Chapter 8: Linear models with both numeric and factor explanatory variables

Part 1: The interaction model

STATS 201/8

University of Auckland

Learning Outcomes

In this chapter you will learn about:

- Models which contain categorical and numeric explanatory variables[1]
- Useful plots to display the data
- The meaning of **interaction**
- Fitting a model with an interaction term
- Interpretting the fitted model
- Relevant R-code.

[1] Models of this type are commonly known as Analysis of Covariance models, which is abbreviated to ANCOVA.

Section 8.1
Example: Using both test score and attendance to explain exam score

Part A: Exploratory analysis

Example – Exam vs. test **and** attendance

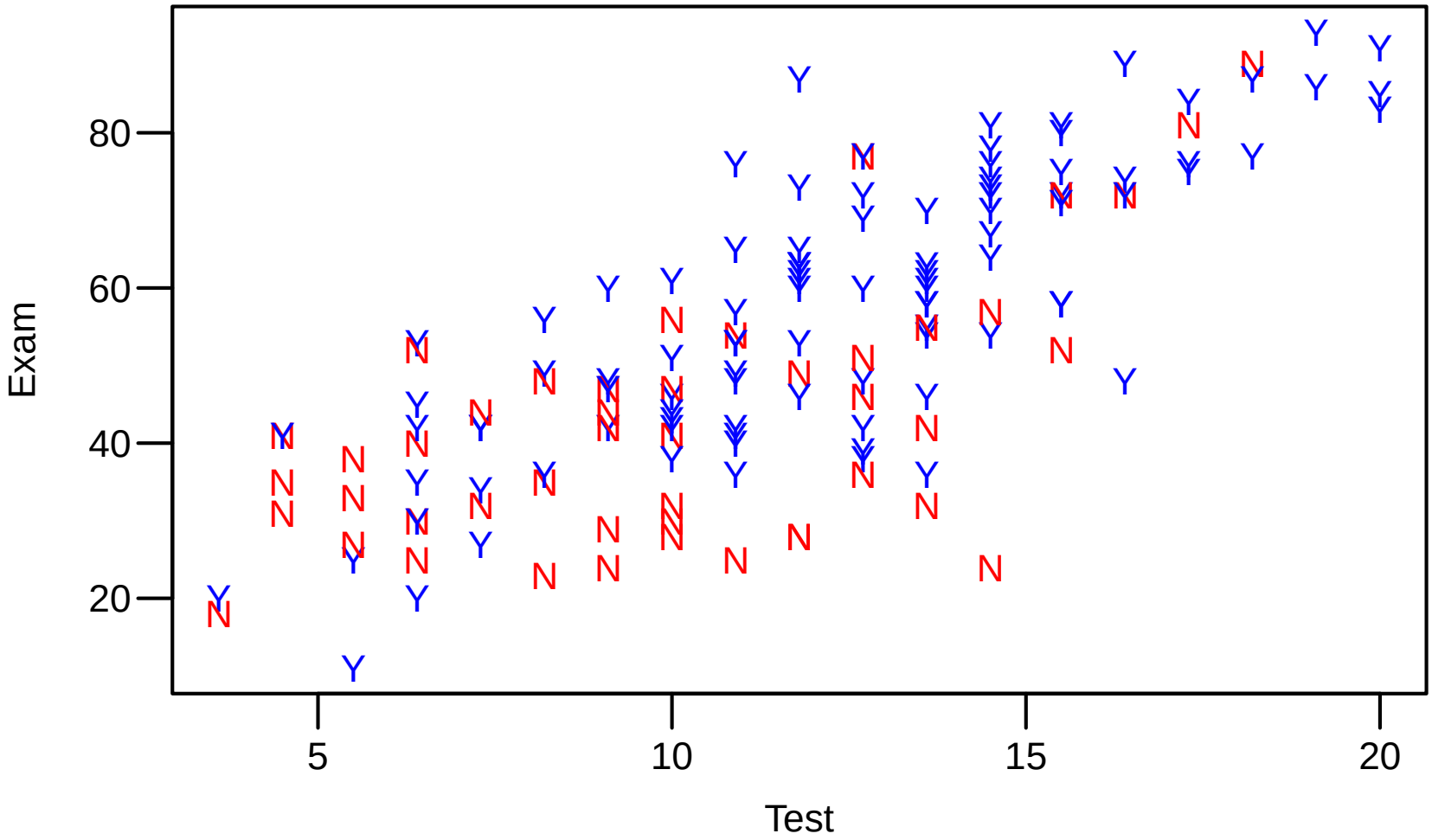
We have have learnt how to deal with the effect of test mark on exam score, and of attendance on exam score, individually.

So what is stopping us from using both? Absolutely nothing.

Let's begin by visualizing how test score relates to the exam score for the attenders and the non-attenders.

```
> ## Invoke the s20x library
> #library(s20x)
> ## Importing data into R
> Stats20x.df = read.table("Data/STATS20x.txt", header=T)
> Stats20x.df$Attend=as.factor(Stats20x.df$Attend)
> ## Plot blue "Y" for "Yes" (regular attenders), and red "N" for "No"
> plot(Exam ~ Test,data = Stats20x.df,pch=substr(Attend,1,1),
+         col=ifelse(Attend=="Yes","blue","red"))
```

Exam vs. test **and** attendance...

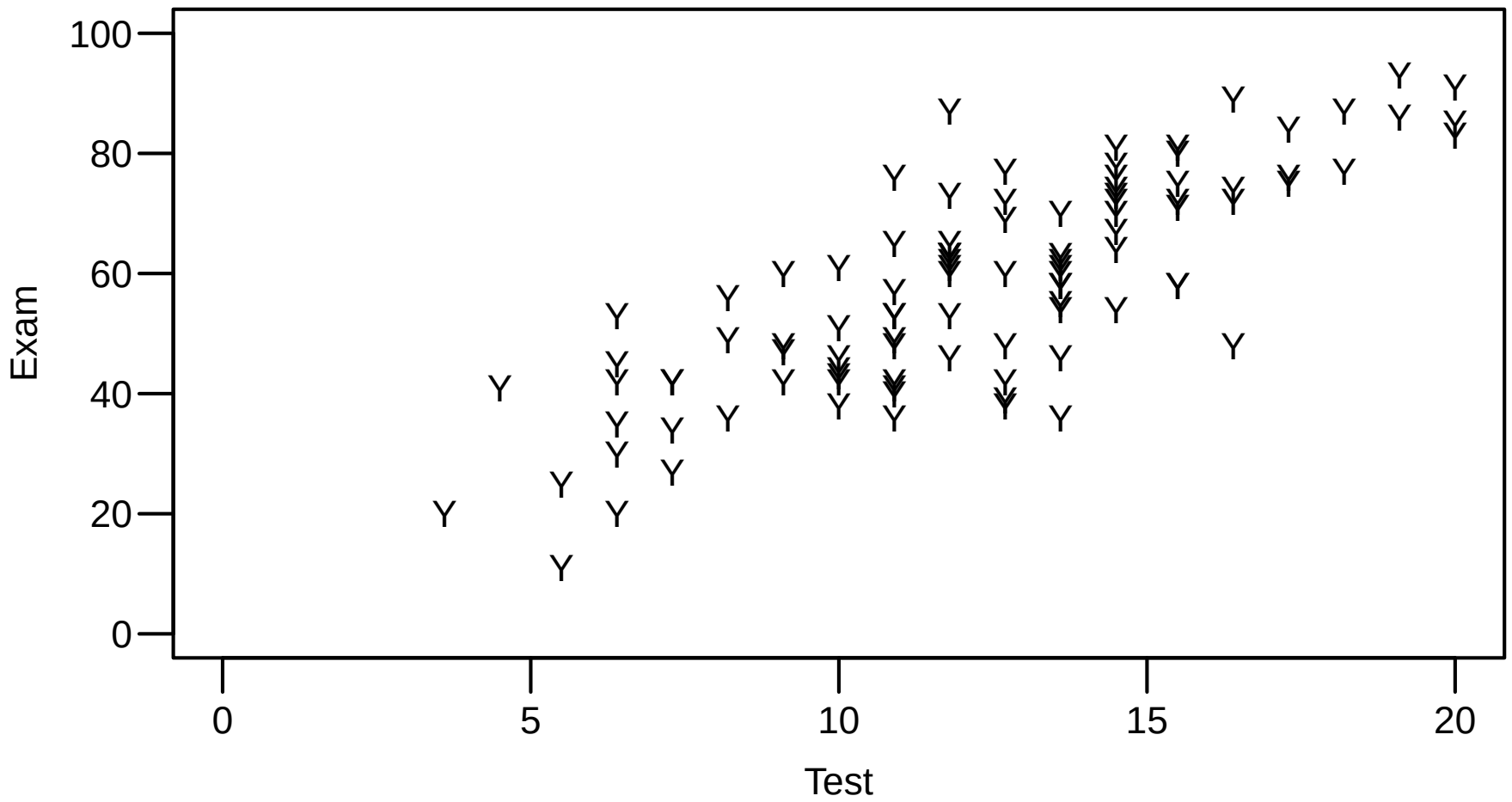


The above plot is a bit cluttered, so we could also draw a plot for each attendance type. For the sake of comparison it is important to ensure that the horizontal and vertical limits of the two scatter plots are the same.

