PSTAT 174/274 Spring 2023 Project

Dylan Berneman

2023-12-14

```
pkgs <- c("pkgbuild", "pkgconfig", "plyr", "dplyr", "MASS", "grid", "gtable", "gtools", "jsonlite", "kn
          "~/PSTAT274/Project/Packages/astsa_2.0.tar.gz",
          "~/PSTAT274/Project/Packages/Dict_0.1.0.tar.gz",
          "~/PSTAT274/Project/Packages/UnitCircle_0.1.3.tar.gz",
          "~/PSTAT274/Project/Packages/expm_0.999-7.tar.gz",
          "~/PSTAT274/Project/Packages/polynom_1.4-1.tar.gz",
          "~/PSTAT274/Project/Packages/PolynomF_2.0-5.tar.gz",
          "~/PSTAT274/Project/Packages/locfit_1.5-9.8.tar.gz",
          "~/PSTAT274/Project/Packages/leaps_3.1.tar.gz",
          "~/PSTAT274/Project/Packages/rbibutils_2.2.15.tar.gz",
          "~/PSTAT274/Project/Packages/Rdpack_2.5.tar.gz",
          "~/PSTAT274/Project/Packages/Formula_1.2-5.tar.gz",
          "~/PSTAT274/Project/Packages/lagged_0.3.2.tar.gz",
          "~/PSTAT274/Project/Packages/ltsa_1.4.6.tar.gz",
          "~/PSTAT274/Project/Packages/insight_0.19.5.tar.gz",
          "~/PSTAT274/Project/Packages/MuMIn_1.47.5.tar.gz",
          "~/PSTAT274/Project/Packages/nortest_1.0-4.tar.gz",
          "~/PSTAT274/Project/Packages/nortsTest_1.0.3.tar.gz",
          "~/PSTAT274/Project/Packages/sarima_0.9.1.tar.gz",
          "~/PSTAT274/Project/Packages/sae_1.3.tar.gz",
          "~/PSTAT274/Project/Packages/tseries_0.10-54.tar.gz",
          "~/PSTAT274/Project/Packages/TSA_1.3.1.tar.gz")

for(i in 1:length(pkgs)){
  if(substr(pkgs[i], 1, 28) != "~/PSTAT274/Project/Packages/"){
    res = pkgs[i]
    if(res %in% installed.packages() == FALSE){
      install.packages(res, dependencies = TRUE)}}
  if(substr(pkgs[i], 1, 28) == "~/PSTAT274/Project/Packages/"){
    res = str_match(pkgs[i], "~/PSTAT274/Project/Packages/(.*?)_")[2]
    if(res %in% installed.packages() == FALSE){
      install.packages(pkgs[i], repos=NULL, type="source")}}
  library(res, character.only = TRUE)}
```

We will be testing the datasets corresponding to:
- The smoothed amount of daily new COVID cases (Cases) - The smoothed amount of daily new deaths from COVID (Deathss) - The daily mortality rate of those who contract COVID (Mortality) - The daily percent of the world that has been vaccinated (Vaccinated)

For each data set, the length of the time series will be evaluated in at most 3 ways:
- [January 8, 2020 - May 24, 2023] - Cases will begin on the 6th observation day, January 8, 2020, since there were some observations of 0 new cases before then - [January 12, 2020 - May 24, 2023] - Deaths and Mortality will begin on the 10th observation day, January 12, 2020, since that was the time of the first death - [January 8, 2020 - December 3, 2020] - This will account for all daily observations of Cases before the first person got vaccinated - [January 12, 2020 - December 3, 2020] - This will account for all daily observations for Deaths and Mortality before the first person got vaccinated - [December 4, 2020 - May 24, 2023] - This will account for all daily observations for Cases, Deaths, Mortality, and Vaccinated since the first person got vaccinated.

Additionally, since daily observations can be very noisy, we will also be testing the weekly average for each of the previously mentioned time series. In total, that will be 20 data sets.

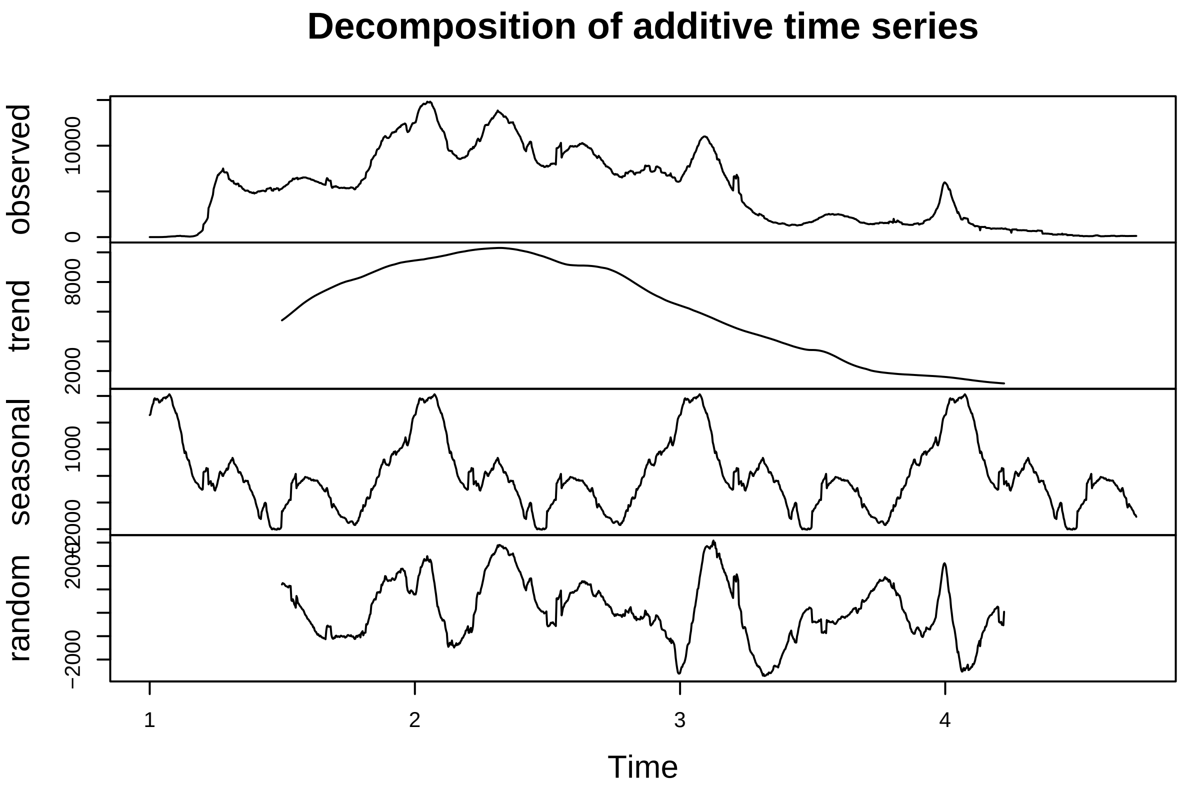
In regards to the names of the data sets:
- '1' refers to the entire pandemic - '2' refers to the time before the first vaccination - '3' refers to the time since the first vaccination - 'a' denotes daily observations - 'b' denotes average weekly observations

Original Datasets

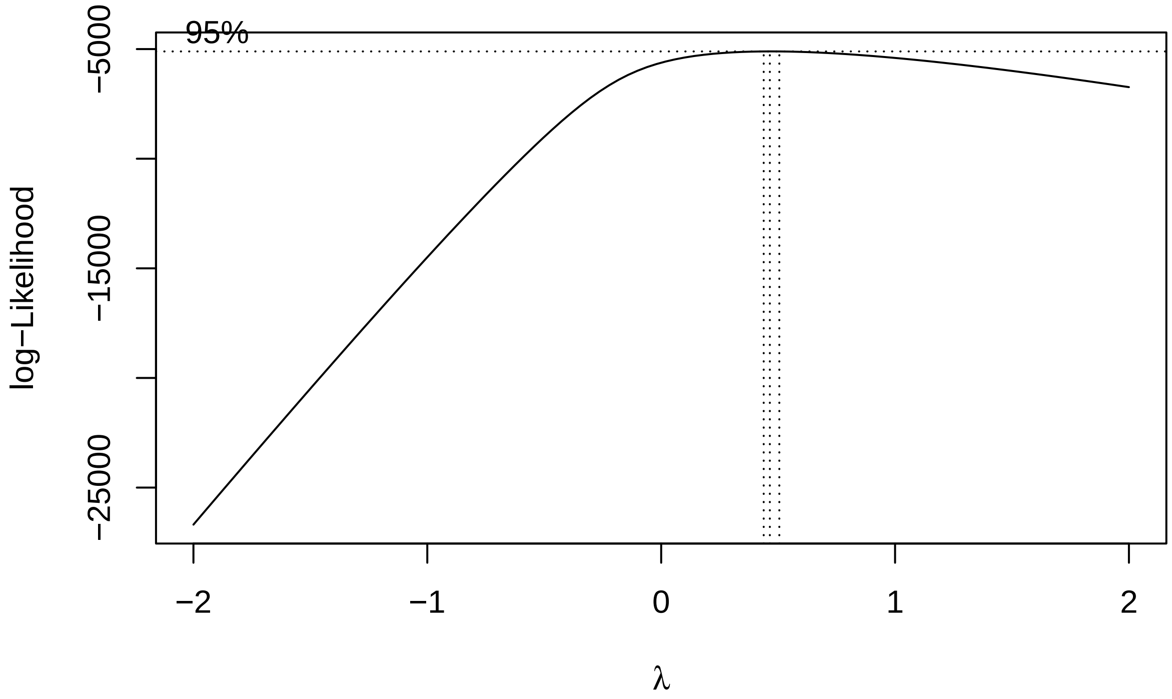
```
options(scipen = 9)
covid <- read_excel("covid_final_data.xlsx")
```

Deaths: [January 8, 2020 - September 27, 2023]

```
# Original
Deaths = ts(covid$new_deaths_smoothed[1:1359])
y <- ts(as.ts(Deaths), frequency = 365)
decomp <- decompose(y)
plot(decomp)
```



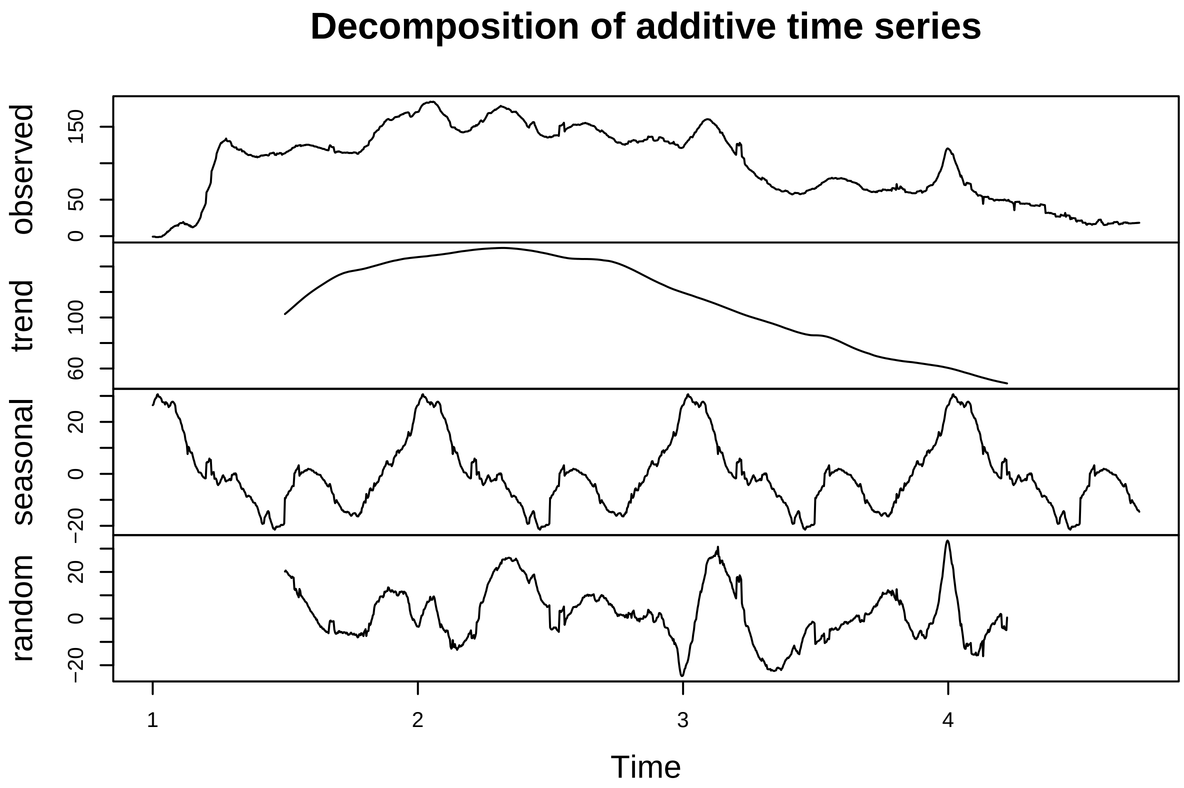
```
# Box-Cox
bcTransform <- boxcox(Deaths~ as.numeric(1:length(Deaths)))
```



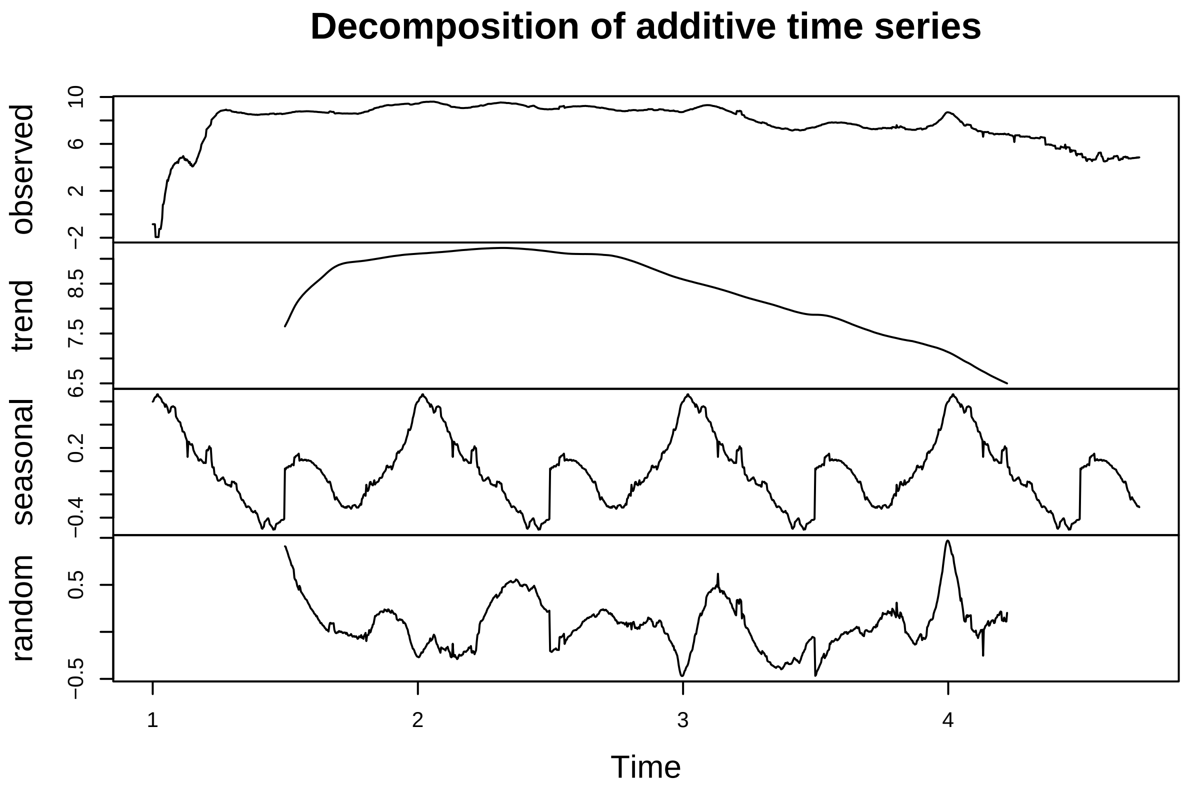
```
bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
```

```
## [1] 0.4646465
```

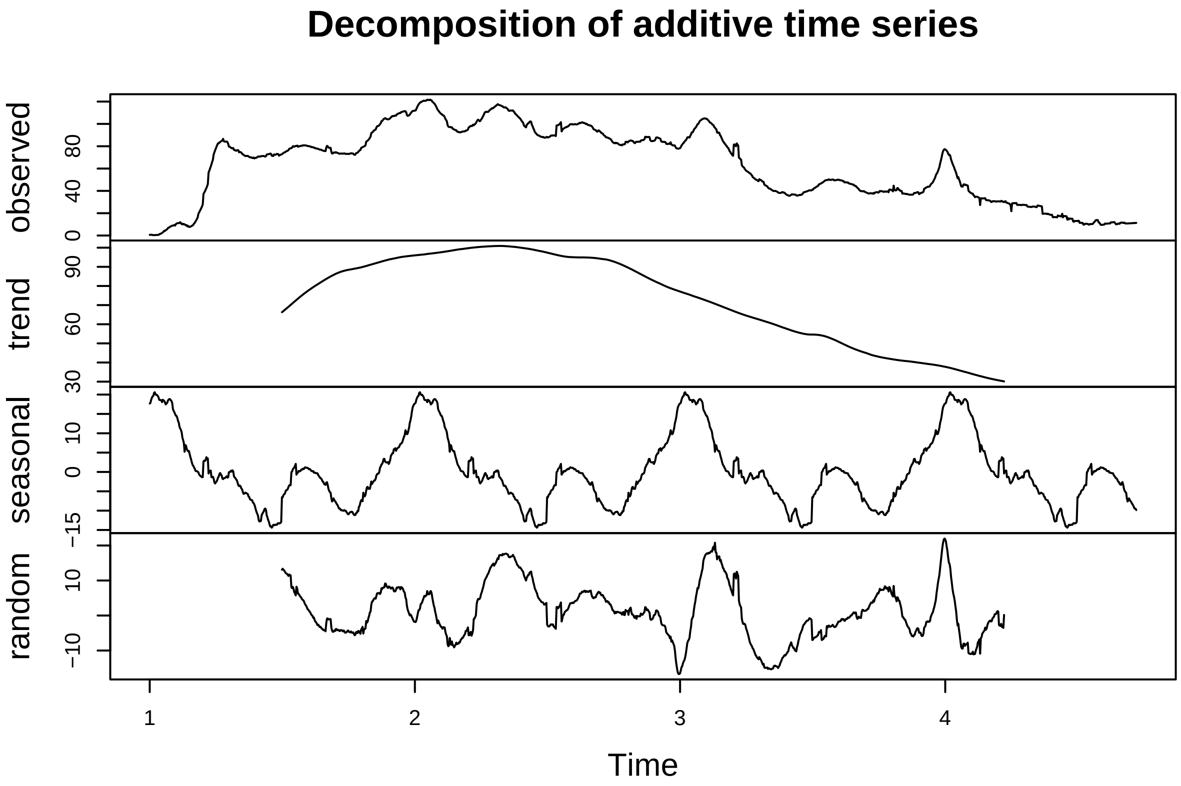
```
lambda=bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
Deaths.bc = (1/lambda)*(Deaths^lambda-1)
y <- ts(as.ts(Deaths.bc), frequency = 365)
decomp <- decompose(y)
plot(decomp)
```



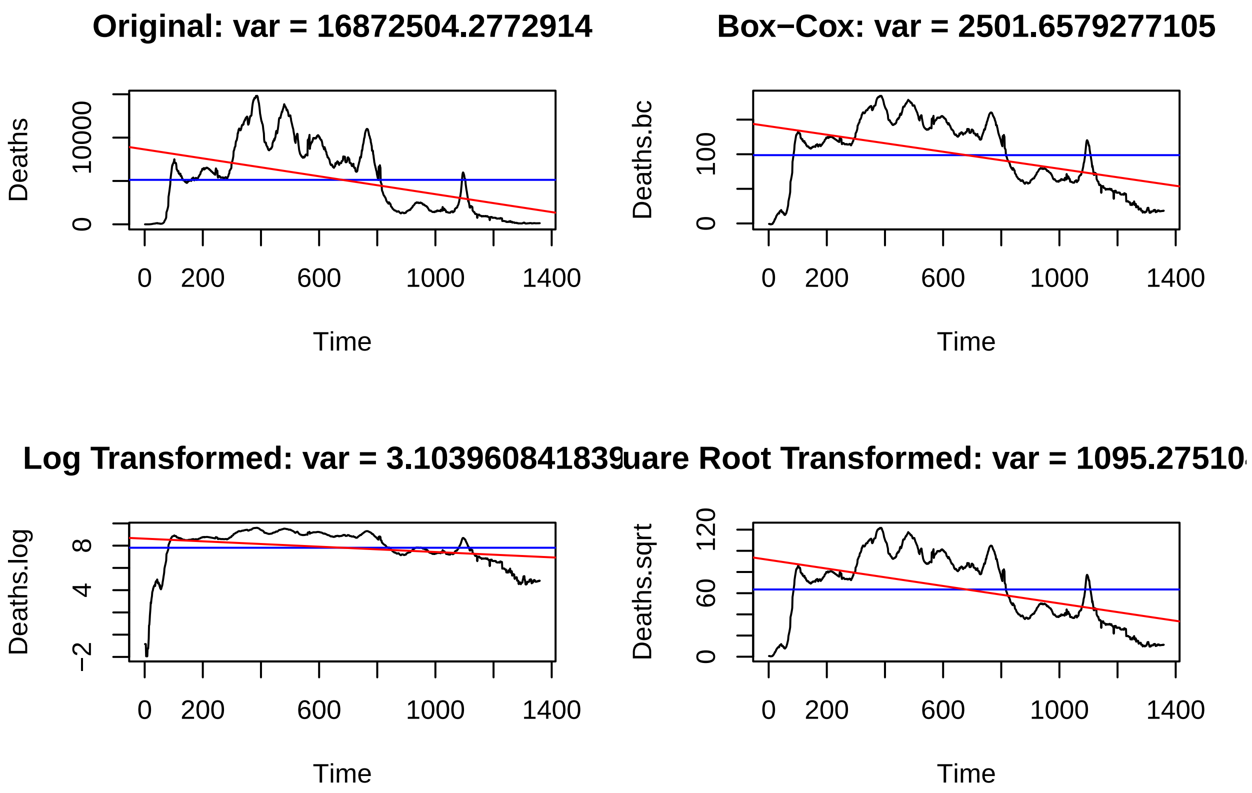
```
# Log
Deaths.log = log(Deaths)
y <- ts(as.ts(Deaths.log), frequency = 365)
decomp <- decompose(y)
plot(decomp)
```



