Correlograms

Peter Claussen

9/5/2017

Libraries

nlme : Linear and Nonlinear Mixed Effects Models

```
library(nlme)
```

```
## Warning: package 'nlme' was built under R version 3.3.2
```

Data

```
load(file="autocorrelation.Rda")
attach(autocorrelation.dat)

load(file="sample.dat.Rda")
sample.pass14.dat <- sample.dat[sample.dat$PassNum==14,]
sample.pass15.dat <- sample.dat[sample.dat$PassNum==15,]
```

Let's reconsider the sample lag-1 autocorrelation coefficient r_1. Suppose we generalize this to an arbitrary lag distance k, by

$$r_k = \frac{\Sigma_{i=1}^{n-k}(y_i - \bar{y})(y_{i-k} - \bar{y})}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2}$$

and

$$\bar{y} = (\Sigma y_i)/n$$

is the sample mean.

Note that for lag 0, we have

$$r_0 = \frac{\Sigma_{i=1}^{n}(y_i - \bar{y})(y_i - \bar{y})}{\Sigma_{i=1}^{n}(y_i - \bar{y})^2} = 1$$

We can modify our autocorrelation function to work with an arbitrary lag,

```
auto.correlation <- function(univariate,k=1) {
  x.bar <- mean(univariate)
  ss <- sum((univariate-x.bar)^2)
  n <- length(univariate)
  lag.ss <- sum((univariate[(1+k):n]-x.bar)*(univariate[1:(n-k)]-x.bar))
  return(lag.ss/ss)
}
```

Example (Simulated) Data

We'll keep using the same simulated data as before.

First, test this function with white noise

```
auto.correlation(white.noise,k=1)
```

```
## [1] -0.01600082
```

```
auto.correlation(white.noise,k=2)
```

```
## [1] -0.0994322
```

```
auto.correlation(white.noise,k=3)
```

```
## [1] -0.1287819
```

We expect small r_k for all k, since white noise values should be independent.

```
auto.correlation(autoregressive,k=1)
```

```
## [1] 0.734061
```

```
auto.correlation(autoregressive,k=2)
```

```
## [1] 0.5145621
```

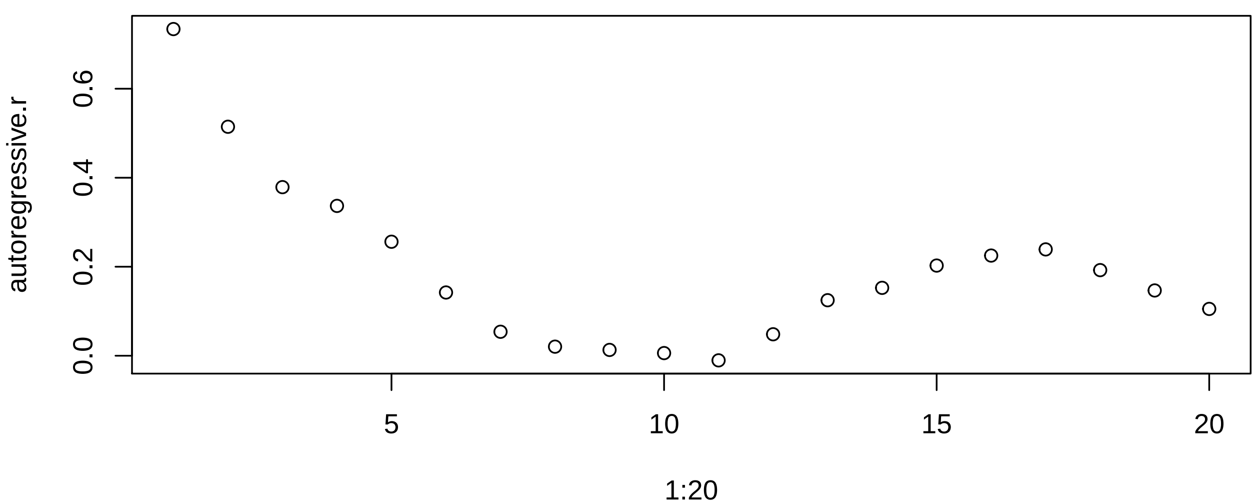
```
auto.correlation(autoregressive,k=3)
```

```
## [1] 0.378837
```

With simple autoregression, we see correlation coefficient approximating α at $k = 1$ and decreasing as we increase the gap k between values.

We might want to visualize how autocorrelation changes with lag; this will help us understand the process that creates a sequence of values.

