

# Time Series Midterm Review

*David Josephs*

*2019-06-28*



# Contents

<b>1</b>	<b>Time series o clock</b>	<b>5</b>
<b>2</b>	<b>Expected Values, Variance, etc.</b>	<b>7</b>
2.1	Notation and Some Definitions . . . . .	7
2.2	Population Means and Variances . . . . .	7
2.3	Estimating Mean and Variance for Multiple Realizations . . . . .	7
2.4	Expected Value . . . . .	8
2.5	Variance and Covariance . . . . .	9
<b>3</b>	<b>A Brief Discussion of Stationarity</b>	<b>11</b>
3.1	Condition One: Constant Mean . . . . .	11
3.2	Condition Two: Constant Variance . . . . .	13
3.3	Condition 3: Constant autocorrelation . . . . .	13
<b>4</b>	<b>Autocorrelation</b>	<b>15</b>
4.1	Independence . . . . .	15
4.2	Serial Dependence / Autocorrelation . . . . .	15
<b>5</b>	<b>Autocorrelation Concepts and Notation</b>	<b>21</b>
5.1	Theoretical concept of $\rho$ . . . . .	21
5.2	autocovariance . . . . .	21
5.3	autocorrelation . . . . .	21
5.4	Stationary Covariance . . . . .	21
5.5	Stationary Time Series: Putting it Together . . . . .	26
<b>6</b>	<b>Practical Autocorrelation</b>	<b>27</b>
6.1	Estimation! . . . . .	27
<b>7</b>	<b>The Frequency Domain</b>	<b>29</b>
7.1	Review of sine and cosine . . . . .	29
7.2	Frequency and Period . . . . .	30
7.3	psueudo-periodicity . . . . .	31
7.4	Aperiodicity . . . . .	31
7.5	Frequency . . . . .	31
7.6	Frequency Terminology . . . . .	31
7.7	Composite sine function . . . . .	31
7.8	Fundamental idea . . . . .	33
<b>8</b>	<b>Spectral Density</b>	<b>35</b>
8.1	Derivation of spectral density of white noise . . . . .	35
8.2	Estimating Spectral Density from Real Data . . . . .	35
8.3	Spectral Density in the Wild . . . . .	36
8.4	Why do we do spectral density from -0.5 to 0.5? . . . . .	39

<b>9</b>	<b>AR(1) Models and Filtering</b>	<b>41</b>
9.1	Algebra review . . . . .	41
9.2	Linear filters . . . . .	42
9.3	Types of filters: . . . . .	42
9.4	GLPs . . . . .	49
9.5	AR(1) Intro . . . . .	49
<b>10</b>	<b>Characteristic Equations for AR(1) Models</b>	<b>53</b>
<b>11</b>	<b>AR(2) models</b>	<b>55</b>
11.1	Notes . . . . .	55
11.2	Stationarity in AR(2) . . . . .	55
11.3	AR(2) zero mean form . . . . .	55
11.4	More on backshift operator notation . . . . .	55
11.5	Characteristic equation . . . . .	56
11.6	Key result . . . . .	56
<b>12</b>	<b>AR(p)</b>	<b>59</b>
<b>13</b>	<b>MA(q) Models</b>	<b>61</b>
13.1	Properties and Characteristics . . . . .	61
13.2	MA(q) . . . . .	61

## Chapter 1

### Time series o clock



## Chapter 2

# Expected Values, Variance, etc.

### 2.1 Notation and Some Definitions

In time series, we denote the response variable as  $X_t$ , and clearly the explanatory variable as  $t$ . To denote a specific point in time, let us say  $t = 9$ , we would say  $X_9$ .

The collection of all  $X_n$  is a stochastic process known as a time series.

#### 2.1.1 Realizations and Ensembles

A **realization** is a particular instance of a time series. It is one of the infinitely many possible time series that could have been observed. As a time series is a stochastic process, each one could be entirely different. The **realization** is the one that was *actually* observed.

Sometimes a realization is the only realization possible. We can theorize what an infinite population of those realizations, but it is often the case that there is only one.

An **ensemble** is the theoretical totality of all possible realizations. It is the *ensemble of realizations*. We can think of that as the *population* of realizations. When we are thinking of means and variances, we think of them in terms of the ensemble, even though we may only be able to look at a single realization.

### 2.2 Population Means and Variances

It is important to note that when dealing with an ensemble of realizations, each time point is allowed to have its own mean and variance.

#### 2.2.1 The Mean of $X_t$

$\mu_t$  is the mean of all possible realizations of  $X_t$  *for a fixed  $t$* .

#### 2.2.2 Variance of $X_t$

$\sigma_t^2$  is the variance of all possible realizations of  $X_t$  *for a fixed  $t$*

### 2.3 Estimating Mean and Variance for Multiple Realizations

Let us say we have  $n$  realizations of  $X_t$  at time  $t$ . The average can be then written as

$$\hat{\mu}_t = \frac{1}{n} \sum_{i=1}^n X_{t_n}$$

Variance is calculated the same way. We are making sample means and sample variances. If we assume the population is normally distributed, this allows us to take confidence intervals on the mean of our set of realizations.

However, We have a problem: What to do when we only have one realization?

## 2.4 Expected Value

In short, the expected value ( $E[X]$ ) of a random variable (RV) denoted by  $X$  is the **mean** or more intuitively the **long run average** of the event that variable represents.

### 2.4.1 Discrete RVs

A discrete RV is an RV in which  $X$  cannot take on any value, it has specific values it can exist at and that is it.

The formula for EV of a discrete RV is as follows:

$$E[X] = \sum X P(X) = \mu$$

### 2.4.2 Continuous RVs

A continuous RV has instead of discrete values a **probability distribution/density function**,  $f(x)$ .

The formula for EV of a continuous RV is as follows:

$$E[X] = \int_a^b x f(x) dx = \mu$$

Note that is directly analogous to the discrete RV, and that a and b can span to  $\pm\infty$

**Important Note**

$$E[X] = \mu$$

That is the expected value of a RV is equivalent to the population mean

### 2.4.3 Some Rules

Let  $a$  and  $b$  be constants

$$E[a] = a$$

$$E[aX] = aE[X] = a\mu$$

$$E[aX + b] = aE[X] + E[b] = aE[x] + b = a\mu + b$$

Knowing these rules, lets try out a challenge:

### 2.4.4 A challenge in Expected Values

Find  $E[\bar{x}]$  where  $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

E.G find the expected value of an average



### 2.4.5 A solution

First, we can factor out the  $1/n$ , so we have

$$\frac{1}{n} E \left[ \sum_{i=1}^n x_i \right]$$

then we can say that the sum is just  $x_1 + x_2 + x_3 + \dots + x_n$ . This would lead us to the conclusion that

$$E[\bar{x}] = \frac{1}{n} \sum_{i=1}^n E[x_i]$$

If we let each  $x_i$  be a constant then we have:

$$E[\bar{x}] = \frac{1}{n} \sum_{i=1}^n x_i = \bar{x}$$

This makes sense as  $\bar{x}$  is the average value of  $x$ , and so the expected value of  $x$  for a very high  $n$  is the average value of  $x$

## 2.5 Variance and Covariance

### 2.5.1 Variance

The variance is the *dispersion*. Variance is defined as

$$\text{Var}(X) = E[(X - \mu)^2] = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx = \sigma^2$$

$$\widehat{\text{Var}(X)} = \sum (x_i - \bar{x})^2$$

### 2.5.2 Covariance

Assume we have two RVs,  $X$  and  $Y$ :

$$\text{Cov}(X, Y) = E[(X - \mu_x)(Y - \mu_y)]$$

$$\widehat{\text{Cov}(X, Y)} = \sum (x_i - \bar{x})(y_i - \bar{y})$$

The variance is, in essence, the sum of cross products.

### 2.5.3 Some generalizations on covariance:

If  $y$  tends to increase with  $x$ , then  $\text{Cov}(X, Y) > 0$

If  $Y$  tends to decrease with  $X$ , then  $\text{Cov}(X, Y)$  is **Less than 0**

If it appears as a random cloud, then  $\text{Cov}(X, Y)$  is approximately 0

### 2.5.4 A challenge

Let  $X$  be a RV. What is  $\text{Cov}(X, X)$ ?

### 2.5.5 A solution

$$\text{Cov}(X, X) = E[(X - \mu_x)(X - \mu_x)]$$

We see that the arguments in the left and right parens are exactly the same. Thus, we can rewrite the equation as:

$$\text{Cov}(X, X) = E[(X - \mu_x)^2] = \text{Var}(X)$$

### 2.5.6 A note on covariance and correlation

Correlation is covariance but standardized.

$$\begin{aligned} \text{Corr}(X, Y) &= \frac{\text{Cov}(X, Y)}{\text{SD}(X)\text{SD}(Y)} \\ &= \frac{E[(X - \mu_x)(Y - \mu_y)]}{\sigma_x \sigma_y} \\ \widehat{\text{Corr}(X, Y)} &= \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{ns_x s_y} \end{aligned}$$

## Chapter 3

# A Brief Discussion of Stationarity

In order for a time series to be considered stationary, it must satisfy *three* conditions:

- Constant Mean with Time
- Constant Variance with Time
- Constant Autocorellation with Time

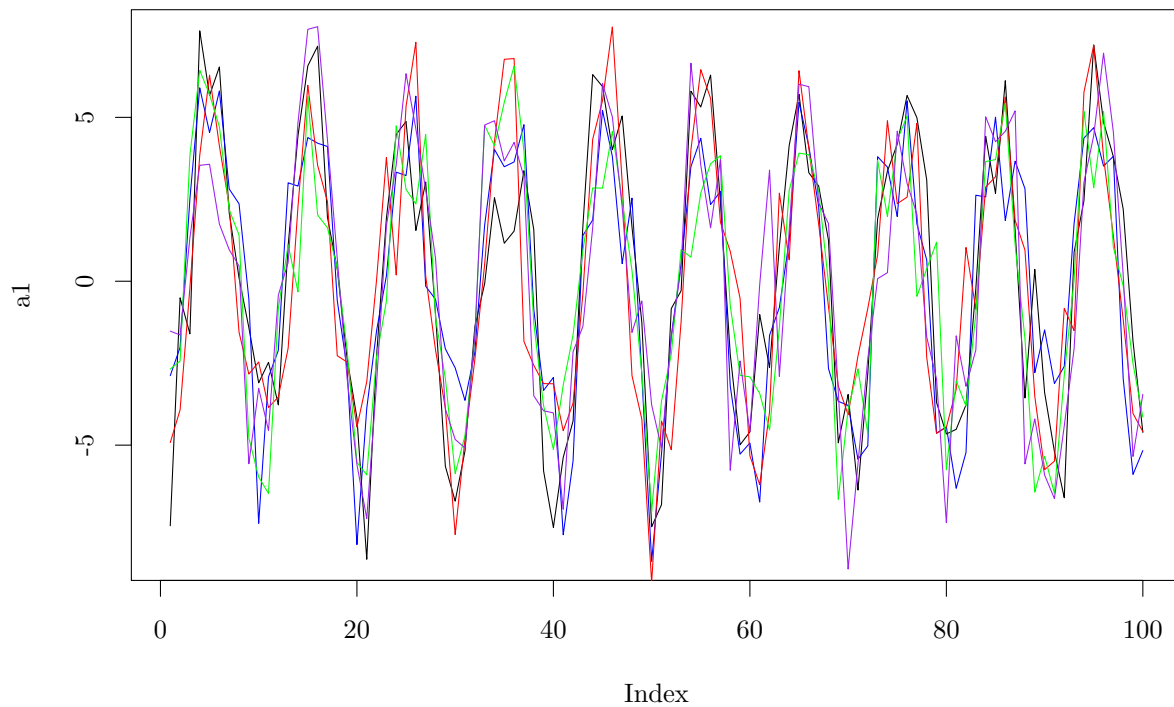
### 3.1 Condition One: Constant Mean

Mean is constant with time. That is  $E[X_t] = \mu$ . Note the lack of a little tiny t in mu! It is indepoent of tine! That is stationarity condition number 1!

Important result: If we assume constant mean, we can use *all* of the data to estimate the mean

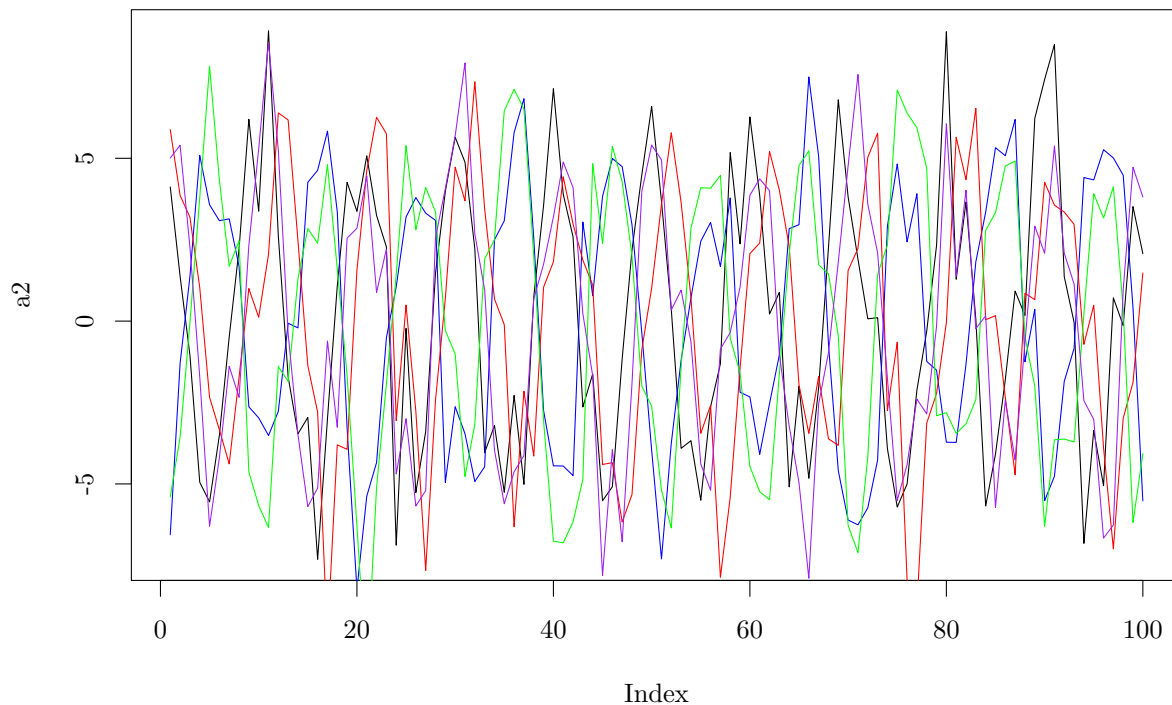
Here we can see after 5 realizations that the mean is clearly not constant with time

```
library(tswge)
a1 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(3,
0), vara = 3, plot = F)
b1 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(3,
0), vara = 3, plot = F)
c1 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(3,
0), vara = 3, plot = F)
d1 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(3,
0), vara = 3, plot = F)
e1 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(3,
0), vara = 3, plot = F)
```



Now let's look at 5 realizations which come from the same realization, which might not have a constant mean:

```
a2 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(runif(1,
0, 2 * pi), 0), vara = 3, plot = F)
b2 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(runif(1,
0, 2 * pi), 0), vara = 3, plot = F)
c2 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(runif(1,
0, 2 * pi), 0), vara = 3, plot = F)
d2 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(runif(1,
0, 2 * pi), 0), vara = 3, plot = F)
e2 = gen.sigplusnoise.wge(100, coef = c(5, 0), freq = c(0.1, 0), psi = c(runif(1,
0, 2 * pi), 0), vara = 3, plot = F)
```



We cannot say for sure whether the mean is constant or not in this case!

## 3.2 Condition Two: Constant Variance

*Variance does not depend on time*

Variance is *constant* and finite\*

That is :

$$\text{Var}[X_t] = \sigma^2 \neq \infty$$

If we can make this assumption, then we can use all of the data to make the variance. This is typically the hardest one to tell, and it will take some practice to see.

## 3.3 Condition 3: Constant autocorrelation

*Correlation of  $X_{t_1}$  and  $X_{t_2}$  only depends on  $t_2 - t_1$*

That is, the correlation between points depends only on how far apart they are in time, not where they are in time.

