Outline

1. Regression with ARIMA errors
2. Lab session 4
3. Dynamic harmonic regression
4. Lagged predictors
Regression with ARIMA errors

Regression models

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

- \( y_t \) modeled as function of \( k \) explanatory variables \( x_{1,t}, \ldots, x_{k,t} \).
- In regression, we assume that \( \varepsilon_t \) was WN.
- Now we want to allow \( \varepsilon_t \) to be autocorrelated, and potentially non-stationary.
### Regression with ARIMA errors

#### Regression models

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

- \( y_t \) modeled as function of \( k \) explanatory variables \( x_{1,t}, \ldots, x_{k,t}. \)
- In regression, we assume that \( \varepsilon_t \) was WN.
- Now we want to allow \( \varepsilon_t \) to be autocorrelated, and potentially non-stationary.

#### Example: ARIMA(1,1,1) errors

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

\[ (1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t, \]

where \( \varepsilon_t \) is white noise.
Example: $\eta_t = \text{ARIMA}(1,1,1)$

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

\[
(1 - \phi_1 B)(1 - B)\eta_t = (1 + \theta_1 B)\varepsilon_t,
\]
Residuals and errors

Example: $\eta_t = \text{ARIMA}(1,1,1)$

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

- Be careful in distinguishing $\eta_t$ from $\varepsilon_t$.
- Only the errors $\eta_t$ are assumed to be white noise.
- In ordinary regression, $\eta_t$ is assumed to be white noise and so $\eta_t = \varepsilon_t$. 
Any regression with an ARIMA error can be rewritten as a regression with an ARMA error by differencing all variables with the same differencing operator as in the ARIMA model.
Regression with ARIMA errors

Any regression with an ARIMA error can be rewritten as a regression with an ARMA error by differencing all variables with the same differencing operator as in the ARIMA model.

Original data

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

where \( \phi(B)(1 - B)^d \eta_t = \theta(B)\varepsilon_t \)
Regression with ARIMA errors

Any regression with an ARIMA error can be rewritten as a regression with an ARMA error by differencing all variables with the same differencing operator as in the ARIMA model.

**Original data**

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

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

**After differencing all variables**

\[ y'_t = \beta_1 x'_{1,t} + \cdots + \beta_k x'_{k,t} + \eta'_t. \]

where  \( \phi(B)\eta'_t = \theta(B)\varepsilon_t \)

and  \( y'_t = (1 - B)^d y_t \)
Variable selection

- Fit regression model with automatically selected ARIMA errors.
- Check that $\varepsilon_t$ series looks like white noise.

Selecting predictors

- AICc can be calculated for final model.
- Repeat procedure for all subsets of predictors to be considered, and select model with lowest AICc value.
US personal consumption and income

```r
autoplot(uschange[,1:2], facets=TRUE) +
  xlab("Year") + ylab("") +
  ggtitle("Quarterly changes in US consumption and personal income")
```
US personal consumption and income

\texttt{qplot(Income, Consumption, data=as.data.frame(uschange)) + ggtitle("Quarterly changes in US consumption and personal income")}
No need for transformations or further differencing.

Increase in income does not necessarily translate into instant increase in consumption (e.g., after the loss of a job, it may take a few months for expenses to be reduced to allow for the new circumstances). We will ignore this for now.
US personal consumption and income

(fit <- \texttt{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 estimated as 0.322: log likelihood=-156.9
## AIC=325.9  AICc=326.4  BIC=345.3
US personal consumption and income

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

```r
## 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 estimated as 0.322:  log likelihood=-156.9
## AIC=325.9  AICc=326.4  BIC=345.3
```

Write down the equations for the fitted model.
US personal consumption and income

\textbf{checkresiduals(\texttt{fit, test=FALSE})}

- Residuals from Regression with ARIMA(1,0,2) errors
- ACF
- Lag
- Count
- Residuals
US personal consumption and income

\[
fcast <- \text{forecast}(fit, \\
\quad \text{xreg=rep(mean(uschange[,2]),8), h=8})\\
\text{autoplot}(fcast) + \text{xlab}("Year") + \text{ylab}("Percentage change") + \text{ggtitle}("Forecasts from regression with ARIMA(1,0,2) errors")
\]
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, dummies).

