Economic Forecasting

Dynamic regression models

Sebastian Fossati

University of Alberta | E493 | 2023

Outline

Regression models

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t$$

Remarks:

- y_t modeled as function of k predictors $x_{1,t}, \ldots, x_{k,t}$
- in regression, we assume that ε_t is WN
- now we want to allow ε_t to be autocorrelated

Residuals and errors

Example: η_t = ARIMA(1,1,1)

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t,$$

$$(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t, \text{ where } \varepsilon_t \text{ is white noise.}$$

Remarks:

- be careful in distinguishing η_t from ε_t
- only the errors ε_t are assumed to be white noise
- in ordinary regression, η_t is assumed to be white noise, so $\eta_t = \varepsilon_t$

Estimation

If we minimize $\sum \eta_t^2$ (by using ordinary regression):

1. Estimated coefficients $\hat{\beta}_0, \dots, \hat{\beta}_k$ are no longer optimal as some information ignored.
2. Statistical tests associated with the model (e.g., t-tests on the coefficients) are incorrect.
3. p-values for coefficients usually too small.
4. AIC/BIC/etc of fitted models misleading.

Minimizing $\sum \varepsilon_t^2$ avoids these problems.

Regression with ARMA errors

$$y_t = \beta_0 + \beta_1 x_{1,t} + \dots + \beta_k x_{k,t} + \eta_t, \text{ where } \eta_t \text{ is an ARMA process.}$$

Remarks:

- all variables in the model must be stationary
- if we estimate a model whith non-stationary variables, the estimated coefficients can be incorrect
- difference variables until all stationary
- if necessary, apply same differencing to all variables

Regression with ARIMA errors

Any regression with an ARIMA(p,d,q) error can be rewritten as a regression with an ARMA error by differencing all variables d times.

Regression with ARIMA errors

Any regression with an ARIMA(p,d,q) error can be rewritten as a regression with an ARMA error by differencing all variables d times.

Original data:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t$$

$$\phi(B)(1-B)^d \eta_t = \theta(B)\varepsilon_t$$

Regression with ARIMA errors

Any regression with an ARIMA(p,d,q) error can be rewritten as a regression with an ARMA error by differencing all variables d times.

Original data:

$$\begin{aligned} y_t &= \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t \\ \phi(B)(1-B)^d \eta_t &= \theta(B)\varepsilon_t \end{aligned}$$

After differencing all variables:

$$\begin{aligned} y'_t &= \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + \eta'_t \\ \phi(B)\eta'_t &= \theta(B)\varepsilon_t \\ y'_t &= (1-B)^d y_t \end{aligned}$$

Regression with ARIMA errors

Model with ARIMA(1,1,1) errors:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t$$

$$(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t$$

Regression with ARIMA errors

Model with ARIMA(1,1,1) errors:

$$y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \eta_t$$
$$(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t$$

Equivalent to model with ARIMA(1,0,1) errors:

$$y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + \eta'_t$$
$$(1 - \phi_1 B)\eta'_t = (1 + \theta_1 B)\varepsilon_t$$

where $y'_t = y_t - y_{t-1}$, $x'_{t,i} = x_{t,i} - x_{t-1,i}$ and $\eta'_t = \eta_t - \eta_{t-1}$.

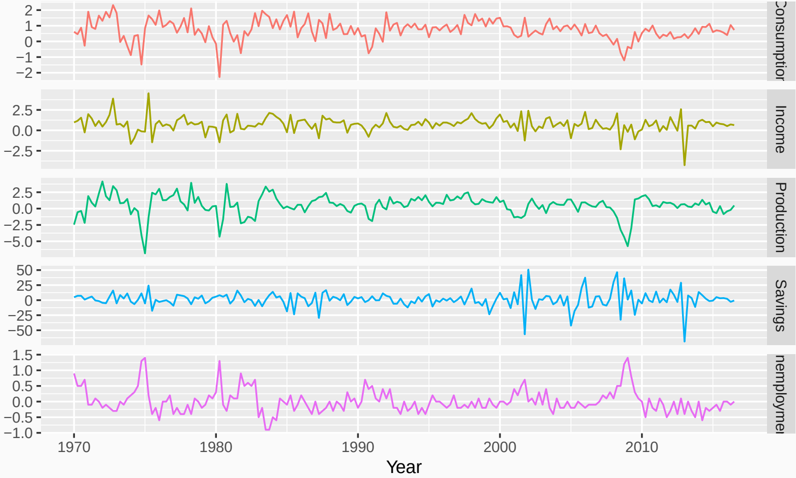
Model selection

Remarks:

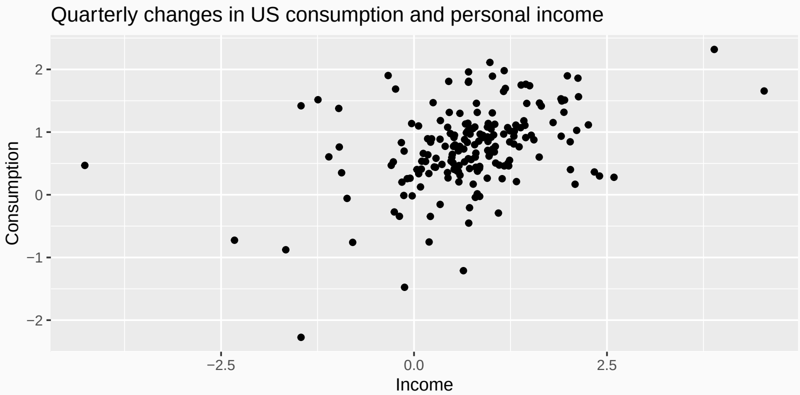
- check that all variables are stationary
- if not, apply differencing where appropriate (use the same differencing for all variables to preserve interpretability)
- fit regression model with ARIMA errors
- check that ε_t series looks like white noise
- repeat procedure for all subsets of predictors to be considered, and select model with lowest AIC/AICc value

Example: US consumption expenditure

```
autoplot(uschange[,1:5], facets = TRUE, colour=TRUE) +
  ylab("") + xlab("Year") + guides(colour="none")
```



