# Autocorrelation and Autoregression, State Space Embedding, Vector Fields, and Time Shuffling

Steven M. Boker

Department of Psychology
University of Virginia

Dynamical Systems Analysis Workshop Part 2

Modern Modeling Methods
May 22, 2017

# Longitudinal and Burst Measurement

- *Longitudinal Measurement* is within–subjects measurement where occasions are separated by an interval of time.
- *Burst Measurement* is within–subjects measurement where during one burst a variable is measured many times separated by short intervals of time, but longer periods of no measurement occur between these bursts.

# Examining Time Dependence

- Time dependence refers to information contained in the order of the sequence in which the measurements were obtained.
- If the ordering of the sequence doesn't matter, there is no time dependence.
- Some analyses assume time–independent observations.
- Other analyses assume time–independent residuals.

# Autocorrelation

- Autocorrelation is the correlation of a variable at one occasion with the same variable at a subsequent occasion.
- For a time series $X = \{x_1, x_2, \ldots, x_N\}$, a first order autoregression equation would be

$$x_t = bx_{t-1} + e_t$$

  where $t$ is the index into the time series, $b$ is the autoregression coefficient, and $e_t$ is frequently called the *innovation* or *shock* term and is assumed to be time–independent.
- Such a model would result in data in which $x$ showed autocorrelation.

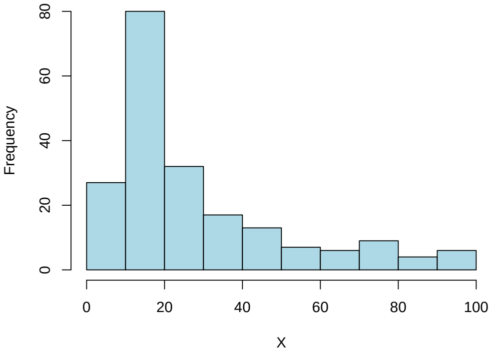
# Autoregression Example

TimeSeries1.R -- MARKER A

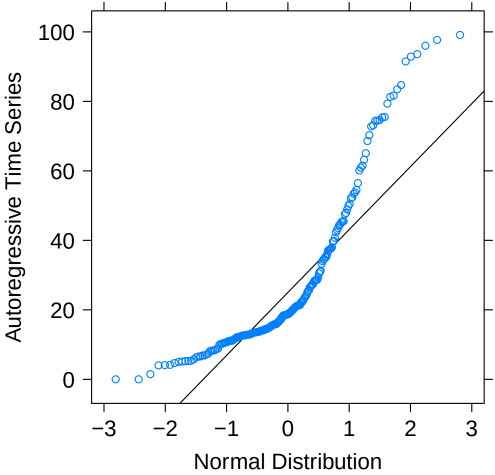
- We will build a first order autoregressive time series with 200 observations and an autoregression coefficient of 0.975.

```
tX1 <- rep(NA, 201)
tX1[1] <- 90
for (i in 2:201) {
   tX1[i] <- .975 * tX1[i-1]
}
tY1 <- 10 + .5 * tX1 + rnorm(201, mean=0, sd=5)
tX1 <- 10 + tX1 + rnorm(201, mean=0, sd=5)
tX1[tX1 <= 0] <- 0.001
tData1 <- data.frame(x=tX1, y=tY1)
summary(tData1)
```

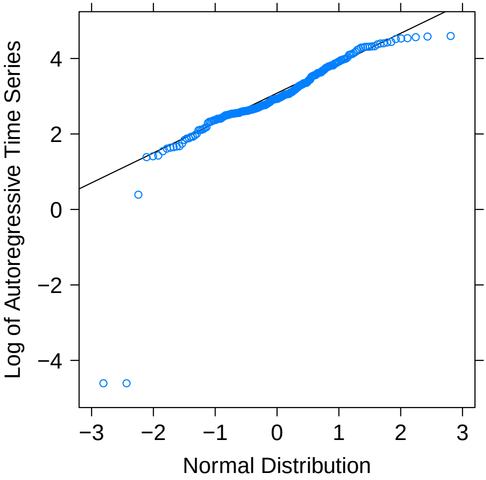
# Autoregression Example



# Autoregression Example



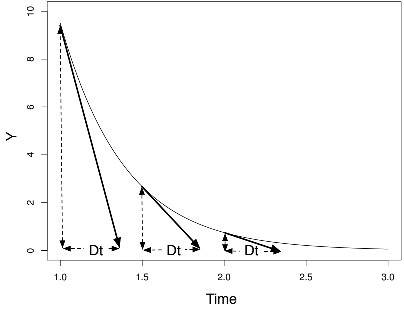
# Autoregression Example (log transformed)



# Time Series Plots

- But why did a log transformation work?
- The log transformation tells us something about what kinds of processes might be at work.
- One process that exhibits a log distribution of scores is *exponential decay*.
- With exponential decay, there is time dependence such that subsequent measurements are proportionately closer to some asymptotic value.

# Exponential Decay




- The tangent crosses zero after exactly the same elapsed time $\Delta t$ no matter what the value of $y$.
- Thus the slope of the curve is proportional to the value of $y$.
- And so, a first order linear differential equation is the continuous time equivalent to an AR(1) process.

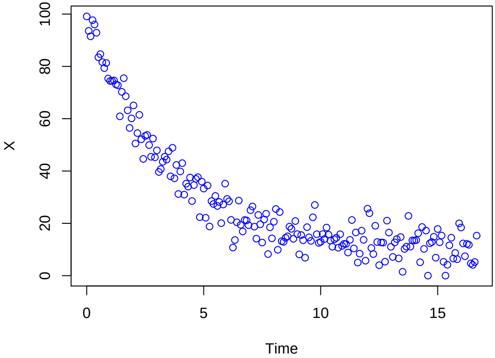
# Time Series Plot

TimeSeries1.R -- MARKER B

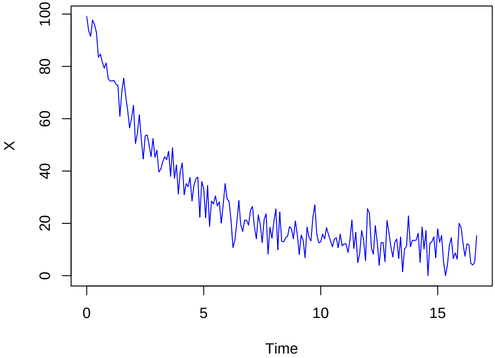
- Create a dataframe with a time variable scaled in minutes where observations were taken every 5 seconds.
- Create a scatterplot of time versus x.

```
tTime <- seq(0,1000, by=5)/60
tData1a <- data.frame(x=tX1, y=tY1, time=tTime)
pdf("TimeSeries1TSPlot1.pdf", height=5, width=5)
plot(tData1a$time, tData1a$x,
     xlab="Time",
     ylab="X",
     type='p',
     pch=1,
     col="blue")
dev.off()
```

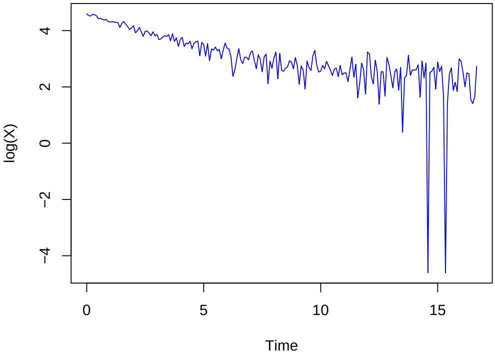
# Time Series Plot



# Time Series Plot



# Time Series Plot (log transformed)



# Surrogate Data

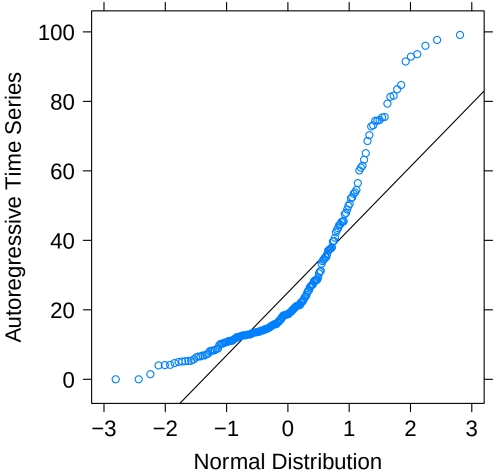
`TimeSeries1.R -- MARKER C`

- If there is no time dependence, it shouldn't matter in what order we plot the values.

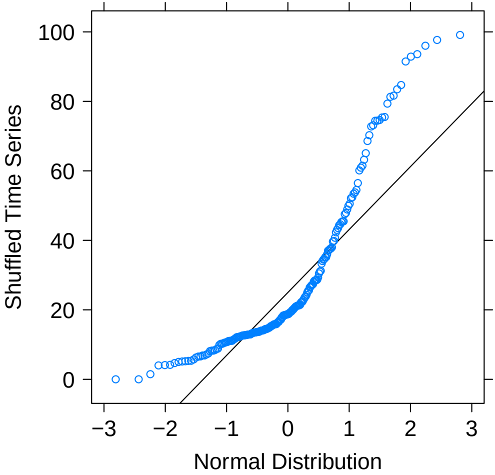
```
tTimeRand <- order(runif(201,0,1))
tData1b <- data.frame(x=tX1, y=tY1, time=tTimeRand)
```

- `tTimeRand` now has a random ordering of the data.
- This is called a *time shuffled surrogate* data set.
- There are many types of surrogate data methods, each ensuring that some null hypothesis is true.
- In general, we create many surrogate data sets in which we know the null hypothesis is true and then see if the real data are unusual in comparison to the distribution of some chosen statistic calculated on the surrogates and on the real data.

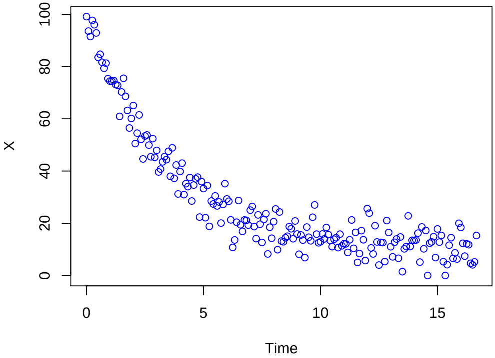
# QQ Plot of Original Ordering



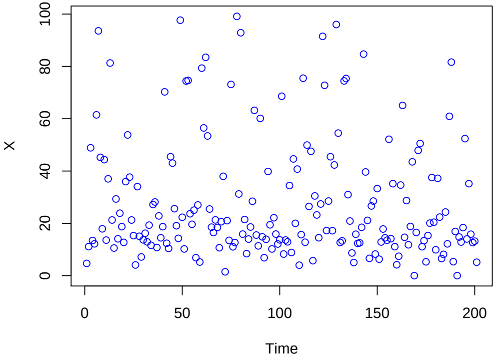
# QQ Plot of Randomized Ordering



# Time Series Plot of Original Ordering



# Time Series Plot of Randomized Ordering



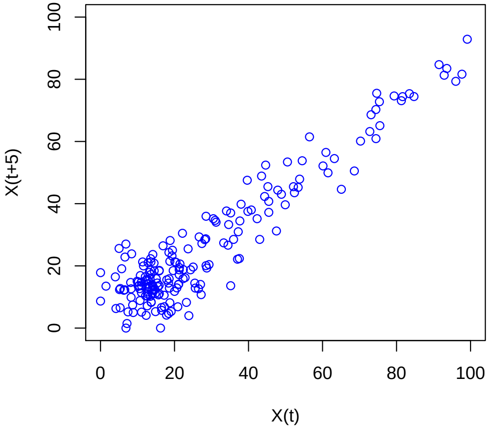
# State Space Plots

TimeSeries1.R -- MARKER D

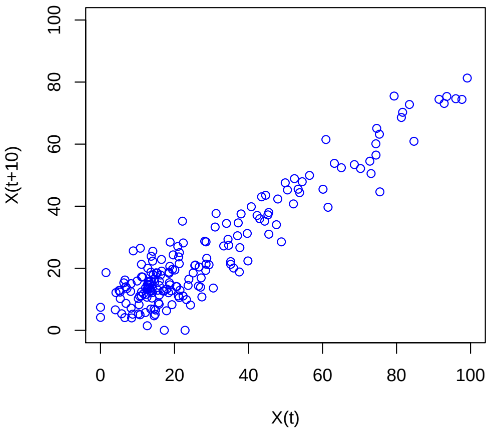
- A state space plot is created when a time series is plotted against a time lagged version of itself.
- This is a plot of a two dimensional time delay embedding.
- When time dependence is a question for exploration, state space plots can help visualize the structure of the time dependence.
- The time lag (often denoted $\tau$) is frequently varied to see if there are any “special” time intervals that exhibit greater or less time dependence.

```
tau <- 5
tLen <- length(tX1)
tDataLag1 <- data.frame(x1=tX1[1:(tLen-tau)],
                        x2=tX1[(1+tau):tLen])
```

