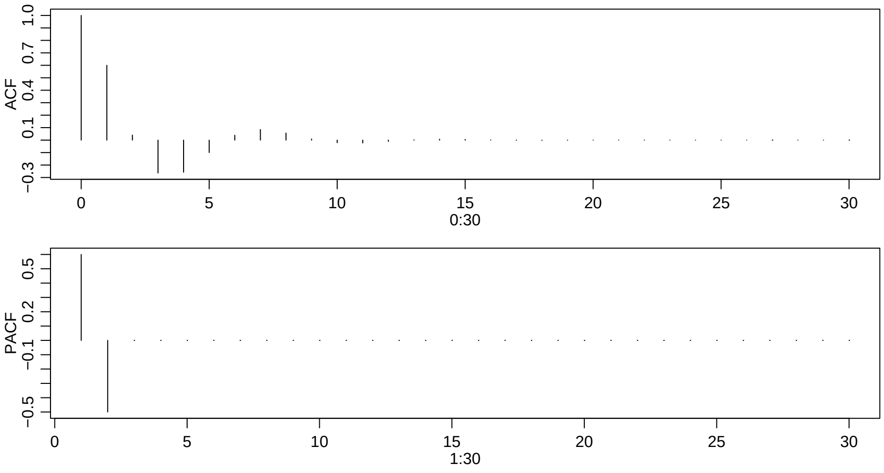
Solution to Series 4

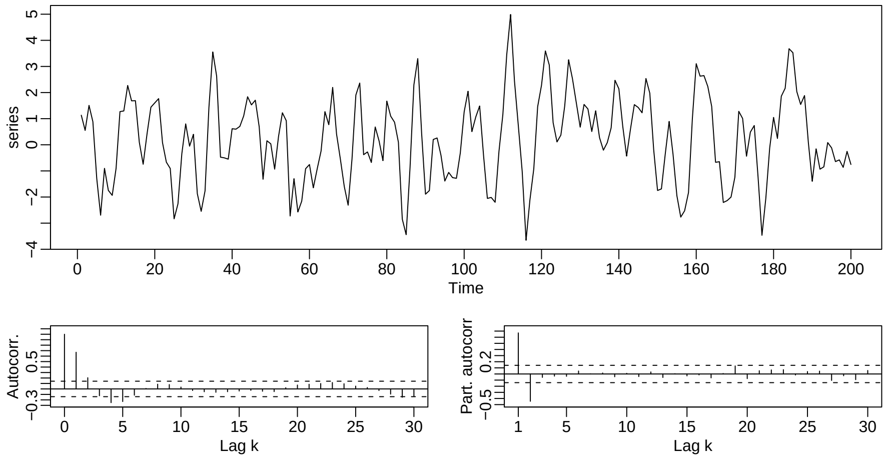
1. a) AR(2) with $\alpha_1 = 0.9$ and $\alpha_2 = -0.5$:

The ordinary autocorrelations describe a dampened sine curve, and the partial autocorrelations are cut off at lag $k = 2$. For the simulated model (`set.seed(79)`), the estimate of partial autocorrelation at lag 2 is $\hat{\rho}_{part}(2) = -0.45$, which is a reasonably close to its theoretical counterpart, $\rho_{part}(2) = \alpha_2 = -0.5$.

Plot of theoretical ordinary and partial autocorrelations:



Plots for a simulated time series of length $n = 200$:



AR(2) with ar=c(0.9,-0.5)

Complete R code:

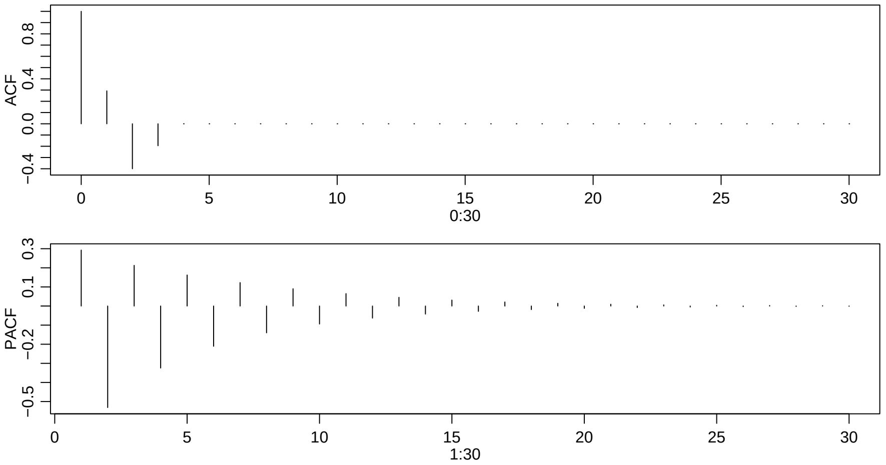
```
## Plotting theoretical ordinary autocorrelations
> plot(0:30, ARMAacf(ar=c(0.9,-0.5), lag.max=30), type="h", ylab="ACF")

# Plotting theoretical partial autocorrelations
> plot(1:30, ARMAacf(ar=c(0.9,-0.5), lag.max=30, pacf=T), type="h",
       ylab="PACF")

## Simulation
> set.seed(79)
> r.sim1 <- arima.sim(n=200, model=list(ar=c(0.9,-0.5)))
## Plotting
> plot(r.sim1)
> acf(r.sim1, lag=30)
> acf(r.sim1, type="partial", lag=30)
> str(acf(r.sim1, type="partial"))
> acf(r.sim1, type="partial")$acf[2]
```

```
[1] -0.44574
## Estimate of 2nd partial autocorrelation: -0.44574
```

