STAT2401: Analysis of Experiments
Computer Lab **11 Part 1**

Contents

1 Exercise 1: Analysis of covariance — wheat data

The following data gives the results of an experiment that examined the yield of wheat plant when two treatments were applied to each of 5 plots. However, the plots each had different numbers of plants on and hence this would have an impact on the plot yield. We will carry out an ANCOVA to make adjustments for the numbers of plants per plot.

The variables are the following:

`Treat`	the treatment applied to the plot
`yield`	the response yield per plot
`plants`	the number of plants per plot

First fit an ANOVA comparing just treatments. Read these data into R, and store them as a data frame `wheat`.

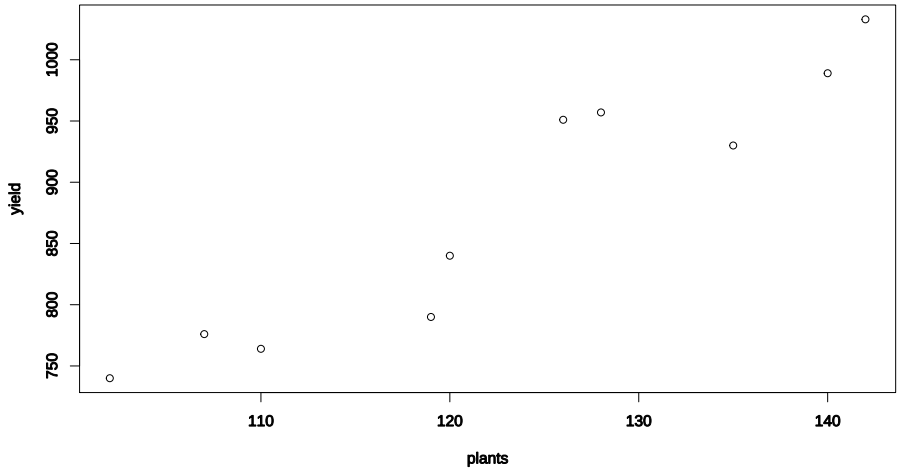
Reading in and visualising the data:

```
wheat = read.csv(file="wheat.csv",header=T)
str(wheat)
```

```
## 'data.frame': 10 obs. of  3 variables:
##  $ Treat : int  1 1 1 1 1 2 2 2 2 2
##  $ yield : int  951 957 776 1033 840 930 790 764 989 740
##  $ plants: int  126 128 107 142 120 135 119 110 140 102
```

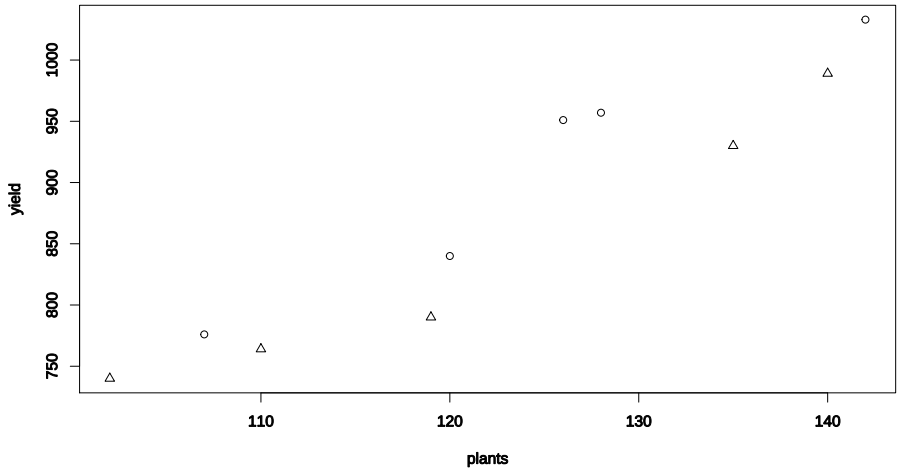
Have a look at the data. Let's plot the yield by the number of plants per plot.

```
plot(yield~plants, data=wheat)
```



Adding symbols to represent the different treatment:

```
plot(yield~plants, pch=Treat, data=wheat)
```



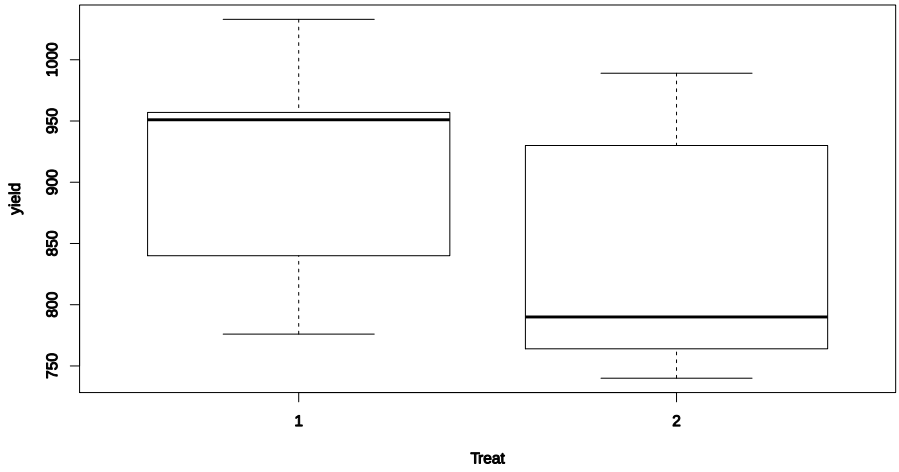
Treatment differences appear to be difficult to detect however a first impression would indicate that treatment 2 is below treatment 1 for any given value of plants.

