Analysis of Traffic Through Fastrak

Clark Fitzgerald, Oliver Paisley

March 2, 2015

Almost done. 1. Error in boxcox 2. Error in inverse.rle 3. Put labels on all plots 4. forecast 5. Spectral write-up 6. Interpret ACF/PACF 7. Put comments in appendix

• Style: The write-up should take the form of a scientific paper. The following parts should be included: (1) Title, author(s) and abstract; (2) Introduction; (3) Data description; (4) Data analysis; (5) Discussion; (6) Conclusions; (7) References; (8) Appendix.

• Title, author(s) and abstract: Find a meaningful title and provide your names. The abstract should contain a concise summary of what is to come in the paper.

• The exposition should be clear and easy to follow. Please check spelling and grammar carefully.

• Data visualization: In all the following steps (also in the Data Analysis part below), make sure to provide appropriate plots in order to motivate your choices. This is a necessary non-mathematical, common sense approach to the data analysis.

• Define the goal: Make sure there is a clear aim for your time series analysis. Writing "I would like to forecast the next value" is not ideal. Have a good motivation as to why you chose your data and why it should be useful to look at prediction of future values. What is the information you would like to extract?

Abstract

TODO:

Introduction

** TODO: ** Be more clear on the main goal of the project. Expand.

• Introduction: This part should broadly answer the questions: What is the problem considered? Why is it interesting? What is the proposed solution (and how does it relate to existing literature)? How is the rest of the write-up organized?

Fastrak is the California highway toll system. Thousands of cars pass through Fastrak stations each day, and the data generated is incredibly valuable. By analyzing the number of cars that pass through Fastrak we can gain insight into the flow of traffic throughout California. This information would be useful for any entity interested in traffic flow, of which there are many.

This article will be will include three main parts. The first part will be an extensive cleaning of the data. It was quite dirty and there were a multitude of issues. The second part will be a time series analysis of the data, including prediction and a spectral analysis. The third part will be a discussion of our findings in part two.

Data Description

• Data description: A detailed introduction to the time series data should be given. What is it that you are looking at? Where did it come from? What are the interesting features? Give summary statistics and plots to make your points.

The data consists of counts of the number of cars that pass through a given Fastrak station per hour. The data was acquired through the Caltrans Performance Measurement System, which is apart of the California Department of Transportation. We have counts for roughly 6 years worth of data across 150 stations, and there are about 8 million observations.

Here is what the data looks like for the Fastrak station in Santa Rosa (station 4300).

	time	station	count	year	month	weekday	hour	residuals
25	2007-03-23 03:00:00	4300	32	2007	March	Friday	03	-2.4031615
170	2007-03-23 04:00:00	4300	35	2007	March	Friday	04	-2.9898080
315	2007-03-23 05:00:00	4300	103	2007	March	Friday	05	-1.5040558
460	2007-03-23 06:00:00	4300	231	2007	March	Friday	06	-0.7732094
605	2007-03-23 07:00:00	4300	357	2007	March	Friday	07	-0.8023823
750	2007-03-23 08:00:00	4300	411	2007	March	Friday	08	-1.5260884
[](Dr	aft_files/figure-latex	/unnamed-c	hunk-1-1	.pdf)				

This plot indicates that a cubic polynomial will adequately approximate the shape of the long term trend. A cubic polynomial fit is preferable to a fit based on the categorical variable *year* since the cubic polynomial is smooth, and therefore it will not contribute to any discontinuities in the time series.

Based on experience and common sense, we know that there are a few elements that primarily determine traffic flow. These elements include:

- The time of day.
- The day of the week.

There is also the possibility of monthly or annual trends.

Based on the above, we have developed the following general linear model:

$count = time + time^2 + time^3 + month + weekday + hour + weekday : hour + \epsilon$

time is a continuous variable. *hour*, *weekday*, and *month* are categorical variables. $weekday : hour$ represents the interaction between *weekday* and *hour*.

Note that the three terms in the *time* polynomial are orthogonal, which is the default in the **poly()** function.

Now we can represent our model in terms of the classic linear time series decomposition.

Let $Y_t = count$, $m_t = time + time^2 + time^3$, $s_t = month + weekday + hour + weekday : hour$, and $X_t = \epsilon$.

We can now write our model as

$Y_t = m_t + s_t + X_t$.

We can now do our time series analyis on the residual series (X_t).

Data Analysis

** TODO: ** Re-write. Write more. • Data analysis: This section should include the remaining items from the Data Processing and Data Analysis sections above.

Significant time was spent preparing this data. There were three major issues:

- Missing data.
- Too many counts of zero.

- A month long period where the traffic was double what it should have been across all stations.

We dealt with these issues by writing a small library of tested functions to prepare the data in a disciplined, repeatable way.

Since we have an abundance of data, we dropped all of the observations that gave us major issues. It would be certainly be possible to interpolate or backcast to fill in the missing data as well.

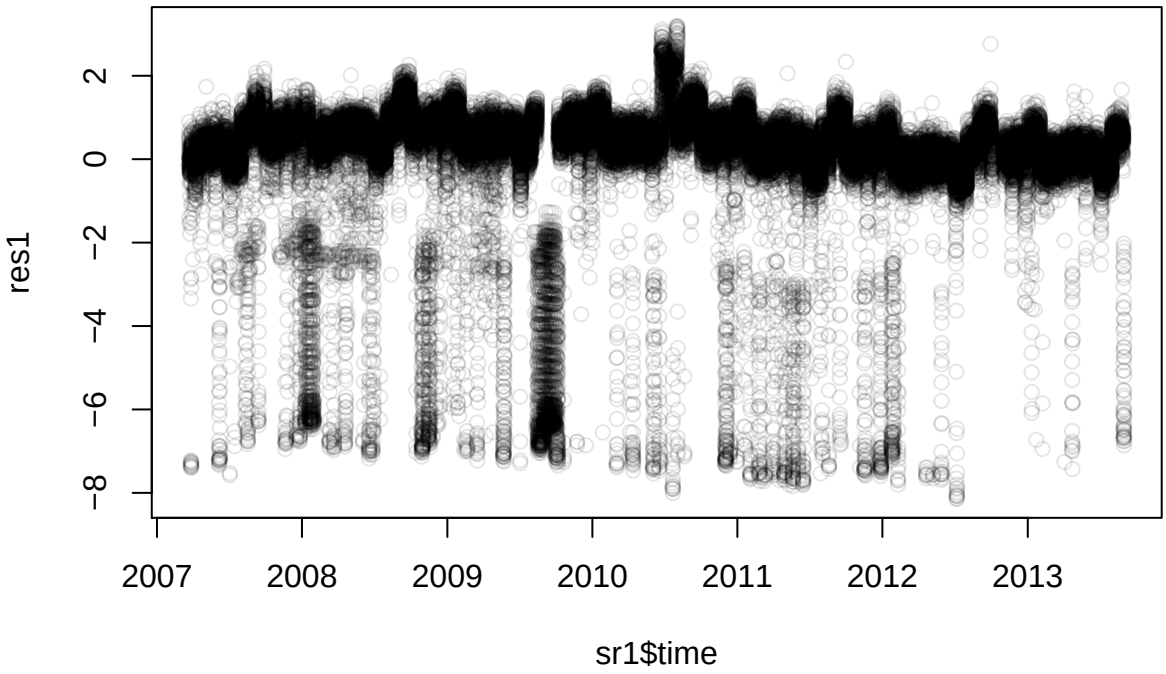
Variance Stabilizing Transform

• Initial data transformations: Check if it is necessary to stabilize the variance of the data or if transformations to symmetry are necessary. Then start modeling with the classical decomposition of a time series into trend plus seasonality plus stationary errors.

The box cox method indicates using the cube root function on the counts as a variance stabilizing transformation. Therefore, we will be using this transformation for further analysis

Outlier Detection and Removal

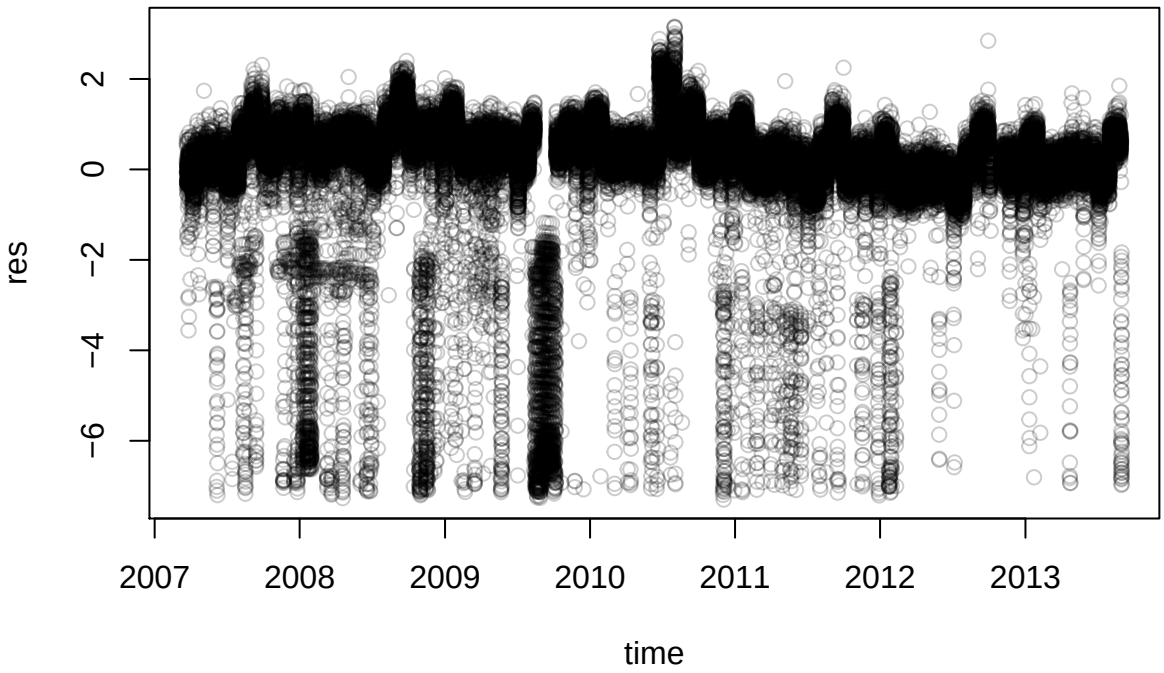
Our initial plot identified quite a few suspicious observations where the sensor may have stopped working.



