

W271-2 – Spring 2016 – HW 6

Juanjo Carin, Kevin Davis, Ashley Levato, Minghu Song

March 16, 2016

Contents

Exercises	2
Exercise 1	2
Exercise 2	3
Exercise 3	6
Exercise 4	9

Exercises

Exercise 1

- a. **Discuss the mean and variance functions and how the similarities and differences from those we studied in classical linear model.**

The mean function for a time series is defined by the function:

$$\mu_x(t) = E(x_t) = \int_{-\infty}^{+\infty} x_t f_t(x_t) dx_t$$

This function has a time component so the mean could be different in different time periods. This is different from a mean in classical linear models where the mean is constant.

The variance function for a time series analysis is defined by the function:

$$\sigma_x^2(t) = E(x_t - \mu_x(t))^2 = \int_{-\infty}^{+\infty} (x_t - \mu_x(t))^2 f_t(x_t) dx_t$$

Again this function is time dependent which means it varies with time unlike the variance in a classical linear model.

...

- b. **Define strict and weak stationarity**

Stationarity indicates the parameter is consistent across time.

Strict stationarity is when the joint distributions $F(x_{t_1}, \dots, x_{t_n})$ and $F(x_{t_1+m}, \dots, x_{t_n+m})$ are the same implying that the distribution is unchanged for any time shift.

Weak stationarity (also called second-order stationary) is when its mean and variance are stationary and its autocovariance $Cov(x_t, x_{t+k})$ depends on the time placement k and can be written as $\gamma^{(k)}$. Once a distribution assumption is imposed the series can be completely characterized by the mean and covariance.

...

Exercise 2

- a. Generate a zero-drift random walk model using 500 simulation.

```
# QUESTION 2 -----  
x<-w<-rnorm(500)  
for (t in 2:500) x[t] <- x[t-1] + w[t]
```

...

- b. Provide the descriptive statistics of the simulated realizations. The descriptive statistics should include the mean, standard deviation, 25th, 50th, and 75th quantiles, minimum, and maximum.

```
mean(x)
```

```
[1] 14.99322
```

```
sd(x)
```

```
[1] 6.840745
```

```
min(x)
```

```
[1] -2.386488
```

```
max(x)
```

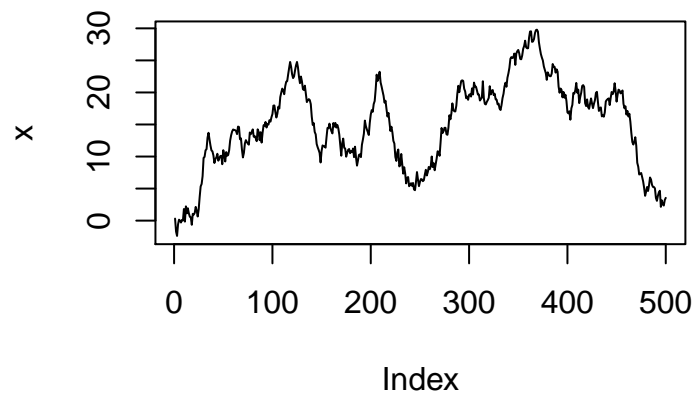
```
[1] 29.79157
```

```
quantile(x, probs=c(.25, .5, .75))
```

```
      25%      50%      75%  
10.32172 15.39484 19.94769
```

- c. Plot the time-series plot of the simulated realizations.

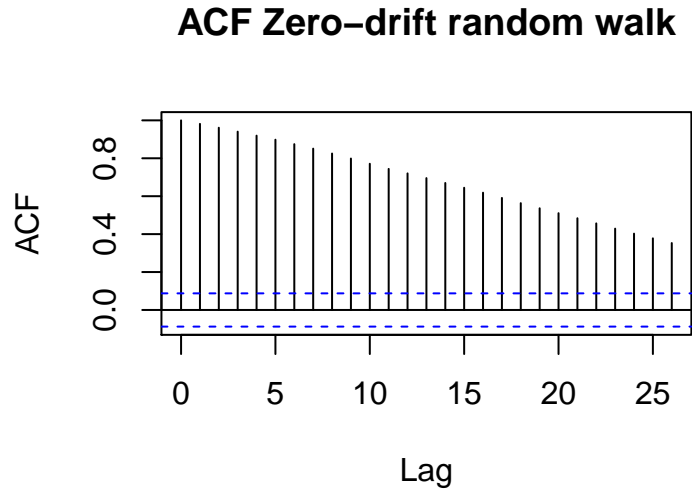
```
plot(x, type= "l")
```



...

d. Plot the autocorrelation graph.

