Case Study 2 - MAP566

Code

1. Fitting a linear model

The file sales1.csv consists of quarterly sales volumes (in % and indexed to the time 0) of a product.

1.1 Plot the data

Hide

```
data <- read.csv("/Users/haliouanaomie/PolytechniqueS2/MAP566/salesData/sales1.csv")
head(data)
```

	time <int>	y <dbl>
1	1	101.82562
2	4	103.16069
3	7	99.48002
4	10	103.25016
5	13	106.46870
6	16	108.53795

6 rows

Hide

```
summary(data)
```

```
     time          y         
 Min.   : 1   Min.   : 99.48  
 1st Qu.:16   1st Qu.:106.47  
 Median :31   Median :110.42  
 Mean   :31   Mean   :111.77  
 3rd Qu.:46   3rd Qu.:116.83  
 Max.   :61   Max.   :125.07  
```

Hide

```
dim(data)
```

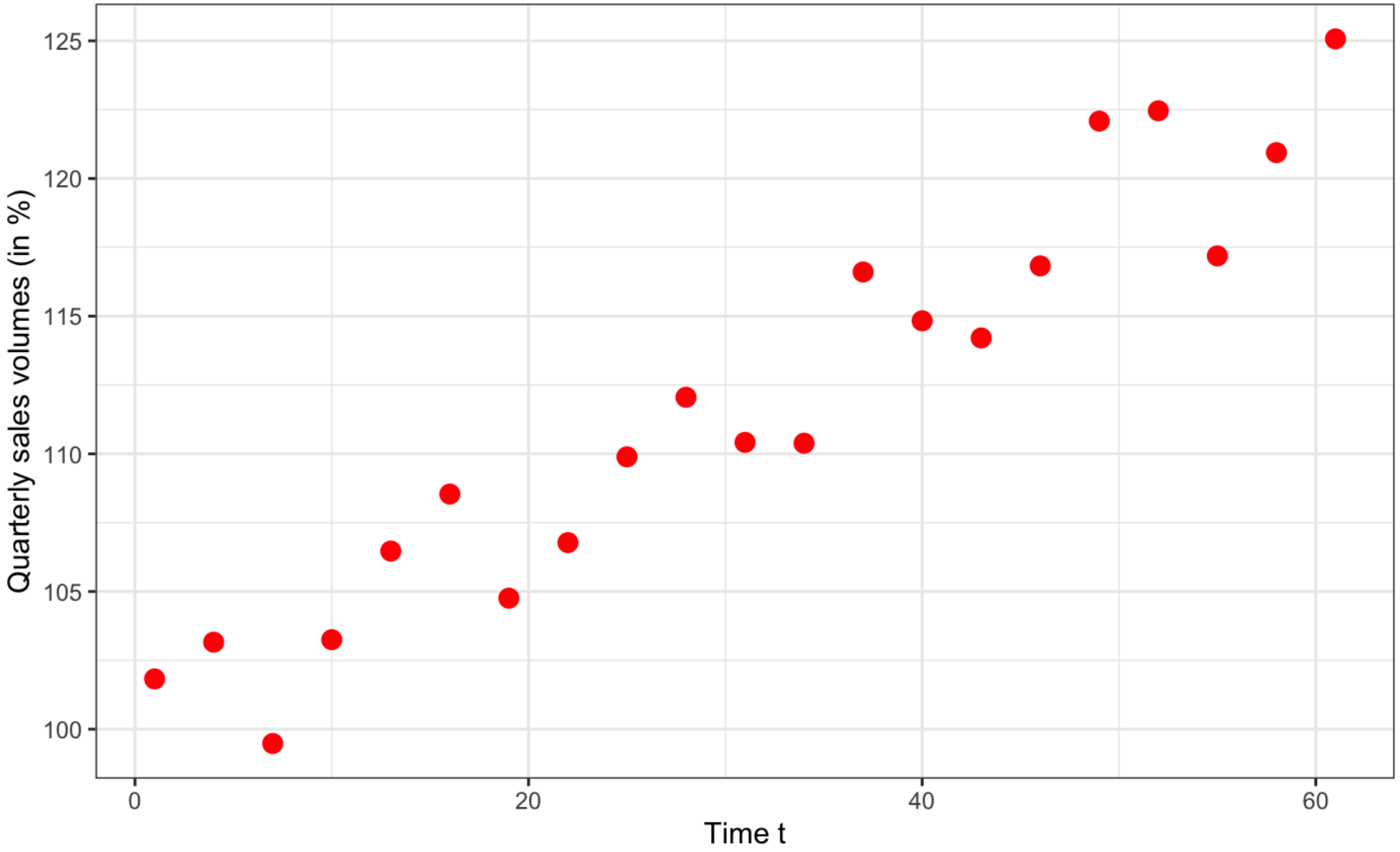
```
[1] 21  2
```

The data consists of 21 observations (rows) and 2 variables (columns): time in months and the quarterly sales volumes in percentage y.

Let's scatter plot the data in order to better visualize the relationship between the explanatory variable and the response variable.

Hide

```
library(ggplot2)
theme_set(theme_bw())
pl <- ggplot(data=data) + geom_point(aes(x=time,y=y), color="red", size=3) + xlab("Time t") + ylab("Quarterly sal
es volumes (in %)")
pl
```



We easily observe on the above scatter plot that there is a clear increasing trend in our data. It suggests a linearly increasing relationship between the explanatory and response variables.

1.2 Fit a polynomial model to this data (justify the choice of the degree)

Based on this data, our objective is to fit a polynomial model to this data by building a regression model of the form:

$$y_j = f(x_j) + e_j \quad ; \quad 1 \leq j \leq n$$

where $(xj, 1 \leq j \leq n)$ and $(yj, 1 \leq j \leq n)$ represent, respectively, the n=21 measured time and quarterly sales volumes and where $(ej, 1 \leq j \leq n)$ is a sequence of residual errors. In other words, ej represents the difference between the sale volume predicted by the model f(xj) and the observed sale volume yj.

We will restrict ourselves to polynomial regression, by considering functions of the form

$$f(x) = f(x; c0, c1, c2, \ldots, cd) = c0 + c1x + c2x2 + \ldots + cdxd$$

Fitting a polynomial of degree 1

As said earlier, we can easily observe that there is a clear increasing trend in our data. Thus, the function we should be considering is at least of degree 1, and intuitively we can think that the best model will be of degree 1. Let us therefore assume a linear trend and fit a polynomial of degree 1 using the lm function:

$$yj = c0 + c1xj + ej; 1 \leq j \leq n$$

Hide

```
lm1 <- lm(y ~ time, data=data)
coef(lm1)
```

```
(Intercept)        time
 99.9968826   0.3798515
```

These coefficients are the intercept and the slope of the regression line, but more informative results about this model are available.

Hide

```
summary(lm1)
```

```

Call:
lm(formula = y ~ time, data = data)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-3.6994 -1.5777 -0.3596  1.6444  3.4747

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  99.99688    0.94971  105.29  < 2e-16 ***
time          0.37985    0.02643   14.37 1.17e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.2 on 19 degrees of freedom
Multiple R-squared:  0.9158,    Adjusted R-squared:  0.9113
F-statistic: 206.5 on 1 and 19 DF,  p-value: 1.166e-11
```

The slope `c1` is clearly statistically significant (p-value = 1.17e-11) while the model explains about 91% of the variability of the data. The confidence interval for c1 confirms that an increase of the time leads to a significant increase of the variable y. The residuals are approximately zero.

Hide

```
confint(lm1)
```

```
                2.5 %      97.5 %
(Intercept) 98.0091258 101.9846394
time         0.3245292   0.4351738
```

Rhese numbers refer to how percentage of the data is bellow of these limits. So, we have 2.5% of the value bellow 98.009 and 97.5% of the value bellow 101.98. The confidence interval for c1 confirms that an increase of the time leads to an increase of the sale volume.

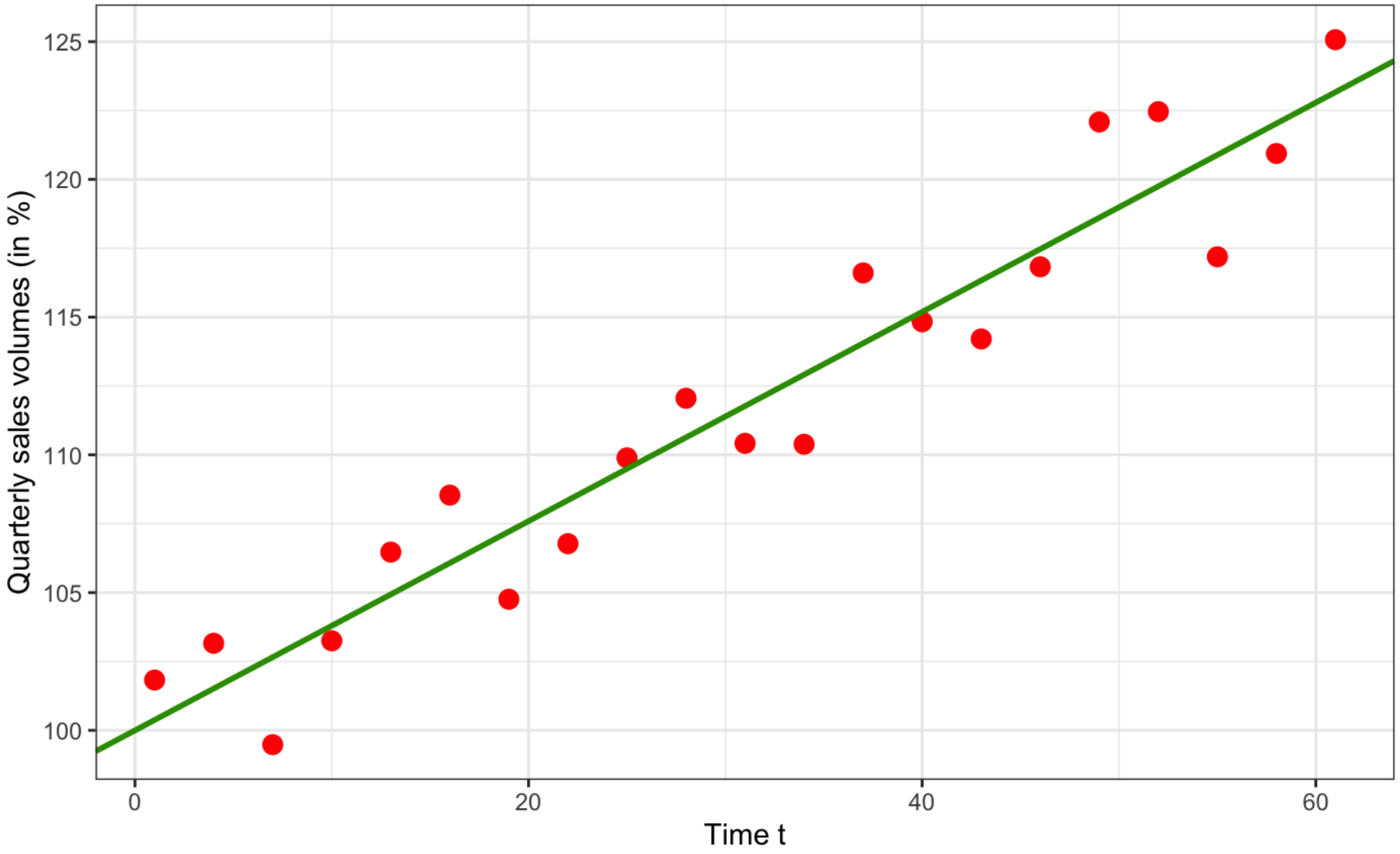
The fact that the slope is significantly different from zero does not imply that this polynomial model of degree 1 correctly describes the data: at this stage, we can only conclude that a polynomial of degree 1 better explains the variability of the data than a constant model.

Diagnostic plots are visual tools that allows one to see if something is not right between a chosen model and the data it is hypothesized to describe.

First, we can add the regression line to the plot of the data.

Hide

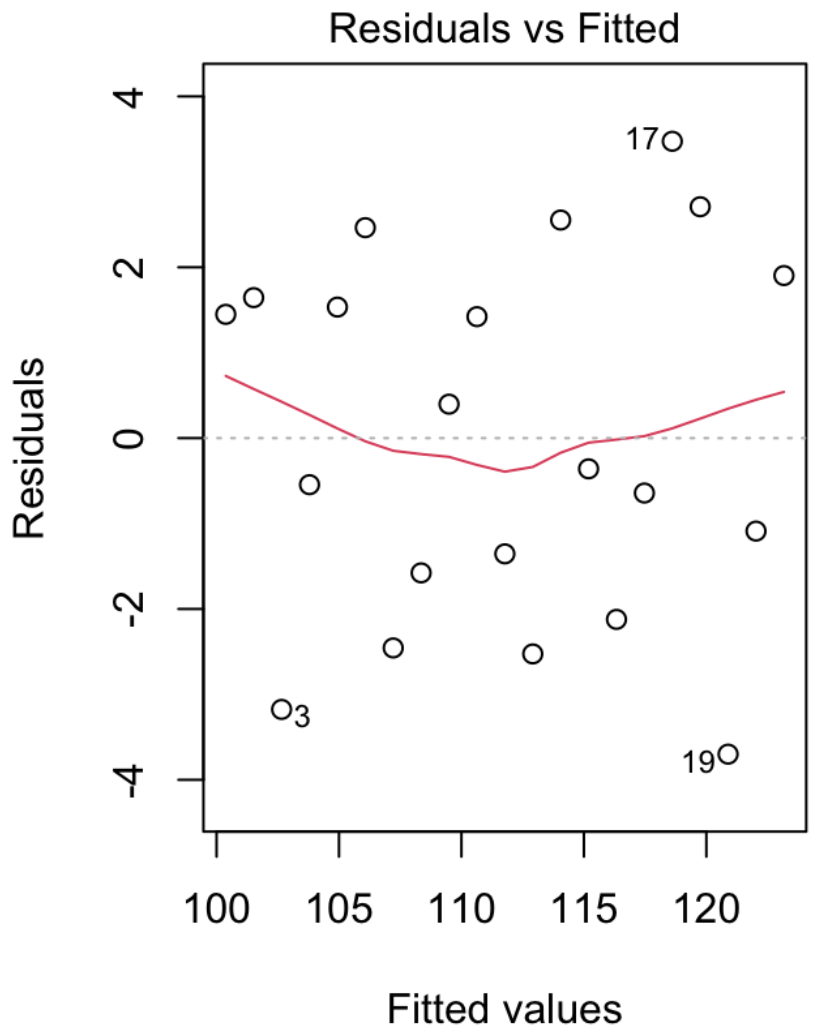
```
pl  + geom_abline(intercept=coef(lm1)[1],slope=coef(lm1)[2],size=1, colour="#339900")
```

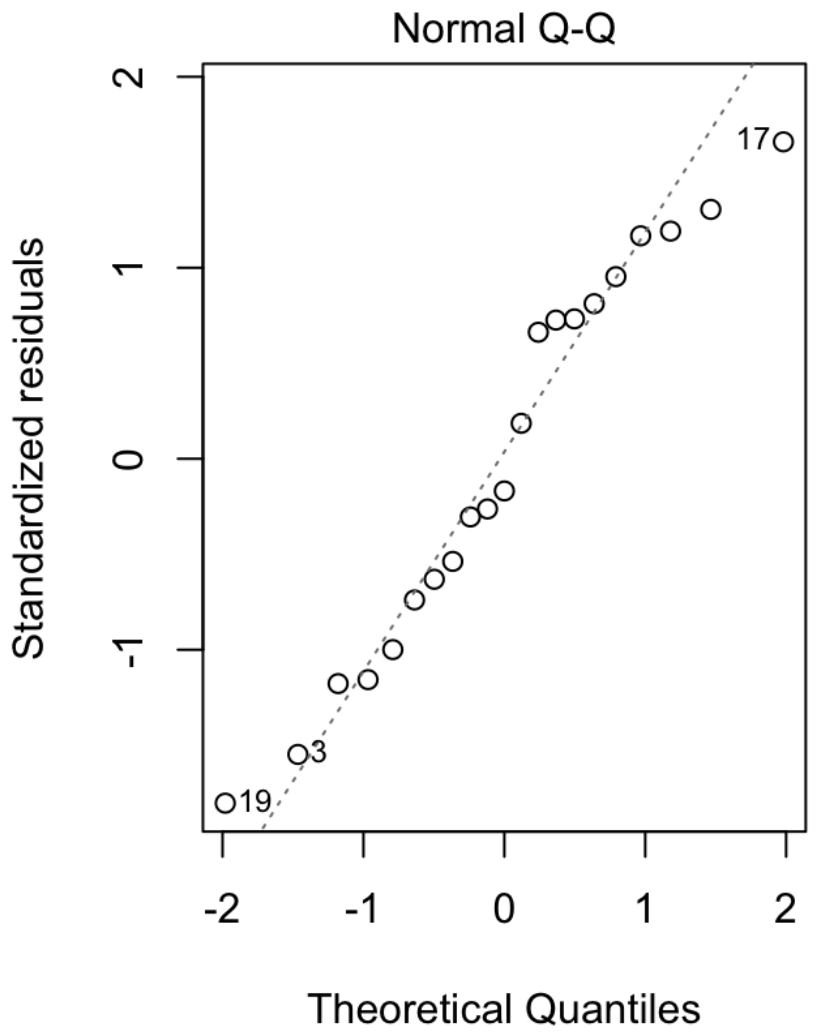


The regression line decribes pretty well the global trend in the data: based on this graphic, there is no reason to reject the model. At this stage, we can only conclude that a polynomial of degree 1 visually seems to fit our data. Several diagnotic plots are available for a lm object. The first two are a plot of residuals against fitted values and a normal QQ plot.

