Economic Forecasting

Regression models with time series data

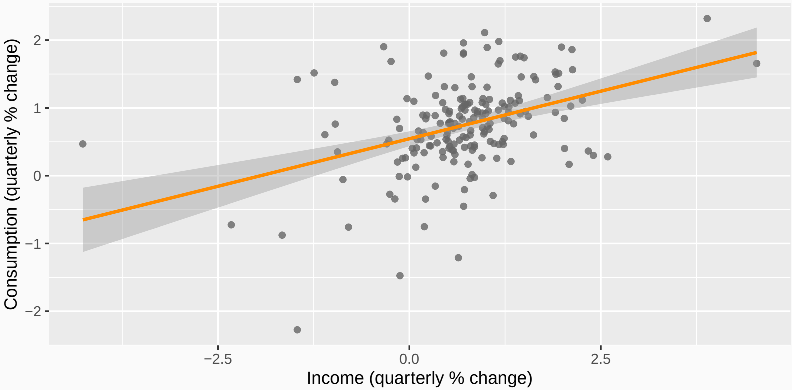
Sebastian Fossati

University of Alberta | E493 | 2023

Outline

Example: US consumption expenditure

Example: US consumption expenditure



Example: US consumption expenditure

```
fit.cons <- tslm(Consumption ~ Income, data = uschange)
coeftest(fit.cons)
```

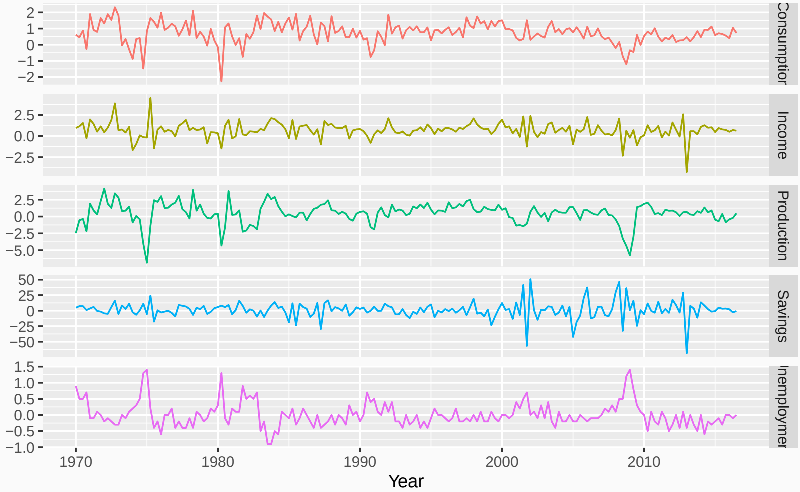
```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.5451     0.0557    9.79  < 2e-16 ***
## Income        0.2806     0.0474    5.91  1.6e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Regression models in R

Related functions:

- `tslm()`: regression model for time series data
- `coeftest()`, `summary()`: prints standard regression output
- `coef()`, `vcov()`, `resid()`, `fitted()`: extract the regression coefficients, (estimated) covariance matrix, residuals, and fitted values respectively
- `confint()`: confidence intervals for the regression coefficient

Example: US consumption expenditure

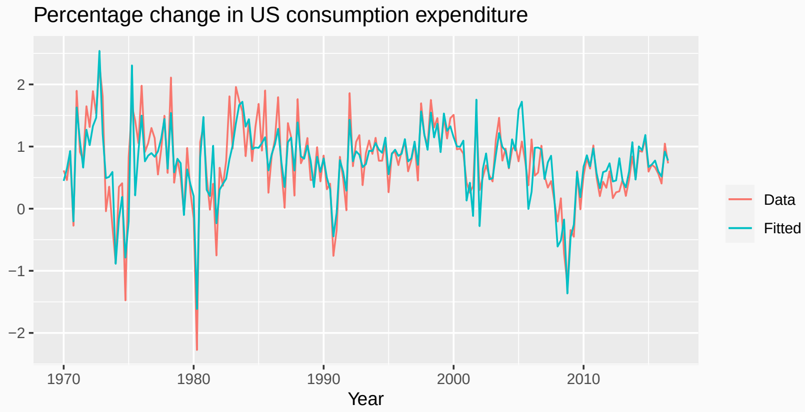


Example: US consumption expenditure

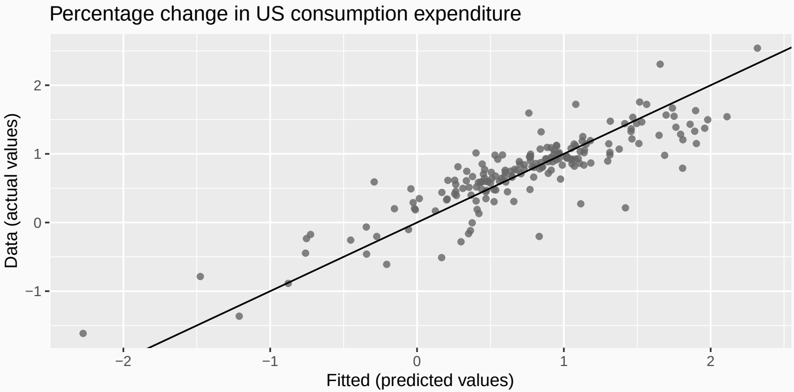
```r
fit.consMR <-
  tslm(
    Consumption ~ Income + Production + Unemployment + Savings,
    data = uschange
  )
coeftest(fit.consMR)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.26729    0.03721    7.18  1.7e-11 ***
## Income        0.71448    0.04219   16.93  < 2e-16 ***
## Production    0.04589    0.02588    1.77    0.078 .
## Unemployment -0.20477    0.10550   -1.94    0.054 .
## Savings      -0.04527    0.00278  -16.29  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Example: US consumption expenditure



