

# ST565: Time Series HW7

Amirhosein “Emerson” Azarbakht [azarbaka@oregonstate.edu](mailto:azarbaka@oregonstate.edu)

## Question 1

1. Derive the spectrum for an MA(1) process.

$$\gamma(k) = \begin{cases} \beta_1^2 \sigma^2 + \sigma^2 & \text{for } k = 0 \\ \beta_1 \sigma^2 & \text{for } k = 1 \\ 0 & \text{for } k \geq 2 \end{cases}$$

$$f(\omega) = 1/\pi [\gamma(0) + 2\sum_{k=1}^{\infty} \gamma(k) \cos(\omega k)]$$

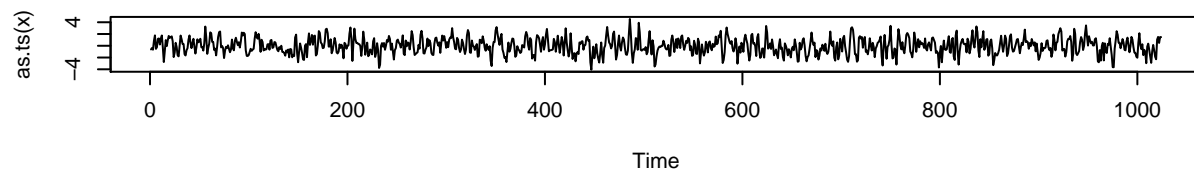
$$f(\omega) = 1/\pi [\beta_1^2 \sigma^2 + \sigma^2 + 2[\beta_1 \sigma^2] \cos(\omega)]$$

$$f(\omega) = 1/\pi [\beta_1^2 \sigma^2 + \sigma^2 + 2\beta_1 \sigma^2 \cos(\omega)]$$

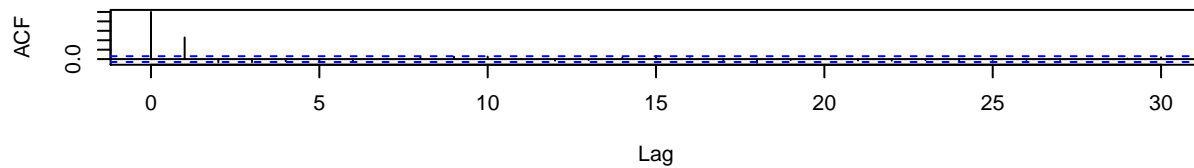
$$f(\omega) = \sigma^2/\pi [\beta_1^2 + 1 + 2\beta_1 \cos(\omega)]$$

Produce a plot of the spectrum showing the shape for a few values of  $\beta$ .

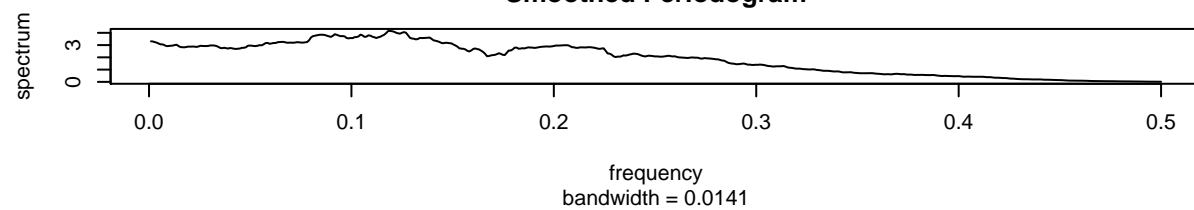
```
# MA(1)
set.seed(1)
x <- w <- rnorm(1024)
# beta = 0.9
for (t in 2:1024) x[t] <- 0.9 * w[t-1] + w[t]
layout(1:3)
plot(as.ts(x))
acf(x)
spectrum(x, span = 51, log = c("no"))
```



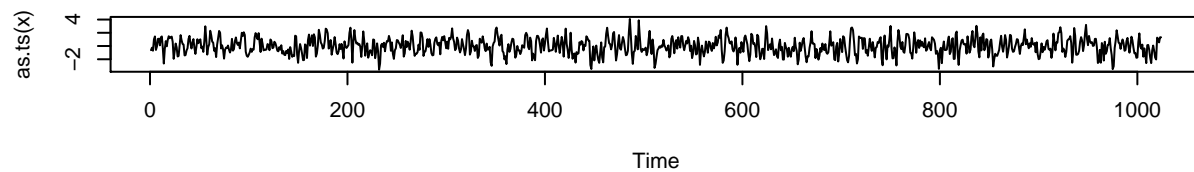
**Series x**



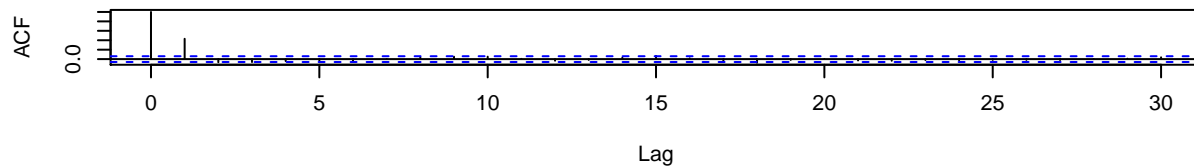
**Series: x  
Smoothed Periodogram**



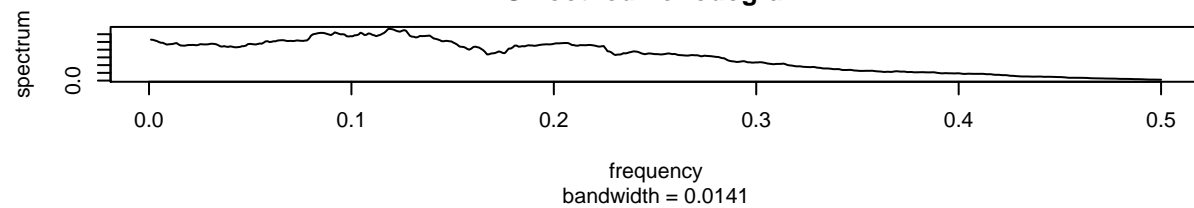
```
# beta = 0.7
for (t in 2:1024) x[t]<- 0.7 * w[t-1] + w[t]
layout(1:3)
plot(as.ts(x))
acf(x)
spectrum(x, span = 51, log = c("no"))
```



