STAT 221 Final Project - Mammoth Snow Depth

John Rapp Farnes | 405461225

3/17 2020

Contents

1 Introduction **1**

1.1 Background . . . 1

1.2 Data set . . . 1

1.3 Cleaning data . . . 2

2 Analyis **2**

2.1 Series properties . . . 2

2.2 Detrending . . . 2

2.3 Detrended series properties . . . 3

2.4 Model fitting . . . 3

2.5 Model interpretation . . . 3

2.6 Model evaluation . . . 3

2.7 Frequency domain . . . 3

2.8 Alternative model . . . 3

3 Results **3**

3.1 Conclusion . . . 3

3.2 Dicsussion . . . 3

3.3 Next steps . . . 3

4 Appendix - Figures and graphs **3**

5 References **3**

6 ACF **34**

7 Fit ARIMA **35**

% TODO - Prediction på known värden - Tillbaka trend på pred

- Kommentera stora residualer? datum?
- Log: större skillnad större värden, plotta
- Periodogram residualer
- Tapering

1 Introduction

1.1 Background

Mammoth is a mountain in Northern California know for its great alpine ski and snowboarding conditions. For this purpose, the mountain features a ski resort with the same name. This resort has has more than 3,500 acres of ski-able terrain and is serviced by 28 lifts and recieves over 1 million anuual visitors. For Southern Californian residents, the mountain is of interest as it is one of the closest "good" ski resorts, about a 4-6 hour drive away from Los Angeles. **??**

As people familiar with alpine sports know, one of the most important conditions for the sport is the snow depth at the mountain, as this affects which runs are open and how "good" the skiing is. As decisions about travelling to a ski resort generally are done in advance and require some planning (e.g. booking a cabin), being able to predicts future conditions would be useful to the alpine skiier. In addition, the entire ski economy of a mountain such as Mammoth, including the resort, workers as well as restaurants and stores in the sorrounding city, face uncertainty over how many visitors the mountain will get a given week or year as this drives revenue. As visitorships likely is correlated with snow depth, forecasting it would be an important tool also for those stake holders. This paper will attempt to do just that: **model the snow depth at Mammoth mountain in order to make predictions on future skiing conditions and mountain visitorship**.

1.2 Data set

Data on historic snow depth at Mammoth were obtained from the reporting of Mammoth Mountain Ski Area, through a third party website **??**. The website does not give the data easily in a downloadable form, hence the data was obtained through injecting JavaScript into a browser client that took the data from the browser JavaScript enviroment and printed it in a PDF format. This raw data is shown in figure 1, featuring 1791 recordings from 2011-12-01 through 2020-03-02 of daily snow depth measured in incehs. Opon looking at the graph, two issues with the raw data are found: 1) The dates in the off-season (outside of the winter months) are not included, rather the years are concatinated together in a single time-frame 2) some values within the recorded period are missing and reported as 0. Hence, the data needed to be cleaned.

1.3 Cleaning data

The first step in cleaning the data was to include the missing dates in order to capture the full time-frame of the data. The next step was handling the missing values, both in the off-season as well as the missing recorded values. In order to handle the missing recorded values, as well as to make lower variance predictions on snow depth further in the future than a couple of days, the data was aggregated and averaged (disregarding the missing values) per week, creating a weekly time series. This week was defined as starting a Saturday, as this a day of interest for many weekend skieers. The off-season missing values were replaced by 0s, as this is an accurate description of the snow depth during those months – there rarely is not snow on Mammoth during the summer. The resulting data after cleaning is shown in figure **??**, featuring 431 weeks. The availability of data per month is displayed in **??**, showing that data exists for the most part Dec-May, with less than 50% Jun-Oct, reaching ~80% in November.

2 Analyis

Looking at the grapth, it is clear that the ski season of 2020 has a far greater snow depth than prior years, an unfortunate fact for skiiers this season. To study other properties than this obvious observation, time-series methods were applied.

2.1 Series properties

First, the properties of the time-series were studies. Figure ref{fig:acf_series} shows the ACF and PACF of the series, where the ACF shows periodic behaviour with length 1 year that tails off, and the PACF has 2 (barely) significant values and then cuts off, implying that a seasonal AR model may be a good description of the series. The 1 year period is easily seen also in the periodogram (figure **??**), together with a 4 year period, both significant peaks. The 4 year period may be an artifact of the data being recorded for 4 years and these years having a pattern of yearly depth by chance.

The data does not appear to be stationary as it has an obvious yearly trend, and ARIMA models can not be applied. Hence the data must first be detrended.

2.2 Detrending

The most obvious trend in the data, that must first be removed, is the seasonal trend. This trend can be seen in figure **??** which shows the average snow depth during the differing months of the year. This graph has a clear sinusoidal shape, that peaks in X and reaches its lowest value in Y.

2.3 Detrended series properties

2.4 Model fitting

2.5 Model interpretation

2.6 Model evaluation

2.6.1 Residual analysis

2.7 Frequency domain

2.8 Alternative model

3 Results

3.1 Conclusion

3.2 Dicsussion

3.3 Next steps

4 Appendix - Figures and graphs

5 References

1. https://www.mammothmountain.com/

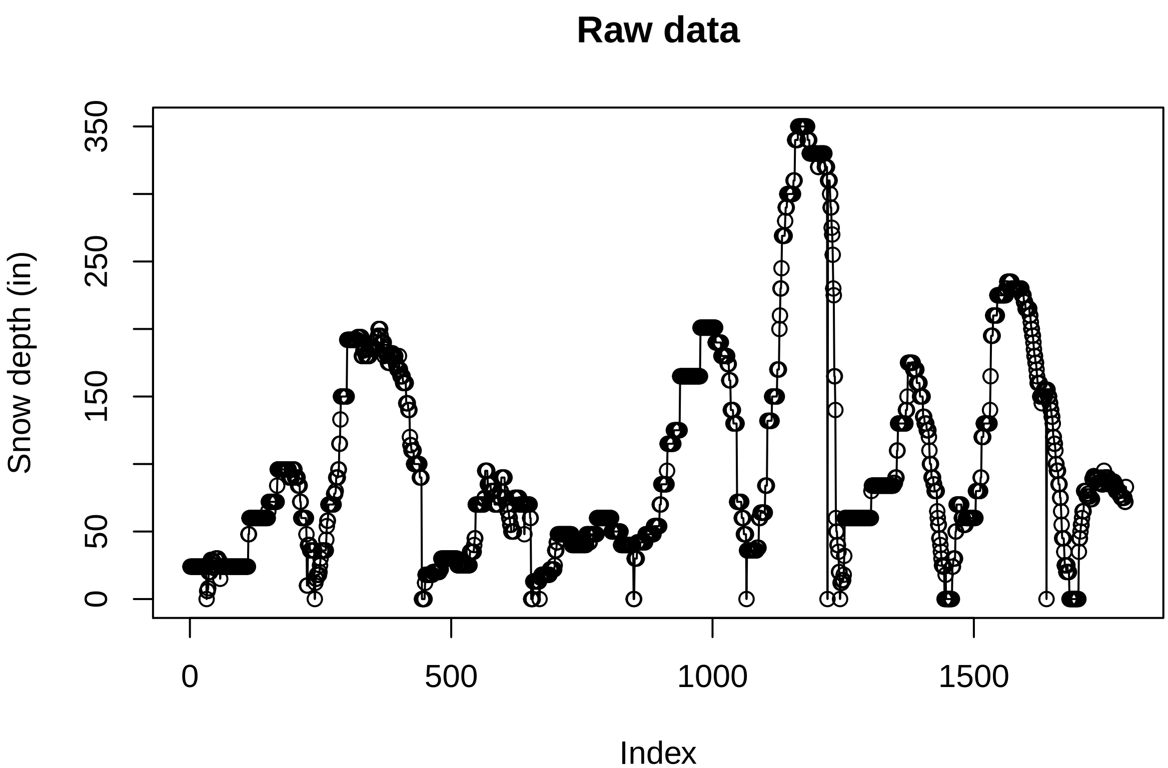
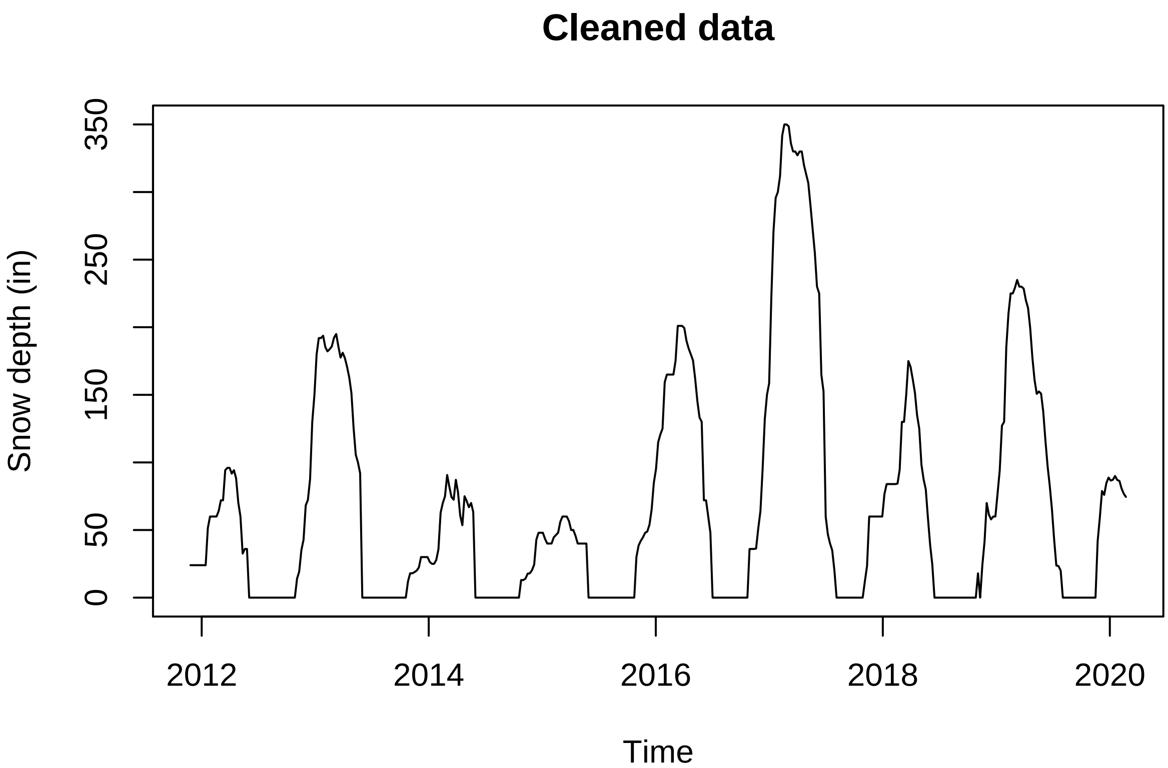
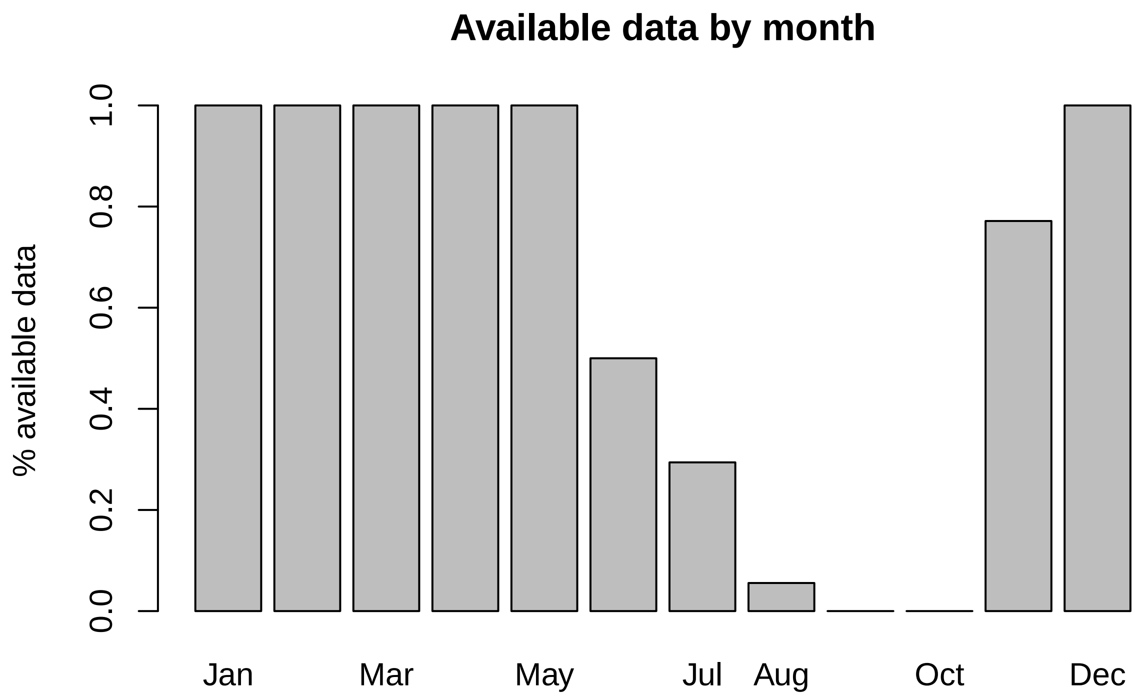


Figure 1: Raw Mammoth snow depth data


Cleaned data
Snow depth (in)
0
50
150
250
350
2012
2014
2016
2018
2020
Time


Available data by month
% available data
0.0
0.2
0.4
0.6
0.8
1.0
Jan
Mar
May
Jul
Aug
Oct
Dec