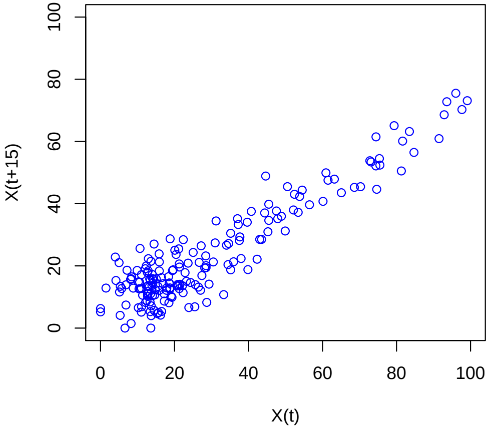
# State Space Plot: $\tau = 5$



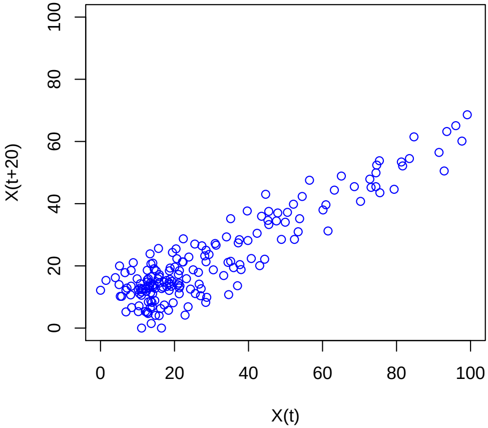
# State Space Plot: $\tau = 10$



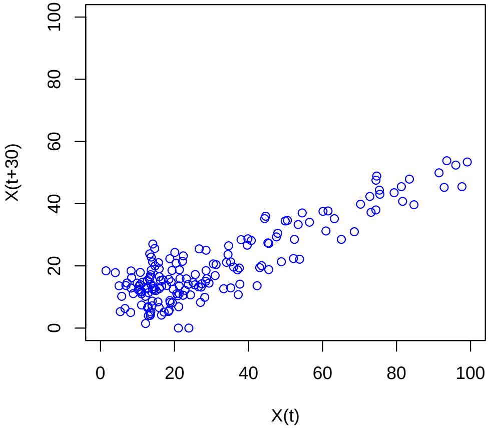
# State Space Plot: $\tau = 15$



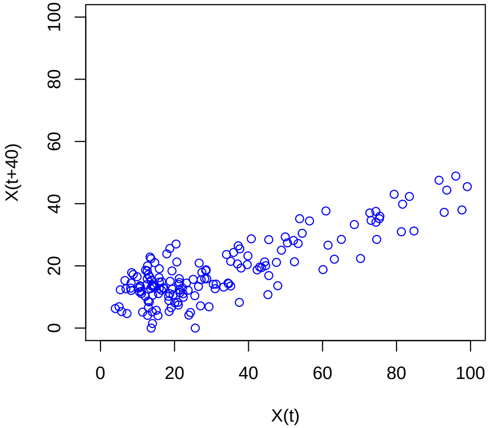
# State Space Plot: $\tau = 20$



# State Space Plot: $\tau = 30$



# State Space Plot: $\tau = 40$



# State Space Plots of Time Shuffled Surrogates

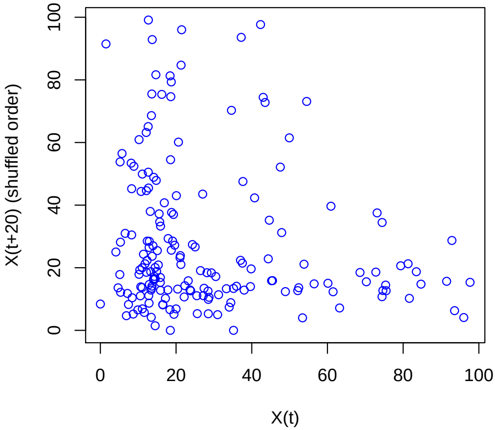
TimeSeries1.R -- MARKER E

- If there is time dependence, time shuffled surrogate data will appear different than time ordered data in a state space plot.

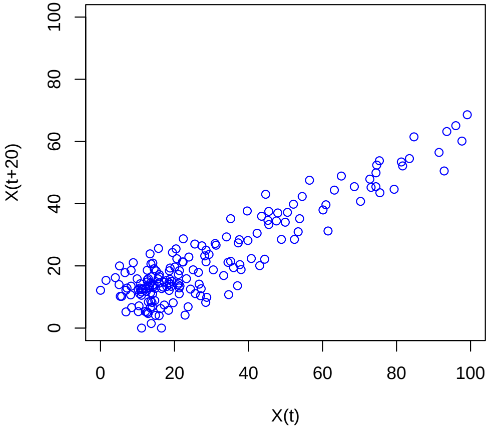
```
tau <- 20
tLen <- length(tX1)
tShuffledX <- tX1[tTimeRand]
tDataLagShuf <- data.frame(x1=tShuffledX[1:(tLen-tau)],
                           x2=tShuffledX[(1+tau):tLen])
```

- Remember, time shuffled surrogates ensure that the null hypothesis of no time dependence is true.

# Time Shuffled State Space Plot: $\tau = 20$



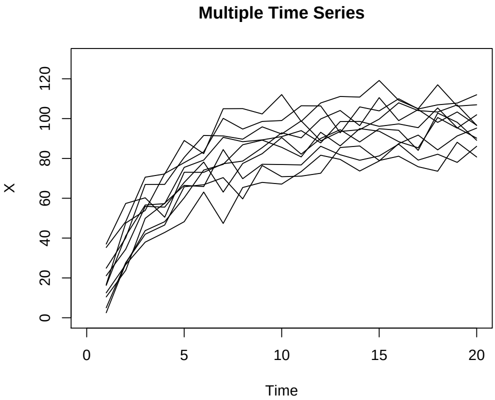
# State Space Plot: $\tau = 20$



# Mixed Longitudinal Data

- Generally we gather data on a sample composed of many participants.
- We may also measure each participant many times.
- These data are frequently stored in a matrix where each individual has one row and the multiple observations are in a range of columns.
- Let's simulate a data set where each of 10 participants has their own autoregression coefficient and is measured 20 times.
- We will plot each person's data onto the same plot.

# Time Series of Mixed Longitudinal Data



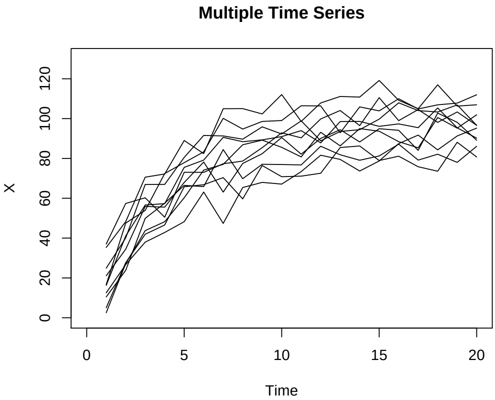
# Converting Longitudinal Data to Columnar Format

## TimeSeries1.R -- MARKER F

- When one wishes to fit random coefficients (mixed effects) models to these longitudinal data, we need to convert to columnar format.

```
mixedMatrix <- matrix(NA, totalSubjects*totalObservations, 3)
for (i in 1:totalSubjects) {
    tRows <- c(1:totalObservations) +
           ((i - 1) * totalObservations)
    mixedMatrix[tRows, 1] <- rep(longMatrix[i,1],
           totalObservations)
    mixedMatrix[tRows, 2] <- c(1:totalObservations)
    mixedMatrix[tRows, 3] <-
           longMatrix[i,2:(totalObservations+1)]
}
summary(mixedMatrix)
```

# Time Series of Mixed Longitudinal in Columnar Format



# Autocorrelation with Time Dependent Error

- If the error is independent of time (measurement error) then

$$\begin{aligned} \hat{x}_t &= b\hat{x}_{t-\tau} \\ x_t &= \hat{x}_t + e_t \end{aligned}$$

- If the error from the previous trial is added into a time-forward prediction (dynamic error) then

$$x_t = bx_{t-\tau} + e_t$$

- Dynamic error produces a form of random walk.
- This accumulated error is alternatively called *dynamic error*, *stochastic error*, or *shocks*.
- In psychological data, there is likely to be both measurement error and dynamic error.

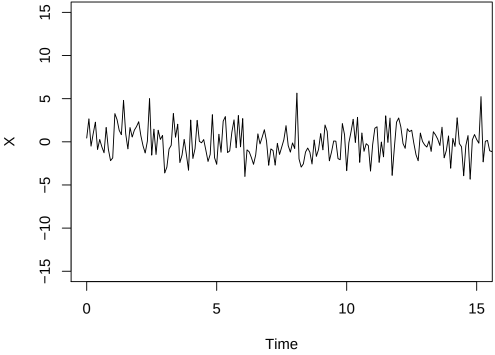
# Two Example Random Walks

TimeSeries2.R -- MARKER A

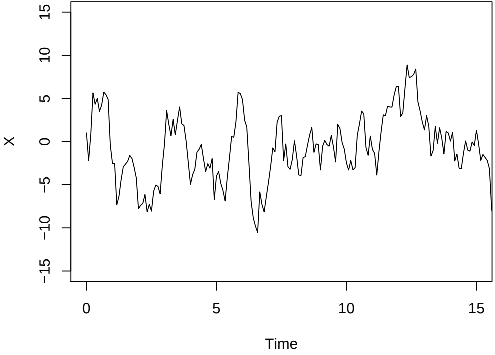
```
tX2 <- rep(NA, 201)
tX2[1] <- 1
for (i in 2:201) {
   tX2[i] <- .9 * tX2[i-1] + rnorm(1, mean=0, sd=2)
}
tY2 <- 5 + .5 * tX2 + rnorm(201, mean=0, sd=5)

tTime <- seq(0,1000, by=5)/60
tData2 <- data.frame(x=tX2, y=tY2, time=tTime)
summary(tData2)
```

```
       x                  y                time
 Min.   :-12.596   Min.   :-9.6538   Min.   : 0.000
 1st Qu.: -4.527   1st Qu.: 0.3361   1st Qu.: 4.167
 Median : -1.722   Median : 4.0269   Median : 8.333
 Mean   : -1.676   Mean   : 3.9543   Mean   : 8.333
 3rd Qu.:  1.330   3rd Qu.: 7.7170   3rd Qu.:12.500
 Max.   :  8.754   Max.   :17.0844   Max.   :16.667
```

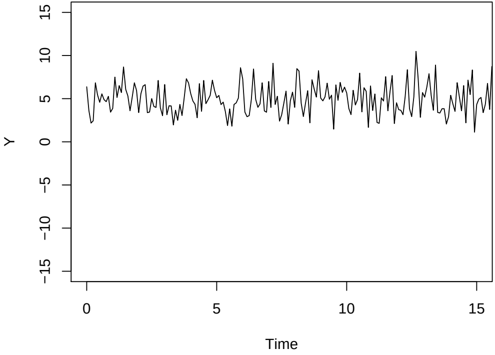
# Autoregression Series at Equilibrium with Measurement Error



