Categorical Predictors

- Predictors that are qualitative in nature are sometimes described as *categorical* or called *factors*.
- The different categories of a factor variable are called *levels*.
- We wish to incorporate these predictors into the regression analysis. We start with the example of a factor with just two levels, then show how to introduce quantitative predictors into the model and end with an example using a factor with more than two levels.

A two-level factor

We take a look at the data and produce a summary subsetted by `csa`:

```
data(sexab)
head(sexab)
```

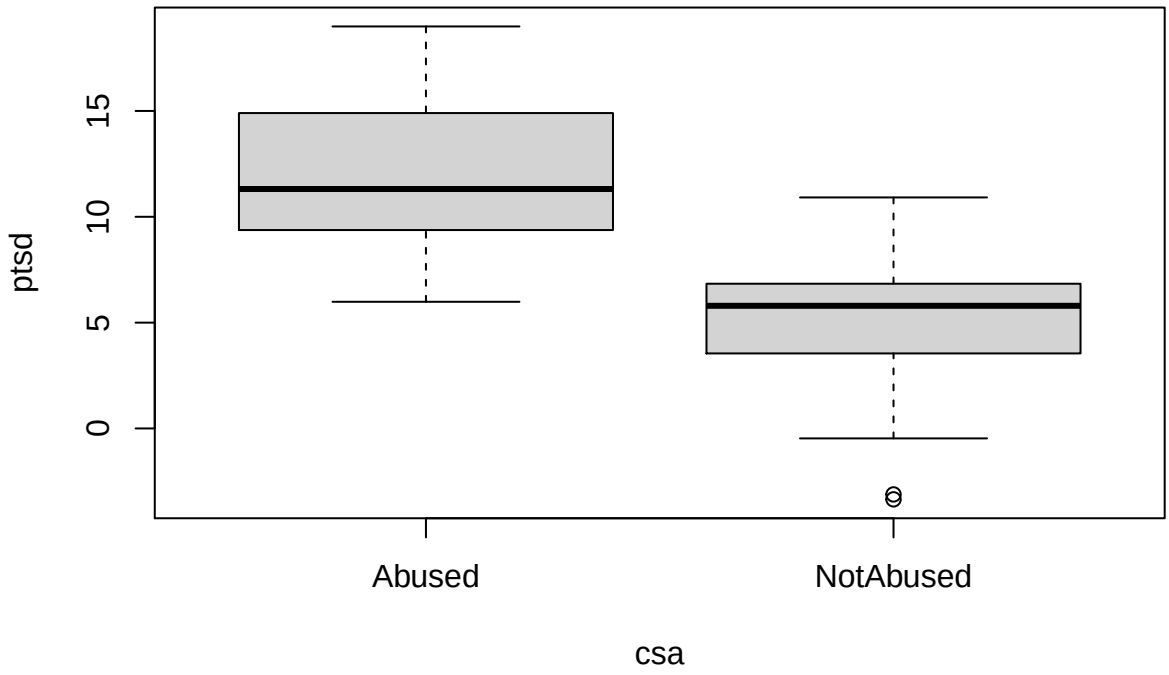
```
##        cpa     ptsd    csa
## 1  2.04786  9.71365 Abused
## 2  0.83895  6.16933 Abused
## 3 -0.24139 15.15926 Abused
## 4 -1.11461 11.31277 Abused
## 5  2.01468  9.95384 Abused
## 6  6.71131  9.83884 Abused
```

```
by(sexab, sexab$csa, summary)
```

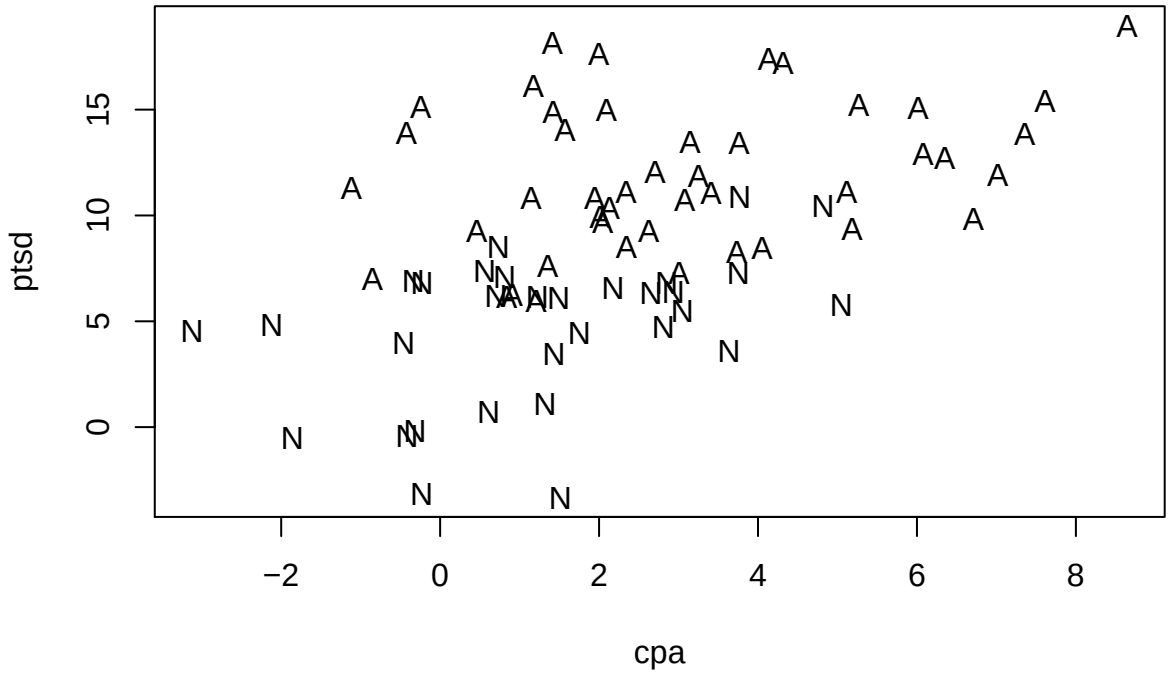
```
## sexab$csa: Abused
##       cpa              ptsd              csa
##  Min.   :-1.115   Min.   : 5.985   Abused   :45
##  1st Qu.: 1.415   1st Qu.: 9.374   NotAbused: 0
##  Median : 2.627   Median :11.313
##  Mean   : 3.075   Mean   :11.941
##  3rd Qu.: 4.317   3rd Qu.:14.901
##  Max.   : 8.647   Max.   :18.993
## ------------------------------------------------------------
## sexab$csa: NotAbused
##       cpa               ptsd              csa
##  Min.   :-3.1204   Min.   :-3.349   Abused   : 0
##  1st Qu.:-0.2299   1st Qu.: 3.544   NotAbused:31
##  Median : 1.3216   Median : 5.794
##  Mean   : 1.3088   Mean   : 4.696
##  3rd Qu.: 2.8309   3rd Qu.: 6.838
##  Max.   : 5.0497   Max.   :10.914
```

Now plot the data:

```
plot(ptsd ~ csa, sexab)
```




```
plot(ptsd ~ cpa, pch = as.character(csa), sexab)
```



We see that those in the abused group have higher levels of PTSD than those in the non-abused in the left panel.

We can test this difference:

```r
# Assume that the variance is eqaul in the two groups.
t.test(ptsd ~ csa, sexab, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  ptsd by csa
## t = 8.9387, df = 74, p-value = 2.172e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  5.630165 8.860273
## sample estimates:
##    mean in group Abused mean in group NotAbused
##               11.941093                4.695874
```

Since p-value is less than 0.05, we find that it is clearly significant.

Our strategy is to incorporate qualitative predictors within the $Y = X\beta + \epsilon$ framework. We can then use the estimation, inferential and diagnostic techiniques.

- To put qulitative predictors into the $Y = X\beta + \epsilon$ form we need to code the qualitative predictors.
- We can do this using *dummy variables*.
- For a catetgorial predictor (or factor) with two levels, we define dummy variables d_1 and d_2:

$$d_i = \begin{cases} 0 & \text{is not level i} \\ 1 & \text{is level i} \end{cases}$$

Let's create dummy variables and fit them using a linear model:

```r
d1 <- ifelse(sexab$csa == "Abused", 1, 0)
d2 <- ifelse(sexab$csa == "NotAbused", 1, 0)
lmod <- lm(ptsd ~ d1 + d2, sexab)
summary(lmod)
```

```
##
## Call:
## lm(formula = ptsd ~ d1 + d2, data = sexab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0451 -2.3123  0.0951  2.1645  7.0514
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.6959     0.6237   7.529 1.00e-10 ***
## d1            7.2452     0.8105   8.939 2.17e-13 ***
## d2                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.473 on 74 degrees of freedom
## Multiple R-squared:  0.5192, Adjusted R-squared:  0.5127
## F-statistic:  79.9 on 1 and 74 DF,  p-value: 2.172e-13
```

We can see a warning about singularities and that the parameter for the second dummy variables has not been estimated.