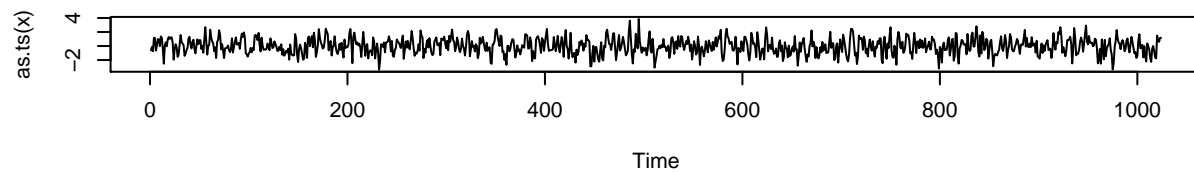
**Series x**



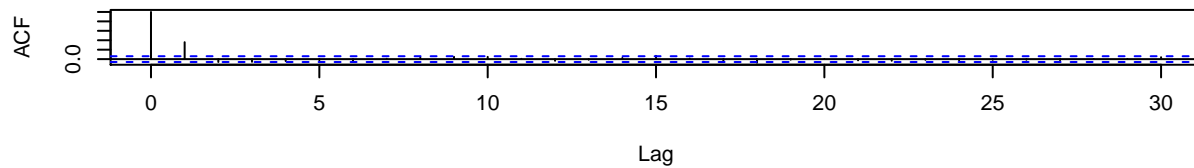
**Series: x  
Smoothed Periodogram**



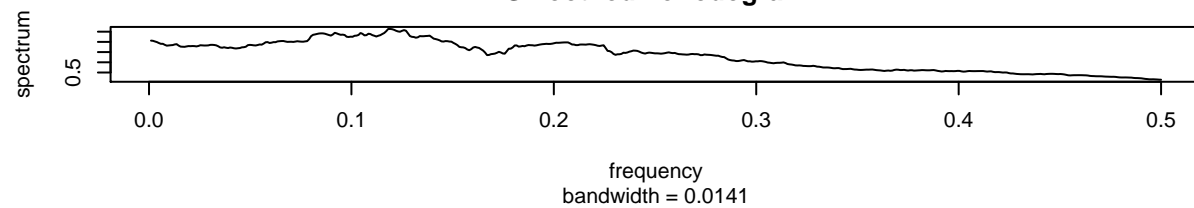
```
# beta = 0.5
for (t in 2:1024) x[t] <- 0.5 * w[t-1] + w[t]
layout(1:3)
plot(as.ts(x))
acf(x)
spectrum(x, span = 51, log = c("no"))
```



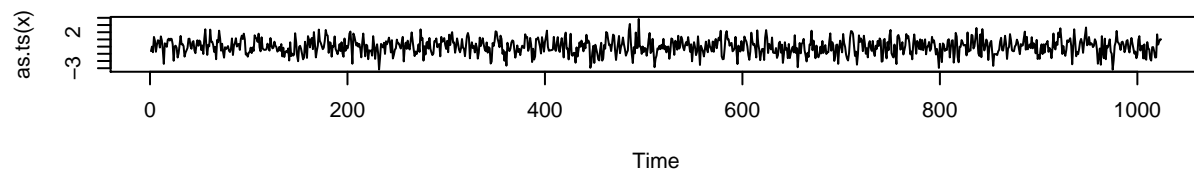
**Series x**



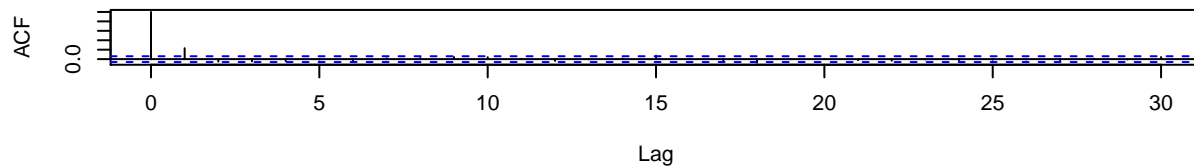
**Series: x  
Smoothed Periodogram**



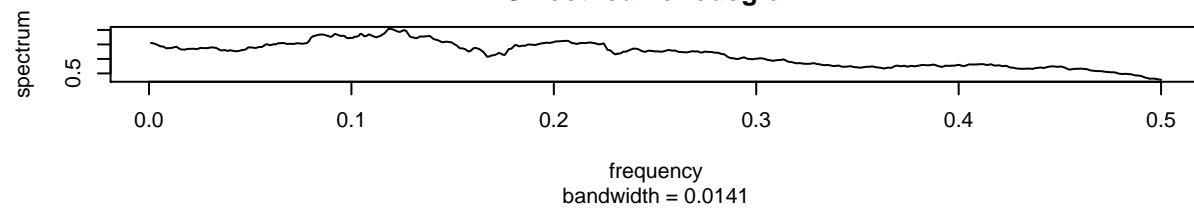
```
# beta = 0.3
for (t in 2:1024) x[t]<- 0.3 * w[t-1] + w[t]
layout(1:3)
plot(as.ts(x))
acf(x)
spectrum(x, span = 51, log = c("no"))
```