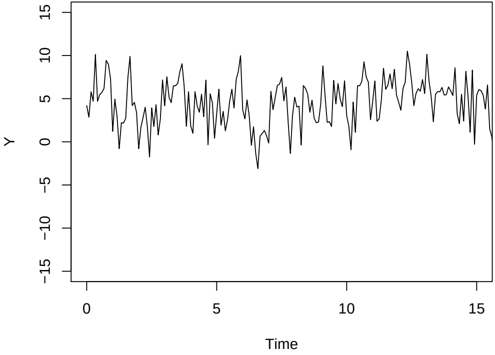
# Autoregression Series at Equilibrium with Stochastic Error



# Crossregression Series at Equilibrium with No Stochastic Error



# Crossregression Series at Equilibrium with Stochastic Error



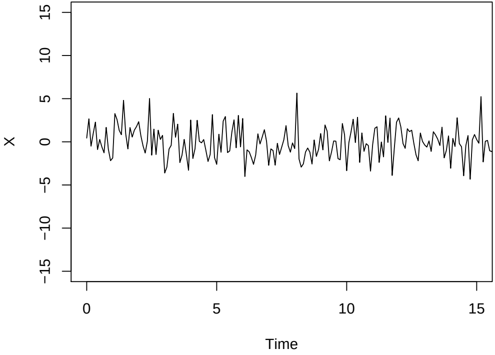
# Time Shuffled Surrogates

`TimeSeries2.R -- MARKER B`

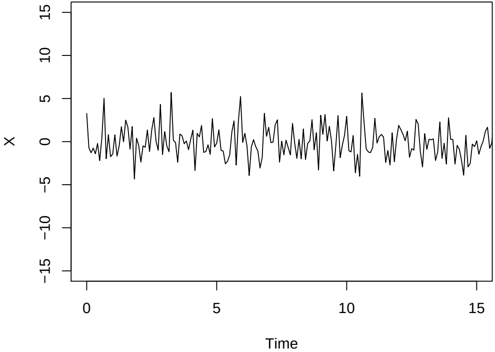
- A time shuffled surrogate of a autoregressive series at equilibrium with only measurement error may not look a lot different from the original time series.

```
tTimeRand1 <- order(runif(201,0,1))
tData1b <- data.frame(x=tX1[tTimeRand1], time=tTime)
```

# Autoregression Series at Equilibrium with Measurement Error



# Time Shuffled Autoregression Series at Equilibrium with Measurement Error



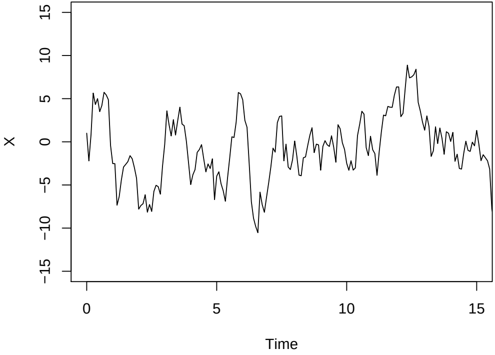
# Time Shuffled Surrogate of a Random Walk

`TimeSeries2.R -- MARKER C`

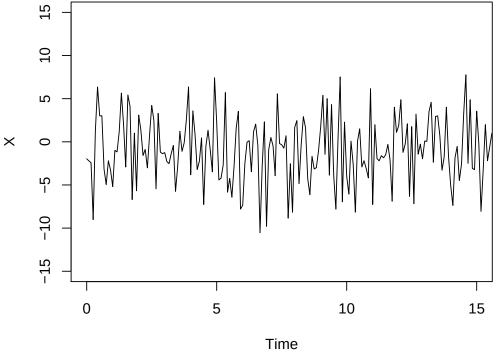
- A time shuffled surrogate of a autoregressive series at equilibrium with stochastic error may show you the effect of the dynamic error.

```
tTimeRand2 <- order(runif(201,0,1))
tData2b <- data.frame(x=tX2[tTimeRand2], time=tTime)
```

# Autoregression Series at Equilibrium with Stochastic Error



# Time Shuffled Autoregression Series at Equilibrium with Stochastic Error



# State Space Plot of a Random Walk

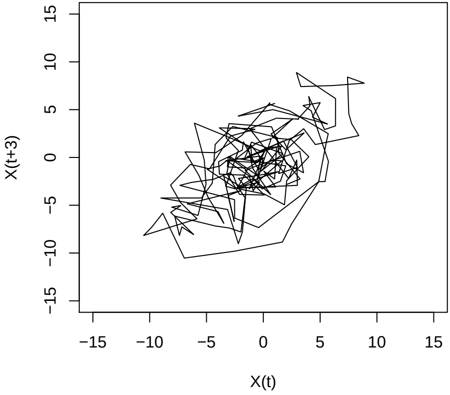
`TimeSeries2.R -- MARKER D`

- If we create a state space plot of the Random Walk data, the time dependence becomes more clear.

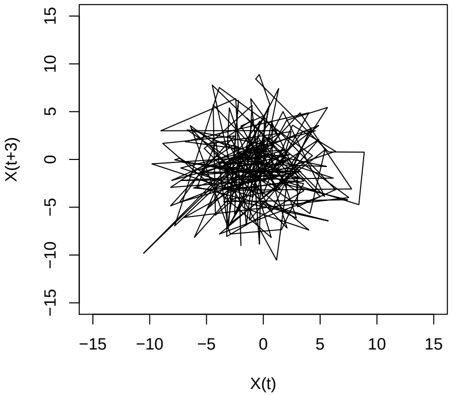
```
tau <- 3
tLen <- length(tX2)
tDataLag2 <- data.frame(x1=tX2[1:(tLen-tau)],
                        x2=tX2[(1+tau):tLen])
```

# Autoregression Series at Equilibrium with Stochastic Error

## Embedding Time Lag = 3

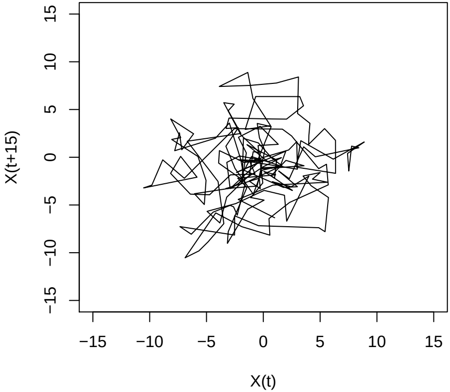


# Time Shuffled Autoregression Series at Equilibrium with Stochastic Error



# Autoregression Series at Equilibrium with Stochastic Error

## Embedding Time Lag = 15



# Autocorrelation Function (ACF)

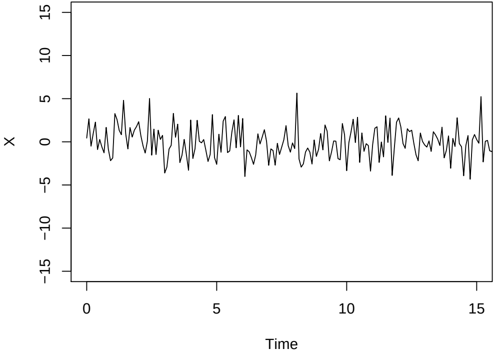
- The autocorrelation function calculates autocorrelation at a range of lags.
- The relationship between the lag and the autocorrelation can be diagnostic.
- Autoregressive functions show a characteristic decay of autocorrelation as lag increases.
- Cyclic functions show peaks in the autocorrelation where the lag is equal to the period of the cycle.

# Autocorrelation Function

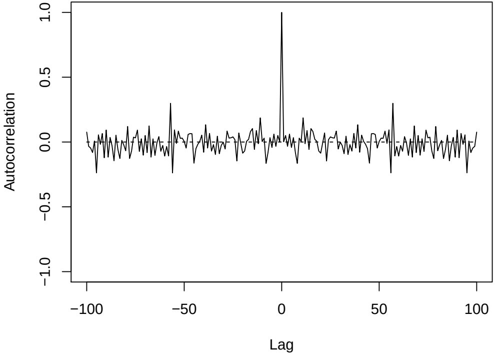
TimeSeries2.R -- MARKER E

```
maxTau <- 100
tLen <- length(tX1)
tACF1 <- rep(NA, maxTau+1)

for (tau in (0:maxTau)) {
   t1 <- tX1[1:(tLen-tau)]
   t2 <- tX1[(1+tau):tLen]
   tSelect <- !is.na(t1) & !is.na(t2)
   if (length(t1[tSelect]) < 5) next
   tACF1[tau+1] <- cor(t1[tSelect], t2[tSelect])
}
```

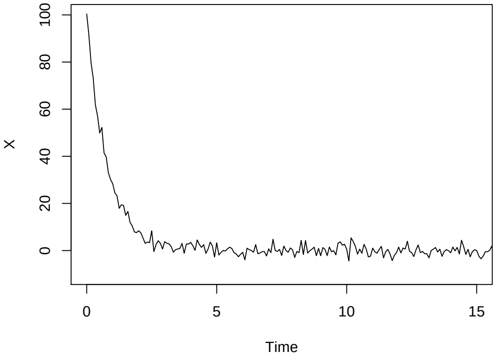
# Autoregression Series at Equilibrium with Measurement Error



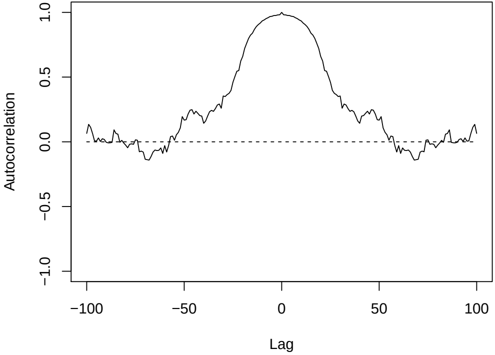
# Autoregression Series at Equilibrium with Measurement Error



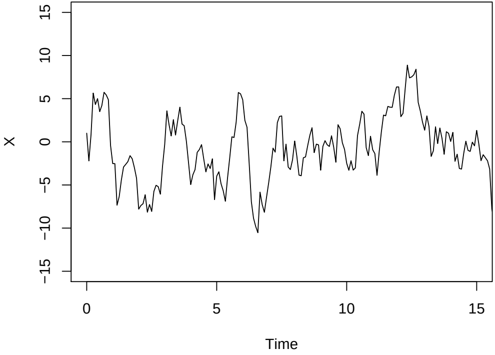
# Autoregression Series Far from Equilibrium with Measurement Error



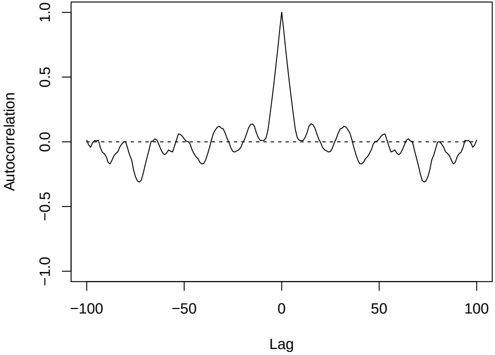
# Autoregression Series Far from Equilibrium with Measurement Error



