Longitudinal Analysis

Richard White

2018-04-12

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 “sometime” 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

2.2 Identifying your scenario

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.

Step 2: Do you have multiple geographical areas?

Again, fairly simple, just look at your data.

Step 3: Do you have autocorrelation?

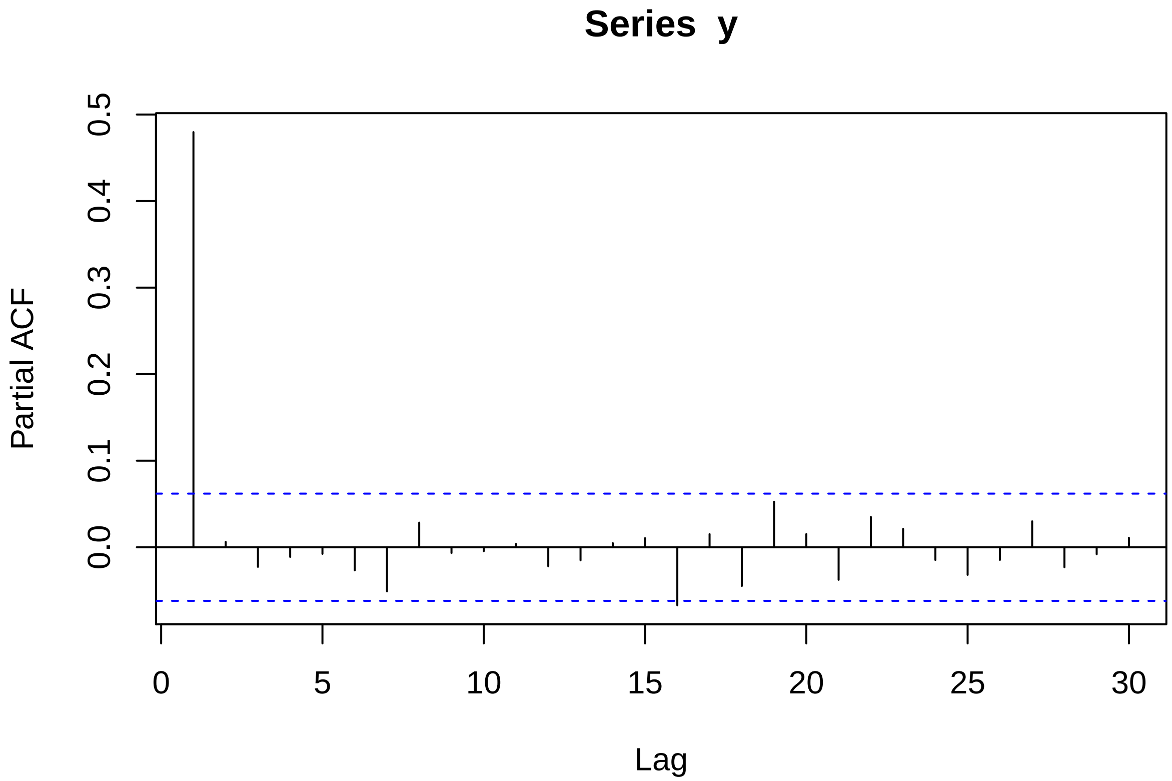
Firstly, you must run a model pretending that you do not have autocorrelation. You then inspect the residuals from the model and see if autocorrelation exists. This is done with two statistical procedures: `pacf` (for `autoregressive models`, the most common type of autocorrelation), and `acf` (for `moving average models`, a less common type of autocorrelation).

2.2.1 AR(1) data

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

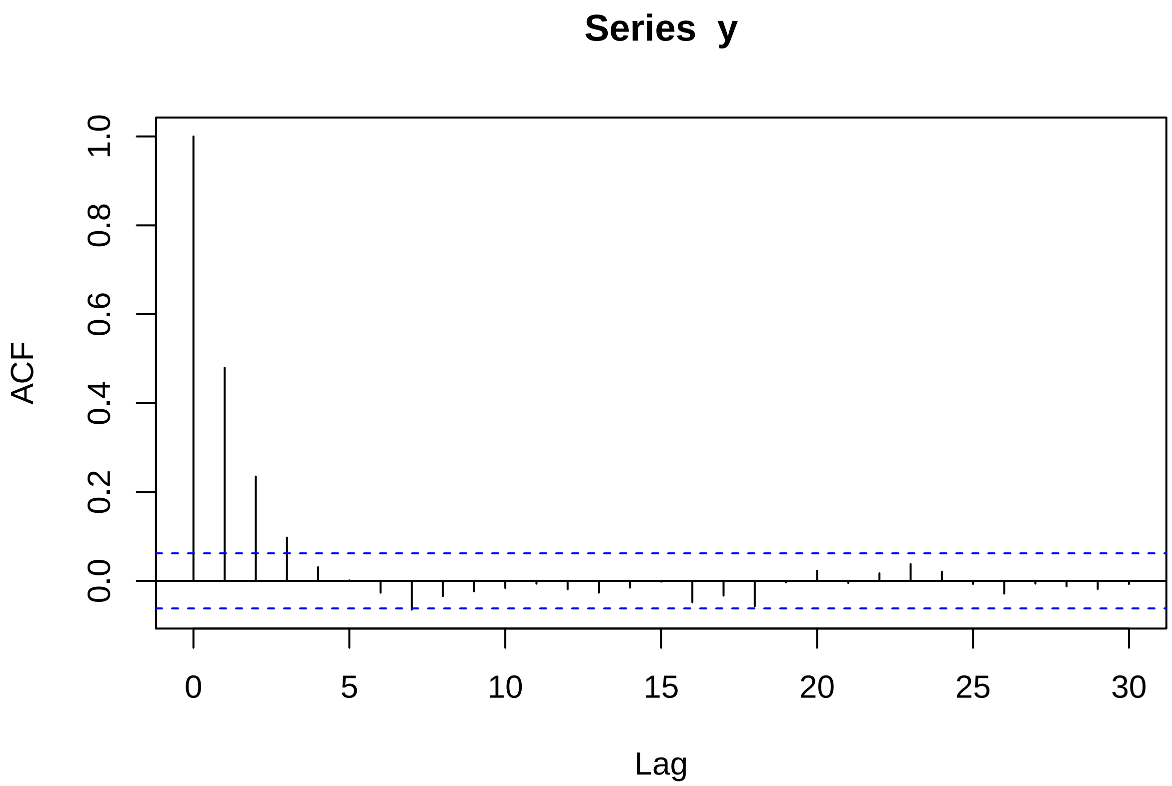
With `autoregressive` data, a `pacf` plot contains a number of sharp significant lines, demarking how many subsequent observations have autocorrelation. i.e. if one line is significant, it means that each observation is only correlated with its preceeding observation. If two lines are significant, it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

```
pacf(y)
```



With `autoregressive` data, an `acf` plot contains a number of decreasing lines. The following 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)
```

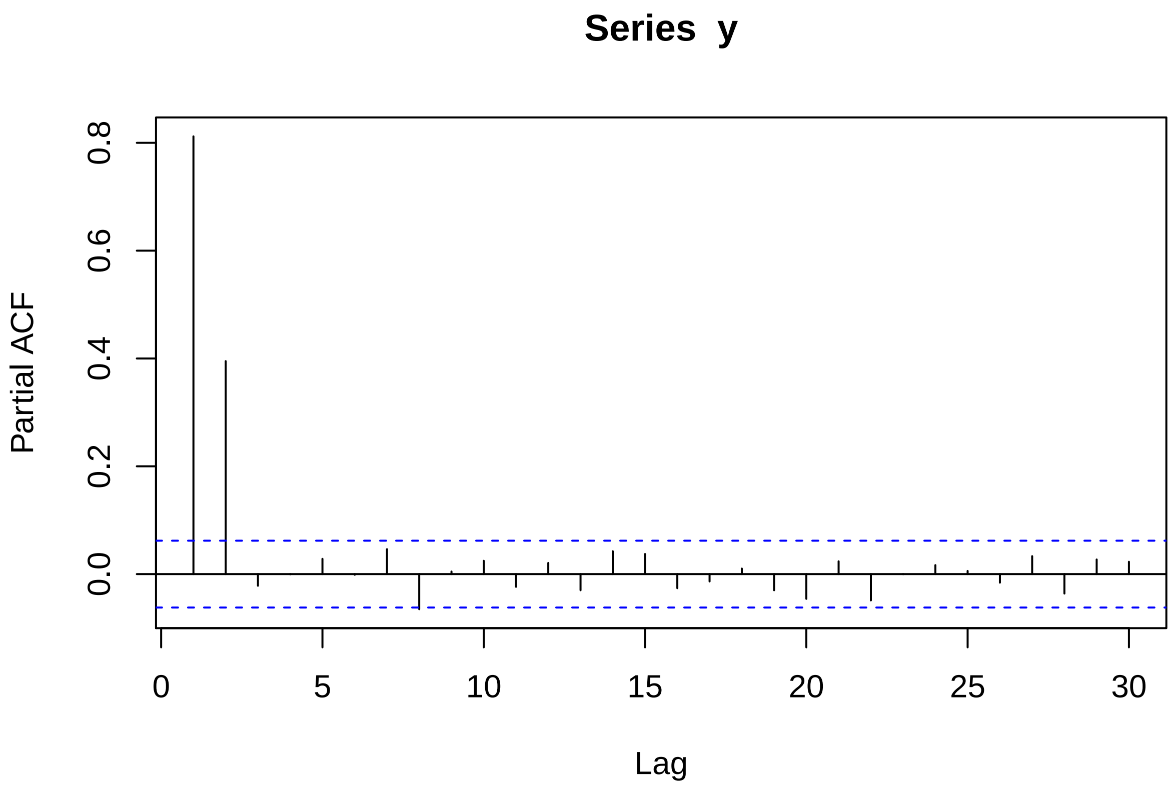


2.2.2 AR(2) data

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

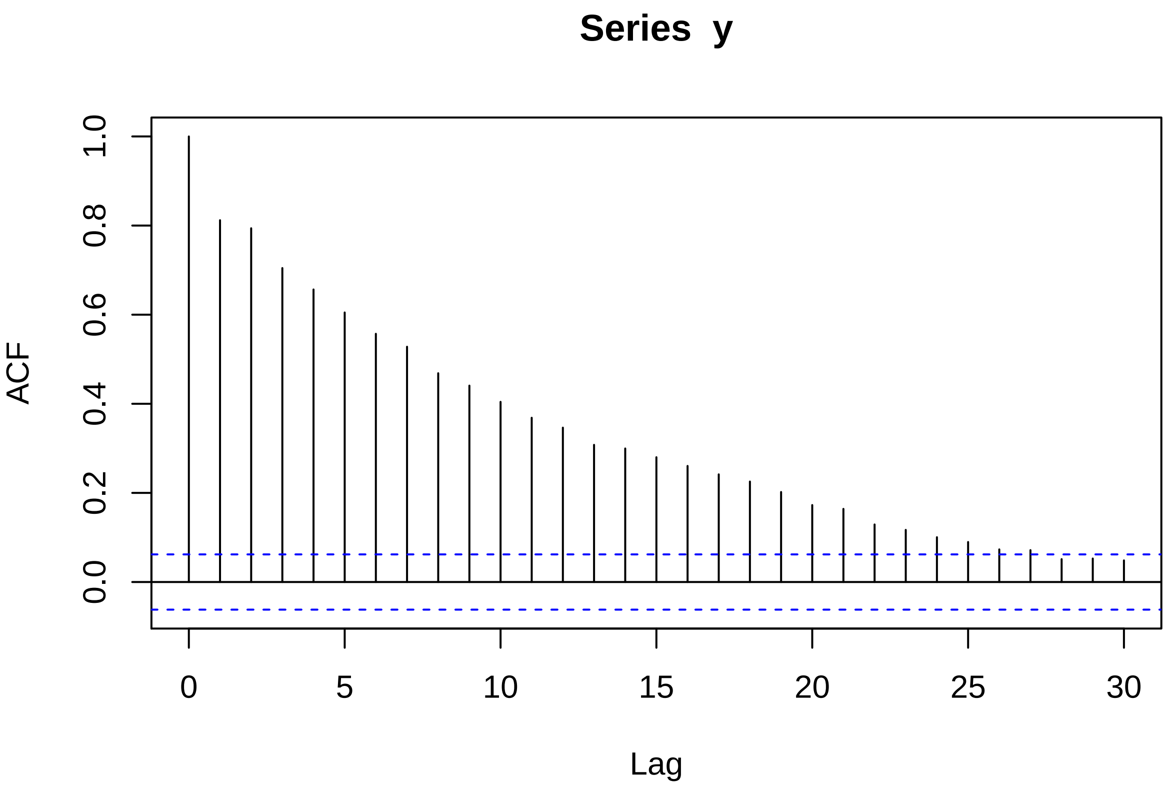
With `autoregressive` data, a `pacf` plot contains a number of sharp significant lines, demarking how many subsequent observations have autocorrelation. i.e. if one line is significant, it means that each observation is only correlated with its preceeding observation. If two lines are significant, it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

