Longitudinal Analysis

Richard White

2018-05-09

Contents

Chapter 1

Syllabus

Instructor: Richard White [richard.white@fhi.no]

Time: 09:30 - 15:00, 18th September 2017

Location: Main auditorium, L8, Lindern Campus, Folkehelseinstittutet, Oslo

Language: English

Format and Procedures

09:00 - 10:00: Lecture 1

10:00 - 10:10: Break

10:10 - 11:10: Lecture 2

10:10 - 10:15: Break

11:15 - 11:45: Examples from FHI

Description

This course will provide a basic overview of general statistical methodology that can be useful in the areas of infectious diseases, environmental medicine, and labwork. By the end of this course, students will be able to identify appropriate statistical methods for a variety of circumstances.

This course will **not** teach students how to implement these statistical methods, as there is not sufficient time. The aim of this course is to enable the student to identify which methods are required for their study, allowing the student to identify their needs for subsequent methods courses, self-learning, or external help.

You should register for this course if you are one of the following:

- Have experience with applying statistical methods, but are sometimes confused or uncertain as to whether or not you have selected the correct method.
- Do not have experience with applying statistical methods, and would like to get an overview over which methods are applicable for your projects so that you can then undertake further studies in these areas.

Lecture 1

1. Identifying continuous, categorical, count, and censored variables
2. Identifying exposure and outcome variables
3. Identifying when t-tests (paired and unpaired) should be used
4. Identifying when non-parametric t-test equivalents should be used
5. Identifying when ANOVA should be used
6. Identifying when linear regression should be used

7. Identifying the similarities between t-tests, ANOVA, and regression
8. Identifying when logistic regression models should be used
9. Identifying when Poisson/negative binomial and cox regression models should be used
10. Identifying when chi-squared/fisher's exact test should be used

Lecture 2

1. Identifying when data does not have any dependencies (i.e. all observations are independent of each other) versus when data has complicated dependencies (i.e. longitudinal data, matched data, multiple cohorts)
2. Identifying when mixed effects regression models should be used
3. Identifying when conditional logistic regression models should be used
4. (TBD) Understanding the different imputation methods used when lab data is below the limit of detection (LOD)
5. (TBD) Understanding the best practices for data files and project folders

Prerequisites

To participate in this course it is recommended that you have some experience with either research or data.

Additional information

For the last 30 minutes of the course we will be going through examples of analyses performed at FHI and identifying which statistical methods are appropriate. If you would like your analysis to be featured/included in this section, please send an email to richard.white@fhi.no briefly describing your problem.

Chapter 2

Reference

2.1 Introduction

There are two important definitions in this course:

- Panel data
- Autocorrelation

Panel data is a set of data with measurements repeated at equally spaced points. For example, weight data recorded every day, or every week, or every year would be considered panel data. A person who records three weight measurements randomly in 2018 would not be considered panel data.

When you have panel data, autocorrelation is the correlation between subsequent observations. For example, if you have daily observations, then the 1 day autocorrelation is the correlation between observations 1 day apart, and likewise the 2 day autocorrelation is the correlation between observations 2 days apart.

In this course we will consider 5 scenarios where we have multiple observations for each geographical area:

- Panel data: One geographical area, no autocorrelation
- Panel data: One geographical area, with autocorrelation
- Not panel data: Multiple geographical areas
- Panel data: Multiple geographical areas, no autocorrelation
- Panel data: Multiple geographical areas, with autocorrelation

Note, the following scenario can be covered by standard regression models:

- Multiple geographical areas, one time point/observation per geographical area

2.2 Identifying your scenario

2.2.1 Step 1: Do you have panel data?

This step should be fairly simple. If your data has equally spaced intervals between them, you have panel data.

2.2.2 Step 2: Do you have multiple geographical areas?

Again, fairly simple, just look at your data.

2.2.3 Step 3: Do you have autocorrelation?

Firstly, you must run a model pretending that you do not have autocorrelation.

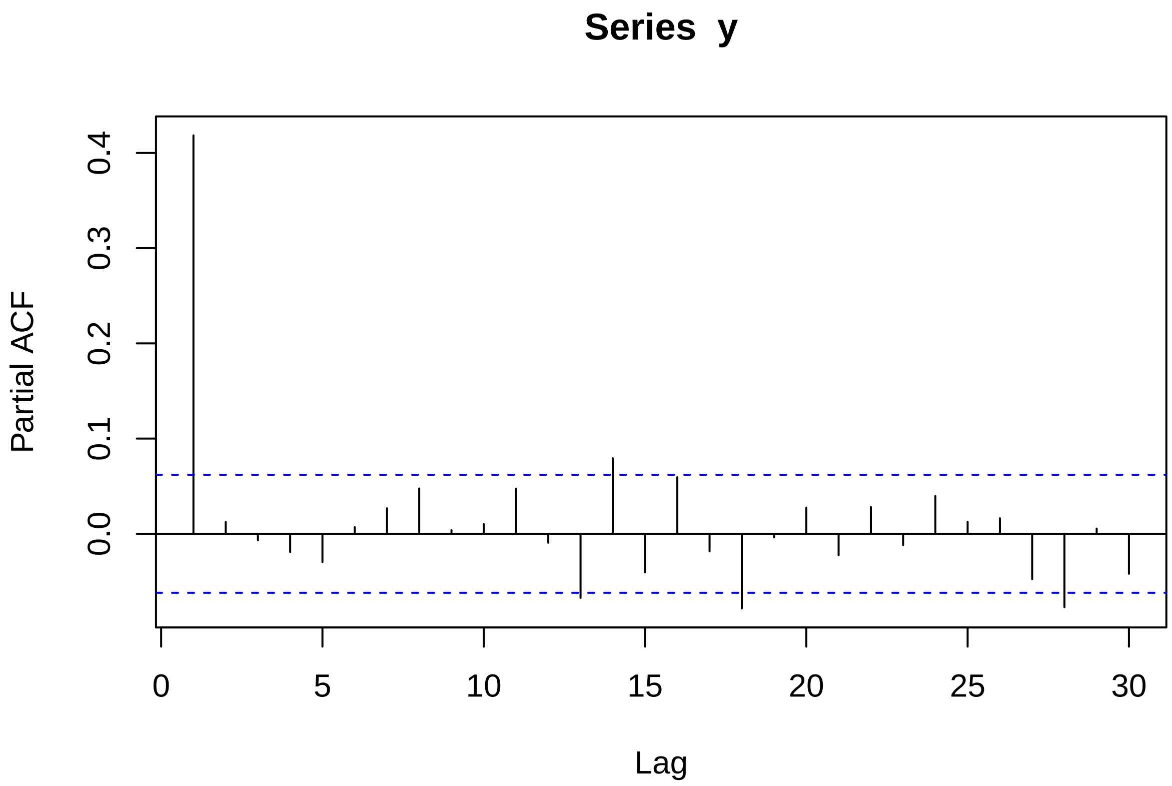
You then inspect the residuals from the model and see if autocorrelation exists. This is done with two statistical procedures: `pacf` (for `autoregressive models`, the most common type of autocorrelation), and `acf` (for `moving average models`, a less common type of autocorrelation).

2.2.4 AR(1) data

```
y <- round(as.numeric(arima.sim(model=list("ar"=c(0.5)), rand.gen = rnorm, n=1000)))
```

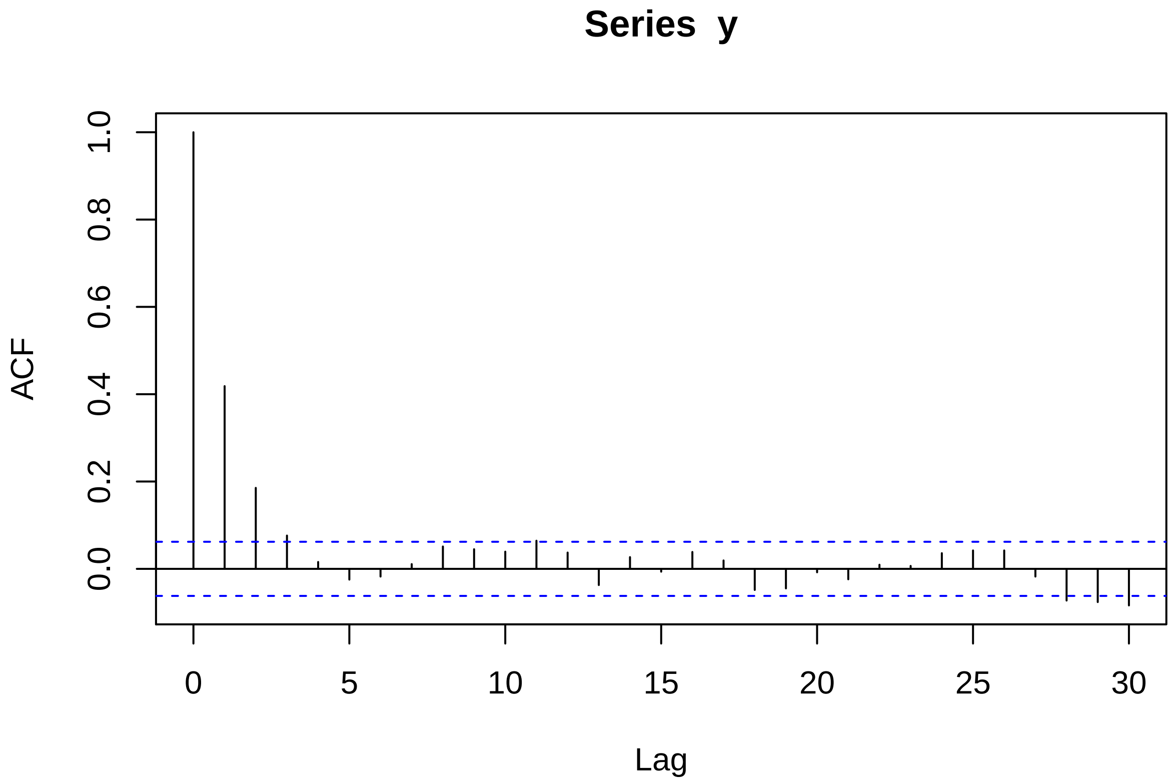
With `autoregressive` data, a `pacf` plot contains a number of sharp significant lines, indicating how many subsequent observations have autocorrelation. i.e. if one line is significant, it means that each observation is only correlated with its preceeding observation (`AR(1)`). If two lines are significant, it means that each observation is correlated with its two preecing observations (`AR(2)`). The following plot represents `AR(1)` data.

```
pacf(y)
```



With `autoregressive` data, an `acf` plot contains a number of decreasing lines. The following `acf` plot represents some sort of `AR` data. Note that the `acf` plot displays `lag 0` (which is pointless and can be ignored), while the `pacf` plot does not.

```
acf(y)
```

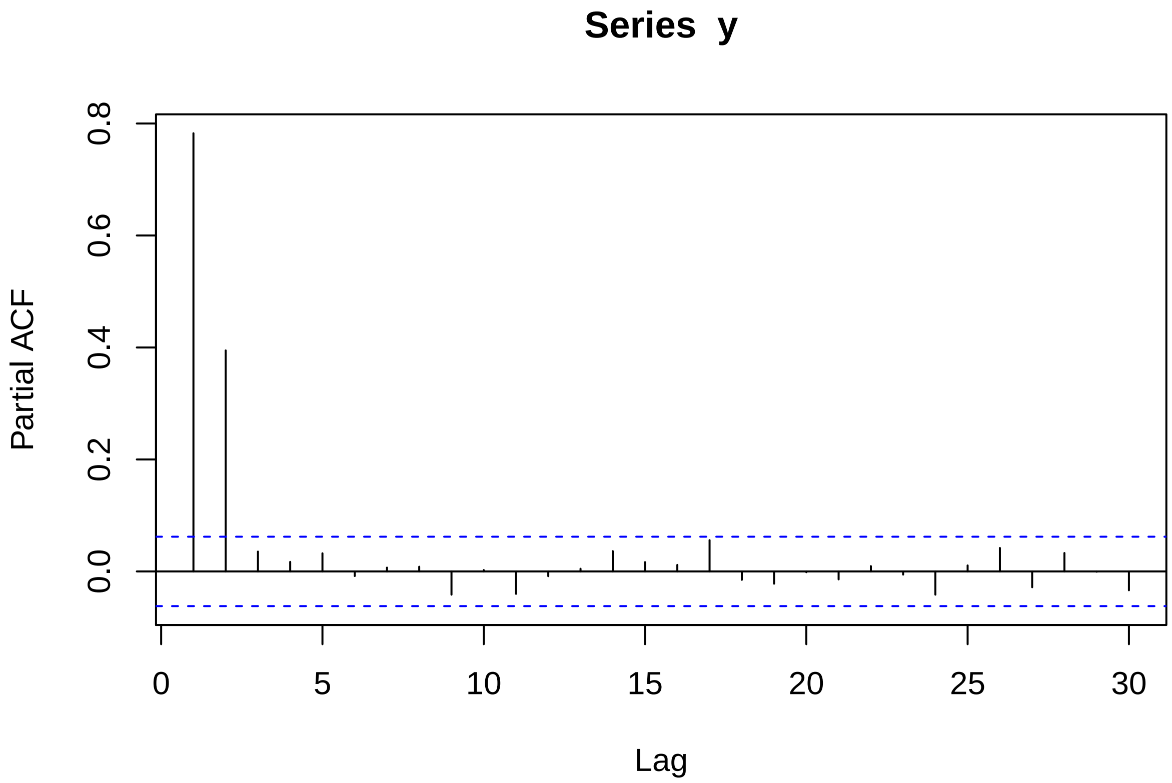


2.2.5 AR(2) data

```r
y <- round(as.numeric(arima.sim(model=list("ar"=c(0.5,0.4)), rand.gen = rnorm, n=1000)))
```

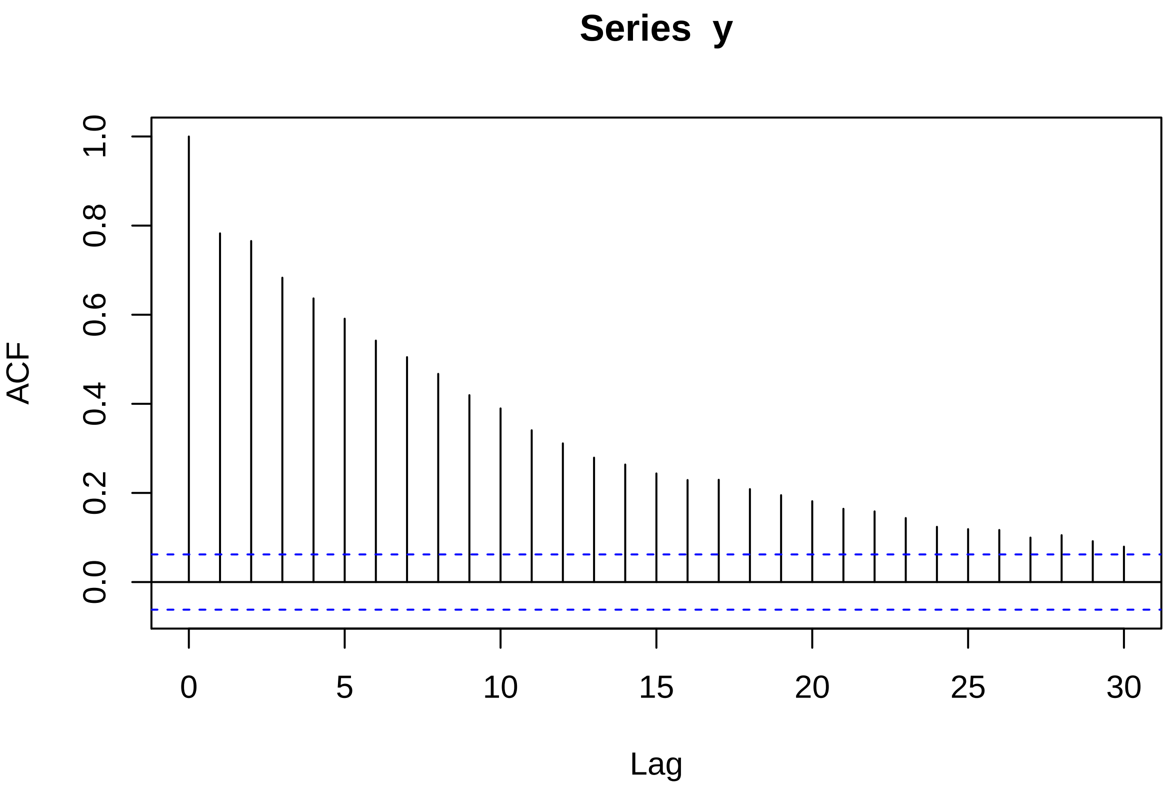
The following `pacf` plot represents `AR(2)` data:

```r
pacf(y)
```



