

Demo 3: T tests and ANOVA

Experimental and Statistical Methods in Biological Sciences I

September 29, 2014

1 Data preparations and descriptive statistics

1.1 Description of the data set

Today we will use data from a sleep deprivation study. The following variables are included:

- ID = participant ID
- age = age group (1 = young, 2 = old)
- education = education group (1 = comprehensive school, 2 = secondary, 3 = higher)
- ht_group = hormone treatment (user, control)
- digitssymbol = digit symbol test score at times 1, 2 and 3
- bentonerror = Benton test error score at times 1, 2 and 3

This study investigated the effects of sleep deprivation on two different tests: digit-symbol test (from WAIS-III) and Benton Visual Retention Test. Test scores were collected on three consecutive days always in the morning: after normal sleep, after sleep deprivation, and after normal sleep again. Some participants receive hormone treatment (HT) (coded as 'users'), some do not (coded as 'controls').

1.2 Load and prepare your data

Load the data from <http://becs.aalto.fi/~heikkih3/deprivation.csv> into a data frame `deprivation`.

Prepare the data as usual - check that factors are factors, and look for missing values and check how they are coded. Useful functions: `summary`, `head`, `boxplot`. (Refer to Demo 1 for help.)

1.3 Describe your data

Show descriptive statistics for your numeric variables in a table. Useful functions: `summary`, `describe`.

Plot test scores for Digit symbol and Benton tasks in all different time points. At this point, you can simply take box plots separately for each time point. Remember to adjust scales so that the scores are comparable (tip: named argument `ylim=` included in your `boxplot` function).

Finally, attach the `deprivation` data using `attach`.

(Refer to Demo 2 for help.)

2 Primer: distributions

Next, we will run a few simulations to demonstrate how different distributions look like. Do not worry too much about the commands we use to make simulations, the important thing is to focus on the behaviour and characteristics of different distributions.

2.1 Normal distribution

The great thing about normal distribution is that we know so much about it. We can make inferences regarding the position of different values along the distribution. For instance, we know how far from mean 95% of observations lie. This helps us to infer which phenomena are rare and which are not. This is essential in statistics: usually our goal is to decide whether we are confident enough to accept or reject our hypothesis, and this decision is based on distributions.

Many things in nature are normally distributed, including for instance height and psychological constructs such as IQ (which by definition is normally distributed). However, normal distribution is also useful in cases where the actual values do not follow a normal distribution. This is due to the **central limit theorem** which states that if you take repeated samples from a population and calculate their averages, then these averages will be normally distributed.

To demonstrate this, simulate five uniformly distributed random numbers between 0 and 10 and work out the average. Let's do this 10 000 times and look at the distribution of the 10 000 means. Distribution of the raw data is flat-topped:

```
> hist(runif(10000)*10, main="")
```

What about the distribution of sample means, based on taking just five uniformly distributed random numbers?

```
> means <- numeric(10000)
```

```
> for (i in 1:10000) {
```

```
means[i] <- mean(runif(5)*10)
}
> hist(means, ylim=c(0,1600))
```

Let's add a normal curve:

```
> h <- hist(means, freq=F, ylim=c(0,0.35))
> x <- seq(min(h$breaks),max(h$breaks), by=0.1)
> y <- dnorm(x, mean=mean(means), sd=sd(means))
> lines(x,y,col="red", lty="dashed", lwd=3)
```

The fit is excellent - the central limit theorem works.

2.2 Student's t-distribution

Student's t-distribution is used instead of the normal distribution when sample sizes are small ($n < 30$). Compared to the normal distribution, Student's t-distribution has flatter tails. The t-distribution depends on one parameter: the sample size.

In sample sizes close to 30 and higher, Student's t-distribution is close to normal distribution. Let's demonstrate this.

```
# create a vector
> simu <- seq(-4,4,0.01)
# draw a normal distribution:
> plot(simu, dnorm(simu), type="l", lty=2, ylab="Probability density", xlab="Deviates")
# add different t-distributions:
> lines(simu, dt(simu, df=5), col="red") # sample size = 5
> lines(simu, dt(simu, df=10), col="green") # sample size = 10
# ETC. play around by increasing sample size.
# What happens when you get close or beyond 30?
```

In Section 4, we will see how t-distribution relates to statistical testing of hypotheses.

