assumptions are satisfied. An assumption of the linear model is that the errors are (approximately) normally distributed We check this by looking at whether the residuals are (approximately) normally distributed: Store the residuals:

```
## residuals:
res.m<-residuals(m)
```

Check assumptions

Plot the residuals by comparing them to a normal distribution (qq plot):

```
library(car)
qqPlot(res.m)
```



Results

R estimated the intercept and slope of the line that best accounts for the data.

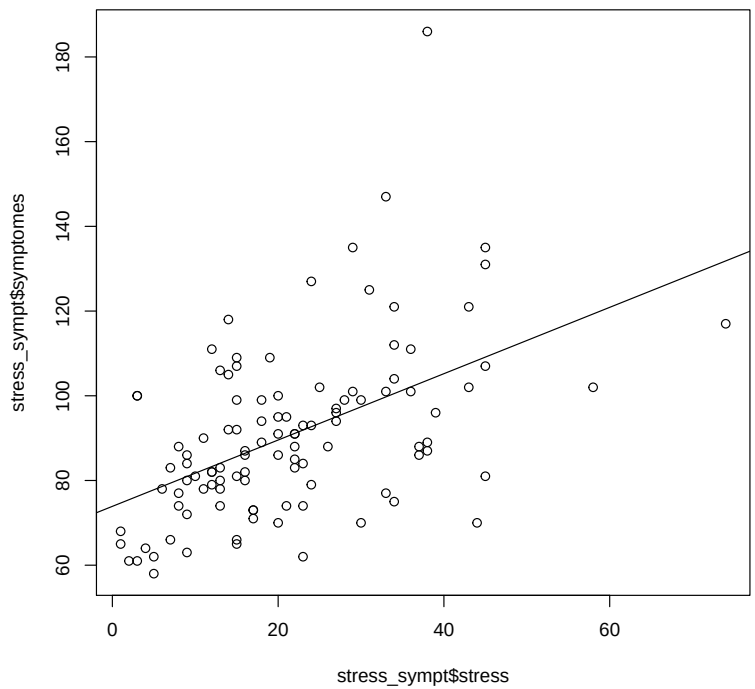
```
coef(m)

## (Intercept)      stress
##  73.8895885   0.7831145
```

- Intercept is: value of y when x = 0 (here, stress = 0)
- Slope is: how much more symptoms for each additional unit of stress level

Results

We can also plot the regression line

```
plot(stress_sympt$symptomes~stress_sympt$stress)
abline(m)
```



Linear models

Next, we will look at the matrix formulation of the linear model. Underlyingly, we have a design matrix or model matrix, which is being used by R to estimate the coefficients:

```
head(model.matrix(m),n=7)

##   (Intercept) stress
## 1           1     30
## 2           1     27
## 3           1      9
## 4           1     20
## 5           1      3
## 6           1     15
## 7           1      5
```

Linear models

Our linear model equation for the regression above is a system of equations. The single equation:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{1}$$

Linear models

Our linear model equation for the regression above is a system of equations. The single equation:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \tag{1}$$

can be expanded to:

$$\begin{array}{rcccccc} Y_1 & = & \beta_0 & + & X_1\beta_1 & + & \epsilon_1 \\ Y_2 & = & \beta_0 & + & X_2\beta_1 & + & \epsilon_2 \\ Y_3 & = & \beta_0 & + & X_3\beta_1 & + & \epsilon_3 \\ Y_4 & = & \beta_0 & + & X_4\beta_1 & + & \epsilon_4 \\ \vdots & & \vdots & & \vdots & & \vdots \\ Y_n & = & \beta_0 & + & X_n\beta_1 & + & \epsilon_n \end{array} \tag{2}$$

Linear models

And this system of linear equations can be restated in compact matrix form:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon \tag{3}$$

where

Vector of responses:

$$\mathbf{Y} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ \vdots \\ Y_n \end{pmatrix} \tag{4}$$

