# Time Series: A First Course with Bootstrap Starter

## Contents

# Lesson 9-1: Weak Dependence

- We can quantify serial dependence through the ACVF.
- Recall entropy mixing (Paradigm 8.2.11) as another measure of serial dependence.

### Example 9.1.1. Slow Polynomial Decay – Long Memory

- Suppose that a time series has ACVF $\gamma(h) = O(|h|^{-a})$ for $0 < a \leq 1$.
- The notation means the ACVF is bounded by a constant times that rate.
- There is fairly slow decay in lag when $a$ is small: this means persistence, or high association, across large time lags. This is called *long-range dependence* or *long memory*.

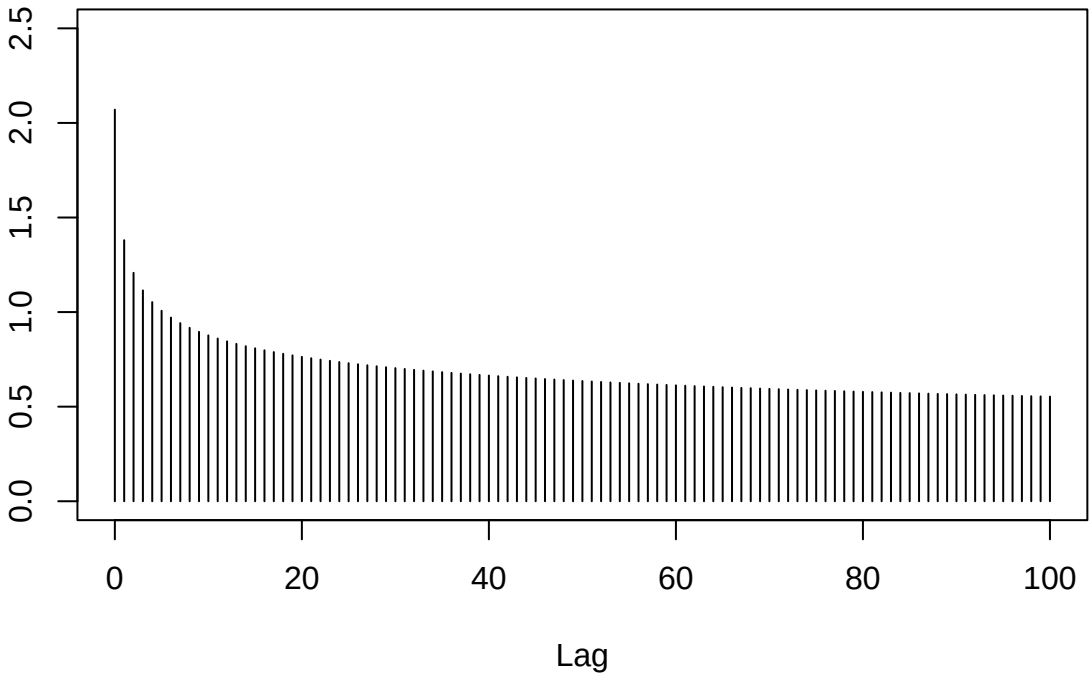
### Exercise 9.2. Long-Range Dependence

- An example of long-range dependence is given by the ACVF recursively defined via

$$\gamma(h+1) = \frac{h+d}{h+1-d}\,\gamma(h)$$

  for $d \in (0, 1/2)$, where $\Gamma$ denotes the gamma function, and $\gamma(0) = \Gamma(1-2d)/\Gamma(1-d)^2$.
- This corresponds to $\gamma(h) = O(|h|^{2d-1})$, or the case of long memory with decay rate $a = 1 - 2d$.
- We plot with $d = .4$.

```
lags <- seq(1,100)
d <- .4
gamma.lm <- gamma(1-2*d)/(gamma(1-d))^2
for(i in 1:length(lags))
{
  gamma.new <- gamma.lm[i] * (i-1+d)/(i-d)
  gamma.lm <- c(gamma.lm,gamma.new)
}
plot(ts(gamma.lm,start=0),type="h",xlab="Lag",ylab="",ylim=c(0,2.5))
```




### Example 9.1.3. Geometric Decay

- Suppose that a time series has ACVF $\gamma(h) = O(r^{|h|})$ for $0 < r < 1$.
- The notation means the ACVF is bounded by a constant times that rate.
- So there is a *geometric*, or *exponential*, rate of decay for the ACVF.
- This holds for ARMA processes by Proposition 5.8.3.

### Example 9.1.4. MA($q$) Correlation

- Suppose the a time series has ACVF $\gamma(h)$ that is zero if $|h| > q$.
- Then this can be represented as an MA($q$) process.
- This is an example of a *q-dependent* process: $X_t$ and $X_{t-h}$ are independent for any $t$ when $|h| > q$.