There are quite a lot of long tails. We are going to remove the bottom 1% of residuals in hopes of having more consistency. For symmetry we are also going to remove the top 1%.

```
##        1%
## -6.922812
```

```
## [1] 54517     8
```



This removed about 500 values below the bottom threshold and about 5 values from the top threshold. There are still tails, but they are not as severe.

Interpretation

Inspecting the corresponding ANOVA table tells us much about the relative sources of variation in the count data.

To determine the sources of variation in our count data we will look at the corresponding R^2 value and ANOVA table of our model.

```
## [1] 0.6039812
```

This model accounts for 97% of the variation in the cube root of counts.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	percent
hour	23	149762.555	6511.415426	3137.24106	0	52.416438
weekday	6	5742.426	957.070919	461.12281	0	2.009832
month	11	6173.351	561.213725	270.39632	0	2.160654
poly(time, 3)	3	5657.312	1885.770736	908.57625	0	1.980042
hour:weekday	138	5607.593	40.634734	19.57807	0	1.962641
Residuals	54335	112773.533	2.075523	NA	NA	39.470393

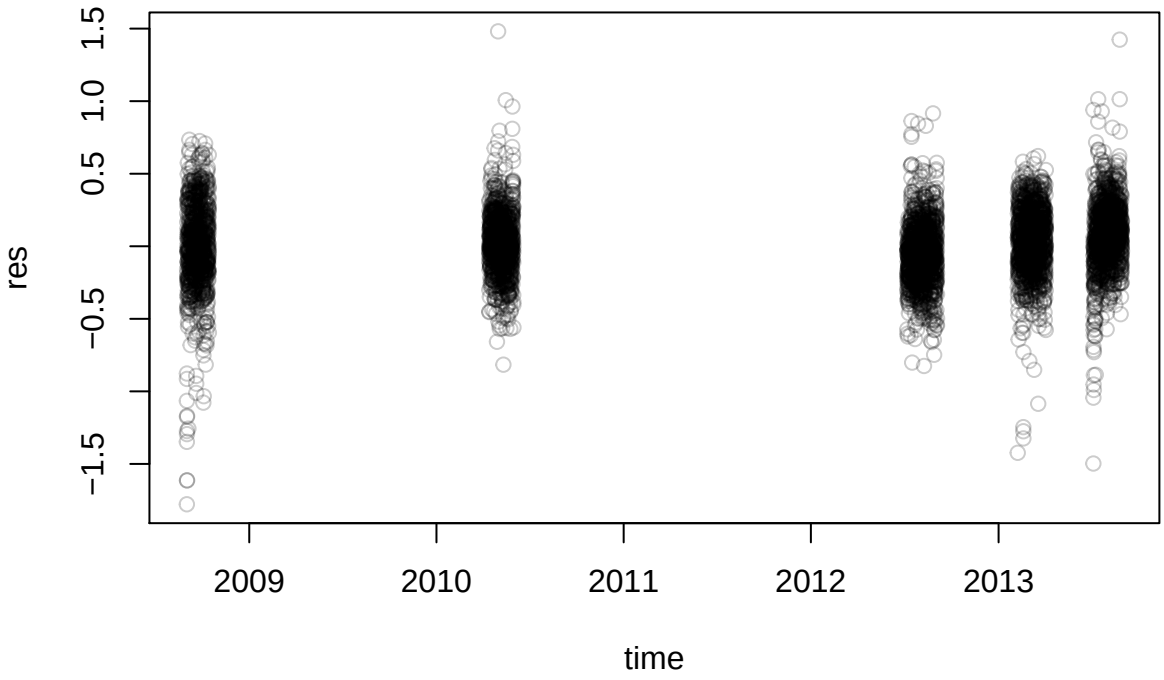
We can see that *hour* alone accounts for about 89% of the variation, while *month* and the long term trend of time account for only 0.14% and 1% of the variation, respectively.

This tells us that traffic through Fastrak stations changes mostly based on the time of day, but remains relatively constant throughout a long period of time.

Finding Consecutive Runs

** TODO: ** Write more. Make that thing a function. Test it. Maybe use testthat package.

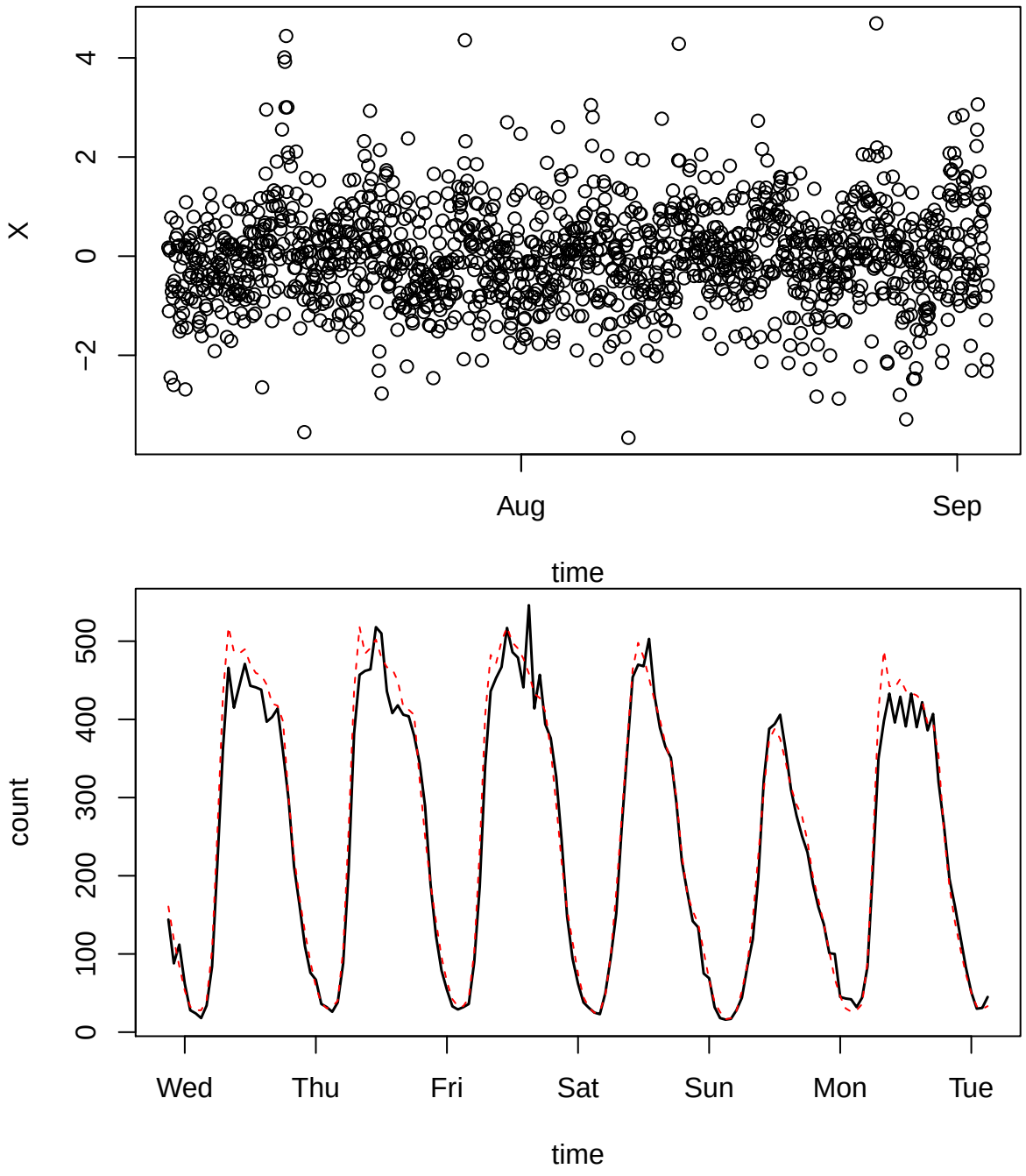
To deal with the issue of missing data and counts of zero we found as many consistently spaced residuals that were longer than 1000 on the filtered data. There were five groups.

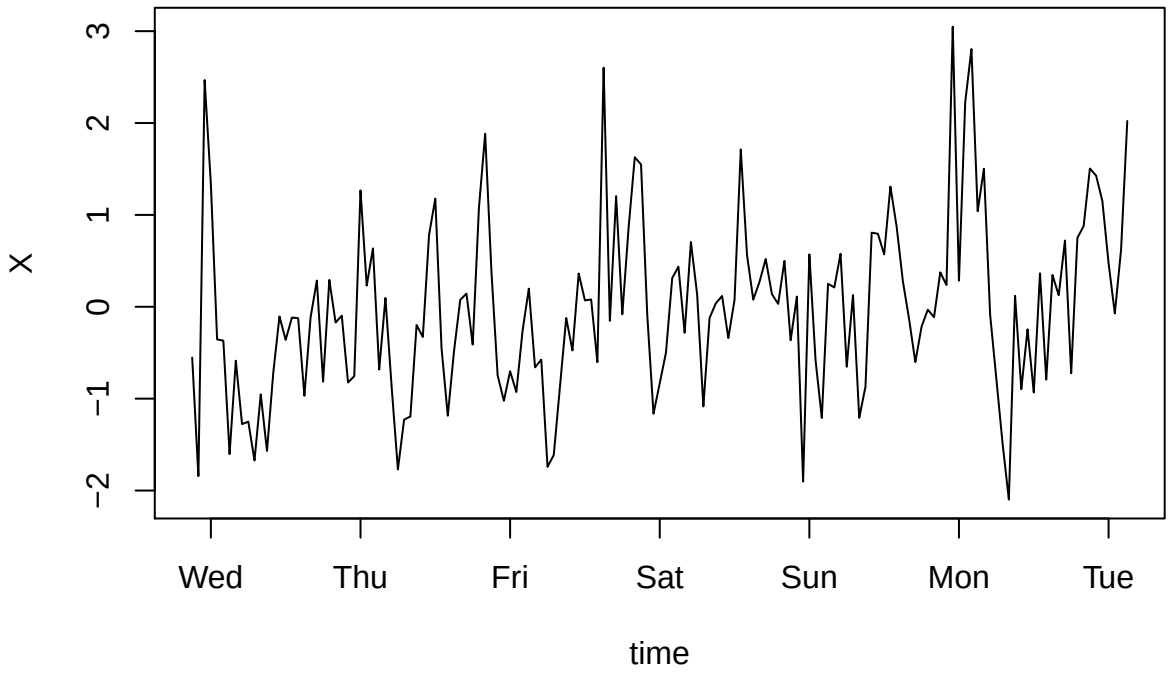


Common Sense Check

** TODO: ** Write more.

We are going to focus on the third group for the rest of the analysis as it seems to be the most consistent. For ease of further analysis we center and scale the residuals. We also confirm our fit worked correctly by plotting one week's worth of data. The final plot is the residuals.


