Dummy Variables

May 9, 2020

```
In [10]: library(MASS)
```

0.1 Water data

> For 61 towns in the UK, collected data on mortality rates for males over a 5-6 year period, as well as knowing how much calcium is in the water. We also have a 2 level factor classification for north and south.

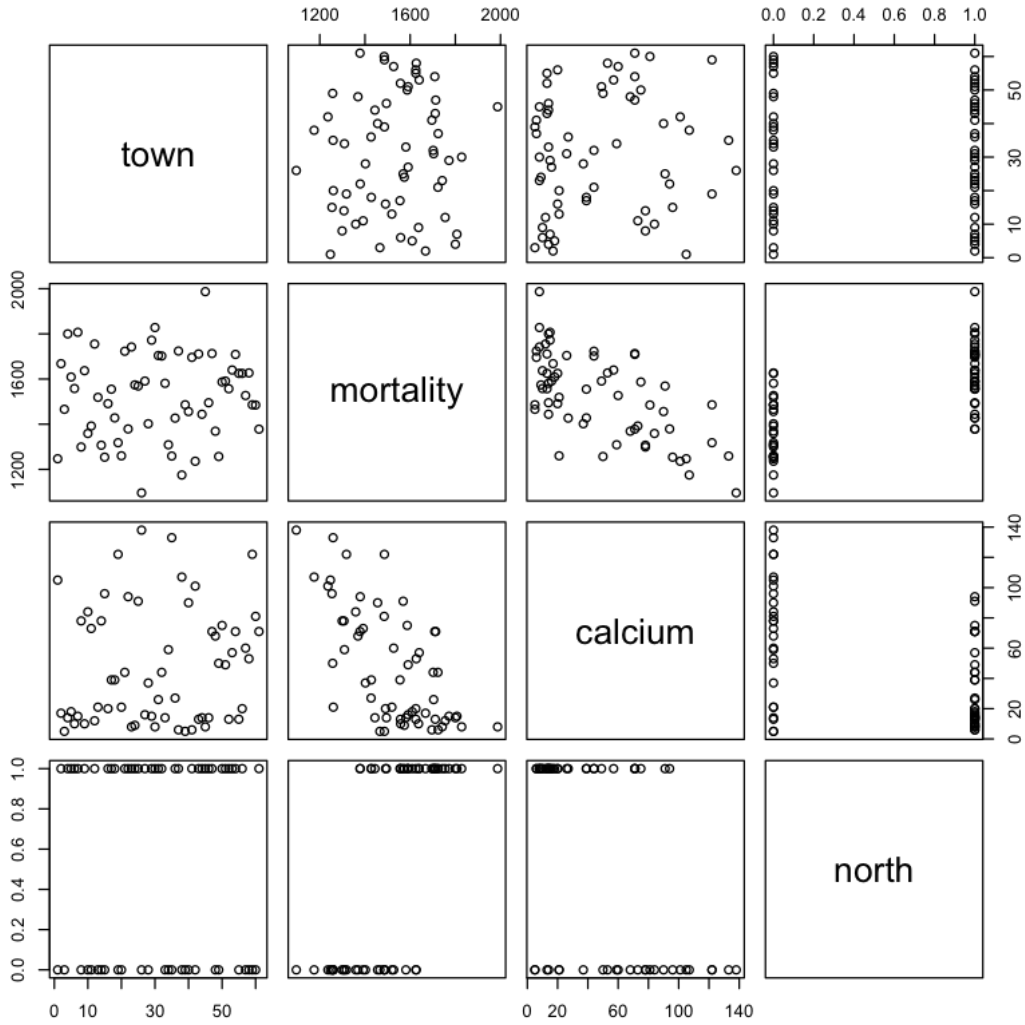
This two level classification is modelled using a dummy variable, z_i, where $z_i = 0$ if South (control group, acting as a baseline), and $z_i = 1$ if North.

```
In [45]: water <- read.csv('water1.csv')

         water[1:5,]
```

town	mortality	calcium	north
Bath	1247	105	0
Birkenhead	1668	17	1
Birmingham	1466	5	0
Blackburn	1800	14	1
Blackpool	1609	18	1

0.2 Plotting the data

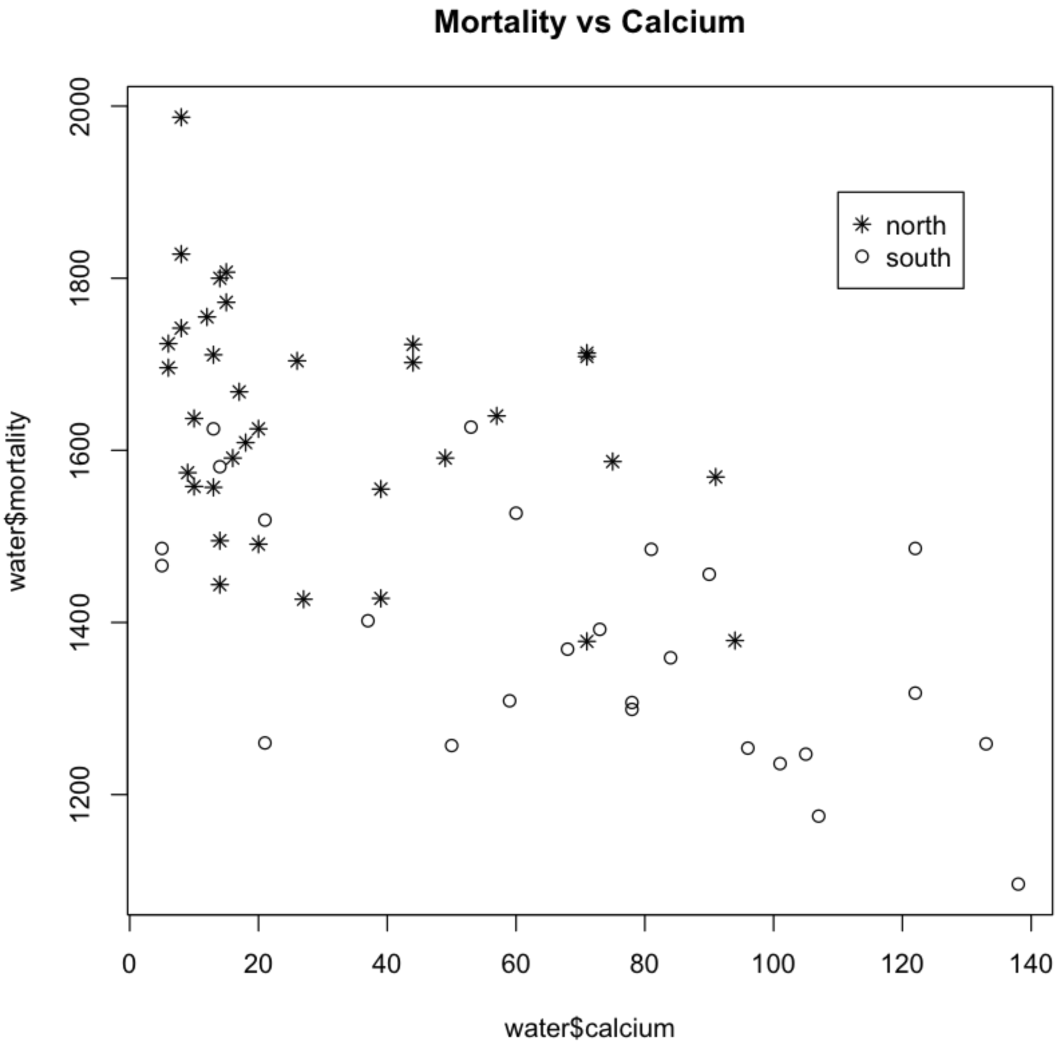
```
In [46]: pairs(water)
```



0.3 Plot from the notes

Fancy plots to have different markers for north and south, so you can start to see higher mortality rates for the North, and what looks like a linear relationship with calcium for both levels.

```
In [47]: plot(water$calcium, water$mortality, type = "n", main = "Mortality vs Calcium")
         points(water$calcium[water$north==0], water$mortality[water$north==0], pch=1)
         points(water$calcium[water$north==1], water$mortality[water$north==1], pch=8)
         legend(110, 1900, c("north", "south"), pch =c(8,1))
```



