MATH 4070: ARMA Case Study and ARMIA

Whitney Huang, Clemson University

Contents

Ireland wind data case study . . . 1
- Load the dataset . . . 1
- Extract and plot the data . . . 3
- Deseasonalization: Harmonic Regression . . . 4
- ACF Plots: Original and Deseasonalized Series . . . 5
- Apply transformation to make wind speed more Gaussian like . . . 6
- Now take square roots of the original data and deseasonalizeagain! . . . 7
- Checking Normality ACF/PACF . . . 8
- Model identifcation, fitting, and selection . . . 9
  - Let's first fit an AR(1) . . . 9
  - Fit an AR(2) model . . . 12
  - Fit an ARMA(1,1) model . . . 15
- Use AIC to conduct model selection . . . 21
- Forecasting . . . 22
- Visualizing the Forecasts . . . 23

ARIMA . . . 25
- Monthly Price of Oil: January 1986–January 2006 . . . 25
- An explosive AR model . . . 25
- ARIMA(1,1,0) . . . 27

Ireland wind data case study

Acknowledgement: These materials are largely based on Dr. Peter Craigmile's time series class materials, with some modifications.

Load the dataset

We can load the Ireland wind data using the *gstat* package (Spatial and Spatio-Temporal Geostatistical Modelling, Prediction, and Simulation in `R`) and access the *wind* dataset. Type `?wind` to get more information.

```r
library(gstat)
data(wind)
head(wind)
```

```
##   year month day   RPT   VAL   ROS   KIL   SHA  BIR   DUB   CLA   MUL   CLO
## 1   61     1   1 15.04 14.96 13.17  9.29 13.96 9.87 13.67 10.25 10.83 12.58
## 2   61     1   2 14.71 16.88 10.83  6.50 12.62 7.67 11.50 10.04  9.79  9.67
## 3   61     1   3 18.50 16.88 12.33 10.13 11.17 6.17 11.25  8.04  8.50  7.67
## 4   61     1   4 10.58  6.63 11.75  4.58  4.54 2.88  8.63  1.79  5.83  5.88
## 5   61     1   5 13.33 13.25 11.42  6.17 10.71 8.21 11.92  6.54 10.92 10.34
## 6   61     1   6 13.21  8.12  9.96  6.67  5.37 4.50 10.67  4.42  7.17  7.50
##     BEL   MAL
## 1 18.50 15.04
## 2 17.54 13.83
## 3 12.75 12.71
## 4  5.46 10.88
## 5 12.92 11.83
## 6  8.12 13.17
```

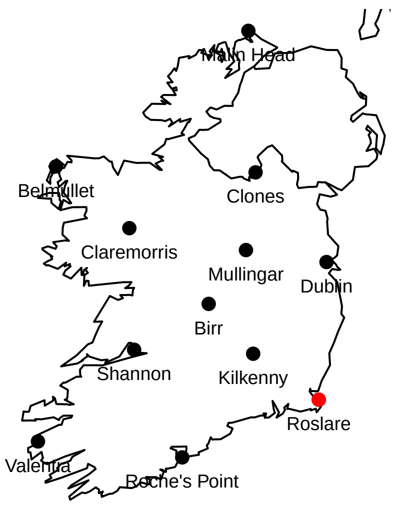
```r
summary(wind)
```

```
##       year          month             day             RPT
##  Min.   :61.0   Min.   : 1.000   Min.   : 1.00   Min.   : 0.67
##  1st Qu.:65.0   1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.: 8.12
##  Median :69.5   Median : 7.000   Median :16.00   Median :11.71
##  Mean   :69.5   Mean   : 6.523   Mean   :15.73   Mean   :12.36
##  3rd Qu.:74.0   3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:15.92
##  Max.   :78.0   Max.   :12.000   Max.   :31.00   Max.   :35.80
##       VAL             ROS             KIL              SHA
##  Min.   : 0.21   Min.   : 1.50   Min.   : 0.000   Min.   : 0.13
##  1st Qu.: 6.67   1st Qu.: 8.00   1st Qu.: 3.580   1st Qu.: 6.75
##  Median :10.17   Median :10.92   Median : 5.750   Median : 9.96
##  Mean   :10.65   Mean   :11.66   Mean   : 6.306   Mean   :10.46
##  3rd Qu.:14.04   3rd Qu.:14.67   3rd Qu.: 8.420   3rd Qu.:13.54
##  Max.   :33.37   Max.   :33.84   Max.   :28.460   Max.   :37.54
##       BIR              DUB              CLA              MUL
##  Min.   : 0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.000   1st Qu.: 6.000   1st Qu.: 5.090   1st Qu.: 5.370
##  Median : 6.830   Median : 9.210   Median : 8.080   Median : 8.170
##  Mean   : 7.092   Mean   : 9.797   Mean   : 8.494   Mean   : 8.496
##  3rd Qu.: 9.670   3rd Qu.:12.960   3rd Qu.:11.420   3rd Qu.:11.210
##  Max.   :26.160   Max.   :30.370   Max.   :31.080   Max.   :25.880
##       CLO              BEL             MAL
##  Min.   : 0.040   Min.   : 0.13   Min.   : 0.67
##  1st Qu.: 5.330   1st Qu.: 8.71   1st Qu.:10.71
##  Median : 8.290   Median :12.50   Median :15.00
##  Mean   : 8.707   Mean   :13.12   Mean   :15.60
##  3rd Qu.:11.630   3rd Qu.:16.88   3rd Qu.:19.83
##  Max.   :28.210   Max.   :42.38   Max.   :42.54
```

Extract and plot the data

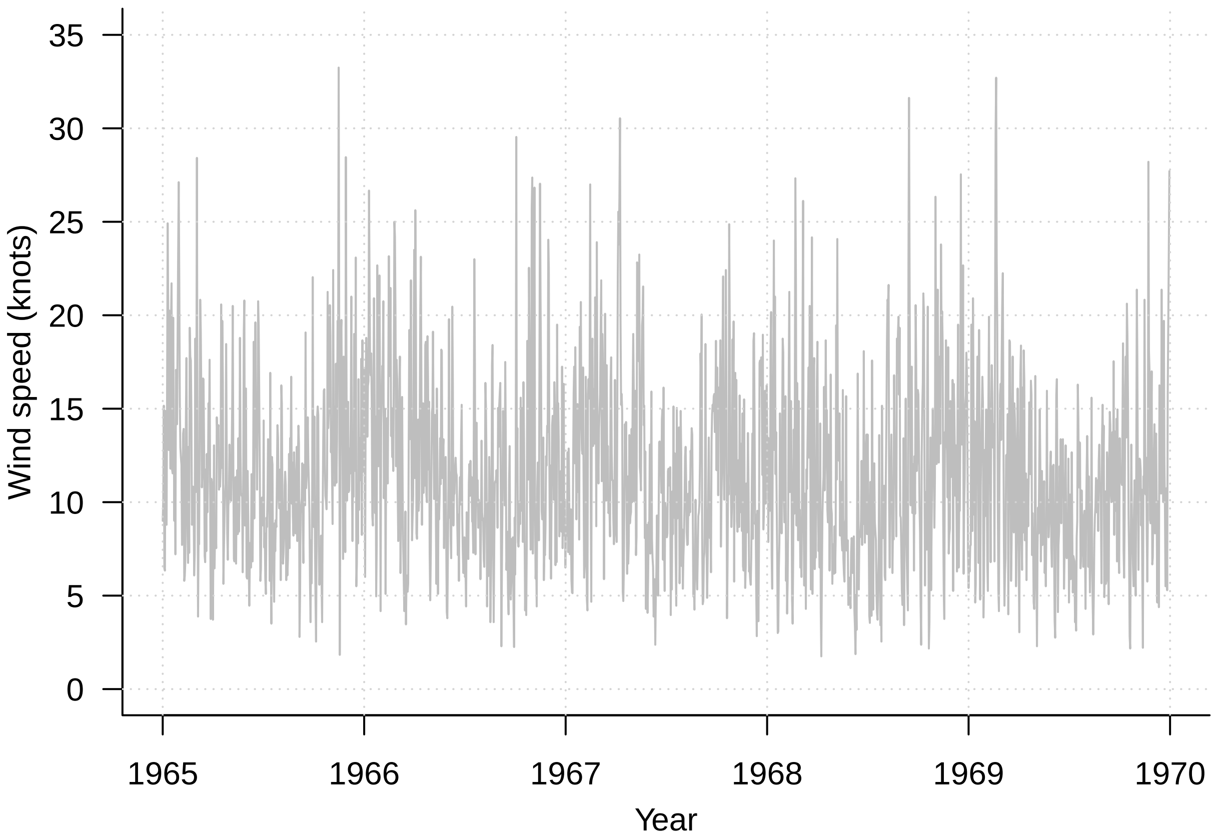
In this case study, we use data from the Rosslare station from 1965 to 1969. First, let's plot the spatial region with these station locations.

```
library(maps)
library(mapdata)
map("worldHires", xlim = c(-11, -5.4), ylim = c(51, 55.5))
library(sp) # char2dms
wind.loc$y <- as.numeric(char2dms(as.character(wind.loc[["Latitude"]])))
wind.loc$x <- as.numeric(char2dms(as.character(wind.loc[["Longitude"]])))
coordinates(wind.loc) = ~ x + y
text(coordinates(wind.loc), pos = 1, label = wind.loc$Station, cex = 0.6)
points(wind.loc, pch = 16); points(wind.loc[12,], pch = 16, col = "red")
```



Subset the data from the Rosslare station for the years 1965 to 1969.

```
id <- which(wind$year %in% 65:69)
rosslare <- wind$ROS[id]
## set up  the year variable
year <- seq(from = 1965, by = 1 / 365.25, length = length(rosslare))
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), mgp = c(2.2, 1, 0), las = 1)
plot(year, rosslare, type = "l", ylim = c(0, 35), lwd = 1, col = "gray",
     xlab = "Year", ylab = "Wind speed (knots)")
grid()
```



