- Approximate cut-offs for a 2-sided test that autocorrelation at lag k is 0:

- More interesting test with an ACF – actual method to do this developed later this semester:

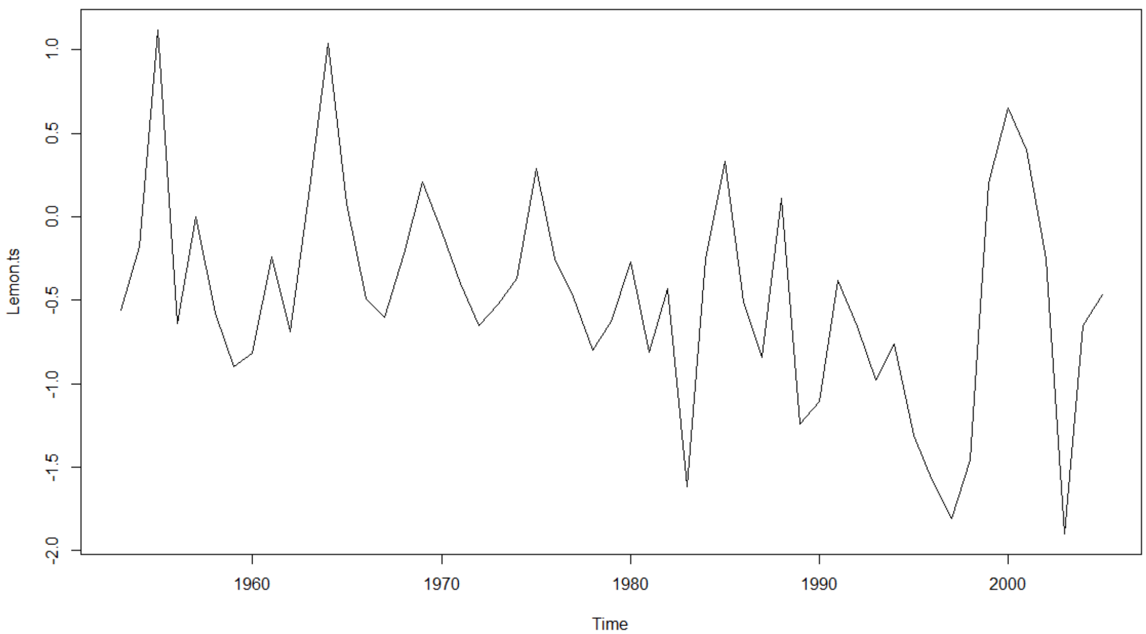
- Correlogram: plot of r_k vs k
  - also called ACF plot or even just ACF
  - `require(TSA); acf()` excludes k=0
  - Rarely are estimated correlations exactly 0
  - and if we use a test 100 times, we should reject the null hypothesis 5 times on average if testing at the 5% significance level.
    - So be a bit careful about getting excited when 1 correlation is "large" especially if it is a higher lag value and there is no reason that the process should maintain that level of dependence
    - Look for patterns using boundaries "approximately"

ACF interpretation "rules":

A new example:

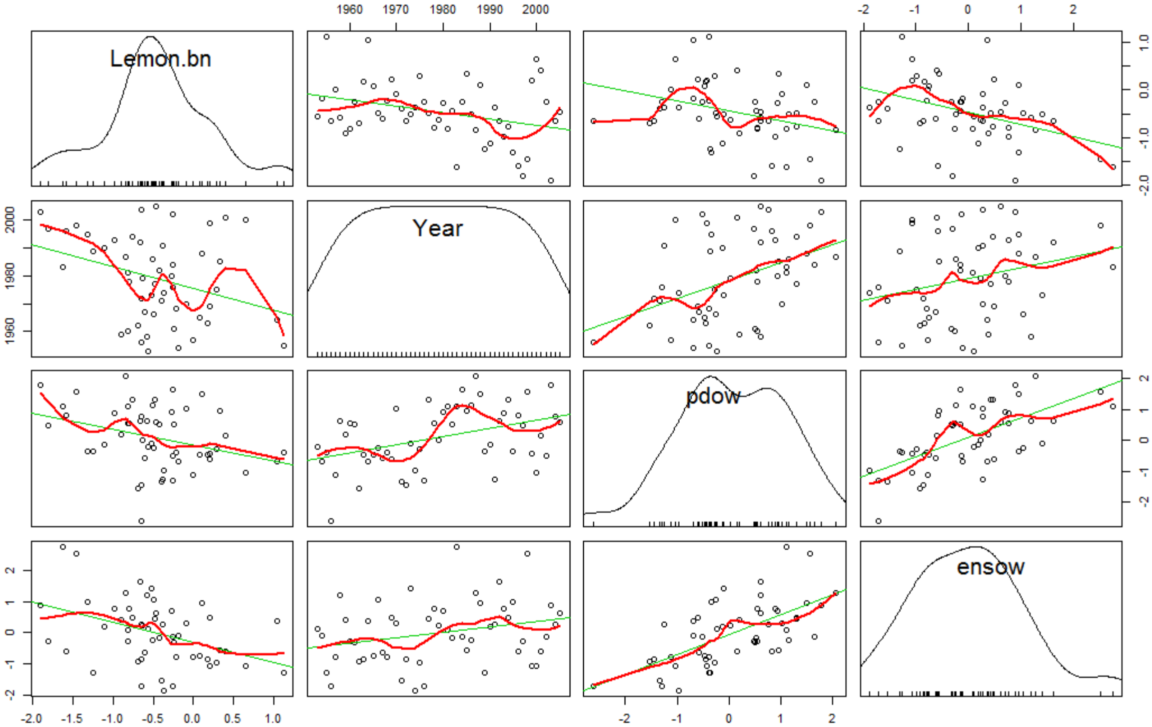
Winter Mass Balance (MB) of the Lemon Glacier in Alaska. MB is measured in meters of Water Equivalent based on the difference between accumulation and ablation (sublimation and melting) of the glacier measured by a field survey of the glacier. There are two climatic forcing functions that might play a role in the variation of the Mass Balance over time, the El Nino Seasonal Oscillation (ENSO) and the Pacific Decadal Oscillation (PDO). They are calculated here as water-year averaged values from their original monthly values and I converted them to z-scores so their units are both in SDs of the original variable. You can read more about the response variable for this data set at http://www.nichols.edu/departments/glacier/lemon.html but it might or might not be helpful. In the data set, Lemon.bn is the Lemon Glacier Mass Balance, pdow is the PDO for each year, and ensow is the ENSO for each year.

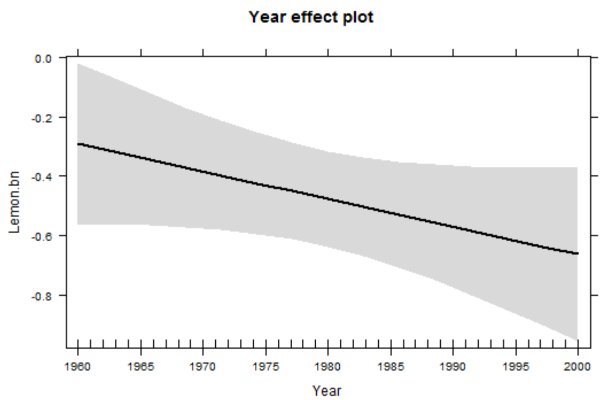
```
> MBdata<-read.csv("http://dl.dropboxusercontent.com/u/77307195/MB.csv",header=T)
>  LemonD<-MBdata[MBdata$Year>1952,]
>  Lemon.ts<-ts(LemonD$Lemon.bn,start=1953)
>  plot(Lemon.ts)
```