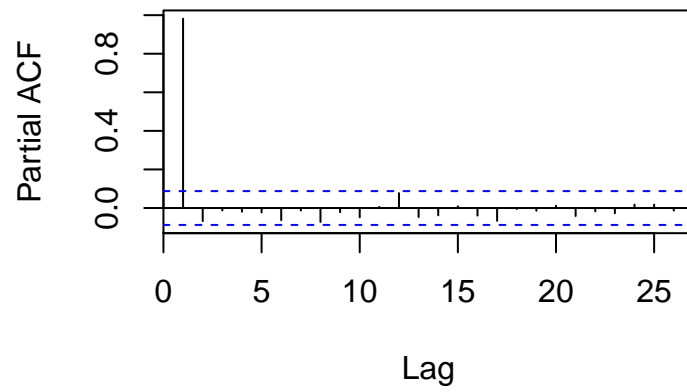
```
acf(x, main="ACF Zero-drift random walk")
```



...

e. Plot the partial autocorrelation graph.

```
pacf(x, main = "PACF Zero-drift random walk")
```

PACF Zero-drift random walk

...

Exercise 3

- a. Generate arandom walk with drift model using 500 simulation, with the drift = 0.5.

```
x1<-w1<-rnorm(500)
d<-0.5
for (t in 2:500) x1[t] <- x1[t-1] + w1[t] + d
```

...

- b. Provide the descriptive statistics of the simulated realizations. The descriptive statistics should include the mean, standard deviation, 25th, 50th, and 75th quantiles, minimum, and maximum.

```
mean(x1)
```

```
## [1] 137.3032
```

```
sd(x1)
```

```
## [1] 79.47487
```

```
min(x1)
```

```
## [1] -1.020947
```

```
max(x1)
```

```
## [1] 283.3976
```

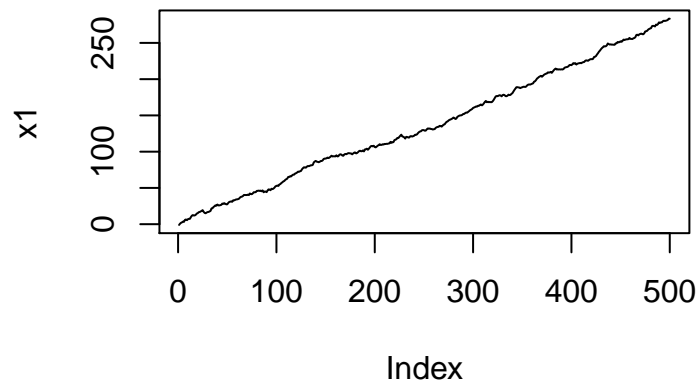
```
quantile(x1, probs=c(.25, .5, .75))
```

```
##      25%      50%      75%
## 75.14559 129.55476 208.08494
```

...

- c. Plot the time-series plot of the simulated realizations.

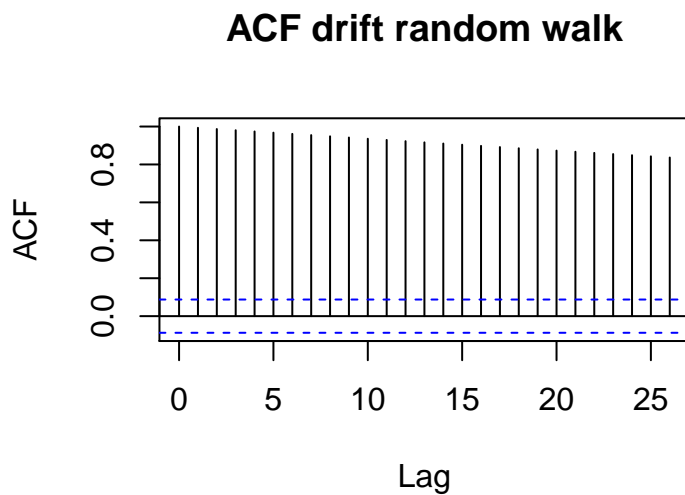
```
plot(x1, type= "l")
```



...

d. Plot the autocorrelation graph.

