TABLE OF CONTENTS

Preface

The methods of forecasting have undergone a big change in the recent past with data becoming bigger and variables becoming complicated and multiple. This has necessitated a move in platforms used hitherto like Microsoft Excel to programmes like Python and R. In this book we integrate simple regression forecasting on Microsoft Excel and then assist in learning how this could be done in R language with different methods. If you do not have any programming experience then you can learn forecasting without doing any programming or you may like to take the first step in learning R language which has become indispensable for complicated statistical problems the world tries to find solutions to.

Forecasting is both a science and an art. It is the process of making predictions about future events based on past and present data with parameters that can influence in the future, aiming to reduce uncertainty and assist decision-making. From weather forecasts to artificial intelligence based financial projections, from sales predictions to demographic trends, forecasting permeates nearly every aspect of our lives and endeavours.

In this book, we delve into the fascinating world of forecasting, exploring its principles, methods, and applications across various domains. Whether you are a seasoned professional seeking to refine your forecasting skills or a newcomer curious about the mechanics behind predicting the future, this book is designed to serve as a comprehensive guide.

Our journey begins by laying the groundwork, examining the fundamental concepts of forecasting and the underlying principles that govern it. We then progress to explore a diverse array of forecasting techniques, ranging from classical time series analysis to advanced machine learning algorithms. Through clear explanations, illustrative examples, and practical insights, we aim to equip readers with the knowledge and tools necessary to tackle real-world forecasting challenges with confidence.

However, forecasting is not merely about crunching numbers or applying algorithms. It also requires a deep understanding of the context, domain expertise, and human judgment. Throughout this book, we emphasize the importance of combining quantitative analysis with qualitative insights, embracing uncertainty, and continuously refining our models in light of new information.

Moreover, as the world becomes increasingly complex and inter-connected, the challenges of forecasting grow more intricate. Rapid technological advancements, economic volatility, geopolitical shifts, and societal changes all contribute to the dynamic landscape in which forecasts are made. Thus, we must adapt our approaches and strategies accordingly, embracing agility and innovation in our forecasting endeavours.

Ultimately, the goal of this book is not merely to impart knowledge, but to cultivate a mind-set that embraces uncertainty as an opportunity, that harnesses the power of data and technology while respecting the nuances of human judgment, and that recognises the profound impact forecasting can have on shaping our future.

At the end of each chapter we provide a list of “further reading”. In general, these lists comprise suggested textbooks that provide a more advanced or detailed treatment of the subject. Where there is no suitable textbook, we suggest journal articles that provide more information.

We use R throughout the book and we intend students to learn how to forecast with R. R is free and available on almost every operating system. It is a wonderful tool for all statistical analysis, not just for forecasting. See the Using R appendix for instructions on installing and using R.

All R examples in the book assume you have loaded the *fpp3* package first:

```
library(fpp3)
#> ── Attaching packages ─────────────────────────────────── fpp3 0.5 ──
#> ✓ tibble      3.2.1     ✓ tsibble     1.1.4
#> ✓ dplyr       1.1.4     ✓ tsibbledata 0.4.1
#> ✓ tidyr       1.3.1     ✓ feasts      0.3.2
#> ✓ lubridate   1.9.3     ✓ fable       0.3.4
#> ✓ ggplot2     3.5.0     ✓ fabletools  0.4.1
#> ── Conflicts ───────────────────────────────────── fpp3_conflicts ──
#> ✖ lubridate::date()    masks base::date()
#> ✖ dplyr::filter()      masks stats::filter()
#> ✖ tsibble::intersect() masks base::intersect()
#> ✖ tsibble::interval()  masks lubridate::interval()
#> ✖ dplyr::lag()         masks stats::lag()
#> ✖ tsibble::setdiff()   masks base::setdiff()
#> ✖ tsibble::union()     masks base::union()
```

This will load the relevant data sets, and attach several packages as listed above. These include several tidyverse packages, and packages to handle time series and forecasting in a “tidy” framework. The above output also shows the package versions

we have used in compiling this edition of the book. Some examples in the book will not work with earlier versions of the packages. Finally, the output lists some conflicts showing which function will be preferenced when a function of the same name is in multiple packages.

The book is written for:

(1) people doing forecasting without formal training in the area;

(2) undergraduate students studying business or engineering;

(3) MBA students doing a forecasting elective.

For most sections, we only assume that readers are familiar with introductory statistics, and with high-school algebra. There are a couple of sections that also require knowledge of matrices, but these are also learnt by most in high schools today.

As we embark on this exploration of forecasting, we invite you to approach it with curiosity, humility, and a willingness to engage with both the successes and limitations of our predictive capabilities. May this book serve as a beacon, guiding you through the intricate terrain of forecasting and empowering you to navigate the uncertain seas of tomorrow with clarity and insight.

About the Authors

Dr. Mayur Doshi

Dr. Mayur Doshi is a veteran in research & development industry including Artificial Intelligence with 35 years of experience. A mathematician by practice and Phd in Organic Chemistry by qualification.

He is a director on the board of directors of Falkonry Software Private Limited, India whose parent Company is in US (www.falkonry.com). Falkonry has created Time Series AI - a GPU-scale breakthrough on real-time operational data (patented). By analyzing terabytes of machine and sensor data, Falkonry AI applications identify developing faults earlier and better than would ever be possible with manual systems. Today, Falkonry's time series AI solutions are used by companies – both large multinationals and regional manufacturers – to power their digital transformation and achieve significant improvements in production uptime, quality, yield, and safety. Falkonry can discover insights hidden in operational data and deliver timely, actionable intelligence. They empower our users – plant personnel, process or maintenance engineers, line operators, and analysts – to make better operational decisions with evidence-based approaches.

In the bustling world of finance, where every tick of the clock carries the weight of fortunes and the risk of losses, a new voice has emerged—a first-time author whose ground breaking book is poised to assist the financial markets in redefining themselves. Meet Dr. Mayur Doshi, a visionary mind and the author of the highly anticipated book, "Application of Artificial Intelligence in Finance". This book on Forecasting is essential prequel to the book on "Application of Artificial Intelligence in Finance".

Dr. Mayur, a seasoned expert in artificial intelligence, brings a fresh perspective to the intersection of technology and finance. His journey into the realm of AI applied to finance began with a deep-seated curiosity about the untapped potential of cutting-edge technologies and mathematical curiosity.

Suketu Sanghvi has a diploma in Computer Programming and is a rank holder Chartered Accountant and also a Company Secretary (ranked # 2nd on all India basis) and secured the first rank in computer systems in his bachelor degree from the Mumbai University. He has a work experience of 28 years and started his career in investment banking in India and thereafter, moved to financial centres of Singapore, Hong Kong and Dubai in the United Arab Emirates.

Suketu is an illustrious investment banker who has worked on deals and desks which have won the most prestigious awards globally such as Euromoney awards. In his last employment he was a hedge fund manager at Essdar Capital group in the UAE, managing money and providing investment advisory. Couple of his transactions in middle east won the best Euromoney award deals globally. Where after, he moved to India in 2014 to assist a large number of tech start-ups in India on

a pro-bono basis. Suketu is a co-founder of multiple tech start-ups and other ventures including Zetheta Algorithms Private Limited, Finarcadia Holdings Private Limited, Nashvan Horticulture Industries Private Limited and Essdar Capital (UAE).

He worked for ABN AMRO Bank, India, Singapore & Hong Kong from Jan 2000 to March 2006. His primary assignment was to originate, structure, underwrite and invest in structured debt including special situation, high yield type fixed income transactions (for underwriting and for proprietary investments in ABN AMRO Bank's hold to maturity Asian ABCP conduit book) with embedded derivatives. Part of the most successful DCM teams in Asia (including no # 1 in India) and received Euromoney awards for multiple years, Business India deal award in 2002 and Finance Asia deal award.

Prior to that he worked for UTI Bank (now Axis Bank) from 1997 to Dec 1999. He was part of the team that set-up the investment banking platform for the bank (which grew from around 3 employees when joined to more than 1,000 employees today and laid the foundation to take it to the number # 1 investment bank position in India). While being primarily responsible as credit underwriter and trader, he was also involved in idea generation and setting up of new businesses for the Bank. Prior to UTI Bank, he started his career in Mergers and Acquisitions, Takeovers, Equity IPOs and Venture Capital Investments with marquee investment banking firms.

Copyright

Disclaimer

Neither ZeTheta nor the authors guarantees the accuracy or completeness of any information published herein, and neither ZeTheta nor the authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This book is made available with the understanding that ZeTheta and its authors are providing information to arouse intellectual curiosity/ research on the subject matter but are not attempting to render any professional services, or give recommendations, advice or opinions. If such services are required, the assistance of an appropriate professional should be sought.

ZeTheta is a technology platform and does not act or provide any services of a publisher, broker, sub-broker, investment adviser, financial product distributor, insurance agent, research analyst, portfolio manager, wealth manager, banker, financial adviser in any capacity which is regulated. ZeTheta are not regulated by any Board/ Authority and do not carry out any business which requires licensing/ registration from any Board/ Authority. ZeTheta and the authors are not financial influencer and do not make any recommendation to buy or sell any securities or provide any research on any listed security.

No representation or guarantee is being made in this book as to the future performance/ outcome or forecasts. Any information, and in particular any forecast, outcome or opinion, contained in this book is not intended to predict actual performance, which will differ, and may differ substantially, from those illustrated in the information in this book. In evaluating any illustrative performance, outcome or forecast contained herein or any other information provided in this book, you should understand that not all of the assumptions used herein are described in this book. Conditions and events that are not accounted for or discussed in this book may also have a significant effect on the outcome. Projections are subject to much uncertainty because many of the events that shape the markets as well as future developments in technologies, demographics and resources cannot be foreseen.

Nothing in this book should be construed as investment, legal, tax or financial advice; nor any advice to purchase any security, commodities, futures or options. You should carry out your own due diligence and place no reliance on ZeTheta or the authors. ZeTheta and the authors do not, nor any person connected with it, accept any liability arising from the use of anything in this book. The reader of this book should rely on their own investigations and take their own professional advice.

Chapter 0 Basic concepts before getting started

Here we discuss general non-statistical and few simple statistical aspects of forecasting future cash flows of a business which can be done easily by everyone using simply Microsoft Excel.

There are multiple types of forecasting methods that financial analysts use to predict future revenues, expenses, and capital costs for a business. While there is a wide range of frequently used quantitative budget forecasting tools, the four main methods are: (1) straight-line, (2) moving average, (3) simple linear regression and (4) multiple linear regressions.

Technique	Use	Data needed
Straight line	Constant growth rate	Historical data
Moving average	Repeated forecasts	Historical data
Simple linear regression	Compare one independent with one dependent variable	A sample of relevant observations
Multiple linear regression	Compare more than one independent variable with one dependent variable	A sample of relevant observations

Top down and Bottom Up forecasting

The expanding globalisation of business, the continuing move from push to pull manufacturing, and the rise in consumer oriented economies and services industry with disruptive tech start-ups, have led to a much more complex forecasting and planning world. Forecasters and planners are being asked to create plans for expanding geographies, increased numbers of sales channels, and broader, more diverse, and shorter life cycle product lines. This complexity means that markets are more dynamic and quantitatively based statistical forecast methods are becoming less effective in capturing all that is happening in today's rapidly changing business environment.

More market intelligence now needs to be incorporated during the development of forecasts. In this regard, each Sales and Operations Planning (S&OP) team member may have to generate, review, and revise demand forecasts that reflect the aspects of a business with which they are most familiar. This requires leveraging Top-Down & Bottom-Up forecasting in the process and also usage of other tools such as regression analysis.

Bottom-up forecasting is a method of estimating a company's future performance by starting with low level company data and working "up" to revenue. The opposite approach to bottom-up forecasting is called top-down forecasting, which begins with broad assumptions like Total Addressable Market (TAM) and assuming the company's market share to work "down" to revenue.

Top-down forecasting is extremely useful for improving the accuracy of detailed forecasts. Aggregated demand is less volatile than its individual components, so on a relative basis a forecast of the aggregate is more accurate than the forecasts of its individual components. This is due to the phenomenon of compensating errors where random errors and variations tend to cancel each other out. This is the principle behind the concept of Top-Down forecasting where, rather than forecasting each component separately, it is better to first forecast the aggregated group and then disaggregate the resulting forecast to derive the forecasts of the individual components. The good news is that this principle can be leveraged for any type of aggregation, such as aggregations across companies in an industry, products, sales channels (e.g., stores), geographies, and even time itself.

The use of Bottom-Up forecasting is better for situations where the individual components have different patterns of variation. Under the concept of Bottom-Up forecasting, one forecasts the individual components separately and then adds the forecasts up to get the forecast for the aggregated group.

Generally, Top-Down or Bottom-Up used on an exclusive basis is not the best way to forecast. Often the aggregate group's Bottom-Up forecast can be improved by replacing it with a Top-Down forecast. The individual Bottom-Up component forecasts can be then improved by adjusting each using correction factors derived from looking at the aggregated group's Bottom-Up versus its Top-Down forecast. (For example, if the Bottom-Up forecast predicts aggregate sales to remain flat, while the Top-Down forecast predicts it to grow by 10%, then the correction factor to apply to the bottom level forecasts would be 1.1). Thus, TopDown in conjunction with Bottom-Up, and even Middle-Out is recommended.

There are two ways in which TopDown & Bottom-Up forecasting is useful. Cross-function teams comprised of members from the supply chain, operations, marketing, sales, and finance organisations meet to discuss their plans for generating and satisfying customer demand. The process is driven by a baseline demand forecast that reflects the demand expected from the marketing and sales plans, which in turn

drives the supply plans reflecting the future activities of the operations, manufacturing, logistics, and procurement organisations. Thus, the first (obvious) way in which Top-Down & Bottom-Up forecasting is useful in the S&OP process is during the development of the baseline forecast, in order to take advantage of the accuracy that can be achieved from using both types in conjunction with each other. For example, brand-level forecasts may be most accurately generated at the brand level, and SKU-level forecasts might best be derived from disaggregating the brand-level forecasts using Top-Down forecasting. In turn, product group forecasts might best be derived by aggregating the brand-level forecasts using Bottom-Up forecasting.

The S&OP process also involves refining the supply and demand plans, as well as the baseline-demand forecast. The resulting consensus-based supply and demand plans developed during the process require accountability and commitment from each of the stakeholder organizations involved to ensure each will execute as close as possible to what is embodied in the plans. In order to get this type of buy-in and increase forecast accuracy, each organization needs to participate in the development of the forecasts in terms of reviewing and revising them as necessary.

This is best accomplished by translating and representing the demand forecasts in a form in which each organization is used to dealing with. If marketing's approach to planning, for example, focuses on revenues generated by product groups and brands rather than by unit-based Stock-Keeping Units (SKUs), then any unit-based SKU demand forecasts needs to be aggregated to these product levels on a dollar basis before Marketing could effectively review and revise the forecasts. Meanwhile, if Sales is most familiar with dealing with sales (in INR) by customer accounts and/or sales districts and channels, then demand forecasts needs to be aggregated, disaggregated, and translated into these account groupings before Sales can usefully play its role in the S&OP process. Similarly, Supply Chain managers are most comfortable dealing with forecasts that reflect unit-based SKU and case-level demand, for example; while Finance relates best to forecasts that are aggregated into budgetary units in terms of revenues, costs, and margins.

Thus to get the requisite accountability and commitment from all the organizations involved in the S&OP process requires that forecasts be aggregated and disaggregated (and possibly translated) to various levels to be reviewed and revised by each one, in terms they best understand. This represents another way in which Top-Down & Bottom Up forecasting is useful to the S&OP process. For example, if an organization revises a demand forecast at an aggregated level, then the revision needs to percolate up and down, using Top-Down, BottomUp, and Middle-Out forecasting methods.

There are several other forecast methods, in addition to top-down and bottom-up forecasting, such as regression analysis and Year-over-Year (YoY) analysis. In regression analysis, a financial analyst calculates how changes in independent

variables impact the dependent variable (revenue). Year-over-Year analysis is the simplest method of forecasting where an analyst will look at historical growth rates and apply a growth rate percentage to historical revenue.

In regression, the premise is that changes in the value of a main variable (for example, the sales of Product A) are closely associated with changes in some other variable(s) (for example, the cost of Product B). So, if future values of these other variables (cost of Product B) can be estimated, it can be used to forecast the main variable (sales of Product A).

Regression analysis is a statistical technique for quantifying the relationship between variables. In simple regression analysis, there is one dependent variable (e.g. sales) to be forecast and one independent variable. The values of the independent variable are typically those assumed to "cause" or determine the values of the dependent variable. Thus, if we assume that the amount of advertising dollars spent on a product determines the amount of its sales, we could use regression analysis to quantify the precise nature of the relationship between advertising and sales. For forecasting purposes, knowing the quantified relationship between the variables allows us to provide forecasting estimates.

The simplest regression analysis models the relationship between two variables using the following equation: $Y = a + bX$, where Y is the dependent variable and X is the independent variable. Notice that this simple equation denotes a "linear" relationship between X and Y. So this form would be appropriate if, when you plotted a graph of Y and X, you tended to see the points roughly form along a straight line (as compared to having a curvilinear relationship).

When you have several past concurrent observations of Y and X, regression analysis provides a means to calculate the values of a and b, which are assumed to be constant. Since you will then know a and b, if you can provide an estimate of X in some future period, you can calculate a future value of Y from the above equation.

When using regression models for time series data, we need to distinguish between the different types of forecasts that can be produced, depending on what is assumed to be known when the forecasts are computed.

Ex-ante forecasts are those that are made using only the information that is available in advance. For example, ex-ante forecasts for the percentage change in Indian consumption for quarters following the end of the sample, should only use information that was available up to and including 2024 Q3. These are genuine forecasts, made in advance using whatever information is available at the time. Therefore in order to generate ex-ante forecasts, the model requires forecasts of the predictors. To obtain these we can use pure time series approaches. Alternatively, forecasts from some other source, such as a government agency, may be available and can be used.

Ex-post forecasts are those that are made using later information on the predictors. For example, ex-post forecasts of consumption may use the actual observations of the predictors, once these have been observed. These are not genuine forecasts, but are useful for studying the behaviour of forecasting models. The model from which ex-post forecasts are produced should not be estimated using data from the forecast period. That is, ex-post forecasts can assume knowledge of the predictor variables (the x variables), but should not assume knowledge of the data that are to be forecast (the y variable).

A comparative evaluation of ex-ante forecasts and ex-post forecasts can help to separate out the sources of forecast uncertainty. This will show whether forecast errors have arisen due to poor forecasts of the predictor or due to a poor forecasting model.

The great advantage of regression models is that they can be used to capture important relationships between the forecast variable of interest and the predictor variables. A major challenge however, is that in order to generate ex-ante forecasts, the model require future values of each predictor. If scenario based forecasting is of interest then these models are extremely useful. However, if ex-ante forecasting is the main focus, obtaining forecasts of the predictors can be challenging (in many cases generating forecasts for the predictor variables can be more challenging than forecasting directly the forecast variable without using predictors).

Simple Linear Regression - Example

Regression analysis is a widely used tool for analysing the relationship between variables for prediction purposes. In this example, we will look at the relationship between social media ads and revenue by running a regression analysis on the two variables on Microsoft Excel.

Column A: Month
Column B: Social Media Ads (independent variable x)
Column C: Revenues (dependent variable y)
Revenues based on social media ads appear in rows 5 to 16.

Two formulae's can be used in Microsoft Excel to forecast Revenues based on proposed social media ads for forecast months:

1. FORECAST function.
2. SLOPE FUNCTION and INTERCEPT FUNCTION

We will also draw a scatter chart with the regression line.

Step 1: Populate the data

	A	B	C	D
1	Simple Linear Regression			
2		x	y	
3		Independent variable	Dependent variable	
4	Month	Social Media Ads	Revenue (INR 000)	
5	Jan	764	76,400	
6	Feb	873	89,483	
7	Mar	345	27,600	
8	Apr	654	62,130	
9	May	786	78,600	
10	Jun	380	30,400	
11	Jul	874	89,585	
12	Aug	893	91,533	
13	Sep	897	91,943	
14	Oct	542	48,780	
15	Nov	983	1,03,215	
16	Dec	534	48,060	
17				

Step 2: Use Microsoft Excel formulae's

Forecast the revenue using the FORECAST function. For example, the company proposes to release 1125 social media ads in forecast month 1 and wants to forecast its revenue based on regression. In cell C22, use the formula = FORECAST(B22,C5:C16,B5:B16). This formulae takes data from the Social Media ads and regression data from the actual sales figures based on past performance of social media ads to generate a forecast.

Traditional formulae's of regression forecasting done from calculating slope first and then the Y-Intercept is also available in Miscrosoft Excel.

In cell C18, the formulae is =SLOPE(C5:C16,B5:B16).
In cell C19, the formulae is=INTERCEPT(C5:C16,B5:B16).

Thus the formulae to compute revenue forecast using Slope and the Y-intercept is product of proposed social media ads in a month multiplied by the slope and add the Y-intercept value (if the value is negative then deduct).
Cell D29= A29*B29+C29

	A	B	C	D
1	**Simple Linear Regression**			
2		**x**	**y**	
3		**Independent variable**	**Dependent variable**	
4	**Month**	**Social Media Ads**	**Revenue (INR 000)**	
5	Jan	764	76,400	
6	Feb	873	89,483	
7	Mar	345	27,600	
8	Apr	654	62,130	
9	May	786	78,600	
10	Jun	380	30,400	
11	Jul	874	89,585	
12	Aug	893	91,533	
13	Sep	897	91,943	
14	Oct	542	48,780	
15	Nov	983	1,03,215	
16	Dec	534	48,060	
17				
18	**Slope of the line**		120	
19	**Y-intercept**		-15102	
20				
21	**Forecast with Forecast function**			
22	Month 1	1125	1,19,364	
23	Month 2	1550	1,70,162	
24	Month 3	1750	1,94,067	
25	Month 4	1925	2,14,984	
26				
27	**Forecast using Slope & Y Intercept (check)**			
28	**(Social Media Ads x Slope) + Y-Intercept**			
29	1125	120	-15102	1,19,364
30	1550	120	-15102	1,70,162
31	1750	120	-15102	1,94,067
32	1925	120	-15102	2,14,984

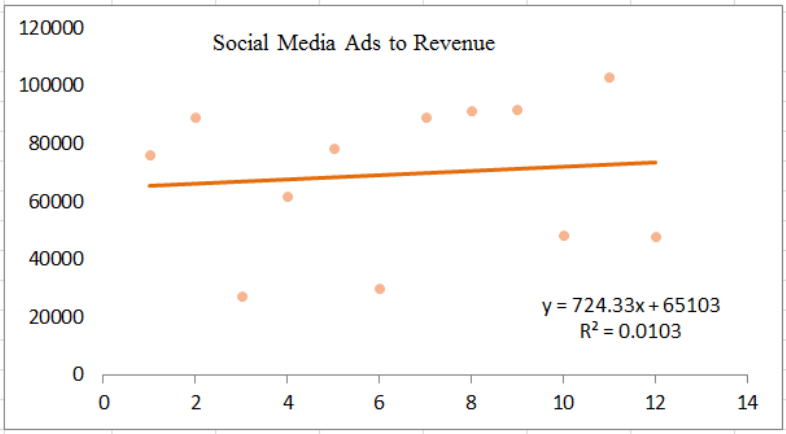
Step 3: Use Microsoft Excel chart for best fit line view

Select the Social Media ads and Revenue data in cell B5 to C16, then go to Insert > Chart > Scatter.

Right-click on the data points and select Format Data Series. Under Marker Options, change the colour to desired and choose no borderline.

Right-click on data points and select Add Trendline. Choose Linear line and check the boxes for Display Equation on the chart and Display R-squared value on the chart. Make other display and clean up changes as desired.

	A	B	C	D
1	**Simple Linear Regression**			
2		**x**	**y**	
3		**Independent variable**	**Dependent variable**	
4	**Month**	**Social Media Ads**	**Revenue (INR 000)**	
5	Jan	764	76,400	
6	Feb	873	89,483	
7	Mar	345	27,600	
8	Apr	654	62,130	
9	May	786	78,600	
10	Jun	380	30,400	
11	Jul	874	89,585	
12	Aug	893	91,533	
13	Sep	897	91,943	
14	Oct	542	48,780	
15	Nov	983	1,03,215	
16	Dec	534	48,060	
17				
18	**Slope of the line**		120	
19	**Y-intercept**		-15102	
20				
21	**Forecast with Forecast function**			
22	Month 1	1125	1,19,364	
23	Month 2	1550	1,70,162	
24	Month 3	1750	1,94,067	
25	Month 4	1925	2,14,984	
26				
27	**Forecast using Slope & Y Intercept (check)**			
28	**(Social Media Ads x Slope) + Y-Intercept**			
29	1125	120	-15102	1,19,364
30	1550	120	-15102	1,70,162
31	1750	120	-15102	1,94,067



Multiple Linear Regression - Example

Multiple linear regressions can be used to forecast revenues when two or more independent variables are required for a projection of a dependent variable. In the example below, we run a regression on direct marketing cost, social media advertising cost and revenue to identify the relationships between these variables using Microsoft Excel.

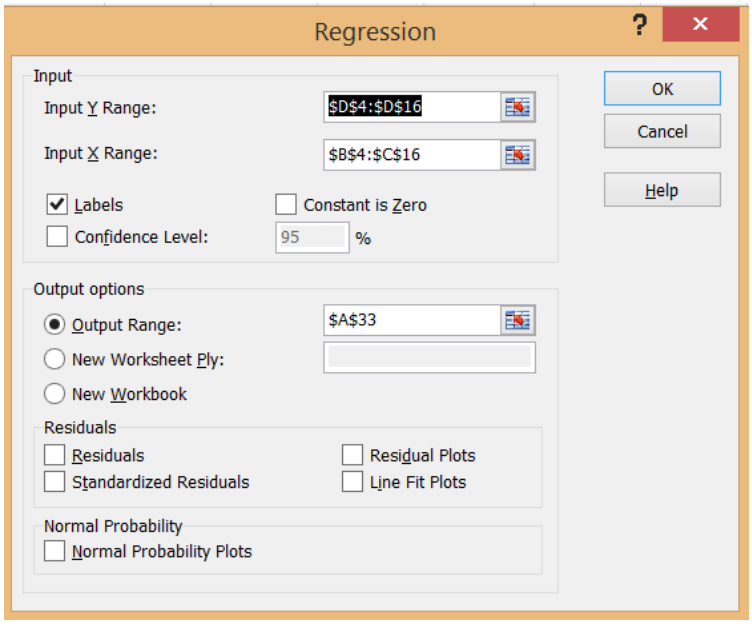
Column A: Month
Column B: Direct Marketing cost (independent variable x1)
Column B: Social Media Ads (independent variable x2)
Column D: Revenues (dependent variable y)
Revenues therefrom appear in column D rows 5 to 16.

Step 1: Populate the data

	A	B	C	D
1	Multiple Linear Regression			
2		x1	x2	y
3		Independent variable	Independent variable	Dependent variable
4	Month	Direct Marketing costs	Social Media Ad costs	Revenue (INR 000)
5	Jan	87300	3,82,000	1,52,800
6	Feb	76210	4,36,500	1,78,965
7	Mar	89720	1,72,500	1,55,200
8	Apr	54320	3,27,000	1,24,260
9	May	54600	3,93,000	1,57,200
10	Jun	90750	1,90,000	1,60,800
11	Jul	34590	4,37,000	1,79,170
12	Aug	45309	4,46,500	1,83,065
13	Sep	34098	4,48,500	1,83,885
14	Oct	87650	2,71,000	1,97,560
15	Nov	34590	4,91,500	2,06,430
16	Dec	78965	2,67,000	1,96,120

Step 2: Run Regression Model

Go to the menu tab>>Data>>Data Analysis and select Regression in the drop down menu items.
Select D4 to D16 for Input Y Range
Select B4 to C16 for Input X Range
Check the box for Labels (*note row 4 is included in formulae above to capture label titles*)
Set Output Range at cell A33.



Step 3: Result of the run on A33 cell

	A	B	C	D	E	F	G	H	I
33	SUMMARY OUTPUT								
34									
35	*Regression Statistics*								
36	Multiple R	0.324599705							
37	R Square	0.105364968							
38	Adjusted R Square	-0.09344282							
39	Standard Error	24518.57011							
40	Observations	12							
41									
42	ANOVA								
43		*df*	*SS*	*MS*	*F*	*ignificance F*			
44	Regression	2	637210800.4	318605400.2	0.529984	0.605908			
45	Residual	9	5410442522	601160280.3					
46	Total	11	6047653323						
47									
48		*Coefficients*	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*Upper 95.0%*
49	Intercept	136893.2315	68993.84996	1.984136726	0.078535	-19181.7	292968.2	-19181.7	292968.2
50	Direct Marketing costs	0.090633729	0.513971667	0.176339932	0.863931	-1.07205	1.253318	-1.07205	1.253318
51	Social Media Ad costs	0.085189507	0.111611734	0.763266586	0.464841	-0.16729	0.337673	-0.16729	0.337673
52									
53		**Relevant Data**							

The relevant data is in cells B49, B50 and B51 which shall be used for computation of the forecasted sales.

Step 4: Result of the run on A33 cell

Copy the co-efficients in the cells B48, B49, B50 and B51 from the summary output and paste it in cell A24, A25, A26 and A27 for ease of applying formulae. We can now forecast the revenue for each future month based on the proposed budget of the direct marketing costs and the social media ad cost.

We can use the equation in cell D18 to forecast revenue:

=$A$25+(B18*$A$26)+(C18*$A$27) for month 1.

We can use the equation in cell D19 to forecast revenue:

=$A$25+(B19*$A$26)+(C19*$A$27) for month 2.