Hide

```
par(mfrow = c(1, 2))
plot(lm1, which=c(1,2))
```

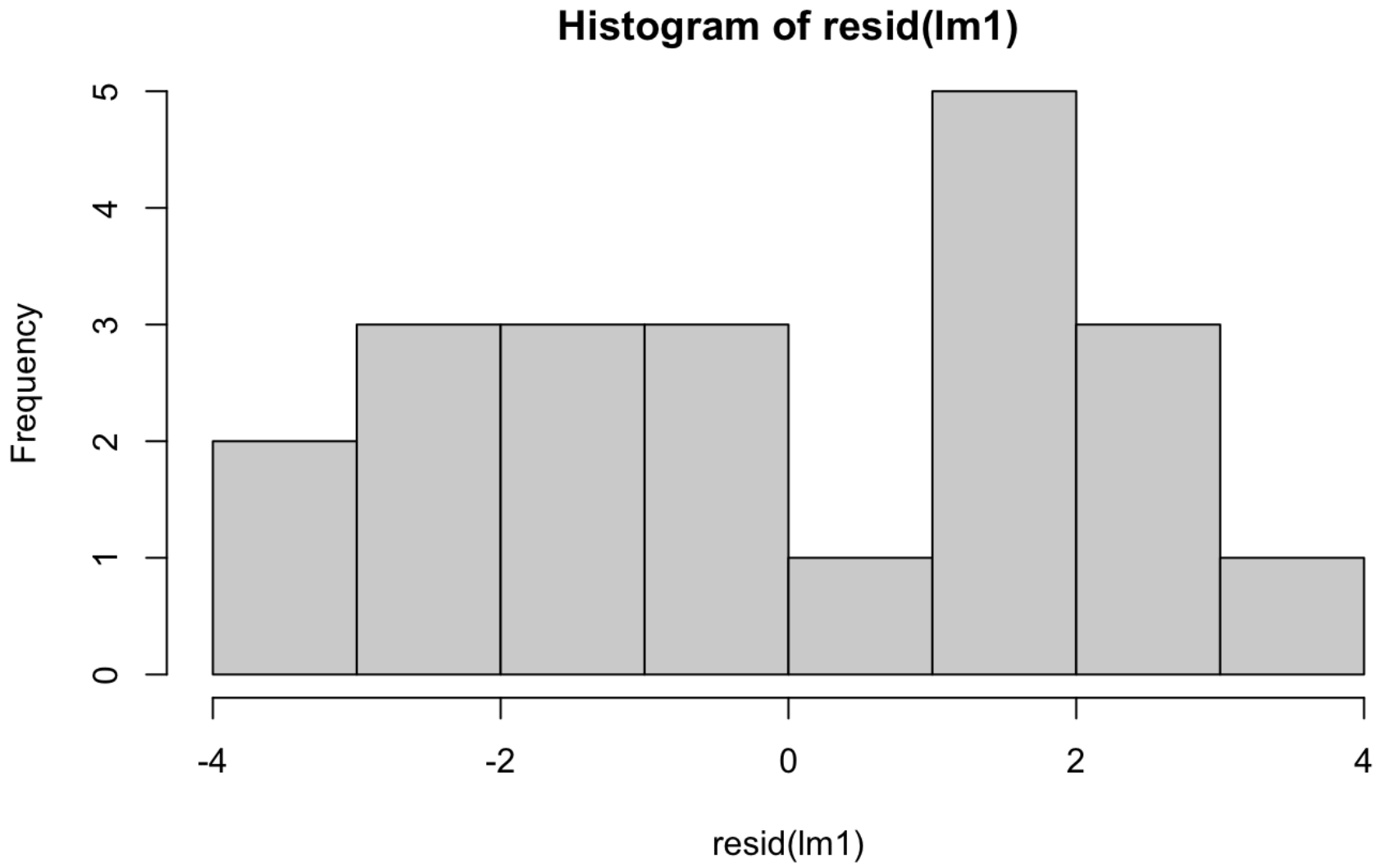




The residual plot shows a slight decreasing then increasing trend which suggests that the residuals are not identically distributed around 0 and that linearity is violated There seems to be a polynomial relationship (with degree largr than 1). Furthermore, the Quantile-Quantile plot shows that the extreme residual values are not the extreme values of a normal distribution. We observe that, points 3, 17 and 19 may be outliers, with large residual values.

Hide

```
hist(resid(lm1))
```



Histogram of the Residuals show that the deviation is not normally distributed.

Fitting a polynomial of degree 2

We can expect to better describe the extreme values by using a polynomial of higher degree. Let us therefore fit a polynomial of degree 2 to the data. $yj = c0 + c1xj + c2x2j + ej; 1 \leq j \leq n$

Hide

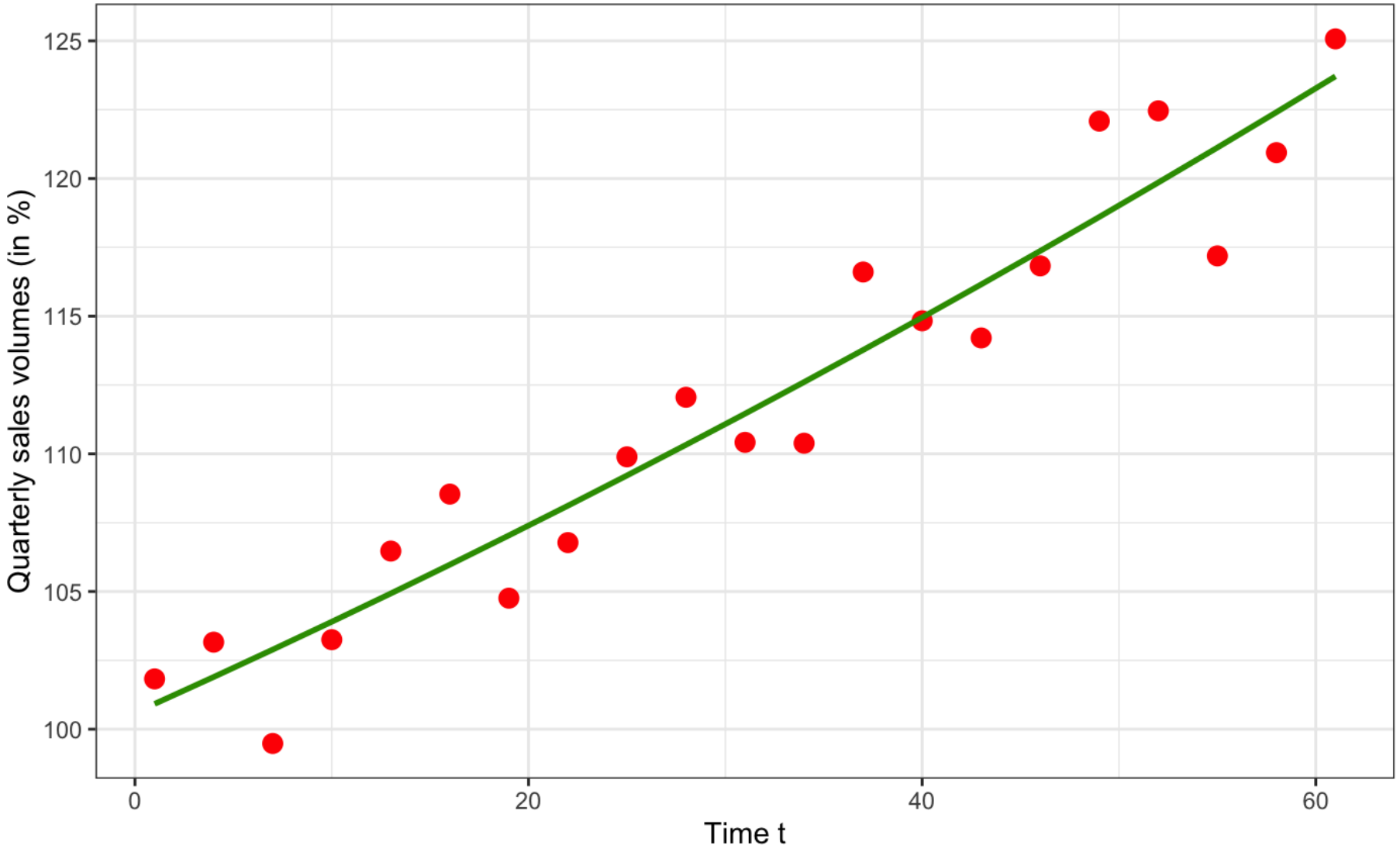
```
lm2 <- lm (y ~  time + I(time^2), data =data)
summary(lm2)
```

```
Call:
lm(formula = y ~ time + I(time^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-3.9334 -1.4679 -0.1228  1.5395  3.4804

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.006e+02  1.426e+00  70.521  < 2e-16 ***
time        3.209e-01  1.065e-01   3.013  0.00747 **
I(time^2)   9.511e-04  1.662e-03   0.572  0.57422
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.24 on 18 degrees of freedom
Multiple R-squared:  0.9173,    Adjusted R-squared:  0.9081
F-statistic: 99.77 on 2 and 18 DF,  p-value: 1.818e-10
```

c2 is clearly statistically significant while the model explains about 91.7% of the variability of the data. The residuals are approximately zero.

Hide

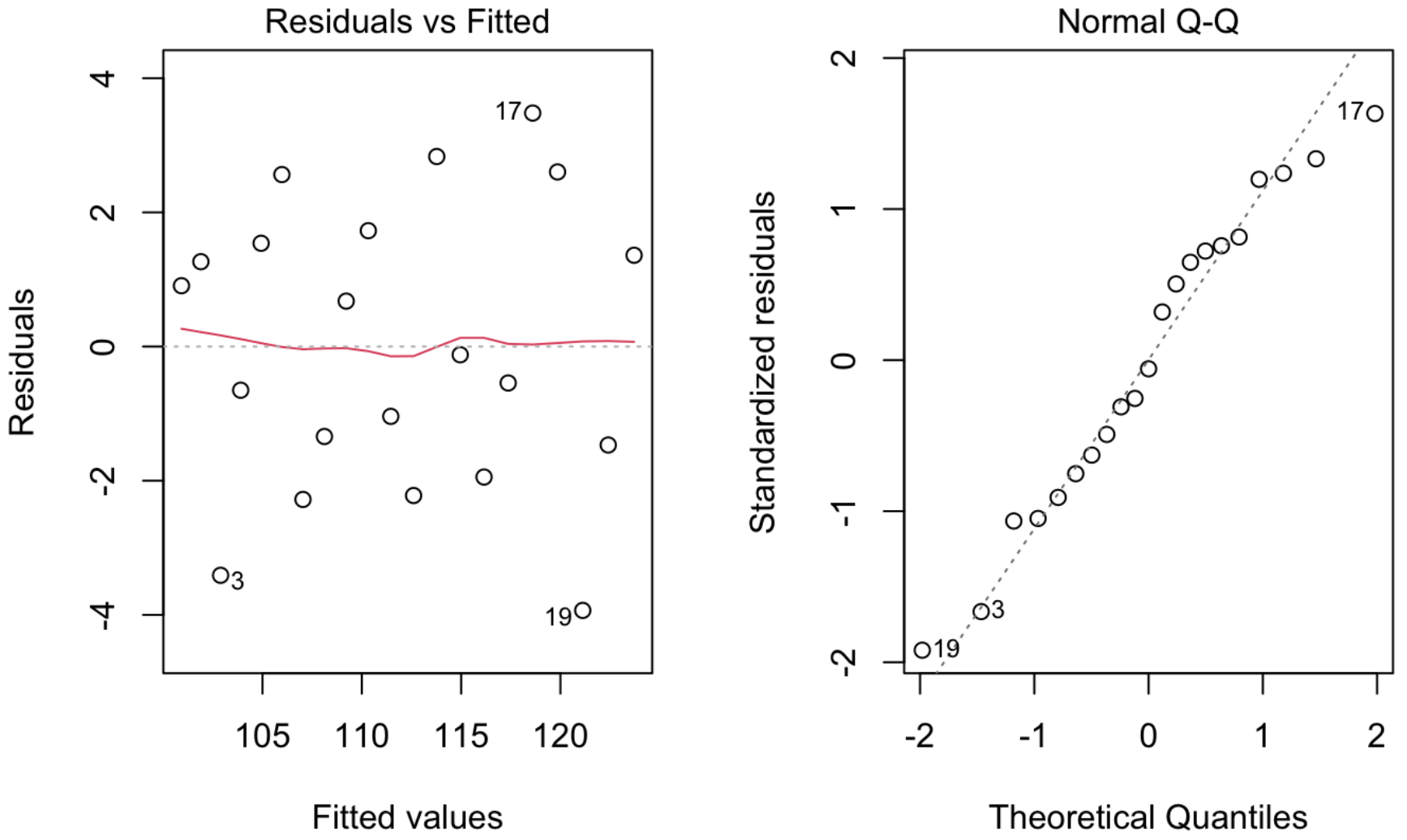
```
pl  + geom_line(aes(x=time, y=predict(lm2)),size=1, colour="#339900")
```



Again, the regression line decribes pretty well the global trend in the data: based on this graphic, there is no reason to reject the model. Several diagnotic plots are available for a lm object. The first two are a plot of residuals against fitted values and a normal QQ plot.

Hide

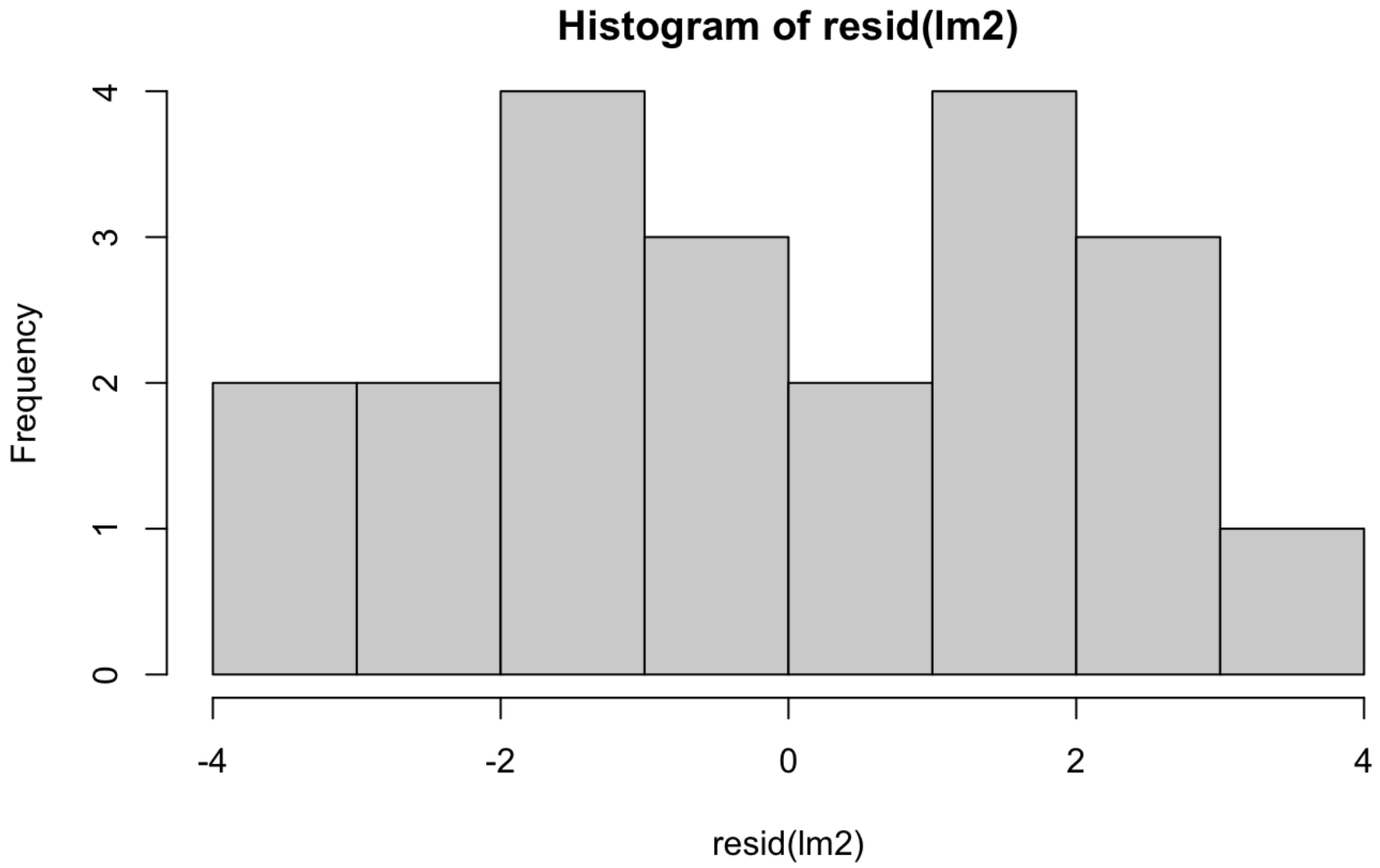
```
par(mfrow = c(1, 2))
plot(lm2, which=c(1,2))
```



The residual plot shows a linear trend which suggests that the residuals are identically distributed around 0. Furthermore, the QQ plot shows that the extreme residual values are not the extreme values of a normal distribution.

