

TMA4315: Compulsory exercise 1

Group 13: Magnus Liland, Jakob Gerhard Martinussen and Emma Skarstein

28.09.2018

Contents

Part 1: Explanatory analysis of the dataset	1
Part 2: Simple linear regression with the <code>mylm</code> package	3
Part 3: Multiple linear regression	8
Part 4: Testing the <code>mylm</code> package	10
Source code	13

Part 1: Explanatory analysis of the dataset

Importing the data set

The dataset we will work on is from Canada, and consists of 3987 observations on the following 5 variables:

- `wages`, composite hourly wage rate from all jobs
- `education`, number of years in schooling
- `age`, in years
- `sex`, Male or Female
- `language`, English, French or Other

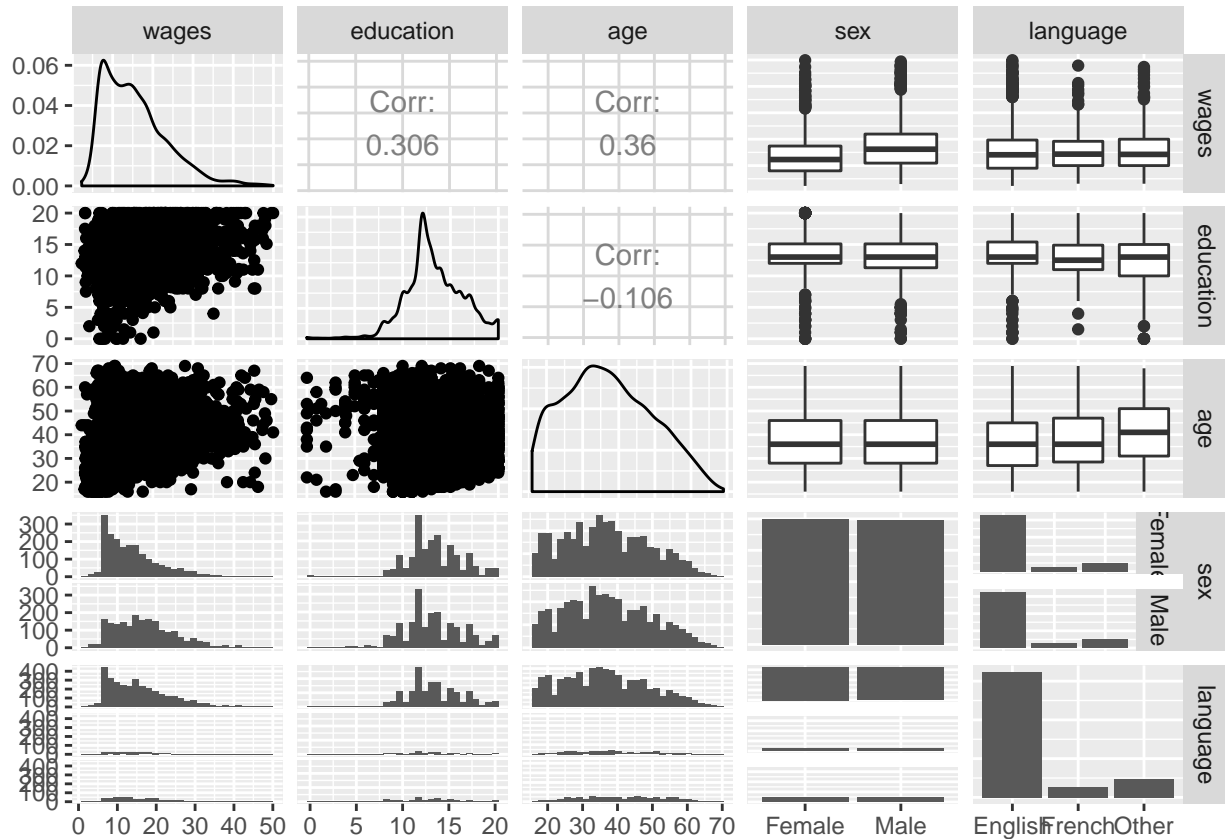
We will store this data set in a SLID dataframe, and remove all rows with incomplete data:

```
library(car)
data(SLID, package = "carData")
SLID <- SLID[complete.cases(SLID), ]
```

a) Analyze relationships between some of the variables

First, let's plot a diagnostic plot of all the variables:

```
library(GGally)
ggpairs(data = SLID)
```



We can now make some remarks regarding the relationship between **wages** and some of the explanatory variables.

Effect of education on wages

There is correlation of 0.306 between **education** and **wages**, meaning that you would *expect* more educated workers to earn more.

Looking at the scatter plot, it shows that there is a much greater spread in education among low paid workers, while better wages require a minimum level of education in most cases.

With other words, a high degree of education does not guarantee greater wages, while high wages require higher education.