We can use the equation in cell D20 to forecast revenue:

=$A$25+(B20*$A$26)+(C20*$A$27) for month 3.

And so on.

	A	B	C	D
1	**Multiple Linear Regression**			
2		**x1**	**x2**	**y**
3		**Independent variable**	**Independent variable**	**Dependent variable**
4	**Month**	**Direct Marketing costs**	**Social Media Ad costs**	**Revenue (INR 000)**
5	Jan	87,300	3,82,000	1,52,800
6	Feb	76,210	4,36,500	1,78,965
7	Mar	89,720	1,72,500	1,55,200
8	Apr	54,320	3,27,000	1,24,260
9	May	54,600	3,93,000	1,57,200
10	Jun	90,750	1,90,000	1,60,800
11	Jul	34,590	4,37,000	1,79,170
12	Aug	45,309	4,46,500	1,83,065
13	Sep	34,098	4,48,500	1,83,885
14	Oct	87,650	2,71,000	1,97,560
15	Nov	34,590	4,91,500	2,06,430
16	Dec	78,965	2,67,000	1,96,120
17	**Forecast:**			
18	**Month 1**	75,000	3,00,000	1,69,248
19	**Month 2**	98,900	3,25,000	1,73,543
20	**Month 3**	1,25,000	3,75,000	1,80,169
21	**Month 4**	1,40,000	4,60,000	1,88,769
22				
23				
24	*Coefficients*			
25	136893			
26	0.09063			
27	0.08519			

In conclusion, while linear regression models are simple and widely used, but they too have several drawbacks that one should be aware. These drawbacks include limited flexibility, susceptibility to outliers, assumptions of linearity, overfitting, multicollinearity, inability to handle categorical variables, and assumptions of homoscedasticity.

Cost forecasting

Forecasting revenues through any of the approaches above and its relationship with variable costs can be computed through statistical tools. However, cost forecasting has its own nuances. Firstly, there are overheads and fixed costs which may change disproportionately to the change in revenues and then there is inflation and other factors which are external factors not in the control of the Company.

There are many factors that can affect the accuracy of cost forecasting. Some of the most important factors include:

1. The level of detail required for the cost estimate.

2. The complexity of the business.

3. The timeframe involved in the forecast.

4. The size and type of the project or product.

5. The industry in which the project or product is being undertaken.

6. The past performance is critical aspect but will it help for new products to be launched.

7. Changes in economic conditions.

8. Changes in technology.

9. Changes in customer demands.

10. Unconventional or innovative methods or techniques being used in the project or product.

11. The availability of qualified personnel.

12. The availability of necessary resources including fund raising plans and its costs.

13. The impact of changes in government regulations or policies on the project or product.

14. The availability of suitable consultants or advisors.

15. Other factors that may affect cost estimates, such as political and social conditions, weather, etc.

There are a number of methods that can be used to forecast costs, including:

1. Statistical tools – modelling to simulate future events and calculate possible outcomes;

2. Extrapolation - projecting results from known data;

3. Trend analysis - assessing whether a particular pattern is likely to continue;

4. Scenario analysis - creating various possible future scenarios and assessing their effects on costs;

5. Sensitivity analysis - identifying which parameters (elements) have the biggest effect on costs; and

6. Benchmarking - comparing costs against those of similar projects or products to determine relative improvement/ decline (or stability) over time/space etc.

In order to improve accuracy, it is often helpful to use multiple methods to predict costs, as different types of information can give different results (e.g., trend analysis vs scenario analysis). Additionally, it is important to regularly review cost forecasts to ensure that they remain accurate and up-to-date (e.g., by incorporating feedback from stakeholders). If a forecast does not meet expectations, it is often necessary to reassess all assumptions underlying the forecast (e.g., level of detail, complexity, timeframe, etc.).

One material area on cost forecasting that valuers focus on is unstated expenses. For example, the promoters or founders may decide to avoid full salary or to withdraw it gradually after reaching a particular goal. Under such circumstances, the business forecast ought to account for such hidden costs. The valuations of any enterprise will drop if such hidden costs are accounted for in the cash flow forecasts.

Forecasting and risk determination are very much at the heart of practical valuation. Asset value bears on future, uncertain payoffs, so valuation requires forecasting under uncertainty, with both the forecast and the uncertainty priced in the valuation.

Chapter 1 Getting started

Forecasting has fascinated people for thousands of years, sometimes being considered a sign of divine inspiration, and sometimes being seen as a criminal activity. The Jewish prophet Isaiah wrote in about 700 BC

> *Tell us what the future holds, so we may know that you are gods.*
> (Isaiah 41:23)

One hundred years later, in ancient Babylon, forecasters would foretell the future based on the distribution of maggots in a rotten sheep's liver. Around the same time, people wanting forecasts would journey to Delphi in Greece to consult the Oracle, who would provide her predictions while intoxicated by ethylene vapours. Forecasters had a tougher time under the emperor Constantius, who issued a decree in AD357 forbidding anyone "to consult a soothsayer, a mathematician, or a forecaster . . . May curiosity to foretell the future be silenced forever." A similar ban on forecasting occurred in England in 1736 when it became an offence to defraud by charging money for predictions. The punishment was three months' imprisonment with hard labour!

The varying fortunes of forecasters arise because good forecasts can seem almost magical, while bad forecasts may be dangerous. Consider the following famous predictions about computing.

- *I think there is a world market for maybe five computers.* (Chairman of IBM, 1943)
- *Computers in the future may weigh no more than 1.5 tons.* (Popular Mechanics, 1949)
- *There is no reason anyone would want a computer in their home.* (President, DEC, 1977)

The last of these was made only three years before IBM produced the first personal computer. Not surprisingly, you can no longer buy a DEC computer. Forecasting is obviously a difficult activity, and businesses that do it well have a big advantage over those whose forecasts fail.

In this book, we will explore the most reliable methods for producing forecasts. The emphasis will be on methods that are replicable and testable, and have been shown to work.

1.1 What can be forecast?

Forecasting is required in many situations: deciding whether to build another power generation plant in the next five years requires forecasts of future demand; scheduling staff in a call centre next week requires forecasts of call volumes; stocking an inventory requires forecasts of stock requirements. Forecasts can be required several years in advance (for the case of capital investments), or only a few minutes beforehand (for telecommunication routing). Whatever the circumstances or time horizons involved, forecasting is an important aid to effective and efficient planning.

Some things are easier to forecast than others. The time of the sunrise tomorrow morning can be forecast precisely. On the other hand, tomorrow's lotto numbers cannot be forecast with any accuracy. The predictability of an event or a quantity depends on several factors including:

1. how well we understand the factors that contribute to it;
2. how much data is available;
3. whether the forecasts can affect the thing we are trying to forecast.

For example, forecasts of electricity demand can be highly accurate because all three conditions are usually satisfied. We have a good idea of the contributing factors: electricity demand is driven largely by temperatures, with smaller effects for calendar variation such as holidays, and economic conditions. Provided there is a sufficient history of data on electricity demand and weather conditions, and we have the skills to develop a good model linking electricity demand and the key driver variables, the forecasts can be remarkably accurate.

On the other hand, when forecasting currency exchange rates, only one of the conditions is satisfied: there is plenty of available data. However, we have a limited understanding of the factors that affect exchange rates, and forecasts of the exchange rate have a direct effect on the rates themselves. If there are well-publicised forecasts that the exchange rate will increase, then people will immediately adjust the price they are willing to pay and so the forecasts are self-fulfilling. In a sense, the exchange rates become their own forecasts. This is an example of the "efficient market hypothesis". Consequently, forecasting whether the exchange rate will rise or fall tomorrow is about as predictable as forecasting

whether a tossed coin will come down as a head or a tail. In both situations, you will be correct about 50% of the time, whatever you forecast. In situations like this, forecasters need to be aware of their own limitations, and not claim more than is possible.

Often in forecasting, a key step is knowing when something can be forecast accurately, and when forecasts will be no better than tossing a coin. Good forecasts capture the genuine patterns and relationships which exist in the historical data, but do not replicate past events that will not occur again. In this book, we will learn how to tell the difference between a random fluctuation in the past data that should be ignored, and a genuine pattern that should be modelled and extrapolated.

Many people wrongly assume that forecasts are not possible in a changing environment. Every environment is changing, and a good forecasting model captures the way in which things are changing. Forecasts rarely assume that the environment is unchanging. What is normally assumed is that *the way in which the environment is changing* will continue into the future. That is, a highly volatile environment will continue to be highly volatile; a business with fluctuating sales will continue to have fluctuating sales; and an economy that has gone through booms and busts will continue to go through booms and busts. A forecasting model is intended to capture the way things move, not just where things are. As Abraham Lincoln said, "If we could first know where we are and whither we are tending, we could better judge what to do and how to do it".

Forecasting situations vary widely in their time horizons, factors determining actual outcomes, types of data patterns, and many other aspects. Forecasting methods can be simple, such as using the most recent observation as a forecast (which is called the **naïve method**), or highly complex, such as neural nets and econometric systems of simultaneous equations. Sometimes, there will be no data available at all. For example, we may wish to forecast the sales of a new product in its first year, but there are obviously no data to work with. In situations like this, we use judgmental forecasting, discussed in Chapter 4. The choice of method depends on what data are available and the predictability of the quantity to be forecast.

1.2 Forecasting, planning and goals

Forecasting is a common statistical task in business, where it helps to inform decisions about the scheduling of production, transportation and personnel, and provides a guide to long-term strategic planning. However, business forecasting is often done poorly, and is frequently confused with planning and goals. They are three different things.

Forecasting
: is about predicting the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that might impact the forecasts.

Goals
: are what you would like to have happen. Goals should be linked to forecasts and plans, but this does not always occur. Too often, goals are set without any plan for how to achieve them, and no forecasts for whether they are realistic.

Planning
: is a response to forecasts and goals. Planning involves determining the appropriate actions that are required to make your forecasts match your goals.

Forecasting should be an integral part of the decision-making activities of management, as it can play an important role in many areas of a company. Modern organisations require short-term, medium-term and long-term forecasts, depending on the specific application.

Short-term forecasts
: are needed for the scheduling of personnel, production and transportation. As part of the scheduling process, forecasts of demand are often also required.

Medium-term forecasts
: are needed to determine future resource requirements, in order to purchase raw materials, hire personnel, or buy machinery and equipment.

Long-term forecasts
: are used in strategic planning. Such decisions must take account of market opportunities, environmental factors and internal resources.

An organisation needs to develop a forecasting system that involves several approaches to predicting uncertain events. Such forecasting systems require the development of expertise in identifying forecasting problems, applying a range of forecasting methods, selecting appropriate methods for each problem, and evaluating and refining forecasting methods over time. It is also important to have strong organisational support for the use of formal forecasting methods if they are to be used successfully.

1.3 Determining what to forecast

In the early stages of a forecasting project, decisions need to be made about what should be forecast. For example, if forecasts are required for items in a manufacturing environment, it is necessary to ask whether forecasts are needed for:

1. every product line, or for groups of products?
2. every sales outlet, or for outlets grouped by region, or only for total sales?
3. weekly data, monthly data or annual data?

It is also necessary to consider the forecasting horizon. Will forecasts be required for one month in advance, for 6 months, or for ten years? Different types of models will be necessary, depending on what forecast horizon is most important.

How frequently are forecasts required? Forecasts that need to be produced frequently are better done using an automated system than with methods that require careful manual work.

It is worth spending time talking to the people who will use the forecasts to ensure that you understand their needs, and how the forecasts are to be used, before embarking on extensive work in producing the forecasts.

Once it has been determined what forecasts are required, it is then necessary to find or collect the data on which the forecasts will be based. The data required for forecasting may already exist. These days, a lot of data are recorded, and the forecaster's task is often to identify where and how the required data are stored. The data may include sales records of a company, the historical demand for a product, or the unemployment rate for a geographic region. A large part of a forecaster's time can be spent in locating and collating the available data prior to developing suitable forecasting methods.

1.4 Forecasting data and methods

The appropriate forecasting methods depend largely on what data are available.

If there are no data available, or if the data available are not relevant to the forecasts, then **qualitative forecasting** methods must be used. These methods are not purely guesswork—there are well-developed structured approaches to obtaining good forecasts without using historical data. These methods are discussed in Chapter 4.

Quantitative forecasting can be applied when two conditions are satisfied:

1. numerical information about the past is available;
2. it is reasonable to assume that some aspects of the past patterns will continue into the future.

There is a wide range of quantitative forecasting methods, often developed within specific disciplines for specific purposes. Each method has its own properties, accuracies, and costs that must be considered when choosing a specific method.

Most quantitative prediction problems use either time series data (collected at regular intervals over time) or cross-sectional data (collected at a single point in time). In this book we are concerned with forecasting future data, and we concentrate on the time series domain.

Time series forecasting

Examples of time series data include:

- Daily IBM stock prices
- Monthly rainfall
- Quarterly sales results for Amazon
- Annual Google profits

Anything that is observed sequentially over time is a time series. In this book, we will only consider time series that are observed at regular intervals of time (e.g., hourly, daily, weekly, monthly, quarterly, annually). Irregularly spaced time series can also occur, but are beyond the scope of this book.

When forecasting time series data, the aim is to estimate how the sequence of observations will continue into the future. Figure 1.1 shows the quarterly Australian beer production from 1992 to the second quarter of 2010.

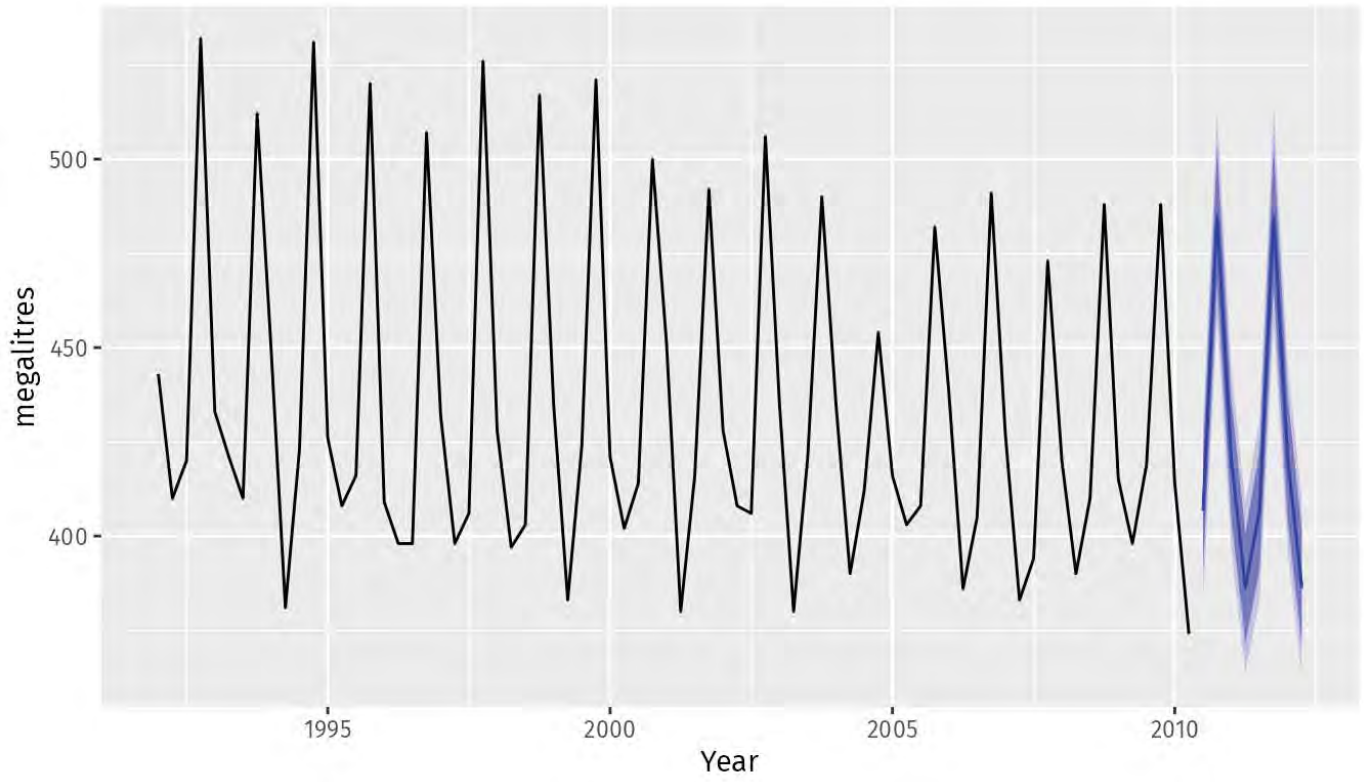


Figure 1.1: Australian quarterly beer production: 1992Q1–2010Q2, with two years of forecasts.

The blue lines show forecasts for the next two years. Notice how the forecasts have captured the seasonal pattern seen in the historical data and replicated it for the next two years. The dark shaded region shows 80% prediction intervals. That is, each future value is expected to lie in the dark shaded region with a probability of 80%. The light shaded region shows 95% prediction intervals. These prediction intervals are a useful way of displaying the uncertainty in forecasts. In this case the forecasts are expected to be accurate, and hence the prediction intervals are quite narrow.

The simplest time series forecasting methods use only information on the variable to be forecast, and make no attempt to discover the factors that affect its behaviour. Therefore they will extrapolate trend and seasonal patterns, but they ignore all other information such as marketing initiatives, competitor activity, changes in economic conditions, and so on.

Time series models used for forecasting include decomposition models, exponential smoothing models and ARIMA models. These models are discussed in Chapters 6, 7 and 8, respectively.

Predictor variables and time series forecasting

Predictor variables are often useful in time series forecasting. For example, suppose we wish to forecast the hourly electricity demand (ED) of a hot region during the summer period. A model with predictor variables might be of the form

$$\begin{aligned}\text{ED} = f(&\text{current temperature, strength of economy, population,}\\ &\text{time of day, day of week, error}).\end{aligned}$$

The relationship is not exact — there will always be changes in electricity demand that cannot be accounted for by the predictor variables. The "error" term on the right allows for random variation and the effects of relevant variables that are not included in the model. We call this an **explanatory model** because it helps explain what causes the variation in electricity demand.

Because the electricity demand data form a time series, we could also use a **time series model** for forecasting. In this case, a suitable time series forecasting equation is of the form

$$\text{ED}_{t+1} = f(\text{ED}_t, \text{ED}_{t-1}, \text{ED}_{t-2}, \text{ED}_{t-3}, \dots, \text{error}),$$

where t is the present hour, $t+1$ is the next hour, $t-1$ is the previous hour, $t-2$ is two hours ago, and so on. Here, prediction of the future is based on past values of a variable, but not on external variables which may affect the system. Again, the "error" term on the right allows for random variation and the effects of relevant variables that are not included in the model.

There is also a third type of model which combines the features of the above two models. For example, it might be given by

$$\text{ED}_{t+1} = f(\text{ED}_t, \text{current temperature, time of day, day of week, error}).$$

These types of **mixed models** have been given various names in different disciplines. They are known as dynamic regression models, panel data models, longitudinal models, transfer function models, and linear system models (assuming that f is linear). These models are discussed in Chapter 9.

An explanatory model is useful because it incorporates information about other variables, rather than only historical values of the variable to be forecast. However, there are several reasons a forecaster might select a time series model rather than an explanatory or mixed model. First, the system may not be understood, and even if it was understood it may be extremely difficult to measure the relationships that are

assumed to govern its behaviour. Second, it is necessary to know or forecast the future values of the various predictors in order to be able to forecast the variable of interest, and this may be too difficult. Third, the main concern may be only to predict what will happen, not to know why it happens. Finally, the time series model may give more accurate forecasts than an explanatory or mixed model.

The model to be used in forecasting depends on the resources and data available, the accuracy of the competing models, and the way in which the forecasting model is to be used.

1.5 Some case studies

The following four cases are from our consulting practice and demonstrate different types of forecasting situations and the associated problems that often arise.

Case 1

The client was a large company manufacturing disposable tableware such as napkins and paper plates. They needed forecasts of each of hundreds of items every month. The time series data showed a range of patterns, some with trends, some seasonal, and some with neither. At the time, they were using their own software, written in-house, but it often produced forecasts that did not seem sensible. The methods that were being used were the following:

1. average of the last 12 months data;
2. average of the last 6 months data;
3. prediction from a straight line regression over the last 12 months;
4. prediction from a straight line regression over the last 6 months;
5. prediction obtained by a straight line through the last observation with slope equal to the average slope of the lines connecting last year's and this year's values;
6. prediction obtained by a straight line through the last observation with slope equal to the average slope of the lines connecting last year's and this year's values, where the average is taken only over the last 6 months.

They required us to tell them what was going wrong and to modify the software to provide more accurate forecasts. The software was written in COBOL, making it difficult to do any sophisticated numerical computation.

Case 2

In this case, the client was the Australian federal government, who needed to forecast the annual budget for the Pharmaceutical Benefit Scheme (PBS). The PBS provides a subsidy for many pharmaceutical products sold in Australia, and the expenditure depends on what people purchase during the year. The total expenditure

was around A$7 billion in 2009, and had been underestimated by nearly $1 billion in each of the two years before we were asked to assist in developing a more accurate forecasting approach.

In order to forecast the total expenditure, it is necessary to forecast the sales volumes of hundreds of groups of pharmaceutical products using monthly data. Almost all of the groups have trends and seasonal patterns. The sales volumes for many groups have sudden jumps up or down due to changes in what drugs are subsidised. The expenditures for many groups also have sudden changes due to cheaper competitor drugs becoming available.

Thus we needed to find a forecasting method that allowed for trend and seasonality if they were present, and at the same time was robust to sudden changes in the underlying patterns. It also needed to be able to be applied automatically to a large number of time series.

Case 3

A large car fleet company asked us to help them forecast vehicle re-sale values. They purchase new vehicles, lease them out for three years, and then sell them. Better forecasts of vehicle sales values would mean better control of profits; understanding what affects resale values may allow leasing and sales policies to be developed in order to maximise profits.

At the time, the resale values were being forecast by a group of specialists. Unfortunately, they saw any statistical model as a threat to their jobs, and were uncooperative in providing information. Nevertheless, the company provided a large amount of data on previous vehicles and their eventual resale values.

Case 4

In this project, we needed to develop a model for forecasting weekly air passenger traffic on major domestic routes for one of Australia's leading airlines. The company required forecasts of passenger numbers for each major domestic route and for each class of passenger (economy class, business class and first class). The company provided weekly traffic data from the previous six years.

Air passenger numbers are affected by school holidays, major sporting events, advertising campaigns, competition behaviour, etc. School holidays often do not coincide in different Australian cities, and sporting events sometimes move from

one city to another. During the period of the historical data, there was a major pilots' strike during which there was no traffic for several months. A new cut-price airline also launched and folded. Towards the end of the historical data, the airline had trialled a redistribution of some economy class seats to business class, and some business class seats to first class. After several months, however, the seat classifications reverted to the original distribution.

1.6 The basic steps in a forecasting task

A forecasting task usually involves five basic steps.

Step 1: Problem definition.

Often this is the most difficult part of forecasting. Defining the problem carefully requires an understanding of the way the forecasts will be used, who requires the forecasts, and how the forecasting function fits within the organisation requiring the forecasts. A forecaster needs to spend time talking to everyone who will be involved in collecting data, maintaining databases, and using the forecasts for future planning.

Step 2: Gathering information.

There are always at least two kinds of information required: (a) statistical data, and (b) the accumulated expertise of the people who collect the data and use the forecasts. Often, it will be difficult to obtain enough historical data to be able to fit a good statistical model. In that case, the judgmental forecasting methods of Chapter 4 can be used. Occasionally, old data will be less useful due to structural changes in the system being forecast; then we may choose to use only the most recent data. However, remember that good statistical models will handle evolutionary changes in the system; don't throw away good data unnecessarily.

Step 3: Preliminary (exploratory) analysis.

Always start by graphing the data. Are there consistent patterns? Is there a significant trend? Is seasonality important? Is there evidence of the presence of business cycles? Are there any outliers in the data that need to be explained by those with expert knowledge? How strong are the relationships among the variables available for analysis? Various tools have been developed to help with this analysis. These are discussed in Chapters 2 and 6.

Step 4: Choosing and fitting models.

The best model to use depends on the availability of historical data, the strength of relationships between the forecast variable and any explanatory variables, and the way in which the forecasts are to be used. It is common to compare two or three potential models. Each model is itself an artificial construct that is based on a set of assumptions (explicit and implicit) and usually involves one or more parameters which must be estimated using the known historical data. We will

discuss regression models (Chapter 5), exponential smoothing methods (Chapter 7), Box-Jenkins ARIMA models (Chapter 8), Dynamic regression models (Chapter 9), Hierarchical forecasting (Chapter 10), and several advanced methods including neural networks and vector autoregression in Chapter 11.

Step 5: Using and evaluating a forecasting model.

Once a model has been selected and its parameters estimated, the model is used to make forecasts. The performance of the model can only be properly evaluated after the data for the forecast period have become available. A number of methods have been developed to help in assessing the accuracy of forecasts. There are also organisational issues in using and acting on the forecasts. A brief discussion of some of these issues is given in Chapter 3. When using a forecasting model in practice, numerous practical issues arise such as how to handle missing values and outliers, or how to deal with short time series. These are discussed in Chapter 12.

1.7 The statistical forecasting perspective

The thing we are trying to forecast is unknown (or we would not be forecasting it), and so we can think of it as a *random variable*. For example, the total sales for next month could take a range of possible values, and until we add up the actual sales at the end of the month, we don't know what the value will be. So until we know the sales for next month, it is a random quantity.

Because next month is relatively close, we usually have a good idea what the likely sales values could be. On the other hand, if we are forecasting the sales for the same month next year, the possible values it could take are much more variable. In most forecasting situations, the variation associated with the thing we are forecasting will shrink as the event approaches. In other words, the further ahead we forecast, the more uncertain we are.

We can imagine many possible futures, each yielding a different value for the thing we wish to forecast. Plotted in black in Figure 1.2 are the total international visitors to Australia from 1980 to 2015. Also shown are ten possible futures from 2016–2025.

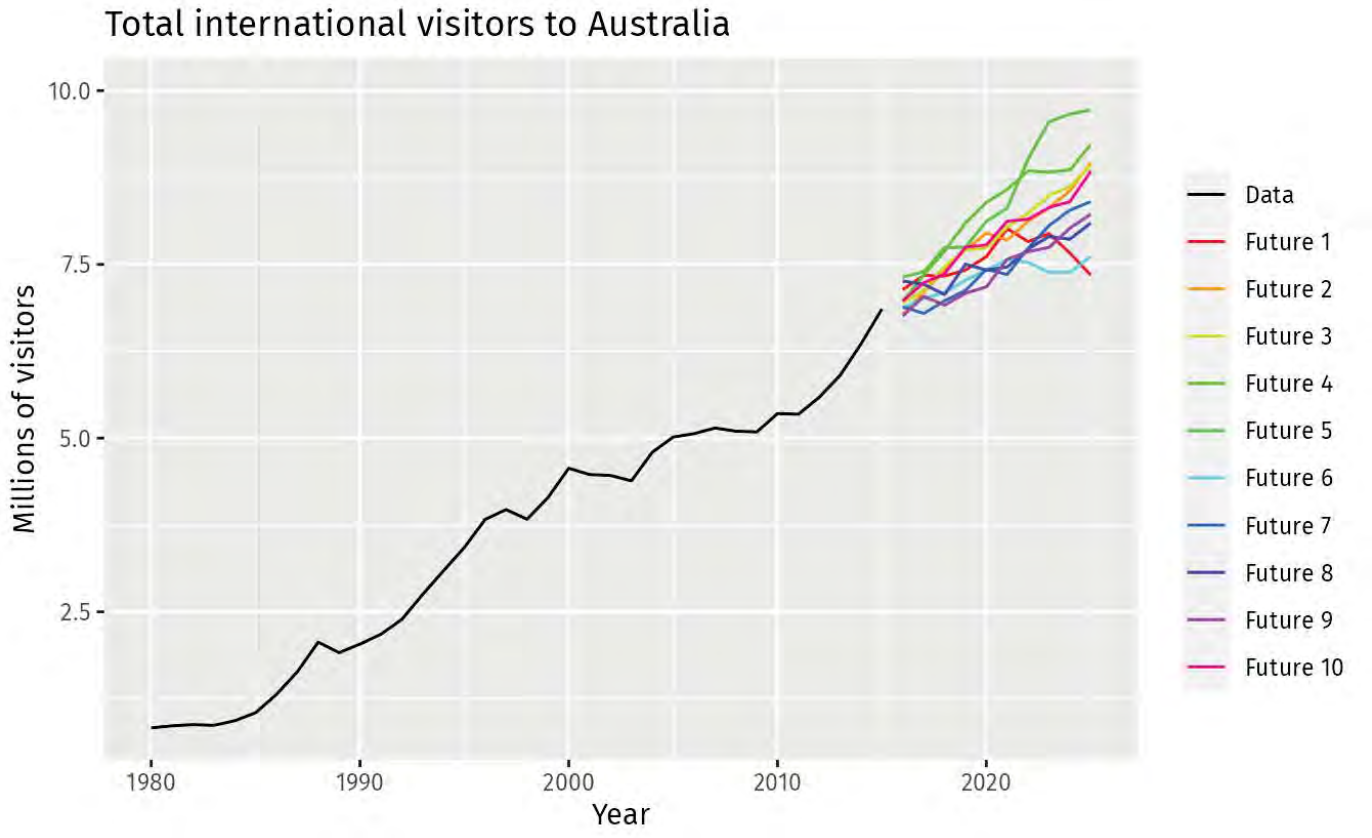


Figure 1.2: Total international visitors to Australia (1980-2015) along with ten possible futures.