2.3 F-distribution

Another important distribution is the F-distribution. It is especially relevant for ANOVA, since the ratio between two quotients follows F-distribution. F-distribution depends on two parameters, which are the degrees of freedom (DF) from the two quotients we are dividing. We can demonstrate the behaviour of F-distribution:

```

# create a vector
> simu2 <- seq(0,4,0.01)
# draw an F-distribution
> plot(simu2, df(simu2,1,1), type="l", lty=2, ylab="Probability density", xlab="Deviates")
# try changing degrees of freedom:
> lines(simu2, df(simu2, 1,10), col="red") # DF 1 = 1, DF = 10
> lines(simu2, df(simu2, 10,10), col="green") # DF 1 = 10, DF = 10
# ETC. play around by increasing both degrees of freedom separately.

```

In Section 5, we will see how F-distribution relates to statistical testing of hypotheses.

3 Tests for normality

There are several ways to test whether your variables are normally distributed.

First, you can simply use descriptive statistics to test for normality. Let's see the descriptive statistics for our numeric variables (Digit symbol and Benton task scores):

Check out normality for digit symbol and benton scores with `describe`:

```

> describe(deprivation)
# mean, st dev, median, skewness

```

Pay attention to the mean and median: are they close to each other? What about standard deviation, are the values widely scattered around the mean? How high is the skewness - values close to 1 hint for non-normality (exact limits depend on sample size)?

Second, while the descriptive statistics can give you hints, examining your variables visually is a much more powerful way. Remember histograms:

```

# Plot all numeric variables in same layout matrix:
> layout(matrix(c(1,2,3,4,5,6), 3, 2))
> hist(digitsymbol_1, main="Digit symbol 1")
> hist(digitsymbol_2, main="Digit symbol 2")
> hist(digitsymbol_3, main="Digit symbol 3")
> hist(bentonerror_1, main="Benton error 1")
> hist(bentonerror_2, main="Benton error 2")
> hist(bentonerror_3, main="Benton error 3")

```

Also, you can take “quantile-quantile plots” of your variables. If the qq-plot follows the line, the distribution is normal.

```
> qqnorm(digitsymbol_1)
> qqline(digitsymbol_1)
```

Third, you can also test for normality explicitly:

```
# Either with Shapiro-Wilk test:
> shapiro.test(digitsymbol_1) # Shapiro-Wilk

# Or with Kolmogorov-Smirnov test:
> ks.test(digitsymbol_1, "pnorm", mean=mean(digitsymbol_1), sd=sd(digitsymbol_1))
```

3.1 Exercises

Test for normality in all our numeric variables (Digit symbol task at time points 1, 2, and 3; Benton task at time points 1, 2, and 3):

Question 1 Take quantile-quantile plots.

Question 2 Perform normality tests and report the results.

4 T tests

T-tests are used to test whether the same distribution underlies two occasions. These two occasions can be...

1. one sample and known population
2. two independent samples
3. one sample in two different time points

All t-tests assume (i) that your sample size is at least 20, and (ii) that your variable is normally distributed. If your sample size is smaller, you should look for non-parametric tests. If your variable is not normally distributed, you should consider data transformations or non-parametric tests.

Check our normality tests from the previous section. Do all the variables fulfill these assumptions?

4.1 One-sample t-test

One-sample t-test can be used to compare one sample to the known population parameters. Therefore, we need to know what the real population mean is.

Let's test whether the participants in our study are representative for the population in Digit symbol test, i.e., whether the sample mean differs from the population mean. Assume that we know the population mean in Digit symbol test, and it is 40.

Check the descriptive statistics first:

```
> summary(digitsymbol_1)
# check out mean: 42.23
# slightly above the population mean (40)
# but is the sample mean statistically significantly different from the
# population mean?
# i.e. are sample and population from the same distribution?
```

Use one-sample t-test to compare sample mean to assumed population mean 40:

```
> t.test(digitsymbol_1, mu=40)
```

T test shows that the mean of our sample does not differ from the assumed population mean ($t(46) = 1.33, p > .05$).

4.2 Independent samples t-test

Independent samples t-test compares two sample means to each other and tests whether the two samples are derived from the same underlying distribution.

Let's test whether baseline performance in Digit symbol task varies depending on age group. This is done by performing a two-sample independent t-test for the digit symbol score in the two age groups:

```
> t.test(digitsymbol_1 ~ age)
```

We find out that there are no differences in Digit symbol task baseline performance between different age groups ($t(35) = 0.27, p > .05$).

By default, R assumes that the variances for the two populations are unequal, and it applies the Welch df modification when running the analysis. If you can assume that the variances are equal, you can also use

```
> t.test(digitsymbol_1 ~ age, var.equal=T)
```

How do the results of the t-test differ under the equal variances and unequal variances assumptions?