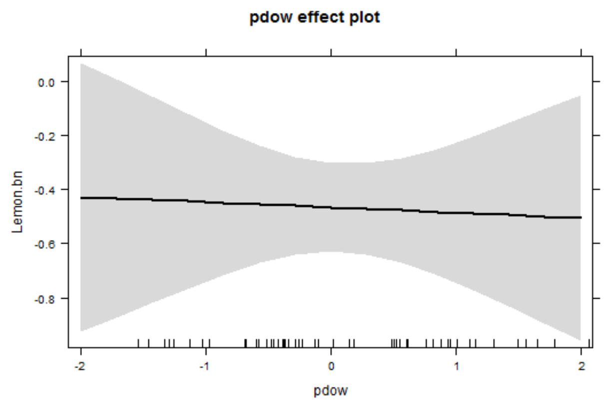


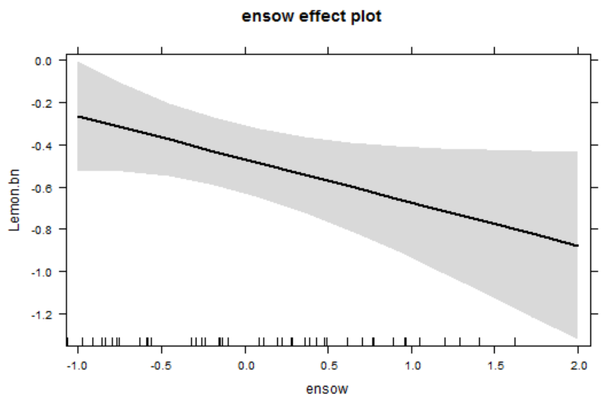

```
> require(car)
>  scatterplotMatrix(~Lemon.bn+Year+pdow+ensow,data=LemonD,ellipse=F,spread=F)
>  m1<-lm(Lemon.bn~Year+pdow+ensow,data=LemonD)
>  summary(m1)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 17.967028  11.412394   1.574   0.1218
Year        -0.009315   0.005769  -1.615   0.1128
pdow        -0.019356   0.111066  -0.174   0.8624
ensow       -0.204469   0.103244  -1.980   0.0533 .

Residual standard error: 0.5826 on 49 degrees of freedom
Multiple R-squared:  0.2123,   Adjusted R-squared:  0.164
F-statistic: 4.401 on 3 and 49 DF,  p-value: 0.008075

>  require(effects)
>  plot(allEffects(m1))
>  vif(m1)
    Year     pdow    ensow
1.215871 1.885767 1.660639
```

```
>  confint(m1)
                  2.5 %       97.5 %
(Intercept) -4.96703738 40.901092734
Year        -0.02090732  0.002277361
pdow        -0.24255060  0.203839063
ensow       -0.41194609  0.003007880
```