X
4
2
0
−2
Aug
Sep
time
count
500
400
300
200
100
0
Wed
Thu
Fri
Sat
Sun
Mon
Tue
time

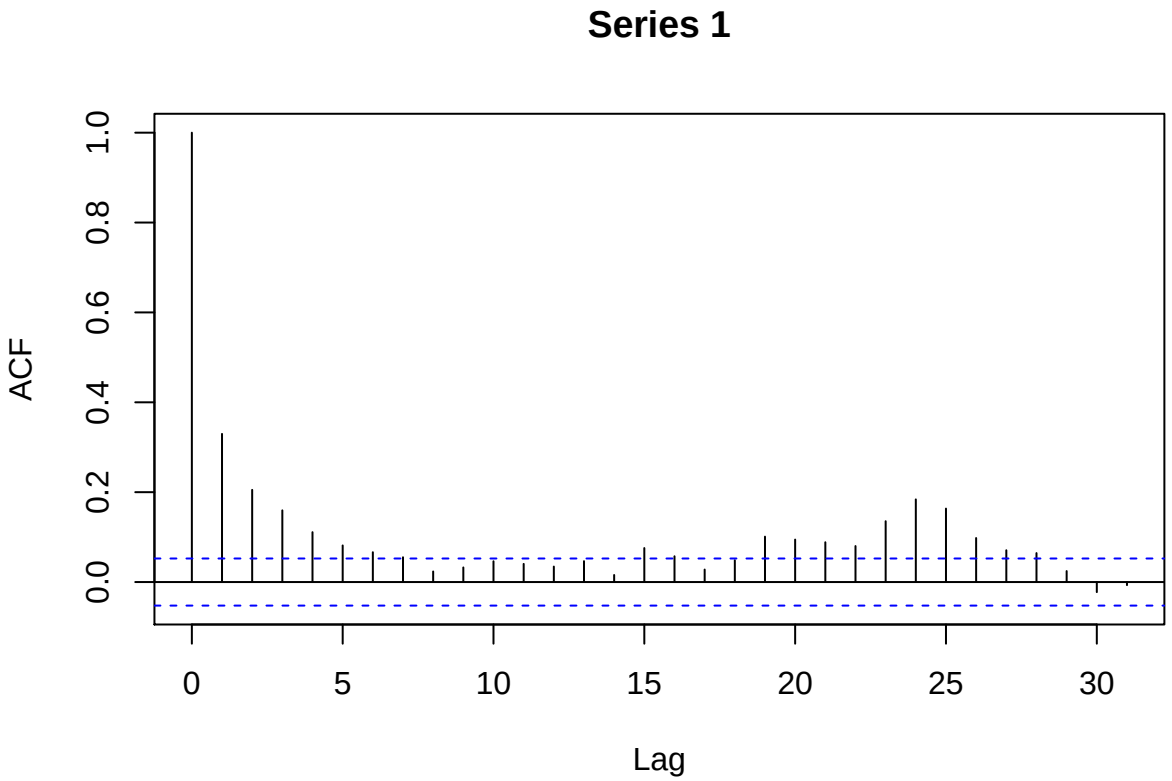


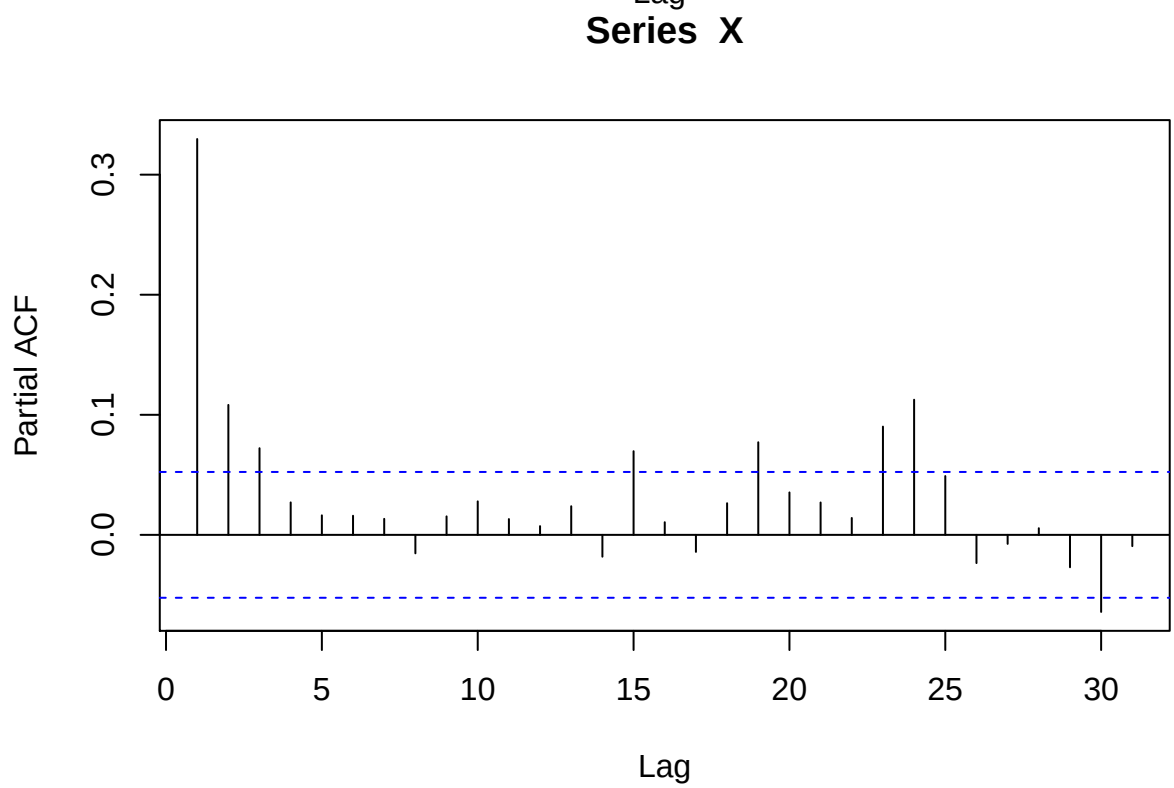
Time Series Analysis

** TODO: ** Interpret ACF and PACF. Write more. Maybe auto.arima? Maybe look at cool time series packages? • Analyzing the "smooth" component: Perform a trend and seasonality analysis, choosing from the methods given in Sections 1.3 and 1.4 of the Lecture Notes. After this step, the resulting residuals should pass as stationary time series. • Analyzing the residuals: Check the residuals for whiteness, remaining trends and normality as outlined in Section 1.5 of the Lecture Notes. Hopefully your residuals display some stationary time series features. • Analyzing the "rough" component: Fit stationary ARMA models to the residuals obtained after the analysis of the smooth component, as outlined in Chapter 3 (see Section 3.6 for a summary) of the Lecture Notes. Make sure to provide ACF and PACF plots for support. Check if the residuals conform to white noise. If not, there is still dependence left in the data that could be utilized for improved modeling.

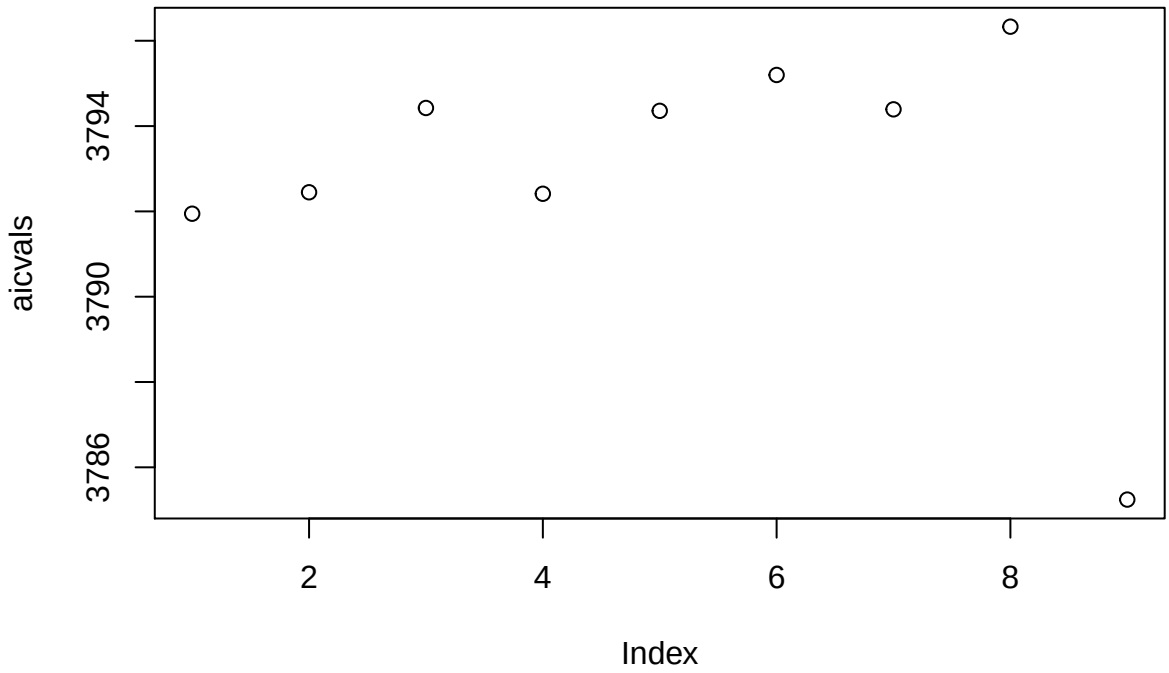
Now that all of the cleaning and processing is out of the way, we can finally begin a time series analysis on our residual series X.