# Autoregression Series at Equilibrium with Stochastic Error



# Autoregression Series at Equilibrium with Stochastic Error



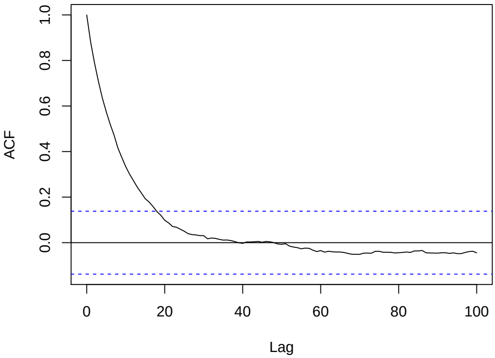
# Using the acf() Function

## TimeSeries2.R -- MARKER F

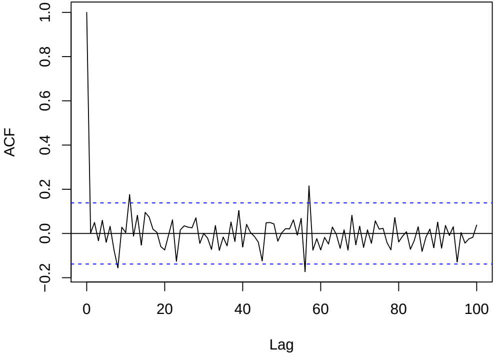
- There is a function acf() in the package stats that hides the complexity of calculating an autocorrelation function.

```
pdf("TimeSeries2ACF0b.pdf", height=5, width=6)
tACF = acf(tX0,
           lag.max=maxTau,
           type="correlation",
           plot=FALSE,
           na.action=na.fail)
plot(tACF,
     main="",
     ci=.95,
     ci.col="blue",
     ci.type="white",
     type="l",
     cex=1.25)
dev.off()
```

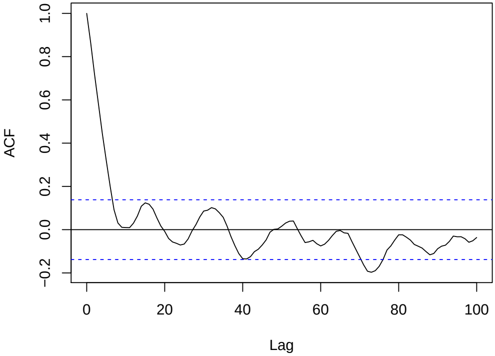
# Autoregression Series Far from Equilibrium with Measurement Error



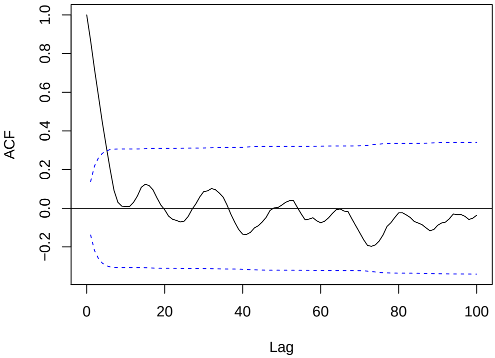
# Autoregression Series at Equilibrium with Measurement Error



# Autoregression Series at Equilibrium with Stochastic Error



# Autoregression Series at Equilibrium with Stochastic Error and Moving Average SE



# Cyclic Functions

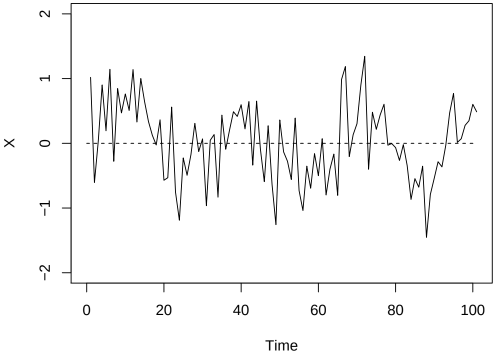
TimeSeries2.R -- MARKER G

- Sometimes the data may contain a cyclic function.

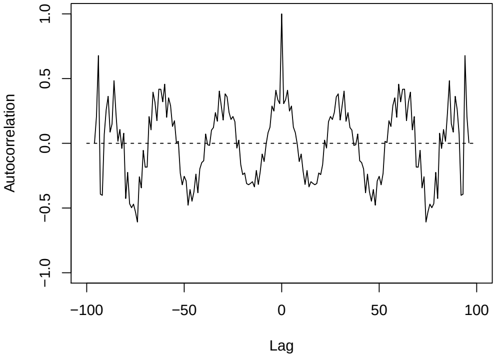
```
tX3 <- sin(seq(0,20, by=.2)) * .5
tY3 <- 5 + .5 * tX3 + rnorm(101,mean=0, sd=.5)
tX3 <- tX3 + rnorm(101,mean=0,sd=.5)

tData3 <- data.frame(x=tX3,y=tY3)
summary(tData3)
```

```
       x                  y
 Min.   :-1.73618   Min.   :3.184
 1st Qu.:-0.52803   1st Qu.:4.692
 Median :-0.08019   Median :4.959
 Mean   :-0.04978   Mean   :5.000
 3rd Qu.: 0.43114   3rd Qu.:5.341
 Max.   : 1.35803   Max.   :6.238
```

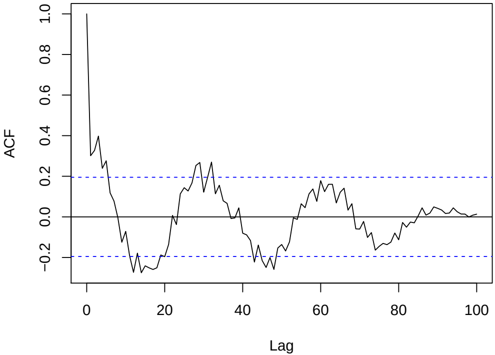
# Cyclic Function with Measurement Error



# Cyclic Function with Measurement Error



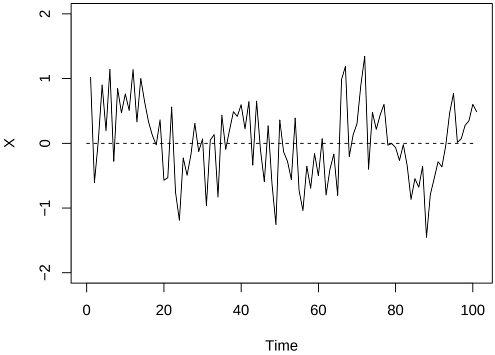
# Cyclic Function with Measurement Error



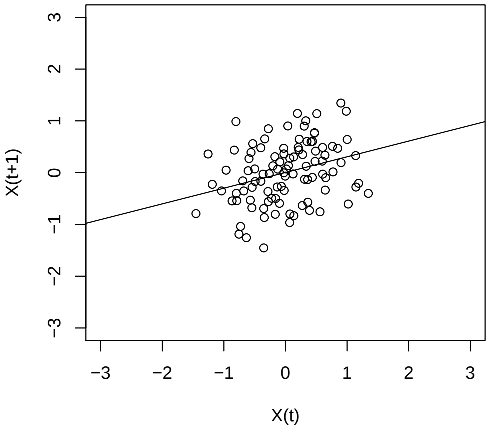
# State Space and Autocorrelation of a Cyclic Function

- Plotting the autocorrelation function of a timeseries can reveal the period of a cyclic function.
- Plotting the state space of a periodic function using a lag of 1/4 the period will maximally expand the geometry of the state space.
- The period of the cyclic function can be found from the ACF plot.

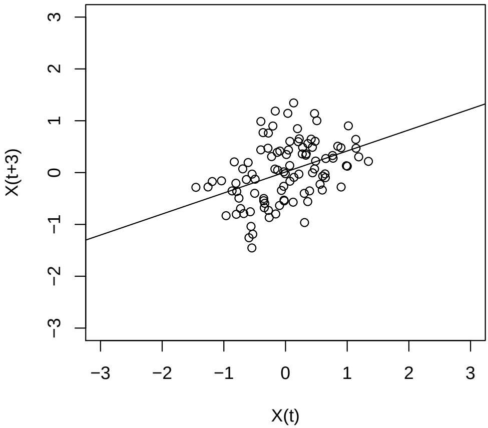
# Cyclic Function with Measurement Error



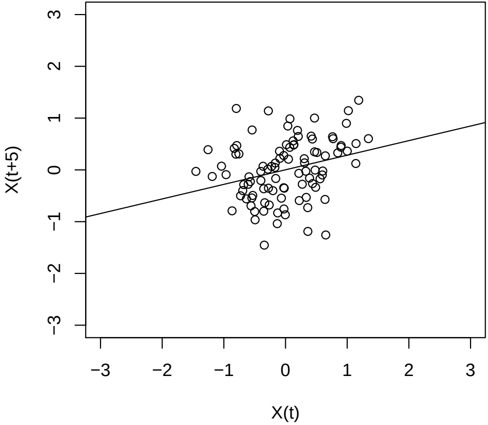
# Cyclic Function with Measurement Error ($\tau = 1$)



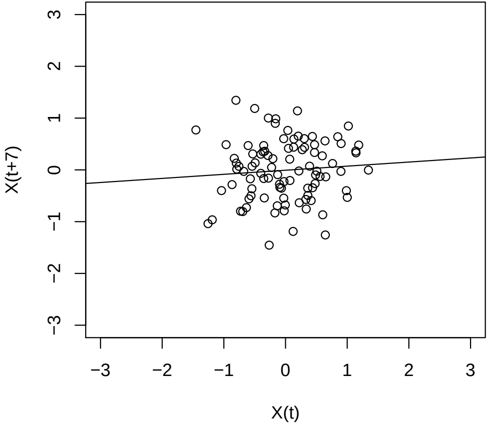
# Cyclic Function with Measurement Error ($\tau = 3$)



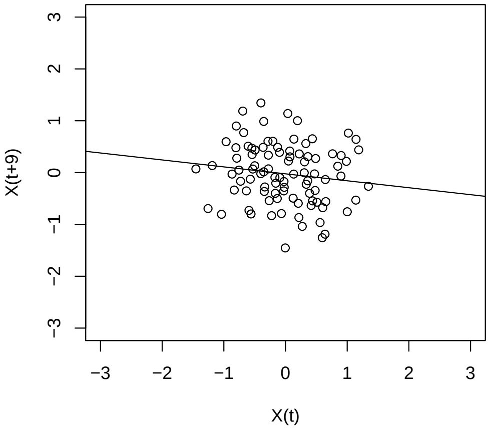
# Cyclic Function with Measurement Error ($\tau = 5$)



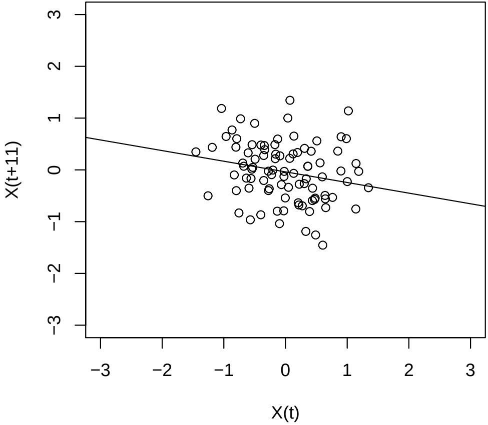
# Cyclic Function with Measurement Error ($\tau = 7$)



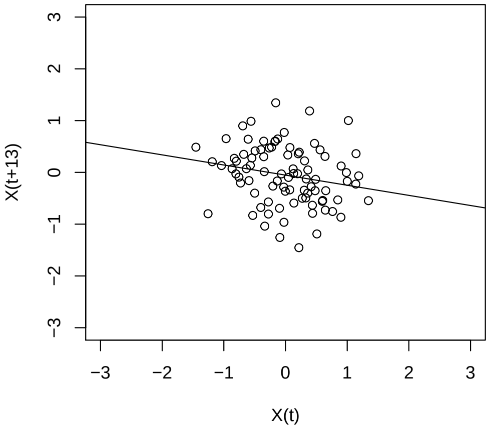
# Cyclic Function with Measurement Error ($\tau = 9$)



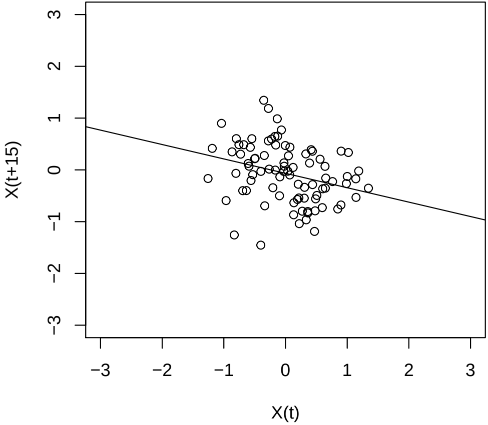
# Cyclic Function with Measurement Error ($\tau = 11$)