*PS: Much of the same can be said on the effect of **age** on **wages**.*

Effect of sex on wages

Men earn on average more than women, as low paid jobs are over-represented by women, and high paid jobs are over-represented by men. The extremal values (minimum and maximum wages) are though *approximately* equal among the sexes.

Assumptions of the MLR analysis

Some assumptions must be made regarding the data if we are to perform a successful multiple linear regression analysis on the data.

We construct a classical linear model of the form:

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon$$

Let's explain the notation:

- We have n observations and k covariates.
- Denote $p := k + 1$, as we will include an intercept parameter.
- \mathbf{Y} is a $n \times 1$ column vector containing respective response values for each observation.
- \mathbf{X} is a $n \times p$ matrix, containing individual observations on each row, and respective covariates on each column. The first column contains solely 1s, as this is the intercept "covariate".
- β is $p \times 1$ column vector containing model coefficient parameters $[\beta_0, \beta_1, \dots, \beta_k]^T$.
- ε is the error term, being a $n \times 1$ column vector.

Now onto the assumptions we must make regarding this classical linear model.

1) $E[\varepsilon] = \mathbf{0}$

The sum of error terms must converge to zero as n approaches infinity.

2) $\text{Cov}(\varepsilon) = E(\varepsilon\varepsilon^T) = \sigma^2\mathbf{I}$

Error terms must be completely independent and have identical variances.

3) $\text{rank}(\mathbf{X}) = \mathbf{k} + \mathbf{1} = \mathbf{p}$

\mathbf{X} must have full rank, i.e. no column should be a linear combination of the other columns. We must also have $p \leq n$. In most cases we have $p \ll n$.

In addition, if we want a classical *normal* linear regression model, we must assume:

4) $\varepsilon \sim N_n(\mathbf{0}, \sigma^2\mathbf{I})$

The error terms must be *normally* distributed, in addition to the earlier assumptions (above).

Part 2: Simple linear regression with the `mylm` package

a) Estimation of Coefficients

Theory

The parameter estimates are calculated by using the maximum likelihood estimator:

$$\hat{\beta} = (X^T X)^{-1} X^T \mathbf{Y}.$$

This estimator has the distribution $\hat{\beta} \sim N_p(\beta, \sigma^2(\mathbf{X}^T \mathbf{X})^{-1})$.

Implementation

This parameter estimation is implemented in the `mylm` module and the results can be shown with `print.mylm` on the resulting model object.

```
library(mylm)
modell1 <- mylm(wages ~ education, data = SLID)
print(modell1)
```

```
## Call:
## mylm(formula = wages ~ education, data = SLID)
##
## Coefficients:
```

```
## (Intercept) education
## 1 4.971691 0.7923091
```

We can compare these results with the `lm` module in R's standard library

```
model1b <- lm(wages ~ education, data = SLID)
print(model1b)
```

```
##
## Call:
## lm(formula = wages ~ education, data = SLID)
##
## Coefficients:
## (Intercept)    education
## 4.9717      0.7923
```

The results are numerically identical, as intended.

b) Estimation of Covariance Matrix

Theory

Estimators

The restricted maximum likelihood estimator for σ^2 , denoted as $\hat{\sigma}^2$, is

$$\hat{\sigma}^2 = \frac{1}{n-p} (\mathbf{Y} - \mathbf{X}\hat{\beta})^T (\mathbf{Y} - \mathbf{X}\hat{\beta}) = \frac{\text{SSE}}{n-p},$$

with the following chi-squared distribution

$$\frac{(n-p)\hat{\sigma}^2}{\sigma^2} \sim \chi_{n-p}^2.$$

We can therefore estimate the covariance matrix for the parameter estimates as

$$\widehat{\text{Cov}}(\hat{\beta}) = \hat{\sigma}^2 (\mathbf{X}^T \mathbf{X})^{-1}.$$

We will use the residuals as estimators for the standard errors

$$\hat{\mathbf{e}} = \mathbf{Y} - \mathbf{X}^T \hat{\beta}.$$

Test statistics

Now let's define the following notation

$$c_{jj} := (X^T X)_{jj}^{-1},$$

i.e. c_{jj} being the j th diagonal element in $(X^T X)_{jj}^{-1}$.

We can now formulate a test statistic for the model parameter estimates

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\hat{\sigma}} \sim t_{n-p}.$$

In our implementation we will approximate this T_j test statistic with the asymptotic distribution

$$T_j = \frac{\hat{\beta}_j - \beta_j}{\sqrt{c_{jj}}\hat{\sigma}} \approx N(0, 1),$$

and denote it as Z_j from here on.

The distribution of $\hat{\sigma}^2$, it is actually distributed according to a t -distribution $n - p$ degrees of freedom. However, when n is large it is reasonable to approximate this distribution by a normal distribution.

The corresponding p-value for the test statistic is the probability of observing the test statistic or a more extreme outcome, i.e. in a world where the null hypothesis holds true, and there is really no dependence of the j th covariate on the response Y .

This will also make the extension to generalized linear models in the next report simpler.