We will begin by looking at ACF and PACF plots.





To find the best ARMA model we searched over a grid and used AIC as a determining factor.

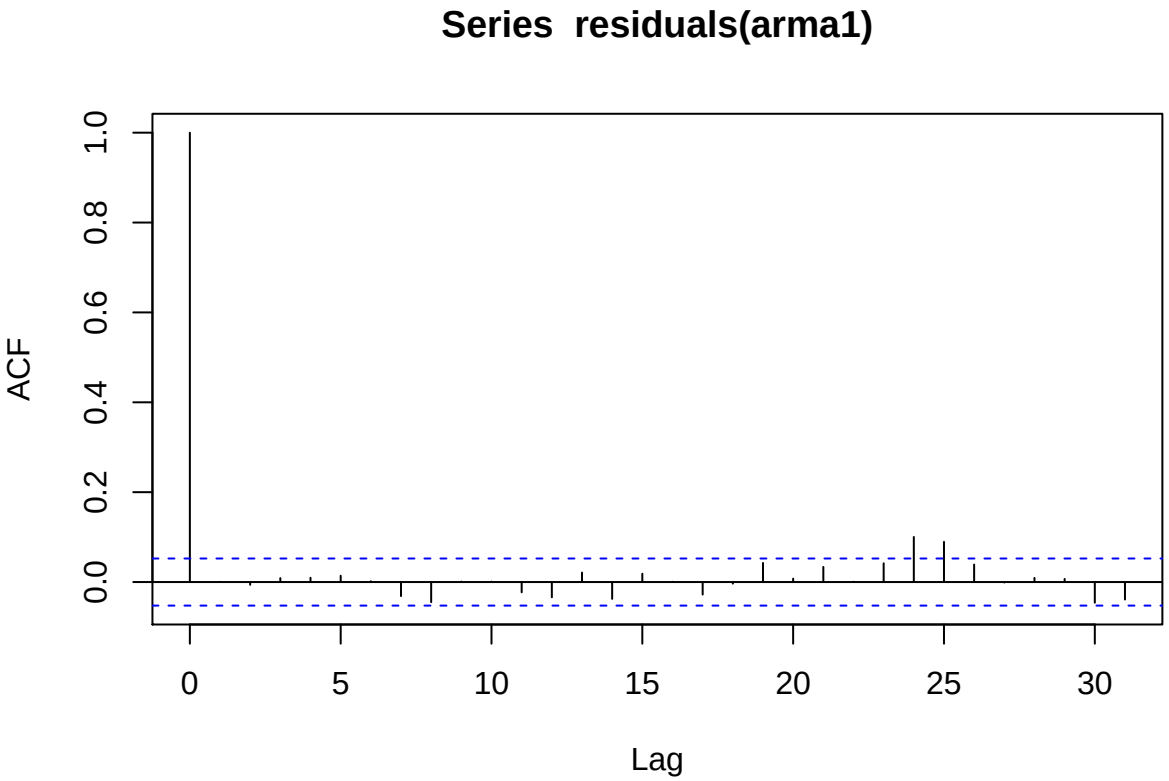


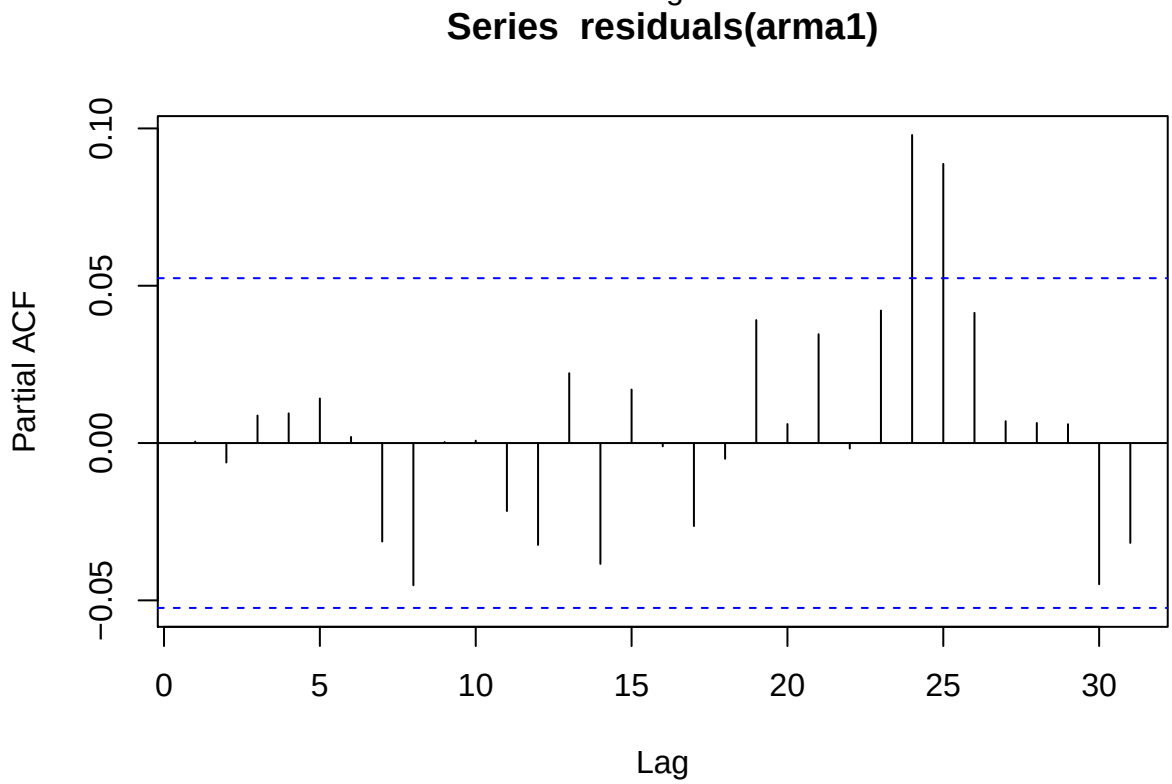
We needed to extend the maximum number of iterations in order to compute accurate AIC values for the larger models. Surprising how fast this becomes computationally expensive.

```
##   ar ma
## 9  3  3
```

```
## [1] 3776.977
```

The AIC criteria chose an ARMA(6, 6) model. We will now check to see if the residuals resemble white noise.