```
pacf(y)
```



With `autoregressive` data, an `acf` plot contains a number of decreasing lines. The following 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)
```

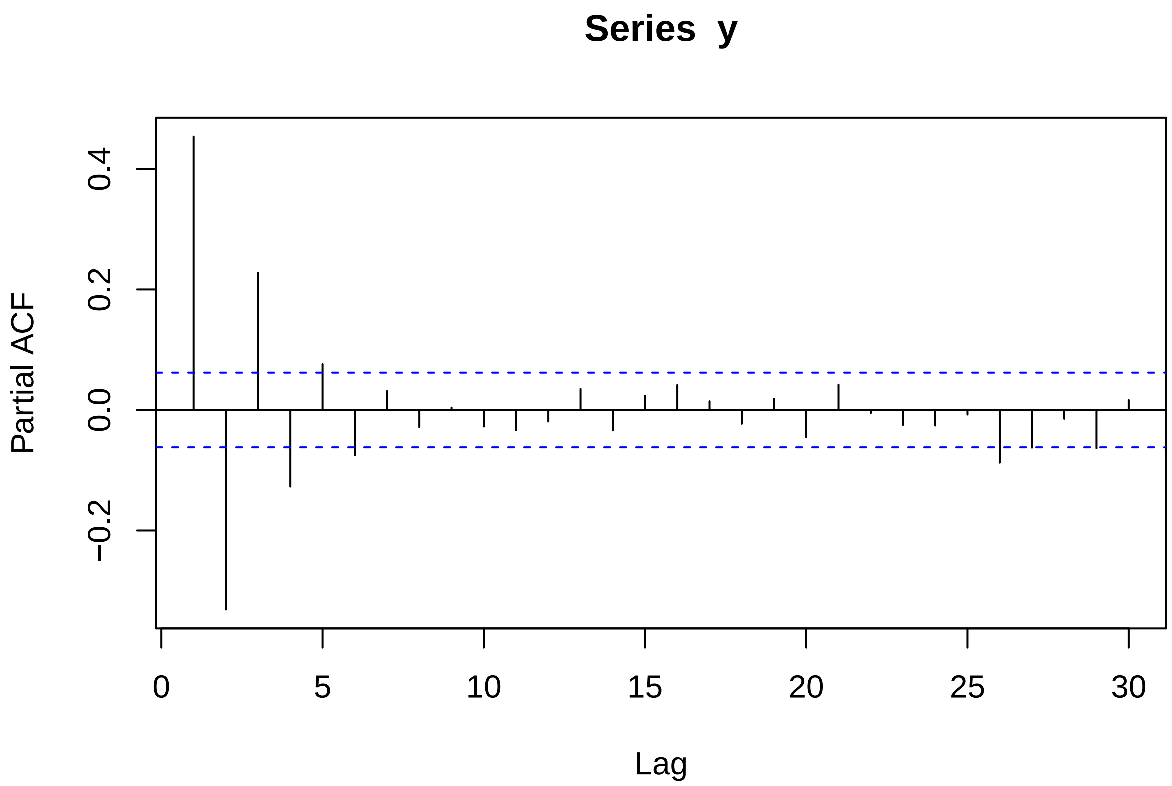


2.2.3 MA(1) data

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

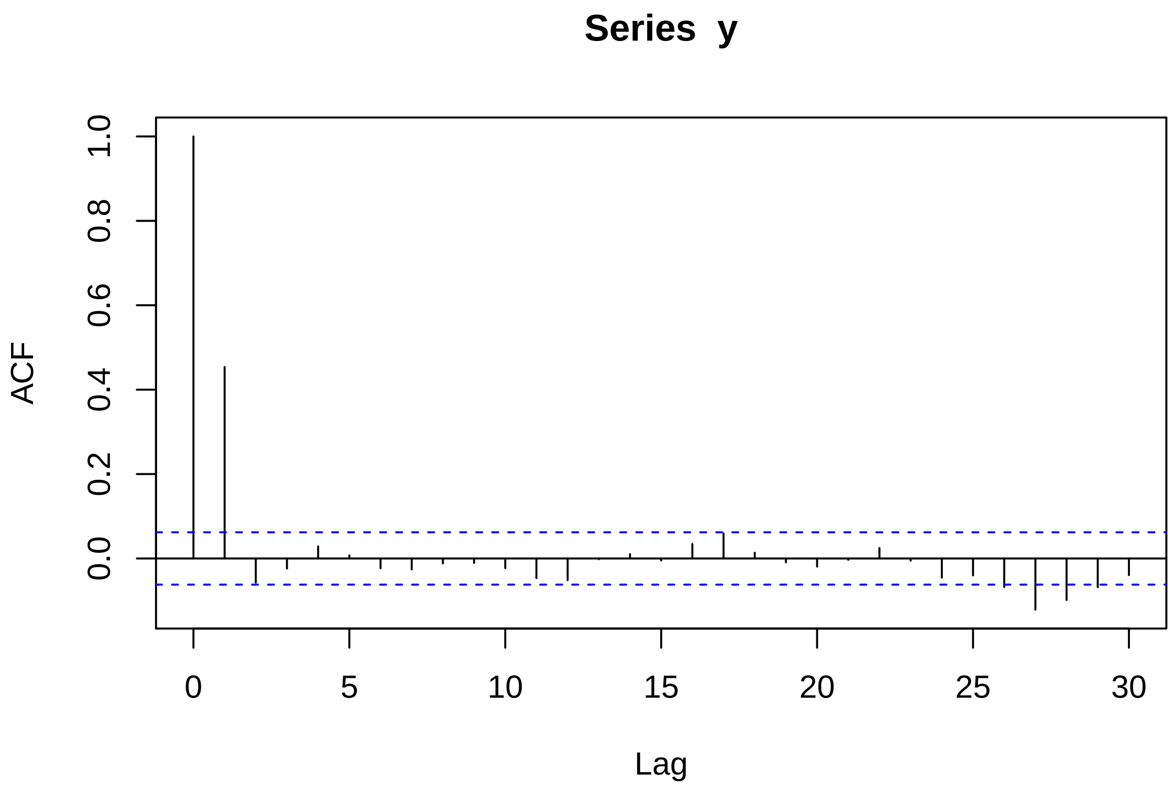
With `moving average` data, a `pacf` plot contains a number of decreasing lines. The following plot represents some sort of `MA` data.

```
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)
```

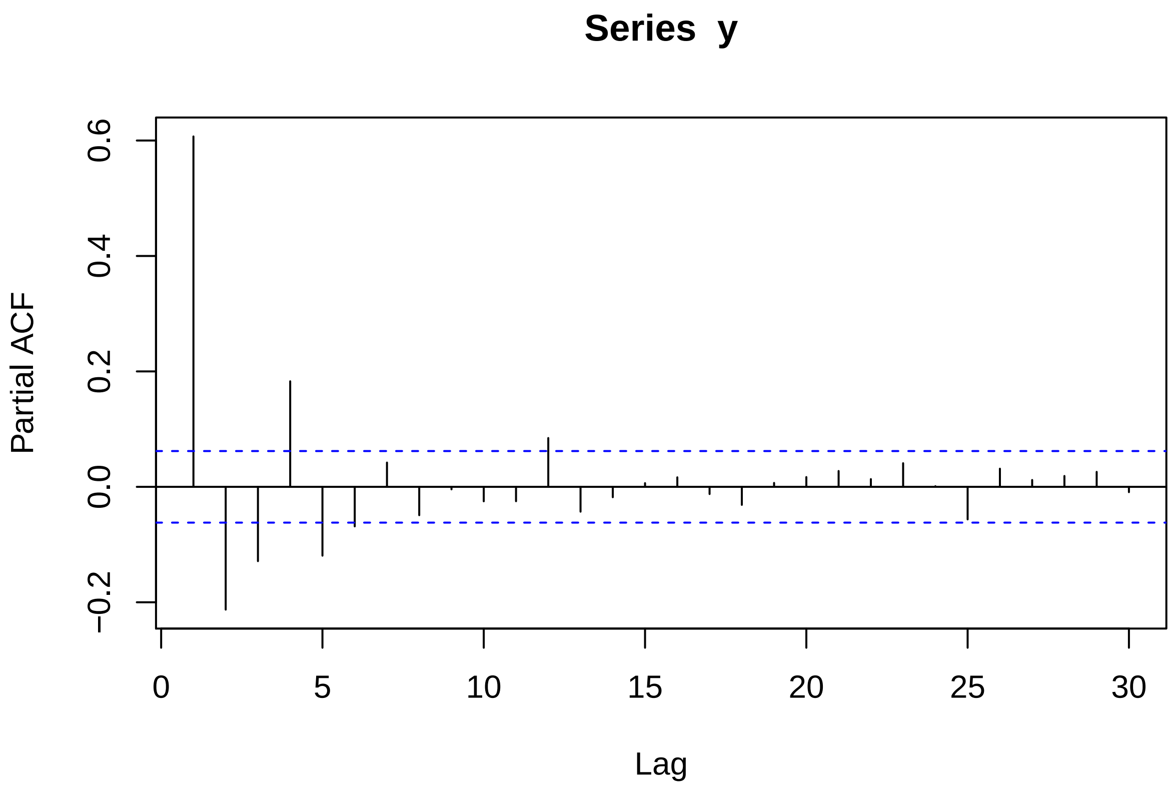


2.2.4 MA(1) data

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

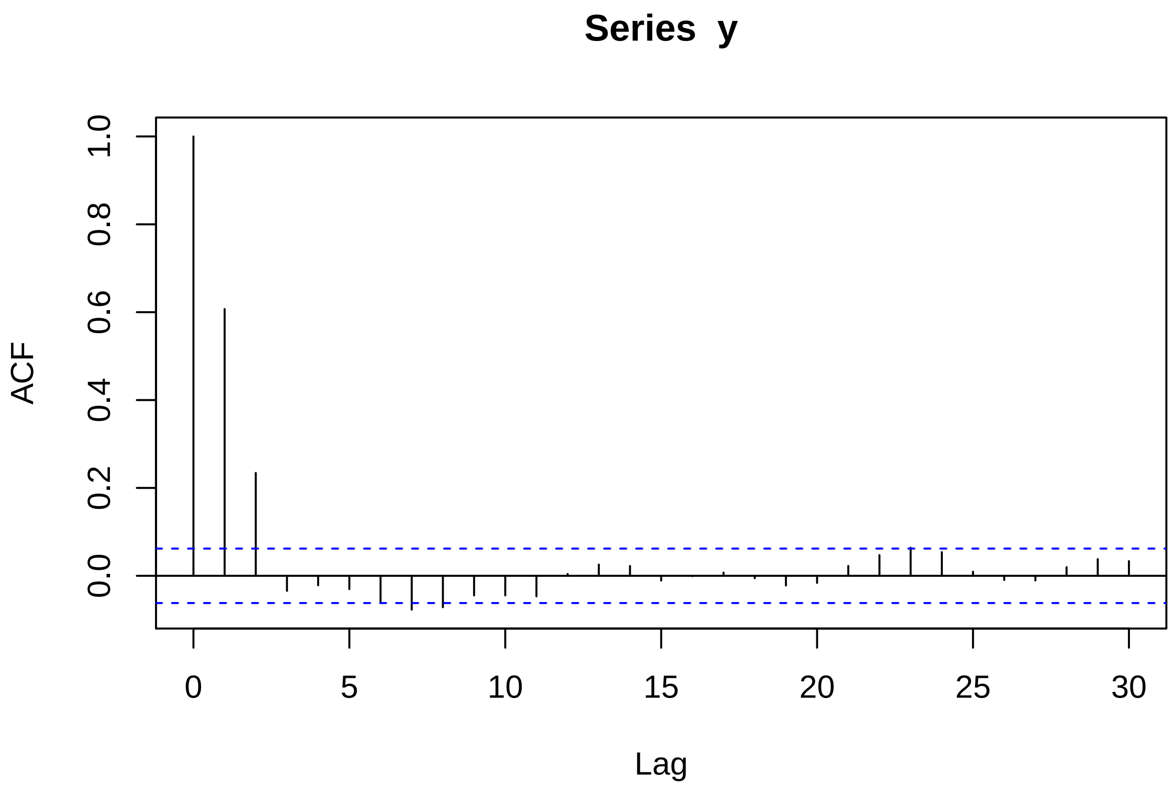
With `moving average` data, a `pacf` plot contains a number of decreasing lines. The following plot represents some sort of `MA` data.

```
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(2)` data. Note that the `acf` plot displays `lag 0` (which is pointless and can be ignored), while the `pacf` plot does not.