Inference

To obtain the test statistic for each of the parameters, we calculate the test statistic under the null hypothesis that the true parameter value is 0. That is

$$z_i = \frac{\hat{\beta}_i - 0}{\hat{\sigma}\sqrt{c_{ii}}},$$

The p -values for the parameter estimates can therefore be calculated as

$$p_i = P(Z_i > |z_i|).$$

Implementation

The formula for $\widehat{\text{Cov}}(\hat{\beta})$ has been implemented in the `mylm` package, and can be accessed from the returned model object as follows

```
modell$covmatrix
```

```
##           (Intercept)    education
## (Intercept)  0.28532651 -0.020338410
## education   -0.02033841  0.001524956
```

By implementing the `summary` function, using the formulas explained above, we are able to calculate the estimates and standard errors of the intercept and regression coefficient for this model, and test their significance.

```
summary(modell)
```

```
## Call:
## mylm(formula = wages ~ education, data = SLID)
##
## Coefficients:
##           Estimate Std. Error  z-value    p-value
## (Intercept)  4.9716912  0.53415963  9.307501 1.308757e-20
## education    0.7923091  0.03905069 20.289248 1.600242e-91
```

```
## F-statistic: [1] 411.4471
## Chi-statistic: [1] 1639617
## p-value chi-statistic: [1] 0
## R-squared:[1] 0.09358627
```

The interpretation of the parameter estimates is that as the corresponding x_i increases by one, y is expected to increase by $\hat{\beta}_i$.

We see that both z-values are very large for a standard normal distribution, and correspondingly, the p -values are very small, which seems quite reasonable.

We may also compare to the model implemented with the regular `lm` function:

```
summary(model1b)
```

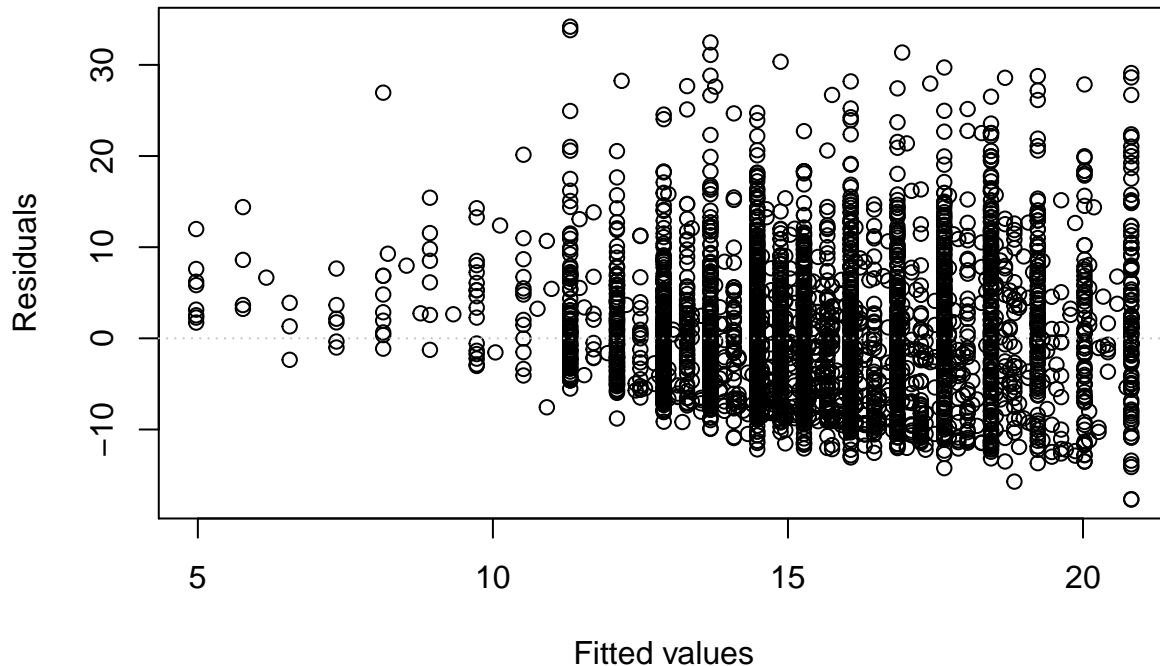
```
##
## Call:
## lm(formula = wages ~ education, data = SLID)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.688  -5.822  -1.039   4.148  34.190
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.97169    0.53429   9.305  <2e-16 ***
## education    0.79231    0.03906  20.284  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.492 on 3985 degrees of freedom
## Multiple R-squared:  0.09359,    Adjusted R-squared:  0.09336
## F-statistic: 411.4 on 1 and 3985 DF,  p-value: < 2.2e-16
```

We see that the results are the same, except for the z-values, which are not supposed to be the same.

c) Residual Plot

We then implement the plot function, in order to plot the fitted values against the residuals.

```
plot(model1)
```



It seems that the mean of the residuals is approximately 0 - that is good, as this is one of our assumptions. However, there is asymmetry, as there is a thicker positive tail. We notice a hint of multiplicative residuals, which means the residuals are not independent of the fitted values.

d) Residual Analysis

Q: What is the residual sum of squares (SSE) and the degrees of freedom for this model?