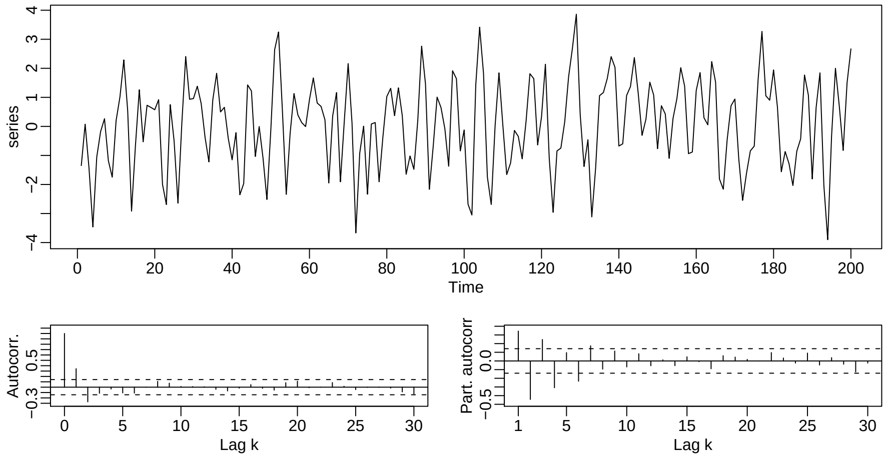
b) MA(3) with $\beta_1 = 0.8$, $\beta_2 = -0.5$ and $\beta_3 = -0.4$:

The ordinary autocorrelations are cut off at lag $k = 3$.

Plot of theoretical ordinary and partial autocorrelations:



Plots for a simulated time series of length $n = 200$:



MA(3) with ma=c(0.8,-0.5,-0.4)

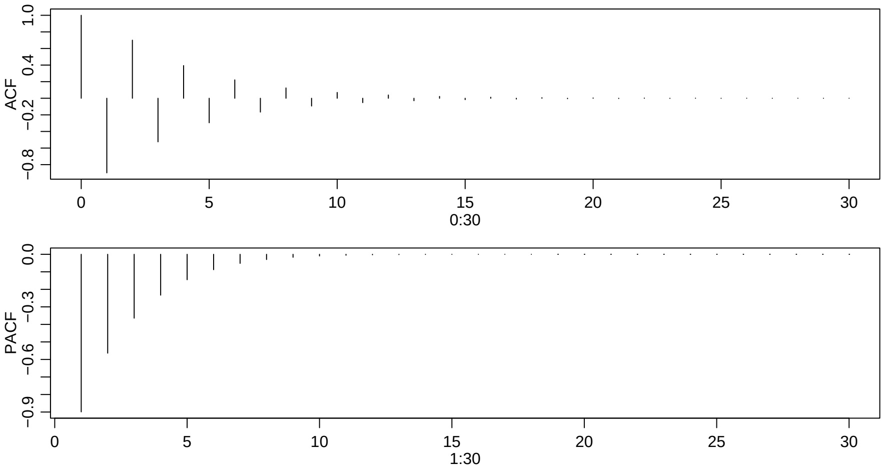
Complete R code:

```
> plot(0:30, ARMAacf(ma=c(0.8,-0.5,-0.4), lag.max=30), type="h",
       ylab="ACF")
> plot(1:30, ARMAacf(ma=c(0.8,-0.5,-0.4), lag.max=30, pacf=T),
       type="h", ylab="PACF")
> set.seed(79)
> r.sim2 <- arima.sim(n=200, model=list(ma=c(0.8,-0.5,-0.4)))
> plot(r.sim2)
> acf(r.sim2, lag=30)
> pacf(r.sim2, lag=30)
```

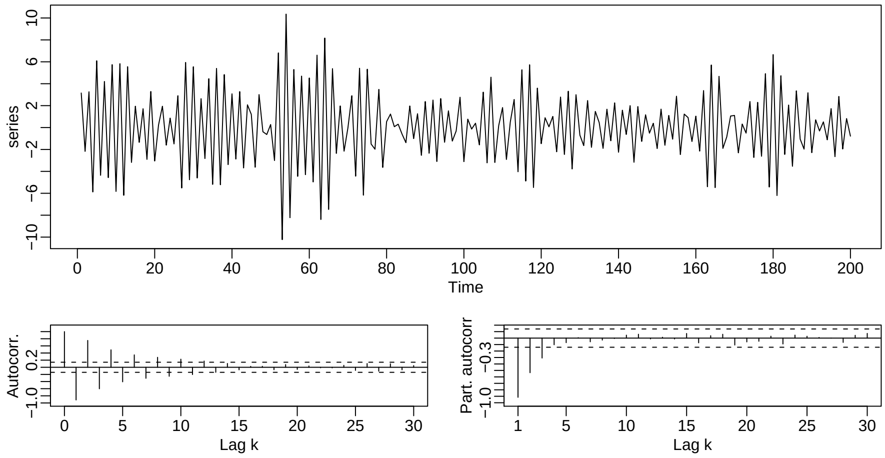
c) ARMA(1,2) with $\alpha_1 = -0.75$, $\beta_1 = -1$ and $\beta_2 = 0.25$:

The ordinary autocorrelations do not have a real cut-off point. Because of this, the MA(2) part of the model is difficult to identify in the correlogram of ordinary autocorrelations in this example. However, the autocorrelations do decay exponentially (in absolute terms – their signs alternate), which indicates that an AR component is present. The partial autocorrelations of an ARMA(1,2) process should behave like MA(2) for lags $k > 1$, and its ordinary autocorrelations like AR(1) for lags $k > 2$. The partial autocorrelation present here might also allow for an AR(3) model to describe the ARMA(1,2) model. (This goes into the topic of approximating ARMA by AR(∞) models.

