

# ETC3550/ETC5550

## Applied forecasting

Ch7. Regression models

[OTexts.org/fpp3/](https://OTexts.org/fpp3/)

# Outline

- 1 The linear model with time series
- 2 Some useful predictors for linear models
- 3 Selecting predictors and forecast evaluation
- 4 Forecasting with regression
- 5 Matrix formulation
- 6 Correlation, causation and forecasting

# Outline

- 1 The linear model with time series
- 2 Some useful predictors for linear models
- 3 Selecting predictors and forecast evaluation
- 4 Forecasting with regression
- 5 Matrix formulation
- 6 Correlation, causation and forecasting

# Multiple regression and forecasting

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t.$$

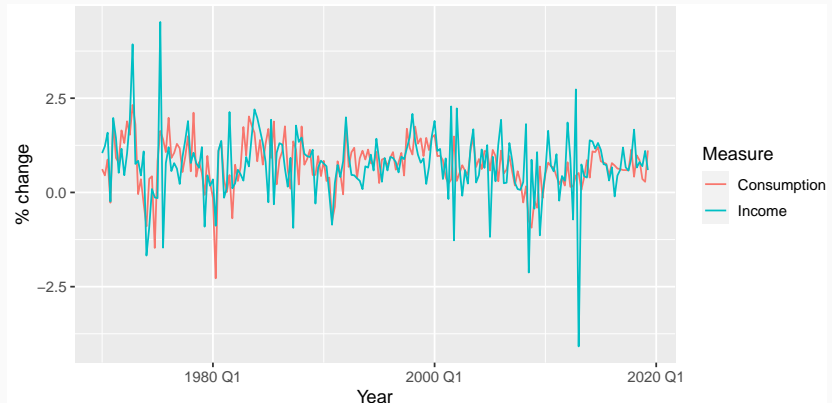
- $y_t$  is the variable we want to predict: the “response” variable
- Each  $x_{j,t}$  is numerical and is called a “predictor”. They are usually assumed to be known for all past and future times.
- The coefficients  $\beta_1, \dots, \beta_k$  measure the effect of each predictor after taking account of the effect of all other predictors in the model.

That is, the coefficients measure the **marginal effects**.

- $\varepsilon_t$  is a white noise error term

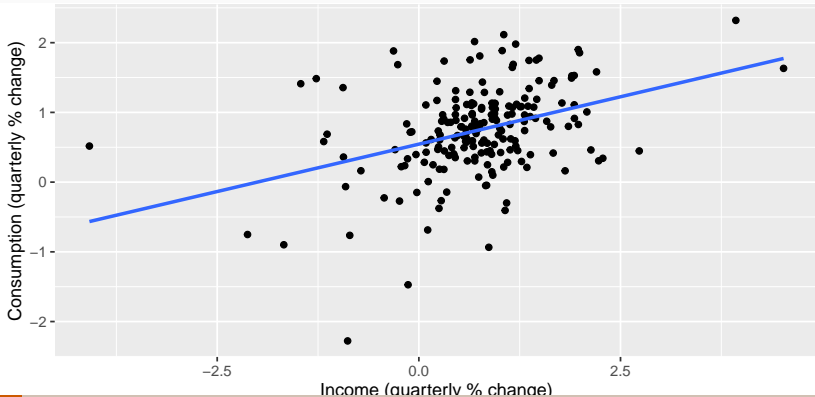
# Example: US consumption expenditure

```
us_change %>%  
  gather("Measure", "Change", Consumption, Income) %>%  
  autoplot(Change) +  
  ylab("% change") + xlab("Year")
```



# Example: US consumption expenditure

```
us_change %>%  
  ggplot(aes(x=Income, y=Consumption)) +  
    ylab("Consumption (quarterly % change)") +  
    xlab("Income (quarterly % change)") +  
    geom_point() + geom_smooth(method="lm", se=FALSE)
```

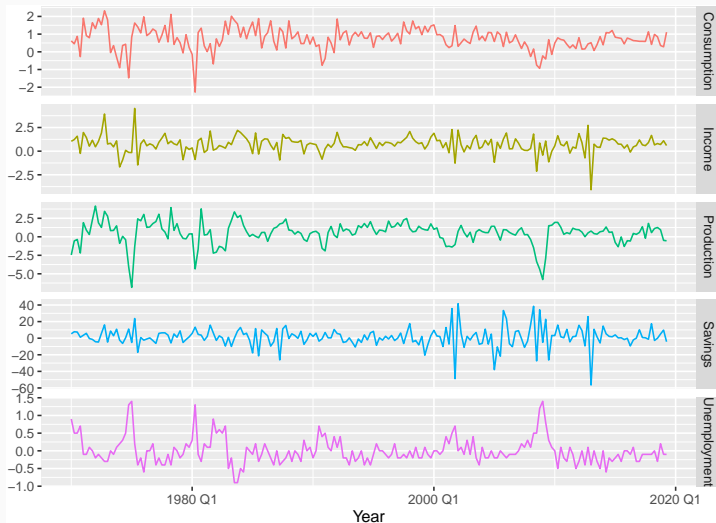


# Example: US consumption expenditure

```
fit_cons <- us_change %>%  
  model(lm = TSLM(Consumption ~ Income))  
report(fit_cons)
```