```
acf(y)
```


Series y
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30
Lag

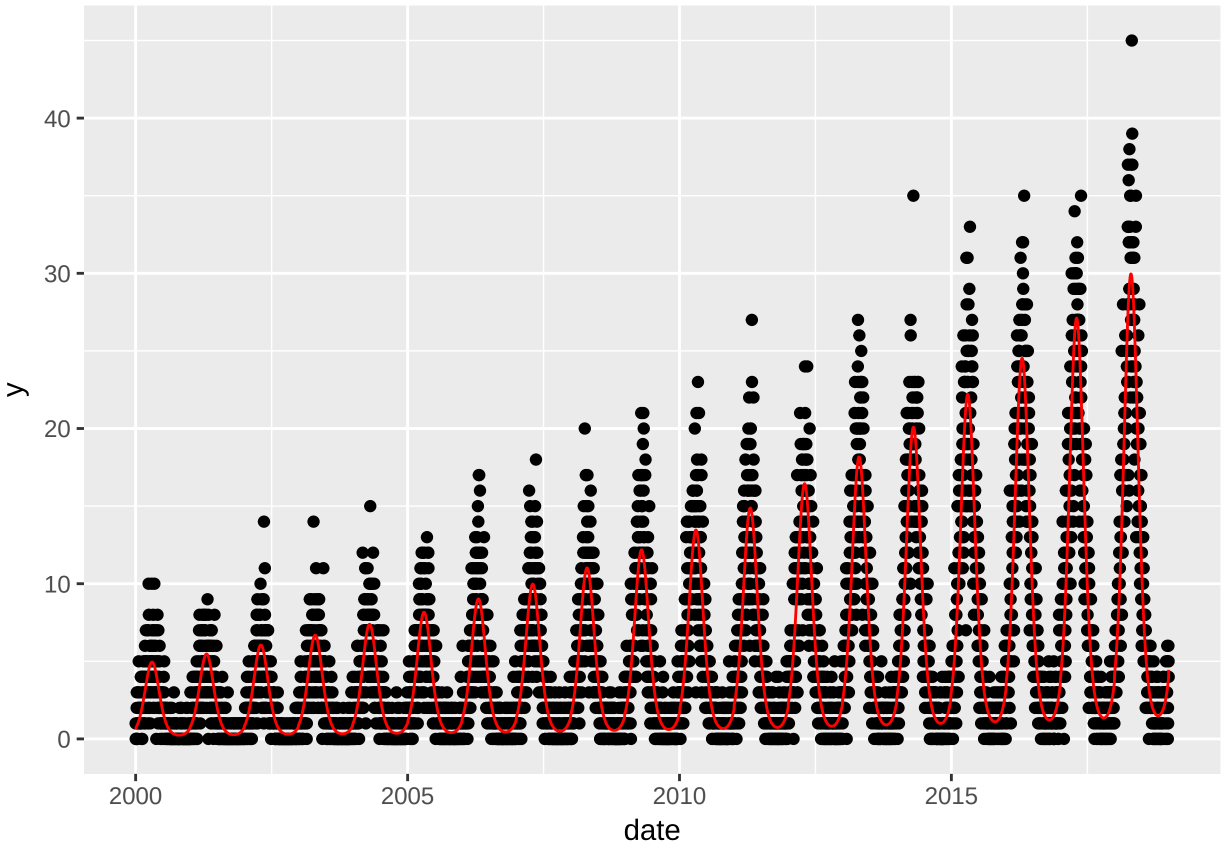
Chapter 3

Panel data: One area without 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[,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)]
```

Showing the true data

```
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
```

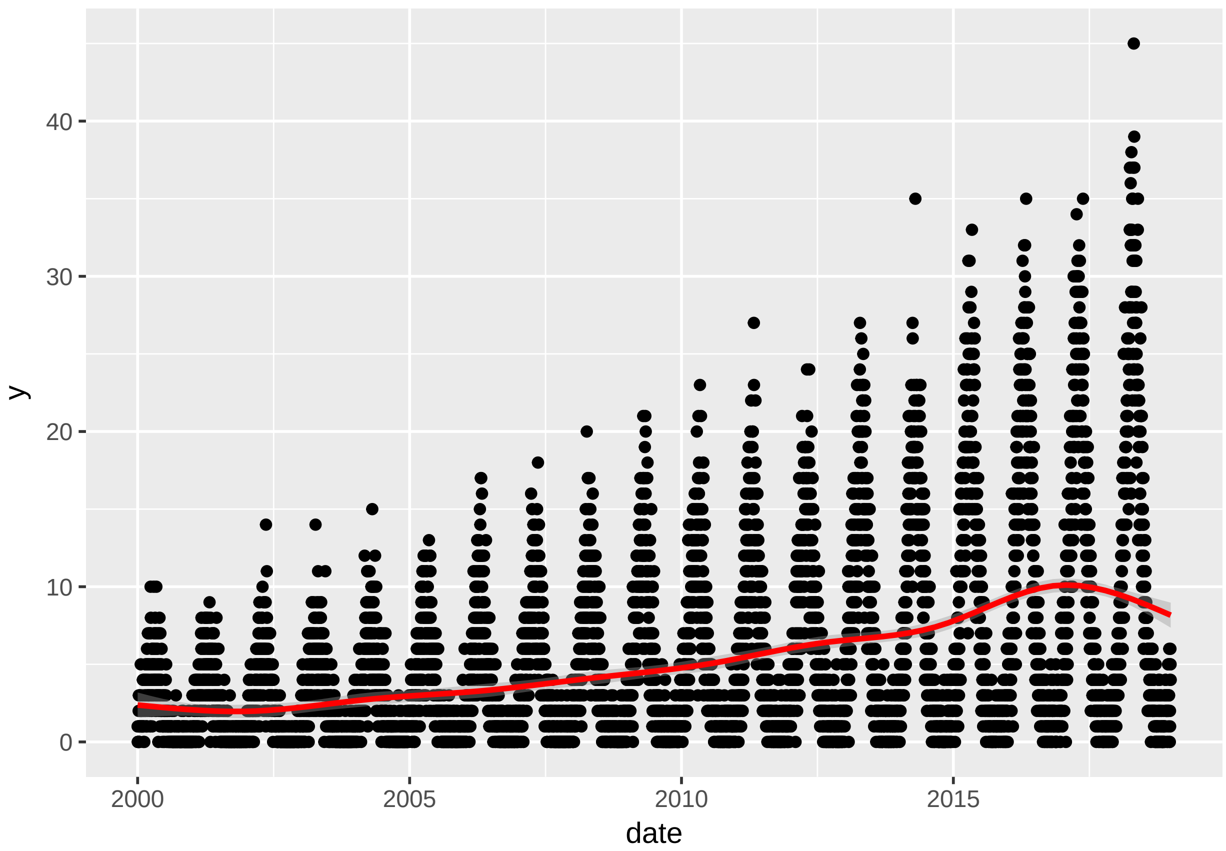


Investigating the data

We take a quick look, but don't see much

```r
q <- ggplot(d,aes(x=date,y=y))
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

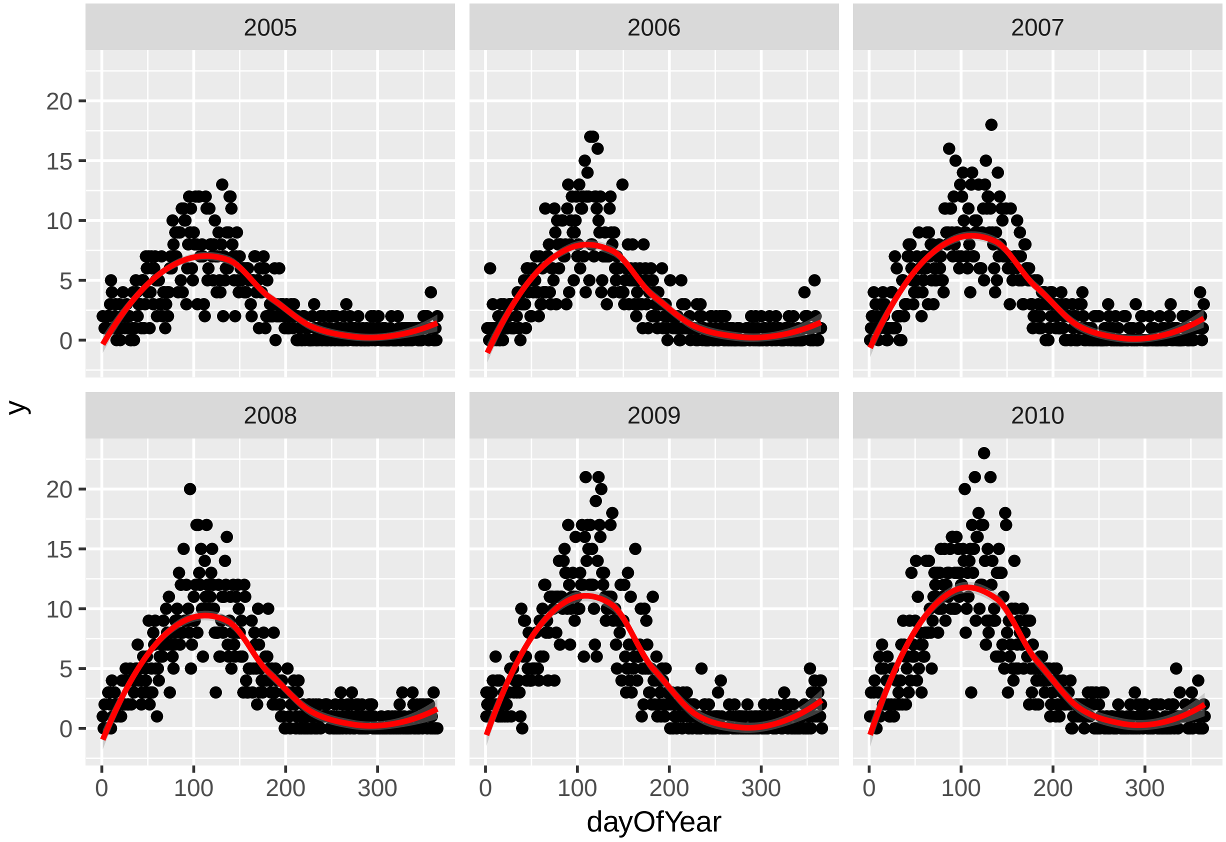
```
## `geom_smooth()` using method = 'gam'
```



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

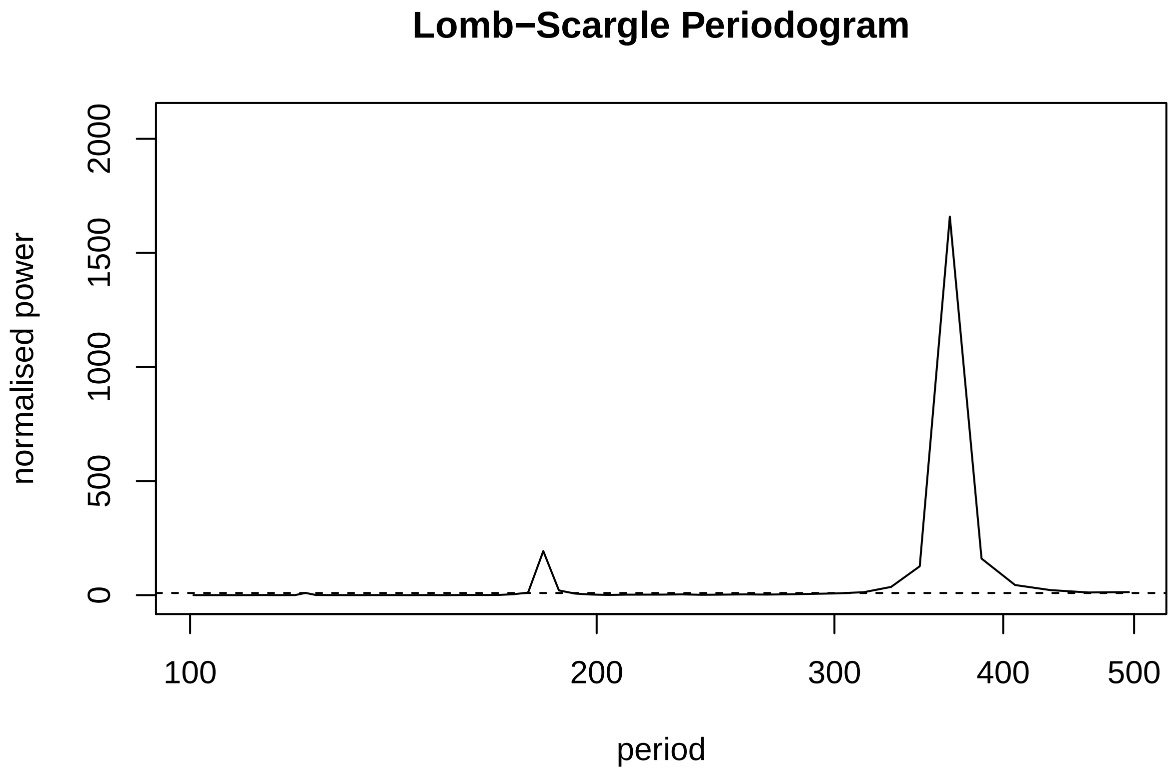
```
q <- ggplot(d[year %in% c(2005:2010)],aes(x=dayOfYear,y=y))
q <- q + facet_wrap(~year)
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