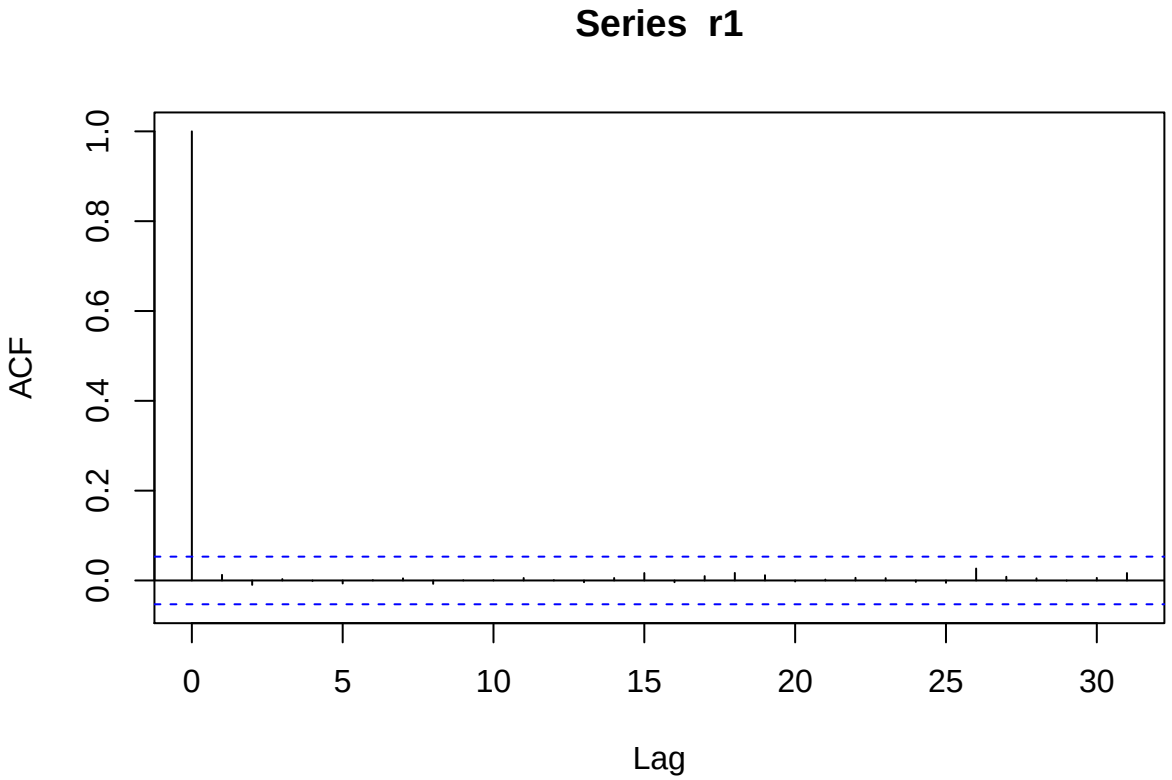


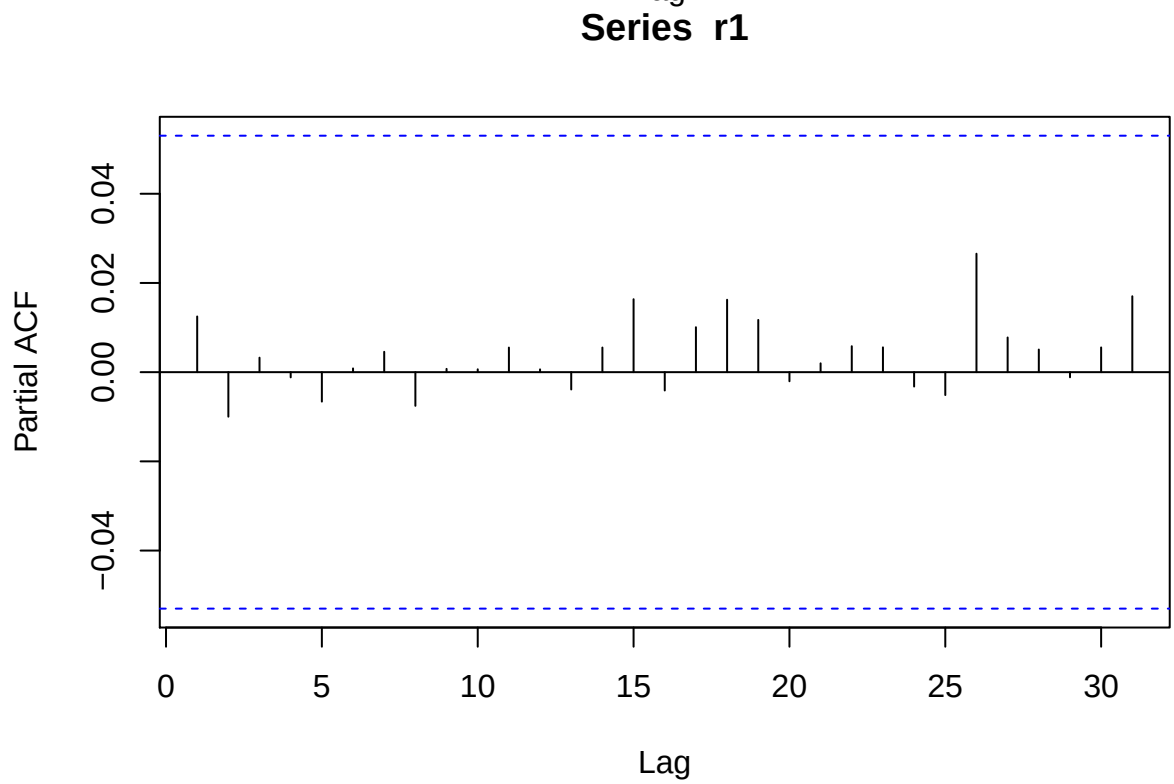


We still see spikes at lag 24. This makes sense because they represent the following day. We will now compare this with a larger AR model.

```
## [1] 32
```

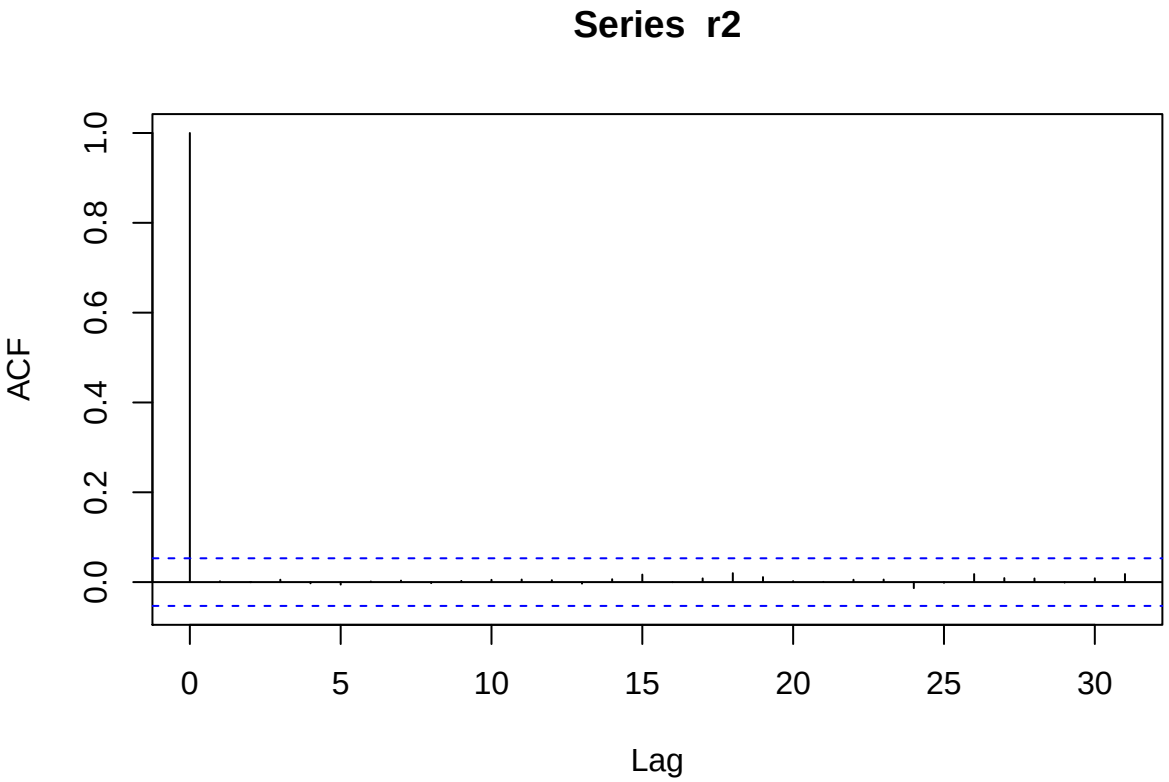
Yule-Walker is the default. We'll bump the maximum order from 31 up to 35. The AR model chose an AR(32) model based on AIC. Let's see how these residuals look.

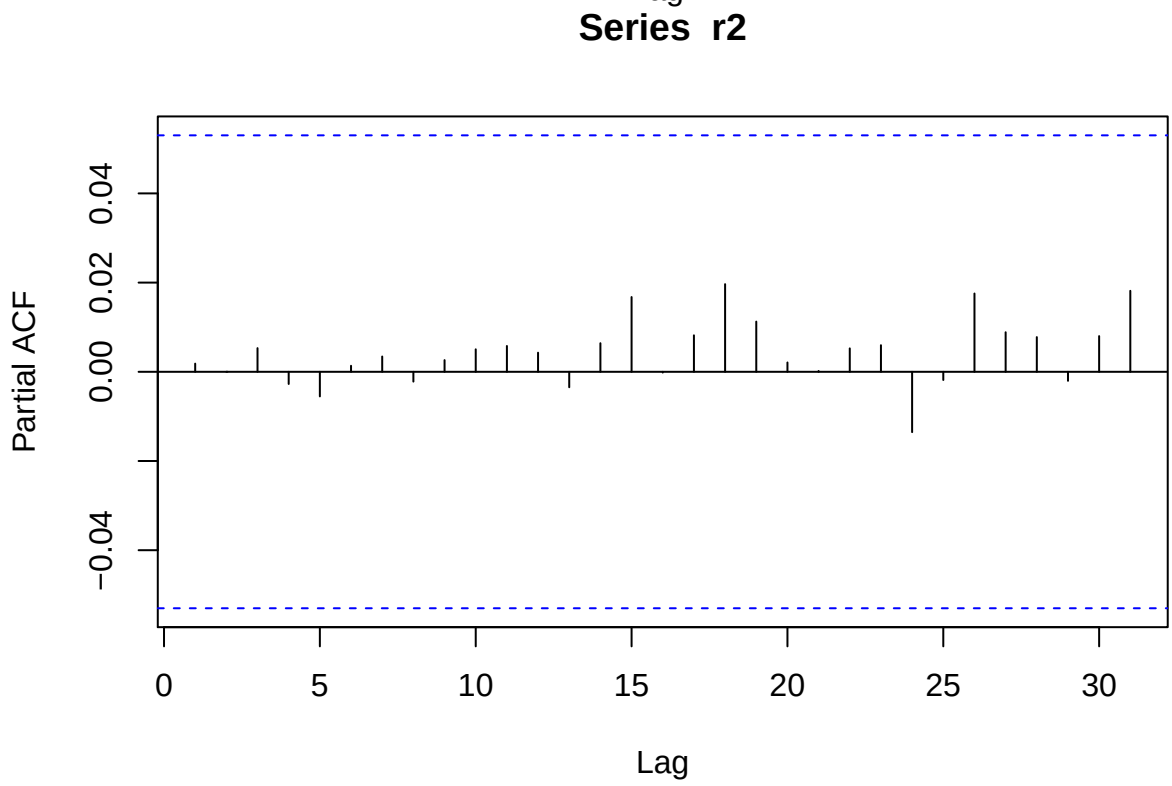




They look more like white noise than the ARMA(6, 6) model. I wonder if the fit looks any different with other algorithms? The default uses Yule-Walker, but there's also MLE and OLS to try.

