TMA4315 Generalized Linear Models

Compulsory exercise 1: Linear models for Gaussian data

Håvard Fossdal* Rasmus Grødeland† Yawar Mahmood‡

13 September, 2024

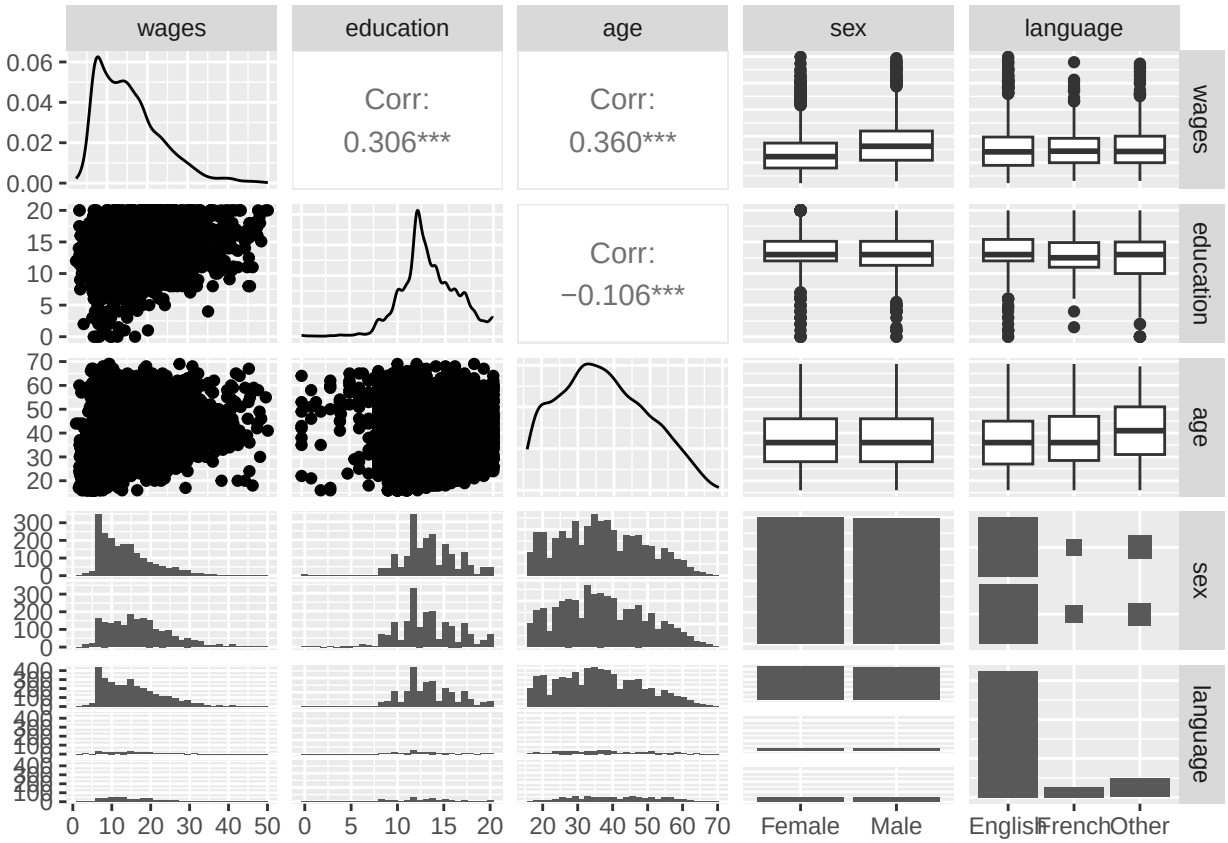
Explanatory analysis of the dataset

In this exercise, we will analyse a dataset from Canada consisting of 3987 observations. The observed variables are given in the table below.

Variable	Description
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

After removing the observations that do not contain all the variables above, we are left with a complete dataset whose diagnostic plots are given below.

*Group 2, Department of Mathematical Sciences, haafo@stud.ntnu.no
†Group 2, Department of Mathematical Sciences, rasmug@stud.ntnu.no
‡Group 2, Department of Mathematical Sciences, yawarm@stud.ntnu.no



We observe from the diagnostic plots, that there is a positive correlation between the wages and the level of education and age of the people in question. This seems reasonable, as one would expect that higher education and/or seniority would have a positive impact on the wages you are getting.
Furthermore, the correlation between education and age is negative, implying that younger people are on average spending a longer time in the schooling system, as compared to the older citizens. This might be the result of it becoming easier for people of all socioeconomic statuses to attend higher education, as a result of recent changes in scholarships and federal subsidies.
We observe that the language spoken by, and the age of the people in the dataset, is distributed equally among the genders. Although the wages seem to be higher for the male citizens.

To perform a multiple linear regression analysis on the dataset, we would as the name suggests, have to assume that there is a linear relationship between the response variable and the covariate effects. Furthermore, we need to assume that the errors are homoscedastic and uncorrelated with mean 0.

Linear regression with the `mylm` package

One of the tasks we were given as part of this exercise, was to develop an R-package for fitting linear models, similar to the built-in `lm`-function. We were then instructed to test our package by fitting a simple linear regression model to the dataset described above, using *wages* as the response variable and *education* as the covariate.

