An Overview of Time Series Analysis
Global Temperatures since 1880

<table>
<thead>
<tr>
<th>Time</th>
<th>Global Temperature Deviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1880</td>
<td>-0.4</td>
</tr>
<tr>
<td>1900</td>
<td>-0.2</td>
</tr>
<tr>
<td>1920</td>
<td>0.0</td>
</tr>
<tr>
<td>1940</td>
<td>0.2</td>
</tr>
<tr>
<td>1960</td>
<td>0.4</td>
</tr>
<tr>
<td>1980</td>
<td>0.6</td>
</tr>
<tr>
<td>2000</td>
<td></td>
</tr>
</tbody>
</table>

Global Temperatures since 1880
Southern Oscillation Index (SOI): Index which measures sea level pressure – typically correspond to intensity of El Nino and La Nina.
Common Problems with Times Series Data

1. Temporal Correlation
2. Temporal Trends
3. Seasonal Variation
4. Heteroskedasticity (particularly for stock data)
Important Definitions for Time Series Data

Mean Function:

\[ \mu_t = \mathbb{E}(x_t) \]

Autocovariance Function:

\[ \gamma(t_1, t_2) = \text{Cov}(x_{t_1}, x_{t_2}) \]

Autocorrelation function:

\[ \rho(t_1, t_2) = \text{Corr}(x_{t_1}, x_{t_2}) \]
Important Definitions for Time Series Data

**Stationary Time Series**: Distribution of a time series is invariant to shifts in time,

\[
\text{Prob}(x_{t_1} \leq c_1, x_{t_2} \leq c_2, \ldots, x_{t_T} \leq c_T) = \text{Prob}(x_{t_1+h} \leq c_1, x_{t_2+h} \leq c_2, \ldots, x_{t_T+h} \leq c_T)
\]

**Weakly Stationary Time Series**:  
1. Mean does not change in time: \( \mu_t = \mu \)  
2. Covariance only depends on separation:  
   \[
   \gamma(t_1, t_2) = \gamma(t_1 - t_2) = \gamma(h)
   \]
Examples Autocorrelation Functions

Global Temperatures

NYSE

Southern Oscillation Index
What would you do with the data: $x_1, \ldots, x_T$?
Regression for Time Series
Regression for Time Series

**Big Idea:** Treat time \((t)\) as a “covariate” in a regression model.
Gaussian Process Regression for Time Series

Big Idea: Treat $x_{t_1}, \ldots, x_{t_T}$ as a realization of a GP over the time domain $t_1, \ldots, t_T$.

$$(x_{t_1}, \ldots, x_{t_T})' \sim \mathcal{N}(\beta_0 + \text{ns}(t_1, \ldots, t_T, \text{df} = 5)\beta, \Sigma)$$

$$\Sigma = \text{Matern}(\nu = 2, \phi)$$
How would you account for the seasonality in the SOI data?
Regression for Seasonal Time Series

**Big Idea:** Use a seasonal function (like a cosine)

\[ x_t = \beta_0 + \alpha \cos(2\pi \omega t + \phi) + \epsilon_t \]

\[ = \beta_0 + \alpha \cos(\phi) \cos(2\pi \omega t) + \alpha \sin(\phi) \sin(2\pi \omega t) + \epsilon_t \]

\[ = \beta_0 + \beta_1 \cos(2\pi \omega t) + \beta_2 \sin(2\pi \omega t) + \epsilon_t \]

\( \alpha = \text{Amplitude: Seasonal highs/lows} \)

\( \omega = \text{Frequency in cycles per time period} \)

\( \phi = \text{Phase shift:} \)
Regression for Seasonal Time Series

$\alpha=2, \omega=1/12, \phi=\pi/12$

$\alpha=2, \omega=1/24, \phi=\pi/12$

$\alpha=1, \omega=1/12, \phi=\pi$
Regression for Seasonal Time Series
Regression for
Seasonal Time Series

![ACF Plot]