Deseasonalization: Harmonic Regression

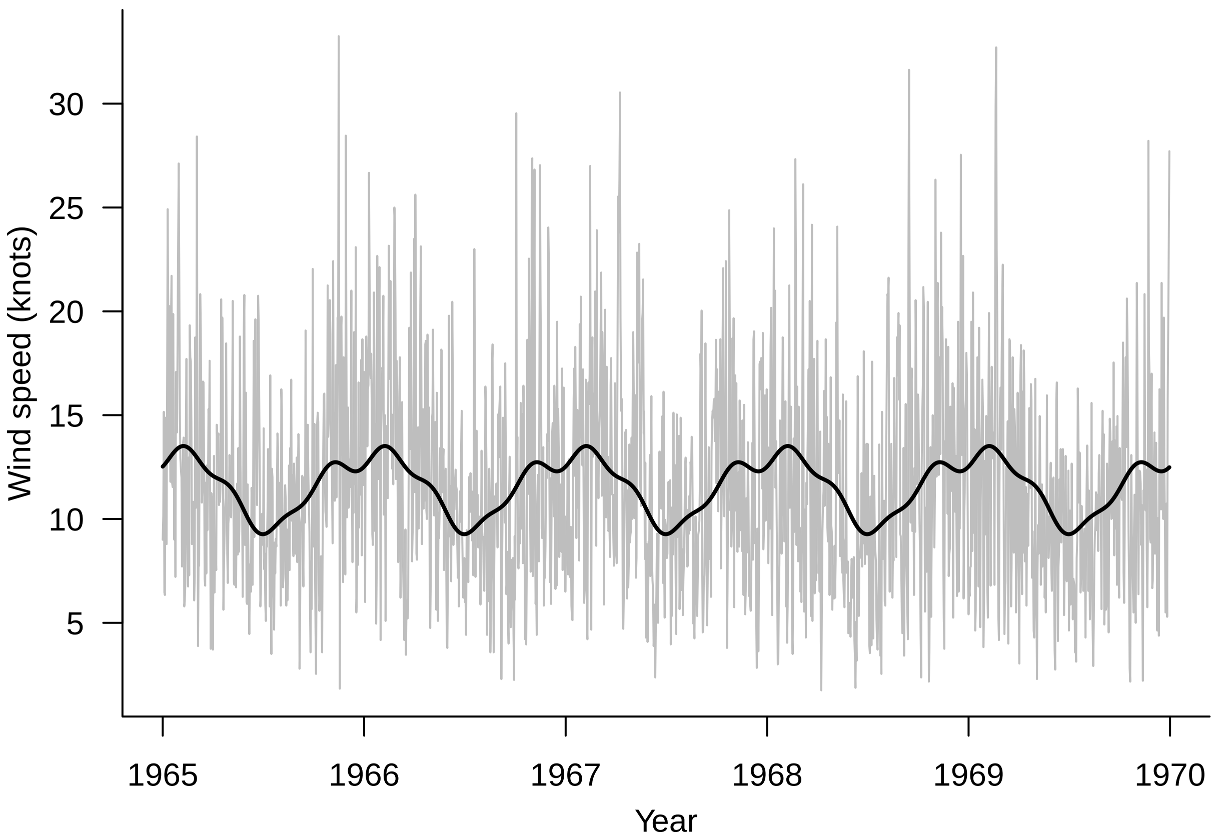
We use harmonic regression with 4 harmonics per year to model the seasonal components.

```r
## create harmonic terms
Harmonic <- function(year, K){
  t <- outer(2 * pi * year, 1:K)
  return(cbind(apply(t, 2, cos), apply(t, 2, sin)))
}
harmonics <- Harmonic(year, 4)
## fit a harmonic regression
harm.model <- lm(rosslare ~ harmonics)
summary(harm.model)
```

```
##
## Call:
## lm(formula = rosslare ~ harmonics)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8538  -3.3813  -0.4892   2.8395  20.8290
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.584141   0.112377 103.083  < 2e-16 ***
## harmonics1   1.687468   0.158936  10.617  < 2e-16 ***
```

```
## harmonics2  -0.435273   0.158936  -2.739  0.00623 **
## harmonics3  -0.060047   0.158936  -0.378  0.70562
## harmonics4  -0.251396   0.158936  -1.582  0.11388
## harmonics5   0.412363   0.158915   2.595  0.00954 **
## harmonics6   0.003874   0.158915   0.024  0.98055
## harmonics7   0.107245   0.158915   0.675  0.49985
## harmonics8   0.217870   0.158915   1.371  0.17055
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.802 on 1817 degrees of freedom
## Multiple R-squared:  0.06771,    Adjusted R-squared:  0.06361
## F-statistic:  16.5 on 8 and 1817 DF,  p-value: < 2.2e-16
```

```r
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), mgp = c(2.2, 1, 0), las = 1)
plot(year, rosslare, type = "l",
     xlab = "Year", ylab = "Wind speed (knots)", col = "grey")
lines(year, fitted(harm.model), lwd = 2)
```



ACF Plots: Original and Deseasonalized Series

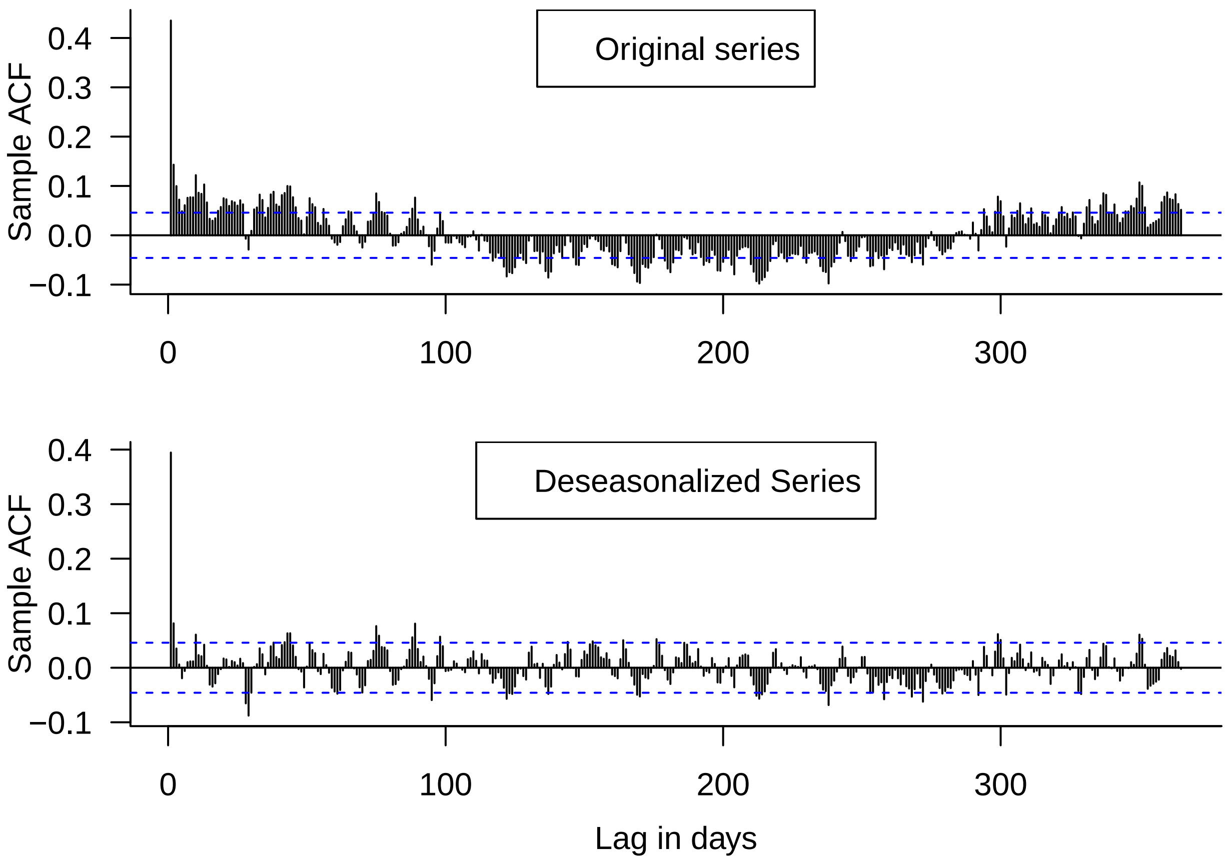
Let's plot the ACF and PACF plots to investigate the possible order for the ARMA model.

```r
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
```

```
##   method             from
##   as.zoo.data.frame zoo
```

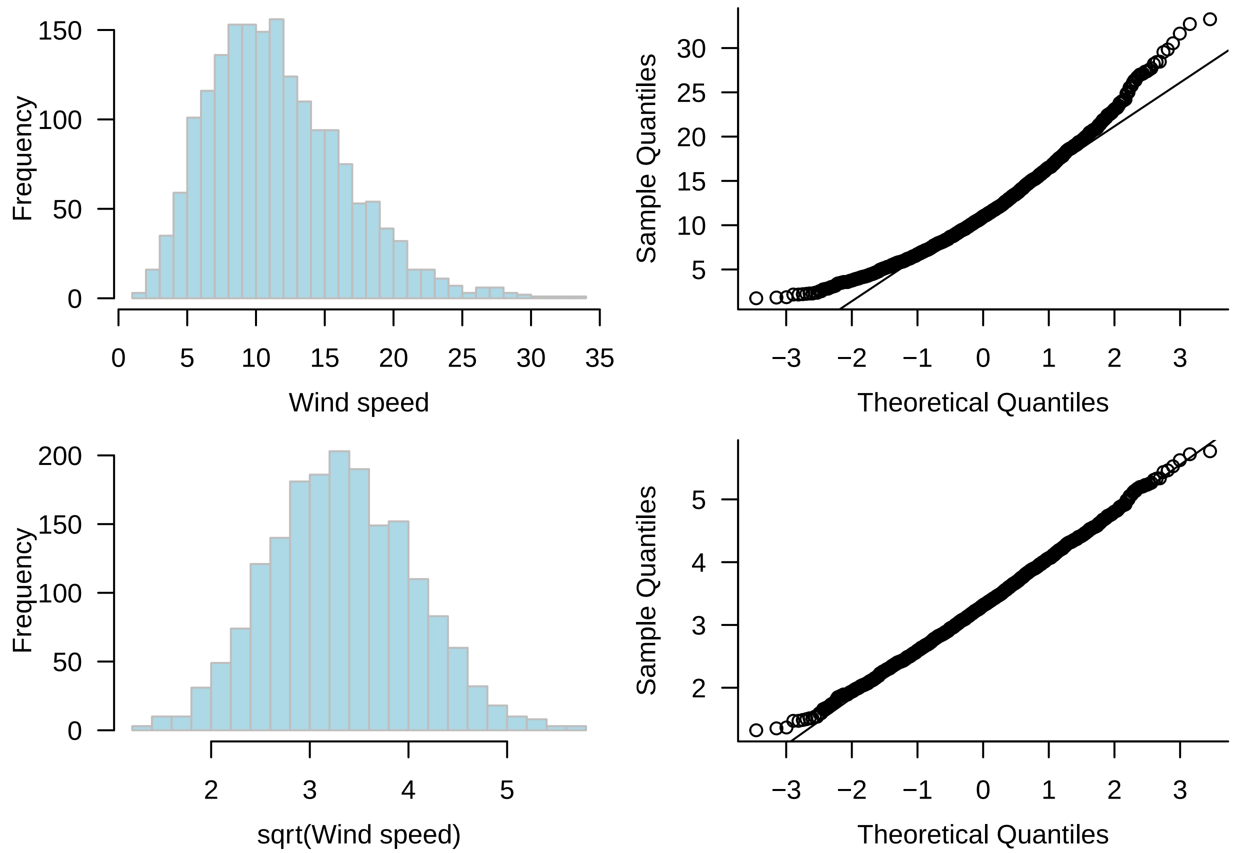
```r
par(bty = "L", mar = c(3.6, 3.6, 0.25, 0.5), mgp = c(2.4, 1, 0), las = 1,
    mfrow = c(2, 1))
Acf(rosslare, lag.max = 365, xlab = "", ylab = "Sample ACF", main = "")
legend("top", legend = "Original series")
Acf(resid(harm.model), lag.max = 365, xlab = "Lag in days",
    ylab = "Sample ACF", main = "")
legend("top", legend = "Deseasonalized Series")
```



Apply transformation to make wind speed more Gaussian like

Now look at a histogram of the values, along with the normal quantile-quantile plot.

```r
par(mfrow = c(2, 2), bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1,
    mgp = c(2.4, 1, 0))
hist(rosslare, 40, main = "", xlab = "Wind speed", col = "lightblue", border = "gray")
qqnorm(rosslare, main = "")
qqline(rosslare)
## Histogram/Q-Q plot of 1/2 root transformation
hist(sqrt(rosslare), 25, main = "", xlab = "sqrt(Wind speed)", col = "lightblue", border = "gray")
qqnorm(sqrt(rosslare), main=""); qqline(sqrt(rosslare))
```