When we obtain a forecast, we are estimating the *middle* of the range of possible values the random variable could take. Often, a forecast is accompanied by a **prediction interval** giving a *range* of values the random variable could take with relatively high probability. For example, a 95% prediction interval contains a range of values which should include the actual future value with probability 95%.

Rather than plotting individual possible futures as shown in Figure 1.2, we usually show these prediction intervals instead. The plot below shows 80% and 95% intervals for the future Australian international visitors. The blue line is the average of the possible future values, which we call the **point forecasts**.

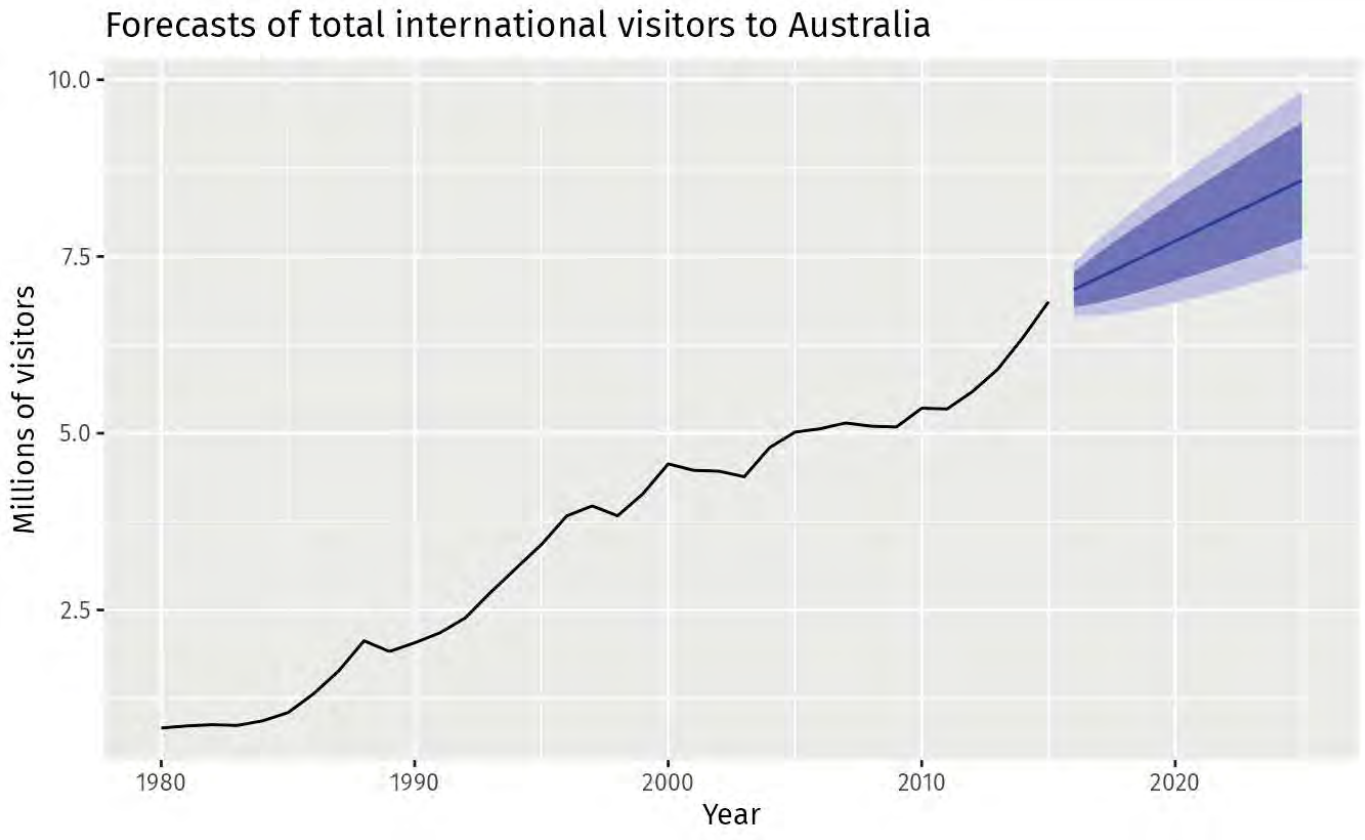


Figure 1.3: Total international visitors to Australia (1980–2015) along with 10-year forecasts and 80% and 95% prediction intervals.

We will use the subscript t for time. For example, y_t will denote the observation at time t. Suppose we denote all the information we have observed as $\mathcal{I}$ and we want to forecast y_t. We then write $y_t|\mathcal{I}$ meaning "the random variable y_t given what we know in $\mathcal{I}$". The set of values that this random variable could take, along with their relative probabilities, is known as the "probability distribution" of $y_t|\mathcal{I}$. In forecasting, we call this the **forecast distribution**.

When we talk about the "forecast", we usually mean the average value of the forecast distribution, and we put a "hat" over y to show this. Thus, we write the forecast of y_t as $\hat{y}_t$, meaning the average of the possible values that y_t could take

given everything we know. Occasionally, we will use $\hat{y}_t$ to refer to the *median* (or middle value) of the forecast distribution instead.

It is often useful to specify exactly what information we have used in calculating the forecast. Then we will write, for example, $\hat{y}_{t|t-1}$ to mean the forecast of y_t taking account of all previous observations $(y_1, \dots, y_{t-1})$. Similarly, $\hat{y}_{T+h|T}$ means the forecast of y_{T+h} taking account of $y_1, \dots, y_T$ (i.e., an h-step forecast taking account of all observations up to time T).

1.8 Exercises

1. For cases 3 and 4 in Section 1.5, list the possible predictor variables that might be useful, assuming that the relevant data are available.
2. For case 3 in Section 1.5, describe the five steps of forecasting in the context of this project.

1.9 Further reading

- Armstrong (2001) covers the whole field of forecasting, with each chapter written by different experts. It is highly opinionated at times (and we don't agree with everything in it), but it is full of excellent general advice on tackling forecasting problems.
- Ord, Fildes, & Kourentzes (2017) is a forecasting textbook covering some of the same areas as this book, but with a different emphasis and not focused around any particular software environment. It is written by three highly respected forecasters, with many decades of experience between them.

Bibliography

Armstrong, J. S. (Ed.). (2001). *Principles of forecasting: A handbook for researchers and practitioners*. Kluwer Academic Publishers. [Amazon]

Ord, J. K., Fildes, R., & Kourentzes, N. (2017). *Principles of business forecasting* (2nd ed.). Wessex Press Publishing Co. [Amazon]

Chapter 2 Time series graphics

The first thing to do in any data analysis task is to plot the data. Graphs enable many features of the data to be visualised, including patterns, unusual observations, changes over time, and relationships between variables. The features that are seen in plots of the data must then be incorporated, as much as possible, into the forecasting methods to be used. Just as the type of data determines what forecasting method to use, it also determines what graphs are appropriate. But before we produce graphs, we need to set up our time series in R.

2.1 `ts` objects

A time series can be thought of as a list of numbers, along with some information about what times those numbers were recorded. This information can be stored as a `ts` object in R.

Suppose you have annual observations for the last few years:

Year	Observation
2012	123
2013	39
2014	78
2015	52
2016	110

We turn this into a `ts` object using the `ts()` function:

```
y <- ts(c(123,39,78,52,110), start=2012)
```

If you have annual data, with one observation per year, you only need to provide the starting year (or the ending year).

For observations that are more frequent than once per year, you simply add a `frequency` argument. For example, if your monthly data is already stored as a numerical vector `z`, then it can be converted to a `ts` object like this:

```
y <- ts(z, start=2003, frequency=12)
```

Almost all of the data used in this book is already stored as `ts` objects. But if you want to work with your own data, you will need to use the `ts()` function before proceeding with the analysis.

Frequency of a time series

The "frequency" is the number of observations before the seasonal pattern repeats.[1] When using the `ts()` function in R, the following choices should be used.

Data	frequency
Annual	1
Quarterly	4
Monthly	12
Weekly	52

Actually, there are not 52 weeks in a year, but $365.25/7 = 52.18$ on average, allowing for a leap year every fourth year. But most functions which use `ts` objects require integer frequency.

If the frequency of observations is greater than once per week, then there is usually more than one way of handling the frequency. For example, data with daily observations might have a weekly seasonality (frequency$= 7$) or an annual seasonality (frequency$= 365.25$). Similarly, data that are observed every minute might have an hourly seasonality (frequency$= 60$), a daily seasonality (frequency $= 24 \times 60 = 1440$), a weekly seasonality (frequency$= 24 \times 60 \times 7 = 10080$) and an annual seasonality (frequency$= 24 \times 60 \times 365.25 = 525960$). If you want to use a `ts` object, then you need to decide which of these is the most important.

In chapter 11 we will look at handling these types of multiple seasonality, without having to choose just one of the frequencies.

1. This is the opposite of the definition of frequency in physics, or in Fourier analysis, where this would be called the "period".↩

2.2 Time plots

For time series data, the obvious graph to start with is a time plot. That is, the observations are plotted against the time of observation, with consecutive observations joined by straight lines. Figure 2.1 below shows the weekly economy passenger load on Ansett Airlines between Australia's two largest cities.

```
autoplot(melsyd[,"Economy.Class"]) +
  ggtitle("Economy class passengers: Melbourne-Sydney") +
  xlab("Year") +
  ylab("Thousands")
```

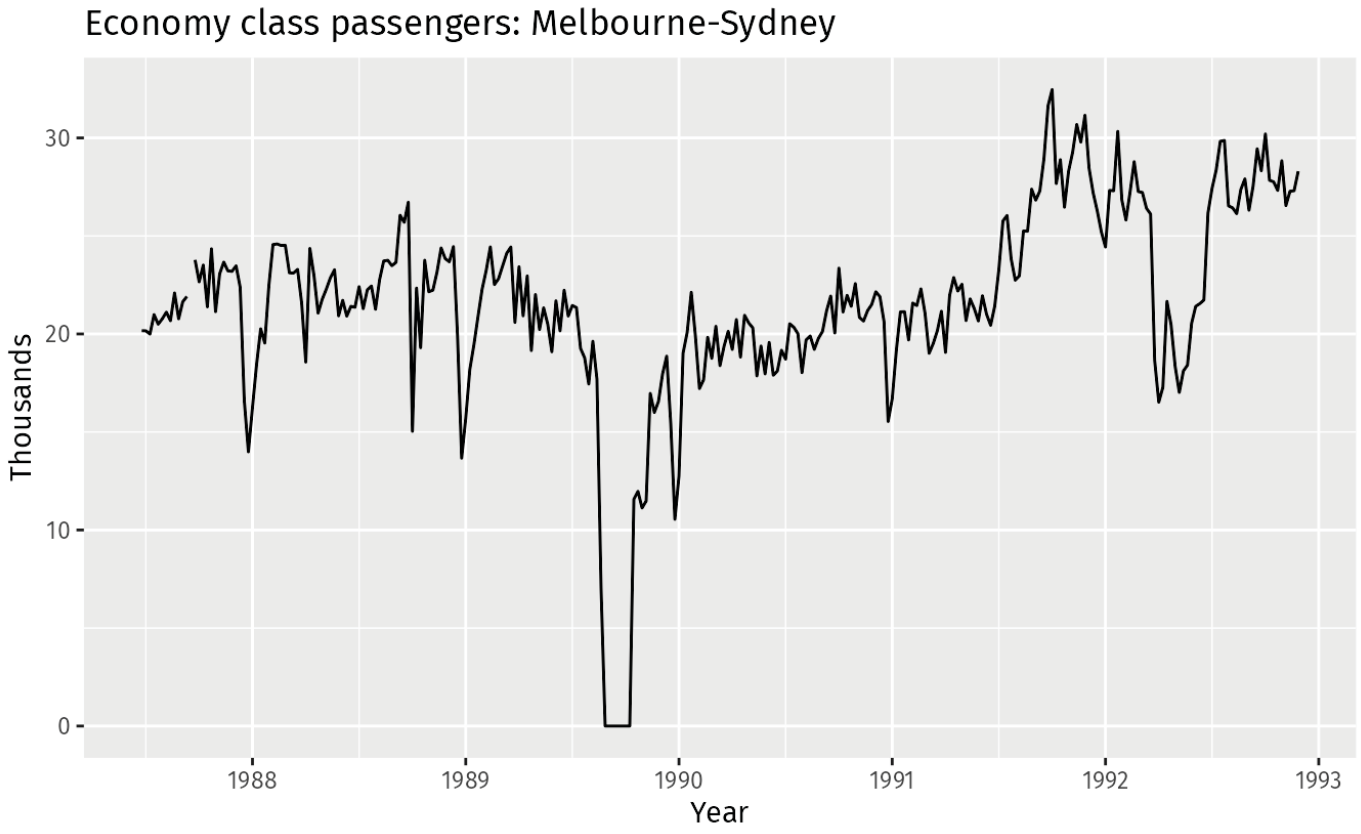


Figure 2.1: Weekly economy passenger load on Ansett Airlines.

We will use the `autoplot()` command frequently. It automatically produces an appropriate plot of whatever you pass to it in the first argument. In this case, it recognises `melsyd[,"Economy.Class"]` as a time series and produces a time plot.

The time plot immediately reveals some interesting features.

- There was a period in 1989 when no passengers were carried — this was due to an industrial dispute.

- There was a period of reduced load in 1992. This was due to a trial in which some economy class seats were replaced by business class seats.
- A large increase in passenger load occurred in the second half of 1991.
- There are some large dips in load around the start of each year. These are due to holiday effects.
- There is a long-term fluctuation in the level of the series which increases during 1987, decreases in 1989, and increases again through 1990 and 1991.
- There are some periods of missing observations.

Any model will need to take all these features into account in order to effectively forecast the passenger load into the future.

A simpler time series is shown in Figure 2.2.

```
autoplot(a10) +
  ggtitle("Antidiabetic drug sales") +
  ylab("$ million") +
  xlab("Year")
```

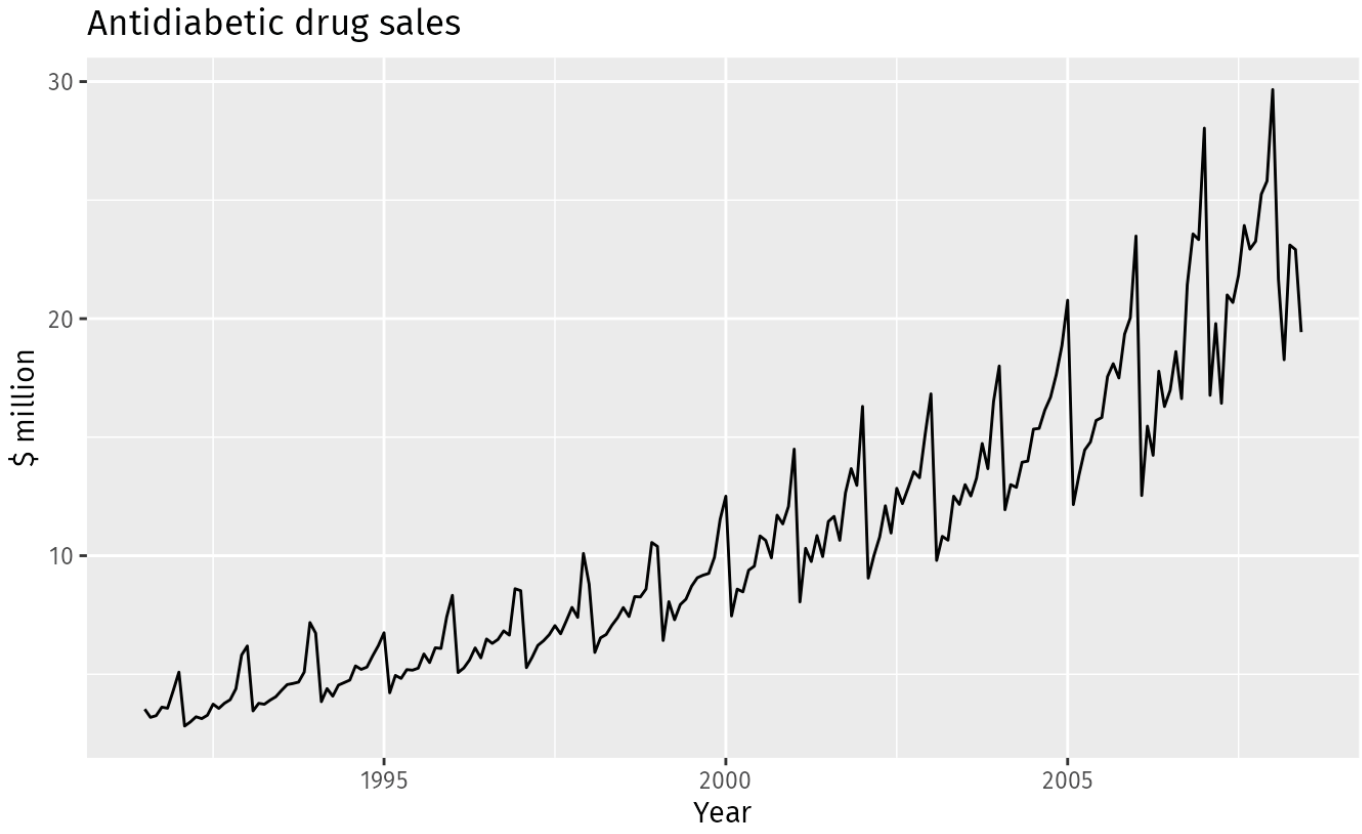


Figure 2.2: Monthly sales of antidiabetic drugs in Australia.

Here, there is a clear and increasing trend. There is also a strong seasonal pattern that increases in size as the level of the series increases. The sudden drop at the start of each year is caused by a government subsidisation scheme that makes it cost-

effective for patients to stockpile drugs at the end of the calendar year. Any forecasts of this series would need to capture the seasonal pattern, and the fact that the trend is changing slowly.

2.3 Time series patterns

In describing these time series, we have used words such as “trend” and “seasonal” which need to be defined more carefully.

Trend

A *trend* exists when there is a long-term increase or decrease in the data. It does not have to be linear. Sometimes we will refer to a trend as “changing direction”, when it might go from an increasing trend to a decreasing trend. There is a trend in the antidiabetic drug sales data shown in Figure 2.2.

Seasonal

A *seasonal* pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. Seasonality is always of a fixed and known frequency. The monthly sales of antidiabetic drugs above shows seasonality which is induced partly by the change in the cost of the drugs at the end of the calendar year.

Cyclic

A *cycle* occurs when the data exhibit rises and falls that are not of a fixed frequency. These fluctuations are usually due to economic conditions, and are often related to the “business cycle”. The duration of these fluctuations is usually at least 2 years.

Many people confuse cyclic behaviour with seasonal behaviour, but they are really quite different. If the fluctuations are not of a fixed frequency then they are cyclic; if the frequency is unchanging and associated with some aspect of the calendar, then the pattern is seasonal. In general, the average length of cycles is longer than the length of a seasonal pattern, and the magnitudes of cycles tend to be more variable than the magnitudes of seasonal patterns.

Many time series include trend, cycles and seasonality. When choosing a forecasting method, we will first need to identify the time series patterns in the data, and then choose a method that is able to capture the patterns properly.

The examples in Figure 2.3 show different combinations of the above components.

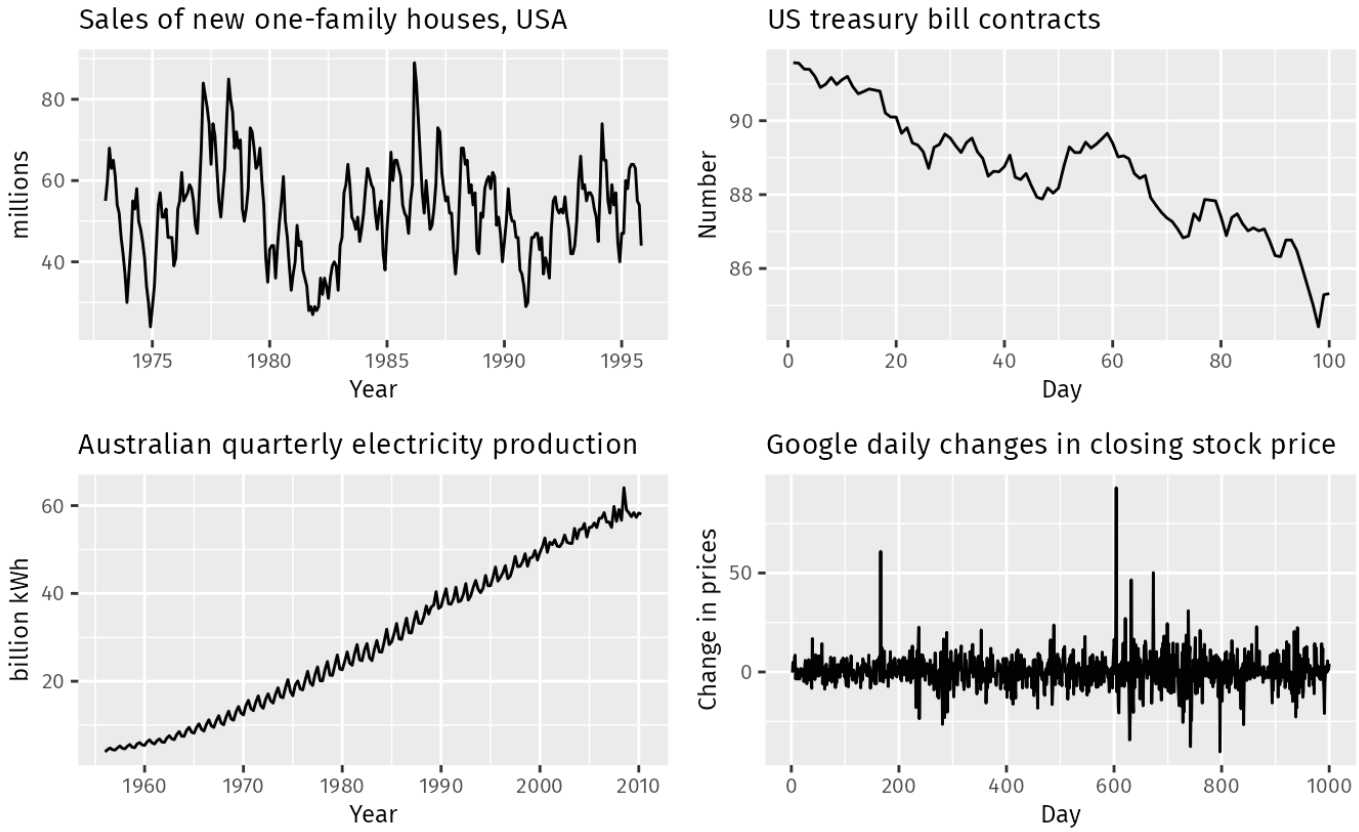


Figure 2.3: Four examples of time series showing different patterns.

1. The monthly housing sales (top left) show strong seasonality within each year, as well as some strong cyclic behaviour with a period of about 6–10 years. There is no apparent trend in the data over this period.
2. The US treasury bill contracts (top right) show results from the Chicago market for 100 consecutive trading days in 1981. Here there is no seasonality, but an obvious downward trend. Possibly, if we had a much longer series, we would see that this downward trend is actually part of a long cycle, but when viewed over only 100 days it appears to be a trend.
3. The Australian quarterly electricity production (bottom left) shows a strong increasing trend, with strong seasonality. There is no evidence of any cyclic behaviour here.
4. The daily change in the Google closing stock price (bottom right) has no trend, seasonality or cyclic behaviour. There are random fluctuations which do not appear to be very predictable, and no strong patterns that would help with developing a forecasting model.

2.4 Seasonal plots

A seasonal plot is similar to a time plot except that the data are plotted against the individual "seasons" in which the data were observed. An example is given below showing the antidiabetic drug sales.

```
ggseasonplot(a10, year.labels=TRUE, year.labels.left=TRUE) +
  ylab("$ million") +
  ggtitle("Seasonal plot: antidiabetic drug sales")
```

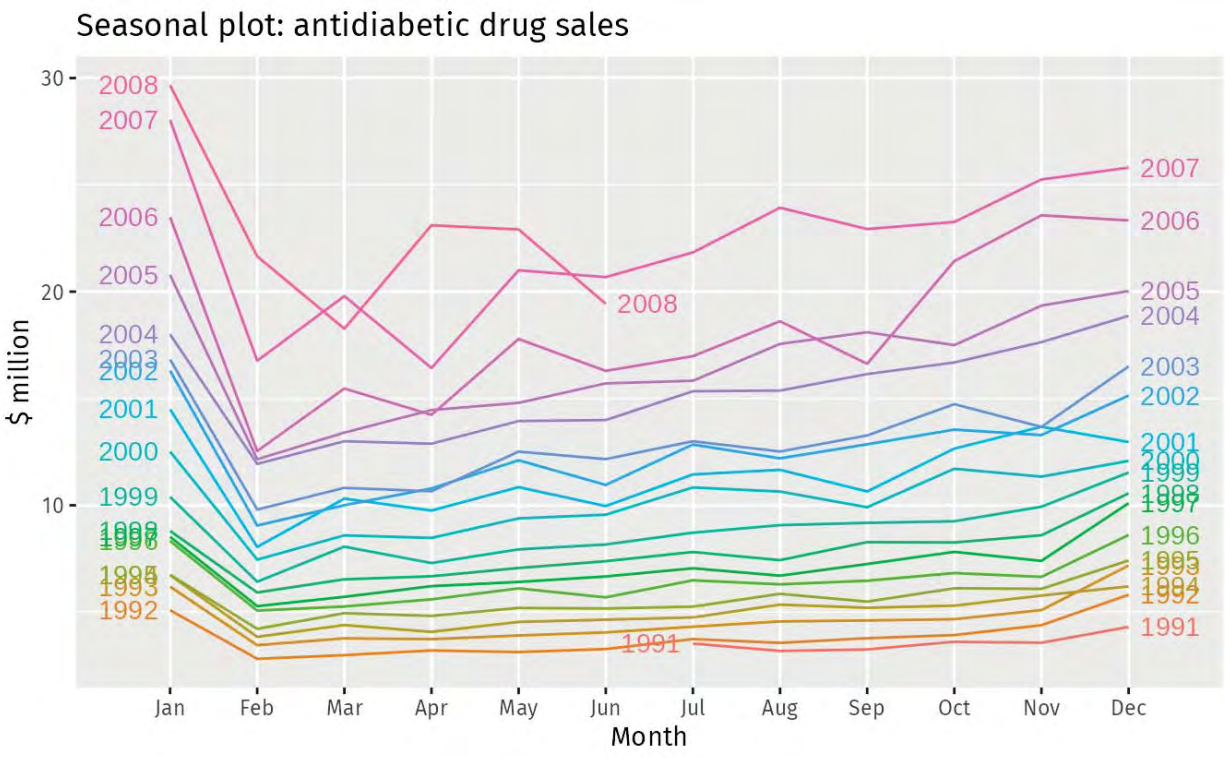


Figure 2.4: Seasonal plot of monthly antidiabetic drug sales in Australia.

These are exactly the same data as were shown earlier, but now the data from each season are overlapped. A seasonal plot allows the underlying seasonal pattern to be seen more clearly, and is especially useful in identifying years in which the pattern changes.

In this case, it is clear that there is a large jump in sales in January each year. Actually, these are probably sales in late December as customers stockpile before the end of the calendar year, but the sales are not registered with the government until a week or two later. The graph also shows that there was an unusually small number of

sales in March 2008 (most other years show an increase between February and March). The small number of sales in June 2008 is probably due to incomplete counting of sales at the time the data were collected.

A useful variation on the seasonal plot uses polar coordinates. Setting `polar=TRUE` makes the time series axis circular rather than horizontal, as shown below.

```
ggseasonplot(a10, polar=TRUE) +
  ylab("$ million") +
  ggtitle("Polar seasonal plot: antidiabetic drug sales")
```

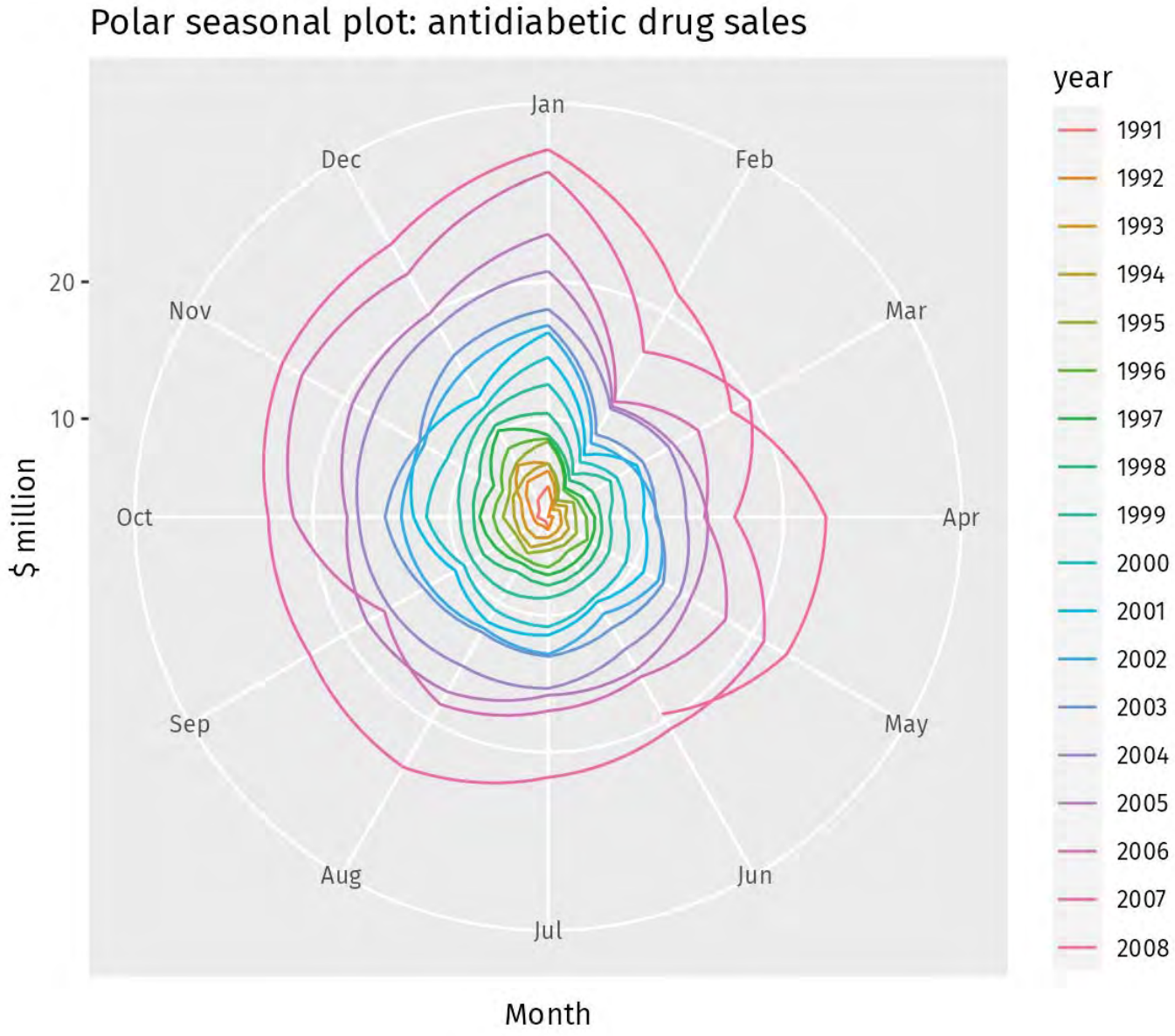


Figure 2.5: Polar seasonal plot of monthly antidiabetic drug sales in Australia.