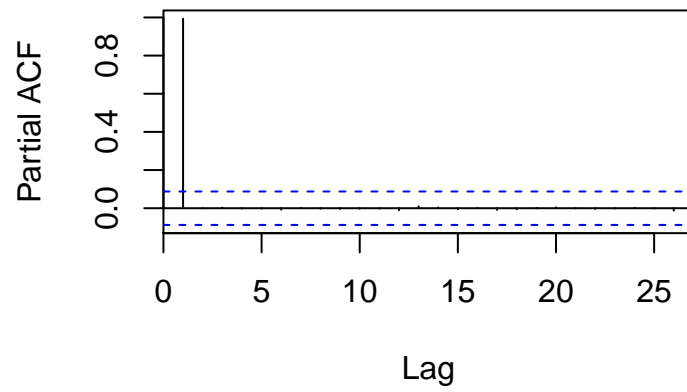
```
acf(x1, main="ACF drift random walk")
```



...

e. Plot the partial autocorrelation graph.

```
pacf(x1, main="PACF drift random walk")
```

PACF drift random walk

...

Exercise 4

Use the series from INJCJC.csv.

- a. Load the data and examine the basic structure of the data using `str()`, `dim()`, `head()`, and `tail()` functions.

```
INJCJC <- read.table("INJCJC.csv", header = TRUE, sep=",")
str(INJCJC)
```

```
'data.frame': 1300 obs. of 3 variables:
 $ Date : Factor w/ 1300 levels "1-Apr-05","1-Apr-11",...: 1102 143 442 784 483 1271 312 654 498 1286
 $ INJCJC : int 355 369 375 345 368 367 348 350 351 349 ...
 $ INJCJC4: num 362 366 364 361 364 ...
```

```
dim(INJCJC)
```

```
[1] 1300 3
```

```
head(INJCJC)
```

	Date	INJCJC	INJCJC4
1	5-Jan-90	355	362.25
2	12-Jan-90	369	365.75
3	19-Jan-90	375	364.25
4	26-Jan-90	345	361.00
5	2-Feb-90	368	364.25
6	9-Feb-90	367	363.75

```
tail(INJCJC)
```

	Date	INJCJC	INJCJC4
1295	24-Oct-14	288	281.25
1296	31-Oct-14	278	279.00
1297	7-Nov-14	293	285.75
1298	14-Nov-14	292	294.25
1299	21-Nov-14	314	294.25
1300	28-Nov-14	297	299.00

- b. Convert the variables INJCJC into a time series object `frequency=52`, `start=c(1990,1,1)`, `end=c(2014,11,28)`. Examine the converted data series.

```
INJCJC_ts<- ts(INJCJC$INJCJC, start=c(1990,1,1), end=c(2014,11,28), frequency=52)
str(INJCJC_ts)
```

```
## Time-Series [1:1259] from 1990 to 2014: 355 369 375 345 368 367 348 350 351 349 ...
```

```
dim(INJCJC_ts)
```

```
## NULL
```

```
head(INJCJC_ts)
```

```
## [1] 355 369 375 345 368 367
```

```
tail(INJCJC_ts)
```

```
## [1] 329 334 345 328 343 330
```

- c. Define a variable using the command `INJCJC.time<-time(INJCJC)`.

```
INJCJC_ts.time<-time(INJCJC_ts)
```

- d. Using the following command to examine the first 10 rows of the data. Change the parameter to examine different number of rows of data.

```
head(cbind(INJCJC.time, INJCJC),10)
```