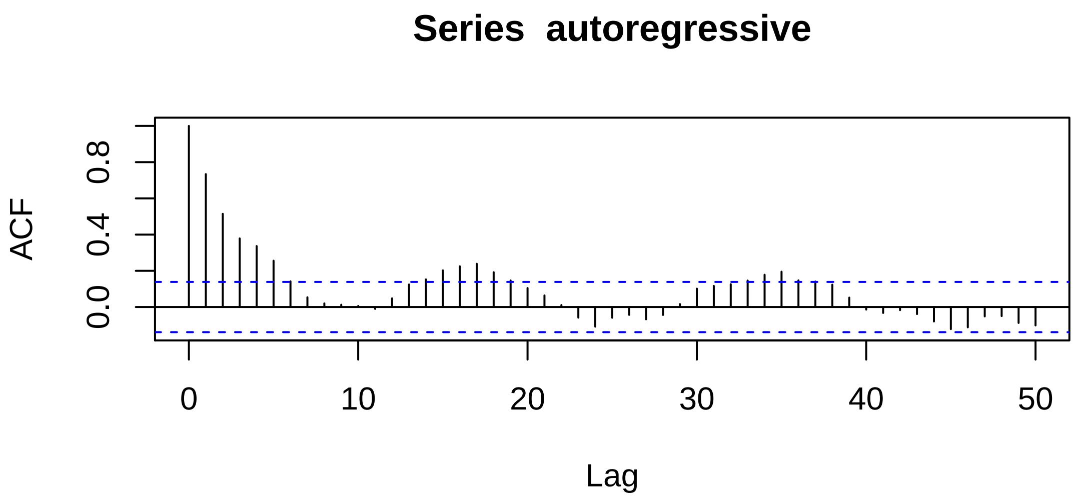
```
autoregressive.r <- rep(0,20)
for(i in 1:20) {
  autoregressive.r[i] <- auto.correlation(autoregressive,k=i)
}
plot(1:20,autoregressive.r)
```



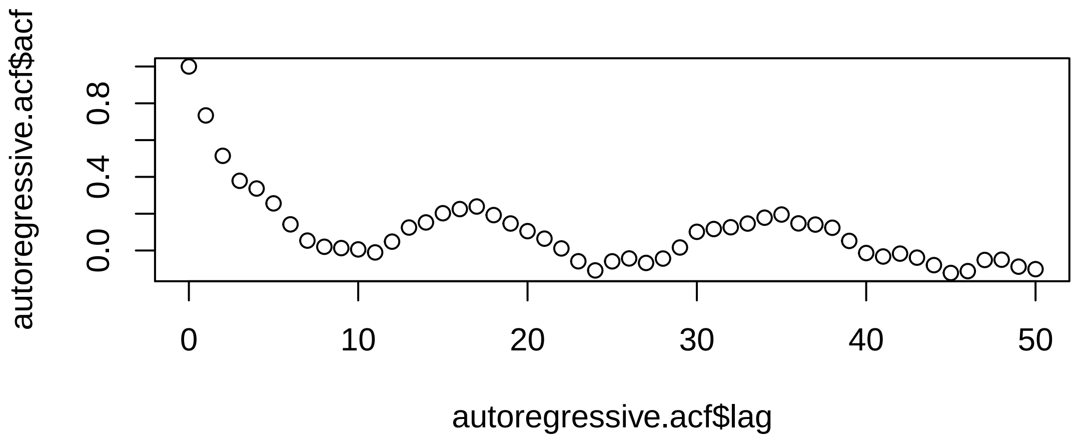
This is an example of an autocorrelation plot, sometimes called a correlogram.

We can also use the function `acf` from `nlme`

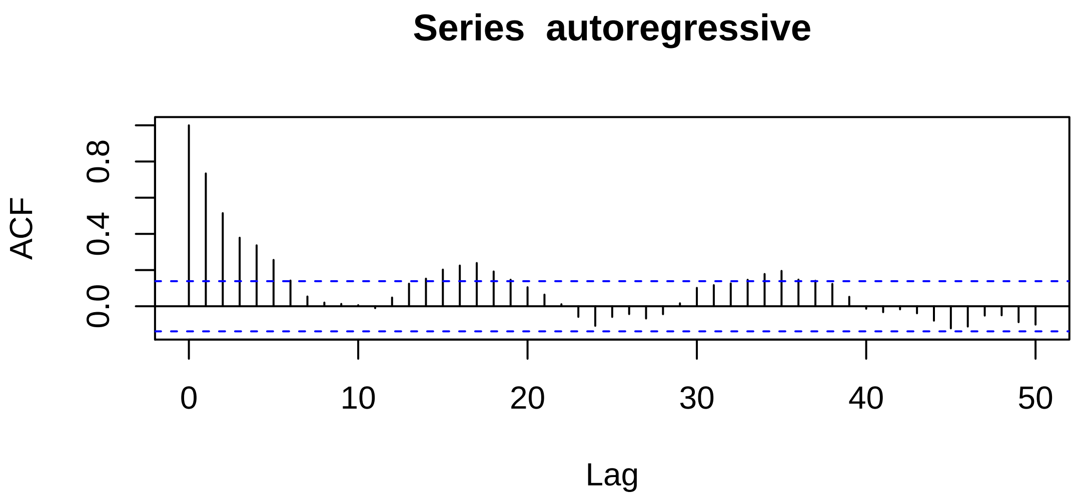
```
autoregressive.acf = acf(autoregressive,lag.max=50)
```



