MATH 60604A Statistical Modelling

Chapter 2 Part 2: Multiple Linear Regression

Juliana Schulz

---

0) The data

We'll consider the same example explored in part 1 of Chapter 2: modelling the intention to buy as a function of the duration of fixation. In part 1, we only considered fixation as an explanatory variable (simple linear regression). In reality, it's quite possible that the intention to buy is not only related to fixation, but to several other variables (e.g., age, revenue, etc.). The goal of the study was to examine the impact of both fixation and emotion on intention, while accounting for socio-demographic variables. The multiple linear regression model allows us to do this.

Recall the data:

```
intention<-read.csv("Data/intention.csv")
head(intention)
```

```
##     fix   emo sex age rev educ stat intent
## 1 0.081 1.417   1  27   1    2    0     11
## 2 2.235 1.146   0  27   1    1    0     12
## 3 1.675 0.296   1  26   1    2    1      6
## 4 0.630 0.731   1  34   3    3    0      4
## 5 2.197 0.841   1  30   1    2    1     11
## 6 0.424 0.334   0  29   3    3    1      4
```

The explanatory variables include `fix`, `emo`, `sex`, `age`, `rev`, `educ`, `stat`. While we ultimately want to include all of the explanatory variables, we'll begin by focusing on `fix`,`emo`,`age`,`sex` and `stat` (these are all either continuous or binary variables, which we know how to handle; we'll later see how to include the variables `rev` and `educ`, which are categorical variables).

1) The model

We can fit a multiple linear regression model including `fix`,`emo`,`age`,`sex` and `stat`:

```
mod<-lm(intent~fix+emo+age+sex+stat,data=intention)
summary(mod)
```

```
##
## Call:
## lm(formula = intent ~ fix + emo + age + sex + stat, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5740 -1.8221 -0.0377  1.4214  5.6091
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.2011     1.4579   6.997 1.91e-10 ***
## fix           1.1152     0.1999   5.579 1.66e-07 ***
## emo           1.3229     0.4149   3.188  0.00185 **
## age          -0.1843     0.0449  -4.104 7.66e-05 ***
## sex           1.2112     0.4450   2.722  0.00752 **
## stat         -0.2996     0.4410  -0.679  0.49825
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.349 on 114 degrees of freedom
## Multiple R-squared:  0.3861, Adjusted R-squared:  0.3592
## F-statistic: 14.34 on 5 and 114 DF,  p-value: 6.953e-11
```

The fitted model is given by

$$\widehat{intent} = 10.20 + 1.11\, fix + 1.32\, emo - 0.18\, age + 1.21\, sex - 0.29\, stat$$

Interpretations

- Fixation: when the fixation time increases by one second, the mean intention to buy score increases by 1.11 points, *when all other variables are held constant.*
- Age: for every one year increase in age, the intention to buy will decrease by 0.18 points, on average, *holding all other variables fixed.*
- Sex: the mean difference in the intention to buy score between women (`sex`=1) and men (`sex`=0) is 1.21 points, *when all other variable in the model are held constant.* In other words, the difference in the intention score between a woman and man of the same age, same martial status, who had the same fixation duration and the same emotion, is on average 1.21.
- Etc. . .

Hypothesis tests

The R output provides the estimated regression coefficients $\hat{\beta}_j$ along with corresponding p-values for the test

$$H_0 : \beta_j = 0 \quad \text{vs.} \quad H_1 : \beta_j \neq 0$$

Recall the output:

```
summary(mod)
```

```
##
## Call:
## lm(formula = intent ~ fix + emo + age + sex + stat, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5740 -1.8221 -0.0377  1.4214  5.6091
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.2011     1.4579   6.997 1.91e-10 ***
## fix           1.1152     0.1999   5.579 1.66e-07 ***
## emo           1.3229     0.4149   3.188  0.00185 **
## age          -0.1843     0.0449  -4.104 7.66e-05 ***
```

```
## sex           1.2112     0.4450   2.722  0.00752 **
## stat         -0.2996     0.4410  -0.679  0.49825
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.349 on 114 degrees of freedom
## Multiple R-squared:  0.3861, Adjusted R-squared:  0.3592
## F-statistic: 14.34 on 5 and 114 DF,  p-value: 6.953e-11
```

Recall: In the case of multiple linear regression, this tests the **marginal** contribution of X_j when all other variables are included in the model. From the R output, we see that both fixation and emotion have small p-values (1.66×10^{-7} and 0.00185, respectively). Thus, we can reject $H_0 : \beta_j = 0$ in favor of $H_1 : \beta_j \neq 0$ for, say, $\alpha = 0.01$. Thus, we can conclude that the variables fixation and emotion of a significant impact on the intention to buy, while controlling for socio-demographic variables (age, sex, status). (That is, the effects of fixation and emotion are significantly different from 0.) Both $\hat{\beta}_1$ (estimated effect of fixation) and $\hat{\beta}_2$ (estimated effect of emotion) are positive, and thus we can say that

- as fixation increases, the intention to buy tends to increase as well (holding all other variables in the model fixed)
- as emotion increases, the intention to buy tends to increase (holding all other variables in the model fixed)

2) Categorical variables

In the model considered so far for the intention to buy, we did not include the variables `rev` and ‘educ. These variables are categorical, each with 3 categories:

- `rev`: annual income of the subject (1=[0, 20 000], 2=[20 000, 60 000], 3=[60 000, 60 0000 +])
- `educ`: education level of the subject (1=less than high school, 2=high school, 3=university)

We’ll begin by exploring how to incude `educ` in the model. To start, we’ll only include education as an explanatory variable, but remember this is NOT a simple linear regression model since `educ` has 3 levels and thus will be represented by 2 dummy variables in the model.

We’ll begin by creating the indicator variables `educ1`, `educ2`, `educ3` to respectively denote levels 1, 2, and 3 of the variable `educ`:

```
# creation de variables indicatrice
# creating indicator variables
intention$educ1<-as.numeric(intention$educ==1)
intention$educ2<-as.numeric(intention$educ==2)
intention$educ3<-as.numeric(intention$educ==3)
head(intention)
```

```
##     fix   emo sex age rev educ stat intent educ1 educ2 educ3
## 1 0.081 1.417   1  27   1    2    0     11     0     1     0
## 2 2.235 1.146   0  27   1    1    0     12     1     0     0
## 3 1.675 0.296   1  26   1    2    1      6     0     1     0
## 4 0.630 0.731   1  34   3    3    0      4     0     0     1
## 5 2.197 0.841   1  30   1    2    1     11     0     1     0
## 6 0.424 0.334   0  29   3    3    1      4     0     0     1
```

```
# verification
attach(intention)
table(educ,educ1)
```

```
##     educ1
```

```
## educ  0  1
##    1  0 30
##    2 55  0
##    3 35  0
```

```
table(educ,educ2)
```

```
##     educ2
## educ  0  1
##    1 30  0
##    2  0 55
##    3 35  0
```

```
table(educ,educ3)
```

```
##     educ3
## educ  0  1
##    1 30  0
##    2 55  0
##    3  0 35
```

```
summary(educ1+educ2+educ3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

To start, we'll fit the model which includes only `educ1` and `educ2`:

```
mod2.1<-lm(intent~educ1+educ2,data=intention)
summary(mod2.1)
```

```
##
## Call:
## lm(formula = intent ~ educ1 + educ2, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7091 -2.1143 -0.7091  2.2909  5.8857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.1143     0.4842  14.691   <2e-16 ***
## educ1         1.6524     0.7128   2.318   0.0222 *
## educ2         1.5948     0.6194   2.575   0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.865 on 117 degrees of freedom
## Multiple R-squared:  0.06316,    Adjusted R-squared:  0.04714
## F-statistic: 3.944 on 2 and 117 DF,  p-value: 0.022
```

The fitted model has the form

$$\widehat{intent} = 7.11 + 1.65\, educ1 + 1.59\, educ2$$

and thus

$$\widehat{E}(intention|educ = 1) = \hat{\beta}_0 + \hat{\beta}_1 = 8.77$$
$$\widehat{E}(intention|educ = 2) = \hat{\beta}_0 + \hat{\beta}_2 = 8.71$$
$$\widehat{E}(intention|educ = 3) = \hat{\beta}_0 = 7.11$$

- Thus, on average, the intention to buy is 1.65 units higher for education level 1 as compared to education level 3. This difference is significantly different from 0 (p-value 0.022 for the test $H_0 : \beta_1 = 0$ vs. $H_1 : \beta_1 \neq 0$).
- On average, the intention to buy is 1.59 units higher for education level 2 as compared to education level 3. This difference is significantly different from 0 (p-value 0.011 for the test $H_0 : \beta_2 = 0$ vs. $H_1 : \beta_2 \neq 0$).

Here, the reference level is 3, and thus all comparisons are made with respect to level 3. If we wish to assess whether or not there is a significant difference in the mean intention to buy for educ=1 vs. educ=2, we would be interested in testing $H_0 : \beta_1 - \beta_2 = 0$ vs. $H_1 : \beta_1 - \beta_2 \neq 0$. Note that while we can *estimate* this difference ($\hat{\beta}_1 - \hat{\beta}_2 = 1.6524 - 1.5948 = 0.0576$), we do not directly have a p-value for testing $H_0 : \beta_1 - \beta_2 = 0$. One approach is to simply consider a reparameterization of the model, using a different reference level. For example, we can consider setting level 2 as the reference and then include `educ1` and `educ3` as covariates in the model:

```
mod2.2<-lm(intent~educ1+educ3,data=intention)
summary(mod2.2)
```

```
##
## Call:
## lm(formula = intent ~ educ1 + educ3, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7091 -2.1143 -0.7091  2.2909  5.8857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.70909    0.38629  22.545   <2e-16 ***
## educ1        0.05758    0.65023   0.089   0.9296
## educ3       -1.59481    0.61945  -2.575   0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.865 on 117 degrees of freedom
## Multiple R-squared:  0.06316,    Adjusted R-squared:  0.04714
## F-statistic: 3.944 on 2 and 117 DF,  p-value: 0.022
```

The results are the same, just parameterized differently:

$$\widehat{E}(intention|educ = 1) = \hat{\beta}_0^* + \hat{\beta}_1^* = 8.77$$
$$\widehat{E}(intention|educ = 2) = \hat{\beta}_0^* = 8.71$$
$$\widehat{E}(intention|educ = 3) = \hat{\beta}_0^* + \hat{\beta}_2^* = 7.11$$

From this model, we directly see that the estimated difference between the mean intention for education levels 1 and 2 is

$$\widehat{E}(intention|educ = 1) - \widehat{E}(intention|educ = 2) = \hat{\beta}_1^* = 0.05758$$

And we can test whether this difference is signficantly different from 0, i.e. we can test $H_0 : \beta_1^* = 0$ vs. $H_1 : \beta_1^* \neq 0$. From the output of the above model, we obtain a p-value of $0.92964 < \alpha$ (for $\alpha = 5\%$, e.g.), and thus we can conclude that there is NOT a significant difference in the mean intention to buy the candy for individuals with an education level less than high school (educ=1) and those with an education level of high school (educ=2).

Note that we can do the same thing again, changing the reference level, this time setting level 1 as the reference by including `educ2` and `educ3` in the model:

```
mod2.3<-lm(intent~educ2+educ3,data=intention)
summary(mod2.3)
```

```
##
## Call:
## lm(formula = intent ~ educ2 + educ3, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7091 -2.1143 -0.7091  2.2909  5.8857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.76667    0.52305  16.761   <2e-16 ***
## educ2       -0.05758    0.65023  -0.089   0.9296
## educ3       -1.65238    0.71279  -2.318   0.0222 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.865 on 117 degrees of freedom
## Multiple R-squared:  0.06316,    Adjusted R-squared:  0.04714
## F-statistic: 3.944 on 2 and 117 DF,  p-value: 0.022
```

Once again, we obtain the same model, just parametrized differently:

$$\widehat{E}(intention|educ = 1) = \hat{\beta}_0^\dagger = 8.77$$
$$\widehat{E}(intention|educ = 2) = \hat{\beta}_0^\dagger + \hat{\beta}_1^\dagger = 8.71$$
$$\widehat{E}(intention|educ = 3) = \hat{\beta}_0^\dagger + \hat{\beta}_2^\dagger = 7.11$$

Using education level 1 as the reference now means that all comparisons are with respect to this level. What can we conclude from this model?

Note that it would be redundant to include all three levels in the model as we have a perfect relationship:

$$educ1 + educ2 + educ3 = 1$$

In fact, we cannot even fit this model:

```
mod2.4<-lm(intent~educ1+educ2+educ3,data=intention)
summary(mod2.4)
```

```
##
## Call:
## lm(formula = intent ~ educ1 + educ2 + educ3, data = intention)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -6.7091 -2.1143 -0.7091  2.2909  5.8857
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.1143     0.4842  14.691   <2e-16 ***
## educ1         1.6524     0.7128   2.318   0.0222 *
## educ2         1.5948     0.6194   2.575   0.0113 *
## educ3             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.865 on 117 degrees of freedom
## Multiple R-squared:  0.06316,    Adjusted R-squared:  0.04714
## F-statistic: 3.944 on 2 and 117 DF,  p-value: 0.022
```

So far, we've considered *manually* creating the necessary indicator variables for the various levels of the `educ` variable. Note that there is a way for R to do this automatically for us: by declaring `educ` as a categorical variable using the `as.factor()` function:

```
# as.factor()
mod2.5<-lm(intent~as.factor(educ),data=intention)
summary(mod2.5)
```

```
##
## Call:
## lm(formula = intent ~ as.factor(educ), data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7091 -2.1143 -0.7091  2.2909  5.8857
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8.76667    0.52305  16.761   <2e-16 ***
## as.factor(educ)2 -0.05758    0.65023  -0.089   0.9296
## as.factor(educ)3 -1.65238    0.71279  -2.318   0.0222 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.865 on 117 degrees of freedom
## Multiple R-squared:  0.06316,    Adjusted R-squared:  0.04714
## F-statistic: 3.944 on 2 and 117 DF,  p-value: 0.022
```

We obtain the same results, only the output is displayed in a slightly different manner.

We can also consider changing the reference level using the `relevel()` function:

```
intention$educ<-relevel(as.factor(intention$educ),ref=2)
mod2.6<-lm(intent~educ,data=intention)
summary(mod2.6)
```

```
##
## Call:
## lm(formula = intent ~ educ, data = intention)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -6.7091 -2.1143 -0.7091  2.2909  5.8857
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.70909    0.38629  22.545   <2e-16 ***
## educ1        0.05758    0.65023   0.089   0.9296
## educ3       -1.59481    0.61945  -2.575   0.0113 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.865 on 117 degrees of freedom
## Multiple R-squared:  0.06316,    Adjusted R-squared:  0.04714
## F-statistic: 3.944 on 2 and 117 DF,  p-value: 0.022
```

For a categorical variable with several levels, it would become very cumbersome to consider all possible different models by changing the reference level. There exists functions in R to consider all possible pairwise comparisons, without having to reparameterize and refit the models. The code below shows how to do this using the `contrast` function in the `emmeans` library:

```
# toutes comparaisons
# all comparisons
library(emmeans)
```

```
## Welcome to emmeans.
## Caution: You lose important information if you filter this package's results.
## See '? untidy'
```

```
comp<-emmeans(mod2.6,~educ)
comp
```

```
##  educ emmean    SE  df lower.CL upper.CL
##  2      8.71 0.386 117     7.94     9.47
##  1      8.77 0.523 117     7.73     9.80
##  3      7.11 0.484 117     6.16     8.07
##
## Confidence level used: 0.95
```

```
contrast(comp,method="pairwise",adjust="none")
```

```
##  contrast      estimate    SE  df t.ratio p.value
##  educ2 - educ1  -0.0576 0.650 117  -0.089  0.9296
##  educ2 - educ3   1.5948 0.619 117   2.575  0.0113
##  educ1 - educ3   1.6524 0.713 117   2.318  0.0222
```

We obtain the same results we have already explored. We see that, on average,

- the intention to buy is significantly different for educ=1 (less than high school) comapred to educ=3 (university); the estimated mean difference between these two groups is 1.65, suggesting that on average, those with less than a high school education have a higher intention to buy than those with a university degree
- the intention to buy is significantly different for educ=2 (high school) compared to educ=3 (university); the estimated mean difference between these two groups is 1.59, suggesting that on average, those with a high school education have a higher intention to buy than those with a university degree
- the intention to buy is not significantly different between educ=2 (high school) and educ=1 (less than high school).

Can we conclude that higher education levels cause you to have a lower intention to buy candy?

3) Test for global effects

Return to our model which includes all possible variables:

```
mod<-lm(intent~sex+age+as.factor(rev)+as.factor(educ)+stat+fix+emo,data=intention)
summary(mod)
```

```
##
## Call:
## lm(formula = intent ~ sex + age + as.factor(rev) + as.factor(educ) +
##     stat + fix + emo, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4673 -1.7410  0.0477  1.5101  5.2422
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.2288     1.4663   6.976 2.39e-10 ***
## sex                1.1198     0.4416   2.536  0.01262 *
## age               -0.1285     0.0477  -2.693  0.00818 **
## as.factor(rev)2   -1.4605     0.5370  -2.720  0.00759 **
## as.factor(rev)3   -1.7018     0.6153  -2.766  0.00666 **
## as.factor(educ)1  -0.6770     0.5409  -1.252  0.21340
## as.factor(educ)3  -0.7696     0.5133  -1.499  0.13669
## stat              -0.2867     0.4342  -0.660  0.51050
## fix                1.1684     0.1946   6.003 2.54e-08 ***
## emo                1.0972     0.4089   2.683  0.00842 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.264 on 110 degrees of freedom
## Multiple R-squared:  0.4497, Adjusted R-squared:  0.4047
## F-statistic: 9.989 on 9 and 110 DF,  p-value: 4.337e-11
```

Part of the output provided from the `summary` of the model gives an `F-statistic` of 9.989 with df1=9 and df2=119, yielding a p-value of 4.337×10^{-11}. These are precisely the results for the global F-test

$$H_0 : \beta_{sex} = \beta_{age} = \beta_{rev} = \beta_{educ} = \beta_{stat} = \beta_{fix} = \beta_{emo} = 0$$
$$H_1 : \text{at least one of the parameters is different from 0}$$

The p-value is smaller than any reasonable significance level α, and thus we can reject H_0 and conclude that the model contains at least one varaible useful for explaining an inidividual's intention to buy.

Note that we can also obtain the results on a more granular level working with the `anova` function:

```
anova(mod) # (sequential SS)
```

```
## Analysis of Variance Table
##
## Response: intent
##                   Df Sum Sq Mean Sq F value    Pr(>F)
## sex                1  56.05  56.050 10.9312 0.0012775 **
## age                1  76.94  76.937 15.0048 0.0001825 ***
## as.factor(rev)     2  42.81  21.403  4.1742 0.0178923 *
## as.factor(educ)    2  35.88  17.939  3.4987 0.0336453 *
## stat               1   1.59   1.591  0.3103 0.5786522
```

```
## fix               1 210.79 210.790 41.1096 3.704e-09 ***
## emo               1  36.91  36.912  7.1989 0.0084212 **
## Residuals       110 564.03   5.128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The output from `anova(mod)` will yield a breakdown on *sequential sum of squares* for each variable in the model, along with the corresponding df, mean squared error, F-value and p-value. The `Residuals` line corresponds to the error portion. Thus, the breakdown can be obtained as follows:

```
# SSR (here, i.e. SSE(reduced) - SSE(full) where reduced model is an intercept only model )
sum(anova(mod)[1:7,2])
```

```
## [1] 460.9654
```

```
# p
sum(anova(mod)[1:7,1])
```

```
## [1] 9
```

```
# SSE(full)
anova(mod)[8,2]
```

```
## [1] 564.0262
```

```
# n-p-1
anova(mod)[8,1]
```

```
## [1] 110
```

```
# ... #
# p-value
pf(9.988936,9,110,lower.tail=FALSE)
```

```
## [1] 4.337331e-11
```

From this, we could fill out a table of the form

Source	DF	SS	MS	F-value	p-value
Model	9	460.9654	51.21838	9.988936	4.337331×10^{-11}
Error	110	564.0262	5.127511		
Total	119	1024.992			

Another way to proceed is to compare the null (intercept only) model $Y = \beta_0 + \epsilon$ with the fitted model including all covariates:

```
# modele sans X1,...,Xp
# model without X1,...,Xp
null<-lm(intent~1,data=intention)
summary(null)
```

```
##
## Call:
## lm(formula = intent ~ 1, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2583 -2.2583 -0.2583  2.7417  5.7417
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)   8.2583     0.2679   30.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.935 on 119 degrees of freedom
```

```
# SST
anova(null)
```

```
## Analysis of Variance Table
##
## Response: intent
##            Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 119   1025  8.6134
```

```
## global f-test
anova(mod,null)
```

```
## Analysis of Variance Table
##
## Model 1: intent ~ sex + age + as.factor(rev) + as.factor(educ) + stat +
##     fix + emo
## Model 2: intent ~ 1
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    110  564.03
## 2    119 1024.99 -9   -460.97 9.9889 4.337e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We obtain the same results, exactly as before (although shown a bit differently).

We can also obtain global tests for each variable included in the model. Note that if a variable is included as a continuous covariate, that is, the variable has a single coefficient say β_j, then this test is the same as the t-test of the form

$$H_0 : \beta_j = 0 \quad \text{vs.} \quad H_1 : \beta_j \neq 0$$

When a variable is treated as categorical, and is thus represented by a series of coefficients say $\beta_1, \ldots, \beta_k$, then this will test

$$H_0 : \beta_1 = \cdots = \beta_k = 0 \quad \text{vs.} \quad H_1 : \text{ at least one } \beta_j \neq 0, j \in \{1, \ldots, k\}$$

We can do this in R as follows:

```
# Test F:
# effets des variables individuellement
# individual variable effects
library(car)
```

```
## Loading required package: carData
```

```
Anova(mod)
```

```
## Anova Table (Type II tests)
##
## Response: intent
##                 Sum Sq  Df F value    Pr(>F)
## sex              32.98   1  6.4312  0.012617 *
## age              37.20   1  7.2544  0.008180 **
## as.factor(rev)   49.86   2  4.8616  0.009480 **
## as.factor(educ)  14.84   2  1.4469  0.239740
```

```
## stat              2.23   1  0.4359  0.510498
## fix             184.76   1 36.0338 2.539e-08 ***
## emo              36.91   1  7.1989  0.008421 **
## Residuals       564.03 110
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
?Anova
```

```
## starting httpd help server ...
```

```
##  done
```

- For the variables sex, age, stat, fix and emo (all with df=1), the resulting p-value is identical to that obtained in the `summary` of the model (the only difference is that the F-value is the t-value squared). *Note that this follows from the fact that if a random variable T follows a Student t distribution with ν degrees of freedom, then T^2 follows an F-distribution with 1 and ν degrees of freedom.*
- For the varaibles rev and educ, the df=2 as these are both categorical variables with 3 levels, and hence both of these variables have two corresponding regression coefficients (β's).
- For the global test involding education, the hypothesis are $H_0 : \beta_{educ1} = \beta_{educ3} = 0$ vs. H_1 at least one of β_{educ1} or β_{educ3} are $\neq 0$, i.e., we are testing whether education has a significant effect in the model. The p-value is 0.24, which is greater than any reasonable α. Thus we fail to reject H_0 and thus we can conclude that education isn't globally significant (in the model which includes all the other variables).
- What can we conclude for revenue?

Note: another way to proceed for categorical variables, for example, education, is using the ***anova*** *command to compare the model with and without education:*

```
mod.no.edu<-lm(intent~fix+emo+sex+age+as.factor(rev)+stat,data=intention)
anova(mod,mod.no.edu)
```

```
## Analysis of Variance Table
##
## Model 1: intent ~ sex + age + as.factor(rev) + as.factor(educ) + stat +
##     fix + emo
## Model 2: intent ~ fix + emo + sex + age + as.factor(rev) + stat
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    110 564.03
## 2    112 578.86 -2   -14.838 1.4469 0.2397
```

4) Prediction

We'll consider a somewhat artificial example to simply illustrate how to obtain predictions. Here, rather than creating 10 new "individuals" for whom we have covariate values X, we'll simply use the first 10 observations in the dataset. To start, we'll create a new dataset which takes the first 10 lines of the original dataset:

```
datapred<-intention[1:10,]
datapred
```

```
##      fix   emo sex age rev educ stat intent educ1 educ2 educ3
## 1  0.081 1.417   1  27   1    2    0     11     0     1     0
## 2  2.235 1.146   0  27   1    1    0     12     1     0     0
## 3  1.675 0.296   1  26   1    2    1      6     0     1     0
## 4  0.630 0.731   1  34   3    3    0      4     0     0     1
## 5  2.197 0.841   1  30   1    2    1     11     0     1     0
## 6  0.424 0.334   0  29   3    3    1      4     0     0     1
## 7  4.378 0.746   1  27   2    1    0     14     1     0     0
```

```
## 8  2.588 0.808   0  31   2    2    1      8     0     1     0
## 9  2.036 2.797   0  32   3    2    0      9     0     1     0
## 10 0.104 0.053   1  45   3    3    0      3     0     0     1
```

Recall: it's important for the data on which we wish to obtain predictions use the same variable names as that considered in the model used to predict.

We can then use the `predict` function to obtain predictions, exactly as we saw for simple linear regression:

```
pred<-predict.lm(mod,newdata=datapred)
cbind(datapred,pred)
```

```
##      fix   emo sex age rev educ stat intent educ1 educ2 educ3      pred
## 1  0.081 1.417   1  27   1    2    0     11     0     1     0  9.529442
## 2  2.235 1.146   0  27   1    1    0     12     1     0     0  9.952029
## 3  1.675 0.296   1  26   1    2    1      6     0     1     0 10.003629
## 4  0.630 0.731   1  34   3    3    0      4     0     0     1  6.047584
## 5  2.197 0.841   1  30   1    2    1     11     0     1     0 10.697642
## 6  0.424 0.334   0  29   3    3    1      4     0     0     1  4.607141
## 7  4.378 0.746   1  27   2    1    0     14     1     0     0 11.676233
## 8  2.588 0.808   0  31   2    2    1      8     0     1     0  8.409493
## 9  2.036 2.797   0  32   3    2    0      9     0     1     0  9.863863
## 10 0.104 0.053   1  45   3    3    0      3     0     0     1  3.276027
```

(As we had seen in the case of simple linear regression, we can also obtain confidence intervals and prediction intervals for this.)

5) Analysis of residuals

First, we'll add the (jackknife studentized) residuals and the fitted values to the dataset so that we have all the information necessary to carry out the residual analysis:

```
resid<-rstudent(mod)
fitted<-mod$fitted.values
res.dat<-cbind(intention,fitted,resid)
head(res.dat)
```

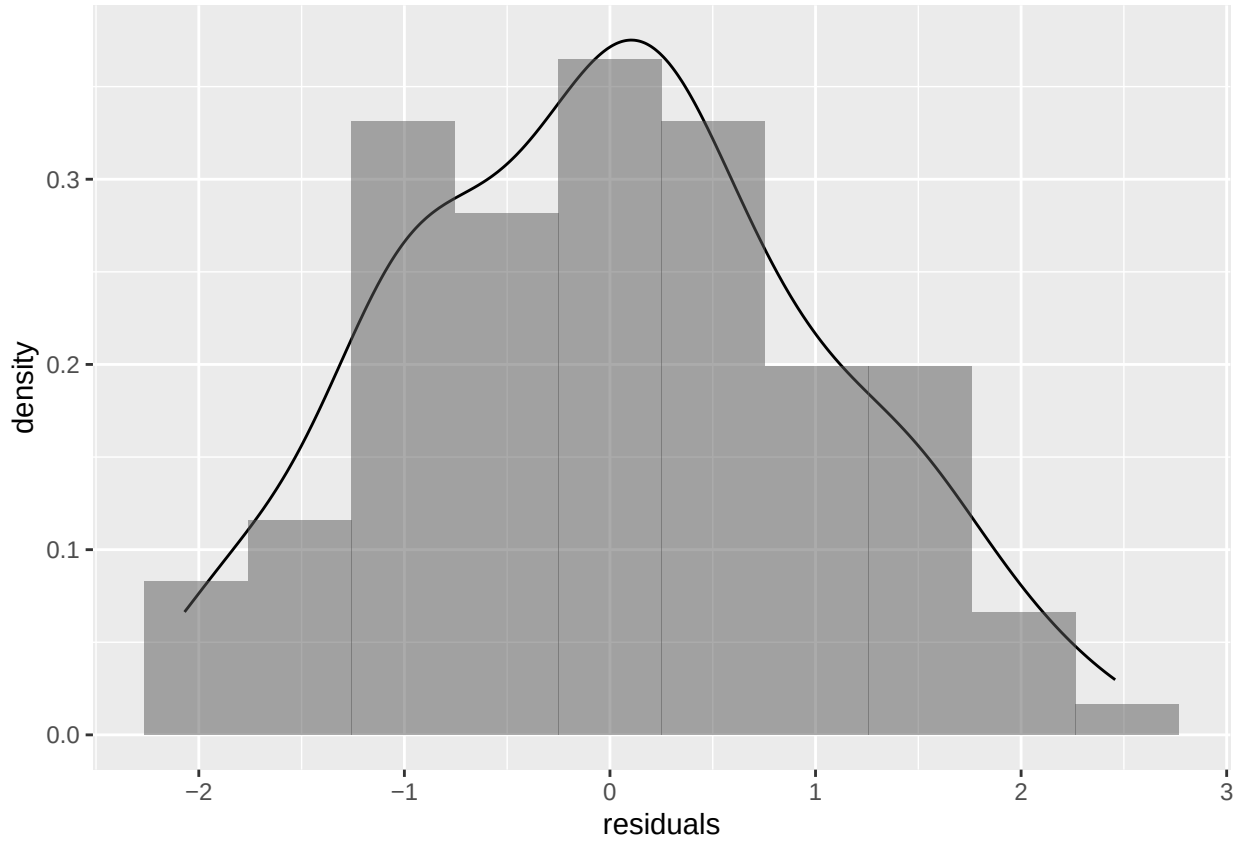
```
##     fix   emo sex age rev educ stat intent educ1 educ2 educ3    fitted
## 1 0.081 1.417   1  27   1    2    0     11     0     1     0  9.529442
## 2 2.235 1.146   0  27   1    1    0     12     1     0     0  9.952029
## 3 1.675 0.296   1  26   1    2    1      6     0     1     0 10.003629
## 4 0.630 0.731   1  34   3    3    0      4     0     0     1  6.047584
## 5 2.197 0.841   1  30   1    2    1     11     0     1     0 10.697642
## 6 0.424 0.334   0  29   3    3    1      4     0     0     1  4.607141
##        resid
## 1  0.6758285
## 2  0.9410718
## 3 -1.8648450
## 4 -0.9412665
## 5  0.1377251
## 6 -0.2772972
```

Histogram and qq-plot:

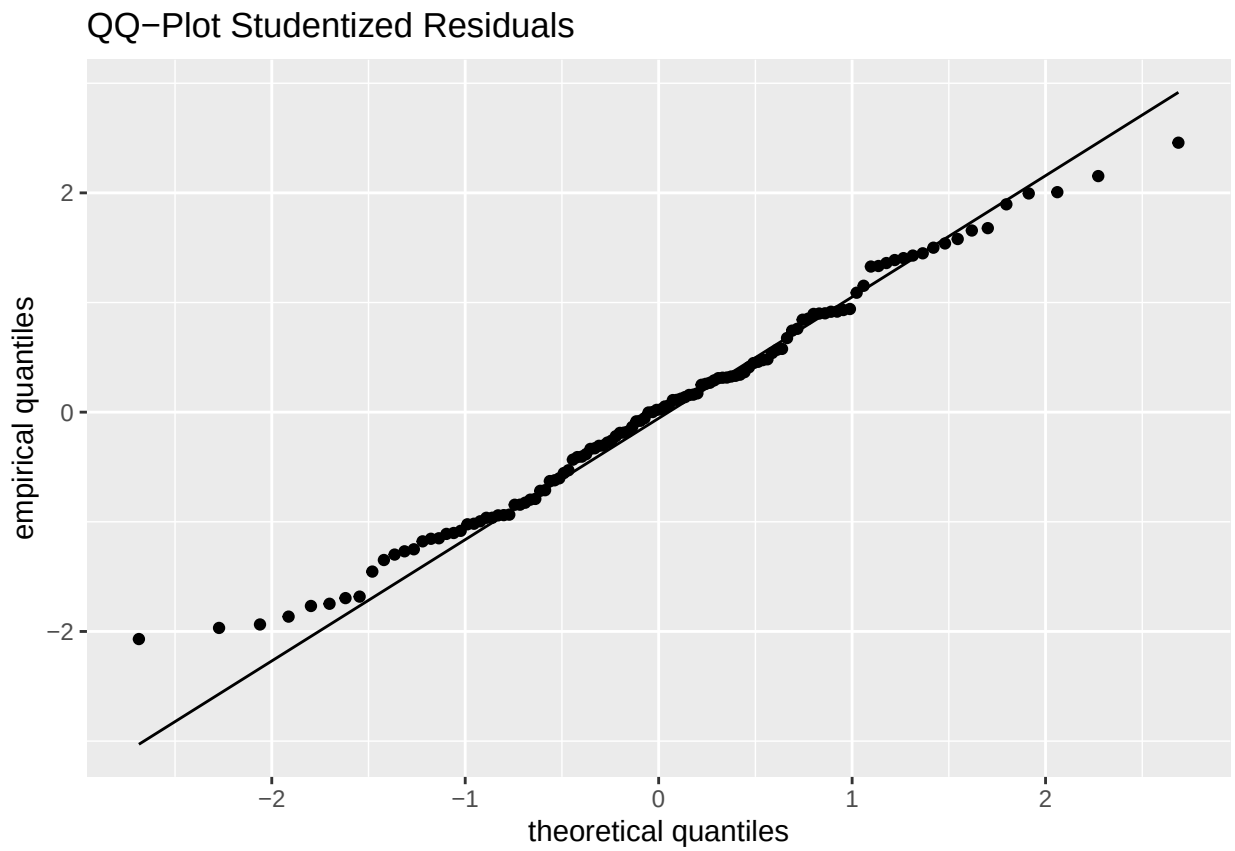
```
#
# histogram
library(ggplot2)
```

```
ggplot(data = res.dat, mapping = aes(x = resid)) +
  geom_density() +
  geom_histogram(aes(y = ..density..), bins = 10, alpha = 0.5) +
  xlab("residuals")
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
#
# qqplot
ggplot(data = res.dat, mapping = aes(sample = resid)) +
  stat_qq(distribution = qt, dparams = mod$df.residua) +
  stat_qq_line(distribution = qt, dparams = mod$df.residual) +
  labs(x = "theoretical quantiles",
       y = "empirical quantiles") +
  ggtitle("QQ-Plot Studentized Residuals")
```

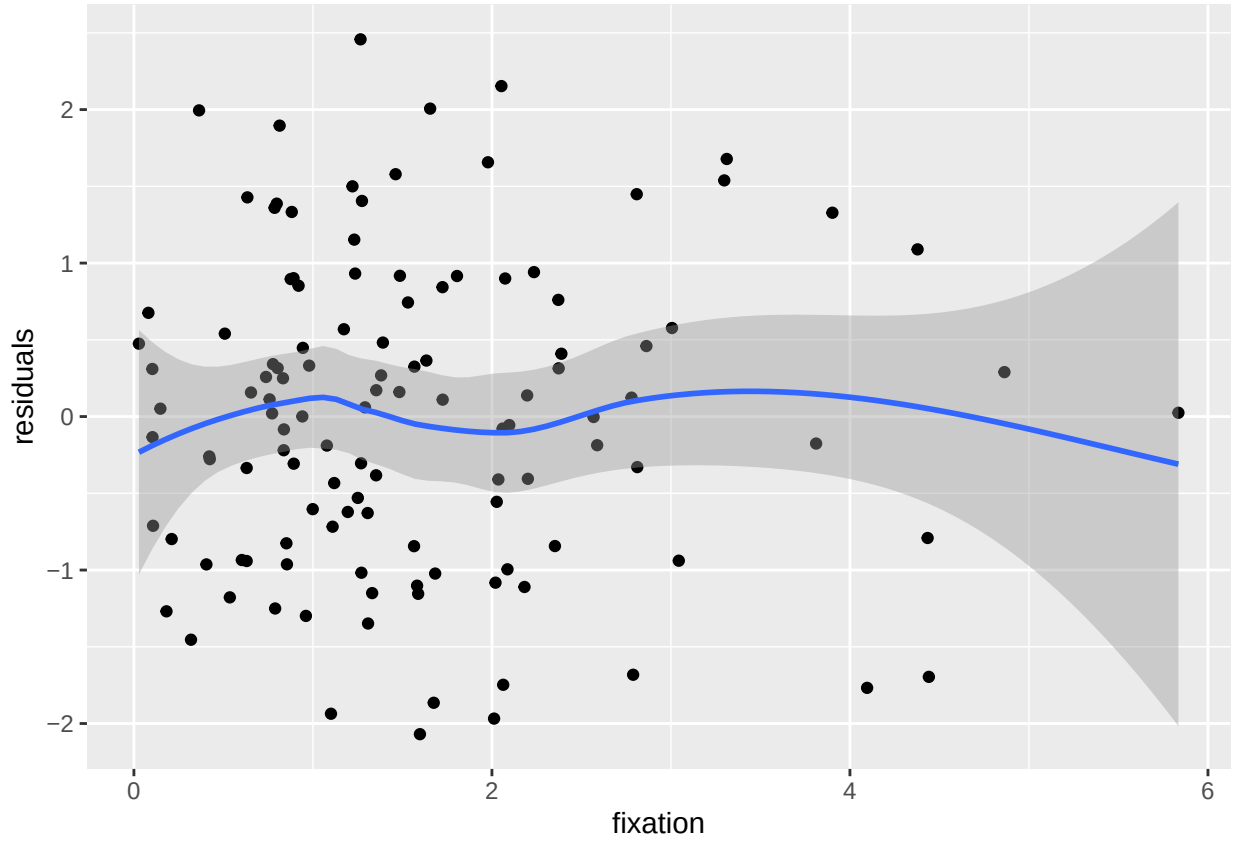


While not "perfect'', these plots do seem to show an approximate normal distribution.

Plots of residuals vs. covariates, and residuals vs. fitted values:

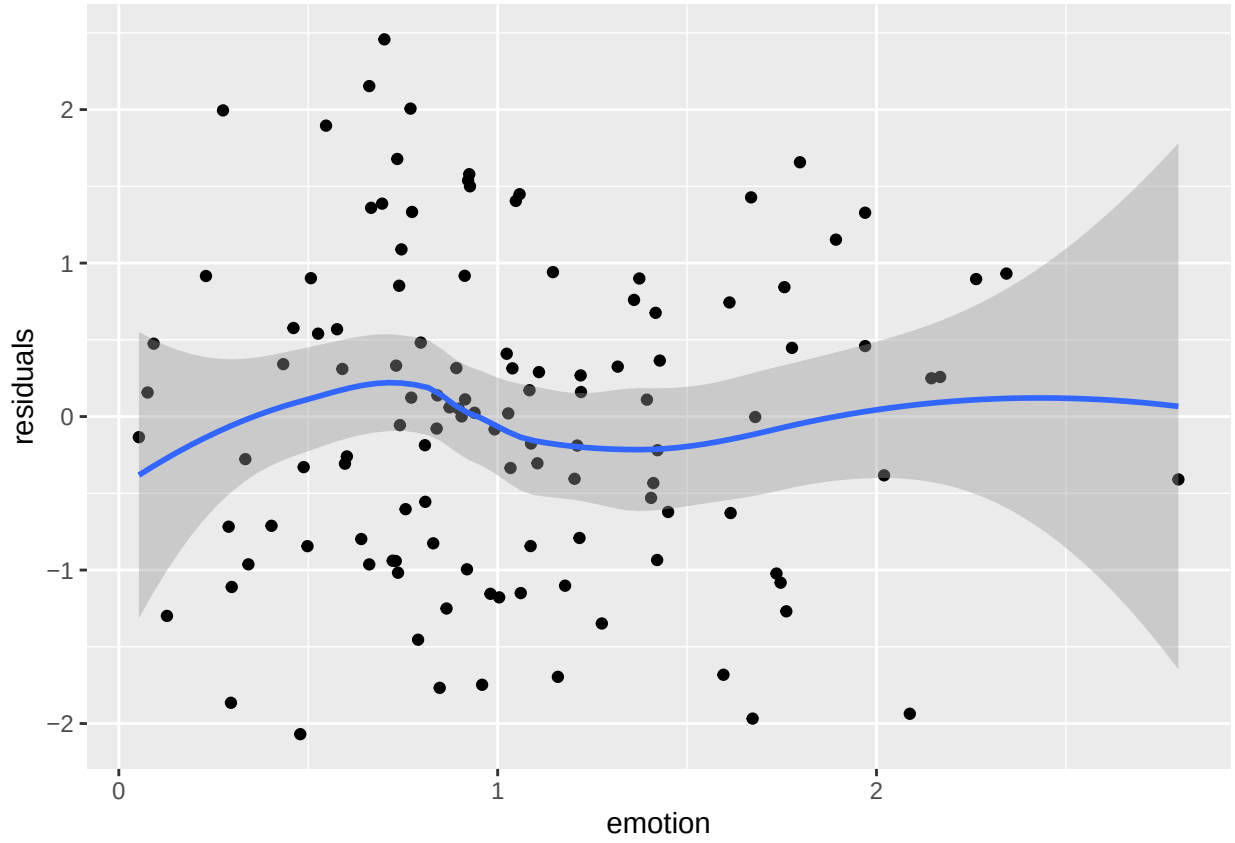
```
#
# resid vs. fix + smooth
ggplot(data = res.dat,
       aes(x = fix, y = resid)) +
  geom_point() +
  geom_smooth() +
  theme(legend.position = "bottom") +
  ylab("residuals") +
  xlab("fixation")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