US personal consumption and income



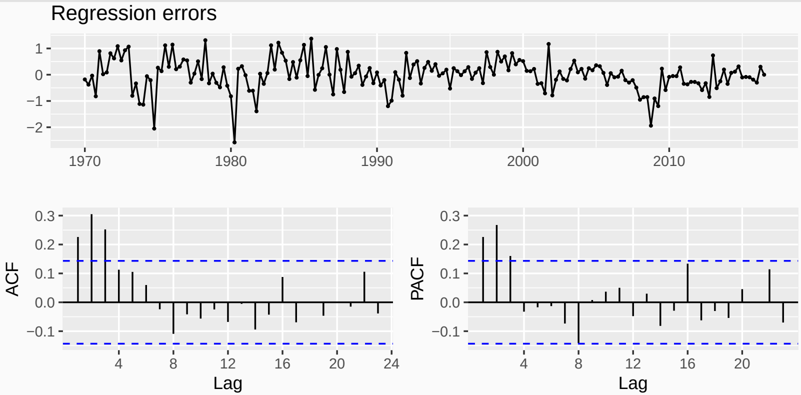
US personal consumption and income

```r
# fit regression
(fit <- auto.arima(uschange[,1], xreg=uschange[,2]))
```

```
## Series: uschange[, 1]
## Regression with ARIMA(1,0,2) errors
##
## Coefficients:
##          ar1      ma1     ma2  intercept    xreg
##        0.692  -0.576   0.198      0.599   0.203
## s.e.   0.116   0.130   0.076      0.088   0.046
##
## sigma^2 = 0.322:  log likelihood = -157
## AIC=325.9   AICc=326.4   BIC=345.3
```

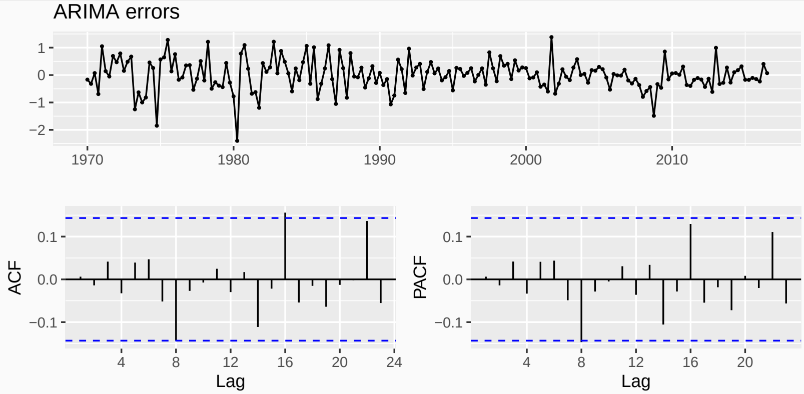
US personal consumption and income

```r
ggtsdisplay(residuals(fit, type='regression'),
  main="Regression errors")
```



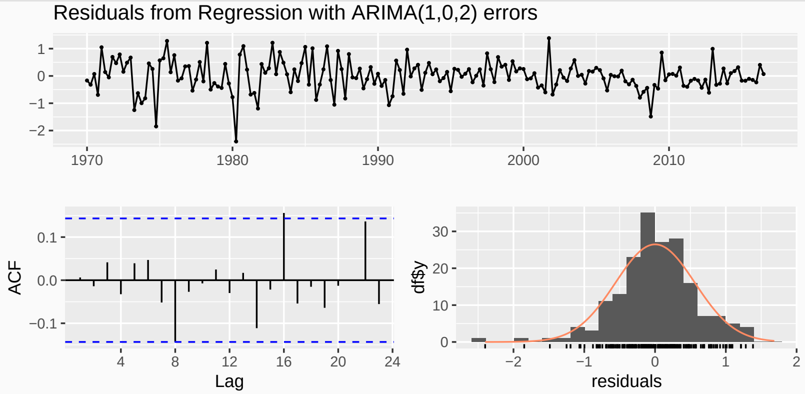
US personal consumption and income

```r
ggtsdisplay(residuals(fit, type='response'),
  main="ARIMA errors")
```



US personal consumption and income

```r
# check residuals
checkresiduals(fit, test=FALSE)
```

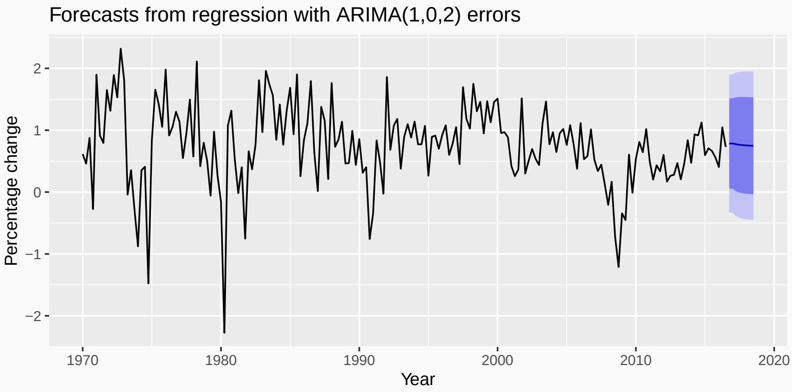


US personal consumption and income

```r
# check residuals
checkresiduals(fit, plot=FALSE)
```

```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(1,0,2) errors
## Q* = 5.9, df = 5, p-value = 0.3
##
## Model df: 3.   Total lags used: 8
```

US personal consumption and income



Forecasting

Remarks:

- to forecast a regression model with ARIMA errors, we need to forecast the regression part of the model and the ARIMA part of the model and combine the results
- some predictors are known into the future (e.g., time trend, dummies)
- separate forecasting models may be needed for other predictors
- forecast intervals ignore the uncertainty in forecasting the predictors

Outline

A convenient way of representing an economic time series y_t is through the so-called *trend-cycle decomposition*:

$$\begin{aligned} y_t &= TD_t + \eta_t \\ TD_t &= \text{deterministic trend} \\ \eta_t &= \text{random cycle/noise} \end{aligned}$$

For simplicity, assume

- $TD_t = \beta_0 + \beta_1 t$
- $\phi(B)\eta_t = \theta(B)\varepsilon_t$ is ARIMA process with $d \geq 0$

Nonstationary time series

Definitions:

- the series y_t is called **trend stationary** if the roots of $\phi(z) = 0$ are outside the unit circle
- the series y_t is called **difference stationary** if $\phi(z) = 0$ has one root on the unit circle and the others outside the unit circle

Difference stationary AR(2)

Let $\phi(B)\eta_t = \varepsilon_t$ with $\phi(B) = 1 - \phi_1 B - \phi_2 B^2$. Assume that $\phi(z) = 0$ has one root equal to unity and the other root real valued with absolute value less than 1. Factor $\phi(B)$ so that

$$\begin{aligned}\phi(B) &= (1-\phi^* B)(1-B) = \phi^*(B)(1-B)\\ \phi^*(B) &= 1-\phi^* B \text{ with } |\phi^*| < 1\end{aligned}$$

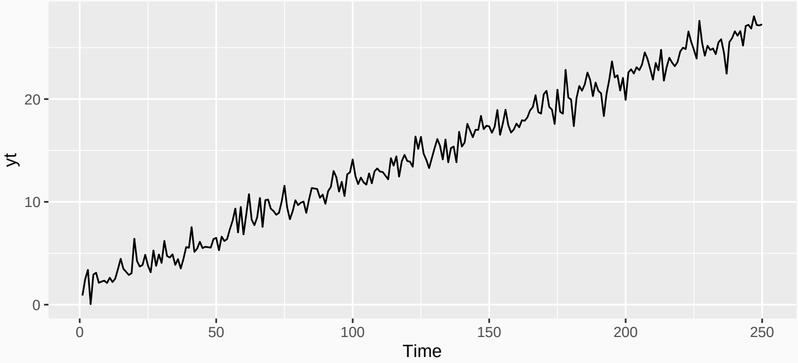
Then

$$\phi(B)\eta_t = (1-\phi^* B)(1-B)\eta_t = (1-\phi^* B)\eta_t'$$

so that η_t' follows an AR(1) process.