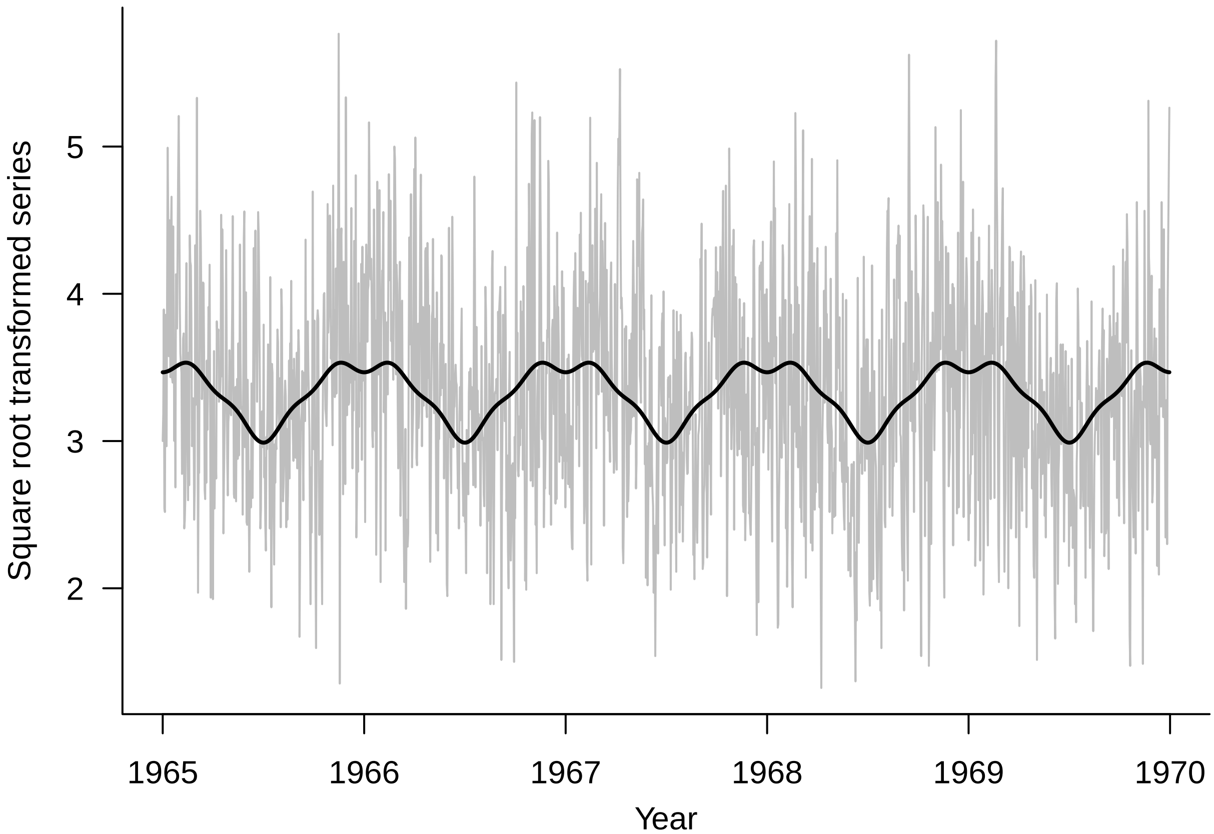
Now take square roots of the original data and deseasonalizeagain!

```
## now we start again from the beginning with a sqrt transformation
sqrt.rosslare <- sqrt(rosslare)
## refit the periodicity, without the intercept term
harm.model <- lm(sqrt.rosslare ~ harmonics[, 1:4] - 1)
summary(harm.model)
```

```
##
## Call:
## lm(formula = sqrt.rosslare ~ harmonics[, 1:4] - 1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  1.146  2.848  3.316  3.799  5.656
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## harmonics[, 1:4]1  0.2391111  0.1126203   2.123   0.0339 *
## harmonics[, 1:4]2 -0.0606520  0.1126203  -0.539   0.5903
## harmonics[, 1:4]3 -0.0001588  0.1126203  -0.001   0.9989
## harmonics[, 1:4]4 -0.0363877  0.1126202  -0.323   0.7467
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

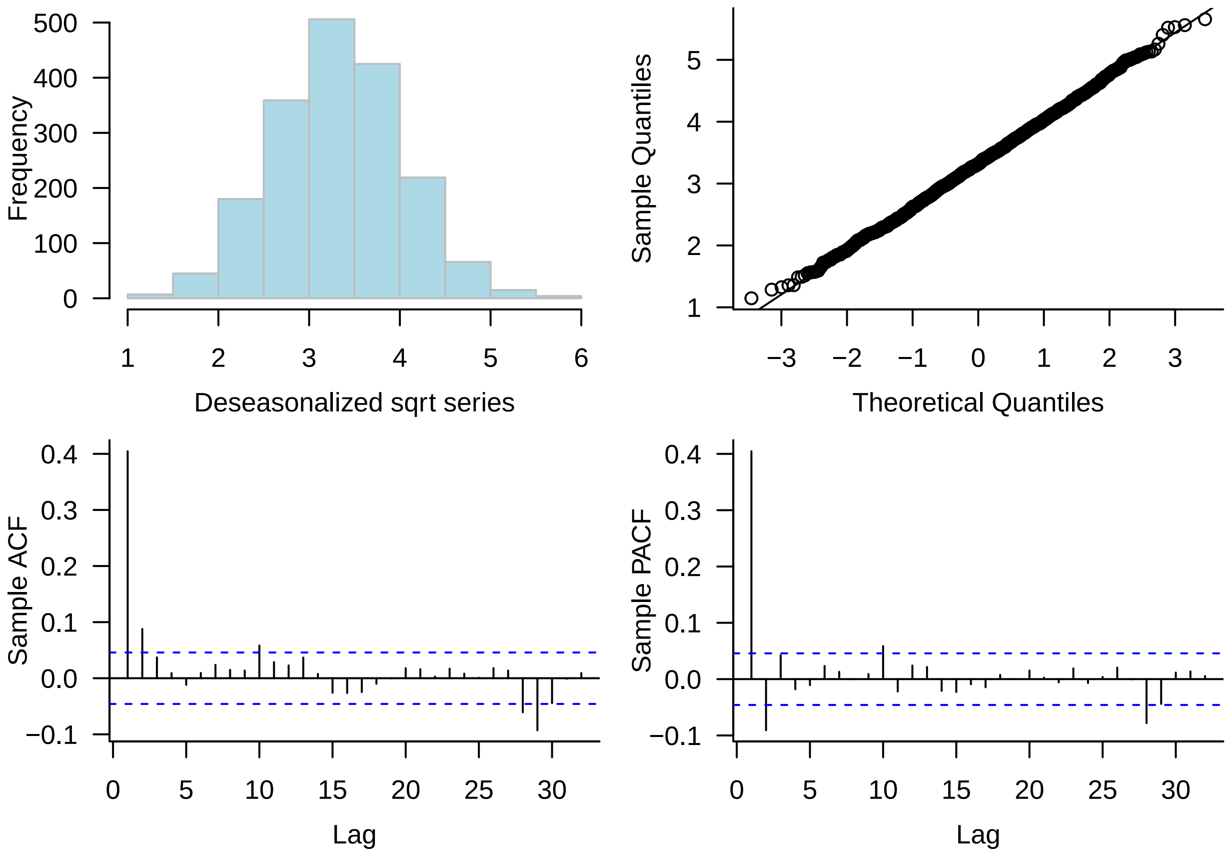
```
##
## Residual standard error: 3.403 on 1822 degrees of freedom
## Multiple R-squared:  0.002684,   Adjusted R-squared:  0.0004944
## F-statistic: 1.226 on 4 and 1822 DF,  p-value: 0.2978
```

```
## calculate the estimate of the deseasonalized series
sqrt.rosslare.ds <- resid(harm.model)
## Produce time series plots of the sqrt data
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.2, 1, 0))
plot(year, sqrt.rosslare, type = "l", col = "gray",
     xlab = "Year", ylab = "Square root transformed series")
lines(year, fitted(harm.model) + mean(sqrt.rosslare.ds), lwd = 2)
```



Checking Normality ACF/PACF

```
## And check the distribution
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.4, 1, 0),
    mfrow = c(2, 2))
hist(sqrt.rosslare.ds, main = "", xlab = "Deseasonalized sqrt series",
     col = "lightblue", border = "gray")
qqnorm(sqrt.rosslare.ds, main="")
qqline(sqrt.rosslare.ds)
## Now let's examine the sample ACF and PACF
Acf(sqrt.rosslare.ds, main = "", ylab = "Sample ACF")
Acf(sqrt.rosslare.ds, main = "", type = "partial", ylab = "Sample PACF")
```



Model identifcation, fitting, and selection

Let's first fit an AR(1) Fit an AR(1) model

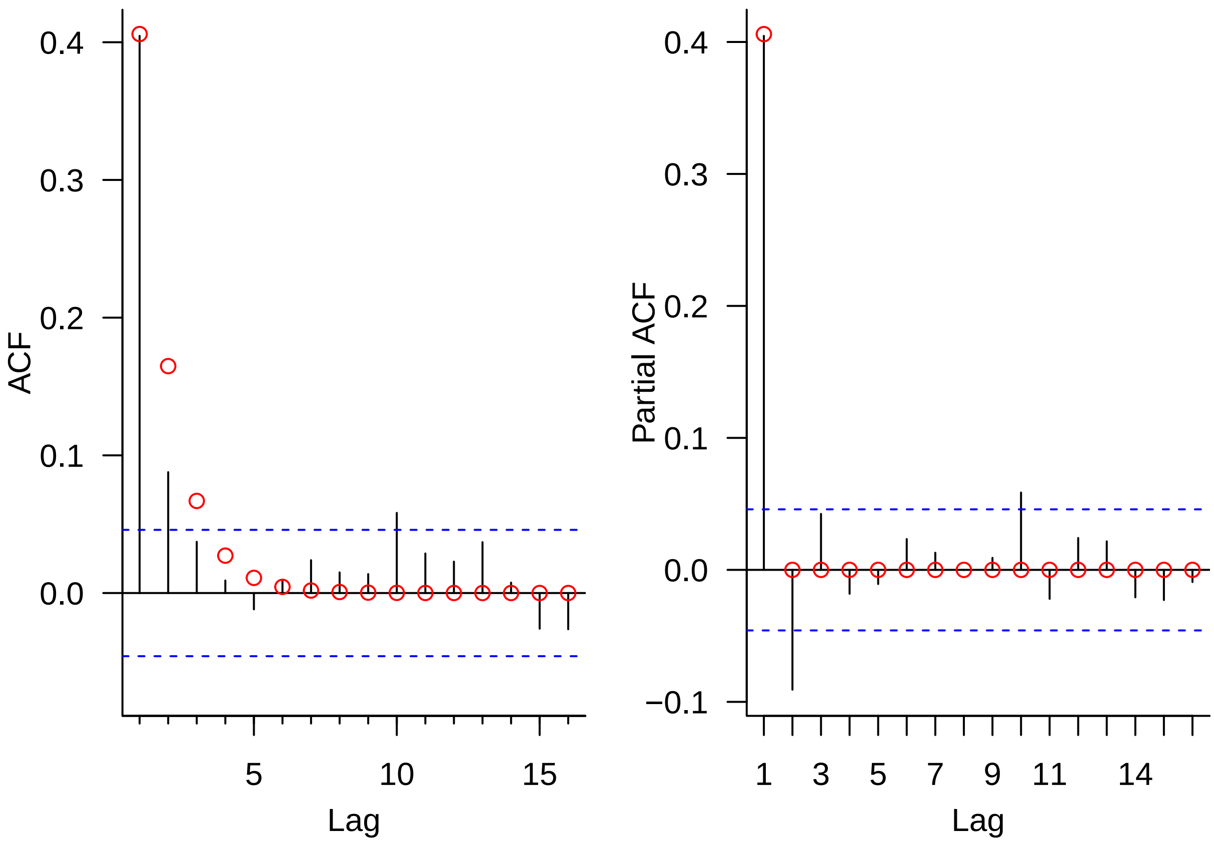
```
ar1.model <- arima(sqrt.rosslare.ds, order = c(1, 0, 0))
ar1.model
```