Estimation of model coefficients

From earlier courses, we know that the maximum likelihood (ML) estimator for the regression coefficients are given by,

$$\hat{\beta} = (X^T X)^{-1} X^T y,$$

where X is the design matrix and y the vector of responses. By coding this into `mylm`, the regression coefficients of the model we are testing may be estimated by the following simple script.

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

```
##
## Call:
## mylm(formula = wages ~ education, data = SLID)
##
## Coefficients:
##      (Intercept) education
## [1,]      4.9717   0.79231
```

We may test that the computed values are correct, by writing a similar script using the built-in `lm`-function.

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

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

We conclude, based on the R-output above, that `mylm` computes the estimates correctly. At least for the simple linear model that we are testing.

Another nice property of the `lm`-function, which we would like to incorporate in the `mylm`-package, is a way to print a summary of the model fit to the terminal. This requires some understanding of how one might go about computing the different statistics which the `lm`-function uses to summarize the model fit.

Estimation of the standard error of the coefficient estimates

From earlier courses, we know that the ML estimator $\hat{\beta}$ has covariance matrix,

$$\mathrm{Cov}(\hat{\beta}) = \sigma^2 (X^T X)^{-1},$$

where σ^2 is the unknown model variance.

Since the model variance is unknown, we need to estimate it. In our case, we choose to estimate the model variance using the restricted maximum likelihood (REML) estimator, which takes the form,

$$\hat{\sigma}^2 = \frac{1}{n-p} \left(y - X\hat{\beta}\right)^T \left(y - X\hat{\beta}\right) =: \frac{1}{n-p} \varepsilon^T \varepsilon,$$

where n is the number of observations, p the number of regression coefficients and ε the residuals of the model.

Inserting this expression in the expression for the REML estimator, we get that

$$\widehat{\mathrm{Cov}}(\hat{\beta}) = \frac{1}{n-p} \left(y - X\hat{\beta}\right)^T \left(y - X\hat{\beta}\right) (X^T X)^{-1}.$$

Which we will use to estimate the standard error of the ML estimator $\hat{\beta}$. This is done by extracting the diagonal elements of $\widehat{\mathrm{Cov}}(\hat{\beta})$, corresponding to the estimated variances of the components of $\hat{\beta}$, and then taking the square root of the resulting variances.

We are left with a total of p standard error estimates $\left\{\hat{\sigma}_{\beta_i} : i = 1, \ldots, p\right\}$, corresponding to each of the p components of $\hat{\beta}$. We can now carry out a significance test for each component.

Significance test for coefficient estimates

Both $\hat{\beta}_i$ and $\hat{\sigma}_{\beta_i}$ are unbiased, so we get that

$$\frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}_{\beta_i}} \sim \mathrm{t}_{n-p},$$

is Student's t-distributed with $n - p$ degrees of freedom. Assuming that $n - p$ is sufficiently large for the Central Limit Theorem to be applicable, we get that

$$\frac{\hat{\beta}_i - \beta_i}{\hat{\sigma}_{\beta_i}} \overset{a}{\sim} \mathrm{N}(0, 1),$$

is approximately standard normal.

Under the significance test, we assume that the regressor x_i is of no explanatory value, i.e. that β_i equates to zero. We are then interested in finding the probability of obtaining observations at least as extreme as the obtained observations, conditional on the regressor having no explanatory power. If the resulting probability is low, our observations would indicate that the initial assumption of x_i having no explanatory power is likely false, thus making the regressor significant.

From the approximately normal test statistic, we deduce that the probability of obtaining observations at least as extreme as the ones we made, conditional on x_i having no explanatory power, is

$$2 \cdot \Phi\left(-\left|\frac{\hat{\beta}_i}{\hat{\sigma}_{\beta_i}}\right|\right),$$

where Φ is the cumulative distribution function of the standard normal. This follows from the fact that the normal distribution is two-tailed. We further denote the quantity $\hat{\beta}_i / \hat{\sigma}_{\beta_i}$ as the *z-value* of component $\hat{\beta}_i$.

Analysis of variance

When analyzing the variance of a linear model, one typically considers the model's sum of squares. In particular, the residual sum of squares (SSE) and the total sum of squares (SST). The SST is a measurement of the deviation in the dataset, defined as the sum of the squared deviation of the responses from their mean, i.e.

$$\mathrm{SST} := \sum_i \left(y_i - \bar{y}\right)^2.$$

The SSE, sometimes referred to as the sum of squared errors, is defined as the squared deviation of of the responses from the regressed estimate, i.e.

$$\mathrm{SSE} := \sum_i \left(y_i - \hat{y}_i\right)^2 = \varepsilon^T \varepsilon,$$

ε being the residuals of the model.

Using the sum of squares, we can compute the proportion of the variance in the dataset described by the regression model, known as the coefficient of determination,

$$R^2 := 1 - \frac{\mathrm{SSE}}{\mathrm{SST}} = 1 - \frac{\varepsilon^T \varepsilon}{\sum_i \left(y_i - \bar{y}\right)^2}.$$

For each additional regressor one chooses to include in the model, R^2 only increases. Thus, when comparing models with different numbers of regressors, it is sometimes beneficial to scale R^2 by the model's degrees of freedom. This gives rise to the adjusted coefficient of determination, defined as

$$R^2_{\text{adj}} := 1 - \frac{n-1}{n-p}\left(1 - R^2\right),$$

where n is the number of observations and p the number of regression coefficients. We call $n-p$ the residual degrees of freedom, whilst $p-1$ is the degrees of freedom for the model. If we divide the residual sum of squares by the residual degrees of freedom and compute the square root of the result, we obtain the residual standard error.