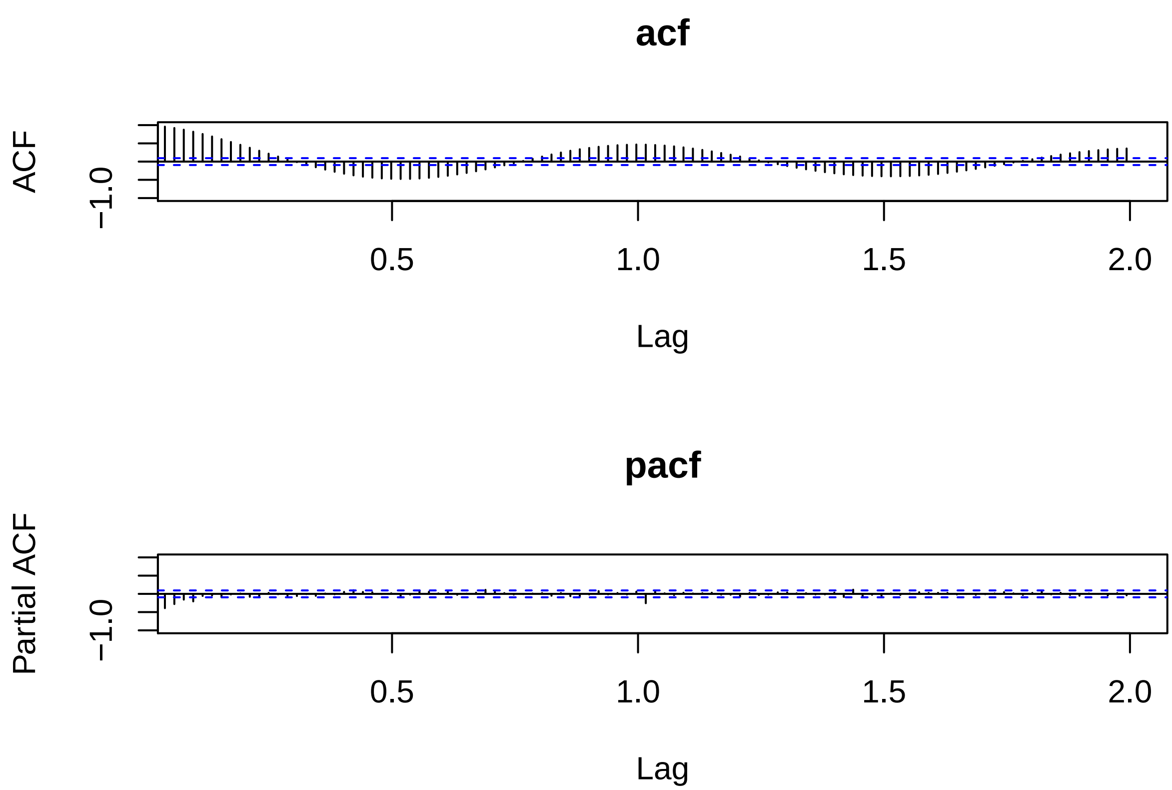
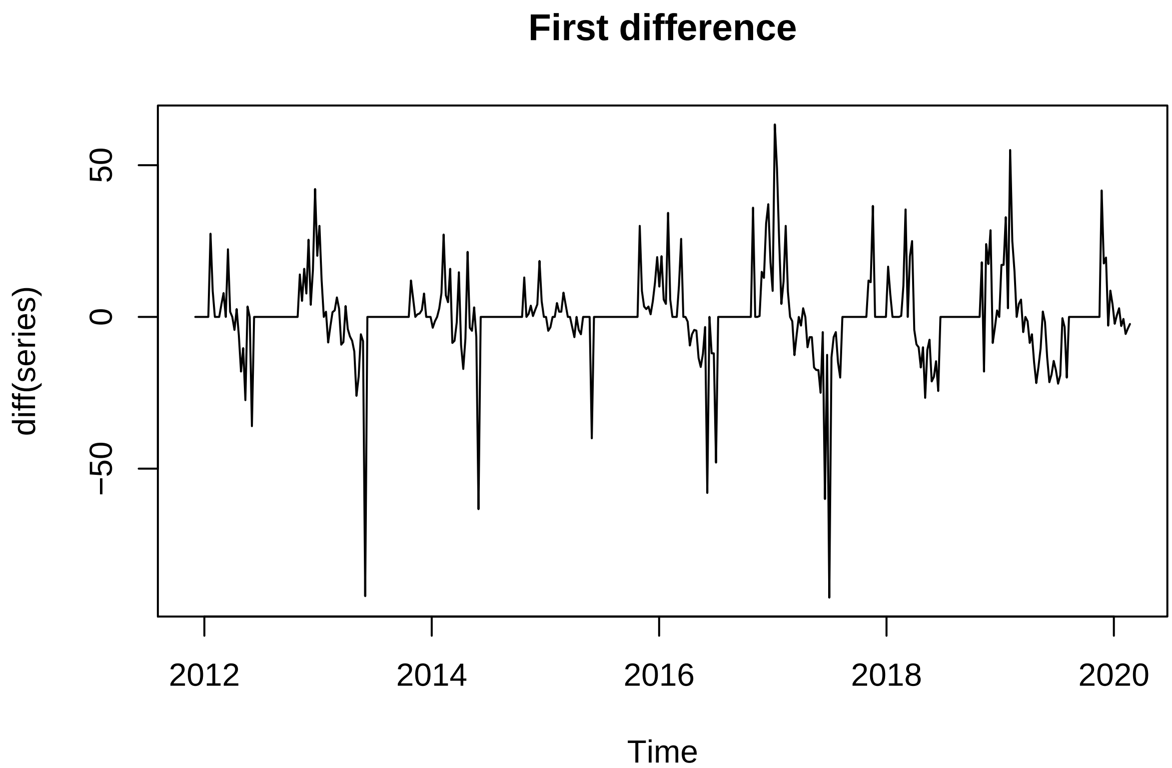
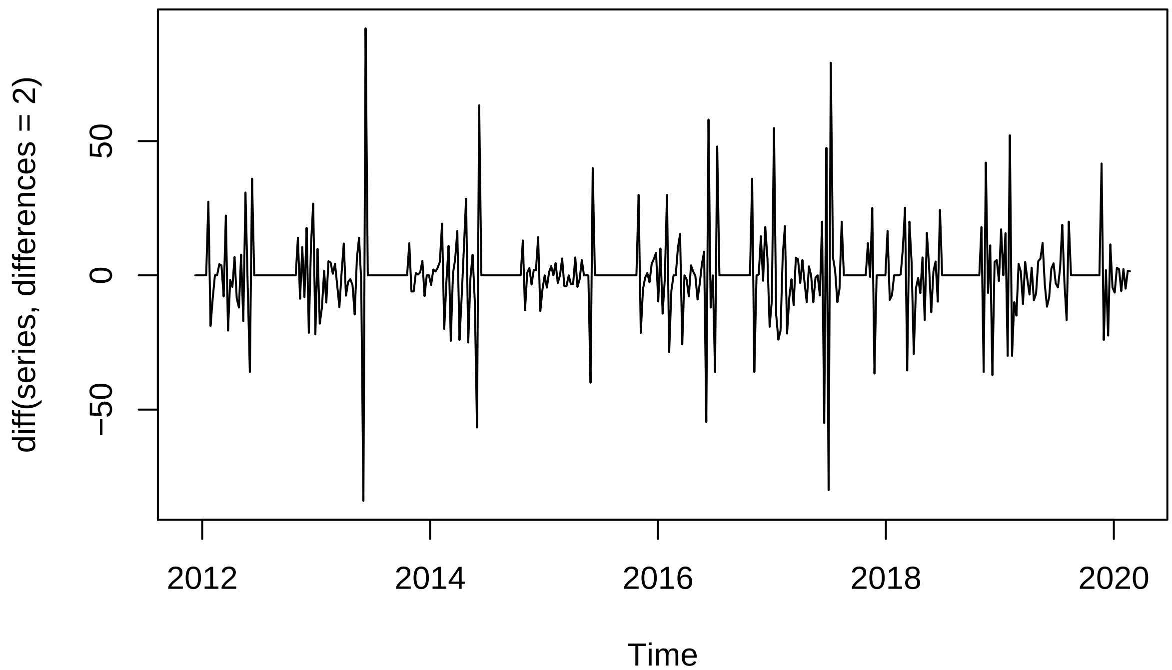
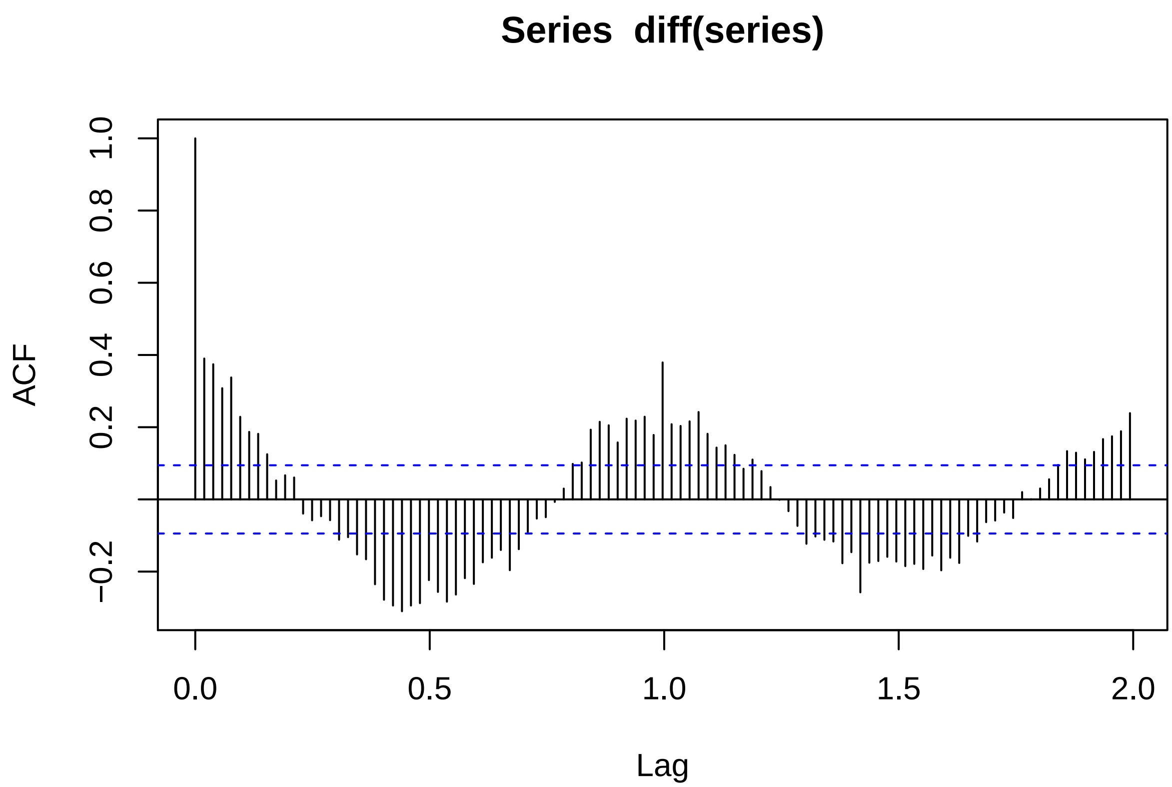


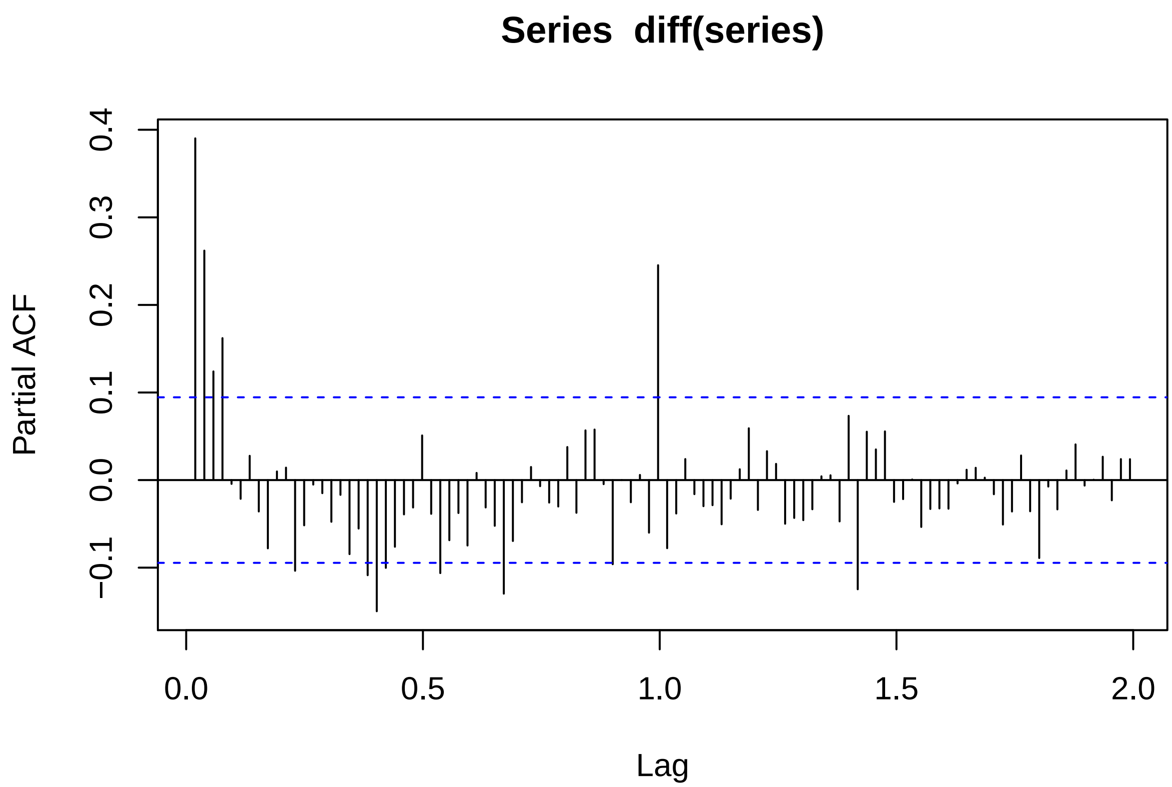
Figure 2: ACF and PACF for the cleaned time-series


First difference
diff(series)
50
0
−50
2012
2014
2016
2018
2020
Time


diff(series, differences = 2)
50
0
−50
2012
2014
2016
2018
2020
Time



Series diff(series)
ACF
−0.2
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Lag




```
## Warning in adf.test(series): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  series
## Dickey-Fuller = -5.5486, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary

## [1] 4.139630 1.034908
```

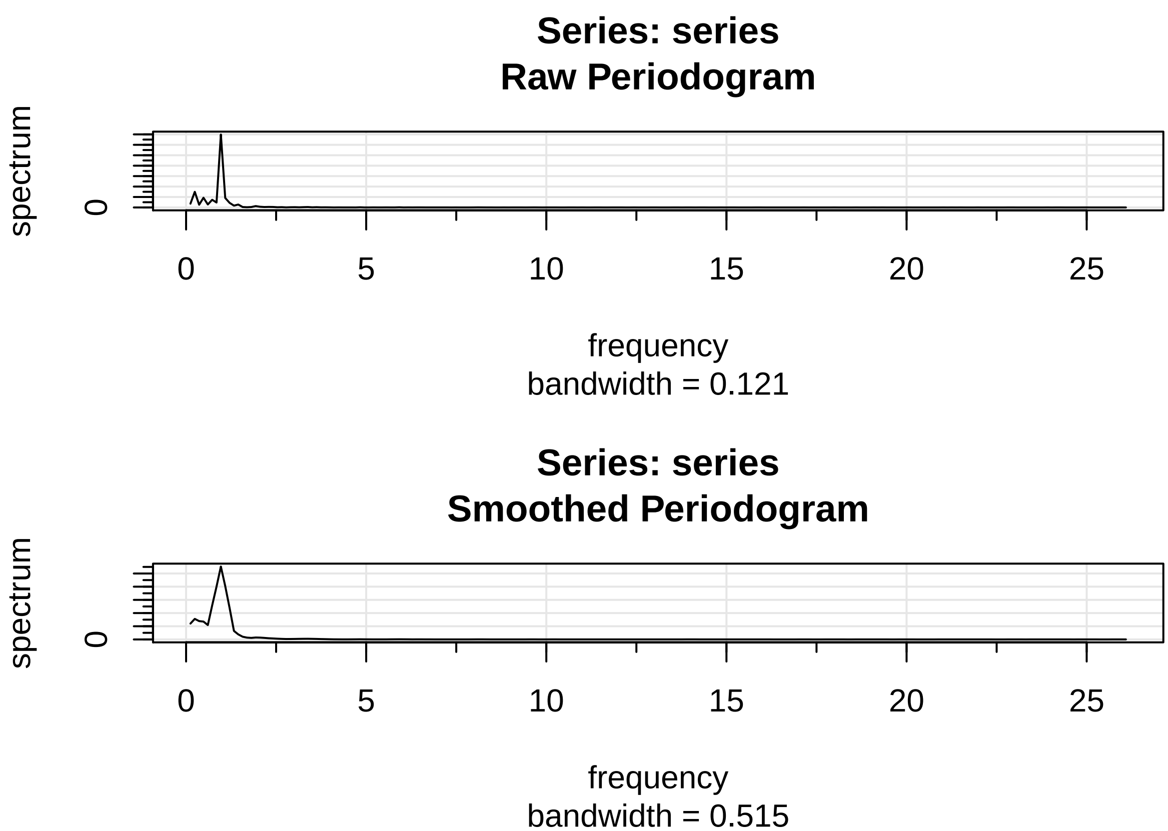

Series: series
Raw Periodogram
spectrum
0
0
5
10
15
20
25
frequency
bandwidth = 0.121
Series: series
Smoothed Periodogram
spectrum
0
0
5
10
15
20
25
frequency
bandwidth = 0.515

