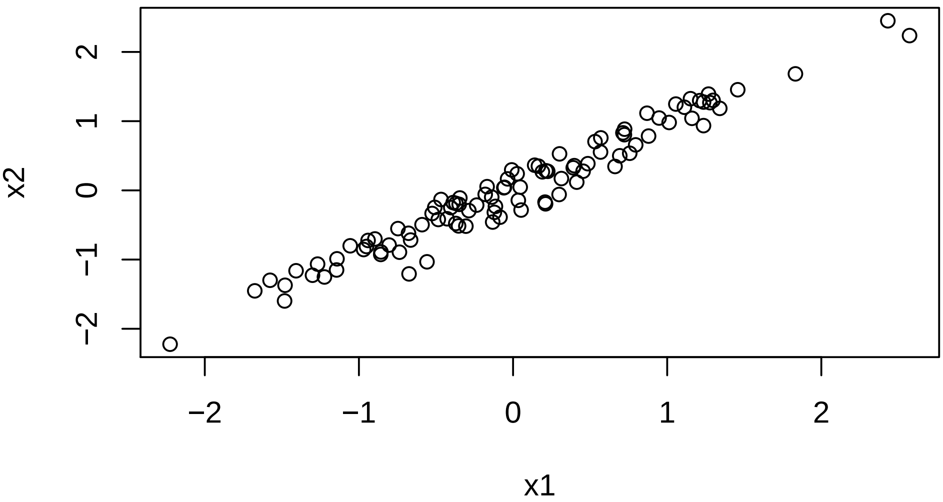
Solution to Series 2

1. **a)**
```
> set.seed(0)
> n<-100
> z1<-rnorm(n)
> z2<-rnorm(n)
> M=matrix(c(1,1,0.1,-0.1),2,2)
> X=t(M%*%rbind(z1,z2))
> beta<-c(0.5,-1.0)
> x1=X[,1]
> x2=X[,2]
> y=5+beta[1]*x1+beta[2]*x2 +rnorm(n)
> plot(x1,x2) # you can also use plot(X)
```



b)
```
> fit1<-lm(y~x1+x2)
> summary(fit1)

Call:
lm(formula = y ~ x1 + x2)

Residuals:
     Min       1Q   Median       3Q      Max
-2.89540 -0.73467 -0.01828  0.58897  2.43687

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0645     0.1062  47.678   <2e-16 ***
x1            0.4440     0.5521   0.804    0.423
x2           -0.8638     0.5674  -1.522    0.131
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.061 on 97 degrees of freedom
Multiple R-squared:  0.1137,        Adjusted R-squared:  0.09542
F-statistic: 6.222 on 2 and 97 DF,  p-value: 0.002869
```

c)
```
> s<-summary(fit1)
> coef<-s$coefficients
> se1<-coef["x1","Std. Error"]
> beta1<-coef["x1","Estimate"]
> t1<-beta1/se1
> t1
[1] 0.8041907
```

d) A p-value is the probability of observing a test statistic that is at least as extreme as the one we saw under the null-hypothesis. The notion of "extreme" depends on the alternative hypothesis.

```
> p1<-2*pt(-abs(t1),df=n-3) #p-value for x1 = 0.42325
> p1
[1] 0.4232534
```

e) The p-value of the overall F-test can be read directly from the `summary(fit)` output above. It is approximately 0.002869. We reproduce it as follows

```
> fit2<-lm(y~1)
> anova(fit2,fit1)
Analysis of Variance Table

Model 1: y ~ 1
Model 2: y ~ x1 + x2
  Res.Df    RSS Df Sum of Sq      F   Pr(>F)
1     99 123.09
2     97 109.09  2    13.995 6.2218 0.002869 **
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see the same p-value in the above output. It compares the following models:

$$
\begin{aligned}
H_0: &\quad y_i = \beta_0 + \varepsilon_i \\
H_a: &\quad y_i = \beta_0 + \beta_2 X_{i2} + \beta_3 X_{i3} + \varepsilon_i
\end{aligned}
$$

The null model can be rejected at the 5% level.

f) The p-value is insignificant at $\alpha = 0.05$. This means that there is not enough evidence in the data to favor H_a over H_0. When already using `x2`, adding `x1` does not significantly improve the model. This is no contradiction to the fact that the F-test shows significance.

g) The summary output says `Residual standard error: 1.061`. The residual standard error is an estimate of the standard deviation of the noise involved in the linear model.