Test of sigificance of regression

By defining the regression sum of squares (SSR) as the difference between SST and SSE, it can be shown that

$$\frac{\text{SSR}/p-1}{\text{SSE}/n-p} = \frac{n-p}{p-1}\frac{\text{SSR}}{\text{SSE}} \sim F_{p-1,n-p}$$

is Fisher distributed with $p-1$ and $n-p$ degrees of freedom. Assuming that the residual degrees of freedom are sufficiently large, we can use the connection between the above F-statistic and the Wald test, together with the asymptotic properties of the Wald test, to find that

$$(n-p)\frac{\text{SSR}}{\text{SSE}} \overset{a}{\sim} \chi^2_{p-1}$$

is approximately χ^2-distributed with $p-1$ degrees of freedom. From this, we find that the p-value for the above χ^2-statistic is given by,

$$p\text{-value} = 1 - F\left((n-p)\frac{\text{SSR}}{\text{SSE}}\right),$$

where F is the cumulative distribution function of a χ^2-distribution with $p-1$ degrees of freedom.

In a significance of regression test, the p-value serves the same purpose as the probability computed as part of the significance test for the coefficient estimates; A low p-value makes it likely that the regression is significant.

Summarizing the model fit using using the `mylm`-package

By implementing the above functionality in the `mylm`-package, we are able to make a summary of the model fit, similarly to what we are able to do using the built-in *lm*-function. For the simple linear model, the summary is as follows:

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

```
##
## Call:
## mylm(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 Z value Pr(>|Z|)
```

```
## (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
## Chi-squared-statistic: 411.4 on 1 DF,  p-value: < 2.2e-16
```

We observe from the print-out, that the estimates for the *intercept* and *education* coefficient is 4.97169 and 0.79231 respectively, with standard errors 0.53429 and 0.03906. Furthermore, the Z-test implies that the *education* coefficient is significant at a significance-level of 0.001 and higher.

We are able to draw the following conclusions based on the coefficient estimates,

- Education has a positive effect on people's wages; We expect an increase in the composite hourly wage rate by 0.79231 for every additional year spent in the education system.
- Even if you have no education at all, your composite wage rate is still positive. Meaning that people with no education are still being paid to do their jobs.

We also observe that $R^2 = 0.09359$ for this model, meaning that the model explains approximately 9 % of the variance in the dataset. Thus, people's level of education is not enough to explain the different composite wage rates in their entirety.

Using the ANOVA functionality of the `mylm`-package, we are also able to create the following table:

```
anova.mylm(model1)
```

```
## Analysis of Variance Table
##
## Response: wages
##           Df   Sum Sq Mean Sq Chi^2 value Pr(>Chi^2)
## education 1    23096  23096.2 411.4471    < 2e-16    ***
## Residuals 3985 223694 56.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We observe that the SSE for the linear model is 223 694 on 3 985 degrees of freedom. Since the SST equates to the sum of the SSE and the SSR, we can easily determine its value by adding the SSR for *education* to the model's SSE,

$$\text{SST} = 23\ 096 + 223\ 694 = 246\ 790.$$

Using the χ^2-test described earlier in the report, we observe from the ANOVA-table above, that the regression is indeed significant using a significance level of 0.001 or higher. This is to be expected, since we are regressing on only one covariate and the significance test for the covariate in question shows that the covariate is significant. The χ^2- and Z-statistics thus serve the same purpose in the case of simple linear regression.

By implementing a simple function in the `mylm`-package that uses the built-in *qnorm*- and *qchisq*-functions to compute critical values, we find that the statistical tests have the following critical values for a significance level of $\alpha = 0.001$:

```
critical.mylm(model1, alpha = 0.001)
```

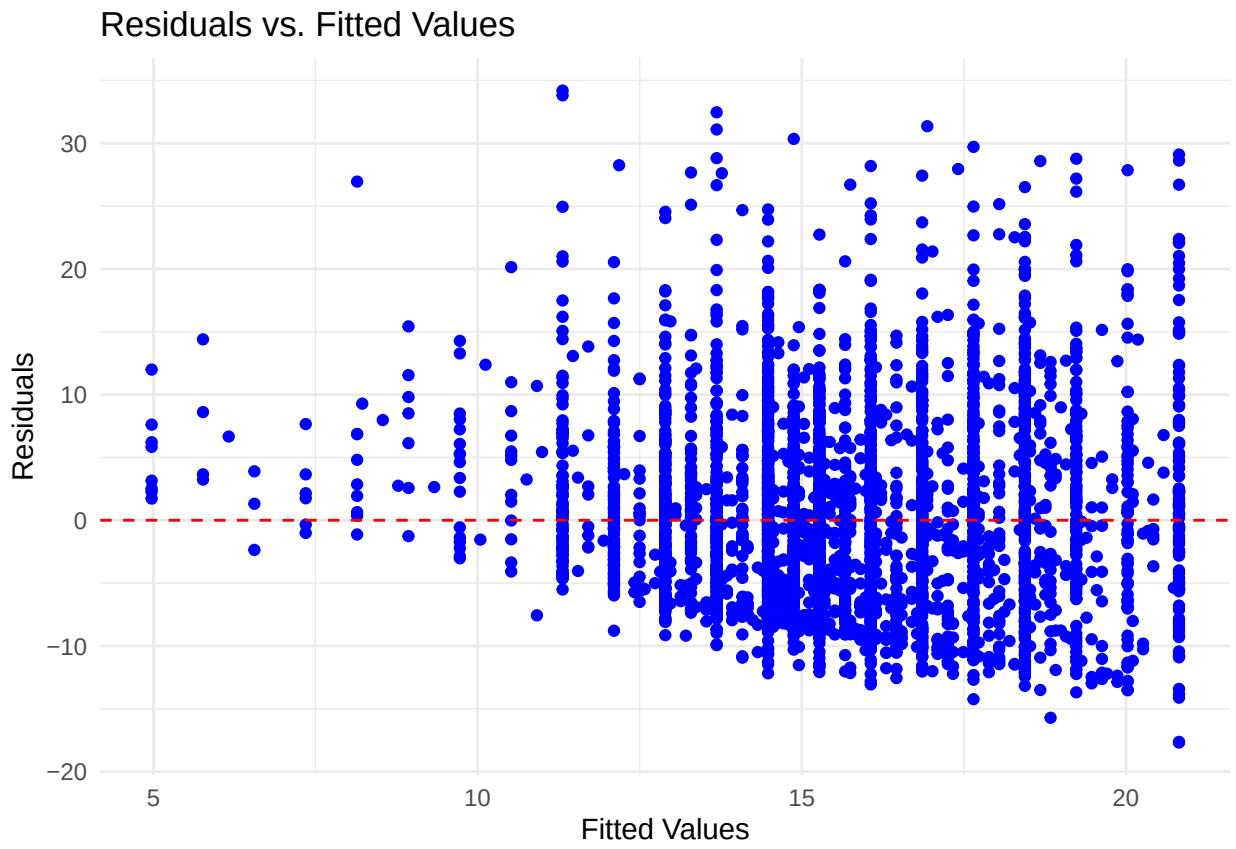
```
## The critical values for the Z-test and the Chi-squared-test are
## 3.29053 and 10.82757 respectively.
```

Residual vs. fitted values

Yet another invaluable feature of the *lm*-function is the ability to plot the residuals vs. the fitted values. This is especially useful when checking that the model assumptions are satisfied, and as part of this exercise, we have therefore implemented such functionality in the `mylm`-package.

Plotting the residuals vs. fitted values for the simple regression model, we get the following figure:

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



We observe from the plot that the residuals form a 'cone-like' shape when plotted against the fitted values. This is typical for heteroscedastic datasets, in which the variance is not constant over different observations. We clearly observe that the residuals tend to be larger for the larger fitted values, which contradict the model assumption of homoscedastic errors. We can not conclude that the composite hourly wage rate does not depend on the level of education. However, the residual plot indicates that the model is likely misspecified, possibly due to the small number of regressors.

Multiple linear regression with the `mylm`-package

We extend the capabilities of the `mylm` function to handle multiple linear regression. The model we are going to look at is the dependent variable (in this case, `wages`) and two (or more) independent variables (here, `education` and `age`).

Below, a visualization of the residuals against the fitted values is also given, which helps us assess the assumptions of the linear regression model, such as homoscedasticity and the independence of errors. In this particular case, it helps us to determine how well the predictors (education and age) explain the variation in wages - in general, how well the predictors explain the dependent variable.

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

```
##
## Call:
## mylm(formula = wages ~ education + age, data = SLID)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.303  -4.495  -0.807   3.674  37.628
##
## Coefficients:
##             Estimate Std. Error Z value Pr(>|Z|)
## (Intercept) -6.02165 0.61892    -9.729  <2e-16   ***
## education    0.90146  0.03576    25.209  <2e-16   ***
## age          0.25709  0.00895    28.721  <2e-16   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.820 on 3984 degrees of freedom
## Multiple R-squared:  0.24907,   Adjusted R_squared:  0.24869
## Chi-squared-statistic: 1321.4 on 2 DF,  p-value: < 2.2e-16
```

```r
model2_lm <- lm(wages ~ education + age, data = SLID)
summary(model2_lm)
```

