16. Correlation & Linear Regression

Correlation analysis investigates the relationship between two continuous variables. If a significant relationship exists, you may want to predict the values of one variable from another, which is regression analysis. In linear regression analysis there are exact methods to fit parameters "a" and "b" of the function y=a+b*x, where "a" is the intercept with the y axis and b the slope of the regression line. However, for more complex functions, curve fitting in R and SAS is based on trial and error. You have to know what the parameters of your function represent, and provide the SAS or R routines with start values for each parameter that you need to guess by looking at your data.

Correlation and regression assume that your data is normally distributed and that variances are equal (called homogeneity of variance, or homoscedasticity).

16.1 Pearson's Correlation

- Download (trees.csv on the website) or manually enter the following data and import them into R:

ID	DBH	VOL	AGE	DENSITY
2	11.5	1.09	23	0.55
3	5.5	0.52	24	0.74
4	11	1.05	27	0.56
5	7.6	0.71	23	0.71
6	10	0.95	22	0.63
7	8.4	0.78	29	0.63
9	8.4	0.77	21	0.64
12	9	0.87	27	0.6

```
dat1 = read.csv("trees.csv")
attach(dat1)
```

- Explore the relationship between the variables DBH, VOL, AGE and DENSITY and the significance of the relationships. This is the R code to calculate (1) a correlation coefficient, (2) test the significance of a correlation, (3) to create a matrix of correlation coefficients, (4) to create a matrix of r^2 values for all pairs of variables. For the matrix below, note that we should remove the ID column, which is not an independent or dependent variable (so we don't want to include it in the correlation matrix):

```
cor(DBH,DENSITY)  #calculate the correlation coefficient
cor.test(DBH,DENSITY)   #test significance of the correlation
cor.test(DBH,VOL)
cor(DBH,VOL)

dat2=dat1[,2:5]
cor(dat2)   #create a matrix of correlation coefficients
cor(dat2)^2 #create a matrix of r2 values
```

- The equivalent in SAS:

```
proc corr data=trees;
var DBH VOL AGE DENSITY;
run;
```

16.2 Linear Regression

- If you want to use a significant linear relationship to make predictions, you need to derive an equation of the format:

  y=m * x + b *or* "dependent variable" = "slope" * "independent variable" + "intercept"

  The statistical test for the significance of a regression function is actually a test of whether the slope of the regression is significantly different from zero. This is identical to the test of significant correlations above. You may also get an output that tests whether the intercept is significantly different from zero (i.e. the value of y ≠ 0 when x = 0), which you would usually ignore unless this is of interest in the context of your analysis.

- This is the code in R. The first line just gives you the formula. The second line returns the full range of statistics. Subsequent commands give you two different ways to fit the regression line to a plot. The curve function has the advantage that you can type any formula and it does not exceed the data range like the `abline()` function.

```
lm(VOL~DBH)
summary(lm(VOL~DBH))
plot(VOL~DBH)
abline(lm(VOL~DBH))
plot(VOL~DBH)
curve(-0.0237+x*0.097, add=T, lty=2)
```

- The equivalent in SAS.

```
proc glm data=trees;
model DENSITY=VOL;
run;
```

- Below is a useful table relating correlation coefficients as a function of the proportion of explained variance (r^2) to their corresponding level of significance (p-value). The top rows contains your degrees of freedom (df=n-2), and the bottom rows are the minimum r² for significance at the α=0.05 level. For example, in our case with n=8 observations (thus df=6), all correlation coefficients above 0.71 are significant at α=0.05. This is a handy table to determine the significance of correlation coefficients without using the software.

df	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r^2	0.99	0.95	0.88	0.81	0.75	0.71	0.67	0.63	0.60	0.58	0.55	0.53	0.51	0.50	0.48	0.47	0.46	0.44	0.43	0.42

df	21	22	23	24	25	26	27	28	29	30	35	40	45	50	60	70	80	90	100
r^2	0.41	0.40	0.40	0.39	0.38	0.37	0.37	0.36	0.36	0.35	0.33	0.30	0.29	0.27	0.25	0.23	0.22	0.21	0.20

16.3 Testing Assumptions

- Homogeneity of variances (homoscedasticity) and normality in Y for a given X value can best be explored with residual plots. We have discussed how to test for both normality and equal variances many times in in previous labs. Testing for homoscedasticity usually involves an examination of the "residuals", which are simply the "leftovers" after you deduct you data values from a given value (e.g. from the mean). Residual plots, residual calculation, histogram or residuals, and the Shapiro test for normality in R can be generated with:

```
plot(lm(DENSITY~VOL))
res=residuals(lm(DENSITY~VOL))
hist(res)
shapiro.test(res)
```

- Residual plots in SAS:

```
proc glm data=trees;
model DENSITY=VOL;
output out=res r=residuals p=predicted;
run;
proc gplot data=res;
plot residuals*predicted;
run;
```

