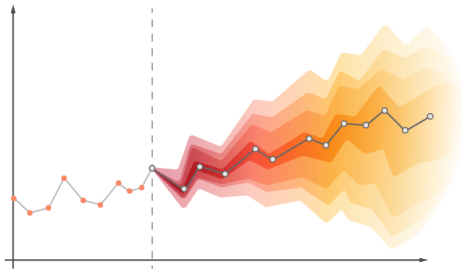


Time Series Regression Models

DS-5740 Advanced Statistics



Overview: Week 2

Preliminaries

- Quick coverage of important syllabus details
- Project 1 rubric

Goals for the Week

- Consider factors involved in forecasting
- Make your first forecast and see forecasts
- Cover linear regression models with time series
- Make forecasts and check their accuracy

Syllabus

- Office Hours:
 - Alex: Tuesdays 2-3pm (Hobbs Building, Room 221); by appointment
 - Marisa: Thursdays 1-2pm (Sony Building, Room 2062)
- Assignments:
 - Due on Sundays at midnight (.Rmd or .html over Brightspace)
 - Late work policy: can be turned in any time there after for 80% of grade (until 11.20.2022)
 - 12 assignments but only your 10 highest grades count

Project 1 Rubric

What can we forecast?

Difficulty of Forecasts

- daily electricity demand in three days
- time of sunrise this day next year
- Google stock price in 6 months (USD)
- maximum temperature tomorrow
- next week's gas prices (USD)

Difficulty of Forecasts

- 1 time of sunrise this day next year
- 2 maximum temperature tomorrow
- 3 daily electricity demand in three days
- 4 next week's gas prices (USD)
- 5 Google stock price in 6 months (USD)

Factors Affecting Difficulty

- 1 knowledge of variables involved
- 2 how much data are available
- 3 similarity of future to past
- 4 whether forecasts are useful

Consideration of Factors

- 1 time of sunrise this day next year

Consideration of Factors

- 1 time of sunrise this day next year
- 2 maximum temperature tomorrow

Consideration of Factors

- 1 time of sunrise this day next year
- 2 maximum temperature tomorrow
- 3 daily electricity demand in three days

Consideration of Factors

- 1 time of sunrise this day next year
- 2 maximum temperature tomorrow
- 3 daily electricity demand in three days
- 4 next week's gas prices (USD)

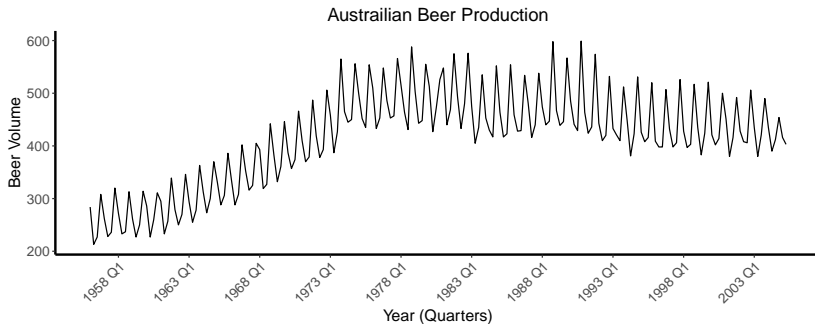
Consideration of Factors

- 1 time of sunrise this day next year
- 2 maximum temperature tomorrow
- 3 daily electricity demand in three days
- 4 next week's gas prices (USD)
- 5 Google stock price in 6 months (USD)

Make a Forecast

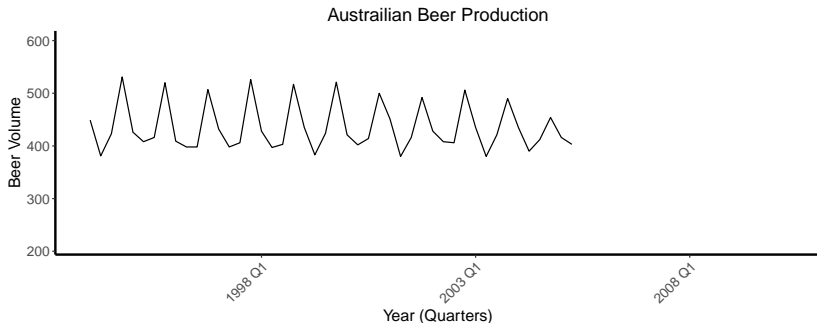
Your Chance to Forecast

forecast: an estimate of the probabilities of possible futures



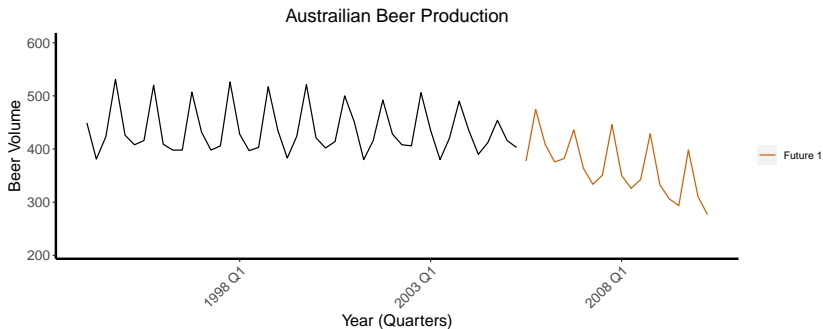
- 1 Problem definition
- 2 Gathering information
- 3 Preliminary (exploratory) analysis
- 4 Choosing and fitting models
- 5 Using and evaluating a forecasting model

