Outline

1. Preliminaries
   - Software Installation
     - R Help
     - Importing Data Sets into R
       - Importing Data from the Internet
       - Importing Data from Your Computer
     - Using Data Available in R
2. Data Management
3. Plots of Time Series in R
4. Why Time Series Analysis?
5. Ad-Hoc Methods for Time Series Analysis
6. Autoregressive Integrated Moving Average (ARIMA)
7. Online Resources for R
8. Upcoming Mini-Courses
9. Exercises

Installing R on a Mac

1. Go to http://cran.r-project.org/ and select MacOS X
2. Select to download the latest version: 2.8.1 (2008-12-22)
3. Install and Open. The R window should look like this:
R Help

For help with any function in R, put a question mark before the function name to determine what arguments to use, examples and background information.

```
?plot
```

Importing Data

Data from the Internet

When downloading data from the internet, use `read.table()`. In the arguments of the function:

- `header`: if TRUE, tells R to include variables names when importing
- `sep`: tells R how the entries in the data set are separated
  - `sep=",":` when entries are separated by COMMAS
  - `sep="\t":` when entries are separated by TAB
  - `sep="":` when entries are separated by SPACE

```
data<-read.table("http://www.stat.ucla.edu/~vlew/stat130a/datasets/twins.csv", header=TRUE, sep="",)
```

Importing Data from Your Computer

1. Check what folder R is working with now:
   ```
   getwd()
   ```

2. Tell R in what folder the data set is stored (if different from (1)). Suppose your data set is on your desktop:
   ```
   setwd("~/Desktop")
   ```

3. Now use the `read.table()` command to read in the data, substituting the name of the file for the website.

Using Data Available in R

1. To use a data set available in one of the R packages, install that package (if needed).

2. Load the package into R, using the `library()` function.
   ```
   library(alr3)
   ```

3. Extract the data set you want from that package, using the `data()` function. In our case, the data set is called `UN2`.
   ```
   data(UN2)
   ```
Creating Time Series Variables I
For Regular Time Series Data

**Definition: Regular Time Series**

A *regular* time series has equally spaced observations and no missing data.

To coerce the variable to be in time series format (recognized by R), use `ts()`:

- **Example 1: Monthly Data**
  ```
  time1 <- ts(1:10, frequency = 12, start = 1900)
  ```

Creating Time Series Variables II
For Irregular Time Series Data

- **Example 2: Quarterly Data Example**
  ```
  time2 <- ts(1:10, frequency = 4, start = 1900)
  ```

For handling missing data, please see Appendix A.
Creating Time Series Variables

For Irregular Time Series Data

**Case 1:** The time series does not have equally spaced observations.

1. Create a time variable that is not equally spaced:

```r
# Create a vector that is not equally spaced:
x = c(1:12, 18:23, 25)
set.seed(4302009)
# Create a vector for corresponding values of x:
y = rnorm(length(x))
# Find its starting and ending values
x.min = x[which.min(x)]; x.min
x.max = x[which.max(x)]; x.max
```

Irina Kukuyeva  ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC

Creating Time Series Variables II

**Case 2:** The time series has missing data.

1. Load the package **zoo**:

```r
library(zoo)
```

2. To coerce the variables to be in time series format (allowing missing data and recognized by R), use **zoo()**:

```r
out <- zoo(round(y.compl, 2), x.compl)
```

Irina Kukuyeva  ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC

Creating Time Series Variables III

For Irregular Time Series Data

1. Complete the series with missing observations:

```r
x.compl <- seq(from=x.min, to=x.max, by=1);
y.compl
y.compl[1+x-x[1]] <- y
round(y.compl, 2)
```

Now we are in Case 2.