- If assumptions of normality and/or homogeneity of variances are violated, you can calculate correlation coefficients for non-parametric (i.e. non-normal) data. Both Kendall (shown here) and Spearman (see 16.4) rank correlations are very well regarded robust test statistics for non-parametric correlations. However, keep in mind that they also assume linearity of the relationship.

```
cor(DBH,VOL,method="kendall")
cor.test(DBH,VOL,method="kendall")
```

16.4 Spearman Rank Correlation

- A Spearman correlation coefficient, represented by the greek letter "rho" (ρ) instead of r, is essentially the Pearson correlation between the ranked values of the variables. It is a useful calculation if 1) you are only interested in the direction of a relationship and not the magnitude, or 2) when you have highly non-normal data that you cannot transform to normality.
- While you may get a higher correlation coefficient with a Spearman test, we must be careful with this. Remember that we're asking a fundamentally different question with a Pearson correlation (relationship between the **order and magnitude** of the data values) than with a Spearman correlation (relationship between the **order** of the data values).
- Let's test it out. We can calculate the ranks for our data using the simple `rank()` command in R. Remember that we have to re-attach the data when we add new columns.

```
dat1$rankDBH = rank(DBH)
dat1$rankDENSITY = rank(DENSITY)
dat1
attach(dat1)
```

- Now, let's run a Pearson correlation (the default setting in the `cor()` and `cor.test()` command) for the ranked data and see how it compares to the Pearson correlation that we ran before on the regular data.

```
cor.test(rankDBH,rankDENSITY)
```

- We can run a Spearman correlation in R easily in the cor.test() command without making ranked data columns, however. The result should be exactly the same (though you will be warned about ties, which cannot really be ranked). You just need to specify the `method="spearman"` in the regular R correlation commands:

```
cor(DBH,DENSITY,method="spearman")
cor.test(DBH,DENSITY,method="spearman")
```

16.5 The Problem of Multiple Inferences

- If you want to draw general conclusions from a table of correlations (or in fact tables of any kind of statistical test) you need to make adjustment for multiple inference. This is because the probability of making a type I error by pure chance increases every time you run another statistical test.
- To illustrate the point for yourself, do the following experiment: The code below generates two random datasets and then carries out a correlation analysis. Obviously, the correlation between two random datasets should be zero. Execute the code multiple times (all three lines) and after a while you will find a significant relationship (about 1 out of 20 times if your alpha level is 0.05).

```
r1=rnorm(10)
r2=rnorm(10)
cor.test(r1,r2)
```

  Hint: you can run all three lines together quickly by separating them with a semi-colon instead of three separate lines:

```
r1=rnorm(10); r2=rnorm(10); cor.test(r1,r2)
```

- The more tests you make, the more likely you will eventually make a type I error: rejecting the null hypothesis when it is true. If you are asking general questions, you have to protect yourself against this type I error inflation. To do this, we use a simple p-value adjustment every time we perform multiple tests. This is not just for correlation, but any time we perform multiple tests on the same data.
- Below is a data table (also posted on website as multiple_inference.csv) where we examined plant growth as a function of climate variables. The correlation is reported between growth and monthly temperature with the corresponding p-value. The question is: "Is growth dependent on climate?" and the answer, based on a cursory examination of this table, is "Yes, there are significant relationships with temperature in April, May, July, and August at α=0.05".
- However, that's not quite right. You are drawing a general inference: "Climate influences growth", so you have to adjust for multiple inferences to account for the increased probability of making a Type I error by chance. We do this with the command `p.adjust()` in R, which returns adjusted p-values based on the number of tests or comparisons that you are running (12, in this case). There are many methods for adjusting p-values (see the `?p.value` help file). For now, try out the Holm and Bonferroni adjustments. What are your conclusions based on this table after adjustments? Which adjustment appears to be more conservative?

```
inference = read.csv("multiple_inference.csv")
inference$p_holm = p.adjust(inference$Pvalue,method="holm",n=12)
inference$p_bonf = p.adjust(inference$Pvalue,method="bonferroni",n=12)
inference
```

Climate variable	Correlation w/ growth (r2)	p-value
Temp Jan	0.03	0.4700
Temp Feb	0.24	0.2631
Temp Mar	0.38	0.1235
Temp Apr	0.66	0.0063
Temp May	0.57	0.0236
Temp Jun	0.46	0.1465
Temp Jul	0.86	0.0001
Temp Aug	0.81	0.0036
Temp Sep	0.62	0.0669
Temp Oct	0.43	0.1801
Temp Nov	0.46	0.1465
Temp Dec	0.07	0.4282

CHALLENGE:

1. Why is the fundamental purpose of regression? Of correlation?
2. How are the 3 methods we use to calculate a correlation coefficient different?
3. Why should we be concerned about multiple inferences?