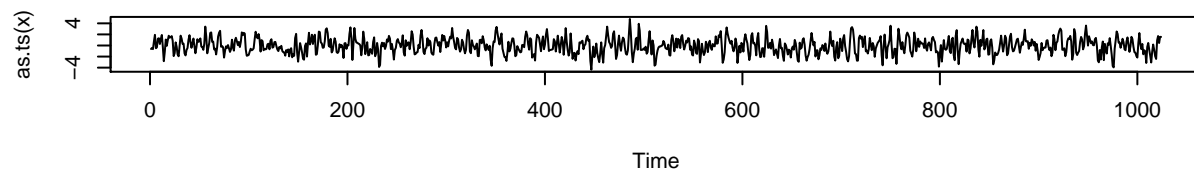
**Series x**



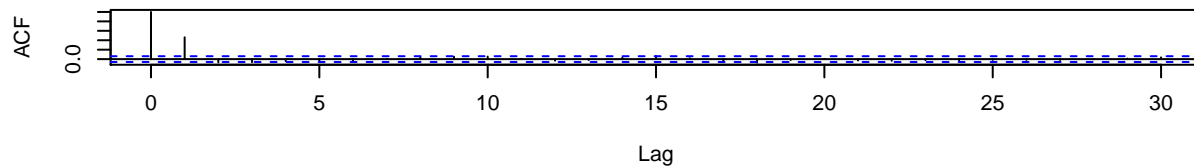
**Series: x  
Smoothed Periodogram**



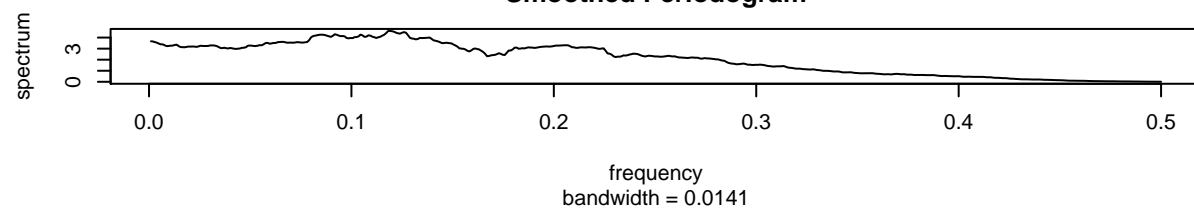
```
# beta = 1
for (t in 2:1024) x[t] <- 1 * w[t-1] + w[t]
layout(1:3)
plot(as.ts(x))
acf(x)
spectrum(x, span = 51, log = c("no"))
```



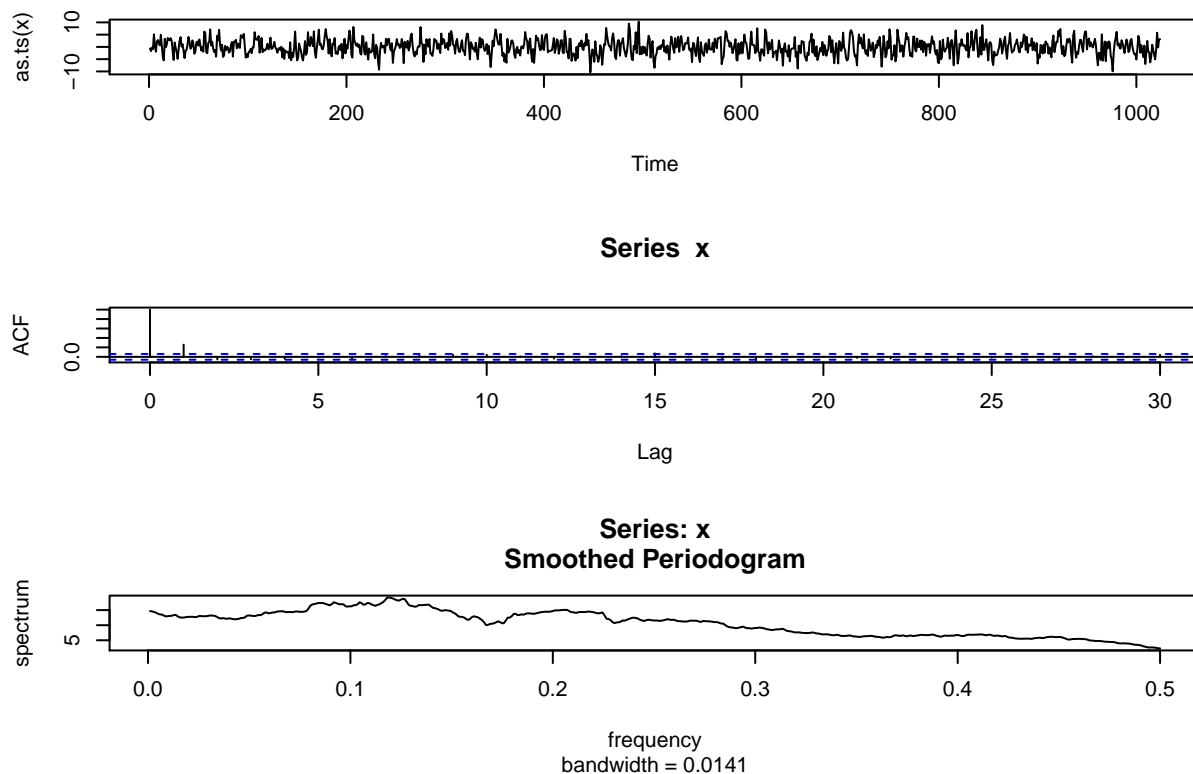
**Series x**



**Series: x  
Smoothed Periodogram**



```
# beta = 3
for (t in 2:1024) x[t] <- 3 * w[t-1] + w[t]
layout(1:3)
plot(as.ts(x))
acf(x)
spectrum(x, span = 51, log = c("no"))
```



2.