forecast: an estimate of the probabilities of possible futures



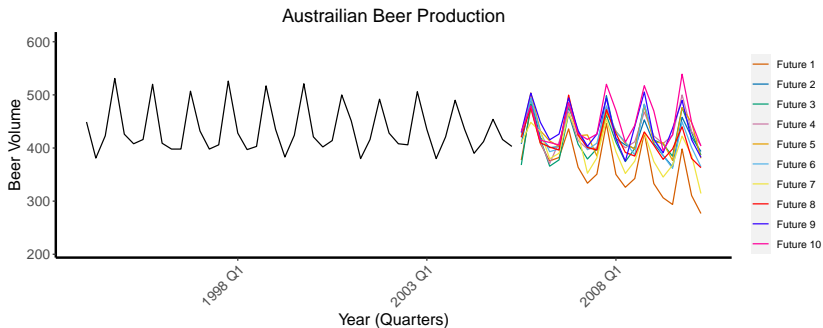
Make a Forecast | One Random Future

forecast: an estimate of the probabilities of possible futures

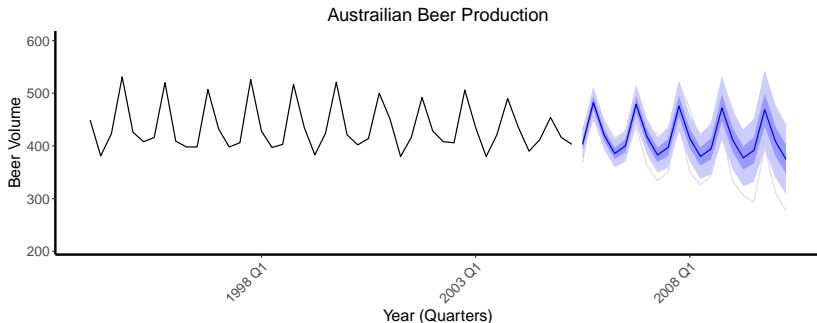


Make a Forecast | Ten Random Futures

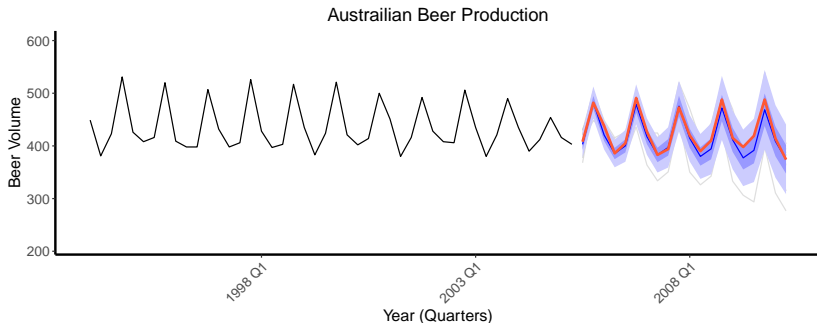
forecast: an estimate of the probabilities of possible futures



forecast: an estimate of the probabilities of possible futures



forecast: an estimate of the probabilities of possible futures



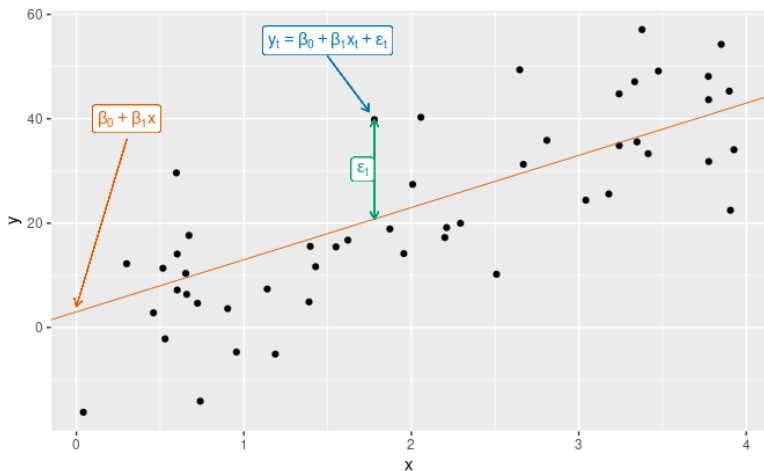
Times Series Linear Model (TSLM)

$$y = \beta_0 + \sum_k^n \beta_k x_k + \epsilon$$

The diagram illustrates the linear regression equation $y = \beta_0 + \sum_k^n \beta_k x_k + \epsilon$. Each term is enclosed in a colored box: y is in a red box, β_0 is in a light blue box, the summation term $\sum_k^n \beta_k x_k$ is in a light green box, and the error term ϵ is in a light orange box. Labels with arrows point to these components: 'outcome' (red) points to y ; 'intercept' (light blue) points to β_0 ; 'sum of weights by predictor' (light green) points to the summation term; and 'error' (light orange) points to ϵ .

$$\text{outcome } y = \text{intercept } \beta_0 + \text{sum of weights by predictor } \sum_k^n \beta_k x_k + \text{error } \epsilon$$

Forecasting | Regression



$$y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$$

outcome (at time t)

sum of weights by predictor (at time t)

$$y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$$

intercept

error (at time t)

The diagram illustrates the components of the time series regression equation. The outcome y_t is shown in a red box, the intercept β_0 in a blue box, the sum of weights by predictor $\sum_k^n \beta_k x_{k,t}$ in a green box, and the error term ϵ_t in an orange box. Arrows connect the descriptive labels to their respective terms in the equation.

The diagram illustrates the components of the time series regression equation $y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$. Each term is enclosed in a colored box, and arrows point from descriptive labels to these boxes:

- outcome (at time t)**: A red label with an arrow pointing to the y_t box.
- intercept**: A blue label with an arrow pointing to the β_0 box.
- sum of weights by predictor (at time t)**: A green label with an arrow pointing to the $\sum_k^n \beta_k x_{k,t}$ box.
- error (at time t)**: An orange label with an arrow pointing to the ϵ_t box.

$$y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$$

- y_t = **outcome** or variable we want to predict

The diagram illustrates the time series regression equation: $y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$. The components are color-coded and labeled with arrows:

- y_t (red box) is labeled "outcome (at time t)" (red text).
- β_0 (blue box) is labeled "intercept" (blue text).
- $\sum_k^n \beta_k x_{k,t}$ (green box) is labeled "sum of weights by predictor (at time t)" (green text).
- ϵ_t (orange box) is labeled "error (at time t)" (orange text).