```
##
## Call:
## arima(x = sqrt.rosslare.ds, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.4060     3.3257
## s.e.  0.0214     0.0254
##
## sigma^2 estimated as 0.4148:  log likelihood = -1787.72,  aic = 3581.43
```

Sample and fitted ACF/PACF

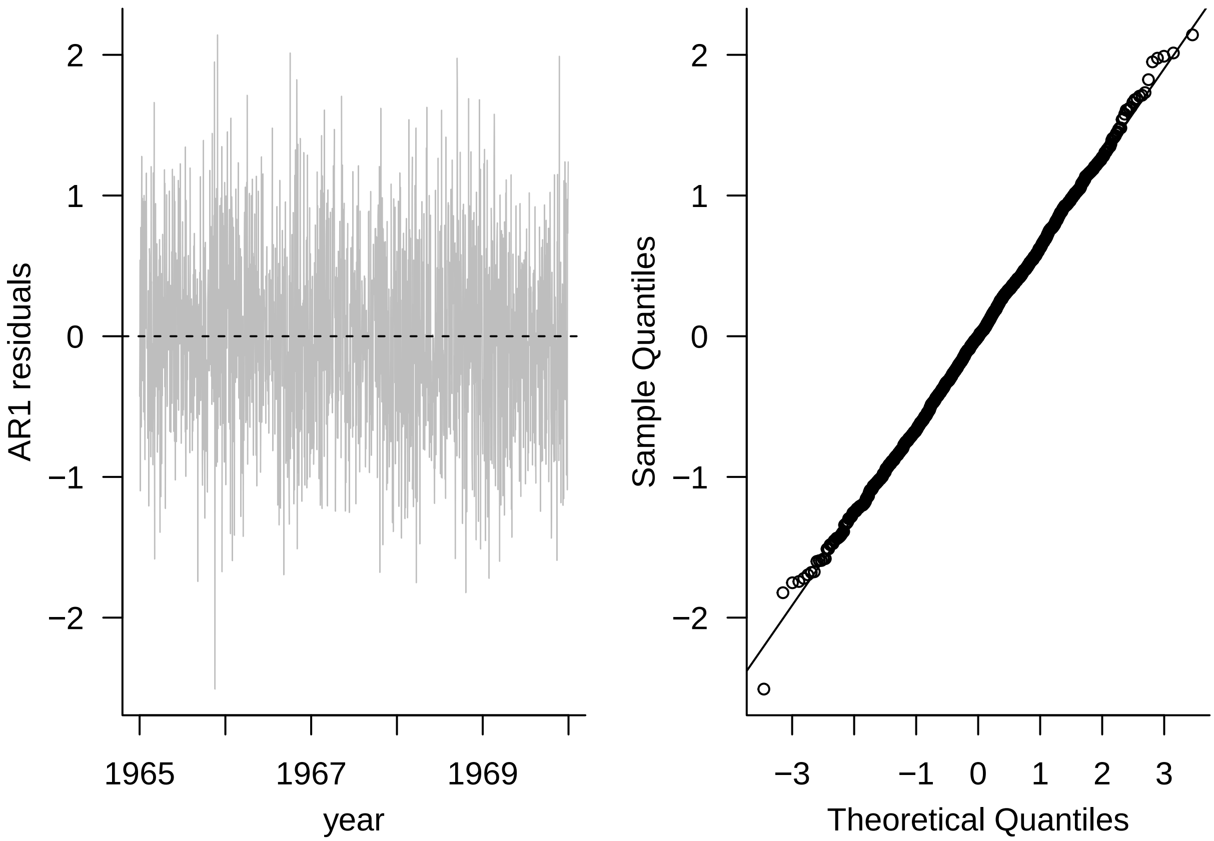
```
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.2, 1, 0),
    mfrow = c(1,2))
Acf(sqrt.rosslare.ds, main = "", lag.max = 16)
acf_true <- ARMAacf(ar = c(ar1.model$coef[1]), lag.max = 16)[-1]
points(1:16, acf_true, col = "red")
```

```
Acf(sqrt.rosslare.ds, main = "", lag.max = 16, type = "partial")
pacf_true <- ARMAacf(ar = c(ar1.model$coef[1]), lag.max = 16, pacf = T)
points(1:16, pacf_true, col = "red")
```



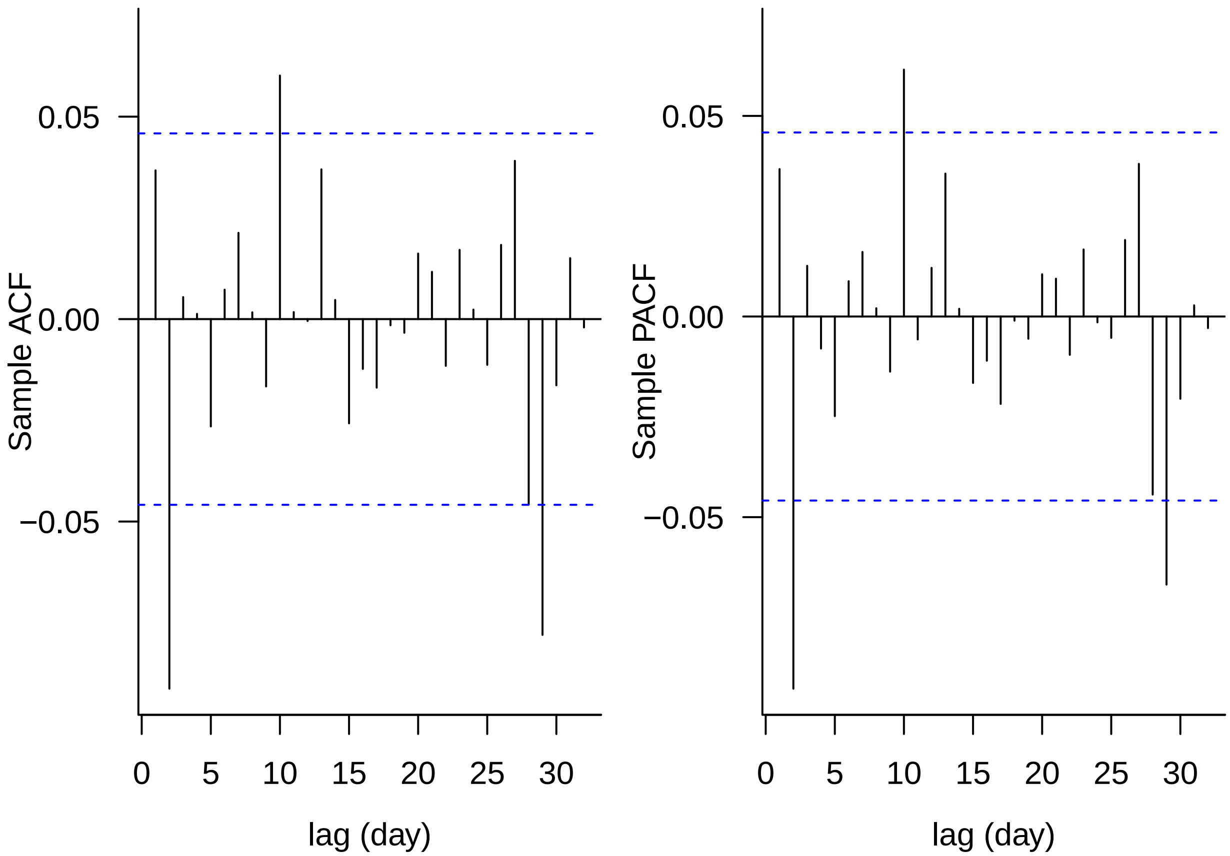
Extract residuals

```
ar1.resids <- resid(ar1.model)
## time series plot of the residuals
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.2, 1, 0),
    mfrow = c(1, 2))
plot(year, ar1.resids, type = "l", xlab = "year", ylab = "AR1 residuals",
     lwd = 0.6, col = "gray")
abline(h = 0, lty = 2)
## Normal Q-Q plot for the residuals
qqnorm(ar1.resids, main = "", cex = 0.75); qqline(ar1.resids)
```



Sample ACF and PACF of the residuals

```
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.6, 1, 0), mfrow = c(1, 2))
Acf(ar1.resids, ylab = "Sample ACF", xlab = "lag (day)", main = "")
Acf(ar1.resids, ylab = "Sample PACF",type = "partial",  xlab = "lag (day)")
```




```r
## Carry out the Ljung-Box test
Box.test(ar1.resids, lag = 32, fitdf = 1, type = "Ljung-Box")
```

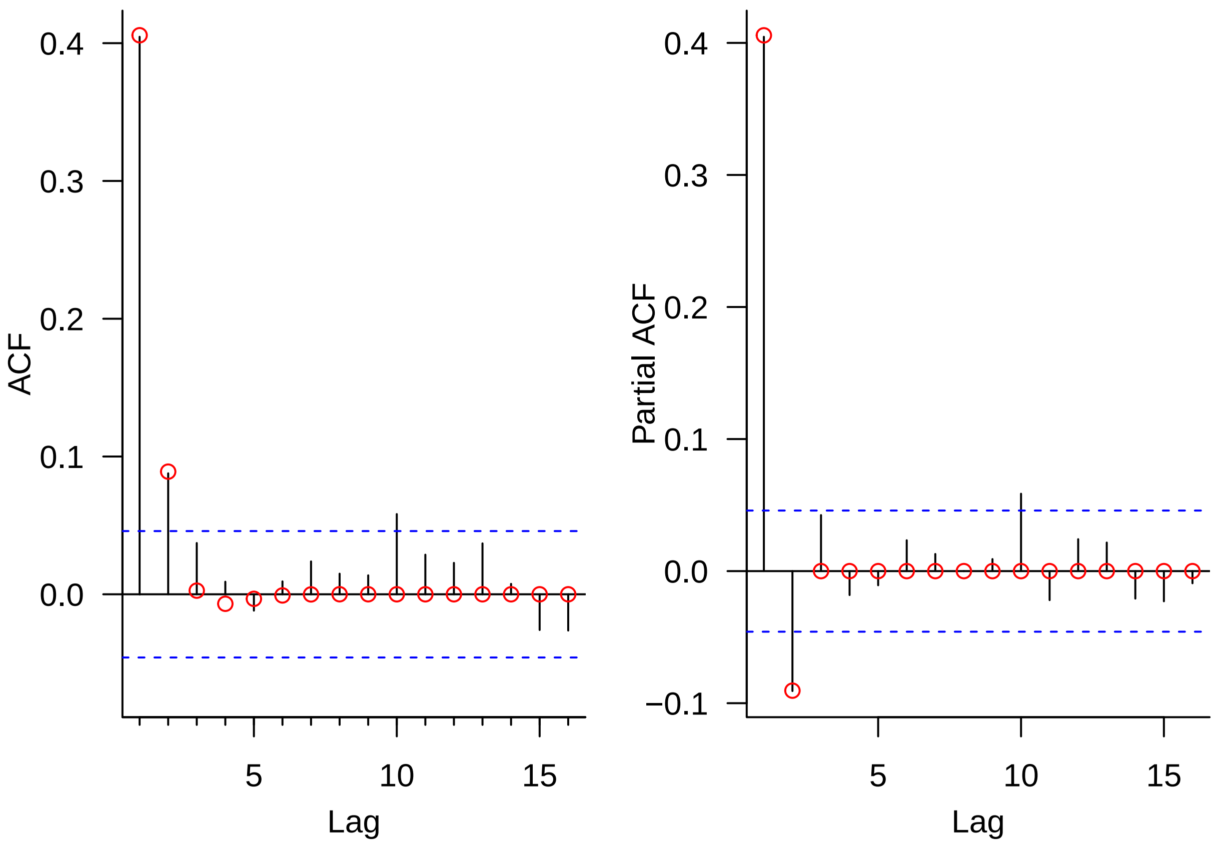
```
##
##  Box-Ljung test
##
## data:  ar1.resids
## X-squared = 53.142, df = 31, p-value = 0.00794
```

```r
(ar2.model <- arima(sqrt.rosslare.ds, order = c(2, 0, 0)))
```

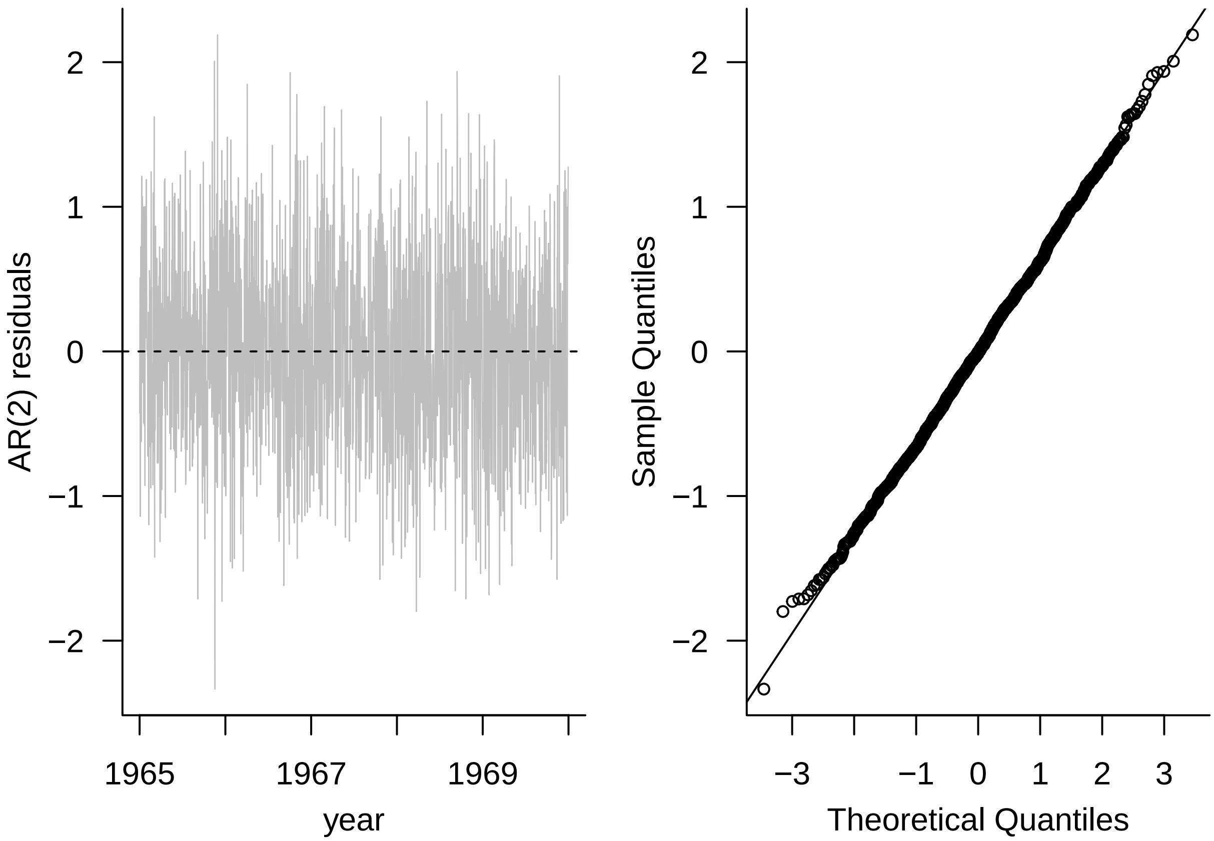
Fit an AR(2) model

