4.2 Generalized Least Squares

In the previous Chapter, we discussed that the easiest approach in *reducing* or *eliminating* unequal variances of the error terms are transformations of Y. However, transformations of the response might create an inappropriate regression relationship. Therefore, we need to consider an alternative approach to transformations. These are the **Generalized Least Squares**. In fact, this is a quite general method that can be tailored to deal with correlated error terms as well.

Throughout this section we make the following assumption about the error terms:

Model Assumptions

$$\mathbf{y} = \mathbf{X}\beta + \varepsilon, \text{ where } \varepsilon \sim N(0, \Sigma)$$

Σ is the variance-covariance matrix and is assumed to be *symmetric* and *positive definite*.

We are going to consider two scenarios:

- Σ *known*: this is an idealized case from which we can get some insight.
- Σ *unknown*: the more realistic scenario.

4.2.1 GLS: Σ known

In this section, the linear model we consider is

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

where $\varepsilon \sim \mathrm{N}_n(\mathbf{0}, \Sigma)$ and Σ is a **known**, *symmetric*, *positive definite* covariance matrix.

Let us assume that the errors are heteroscedastic and/or correlated. Then, assume that the variance-covariance matrix Σ can be decomposed as

$$\Sigma = SS^T,$$

where S is invertible (i.e. S^{-1} exists). This decomposition can be done using *Cholesky's factorization*[16], for instance, and this is something that `R` can do for us. Starting with our model

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

and multiplying the model equation by S^{-1} on both sides, we have

$$\begin{aligned} S^{-1}\mathbf{y} &= S^{-1}(\mathbf{X}\beta + \varepsilon) \\ S^{-1}\underbrace{\mathbf{y}}_{:=\mathbf{y}^*} &= \underbrace{S^{-1}}_{:=\mathbf{X}^*}\mathbf{X}\,\beta + \underbrace{S^{-1}}_{:=\varepsilon^*}\varepsilon \\ \mathbf{y}^* &= \mathbf{X}^*\beta + \varepsilon^* \end{aligned}$$

Let's identify the distribution of the error terms ε^* of the transformed model:

- Since ε^* is a scaling of the original ε, then ε^*s are still Normally distributed.
- They still have mean zero. Indeed,

$$\mathrm{E}(\varepsilon^*) = \mathrm{E}(S^{-1}\varepsilon) = 0$$

- The variance of ε^* becomes

$$\begin{aligned} Var(\varepsilon^*) &= Var(S^{-1}\varepsilon) = S^{-1}Var(\varepsilon)(S^{-1})^T \\ &= S^{-1}\Sigma(S^{-1})^T = S^{-1}SS^T(S^{-1})^T = \mathbf{I} \end{aligned}$$

Therefore, we have that

$$\varepsilon^* \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$$

Working now with the transformed model, if we do least-squares, we can compute the estimator for β:

$$\begin{aligned} \hat{\beta}_{GLS} &= (\mathbf{X}^{*T}\mathbf{X}^*)^{-1}\mathbf{X}^{*\top}\mathbf{y}^* \\ &= (\mathbf{X}^T\underbrace{(S^{-1})^T}_{=\Sigma^{-1}}S^{-1}\mathbf{X})^{-1}\mathbf{X}^T\underbrace{(S^{-1})^T}_{=\Sigma^{-1}}S^{-1}\mathbf{y} \\ &= (\mathbf{X}^T\Sigma^{-1}\mathbf{X})^{-1}\mathbf{X}^T\Sigma^{-1}\mathbf{y} \end{aligned}$$

This is the so-called **Generalized Least Squares** estimator.

Note that is is the solution that we obtain minimizing the following *RSS* criterion:

$$RSS = ||\mathbf{y}^* - \mathbf{X}^*\beta||^2 = (\mathbf{y} - \mathbf{X}\beta)^T\Sigma^{-1}(\mathbf{y} - \mathbf{X}\beta)$$

The Generalized Least Squares estimators are **unbiased**, **consistent** and have *minimum variance among unbiased linear estimators*. In fact, we can compute the variance of $\hat{\beta}_{GLS}$ to be:

$$Cov(\hat{\beta}_{GLS}) = (\mathbf{X}^T\Sigma^{-1}\mathbf{X})^{-1}$$

In general, when Σ is known, $\hat{\beta}_{GLS}$ exhibits less variability than $\hat{\beta}_{OLS}$.

4.2.2 Special Case: Weighted Least Squares (WLS)

Suppose that Σ is a diagonal matrix of unequal error variances:

$$\Sigma = diag(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2)$$

in other words

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{pmatrix}$$

In this case, the RSS criterion can be written as

$$\begin{aligned} RSS &= (\mathbf{y} - \mathbf{X}\beta)^T \Sigma^{-1} (\mathbf{y} - \mathbf{X}\beta) \\ &= \sum_{i=1}^{n} \frac{(y_i - \mathbf{x}_i^T \beta)^2}{\sigma_i^2} \\ &= \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \left(y_i - \beta_1 x_{i1} - \beta_2 x_{i2} - \ldots - \beta_p x_{ip} \right)^2 \end{aligned}$$

and the *Weighted Least Squares estimator* is the one that minimizes the *RSS* above with respect to β_1, ... β_p. Note that the errors here are weighted by

$$w_i = \frac{1}{\sigma_i^2}$$

The Weighted Least Squares criterion generalizes the Ordinary Least Squares criterion by replacing equal weights of 1 by w_i. Since the weight is inversely related to the variance σ_i^2, it *reflects the amount of information contained in the observation* y_i. Thus, an observation with large variance receives less weight than another observation that has smaller variance. This is very intuitive, since the more precise y_i is, the more information y_i provides about $\mathrm{E}(y_i)$, and therefore more weight should be assigned when fitting the regression function.