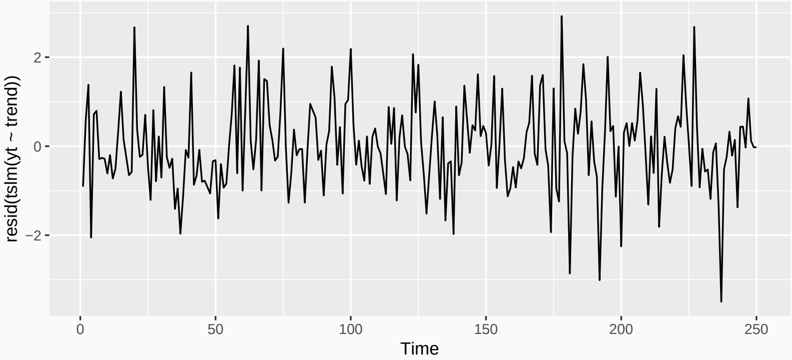
Trend stationary process

```
# trend stationary process
yt <- ts(2 + .1*seq(1:250) + rnorm(250))
autoplot(yt)
```



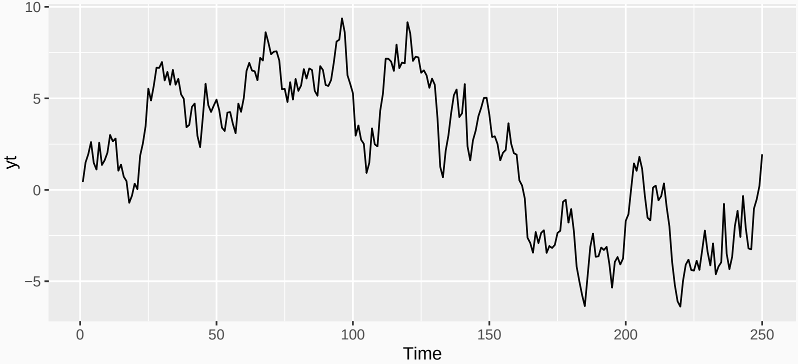
Trend stationary process

```
# trend stationary process
autoplot(resid(tslm(yt ~ trend)))
```



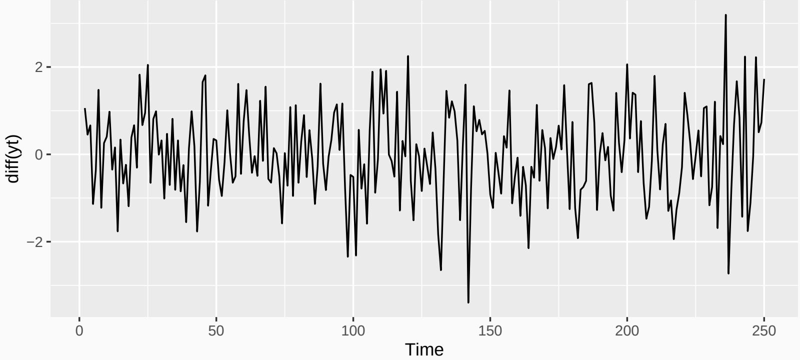
Difference stationary process

```
# random walk
yt <- ts(cumsum(rnorm(250)))
autoplot(yt)
```



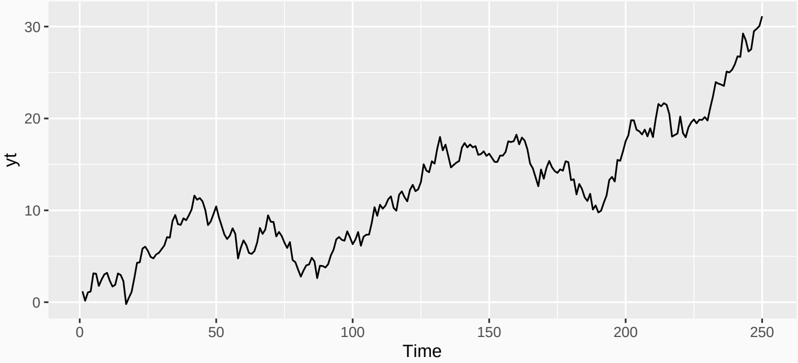
Difference stationary process

```
# random walk
autoplot(diff(yt))
```



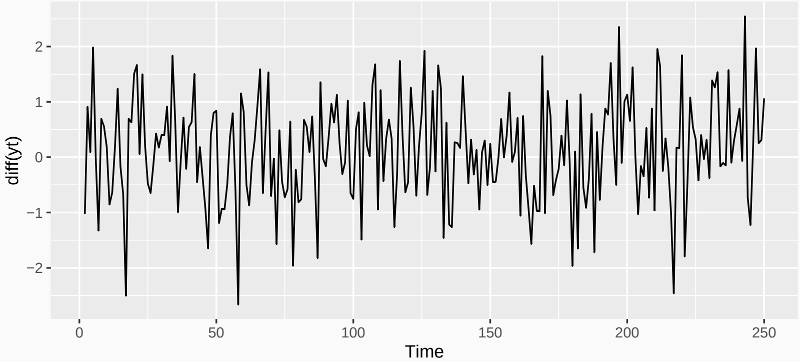
Difference stationary process

```r
# random walk w/ drift
yt <- ts(cumsum(.2 + rnorm(250)))
autoplot(yt)
```



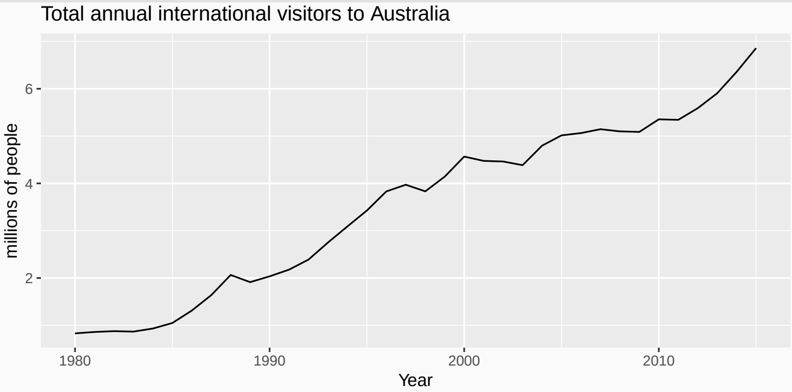
Difference stationary process

```
# random walk w/ drift
autoplot(diff(yt))
```