The following acf plot represents some sort of AR data:

```
acf(y)
```

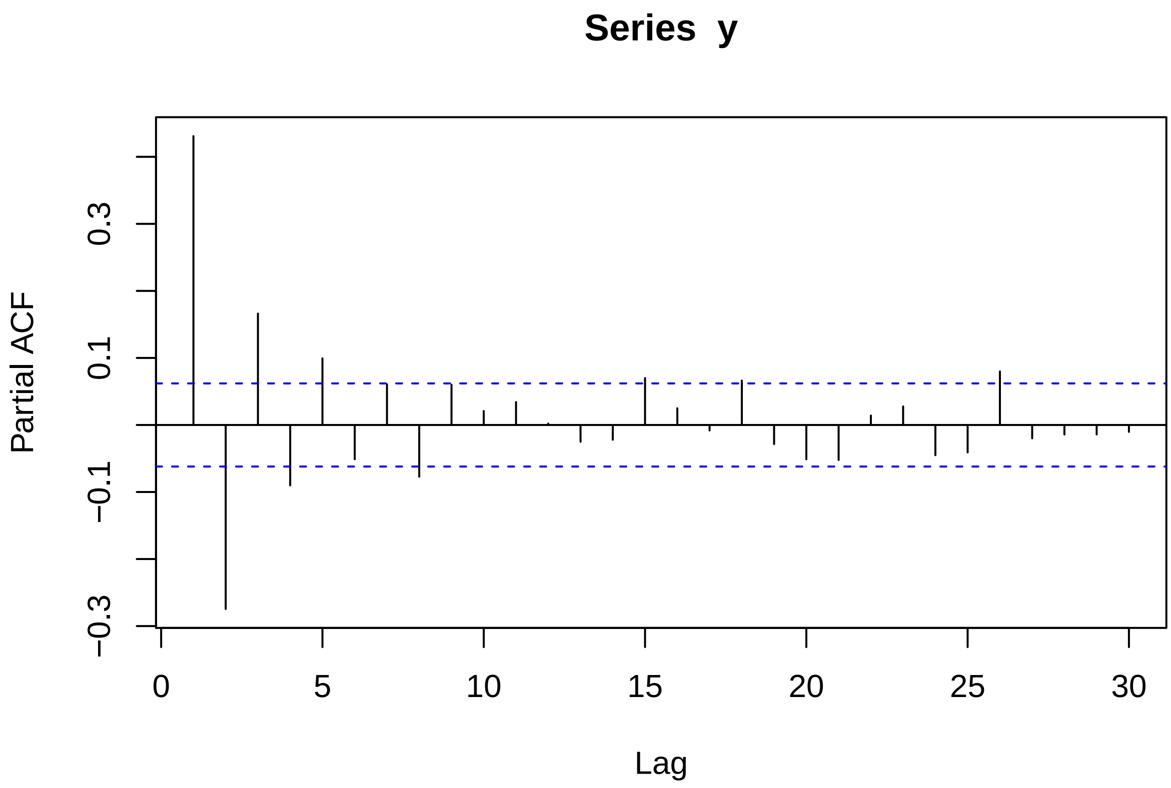


2.2.6 MA(1) data

```
y <- round(as.numeric(arima.sim(model=list("ma"=c(0.9)), rand.gen = rnorm, n=1000)))
```

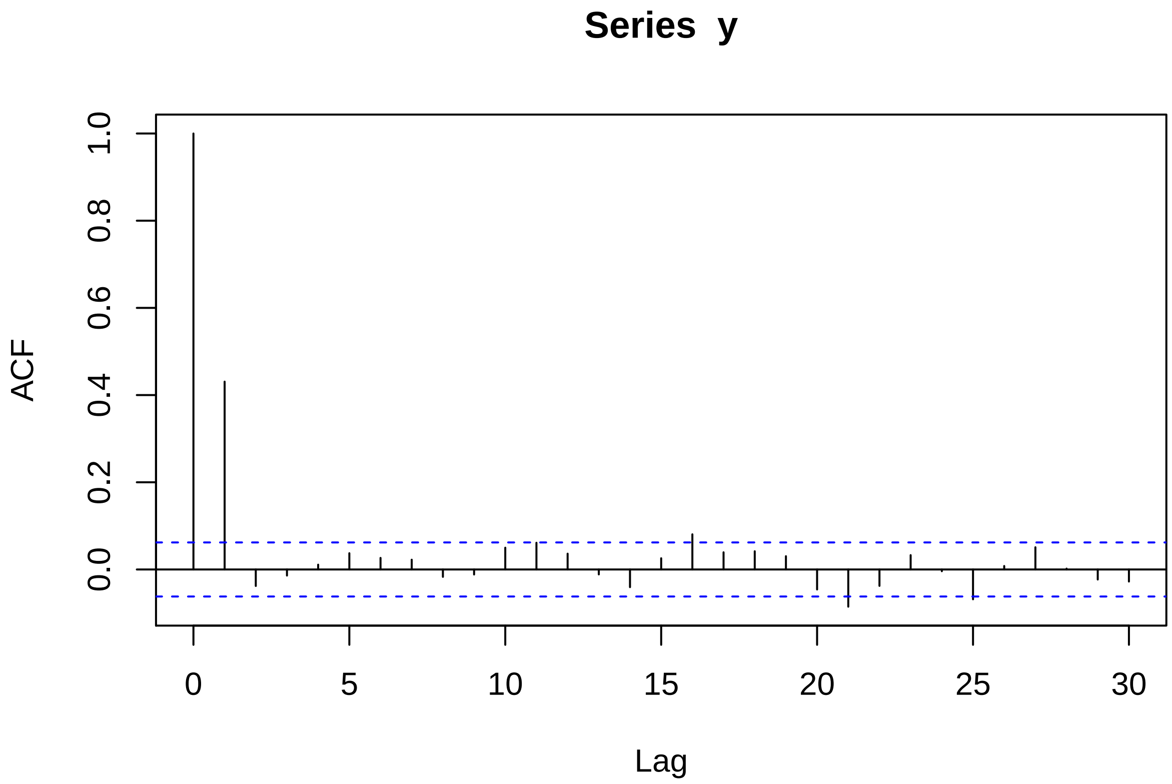
With `moving average` data, a `pacf` plot contains a number of decreasing lines. The following `pacf` plot represents some sort of `MA` data.:

```
pacf(y)
```



With `moving average` data, an `acf` plot contains a number of sharp significant lines, demarking how many subsequent observations have autocorrelation. i.e. if one line is significant, it means that each observation is only correlated with its preceeding observation. If two lines are significant, it means that each observation is correlated with its two preecing observations. The following plot represents `MA(1)` data. Note that the `acf` plot displays `lag 0` (which is pointless and can be ignored), while the `pacf` plot does not.

```
acf(y)
```

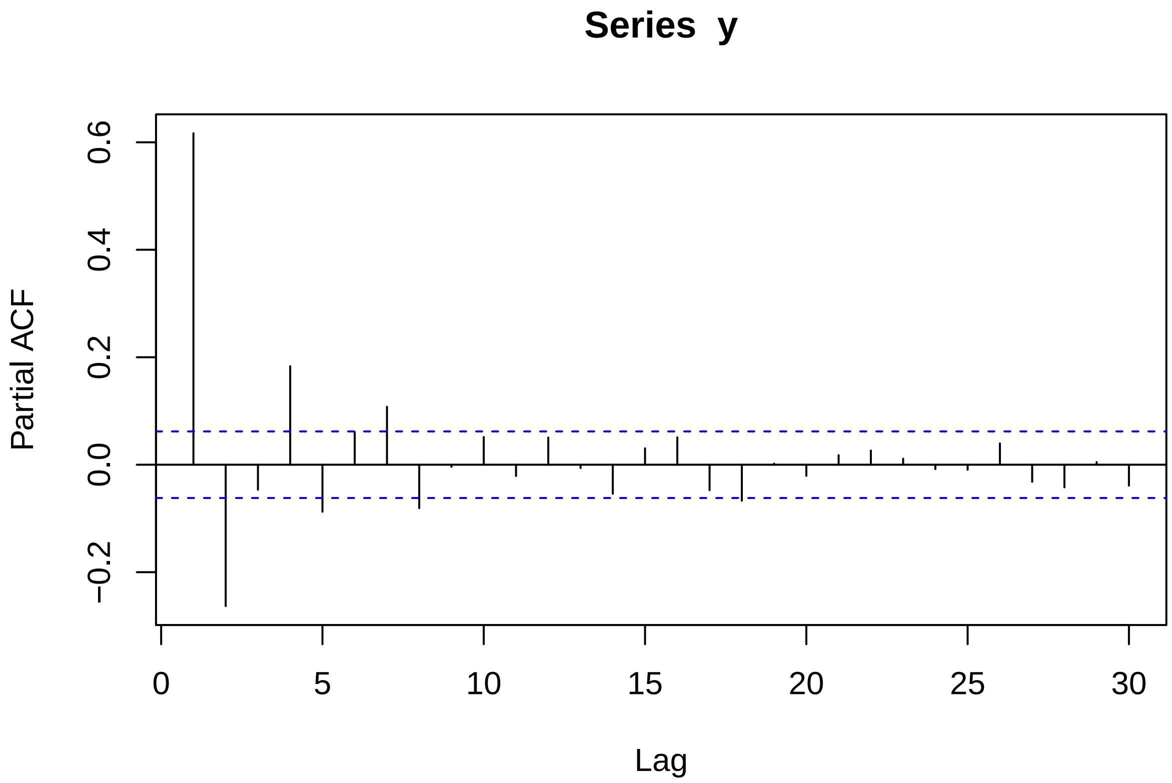


2.2.7 MA(2) data

```
y <- round(as.numeric(arima.sim(model=list("ma"=c(0.9,0.6)), rand.gen = rnorm, n=1000)))
```

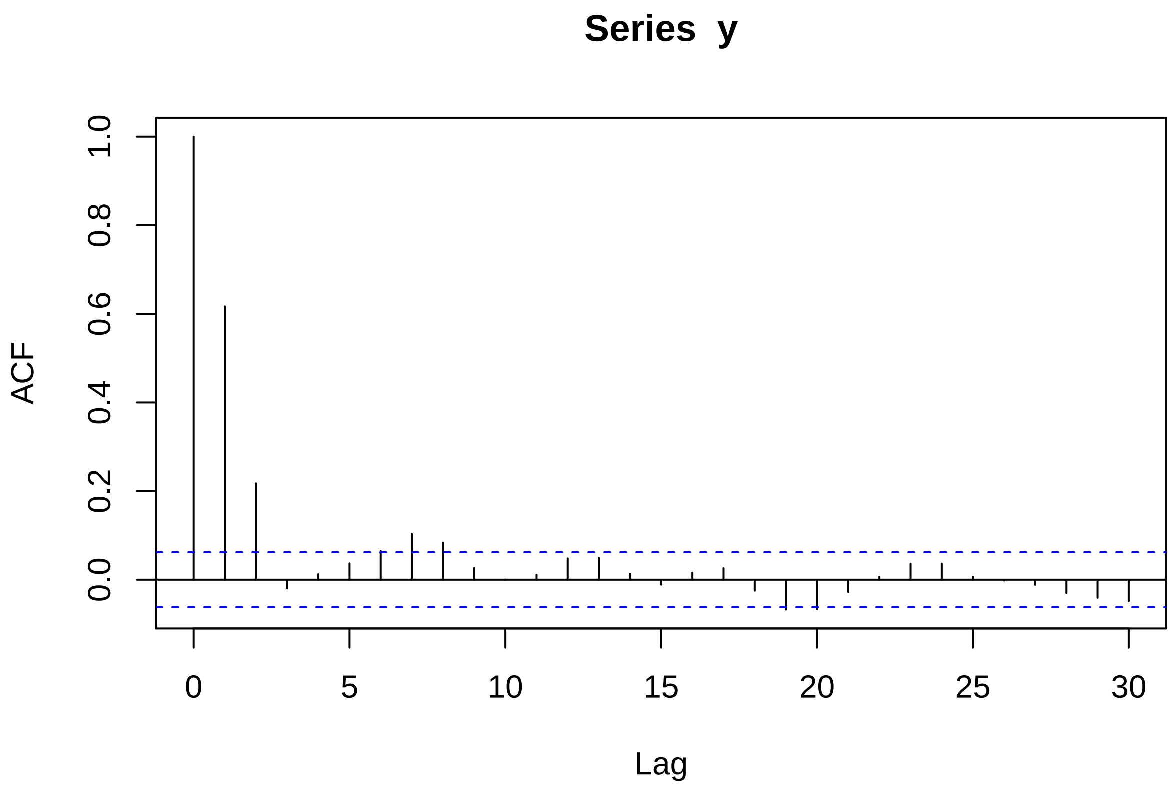
The following `pacf` plot represents some sort of `MA` data.

```
pacf(y)
```