Series resid

Lag
Regression for Seasonal Time Series
SARIMA Models
Autoregressive Processes

Big Idea: Treat past values as predictors

\[ \text{AR}(p) : x_t = \mu + \phi_1(x_{t-1} - \mu) + \cdots + \phi_p(x_{t-p} - \mu) + \epsilon_t \]

\[ = \alpha + \sum_{l=1}^{p} \phi_l x_{t-l} + \epsilon_t \]

\[ \alpha = \mu(1 - \sum_{l=1}^{p} \phi_l), \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \]

\[ \text{AR}(1) : x_t = \mu + \phi_1(x_{t-1} - \mu) + \epsilon_t \]

\[ \mathbb{E}(\text{AR}(1)) = \mu \]

\[ \rho(x_t, x_{t-l}) = \phi_1^l \rightarrow |\phi_1| < 1 \]

\[ \phi_1 < 0 \Rightarrow \text{Negative Correlation} \]

Side Note: You can get similar behavior using the Matern covariance function in a Gaussian process.
Autoregressive Processes

AR(1) $\phi = 0.9$

Time

AR(1) $\phi = -0.9$

Time
Moving Average Processes

**Big Idea:** Smooth-out Random Shocks

\[
\text{MA}(q): x_t = \mu + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t
\]

\[
\varepsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2)
\]

**MA(1):**

\[
\text{MA}(1): x_t = \mu + \theta_1 \varepsilon_{t-1} + \varepsilon_t
\]

\[
\mathbb{E}(\text{MA}(1)) = \mu
\]

\[
\rho(x_t, x_{t-l}) = \begin{cases} 
\frac{\theta_1}{1 + \theta_1^2} & \text{if } l = 1 \\
0 & \text{otherwise}
\end{cases}
\]

**Side Note:** You can get similar behavior using a tapered covariance function in a Gaussian process (e.g. Wendland).
Moving Average Processes

MA(1) $\theta = +0.9$

MA(1) $\theta = -0.9$
Big Idea: Regress on past values of the process but have correlated error terms:

\[
\text{ARMA}(p, q) : x_t = \mu + \sum_{l_1=1}^{p} \phi_{l_1}(x_{t-l_1} - \mu) + \sum_{l_2=1}^{q} \theta_{l_2} \epsilon_{t-l_2} + \epsilon_t
\]

\[
x_t = \beta_0 + \sum_{l=1}^{p} \beta_l x_{t-1} + \epsilon_t^*\]

\[
\epsilon_t^* = \sum_{l_2=1}^{q} \theta_{l_2} \epsilon_{t-l_2} + \epsilon_t \Rightarrow \text{Temporally Correlated Process}
\]

Side Note: You can essentially do this using autoregressive covariates in a GP
ARIMA Processes

Differencing in Finance:

\[
\text{profit}_t = P_t - P_{t-1}
\]

\[
\text{net return}_t = \frac{P_t - P_{t-1}}{P_{t-1}}
\]

\[
\text{gross return}_t = \frac{P_t}{P_{t-1}}
\]

\[
\text{risk}_t = \text{Prob}(\text{net } r_t < 0)
\]

Differencing Operator:

\[
\Delta x_t = x_t - x_{t-1}
\]

\[
\Delta^2 x_t = \Delta(\Delta x_t) = x_t - 2x_{t-1} + x_{t-2}
\]

and so on...
ARIMA Processes

**ARIMA:** Integrated ARMA – differences follow an ARMA

\[ \text{ARIMA}(p, d, q) : x_t = \Delta^d x_t \sim \text{ARMA}(p, q) \]

1. Taking differences essentially says that “slope” from time period to time period changes (type of non-linear regression).
SARIMA Processes

We build in seasonality via seasonal differences. For example, suppose the seasonal cycle is 1 year (12 months). Then, we’d want to model $y_t - y_{t-12}$.

SARIMA$(0,0,0)x(0,1,0)_{12}$:

$$y_t - y_{t-12} = \epsilon_t$$

$$\epsilon_t \sim \mathcal{N}(0, \sigma^2)$$

SARIMA$(0,1,0)x(0,1,0)_{12}$:

$$\Delta(y_t - y_{t-12}) = (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) = \epsilon_t$$

$$\epsilon_t \sim \mathcal{N}(0, \sigma^2)$$
SARIMA Processes

SARIMA(1,1,0)x(1,1,0)_{12}:
\[ d_t = (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \]
\[ d_t = \phi_1 d_{t-1} + \phi_2 d_{t-12} + \epsilon_t \]
\[ \epsilon_t \sim \mathcal{N}(0, \sigma^2) \]