There appears to be a linear trend. As the number of plants per pot increases the yield increases too.

Now fit a standard one way ANOVA model comparing treatments. First visualise the comparison. Boxplots might not be that useful with small data so we could just plot the whole data instead.

Fitting the one way ANOVA

```
  boxplot(yield~Treat, data=wheat)
anova(lm(yield~factor(Treat), data=wheat))
## Analysis of Variance Table
##
## Response: yield
##               Df Sum Sq Mean Sq F value Pr(>F)
## factor(Treat)  1  11834   11834  1.0471 0.3361
## Residuals      8  90408   11301
```



No differnces from the plot and none from the ANOVA. However, if anything treatment 1 appears to be slightly larger than treatment 2.

Let us try now an ANCOVA.

```
wheat.acov=lm(yield~factor(Treat) + plants , data=wheat)
summary(wheat.acov)
```

```
##
## Call:
## lm(formula = yield ~ factor(Treat) + plants, data = wheat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -38.840 -10.687  -0.442  19.483  33.302
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     29.4532    87.7111   0.336   0.7469
## factor(Treat)2 -44.7340    18.2606  -2.450   0.0441 *
## plants           7.0782     0.6964  10.164 1.92e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28.63 on 7 degrees of freedom
## Multiple R-squared:  0.9439,Adjusted R-squared:  0.9279
## F-statistic: 58.87 on 2 and 7 DF,  p-value: 4.186e-05
```

Checking the parallel lines assumption:

```
wheat.np=lm(yield~factor(Treat)*plants , data=wheat)
summary(wheat.np)
```

```
##
## Call:
## lm(formula = yield ~ factor(Treat) * plants, data = wheat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.992 -11.440  -1.316  20.957  28.702
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)            -58.485    144.992  -0.403 0.700658
## factor(Treat)2          96.293    182.865   0.527 0.617375
## plants                   7.784      1.159   6.717 0.000529 ***
## factor(Treat)2:plants   -1.144      1.475  -0.775 0.467598
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29.48 on 6 degrees of freedom
## Multiple R-squared:  0.949,Adjusted R-squared:  0.9235
## F-statistic: 37.21 on 3 and 6 DF,  p-value: 0.0002846
```

the interaction term in the non-parrallel lines model is not significant indiciting a parrallel lines regression model is sufficient in this instance.

2 Exercise 2: Linear Modelling for Fishers Iris data

This question analyses the maybe most famous data set in statistics. Type `help(iris)` in R to get more information.

Let's have a look at the data to start with. Type `iris`

- Fit the following models in order to explain the response variable `Sepal.Length` based on the information of `Petal.Length`:
  - $M1$, a single linear regression for all observations (i.e. intercept and slope not dependent on species)
  - $M2$, parallel regressions for observations from each species (i.e. regressions have the same slope but the intercept varies from species to species).
  - $M3$, separate regression for observations from each species (i.e. regressions have intercept and slope that varies from species to species).

Fisher's Iris data is a well used dataset in statistical education and is pre-loaded into R. To see this data type `iris` and to get more information on the dataset type `help(iris)`

We want to fit 3 models to this data.

```
M1=lm(Sepal.Length~Petal.Length, data=iris)
summary(M1)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.24675 -0.29657 -0.01515  0.27676  1.00269
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.30660    0.07839   54.94   <2e-16 ***
## Petal.Length  0.40892    0.01889   21.65   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4071 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
```

```
M2=lm(Sepal.Length~Petal.Length+ Species, data=iris)
summary(M2)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length + Species, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.75310 -0.23142 -0.00081  0.23085  1.03100
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.68353    0.10610  34.719  < 2e-16 ***
## Petal.Length       0.90456    0.06479  13.962  < 2e-16 ***
## Speciesversicolor -1.60097    0.19347  -8.275 7.37e-14 ***
## Speciesvirginica  -2.11767    0.27346  -7.744 1.48e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.338 on 146 degrees of freedom
## Multiple R-squared:  0.8367,Adjusted R-squared:  0.8334
## F-statistic: 249.4 on 3 and 146 DF,  p-value: < 2.2e-16
```

```
M3=lm(Sepal.Length~Petal.Length*Species, data=iris)
summary(M3)
```

```
##
## Call:
## lm(formula = Sepal.Length ~ Petal.Length * Species, data = iris)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.73479 -0.22785 -0.03132  0.24375  0.93608
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      4.2132     0.4074  10.341  < 2e-16 ***
## Petal.Length                     0.5423     0.2768   1.959  0.05200 .
## Speciesversicolor               -1.8056     0.5984  -3.017  0.00302 **
## Speciesvirginica                -3.1535     0.6341  -4.973 1.85e-06 ***
## Petal.Length:Speciesversicolor   0.2860     0.2951   0.969  0.33405
## Petal.Length:Speciesvirginica    0.4534     0.2901   1.563  0.12029
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3365 on 144 degrees of freedom
## Multiple R-squared:  0.8405,Adjusted R-squared:  0.8349
## F-statistic: 151.7 on 5 and 144 DF,  p-value: < 2.2e-16
```

Use F tests to select the most appropriate model from $M1$, $M2$, and $M3$, working at a 5% significance level. Explain your reasoning clearly, and include the P-values that you obtain for your tests. Remember you can use the `anova(mod1,mod2)` function to compare two nested models, with $mod1$ being the smaller of the two models.

Look briefly at the model diagnostics for your final model. Do you see any glaring problems?