The following `acf` plot represents `MA(2)` data:

```r
acf(y)
```



Chapter 3

Panel data: One area without autocorrelation

```r
dir.create("data")
```

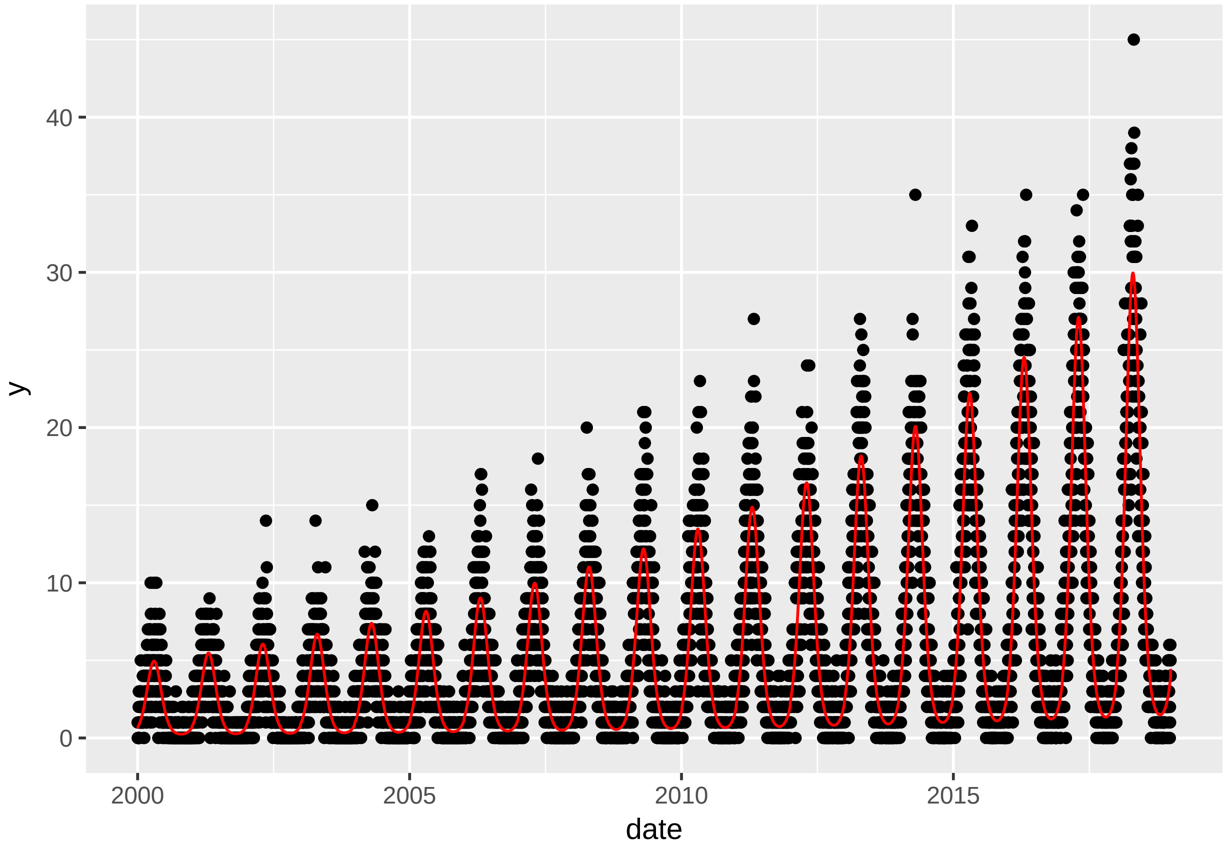
```
## Warning in dir.create("data"): 'data' already exists
```

```r
library(data.table)
library(ggplot2)
set.seed(4)

AMPLITUDE <- 1.5
SEASONAL_HORIZONTAL_SHIFT <- 20

d <- data.table(date=seq.Date(
  from=as.Date("2000-01-01"),
  to=as.Date("2018-12-31"),
  by=1))
d[,year:=as.numeric(format.Date(date,"%G"))]
d[,week:=as.numeric(format.Date(date,"%V"))]
d[,month:=as.numeric(format.Date(date,"%m"))]
d[,yearMinus2000:=year-2000]

d[,dayOfYear:=as.numeric(format.Date(date,"%j"))]
d[,seasonalEffect:=sin(2*pi*(dayOfYear-SEASONAL_HORIZONTAL_SHIFT)/365)]
d[,mu := exp(0.1 + yearMinus2000*0.1 + seasonalEffect*AMPLITUDE)]
d[,y:=rpois(.N,mu)]

fwrite(d,"data/chapter_3.csv")
```

3.1 True data

Here we show the true data, and note that there is an increasing annual trend (the data gets higher as time goes on) and there is a seasonal pattern (one peak/trough per year)

```
q <- ggplot(d,aes(x=date))
q <- q + geom_point(mapping=aes(y=y))
q <- q + geom_line(mapping=aes(y=mu),colour="red")
q
```

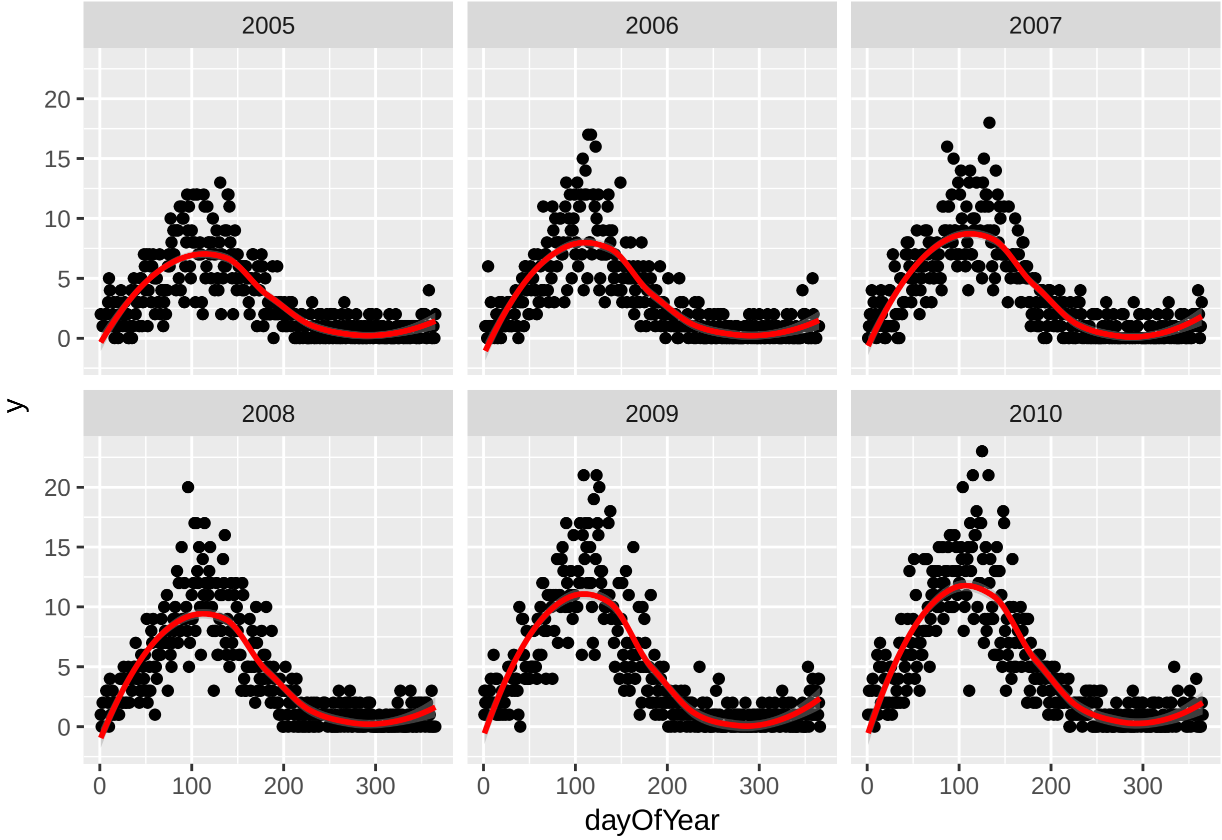


3.2 Investigation

Pretending we have no prior knowledge of our dataset, we display the data for few years and see a clear seasonal trend

```
q <- ggplot(d[year %in% c(2005:2010)],aes(x=dayOfYear,y=y))
q <- q + facet_wrap(~year)
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

```
## `geom_smooth()` using method = 'loess'
```



3.3 Seasonality

If we want to investigate the seasonality of our data, and identify when are the peaks and troughs, we have a few ways to approach this.

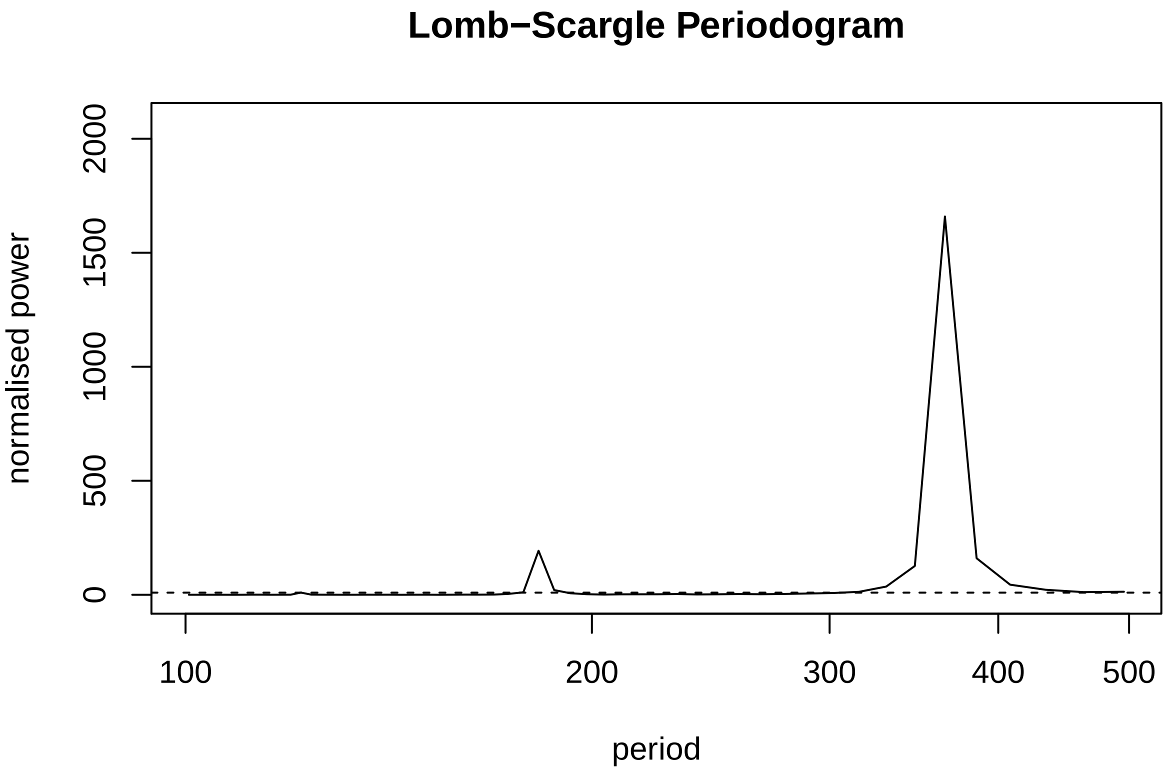
Non-parametric approaches are flexible and easy to implement, but they can lack power and be hard to interpret: - Create a categorical variable for the seasons (e.g. `spring`, `summer`, `autumn`, `winter`) and include this in the regression model - Create a categorical variable for the months (e.g. `Jan`, `Feb`, …, `Dec`) and include this in the regression model

Parametric approaches are more powerful but require more effort: - Identify the periodicity of the seasonality (how many days between peaks?) - Using trigonometry, transform `day of year` into variables that appropriately model the observed periodicity - Obtain coefficient estimates - Back-transform these estimates into human-understandable values (day of peak, day of trough)

The non-parametric approaches are simple and we will therefore not cover them in this course. We will briefly examine the parametric approach.

The Lomb-Scargle Periodogram shows a clear seasonality with a period of 365 days

```
lomb::lsp(d$y,from=100,to=500,ofac=1,type="period")
```



We then generate two new variables `cos365` and `sin365` and perform a likelihood ratio test to see if they are significant or not. This is done with two simple poisson regressions.

When we do not have autocorrelation, we can use the `glm` function in R.

```
d[,cos365:=cos(dayOfYear*2*pi/365)]
d[,sin365:=sin(dayOfYear*2*pi/365)]

fit0 <- glm(y~yearMinus2000, data=d, family=poisson())
fit1 <- glm(y~yearMinus2000 + sin365 + cos365, data=d, family=poisson())

print(lmtest::lrtest(fit0, fit1))
```

```
## Likelihood ratio test
##
## Model 1: y ~ yearMinus2000
## Model 2: y ~ yearMinus2000 + sin365 + cos365
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1   2 -27287
## 2   4 -12805  2 28963  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the likelihood ratio test for `sin365` and `cos365` was significant, meaning that there is significant seasonality with a 365 day periodicity in our data.

```
print(summary(fit1))
```

```
##
## Call:
## glm(formula = y ~ yearMinus2000 + sin365 + cos365, family = poisson(),
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.7499  -0.9167  -0.1370   0.5955   3.2193
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.086654   0.014940    5.80 6.62e-09 ***
## yearMinus2000  0.100461   0.001049   95.75  < 2e-16 ***
## sin365         1.428417   0.010434  136.90  < 2e-16 ***
## cos365        -0.512912   0.008666  -59.19  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 46221.4  on 6939  degrees of freedom
## Residual deviance:  7259.2  on 6936  degrees of freedom
## AIC: 25619
##
## Number of Fisher Scoring iterations: 5
```

We also see that the coefficient for year is `0.1` which means that for each additional year, the outcome increases by `exp(0.1)=1.11`.

Through the likelihood ratio test we saw a clear significant seasonal effect. We can now use trigonometry to back-calculate the amplitude and location of peak/troughs from the `cos365` and `sin365` estimates:

```
b1 <- 1.428417 # sin coefficient
b2 <- -0.512912 # cos coefficient
amplitude <- sqrt(b1^2 + b2^2)
p <- atan(b1/b2) * 365/2/pi
if (p > 0) {
    peak <- p
    trough <- p + 365/2
} else {
    peak <- p + 365/2
    trough <- p + 365
}
if (b1 < 0) {
    g <- peak
    peak <- trough
    trough <- g
}
print(sprintf("amplitude is estimated as %s, peak is estimated as %s, trough is estimated as %s",round(
```

```
## [1] "amplitude is estimated as 1.52, peak is estimated as 111, trough is estimated as 294"
```

```
print(sprintf("true values are: amplitude: %s, peak: %s, trough: %s",round(AMPLITUDE,2),round(365/4+SEA
```

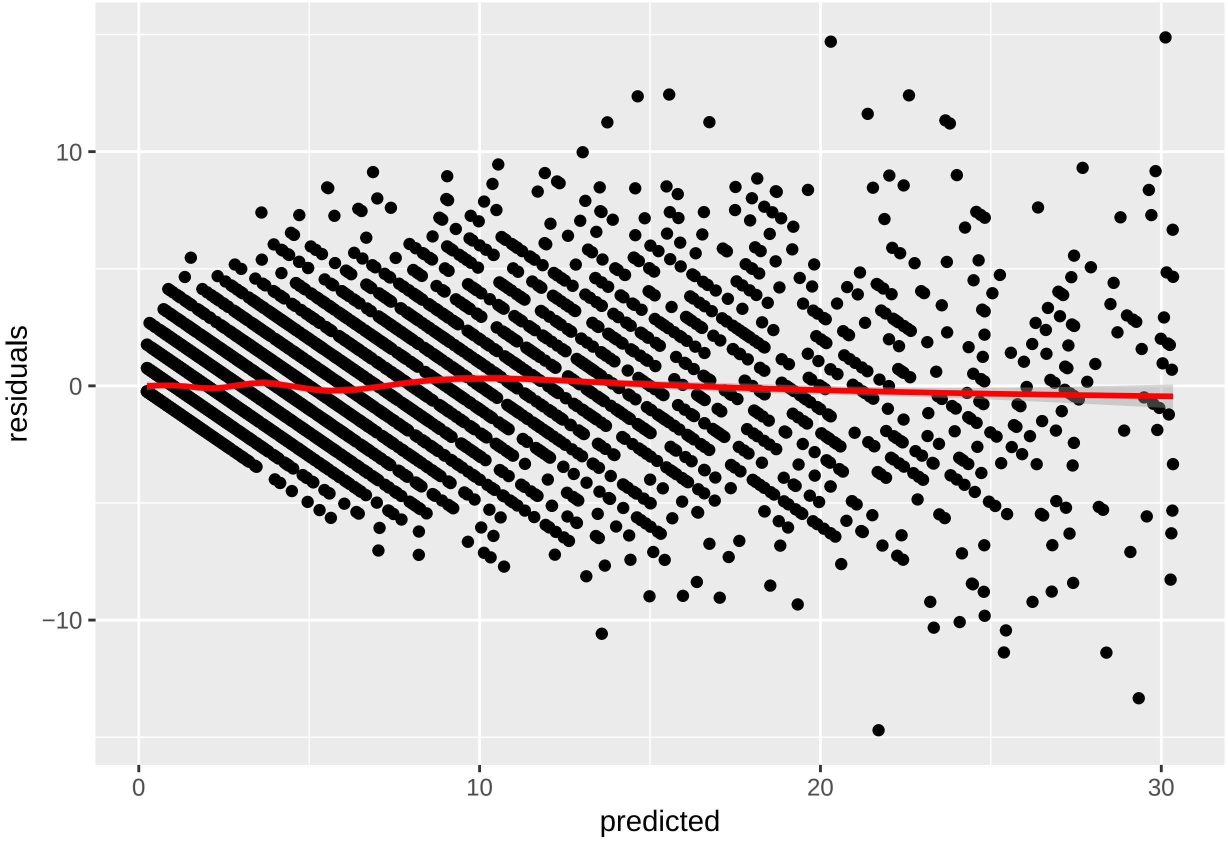
```
## [1] "true values are: amplitude: 1.5, peak: 111, trough: 294"
```

Note: I have no idea what amplitude means outside of a linear regression!