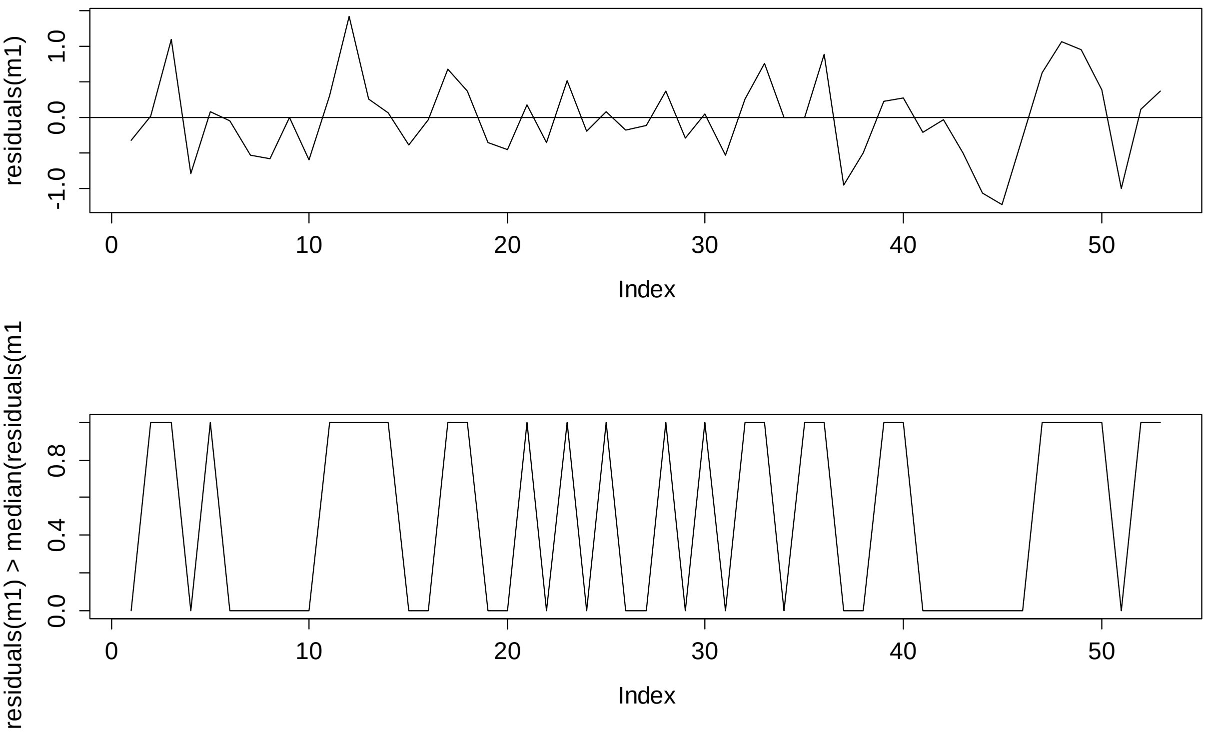






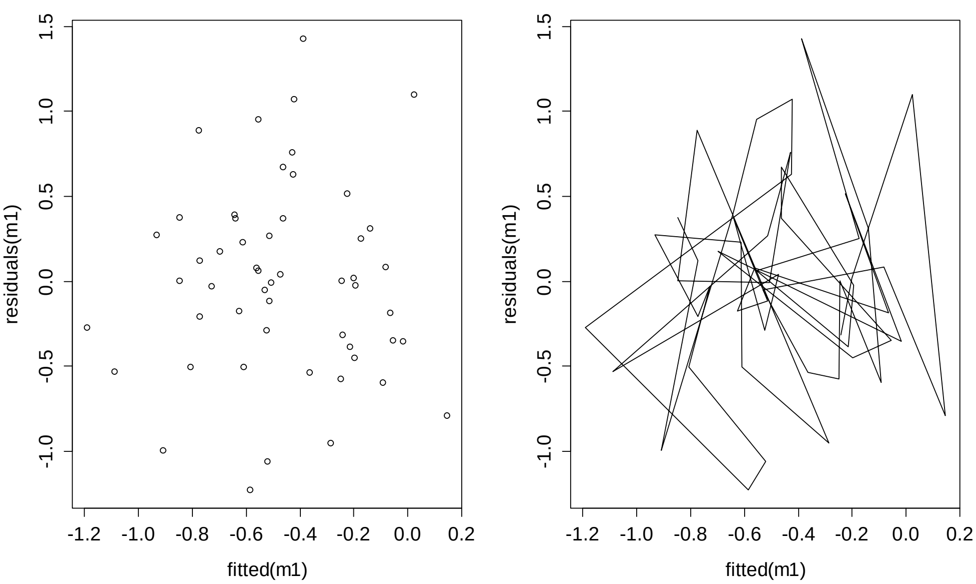
Patterns in residuals vs time?

```
> par(mfrow=c(2,1))
>  plot(residuals(m1),type="l")
>  abline(h=median(residuals(m1)))
>  plot(residuals(m1)>median(residuals(m1)),type="l")
```