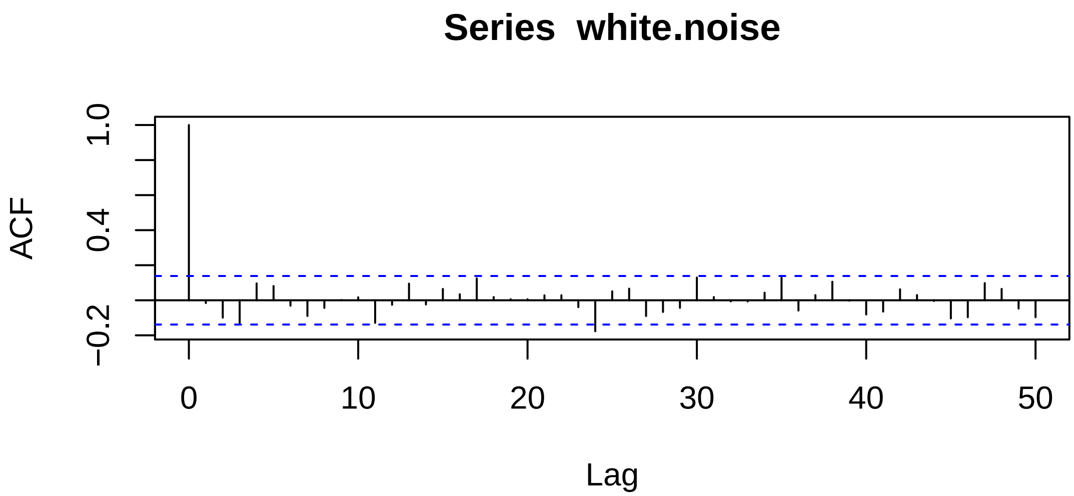

```
plot(autoregressive.acf$lag,autoregressive.acf$acf)
```



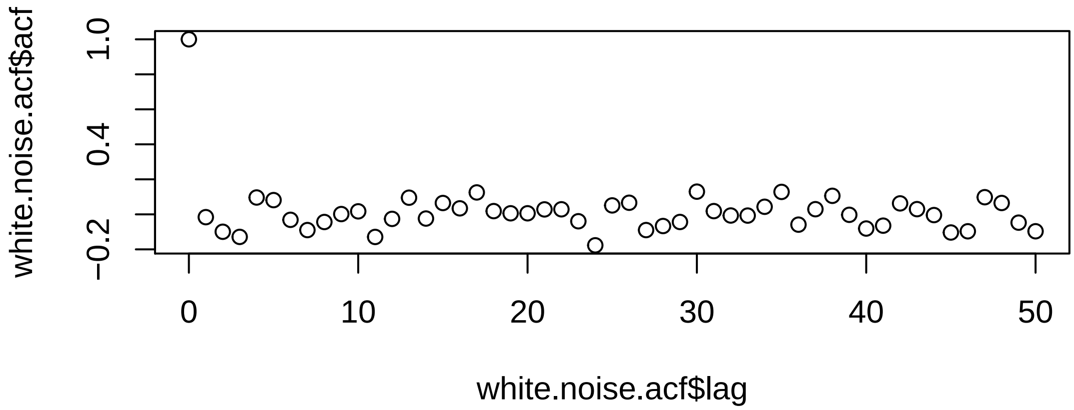

```
plot(autoregressive.acf)
```



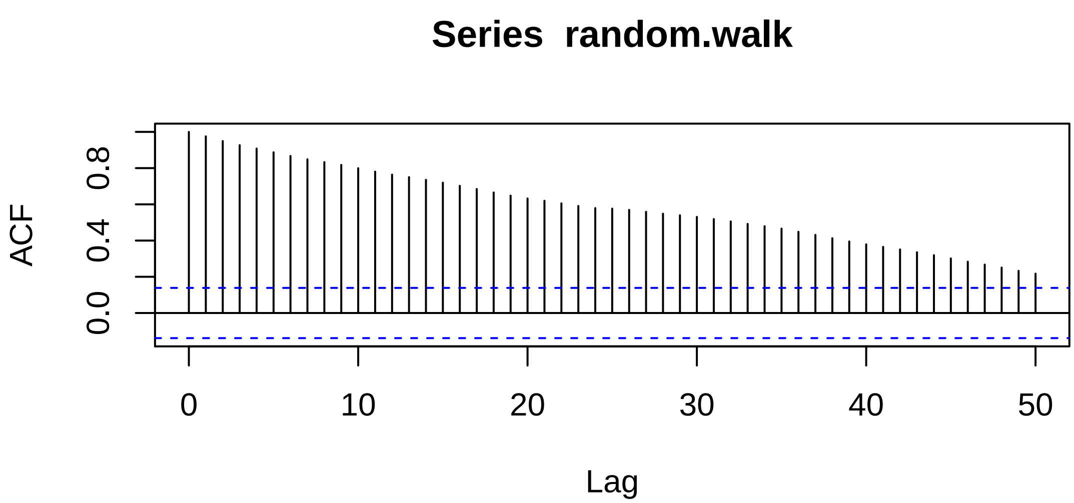

```
white.noise.acf = acf(white.noise,lag.max=50)
```