Example: International visitors

```
autoplot(austa) + xlab("Year") + ylab("millions of people") +
  ggtitle("Total annual international visitors to Australia")
```



Example: International visitors

Deterministic trend:

```
trend <- seq_along(austa)
(fit1 <- auto.arima(austa, d=0, xreg=trend))
```

```
## Series: austa
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2  intercept   xreg
##        1.113  -0.380      0.416  0.171
## s.e.  0.160   0.158      0.190  0.009
##
## sigma^2 = 0.0298:  log likelihood = 13.6
## AIC=-17.2   AICc=-15.2   BIC=-9.28
```

Example: International visitors

Deterministic trend:

```r
trend <- seq_along(austa)
(fit1 <- auto.arima(austa, d=0, xreg=trend))
```

```
## Series: austa
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2  intercept   xreg
##        1.113  -0.380      0.416  0.171
## s.e.   0.160   0.158      0.190  0.009
##
## sigma^2 = 0.0298:  log likelihood = 13.6
## AIC=-17.2   AICc=-15.2   BIC=-9.28
```

$$y_t = 0.42 + 0.17t + \eta_t$$

$$\eta_t = 1.11\eta_{t-1} - 0.38\eta_{t-2} + \varepsilon_t, \quad \varepsilon_t \sim \text{NID}(0, 0.0298).$$

Example: International visitors

Stochastic trend:

```
#
(fit2 <- auto.arima(austa, d=1))
```

```
## Series: austa
## ARIMA(0,1,1) with drift
##
## Coefficients:
##          ma1   drift
##        0.301   0.173
## s.e.   0.165   0.039
##
## sigma^2 = 0.0338:  log likelihood = 10.62
## AIC=-15.24   AICc=-14.46   BIC=-10.57
```

Example: International visitors

Stochastic trend:

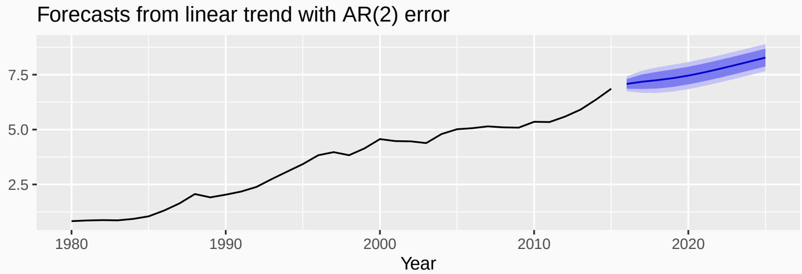
```
#
(fit2 <- auto.arima(austa, d=1))
```

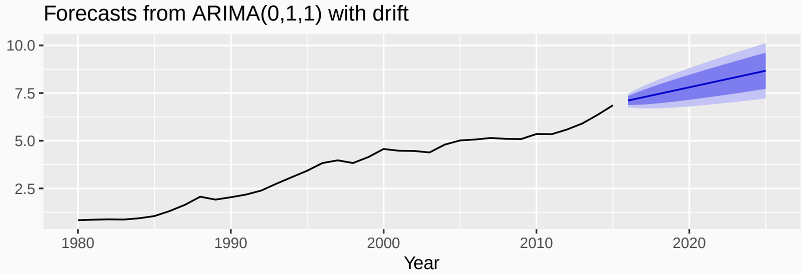
```
## Series: austa
## ARIMA(0,1,1) with drift
##
## Coefficients:
##          ma1   drift
##        0.301   0.173
## s.e.   0.165   0.039
##
## sigma^2 = 0.0338:  log likelihood = 10.62
## AIC=-15.24   AICc=-14.46   BIC=-10.57
```

$$y'_t = 0.17 + \eta'_t$$

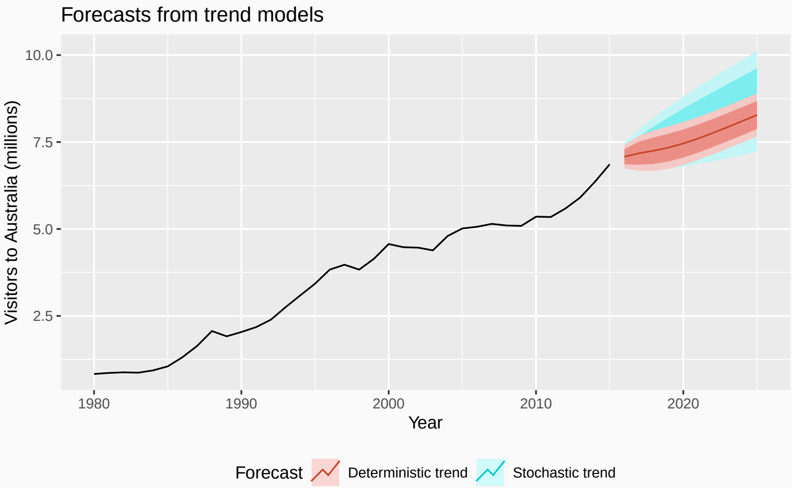
$$\eta'_t = 0.30\varepsilon_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim \text{NID}(0, 0.0338).$$

Example: International visitors





Example: International visitors



Forecasting with trend

Remarks:

- point forecasts are almost identical
- prediction intervals differ substantially
- stochastic trends have much wider prediction intervals because the errors are non-stationary
- be careful of forecasting with deterministic trends too far ahead

Outline

Unit root tests

These tests consider the following hypotheses:

$$H_0: \quad \phi = 1 \quad \text{(i.e., } \phi(z) = 0 \text{ has a unit root)}$$

$$H_1: \quad |\phi| < 1 \quad \text{(i.e., } \phi(z) = 0 \text{ has root outside unit circle)}$$

Remarks:

- the most popular of these tests are the Dickey-Fuller (ADF) test and the Phillips-Perron (PP) test
- the ADF and PP tests differ mainly in how they treat serial correlation in the test regressions

Statistical issues with unit root tests

Conceptually the unit root tests are straightforward. In practice, however, there are some difficulties:

- unit root tests generally have nonstandard and non-normal asymptotic distributions
- these distributions do not have convenient closed form expressions and critical values must be calculated using simulation methods
- the distributions are affected by the inclusion of deterministic terms (e.g. constant, time trend, dummy variables), and so different sets of critical values must be used for test regressions with different deterministic terms

Dickey-Fuller unit root tests

The Dickey-Fuller (ADF) test is based on estimating the test regression:

$$
\begin{aligned}
y_t &= TD_t + \phi y_{t-1} + \sum_{j=1}^{p} \psi_j y'_{t-j} + \varepsilon_t \\
TD_t &= \text{ deterministic terms} \\
y'_{t-j} &: \text{ captures serial correlation}
\end{aligned}
$$

