Table of Contents

Module 1: Basic Time Series Analysis

1.4 Data Example: Emergency Department Volume

1.4.1 Case Study: Emergency Department Volume

In this lesson, I will illustrate the basics of Time Series Analysis with a particular data example, specifically predicting volume of patients in emergency department care.

Emergency Department Care

Emergency Department Care

Have you ever experienced long waits in the Emergency Department?

- Good predictions of daily inflow in an emergency department can assist in staffing and diversion
- Time series modeing can be useful in achieving good predictions.

Georgia Tech

Reports from different countries including the United States, the United Kingdom, China and other countries have a shown an increase in demand for emergency department care resulting in frequently overcrowded ED's, lengthy waiting times for assistance, and an overall perception by patients of poor healthcare. Prolonged waiting times are described as a major factor for dissatisfaction with ED care, and patients are more likely to leave without being seen as waiting time increases. While common practices to using data on daily patient volume for better planning of personal resources might increase ED efficiency as well as improve ED patients' care quality.

Case Study Overview

Case Study Overview

Objective:

- Identify temporal patterns in the Emergency Department (ED) volume of patients
- Develop a model to predict ED volume

Time Series Data:

- Daily number of patients visiting an emergency department of a hospital in the Atlanta area with observations from 2010 until mid 2015
- Other predicting variables were made available by the hospital but we will only focus on the predictability of the time series with respect to temporal factors

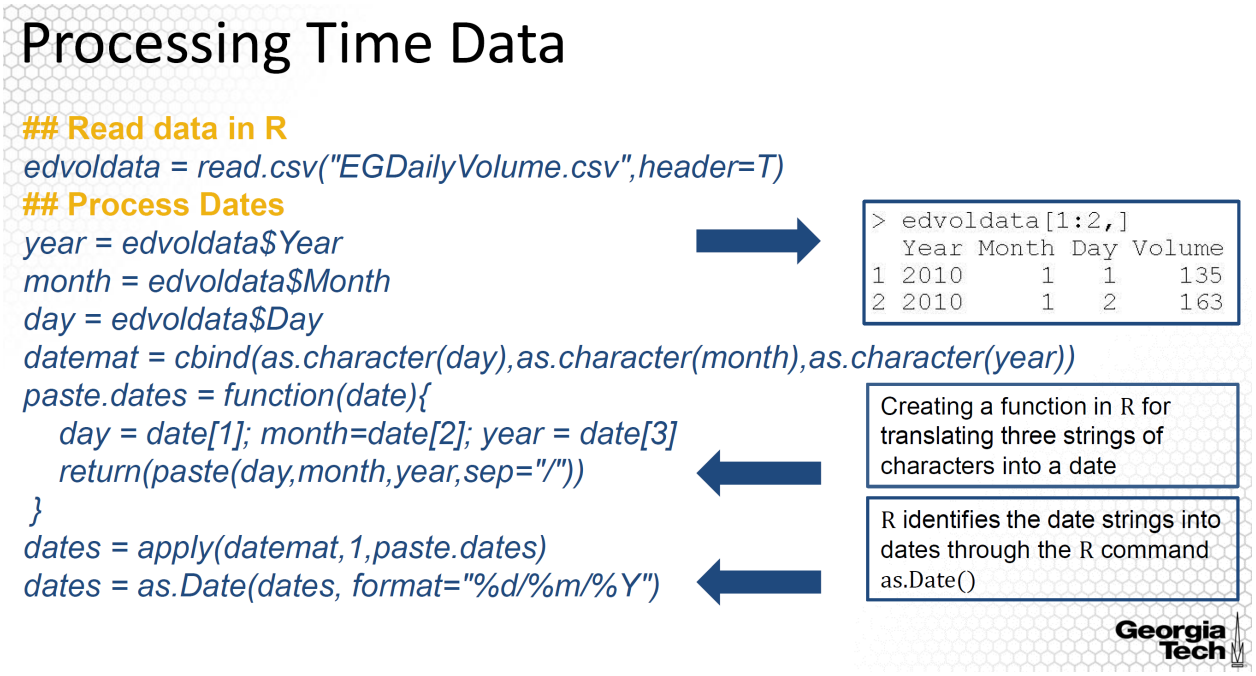
Georgia Tech

Time series models can be used to forecast future ED patient visit volume based on the estimated effect of predictor variables. And such forecasts can be useful for active bed and staff management and for facilitating patient flow.

A number of factors can influence daily ED visits and the patient visits forecasting model should include those factors. Previous studies have shown that ED visits present cyclical variations according to calendar variables such as day of the week, and time of the year. The objective of this analysis is to develop models for identifying temporal patterns and for predicting the volume of patients in an Emergency Department.

The data consists of daily number of patients seeking ED care in a hospital in Georgia in the United States. The ED volume was observed over a period of more than five years from 2010 until about mid 2015. In the study, we'll only consider temporal factors, although other factors can be used. For example, external factors such as temperature, rainfall, major holidays school season, among others, along with hospital data. For example, the percentage of patients seeking care in ED who have public insurance versus commercial insurance among others.

Processing Time Data



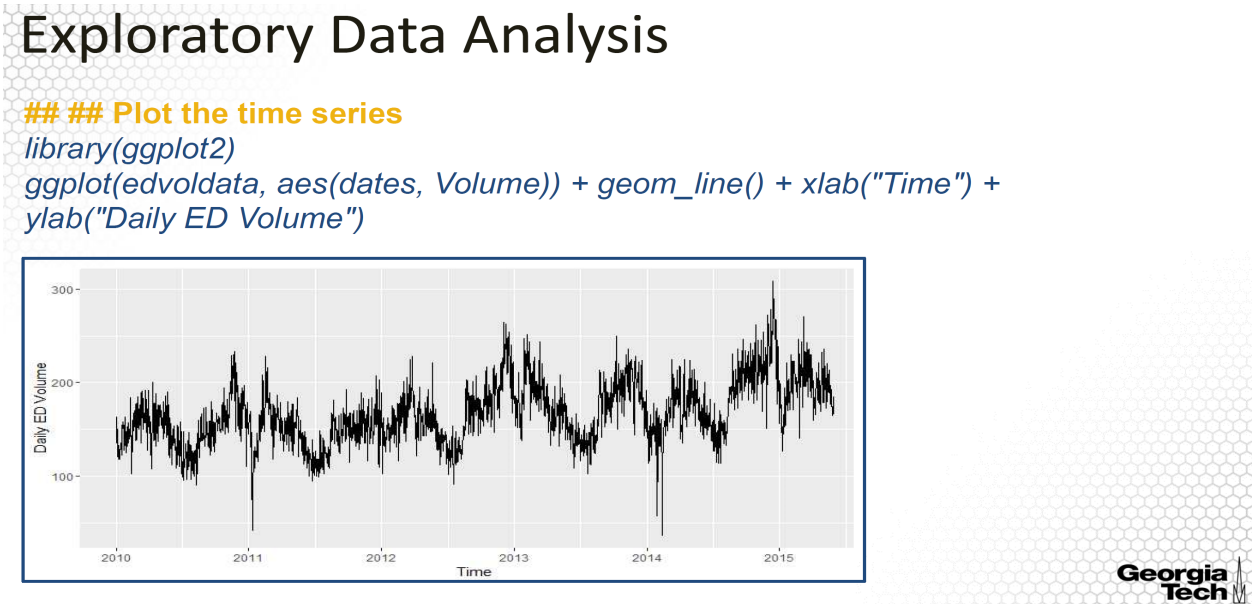
We'll begin with one important aspect in the time series analysis, processing time or dates of the observed time series data. The data columns include year, ranging from 2010 to 2015, month, taking value 1 to 12 and day of the month along with the column volume, providing the number of patients seeking care in the ED for the corresponding day. The first three columns provide the complete information on the date of the observations. But, because the information is provided

across the three columns, it needs to be converted into one column of dates to be used in visualizing the data.

To do so, we pull the three columns into one matrix. I define here a function paste.dates which takes as input a vector of three-variables: day, month and year. The function returns a date using the information from these three date factors. I apply this function to each row in the date map to output a date for each row in this matrix.

Last, I converted the dates as output from this R command into dates that R can identify as such, using the as.Date command. This vector of dates will be the input into the visual analytics displays to display time.