```
##
## Call:
## arima(x = sqrt.rosslare.ds, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       0.4425  -0.0905     3.3254
## s.e.  0.0233   0.0233     0.0232
##
## sigma^2 estimated as 0.4114:  log likelihood = -1780.23,  aic = 3568.46
```

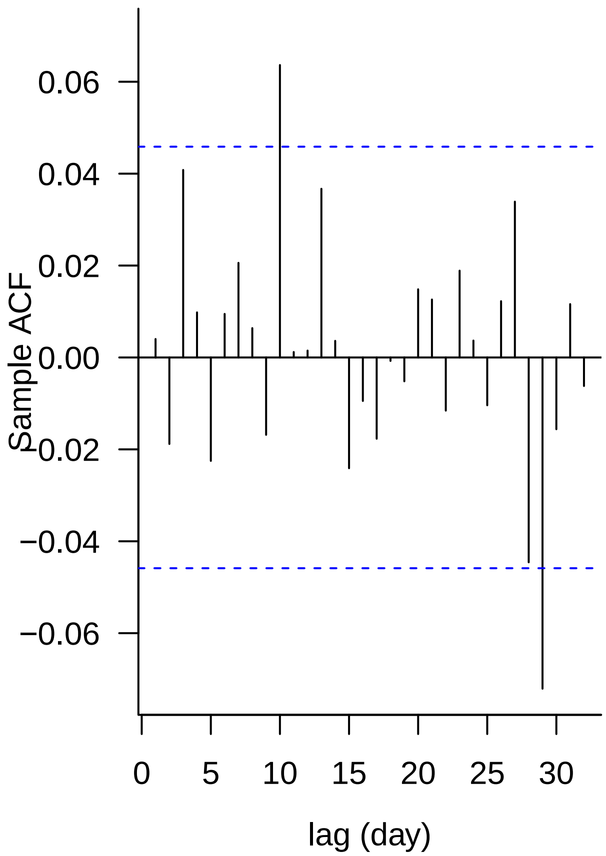
```r
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.2, 1, 0),
    mfrow = c(1, 2))
Acf(sqrt.rosslare.ds, main = "", lag.max = 16)
acf_true <- ARMAacf(ar = c(ar2.model$coef[1:2]), lag.max = 16)[-1]
points(1:16, acf_true, col = "red")
acf(sqrt.rosslare.ds, main = "", lag.max = 16, type = "partial")
pacf_true <- ARMAacf(ar = c(ar2.model$coef[1:2]), lag.max = 16, pacf = T)
points(1:16, pacf_true, col = "red")
```

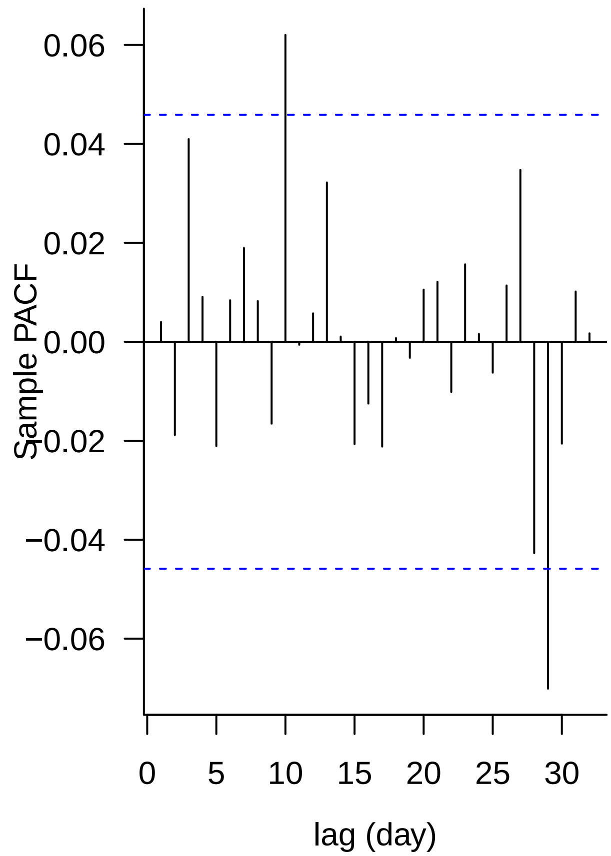



```r
## extract the residuals
ar2.resids <- resid(ar2.model)

## time series plot of the residuals
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.2, 1, 0), mfrow = c(1, 2))
plot(year, ar2.resids, type = "l", xlab = "year",
     ylab = "AR(2) residuals", lwd = 0.6, col = "gray")
abline(h = 0, lty = 2)
## Normal Q-Q plot for the residuals
qqnorm(ar2.resids, main = "", cex = 0.75); qqline(ar2.resids)
```




```
## Sample ACF and PACF of the residuals
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.6, 1, 0), mfrow = c(1, 2))
Acf(ar2.resids, ylab = "Sample ACF", xlab = "lag (day)", main = "")
pacf(ar2.resids, ylab = "Sample PACF", xlab = "lag (day)")
```






```r
## Carry out the Ljung-Box test
Box.test(ar2.resids, lag = 32, fitdf = 2, type = "Ljung-Box")
```

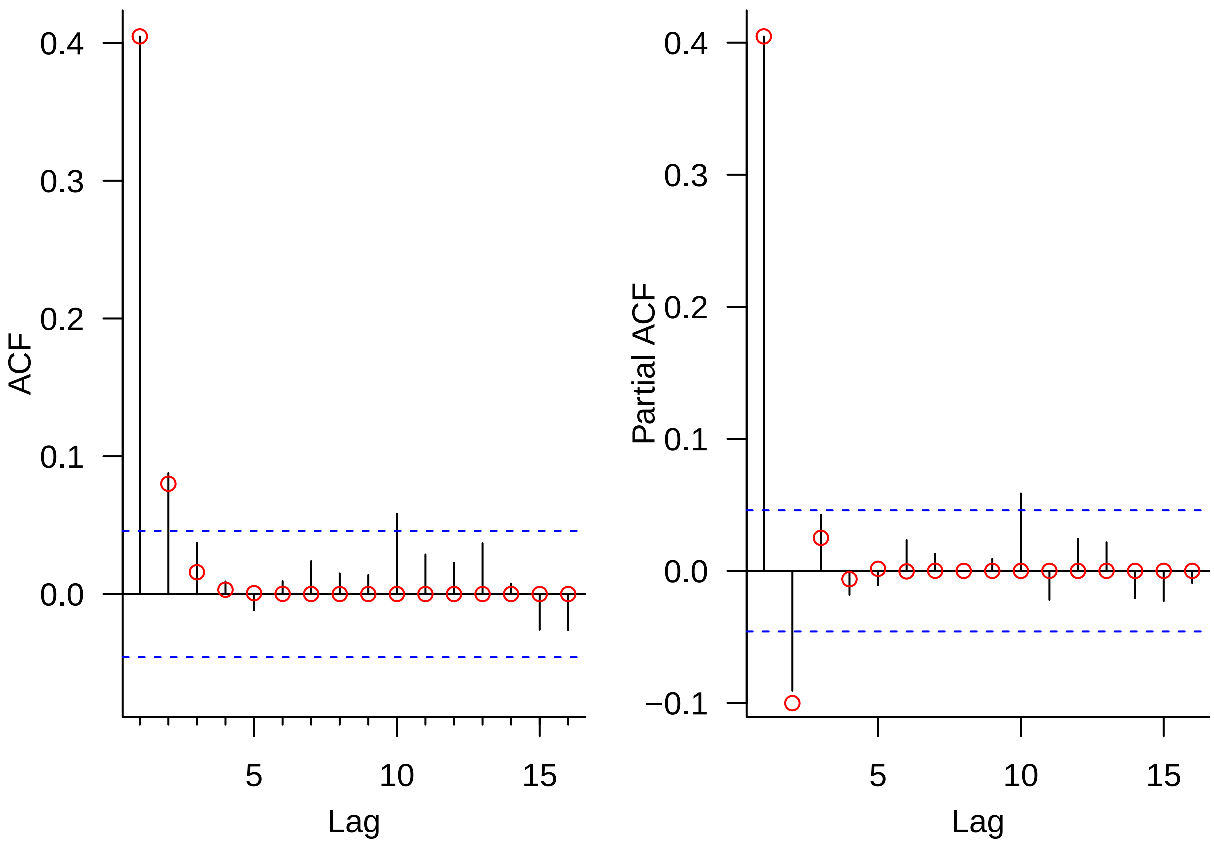
```
##
##  Box-Ljung test
##
## data:  ar2.resids
## X-squared = 36.548, df = 30, p-value = 0.1907
```

```r
(arma11.model <- arima(sqrt.rosslare.ds, order = c(1, 0, 1)))
```

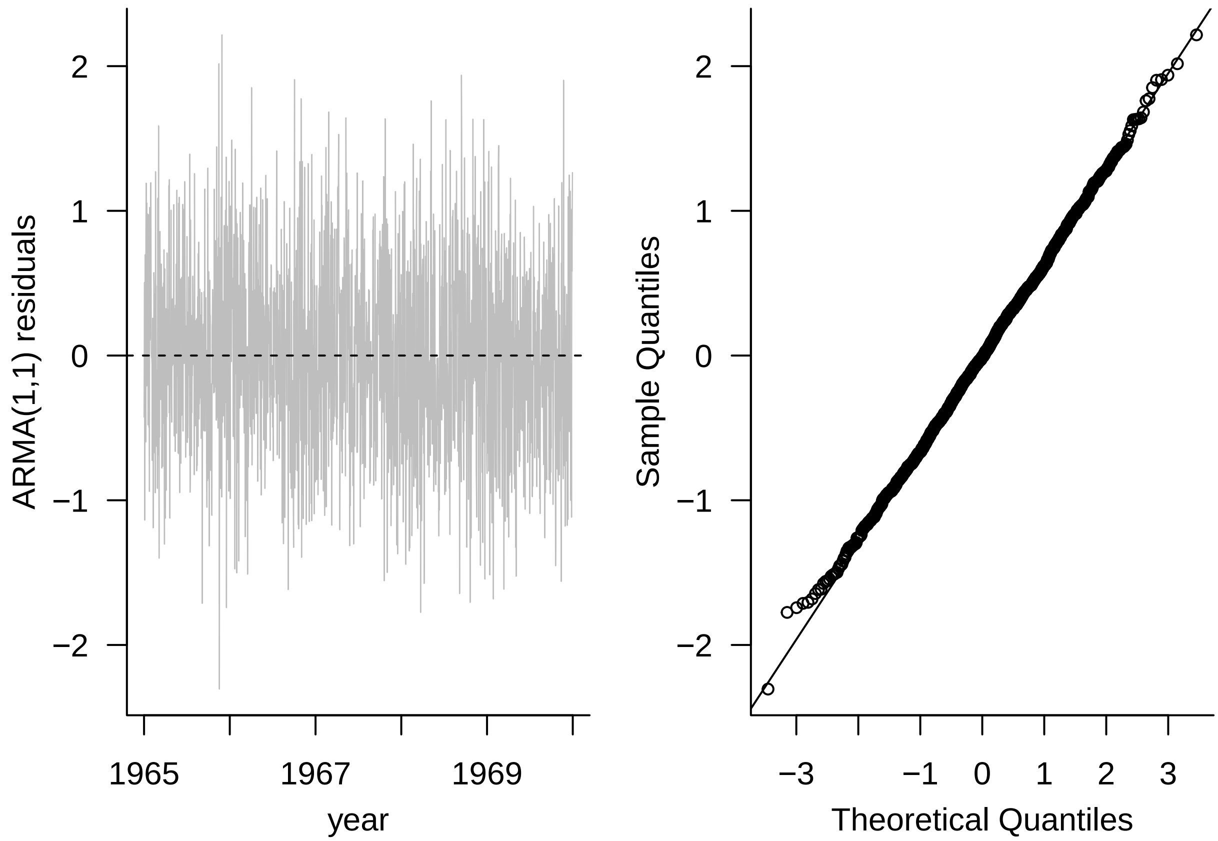
Fit an ARMA(1,1) model

