Solution to Series 3

1. In general, if the autocorrelations are positive for small lags and effectively zero for the others, the time series has the tendency to stay more or less around the same value for a few timesteps before any dramatic changes in value happen. There is some inertia in the time series.

If, on the other hand, the autocorrelation is (strongly) negative for small lags, this indicates a sort of oscillation where the time series tends to want to oscillate around its mean quickly and often.

These kinds of interpretations are easy to make if you think about how the lagged scatterplot would look like for the different lags. Think about how the lagged scatterplot would look like for extreme values of positive or negative correlation. What would that say about what sort of value is likely to follow after the lag? What if the scatterplot for lag 1 shows a correlation close to 1 and all higher lags have tiny correlations?

After understanding this, you should have an easier time solving this exercise.

The **time series plot 1** belongs to correlogram II. The autocorrelations reflect the periodic fluctuations of the series.

The **time series plot 2** belongs to correlogram I. On average, values from this time series are around the same as their predecessors. Thus the autocorrelations are positive for small k, and then become 0.

The **time series plot 3** belongs to correlogram III. The strong oscillation exhibited by this time series is evident in the changing signs of its autocorrelations.

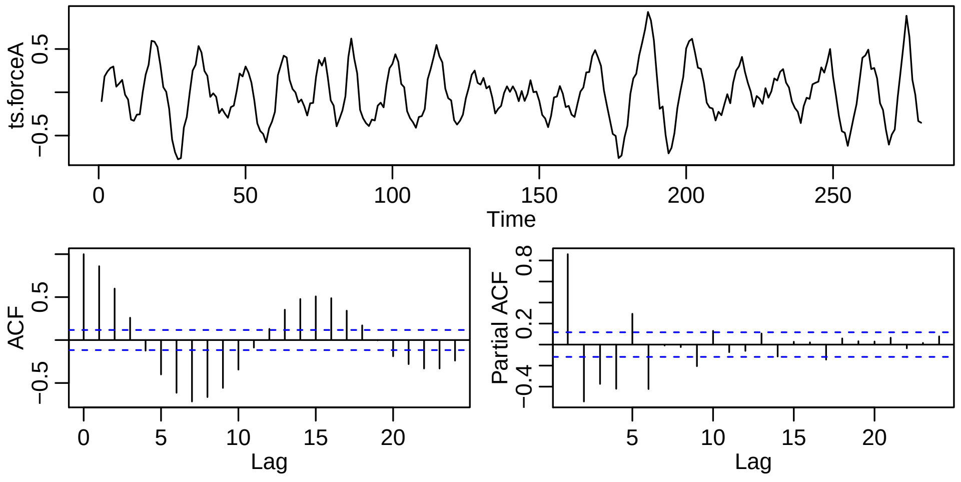
In order to identify which correlogram now belongs to which set of partial autocorrelations, we use the fact that the standard ($\widehat{\rho}(k)$) and partial ($\widehat{\pi}(k)$) autocorrelations are identical at lag 1:

$$\widehat{\pi}(1) = \widehat{\rho}(1)$$

It is thus clear which set of partial correlations belongs to which correlogram:

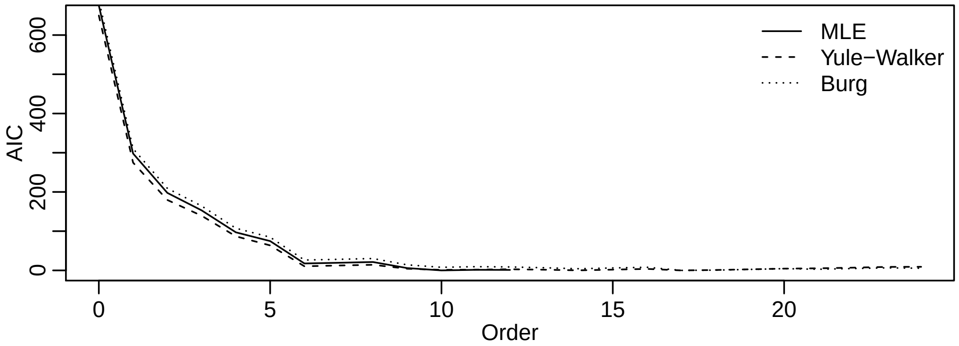
Time series	std. autocorr.	part. autocorr.
1	II	C
2	I	A
3	III	B