The cause of this problem can be revealed by studying the X model matrix:

```
model.matrix(lmod)
```

```
##    (Intercept) d1 d2
## 1            1  1  0
## 2            1  1  0
## 3            1  1  0
## 4            1  1  0
## 5            1  1  0
## 6            1  1  0
## 7            1  1  0
## 8            1  1  0
## 9            1  1  0
## 10           1  1  0
## 11           1  1  0
## 12           1  1  0
## 13           1  1  0
## 14           1  1  0
## 15           1  1  0
## 16           1  1  0
## 17           1  1  0
## 18           1  1  0
## 19           1  1  0
## 20           1  1  0
## 21           1  1  0
## 22           1  1  0
## 23           1  1  0
## 24           1  1  0
## 25           1  1  0
## 26           1  1  0
## 27           1  1  0
## 28           1  1  0
## 29           1  1  0
## 30           1  1  0
## 31           1  1  0
## 32           1  1  0
## 33           1  1  0
## 34           1  1  0
## 35           1  1  0
## 36           1  1  0
## 37           1  1  0
## 38           1  1  0
## 39           1  1  0
## 40           1  1  0
## 41           1  1  0
## 42           1  1  0
```

```
## 43           1  1  0
## 44           1  1  0
## 45           1  1  0
## 46           1  0  1
## 47           1  0  1
## 48           1  0  1
## 49           1  0  1
## 50           1  0  1
## 51           1  0  1
## 52           1  0  1
## 53           1  0  1
## 54           1  0  1
## 55           1  0  1
## 56           1  0  1
## 57           1  0  1
## 58           1  0  1
## 59           1  0  1
## 60           1  0  1
## 61           1  0  1
## 62           1  0  1
## 63           1  0  1
## 64           1  0  1
## 65           1  0  1
## 66           1  0  1
## 67           1  0  1
## 68           1  0  1
## 69           1  0  1
## 70           1  0  1
## 71           1  0  1
## 72           1  0  1
## 73           1  0  1
## 74           1  0  1
## 75           1  0  1
## 76           1  0  1
## attr(,"assign")
## [1] 0 1 2
```

- We can see that the sum of the second and third columns equals the first column.
- This means that X is not of full rank, having a rank of two, not three.
- Hence not all the parameters can be identifide.

We have more parameteres than we need so the solution is to get rid of one of them. Once choice would be to eliminate d_1:

```
lmod <- lm(ptsd ~ d2, sexab)
summary(lmod)
```

```
##
## Call:
## lm(formula = ptsd ~ d2, data = sexab)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0451 -2.3123  0.0951  2.1645  7.0514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  11.9411     0.5177  23.067  < 2e-16 ***
## d2           -7.2452     0.8105  -8.939 2.17e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.473 on 74 degrees of freedom
## Multiple R-squared:  0.5192, Adjusted R-squared:  0.5127
## F-statistic:  79.9 on 1 and 74 DF,  p-value: 2.172e-13
```

Compare to the output of the t-test:

```
t.test(ptsd ~ csa, sexab, var.equal = TRUE)
```

```
##
##  Two Sample t-test
##
## data:  ptsd by csa
## t = 8.9387, df = 74, p-value = 2.172e-13
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  5.630165 8.860273
## sample estimates:
##    mean in group Abused mean in group NotAbused
##               11.941093                4.695874
```

- The intercept of 11.941 is the eman of the first group ("Abused").
- The parameter for d_2 represents the difference between the second and the first group, i.e., 11.941 - 7.245 = 4.694.
- The t-value for d_2 of -8.94 is the test statistic for the test that the difference is zero and is identical (excepting the sign) to the test statistic from the t-test.
- One assumption of the linear model is that the variacnes of the errors are equal which explains why we specified this option when computing the t-test earlier.

An alternative approach is to eliminate the intercept term:

```
lmod <- lm(ptsd ~ d1 + d2 -1, sexab)
summary(lmod)
```

```
##
## Call:
## lm(formula = ptsd ~ d1 + d2 - 1, data = sexab)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -8.0451 -2.3123  0.0951  2.1645  7.0514
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## d1  11.9411     0.5177  23.067   <2e-16 ***
## d2   4.6959     0.6237   7.529    1e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.473 on 74 degrees of freedom
## Multiple R-squared:  0.8883, Adjusted R-squared:  0.8853
## F-statistic: 294.4 on 2 and 74 DF,  p-value: < 2.2e-16
```

Advantages:

- The means of the two groups are directly supplied by the parameter estiamtes of the two dummy variables.

Disadvantages:

- We do not get the t-test for the difference.
- The tests in the output correspond to hypotheses claiming the mean response in the group is zero.
- These are not interesting because these hypotheses are unbelievable.
- The solution of dropping the intercept only works when there is a single factor and does not generalize to the multiple factor case.
- The R^2 is not correctly computed when the intercept is omitted.

For these reasons, we prefer approach of dropping one of the dummy variables to dropping the intercept.

It is not necessary to explicitly form the dummy variables as R can produce these directly by just including the factor in the model formula:

```
lmod <- lm(ptsd ~ csa, sexab)
summary(lmod)
```

```
##
## Call:
## lm(formula = ptsd ~ csa, data = sexab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0451 -2.3123  0.0951  2.1645  7.0514
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   11.9411     0.5177  23.067  < 2e-16 ***
## csaNotAbused  -7.2452     0.8105  -8.939 2.17e-13 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.473 on 74 degrees of freedom
## Multiple R-squared:  0.5192, Adjusted R-squared:  0.5127
## F-statistic:  79.9 on 1 and 74 DF,  p-value: 2.172e-13
```

We can check that `csa` is a factor variable:

```
class(sexab$csa)
```

```
## [1] "factor"
```

- This usually happens automatically when a variable takes non-numeric values.
- It can be imposed directly if necessary using the `factor()` command.
- The dummy variables are created but one is dropped to ensure identifiability.
- This is known as the *reference level*.
- In this example, the reference level is "Abused".
- The mean response for the reference level is encoded in the intercept of 11.941.
- The parameter estimate for "NotAbused" of -7.245 is the difference from the reference level.
- Hence the mean response for the "NotAbused" level is 11.941 - 7.245 = 4.696.

Reference Levels:

- The choice of reference level is arbitrary.
- The default choice of reference level by R is the first level in alphabetical order.
- Because this choice is inconvenient, we change the reference level using the `relevel` command.

Change the reference level using the `relevel`:

```
sexab$csa <- relevel(sexab$csa, ref = "NotAbused")
lmod <- lm(ptsd ~ csa, sexab)
summary(lmod)
```

```
##
## Call:
## lm(formula = ptsd ~ csa, data = sexab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0451 -2.3123  0.0951  2.1645  7.0514
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   4.6959     0.6237   7.529 1.00e-10 ***
## csaAbused     7.2452     0.8105   8.939 2.17e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.473 on 74 degrees of freedom
## Multiple R-squared:  0.5192, Adjusted R-squared:  0.5127
## F-statistic:  79.9 on 1 and 74 DF,  p-value: 2.172e-13
```

A comparison of the outputs reveals that the fitted values and residuals are the same for either choice. But the parametrization is different.

Factors and Quantitative Predictors

Suppose we have a response y, a quantitative predictor x and a two-level factor variable represented by a dummy variable d:

$$d = \begin{cases} 0 & \text{reference level} \\ 1 & \text{treatment level} \end{cases}$$

Several possible linear models may be considered here:

1. The same regression line for both levels: y ~ x
2. A factor predictor but no quantitative predictor: y ~ d
3. Separate regression lines for each group with the same slope: y ~ x + d
4. Separate regression lines for each group with the different slopes: y ~ x + d + x:d or y ~ x*d

We start with the separate regression lines model:

```
lmod4 <- lm(ptsd~ cpa + csa +cpa:csa, sexab)
summary(lmod4)
```

```
##
## Call:
## lm(formula = ptsd ~ cpa + csa + cpa:csa, data = sexab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1999 -2.5313 -0.1807  2.7744  6.9748
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6959     0.7107   5.201 1.79e-06 ***
## cpa             0.7640     0.3038   2.515   0.0142 *
## csaAbused       6.8612     1.0747   6.384 1.48e-08 ***
## cpa:csaAbused  -0.3140     0.3685  -0.852   0.3970
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.279 on 72 degrees of freedom
## Multiple R-squared:  0.5828, Adjusted R-squared:  0.5654
## F-statistic: 33.53 on 3 and 72 DF,  p-value: 1.133e-13
```

The model can be simplified becasue the interaction term is not significant.