Show that if  $X_t$  and  $Y_t$  are independent, stationary processes with power spectral density functions  $f_x(\omega)$  and  $f_y(\omega)$ , then  $V_t = X_t + Y_t$  is also stationary with power spectral density  $f_v(\omega) = f_x(\omega) + f_y(\omega)$

## Q2

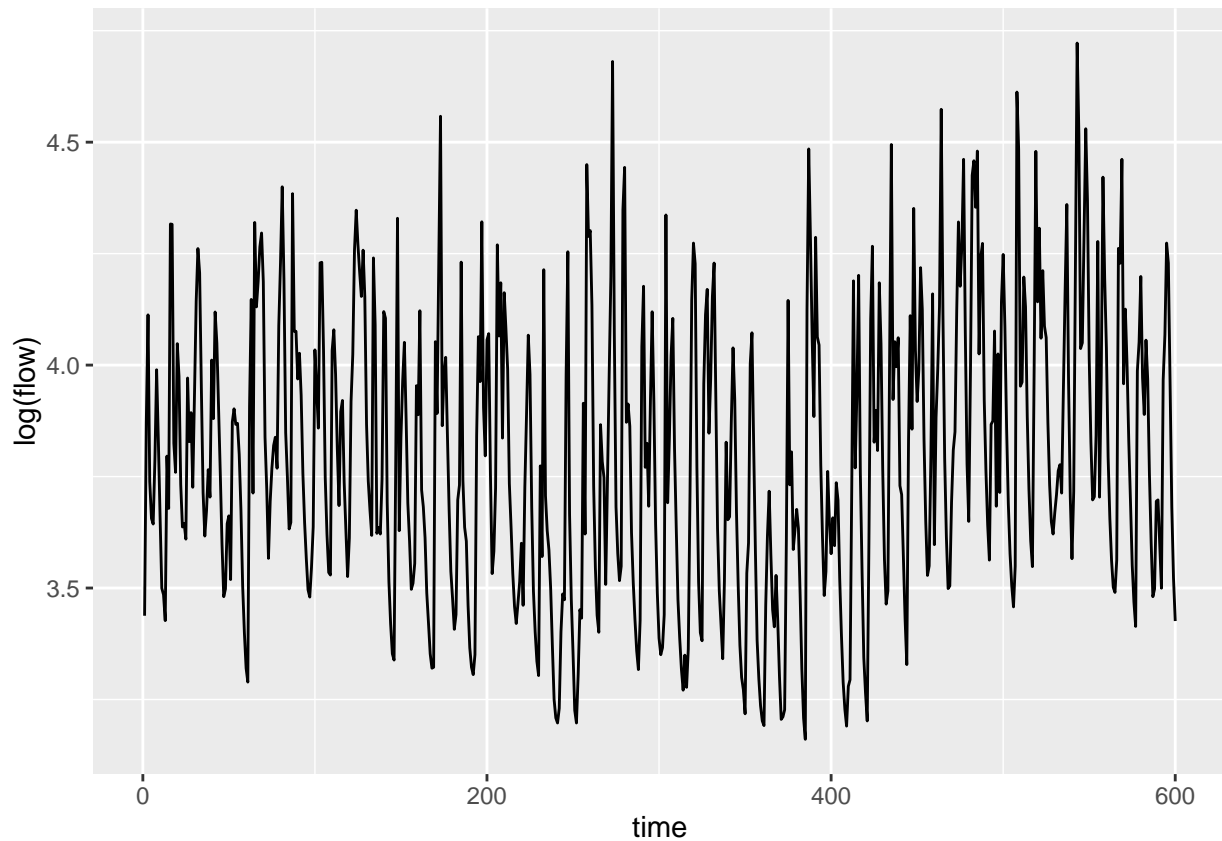
The data.frame `flow_df` contains the average monthly river flow  $m^3/\text{sec}$

in the Mckenzie river at Mckenzie Bridge, Oregon. (I got this from <http://robjhyndman.com/tsdldata/askew/askew7.dat> who quotes the source: Hipel and Mcleod (1994))

The column `time` contains a simple time index, the number of months since the start of the record. The column `date` contains a decimal representation of the date, i.e. 1911.750 is October 1911.

1. Estimate the spectrum of the logarithm of flow. Make sure you show evidence you experimented with the amount of smoothing, but you need only show your final plot.

```
library(ggplot2)
load(url("http://stat565.cwick.co.nz/data/flow_df.rda"))
qplot(time, log(flow), data = flow_df, geom = "line")
```



```
attach(flow_df)
flow_log <- log(flow)
flow_log
```