```
> res=residuals(fit1)
> sigma2hat<- sum((res)^2)/(n-3)
> residualStandardError= sqrt(sigma2hat)
> residualStandardError
[1] 1.060502
```

h) From the summary output we see that `Multiple R-squared: 0.1137`. The R^2 value is the proportion of the variance of y that is explained by the fitted linear model.

```
> RSquared=1-sum((residuals(fit1))^2)/ sum((y-mean(y))^2)
> RSquared
[1] 0.1136987
```

i)
```
> fit3<-lm(y~x1)
> fit3
Call:
lm(formula = y ~ x1)

Coefficients:
(Intercept)           x1
     5.0559      -0.3769
```

The coefficient has a different sign than before. In both models, this is the amount by which the fitted values change if x1 is increased by 1. However, in the first model, we fix x2 whereas in the second model, there is no such second predictor. The "effect" of a certain predictor depends on the specified model.

2. **a)** At first we use the commands given in the exercise.

```
> # prepare data
> library(ISLR)
> data(Carseats)
> shelveloc=Carseats$ShelveLoc
> sales=Carseats$Sales
> advertising=Carseats$Advertising
> # fit using automatic coding
> fit<-lm(sales~shelveloc+advertising)
> summary(fit)
Call:
lm(formula = sales ~ shelveloc + advertising)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6480 -1.6198 -0.0476  1.5308  6.4098

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      4.89662    0.25207  19.426  < 2e-16 ***
shelvelocGood    4.57686    0.33479  13.671  < 2e-16 ***
shelvelocMedium  1.75142    0.27475   6.375 5.11e-10 ***
advertising      0.10071    0.01692   5.951 5.88e-09 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.244 on 396 degrees of freedom
Multiple R-squared:  0.3733,       Adjusted R-squared:  0.3685
F-statistic: 78.62 on 3 and 396 DF,  p-value: < 2.2e-16
> # boolean vectors for easy construction of a1, a2, b1,...
> bad<- levels(shelveloc)[1]==shelveloc
> medium<- levels(shelveloc)[3]==shelveloc
> good<- levels(shelveloc)[2]==shelveloc
```

We define the predictors a1 and a2 as follows.

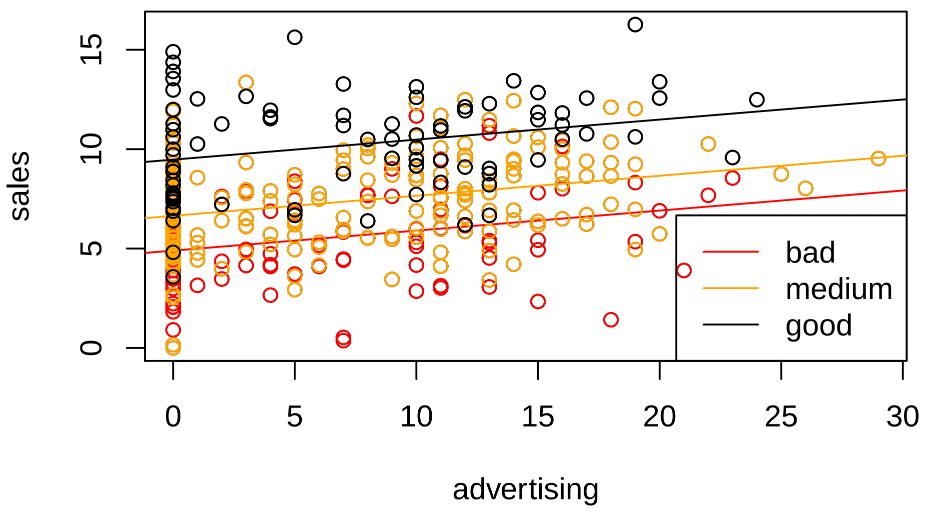
```
> a1=medium*1
> a2=good*1
> fit_a<-lm(sales~1+a1+a2+advertising)
> summary(fit_a)
Call:
lm(formula = sales ~ 1 + a1 + a2 + advertising)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6480 -1.6198 -0.0476  1.5308  6.4098