To describe this mathematically:

Let

$$h = t_2 - t_1$$

then

$$\text{Cor}(X_t, X_{t+h}) = \rho_h$$

$\rho$  represents the population correlation coefficient.



## Chapter 4

# Autocorrelation

```
library(tswge)
library(ggplot2)
acdf <- function(vec) {
  vacf <- acf(vec, plot = F)
  with(vacf, data.frame(lag, acf))
}
ggacf <- function(vec) {
  ac <- acdf(vec)
  ggplot(data = ac, aes(x = lag, y = acf)) + geom_hline(aes(yintercept = 0)) +
    geom_segment(mapping = aes(xend = lag, yend = 0))
}

tplot <- function(vec) {
  df <- data.frame(X = vec, t = seq_along(vec))
  ggplot(data = df, aes(x = t, y = X)) + geom_point() + geom_line()
}
```

### 4.1 Independence

Two events are independent in a time series if the probability that an event at time  $t$  occurs in no way depends on the occurrence of any event in the past or affects any event in the future. Mathematically, this is written as:

$$\text{Independence : } P(x_{t+1}|X_t) = P(X_{t+1})$$

### 4.2 Serial Dependence / Autocorrelation

#### 4.2.1 A definition

If two events are independent, their correlation is 0. That is if  $X_t$  and  $X_{t+k}$  are independent,  $\rho_{x_t, x_{t+k}} = 0$

*Corollary: If the correlation between two variables is not zero, then they are not independent*

In other words if  $\rho_{x_t, x_{t+k}} \neq 0$ , they are **not** independent.

### 4.2.2 Autocorrelation Plots

### 4.2.3 Dependent(ish) data

In time series we look at the autocorrelation between  $X_t$  and  $X_{t+1}$  etc (with  $t$  and  $t+1$  it is lag 1 autocorrelation)

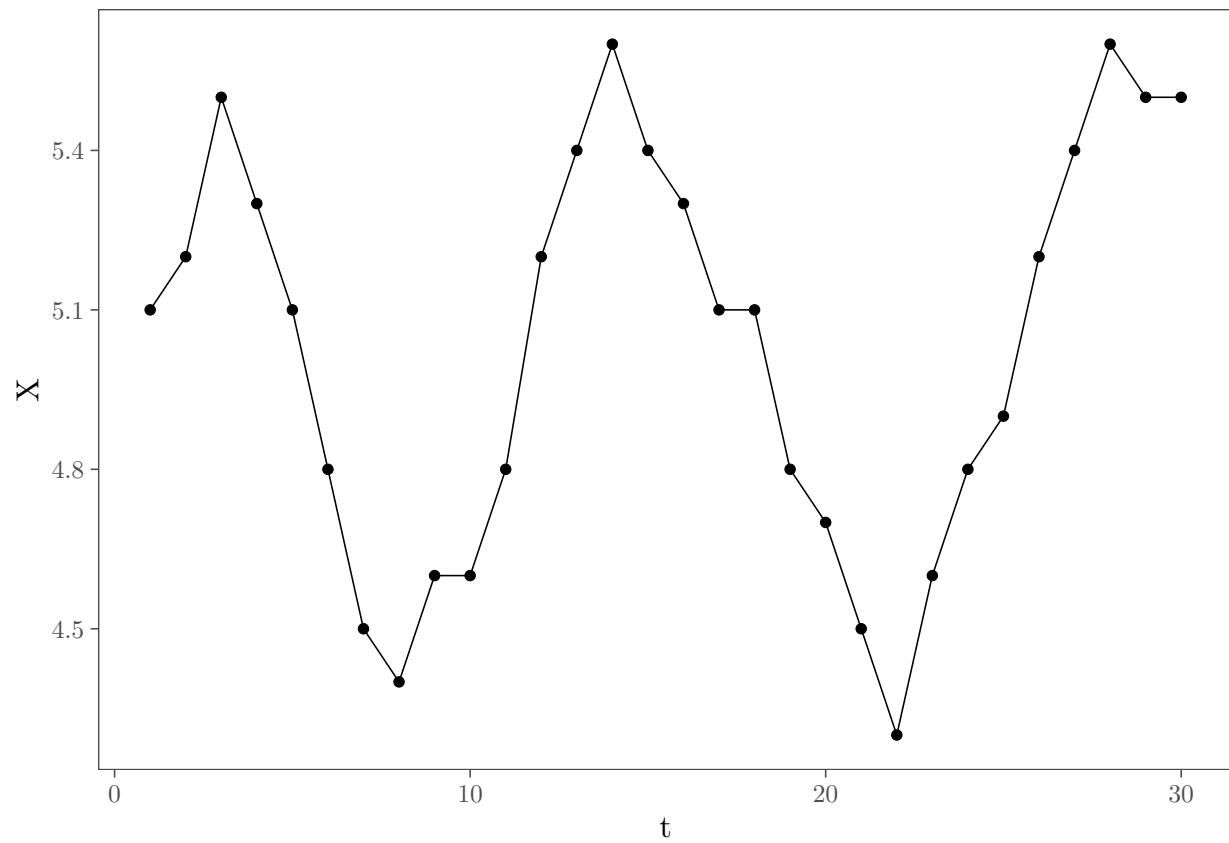
For example, visually, let us define a vector,  $Y5$  and take its autocorrelation:

```
Y5 <- c(5.1, 5.2, 5.5, 5.3, 5.1, 4.8, 4.5, 4.4, 4.6, 4.6, 4.8, 5.2, 5.4, 5.6,
        5.4, 5.3, 5.1, 5.1, 4.8, 4.7, 4.5, 4.3, 4.6, 4.8, 4.9, 5.2, 5.4, 5.6, 5.5,
        5.5)
knitr::kable(acfdf(Y5))
```

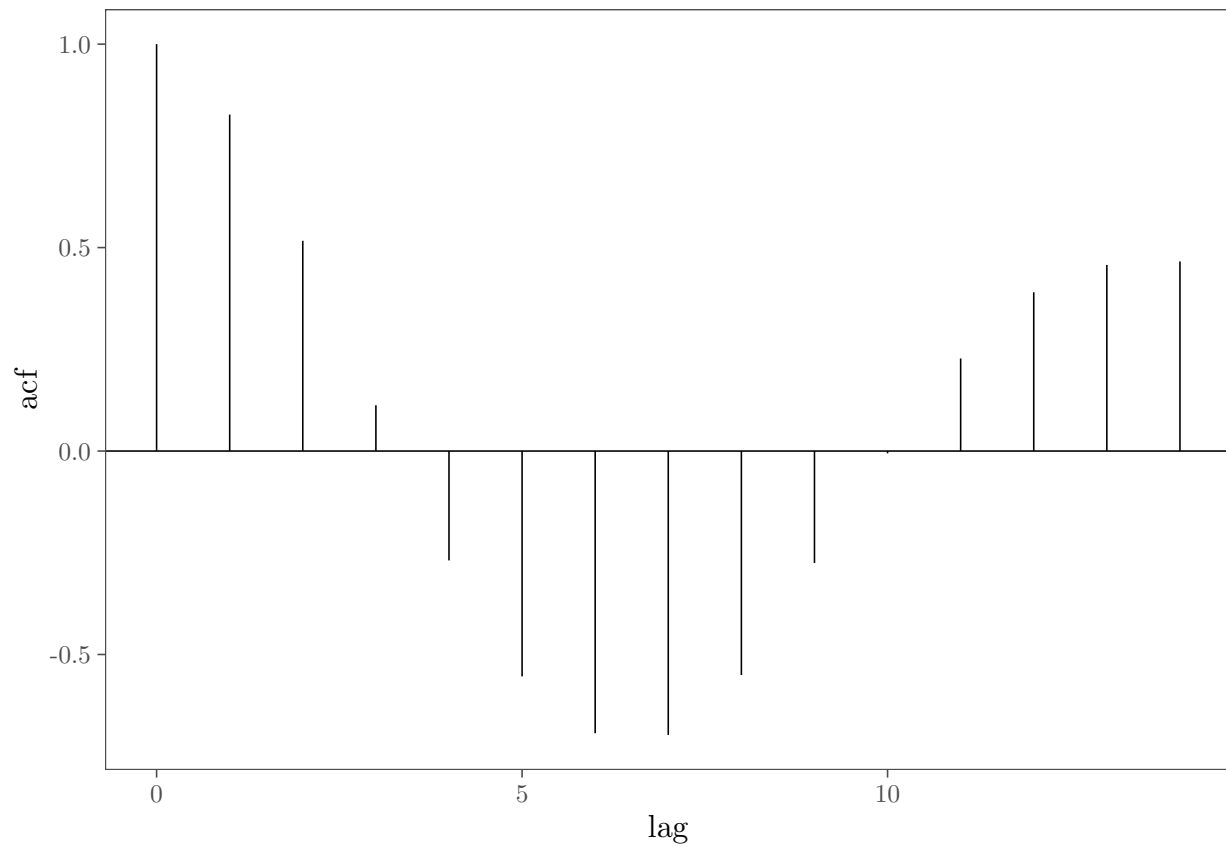
lag	acf
0	1.0000000
1	0.8268482
2	0.5166514
3	0.1125421
4	-0.2687774
5	-0.5538901
6	-0.6935454
7	-0.6979912
8	-0.5505557
9	-0.2752524
10	-0.0056082
11	0.2274804
12	0.3902111
13	0.4573468
14	0.4660549

```
tplot(Y5) + ggthemes::theme_few()
```





```
ggacf(Y5) + ggthemes::theme_few()
```



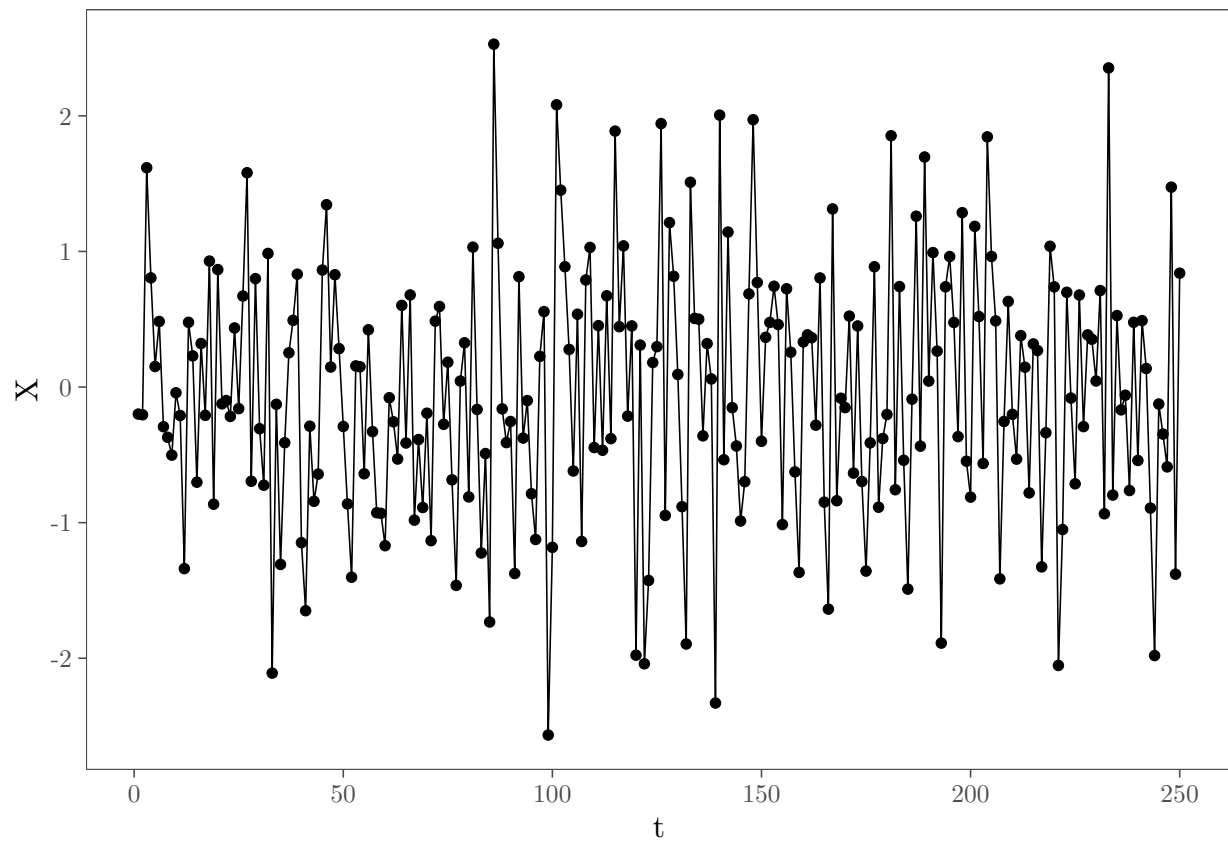
#### 4.2.4 Independent(ish) data!

Now let us look at autocorrelation of *independent* data

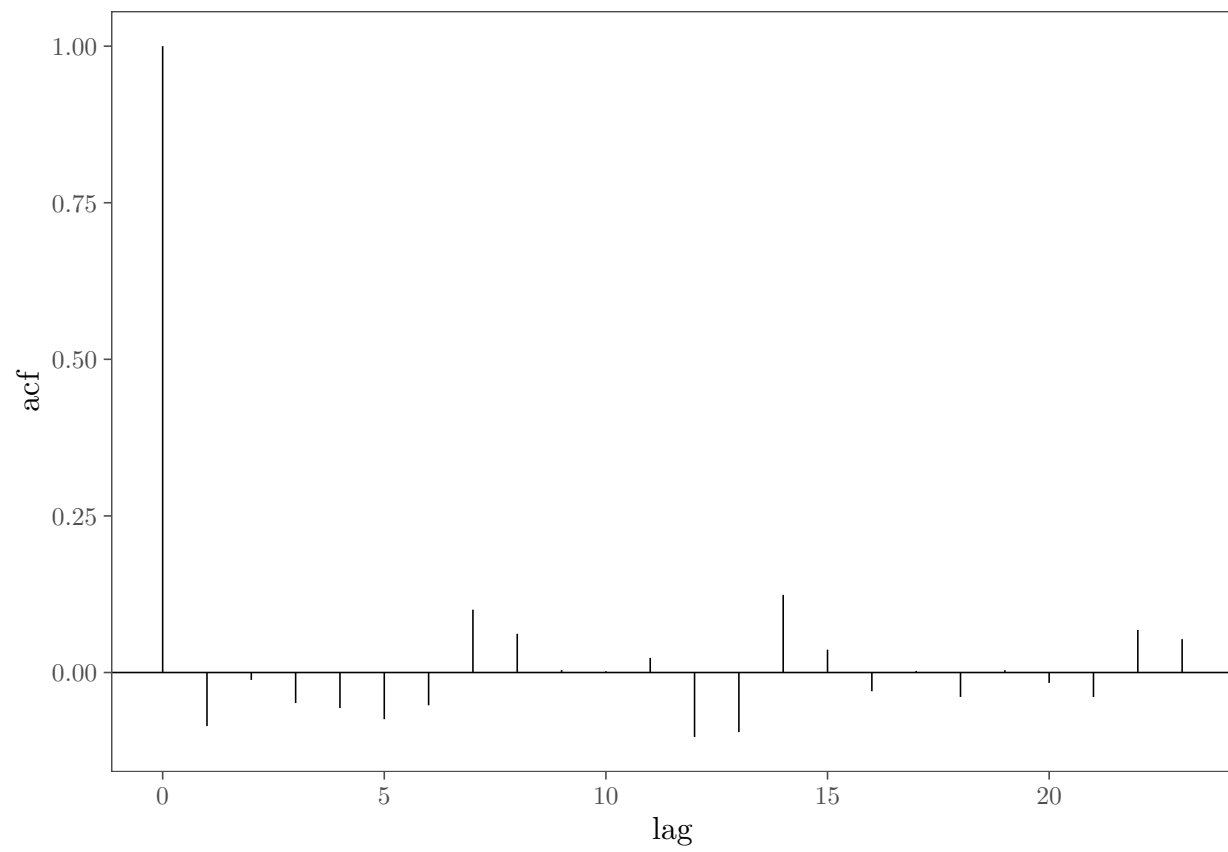
```
xs = gen.arma.wge(n = 250)
```

Let's check it out

```
tplot(xs) + ggthemes::theme_few()
```



```
ggacf(xs) + ggthemes::theme_few()
```



We see that the autocorrelation is more or less zero

## Chapter 5

# Autocorrelation Concepts and Notation

### 5.1 Theoretical concept of $\rho$

$$\rho = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} = \frac{\text{covariance between } X \text{ and } Y}{\sigma_X \sigma_Y}$$

In a stationary time series, we have autocorrelation and autocovariance

### 5.2 autocovariance

$$\gamma_k = E[(X_t - \mu)(X_{t+k} - \mu)]$$

### 5.3 autocorrelation

$$\rho_k = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma_{X_t} \sigma_{X_{t+k}}} = \frac{E[(X_t - \mu)(X_{t+k} - \mu)]}{\sigma_X^2} = \frac{\gamma_k}{\sigma_X^2}$$

This works because of constant variance and constant mean!

