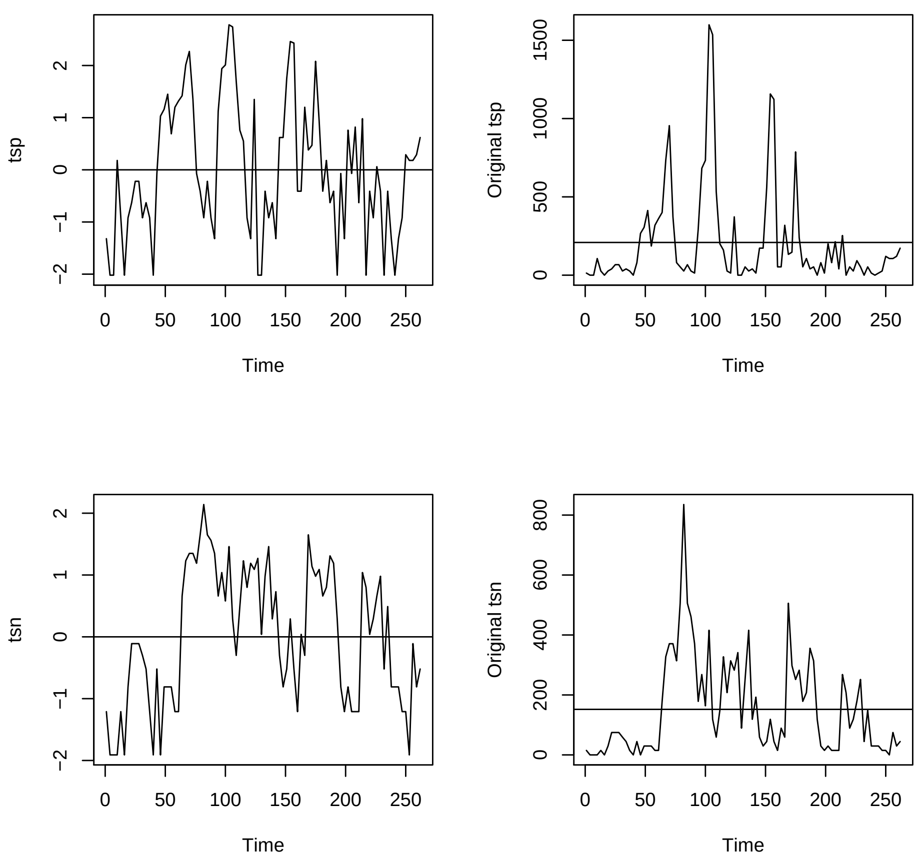
2/10/2010 36-402/608 ADA-II H. Seltman

Breakout #10 Results

The analyses here are from the first Chapter 15 dataset about nitrate runoff from two streams treated differently ("patch" vs. "nocut") and measured every 3 weeks for five years.