To see if the models are different from eachother we can use the `anova()` function.

```
anova(M1,M2)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: Sepal.Length ~ Petal.Length
## Model 2: Sepal.Length ~ Petal.Length + Species
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    148 24.525
## 2    146 16.682  2    7.8434 34.323 6.053e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

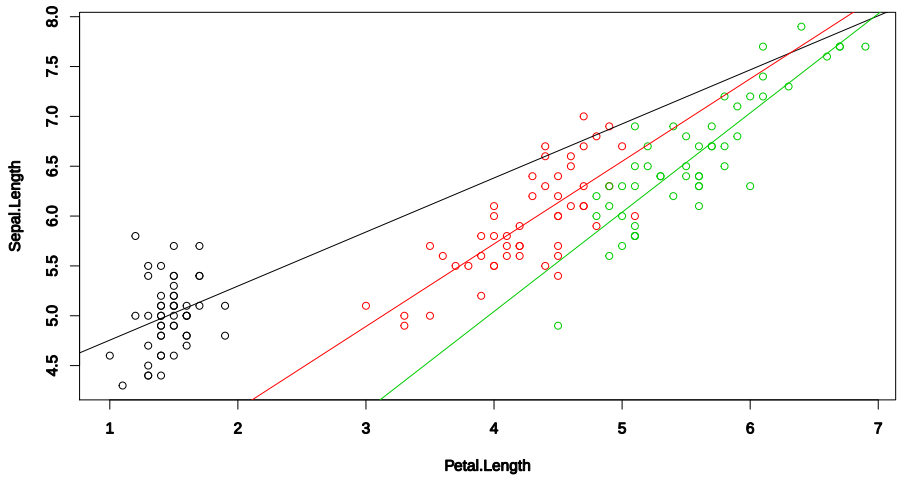
```
anova(M2,M3)
```

```
## Analysis of Variance Table
##
## Model 1: Sepal.Length ~ Petal.Length + Species
## Model 2: Sepal.Length ~ Petal.Length * Species
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    146 16.682
## 2    144 16.301  2   0.38098 1.6828 0.1895
```

These results would indicate that we need separate intercepts but that one slope is sufficient.

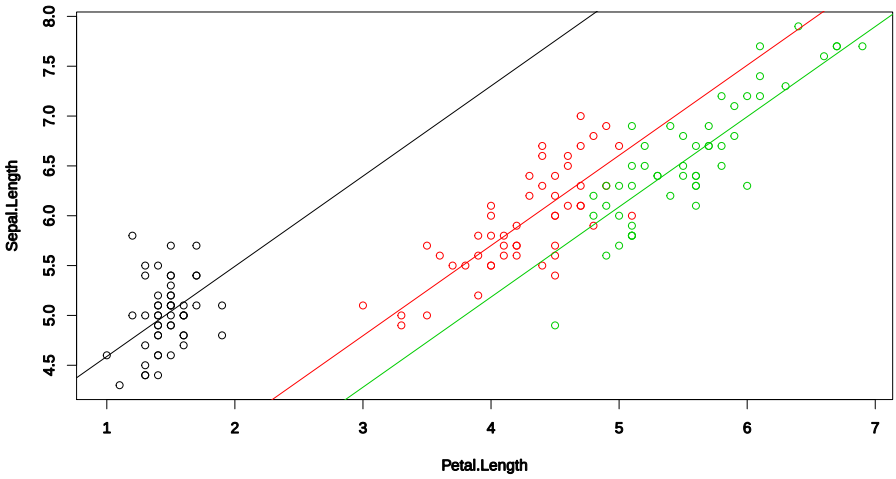
Visualising this:

```
plot(Sepal.Length ~ Petal.Length, data=iris, col=as.numeric(Species))
abline(coefficients(M3)[1], coefficients(M3)[2], col=1)
abline(coefficients(M3)[1]+coefficients(M3)[3], coefficients(M3)[2]+coefficients(M3)[5], col=2)
abline(coefficients(M3)[1]+coefficients(M3)[4], coefficients(M3)[2]+coefficients(M3)[6], col=3)
```



Whilst these slopes look marginally different they are not sufficiently so to make the unequal slopes model statsitically significanlty different to the parrallel lines regression model.

```
plot(Sepal.Length ~ Petal.Length, data=iris, col=as.numeric(Species))
abline(coefficients(M2)[1],coefficients(M2)[2],col=1)
abline(coefficients(M2)[1]+coefficients(M2)[3],coefficients(M2)[2], col=2)
abline(coefficients(M2)[1]+coefficients(M2)[4],coefficients(M2)[2], col=3)
```


Sepal.Length
4.5
5.0
5.5
6.0
6.5
7.0
7.5
8.0
Petal.Length
1
2
3
4
5
6
7

STAT2401: Analysis of Experiments
Computer Lab **Week 11 Part 2**

Contents

1 Exercise 1: Polynomial Regression for `reef` data

This lab uses functions from the R package **`Rcmdr`**. To be able to use these functions, you need to open the package using **`library(Rcmdr)`**. This will also open up the **`RCommander`** GUI interface. We will not be using this interface for this unit, so you can close the interface straight away.

```
library("Rcmdr")
```

This exercise is concerned with the density of the coral **`Porites lobata`** on the Great Barrier Reef. The following variables are recorded.

`Reef`	Name of reef.
`Distance`	Distance to the Australian shore (in km).
`Density`	Coral head density (in g/cm^3).

The data are taken from the following paper:

Risk, M. J., and Sammarco, P. W. (1981). Cross-shelf trends in skeletal density of the massive coral Porites lobata from the Great Barrier Reef. *Marine Ecology Progress Series* **69**, 195-200.

The authors fitted a second order polynomial regression (i.e. a quadratic regression) of **`Density`** on **`Distance`**. We are going to fit such a model using R, and store the result as **`reef.lm.2`**. Also, fit a fourth order polynomial regression model, **`reef.lm.4`**. Perform an F-test to see which of these models is preferable. (Hint: use the **`anova()`** command.)

First we set the working directory, load the data, look at its structure (the R command **`str()`** is very useful to obtain an idea about what is stored in an object) and look at the data.