We can discover the coding by examining the X-matrix:

```
model.matrix(lmod4)
```

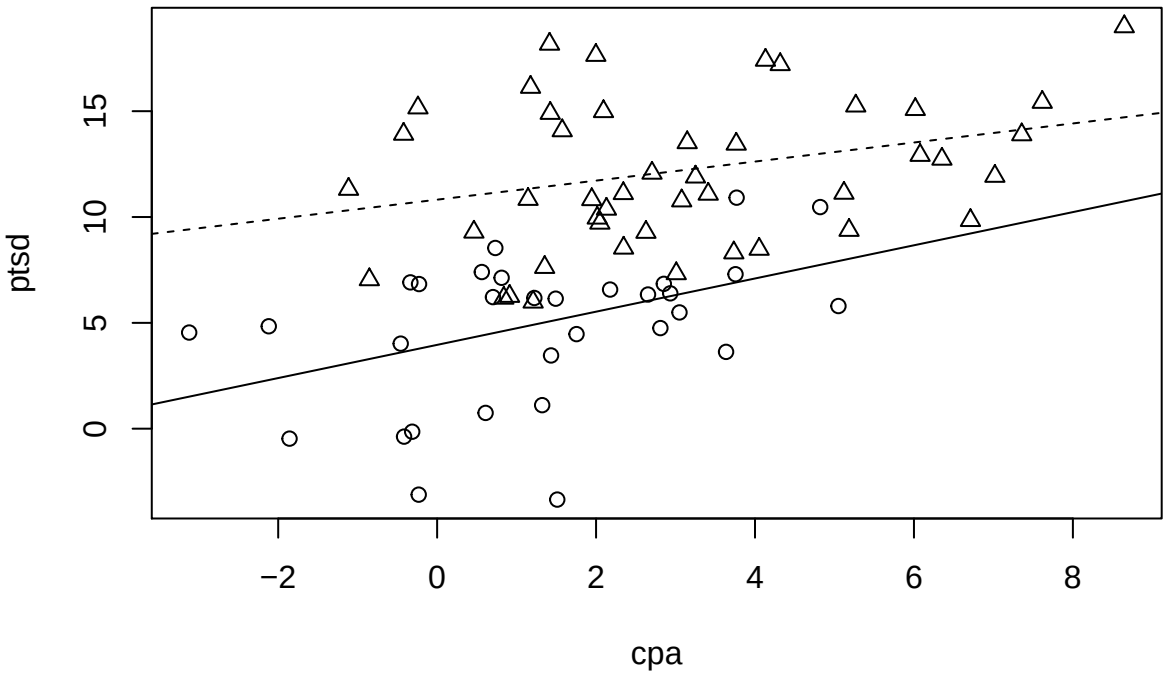
```
##    (Intercept)      cpa csaAbused cpa:csaAbused
## 1            1  2.04786         1       2.04786
## 2            1  0.83895         1       0.83895
## 3            1 -0.24139         1      -0.24139
## 4            1 -1.11461         1      -1.11461
## 5            1  2.01468         1       2.01468
## 6            1  6.71131         1       6.71131
## 7            1  1.20814         1       1.20814
## 8            1  2.34284         1       2.34284
## 9            1  0.91188         1       0.91188
## 10           1 -0.85308         1      -0.85308
## 11           1  7.35666         1       7.35666
## 12           1  2.09361         1       2.09361
## 13           1  1.94568         1       1.94568
## 14           1 -0.42222         1      -0.42222
## 15           1  1.41462         1       1.41462
## 16           1  6.07760         1       6.07760
## 17           1  6.01699         1       6.01699
## 18           1  3.73342         1       3.73342
## 19           1  2.62746         1       2.62746
## 20           1  0.46258         1       0.46258
## 21           1  7.01843         1       7.01843
## 22           1  3.14657         1       3.14657
## 23           1  2.34643         1       2.34643
## 24           1  8.64690         1       8.64690
## 25           1  4.31689         1       4.31689
## 26           1  2.13049         1       2.13049
## 27           1  4.05211         1       4.05211
## 28           1  1.57414         1       1.57414
## 29           1  3.76375         1       3.76375
## 30           1  5.18354         1       5.18354
## 31           1  1.17564         1       1.17564
## 32           1  2.70402         1       2.70402
## 33           1  1.14423         1       1.14423
## 34           1  4.13158         1       4.13158
## 35           1  1.42299         1       1.42299
## 36           1  6.35229         1       6.35229
## 37           1  7.61474         1       7.61474
## 38           1  1.99700         1       1.99700
## 39           1  3.25103         1       3.25103
## 40           1  3.00905         1       3.00905
## 41           1  3.07750         1       3.07750
## 42           1  5.26785         1       5.26785
```

```
## 43           1  3.41136         1       3.41136
## 44           1  1.35316         1       1.35316
## 45           1  5.11921         1       5.11921
## 46           1  1.49181         0       0.00000
## 47           1  0.60961         0       0.00000
## 48           1  1.43335         0       0.00000
## 49           1 -0.33664         0       0.00000
## 50           1 -3.12036         0       0.00000
## 51           1  2.65339         0       0.00000
## 52           1  3.75443         0       0.00000
## 53           1  1.51153         0       0.00000
## 54           1  1.75392         0       0.00000
## 55           1 -0.45860         0       0.00000
## 56           1  0.70258         0       0.00000
## 57           1  5.04974         0       0.00000
## 58           1  0.73195         0       0.00000
## 59           1 -0.41639         0       0.00000
## 60           1  2.80928         0       0.00000
## 61           1  2.93373         0       0.00000
## 62           1 -0.22780         0       0.00000
## 63           1  4.82039         0       0.00000
## 64           1  1.32165         0       0.00000
## 65           1  0.56215         0       0.00000
## 66           1  1.22299         0       0.00000
## 67           1  3.04951         0       0.00000
## 68           1  3.76859         0       0.00000
## 69           1 -2.11876         0       0.00000
## 70           1  3.63574         0       0.00000
## 71           1 -0.31402         0       0.00000
## 72           1  2.17626         0       0.00000
## 73           1 -0.23208         0       0.00000
## 74           1 -1.85753         0       0.00000
## 75           1  2.85253         0       0.00000
## 76           1  0.81138         0       0.00000
## attr(,"assign")
## [1] 0 1 2 3
## attr(,"contrasts")
## attr(,"contrasts")$csa
## [1] "contr.treatment"
```

The interaction term `cpa:csaAbused` = (2nd column) * (3rd column).

We showed that the fitted regression lines:

```
plot(ptsd ~ cpa, sexab, pch=as.numeric(csa))
abline(3.96, 0.784)
abline(3.96 + 6.86, 0.764-0.314, lty=2)
```



We reduce to this model:

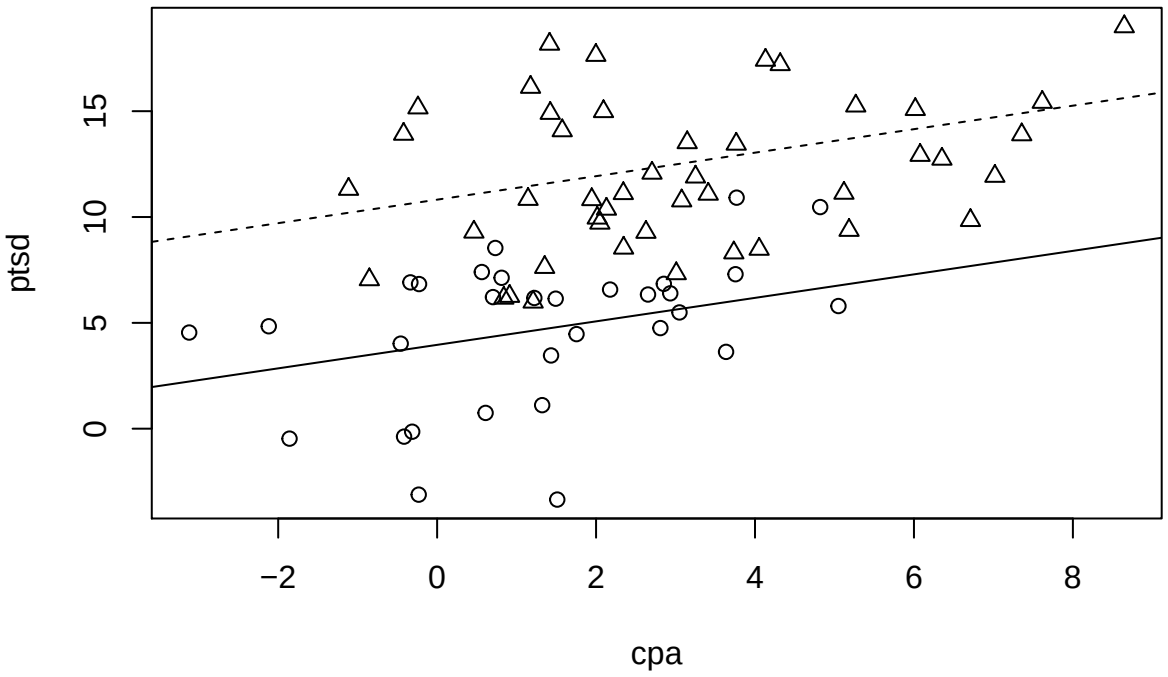
```
lmod3 <- lm(ptsd~ cpa + csa, sexab)
summary(lmod3)
```

```
##
## Call:
## lm(formula = ptsd ~ cpa + csa, data = sexab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1567 -2.3643 -0.1533  2.1466  7.1417
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.9753     0.6293   6.317 1.87e-08 ***
## cpa           0.5506     0.1716   3.209  0.00198 **
## csaAbused     6.2728     0.8219   7.632 6.91e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.273 on 73 degrees of freedom
## Multiple R-squared:  0.5786, Adjusted R-squared:  0.5671
## F-statistic: 50.12 on 2 and 73 DF,  p-value: 2.002e-14
```

No further simplification is possible because the remaining predictors are statistically significant.

Put the parallel regrssion lines on the plot:

```
plot(ptsd ~ cpa, sexab, pch=as.numeric(csa))
abline(3.96, 0.5551)
abline(3.96 + 6.86, 0.5551, lty=2)
```



- The slope of bothe lines is 0.5551, but the "Abused" line is 6.273 higher than the "NonAbused."
- From the t-test earlier, the unadjusted estimated effect of childhood sexual abuse is 7.245.
- So after adjusting for the effect of childhood physical abuse, our estimate of the effect of childhood sexual abuse on PTSD is midlly reduced.

We can also compare confidence interval for the effect of `csa`:

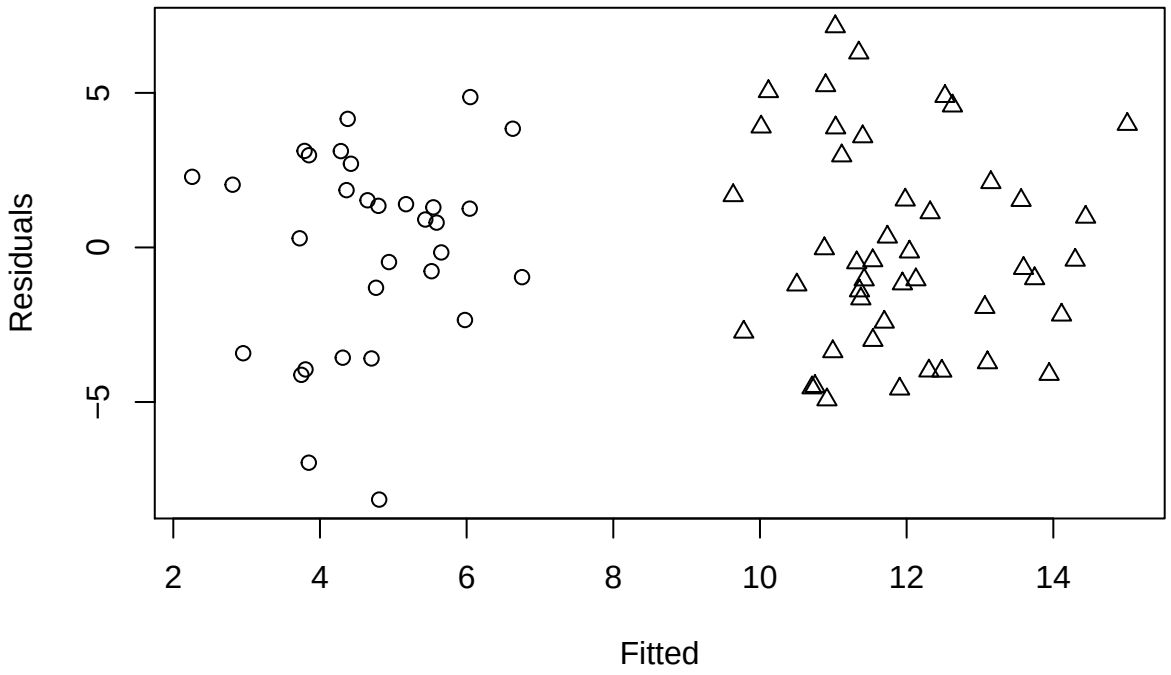
```
confint(lmod3)[3,]
```

```
##    2.5 %   97.5 %
## 4.634696 7.910809
```

- Compare to the (5.6302, 8.8603) found for the unadjusted difference.
- The confidence intervals are about the same width.

The usual diagnostics should be checked. It is worth checking whether there is ssome difference related to the categorical variable:

```
plot(fitted(lmod3),residuals(lmod3),pch=as.numeric(sexab$csa),
     xlab="Fitted",ylab="Residuals")
```



- We see that there are no clear problems.
- The variation in the two group is about the same.
- If this were not so, we would need to make some adjustments to the analysis, possibly using weights.

We have seen that the effect of `csa` can be adjusted for `cpa`. The reverse is also true. Consider a model with just `cpa`:

```
lmod1 <- lm(ptsd ~ cpa, sexab)
summary(lmod1)
```

```
##
## Call:
## lm(formula = ptsd ~ cpa, data = sexab)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.4636  -2.3855  -0.1246   2.2610  10.1543
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.5523     0.7072   9.265 5.27e-14 ***
## cpa           1.0334     0.2124   4.865 6.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.359 on 74 degrees of freedom
## Multiple R-squared:  0.2424, Adjusted R-squared:  0.2321
## F-statistic: 23.67 on 1 and 74 DF,  p-value: 6.272e-06
```

After adjusting for the effect of `csa`, we see size of the effect of `cpa` is reduced from 1.044 to 0.551.