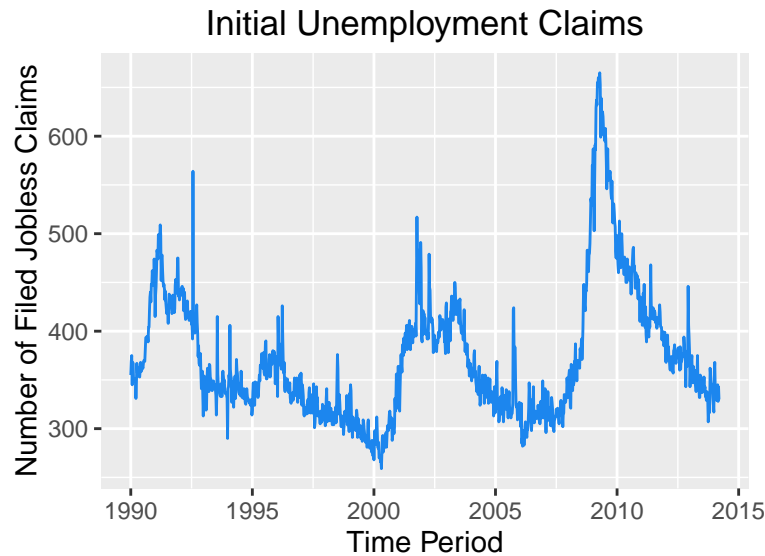
```
head(cbind(INJCJC_ts.time, INJCJC_ts),10)
```

	INJCJC_ts.time	INJCJC_ts
[1,]	1990.000	355
[2,]	1990.019	369
[3,]	1990.038	375
[4,]	1990.058	345
[5,]	1990.077	368
[6,]	1990.096	367
[7,]	1990.115	348
[8,]	1990.135	350
[9,]	1990.154	351
[10,]	1990.173	349

- e.

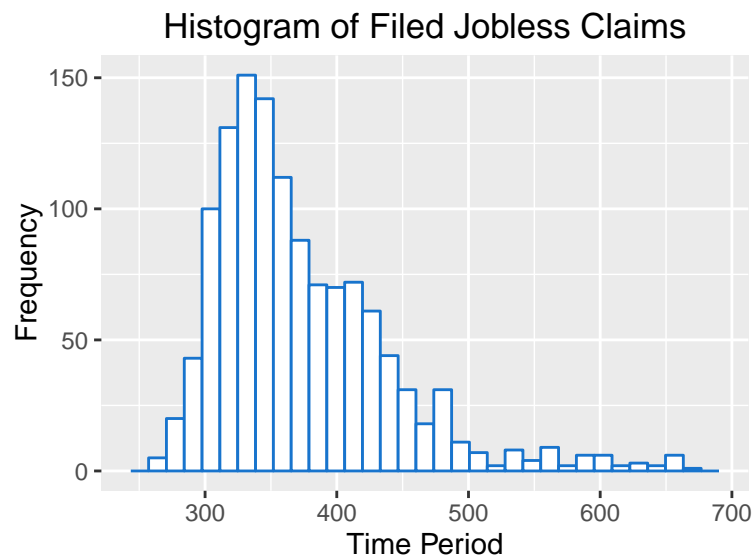
1. Plot the time series plot of `INJCJC`. Remember that the graph must be well labelled.

```
autoplot(INJCJC_ts , xlab = "Time Period", ylab = "Number of Filed Jobless Claims",  
         main="Initial Unemployment Claims", ts.colour = 'dodgerblue2' )
```