Exam vs. test **and** attendance. . .

Here is the plot for the regular attenders.

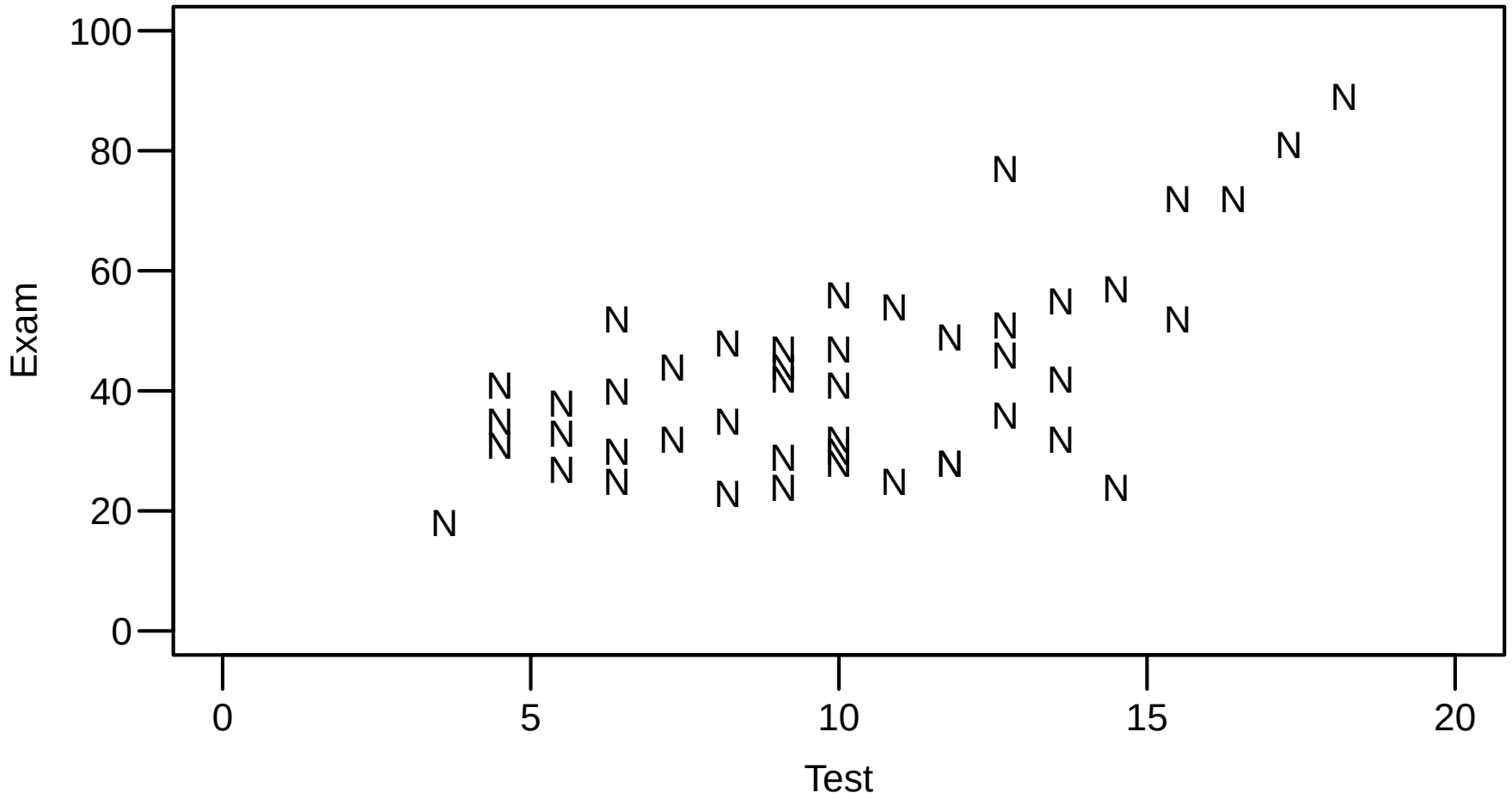
```
> Attendees.df = subset(Stats20x.df, Attend == "Yes")
> plot(Exam ~ Test, data = Attendees.df, xlim = c(0, 20), ylim = c(0, 100),
+      pch = "Y", cex = 0.7)
```



Exam vs. test **and** attendance...

Here is the plot for the non-attenders.

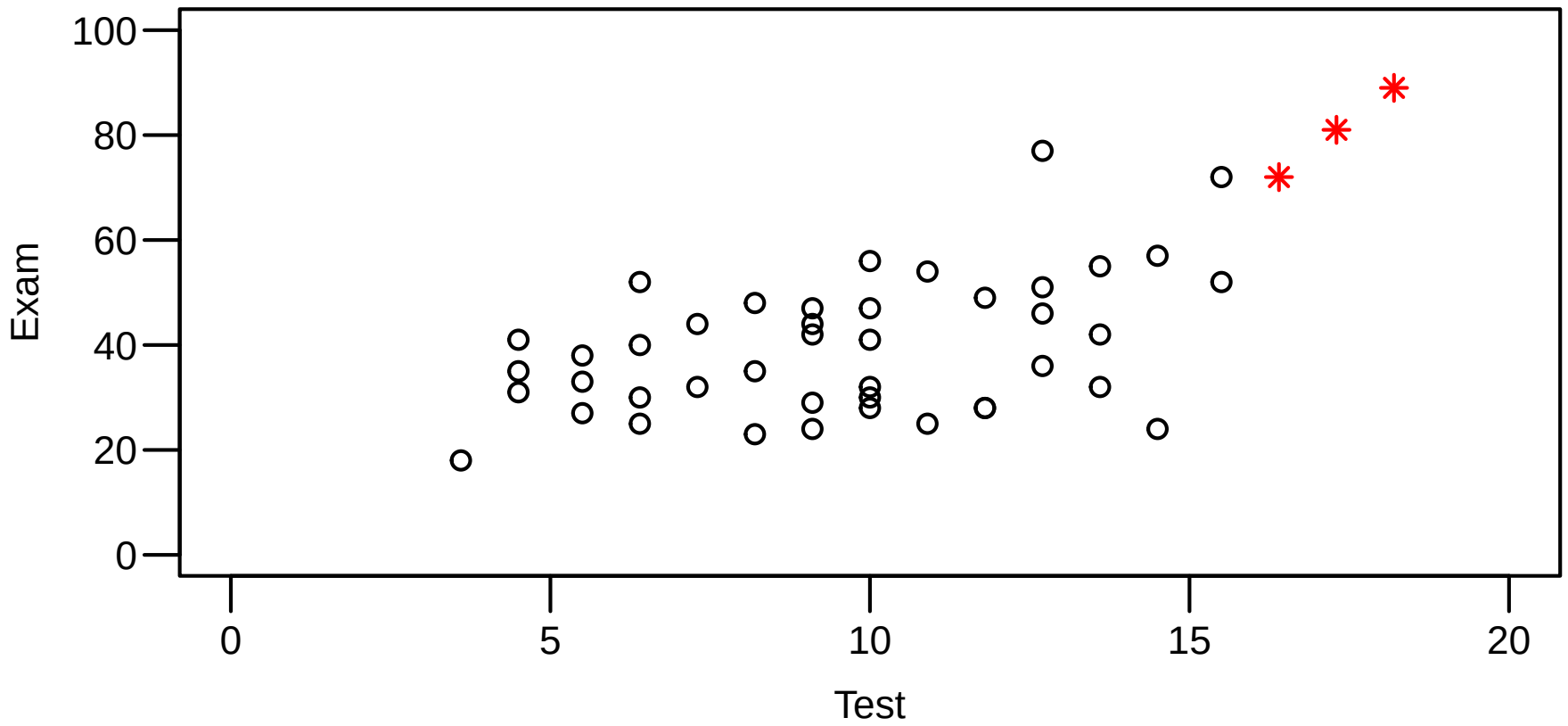
```
> Absentees.df = subset(Stats20x.df, Attend == "No")
> plot(Exam ~ Test, data = Absentees.df, xlim = c(0, 20), ylim = c(0, 100),
+      pch = "N", cex = 0.7)
```



Exam vs. test **and** attendance...