```
##
## Call:
## lm(formula = wages ~ education + age, data = SLID)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.303  -4.495  -0.807   3.674  37.628
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.021653   0.618924  -9.729   <2e-16 ***
## education    0.901464   0.035760  25.209   <2e-16 ***
## age          0.257090   0.008951  28.721   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.82 on 3984 degrees of freedom
## Multiple R-squared:  0.2491, Adjusted R-squared:  0.2487
## F-statistic: 660.7 on 2 and 3984 DF,  p-value: < 2.2e-16
```

We now consider the results of our analysis:

- **Intercept**:
  The intercept estimate is `-6.02165` with a standard error of `0.61892`. This estimate indicates that, hypothetically, the expected wage when both education and age are 0 would be around `-6.02`. This value does not (most probably) hold any meaning in practice, as it is unrealistic to have someone with

zero years of education or age. However, the intercept is still an important parameter because it serves as the baseline level from which the effects of the predictors (education and age) are measured. In other words, it helps to shift the regression line so that it better fits the observed data, even if its direct interpretation is not meaningful in real-world scenarios. Looking at the `Z-value` and `p-value`, it is significant.

- **Education Coefficient**:
  The coefficient estimate for education is `0.90146` with a standard error of `0.03576`. This result implies that for each additional year of education, wages are expected to increase by approximately `0.90` units, holding age constant. The Z-value of `25.209` and the p-value of `< 2e-16` suggest that education is a highly significant predictor.

- **Age Coefficient**:
  The coefficient estimate for age is `0.25709` with a standard error of `0.00895`. This result implies that for each additional year in age, wages are expected to increase by approximately `0.257` units, holding education constant. The Z-value of `28.721` and the p-value of `< 2e-16` suggest that age is also a highly significant predictor of wages.

In practical terms, the significance of education and age in predicting wages means that both factors play a crucial role in determining income. Higher education leads to better-paying jobs, while more years of experience in an industry also results in higher wages. However, education has a stronger influence on wage growth compared to experience alone. This suggests that staying in a field for many years without obtaining formal higher education can lead to missed opportunities for potential wage increases. The optimal combination, of course, is having both a high level of education and extensive experience in the industry.

A valid question to ask could be, how does the results from a multiple regression model—where both education and age are used as predictors of wages—differ from two separate simple linear regression models, where each predictor (education and age) is analyzed individually?

The goal is to understand the relationship between wages, education, and age, and how these relationships might change when we consider them together versus separately.

```r
model_education <- mylm(wages ~ education, data = SLID)
summary.mylm(model_education)
```

```
##
## Call:
## mylm(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 Z value Pr(>|Z|)
## (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
## Chi-squared-statistic: 411.4 on 1 DF,  p-value: < 2.2e-16
```

```r
model_age <- mylm(wages ~ age, data = SLID)
summary.mylm(model_age)
```

```
##
## Call:
## mylm(formula = wages ~ age, data = SLID)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.747  -4.847  -1.507   3.914  35.063
##
## Coefficients:
##             Estimate Std. Error Z value Pr(>|Z|)
## (Intercept) 6.8909   0.37405    18.423  <2e-16   ***
## age         0.23311  0.00958    24.325  <2e-16   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.343 on 3985 degrees of freedom
## Multiple R-squared:  0.12929,   Adjusted R_squared:  0.12907
## Chi-squared-statistic: 591.7 on 1 DF,  p-value: < 2.2e-16
```

When we perform a simple linear regression with each predictor independently (education or age), we are not accounting for the influence of the other variable. In contrast, in multiple linear regression, the effect of each predictor is adjusted for the influence of the other. This adjustment often results in different parameter estimates because the model takes into account how the predictors interact or covary with each other.

In the dataset, education and age have a statistically significant negative correlation (-0.106). This implies that younger individuals tend to have higher levels of education, while older individuals tend to have lower levels of education. This trend likely reflects societal changes over time, such as increased access to higher education for younger generations.

Simple Linear Regressions:
When regressing wages on education alone, the model estimates a coefficient of 0.79231 for education. This suggests that for each additional year of education, wages increase by approximately 0.79 units. However, this model does not account for the influence of age. Similarly, when regressing wages on age alone, the model estimates a coefficient of 0.23311 for age, implying that each additional year of age increases wages by approximately 0.23 units. In these simple models, each predictor is treated in isolation, which can lead to over- or under-estimation of their effects because we are not considering how education and age covary.

Multiple Linear Regression:
In the multiple regression model, where both education and age are included as predictors, the coefficient for education increases to 0.90146, and the coefficient for age increases to 0.25709. This adjustment suggests that when we account for the correlation between education and age, the effect of both predictors on wages becomes more accurate. The increase in both coefficients indicates that the simple models were underestimating the true impact of each predictor when the other was not accounted for. In particular, the effect of education is more substantial when we adjust for age.

Looking at the adjusted R^2, we can also see that adding both covariates in the model, increases the adjusted R^2 significantly, and hence, it is justifiable to have both covariates present in the model.

Simple Regression with Education: The R-squared value for the simple regression with education is 0.09359, indicating that education alone explains about 9.36% of the variance in wages. The adjusted R-squared is almost the same at 0.09336, suggesting that the model is relatively stable.

Simple Regression with Age: The R-squared value for the simple regression with age is slightly higher at 0.12929, indicating that age explains about 12.93% of the variance in wages. The adjusted R-squared is 0.12907, again showing a stable model.

Multiple Regression with Education and Age: In the multiple regression model, the R-squared value increases to 0.24907, meaning that both education and age together explain about 24.91% of the variance in wages. The adjusted R-squared is 0.24869, which confirms that adding both covariates improves the explanatory power of the model.

Further testing of the `mylm`-package

We will now proceed with further testing of the `mylm` package by fitting three different models:

```
model1_mylm <- mylm(wages ~ sex + age + language + I(education^2), data = SLID)
summary.mylm(model1_mylm)
```