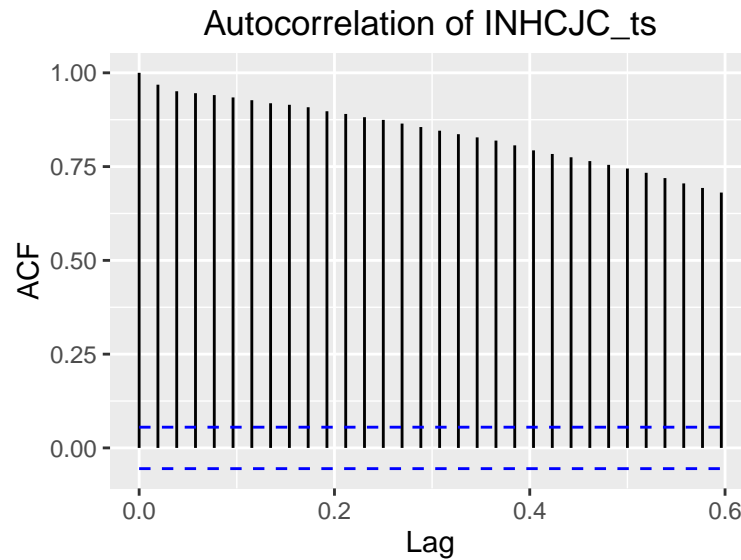
2. Plot the histogram of INJCJC. What is shown and not shown in a histogram?
How do you decide the number of bins used?

```
qplot(INJCJC_ts , geom="histogram", main='Histogram of Filed Jobless Claims',
      ylab='Frequency', xlab='Time Period', colour = I('dodgerblue3'),
      fill = I("white"))
```



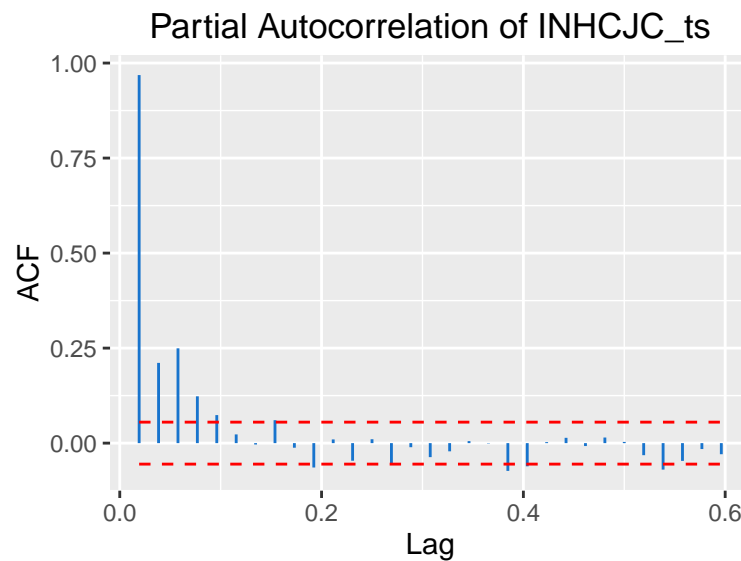
3. Plot the autocorrelation graph of INJCJC series.

```
autoplot(acf(INJCJC_ts, plot = FALSE), xlab = "Lag", main="Autocorrelation of INHCJC_ts",
        ts.colour = 'dodgerblue2')
```