Also, there seems to be some non-attenders who do well in the test and exam so we could (and will) see whether we should include these people. They are identified in red (stars) with the R code below.

```
> plot(Exam ~ Test, data = Absentees.df, xlim = c(0, 20), ylim = c(0, 100),
+      cex = 0.7, col = ifelse(Absentees.df$Test <=16 , "black", "red"),
+      pch = ifelse(Absentees.df$Test <=16 , 1, 8))
```



Exam vs. test **and** attendance...

Hmmm—it seems that the non-attenders may get less 'return on investment' on test efforts compared to the regular attenders.

What we mean is that is that the slope looks less steep for non-attenders than regular attenders.

Can we explore this idea with a linear model? **Yes!**

Section 8.2
Example: Using both test score and attendance to explain exam score

Part B: Fitting the linear model

Exam vs. test **and** attendance...

So, it looks like we need to fit two different lines depending on whether the student is a regular attender or not.

One approach would be to fit separate linear models to the data in the `Attendees.df` and `Absentees.df` dataframes. However, this approach limits the questions that we can answer.

A more powerful approach is to use a single `lm` model to fit the two lines.

We can do this by using indicator variables (recall Chap 5).

Let us recode attendance using an indicator variable to indicate whether the student was an attender or not.

Exam vs. test **and** attendance. . .

We will call our indicator variable D for greater convenience of notation.[2]

```
> ## Boolean statement if Attend ="Yes" (TRUE) D=1, othwerwise 0 (FALSE);
> Stats20x.df$D = as.numeric(Stats20x.df$Attend=="Yes")
> table(Stats20x.df$Attend, Stats20x.df$D) ## Check it is okay

       0   1
  No   46   0
  Yes   0 100
```

[2] Last time (see Chapter 5) we called this indicator variable Attend2 – but D is easier to use in an equation.

Exam vs. test **and** attendance...

Formulating the model

Our straight line model for the non-attenders (i.e., `D = 0`) students will be:

$$Exam = \beta_0 + \beta_1 \times \texttt{Test} + \varepsilon \text{ where } \varepsilon \overset{iid}{\sim} N(0, \sigma^2).$$

We would expect some benefit from doing better in the test, so we would suspect that $\beta_1 > 0$. The null hypothesis is, as usual,

$$H_0 : \beta_1 = 0.$$

Exam vs. test **and** attendance...

Formulating the model...

From the scatter plots we suspect there may be a different slope value for the regular attenders (in fact steeper, i.e., greater positive value.)

So what we are saying is that the slope for attenders is the slope for non-attenders plus a positive number so that it increases the slope.

We will call this additional (positive) number β_3.[3]

So we can say that the slope for any student is:

$$\beta_1 + D \times \beta_3$$

where `D = 0` when the student is a non-attender and `D = 1` when they attend regularly.

[3] The choice of symbol β_3 will be come obvious soon.

Exam vs. test **and** attendance...

Formulating the model...

For the non-attenders (`D = 0`) the slope is:

$$\begin{aligned}\beta_1 + D \times \beta_3 &= \beta_1 + 0 \times \beta_3 \\ &= \beta_1.\end{aligned}$$

For the regular attenders (`D = 1`) the slope is:

$$\begin{aligned}\beta_1 + 1 \times \beta_3 &= \beta_1 + 1 \times \beta_3 \\ &= \beta_1 + \beta_3.\end{aligned}$$

We suspect that $\beta_3 > 0$. The null hypothesis, as usual, is $H_0 : \beta_3 = 0$. The scatter plots suggest that the slope associated with `Test` changes depending on whether the student attends or not. This idea is known as interaction. That is, the effect of `Test` interacts with the students' attendance behaviour.

In our example, students who attend regularly, we suspect, get 'more' from the `Test` than non-attenders.

Exam vs. test **and** attendance...

Formulating the model...

The intercept for both groups can be formulated in a similar way. That is:

$$\beta_0 + D \times \beta_2.$$

So, for the non-attenders (`D = 0`) the intercept is β_0.

For the regular attenders (`D = 1`): the intercept is $\beta_0 + \beta_2$.

We do not have much interest in the intercept terms if the equal slope hypothesis ($H_0 : \beta_3 = 0$) is rejected[4], since they give the expected exam scores for students who get zero test score. So, we would then not be interested in testing $H_0 : \beta_2 = 0$.

[4] If the equal slope hypothesis is **not** rejected then we **do** have interest in β_2 since it is then the difference between attenders and non-attenders for any given test score

Exam vs. test **and** attendance...

Formulating the model...

So, our model is:

$$\begin{aligned}\texttt{Exam} &= \beta_0 + \beta_2 \times D + (\beta_1 + \beta_3 \times D)\texttt{Test} + \varepsilon \\ &= (\beta_0 + \beta_2 \times D) + (\beta_1 + \beta_3 \times D)\texttt{Test} + \varepsilon \\ &= \beta_0 + \beta_1 \times \texttt{Test} + \beta_2 \times D + \beta_3 \times D \times \texttt{Test} + \varepsilon\end{aligned}$$

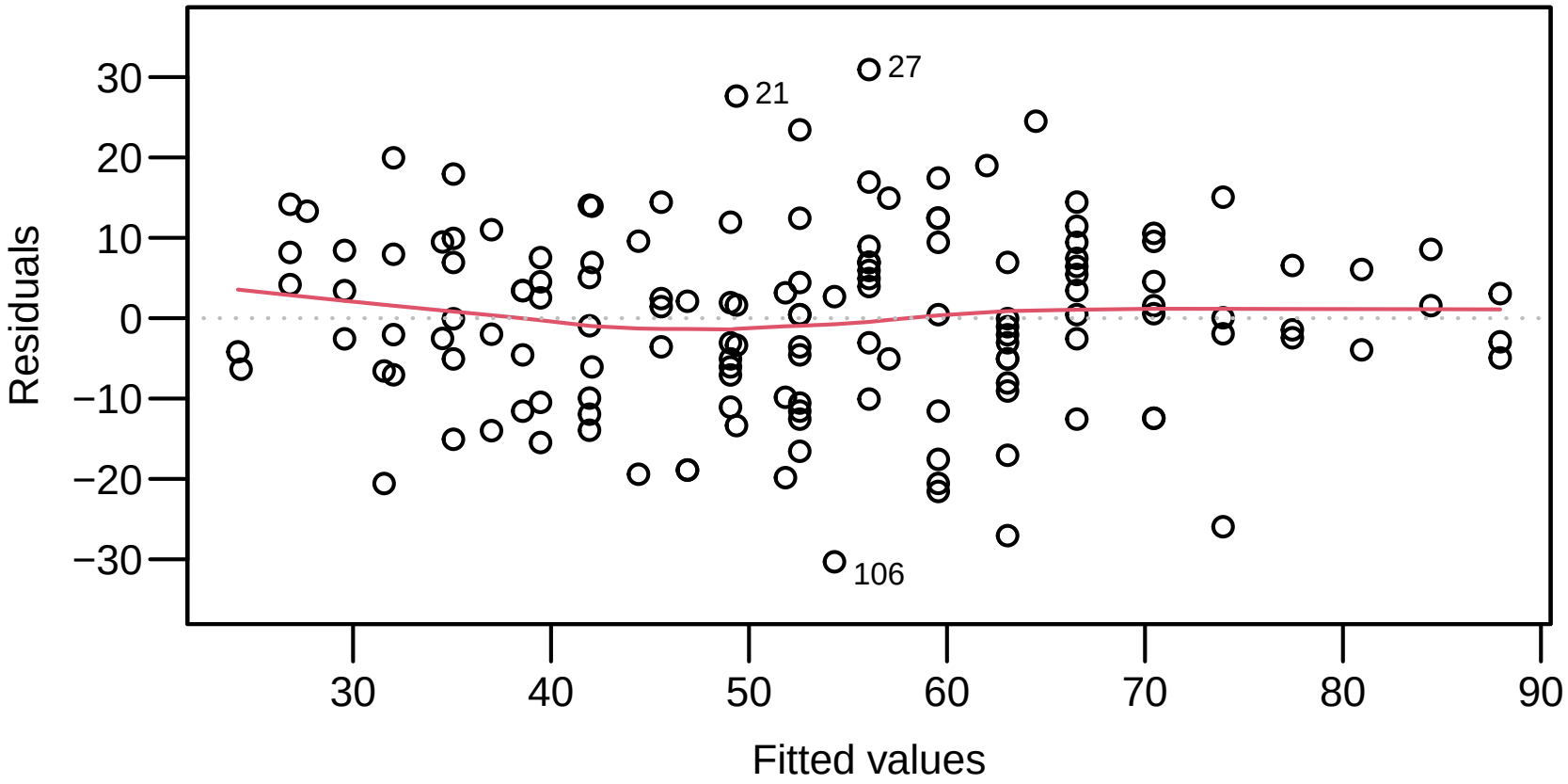
In order to fit this model we create an additional explanatory variable $D \times \texttt{Test}$.

```
> Stats20x.df$TestD  = with(Stats20x.df, {TestD = D * Test})
> TestAttend.fit = lm(Exam ~ Test + D + TestD, data = Stats20x.df)
```