0.4 Simple model

$y = \beta_0 + \beta_1 x$

Not taking the North/South factor into account

```
In [48]: lm.simple <- lm(mortality ~ calcium, data=water)
         summary(lm.simple)
```

```
Call:
lm(formula = mortality ~ calcium, data = water)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-348.61 -114.52   -7.09  111.52  336.45

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1676.3556    29.2981  57.217  < 2e-16 ***
calcium       -3.2261     0.4847  -6.656 1.03e-08 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 143 on 59 degrees of freedom
Multiple R-squared:  0.4288,Adjusted R-squared:  0.4191
F-statistic:  44.3 on 1 and 59 DF,  p-value: 1.033e-08
```

```
In [49]: anova(lm.simple)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
calcium	1	906185.3	906185.33	44.29615	1.033134e-08
Residuals	59	1206988.3	20457.43	NA	NA

```
In [12]: sqrt(44.29615)
```

6.65553529026779
Summary:

- F statistic highly significant, sufficient evidence to reject the null hypothesis that at least one of the parameters is non-zero
- Residual standard error, $s = \sqrt{s^2} = \sqrt{MS_R}$ = 143
- $R^2 = 0.42$, so there's a fair chunk of the variation yet to be explained by the regresssion sum of squares.

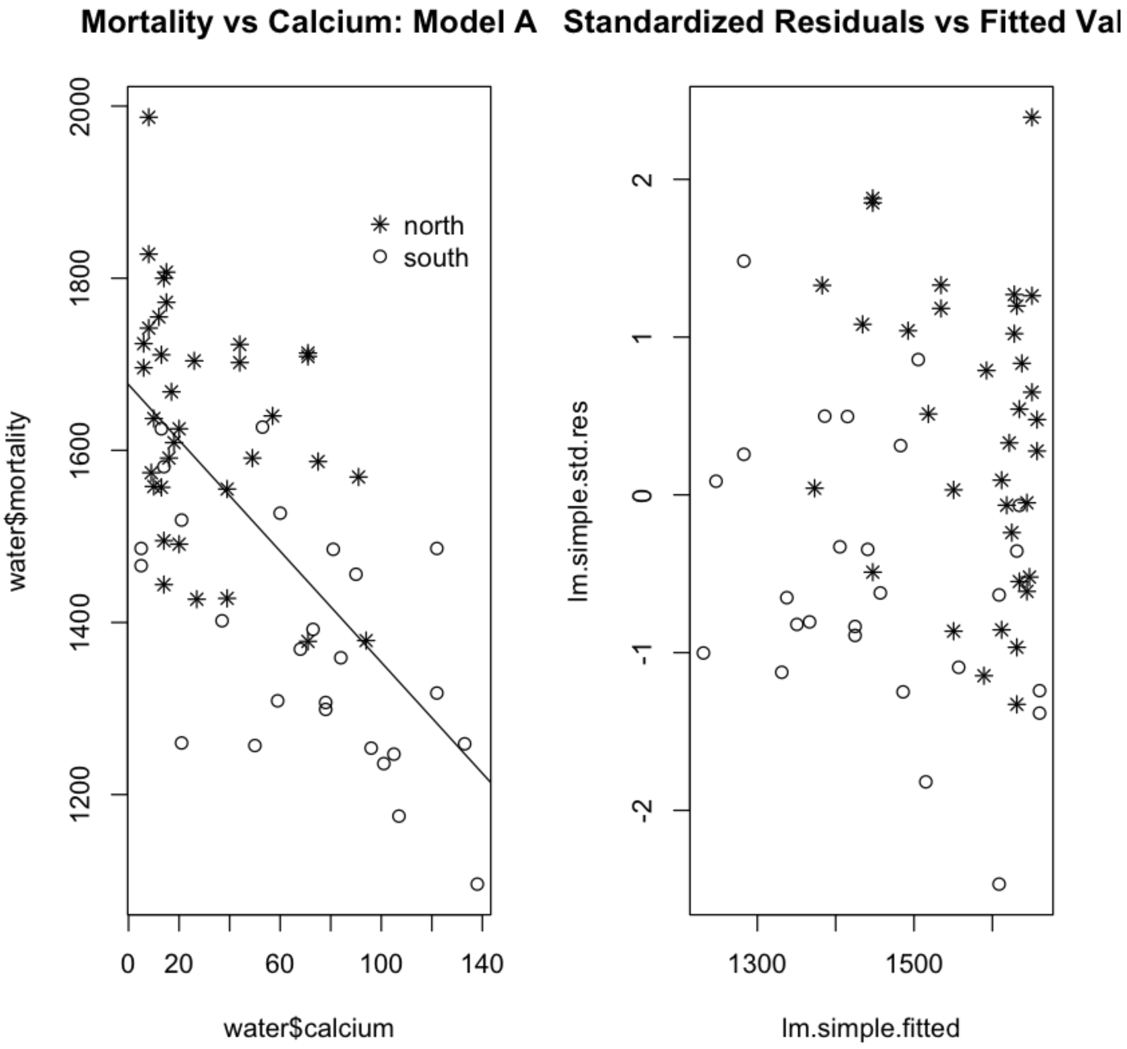
0.5 More plotting and fitting code from the lecture notes:

Looking at model adequecy:

- Model shows linear fit;
- Plot of standardised residuals Vs fitted values shows significantly more northern values above zero, meaning there's a tension with our mean zero random error assumption, and that we may want to explore adding another variable into our model to explain this variation.

```
In [52]: par(mfrow = c(1, 2))
```

```
####
## FIRST PLOT
plot(water$calcium, water$mortality, type = "n", main = "Mortality vs Calcium: Model A
points(water$calcium[water$north == 0], water$mortality[water$north == 0], pch = 1)
points(water$calcium[water$north == 1], water$mortality[water$north == 1], pch = 8)
legend(90, 1900, c("north", "south"), pch =c(8,1), bty="n")

#ăhere's a quick way to display the
abline(lm.simple)

####
## SECOND PLOT

##ăgetting the standardised residuals and the fitted values so we can do some
## MODEL ADEQUECY TESTS
lm.simple.std.res <- stdres(lm.simple)
lm.simple.fitted <- fitted(lm.simple)

#ăplot plots x, y
# thus to plot y against x, and hence to plot standardised residuals Vs fitted values
#ăplot fitted, std.res
plot(lm.simple.fitted, lm.simple.std.res, type = "n", main = "Standardized Residuals v
points(lm.simple.fitted[water$north == 0], lm.simple.std.res[water$north == 0], pch =
points(lm.simple.fitted[water$north == 1], lm.simple.std.res[water$north == 1], pch =
```



0.6 Fitting new model with north as a factor, but no interaction term, so different intercepts but same slopes.