Exploratory Data Analysis



Our first plot of the time series is the plot of the daily ED volume. A few observations to point out are as follows. There is a generally increasing trend in the ED volume. There is also a cyclical pattern, although not necessarily a seasonality pattern, since it does not repeat at exact periods of time. There are some clear outliers, for example beginning of 2011 and in 2014.

Count Data Transformation

Count Data Transformation

ED Volume = Number of patients visiting ED per day ~ Poisson Distribution

- Poisson distribution – mean and variance are equal; if mean varies over time so does the variance
- Standard linear regression model assumes normality with constant variance
- Variance Stabilizing Transformation

```
## Apply Transformation
Volume.tr = sqrt(Volume+3/8)
## Compare Distribution
hist(Volume,nclass=20,xlab="ED Volume", main="",col="brown")
hist(Volume.tr,nclass=20,xlab= "Transformed ED Volume", main="",col="blue")
```

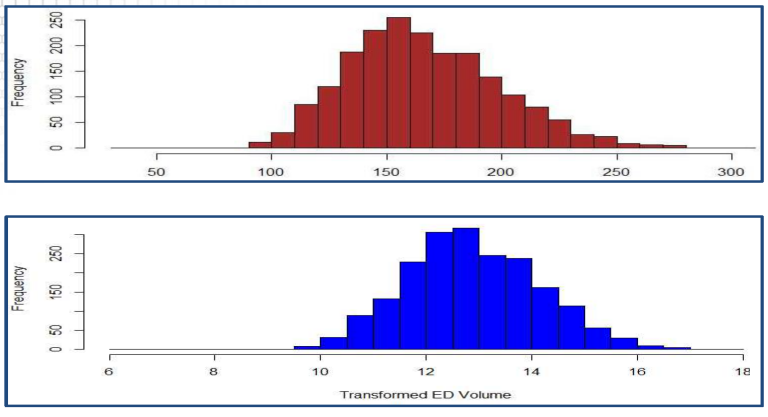
Georgia Tech

Count data within a time unit, such as the number of patients seeking care per day, has a Poisson distribution. To remind you, the mean and variance of the Poisson distribution are equal. Hence, if the mean, represented by the trend and/or seasonality, varies over time so does the variance. The models for estimating the trend and seasonality described in the previous lessons are based on the standard linear regression model, assuming that the response time series is normally distributed. One of the limitations of using the standard linear regression model under normality to model count data is that the variance of the error terms is non constant, and thus, a departure from the model assumptions. In order to use this model, we thus need to transform the data using a variance stabilizing transformation. That is, transform the count data, or Poisson distributed data, such that the data have approximately constant variance. The transformation has also the role of normalization of the count data.

The classic transformation used for count data is the square root of the counts +3/8. We apply this transformation to the count data or the ED volume and provide the time series analysis on these transformed data.

Count Data Transformation

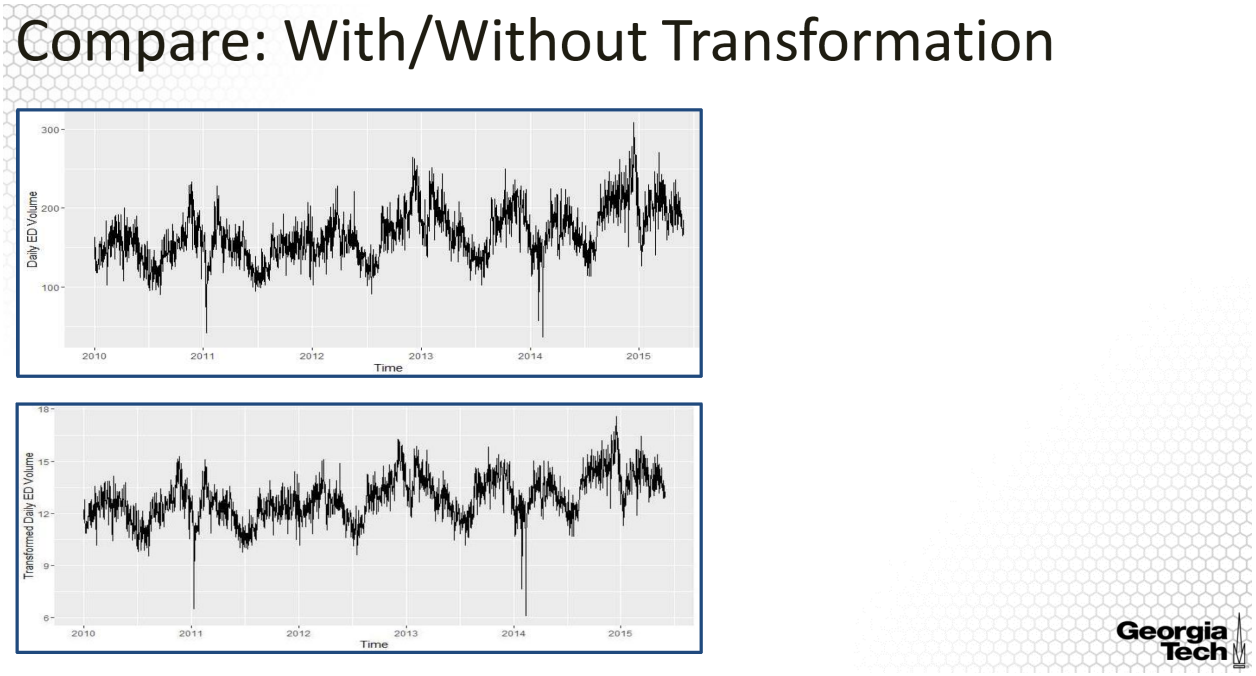
Count Data Transformation



Georgia Tech

We next compare the histograms of the ED Volume and the Transformed ED Volume to evaluate the distribution without and with transformation. These are the histograms, the first one is for untransformed and the second one is for the transformed ED Volume data. The transformation has also the role of normalization, since with this transformation the distribution is approximately symmetric.

Compare: With/Without Transformation



This slide compares the time series without the transformation in the upper plot versus with transformation in the lower plot. The pattern is generally similar with a cyclical trend with outliers at similar time points. The main difference is that there is a reduction in the variability.

Summary: This lesson introduces a case study on the time variations for the volume of patients in the Emergency Department. I will explore these data using the basic analysis introduced in the lessons in this module.

1.4.2 Trend and Seasonality Estimation

In this lesson, I will introduce illustrate the applicability of the trend and seasonality estimation methods with the data example on modelling ED volume.

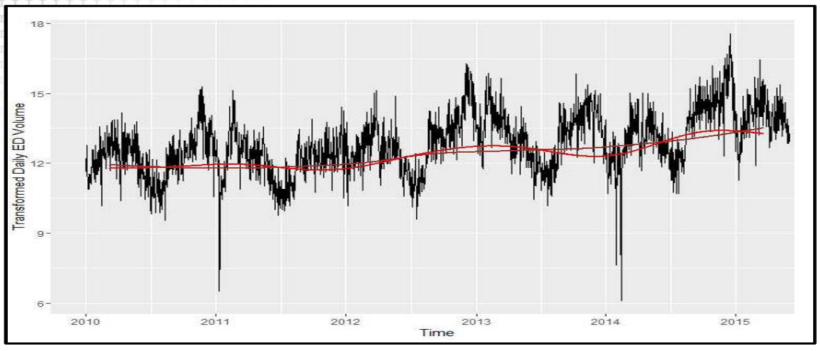
Trend Estimation

Trend Estimation

```
## Equally spaced time points
time.pts = c(1:length(Volume))
time.pts = c(time.pts - min(time.pts))/max(time.pts)
## Local Polynomial Trend Estimation
loc.fit = loess(Volume.tr~time.pts)
vol.fit.loc = fitted(loc.fit)
## Splines Trend Estimation
library(mgcv)
gam.fit = gam(Volume.tr~s(time.pts))
vol.fit.gam = fitted(gam.fit)
## Is there a trend?
plot(dates,Volume.tr,ylab="ED Volume")
lines(dates,vol.fit.loc,lwd=2,col="brown")
lines(dates,vol.fit.gam,lwd=2,col="red")
```

Let's begin with the estimation of the trend. I applied here only non-parametric estimation approaches, the local polynomial and the Splines Regression methods. Here I apply the R commands, the loess() command for local polynomial, and gam() for splines regression.