Plot of theoretical ordinary and partial autocorrelations:



Plots for a time series of length $n = 200$:



ARMA(1,2) with ar=c(-0.75), ma=c(1,-0.25)

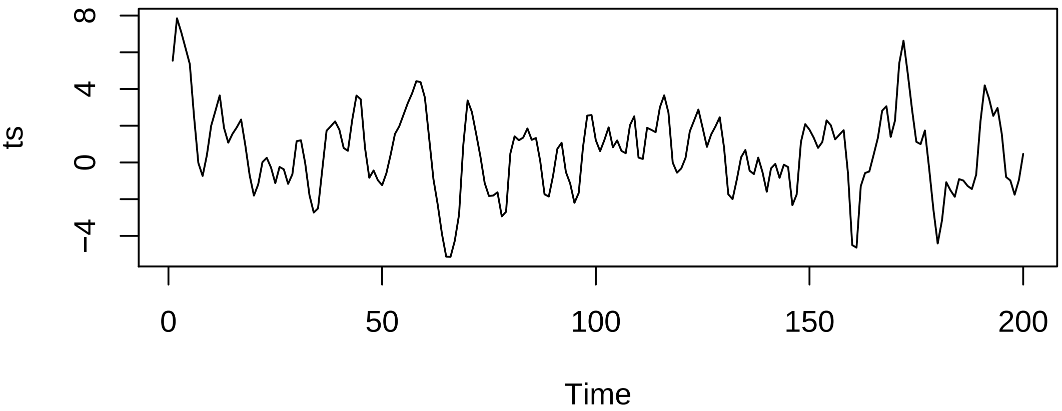
(The R code is along the same lines as that of the previous parts.)

2.

```
> dat <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/mcARMA.dat",
                    header = TRUE)
> ts <- ts(dat$x)
```

a)

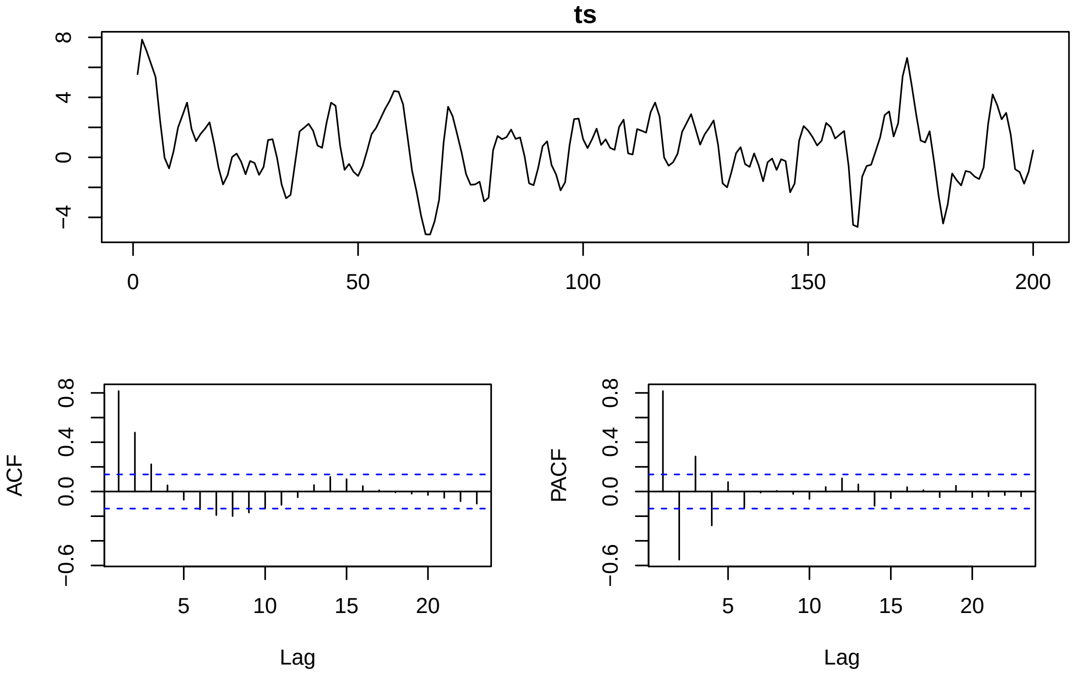
```
> plot(ts)
```



The time series looks stationary.