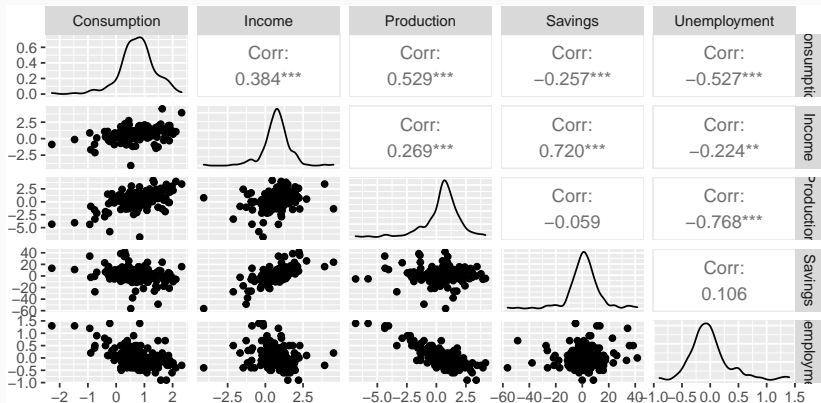
```
## Series: Consumption  
## Model: TSLM  
##  
## Residuals:  
##      Min      1Q  Median      3Q      Max  
## -2.582 -0.278  0.019  0.323  1.422  
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept)   0.5445     0.0540   10.08 < 2e-16 ***  
## Income        0.2718     0.0467    5.82 2.4e-08 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.591 on 196 degrees of freedom  
## Multiple R-squared: 0.147,    Adjusted R-squared: 0.143
```

# Example: US consumption expenditure





# Example: US consumption expenditure

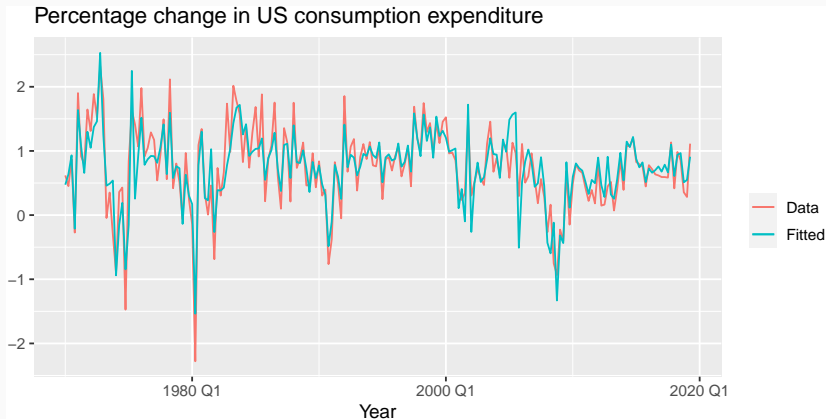


# Example: US consumption expenditure

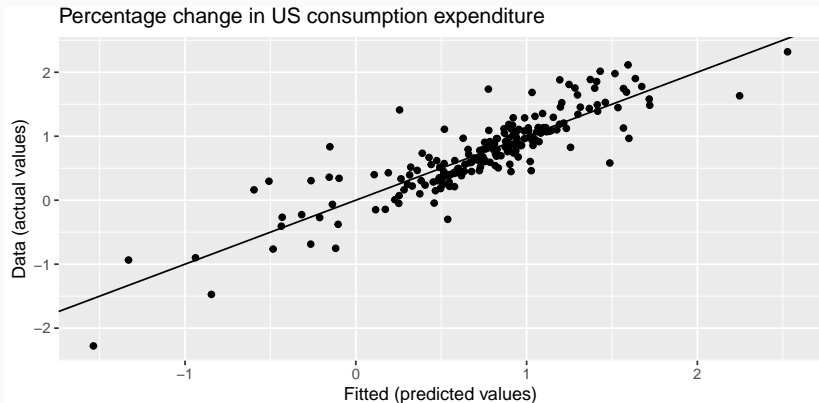
```
fit_consMR <- us_change %>%  
  model(lm = TSLM(Consumption ~ Income + Production + Unemployment + Savings))  
report(fit_consMR)
```

```
## Series: Consumption  
## Model: TSLM  
##  
## Residuals:  
##      Min      1Q  Median      3Q      Max  
## -0.906 -0.158 -0.036  0.136  1.155  
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept)   0.25311   0.03447    7.34 5.7e-12 ***  
## Income        0.74058   0.04012   18.46 < 2e-16 ***  
## Production    0.04717   0.02314    2.04  0.043 *  
## Unemployment -0.17469   0.09551   -1.83  0.069 .  
## Savings       -0.05289   0.00292  -18.09 < 2e-16 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 0.31 on 193 degrees of freedom  
## Multiple R-squared: 0.768, Adjusted R-squared: 0.763  
## F-statistic: 160 on 4 and 193 DF, p-value: <2e-16
```