Note that in a stationary time series:

$$\sigma_X^2 = 1E[(X_t - \mu)^2] = E[(X_t - \mu)(X_t - \mu)] = \gamma_0$$

Therefore:

$$\rho_k = \frac{\gamma_k}{\gamma_0}$$

### 5.4 Stationary Covariance

Correlation is not affected by where, only by how far apart, that is:

$$\rho_h = \text{Cor}(X_t, X_{t+h})$$

Let us try it out with this little snippet:

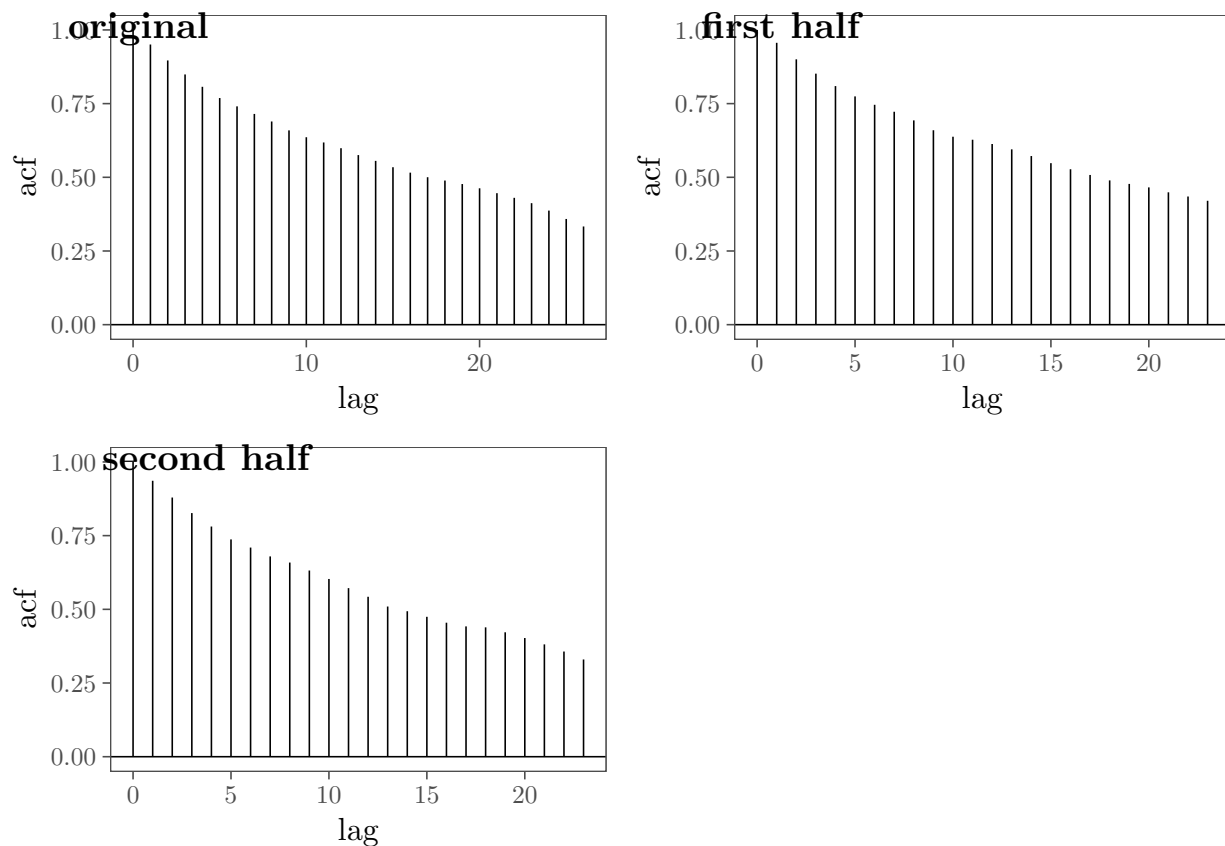
```

ggsplitacf <- function(vec) {
  h1 <- vec[1:(length(vec)/2)]
  h2 <- vec[(length(vec)/2):length(vec)]
  first <- ggacf(vec) + ggthemes::theme_few()
  second <- ggacf(h1) + ggthemes::theme_few()
  third <- ggacf(h2) + ggthemes::theme_few()
  cowplot::plot_grid(first, second, third, labels = c("original", "first half",
    "second half"), nrow = 2, align = "v")
}

```

```
Realize = gen.arma.wge(500, 0.95, 0, plot = T, sn = 784)
```

```
ggsplitacf(Realize)
```

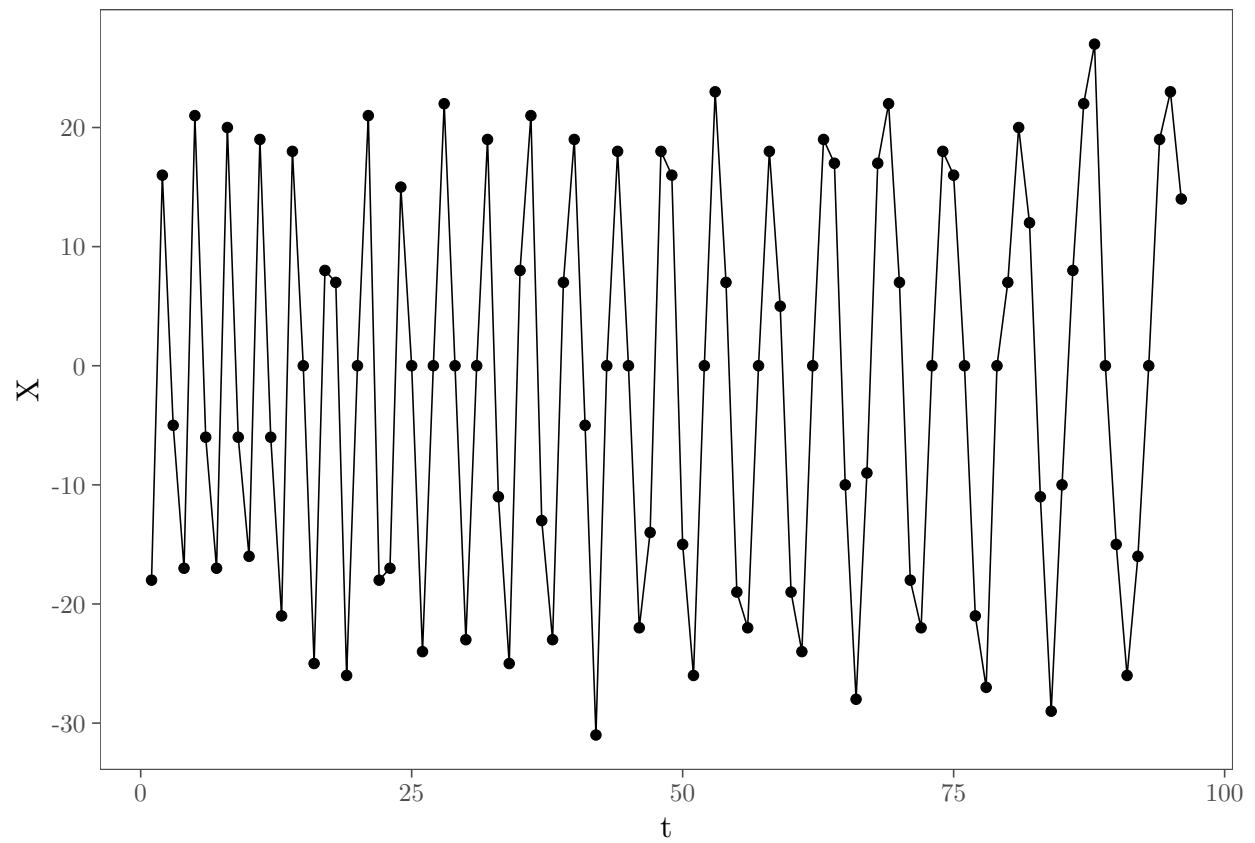


This looks pretty stationary to me! Let's try with some sample data:

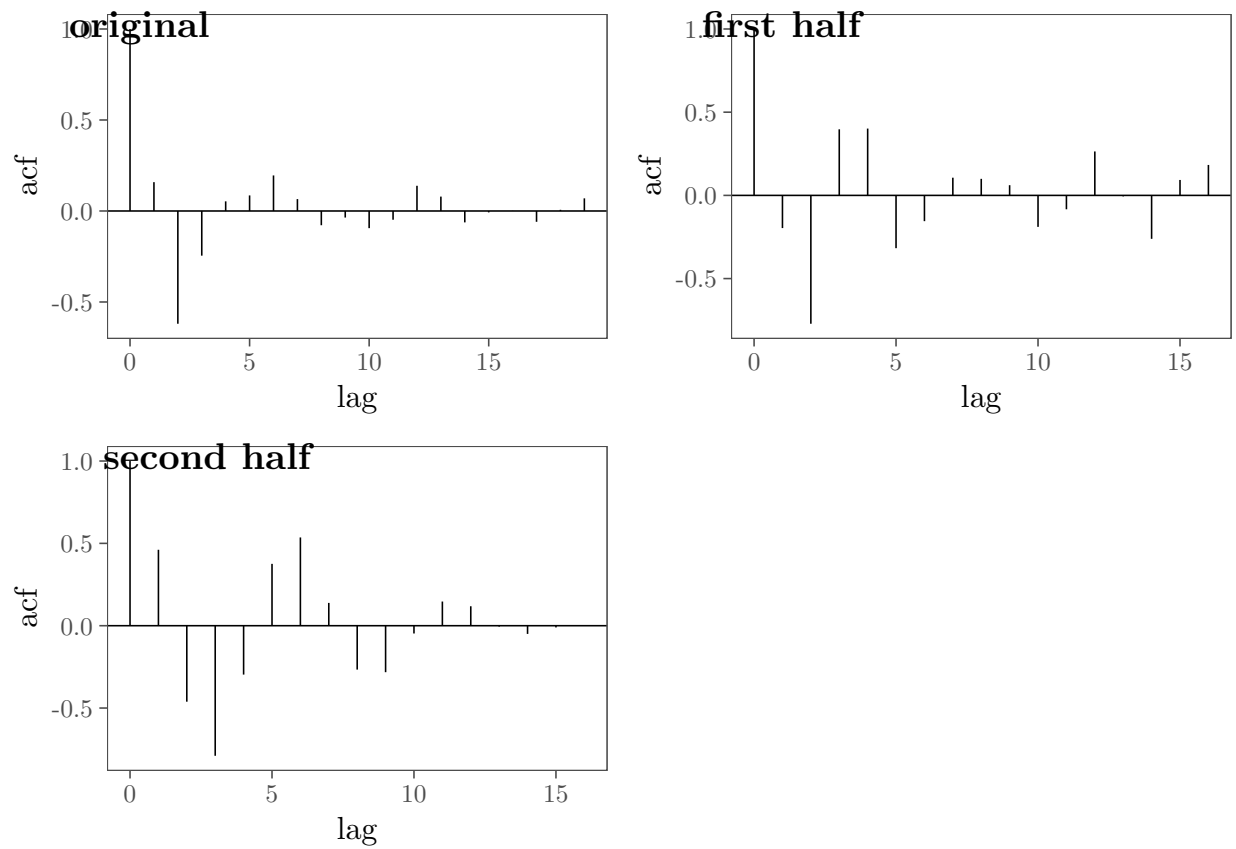
```

data("noctula")
tplot(noctula) + ggthemes::theme_few()