b)

```
> library(forecast)
> tsdisplay(ts, points = FALSE)
```

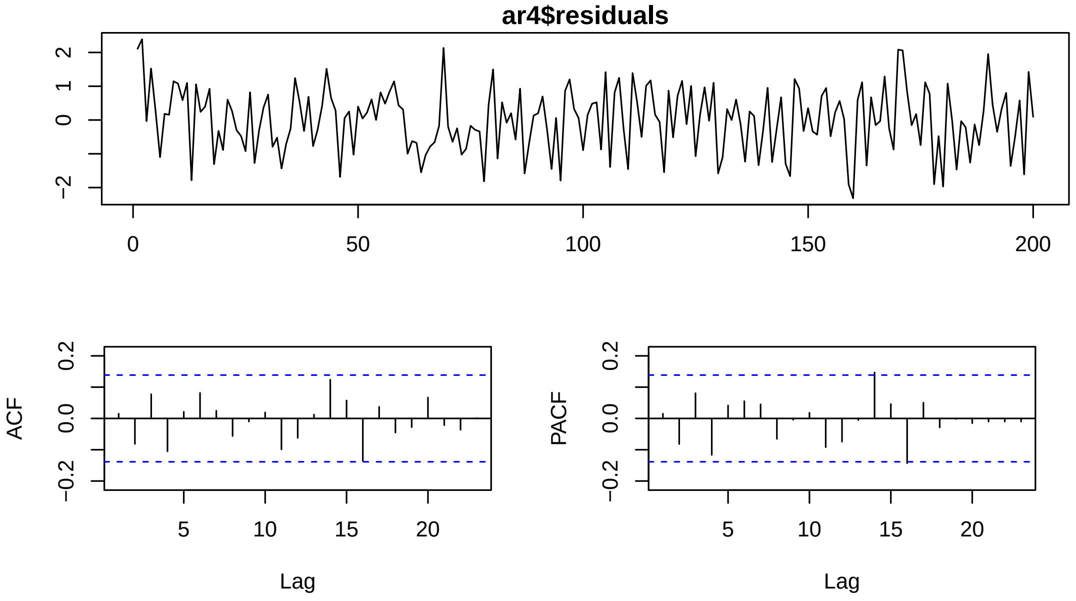


Let us first try an AR-process. By looking at the PACF, we see there is a cutoff at lag 4, and the ACF is a damped sinusoid, hence we try an AR(4)-process.

```
> ar4 <- arima(ts, order = c(4,0,0))
> ar4
```

```
Call:
arima(x = ts, order = c(4, 0, 0))

Coefficients:
         ar1      ar2     ar3      ar4  intercept
      1.5430  -1.2310  0.7284  -0.3000     0.6197
s.e.  0.0676   0.1189  0.1189   0.0697     0.2573

sigma^2 estimated as 0.8923:  log likelihood = -273.67,  aic = 559.33
```

```
> tsdisplay(ar4$residuals, points = FALSE)
```



The residuals look stationary. However, the ACF and PACF show that there are some minor dependencies left at lag 14 and 16. So let us try some other models.

If we want to try an MA(q) model, we need to look for a cutoff in the ACF function for deciding on the order q and the PACF should be exponentially decreasing, which is the case here. It is not entirely clear which cutoff to choose in the ACF, let us try $q = 9$ and $q = 3$.

```
> ma9 <- arima(ts, order = c(0,0,9))
> ma9
```

```
Call:
arima(x = ts, order = c(0, 0, 9))

Coefficients:
         ma1     ma2     ma3     ma4     ma5      ma6
      1.5787  1.1596  0.6559  0.3246  0.0650  -0.0379
s.e.  0.0723  0.1338  0.1638  0.1783  0.2028   0.2239
          ma7      ma8      ma9  intercept
      -0.0603  -0.1521  -0.1367     0.6285
s.e.   0.2243   0.1662   0.0929     0.2882

sigma^2 estimated as 0.8639:  log likelihood = -270.57,  aic = 563.14
```

By looking at (approximate) confidence intervals for the coefficients $\alpha_i, i = 1, ..., 9$, which are given by

$$\widehat{\alpha}_i \pm 1.96 \cdot \texttt{s.e.}(\widehat{\alpha}_i),$$

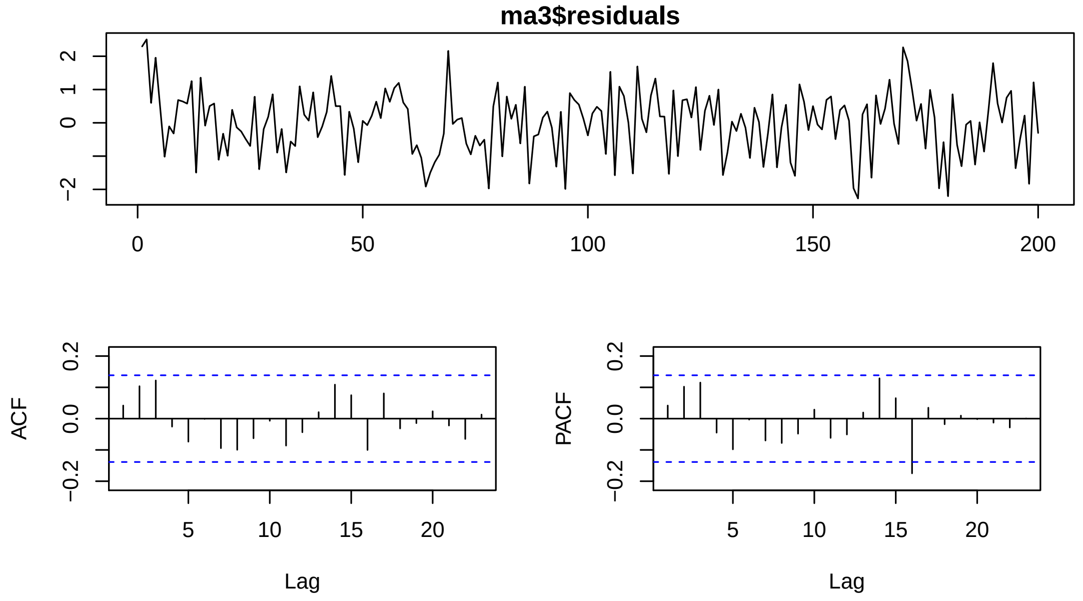
we see that the coefficients $\alpha_4, ..., \alpha_9$ are not significantly different from zero, hence an MA(9) model does not seem to be a good choice. Let us try an MA(3) instead:

```
> ma3 <- arima(ts, order = c(0,0,3))
> ma3
```

```
Call:
arima(x = ts, order = c(0, 0, 3))

Coefficients:
         ma1     ma2     ma3  intercept
      1.5711  1.0056  0.3057     0.6359
s.e.  0.0662  0.0966  0.0615     0.2604

sigma^2 estimated as 0.9098:  log likelihood = -275.64,  aic = 561.29
```

Here all coefficients are significantly different from zero, but what about the residuals?

```
> tsdisplay(ma3$residuals, points = FALSE)
```



The residuals look stationary, but they don't seem to be independent, because the PACF at lag 16 is significantly different from zero.