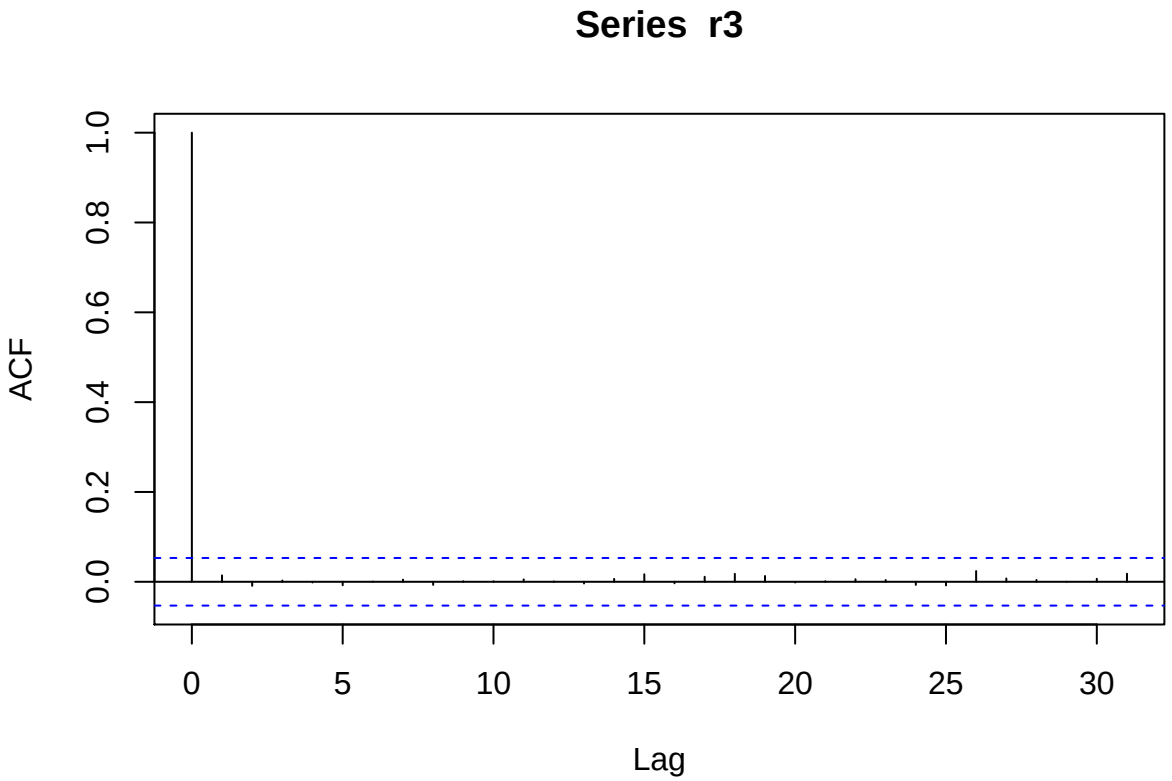
```
## [1] 32
```

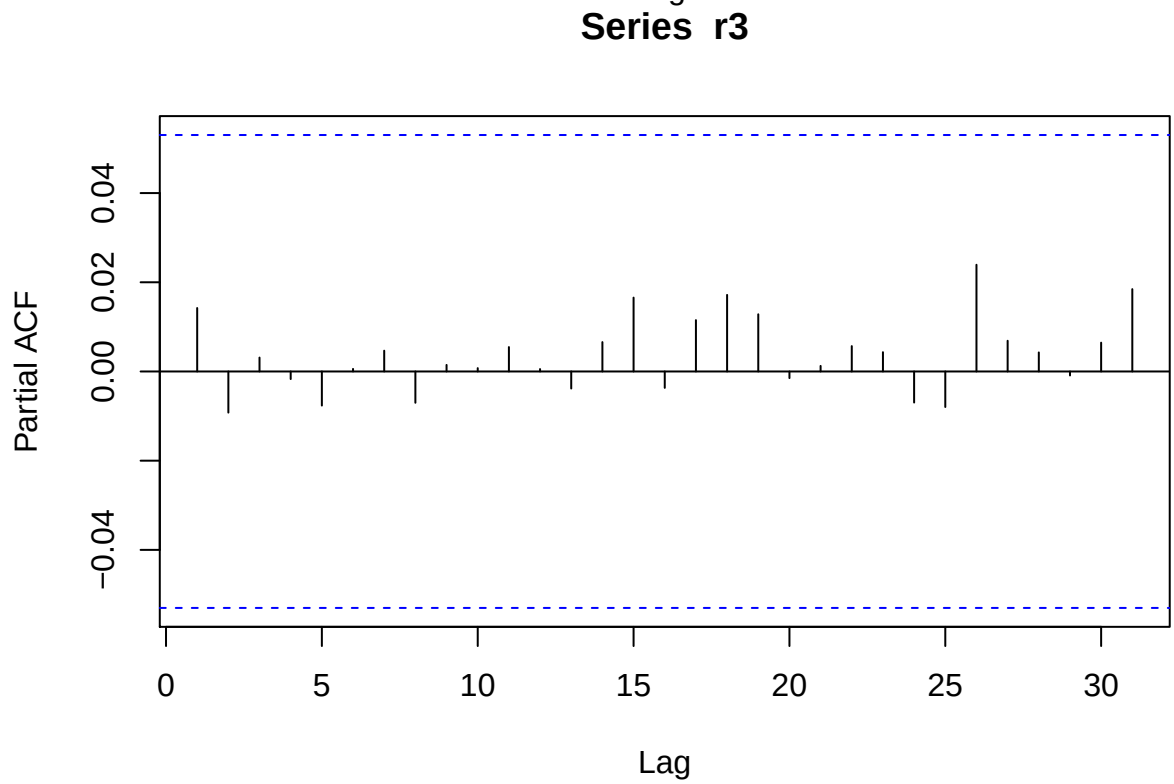




We don't need to fit an intercept because the data is centered. So OLS chose an AR(32) model.

```
## [1] 32
```





Surprising that the coefficients are all different. For example, we look at the first three:

```
##   yule.walker         ols        burg
## 1 0.260738636 0.271636613 0.258953862
## 2 0.085682152 0.072707064 0.085358004
```

```
## 3 0.055885363 0.055592143 0.056362762
## 4 0.005726957 0.008108437 0.006372378
```

Prediction

** TODO: ** Re-write. Write more. Better graphs. Look at forecast and other packages. • Predict future values: The prediction of the time series is given by the prediction of the smooth component plus the prediction of the rough component. For the smooth part, you need to make sure to be able to predict trend and seasonality. For the rough part, use one of the algorithms in Section 3.5 of the Lecture Notes.

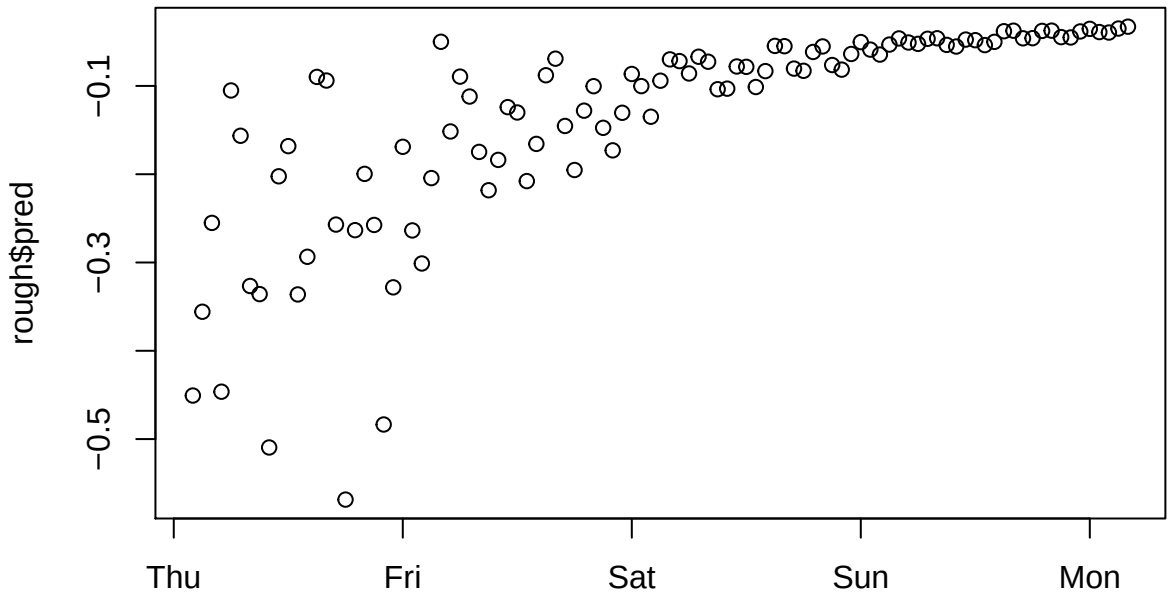
We will now do prediction. We will predict our residuals, and then put them back into the original scale by adding back on trend and seasonality and reversing the cube root transformation to determine our predictions for *count*.

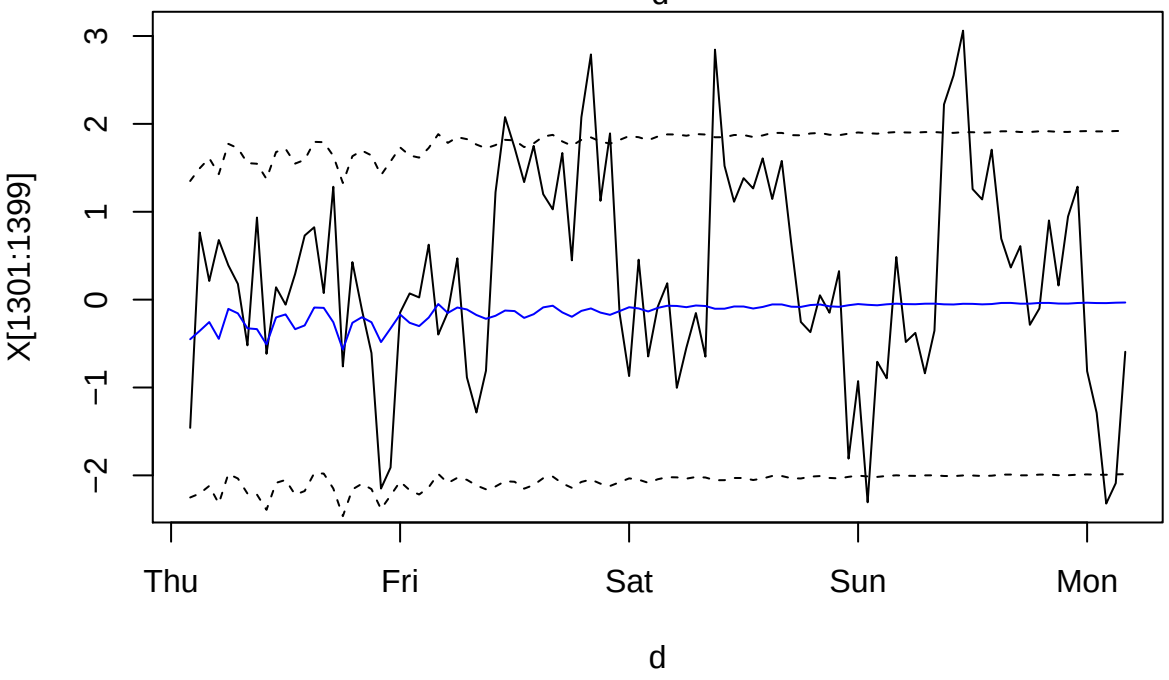
The adjusted R^2 for the ANOVA model is 0.972. We also observe that the residuals from this model do not exhibit any trend or seasonality. We conclude that the ANOVA fit adequately explains the smooth component.

