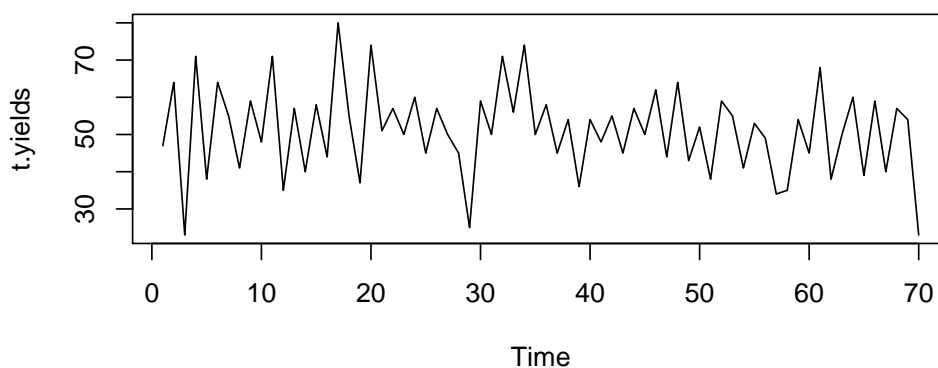


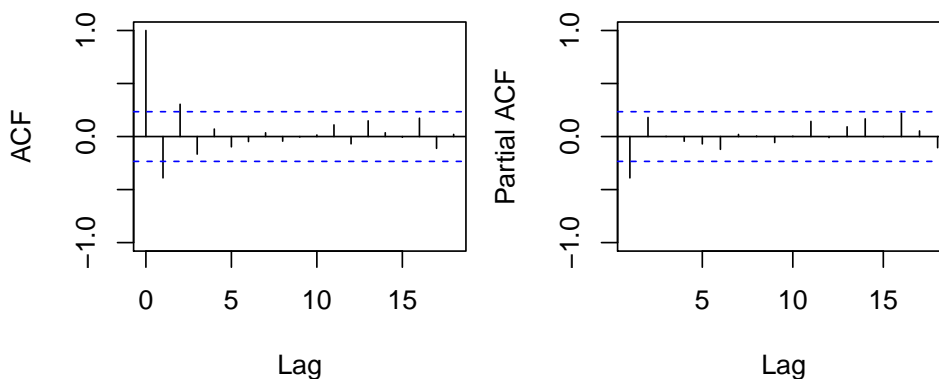
Solution to Series 5

```
1. a) > yields <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/yields.dat",
                             header = FALSE)
> t.yields <- ts(yields[, 1])
> plot(t.yields)
```



The series looks stationary, but the mean is different from 0, it must therefore be a shifted AR-process. To see whether the data really is generated from a (shifted) AR-process, we look at the autocorrelation plots:

```
> par(mfrow=c(1,2))
> acf(t.yields, ylim=c(-1,1))
> pacf(t.yields, yli=c(-1,1))
```



The exponential decay of the magnitude of the coefficients in the acf plot, tells us that data really could be generated from an AR-process, the pacf plot tells us that its order p must have been 1. Overall, we conclude that the underlying model could be an AR(1)-process. The general AR(1) process is

$$X_t - \mu = \alpha(X_{t-1} - \mu) + E_t$$

$$E_t \text{ i.i.d. , } \mathcal{E}[E_t] = 0, \quad \text{Var}[E_t] = \sigma^2, \quad E_t \text{ causal}$$

b) Computing the parameters using the Yule-Walker equations:

- The first estimated autocorrelation is given by $\hat{\rho}(1) = -0.390$ (R-Code: `acf(t.yields, plot=F)`). To fit an AR(1) model, the Yule-Walker equations gives us the following:

$$\hat{\rho}(1) = \hat{\alpha}_1 \cdot \hat{\rho}(0), \quad \hat{\rho}(0) = 1 \quad \implies \quad \hat{\alpha}_1 = \hat{\rho}(1) = -0.390$$

All the others $\alpha_i = 0$ for $i \neq 1$, since we fitted an AR(1) process.