```
In [53]: lm.factor.no.int <- lm(mortality ~ calcium + factor(north), data=water)

         summary(lm.factor.no.int)


Call:
lm(formula = mortality ~ calcium + factor(north), data = water)

Residuals:
     Min       1Q   Median       3Q      Max
```

```
-222.959  -77.281    7.143   90.751  307.836

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1518.7263    41.3350  36.742  < 2e-16 ***
calcium          -2.0341     0.4829  -4.212 8.93e-05 ***
factor(north)1  176.7108    36.8913   4.790 1.19e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 122.1 on 58 degrees of freedom
Multiple R-squared:  0.5907,Adjusted R-squared:  0.5766
F-statistic: 41.86 on 2 and 58 DF,  p-value: 5.601e-12
```

0.7 Analysis of the summary:

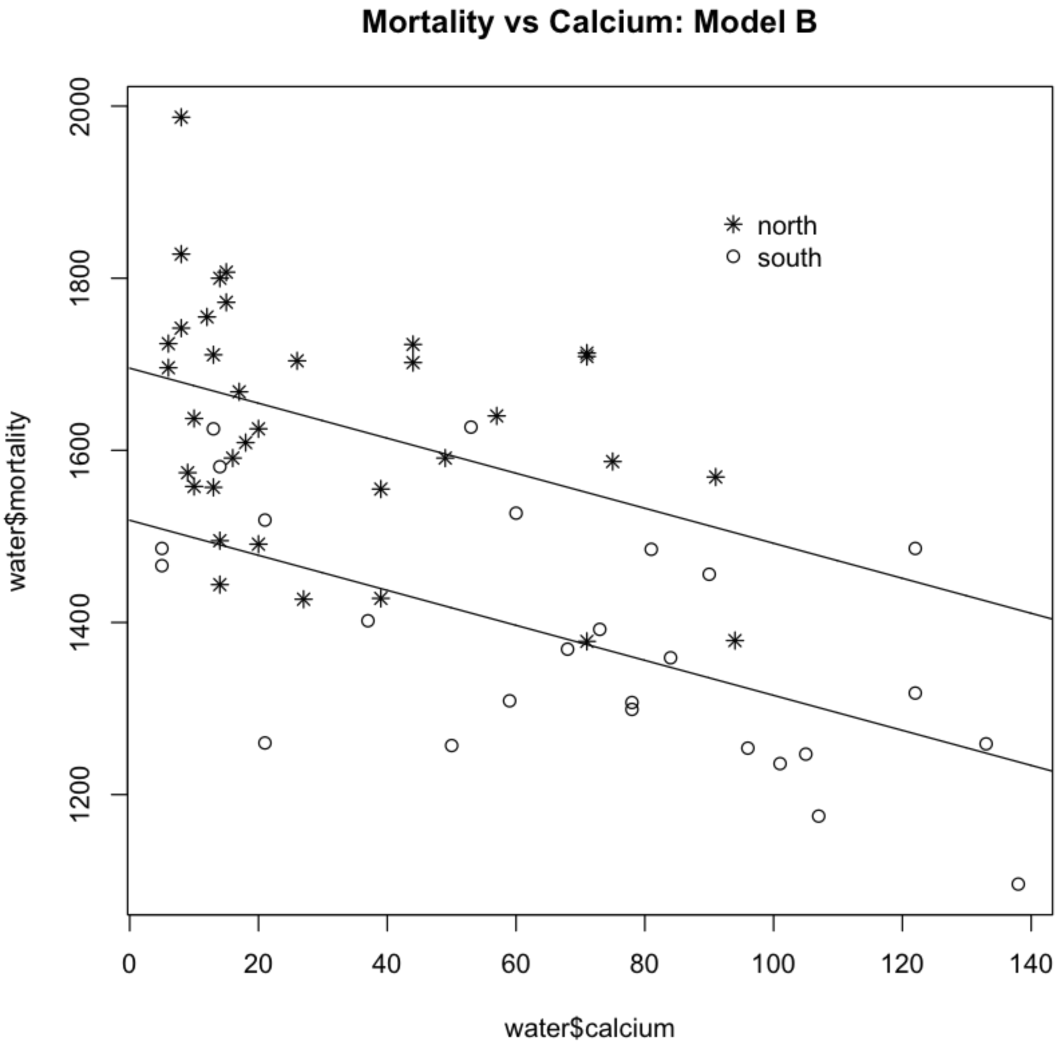
- Overall F statistic is highly significant, which is the statistic used to test the null hypothesis that all parameters are zero, i.e. that none of the explanatory variables provide any benefit in the model over the mean of the dependent variable
- Calcium parameter highly significant;
- Factor for North highly significant too, meaning there's evidence to suggest there's a **difference in the intercept** for North and South at the 0.1% level of significance.
- R^2 of 0.6, thus 60% of the variation in the dependent variable is being explained by the fitted model, where $R = \frac{SS_{Reg}}{SS_T} = 1 - \frac{SS_R}{SS_T}$
  - This R squared value is better than the R squared value without the N/S classification factor - so adding the two-level factor to the model explains a lot more of the variation in the dependent variable, **mortality**.
- Residual plot (plotting the standardised residuals Vs the fitted values) is much more acceptable, as we have a more even spread of the two different classifications (N/S) around mean zero.
- Lastly, the residual standard error - which is the square of the residual variance, s^2, which is equal to the residual mean square error, is lower. The lower the sum of squared residuals / mean square residuals, the lower the sample variance, and the better the fitted data compares to the real observations.

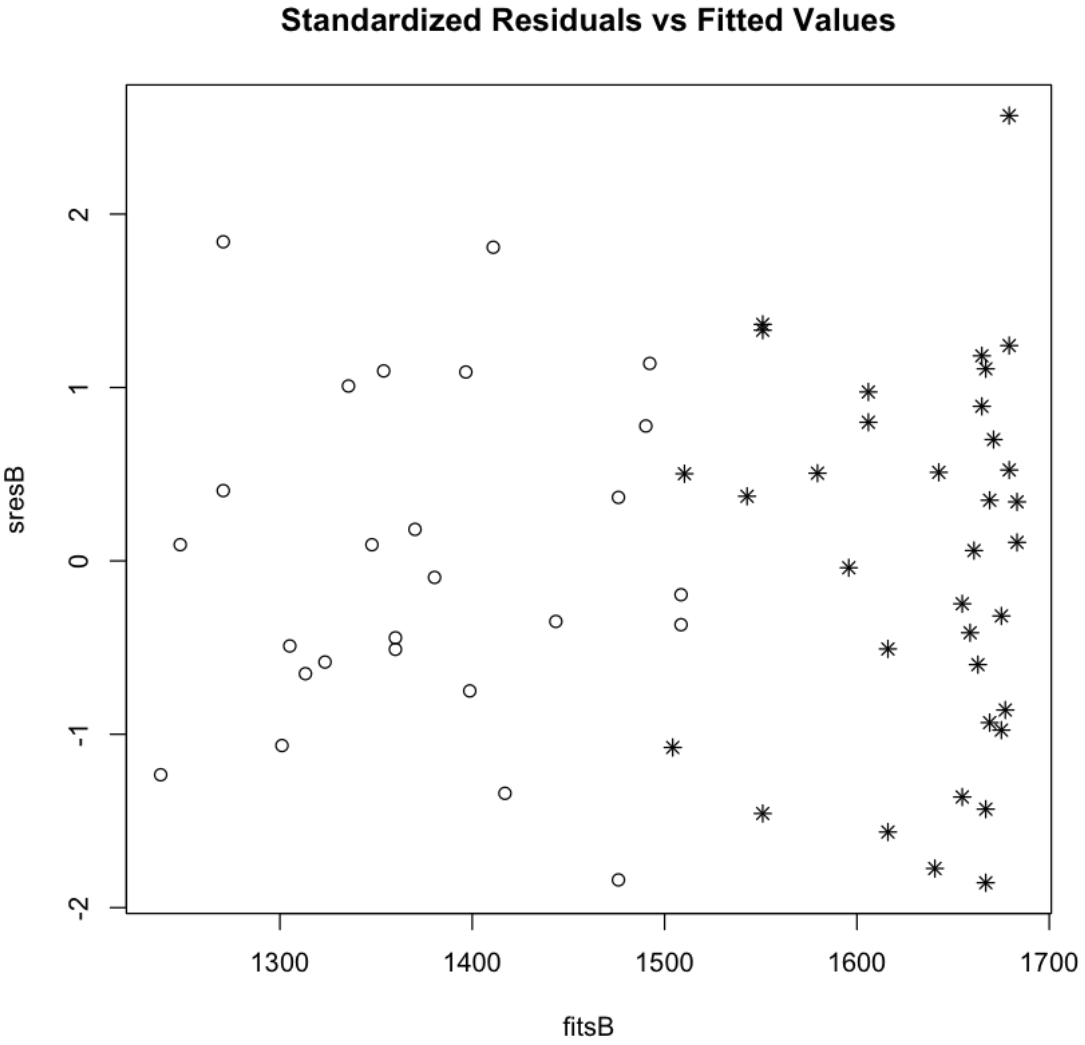
  DON'T FORGET TO MENTION THE RESIDUAL STANDARD ERROR AND IT'S LINK TO SAMPLE VARIANCE AND MEAN SQUARE RESIDUAL

0.7.1 And a bunch of fancy plotting code:

Can clearly see that this model produces two parallel lines: different intercepts and the same slope.