## Lesson 9-2: Sample Mean

- We need to know the statistical properties of common time series estimators.
- We here focus on the sample mean, which estimates the population mean of a stationary time series.
- The sample mean is

$$\overline{X} = n^{-1} \sum_{t=1}^{n} X_t.$$

### Remark 9.2.1. The Long-Run Variance

- Suppose $\{X_t\}$ is stationary and the autocovariance function (ACVF) is absolutely summable:

$$\sum_{h=-\infty}^{\infty} |\gamma(h)| < \infty.$$

- Then the *long-run variance* is defined as

$$\sum_{h=-\infty}^{\infty} \gamma(h).$$

- This is denoted by $\sigma^2_\infty$. Observe this is equal to $G(1)$, i.e., setting $z = 1$ in the AGF.
- The long-run variance can be zero: suppose $\{X_t\}$ has a MA representation $X_t = \psi(B)Z_t$, so that

$$\sigma^2_\infty = G(1) = \psi(1)^2\sigma^2.$$

  Then if $\psi(1) = 0$, the long-run variance is zero. For example, consider an MA(1) process with $\theta(z) = 1 - z$.

### Proposition 9.2.2.

Suppose $\{X_t\}$ is stationary with mean $\mu$, and ACVF that is absolutely summable. Then $\overline{X}$ is unbiased for $\mu$, and its scaled variance tends to the long-run variance:

$$\text{Var}[\sqrt{n}\overline{X}] \to \sigma^2_\infty.$$

### Theorem 9.2.7.

- So long as the long-run variance is non-zero, we can establish a central limit theorem for the sample mean under the condition that the inputs (in the MA representation) are i.i.d.
- Suppose $\{X_t\}$ has causal MA representation

$$X_t = \mu + \psi(B)Z_t,$$

  where $Z_t \sim i.i.d.(0, \sigma^2)$. Suppose $\sum_{j\geq 0} j|\psi_j| < \infty$ and $\psi(1) \neq 0$. Then

$$\sqrt{n}(\overline{X} - \mu) \Rightarrow \mathcal{N}(0, \sigma^2_\infty).$$

### Remark 9.2.8. Inference for the Mean

As an application, suppose we want to construct a confidence interval for the mean based on our estimator $\overline{X}$. To get the standard error, we need to know the long-run variance. This suggests the need to estimate all the autocovariances (however, that is not possible).

### Exercise 9.13. Simulating an AR(1) Limiting Variance

- We simulate a Gaussian AR(1) process with parameter $\phi_1 = .8$ and $\sigma^2 = 1$.

```
armapq.sim <- function(n,burn,ar.coefs,ma.coefs,innovar)
{
    p <- length(ar.coefs)
    q <- length(ma.coefs)
    z <- rnorm(n+burn+p+q,sd=sqrt(innovar))
    x <- filter(z,c(1,ma.coefs),method="convolution",sides=1)
    x <- x[(q+1):(q+n+burn+p)]
    y <- x[1:p]
```

```
    for(t in (p+1):(p+n+burn))
    {
        next.y <- sum(ar.coefs*y[(t-1):(t-p)]) + x[t]
        y <- c(y,next.y)
    }
    y <- y[(p+burn+1):(p+burn+n)]
    return(y)
}
phi <- .8
```

- Based on sample sizes $n = 50, 100, 200$, we compute the sample mean.
- Repeating over $10,000$ simulations, we approximate the scaled variance of the sample mean, and see how close this is to the long-run variance.

```
# small sample
n <- 50
x.means <- NULL
for(i in 1:10000)
{
    x.sim <- armapq.sim(n,500,phi,NULL,1)
    x.means <- c(x.means,mean(x.sim))
}
y <- sqrt(n)*x.means
var(y)
```

```
## [1] 22.99445
```

```
# long-run variance for an AR(1)
(1-phi)^{-2}
```

```
## [1] 25
```

```
# moderate sample
n <- 100
x.means <- NULL
for(i in 1:10000)
{
    x.sim <- armapq.sim(n,500,phi,NULL,1)
    x.means <- c(x.means,mean(x.sim))
}
y <- sqrt(n)*x.means
var(y)
```

```
## [1] 23.15012
```

```
# long-run variance for an AR(1)
(1-phi)^{-2}
```

```
## [1] 25
```

```
# large sample
n <- 200
x.means <- NULL
for(i in 1:10000)
{
    x.sim <- armapq.sim(n,500,phi,NULL,1)
    x.means <- c(x.means,mean(x.sim))
```

```
}
y <- sqrt(n)*x.means
var(y)
```

```
## [1] 25.02043
```

```
# long-run variance for an AR(1)
(1-phi)^{-2}
```

```
## [1] 25
```

# Lesson 9-4: Serial Correlation

We want to estimate the ACVF: they are used in prediction, and are helpful in understanding serial dependence in the time series.

### Remark 9.4.1. ACVF Estimator for Known Mean

- Suppose that the mean $\mu$ of the stationary time series $\{X_t\}$ was known, and we want to estimate $\gamma(h)$.
- This is the expectation of $Y_t$, where

$$Y_t = (X_{t+h} - \mu)(X_t - \mu).$$

- We can compute $Y_1, \ldots, Y_{n-h}$ for any $0 \leq h < n$, because $\mu$ is known.
- The sample mean of these would be

$$\frac{1}{n-h}\sum_{t=1}^{n-h} Y_t = \frac{1}{n-h}\sum_{t=1}^{n-h}(X_{t+h} - \mu)(X_t - \mu).$$

  This has expectation $\gamma(h)$.
- The long-run variance of this sample mean is based on the ACVF of $\{Y_t\}$, and has a complicated expression known as *Bartlett's Formula*.
- This suggests an estimator

$$\overline{\gamma}(h) = \frac{1}{n-h}\sum_{t=1}^{n-h}(X_{t+h} - \mu)(X_t - \mu),$$

  which is unbiased for $\gamma(h)$.