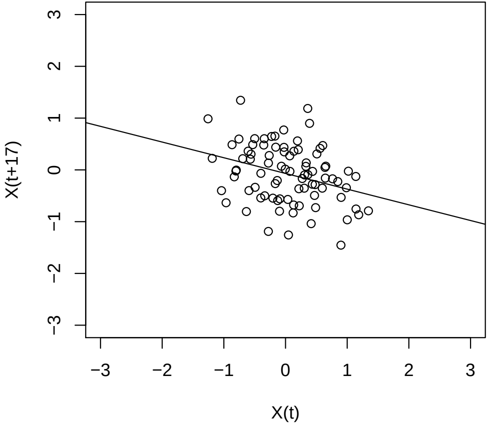
# Cyclic Function with Measurement Error ($\tau = 13$)



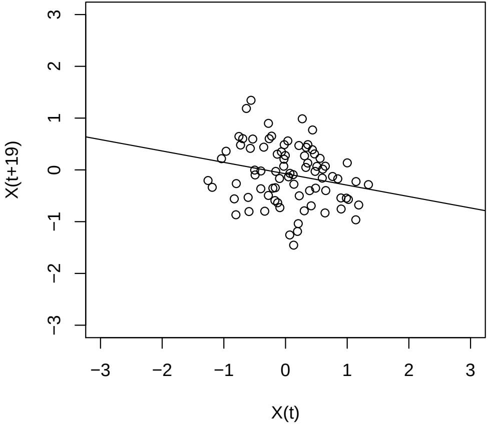
# Cyclic Function with Measurement Error ($\tau = 15$)



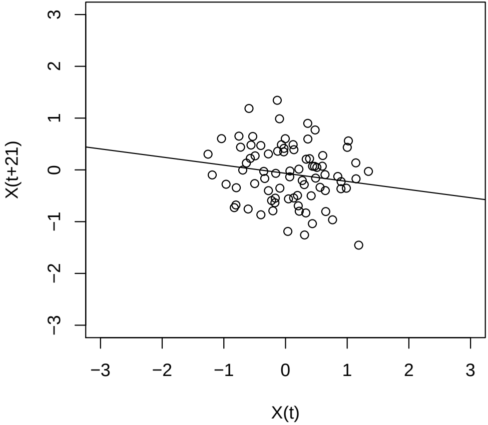
# Cyclic Function with Measurement Error ($\tau = 17$)



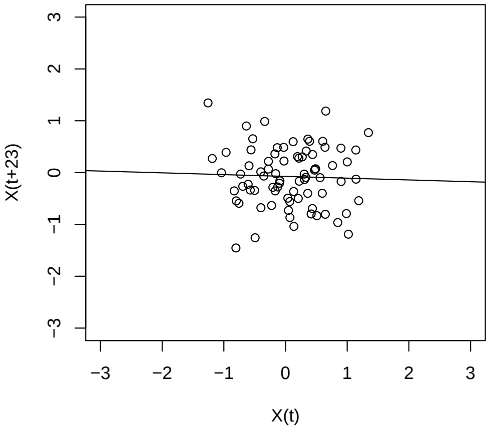
# Cyclic Function with Measurement Error ($\tau = 19$)



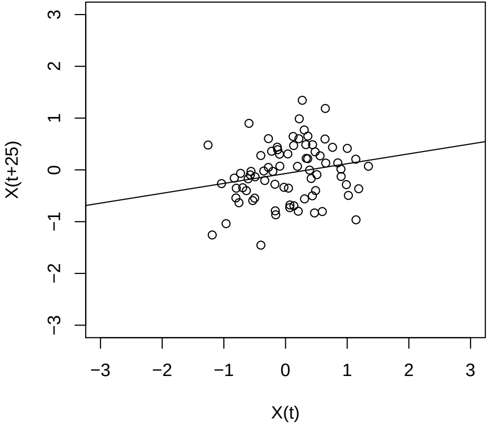
# Cyclic Function with Measurement Error ($\tau = 21$)



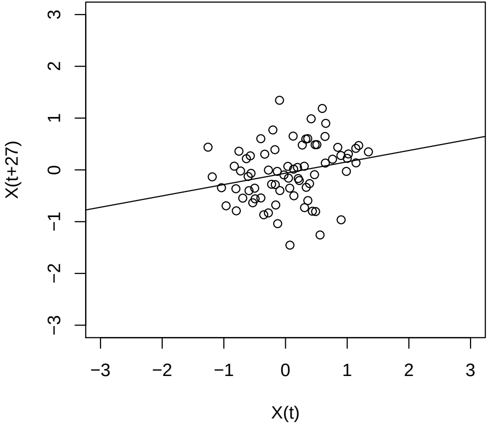
# Cyclic Function with Measurement Error ($\tau = 23$)



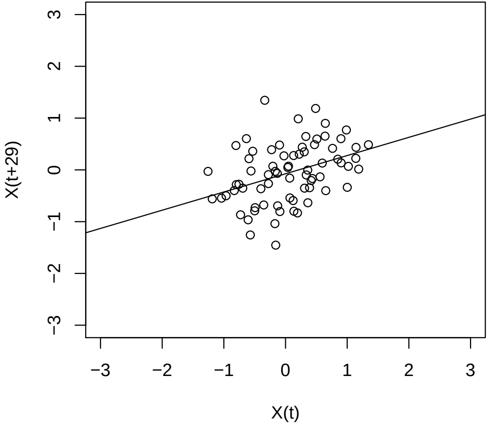
# Cyclic Function with Measurement Error ($\tau = 25$)



# Cyclic Function with Measurement Error ($\tau = 27$)



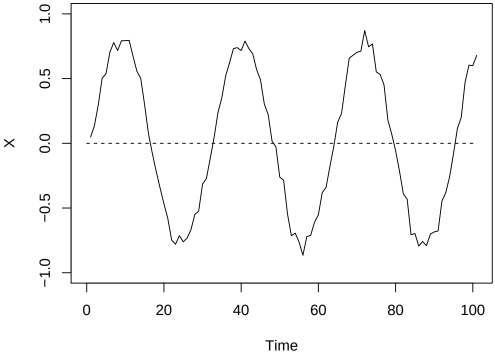
# Cyclic Function with Measurement Error ($\tau = 29$)



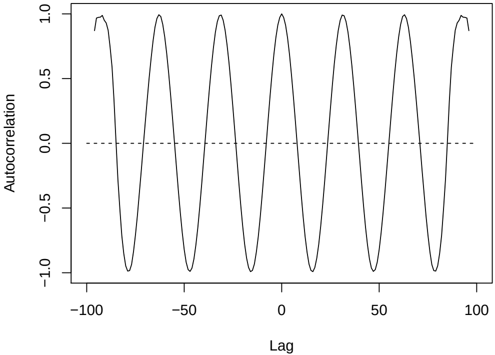
# State Space of a Cyclic Function

- Let's try again with only a little measurement error.
- Plotting the state space of a periodic function can reveal new ways of thinking about autoregression.

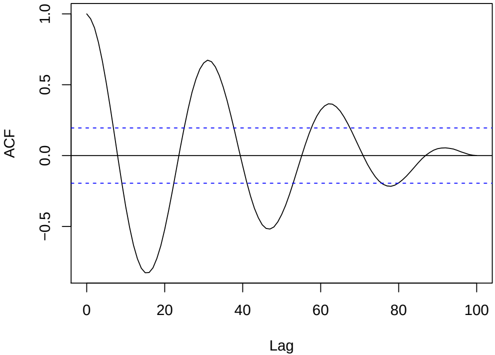
# Cyclic Function with Little Error



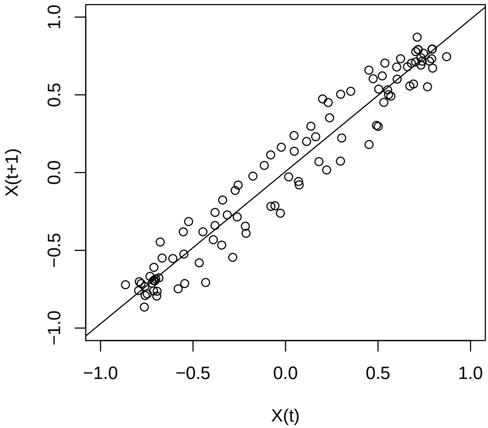
# Cyclic Function with Little Error



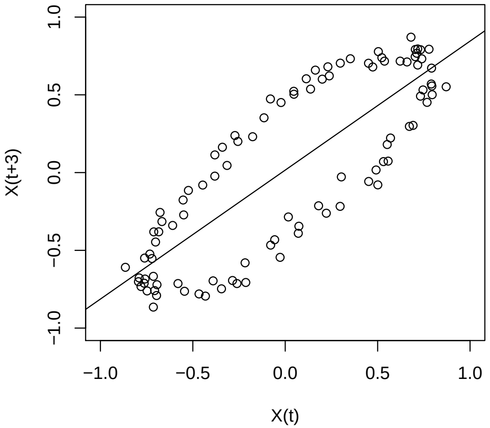
# Cyclic Function with Little Error



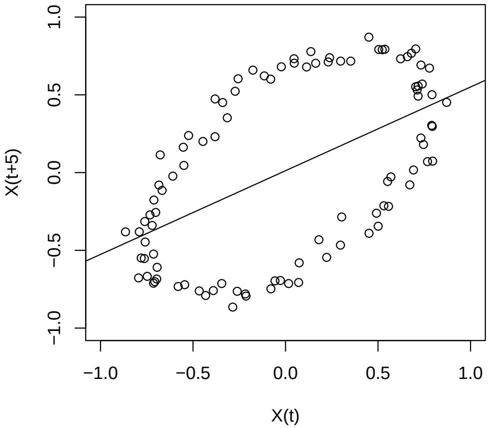
# Cyclic Function with Little Error ($\tau = 1$)



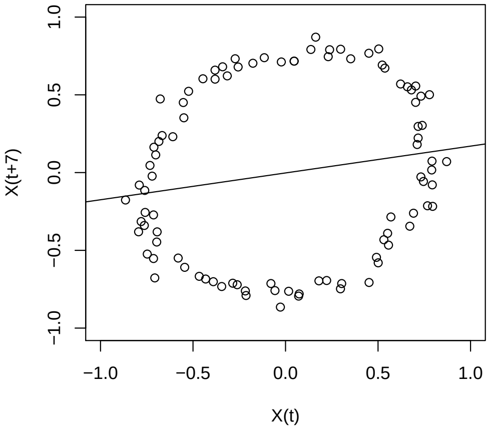
# Cyclic Function with Little Error ($\tau = 3$)



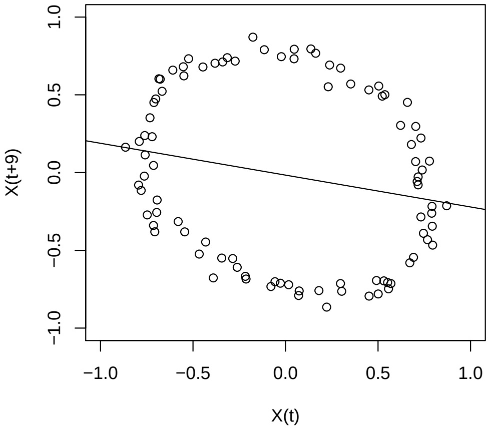
# Cyclic Function with Little Error ($\tau = 5$)



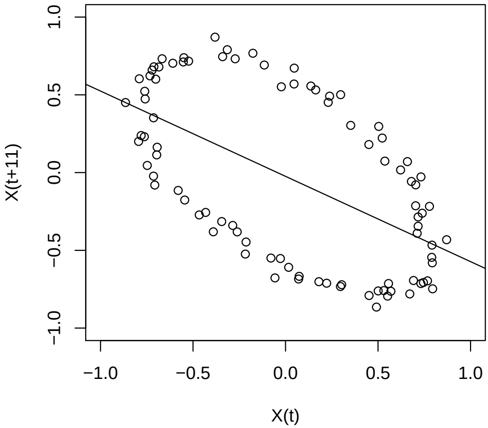
# Cyclic Function with Little Error ($\tau = 7$)