```
In [54]: plot(water$calcium, water$mortality, type = "n", main = "Mortality vs Calcium: Model
         points(water$calcium[water$north == 1], water$mortality[water$north == 1], pch = 8)
         legend(90, 1900, c("north", "south"), pch =c(8,1), bty="n")

         ld <- seq(0, 145, 0.1)
         lines(ld, predict(lm.factor.no.int, data.frame(calcium = ld, north = rep(0, length(ld
         type = "response"))

         lines(ld, predict(lm.factor.no.int, data.frame(calcium = ld, north = rep(1, length(ld
         type = "response"))

         sresB <- stdres(lm.factor.no.int)
         fitsB <- fitted(lm.factor.no.int)

         plot(fitsB, sresB, type = "n", main = "Standardized Residuals vs Fitted Values") > po
         points(fitsB[water$north == 1], sresB[water$north == 1], pch = 8)
```




Standardized Residuals vs Fitted Values
sresB
2
1
0
-1
-2
fitsB
1300
1400
1500
1600
1700

1 WANT TO TEST WHETHER DIFFERENT SLOPES FOR DIFFERENT FACTOR LEVELS -> ADD INTERACTION TERMS BETWEEN EXPLANATORY VARIABLE AND FACTOR

1.1 Fitting new model with interaction term, so essentially fitting two different simple models, as can have different intercepts and different slopes

```
In [57]: lm.factor.interaction <- lm(mortality ~ calcium*factor(north), data=water)

         summary(lm.factor.interaction)
```

```
Call:
lm(formula = mortality ~ calcium * factor(north), data = water)

Residuals:
    Min      1Q  Median      3Q     Max
-221.27  -75.45    8.52   87.48  310.14

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            1522.8150    48.9500  31.110  < 2e-16 ***
calcium                  -2.0927     0.6103  -3.429  0.00113 **
factor(north)1          169.4978    58.5916   2.893  0.00540 **
calcium:factor(north)1    0.1614     1.0127   0.159  0.87395
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 123.2 on 57 degrees of freedom
Multiple R-squared:  0.5909,Adjusted R-squared:  0.5694
F-statistic: 27.44 on 3 and 57 DF,  p-value: 4.102e-11
```

1.2 (...which is the same as...)

```
In [58]: lm.factor.interaction <- lm(mortality ~ calcium + factor(north) + calcium:factor(nort

         summary(lm.factor.interaction)
```

```
Call:
lm(formula = mortality ~ calcium + factor(north) + calcium:factor(north),
    data = water)