- y_t = **outcome** or variable we want to predict
- x_k, t = **predictor** or variable used to predict the outcome
 - Usually assumed to be known for all *past* and *future*

The diagram shows the regression equation $y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$ with the following components and labels:

- outcome (at time t)**: Points to y_t (pink box).
- intercept**: Points to β_0 (light blue box).
- sum of weights by predictor (at time t)**: Points to $\sum_k^n \beta_k x_{k,t}$ (light green box).
- error (at time t)**: Points to ϵ_t (orange box).

- y_t = **outcome** or variable we want to predict
- x_k, t = **predictor** or variable used to predict the outcome
 - Usually assumed to be known for all *past* and *future*
- β_k = **coefficients** that measure the effect of each predictor (after taking into account all other predictors)

The diagram shows the equation $y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$ with the following labels and arrows:

- outcome (at time t)**: A red line points to the y_t term.
- intercept**: A blue line points to the β_0 term.
- sum of weights by predictor (at time t)**: A green line points to the $\sum_k^n \beta_k x_{k,t}$ term.
- error (at time t)**: An orange line points to the ϵ_t term.

$$y_t = \beta_0 + \sum_k^n \beta_k x_{k,t} + \epsilon_t$$

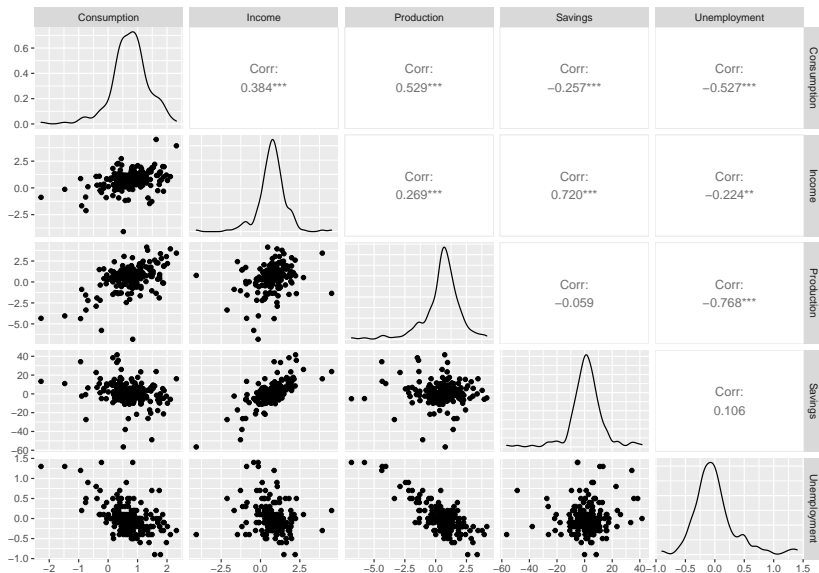
- y_t = **outcome** or variable we want to predict
- x_k, t = **predictor** or variable used to predict the outcome
 - Usually assumed to be known for all *past* and *future*
- β_k = **coefficients** that measure the effect of each predictor (after taking into account all other predictors)
- ϵ_t = white noise error term (we'll talk more on this later)

Regression Example

Forecasting | US Consumption Expenditure



Forecasting | US Consumption Expenditure



Forecasting | US Consumption Expenditure

Series: Consumption

Model: TSLM

Residuals:

	Min	1Q	Median	3Q	Max
	-0.90555	-0.15821	-0.03608	0.13618	1.15471

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.253105	0.034470	7.343	5.71e-12	***
Income	0.740583	0.040115	18.461	< 2e-16	***
Production	0.047173	0.023142	2.038	0.0429	*
Unemployment	-0.174685	0.095511	-1.829	0.0689	.
Savings	-0.052890	0.002924	-18.088	< 2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3102 on 193 degrees of freedom

Multiple R-squared: 0.7683, Adjusted R-squared: 0.7635

F-statistic: 160 on 4 and 193 DF, p-value: < 2.22e-16

Forecasting | Regression Example

```
# Load {fpp3}
library(fpp3)

# Load US Consumption data
data("us_change")

# Length of time series
ts_length <- nrow(us_change)

# Remove last five years (we'll make a prediction later)
us_prediction <- us_change[
  -c((ts_length - 19):ts_length), # remove last 5 years
]

# Save last five years (we'll compare with prediction)
us_actual <- us_change[
  c((ts_length - 19):ts_length), # keeps last 5 years
]
```

Forecasting | Regression Example

```
# Fit linear model  
fit_us_lm <- us_prediction %>% # our data  
  model( # model for time series  
    tslm = TSLM( # time series linear model  
      Consumption ~ Income + Production + Savings + Unemployment  
    )  
  )
```

Forecasting | Regression Example

```
# Report fit
report(fit_us_lm)
```

Series: Consumption
Model: TSLM

Residuals:

Min	1Q	Median	3Q	Max
-0.89952	-0.16879	-0.03979	0.13944	1.14909

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.261795	0.037847	6.917	8.56e-11 ***
Income	0.737779	0.042300	17.442	< 2e-16 ***
Production	0.044788	0.026403	1.696	0.0916 .
Savings	-0.052416	0.003091	-16.960	< 2e-16 ***
Unemployment	-0.191468	0.107811	-1.776	0.0775 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3251 on 173 degrees of freedom