- We estimate μ to be $\hat{\mu}_X = \frac{1}{n} \sum_{t=1}^n X_t = 51.129$

- We estimate σ^2 to be $\hat{\sigma}^2 = \hat{\sigma}_X^2 \cdot (1 - \hat{\alpha}^2)$ (This follows from taking variances on both sides of the equation $X_t = \alpha_1 \cdot X_{t-1} + E_t$ and then solving for σ^2 .)

Thus we write the fitted model as

$$\begin{aligned} X_t - 51.129 &= -0.390(X_{t-1} - 51.129) + E_t && \text{or} \\ X_t &= 71.069 - 0.390 \cdot X_{t-1} + E_t \end{aligned}$$

Computing the parameters using R:

```
> r.yw <- ar(t.yields, method="yw", order.max=1)
```

By calling `str(r.yw)`, we can see all the elements of the object "r.yw", including `r.yw$ar = -0.39`, `r.yw$x.mean = 51.1` and `r.yw$var.pred = 122`.

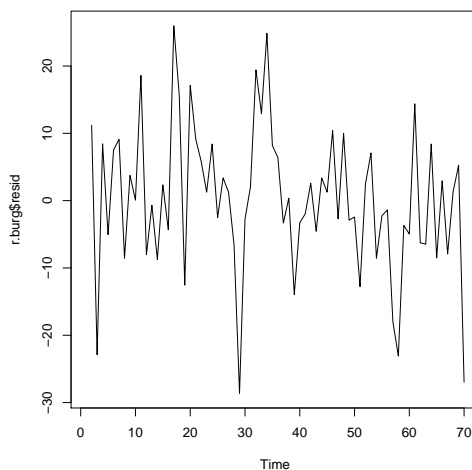
c) Burg algorithm:

```
> r.burg <- ar(t.yields, method="burg", order.max=1)
```

When we look at the structure of `r.burg` we see: `r.burg$ar = -0.407`, `r.burg$x.mean = 51.1` and `r.burg$var.pred = 117` for the estimated variance of innovations.

Analysis of residuals:

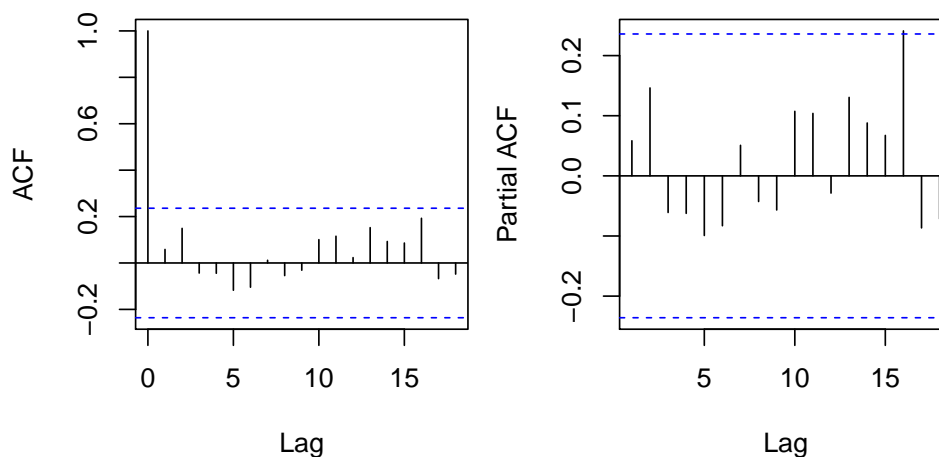
```
> plot(r.burg$resid)
```



```
> par(mfrow=c(1,2))
```