Irina Kukuyeva  ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC

Creating Time Series Variables IV

For Irregular Time Series Data

1. If the time variable is a number, use **as.Date()**:

```r
x.date <- as.Date(x, origin="2000-01-05"); x.date
```

Irina Kukuyeva  ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC

Working with Dates I

Converting Numbers to Dates

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
-0.06 1.33 0.97 1.66 0.20 0.36 0.92 0.11 1.09 -0.56 -0.09 0.30 NA NA
[16] NA NA NA -0.38 2.01 0.58 -0.57 1.79 NA -0.29
```

Irina Kukuyeva  ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC
Creating Time Series Variables

Working with Dates I
Converting Dates to Numbers

If the \textit{time} variable is a date, use \texttt{as.integer}():

\begin{verbatim}
# Create the vector
x1 <- c("01/05/2009", "01/06/2009", "01/07/2009", "01/08/2009", "01/09/2009", "01/10/2009")

# Convert to Date
x.date1 <- as.Date(x1, format="%m/%d/%y")

# Convert to Integer
x.int <- as.integer(x.date1)
\end{verbatim}

\>[1] 18266 18267 18268 18269 18270 18271

Univariate Time Series Plots I

Approach 1

To plot one variables one at a time, use \texttt{plot}():

\begin{verbatim}
data(EuStockMarkets)
dax <- EuStockMarkets[, 1]
plot(dax)
\end{verbatim}

Multivariate Time Series Plots I

Approach 1

To plot more than variables one at a time, use \texttt{plot}():

\begin{verbatim}
data(airquality)
a <- airquality[, 1:3]
time <- ts(1:nrow(a), start=c(1973, 5), frequency=365)

# If your data is stored as a data frame,
# coerce it to be a matrix via \texttt{as.matrix}()
class(a)
a.mat <- as.matrix(a)

# Make a time series of the two (or more
variables)
\end{verbatim}
Multivariate Time Series Plots II
Approach 1

```r	name.zoo <- zoo(cbind(a.mat[, 1], a.mat[, 2]))
colnames(name.zoo) <- c("Ozone", "Solar")
### Plot the variables
plot(name.zoo)
```

Plots of Ozone and Solar

Multivariate Time Series Plots III
Approach 2

To plot more than variables one at a time, use `mvtsplot()`:

- Go to: http://www.biostat.jhsph.edu/~rpeng/RR/mvtsplot/
- Copy the relevant R Code and paste it into the R Console. Press ENTER.
- Plot your data
  ```r
  # After processing data as in Approach 1
  # Plot the variables
  mvtsplot(name.zoo)
  ```

Plots of Ozone and Solar

Multivariate Time Series Plots I
Approach 2

To plot more than variables one at a time, use `mvtsplot()`:

- Go to: http://www.biostat.jhsph.edu/~rpeng/RR/mvtsplot/
- Copy the relevant R Code and paste it into the R Console. Press ENTER.
- Plot your data
  ```r
  # After processing data as in Approach 1
  # Plot the variables
  mvtsplot(name.zoo)
  ```
To plot more than variables one at a time, use `xyplot()`:

```r
# After processing data as in Approach 1
# load both libraries:
library(lattice)
library(zoo)
data(EuStockMarkets)
z <- EuStockMarkets
xyplot(z, screen = c(1, 1, 1, 1), col = 1:4, strip = FALSE)
legend(1992, 5000, colnames(z), lty = 1, col = 1:4)
```

Irina Kukuyeva
ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC

---

Model Characteristics

1. **Ordinary Least Squares (OLS):**
   - Residuals have mean zero and constant variance.
   - Observations are independently, identically distributed.

2. **Autoregressive Integrated Moving Average (ARIMA):**
   - Residuals are have mean zero and constant variance.
   - Observations are correlated with one another and the autocorrelations do not change with time.

3. **Generalized Autoregressive Conditional Heteroskedasticity (GARCH):**
   - Residuals have mean zero and non-constant variance.
   - Observations are correlated with one another.
Time Series Analysis
Effects of Correlated Observations

**Warning**
Statistical procedures that assume independent and identically distributed observations are no longer valid in the case of time series analysis \(^a\).


---

**Ad-Hoc Methods for Time Series Analysis I**

**Definition: Ad-Hoc Method**
A procedure one can apply, but one that has no statistical background associated with it.

---

**Regression Model for Time Series Data I**

**Example: Rooms Data**

- **Step 1:** Import data into R and plot it.