`strongx` data set from the Faraway library

A large number of observations taken for each `momentum` measurement, allows to have a good estimate of the standard deviation sd for each value of the response `crossx` at each energy level.

```
## [1] "momentum" "energy"   "crossx"   "sd"
```

```
head(strongx)
```

```
##   momentum energy crossx sd
## 1        4  0.345    367 17
## 2        6  0.287    311  9
## 3        8  0.251    295  9
## 4       10  0.225    268  7
## 5       12  0.207    253  7
## 6       15  0.186    239  6
```

A glimpse of the data reveils that in this case we know the true variances. So, we are going to use them as **weights** to run a Weighted Least Squares regression as follows:

```
strong.weights = lm(crossx ~ energy, strongx, weights=1/sd^2)
summary(strong.weights)
```

```
##
## Call:
## lm(formula = crossx ~ energy, data = strongx, weights = 1/sd^2)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3230 -0.8842  0.0000  1.3900  2.3353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  148.473      8.079   18.38 7.91e-08 ***
## energy       530.835     47.550   11.16 3.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.657 on 8 degrees of freedom
## Multiple R-squared:  0.9397, Adjusted R-squared:  0.9321
## F-statistic: 124.6 on 1 and 8 DF,  p-value: 3.71e-06
```

Let's see what the results would be if we run a standard regression with no weights:

```
strong=lm(crossx ~ energy, strongx);
summary(strong)
```

```
##
## Call:
## lm(formula = crossx ~ energy, data = strongx)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -14.773  -9.319  -2.829   5.571  19.817
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   135.00      10.08    13.4 9.21e-07 ***
## energy        619.71      47.68    13.0 1.16e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.69 on 8 degrees of freedom
## Multiple R-squared:  0.9548, Adjusted R-squared:  0.9491
## F-statistic: 168.9 on 1 and 8 DF,  p-value: 1.165e-06
```

As we can observe, there is a *small difference in the estimators*. Let's now take a look at the estimated variance, i.e. the $\hat{\sigma}^2$ term.

In the model without weights, we have

```
cbind(summary(strong)$sig^2, sum(strong$res^2)/8)
```

```
##          [,1]     [,2]
## [1,] 161.1616 161.1616
```

These two terms coincide.

However, when we use WLS, these two estimates do not agree. In fact, they are

```
cbind(summary(strong.weights)$sig^2, sum(strong.weights$res^2)/8)
```

```
##          [,1]     [,2]
## [1,] 2.744081 248.7354
```

If we want to estimate $\hat{\sigma}^2$ by hand, we will need to divide the residuals with the corresponding variance terms as follows:

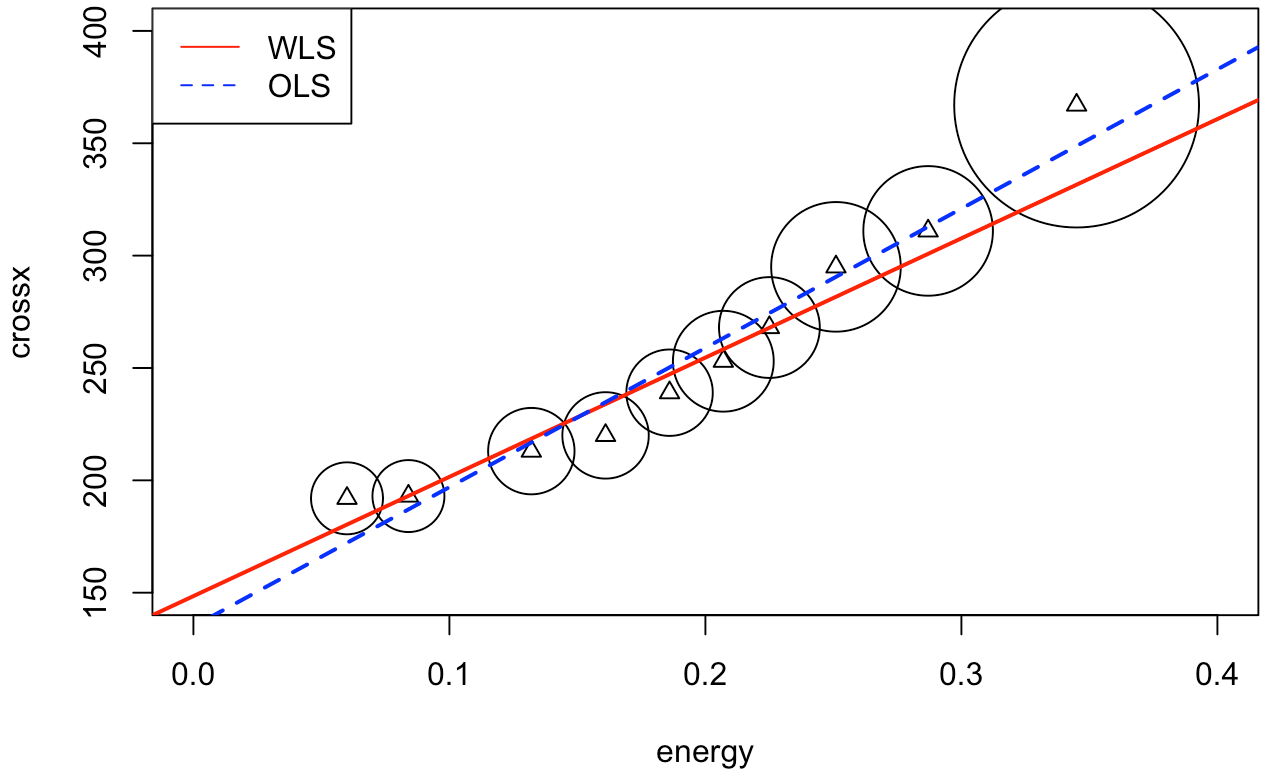
```
sum(strong.weights$res^2/strongx$sd^2)/8
```

```
## [1] 2.744081
```

Now, this quantity matched the estimated $\hat{\sigma}^2$ by `R` .

Last, let us create a scatteplot (triangles are the observations) including the OLS (blue) and WLS (red) regression lines. We also ploted the correponsing standard deviations for each observation (circles around the triangles):

```
plot(crossx ~ energy, data=strongx, cex=sd, xlim=c(0, 0.4), ylim=c(150, 400));
points(crossx ~ energy, data=strongx, pch=2)
abline(strong.weights, col="red", lty=1, lwd=2);
abline(strong, col="blue", lty=2, lwd=2);
legend("topleft", col=c("red", "blue"), lty=c(1,2), legend=c("WLS", "OLS"))
```



We observe that the WLS line departs from values with higher variance (smaller weights).

4.2.3 WLS Special case: Replicated Observations

A special case in the Weighted Least Squares method is when we have multiple observations available for each $\mathbf{x}_i$. The typical notation is to use double subscripts in y to indicate the replicate observations:

$$(\mathbf{x}_i, y_{i1}, y_{i2}, \ldots, y_{in_i})$$

This is a common situation in experiments where replicate observations are made for each combination of the levels of the predictors. If the number of replications is *large*, then the weights w_i can be directly obtained from the sample variances of each combination of levels of x.