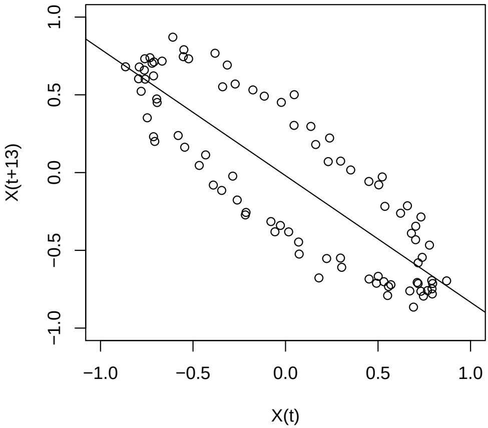
# Cyclic Function with Little Error ($\tau = 9$)



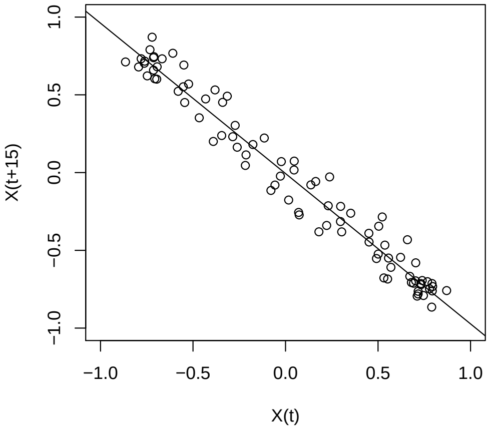
# Cyclic Function with Little Error ($\tau = 11$)



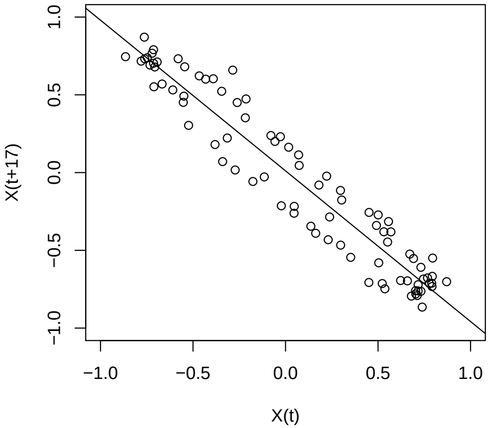
# Cyclic Function with Little Error ($\tau = 13$)



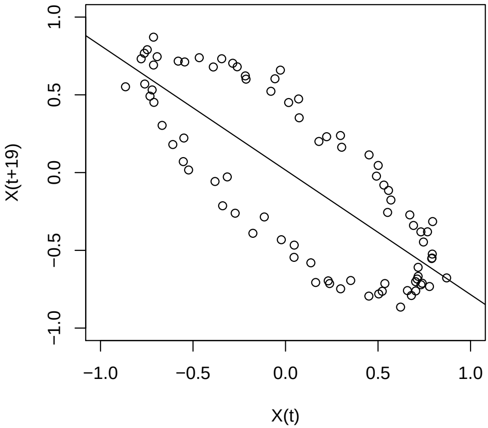
# Cyclic Function with Little Error ($\tau = 15$)



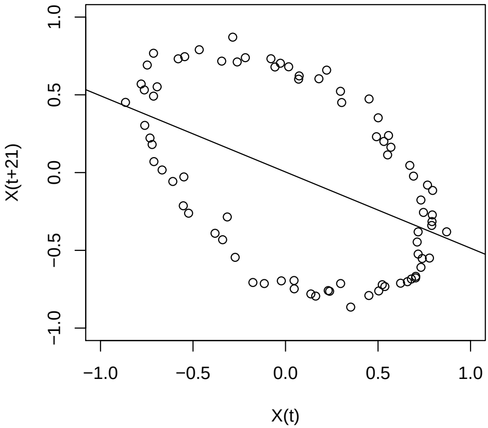
# Cyclic Function with Little Error ($\tau = 17$)



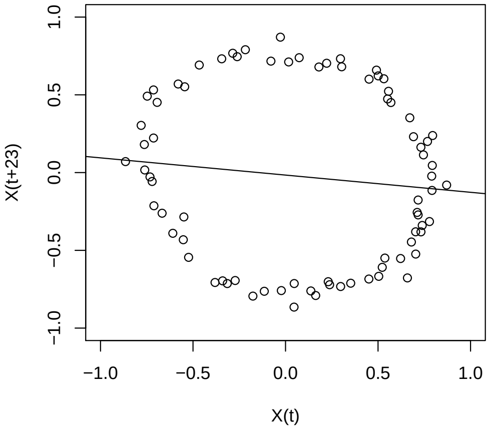
# Cyclic Function with Little Error ($\tau = 19$)



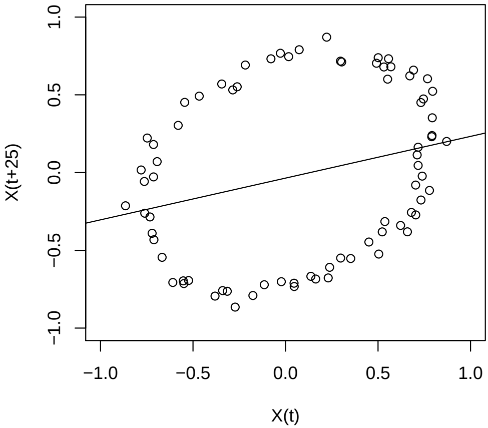
# Cyclic Function with Little Error ($\tau = 21$)



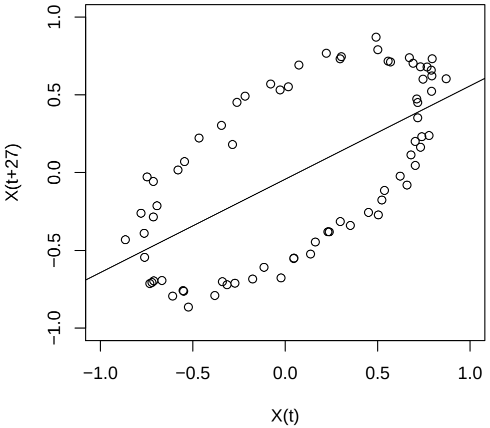
# Cyclic Function with Little Error ($\tau = 23$)



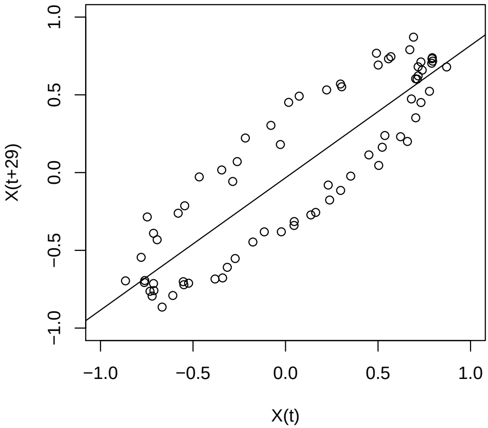
# Cyclic Function with Little Error ($\tau = 25$)



# Cyclic Function with Little Error ($\tau = 27$)



# Cyclic Function with Little Error ($\tau = 29$)



# Slope Field Plotting of Growth Curves

VectorFields1.R -- MARKER A

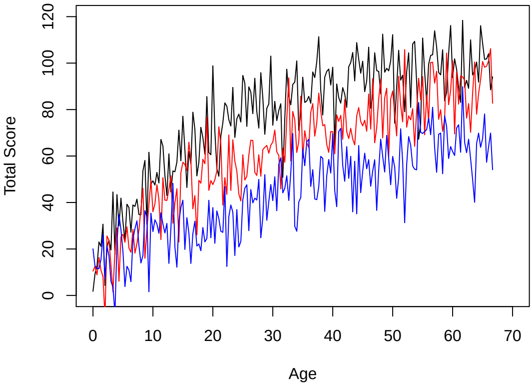
- First we will simulate a growth curve for one individual using an autoregressive model.

```
tX1 <- rep(NA, 201)
tX1[1] <- 90
for (i in 2:201) {
   tX1[i] <- .98 * tX1[i-1]
}

tX1 <- 100 - tX1 + rnorm(201, mean=0, sd=10)
```

# Simulation of Growth Curves

- Next we repeat the process using a slightly different autoregression coefficient for two more individuals.
- Consider that these three individual curves could represent individual differences in growth.
- Here are timeseries plots of the curves.

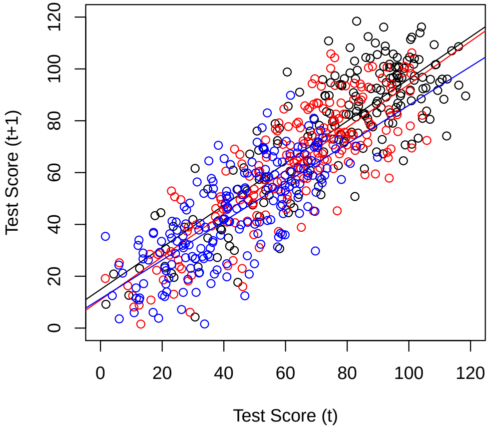
# Simulation of Growth Curves



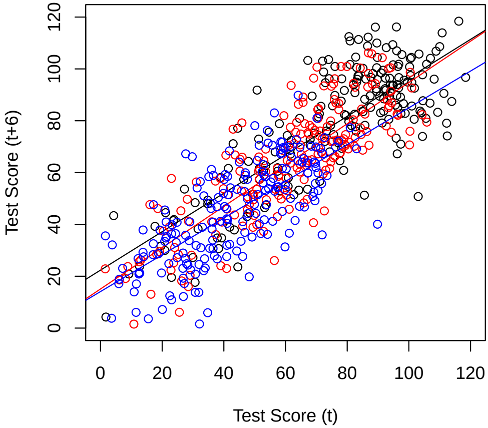
# State Space Embedding

- If the data are truly autoregressive, the embedded state space can be fit with a regression line.
- We will first plot the growth curve at a variety of lags.
- We will also color code the subjects.

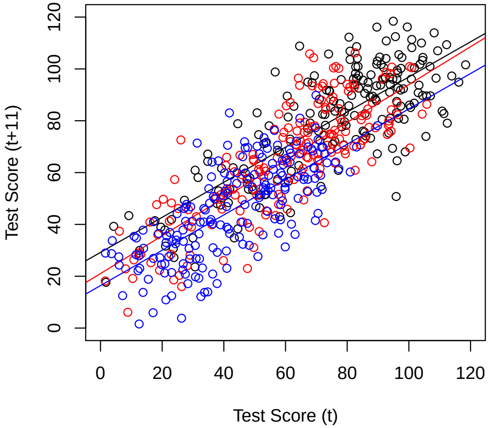
# State Space Plot of Growth Curves ($\tau = 1$)



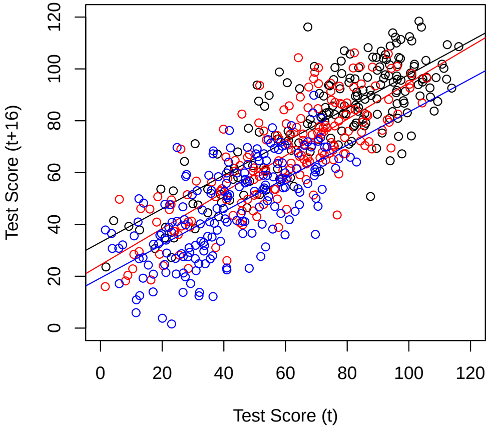
# State Space Plot of Growth Curves ($\tau = 6$)