Figure 3:

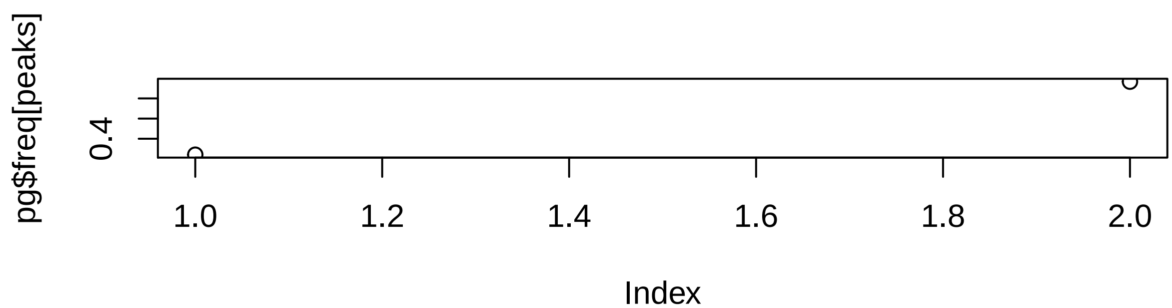
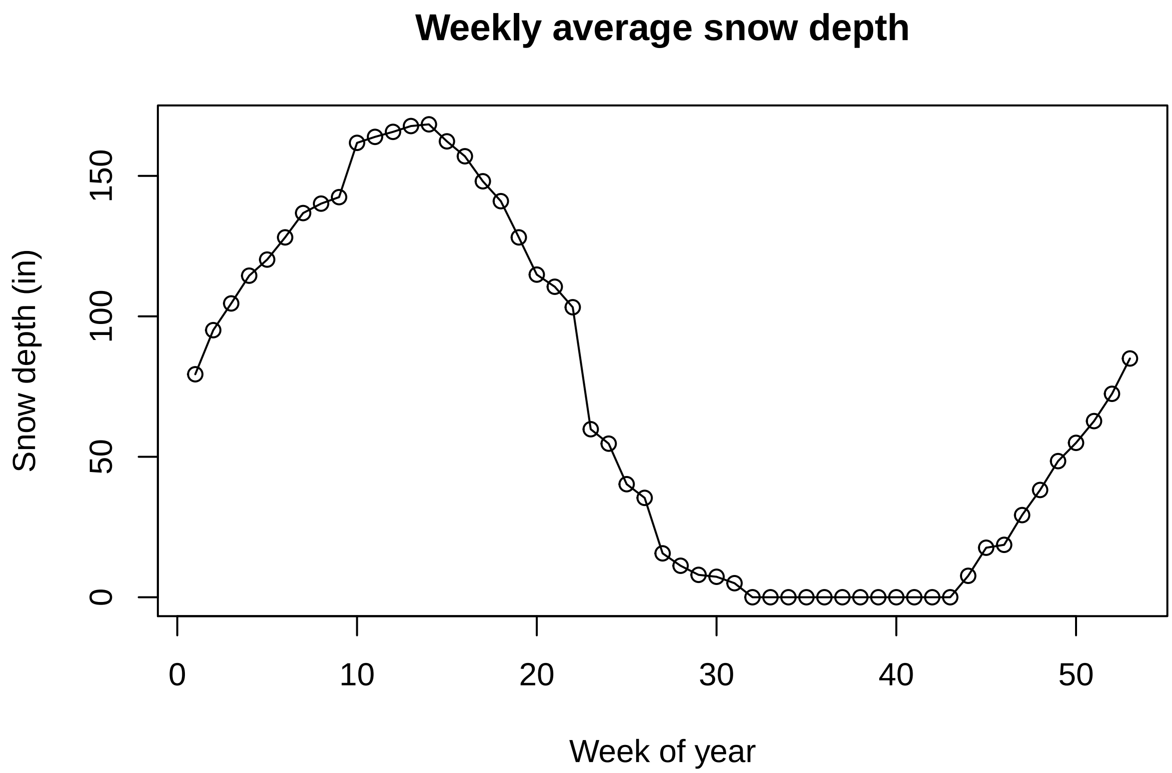
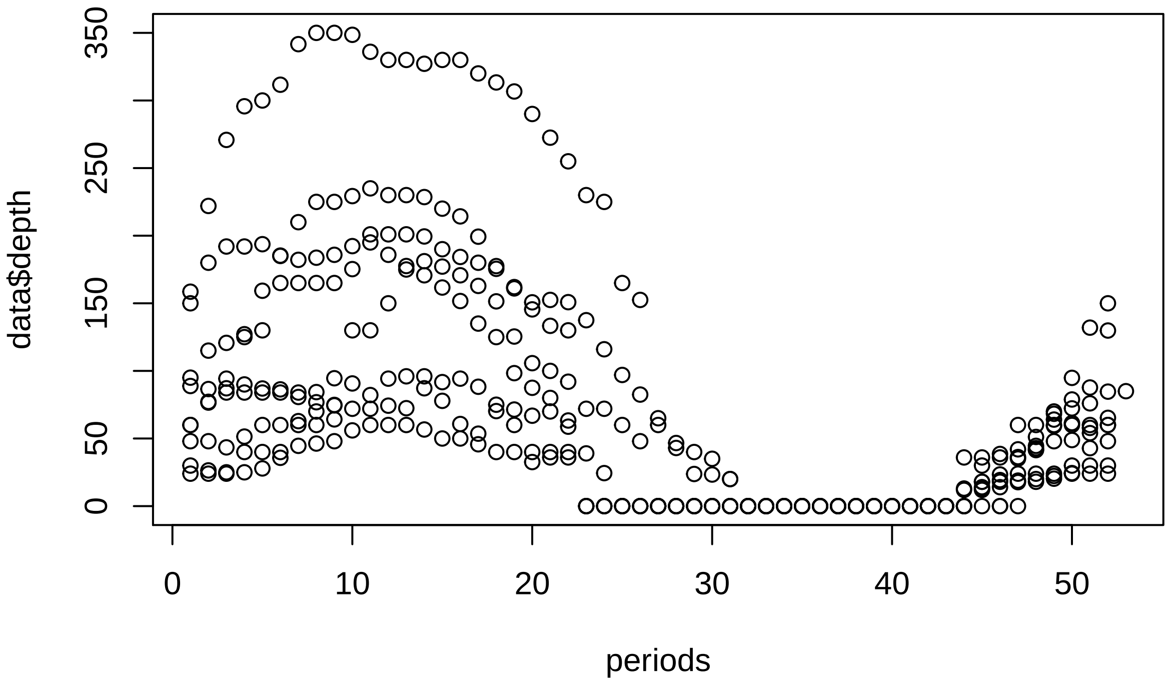
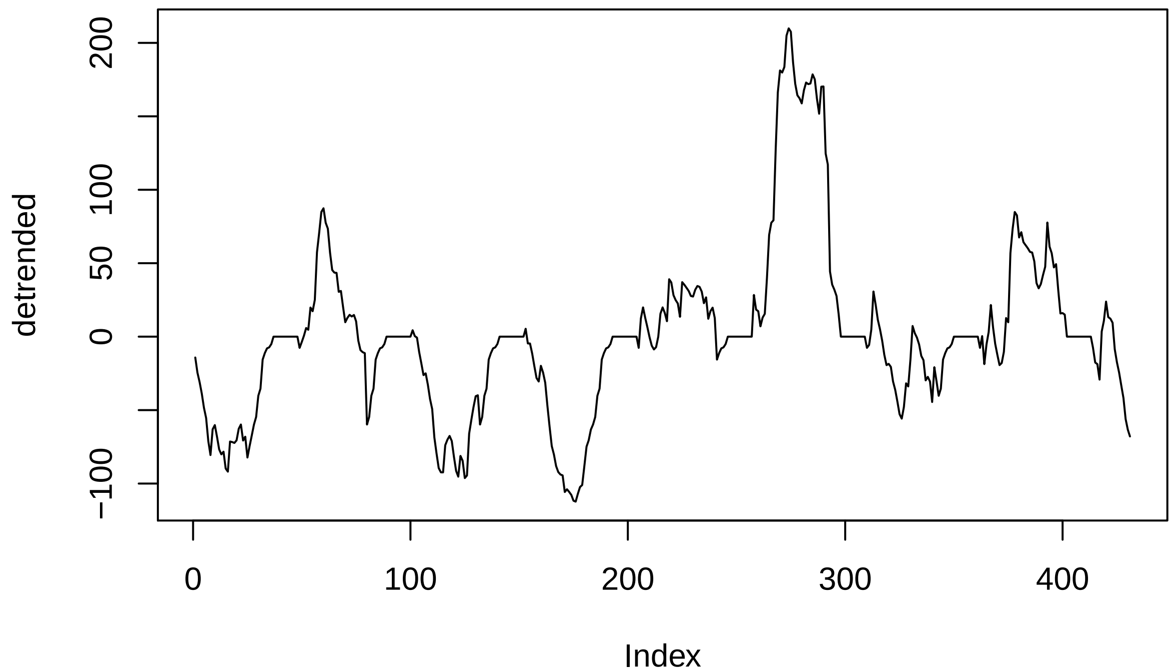
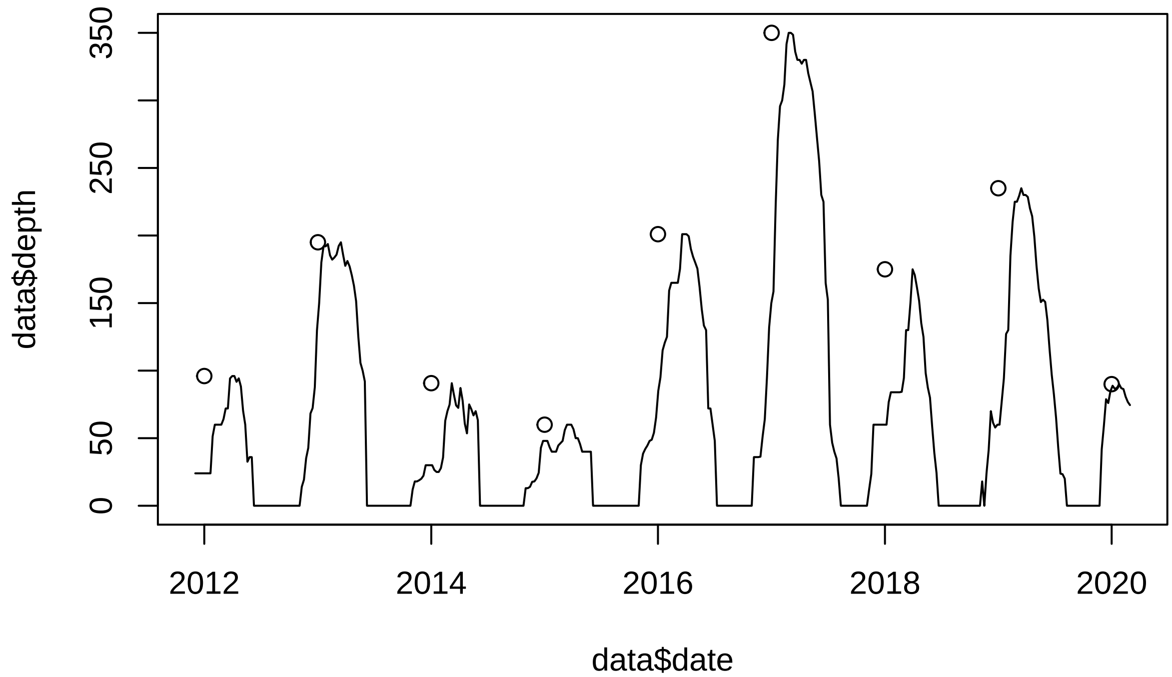


Figure 4:

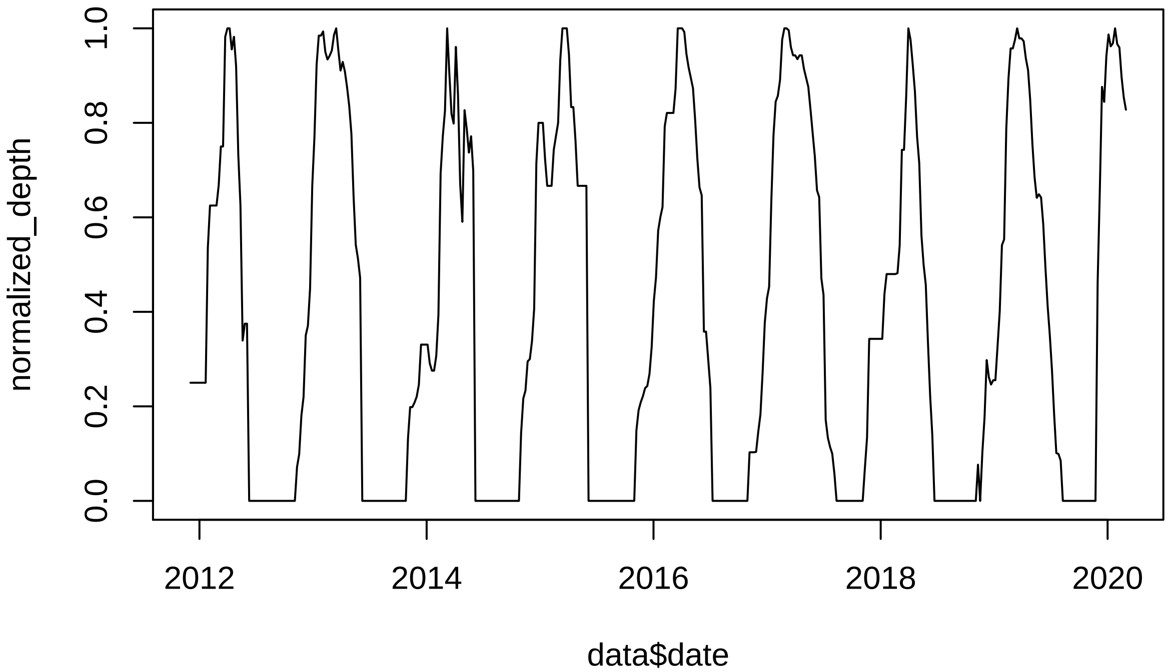

Weekly average snow depth
Snow depth (in)
0
50
100
150
0
10
20
30
40
50
Week of year