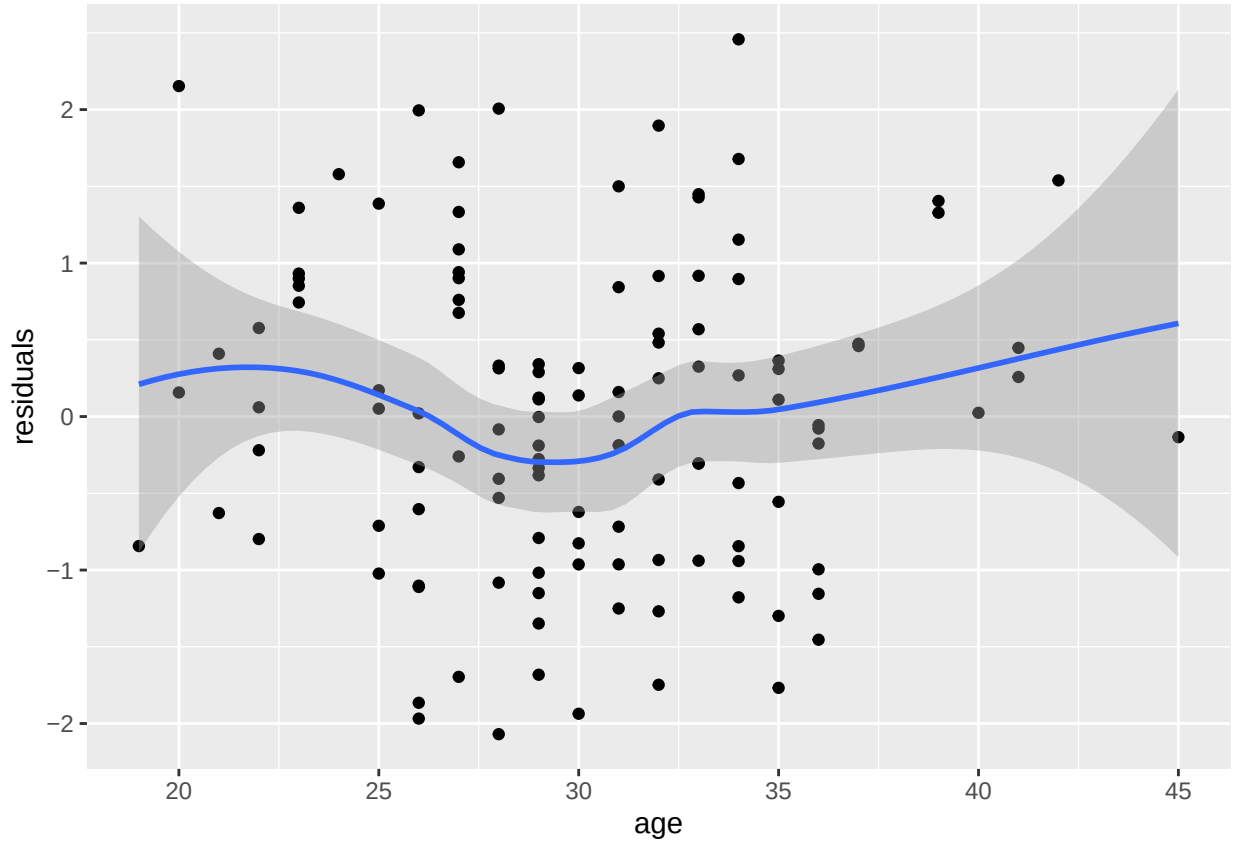

```
#
# resid vs. emo + smooth
ggplot(data = res.dat,
       aes(x = emo, y = resid)) +
  geom_point() +
  geom_smooth() +
  theme(legend.position = "bottom") +
  ylab("residuals") +
  xlab("emotion")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

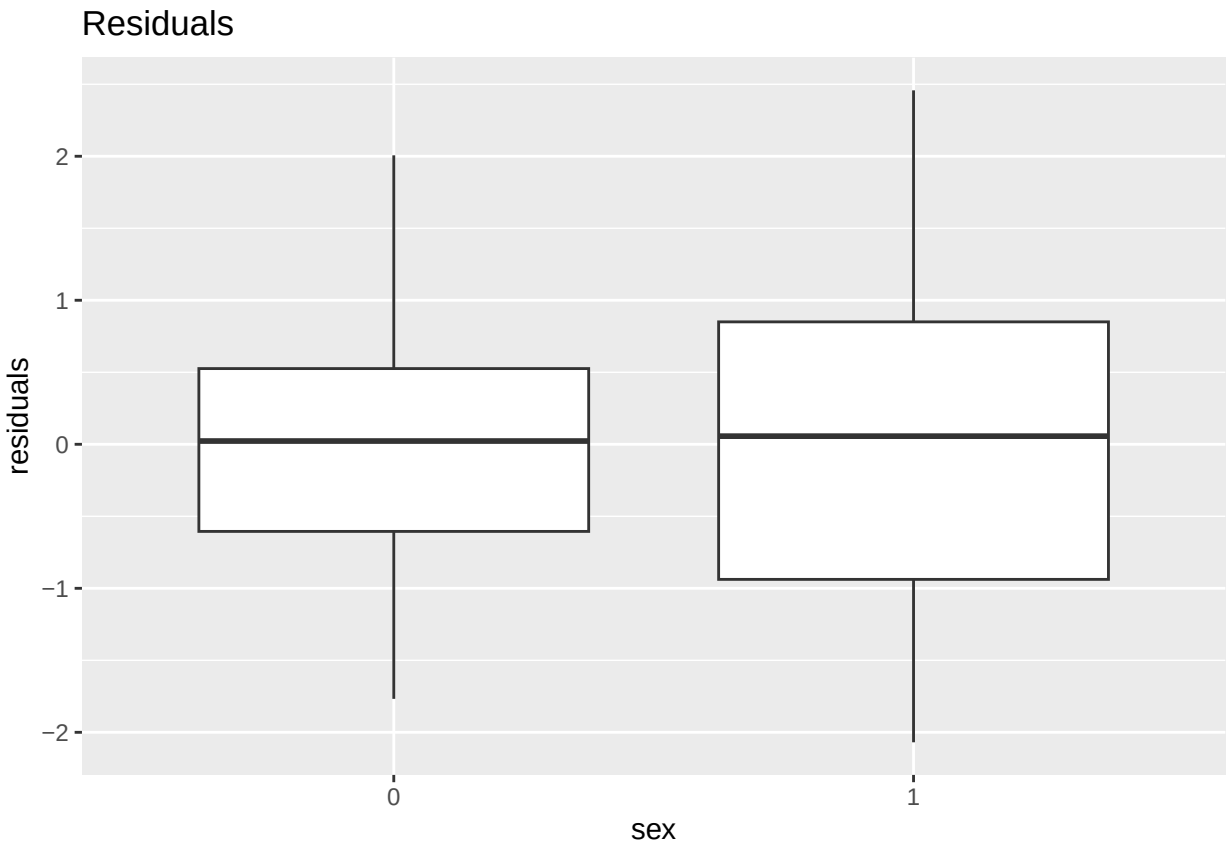



```
#
# resid vs. age + smooth
ggplot(data = res.dat,
       aes(x = age, y = resid)) +
  geom_point() +
  geom_smooth() +
  theme(legend.position = "bottom") +
  ylab("residuals") +
  xlab("age")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



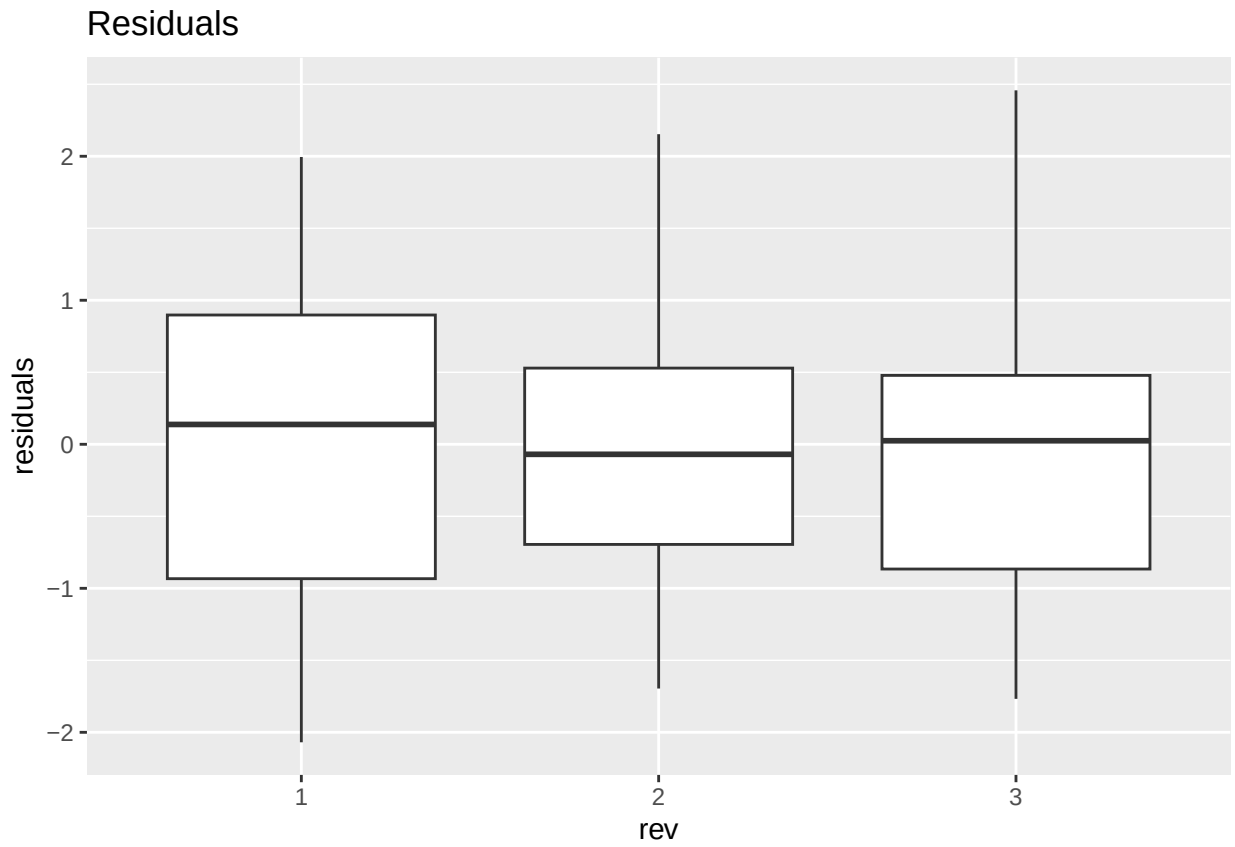

```
#
# resid vs. sex
ggplot(res.dat, aes(x=as.factor(sex), y=resid)) +
  geom_boxplot() +
  labs(title="Residuals",x="sex", y = "residuals")
```




```
tapply(res.dat$resid,as.factor(res.dat$sex),function(x) c(mean(x),var(x)) )
```

```
## $`0`
## [1] 0.00249719 0.79502205
##
## $`1`
## [1] 0.000562229 1.253350149
```

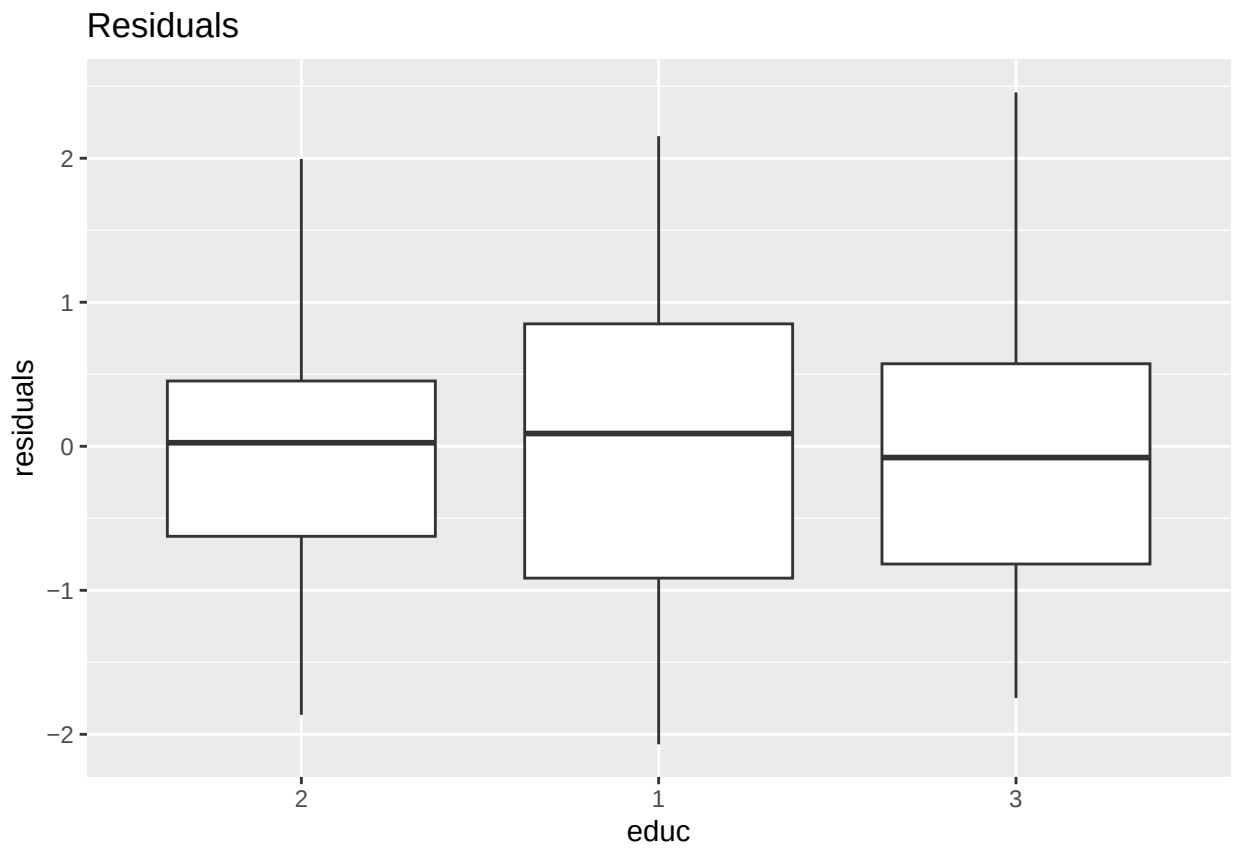
```
#
# resid vs. rev
ggplot(res.dat, aes(x=as.factor(rev), y=resid)) +
  geom_boxplot() +
  labs(title="Residuals",x="rev", y = "residuals")
```




```
tapply(res.dat$resid,as.factor(res.dat$rev),function(x) c(mean(x),var(x)) )
```

```
## $`1`
## [1] -0.003032962  1.345618500
##
## $`2`
## [1] 0.003952771 0.838965413
##
## $`3`
## [1] 0.002786802 0.990977962
```