Multiple R-squared: 0.768, Adjusted R-squared: 0.7627

F-statistic: 143.2 on 4 and 173 DF, p-value: < 2.22e-16

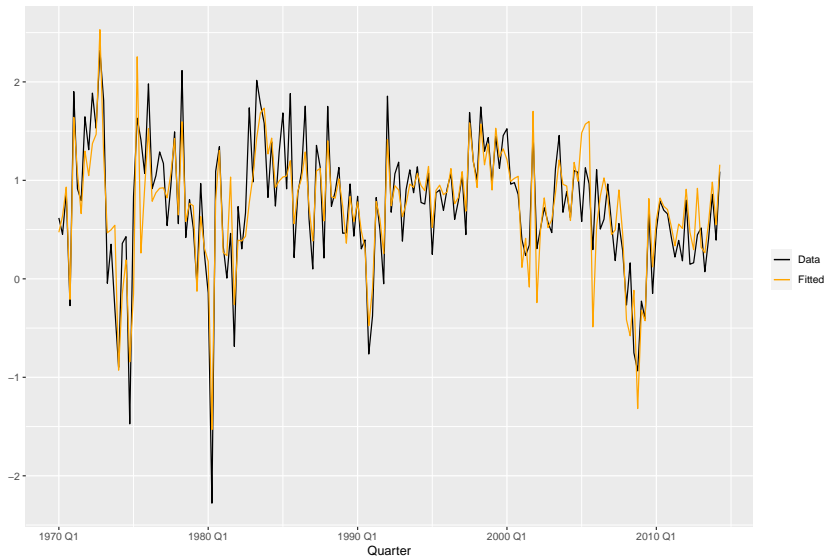
Forecasting with Regression

Forecasting | Regression Example

```
# Plot model
augment(fit_us_lm) %>%
  # Plot quarter on x-axis
  ggplot(aes(x = Quarter)) +
  # Plot actual values
  geom_line(aes(y = Consumption, colour = "Data")) +
  # Plot fit values
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(
    # No y-axis label
    y = NULL,
    # Change title
    title = "Percent change in US consumption expenditure"
  ) +
  # Change colors
  scale_colour_manual(
    values = c(
      Data = "black", # Make data line black
      Fitted = "orange" # Make fitted line orange
    )
  ) +
  # No title for legend
  guides(colour = guide_legend(title = NULL))
```


Forecasting | Regression Example

Percent change in US consumption expenditure



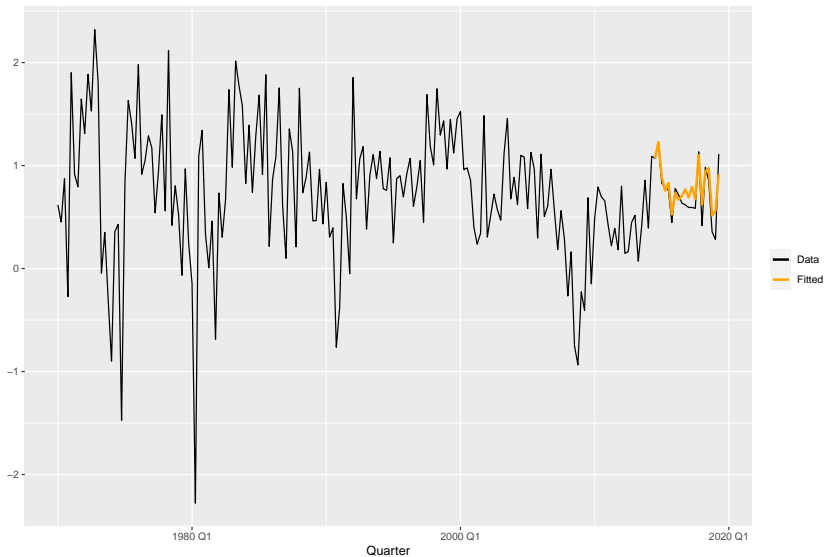
Forecasting | Regression Forecast

```
# Forecast
fc <- forecast(fit_us_lm, new_data = us_actual)

# Plot forecast
us_change %>%
  # Plot quarter on x-axis
  ggplot(aes(x = Quarter)) +
  # Plot actual values
  geom_line(aes(y = Consumption, colour = "Data")) +
  # Plot predicted values
  geom_line(
    data = fc,
    aes(y = .mean, colour = "Fitted"),
    size = 1
  ) +
  labs(
    # No y-axis label
    y = NULL,
    # Change title
    title = "Percent change in US consumption expenditure"
  ) +
  # Change colors
  scale_colour_manual(
    values = c(
      Data = "black", # Make data line black
      Fitted = "orange" # Make fitted line orange
    )
  ) +
  # No title for legend
  guides(colour = guide_legend(title = NULL))
```

Forecasting | Regression Forecast

Percent change in US consumption expenditure



Measures of Accuracy

- R-squared: proportion of variance explained

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

- Mean absolute error: average error

$$MAE = \frac{\sum |\hat{y}_t - y_t|}{T}$$

- Root mean square error: standard deviation of error

$$RMSE = \sqrt{\frac{\sum (\hat{y}_t - y_t)^2}{T}}$$

- Mean bias error: tendency to over- (+) or underestimate (-)

$$MBE = \frac{\sum \hat{y}_t - y_t}{T}$$

```
# R-squared
```

```
cor(fc$.mean, us_actual$Consumption)^2
```

```
[1] 0.8647245
```

```
# MAE
```

```
mean(abs(fc$.mean - us_actual$Consumption))
```

```
[1] 0.1000182
```

```
# RMSE
```

```
sqrt(mean((fc$.mean - us_actual$Consumption)^2))
```

```
[1] 0.1235474
```

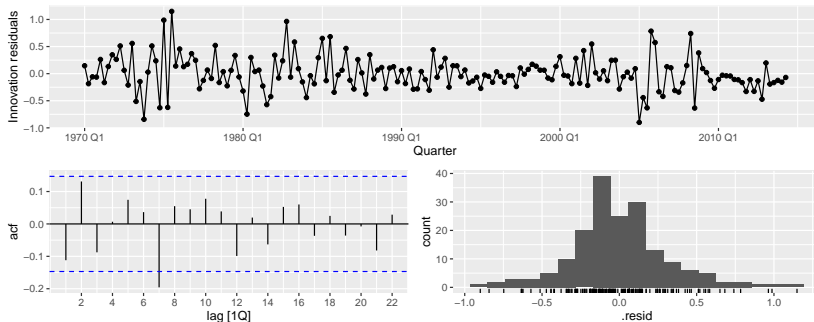
```
# MBE
```

```
mean(fc$.mean - us_actual$Consumption)
```

```
[1] 0.06020543
```

Forecasting | Regression Residuals

```
# Check residuals  
gg_tsresiduals(fit_us_lm)
```