Another idea is to let y_i denote the average of the n_i observations sharing $\mathbf{x}_i$ and define the residual sum of squares for β as

$$RSS = \sum_{i=1}^{n}\sum_{j=1}^{n_i}(y_{ij} - \mathbf{x}_i^T\beta)^2 = \sum_{i=1}^{n} n_i(\bar{y}_{i\cdot} - \mathbf{x}_i^T\beta)^2 + \sum_{i=1}^{n}\sum_{j=1}^{n_i}(y_{ij} - \bar{y}_{i\cdot})^2$$

Minimizing the *RSS* to solve for β is the same as minimizing the first term on the right only (why?). Because $Var(y_i) = \sigma^2/n_i$, we use WLS on the y_i:

$$\hat{\beta} = \arg\min_{\beta} \sum_{i=1}^{n} n_i(\bar{y}_{i\cdot} - \mathbf{x}_i^T\beta)^2.$$

4.2.4 Maximum Likelihood Estimation when Σ is known

As in the Ordinary Least Squares approach, we can use *Maximum Likelihood* to estimate the model parameters when the variance is not constant or when the error terms are correlated. Indeed, since $\mathbf{y} \sim N_n(\mathbf{X}\beta, \Sigma)$, the log-likelihood function can be written as

$$\log(p(\mathbf{y}\,|\,\beta, \Sigma)) = \log\left\{\frac{1}{(2\pi)^{n/2}\,|\,\Sigma\,|^{1/2}}\exp\left[-\frac{1}{2}(\mathbf{y} - \mathbf{X}\beta)^{\top}\Sigma^{-1}(\mathbf{y} - \mathbf{X}\beta)\right]\right\}$$

$$= -\frac{1}{2}(\mathbf{y} - \mathbf{X}\beta)^{\top}\Sigma^{-1}(\mathbf{y} - \mathbf{X}\beta) + Constant.$$

Therefore the Maximum Likelihood Estimator (MLE) for β, when Σ is known is given by

$$\hat{\beta}_{mle} = \arg\min_{\beta} (\mathbf{y} - \mathbf{X}\beta)^T\Sigma^{-1}(\mathbf{y} - \mathbf{X}\beta)$$

which is the same as the Generalized Least Squares estimator we computed in the beginning of this section.

4.2.5 GLS: Σ unknown, Uncorrelated Errors

When the variances are known, or even known up to a proportionality constant, the use of Weighted Least Squares with weights

$$w_i = k\frac{1}{\sigma_i^2}, \text{ where } k \text{ is a proportionality constant}$$

is straightforward. Unfortunately, one *rarely* has knowledge of the variances σ_i^2, let alone the full matrix Σ. Therefore, we are forced to use **estimates** of the variances/correlations to proceed.

Estimating all the entries of the Σ matrix, with no additional information on the structure of the matrix, is nearly impossible, since there are too many parameters to consider. Therefore, in order to proceed we need to make additional assumptions on the structure of the matrix that reduce the number of parameters to estimate.

Σ Unknown Diagonal Matrix

Assume that

$$\Sigma = diag(\sigma_1^2, \sigma_2^2, \ldots, \sigma_n^2).$$

The goal is to estimate the σ_i^2's or what we call the variance (or standard deviation) function.

The variance of the error terms ε_i, denoted by σ_i^2 can be expressed as

$$\sigma_i^2 = \mathrm{E}\left(\varepsilon_i^2\right) - \left(\mathrm{E}(\varepsilon_i)\right)^2$$

Since we assume that $\mathrm{E}(\varepsilon_i) = 0$, we have

$$\sigma_i^2 = \mathrm{E}\left(\varepsilon_i^2\right)$$

This implies that

- the squared residual r_i^2 is an estimator of σ_i^2, or
- the absolute residual $|r_i|$ is an estimator of the standard deviation σ_i.

Estimation of Variance/Standard Deviation Function

Variance Function

1. Fit a regression model using Ordinary Least Squares.
2. Obtain the *residuals* r_i.
3. Regress the *squared residuals* r_i^2 against the appropriate predictor variables (the same as in step 1). Denote the *fitted values* of this regression as $\hat{v}_i$.
4. The **estimated weights** are computed using the fitted values from the regression in Step 3, i.e.

$$w_i = \frac{1}{\hat{v}_i}$$

Standard Deviation Function

Steps 1 and 2 the same as above.

3. Regress the *absolute residuals* $|r_i|$ against the appropriate predictor variables. Denote the *fitted values* of this regression as $\hat{s}_i$.
4. The **estimated weights** are computed using the fitted values from the regression in Step 3, i.e.

$$w_i = \frac{1}{\hat{s}_i^2}$$

The estimated variances are then placed in the variance-covariance matrix Σ and the regression coefficients are estimated using the *Weighted Least Squares* method.

Blood Pressure Data Example

A health researcher interested in studying the relationship between `diastolic blood pressure` and `age` among healthy women 20 to 60 years old, collected data on 54 subjects.

```
pressure <- read.table("data/ch4/blood_pressure.txt", header=FALSE)
names(pressure)=c("age", "pressure")
head(pressure)
```

```
##   age pressure
## 1  27       73
## 2  21       66
## 3  22       63
## 4  24       75
## 5  25       71
## 6  23       70
```

We start by fitting a linear regression:

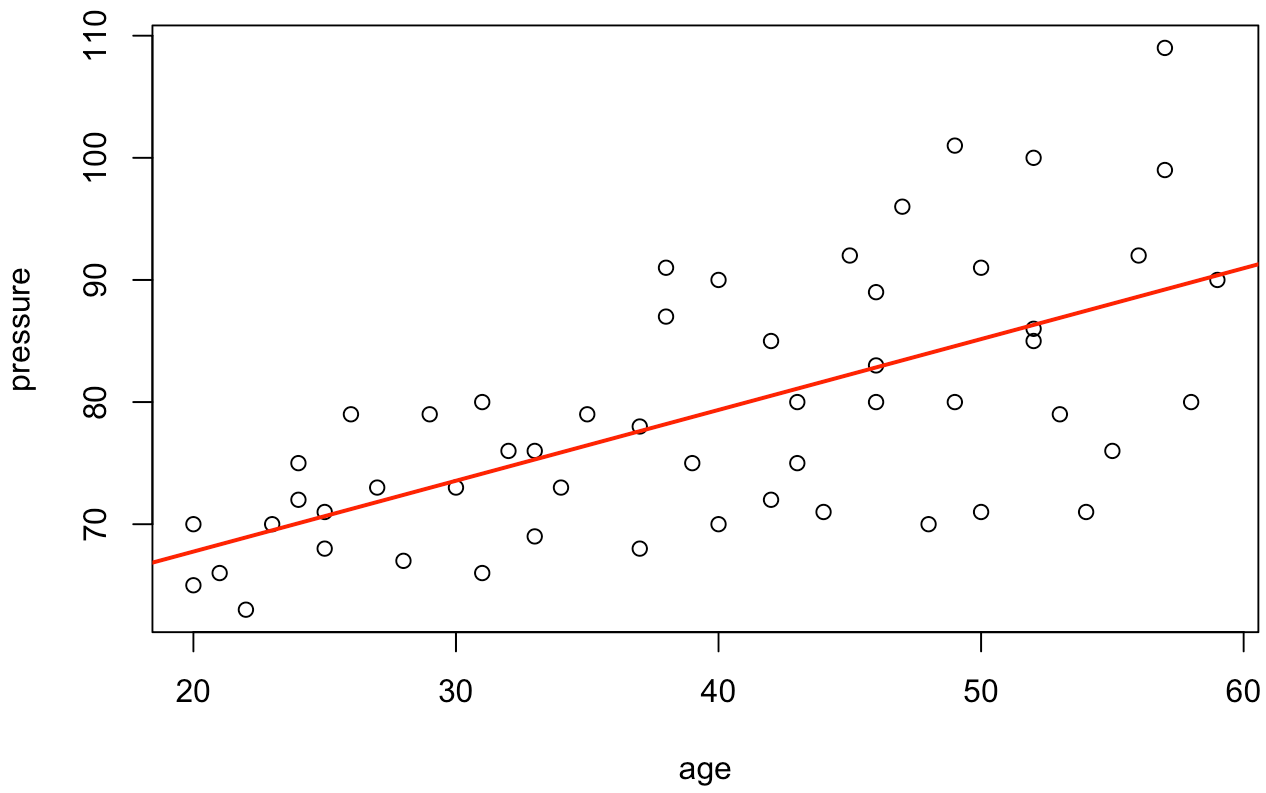
```
lm.pressure = lm(pressure~age, data=pressure)
summary(lm.pressure)
```

```
##
## Call:
## lm(formula = pressure ~ age, data = pressure)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4786  -5.7877  -0.0784   5.6117  19.7813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 56.15693    3.99367  14.061  < 2e-16 ***
## age          0.58003    0.09695   5.983 2.05e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.146 on 52 degrees of freedom
## Multiple R-squared:  0.4077, Adjusted R-squared:  0.3963
## F-statistic: 35.79 on 1 and 52 DF,  p-value: 2.05e-07
```

The estimators for the slope and intercept are statistically significant, the F-test for regression has a low p-value, which means that the predictor (`Age`) significant in estimating the response. The R^2 is relatively low, around 40% indicating a weak relationship between `age` and `blood pressure` .

We can also plot the raw data and the fitted regression line:

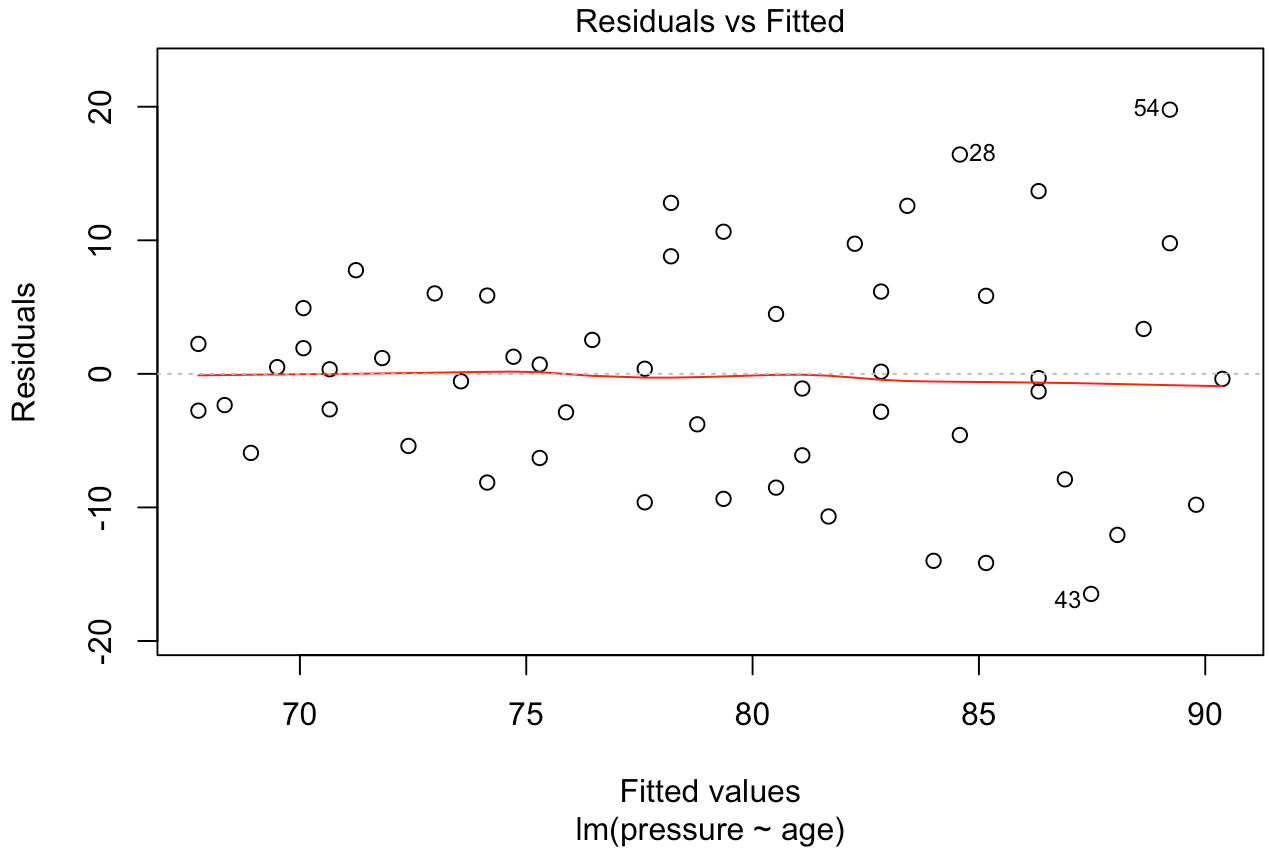
```
plot(pressure ~ age, data=pressure)
abline(lm.pressure, col="red", lty=1, lwd=2)
```



where we observe a linear relationship between `diastolic blood pressure` and `age`.