```
##
## Call:
## mylm(formula = wages ~ sex + age + language + I(education^2),
##     data = SLID)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.1712  -4.2762  -0.7631   3.2176  35.9289
##
## Coefficients:
##                Estimate Std. Error Z value Pr(>|Z|)
## (Intercept)    -1.87553 0.44035    -4.259  2.05e-05 ***
## sexMale        3.4087   0.20842    16.355  <2e-16   ***
## age            0.24862  0.00866    28.701  <2e-16   ***
## languageFrench -0.07553 0.42514    -0.178  0.859
## languageOther  -0.13454 0.32315    -0.416  0.677
## I(education^2) 0.03482  0.00129    26.991  <2e-16   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.577 on 3981 degrees of freedom
## Multiple R-squared:  0.30222,   Adjusted R_squared:  0.30134
## Chi-squared-statistic: 1724.2 on 5 DF,  p-value: < 2.2e-16
```

```
model1_lm <- lm(wages ~ sex + age + language + I(education^2), data = SLID)
summary(model1_lm)
```

```
##
## Call:
## lm(formula = wages ~ sex + age + language + I(education^2), data = SLID)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -27.171  -4.276  -0.763   3.218  35.929
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.875531   0.440345  -4.259  2.1e-05 ***
## sexMale         3.408700   0.208420  16.355  < 2e-16 ***
## age             0.248625   0.008663  28.701  < 2e-16 ***
## languageFrench -0.075532   0.425136  -0.178    0.859
## languageOther  -0.134540   0.323153  -0.416    0.677
## I(education^2)  0.034815   0.001290  26.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.577 on 3981 degrees of freedom
## Multiple R-squared:  0.3022, Adjusted R-squared:  0.3013
## F-statistic: 344.8 on 5 and 3981 DF,  p-value: < 2.2e-16
```

```
model2_mylm <- mylm(wages ~ language * age, data = SLID)
summary.mylm(model2_mylm)
```

```
##
## Call:
## mylm(formula = wages ~ language * age, data = SLID)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.751  -4.832  -1.412   3.938  35.187
##
## Coefficients:
##                     Estimate Std. Error Z value Pr(>|Z|)
## (Intercept)         6.55579  0.41068    15.963  <2e-16   ***
## languageFrench      2.86063  1.59607    1.792   0.0731   .
## languageOther       0.84862  1.23518    0.687   0.492
## age                 0.24485  0.01069    22.91   <2e-16   ***
## languageFrench:age -0.08393 0.04046    -2.075  0.038    *
## languageOther:age  -0.03701 0.02934    -1.262  0.207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.339 on 3981 degrees of freedom
## Multiple R-squared:  0.13117,   Adjusted R_squared:  0.13008
## Chi-squared-statistic: 601.0 on 5 DF,  p-value: < 2.2e-16
```

```
model2_lm <- lm(wages ~ language * age, data = SLID)
summary(model2_lm)
```

```
##
## Call:
## lm(formula = wages ~ language * age, data = SLID)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.751  -4.832  -1.412   3.938  35.187
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)         6.55579    0.41068  15.963   <2e-16 ***
## languageFrench      2.86063    1.59607   1.792   0.0732 .
## languageOther       0.84862    1.23518   0.687   0.4921
## age                 0.24485    0.01069  22.910   <2e-16 ***
## languageFrench:age -0.08393    0.04046  -2.075   0.0381 *
## languageOther:age  -0.03701    0.02934  -1.262   0.2072
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.339 on 3981 degrees of freedom
## Multiple R-squared:  0.1312, Adjusted R-squared:  0.1301
## F-statistic: 120.2 on 5 and 3981 DF,  p-value: < 2.2e-16
```

```
model3_mylm <- mylm(wages ~ education - 1, data = SLID)
summary.mylm(model3_mylm)
```

```
##
## Call:
## mylm(formula = wages ~ education - 1, data = SLID)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.8039  -5.3421  -0.6624   4.4646  36.3264
##
## Coefficients:
##           Estimate Std. Error Z value Pr(>|Z|)
## education 1.1467   0.00877    130.794 <2e-16   ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.572 on 3986 degrees of freedom
## Multiple R-squared:  0.07389,   Adjusted R_squared:  0.07389
## Chi-squared-statistic: 318.0 on 0 DF,  p-value: < 2.2e-16
```