Thus a pure MA-process does not seem to be the answer as well. Let us try ARMA(p, q) models. Combining the models found so far, we might try to fit an ARMA(4,3)-process, representing the cutoffs in the ACF and PACF:

```
> arma43 <- arima(ts, order = c(4,0,3))
> arma43
```

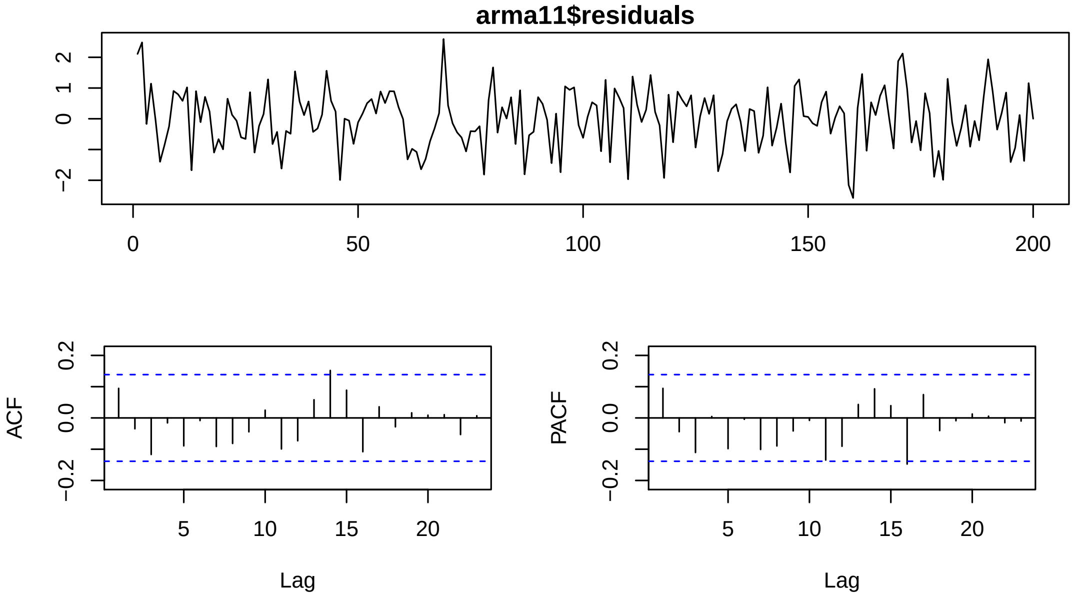
```
Call:
arima(x = ts, order = c(4, 0, 3))

Coefficients:
         ar1      ar2     ar3      ar4      ma1      ma2
      1.5675  -0.6263  0.0415  -0.0325  -0.0129  -0.7569
s.e.  0.9325   1.7066  0.9829   0.1885   0.9329   0.2641
          ma3  intercept
      -0.2301     0.5124
s.e.   0.6771     0.0465

sigma^2 estimated as 0.8343:  log likelihood = -268.36,  aic = 554.71
```

However, almost all of the coefficients are not significantly different from 0. So let's try from below with ARMA(1,1):

```
> arma11 <- arima(ts, order = c(1,0,1))
> arma11
```

```
Call:
arima(x = ts, order = c(1, 0, 1))

Coefficients:
         ar1     ma1  intercept
      0.6965  0.7981     0.6674
s.e.  0.0521  0.0400     0.3945

sigma^2 estimated as 0.9107:  log likelihood = -275.72,  aic = 559.43
```

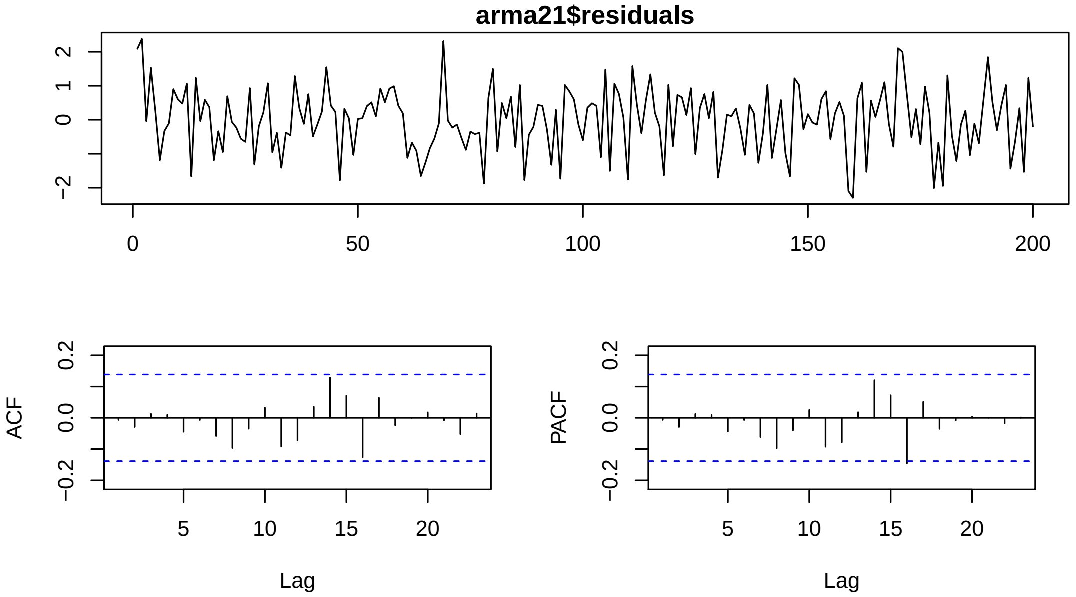
```
> tsdisplay(arma11$residuals, points = FALSE)
```



Both coefficients are significant, but the ACF shows dependency at lag 14.
If we look at the ACF/PACF of our time series, it seems that the AR-part of the ARMA model might be stronger, since the ACF looks nicely like a damped sinusoid and in the PACF, the cutoffs are more evident than in the ACF, so let's try increasing p :

```
> arma21 <- arima(ts, order = c(2,0,1))
> arma21
```

```
Call:
arima(x = ts, order = c(2, 0, 1))

Coefficients:
         ar1      ar2     ma1  intercept
      0.8915  -0.2411  0.7061     0.6420
s.e.  0.0855   0.0856  0.0625     0.3208

sigma^2 estimated as 0.8772:  log likelihood = -272.01,  aic = 554.02
> tsdisplay(arma21$residuals, points = FALSE)
```