### Proposition 9.4.3. Bartlett's Formula

Suppose $\{X_t\}$ is a causal linear time series with i.i.d. inputs $\{Z_t\}$, which have variance $\sigma^2$ and kurtosis $\eta$. Then the long-run variance of $\overline{\gamma}(h)$ is

$$\tau_\infty^2 = \sum_{k=-\infty}^{\infty}\left(\gamma(k+h)\gamma(k-h) + \gamma(k)^2\right) + \gamma(h)^2(\eta - 3).$$

### Remark 9.4.4. Limiting Variance of the ACVF Estimator

- Recall that the Fourier coefficients of a spectral density are denoted

$$\langle f \rangle_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\lambda)e^{i\lambda k}\, d\lambda.$$

- For a stationary Gaussian process with spectral density $f$, then Bartlett's Formula is

$$\tau_\infty^2 = \langle f^2 \rangle_{2k} + \langle f^2 \rangle_0.$$

- Example 1: suppose $k = 1$ and the process is a white noise (with variance $\sigma^2$). Then

$$\tau_\infty^2 = \langle \sigma^4 \rangle_2 + \langle \sigma^4 \rangle_0 = \sigma^4.$$

- Example 2: suppose $k = 1$ and the process is a MA(1). So $f(\lambda) = \sigma^2 |1 + \theta e^{-i\lambda}|^2$, and

$$\tau_\infty^2 = \sigma^4 \langle |1 + \theta e^{-i\lambda}|^4 \rangle_2 + \sigma^4 \langle |1 + \theta e^{-i\lambda}|^2 \rangle_0 = \sigma^4 \left(1 + 5\theta^2 + \theta^4\right).$$

### Remark 9.4.6. ACVF Estimator for Unknown Mean

- Since $\mu$ is usually unknown, we can replace it by the sample mean $\overline{X}$.
- The resulting estimator is denoted $\widetilde{\gamma}(h)$.
- This estimator is asymptotically normal when the process $\{X_t\}$ is $m$-dependent (i.e., random variables of at least lag $m$ between them are independent).

## Lesson 9-5: Sample Autocovariance

- We renormalize the ACVF estimator via tapering.

### Remark 9.5.2. Tapering the ACVF to Reduce Variability in High Lags

- The estimator $\widetilde{\gamma}(h)$ is approximately unbiased, but has high variance for large $h$.
- We can replace the $n - h$ divisor by $n$, in order to decrease variance for larger $h$.
- This is like multiplying $\widetilde{\gamma}(h)$ by $1 - h/n$, which is an example of a taper.

### Definition 9.5.3.

- An autocovariance *taper* is a bounded, even real-valued function $\Lambda$ on $[-1, 1]$, such that $\Lambda(0) = 1$ and $\Lambda(x) \leq 1$.
- We multiply the ACVF estimator at lag $h$ by $\Lambda(h/n)$, where $\Lambda(x) = 1 - |x|$.

### Example 9.5.4. Bartlett Taper

- The Bartlett taper is $\Lambda(x) = 1 - |x|$.
- The resulting estimator is

$$\widehat{\gamma}(h) = \Lambda(h/n)\widetilde{\gamma}(h) = \frac{1}{n} \sum_{t=1}^{n-|h|} (X_{t+|h|} - \overline{X})(X_t - \overline{X}).$$

- This is called the *sample autocovariance function* (sample ACVF)
- Under $m$-dependence, Corollary 9.5.8 establishes a central limit theorem for the sample ACVF:

$$\sqrt{n}(\widehat{\gamma}(h) - \gamma(h)) \Rightarrow \mathcal{N}(0, \tau_\infty^2).$$

### Exercise 9.28. Sample ACVF of ARMA(1,2) Process

- We load some functions needed to compute the true ACVF.

```
polymult <- function(a,b) {
bb <- c(b,rep(0,length(a)-1))
B <- toeplitz(bb)
B[lower.tri(B)] <- 0
aa <- rev(c(a,rep(0,length(b)-1)))
prod <- B %*% matrix(aa,length(aa),1)
return(rev(prod[,1]))
```

```
}

ARMAauto <- function(phi,theta,maxlag)
{
    p <- length(phi)
    q <- length(theta)
    gamMA <- polymult(c(1,theta),rev(c(1,theta)))
    gamMA <- gamMA[(q+1):(2*q+1)]
    if (p > 0)
    {
        Amat <- matrix(0,nrow=(p+1),ncol=(2*p+1))
        for(i in 1:(p+1))
        {
            Amat[i,i:(i+p)] <- c(-1*rev(phi),1)
        }
        Amat <- cbind(Amat[,(p+1)],as.matrix(Amat[,(p+2):(2*p+1)]) +
            t(matrix(apply(t(matrix(Amat[,1:p],p+1,p)),2,rev),p,p+1)))
        Bmat <- matrix(0,nrow=(q+1),ncol=(p+q+1))
        for(i in 1:(q+1))
        {
            Bmat[i,i:(i+p)] <- c(-1*rev(phi),1)
        }
        Bmat <- t(matrix(apply(t(Bmat),2,rev),p+q+1,q+1))
        Bmat <- matrix(apply(Bmat,2,rev),q+1,p+q+1)
        Bmat <- Bmat[,1:(q+1)]
        Binv <- solve(Bmat)
        gamMix <- Binv %*% gamMA
        if (p <= q) { gamMix <- matrix(gamMix[1:(p+1),],p+1,1)
            } else gamMix <- matrix(c(gamMix,rep(0,(p-q))),p+1,1)
        gamARMA <- solve(Amat) %*% gamMix
    } else gamARMA <- gamMA[1]

    gamMA <- as.vector(gamMA)
    if (maxlag <= q) gamMA <- gamMA[1:(maxlag+1)] else gamMA <- c(gamMA,rep(0,(maxlag-q)))
    gamARMA <- as.vector(gamARMA)
    if (maxlag <= p) gamARMA <- gamARMA[1:(maxlag+1)] else {
    for(k in 1:(maxlag-p))
    {
        len <- length(gamARMA)
        acf <- gamMA[p+1+k]
        if (p > 0) acf <- acf + sum(phi*rev(gamARMA[(len-p+1):len]))
        gamARMA <- c(gamARMA,acf)
    } }
    return(gamARMA)
}
```