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



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

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



We then generate two new variables cos365 and sin365 and perform a simple poisson regression:

```
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
```

```
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 see a clear significant seasonal effect. We can then use trigonometry to back-calculate the `cos365` and `sin365` variables to amplitude and location of peak/troughs:

```
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(a
```

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

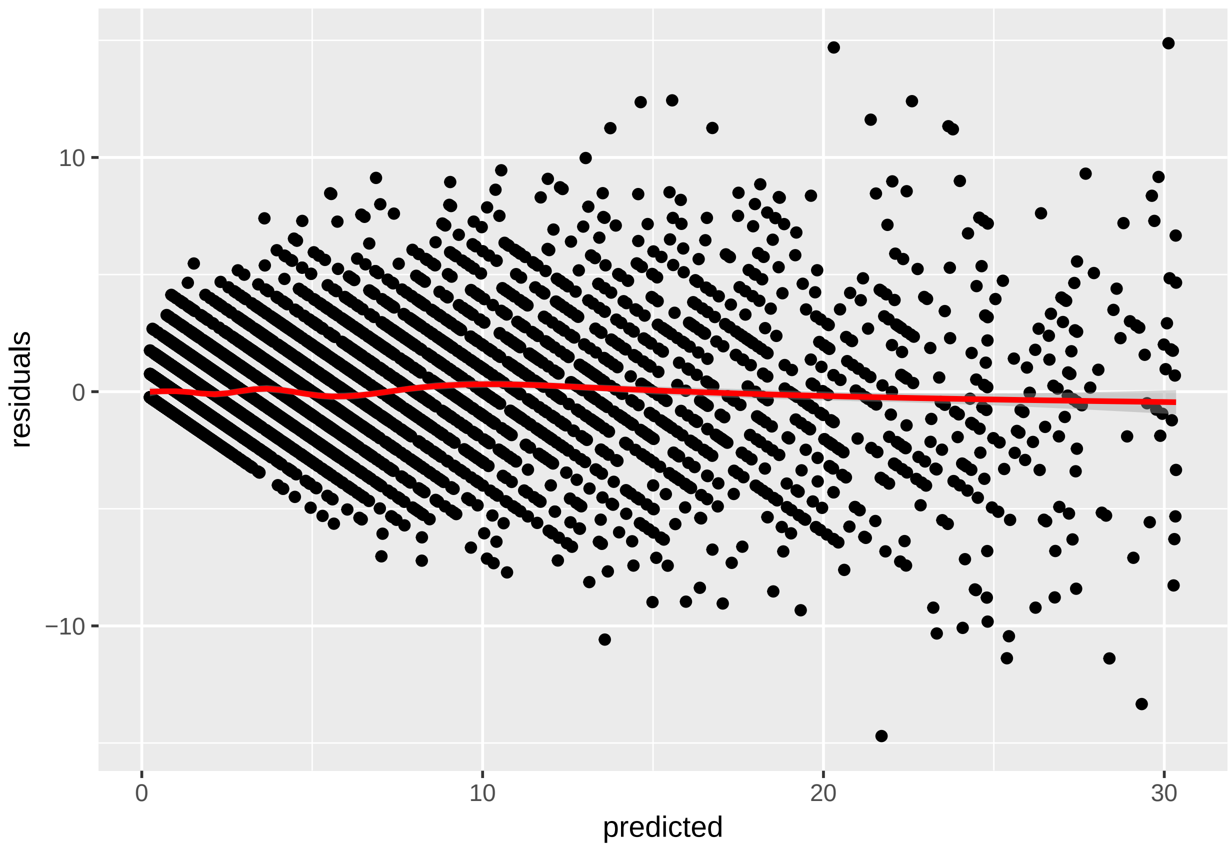
```
print(sprintf("true values are: amplitude: %s, peak: %s, trough: %s",round(AMPLITUDE,2),round(365/4+SEAS
```

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

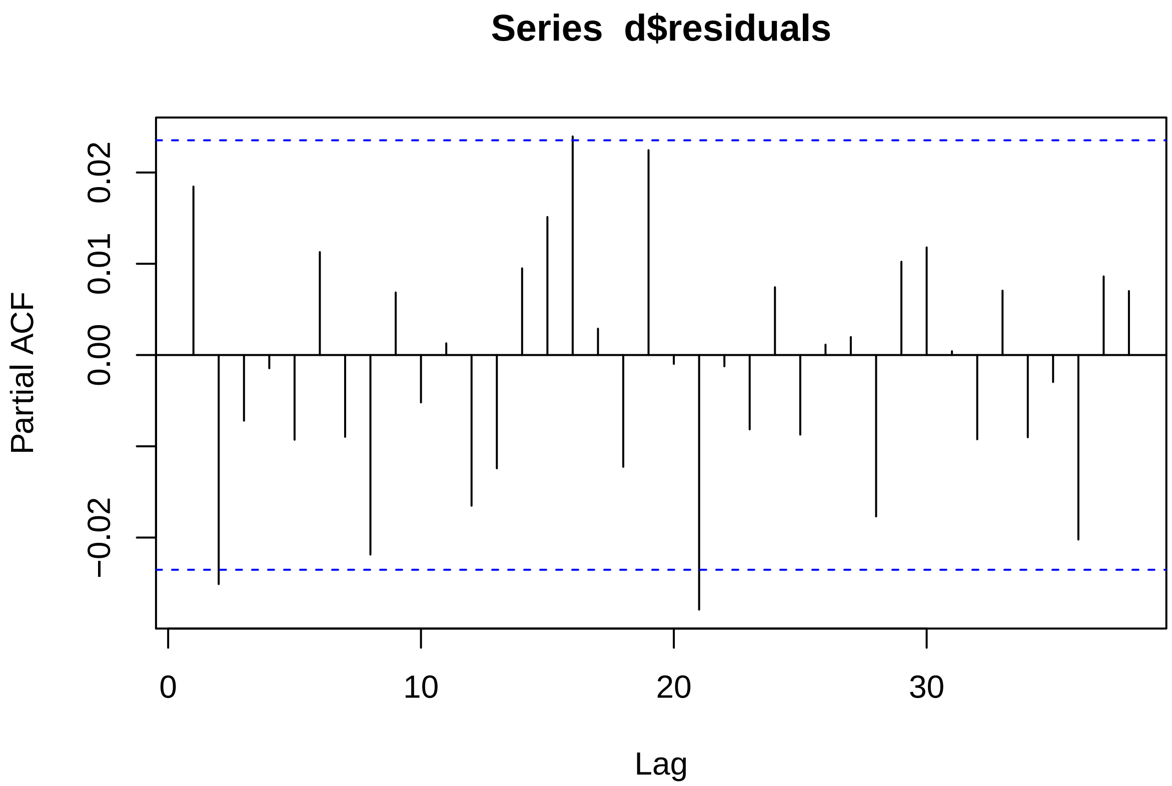
We now investigate our residuals to determine if we have a good fit:

```
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
```

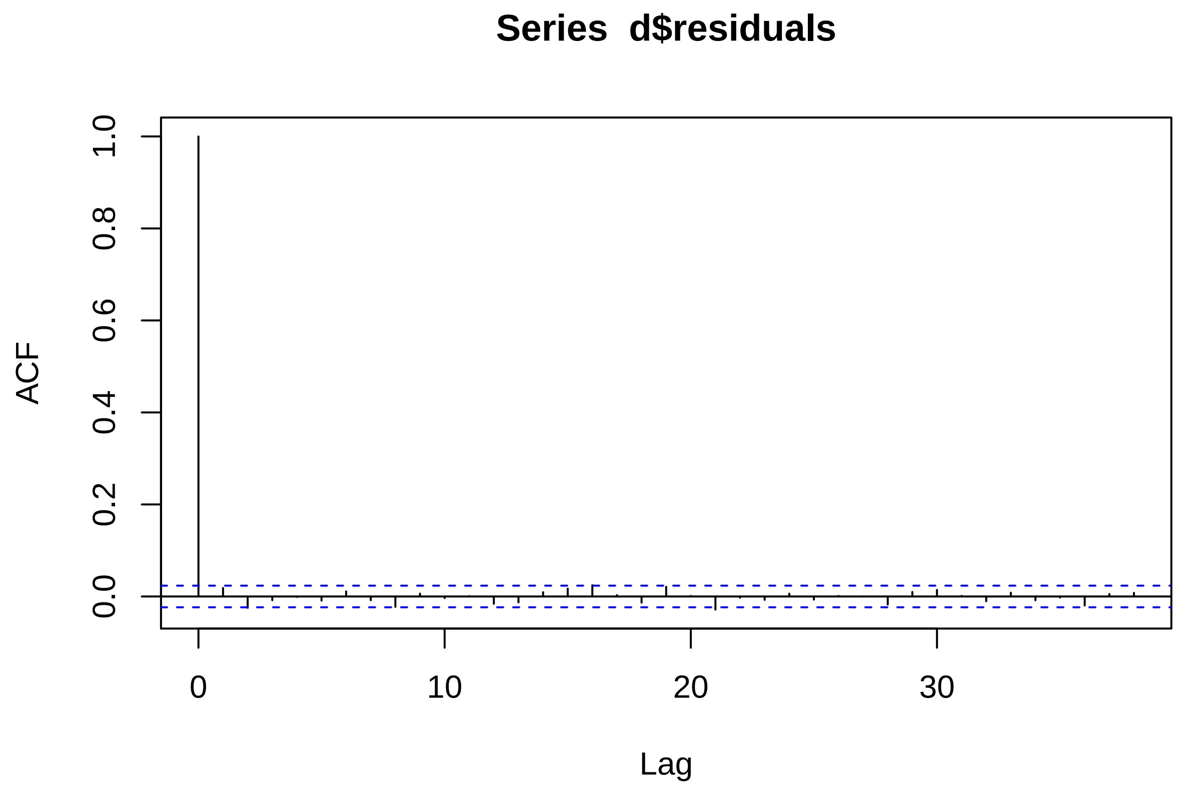
```
## `geom_smooth()` using method = 'gam'
```



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



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



We see a clear significant seasonal effect. We can then use trigonometry to back-calculate the `cos365` and `sin365` variables to amplitude and location of peak/troughs:

```
b1 <- 0.1934 # sin coefficient
b2 <- 0.1018 # 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 %s, peak is at %s, trough is at %s",round(amplitude,2),round(peak),round(tr
```

```
## [1] "amplitude is 0.22, peak is at 63, trough is at 246"
```

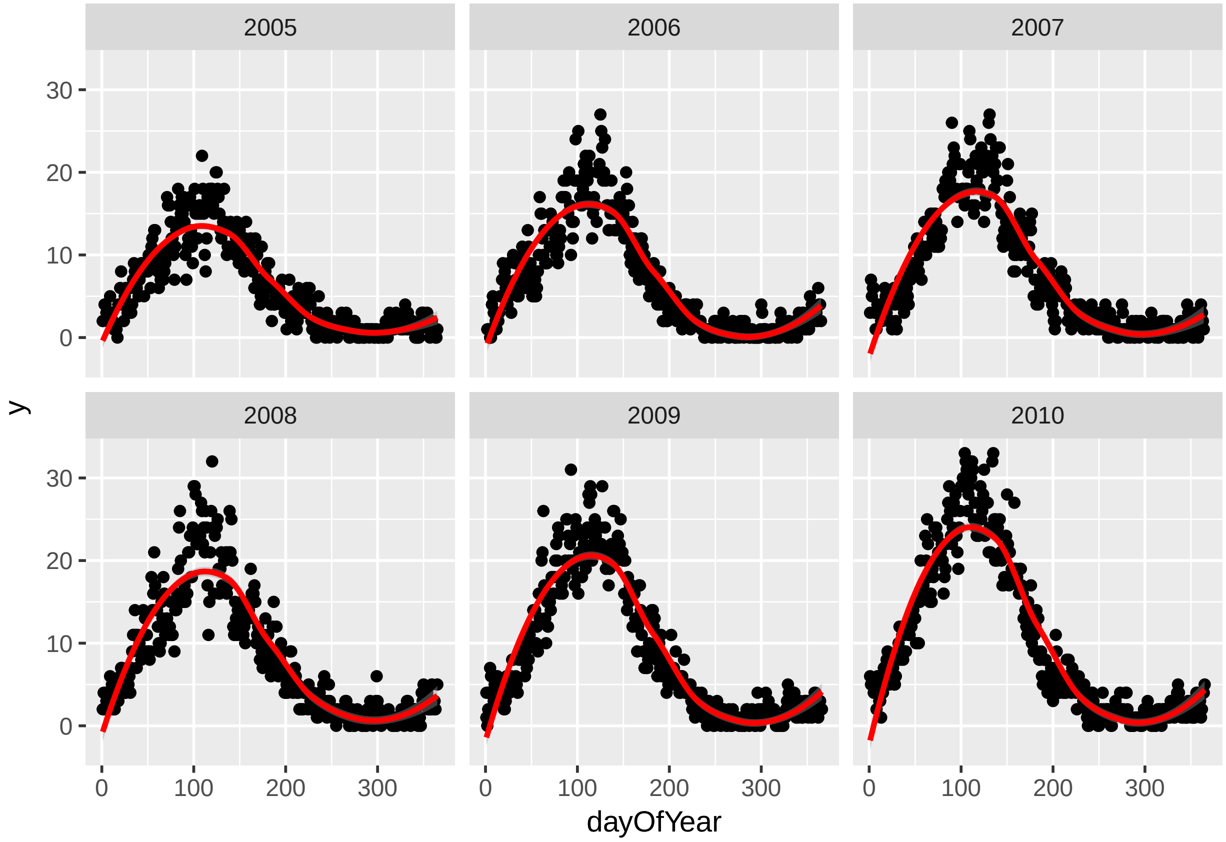
Chapter 4

Panel data: One area with autocorrelation

```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[,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)))]
```

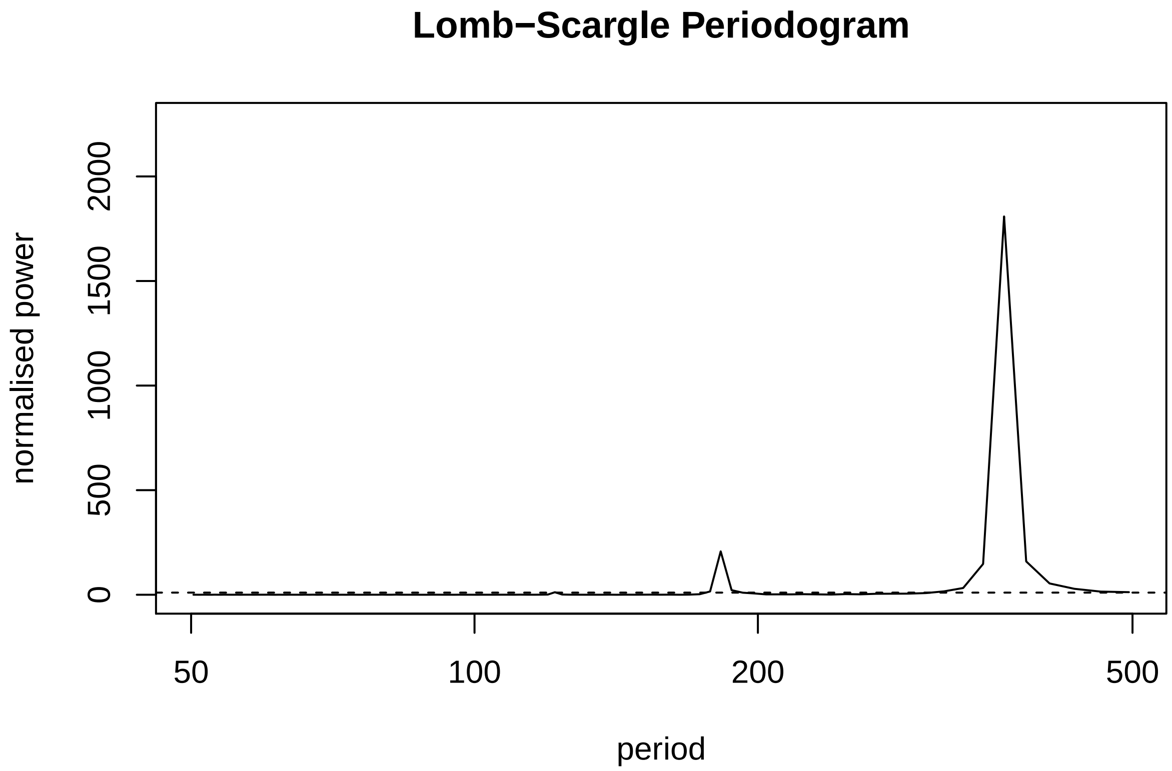
```r
q <- ggplot(d[year %in% c(2005:2010)],aes(x=dayOfYear,y=y))
q <- q + facet_wrap(~year)
q <- q + geom_point()
q <- q + stat_smooth(colour="red")
q
```

```
## `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")
```



```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))
```

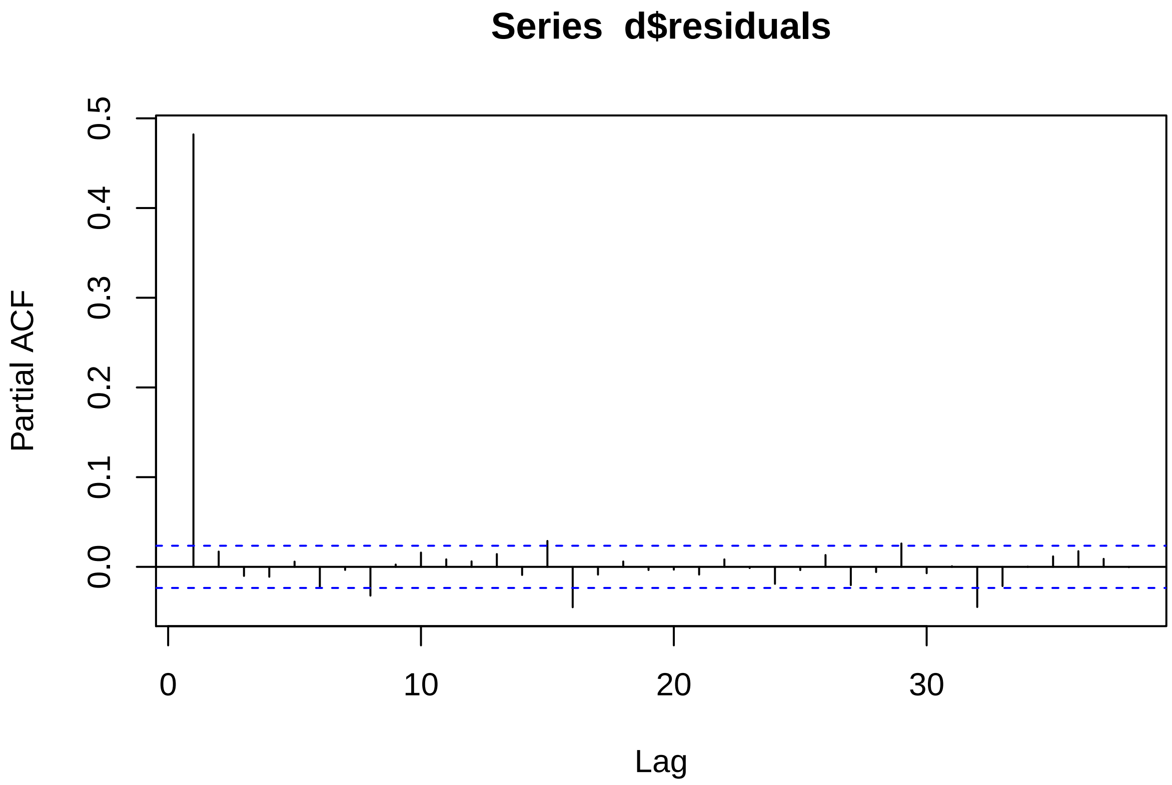
```
## Likelihood ratio test
##
## Model 1: y ~ yearMinus2000
## Model 2: y ~ yearMinus2000 + sin365 + cos365
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1   2 -43124
## 2   4 -14542  2 57163  < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