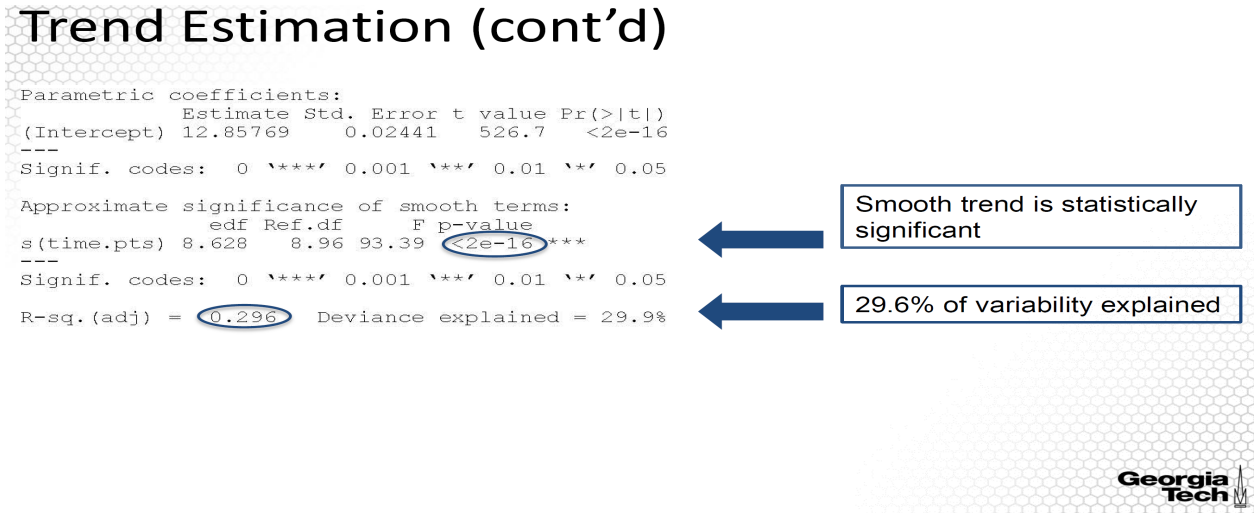
Trend Estimation



I am overlaying here the estimated trends from the two models applied to the transformed ED volume data. In brown is the local polynomial trend, and in red is the splines regression trend.

On the slide is the resulting plot. The trends are similar. They both show an increasing pattern over time, with some smooth variations over time. This splines regression trend shows more variation capturing some of the cyclical pattern.

Trend Estimation (cont'd)

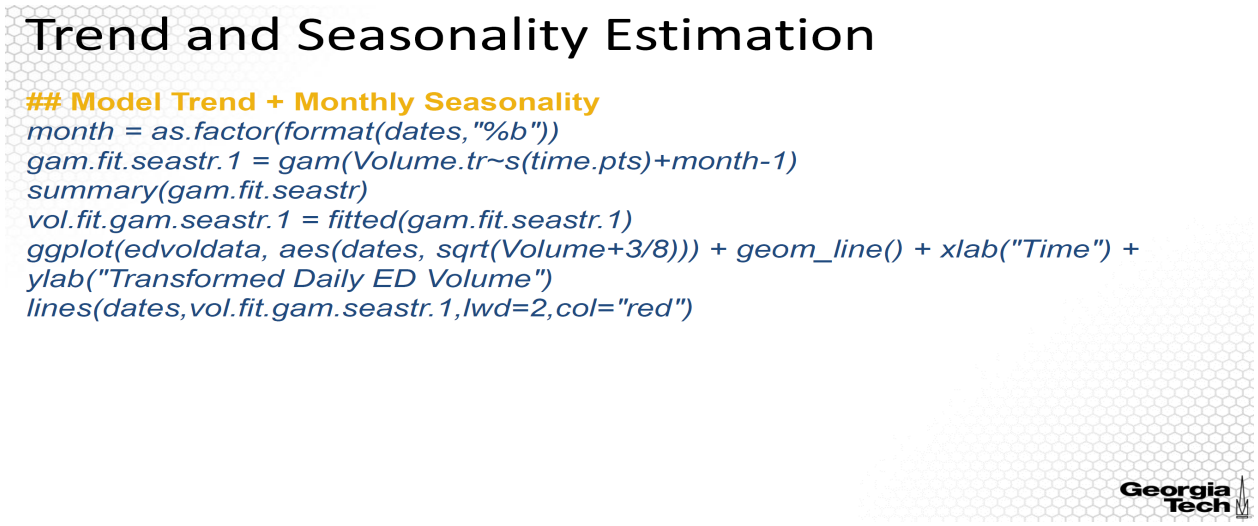


This is the output of the summary of the fit using splines regression.

First, the p value for the statistical significance of the smooth term of the trend is statistically significant indicating a statistically non-constant trend.

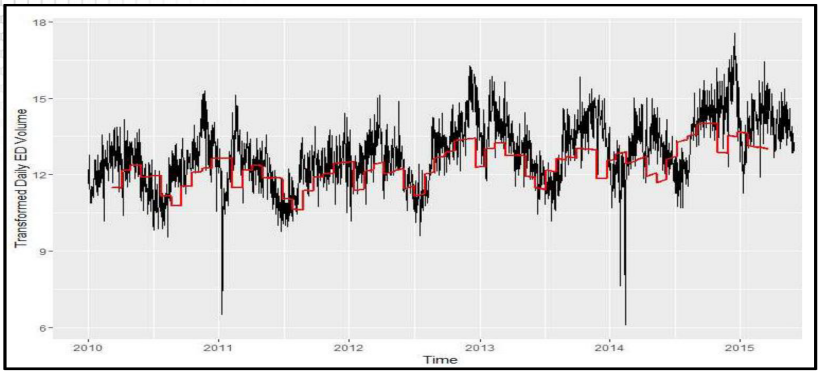
Second, the adjusted R square is 0.296, implying that about 30% of the variability in the ED Volume is explained by the trend alone.

Trend and Seasonality Estimation



Next I am applying the seasonal means model along the nonparametric trend estimation. First, I am considering the monthly seasonality.

Trend and Seasonality Estimation



On the slide is the resulting plot. We can see that the fitted values are a step function. This is because I fitted the seasonal means model. The fitted line in red does capture the trend. However, the seasonality shows differences from the observed data in some places and for some period of time, indicating that there may be more of a cyclical pattern than a monthly seasonality. The monthly cyclical pattern may be more related to other factors than the month seasonality itself. For example, school seasons or flu season may be more predictive of the cyclical pattern.

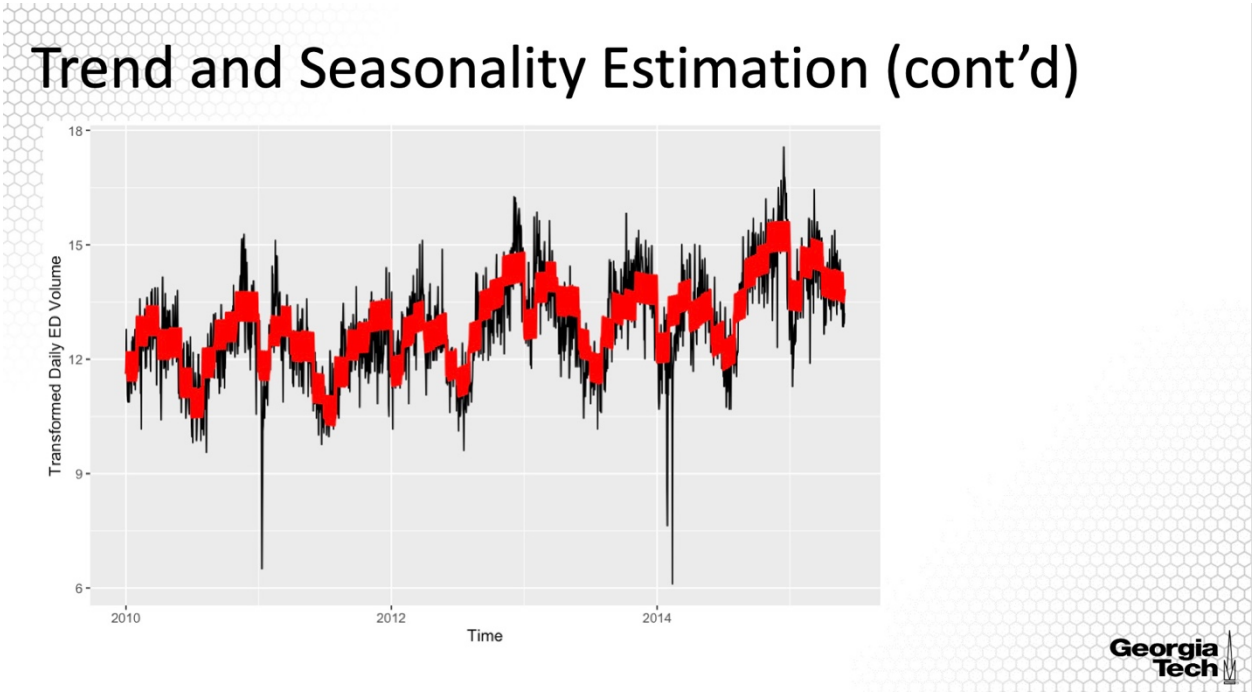
Trend and Seasonality Estimation (cont'd)