```r
1 rooms <- scan("http://www.stat.ucla.edu/~jsanchez/rooms.txt")
2 plot(rooms, type="l")
```
Regression Model for Time Series Data II

Example: Rooms Data

- **Step 2:** There is non-constant variance in the observations – use a transformation.
  - Suggestions: logarithm, square root, quartic root, etc.

```
lnrooms <- log(rooms)
pplot(lnrooms, type="l")
```

Regression Model for Time Series Data III

Example: Rooms Data

- **Step 3:** Model the trend and analyze the residuals.

```
time = seq(1:length(lnrooms))
fit1 <- lm(lnrooms ~ time)
summary(fit1)
```

```
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.3299787 0.0212156 298.36 <2e-16 *** 
time 0.0027681 0.0002178 12.71 <2e-16 ***
---
Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1
Residual standard error: 0.1369 on 166 degrees of freedom
Multiple R-squared: 0.4933, Adjusted R-squared: 0.4902
F-statistic: 161.6 on 1 and 166 DF, p-value: < 2.2e-16
```

Regression Model for Time Series Data IV

Example: Rooms Data

- **Step 4:** Model the seasonal component and analyze the residuals.

```
D1 <- rep(c(1,0,0,0,0,0,0,0,0,0,0,0), (length(rooms)/12))
D2 <- rep(c(0,1,0,0,0,0,0,0,0,0,0,0), (length(rooms)/12))
D3 <- rep(c(0,0,1,0,0,0,0,0,0,0,0,0), (length(rooms)/12))
D4 <- rep(c(0,0,0,1,0,0,0,0,0,0,0,0), (length(rooms)/12))
D5 <- rep(c(0,0,0,0,1,0,0,0,0,0,0,0), (length(rooms)/12))
D6 <- rep(c(0,0,0,0,0,1,0,0,0,0,0,0), (length(rooms)/12))
```

Regression Model for Time Series Data V

Example: Rooms Data

```
D7 <- rep(c(0,0,0,0,0,0,0,0,0,0,0,0), (length(rooms)/12))
D8 <- rep(c(0,0,0,0,0,0,0,0,0,0,0,0), (length(rooms)/12))
D9 <- rep(c(0,0,0,0,0,0,0,0,0,0,0,0), (length(rooms)/12))
D10 <- rep(c(0,0,0,0,0,0,0,0,0,0,0,0), (length(rooms)/12))
D11 <- rep(c(0,0,0,0,0,0,0,0,0,0,0,0), (length(rooms)/12))