data$depth
periods
0
50
150
250
350
0
10
20
30
40
50

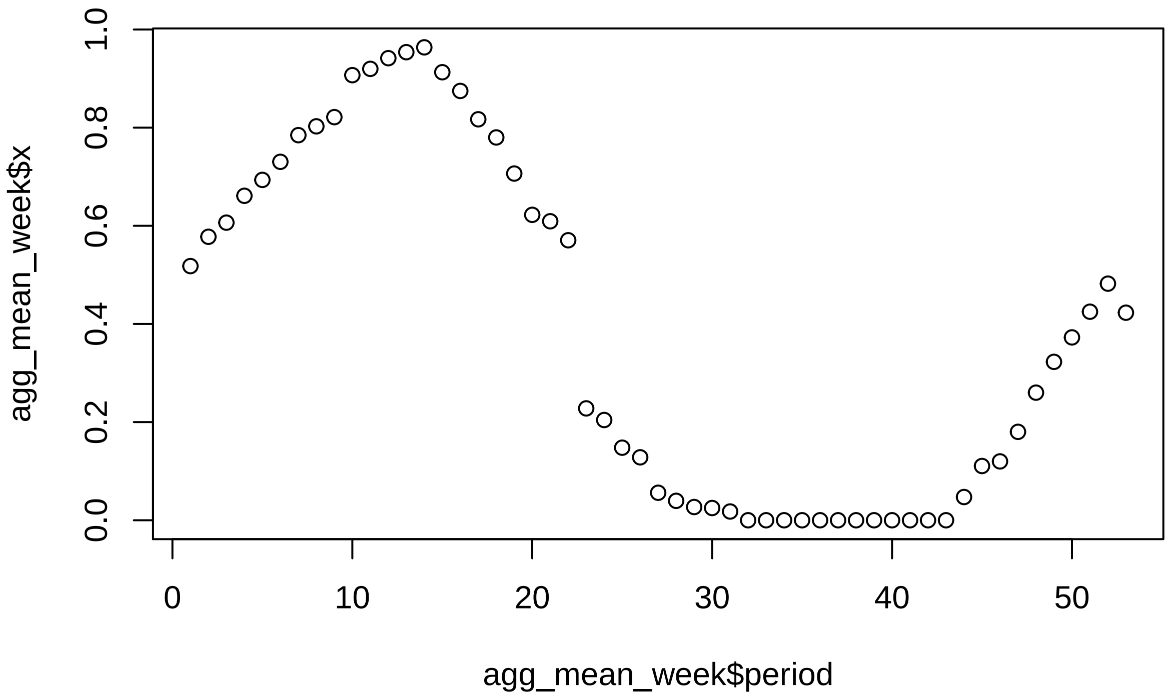

detrended
200
100
50
0
−100
0
100
200
300
400
Index



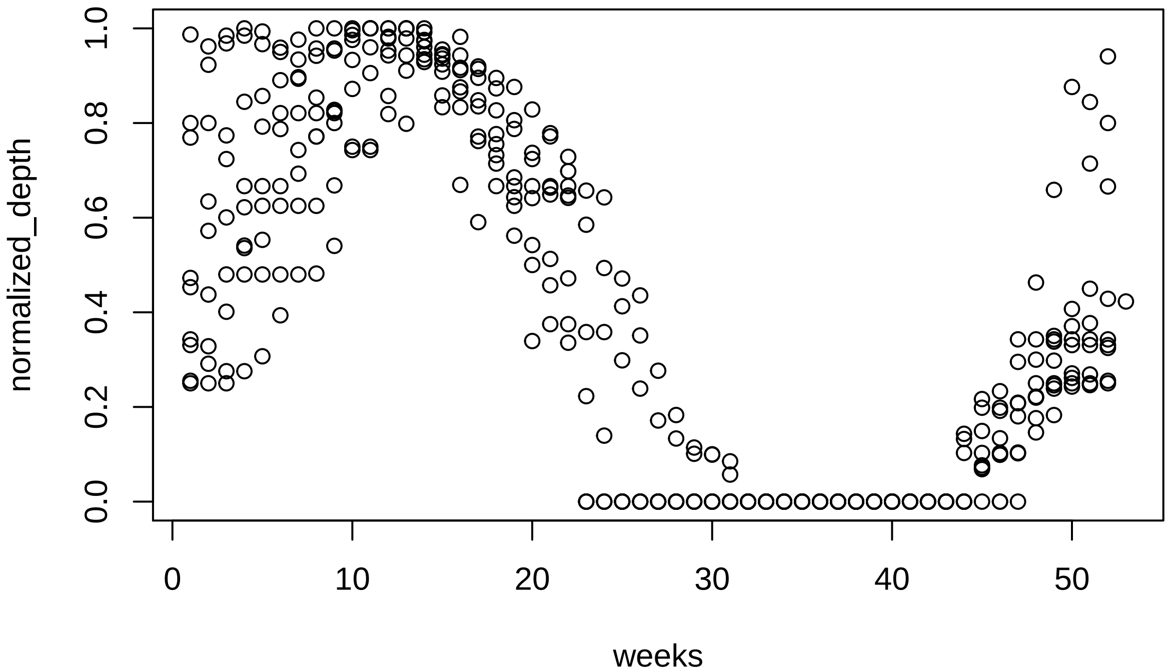
data$depth
0
50
150
250
350
2012
2014
2016
2018
2020
data$date



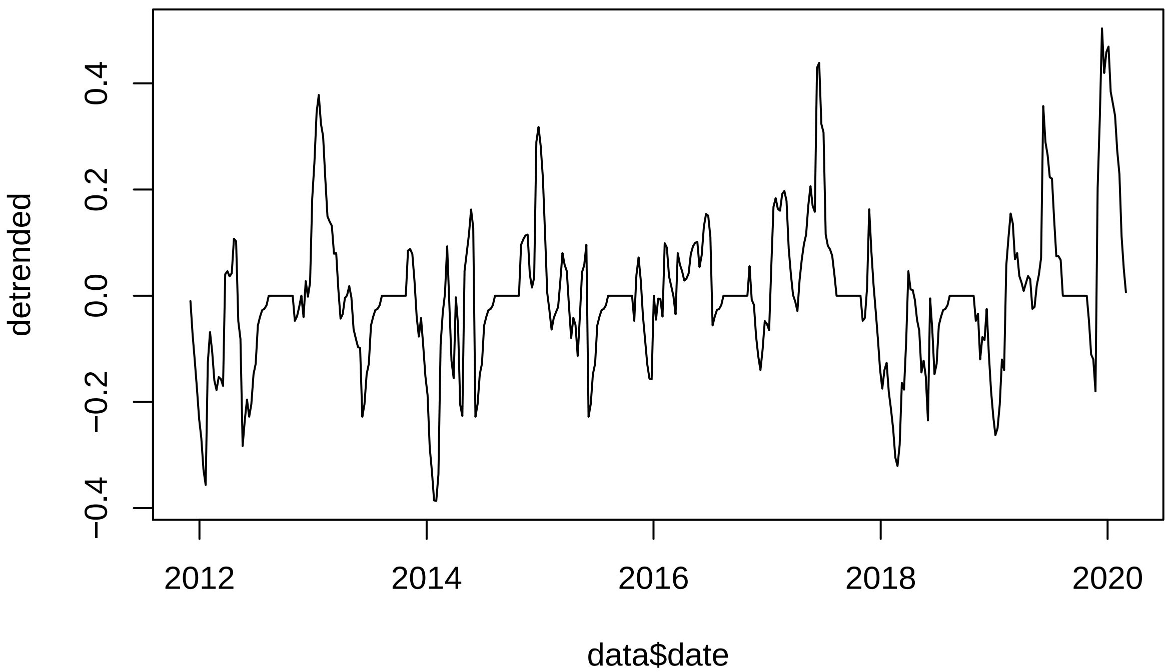
normalized_depth
0.0
0.2
0.4
0.6
0.8
1.0
2012
2014
2016
2018
2020
data$date



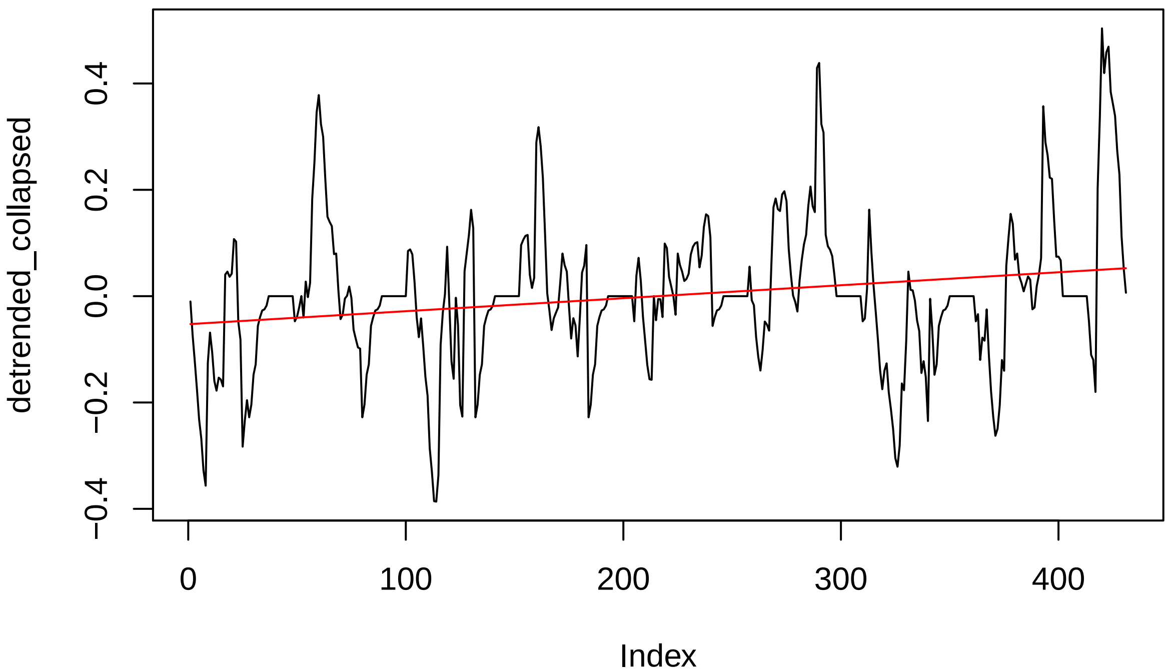
agg_mean_week$x
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50
agg_mean_week$period



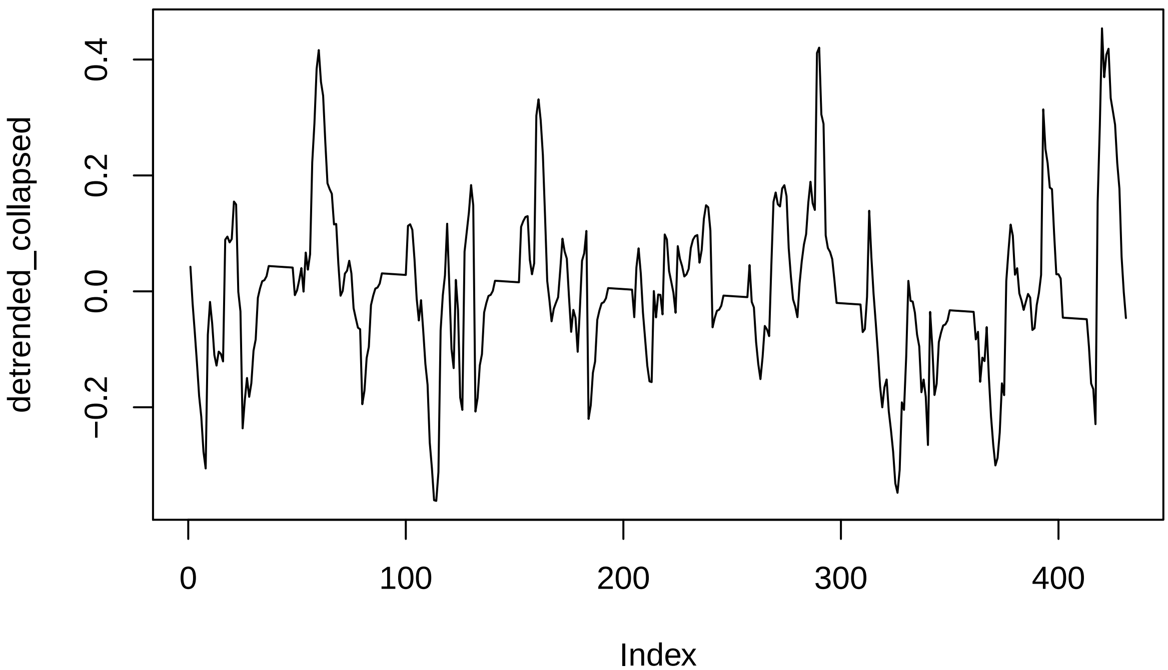
normalized_depth
0.0
0.2
0.4
0.6
0.8
1.0
0
10
20
30
40
50
weeks

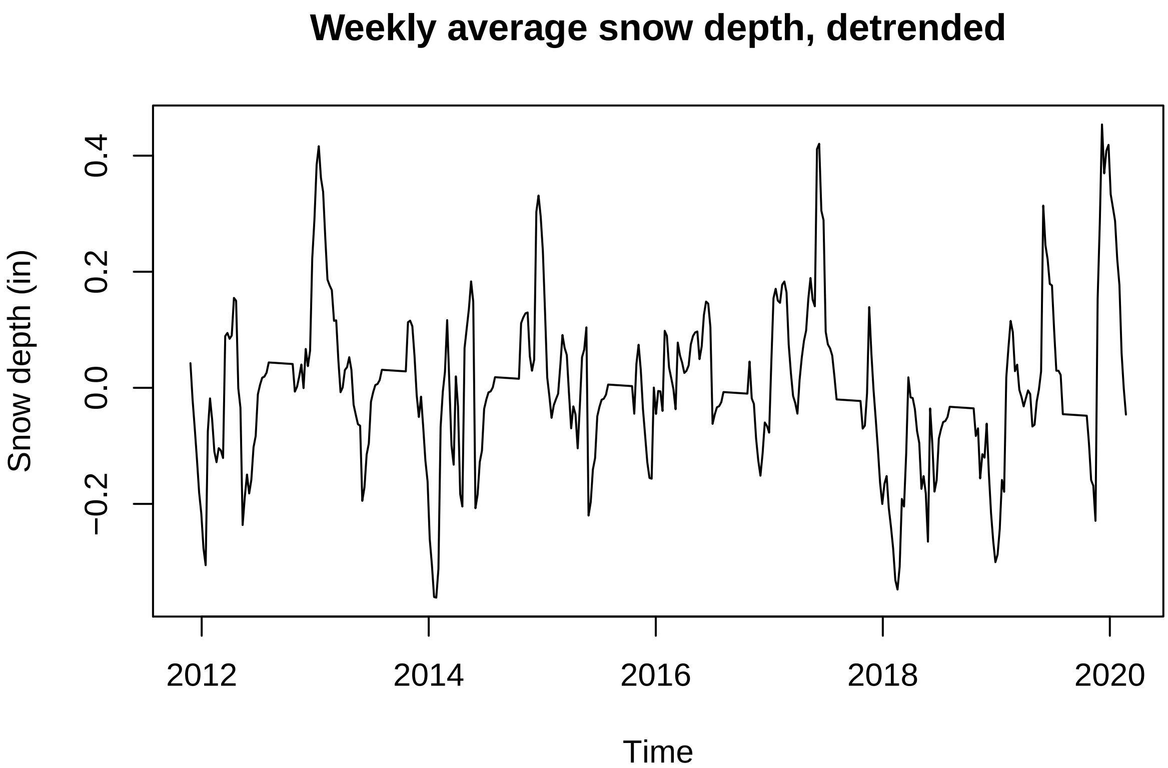



```
##
## Call:
## lm(formula = detrended_collapsed ~ time, data = regr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36153 -0.05802 -0.00686  0.04897  0.45383
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.273e-02  1.281e-02  -4.117 4.60e-05 ***
## time         2.441e-04  5.138e-05   4.751 2.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1327 on 429 degrees of freedom
## Multiple R-squared:  0.04999,    Adjusted R-squared:  0.04778
## F-statistic: 22.57 on 1 and 429 DF,  p-value: 2.762e-06
```