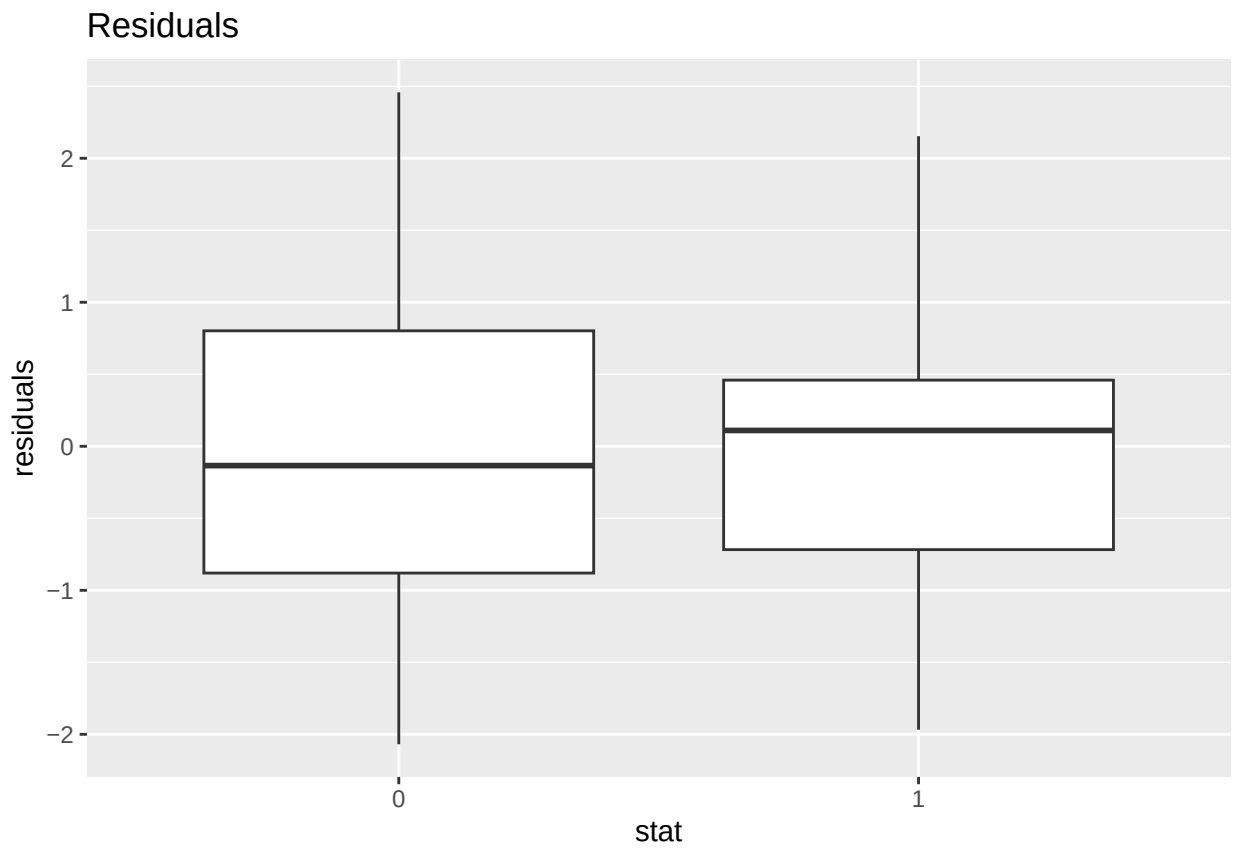
```
#
# resid vs. educ
ggplot(res.dat, aes(x=as.factor(educ), y=resid)) +
  geom_boxplot() +
  labs(title="Residuals",x="educ", y = "residuals")
```




```
tapply(res.dat$resid,as.factor(res.dat$educ),function(x) c(mean(x),var(x)) )
```

```
## $`2`
## [1] 0.001196787 0.894100202
##
## $`1`
## [1] 0.001920443 1.325249341
##
## $`3`
## [1] 0.00160739 1.03109047
```

```
#
# resid vs. stat
ggplot(res.dat, aes(x=as.factor(stat), y=resid)) +
  geom_boxplot() +
  labs(title="Residuals",x="stat", y = "residuals")
```

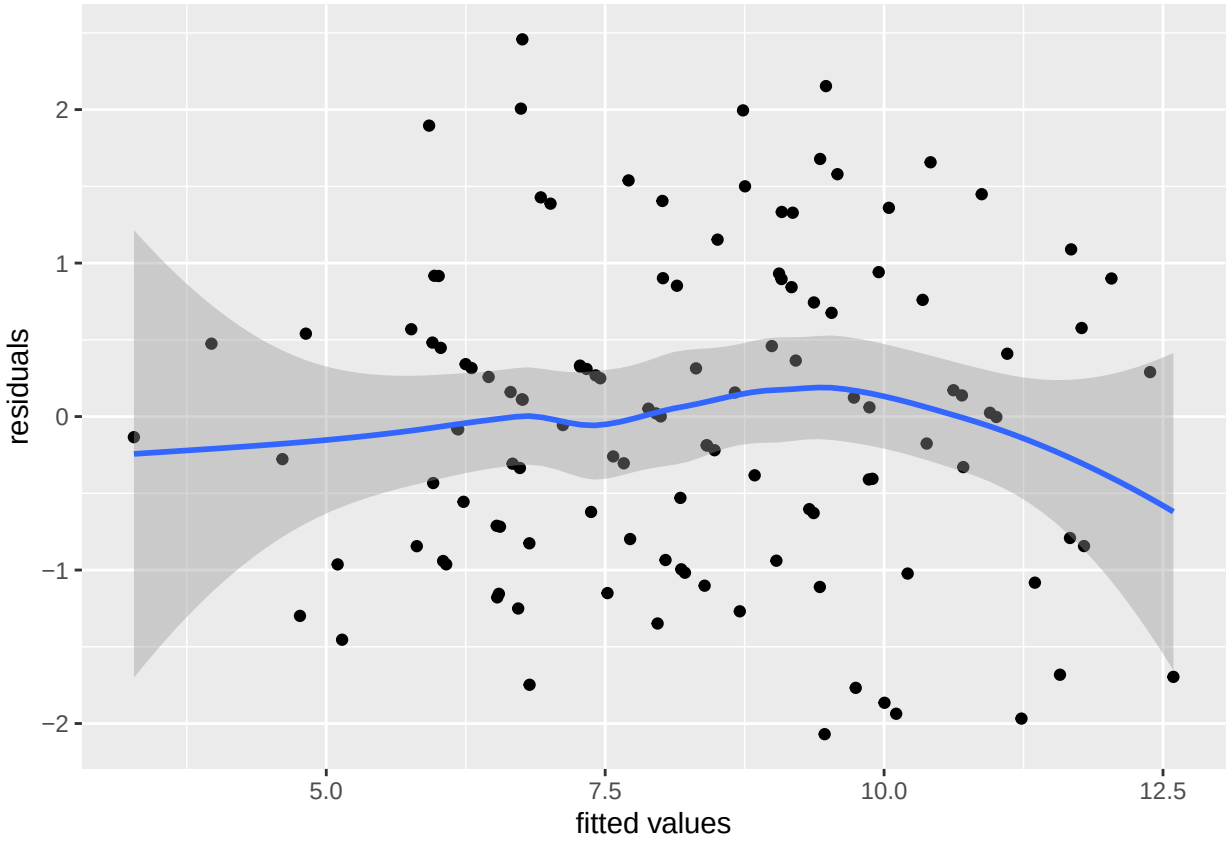



```
tapply(res.dat$resid,as.factor(res.dat$stat),function(x) c(mean(x),var(x)) )
```

```
## $`0`
## [1] 0.00167606 1.03328800
##
## $`1`
## [1] 0.001346337 1.030830829
```

```
#
# resid vs. fitted + smooth
ggplot(data = res.dat,
       aes(x = fitted, y = resid)) +
  geom_point() +
  geom_smooth() +
  theme(legend.position = "bottom") +
  ylab("residuals") +
  xlab("fitted values")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



Overall, there do not appear to be any trends suggesting that the model is misspecified, nor does there appear to be heteroscedasticity. There do not seem to be any anomalies or strange things happening.

For this example, the residual analysis did not give any reason for us to doubt the model assumptions. Therefore, the model seems to be appropriate.

6) R^2

Again, we'll consider the model which includes all covariates, and the intercept-only (null) model:

```
mod<-lm(intent~sex+age+as.factor(rev)+as.factor(educ)+stat+fix+emo,data=intention)
summary(mod)
```

```
##
## Call:
## lm(formula = intent ~ sex + age + as.factor(rev) + as.factor(educ) +
##     stat + fix + emo, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4673 -1.7410  0.0477  1.5101  5.2422
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.2288     1.4663   6.976 2.39e-10 ***
## sex                1.1198     0.4416   2.536  0.01262 *
## age               -0.1285     0.0477  -2.693  0.00818 **
```

```
## as.factor(rev)2   -1.4605     0.5370  -2.720  0.00759 **
## as.factor(rev)3   -1.7018     0.6153  -2.766  0.00666 **
## as.factor(educ)1  -0.6770     0.5409  -1.252  0.21340
## as.factor(educ)3  -0.7696     0.5133  -1.499  0.13669
## stat              -0.2867     0.4342  -0.660  0.51050
## fix                1.1684     0.1946   6.003 2.54e-08 ***
## emo                1.0972     0.4089   2.683  0.00842 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.264 on 110 degrees of freedom
## Multiple R-squared:  0.4497, Adjusted R-squared:  0.4047
## F-statistic: 9.989 on 9 and 110 DF,  p-value: 4.337e-11
```

```
null<-lm(intent~1,data=intention)
summary(null)
```

```
##
## Call:
## lm(formula = intent ~ 1, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.2583 -2.2583 -0.2583  2.7417  5.7417
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.2583     0.2679   30.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.935 on 119 degrees of freedom
```

Similarly to what we explored in the context of global F-tests, we can consider the breakdown $SS_T = SS_R + SS_E$:

```
# by hand / manuellement
# break-down:
anova(mod) # (sequential SS)
```

```
## Analysis of Variance Table
##
## Response: intent
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## sex               1  56.05  56.050 10.9312 0.0012775 **
## age               1  76.94  76.937 15.0048 0.0001825 ***
## as.factor(rev)    2  42.81  21.403  4.1742 0.0178923 *
## as.factor(educ)   2  35.88  17.939  3.4987 0.0336453 *
## stat              1   1.59   1.591  0.3103 0.5786522
## fix               1 210.79 210.790 41.1096 3.704e-09 ***
## emo               1  36.91  36.912  7.1989 0.0084212 **
## Residuals       110 564.03   5.128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# SSR
SSR<-sum(anova(mod)[1:7,2])
SSR
```

```
## [1] 460.9654
```

```r
# SSE
SSE<-anova(mod)[8,2]
SSE
```

```
## [1] 564.0262
```

```r
# SST
anova(null)
```

```
## Analysis of Variance Table
##
## Response: intent
##            Df Sum Sq Mean Sq F value Pr(>F)
## Residuals 119   1025  8.6134
```

```r
SST<-anova(null)[,2]
SST
```

```
## [1] 1024.992
```

```r
# R2
SSR/SST
```

```
## [1] 0.449726
```

We obtain $R^2 = 0.449726$.

We can also manually compute $\hat{\sigma}$:

```r
# sigma-hat
np1<-anova(mod)[8,1]
sqrt(SSE/np1)
```

```
## [1] 2.264401
```

We obtain $\hat{\sigma} = 2.264401$.

Note that all this information is also given in the output from the `summary` of the model:

```r
summary(mod)
```

```
##
## Call:
## lm(formula = intent ~ sex + age + as.factor(rev) + as.factor(educ) +
##     stat + fix + emo, data = intention)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4673 -1.7410  0.0477  1.5101  5.2422
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       10.2288     1.4663   6.976 2.39e-10 ***
## sex                1.1198     0.4416   2.536  0.01262 *
## age               -0.1285     0.0477  -2.693  0.00818 **
```

```
## as.factor(rev)2   -1.4605     0.5370  -2.720  0.00759 **
## as.factor(rev)3   -1.7018     0.6153  -2.766  0.00666 **
## as.factor(educ)1  -0.6770     0.5409  -1.252  0.21340
## as.factor(educ)3  -0.7696     0.5133  -1.499  0.13669
## stat              -0.2867     0.4342  -0.660  0.51050
## fix                1.1684     0.1946   6.003 2.54e-08 ***
## emo                1.0972     0.4089   2.683  0.00842 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.264 on 110 degrees of freedom
## Multiple R-squared:  0.4497, Adjusted R-squared:  0.4047
## F-statistic: 9.989 on 9 and 110 DF,  p-value: 4.337e-11
```

7) Non-linear effects

To explore the inclusion of non-linear covariate effects, we'll consider the data saved in `reglin2`:

```
reglin2<-read.csv("Data/reglin2.csv")
head(reglin2)
```

```
##           x          y
## 1 4.6062230 -4.3068887
## 2 0.9740215 11.4859722
## 3 7.9009642 16.1603293
## 4 1.7884170  2.9177222
## 5 2.6203360  1.2378640
## 6 3.7459341  0.8398478
```

Here there are two variables: the response variable `y` and the explanatory variable `x`.

We can include a quadratic term for X directly within the model specification in the `lm` function:

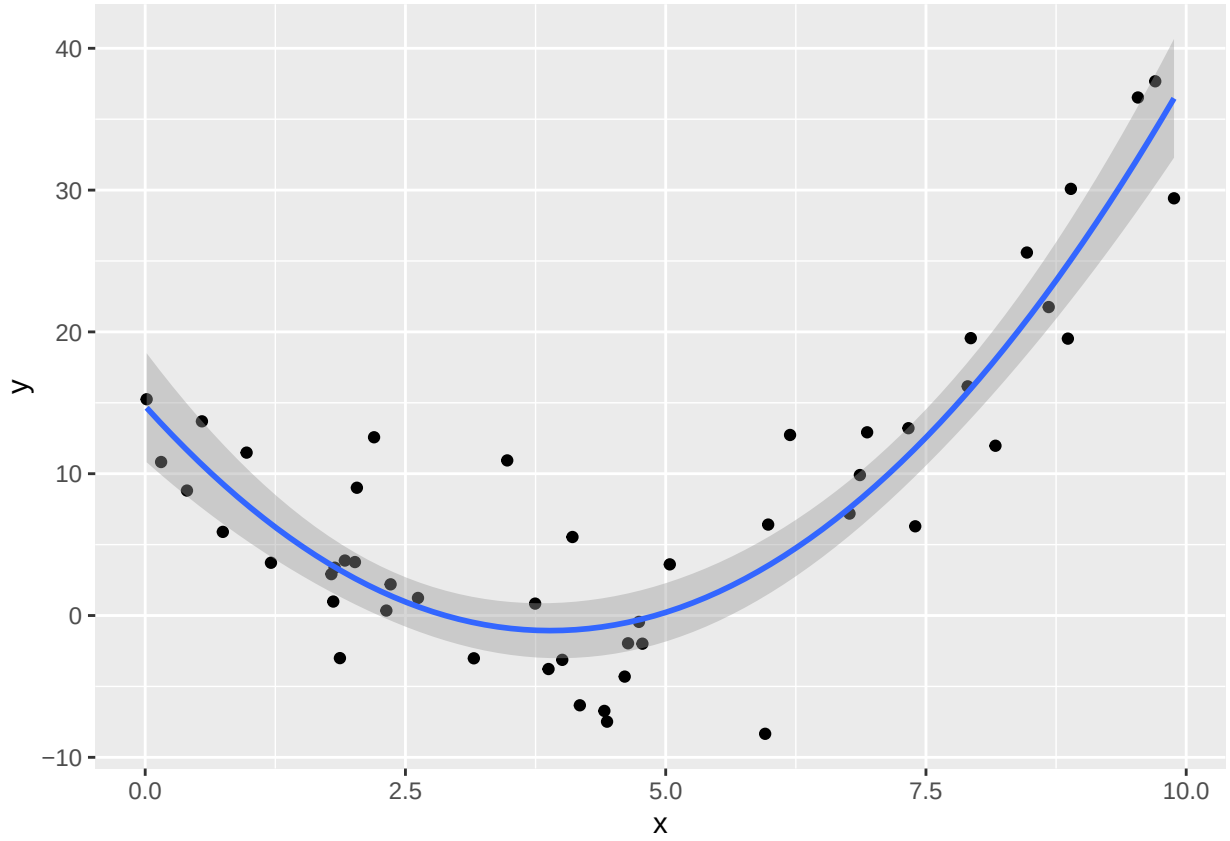
```
# quadratic
nlmod1<-lm(y~x+I(x^2),data=reglin2)
summary(nlmod1)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2), data = reglin2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7319  -2.7356  -0.0933   3.2332  11.8241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.76504    1.92196   7.682 7.62e-10 ***
## x           -8.13665    0.92734  -8.774 1.83e-11 ***
## I(x^2)       1.04557    0.09139  11.440 3.50e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.797 on 47 degrees of freedom
## Multiple R-squared:  0.8177, Adjusted R-squared:  0.8099
## F-statistic: 105.4 on 2 and 47 DF,  p-value: < 2.2e-16
```

To visualize this relationship:

```
# plot
ggplot(reglin2, aes(x=x, y=y)) +
  geom_point() +
  stat_smooth(aes(y = y),method = "lm",
              formula = y ~ x + I(x^2), size = 1)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Note that as an alternative, we could have created a new variable `x2` consisting of the squared value of `x`, and then include this new variable in the model. Obviously, this is a completely equivalent way to proceed.

```
# alternative
reglin2$x2<-reglin2$x^2
head(reglin2)
```

```
##           x          y         x2
## 1 4.6062230 -4.3068887 21.2172906
## 2 0.9740215 11.4859722  0.9487178
## 3 7.9009642 16.1603293 62.4252353
## 4 1.7884170  2.9177222  3.1984354
## 5 2.6203360  1.2378640  6.8661608
## 6 3.7459341  0.8398478 14.0320221
```

```
nlmod2<-lm(y~x+x2,data=reglin2)
summary(nlmod2)
```

```
##
## Call:
## lm(formula = y ~ x + x2, data = reglin2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.7319  -2.7356  -0.0933   3.2332  11.8241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.76504    1.92196   7.682 7.62e-10 ***
## x           -8.13665    0.92734  -8.774 1.83e-11 ***
## x2           1.04557    0.09139  11.440 3.50e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.797 on 47 degrees of freedom
## Multiple R-squared:  0.8177, Adjusted R-squared:  0.8099
## F-statistic: 105.4 on 2 and 47 DF,  p-value: < 2.2e-16
```

We could include higher order terms as well, e.g. X^3:

```
# cubic
nlmod3<-lm(y~x+I(x^2)+I(x^3),data=reglin2)
summary(nlmod3)
```

```
##
## Call:
## lm(formula = y ~ x + I(x^2) + I(x^3), data = reglin2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5446  -2.7575  -0.1315   3.2069  11.6618
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.29452    2.51222   5.690 8.42e-07 ***
## x           -7.54872    2.20209  -3.428  0.00129 **
## I(x^2)       0.89235    0.52755   1.691  0.09751 .
## I(x^3)       0.01049    0.03557   0.295  0.76933
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.845 on 46 degrees of freedom
## Multiple R-squared:  0.818,  Adjusted R-squared:  0.8061
## F-statistic: 68.92 on 3 and 46 DF,  p-value: < 2.2e-16
```

8) Interactions

Recall the fixation-intention to buy example, but here we'll use another version of this dataset that will make the example more interesting (since there isn't acually a significant interaction effect in the dataset we've

been using up till now). The data only include the variables `intent`, `fix` and `sex`, and are found in the file `reglin3`.

```
reglin3<-read.csv("Data/reglin3.csv")
head(reglin3)
```

```
##     fix sex intent
## 1 0.081   1      4
## 2 2.235   0      4
## 3 1.675   1      7
## 4 0.630   1      4
## 5 2.197   1     11
## 6 0.424   0      3
```

First, let's consider the model with `fix` only, that is, ignoring the effect of sex. We can fit this model:

```
# modele sans effet de sexe
# model without sex effect
mod.int<-lm(intent~fix,data=reglin3)
summary(mod.int)
```

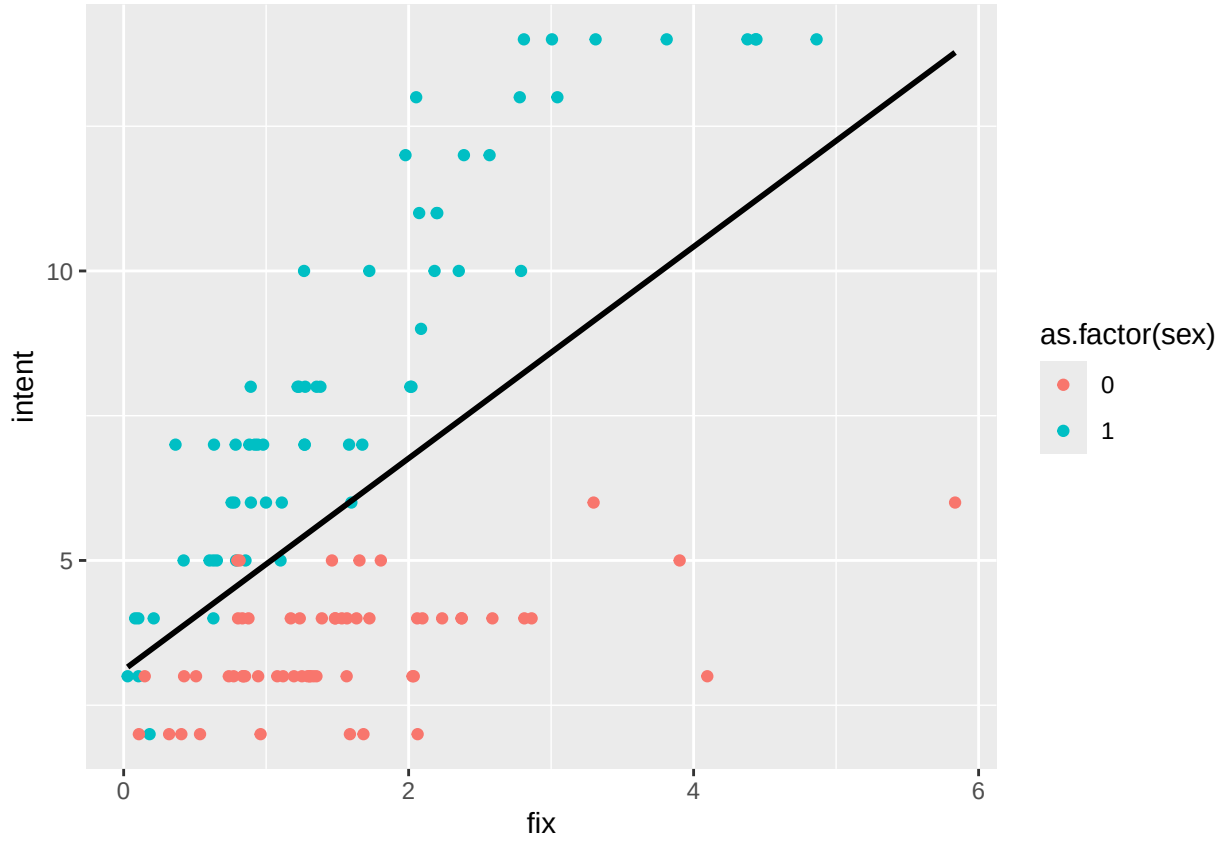
```
##
## Call:
## lm(formula = intent ~ fix, data = reglin3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.7731 -2.0804 -0.1395  2.2281  6.1398
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.1077     0.4602   6.752 5.78e-10 ***
## fix           1.8278     0.2401   7.614 7.24e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.864 on 118 degrees of freedom
## Multiple R-squared:  0.3294, Adjusted R-squared:  0.3237
## F-statistic: 57.97 on 1 and 118 DF,  p-value: 7.24e-12
```

The fitted model is

$$\widehat{intent} = 3.11 + 1.83 fix$$

We can also visualize this model:

```
# visualisation
ggplot(data = reglin3,
       aes(x = fix, y = intent)) +
  geom_point(aes(col = as.factor(sex))) +
  geom_smooth(method = "lm",
              se = FALSE,
              formula = "y ~ x",
                col = "black",
              fullrange = TRUE)
```



Color-coding the observations according to sex suggests that there seems to be a sex effect. We can then consider a model which includes both `fix` and `sex` next.

We'll consider the main effects only model (i.e., the model without interaction):

$$intent = \beta_0 + \beta_1 sex + \beta_2 fix + \epsilon$$

We can fit this model:

```
# modele sans interactions
# model without interaction
mod.int1<-lm(intent~as.factor(sex)+fix,data=reglin3)
summary(mod.int1)
```

```
##
## Call:
## lm(formula = intent ~ as.factor(sex) + fix, data = reglin3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0662 -1.0847  0.1229  0.9109  3.9185
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        0.8078     0.3030   2.666  0.00876 **
## as.factor(sex)1    4.6644     0.3005  15.520  < 2e-16 ***
## fix                1.7581     0.1379  12.746  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 1.644 on 117 degrees of freedom
## Multiple R-squared:  0.7808, Adjusted R-squared:  0.777
## F-statistic: 208.3 on 2 and 117 DF,  p-value: < 2.2e-16
```

The fitted model has the form

$$\widehat{intent} = 0.81 + 4.66sex + 1.76fix$$

We can split up the model based on the level of `sex`:

$$\begin{aligned}
\widehat{E}(intention|man, fixation) &= \widehat{E}(intention|sex = 0, fixation) \\
&= 0.81 + 1.76fixation \\
\widehat{E}(intention|woman, fixation) &= \widehat{E}(intention|sex = 1, fixation) \\
&= (0.81 + 4.66) + 1.76fixation \\
&= 5.47 + 1.76fixation
\end{aligned}$$

Thus, in this model, we assume that the effect of the fixation is the same for the two levels of sex.

- This means that the effect of fixation on intention is the same for both men and women. The model implies that for every additional 1 second of fixation, the intention to buy will increase on average by 1.76, for both men and women.