A: SSE is 223694, with number of degrees of freedom $n - p = 3987 - 2 = 3985$

Q: What is total sum of squares (SST) for this model? Test the significance of the regression using a χ^2 -test.

A: SST is 246790, with p-value of the χ^2 -test equal to 0

In order to test the significance of regression we calculate the F-statistic

$$F = \frac{n - p}{k} \frac{R^2}{1 - R^2},$$

where $R^2 = 1 - \frac{SSE}{SST}$ is the coefficient of determination (see next point). Since this has a Fisher distribution with k and $n - p$ degrees of freedom, we know that $k \cdot F$ has a chi squared distribution with k degrees of freedom. This is what we use to calculate the p-value.

Q: What is the relationship between the χ^2 - and z -statistic in simple linear regression? Find the critical value(s) for both tests.

A: In the simple linear regression case we have that the χ^2 -statistic is the squared of the z -statistic. This we can also see from the `summary` of `model1`:

```
model1$chistatistic
```

```
## [1] 1639617
```

```
model1$zvalues[2]^2
```

```
## [1] 411.6536
```

e) Coefficient of Determination

The coefficient of determination R^2 gives the proportion of variance in the data explained by the model. We calculate it as mentioned above,

$$R^2 = 1 - \frac{SSE}{SST},$$

```
model1$Rsq
```

```
## [1] 0.09358627
```

Part 3: Multiple linear regression

a) Multiple Linear Regression

The theory mentioned earlier has already been provided in vector notation, and thus applies to both simple and multiple linear regression. Here we show the functions applied on a model with multiple covariates.

```
model1 <- mylm(wages ~ education, data = SLID)
model2 <- mylm(wages ~ education + age, data = SLID)
print(model2)
```

```
## Call:
## mylm(formula = wages ~ education + age, data = SLID)
##
## Coefficients:
## (Intercept) education
## 1 -6.021653 0.9014644
```

b) Model Analysis

```
summary(model2)
```

```
## Call:
## mylm(formula = wages ~ education + age, data = SLID)
##
## Coefficients:
##           Estimate Std. Error  z-value    p-value
## (Intercept) -6.0216529 0.618690864 -9.732894 2.182914e-22
## education    0.9014644 0.035746370 25.218347 2.520521e-140
## age          0.2570898 0.008947866 28.731967 1.521861e-181
## F-statistic: [1] 660.7096
## Chi-statistic: [1] 2632267
## p-value chi-statistic: [1] 0
## R-squared: [1] 0.2490697
```

The estimated parameters for the multiple linear regression case are calculated as described in part 1. For the model with `education` and `age` as predictors for the response `wage`, the estimates for the coefficients are given in the summary above. They are 0.9014644 and 0.2570898 for `education` and `age` respectively.

With a similar approach as described in part 2, we calculate the observed z -statistic for each of the coefficients under the null hypothesis that the true parameter values are 0. The test statistics can be found in the

summary, and the p-values for both the coefficients, and the intercept, are highly significant. Hence, we can with some confidence say that both **education** and **age** have some predictive power with respect to the **wage**.

The interpretation of the parameters are simply that as one of the covariates increase by one unit, the expected wage increase by the value of the coefficient. This means that as a person grows one year older, he/she is expected to increase his/her income by 0.2570898. Similarly, as a person takes one year with education he/she is expected to grow his/her income by 0.9014644.

c) Effect of Education and Age on Wages

In order to investigate the effect on **age** and **education** both separately, and together, we fit three models - one with only **age**, one with only **education** and one with both. We begin by looking at the estimated coefficients in all three models again.

```
print(model1)
```

```
## Call:
## mylm(formula = wages ~ education, data = SLID)
##
## Coefficients:
## (Intercept) education
## 1 4.971691 0.7923091
```

```
print(model2)
```

```
## Call:
## mylm(formula = wages ~ education + age, data = SLID)
##
## Coefficients:
## (Intercept) education
## 1 -6.021653 0.9014644
```

```
model3 = mylm(wages ~ age, data = SLID)
```

```
print(model3)
```

```
## Call:
## mylm(formula = wages ~ age, data = SLID)
##
## Coefficients:
## (Intercept) education
## 1 6.890901 0.2331079
```

We observe that the parameter estimates differ, due to the fact that the covariates are dependent of each other. This we also observe by studying the covariance matrix for the estimated coefficients in the model with both **age** and **education**, where the elements off the diagonal are non-zero. It means that as one of the covariates varies, it will affect the other covariates that it depends on.

On the other hand, if the covariates were completely independent, then the coefficients would remain the same, even if another covariate was added or removed. This is not the case here, which is quite intuitive considering that as one grows older one would typically have more education - and the other way around.