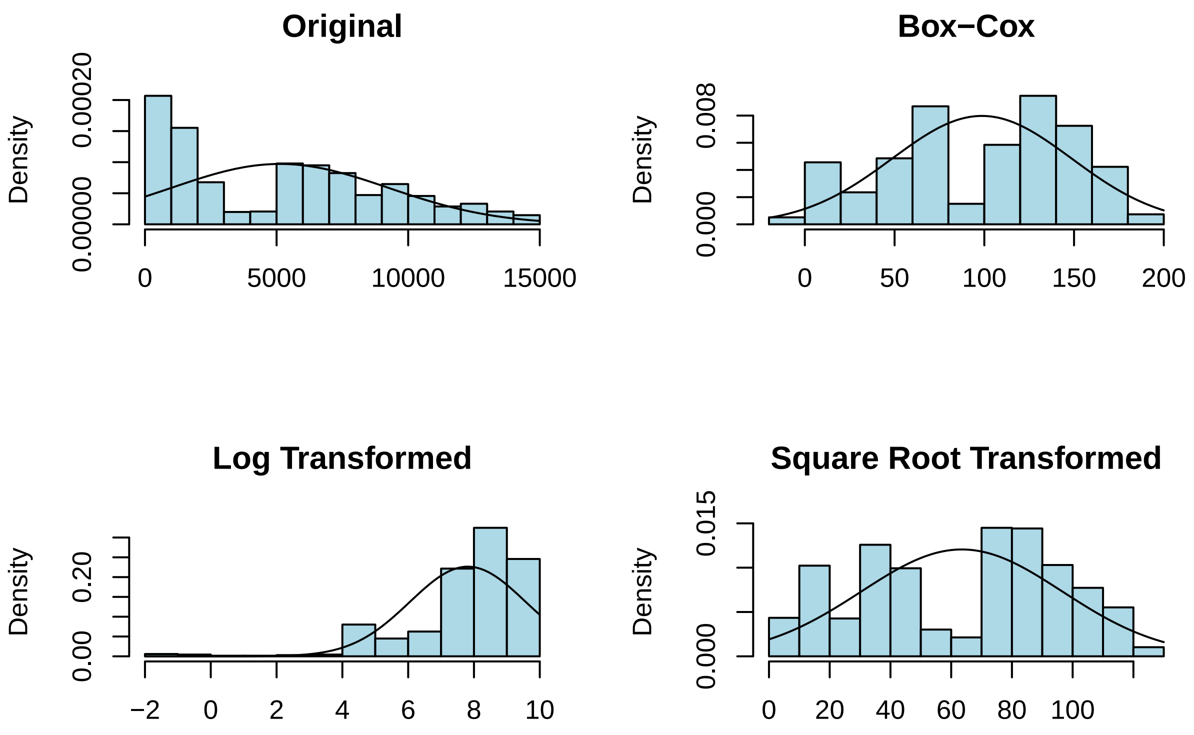
```
# Squared Root
Deaths.sqrt = sqrt(Deaths)
y <- ts(as.ts(Deaths.sqrt), frequency = 365)
decomp <- decompose(y)
plot(decomp)
```



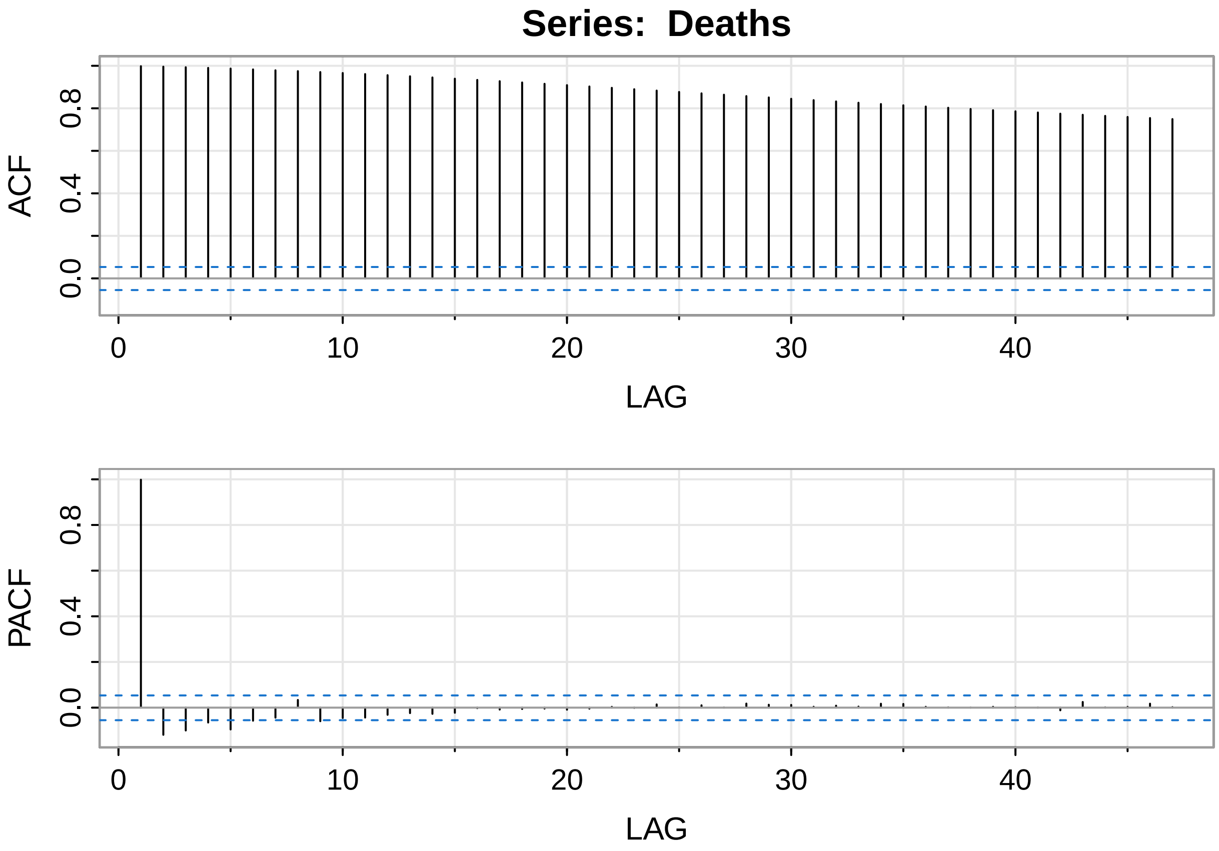
```
# Plots
op <- par("mfrow", "mar")
par(mfrow = c(2, 2))
plot.ts(Deaths, main = paste0("Original: var = ", var(Deaths)))
abline(h=mean(Deaths), col= "blue")
abline(lm(Deaths ~ as.numeric(1:length(Deaths))), col="red")
plot.ts(Deaths.bc, main = paste0("Box-Cox: var = ", var(Deaths.bc)))
abline(h=mean(Deaths.bc), col= "blue")
abline(lm(Deaths.bc ~ as.numeric(1:length(Deaths.bc))), col="red")
plot.ts(Deaths.log, main = paste0("Log Transformed: var = ", var(Deaths.log)))
abline(h=mean(Deaths.log), col= "blue")
abline(lm(Deaths.log ~ as.numeric(1:length(Deaths.log))), col="red")
plot.ts(Deaths.sqrt, main = paste0("Square Root Transformed: var = ", var(Deaths.sqrt)))
abline(h=mean(Deaths.sqrt), col= "blue")
abline(lm(Deaths.sqrt ~ as.numeric(1:length(Deaths.sqrt))), col="red")
```



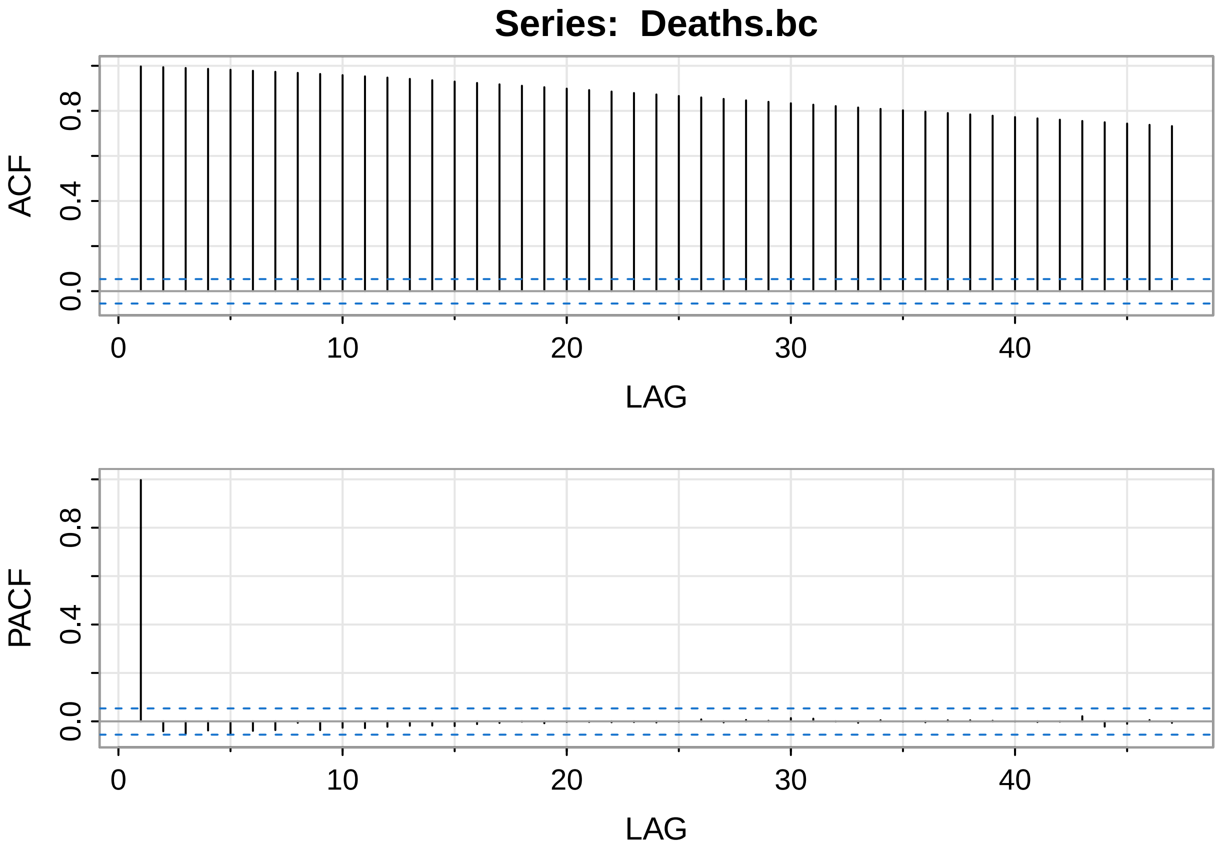
```
par(mfrow = op$mfrow, mar = op$mar)

# Histograms
op <- par("mfrow", "mar")
par(mfrow = c(2, 2))
hist(Deaths, col="light blue", xlab="", main="Original", probability = TRUE)
m <-mean(Deaths)
std <- sqrt(var(Deaths))
curve(dnorm(x,m,std), add=TRUE )
hist(Deaths.bc, col="light blue", xlab="", main="Box-Cox", probability = TRUE)
m <-mean(Deaths.bc)
std <- sqrt(var(Deaths.bc))
curve(dnorm(x,m,std), add=TRUE )
hist(Deaths.log, col="light blue", xlab="", main="Log Transformed", probability = TRUE)
m <-mean(Deaths.log)
std <- sqrt(var(Deaths.log))
curve(dnorm(x,m,std), add=TRUE )
hist(Deaths.sqrt, col="light blue", xlab="", main="Square Root Transformed", probability = TRUE)
m <-mean(Deaths.sqrt)
std <- sqrt(var(Deaths.sqrt))
curve(dnorm(x,m,std), add=TRUE )
```



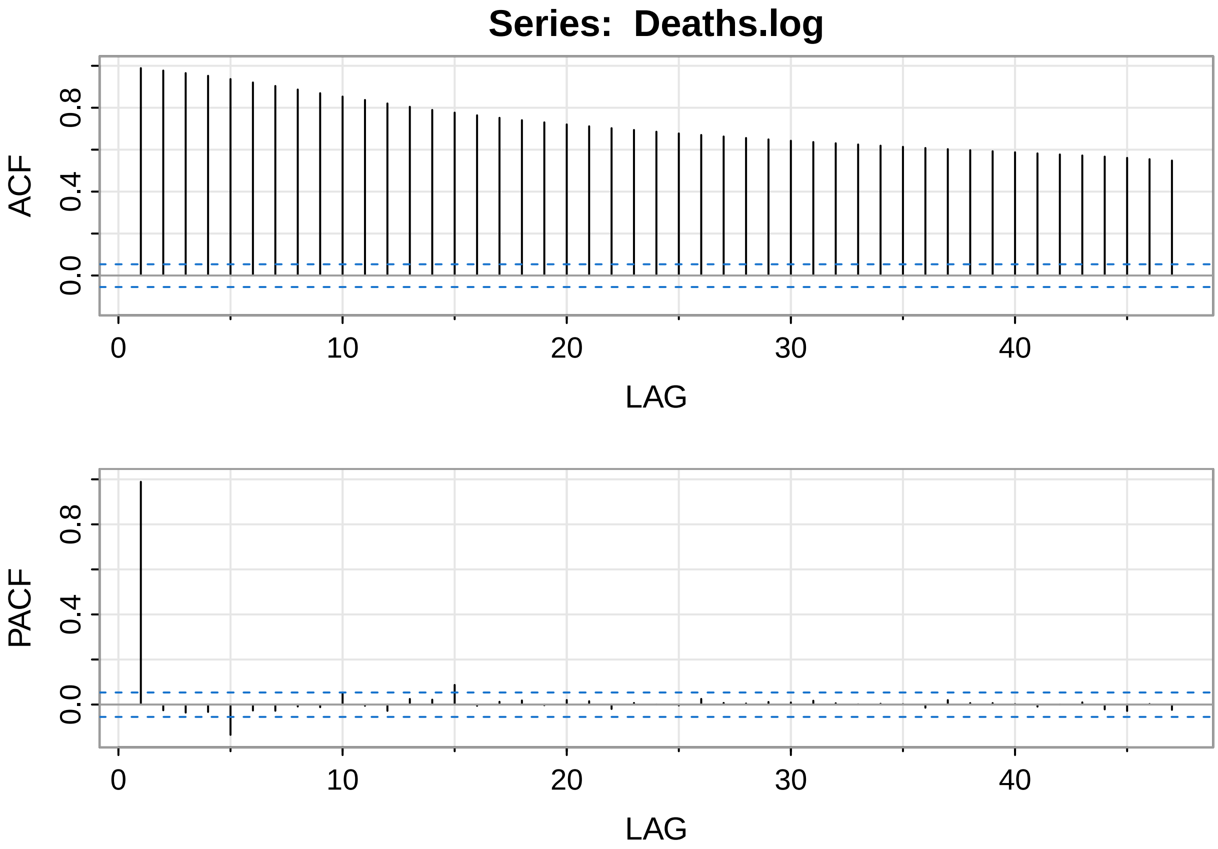
```
par(mfrow = op$mfrow, mar = op$mar)

# ACF & PACF
x = acf2(Deaths)
```



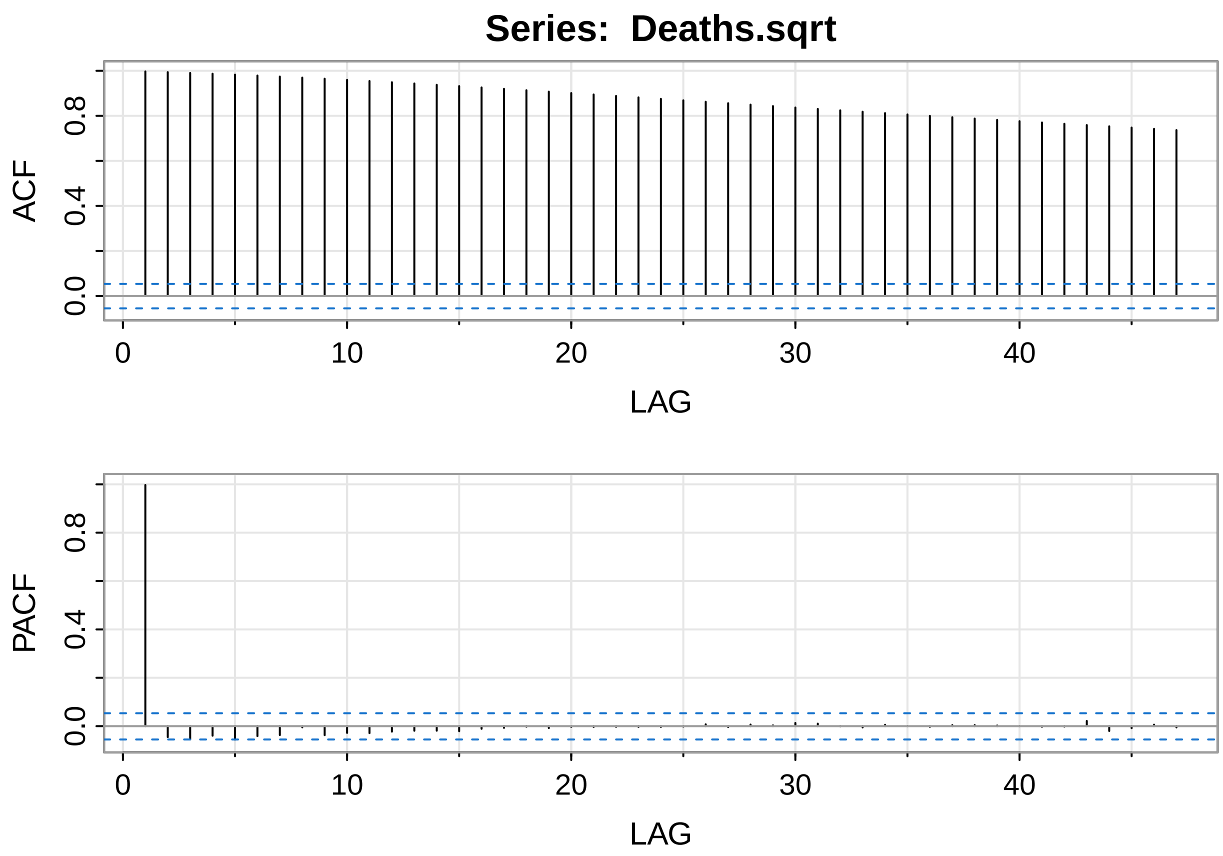
```
x = acf2(Deaths.bc)
```



```r
x = acf2(Deaths.log)
```