Trend and Seasonality Estimation (cont'd)

```
## Add day-of-the-week seasonality
week = as.factor(weekdays(dates))
gam.fit.seastr.2 = gam(Volume.tr~s(time.pts)+month+week)
summary(gam.fit.seastr.2)
vol.fit.gam.seastr.2 = fitted(gam.fit.seastr.2)
## Compare the two fits
ggplot(edvoldata, aes(dates, vol.fit.gam.seastr.2)) + geom_line() + xlab("Time") +
ylab("Seasonality and Trend: Daily ED Volume")
lines(dates,vol.fit.gam.seastr.1,lwd=2,col="red")
```

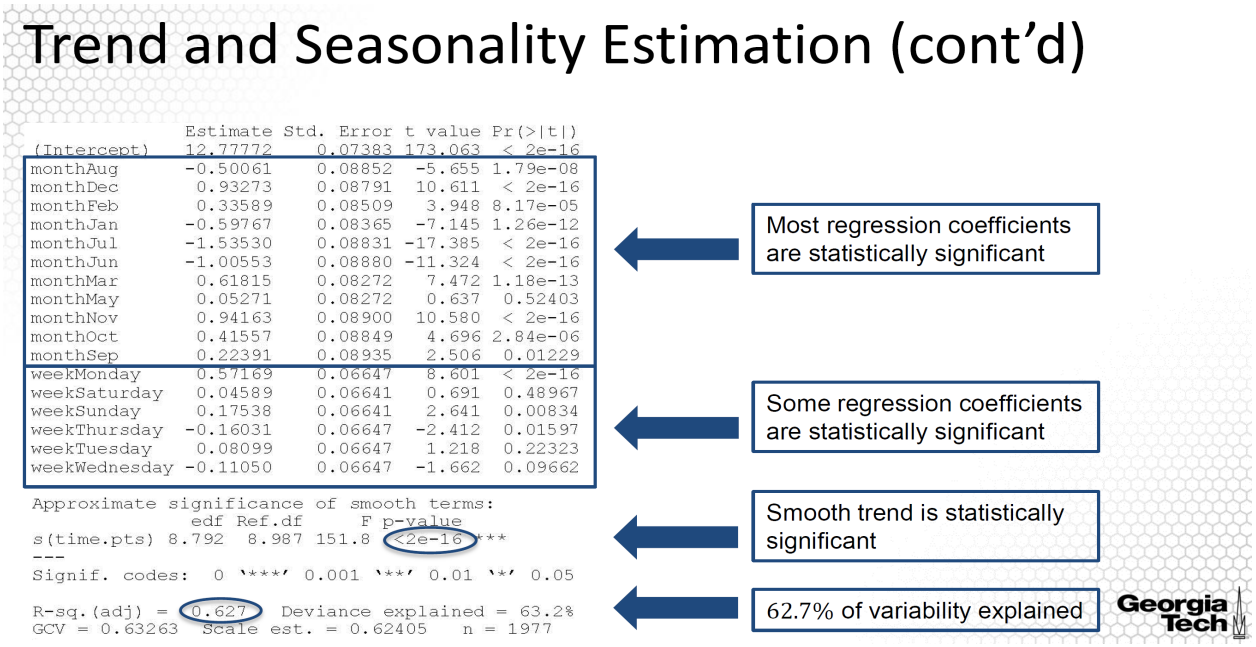
Next, I'm adding another layer of seasonality, due to the day of the week. It may be expected to see a higher volume in the ED on Sundays, since the physician offices are not open, and because people may have more activities that could lead to injuries and hence, ED visits.

For this, we need to first identify which of the dates correspond to Monday's, which correspond to Tuesday's, and so on. A simple way is to use the weekdays command in R, where the input is an object derived using as.Date command. I'm converting this into a factor to specify this is a categorical variable in the regression model. Starting with the model where I fitted the trend and monthly seasonality alone, I'm adding an additional linear term: the categorical variable corresponding to day of the week seasonality. Last, I compared the two fitted models by overlaying the fitted values from the two models where the red line corresponds to the simpler model with monthly seasonality only.



On the slide is the resulting plot. The fitted line including both monthly and day of the week seasonality, clearly shows the added seasonality to the day of the week. Altogether, estimated simultaneously, the monthly seasonality is picked up somewhat better.

Trend and Seasonality Estimation (cont'd)



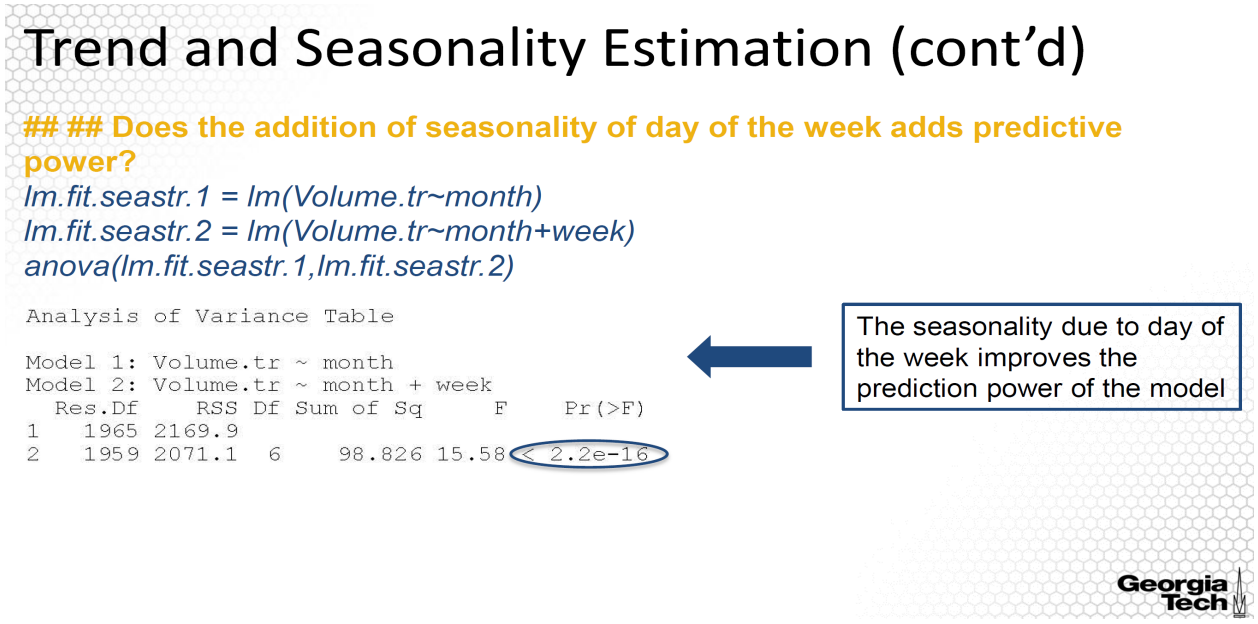
Also provided is a portion of the summary output of the model including non-parametric trend along with the two layers of seasonality.