Part 4: Testing the mylm package

In the following section the `mym` function will be employed to three different model suggestions. To evaluate the fit of these models we'll print the summary of the models to discuss the p-values of the parameters, as well as the residual plots to see if there are any clear trends in the residuals, or any discrepancies from the MLR assumptions.

Model 1: `sex`, `age`, `language` and `education2`

```
model4a <- mylm(formula = wages ~ sex + age + language + I(education2),
  data = SLID)
summary(model4a)
```

```
## Call:
## mylm(formula = wages ~ sex + age + language + I(education2),
##      data = SLID)
##
## Coefficients:
##              Estimate Std. Error   z-value    p-value
## (Intercept)  -1.87553134 0.440013681 -4.2624387 2.022080e-05
## sexMale      3.40870006 0.208262748 16.3673057 3.273901e-60
## age          0.24862499 0.008656104 28.7225042 1.997899e-181
## languageFrench -0.07553202 0.424815732 -0.1777995 8.588804e-01
## languageOther -0.13454020 0.322909303 -0.4166501 6.769343e-01
## I(education2) 0.03481515 0.001288925 27.0110041 1.097508e-160
## F-statistic: [1] 344.8469
## Chi-statistic: [1] 1372836
## p-value chi-statistic: [1] 0
## R-squared: [1] 0.3022198
```

From the summary of the model fit above the first thing to note is that the intercept is negative, implying that the “base case” (all covariates set equal to 0) has negative wage. This may be considered non-intuitive, and an indication that the model fit is perhaps not the best.

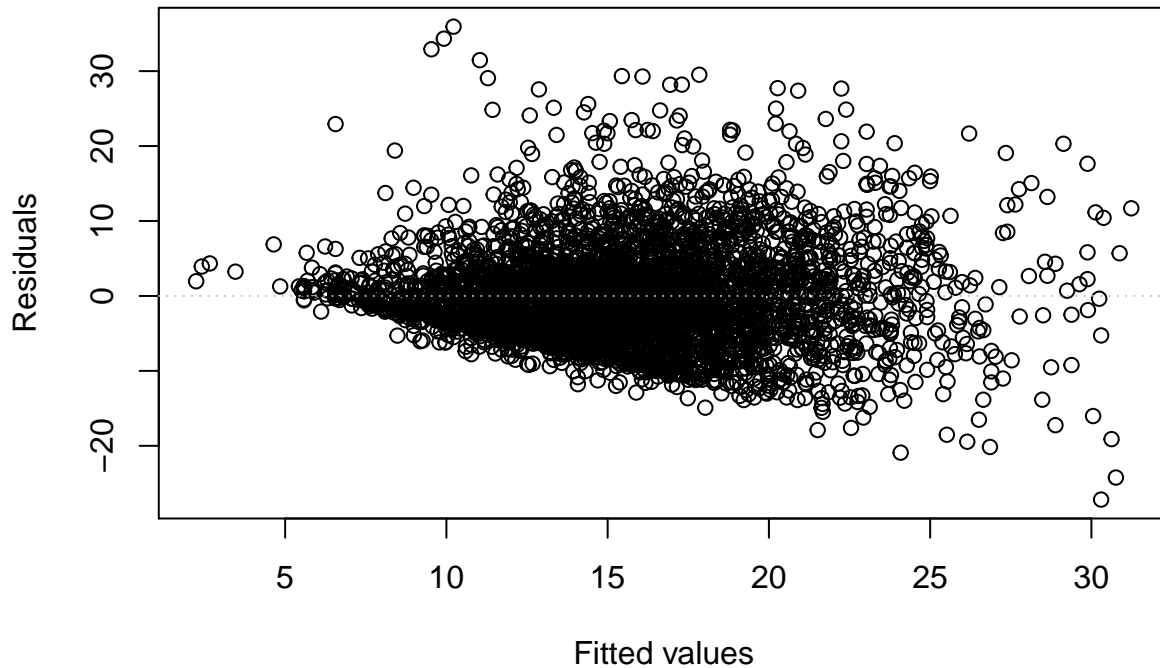
What is the base case in this model? A female, English speaking newborn with no education. You could therefore argue that the non-intuitive intercept is acceptable, as this is not in the domain of interest.

Further, the p-values for `sex`, `age` and `education2` indicate that all these parameters have a significant effect on wages, while the effect from `language` is not quite as clear.

Finally, note that the effect from `education` is aggregated, as the effect of wages from an increase in `education` will be quadratic rather than linear. There is no clear indication why this type of response is better than a linear response.

Residual analysis

```
plot(model4a)
```



In the plot of the residuals a clear trend is visible. From low fitted values the residuals are small, while they increase with larger fitted values. In addition, it seems that the residuals are somewhat asymmetric, with larger positive values than negative. This further support the claim that the model fit is not particularly good, as these observations do not coincide with the assumptions made regarding the distribution of ε .