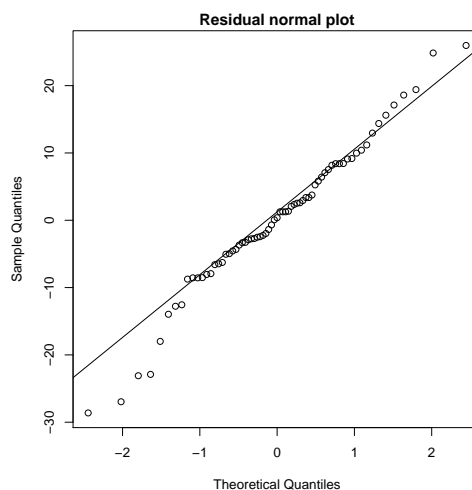
```
> acf(r.burg$resid, na.action=na.omit)
```

```
> pacf(r.burg$resid, na.action=na.omit)
```



```
> qqnorm(as.numeric(r.burg$resid), main="Residual normal plot")
```

```
> qqline(as.numeric(r.burg$resid))
```



The residual plots exhibit stationarity and no longer show any dependence. All autocorrelations lie inside the confidence band.

The residuals are not perfectly normally distributed, but this is not a big problem, since this is not really an assumption. The main idea, why we would like to have normally distributed innovations is because we know that all methods perform best on data which are approximately Gaussian.

d) Maximum-Likelihood method:

Using `str(r.mle <- ar(yields, method="mle", order.max=1))`, we obtain the parameters $\hat{\alpha} = -0.419$, $\hat{\mu} = 51.3$ and $\hat{\sigma}^2 = 117$. However, we do not see whether or not the algorithm actually converges. It is thus advisable to compute the ML estimates using the procedure `arima()`:

```
> r.mle <- arima(yields, order=c(1,0,0), include.mean=T)
```

Then we can call `r.mle$code` to establish whether or not the algorithm converged (`code=0` means that convergence has occurred, cf. the R help file on the procedure `arima()`).

R output:

```
> r.mle <- arima(yields, order=c(1,0,0), include.mean=T)
> r.mle
```

Call:

```
arima(x = yields, order = c(1, 0, 0), include.mean = T)
```

Coefficients:

```
      ar1  intercept
      -0.4191    51.2658
s.e.   0.1129     0.9137
```

σ^2 estimated as 116.6: log likelihood = -265.98, aic = 537.96

We can now use the estimated standard deviation of `intercept` to compute a confidence interval for μ . Incidentally, `r.mle$var.coef` returns the full estimated covariance matrix of the coefficients:

```
      ar1  intercept
ar1      0.012735551 -0.002962835
intercept -0.002962835  0.834897093
```

So the variance of the MLE for μ is estimated at 0.835 (giving us a standard error of 0.9137, cf. above). This now leads to a confidence interval for μ :

$$[51.259 - 1.96\sqrt{0.835}, 51.259 + 1.96\sqrt{0.835}] = [49.47, 53.05]$$

This interval can also be computed directly using the following command in R:

```
> r.mle$coef["intercept"] + c(-1.96,1.96)* sqrt(r.mle$var.coef["intercept", "intercept"])
[1] 49.47491 53.05672
> ## or easier
> confint(r.mle)
```

	2.5 %	97.5 %
ar1	-0.6402648	-0.1978935
intercept	49.4749458	53.0566899

In summary, these three methods lead to the following estimates of coefficients:

Method	$\hat{\alpha}$	$\hat{\mu}$	$\hat{\sigma}^2$
Yule-Walker	-0.390	51.129	122.0
Burg	-0.407	51.129	116.7
Maximum Likelihood	-0.419	51.266	116.6

- e) Taking the AR(1) structure of the time series into account, the estimated variance of μ satisfies the following relation:

$$\begin{aligned}\widehat{\text{Var}}(\hat{\mu}) &= \frac{1}{n^2} \cdot \gamma(0) \cdot \left(n + 2 \cdot \sum_{k=1}^{n-1} (n-k)\rho(k) \right) \\ &= \frac{1}{70^2} \cdot \gamma(0) \cdot \left(70 + 2 \cdot \sum_{k=1}^{69} (70-k)\hat{\alpha}_1^k \right)\end{aligned}$$