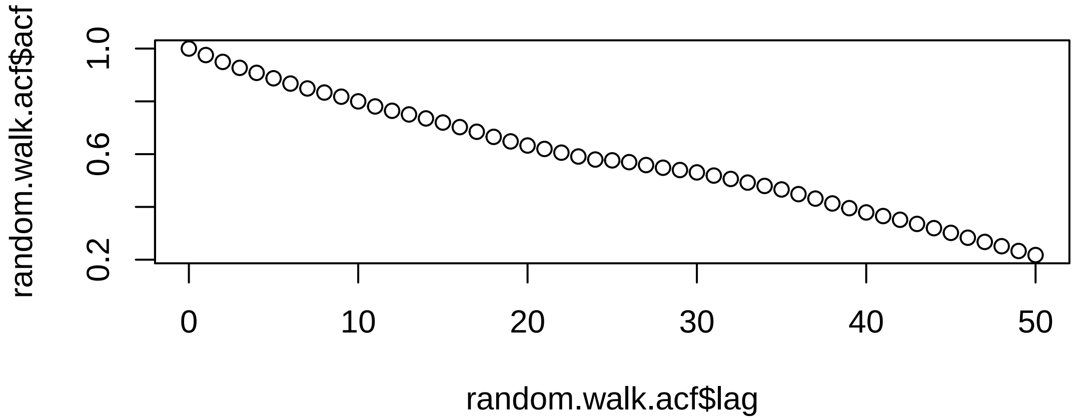

```
plot(white.noise.acf$lag,white.noise.acf$acf)
```



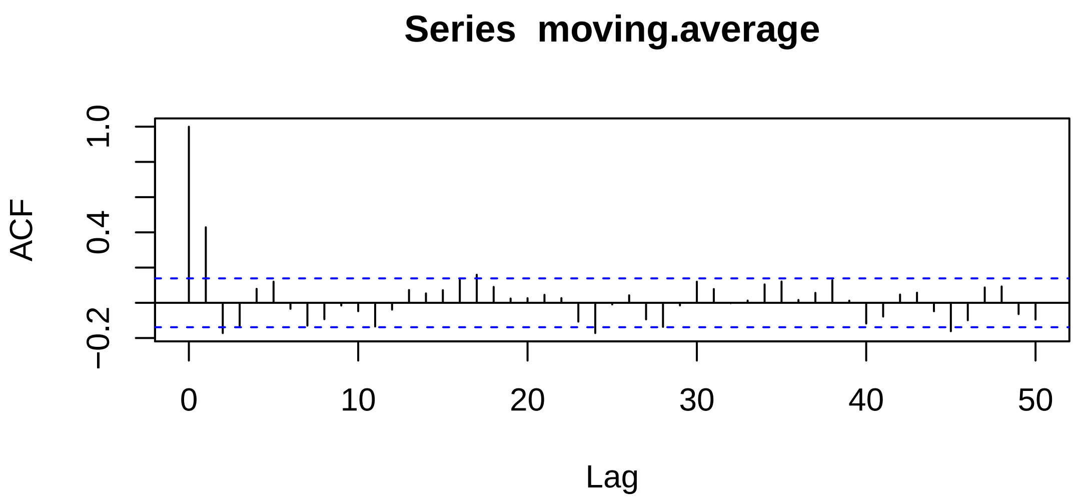

```
random.walk.acf = acf(random.walk,lag.max=50)
```



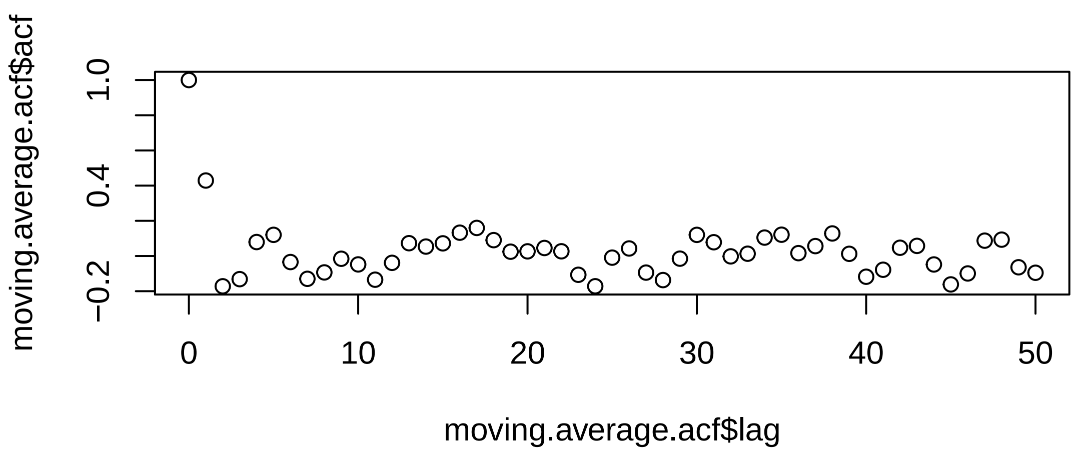

```
plot(random.walk.acf$lag,random.walk.acf$acf)
```