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:



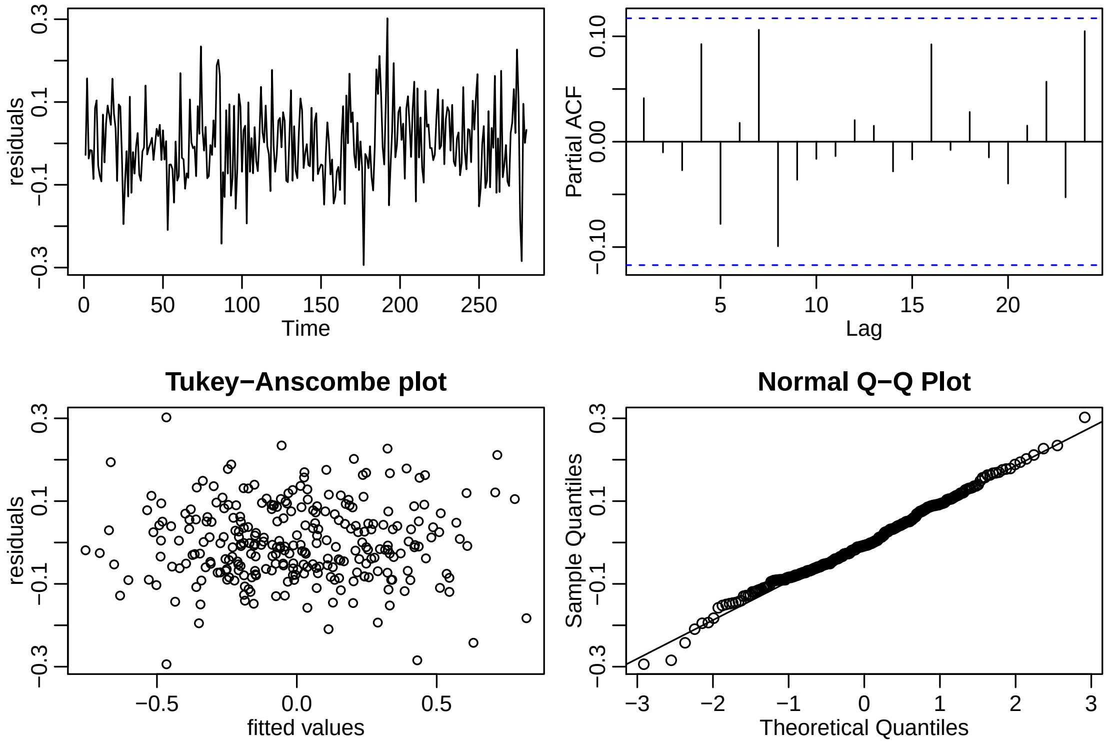
There is a big jump of the AIC at order 6, and a smaller one at order 9; hence one of these orders would be plausible to fit the given time series. However, the minimum of the AIC is attained at order $p = 10$ (MLE), $p = 17$ (Yule-Walker) or $p = 17$ (Burg), respectively.