The Burg algorithm gives us $\widehat{\text{Var}}(\hat{\mu}) = 0.854$. Thus we have a 95% confidence interval for μ :

$$\hat{\mu} \pm 1.96 \cdot \text{se}(\hat{\mu}) = 51.129 \pm 1.96 \cdot \sqrt{0.854} = [49.32, 52.94]$$

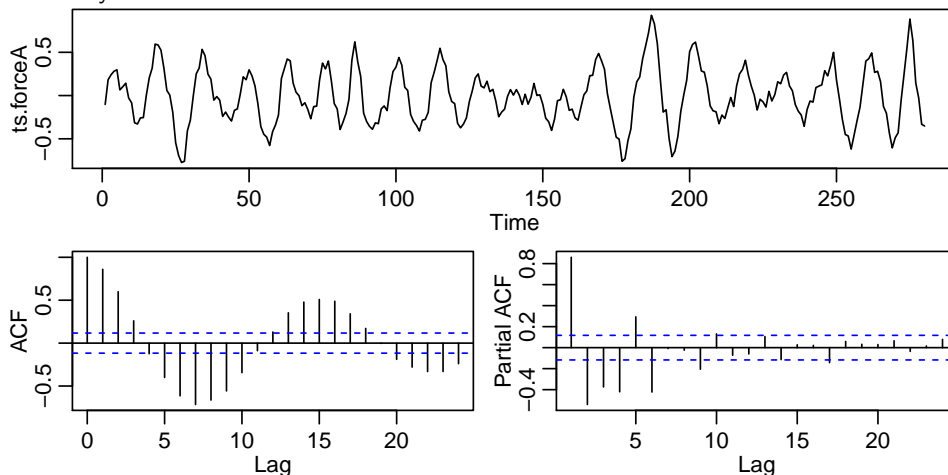
This confidence interval is shorter than the interval [48.36, 53.90] that we obtained in Problem 3 of Sheet 4. The MLE confidence interval in part c) has about the same length, but is shifted somewhat.

Note:

We can specifically compute $\widehat{\text{Var}}(\hat{\mu})$ with R as follows:

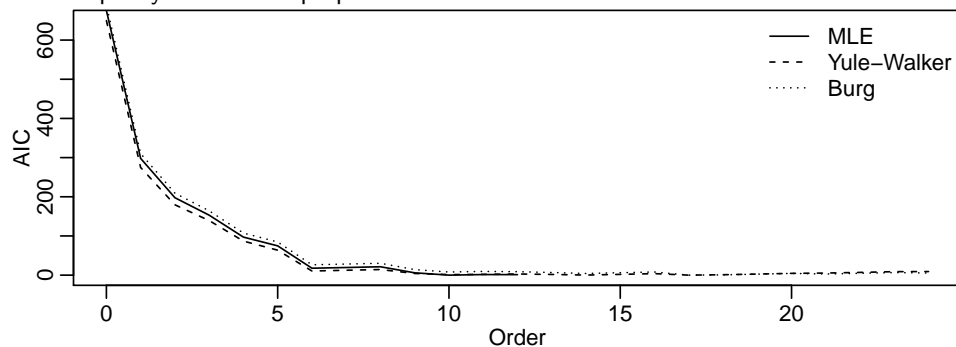
```
> n <- 70
> gamma0 <- r.burg$var.pred / (1 - r.burg$ar^2)
> ## or
> gamma0 <- 69/70 * var(t.yields) # to get the "biased" variance
> alpha.est <- r.burg$ar
> tmp <- ((n-1):1)*(alpha.est^(1:(n-1)))
> var.est <- 1 / (n^2) * gamma0 * (n + 2 * sum(tmp))
> var.est
[1] 0.8540978
> 51.129 + c(-1,1) * 1.96 * sqrt(var.est)
[1] 49.31762 52.94038
```