We now investigate our residuals to determine if we have a good fit:

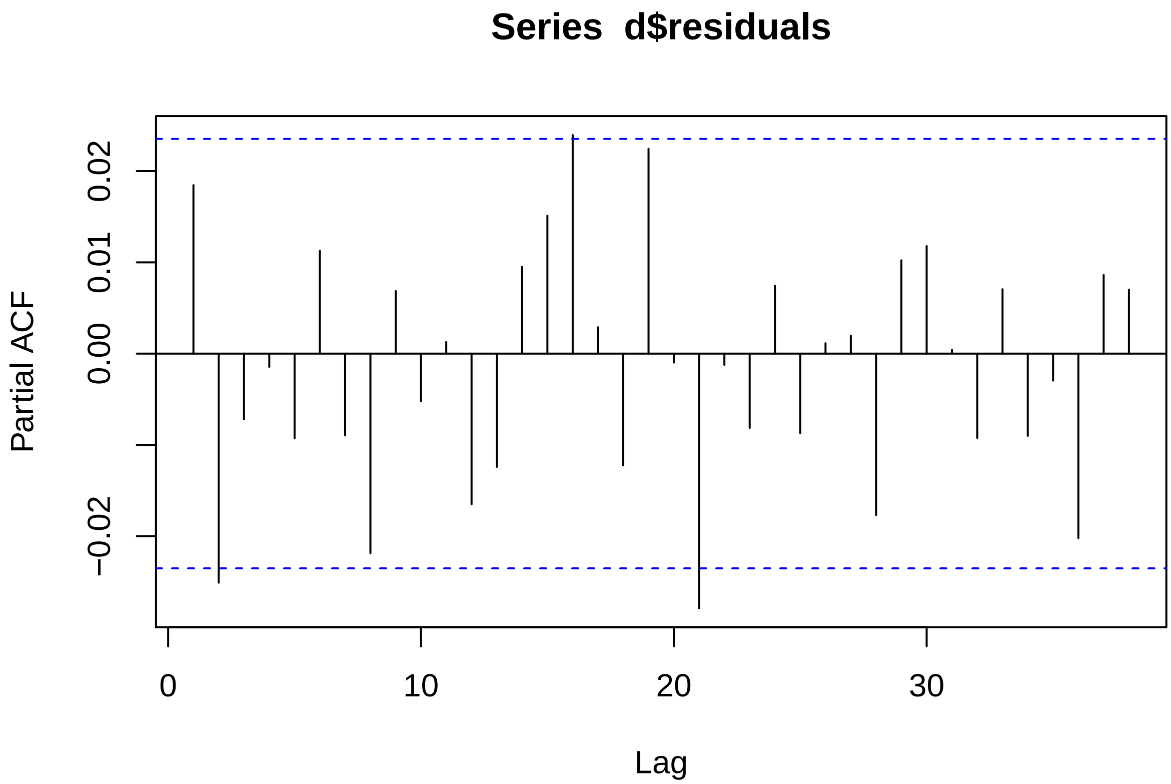
```
d[,residuals:=residuals(fit1, type = "response")]
d[,predicted:=predict(fit1, type = "response")]
q <- ggplot(d,aes(x=predicted,y=residuals))
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

```
## `geom_smooth()` using method = 'gam'
```



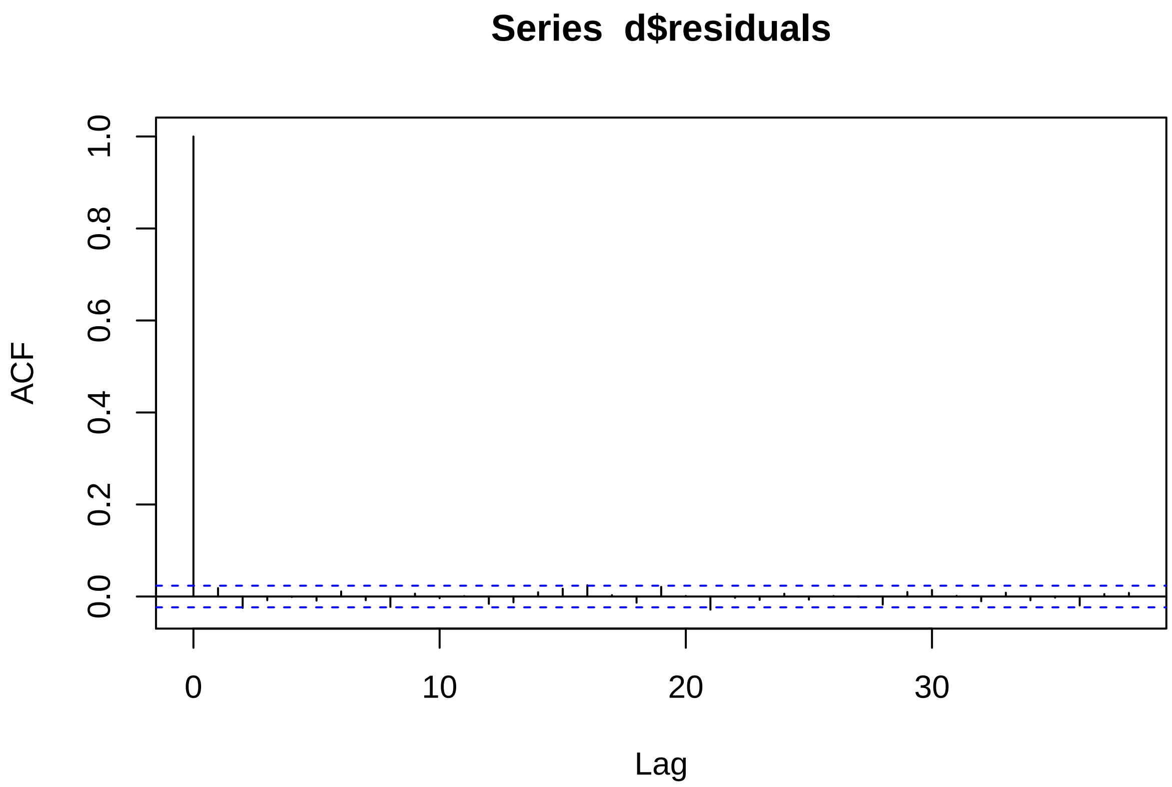
We check the `pacf` of the residuals to ensure that it is not `AR`. If we observe `AR` in our residuals, then this model was not appropriate and we need to use a different model.

```
# this is for AR
pacf(d$residuals)
```



We check the acf of the residuals to ensure that it is not MA. If we observe MA in our residuals, then this model was not appropriate and we need to use a different model.

```
# this is for MA
acf(d$residuals)
```



Chapter 4

Panel data: One area with autocorrelation

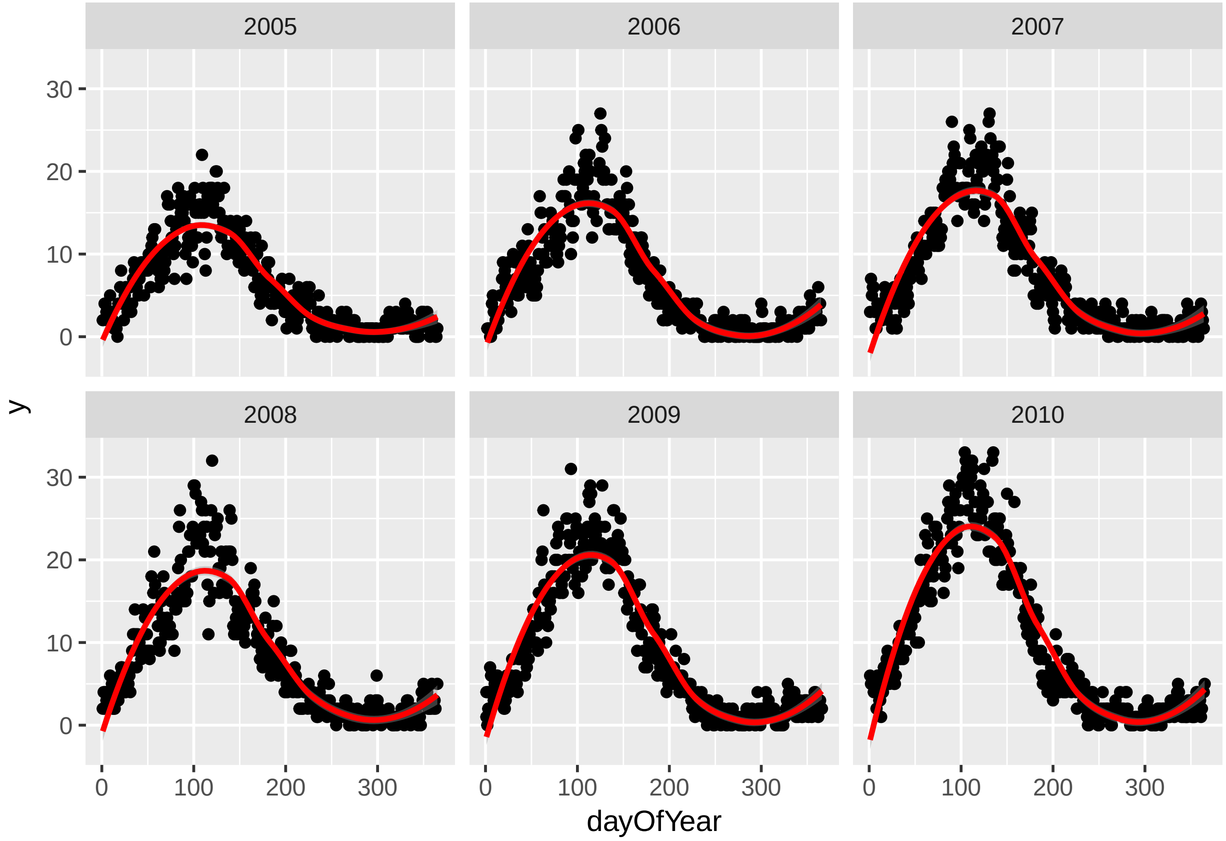
```
library(data.table)
library(ggplot2)
set.seed(4)

AMPLITUDE <- 1.5
SEASONAL_HORIZONTAL_SHIFT <- 20

d <- data.table(date=seq.Date(
  from=as.Date("2000-01-01"),
  to=as.Date("2018-12-31"),
  by=1))
d[,year:=as.numeric(format.Date(date,"%G"))]
d[,week:=as.numeric(format.Date(date,"%V"))]
d[,month:=as.numeric(format.Date(date,"%m"))]
d[,yearMinus2000:=year-2000]
d[,dayOfSeries:=1:.N]

d[,dayOfYear:=as.numeric(format.Date(date,"%j"))]
d[,seasonalEffect:=sin(2*pi*(dayOfYear-SEASONAL_HORIZONTAL_SHIFT)/365)]
d[,mu := exp(0.1 + yearMinus2000*0.1 + seasonalEffect*AMPLITUDE)]
d[,y:=rpois(.N,mu)]
d[,y:=round(as.numeric(arima.sim(model=list("ar"=c(0.5)), rand.gen = rpois, n=nrow(d), lambda=mu)))]
```

4.1 Investigation

We display the data for few years and see a clear seasonal trend

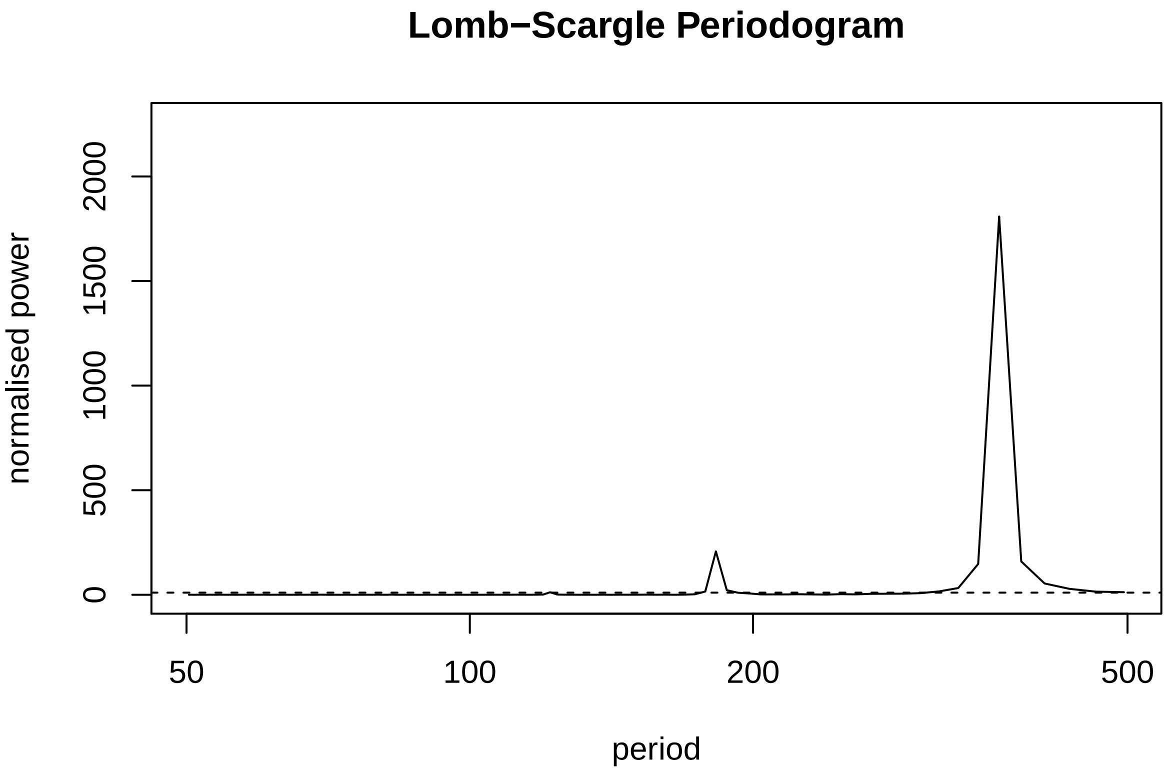
```
q <- ggplot(d[year %in% c(2005:2010)],aes(x=dayOfYear,y=y))
q <- q + facet_wrap(~year)
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

```
## `geom_smooth()` using method = 'loess'
```



The Lomb-Scargle Periodogram shows a clear seasonality with a period of 365 days

```
lomb::lsp(d$y,from=50,to=500,ofac=1,type="period")
```



4.2 Regressions

We then generate two new variables `cos365` and `sin365` and perform a likelihood ratio test to see if they are significant or not. This is done with two simple poisson regressions.

```
d[,cos365:=cos(dayOfYear*2*pi/365)]
d[,sin365:=sin(dayOfYear*2*pi/365)]

fit0 <- glm(y~yearMinus2000, data=d, family=poisson())
fit1 <- glm(y~yearMinus2000+sin365 + cos365, data=d, family=poisson())

print(lmtest::lrtest(fit0, fit1))
```

```
## Likelihood ratio test
##
## Model 1: y ~ yearMinus2000
## Model 2: y ~ yearMinus2000 + sin365 + cos365
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1   2 -43124
## 2   4 -14542  2 57163  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We see that the likelihood ratio test for `sin365` and `cos365` was significant, meaning that there is significant seasonality with a 365 day periodicity in our data.

