Lecture 6 - Linear regression and analysis of variance

Some more about the normal distribution

```r
x = rnorm(10000)
hist(x, n=30); v()
```

![Histogram of x](image)

How far are 5% of the points?

```r
x = sort(x)
length(x)/100*5
[1] 500
x[500]
[1] -1.627749
x[length(x)-500]
[1] 1.629363
```
We see that 5% of the points are 1.6 standard deviations below the mean, and 5% 1.6 above the mean.

```r
> x[250]
[1] -1.945368
> x[length(x)-250]
[1] 1.971970
```

If we look at a distance of 1.96 std deviations away from the mean, or more, we'll see 5% of the points. There is a function that does this:

```r
> qnorm(0.025)
[1] -1.959964
> qnorm(0.05)
[1] -1.644854
> qnorm(0.005)
[1] -2.575829
```

It is convenient to remember that if we are ~2 std. deviations away, we are at the 5% level, and at 3 std. deviations we are at less than 1% level.

Even for this there is a function:

```r
> pnorm(-3)*2
[1] 0.002699796
> pnorm(-2)*2
[1] 0.04550026
> pnorm(-1)*2
[1] 0.3173105
```

### 2 dimensional distributions

Let us take 2 points that are taken from a normal, and look where they lie:

```r
> x=rnorm(2000)
> y=rnorm(2000)
```
> plot(x, y, pch=".", xlim=c(-3, 3), ylim=c(-3, 3)); v()

> grid(lwd=1, lty=1, col=2); v()
Let us count how many points are in each box

```r
x.cut = cut(x, -3:3)
x.cut[1:20]

[1] (0,1] (0,1] (1,2] (-2,-1] (0,1] (0,1] (1,2] (-1,0] (0,1] [10] (0,1] (-3,-2] (-1,0] (-2,-1] (-1,0] (-1,0] (0,1] (1,2] (-1,0] [19] (-1,0] (0,1] Levels: (-3,-2] (-2,-1] (-1,0] (0,1] (1,2] (2,3]
```

You see that each point was replaced by the interval that it is in.

```r
y.cut = cut(y, -3:3)
tab.xy = table(x.cut, y.cut)
tab.xy

y.cut
x.cut  (-3,-2] (-2,-1] (-1,0] (0,1] (1,2] (2,3] (-3,-2] 1 7 15 15 9 3 (-2,-1] 4 31 86 85 37 8 (-1,0] 14 82 238 242 96 13 (0,1] 16 95 226 234 83 16 (1,2] 4 34 94 99 40 7 (2,3] 3 9 13 24 4 2
```

Now we know how many are in each cell. Let us plot these:

```r
library(lattice)
levelplot(tab.xy); v()
```

Now let us do the same, but with a finer grid, and more points:

```r
x=rnorm(1000000); y=rnorm(1000000)
x.cut=cut(x, seq(-3,3, length=50) ); y.cut=cut(y, seq(-3,3,length=50) )
tab.xy=table(x.cut,y.cut)
levelplot(tab.xy,colorkey=F,col.regions=rainbow(100));v()
```
Here we see why it makes sense to talk about sum of squares for normally distributed data:

All points for which $x^2 + y^2 = \text{constant}$, i.e. points on a circle, are equally likely.

That is not true for other distributions. For example, the log-normal distribution:

```r
> hist(exp(x), br = seq(0, 1000, by = 0.05), xlim = c(0, 1.5)); v()
```

![Histogram of exp(x)](image)
Let us do a 2d plot in the same way here:

```r
x.cut=cut(exp(x), seq(0,1.5, length=50) ); y.cut=cut(exp(y), seq(0,1.5,length=50) )
> tab.xy=table(x.cut,y.cut)
> levelplot(tab.xy,colorkey=F,col.regions=rainbow(100));v()
```
This is why it makes sense to try to minimize sum of squares for normally distributed data (with equal variances), but not always for non-normaly distributed data.