Example: US consumption expenditure



Outline

Multiple regression and forecasting

For forecasting purposes, we require the following assumptions:

- ε_t are uncorrelated and zero mean
- ε_t are uncorrelated with each $x_{j,t}$

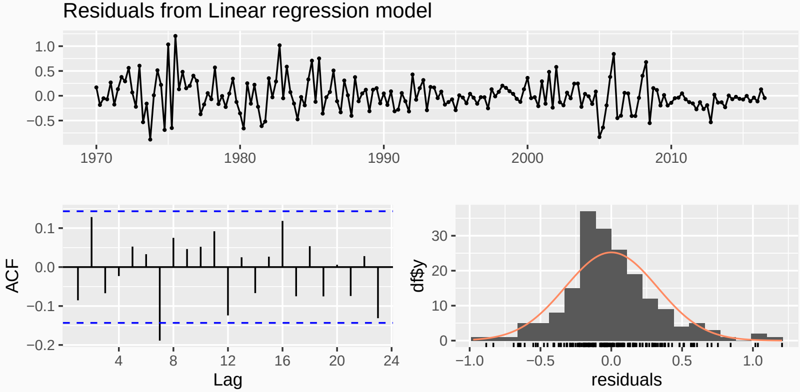
For forecasting purposes, we require the following assumptions:

- ε_t are uncorrelated and zero mean
- ε_t are uncorrelated with each $x_{j,t}$

It is **useful** to also have $\varepsilon_t \sim \mathrm{N}(0, \sigma^2)$ when producing prediction intervals or doing statistical tests.

Example: US consumption expenditure

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



Breusch-Godfrey test

Run the following auxiliary regression:

$$e_t = \beta_0 + \beta_1 x_{t,1} + \cdots + \beta_k x_{t,k} + \rho_1 e_{t-1} + \cdots + \rho_p e_{t-p} + u_t$$

If R^2 statistic is calculated for this model, then

$$(T - p)R^2 \sim \chi^2_p,$$

when there is no serial correlation up to lag p, and T is the length of series.