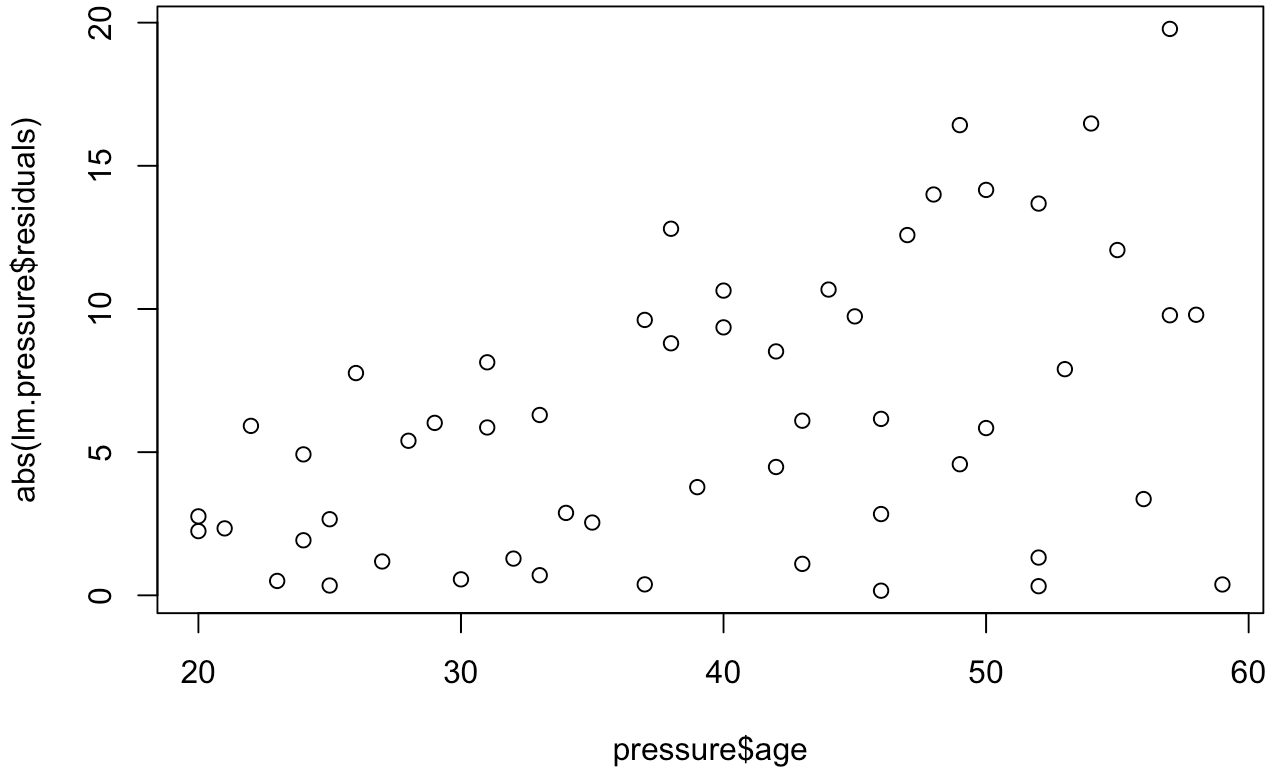
When we look at the residual plot of fitted values against residuals, we observe that the variance is not constant. In fact, it increases with age:

```
plot(lm.pressure, which=1)
```



So, let us plot the **absolute residuals** against `Age` :

```
plot(abs(lm.pressure$residuals) ~ pressure$age)
```



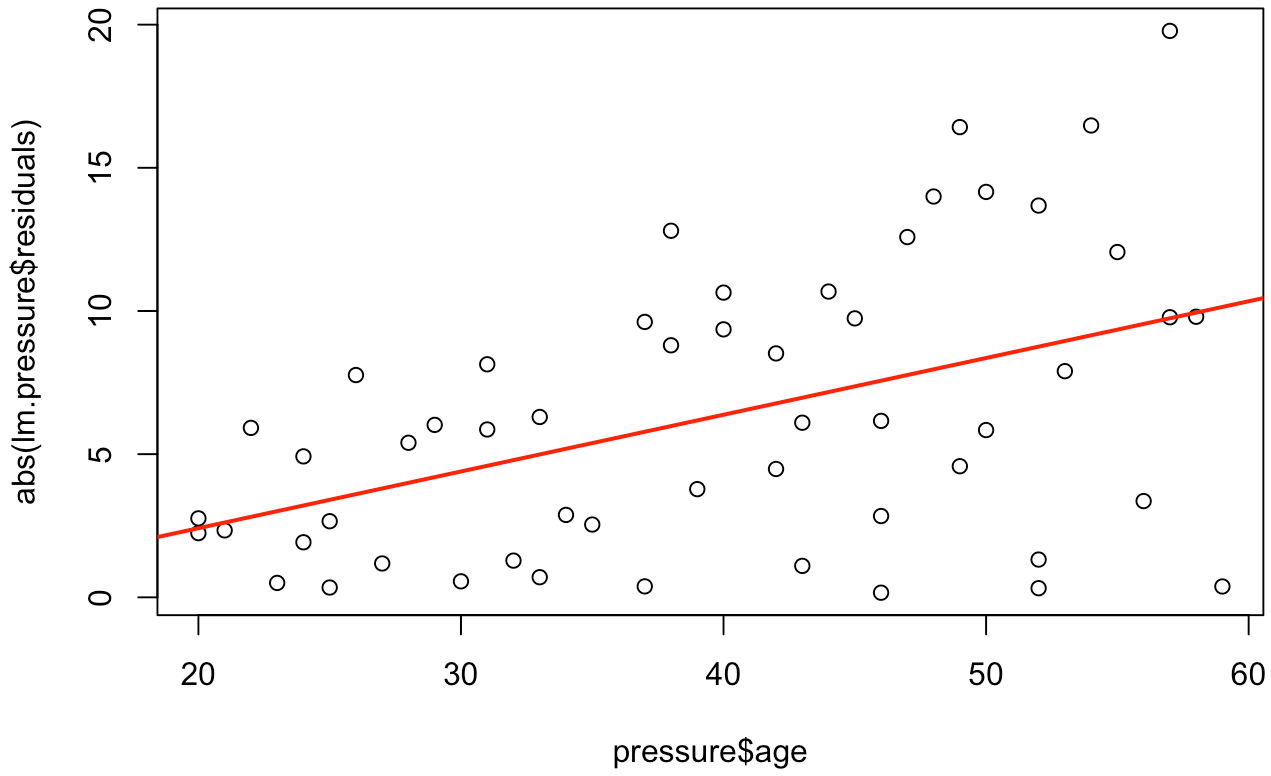
This plot suggests that a linear relation between the error standard deviation and `Age` may be reasonable, which means that we can use it to estimate the unknown standard deviation function. So, we fit a regression line between `Age` and absolute residuals

```
lm.resid = lm(abs(lm.pressure$residuals) ~ pressure$age)
summary(lm.resid)
```

```
##
## Call:
## lm(formula = abs(lm.pressure$residuals) ~ pressure$age)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7639 -2.7882 -0.1587  3.0757 10.0350
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.54948    2.18692  -0.709  0.48179
## pressure$age  0.19817    0.05309   3.733  0.00047 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.461 on 52 degrees of freedom
## Multiple R-squared:  0.2113, Adjusted R-squared:  0.1962
## F-statistic: 13.93 on 1 and 52 DF,  p-value: 0.0004705
```