Hide

```
hist(resid(lm2))
```



Histogram of the Residuals show that the deviation is not normally distributed. **Fitting a polynomial of degree 5**

Hide

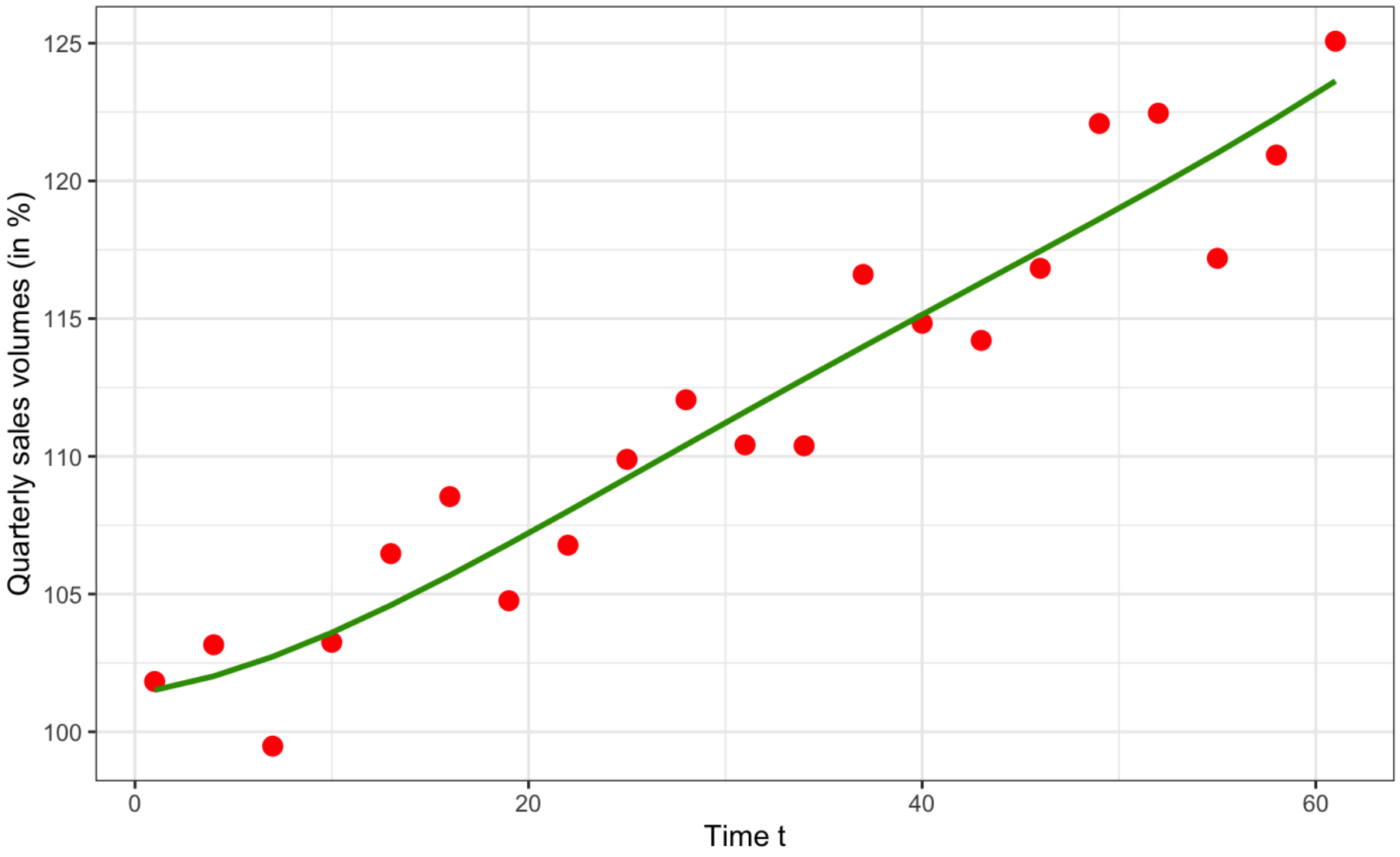
```
lm6 <- lm(y ~ poly(time, degree=5), data=data)
summary(lm6)
```

```

Call:
lm(formula = y ~ poly(time, degree = 5), data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.8282 -1.3464 -0.3117  1.6405  3.4688 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)             111.77228    0.53270 209.823  < 2e-16 ***
poly(time, degree = 5)1  31.62136    2.44113  12.954 1.51e-09 ***
poly(time, degree = 5)2   1.28207    2.44113   0.525    0.607    
poly(time, degree = 5)3  -0.74968    2.44113  -0.307    0.763    
poly(time, degree = 5)4   0.62850    2.44113   0.257    0.800    
poly(time, degree = 5)5  -0.04827    2.44113  -0.020    0.984    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.441 on 15 degrees of freedom
Multiple R-squared:  0.9181,    Adjusted R-squared:  0.8908 
F-statistic: 33.65 on 5 and 15 DF,  p-value: 1.225e-07
```

The slope `c1` is clearly statistically significant (p-value = 3.42-07) while the model explains about 91.8% of the variability of the data. The confidence interval for c1 confirms that an increase of the time leads to a significant increase of the variable y. The residuals are approximately zero.

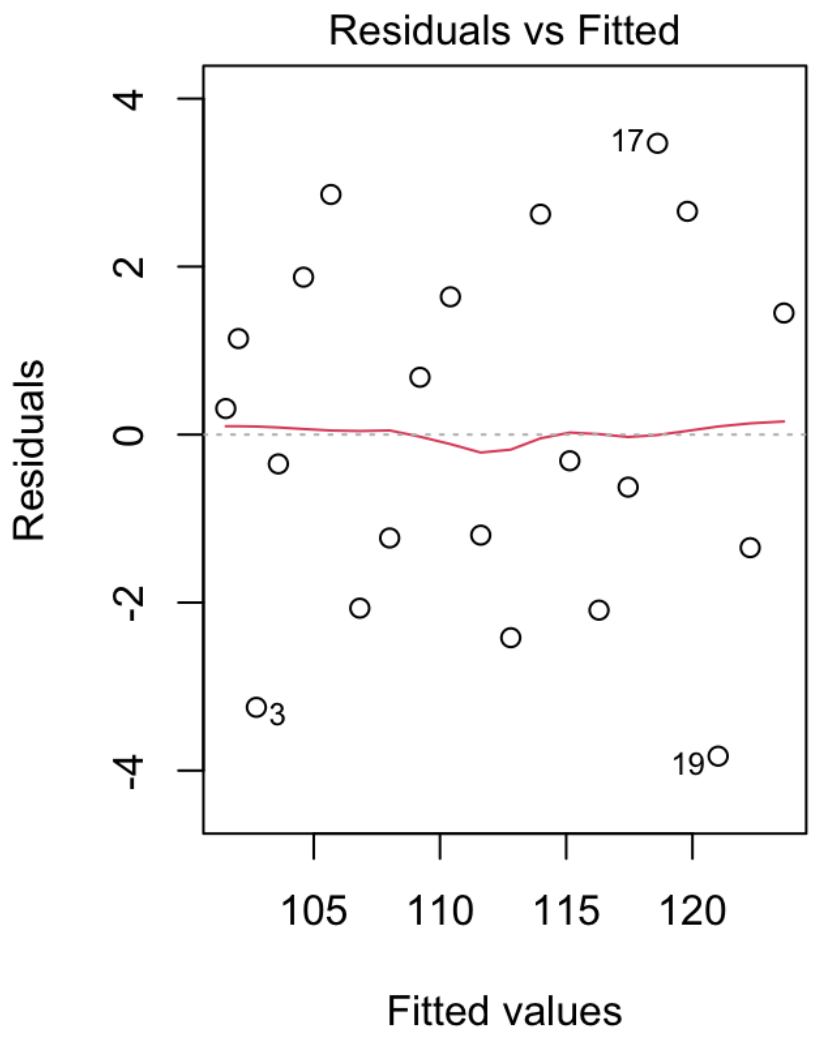
Hide

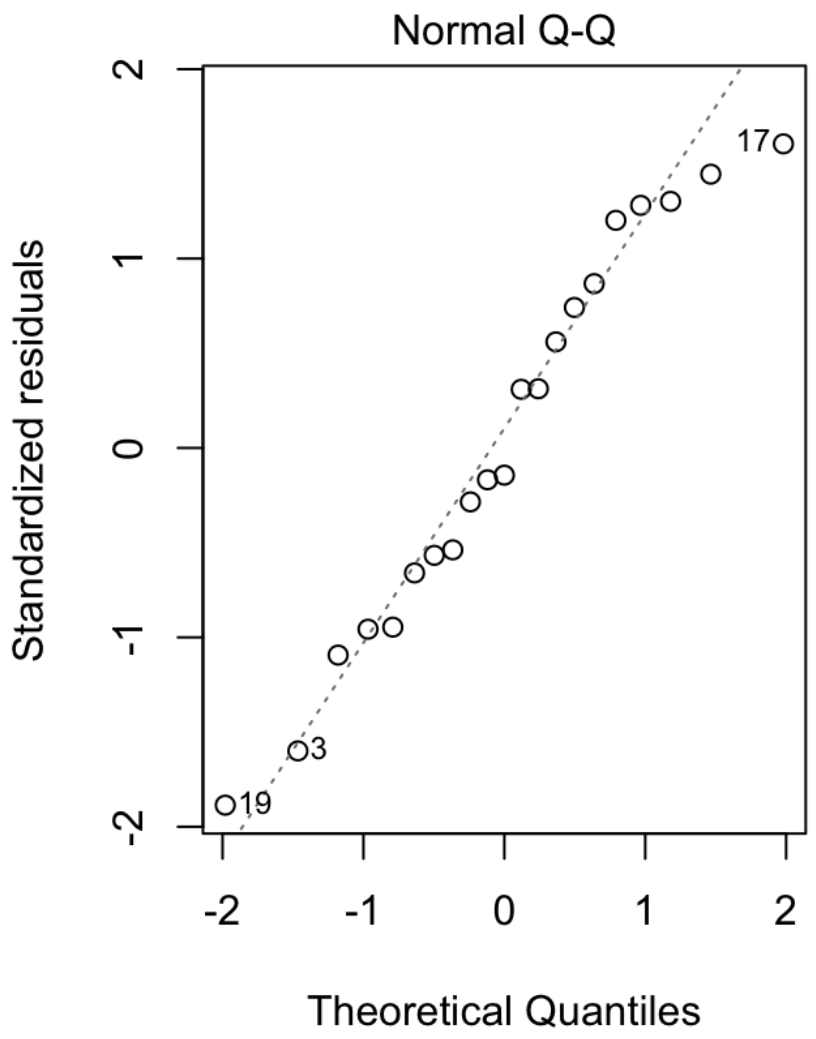
```
pl  + geom_line(aes(x=time, y=predict(lm6)),size=1, colour="#339900")
```



Hide

```
par(mfrow = c(1, 2))
plot(lm6, which=c(1,2))
```

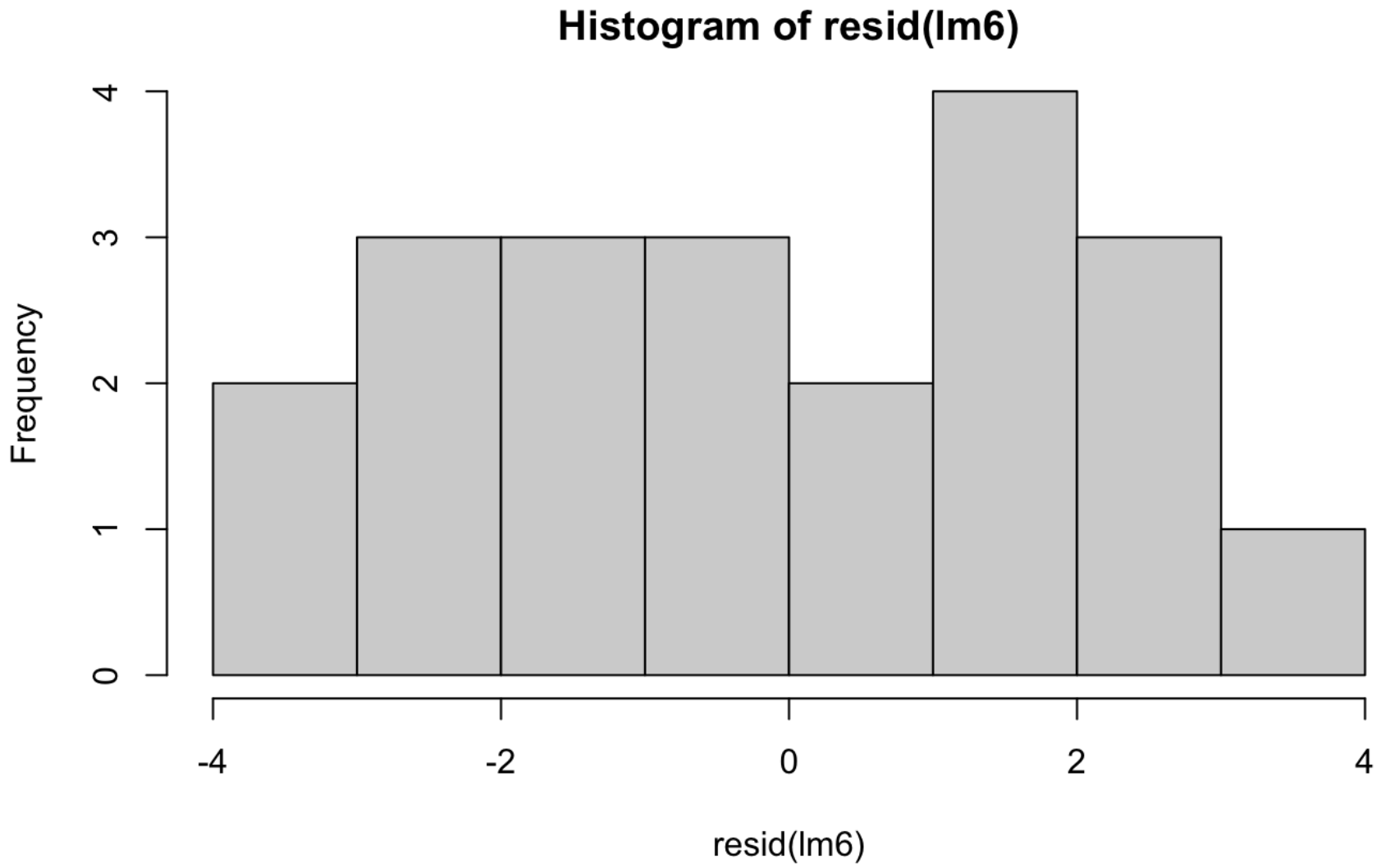




The QQ plot is obtained by plotting the standardized residual. The residual are normally distributed because then the points are randomly distributed around the line y=x. The residual plot shows a linear line which suggests that the residuals are identically distributed around 0. We choose to keep this model with a degree of 5.

Hide

```
hist(resid(lm6))
```



Histogram of the Residuals show that the deviation is not normally distributed.

We use **the Bayesian information criaterion (BIC)** for comparing models which are not necessarily nested.

Hide

```
BIC(lm1,lm2,lm6)
```

	df <dbl>	BIC <dbl>
lm1	3	99.7492
lm2	4	102.4151
lm6	7	111.3245

3 rows

Hide

```
AIC(lm1,lm2,lm6)
```

	df <dbl>	AIC <dbl>
lm1	3	96.61563
lm2	4	98.23700
lm6	7	104.01281

3 rows

Models with lowest BIC and AIC are preferred. Here, both criteria agree for rejecting lm6 with high confidence. Both BIC and AIC has a very slight preference for lm1 and lm2. Nevertheless, these differences are not large enough for selecting definitely any of these 2 models.

1.3 Try to improve the model by adding a periodic component

$cos(2\pi t/T)$ and $sin(2\pi t/T)$ are periodic functions of period T. Looking at the scatter plot we observe a cyclical oscillation of period 1 year with T=12.

$$f(x) = c0 + c1x + c2x2 + c3x3 + c4x4 + c5x5 + e5$$

Hide

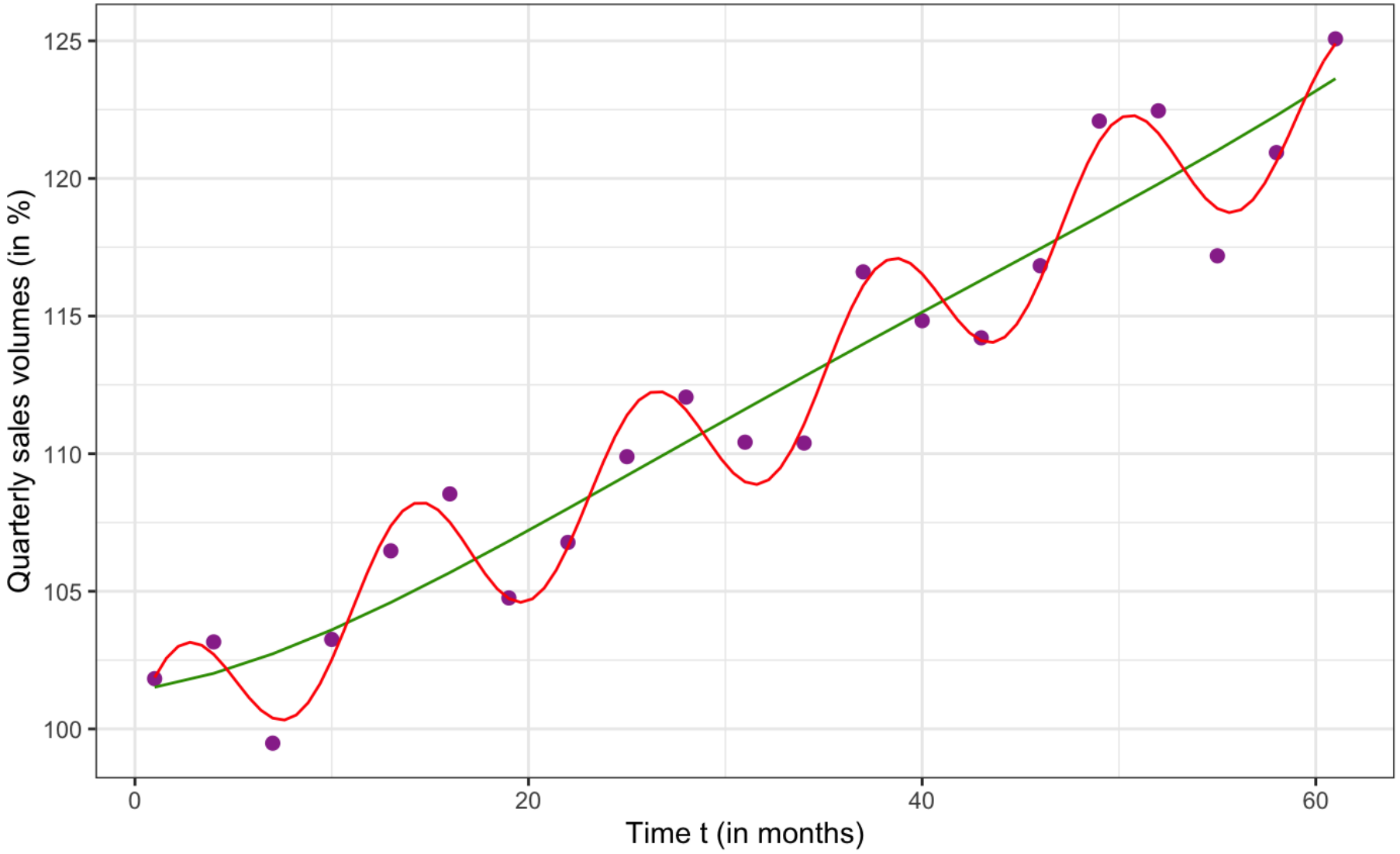
```
T <- 12
mod_per <- lm(y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) +
                I(cos(2*pi*time/T)) + I(sin(2*pi*time/T)), data=data )
summary(mod_per)
```

```
Call:
lm(formula = y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) +
    I(cos(2 * pi * time/T)) + I(sin(2 * pi * time/T)), data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.7175 -0.6859  0.1701  0.5103  1.4439

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             9.889e+01  1.393e+00  70.970  < 2e-16 ***
time                    6.769e-01  4.638e-01   1.460  0.16814
I(time^2)              -2.252e-02  4.673e-02  -0.482  0.63795
I(time^3)               6.043e-04  1.917e-03   0.315  0.75757
I(time^4)              -5.571e-06  3.411e-05  -0.163  0.87278
I(time^5)               7.290e-09  2.189e-07   0.033  0.97394
I(cos(2 * pi * time/T))  1.294e+00  3.570e-01   3.625  0.00308 **
I(sin(2 * pi * time/T))  2.407e+00  3.581e-01   6.723 1.42e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.117 on 13 degrees of freedom
Multiple R-squared:  0.9851,    Adjusted R-squared:  0.9771
F-statistic: 123.1 on 7 and 13 DF,  p-value: 7.433e-11
```