Residuals:
    Min      1Q  Median      3Q     Max
-221.27  -75.45    8.52   87.48  310.14
```

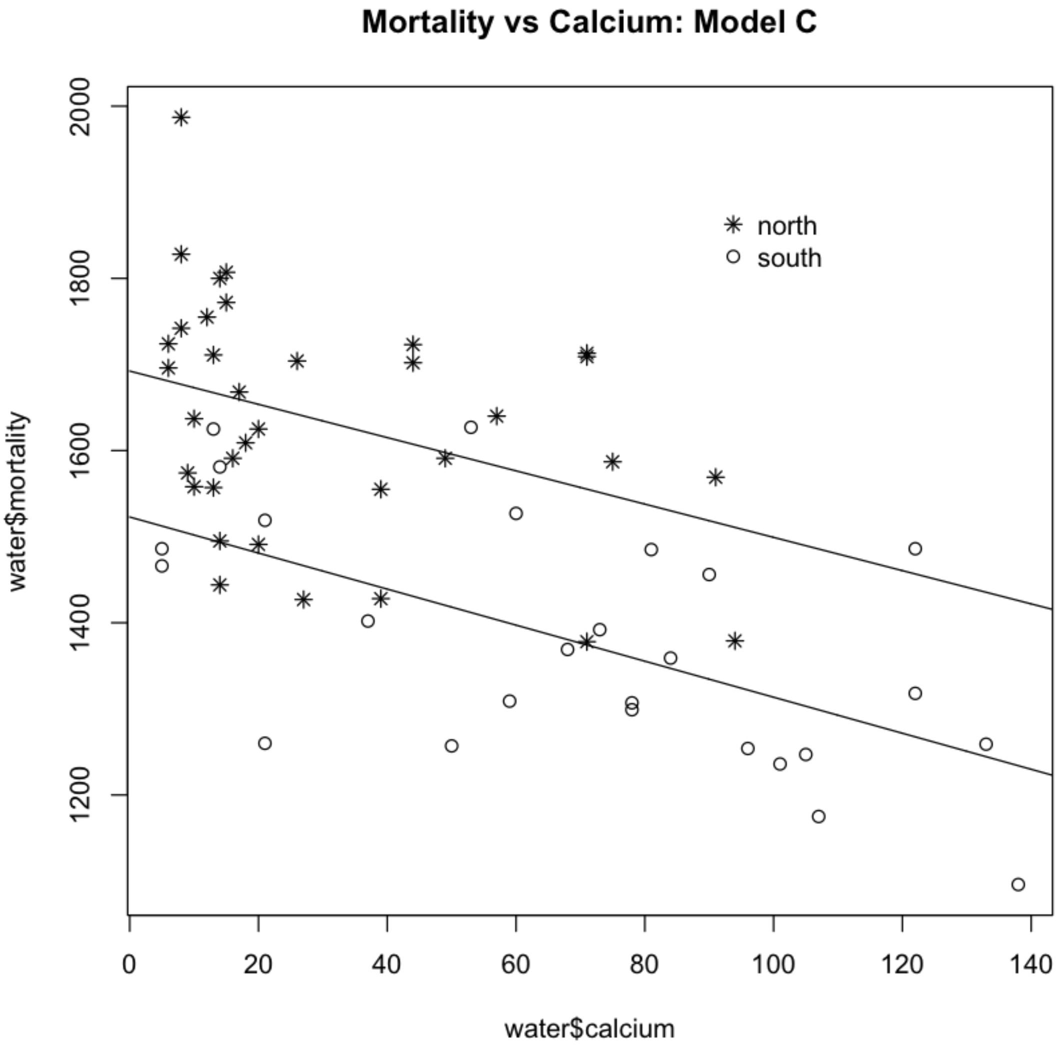
```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)            1522.8150    48.9500  31.110  < 2e-16 ***
calcium                  -2.0927     0.6103  -3.429  0.00113 ** 
factor(north)1          169.4978    58.5916   2.893  0.00540 ** 
calcium:factor(north)1    0.1614     1.0127   0.159  0.87395    
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 123.2 on 57 degrees of freedom
Multiple R-squared:  0.5909,Adjusted R-squared:  0.5694 
F-statistic: 27.44 on 3 and 57 DF,  p-value: 4.102e-11
```

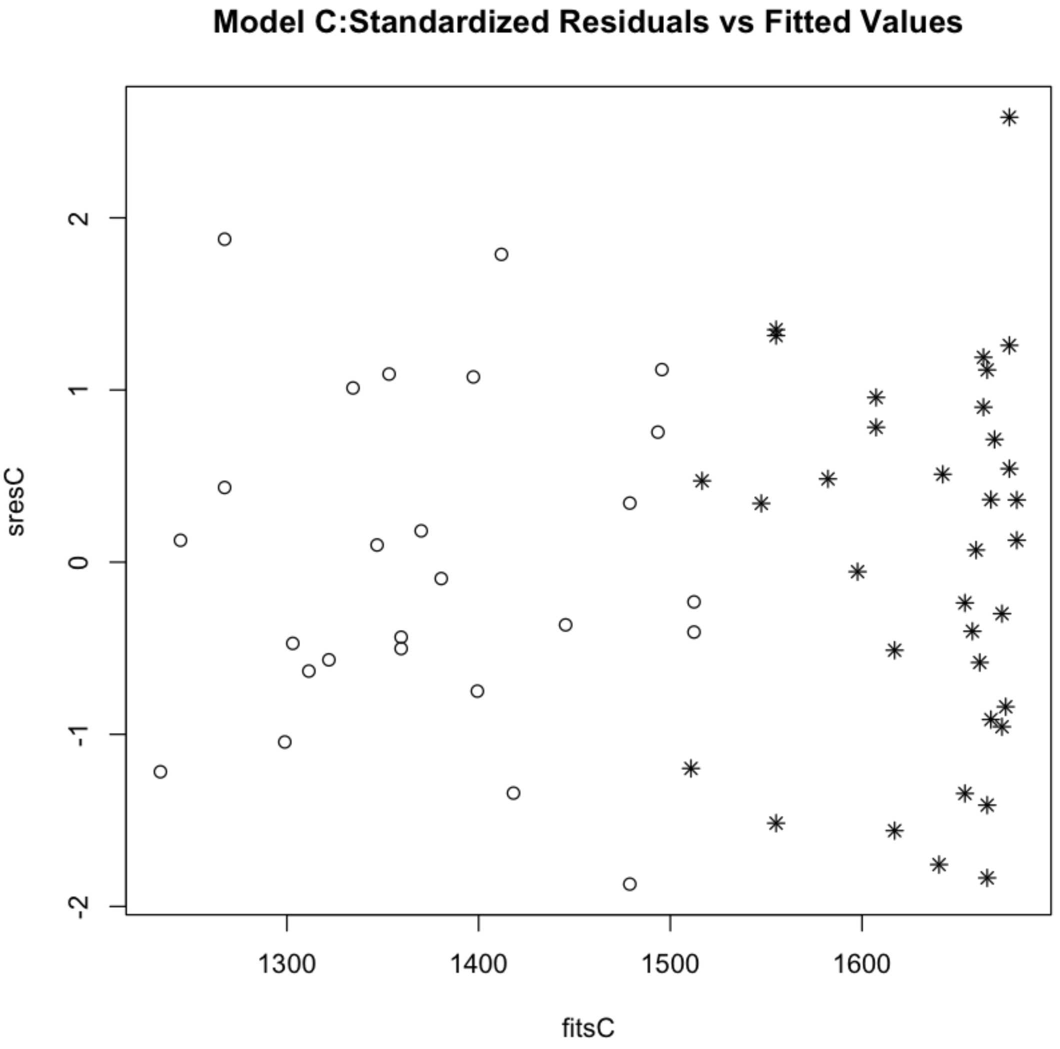
1.3 Analysis of the interaction model:

- Main thing to look at here is the fact that the interaction term is not significant, and therefore there is insufficient evidence to reject the null hypothesis that the slope is the same for the North and South data.
- We therefore prefer model B

1.4 More fancy plots of the fit and the standardised residuals against the fitted values to look at model adequecy:

```
In [61]: plot(water$calcium, water$mortality, type = "n", main = "Mortality vs Calcium: Model (
         points(water$calcium[water$north == 0], water$mortality[water$north == 0], pch = 1)
         points(water$calcium[water$north == 1], water$mortality[water$north == 1], pch = 8)
         legend(90, 1900, c("north", "south"), pch =c(8,1), bty="n")

         ld <- seq(0, 145, 0.1)
         lines(ld, predict(lm.factor.interaction, data.frame(calcium = ld, north = rep(0, leng
         type = "response"))

         lines(ld, predict(lm.factor.interaction, data.frame(calcium = ld, north = rep(1, leng
         type = "response"))

         sresC <- stdres(lm.factor.interaction)
         fitsC <- fitted(lm.factor.interaction)