- the Breusch-Godfrey test is better than Ljung-Box for regression models

US consumption again

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

```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 15, df = 8, p-value = 0.06
```

If the model fails the Breusch-Godfrey test…

- the forecasts are not wrong, but have higher variance than they need to
- there is information in the residuals that we should exploit

Outline

Linear trend model

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t$$

Remarks:

- $x_t = t$ for $t = 1, 2, \ldots, T$
- strong assumption that trend will continue
- specified using the predictor `trend` in the `tslm()` function

Seasonal dummy variables

$$y_t = \beta_1 d_{1,t} + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \varepsilon_t$$

Remarks:

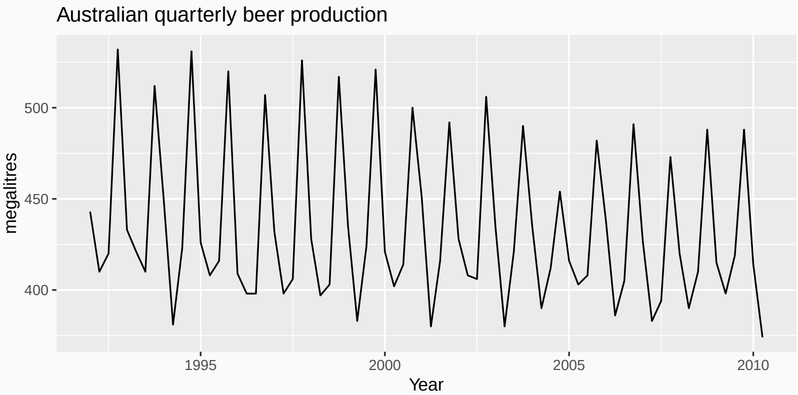
- $x_{i,t} = d_{i,t}$ for $i = 1, \ldots, 4$
- $d_{i,t} = 1$ if t is quarter i and 0 otherwise
- specified using the predictor `season` in the `tslm()` function
- no intercept in this model! why?

Beware of the dummy variable trap!

Remarks:

- using one dummy for each category gives too many dummy variables!
- the regression will then be singular and inestimable
- either omit the constant, or omit the dummy for one category
- the coefficients of the dummies are relative to the omitted category

Beer production revisited



Beer production revisited

We can use a simple trend plus seasonal dummy model to forecast beer production.

Regression model

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \varepsilon_t$$

Remarks:

- $d_{i,t} = 1$ if t is quarter i and 0 otherwise

Beer production revisited

```
fit.beer <- tslm(beer ~ trend + season)
coeftest(fit.beer)
```

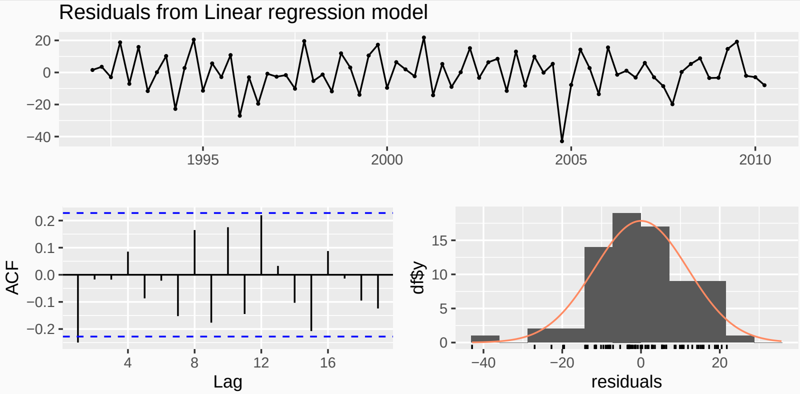
```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 441.8004     3.7335  118.33  < 2e-16 ***
## trend        -0.3403     0.0666   -5.11  2.7e-06 ***
## season2     -34.6597     3.9683   -8.73  9.1e-13 ***
## season3     -17.8216     4.0225   -4.43  3.4e-05 ***
## season4      72.7964     4.0230   18.09  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Beer production revisited

Beer production revisited

Beer production revisited

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



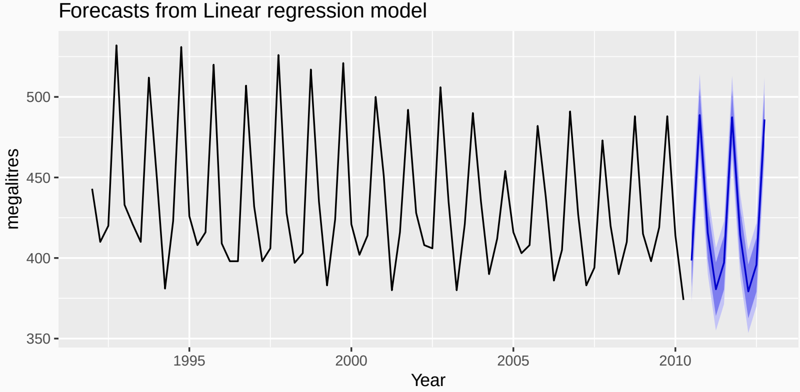
Beer production revisited

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