```
model3_lm <- lm(wages ~ education - 1, data = SLID)
summary(model3_lm)
```

```
##
## Call:
## lm(formula = wages ~ education - 1, data = SLID)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.804  -5.342  -0.662   4.465  36.326
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## education 1.146697   0.008767   130.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.572 on 3986 degrees of freedom
## Multiple R-squared:  0.811,  Adjusted R-squared:  0.811
## F-statistic: 1.711e+04 on 1 and 3986 DF,  p-value: < 2.2e-16
```

Comparison for Model 1:

The results between the `mylm` and `lm` models are almost identical: - **Sex (Male)**: Both models estimate a positive effect of around 3.41 on wages for males compared to females, with a highly significant p-value ($p < 2e\text{-}16$). - **Age**: Both models provide the same estimate of 0.24862, showing a highly significant positive relationship between age and wages. - **Language**: Neither language categories ("French" and "Other") are significant in either model. - **Education^2**: Both models show that education^2 has a highly significant and positive effect on wages (Estimate: 0.03482), suggesting that wages grow more substantially with higher levels of education.

The **R-squared** values for both models are also identical (Multiple R-squared: 0.3022), meaning that both models explain about 30% of the variance in wages. This indicates that the `mylm` package produces results consistent with the standard `lm` function.

Comparison for Model 2:

Again, the results between `mylm` and `lm` are very similar: - **Age**: Both models show a significant positive relationship between age and wages (Estimate: 0.24485, $p < 2e\text{-}16$). - **Language (French)**: Both models show a borderline significant positive effect for French (p: 0.073), but not highly significant. - **Interaction (French:Age)**: The interaction term for French and age is significant in both models (Estimate: -0.08393, p: 0.038), indicating that the positive effect of age on wages decreases for French speakers. - **Residual standard error**: Both models have the same residual standard error of 7.339.

The **R-squared** values are also the same (0.1312), meaning that both models explain about 13% of the variance in wages. This again demonstrates that the `mylm` package performs similarly to `lm`.

Comparison for Model 3:

Both models yield nearly identical results: - **Education**: Both models provide a highly significant estimate for education (Estimate: 1.1467, $p < 2e\text{-}16$), indicating a strong positive relationship between education and wages when no intercept is included. - **Residual standard error**: The residual standard error is also the same (7.572), indicating that the models fit similarly. - **R-squared**: Although R-squared is not directly comparable for models without an intercept, the diagnostics and significance of education show consistent results between the two models.

Summary of Comparisons:

In all three models, the results from the `mylm` function are practically identical to those from the standard `lm` function. The coefficient estimates, p-values, and residual errors are all very similar across the models. This demonstrates that the `mylm` package is performing as expected and can be used interchangeably with `lm` for linear regression modeling. The small differences in precision between the models are within acceptable limits and do not impact the overall conclusions. Thus, the `mylm` package provides reliable estimates consistent with traditional methods.

Source code from `mylm.R`:

```r
library("ggplot2")

mylm <- function(formula, data = list(), contrasts = NULL, ...){
  # extract model matrix & responses
  mf <- model.frame(formula = formula, data = data)
```

```r
X  <- model.matrix(attr(mf, "terms"), data = mf, contrasts.arg = contrasts)
y  <- model.response(mf)
terms <- attr(mf, "terms")

# transpose of the covariate matrix
X_transpose <- t(X)

# ML estimator of the regression coefficients
coefficients <- solve(X_transpose %*% X) %*% X_transpose %*% y

# vector of residuals
residuals <- y - X %*% coefficients

# REML estimator for the model variance
n <- dim(X)[1]
p <- dim(coefficients)[1]
variance <- 1/(n - p) * (t(residuals) %*% residuals)

# covariance matrix
covariance <- variance[1,1] * solve(X_transpose %*% X)

# vector of standard errors
std_error <- sqrt(diag(covariance))

# vector of z values
z_value <- coefficients / std_error

# vector of p values
p_value <- 2 * pnorm(-abs(z_value))

# multiple R-squared
sst <- sum((y - mean(y))^2)
sse <- sum(residuals^2)
r_squared <- 1 - (sse / sst)

# adjusted R-squared
df_residual <- n - p
df_model <- p - 1
df_total <- n - 1
r_squared_adj <- 1 - (df_total/df_residual)*(1 - r_squared)

# residual standard error
residual_std_error <- sqrt(sse / df_residual)

# chi^2-statistic - (scaled with F-statistic)
ssr <- sst - sse
chi_sq <- (ssr / (sse / df_residual))
p_value_chi <- 1 - pchisq(chi_sq, df_model)

# summary matrix
coefficients_summary <- round(coefficients, digits = 5)
std_error_summary <- round(std_error, digits = 5)
z_value_summary <- round(z_value, digits = 3)
```

```r
p_value_summary <- signif(p_value, digits = 3)
p_value_summary[p_value_summary < 2e-16] <- '<2e-16'

significance_marks <- p_value
for (i in 1:dim(p_value)[1]) {
  if (p_value[i] < 0.001) {
    significance_marks[i] <- '***'
  } else if (p_value[i] < 0.01) {
    significance_marks[i] <- '**'
  } else if (p_value[i] < 0.05) {
    significance_marks[i] <- '*'
  } else if (p_value[i] < 0.1) {
    significance_marks[i] <- '.'
  } else {
    significance_marks[i] <- ' '
  }
}
summary_matrix <- cbind(
  coefficients_summary,
  std_error_summary,
  z_value_summary,
  p_value_summary,
  significance_marks
)
colnames(summary_matrix) <- c(
  'Estimate',
  'Std. Error',
  'Z value',
  'Pr(>|Z|)',
  ''
)

# residual quantiles
residual_quantiles <- quantile(residuals)
names(residual_quantiles) <- c('Min', '1Q', 'Median', '3Q', 'Max')

# create list
est <- list(terms = terms, model = mf)

# Store call and formula used
est$call <- match.call()
est$formula <- formula
est$coefficients <- coefficients
est$residuals <- residuals
est$summary_matrix <- summary_matrix
est$residual_quantiles <- residual_quantiles
est$residual_std_error <- residual_std_error
est$r_squared <- r_squared
est$r_squared_adj <- r_squared_adj
est$chi_sq <- chi_sq
est$p_value_chi <- p_value_chi
est$df_residual <- df_residual
est$df_total <- df_total
```