# Example: US consumption expenditure

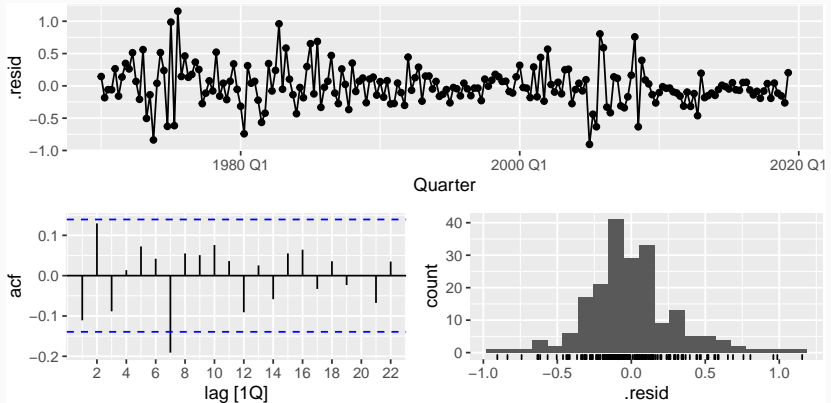


# Example: US consumption expenditure



# Example: US consumption expenditure

```
augment(fit_consMR) %>%  
  gg_tsdisplay(.resid, plot_type="hist")
```



# Outline

- 1 The linear model with time series
- 2 Some useful predictors for linear models
- 3 Selecting predictors and forecast evaluation
- 4 Forecasting with regression
- 5 Matrix formulation
- 6 Correlation, causation and forecasting

## Linear trend

$$x_t = t$$

- $t = 1, 2, \dots, T$
- Strong assumption that trend will continue.

# Dummy variables

If a categorical variable takes only two values (e.g., 'Yes' or 'No'), then an equivalent numerical variable can be constructed taking value 1 if yes and 0 if no. This is called a **dummy variable**.

	A	B
1	Yes	1
2	Yes	1
3	No	0
4	Yes	1
5	No	0
6	No	0
7	Yes	1
8	Yes	1
9	No	0
10	No	0
11	No	0
12	No	0
13	Yes	1
14	No	0



# Dummy variables

If there are more than two categories, then the variable can be coded using several dummy variables (one fewer than the total number of categories).

	A	B	C	D	E
1	Monday	1	0	0	0
2	Tuesday	0	1	0	0
3	Wednesday	0	0	1	0
4	Thursday	0	0	0	1
5	Friday	0	0	0	0
6	Monday	1	0	0	0
7	Tuesday	0	1	0	0
8	Wednesday	0	0	1	0
9	Thursday	0	0	0	1
10	Friday	0	0	0	0
11	Monday	1	0	0	0
12	Tuesday	0	1	0	0
13	Wednesday	0	0	1	0
14	Thursday	0	0	0	1
15	Friday	0	0	0	0

# Beware of the dummy variable trap!

- 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.

# Uses of dummy variables

## Seasonal dummies

- For quarterly data: use 3 dummies
- For monthly data: use 11 dummies
- For daily data: use 6 dummies
- What to do with weekly data?

# Uses of dummy variables

## Seasonal dummies

- For quarterly data: use 3 dummies
- For monthly data: use 11 dummies
- For daily data: use 6 dummies
- What to do with weekly data?

## Outliers

- If there is an outlier, you can use a dummy variable to remove its effect.

# Uses of dummy variables

## Seasonal dummies

- For quarterly data: use 3 dummies
- For monthly data: use 11 dummies
- For daily data: use 6 dummies
- What to do with weekly data?

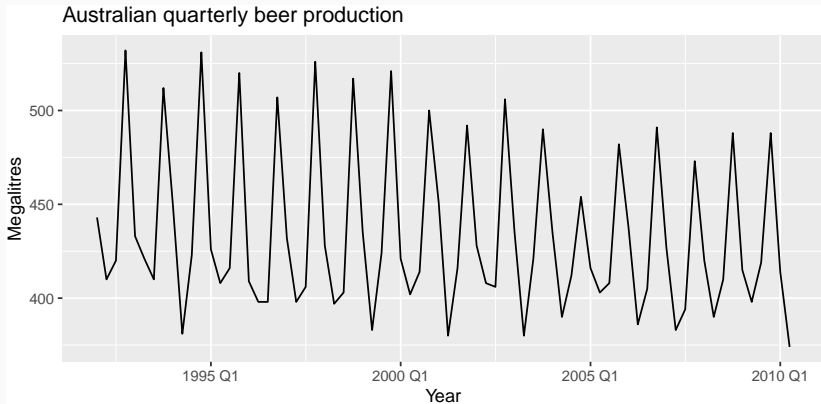
## Outliers

- If there is an outlier, you can use a dummy variable to remove its effect.

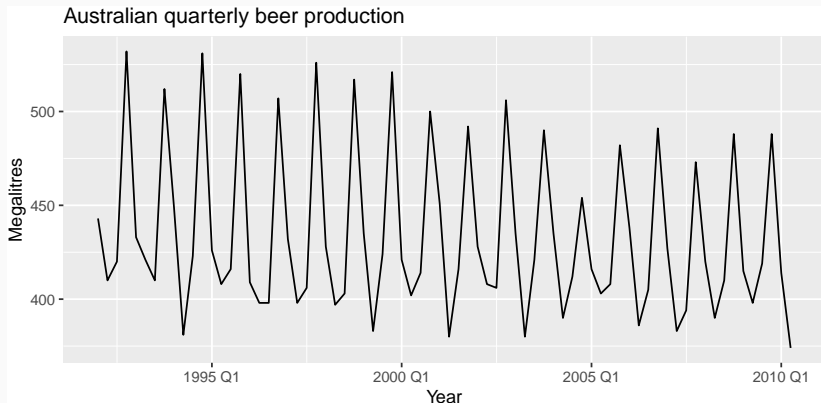
## Public holidays

- For daily data: if it is a public holiday,  $\text{dummy}=1$ , otherwise  $\text{dummy}=0$ .