- We simulate a Gaussian ARMA(1,2) process of length $n = 200$.

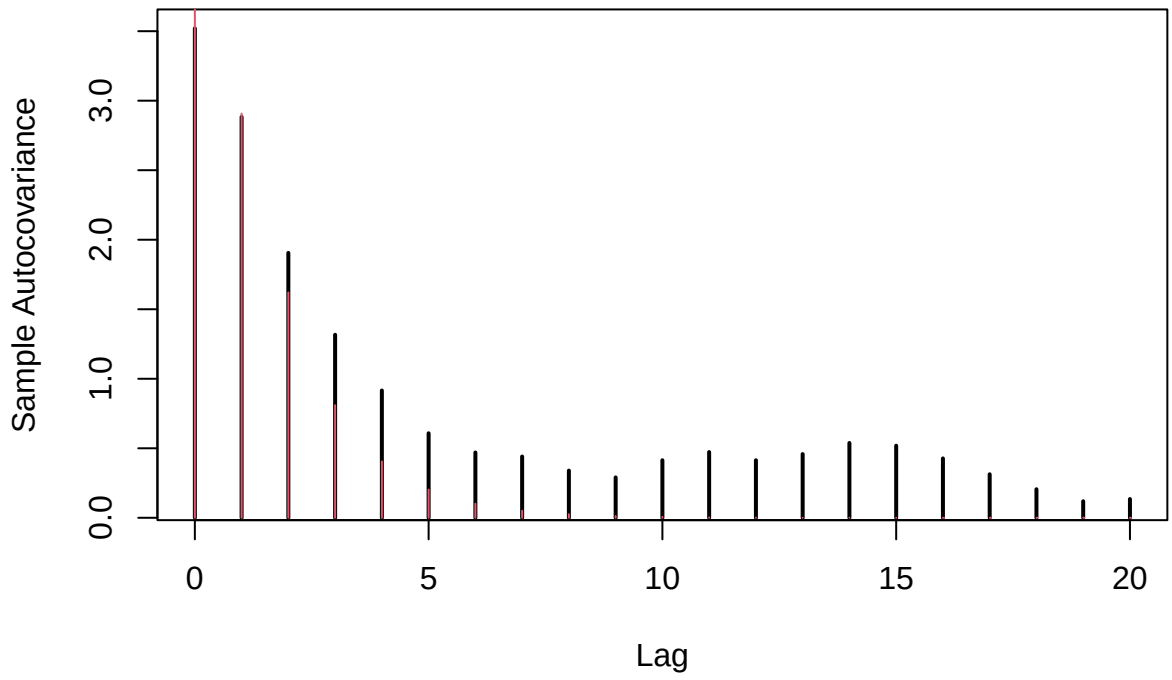
```
armapq.sim <- function(n,burn,ar.coefs,ma.coefs,innovar)
{
    p <- length(ar.coefs)
    q <- length(ma.coefs)
    z <- rnorm(n+burn+p+q,sd=sqrt(innovar))
    x <- filter(z,c(1,ma.coefs),method="convolution",sides=1)
```

```
    x <- x[(q+1):(q+n+burn+p)]
    y <- x[1:p]
    for(t in (p+1):(p+n+burn))
    {
        next.y <- sum(ar.coefs*y[(t-1):(t-p)]) + x[t]
        y <- c(y,next.y)
    }
    y <- y[(p+burn+1):(p+burn+n)]
    return(y)
}

n <- 200
phi1 <- .5
theta1 <- 5/6
theta2 <- 1/6
x.sim <- armapq.sim(n,500,phi1,c(theta1,theta2),1)
```

- We construct and plot the sample acvf.
- We also overlay the true acvf in red.

```
y.sim <- x.sim - mean(x.sim)
x.acf <- mean(y.sim^2)
for(h in 1:20)
{
    x.acf <- c(x.acf,sum(y.sim[1:(n-h)]*y.sim[(h+1):n])/n)
}
gamma <- ARMAauto(phi1,c(theta1,theta2),21)
plot(ts(x.acf,start=0,frequency=1),xlab="Lag",ylab="Sample Autocovariance",type="h",lwd=2)
lines(ts(gamma,start=0,frequency=1),type="h",col=2)
```




# Lesson 9-6: Spectral Means

- We investigate statistics that are weighted sums of the periodogram.