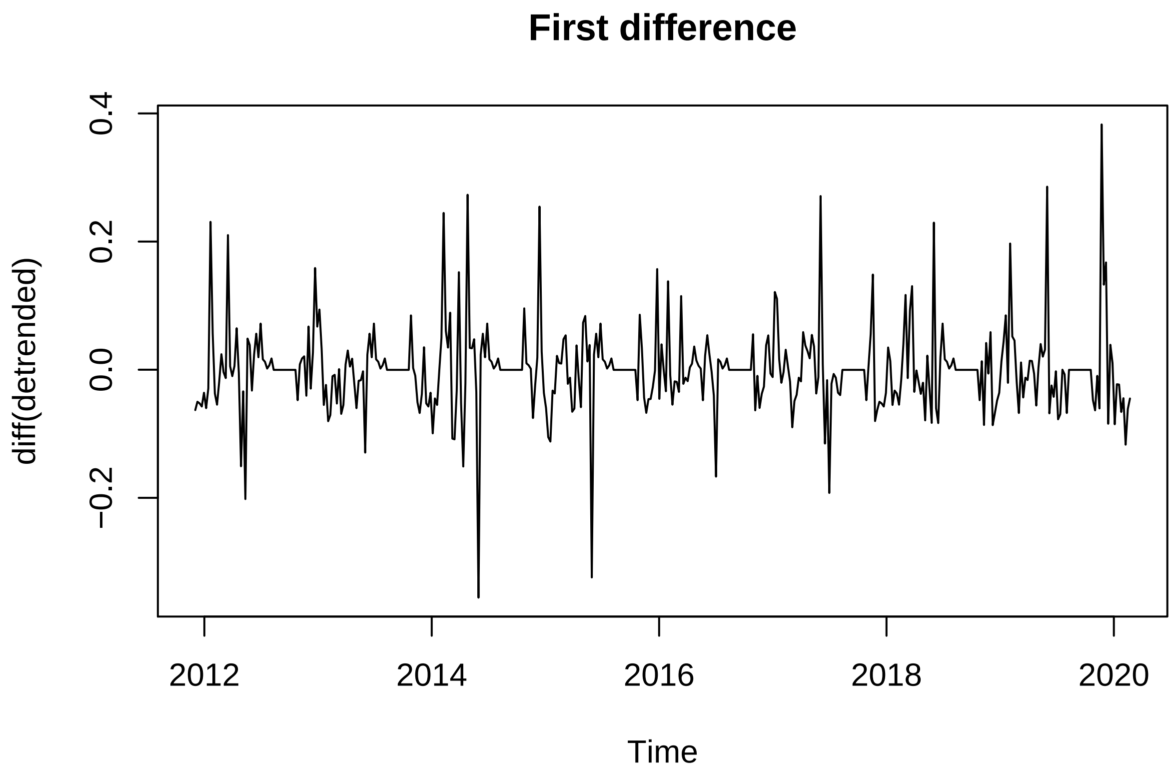


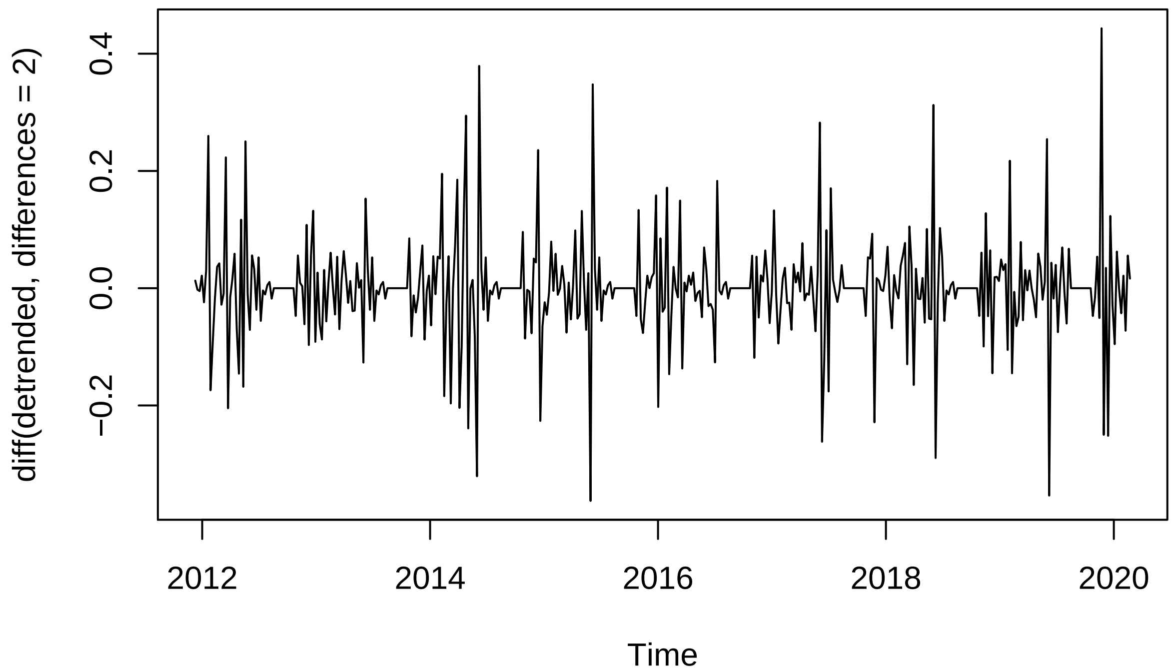
detrended_collapsed
−0.4
−0.2
0.0
0.2
0.4
0
100
200
300
400
Index

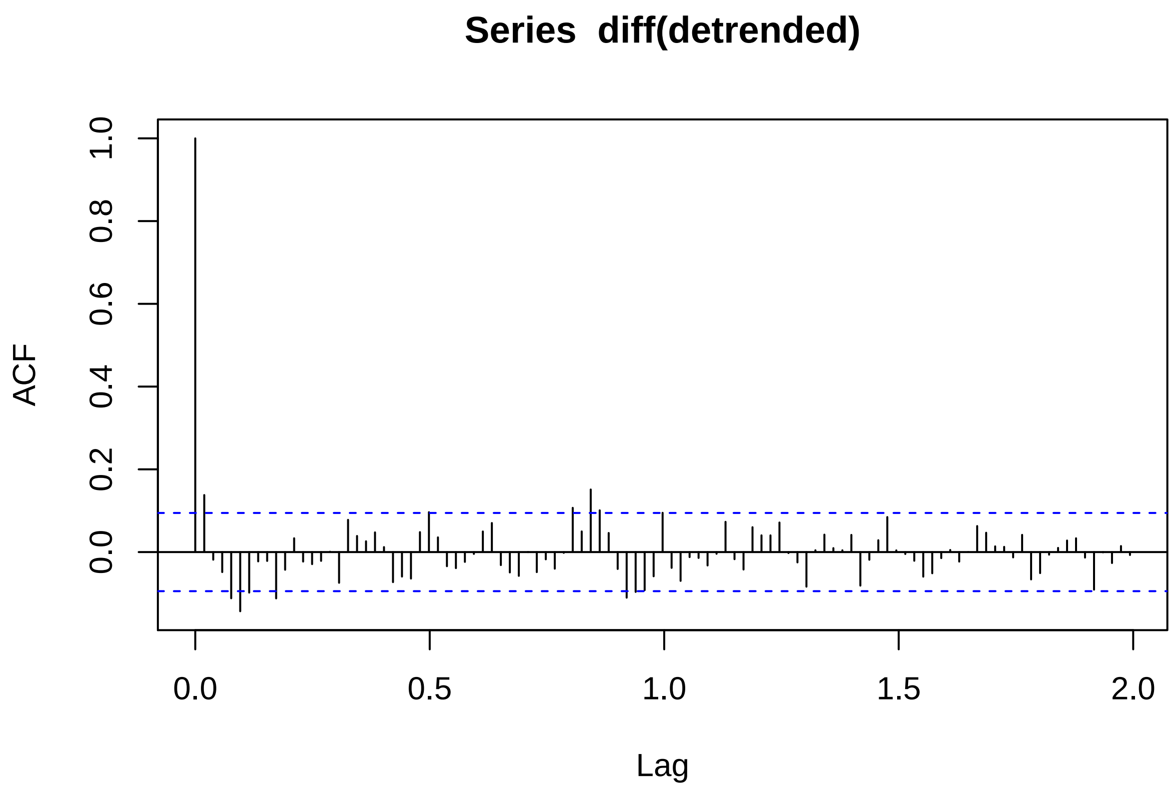

detrended_collapsed
-0.2
0.0
0.2
0.4
0
100
200
300
400
Index

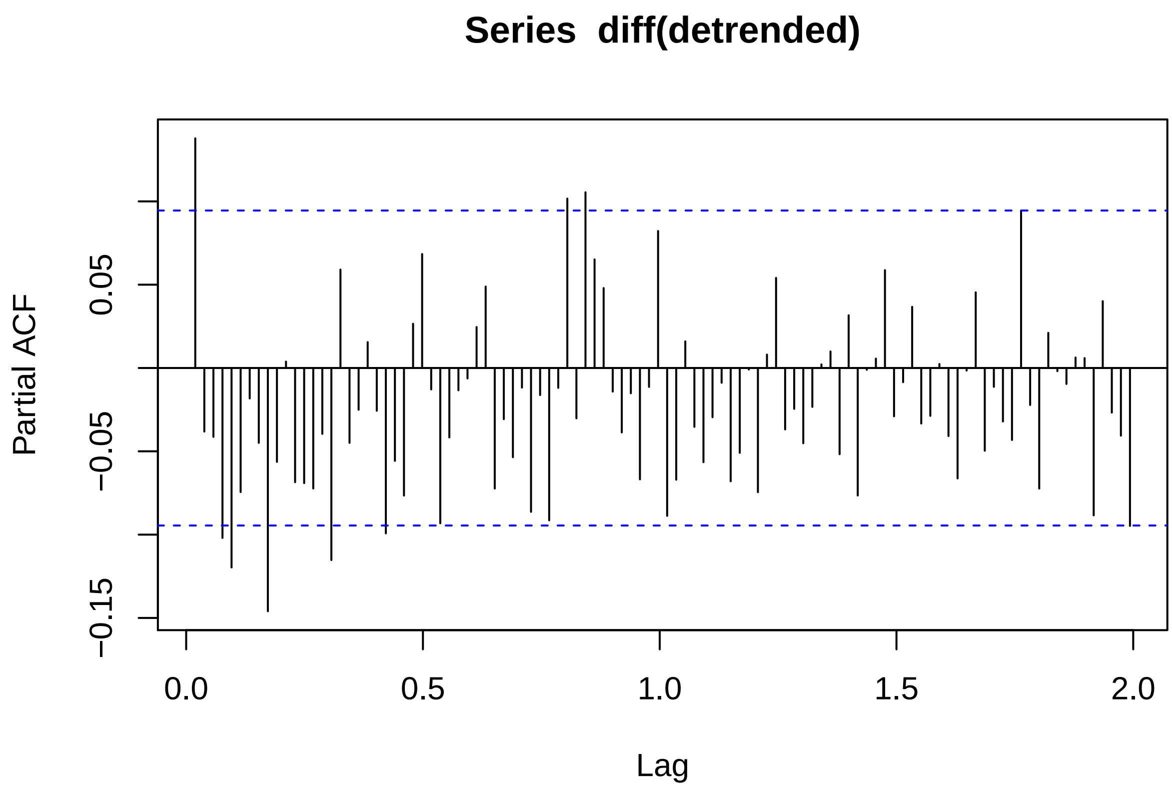


Weekly average snow depth, detrended
Snow depth (in)
−0.2
0.0
0.2
0.4
2012
2014
2016
2018
2020
Time

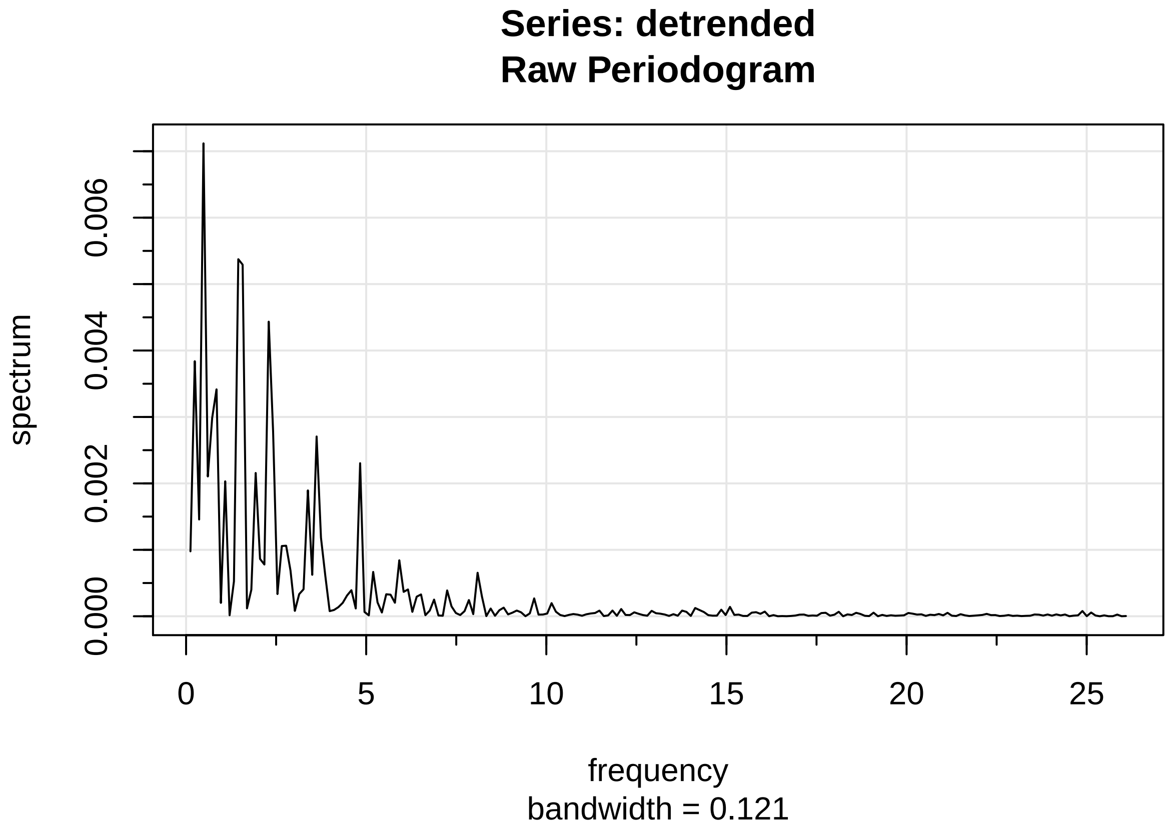

First difference
0.4
0.2
0.0
−0.2
diff(detrended)
2012
2014
2016
2018
2020
Time


0.4
0.2
0.0
−0.2
diff(detrended, differences = 2)
2012
2014
2016
2018
2020
Time

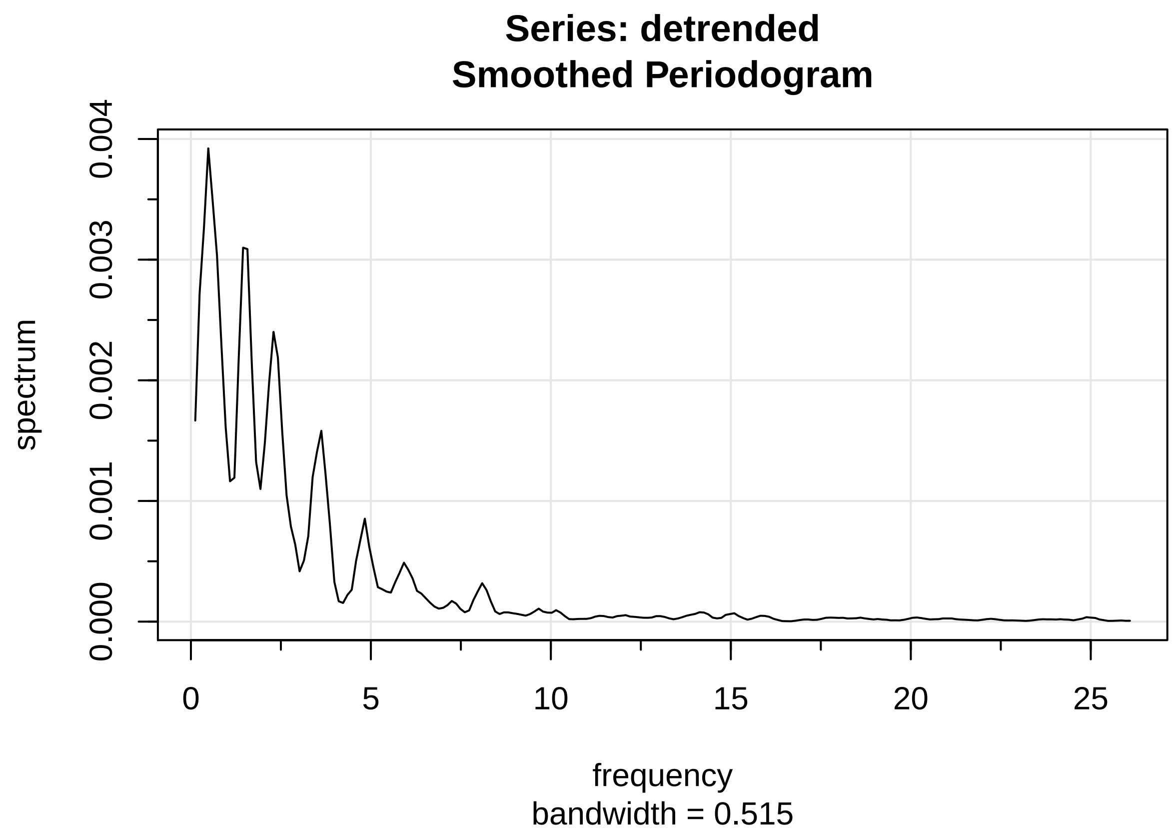

Series diff(detrended)
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
1.5
2.0
Lag