We plot the fitted line along with the residuals:

```
plot(abs(lm.pressure$residuals) ~ pressure$age)
abline(lm.resid, col="red", lty=1, lwd=2)
```



The fitted values of this last regression model will be used to compute the weights:

$$w_i = \frac{1}{\hat{s}_i^2}$$

We compute them as follows:

```
pressure$weight = 1/(lm.resid$fitted.values^2)
head(pressure)
```

```
##   age pressure     weight
## 1  27       73 0.06920928
## 2  21       66 0.14655708
## 3  22       63 0.12661657
## 4  24       75 0.09725115
## 5  25       71 0.08625993
## 6  23       70 0.11048521
```

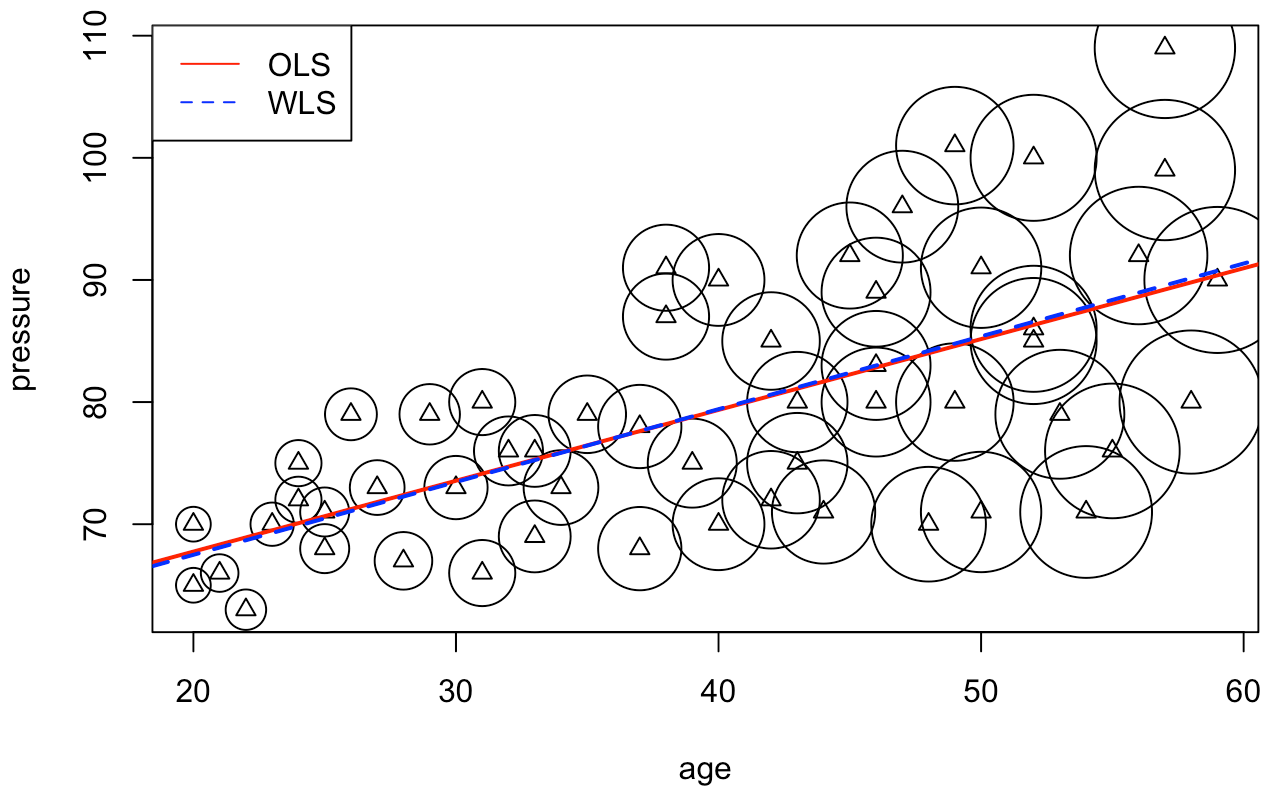
Now, we are going to use these weights in a Weighted Least Squares regression and we will re-fit our model:

```
lm.pressure.weights = lm(pressure~age, data=pressure, weights=weight)
summary(lm.pressure.weights)
```

```
##
## Call:
## lm(formula = pressure ~ age, data = pressure, weights = weight)
##
## Weighted Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0230 -0.9939 -0.0327  0.9250  2.2008
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 55.56577    2.52092  22.042  < 2e-16 ***
## age          0.59634    0.07924   7.526 7.19e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.213 on 52 degrees of freedom
## Multiple R-squared:  0.5214, Adjusted R-squared:  0.5122
## F-statistic: 56.64 on 1 and 52 DF,  p-value: 7.187e-10
```

Finally, let us plot the raw data along with the two regression lines, with and without weights:

```
plot(pressure ~ age, data=pressure, cex=lm.resid$fitted.values);
points(pressure ~ age, data=pressure, pch=2)
abline(lm.pressure, col="red", lty=1, lwd=2);
abline(lm.pressure.weights, col="blue", lty=2, lwd=2);
legend("topleft", col=c("red", "blue"), lty=c(1,2), legend=c("OLS", "WLS"))
```



We observe that the two fitted lines are not much different with each other. However, if we print the variance-covariance matrices

```
vcov(lm.pressure)
```

```
##             (Intercept)          age
## (Intercept)  15.9494301 -0.371977563
## age          -0.3719776  0.009399527
```

```
vcov(lm.pressure.weights)
```

```
##             (Intercept)          age
## (Intercept)   6.3550256 -0.189363636
## age          -0.1893636  0.006278666
```

we observe that **the variance of the WLS estimators are much smaller than the OLS estimators**.