```
  est$df_model <- df_model
  est$sse <- sse
  est$ssr <- ssr
  est$sst <- sst

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

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

print.mylm <- function(object, ...){
  cat('\nCall:\n')
  print(object$call)
  cat('\nCoefficients:\n')
  print(t(object$coefficients), digits = 5)
}

summary.mylm <- function(object, ...){
  cat('\nCall:\n')
  print(object$call)
  cat('\nResiduals:\n')
  print(object$residual_quantiles, digits = 4)
  cat('\nCoefficients:\n')
  print(object$summary_matrix, quote=FALSE)
  cat('---\n')
  cat("Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1\n\n")
  cat(sprintf('Residual standard error: %.3f on %d degrees of freedom\n',
              object$residual_std_error, object$df_residual))
  cat(sprintf('Multiple R-squared:  %.5f,   Adjusted R_squared:  %.5f\n',
              object$r_squared, object$r_squared_adj))
  if (object$p_value_chi < 2.2e-16) {
    cat(sprintf('Chi-squared-statistic: %.1f on %d DF,  p-value: < 2.2e-16',
                object$chi_sq, object$df_model))
  } else {
    cat(sprintf('Chi-squared-statistic: %.1f on %d DF,  p-value: %3e\n',
                object$chi_sq, object$df_model, object$p_value_chil))
  }
}

plot.mylm <- function(object, ...){

  fitted_values <- object$model[[1]] - object$residuals

  plot_data <- data.frame(Fitted = fitted_values, Residuals = object$residuals)

  plot <- ggplot(plot_data, aes(x = Fitted, y = Residuals)) +
    geom_point(color = "blue") +
    geom_hline(yintercept = 0, linetype = "dashed", color = "red") +
    labs(title = "Residuals vs. Fitted Values", x = "Fitted Values",
         y = "Residuals") +
    theme_minimal()
```

```
  print(plot)
}

critical.mylm <- function(object, alpha, ...) {
  # critical values (task d)
  critical_value_chi2 <- qchisq(1 - alpha, object$df_model)
  critical_value_z <- qnorm(1 - alpha/2)
  cat(sprintf('The critical values for the Z-test and the Chi-squared-test are
%.5f and %.5f respectively.\n', critical_value_z, critical_value_chi2))
}

anova.mylm <- function(object, ...){

  # 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]] <- mylm(formula = formula, data = object$model)
  }

  # initialize vectors holding degrees of freedom and sum of squares
  df <- vector(length = length(comp) + 1)
  SS <- vector(length = length(comp) + 1)
  # compute degrees of freedom and sum of squares iteratively
  for(numComp in 1:length(comp)){
    df[numComp+1] <- model[[numComp]]$df_model - sum(df)
    SS[numComp+1] <- model[[numComp]]$ssr - sum(SS)
  }

  # residual degrees of freedom
  df[1] <- model[[length(comp)]]$df_residual

  # residual sum of squares
  SS[1] <- model[[length(comp)]]$sse

  # rearrange vectors
  df <- c(df[-1], df[1]) ; SS <- c(SS[-1], SS[1])

  # compute mean sum of squares
  MSS <- SS / df
```

```r
# compute test statistic
chi_sq <- SS / MSS[length(comp) + 1]

# compute p-values
p_value_chi <- vector(length = length(comp))
for(numComp in 1:length(comp)){
  p_value_chi[numComp] <- 1 - pchisq(chi_sq[numComp], df[numComp])
}

# summary matrix
df_summary <- df
SS_summary <- round(SS)
MSS_summary <- round(MSS, digits = 1)
chi_sq_summary <- round(chi_sq, digits = 4)
p_value_summary <- signif(p_value_chi, digits = 4)
p_value_summary[p_value_summary < 2e-16] <- '< 2e-16'

significance_marks <- p_value_chi
for (i in 1:length(p_value_chi)) {
  if (p_value_chi[i] < 0.001) {
    significance_marks[i] <- '***'
  } else if (p_value_chi[i] < 0.01) {
    significance_marks[i] <- '**'
  } else if (p_value_chi[i] < 0.05) {
    significance_marks[i] <- '*'
  } else if (p_value_chi[i] < 0.1) {
    significance_marks[i] <- '.'
  } else {
    significance_marks[i] <- ' '
  }
}

# remove values corresponding to the residuals
chi_sq_summary[length(comp) + 1] <- ''
p_value_summary[length(comp) + 1] <- ''
significance_marks[length(comp) + 1] <- ''

summary_matrix <- cbind(
  df_summary,
  SS_summary,
  MSS_summary,
  chi_sq_summary,
  p_value_summary,
  significance_marks
)
colnames(summary_matrix) <- c(
  'Df',
  'Sum Sq',
  'Mean Sq',
  'Chi^2 value',
  'Pr(>Chi^2)',
  ''
)
```

```
  rownames(summary_matrix) <- c(
    comp,
    'Residuals'
  )

  # print Analysis of Variance Table
  cat('Analysis of Variance Table\n\n')
  cat(c('Response: ', response, '\n'), sep = '')
  print(summary_matrix, quote=FALSE)
  cat('---\n')
  cat("Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1\n")
}
```