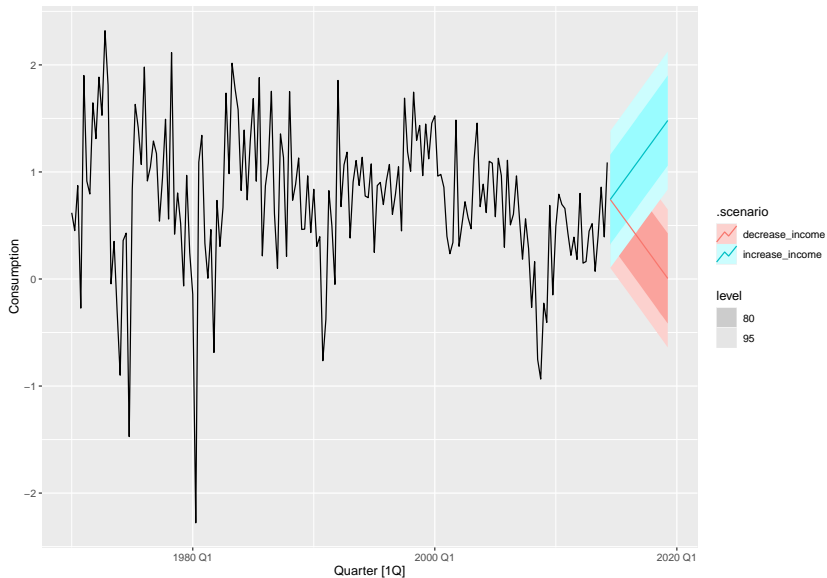


Forecasting | Regression Forecast (no actual data)

```
# Future scenarios
future_scenarios <- scenarios( # Create future scenarios
  increase_income = new_data( # Create new data
    us_prediction, # Original data
    nrow(us_actual) # Number of new data
  ) %>%
  mutate(
    Income = mean(us_prediction$Income) + # Add to mean Income
      seq(0, 1, length = nrow(us_actual)), # Increase from 0 to 1
    # with a length equal to the number of actual data
    Production = mean(us_prediction$Production) +
      rep(0, nrow(us_actual)), # No increase/decrease
    # Repeat 0 with a length equal to the number of actual data
    Savings = mean(us_prediction$Savings) +
      rep(0, nrow(us_actual)),
    Unemployment = mean(us_prediction$Unemployment) +
      rep(0, nrow(us_actual))
  ),
  decrease_income = new_data(
    us_prediction, nrow(us_actual)
  ) %>%
  mutate(
    Income = mean(us_prediction$Income) +
      seq(0, -1, length = nrow(us_actual)),
    Production = mean(us_prediction$Production) +
      rep(0, nrow(us_actual)),
    Savings = mean(us_prediction$Savings) +
      rep(0, nrow(us_actual)),
    Unemployment = mean(us_prediction$Unemployment) +
      rep(0, nrow(us_actual))
  )
)
```

```
# Forecast  
fc_us <- fit_us_lm %>%  
  forecast(new_data = future_scenarios)  
  
# Plot  
autoplot(us_prediction, Consumption) +  
  autolayer(fc_us)
```


Forecasting | Regression Forecast (no actual data)



Regression with Trend and Seasonal Components

The diagram illustrates the components of a regression model. The equation $y_t = \beta_0 + \beta_1 t + \epsilon_t$ is shown with each term in a colored box. Arrows point from descriptive labels to these terms: a red arrow from "outcome (at time t)" to y_t , a blue arrow from "intercept" to β_0 , a green arrow from "trend" to $\beta_1 t$, and an orange arrow from "error (at time t)" to ϵ_t .

$$\text{outcome (at time } t\text{)} \rightarrow y_t = \beta_0 + \beta_1 t + \epsilon_t$$

intercept trend error (at time t)

```
# Fit linear model with trend  
fit_us_trend <- us_prediction %>%  
model( # model for time series  
  tslm = TSLM( # time series linear model  
    Consumption ~ trend() # trend component  
  )  
)
```

Forecasting | Regression Trend Example

```
# Report fit
```

```
report(fit_us_trend)
```

Series: Consumption

Model: TSLM

Residuals:

Min	1Q	Median	3Q	Max
-3.1258	-0.3403	0.0366	0.3867	1.4053

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.9408053	0.0992577	9.478	<2e-16 ***
trend()	-0.0022103	0.0009618	-2.298	0.0227 *

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

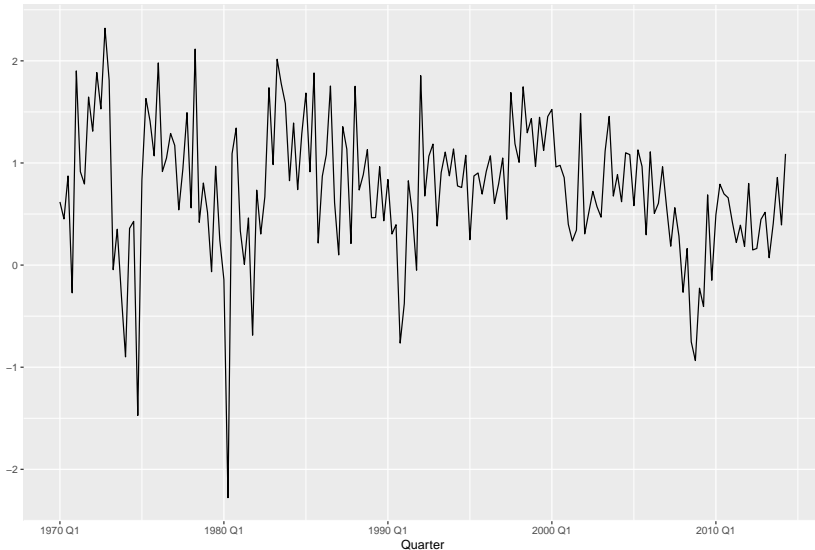
Residual standard error: 0.6593 on 176 degrees of freedom