Exam vs. test **and** attendance...

Assumption checks

```
> plot(TestAttend.fit,which=1)
```

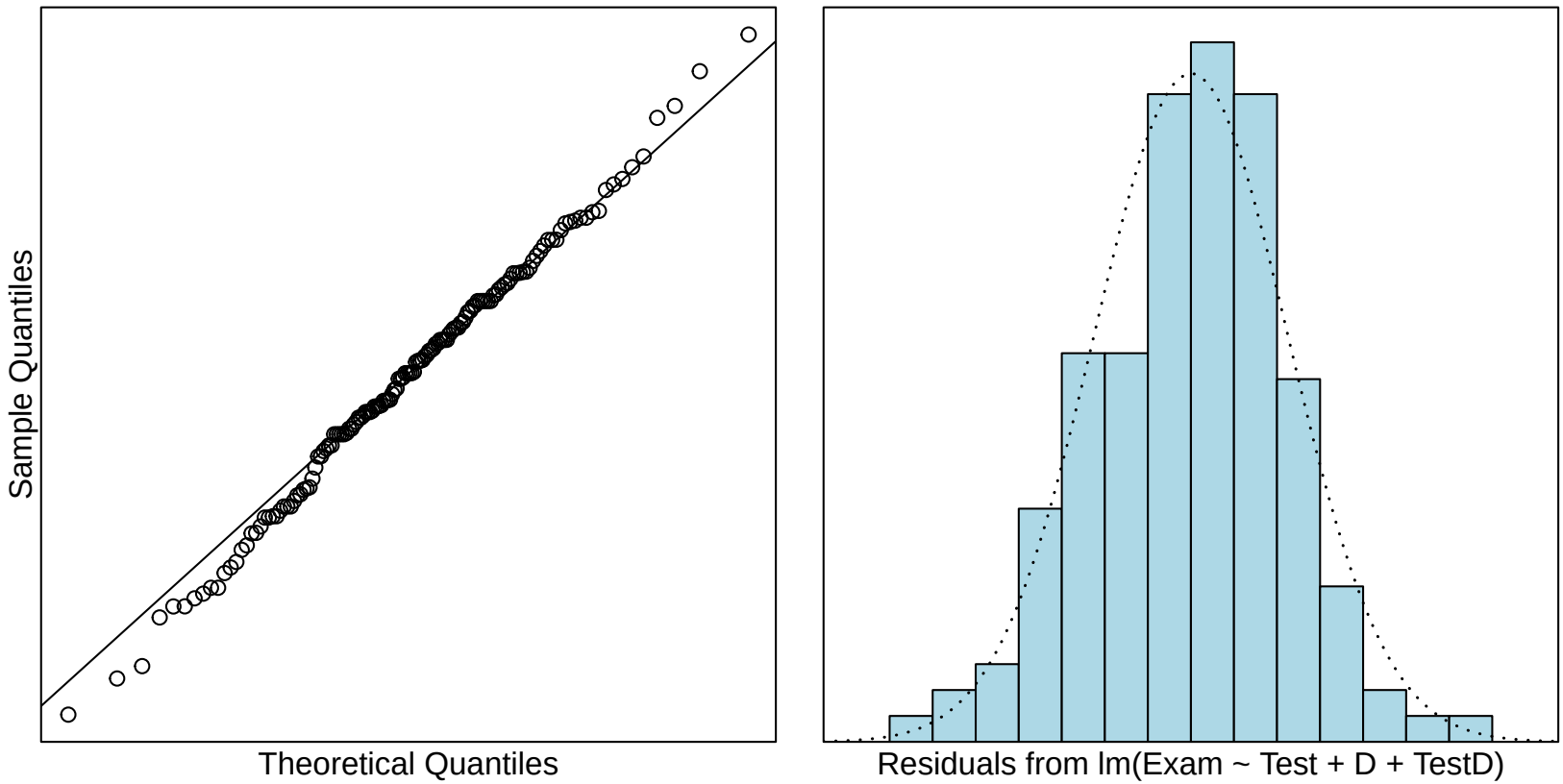


This looks okay. There is a narrowing at the higher values of fit, but there are fewer observations there.

Exam vs. test **and** attendance...