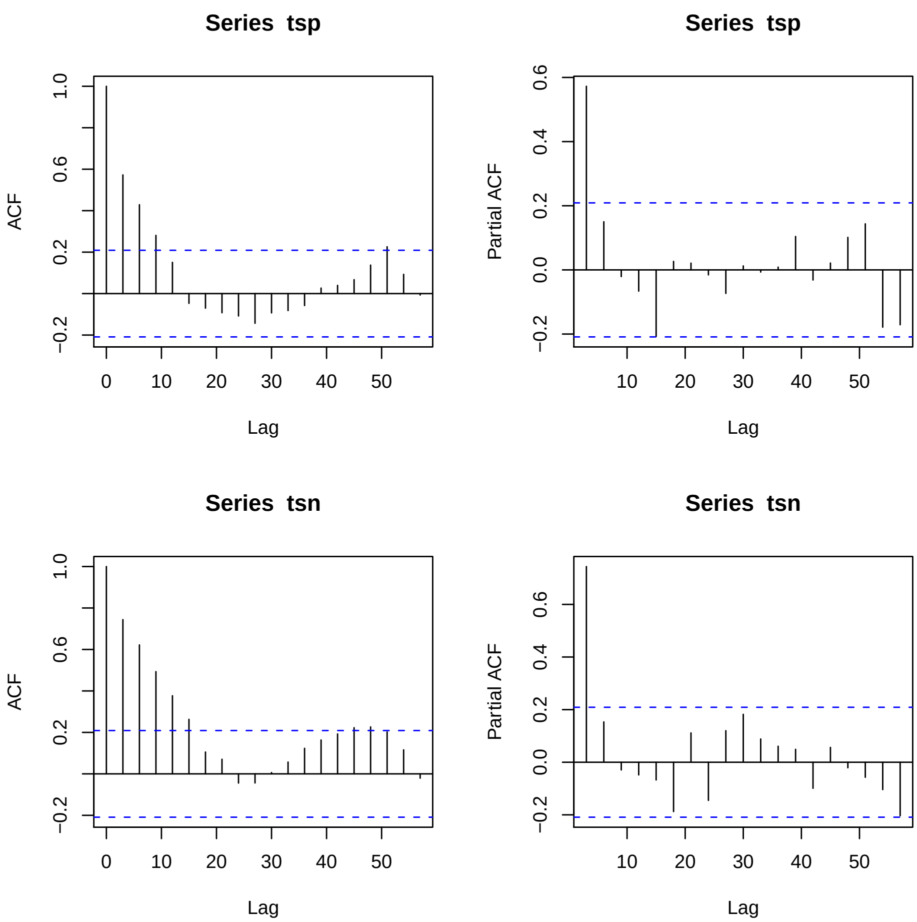
```
forest=read.csv("case1501.csv")
dim(forest) # 88 3
# I hate those all-upper-case names:
names(forest) = casefold(names(forest))
sapply(forest,class)
#     week     patch     nocut
#"integer" "numeric" "numeric"
summary(forest)
#          week               patch                nocut
# Min.   :  1.00   Min.   :-2.020000   Min.   :-1.910000
# 1st Qu.: 66.25   1st Qu.:-0.920000   1st Qu.:-0.810000
# Median :131.50   Median :-0.145000   Median :-0.035000
# Mean   :131.50   Mean   :-0.001023   Mean   :-0.001023
# 3rd Qu.:196.75   3rd Qu.: 0.942500   3rd Qu.: 0.995000
# Max.   :262.00   Max.   : 2.780000   Max.   : 2.140000

# Make time series objects (convenience for good labeling; not needed)
tsp = ts(forest$patch, start=1, deltat=3)
tsn = ts(forest$nocut, start=1, deltat=3)
```

```
# EDA of transformed data (from CD) and back-transformed
# original form:
par(mfrow=c(2,2))
plot(tsp); abline(h=0)
tmp=(exp(tsp)-1)*100
plot(tmp-min(tmp), ylab="Original tsp"); abline(h=mean(tmp-min(tmp)))
plot(tsn); abline(h=0)
tmp=(exp(tsn)-1)*100
plot(tmp-min(tmp), ylab="Original tsn"); abline(h=mean(tmp-min(tmp)))
par(mfrow=c(1,1))
```