```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 9.3, df = 8, p-value = 0.3
```

Beer production revisited

```
# plot forecasts
fcast <- forecast(fit.beer)
autoplot(fcast) + xlab("Year") + ylab("megalitres")
```



Intervention variables

Other useful predictors:

- **spikes**: variable equals 1 at the intervention and 0 elsewhere (useful to remove the effect of an outlier)

Intervention variables

Other useful predictors:

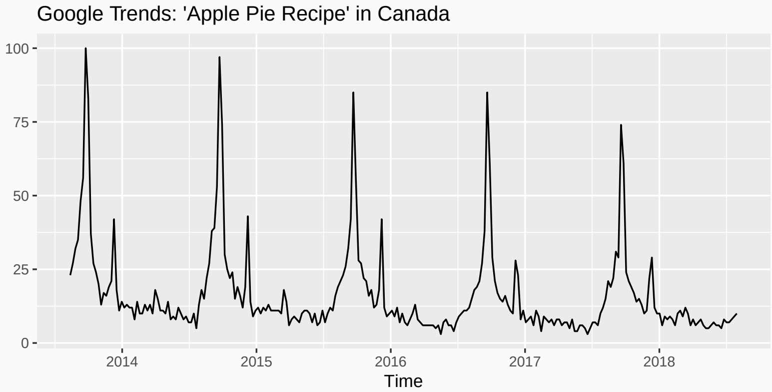
- **spikes**: variable equals 1 at the intervention and 0 elsewhere (useful to remove the effect of an outlier)
- **steps**: variable equals 0 before the intervention and 1 afterwards (useful to model structural breaks)

Intervention variables

Other useful predictors:

- **spikes**: variable equals 1 at the intervention and 0 elsewhere (useful to remove the effect of an outlier)
- **steps**: variable equals 0 before the intervention and 1 afterwards (useful to model structural breaks)
- **change of slope**: variable equals 0 before the intervention and $\{1, 2, 3, \ldots\}$ afterwards (useful to model change in slope)

Holidays



Google Trends: 'Apple Pie Recipe' in Canada
100
75
50
25
0
2014
2015
2016
2017
2018
Time

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.

$$
\begin{aligned}
z_1 &= \text{\# Mondays in month} \\
z_2 &= \text{\# Tuesdays in month} \\
&\vdots \\
z_7 &= \text{\# Sundays in month}
\end{aligned}
$$

Nonlinear trend

Piecewise linear trend with bend at τ:

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$

Quadratic or higher order trend:

$$x_{1,t} = t, \quad x_{2,t} = t^2, \quad \dots$$

Nonlinear trend

Piecewise linear trend with bend at τ:

$$x_{1,t} = t$$

$$x_{2,t} = \begin{cases} 0 & t < \tau \\ (t - \tau) & t \geq \tau \end{cases}$$

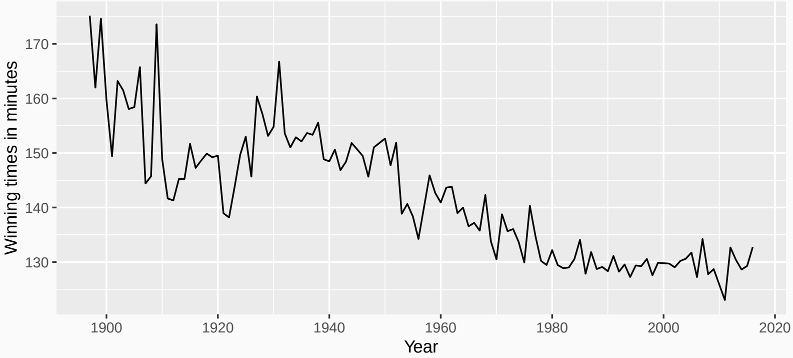
Quadratic or higher order trend:

$$x_{1,t} = t, \quad x_{2,t} = t^2, \quad \dots$$

NOT RECOMMENDED!

Example: Boston marathon winning times