# Beer production revisited



# Beer production revisited



## Regression model

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

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

# Beer production revisited

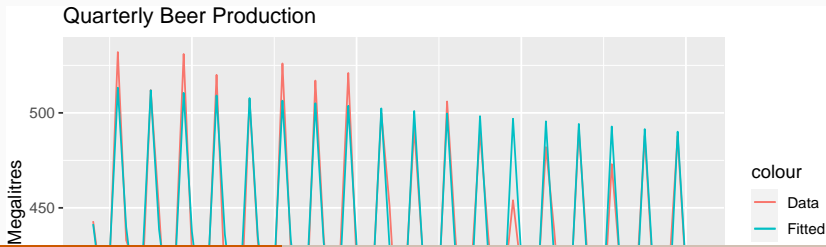
```
fit_beer <- recent_production %>% model(TSLM(Beer ~ trend() + season()))
report(fit_beer)
```

```
## Series: Beer
## Model: TSLM
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -42.9   -7.6    -0.5     8.0    21.8
##
## 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 ***
## season()year2 -34.6597     3.9683   -8.73 9.1e-13 ***
## season()year3 -17.8216     4.0225   -4.43 3.4e-05 ***
## season()year4  72.7964     4.0230   18.09 < 2e-16 ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.2 on 69 degrees of freedom
## Multiple R-squared:  0.924,    Adjusted R-squared:  0.92
## F-statistic: 211 on 4 and 69 DF, p-value: <2e-16
```

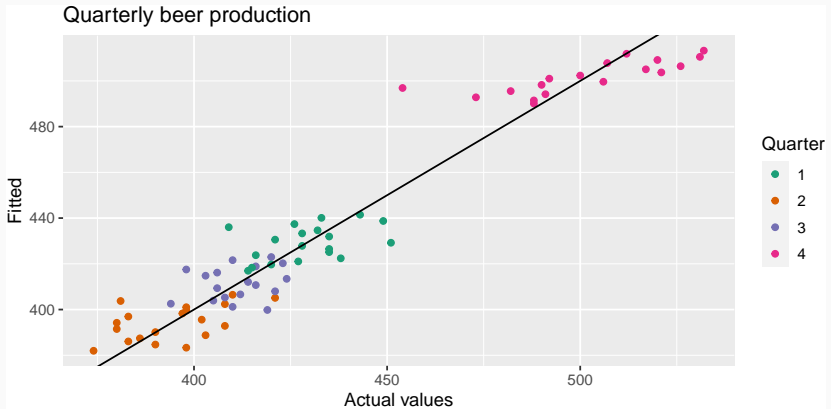


# Beer production revisited

```
augment(fit_beer) %>%  
  ggplot(aes(x = Quarter)) +  
  geom_line(aes(y = Beer, colour = "Data")) +  
  geom_line(aes(y = .fitted, colour = "Fitted")) +  
  xlab("Year") + ylab("Megalitres") +  
  ggtitle("Quarterly Beer Production")
```

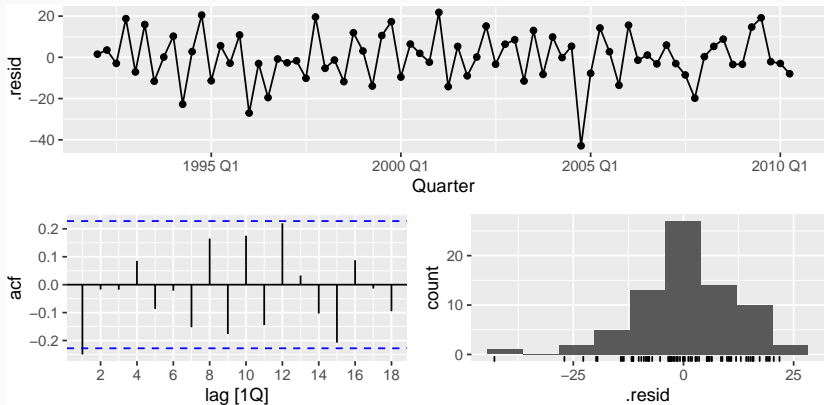


# Beer production revisited



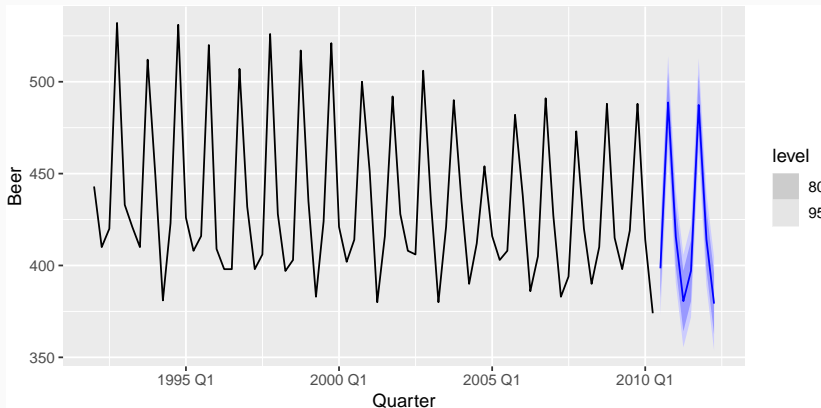
# Beer production revisited

```
augment(fit_beer) %>% gg_tsdisplay(.resid, plot_type="hist")
```