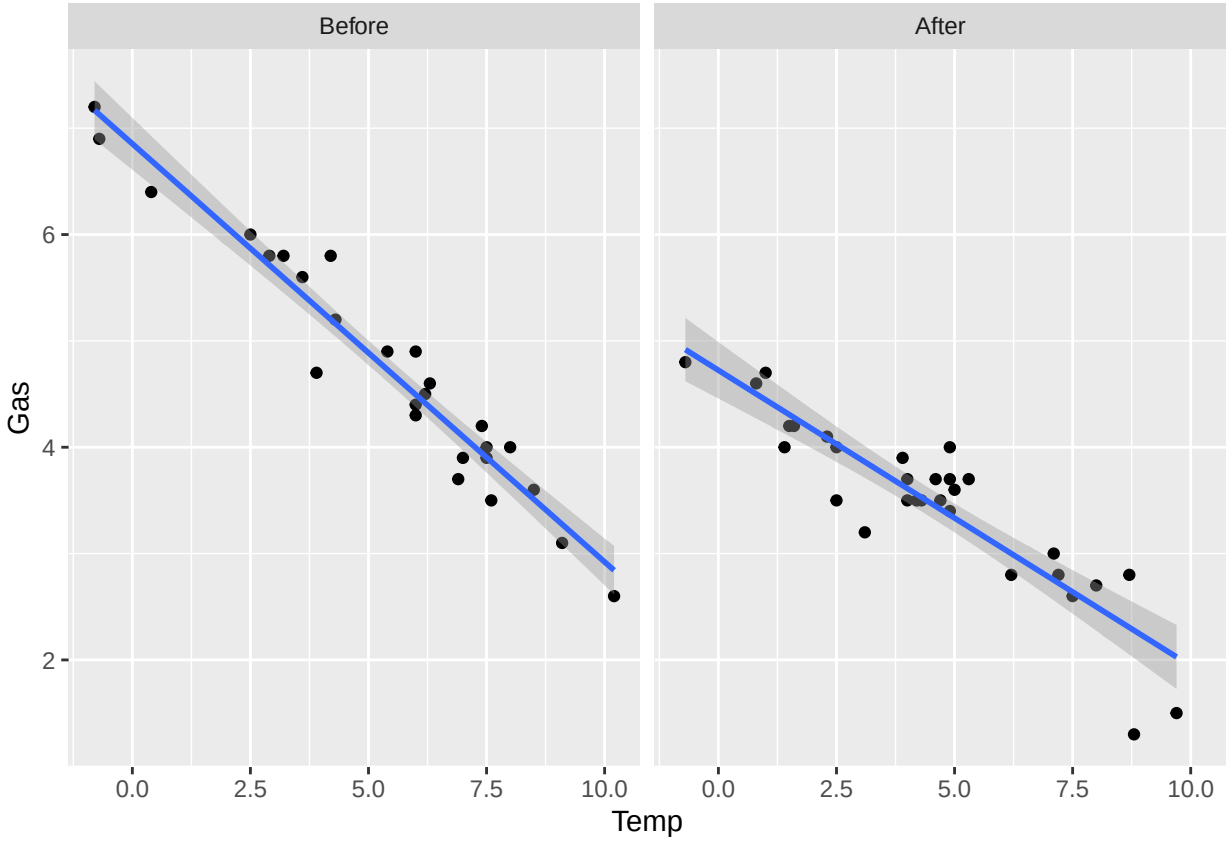
Interpretation with Interaction Term

```
data(whiteside)
```

We plot the data:

```
ggplot(aes(x=Temp,y=Gas),data=whiteside)+
  geom_point()+
  facet_grid(~Insul)+
  geom_smooth(method="lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



We can see that less gas is used after the insulation is installed but the difference varies by temperature.

The relationships appear linear so we fit a model:

```
lmod <- lm(Gas ~ Temp*Insul, whiteside)
summary(lmod)
```

```
##
## Call:
## lm(formula = Gas ~ Temp * Insul, data = whiteside)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.97802 -0.18011  0.03757  0.20930  0.63803
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.85383    0.13596  50.409  < 2e-16 ***
## Temp            -0.39324    0.02249 -17.487  < 2e-16 ***
## InsulAfter      -2.12998    0.18009 -11.827 2.32e-16 ***
## Temp:InsulAfter  0.11530    0.03211   3.591 0.000731 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.323 on 52 degrees of freedom
## Multiple R-squared:  0.9277, Adjusted R-squared:  0.9235
## F-statistic: 222.3 on 3 and 52 DF,  p-value: < 2.2e-16
```

- The gas consumption would fall by 0.393 for each 1 Cel. increase in temperature before insulation.
- After insulation, the fall in consumption per degree is only 0.393 - 0.115 = 0.278.
- The interpretation for the other two parameter estimates is more problematic since these represent prediced consumption when the temperature is zero.

The solution is to center the temperature predictor by its mean value and recompute the linear model:

```
mean(whiteside$Temp)
```

```
## [1] 4.875
```

```
whiteside$ctemp <- whiteside$Temp - mean(whiteside$Temp)
lmodc <- lm(Gas ~ ctemp*Insul, whiteside)
sumary(lmodc)
```

```
##                   Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)       4.936788   0.064241  76.8485 < 2.2e-16
## ctemp            -0.393239   0.022487 -17.4874 < 2.2e-16
## InsulAfter       -1.567872   0.087713 -17.8750 < 2.2e-16
## ctemp:InsulAfter  0.115304   0.032112   3.5907 0.0007307
##
## n = 56, p = 4, Residual SE = 0.32300, R-Squared = 0.93
```

- The average consumption before insulation at the average temperature was 4.94 and 4.94 - 1.57 = 3.37 afterwards.
- The other two coefficients are unchanged and their interpretation remains the same.
- Thus we can see that centering allows a more natural interpretation of parameter estimates in the presence of interaction.

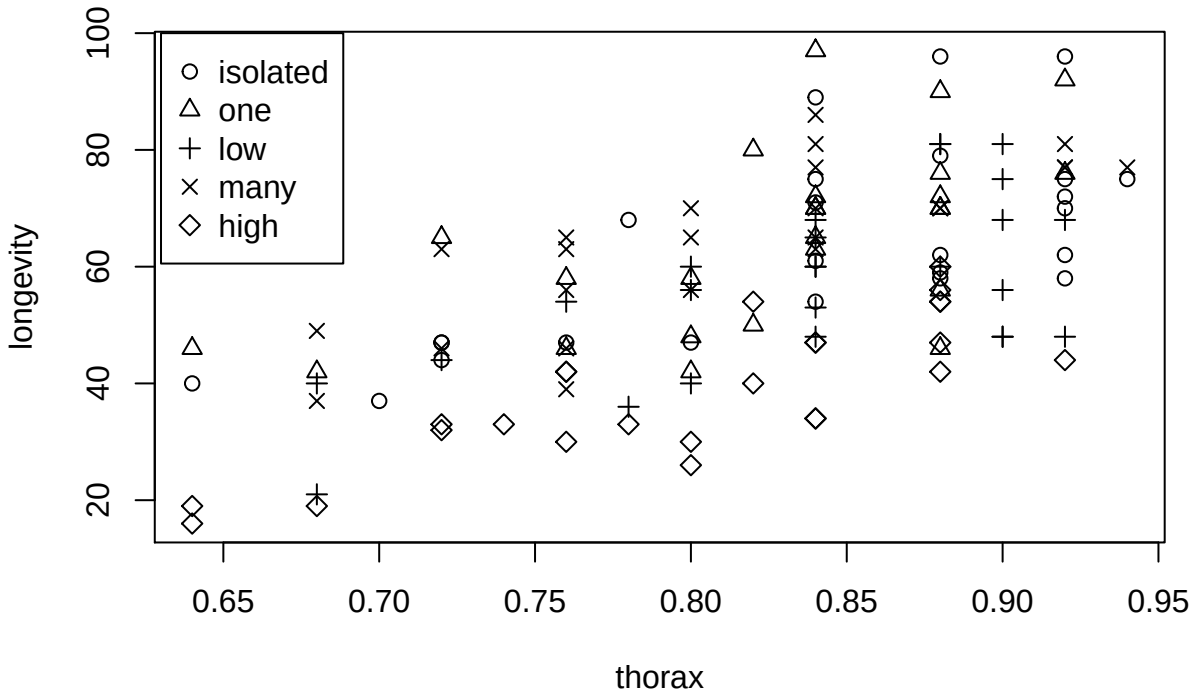
Factors With More Than Two Levels

Suppose we have a factor with f levels, then we create $f-1$ dummy variables $d_2, ..., d_j$ where:

$$d_i = \begin{cases} 0 & \text{is not level i} \\ 1 & \text{is level i} \end{cases}$$

We start with a plot of data:

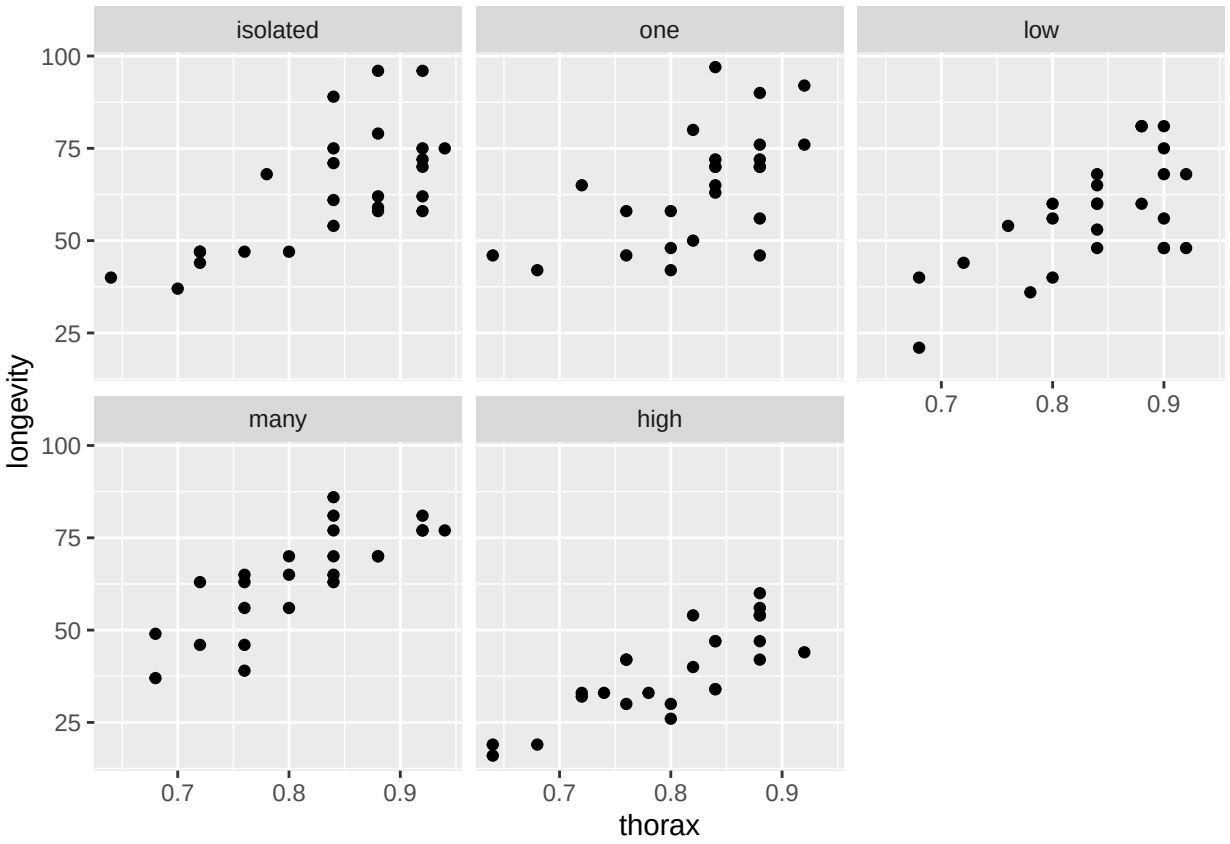
```
data("fruitfly")
plot(longevity ~ thorax, fruitfly, pch=unclass(activity))
legend(0.63,100,levels(fruitfly$activity),pch=1:5)
```



- With multiple levels, it can be hard to distinguish the groups.
- Sometimes it is better to plot each level separately.

This can be achieved nicely with the help of the `ggplot2` pacakge:

```
ggplot(aes(x=thorax,y=longevity),data=fruitfly) +
  geom_point() +
  facet_wrap( ~ activity)
```



The plot makes it clearer that longevity for the high sexual activity group is lower.

We fit and summarize the most genetral linear model:

```
lmod <- lm(longevity ~ thorax*activity, fruitfly)
summary(lmod)
```

```
##
## Call:
## lm(formula = longevity ~ thorax * activity, data = fruitfly)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.9509  -6.7296  -0.9103   6.1854  30.3071
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -50.2420    21.8012  -2.305    0.023 *
## thorax              136.1268    25.9517   5.245 7.27e-07 ***
## activityone           6.5172    33.8708   0.192    0.848
## activitylow          -7.7501    33.9690  -0.228    0.820
## activitymany         -1.1394    32.5298  -0.035    0.972
## activityhigh        -11.0380    31.2866  -0.353    0.725
## thorax:activityone   -4.6771    40.6518  -0.115    0.909
## thorax:activitylow    0.8743    40.4253   0.022    0.983
## thorax:activitymany   6.5478    39.3600   0.166    0.868
## thorax:activityhigh -11.1268    38.1200  -0.292    0.771
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.71 on 114 degrees of freedom
## Multiple R-squared:  0.6534, Adjusted R-squared:  0.626
## F-statistic: 23.88 on 9 and 114 DF,  p-value: < 2.2e-16
```

- Since "isolated" is the reference level, the fitted regression line within this group is `longevity` = -50.2 + 136.1*`thorax`.
- For "many," it is `longevity` = (-50.2-1.1) + (136.1+6.5)*`thorax`.

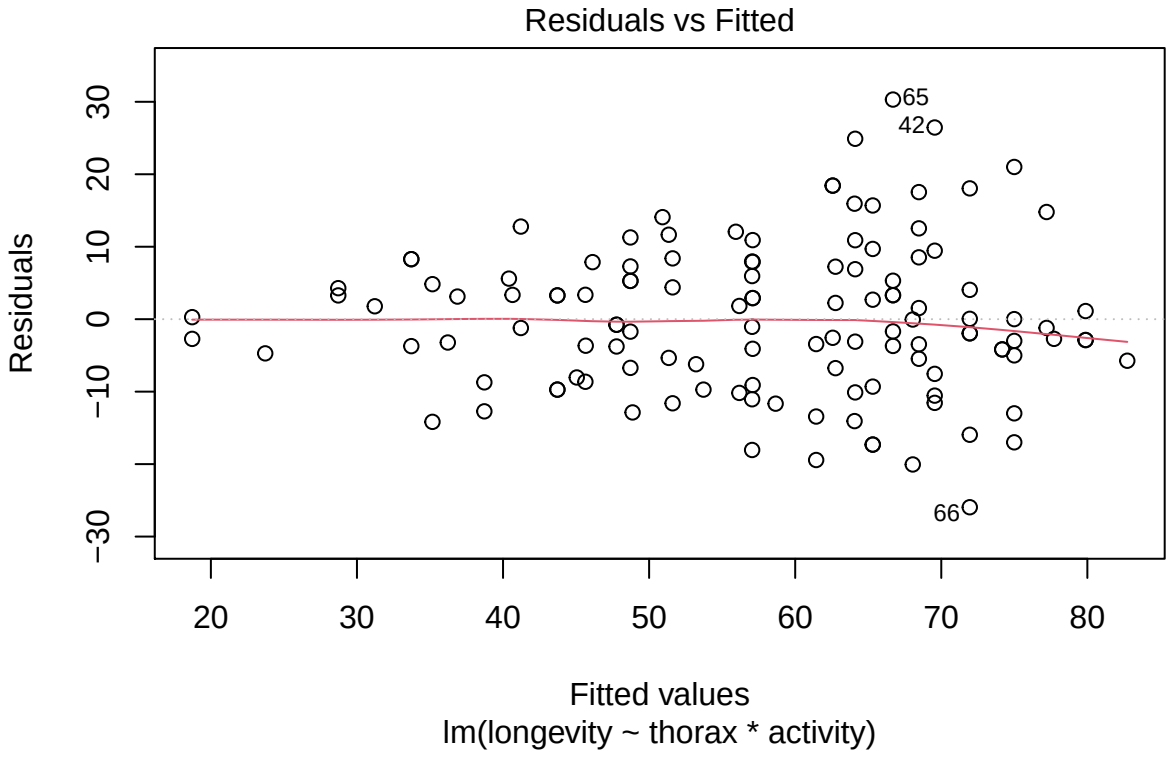
Examine:

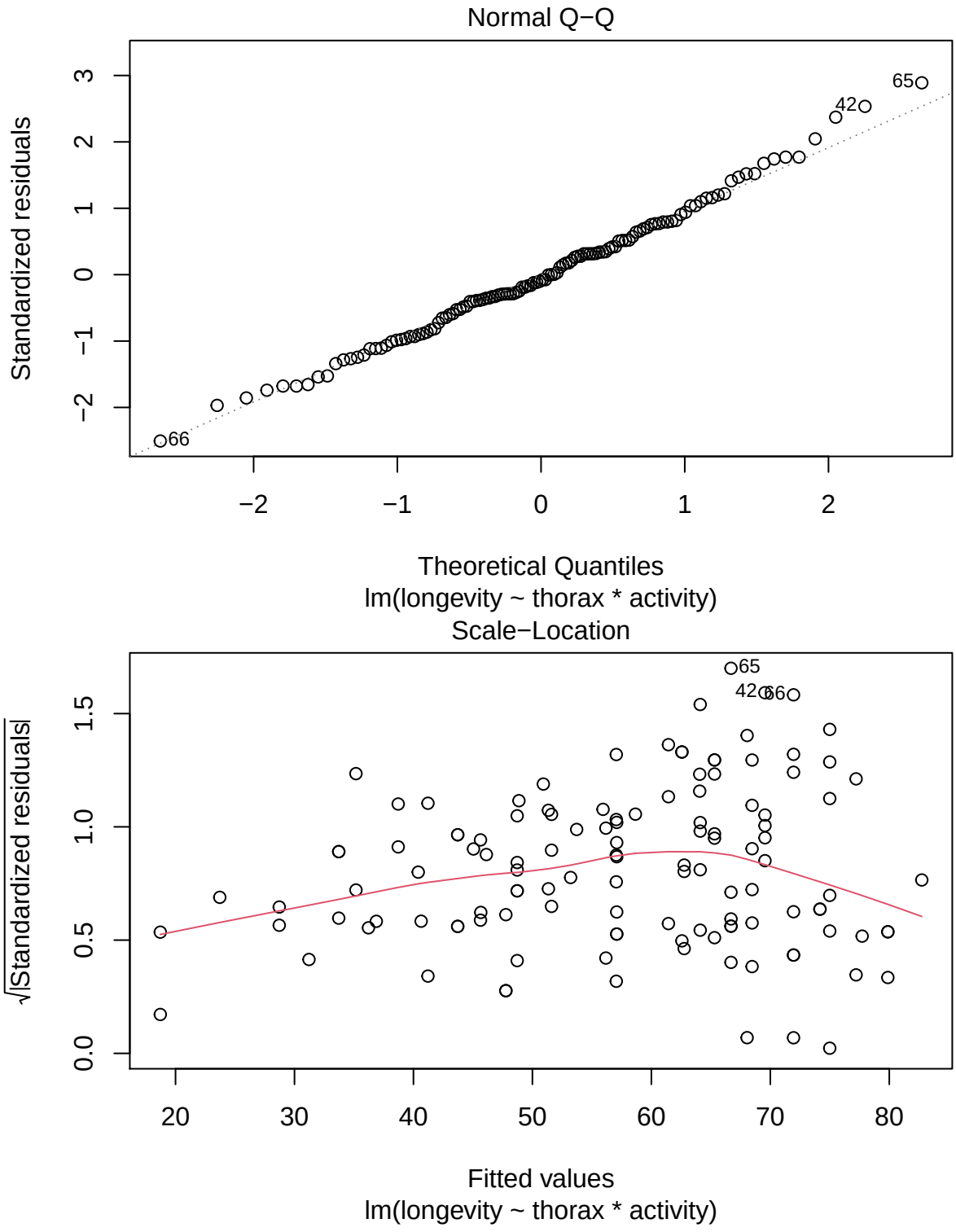
```
model.matrix(lmod)
```

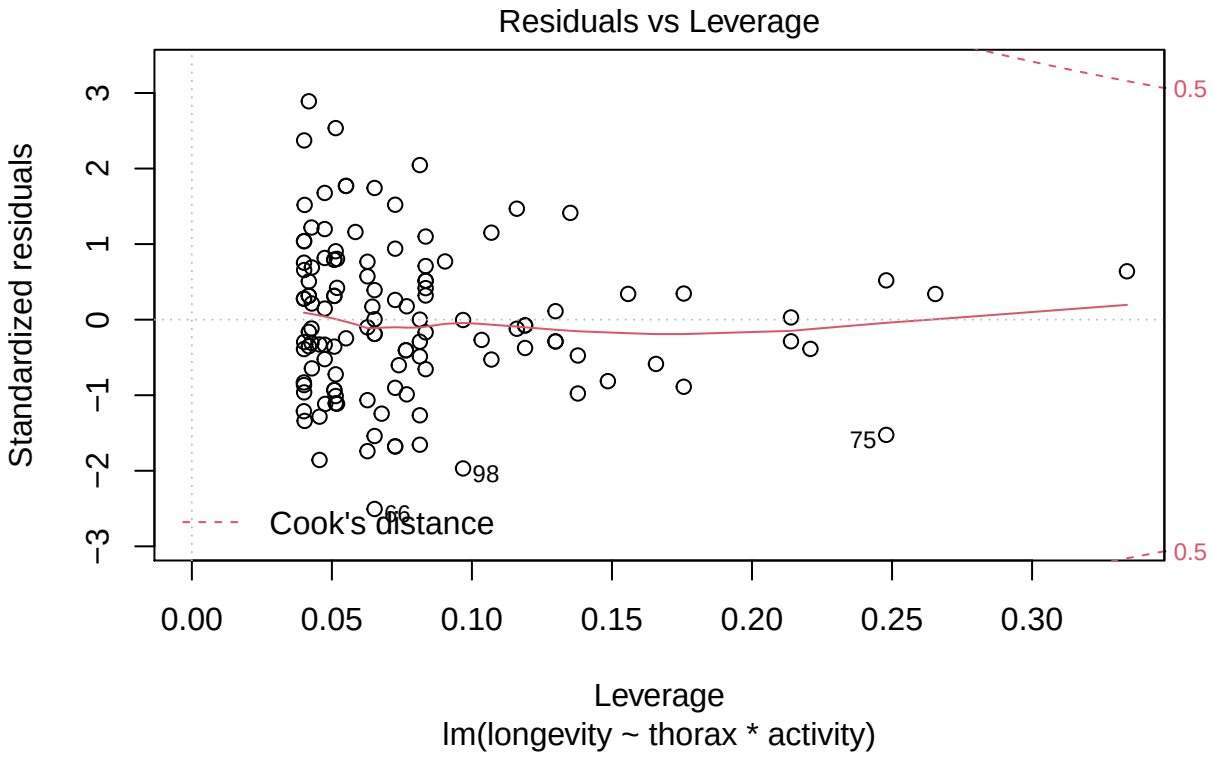
to see how the coding is done.

Some diagnostics should be examined by:

```
plot(lmod)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(longevity ~ thorax * activity)
Scale−Location
√|Standardized residuals|
Fitted values
lm(longevity ~ thorax * activity)
65
42
66



Now we see whether the model can be simplified. The model summary output is not suitable for this purpose because there are four t-tests corresponding to the interaction term while we want just a single test for this term.

We can obtain this using:

```
anova(lmod)
```

```
## Analysis of Variance Table
##
## Response: longevity
##                   Df  Sum Sq Mean Sq F value    Pr(>F)
## thorax             1 15003.3 15003.3 130.733 < 2.2e-16 ***
## activity           4  9634.6  2408.6  20.988 5.503e-13 ***
## thorax:activity    4    24.3     6.1   0.053    0.9947
## Residuals        114 13083.0   114.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- This is a sequential analysis of variance (ANOVA) table.
- Starting from a null model, terms are added and sequentially tested.
- The interactio term `thorax:activity` is not significant, indicating that we can the same slope within each group.
- No further simplication is possible.

We now refit without the interaction term:

```
lmodp <- lm(longevity ~ thorax+activity, fruitfly)
```

We might prefer to check whether each predictor is significant once the other has been taken into account.

We can do this using:

```
drop1(lmodp, test="F")
```

```
## Single term deletions
##
## Model:
## longevity ~ thorax + activity
##          Df Sum of Sq   RSS    AIC F value    Pr(>F)
## <none>                13107 589.92
## thorax    1   12368.4 25476 670.32 111.348 < 2.2e-16 ***
## activity  4    9634.6 22742 650.25  21.684 1.974e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `drop1()` command tests each term relative o the full model. This shows that both terms are significant even after allowing for the effect of the other.