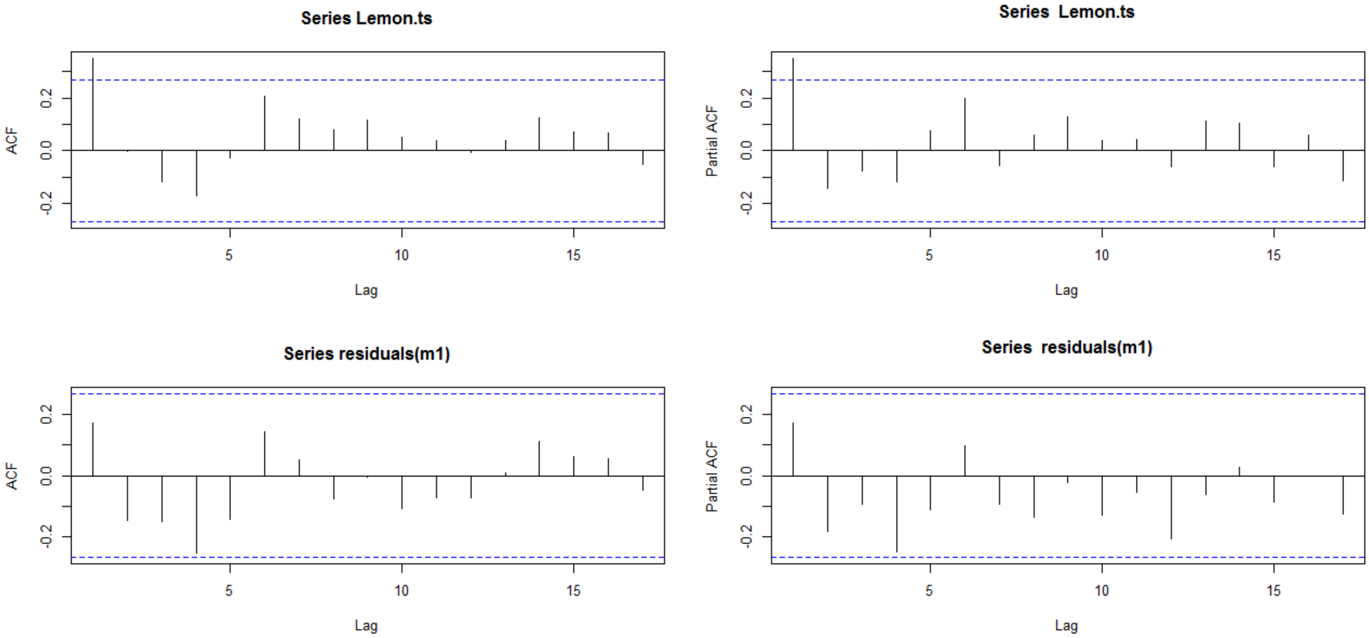


Residuals vs fitted values:

```
> par(mfrow=c(1,2))
>  plot(residuals(m1)~fitted(m1))
>  plot(residuals(m1)~fitted(m1),type="l")
```




```
> #ACFs of the mass balance data and residuals from m1 for mass balance data:
> require(TSA)
>   par(mfrow=c(2,2))
>  acf(Lemon.ts)
>  pacf(Lemon.ts)
>  acf(residuals(m1))
>  pacf(residuals(m1))
```



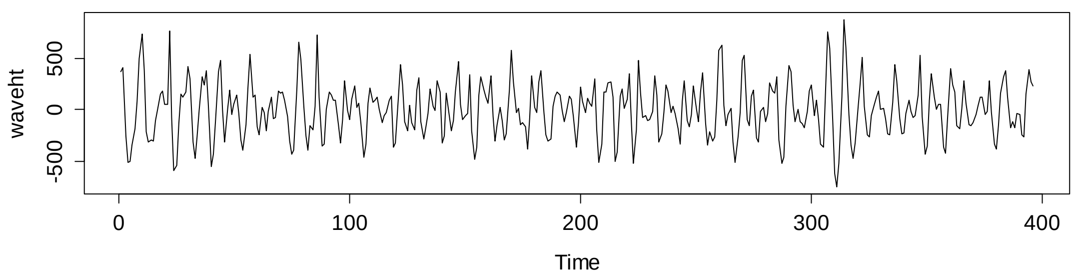
Runs Test for lack of independence (assuming median of residuals is 0 or the median is subtracted from the responses):

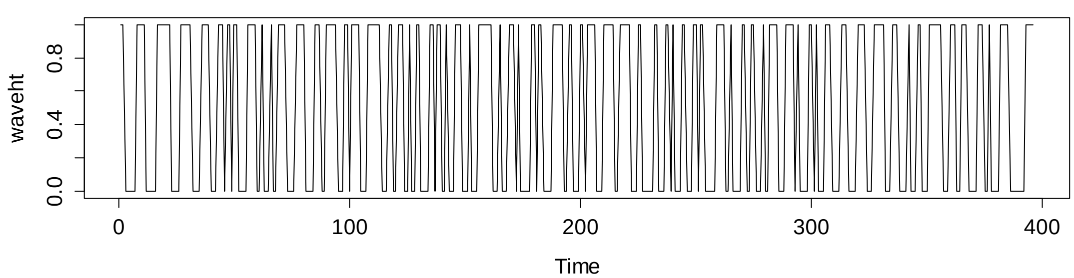
- See Statistical Sleuth (3^{rd} edition) Section 15.4.2 for the "Nonparametric Runs Test":

```
> runs(residuals(m1),k=median(residuals(m1)))
$pvalue
[1] 1
$observed.runs
[1] 28
$expected.runs
[1] 27.49057
$n1
[1] 27

$n2
[1] 26

$k
[1] 0.004230007

>  runs(LemonD$Lemon.bn,k=median(LemonD$Lemon.bn))
$pvalue
[1] 0.786
$observed.runs
[1] 26
$expected.runs
[1] 27.49057
$n1
[1] 27

$n2
[1] 26

$k
[1] -0.49

>  runs(LemonD$Lemon.bn,k=0)
$pvalue
[1] 0.238
$observed.runs
[1] 15
$expected.runs
[1] 18.43396
$n1
[1] 42

$n2
[1] 11

$k
[1] 0
```