```



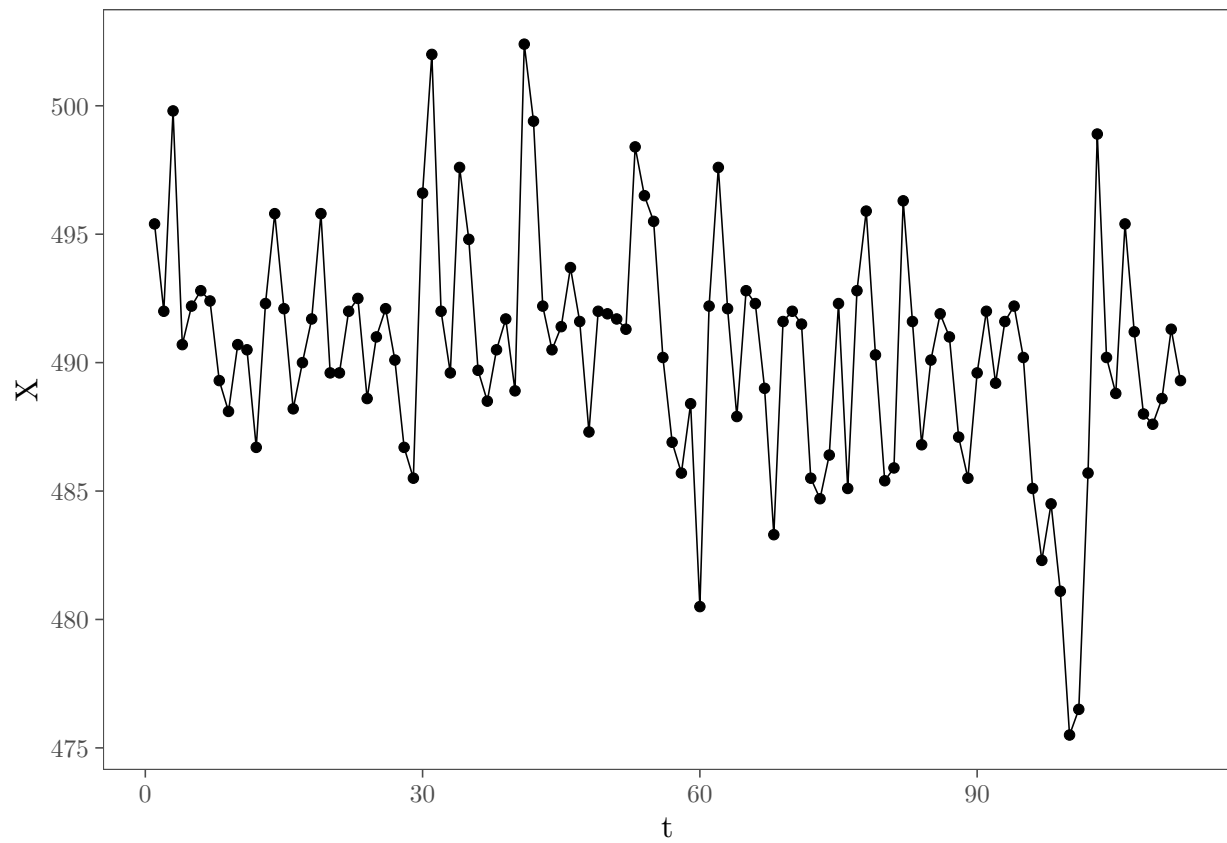
```
ggsplitacf(noctula)
```



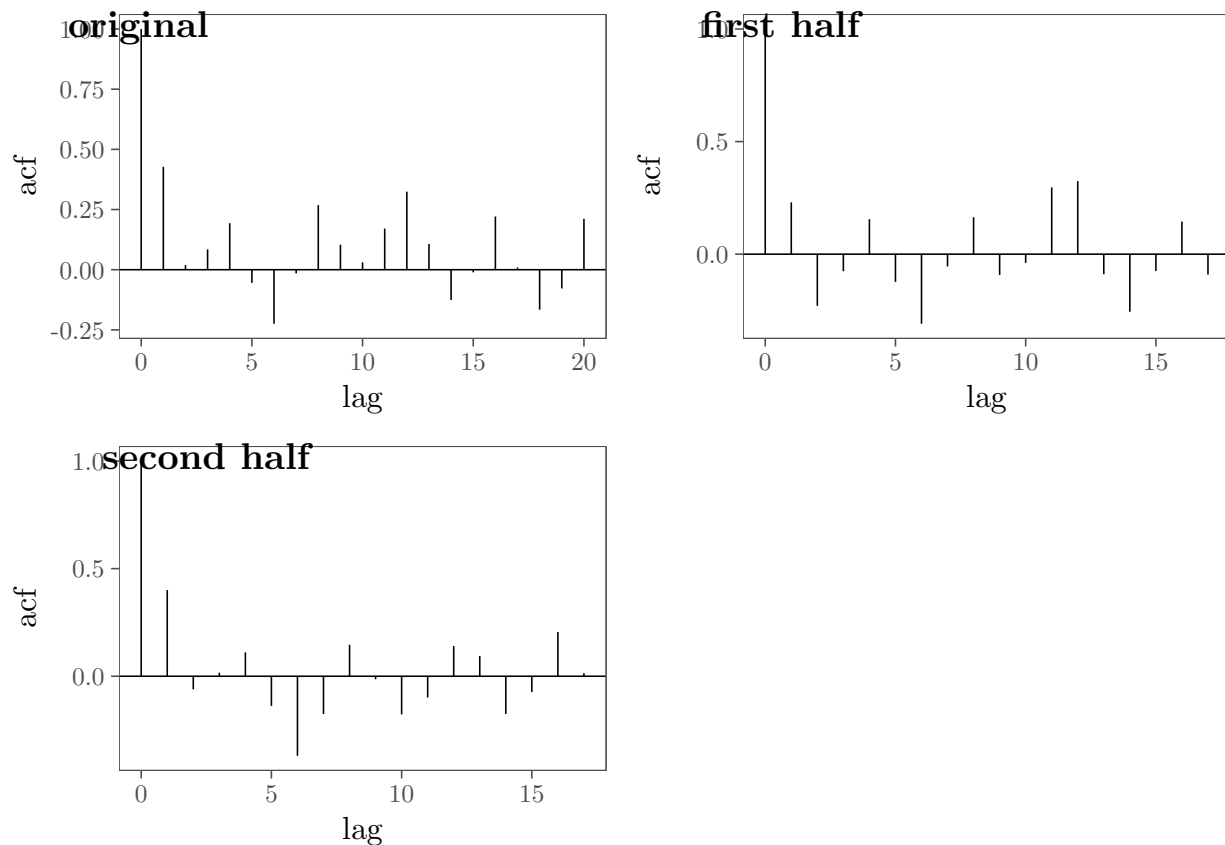
So in this case our mean probably looks constant, our variance could be constant as well, however the ACFs are clearly different. Let's look at a different dataset:

```
data(lavon)
tplot(lavon) + ggthemes::theme_few()
```





```
ggsplitacf(lavon)
```



In this case, our mean looks maybe to be constant around 495, our variance does not appear to be constant, and our ACF looks pretty good, however I would say overall this probably is not stationary, due to the variance and slightly off ACF.

## 5.5 Stationary Time Series: Putting it Together

### 5.5.1 Workflow:

- Step 1: Check condition 1
  - see if the assumption is more or less met
- Step 2: Check condition two
  - same thing
- Step 3: Check for condition three

If one of the things is iffy, and the other two arent, we are going to be just fine.

## Chapter 6

# Practical Autocorrelation

### 6.1 Estimation!

#### 6.1.1 $\rho_k \rightarrow 0$

For a stationary time series, if the autocorrelation approaches zero, then :

A Single realization lets us estimate mean, variance, and autocovariance!

Remember that  $\rho_k = \frac{\gamma_k}{\gamma_0}$ , and we use n-k pairs to calculate it (the summation)

##### 6.1.1.1 Mean

Just calculate the mean normally for this case.

##### 6.1.1.2 Variance

$$\text{Var}(\bar{X}) = \frac{\sigma^2}{n} \sum_{k=-(n-1)}^{n-1} \left(1 - \frac{|k|}{n}\right) \rho_k$$

$\sigma^2$  is calculated as normal, we will see  $\rho_k$  next!

remember !

Now it is time for some code!

```
library(glue)

##
## Attaching package: 'glue'
## The following object is masked from 'package:dplyr':
##
## collapse

xbar <- function(xs) {
  mean(xs)
}

ghat_zero <- function(xs) {
  summand <- (xs - xbar(xs))^2
  mean(summand)
}
```

```
}

ghat_one <- function(xs) {
  lhs <- xs[1:(length(xs) - 1)] - xbar(xs)
  rhs <- xs[2:length(xs)] - xbar(xs)
  summand <- lhs * rhs
  summate <- sum(summand)
  summate/length(xs)
}

rhatat_zero <- 1
rhatat_one <- function(xs) {
  ghat_one(xs)/ghat_zero(xs)
}

v <- c(76, 70, 66, 60, 70, 72, 76, 80)
xbar(v)
```

```
## [1] 71.25
```

```
ghat_zero(v)
```

```
## [1] 34.9375
```

```
ghat_one(v)
```

```
## [1] 14.74219
```

```
rhatat_one(v)
```

```
## [1] 0.4219589
```

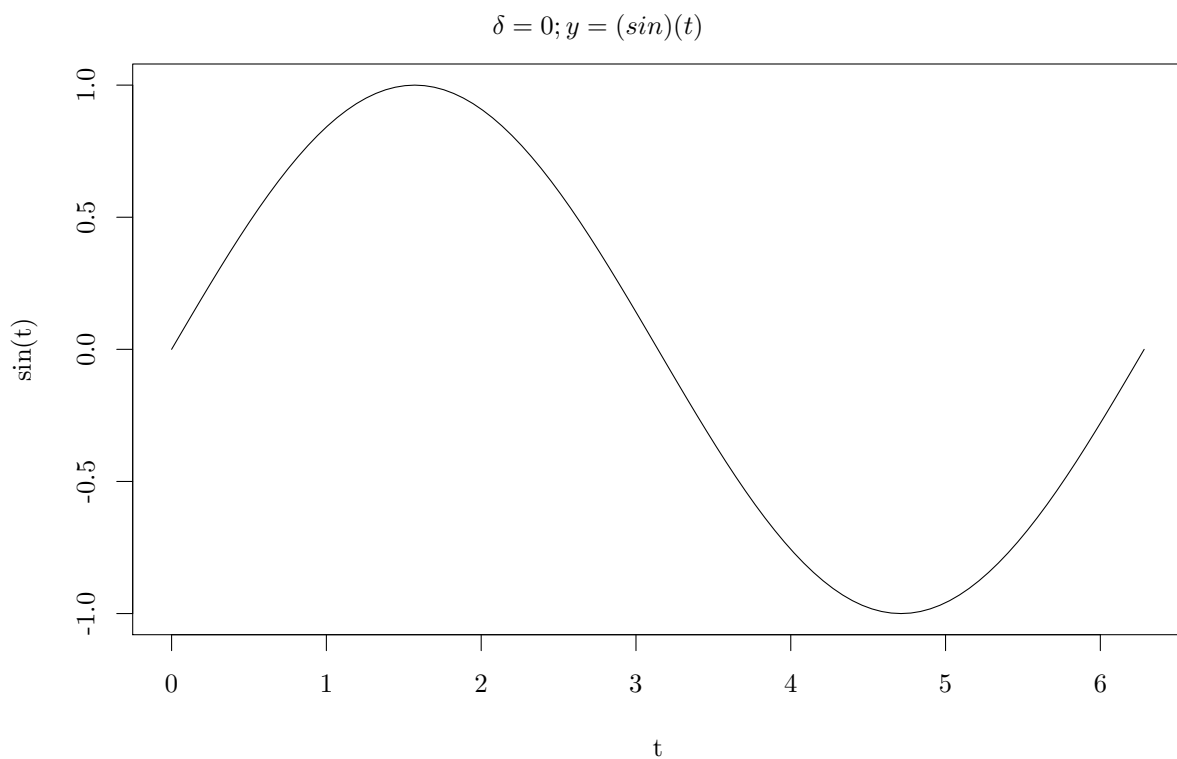
## Chapter 7

# The Frequency Domain

### 7.1 Review of sine and cosine

#### 7.1.1 Phase shifts

This is what a sine wave looks like



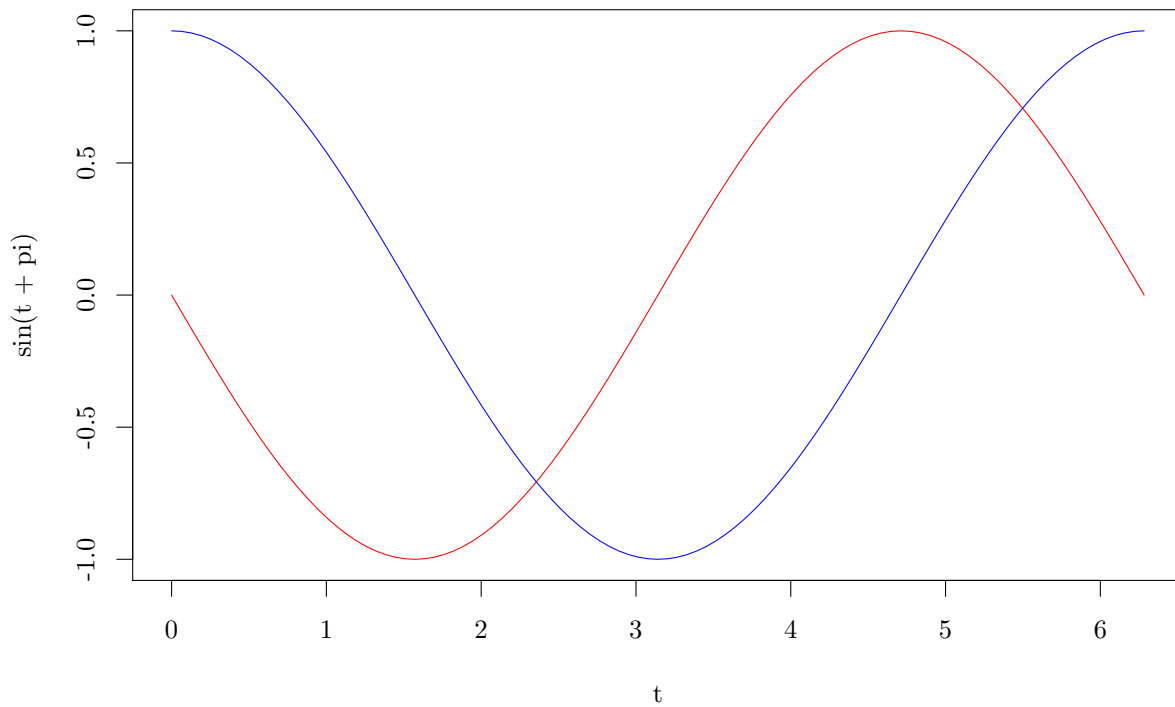
We can phase shift the sin wave by adding

RED:

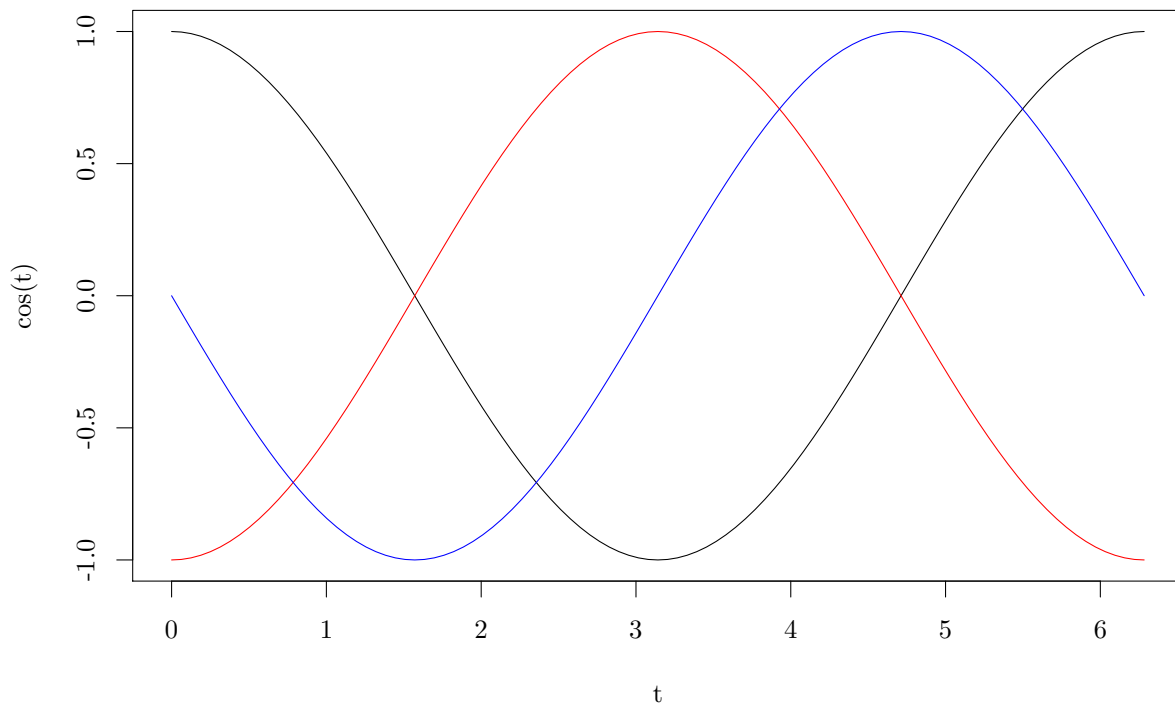
$$\delta = \pi; y = \sin(t + \pi)$$

BLUE:

$$\delta = \frac{\pi}{2}; y = \sin\left(t + \frac{\pi}{2}\right)$$



Now cosine



## 7.2 Frequency and Period

Let us say  $f(x)$  is a periodic function. If  $f(x)$  has period  $p$ , then  $p$  is the smallest value S.T.

$$f(x) = f(x + kp)$$

Cool

## 7.3 psueudo-periodicity

Data are *psuedo-periodic* with period  $p$  if  $p$  is the smallest value such that a cycle appears to repeat itself.

## 7.4 Aperiodicity

Aperiodic data is data in which no  $p$  exists.