Separate forecasting models may be needed for other predictors.

Forecast intervals ignore the uncertainty in forecasting the predictors.
Model daily electricity demand as a function of temperature using quadratic regression with ARMA errors.

\[
\text{qplot}(\text{elecdaily[,"Temperature"], elecdaily[,"Demand"]}) + xlab("Temperature") + ylab("Demand")
\]
Daily electricity demand

```r
autoplot(elecdaily, facets = TRUE)
```
Daily electricity demand

```r
xreg <- cbind(MaxTemp = elecdaily[, "Temperature"],
               MaxTempSq = elecdaily[, "Temperature"]^2,
               Workday = elecdaily[, "WorkDay"],
               Workday2 = elecdaily[, "WorkDay"]^2)
fit <- auto.arima(elecdaily[, "Demand"], xreg = xreg)
checkresiduals(fit)
```

Residuals from Regression with ARIMA(2,1,2)(2,0,0)[7] errors

Ljung-Box test

Data: Residuals from Regression with ARIMA(2,1,2)(2,0,0)[7] errors

Q* = 28, df = 4, p-value = 1e-05

Model df: 10. Total lags used: 14
# Forecast one day ahead

```r
forecast(fit, xreg = cbind(26, 26^2, 1))
```

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
## 53.14 189.8 181.3 198.2 176.8 202.7
Daily electricity demand

\[
\text{fcast} <- \text{forecast}(\text{fit}, \\
\text{xreg} = \text{cbind}(\text{rep}(26, 14), \text{rep}(26^2, 14), \\
\text{c}(0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1)) \\
\text{autoplot}(\text{fcast}) + \text{ylab}("Electicity demand (GW)")
\]

Forecasts from Regression with ARIMA(2,1,2)(2,0,0)[7] errors
Holidays

For daily data

- Use a dummy variable for public holidays. Or several dummy variables for different types of holidays

For monthly data

- Christmas: always in December so part of monthly seasonal effect
- Easter: use a dummy variable $v_t = 1$ if any part of Easter is in that month, $v_t = 0$ otherwise.
- Ramadan and Chinese new year similar.
Trading days

With monthly data, if the observations vary depending on how many different types of days in the month, then trading day predictors can be useful.

\[ z_1 = \# \text{ Mondays in month}; \]
\[ z_2 = \# \text{ Tuesdays in month}; \]
\[ \vdots \]
\[ z_7 = \# \text{ Sundays in month}. \]
Outline

1. Regression with ARIMA errors
2. Lab session 4
3. Dynamic harmonic regression
4. Lagged predictors
Lab Session 4
1. Regression with ARIMA errors
2. Lab session 4
3. Dynamic harmonic regression
4. Lagged predictors
Fourier series

Periodic seasonality can be handled using pairs of Fourier terms:

\[ s_k(t) = \sin \left( \frac{2\pi kt}{m} \right) \quad c_k(t) = \cos \left( \frac{2\pi kt}{m} \right) \]

\[ y_t = a + bt + \sum_{k=1}^{K} \left[ \alpha_k s_k(t) + \beta_k c_k(t) \right] + \varepsilon_t \]

- Every periodic function can be approximated by sums of sin and cos terms for large enough \( K \).
- Choose \( K \) by minimizing AICc.
- Called “harmonic regression”
- `fourier()` function generates these.
Dynamic harmonic regression

Combine Fourier terms with ARIMA errors

**Advantages**
- it allows any length seasonality;
- for data with more than one seasonal period, you can include Fourier terms of different frequencies;
- the seasonal pattern is smooth for small values of $K$ (but more wiggly seasonality can be handled by increasing $K$);
- the short-term dynamics are easily handled with a simple ARMA error.

