TMA4315: Compulsory exercise 1

Group: Jakob Bergset Heide

21.09.2023

In this project, we will build a R package containing a similar implementation of the `lm` function, called `mylm`. The `mylm` function will be able to calculate coefficients with standard errors, as well as hypothesis testing using both z-tests and χ^2-tests. In addition, the package will include a `plot.mylm` function for plotting residuals vs fitted values, and the functions `print.mylm` and `summary.mylm`, which will be similar to those of the standard `lm`.

Part 1

a)

We start by importing the data and performing some explanatory data analysis.

```r
# install.packages('car')
library(car)
data(SLID, package = "carData")
SLID <- SLID[complete.cases(SLID), ]

summary(SLID)
```

```
##      wages          education          age             sex          language   
##  Min.   : 2.30   Min.   : 0.00   Min.   :16.0   Female:2001   English:3244  
##  1st Qu.: 9.25   1st Qu.:12.00   1st Qu.:28.0   Male  :1986   French : 259  
##  Median :14.13   Median :13.00   Median :36.0                 Other  : 484  
##  Mean   :15.54   Mean   :13.34   Mean   :37.1                               
##  3rd Qu.:19.72   3rd Qu.:15.10   3rd Qu.:46.0                               
##  Max.   :49.92   Max.   :20.00   Max.   :69.0                               
```

```r
str(SLID)
```

```
## 'data.frame':    3987 obs. of  5 variables:
##  $ wages    : num  10.6 11 17.8 14 8.2 ...
##  $ education: num  15 13.2 14 16 15 13.5 12 14 18 11 ...
##  $ age      : int  40 19 46 50 31 30 61 46 43 17 ...
##  $ sex      : Factor w/ 2 levels "Female","Male": 2 2 2 1 2 1 1 1 2 2 ...
##  $ language : Factor w/ 3 levels "English","French",..: 1 1 3 1 1 1 1 3 1 1 ...
```

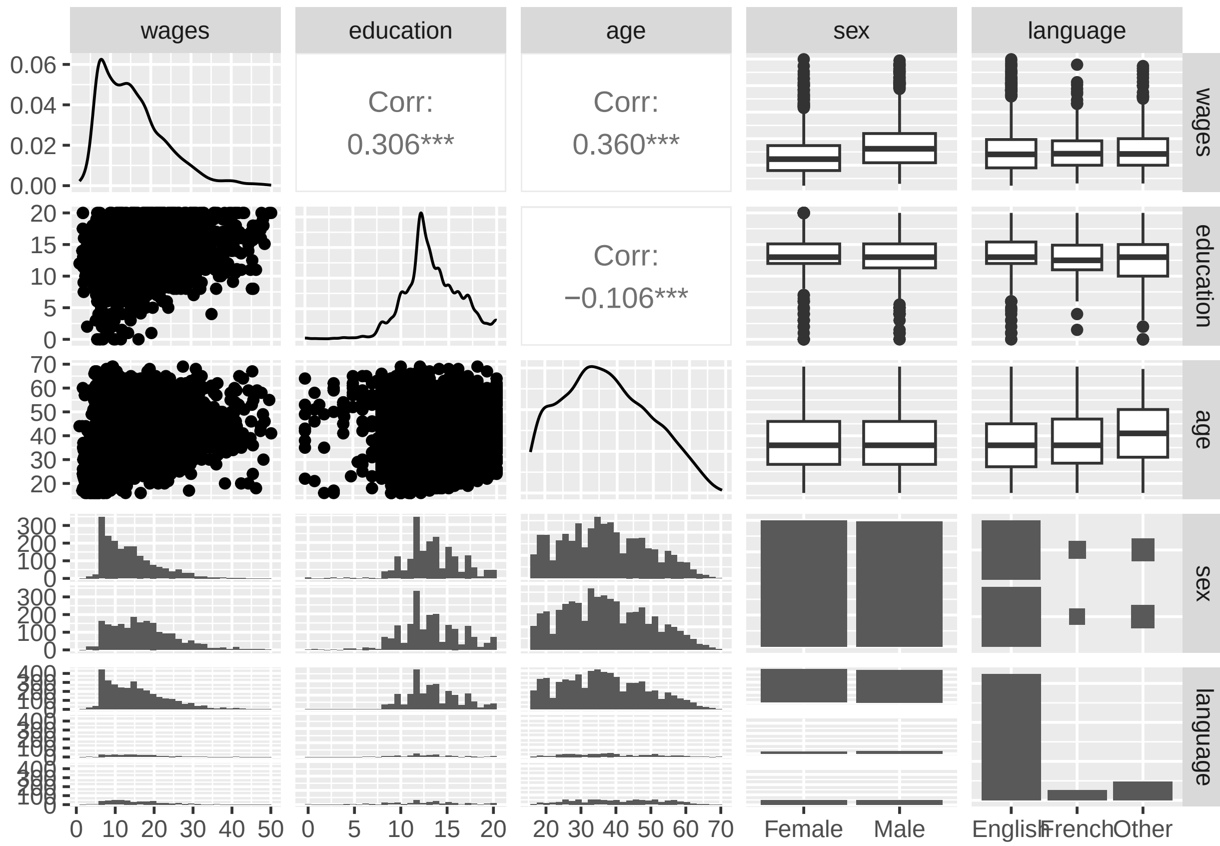
We see that we have the following variables in our dataset:

- `wages`: hourly wage rate - a continuous variable with mean 15.54 and range (2.30,49.92).

- `education`: number of years of education - a continuous variable with mean 13.34 and range (0,20).
- `age`: years of age - integer-valued/continuous variable with mean 37.1 and range (16,69).
- `sex`: gender - categorical variable with 2 levels: "Female","Male".
- `language`: categorical variable with 3 levels: "English","French","Other".

We import the library `ggplot` and use the `ggpairs` function:

```
library(GGally)
ggpairs(SLID)
```



From the plot above, we can draw some conclusions:

- There seems to be a slight correlation between `wages` and `sex`. For `Male`, the median and upper and lower quartiles are higher than for `Female`.
- We have slight positive correlations between `wages` and `education`, as well as `wages` and `age`, with correlations of 0.306 and 0.36, respectively. These numbers indicate a weak correlation.
- On the diagonal of the plot, we can see the distribution of each variable. For example, we see that there are many more data points with `English` than `French` or `Other`. In addition, we note that there are few data points with education less than 10 years.

In order for our model to make sense, we need to make a few key assumptions about our data:

- There exists a linear relationship between response and predictors, that is, $Y = \beta_0 + \beta_1 x_1 + \cdots + \epsilon$
- All observations are observed independently.
- Design matrix $\mathbf{X}$ has full rank: else we cannot invert $(\mathbf{X}^T\mathbf{X})$ and find least squares estimate.
- The error term ϵ is normally distributed with mean 0 and variance $\sigma^2 I$, where I is the identity matrix.

Part 2

We import the package mylm.

```
library(mylm)
```

a)

We fit our first model, which is a simple linear regression with wages as response and education as the only covariate. We can estimate the coefficients β by least squares, i.e.

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

We compare the coefficient estimates from mylm with the ones from mylm using a print.mylm function,

```
model1 <- mylm(wages ~ education, data = SLID)
print.mylm(model1)
```

```
## Info about object
## [1] "Coefficients:"
## (Intercept) education
## 4.971691 0.7923091
```

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

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

We find that the coefficient estimates are the same in both mylm and lm.

b)

We develop the mylm function further, so it can calculate the covariance matrix of the coefficient estimates as $\text{Cov}(\hat{\beta}) = \tilde{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}$ Using this matrix, we can take the square root of the diagonal elements and get the standard errors of the coefficients.

```
summary.mylm(model1)
```

```
## Summary of object
## [1] "Coefficients:"
## (Intercept) education
## 4.971691 0.7923091
## [1] "Std.errors"
## (Intercept)   education
```

```
##  0.53415963  0.03905069
## [1] "z-values"
##                  [,1]
## (Intercept)  9.307501
## education   20.289248
## [1] "p-values"
##                     [,1]
## (Intercept) 1.308757e-20
## education   1.600242e-91
## Chi-test on  1  df:  411.4471  with p-value:  1.774739e-91
## R^2:  0.09358627
```