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

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

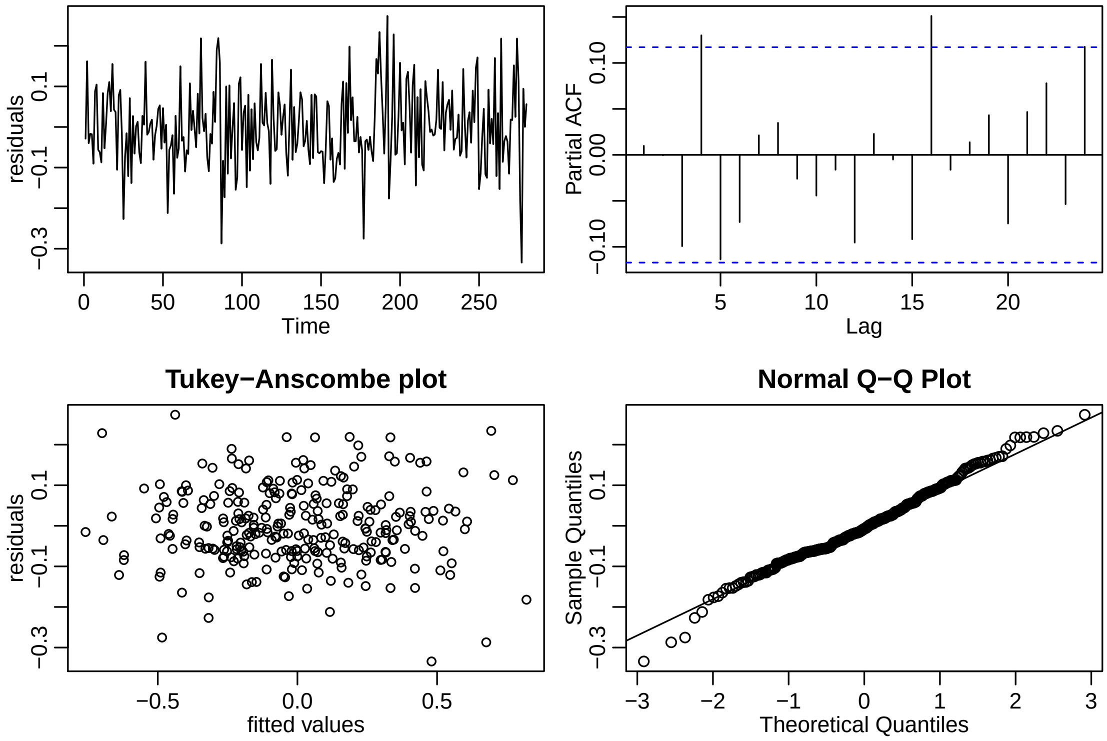
The residuals of this model look as follows:

```
> 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)
```



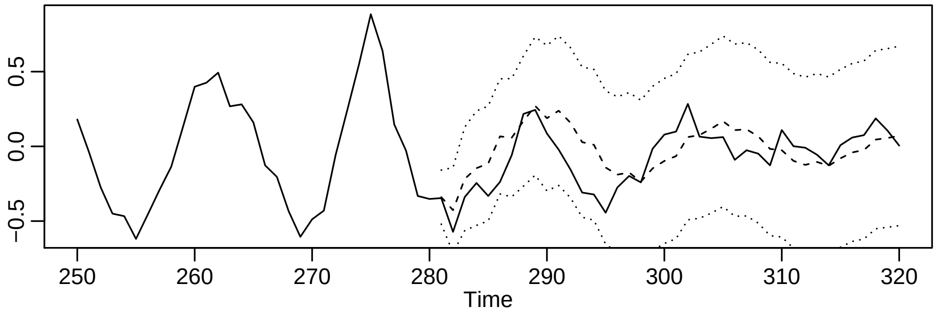
The model with order 9 is acceptable. Its residuals are normally distributed, they have constant variance, and the correlogram as well as the Tukey-Anscombe plot do not indicate any dependence.

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:



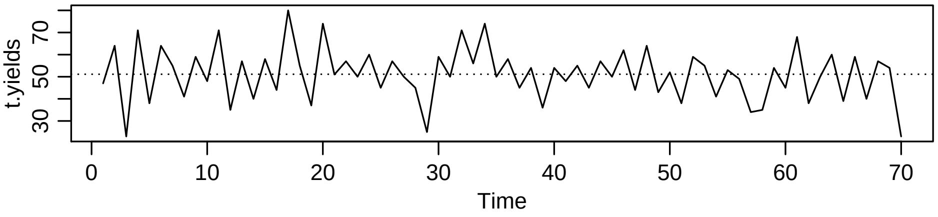
d)

```
> 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)
```



3. a) Plotting and calculating the mean:

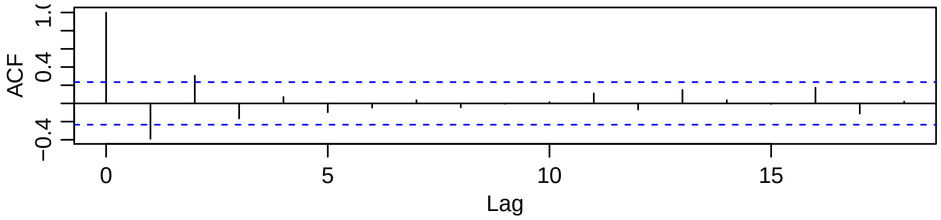
```
> mu <- mean(t.yields)
> plot(t.yields)
> abline(h = mu, lty=3)
```