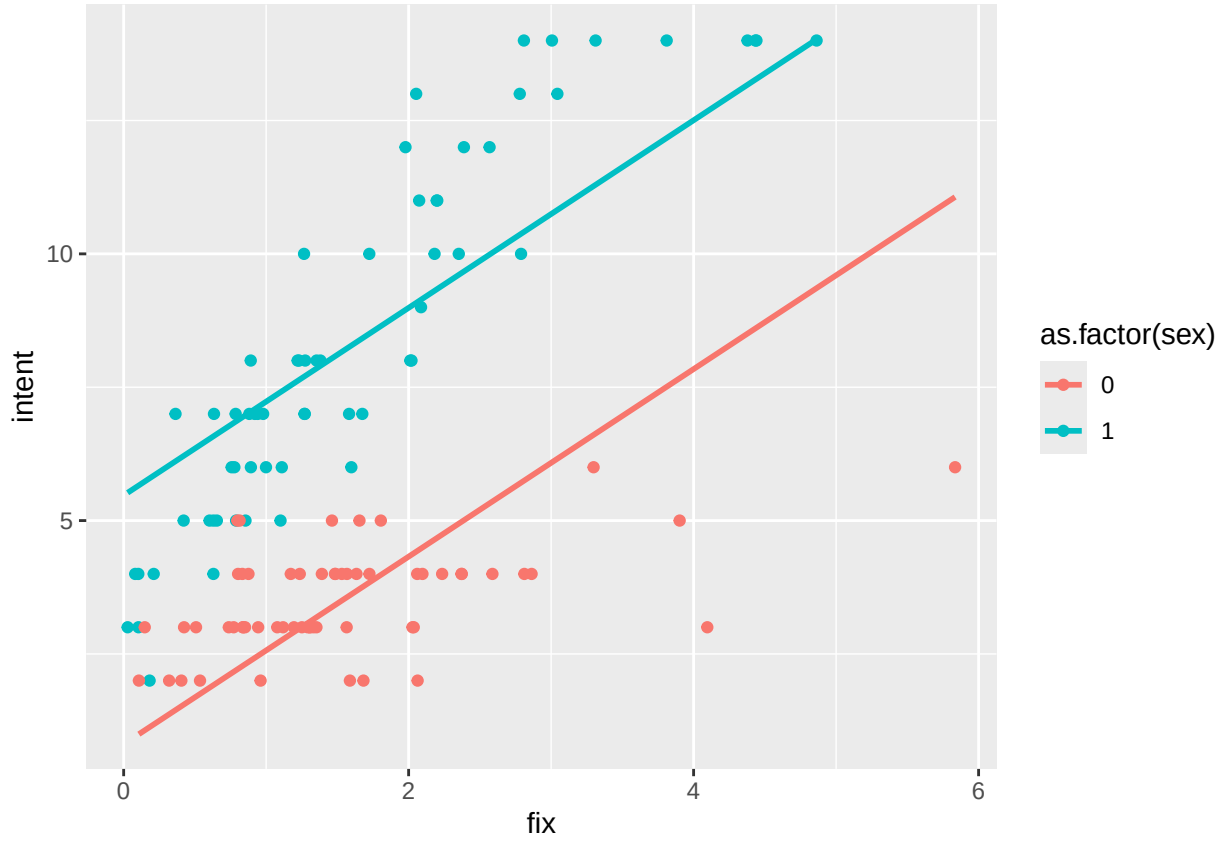
Likewise, the effect of sex is assumed to be the same for all possible values of the continuous variable.

- Here, this means that the effect of sex on intention is the same for all possible values of fixation. The model implies that for any value of fixation, the average difference in the intention to buy between women and men is 4.66.
- We will see this in the next graph, where the difference between the two lines, which represents the effect of sex, is the same for all values of fixation; i.e., the lines are parallel

This follows since the model does not include an interaction between fixation and sex.

We can visualize this model:

```
# fitted values
pred.int1<-mod.int1$fitted.values
# visualisation
ggplot(reglin3, aes(x=fix, y=intent,col=as.factor(sex)))+
  geom_point() +
  geom_line(aes(y = pred.int1), size = 1)
```



We see that fitting this main effects model, with a binary and continuous variable, boils down to simultaneously fitting two lines, one for each level of the binary variable. However, here, the two lines have the **same slope** (1.76), but the **intercepts are different**, to account for the effect of the binary variable. That is, we see parallel lines.

From the model output, we see that both variables are significant (tiny p-values). The R^2 is also quite large (78%). Nonetheless, the model does not seem to be entirely adequate: we see that the model seems to under-estimate the slope for women, and over-estimate the slope for men. That is, it appears that the effect of fixation is not the same for men and women. This would imply there's possibly an interaction between the two variables. We thus need a model where the two lines can have different slopes. This is exactly what an interaction model does!

To fit the interaction model, we simply include an additional covariate `sex*fix` in the model. The model becomes:

$$intent = \beta_0 + \beta_1 sex + \beta_2 fix + \beta_3 sex * fix + \epsilon$$

```
# modele avec interactions
# model with interaction
mod.int2<-lm(intent~as.factor(sex)*fix,data=reglin3)
summary(mod.int2)
```

```
##
## Call:
## lm(formula = intent ~ as.factor(sex) * fix, data = reglin3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8837 -0.7268 -0.1104  0.6443  3.5303
```

```
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.7412     0.2817   9.730  < 2e-16 ***
## as.factor(sex)1       1.3117     0.3799   3.453 0.000774 ***
## fix                   0.5035     0.1531   3.289 0.001333 **
## as.factor(sex)1:fix   2.1349     0.1997  10.688  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.172 on 116 degrees of freedom
## Multiple R-squared:  0.8895, Adjusted R-squared:  0.8867
## F-statistic: 311.4 on 3 and 116 DF,  p-value: < 2.2e-16
```

The fitted model has the form

$$\widehat{intent} = 2.74 + 1.31sex + 0.51fix + 2.13sex * fix$$

We can split up the model based on the level of sex:

$$\begin{aligned}
\widehat{E}(intention|man, fixation) &= \widehat{E}(intention|sex = 0, fixation) \\
&= 2.74 + 0.50fixation \\
\widehat{E}(intention|woman, fixation) &= \widehat{E}(intention|sex = 1, fixation) \\
&= (2.74 + 1.31) + (0.50 + 2.13)fixation \\
&= 4.05 + 2.64fixation
\end{aligned}$$

We see that the fitted model allows for both the intercepts and the slope to differ for the two levels of `sex`. According to the fitted model, the effect of fixation is weaker for males than for females.

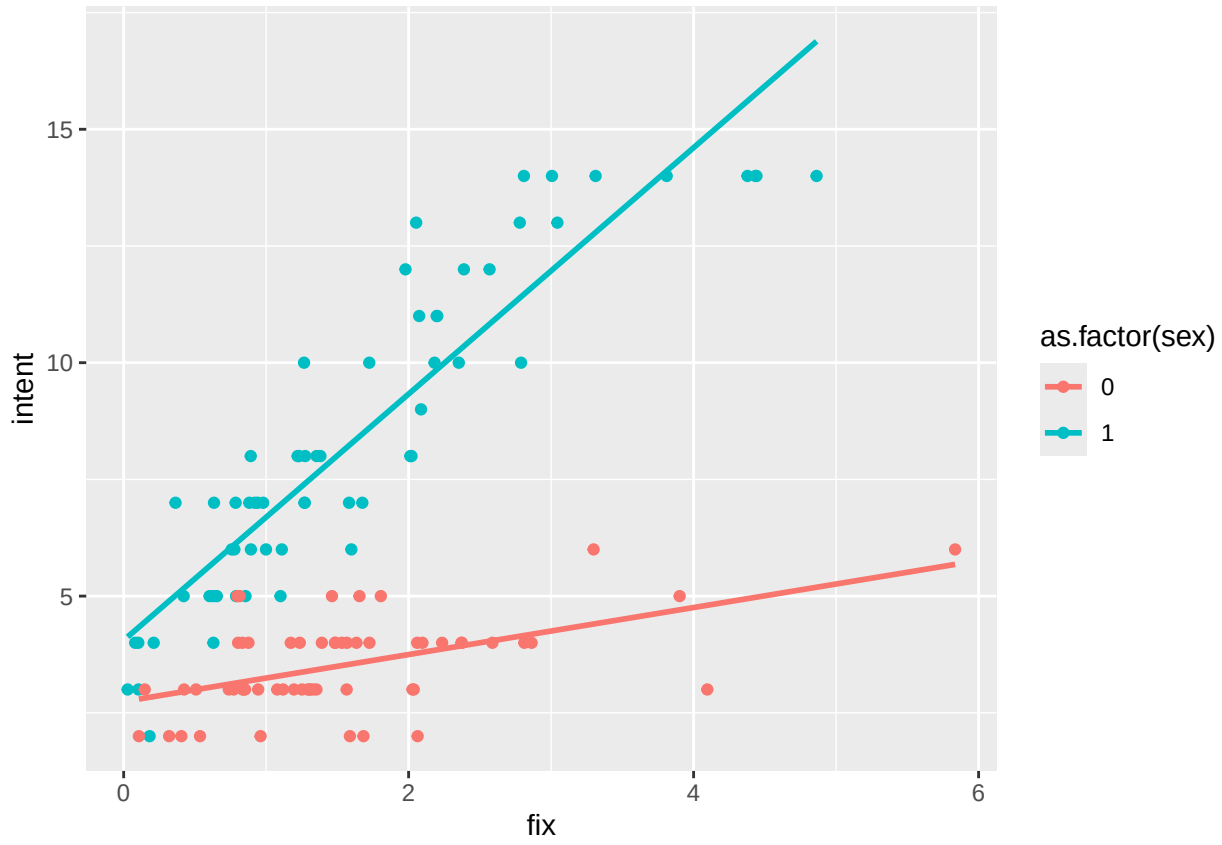
- For men, for each 1 second increase in fixation, the intention increases on average by 0.5.
- For women, for each 1 second increase in fixation, the intention increases on average by 2.64.

This is what we had seen in the previous graph. Notice that the difference between these two effects $2.64 - 0.50 = 2.14$, is exactly $\hat{\beta}_3$ (with some rounding).

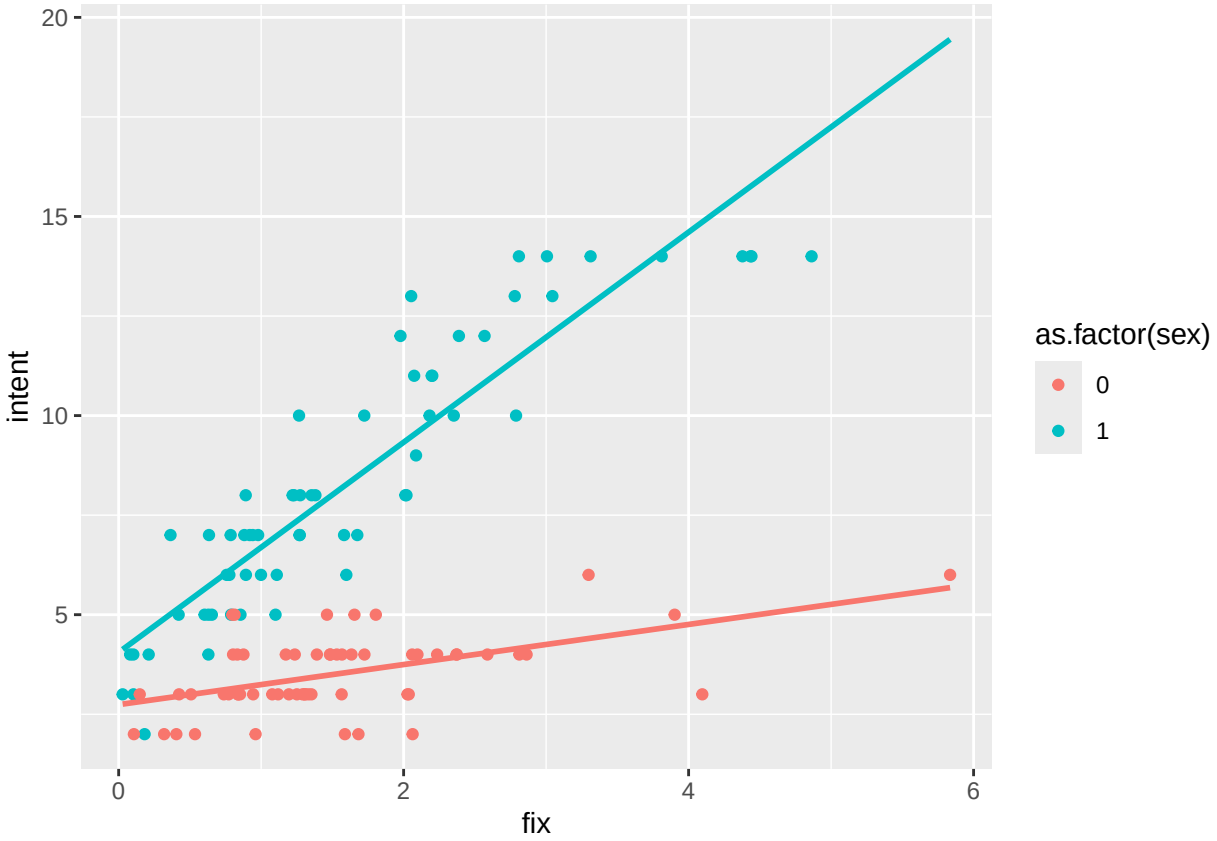
Here we see that the p-value for the interaction term is $< 2 \times 10^{-16}$ and thus we can conclude that theres a significant interaction, that is, the effect of fixation on intention depends on sex, and vice versa.

Recall that an interaction goes in both directions... thus, the effect of sex depends on fixation. In this case, there are an infinite number of possible values for fixation, and so to understand this effects it's best to consider a graphical summary. Visually, the effect of sex is the difference between the two lines on the plots. To visualize this:

```
# fitted value
pred.int2<-mod.int2$fitted.values
# visualisation
ggplot(reglin3, aes(x=fix, y=intent,col=as.factor(sex)))+
  geom_point() +
  geom_line(aes(y = pred.int2), size = 1)
```




```
# alternative
ggplot(data = reglin3,
       aes(x = fix, y = intent, col = as.factor(sex))) +
  geom_point() +
  geom_smooth(method = "lm",
              se = FALSE,
              formula = "y ~ x",
              show.legend = FALSE,
              fullrange = TRUE)
```



Since the two lines are not parallel, the effect of sex changes as a function of fixation. We see that the higher the value of fixation, the larger the difference in the mean intention to buy between males and females. In fact, women generally have a higher intention to buy score. Moreover, this difference becomes more and more pronounced as fixation increases.