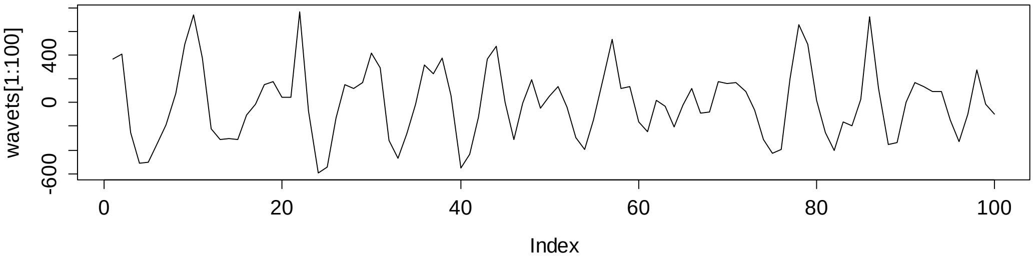
Wave data from p 34 in CM:

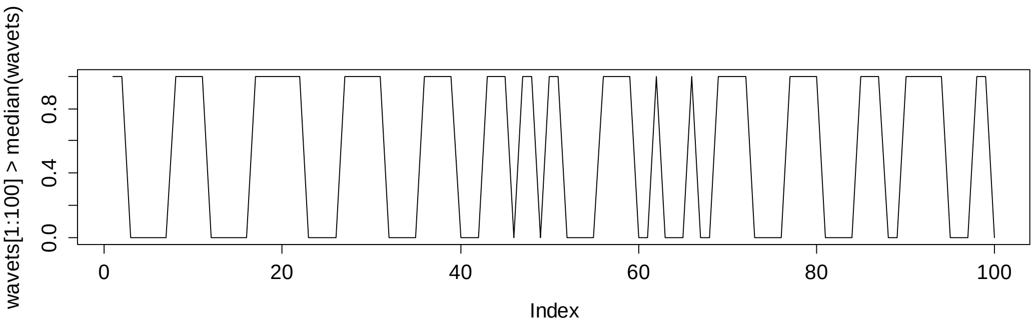
```
> wave.dat<-read.csv("https://dl.dropboxusercontent.com/u/77307195/wave.csv",
header=T)
>  wavets<-ts(wave.dat)
> par(mfrow=c(2,1))
>   plot(wavets)
>  summary(lm(wavets~time(wavets)))

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  -7.52162   26.87402  -0.280    0.780
time(wavets)  0.01737    0.11732   0.148    0.882

Residual standard error: 266.9 on 394 degrees of freedom
Multiple R-squared:  5.565e-05,       Adjusted R-squared:  -0.002482
F-statistic: 0.02193 on 1 and 394 DF,  p-value: 0.8824

>  plot(wavets>median(wavets))
>
>  runs(wavets,k=median(wavets))
$pvalue
[1] 1.61e-09

$observed.runs
[1] 139

$expected.runs
[1] 198.9949

$n1
[1] 199

$n2
[1] 197

$k
[1] -12
```






```
> plot(wavets[1:100],type="l")
>  plot(wavets[1:100]>median(wavets),type="l")
```






```
> runs(wavets[1:100],k=median(wavets[1:100]))
$pvalue
[1] 0.00017
$observed.runs
[1] 32
$expected.runs
[1] 51
$n1
[1] 50
$n2
[1] 50
$k
[1] -6.5
```

ACFs of the wave data:

```
> par(mfrow=c(1,2))
>  acf(wavets)
>  pacf(wavets)
```