# fit the dummy variables and time
fit2 <- lm(lnrooms~time+D1+D2+D3+D4+D5+D6+D7+D8+D9+D10+D11)
```
Regression Model for Time Series Data VI

Example: Rooms Data

1. `summary(fit2)`

Coefficients:

| Estimate | Std. Error | t value | Pr(>|t|) |
|----------|------------|---------|----------|
| (Intercept) | 6.288e+00 | 6.427e-03 | 978.262 | < 2e-16 *** |
| time | 2.725e-03 | 3.379e-05 | 80.653 | < 2e-16 *** |
| D1 | -4.161e-02 | 8.016e-03 | -5.190 | 6.50e-07 *** |
| D2 | -1.121e-01 | 8.015e-03 | -13.984 | < 2e-16 *** |
| D3 | -8.446e-02 | 8.013e-03 | -10.540 | < 2e-16 *** |
| D4 | 3.983e-02 | 8.012e-03 | 4.972 | 1.74e-06 *** |
| D5 | 2.040e-02 | 8.011e-03 | 2.546 | 0.0119 * |
| D6 | 1.469e-01 | 8.010e-03 | 18.340 | < 2e-16 *** |
| D7 | 2.890e-01 | 8.009e-03 | 36.085 | < 2e-16 *** |
| D8 | 3.112e-01 | 8.009e-03 | 38.857 | < 2e-16 *** |
| D9 | 5.599e-02 | 8.008e-03 | 6.991 | 7.63e-11 *** |
| D10 | 3.954e-02 | 8.008e-03 | 4.938 | 2.02e-06 *** |
| D11 | -1.122e-01 | 8.008e-03 | -14.013 | < 2e-16 *** |

Residual standard error: 0.02119 on 155 degrees of freedom
Multiple R-squared: 0.9887, Adjusted R-squared: 0.9878
F-statistic: 1127 on 12 and 155 DF, p-value: < 2.2e-16

Irina Kukuyeva
ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC

Regression Model for Time Series Data VII

Example: Rooms Data

1. `plot(fit2$residuals, type="l")`

Irina Kukuyeva
ikukuyeva@stat.ucla.edu
Introduction to Time Series in R UCLA SCC

Decomposition of Time Series

Goal

Decompose a time series into trend, seasonal and irregular components.

Definition: Additive Model

\[ Y = T + S + C + I \]

where \( Y \) is univariate observations, \( T \) is trend, \( S \) is the seasonal component and \( C \) is the cyclical component.

- Use the additive model if the plot of the observations does not show increasing amplitudes, i.e. constant mean.
Decomposition of Time Series II

Definition: Multiplicative Model

\[ Y = (T)(S)(C)(I) \], where \( Y \) is univariate observations, \( T \) is trend, \( S \) is the seasonal component and \( C \) is the cyclical component.

- Use the multiplicative model if the plot of the observations shows increasing amplitudes, i.e. non-constant mean.

AdHoc

Additive Decomposition Model I
Example: Rooms Data

- **Steps 1 and 2:** See Regression Model for Time Series Data.
- **Step 3:** Perform the decomposition on the data via \( \text{stl()} \):

```
1 library(stats)
2 lnrooms.ts<-ts(log(rooms), frequency=12, start=c(1977,1))
3 fit.stl<-stl(lnrooms.ts, "periodic")
4 plot(fit.stl)
```

Additive Decomposition Model II
Example: Rooms Data

Additive Decomposition Model III
Example: Rooms Data

- **Step 4:** Check that the residuals are not correlated via the Autocorrelation Function (ACF), use \( \text{acf()} \).

Definition: Autocorrelation Function (ACF)

Measure of linear "dependence between variables in the a time series" \( k \) units apart.

Additive Decomposition Model IV
Example: Rooms Data

```r
# save the residuals from the decomposition
resid <- fit.stl$time.series[, 3]
acf(resid)
```

Violation 1: Non-constant Variance I
Variable Transformations

If the variable has non-constant variance, to resolve the problem, you can:

1. Use a logarithm, square root, or other transformations to stabilize the variance:
   ```r
   beer = matrix(scan("http://www.stat.ucla.edu/~jsanchez/beer.txt", n=153), byrow=TRUE)
   beer.ts = ts(beer)
   plot(beer.ts)
   plot(sqrt(beer.ts))
   ```

2. If that does not work, use GARCH.
## Violation 2: There is a Trend I

**Autocorrelation Function**

Recall that to compute the ACF, use `acf()`:

```r
acf(dax)
```

The figure on the right shows a trend.

*Note:* If there is missing data, when computing the ACF, include the argument `na.action=na.pass` in the function call.

## Violation 2A: There is a Trend I

**Differencing the Data**

If there is a trend in the data, difference the variable via use `diff()`:

```r
# Need the variable to be of class "ts"

# compute the ACF on the difference

dax_diff <- diff(dax, lag=1, difference=1)

acf(dax_diff)
```

Then check that the differencing removed the trend.

## Violation 2B: There is a Seasonal Trend I

**Differencing the Data**

If there is a seasonal trend in the data, difference the variable via use `diff()`:

```r
# Need the variable to be of class "ts"

# compute the ACF on the difference

beer.sqrt <- sqrt(beer.ts)

acf(beer.sqrt)
```

## Violation 2B: There is a Seasonal Trend II

**Differencing the Data**