## 7.5 Frequency

Frequency is calculated as :  $\frac{1}{p}$

It symbolizes the number of cycles per unit(number of periods per unit)

## 7.6 Frequency Terminology

Frequency of a trigonometruc function:

Let  $f(x) = \sin(2\pi ft)$

Then we have that :

$$p = 1/f$$

$$\nu = f$$

Consider:

$$\sin(Bt + C)$$

Then

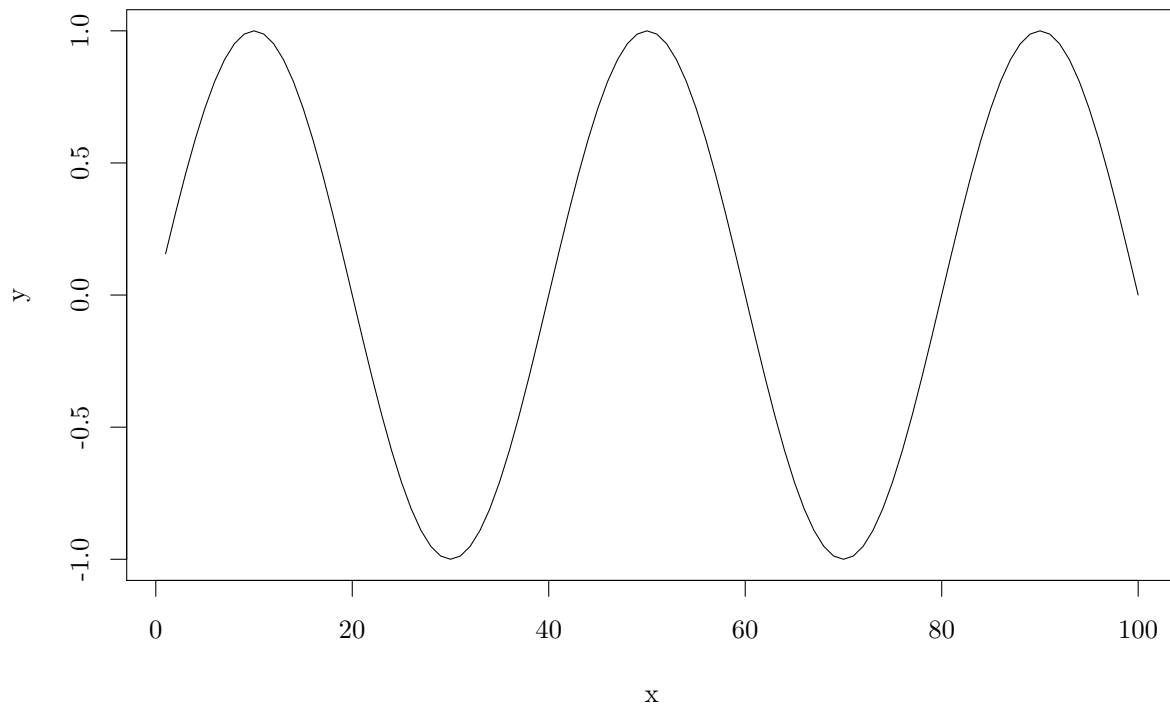
$$p = \frac{2\pi}{B}$$

$$\nu = \frac{B}{2\pi}$$

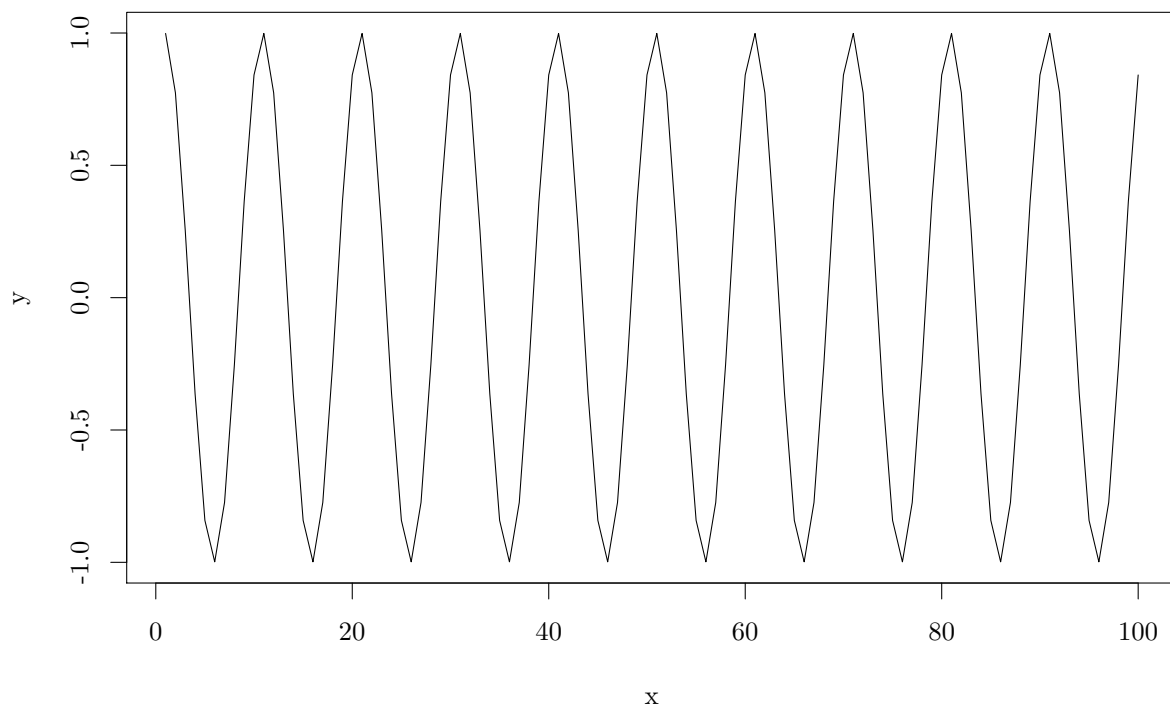
## 7.7 Composite sine function

```
t <- seq(1, 100, length = 100)
y1 <- sin(2 * pi * 0.025 * t)
y2 <- sin(2 * pi * 0.1 * t + 1)
y3 <- sin(2 * pi * 0.15 * t + 2.5)
lineplotter <- function(x, y) {
  plot(x, y, type = "l")
}

lineplotter(t, y1)
```

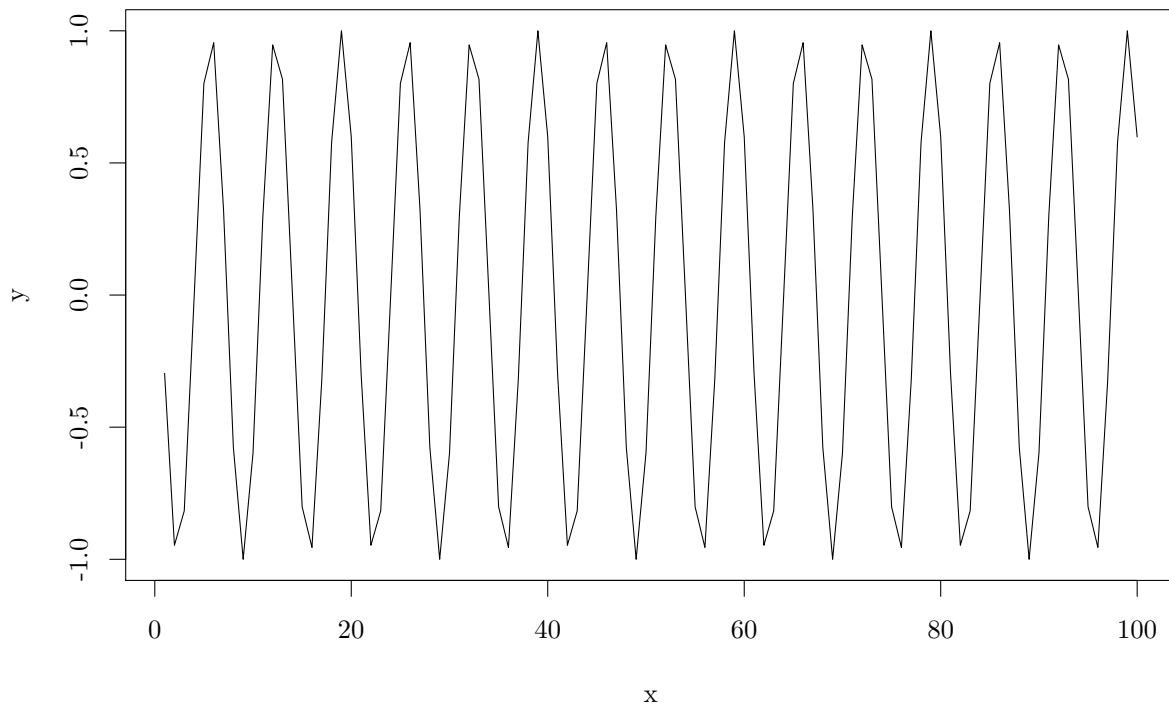


```
lineplotter(t, y2)
```

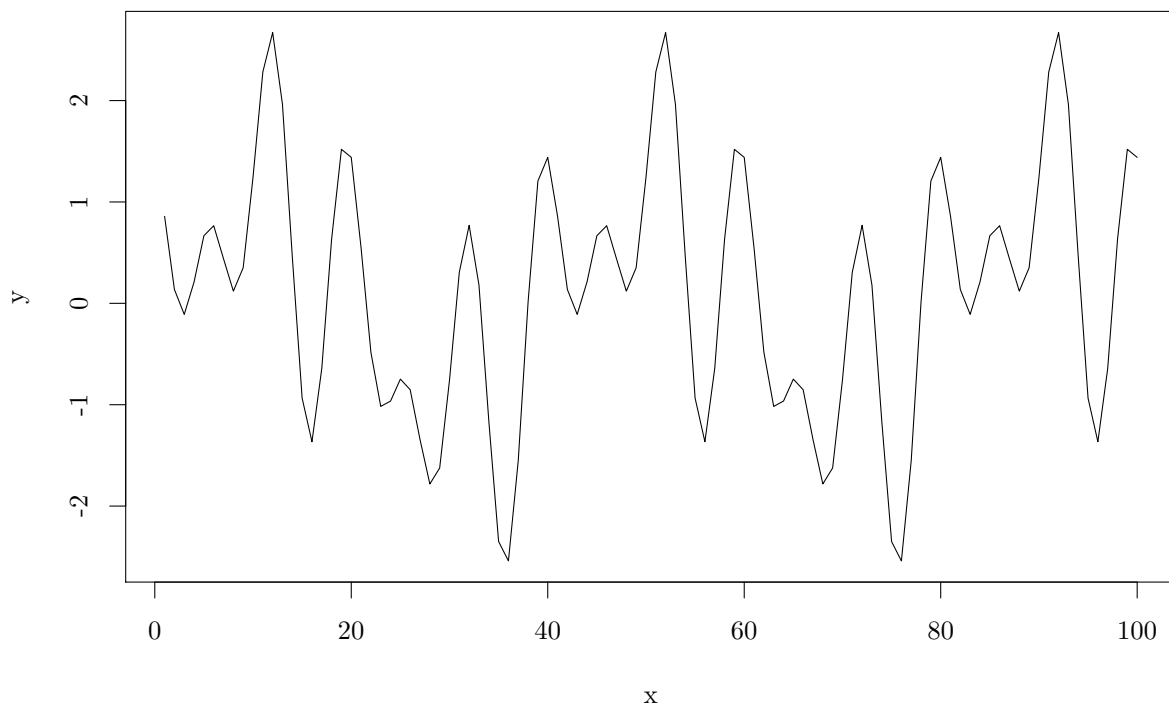


```
lineplotter(t, y3)
```





```
lineplotter(t, y1 + y2 + y3)
```



this is a signal with three frequencies!

## 7.8 Fundamental idea

A lot of realizations are made up of components of various frequencies, and some do not have frequency at all. Uncovering the frequency content of a dataset (or lack thereof) allows us to better understand the process generating the data.

### 7.8.1 A Brief Look at Fourier Series

When we do spectral analysis, we are generally going to be decomposing functions into sine and cosine terms, in order to better analyze the frequency of the data.

#### 7.8.1.1 Fourier Series

You should just know this by this point, if you dont, google it. Too lazy to tex up the equations. Basically, we can decompose any function into a bunch of sines and cosines. It is pretty rad. The important conclusion of the existence of fourier series is that we can represent non cyclic functions as a bunch of sines and cosines, which will allow us to do spectral analysis of them.

## Chapter 8

# Spectral Density

A tool to identify the frequency content of a (stationary) time series.

### 8.1 Derivation of spectral density of white noise

$$S_x(f) = 1 + 2 \sum_{k=1}^{\infty} \rho_k \cos(2\pi f k), |f| \leq 0.5$$

For white noise,  $\rho_0 = 1$ , if  $k \neq 0$ ,  $\rho_k = 0$

This tells us that for white noise:

$$S_x(f) = 1 + 2 * 0 = 1$$

In other words: All frequencies are equally present in white noise

### 8.2 Estimating Spectral Density from Real Data

With real data, we don't have an infinite amount, and we do not know what the true autocorrelation is. So we are probably going to have to make some estimates.

Lets first estimate autocorrelation:

$$S_x(f) = 1 + 2 \sum_{k=1}^{\infty} \widehat{\rho}_k \cos(2\pi f k), |f| \leq 0.5$$

Now lets deal with the infinity, which is killing us. We know that:

$$\hat{\rho}_1 = \frac{\frac{1}{n} \sum_{t=1}^n (X_t - \bar{X})(X_{t+1} - \bar{X})}{\hat{\gamma}_0}$$

And (check yourself to see the math works)

$$\hat{\rho}_{n-1} = (X_1 - \bar{X})(X_n - \bar{x}) / (n\hat{\gamma}_0)$$

We can then rewrite the sample spectral density/sample spectrum as!:

$$S_x(f) = 1 + 2 \sum_{k=1}^{n-1} \widehat{\rho}_k \cos(2\pi f k), |f| \leq 0.5$$

However, this estimate of autocorrelation cannot be very good, since it is estimated using one cross product/pair. That's like estimating the mean with one value!

So, let's stop this sum somewhere more reasonable.

$$\widehat{S_x(f)} = 1 + 2 \sum_{k=1}^M \widehat{\rho_k} \cos(2\pi f k), |f| \leq 0.5$$

Where  $M$  is less than  $n - 1$ . This lets us have a bit more freedom to decide what  $M$  should be.

Let us now rewrite the estimate of spectral density as:

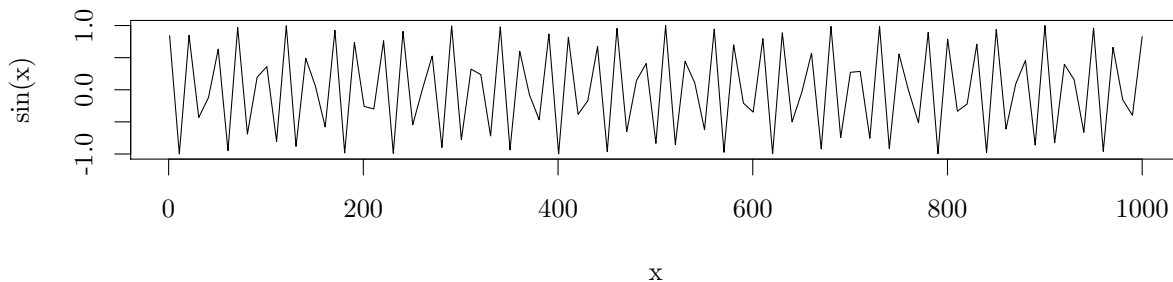
$$\widehat{S_x(f)} = \lambda_0 + 2 \sum_{k=1}^M \lambda_k \widehat{\rho_k} \cos(2\pi f k), |f| \leq 0.5$$

Where  $|\lambda_k|$  decreases as  $k$  increases. With this, we have  $M$  minimizing and limiting the impact of sample autocorrelations when  $k$  gets pretty close to  $n$ , and  $\lambda_k$  further decreases the impact of the poor sample autocorrelations at high values of  $k$ . These are called **window functions**. In our case, we will use what is called the Parzen window. Other windows (every statistician in the 60's had their own window) include: the Tukey window, the Bartlett window, etc. We call this the smooth spectral density estimator.  $M$  is frequently chosen to be  $2\sqrt{n}$ .