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.89662    0.25207  19.426  < 2e-16 ***
a1           1.75142    0.27475   6.375 5.11e-10 ***
a2           4.57686    0.33479  13.671  < 2e-16 ***
advertising  0.10071    0.01692   5.951 5.88e-09 ***
```

```
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.244 on 396 degrees of freedom
Multiple R-squared:  0.3733,	Adjusted R-squared:  0.3685
F-statistic: 78.62 on 3 and 396 DF,  p-value: < 2.2e-16
```

```
> max(abs(fitted(fit_a)-fitted(fit)))
[1] 2.753353e-13
```

The summaries and the fitted values are indeed the same. The coefficient of `a1` is the increase in the intercept for the level `medium` compared to the level `bad` and the coefficient of `a2` is the increase in the intercept of `good` compared to `bad` (concerning `shelveloc`). The slopes with respect to `advertising` are the same for all three levels. Intuitively, if we have two child car seats offered at two different locations with the same amount spent on advertising, according to our model we would expect the difference in the sales between a seat with `medium` and a seat with `bad` shelve location to be equal to the coefficient of `a1` and the difference in the sales between a seat with `good` and a seat with `bad` shelve location to be equal to the coefficient of `a2`. The following R commands visualize the fitted regression lines for all three levels of `shelveloc`:

```
> plot(advertising,sales)
> c=coef(fit_a)
> points(advertising[bad],sales[bad],col="red")
> abline(a=c["(Intercept)"],b=c["advertising"],col="red")
> points(advertising[medium],sales[medium],col="orange")
> abline(a=c["(Intercept)"]+c["a1"],b=c["advertising"],col="orange")
> points(advertising[good],sales[good])
> abline(a=c["(Intercept)"]+c["a2"],b=c["advertising"])
> legend("bottomright", c("bad","medium","good"),col=c("red","orange","black"), lty=1)
```



For fixed `advertising`, the fitted number of sales is higher by 4.577 sold units for a `good` shelve location than for a `bad` shelve location. Similarly, for fixed `advertising`, the fitted number of sales for a `medium` shelve location exceed the fitted number of sales for a `bad` shelve location by about 1.751.

b) Similar to a), we define the predictors `b1` and `b2` and fit the model.

```
> b1=bad*1
> b2=good*1
> fit_b<-lm(sales~1+b1+b2+advertising)
> summary(fit_b)
Call:
lm(formula = sales ~ 1 + b1 + b2 + advertising)

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-6.6480 -1.6198 -0.0476  1.5308  6.4098

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.64805    0.18773  35.413  < 2e-16 ***
b1          -1.75142    0.27475  -6.375 5.11e-10 ***
b2           2.82543    0.28712   9.841  < 2e-16 ***
advertising  0.10071    0.01692   5.951 5.88e-09 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.244 on 396 degrees of freedom
Multiple R-squared:  0.3733,        Adjusted R-squared:  0.3685
F-statistic: 78.62 on 3 and 396 DF,  p-value: < 2.2e-16
```

The coefficient of `b1` is the increase in the intercept for the level `bad` compared to the level `medium` and the coefficient of `b2` is the increase in the intercept of `good` compared to `medium` (concerning `shelveloc`). Again, the slopes with respect to `advertising` are the same for all three levels. Intuitively, if we have two child car seats offered at two different locations with the same amount spent on advertising, according to our model we would expect the difference in the sales between a seat with `bad` and a seat with `medium` shelve location to be equal to the coefficient of `b1` and the difference in the sales between a seat with `good` and a seat with `medium` shelve location to be equal to the coefficient of `b2`.

c) First, we fit the model `fit_c` as follows