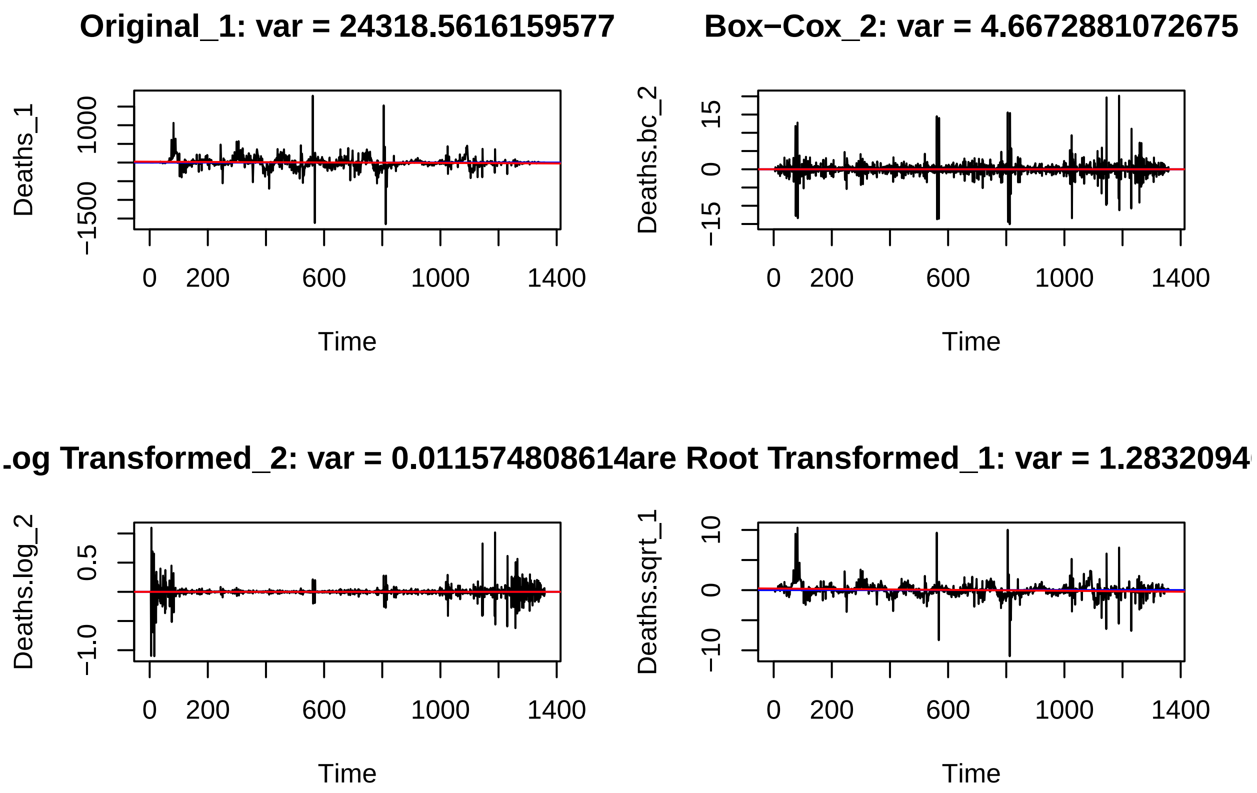
```r
x = acf2(Deaths.sqrt)
```



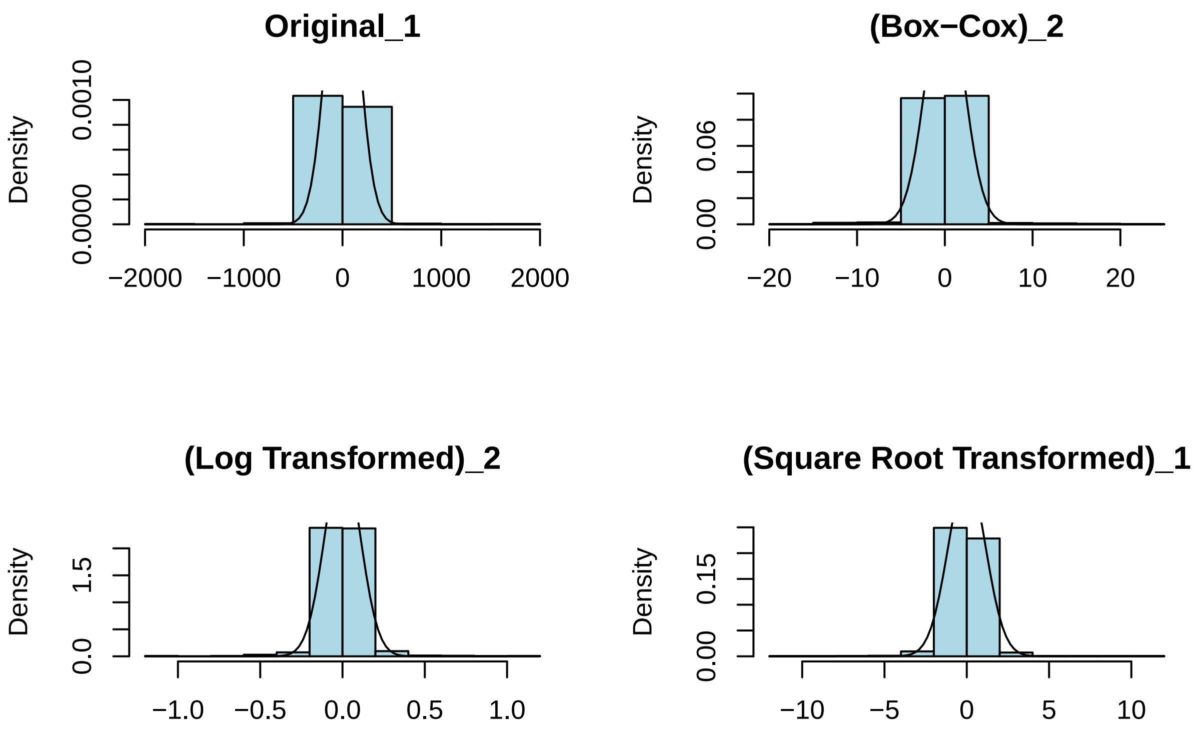
Differencing

```
diff = ndiffs(Deaths)
Deaths_1 <- arima(Deaths, order=c(0,1,0), method="ML")$residuals
diff.bc = ndiffs(Deaths.bc)
Deaths.bc_2 <- arima(Deaths.bc, order=c(0,2,0), method="ML")$residuals
diff.log = ndiffs(Deaths.log)
Deaths.log_2 <- arima(Deaths.log, order=c(0,2,0), method="ML")$residuals
diff.sqrt = ndiffs(Deaths)
Deaths.sqrt_1 <- arima(Deaths.sqrt, order=c(0,1,0), method="ML")$residuals

# Plots
op <- par("mfrow", "mar")
par(mfrow = c(2, 2))
plot.ts(Deaths_1, main = paste0("Original_1: var = ", var(Deaths_1)))
abline(h=mean(Deaths_1), col= "blue")
abline(lm(Deaths_1 ~ as.numeric(1:length(Deaths_1))), col="red")
plot.ts(Deaths.bc_2, main = paste0("Box-Cox_2: var = ", var(Deaths.bc_2)))
abline(h=mean(Deaths.bc_2), col= "blue")
abline(lm(Deaths.bc_2 ~ as.numeric(1:length(Deaths.bc_2))), col="red")
plot.ts(Deaths.log_2, main = paste0("Log Transformed_2: var = ", var(Deaths.log_2)))
abline(h=mean(Deaths.log_2), col= "blue")
abline(lm(Deaths.log_2 ~ as.numeric(1:length(Deaths.log_2))), col="red")
plot.ts(Deaths.sqrt_1, main = paste0("Square Root Transformed_1: var = ", var(Deaths.sqrt_1)))
```

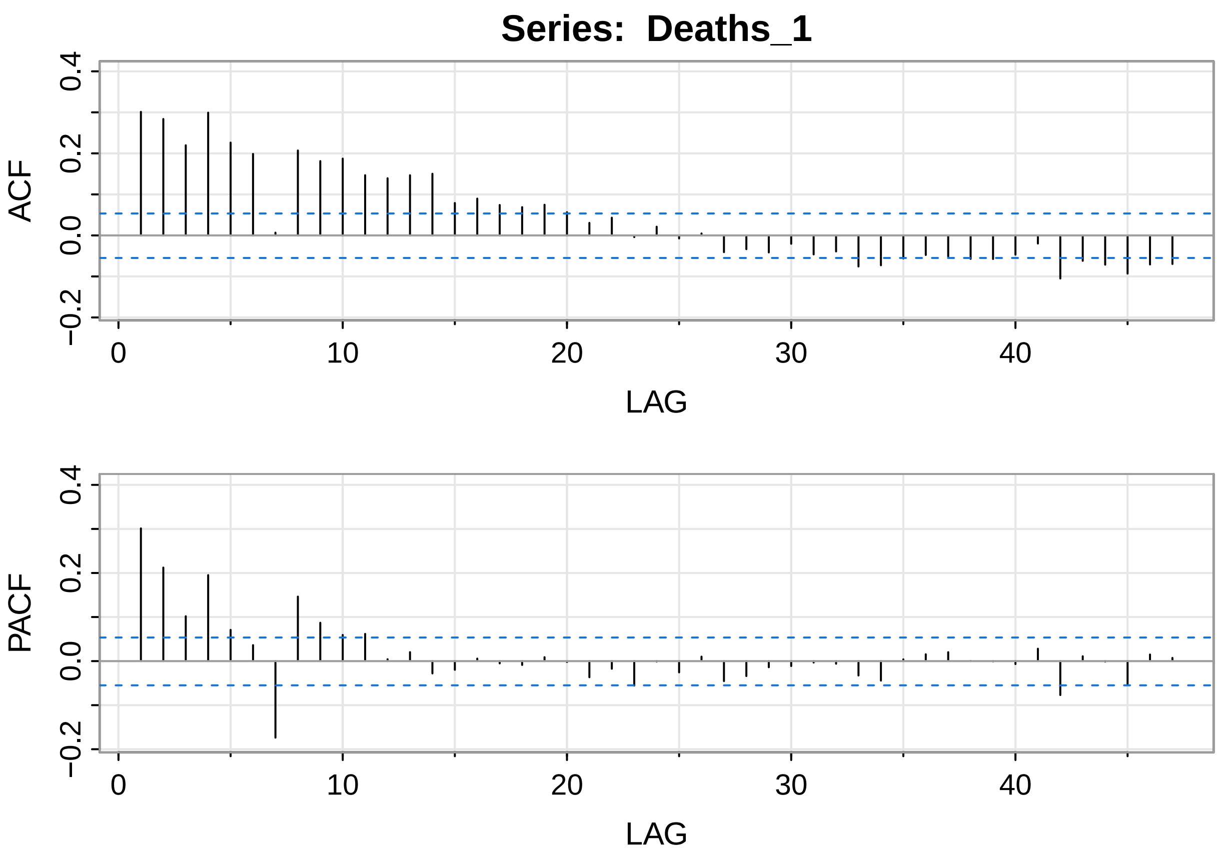
```
abline(h=mean(Deaths.sqrt_1), col= "blue")
abline(lm(Deaths.sqrt_1 ~ as.numeric(1:length(Deaths.sqrt_1))), col="red")
```



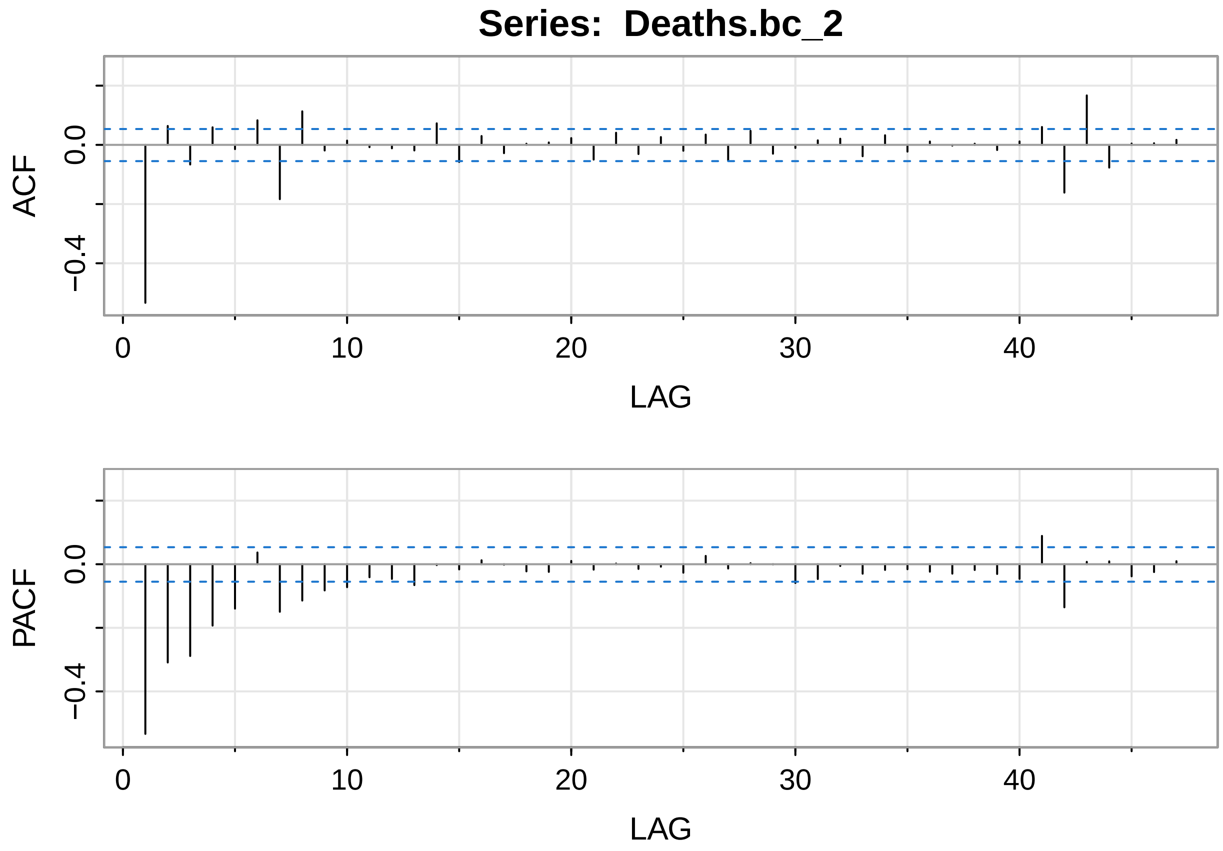
```
par(mfrow = op$mfrow, mar = op$mar)

# Histograms
op <- par("mfrow", "mar")
par(mfrow = c(2, 2))
hist(Deaths_1, col="light blue", xlab="", main="Original_1", probability = TRUE)
m <-mean(Deaths_1)
std <- sqrt(var(Deaths_1))
curve(dnorm(x,m,std), add=TRUE )
hist(Deaths.bc_2, col="light blue", xlab="", main="(Box-Cox)_2", probability = TRUE)
m <-mean(Deaths.bc_2)
std <- sqrt(var(Deaths.bc_2))
curve(dnorm(x,m,std), add=TRUE )
hist(Deaths.log_2, col="light blue", xlab="", main="(Log Transformed)_2", probability = TRUE)
m <-mean(Deaths.log_2)
std <- sqrt(var(Deaths.log_2))
curve(dnorm(x,m,std), add=TRUE )
hist(Deaths.sqrt_1, col="light blue", xlab="", main="(Square Root Transformed)_1", probability = TRUE)
m <-mean(Deaths.sqrt_1)
std <- sqrt(var(Deaths.sqrt_1))
curve(dnorm(x,m,std), add=TRUE )
```



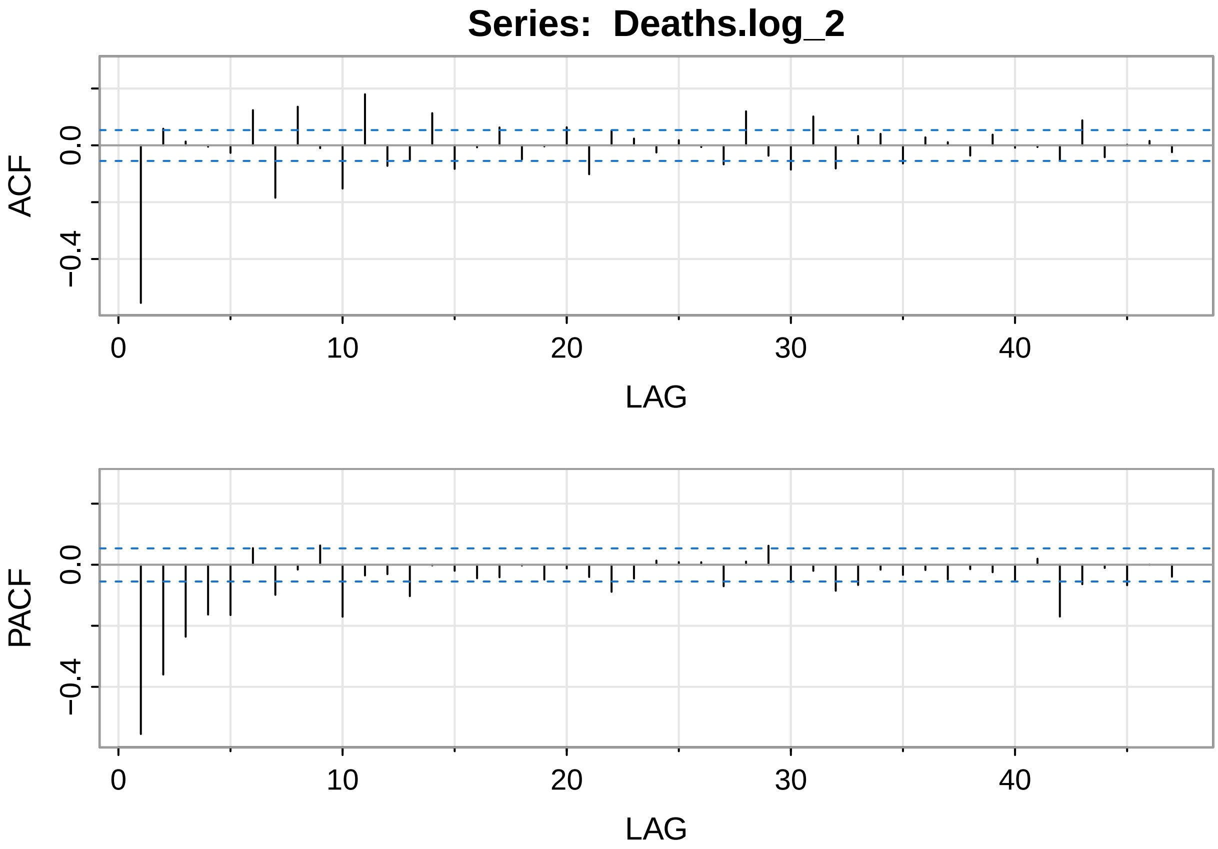
```
par(mfrow = op$mfrow, mar = op$mar)

# ACF & PACF
x = acf2(Deaths_1)
```