SARIMA(0,1,1)x(0,1,1)_{12}:
\[ d_t = (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \]
\[ d_t = \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-12} + \epsilon_t \]
\[ \epsilon_t \sim \mathcal{N}(0, \sigma^2) \]
SARIMA Processes

SARIMA(1,1,1)x(1,1,1)_{12}:

\[ d_t = (y_t - y_{t-12}) - (y_{t-1} - y_{t-13}) \]
\[ d_t = \phi_1 d_{t-1} + \phi_2 d_{t-12} \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-12} + \epsilon_t \]
\[ \epsilon_t \sim \mathcal{N}(0, \sigma^2) \]
Model Selection in Time Series

Which time series model is the best?
- AR(p)
- MA(q)
- ARMA(p,q)
- ARIMA(p,d,q)
- SARIMA(p,d,q)x(P,D,Q)_s

How do we compare time series models?
- AIC or BIC or DIC
Do we make any assumption in fitting time series models?
1. Linearity in time
   • No
2. Independence of Residuals
   • Yes
3. Normality
   • Yes
4. Equal Variance of Residuals
   • Yes

How do we check these assumptions?
1. ACF Plot
2. QQ plot or histogram
3. Resids vs. Fitted Values
Forecasting

Forecasting = Prediction (everyone has to come up with their own name for things just to be confusing)

\[
\text{ARMA}(1, 1) : x_{T+1} \mid x_T = \mu + \phi(x_T - \mu) + \theta \epsilon_T + \epsilon_{T+1} \\
\epsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \\
x_{T+1} \sim \mathcal{N} \left( \mu + \phi(x_T - \mu), \sigma^2(1 + \theta^2) \right)
\]
Forecasting

Forecasting = Prediction (everyone has to come up with their own name for things just to be confusing)

ARMA(1, 1) : \( x_{T+2} \mid x_{T+1} = \mu + \phi(x_{T+1} - \mu) + \theta \epsilon_{T+1} + \epsilon_{T+2} \)

\( \epsilon_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \)

\( x_{T+1} \sim \mathcal{N}(\mu + \phi(x_T - \mu), \sigma^2(1 + \theta^2)) \)

\( x_{T+2} \mid x_T \sim \mathcal{N}(\mu + \phi^2(x_T - \mu), \sigma^2(1 + \theta^2 + \phi^2 \theta^2 + \phi^2)) \)

And so on….
Forecasting

Side Note: You can get similar error behavior using a Gaussian process.
library(astsa)
ts.model <- sarima(series,p,d,q,P,D,Q,s)
preds <- sarima.for(series,n.ahead=num.forecasts,p,d,q,P,D,Q,s)
pred.ints <- preds$pred±qt(0.975,df=length(series)-length(coef(ts.model $fit)))*preds$se
fitted.values <- series-ts.model$fit$residuals
plot(ts.model$fit$residuals,fitted.values)
hist(ts.model$fit$residuals)
How would you model heteroskedasticity?
Regression for Std. Deviation

Setup:
\[ x_t = \mu_t + \epsilon_t \]
\[ \epsilon_t = \sigma_t z_t \]
\[ z_t \overset{iid}{\sim} \mathcal{N}(0, 1) \]
\[ \text{Var}(x_t) = \sigma_t^2 \quad \text{Depends on time} \]

Std. Deviation Regression:
\[ \log(\sigma_t) = f(t) \]
E.g., \[ f(t) = \beta_0 + \beta_1 t \]
\[ f(t) = \beta_0 + \text{Natural Spline}(t) \]
Regression for Std. Deviation

Setup:
\[ x_t = \mu_t + \epsilon_t \]
\[ \epsilon_t = \sigma_t z_t \]
\[ z_t \sim iid \mathcal{N}(0, 1) \]
\[ \text{Var}(x_t) = \sigma_t^2 \quad \text{Depends on time} \]

GP Reg. for Std. Deviation:
\[ \log(\sigma_1, \ldots, \sigma_T) \sim \text{GP}(\mu_\sigma, \gamma^2 \rho(\cdot)) \Rightarrow \mathcal{N}(\mu_\sigma 1, \gamma^2 \mathbf{R}) \]
\[ \log(\hat{\sigma}_1, \ldots, \hat{\sigma}_T) = \max \left[ \text{Likelihood} \times \mathcal{N}(\mu_\sigma 1, \gamma^2 \mathbf{R}) \right] \]
ARCH Models