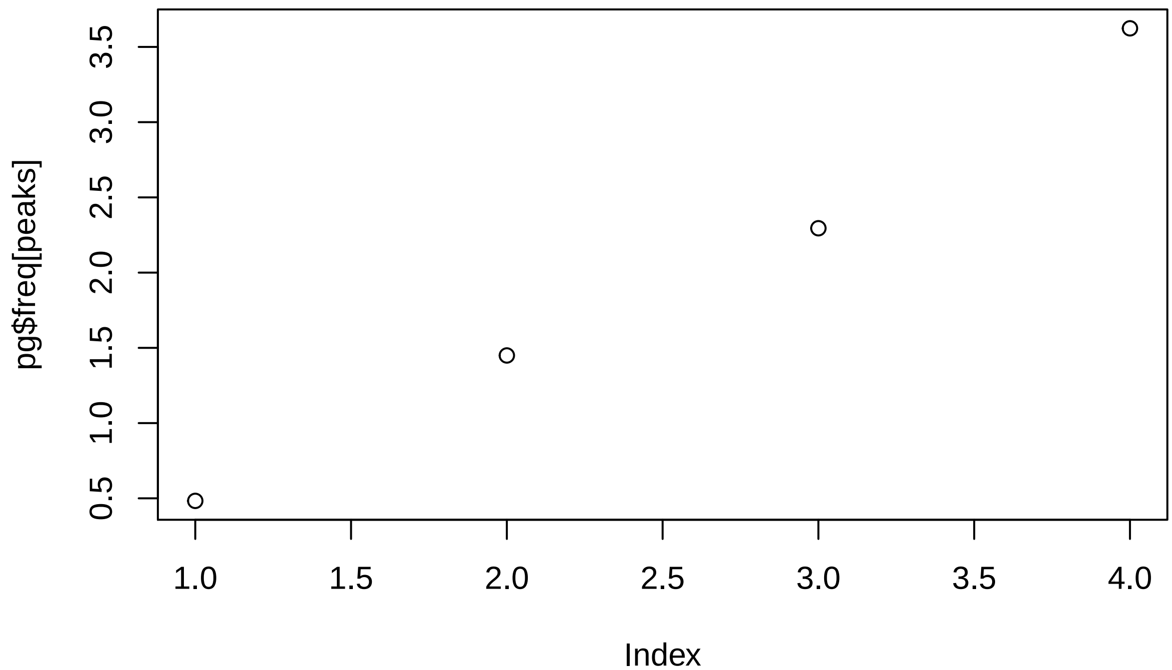

```
## Warning in adf.test(detrended): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  detrended
## Dickey-Fuller = -5.6557, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary

## Warning in adf.test(diff(detrended)): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff(detrended)
## Dickey-Fuller = -8.9512, Lag order = 7, p-value = 0.01
## alternative hypothesis: stationary
```

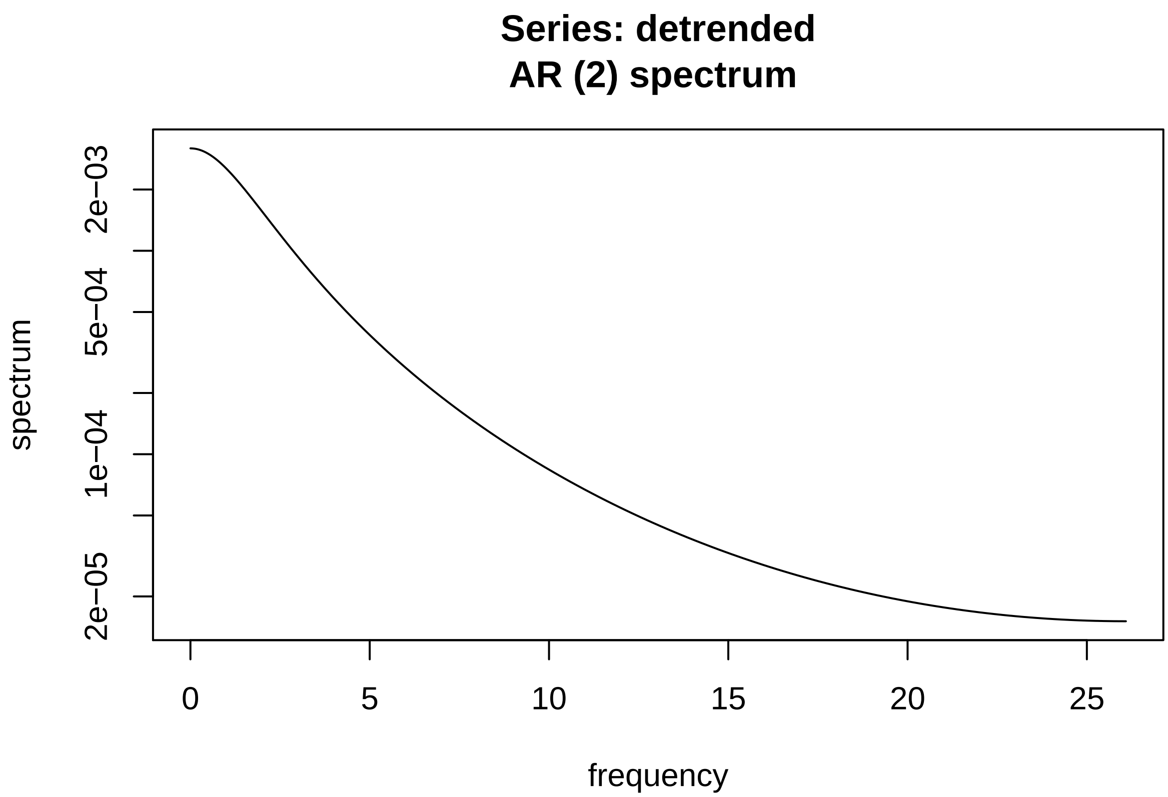


Series: detrended
Raw Periodogram
spectrum
0.000
0.002
0.004
0.006
0
5
10
15
20
25
frequency
bandwidth = 0.121

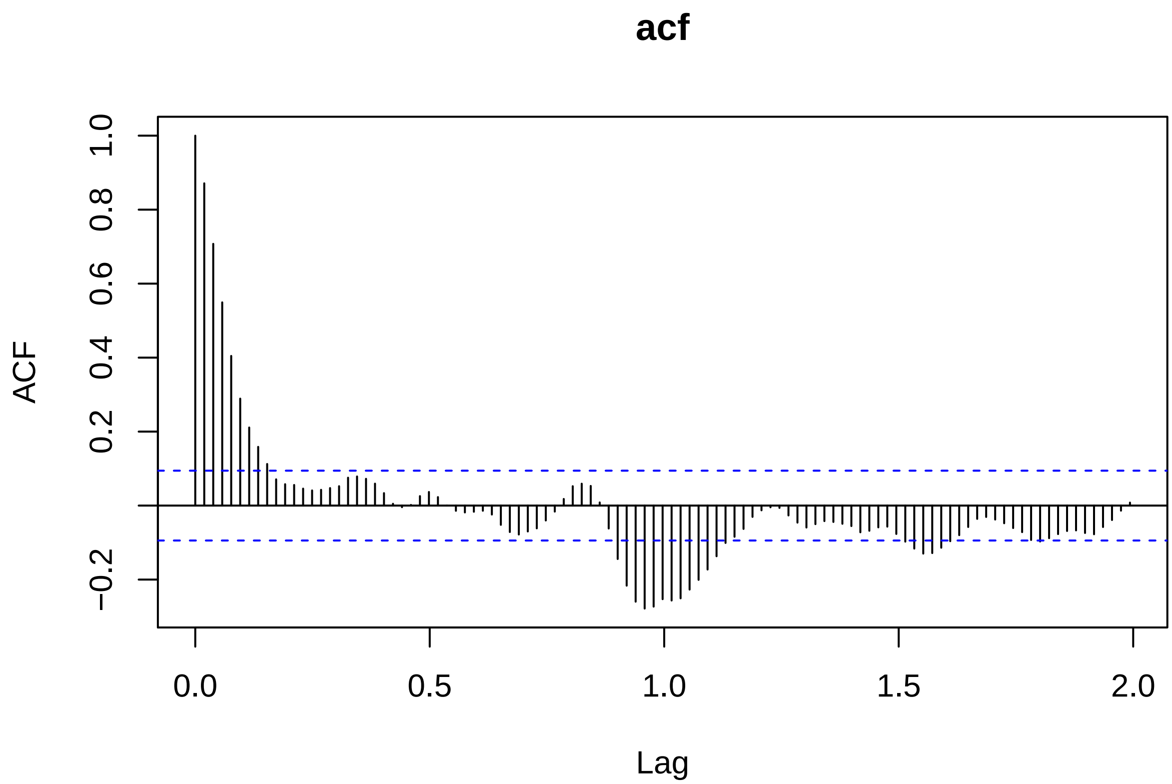


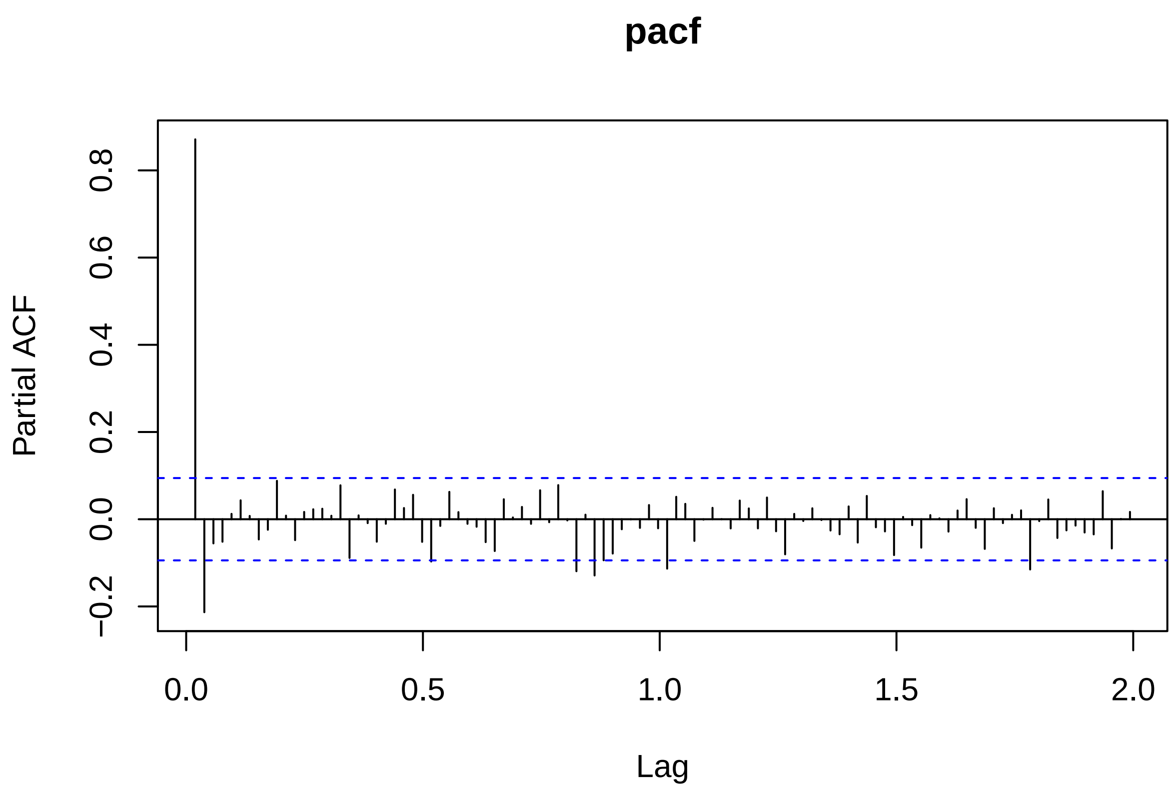

```
## [1] 2.0698152 0.6899384 0.4357506 0.2759754
```


pg$freq[peaks]
0.5
1.0
1.5
2.0
2.5
3.0
3.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Index


Series: detrended
AR (2) spectrum
spectrum
2e−03
5e−04
1e−04
2e−05
0
5
10
15
20
25
frequency

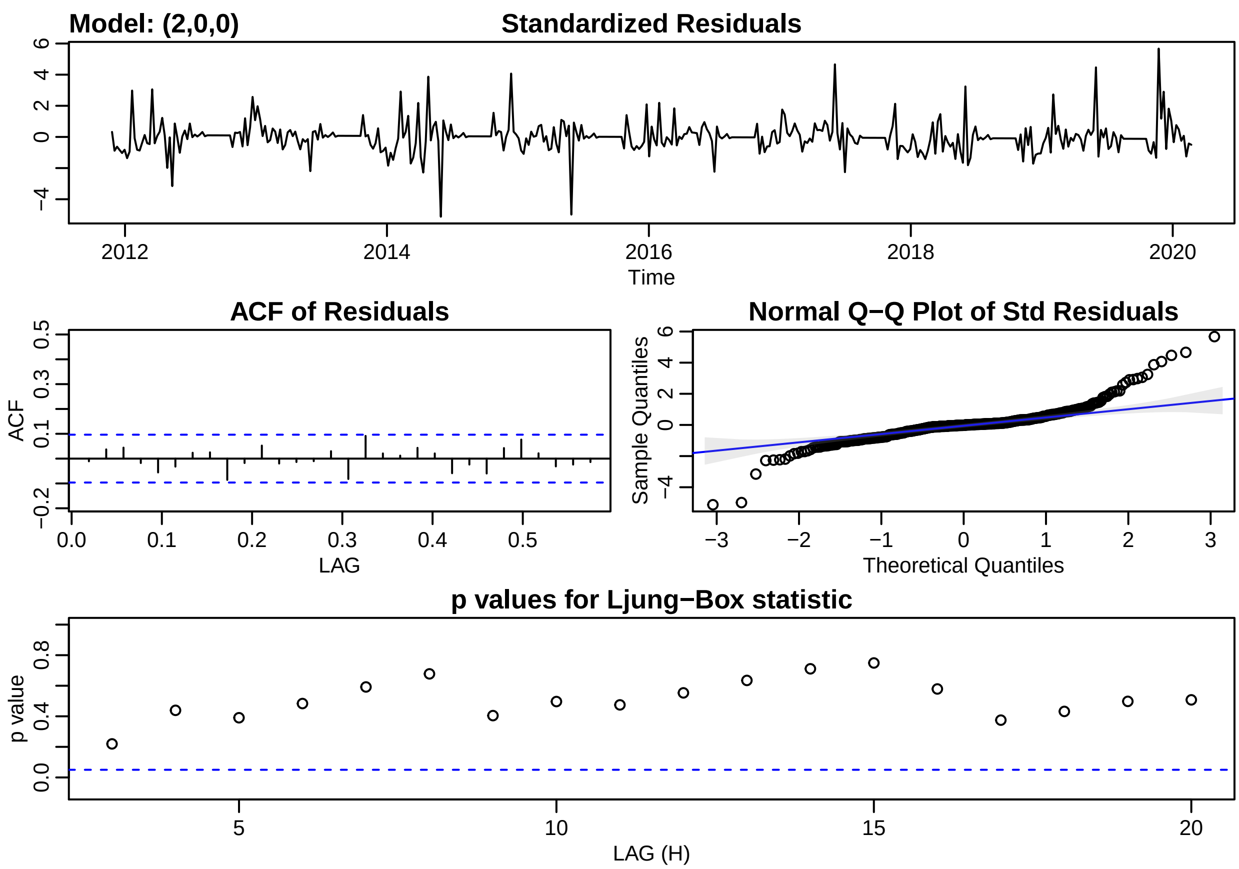
6 ACF


acf
ACF
1.0
0.8
0.6
0.4
0.2
−0.2
0.0
0.5
1.0
1.5
2.0
Lag


pacf
Partial ACF
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.5
1.0
1.5
2.0
Lag