**Linear regression**

```r
> a=read.table("data/regression.txt",head=T)
> plot( growth ~ tannin, data=a, pch=16);v()
```

```r
> reg=lm( growth ~ tannin, data=a)
> summary(reg)
```

```
Call:
  lm(formula = growth ~ tannin, data = a)

Residuals:
     Min       1Q   Median       3Q      Max
-2.4556  -0.8889  -0.2389   0.9778   2.8944

Coefficients:                  Estimate Std. Error t value Pr(>|t|)
(Intercept)            11.7556     1.0408   11.295   9.54e-06 ***
```

```r
```
tannin  -1.2167  0.2186  -5.565  0.000846 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.693 on 7 degrees of freedom
Multiple R-Squared:  0.8157,   Adjusted R-squared:  0.7893
F-statistic: 30.97 on 1 and 7 DF,  p-value: 0.000846

Now let us look at aids to help us see how good the model is:

```r
two.by.two=t(matrix(1:4,2,2))
layout(two.by.two); par(cex=0.7)
plot(reg);v(width=6,height=6)
```

![Residuals vs Fitted](image1)
![Normal Q–Q plot](image2)
![Scale–Location plot](image3)
![Cook's distance plot](image4)
Let us look what these plots look like when a linear fit is not a good approximation:

```r
> x=1:50
> y=x^2+rnorm(50,sd=100)
> plot( y ~ x );v()
```
> reg=lm(y ~ x)
> abline(reg,col=2); v()

> summary(reg)

Call:
  lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-335.50 -191.21  -49.87  137.18  569.01

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -484.816    69.086   -7.018  6.91e-09 ***
x              51.827     2.358   21.981  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
Residual standard error: 240.6 on 48 degrees of freedom
Multiple R-Squared: 0.9096,  Adjusted R-squared: 0.9077
F-statistic: 483.1 on 1 and 48 DF,  p-value: < 2.2e-16

> layout(two.by.two)
> plot(reg); v(width=6,height=6)

Another example:

> x=1:50
> y=x+rnorm(50,sd=1:50/3)
> plot(y~x); v()
```r
> reg=lm(y~x)
> abline(reg,col=2);v()
```
> layout(two.by.two); plot(reg); v( width=6,height=6); layout(1)
Analysis of variance

Let us do the following:

```r
> x=1:2
> y=rnorm(2)
> plot(x,y);v()
```
> reg=lm(y~x)
> abline(reg,col=2);v()

> Oh, nice! Perfect fit!

Compare this with the fit we would get if we had a slope of 0:

> reg0=lm(y~1)
> reg0

Call:
  lm(formula = y ~ 1)

Coefficients:
(Intercept)
  0.03782

> abline(0.03782,0,col=3);v()
Isn’t the red line a much better fit than the green one?

No, it isn’t!

We can fit a line through any two points! So, if we have two measurements, it does not make sense to fit more than one parameter.

```r
anova(reg, reg0)
```

Analysis of Variance Table

<table>
<thead>
<tr>
<th>Res.Df</th>
<th>RSS</th>
<th>Df</th>
<th>Sum of Sq</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.6424</td>
<td>-1</td>
<td>1.6424</td>
<td>-1</td>
<td>-1.6424</td>
</tr>
</tbody>
</table>

These are the degrees of freedom of the model. Every independent measurement adds a degree of freedom, and every (independent) fitted parameter takes one degree away.

Let us do this again

```r
x <- 1:3
y <- rnorm(3)
```
> plot(x,y); v()

> reg=lm(y~x); abline(reg,col=2); v()
So, again - the red line looks like a much better fit than the green one - but is it?
Was it worth it to add another coefficient? Is the added fit significant?

\[
\text{anova(reg,reg0)}
\]

Analysis of Variance Table

<table>
<thead>
<tr>
<th></th>
<th>Res.Df</th>
<th>RSS</th>
<th>Df</th>
<th>Sum of Sq</th>
<th>F</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>1</td>
<td>0.56735</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model2</td>
<td>2</td>
<td>1.55077</td>
<td>-1</td>
<td>0.98342</td>
<td>1.7333</td>
<td>0.4135</td>
</tr>
</tbody>
</table>

Analysis of variance tells us here than adding another coefficient did not make the model better, because the p-value is pretty big.