The ADF t-statistic is

$$
ADF_t = t_{\phi=1} = \frac{\hat{\phi} - 1}{SE(\hat{\phi})}
$$

Dickey-Fuller unit root tests

Alternative formulation of the ADF test regression:

$$\begin{aligned} y'_t &= TD_t + \pi y_{t-1} + \sum_{j=1}^{p} \psi_j y'_{t-j} + \varepsilon_t \\ \pi &= \phi - 1 \end{aligned}$$

Under the null hypothesis $\phi = 1$, $\pi = 0$, and the ADF t-statistic is

$$ADF_t = t_{\pi=0} = \frac{\hat{\pi}}{SE(\hat{\phi})}$$

Trend cases

When testing for unit roots, it is crucial to specify the null and alternative hypotheses appropriately to characterize the trend properties of the data at hand:

- if the observed data does not exhibit an increasing or decreasing trend, then the appropriate null and alternative hypotheses should reflect this
- the trend properties of the data *under the alternative hypothesis* will determine the form of the test regression used
- the type of deterministic terms in the test regression will influence the asymptotic distributions of the unit root test statistics

Critical values

```
# standard normal quantiles
qnorm(c(0.01,0.05,0.10))
```

```
## [1] -2.326 -1.645 -1.282
```

```
# DF asymptotic critical values (requires urca package)
qunitroot(c(.01,.05,.1), trend="nc", statistic="t")
```

```
## [1] -2.565 -1.941 -1.617
```

```
qunitroot(c(.01,.05,.1), trend="c", statistic="t")
```

```
## [1] -3.430 -2.861 -2.567
```

```
qunitroot(c(.01,.05,.1), trend="ct", statistic="t")
```

```
## [1] -3.958 -3.410 -3.127
```

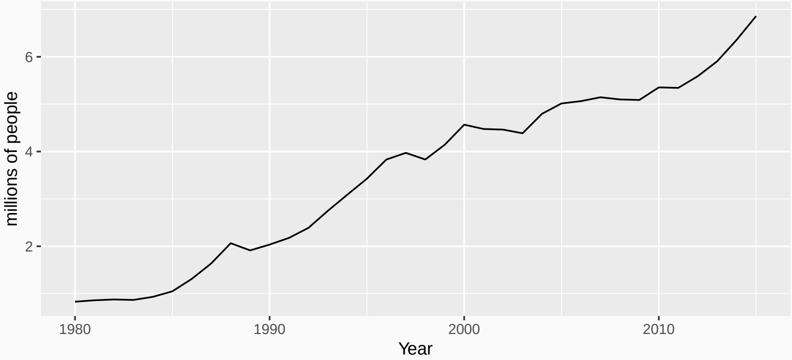
Choosing the lag length for the ADF test

An important practical issue for the implementation of the ADF test is the specification of the lag length p.

- if p is too small then the remaining serial correlation in the errors will bias the test
- if p is too large then the power of the test will suffer
- Monte Carlo experiments suggest it is better to err on the side of including too many lags
- model selection criteria can be used to select p (AIC better than BIC)

Example: International visitors

```r
# plot series
autoplot(austa) + xlab("Year") + ylab("millions of people")
```



Example: International visitors

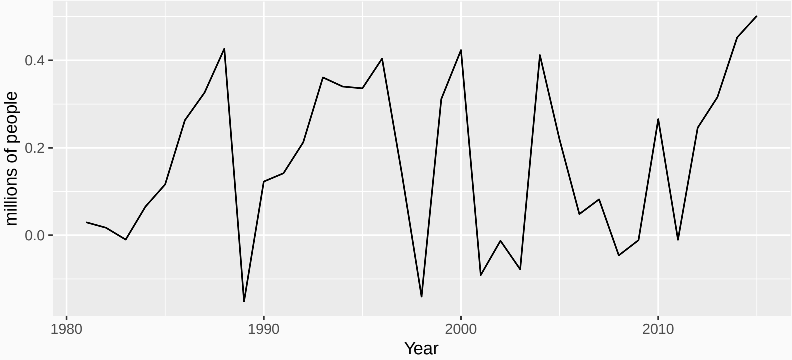
```r
# ADF test, levels
test1 <- ur.df(y = austa, type = 'trend', selectlags = "AIC", lags = 8)
summary(test1)
```

```
## Value of test-statistic is: -1.7707 4.092 1.88
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau3 -4.15 -3.50 -3.18
## phi2  7.02  5.13  4.31
## phi3  9.31  6.73  5.61
```

- $t_{\pi=0} = -1.771 > -3.18$ (10% critical value) so we fail to reject the unit root hypothesis for y_t

Example: International visitors

```
# plot series
autoplot(diff(austa)) + xlab("Year") + ylab("millions of people")
```



Example: International visitors

```r
# ADF test, first diff
test2 <- ur.df(y = diff(austa), type = 'drift', selectlags = "AIC", lags = 8)
summary(test2)
```

```
## Value of test-statistic is: -3.0961 5.0284
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.58 -2.93 -2.60
## phi1  7.06  4.86  3.94
```

- $t_{\pi=0} = -3.096 < -2.93$ (5% critical value) so we reject the unit root hypothesis for the first difference of y_t

How to determine d?

Unit root tests can be used to determine d:

1. test for a unit root in y_t - if H_0 is rejected, then $d = 0$
2. if H_0 is not rejected, then test for a unit root in y'_t - if H_0 is rejected, then $d = 1$
3. if H_0 is not rejected, then test for a unit root in y''_t - if H_0 is rejected, then $d = 2$
4. continue until H_0 is rejected

Outline

Lagged predictors

Sometimes a change in x_t does not affect y_t instantaneously.

Lagged predictors

Sometimes a change in x_t does not affect y_t instantaneously.

Examples

- y_t = sales, x_t = advertising
- y_t = stream flow, x_t = rainfall
- y_t = size of herd, x_t = breeding stock

Lagged predictors

Sometimes a change in x_t does not affect y_t instantaneously.

Examples

- y_t = sales, x_t = advertising
- y_t = stream flow, x_t = rainfall
- y_t = size of herd, x_t = breeding stock

Remarks:

- these are dynamic systems with input (x_t) and output (y_t)
- x_t is often a leading indicator
- there can be multiple predictors

Lagged predictors

The model includes present and past values of x_t.

$$y_t = a + \nu_0 x_t + \nu_1 x_{t-1} + \dots + \nu_k x_{t-k} + \eta_t, \text{ where } \eta_t \text{ is an ARIMA process.}$$

Lagged predictors

The model includes present and past values of x_t.

$$y_t = a + \nu_0 x_t + \nu_1 x_{t-1} + \cdots + \nu_k x_{t-k} + \eta_t, \text{ where } \eta_t \text{ is an ARIMA process.}$$