# Beer production revisited

```
fit_beer %>% forecast %>% autoplot(recent_production)
```



# Fourier series

Periodic seasonality can be handled using pairs of Fourier terms:

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

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

- 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”

```
TSLM(y ~ trend() + fourier(K))
```

# Harmonic regression: beer production

```
fourier_beer <- recent_production %>% model(TSLM(Beer ~ trend() + fourier(K=2)))  
report(fourier_beer)
```

```
## Series: Beer  
## Model: TSLM  
##  
## Residuals:  
##      Min      1Q  Median      3Q      Max  
## -42.9   -7.6   -0.5     8.0    21.8  
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept)    446.8792     2.8732  155.53 < 2e-16 ***  
## trend()        -0.3403     0.0666   -5.11 2.7e-06 ***  
## fourier(K = 2)C1_4  8.9108     2.0112    4.43 3.4e-05 ***  
## fourier(K = 2)S1_4 -53.7281     2.0112  -26.71 < 2e-16 ***  
## fourier(K = 2)C2_4 -13.9896     1.4226   -9.83 9.3e-15 ***  
## ---  
## Signif. codes:  
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 12.2 on 69 degrees of freedom  
## Multiple R-squared: 0.924, Adjusted R-squared: 0.92  
## F-statistic: 211 on 4 and 69 DF, p-value: <2e-16
```

# Intervention variables

## Spikes

- Equivalent to a dummy variable for handling an outlier.

# Intervention variables

## Spikes

- Equivalent to a dummy variable for handling an outlier.

## Steps

- Variable takes value 0 before the intervention and 1 afterwards.



# Intervention variables

## Spikes

- Equivalent to a dummy variable for handling an outlier.

## Steps

- Variable takes value 0 before the intervention and 1 afterwards.

## Change of slope

- Variables take values 0 before the intervention and values  $\{1, 2, 3, \dots\}$  afterwards.

## 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.

$z_1 = \# \text{ Mondays in month;}$

$z_2 = \# \text{ Tuesdays in month;}$

$\vdots$

$z_7 = \# \text{ Sundays in month.}$

# Distributed lags

Lagged values of a predictor.

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

$x_1$  = advertising for previous month;

$x_2$  = advertising for two months previously;

$\vdots$

$x_m$  = advertising for  $m$  months previously.

# Nonlinear trend

## Piecewise linear trend with bend at $\tau$

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

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

# Nonlinear trend

## Piecewise linear trend with bend at $\tau$

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

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

## 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 $\tau$

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

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

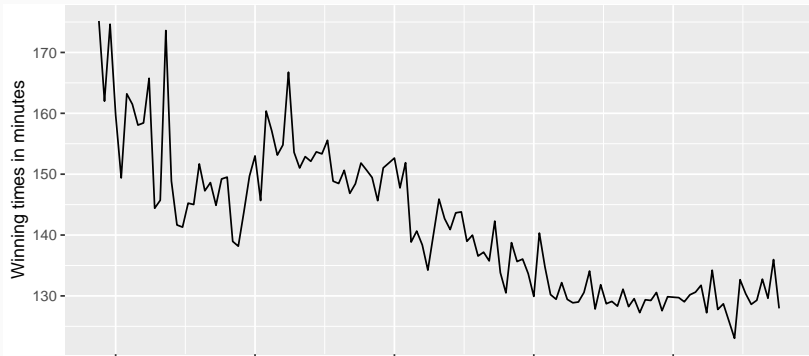
## Quadratic or higher order trend

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

**NOT RECOMMENDED!**

# Example: Boston marathon winning times

```
marathon <- boston_marathon %>%  
  filter(Event == "Men's open division") %>%  
  select(-Event) %>%  
  mutate(Minutes = as.numeric(Time)/60)  
marathon %>% autoplot(Minutes) +  
  xlab("Year") + ylab("Winning times in minutes")
```