```r
# Boston marathon times
autoplot(marathon) +
  xlab("Year") +  ylab("Winning times in minutes")
```



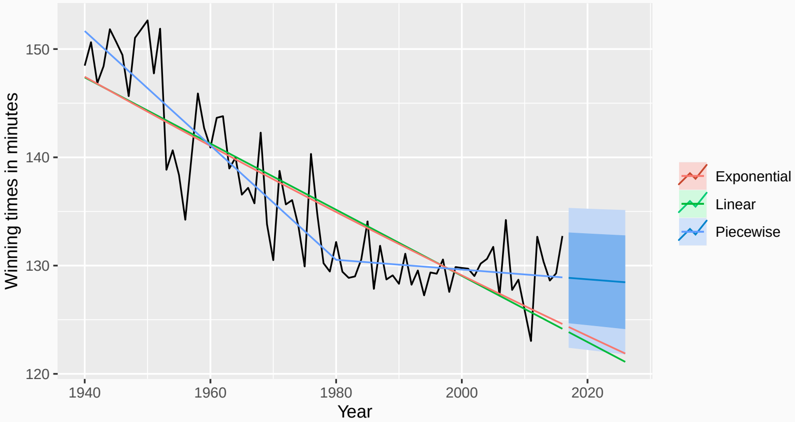
Example: Boston marathon winning times

```r
marathon2 <- window(marathon, start = 1940)
autoplot(marathon2) +
  xlab("Year") +  ylab("Winning times in minutes")
```

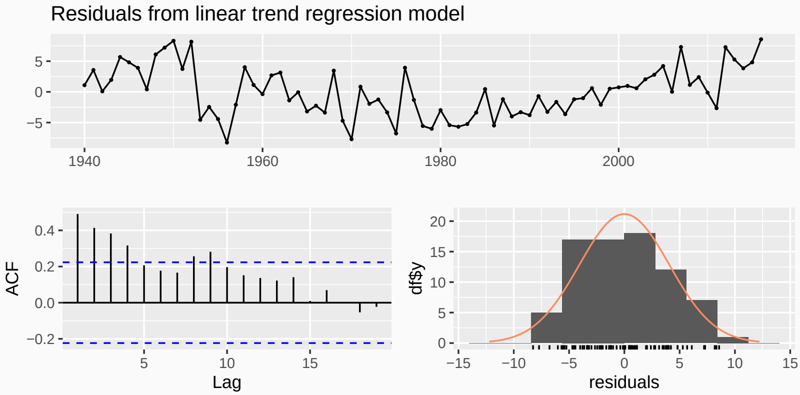
Example: Boston marathon winning times

```r
# linear trend
fit.lin <- tslm(marathon2 ~ trend)
fcasts.lin <- forecast(fit.lin, h = 10)