Note that we can also include other variables in the model with the interaction. Here we'll do this using the same example, but with a different dataset: `reglin6`.

```
# modele avec interactions + autres variables
# model with interactions + other variables
reglin6<-read.csv("Data/reglin6.csv")
head(reglin6)
```

```
##     fix   emo sex age rev educ stat intent
## 1 0.081 1.417   1  27   1    2    0     11
## 2 2.235 1.146   0  27   1    1    0     10
## 3 1.675 0.296   1  26   1    2    1      9
## 4 0.630 0.731   1  34   3    3    0      5
## 5 2.197 0.841   1  30   1    2    1     14
## 6 0.424 0.334   0  29   3    3    1      4
```

```
mod.int3<-lm(intent~as.factor(sex)*fix+emo+age+as.factor(rev)+as.factor(educ)+stat,data=reglin6)
summary(mod.int3)
```

```
##
## Call:
## lm(formula = intent ~ as.factor(sex) * fix + emo + age + as.factor(rev) +
##     as.factor(educ) + stat, data = reglin6)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.4533 -1.3767  0.0739  1.1780  5.8819
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          9.10484    1.33177   6.837 4.88e-10 ***
## as.factor(sex)1      2.16942    0.67294   3.224 0.001669 **
## fix                  0.59334    0.27582   2.151 0.033670 *
## emo                  1.22726    0.36432   3.369 0.001045 **
## age                 -0.13452    0.04365  -3.081 0.002609 **
## as.factor(rev)2     -1.56564    0.48048  -3.259 0.001494 **
## as.factor(rev)3     -1.86589    0.54688  -3.412 0.000906 ***
## as.factor(educ)2     0.87969    0.48110   1.828 0.070211 .
## as.factor(educ)3     0.18975    0.54448   0.348 0.728143
## stat                 0.10555    0.38619   0.273 0.785135
## as.factor(sex)1:fix  1.17063    0.36078   3.245 0.001562 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.013 on 109 degrees of freedom
## Multiple R-squared:  0.7081, Adjusted R-squared:  0.6813
## F-statistic: 26.44 on 10 and 109 DF,  p-value: < 2.2e-16
```

The interaction between sex and fixation is statistically significant, as are the variables age, revenue and emotion. The effects of the variables that are directly included (without interactions) in the model can be interpreted as usual. For example, for every one year increase in age, the intention to buy decreases on average by 0.13, holding all other variables constant. To correctly interpret the effects of sex and fixation, we need to consider the effect of each as a function of the other, but with the additional remark *holding all other variables constant*. This is easy here since `sex` only takes on two possible values:

- when sex=0, the effect of fixation is estimated as 0.59; thus, for a male, when fixation increases by one second, and all other variables remain unchanged, the intention will increase by 0.59 on average.
- when sex=1, the effect of fixation is $0.59 + 1.17 = 1.76$; thus, for a female, when the fixation increases by one second, and all other variables remain unchanged, the intention will increase by 1.76 on average.

The p-value corresponding to the interaction term is small (0.001562) and thus we can conclude that the effect of fixation is significantly different for males and females, even after adjusting for emotion, age, revenue, eduation and status.

9) Multicollinearity

We'll consider an illustration to further illustrate the notion of multicollinearity. Suppose that we wish to model height (response variable) as a function of age (explanatory variable). Suppose that the true relationship between the two is given by

$$height = 20 + 3 \times age + \epsilon$$

Data were generated according to the true model $height = 20 + 3age + \epsilon$, so we should get β_0 close to 20 and β_1 close to 3 in a simple linear regression model including age only.

Now, the data actually contains multiple "copies" of the age variable:

- age is the original variable
- age2 is an exact copy of age

- age3 is highly correlated with age
- age4 is highly correlated with age

```
# illustration
ex<-read.table("Data/colinear.txt",header = TRUE)
head(ex)
```

```
##     height   age  age2     age3     age4
## 1 177.9028 50.59 50.59 49.80756 50.05061
## 2 165.4601 42.94 42.94 43.81477 45.41274
## 3 198.9979 66.13 66.13 65.68266 61.47424
## 4 196.2888 60.10 60.10 60.65516 61.98370
## 5 141.8438 40.55 40.55 40.47446 37.85264
## 6 186.8983 53.96 53.96 55.26368 58.01930
```

```
attach(ex)
```

```
## The following object is masked from intention:
##
##     age
```

```
summary(age-age2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```

```
cor(age,age3)
```

```
## [1] 0.9941008
```

```
cor(age,age4)
```

```
## [1] 0.9114832
```

```
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
rcorr(cbind(age,age2,age3,age4))
```

```
##       age age2 age3 age4
## age  1.00 1.00 0.99 0.91
## age2 1.00 1.00 0.99 0.91
## age3 0.99 0.99 1.00 0.91
## age4 0.91 0.91 0.91 1.00
##
## n= 100
##
##
## P
##      age age2 age3 age4
## age       0    0    0
## age2  0         0    0
## age3  0   0         0
## age4  0   0    0
```

```
#
```

Let's see what happens when we explore linear models including these copies of age:

```
summary(lm(height~age,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age, data = ex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.3290  -6.1464   0.4277   7.3400  19.4001
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.68136    5.01529   4.323  3.7e-05 ***
## age          2.98631    0.09891  30.192  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.357 on 98 degrees of freedom
## Multiple R-squared:  0.9029, Adjusted R-squared:  0.9019
## F-statistic: 911.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm(height~age+age2,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age + age2, data = ex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.3290  -6.1464   0.4277   7.3400  19.4001
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.68136    5.01529   4.323  3.7e-05 ***
## age          2.98631    0.09891  30.192  < 2e-16 ***
## age2              NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.357 on 98 degrees of freedom
## Multiple R-squared:  0.9029, Adjusted R-squared:  0.9019
## F-statistic: 911.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

The model which includes age + age2 cannot be fit since the two variables are identical.

Now let's see what happens with age and age3:

```
summary(lm(height~age,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age, data = ex)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.3290  -6.1464   0.4277   7.3400  19.4001
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.68136    5.01529   4.323  3.7e-05 ***
## age          2.98631    0.09891  30.192  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.357 on 98 degrees of freedom
## Multiple R-squared:  0.9029, Adjusted R-squared:  0.9019
## F-statistic: 911.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm(height~age3,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age3, data = ex)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.344  -6.669   0.492   5.595  18.902
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.88782    4.96495   4.408 2.67e-05 ***
## age3         2.98585    0.09803  30.458  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.283 on 98 degrees of freedom
## Multiple R-squared:  0.9045, Adjusted R-squared:  0.9035
## F-statistic: 927.7 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm(height~age+age3,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age + age3, data = ex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.4781  -6.4765   0.4574   6.3562  18.5538
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.3685     4.9523   4.315 3.85e-05 ***
## age           1.2841     0.9000   1.427   0.1569
## age3          1.7106     0.8991   1.903   0.0601 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.234 on 97 degrees of freedom
```

```
## Multiple R-squared:  0.9064, Adjusted R-squared:  0.9045
## F-statistic: 469.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

We see that not only do the coefficient estimates change (the effect is split up between the two variables in the model including age+age3), but the standard error is 10 times as high!

Finally, the models including age and age4:

```
summary(lm(height~age,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age, data = ex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -25.3290  -6.1464   0.4277   7.3400  19.4001
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 21.68136    5.01529   4.323  3.7e-05 ***
## age          2.98631    0.09891  30.192  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.357 on 98 degrees of freedom
## Multiple R-squared:  0.9029, Adjusted R-squared:  0.9019
## F-statistic: 911.5 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm(height~age4,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age4, data = ex)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.041  -9.169  -0.072  10.630  30.619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  48.0238     7.1518   6.715 1.24e-09 ***
## age4          2.5053     0.1432  17.496  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.79 on 98 degrees of freedom
## Multiple R-squared:  0.7575, Adjusted R-squared:  0.755
## F-statistic: 306.1 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm(height~age+age4,data=ex))
```

```
##
## Call:
## lm(formula = height ~ age + age4, data = ex)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -25.4301  -5.9024   0.3392   7.3558  19.6109
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.7322     5.0408   4.311  3.9e-05 ***
## age           2.9148     0.2416  12.066  < 2e-16 ***
## age4          0.0718     0.2213   0.325    0.746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.4 on 97 degrees of freedom
## Multiple R-squared:  0.903,  Adjusted R-squared:  0.901
## F-statistic: 451.7 on 2 and 97 DF,  p-value: < 2.2e-16
```

```
###
```

We see the same thing as before, but the effects of the collinearity are slightly less pronounced.

Here is another fictional example. The data include a response variable Y and 5 predictor variables, $X_1, \ldots, X_5$. The values of Y were actually randomly generated from the model

$$Y = X_1 + X_2 + X_3 + X_4 + X_5 + \epsilon$$

That is, each $\beta_j = 1$, $j = 1, \ldots, 5$.

```
reglin8<-read.csv("Data/reglin8.csv")
head(reglin8)
```

```
##         Y      X1      X2     X3      X4      X5
## 1 33.6675 1.47107 8.63666 6.4654 5.90786 9.48278
## 2 22.3889 2.34301 4.53524 3.5650 1.98321 2.98616
## 3 32.7527 8.68862 6.56255 9.3035 8.30359 5.19533
## 4 23.2555 3.25433 7.92420 5.3793 9.87484 1.86095
## 5 17.8117 3.61252 2.13514 3.2009 1.04804 6.02107
## 6 30.9873 6.96577 8.33611 6.7651 5.32336 2.08173
```

```
library("Hmisc")
rcorr(as.matrix(reglin8))
```

```
##       Y   X1    X2   X3   X4    X5
## Y  1.00 0.45  0.46 0.65 0.41  0.35
## X1 0.45 1.00  0.06 0.69 0.15  0.02
## X2 0.46 0.06  1.00 0.65 0.07 -0.03
## X3 0.65 0.69  0.65 1.00 0.16  0.01
## X4 0.41 0.15  0.07 0.16 1.00  0.11
## X5 0.35 0.02 -0.03 0.01 0.11  1.00
##
## n= 100
##
##
## P
##    Y      X1     X2     X3     X4     X5
## Y         0.0000 0.0000 0.0000 0.0000 0.0004
## X1 0.0000        0.5795 0.0000 0.1485 0.8532
## X2 0.0000 0.5795        0.0000 0.4736 0.7684
## X3 0.0000 0.0000 0.0000        0.1145 0.9475
## X4 0.0000 0.1485 0.4736 0.1145        0.2644
```

```
## X5 0.0004 0.8532 0.7684 0.9475 0.2644
#
```

We see that the correlation between Y and each X_j is significant and positive. Thus, if we fit a separate simple linear regression model of the form $Y = \beta_0 + \beta_1 X_j + \epsilon$ for each of the five explanatory variables, the slope estimate $\hat{\beta}_1$ would always be positive and significant for each $j = 1, \ldots, 5$. However, X_1, and X_2 are both correlated with X_3, which could cause multicollinearity problems. Fitting the model with all 5 variables, along with the VIFs, yields the following:

```
# VIF
library(car)
mod.col1<-lm(Y~.,data=reglin8)
summary(mod.col1)
```

```
##
## Call:
## lm(formula = Y ~ ., data = reglin8)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6698  -4.7469  -0.2971   4.7987  12.8709
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7611     2.4324  -0.313   0.7551
## X1            0.4315     0.4583   0.942   0.3488
## X2            0.6889     0.4564   1.510   0.1345
## X3            1.9405     0.7731   2.510   0.0138 *
## X4            1.0633     0.2459   4.325 3.80e-05 ***
## X5            1.1443     0.2323   4.926 3.59e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.108 on 94 degrees of freedom
## Multiple R-squared:  0.6193, Adjusted R-squared:  0.5991
## F-statistic: 30.59 on 5 and 94 DF,  p-value: < 2.2e-16
```

```
vif(mod.col1)
```

```
##       X1       X2       X3       X4       X5
## 3.756091 3.383060 6.427886 1.041615 1.015066
```

```
#
```

Overall, the model seems OK. The R^2 is 62%. However, the variables X_1 and X_2 are no longer significant in the model, even though we know that they are individually significant. The VIF of X_3 is quite large (6.43), and the VIFs for X_1 and X_2 are between 3 and 4. This indicates a possible problem of collinearity $\rightarrow$ the estimation for these parameters is not as precise as it would be if there was no multicollinearity. Note that the VIF is an individual measure, it does not tell us which particular variables are correlated with each other. How to proceed completely depends on the research goal and the context of the problem.

```
# une facon d'incorporer les variables collineaires:
#    Combinaison des trois variables X1, X2 et X3
#    en une nouvelle variable (la moyenne des 3)
# one way to incoporate the collinear variables:
#   Combination of the three variables X1, X2, X3
#   and create a new variable that is their average
```

```r
attach(reglin8)
temp<-reglin8
temp$avg123<-(X1+X2+X3)/3
head(temp)
```

```
##         Y      X1      X2     X3      X4      X5   avg123
## 1 33.6675 1.47107 8.63666 6.4654 5.90786 9.48278 5.524377
## 2 22.3889 2.34301 4.53524 3.5650 1.98321 2.98616 3.481083
## 3 32.7527 8.68862 6.56255 9.3035 8.30359 5.19533 8.184890
## 4 23.2555 3.25433 7.92420 5.3793 9.87484 1.86095 5.519277
## 5 17.8117 3.61252 2.13514 3.2009 1.04804 6.02107 2.982853
## 6 30.9873 6.96577 8.33611 6.7651 5.32336 2.08173 7.355660
```

```r
#
mod.col2<-lm(Y~avg123+X4+X5,data=temp)
summary(mod.col2)
```

```
##
## Call:
## lm(formula = Y ~ avg123 + X4 + X5, data = temp)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.0125  -4.4574  -0.1264   4.5203  12.7766
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.8965     2.4170  -0.371    0.712
## avg123        3.1079     0.3334   9.323 4.24e-15 ***
## X4            1.0624     0.2454   4.329 3.67e-05 ***
## X5            1.1475     0.2320   4.947 3.21e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.104 on 96 degrees of freedom
## Multiple R-squared:  0.6117, Adjusted R-squared:  0.5996
## F-statistic: 50.41 on 3 and 96 DF,  p-value: < 2.2e-16
```

```r
vif(mod.col2)
```

```
##   avg123       X4       X5
## 1.025629 1.038809 1.013266
```