An Iteration to estimate Σ

How about using the following iterative approach?

1. Start with some initial guess of Σ
2. Use Σ to estimate β
3. Use residuals (since we have known β) to estimate Σ
4. Iterate until convergence.

Of course, this will work only if we assume some structure about Σ, since otherwise there are too many parameters to be estimated.

4.2.6 GLS: Σ Unknown, Correlated Errors

Based on the application, we can assume a particular structure for Σ that does not involve too many parameters. Then, we can model β and Σ simultaneously.

In many applications, we find that the observations are linearly correlated. For example, today's observation depends on the observation from yesterday or the day before in a linear fashion. To check for linear dependence, we either use the *Sequence Plot* or the *Durbin-Watson* test statistic. Once we identify the presence of correlation in the error ters, then we can use this to information to fit a regression with **autocorrelated errors**.

In fact, the simplest model we can fit is

$$y_i = \beta_0 + \beta_1 y_i + \varepsilon_i$$

where

$$\varepsilon_i = \rho\, \varepsilon_{i-1} + u_i$$

where the u_i terms are independent normal variables (called disturbances) with mean 0 and

variance σ^2. This implies that any error term is the sum of the previous error term and a new disturbance term. The parameter ρ is called the *autocorrelation* parameter. The model we assumed for the error terms is called a *first order autoregressive model*, in short AR(1).

If the error follow an **AR(1) times series model** (auto-regressive model of order 1), the structure of Σ is:

$$\Sigma = \sigma^2 \begin{pmatrix} 1 & \rho & \rho^2 & \rho^3 & \dots \\ \rho & 1 & \rho & \rho^2 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \rho^{n-1} & \rho^{n-2} & \dots & \dots & 1 \end{pmatrix}$$

which means that Σ is a function of **only** two parameters, ρ and σ^2. Therefore, we need to use appropriate methods to estimate them.

We are not going to discuss the theoretical details of this approach, but we are going to discuss how this is implemented in `R` using the `gls` function from the `nlme` package in the example below:

Company Sales Example

A company wants to predict its sales by using industry sales that are available from the industry's trade association as a *predictor*.

```
sales <- read.table("data/ch4/Sales.txt", header=FALSE)
names(sales)=c("company_sales", "industry_sales")
sales$index = seq(1:dim(sales)[1])
head(sales)
```

```
##   company_sales industry_sales index
## 1         20.96          127.3     1
## 2         21.40          130.0     2
## 3         21.96          132.7     3
## 4         21.52          129.4     4
## 5         22.39          135.0     5
## 6         22.76          137.1     6
```

We start by fitting a simple linear regression

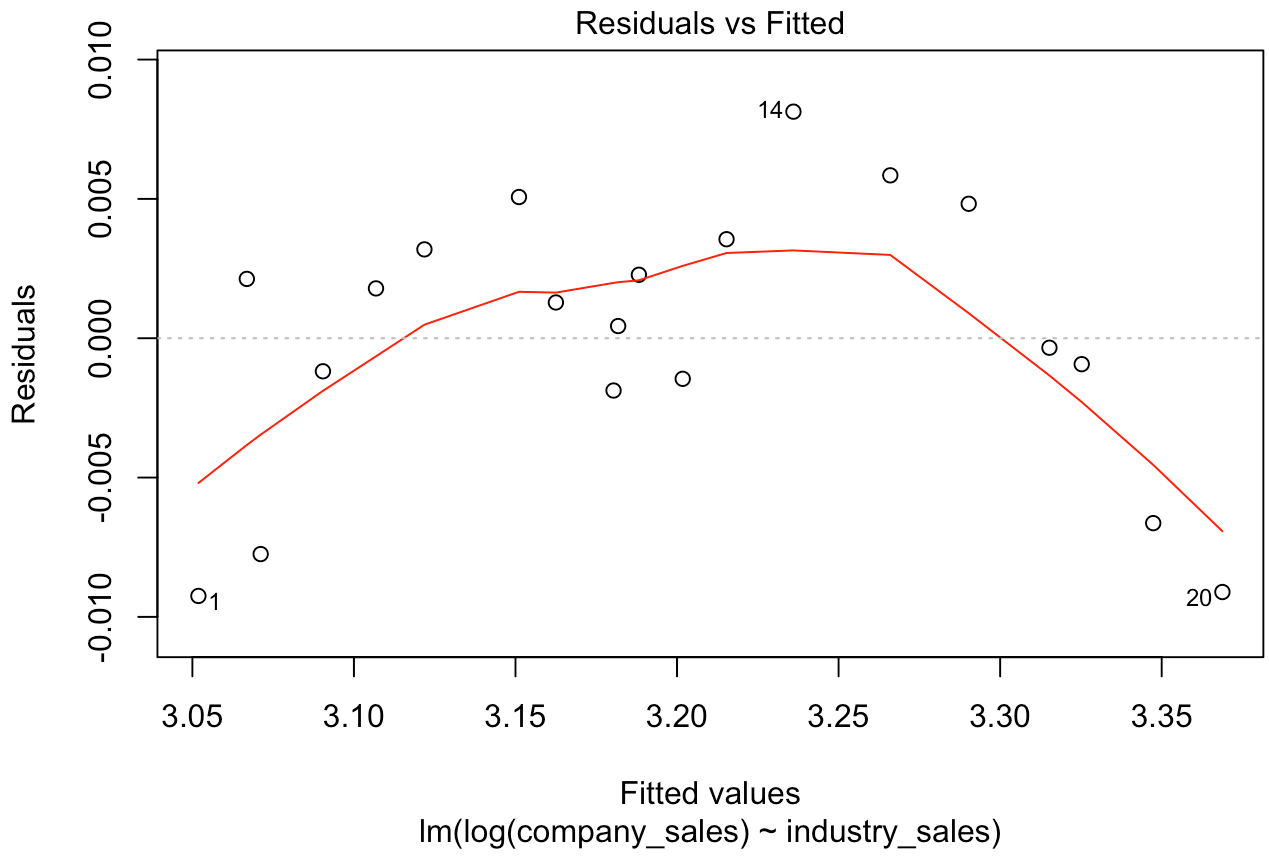
```
lm.sales = lm(log(company_sales)~industry_sales, data=sales)
summary(lm.sales)
```

```
##
## Call:
## lm(formula = log(company_sales) ~ industry_sales, data = sales)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0092483 -0.0015613  0.0008606  0.0032798  0.0081309
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.143e+00  1.268e-02  169.03   <2e-16 ***
## industry_sales 7.138e-03  8.554e-05   83.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005095 on 18 degrees of freedom
## Multiple R-squared:  0.9974, Adjusted R-squared:  0.9973
## F-statistic:  6963 on 1 and 18 DF,  p-value: < 2.2e-16
```

The fit of the model looks great!? All the p-values are low, indicating that the industry sales indeed explain the variation in the response and the R^2 is extremely high.

