STAT 436 / 536 - Lecture 4

September 10, 2018

1. Correlation Structure and Motivation

- We have seen how to decompose a time series model to remove a trend and seasonal components. So what remains?
  - *The random component*
  - *After removing the trend and seasonal components, do we expect the consecutive random components to be similar, in other words will they be correlated?*
  - *Yes, most likely autocorrelation is still present.*

2. Expectation, Variance, and Auto Correlation

- The expected value, or expectation, or a random variable is defined as: $E[(x)] = \int xf(x)dx$.
- In the context of annual measurements of Nile River flows, what is an interpretation of the expectation? *average flow*
- A times series model is stationary in the mean if: *the mean function is constant in time.*
- What is $E[(x - \mu_x)(y - \mu_y)]$? -*This is known as covariance and is a more general form of variance:* $E[(x - \mu_x)^2]$

  -*What is the relationship between covariance and correlation? correlation* $= \frac{E[(x-\mu_x)(y-\mu_y)]}{\sigma_x \sigma_y}$

Sample Based Moments

- Sample based calculations can be made in R mean using `mean(x)`, variance `var(x)`, covariance `cov(x,y)`, and correlation `cor(x,y)`.

- Using a dataset containing information on housing sales in King County, WA http://math.montana.edu/ahoegh/teaching/stat408/datasets/SeattleHousing.csv, compute the following quantities:

```r
Seattle <- read.csv('http://math.montana.edu/ahoegh/teaching/stat408/datasets/SeattleHousing.csv')
```

- mean sales price (price)

```r
signif(mean(Seattle$price), 3)
```

```
## [1] 633000
```

- standard deviation of sales price

```r
signif(sd(Seattle$price), 3)
```

```
## [1] 635000
```

- correlation between sales price and square footage (sqft_living)

```r
round(cor(Seattle$price, Seattle$sqft_living),2)
```

```
## [1] 0.78
```

- Intepret the three quantities above.

3. Autocorrelation and the Correlogram

- In addition to mean and variance, the serial correlation (or autocorrelation) is an important element in time series modeling.

- Autocovariance is defined as: $\gamma_k = E[(x_t - \mu_t)(x_{t+k} - \mu_[t + k])]$.

- A time series model is second-order stationary if *the correlation between time steps only depends on the number of time points between them.*

- The autocorrelation function is defined as $\rho_k = \frac{\gamma_k}{\sigma^2}$

- Similar to variance calculations, the sample autocovariance and autocorrelation functions can be computed:
- sample acvf: $c_k = \frac{1}{n}\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - barx)$. Note n is used in the denominator.

- sample acf: $r_k = \frac{c_k}{c_0}$.

- Note these properties require a stationary process, hence trends and cyclical patterns need to be removed when considering correlated random noise.

Simulating Correlated Time Series Data

- As was mentioned earlier in class, we can think of time series modeling as similar to mixed models. Specifically, there is a specific correlation structure defined for each type of model.

- First construct a covariance matrix between all of the observations.

```
set.seed(09062018)
time.pts <- 200
auto.corr <- 0.9
evolution.matrix <- diag(time.pts)
for (column in 1:time.pts){
  evolution.matrix[,column] <- auto.corr ^ abs((1:time.pts) - column)
}
library(knitr) # for kable
kable(evolution.matrix[1:5,1:5],caption = "Covariance matrix for first 5 time points")
```