Multiple R-squared: 0.02913, Adjusted R-squared: 0.02362

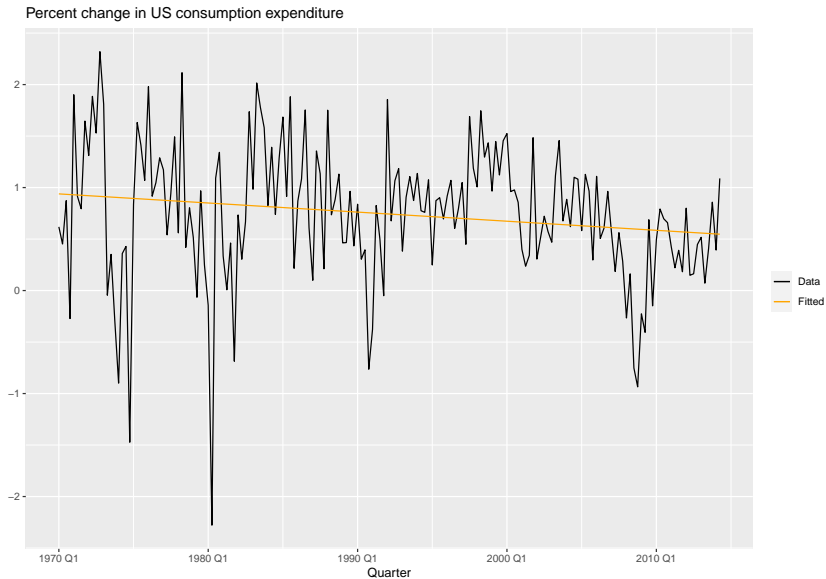
F-statistic: 5.281 on 1 and 176 DF, p-value: 0.022733

Forecasting | Regression Trend Example

Percent change in US consumption expenditure



Forecasting | Regression Trend Example



outcome (at time t)

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

intercept trend season error (at time t)

The diagram illustrates the components of a time series regression model. The equation is $y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + \epsilon_t$. Each term is enclosed in a colored box: y_t is red, β_0 is light blue, $\beta_1 t$ is light green, the seasonal terms $\beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t}$ are purple, and ϵ_t is light orange. Labels with arrows point to these components: 'outcome (at time t)' points to y_t ; 'intercept' points to β_0 ; 'trend' points to $\beta_1 t$; 'season' points to the purple box; and 'error (at time t)' points to ϵ_t .

outcome (at time t)

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

intercept trend season error (at time t)

	$d_{2,t}$	$d_{3,t}$	$d_{4,t}$
Quarter 1	0	0	0
Quarter 2	1	0	0
Quarter 3	0	1	0
Quarter 4	0	0	1
Quarter 1	0	0	0
...

Forecasting | Regression Season Example

```
# Fit linear model with trend and season
fit_us_season <- us_prediction %>%
  model( # model for time series
    tslm = TSLM( # time series linear model
      Consumption ~ trend() + # trend component
      season() # season component
    )
  )
```

Forecasting | Regression Season Example

```
# Report fit
report(fit_us_season)
```

Series: Consumption
Model: TSLM

Residuals:

	Min	1Q	Median	3Q	Max
	-3.07488	-0.33612	0.00766	0.41042	1.46950

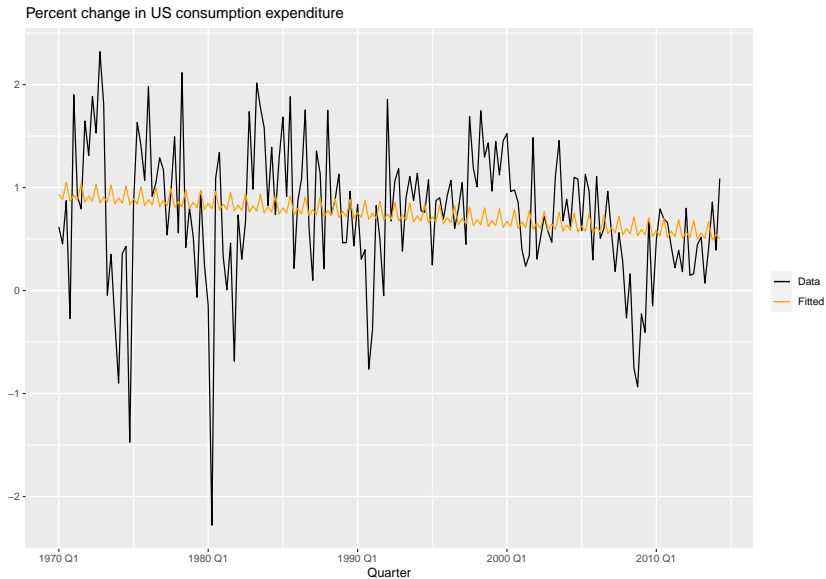
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.9380208	0.1306974	7.177	2.01e-11 ***
trend()	-0.0021995	0.0009645	-2.281	0.0238 *
season()year2	-0.0485962	0.1393858	-0.349	0.7278
season()year3	0.1186395	0.1401721	0.846	0.3985
season()year4	-0.0615712	0.1401754	-0.439	0.6610

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6611 on 173 degrees of freedom
Multiple R-squared: 0.04045, Adjusted R-squared: 0.01826
F-statistic: 1.823 on 4 and 173 DF, p-value: 0.12648

Forecasting | Regression Season Example



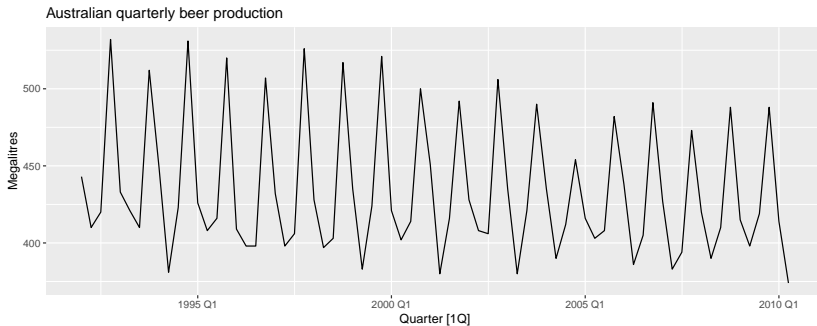
What happened..?

What happened..?

Let's look at a beer-ter example...