```
print(summary(fit1))
```

```
##
## Call:
## glm(formula = y ~ yearMinus2000 + sin365 + cos365, family = poisson(),
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6774  -0.6738  -0.0503   0.4920   3.5820
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.7981246  0.0105300   75.80   <2e-16 ***
## yearMinus2000 0.0991480  0.0007416  133.70   <2e-16 ***
## sin365        1.4074818  0.0073418  191.71   <2e-16 ***
## cos365       -0.5390314  0.0061513  -87.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 81832.6  on 6939  degrees of freedom
## Residual deviance:  5217.8  on 6936  degrees of freedom
## AIC: 29093
##
## Number of Fisher Scoring iterations: 4
```

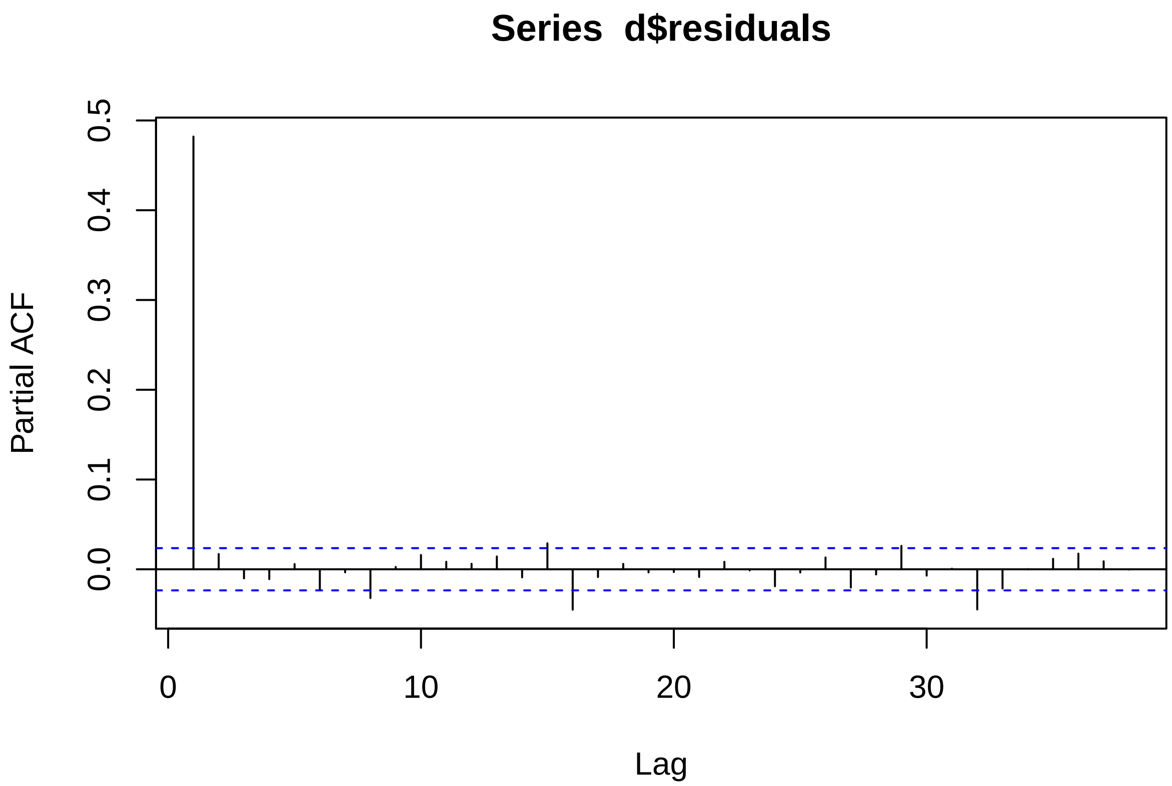
We also see that the coefficient for year is `0.1` which means that for each additional year, the outcome increases by `exp(0.1)=1.11`.

4.3 Residual analysis

```
d[,residuals:=residuals(fit1, type = "response")]
d[,predicted:=predict(fit1, type = "response")]
```

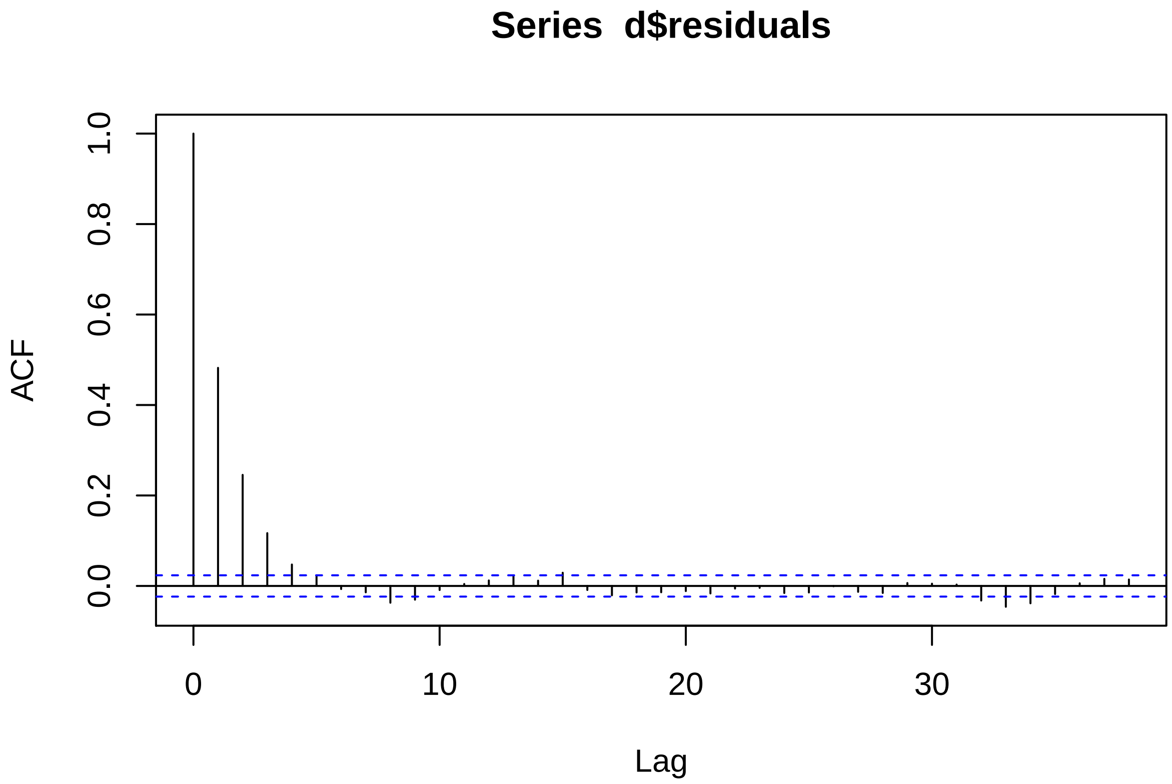
We can see a clear `AR(1)` pattern in our residuals.

```
# this is for AR
pacf(d$residuals)
```



And again we see some sort of `AR` pattern in our residuals.

```
# this is for MA
acf(d$residuals)
```



This means our model is bad, we have autocorrelation. We now need to change our model to account for this `AR(1)` autocorrelation!

4.4 Regression with AR(1) correlation in residuals

First we create an id variable. This generally corresponds to geographical locations, or people. In this case, we only have one geographical location, so our id for all observations is 1. This lets the computer know that all data belongs to the same group.

When we have autocorrelation, we can use the MASS::glmPQL function in R.

```
d[,ID:=1]
# this is for MA
fit <- MASS::glmmPQL(y~yearMinus2000+sin365 + cos365, random = ~ 1 | ID,
                family = poisson, data = d,
                correlation=nlme::corAR1(form=~dayOfSeries|ID))
```

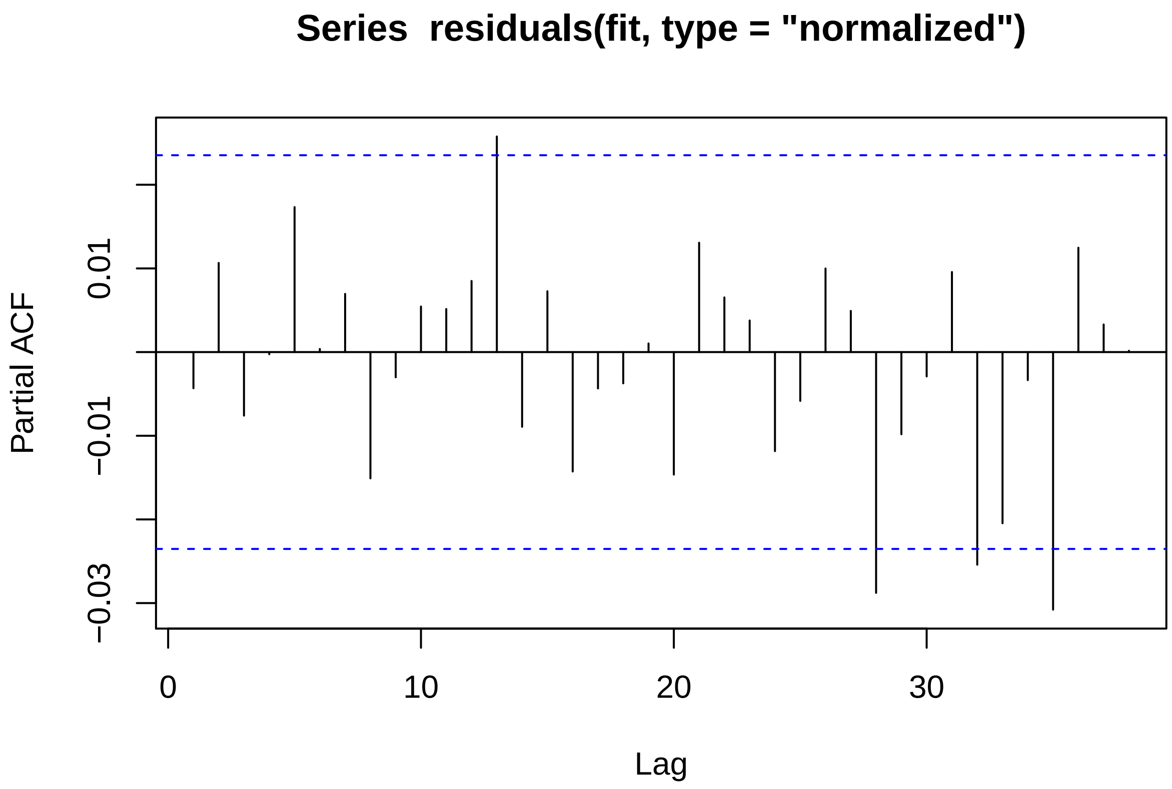
```
## iteration 1
```

```
summary(fit)
```

```
## Linear mixed-effects model fit by maximum likelihood
##  Data: d
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | ID
##          (Intercept) Residual
## StdDev: 1.149069e-05 0.841689
##
## Correlation Structure: AR(1)
##  Formula: ~dayOfSeries | ID
##  Parameter estimate(s):
##       Phi
## 0.4926123
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: y ~ yearMinus2000 + sin365 + cos365
##                    Value   Std.Error   DF   t-value p-value
## (Intercept)    0.7980540 0.015203158 6936  52.49265       0
## yearMinus2000  0.0991582 0.001070583 6936  92.62077       0
## sin365         1.4074339 0.010596649 6936 132.81876       0
## cos365        -0.5389807 0.008876447 6936 -60.72031       0
##  Correlation:
##               (Intr) yM2000 sin365
## yearMinus2000 -0.832
## sin365        -0.409  0.000
## cos365         0.186  0.000 -0.158
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.89886753 -0.75775062 -0.05982255  0.60730690  6.49964494
##
## Number of Observations: 6940
## Number of Groups: 1
```

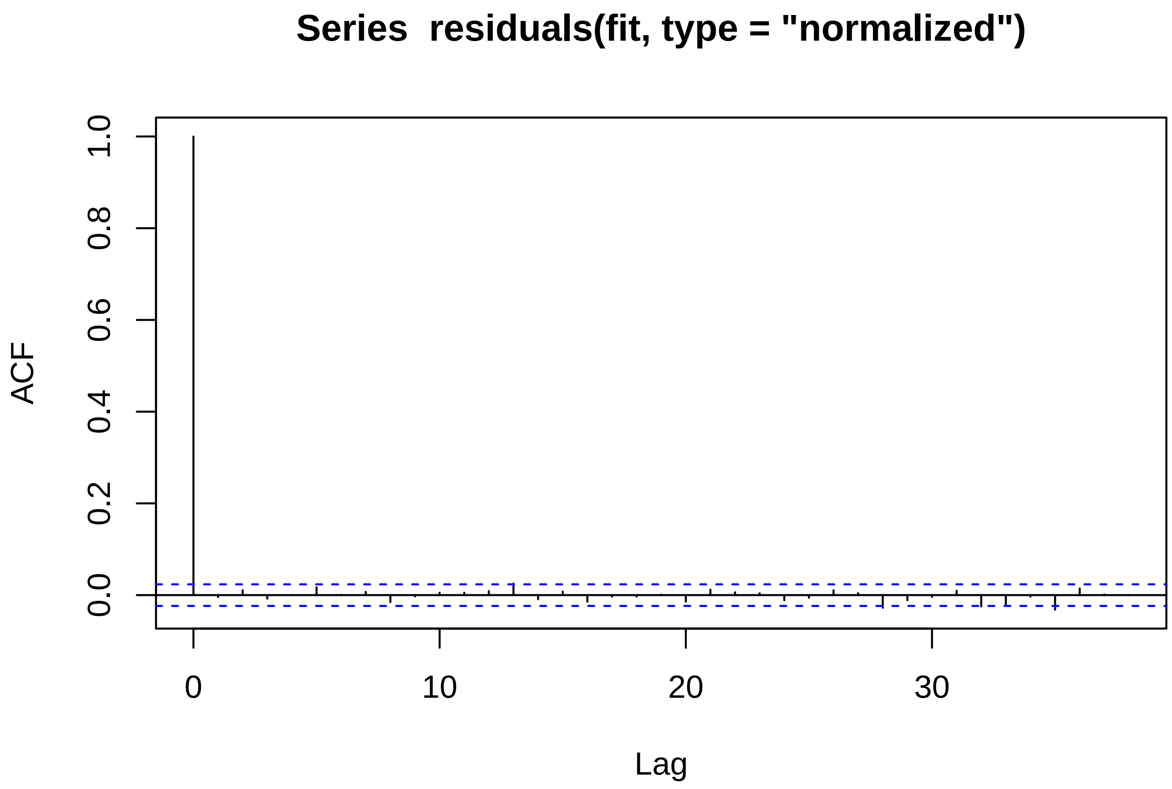
We can see that the residuals no longer display any signs of autocorrelation.