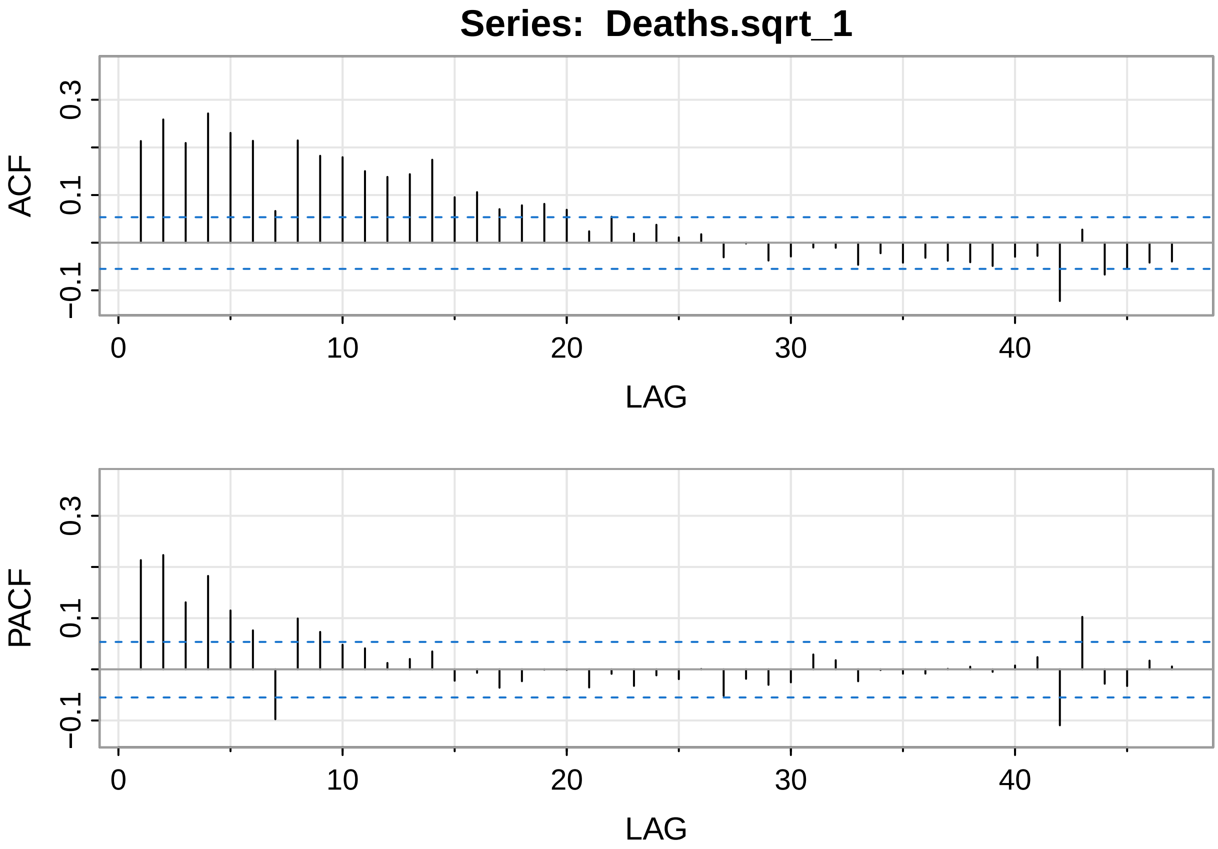
```
x = acf2(Deaths.bc_2)
```



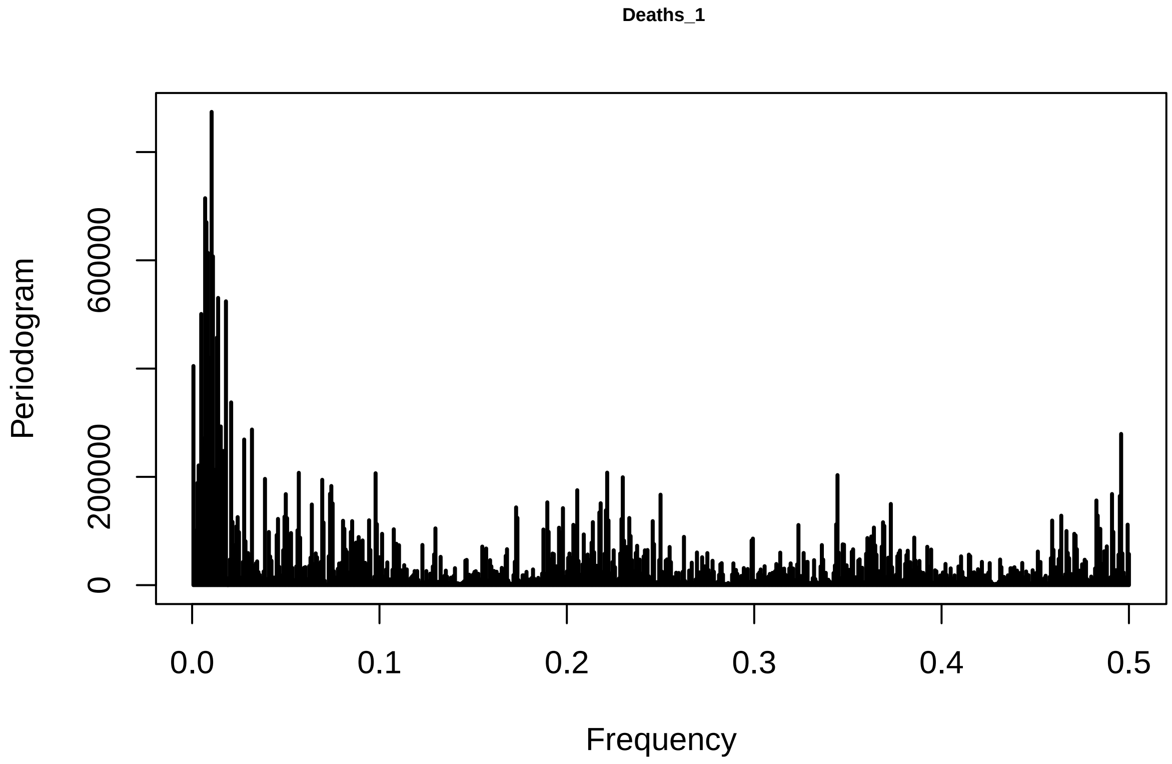
```
x = acf2(Deaths.log_2)
```



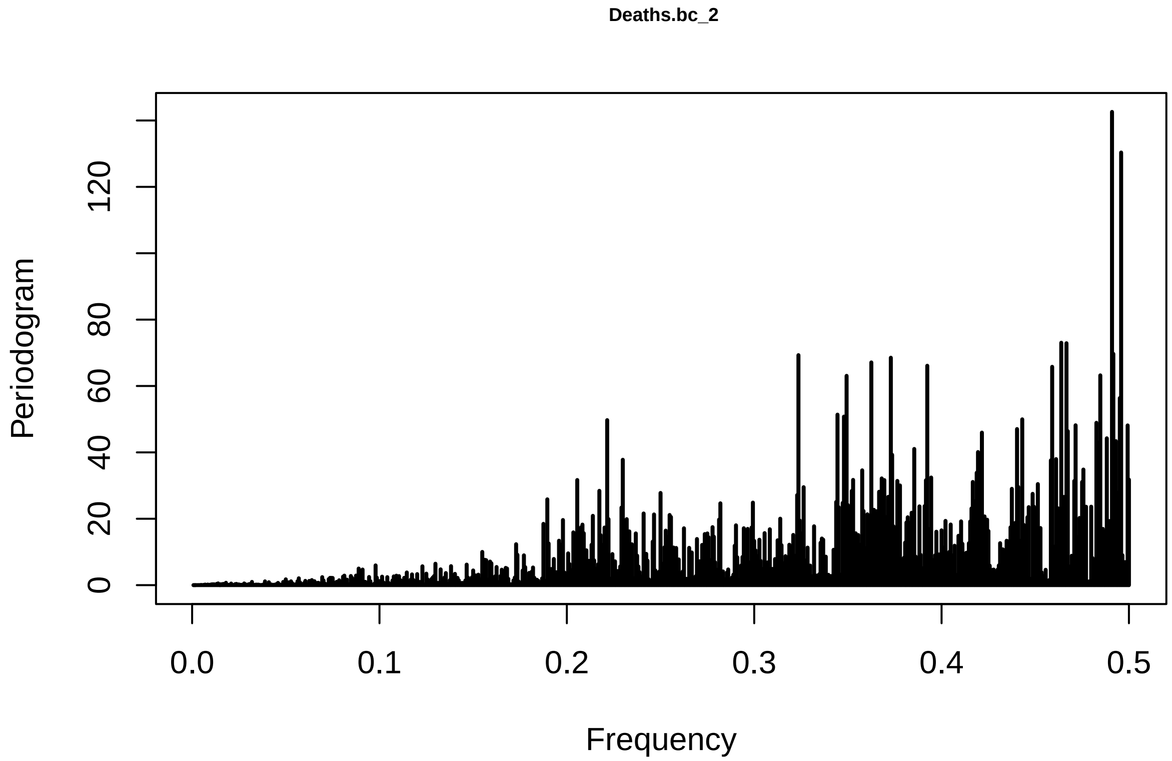
```r
x = acf2(Deaths.sqrt_1)
```



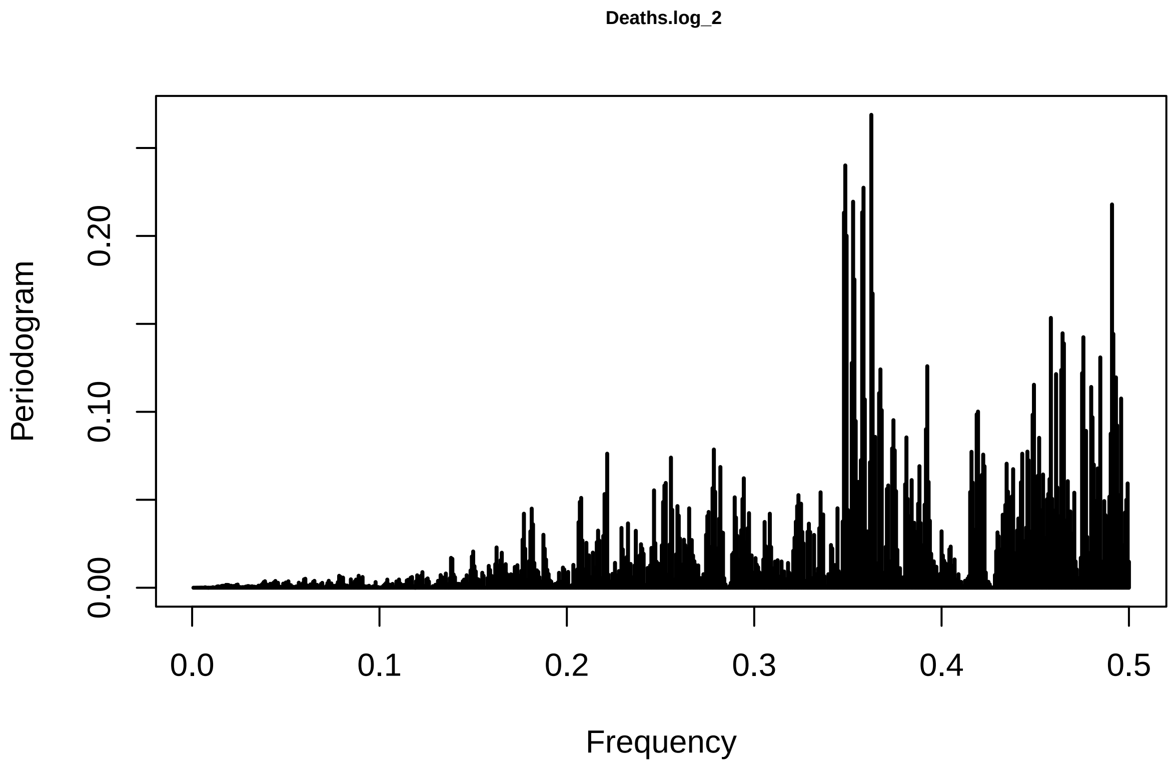
```
# Spectral Analysis using Periodograms
gram = TSA::periodogram(Deaths_1, main=paste0(" Deaths_1"), cex.main=0.6)
```



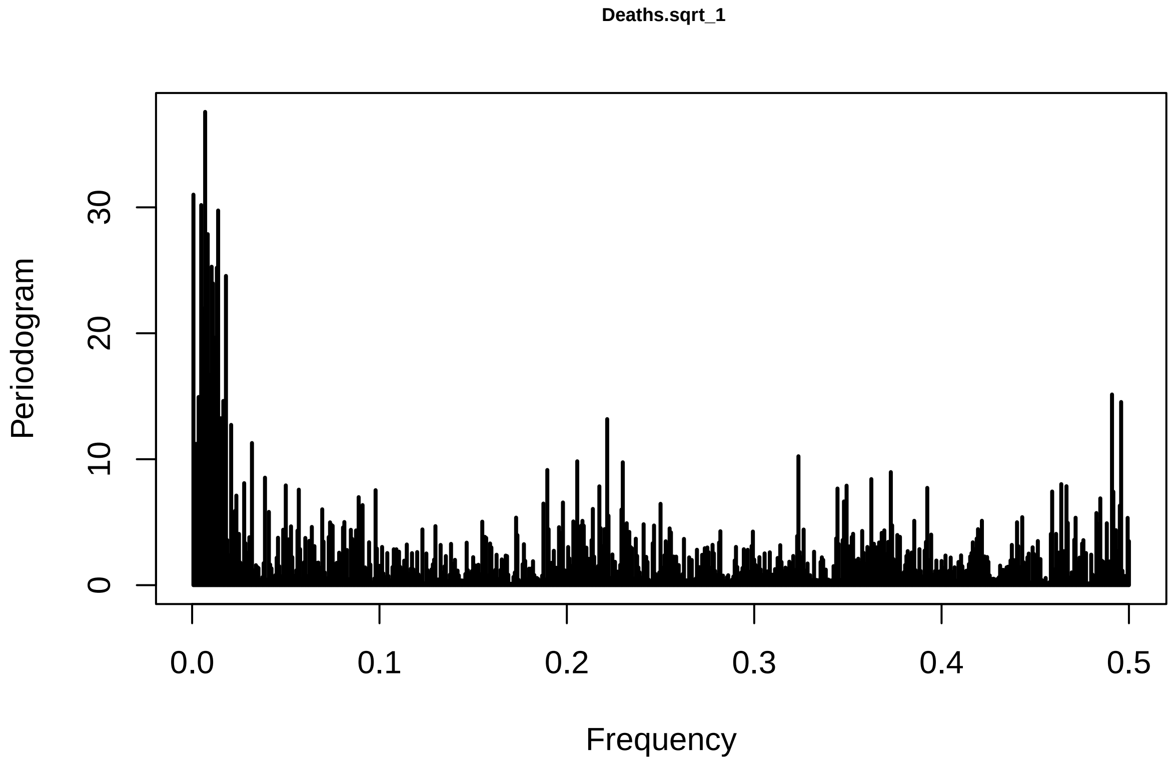
```
gram.bc = TSA::periodogram(Deaths.bc_2, main=paste0(" Deaths.bc_2"), cex.main=0.6)
```



```r
gram.log = TSA::periodogram(Deaths.log_2, main=paste0(" Deaths.log_2"), cex.main=0.6)
```



```
gram.sqrt = TSA::periodogram(Deaths.sqrt_1, main=paste0(" Deaths.sqrt_1"), cex.main=0.6)
```



```
1/gram$freq[which.max(gram$spec)]
```

```
## [1] 96
```

```
1/gram.bc$freq[which.max(gram.bc$spec)]
```

```
## [1] 2.036775
```

```
1/gram.log$freq[which.max(gram.log$spec)]
```

```
## [1] 2.758621
```

```
1/gram.sqrt$freq[which.max(gram.sqrt$spec)]
```

```
## [1] 144
```

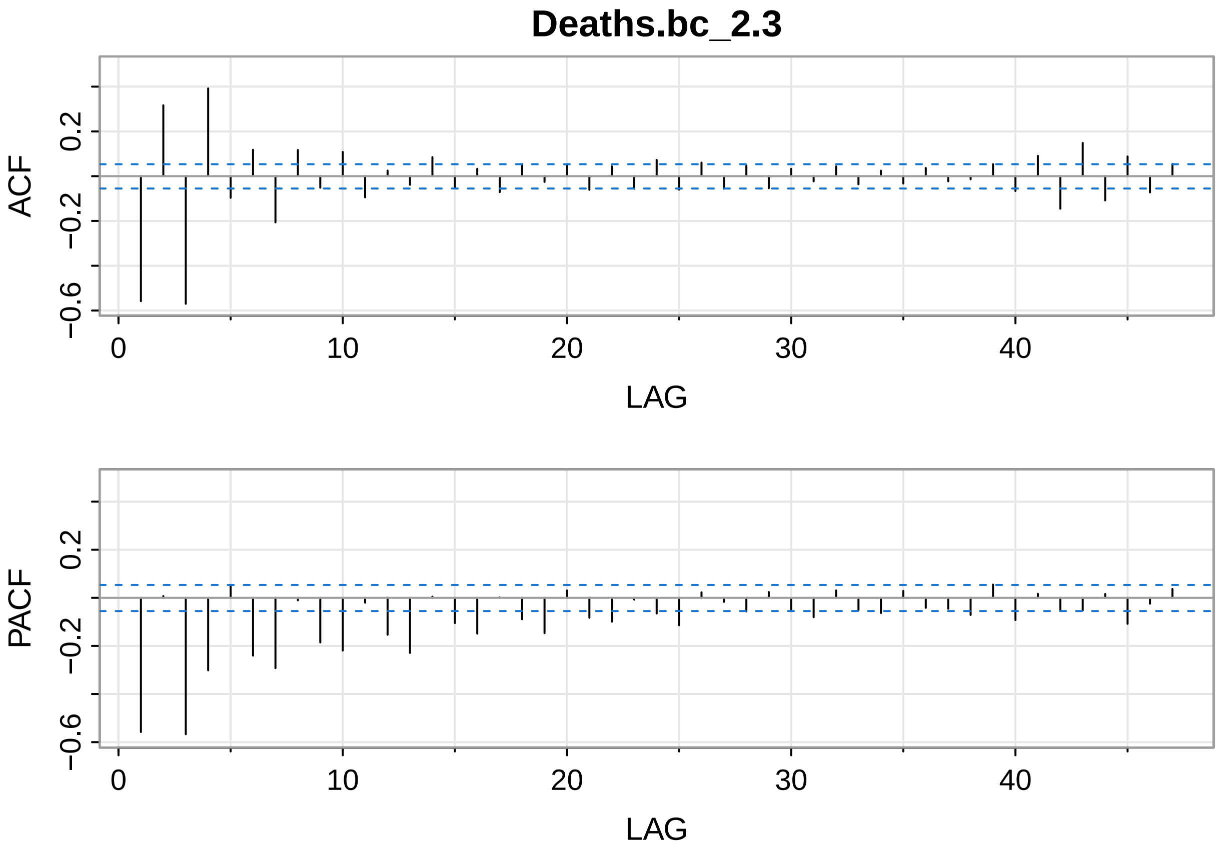
```
try(acf2(arima(Deaths_1, order = c(0,0,0), seasonal = list(order=c(0,1,0), period=96),
                   method="ML")$residuals, max.lag = 40, main = "Deaths_1.96", j[b]))
```

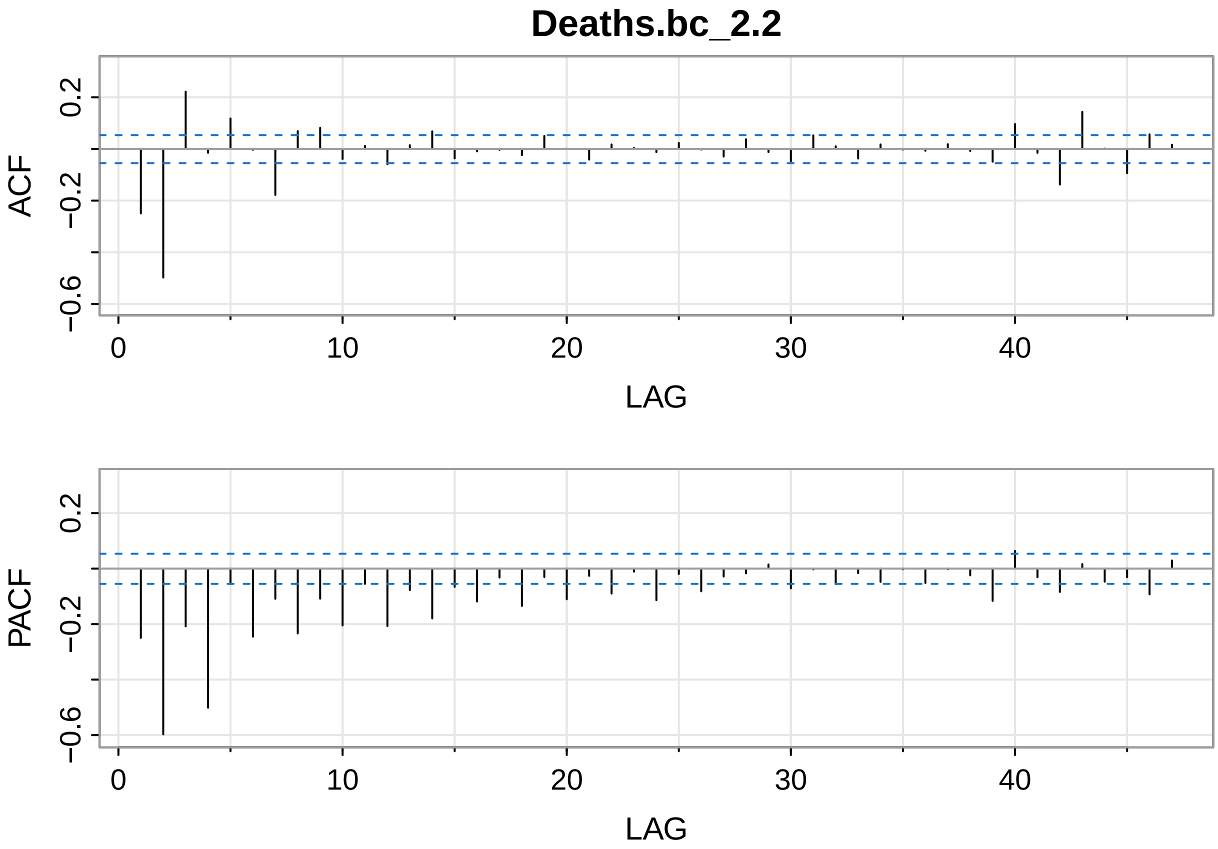
```
## Error in eval(expr, envir, enclos) : object 'j' not found
```

```
frame = data.frame(matrix(nrow=720,ncol=3))
colnames(frame) = c("spec", "freq", "1/freq")
frame$spec = gram.bc$spec
frame$freq = gram.bc$freq
frame$`1/freq` = 1/gram$freq
frame = subset(frame, frame$spec >= 50)
frame
```

```
##           spec      freq   1/freq
## 466  69.28775 0.3236111 3.090129
## 496  51.36836 0.3444444 2.903226
## 501  50.77615 0.3479167 2.874251
## 503  63.05536 0.3493056 2.862823
## 522  67.10312 0.3625000 2.758621
## 537  68.51522 0.3729167 2.681564
## 565  66.09570 0.3923611 2.548673
## 661  65.77668 0.4590278 2.178517
## 668  73.02966 0.4638889 2.155689
## 672  72.87922 0.4666667 2.142857
## 698  63.20530 0.4847222 2.063037
## 707 142.58092 0.4909722 2.036775
## 708  69.68472 0.4916667 2.033898
## 713  56.36836 0.4951389 2.019635
## 714 130.34807 0.4958333 2.016807
```

```
j = unique(round(frame$`1/freq`, 0))

for(b in 1:length(j)){
    try(acf2(arima(Deaths.bc_2, order = c(0,0,0), seasonal = list(order=c(0,1,0), period=j[b]),
                   method="ML")$residuals, main = paste0("Deaths.bc_2.", j[b])))}
```