ARCH(1): Autoregressive, conditionally heteroskedastic

$$\varepsilon_t = \left( \sqrt{\alpha_0 + \alpha_1 \varepsilon_{t-1}^2} \right) z_t \Rightarrow \text{Var}(x_t) = \left( \alpha_0 + \alpha_1 \varepsilon_{t-1}^2 \right), \alpha_0, \alpha_1 > 0$$

**Intuition:** If large error at previous time, then variance now increases.

ARCH(q):

$$\varepsilon_t = \left( \sqrt{\alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2} \right) z_t \Rightarrow \text{Var}(x_t) = \left( \alpha_0 + \sum_{i=1}^{q} \alpha_i \varepsilon_{t-i}^2 \right), \alpha_i > 0 \forall i$$
ARCH Models

Major shortcoming of ARCH: Inflated variance is short lived.

Simulated Path of Series

Simulated Path of Conditional Sigma
GARCH Models

GARCH(p,q):

\[ \epsilon_t = z_t \sqrt{\alpha_0 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2} \]

\[ \Rightarrow \sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 \]

\[ \alpha_0, \alpha_i, \beta_i > 0 \]
GARCH Models

GARCH(1,1):

Simulated Path of Series

Simulated Path of Conditional Sigma
Regime Switching Models

Big Idea: Time series changes “states” (e.g. growth, recession, stagnation, etc.) across time and behavior is state-specific. Also called a Markov-Switching Model.

\[ x_t \mid s_t = k \sim \mathcal{N}(\mu_{kt}, \sigma_k^2) \]

\[ s_t \in \{1, \ldots, K\} : \text{Latent (hidden) State} \]

\[ \mu_{kt} = \text{State Specific Mean} \]

\[ \sigma_k^2 = \text{State Specific Variance} \]

\[ s_t \sim \text{Markov Chain(}\Xi) \]

\[ \Xi = \{\xi_{k_1, k_2}\} = \text{Prob}(s_t = k_2 \mid s_{t-1} = k_1) \]

\[ \pi_k = \text{Prob}(s_t = k) : \text{Stationary Probability} \]

\[ \Rightarrow x_t \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_{kt}, \sigma_k^2) \]

Mixture Model
Regime Switching Models

Code for plotting a Gaussian Mixture:

```r
## Parameters for a Gaussian Mixture Model
mu <- c(-3,0,3)
s <- c(1,1,1)
prob <- c(1/3,1/3,1/3)

## Function to Calculate Density
mix.dens <- function(x){
  as.numeric(dnorm(matrix(x,ncol=length(mu),nrow=length(x)),
  mean=matrix(mu,nrow=length(x),ncol=length(mu),byrow=TRUE),
  sd=matrix(s,nrow=length(x),ncol=length(mu),byrow=TRUE))%*%prob)
}

## Plot Density
xseq <- seq(mu[1]-3*s[1],mu[3]+3*s[3],length=10000)
ylim <- range(c(dnorm(xseq,mu[1],s[1]),dnorm(xseq,mu[2],s[2]),dnorm(xseq,mu[3],s[3]),mix.dens(xseq))
plot(xseq,mix.dens(xseq),type="l",lwd=3,xlab="x",ylab='Mixture(x)',ylim=ylim)
lines(xseq,dnorm(xseq,mu[1],s[1]),col="red")
lines(xseq,dnorm(xseq,mu[2],s[2]),col="blue")
lines(xseq,dnorm(xseq,mu[3],s[3]),col="green")
```
Code for Drawing from a Gaussian Mixture:

```r
## Parameters for a Gaussian Mixture Model
mu <- c(-3,0,3)
s <- c(1,1,1)
prob <- c(1/3,1/3,1/3)

## Draw from a Gaussian Mixture
n <- 10000
z <- sample(1:length(mu),n,replace=TRUE) # Draw which component
draws <- rnorm(n,mean=mu[z],sd=s[z]) # Draw from specific component

## Plot a Histogram of draws with density curve
plot(xseq,mix.dens(xseq),type="l",lwd=3,xlab="x",ylab='Mixture(x)') # see previous page’s code
hist(draws,prob=TRUE,add=TRUE)
```
Inference for Regime Switching Models:

1. Bayesian – Gibbs sampler with no MH steps (easy)
2. Frequentist – Expectation-Maximization (EM) Algorithm (easy if you know the transition probabilities)