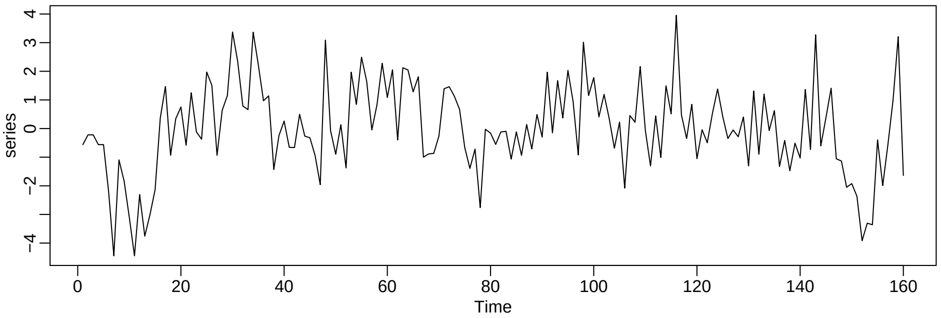


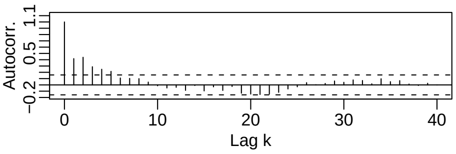
The residuals look stationary, all coefficients are significant and the AIC is the lowest so far. There is still some minor dependence left at lag 16, which could or could not be significant. Regarding the cutoffs in the PACF, one could also try an ARMA(3,1)- and an ARMA(4,1)-model. However, with increasing p also the aic increases (and the minor dependence at lag 16 does not go away), and for the ARMA(4,1) model, not all coefficients are significant anymore. Therefore, we choose to stick with the ARMA(2,1)-model.

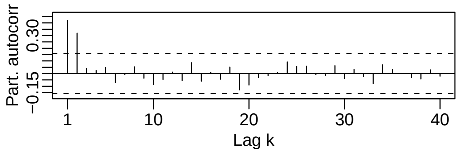
3. **a)**

```
> r.bel.lm <- lm(NURSING ~ ., data=beluga)
> summary(r.bel.lm)
Call:
lm(formula = NURSING ~ ., data = d.beluga)

Residuals:
     Min       1Q   Median       3Q      Max
-4.44568 -0.90180 -0.08505  1.09525  3.95477

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.5602842  0.5502170   1.018  0.31012
PERIOD       0.0001998  0.0031937   0.063  0.95020
BOUTS        0.8784967  0.3336237   2.633  0.00932 **
LOCKONS      2.3903512  0.2035042  11.746  < 2e-16 ***
DAYNIGHT    -0.3416237  0.2510156  -1.361  0.17550
---
Signif. codes:  0 `***' 0.001 `**' 0.01 `*' 0.05 `.' 0.1 ` ' 1

Residual standard error: 1.582 on 155 degrees of freedom
Multiple R-Squared: 0.842,     Adjusted R-squared: 0.8379
F-statistic: 206.5 on 4 and 155 DF,  p-value: < 2.2e-16
```

```
> d.resid <- ts(resid(r.bel.lm))
> plot(d.resid)
> acf(d.resid, lag=40)
> pacf(d.resid, lag=40)
```







The correlogram of the residuals shows that significant correlation is present. Consequently, all confidence intervals and tests in the output of lm can be wildly inaccurate. It is thus impossible for zoologists to conclude which explanatory variables are needed in the model.

b) Due to the partial autocorrelations present, an AR(2) model for the residuals makes sense. Note that the ordinary autocorrelations make up a dampened sine curve, a property typical of AR processes. We can use the Burg algorithm to estimate both AR parameters:

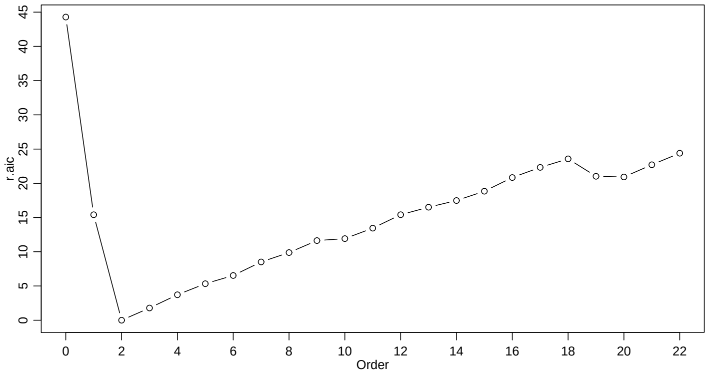
```
> r.burg <- ar(d.resid, method="burg", order.max=2, aic=F)
> str(r.burg)
```

in R, we obtain:

$\alpha_1 = 0.284, \alpha_2 = 0.321.$

Note: We can also use the AIC plot to determine the order of the process:

```
> r.aic <- ar(d.resid, method="burg")$aic
> plot(0:(length(r.aic)-1), r.aic, xlab="Order", type="b")
```



It seems that $p = 2$ is a good order to take.

c) We have

$$\begin{aligned} Y_t &= \beta_0 + \beta_1 \cdot t + \beta_2 X_{2,t} + \beta_3 X_{3,t} + \beta_4 X_{4,t} + E_t \qquad (t = 1, \ldots, 160) \\ \text{with} \quad & E_t = \alpha_1 E_{t-1} + \alpha_2 E_{t-2} + U_t \qquad U_t \text{ i.i.d. }, \; E[U_t] = 0, \; \text{Var}[U_t] = \sigma^2 \; , \end{aligned}$$

where $Y_t =$ `NURSING`, $X_{1,t} = t =$ `PERIOD`, $X_{2,t} =$ `BOUTS`, $X_{3,t} =$ `LOCKONS` and $X_{4,t} =$ `DAYNIGHT`.