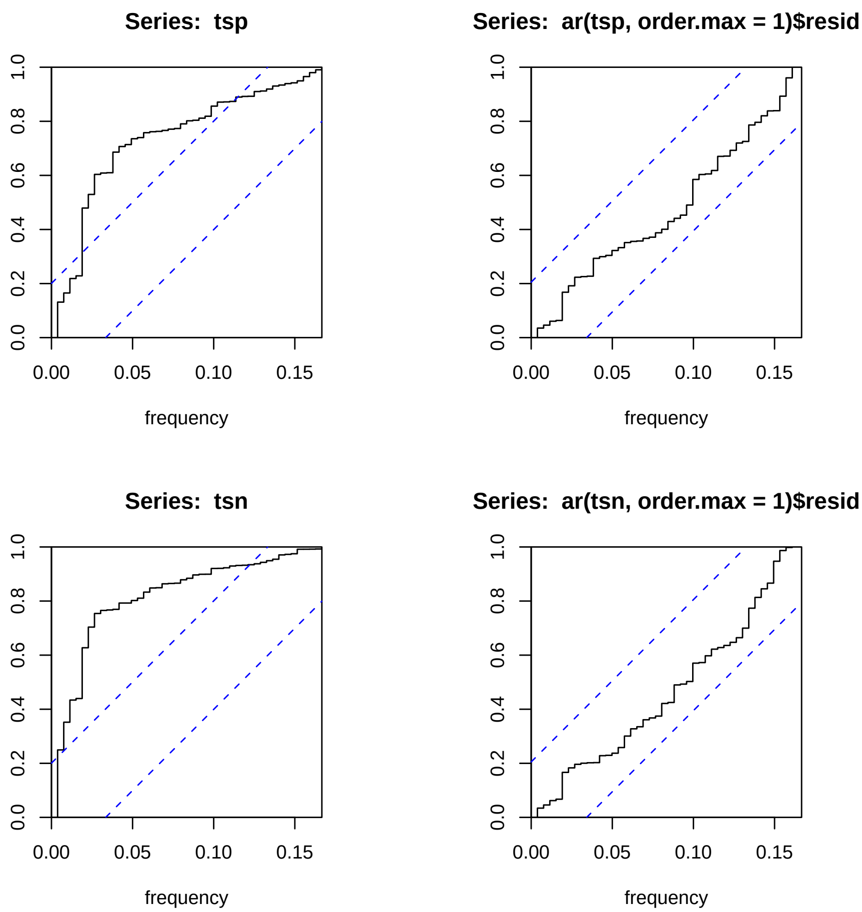
Question 1: What in the plots suggests serial correlation? What suggests the need for the transformation?

```
# Autocorrelation and partial autocorrelation plots:
par(mfrow=c(2,2))
acf(tsp); pacf(tsp)
acf(tsn); pacf(tsn)
par(mfrow=c(1,1))
```



Question 2: What do the plots tell us about which ARMA models will fit the data?

```
# Cummulative periodogram of data and residuals
# after correcting for AR(1):
par(mfrow=c(2,2))
cpgram(tsp)
cpgram(ar(tsp,order.max=1)$resid)
cpgram(tsn)
cpgram(ar(tsn,order.max=1)$resid)
par(mfrow=c(1,1))
```