```
> c1=bad*1
> c2=medium*1
> c3=good*1
> fit_c<-lm(sales~+c1+c2+c3+advertising)
> summary(fit_c)
```

```
Call:
lm(formula = sales ~ +c1 + c2 + c3 + advertising)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6480 -1.6198 -0.0476  1.5308  6.4098

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.47348    0.27338  34.653  < 2e-16 ***
c1          -4.57686    0.33479 -13.671  < 2e-16 ***
c2          -2.82543    0.28712  -9.841  < 2e-16 ***
c3                NA         NA      NA       NA
advertising  0.10071    0.01692   5.951 5.88e-09 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.244 on 396 degrees of freedom
Multiple R-squared:  0.3733,        Adjusted R-squared:  0.3685
F-statistic: 78.62 on 3 and 396 DF,  p-value: < 2.2e-16
```

But we see that `c3` has `NA` values. The reason for this is that the columns of the model matrix in this case are linearly dependent beause we have too many predictors. More precisely, the predictors `(intercept)`, `c1`, `c2` and `c3` are linearly dependent such that there exist infinitely many possible solutions to the least squares problem. To avoid this, we need to fit the model without intercept.

d) First, we fit the model `fit_c` as follows

```
> fit_c<-lm(sales~-1+c1+c2+c3+advertising)
> summary(fit_c)
```

```
Call:
lm(formula = sales ~ -1 + c1 + c2 + c3 + advertising)

Residuals:
    Min      1Q  Median      3Q     Max
-6.6480 -1.6198 -0.0476  1.5308  6.4098

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
c1           4.89662    0.25207  19.426  < 2e-16 ***
c2           6.64805    0.18773  35.413  < 2e-16 ***
c3           9.47348    0.27338  34.653  < 2e-16 ***
advertising  0.10071    0.01692   5.951 5.88e-09 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.244 on 396 degrees of freedom
Multiple R-squared:  0.9223,       Adjusted R-squared:  0.9215
F-statistic:  1175 on 4 and 396 DF,  p-value: < 2.2e-16
```

Now the problem is not present anymore. The coefficient of `c1` is the intercept for the fit when the level of `shelveloc` is bad, similarly the coefficient of `c2` is the intercept for the level `medium` and the coefficient of `c3` is the intercept for the level good. Note that for example the difference in intercepts for `medium` compared to bad is about $6.648 - 4.897$ which is equal the coefficient of `a1` in subtask a).

e) The fitted values are the same up to rounding errors as can be seen from the following R output.