## Proposition 9.6.1.

- The Fourier coefficients of the periodogram are the sample ACVF:

$$I(\lambda) = \sum_{k=-\infty}^{\infty} \widehat{\gamma}(k) e^{-i\lambda k},$$

  and $\langle I \rangle_k = \widehat{\gamma}(k)$.
- So the sample ACVF is positive definite.

## Definition 9.6.2.

A *spectral mean* is a functional of the spectral density $f$, of the form

$$\langle gf \rangle_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} g(\lambda) f(\lambda)\, d\lambda,$$

where $g$ is a real-valued weighting function.

## Remark 9.6.3. Spectral Mean Estimation

A spectral mean can be estimated by substituting the periodogram:

$$\langle gI \rangle_0 = \sum_{|h|<n} \langle g \rangle_h \widehat{\gamma}(h).$$

### Theorem 9.6.6.

Suppose $\{X_t\}$ is a linear time series with mean $\mu$ and causal moving average representation, with inputs that have variance $\sigma^2$ and kurtosis $\eta$ (fourth moment divided by the square of the second moment). For sufficiently smooth $g$,

$$\sqrt{n}\left(\langle gI\rangle_0 - \langle gf\rangle_0\right) \Rightarrow \mathcal{N}\left(0, \langle g(g+g^\sharp)f^2\rangle_0 + (\eta-3)\langle gf\rangle_0^2\right),$$

where $g^\sharp(\lambda) = g(-\lambda)$.

### Remark 9.6.7. Autocovariance Limiting Variance

- If $g(\lambda) = \cos(\lambda k)$, then the spectral mean is $\gamma(k)$, and the estimate is $\widehat{\gamma}(k)$.
- The Bartlett Formula (Proposition 9.4.3) follows from Theorem 9.6.6.

### Corollary 9.6.9. Ratio Statistics

Under same assumptions as Theorem 9.6.6, with smooth weighting functions $a$ and $b$,

$$\sqrt{n}\left(\frac{\langle bI\rangle_0}{\langle aI\rangle_0} - \frac{\langle bf\rangle_0}{\langle af\rangle_0}\right) \Rightarrow \mathcal{N}\left(0, \frac{\langle g(g+g^\sharp)f^2\rangle_0}{\langle af\rangle_0^2}\right),$$

where $g = b - a\langle bf\rangle_0/\langle af\rangle_0$.

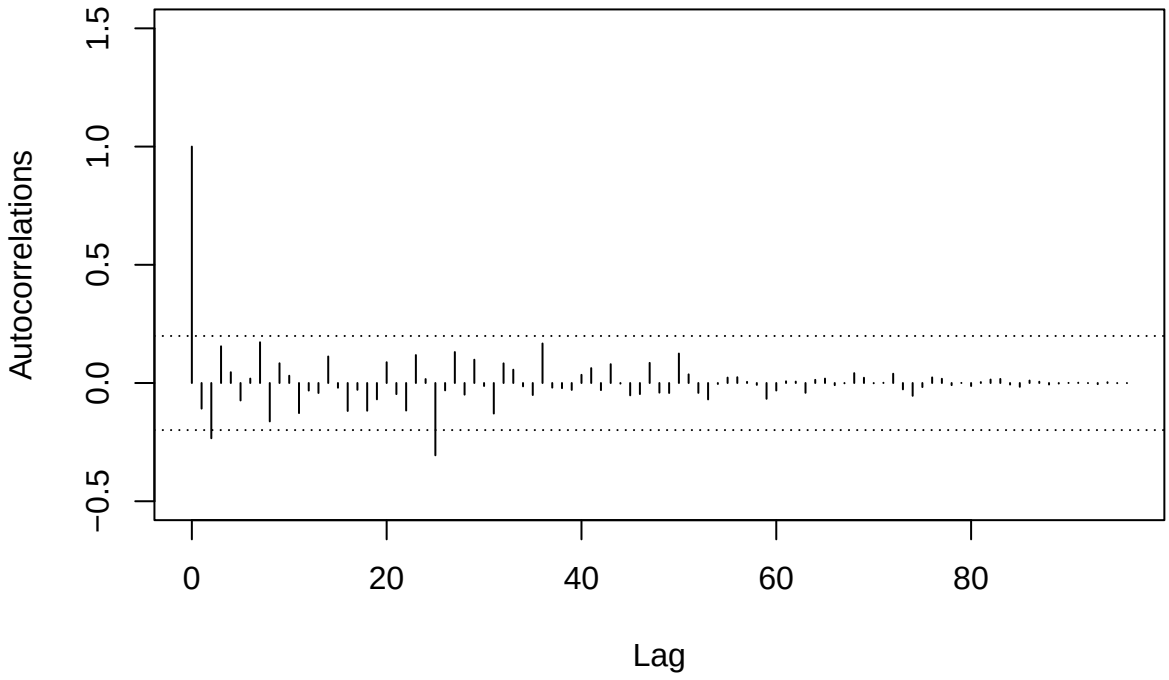
### Remark 9.6.10. Bartlett's Formula for the Autocorrelations

- The sample autocorrelation is defined as $\widehat{\rho}(k) = \widehat{\gamma}(k)/\widehat{\gamma}(0)$.
- Applying Corollary 9.6.9 with $b(\lambda) = \cos(\lambda k)$ and $a \equiv 1$, we find the asymptotic variance for the sample autocorrelation is
$$\frac{\langle f^2\rangle_{2k}}{\langle f\rangle_0^2} + \left(1+2\rho(k)^2\right)\frac{\langle f^2\rangle_0}{\langle f\rangle_0^2} - 4\rho(k)\frac{\langle f^2\rangle_k}{\langle f\rangle_0^2}.$$
- For white noise, this equals 1.