```
##
## Call:
## arima(x = sqrt.rosslare.ds, order = c(1, 0, 1))
##
## Coefficients:
##          ar1     ma1  intercept
##       0.1978  0.2502     3.3254
## s.e.  0.0556  0.0553     0.0234
##
## sigma^2 estimated as 0.4108:  log likelihood = -1778.82,  aic = 3565.64
```

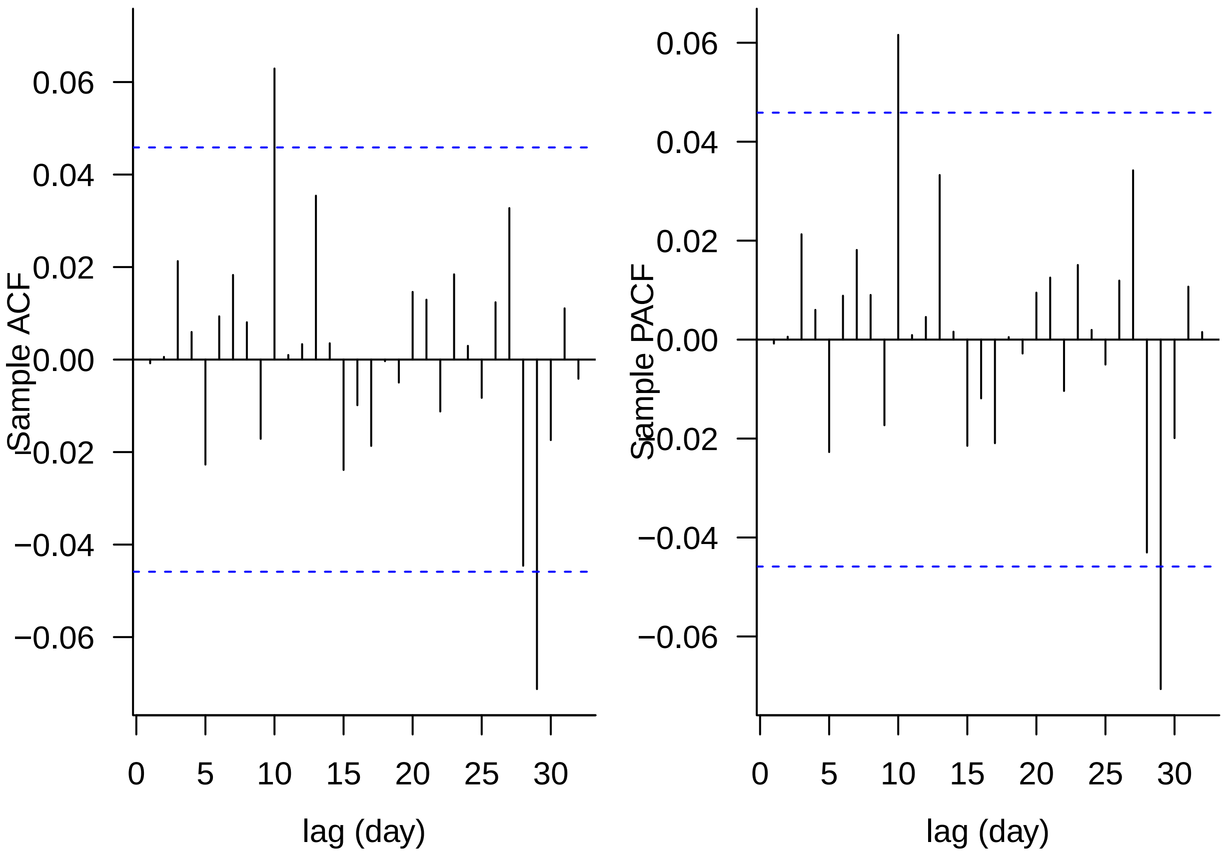
```r
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1,
    mgp = c(2.2, 1, 0), mfrow = c(1, 2))
Acf(sqrt.rosslare.ds, main = "", lag.max = 16)
acf_true <- ARMAacf(ar = arma11.model$coef[1], ma = arma11.model$coef[2], lag.max = 16)[-1]
points(1:16, acf_true, col = "red")
acf(sqrt.rosslare.ds, main = "", lag.max = 16, type = "partial")
pacf_true <- ARMAacf(ar = arma11.model$coef[1], ma = arma11.model$coef[2], lag.max = 16, pacf = T)
points(1:16, pacf_true, col = "red")
```




```r
## extract the residuals
arma11.resids <- resid(arma11.model)
## time series plot of the residuals
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.2, 1, 0), mfrow = c(1, 2))
plot(year, arma11.resids, type = "l", xlab = "year",
     ylab = "ARMA(1,1) residuals", lwd = 0.6, col = "gray")
abline(h = 0, lty = 2)
## Normal Q-Q plot for the residuals
qqnorm(arma11.resids, main = "", cex = 0.75); qqline(arma11.resids)
```




```r
## Sample ACF and PACF of the residuals
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.5, 1, 0),
    mfrow = c(1, 2))
Acf(arma11.resids, ylab = "Sample ACF", xlab = "lag (day)", main = "")
pacf(arma11.resids, ylab = "Sample PACF", xlab = "lag (day)")
```




```
## Carry out the Ljung-Box test
Box.test(arma11.resids, lag = 32, fitdf = 2, type = "Ljung-Box")
```

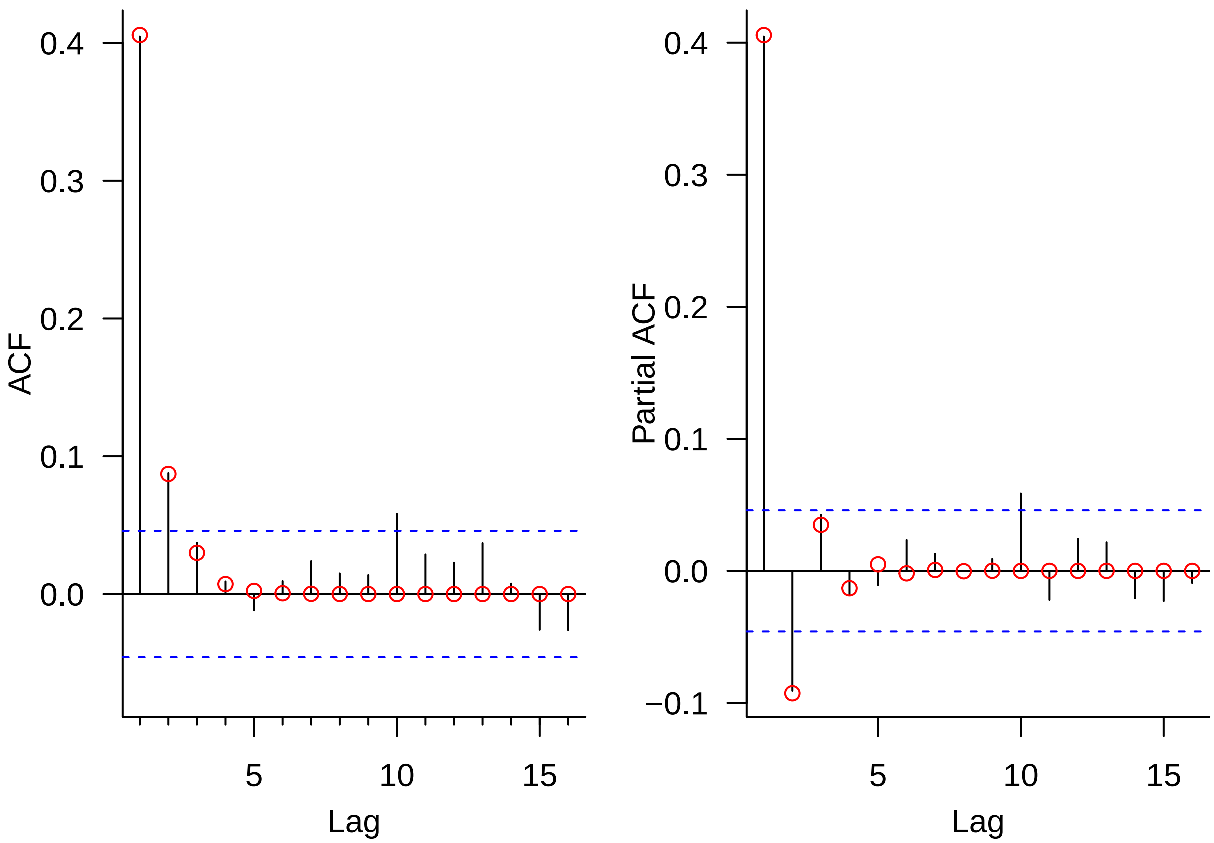
```
##
##  Box-Ljung test
##
## data:  arma11.resids
## X-squared = 32.757, df = 30, p-value = 0.3332
```

```
(arma21.model <- arima(sqrt.rosslare.ds, order = c(2, 0, 1)))
```

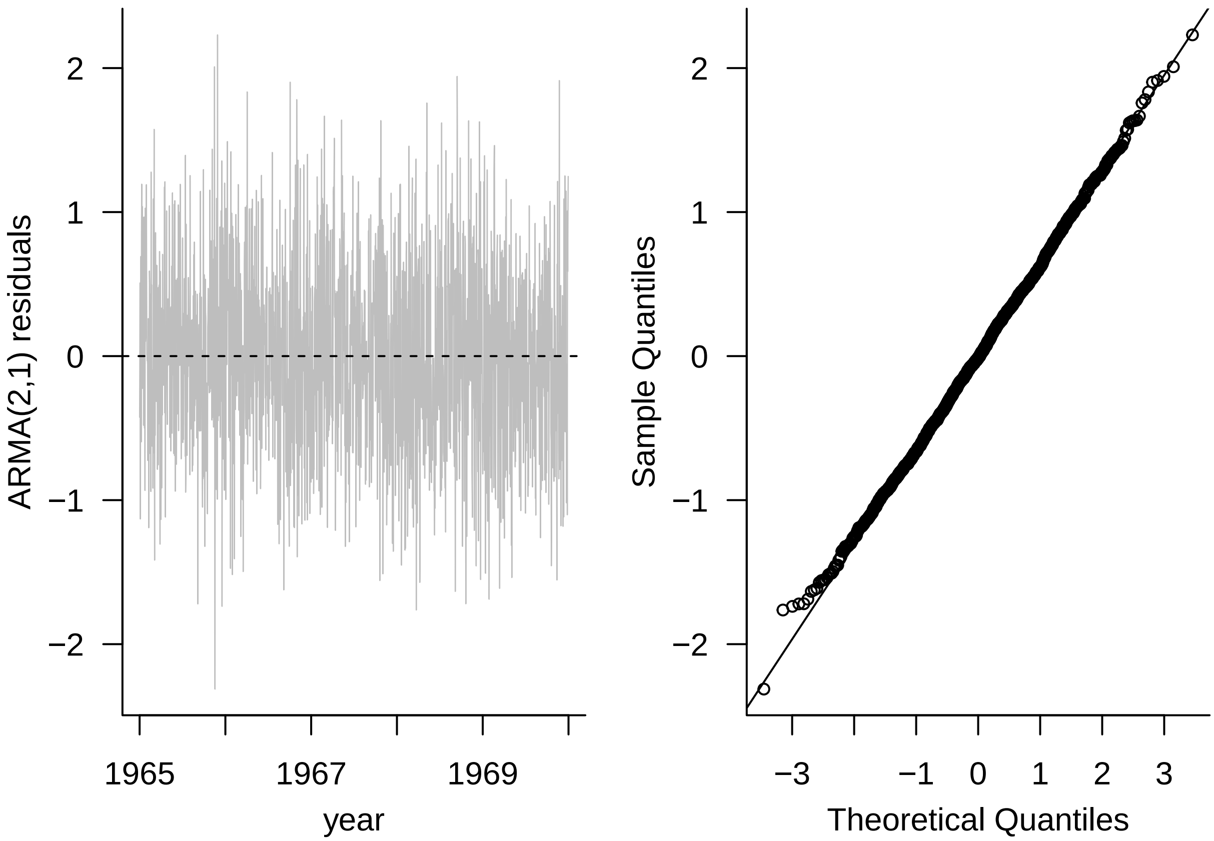
Fit an ARMA(2,1) model