Improvements

A suggested improvement to this model is to not use education as a quadratic parameter, but rather use it as a linear one. Using education as a linear parameter has shown success in earlier model attempts above, and there is no clear quadratic trend to observe in the wage-education distribution shown in part 1.

Model 2: age, language and interaction term

Now we model the effect of language and age on wage, including interaction effects.

```
model4b <- mylm(formula = wages ~ language * age, data = SLID)
summary(model4b)
```

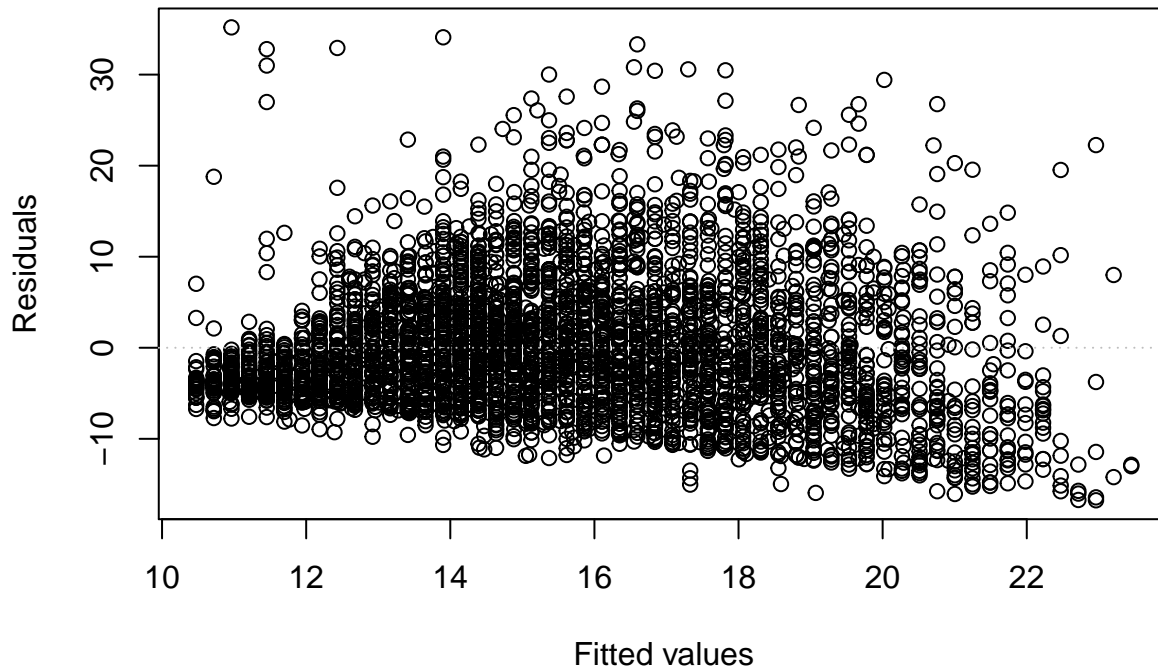
```
## Call:
## mylm(formula = wages ~ language * age, data = SLID)
##
## Coefficients:
##              Estimate Std. Error   z-value    p-value
## (Intercept)   6.55579376  0.41037150  15.9752657  1.900430e-57
## languageFrench  2.86062508  1.59487047   1.7936410  7.287049e-02
## languageOther  0.84862130  1.23425214   0.6875591  4.917305e-01
## age            0.24485160  0.01067947  22.9273089  2.482113e-116
## languageFrench:age -0.08392752  0.04042557  -2.0760996  3.788474e-02
## languageOther:age -0.03701381  0.02931813  -1.2624888  2.067730e-01
## F-statistic: [1] 120.2053
## Chi-statistic: [1] 478537.3
## p-value chi-statistic: [1] 0
```

```
## R-squared: [1] 0.1311705
```

The first thing to notice from the summary of the second model, is that the only two parameters that are *clearly* significant is the intercept term and the effect from `age`. The `language` terms, as well as the interaction effect, are not clearly significant, although the effect of `languageFrench` and `age:languageFrench` would be included with a significance level of 5%.

Residual analysis

```
plot(model4b)
```



The plot of the residuals exhibit a similar trend as the previous model, namely that the residuals increase with larger fitted values. In addition, it also seems in this case as if the residuals are asymmetric, the positive residuals are considerably larger in absolute value than the negative. The residuals plot should, if the model assumptions are in fact correct, be normally distributed around the x -axis. One can with some confidence say that this is not the case for this model, as the residuals exhibits some non-random behaviour.

Improvements

Language doesn't seem to be significant and hasn't been significant in previous models either. So we suggest removing language altogether and replacing it with for example education.