### Remark 9.6.12. Autocorrelations of Reduced Population Data

- Consider second differences of U.S. population data.
- We compute the sample autocorrelations, and test the hypothesis of zero serial correlation (i.e., white noise).
- For each lag $k \geq 1$ (treated separately, ignoring multiple testing...) the asymptotic 95% critical values are $\pm 1.96/\sqrt{n}$.

```
pop <- read.table("USpop.dat")
pop <- ts(pop, start = 1901)
diffdiff.pop <- diff(diff(pop*10e-6))
data <- diffdiff.pop
n <- length(data)
acfs.sample <- NULL
mu.hat <- sum(data)/n
for(k in 0:(n-1))
{
    acf.sample <- sum((data[1:(n-k)]-mu.hat)*(data[(k+1):n]-mu.hat))/n
    acfs.sample <- c(acfs.sample,acf.sample)
}
plot(ts(acfs.sample/acfs.sample[1],start=0),xlab="Lag",ylab="Autocorrelations",ylim=c(-.5,1.5),type="h")
abline(h= 1.96/sqrt(n),lty=3)
abline(h= -1.96/sqrt(n),lty=3)
```




## Lesson 9-7: Periodogram

- We further investigate the periodogram.
- It can be viewed as a crude estimator of the spectral density.

### Definition 9.7.2

- Given a sample of size $n$ from a time series $\{X_t\}$, the centered DFT is

$$\widehat{X}(\lambda) = n^{-1/2} \sum_{t=1}^{n} (X_t - \overline{X}) e^{-i\lambda t}$$

  for $\lambda \in [-\pi, \pi]$.

- So $\left|\widehat{X}(\lambda)\right|^2 = I(\lambda)$.

### Corollary 9.7.5

Suppose $\{X_t\}$ is either $m$-dependent or is a linear time series with mean $\mu$ and causal moving average representation. If $\mu = 0$ then

$$\frac{\widetilde{I}(\lambda)}{f(\lambda)} \Rightarrow \begin{cases} \chi_1^2 & \text{if } |\lambda| = 0, \pi \\ .5\chi_2^2 & \text{if } |\lambda| \in (0, \pi). \end{cases}$$

If $\mu \neq 0$, we use the centered DFT and $I(\lambda)/f(\lambda)$ has the same limit if $\lambda \neq 0$.

### Remark 9.7.6. The Periodogram is Inconsistent

- Corollary 9.7.5 implies the periodogram is not consistent as an estimator of the spectral density.
- In contrast, Theorem 9.6.6 shows that weighted averages of the periodogram are consistent for spectral means.

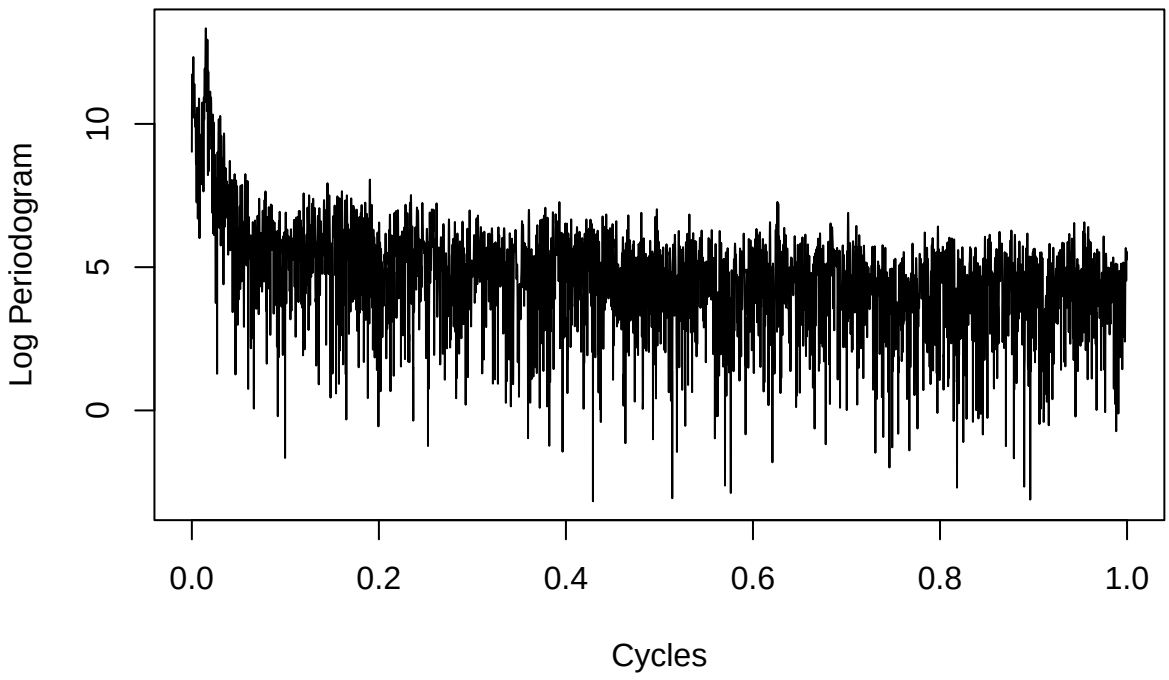
### Fact 9.7.7. Independence of Periodogram Ordinates over Fourier Frequencies