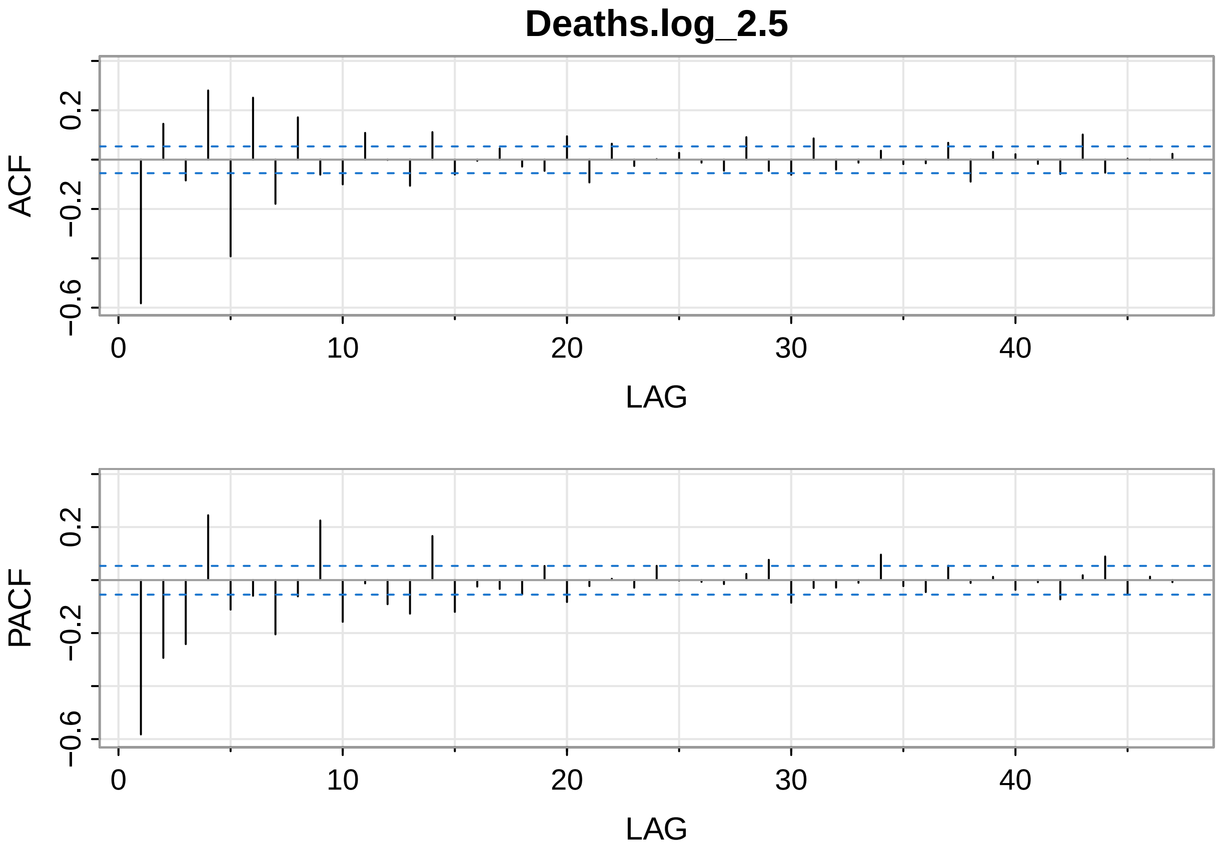


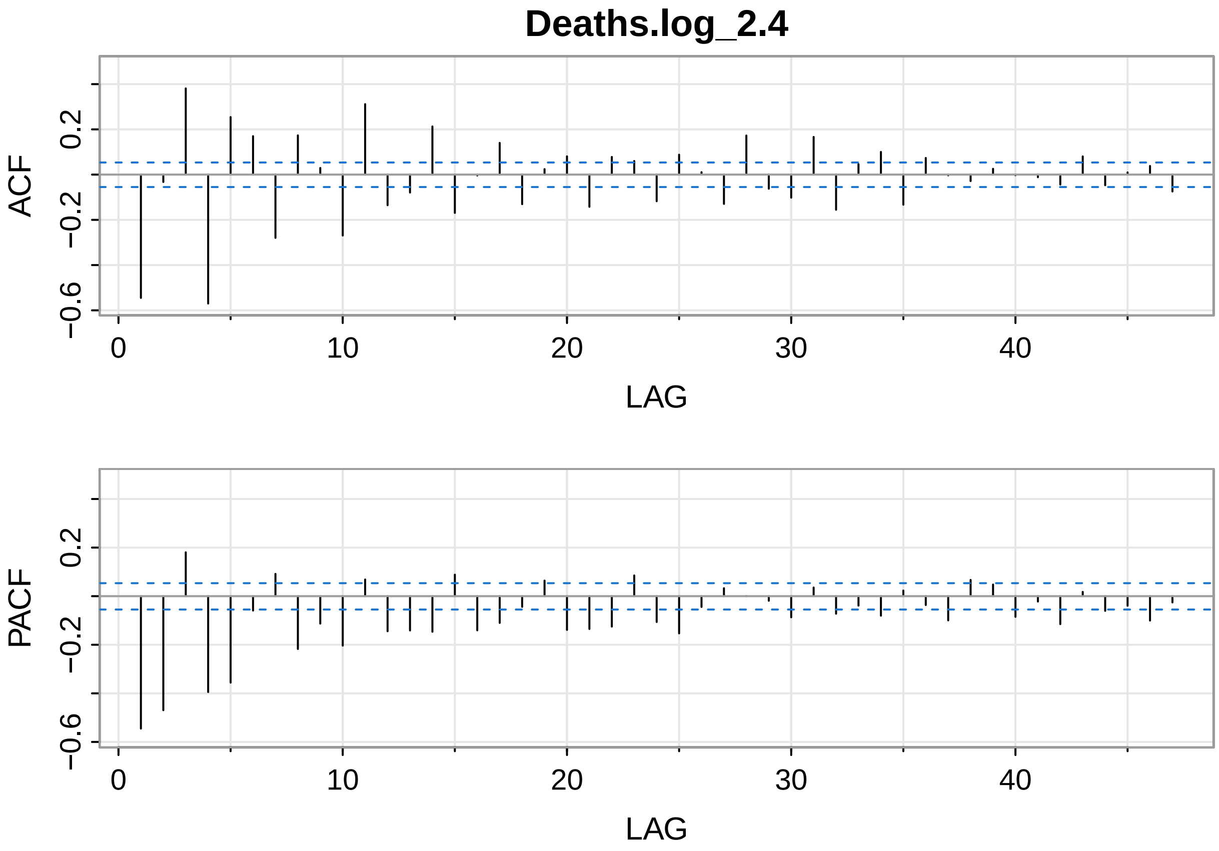


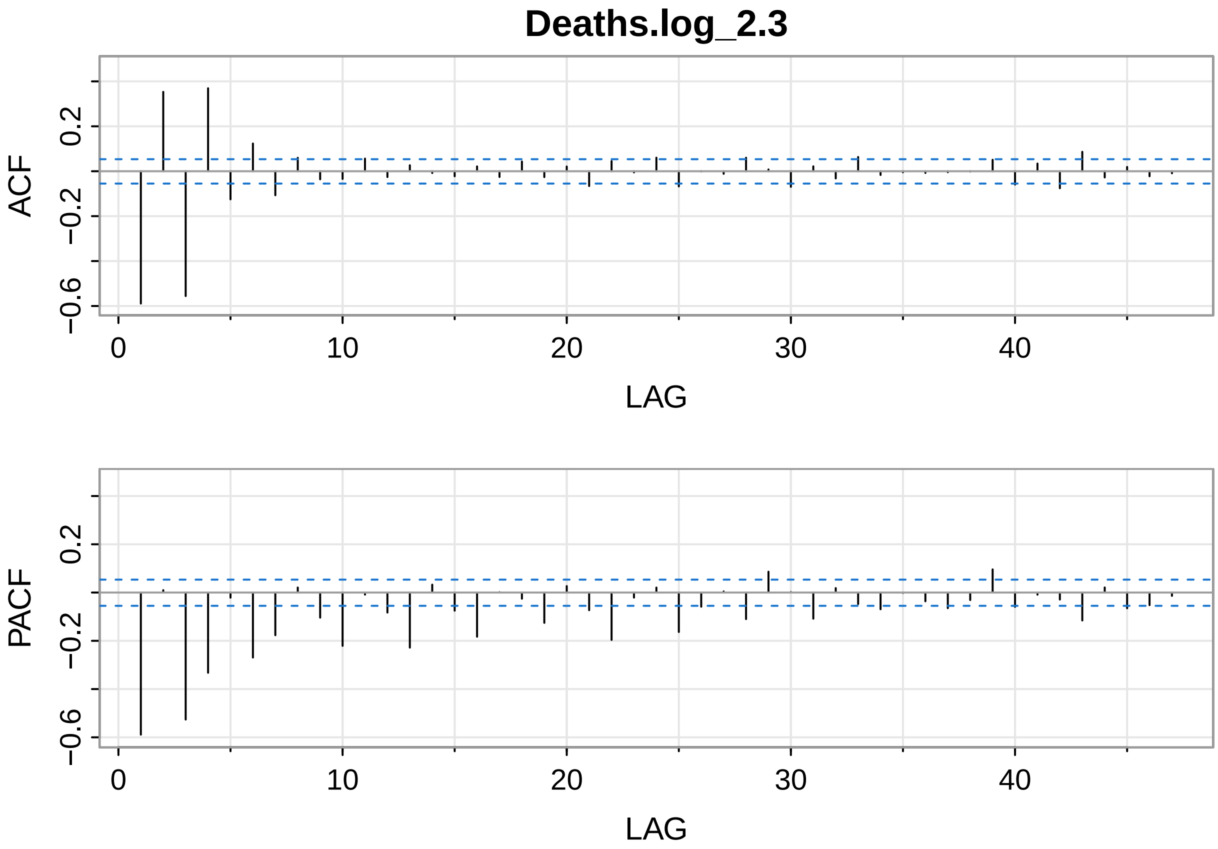
```
frame = data.frame(matrix(nrow=720,ncol=3))
colnames(frame) = c("spec", "freq", "1/freq")
frame$spec = gram.log$spec
frame$freq = gram.log$freq
frame$`1/freq` = 1/gram$freq
frame = subset(frame, frame$spec >= 0.05)
frame
```

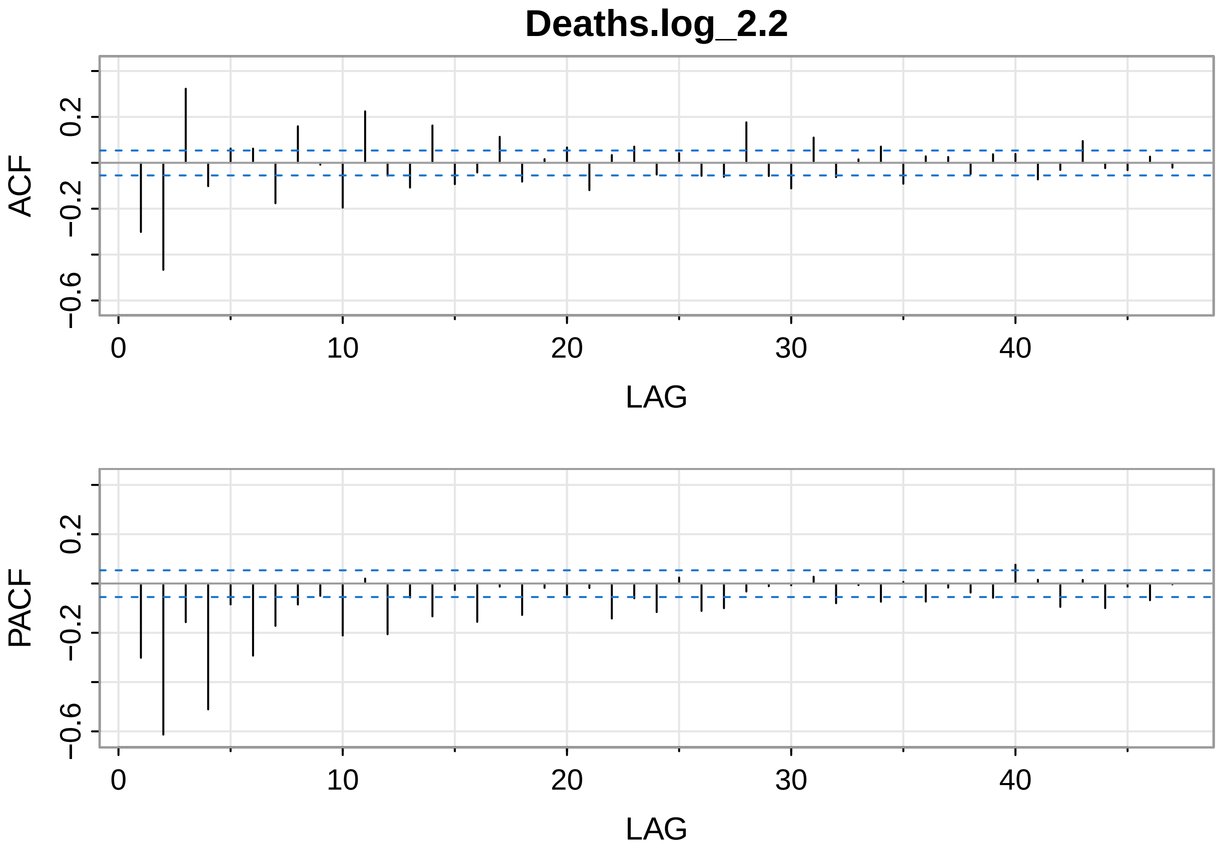
```
##           spec      freq   1/freq
## 299 0.05106505 0.2076389 4.816054
## 317 0.05325488 0.2201389 4.542587
## 319 0.07620465 0.2215278 4.514107
## 355 0.05536474 0.2465278 4.056338
## 363 0.05809710 0.2520833 3.966942
## 364 0.05957197 0.2527778 3.956044
## 368 0.07397001 0.2555556 3.913043
## 400 0.05663184 0.2777778 3.600000
## 401 0.07857096 0.2784722 3.591022
## 402 0.05452280 0.2791667 3.582090
## 406 0.06861661 0.2819444 3.546798
## 417 0.05139270 0.2895833 3.453237
## 423 0.05044458 0.2937500 3.404255
## 424 0.06222021 0.2944444 3.396226
## 466 0.05264773 0.3236111 3.090129
## 483 0.05423130 0.3354167 2.981366
## 501 0.21319562 0.3479167 2.874251
## 502 0.24007839 0.3486111 2.868526
## 503 0.20010927 0.3493056 2.862823
## 507 0.12776342 0.3520833 2.840237
## 508 0.21944209 0.3527778 2.834646
## 509 0.17523928 0.3534722 2.829077
## 510 0.09478028 0.3541667 2.823529
## 512 0.06016910 0.3555556 2.812500
## 514 0.07254093 0.3569444 2.801556
## 515 0.21343016 0.3576389 2.796117
## 516 0.22740617 0.3583333 2.790698
## 517 0.10701648 0.3590278 2.785300
## 521 0.07135805 0.3618056 2.763916
## 522 0.26883309 0.3625000 2.758621
## 523 0.16723548 0.3631944 2.753346
## 524 0.07076799 0.3638889 2.748092
## 525 0.08575998 0.3645833 2.742857
## 528 0.11063321 0.3666667 2.727273
## 529 0.12411918 0.3673611 2.722117
## 530 0.10092462 0.3680556 2.716981
## 534 0.05618377 0.3708333 2.696629
## 535 0.05812753 0.3715278 2.691589
## 538 0.07921302 0.3736111 2.676580
## 539 0.09526166 0.3743056 2.671614
## 540 0.07812888 0.3750000 2.666667
## 541 0.05492457 0.3756944 2.661738
## 548 0.05881394 0.3805556 2.627737
## 549 0.08544492 0.3812500 2.622951
## 550 0.05047880 0.3819444 2.618182
## 553 0.06119260 0.3840278 2.603978
## 559 0.06910676 0.3881944 2.576029
## 564 0.09010523 0.3916667 2.553191
## 565 0.12593097 0.3923611 2.548673
## 566 0.06015490 0.3930556 2.544170
## 598 0.05440091 0.4152778 2.408027
## 599 0.07726458 0.4159722 2.404007
## 600 0.05972216 0.4166667 2.400000
## 603 0.09858863 0.4187500 2.388060
## 604 0.10013754 0.4194444 2.384106
## 607 0.06415191 0.4215278 2.372323
## 608 0.07569039 0.4222222 2.368421
## 609 0.06905550 0.4229167 2.364532
## 626 0.07051744 0.4347222 2.300319
## 627 0.05463705 0.4354167 2.296651
## 630 0.05189873 0.4375000 2.285714
## 631 0.06740910 0.4381944 2.282092
## 637 0.05991938 0.4423611 2.260597
## 638 0.07619164 0.4430556 2.257053
## 642 0.07738667 0.4458333 2.242991
## 643 0.07238095 0.4465278 2.239502
## 646 0.09836250 0.4486111 2.229102
## 647 0.11539180 0.4493056 2.225657
## 650 0.06364767 0.4513889 2.215385
## 651 0.08523048 0.4520833 2.211982
## 654 0.06438113 0.4541667 2.201835
## 657 0.05026326 0.4562500 2.191781
## 658 0.05334160 0.4569444 2.188450
## 659 0.06176402 0.4576389 2.185129
## 660 0.15339153 0.4583333 2.181818
## 661 0.05053828 0.4590278 2.178517
## 664 0.12143942 0.4611111 2.168675
## 665 0.05697151 0.4618056 2.165414
## 668 0.12378434 0.4638889 2.155689
## 669 0.14465302 0.4645833 2.152466
## 670 0.13886215 0.4652778 2.149254
## 673 0.06059486 0.4673611 2.139673
## 678 0.05401255 0.4708333 2.123894
## 684 0.12197554 0.4750000 2.105263
## 685 0.14239918 0.4756944 2.102190
## 687 0.08918566 0.4770833 2.096070
## 691 0.11412585 0.4798611 2.083936
## 692 0.09688214 0.4805556 2.080925
## 693 0.07000733 0.4812500 2.077922
## 696 0.06782633 0.4833333 2.068966
## 698 0.13092692 0.4847222 2.063037
## 705 0.05176964 0.4895833 2.042553
## 706 0.08758266 0.4902778 2.039660
## 707 0.21788949 0.4909722 2.036775
## 708 0.14429479 0.4916667 2.033898
## 709 0.07290573 0.4923611 2.031030
## 710 0.11956783 0.4930556 2.028169
## 711 0.09219008 0.4937500 2.025316
## 713 0.05268470 0.4951389 2.019635
## 714 0.10754628 0.4958333 2.016807
## 719 0.05927750 0.4993056 2.002782
```

```r
j = unique(round(frame$`1/freq`, 0))

for(b in 1:length(j)){
    try(acf2(arima(Deaths.log_2, order = c(0,0,0), seasonal = list(order=c(0,1,0), period=j[b]),
                   method="ML")$residuals, main = paste0("Deaths.log_2.", j[b])))}
```

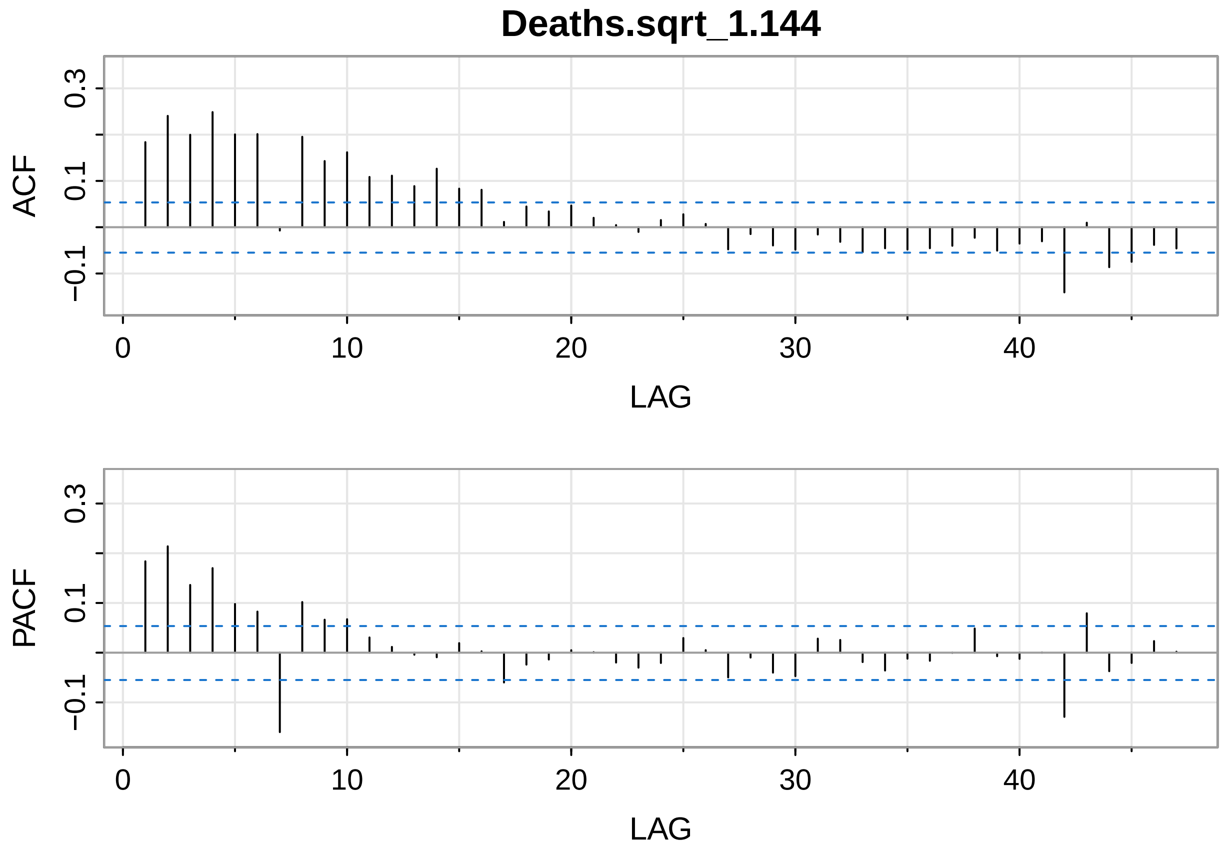

Deaths.log_2.5
ACF
0.2
−0.2
−0.6
0
10
20
30
40
LAG
PACF
0.2
−0.2
−0.6
0
10
20
30
40
LAG