```
reef = read.table(file="reef.txt",header=T)
str(reef)
```

```
## 'data.frame': 27 obs. of  3 variables:
##  $ Reef    : Factor w/ 9 levels "AlmaBay","BowdenReef",..: 6 6 6 1 1 1 8 8 8 9 ...
##  $ Distance: num  3.5 3.5 3.5 14.3 14.3 14.3 15.4 15.4 15.4 15.9 ...
##  $ Density : num  1.34 1.22 1.31 1.05 1.08 ...
```

```
reef
```

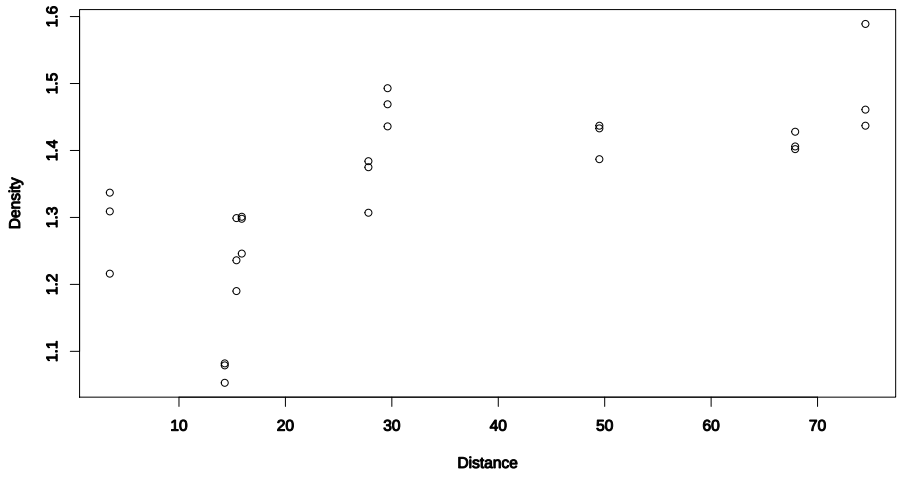
```
##                Reef Distance Density
## 1        MiddleReef      3.5   1.337
## 2        MiddleReef      3.5   1.216
```

```
## 3        MiddleReef      3.5   1.309
## 4           AlmaBay     14.3   1.053
## 5           AlmaBay     14.3   1.082
## 6           AlmaBay     14.3   1.079
## 7         OrpheusIs     15.4   1.236
## 8         OrpheusIs     15.4   1.190
## 9         OrpheusIs     15.4   1.299
## 10      PandoraReef     15.9   1.246
## 11      PandoraReef     15.9   1.298
## 12      PandoraReef     15.9   1.301
## 13      GreatPalmIs     27.8   1.375
## 14      GreatPalmIs     27.8   1.384
## 15      GreatPalmIs     27.8   1.307
## 16    MorindaShoals     29.6   1.436
## 17    MorindaShoals     29.6   1.493
## 18    MorindaShoals     29.6   1.469
## 19 LittleBroadhurst     49.5   1.387
## 20 LittleBroadhurst     49.5   1.437
## 21 LittleBroadhurst     49.5   1.433
## 22       BowdenReef     67.9   1.406
## 23       BowdenReef     67.9   1.402
## 24       BowdenReef     67.9   1.428
## 25         GrubReef     74.5   1.437
## 26         GrubReef     74.5   1.589
## 27         GrubReef     74.5   1.461
```

Take a look at the data. You should observe that multiple observations have been taken at each reef.

As always it is good to visualise the data:

```
plot(Density~Distance, data=reef)
```



Initial impressions should be that there something going on between **Density** and **Distance** but some strangeness in the data may suggest more than a linear or even quadratic relationship.

Let's investigate this:

We fit the model of degree 2 and degree 4 to start with:

```
reef.lm.2 = lm(Density~Distance + I(Distance^2), data=reef)
summary(reef.lm.2)
```

```
##
## Call:
```

```
## lm(formula = Density ~ Distance + I(Distance^2), data = reef)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20988 -0.03427  0.01100  0.04247  0.14731
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.167e+00  5.556e-02  20.995   <2e-16 ***
## Distance       7.380e-03  3.678e-03   2.006   0.0562 .
## I(Distance^2) -4.482e-05  4.447e-05  -1.008   0.3237
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0981 on 24 degrees of freedom
## Multiple R-squared:  0.4935,Adjusted R-squared:  0.4513
## F-statistic: 11.69 on 2 and 24 DF,  p-value: 0.0002851
```

```
reef.lm.4 = lm(Density~Distance + I(Distance^2) + I(Distance^3) +
                 I(Distance^4), data=reef)
summary(reef.lm.4)
```