Forecasting | A Beer-ter Example

```
# Australian beer production
recent_production <- aus_production %>% filter(year(Quarter) >= 1992)
recent_production %>% autoplot(Beer) +
  labs(y="Megalitres",title="Australian quarterly beer production")
```



Forecasting | A Beer-ter Example

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

Series: Beer
Model: TSLM

Residuals:

Min	1Q	Median	3Q	Max
-42.9029	-7.5995	-0.4594	7.9908	21.7895

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	441.80044	3.73353	118.333	< 2e-16 ***
trend()	-0.34027	0.06657	-5.111	2.73e-06 ***
season()year2	-34.65973	3.96832	-8.734	9.10e-13 ***
season()year3	-17.82164	4.02249	-4.430	3.45e-05 ***
season()year4	72.79641	4.02305	18.095	< 2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

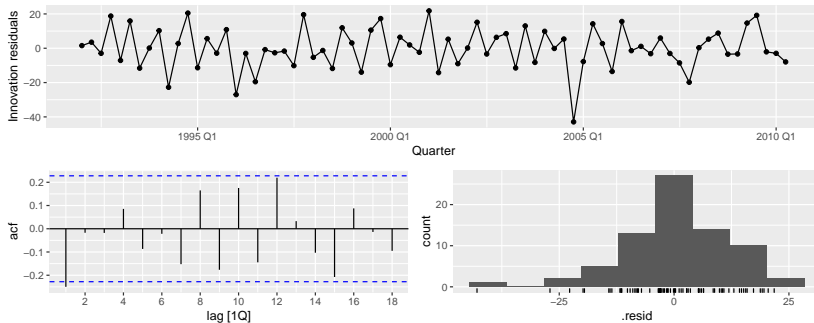
Residual standard error: 12.23 on 69 degrees of freedom

Multiple R-squared: 0.9243, Adjusted R-squared: 0.9199

F-statistic: 210.7 on 4 and 69 DF, p-value: < 2.22e-16

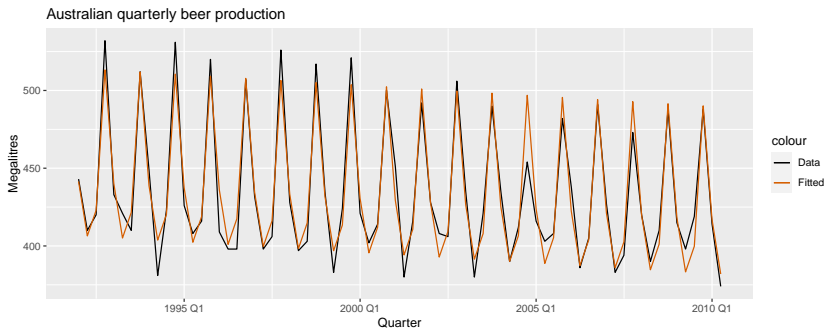
Forecasting | A Beer-ter Example

```
# Residuals  
fit_beer %>%  
  gg_tsresiduals()
```



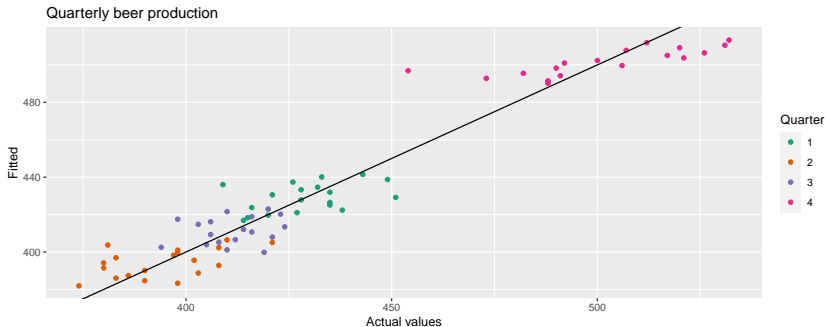
Forecasting | A Beer-ter Example

```
# Plot fitted model
augment(fit_beer) %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Beer, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(y="Megalitres", title = "Australian quarterly beer production") +
  scale_colour_manual(values = c(Data = "black", Fitted = "#D55E00"))
```



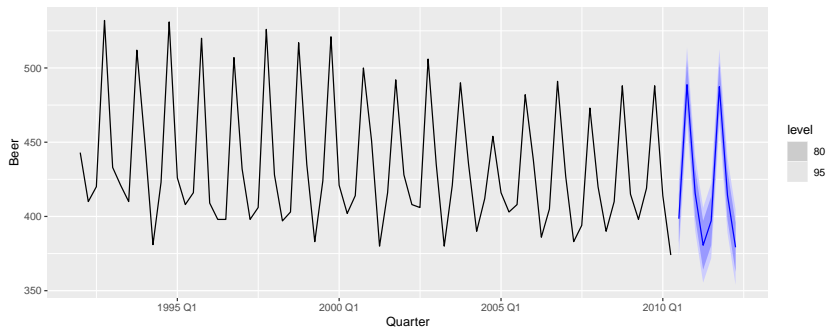
Forecasting | A Beer-ter Example

```
# Examining seasonality
augment(fit_beer) %>%
  ggplot(aes(x=Beer, y=.fitted, colour=factor(quarter(Quarter)))) +
  geom_point() +
  labs(y="Fitted", x="Actual values", title = "Quarterly beer production") +
  scale_colour_brewer(palette="Dark2", name="Quarter") +
  geom_abline(intercept=0, slope=1)
```



Forecasting | A Beer-ter Example

```
# Forecasting prediction
fc <- fit_beer %>% forecast
# Plot forecast
fc %>% autoplot(recent_production)
```



Measures of Fit

- Adjusted R-squared: proportion of variance explained

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

- Cross-validation:

➊ Remove time point t , fit model, and compute error $e_t^* = y_t - \hat{y}_t$

➋ Repeat for each time point T

➌ Compute MSE

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

Measures of Fit

- Akaike's Information Criterion

$$AIC = T \log \left(\frac{SSE}{T} \right) + 2(k + 2)$$

$$SSE = \sum_{t=1}^T e_t^2$$

- Corrected Akaike's Information Criterion

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

- Schwarz's Bayesian Information Criterion

$$BIC = T \log \left(\frac{SSE}{T} \right) + (k + 2) \log(T)$$

```
# Report fit measures
```

```
glance(fit_beer) %>%
```

```
  select(
```

```
    adj_r_squared, CV, AIC, AICc, BIC
```

```
  )
```

```
# A tibble: 1 x 5
```

```
  adj_r_squared    CV    AIC  AICc   BIC
```

```
    <dbl> <dbl> <dbl> <dbl> <dbl>
```

```
1      0.920  160.  377.  379.  391.
```