7 Fit ARIMA

```
## initial  value -2.019724
## iter   2 value -2.122372
## iter   3 value -2.619679
## iter   4 value -2.701383
## iter   5 value -2.738825
## iter   6 value -2.756016
## iter   7 value -2.756759
## iter   8 value -2.756761
## iter   9 value -2.756761
## iter  10 value -2.756762
## iter  10 value -2.756762
## iter  10 value -2.756762
## final  value -2.756762
## converged
## initial  value -2.756290
## iter   2 value -2.756294
## iter   3 value -2.756296
## iter   3 value -2.756296
## iter   3 value -2.756296
## final  value -2.756296
## converged
```

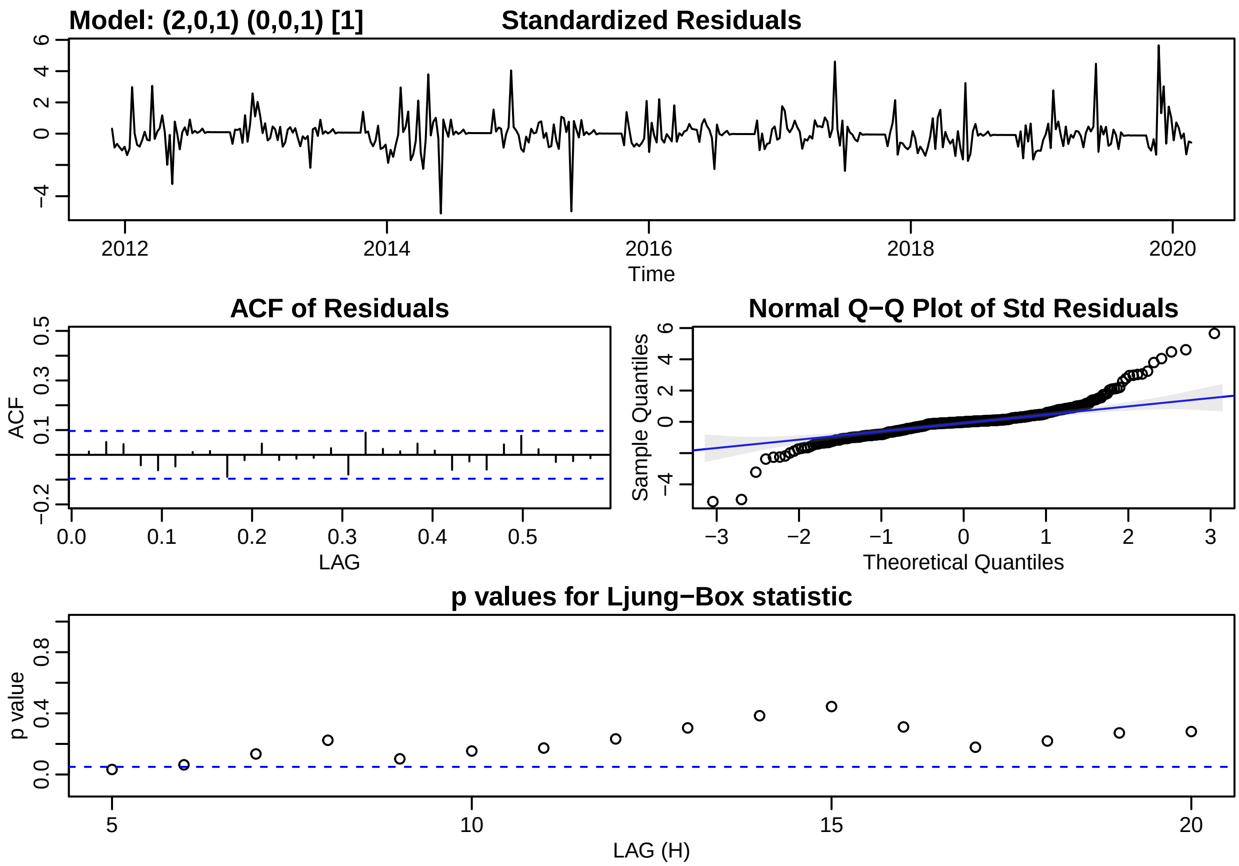



```
## $fit
```

```
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, transform.pars = trans,
##     fixed = fixed, optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ar2   xmean
##       1.0578  -0.2153  0.0000
## s.e.  0.0470   0.0470  0.0192
##
## sigma^2 estimated as 0.004022:  log likelihood = 576.4,  aic = -1144.8
##
## $degrees_of_freedom
## [1] 428
##
## $ttable
##       Estimate     SE t.value p.value
## ar1     1.0578 0.0470 22.5039  0.0000
## ar2    -0.2153 0.0470 -4.5814  0.0000
## xmean   0.0000 0.0192 -0.0023  0.9982
##
## $AIC
## [1] -2.656154
##
## $AICc
## [1] -2.656023
##
## $BIC
## [1] -2.618417
```

```
## initial  value -2.019724
## iter   2 value -2.515325
## iter   3 value -2.734814
## iter   4 value -2.751073
## iter   5 value -2.752822
## iter   6 value -2.752865
## iter   7 value -2.752934
## iter   8 value -2.752934
## iter   9 value -2.752937
## iter  10 value -2.752941
## iter  11 value -2.752947
## iter  12 value -2.752951
## iter  13 value -2.752952
## iter  14 value -2.752953
## iter  15 value -2.752954
## iter  16 value -2.752958
## iter  17 value -2.752994
## iter  18 value -2.752997
## iter  19 value -2.752999
## iter  20 value -2.753002
## iter  21 value -2.753012
## iter  22 value -2.753024
## iter  23 value -2.753040
```

```
## iter  24 value -2.753047
## iter  25 value -2.753050
## iter  26 value -2.753050
## iter  27 value -2.753050
## iter  28 value -2.753050
## iter  29 value -2.753051
## iter  30 value -2.753051
## iter  31 value -2.753051
## iter  32 value -2.753051
## iter  32 value -2.753051
## iter  32 value -2.753051
## final  value -2.753051
## converged
## initial  value -2.753656
## iter   2 value -2.753659
## iter   3 value -2.753668
## iter   4 value -2.753682
## iter   5 value -2.753709
## iter   6 value -2.753747
## iter   7 value -2.753764
## iter   8 value -2.753770
## iter   9 value -2.753771
## iter  10 value -2.753771
## iter  11 value -2.753773
## iter  12 value -2.753774
## iter  13 value -2.753774
## iter  14 value -2.753775
## iter  14 value -2.753775
## iter  14 value -2.753775
## final  value -2.753775
## converged
```

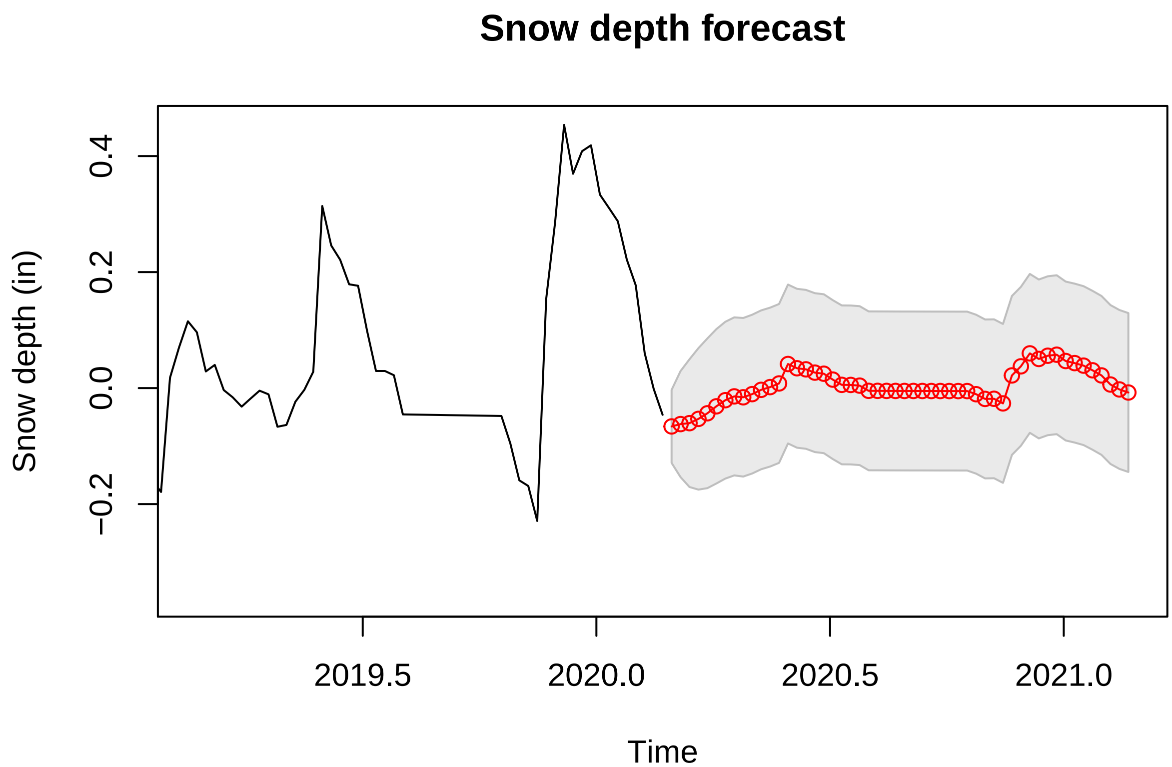


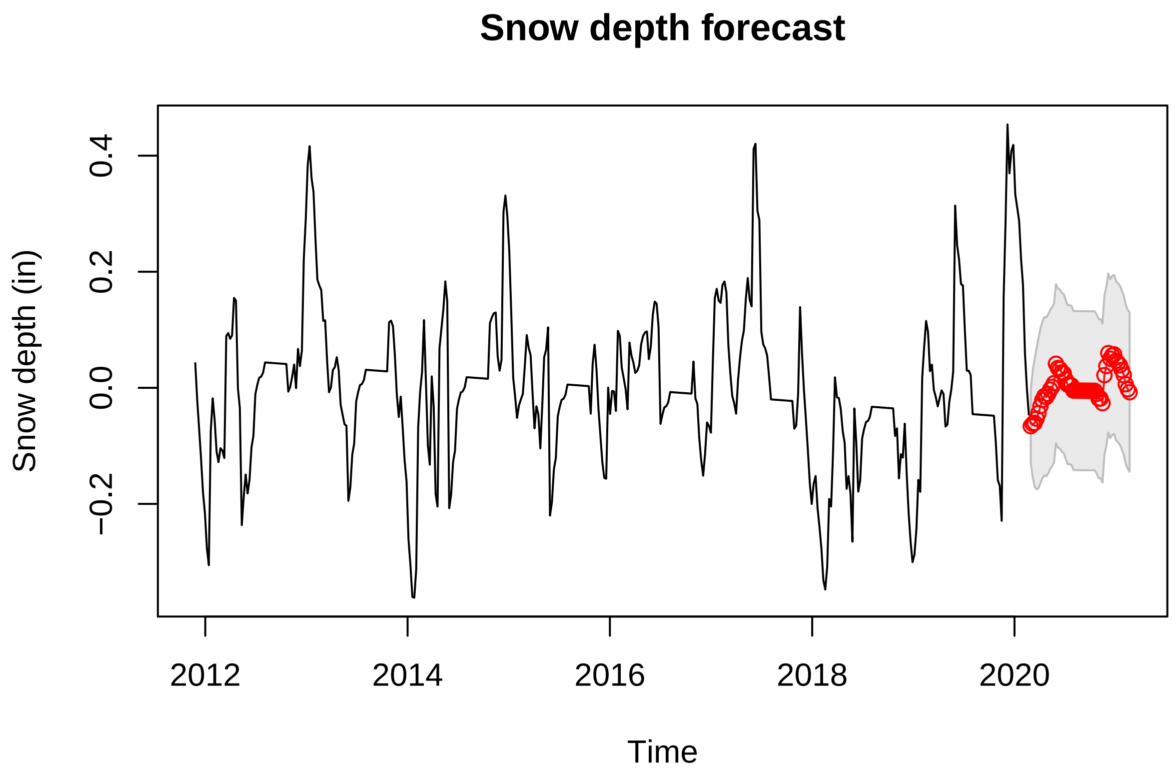

```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xmean, include.mean = FALSE, transform.pars = trans,
##     fixed = fixed, optim.control = list(trace = trc, REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1     ar2     ma1    sma1   xmean
##       0.2922  0.4271  0.3726  0.3726  0.0003
## s.e.  0.2118  0.1600  0.3585  0.3585  0.0203
##
## sigma^2 estimated as 0.004042:  log likelihood = 575.31,  aic = -1138.63
##
## $degrees_of_freedom
## [1] 426
##
## $ttable
##       Estimate     SE t.value p.value
## ar1     0.2922 0.2118  1.3793  0.1685
## ar2     0.4271 0.1600  2.6689  0.0079
## ma1     0.3726 0.3585  1.0393  0.2993
## sma1    0.3726 0.3585  1.0393  0.2993
## xmean   0.0003 0.0203  0.0131  0.9895
##
## $AIC
```