We can also visualize our results using the boxplots from the different samples in the same figure:

```
> boxplot(digitsymbol_1 ~ age, xlab="Age group", ylab="Digit symbol score")
```

4.3 Repeated-measures t-test

Repeated-measures t-test (also called paired t-test) compares measurements from the same sample to each other. For instance, we can test whether task scores differ between different time points or experimental manipulations. The requirement is that we have measurements from the same participants in different occasions.

Let's test whether performance in Digit symbol task varies between baseline (time point 1) and sleep deprivation (time point 2) by using a paired t-test:

```
> t.test(digitsymbol_1, digitsymbol_2, paired=T)
```

The performance in Digit symbol task does not differ at baseline and after sleep deprivation ($t(46) = -0.79, p > .05$).

We can also take boxplots again:

```
> boxplot(digitsymbol_1, digitsymbol_2, ylab="Digit symbol score", xlab="Time point")
```

(You can also use mean plots, these will be introduced later in Section 5.1.2).

4.4 Exercises

Question 3 Are there differences in baseline performance in (i) Digit symbol task, (ii) Benton task between the two different hormone treatment groups?

Question 4 Use repeated-measure t-tests to compare Digit symbol test scores between (i) times 1 and 3, (ii) times 2 and 3.

Question 5 Are there differences in Benton task performance between (i) times 1 and 2, (ii) times 1 and 3, or (iii) times 2 and 3?

5 Analysis of Variance

5.1 One-way ANOVA

One-way ANOVA includes one categorical variable and one continuous variable. Let's use one-way ANOVA to test whether baseline performance in Digit symbol task varies between the three education groups. This is done with `aov` function:

```
> A1 <- aov(digitsymbol_1 ~ education, data=deprivation)
```

Note the order of the arguments. The first argument is always the dependent variable (here, digit symbol score). It is followed by a tilde (`~`) and the independent variable(s). We are interested in the main effect of education so we add `education` as our only independent variable.

The `aov` function creates our ANOVA model. The model should always be saved as an object (here, we used `A1`) so that we can further examine the model details using other functions. The results of the ANOVA are printed with the `summary` command:

```
> summary(A1)
```

We see that the baseline Digit symbol scores do not differ between education groups ($F(2, 44) = 2.16, p > .05$).

The mean values of each condition can be read out from the model using `model.tables`:

```
> model.tables(A1, "means")
```

This function gives you the total mean and the means separately for each category.

You can also get a graphical summary by using `boxplot`:

```
> boxplot(digitsymbol_1 ~ education,
          xlab="Education group", ylab="Digit symbol score",
          main="Baseline Digit symbol score by education group"
          col="grey")
# diagnostic plots:
> layout(matrix(c(1,2,3,4),2,2))
> plot(A1)
```

The diagnostic plots give you the following information:

- check for heteroscedasticity (randomness of errors): residuals vs fitted, scale-location plots
- normality: q-q plot
- influential observations: leverage plot

5.1.1 ANOVA assumptions

Now that we know the basic structure of how ANOVA is run in R, let's return to the assumptions of ANOVA. These included the sample size of > 15 per group, normality, and homogeneity of variance. We already know how to test for normality.

The heteroscedasticity plot in previous Section gives you the graphical information of homogeneity of variance. You can also test for the homogeneity of variance using different tests which are implemented in R. The structure of the functions is similar to that of ANOVA models:

```
# The tests are from package 'car':
library('car')
# Levene's test:
> leveneTest(digitsymbol_1 ~ education)
# Bartlett's test:
bartlett.test(digitsymbol_1 ~ education)
```

5.1.2 Visualizing results

We have seen how to use box plots to visualize group differences:

```
> boxplot(digitsymbol_1 ~ education)
```

There are also other options for visualization, for instance the `plotmeans` function from `gplots` package:

```
> library(gplots)
> plotmeans(digitsymbol_1~education, ylim=c(10,70))
# Check ?plotmeans for different modifications:
> plotmeans(digitsymbol_1~education, ylim=c(10,70), n.label=FALSE)
```

5.2 Two-way between subject ANOVA

When you have more than one independent variable, you can use two-way ANOVA.

To demonstrate this, we want to ask whether age group and hormone treatment affect baseline performance in Digit symbol task. Thus, this is a 2×2 design with the factors being `age` and `ht_group`. The independent variables are separated by an asterisk `*` when added in the ANOVA - the asterisk indicates to R that the interaction between the two factors is interesting and should be analyzed in addition to the main effects of each independent variable. (If interactions are not important, replace the asterisk with a plus sign `+`.)