```r
beer.sqrt_diff <- diff(beer.sqrt, lag=1, difference=1)

acf(beer.sqrt_diff)
```
Violation 2B: There is a Seasonal Trend III
Differencing the Data

1. `beer.sqrt.diff4 <- diff(beer.sqrt.diff, lag=4)`
2. `acf(beer.sqrt.diff4)`

Violation 2C: There is Periodicity I

If there is periodicity in the data, employ a more advanced ARIMA model (to be covered in the Intermediate Time Series class).

1. `# Rooms Example`
2. `acf(diff(lnrooms, lag = 12))`

Violation 3: Observations are not Correlated I

If the observations are not correlated, then we cannot model the data via ARIMA. To test for correlation, use the Ljung-Box test.

- **H₀**: all the correlations are zero
- **H₁**: at least one of the correlations is not zero

1. `Box.test(beer.te, type="Ljung-Box")`

Box-Ljung test

data:  beer.ts
X-squared = 84.1058, df = 1, p-value < 2.2e-16

Proceed to Model If...

1. The variance is now constant.
2. There is no longer a trend.
3. The data is not white noise.
Definitions I

**Definition: Moving Average (MA) Model of Order \( q \)**

\[
x_t = \theta_1 u_1 + \theta_2 u_2 + ... + \theta_q u_q + \epsilon_t \tag{1}
\]

In this case, \( \theta_i \) are parameters, \( \epsilon_t \) is Gaussian white noise and \( u_i \) are unobserved variables (uncorrelated with one another).

**Definition: Autoregressive (AR) Model of Order \( p \)**

\[
x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + \epsilon_t \tag{2}
\]

In this case, \( \phi_i \) are constants (with \( \phi_p \neq 0 \)), \( \epsilon_t \) is Gaussian white noise (with constant mean and variance) and \( x_t \) is stationary with a mean of zero.

Definitions II

**Definition: Autoregressive Moving Average (ARMA) Model of Order \( (p, q) \)**

\[
x_t = \phi_1 x_{t-1} + \phi_2 x_{t-2} + ... + \phi_p x_{t-p} + \theta_1 u_1 + \theta_2 u_2 + ... + \theta_q u_q + \epsilon_t \tag{3}
\]

In this case, \( \phi_i \) are constants (with \( \phi_p \neq 0, \theta_q \neq 0 \)), \( \epsilon_t \) is Gaussian white noise (with constant mean and variance) and \( x_t \) is stationary with a mean of zero.

Definitions III

**Definition: Autoregressive Integrated Moving Average (ARIMA) Model of Order \( (p, d, q) \)**

An ARMA model that incorporates differencing (for non-stationary data).

When to Use Which Model I

**Definition: Partial Autocorrelation Function (PACF)**

Measures the "correlation between some components of the series, eliminating the contribution of other components".

To compute the PACF, use `pacf()`:

```r
pacf(dax)
```

http://www.resample.com/xlminer/help/Time_Series/Timeseries_intro.htm
**When to Use Which Model**

Once the data is stationary, analyze the ACF and PACF simultaneously to determine an appropriate time series model. *

*Theoretically,*

<table>
<thead>
<tr>
<th>Model</th>
<th>ACF</th>
<th>PACF</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA of order $q$</td>
<td>Cuts off after lag $q$</td>
<td>Dies down</td>
</tr>
<tr>
<td>AR of order $p$</td>
<td>Dies down</td>
<td>Cuts off after lag $p$</td>
</tr>
<tr>
<td>ARMA of order $(p, q)$</td>
<td>Dies down</td>
<td>Dies down</td>
</tr>
</tbody>
</table>

*Note:* In practice, your ACF and PACF may not look like the theoretical one, because of the noise in the data.

---

**Example 1: MA** To simulate from various MA $^2$ models and plot the ACF and PACF of the results:

1. `library(stats)`
2. `ma1 <- arima.sim(list(order=c(0,0,1), ma=0.5), n=100)`
3. `par(mfrow=c(2,1))`
4. `acf(ma1, main=(expression("ACF of MA(1), \*theta\* = +0.5")))`
5. `pacf(ma1, main=(expression("PACF of MA(1), \*theta\* = +0.5")))`

---

**Example 2: AR** To simulate from various AR $^3$ models and plot the ACF and PACF of the results:

1. `library(stats)`
2. `ar1 <- arima.sim(list(order=c(1,0,0), ar=0.5), n=100)`
3. `par(mfrow=c(2,1))`
4. `acf(ar1, main=(expression("ACF of AR(1), \*phi\* = 0.5")))`
5. `pacf(ar1, main=(expression("PACF of AR(1), \*phi\* = 0.5")))`

---

$^2$Theoretical ACF cuts off after lag $q$ and PACF dies down.

---

Irina Kukuyeva
ikukuyeva@stat.ucla.edu

Introduction to Time Series in R

UCLA SCC
Example 3: ARMA To simulate from various ARMA \(^4\) models and plot the ACF and PACF of the results:

```
library(stats)
arima1 <- arima.sim(n = 100, list(ar = 0.5, ma = 0.5))
par(mfrow = c(2, 1))
acf(arima1, main = expression("ACF of ARMA(1,1), \(\phi = 0.5\), \(\theta = +0.5\)"))
pacf(arima1, main = expression("PACF of ARMA(1,1), \(\phi = 0.5\), \(\theta = +0.5\)"))
```

\(^3\)Theoretical ACF dies down and PACF cuts off after lag \(p\)

Example 4: ARIMA To simulate from various ARMA \(^5\) models and plot the ACF and PACF of the results:

```
library(stats)
arima1 <- arima.sim(list(order = c(1,1,1), ar = 0.5, ma = 0.5), n = 100)
par(mfrow = c(2, 1))
acf(arima1, main = expression("ACF of ARIMA(1,1,1), \(\phi = 0.5\), \(\theta = +0.5\)"))
pacf(arima1, main = expression("PACF of ARIMA(1,1,1), \(\phi = 0.5\), \(\theta = +0.5\)"))
```

\(^4\)Theoretical ACF and PACF die down
Example 1: Gross National Product Data

Preliminary Data Analysis

The plot of the data shows that there is a trend in the data; the ACF confirms this. As a result, we will take first differences of the data.

To download the data: http://research.stlouisfed.org/fred2/series/GNP/downloaddata?cid=106

- Date range: 1947-01-01 to 2002-08-31 (223 observations)
- Note: The observations are quarterly.

Save as .csv
- Save it to your Desktop
- Import into R using `read.table()`

Example 1: Gross National Product Data I

Removing Trend

The differenced plot of the data showed that there was non-constant variance. As a result, we will take the logarithm of the original data, then the first difference.

```r
data.logDiff <- diff(log(data[, 2]))
```
Example 1: Gross National Product Data I

Modeling the Data

The authors assume that the stationary conditions are now satisfied, and proceed to model the data. (We will also model this via GARCH.)

```r
par(mfrow=c(2,1))
acf(data.logDiff)
pacf(data.logDiff)
```

Example 1: Gross National Product Data II

Modeling the Data

Note: Because the subject matter is GNP, we are also interested in estimating the intercept for this model. It cannot be done under the current `arima()` framework in R. As a result, we will fit an MA(2) and AR(1), respectively to the logarithm of the original data.

There are two potential models:

- Assume that ACF cuts off after lag 2 and PACF dies down. This suggests that the difference of the logarithm of GNP follows an MA(2) or that the logarithm of GNP follows ARIMA(0,1,2).
- Assume that ACF dies down and PACF cuts off after lag 1. This suggests that the difference of the logarithm of GNP follows an AR(1) or that the logarithm of GNP follows ARIMA(1,1,0).

We will fit both models.
Example 1: Gross National Product Data I

Modeling the Data