```
pacf(residuals(fit, type = "normalized")) # this is for AR
```



We can see that the residuals no longer display any signs of autocorrelation.

```
acf(residuals(fit, type = "normalized")) # this is for MA
```



We also obtain the same estimates that we did in the last chapter.

```
b1 <- 1.3936185 # sin coefficient
b2 <- -0.5233866 # cos coefficient
amplitude <- sqrt(b1^2 + b2^2)
p <- atan(b1/b2) * 365/2/pi
if (p > 0) {
    peak <- p
    trough <- p + 365/2
} else {
    peak <- p + 365/2
    trough <- p + 365
}
if (b1 < 0) {
    g <- peak
    peak <- trough
    trough <- g
}
print(sprintf("amplitude is estimated as %s, peak is estimated as %s, trough is estimated as %s",round(a
```

```
## [1] "amplitude is estimated as 1.49, peak is estimated as 112, trough is estimated as 295"
```

```
print(sprintf("true values are: amplitude: %s, peak: %s, trough: %s",round(AMPLITUDE,2),round(365/4+SEAS
```

```
## [1] "true values are: amplitude: 1.5, peak: 111, trough: 294"
```

Chapter 5

Not panel data: Multiple areas

```
library(data.table)
library(lme4)
```

```
## Loading required package: Matrix
```

```
## Loading required package: methods
```

```
set.seed(4)

fylkeIntercepts <- data.table(fylke=1:20,fylkeIntercepts=rnorm(20))

d <- data.table(fylke=rep(1:20,each=100))
d <- merge(d,fylkeIntercepts,by="fylke")
d[,mainIntercept:=3]
d[,x:=runif(.N)]
d[,year:=sample(c(1950:2018),.N,replace=T)]
d[,mu := exp(mainIntercept + fylkeIntercepts + 3*x)]
d[,y:=rpois(.N,mu)]
```

5.1 Investigating the data

We can see from the data that we have 20 geographical areas (`fylke`) with 100 observations for each fylke, but the sampling did not happen consistently (some years have multiple measurements, other years have no measurements).

This means we have: - multiple geographical areas - multiple observations in each geographical area - not panel data

```
print(d)
```

```
##       fylke fylkeIntercepts mainIntercept          x year        mu   y
##    1:     1       0.2167549             3 0.93831909 1966 416.42739 392
##    2:     1       0.2167549             3 0.24217109 1981  51.58692  51
##    3:     1       0.2167549             3 0.56559453 1972 136.12022 135
##    4:     1       0.2167549             3 0.18089910 1950  42.92490  39
##    5:     1       0.2167549             3 0.90449929 1951 376.24959 367
##   ---
## 1996:    20      -0.2834446             3 0.89237059 1995 220.00872 209
## 1997:    20      -0.2834446             3 0.80522348 2006 169.39375 157
## 1998:    20      -0.2834446             3 0.59989167 1955  91.49007  96
## 1999:    20      -0.2834446             3 0.04148228 1996  17.13293  18
## 2000:    20      -0.2834446             3 0.77673920 2002 155.51980 152
```

5.2 Regression

For this scenario, we use the `lme4::glmer` function in R. We need to introduce a `(1|fylke)` term to identify the geographical areas (i.e. clusters).

```
d[,yearMinus2000:=year-2000]
summary(fit <- lme4::glmer(y~x + yearMinus2000 + (1|fylke),data=d,family=poisson()))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: y ~ x + yearMinus2000 + (1 | fylke)
##    Data: d
## 
##      AIC      BIC   logLik deviance df.resid 
##  15415.5  15437.9  -7703.8  15407.5     1996 
## 
## Scaled residuals: 
##     Min      1Q  Median      3Q     Max 
## -3.0448 -0.6432 -0.0067  0.6452  4.2338 
## 
## Random effects:
##  Groups Name        Variance Std.Dev.
##  fylke  (Intercept) 0.6114   0.7819  
## Number of obs: 2000, groups:  fylke, 20
## 
## Fixed effects:
##                 Estimate Std. Error z value Pr(>|z|)    
## (Intercept)    3.375e+00  1.749e-01    19.3   <2e-16 ***
## x              3.002e+00  5.994e-03   500.9   <2e-16 ***
## yearMinus2000 -9.943e-07  7.192e-05     0.0    0.989    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Correlation of Fixed Effects:
##             (Intr) x     
## x           -0.025       
## yearMns2000  0.007 -0.030
## convergence code: 0
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

You can see that the format of the results is the same as an ordinary regression.

Chapter 6

Panel data: multiple areas without autocorrelation

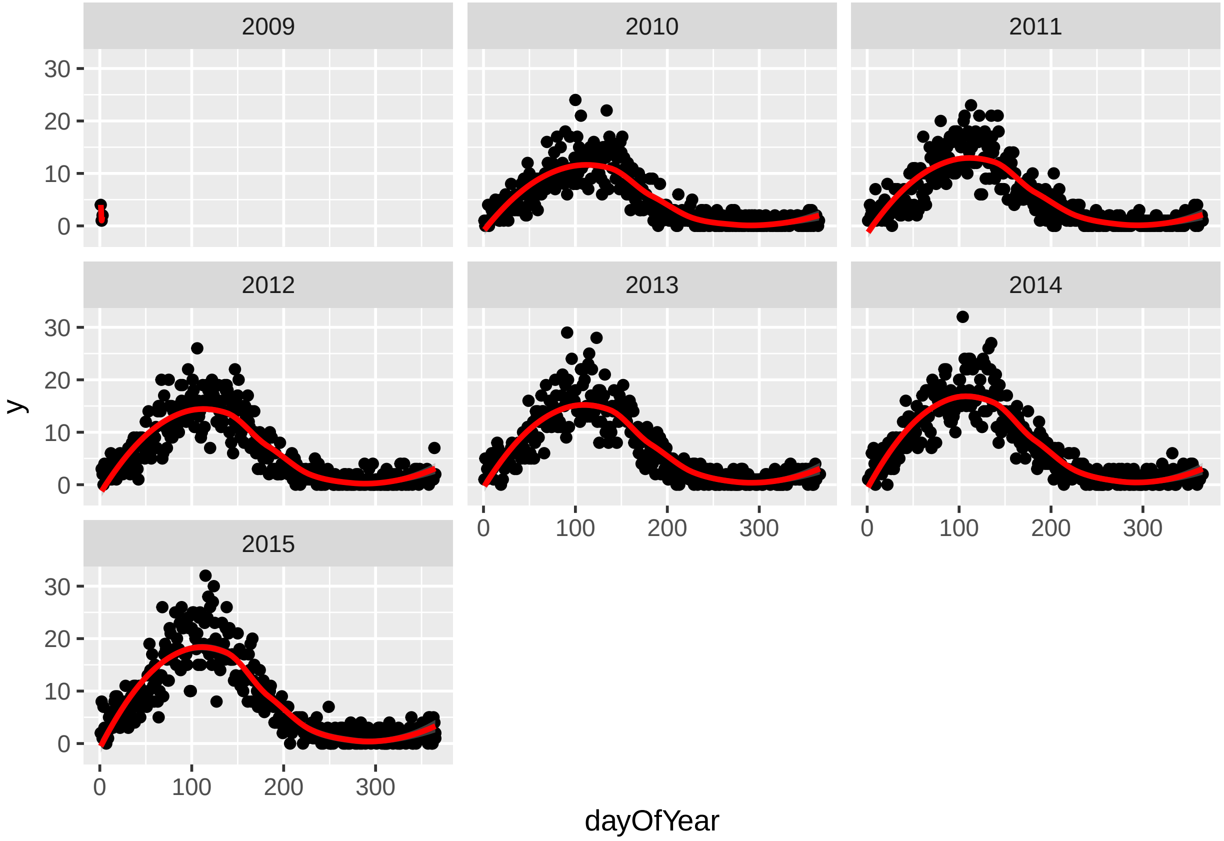
```
library(data.table)
library(ggplot2)
set.seed(4)

AMPLITUDE <- 1.5
SEASONAL_HORIZONTAL_SHIFT <- 20

fylkeIntercepts <- data.table(fylke=1:20,fylkeIntercepts=rnorm(20))

d <- data.table(date=seq.Date(
  from=as.Date("2010-01-01"),
  to=as.Date("2015-12-31"),
  by=1))
d[,year:=as.numeric(format.Date(date,"%G"))]
d[,week:=as.numeric(format.Date(date,"%V"))]
d[,month:=as.numeric(format.Date(date,"%m"))]

temp <- vector("list",length=20)
for(i in 1:20){
  temp[[i]] <- copy(d)
  temp[[i]][,fylke:=i]
}
d <- rbindlist(temp)

d[,yearMinus2000:=year-2000]
d[,dayOfSeries:=1:.N]

d[,dayOfYear:=as.numeric(format.Date(date,"%j"))]
d[,seasonalEffect:=sin(2*pi*(dayOfYear-SEASONAL_HORIZONTAL_SHIFT)/365)]
d[,mu := exp(0.1 + yearMinus2000*0.1 + seasonalEffect*AMPLITUDE)]
d[,y:=rpois(.N,mu)]
#d[,y:=round(as.numeric(arima.sim(model=list("ar"=c(0.5)), rand.gen = rpois, n=nrow(d), lambda=mu)))]
```

6.1 Investigation

We then drill down into a few years for fylke 1, and see a clear seasonal trend

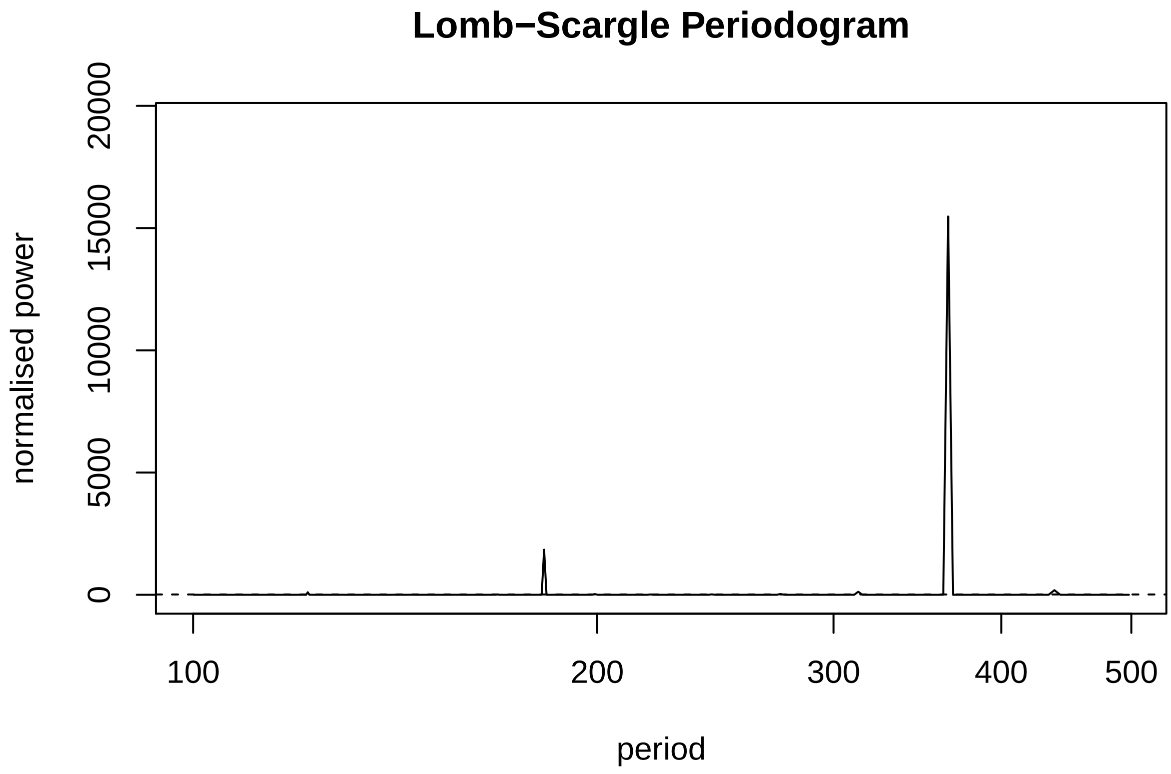
```
q <- ggplot(d[fylke==1],aes(x=dayOfYear,y=y))
q <- q + facet_wrap(~year)
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

```
## `geom_smooth()` using method = 'loess'
```



The Lomb-Scargle Periodogram shows a clear seasonality with a period of 365 days

```
lomb::lsp(d$y,from=100,to=500,ofac=1,type="period")
```



6.2 Regression

First we create an `id` variable. This generally corresponds to geographical locations, or people. In this case, we only have one geographical location, so our `id` for all observations is `1`. This lets the computer know that all data belongs to the same group.

When we have panel data with multiple areas, we use the `MASS::glmPQL` function in R.