# Example: Boston marathon winning times

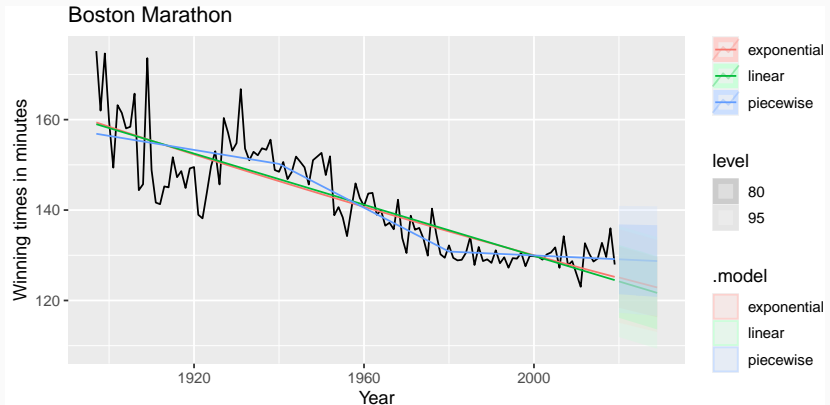
```
fit_trends <- marathon %>%  
  model(  
    # Linear trend  
    linear = TSLM(Minutes ~ trend()),  
    # Exponential trend  
    exponential = TSLM(log(Minutes) ~ trend()),  
    # Piecewise linear trend  
    piecewise = TSLM(Minutes ~ trend(knots = c(1940, 1980)))  
  )
```

```
fit_trends
```

```
## # A mable: 1 x 3  
##   linear exponential piecewise  
##   <model>         <model>     <model>  
## 1  <TSLM>         <TSLM>     <TSLM>
```

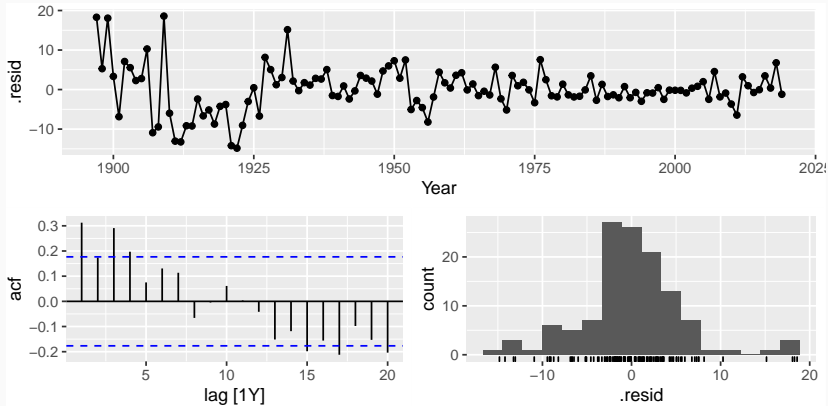
# Example: Boston marathon winning times

```
fit_trends %>% forecast(h=10) %>% autoplot(marathon)
```



# Example: Boston marathon winning times

```
fit_trends %>% select(piecewise) %>%  
  augment() %>% gg_tsdisplay(.resid, plot_type = "residuals")
```



# Outline

- 1 The linear model with time series
- 2 Some useful predictors for linear models
- 3 Selecting predictors and forecast evaluation
- 4 Forecasting with regression
- 5 Matrix formulation
- 6 Correlation, causation and forecasting

# Comparing regression models

Computer output for regression will always give the  $R^2$  value. This is a useful summary of the model.

- It is equal to the square of the correlation between  $y$  and  $\hat{y}$ .
- It is often called the “coefficient of determination”.
- It can also be calculated as follows:

$$R^2 = \frac{\sum(\hat{y}_t - \bar{y})^2}{\sum(y_t - \bar{y})^2}$$

- It is the proportion of variance accounted for (explained) by the predictors.

# Comparing regression models

However ...

- $R^2$  does not allow for “degrees of freedom”.
- Adding *any* variable tends to increase the value of  $R^2$ , even if that variable is irrelevant.

# Comparing regression models

However ...

- $R^2$  does not allow for “degrees of freedom”.
- Adding *any* variable tends to increase the value of  $R^2$ , even if that variable is irrelevant.

To overcome this problem, we can use *adjusted*  $R^2$ :

$$\bar{R}^2 = 1 - (1 - R^2) \frac{T - 1}{T - k - 1}$$

where  $k$  = no. predictors and  $T$  = no. observations.

# Comparing regression models

However ...

- $R^2$  does not allow for “degrees of freedom”.
- Adding *any* variable tends to increase the value of  $R^2$ , even if that variable is irrelevant.

To overcome this problem, we can use *adjusted*  $R^2$ :

$$\bar{R}^2 = 1 - (1 - R^2) \frac{T - 1}{T - k - 1}$$

where  $k$  = no. predictors and  $T$  = no. observations.

**Maximizing  $\bar{R}^2$  is equivalent to minimizing  $\hat{\sigma}^2$ .**

$$\hat{\sigma}^2 = \frac{1}{T - k - 1} \sum_{t=1}^T \varepsilon_t^2$$



# Akaike's Information Criterion

$$AIC = -2 \log(L) + 2(k + 2)$$

where  $L$  is the likelihood and  $k$  is the number of predictors in the model.

# Akaike's Information Criterion

$$\text{AIC} = -2 \log(L) + 2(k + 2)$$

where  $L$  is the likelihood and  $k$  is the number of predictors in the model.

- AIC penalizes terms more heavily than  $\bar{R}^2$ .
- Minimizing the AIC is asymptotically equivalent to minimizing MSE via **leave-one-out cross-validation** (for any linear regression).

## Corrected AIC

For small values of  $T$ , the AIC tends to select too many predictors, and so a bias-corrected version of the AIC has been developed.

$$\text{AIC}_C = \text{AIC} + \frac{2(k+2)(k+3)}{T-k-3}$$

As with the AIC, the  $\text{AIC}_C$  should be minimized.