**Disadvantages**
- seasonality is assumed to be fixed
Eating-out expenditure

cafe04 <- \texttt{window}(auscafe, \texttt{start=2004})
\texttt{autoplot}(cafe04)
Eating-out expenditure

Regression with ARIMA(3, 1, 4) errors and $\lambda = 0$

$K = 1$, AICC = −560.97
Eating-out expenditure

Regression with ARIMA(3, 1, 2) errors and $\lambda = 0$

$K = 2$  $AICC = -617.16$
Eating-out expenditure

Regression with ARIMA(2, 1, 0) errors and $\lambda = 0$

K = 3  AICC = −693.31
Eating-out expenditure

Regression with ARIMA(5, 1, 0) errors and $\lambda = 0$

K= 4    AICC= −790.74
Eating-out expenditure

Regression with ARIMA(0, 1, 1) errors and $\lambda = 0$

K = 5    AICC = −827.77
Eating-out expenditure

Regression with ARIMA(0, 1, 1) errors and $\lambda = 0$

K = 6   AICC = -827.3

series
Data
Regression fit
fit <- `auto.arima`(cafe04, xreg=`fourier`(cafe04, K=5), seasonal = FALSE, lambda = 0)
fc <- `forecast`(fit, xreg=`fourier`(cafe04, K=5, h=24))
autoplot(fc)
Example: weekly gasoline products

harmonics <- fourier(gasoline, K = 13)
(fit <- auto.arima(gasoline, xreg = harmonics, seasonal = FALSE))

## Series: gasoline
## Regression with ARIMA(0,1,2) errors
##
## Coefficients:
##            ma1        ma2       drift       S1-52       C1-52       S2-52
## Estimate  -0.961     0.094     0.001      0.031      -0.255     -0.052
## Std. Error 0.027     0.029     0.001      0.012      0.012      0.009
##            S3-52       C3-52       S4-52       C4-52       S5-52
## Estimate  -0.017     0.024    -0.099      0.032     -0.026     -0.001
## Std. Error 0.009     0.008     0.008      0.008      0.008      0.008
##            S6-52       C6-52       S7-52       C7-52       S8-52
## Estimate  -0.047     0.058    -0.032      0.028     -0.026     -0.001
## Std. Error 0.008     0.008     0.008      0.008      0.008      0.008
##            S9-52       C9-52       S10-52      C10-52      S11-52
## Estimate   0.014    -0.017     0.012     -0.024      0.023     0.000
## Std. Error 0.008     0.008     0.008      0.008      0.008      0.008
##            S12-52      C12-52      S13-52      C13-52
## Estimate  -0.019    -0.029    -0.018      0.001     -0.018
## Std. Error 0.008     0.008     0.008      0.008      0.008
##
## sigma^2 estimated as 0.056:  log likelihood=43.66
## AIC=-27.33    AICc=-25.92    BIC=129
Example: weekly gasoline products

checkresiduals(fit, test=FALSE)

Residuals from Regression with ARIMA(0,1,2) errors

Lag
ACF
0
50
100
0.00 0.05
residuals
count


-0.5
0.0
0.5
1.0

-0.05
0.00
0.05
52.18 104.36 156.54

0.00 0.05

0.00 52.18 104.36 156.54

-0.05
0.00
0.05
1.0

-0.5 0.0 0.5 1.0

residuals

count

0 - 50

0 - 100
Example: weekly gasoline products

```
checkresiduals(fit, plot=FALSE)
```

```r
##
## Ljung-Box test
##
## data: Residuals from Regression with ARIMA(0,1,2) errors
## Q* = 130, df = 75, p-value = 6e-05
##
## Model df: 29. Total lags used: 104.357142857143
```
Example: weekly gasoline products

newharmonics <- \texttt{fourier}(\texttt{gasoline}, K = 13, h = 156)
fc <- \texttt{forecast}(\texttt{fit}, xreg = newharmonics)
\texttt{autoplot}(fc)

Forecasts from Regression with ARIMA(0,1,2) errors

Time
gasoline
level
80
95
5-minute call centre volume

`autoplot(calls)`
5-minute call centre volume

xreg <- `fourier`(calls, K = c(10,0))
(fit <- `auto.arima`(calls, xreg=xreg, seasonal=FALSE, stationary=TRUE))