2.5 Seasonal subseries plots

An alternative plot that emphasises the seasonal patterns is where the data for each season are collected together in separate mini time plots.

```
ggsubseriesplot(a10) +
  ylab("$ million") +
  ggtitle("Seasonal subseries plot: antidiabetic drug sales")
```

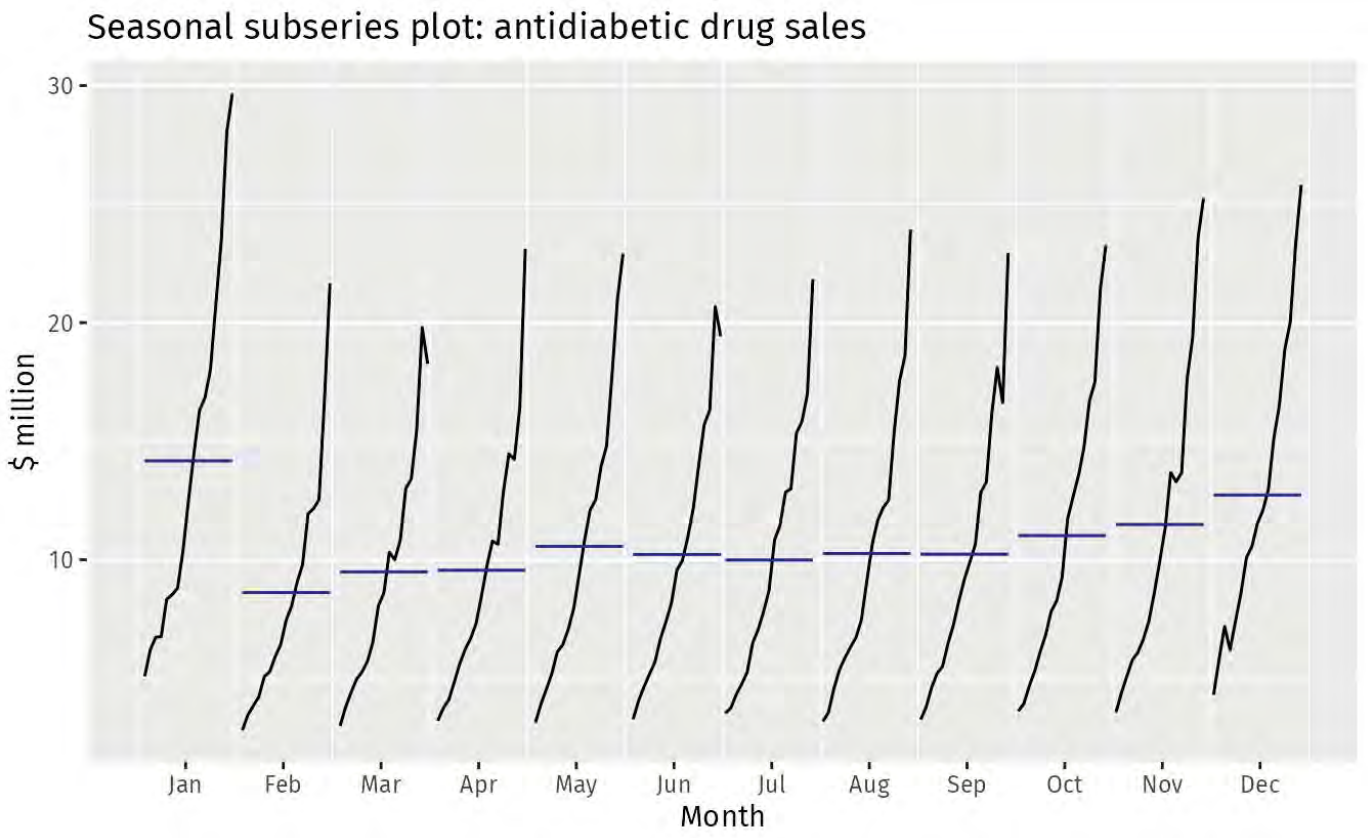


Figure 2.6: Seasonal subseries plot of monthly antidiabetic drug sales in Australia.

The horizontal lines indicate the means for each month. This form of plot enables the underlying seasonal pattern to be seen clearly, and also shows the changes in seasonality over time. It is especially useful in identifying changes within particular seasons. In this example, the plot is not particularly revealing; but in some cases, this is the most useful way of viewing seasonal changes over time.

2.6 Scatterplots

The graphs discussed so far are useful for visualising individual time series. It is also useful to explore relationships *between* time series.

Figure 2.7 shows two time series: half-hourly electricity demand (in Gigawatts) and temperature (in degrees Celsius), for 2014 in Victoria, Australia. The temperatures are for Melbourne, the largest city in Victoria, while the demand values are for the entire state.

```
autoplot(elecdemand[,c("Demand","Temperature")], facets=TRUE) +
  xlab("Year: 2014") + ylab("") +
  ggtitle("Half-hourly electricity demand: Victoria, Australia")
```

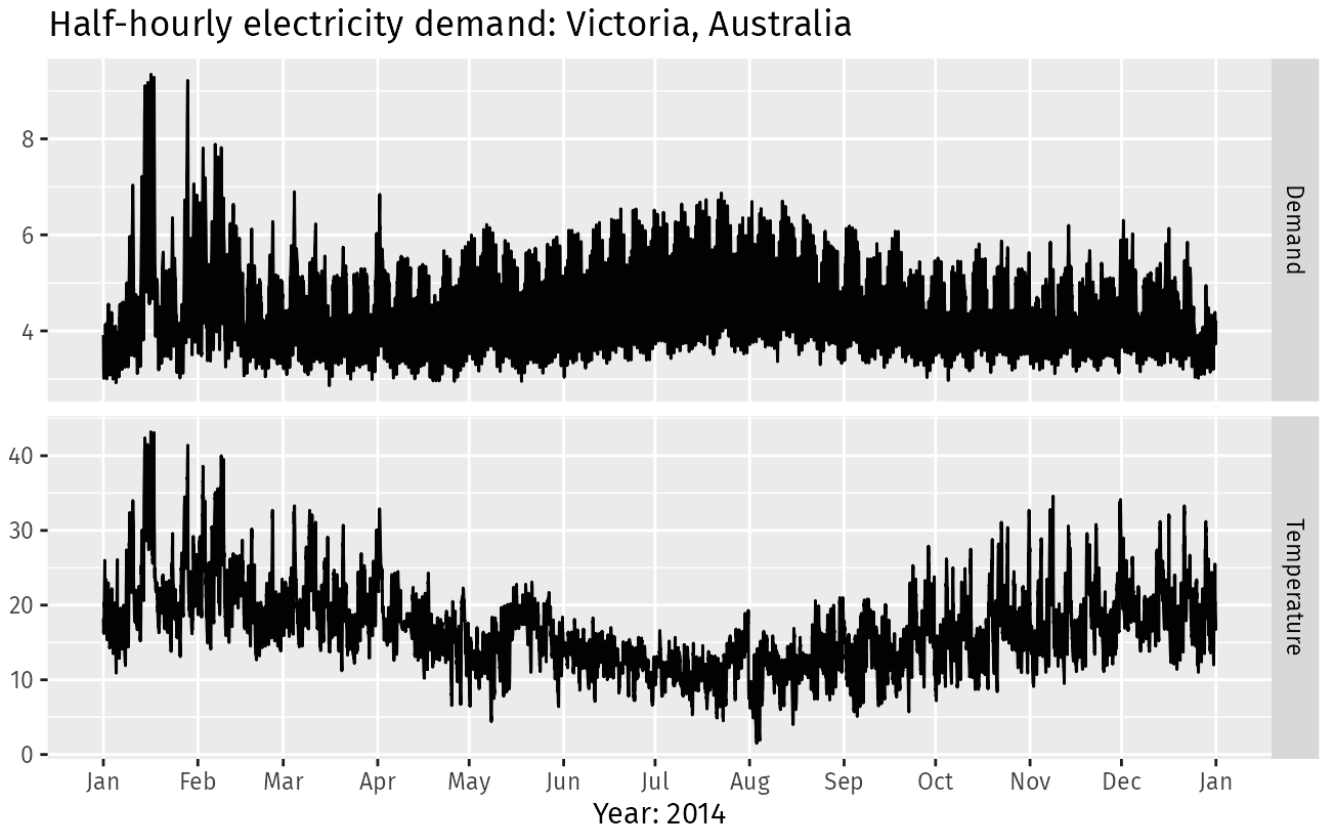


Figure 2.7: Half hourly electricity demand and temperatures in Victoria, Australia, for 2014.

(The actual code for this plot is a little more complicated than what is shown in order to include the months on the x-axis.)

We can study the relationship between demand and temperature by plotting one series against the other.

```
as.data.frame(elecdemand) |>
  ggplot(aes(x=Temperature, y=Demand)) +
  geom_point() +
  ylab("Demand (GW)") + xlab("Temperature (Celsius)")
```

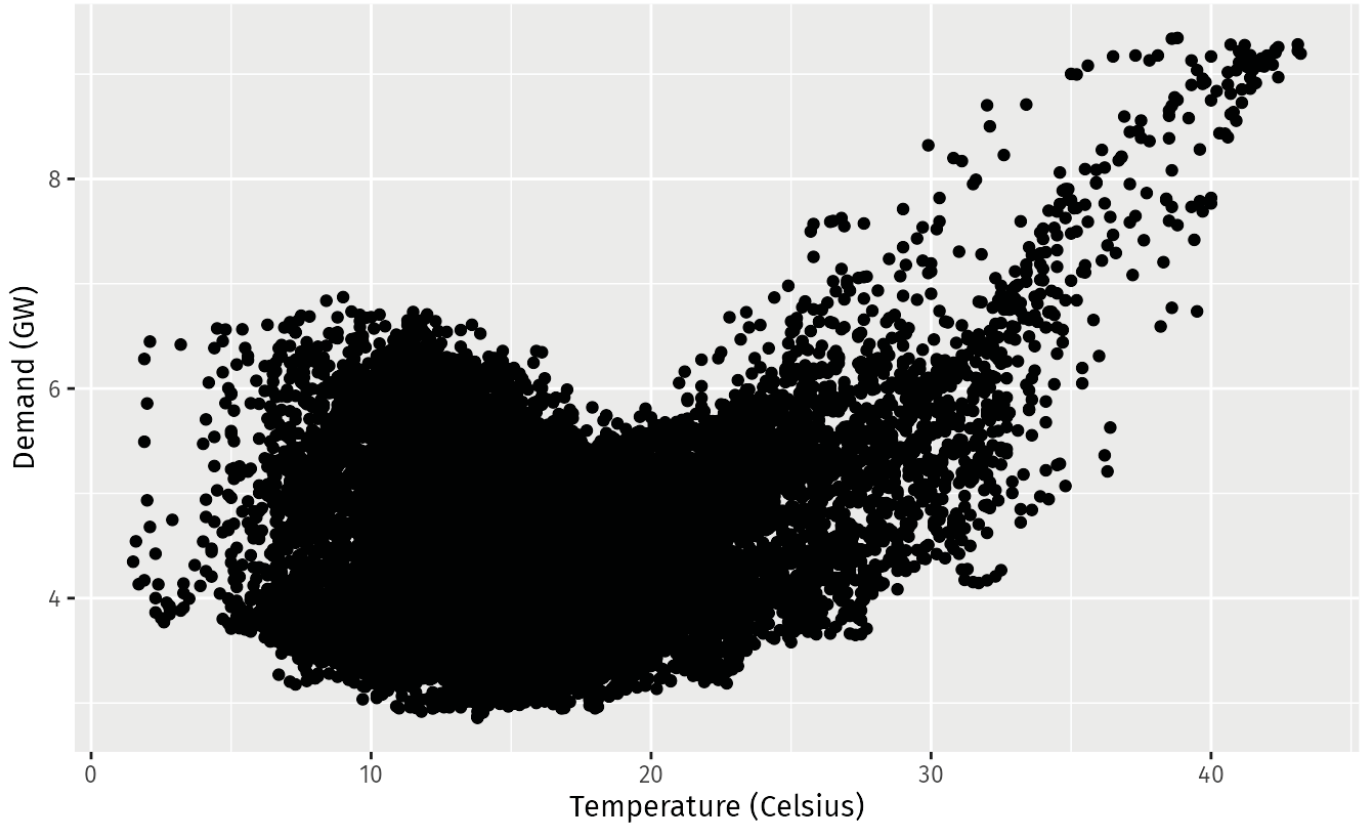


Figure 2.8: Half-hourly electricity demand plotted against temperature for 2014 in Victoria, Australia.

This scatterplot helps us to visualise the relationship between the variables. It is clear that high demand occurs when temperatures are high due to the effect of air-conditioning. But there is also a heating effect, where demand increases for very low temperatures.

Correlation

It is common to compute *correlation coefficients* to measure the strength of the relationship between two variables. The correlation between variables x and y is given by

$$r = \frac{\sum(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum(x_t - \bar{x})^2}\sqrt{\sum(y_t - \bar{y})^2}}.$$

The value of r always lies between -1 and 1 with negative values indicating a negative relationship and positive values indicating a positive relationship. The graphs in Figure 2.9 show examples of data sets with varying levels of correlation.

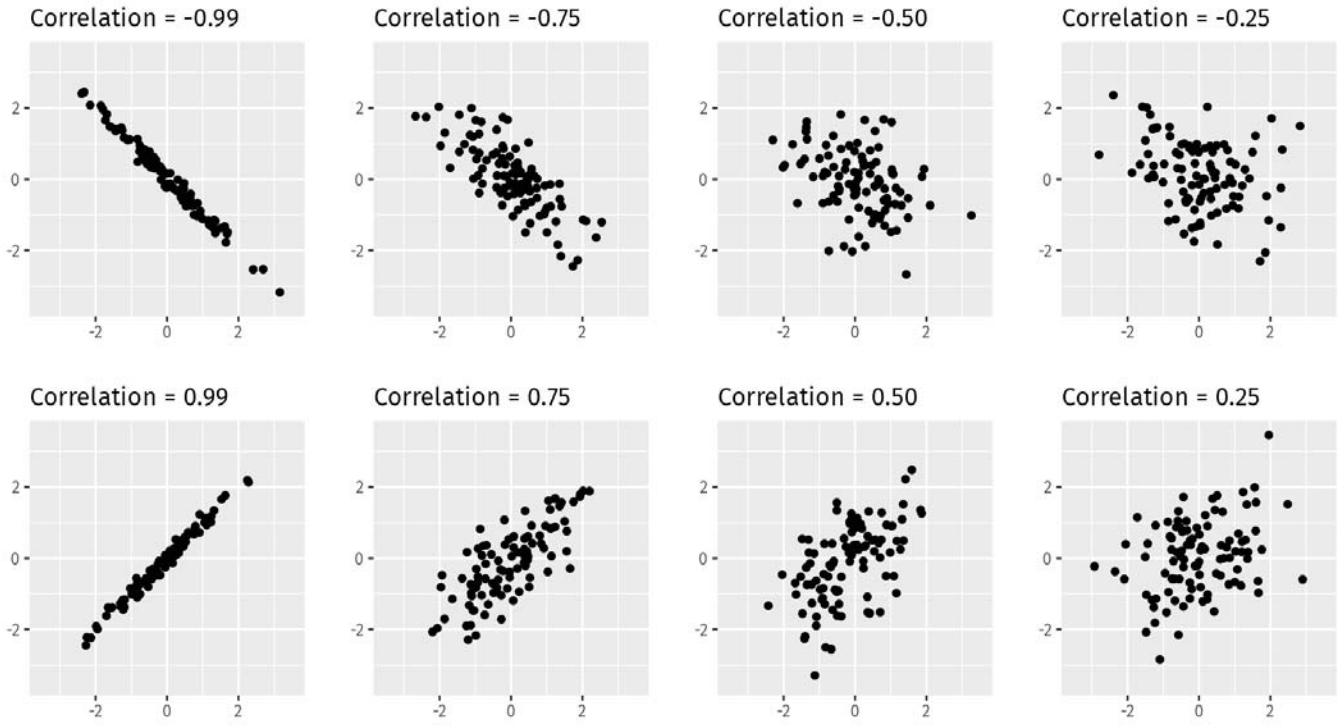


Figure 2.9: Examples of data sets with different levels of correlation.

The correlation coefficient only measures the strength of the *linear* relationship, and can sometimes be misleading. For example, the correlation for the electricity demand and temperature data shown in Figure 2.8 is 0.28, but the *non-linear* relationship is stronger than that.

The plots in Figure 2.10 all have correlation coefficients of 0.82, but they have very different relationships. This shows how important it is to look at the plots of the data and not simply rely on correlation values.

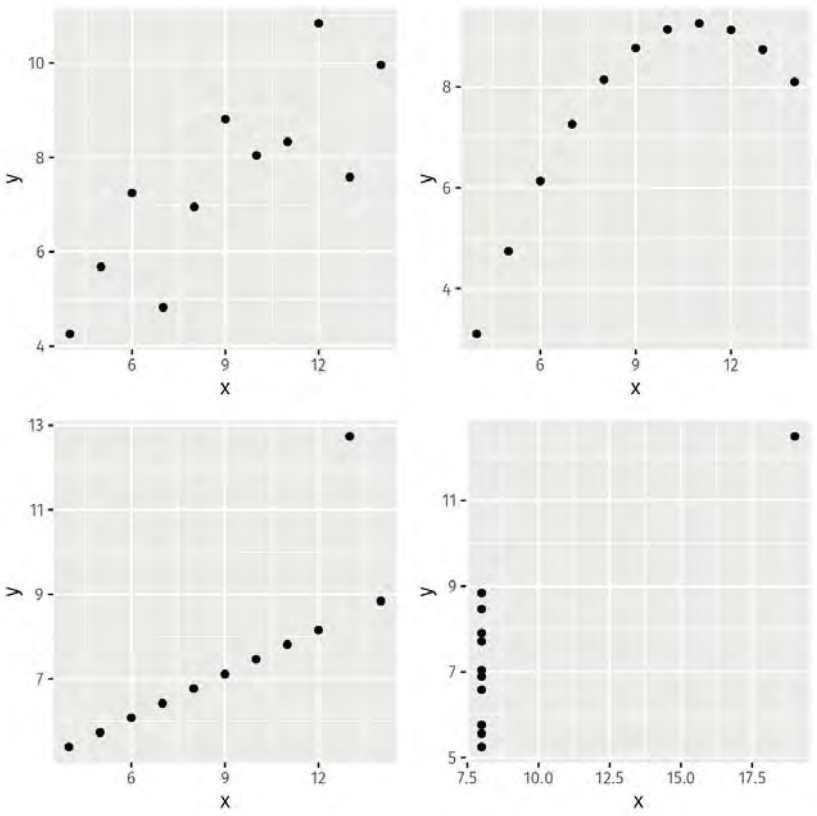


Figure 2.10: Each of these plots has a correlation coefficient of 0.82. Data from FJ Anscombe (1973) Graphs in statistical analysis. *American Statistician*, **27**, 17–21.

Scatterplot matrices

When there are several potential predictor variables, it is useful to plot each variable against each other variable. Consider the five time series shown in Figure 2.11, showing quarterly visitor numbers for five regions of New South Wales, Australia.

```
autoplot(visnights[,1:5], facets=TRUE) +
  ylab("Number of visitor nights each quarter (millions)")
```

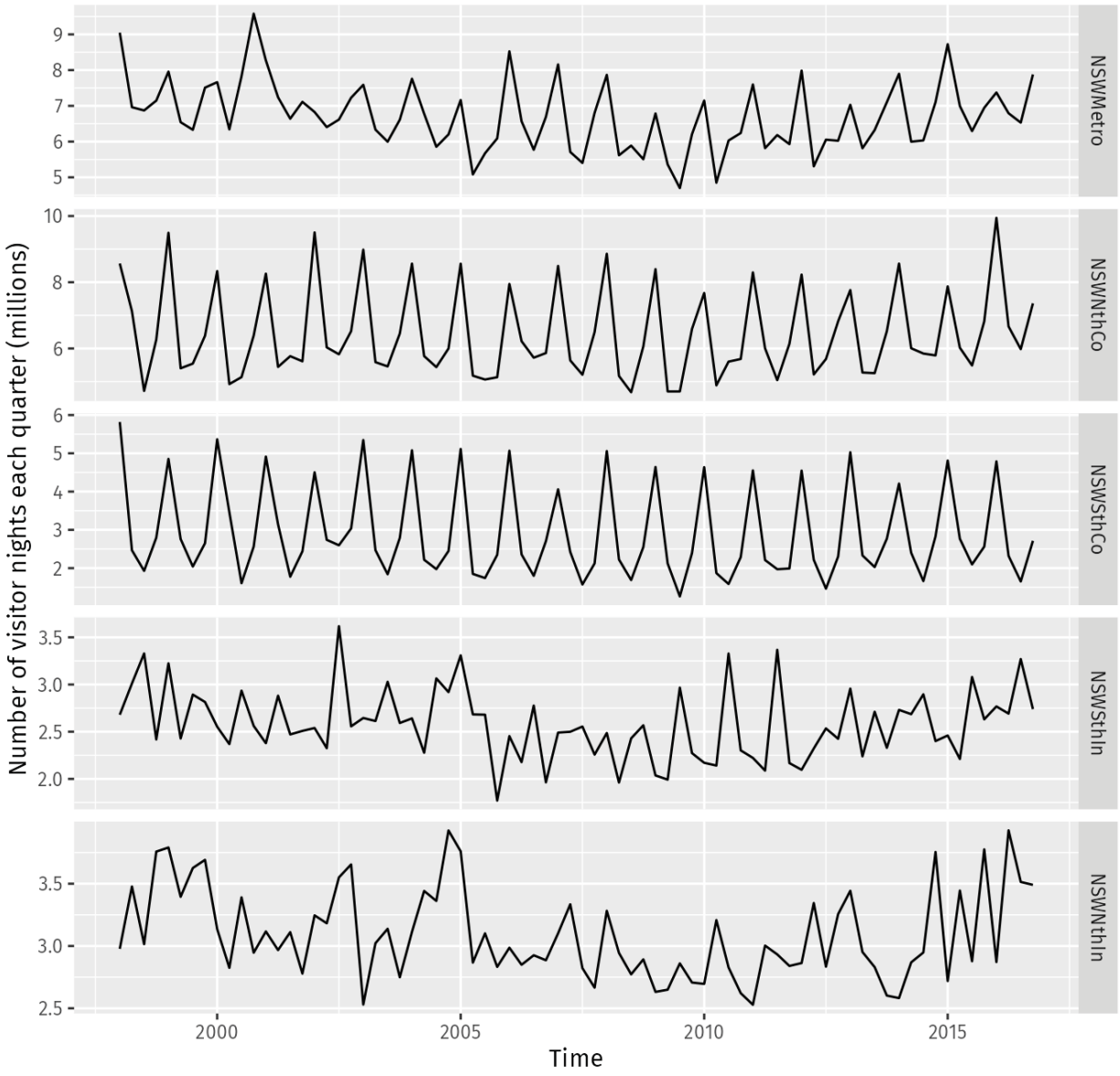


Figure 2.11: Quarterly visitor nights for various regions of NSW, Australia.

To see the relationships between these five time series, we can plot each time series against the others. These plots can be arranged in a scatterplot matrix, as shown in Figure 2.12. (This plot requires the `GGally` package to be installed.)

```
GGally::ggpairs(as.data.frame(visnights[,1:5]))
```

Figure 2.12: A scatterplot matrix of the quarterly visitor nights in five regions of NSW, Australia.

For each panel, the variable on the vertical axis is given by the variable name in that row, and the variable on the horizontal axis is given by the variable name in that column. There are many options available to produce different plots within each panel. In the default version, the correlations are shown in the upper right half of the plot, while the scatterplots are shown in the lower half. On the diagonal are shown density plots.

The value of the scatterplot matrix is that it enables a quick view of the relationships between all pairs of variables. In this example, the second column of plots shows there is a strong positive relationship between visitors to the NSW north coast and visitors to the NSW south coast, but no detectable relationship between visitors to the NSW north coast and visitors to the NSW south inland. Outliers can also be seen. There is one unusually high quarter for the NSW Metropolitan region, corresponding

to the 2000 Sydney Olympics. This is most easily seen in the first two plots in the left column of Figure 2.12, where the largest value for NSW Metro is separate from the main cloud of observations.

Here the colours indicate the quarter of the variable on the vertical axis. The lines connect points in chronological order. The relationship is strongly positive at lags 4 and 8, reflecting the strong seasonality in the data. The negative relationship seen for lags 2 and 6 occurs because peaks (in Q4) are plotted against troughs (in Q2)

The `window()` function used here is very useful when extracting a portion of a time series. In this case, we have extracted the data from `ausbeer`, beginning in 1992.

2.8 Autocorrelation

Just as correlation measures the extent of a linear relationship between two variables, autocorrelation measures the linear relationship between *lagged values* of a time series.

There are several autocorrelation coefficients, corresponding to each panel in the lag plot. For example, r_1 measures the relationship between y_t and y_{t-1}, r_2 measures the relationship between y_t and y_{t-2}, and so on.

The value of r_k can be written as

$$r_k = \frac{\sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2},$$

where T is the length of the time series.

The first nine autocorrelation coefficients for the beer production data are given in the following table.

r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8	r_9
-0.102	-0.657	-0.060	0.869	-0.089	-0.635	-0.054	0.832	-0.108

These correspond to the nine scatterplots in Figure 2.13. The autocorrelation coefficients are plotted to show the *autocorrelation function* or ACF. The plot is also known as a *correlogram*.

```
ggAcf(beer2)
```

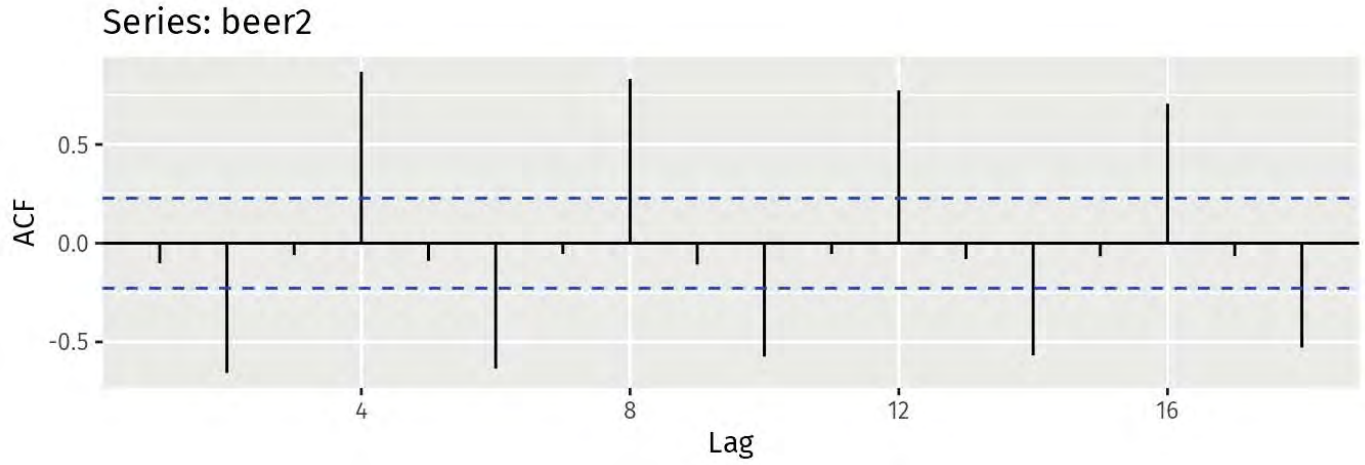


Figure 2.14: Autocorrelation function of quarterly beer production.

In this graph:

- r_4 is higher than for the other lags. This is due to the seasonal pattern in the data: the peaks tend to be four quarters apart and the troughs tend to be four quarters apart.
- r_2 is more negative than for the other lags because troughs tend to be two quarters behind peaks.
- The dashed blue lines indicate whether the correlations are significantly different from zero. These are explained in Section 2.9.