```
##
## Call:
## lm(formula = Density ~ Distance + I(Distance^2) + I(Distance^3) +
##     I(Distance^4), data = reef)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14518 -0.05562  0.02719  0.05763  0.09110
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.403e+00  7.495e-02  18.721 5.28e-15 ***
## Distance      -4.722e-02  1.334e-02  -3.539 0.001843 **
## I(Distance^2)  3.127e-03  7.411e-04   4.219 0.000353 ***
## I(Distance^3) -6.373e-05  1.503e-05  -4.240 0.000336 ***
## I(Distance^4)  4.099e-07  9.911e-08   4.136 0.000433 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07565 on 22 degrees of freedom
## Multiple R-squared:  0.7239,Adjusted R-squared:  0.6737
## F-statistic: 14.42 on 4 and 22 DF,  p-value: 6.375e-06
```

We note several things. First, in R, we make use of the `I(.)` terminology to specify higher degree terms. This terminology is used to inhibit the interpretation of operators such as "+", "−", "*" and "^" as formula operators in model statements and allows us to create our higher degree terms. Second, when we fitted the degree 4 polynomial regression model we not only added the degree 4 term but added the degree 3 term too. This should almost always be done. Higher degree polynomials should always include all lower degree terms. Finally, we note that the quadratic term in `reef.lm.2` is not significant.

If we want to compare these two models we can use the `anova` command. This command tests the two specified models against each other to see if there is a significant difference.

```
anova(reef.lm.2, reef.lm.4)
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Distance + I(Distance^2)
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4)
```

```
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     24 0.23096
## 2     22 0.12591  2   0.10506 9.1782 0.001264 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

This tells us that the additional sums of squares explained by the degree 4 model is 0.10506, at a cost of 2 extra degrees of freedom. The test to see whether this amount is sufficieintly large to declare the degree 4 model better than the degree 2 model is the given F test. In this case the p-value is given by 0.001264, highly significant and hence we conclude the difference is sufficent to declare the larger model more effective.

We can consider all models in this way. Sometimes we would start with the highest polynomial model we are willing to consider. In this case we look at everything up to degree 6:

```
reef.lm.6 = lm(Density~Distance + I(Distance^2) + I(Distance^3) + I(Distance^4) +
                 I(Distance^5) + I(Distance^6), data=reef)
summary(reef.lm.6)
```

```
##
## Call:
## lm(formula = Density ~ Distance + I(Distance^2) + I(Distance^3) +
##     I(Distance^4) + I(Distance^5) + I(Distance^6), data = reef)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.129934 -0.055083  0.009957  0.045142  0.096273
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.546e+00  6.025e-01   4.227 0.000414 ***
## Distance      -5.354e-01  2.597e-01  -2.061 0.052504 .
## I(Distance^2)  5.892e-02  3.013e-02   1.956 0.064601 .
## I(Distance^3) -2.736e-03  1.466e-03  -1.867 0.076630 .
## I(Distance^4)  6.155e-05  3.404e-05   1.808 0.085594 .
## I(Distance^5) -6.626e-07  3.740e-07  -1.771 0.091728 .
## I(Distance^6)  2.735e-09  1.565e-09   1.748 0.095716 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06894 on 20 degrees of freedom
## Multiple R-squared:  0.7916,Adjusted R-squared:  0.729
## F-statistic: 12.66 on 6 and 20 DF,  p-value: 6.718e-06
```

```
reef.lm.5 = lm(Density~Distance + I(Distance^2) + I(Distance^3) + I(Distance^4) +
                 I(Distance^5) , data=reef)
summary(reef.lm.5)
```

```
##
## Call:
## lm(formula = Density ~ Distance + I(Distance^2) + I(Distance^3) +
##     I(Distance^4) + I(Distance^5), data = reef)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.128157 -0.033673  0.005721  0.040669  0.102506
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.504e+00  9.151e-02  16.436 1.82e-13 ***
## Distance      -8.303e-02  2.392e-02  -3.471  0.00228 **
```

```
## I(Distance^2)  6.347e-03  1.953e-03   3.249  0.00384 **
## I(Distance^3) -1.760e-04  6.508e-05  -2.705  0.01328 *
## I(Distance^4)  2.063e-06  9.393e-07   2.196  0.03944 *
## I(Distance^5) -8.635e-09  4.882e-09  -1.769  0.09145 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07224 on 21 degrees of freedom
## Multiple R-squared:  0.7597,Adjusted R-squared:  0.7025
## F-statistic: 13.28 on 5 and 21 DF,  p-value: 6.578e-06
```

```
reef.lm.3 = lm(Density~Distance + I(Distance^2) + I(Distance^3), data=reef)
summary(reef.lm.3)
```

```
##
## Call:
## lm(formula = Density ~ Distance + I(Distance^2) + I(Distance^3),
##     data = reef)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20016 -0.05433 -0.01111  0.05040  0.15270
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.212e+00  7.686e-02  15.765 8.03e-14 ***
## Distance       6.941e-04  8.632e-03   0.080    0.937
## I(Distance^2)  1.826e-04  2.690e-04   0.679    0.504
## I(Distance^3) -2.004e-06  2.338e-06  -0.857    0.400
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09865 on 23 degrees of freedom
## Multiple R-squared:  0.5092,Adjusted R-squared:  0.4452
## F-statistic: 7.953 on 3 and 23 DF,  p-value: 0.0008158
```

```
reef.lm.1 = lm(Density~Distance, data=reef)
summary(reef.lm.1)
```

```
##
## Call:
## lm(formula = Density ~ Distance, data = reef)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.212753 -0.047247 -0.009136  0.062975  0.169705
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.2119729  0.0324376  37.363  < 2e-16 ***
## Distance    0.0037609  0.0007954   4.728 7.54e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09813 on 25 degrees of freedom
## Multiple R-squared:  0.4721,Adjusted R-squared:  0.4509
## F-statistic: 22.35 on 1 and 25 DF,  p-value: 7.54e-05
```