```
1  # ARIMA(0,1,2):
2  data.fit1 <- arima(data.logDiff, order=c(0,0,2),
3                    method="ML", include.mean=TRUE); data.fit1

Coefficients:
      ma1    ma2     intercept
   0.3937  0.2820    0.0170
s.e. 0.0670  0.0544  0.0011

sigma^2 estimated as 0.0001008
log likelihood = 706.35, aic = -1404.69
```

Example 1: Gross National Product Data II

Modeling the Data

```
1  # ARIMA(1,1,0):
2  data.fit2 <- arima(data.logDiff, order=c(1,0,0),
3                            method="ML", include.mean=TRUE); data.fit2

Coefficients:
      ar1   intercept
    0.4603    0.0170
s.e. 0.0593    0.0013

sigma^2 estimated as 0.0001030
log likelihood = 703.96, aic = -1401.92
```

Diagnostic Check

ARIMA: Model Diagnostics I

An adequate model satisfies all of the following:

- The standardized residuals are independently identically distributed with zero mean and variance one. (If not, choose an alternative model.)
- The residuals normally distributed. (If not, use GARCH.)

To check the assumptions in R:

- To check for outliers and autocorrelations, use `tsdiag()`
- To test for normality of the residuals, look at their histogram or the Normal Q-Q plot.

To run the diagnostics on model 1,

```
1  tsdiag(data.fit1)
```

On the following slide, we can see that after the first 4 lags the residuals are autocorrelated with each other, as $p-value < 0.05$. As a result, the model is not adequate.
Example 1: Gross National Product Data II

Model Diagnostics

Example 2: US Presidents’ Approval Ratings I

Preliminary Data Analysis

Load the data into R:

```r
load(presidents)
# to make it easier to type, rename:
z <- presidents
plot(z)
acf(z, na.action=na.pass)
```

Removing Trend

The differenced plot of the data shows that the data is now stationary.

```r
z.diff <- diff(z)
acf(z.diff, na.action=na.pass)
```

Modeling the Data

We proceed to determine a model for the data:

```r
par(mfrow=c(2,1))
acf(z.diff, na.action=na.pass)
pacf(z.diff, na.action=na.pass)
```
Example 2: US Presidents’ Approval Ratings II

Modeling the Data

The potential model:

- Assume that ACF cuts off after lag 1 and PACF dies down.
  
  This suggests that the difference of the president’s approval rating follows an MA(1).

Note: Because the subject matter is approval rating, we are not interested in estimating the intercept for this model.

Example 2: US Presidents’ Approval Ratings I

Modeling the Data

To run the diagnostics on model 1,

```
1   tsdiag(z.fit1)
```

On the following slide, we can see that the residuals are not autocorrelated with each other, as $p-value > 0.05$.
Example 2: US Presidents’ Approval Ratings II
Model Diagnostics

![Standardized Residuals](image)

![ACF of Residuals](image)

![p values for Ljung-Box statistic](image)

There are three methods to test for normality of the residuals.

**Method 1:** we can look at their histogram (the sample density is in red, the theoretical normal density in blue):

```r
par(mfrow=c(1,1))
hist(z.fit1$residuals, prob=T, ylim=c(0, 0.05))
lines(density(z.fit1$residuals, na.rm=TRUE), col="red")
mu <- mean(z.fit1$residuals, na.rm=TRUE)
sigma <- sd(z.fit1$residuals, na.rm=TRUE)
x <- seq(-30,40, length=100)
y <- dnorm(x, mu, sigma)
lines(x,y,lwd=2, col="blue")
```

**Method 2:** we can look at normal Q-Q plot of the residuals, which show that our distribution has heavier tails than the normal distribution.