It seems that the most influant variables of our modl are the sine and cosine terms, then the degree 1 polynomial term. Here is the comparison of the previously selected model and the one with the periodic components:

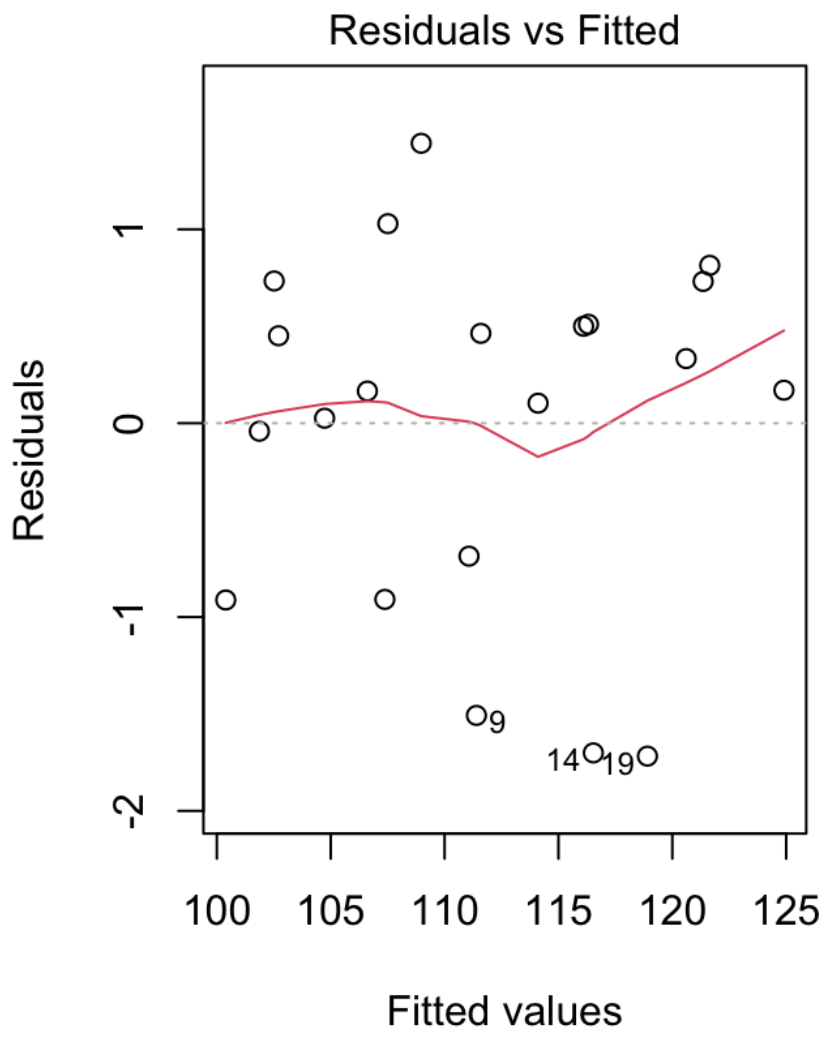
Hide

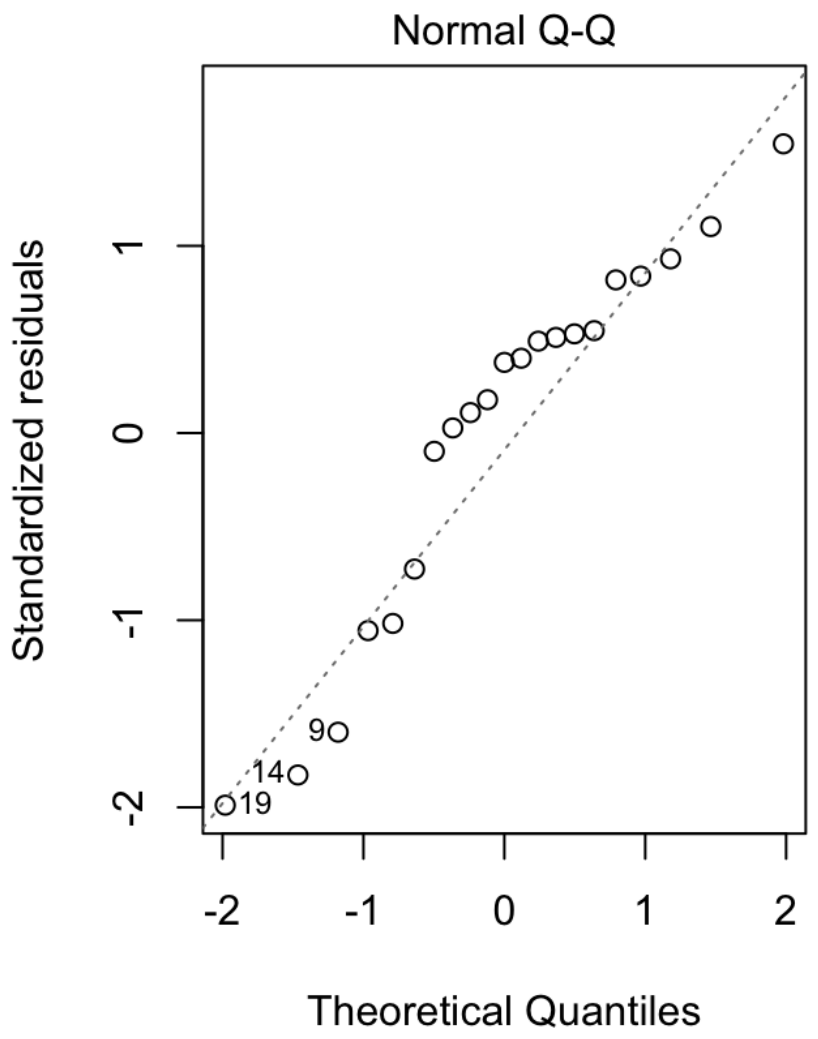
```
f <- function(x,c) coef(mod_per)[1] + coef(mod_per)[2]*x + coef(mod_per)[3]*x^2 + coef(mod_per)[4]*x^3 + coef(mod_per)[5]*x^4 + coef(mod_per)[6]*x^5 + coef(mod_per)[7]*cos(2*pi*x/T) + coef(mod_per)[8]*sin(2*pi*x/T)

pl + geom_line(data=data, aes(x=time, y=predict(poly_reg_5)), size=0.5, colour="#339900") + stat_function(fun=f, colour="red", size=0.5)
```



Hide

```
par(mfrow = c(1, 2))
plot(mod_per, which=c(1,2))
```

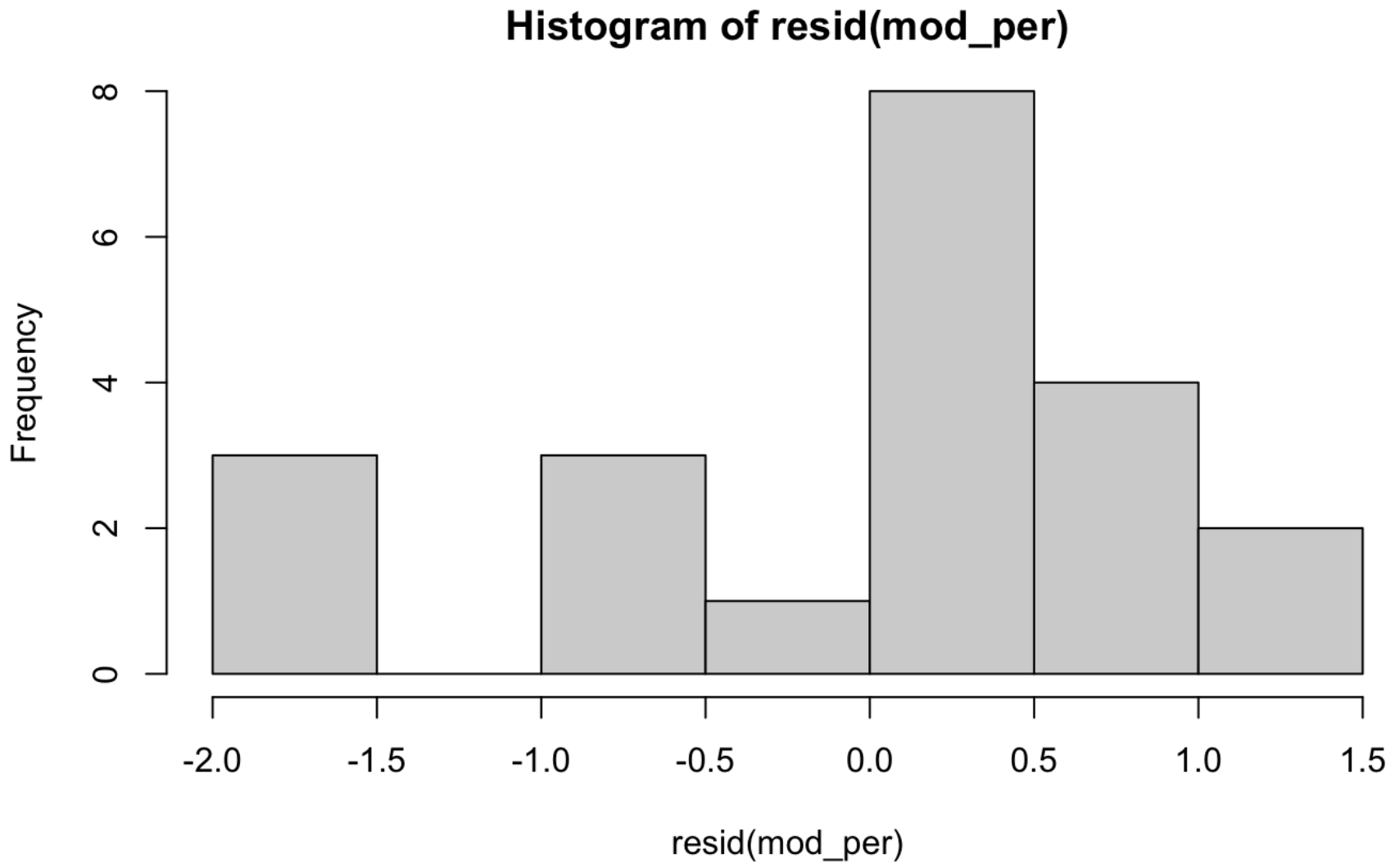




The residual plot shows a slight decreasing then increasing trend which suggests that the residuals are not identically distributed around 0 and that linearity is violated. Furthermore, the Quantile-Quantile plot shows that the extreme residual values are not the extreme values of a normal distribution.

Hide

```
hist(resid(mod_per))
```



Histogram of the Residuals show that the deviation is not normally distributed.

###1.4 Plot on a same graph the observed sales together with the predicted sales given by your final model. What do you think about this model? What about the residuals?

A common practice is to split the dataset into a 80:20 sample (training:test), then, build the model on the 80% sample and then use the model thus built to predict the reponse variable on test data. Doing it this way, we will have the model predicted values for the 20% data (test). We can then see how the model will perform with this ``new'' data, by comparing these predicted values with the original ones. We can alo check the stability of the prediction given by the model, by comparing these predicted values with those obtained previouly, when the complete data were used for building the model. Let us first randomly define the training and test samples:

Hide

```
set.seed(100)
n <- nrow(data)
i.training <- sort(sample(n,round(n*0.8)))
data.training <- data[i.training,]
data.test <- data[-i.training,]

pred1a.test <- predict(lm1, newdata=data.test)
lm6.training <- lm(y ~ time, data=data.training)
pred1b.test <- predict(lm6.training, newdata=data.test)


data.frame(data.test, pred1a.test, pred1b.test)
```

	time <int>	y <dbl>	pred1a.test <dbl>	pred1b.test <dbl>
1	1	101.8256	100.3767	99.74531
5	13	106.4687	104.9350	104.44600
11	31	110.4174	111.7723	111.49703
13	37	116.6042	114.0514	113.84738

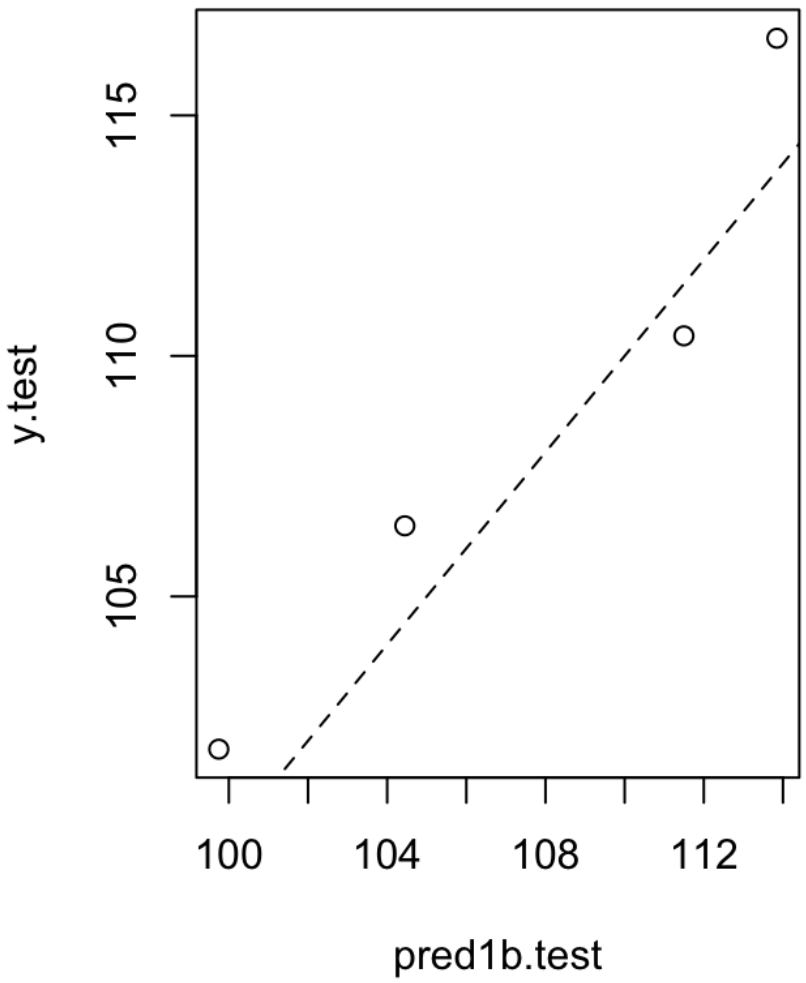
4 rows

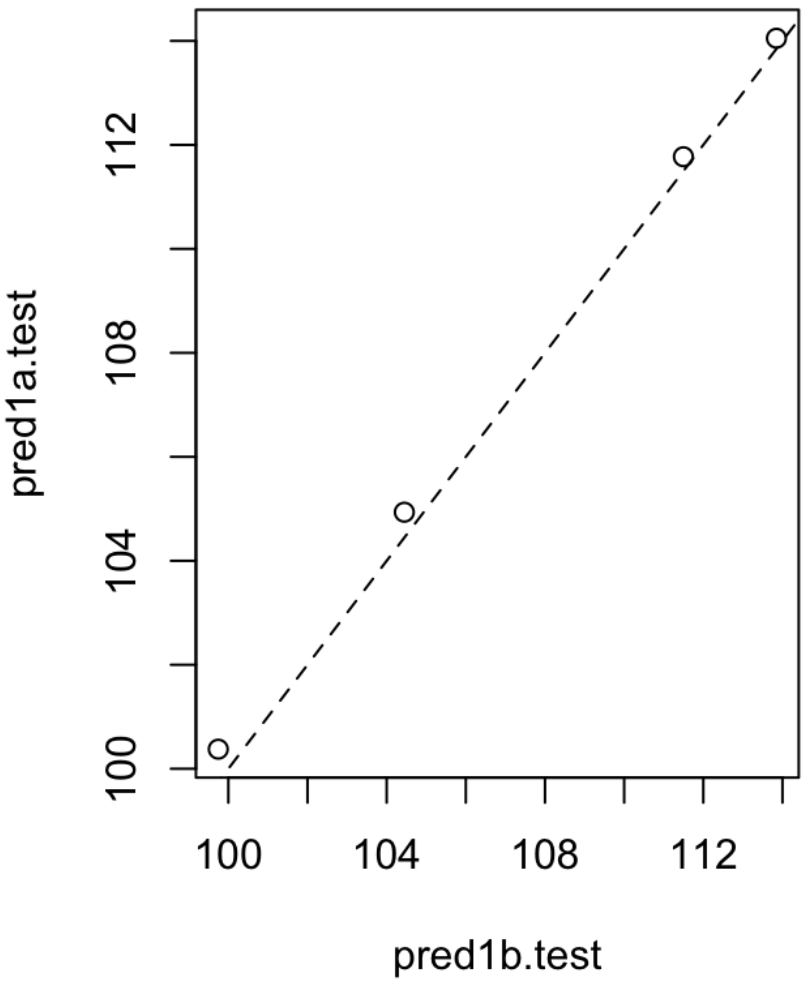
Hide

```
y.test <- data.test$y
par(mfrow=c(1,2))
plot(pred1b.test, y.test)
abline(a=0, b=1, lty=2)
```

Hide

```
plot(pred1b.test, pred1a.test)
abline(a=0, b=1, lty=2)
```





On one hand, it is reassuring to see that removing part of the data has a very little impact on the predictions (right graph). On the other hand, the predictive performance of the model remains limited because of the natural variability of the data (left graph).