Deaths.log_2.4
ACF
0.2
−0.2
−0.6
0
10
20
30
40
LAG
PACF
0.2
−0.2
−0.6
0
10
20
30
40
LAG


Deaths.log_2.3
ACF
0.2
−0.2
−0.6
0
10
20
30
40
LAG
PACF
0.2
−0.2
−0.6
0
10
20
30
40
LAG



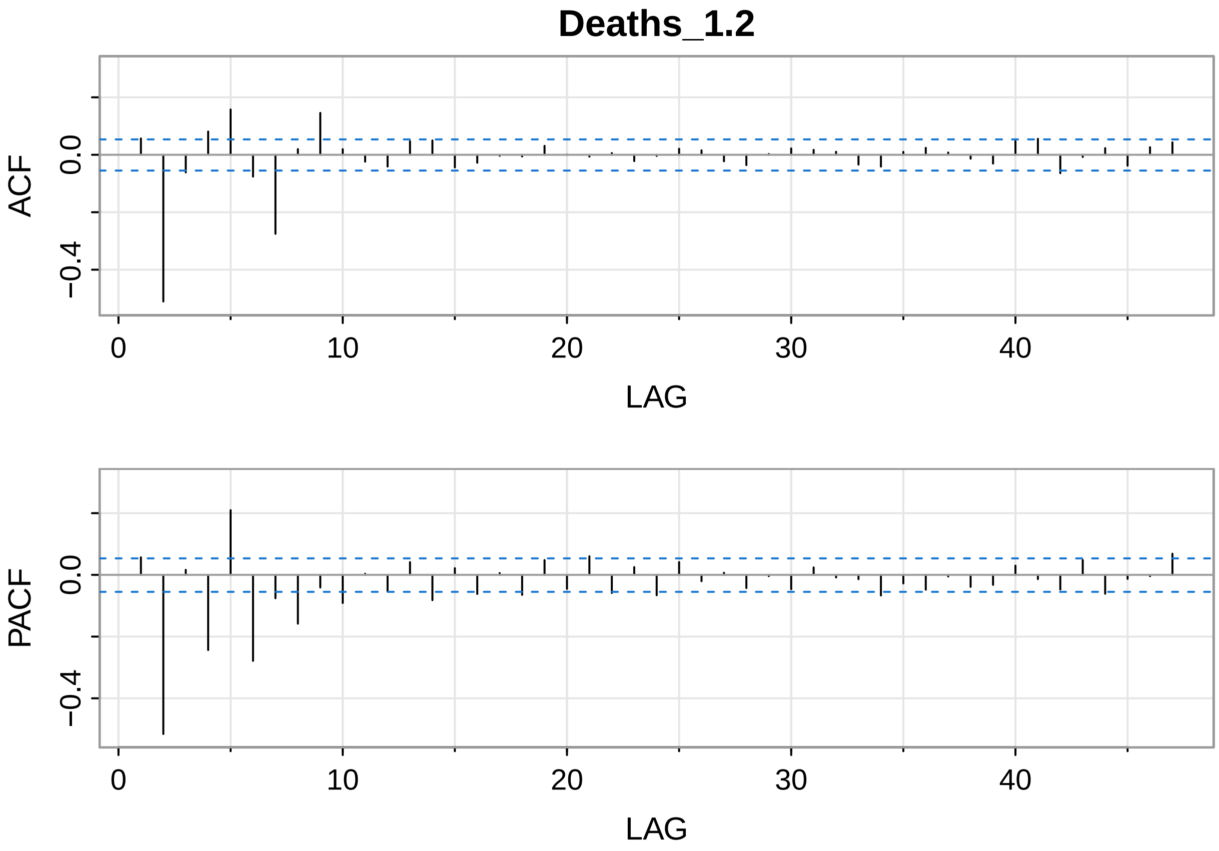
```
try(acf2(arima(Deaths.sqrt_1, order = c(0,0,0), seasonal = list(order=c(0,1,0), period=144),
                   method="ML")$residuals, main = "Deaths.sqrt_1.144"))
```



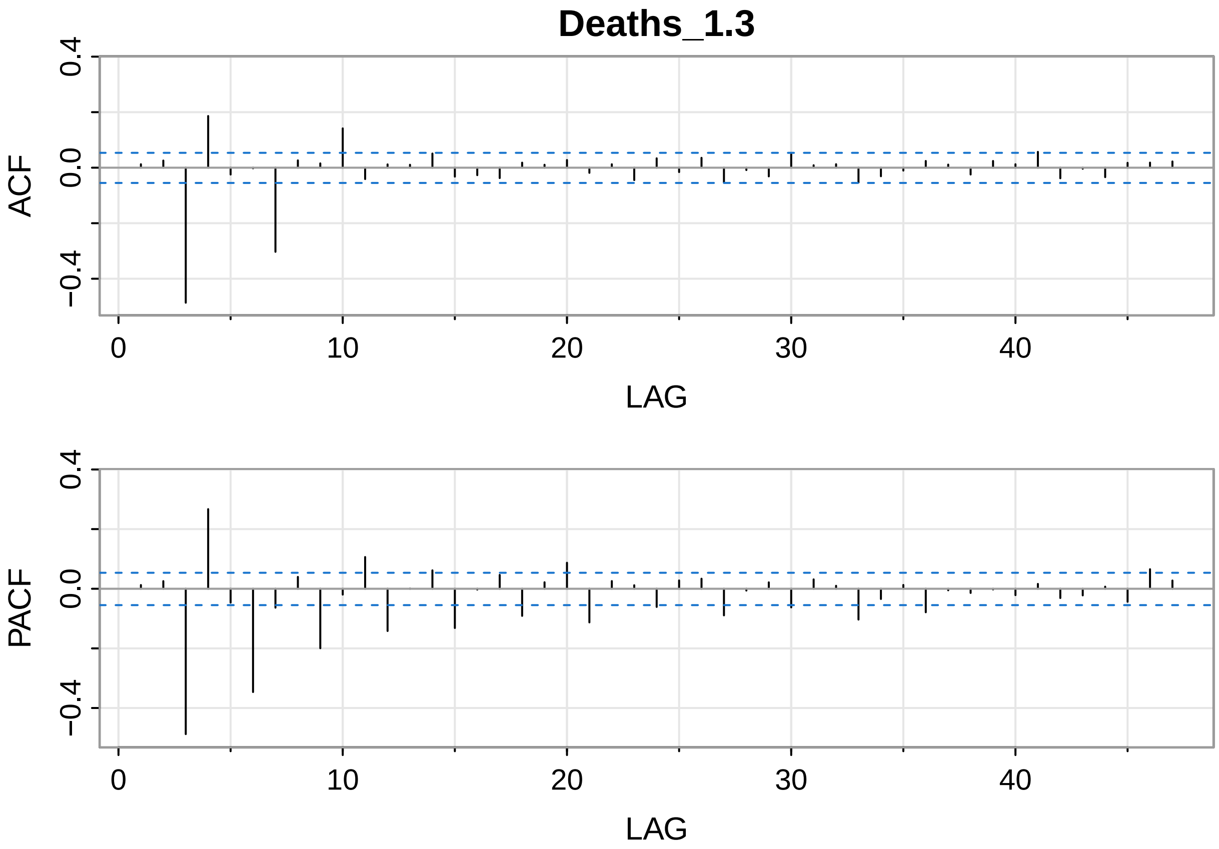
```
##      [,1] [,2] [,3] [,4] [,5] [,6]  [,7] [,8] [,9] [,10] [,11] [,12] [,13]
## ACF  0.18 0.24 0.20 0.25  0.2 0.20 -0.01  0.2 0.14  0.16  0.11  0.11  0.09
## PACF 0.18 0.21 0.14 0.17  0.1 0.08 -0.16  0.1 0.07  0.07  0.03  0.01  0.00
##      [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24] [,25]
## ACF   0.13  0.08  0.08  0.01  0.04  0.03  0.05  0.02  0.00 -0.01  0.02  0.03
## PACF -0.01  0.02  0.00 -0.06 -0.02 -0.01  0.01  0.00 -0.02 -0.03 -0.02  0.03
##      [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37]
## ACF   0.01 -0.05 -0.01 -0.04 -0.05 -0.02 -0.03 -0.05 -0.05 -0.05 -0.05 -0.04
## PACF  0.01 -0.05 -0.01 -0.04 -0.05  0.03  0.03 -0.02 -0.04 -0.01 -0.02  0.00
##      [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47]
## ACF  -0.02 -0.05 -0.04 -0.03 -0.14  0.01 -0.09 -0.07 -0.04 -0.05
## PACF  0.05 -0.01 -0.01  0.00 -0.13  0.08 -0.04 -0.02  0.02  0.00
```

Periodicity at 2-7 and 96

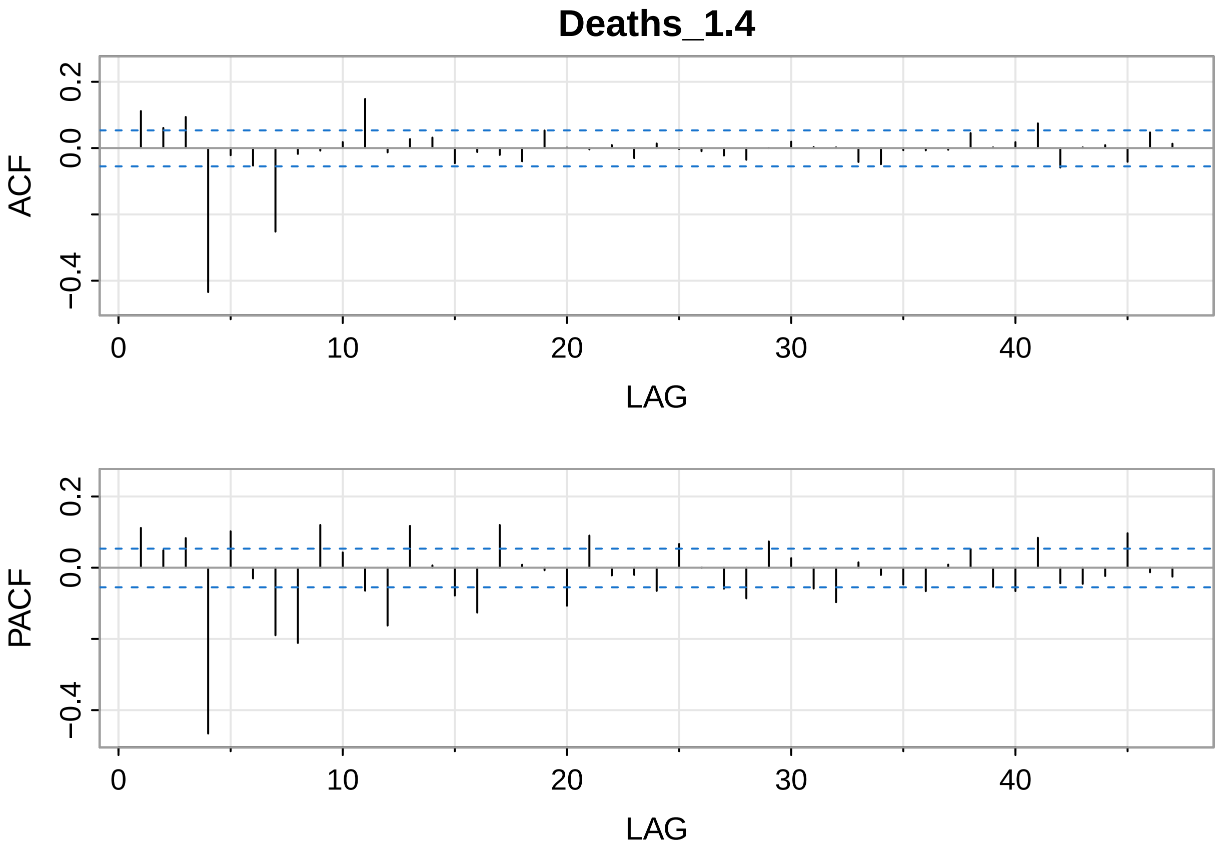
```
Deaths_1.2 <- arima(Deaths_1, order=c(0,0,0), seasonal = list(order=c(0,1,0), period=2),
                    method="ML")$residuals
x = acf2(Deaths_1.2, main = "Deaths_1.2")
```



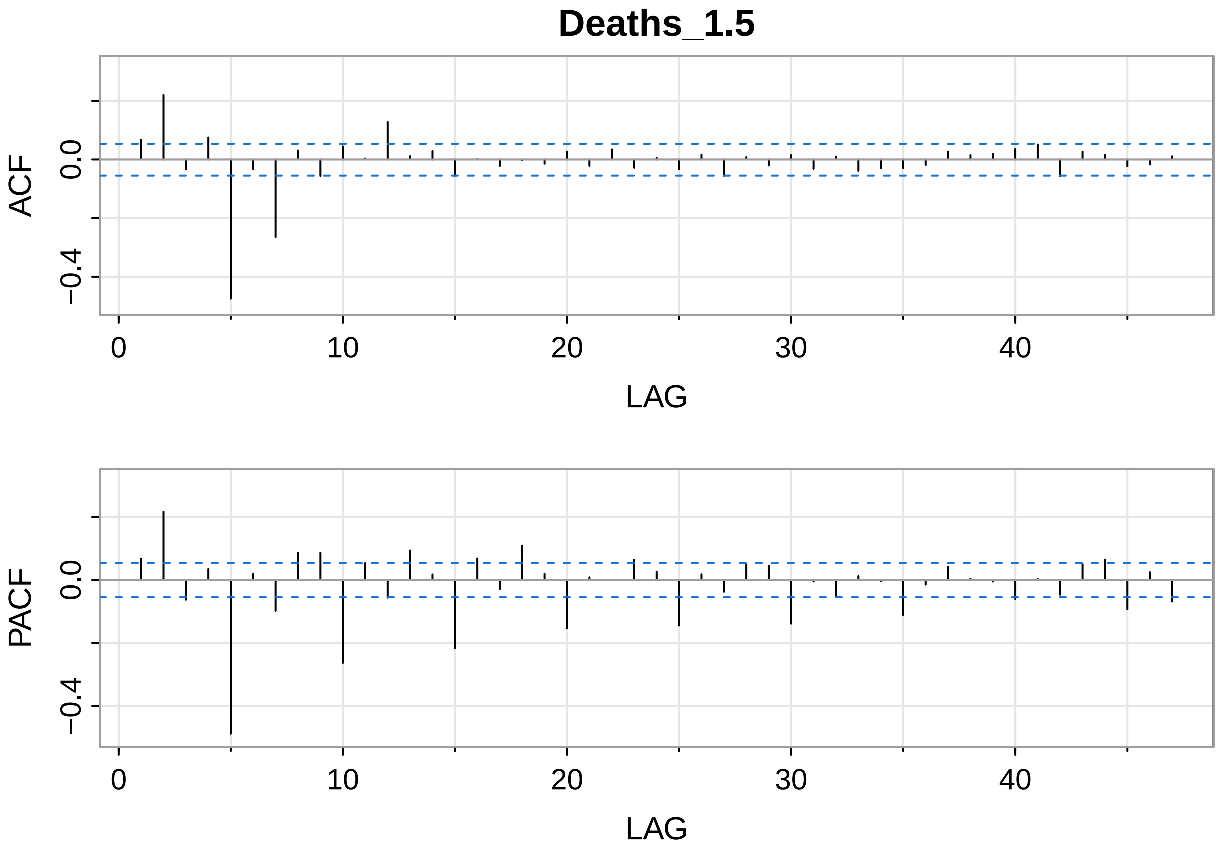
```
Deaths_1.3 <- arima(Deaths_1, order=c(0,0,0), seasonal = list(order=c(0,1,0), period=3),
                   method="ML")$residuals
x = acf2(Deaths_1.3, main = "Deaths_1.3")
```



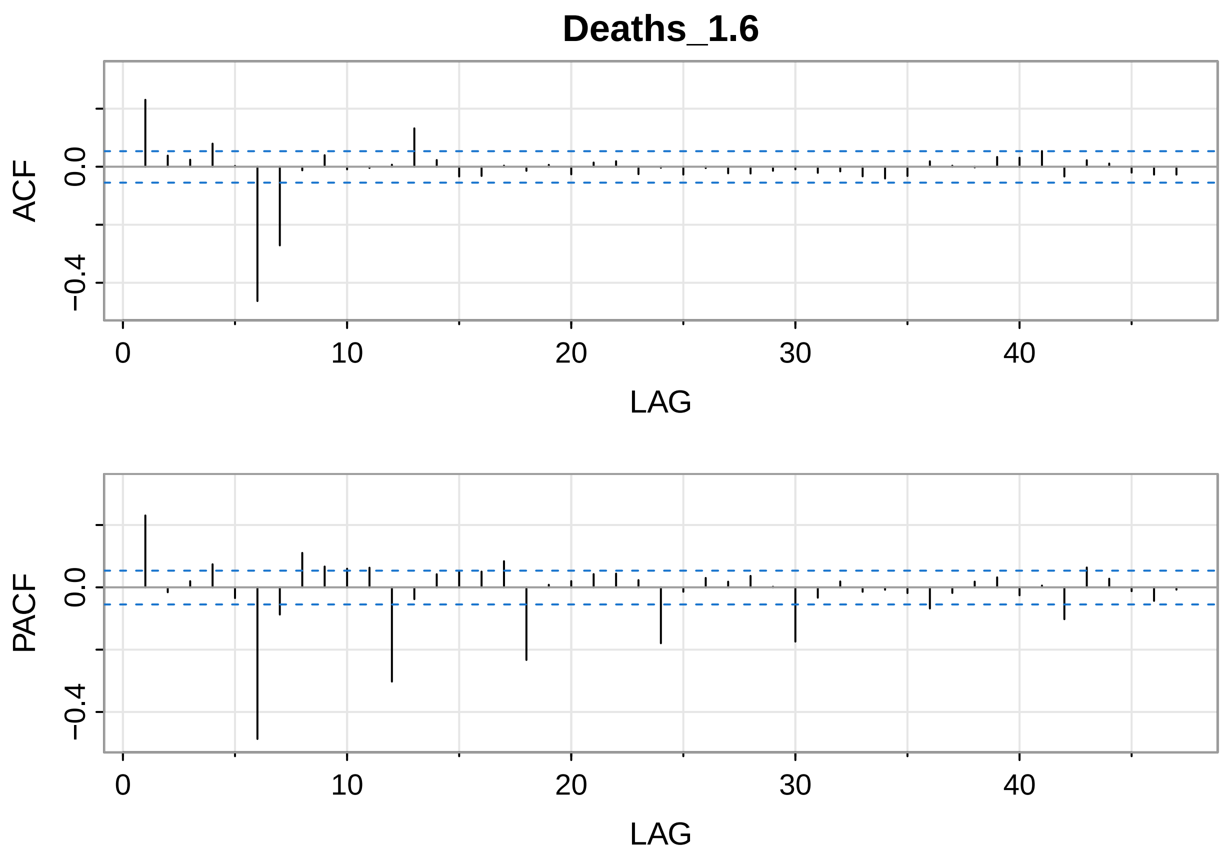
```
Deaths_1.4 <- arima(Deaths_1, order=c(0,0,0), seasonal = list(order=c(0,1,0), period=4),
                    method="ML")$residuals
x = acf2(Deaths_1.4, main = "Deaths_1.4")
```



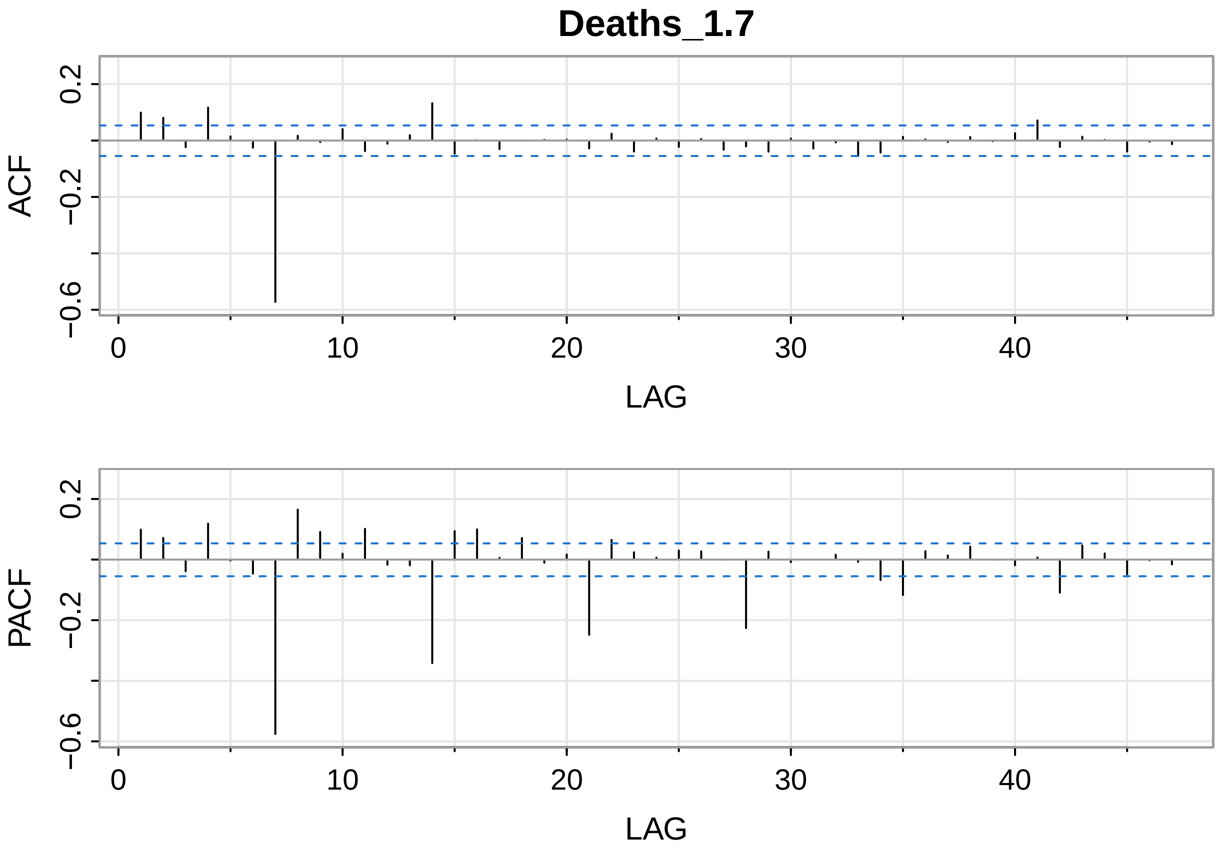
```
Deaths_1.5 <- arima(Deaths_1, order=c(0,0,0), seasonal = list(order=c(0,1,0), period=5),
                    method="ML")$residuals
x = acf2(Deaths_1.5, main = "Deaths_1.5")
```