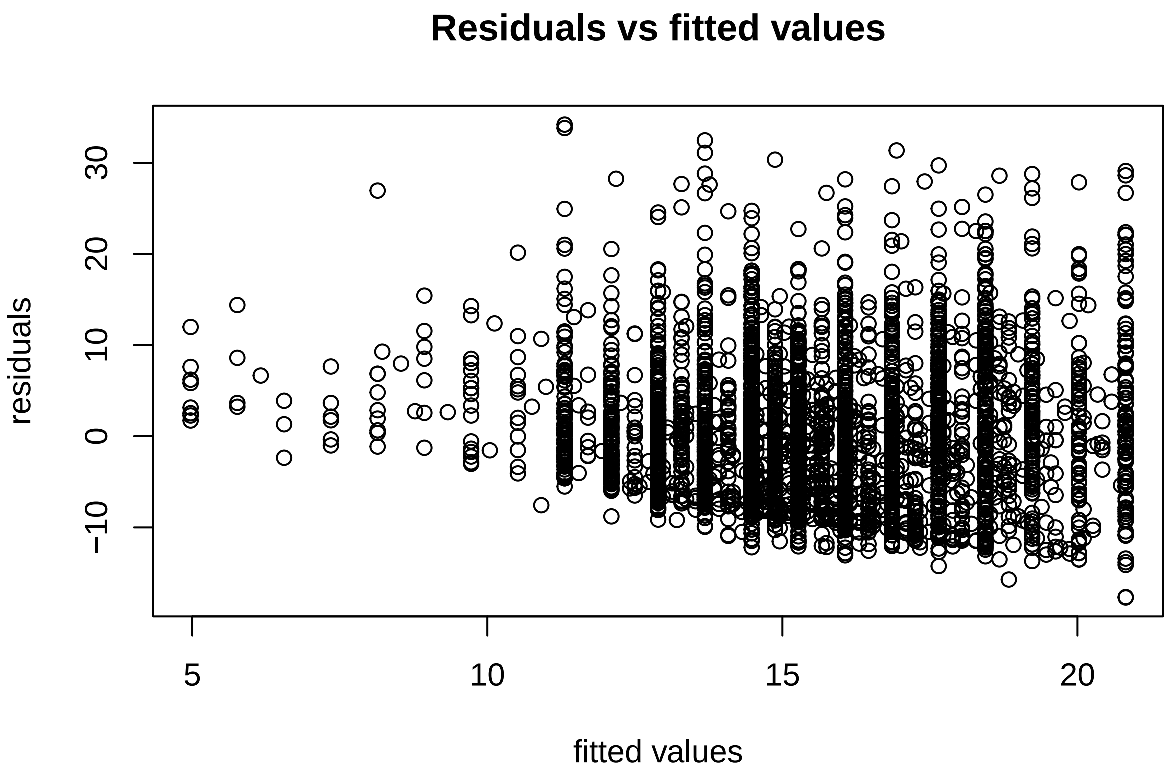
In the summary above, we have the following parameters:

- Coefficient estimates: We have coefficient estimates for the intercept and `education`. The interpretation is that if we increase `education` by 1 (and keep other covariates fixed), then the response will increase by the coefficient estimate of `education`, in this case by 0.7923091.
- Standard errors: these are the estimated standard errors of the coefficient estimates, which we get from the square root of the diagonal elements of $\tilde{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}$ (the covariance matrix of $\hat{\beta}$), where $\tilde{\sigma} = \frac{\text{SSE}}{n}$. For example, in this model, the standard error of $\hat{\beta}_{education}$ is approximately 0.03905.
- z-values: these are the observed test statistics used in the z-test. We can use a z-test instead of a t-test when n is asymptotically large, since the t-distribution then becomes a normal distribution. We calculate the z-statistics as $\frac{\hat{\beta}}{\sqrt{c_{jj}\tilde{\sigma}^2}}$, under the null hypothesis $H_0 : \hat{\beta}_j = 0$, where c_{jj} is the j-th diagonal element of $(\mathbf{X}^T\mathbf{X})^{-1}$.
- p-values: the test statistic is normally distributed under H_0, and so we can calculate a p-value, which is essentially the probability of H_0 being true. For our case, we see that the p-values for the z-tests are low for both the intercept and `education`, and therefore we reject H_0 in all cases.

c)

We implement a `plot.mylm` to create a scatter plot of residuals vs fitted values. The residuals of the model are calculated as $\epsilon = Y - \hat{Y}$.

```r
plot.mylm(model1)
```



We see from the plot that there is some increase in the spread of the residuals as the fitted values increase, which points to heteroscedasticity (variance is not constant as we assumed in the model). In addition, the residuals should have mean 0, but there seems to be more residuals in the region above 0 (although hard to tell from just looking at the plot).

d)

We calculate the sum-of-squares error (SSE), total sum-of-squares (SST), and sum-of-squares regression (SSR) using the following formulas:

- $\text{SSE} = Y^T(I - H)Y$, where $H = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$
- $\text{SSR} = Y^T(H - \frac{1}{n}\mathbf{11})Y$
- $\text{SST} = \text{SSR} + \text{SSE}$

In addition, we can test the hypothesis $H_0 : \beta_1 = \beta_2 = \cdots = \beta_k = 0$, which a test on the significance of the regression. We use the asymptotic χ^2-test with the test statistic $rF_{r,n-p}$, where

$$F_{r,n-p} = \frac{\frac{1}{r}(\text{SST} - \text{SSE})}{\frac{SSE}{n-p}}$$

```r
cat("SSE: ", model1$SSE)
```

```
## SSE:  223694.3
```

```r
cat("\nSST: ", model1$SST)
```

```
##
## SST:  246790.5
```

```r
# Critical values for z-test, with significance level 0.05
cat("Lower:", qnorm(0.025))
```

```
## Lower: -1.959964
```

```r
cat("Upper:", qnorm(0.025, lower.tail = FALSE))
```

```
## Upper: 1.959964
```

```r
# Critical value for X^2-test, with significance level 0.05
cat("Lower:", qchisq(0.05, df = model1$k, lower.tail = FALSE))
```

```
## Lower: 3.841459
```

```r
summary.mylm(model1)
```

```
## Summary of object
## [1] "Coefficients:"
## (Intercept) education
## 4.971691 0.7923091
## [1] "Std.errors"
## (Intercept)   education
##  0.53415963  0.03905069
## [1] "z-values"
##                  [,1]
## (Intercept)  9.307501
## education   20.289248
## [1] "p-values"
##                     [,1]
## (Intercept) 1.308757e-20
## education   1.600242e-91
## Chi-test on  1  df:  411.4471  with p-value:  1.774739e-91
## R^2:  0.09358627
```

In the summary, we see the chi-square test (labelled as F-test) gives a p-value of approximately 0, which deems the regression significant. We also see that this model has $n - p = 3985$ degrees of freedom. For simple linear regression, the z-statistic squared is identical to the χ^2-statistic.

e)

The R^2 is calculated as $\frac{\text{SSR}}{\text{SST}}$.

```r
cat("R^2: ", model1$R2)
```

```
## R^2:  0.09358627
```

The R^2 value tells us the proportion of the variance that is explained by the model.