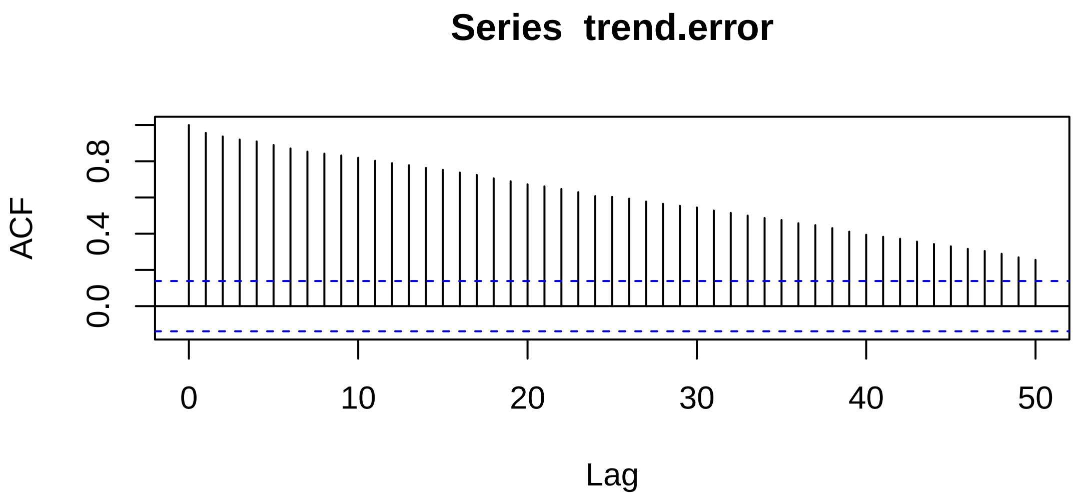

```
moving.average.acf = acf(moving.average,lag.max=50)
```



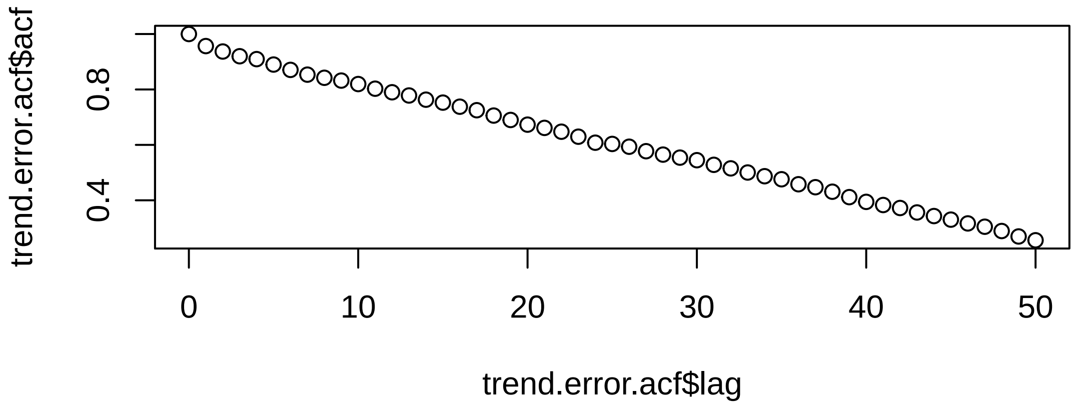

```
plot(moving.average.acf$lag,moving.average.acf$acf)
```




```
trend.error.acf = acf(trend.error,lag.max=50)
```



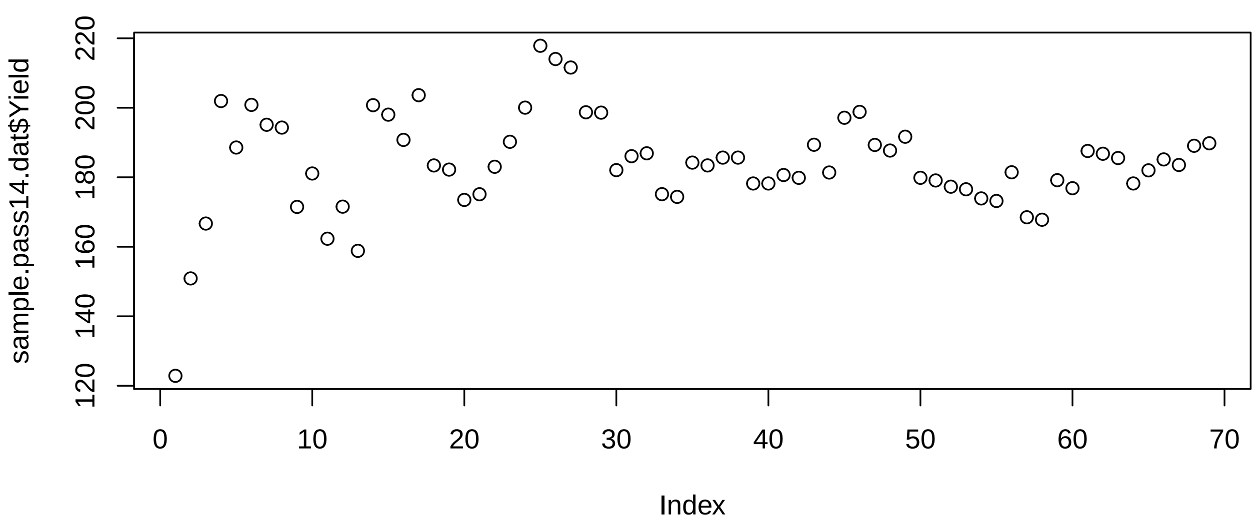

```
plot(trend.error.acf$lag,trend.error.acf$acf)
```