Run the analysis:

```

> A2 <- aov(digitsymbol_1 ~ age*ht_group, data=deprivation)
> summary(A2)
> model.tables(A2, "means")
> layout(matrix(c(1,2,3,4),2,2))
> plot(A2)

```

We see that age and hormone treatment do not affect baseline Digit symbol task performance alone ($F(1, 43) = 0.08, p > .05$ and $F(1, 43) = 0.02, p > .05$, respectively), but that the interaction effect of age and hormone treatment together is significant ($F(1, 43) = 6.81, p < .05$).

Let's use visualizations to help our interpretations. First, the familiar boxplots:

```

> boxplot(digitsymbol_1 ~ age*ht_group)

```

The boxplot does not really help our interpretation. Fortunately, R has many tools for data visualization, and one especially for two-way ANOVA: the `interaction.plot`.

```

> interaction.plot(age, ht_group, digitsymbol_1)
> interaction.plot(age, ht_group, digitsymbol_1, type="b", col=c(1:3), leg.bty="o",
  leg.bg="beige", lwd=2, pch=c(18,24,22), xlab="Age group",
  ylab="Digit symbol task score", main="Interaction plot")

```

5.2.1 Remember the type of Sum of Squares

Note that R provides Type I sequential sum of squares (not the default Type III marginal sum of squares reported by e.g. SPSS!). Thus, the order of independent variables matters. If you want to use the Type III sum of squares, use e.g. `Anova` function from package `car`:

```

> library(car)
> Anova(A2, type=3)

```

For more regarding the different types of sum of squares, see <http://egret.psychol.cam.ac.uk/statistics/R/anova.html>

5.2.2 Post-hoc comparisons

Post-hoc pairwise comparisons are commonly performed after significant effects have been found when there are three or more levels of a factor. After an ANOVA, you may know that the means of your response variable differ significantly across your factor, but you do not know which pairs of the factor levels are significantly different from each other. At this point, you can conduct pairwise comparisons.

Either using `pairwise.t.test`:

```
> pairwise.t.test(digitsymbol_1, education, p.adj="none")
# you can also adjust the p values according to a variety of methods:
> pairwise.t.test(digitsymbol_1, education, p.adj="bonf")
> pairwise.t.test(digitsymbol_1, education, p.adj="holm")
```

Or if you want to run Tukey post-hoc test specifically, you have to do it by using `TukeyHSD` function:

```
> TukeyHSD(A1)
```

5.3 Exercises

Question 6 Does hormone treatment affect Digit symbol baseline performance differently in different education groups?

Question 7 Do education and age together affect the Digit symbol baseline performance?

Question 8 Let's investigate the baseline performance in Benton task next. Do age, education or hormone treatment have a main effect on Benton task performance? What about their interaction effects? Report your results both verbally and visually.

6 Exercises

Next we will apply our skills on another dataset, the already familiar `naming` dataset. This time, it comes in a different format than in previous weeks - namely in the wide format ¹. Also note that participants with missing values have been removed.

The following variables are included:

- `iq` = intelligence (continuous)
- `hrs` = hours spent on reading (continuous)
- `sex` (female, male)
- `ms.regular` = average time taken to read regular words (continuous)

¹This was done with the following command:

```
> naming_wide <- reshape(naming, direction="wide", idvar=c("hrs", "iq", "sex"), timevar="word.type")
```

'reshape' is a useful command when working with longitudinal type of data. It allows you to change your data from long to wide format or the other way round.

- `ms.exception` = average time taken to read exceptional words (continuous)
- `reading_time` = reading category based on hours spent on reading (high, low, medium)

You can find the dataset here: http://becs.aalto.fi/~heikkih3/naming_wide.csv.

Load the dataset into a data frame in R using `read.csv`. Remember to check for factors and missing values first! Correct these if necessary using guidelines from previous weeks.

Once your data frame is in its final format, it is useful to attach it. First, remove other attached datasets with `detach()`, then attach naming instead (`attach`).

In the following exercises, run the relevant analysis and report your results both verbally and visually. Think carefully what kind of plot describes your results best and add necessary modifications (labels, title, ...). Save the plots for the significant results you get.

Question 9 Does the mean intelligence in our sample differ from the population mean of intelligence (=100)? (one-way t-test)

Question 10 Are there differences between men and women in intelligence? (t-test)

Question 11 Are there differences between men and women in hours spent on reading? (t-test)

Question 12 Are there differences between the three reading categories in intelligence? (one-way anova)

Question 13 Does the time taken to read the word differ between regular and exceptional words? (paired t-test)

Question 14 Does hours spend on reading affect the time taken to read regular words? What about exceptional words? (one-way anova - remember that your independent variable needs to be categorical!)

Question 15 Does gender affect how hours spent on reading affect the time taken to read regular words? What about exceptional words? (two-way anova)