Trend and seasonality in ACF plots

When data have a trend, the autocorrelations for small lags tend to be large and positive because observations nearby in time are also nearby in size. So the ACF of trended time series tend to have positive values that slowly decrease as the lags increase.

When data are seasonal, the autocorrelations will be larger for the seasonal lags (at multiples of the seasonal frequency) than for other lags.

When data are both trended and seasonal, you see a combination of these effects. The monthly Australian electricity demand series plotted in Figure 2.15 shows both trend and seasonality. Its ACF is shown in Figure 2.16.

```
aelec <- window(elec, start=1980)
autoplot(aelec) + xlab("Year") + ylab("GWh")
```

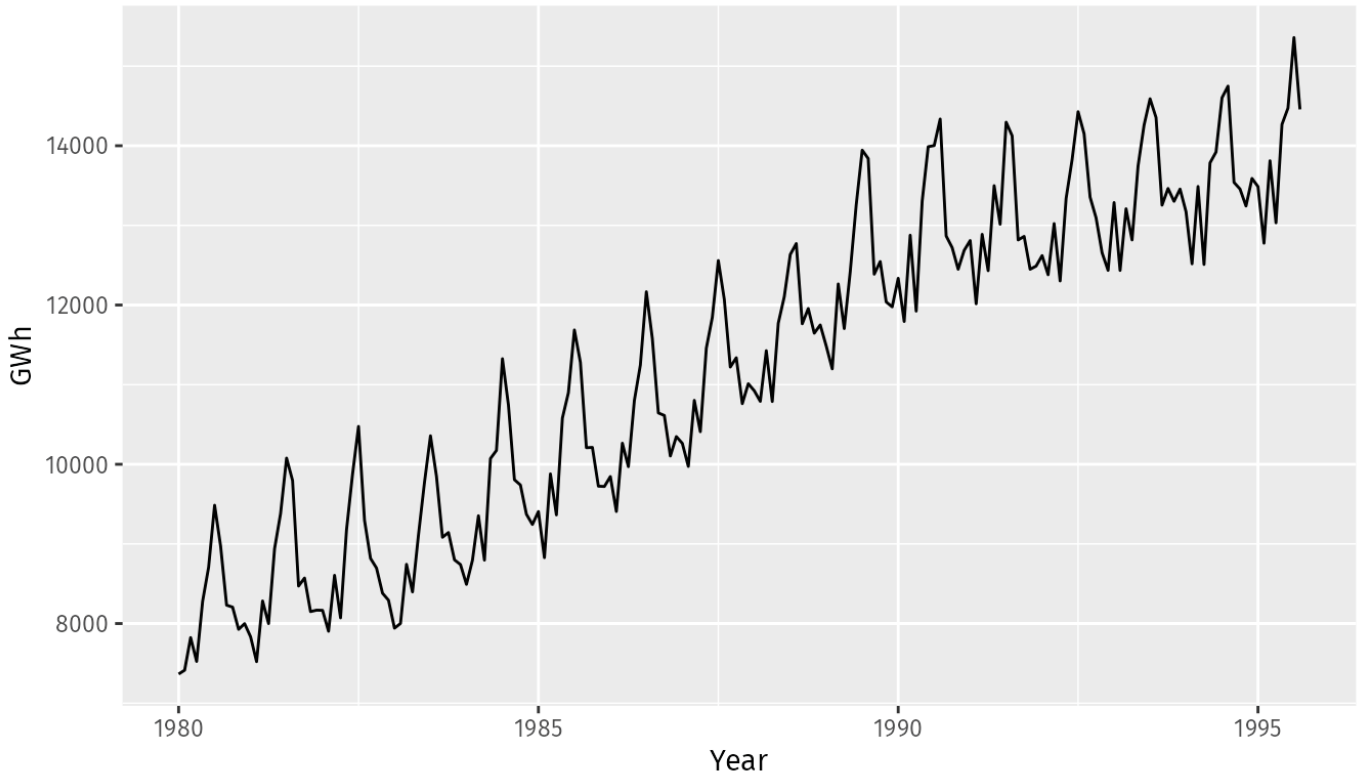


Figure 2.15: Monthly Australian electricity demand from 1980–1995.

```
ggAcf(aelec, lag=48)
```

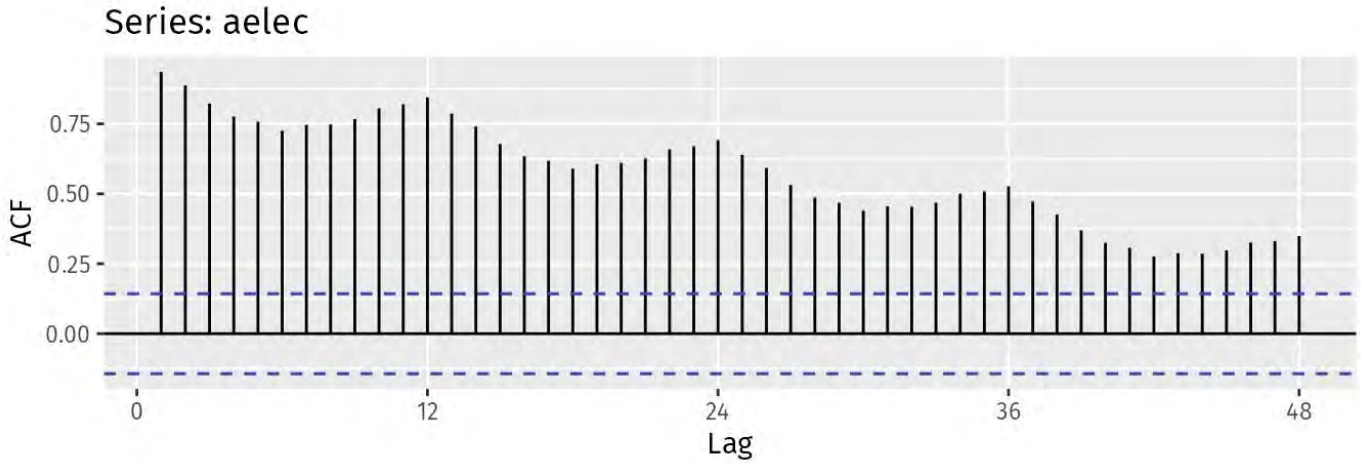


Figure 2.16: ACF of monthly Australian electricity demand.

The slow decrease in the ACF as the lags increase is due to the trend, while the "scalloped" shape is due to the seasonality.

2.9 White noise

Time series that show no autocorrelation are called **white noise**. Figure 2.17 gives an example of a white noise series.

```
set.seed(30)
y <- ts(rnorm(50))
autoplot(y) + ggtitle("White noise")
```

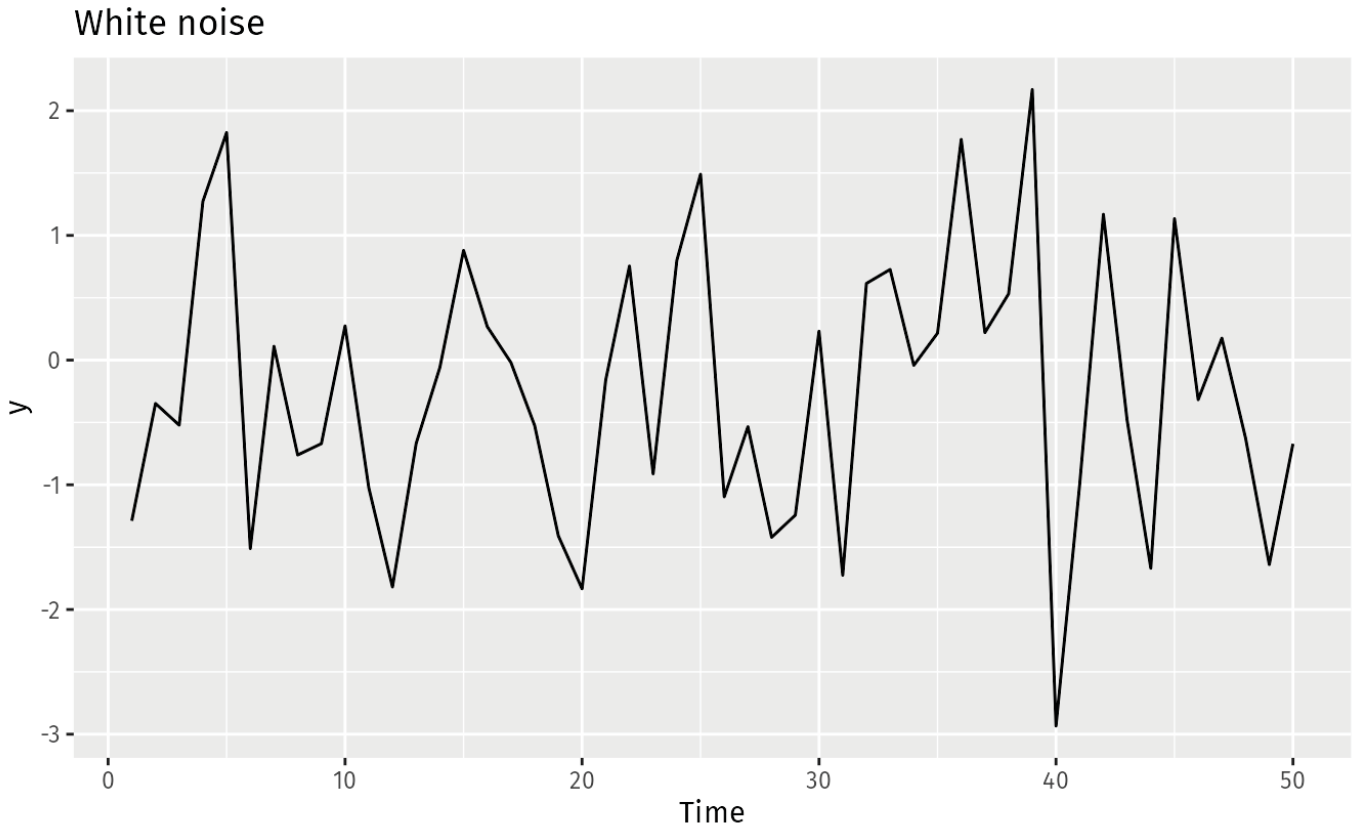


Figure 2.17: A white noise time series.

```
ggAcf(y)
```

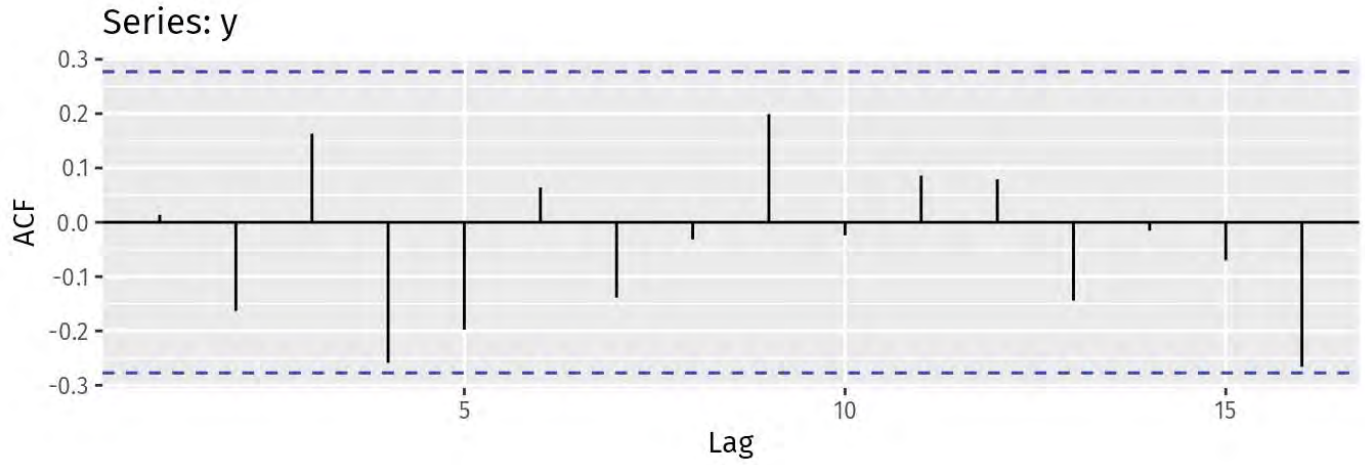


Figure 2.18: Autocorrelation function for the white noise series.

For white noise series, we expect each autocorrelation to be close to zero. Of course, they will not be exactly equal to zero as there is some random variation. For a white noise series, we expect 95% of the spikes in the ACF to lie within $\pm 2/\sqrt{T}$ where T is the length of the time series. It is common to plot these bounds on a graph of the ACF (the blue dashed lines above). If one or more large spikes are outside these bounds, or if substantially more than 5% of spikes are outside these bounds, then the series is probably not white noise.

In this example, $T = 50$ and so the bounds are at $\pm 2/\sqrt{50} = \pm 0.28$. All of the autocorrelation coefficients lie within these limits, confirming that the data are white noise.

2.10 Exercises

1. Use the help function to explore what the series `gold`, `woolyrnq` and `gas` represent.

   a. Use `autoplot()` to plot each of these in separate plots.
   b. What is the frequency of each series? Hint: apply the `frequency()` function.
   c. Use `which.max()` to spot the outlier in the `gold` series. Which observation was it?

2. Download the file `tute1.csv` from the book website, open it in Excel (or some other spreadsheet application), and review its contents. You should find four columns of information. Columns B through D each contain a quarterly series, labelled Sales, AdBudget and GDP. Sales contains the quarterly sales for a small company over the period 1981-2005. AdBudget is the advertising budget and GDP is the gross domestic product. All series have been adjusted for inflation.

   a. You can read the data into R with the following script:

   ```
   tute1 <- read.csv("tute1.csv", header=TRUE)
   View(tute1)
   ```

   b. Convert the data to time series

   ```
   mytimeseries <- ts(tute1[,-1], start=1981, frequency=4)
   ```

   (The `[,-1]` removes the first column which contains the quarters as we don't need them now.)

   c. Construct time series plots of each of the three series

   ```
   autoplot(mytimeseries, facets=TRUE)
   ```

   Check what happens when you don't include `facets=TRUE`.

3. Download some monthly Australian retail data from the book website. These represent retail sales in various categories for different Australian states, and are stored in a MS-Excel file.

   a. You can read the data into R with the following script:

```
retaildata <- readxl::read_excel("retail.xlsx", skip=1)
```

The second argument (`skip=1`) is required because the Excel sheet has two header rows.

b. Select one of the time series as follows (but replace the column name with your own chosen column):

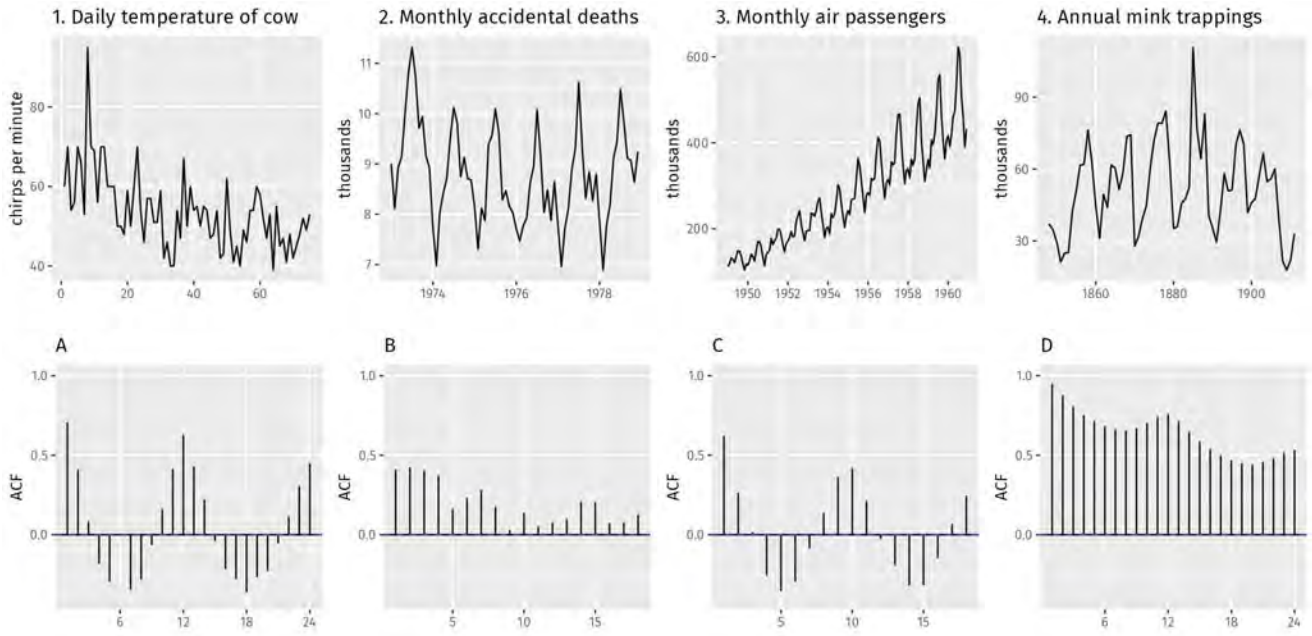
```
myts <- ts(retaildata[,"A3349873A"],
  frequency=12, start=c(1982,4))
```

c. Explore your chosen retail time series using the following functions:

`autoplot()` , `ggseasonplot()` , `ggsubseriesplot()` , `gglagplot()` , `ggAcf()`

Can you spot any seasonality, cyclicity and trend? What do you learn about the series?

4. Create time plots of the following time series: `bicoal` , `chicken` , `dole` , `usdeaths` , `lynx` , `goog` , `writing` , `fancy` , `a10` , `h02` .
   - Use `help()` to find out about the data in each series.
   - For the `goog` plot, modify the axis labels and title.
5. Use the `ggseasonplot()` and `ggsubseriesplot()` functions to explore the seasonal patterns in the following time series: `writing` , `fancy` , `a10` , `h02` .
   - What can you say about the seasonal patterns?
   - Can you identify any unusual years?
6. Use the following graphics functions: `autoplot()` , `ggseasonplot()` , `ggsubseriesplot()` , `gglagplot()` , `ggAcf()` and explore features from the following time series: `hsales` , `usdeaths` , `bricksq` , `sunspotarea` , `gasoline` .
   - Can you spot any seasonality, cyclicity and trend?
   - What do you learn about the series?
7. The `arrivals` data set comprises quarterly international arrivals (in thousands) to Australia from Japan, New Zealand, UK and the US.
   - Use `autoplot()` , `ggseasonplot()` and `ggsubseriesplot()` to compare the differences between the arrivals from these four countries.
   - Can you identify any unusual observations?
8. The following time plots and ACF plots correspond to four different time series. Your task is to match each time plot in the first row with one of the ACF plots in the second row.



9. The `pigs` data shows the monthly total number of pigs slaughtered in Victoria, Australia, from Jan 1980 to Aug 1995. Use `mypigs <- window(pigs, start=1990)` to select the data starting from 1990. Use `autoplot` and `ggAcf` for `mypigs` series and compare these to white noise plots from Figures 2.17 and 2.18.

10. `dj` contains 292 consecutive trading days of the Dow Jones Index. Use `ddj <- diff(dj)` to compute the daily changes in the index. Plot `ddj` and its ACF. Do the changes in the Dow Jones Index look like white noise?

2.11 Further reading

- W. S. Cleveland (1993) is a classic book on the principles of visualisation for data analysis. While it is more than 20 years old, the ideas are timeless.
- Unwin (2015) is a modern introduction to graphical data analysis using R. It does not have much information on time series graphics, but plenty of excellent general advice on using graphics for data analysis.

Bibliography

Cleveland, W. S. (1993). *Visualizing data*. Hobart Press. [Amazon]

Unwin, A. (2015). *Graphical data analysis with R*. Chapman; Hall/CRC. [Amazon]

Chapter 3 The forecaster's toolbox

In this chapter, we discuss some general tools that are useful for many different forecasting situations. We will describe some benchmark forecasting methods, ways of making the forecasting task simpler using transformations and adjustments, methods for checking whether a forecasting method has adequately utilised the available information, and techniques for computing prediction intervals.

Each of the tools discussed in this chapter will be used repeatedly in subsequent chapters as we develop and explore a range of forecasting methods.

3.1 Some simple forecasting methods

Some forecasting methods are extremely simple and surprisingly effective. We will use the following four forecasting methods as benchmarks throughout this book.

Average method

Here, the forecasts of all future values are equal to the average (or "mean") of the historical data. If we let the historical data be denoted by $y_1, \dots, y_T$, then we can write the forecasts as

$$\hat{y}_{T+h|T} = \bar{y} = (y_1 + \cdots + y_T)/T.$$

The notation $\hat{y}_{T+h|T}$ is a short-hand for the estimate of y_{T+h} based on the data $y_1, \dots, y_T$.

```
meanf(y, h)
# y contains the time series
# h is the forecast horizon
```

Naïve method

For naïve forecasts, we simply set all forecasts to be the value of the last observation. That is,

$$\hat{y}_{T+h|T} = y_T.$$

This method works remarkably well for many economic and financial time series.

```
naive(y, h)
rwf(y, h) # Equivalent alternative
```

Because a naïve forecast is optimal when data follow a random walk (see Section 8.1), these are also called **random walk forecasts**.

Seasonal naïve method

A similar method is useful for highly seasonal data. In this case, we set each forecast to be equal to the last observed value from the same season (e.g., the same month of the previous year). Formally, the forecast for time $T+h$ is written as

$$\hat{y}_{T+h|T} = y_{T+h-m(k+1)},$$

where $m=$ the seasonal period, and k is the integer part of $(h-1)/m$ (i.e., the number of complete years in the forecast period prior to time $T+h$). This looks more complicated than it really is. For example, with monthly data, the forecast for all future February values is equal to the last observed February value. With quarterly data, the forecast of all future Q2 values is equal to the last observed Q2 value (where Q2 means the second quarter). Similar rules apply for other months and quarters, and for other seasonal periods.

```
snaive(y, h)
```

Drift method

A variation on the naïve method is to allow the forecasts to increase or decrease over time, where the amount of change over time (called the **drift**) is set to be the average change seen in the historical data. Thus the forecast for time $T+h$ is given by

$$\hat{y}_{T+h|T} = y_T + \frac{h}{T-1}\sum_{t=2}^{T}(y_t - y_{t-1}) = y_T + h\left(\frac{y_T - y_1}{T-1}\right).$$

This is equivalent to drawing a line between the first and last observations, and extrapolating it into the future.

```
rwf(y, h, drift=TRUE)
```

Examples

Figure 3.1 shows the first three methods applied to the quarterly beer production data.

```
# Set training data from 1992 to 2007
beer2 <- window(ausbeer,start=1992,end=c(2007,4))
# Plot some forecasts
autoplot(beer2) +
  autolayer(meanf(beer2, h=11),
    series="Mean", PI=FALSE) +
  autolayer(naive(beer2, h=11),
    series="Naïve", PI=FALSE) +
  autolayer(snaive(beer2, h=11),
    series="Seasonal naïve", PI=FALSE) +
  ggtitle("Forecasts for quarterly beer production") +
  xlab("Year") + ylab("Megalitres") +
  guides(colour=guide_legend(title="Forecast"))
```

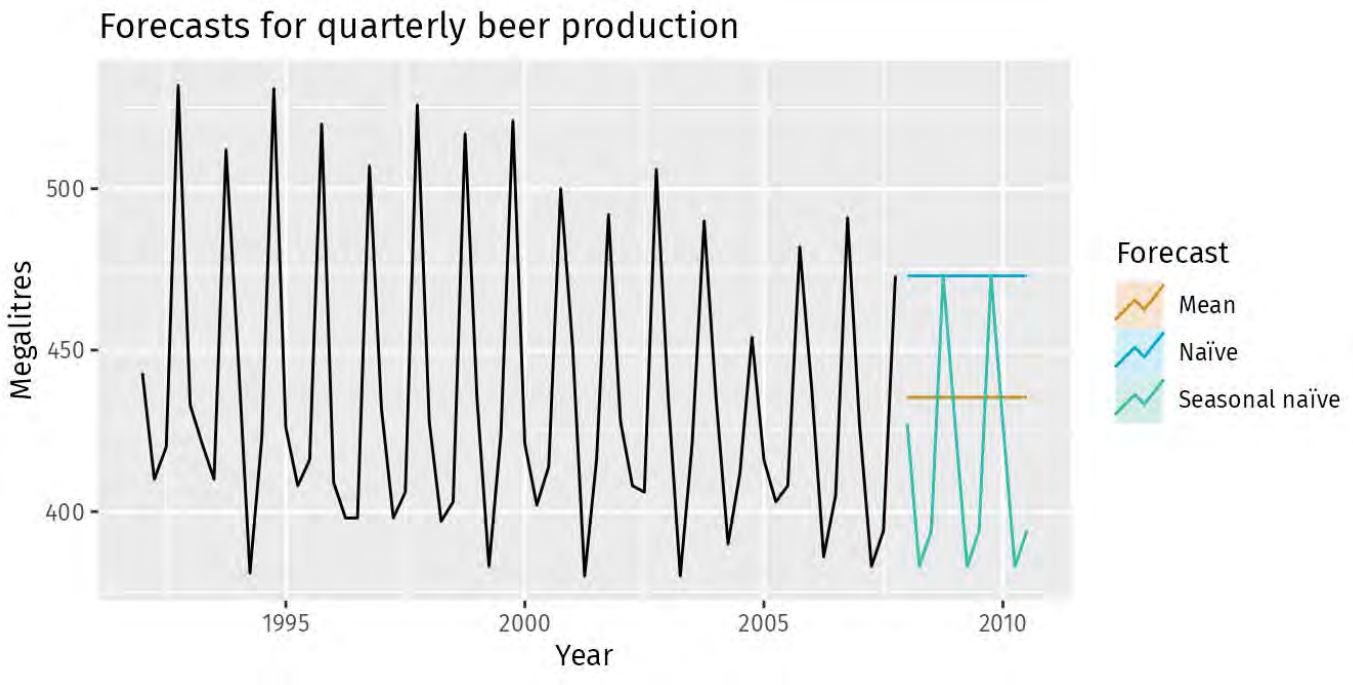


Figure 3.1: Forecasts of Australian quarterly beer production.

In Figure 3.2, the non-seasonal methods are applied to a series of 200 days of the Google daily closing stock price.

```
autoplot(goog200) +
  autolayer(meanf(goog200, h=40),
    series="Mean", PI=FALSE) +
  autolayer(rwf(goog200, h=40),
    series="Naïve", PI=FALSE) +
  autolayer(rwf(goog200, drift=TRUE, h=40),
    series="Drift", PI=FALSE) +
  ggtitle("Google stock (daily ending 6 Dec 2013)") +
  xlab("Day") + ylab("Closing Price (US$)") +
  guides(colour=guide_legend(title="Forecast"))
```

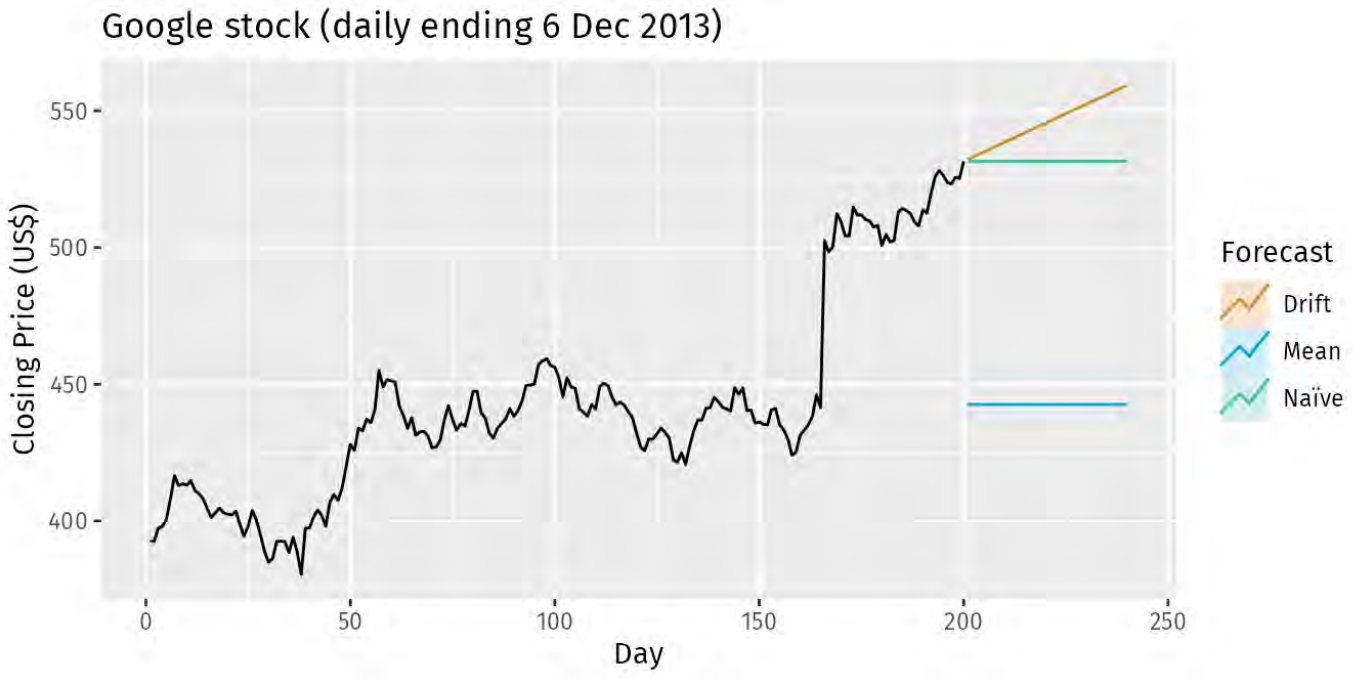


Figure 3.2: Forecasts based on 200 days of the Google daily closing stock price.