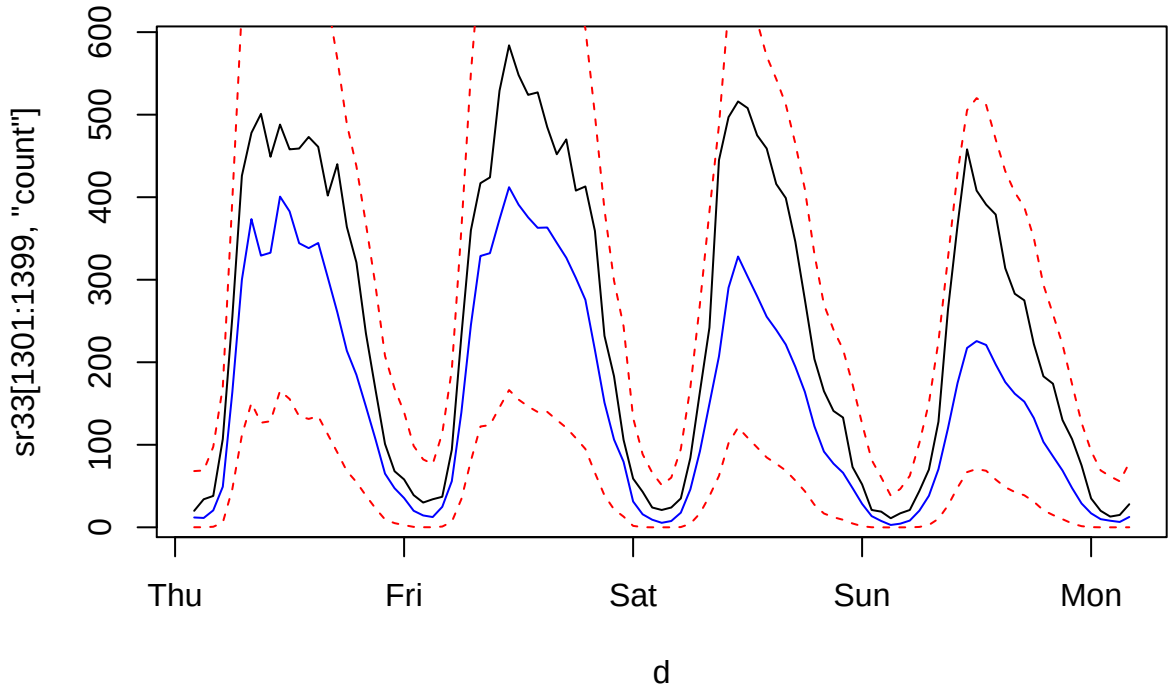
```
## [1] 0.6039812
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
hour	23	149762.555	6511.415426	3137.24106	0
weekday	6	5742.426	957.070919	461.12281	0
month	11	6173.351	561.213725	270.39632	0
poly(time, 3)	3	5657.312	1885.770736	908.57625	0
hour:weekday	138	5607.593	40.634734	19.57807	0
Residuals	54335	112773.533	2.075523	NA	NA

To evaluate the performance on the rough part we will use a fit on the first 1300 data points in the series to predict the remaining 99. We then look at the first 10 predictions compared to the actual values. Finally, we add back on trend and seasonality to see how our count prediction compared to the actual count values.

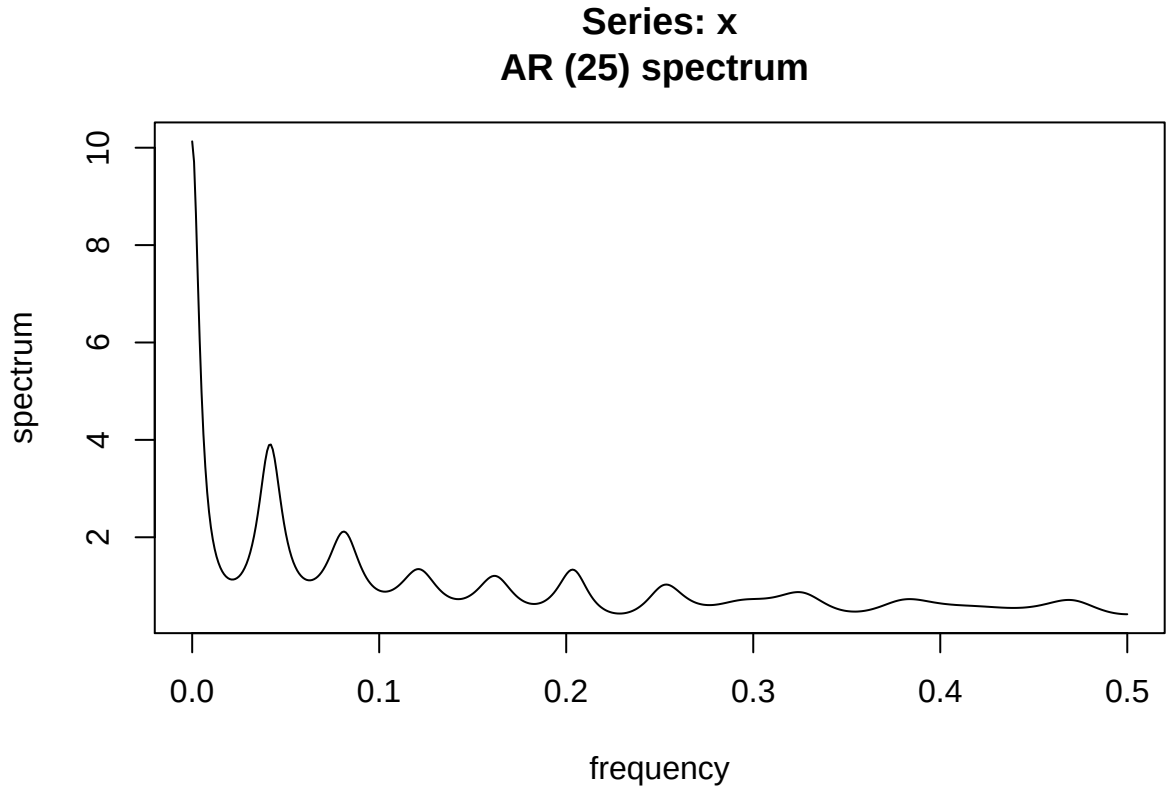

rough$pred
−0.1
−0.3
−0.5
Thu
Fri
Sat
Sun
Mon
d

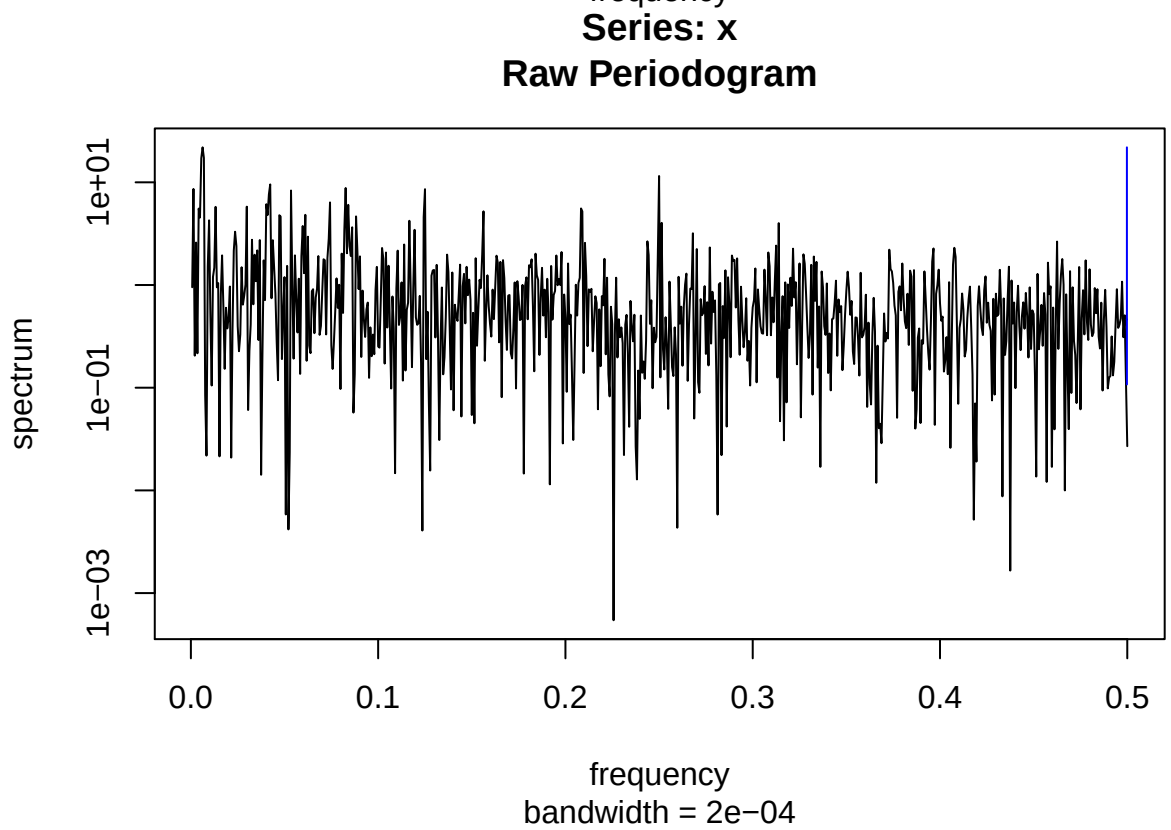

X[1301:1399]
3
2
1
0
−1
−2
Thu
Fri
Sat
Sun
Mon
d

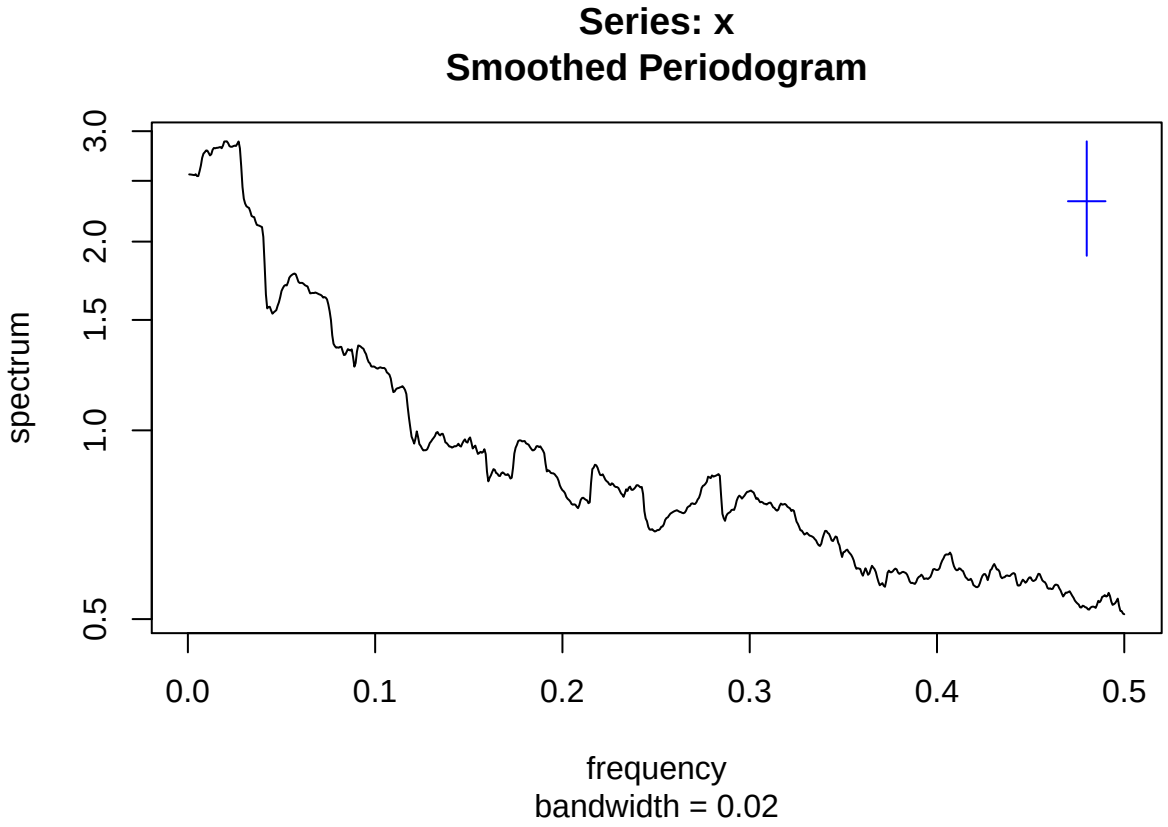


Spectral analysis

** TODO: ** Everything lol?


Series: x
AR (25) spectrum
spectrum
2
4
6
8
10
0.0
0.1
0.2
0.3
0.4
0.5
frequency


Series: x
Raw Periodogram
spectrum
1e−03
1e−01
1e+01
0.0
0.1
0.2
0.3
0.4
0.5
frequency
bandwidth = 2e−04



The increase as the frequency tends to zero is evidence of long-term memory in the traffic counts.

There looks to be a max around 0.04.