Table 1: Covariance matrix for first 5 time points

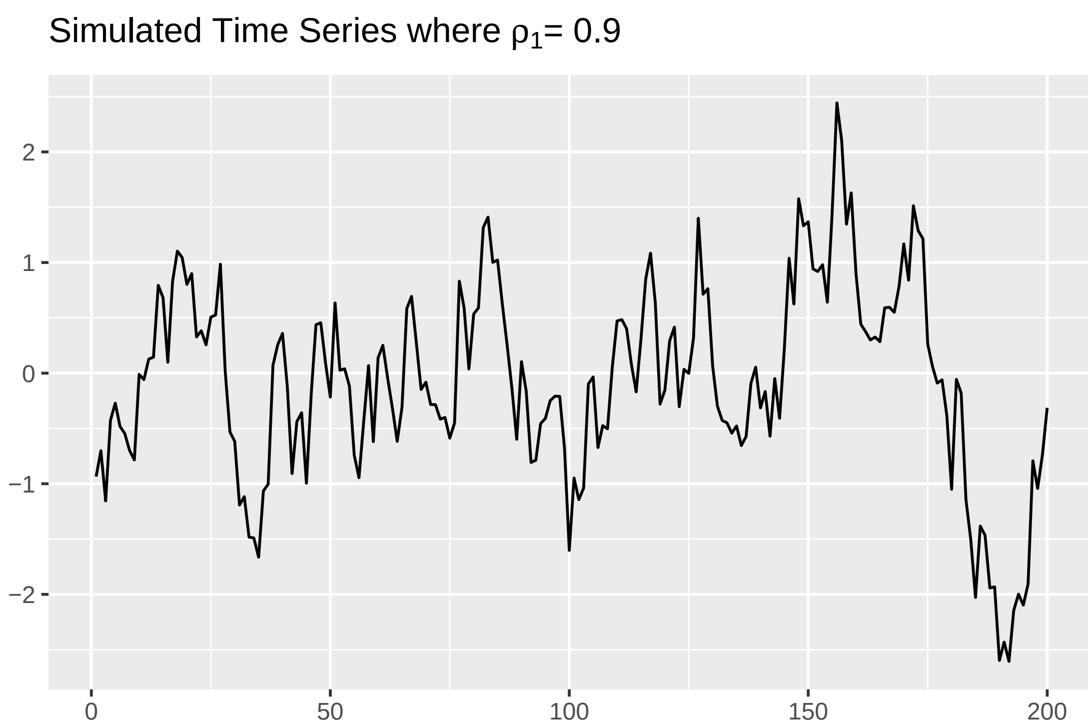
1.0000	0.900	0.81	0.729	0.6561
0.9000	1.000	0.90	0.810	0.7290
0.8100	0.900	1.00	0.900	0.8100
0.7290	0.810	0.90	1.000	0.9000
0.6561	0.729	0.81	0.900	1.0000

- Simulate a vector of correlated normal random variables.

```
library(mnormt) # for rmnorm
Y <- as.ts(rmnorm(n=1, mean=0, varcov=evolution.matrix))
```

- Create time series figure

```
library(ggfortify) # for autoplot
autoplot(Y) + ggtitle(expression(paste('Simulated Time Series where ', rho[1], '= 0.9')))
```



Exercises:

1. Is the simulated time series stationary in the mean, why or why not?

2. What is γ_k +

3. What is ρ_k =

4. Change the `auto.corr` variable, rerun the simulation and describe how your figures are different.

   a. `auto.corr = 0`

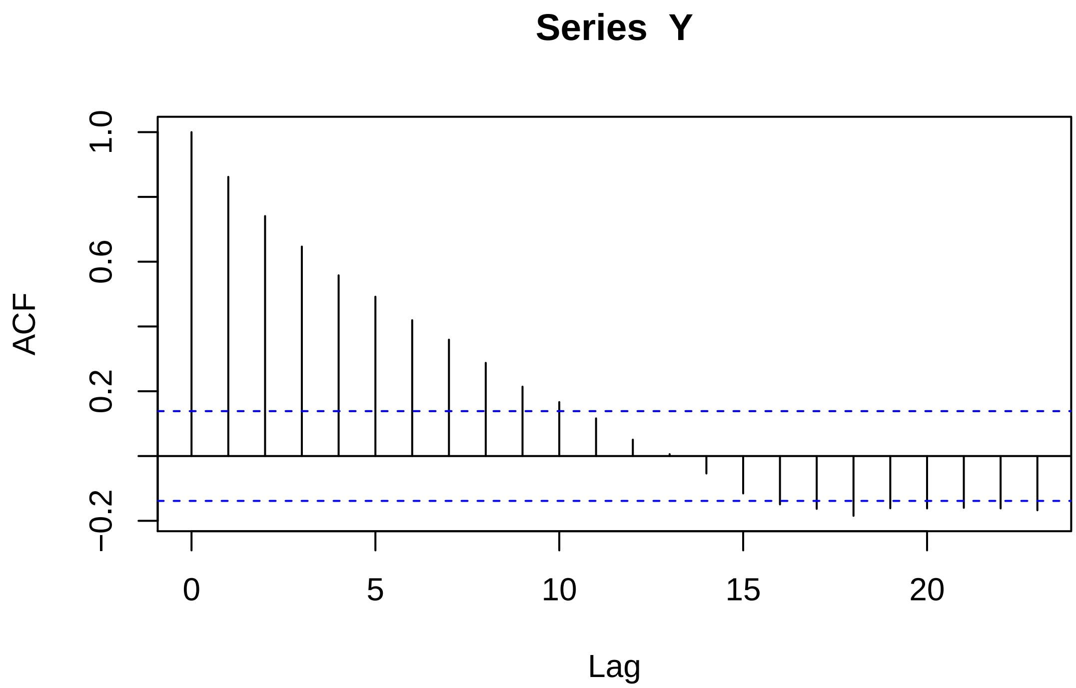
   b. `auto.corr = .5`

   c. `auto.corr = -.9`

5.(536) Adapt the code to include a trend and seasonal cycle in addition to the serial correlated random innovations.

- A useful tool for identifying autocorrelation structure in a time series dataset is the correlogram. The command for this in R is `acf()`.

```
acf(Y)
```



- Correlograms have the following properties: - The *x-axis gives lag(k) and the y-axis gives the autocorrelation.*

- The lag 0 autocorrelation is always 1 and is include for comparison purposes.

- If $\rho_k = 0$, then the sampling distribution of r_k is (approximately) normal with mean -1/n and variance of 1/n. *The dotted lines are drawn on the correlogram at* $-\frac{1}{n} \pm \frac{2}{\sqrt{n}}$.

- It is important to have a stationary time series that does not include deterministic signals, such as a trend or cycle.

Airline Passenger Example Section 2.3.2

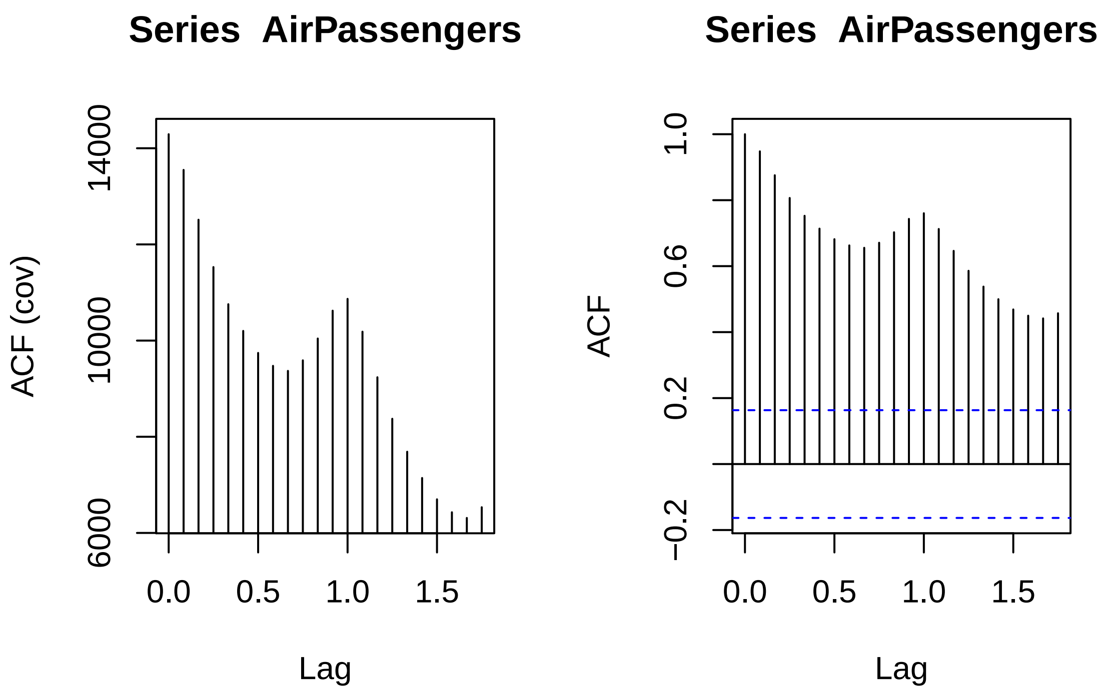
Load Data and Decompose Time Series

```
data("AirPassengers")
AP.decomp <- decompose(AirPassengers, 'additive')
str(AP.decomp)
```

```
## List of 6
##  $ x       : Time-Series [1:144] from 1949 to 1961: 112 118 132 129 121 135 148 148 136 119 ...
##  $ seasonal: Time-Series [1:144] from 1949 to 1961: -24.75 -36.19 -2.24 -8.04 -4.51 ...
##  $ trend   : Time-Series [1:144] from 1949 to 1961: NA NA NA NA NA ...
##  $ random  : Time-Series [1:144] from 1949 to 1961: NA NA NA NA NA ...
##  $ figure  : num [1:12] -24.75 -36.19 -2.24 -8.04 -4.51 ...
##  $ type    : chr "additive"
##  - attr(*, "class")= chr "decomposed.ts"
```