```
##
## Call:
## arima(x = sqrt.rosslare.ds, order = c(2, 0, 1))
##
## Coefficients:
##          ar1     ar2     ma1  intercept
##       0.0703  0.0587  0.3768     3.3253
## s.e.  0.1691  0.0772  0.1663     0.0237
##
## sigma^2 estimated as 0.4107:  log likelihood = -1778.56,  aic = 3567.11
```

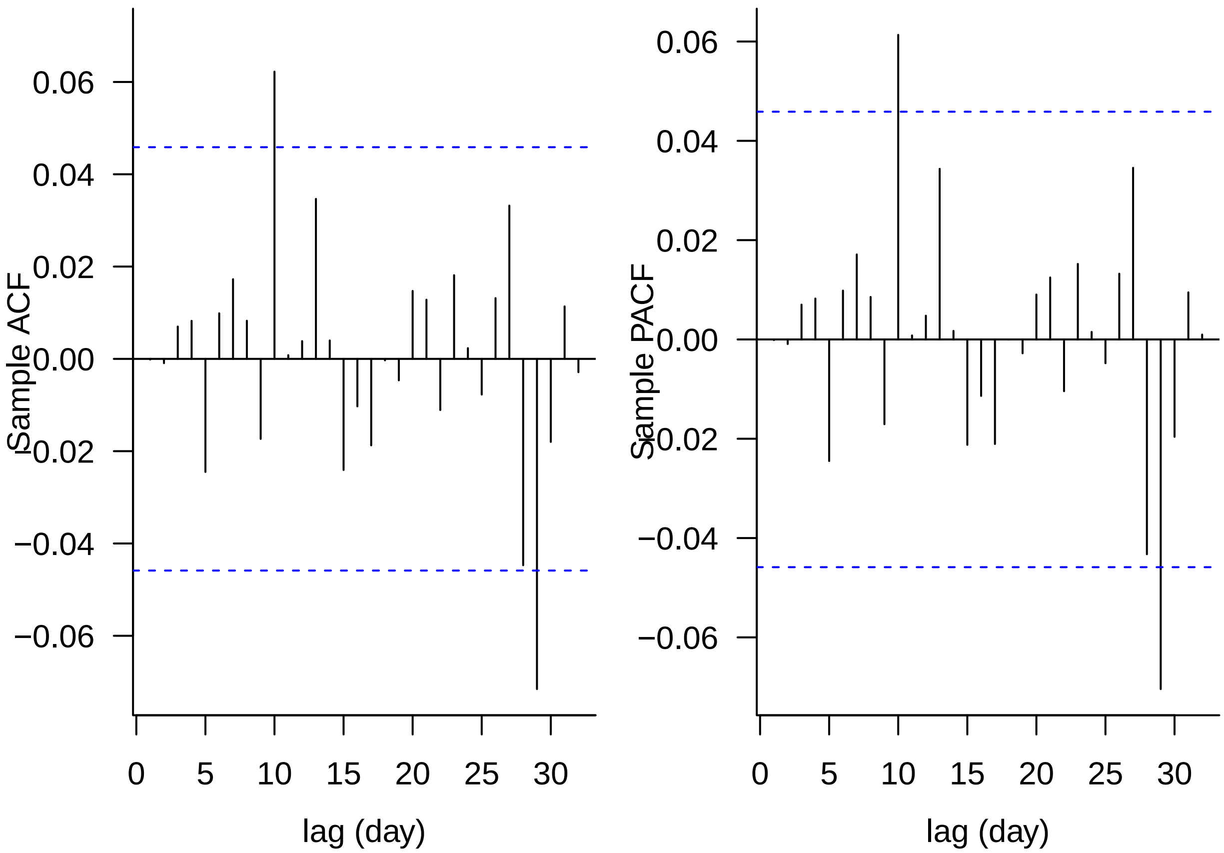
```r
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1,
    mgp = c(2.2, 1, 0), mfrow = c(1, 2))
Acf(sqrt.rosslare.ds, main = "", lag.max = 16)
acf_true <- ARMAacf(ar = c(arma21.model$coef[1:2]), ma = arma21.model$coef[3], lag.max = 16)[-1]
points(1:16, acf_true, col = "red")
acf(sqrt.rosslare.ds, main = "", lag.max = 16, type = "partial")
pacf_true <- ARMAacf(ar = c(arma21.model$coef[1:2]), ma = arma21.model$coef[3], lag.max = 16, pacf = T)
points(1:16, pacf_true, col = "red")
```




```r
## extract the residuals
arma21.resids <- resid(arma21.model)
## time series plot of the residuals
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.2, 1, 0), mfrow = c(1, 2))
plot(year, arma21.resids, type = "l", xlab = "year",
     ylab = "ARMA(2,1) residuals", lwd = 0.6, col = "gray")
abline(h = 0, lty = 2)
## Normal Q-Q plot for the residuals
qqnorm(arma21.resids, main = "", cex = 0.75); qqline(arma21.resids)
```




```r
## Sample ACF and PACF of the residuals
par(bty = "L", mar = c(3.6, 3.6, 0.5, 0.6), las = 1, mgp = c(2.5, 1, 0), mfrow = c(1, 2))
Acf(arma21.resids, ylab = "Sample ACF", xlab = "lag (day)", main = "")
pacf(arma21.resids, ylab = "Sample PACF", xlab = "lag (day)")
```




```r
## Carry out the Ljung-Box test
Box.test(arma21.resids, lag = 32, fitdf = 3, type = "Ljung-Box")
```

```
##
##  Box-Ljung test
##
## data:  arma21.resids
## X-squared = 32.171, df = 29, p-value = 0.3124
```

Use AIC to conduct model selection

```r
AIC.to.AICC <- function (aic, n, npars) {
  aic - 2 * npars * ( 1 - n / (n - 1 - npars))
}
# calculate the length of the time series
n <- length(sqrt.rosslare.ds)

# Here are the AIC values
ar1.model$aic
```

```
## [1] 3581.432
```

```r
ar2.model$aic
```

```
## [1] 3568.46
```

```r
arma11.model$aic
```

```
## [1] 3565.642
```

```r
arma21.model$aic
```

```
## [1] 3567.112
```

```r
# convert the AIC values to AICC values.
AIC.to.AICC(ar1.model$aic, n, 2)
```

```
## [1] 3581.438
```

```r
AIC.to.AICC(ar2.model$aic, n, 3)
```

```
## [1] 3568.473
```

```r
AIC.to.AICC(arma11.model$aic, n, 3)
```

```
## [1] 3565.655
```

```r
AIC.to.AICC(arma21.model$aic, n, 4)
```

```
## [1] 3567.134
```

Based on the AIC (and AICc as well), we choose the ARMA(1,1) model.

Forecasting

```r
## How many days will we predict into the future?
h <- 7
## Predict 'h' days into the future using the ARMA(1,1) model.
sqrt.rosslare.forecast <- predict(arma11.model, h)
sqrt.rosslare.forecast$pred <- ts(sqrt.rosslare.forecast$pred,
                                  start = c(1970, 1),
                                  frequency = 365)
sqrt.rosslare.forecast$se <- ts(sqrt.rosslare.forecast$se,
                                  start = c(1970, 1),
                                  frequency = 365)
round(sqrt.rosslare.forecast$pred, 3)
```

```
## Time Series:
## Start = c(1970, 1)
## End = c(1970, 7)
## Frequency = 365
## [1] 3.997 3.458 3.352 3.331 3.326 3.326 3.325
```

```r
round(sqrt.rosslare.forecast$se, 3)
```

```
## Time Series:
## Start = c(1970, 1)
## End = c(1970, 7)
## Frequency = 365
## [1] 0.641 0.702 0.705 0.705 0.705 0.705 0.705
```

```r
## define the forecast variable
forecast <- sqrt.rosslare.forecast$pred
## The plus or minus value is the z critical value
## times the standard error for the forecast
me <- qnorm(0.975) * sqrt.rosslare.forecast$se
lower <- forecast - me
upper <- forecast + me
## Define the prediction time
fyear <- 1970 + (0:(h - 1)) / 365.25
```

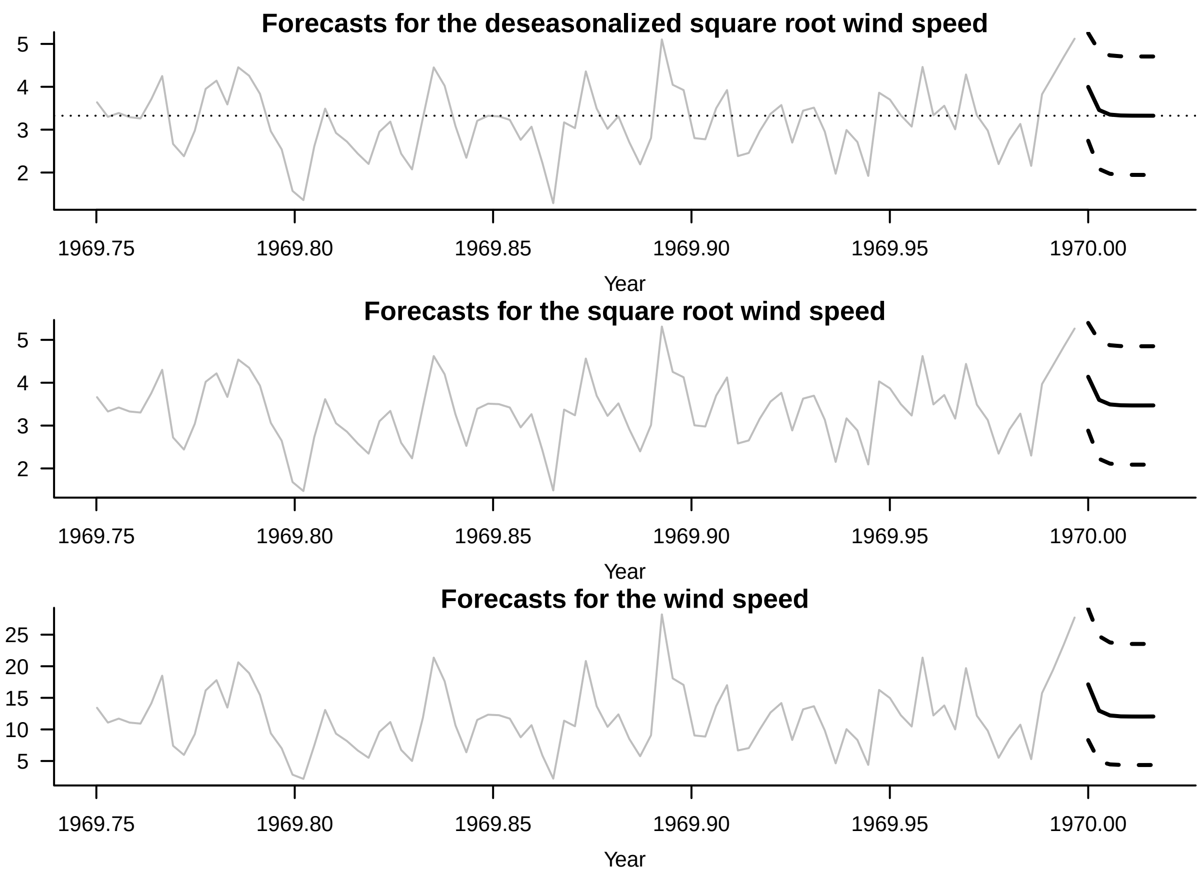
Visualizing the Forecasts