Part 3

We move on to multiple linear regression. We fit a model with `wages` as the response and `education` and `age` as the covariates.

a)

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

b)

```r
summary.mylm(model2)
```

```
## Summary of object
## [1] "Coefficients:"
## (Intercept) age education
## -6.021653 0.2570898 0.9014644
## [1] "Std.errors"
## (Intercept)         age   education
## 0.618690864 0.008947866 0.035746370
## [1] "z-values"
##                  [,1]
## (Intercept) -9.732894
## age         28.731967
## education   25.218347
## [1] "p-values"
##                      [,1]
## (Intercept)  2.182914e-22
## age         1.521861e-181
## education   2.520521e-140
## Chi-test on  2  df:  1321.419  with p-value:  1.141507e-287
## R^2:  0.2490697
```

In the summary above, we see that the z-tests for the coefficients intercept, `age` and `education` all give p-values close to 0, which deems the coefficients significant. The chi-test also indicates that the regression is significant.

c)

The parameter estimates change because in the simple models we try to explain the response using only one variable. When we add another covariate, there might be some relation between the covariates - multi-collinearity. We can fit the models and check the coefficients:

```
model2a <- mylm(wages ~ age, data = SLID)
print.mylm(model1)  #Only education
```

```
## Info about object
## [1] "Coefficients:"
## (Intercept) education
## 4.971691 0.7923091
```

```
print.mylm(model2a)  #Only age
```

```
## Info about object
## [1] "Coefficients:"
## (Intercept) age
## 6.890901 0.2331079
```

```
print.mylm(model2)  #Age + education
```

```
## Info about object
## [1] "Coefficients:"
## (Intercept) age education
## -6.021653 0.2570898 0.9014644
```

We see that there is a slight change in the coefficient estimates. We can calculate the correlation between the two covariates using the covariance matrix of $\hat{\beta}$:

```
print(model2$beta.matrix)
```

```
##              (Intercept)           age     education
## (Intercept)  0.382778385 -3.423607e-03 -1.830322e-02
## age         -0.003423607  8.006431e-05  3.399374e-05
## education   -0.018303218  3.399374e-05  1.277803e-03
```

```
cat("The correlation between education and age is: ", model2$beta.matrix[3,
    2]/(model2$std.errors[2] * model2$std.errors[3]))
```

```
## The correlation between education and age is:  0.106279
```

The value of 0.106279 indicates a weak correlation between age and education - which might explain the change in coefficient estimates from the two simple models to the one multiple.