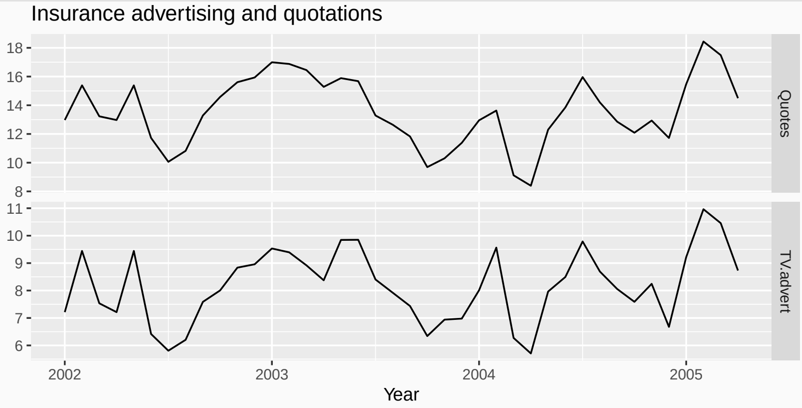
Rewrite model as

$$\begin{aligned} y_t &= a + (\nu_0 + \nu_1 B + \nu_2 B^2 + \cdots + \nu_k B^k) x_t + \eta_t \\ &= a + \nu(B) x_t + \eta_t \end{aligned}$$

- $\nu(B)$ is called a *transfer function* since it describes how change in x_t is transferred to y_t
- x can influence y, but y is not allowed to influence x

Example: Insurance quotes and TV adverts

```
# plot
autoplot(insurance, facets=TRUE) +
  xlab("Year") + ylab("") + ggtitle("Insurance advertising and quotations")
```



Example: Insurance quotes and TV adverts

```
# generate lagged predictors
Advert <-
  cbind(
    AdLag0 = insurance[,"TV.advert"],
    AdLag1 = lag(insurance[,"TV.advert"],-1),
    AdLag2 = lag(insurance[,"TV.advert"],-2),
    AdLag3 = lag(insurance[,"TV.advert"],-3)
  )
Advert <- head(Advert, NROW(insurance))
# restrict data so models use same fitting period
fit1 <- auto.arima(insurance[4:40,1], xreg=Advert[4:40,1], stationary=TRUE)
fit2 <- auto.arima(insurance[4:40,1], xreg=Advert[4:40,1:2], stationary=TRUE)
fit3 <- auto.arima(insurance[4:40,1], xreg=Advert[4:40,1:3], stationary=TRUE)
fit4 <- auto.arima(insurance[4:40,1], xreg=Advert[4:40,1:4], stationary=TRUE)
# evaluate fit
c(fit1$aicc,fit2$aicc,fit3$aicc,fit4$aicc)
```

```
## [1] 68.50 60.02 62.83 65.46
```

Example: Insurance quotes and TV adverts

```r
# fit model
(fit <- auto.arima(insurance[,1], xreg=Advert[,1:2], stationary=TRUE))
```

```
## Series: insurance[, 1]
## Regression with ARIMA(3,0,0) errors
##
## Coefficients:
##          ar1      ar2    ar3  intercept  AdLag0  AdLag1
##        1.412  -0.932  0.359      2.039   1.256   0.162
## s.e.   0.170   0.255  0.159      0.993   0.067   0.059
##
## sigma^2 = 0.217:  log likelihood = -23.89
## AIC=61.78   AICc=65.4   BIC=73.43
```

Example: Insurance quotes and TV adverts

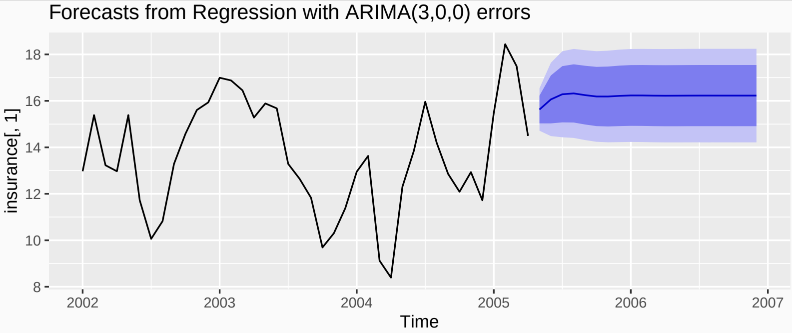
```r
# fit model
(fit <- auto.arima(insurance[,1], xreg=Advert[,1:2], stationary=TRUE))
```

```
## Series: insurance[, 1]
## Regression with ARIMA(3,0,0) errors
##
## Coefficients:
##          ar1     ar2    ar3  intercept  AdLag0  AdLag1
##        1.412  -0.932  0.359      2.039   1.256   0.162
## s.e.   0.170   0.255  0.159      0.993   0.067   0.059
##
## sigma^2 = 0.217:  log likelihood = -23.89
## AIC=61.78   AICc=65.4   BIC=73.43
```

$$y_t = 2.04 + 1.26x_t + 0.16x_{t-1} + \eta_t,$$
$$\eta_t = 1.41\eta_{t-1} - 0.93\eta_{t-2} + 0.36\eta_{t-3} + \varepsilon_t,$$

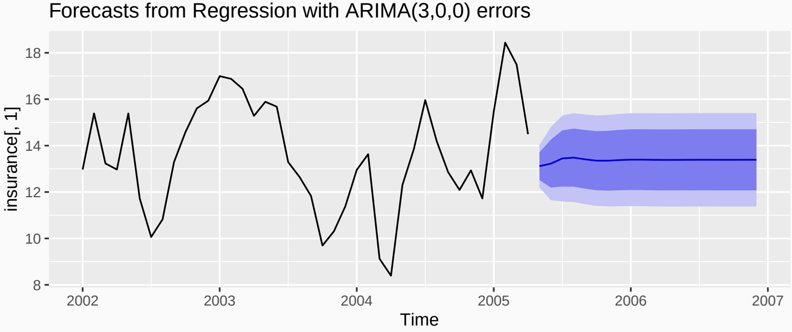
Example: Insurance quotes and TV adverts

```
fc <- forecast(fit, h = 20,
              xreg = cbind(
                AdLag0 = rep(10,20),
                AdLag1 = c(Advert[40,1], rep(10,19))
              ))
autoplot(fc)
```



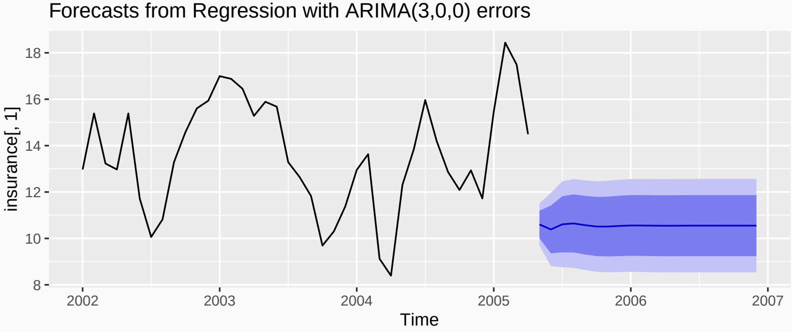
Example: Insurance quotes and TV adverts

```
fc <- forecast(fit, h = 20,
               xreg = cbind(
                 AdLag0 = rep(8,20),
                 AdLag1 = c(Advert[40,1], rep(8,19))
               ))
autoplot(fc)
```