- From Corollary 7.2.9, DFT ordinates are approximately uncorrelated.
- Because they are also asymptotically complex Gaussian (see book), the DFT ordinates are asymptotically independent.
- Hence $I(\lambda_j)$ is asymptotically independent of $I(\lambda_k)$ for $j \neq k$.

### Example 9.7.9. Periodogram of the Wolfer Sunspots

- We can plot the periodogram to identify cycles in a time series.
- We plot the periodogram of the Wolfer sunspots (recall Exercise 7.18).
- We omit the value $I(0) = 0$ so that we can plot in logs.
- The maximum occurs at $\lambda = .047$, corresponding to a period of 133.33 months, or 11.08 years.

```
wolfer <- read.table("wolfer.dat")
wolfer <- ts(wolfer[,1],start=1749,frequency=12)
data <- wolfer
n <- length(data)
acfs.sample <- NULL
mu.hat <- sum(data)/n
for(k in 0:(n-1))
{
    acf.sample <- sum((data[1:(n-k)]-mu.hat)*(data[(k+1):n]-mu.hat))/n
    acfs.sample <- c(acfs.sample,acf.sample)
}
grid <- 10000
lambda <- seq(0,grid)*pi/grid
pgram <- cos(0*lambda)*acfs.sample[1]
for(h in 1:(n-1))
{
    pgram <- pgram + 2*cos(h*lambda)*acfs.sample[h+1]
}
pgram <- ts(pgram[-1],start=0,frequency=grid)
plot(log(pgram),xlab="Cycles",ylab="Log Periodogram")
```




```
print(pi*which.max(pgram)/grid)
```

```
## [1] 0.04712389
```

# Lesson 9-8: Spectral Density Estimation

- We estimate the spectral density by modifying the periodogram.

### Paradigm 9.8.1. Smoothing the Periodogram

- We can average the periodogram over nearby Fourier frequencies: for integer $m$,

$$\widehat{f}(\omega) = \frac{1}{2m+1} \sum_{j=-m}^{m} I(\omega + \lambda_j).$$

- Recall nonparametric smoothing: higher $m$ to reduce variance, but bias increases.
- More general smoothing of the periodogram:

$$\widehat{f}(\omega) = \frac{\sum_\ell W_n((\lambda_\ell - \omega)/m)\, I(\lambda_\ell)}{\sum_\ell W_n((\lambda_\ell - \omega)/m)},$$

where $W_n(x)$ is a kernel function, or *spectral window*, depending on $n$. The estimator also depends on the *bandwidth* $m$.

### Paradigm 9.8.3. Tapering the ACVF

- Recall from Remark 9.5.2 that tapering reduces variability in ACVF estimation. We can insert tapered sample ACVF into the periodogram formula:

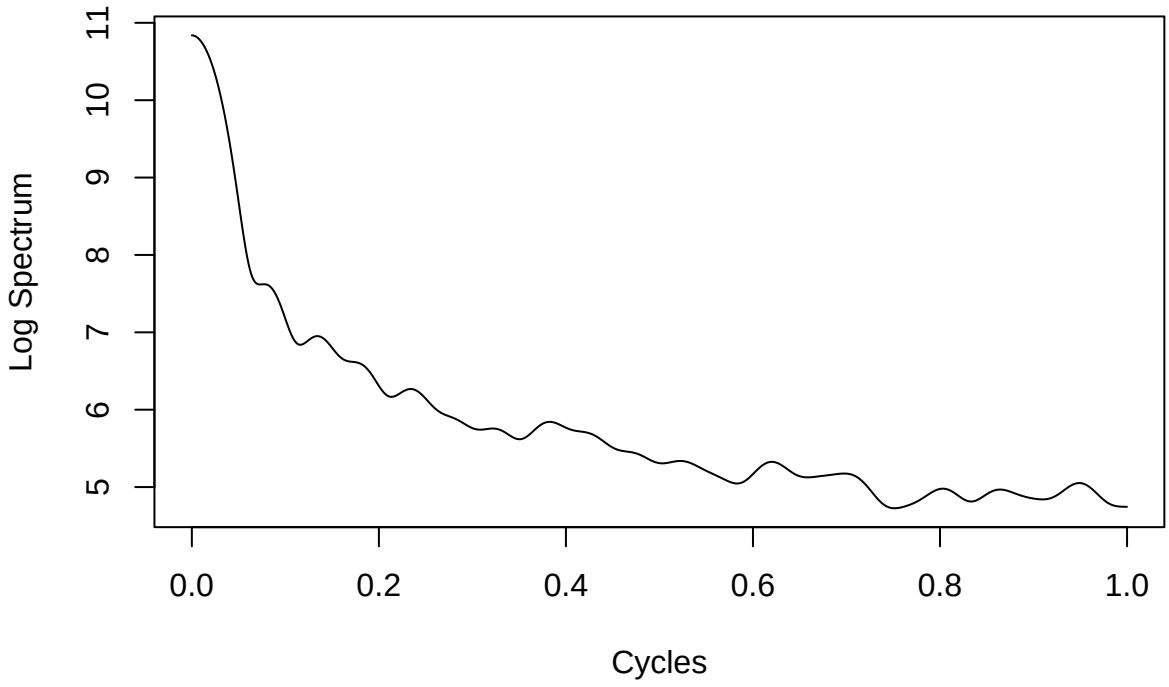
$$\widetilde{f}(\lambda) = \sum_{|h|\leq d} \Lambda(h/d)\widehat{\gamma}(h)e^{-i\lambda h},$$