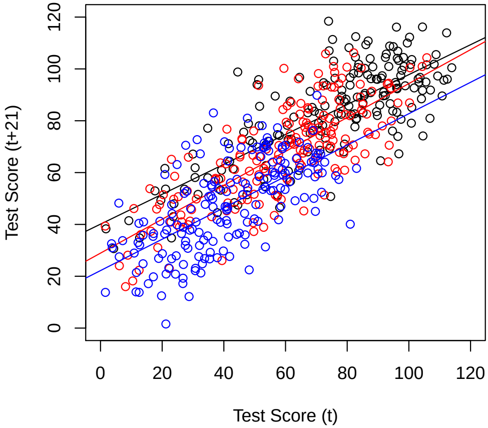
# State Space Plot of Growth Curves ($\tau = 11$)



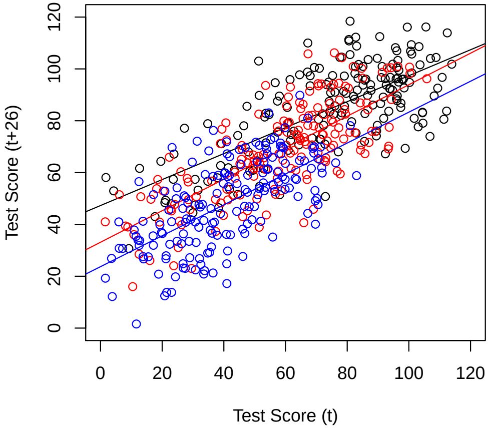
# State Space Plot of Growth Curves ($\tau = 16$)



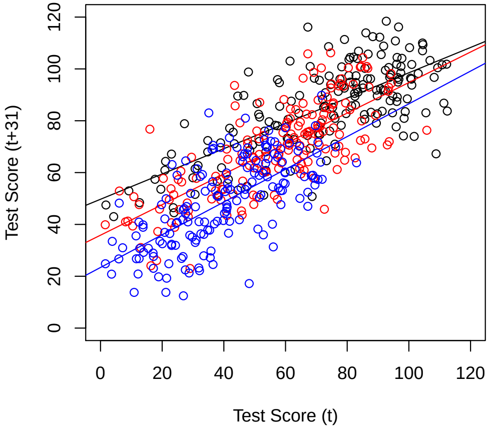
# State Space Plot of Growth Curves ($\tau = 21$)



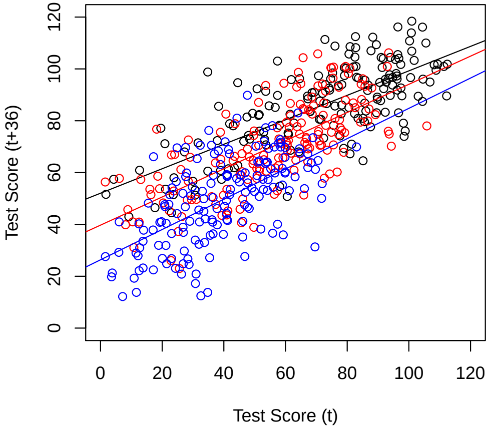
# State Space Plot of Growth Curves ($\tau = 26$)



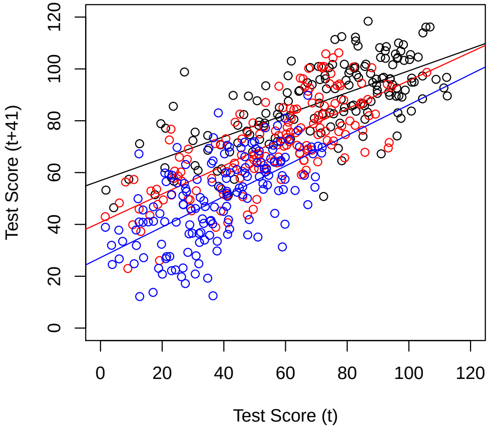
# State Space Plot of Growth Curves ($\tau = 31$)



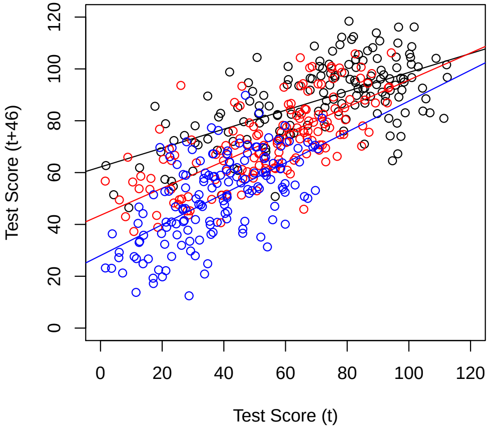
# State Space Plot of Growth Curves ($\tau = 36$)



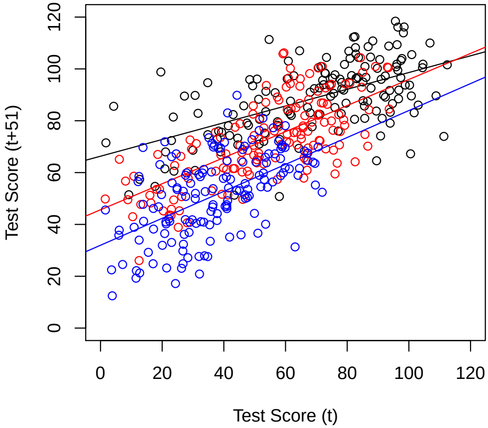
# State Space Plot of Growth Curves ($\tau = 41$)



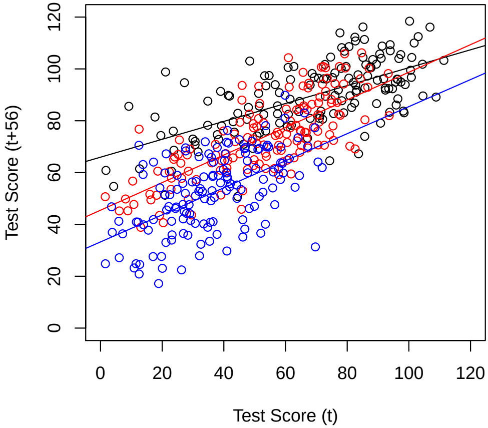
# State Space Plot of Growth Curves ($\tau = 46$)



# State Space Plot of Growth Curves ($\tau = 51$)



# State Space Plot of Growth Curves ($\tau = 56$)



# Two Things to Remember

- Notice how, as the lag increases, it becomes easier to tell the subjects' regression lines apart.

**In discrete time modeling, the interval between observations can impact the magnitude of individual differences.**

- Regulation disappears if everyone is at equilibrium.

**"In order to understand stabilization you must study instability."** (Boker, 2015)

# Line Segment Plots

VectorFields1.R -- MARKER B

- Suppose *x*1, *x*2 and *x*3 are measurements from the same individual, *age* is the age at the first measurement, and *group* is one of three groups.

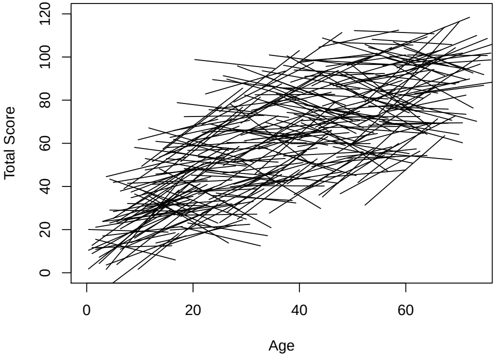
```
tau <- 15
tX <- c(tX1, rep(NA,2*tau+1), tX2, rep(NA,2*tau+1), tX3)
tGroup <- c(rep(1, length(tX1)), rep(NA,2*tau+1),
            rep(2, length(tX2)), rep(NA,2*tau+1),
            rep(4, length(tX3)), rep(NA,2*tau+1))
tAge <- c(1:length(tX1), rep(NA,2*tau+1),
           1:length(tX2), rep(NA,2*tau+1),
           1:length(tX3))/3
tLen <- length(tX)
ageData1 <- data.frame(x1=tX[1:(tLen-(2*tau))],
                          x2=tX[(1+tau):(tLen-tau)],
                          x3=tX[(1+(2*tau)):tLen],
                          age=tAge[1:(tLen-(2*tau))],
                          group=tGroup[1:(tLen-(2*tau))])
```

# Line Segment Plots

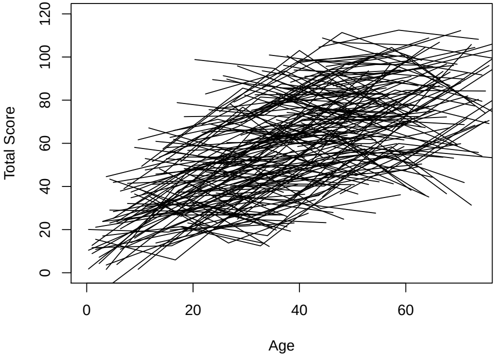
VectorFields1.R -- MARKER C

```
pdf("VF1ARLong2.pdf", height=5, width=6)
plot(c(0,201)/3, c(0,120), type="n" ,
     xlab="Age",
     ylab="Total Score")
for (i in seq(1,length(ageData1$x1),by=2)) {
   if (is.na(ageData1$age[i]) |
       ageData1$age[i] > (68-(2*tau))) {
          next
   }
   lines(c(ageData1$age[i],
           ageData1$age[i]+tau,
           ageData1$age[i]+(2*tau)),
         c(ageData1$x1[i],
           ageData1$x2[i],
           ageData1$x3[i]), type="l", lty=1)
}
dev.off()
```

# Line Segment Plots: Two wave panel data



# Line Segment Plots: Three wave panel data



# Slope Fields

- Sometimes you might be interested in how rapidly people are changing.
- Using smoothing we can calculate the mean slope within local areas.
- First we transform our data to slopes.

```
ageMid <- ageData1$age + (tau/2)
scoreMid <- (ageData1$x1 + ageData1$x2) / 2
slopeMid <- (ageData1$x2 - ageData1$x1) / tau
tSelect <- !is.na(ageMid) & !is.na(scoreMid) & !is.na(slopeMid)
ageSlopeFrame <- data.frame(
                 age = ageMid[tSelect],
                 group = ageData1$group[tSelect],
                 score = scoreMid[tSelect],
                 slope = slopeMid[tSelect])
```

# Slope Fields

VectorFields1.R -- MARKER D

- Next we smooth the slope data set so that for every combination of age and score we have an expected slope.

```
slopeLoess <- loess(slope ~ age * score, data=ageSlopeFrame)

ageSlopeFrame2 <- data.frame(age=ageSlopeFrame$age,
                   group=ageSlopeFrame$group,
                   score=ageSlopeFrame$score,
                   slope=ageSlopeFrame$slope,
                   fitted=predict(slopeLoess))
```

# Slope Fields

## VectorFields1.R -- MARKER E

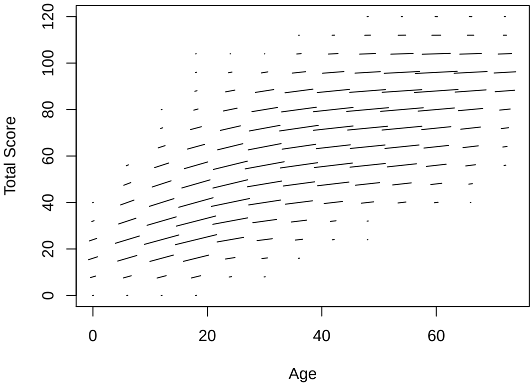
```
smoothX <- seq(0, 75, by=6)
smoothY <- seq(0, 120, by=8)
smoothP <- matrix(NA, length(smoothX), length(smoothY))
smoothZ <- matrix(NA, length(smoothX), length(smoothY))
h <- 10
for (i in 1:length(smoothX)) {
   x <- smoothX[i]
   t1x <- abs(x - ageSlopeFrame2$age)/h
   for (j in 1:length(smoothY)) {
      y <- smoothY[j]
      t1y <- abs(y - ageSlopeFrame2$score)/h
      t2 <- rep(0, length(t1x))
      t2[t1x < 1 & t1y < 1] <- 1/2
      smoothP[i,j] <- sum(t2)/(length(ageSlopeFrame2$age)*h)
      smoothZ[i,j] <- mean(ageSlopeFrame2$fitted[t1x < 1 &
                                                 t1y < 1])
}}
smoothP <- 100 * (smoothP/sum(smoothP))
```