We note for the degree 6 model the highest degree polynomial term is not significant ($P = 0.09572$). Hence we can remove this. Of course the t-test looking at the summary of the model coefficients for the degree 6 term is equivalent to that if we compared two models with and without the degree 6 term. I.e.

```
anova(reef.lm.5, reef.lm.6)
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4) +
##     I(Distance^5)
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4) +
##     I(Distance^5) + I(Distance^6)
##   Res.Df      RSS Df Sum of Sq      F  Pr(>F)
## 1     21 0.109582
## 2     20 0.095052  1   0.01453 3.0572 0.09572 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note the p-value is the same here p-value $= 0.09572$

Given the degree 6 term is not needed we need to work out whether or not the degree 5 term is needed. Again either looking at the coefficents and the p-value associated with this term in model `reef.lm.5`. Again this is not significant suggesting a degree 4 polynomial would be just as good.

We know previously that the degree 4 polynomial has a signficant degree 4 term so we could argue that this is the best place to stop. However, for completeness we may wish to compare it to all the lower degree models just in case. The only model we haven't compared this to is the linear model. Let's

```
anova(reef.lm.1, reef.lm.4)
```

```
## Analysis of Variance Table
##
## Model 1: Density ~ Distance
## Model 2: Density ~ Distance + I(Distance^2) + I(Distance^3) + I(Distance^4)
##   Res.Df     RSS Df Sum of Sq     F   Pr(>F)
## 1     25 0.24074
## 2     22 0.12591  3   0.11483 6.688 0.002234 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The result of this test indicates the degree 4 is better than the straight line (degree 1) polynomial regression model.

Hence we stick with our 'best' model as `reef.lm.4`.

To visualise the fitted model we can make use of the predict function.

```
x = 1:80
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
## [47] 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69
## [70] 70 71 72 73 74 75 76 77 78 79 80
```

```
y = predict(reef.lm.4,newdata=data.frame(Distance=x))
y
```

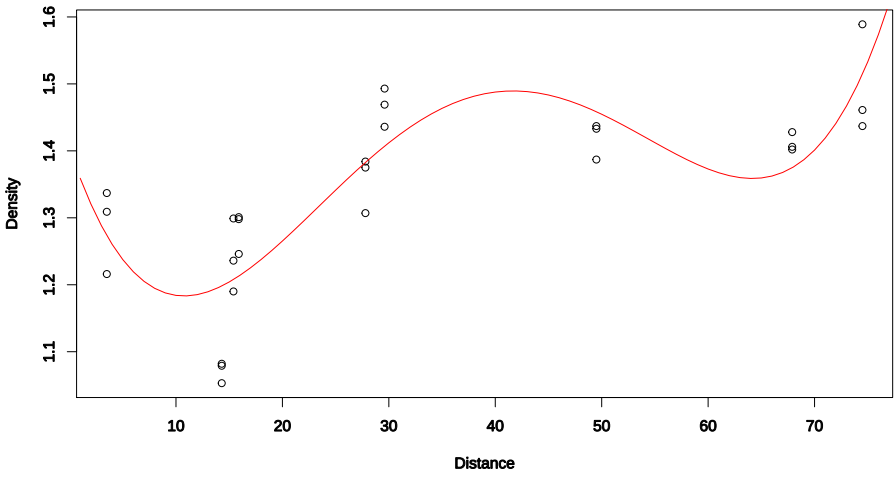
```
##        1        2        3        4        5        6        7        8
## 1.359098 1.320818 1.288047 1.260428 1.237612 1.219262 1.205049 1.194655
##        9       10       11       12       13       14       15       16
## 1.187771 1.184098 1.183347 1.185240 1.189507 1.195888 1.204134 1.214006
##       17       18       19       20       21       22       23       24
## 1.225273 1.237715 1.251123 1.265295 1.280041 1.295181 1.310543 1.325967
##       25       26       27       28       29       30       31       32
## 1.341301 1.356404 1.371144 1.385401 1.399061 1.412023 1.424194 1.435494
##       33       34       35       36       37       38       39       40
## 1.445848 1.455194 1.463480 1.470662 1.476707 1.481593 1.485305 1.487840
```

```
##       41       42       43       44       45       46       47       48
## 1.489204 1.489413 1.488493 1.486480 1.483419 1.479366 1.474386 1.468554
##       49       50       51       52       53       54       55       56
## 1.461956 1.454684 1.446846 1.438553 1.429932 1.421116 1.412249 1.403485
##       57       58       59       60       61       62       63       64
## 1.394988 1.386930 1.379496 1.372878 1.367279 1.362912 1.360000 1.358776
##       65       66       67       68       69       70       71       72
## 1.359481 1.362368 1.367698 1.375745 1.386788 1.401120 1.419043 1.440867
##       73       74       75       76       77       78       79       80
## 1.466913 1.497512 1.533006 1.573744 1.620086 1.672404 1.731077 1.796495
```

Notice now the x is just a sequence from 1 to 80. These are all the values of x we want to predict y at. The ys are the predicted values from the `reef.lm.4`. Note: we could have followed the instructions in the lab sheet and created `Distance.2` etc but in this instance it is not necessary as R is clever enough to work it out.

Plotting the data again with the fitted lines on is easy:

```
plot(Density~Distance, data=reef)
lines(x,y, col="red")
```



2 Exercise 2: Polynomial Regression for `baseball` data

In 1954 Branch Rickey wrote an article for Life Magazine entitled Goodbye to some old baseball ideas. He criticized some traditional baseball statistics and proposed some of his own that he thought more useful. For individual hitting Rickey proposed the sum of on-base average (OBA) and extra-base power (EBP). A main question is whether OBA and EBP are distinct components of hitting ability.