  where $\Lambda(x)$ is a taper (cf. Definition 9.5.3), and $d$ is the *bandwidth*.
- We can rewrite $\widetilde{f}$ as a spectral window estimator as well, where the spectral window is proportional to $\sum_{|h|\leq d} \Lambda(h/d)e^{-i\lambda h}$.

### Example 9.8.6. Bartlett Tapered Spectral Estimator

- We apply the tapered acvf spectral estimator of Paradigm 9.8.3, using the Bartlett taper (Example 9.5.4), to the Wolfer sunspot data.
- We choose the bandwidth $d = 3n^{1/3} = 42$, based on some asymptotic theory.
- We plot in logs, and so remove the frequency zero value.

```
wolfer <- read.table("wolfer.dat")
wolfer <- ts(wolfer[,1],start=1749,frequency=12)
data <- wolfer
n <- length(data)
d <- 3*floor(n^{1/3})
acfs.sample <- NULL
mu.hat <- sum(data)/n
for(k in 0:(n-1))
{
    acf.sample <- sum((data[1:(n-k)]-mu.hat)*(data[(k+1):n]-mu.hat))/n
    acfs.sample <- c(acfs.sample,acf.sample)
}
grid <- 10000
lambda <- seq(0,grid)*pi/grid
pgram <- cos(0*lambda)*acfs.sample[1]
for(h in 1:(n-1))
{
    pgram <- pgram + 2*(max(1-h/d,0))*cos(h*lambda)*acfs.sample[h+1]
}
pgram <- ts(pgram[-1],start=0,frequency=grid)
plot(log(pgram),xlab="Cycles",ylab="Log Spectrum")
```




## Lesson 9-9: Spectral Analysis

- We refine the analysis of spectral density estimation.
- We consider the class of flat-top tapers.

### Paradigm 9.9.1. An Interesting Class of Tapers

- Consider a taper that takes the value 1 in a neighborhood of zero: this improves the bias.
- This suggests the definition $\Lambda_\infty(x) = 1$ for $|x| \leq c$, for some $c \in (0, 1]$. We call these *flat-top* tapers.
- Example: Bartlett taper (Example 9.5.4) is a limiting case with $c = 0$.
- Example: rectangular taper with $c = 1$, where $\Lambda(x) = 1$ for $|x| \leq 1$, zero otherwise.
- Example: trapezoidal taper (depending on choice of $c$):

$$\Lambda_\infty(x) = \begin{cases} 1 & \text{if } |x| \leq c \\ \frac{1-|x|}{1-c} & \text{if } c < |x| \leq 1. \end{cases}$$

- Flat-top tapers correct bias by shifting the spectral estimate down, which can generate negative estimates. One can take the maximum with zero, so as to enforce a non-negative spectrum estimate.

### Example 9.9.6. Application of a Trapezoidal Taper

- We apply a trapezoidal taper to the Wolfer sunspots, using $c = 1/3$ and $d = 42$.
- The spectral estimate has more variability, and is no longer positive (log of negative values is cut off).

```
wolfer <- read.table("wolfer.dat")
wolfer <- ts(wolfer[,1],start=1749,frequency=12)
```

```
data <- wolfer
n <- length(data)
d <- 3*floor(n^{1/3})
acfs.sample <- NULL
mu.hat <- sum(data)/n
for(k in 0:(n-1))
{
    acf.sample <- sum((data[1:(n-k)]-mu.hat)*(data[(k+1):n]-mu.hat))/n
    acfs.sample <- c(acfs.sample,acf.sample)
}
cutoff <- 1/3
trap <- function(h)
{
    val <- min(max((1 - h/d)/(1 - cutoff),0),1)
    return(val)
}
grid <- 10000
lambda <- seq(0,grid)*pi/grid
pgram <- cos(0*lambda)*acfs.sample[1]
for(h in 1:(n-1))
{
    pgram <- pgram + 2*trap(h)*cos(h*lambda)*acfs.sample[h+1]
}
pgram <- pmax(0,pgram)
pgram <- ts(pgram[-1],start=0,frequency=grid)
plot(log(pgram),xlab="Cycles",ylab="Log Spectrum",ylim=c(3,12))
```