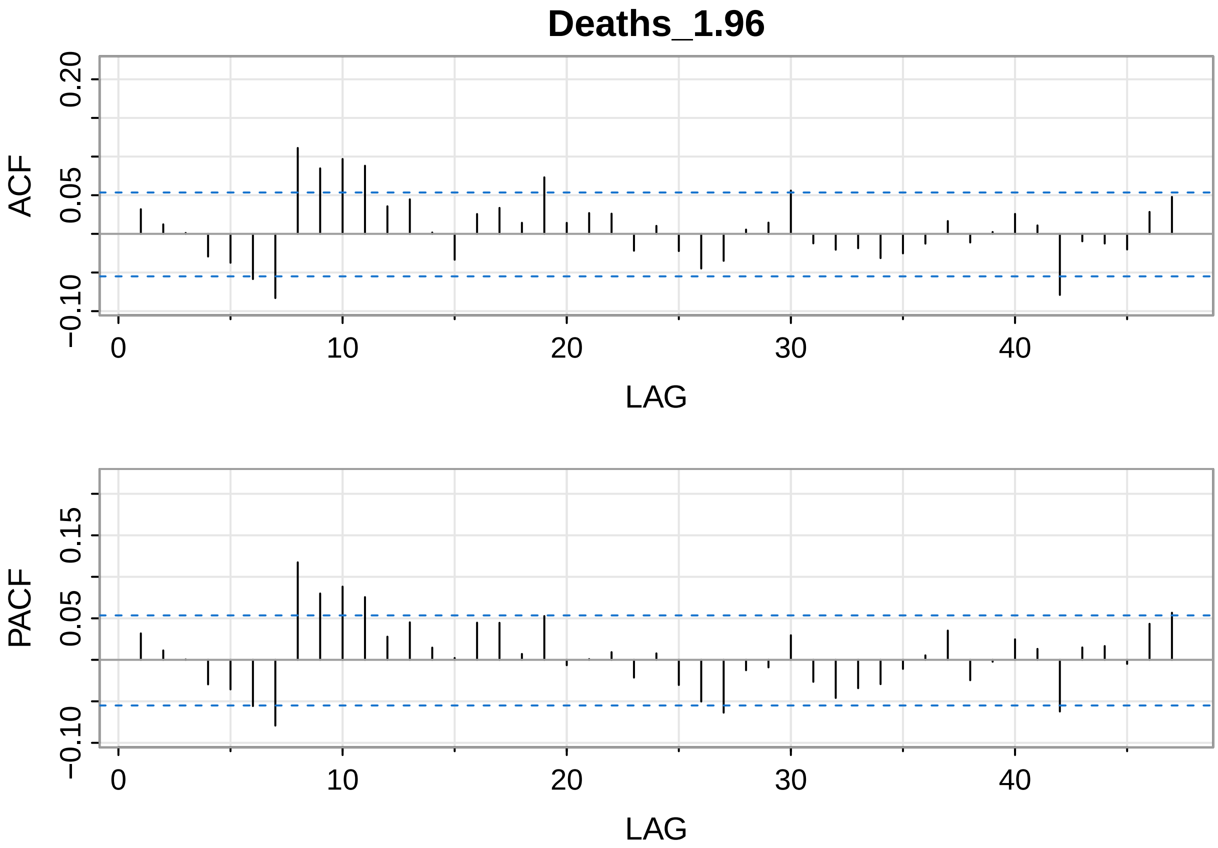
```
Deaths_1.6 <- arima(Deaths_1, order=c(0,0,0), seasonal = list(order=c(0,1,0), period=6),
                    method="ML")$residuals
x = acf2(Deaths_1.6, main = "Deaths_1.6")
```



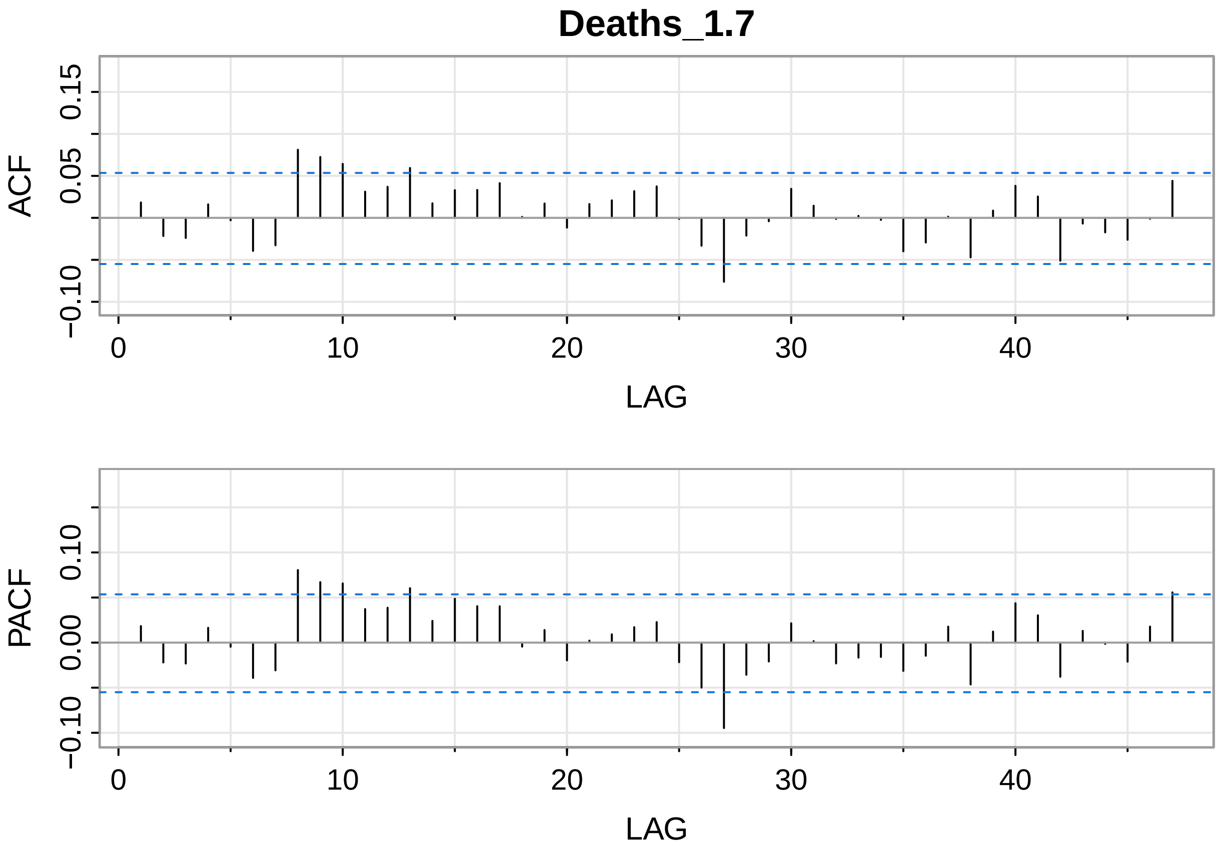
```
Deaths_1.7 <- arima(Deaths_1, order=c(0,0,0), seasonal = list(order=c(0,1,0), period=7),
                    method="ML")$residuals
x = acf2(Deaths_1.7, main = "Deaths_1.7")
```



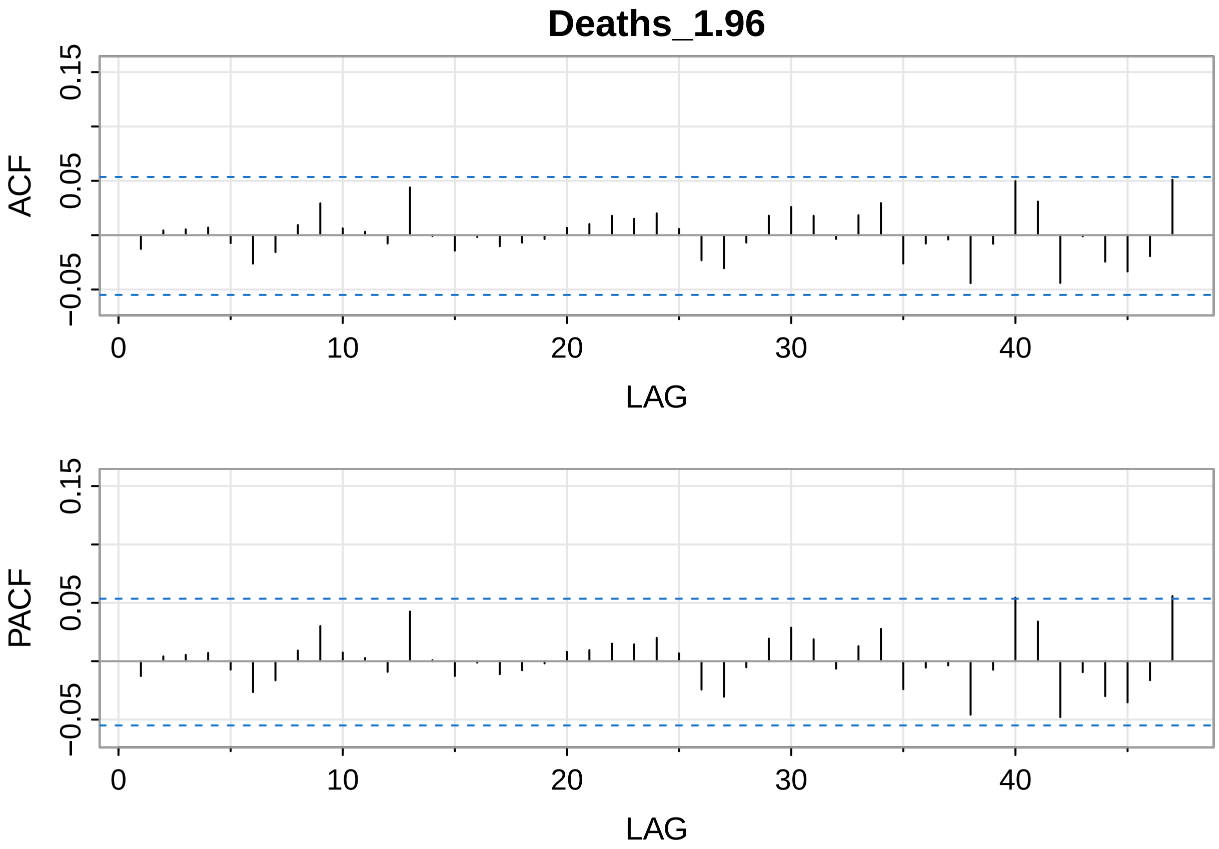
```
Deaths_1.7 <- arima(Deaths_1, order=c(7,0,0), seasonal = list(order=c(0,1,0), period=96),
                    method="ML")$residuals
x = acf2(Deaths_1.7, main = "Deaths_1.96")
```



```
Deaths_1.7 <- arima(Deaths_1.7, order=c(5,0,5),method="ML")$residuals
x = acf2(Deaths_1.7, main = "Deaths_1.7")
```



```
arima.obj <- arima(Deaths_1.7, order=c(8,0,8), method="ML")
x = acf2(arima.obj$residuals, main = "Deaths_1.96")
```



```r
round(ad.test(arima.obj$residuals)$p.value, 5)
```

```
## [1] 0
```

```r
round(shapiro.test(arima.obj$residuals)$p.value, 5)
```

```
## [1] 0
```

```r
round(adf.test(arima.obj$residuals)$p.value, 5)
```

```
## [1] 0.01
```

```r
round(kpss.test(arima.obj$residuals)$p.value, 5)
```

```
## [1] 0.1
```

```r
round(Box.test(arima.obj$residuals, lag=sqrt(length(arima.obj$residuals)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.9997
```

```r
round(Box.test(arima.obj$residuals, lag=sqrt(length(arima.obj$residuals)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

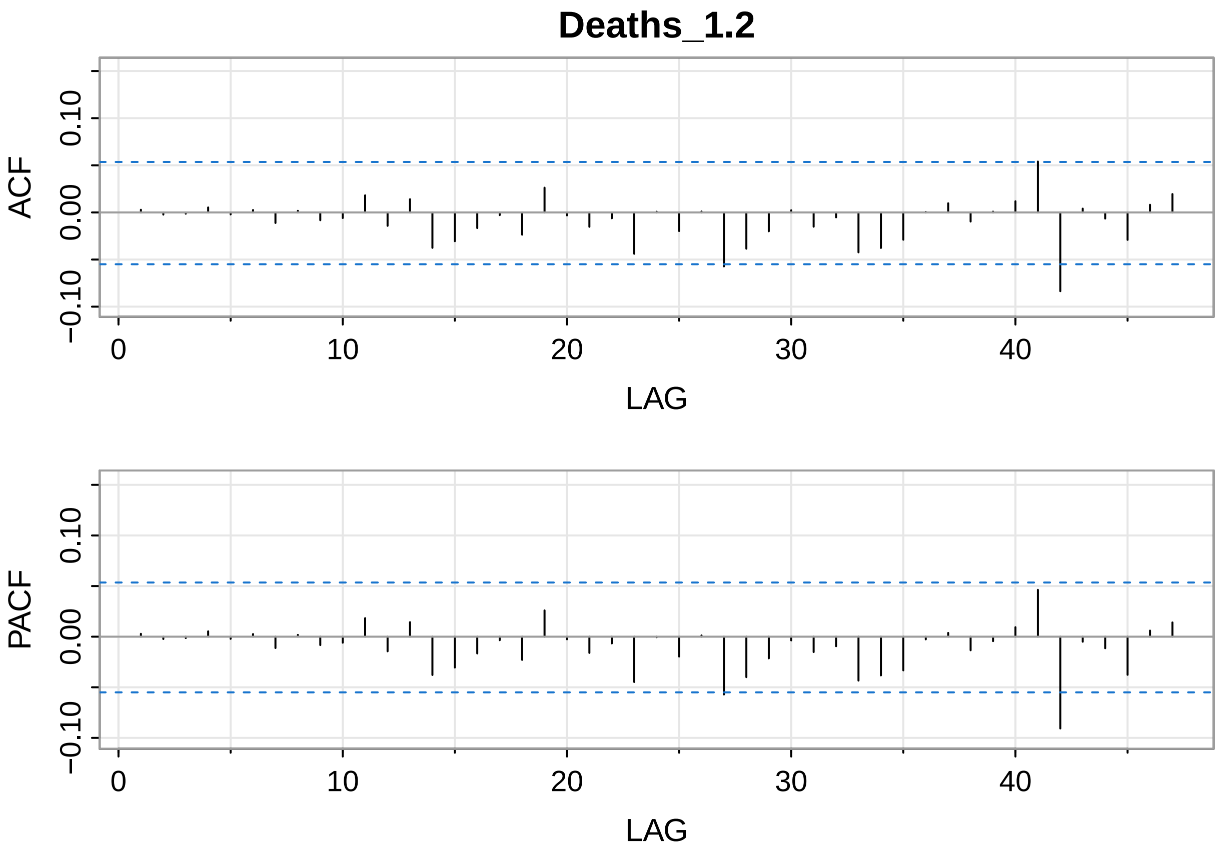
```
## [1] 0.99964
```

```r
round(Box.test((arima.obj$residuals)^2, lag=sqrt(length(arima.obj$residuals)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```

Periodicity of 2

```
Deaths_1.2 <- arima(Deaths_1, order=c(7,0,2), seasonal = list(order=c(0,1,0), period=2),
                    method="ML")$residuals
x = acf2(Deaths_1.2, main = "Deaths_1.2")
```



```
round(ad.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0
```

```
round(shapiro.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0
```

```
round(adf.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0.01
```

```
round(kpss.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0.1
```

```
round(Box.test(Deaths_1.2, lag=sqrt(length(Deaths_1.2)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.96271
```

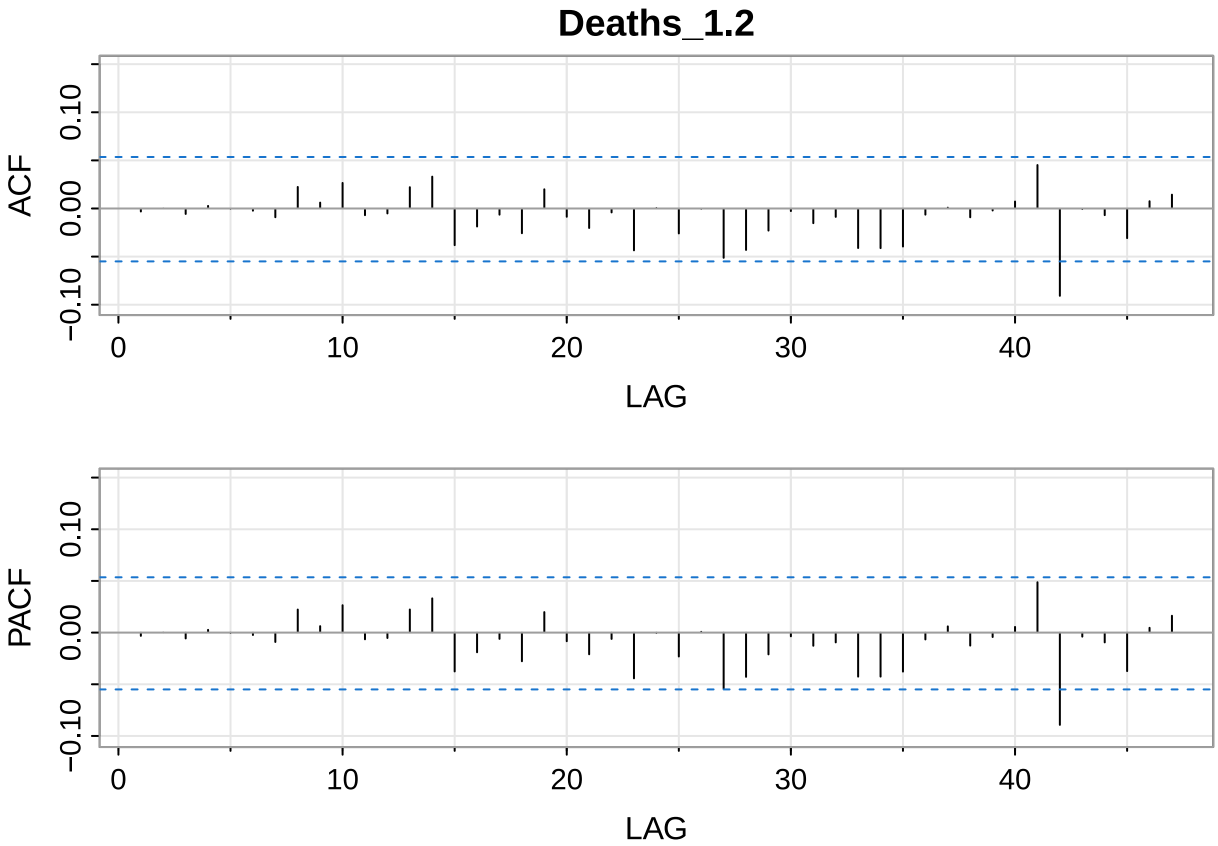
```
round(Box.test(Deaths_1.2, lag=sqrt(length(Deaths_1.2)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.95633
```

```
round(Box.test((Deaths_1.2)^2, lag=sqrt(length(Deaths_1.2)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```

```
Deaths_1.2 <- arima(Deaths_1, order=c(0,0,9), seasonal = list(order=c(0,1,0), period=2),
                    method="ML")$residuals
x = acf2(Deaths_1.2, main = "Deaths_1.2")
```



```
round(ad.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0
```

```
round(shapiro.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0
```

```
round(adf.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0.01
```

```
round(kpss.test(Deaths_1.2)$p.value, 5)
```

```
## [1] 0.1
```

```
round(Box.test(Deaths_1.2, lag=sqrt(length(Deaths_1.2)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.90362
```