## Series: calls
## Regression with ARIMA(3,0,2) errors
##
## Coefficients:
## ar1  ar2  ar3  ma1  ma2 intercept
## 0.841 0.192 -0.044 -0.590 -0.189  192.070
## s.e. 0.169 0.178 0.013 0.169 0.137 1.764
## S1-169 C1-169 S2-169 C2-169 S3-169
## s.e. 0.701 0.701 0.379 0.379 0.273
## C3-169 S4-169 C4-169 S5-169 C5-169 S6-169
## -9.327 -9.532 -2.797 -2.239  2.893  0.173
## s.e. 0.273 0.223 0.223 0.196 0.196 0.179
## C6-169 S7-169 C7-169 S8-169 C8-169 S9-169
## 3.305  0.855  0.294  0.857 -1.391 -0.986
## s.e. 0.179 0.168 0.168 0.160 0.160 0.155
## C9-169 S10-169 C10-169
## -0.345 -1.196  0.801
## s.e. 0.155 0.150 0.150

## sigma^2 estimated as 243:  log likelihood=-115412
## AIC=230877  AICc=230877  BIC=231099
checkresiduals(fit, test=FALSE)

Residuals from Regression with ARIMA(3,0,2) errors

ACF

Lag

count

residuals
fc <- forecast(fit, xreg = fourier(calls, c(10, 0), 1690))
autoplot(fc)
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

- $y_t = \text{sales, } x_t = \text{advertising.}$
- $y_t = \text{stream flow, } x_t = \text{rainfall.}$
- $y_t = \text{size of herd, } x_t = \text{breeding stock.}$
Lagged predictors

Sometimes a change in $x_t$ does not affect $y_t$ instantaneously

- $y_t = \text{sales}, x_t = \text{advertising}$.  
- $y_t = \text{stream flow}, x_t = \text{rainfall}$.  
- $y_t = \text{size of herd}, x_t = \text{breeding stock}$.  

- These are dynamic systems with input ($x_t$) and output ($y_t$).  
- $x_t$ is often a leading indicator.  
- There can be multiple predictors.
Distributed lags

Lagged values of a predictor.

Example: $x$ is advertising which has a delayed effect

$$x_1 = \text{advertising for previous month;}$$
$$x_2 = \text{advertising for two months previously;}$$
$$\vdots$$
$$x_m = \text{advertising for } m \text{ months previously.}$$
Example: Insurance quotes and TV adverts

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

Insurance advertising and quotations

Quotes

TV .advert

Year

2002 2003 2004 2005

8
10
12
14
16
18
6
7
8
9
10
11

Year
Example: Insurance quotes and TV adverts

```
Advert <- cbind(
  AdLag0 = insurance[, "TV.advert"],
  AdLag1 = lag(insurance[, "TV.advert"], -1),
  AdLag2 = lag(insurance[, "TV.advert"], -2),
  AdLag3 = lag(insurance[, "TV.advert"], -3)) %>%
  head(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)
c(fit1$aicc, fit2$aicc, fit3$aicc, fit4$aicc)
```

```
## [1] 68.50 60.02 62.83 68.02
```
Example: Insurance quotes and TV adverts

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

```r
## 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 estimated as 0.217:  log likelihood=-23.89
## AIC=61.78  AICc=65.28  BIC=73.6
```
Example: Insurance quotes and TV adverts

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

```r
## 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 estimated as 0.217: log likelihood=-23.89
## AIC=61.78   AICc=65.28   BIC=73.6
```

\[
y_t = 2.04 + 1.26 x_t + 0.16 x_{t-1} + \eta_t,
\]

\[
\eta_t = 1.41 \eta_{t-1} - 0.93 \eta_{t-2} + 0.36 \eta_{t-3} + \varepsilon_t,
\]
Example: Insurance quotes and TV adverts

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

Forecasts from Regression with ARIMA(3,0,0) errors
Example: Insurance quotes and TV adverts

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

Forecasts from Regression with ARIMA(3,0,0) errors
Example: Insurance quotes and TV adverts

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

Forecasts from Regression with ARIMA(3,0,0) errors