```
> max(abs(fitted(fit_a)-fitted(fit_b)))
[1] 2.096101e-13
> max(abs(fitted(fit_b)-fitted(fit_c)))
[1] 1.900702e-13
```

You can verify that one can obtain all point estimates of one model from any of the other models.

f) This can be seen in the summary of `fit_b`. If we drop predictor `b2`, we do not distinguish anymore between the levels `medium` and good. This means we only have to consider the p-value corresponding to `b2` which is smaller than 2e-16, which is smaller than 0.05. We conclude that we should distinguish between `medium` and good.

g) The model from part a) is

$$y_i = \beta_0 + \beta_1(\texttt{advertising})_i + \alpha_1(\texttt{a1})_i + \alpha_2(\texttt{a2})_i + \varepsilon_i \quad \text{for } \varepsilon_1, \cdots, \varepsilon_n \text{ i.i.d } \sim \mathcal{N}(0, \sigma^2)$$

When we do not distinguish between the levels `medium` and good, then $\alpha_1 = \alpha_2$. The new model can be written as

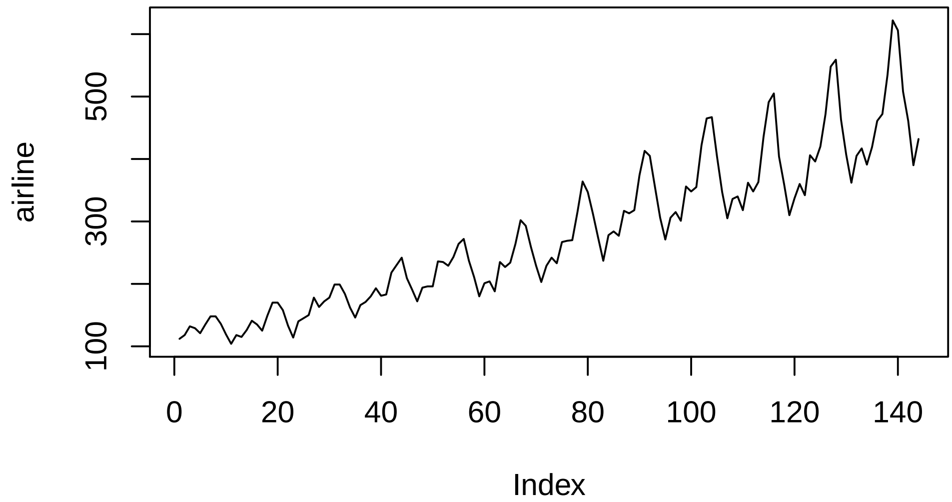
$$y_i = \beta_0 + \beta_1(\texttt{advertising})_i + \phi_1(\texttt{a1} + \texttt{a2})_i + \varepsilon_i \quad \text{for } \varepsilon_1, \cdots, \varepsilon_n \text{ i.i.d } \sim \mathcal{N}(0, \sigma^2)$$

Hence, this model is clearly a submodel of the model from part a) such that we can use the partial F-test, which is highly significant.

```
> f1=medium*1+good*1
> fit_d<-lm(sales~f1+advertising)
> anova(fit_d,fit_a)
Analysis of Variance Table

Model 1: sales ~ f1 + advertising
Model 2: sales ~ 1 + a1 + a2 + advertising
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    397 2482.1
2    396 1994.4  1    487.71 96.837 < 2.2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

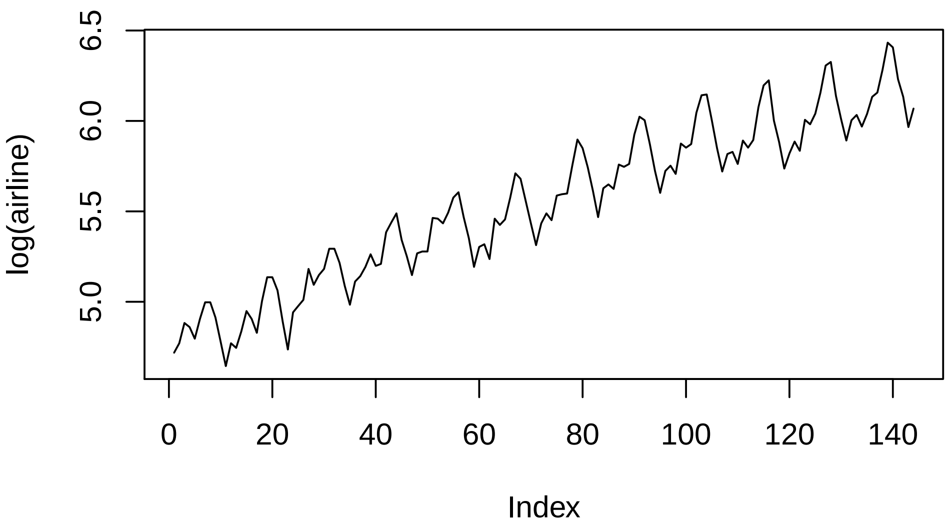
The p-value is again smaller than 2e−16

3. **a)**
```
> airline <- scan("http://stat.ethz.ch/Teaching/Datasets/airline.dat")
> plot(airline, type="l")
```



The data show an increasing trend, which might be linear. Moreover, there are monthly fluctuations, which get stronger with time. If a linear model would be fitted to these data, we could observe the residual variance increasing with time.

b)
```
> airline <- scan("http://stat.ethz.ch/Teaching/Datasets/airline.dat")
> plot(log(airline),type="l")
```



With logarithmized data, the global trend remains more or less linear, while the monthly fluctuations get stable. The fit of a linear model is much more reasonable here. Taking logarithms or other transformations of the target variable is often a good method to remove monotone trends in the variation. In terms of the original variables, this means that a multiplicative model is fitted instead of an additive one (see part e)).