         plot(fitsC, sresC, type = "n", main = "Model C:Standardized Residuals vs Fitted Value
         points(fitsC[water$north == 0], sresC[water$north == 0], pch = 1)
         points(fitsC[water$north == 1], sresC[water$north == 1], pch = 8)
```


Mortality vs Calcium: Model C
✳ north
○ south
water$mortality
2000
1800
1600
1400
1200
water$calcium
0
20
40
60
80
100
120
140



2 Looking at dummy variables with more than two classificatios - going beyond binary

2.1 Using the iris dataset

Have three species of iris: *setosa*, *versicolor*, and *virginica*.

50 plants from each species - therefore $n = 150$ in total.

Species is therefore our three-level factor, $m = 3$, where m is the number of levels in our species factor

We know that we we need $m - 1$ dummy variables to code this factor into our regression

The factor (species) name ordered first alphabetically will form a baseline, and the other two factors will be w.r.t. that baseline.

2.2 The question we're trying to probe here:

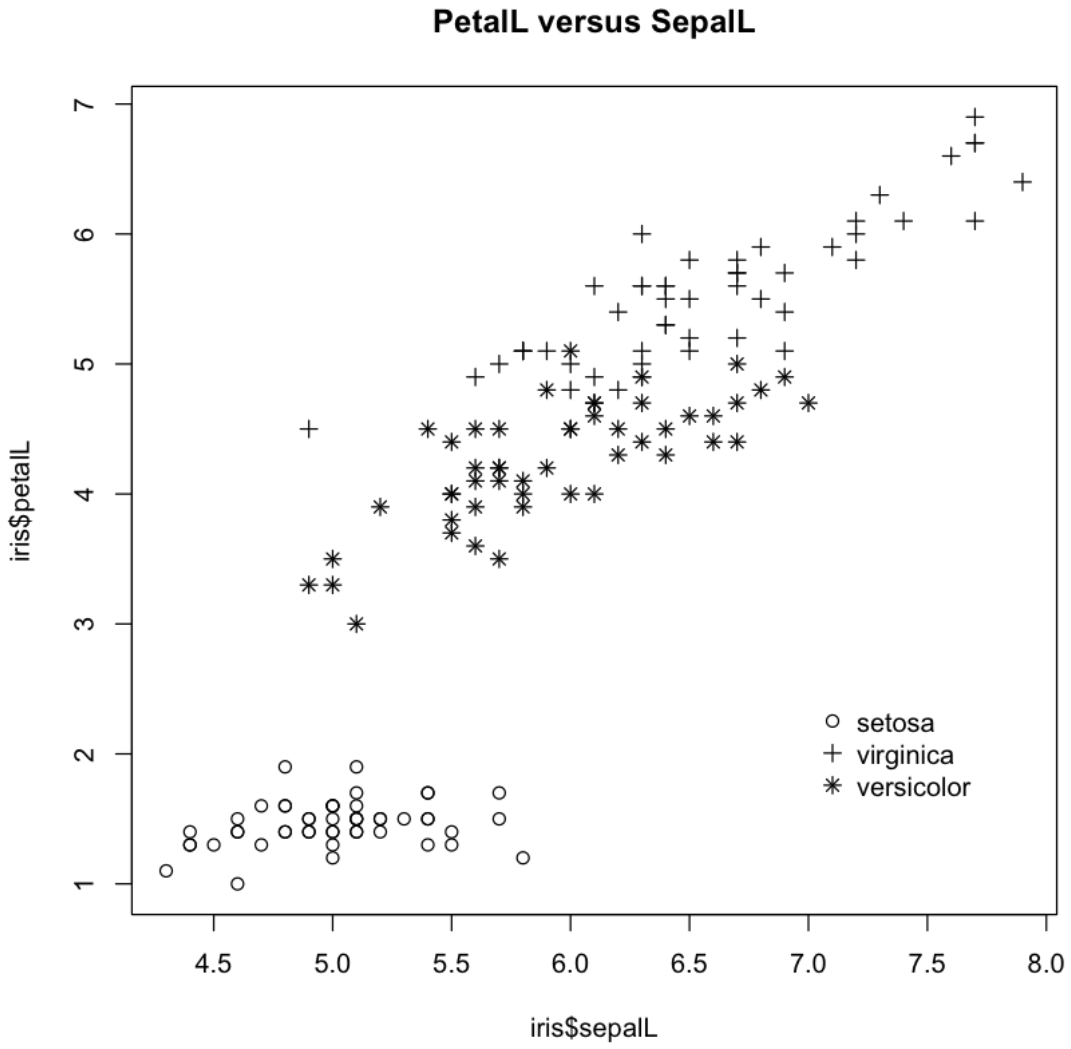
2.2.1 Relationship between petal length and sepal length, and does this relationship change between species (i.e. will they need different intercepts and different slopes?)

```
In [63]: library(datasets)

         data(iris)

         colnames(iris) <- c('sepalL','sepalW','petalL','petalW','species')

         iris[1:5,]
```

sepalL	sepalW	petalL	petalW	species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

2.3 Fancy plotting of the Iris dataset

```
In [75]: plot(iris$sepalL, iris$petalL, type = "n", main = "PetalL versus SepalL")
         points(iris$sepalL[iris$species == "setosa"], iris$petalL[iris$species == "setosa"], 
         points(iris$sepalL[iris$species == "virginica"], iris$petalL[iris$species == "virgini
         points(iris$sepalL[iris$species == "versicolor"], iris$petalL[iris$species == "versico
         legend(7, 2.5, c("setosa", "virginica", "versicolor"), pch =c(1, 3, 8), bty="n")
```



2.4 Starting with the simple model:

$y_i = \beta_0 + \beta_1 x_i$, where y_i is our petal length dependent variable for the i-th observation, and x_i is our sepal length explanatory variable for the i-th observation, for $i = 1, 2, ..., 150$.

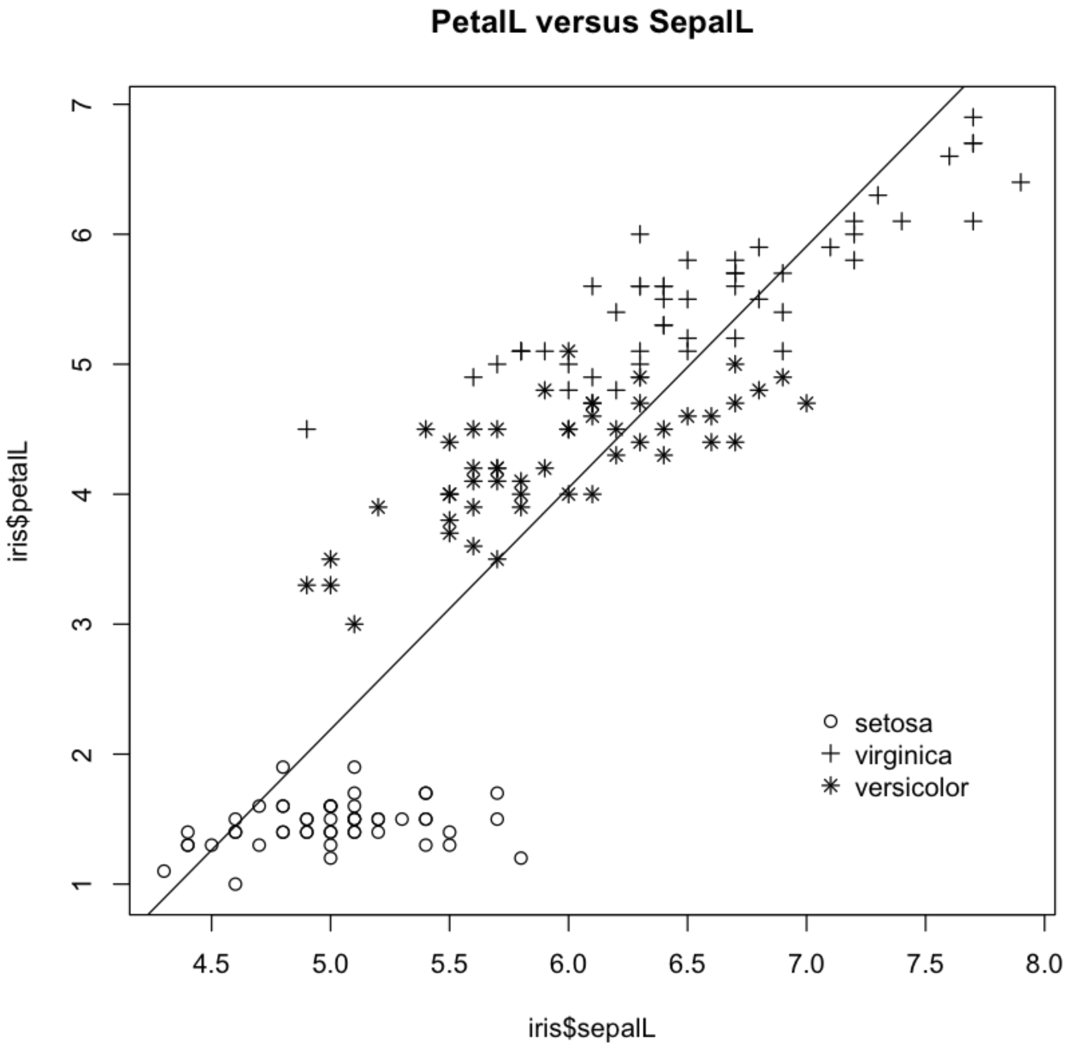
```
In [65]: lm.iris.simple <- lm(petalL ~ sepalL, data=iris)

         summary(lm.iris.simple)
```

```
Call:
lm(formula = petalL ~ sepalL, data = iris)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-2.47747 -0.59072 -0.00668  0.60484  2.49512