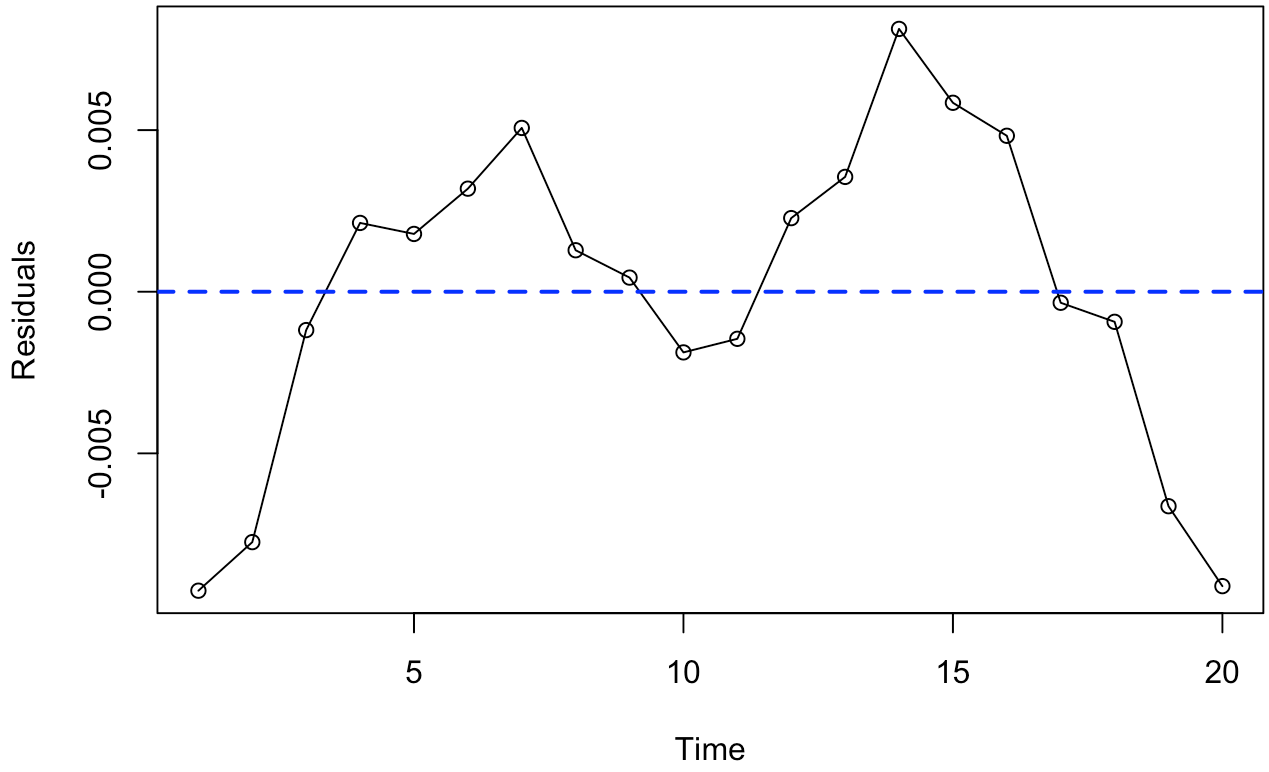
Let's take a closer look at the model and check the residual plots:

```
plot(lm.sales, which=1)
```



The fitted against residuals plots show that the variance may not be constant. Let's also plot the residuals against time (the index column in the data frame):

```
plot(lm.sales$residuals ~ sales$index, type='o', xlab="Time", ylab="Residuals")
abline(h=0, lty=2, col="blue", lwd=2)
```



The sequence plot suggests a *pattern* in the residuals. Let us confirm that via a proper hypothesis test, such as the **Durbin-Watson** test:

```
library(lmtest)
dwtest(lm.sales)
```

```
##
##  Durbin-Watson test
##
## data:  lm.sales
## DW = 0.40854, p-value = 3.281e-07
## alternative hypothesis: true autocorrelation is greater than 0
```

Based on the p-value of the DW test, we conclude that the error terms are *positively auto-correlated*.

In order to fit such a model in `R`, we use the `gls` function in the `nlme` package as follows:

```
library(nlme)
lm.sales.cor = gls(company_sales~industry_sales, correlation = corAR1(form= ~ inc
```

Below is the summary output for the new regression model. It is different than the output we are used to with the `lm` function, but it contains the information about model estimates, st. errors and t test, as will as the estimated ρ coefficient (called phi in R) as well as the estimated variance.

```
summary(lm.sales.cor)
```

```
## Generalized least squares fit by REML
##   Model: company_sales ~ industry_sales
##   Data: sales
##         AIC       BIC   logLik
##   -31.74311 -28.18162 19.87156
##
## Correlation Structure: AR(1)
##  Formula: ~index
##  Parameter estimate(s):
## Phi
##   1
##
## Coefficients:
##                     Value Std.Error  t-value p-value
## (Intercept)    -0.3189197 2041.6945 -0.00016  0.9999
## industry_sales  0.1684878    0.0051 33.06272  0.0000
##
##  Correlation:
##                (Intr)
## industry_sales 0
##
## Standardized residuals:
##           Min            Q1           Med            Q3           Max
## -9.036061e-05 -4.156415e-05 -3.013053e-06  8.080346e-05  1.091922e-04
##
## Residual standard error: 2041.694
## Degrees of freedom: 20 total; 18 residual
```

Because of the complexity of the model, all parameters are estimated using a different method called *Restricted Maximum Likelihood*. The details of this method are beyond the scope of the course.