Part 4

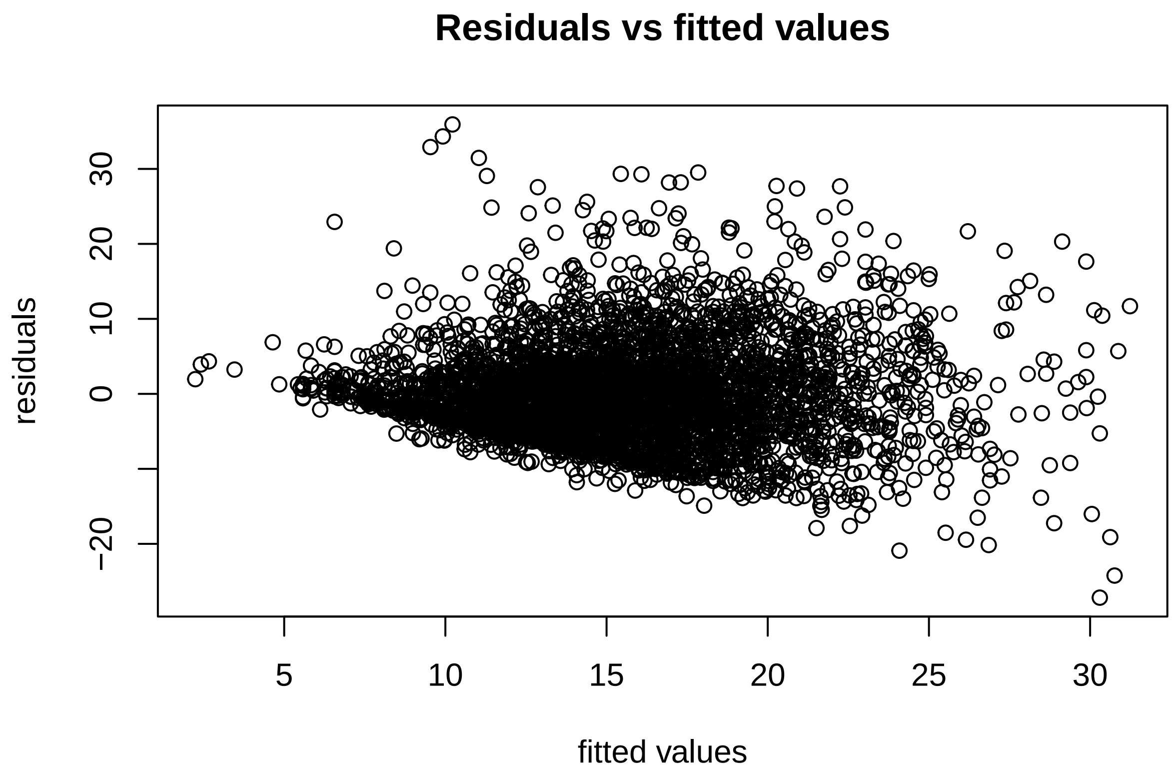
We fit a few different models, and check the various parameters and plots for each model. The first model we fit is a model with `wages` against `sex`, `language`, `age` and `education^2`. To handle the multiple classes of `language`, we employ dummy variable coding.

```
model4a <- mylm(wages ~ sex + language + age + I(education^2), data = SLID,
    contrasts = list(language = "contr.treatment"))  #Dummy variable coding
summary.mylm(model4a)
```

```
## Summary of object
## [1] "Coefficients:"
## (Intercept) sexMale languageFrench languageOther age I(education^2)
## -1.875531 3.4087 -0.07553202 -0.1345402 0.248625 0.03481515
## [1] "Std.errors"
##    (Intercept)        sexMale languageFrench  languageOther            age
##    0.440013681    0.208262748    0.424815732    0.322909303    0.008656104
## I(education^2)
##    0.001288925
## [1] "z-values"
##                      [,1]
## (Intercept)    -4.2624387
## sexMale        16.3673057
## languageFrench -0.1777995
## languageOther  -0.4166501
## age            28.7225042
## I(education^2) 27.0110041
## [1] "p-values"
##                         [,1]
## (Intercept)     2.022080e-05
## sexMale         3.273901e-60
## languageFrench  8.588804e-01
## languageOther   6.769343e-01
## age            1.997899e-181
## I(education^2) 1.097508e-160
## Chi-test on  5  df:  1724.235  with p-value:  0
## R^2:  0.3022198
```

From the output of `summary.mylm`, we see that the z-tests of the coefficients deem the covariates `sexMale`, `age` and `education^2` as statistically significant, while the covariates `languageFrench` and `languageOther` are not, with p-values 0.859 and 0.677, respectively. The χ^2-test deems the regression significant. The model explains 30% of the variance. A plot of the residuals vs fitted values is shown below:

```
plot.mylm(model4a)
```



There is a clear trend in the plot, which is the slope that starts from zero residuals and travels downwards with increasing fitted values.

We fit a new model with the covariates `language`, `age` and the interaction between these two.

```
model4b <- mylm(wages ~ language + age + language:age, data = SLID, contrasts = list(language = "contr.
summary.mylm(model4b)
```

```
## Summary of object
## [1] "Coefficients:"
## (Intercept) languageFrench languageOther age languageFrench:age languageOther:age
## 6.555794 2.860625 0.8486213 0.2448516 -0.08392752 -0.03701381
## [1] "Std.errors"
##        (Intercept)     languageFrench      languageOther                age
##         0.41037150         1.59487047         1.23425214         0.01067947
## languageFrench:age  languageOther:age
##         0.04042557         0.02931813
## [1] "z-values"
##                          [,1]
## (Intercept)        15.9752657
## languageFrench      1.7936410
## languageOther       0.6875591
## age                22.9273089
## languageFrench:age -2.0760996
## languageOther:age  -1.2624888
## [1] "p-values"
##                             [,1]
```

```
## (Intercept)          1.900430e-57
## languageFrench       7.287049e-02
## languageOther        4.917305e-01
## age                 2.482113e-116
## languageFrench:age   3.788474e-02
## languageOther:age    2.067730e-01
## Chi-test on  5  df:  601.0265  with p-value:  1.213615e-127
## R^2:  0.1311705
```