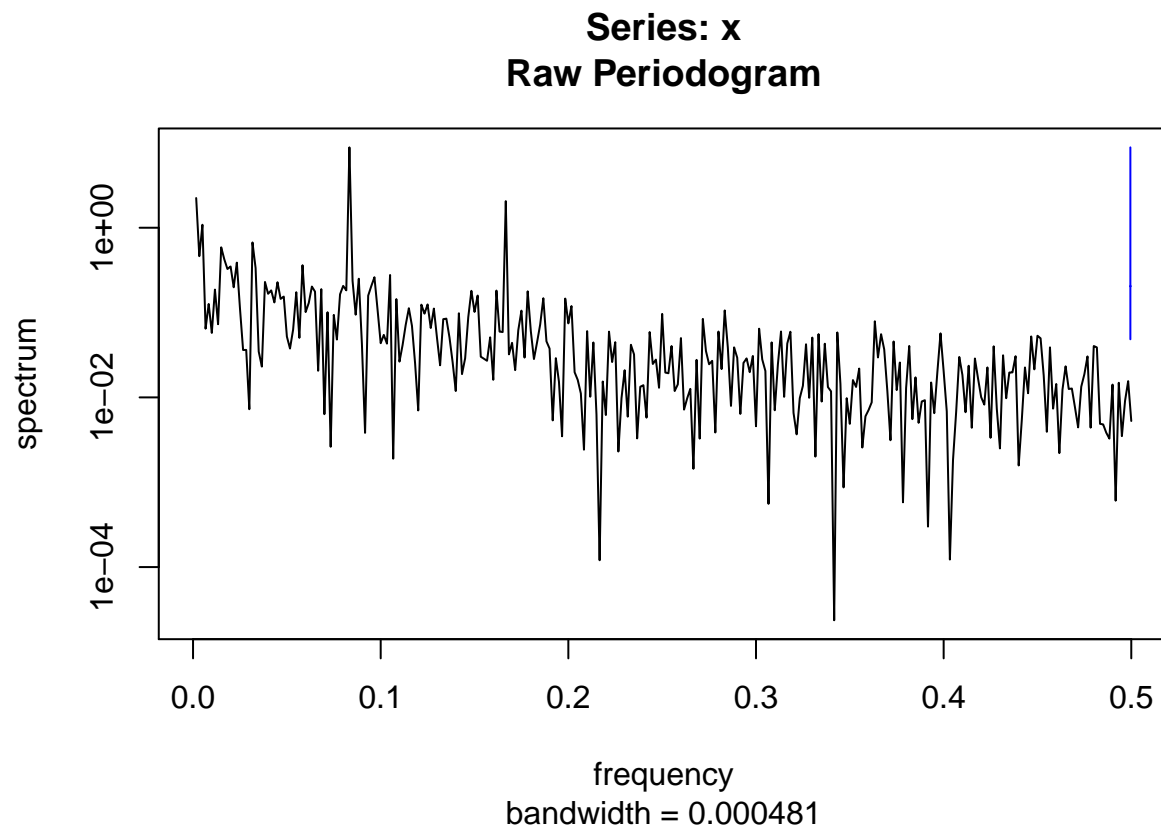
```
##      [1] 3.437857 3.869959 4.112640 3.738871 3.656075 3.643560 3.787502
##      [8] 3.989514 3.830809 3.658997 3.499606 3.484956 3.426878 3.795806
##     [15] 3.678501 4.316506 4.315751 3.824649 3.759531 4.048039 3.975261
##     [22] 3.732116 3.636873 3.645043 3.609660 3.970999 3.828350 3.893311
##     [29] 3.725995 3.932909 4.143214 4.261347 4.206770 3.947772 3.713640
##     [36] 3.616532 3.674200 3.765452 3.703926 4.011287 3.879950 4.119105
##     [43] 4.049520 3.904215 3.756230 3.598875 3.480606 3.497894 3.644303
##     [50] 3.661912 3.519090 3.872319 3.901930 3.867001 3.869368 3.799615
##     [57] 3.675635 3.505575 3.399836 3.320190 3.289002 3.907633 4.147250
##     [64] 3.712950 4.319901 4.130550 4.187177 4.265333 4.296273 4.186747
##     [71] 3.838153 3.721209 3.566602 3.694113 3.768725 3.819691 3.838153
##     [78] 3.768725 4.088248 4.230113 4.399858 4.190612 3.848465 3.748923
##     [85] 3.632388 3.647258 4.384440 4.075824 4.075824 3.968861 4.026555
##     [92] 3.932355 3.756891 3.647258 3.564998 3.496177 3.479734 3.545581
##     [99] 3.636873 4.034101 3.946133 3.858673 4.229289 4.230525 4.032597
##    [106] 3.747587 3.621088 3.534903 3.529107 4.032597 4.079185 3.978975
##    [113] 3.813462 3.684917 3.895617 3.920632 3.737524 3.629387 3.525778
##    [120] 3.611955 3.918383 4.024026 4.257745 4.347392 4.270490 4.208033
##    [127] 4.153943 4.257745 4.153498 3.867001 3.739544 3.674200 3.618055
##    [134] 4.240361 4.090146 3.622603 3.636127 3.621088 3.743574 4.120025
##    [141] 4.105197 3.692705 3.514044 3.419492 3.352417 3.338446 3.798348
```