The first portion of the output shows the estimated coefficients for the monthly seasonality. In the first column we have the estimated coefficients, in the second column we have the standard errors, and in the last column we have the p-values for statistical significance. From this output we find that most of the regression coefficients corresponding to the monthly effect are statistically significant given the day of the week's seasonality and the non-parametric trend in the model.

The second portion of the model output shows estimated coefficients along with the p values for statistical inference for the day of the week seasonality. From this output we find that some of the regression coefficients corresponding to the day of the week effect are statistically significant given that monthly seasonality and the non-parametric trend are in the model. The output for the trend also shows statistical significance of the smooth component given the seasonality in the model.

Last, the R squared is 0.627, or 62.7% of the variability in the time series is explained by the trend and the two layers of seasonality. This is to be compared to 0.296, the r square for the model where only the trend was fitted.

Trend and Seasonality Estimation (cont'd)



From the previous output, we found that for both layers of seasonality, there is statistical significance for some of the regression coefficients. The question that we will address next is: Does the day of the week seasonality add predictive power to the model? For this, we will compare the models without and with this layer of seasonality. I thus fitted a linear regression model with both layers of seasonality, and the model with only monthly seasonality, and compare them using the partial F-test through the command anova() in R. The input in this command consists of the two models.

The output of the anova command is on the slide. The p-value of the test is very small, indicating that we reject the null hypothesis that the simpler model with monthly seasonality only is as good as the one with both layers of seasonality, suggesting that both the monthly seasonality and the day-of-the-week seasonality explain the variability in the ED volume.

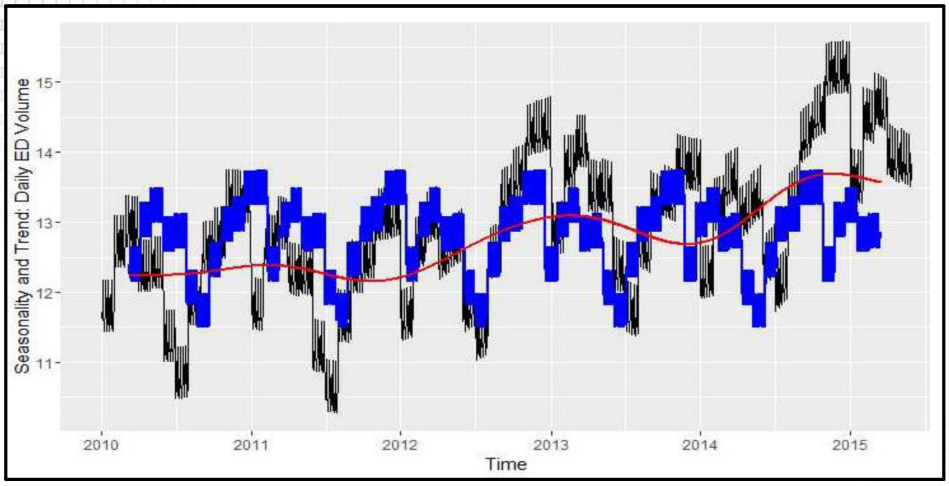
Seasonality vs Trend

Seasonality vs Trend

```
## Compare with & without trend
ggplot(edvoldata, aes(dates, vol.fit.gam.seastr.2)) + geom_line() + xlab("Time") +
ylab("Seasonality and Trend: Daily ED Volume")
lines(dates,vol.fit.lm.seastr.2,lwd=2,col="blue")
lines(dates,vol.fit.gam,lwd=2,col="red")
```

I'm also comparing the fitted regressions without and with trend to assess whether the trend adds any additional predictive power.

Seasonality vs Trend



The plot is on the slide. In black, it is the fitted regression for the full model, the one in blue is for seasonality with the two layers of seasonality excluding the trend, and the one in red is the fitted trend alone. The trend does affect the fit significantly, thus the full model provides the best representation of the nonstationary components of the time series represented by ED volume.

1.4.3 Stationarity of ED Volume Time Series

In this lesson, I will continue with the data example for modeling the ED volume of patients by analyzing the stationarity of the residual process after removing the trend and seasonality.

Residual Process

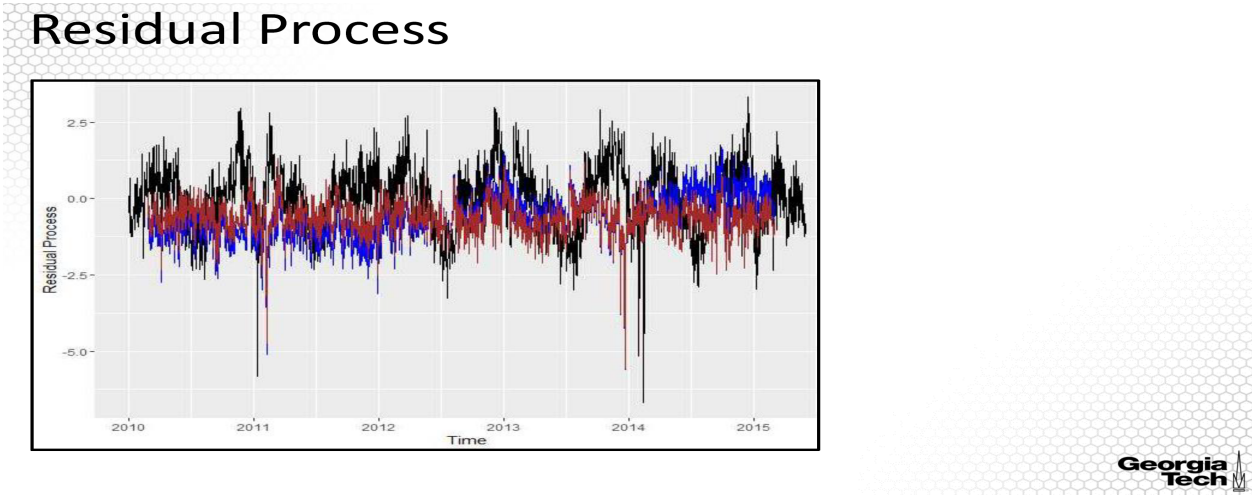
Residual Process

```
## Residual Process: Trend Removal
resid.1 = Volume.tr-vol.fit.gam
## Residual Process: Seasonality Removal
resid.2 = Volume.tr-vol.fit.lm.seastr.2
## Residual Process: Trend and Seasonality Removal
resid.3 = Volume.tr-vol.fit.gam.seastr.2
y.min = min(c(resid.1,resid.2,resid.3))
y.max = max(c(resid.1,resid.2,resid.3))

ggplot(edvoldata, aes(dates, resid.1),ymin=y.min,ymax=y.max) + geom_line() +
xlab("Time") + ylab("Residual Process")
lines(dates,resid.2,col="blue")
lines(dates,resid.3,col="brown")
```

Georgia Tech

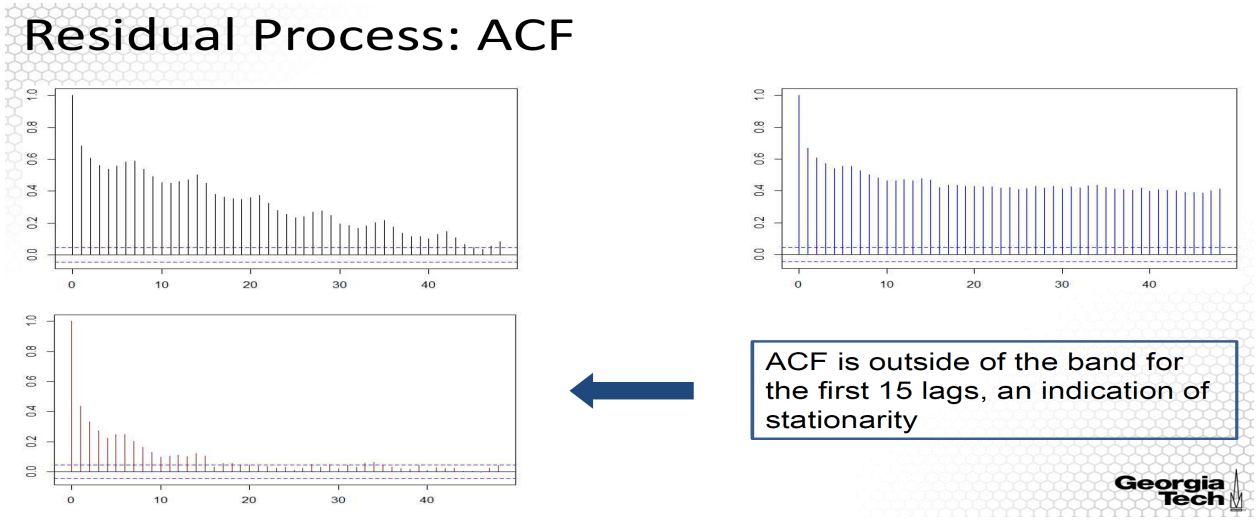
Here we'll compare three residual processes. First, it is the residual process after removing the trend alone. Second is a time series after removing seasonality, including monthly and day of the week. Third, it is the residual time series removing both trend and seasonality. The first R commands are to obtain these three residual time series. Next, I define the minimum and maximum values across all three processes in order to plot those time series on the same scale. The next set of R commands plots all three residual time series overlaying them into one plot.