2. a) The experimental setup leads us to expect a period of 2 seconds. Since our measurements are spaced apart by 0.15 seconds, each 2-second period covers $2/0.15 = 13.3$ measurements. This period, however, is subject to fluctuations, which are visible in both the time series plot and the correlogram of ordinary autocorrelations:



- b) The PACF (see Part a)) is clearly significant for lags up to 6; also lags 9, 10 and 17 are slightly significant. We could therefore use an AR model of order 6, 9 or 17. Order 17 seems quite high (and hence difficult to interpret), so orders 6 or 9 would be preferred. However, we cannot see by eye whether order 6 or 9 is really sufficient; this can be done with a residual analysis, see Part c).

The AIC plot yields similar proposes:



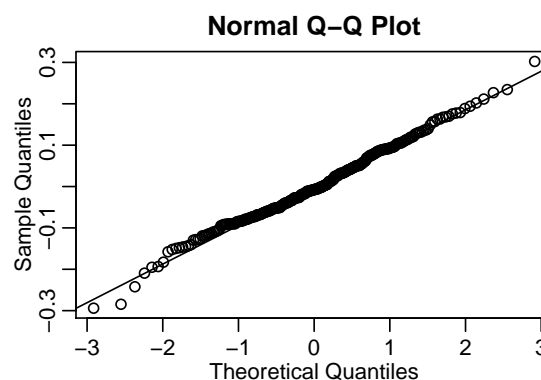
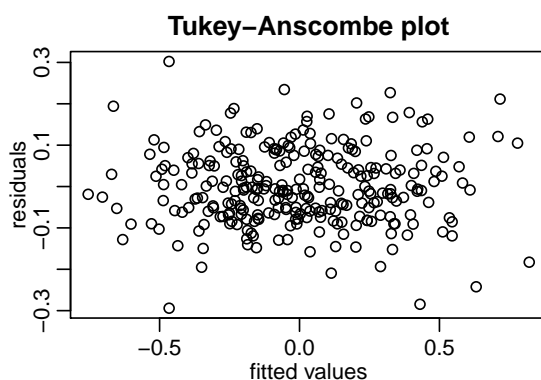
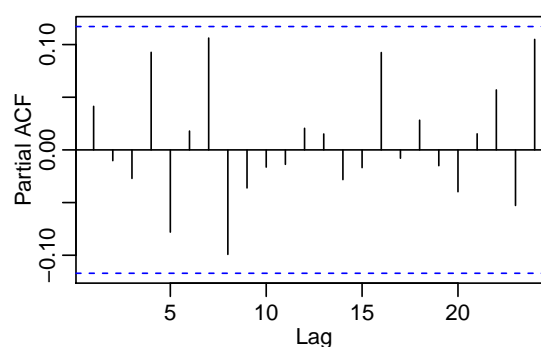
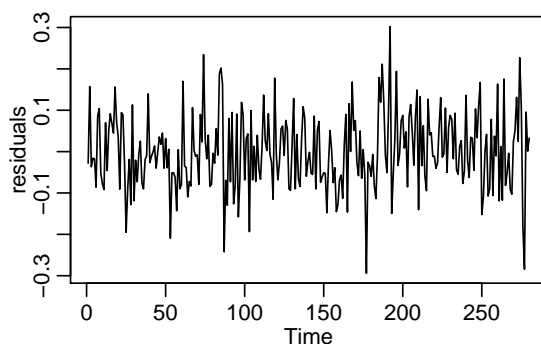
There is a big jump of the AIC at order 6, and a smaller one at order 9; hence one of these orders would be plausible to fit the given time series. However, the minimum of the AIC is attained at order $p = 10$ (MLE), $p = 17$ (Yule-Walker) or $p = 17$ (Burg), respectively.