Example: Insurance quotes and TV adverts

```
fc <- forecast(fit, h = 20,
                  xreg = cbind(
                    AdLag0 = rep(6,20),
                    AdLag1 = c(Advert[40,1], rep(6,19))
                  ))
autoplot(fc)
```



Outline

Granger and Newbold (1974, JoE)

Consider two independent and unrelated $I(1)$ processes.

```
# simulate two rw
yt <- ts(cumsum(rnorm(250)))
xt <- ts(cumsum(rnorm(250)))

# regression (1) in levels
reg1 <- tslm(yt ~ xt)

# regression (2) in differences
reg2 <- tslm(diff(yt) ~ diff(xt))
```

Granger and Newbold (1974, JoE)

Consider two independent and unrelated $I(1)$ processes.

```
# simulate two rw
yt <- ts(cumsum(rnorm(250)))
xt <- ts(cumsum(rnorm(250)))

# regression (1) in levels
reg1 <- tslm(yt ~ xt)

# regression (2) in differences
reg2 <- tslm(diff(yt) ~ diff(xt))
```

What do *you* expect?

Models with Nonstationary Variables

```r
# estimates
coeftest(reg1)
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.2953     0.5720   -14.5  < 2e-16 ***
## xt           -0.3123     0.0744    -4.2  3.8e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# R2
summary(reg1)$r.squared
```

```
## [1] 0.06635
```

Models with Nonstationary Variables

```r
# estimates
coeftest(reg2)
```

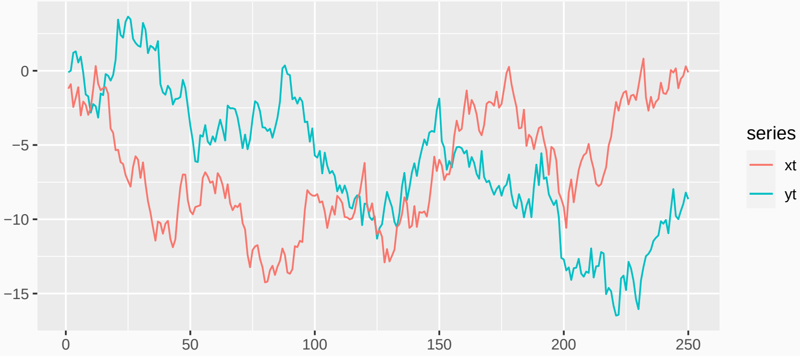
```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.0348     0.0619   -0.56    0.574
## diff(xt)      0.1166     0.0659    1.77    0.078 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# R2
summary(reg2)$r.squared
```

```
## [1] 0.01253
```

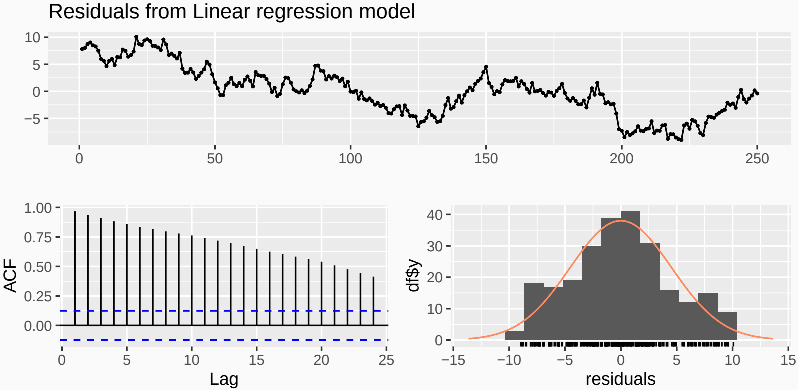
Models with Nonstationary Variables

```
# plot time series
autoplot(yt, series="yt") +
  autolayer(xt, series="xt") + xlab("") + ylab("")
```

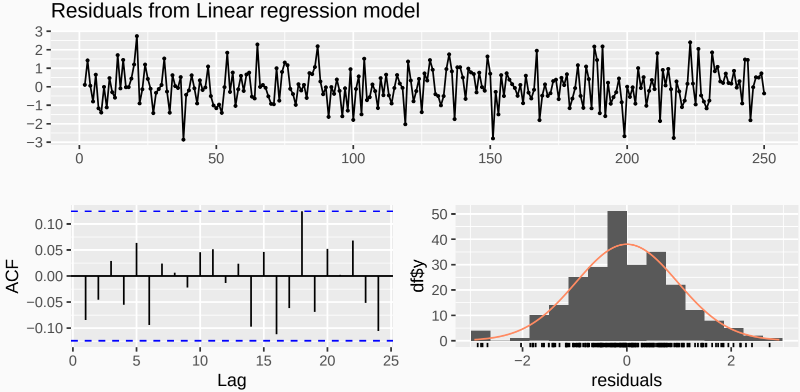


Models with Nonstationary Variables

```r
# check residuals
checkresiduals(reg1, test = FALSE)
```



```r
# check residuals
checkresiduals(reg2, test = FALSE)
```



Granger and Newbold (1974, JoE)

Consider two independent and unrelated $I(1)$ processes with drift.

```
# simulate two rw
yt <- ts(cumsum(.5 + rnorm(250)))
xt <- ts(cumsum(.3 + rnorm(250)))

# regression (3) in levels
reg3 <- tslm(yt ~ xt)

# regression (4) in differences
reg4 <- tslm(diff(yt) ~ diff(xt))
```

Granger and Newbold (1974, JoE)

Consider two independent and unrelated $I(1)$ processes with drift.

```
# simulate two rw
yt <- ts(cumsum(.5 + rnorm(250)))
xt <- ts(cumsum(.3 + rnorm(250)))

# regression (3) in levels
reg3 <- tslm(yt ~ xt)

# regression (4) in differences
reg4 <- tslm(diff(yt) ~ diff(xt))
```

What do *you* expect?