The resulting plot is on the slide. In black is the time series after removing the trend alone, in blue is the time series after removing the seasonality. The time series in black clearly have some non-stationarity patterns while all three have outliers.

Residual Process: ACF

To better evaluate the stationarity of a time series, we can instead evaluate the auto-correlation plot of the time series. Here I'm providing the R commands for the sample autocorrelation of all three time series.

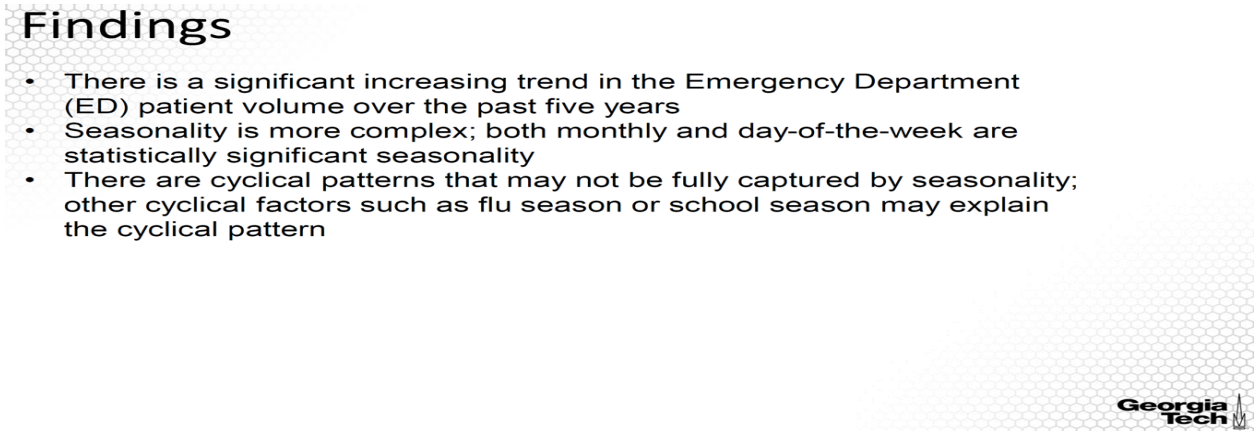


The first plot is for the residual time series after removing only the trend. An indication of non-stationarity is that the sample autocorrelation is large, or outside of significance bands for many large lags. This is the case here. We see that even for a lag of 45 the autocorrelation barely drops within the band. Moreover, we can see a pattern of ups and downs, an indication of seasonality.

The second plot is for the residual time series after removing the seasonality. While we do not see the cyclical, or the seasonal pattern in the ACF values, the sample autocorrelation decreases slowly, an indication of the presence of a trend in this process.

Last ACF plot is for the residual time series after removing the seasonality and the trend. The sample ACF values clearly decrease faster than for the previous residual time series with small values within the significance band, starting around the lag 15. This is an indication of possibly a stationary process. We'll expand more on stationarity based on the ACF plot and more rigorous ways to evaluate in the next module of this course.

Findings



I will conclude with some of the findings from this time series analysis. There is a significant increasing trend in the Emergency Department patient volume over the past five years. Seasonality is more complex, both monthly and day of the week are statistically significant. There are cyclical patterns that may not be fully captured by seasonality. Other cyclical factors, such as flu season, or school season may explain the cyclical pattern.

Summary: This concludes the exploratory analysis of the Emergency Department volume time series.

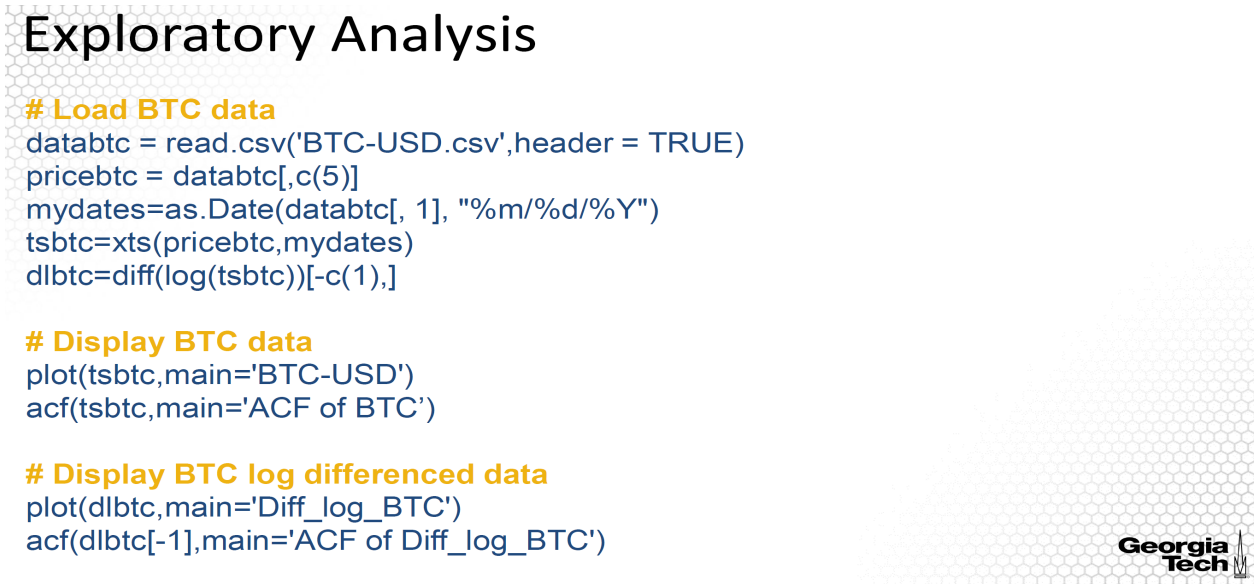
1.4.4 Bitcoin Price Exploratory Analysis

In the last two lessons of this Module. I will present a second data example. The data example is an exploratory analysis of the Bitcoin price. I will return to this data analysis in later Modules of this course.

Bitcoin Price Analysis

Bitcoin is the first decentralized cryptocurrency in the world and possesses characteristics different from traditional financial assets. Its price surge then when down in recent years, triggering enormous interest among the investing public. As we will discuss later in this course, the price of bitcoin has very wide volatility with large swings in the price ups and downs. While bitcoin has been invented in 2008, I will only analyze its price starting January 2017 because of its very interesting behavior within the recent years.

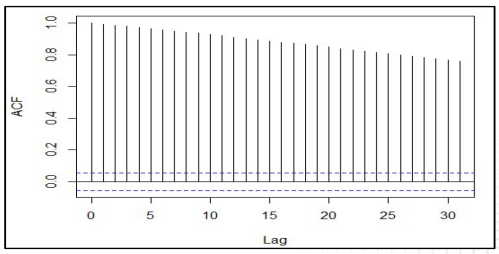
Exploratory Analysis



We will begin here with an exploratory analysis of the bitcoin price. On this slide, I am providing the code for reading the data and for transforming the dates then append both the price and the dates into a time series. Next, I am plotting the time series along with the sample auto correlation plot. Then plot the differenced log-transformed time series and the corresponding sample auto correlation plots.