```
d[,cos365:=cos(dayOfYear*2*pi/365)]
d[,sin365:=sin(dayOfYear*2*pi/365)]

fit <- MASS::glmmPQL(y~yearMinus2000+sin365 + cos365, random = ~ 1 | fylke,
                family = poisson, data = d)
```

```
## iteration 1
```

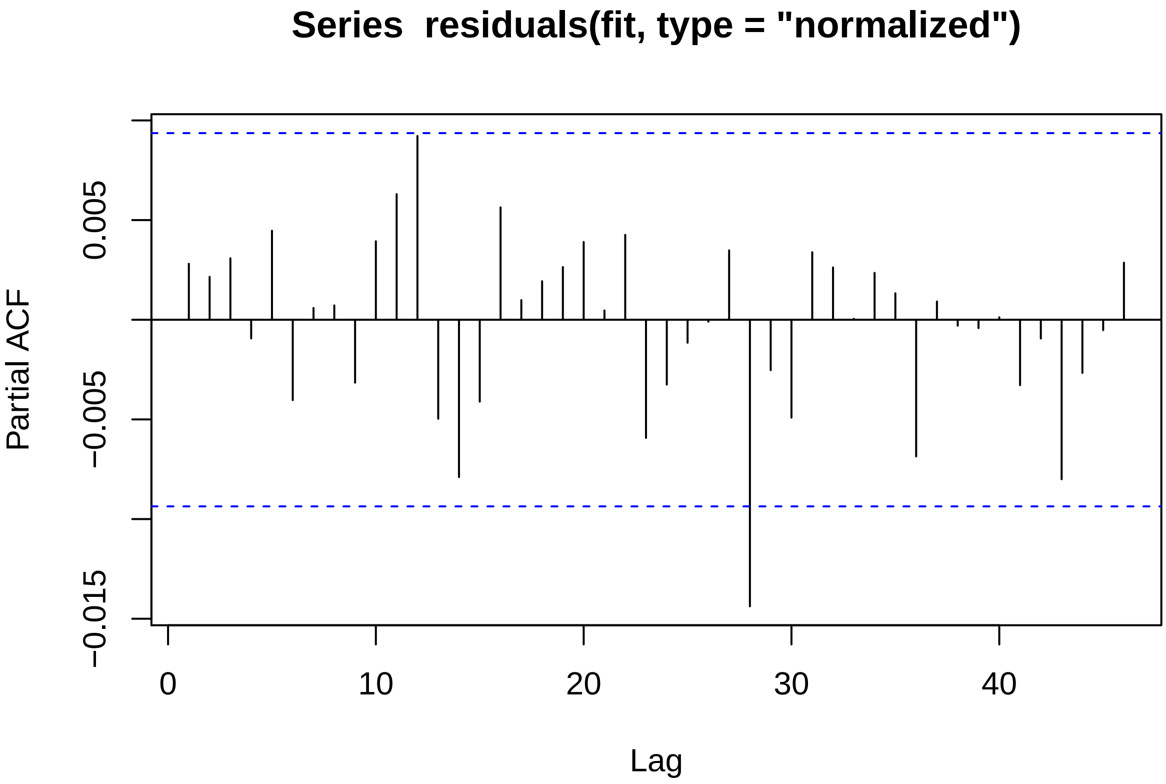
```
summary(fit)
```

```
## Linear mixed-effects model fit by maximum likelihood
##  Data: d
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | fylke
##          (Intercept)  Residual
## StdDev: 1.584549e-05 0.9976713
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: y ~ yearMinus2000 + sin365 + cos365
##                    Value   Std.Error    DF   t-value p-value
## (Intercept)    0.1122536 0.014488403 43797    7.7478       0
## yearMinus2000  0.0989047 0.001109477 43797   89.1453       0
## sin365         1.4095095 0.003695341 43797  381.4288       0
## cos365        -0.5109375 0.003083683 43797 -165.6907       0
##  Correlation:
##               (Intr) yM2000 sin365
## yearMinus2000 -0.979
## sin365        -0.150  0.000
## cos365         0.065 -0.001 -0.151
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.19682240 -0.82387498 -0.07501834  0.63400484  5.82452468
##
## Number of Observations: 43820
## Number of Groups: 20
```

6.3 Residual analysis

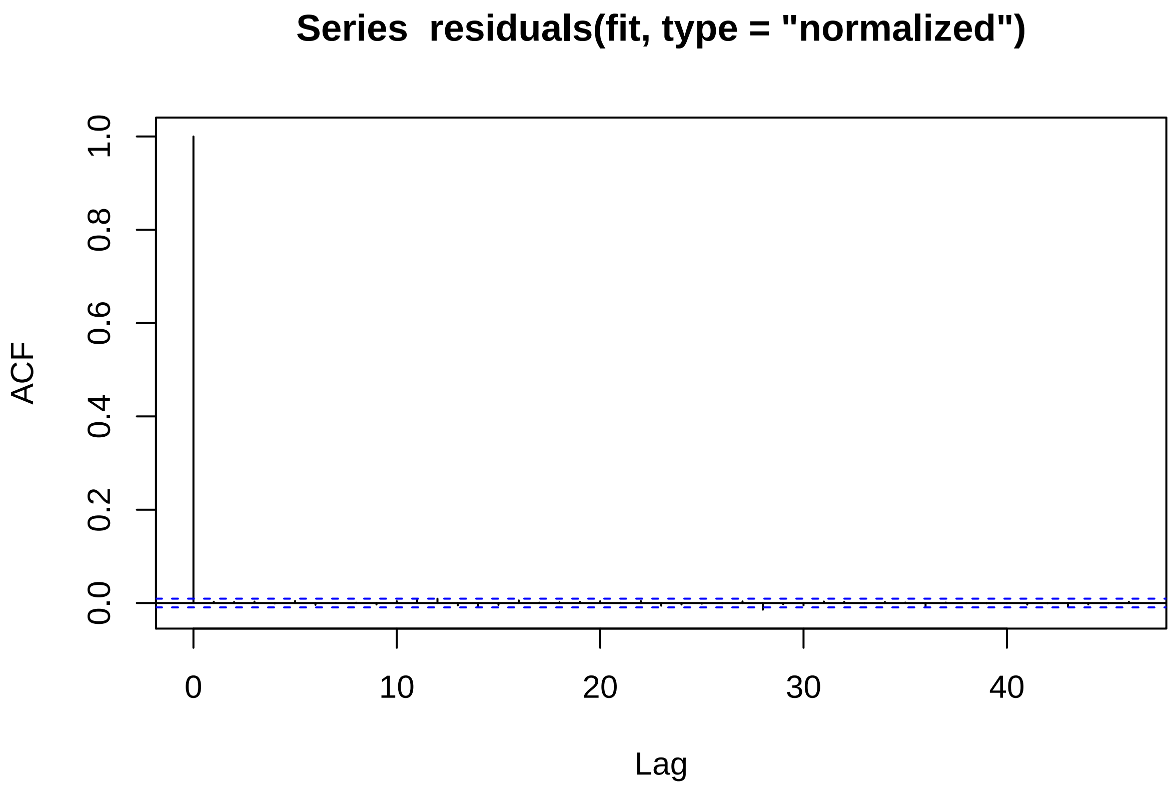
We see that there is no evidence of autoregression in the residuals

```
pacf(residuals(fit, type = "normalized")) # this is for AR
```



We see that there is no evidence of autoregression in the residuals

```
acf(residuals(fit, type = "normalized")) # this is for MA
```



We also obtain the same estimates that we did in the last chapter.

```
b1 <- 1.4007640 # sin coefficient
b2 <- -0.5234863 # cos coefficient
amplitude <- sqrt(b1^2 + b2^2)
p <- atan(b1/b2) * 365/2/pi
if (p > 0) {
    peak <- p
    trough <- p + 365/2
} else {
    peak <- p + 365/2
    trough <- p + 365
}
if (b1 < 0) {
    g <- peak
    peak <- trough
    trough <- g
}
print(sprintf("amplitude is estimated as %s, peak is estimated as %s, trough is estimated as %s",round(a
```

```
## [1] "amplitude is estimated as 1.5, peak is estimated as 112, trough is estimated as 295"
```

```
print(sprintf("true values are: amplitude: %s, peak: %s, trough: %s",round(AMPLITUDE,2),round(365/4+SEAS
```

```
## [1] "true values are: amplitude: 1.5, peak: 111, trough: 294"
```

Chapter 7

Panel data: multiple areas with autocorrelation

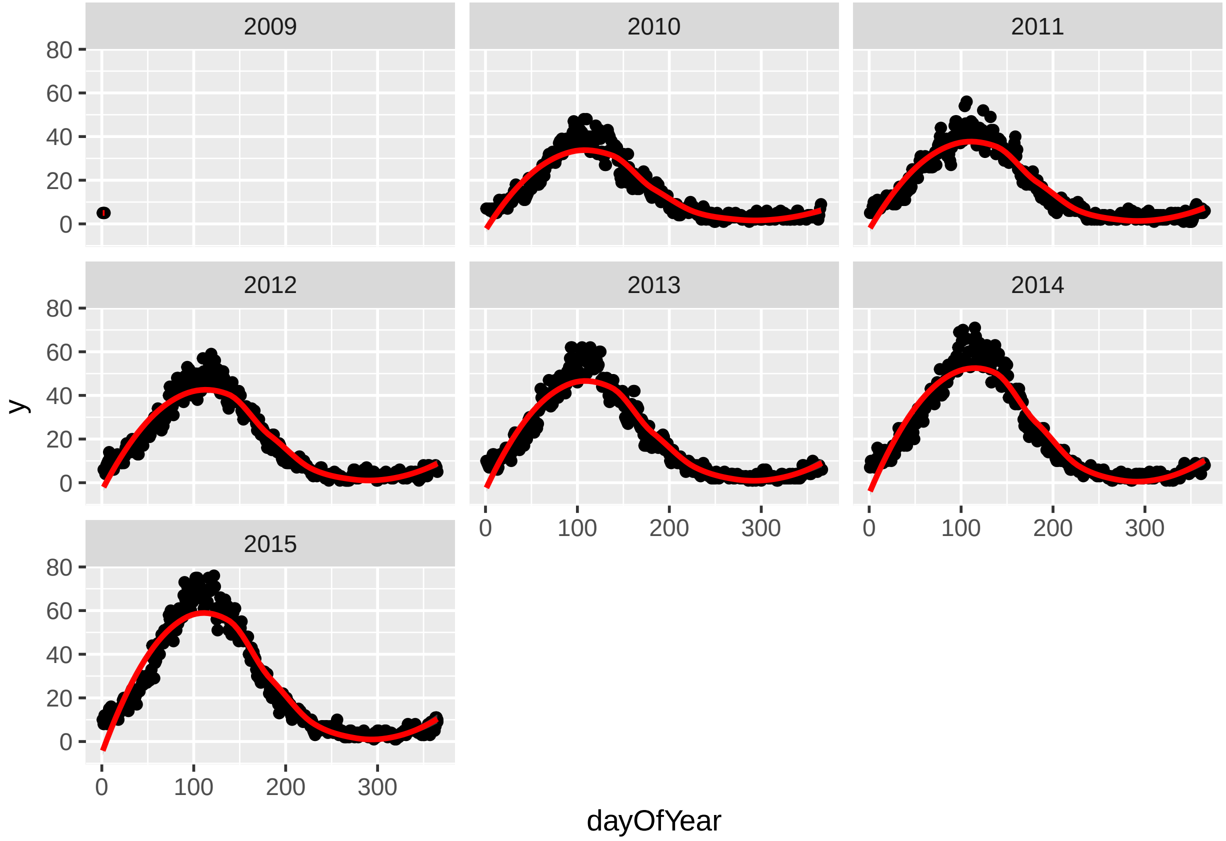
```r
library(data.table)
library(ggplot2)
set.seed(4)

AMPLITUDE <- 1.5
SEASONAL_HORIZONTAL_SHIFT <- 20

fylkeIntercepts <- data.table(fylke=1:20,fylkeIntercepts=rnorm(20))

d <- data.table(date=seq.Date(
  from=as.Date("2010-01-01"),
  to=as.Date("2015-12-31"),
  by=1))
d[,year:=as.numeric(format.Date(date,"%G"))]
d[,week:=as.numeric(format.Date(date,"%V"))]
d[,month:=as.numeric(format.Date(date,"%m"))]

temp <- vector("list",length=20)
for(i in 1:20){
  temp[[i]] <- copy(d)
  temp[[i]][,fylke:=i]
}
d <- rbindlist(temp)

d[,yearMinus2000:=year-2000]
d[,dayOfSeries:=1:.N]

d[,dayOfYear:=as.numeric(format.Date(date,"%j"))]
d[,seasonalEffect:=sin(2*pi*(dayOfYear-SEASONAL_HORIZONTAL_SHIFT)/365)]
d[,mu := round(exp(0.1 + yearMinus2000*0.1 + seasonalEffect*AMPLITUDE))]
d[,y:=rpois(.N,mu)]
d[,y:=mu+round(as.numeric(arima.sim(model=list("ar"=c(0.5)), rand.gen = rpois, n=nrow(d), lambda=mu)))]
```

7.1 Investigation

We drill down into a few years in fylke 1, and see a clear seasonal trend

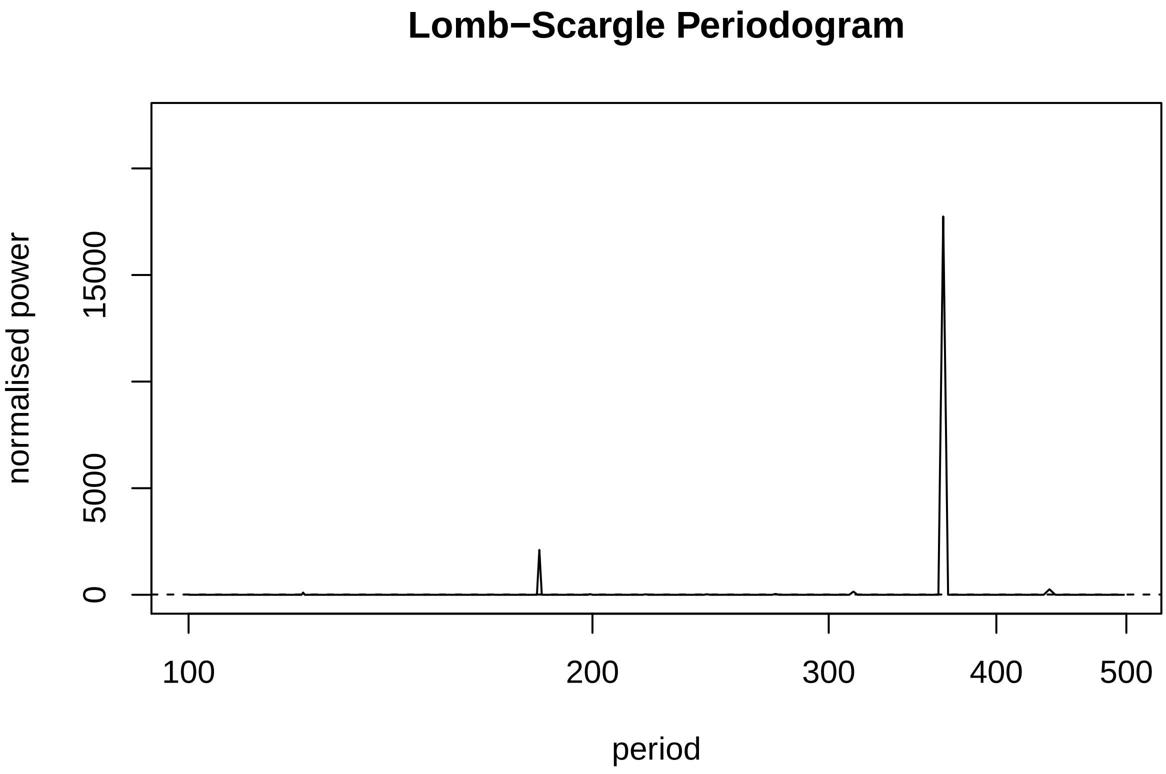
```
q <- ggplot(d[fylke==1],aes(x=dayOfYear,y=y))
q <- q + facet_wrap(~year)
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

```
## `geom_smooth()` using method = 'loess'
```



The Lomb-Scargle Periodogram shows a clear seasonality with a period of 365 days

```
lomb::lsp(d$y,from=100,to=500,ofac=1,type="period")
```



7.2 Regressions

```
d[,cos365:=cos(dayOfYear*2*pi/365)]
d[,sin365:=sin(dayOfYear*2*pi/365)]

fit <- MASS::glmmPQL(y~yearMinus2000+sin365 + cos365, random = ~ 1 | fylke,
                family = poisson, data = d)
```

```
## iteration 1
```

```
## iteration 2
```