Sometimes one of these simple methods will be the best forecasting method available; but in many cases, these methods will serve as benchmarks rather than the method of choice. That is, any forecasting methods we develop will be compared to these simple methods to ensure that the new method is better than these simple alternatives. If not, the new method is not worth considering.

3.2 Transformations and adjustments

Adjusting the historical data can often lead to a simpler forecasting task. Here, we deal with four kinds of adjustments: calendar adjustments, population adjustments, inflation adjustments and mathematical transformations. The purpose of these adjustments and transformations is to simplify the patterns in the historical data by removing known sources of variation or by making the pattern more consistent across the whole data set. Simpler patterns usually lead to more accurate forecasts.

Calendar adjustments

Some of the variation seen in seasonal data may be due to simple calendar effects. In such cases, it is usually much easier to remove the variation before fitting a forecasting model. The `monthdays()` function will compute the number of days in each month or quarter.

For example, if you are studying the monthly milk production on a farm, there will be variation between the months simply because of the different numbers of days in each month, in addition to the seasonal variation across the year.

```
dframe <- cbind(Monthly = milk,
                DailyAverage = milk/monthdays(milk))
  autoplot(dframe, facet=TRUE) +
    xlab("Years") + ylab("Pounds") +
    ggtitle("Milk production per cow")
```

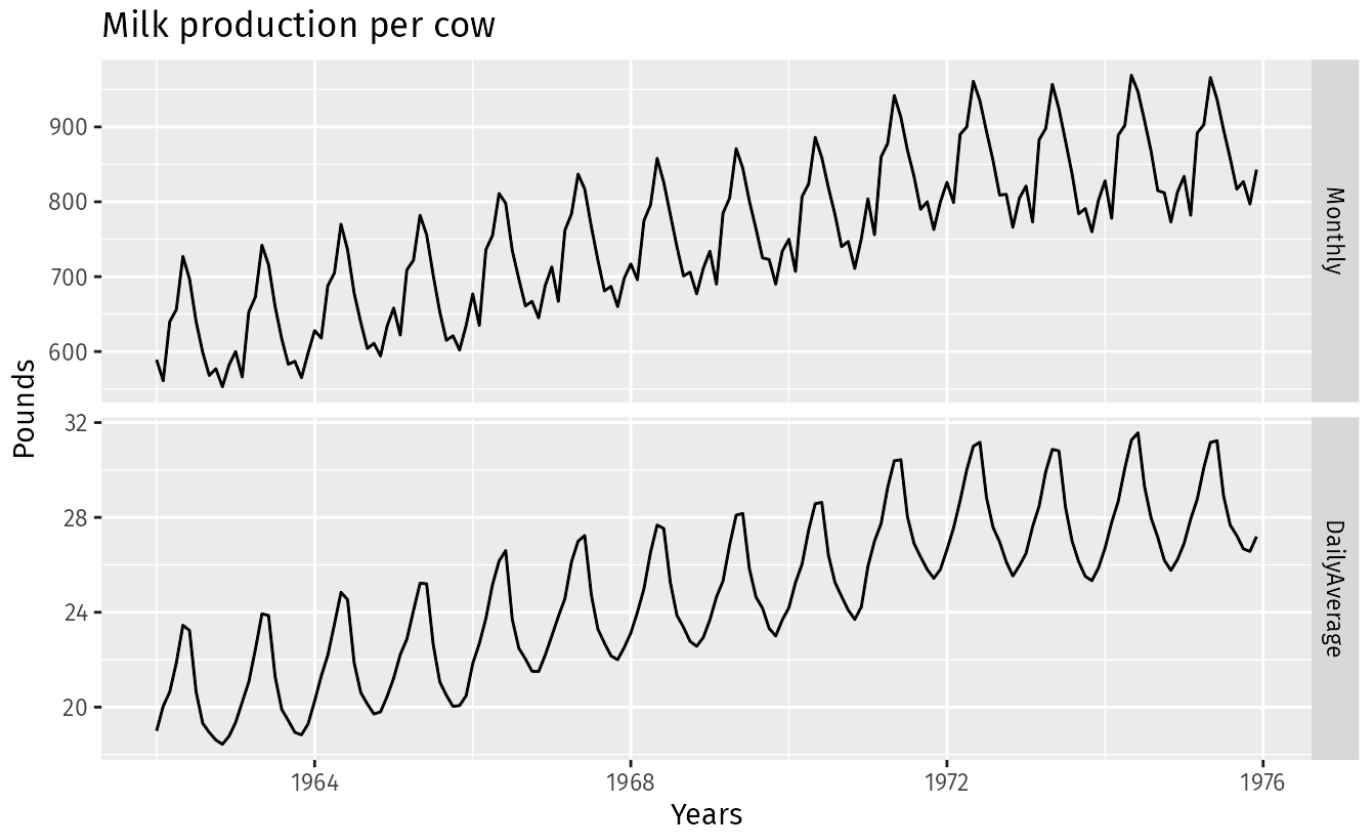


Figure 3.3: Monthly milk production per cow.

Notice how much simpler the seasonal pattern is in the average daily production plot compared to the total monthly production plot. By looking at the average daily production instead of the total monthly production, we effectively remove the variation due to the different month lengths. Simpler patterns are usually easier to model and lead to more accurate forecasts.

A similar adjustment can be done for sales data when the number of trading days in each month varies. In this case, the sales per trading day can be modelled instead of the total sales for each month.

Population adjustments

Any data that are affected by population changes can be adjusted to give per-capita data. That is, consider the data per person (or per thousand people, or per million people) rather than the total. For example, if you are studying the number of hospital beds in a particular region over time, the results are much easier to interpret if you remove the effects of population changes by considering the number of beds per thousand people. Then you can see whether there have been real increases in the number of beds, or whether the increases are due entirely to population increases. It is possible for the total number of beds to increase, but the number of beds per

thousand people to decrease. This occurs when the population is increasing faster than the number of hospital beds. For most data that are affected by population changes, it is best to use per-capita data rather than the totals.

Inflation adjustments

Data which are affected by the value of money are best adjusted before modelling. For example, the average cost of a new house will have increased over the last few decades due to inflation. A $200,000 house this year is not the same as a $200,000 house twenty years ago. For this reason, financial time series are usually adjusted so that all values are stated in dollar values from a particular year. For example, the house price data may be stated in year 2000 dollars.

To make these adjustments, a price index is used. If z_t denotes the price index and y_t denotes the original house price in year t, then $x_t = y_t/z_t * z_{2000}$ gives the adjusted house price at year 2000 dollar values. Price indexes are often constructed by government agencies. For consumer goods, a common price index is the Consumer Price Index (or CPI).

Mathematical transformations

If the data show variation that increases or decreases with the level of the series, then a transformation can be useful. For example, a logarithmic transformation is often useful. If we denote the original observations as $y_1, \dots, y_T$ and the transformed observations as $w_1, \dots, w_T$, then $w_t = \log(y_t)$. Logarithms are useful because they are interpretable: changes in a log value are relative (or percentage) changes on the original scale. So if log base 10 is used, then an increase of 1 on the log scale corresponds to a multiplication of 10 on the original scale. Another useful feature of log transformations is that they constrain the forecasts to stay positive on the original scale.

Sometimes other transformations are also used (although they are not so interpretable). For example, square roots and cube roots can be used. These are called **power transformations** because they can be written in the form $w_t = y_t^p$.

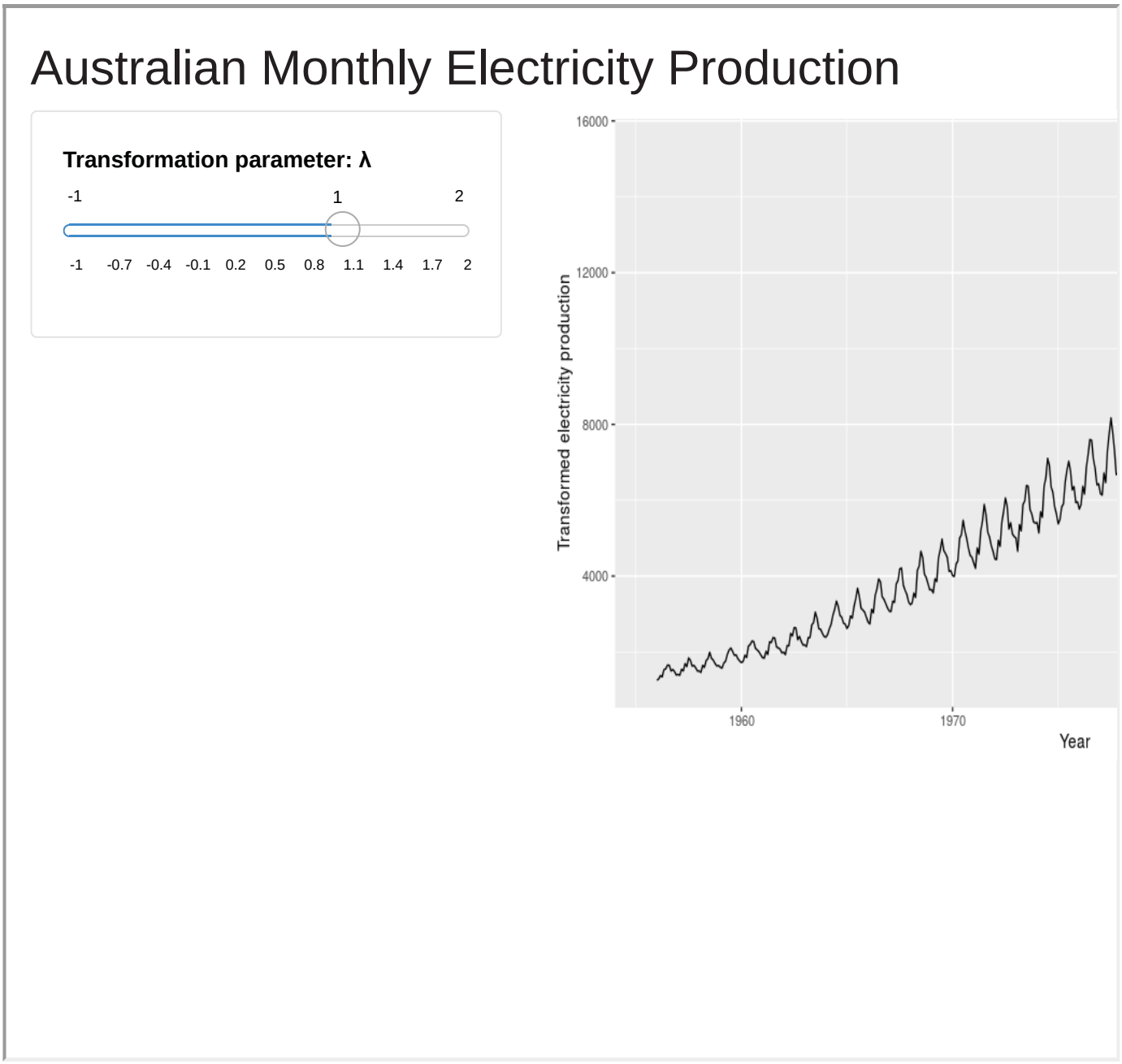
A useful family of transformations, that includes both logarithms and power transformations, is the family of **Box-Cox transformations** (Box & Cox, 1964), which depend on the parameter λ and are defined as follows:

$$w_t = \begin{cases} \log(y_t) & \text{if } \lambda = 0; \\ (y_t^\lambda - 1)/\lambda & \text{otherwise.} \end{cases}$$

The logarithm in a Box-Cox transformation is always a natural logarithm (i.e., to base e). So if $\lambda = 0$, natural logarithms are used, but if $\lambda \neq 0$, a power transformation is used, followed by some simple scaling.

If $\lambda = 1$, then $w_t = y_t - 1$, so the transformed data is shifted downwards but there is no change in the shape of the time series. But for all other values of λ, the time series will change shape.

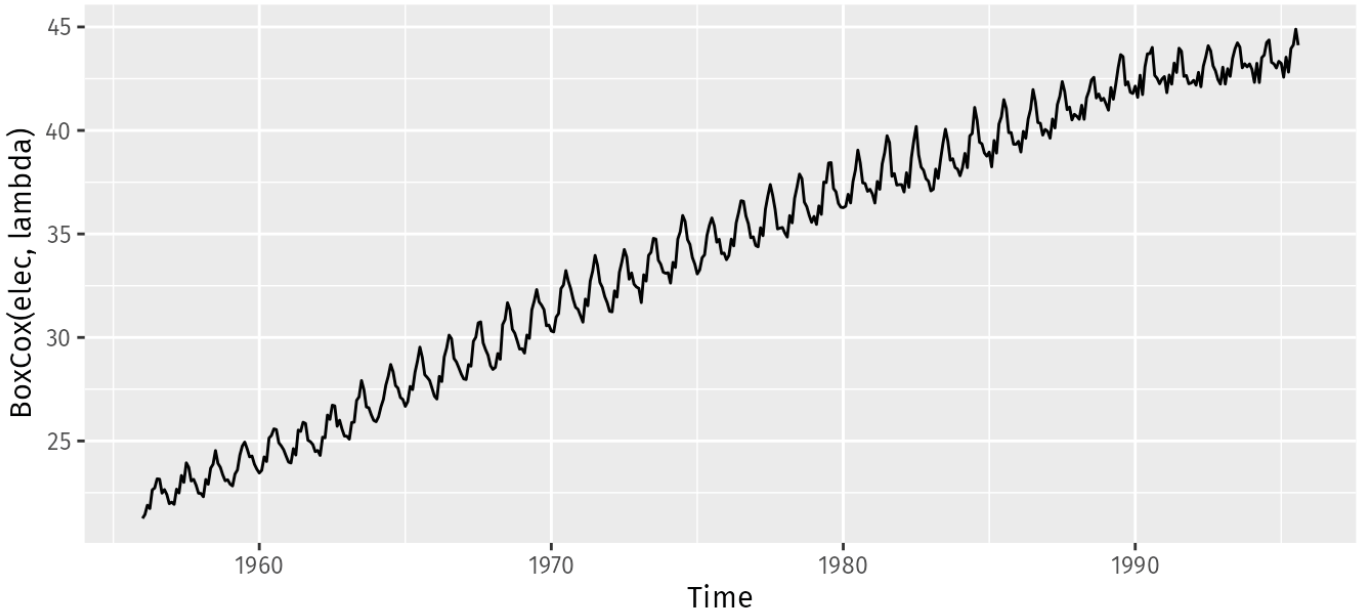
Use the slider below to see the effect of varying λ to transform Australian monthly electricity production:



A good value of λ is one which makes the size of the seasonal variation about the same across the whole series, as that makes the forecasting model simpler. In this case, $\lambda = 0.30$ works quite well, although any value of λ between 0 and 0.5 would give similar results.

The `BoxCox.lambda()` function will choose a value of lambda for you.

```
(lambda <- BoxCox.lambda(elec))
#> [1] 0.2654
autoplot(BoxCox(elec,lambda))
```



The `BoxCox()` command actually implements a slight modification of the Box-Cox transformation, discussed in Bickel & Doksum (1981), which allows for negative values of y_t when $\lambda > 0$:

$$w_t = \begin{cases} \log(y_t) & \text{if } \lambda = 0; \\ \text{sign}(y_t)(|y_t|^\lambda - 1)/\lambda & \text{otherwise.} \end{cases}$$

For positive values of y_t, this is the same as the original Box-Cox transformation.

Having chosen a transformation, we need to forecast the transformed data. Then, we need to reverse the transformation (or *back-transform*) to obtain forecasts on the original scale. The reverse Box-Cox transformation is given by

$$y_t = \begin{cases} \exp(w_t) & \text{if } \lambda = 0; \\ \text{sign}(\lambda w_t + 1)|\lambda w_t + 1|^{1/\lambda} & \text{otherwise.} \end{cases} \tag{3.1}$$

Features of power transformations

- Choose a simple value of λ. It makes explanations easier.
- The forecasting results are relatively insensitive to the value of λ.
- Often no transformation is needed.
- Transformations sometimes make little difference to the forecasts but have a large effect on prediction intervals.

Bias adjustments

One issue with using mathematical transformations such as Box-Cox transformations is that the back-transformed point forecast will not be the mean of the forecast distribution. In fact, it will usually be the median of the forecast distribution (assuming that the distribution on the transformed space is symmetric). For many purposes, this is acceptable, but occasionally the mean forecast is required. For example, you may wish to add up sales forecasts from various regions to form a forecast for the whole country. But medians do not add up, whereas means do.

For a Box-Cox transformation, the back-transformed mean is given by

$$y_t = \begin{cases} \exp(w_t)\left[1+\frac{\sigma_h^2}{2}\right] & \text{if } \lambda=0; \\ (\lambda w_t+1)^{1/\lambda}\left[1+\frac{\sigma_h^2(1-\lambda)}{2(\lambda w_t+1)^2}\right] & \text{otherwise;} \end{cases} \tag{3.2}$$

where σ_h^2 is the h-step forecast variance. The larger the forecast variance, the bigger the difference between the mean and the median.

The difference between the simple back-transformed forecast given by (3.1) and the mean given by (3.2) is called the **bias**. When we use the mean, rather than the median, we say the point forecasts have been **bias-adjusted**.

To see how much difference this bias-adjustment makes, consider the following example, where we forecast average annual price of eggs using the drift method with a log transformation ($\lambda=0$). The log transformation is useful in this case to ensure the forecasts and the prediction intervals stay positive.

```
fc <- rwf(eggs, drift=TRUE, lambda=0, h=50, level=80)
fc2 <- rwf(eggs, drift=TRUE, lambda=0, h=50, level=80,
  biasadj=TRUE)
autoplot(eggs) +
  autolayer(fc, series="Simple back transformation") +
  autolayer(fc2, series="Bias adjusted", PI=FALSE) +
  guides(colour=guide_legend(title="Forecast"))
```

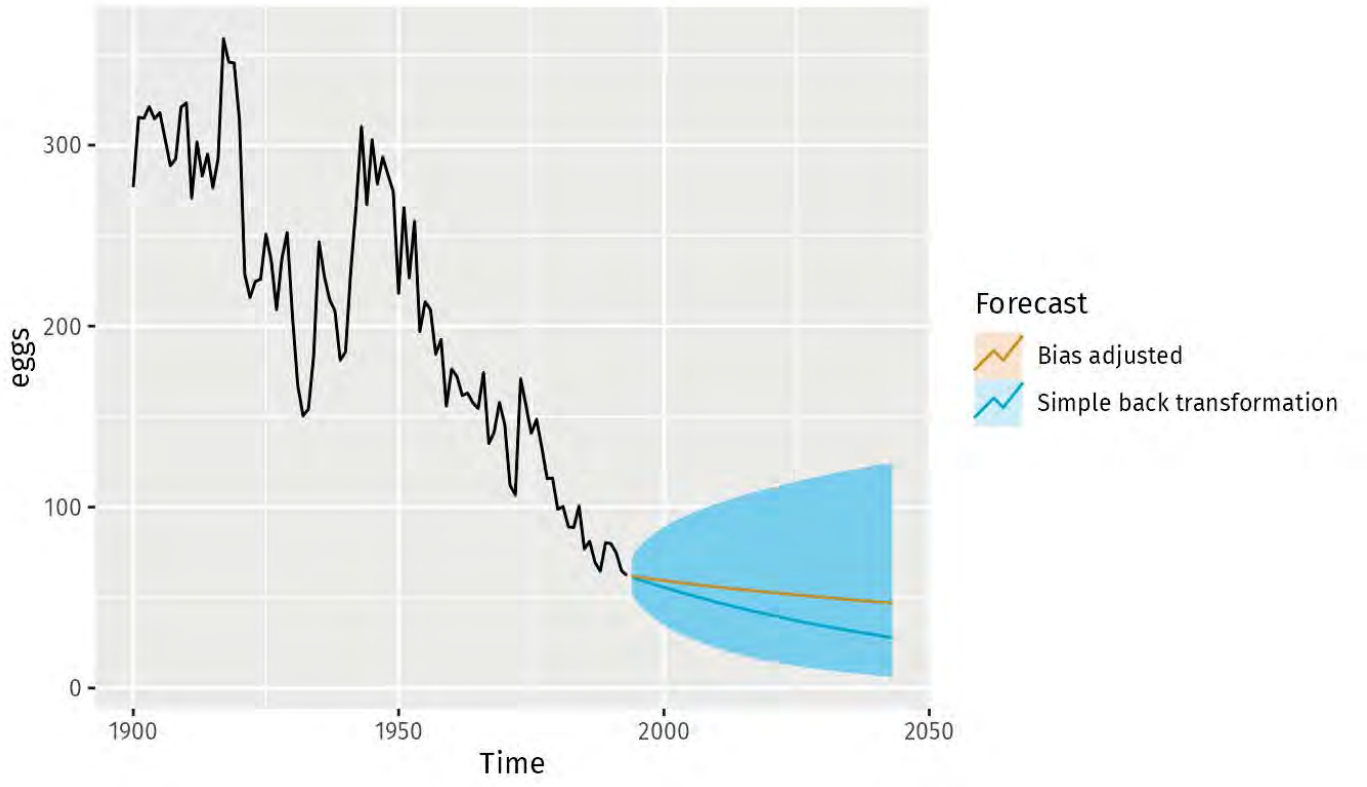


Figure 3.4: Forecasts of egg prices using a random walk with drift applied to the logged data.

The blue line in Figure 3.4 shows the forecast medians while the red line shows the forecast means. Notice how the skewed forecast distribution pulls up the point forecast when we use the bias adjustment.

Bias adjustment is not done by default in the **forecast** package. If you want your forecasts to be means rather than medians, use the argument `biasadj=TRUE` when you select your Box-Cox transformation parameter.

Bibliography

Bickel, P. J., & Doksum, K. A. (1981). An analysis of transformations revisited. *Journal of the American Statistical Association*, *76*(374), 296–311.

Box, G. E. P., & Cox, D. R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, *26*(2), 211–252. [DOI]

3.3 Residual diagnostics

Fitted values

Each observation in a time series can be forecast using all previous observations. We call these **fitted values** and they are denoted by $\hat{y}_{t|t-1}$, meaning the forecast of y_t based on observations $y_1, \dots, y_{t-1}$. We use these so often, we sometimes drop part of the subscript and just write $\hat{y}_t$ instead of $\hat{y}_{t|t-1}$. Fitted values always involve one-step forecasts.

Actually, fitted values are often not true forecasts because any parameters involved in the forecasting method are estimated using all available observations in the time series, including future observations. For example, if we use the average method, the fitted values are given by

$$\hat{y}_t = \hat{c}$$

where $\hat{c}$ is the average computed over all available observations, including those at times *after* t. Similarly, for the drift method, the drift parameter is estimated using all available observations. In this case, the fitted values are given by

$$\hat{y}_t = y_{t-1} + \hat{c}$$

where $\hat{c} = (y_T - y_1)/(T-1)$. In both cases, there is a parameter to be estimated from the data. The "hat" above the c reminds us that this is an estimate. When the estimate of c involves observations after time t, the fitted values are not true forecasts. On the other hand, naïve or seasonal naïve forecasts do not involve any parameters, and so fitted values are true forecasts in such cases.

Residuals

The "residuals" in a time series model are what is left over after fitting a model. For many (but not all) time series models, the residuals are equal to the difference between the observations and the corresponding fitted values:

$$e_t = y_t - \hat{y}_t.$$

Residuals are useful in checking whether a model has adequately captured the information in the data. A good forecasting method will yield residuals with the following properties:

1. The residuals are uncorrelated. If there are correlations between residuals, then there is information left in the residuals which should be used in computing forecasts.
2. The residuals have zero mean. If the residuals have a mean other than zero, then the forecasts are biased.

Any forecasting method that does not satisfy these properties can be improved. However, that does not mean that forecasting methods that satisfy these properties cannot be improved. It is possible to have several different forecasting methods for the same data set, all of which satisfy these properties. Checking these properties is important in order to see whether a method is using all of the available information, but it is not a good way to select a forecasting method.

If either of these properties is not satisfied, then the forecasting method can be modified to give better forecasts. Adjusting for bias is easy: if the residuals have mean m, then simply add m to all forecasts and the bias problem is solved. Fixing the correlation problem is harder, and we will not address it until Chapter 9.

In addition to these essential properties, it is useful (but not necessary) for the residuals to also have the following two properties.

3. The residuals have constant variance.
4. The residuals are normally distributed.

These two properties make the calculation of prediction intervals easier (see Section 3.5 for an example). However, a forecasting method that does not satisfy these properties cannot necessarily be improved. Sometimes applying a Box-Cox transformation may assist with these properties, but otherwise there is usually little that you can do to ensure that your residuals have constant variance and a normal distribution. Instead, an alternative approach to obtaining prediction intervals is necessary. Again, we will not address how to do this until later in the book.

Example: Forecasting the Google daily closing stock price

For stock market prices and indexes, the best forecasting method is often the naïve method. That is, each forecast is simply equal to the last observed value, or $\hat{y}_t = y_{t-1}$. Hence, the residuals are simply equal to the difference between consecutive

observations:

$$e_t = y_t - \hat{y}_t = y_t - y_{t-1}.$$

The following graph shows the Google daily closing stock price (GOOG). The large jump at day 166 corresponds to 18 October 2013 when the price jumped 12% due to unexpectedly strong third quarter results.

```
autoplot(goog200) +
  xlab("Day") + ylab("Closing Price (US$)") +
  ggtitle("Google Stock (daily ending 6 December 2013)")
```

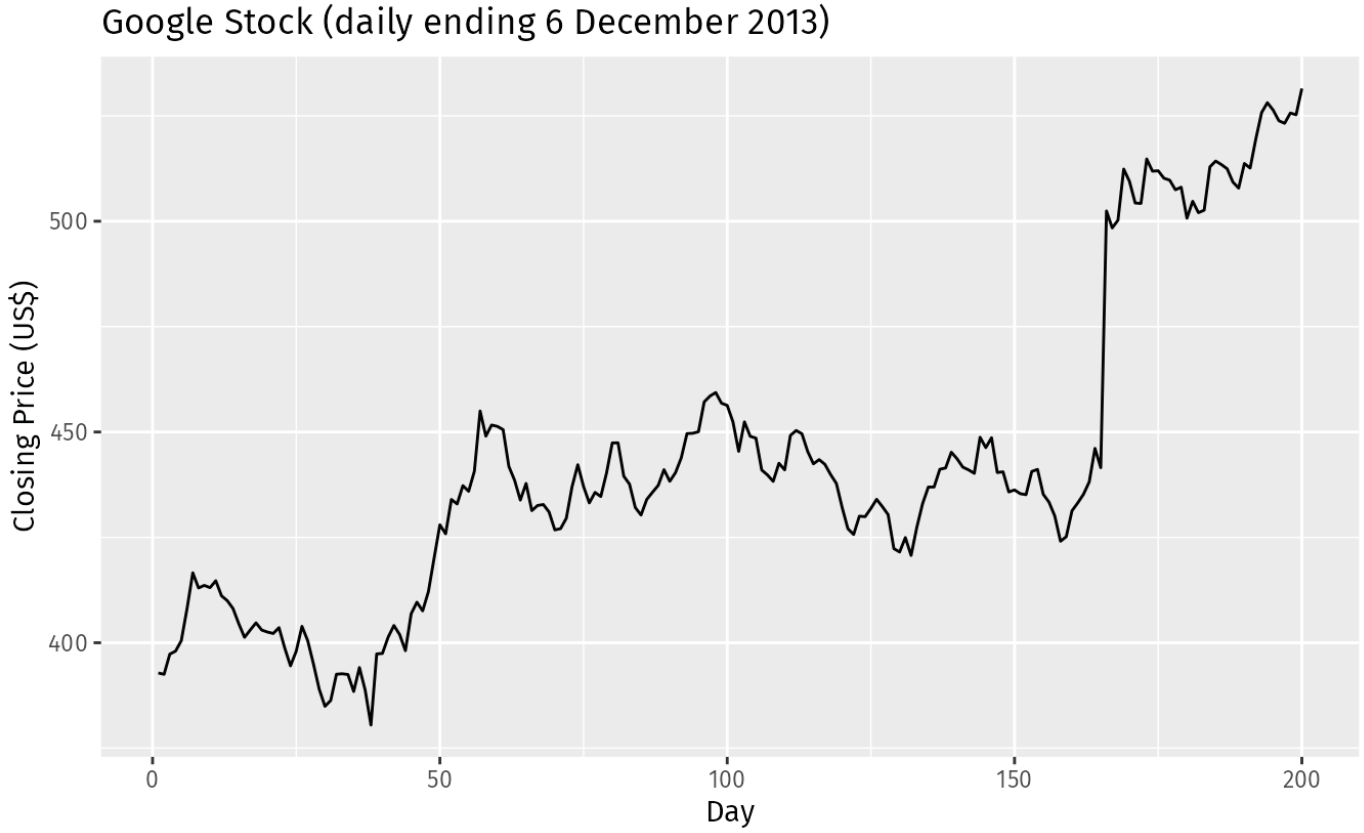


Figure 3.5: The daily Google stock price to 6 Dec 2013.