# Comparing regression models

```
glance(fit_trends) %>%  
  select(.model, r_squared, adj_r_squared, AICc, CV)
```

```
## # A tibble: 3 x 5
```

##	.model	r_squared	adj_r_squared	AICc	CV
##	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
## 1	linear	0.728	0.726	452.	39.1
## 2	exponential	0.744	0.742	-779.	0.00176
## 3	piecewise	0.767	0.761	438.	34.8

- Be careful making comparisons when transformations are used.

# Choosing regression variables

## Best subsets regression

- Fit all possible regression models using one or more of the predictors.
- Choose the best model based on one of the measures of predictive ability (CV, AIC, AICc).

# Choosing regression variables

## Best subsets regression

- Fit all possible regression models using one or more of the predictors.
- Choose the best model based on one of the measures of predictive ability (CV, AIC, AICc).

## Warning!

- If there are a large number of predictors, this is not possible.
- For example, 44 predictors leads to 18 trillion possible models!

# Choosing regression variables

## Backwards stepwise regression

- Start with a model containing all variables.
- Try subtracting one variable at a time. Keep the model if it has lower CV or AICc.
- Iterate until no further improvement.

# Choosing regression variables

## Backwards stepwise regression

- Start with a model containing all variables.
- Try subtracting one variable at a time. Keep the model if it has lower CV or AICc.
- Iterate until no further improvement.

## Notes

- Stepwise regression is not guaranteed to lead to the best possible model.
- Inference on coefficients of final model will be wrong.



# Outline

- 1 The linear model with time series
- 2 Some useful predictors for linear models
- 3 Selecting predictors and forecast evaluation
- 4 Forecasting with regression
- 5 Matrix formulation
- 6 Correlation, causation and forecasting

# Ex-ante versus ex-post forecasts

- *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 behaviour of forecasting models.
- trend, seasonal and calendar variables are all known in advance, so these don't need to be forecast.

# Scenario based forecasting

- Assumes possible scenarios for the predictor variables
- Prediction intervals for scenario based forecasts do not include the uncertainty associated with the future values of the predictor variables.

# Building a predictive regression model

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

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

- A different model for each forecast horizon  $h$ .

# Beer production

```
recent_production <- aus_production %>% filter(year(Quarter) >= 1992)
fit_beer <- recent_production %>% model(TSLM(Beer ~ trend() + season()))
fc_beer <- forecast(fit_beer)
fc_beer %>% autoplot(recent_production) +
  ggtitle("Forecasts of beer production using regression") +
  xlab("Year") + ylab("megalitres")
```

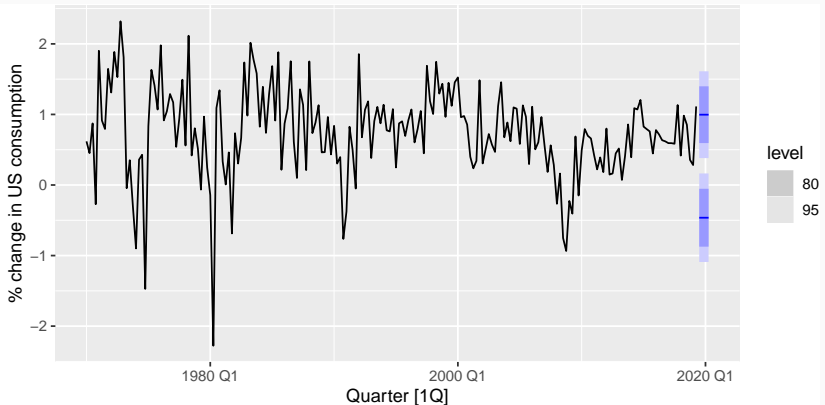


# US Consumption

```
fit_consBest <- us_change %>%  
  model(  
    TSLM(Consumption ~ Income + Savings + Unemployment)  
  )  
  
down_future <- new_data(us_change, 4) %>%  
  mutate(Income = -1, Savings = -0.5, Unemployment = 0)  
fc_down <- forecast(fit_consBest, new_data = down_future)  
  
up_future <- new_data(us_change, 4) %>%  
  mutate(Income = 1, Savings = 0.5, Unemployment = 0)  
fc_up <- forecast(fit_consBest, new_data = up_future)
```

# US Consumption

```
us_change %>% autoplot(Consumption) +  
  ylab("% change in US consumption") +  
  autolayer(fc_up, series = "increase") +  
  autolayer(fc_down, series = "decrease") +  
  guides(colour = guide_legend(title = "Scenario"))
```



# Outline

- 1 The linear model with time series
- 2 Some useful predictors for linear models
- 3 Selecting predictors and forecast evaluation
- 4 Forecasting with regression
- 5 Matrix formulation
- 6 Correlation, causation and forecasting



# Matrix formulation

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t.$$

# Matrix formulation

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \dots + \beta_k x_{k,t} + \varepsilon_t.$$

Let  $\mathbf{y} = (y_1, \dots, y_T)'$ ,  $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_T)'$ ,

$\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_k)'$  and

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \dots & x_{k,1} \\ 1 & x_{1,2} & x_{2,2} & \dots & x_{k,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1,T} & x_{2,T} & \dots & x_{k,T} \end{bmatrix}.$$

# Matrix formulation

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \dots + \beta_k x_{k,t} + \varepsilon_t.$$

Let  $\mathbf{y} = (y_1, \dots, y_T)'$ ,  $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_T)'$ ,

$\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_k)'$  and

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & x_{2,1} & \dots & x_{k,1} \\ 1 & x_{1,2} & x_{2,2} & \dots & x_{k,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1,T} & x_{2,T} & \dots & x_{k,T} \end{bmatrix}.$$

Then

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}.$$

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$

Differentiate wrt  $\boldsymbol{\beta}$  gives

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$

Differentiate wrt  $\beta$  gives

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

(The “normal equation” ’.)