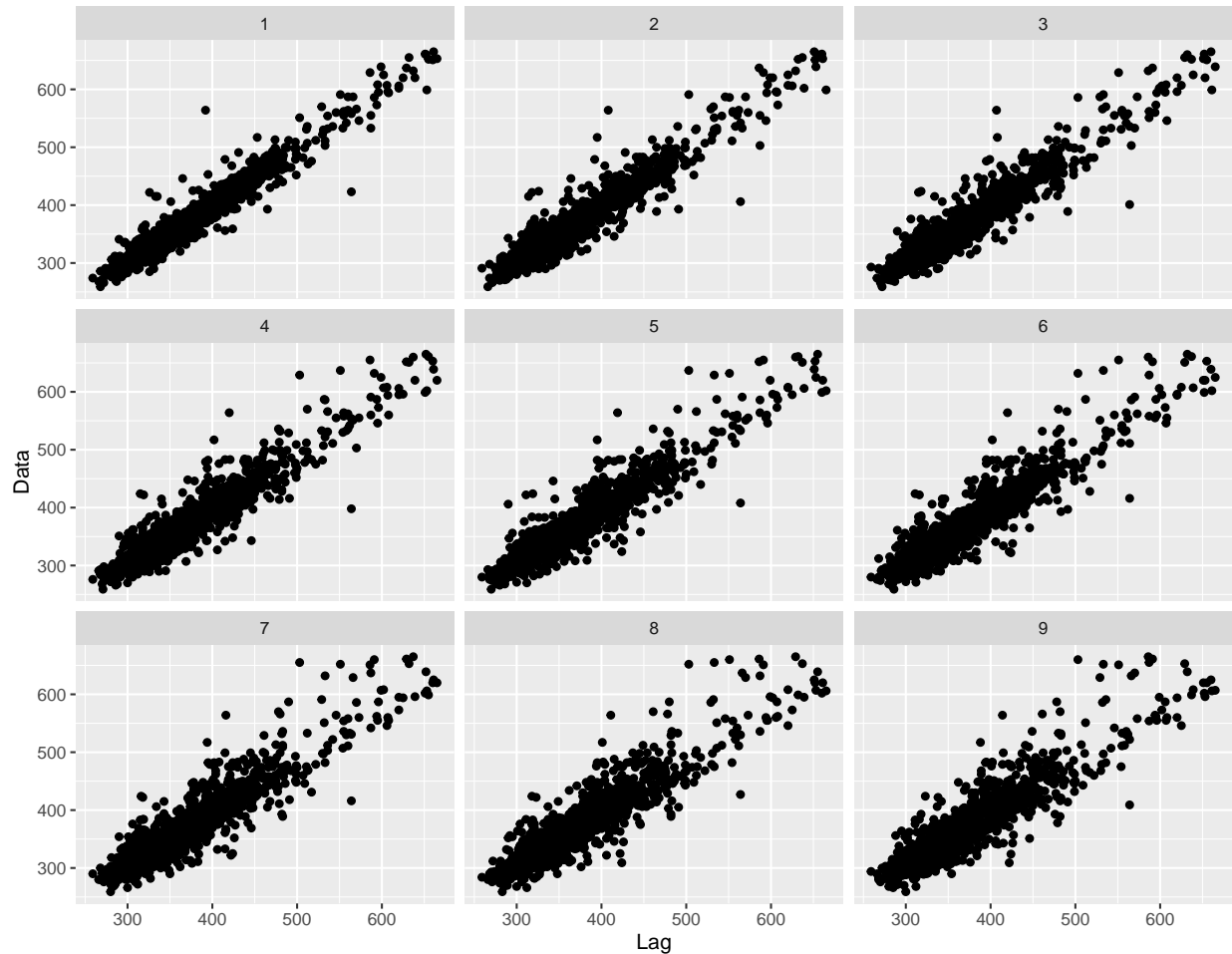
4. Plot the partial autocorrelation graph of INJCJC series.

```
autoplot(pacf(INJCJC_ts, plot = FALSE), xlab = "Lag", main = "Partial Autocorrelation of INHCJC_ts",
  colour = "dodgerblue3", conf.int.colour = "red")
```



5. Plot a 3x3 Scatterplot Matrix of correlation against lag values.

```
gglagplot(INJCJC_ts, lags = 9, nrow = 3, ncol = 3)
```



f.

1. Generate two symmetric Moving Average Smoothers. Choose the number of moving average terms such that one of the smoothers is very smoother and the other one can trace through the dynamics of the series. Plot the smoothers and the original series in one graph.

...

2. Generate two regression smoothers, one being a cubic trend regression and the other being a periodic regression. Plot the smoothers and the original series in one graph.

...

3. Generate kernel smoothers. Choose the smoothing parameters such that one of the smoothers is very smoother and the other one can trace through the dynamics of the series. Plot the smoothers and the original series in one graph.

...

4. Generate two nearest neighborhood smoothers. Choose the smoothing parametrts such that one of the smoothers is very smoother and the other one can trace through the dynamics of the series. Plot the smoothers and the original series in one graph.

...

5. Generate two LOWESS smoothers. Choose the smoothing parametrts such that one of the smoothers is very smoother and the other one can trace through the dynamics of the series. Plot the smoothers and the original series in one graph.

...

6. Generate two spline smoothers. Choose the smoothing parametrts such that one of the smoothers is very smoother and the other one can trace through the dynamics of the series. Plot the smoothers and the original series in one graph.

...