Now examine the model coefficients:

```
summary(lmodp)
```

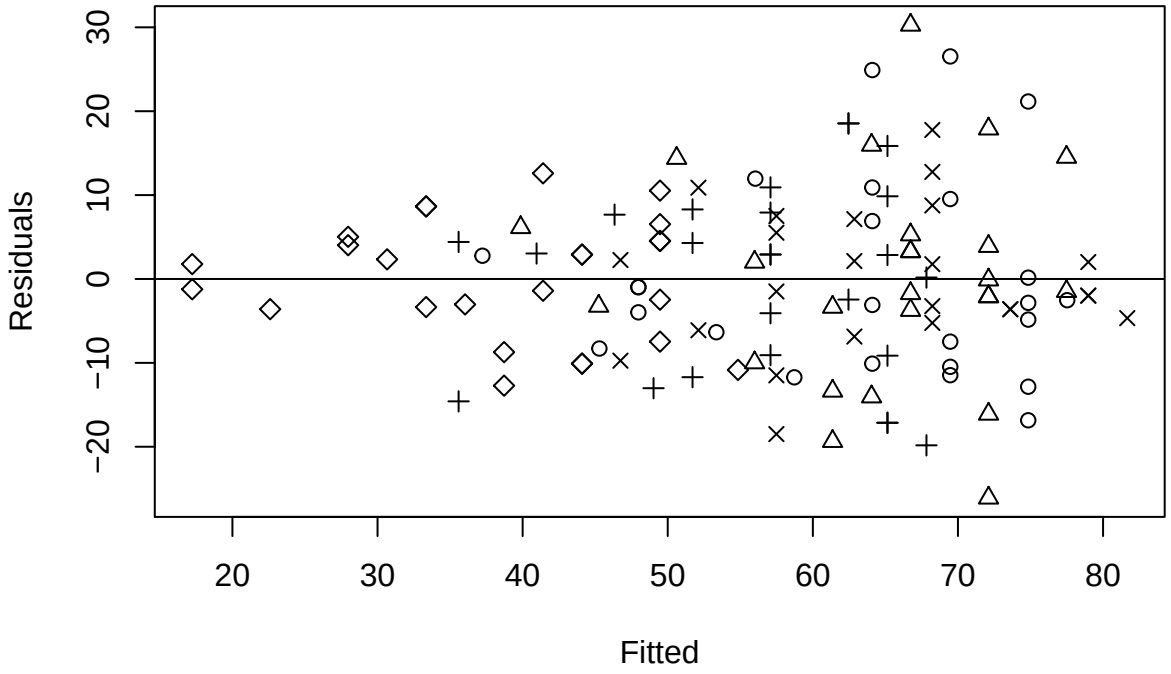
```
##
## Call:
## lm(formula = longevity ~ thorax + activity, data = fruitfly)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.108  -7.014  -1.101   6.234  30.265
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -48.749     10.850  -4.493 1.65e-05 ***
## thorax        134.341     12.731  10.552  < 2e-16 ***
## activityone     2.637      2.984   0.884   0.3786
## activitylow    -7.015      2.981  -2.353   0.0203 *
## activitymany    4.139      3.027   1.367   0.1741
## activityhigh  -20.004      3.016  -6.632 1.05e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.54 on 118 degrees of freedom
## Multiple R-squared:  0.6527, Adjusted R-squared:  0.638
## F-statistic: 44.36 on 5 and 118 DF,  p-value: < 2.2e-16
```

- "Isolated" is the reference level.
- The intercepts of "one" and "many" are not significantly different from this reference level.

- The low sexual activity group, "low," survives about seven days less.
- The p-value is 0.02 and is enough for statistical significance if only one comparison is made.
- However, we are making more than one comparison, and so, as with outliers, a Bonferroni-type adjustment might be considered. This would erase the statistical significance of the difference.
- However, the high sexual activity group, "high," has a life span 20 days less than the reference group and this is strongly significant.

Returning to the diagnostics:

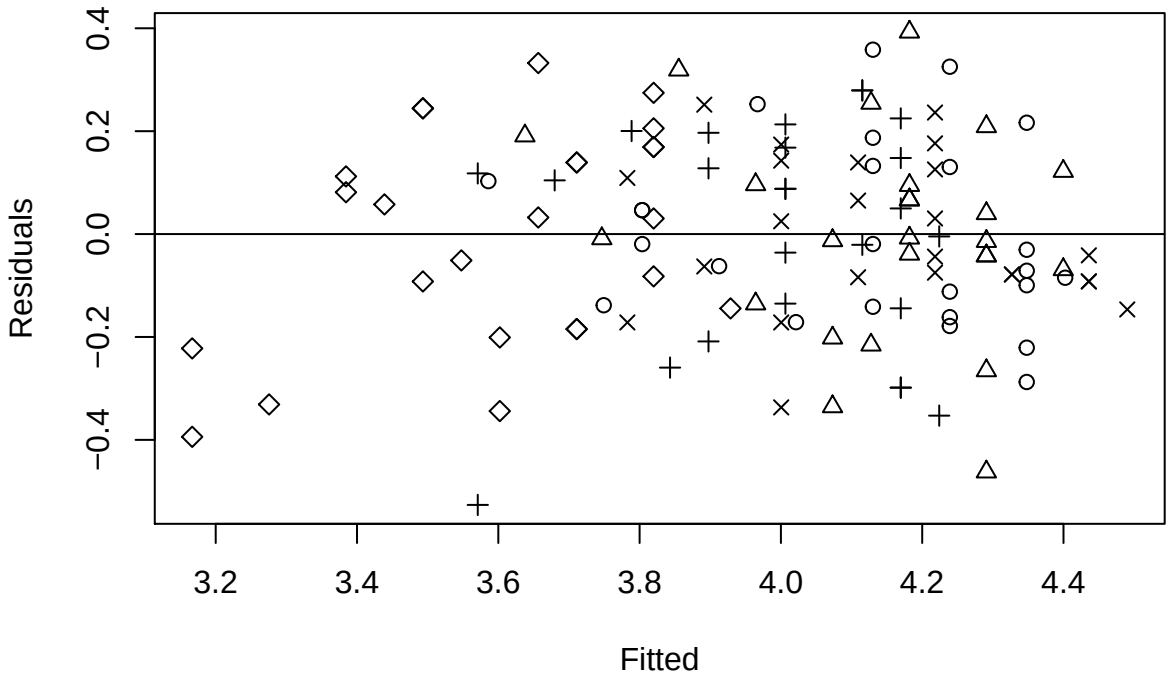
```
plot(residuals(lmodp) ~fitted(lmodp),pch=unclass(fruitfly$activity),
     xlab="Fitted",ylab="Residuals")
abline(h=0)
```



We have some non-constant variance although it does not appear to be related to the five groups.

A log transformation can remove the heteroscedasticity:

```
lmod1 <- lm(log(longevity) ~ thorax+activity, fruitfly)
plot(residuals(lmod1) ~ fitted(lmod1),pch=unclass(fruitfly$activity),
     xlab="Fitted",ylab="Residuals")
abline(h=0)
```



One disadvantage of transformation is that it can make interpretation of the model more difficult.

Let's examine the model fit:

```
summary(lmod1)
```

```
##
## Call:
## lm(formula = log(longevity) ~ thorax + activity, data = fruitfly)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.52641 -0.13629 -0.00823  0.13918  0.39273
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.84421    0.19882   9.276 1.04e-15 ***
## thorax        2.72146    0.23329  11.666  < 2e-16 ***
## activityone   0.05174    0.05468   0.946   0.3459
## activitylow  -0.12387    0.05463  -2.268   0.0252 *
## activitymany  0.08791    0.05546   1.585   0.1156
## activityhigh -0.41925    0.05527  -7.586 8.35e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1931 on 118 degrees of freedom
## Multiple R-squared:  0.7025, Adjusted R-squared:  0.6899
## F-statistic: 55.72 on 5 and 118 DF,  p-value: < 2.2e-16
```

Notice that the R^2 is higher for this model, but the p-values are similar.

Because of the log transformation, we can interpret the coefficients as having a multiplicative effect:

```
exp(coef(lmod1)[3:6])
```

```
##  activityone  activitylow activitymany activityhigh
##    1.0531064    0.8834971    1.0918894    0.6575384
```

Compared to the reference level, we see that the high sexual activity group has 0.66 times the life span (i.e., 34% less).

Why did we include `thorax` in the model?

Its effect on longevity was known, but because of the random assignment of the files to the groups, this variable will not bias the estimates of the effects of the activities.

We can verify that `thorax` is unrelated to the activities:

```
lmodh <- lm(thorax ~ activity, fruitfly)
anova(lmodh)
```

```
## Analysis of Variance Table
##
## Response: thorax
##            Df  Sum Sq  Mean Sq F value Pr(>F)
## activity    4 0.02555 0.006388  1.1092 0.3555
## Residuals 119 0.68532 0.005759
```

However, look what happens if we omit `thorax` from the model for `longevity`:

```
lmodu <- lm(log(longevity) ~ activity, fruitfly)
sumary(lmodu)
```

```
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   4.119349   0.056440 72.9860 < 2.2e-16
## activityone   0.023441   0.079819  0.2937    0.7695
## activitylow  -0.119513   0.079819 -1.4973    0.1370
## activitymany  0.023955   0.080646  0.2970    0.7670
## activityhigh -0.517225   0.079819 -6.4800 2.167e-09
##
## n = 124, p = 5, Residual SE = 0.28220, R-Squared = 0.36
```

The magnitudes of the effects do not change that much but the standard errors are substantially larger. The value of including `thorax` in this model is to increase the precision of the estimates.