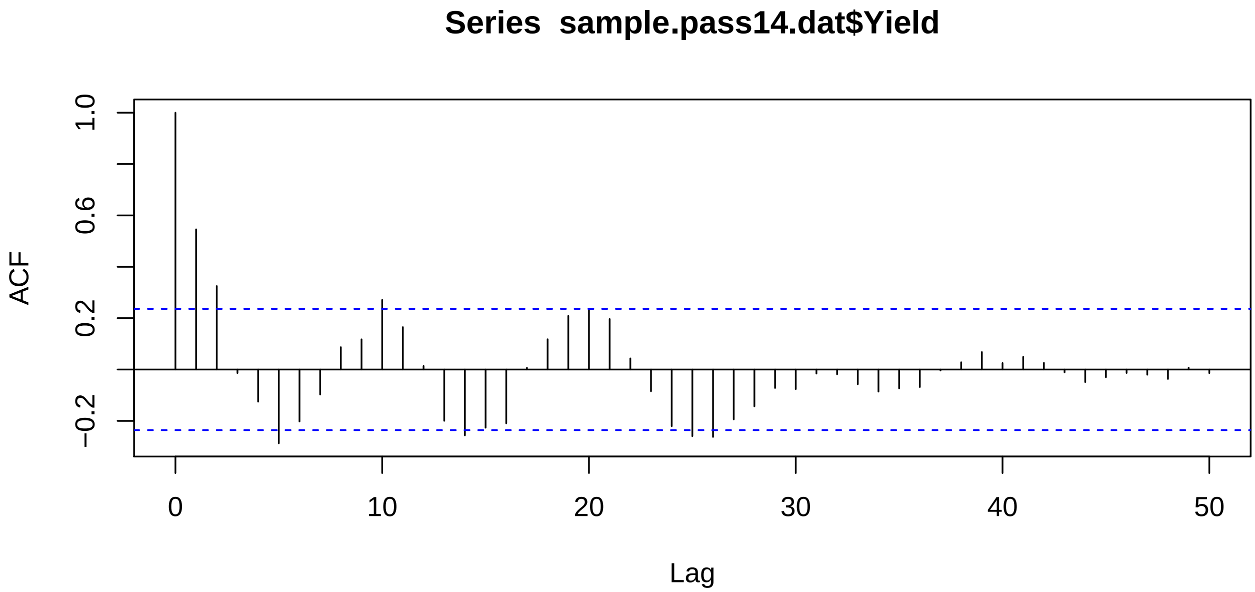
Autocorrelation plots from Yield Monitor Data

We'll consider autocorrelation by observation within a single pass.