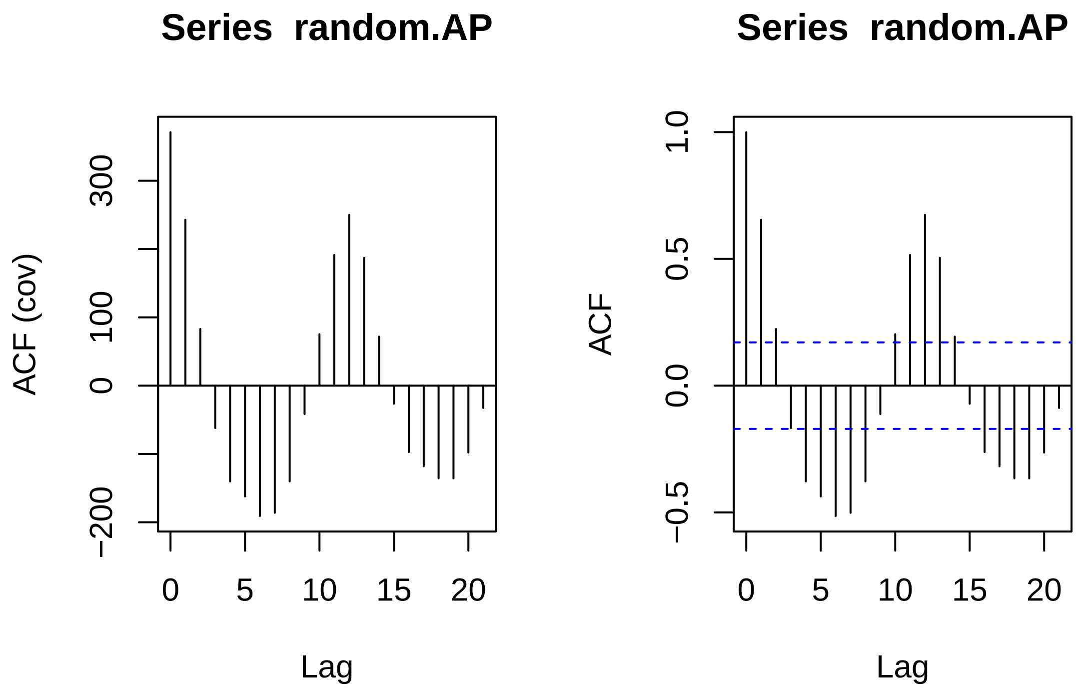
ACF Plot with Air Passengers Data (Covariance)

```
par(mfcol=(c(1,2)))
acf(AirPassengers, type = 'covariance')
acf(AirPassengers)
```



ACF Plot on Decomposed Random Component (Covariance)

```
#exclude NA's
random.AP <- AP.decomp$random[!is.na(AP.decomp$random)]
par(mfcol=(c(1,2)))
acf(random.AP, type='covariance')
acf(random.AP)
```


Series random.AP
ACF (cov)
300
100
0
−200
0
5
10
15
20
Lag
Series random.AP
ACF
1.0
0.5
0.0
−0.5
0
5
10
15
20
Lag