Model 3: education without intercept

Now we fit a new model with education as covariate and *without* an intercept.

```
model4c <- mylm(formula = wages ~ -1 + education, data = SLID)
summary(model4c)
```

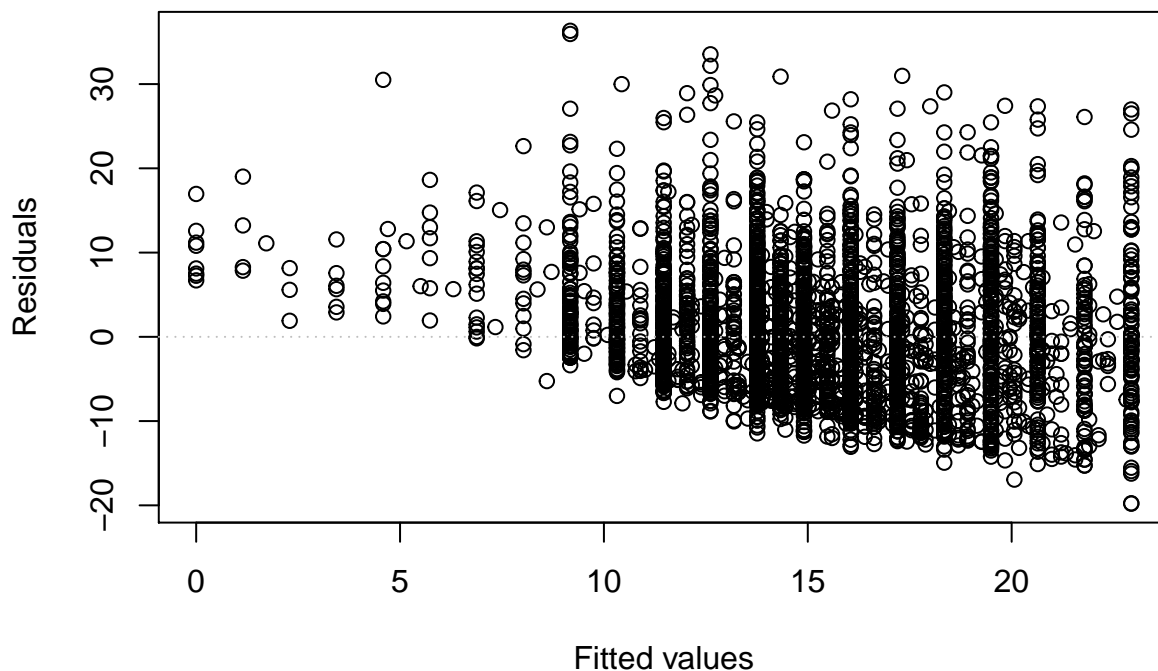
```
## Call:
## mylm(formula = wages ~ -1 + education, data = SLID)
##
## Coefficients:
```

```
##           Estimate Std. Error z-value p-value
## education 1.146697 0.008766101 130.8104      0
## F-statistic: [1] Inf
## Chi-statistic: [1] Inf
## p-value chi-statistic: [1] 0
## R-squared:[1] 0.07389171
```

First thing to notice from this model is that, as there is now interaction term, the only coefficient, `education`, is highly significant. The statistics does not make sense in this case, and can therefore be discarded. In all the previous models the interaction term has been significantly different from 0, indicating that it should be included - there is no indication it should be dropped.

Residual analysis

```
plot(model4c)
```



The residual plot seem to have a decreasing trend with the fitted values, as all residuals are larger than zero for small fitted values, while the majority seem to be negative for larger fitted values. This residual plot does also exhibit asymmetric behaviour, similar to the previous models.

Improvement

An improvement to this model could be to include an intercept, for the reasons mentioned above.

Source code

The source code of the `mylm` package can be found in `R/mylm.R`. The source code repository can be found at github.com/jakobgm/mylm.

Here is the main content of the source code:

```

read_chunk("../R/mylm.R")

if (FALSE) {
  # Part3c
  model_age <- mylm(wages ~ age, data = SLID)
  model_edu <- mylm(wages ~ education, data = SLID)
  model_both <- mylm(wages ~ education + age, data = SLID)
  model_age$coefficients
  model_edu$coefficients
  model_both$coefficients
}

if (FALSE) {
  # Part4
  model4a <- mylm(formula = wages ~ sex + age + language + I(education^2),
    data = SLID)
  summary(model4a)
  # negative intercept aggregate response in education
  plot(model4a)
  model4b <- mylm(formula = wages ~ language * age, data = SLID)
  summary(model4b)
  plot(model4b)
  model4c <- mylm(formula = wages ~ -1 + education, data = SLID)
  summary(model4c)
  plot(model4c)
  # although some of these plots do not look horrible, they all have a
  # clear trend. They should be completely random, which they are not.
  # The intercept are clearly different from 0, indicating it should be
  # included.
}

# Select Build, Build and reload to build and lode into the
# R-session.

mylm <- function(formula, data = list(), contrasts = NULL, ...) {
  # Extract model matrix & responses
  mf <- model.frame(formula = formula, data = data)
  X <- model.matrix(attr(mf, "terms"), data = mf, contrasts.arg = contrasts)
  y <- model.response(mf)
  terms <- attr(mf, "terms")

  # Add code here to calculate coefficients, residuals, fitted values,
  # etc... and store the results in the list est
  inv_XtX <- solve(t(X) %*% X)

  # This is our estimation for \beta
  beta <- inv_XtX %*% t(X) %*% y

  # The fitted values are easily calculated
  fitted_values <- X %*% beta

  # These are the predicted values for the simplest possible model,