Linear models

The design matrix (in R this is called the model matrix):

$$\mathbf{X} = \begin{pmatrix} 1 & 30 \\ 1 & 27 \\ 1 & 9 \\ \vdots & \vdots \\ 1 & 104 \\ 1 & 86 \\ 1 & 97 \end{pmatrix} \tag{5}$$

Linear models

Vector of parameters to be estimated:

$$\beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \tag{6}$$

and
Vector of error terms (residuals):

$$\epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \vdots \\ \epsilon_n \end{pmatrix} \tag{7}$$

Linear models

We could write the whole equation as:

$$\begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ \vdots \\ Y_n \end{pmatrix} = \begin{pmatrix} 1 & X_1 \\ 1 & X_2 \\ 1 & X_3 \\ 1 & X_4 \\ \vdots & \vdots \\ 1 & X_n \end{pmatrix} \times \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \vdots \\ \epsilon_n \end{pmatrix} \tag{8}$$

Linear models

1. Our main goal when we fit a linear model is to find estimates of the parameters β_0 and β_1, the intercept and slope respectively.
2. We will call the estimates $\hat{\beta}_0$ and $\hat{\beta}_1$.

Linear models

This can be done by "solving" for the beta's. X is the model matrix, and X′ is the transpose of the model matrix. Y is the vector of dependent variables.

$$\beta = (X'X)^{-1}X'Y \tag{9}$$

You do not need to know how the above equation comes about; all you need to know is that given the design matrix X, and the data Y, we can estimate the parameters.
In case you are interested in more detail, see the derivation here: https://github.com/vasishth/LM

Maxwell and Delaney dataset

The same can be done with categorical predictors. And of course, we can have more than one predictor

1. I will illustrate this with a simplified data set from Maxwell and Delaney's book (Designing Expts and Analyzing Data).
2. This is **within-subjects** (fake) experimental data; it has a 2×2 design.
3. Within-subjects designs have the property that each subject sees every condition in the experiment.
4. Contrast this to **between**-subjects designs, e.g., medical trials with treatment vs placebo conditions.

Maxwell and Delaney dataset

Imagine that subjects are shown a stimulus (a picture) on the screen, and it's either shown with no noise (distortion, say) in the background, or with noise; in addition, the stimulus was either horizontal (tilted 0 degrees), or tilted by 4 degrees.
The experiment has two levels of noise, and two levels of tilt. Each participant has done the experiment on several trials per condition. But we need to aggregate because the data must be independent if we want to use a linear regression model.

Load the data

```
noisedeg<-read.table("data/noisedeg.txt",header=TRUE)
head(noisedeg)

##    rt subj deg    noise
## 1 420   s1   0 no.noise
## 2 420   s2   0 no.noise
## 3 480   s3   0 no.noise
## 4 420   s4   0 no.noise
## 5 540   s5   0 no.noise
## 6 360   s6   0 no.noise
```

Means

Let's compute the means by factor levels:

```
means<-with(noisedeg,tapply(rt,IND=list(noise,deg),
                            mean))
means

##             0   4
## no.noise 462 510
## noise    492 660
```

Paired t-test on the data

Let's say we are interested in the factor noise only. Is noise slowing down the participants? We can do a paired t-test to answer this question.
Here is how one can do a t-test with such data, to compare means across (sets of) conditions:

```
no.noise<-subset(noisedeg,noise=="no.noise")
no.noise.means<-with(no.noise,tapply(rt,subj,mean))

noise<-subset(noisedeg,noise=="noise")
noise.means<-with(noise,tapply(rt,subj,mean))
```

Paired t-test on the data

```
no.noise.means

##  s1 s10  s2  s3  s4  s5  s6  s7  s8  s9
## 420 450 450 480 480 600 390 480 540 570

noise.means

##  s1 s10  s2  s3  s4  s5  s6  s7  s8  s9
## 540 600 420 720 630 570 420 630 630 600
```