Hide

```
cor.test <- cor(pred1a.test, y.test)
R2.test <- cor.test^2
R2.test
```

```
[1] 0.9297326
```

Indeed, this model built with the training sample explains 92. % of the variability of the new test sample.

Confidence intervale and prediction intervale:

Hide

```
alpha <- 0.05
df.new <- data.frame(time=(0:60))
conf.weight <- predict(lm2, newdata = df.new, interval="confidence", level=1-alpha)
```

```
Error in eval(predvars, data, env) : object 'id' not found
```

A prediction interval for a new measured distance $y = f(x) + e$ can also be computed. This prediction interval takes into account both the uncertainty on the predicted distance f(x) and the variability of the measure, represented in the model by the residual error e.

Hide

```
pred.weight <- predict(lm2, newdata = df.new, interval="prediction", level=1-alpha)
```

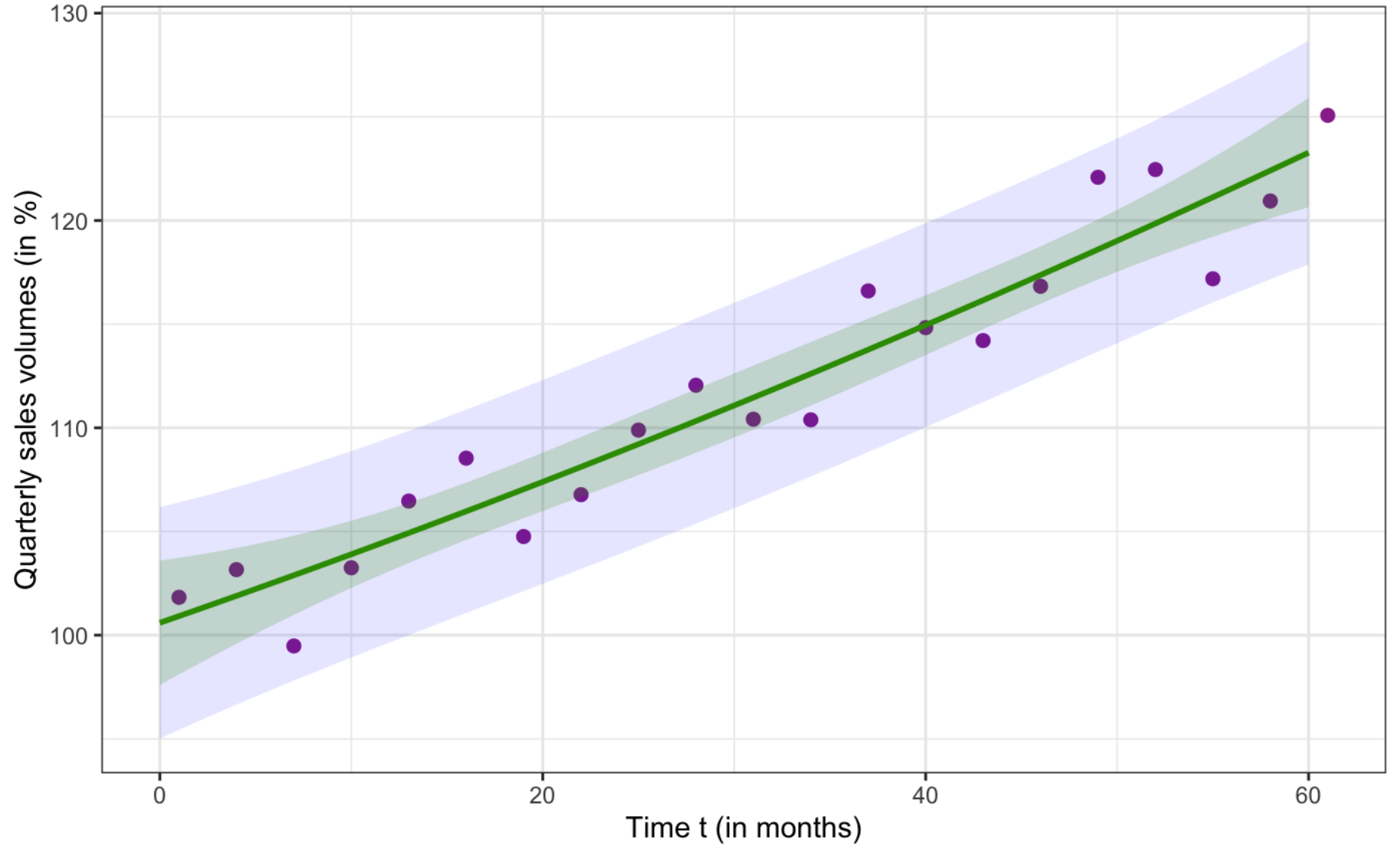
```
Error in eval(predvars, data, env) : object 'id' not found
```

Let us plot these two intervals.

Hide

```
df.new[c("fit","lwr.conf", "upr.conf")] <- conf.weight
df.new[c("lwr.pred", "upr.pred")] <- pred.weight[,2:3]
pl +
  geom_ribbon(data=df.new, aes(x=time, ymin=lwr.pred, ymax=upr.pred), alpha=0.1, inherit.aes=F, fill="blue") +
```

```
  geom_ribbon(data=df.new, aes(x=time, ymin=lwr.conf, ymax=upr.conf), alpha=0.2, inherit.aes=F, fill="#339900") +
  geom_line(data=df.new, aes(x=time, y=fit), colour="#339900", size=1)
```



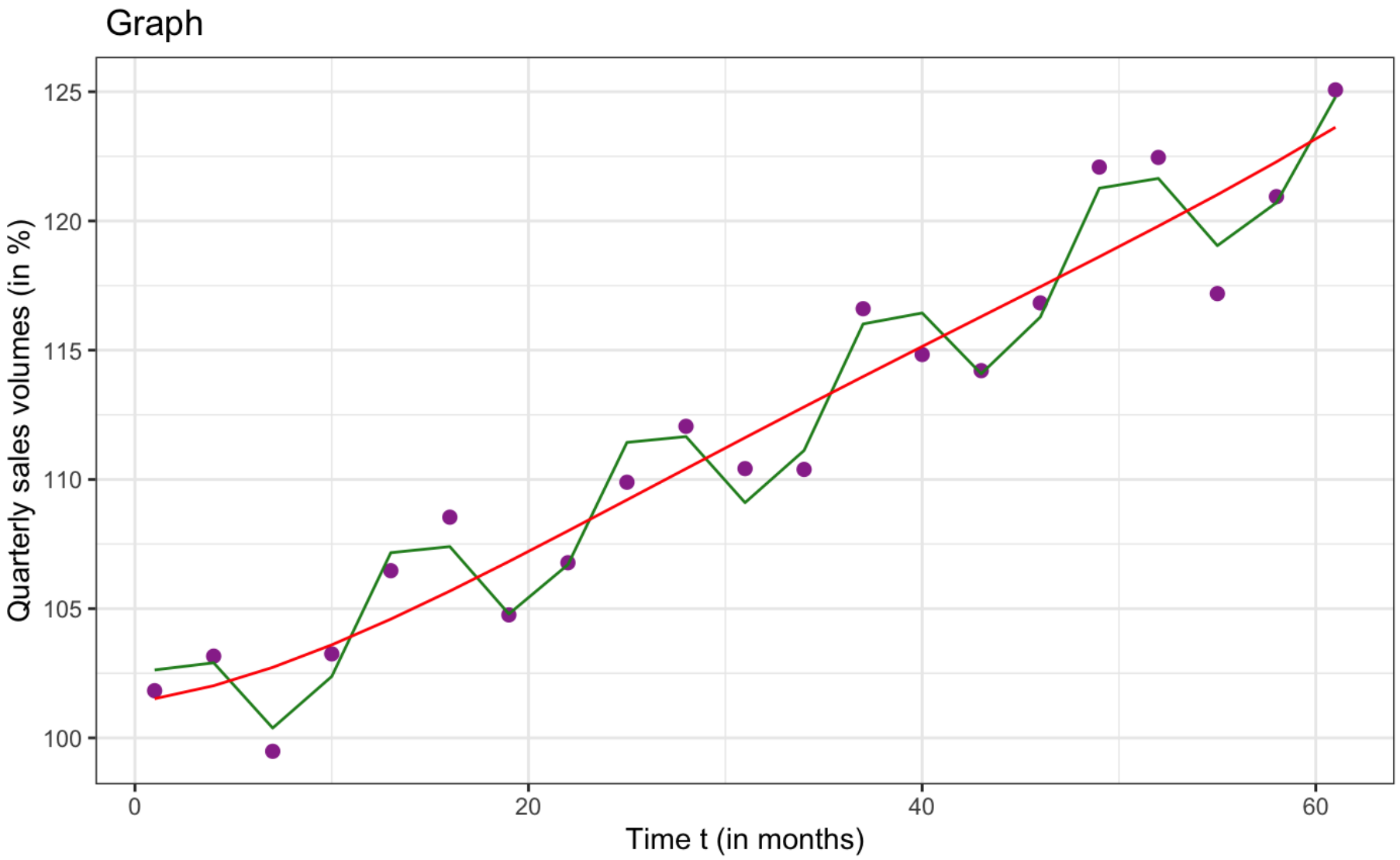
Increasing predicted quaterly sales volume going on.

1.5 We want the predicted sales volume to be equal to 100 at time 0. Modify your final model in order to take this constraint into account.

The sales volume predicted by the model should be 100 at time 0. This constraint can easily be achieved by fixing the intercept of the regression model to 100. On the following graph we see in greeen our final model with the constraint taken into account, and in red the previous model that did not take the constraint into account:

Hide

```
intercept <- 100
poly_reg_100int <- lm(y ~ 0 + time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2*pi*time/T)) + I(sin(
2*pi*time/T)), offset=rep(intercept, length(time)),data=data)

pl + geom_line(data=data, aes(x=time, y=predict(poly_reg_100int)), size=0.5, colour="forestgreen") + geom_line(da
ta=data, aes(x=time, y=predict(poly_reg_5)), size=0.5, colour="red") + ggtitle(" Graph")
```



Hide

```
print(coef(poly_reg_100int))
```

```
                  time               I(time^2)               I(time^3)               I(time^4)                I(t
ime^5) I(cos(2 * pi * time/T)) I(sin(2 * pi * time/T))
           3.605626e-01            4.606847e-03           -3.730297e-04            1.005078e-05           -8.4180
92e-08            1.255295e+00            2.351996e+00
```

2 Fitting a linear mixed effects model

The file sales30.csv now consists of quarterly sales volumes (still in % and indexed to the time 0) of 30 different products.

Hide

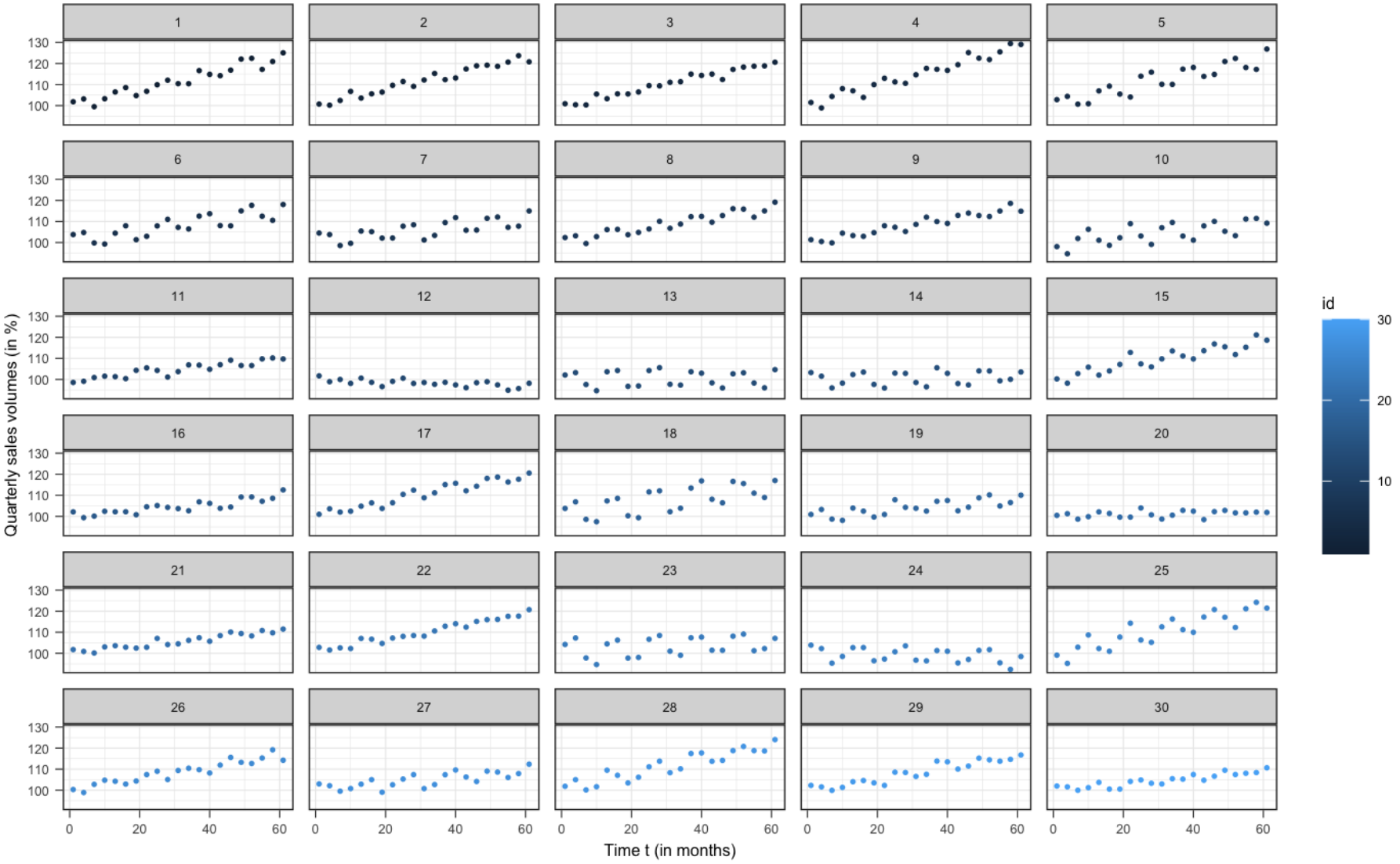
```
sales30 <- read.csv("/Users/haliouanaomie/PolytechniqueS2/MAP566/salesData/sales30.csv")
head(sales30)
summary(sales30)
dim(sales30)
```

The data consists of 630 observations (rows) and 3 variables (columns): time in months, sales volume y in percentage, product identifier id There is one instance per quarter (3 months intervals) and 30 different products in total. Let's scatter plot the data in order to better visualize the relationship between the explanatory variable and the response variable. ### 2.1 Plot this data

Let us plot the data, i.e. the time versus y by id, We plot 30 graphs, each one corresponding to one of the 30 differents products:

Hide

```
library(ggplot2)
options(repr.plot.width=4, repr.plot.height=3)
pl <- ggplot(data=data30, aes(x=time, y=y, color=id)) + geom_point(size=0.3) + facet_wrap(~id, nrow=6, ncol=5) +
 xlab("Time t (in months)") + ylab("Quarterly sales volumes (in %)") + theme(text=element_text(size=6), element_l
ine(size=0.2))
pl
```



We observe on the above scatter plots that for most of the products, there is a clear increasing trend in the data as it was the case for the sales1.csv dataset. However, some of the products do not share this pattern. For example, products 13, 14, 20, and 23 seem to have a rather constant periodic trend. Products 12 and 24 on the other hand seem to have a slightly decreasing trend over the time. For each product, it seems that the sale volume follows a periodical pattern as for the previous exercise with a same period of 1 year but different magnitudes.

2.2 Fit the model used previously for fitting the first series to this data and comment the results

A linear model by definition assumes there is a linear relationship between the observations (yj,1≤j≤n) and m series of variables $(x(1)j,\ldots,x(m)j,1\leq j\leq n)$: $yj=c0+c1x(1)j+c2x(2)j+\cdots+cmx(m)j+ej,1\leq j\leq n,$ where $(ej,1\leq j\leq n)$ is a sequence of residual errors. In our example, the observations $(yj,1\leq j\leq n)$ are the n=630 measured distances.

We are fitting the second model used previously.