Exploratory Analysis: Time Series

Exploratory Analysis: Time Series



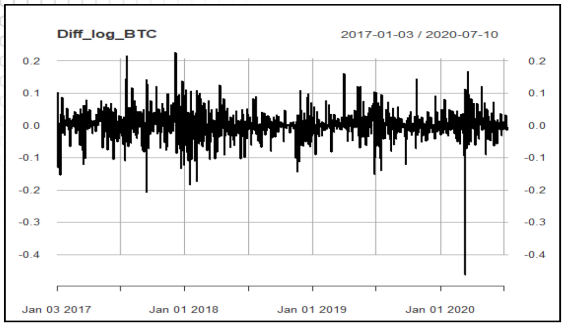
The time series is nonstationary with a non-linear trend

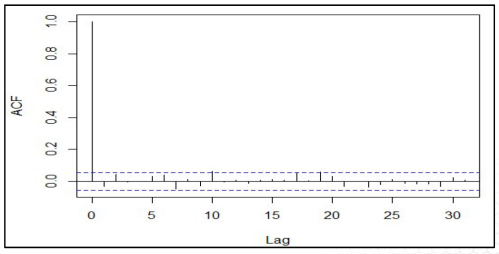
Georgia Tech

On this slide are the plots for the time series. The time series is clearly non-stationarity with a nonlinear trend. This is also clear from the acf plot; the autocorrelation decreases slowly with an increase of the lag.

Exploratory Analysis: Log-Differenced Data

Exploratory Analysis: Log-Differenced Data



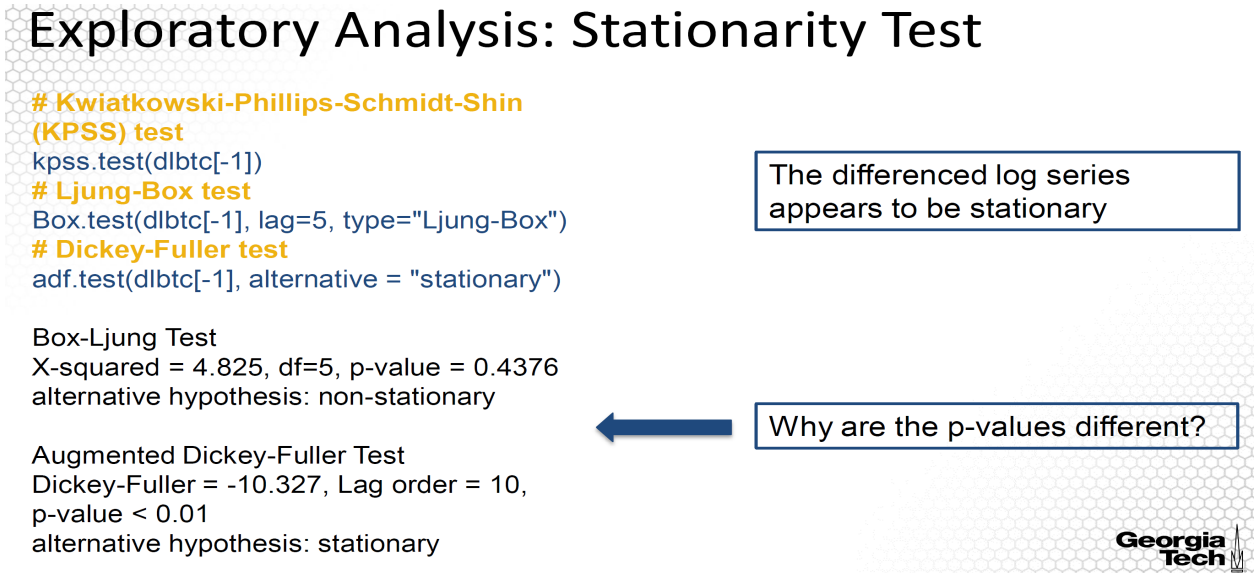


The time series appears to be (weakly) stationary by looking at the acf plot, although with non-constant variability over time.

Georgia Tech

On this slide are the plots for the differenced log time series. For this time series, we can see that the difference has removed the trend. The variability of the time series is however changing over time. If we look at the acf plot, the auto correlation is small for all lags larger than 0 an indication of week stationarity. We will see later in this course that this is common for the price of financial instruments such as stocks. Please note that we can have a weakly stationary process with volatility being different over time. In fact, we will learn in Module 3 how to model such time series.

Exploratory Analysis: Stationarity Test



I will explore here stationarity using hypothesis testing. In this test, the null hypothesis is that the time series is stationary while the alternative is that it is not stationary. We will learn the details of such tests in the next module. Here I simply would like to illustrate this test using various commands in R. First, the Ljung-Box test examines whether there is significant evidence for non-zero correlations at given lags, with the null hypothesis of lack of autocorrelation in a given time series (a non-stationary signal will have a low *p*-value). This is test is most commonly used. Another test is the Augmented Dickey–Fuller (ADF) test to find if the series has a unit root, for example, a series with a trend line will have a unit root and result in a large *p*-value. Lastly, we can test if the time series is level or trend stationary using the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test. Here we will test the null hypothesis of trend stationarity; a low *p*-value will indicate a signal that is not trend stationary, has a unit root. Again, we will come back to the concept of unit root and these tests in Module 2.

What we learn from this R implementation is that the p-value is small for the ADF test but large for the Box-Ljung test. But be careful about the differences between the two tests. For the Box-Ljung test, we test the null hypothesis of stationarity while for the AF test, the alternative hypothesis is stationarity. This means that we seek large p-values for the Box-Ljung test, that is do not reject the null hypothesis, while we seek small p-values for ADF test since we want to reject the null hypothesis. This example illustrates the need for knowing what a hypothesis test is about, specifically, what the null and the alternative hypotheses are stating. Please read carefully all the help menu for all testing procedure you will be employing in all statistical modeling and inference.

To conclude, as we visually hinted to on the previous slide, the time series is plausibly stationary as provided by these tests.

Summary: In this lesson, I provided an exploratory analysis of the Bitcon price. I will continue with a trend analysis in the next lesson.

1.4.5 Bitcoin Price: Trend Analysis

The data example in this lesson is an exploratory analysis of the trend of the Bitcoin price.

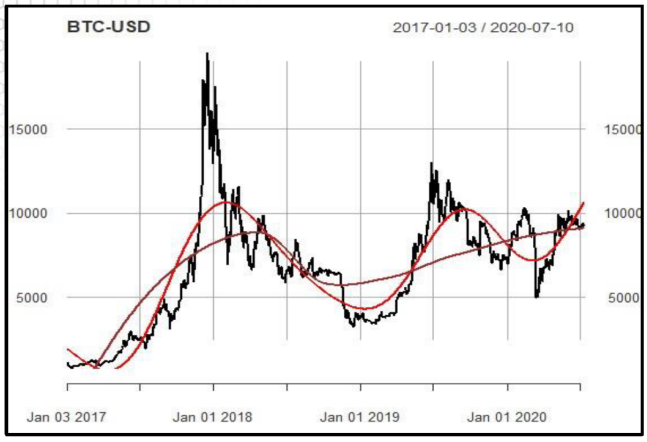
Trend Estimation: Bitcoin Price

Trend Estimation: Bitcoin Price

```
# Trend Estimation for original time series
library(mgcv)
time.pts <- c(1:length(mydates))
time.pts <- c(time.pts - min(time.pts))/max(time.pts)
# Local Polynomial Trend Estimation
loc.fit <- loess(pricebtc~time.pts)
loc.tsbtc <- xts(fitted(loc.fit),mydates)
# Splines Trend Estimation
gam.fit <- gam(pricebtc~s(time.pts))
fit.tsbtc <- xts(fitted(gam.fit),mydates)
# Display BTC data & fitted trend
plot(tsbtc,main='BTC-USD')
lines(fit.tsbtc,lwd=2,col="red")
lines(loc.tsbtc,lwd=2,col="brown")
```

We are applying here the same analysis for estimating the trend of the bitcoin price using nonparametric approach, specifically for fitting both the local polynomial and the splines regression.

Trend Estimation: Bitcoin Price



The splines regression captures the non-linear trend better than polynomial smoothing.

The plot of the observed price along with the fitted lines is on the slide. The local polynomial is in brown and the splines regression fit is in red. The splines regression captures the non-linear trend better than polynomial smoothing. However, both are smooth fits of the trend, not capturing the more settle variations in the price.