Dummy Variables

- + Interventions (one time): An effect that lasts only one period. Add a dummy variable with 1 at time point (t)
- + Interventions (permanent): An effect that continues. Add a dummy variable with 1 at time point (t) and each time point there after (t, t_{+1}, \dots, t_n)
- + Number of days: Use number of days in each month as a regressor
- + Lags: Inclusion of previous time points to predict current time point
- + Holidays: Adjust placement of 1 with each year
- + Fourier series (alternative to season): sine and cosine based on m periods (e.g., $m = 52$ for weeks in a year)

Fourier Example

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))
```

Forecasting | Fourier Series

```
# Harmonic regression
fourier_beer <- recent_production %>%
  model( # model for time series
    tslm = TSLM( # time series linear model
      Beer ~ trend() + # trend component
      fourier(K = 2) # harmonic regression
    )
  )

# Report fit
report(fourier_beer)
```

Series: Beer
Model: TSLM

Residuals:

Min	1Q	Median	3Q	Max
-42.9029	-7.5995	-0.4594	7.9908	21.7895

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	446.87920	2.87321	155.533	< 2e-16 ***
trend()	-0.34027	0.06657	-5.111	2.73e-06 ***
fourier(K = 2)C1_4	8.91082	2.01125	4.430	3.45e-05 ***
fourier(K = 2)S1_4	-53.72807	2.01125	-26.714	< 2e-16 ***
fourier(K = 2)C2_4	-13.98958	1.42256	-9.834	9.26e-15 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 12.23 on 69 degrees of freedom

Multiple R-squared: 0.9243, Adjusted R-squared: 0.9199

F-statistic: 210.7 on 4 and 69 DF, p-value: < 2.22e-16

Selecting a model:

```
# Fit multiple models
fit <- recent_production %>%
  model(
    K1 = TSLM(Beer ~ trend() + fourier(K = 1)),
    K2 = TSLM(Beer ~ trend() + fourier(K = 2)),
    K3 = TSLM(Beer ~ trend() + fourier(K = 3)),
    K4 = TSLM(Beer ~ trend() + fourier(K = 4)),
    K5 = TSLM(Beer ~ trend() + fourier(K = 5)),
    K6 = TSLM(Beer ~ trend() + fourier(K = 6))
  )

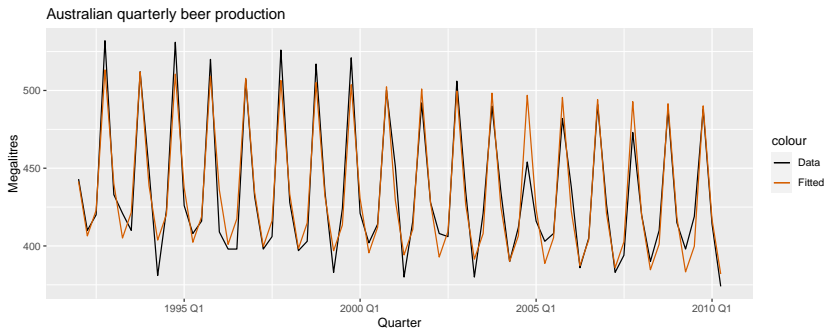
# Check fit
glance(fit) %>% select(.model, r_squared, adj_r_squared, AICc)
```

A tibble: 2 x 4

	.model	r_squared	adj_r_squared	AICc
	<chr>	<dbl>	<dbl>	<dbl>
1	K1	0.818	0.810	441.
2	K2	0.924	0.920	379.

Forecasting | Fourier Series

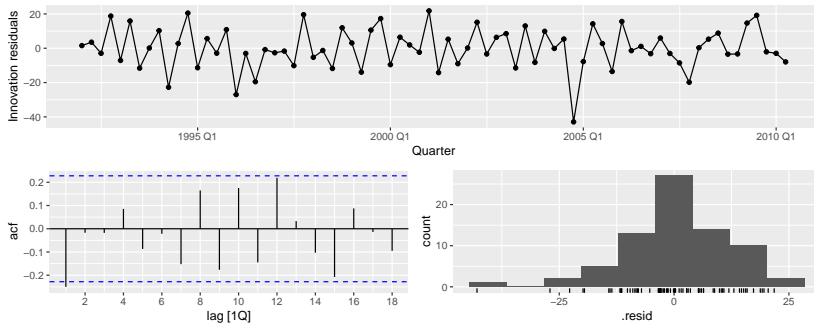
```
# Plot fitted model
augment(fourier_beer) %>%
  ggplot(aes(x = Quarter)) +
  geom_line(aes(y = Beer, colour = "Data")) +
  geom_line(aes(y = .fitted, colour = "Fitted")) +
  labs(y="Megalitres", title="Australian quarterly beer production") +
  scale_colour_manual(values = c(Data = "black", Fitted = "#D55E00"))
```



Residual Diagnostics

Residuals | Computing

```
# Plot fitted model  
fourier_beer %>%  
  gg_tsresiduals()
```



- ϵ_t have zero mean, uncorrelated, and uncorrelated with each $x_{k,t}$
- Normal distribution ($\epsilon_t \sim N(0, \sigma^2)$) **useful** for prediction intervals and statistical tests
- If there is a pattern:
 - predictor used: possible *nonlinear* relationship between residual and predictor
 - predictor *not* used: predictor should be added to model

Reading and Homework

Reading and Homework

- 1 Read Chapter 7 in FPP3 (7.8 and 7.9 optional)
- 2 Use and complete `Week2-Homework.Rmd`
- 3 Save file: `[LASTNAME]_[FIRSTNAME]_Week2-Homework.Rmd`
- 4 Turn in `.Rmd` or HTML over Brightspace by Sunday (04.09.2022)