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -7.10144    0.50666  -14.02   <2e-16 ***
sepalL       1.85843    0.08586   21.65   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.8678 on 148 degrees of freedom
Multiple R-squared:   0.76,Adjusted R-squared:  0.7583
F-statistic: 468.6 on 1 and 148 DF,  p-value: < 2.2e-16
```

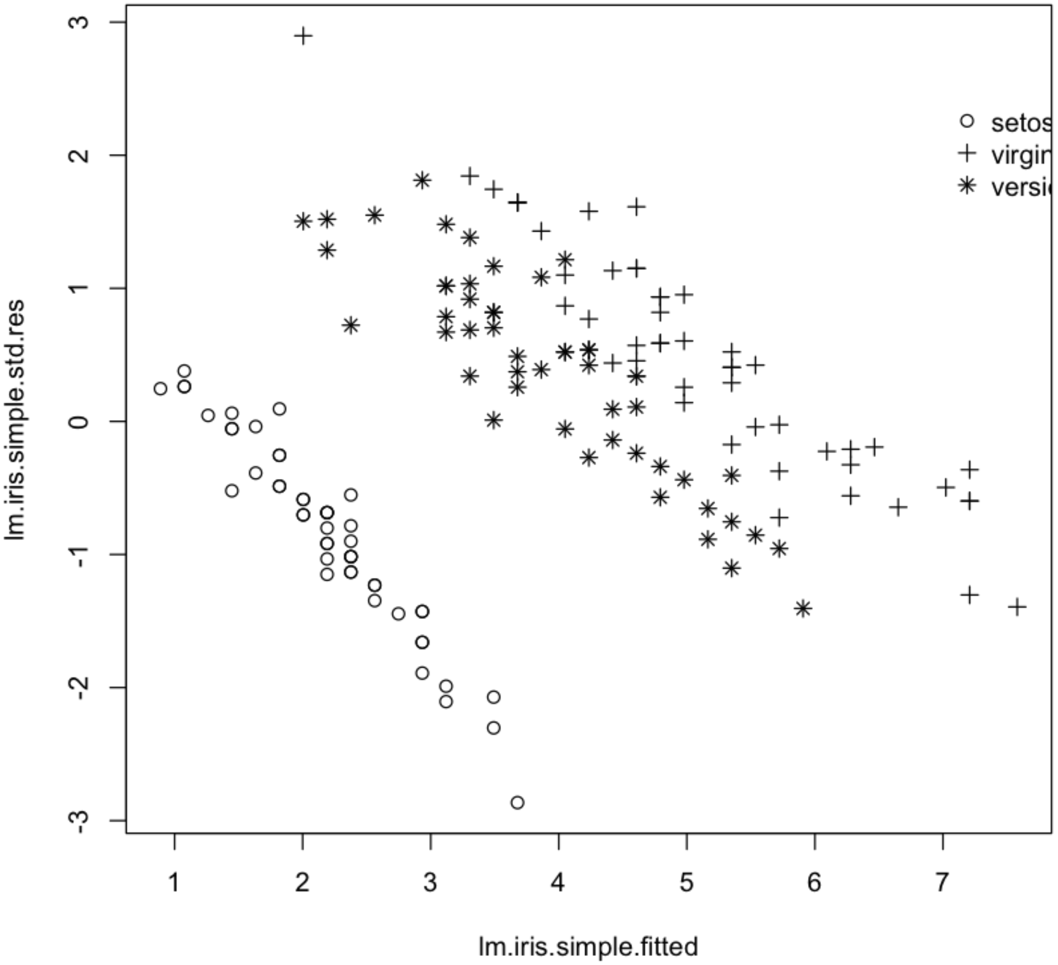
2.5 Plot of the fitted model with the observed data

```
In [68]: plot(iris$sepalL, iris$petalL, type = "n", main = "PetalL versus SepalL")
         points(iris$sepalL[iris$species == "setosa"], iris$petalL[iris$species == "setosa"], p
         points(iris$sepalL[iris$species == "virginica"], iris$petalL[iris$species == "virginic
         points(iris$sepalL[iris$species == "versicolor"], iris$petalL[iris$species == "versico
         legend(7, 2.5, c("setosa", "virginica", "versicolor"), pch =c(1, 3, 8), bty="n")


         #ăhere's a quick way to display the
         abline(lm.iris.simple)
```



2.6 Model adequecy plot: standardised residuals Vs the fitted values

```
In [77]: lm.iris.simple.std.res <- stdres(lm.iris.simple)

         lm.iris.simple.fitted <- fitted(lm.iris.simple)

         plot(lm.iris.simple.fitted, lm.iris.simple.std.res, type='n')
         points(lm.iris.simple.fitted[iris$species == 'setosa'], lm.iris.simple.std.res[iris$sp
         points(lm.iris.simple.fitted[iris$species == 'virginica'], lm.iris.simple.std.res[iris
         points(lm.iris.simple.fitted[iris$species == 'versicolor'], lm.iris.simple.std.res[ir
         legend(7, 2.5, c("setosa", "virginica", "versicolor"), pch =c(1, 3, 8), bty="n")
```



2.6.1 Comments on the adequecy plot:

- Clear trend between the three levels of the factor
- The setosa group clearly do not have mean zero
- Clearly also do not have constant variance as the points aren't evenly and randomly spaced around zero mean, there's clear trend from top left to bottom right.

2.7 Adding the species into the mix

- Not adding an interaction term yet
- Model will look like:

$$y_i = \alpha + \beta x_i + \gamma_2 z_{i2} + \gamma_3 z_{i3} + \epsilon_i, \text{ for } i = 1, 2, ..., 150$$

This can be simplified to:
$y_i = (\alpha + \gamma_j) + \beta x_i + \epsilon_i$, for $i = 1, 2, ..., 150, j = 2, 3$.
Our control group, baseline case, for the **setosa** species, since $s < v$.
This control group **DOES NOT HAVE A γ PARAMETER, OTHERWISE THE MODEL WOULD BE OVER PARAMETERISED!**

$$y_i = \alpha + \beta x_i + \epsilon_i$$

When species = **versicolor**, $z_2 = 1, z_3 = 0$, and therefore our model becomes:

$$y_i = (\alpha + \gamma_2) + \beta x_i + \epsilon_i,$$

And when species = **virginica**, the model becomes:

$$y_i = (\alpha + \gamma_3) + \beta x_i + \epsilon_i.$$

This helps with our interpretation of the 2nd and 3rd levels of the factor w.r.t. the **setosa** baseline.
Significance in the parameters for these 2nd and 3rd factors means that there's strong evidence to suggest there should be separate intercepts for those levels w.r.t. the baseline level.