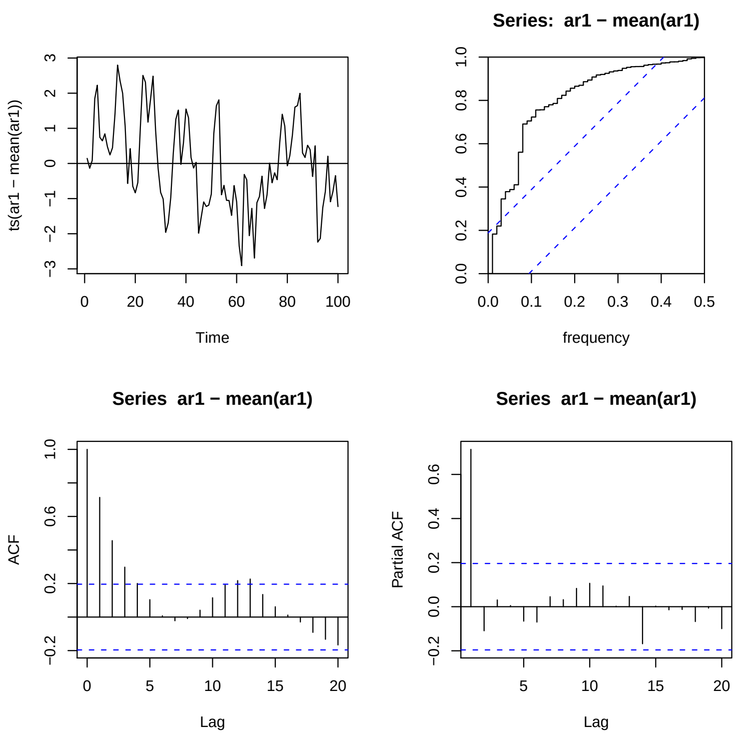
Question 3: Which plots are consistent with white noise?

```
# Standard AR(I)MA fitting function has AR coefficient as the first
# "order" value and MA as the last.  AR and MA coefficient (alpha
and beta from handout notes) along with $\sigma^2$ are estimated,
and aic is calculated.
arima(tsp, order=c(1,0,1))
# Coefficients:
#          ar1      ma1  intercept
#       0.7240  -0.2215    -0.0240
# s.e.  0.1106   0.1450     0.3008
# sigma^2 estimated as 1.052:  log likelihood = -127.31,  aic = 262.61

arima(tsp, order=c(1,0,0))
# Coefficients:
#          ar1  intercept
#       0.5746    -0.0114
# s.e.  0.0867     0.2561
# sigma^2 estimated as 1.076:  log likelihood = -128.29,  aic = 262.58

arima(tsp, order=c(0,0,1))
# Coefficients:
#          ma1  intercept
#       0.4089    -0.0025
# s.e.  0.0757     0.1673
# sigma^2 estimated as 1.248:  log likelihood = -134.71,  aic = 275.43

arima(tsp, order=c(0,0,2))
# Coefficients:
#          ma1     ma2  intercept
#       0.4815  0.2579    -0.0011
# s.e.  0.1062  0.0832     0.1958
# sigma^2 estimated as 1.13:  log likelihood = -130.39,  aic = 268.78

arima(tsp, order=c(2,0,0))
# Coefficients:
#          ar1     ar2  intercept
#       0.4822  0.1629    -0.0233
# s.e.  0.1043  0.1056     0.2999
# sigma^2 estimated as 1.047:  log likelihood = -127.12,  aic = 262.23
```

Question 4: Using a lower-is-better rule along with "parsimony" and a difference-of-aic 'gray zone" of about 2, which model(s) are most worthy of further study?

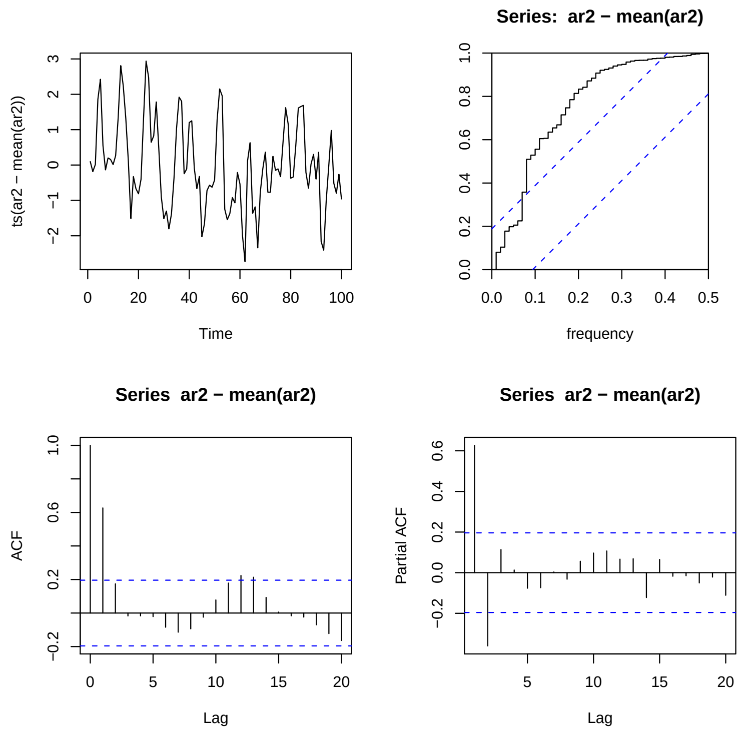
If time permits, study the code below to gain insight into the meaning of AR and MA.

```
# white noise:
N=100
barry = rnorm(N)

# Generate "ar1" time series as ar(1).
ar.p1 = 0.6
ar1 = c(barry[1], rep(NA, N-1))
for (i in 2:N) ar1[i] = ar.p1*ar1[i-1] + barry[i]
```