```
## [1] 25.58974
```

Computing

** TODO: ** Just go over this? Make sure it flows with the rest of the article.

The initial size of this data set, 8 million points, presented some unique computational challenges.

We used R's biglm package together with chunking to fit a linear ANOVA model with 5000 rows. It took 5 days to run on the department server (gumbel). Additionally, it used all 256 GB of memory available. It's possible that different chunk sizes would have performed better. lme4 solves the problem with sparse matrices, which is ideal, but it wouldn't fit the full model, even with the 256 GB of memory on the server.

1. Detrend
2. Deseasonalize
3. Check residuals. Only fit an ARMA model if residuals are not IID.
4. Portmanteau test
5. ACF, PACF

The prediction will be off if the model is over-fitted.

Discussion

TODO: • Discussion: Explain the findings of your data analysis. Try to be critical of what you have done and make sure you don't hush over potential shortcomings of your analysis. It is better to be aware of something that is not ideal than to try covering it up. (Importantly, this is not being viewed negatively in the evaluation of your time series analysis. It can easily happen that you face an issue you cannot deal with based on the methods you learned in STA 137. When this happens try to use your best judgement to proceed in a reasonable way.)

Conclusion

TODO: • Conclusions: Give a final verdict of your analysis.

References

TODO: • References: Name all resources you have used. Do not copy from the web and existing papers. You can and should use other resources, but they have to be clearly identified. Line-by-line copying from existing contributions will be considered plagiarism.

Appendix

TODO: • Appendix: Put all codes and additional supporting calculations here.

```
# To produce the report
library(knitr)

# Box cox transformation
library(MASS)

# Semitransparent plotting with alpha
library(scales)

# Tested helper functions
source('../functions.R')

# Contains the `fastrak` data frame with 8 million rows
load('../fastrak.Rda')

sr1 = getstation(4300, fastrak)

kable(head(sr1))

with(sr1, plot(time, count, col=alpha('black', 0.2)))
fit0 = lm(count ~ poly(time, 3), sr1)
lines(sr1$time, predict(fit0), lwd=4, col='blue')

boxcox(lm(count ~ hour + weekday + hour:weekday + month + poly(time, 3), sr1))
f = count^(1/3) ~  hour + weekday + hour:weekday + month + poly(time, 3)
```

```r
fit1 = lm(f, sr1)
res1 = residuals(fit1)
plot(sr1$time, res1, col=alpha('black', 0.1))

(cutoff = quantile(res1, 0.01))
sr2     = sr1[(res1 > cutoff) & (res1 < -cutoff), ]
dim(sr2)

fit2    = lm(f, sr2)
sr2$res = residuals(fit2)

with(sr2, plot(time, res, col=alpha('black', 0.2)))


summary(fit2)$adj.r.squared

a2 = anova(fit2)
a2$percent = 100 * a2[, 'Sum Sq'] / sum(a2[, 'Sum Sq'])
kable(a2)


diffsr = diff(sr2$time)
islong = longrun(diffsr, 1, 1000)
long   = rle(islong)
sr3    = sr2[islong, ]

# TODO - this should be it's own tested function - this code is a bit
# too hacky

# Infer the time spaced groups
a         = rle(as.numeric(diff(sr3$time)))
a$values  = 1:((length(a$values) + 1) / 2)
a$lengths = 1 + a$lengths[a$lengths != 1]
sr3$group = inverse.rle(a)

# We can add the fitted values in for further analysis
# Exponentiating since we fitted the cube root
sr3$fitted = predict(fit2, sr3) ** 3

dim(sr3)
## [1] 6228   11
with(sr3, plot(time, res, col=alpha('black', 0.2)))
# save(sr3, file='cleaned.Rda')


# Time Series Analysis
acf(X)
pacf(X)

# Parameters to search over:
#ap = expand.grid(ar=3:6, ma=3:6)
ap = expand.grid(ar=1:3, ma=1:3)
getaic = function(ar, ma, x=X){
```

```
    arima(x, order=c(ar, 0, ma), optim.control=list(maxit=1000))$aic
}
aicvals = mapply(getaic, ap$ar, ap$ma)
plot(aicvals)

ap[which.min(aicvals), ]
arma1 = arima(X, order=c(6, 0, 6), optim.control=list(maxit=1000))
arma1$aic

acf(residuals(arma1))
pacf(residuals(arma1))

ar1 = ar(X, method='yw', order.max=35)
ar1$order

r1 = na.omit(ar1$resid)
acf(r1)
pacf(r1)

ar2 = ar(X, order.max=35, method='ols', intercept=FALSE)
ar2$order
## [1] 32
r2 = na.omit(ar2$resid)
acf(r2)
pacf(r2)

ar3 = ar(X, order.max=35, method='burg', intercept=FALSE)
ar3$order
## [1] 32
r3 = na.omit(ar3$resid)
acf(r3)
pacf(r3)

first4 = data.frame('yule-walker'=ar1$ar[1:4], 'ols'=ar2$ar[1:4],
           'burg'=ar3$ar[1:4])
first4


# Prediction
s = summary(fit2)
s$adj.r.squared
## [1] 0.9720136
kable(anova(fit2))

d = sr33$time[1301:1399]
ar1300 = ar(X[1:1300], order.max=35)

rough = predict(ar1300, n.ahead=99)
plot(d, rough$pred)

plot(d, X[1301:1399], type='l')

# Our predicitons
```

```
lines(d, rough$pred, col='blue')

# Add a line for 95% confidence
alpha = 0.05
zscale = qnorm(1 - alpha/2)

roughpreds = data.frame(time = d
                        , estimate = rough$pred
                        , upper = rough$pred + zscale * rough$se
                        , lower = rough$pred - zscale * rough$se
                        )

with(roughpreds, lines(time, upper, lty=2))
with(roughpreds, lines(time, lower, lty=2))

# Add back in the smooth component.
smoothpreds = roughpreds[, -1] + predict(fit2, sr33[1301:1399, ])

# Cube it to get back to the correct scale
smoothpreds = as.data.frame(smoothpreds ** 3)

# True counts
plot(d, sr33[1301:1399, 'count'], type='l')

# Our estimates
smoothpreds$time = d
with(smoothpreds, lines(time, estimate, col='blue'))
with(smoothpreds, lines(time, upper, lty=2, col='red'))
with(smoothpreds, lines(time, lower, lty=2, col='red'))


# Spectral Analysis
Xts = ts(X)
sp = spectrum(Xts, log='no', method='ar')
spectrum(Xts)
spectrum(Xts, span=100)

bigone = which.max(sp$spec[0.02 < sp$freq & sp$freq < 0.04])
freq = sp$freq[0.02 < sp$freq & sp$freq < 0.04][bigone]
1 / freq
```