```

```

# i.e. no covariates at all, only an intercept.
Y_hat_HO <- rep(1, length(fitted_values)) * mean(y)
residuals <- y - fitted_values
SSE <- sum((y - fitted_values)^2)
SST <- sum((y - Y_hat_HO)^2)
sigma_tilde <- SSE/length(y)
covariance_matrix <- sigma_tilde * solve(t(X) %*% X)
# z-values and p-values
t_value <- vector("numeric", length(beta))
p_value <- vector("numeric", length(beta))
for (i in 1:length(beta)) {
  t_value[i] <- (beta[i] - 0)/sqrt(diag(covariance_matrix)[i])
  p_value[i] <- 2 * pnorm(abs(t_value[i]), lower.tail = FALSE)
}
F_obs = ((length(fitted_values) - length(beta)) * (SST - SSE))/(SSE *
  (length(beta) - 1))
Chi <- (length(beta) - 1) * F_obs
p_chi <- pchisq(Chi, df = 1, lower.tail = FALSE)
Rsqr <- 1 - SSE/SST

est <- list(coefficients = beta, pvalues = p_value, zvalues = t_value,
  covmatrix = covariance_matrix, fitted_values = fitted_values,
  residuals = residuals, fstatistic = F_obs, SSE = SSE, SST = SST,
  chistatistic = Chi, p_chi = p_chi, Rsqr = Rsqr, model = mf)

# Store call and formula used
est$call <- match.call()
est$formula <- formula

# Set class name. This is very important!
class(est) <- "mylm"

# Return the object with all results
return(est)
}

print.mylm <- function(object, ...) {
  # Code here is used when print(object) is used on objects of class
  # 'mylm' Useful functions include cat, print.default and format
  cat("Call: \n")
  print(object$call)
  cat("\nCoefficients: \n")
  temp_matrix <- matrix(object$coefficients)
  coeff_matrix <- data.frame(t(temp_matrix))
  colnames(coeff_matrix) <- rownames(object$coefficients)
  rownames(coeff_matrix) <- " "
  print(coeff_matrix)
}

summary.mylm <- function(object, ...) {
  # Code here is used when summary(object) is used on objects of class

```

```

# 'mylm' Useful functions include cat, print.default and format
cat("Call: \n")
print(object$call)
cat("\nCoefficients: \n")
temp_matrix <- matrix(c(object$coefficients, sqrt(diag(object$covmatrix)),
  object$zvalues, object$pvalues), nrow = length(object$coefficients))
coeff_matrix <- data.frame(temp_matrix)
rownames(coeff_matrix) <- rownames(object$coefficients)
colnames(coeff_matrix) <- c("Estimate", "Std. Error", "z-value",
  "p-value")
print(coeff_matrix)
cat("F-statistic: ")
print(object$fstatistic)
cat("Chi-statistic: ")
print(object$chistatistic)
cat("p-value chi-statistic: ")
print(object$p_chi)
cat("R-squared:")
print(object$Rsqr)
}

plot.mylm <- function(object, ...) {
  # Code here is used when plot(object) is used on objects of class
  # 'mylm'
  plot(object$fitted_values, object$residuals, xlab = "Fitted values",
    ylab = "Residuals")
  abline(a = 0, b = 0, col = "gray", lty = 3)
}

# This part is optional! You do not have to implement anova
anova.mylm <- function(object, ...) {
  # Code here is used when anova(object) is used on objects of class
  # 'mylm'

  # Components to test
  comp <- attr(object$terms, "term.labels")

  # Name of response
  response <- deparse(object$terms[[2]])

  # Fit the sequence of models
  txtFormula <- paste(response, "~", sep = "")
  model <- list()
  for (numComp in 1:length(comp)) {
    if (numComp == 1) {
      txtFormula <- paste(txtFormula, comp[numComp])
    } else {
      txtFormula <- paste(txtFormula, comp[numComp], sep = "+")
    }
    formula <- formula(txtFormula)
    model[[numComp]] <- lm(formula = formula, data = object$model)
  }
}

```



```
}  
  
# Print Analysis of Variance Table  
cat("Analysis of Variance Table\n")  
cat(c("Response: ", response, "\n"), sep = "")  
cat("      Df  Sum sq X2 value Pr(>X2)\n")  
for (numComp in 1:length(comp)) {  
  # Add code to print the line for each model tested  
}  
  
return(model)  
}
```