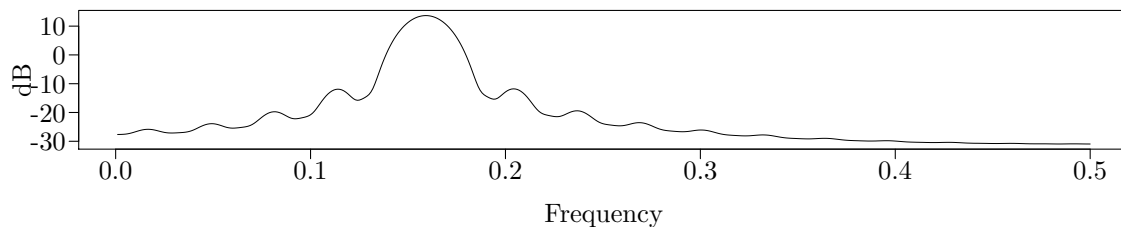
### 8.3 Spectral Density in the Wild

First, let's look at a simple example:

```
x <- 1:1000
y <- sin(x)
par(mfrow = c(2, 1))
curve(sin(x), 1, 1000)
parzen.wge(y)
```



Parzen Window Truncation point:  $M = 63$

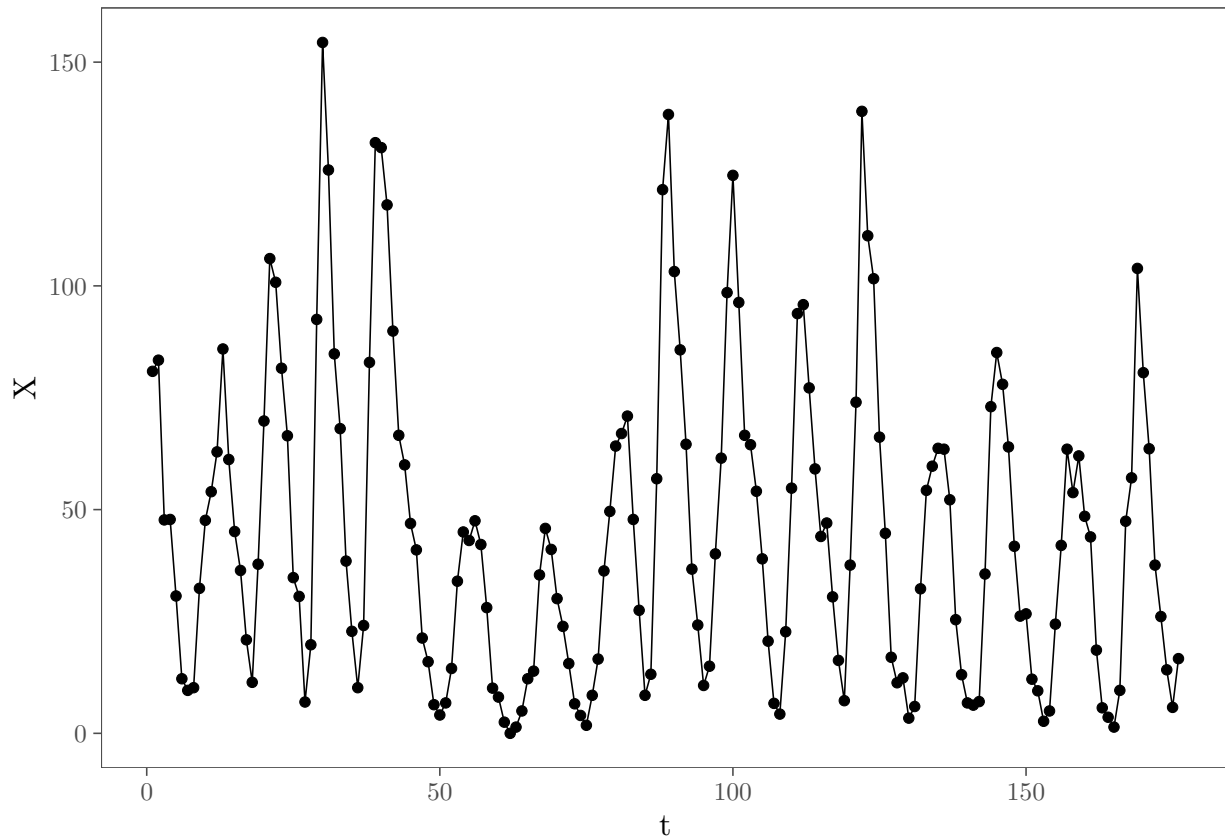


Shockingly, we have a frequency peak at roughly  $\frac{1}{2\pi} = 0.1591549$ . If I did a better job of generating the data, it would be a super shark peak and nothing else, however I did not. Let's look at sunspot data now:

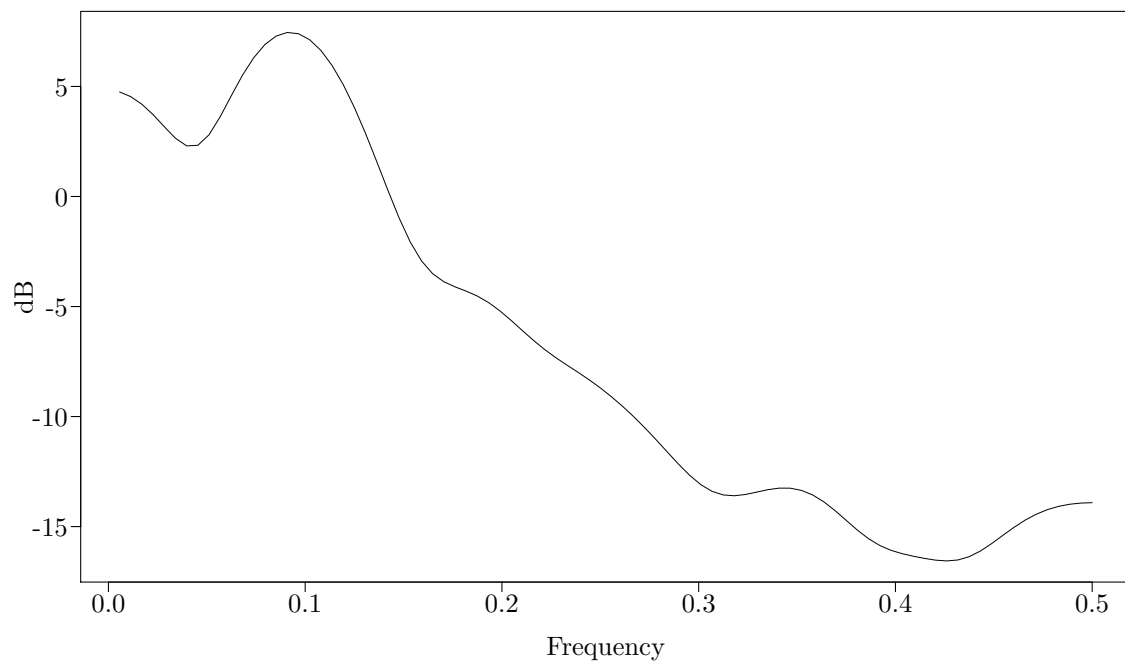
```

par(mfrow = c(1, 1))
data(sunspot.classic)
tplot(sunspot.classic) + ggthemes::theme_few()
parzen.wge(sunspot.classic)

```

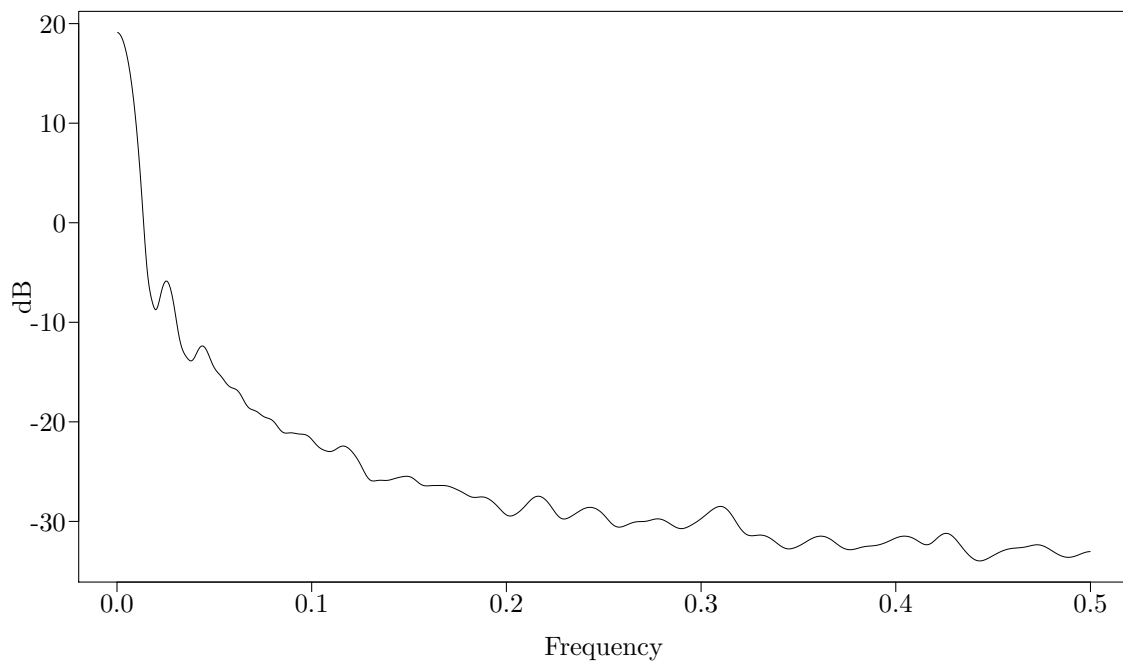


Parzen Window Truncation point:  $M = 26$



It looks like we have a frequency peak at about 0.1, which makes sense because the classic sunspot pseudoperiod is about 11 years. Lets look at some aperiodic stock data now:

```
getSymbols("GOOG", src = "yahoo")
plot(GOOG[, "GOOG.Close"])
parzen.wge(GOOG$GOOG.Close)
```









## Chapter 9

# AR(1) Models and Filtering

### 9.1 Algebra review

Be able to find the roots of a 2nd order polynomial

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$z = a + bi$$

$$i = \sqrt{-1}$$

a is real b is imaginary

$$z^* = a - bi$$

$z^*$  is complex conjugate

It is basically a vector so absolute value is just the magnitude

Unit circle is the values in the complex plane in which the magnitude of z is equal to one. If Z is a complex number with magnitude greater than one, it is outside the unit circle.

```
quad_form <- function(a, b, c) {  
  rad <- b^2 - 4 * a * c  
  if (is.complex(rad) || all(rad >= 0)) {  
    rad <- sqrt(rad)  
  } else {  
    rad <- sqrt(as.complex(rad))  
  }  
  round(cbind(-b - rad, -b + rad)/(2 * a), 3)  
}
```

```
quad_form(c(1, 4), c(1, -5), c(6, 2))
```

```
##           [,1]      [,2]  
## [1,] -0.500-2.398i -0.500+2.398i  
## [2,]  0.625-0.331i  0.625+0.331i
```

```
# [,1] [,2] [1,] -0.500-2.398i -0.500+2.398i [2,] 0.625-0.331i 0.625+0.331i
```

## 9.2 Linear filters

A filter will turn  $Z_t$  into  $X_t$

for example:

$$\text{difference : } X_t = Z_t - Z_{t-1}$$

Moving average also a filter (smoother)

### 9.2.1 Example

Consider  $Z_1 = 8, Z_2 = 14, Z_3 = 14, Z_4 = 7$

If we apply the difference filter to this, we have that:

$$X_2 = 6, X_3 = 0, X_4 = 7$$

**Note:** the differenced data are a realization of length  $n-1$

We can use the difference filter to remove the wandering behavior

### 9.2.2 5 point moving average

We can only get a realization of  $n-4$ . ( $n$ -npoints-1). What does the 5 point moving average do (it averages a point and two points ahead and two behind). This filter can filter out some frequencies

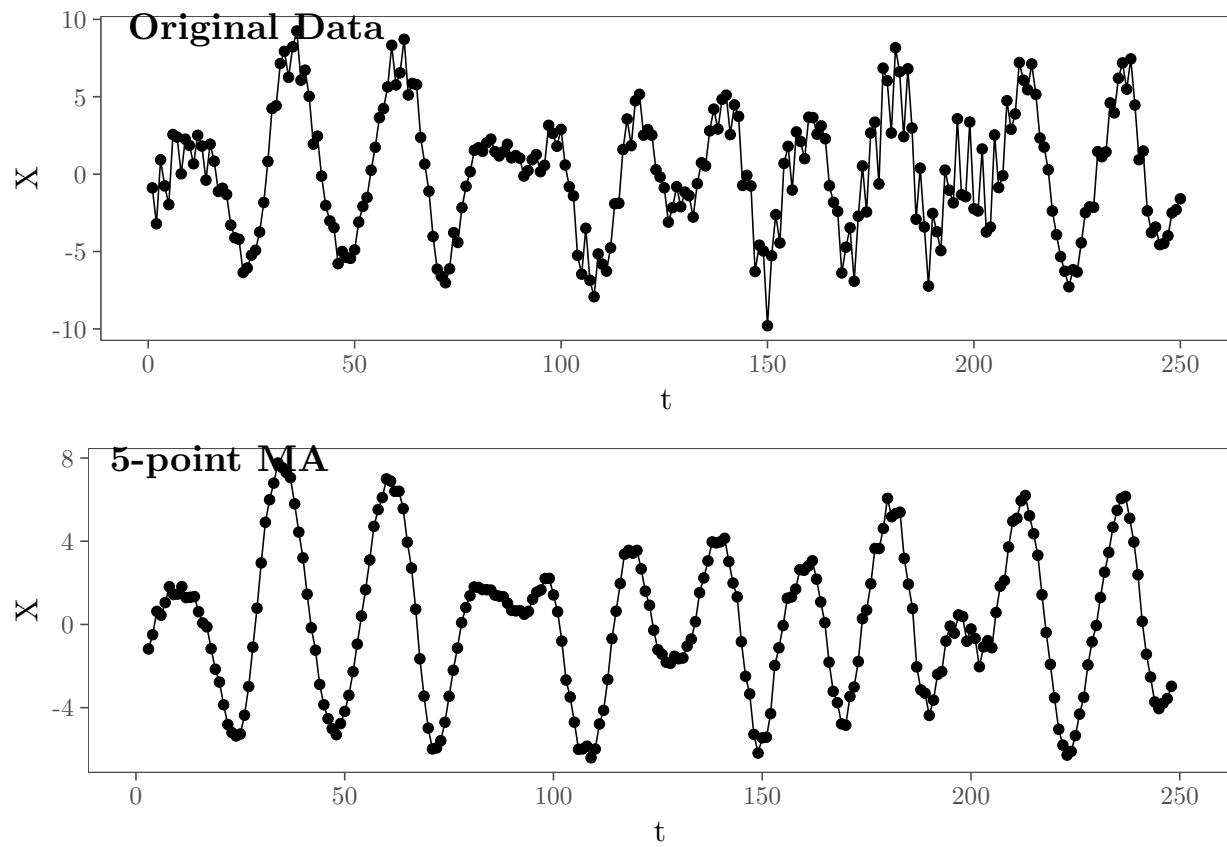
## 9.3 Types of filters:

- Low pass filter
  - Filters out high frequency
  - Such as 5 point moving average, it smooths
- High pass filter
  - Leaves high freq but removes low freq
  - Such as differencing

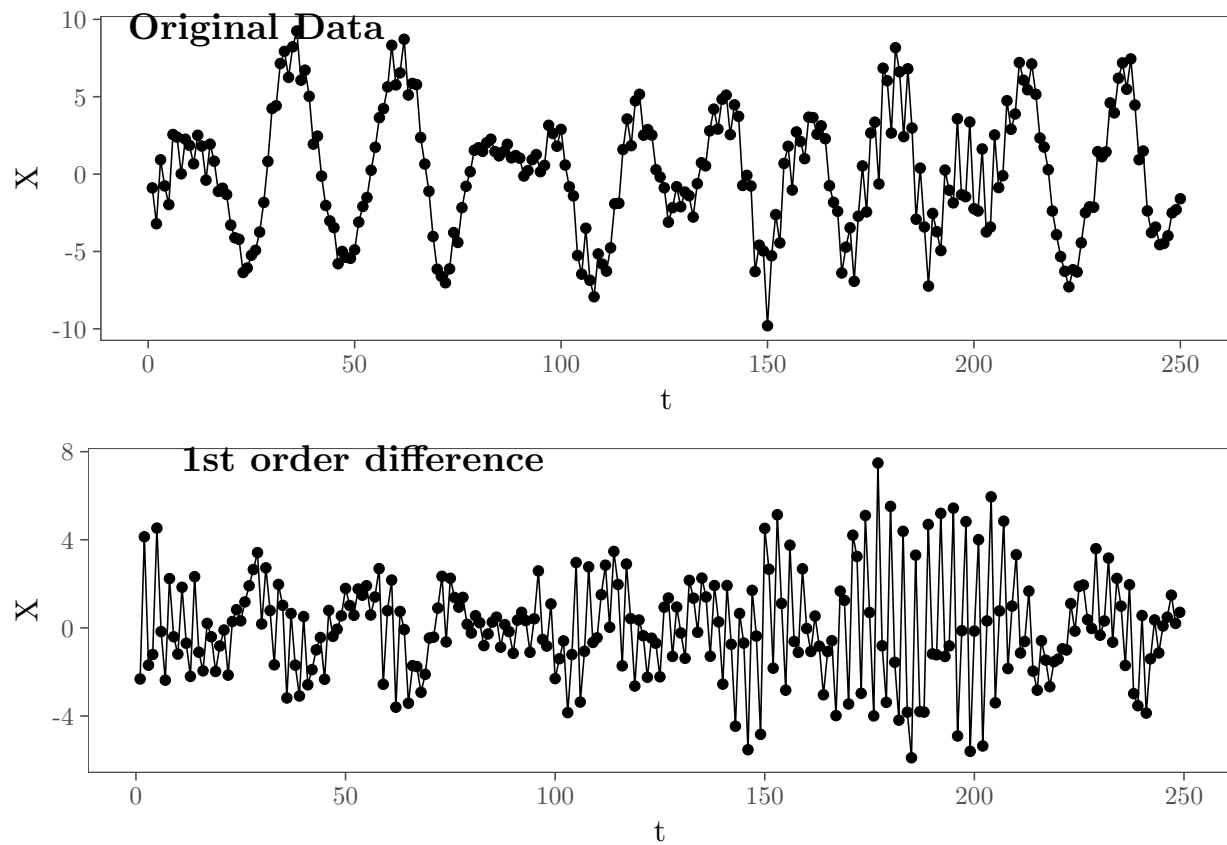
### 9.3.1 An example in R

```
mafun <- function(xs, n) {
  stats::filter(xs, rep(1, n))/n
}
dfun <- function(xs, n) {
  diff(xs, lag = n)
}
th <- ggthemes::theme_few()
library(tswge)
data(fig1.21a)
ma <- mafun(fig1.21a, 5)
d <- dfun(fig1.21a, 1)
fp <- tplot(fig1.21a) + th
mp <- tplot(ma) + th
dp <- tplot(d) + th
```