Hide

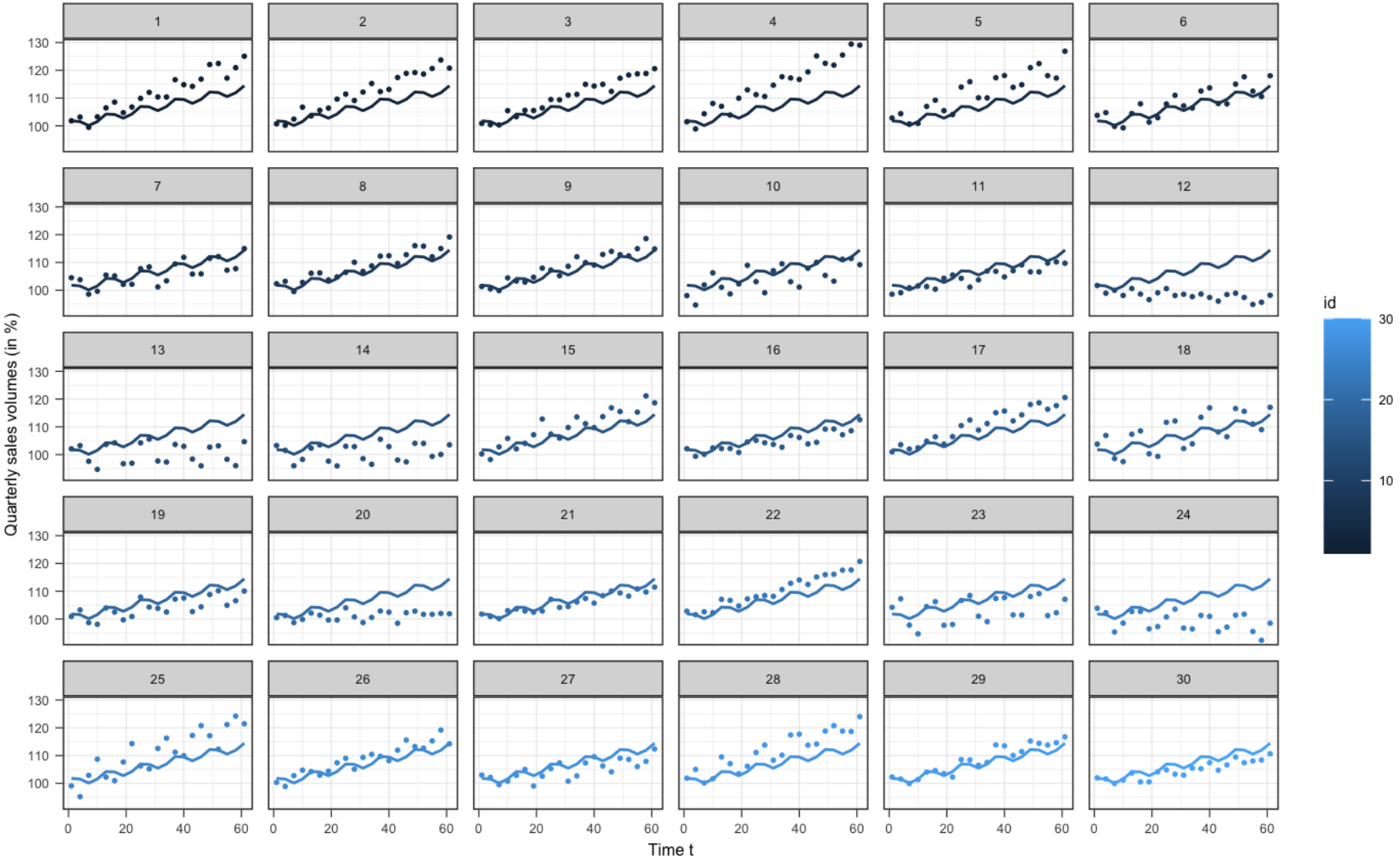
```
model30 <- lm(y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2*pi*time/T)) + I(sin(2*pi*time/T
)), data=data30)
lm2 <- lm(y~time+id, data=sales30)
summary(model30)
```

```

Call:
lm(formula = y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) +
    I(cos(2 * pi * time/T)) + I(sin(2 * pi * time/T)), data = data30)

Residuals:
     Min       1Q   Median       3Q      Max
-19.5795  -2.9541   0.0852   3.3104  17.5351

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)              1.002e+02  1.224e+00  81.864  < 2e-16 ***
time                     1.635e-01  4.074e-01   0.401   0.6883
I(time^2)                4.244e-03  4.106e-02   0.103   0.9177
I(time^3)               -1.295e-04  1.684e-03  -0.077   0.9387
I(time^4)                1.844e-06  2.996e-05   0.062   0.9510
I(time^5)               -1.073e-08  1.923e-07  -0.056   0.9555
I(cos(2 * pi * time/T))  9.138e-01  3.136e-01   2.914   0.0037 **
I(sin(2 * pi * time/T))  1.237e+00  3.146e-01   3.931 9.41e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 5.377 on 622 degrees of freedom
Multiple R-squared:  0.3636,    Adjusted R-squared:  0.3564
F-statistic: 50.76 on 7 and 622 DF,  p-value: < 2.2e-16
```

Hide

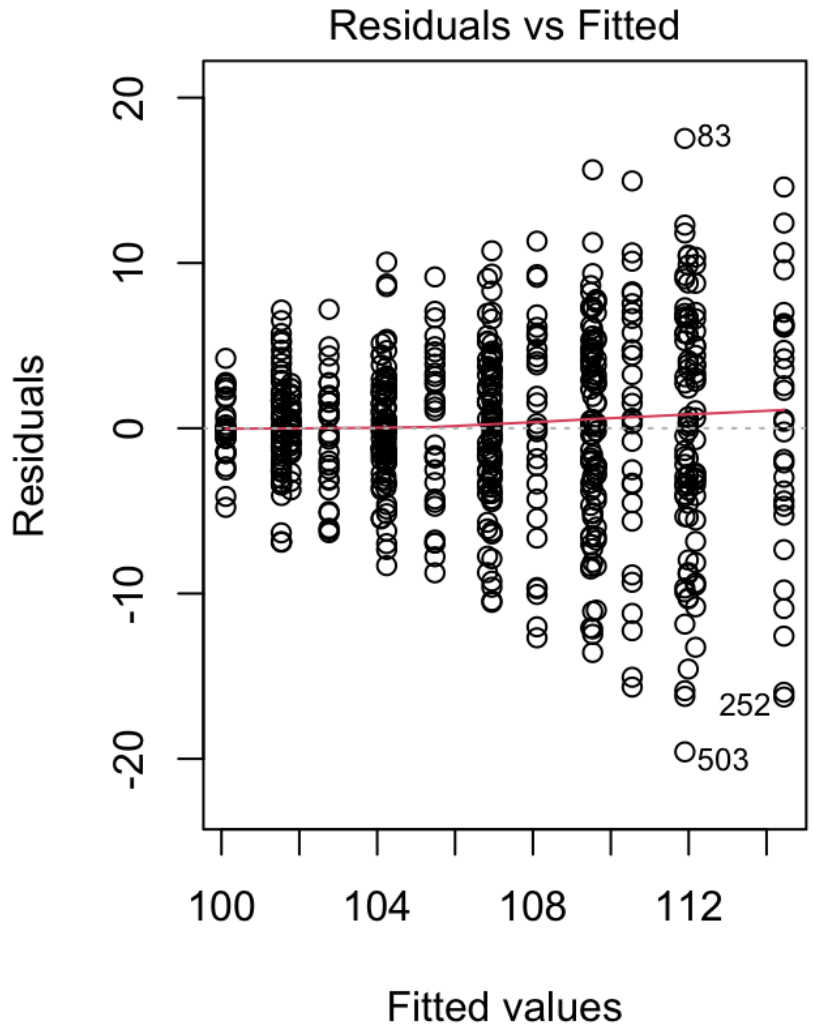
```
sales30$pred.lm2 <- predict(model30)
pl + geom_line(data=sales30,aes(x=time,y=pred.lm2)) + facet_wrap(~id) + xlab("Time t") + ylab("Quarterly sales vo
lumes (in %)")
```

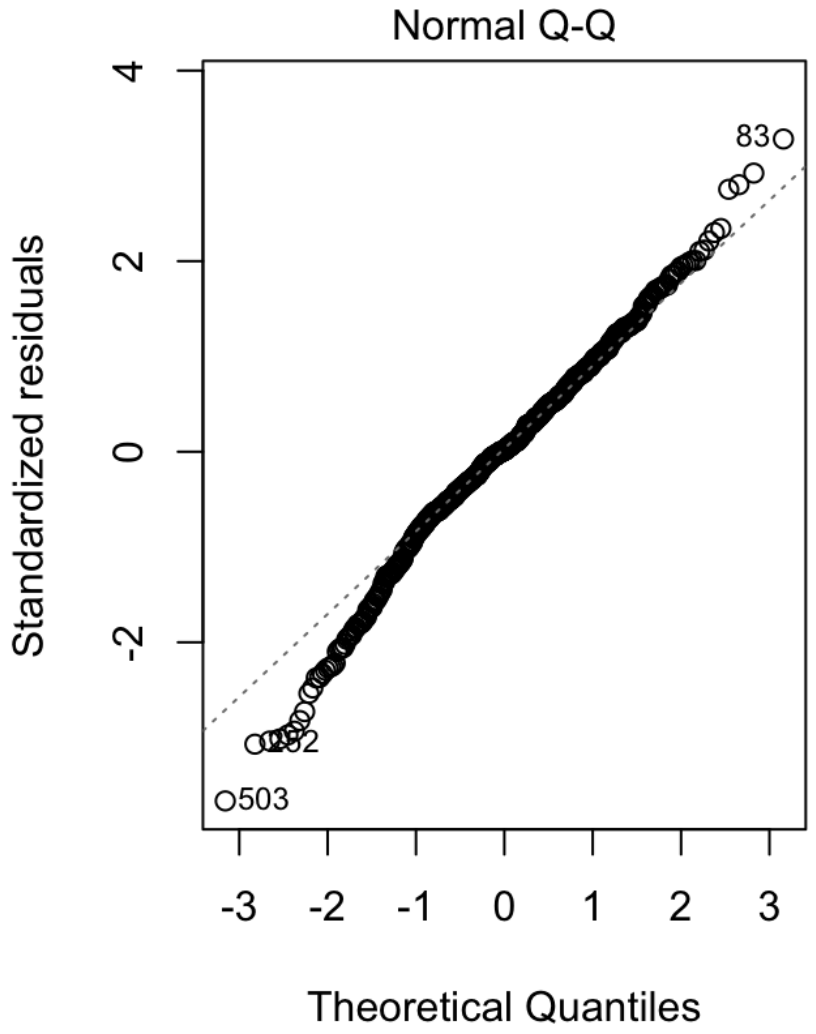


We observe that our model trained on the whole set clearly underestimate and overestimate the sales volume of most products. Intuitively we understand that each product being different, the id variable has an importance in our prediction and that for example, by splitting the data set into 30 data sets corresponding to the 30 products, we could train our models on the different products independently. However this is not what we want to do, otherwise we would have as many models as products, we have to consider the identifier directly into the model!

Hide

```
par(mfrow = c(1, 2))
plot(model30, which=c(1,2))
```

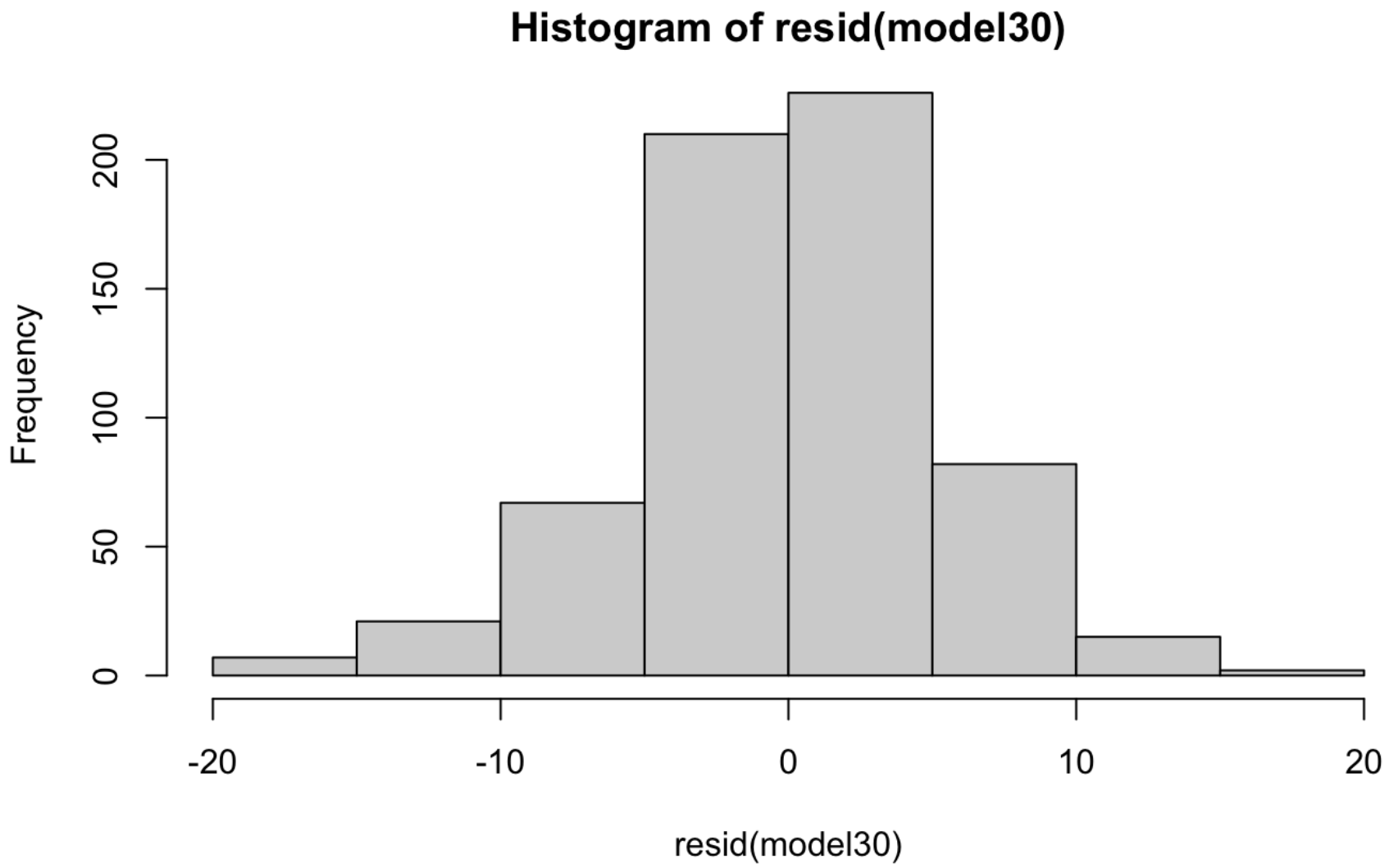




The residual plot don’t shows a slight (decreasing and increasing) trend which suggests that the residuals are identically distributed around 0. Furthermore, the QQ plot shows that the extreme residual values are not the extreme values of a normal distribution. This is due to the fact that several ids are involved and not just one id this time.

Hide

```
hist(resid(model30))
```



Histogram of the Residuals show that the deviation is normally distributed

2.3 Fit a mixed effect model to this data

The model is called linear mixed effects model because it is a linear combination of fixed and random effects. We can use the function lmer for fitting this model. By default, the restricted mximum likelihood (REML) method is used.

Hide

```
library(lme4)
library(Matrix)
lme1 <- lmer(y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2*pi*time/T)) + I(sin(2*pi*time/T))
+ (1|id), data=data30)
```

```
Some predictor variables are on very different scales: consider rescaling
```

Hide

```
summary(lme1)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2 *      pi * time/T)) + I(sin(2 * pi *
time/T)) + (1 | id)
   Data: data30

REML criterion at convergence: 3531.4

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.5111 -0.6554 -0.0017  0.6294  2.9194

Random effects:
 Groups   Name        Variance Std.Dev.
 id       (Intercept) 17.55    4.190
 Residual             11.72    3.424
Number of obs: 630, groups:  id, 30

Fixed effects:
                         Estimate Std. Error t value
(Intercept)             1.002e+02  1.092e+00  91.757
```

```
time                     1.635e-01  2.594e-01   0.630
I(time^2)                4.244e-03  2.614e-02   0.162
I(time^3)               -1.295e-04  1.072e-03  -0.121
I(time^4)                1.844e-06  1.908e-05   0.097
I(time^5)               -1.073e-08  1.224e-07  -0.088
I(cos(2 * pi * time/T))  9.138e-01  1.997e-01   4.575
I(sin(2 * pi * time/T))  1.237e+00  2.003e-01   6.173

Correlation of Fixed Effects:
             (Intr) time   I(t^2) I(t^3) I(t^4) I(t^5) I(c(2*p*t/T))
time         -0.610
I(time^2)     0.519 -0.967
I(time^3)    -0.456  0.912 -0.985
I(time^4)     0.409 -0.859  0.956 -0.992
I(time^5)    -0.374  0.811 -0.923  0.974 -0.995
I(c(2*p*t/T)) -0.098  0.112 -0.076  0.043 -0.015 -0.010
I(s(2*p*t/T)) -0.138  0.124 -0.079  0.050 -0.031  0.017 -0.005
fit warnings:
Some predictor variables are on very different scales: consider rescaling
```

The model summary doesn't show you all of these; it only tells you what the variance (and standard deviation) of each group of coefficients is. It's still possible to find out what the coefficients for the random effects of subject is by calling the function ranef on the model:

Hide

```
ranef(lme1)
```

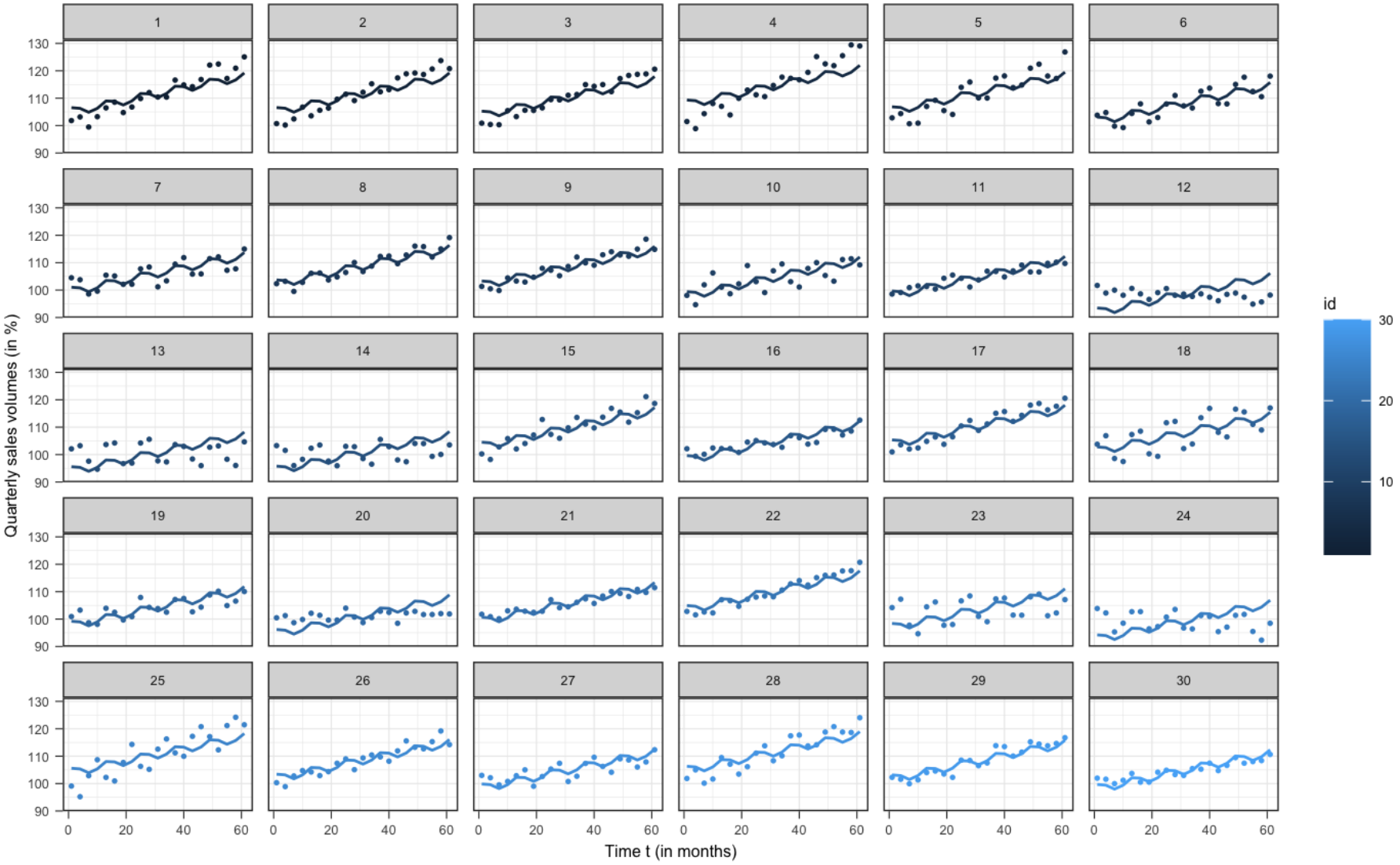
```
$id
   (Intercept)
1    4.7483695
2    4.7839623
3    3.4681206
4    7.5397770
5    5.0702156
6    1.3088463
7   -0.7232037
8    1.9237695
```

```
9    1.5355026
10  -2.3652589
11  -2.1228331
12  -8.3125588
13  -6.2111064
14  -6.0045977
15   2.7334443
16  -2.1649437
17   3.5755827
18   1.0010861
19  -2.5688157
20  -5.5956368
21  -1.0973061
22   3.1440067
23  -3.3768390
24  -7.5581764
25   3.8070548
26   1.6260310
27  -1.9388348
28   4.5213757
29   1.3772704
30  -2.1243041

with conditional variances for “id”
```

We can see, for example, that subjects number 24 responded exceptionally slowly and subjects number 4, very quickly.