Trend Estimation: Log Differenced Bitcoin Price

Volatility Estimation: Log Difference Data

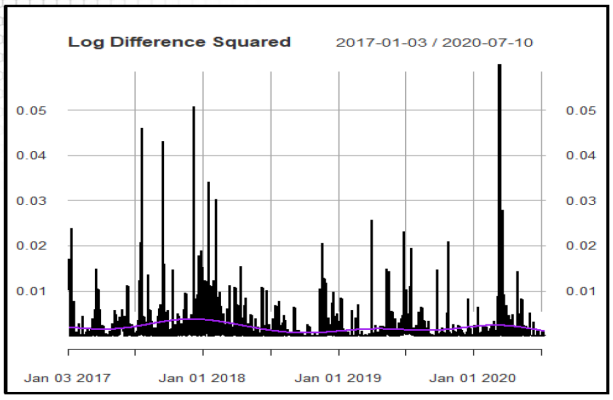
```
# Volatility Estimation for the log-difference time series
diff.ts.sq <- diff.ts^2
gam.fit.dif.sq <- gam(diff.ts.sq~s(time.pts[-1]))
summary(gam.fit.dif.sq)
difprice.fit.gam <- fitted(gam.fit.dif)
fit.tsbtc.dif <- xts(difprice.fit.gam,mydates[-1])

# Display BTC log diff data squared & estimated volatility
plot(dlbtc^2,main='BTC-USD - Log Difference Squared',ylim=c(0,0.06))
lines(fit.tsbtc.dif,lwd=2,col="purple")
```

Georgia Tech

If you recall, I mentioned that the differenced log time series for bitcoin shows different variability over time. That is, while the time series itself is weakly stationary, we do see a varying trend in the squared time series. Here we will estimate the volatility of the difference log time series using the same trend estimation analysis, but this time not applied to the time series itself but the squared time series.

Trend Estimation: Log Difference Data



There are periods of high volatility, for example, around the time when the Covid-19 crisis hit.

Georgia Tech

The plot the squared differenced time series along with the estimated volatility in purple is on the slide. Please note that I restricted the y axis to be smaller than 0.06 since during the covid19

crisis, we have seen very large swings in the log price. From this plot, we see that there have been several periods of time of high volatility including the covid19 crisis.

Is there seasonality in bitcoin price?

Is there seasonality in bitcoin price?

```
# Trend & Seasonality Estimation for original time series
month = as.factor(format(mydates,"%b"))
gam.fit.seastr.1 = gam(pricebtc~s(time.pts)+month)
summary(gam.fit.seastr.1)
fitseastr.tsbtc=xts(fitted(gam.fit.seastr.1),mydates)

# Display BTC data & fitted trend
plot(tsbtc,main='BTC-USD')
lines(fit.tsbtc,lwd=2,col="red")
lines(fitseastr.tsbtc,lwd=2,col="blue")
```

Georgia Tech

We have seen that the fitted trend using the nonparametric approach has not captured some of the smaller variations. Could then a seasonality fit capture that? Here we are adding a seasonal component to the model using the seasonal means model. I then am comparing the fitted trend and the fitted seasonality and trend together.

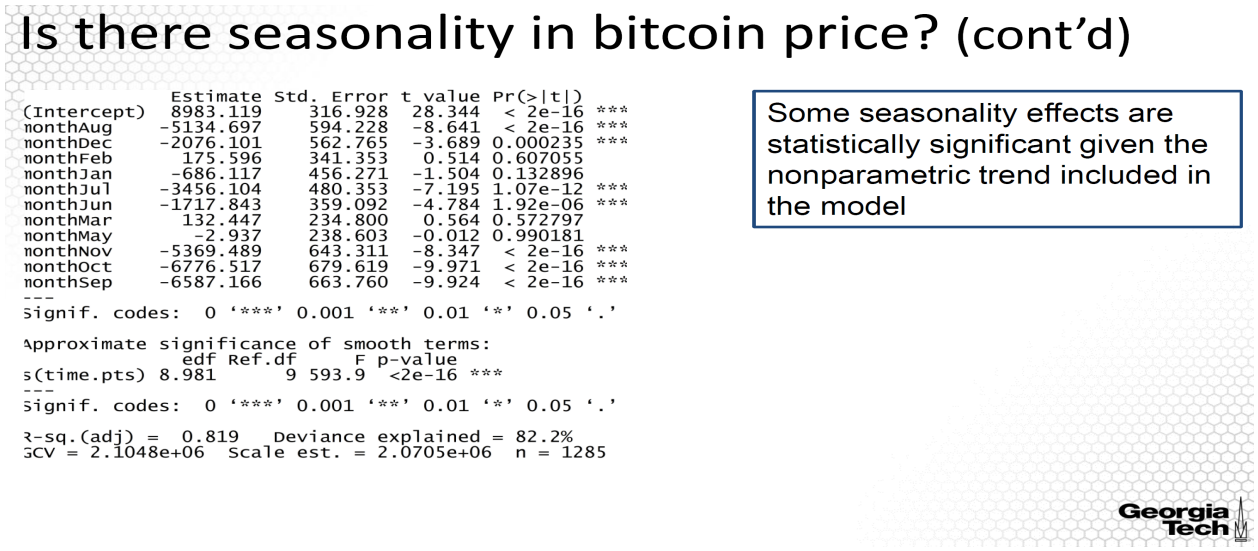
Seasonality-Trend Fit

The seasonality-trend fit does not improve the fit

Georgia Tech

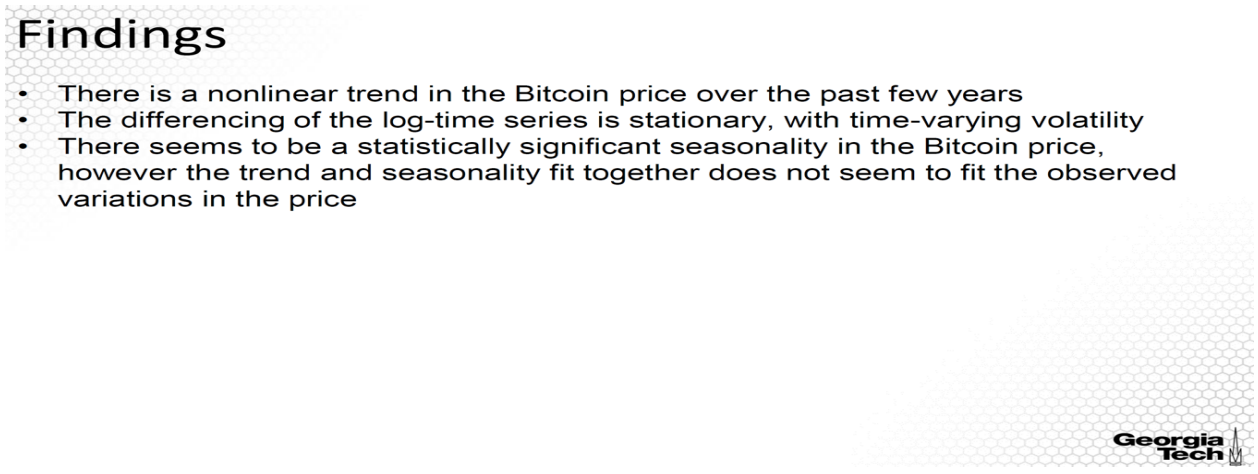
The plot is on the slide. The seasonality-trend fit does not improve the fit as shown by the blue fit. While the blue fit adds the seasonal variations, those don't seem to follow the variations in the time series.

Is there seasonality in bitcoin price? (cont'd)



However, if we look at the statistical significance of the regression coefficients corresponding to the seasonal effects, we see that most of them are statistically significant given the nonparametric trend included in the model. It is possible that the time series to present some cyclical patterns.

Findings



The main findings from this study are as follows. There is a nonlinear trend in the Bitcoin price over the past few years. The differencing of the log-time series is weakly stationary although with time-varying volatility. We will come back to this concept in Module 3 of this course. There

seems to be a statistically significant seasonality in the Bitcoin price, however the trend and seasonality fit together does not seem to fit the observed variations in the price.

Summary: I will conclude here the exploratory analysis of the Bitcoin price and the introduction of basic time series analysis.