We would like to determine an appropriate polynomial regression model of OBA (the response) on EBP. We should consider models up to degree 6 (i.e. with the highest power of EBP being at most 6).

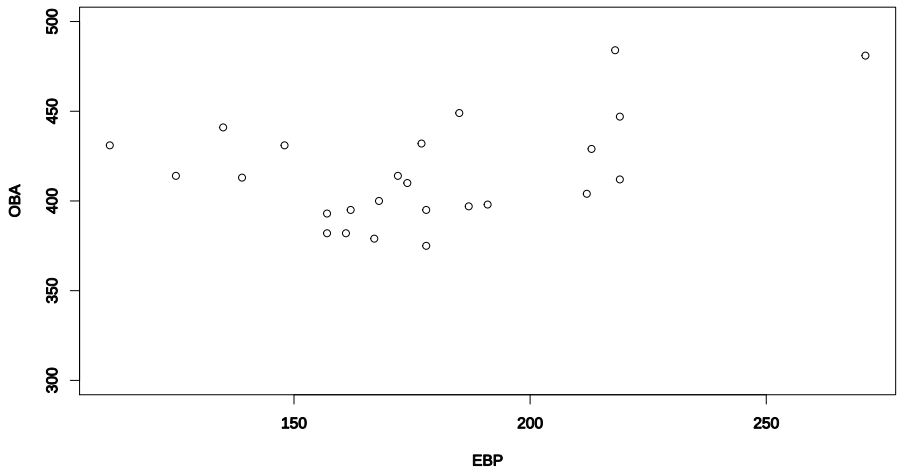
In this exercise we look at the relationship between EBP (predictor) and OBA (response) from the baseball data. First we read in the data and look at it.

```
baseball = read.table(file="baseball.txt",header=T)
baseball
```

```
##    OBA EBP
## 1  481 271
## 2  484 218
## 3  447 219
```

```
## 4  429 213
## 5  449 185
## 6  412 219
## 7  404 212
## 8  432 177
## 9  398 191
## 10 414 172
## 11 410 174
## 12 397 187
## 13 431 148
## 14 441 135
## 15 395 178
## 16 400 168
## 17 395 162
## 18 375 178
## 19 413 139
## 20 393 157
## 21 379 167
## 22 382 161
## 23 431 111
## 24 414 125
## 25 382 157
```

```
plot(OBA~EBP, data=baseball, ylim=c(300,500))
```



We see the data is pretty messy and predicting the response OBA from EBP may not be the most straight forward of tasks. However, we consdier all polynomial regression models up to degree 6. First we fit these models:

```
#straight line
fm1=lm(OBA~EBP, data=baseball)
summary(fm1)
```

```
##
## Call:
## lm(formula = OBA ~ EBP, data = baseball)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -40.876 -21.962  -4.505  17.069  54.412
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 354.8604    28.3922  12.499 9.77e-12 ***
## EBP           0.3428     0.1575   2.176     0.04 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.07 on 23 degrees of freedom
## Multiple R-squared:  0.1708,Adjusted R-squared:  0.1347
## F-statistic: 4.737 on 1 and 23 DF,  p-value: 0.04005
```

```
#quadratic
fm2=lm(OBA~EBP+I(EBP^2), data=baseball)
#cubic
fm3=lm(OBA~EBP+I(EBP^2)+I(EBP^3), data=baseball)
#quartic
fm4=lm(OBA~EBP+I(EBP^2)+I(EBP^3)+I(EBP^4), data=baseball)
#quintic
fm5=lm(OBA~EBP+I(EBP^2)+I(EBP^3)+I(EBP^4)+I(EBP^5), data=baseball)
# sextic (degree 6)
fm6=lm(OBA~EBP+I(EBP^2)+I(EBP^3)+I(EBP^4)+I(EBP^5)+ I(EBP^6), data=baseball)
```

In order to compare these models we can use the **anova** command to compare two models or simply look at the coefficients on the higest degree polynomial. We will use the former. It is usually sensible to start with the highest degree polynomial:

```
# comparing the sextic to the quintic.
anova(fm5, fm6)
```

```
## Analysis of Variance Table
##
## Model 1: OBA ~ EBP + I(EBP^2) + I(EBP^3) + I(EBP^4) + I(EBP^5)
## Model 2: OBA ~ EBP + I(EBP^2) + I(EBP^3) + I(EBP^4) + I(EBP^5) + I(EBP^6)
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     19 10262
## 2     18 10184  1    77.982 0.1378 0.7148
```

```
# no significant difference let's continue
```

We could of course have looked at the p-value for the degree 6 term from the degree 6 polynomial model:

```
summary(fm6)
```

```
##
## Call:
## lm(formula = OBA ~ EBP + I(EBP^2) + I(EBP^3) + I(EBP^4) + I(EBP^5) +
##     I(EBP^6), data = baseball)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.428 -16.450  -1.945   9.169  44.511
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.670e+04  8.256e+04  -0.444    0.662
## EBP          1.298e+03  2.981e+03   0.435    0.669
## I(EBP^2)    -1.856e+01  4.414e+01  -0.421    0.679
## I(EBP^3)     1.393e-01  3.429e-01   0.406    0.689
## I(EBP^4)    -5.793e-04  1.474e-03  -0.393    0.699
## I(EBP^5)     1.268e-06  3.326e-06   0.381    0.707
## I(EBP^6)    -1.142e-09  3.075e-09  -0.371    0.715
##
## Residual standard error: 23.79 on 18 degrees of freedom
```

```
## Multiple R-squared:  0.4988,Adjusted R-squared:  0.3318
## F-statistic: 2.986 on 6 and 18 DF,  p-value: 0.03328
```