We see that the interaction `languageFrench:age` is deemed significant (with significance level $\alpha = 0.05$), with a p-value of 0.0379. `age` is also significant, while the other covariates are not. The regression is significant, and explains 13% of the variation in the data.

We fit a new model with the covariate `education`, and remove the intercept. When we remove the intercept, we ensure that the regression line passes through the origin (if education is 0, then the wages will also be zero, in this case).

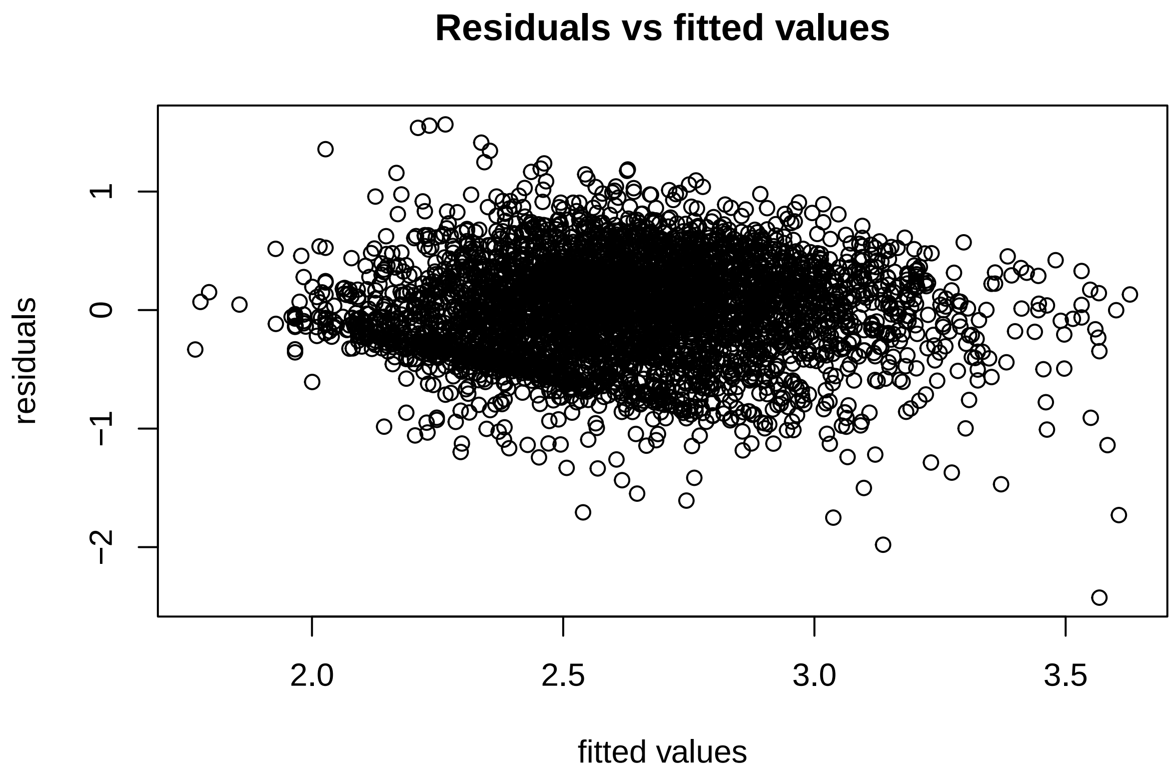
```
model4c <- mylm(wages ~ education - 1, data = SLID)
summary.mylm(model4c)
```

```
## Summary of object
## [1] "Coefficients:"
## education
## 1.146697
## [1] "Std.errors"
##   education
## 0.008766101
## [1] "z-values"
##               [,1]
## education 130.8104
## [1] "p-values"
##           [,1]
## education    0
## Chi-test on  0  df:  318.0323  with p-value:  0
## R^2:  0.07389171
```

The only covariate education is significant, with a z-statistic of 130.8 and p-value approximately 0. The regression is significant from the χ^2-test.