Hide

```
pl + geom_line(aes(x=time,y=predict(lme1),color=id)) + facet_wrap(~id)
```



Hide

```
lme2 <- lmer(y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2*pi*time/T)) + I(sin(2*pi*time/T))
+ (-1+time|id), data=data30)
```

```
Some predictor variables are on very different scales: consider rescaling
```

Hide

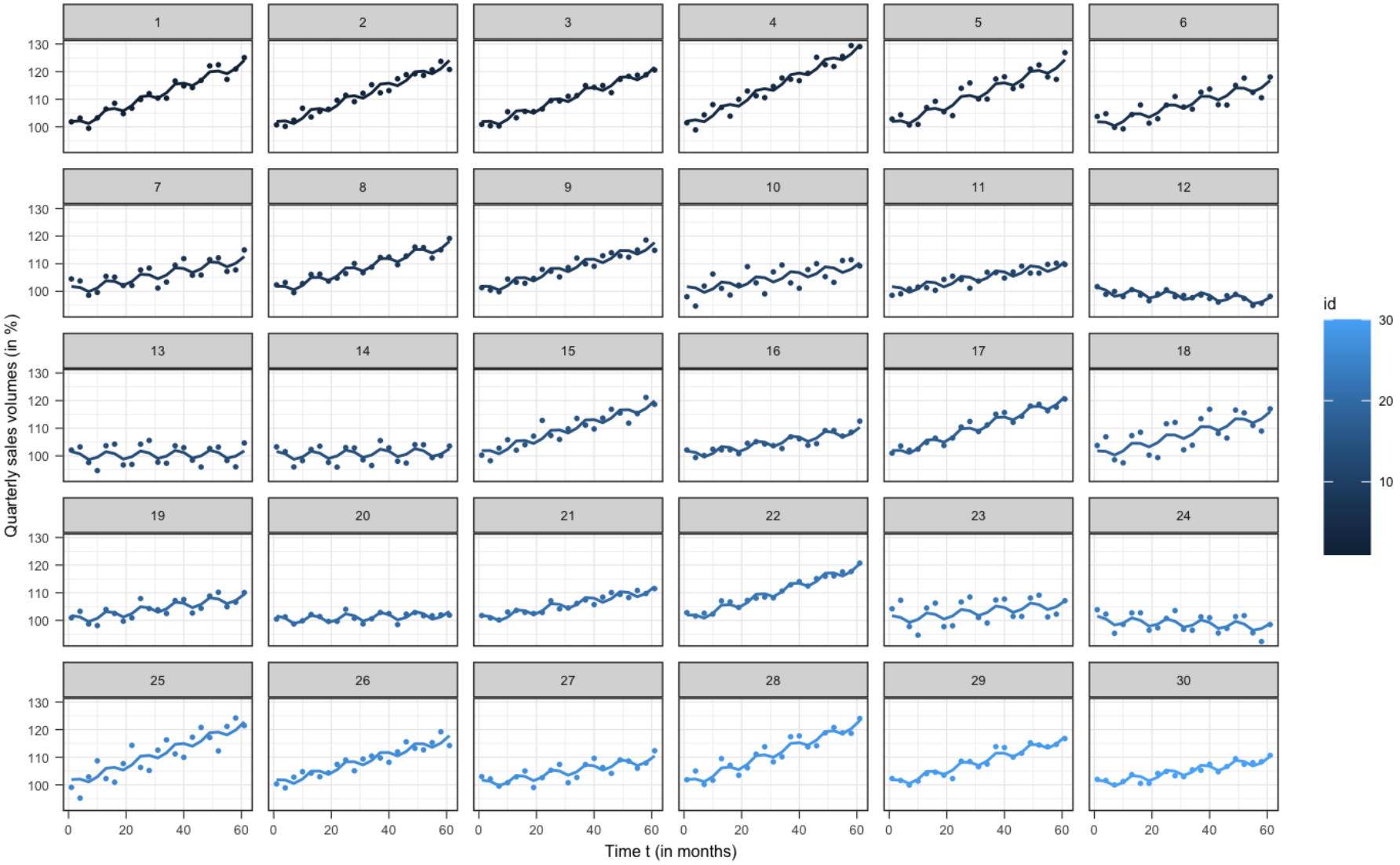
```
summary(lme2)
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2 *      pi * time/T)) + I(sin(2 * pi *
time/T)) + (-1 + time | id)
   Data: data30

REML criterion at convergence: 3064.3

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.99876 -0.61197 -0.03005  0.66933  3.07287

Random effects:
 Groups   Name Variance Std.Dev.
 id       time 0.01871  0.1368
 Residual      5.25477  2.2923
Number of obs: 630, groups:  id, 30

Fixed effects:
                          Estimate Std. Error t value
(Intercept)              1.002e+02  5.219e-01 192.015
time                     1.635e-01  1.755e-01   0.932
I(time^2)                4.244e-03  1.750e-02   0.242
I(time^3)               -1.295e-04  7.179e-04  -0.180
I(time^4)                1.844e-06  1.277e-05   0.144
I(time^5)               -1.073e-08  8.198e-08  -0.131
I(cos(2 * pi * time/T))  9.138e-01  1.337e-01   6.834
I(sin(2 * pi * time/T))  1.237e+00  1.341e-01   9.220

Correlation of Fixed Effects:
             (Intr) time   I(t^2) I(t^3) I(t^4) I(t^5) I(c(2*p*t/T))
```

```
time          -0.846
I(time^2)      0.727 -0.957
I(time^3)     -0.639  0.903 -0.985
I(time^4)      0.574 -0.850  0.956 -0.992
I(time^5)     -0.523  0.802 -0.923  0.974 -0.995
I(c(2*p*t/T)) -0.137  0.111 -0.076  0.043 -0.015 -0.010
I(s(2*p*t/T)) -0.193  0.123 -0.079  0.050 -0.031  0.017 -0.005
fit warnings:
Some predictor variables are on very different scales: consider rescaling
```

Hide

```
pl + geom_line(aes(x=time,y=predict(lme2),color=id)) + facet_wrap(~id)
```



Hide

```
lme3 <- lmer(y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2*pi*time/T)) + I(sin(2*pi*time/T))
+ (time|id), data=sales30)
```

```
Some predictor variables are on very different scales: consider rescalingboundary (singular) fit: see ?isSingular
```

Hide

```
summary(lme3)
```

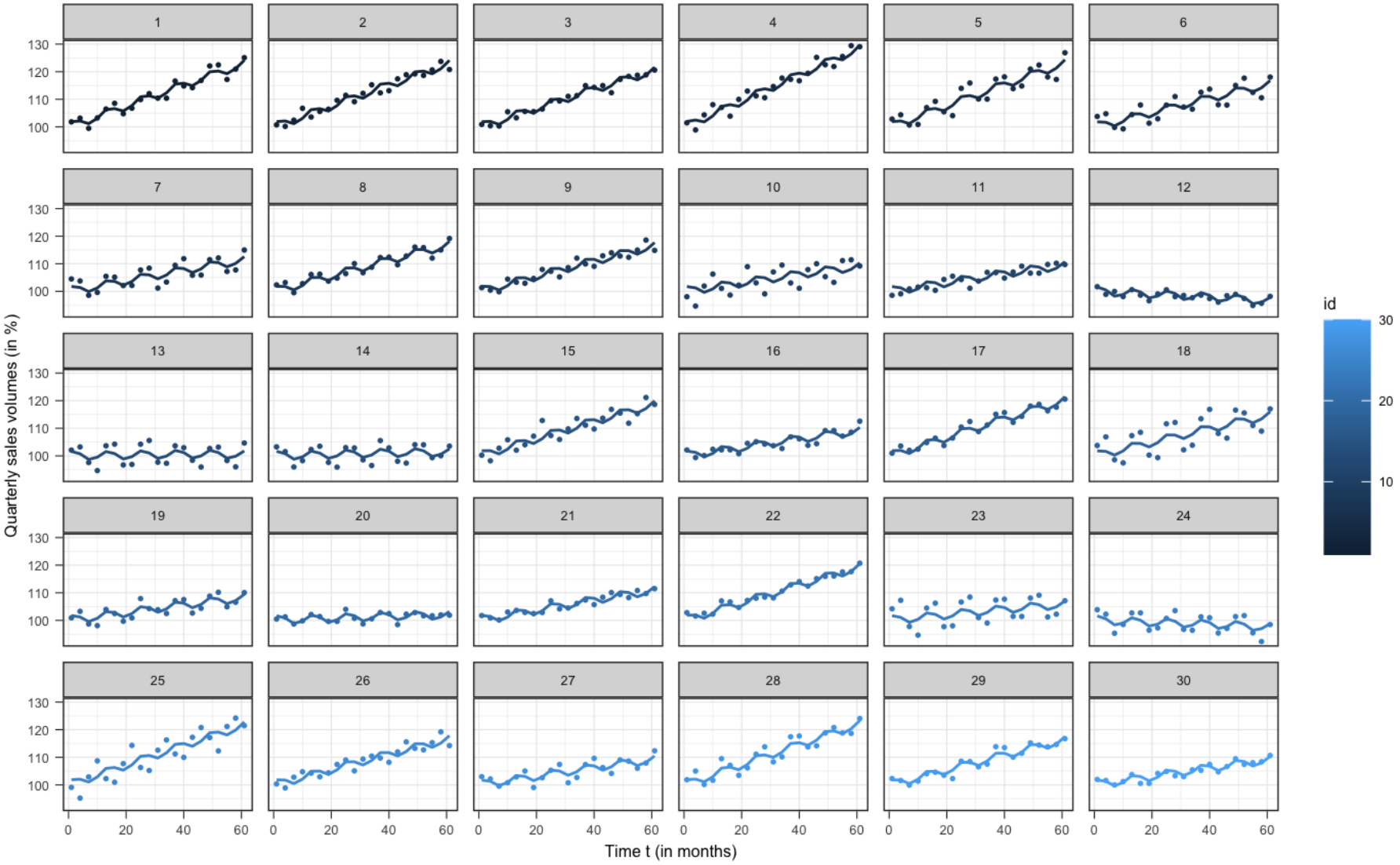
```
Linear mixed model fit by REML ['lmerMod']
Formula: y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2 *      pi * time/T)) + I(sin(2 * pi *
time/T)) + (time | id)
   Data: sales30

REML criterion at convergence: 3064.2

Scaled residuals:
     Min       1Q   Median       3Q      Max
-2.98386 -0.60973 -0.02753  0.67432  3.08017

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 id       (Intercept) 0.001473 0.03838
          time        0.018967 0.13772  -1.00
 Residual             5.254375 2.29224
Number of obs: 630, groups:  id, 30

Fixed effects:
                          Estimate Std. Error t value
(Intercept)              1.002e+02  5.219e-01 192.005
time                     1.635e-01  1.755e-01   0.932
I(time^2)                4.244e-03  1.750e-02   0.242
I(time^3)               -1.295e-04  7.179e-04  -0.180
I(time^4)                1.844e-06  1.277e-05   0.144
I(time^5)               -1.073e-08  8.198e-08  -0.131
I(cos(2 * pi * time/T))  9.138e-01  1.337e-01   6.834
I(sin(2 * pi * time/T))  1.237e+00  1.341e-01   9.221

Correlation of Fixed Effects:
```

```
              (Intr) time   I(t^2) I(t^3) I(t^4) I(t^5) I(c(2*p*t/T))
time          -0.847
I(time^2)      0.727 -0.957
I(time^3)     -0.639  0.903 -0.985
I(time^4)      0.574 -0.850  0.956 -0.992
I(time^5)     -0.523  0.802 -0.923  0.974 -0.995
I(c(2*p*t/T)) -0.137  0.111 -0.076  0.043 -0.015 -0.010
I(s(2*p*t/T)) -0.193  0.123 -0.079  0.050 -0.031  0.017 -0.005
fit warnings:
Some predictor variables are on very different scales: consider rescaling
optimizer (nloptwrap) convergence code: 0 (OK)
boundary (singular) fit: see ?isSingular
```

The warning just indicates that one or more variances are (very close to) zero.

Hide

```
pl + geom_line(aes(x=time,y=predict(lme3),color=id)) + facet_wrap(~id)
```



Hide

```
BIC(lme1,lme2,lme3)
```

	df <dbl>	BIC <dbl>
lme1	10	3595.899
lme2	10	3128.750
lme3	12	3141.598

3 rows

Hide

```
AIC(lme1,lme2,lme3)
```

	df <dbl>	AIC <dbl>
lme1	10	3551.442
lme2	10	3084.293
lme3	12	3088.250

3 rows

The best model, according to BIC, is model lme2 that assumes different fixed slopes for different ids and a random intercept.

We can compute 95% profile-based confidence intervals for the parameters of the model:

Hide

```
confint(lme2)
```

```
Computing profile confidence intervals ...
```

```
                            2.5 %       97.5 %
.sig01                  1.061553e-01 1.776786e-01
```

```
.sigma                   2.155814e+00 2.414198e+00
(Intercept)              9.918968e+01 1.012267e+02
time                    -1.788676e-01 5.059060e-01
I(time^2)               -2.991789e-02 3.840545e-02
I(time^3)               -1.530597e-03 1.271552e-03
I(time^4)               -2.308809e-05 2.677560e-05
I(time^5)               -1.707355e-07 1.492656e-07
I(cos(2 * pi * time/T))  6.528538e-01 1.174804e+00
I(sin(2 * pi * time/T))  9.748236e-01 1.498297e+00
```

These numbers refer to how percentage of the data is bellow of these limits. So, we have 2.5% of the value bellow 99.85 and 97.5% of the value bellow 100.63.