Assumption checks...

```
> normcheck(TestAttend.fit)
```

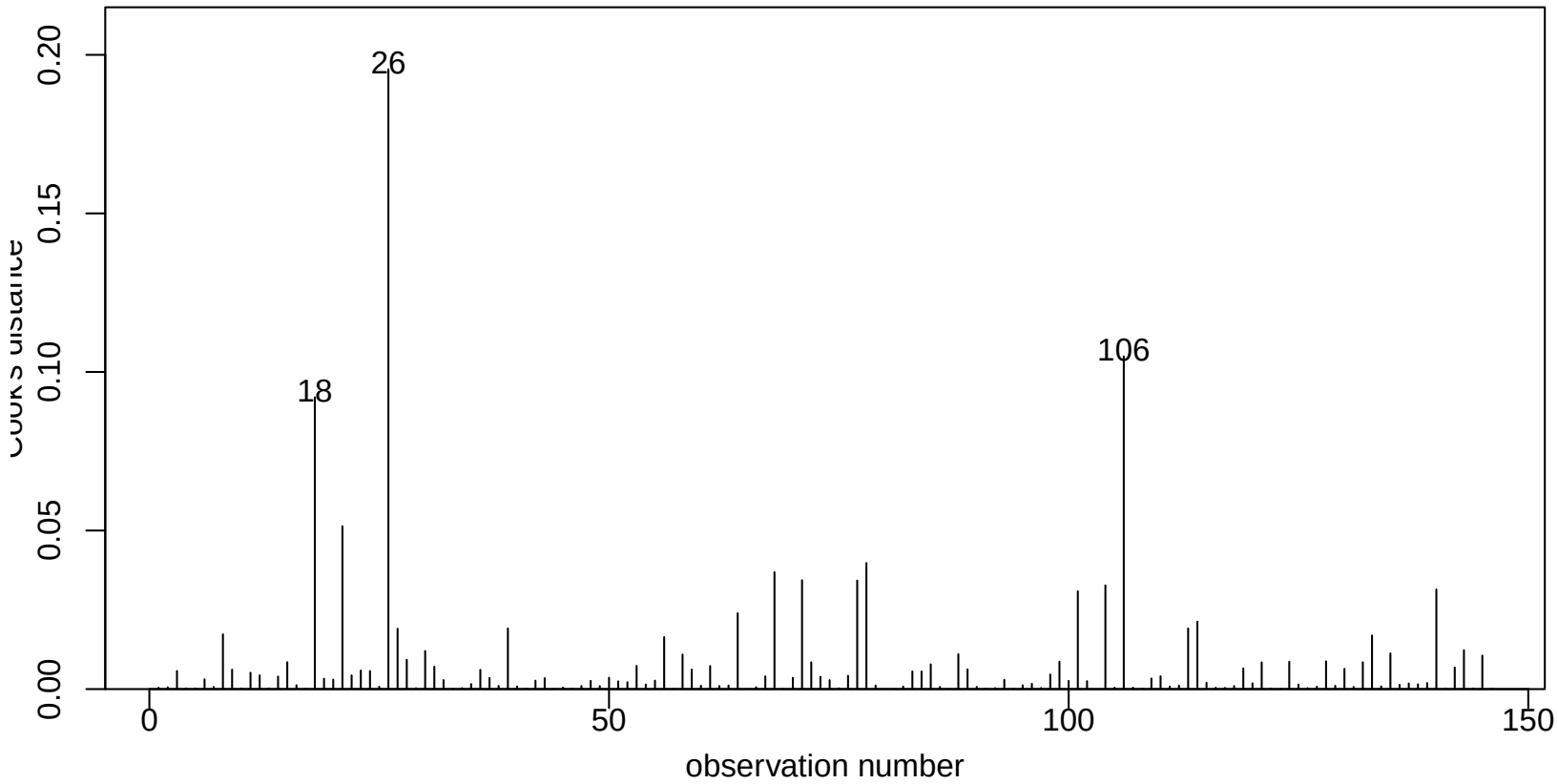


The normality assumption seems fine.

Exam vs. test **and** attendance...

Assumption checks...

```
> cooks20x(TestAttend.fit)
```



No unduly influential points here.

Exam vs. test **and** attendance...

Let us look at the fit

We can now trust the fitted `lm`. The summary output is:

```
> summary(TestAttend.fit)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.4467     4.9443   2.922  0.00405 **
Test          2.7496     0.4603   5.973 1.78e-08 ***
D            -4.2582     6.3723  -0.668  0.50506
TestD         1.1380     0.5577   2.040  0.04316 *
---
Residual standard error: 11.41 on 142 degrees of freedom
Multiple R-squared:  0.6347,Adjusted R-squared:  0.627
F-statistic: 82.25 on 3 and 142 DF,  p-value: < 2.2e-16
```

Note that the coefficient for `TestD` is significant which means our intuition seem correct. That is, the regular attenders do get a 'greater return' on their `Test` 'investment'.

Exam vs. test **and** attendance...

Making sense of it all

Let's take a closer look at the model we have just fitted. We will produce a separate plot for each attender group.

Recall: `D = 0` if the student was a non-attender (the baseline level).

Therefore, the estimated coefficients that are associated with non-attenders are $\hat{\beta}_0$ and $\hat{\beta}_1$:

```
> coef(TestAttend.fit)[1:2]
(Intercept)        Test
  14.446750    2.749568
```

Exam vs. test **and** attendance...

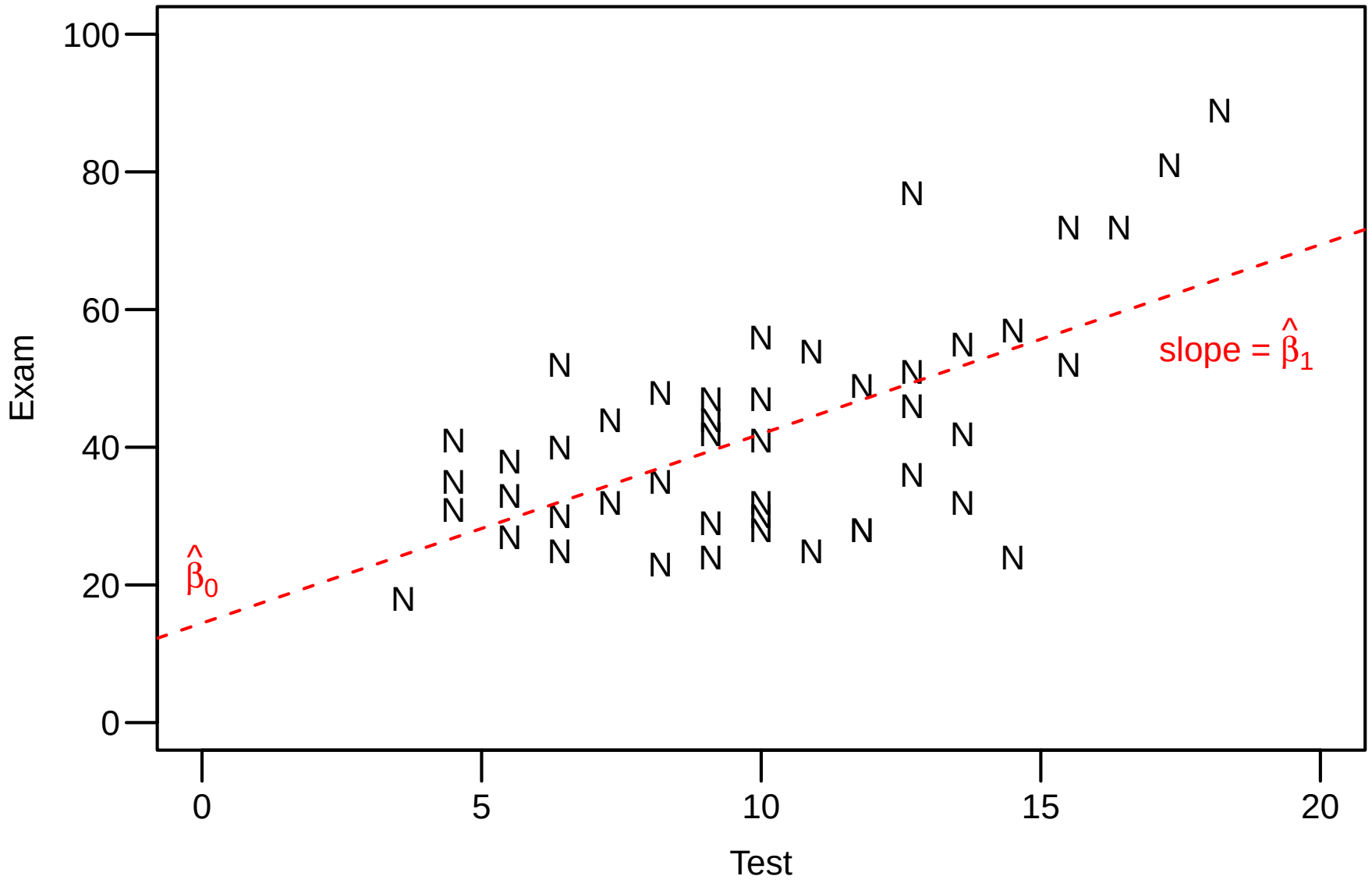
Making sense of it all...

Here is the R code for a plot of the non-attender group with their fitted line (D = 0).

```
> plot(Exam~Test,data=Absentees.df,pch="N",cex=0.7,xlim=c(0,20),ylim=c(0,100))
> abline(TestAttend.fit$coef[1:2],lty=2, col="red")
> text(0, 22,expression(hat(beta)[0]),col="red", cex = 0.7)
> text(18.5, 55, expression("slope = "*hat(beta)[1]),col="red", cex = 0.7)
```

Exam vs. test **and** attendance...

Making sense of it all...



Exam vs. test **and** attendance...

Making sense of it all...