We can regard this time series as being stationary.

b) Plotting the ACF:

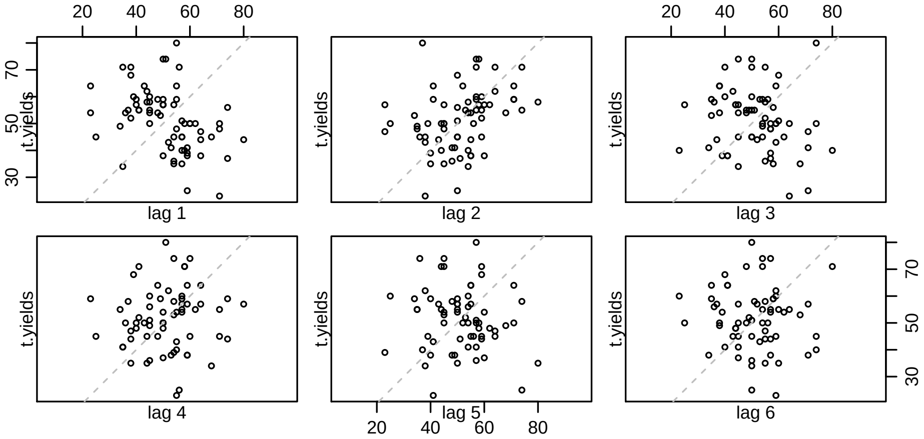
```
> acf(t.yields, plot = TRUE)
```



The correlogram shows us that for lags $k \geq 3$, the estimated autocorrelations $\widehat{\rho}\langle k\rangle$ do not differ significantly from 0. The first of these autocorrelations is negative; as the time series oscillates very noticeably, this negativity is not at all surprising.

Looking at lagged scatterplots:

```
> lag.plot(t.yields, lag = 6, layout = c(2, 3), do.lines = FALSE)
```



In the lagged scatterplot with lag 1 the pairs $[x_t, x_{t+1}]$ show the negative linear relationship we expected from the correlogram. For lag 2, however, the lagged scatterplot shows up a positive linear relationship, and for lag $k \geq 4$ we see no further correlation. The pairs $[x_t, x_{t+3}]$ (lagged scatterplot at lag 3) still have a slightly negative connection, but the correlogram tells us that we can assume $\widehat{\rho}\langle 3\rangle = 0$.

c) The variance of the arithmetic mean $\widehat{\mu}$ is

$$\operatorname{Var}(\widehat{\mu}) = \frac{1}{n^2}\,\gamma(0)\left(n + 2\sum_{k=1}^{n-1}(n-k)\,\rho(k)\right) .$$

From the correlogram in Part b) we see that the estimated autocorrelations $\widehat{\rho}\langle k\rangle$ do not differ significantly from 0 for lags $k \geq 3$. Thus we can set all the autocorrelations $\rho(k)$ for $k \geq 3$ to 0. We obtain

$$\operatorname{Var}(\widehat{\mu}) = \frac{1}{n^2}\,\gamma(0)\Big(n + 2(n-1)\rho(1) + 2(n-2)\rho(2)\Big).$$

To estimate the variance of $\widehat{\mu}$, we replace $\gamma(0), \rho(1)$ and $\rho(2)$ by their estimates.

R code:

```
> n <- length(t.yields)
> gamma0 <- var(t.yields) * (n - 1)/n
> rho <- acf(t.yields, plot=F)$acf
> Var.mu <- n^(-2) * gamma0 * (n + 2*sum((n - 1:2)*rho[2:3]))
```

This yields an estimated variance of $\widehat{\operatorname{Var}}(\widehat{\mu}) = 1.643$.

The bounds of an approximate 95% confidence interval for the mean yield are then given by

$$\widehat{\mu} \pm 1.96 \cdot \operatorname{se}(\widehat{\mu}) = \widehat{\mu} \pm 1.96 \cdot \sqrt{\widehat{\operatorname{Var}}(\widehat{\mu})} .$$

In our case, we get a confidence interval of $[48.62, 53.64]$.