## [148] 4.329274 3.628636 3.838153 3.964572 4.051000 3.878781 3.690585  
 ## [155] 3.605052 3.497035 3.510664 3.555337 3.953766 3.888685 4.121864  
 ## [162] 3.719838 3.682069 3.612722 3.489288 3.421343 3.352417 3.320190  
 ## [169] 3.322234 4.052477 3.891579 4.236684 4.558073 3.864036 3.979504  
 ## [176] 4.017422 3.844838 3.674917 3.537379 3.476237 3.407372 3.440585  
 ## [183] 3.697629 3.730758 4.230936 3.742233 3.636127 3.605052 3.461241  
 ## [190] 3.366195 3.322234 3.305754 3.350434 3.842414 4.063732 3.962958  
 ## [197] 4.321407 3.905357 3.796442 4.056407 4.070523 3.758212 3.532424  
 ## [204] 3.582474 3.724630 4.269697 4.065192 4.184593 3.836323 4.162355  
 ## [211] 4.082056 3.991084 3.734147 3.636127 3.532424 3.458570 3.420418  
 ## [218] 3.465673 3.514044 3.600422 3.461241 3.725995 3.891000 4.067133  
 ## [225] 3.978975 3.681356 3.493600 3.401727 3.335424 3.303676 3.773940  
 ## [232] 3.570591 4.213913 3.706710 3.628636 3.585620 3.499606 3.371071  
 ## [239] 3.250247 3.208781 3.197276 3.230286 3.401727 3.486690 3.473608  
 ## [246] 3.976323 4.254132 3.665542 3.474485 3.356374 3.225800 3.197276  
 ## [253] 3.305754 3.451415 3.432383 3.914436 3.621088 4.449706 4.288530  
 ## [260] 4.301657 4.133730 3.728040 3.545581 3.438766 3.400781 3.866408  
 ## [267] 3.783646 3.747587 3.508125 3.735498 3.974197 4.197447 4.680824  
 ## [274] 4.071971 3.682069 3.577738 3.516569 3.548844 4.348491 4.443402  
 ## [281] 3.871730 3.912741 3.862845 3.624114 3.512357 3.429636 3.359330  
 ## [288] 3.317113 3.426878 4.041592 4.176801 3.770683 3.824649 3.683495  
 ## [295] 3.920632 4.119565 3.916131 3.642820 3.488423 3.387474 3.350434  
 ## [302] 3.366195 3.439677 4.336709 3.691294 3.835099 4.020983 4.104730  
 ## [309] 3.812835 3.633885 3.473608 3.387474 3.318138 3.270887 3.349441  
 ## [316] 3.277318 3.363260 3.715020 4.146355 4.273650 4.227225 3.771988  
 ## [323] 3.513201 3.399836 3.381716 3.957560 4.113103 4.169823 3.847258  
 ## [330] 3.967791 4.153498 4.228875 3.925673 3.633885 3.494460 3.419492  
 ## [337] 3.341454 3.515728 3.827116 3.653146 3.659726 3.872319 4.038603  
 ## [344] 3.922876 3.670601 3.506425 3.371071 3.299504 3.271962 3.217895  
 ## [351] 3.533251 3.601195 4.002014 4.072453 3.819070 3.582474 3.382677  
 ## [358] 3.296366 3.235871 3.201892 3.191472 3.461241 3.616532 3.717088  
 ## [365] 3.573774 3.451415 3.412984 3.528276 3.425038 3.300548 3.205345  
 ## [372] 3.211069 3.226923 3.609660 4.145011 3.731436 3.805302 3.586404  
 ## [379] 3.630137 3.676351 3.633885 3.484956 3.317113 3.208781 3.160734  
 ## [386] 4.125073 4.484810 4.284246 4.059344 3.884617 4.286585 4.063246  
 ## [393] 4.045069 3.776538 3.612722 3.483220 3.537379 3.761509 3.650941  
 ## [400] 3.576947 3.657538 3.594995 3.736176 3.696927 3.521602 3.394149  
 ## [407] 3.289002 3.228046 3.190308 3.279452 3.294265 3.726676 4.188895  
 ## [414] 3.769378 3.937888 4.201270 3.757552 3.534903 3.344458 3.264415  
 ## [421] 3.201892 3.611191 4.103327 4.266525 3.827116 3.898492 3.808448  
 ## [428] 4.184593 4.047545 3.766107 3.564197 3.463903 3.493600 4.011800  
 ## [435] 4.494662 3.922876 4.052477 3.996303 4.061296 3.729400 3.709488  
 ## [442] 3.571390 3.430552 3.328343 3.830809 4.110784 3.856280 4.351415  
 ## [449] 4.034101 3.918383 3.990037 4.218509 4.124615 3.832038 3.650206  
 ## [456] 3.528276 3.549658 3.797077 4.159706 3.597323 3.882870 3.993697  
 ## [463] 4.131915 4.573683 4.115878 3.790705 3.615010 3.498750 3.504725  
 ## [470] 3.692705 3.809075 3.850275 4.120946 4.321031 4.176801 4.270095  
 ## [477] 4.461545 4.071971 3.778480 3.649471 3.987414 4.425601 4.458271  
 ## [484] 4.353967 4.480007 4.025037 4.243211 4.272862 3.921193 3.762167  
 ## [491] 3.641337 3.562593 3.867001 3.874674 4.076306 3.683495 4.025037  
 ## [498] 3.714330 4.141414 4.248079 4.086821 3.858076 3.689172 3.576156  
 ## [505] 3.504725 3.457679 3.548844 4.612374 4.484810 3.952679 3.961881  
 ## [512] 4.197447 4.130550 3.884035 3.720523 3.605821 3.548030 3.991084  
 ## [519] 4.479365 4.141864 4.307013 4.060321 4.211818 4.087298 4.061785