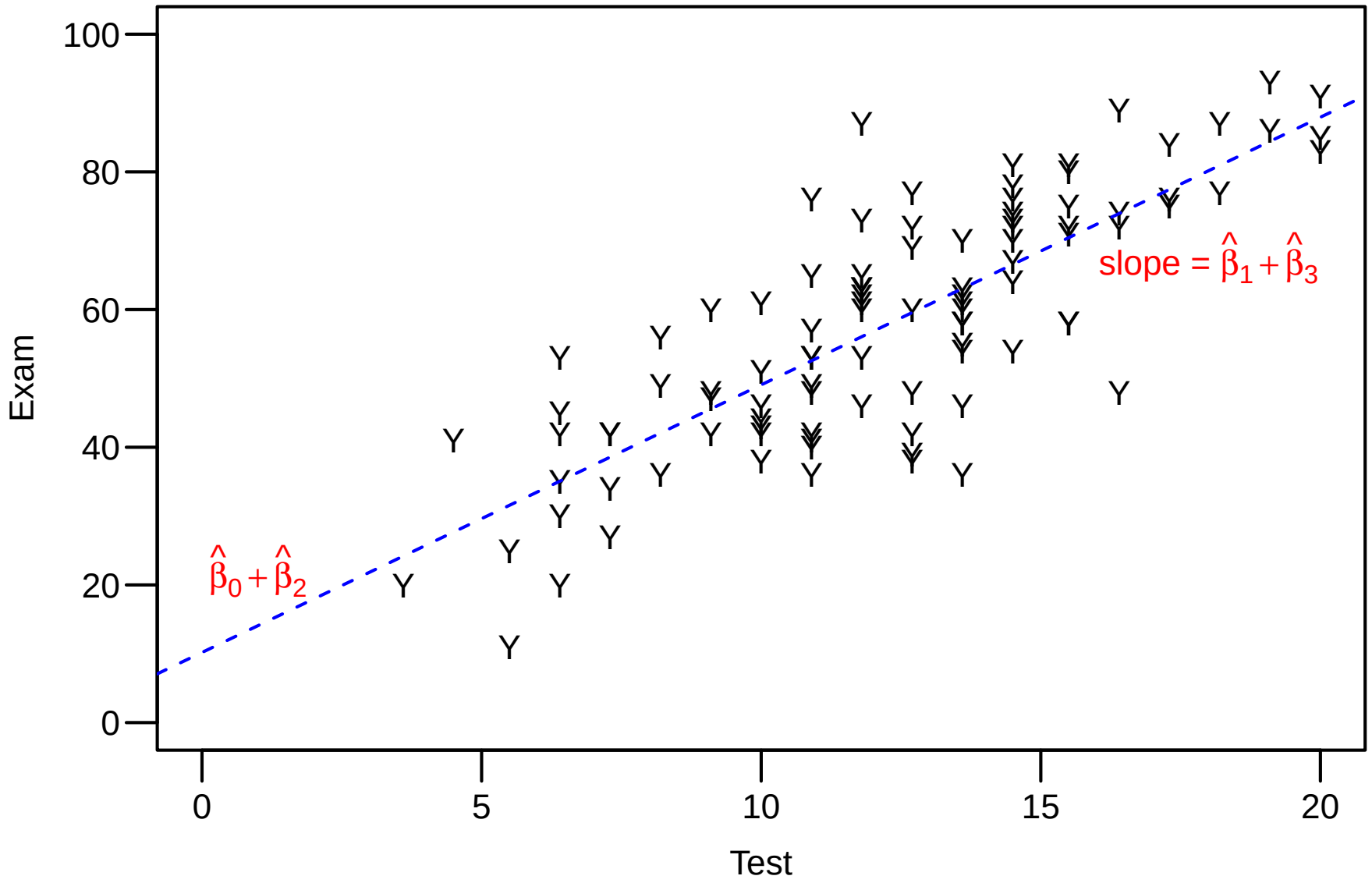
Here is the R code for a plot of the regular attender group with their fitted line (D = 1).

```
> plot(Exam ~ Test, data = Attendees.df, pch = "Y", cex = 0.7,
+      xlim = c(0, 20), ylim = c(0, 100))
> coeffs = TestAttend.fit$coef ## Easier to work with these terms
> abline(coeffs[1:2] + coeffs[3:4], lty = 2, col = "blue")
> text(1, 22, expression(hat(beta)[0] + hat(beta)[2]), col = "red", cex = 0.7)
> text(18, 67.5,expression(paste("slope = ", hat(beta)[1] + hat(beta)[3])),
+      col = "red", cex = 0.7)
```

Note that as D = 1 we add $\hat{\beta}_2$ and $\hat{\beta}_3$, to the baseline intercept and slope terms, respectively.

Exam vs. test **and** attendance...

Making sense of it all...



Exam vs. test **and** attendance...

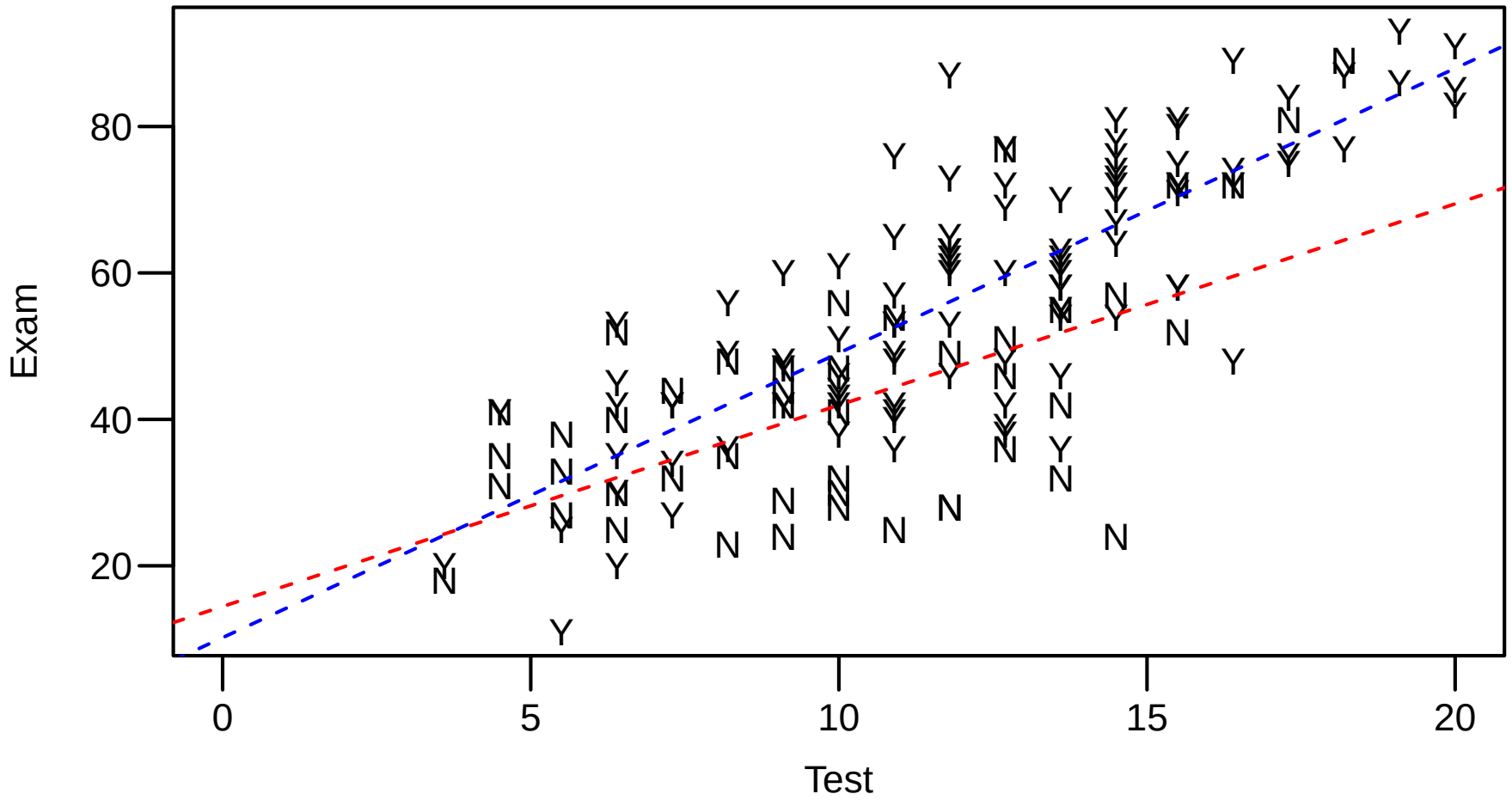
Making sense of it all...

All together now:

```
> ## Plot these data all together
> b=coef(TestAttend.fit) # easier to work with these terms
> plot(Exam ~ Test,data = Stats20x.df,pch=substr(Attend,1,1),cex=0.7,xlim=c(0,20))
> ## Red for "No" and blue for "Yes".
> abline(b[1:2], lty =2, col="red")
> abline(b[1]+b[3],b[2]+b[4],lty=2, col="blue" )
```

Exam vs. test **and** attendance...