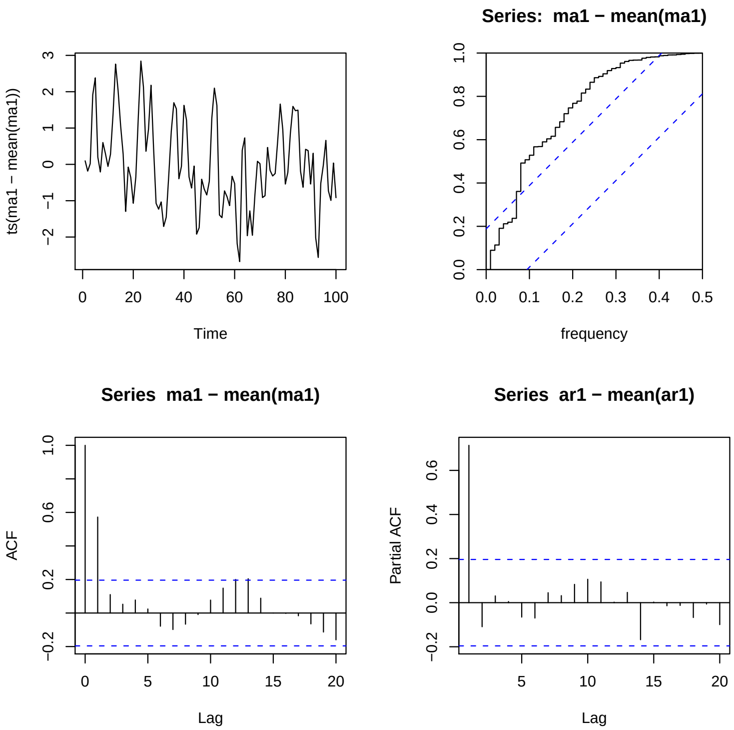

```
arima(ar1-mean(ar1), order=c(1,0,0))
# Coefficients:
#          ar1  intercept
#       0.6486    -0.0281
# s.e.  0.0753     0.2977
# sigma^2 estimated as 1.133:  log likelihood = -148.43,  aic = 302.86
```

```
# Generate ar2 time series
ar.p2 = c(0.7, -0.3)
ar2 = c(barry[1], rep(NA, N-1))
ar2[2] = ar.p2[1]*ar2[1] + barry[2]
for (i in 3:100) ar2[i] = ar.p2[1]*ar2[i-1] + ar.p2[2]*ar2[i-2] + barry[i]
```




```
arima(ar2-mean(ar2), order=c(2,0,0))
# Coefficients:
#           ar1      ar2  intercept
#        0.8532  -0.3600    -0.0052
# s.e.   0.0928   0.0923     0.1746
# sigma^2 estimated as 0.7875:  log likelihood = -130.34,  aic = 268.67
```

```
# Generate ma1 time series
ma.p1 = 0.7
ma1 = c(barry[1], rep(NA, N-1))
for (i in 2:N) ma1[i] = ma.p1*barry[i-1] + barry[i]
```




```
arima(ma1-mean(ma1), order=c(0,0,1))
# Coefficients:
#          ma1  intercept
#       0.8000    -0.0073
# s.e.  0.0631     0.1597
# sigma^2 estimated as 0.794:  log likelihood = -130.87,  aic = 267.74
```