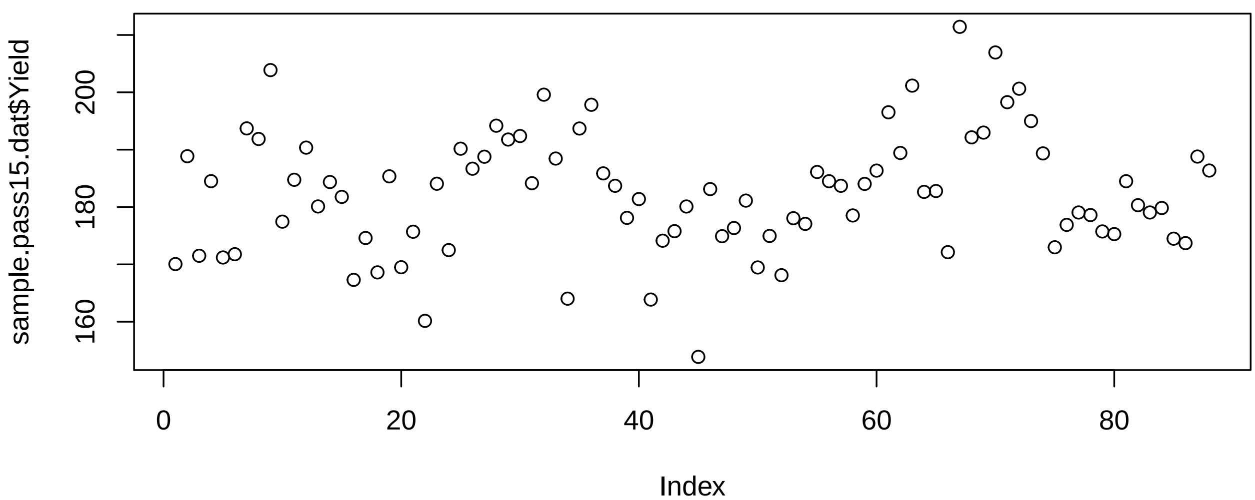
```
plot(sample.pass14.dat$Yield)
```



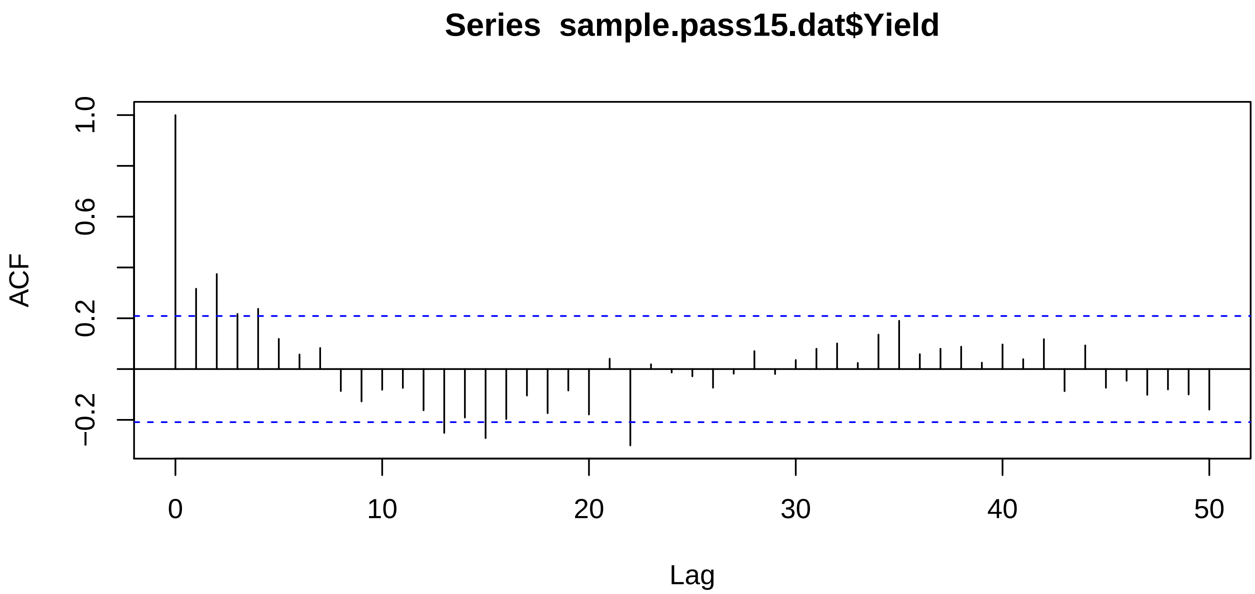
```
trend.error.acf = acf(sample.pass14.dat$Yield,lag.max=50)
```