# Matrix formulation

## Least squares estimation

Minimize:  $(\mathbf{y} - \mathbf{X}\beta)'(\mathbf{y} - \mathbf{X}\beta)$

Differentiate wrt  $\beta$  gives

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

(The “normal equation”.)

$$\hat{\sigma}^2 = \frac{1}{T - k - 1}(\mathbf{y} - \mathbf{X}\hat{\beta})'(\mathbf{y} - \mathbf{X}\hat{\beta})$$

**Note:** If you fall for the dummy variable trap,  $(\mathbf{X}'\mathbf{X})$  is a singular matrix.

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$



# Likelihood

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^T (2\pi)^{T/2}} \exp \left( -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right)$$

# Likelihood

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^T (2\pi)^{T/2}} \exp \left( -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right)$$

which is maximized when  $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$  is minimized.

# Likelihood

If the errors are iid and normally distributed, then

$$\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}).$$

So the likelihood is

$$L = \frac{1}{\sigma^T (2\pi)^{T/2}} \exp \left( -\frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \right)$$

which is maximized when  $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$  is minimized.

**So MLE = OLS.**

# Multiple regression forecasts

## Optimal forecasts

$$\hat{y}^* = E(y^* | \mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \mathbf{x}^* \hat{\boldsymbol{\beta}} = \mathbf{x}^* (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}$$

where  $\mathbf{x}^*$  is a row vector containing the values of the predictors for the forecasts (in the same format as  $\mathbf{X}$ ).

# Multiple regression forecasts

## Optimal forecasts

$$\hat{y}^* = E(y^* | \mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \mathbf{x}^* \hat{\boldsymbol{\beta}} = \mathbf{x}^* (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}$$

where  $\mathbf{x}^*$  is a row vector containing the values of the predictors for the forecasts (in the same format as  $\mathbf{X}$ ).

## Forecast variance

$$\text{Var}(y^* | \mathbf{X}, \mathbf{x}^*) = \sigma^2 \left[ 1 + \mathbf{x}^* (\mathbf{X}'\mathbf{X})^{-1} (\mathbf{x}^*)' \right]$$

# Multiple regression forecasts

## Optimal forecasts

$$\hat{y}^* = E(y^* | \mathbf{y}, \mathbf{X}, \mathbf{x}^*) = \mathbf{x}^* \hat{\boldsymbol{\beta}} = \mathbf{x}^* (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{y}$$

where  $\mathbf{x}^*$  is a row vector containing the values of the predictors for the forecasts (in the same format as  $\mathbf{X}$ ).

## Forecast variance

$$\text{Var}(y^* | \mathbf{X}, \mathbf{x}^*) = \sigma^2 \left[ 1 + \mathbf{x}^* (\mathbf{X}'\mathbf{X})^{-1} (\mathbf{x}^*)' \right]$$

- This ignores any errors in  $\mathbf{x}^*$ .
- 95% prediction intervals assuming normal errors:

$$\hat{y}^* \pm 1.96 \sqrt{\text{Var}(y^* | \mathbf{X}, \mathbf{x}^*)}.$$

# Outline

- 1 The linear model with time series
- 2 Some useful predictors for linear models
- 3 Selecting predictors and forecast evaluation
- 4 Forecasting with regression
- 5 Matrix formulation
- 6 Correlation, causation and forecasting

# Correlation is not causation

- When  $x$  is useful for predicting  $y$ , it is not necessarily causing  $y$ .
- e.g., predict number of drownings  $y$  using number of ice-creams sold  $x$ .
- Correlations are useful for forecasting, even when there is no causality.
- Better models usually involve causal relationships (e.g., temperature  $x$  and people  $z$  to predict drownings  $y$ ).



# Multicollinearity

In regression analysis, multicollinearity occurs when:

- Two predictors are highly correlated (i.e., the correlation between them is close to  $\pm 1$ ).
- A linear combination of some of the predictors is highly correlated with another predictor.
- A linear combination of one subset of predictors is highly correlated with a linear combination of another subset of predictors.

# Multicollinearity

If multicollinearity exists...

- the numerical estimates of coefficients may be wrong (worse in Excel than in a statistics package)
- don't rely on the  $p$ -values to determine significance.
- there is no problem with model *predictions* provided the predictors used for forecasting are within the range used for fitting.
- omitting variables can help.
- combining variables can help.

# Outliers and influential observations

## Things to watch for

- *Outliers*: observations that produce large residuals.
- *Influential observations*: removing them would markedly change the coefficients. (Often outliers in the  $x$  variable).
- *Lurking variable*: a predictor not included in the regression but which has an important effect on the response.
- Points should not normally be removed without a good explanation of why they are different.