```
In [94]: lm.iris.species.no.interaction <- lm(petalL ~ sepalL + species, data=iris)

         summary(lm.iris.species.no.interaction)
```

```

Call:
lm(formula = petalL ~ sepalL + species, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.76390 -0.17875  0.00716  0.17461  0.79954 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)       -1.70234    0.23013  -7.397 1.01e-11 ***
sepalL             0.63211    0.04527  13.962  < 2e-16 ***
speciesversicolor  2.21014    0.07047  31.362  < 2e-16 ***
speciesvirginica   3.09000    0.09123  33.870  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2826 on 146 degrees of freedom
Multiple R-squared:  0.9749,Adjusted R-squared:  0.9744 
F-statistic:  1890 on 3 and 146 DF,  p-value: < 2.2e-16
```

2.8 Commentary on LM output:

- F-stat highly significant: reject the null hypothesis that all explanatory variables have parameters = zero (i.e. that none of the explanatory variables explain the variation in the dependent variable)

- As expected, sepal length remains a highly significant explanatory variable.

- Both species levels from the species factor are highly significant, providing strong evidence that versicolor and virginica species should have their own intercepts with respect to the baseline species of setosa, that has intercept α.

- The R^2 has increased dramatically from the model without the species factor, from 0.76 to 0.975. This dramatic increase in this goodness-of-fit statistic provides us with weight to endorse this as a good fitting model, and to take this particular model to the next stage of validation: model adequecy testing.

- The Residual standard error, $s = \sqrt{s^2} = \sqrt{MS_R}$, has decreased from 0.8678 to 0.2826, which is another goodness-of-fit statistic that is going in the right direction.

2.9 Let's plot those three different intercept models:

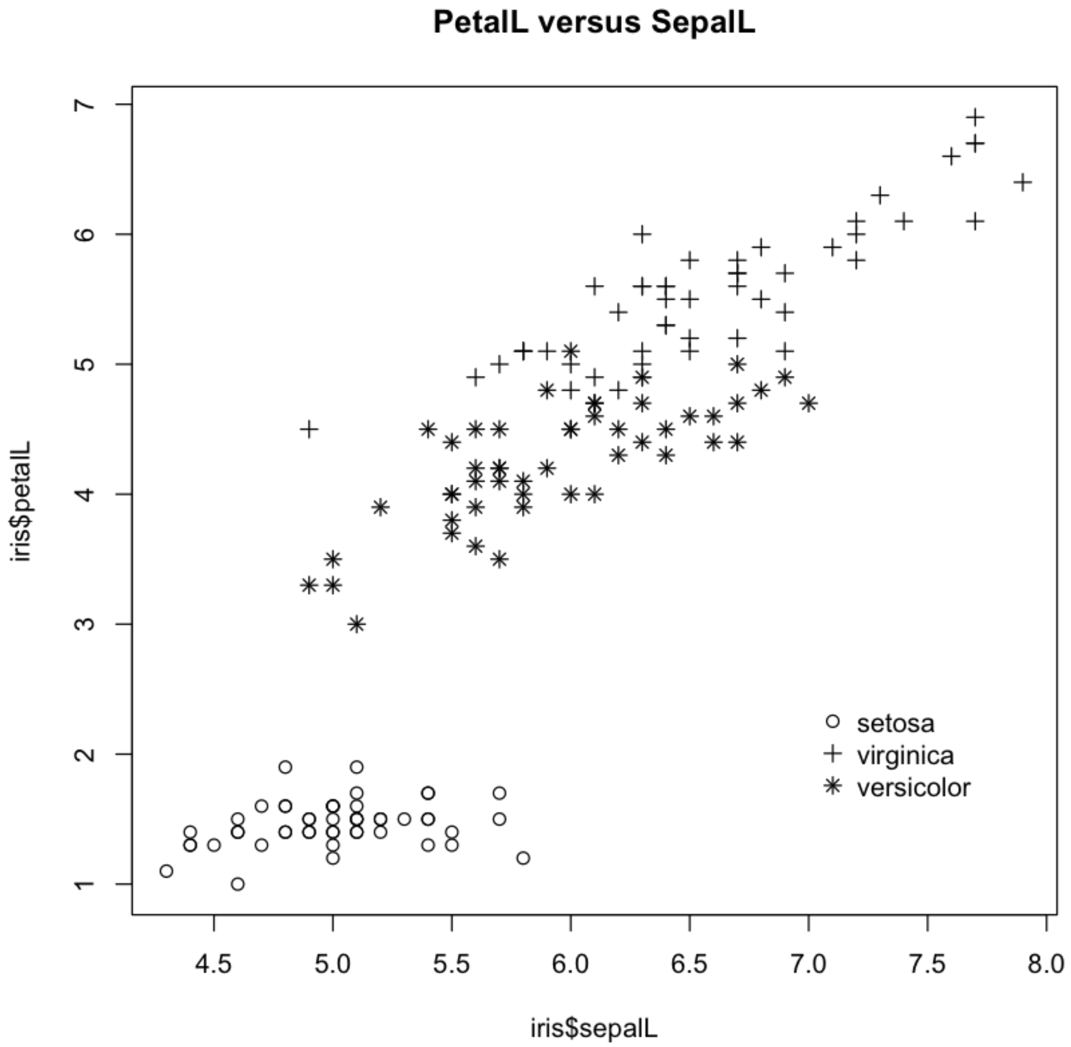
```
In [95]: iris[1:5,]
```

sepalL	sepalW	petalL	petalW	species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa

```
In [96]: lm.iris.species.no.interaction
```

```


Call:
lm(formula = petalL ~ sepalL + species, data = iris)

Coefficients:
      (Intercept)             sepalL  speciesversicolor   speciesvirginica
          -1.7023             0.6321             2.2101             3.0900
```

```
In [106]: plot(iris$sepalL, iris$petalL, type = "n", main = "PetalL versus SepalL")
          points(iris$sepalL[iris$species == "setosa"], iris$petalL[iris$species == "setosa"],
          points(iris$sepalL[iris$species == "virginica"], iris$petalL[iris$species == "virgin
          points(iris$sepalL[iris$species == "versicolor"], iris$petalL[iris$species == "versic
          legend(7, 2.5, c("setosa", "virginica", "versicolor"), pch =c(1, 3, 8), bty="n")

          ld <- seq(0, 7, 0.1)
```



2.10 Okay this is a nightmare to try and plot, move on to the third model with the interaction terms:

2.10.1 The model we're fitting here is:

$$y_i = \alpha + \beta x_{i1} + \gamma_2 z_{i2} + \gamma_3 z_{i3} + \delta_2 x_{i1} zi2 + \delta_3 x_{i1} z_{i3} + \epsilon_i, \, i = 1, 2, ..., n.$$

$$z_{i1} = 0 \text{ for all } i.$$

```
In [107]: lm.iris.interactions <- lm(petalL ~ sepalL +  species + sepalL*species, data=iris)

          summary(lm.iris.interactions)
```

```
Call:
lm(formula = petalL ~ sepalL + species + sepalL * species, data = iris)

Residuals:
     Min       1Q   Median       3Q      Max
-0.68611 -0.13442 -0.00856  0.15966  0.79607

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)                0.8031     0.5310   1.512    0.133
sepalL                     0.1316     0.1058   1.244    0.216
speciesversicolor         -0.6179     0.6837  -0.904    0.368
speciesvirginica          -0.1926     0.6578  -0.293    0.770
sepalL:speciesversicolor   0.5548     0.1281   4.330 2.78e-05 ***
sepalL:speciesvirginica    0.6184     0.1210   5.111 1.00e-06 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.2611 on 144 degrees of freedom
Multiple R-squared:  0.9789,Adjusted R-squared:  0.9781
F-statistic:  1333 on 5 and 144 DF,  p-value: < 2.2e-16
```

2.11 Summary:

- Wow, our adjusted R-squared has gone even higher, which is a goodness-of-fit statistic that is showing that our new model is superior to the previous one without interaction terms
- Very interesting, the two interaction terms are both highly significant, whereas the individual terms are not.
- The standard practice here is that we leave in the non-interacting terms in the model.

  Let's look at some model adequecy plots (in the notes).

```
In [ ]:
```