Making sense of it all...



This a a visual confirmation of our intuition that the regular attenders get a 'greater return' on their `Test` 'investment'.

Fitting the interaction model directly with `lm`

All that hard work we did with constructing `D` and `TestD` can be avoided since `lm` will automatically do this for us.

We were interested to see whether the effect of `Test` interacts with the students `Attend` variable. Using `lm` we simply specify `Test * Attend` to fit the model with interaction. That is,

```
> TestAttend.fit2=lm(Exam~Test*Attend, data=Stats20x.df)
> summary(TestAttend.fit2)
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     14.4467     4.9443   2.922  0.00405 **
Test             2.7496     0.4603   5.973 1.78e-08 ***
AttendYes       -4.2582     6.3723  -0.668  0.50506
Test:AttendYes   1.1380     0.5577   2.040  0.04316 *
---
Residual standard error: 11.41 on 142 degrees of freedom
Multiple R-squared:  0.6347,Adjusted R-squared:  0.627
F-statistic: 82.25 on 3 and 142 DF,  p-value: < 2.2e-16
```

Fitting the interaction model directly with `lm`...

Compare this with

```
> Summary(TestAttend.fit)
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  14.4467     4.9443   2.922  0.00405 **
Test          2.7496     0.4603   5.973 1.78e-08 ***
D            -4.2582     6.3723  -0.668  0.50506
TestD         1.1380     0.5577   2.040  0.04316 *
---
Residual standard error: 11.41 on 142 degrees of freedom
Multiple R-squared:  0.6347,Adjusted R-squared:  0.627
F-statistic: 82.25 on 3 and 142 DF,  p-value: < 2.2e-16
```

We have the same outputs, but with slightly different names.

Note: `Test * Attend` is shorthand notation. You can be more explicit about the individual terms in the model by writing

```
> TestAttend.fit2=lm(Exam ~Test + Attend + Test:Attend, data=Stats20x.df)
```

We read this as, *the effect of `Test`, plus the effect of `Attend`, plus the interaction between `Test` and `Attend`, which is denoted by `Test:Attend`.*

Section 8.3
Interpretting the fitted model

Exam vs. test **and** attendance...

Let us take stock of this model

```
> summary(TestAttend.fit2)
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     14.4467     4.9443   2.922  0.00405 **
Test             2.7496     0.4603   5.973 1.78e-08 ***
AttendYes       -4.2582     6.3723  -0.668  0.50506
Test:AttendYes   1.1380     0.5577   2.040  0.04316 *
---
Residual standard error: 11.41 on 142 degrees of freedom
Multiple R-squared:  0.6347,Adjusted R-squared:  0.627
F-statistic: 82.25 on 3 and 142 DF,  p-value: < 2.2e-16
```

When we looked at the models that used explanatory variables `Test` and `Attend` by themselves (in Chapter 2 & 5 resp.), they explained 59% and 15% of the variability of `Exam` respectively. When we use them together, we explain about **63**% of the variability. **Why is it not** $59 + 15 = 74\%$**?**.

It is because the addition of `Attend` can only explain variability that has not already been explained by `Test`. If `Attend` and `Test` are closely related then this may not be much – this is called **multi-collinearity**.

Exam vs. test **and** attendance...

Let us take stock of this model...

We see that our intuition was correct. That is, the slope for `Test` of attenders is greater that for non-attenders. This is because the estimate of the difference in these slopes `TestD`.

```
> coef(TestAttend.fit2)[4]
Test:AttendYes
       1.13799
```

is positive and statistically significant (P-value ≈ 0.04).

We were not sure about the differences in intercept and we see that this estimate is not significantly different from the hypothesised value of 0 (P-value ≈ 0.51).

Recall that we are not particularly interested in the null hypothesis $H_0 : \beta_2 = 0$ when the slopes are different, so it is standard practice to leave this term in the model when β_3 is statistically significant.

Exam vs. test **and** attendance...

Let us take stock of this model...

The baseline slope that measures the effect of `Test` for non-attenders is statistically significant.

The slope for attenders is significantly higher, so we say that the effect of `Test` interacts with the `Attend` variable, as the effect depends on the level of `Attend`.

Exam vs. test **and** attendance...

Some Inference

Confidence intervals may be needed for the coefficients:

```
> confint(TestAttend.fit2)
                     2.5 %    97.5 %
(Intercept)      4.67287511 24.220625
Test             1.83956971  3.659567
AttendYes      -16.85506294  8.338572
Test:AttendYes   0.03547053  2.240510
```

Recall: Statistical significance (at the 5% level) of a coefficient is equivalent to the (95%) confidence interval **NOT** containing zero.

Exam vs. test **and** attendance...

Some predictions

```
> predTestAttend.df = data.frame(Test = c(0, 10, 10, 20),
+                                Attend = factor(c("No", "No", "Yes", "Yes"))
+                                )
> predTestAttend.df
  Test Attend
1    0     No
2   10     No
3   10    Yes
4   20    Yes
```

Let us estimate the expected exam scores for these values of test score and attendance:

```
> predict(TestAttend.fit2,predTestAttend.df, interval="confidence")
       fit       lwr      upr
1 14.44675  4.672875 24.22062
2 41.94243 38.616376 45.26849
3 49.06409 46.412194 51.71599
4 87.93968 82.610100 93.26926
```

Exam vs. test **and** attendance...

Some predictions...

Now, let us predict the exam score for individual students with those test scores and attendance:

```
> predict(TestAttend.fit2,predTestAttend.df, interval="prediction")
       fit       lwr       upr
1 14.44675 -10.13028  39.02378
2 41.94243  19.14848  64.73639
3 49.06409  26.35871  71.76947
4 87.93968  64.76845 111.11092
```

This is not the best model for predicting individual student exam scores as the intervals are too wide and in some case are meaningless (at the extremes of `Test`).

This is related to the fact that we can only explain 63% of the variability in `Exam` or, equivalently, we can not account for 37% of this variability. Also, our linear model does not "know" about the constraint that exam scores must be between 0 and 100.

Exam vs. test **and** attendance...

Methods and assumption checks

A plot of the data showed that exam scores appear to increase linearly with test score, but with different lines for attenders and non-attenders.

The model with interaction was fitted and the attendance/test score interaction was found to be significant.[5]

The students should be acting independently of each other as this was an exam.

The EOV assumption appears to be reasonable notwithstanding some reduced variability at the high end of test and exam scores. The data also look approximately normal. There do not appear to be any unduly influential data points.

Our model explains 63% of the total variability in exam scores.

[5] See the Case Study for the model equation.

Exam vs. test **and** attendance...

Executive summary

There is a clear linear relationship between Test and Exam, but it differs in strength between non-attenders and attenders.

We estimate that each additional test mark (out of 20) will increase the expected exam mark of a non-attender by 1.8 to 3.7.

There is a further increase of between 0 to 2.2 marks for regular attenders.

Note: In an assignment or exam situation you might also be required to comment on additional confidence and/or prediction intervals—this will be made clear in the instructions.

Exam vs. test **and** attendance...

Executive summary...

Note that the above Executive Summary is missing the confidence interval for the effect of test mark on attenders. To obtain this CI we need to change attenders to the baseline level of `Attend`.

```
> Stats20x.df$Attend2=relevel(Stats20x.df$Attend,ref="Yes")
> TestAttend.fit2b=lm(Exam ~Test*Attend2, data=Stats20x.df)
> coef(summary(TestAttend.fit2b))
                Estimate Std. Error    t value     Pr(>|t|)
(Intercept)    10.188504  4.0199956  2.5344566 1.234648e-02
Test            3.887559  0.3148792 12.3461898 3.020895e-24
Attend2No       4.258246  6.3722922  0.6682439 5.050626e-01
Test:Attend2No -1.137990  0.5577265 -2.0404094 4.316270e-02
> confint(TestAttend.fit2b)
                   2.5 %      97.5 %
(Intercept)     2.241733 18.13527583
Test            3.265102  4.51001561
Attend2No      -8.338572 16.85506294
Test:Attend2No -2.240510 -0.03547053
```

We estimate that each additional test mark will increase the expected exam mark of an attender by 3.3 to 4.5.