# exponential trend
fit.exp <- tslm(marathon2 ~ trend, lambda = 0)
fcasts.exp <- forecast(fit.exp, h = 10)

# piecewise linear trend
t.break1 <- 1980
t <- time(marathon2)
t1 <- ts(pmax(0, t-t.break1), start = 1940)
fit.pw <- tslm(marathon2 ~ t + t1)
t.new <- t[length(t)] + seq(10)
t1.new <- t1[length(t1)] + seq(10)
newdata <- data.frame("t" = t.new, "t1" = t1.new)
fcasts.pw <- forecast(fit.pw, newdata = newdata)
```

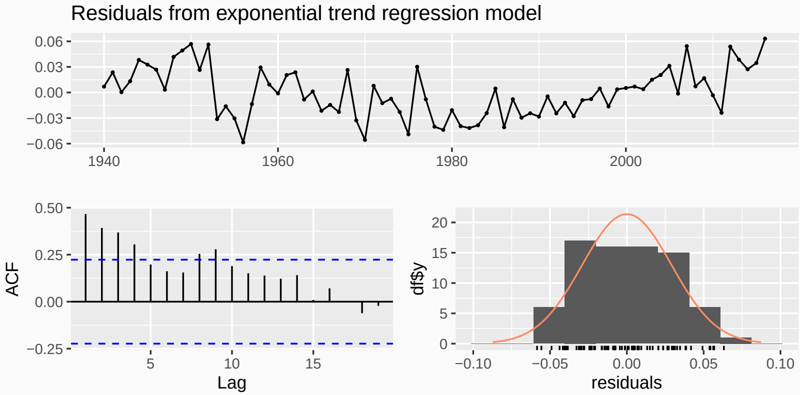
Example: Boston marathon winning times



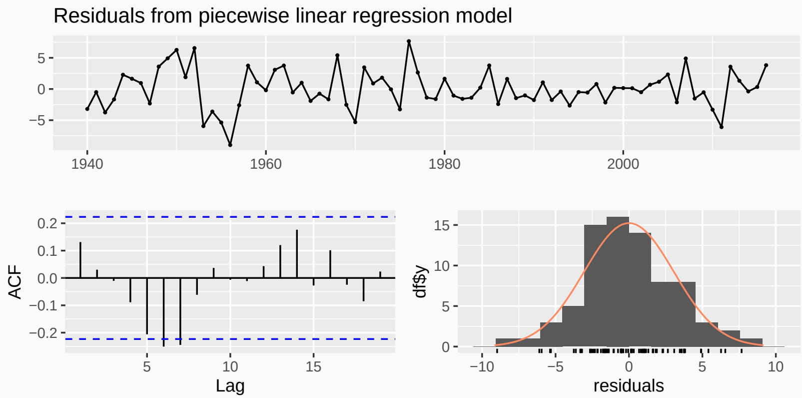
Example: Boston marathon winning times



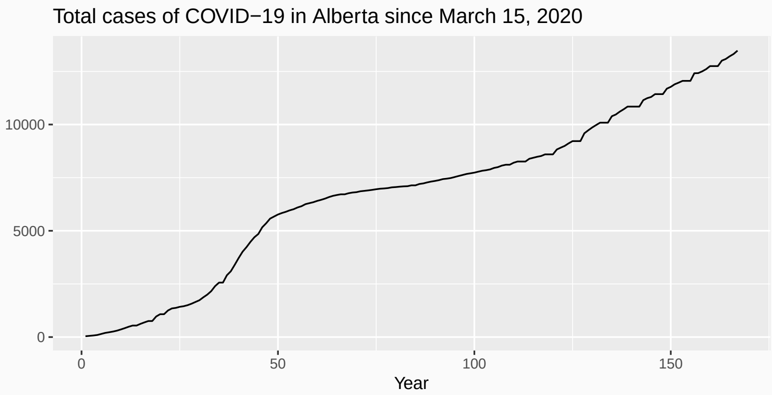
Example: Boston marathon winning times



Example: Boston marathon winning times

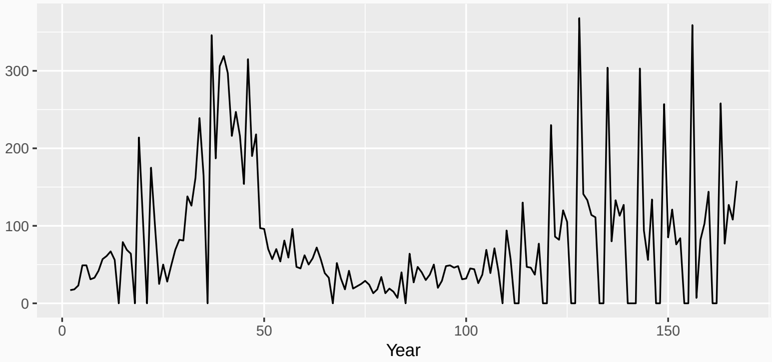


Your turn


Total cases of COVID−19 in Alberta since March 15, 2020
10000
5000
0
0
50
100
150
Year

Your turn

New cases of COVID−19 in Alberta since March 15, 2020



Outline

Set up:

- let y^0 be the **new** value for which we would like a forecast
- and $x_1^0, \cdots, x_k^0$ the values of the predictors of y^0

Predicted value

$$\hat{y}^0 = \hat{\beta}_0 + \hat{\beta}_1 x_1^0 + \cdots + \hat{\beta}_k x_k^0$$

To compute a prediction interval ...

- ignoring parameter estimation uncertainty (that is, sampling error in $\hat{y}^0$)

and assuming forecast errors are normally distributed, then an approximate 95% PI is

Prediction interval

$$\hat{y}^0 \pm 1.96\hat{\sigma}_e$$

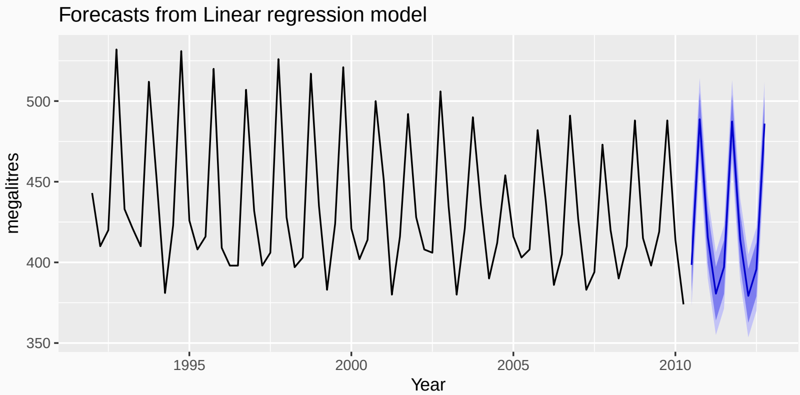
where $\hat{\sigma}_e$ is the standard error of the regression.