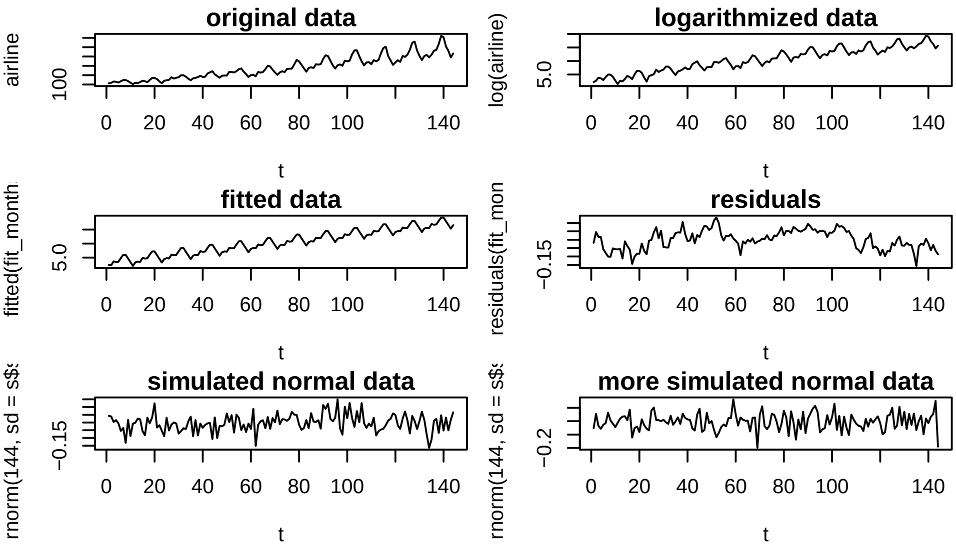
c) The predictors and fit can be obtained as follows:

```
> airline <- scan("http://stat.ethz.ch/Teaching/Datasets/airline.dat")
> x1<-rep(c(1,rep(0,11)),12)
> x2<-rep(c(rep(0,1),1,rep(0,10)),12)
> x3<-rep(c(rep(0,2),1,rep(0,9)),12)
> x4<-rep(c(rep(0,3),1,rep(0,8)),12)
> x5<-rep(c(rep(0,4),1,rep(0,7)),12)
```

```
> x6<-rep(c(rep(0,5),1,rep(0,6)),12)
> x7<-rep(c(rep(0,6),1,rep(0,5)),12)
> x8<-rep(c(rep(0,7),1,rep(0,4)),12)
> x9<-rep(c(rep(0,8),1,rep(0,3)),12)
> x10<-rep(c(rep(0,9),1,rep(0,2)),12)
> x11<-rep(c(rep(0,10),1,rep(0,1)),12)
> x12<-rep(c(rep(0,11),1,rep(0,0)),12)
> t<-1:144
> fit_months<-lm(log(airline)~-1+t+x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12)
> summary(fit_months)
Call:
lm(formula = log(airline) ~ -1 + t + x1 + x2 + x3 + x4 + x5 +
    x6 + x7 + x8 + x9 + x10 + x11 + x12)

Residuals:
      Min        1Q    Median        3Q       Max
-0.156370 -0.041016  0.003677  0.044069  0.132324

Coefficients:
     Estimate Std. Error t value Pr(>|t|)
t   0.0100688  0.0001193    84.4   <2e-16 ***
x1  4.7267804  0.0188935   250.2   <2e-16 ***
x2  4.7047255  0.0189443   248.3   <2e-16 ***
x3  4.8349527  0.0189957   254.5   <2e-16 ***
x4  4.8036838  0.0190477   252.2   <2e-16 ***
x5  4.8013112  0.0191003   251.4   <2e-16 ***
x6  4.9234574  0.0191535   257.1   <2e-16 ***
x7  5.0273997  0.0192073   261.7   <2e-16 ***
x8  5.0181049  0.0192617   260.5   <2e-16 ***
x9  4.8734703  0.0193167   252.3   <2e-16 ***
x10 4.7353120  0.0193722   244.4   <2e-16 ***
x11 4.5915943  0.0194283   236.3   <2e-16 ***
x12 4.7054593  0.0194850   241.5   <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0593 on 131 degrees of freedom
Multiple R-squared:  0.9999,        Adjusted R-squared:  0.9999
F-statistic: 9.734e+04 on 13 and 131 DF,  p-value: < 2.2e-16
```