```r
par(bty = "L", mar = c(3.6, 3.6, 0.75, 0.6), las = 1, mgp = c(2.4, 1, 0),
    mfrow = c(3, 1))
## Show the data for 1969 onwards
plot(year[year > 1969.75], sqrt.rosslare.ds[year > 1969.75], type = "l",
     xlim = c(1969.75, max(fyear)), col = "grey", xlab = "Year", ylab = "")
## Add the BLUP, along with the prediction limits
lines(fyear, forecast, lwd = 2)
lines(fyear, lower, lty = 2, lwd = 2)
lines(fyear, upper, lty = 2, lwd = 2)
## add a horizontal line at the mean
abline(h = mean(sqrt.rosslare.ds), lty = 3)
title("Forecasts for the deseasonalized square root wind speed")

## now add the seasonality estimate for the first 10 days in a year.
adj.forecast <- fitted(harm.model)[1:h] + sqrt.rosslare.forecast$pred
round(adj.forecast, 3)
```

```
## Time Series:
## Start = c(1970, 1)
## End = c(1970, 7)
## Frequency = 365
##     1     2     3     4     5     6     7
## 4.139 3.600 3.494 3.473 3.470 3.470 3.470
```

```r
## adjust the lower and upper values of the interval
lower <- adj.forecast - me
upper <- adj.forecast + me

## Show the data for 1969 onwards
plot(year[year > 1969.75], sqrt.rosslare[year > 1969.75], type = "l",
```

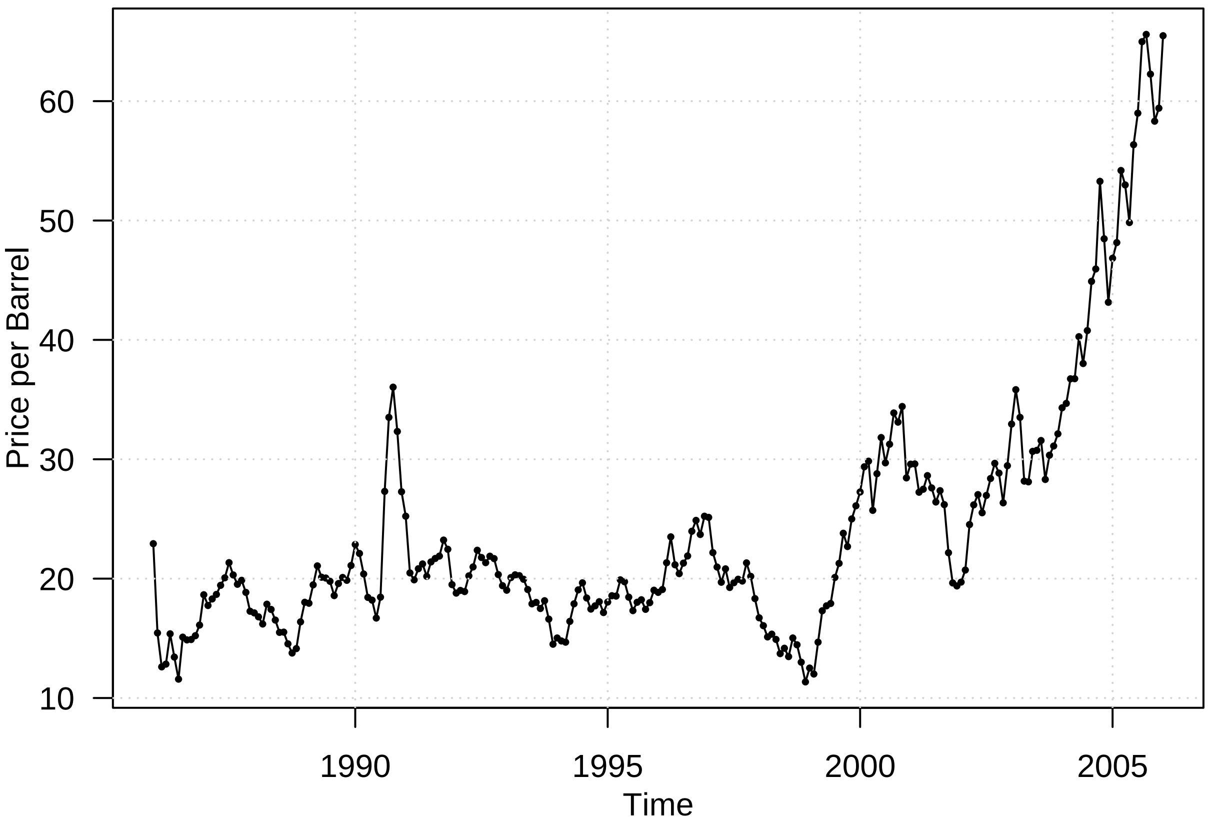
```
     xlim = c(1969.75, max(fyear)), col = "grey", xlab = "Year", ylab = "")
title("Forecasts for the square root wind speed")

## Add the BLUP, along with the prediction limits
lines(fyear, adj.forecast, lwd = 2)
lines(fyear, lower, lty = 2, lwd = 2)
lines(fyear, upper, lty =2 , lwd = 2)

## We square everything (forecast, lower limit, and upper limit)
## to get the forecast on the original wind speed (knots) scale.
## Show the data for 1969 onwards
plot(year[year > 1969.75], rosslare[year > 1969.75], type = "l",
     xlim = c(1969.75, max(fyear)), col = "grey", xlab = "Year", ylab = "")
title("Forecasts for the wind speed")

## Add the BLUP, along with the prediction limits
lines(fyear, adj.forecast^2, lwd = 2)
lines(fyear, lower^2, lty = 2, lwd = 2)
lines(fyear, upper^2, lty = 2, lwd = 2)
```



ARIMA

Monthly Price of Oil: January 1986–January 2006

```
library(TSA)
data(oil.price)

par(las = 1, mgp = c(2, 1, 0), mar = c(3.5, 3.5, 0.8, 0.6))
plot(oil.price, ylab = 'Price per Barrel', type = 'l')
points(oil.price, pch = 16, cex = 0.5)
grid()
```



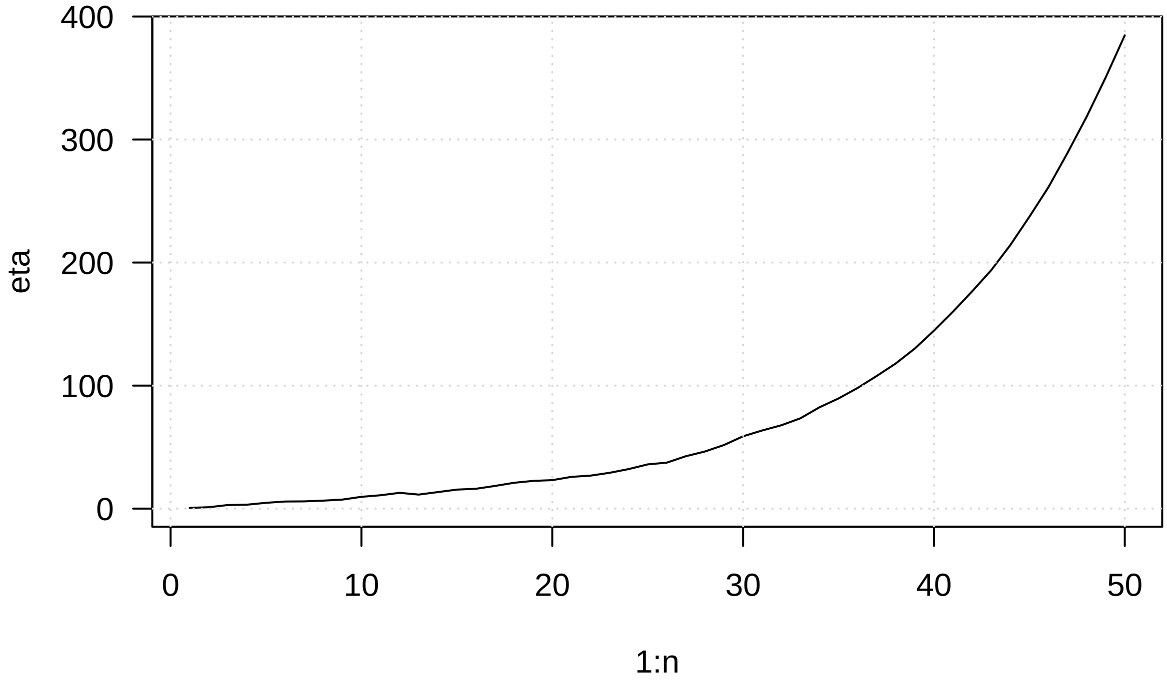
A stationary model does not seem to be reasonable. However, it is also not clear which (deterministic) trend model is appropriate

An explosive AR model

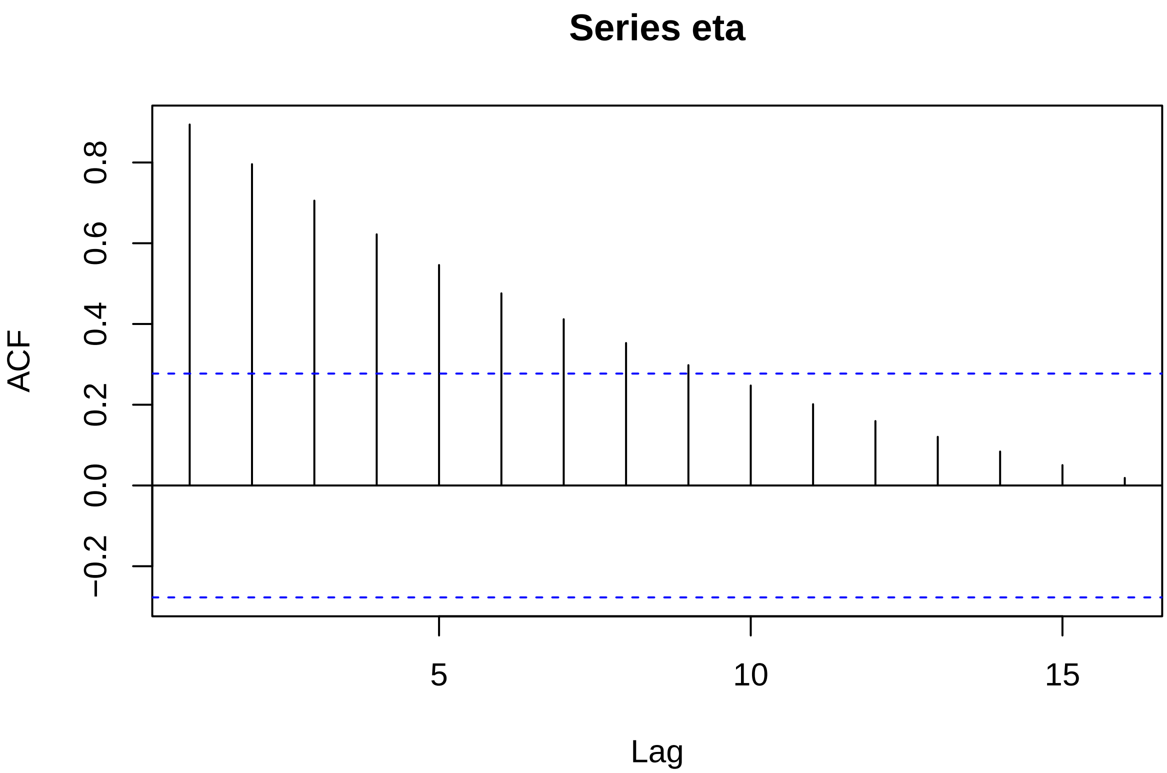
$\eta_t = 1.1\eta_{t-1} + Z_t$

```
n <- 50; phi <- 1.1
set.seed(128)
z <- rnorm(n)
eta <- c()
eta[1] <- z[1]
for (i in 2:n) eta[i] <- phi * eta[i - 1] + z[i]
```

```r
plot(1:n, eta, las = 1, type = "l")
grid()
```




```r
acf(eta)
```



ARIMA(1,1,0)

```r
sim <- arima.sim(list(order = c(1, 1, 0), ar = 0.5), n = 200)
sim_diff <- diff(sim)

par(las = 1, mgp = c(2, 1, 0), mar = c(3.5, 3.5, 0.8, 0.6), mfrow = c(3, 2))
plot(1:201, sim, type = "l", ylab = expression(X[t]), xlab = "Time")
plot(1:200, sim_diff, type = "l", ylab = expression(Y[t]), xlab = "Time")
acf(sim)
acf(sim_diff)
pacf(sim)
pacf(sim_diff)
```