Models with Nonstationary Variables

```r
# estimates
coeftest(reg3)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.9848     1.3627   -8.06  3.2e-14 ***
## xt            3.4546     0.0512   67.53  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# R2
summary(reg3)$r.squared
```

```
## [1] 0.9484
```

Models with Nonstationary Variables

```r
# estimates
coeftest(reg4)
```

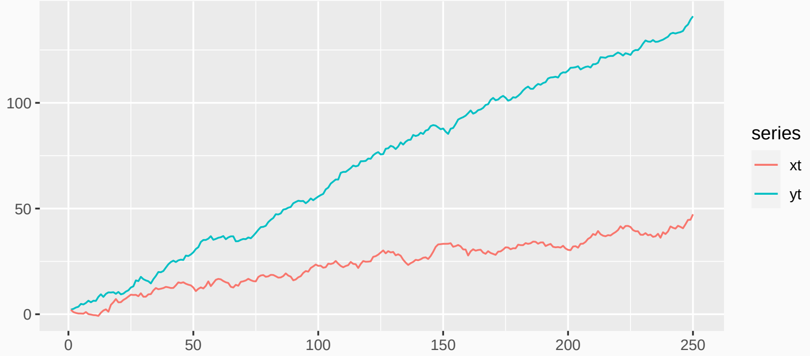
```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5651     0.0599    9.44   <2e-16 ***
## diff(xt)     -0.0429     0.0570   -0.75     0.45
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# R2
summary(reg4)$r.squared
```

```
## [1] 0.00229
```

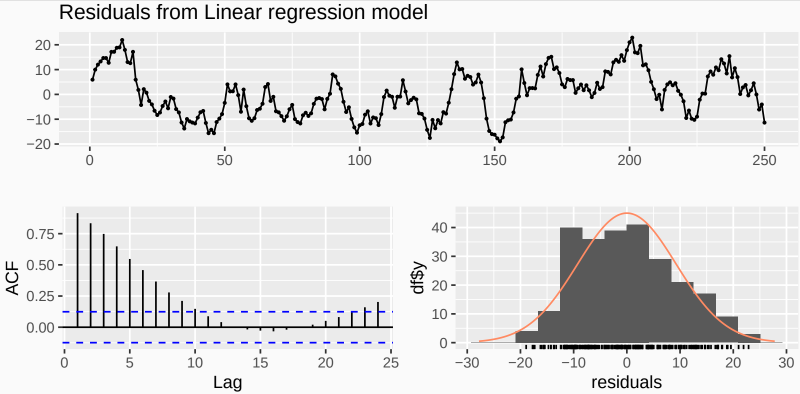
Models with Nonstationary Variables

```
# plot time series
autoplot(yt, series="yt") +
  autolayer(xt, series="xt") + xlab("") + ylab("")
```



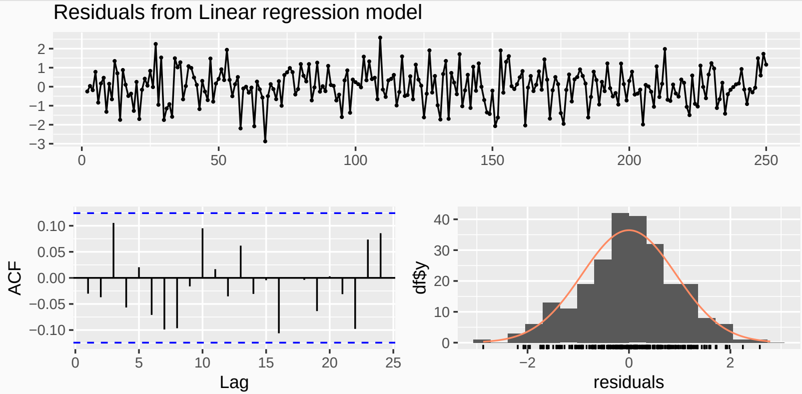
Models with Nonstationary Variables

```r
# check residuals
checkresiduals(reg3, test = FALSE)
```



Models with Nonstationary Variables

```r
# check residuals
checkresiduals(reg4, test = FALSE)
```



Statistical implications of spurious regression

Let y_t and x_t be **two $I(1)$ time series that are not related**. Consider regressing of y_t on x_t giving

$$y_t = \hat{\beta}_0 + \hat{\beta}_1 x_t + \hat{u}_t.$$

Since y_t is not related with x_t:

- true value of β_1 is zero
- the above is a *spurious regression* and $\hat{u}_t \sim I(1)$

As a result, if some or all of the variables in a regression are $I(1)$ then the usual statistical results may or may not hold.

Statistical implications of spurious regression

The following results about the behavior of $\hat{\beta}_1$ in the spurious regression are due to Phillips (1986):

- $\hat{\beta}_1$ does not converge in probability to zero but instead converges in distribution to a non-normal random variable not necessarily centered at zero
- the t-statistics for testing $\beta_1 = 0$ diverge to $\pm\infty$ as $T \to \infty$
- the R^2 from the regression converges to 1 as $T \to \infty$ so that the model will appear to fit well even though it is misspecified
- regression with $I(1)$ data only makes sense when the data are cointegrated

Cointegration

Definition: y_t and x_t, both $I(1)$ time series, are *cointegrated* if there exists a β_1 such that

$$y_t - \beta_1 x_t \sim I(0).$$

In words, the nonstationary time series y_t and x_t are cointegrated if there is a linear combination of them that is stationary or $I(0)$.

- the linear combination is often motivated by economic theory and referred to as a *long-run equilibrium* relationship: $y_t = \beta_0 + \beta_1 x_t$
- intuition: $I(1)$ time series with a long-run equilibrium relationship cannot drift too far apart from the equilibrium because economic forces will act to restore the equilibrium relationship

Cointegration

In macroeconomics…

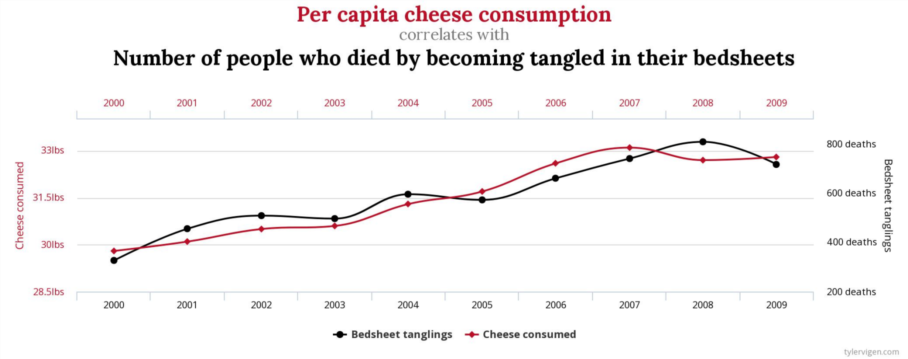
- the equilibrium relationships implied by these economic theories are referred to as *long-run equilibrium* relationships, because the economic forces that act in response to deviations from equilibriium may take a long time to restore equilibrium
- cointegration is modeled using long spans of low frequency time series data measured monthly, quarterly or annually

Cointegration

In finance…

- cointegration may be a high frequency relationship or a low frequency relationship
- cointegration at a high frequency is motivated by arbitrage arguments
- here the terminology long-run equilibrium relationship is somewhat misleading because the economic forces acting to eliminate arbitrage opportunities work very quickly
- cointegration may be appropriately modeled using short spans of high frequency data in seconds, minutes, hours or days

Your turn



- correlation: 0.9471!
- Source: http://tylervigen.com/spurious-correlations