And here lets view the 5 point moving average

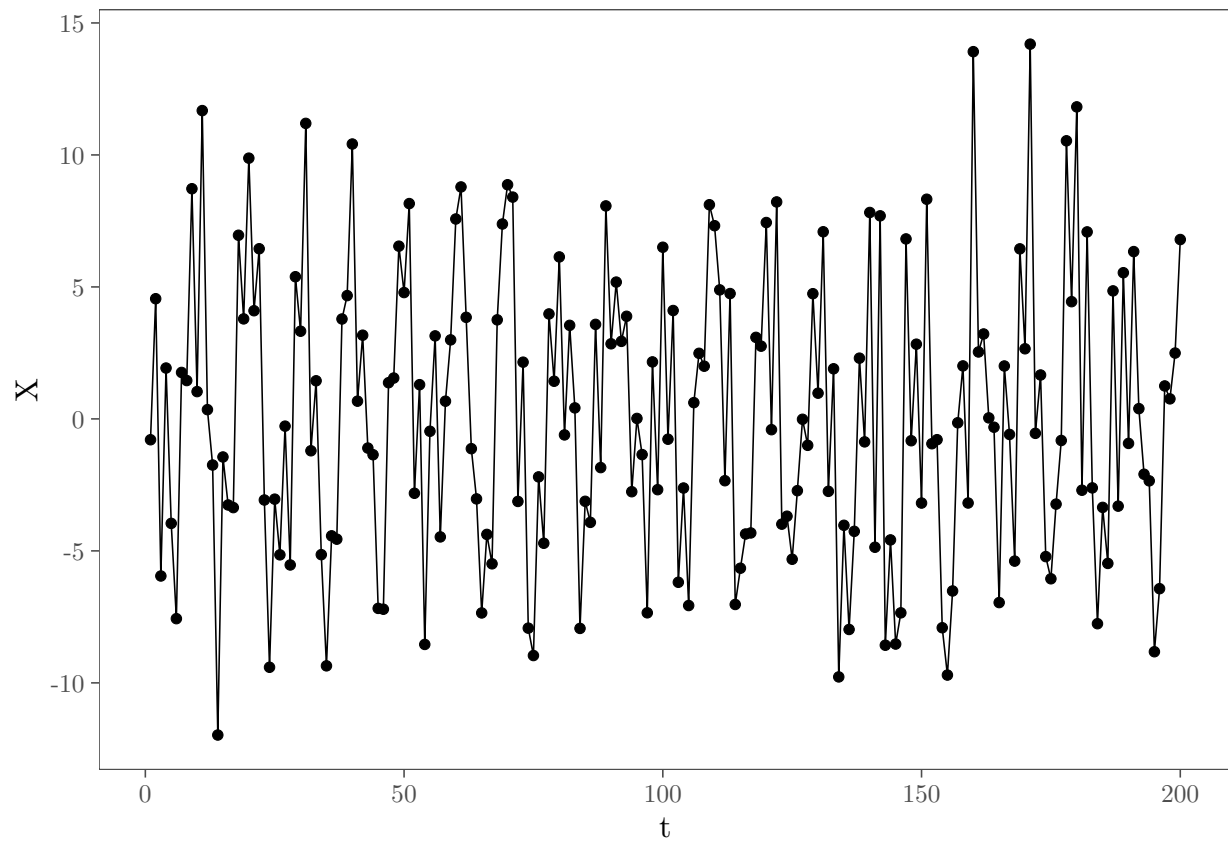


And now a look at the difference filter

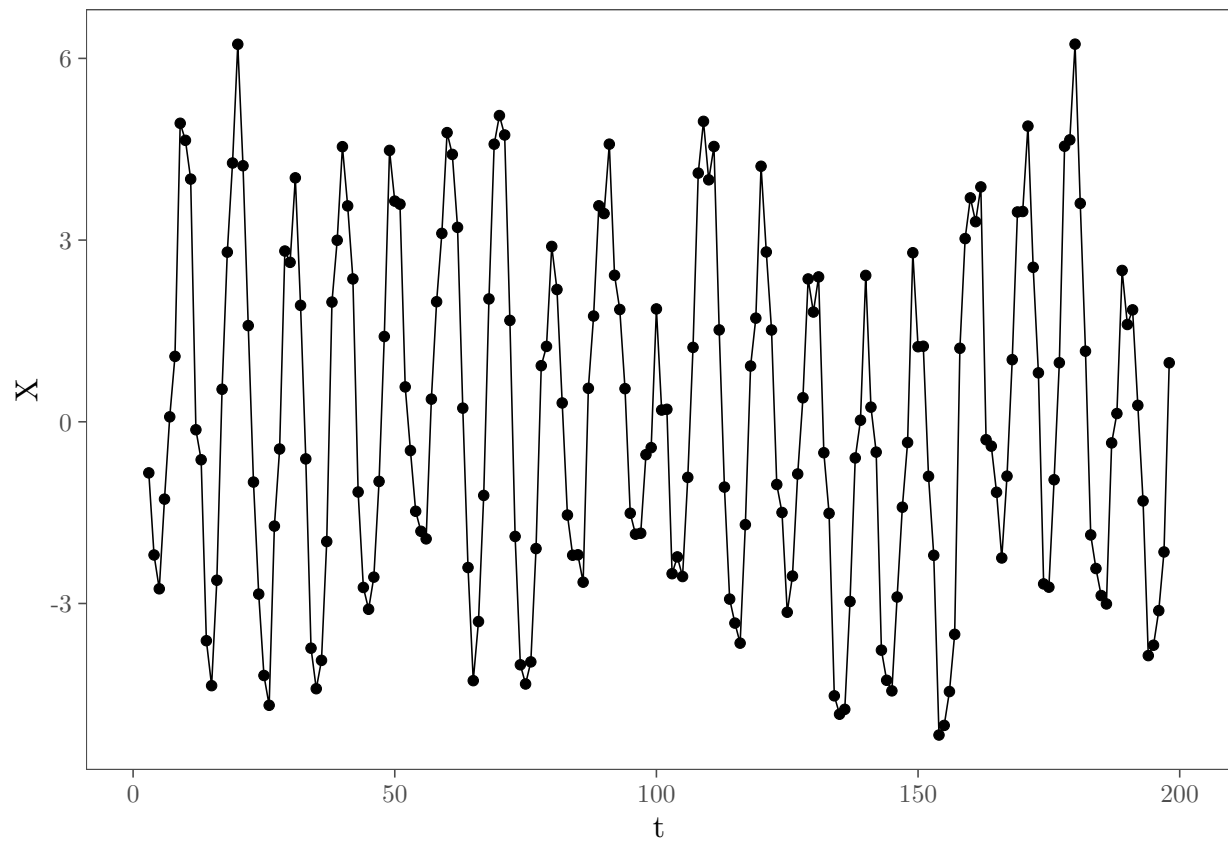


```
rlz <- gen.sigplusnoise.wge(200, coef = c(5, 3), freq = c(0.1, 0.45), vara = 10,
  sn = 1, plot = F)
```

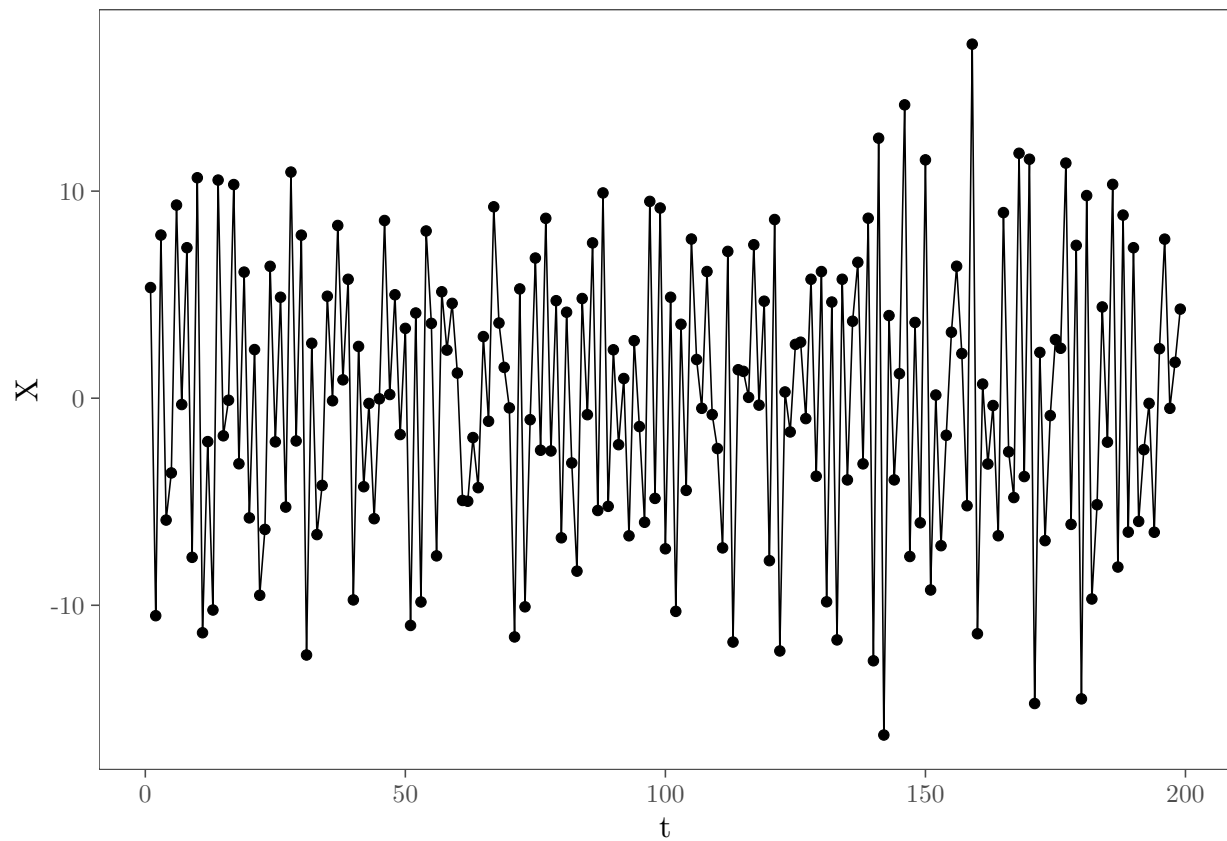
```
pfun <- function(x, n, l) {
  fp <- tplot(x) + th
  mp <- tplot(mafun(x, n)) + th
  dp <- tplot(dfun(x, l)) + th
  list(original = fp, MA = mp, dif = dp)
}
plts <- pfun(rlz, 5, 1)
plts$original
```



```
plots$MA
```