Let us look at an example (from the book...)

\[
a = \text{read.table("data/factorial.txt",head=T)}
\]

\[
a
\]

growth diet coat
1   6.6   A light
2   7.2   A light
3   6.9   B light
4   8.3   B light
5   7.9   C light
Now we can try to fit linear models:

```r
reg1 = lm(growth ~ coat, data=a)
reg2 = lm(growth ~ diet + coat, data=a)
reg3 = lm(growth ~ coat:diet + coat + diet, data=a)
anova(reg1, reg2, reg3)
```

Analysis of Variance Table

<table>
<thead>
<tr>
<th>Model 1: growth ~ coat</th>
<th>Model 2: growth ~ diet + coat</th>
<th>Model 3: growth ~ coat:diet + coat + diet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.Df</td>
<td>RSS</td>
<td>Df</td>
</tr>
<tr>
<td>----------</td>
<td>-----</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>5.5167</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>2.8567</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>2.1700</td>
</tr>
</tbody>
</table>

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

It is easier to call the function `aov`, which will call `lm` along the formula:
> model1=aov( growth ~ coat+diet+diet:coat, data=a)
diet*coat is a shorthand for coat+diet+diet:coat

> model1=aov(growth~coat*diet, data=a)
> summary(model1)

<table>
<thead>
<tr>
<th>Df</th>
<th>Sum Sq</th>
<th>Mean Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>coat</td>
<td>1</td>
<td>2.61333</td>
<td>2.61333</td>
<td>7.2258</td>
</tr>
<tr>
<td>diet</td>
<td>2</td>
<td>2.66000</td>
<td>1.33000</td>
<td>3.6774</td>
</tr>
<tr>
<td>coat:diet</td>
<td>2</td>
<td>0.68667</td>
<td>0.34333</td>
<td>0.9493</td>
</tr>
<tr>
<td>Residuals</td>
<td>6</td>
<td>2.17000</td>
<td>0.36167</td>
<td></td>
</tr>
</tbody>
</table>

---

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

> layout(two.by.two);par(cex=0.8);plot(model1)
> v(width=6,height=6);layout(1)
model2=update(model1,".-diet:coat")

anova(model1,model2)

Analysis of Variance Table

Model 1: growth ~ coat + diet + diet:coat
Model 2: growth ~ coat + diet

<table>
<thead>
<tr>
<th></th>
<th>RSS</th>
<th>Df</th>
<th>Sum of Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.Df</td>
<td>6</td>
<td>1</td>
<td>2.17000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>-2</td>
<td>-0.68667</td>
<td>0.9493</td>
<td>0.4383</td>
</tr>
</tbody>
</table>

>

So, it says that model1 is not significantly better than model2.

summary(model2)

Df Sum Sq Mean Sq F value Pr(>F)
coat 1 2.61333 2.61333 7.3186 0.02685 *
diet 2 2.66000 1.33000 3.7246 0.07190 .
Residuals 8 2.85667 0.35708

---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>

Let us remove diet:

model3=update(model2,".-diet")

anova(model2,model3)

Analysis of Variance Table

Model 1: growth ~ coat + diet
Model 2: growth ~ coat

<table>
<thead>
<tr>
<th></th>
<th>RSS</th>
<th>Df</th>
<th>Sum of Sq</th>
<th>F value</th>
<th>Pr(&gt;F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.Df</td>
<td>8</td>
<td>1</td>
<td>2.8567</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>5.5167</td>
<td>-2</td>
<td>-2.6600</td>
<td>3.7246</td>
<td>0.0719</td>
</tr>
</tbody>
</table>

---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>

So, model2 is not significantly better than model3.

summary(model3)

Df Sum Sq Mean Sq F value Pr(>F)
coat 1 2.6133 2.6133 4.7372 0.05457 .
Residuals 10 5.5167 0.5517

---
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

>

Now coat is not significant either!

layout(matrix(1:2,1,2));plot( growth ~ diet, data=a);
plot( growth ~ coat, data=a);v();layout(1)
We see that diet C seems very different from diet A and B, but that they are not very different from each other. Let us just make a model that depends on whether the diet is C or not.

```r
> dietC = as.factor(a$diet == "C")
> model4 = update(model3, . + dietC)
> anova(model3, model4)
```

Analysis of Variance Table

<table>
<thead>
<tr>
<th>Model 1: growth ~ coat</th>
<th>Model 2: growth ~ coat + dietC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.Df</td>
<td>RSS</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

So, model4 is significantly better than model3.

Let us try adding interaction:

```r
> model5 = update(model4, . + dietC:coat)
> anova(model4, model5)
```

Analysis of Variance Table

<table>
<thead>
<tr>
<th>Model 1: growth ~ coat + dietC</th>
<th>Model 2: growth ~ coat + dietC + coat:dietC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Res.Df</td>
<td>RSS</td>
</tr>
<tr>
<td>1</td>
<td>9</td>
</tr>
</tbody>
</table>

---

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
This addition is not worthwhile.