```r
par(mfrow=c(1,1))
qqnorm(as.numeric(z.fit1$residuals))
qqline(as.numeric(z.fit1$residuals), col="red")
```
Example 2: US Presidents’ Approval Ratings II
Model Diagnostics: Method 2

Normal Q-Q Plot

Example 2: US Presidents’ Approval Ratings I
Model Diagnostics: Method 3

Method 3: we can look at output from Shapiro-Wilk normality test:
\( H_0 \): the residuals are normally distributed
\( H_a \): the residuals are not normally distributed.
At the 5% significance level, we cannot reject the null hypothesis that the residuals are normally distributed.

![](image)

```
shapiro.test(z.fit1$residuals)
```

Shapiro-Wilk normality test
data: z.fit1$residuals
W = 0.9831, p-value = 0.1788

Forecasting I

To forecast the time series \( h \) steps ahead, use predict():

```
# predict 3 quarters ahead:
predict(z.fit1, 3) -&gt; pred
# plot the predictions and their respective bounds:
seq(from=1975.25, by=0.25, length=3)
plot(z.diff, xlab=c(1945,1980), type="b")
points(n, as.numeric(pred$pred), col="red", type="b")
points(n, (as.numeric(pred$pred)+2*as.numeric(pred$se)), col="green", type="b")
points(n, (as.numeric(pred$pred)-2*as.numeric(pred$se)), col="green", type="b")
```
Forecasting I

To convert the predictions to the original units, use: `diffinv()`:

```r
diffinv(pred$pred, lag=1, xi=24)
```

# Original Data:

<table>
<thead>
<tr>
<th>Qtr1</th>
<th>Qtr2</th>
<th>Qtr3</th>
<th>Qtr4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1973</td>
<td>68</td>
<td>44</td>
<td>40</td>
</tr>
<tr>
<td>1974</td>
<td>28</td>
<td>25</td>
<td>24</td>
</tr>
</tbody>
</table>

# Forecasted Data:

<table>
<thead>
<tr>
<th>Qtr1</th>
<th>Qtr2</th>
<th>Qtr3</th>
<th>Qtr4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1974</td>
<td>24.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1975</td>
<td>24.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>24.06</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Online Resources for R

Download R:  [http://cran.stat.ucla.edu/](http://cran.stat.ucla.edu/)
Search Engine for R: [rseek.org](http://rseek.org)
R Reference Card:  
[http://cran.r-project.org/doc/contrib/Short-refcard.pdf](http://cran.r-project.org/doc/contrib/Short-refcard.pdf)
R Reference Card for Time Series:  
[http://www.statmethods.net/advstats/timeseries.html](http://www.statmethods.net/advstats/timeseries.html)
Time Series Analysis packages in R:  
[http://cran.r-project.org/web/views/TimeSeries.html](http://cran.r-project.org/web/views/TimeSeries.html)
Time Series Analysis with R Tutorial:  
[www.statoeck.wiso.uni-goettingen.de/veranstaltungen/zeitreihen/sommer03/ts_r_intro.pdf](http://www.statoeck.wiso.uni-goettingen.de/veranstaltungen/zeitreihen/sommer03/ts_r_intro.pdf)
UCLA Statistics Information Portal:  
[http://info.stat.ucla.edu/grad/](http://info.stat.ucla.edu/grad/)
UCLA Statistical Consulting Center:  
[http://scc.stat.ucla.edu](http://scc.stat.ucla.edu)
Upcoming Mini-Courses

- **This week:**
  - Intermediate R (April 30, Thursday)

- **Next week:**
  - Regression in R (May 5, Tuesday)
  - Basic R (May 7, Thursday)

- For a schedule of all mini-courses offered please visit [http://scc.stat.ucla.edu/mini-courses](http://scc.stat.ucla.edu/mini-courses).

Feedback Survey

PLEASE follow this link and take our brief survey: [http://scc.stat.ucla.edu/survey](http://scc.stat.ucla.edu/survey)

It will help us improve this course. Thank you.

References

- Grant Farnsworth. *Econometrics in R*. 2008. cran.r-project.org/doc/contrib/Farnsworth-EconometricsInR.pdf
Exercises in R

1. What model would you recommend for the beer data? (You do not have to implement it.)

2. For the GNP example, analyze the second proposed model for the data.