Paired t-test on the data

These are the means we are comparing with the t-test:

```
## means of noise levels:
with(noisedeg,tapply(rt,noise,mean))

## no.noise    noise
##      486      576

## mean of no.noise=486
## mean of noise=486+90=576
```

Paired t-test on the data

```
t.test(noise.means,no.noise.means,paired=TRUE)

##
##  Paired t-test
##
## data:  noise.means and no.noise.means
## t = 3.2225, df = 9, p-value = 0.01045
## alternative hypothesis: true difference in means is not
## 95 percent confidence interval:
##   26.82139 153.17861
## sample estimates:
## mean of the differences
##                      90
```

Paired t-test on the data

Note that the order in which we compare the two vectors does not matter:

```
t.test(noise.means,no.noise.means,
       paired=TRUE)$statistic

##         t
## 3.222517

t.test(no.noise.means,noise.means,
       paired=TRUE)$statistic

##          t
## -3.222517
```

All that changes is the sign of the t-value.

Linear models

Now we can also run a **linear model**, which evaluates rt as a function of noise (Note: this model is incorrect for the present dataset).
Important note: When given names for factor levels, R will change them into a dummy variable (0 to one level, 1 to the other). This is the default in R (we will see in a minute why it is useful to do it this way). This is called treatment coding: one level is the baseline to which the other level is compared.We can anticipate and recode the variable ourselves. We can then choose wich level is the reference level.

Linear models

```
noisedeg$noise_R<-ifelse(noisedeg$noise=="no.noise",0,1)
head(noisedeg)

##    rt subj deg    noise noise_R
## 1 420   s1   0 no.noise       0
## 2 420   s2   0 no.noise       0
## 3 480   s3   0 no.noise       0
## 4 420   s4   0 no.noise       0
## 5 540   s5   0 no.noise       0
## 6 360   s6   0 no.noise       0
```

Linear models

Run the model

```
m0<-lm(rt~noise_R,noisedeg)
round(summary(m0)$coefficients,digits=2)

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)      486      23.70   20.50     0.00
## noise_R           90      33.52    2.68     0.01
```

Linear models

One instructive exercise is to compare the coefficients with the means we have for noise.
Coefficients:

```
coef(m0)

## (Intercept)     noise_R
##         486          90
```

The mean of no.noise is 486, and the mean of noise is 486+90=576:

```
## means of noise levels:
with(noisedeg,tapply(rt,noise,mean))

## no.noise    noise
##      486      576
```

Linear models

1. The intercept is giving us the mean of the no-noise condition. The intercept corresponds to the level of noise we recoded as "0".
2. The slope is giving us the difference in means between no-noise and noise.

Linear models

We can also fit a linear model where we examine the effect of deg on rt, or a model in which we examine together the effects of noise and deg. You will do this now, in Exercise 9.

Generating fake data for linear models

Statistical models express the generative process that led to the data. In Exercise 10, we will turn the problem of data analysis on its head: we will define the generative process, and then look at how well the model can recover the parameters.
Here is for instance how we could simulate a dataset that would correspond to the output of our M0 model

```
n<-10
beta1<-486
beta2<- 90
sigma <-105
x<-rep(c(0,1),each=n/2)
y <- beta1+ beta2*x + rnorm(n,mean=0,sd=sigma)
```

Generating fake data for linear models

This corresponds to the simple linear model with one categorical predictor coded as a dummy variable. When we fit such a model to the data, we are expressing a belief about how we think the data might be generated:

```
print(m<-lm(y~x))

##
## Call:
## lm(formula = y ~ x)
##
## Coefficients:
## (Intercept)            x
##      487.40         94.08
```

Generating fake data for linear models

In Exercise 9 you will repeatedly (1000 times) generate fake data as shown above; fit a linear model to the data (as shown above) each time and record the parameters from the model.

Summary

We now know how to fit

1. Simple linear models with continuous predictors.
2. Simple linear models with categorical predictors (the noise and degree data)

Next, we will discuss linear mixed models.