Computing $Y_t^* = Y_t - \alpha_1 Y_{t-1} - \alpha_2 Y_{t-2}$:

$$\begin{aligned}
Y_t^* &= Y_t - \alpha_1 Y_{t-1} - \alpha_2 Y_{t-2} \\
&= \beta_0 + \beta_1 \cdot t + \beta_2 X_{2,t} + \beta_3 X_{3,t} + \beta_4 X_{4,t} + E_t \\
&\quad -\alpha_1\big(\beta_0 + \beta_1 \cdot (t-1) + \beta_2 X_{2,t-1} + \beta_3 X_{3,t-1} + \beta_4 X_{4,t-1} + E_{t-1}\big) \\
&\quad -\alpha_2\big(\beta_0 + \beta_1 \cdot (t-2) + \beta_2 X_{2,t-2} + \beta_3 X_{3,t-2} + \beta_4 X_{4,t-2} + E_{t-2}\big) \\
&= \beta_0(1-\alpha_1-\alpha_2) + \beta_1(t - \alpha_1(t-1) - \alpha_2(t-2)) \\
&\quad +\beta_2(X_{2,t} - \alpha_1 X_{2,t-1} - \alpha_2 X_{2,t-2}) + \ldots + E_t - \alpha_1 E_{t-1} - \alpha_2 E_{t-2} \\
&= \beta_o^* + \beta_1 X_{1,t}^* + \beta_2 X_{2,t}^* + \beta_3 X_{3,t}^* + \beta_4 X_{4,t}^* + U_t
\end{aligned}$$

The explanatory variables and the target must all be transformed as follows:

$$x_t^* = x_t - \widehat{\alpha}_1 x_{t-1} - \widehat{\alpha}_2 x_{t-2} = x_t - 0.284 \cdot x_{t-1} - 0.321 \cdot x_{t-2}$$

d) (*) The transformation, and the subsequent normal regression, can be performed in R using the following code. Note that the residuals now no longer exhibit correlation.

```
> t.ar <- r.burg$ar
> ## Transform the entire multivariate time series
> d.beluga.tr <- d.beluga - t.ar[1]*lag(d.beluga,-1) - t.ar[2]*lag(d.beluga,-2)
> ## Set new (meaningful) colnames
> colnames(d.beluga.tr) <- paste(colnames(d.beluga),".tr",sep="")
[1] "PERIOD.tr"   "BOUTS.tr"    "NURSING.tr"  "LOCKONS.tr"  "DAYNIGHT.tr"
> t.intercept <- rep((1-t.ar[1]-t.ar[2]),nrow(d.beluga.tr))
> r.lm.tr <- lm(NURSING.tr ~ -1 + t.intercept + PERIOD.tr + BOUTS.tr +
+              LOCKONS.tr + DAYNIGHT.tr, data=d.beluga.tr)
> plot(r.lm.tr$resid)
> acf(r.lm.tr$resid)
> pacf(r.lm.tr$resid)
```

e) The procedure `gls()` can be used for much more general models than those you have already seen. The argument `correlation` can be used for specifying the correlation structure of the residuals. In principle an AR(p) model is merely a special case of the ARMA(p, q) model taking $q = 0$. This explains the overly complex expression `corARMA(value=c(...,...), p=2, q=0, fixed=F)`. The AR coefficients computed in Part b) can be used as starting values by specifying them in the argument `value`. Errors in different time periods can be specified as being correlated by means of the argument `form=` $\sim$ `PERIOD` of `corARMA`. This is necessary, as the entries in the data matrix can be arranged in any way.

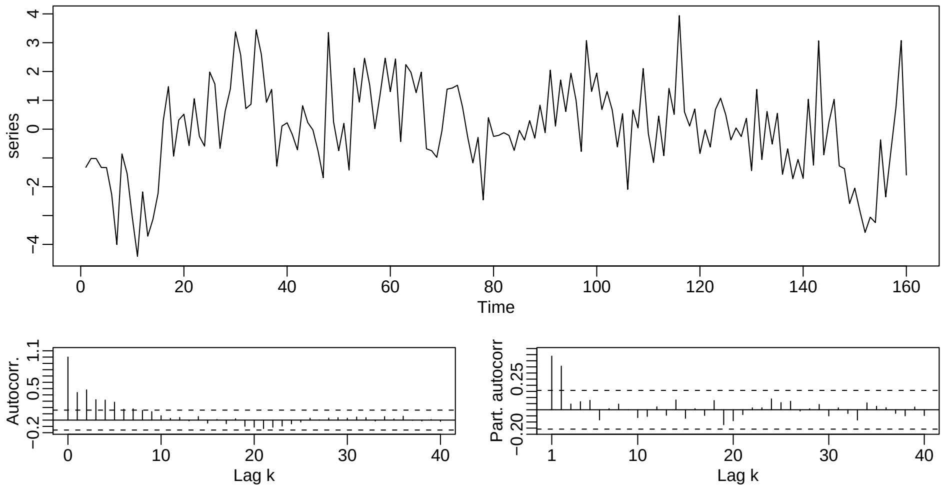
R-output from `summary(r.bel.gls)`:

```
Generalized least squares fit by maximum likelihood
  Model: NURSING ~ BOUTS + LOCKONS + DAYNIGHT + PERIOD
  Data: d.beluga
      AIC      BIC   logLik
  560.396 584.9974 -272.198

Correlation Structure: ARMA(2,0)
 Formula: ~PERIOD
 Parameter estimate(s):
     Phi1      Phi2
0.2739964 0.3653668

Coefficients:
                 Value Std.Error   t-value p-value
(Intercept)  1.3218871 0.7678364  1.721574  0.0871
BOUTS        0.2961684 0.3370588  0.878685  0.3809
LOCKONS      2.5681923 0.1964012 13.076257  <.0001
DAYNIGHT    -0.3080293 0.1549160 -1.988363  0.0485
PERIOD       0.0024982 0.0062754  0.398090  0.6911

 Correlation:
        (Intr) BOUTS  LOCKON DAYNIG
BOUTS    -0.303
```

```
LOCKONS  -0.101 -0.811
DAYNIGHT -0.014 -0.135  0.067
PERIOD   -0.607 -0.233  0.251  0.024

Standardized residuals:
        Min          Q1         Med          Q3         Max
-2.80055625 -0.58763749  0.01738824  0.65602061  2.49854120

Residual standard error: 1.577031
Degrees of freedom: 160 total; 155 residual
```