Ex-ante versus ex-post forecasts

Remarks:

- *ex ante forecasts* are made using only information available in advance
    - require forecasts of predictors
- *ex post forecasts* are made using later information on the predictors
    - useful for studying behavior of forecasting models
- trend, seasonal and calendar variables are all known in advance, so these don't need to be forecast

Beer production



Scenario based forecasting

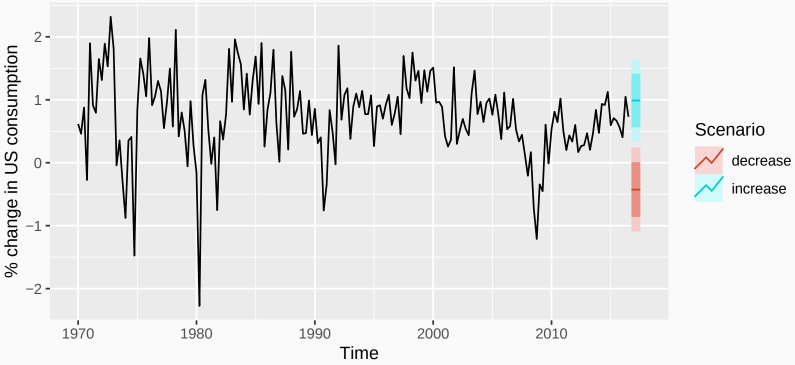
Scenario based forecasting:

- assumes possible scenarios for the predictor variables
- note: prediction intervals for scenario based forecasts do not include the uncertainty associated with the future values of the predictor variables

US Consumption

```r
fit.consBest <- tslm(Consumption ~ Income + Savings + Unemployment, data = uschange)
h <- 4
# increase
newdata <- data.frame(
    Income = c(1, 1, 1, 1),
    Savings = c(0.5, 0.5, 0.5, 0.5),
    Unemployment = c(0, 0, 0, 0)
)
fcast.up <- forecast(fit.consBest, newdata = newdata)
# decrease
newdata <- data.frame(
    Income = rep(-1, h),
    Savings = rep(-0.5, h),
    Unemployment = rep(0, h)
)
fcast.down <- forecast(fit.consBest, newdata = newdata)
```

US Consumption



Building a predictive regression model

Remarks:

- if getting forecasts of predictors is difficult, you can use lagged predictors instead

$$y_{t+h} = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + \varepsilon_{t+h}$$

- implies a different model for each forecast horizon h