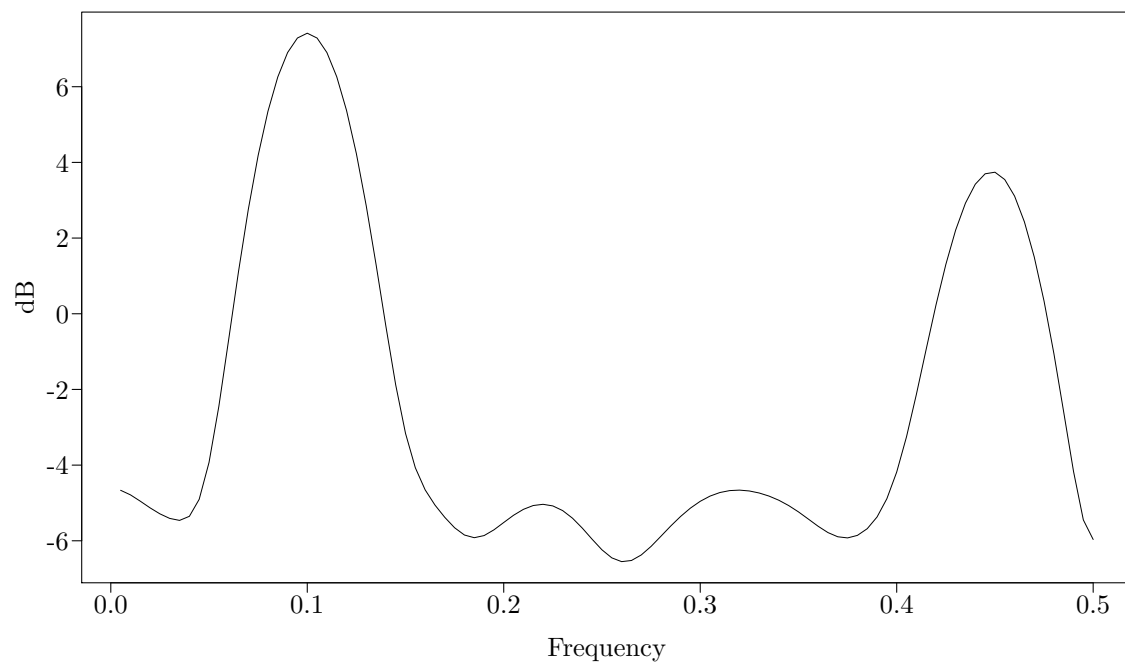


```
plots$dif
```



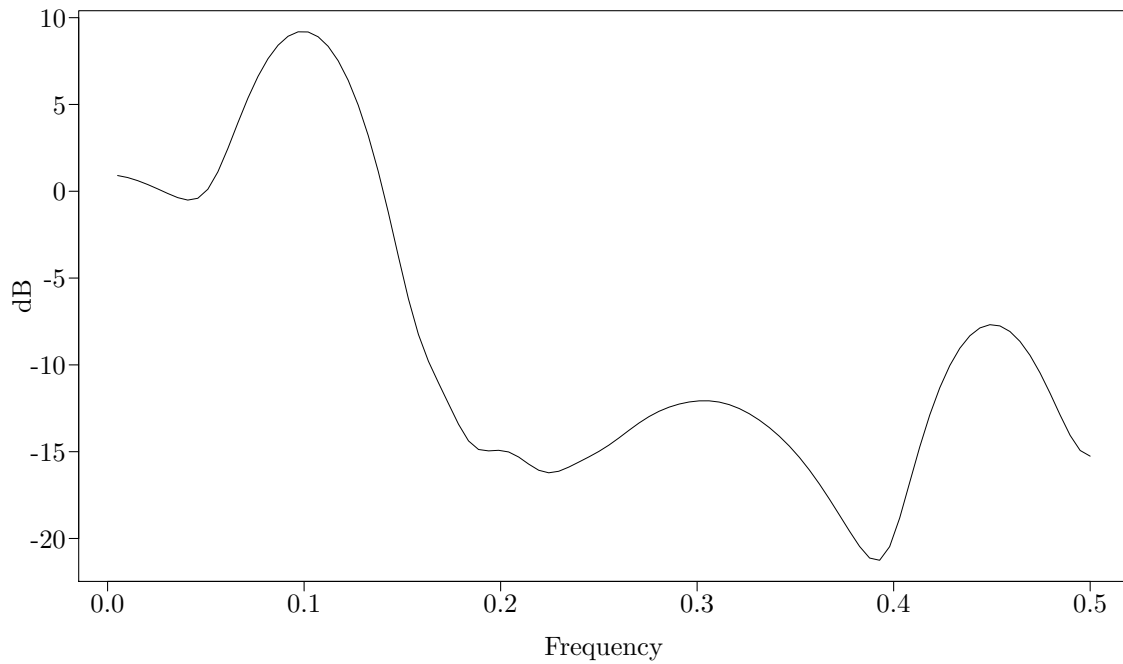
```
parzen.wge(rlz)
```

**Parzen Window Truncation point:  $M = 28$**



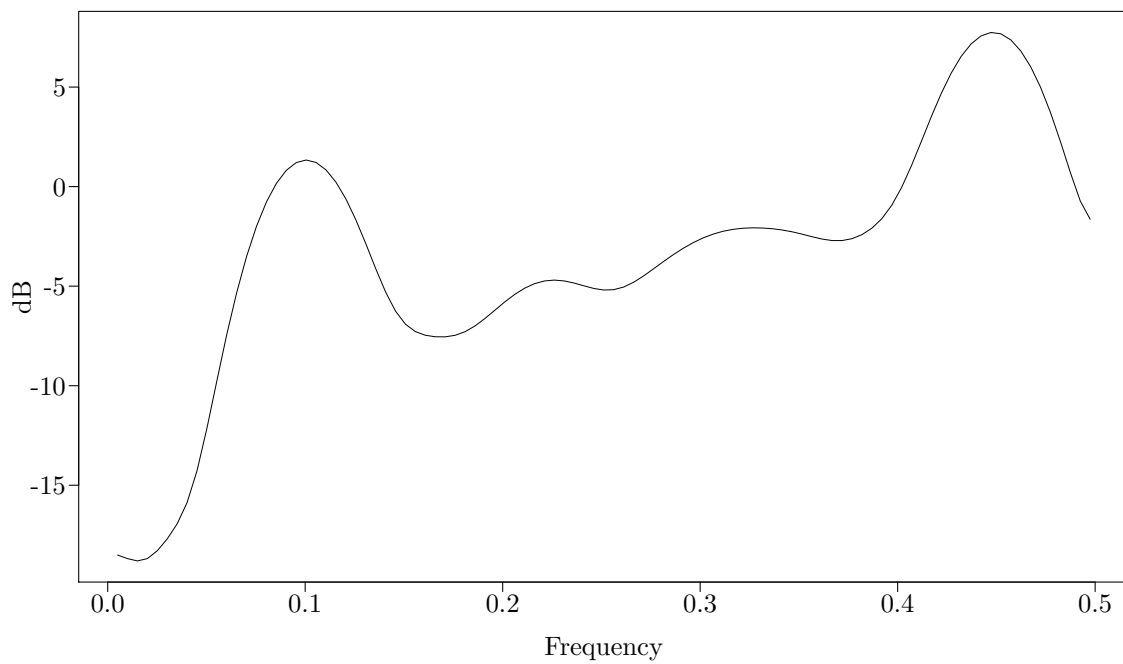
```
parzen.wge(na.omit(mafun(rlz, 5)))
```

**Parzen Window Truncation point:  $M = 28$**



```
parzen.wge(na.omit(dfun(rlz, 1)))
```

**Parzen Window Truncation point:  $M = 28$**





## 9.4 GLPs

a GLP is a linear filter with a *white noise input*

We take white noise input and put it through a filter and get something that isn't white noise.

$$\sum_{j=0}^{\infty} \psi_j a_{t-j} = X_t - \mu$$

GLPs are an infinite sum of white noise terms. This might be weird now but later on this concept will return.

The  $\psi_j$ 's are called psi weights, these will be useful again later. AR, MA, and ARMA are all special cases of GLPs, and will be useful when we study confidence intervals.

## 9.5 AR(1) Intro

AR(p) in which  $p = 1$

We will go through well known forms of the AR(1) model.

$$X_t = \beta + \phi_1 X_{t-1} + a_t$$

$$\beta = (1 - \phi_1)\mu$$

Beta is moving average constant

We are saying that the value of  $x$  depends on some constant, the previous value of  $x$ , and some random white noise. Looks a lot like regression except there is a weird variable

we can rewrite this as

$$X_t = (1 - \phi_1)\mu + \phi_1 X_{t-1} + a_t$$

An AR(1) process is stationary iff the magnitude of  $\phi_1$  is less than 1

We will deal with this a lot in the near future

### 9.5.1 AR(1) math zone

$$E[X_t] = \mu?$$

$$E[X_t] = E[(1 - \phi_1)\mu] + E[\phi_1 X_{t-1}] + E[a_t]$$

We can rewrite this as

$$E[X_t] = (1 - \phi_1)\mu + \phi_1 E[X_t] + 0$$

$$E[X_t](1 - \phi_1) = (1 - \phi_1)\mu$$

$$E[X_t] = \mu$$

Mean does not depend on  $T$

The variance also does not, and if  $|\phi_1|$  is less than one variance is finite

$$\sigma_X^2 = \frac{\sigma_a^2}{1 - \phi_1^2}$$

and  $\rho_k$  is  $\phi_1^k$  to the k

$$\rho_k = \phi_1^k, k \geq 0$$

Spectral Density of AR(1) also does not depend on time, it just monotonically increases or decreases depending on  $\phi_1$ :

$$S_X(f) = \frac{\sigma_a^2}{\sigma_X^2} \left( \frac{1}{|1 - \phi_1 e^{-2\pi i f}|^2} \right)$$

### 9.5.2 the zero mean form of AR1

with zero mean,

$$X_t = \phi_1 X_{t-1} + a_t$$

OR

$$X_t - \phi_1 X_{t-1}$$

A

### 9.5.3 AR1 with positive phi

remember we have

$$\rho_k = \phi_1^k$$

With positive  $\phi_1$ , we have:

*realizations* are wandering and aperiodic

*Autocorrelations* are damped exponentials

*Spectral density* Peaks at zero.

Higher values = stronger versions of these characteristics

### 9.5.4 AR1 with negative phi

We have:

*realization* are oscillating

*Autocorrelations* are damped and alternating (negative to a power you fool)

*Spectral density*\* has a peak at  $f = 0.5$  (cycle length of 2)

Higher magnitude = stronger characteristics

### 9.5.5 Nonstationary

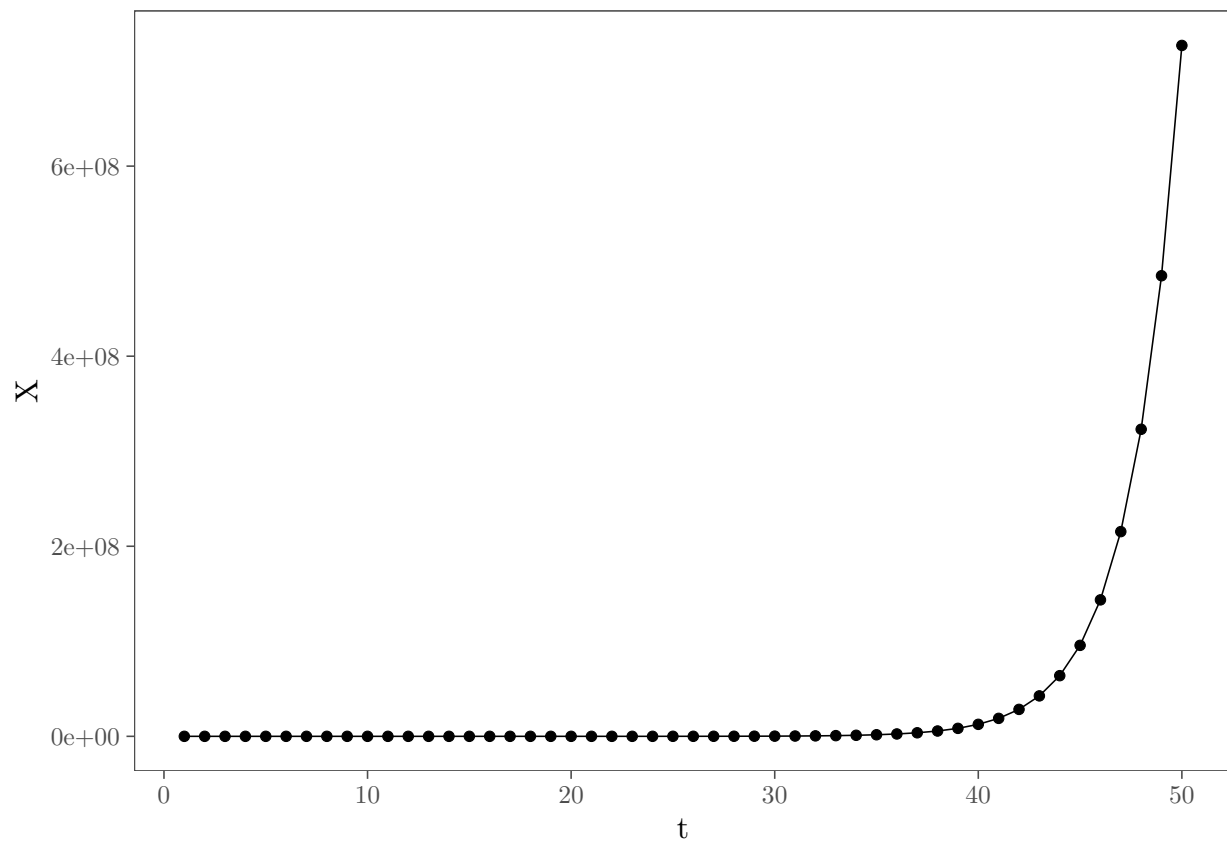
if  $\phi_1 = 1$  or  $\phi_1 > 1$ , we have that realizations are from a nonstationary process. with it equal to one, it looks ok, and is actually a special ARIMA model. With it  $> 1$ , we have crazy explosive realizations. Check this out:

```

nonstarma <- function(n, phi) {
  x <- rep(0, n)
  a <- rnorm(n)
  x[1:n] <- 0
  for (k in 2:n) {
    x[k] = phi * x[k - 1] + a[k]
  }
  tplot(x)
}

nonstarma(n = 50, phi = 1.5) + th

```





## Chapter 10

# Characteristic Equations for AR(1) Models

Consider the zero-mean form of an AR(1) model

$$X_t - \phi_1 X_{t-1} = a_t$$

We can rewrite in operator form as

$$(1 - \phi_1 B)X_t = a_t$$

Where B is an operator that turns  $X_t$  into  $X_{t-1}$ . We can then rewrite this algebraically (for solving purposes) as a characteristic equation:

$$1 - \phi_1 Z = 0$$

Now recall that AR(1) is stationary *iff*  $|\phi_1|$  is less than 1. Lets solve the characteristic equation for the root now:

$$r = z = \frac{1}{\phi_1}$$

If  $X_t$  is stationary,  $|r|$  is greater than one. This does not feel important now, but when we get to higher order time series this will save us a lot of thinking. Lets look at a numerical example just to make sure we got it:

$$X_t = -1.2X_{t-1} + a_t$$

$$X_t + 1.2X_{t-1} = a_t$$

$$(1 + 1.2B)X_t = a_t$$

$$1 + 1.2Z = 0$$

$$r = z = -\frac{1}{1.2} = -0.8333333$$

Note that in this case,  $\phi_1 = -1.2$ . This is a weird thing with this notation, get used to it.



## Chapter 11

### AR(2) models

$$X_t = \beta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + a_t$$

where  $\beta = (1 - \phi_1 - \phi_2)\mu$  is the moving average constant.

#### 11.1 Notes

It looks like MLR, however the independent variables are values of the dependent variable at two previous time periods

#### 11.2 Stationarity in AR(2)

$$E[X_t] = \mu$$

$$\sigma_X^2 = \frac{\sigma_a^2}{1 - \phi_1 \rho_1 - \phi_2 \rho_2}$$

Instead of depending on positive or negative roots, like AR(1), the behavior of the autocorrelations and spectral density depends on whether or not the roots of the characteristic equation are complex. This means it is already hard to tell from the equation whether or not it is stationary just by looking at the equation.

#### 11.3 AR(2) zero mean form

If  $\mu = 0$ , then the AR(2) model takes the form :

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + a_t$$

We can study this much easier and then add the mean back later. A more common way of writing this, and more useful to us is the form:

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} = a_t$$

#### 11.4 More on backshift operator notation

Remember, we defined the backshift operator as  $B$  such that  $BX_t = X_{t-1}$

We can extend this and say:

$$B^k X_t = X_{t-k}$$

This means we can rewrite:

$$X_t - \phi_1 B X_t - \phi_2 B^2 X_t = a_t$$

$$X_t [1 - \phi_1 B - \phi_2 B^2] = a_t$$

or, more tersely:

$$\phi(B) X_t = a_t$$

Where  $\phi(B) = 1 - \phi_1 B - \phi_2 B^2$

We can rewrite this  $\phi(B)$  as a characteristic equation now:

$$1 - \phi_1 Z - \phi_2 Z^2 = 0$$

Lets quickly expand on our quadratic solver to get the roots of an AR(2) model:

```
ar2solver <- function(p1, p2) {
  res <- as.complex(quad_form(p2, p1, 1))
  mag <- Mod(res)
  absrecip <- 1/mag
  data.frame(roots = res, magnitude = mag, abs.recip = absrecip)
}
ar2solver(0.4, 0.8)
```

```
##           roots magnitude abs.recip
## 1 -0.25-1.09i  1.118302 0.8942126
## 2 -0.25+1.09i  1.118302 0.8942126
```

## 11.5 Characteristic equation

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$1 - \phi_1 Z - \phi_2 Z^2 = 0$$

Just solve the quadratic equation using the quadratic formula:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Two cases: \* Two real roots \* Two complex roots

## 11.6 Key result

An AR(2) is stationary *iff* all roots are outside the unit circle



### 11.6.1 Two real roots

We can factor the AR(2) into two AR(1)s

- one part positive one part negative
  - Wandering but oscillating
  - Damping ACF but slightly oscillatory
  - S(f) has a peak at 0 and at 0.5
- Two Positive roots
  - Super wandering
  - Super Damped
  - S(f) peak at 0
- Two Negative
  - super oscillating

### 11.6.2 Complex conjugate roots

Stationary AR(2)'s with complex conjugate roots have an autocorrelation function which looks more like a damped sinusoid instead of a damped exponential, and a system frequency,  $f_0$  defined as+

$$f_0 = \frac{1}{2\pi} \cos^{-1} \left( \frac{\phi_1}{2\sqrt{-\phi_2}} \right)$$

. Pretty easy stuff.



## Chapter 12

### AR(p)

This is basically the same thing, with the same conclusion, except instead of having clearly defined behavior based on the sine of the roots, the closer the absolute value of the roots is to one the more dominant that behavior is in the realization. To solve these, we will use this brilliant `factor.wge` function:

```
library(tswge)
factor.wge(phi = c(0.7, 0.4, 0.3, -0.4, -0.7, 0.1, -0.1))
```

```
##
## Coefficients of Original polynomial:
## 0.7000 0.4000 0.3000 -0.4000 -0.7000 0.1000 -0.1000
##
## Factor          Roots          Abs Recip    System Freq
## 1-2.0903B+1.2507B^2  0.8357+-0.3182i    1.1183    0.0579
## 1+0.7598B+0.8037B^2 -0.4727+-1.0104i    0.8965    0.3196
## 1+0.8141B          -1.2283      0.8141    0.5000
## 1-0.1836B+0.1222B^2  0.7512+-2.7602i    0.3496    0.2077
##
##
```

We can use the absolute reciporacal (speling) of the root to tell if it is stationary easily. If it is less than one, we have stationarity. It has the same property in that if it is closer to one, that is a stronger characteristic in the realization.



## Chapter 13

# MA(q) Models

### 13.1 Properties and Characteristics

We use MA models to model stationary data. They are not quite as useful as AR models, but in conjunction with them, we can create ARMA(p,q) models and so forth.

#### 13.1.1 A quote for your thoughts:

“All models are wrong... but some are useful” – George Box

### 13.2 MA(q)

First of all, we need to define the equation for a Moving Average Model of order q:

$$X_t = \mu + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q}$$