If we assume independence, the variance of $\widehat{\mu}$ is estimated as

$$\widehat{\mathrm{Var}}(\widehat{\mu}) = \frac{1}{n^2}\sum_{s=1}^{n}\widehat{\mathrm{Var}}(X_s) = \frac{\widehat{\gamma}(0)}{n} = 1.997 \ .$$

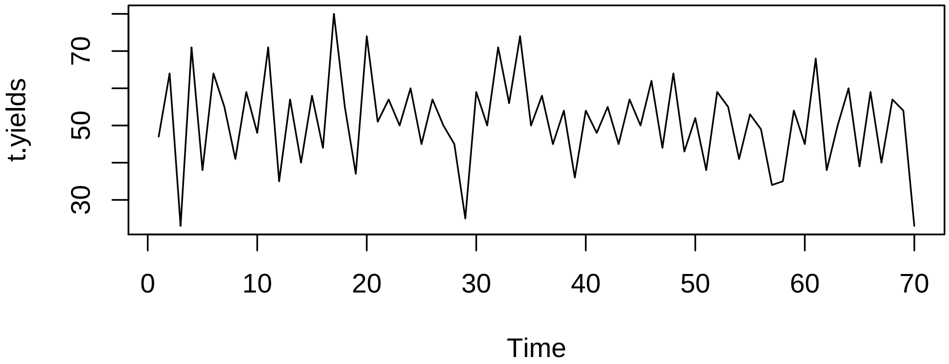
Under this independence assumption, therefore, an approximate 95% confidence interval for the mean yield is given by

$$[\widehat{\mu} - 1.96 \cdot \mathrm{se}(\widehat{\mu}), \widehat{\mu} + 1.96 \cdot \mathrm{se}(\widehat{\mu})] = [48.36, 53.90] \ .$$

Thus the correct specification of the independence structure here leads to a confidence interval which is 10% narrower.