The residuals obtained from forecasting this series using the naïve method are shown in Figure 3.6. The large positive residual is a result of the unexpected price jump at day 166.

```
res <- residuals(naive(goog200))
autoplot(res) + xlab("Day") + ylab("") +
  ggtitle("Residuals from naïve method")
```

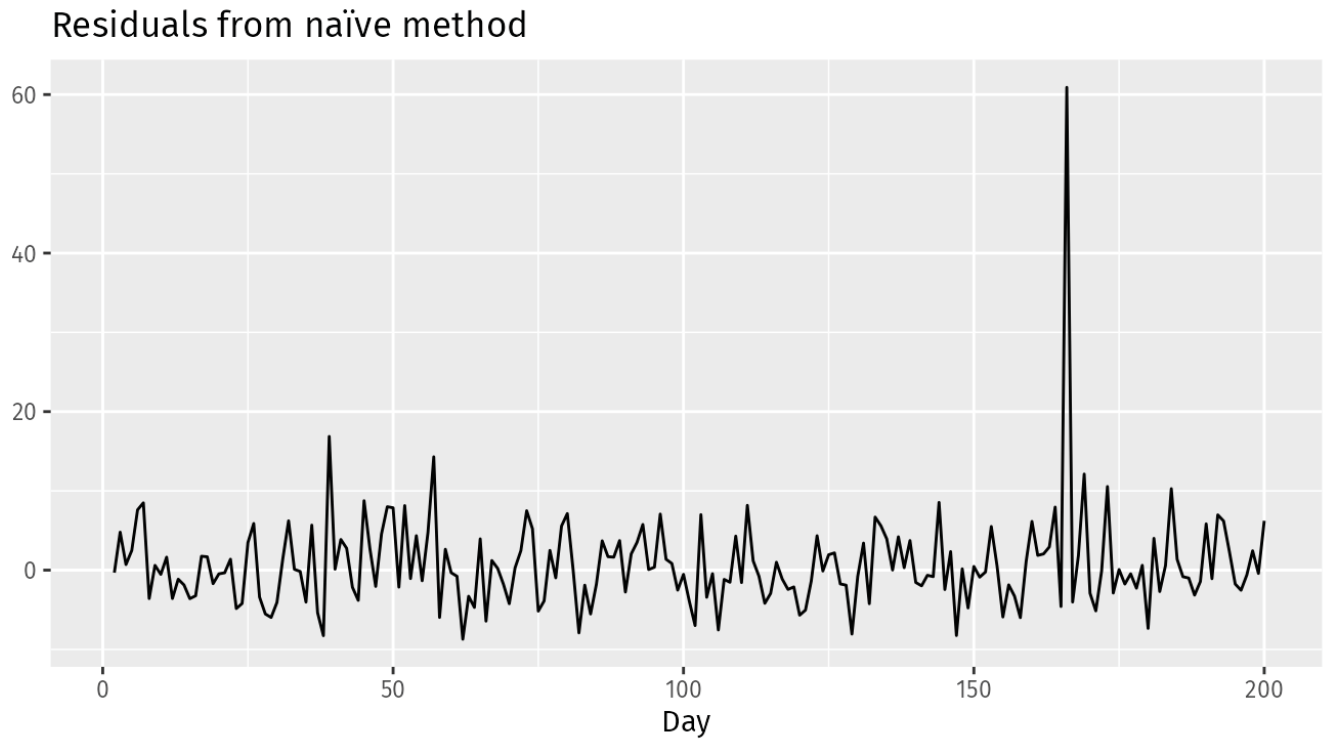


Figure 3.6: Residuals from forecasting the Google stock price using the naïve method.

```
gghistogram(res) + ggtitle("Histogram of residuals")
```

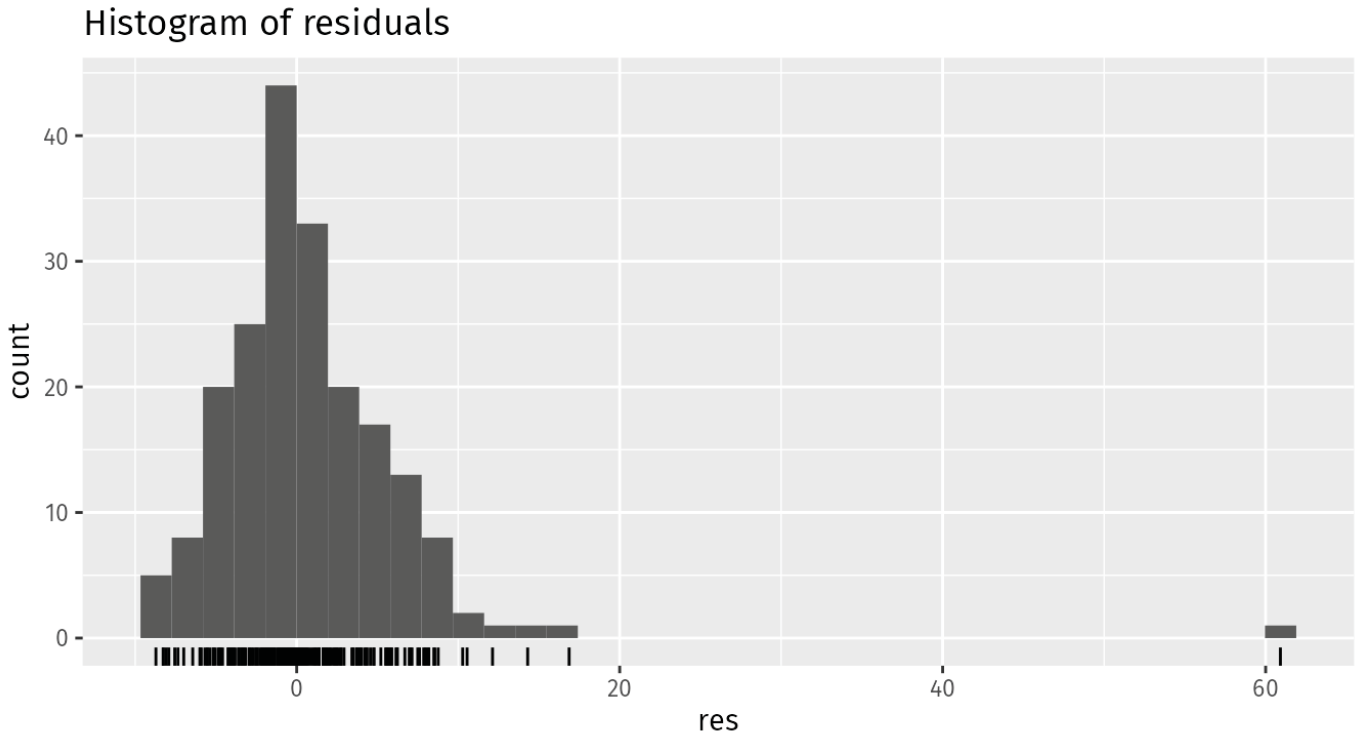


Figure 3.7: Histogram of the residuals from the naïve method applied to the Google stock price. The right tail seems a little too long for a normal distribution.

```
ggAcf(res) + ggtitle("ACF of residuals")
```

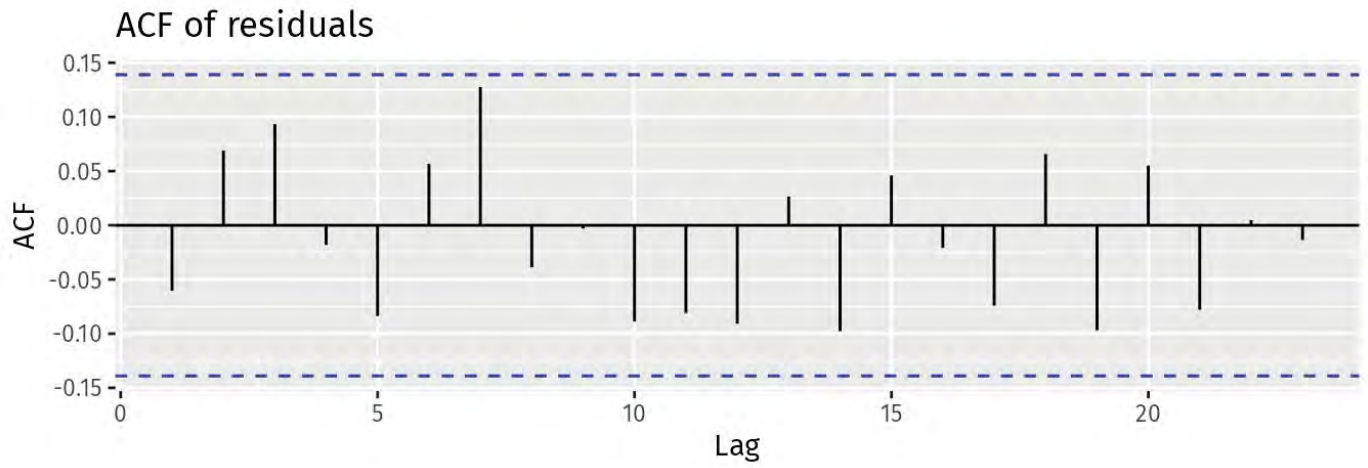


Figure 3.8: ACF of the residuals from the naïve method applied to the Google stock price. The lack of correlation suggesting the forecasts are good.

These graphs show that the naïve method produces forecasts that appear to account for all available information. The mean of the residuals is close to zero and there is no significant correlation in the residuals series. The time plot of the residuals shows that the variation of the residuals stays much the same across the historical data, apart from the one outlier, and therefore the residual variance can be treated as constant. This can also be seen on the histogram of the residuals. The histogram suggests that the residuals may not be normal — the right tail seems a little too long, even when we ignore the outlier. Consequently, forecasts from this method will probably be quite good, but prediction intervals that are computed assuming a normal distribution may be inaccurate.

Portmanteau tests for autocorrelation

In addition to looking at the ACF plot, we can also do a more formal test for autocorrelation by considering a whole set of r_k values as a group, rather than treating each one separately.

Recall that r_k is the autocorrelation for lag k. When we look at the ACF plot to see whether each spike is within the required limits, we are implicitly carrying out multiple hypothesis tests, each one with a small probability of giving a false positive. When enough of these tests are done, it is likely that at least one will give a false positive, and so we may conclude that the residuals have some remaining autocorrelation, when in fact they do not.

In order to overcome this problem, we test whether the first h autocorrelations are significantly different from what would be expected from a white noise process. A test for a group of autocorrelations is called a **portmanteau test**, from a French word

describing a suitcase or coat rack carrying several items of clothing.

One such test is the **Box-Pierce test**, based on the following statistic

$$Q = T\sum_{k=1}^{\ell} r_k^2,$$

where ℓ is the maximum lag being considered and T is the number of observations. If each r_k is close to zero, then Q will be small. If some r_k values are large (positive or negative), then Q will be large. We suggest using $\ell = 10$ for non-seasonal data and $\ell = 2m$ for seasonal data, where m is the period of seasonality. However, the test is not good when ℓ is large, so if these values are larger than $T/5$, then use $\ell = T/5$

A related (and more accurate) test is the **Ljung-Box test**, based on

$$Q^* = T(T+2)\sum_{k=1}^{\ell}(T-k)^{-1}r_k^2.$$

Again, large values of Q^* suggest that the autocorrelations do not come from a white noise series.

How large is too large? If the autocorrelations did come from a white noise series, then both Q and Q^* would have a χ^2 distribution with ℓ degrees of freedom.[2].

In the following code, `lag` $= \ell$.

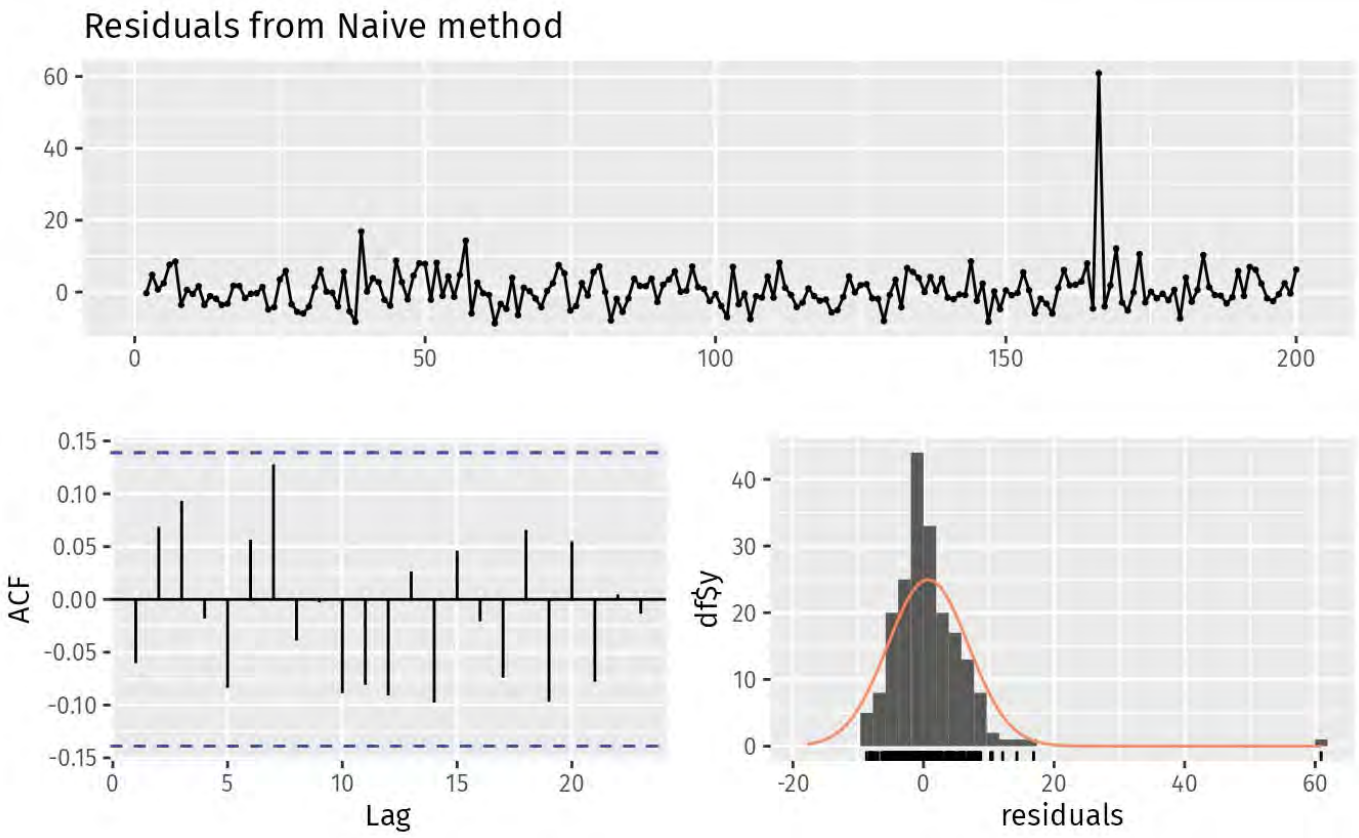
```
# lag=h and fitdf=K
Box.test(res, lag=10)
#>
#>  Box-Pierce test
#>
#> data:  res
#> X-squared = 11, df = 10, p-value = 0.4

Box.test(res,lag=10, type="Lj")
#>
#>  Box-Ljung test
#>
#> data:  res
#> X-squared = 11, df = 10, p-value = 0.4
```

For both Q and Q^*, the results are not significant (i.e., the p-values are relatively large). Thus, we can conclude that the residuals are not distinguishable from a white noise series.

All of these methods for checking residuals are conveniently packaged into one R function `checkresiduals()`, which will produce a time plot, ACF plot and histogram of the residuals (with an overlaid normal distribution for comparison), and do a Ljung-Box test.

```
checkresiduals(naive(goog200))
```




```
#>
#>  Ljung-Box test
#>
#> data:  Residuals from Naive method
#> Q* = 11, df = 10, p-value = 0.4
#>
#> Model df: 0.   Total lags used: 10
```

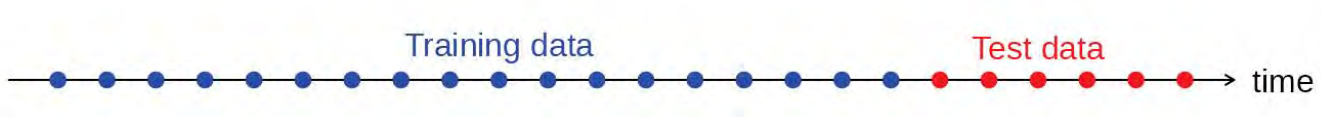
2. For the ARIMA models discussed in chapters 8 and 9, the degrees of freedom is adjusted to give better results.↩

3.4 Evaluating forecast accuracy

Training and test sets

It is important to evaluate forecast accuracy using genuine forecasts. Consequently, the size of the residuals is not a reliable indication of how large true forecast errors are likely to be. The accuracy of forecasts can only be determined by considering how well a model performs on new data that were not used when fitting the model.

When choosing models, it is common practice to separate the available data into two portions, **training** and **test** data, where the training data is used to estimate any parameters of a forecasting method and the test data is used to evaluate its accuracy. Because the test data is not used in determining the forecasts, it should provide a reliable indication of how well the model is likely to forecast on new data.



The size of the test set is typically about 20% of the total sample, although this value depends on how long the sample is and how far ahead you want to forecast. The test set should ideally be at least as large as the maximum forecast horizon required. The following points should be noted.

- A model which fits the training data well will not necessarily forecast well.
- A perfect fit can always be obtained by using a model with enough parameters.
- Over-fitting a model to data is just as bad as failing to identify a systematic pattern in the data.

Some references describe the test set as the “hold-out set” because these data are “held out” of the data used for fitting. Other references call the training set the “in-sample data” and the test set the “out-of-sample data”. We prefer to use “training data” and “test data” in this book.

Functions to subset a time series

The `window()` function introduced in Chapter 2 is useful when extracting a portion of a time series, such as we need when creating training and test sets. In the `window()` function, we specify the start and/or end of the portion of time series required using time values. For example,

```
window(ausbeer, start=1995)
```

extracts all data from 1995 onward.

Another useful function is `subset()` which allows for more types of subsetting. A great advantage of this function is that it allows the use of indices to choose a subset. For example,

```
subset(ausbeer, start=length(ausbeer)-4*5)
```

extracts the last 5 years of observations from `ausbeer`. It also allows extracting all values for a specific season. For example,

```
subset(ausbeer, quarter = 1)
```

extracts the first quarters for all years.

Finally, `head` and `tail` are useful for extracting the first few or last few observations. For example, the last 5 years of `ausbeer` can also be obtained using

```
tail(ausbeer, 4*5)
```

Forecast errors

A forecast "error" is the difference between an observed value and its forecast. Here "error" does not mean a mistake, it means the unpredictable part of an observation. It can be written as

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

where the training data is given by $\{y_1, \dots, y_T\}$ and the test data is given by $\{y_{T+1}, y_{T+2}, \dots\}$.

Note that forecast errors are different from residuals in two ways. First, residuals are calculated on the *training* set while forecast errors are calculated on the *test* set. Second, residuals are based on *one-step* forecasts while forecast errors can involve

multi-step forecasts.

We can measure forecast accuracy by summarising the forecast errors in different ways.

Scale-dependent errors

The forecast errors are on the same scale as the data. Accuracy measures that are based only on e_t are therefore scale-dependent and cannot be used to make comparisons between series that involve different units.

The two most commonly used scale-dependent measures are based on the absolute errors or squared errors:

$$\text{Mean absolute error: MAE} = \text{mean}(|e_t|),$$
$$\text{Root mean squared error: RMSE} = \sqrt{\text{mean}(e_t^2)}.$$

When comparing forecast methods applied to a single time series, or to several time series with the same units, the MAE is popular as it is easy to both understand and compute. A forecast method that minimises the MAE will lead to forecasts of the median, while minimising the RMSE will lead to forecasts of the mean. Consequently, the RMSE is also widely used, despite being more difficult to interpret.

Percentage errors

The percentage error is given by $p_t = 100e_t/y_t$. Percentage errors have the advantage of being unit-free, and so are frequently used to compare forecast performances between data sets. The most commonly used measure is:

$$\text{Mean absolute percentage error: MAPE} = \text{mean}(|p_t|).$$

Measures based on percentage errors have the disadvantage of being infinite or undefined if $y_t = 0$ for any t in the period of interest, and having extreme values if any y_t is close to zero. Another problem with percentage errors that is often overlooked is that they assume the unit of measurement has a meaningful zero.[3] For example, a percentage error makes no sense when measuring the accuracy of temperature forecasts on either the Fahrenheit or Celsius scales, because temperature has an arbitrary zero point.

They also have the disadvantage that they put a heavier penalty on negative errors than on positive errors. This observation led to the use of the so-called "symmetric" MAPE (sMAPE) proposed by Armstrong (1978, p. 348), which was used in the M3 forecasting competition. It is defined by

$$\text{sMAPE} = \text{mean}\left(200|y_t - \hat{y}_t|/(y_t + \hat{y}_t)\right).$$

However, if y_t is close to zero, $\hat{y}_t$ is also likely to be close to zero. Thus, the measure still involves division by a number close to zero, making the calculation unstable. Also, the value of sMAPE can be negative, so it is not really a measure of "absolute percentage errors" at all.

Hyndman & Koehler (2006) recommend that the sMAPE not be used. It is included here only because it is widely used, although we will not use it in this book.

Scaled errors

Scaled errors were proposed by Hyndman & Koehler (2006) as an alternative to using percentage errors when comparing forecast accuracy across series with different units. They proposed scaling the errors based on the *training* MAE from a simple forecast method.

For a non-seasonal time series, a useful way to define a scaled error uses naïve forecasts:

$$q_j = \frac{e_j}{\frac{1}{T-1}\sum_{t=2}^{T}|y_t - y_{t-1}|}.$$

Because the numerator and denominator both involve values on the scale of the original data, q_j is independent of the scale of the data. A scaled error is less than one if it arises from a better forecast than the average naïve forecast computed on the training data. Conversely, it is greater than one if the forecast is worse than the average naïve forecast computed on the training data.

For seasonal time series, a scaled error can be defined using seasonal naïve forecasts:

$$q_j = \frac{e_j}{\frac{1}{T-m}\sum_{t=m+1}^{T}|y_t - y_{t-m}|}.$$

The *mean absolute scaled error* is simply

$$\text{MASE} = \text{mean}(|q_j|).$$

Examples

```
beer2 <- window(ausbeer,start=1992,end=c(2007,4))
beerfit1 <- meanf(beer2,h=10)
beerfit2 <- rwf(beer2,h=10)
beerfit3 <- snaive(beer2,h=10)
autoplot(window(ausbeer, start=1992)) +
  autolayer(beerfit1, series="Mean", PI=FALSE) +
  autolayer(beerfit2, series="Naïve", PI=FALSE) +
  autolayer(beerfit3, series="Seasonal naïve", PI=FALSE) +
  xlab("Year") + ylab("Megalitres") +
  ggtitle("Forecasts for quarterly beer production") +
  guides(colour=guide_legend(title="Forecast"))
```

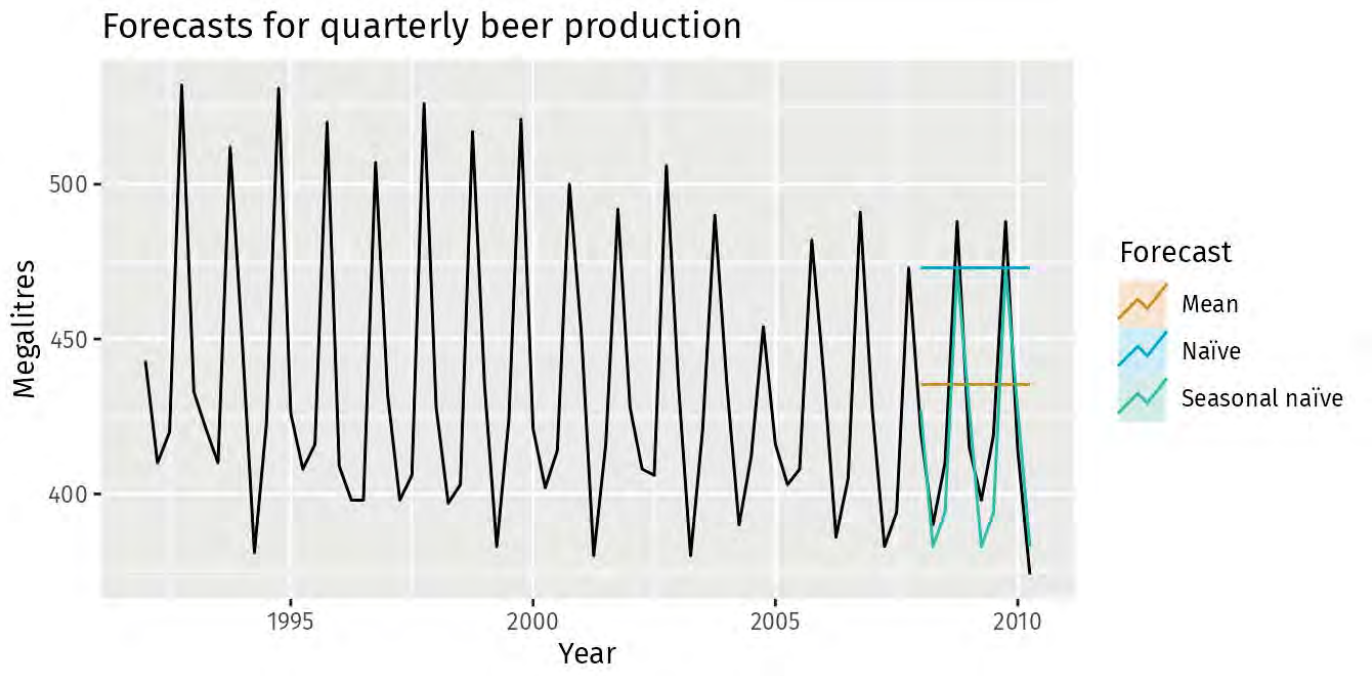


Figure 3.9: Forecasts of Australian quarterly beer production using data up to the end of 2007.

Figure 3.9 shows three forecast methods applied to the quarterly Australian beer production using data only to the end of 2007. The actual values for the period 2008–2010 are also shown. We compute the forecast accuracy measures for this period.

```
beer3 <- window(ausbeer, start=2008)
accuracy(beerfit1, beer3)
accuracy(beerfit2, beer3)
accuracy(beerfit3, beer3)
```

	RMSE	MAE	MAPE	MASE
Mean method	38.45	34.83	8.28	2.44
Naïve method	62.69	57.40	14.18	4.01
Seasonal naïve method	14.31	13.40	3.17	0.94

It is obvious from the graph that the seasonal naïve method is best for these data, although it can still be improved, as we will discover later. Sometimes, different accuracy measures will lead to different results as to which forecast method is best. However, in this case, all of the results point to the seasonal naïve method as the best of these three methods for this data set.