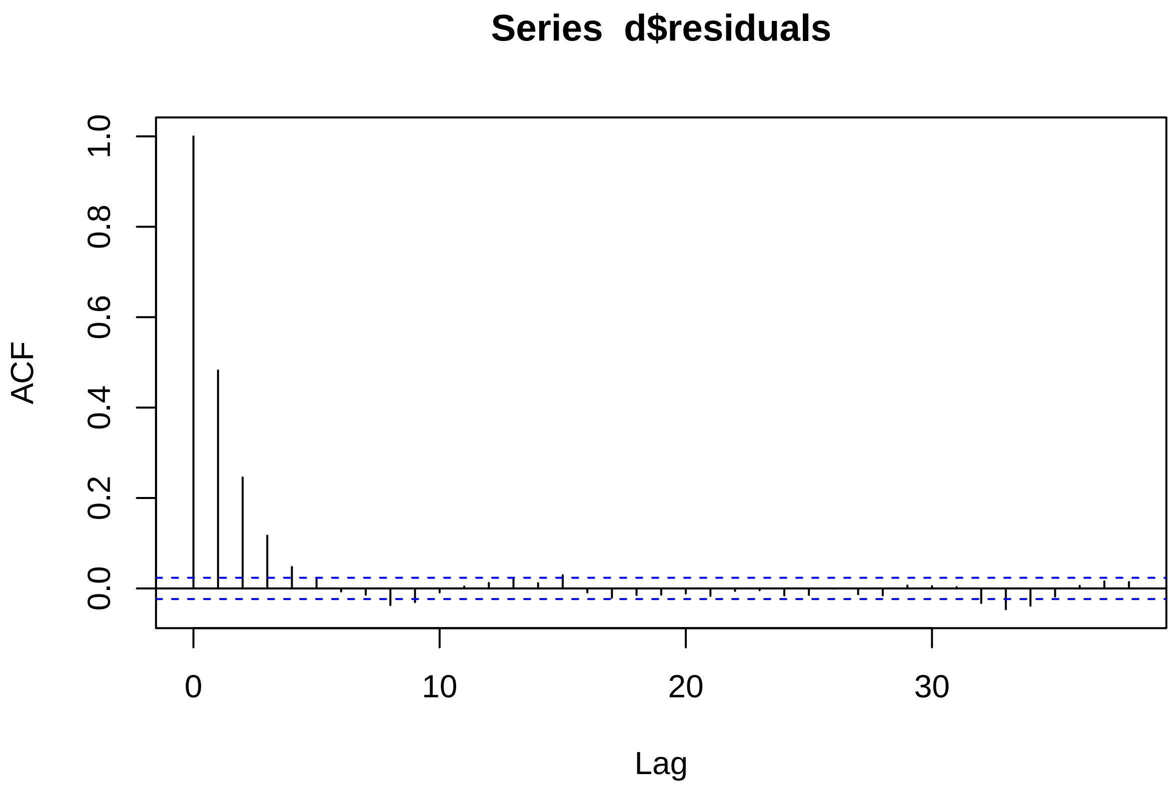
```
##
## Call:
## glm(formula = y ~ yearMinus2000 + sin365 + cos365, family = poisson(),
##     data = d)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6774  -0.6738  -0.0503   0.4920   3.5820
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (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
```

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

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



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



This means our model is bad, we have autocorrelation.

```
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))
```

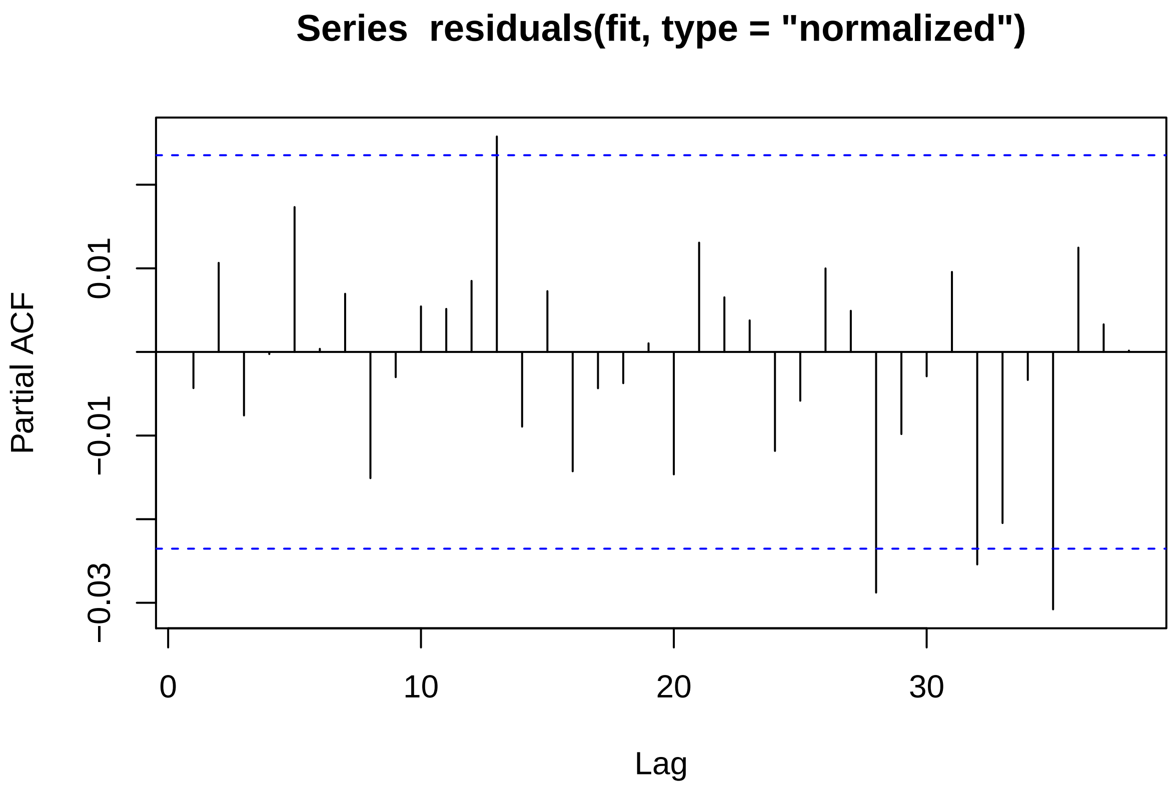
```
## iteration 1
```

```
summary(fit)
```

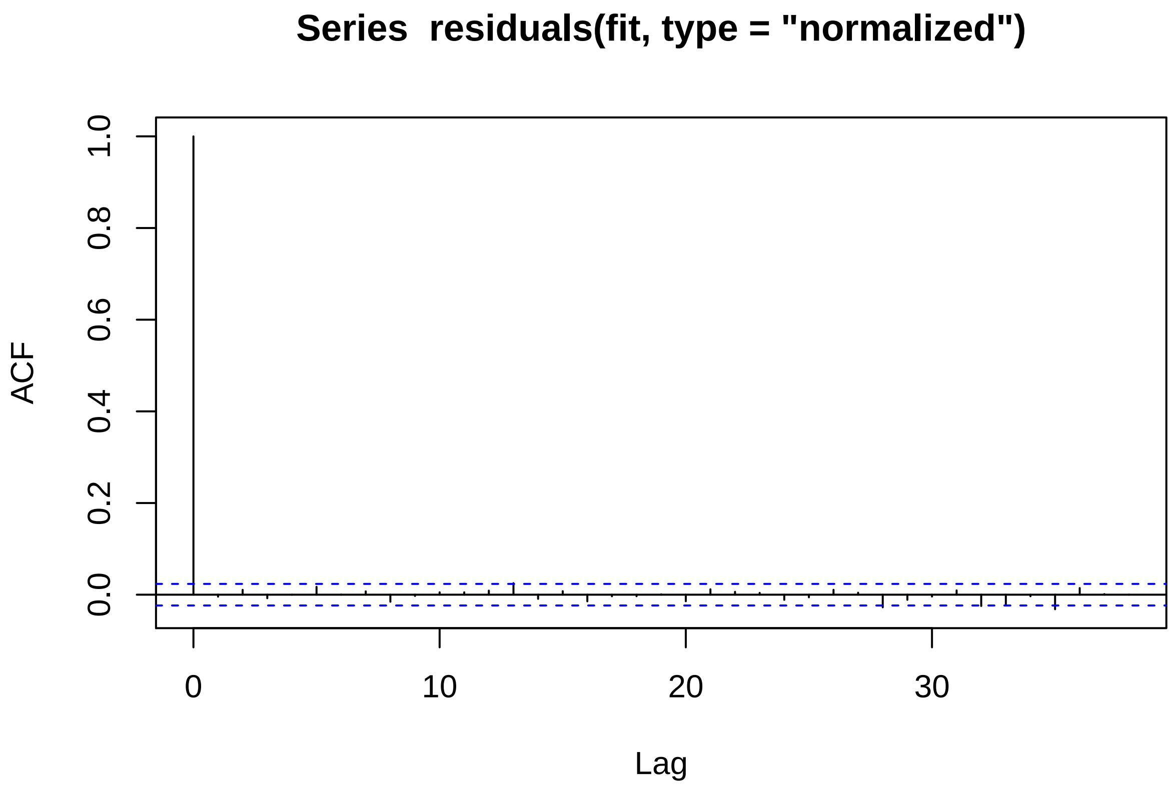
```
## Linear mixed-effects model fit by maximum likelihood
##  Data: d
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | ID
##         (Intercept) Residual
## StdDev: 1.149069e-05 0.841689
##
## Correlation Structure: AR(1)
##  Formula: ~dayOfSeries | ID
##  Parameter estimate(s):
##       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
```

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



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



```
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[,mu := exp(mainIntercept + fylkeIntercepts + 3*x)]
d[,y:=rpois(.N,mu)]
```

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

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: y ~ x + (1 | fylke)
##    Data: d
##
##      AIC      BIC   logLik deviance df.resid
##  15508.5  15525.3  -7751.3  15502.5     1997
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1132 -0.6422 -0.0260  0.6556  3.6029
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  fylke  (Intercept) 0.6167   0.7853
## Number of obs: 2000, groups:  fylke, 20
##
## Fixed effects:
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 3.378858   0.175660    19.2   <2e-16 ***
## x           2.990811   0.005991   499.2   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr)
## x -0.024
```

Chapter 6

Panel data: multiple areas without 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 := 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)))]
```