- c) Fitting the AR model (with order $p = 9$):

```
> p <- 9
> ar.force <- arima(ts.forceA, order = c(p, 0, 0), method = "ML")
```

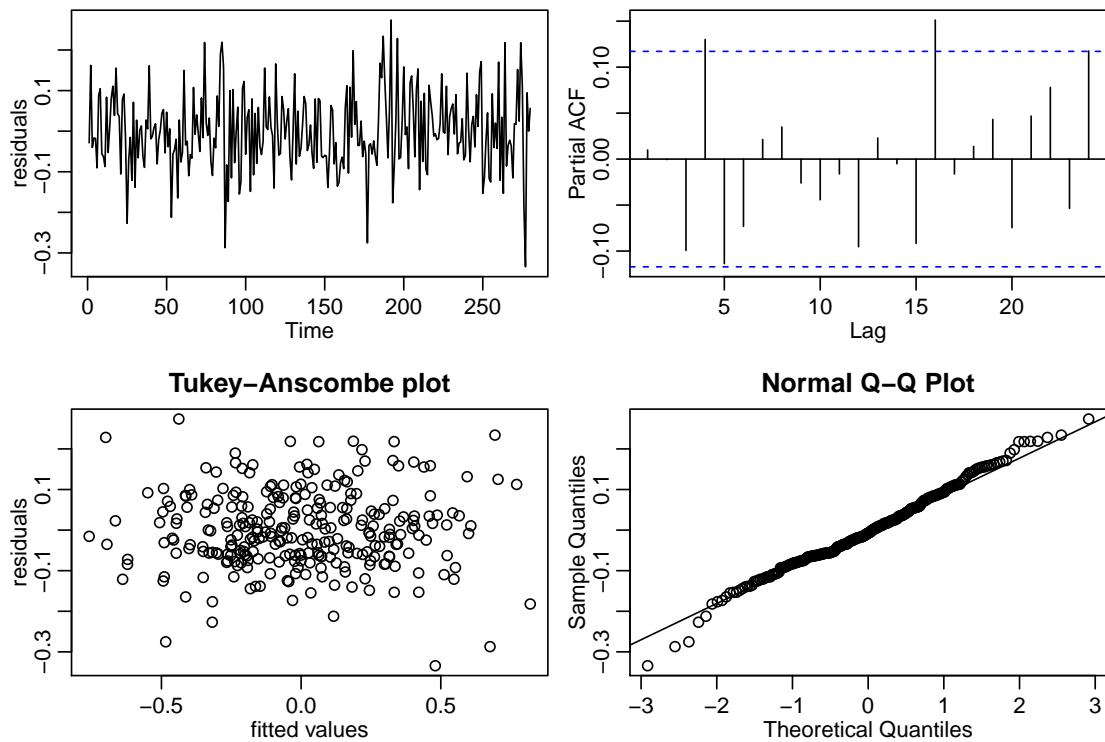
The residuals of this model look as follows:

```
> par(mfrow = c(2, 2), mar = c(3, 3, 2, 0.1))
> plot(ar.force$residuals, ylab = "residuals")
> acf(ar.force$residuals, type = "partial", plot = TRUE, main = "")
> plot(ts.forceA - ar.force$residuals, ar.force$residuals, xlab = "fitted values",
      ylab = "residuals", main = "Tukey-Anscombe plot")
> qqnorm(ar.force$residuals)
> qqline(ar.force$residuals)
```



The model with order 9 is acceptable. Its residuals are normally distributed, they have constant variance, and the correlogram as well as the Tukey-Anscombe plot do not indicate any dependence.

If we take order $p = 6$ instead of order $p = 9$, the residuals look different. They still show some (weak) correlation, indicating that the order is not sufficient for this time series:



```
d) > force.pred <- predict(ar.force, n.ahead = 40)
> plot(window(ts.force, start = 250), ylab = "")
> lines(force.pred$pred, lty = 2)
> lines(force.pred$pred + 1.96*force.pred$se, lty = 3)
> lines(force.pred$pred - 1.96*force.pred$se, lty = 3)
```