Section 8.4
Assessing influence of the atypical students

Exam vs. test **and** attendance. . .

Sensitivity check

In the exploratory phase of the analysis we identified three students as being potentially anomolous. Recall, these were the 3 non-attending students who scored greater than 16 on the test.

If we have reason to think these students are 'atypical' (do we?) then it might make sense to do the analysis without them.

What do you think happens to our respective straight lines?

Let us see what happens. We will pull these three students out of the dataframe.

```
> ## Remove atypical points - Note that ! means 'not'
> Subset.df=subset(Stats20x.df, !(Test>16&Attend=="No"))
> TestAttend.fit3=lm(Exam ~Test*Attend, data=Subset.df)
> summary(TestAttend.fit3)
```

Exam vs. test **and** attendance...

Sensitivity check...

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     22.6522     5.2776   4.292  3.3e-05 ***
Test             1.7590     0.5213   3.374 0.000960 ***
AttendYes      -12.4637     6.5505  -1.903 0.059146 .
Test:AttendYes   2.1285     0.6034   3.527 0.000569 ***
---
Residual standard error: 11.01 on 139 degrees of freedom
Multiple R-squared:  0.6495,Adjusted R-squared:  0.6419
F-statistic: 85.86 on 3 and 139 DF,  p-value: < 2.2e-16
```

Note that we have smaller degrees of freedom as we have 3 fewer data points. Look at how the parameters have changed.

Our R^2 value has increased slightly as we have less variability overall.

Exam vs. test **and** attendance...

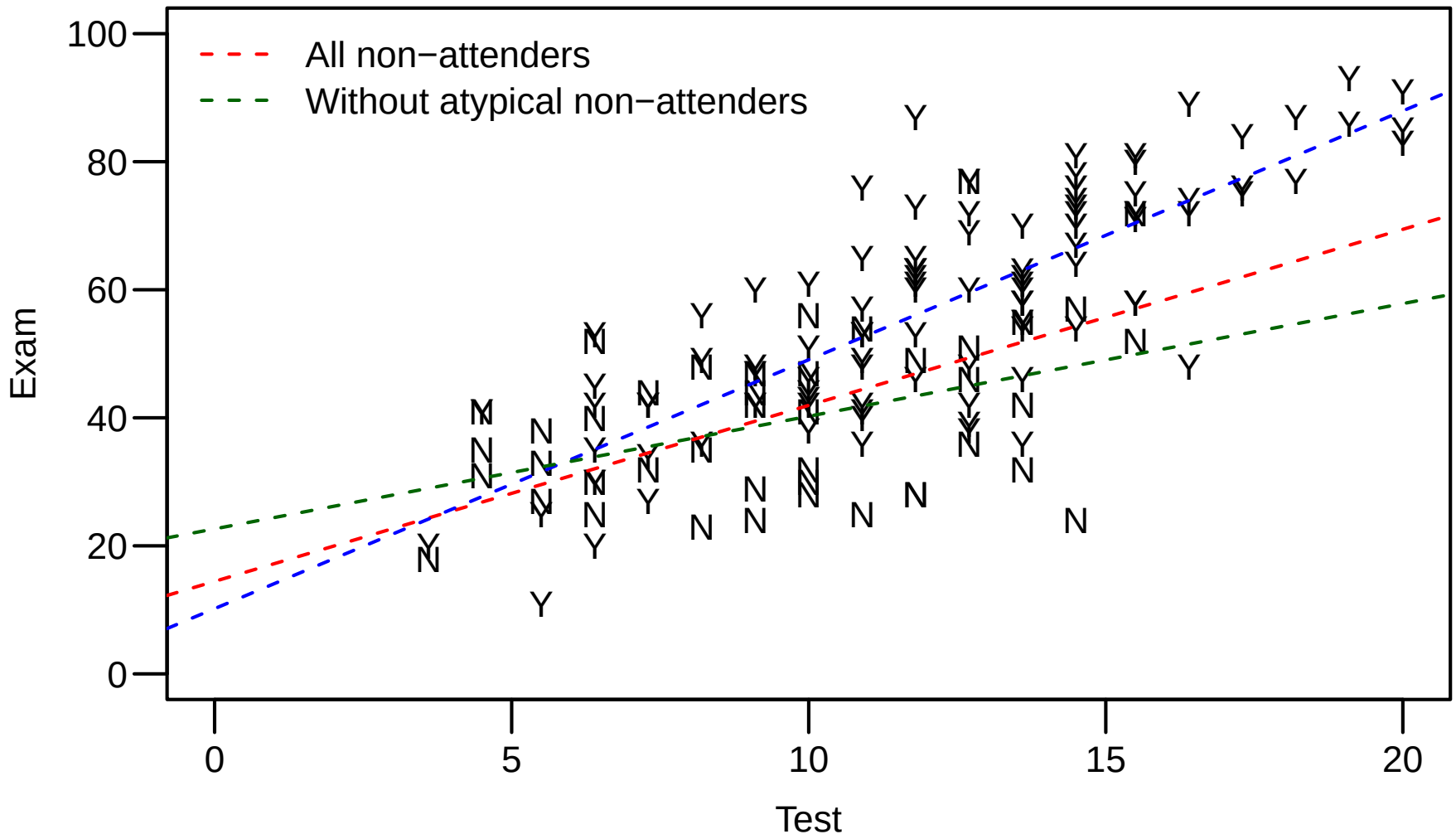
Sensitivity check...

R code to generate the plot of the full data with all three of the fitted lines:

```
> ## Plot these data all together
> plot(Exam ~ Test, data = Subset.df,pch = substr(Attend, 1, 1),cex = 0.7)
>
> ## Remember that we've defined b in Slide 26
> ## Each abline() will have a different colour
> abline(b[1:2], lty = 2, col = "red")
> abline(b[1] + b[3], b[2] + b[4], lty = 2, col = "blue")
>
> ## The fitted line without the 3 atypical points
> b2 = coef(TestAttend.fit3) ## Easier to work with these terms
> abline(b2[1:2], lty = 2, col = "green")
>
> ## Add a legend to help us differentiate between the lines for non-attenders
> legend("topleft", legend = c("All non-attenders", "Without atypical non-attenders
+        lty = 2, col = c("red", "green"), bty = "n")
```

Exam vs. test **and** attendance...

Sensitivity check...



Comments?

Section 8.5
Some insight and relevant R-code.

R tips and tricks

model.matrix

Recall that `lm` automatically created the necessary indicator variables to fit the above models.

To explicitly see the model formula that `lm` is using, we only have to ask:

```
> ModMat=model.matrix(~Test*Attend,data=Stats20x.df)
> cbind(Stats20x.df[,c("Test","Attend")],ModMat)[1:10,]
   Test Attend (Intercept) Test AttendYes Test:AttendYes
1   9.1    Yes           1  9.1         1            9.1
2  13.6    Yes           1 13.6         1           13.6
3  14.5    Yes           1 14.5         1           14.5
4  19.1    Yes           1 19.1         1           19.1
5   8.2     No           1  8.2         0            0.0
6  12.7    Yes           1 12.7         1           12.7
7   7.3    Yes           1  7.3         1            7.3
8  10.9     No           1 10.9         0            0.0
9  10.9    Yes           1 10.9         1           10.9
10  9.1    Yes           1  9.1         1            9.1
```

The `AttendYes` variable is `D`, and `Test:AttendYes` is `D` $\times$ `Test`.

Most of the R-code you need for this chapter

As always, your code requires the usual code (data exploration, etc) and model checks discussed in chapters 1 and 2.[6]

When y can be explained by a categorical (i.e., factor) variable and also a numeric (i.e., continuous) variable then you can use both.

You do not need to create indicator variables as R does this for you. It will choose the baseline for you, so be careful. You can change this if needed, using the `relevel` function.

Fit as follows:

```
> TestAttend.fit2=lm(Exam ~Test*Attend, data=Stats20x.df)
> #check to see it's okay
> plot(TestAttend.fit2, which=1) #followed by normcheck and cooks20x
> # then see if you need a separate slope for each level of your factor var.
> summary(TestAttend.fit2)
```

Interpret accordingly. In particular, if the interaction term is not significant then proceed to the next Chapter.

[6] That is, until we reach Chapter 13.