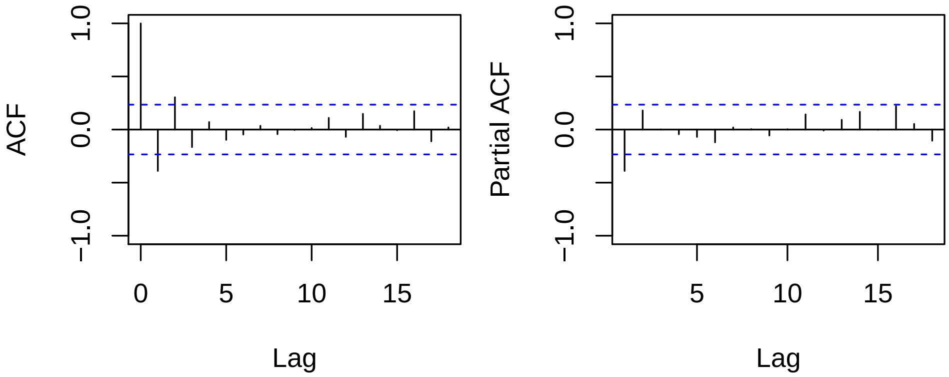
4. a)

```
> 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 oder p must have been 1. Overall, we conclude that the underlying model could be an AR(1)-process. The general AR(1) process is

$$\begin{aligned} X_t - \mu &= \alpha(X_{t-1} - \mu) + E_t \\ & \quad E_t \text{ i.i.d. }, \quad \mathcal{E}[E_t] = 0, \quad \mathrm{Var}[E_t] = \sigma^2, \quad E_t \text{ causal} \end{aligned}$$

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

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

$$\widehat{\rho}(1) = \widehat{\alpha}_1 \cdot \widehat{\rho}(0)\,, \quad \widehat{\rho}(0) = 1 \qquad \Longrightarrow \qquad \widehat{\alpha}_1 = \widehat{\rho}(1) = -0.390$$

  All the others $\alpha_i = 0$ for $i \neq 1$, since we fitted an AR(1) process.
- We estimate μ to be $\widehat{\mu}_X = \frac{1}{n}\sum_{t=1}^{n} X_t = 51.129$
- We estimate σ^2 to be $\widehat{\sigma}^2 = \widehat{\sigma}_X^2 \cdot (1 - \widehat{\alpha}^2)$ (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

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

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`.

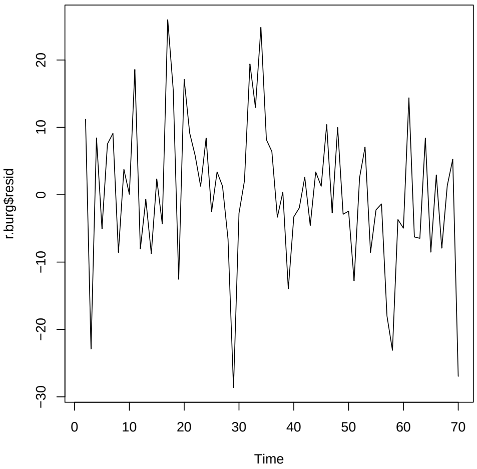
c) Burg algorithm:

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

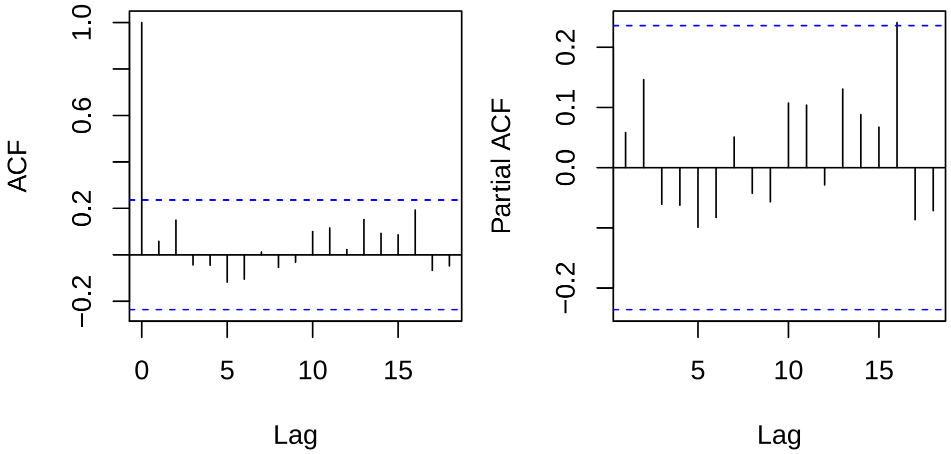
When we look at the structure of `r.burg` we see: `r.burg$ar = -0.407`, `r.burg$x.mean = 51.1` and `r.burg$var.pred = 117` for the estimated variance of innovations.

Analysis of residuals:

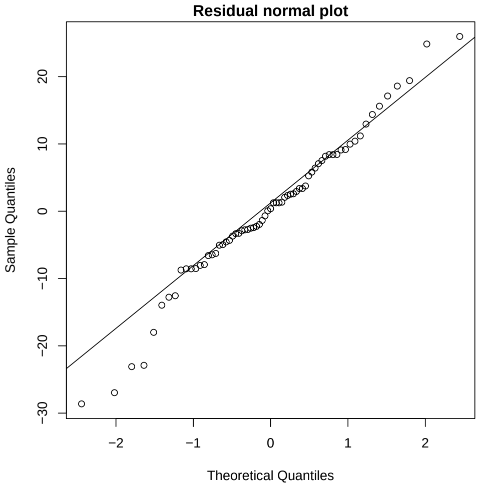
```
> plot(r.burg$resid)
```



```
> 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 problems, 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 $\widehat{\alpha} = -0.419, \widehat{\mu} = 51.3$ and $\widehat{\sigma}^2 = 117$. 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

sigma^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	$\widehat{\alpha}$	$\widehat{\mu}$	$\widehat{\sigma}^2$
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:

$$\begin{aligned}\widehat{\mathrm{Var}}(\widehat{\mu}) &= \frac{1}{n^2}\cdot\gamma(0)\cdot\left(n+2\cdot\sum_{k=1}^{n-1}(n-k)\rho(k)\right)\\ &= \frac{1}{70^2}\cdot\gamma(0)\cdot\left(70+2\cdot\sum_{k=1}^{69}(70-k)\widehat{\alpha_1}^k\right)\end{aligned}$$

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

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

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 $\widehat{\mathrm{Var}}(\widehat{\mu})$ 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
```