The residual plots in the three models above all point toward heteroscedasticity, which violates our assumption of constant variance. A common way of handling this is with a transformation of the response. In our case, taking the log of the response seems to improve the plots, which we show by transforming the first of our three models:

```
model4a_transformed <- mylm(I(log(wages)) ~ sex + language + age + I(education^2),
    data = SLID, contrasts = list(language = "contr.treatment"))
plot.mylm(model4a_transformed)
```



The code for `mylm` is found below.

```
# 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")

    # Code to calculate coefficients, residuals, fitted values,
    # etc...
    n = dim(X)[1]
    p = dim(X)[2]
    XTX_inv = solve(t(X) %*% X)
    beta = XTX_inv %*% t(X) %*% y  #Coefficient estimates; least squares
    SSE = t(y - X %*% beta) %*% (y - X %*% beta)
    biased_estimator = as.numeric(SSE/n)
    REML_estimator = as.numeric(SSE/(n - p))  #Unbiased
    beta_cov = biased_estimator * XTX_inv  #Covariance matrix of beta-hat
    std_errors = sqrt(diag(beta_cov))

    # Hypothesis testing for coefficients
    z_value = beta/std_errors  #Observed z-values under H_0, which are standard normally dist.
```

```r
    p_value = 2 * pnorm(abs(z_value), lower.tail = FALSE)  #z-test

    # Residuals
    H = X %*% XTX_inv %*% t(X)
    y_hat = H %*% y
    residual = y - y_hat

    # SST, SSR
    ones = rep(1, n)
    SSR = as.numeric(t(y) %*% (H - (1/n) * ones %*% t(ones)) %*% y)
    SST = SSR + SSE
    R2 = SSR/SST

    # Testing significance of regression with chi-square test
    k = length(beta) - 1
    chi_obs = (SST - SSE)/(SSE/(n - p))  #Approx. chi-square distributed (normalised)
    p_value_chi = pchisq(chi_obs, df = k, lower.tail = FALSE)

    # and store the results in the list est
    est <- list(terms = terms, model = mf)
    est$coeffs <- beta
    est$beta.matrix <- beta_cov
    est$std.errors <- std_errors
    est$z.values <- z_value
    est$p.values <- p_value
    est$residuals <- residual
    est$y.hat <- y_hat
    est$SSE <- SSE
    est$SST <- SST
    est$SSR <- SSR
    est$df <- n - p
    est$chi.value <- chi_obs
    est$p.value.chi <- p_value_chi
    est$R2 <- R2
    est$k <- k
    est$colnames <- colnames(X)

    # 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
```

```r
    cat("Info about object\n")
    print("Coefficients:")
    cat(object$colnames, "\n")
    cat(object$coeffs, "\n")
}

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("Summary of object\n")
    print("Coefficients:")
    cat(object$colnames, "\n")
    cat(object$coeffs, "\n")
    print("Std.errors")
    print(object$std.errors)
    print("z-values")
    print(object$z.values)
    print("p-values")
    print(object$p.values)
    cat("Chi-test on ", object$k, " df: ", object$chi.value, " with p-value: ",
        object$p.value.chi)
    cat("\nR^2: ", object$R2)

}

plot.mylm <- function(object, ...) {
    # Code here is used when plot(object) is used on objects of
    # class 'mylm'
    plot(object$y.hat, object$residuals, main = "Residuals vs fitted values",
        xlab = "fitted values", ylab = "residuals")
}

# 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 = "+")
        }
```

```r
        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)
}
```