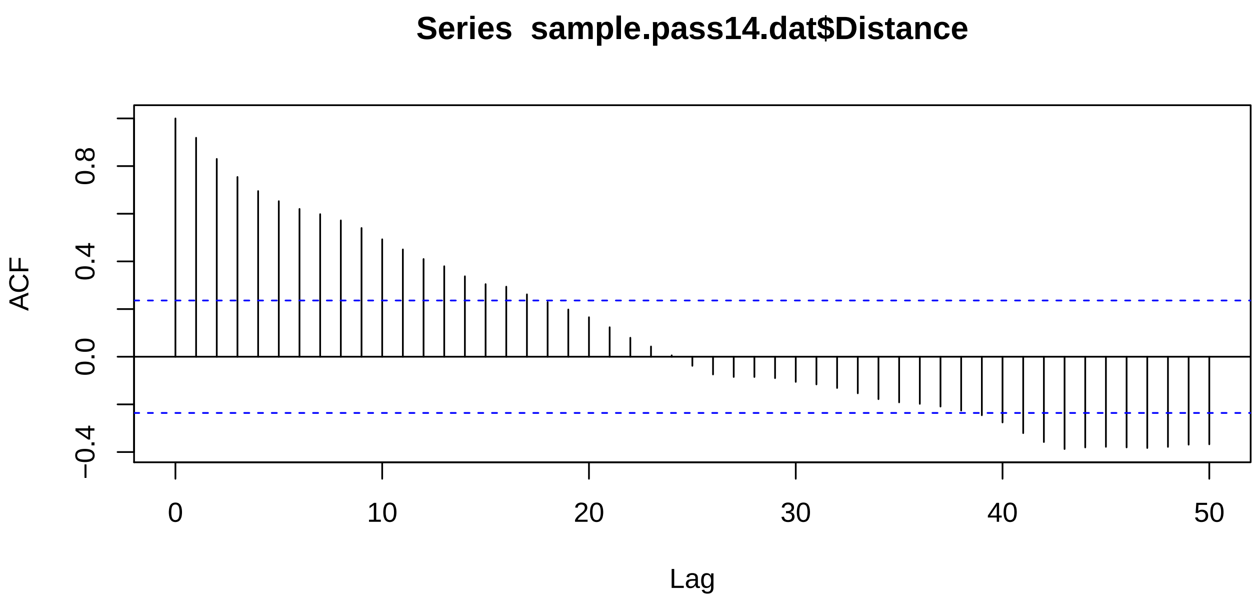

```
plot(sample.pass15.dat$Yield)
```



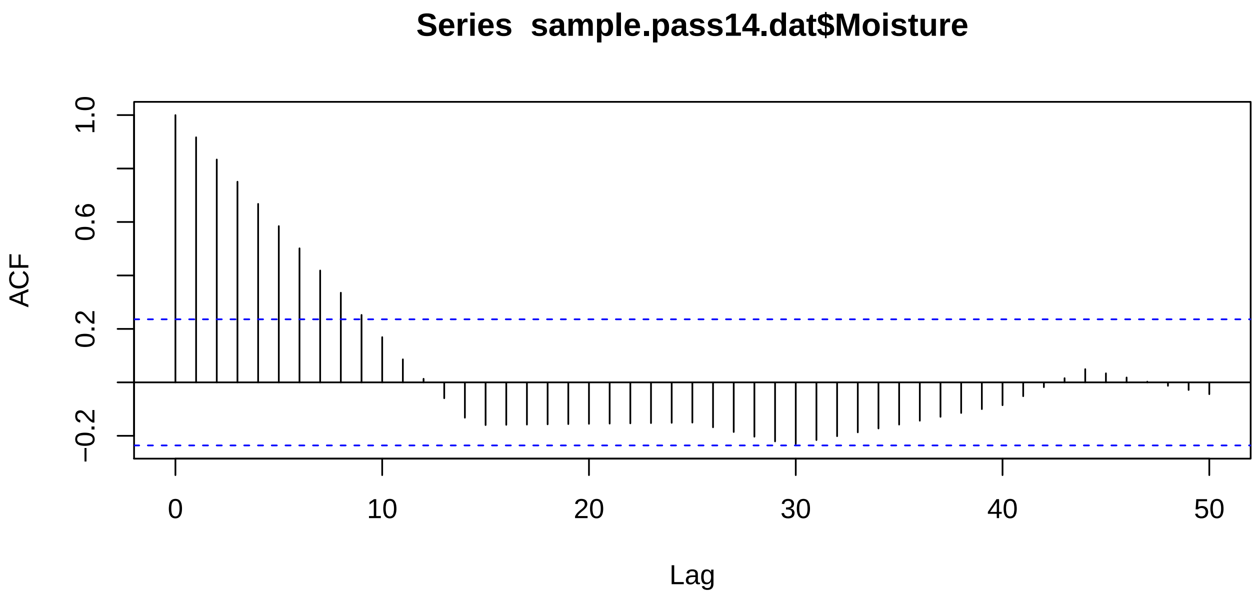

```
trend.error.acf = acf(sample.pass15.dat$Yield,lag.max=50)
```




```
trend.error.acf = acf(sample.pass14.dat$Distance,lag.max=50)
```




```
trend.error.acf = acf(sample.pass14.dat$Moisture,lag.max=50)
```



Variogram

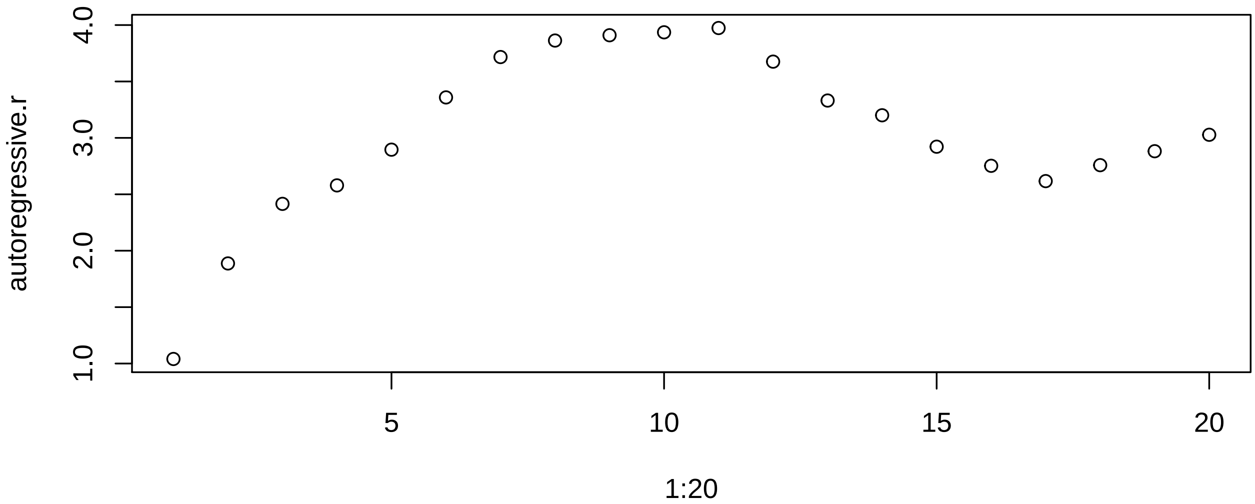
We've used a measure of correlation that depends on calculating a mean $\bar{y}$. Now suppose we wish to use a measure of difference between observations that is independent of a mean. We write

$$\gamma_k = \frac{\Sigma_{i=1}^{n-k}(y_i - y_{i-k})^2}{2(n-k)}$$

and implement this as

```
auto.correlation <- function(univariate,k=1) {
  n <- length(univariate)
  lag.ss <- sum((univariate[(1+k):n]-univariate[1:(n-k)])^2)
  return(lag.ss/(n-k))
}
```

```
autoregressive.r <- rep(0,20)
for(i in 1:20) {
  autoregressive.r[i] <- auto.correlation(autoregressive,k=i)
}
plot(1:20,autoregressive.r)
```



This is an example of a variogram, which we'll consider in more detail when we move to two dimensional analysis.