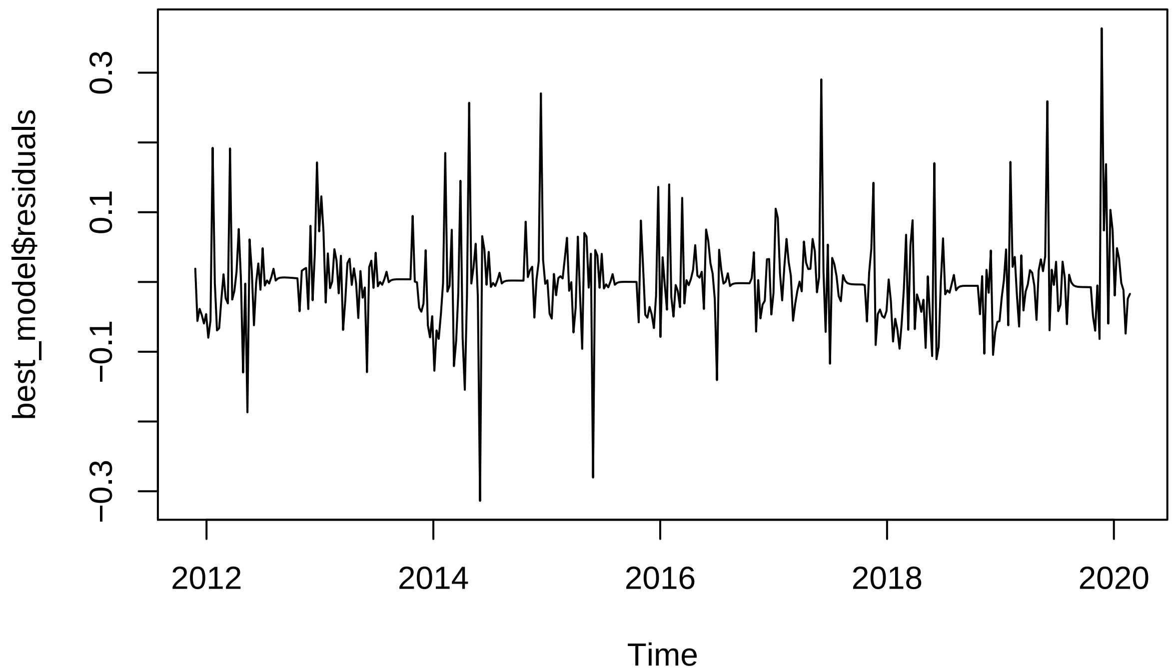
```
## [1] -2.64183
##
## $AICc
## [1] -2.641502
##
## $BIC
## [1] -2.585225

## [[1]]
##
## Call:
## arima(x = detrended, order = c(2, 0, 1), seasonal = list(order = c(0, 0, 1),
##     period = 52))
##
## Coefficients:
##          ar1      ar2      ma1    sma1  intercept
##       1.4321  -0.5351  -0.3756  0.1261     0.0004
## s.e.  0.1595   0.1381   0.1766  0.0549     0.0202
##
## sigma^2 estimated as 0.003945:  log likelihood = 580.04,  aic = -1148.08
##
## [[2]]
##
## Call:
## arima(x = detrended, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       1.0578  -0.2153     0.0000
## s.e.  0.0470   0.0470     0.0192
##
## sigma^2 estimated as 0.004022:  log likelihood = 576.4,  aic = -1144.8
##
## [[3]]
##
## Call:
## arima(x = detrended, order = c(2, 0, 2))
##
## Coefficients:
##          ar1      ar2      ma1      ma2  intercept
##       1.4077  -0.5153  -0.3646  -0.0195     0.0000
## s.e.  0.2471   0.1999   0.2495   0.0899     0.0173
##
## sigma^2 estimated as 0.004002:  log likelihood = 577.46,  aic = -1142.93
##
## [[4]]
##
## Call:
## arima(x = detrended, order = c(2, 1, 2))
##
## Coefficients:
##          ar1      ar2      ma1     ma2
##       1.3534  -0.4699  -1.3101  0.3102
## s.e.  0.1775   0.1536   0.1937  0.1936
```

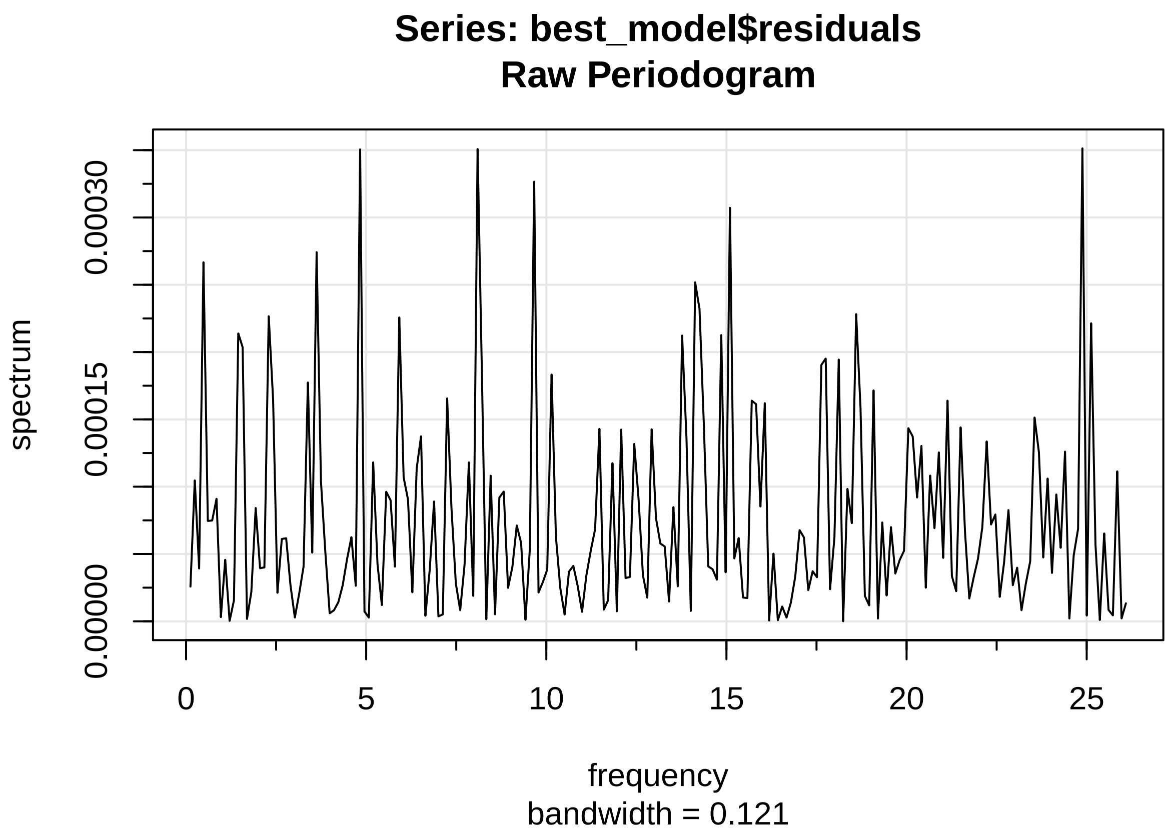
```
##
## sigma^2 estimated as 0.004012:  log likelihood = 574.32,  aic = -1138.65
```

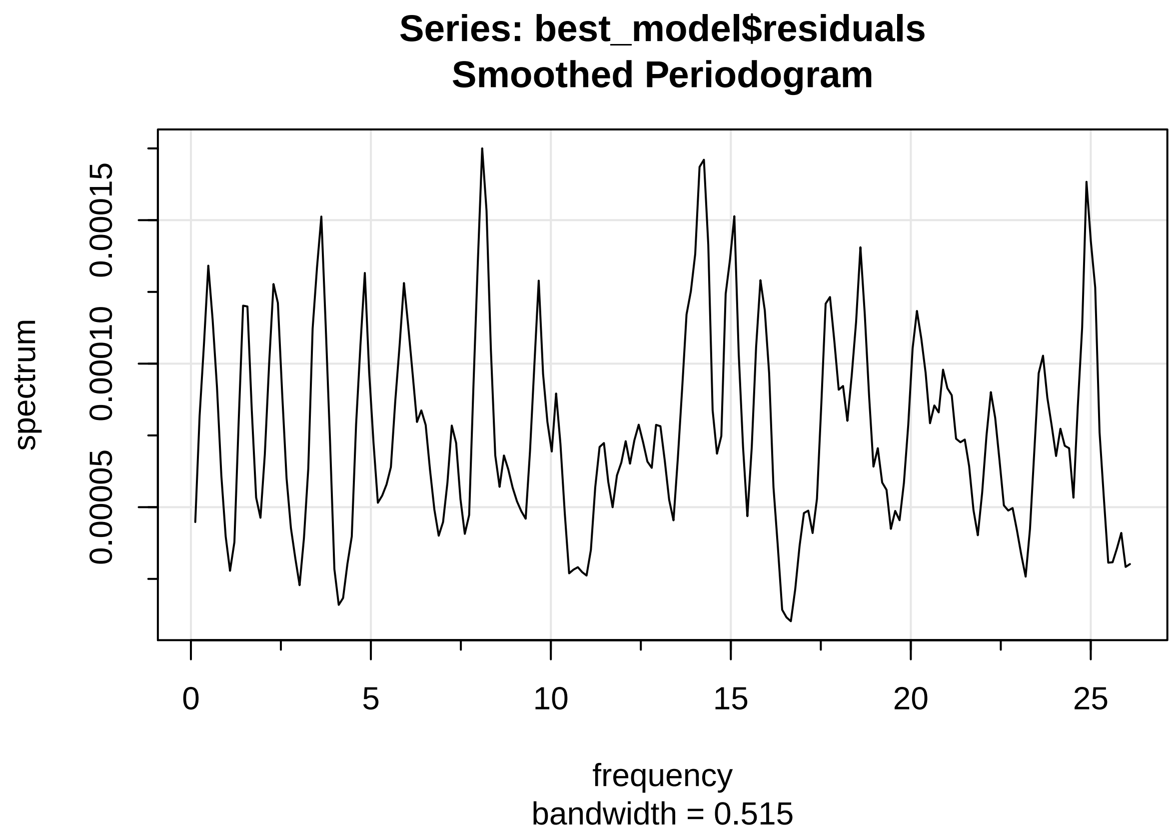
```
## [[1]]
## [1] -1148.077
##
## [[2]]
## [1] -1144.802
##
## [[3]]
## [1] -1142.925
##
## [[4]]
## [1] -1138.646
```



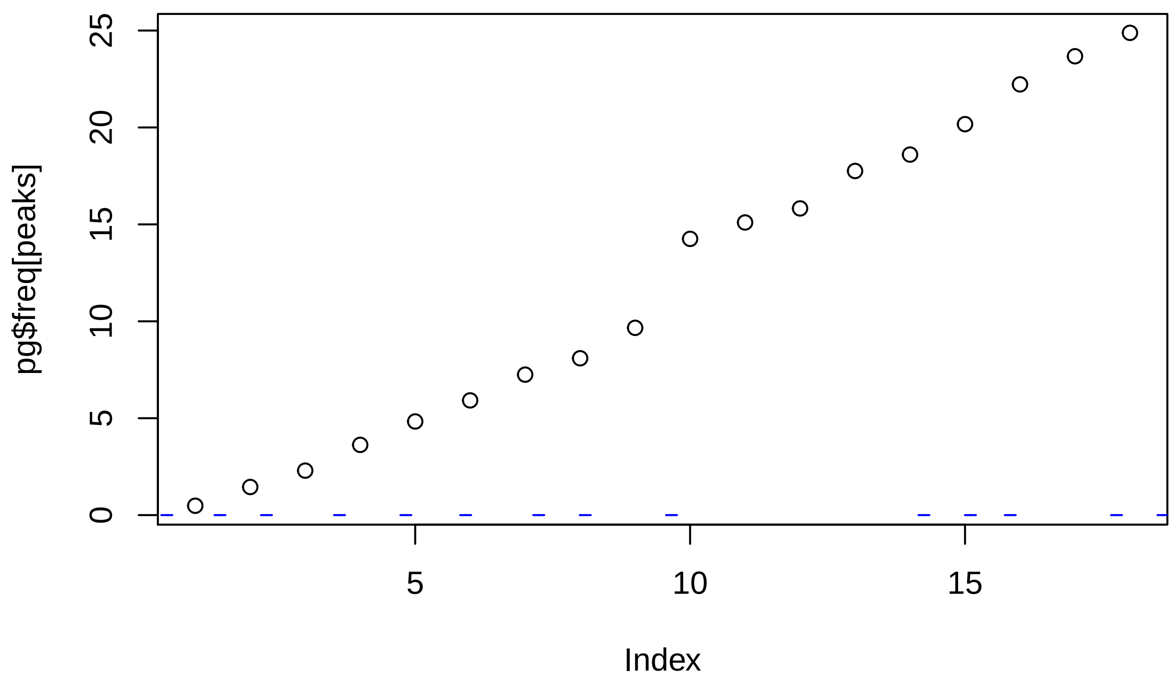

Snow depth forecast
Snow depth (in)
0.4
0.2
0.0
−0.2
2012
2014
2016
2018
2020
Time


best_model$residuals
−0.3
−0.1
0.1
0.3
2012
2014
2016
2018
2020
Time

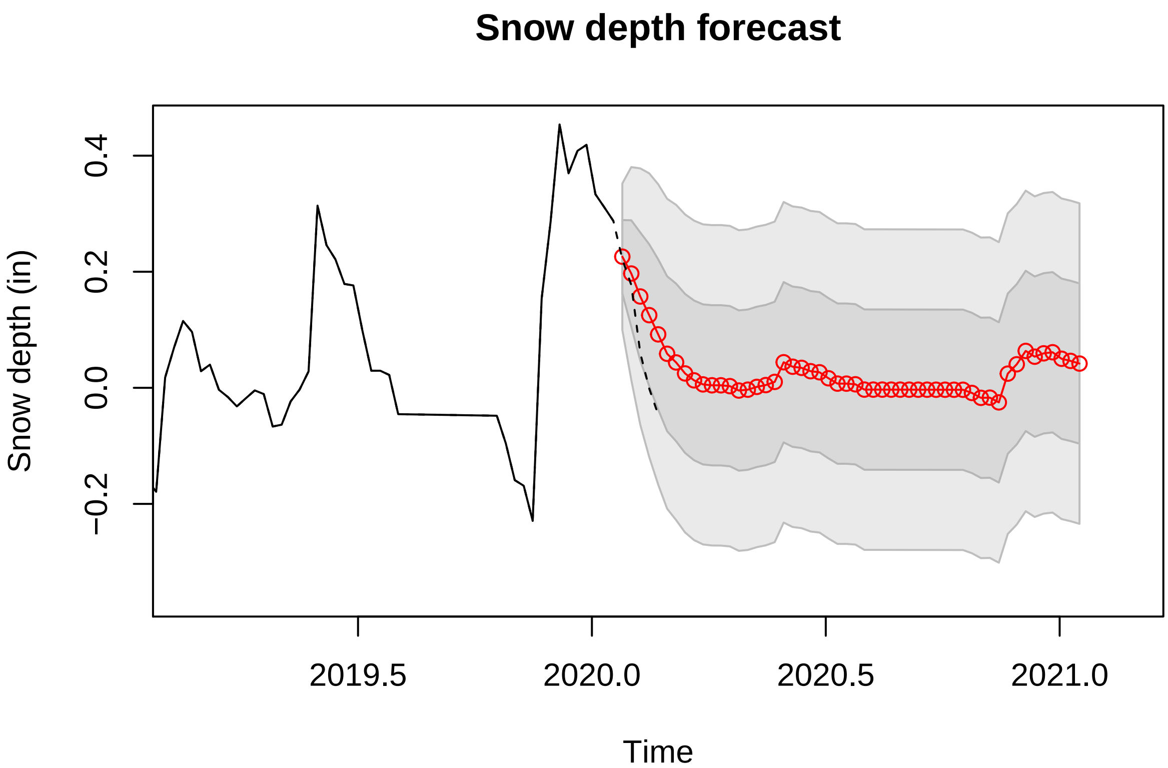

Series: best_model$residuals
Raw Periodogram
spectrum
0.00000
0.00015
0.00030
0
5
10
15
20
25
frequency
bandwidth = 0.121




```
##  [1] 2.06981520 0.68993840 0.43575057 0.27597536 0.20698152 0.16896451
##  [7] 0.13798768 0.12357106 0.10349076 0.07016323 0.06623409 0.06320046
## [13] 0.05632150 0.05376143 0.04957641 0.04499598 0.04224113 0.04019059
```



pg$freq[peaks]
0
5
10
15
20
25
5
10
15
Index


Snow depth forecast
Snow depth (in)
0.4
0.2
0.0
−0.2
2019.5
2020.0
2020.5
2021.0
Time