d) The relevant plots are shown below. For the model assumptions we observe that departures from normality, heteroscedasticity of variances or violations of linearity are not clearly visible, but the residuals seem to be correlated. Some experience is needed to assess such plots. One possibility to acquire such experience is to take a look at artificial data generated according to the model which we want to check (i.i.d. normally distributed residuals). You may compare the actual residuals with the two plots from artificial data. Since there seems to be serial correlation (violation of model assumptions), the standard errors and p-values are not valid.

```
> par(mfrow=c(3,2))
> plot(t, airline, type="l", main="original data")
> plot(t, log(airline), type="l", main="logarithmized data")
> ## plots of fitted values and residuals
> plot(t, fitted(fit_months), type="l", main="fitted data")
> plot(t, residuals(fit_months), type="l", main="residuals")
> ## two artificial normal datasets to compare
> s=summary(fit_months)
> plot(t, rnorm(144,sd=s$sigma), type="l", main="simulated normal data")
> plot(t, rnorm(144,sd=s$sigma), type="l", main="more simulated normal data")
```



e) The fitted values are defined by

$$\widehat{\log(y_t)} = \widehat{\beta} t + \sum_{j=1}^{12} \widehat{\gamma}_j x_{tj}.$$

Hence,

$$\widehat{y_{t+12}} = \exp(\widehat{\log(y_{t+12})}) = \exp(\widehat{\beta}(t+12) + \sum_{j=1}^{12} \widehat{\gamma}_j \underbrace{x_{(t+12)j}}_{x_{tj}}) = \exp(12\widehat{\beta})\exp(\widehat{\log(y_t)}) = \exp(12\widehat{\beta})\widehat{y_t}.$$

We have used that $x_{(t+12)j} = x_{tj}$ for all $j \in \{1, \cdots, 12\}$ because if we increase the month index by 12, the same month indicator will be active (one year has 12 months). This means that if we increase t by 12, the fitted values are multiplied by $\exp(12\widehat{\beta})$. Hence we have a multiplicative instead of an additive model. The larger $\widehat{\beta}$, the larger the multiplication factor.

f)
```
> s1<-rep(c(rep(0,2),rep(1,3),rep(0,7)),12)
> s2<-rep(c(rep(0,5),rep(1,3),rep(0,4)),12)
> s3<-rep(c(rep(0,8),rep(1,3),rep(0,1)),12)
> s4<-rep(c(1,1,rep(0,9),1),12)
> fit_seasons<-lm(log(airline)~-1+t+s1+s2+s3+s4)
> anova(fit_seasons,fit_months)
```

```
Analysis of Variance Table

Model 1: log(airline) ~ -1 + t + s1 + s2 + s3 + s4
Model 2: log(airline) ~ -1 + t + x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 +
    x9 + x10 + x11 + x12
  Res.Df     RSS Df Sum of Sq      F    Pr(>F)
1    139 1.02907
2    131 0.46072  8   0.56835 20.201 < 2.2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The partial F-test is significant, i.e. the larger model with all monthly predictors is significantly better than the smaller model with only four predictors, one for each season.