# Slope Field Plots

VectorFields1.R -- MARKER F

```
pdf("VF1ARSlope1.pdf", height=5, width=6)
plot(c(0,220)/3, c(0,121), type="n" ,
     xlab="Age",
     ylab="Total Score")
scale <- 2.5
for (i in 1:length(smoothSlopeFrame$age)) {
   tx <- smoothSlopeFrame$age[i]
   ty <- smoothSlopeFrame$score[i]
   ta <- smoothSlopeFrame$slope[i]
   td <- smoothSlopeFrame$density[i]
   txinc <- (scale * td) / sqrt( 1 + (ta^2))
   tyinc <- ta * txinc
   lines(c(tx+txinc, tx-txinc), c(ty+tyinc, ty-tyinc))
}
dev.off()
```

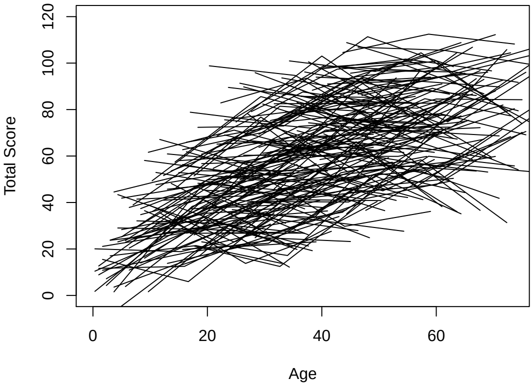
# Slope Field Plots



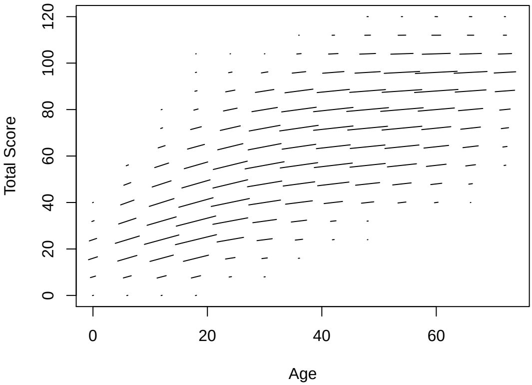
# Slope Field Plots

- The length of the line is proportional to the density of the data.
- The slope of the line at $(x, y)$ is the mean slope for that combination of $x$ and $y$.
- Slope field plots are a compact way to visualize the relationships between slope, level, and age.
- Slope fields are only locally parametric: Aggregation does not happen globally.

# Line Segment Plot



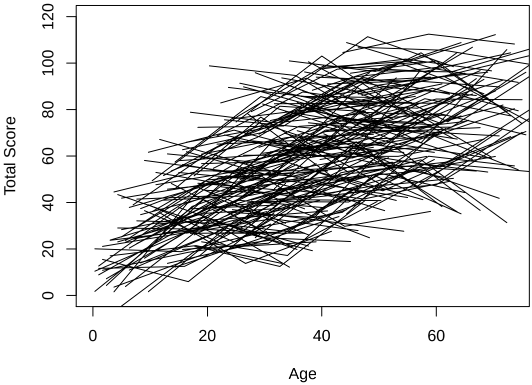
# Slope Field Plot



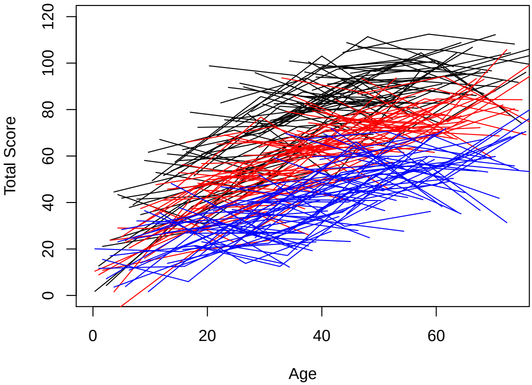
# Group Differences

- Sometimes you may want to compare groups and the way they change over time.
  - Treatment versus control in a longitudinal study.
  - Males versus females in growth curves.
  - Smokers versus non smokers over age.

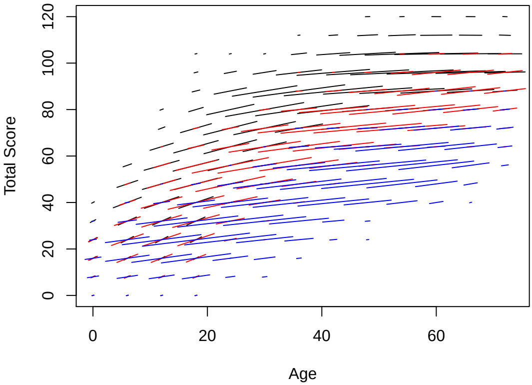
# Group Differences



# Group Differences



# Group Differences



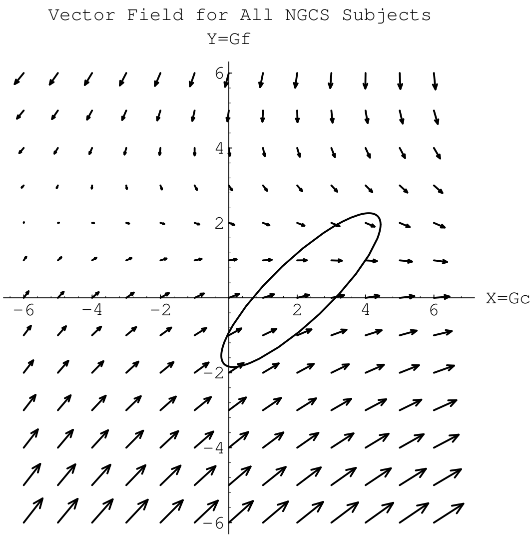
# Group Differences

- For ages younger than 20
  - Large group differences in slope.
  - Small group differences in level.
- Ages 30 and older.
  - Small group differences in slope.
  - Large group differences in level.
- The assumption we are making is *locality*: scores that are separated by a small distance within group have change that is more similar than scores that that are separated by a large distance.

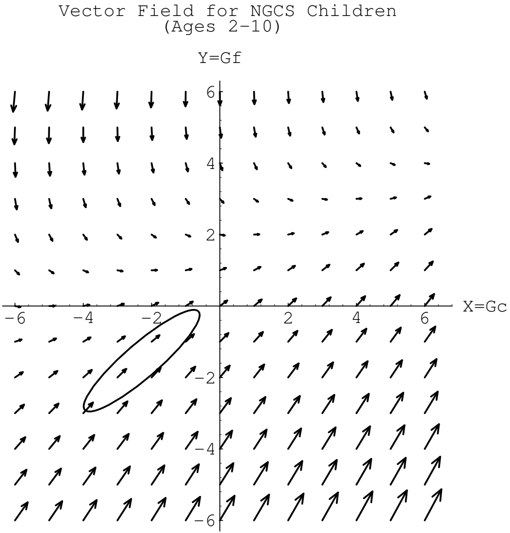
# Displaying Modeling Results with Vector Fields

- Vector fields can also be used to display model results.
- Again, we wish to separate the treatment of initial values and parametric modeling.
- The following data are from a large sample of people who took the WAIS at different ages and at multiple occasions of measurement.
- The model (developed by Jack McArdle, Aki Hamagami, and myself) was a test of Cattell's Investment Theory of Intelligence.
- We used a vector field plotted with Mathematica to visualize the model results.

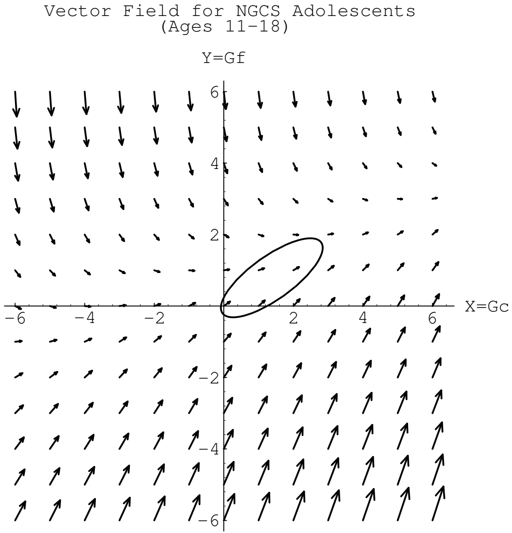
# NGCS Data All Ages



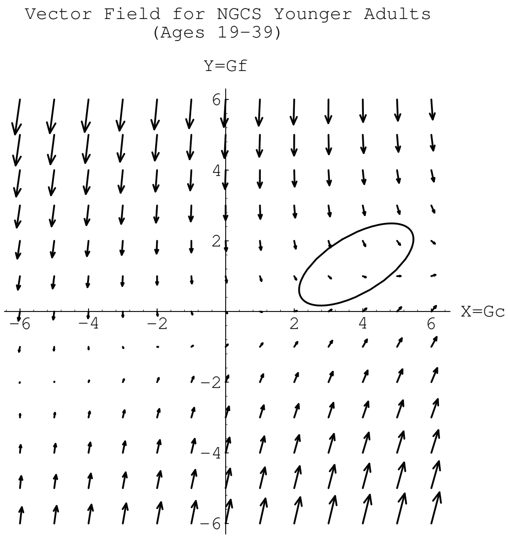
# NGCS Data Ages 2–10



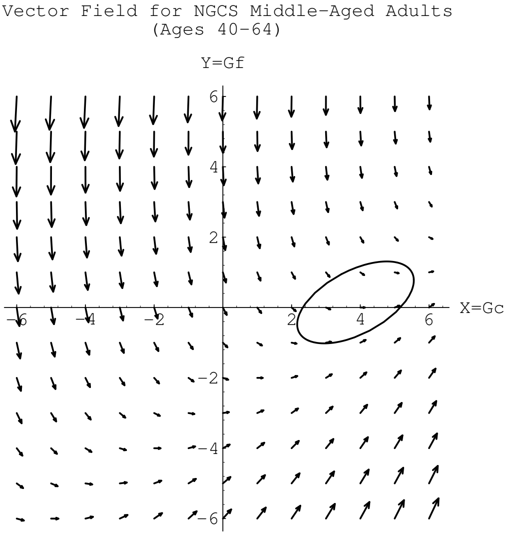
# NGCS Data Ages 11–18



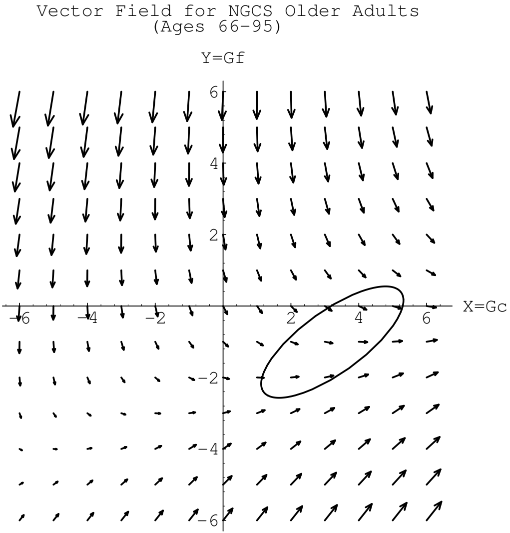
# NGCS Data Ages 19–39



# NGCS Data Ages 40–64



# NGCS Data Ages 65–95



Boker, S. M. (2015). Adaptive equilibrium regulation: A balancing act in two timescales. *Journal of Person-Oriented Research*, *1*(1–2), 99–109. (doi: 10.17505/jpor.2015.10)

Boker, S. M., & McArdle, J. J. (1995). Statistical vector field analysis applied to mixed cross–sectional and longitudinal data. *Experimental Aging Research*, *21*(1), 77–93.

Boker, S. M., & McArdle, J. J. (2005). Vector field plots. In P. Armitage & T. Colton (Eds.), *Encyclopedia of biostatistics (2nd edition)* (pp. 5700–5704). Chichester, UK: Wiley.