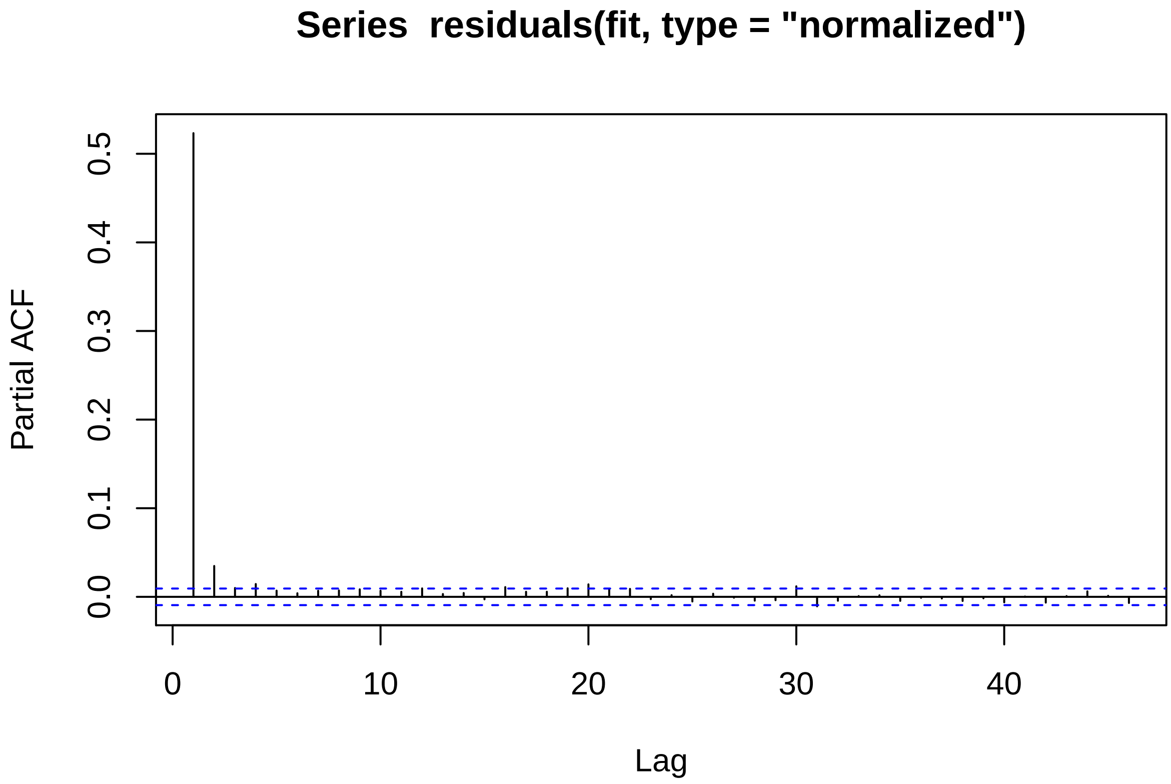
```
summary(fit)
```

```
## Linear mixed-effects model fit by maximum likelihood
##  Data: d
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | fylke
##         (Intercept)  Residual
## StdDev: 0.004579768 0.7191519
##
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: y ~ yearMinus2000 + sin365 + cos365
##                    Value   Std.Error    DF   t-value p-value
## (Intercept)    1.2189925 0.006110555 43797  199.4896       0
## yearMinus2000  0.0987374 0.000461394 43797  213.9980       0
## sin365         1.3990267 0.001531179 43797  913.6928       0
## cos365        -0.5171211 0.001282191 43797 -403.3106       0
##  Correlation:
##               (Intr) yM2000 sin365
## yearMinus2000 -0.966
## sin365        -0.147  0.000
## cos365         0.065 -0.001 -0.152
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.00864057 -0.70228031 -0.06334676  0.64274011  5.21710225
##
## Number of Observations: 43820
## Number of Groups: 20
```

7.3 Residual analysis

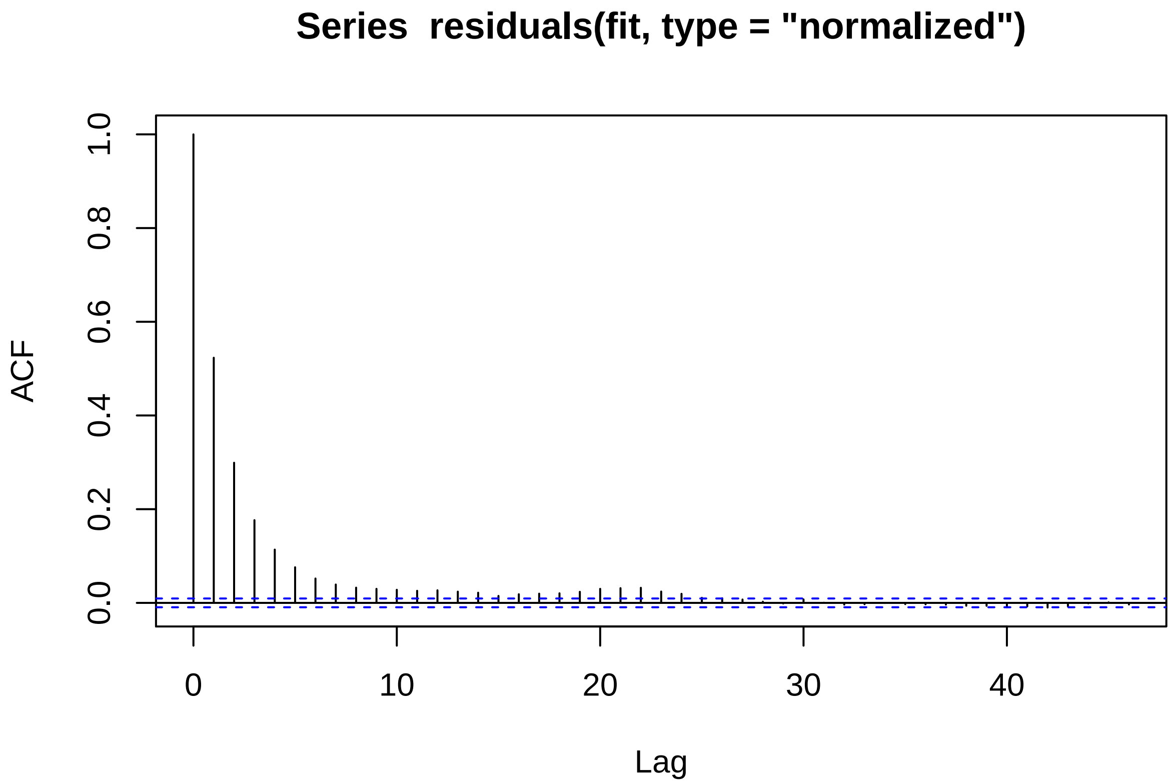
We see that there is an AR(1) autocorrelation in the residuals, meaning that our model is not appropriate.

```
pacf(residuals(fit, type = "normalized")) # this is for AR
```



We see that there is some sort of `AR` autocorrelation in the residuals, meaning that our model is not appropriate.

```
acf(residuals(fit, type = "normalized")) # this is for MA
```



7.4 Regression with AR(1) correlation in residuals

We include `correlation=nlme::corAR1(form=~dayOfSeries|fylke)` or in other words `correlation=nlme::corAR1(form=`
to let the computer know what is the time variable and what is the group variable.

```
fit <- MASS::glmmPQL(y~yearMinus2000+sin365 + cos365, random = ~ 1 | fylke,
                family = poisson, data = d,
                correlation=nlme::corAR1(form=~dayOfSeries|fylke))
```

```
## iteration 1
```

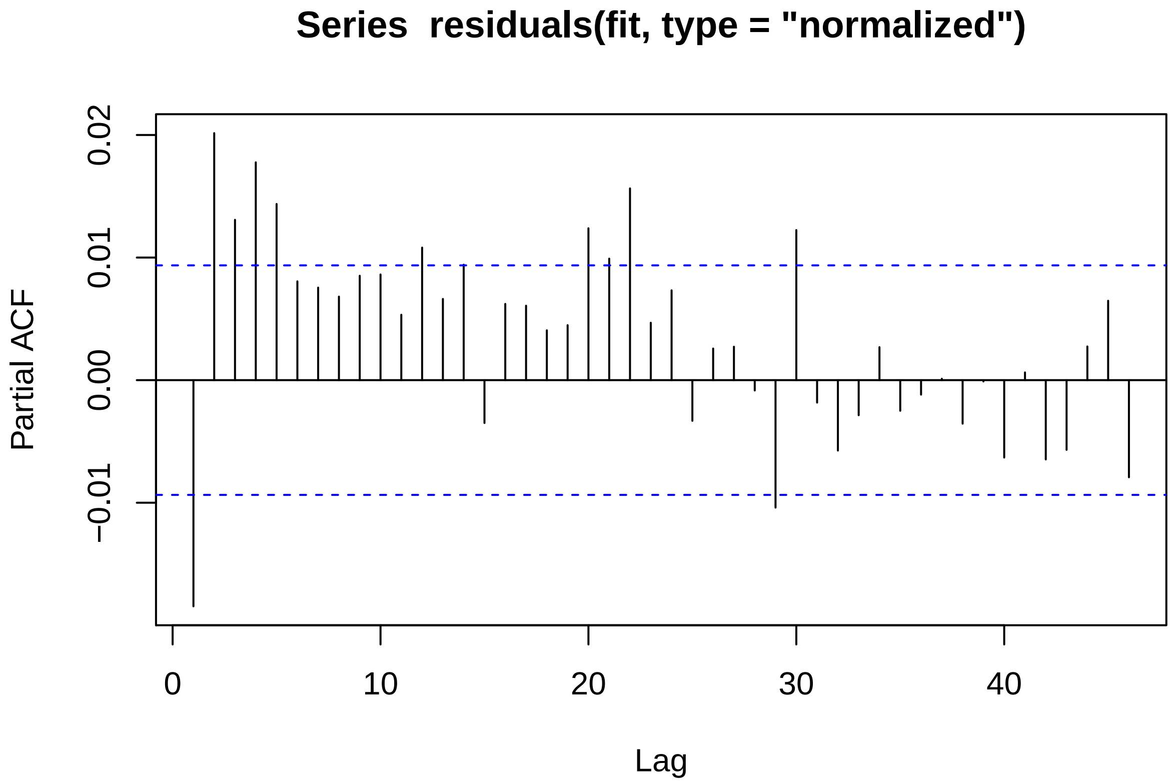
```
summary(fit)
```

```
## Linear mixed-effects model fit by maximum likelihood
##  Data: d
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | fylke
##         (Intercept)  Residual
## StdDev: 2.405003e-05 0.7195239
##
## Correlation Structure: AR(1)
##  Formula: ~dayOfSeries | fylke
##  Parameter estimate(s):
##       Phi
## 0.5240054
## Variance function:
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: y ~ yearMinus2000 + sin365 + cos365
##                    Value   Std.Error    DF   t-value p-value
## (Intercept)    1.2195477 0.010774796 43797  113.1852       0
## yearMinus2000  0.0987065 0.000825226 43797  119.6115       0
## sin365         1.3988945 0.002739109 43797  510.7116       0
## cos365        -0.5169579 0.002292465 43797 -225.5030       0
##  Correlation:
##               (Intr) yM2000 sin365
## yearMinus2000 -0.979
## sin365        -0.149  0.001
## cos365         0.066 -0.001 -0.151
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -2.99731654 -0.70249782 -0.06736726  0.64264790  5.20296607
##
## Number of Observations: 43820
## Number of Groups: 20
```

7.5 Residual analysis

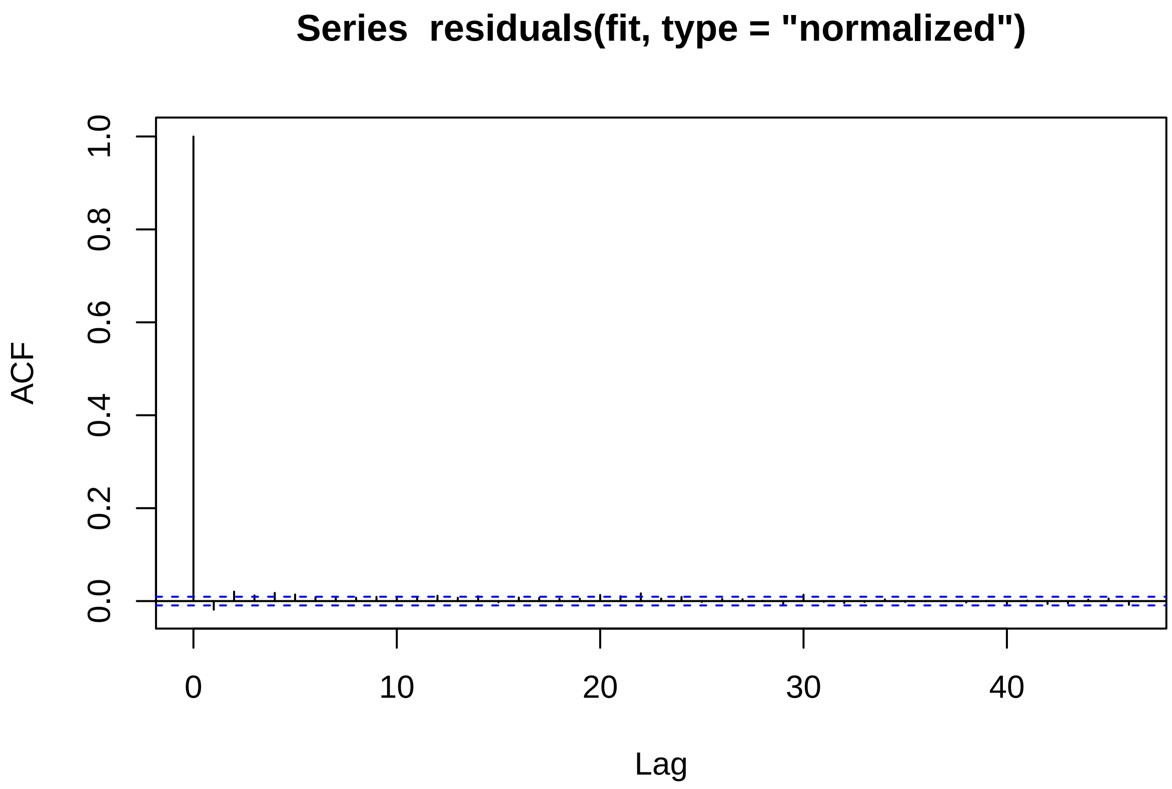
We see that the vast majority of the autoregression in the residuals has been removed.

```
pacf(residuals(fit, type = "normalized")) # this is for AR
```



We see that the vast majority of the autoregression in the residuals has been removed.

```r
acf(residuals(fit, type = "normalized")) # this is for MA
```



We obtain the same estimates that we did in the last chapter.

```
b1 <- 1.4007640 # sin coefficient
b2 <- -0.5234863 # cos coefficient
amplitude <- sqrt(b1^2 + b2^2)
p <- atan(b1/b2) * 365/2/pi
if (p > 0) {
    peak <- p
    trough <- p + 365/2
} else {
    peak <- p + 365/2
    trough <- p + 365
}
if (b1 < 0) {
    g <- peak
    peak <- trough
    trough <- g
}
print(sprintf("amplitude is estimated as %s, peak is estimated as %s, trough is estimated as %s",round(a
```

```
## [1] "amplitude is estimated as 1.5, peak is estimated as 112, trough is estimated as 295"
```

```
print(sprintf("true values are: amplitude: %s, peak: %s, trough: %s",round(AMPLITUDE,2),round(365/4+SEAS
```

```
## [1] "true values are: amplitude: 1.5, peak: 111, trough: 294"
```