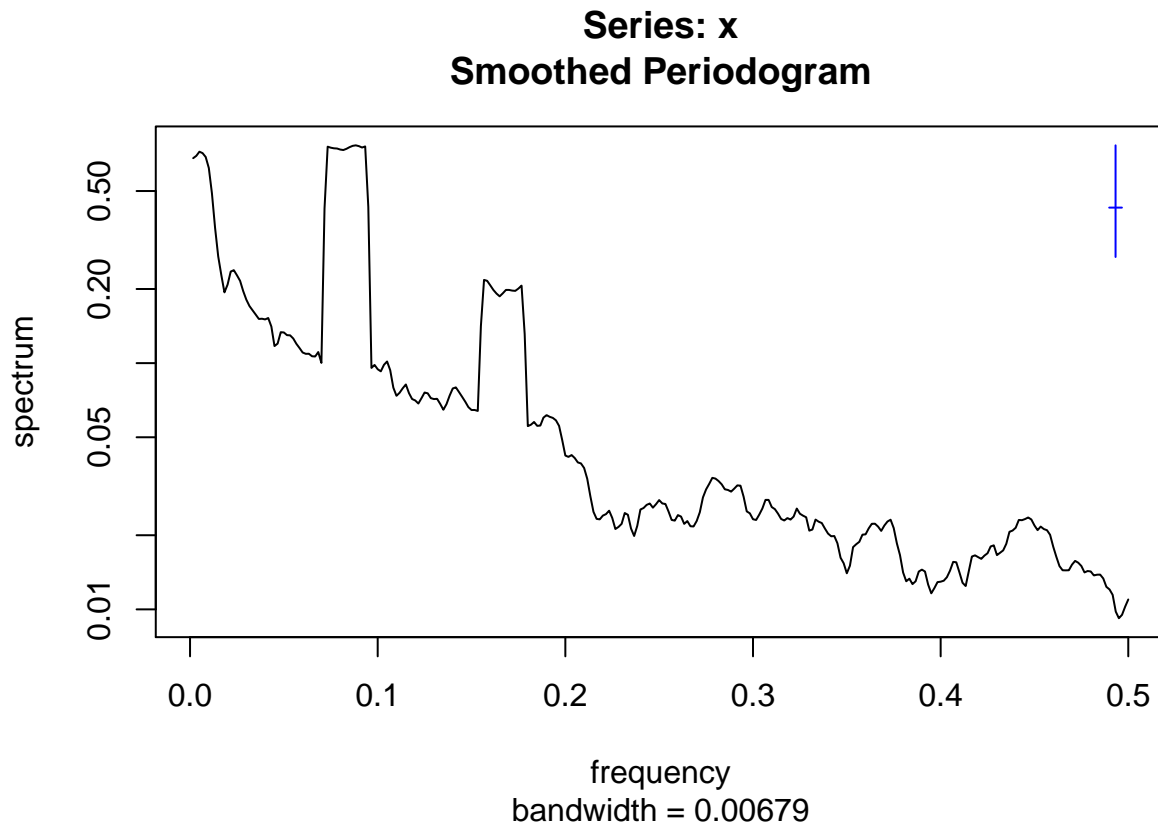
```
## [526] 3.871141 3.731436 3.648734 3.621088 3.669156 3.709488 3.762824
## [533] 3.776538 3.712950 3.945585 4.163678 4.360136 3.966182 3.703926
## [540] 3.565799 3.716400 4.219343 4.722223 4.516868 4.037104 4.050015
## [547] 4.290083 4.530080 4.368782 4.015381 3.827116 3.697629 3.704622
## [554] 3.843020 4.277195 3.703926 3.973131 4.421526 4.195316 4.038603
## [561] 3.793895 3.656075 3.561790 3.501314 3.490151 3.562593 4.261746
## [568] 4.228463 4.461545 3.958103 4.125530 4.000977 3.828965 3.693409
## [575] 3.551283 3.468326 3.413916 3.986889 4.051985 4.198723 3.971533
## [582] 3.889264 4.055917 3.968861 3.756230 3.603512 3.480606 3.497894
## [589] 3.694817 3.697629 3.597323 3.499606 3.966718 4.068587 4.273650
## [596] 4.228875 3.961881 3.682069 3.522441 3.425958
```

```
spectrum(flow_log)
```



```
# freq = 1, corresponds to 1 cycle per year
```

```
## try some spans
# spectrum(flow_log, taper = 0)
# spectrum(flow_log, spans = 5, taper = 0)
# spectrum(flow_log, spans = 10, taper = 0)
spectrum(flow_log, spans = 15, taper = 0)
```



```
# looks good
# spectrum(flow_log, spans = 20, taper = 0)
# spectrum(flow_log, spans = 23, taper = 0)
# spectrum(flow_log, spans = 50, taper = 0)
```

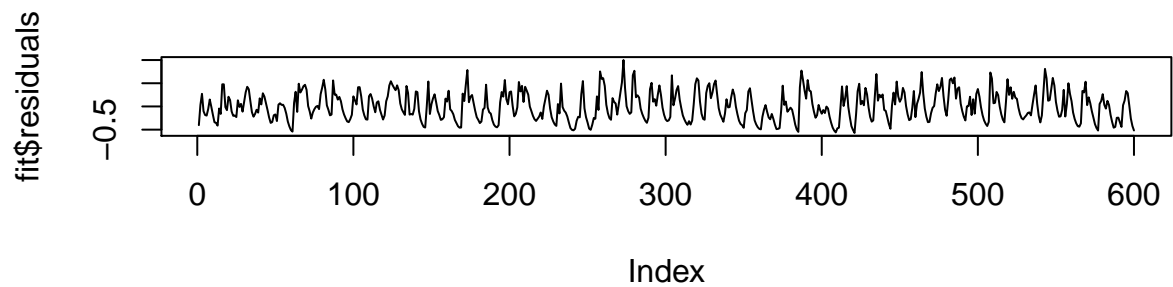
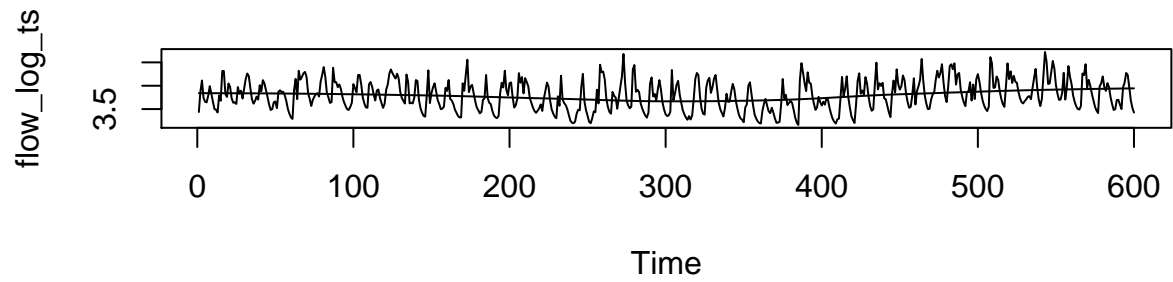
2. Fit a smooth trend to the logarithm of flow, and estimate the spectrum of the residuals. How does this spectrum differ from the one in part 1.?

```
attach(flow_df)
```