We then drill down into a few years, 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'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.99
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.0201
```

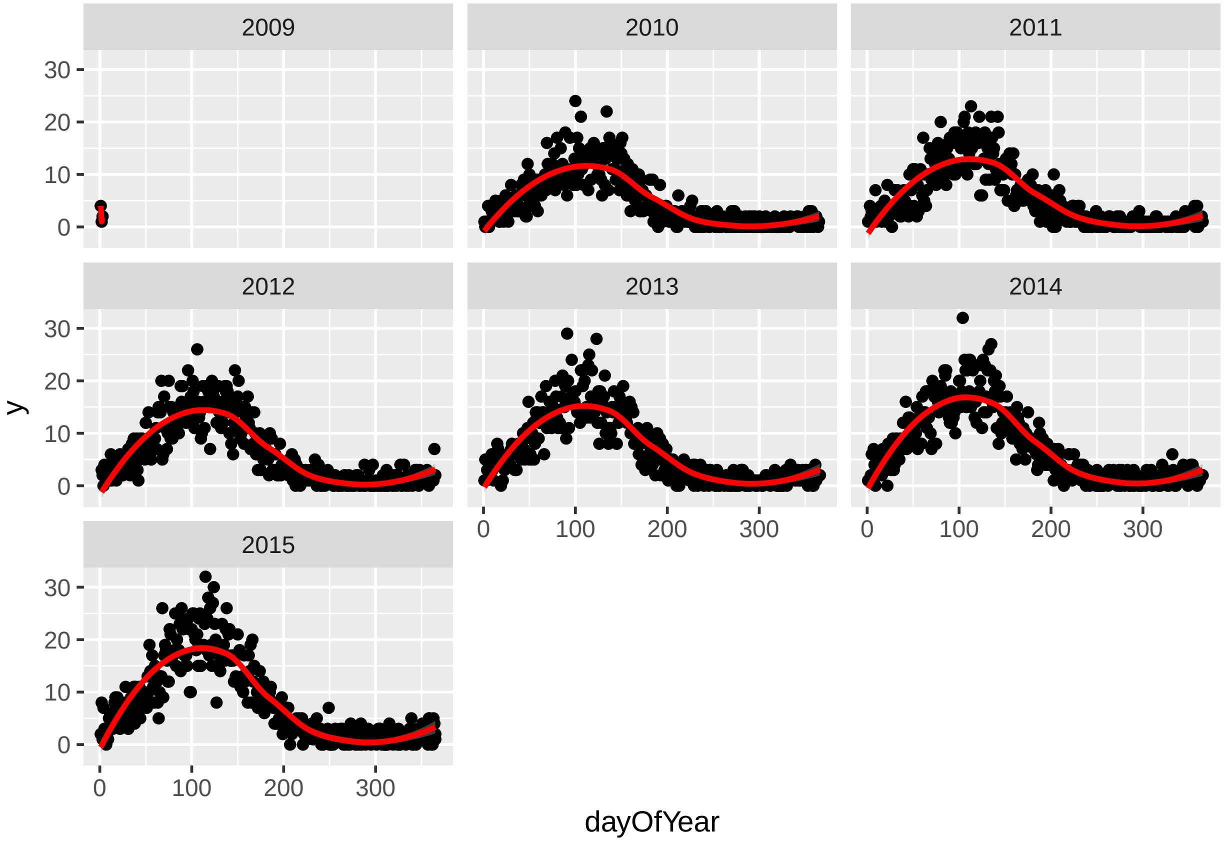
```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : span too small.
## fewer data values than degrees of freedom.
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used
## at 0.99
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 1.01
```

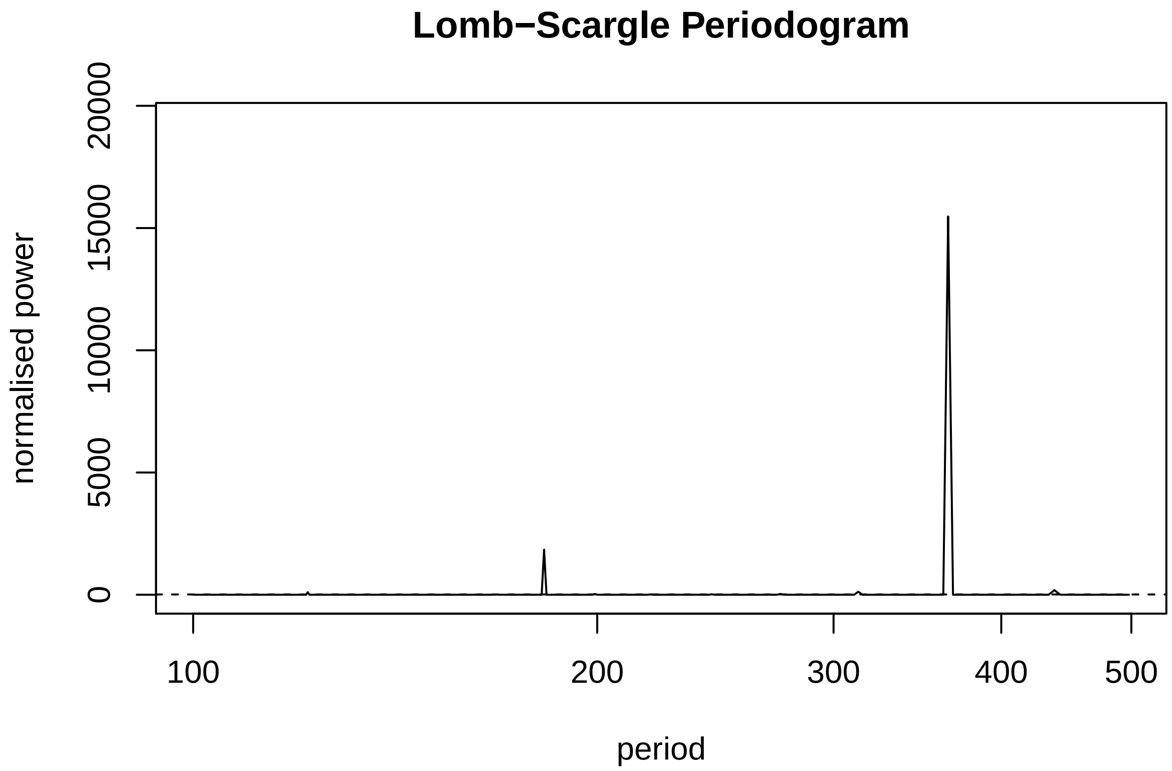
```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal
## condition number 0
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other
## near singularities as well. 1.0201
```



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")
```



```
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,
                correlation=nlme::corAR1(form=~dayOfSeries|fylke))
```

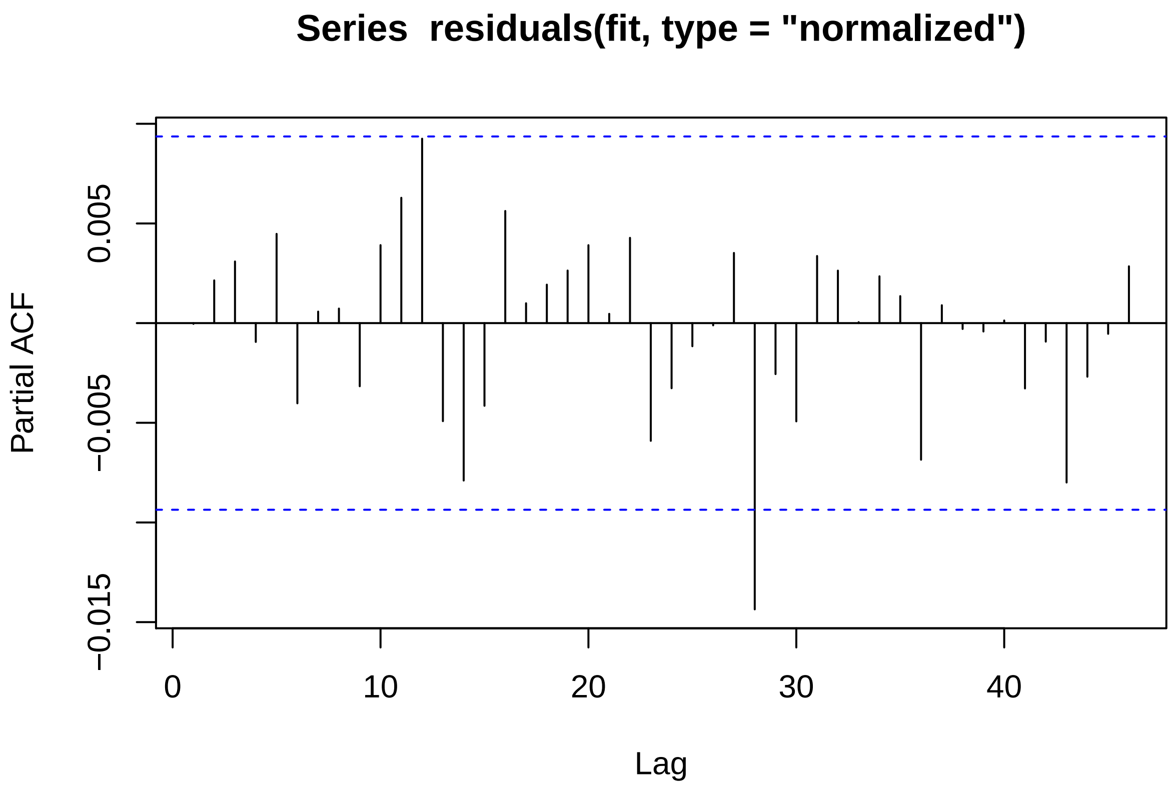
```
## iteration 1
```

```
summary(fit)
```

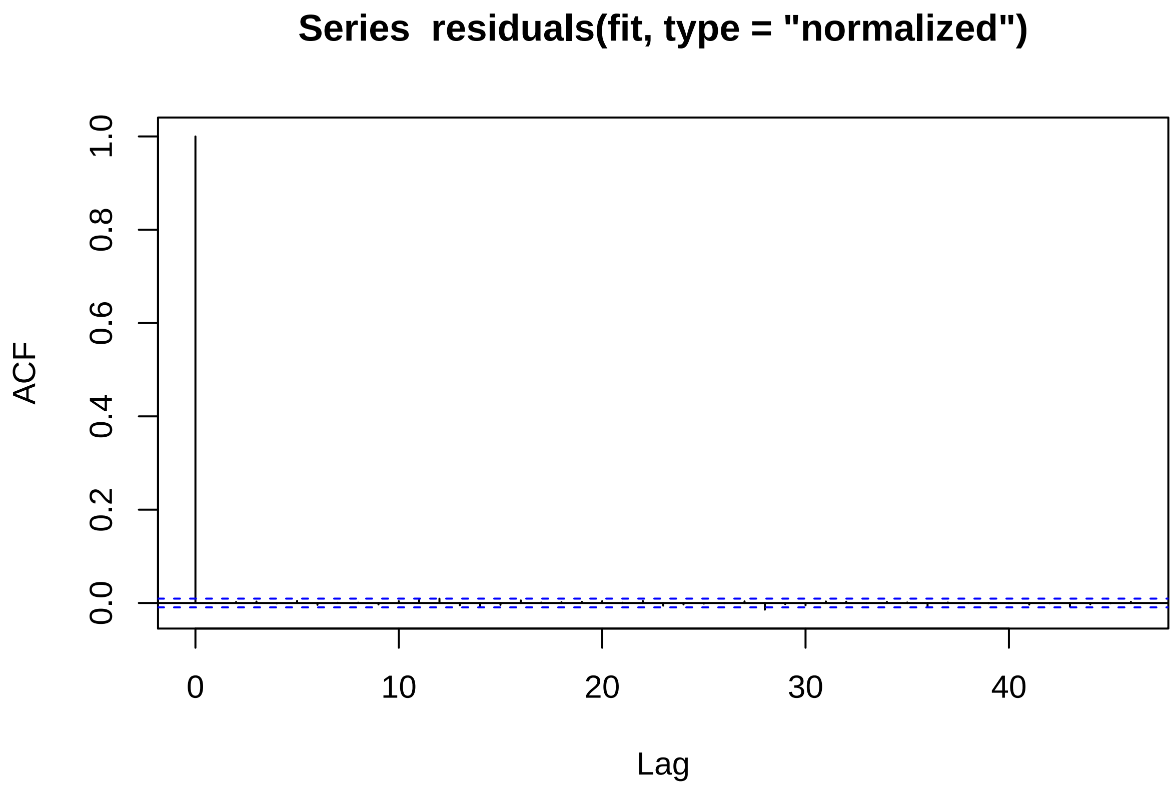
```
## Linear mixed-effects model fit by maximum likelihood
##  Data: d
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | fylke
##          (Intercept)  Residual
## StdDev: 1.708256e-05 0.9976713
##
## Correlation Structure: AR(1)
##  Formula: ~dayOfSeries | fylke
##  Parameter estimate(s):
##         Phi
## 0.002841665
## Variance function:
```

```
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: y ~ yearMinus2000 + sin365 + cos365
##                     Value   Std.Error    DF   t-value p-value
## (Intercept)     0.1122528 0.014529606 43797    7.7258       0
## yearMinus2000   0.0989047 0.001112632 43797   88.8926       0
## sin365          1.4095094 0.003705852 43797  380.3469       0
## cos365         -0.5109372 0.003092449 43797 -165.2209       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
## -3.1968230 -0.8238741 -0.0750183  0.6340046  5.8245241
##
## Number of Observations: 43820
## Number of Groups: 20
```

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



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



```
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 := 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)))]
```

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

```r
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'
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 0.99
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 1.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 1.0201
```

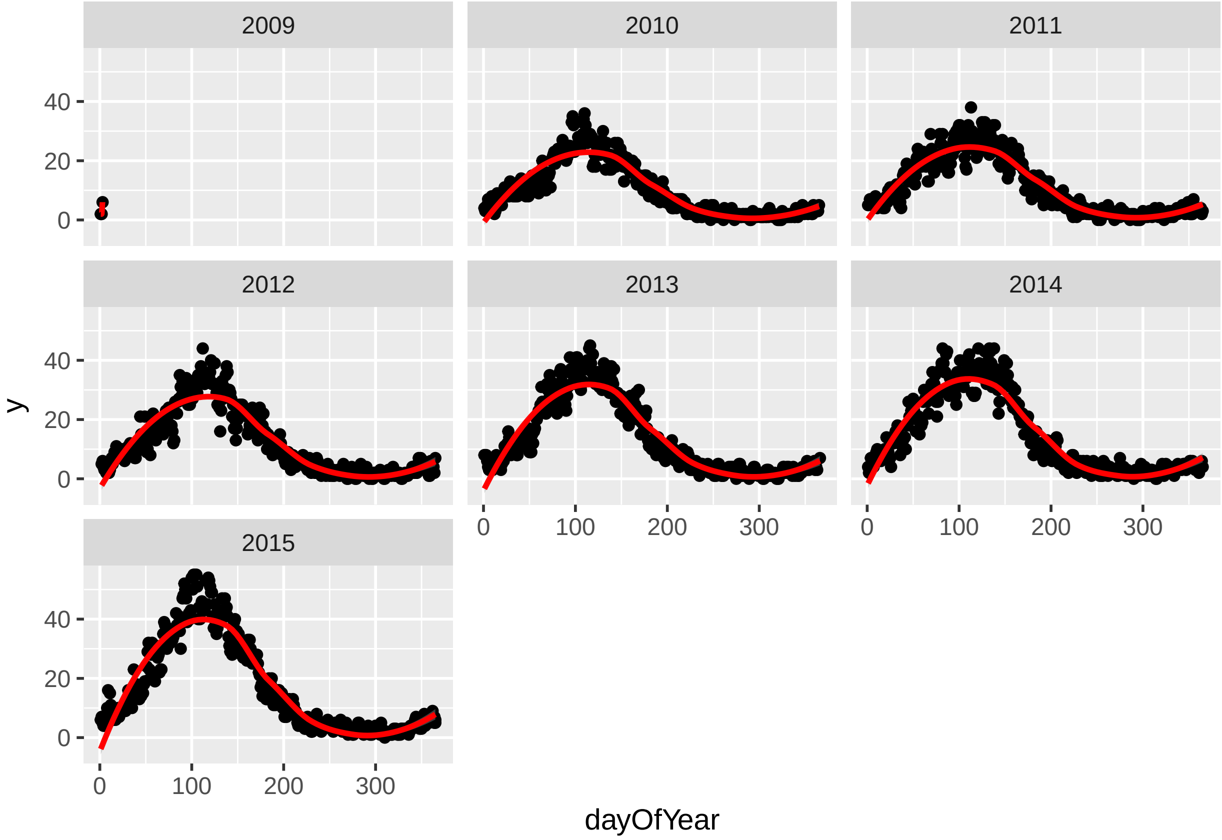
```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : span too small.
## fewer data values than degrees of freedom.
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used
## at 0.99
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 1.01
```