These coefficient estimates differ markedly from those in Part a). We obtain $\alpha_1 = 0.274$ and $\alpha_2 = 0.365$, which can be found in the above R output at `Parameter estimate(s)` (here labelled as `Phi1` and `Phi2`). In particular note that the standard errors of the explanatory variables sometimes differ greatly from those in the regression model.

Residual analysis:



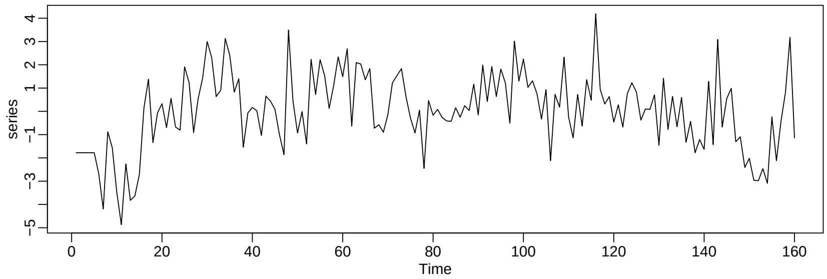
There are only small differences to the model using ordinary regression. This is because residuals denote the difference between observations and model-derived fitted values – and the least squares estimates of coefficients do make sense here. It is merely the standard errors of the least squares method that are wrong. The residuals form an AR(2) process; thus the chosen correlation structure is correct.

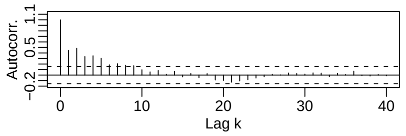
f) Successively eliminating redundant variables (`PERIOD, BOUTS` and then `DAYNIGHT`) reduces the model.

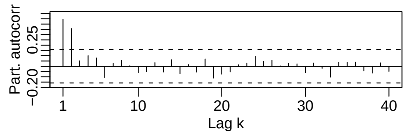
R output from `summary(r.red.bel.gls)`:

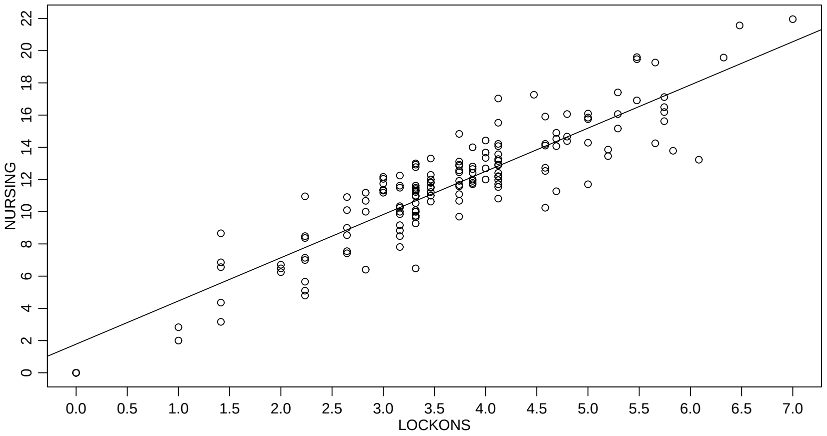
```
  Generalized least squares fit by maximum likelihood
   Model: NURSING ~ LOCKONS
   Data: d.beluga
        AIC      BIC    logLik
   559.1555 574.5314 -274.5778

 Correlation Structure: ARMA(2,0)
  Formula: ~PERIOD
  Parameter estimate(s):
      Phi1      Phi2
 0.2803981 0.3696418

 Coefficients:
                Value Std.Error   t-value p-value
 (Intercept) 1.778230 0.5048868  3.522037   6e-04
 LOCKONS     2.682246 0.1147227 23.380250  <.0001

  Correlation:
         (Intr)
 LOCKONS -0.804

 Standardized residuals:
        Min          Q1         Med          Q3         Max
-3.01255719 -0.57430640  0.05979804  0.69560485  2.59582932

 Residual standard error: 1.614887
 Degrees of freedom: 160 total; 158 residual
```

Note:
As you are not using an ordinary lm object, you cannot use the function `step()`. You will need to eliminate variables individually until all remaining variables are significant.
The analysis of residuals does not show any breach of the assumptions on errors, i.e. the residuals do still constitute an AR(2) process as assumed in the construction of the model. The fitted line is given in the last plot.









R commands for these plots:
> `plot(ts(resid(r.red.bel.gls)))`
> `acf(ts(resid(r.red.bel.gls)))`
> `pacf(ts(resid(r.red.bel.gls)))`
and
> `plot(d.beluga[,4], d.beluga[,3], xlab="LOCKONS", ylab="NURSING")`
> `abline(r.red.bel.gls)`