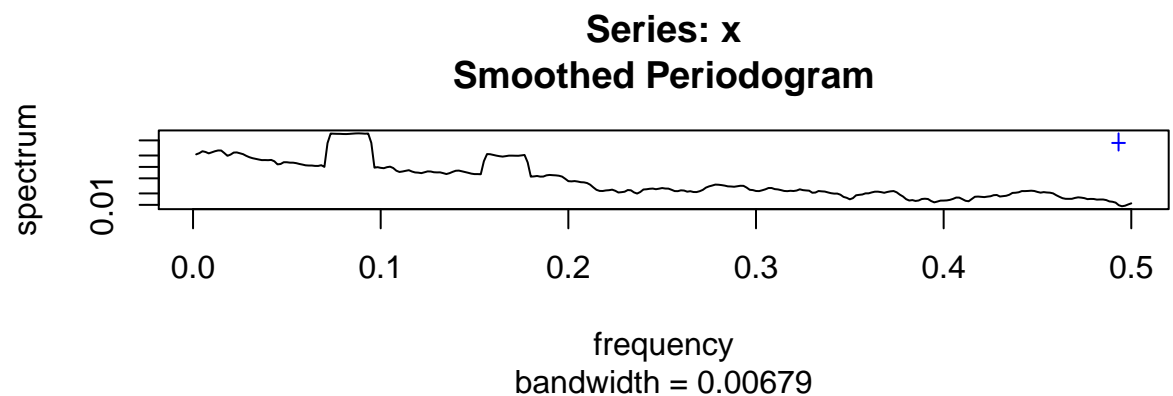
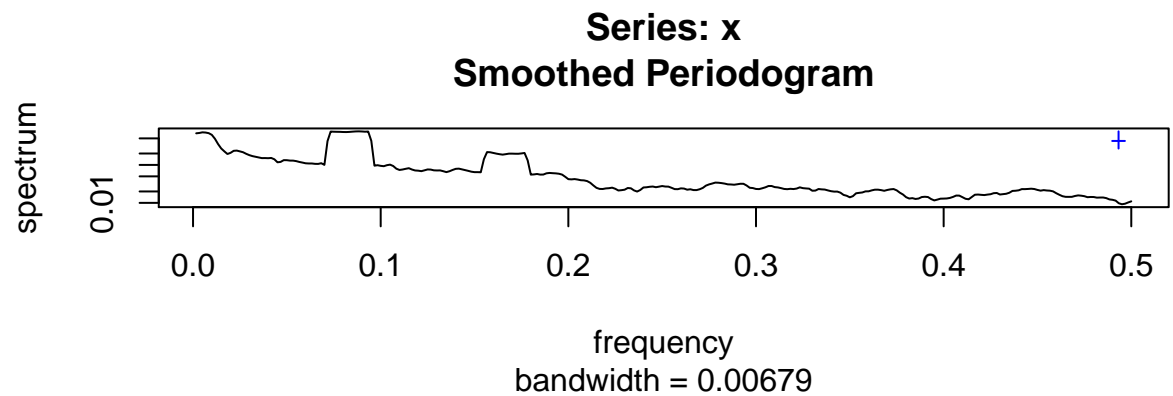
```
## The following objects are masked from flow_df (pos = 3):
##
##     date, flow, time
```

```
fit <- loess(flow_log ~ time, method = "loess")

par(mfrow = c(2,1))
flow_log_ts <- ts(flow_log, start = 1, frequency = 1)
plot(flow_log_ts)
lines(fit$fitted)
plot(fit$residuals, type = "l")
```



```
par(mfrow = c(2,1))
spectrum(flow_log, spans = 15, taper = 0)
spectrum(fit$residuals, spans = 15, taper = 0)
```



```
str(flow_log_ts)
```

```
## Time-Series [1:600] from 1 to 600: 3.44 3.87 4.11 3.74 3.66 ...
```

3. Fit a harmonic regression to the residuals from 2 using the estimated spectrum to choose the number and frequencies of the periodic components.

```
# par(mfrow = c(1,1))
#
# periodic <- function(x, frequency = 1, order = 1){
#   do.call(cbind, lapply(1:order, function(ord){
#     cbind(cos(2*pi*ord*frequency*x), sin(2*pi*ord*frequency*x))
#   }))
# }
#
# fit_harm <- lm(fit$residuals ~ time + periodic(time, frequency = 1, order = 3) + periodic(time, frequency = 2, order = 3))
```

4. Examine the residuals from the harmonic regression using both the ACF/PACF and periodogram. Is there any evidence of remaining autocorrelation?