Parametric bootstrap can also be used for computing confidence intervals:

Hide

```
confint(lme2,method="boot")
```

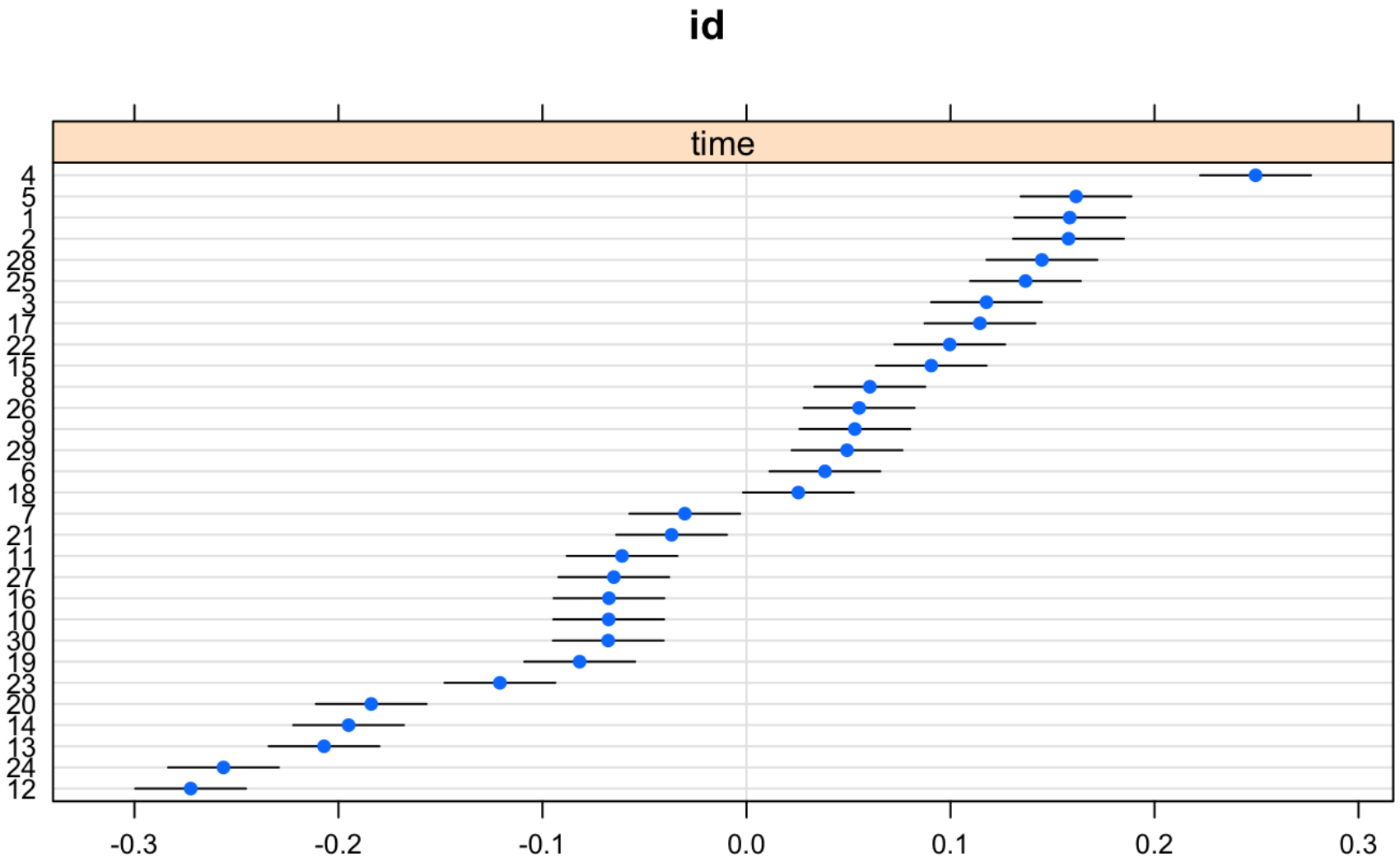
```
Computing bootstrap confidence intervals ...
```

```
                            2.5 %       97.5 %
.sig01                  9.871853e-02 1.752095e-01
.sigma                  2.152193e+00 2.424236e+00
(Intercept)             9.914681e+01 1.011775e+02
time                   -2.149275e-01 5.413612e-01
I(time^2)              -3.131927e-02 4.184700e-02
I(time^3)              -1.654887e-03 1.413095e-03
I(time^4)              -2.735806e-05 2.908986e-05
I(time^5)              -1.823447e-07 1.860239e-07
I(cos(2 * pi * time/T)) 6.373020e-01 1.191464e+00
I(sin(2 * pi * time/T)) 9.826061e-01 1.518662e+00
```

There is only one random effect in the final model. We can plot 95% prediction intervals on the random effects (ηi)

Hide

```
library(lattice)
d = dotplot(ranef(lme2, condVar = TRUE))
print(d[[1]])
```

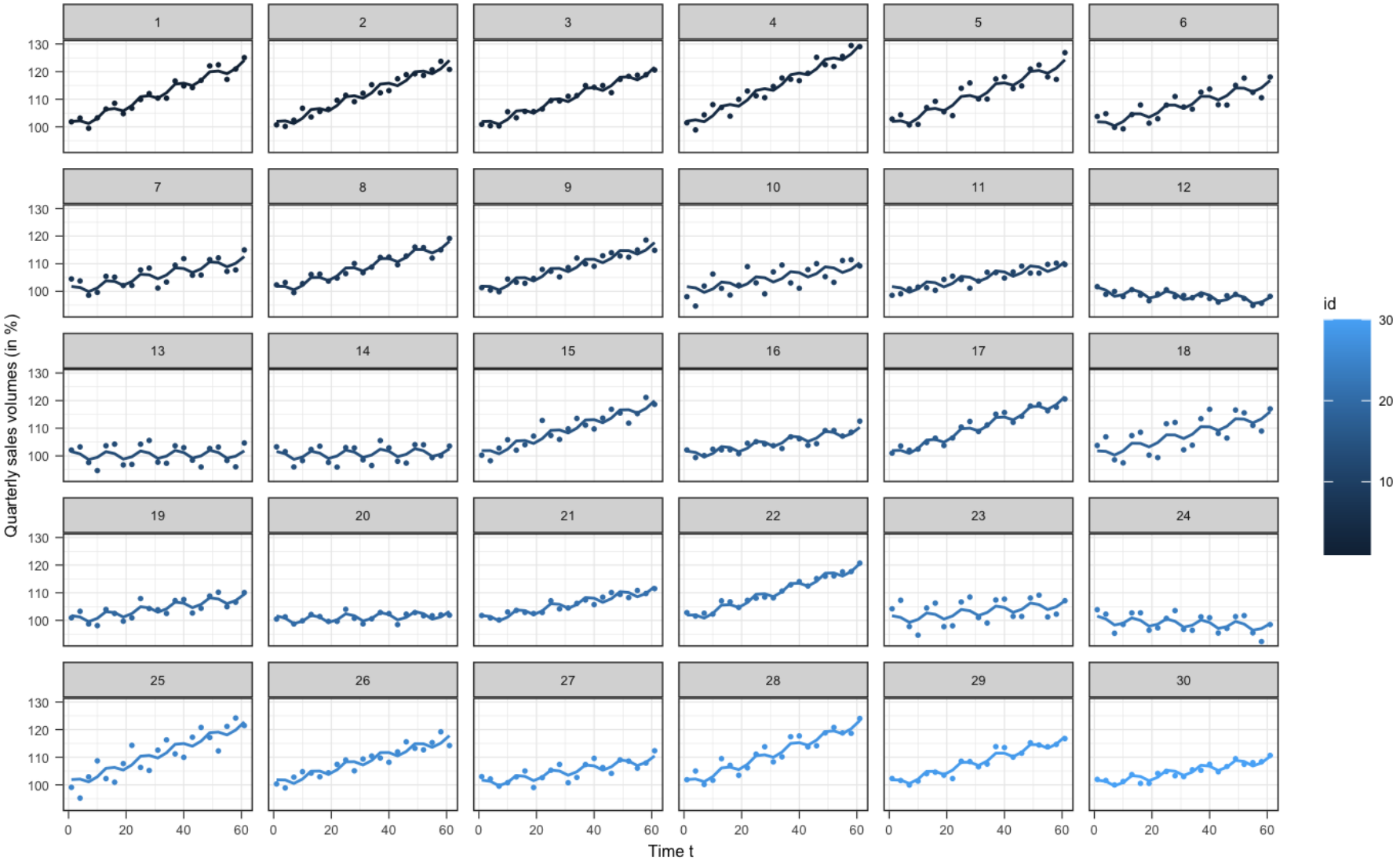


2.4 Plot the data with the predicted sales given by your final model.

Let us plot the predicted time together with the observed time.

Hide

```
 pl + geom_line(data=sales30 ,aes(x=time,y=predict(lme2))) + facet_wrap(~ id ) + xlab("Time t") + ylab("Quarterly sales volumes (in %)")
```



We can also check that the predicted distances for a given individual (“id=2” for instance)

Hide

```
subset(sales30,id == "2")
```

	id <int>	time <int>	y <dbl>	pred.lm2 <dbl>
22	2	1	100.7274	101.7855
23	2	4	100.1915	101.5363
24	2	7	102.4322	100.1109
25	2	10	106.7346	101.5416
26	2	13	103.5701	104.2249
27	2	16	105.5952	104.1039
28	2	19	106.4030	102.7627
29	2	22	109.6198	104.2430
30	2	25	111.4352	106.9498
31	2	28	109.1279	106.8331

1-10 of 21 rows Previous **1** 2 3 Next

Prediction are in accordance with the current value.

2.5 How could you take into account the previous constraint (predicted sales volume are all equal to 100 at time 0)?

We would just have to fix the intercept of the regression model to 100 by adding an offset to the model (same as previous exercise)

3. Individual prediction

The file salesNew.csv consists of quarterly sales volumes of another product.

The final model of part 2 will be used here. In other words, you should not use the new data to fit any new model.

3.1 Suppose first that we don’t have any data for this product (although data are available for this product, we act as if we do not know them). How can we predict the sales volumes for this product? plot the data and the prediction on a same graph.

Hide

```
salesnew <- read.csv("/Users/haliouanaomie/PolytechniqueS2/MAP566/salesData/salesNew.csv")
head(salesnew)
```

	time <int>	y <dbl>
1	1	100.21211
2	4	98.82607
3	7	101.85268
4	10	106.67441
5	13	105.25327
6	16	105.72561

6 rows

Hide

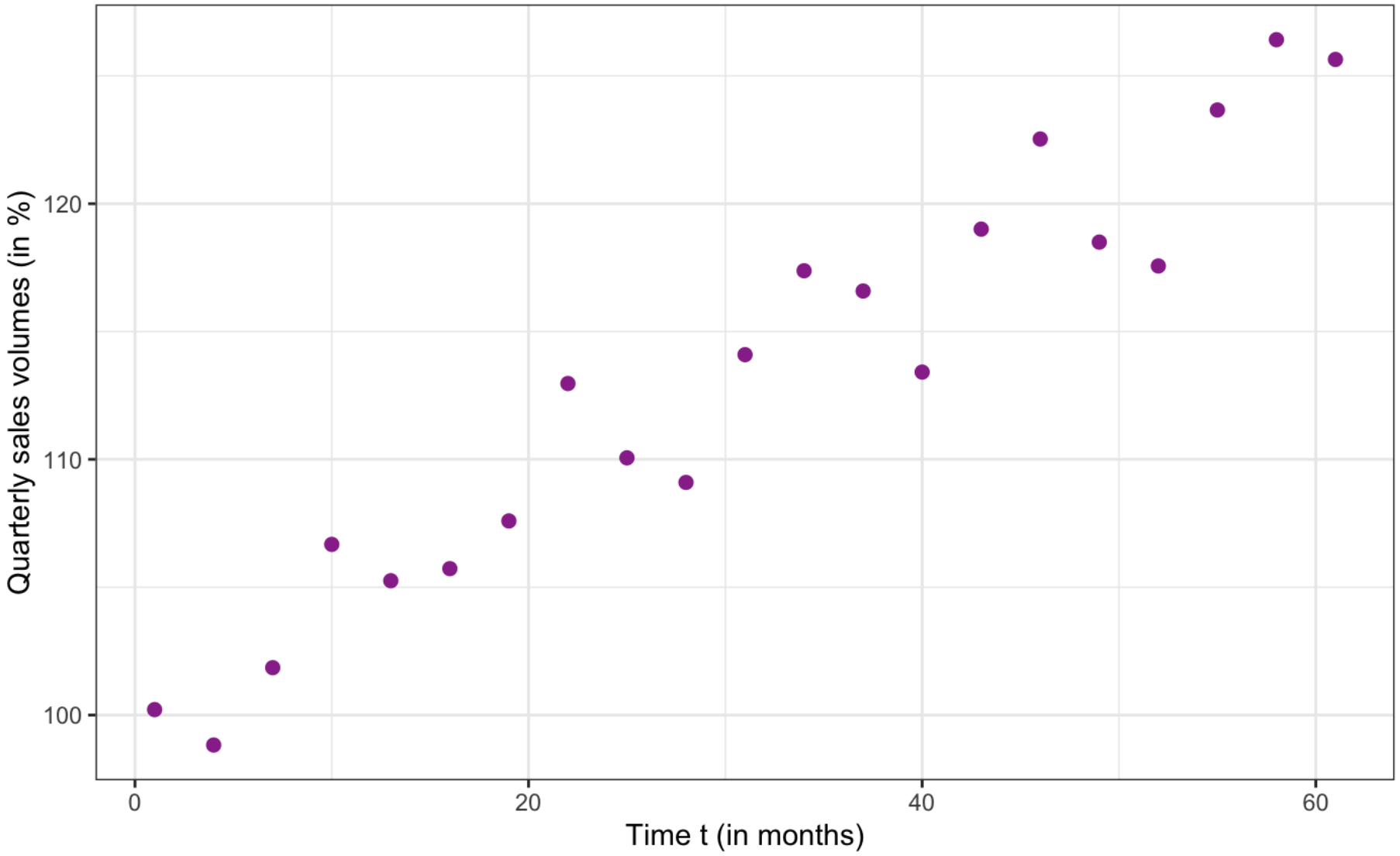
```
dim(salesnew)
```

```
[1] 21  2
```

Hide

```
library(ggplot2)
theme_set(theme_bw())
```

```
pl <- ggplot(salesnew) + geom_point(aes(x=time, y=y), size=2, colour="#993399") + xlab("Time t (in months)") + yl
ab("Quarterly sales volumes (in %)")
print(pl)
```



Given that we don't have any data for this new product we are going to predict the sales volumes using our previous model trained on the 30 products of exercise 2:

Hide

```
model30 <- lm(y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2*pi*time/T)) + I(sin(2*pi*time/T
)), data=data30)
summary(model30)
```

```

Call:
lm(formula = y ~ time + I(time^2) + I(time^3) + I(time^4) + I(time^5) +
    I(cos(2 * pi * time/T)) + I(sin(2 * pi * time/T)), data = data30)

Residuals:
     Min       1Q   Median       3Q      Max
-19.5795  -2.9541   0.0852   3.3104  17.5351

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)             1.002e+02  1.224e+00  81.864  < 2e-16 ***
time                    1.635e-01  4.074e-01   0.401   0.6883
I(time^2)               4.244e-03  4.106e-02   0.103   0.9177
I(time^3)              -1.295e-04  1.684e-03  -0.077   0.9387
I(time^4)               1.844e-06  2.996e-05   0.062   0.9510
I(time^5)              -1.073e-08  1.923e-07  -0.056   0.9555
I(cos(2 * pi * time/T))  9.138e-01  3.136e-01   2.914   0.0037 **
I(sin(2 * pi * time/T))  1.237e+00  3.146e-01   3.931 9.41e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.377 on 622 degrees of freedom
Multiple R-squared:  0.3636,    Adjusted R-squared:  0.3564
F-statistic: 50.76 on 7 and 622 DF,  p-value: < 2.2e-16
```

Hide

```
predictions <- predict(model30, newdata=salesnew)

data.frame(salesnew, predictions)
```

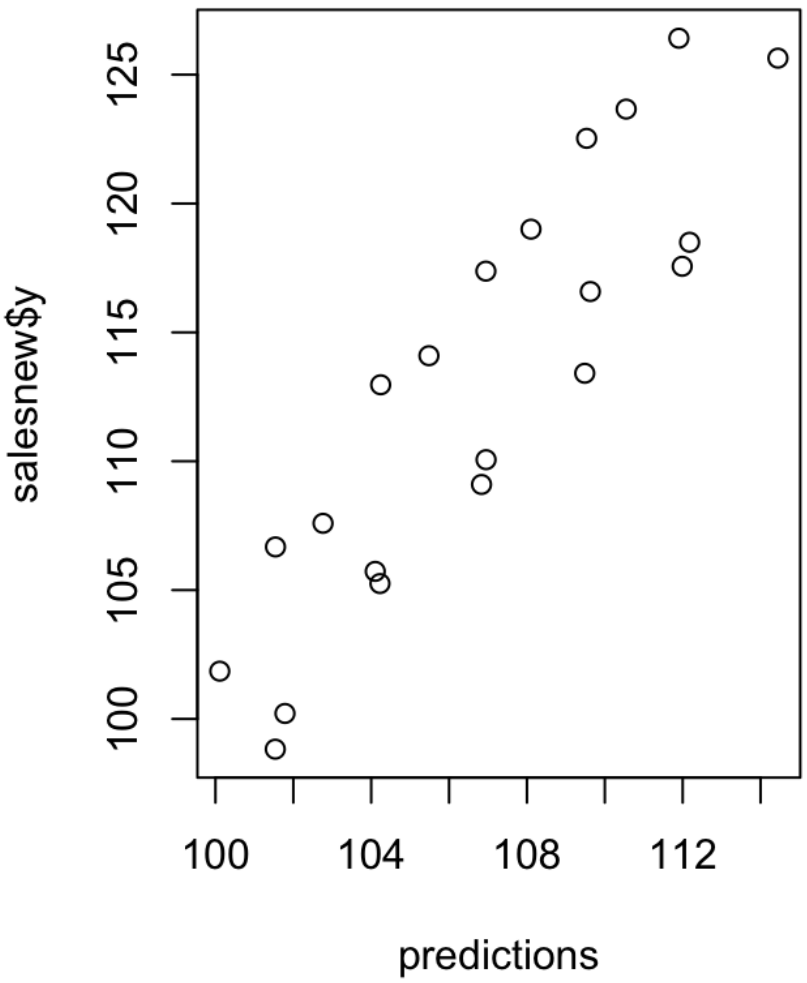
	time <int>	y <dbl>	predictions <dbl>
1	1	100.21211	101.7855
2	4	98.82607	101.5363
3	7	101.85268	100.1109
4	10	106.67441	101.5416
5	13	105.25327	104.2249
6	16	105.72561	104.1039
7	19	107.59093	102.7627
8	22	112.96610	104.2430
9	25	110.05979	106.9498
10	28	109.09797	106.8331

1-10 of 21 rows

Previous 1 2 3 Next

Hide

```
par(mfrow=c(1,2))
plot(predictions, salesnew$y)
```



Question 2: Suppose now that only the first data at time 1 is available for this product. Compute and plot the new predictions.

Hide

```
first_data <- head(salesnew, 1)$y
first_data
```

```
[1] 100.2121
```

Hide

```
model30 <- lm(y ~ 0 + time + I(time^2) + I(time^3) + I(time^4) + I(time^5) + I(cos(2*pi*time/T)) + I(sin(2*pi*tim
e/T)), offset=rep(first_data, length(time)), data=data30)
summary(model30)
```

```

Call:
lm(formula = y ~ 0 + time + I(time^2) + I(time^3) + I(time^4) +
    I(time^5) + I(cos(2 * pi * time/T)) + I(sin(2 * pi * time/T)),
    data = data30, offset = rep(first_data, length(time)))

Residuals:
    Min      1Q  Median      3Q     Max
-19.580  -2.954   0.084   3.311  17.535

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
time                     1.624e-01  2.115e-01   0.768  0.44290
I(time^2)                4.340e-03  2.817e-02   0.154  0.87762
I(time^3)               -1.330e-04  1.295e-03  -0.103  0.91823
I(time^4)                1.899e-06  2.452e-05   0.077  0.93830
I(time^5)               -1.106e-08  1.637e-07  -0.068  0.94617
I(cos(2 * pi * time/T))  9.137e-01  3.104e-01   2.943  0.00337 **
I(sin(2 * pi * time/T))  1.236e+00  3.084e-01   4.009 6.83e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.372 on 623 degrees of freedom
```

```
Multiple R-squared:  0.9975,    Adjusted R-squared:  0.9975
F-statistic: 3.567e+04 on 7 and 623 DF,  p-value: < 2.2e-16
```