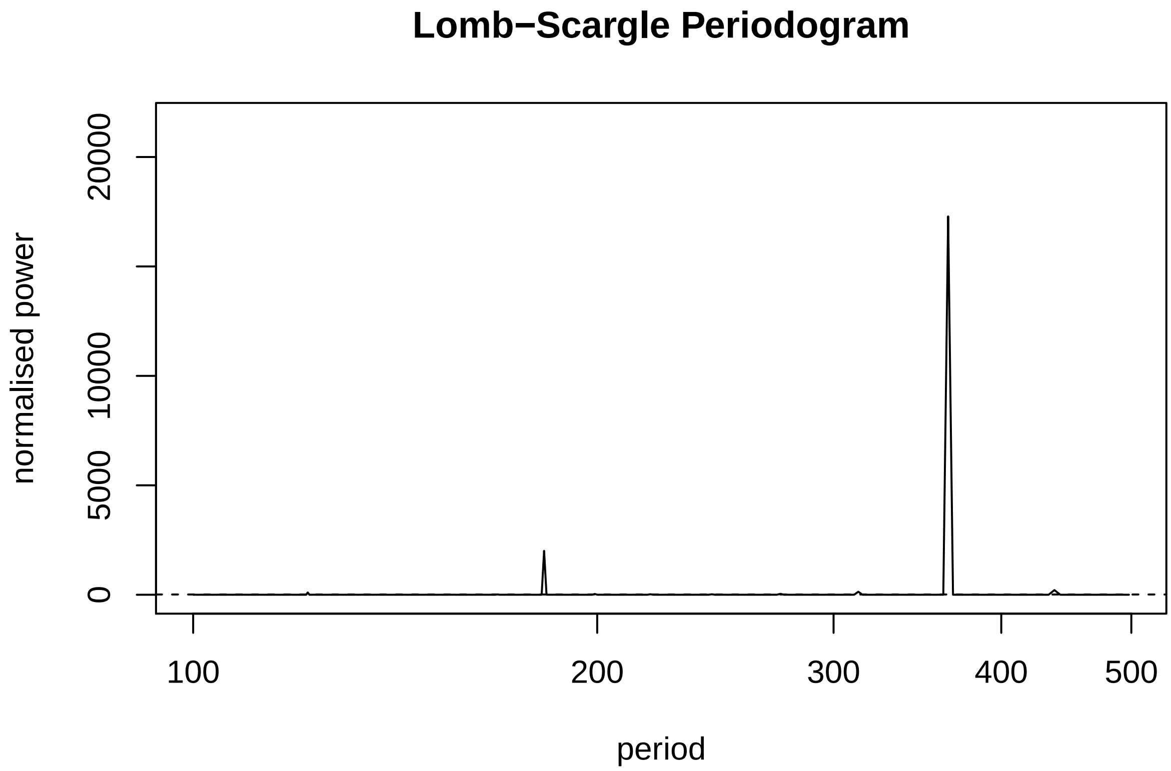
```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal
## condition number 0
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other
## near singularities as well. 1.0201
```



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")
```



```
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,
                correlation=nlme::corAR1(form=~dayOfSeries|fylke))
```

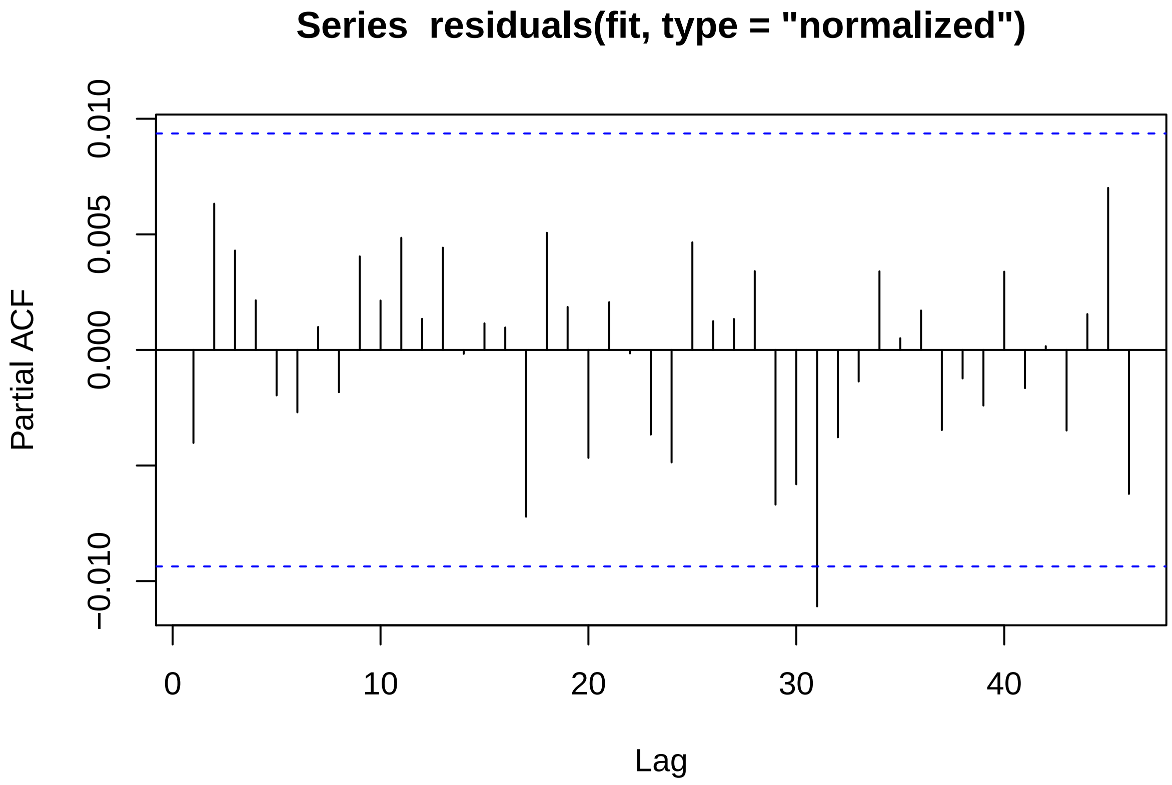
```
## iteration 1
```

```
summary(fit)
```

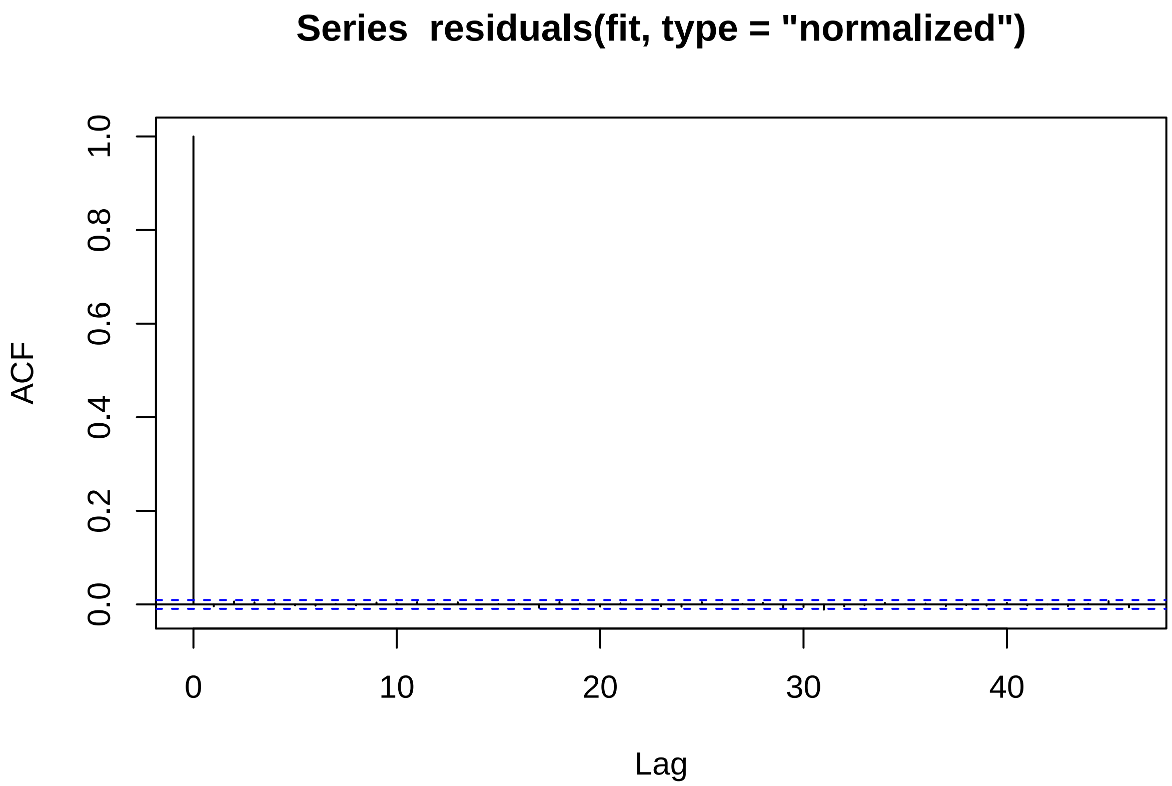
```
## Linear mixed-effects model fit by maximum likelihood
##  Data: d
##   AIC BIC logLik
##    NA  NA     NA
##
## Random effects:
##  Formula: ~1 | fylke
##         (Intercept) Residual
## StdDev: 0.003916838 0.822866
##
## Correlation Structure: AR(1)
##  Formula: ~dayOfSeries | fylke
##  Parameter estimate(s):
##       Phi
## 0.4771948
## Variance function:
```

```
##  Structure: fixed weights
##  Formula: ~invwt
## Fixed effects: y ~ yearMinus2000 + sin365 + cos365
##                     Value   Std.Error    DF   t-value p-value
## (Intercept)     0.8181967 0.014201621 43797   57.6129       0
## yearMinus2000   0.0982444 0.001085637 43797   90.4947       0
## sin365          1.4007640 0.003607254 43797  388.3187       0
## cos365         -0.5234863 0.003020395 43797 -173.3171       0
##  Correlation:
##                (Intr) yM2000 sin365
## yearMinus2000 -0.977
## sin365        -0.149  0.001
## cos365         0.067 -0.001 -0.153
##
## Standardized Within-Group Residuals:
##         Min          Q1         Med          Q3         Max
## -3.40197340 -0.70762882 -0.06465241  0.62676531  5.47900204
##
## Number of Observations: 43820
## Number of Groups: 20
```

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



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



```
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"
```