Note this p-value is 0.715 which is the same as using the **anova** comparison. So we continue in the same fashion until we can not reduce the polynomial further:

```
# comparing the quintic to the quartic
anova(fm4, fm5)
```

```
## Analysis of Variance Table
##
## Model 1: OBA ~ EBP + I(EBP^2) + I(EBP^3) + I(EBP^4)
## Model 2: OBA ~ EBP + I(EBP^2) + I(EBP^3) + I(EBP^4) + I(EBP^5)
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     20 10379
## 2     19 10262  1    117.12 0.2168 0.6467
```

```
# no significant difference let's continue
# comparing the quartic to the cubic
anova(fm3, fm4)
```

```
## Analysis of Variance Table
##
## Model 1: OBA ~ EBP + I(EBP^2) + I(EBP^3)
## Model 2: OBA ~ EBP + I(EBP^2) + I(EBP^3) + I(EBP^4)
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     21 10900
## 2     20 10379  1    521.66 1.0052  0.328
```

```
# no significant difference let's continue
# comparing the cubic to the quadratic
anova(fm2, fm3)
```

```
## Analysis of Variance Table
##
## Model 1: OBA ~ EBP + I(EBP^2)
## Model 2: OBA ~ EBP + I(EBP^2) + I(EBP^3)
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     22 11682
## 2     21 10900  1    780.98 1.5046 0.2335
```

```
# no significant difference let's continue
# comparing the quadratic to the linear model
anova(fm1, fm2)
```

```
## Analysis of Variance Table
##
## Model 1: OBA ~ EBP
## Model 2: OBA ~ EBP + I(EBP^2)
##   Res.Df   RSS Df Sum of Sq      F   Pr(>F)
## 1     23 16850
## 2     22 11682  1    5168.6 9.7341 0.004987 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(fm2)
```

```
##
## Call:
## lm(formula = OBA ~ EBP + I(EBP^2), data = baseball)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.139 -15.057  -4.834   8.839  56.401
##
```

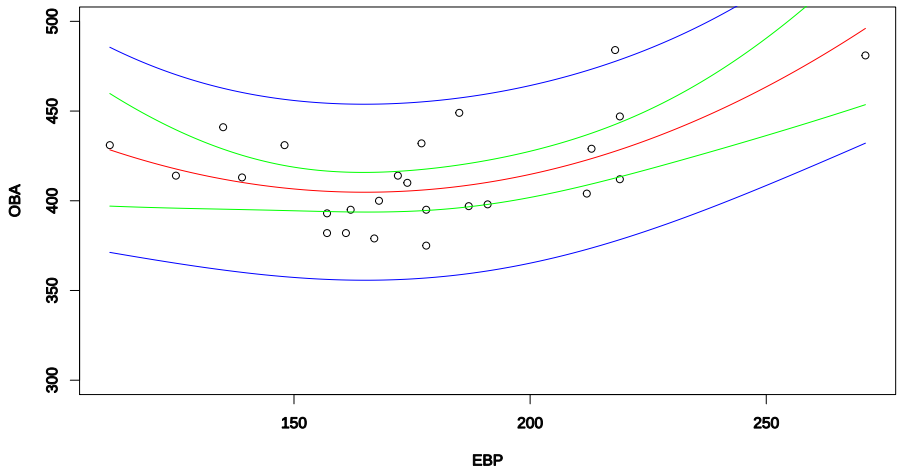
```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 625.729571  90.120360   6.943  5.7e-07 ***
## EBP          -2.679007   0.977776  -2.740  0.01196 *
## I(EBP^2)      0.008120   0.002603   3.120  0.00499 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 23.04 on 22 degrees of freedom
## Multiple R-squared:  0.4251,Adjusted R-squared:  0.3729
## F-statistic: 8.135 on 2 and 22 DF,  p-value: 0.002266
```

This final comparison provides a significant difference and suggests the quadratic would be the most suitable based on the data.

We should plot our selectd model with confidence and prediction bands:

```
min(baseball$EBP)
## [1] 111
max(baseball$EBP)
## [1] 271
x=111:271
x
##   [1] 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127
##  [18] 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##  [35] 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161
##  [52] 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178
##  [69] 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195
##  [86] 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
## [103] 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229
## [120] 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246
## [137] 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263
## [154] 264 265 266 267 268 269 270 271
y=predict(fm2, newdata = data.frame(EBP=x), interval="confidence")
head(y)
##        fit      lwr      upr
## 1 428.4058 397.0293 459.7823
## 2 427.5376 396.9267 458.1484
## 3 426.6856 396.8278 456.5433
## 4 425.8498 396.7324 454.9672
## 5 425.0302 396.6404 453.4201
## 6 424.2269 396.5518 451.9021
plot(OBA~EBP, data=baseball, ylim=c(300,500))
lines(y[,1]~x, col='red')
lines(y[,2]~x, col='green')
lines(y[,3]~x, col='green')

y.p=predict(fm2, newdata = data.frame(EBP=x), interval="prediction")
head(y.p)
##        fit      lwr      upr
## 1 428.4058 371.2378 485.5739
## 2 427.5376 370.7862 484.2890
## 3 426.6856 370.3368 483.0343
## 4 425.8498 369.8898 481.8098
## 5 425.0302 369.4453 480.6151
## 6 424.2269 369.0036 479.4502
lines(y.p[,2]~x, col='blue')
lines(y.p[,3]~x, col='blue')
```



Note the red line is the fitted curve, the green the confidence bands, and the blue the prediction bands. The fit is not brilliant and the prediction band is very wide indiciating that any prediction of the response from the predictor variable may be unreliable.