```
round(Box.test(Deaths_1.2, lag=sqrt(length(Deaths_1.2)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

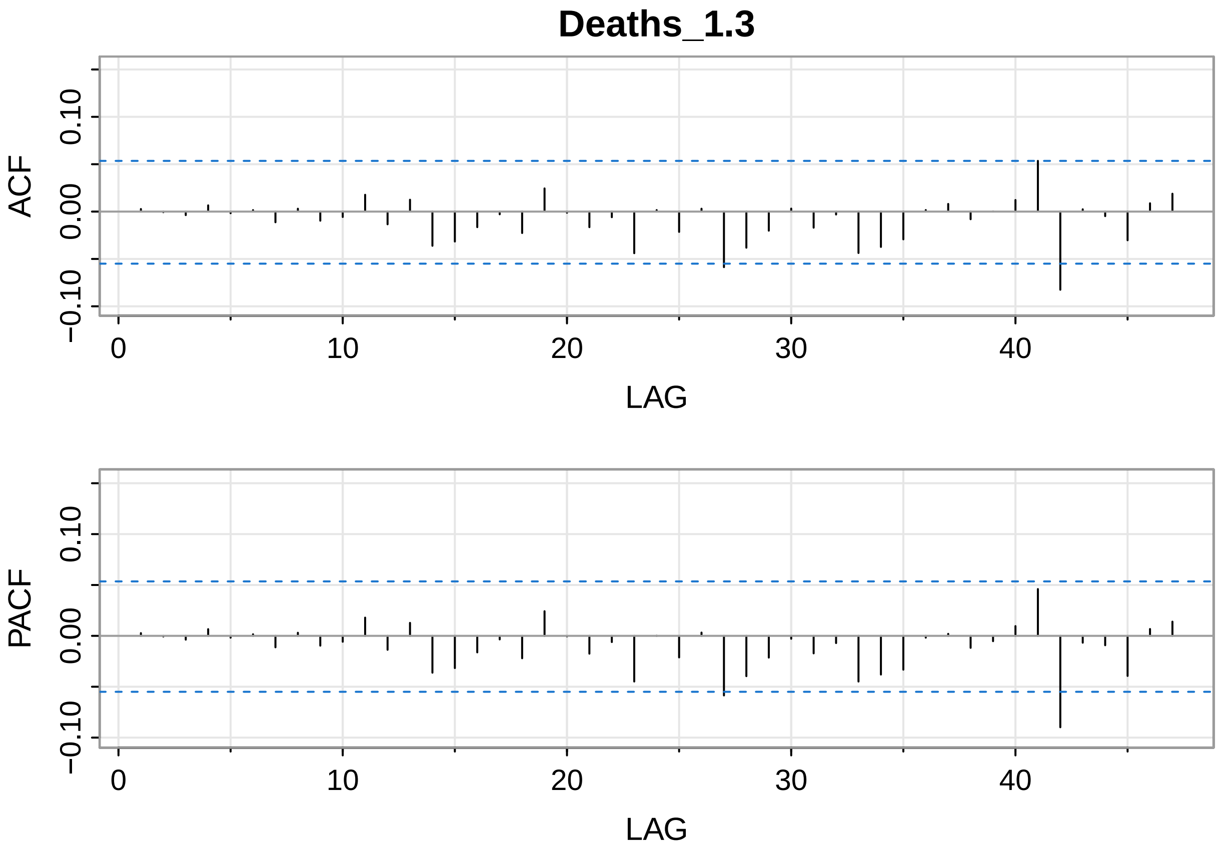
```
## [1] 0.89008
```

```
round(Box.test((Deaths_1.2)^2, lag=sqrt(length(Deaths_1.2)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```

Periodicity of 3

```
Deaths_1.3 <- arima(Deaths_1, order=c(7,0,3), seasonal = list(order=c(0,1,0), period=3),
                    method="ML")$residuals
x = acf2(Deaths_1.3, main = "Deaths_1.3")
```



```
round(ad.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0
```

```
round(shapiro.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0
```

```
round(adf.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0.01
```

```
round(kpss.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0.1
```

```
round(Box.test(Deaths_1.3, lag=sqrt(length(Deaths_1.3)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.95937
```

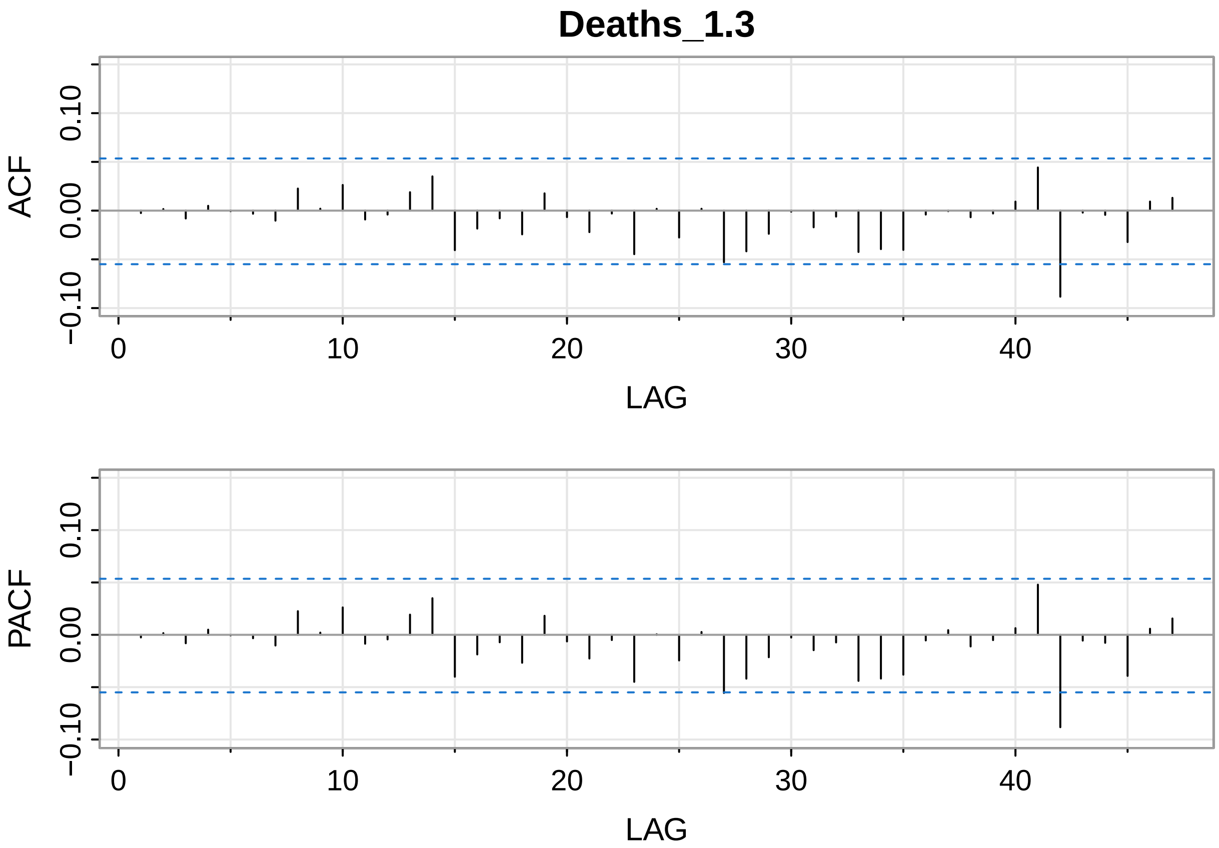
```
round(Box.test(Deaths_1.3, lag=sqrt(length(Deaths_1.3)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.95246
```

```
round(Box.test((Deaths_1.3)^2, lag=sqrt(length(Deaths_1.3)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```

```
Deaths_1.3 <- arima(Deaths_1, order=c(0,0,10), seasonal = list(order=c(0,1,0), period=3),
                    method="ML")$residuals
x = acf2(Deaths_1.3, main = "Deaths_1.3")
```



```
round(ad.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0
```

```
round(shapiro.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0
```

```
round(adf.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0.01
```

```
round(kpss.test(Deaths_1.3)$p.value, 5)
```

```
## [1] 0.1
```

```
round(Box.test(Deaths_1.3, lag=sqrt(length(Deaths_1.3)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.88468
```

```
round(Box.test(Deaths_1.3, lag=sqrt(length(Deaths_1.3)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

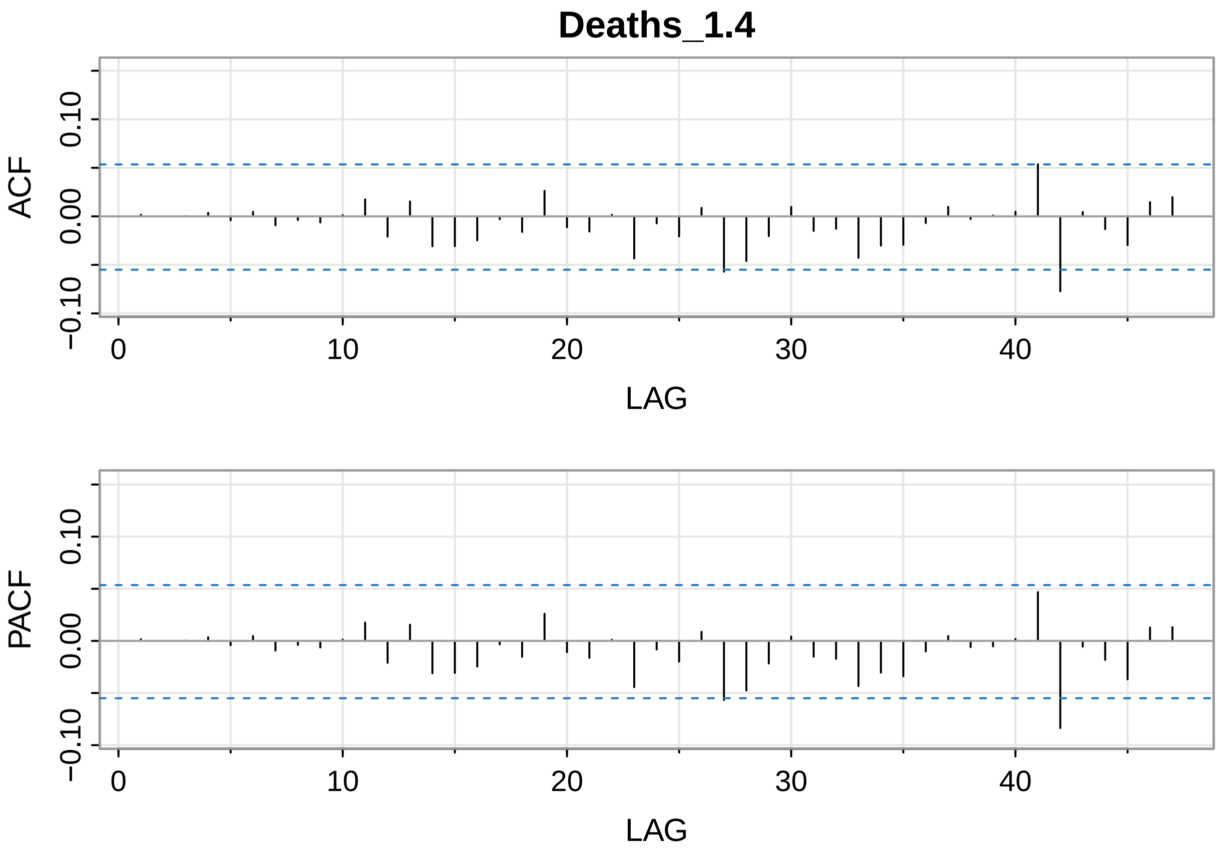
```
## [1] 0.86927
```

```
round(Box.test((Deaths_1.3)^2, lag=sqrt(length(Deaths_1.3)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```

Periodicity of 4

```
Deaths_1.4 <- arima(Deaths_1, order=c(7,0,4), seasonal = list(order=c(0,1,0), period=4),
                    method="ML")$residuals
x = acf2(Deaths_1.4, main = "Deaths_1.4")
```



```
round(ad.test(Deaths_1.4)$p.value, 5)
```

```
## [1] 0
```

```
round(shapiro.test(Deaths_1.4)$p.value, 5)
```

```
## [1] 0
```

```
round(adf.test(Deaths_1.4)$p.value, 5)
```

```
## [1] 0.01
```

```
round(kpss.test(Deaths_1.4)$p.value, 5)
```

```
## [1] 0.1
```

```
round(Box.test(Deaths_1.4, lag=sqrt(length(Deaths_1.4)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.95564
```

```
round(Box.test(Deaths_1.4, lag=sqrt(length(Deaths_1.4)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

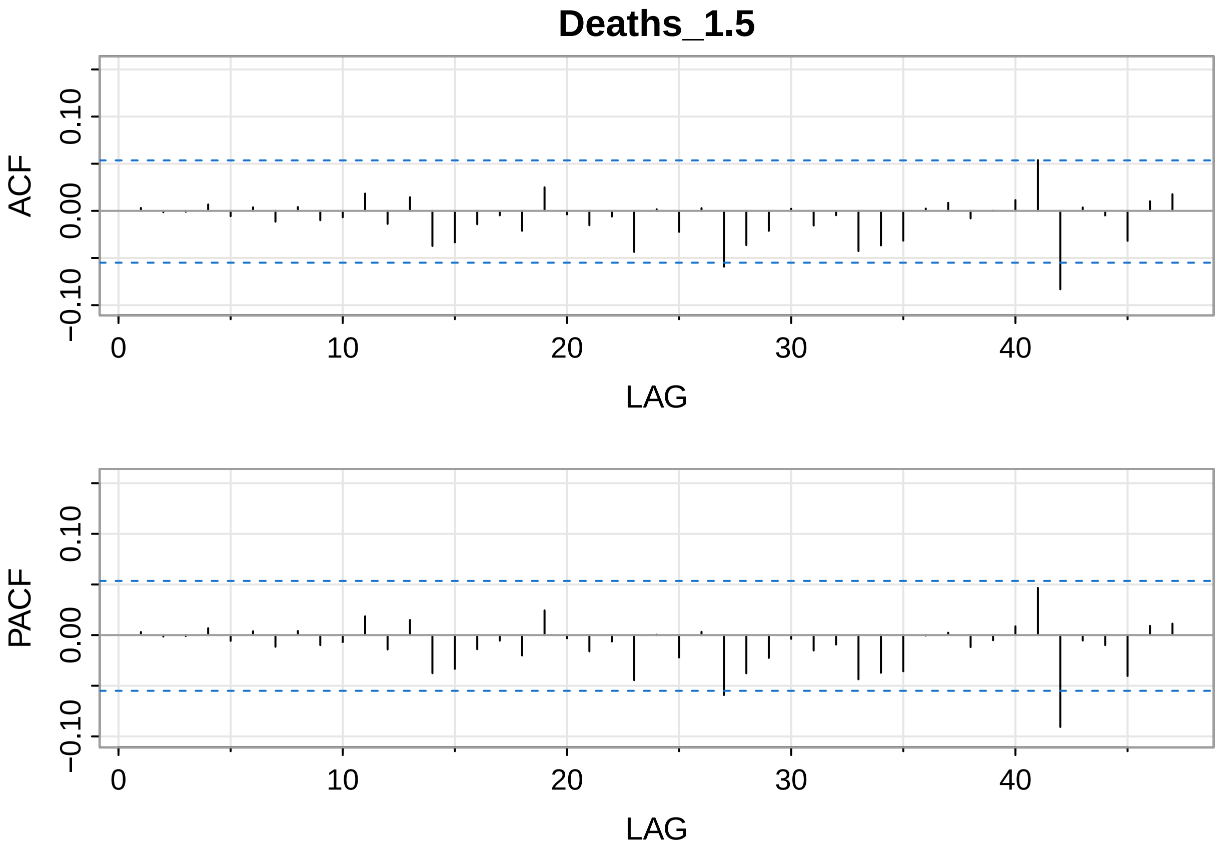
```
## [1] 0.94824
```

```
round(Box.test((Deaths_1.4)^2, lag=sqrt(length(Deaths_1.4)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```

Periodicity of 5

```
Deaths_1.5 <- arima(Deaths_1, order=c(7,0,5), seasonal = list(order=c(0,1,0), period=5),
                    method="ML")$residuals
x = acf2(Deaths_1.5, main = "Deaths_1.5")
```



```
round(ad.test(Deaths_1.5)$p.value, 5)
```

```
## [1] 0
```

```
round(shapiro.test(Deaths_1.5)$p.value, 5)
```

```
## [1] 0
```

```
round(adf.test(Deaths_1.5)$p.value, 5)
```

```
## [1] 0.01
```

```
round(kpss.test(Deaths_1.5)$p.value, 5)
```

```
## [1] 0.1
```

```
round(Box.test(Deaths_1.5, lag=sqrt(length(Deaths_1.5)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.95627
```

```
round(Box.test(Deaths_1.5, lag=sqrt(length(Deaths_1.5)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

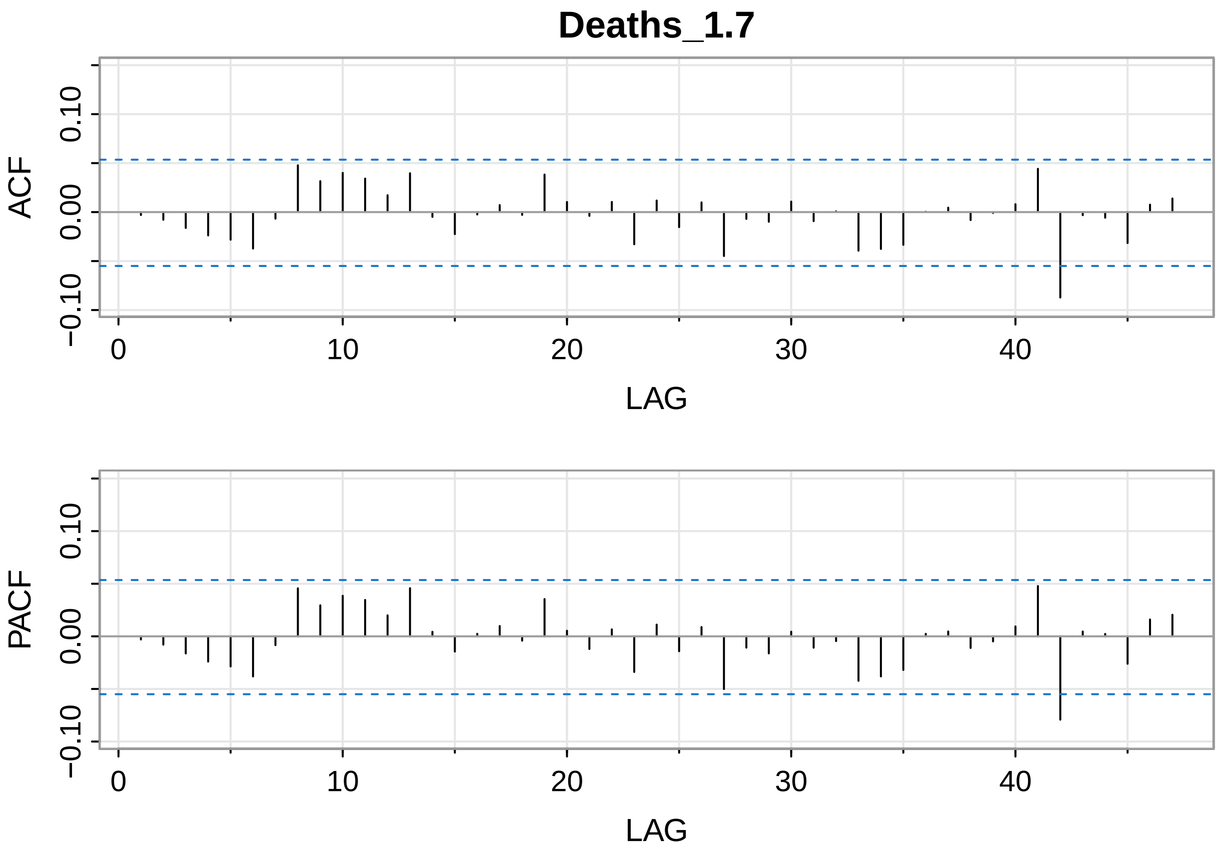
```
## [1] 0.94899
```

```
round(Box.test((Deaths_1.5)^2, lag=sqrt(length(Deaths_1.5)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```

Periodicity of 7

```
Deaths_1.7 <- arima(Deaths_1, order=c(6,0,0), seasonal = list(order=c(5,1,2), period=7),
                    method="ML")$residuals
x = acf2(Deaths_1.7, main = "Deaths_1.7")
```



```
round(ad.test(Deaths_1.7)$p.value, 5)
```

```
## [1] 0
```

```
round(shapiro.test(Deaths_1.7)$p.value, 5)
```

```
## [1] 0
```

```
round(adf.test(Deaths_1.7)$p.value, 5)
```

```
## [1] 0.01
```

```
round(kpss.test(Deaths_1.7)$p.value, 5)
```

```
## [1] 0.1
```

```
round(Box.test(Deaths_1.7, lag=sqrt(length(Deaths_1.7)), type=c("Box-Pierce"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.81835
```

```
round(Box.test(Deaths_1.7, lag=sqrt(length(Deaths_1.7)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0.8026
```

```
round(Box.test((Deaths_1.7)^2, lag=sqrt(length(Deaths_1.7)), type=c("Ljung-Box"),
               fitdf = 0)$p.value, 5)
```

```
## [1] 0
```