To take a non-seasonal example, consider the Google stock price. The following graph shows the 200 observations ending on 6 Dec 2013, along with forecasts of the next 40 days obtained from three different methods.

```
googfc1 <- meanf(goog200, h=40)
googfc2 <- rwf(goog200, h=40)
googfc3 <- rwf(goog200, drift=TRUE, h=40)
autoplot(subset(goog, end = 240)) +
  autolayer(googfc1, PI=FALSE, series="Mean") +
  autolayer(googfc2, PI=FALSE, series="Naïve") +
  autolayer(googfc3, PI=FALSE, series="Drift") +
  xlab("Day") + ylab("Closing Price (US$)") +
  ggtitle("Google stock price (daily ending 6 Dec 13)") +
  guides(colour=guide_legend(title="Forecast"))
```

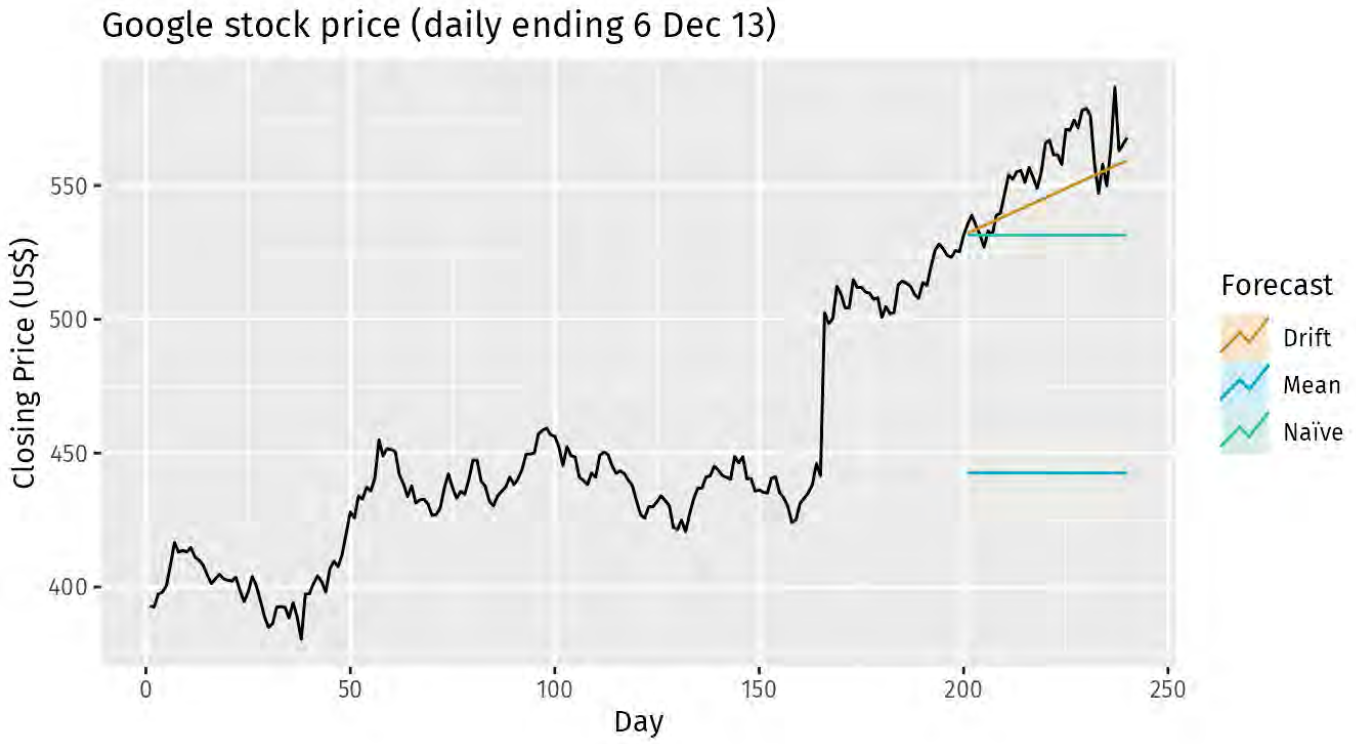


Figure 3.10: Forecasts of the Google stock price from 7 Dec 2013.

```
googtest <- window(goog, start=201, end=240)
accuracy(googfc1, googtest)
accuracy(googfc2, googtest)
accuracy(googfc3, googtest)
```

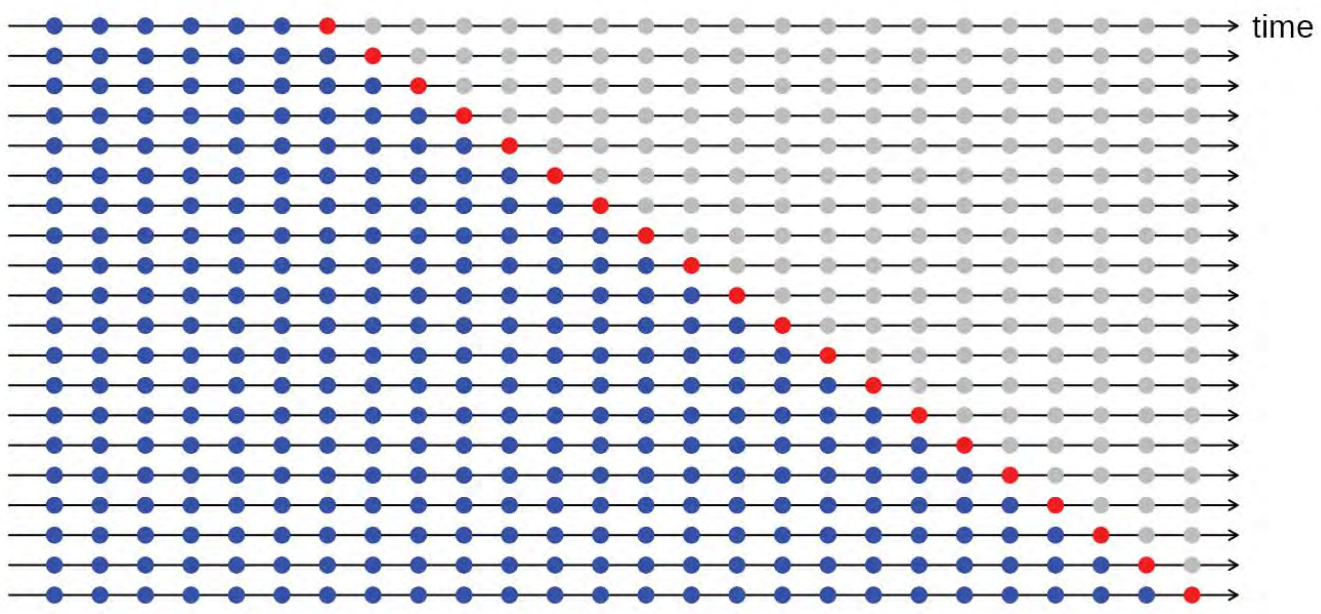
	RMSE	MAE	MAPE	MASE
Mean method	114.21	113.27	20.32	30.28
Naïve method	28.43	24.59	4.36	6.57
Drift method	14.08	11.67	2.07	3.12

Here, the best method is the drift method (regardless of which accuracy measure is used).

Time series cross-validation

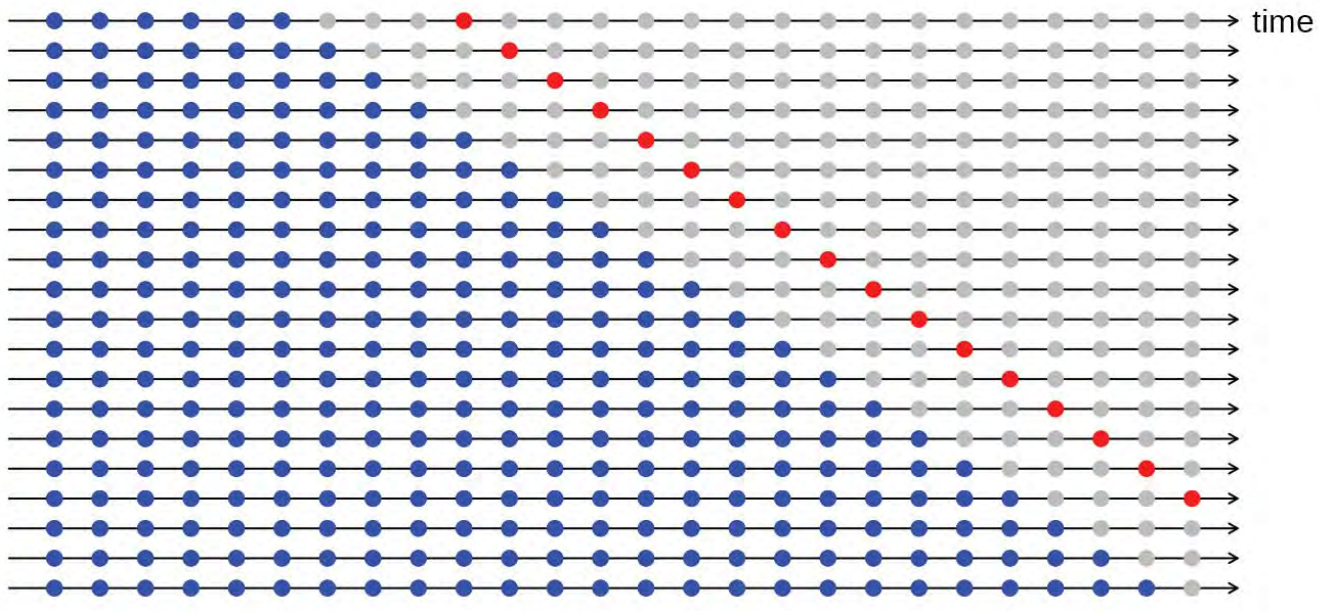
A more sophisticated version of training/test sets is time series cross-validation. In this procedure, there are a series of test sets, each consisting of a single observation. The corresponding training set consists only of observations that occurred *prior* to the observation that forms the test set. Thus, no future observations can be used in constructing the forecast. Since it is not possible to obtain a reliable forecast based on a small training set, the earliest observations are not considered as test sets.

The following diagram illustrates the series of training and test sets, where the blue observations form the training sets, and the red observations form the test sets.



The forecast accuracy is computed by averaging over the test sets. This procedure is sometimes known as "evaluation on a rolling forecasting origin" because the "origin" at which the forecast is based rolls forward in time.

With time series forecasting, one-step forecasts may not be as relevant as multi-step forecasts. In this case, the cross-validation procedure based on a rolling forecasting origin can be modified to allow multi-step errors to be used. Suppose that we are interested in models that produce good 4-step-ahead forecasts. Then the corresponding diagram is shown below.



Time series cross-validation is implemented with the `tsCV()` function. In the following example, we compare the RMSE obtained via time series cross-validation with the residual RMSE.

```
e <- tsCV(goog200, rwf, drift=TRUE, h=1)
sqrt(mean(e^2, na.rm=TRUE))
#> [1] 6.233
sqrt(mean(residuals(rwf(goog200, drift=TRUE))^2, na.rm=TRUE))
#> [1] 6.169
```

As expected, the RMSE from the residuals is smaller, as the corresponding "forecasts" are based on a model fitted to the entire data set, rather than being true forecasts.

A good way to choose the best forecasting model is to find the model with the smallest RMSE computed using time series cross-validation.

Pipe operator

The ugliness of the above R code makes this a good opportunity to introduce some alternative ways of stringing R functions together. In the above code, we are nesting functions within functions within functions, so you have to read the code from the inside out, making it difficult to understand what is being computed. Instead, we can use the pipe operator `%>%` as follows.

```
goog200 %>% tsCV(forecastfunction=rwf, drift=TRUE, h=1) -> e
e^2 %>% mean(na.rm=TRUE) %>% sqrt()
#> [1] 6.233
goog200 %>% rwf(drift=TRUE) %>% residuals() -> res
res^2 %>% mean(na.rm=TRUE) %>% sqrt()
#> [1] 6.169
```

The left hand side of each pipe is passed as the first argument to the function on the right hand side. This is consistent with the way we read from left to right in English. When using pipes, all other arguments must be named, which also helps readability. When using pipes, it is natural to use the right arrow assignment `->` rather than the left arrow. For example, the third line above can be read as "Take the `goog200` series, pass it to `rwf()` with `drift=TRUE`, compute the resulting residuals, and store them as `res`".

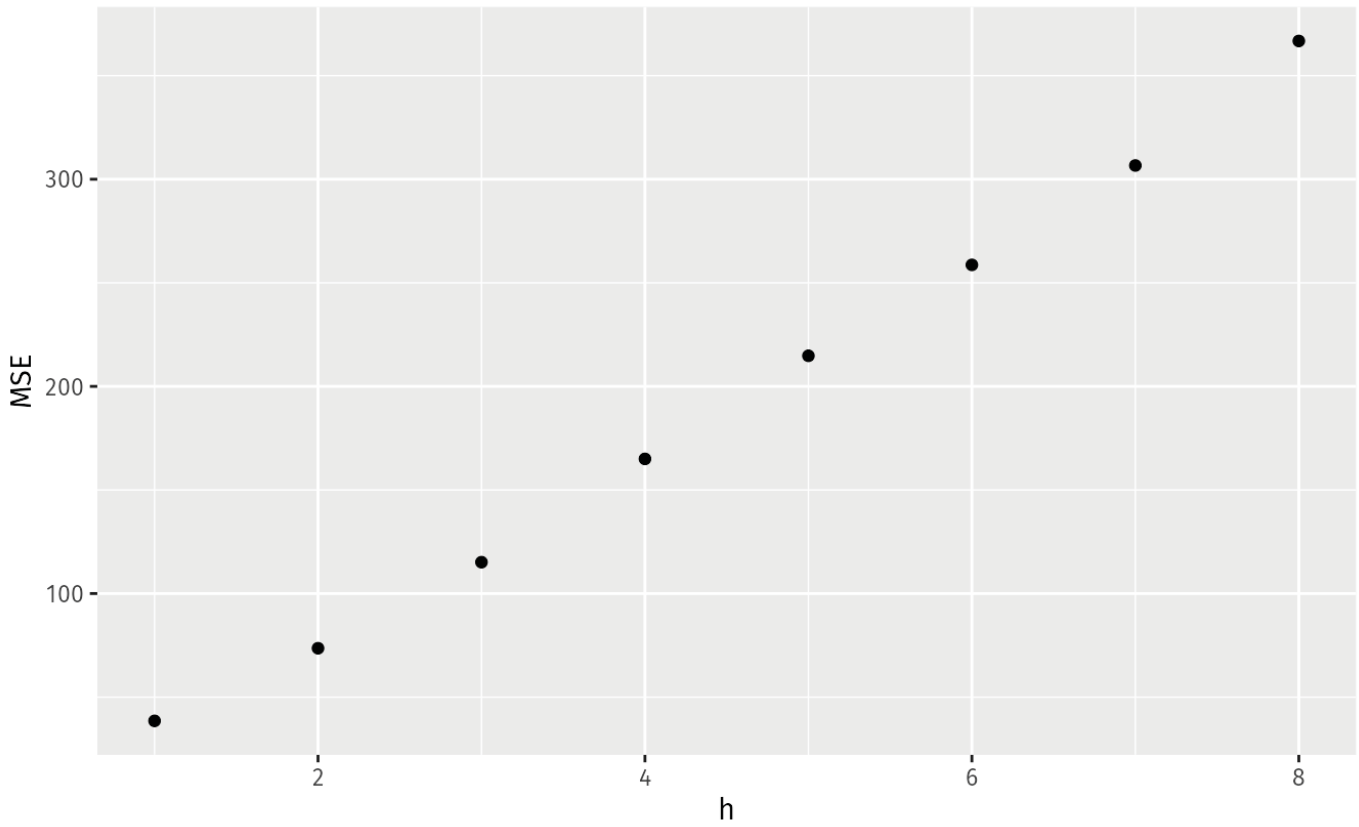
We will use the pipe operator whenever it makes the code easier to read. In order to be consistent, we will always follow a function with parentheses to differentiate it from other objects, even if it has no arguments. See, for example, the use of `sqrt()` and `residuals()` in the code above.

Example: using tsCV()

The `goog200` data, plotted in Figure 3.5, includes daily closing stock price of Google Inc from the NASDAQ exchange for 200 consecutive trading days starting on 25 February 2013.

The code below evaluates the forecasting performance of 1- to 8-step-ahead naïve forecasts with `tsCV()`, using MSE as the forecast error measure. The plot shows that the forecast error increases as the forecast horizon increases, as we would expect.

```
e <- tsCV(goog200, forecastfunction=naive, h=8)
# Compute the MSE values and remove missing values
mse <- colMeans(e^2, na.rm = T)
# Plot the MSE values against the forecast horizon
data.frame(h = 1:8, MSE = mse) %>%
  ggplot(aes(x = h, y = MSE)) + geom_point()
```



Bibliography

Armstrong, J. S. (1978). *Long-range forecasting: From crystal ball to computer*. John Wiley & Sons. [Amazon]

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. [DOI]

3. That is, a percentage is valid on a ratio scale, but not on an interval scale. Only ratio scale variables have meaningful zeros.↩

3.5 Prediction intervals

As discussed in Section 1.7, a prediction interval gives an interval within which we expect y_t to lie with a specified probability. For example, assuming that the forecast errors are normally distributed, a 95% prediction interval for the h-step forecast is

$$\hat{y}_{T+h|T} \pm 1.96\hat{\sigma}_h,$$

where $\hat{\sigma}_h$ is an estimate of the standard deviation of the h-step forecast distribution.

More generally, a prediction interval can be written as

$$\hat{y}_{T+h|T} \pm c\hat{\sigma}_h$$

where the multiplier c depends on the coverage probability. In this book we usually calculate 80% intervals and 95% intervals, although any percentage may be used. The following table gives the value of c for a range of coverage probabilities assuming normally distributed forecast errors.

Table 3.1: Multipliers to be used for prediction intervals.

Percentage	Multiplier
50	0.67
55	0.76
60	0.84
65	0.93
70	1.04
75	1.15
80	1.28
85	1.44
90	1.64
95	1.96
96	2.05
97	2.17
98	2.33
99	2.58

The value of prediction intervals is that they express the uncertainty in the forecasts. If we only produce point forecasts, there is no way of telling how accurate the forecasts are. However, if we also produce prediction intervals, then it is clear how much uncertainty is associated with each forecast. For this reason, point forecasts can be of almost no value without the accompanying prediction intervals.

One-step prediction intervals

When forecasting one step ahead, the standard deviation of the forecast distribution is almost the same as the standard deviation of the residuals. (In fact, the two standard deviations are identical if there are no parameters to be estimated, as is the case with the naïve method. For forecasting methods involving parameters to be estimated, the standard deviation of the forecast distribution is slightly larger than the residual standard deviation, although this difference is often ignored.)

For example, consider a naïve forecast for the Google stock price data `goog200` (shown in Figure 3.5). The last value of the observed series is 531.48, so the forecast of the next value of the GSP is 531.48. The standard deviation of the residuals from the naïve method is 6.21. Hence, a 95% prediction interval for the next value of the GSP is

$$531.48 \pm 1.96(6.21) = [519.3, 543.6].$$

Similarly, an 80% prediction interval is given by

$$531.48 \pm 1.28(6.21) = [523.5, 539.4].$$

The value of the multiplier (1.96 or 1.28) is taken from Table 3.1.

Multi-step prediction intervals

A common feature of prediction intervals is that they increase in length as the forecast horizon increases. The further ahead we forecast, the more uncertainty is associated with the forecast, and thus the wider the prediction intervals. That is, σ_h usually increases with h (although there are some non-linear forecasting methods that do not have this property).

To produce a prediction interval, it is necessary to have an estimate of σ_h. As already noted, for one-step forecasts ($h = 1$), the residual standard deviation provides a good estimate of the forecast standard deviation σ_1. For multi-step forecasts, a more

complicated method of calculation is required. These calculations assume that the residuals are uncorrelated.

Benchmark methods

For the four benchmark methods, it is possible to mathematically derive the forecast standard deviation under the assumption of uncorrelated residuals. If $\hat{\sigma}_h$ denotes the standard deviation of the h-step forecast distribution, and $\hat{\sigma}$ is the residual standard deviation, then we can use the following expressions.

Mean forecasts: $\hat{\sigma}_h = \hat{\sigma}\sqrt{1 + 1/T}$

Naïve forecasts: $\hat{\sigma}_h = \hat{\sigma}\sqrt{h}$

Seasonal naïve forecasts $\hat{\sigma}_h = \hat{\sigma}\sqrt{k + 1}$, where k is the integer part of $(h - 1)/m$ and m is the seasonal period.

Drift forecasts: $\hat{\sigma}_h = \hat{\sigma}\sqrt{h(1 + h/T)}$.

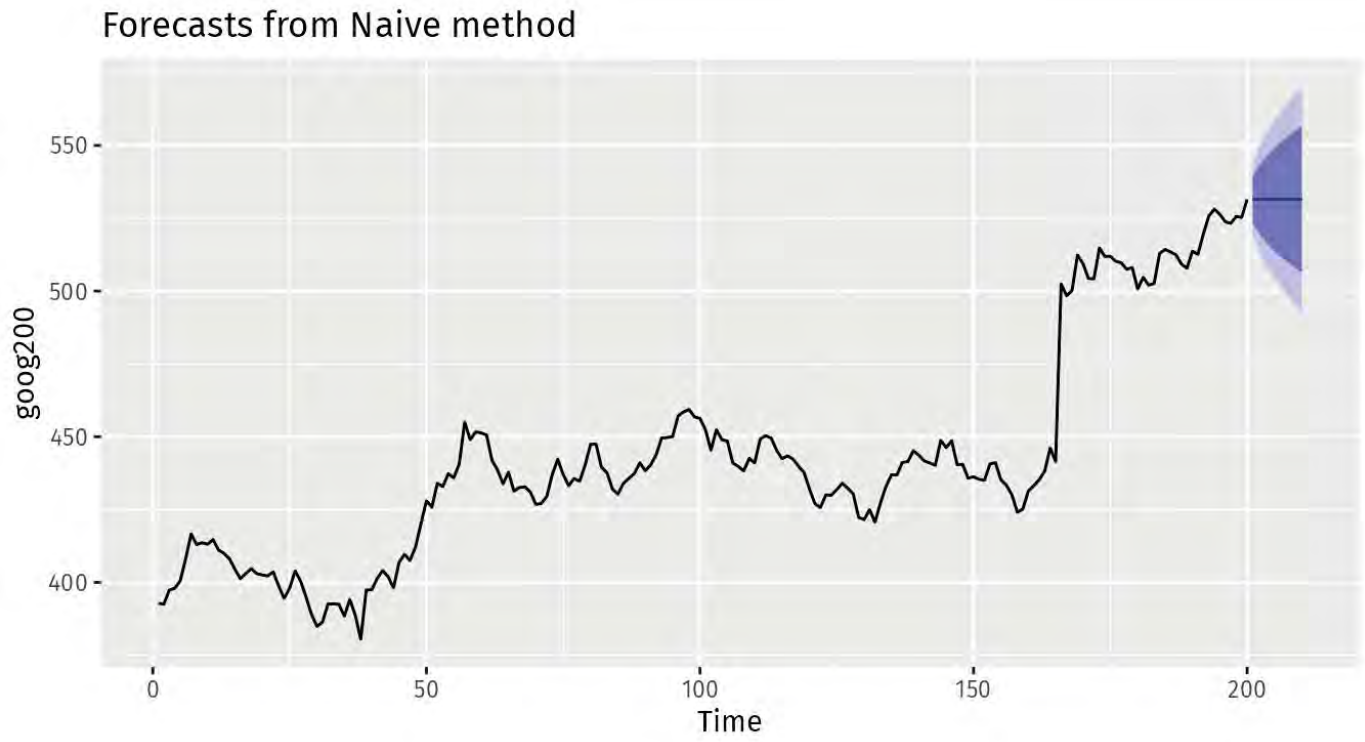
Note that when $h = 1$ and T is large, these all give the same approximate value $\hat{\sigma}$.

Prediction intervals will be computed for you when using any of the benchmark forecasting methods. For example, here is the output when using the naïve method for the Google stock price.

```
naive(goog200)
#>     Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
#> 201          531.5 523.5 539.4 519.3 543.6
#> 202          531.5 520.2 542.7 514.3 548.7
#> 203          531.5 517.7 545.3 510.4 552.6
#> 204          531.5 515.6 547.4 507.1 555.8
#> 205          531.5 513.7 549.3 504.3 558.7
#> 206          531.5 512.0 551.0 501.7 561.3
#> 207          531.5 510.4 552.5 499.3 563.7
#> 208          531.5 509.0 554.0 497.1 565.9
#> 209          531.5 507.6 555.3 495.0 568.0
#> 210          531.5 506.3 556.6 493.0 570.0
```

When plotted, the prediction intervals are shown as shaded region, with the strength of colour indicating the probability associated with the interval.

```
autoplot(naive(goog200))
```



Prediction intervals from bootstrapped residuals

When a normal distribution for the forecast errors is an unreasonable assumption, one alternative is to use bootstrapping, which only assumes that the forecast errors are uncorrelated.

A forecast error is defined as $e_t = y_t - \hat{y}_{t|t-1}$. We can re-write this as

$$y_t = \hat{y}_{t|t-1} + e_t.$$

So we can simulate the next observation of a time series using

$$y_{T+1} = \hat{y}_{T+1|T} + e_{T+1}$$

where $\hat{y}_{T+1|T}$ is the one-step forecast and e_{T+1} is the unknown future error. Assuming future errors will be similar to past errors, we can replace e_{T+1} by sampling from the collection of errors we have seen in the past (i.e., the residuals). Adding the new simulated observation to our data set, we can repeat the process to obtain

$$y_{T+2} = \hat{y}_{T+2|T+1} + e_{T+2}$$

where e_{T+2} is another draw from the collection of residuals. Continuing in this way, we can simulate an entire set of future values for our time series.

Doing this repeatedly, we obtain many possible futures. Then we can compute prediction intervals by calculating percentiles for each forecast horizon. The result is called a **bootstrapped** prediction interval. The name "bootstrap" is a reference to pulling ourselves up by our bootstraps, because the process allows us to measure future uncertainty by only using the historical data.

To generate such intervals, we can simply add the `bootstrap` argument to our forecasting functions. For example:

```
naive(goog200, bootstrap=TRUE)
#>     Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
#> 201          531.5 525.7 537.8 522.9 542.9
#> 202          531.5 523.2 539.5 519.4 547.0
#> 203          531.5 520.9 541.2 516.7 552.3
#> 204          531.5 519.0 543.0 514.0 560.3
#> 205          531.5 517.5 544.6 511.8 582.1
#> 206          531.5 516.1 545.9 509.5 582.4
#> 207          531.5 514.8 547.3 508.0 583.5
#> 208          531.5 513.5 548.9 505.8 584.9
#> 209          531.5 512.3 549.8 503.9 586.6
#> 210          531.5 510.7 551.4 502.1 587.5
```

In this case, they are similar (but not identical) to the prediction intervals based on the normal distribution.

Prediction intervals with transformations

If a transformation has been used, then the prediction interval should be computed on the transformed scale, and the end points back-transformed to give a prediction interval on the original scale. This approach preserves the probability coverage of the prediction interval, although it will no longer be symmetric around the point forecast.

The back-transformation of prediction intervals is done automatically using the functions in the **forecast** package in R, provided you have used the `lambda` argument when computing the forecasts.

3.6 The forecast package in R

This book uses the facilities in the **forecast** package in R (which is loaded automatically whenever you load the **fpp2** package). This appendix briefly summarises some of the features of the package. Please refer to the help files for individual functions to learn more, and to see some examples of their use.

Functions that output a forecast object:

Many functions, including `meanf()`, `naive()`, `snaive()` and `rwf()`, produce output in the form of a `forecast` object (i.e., an object of class `forecast`). This allows other functions (such as `autoplot()`) to work consistently across a range of forecasting models.

Objects of class `forecast` contain information about the forecasting method, the data used, the point forecasts obtained, prediction intervals, residuals and fitted values. There are several functions designed to work with these objects including `autoplot()`, `summary()` and `print()`.

The following list shows all the functions that produce `forecast` objects.

- `meanf()`
- `naive()`, `snaive()`
- `rwf()`
- `croston()`
- `stlf()`
- `ses()`
- `holt()`, `hw()`
- `splinef()`
- `thetaf()`
- `forecast()`

`forecast()` function

So far we have used functions which produce a `forecast` object directly. But a more common approach, which we will focus on in the rest of the book, will be to fit a model to the data, and then use the `forecast()` function to produce forecasts from that model.

The `forecast()` function works with many different types of inputs. It generally takes a time series or time series model as its main argument, and produces forecasts appropriately. It always returns objects of class `forecast` .

If the first argument is of class `ts` , it returns forecasts from the automatic ETS algorithm discussed in Chapter 7.

Here is a simple example, applying `forecast()` to the `ausbeer` data:

```
forecast(ausbeer, h=4)
#>         Point Forecast Lo 80 Hi 80 Lo 95 Hi 95
#> 2010 Q3          404.6 385.9 423.3 376.0 433.3
#> 2010 Q4          480.4 457.5 503.3 445.4 515.4
#> 2011 Q1          417.0 396.5 437.6 385.6 448.4
#> 2011 Q2          383.1 363.5 402.7 353.1 413.1
```

That works quite well if you have no idea what sort of model to use. But by the end of this book, you should not need to use `forecast()` in this “blind” fashion. Instead, you will fit a model appropriate to the data, and then use `forecast()` to produce forecasts from that model.

3.7 Exercises

1. For the following series, find an appropriate Box-Cox transformation in order to stabilise the variance.
   - `usnetelec`
   - `usgdp`
   - `mcopper`
   - `enplanements`
2. Why is a Box-Cox transformation unhelpful for the `cangas` data?
3. What Box-Cox transformation would you select for your retail data (from Exercise 3 in Section 2.10)?
4. For each of the following series, make a graph of the data. If transforming seems appropriate, do so and describe the effect. `dole`, `usdeaths`, `bricksq`.
5. Calculate the residuals from a seasonal naïve forecast applied to the quarterly Australian beer production data from 1992. The following code will help.

```
beer <- window(ausbeer, start=1992)
fc <- snaive(beer)
autoplot(fc)
res <- residuals(fc)
autoplot(res)
```

   Test if the residuals are white noise and normally distributed.

```
checkresiduals(fc)
```

   What do you conclude?
6. Repeat the exercise for the `WWWusage` and `bricksq` data. Use whichever of `naive()` or `snaive()` is more appropriate in each case.
7. Are the following statements true or false? Explain your answer.
   a. Good forecast methods should have normally distributed residuals.
   b. A model with small residuals will give good forecasts.
   c. The best measure of forecast accuracy is MAPE.

d. If your model doesn't forecast well, you should make it more complicated.
e. Always choose the model with the best forecast accuracy as measured on the test set.

8. For your retail time series (from Exercise 3 in Section 2.10):

a. Split the data into two parts using

```
myts.train <- window(myts, end=c(2010,12))
myts.test <- window(myts, start=2011)
```

b. Check that your data have been split appropriately by producing the following plot.

```
autoplot(myts) +
  autolayer(myts.train, series="Training") +
  autolayer(myts.test, series="Test")
```

c. Calculate forecasts using `snaive` applied to `myts.train`.

```
fc <- snaive(myts.train)
```

d. Compare the accuracy of your forecasts against the actual values stored in `myts.test`.

```
accuracy(fc,myts.test)
```

e. Check the residuals.

```
checkresiduals(fc)
```

Do the residuals appear to be uncorrelated and normally distributed?

f. How sensitive are the accuracy measures to the training/test split?

9. `visnights` contains quarterly visitor nights (in millions) from 1998 to 2016 for twenty regions of Australia.

a. Use `window()` to create three training sets for `visnights[,"QLDMetro"]`, omitting the last 1, 2 and 3 years; call these train1, train2, and train3, respectively. For example `train1 <- window(visnights[, "QLDMetro"], end = c(2015, 4))`.

b. Compute one year of forecasts for each training set using the `snaive()` method. Call these `fc1`, `fc2` and `fc3`, respectively.

c. Use `accuracy()` to compare the MAPE over the three test sets. Comment on these.

10. Use the Dow Jones index (data set `dowjones`) to do the following:

    a. Produce a time plot of the series.
    b. Produce forecasts using the drift method and plot them.
    c. Show that the forecasts are identical to extending the line drawn between the first and last observations.
    d. Try using some of the other benchmark functions to forecast the same data set. Which do you think is best? Why?

11. Consider the daily closing IBM stock prices (data set `ibmclose`).

    a. Produce some plots of the data in order to become familiar with it.
    b. Split the data into a training set of 300 observations and a test set of 69 observations.
    c. Try using various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?
    d. Check the residuals of your preferred method. Do they resemble white noise?

12. Consider the sales of new one-family houses in the USA, Jan 1973 – Nov 1995 (data set `hsales`).

    a. Produce some plots of the data in order to become familiar with it.
    b. Split the `hsales` data set into a training set and a test set, where the test set is the last two years of data.
    c. Try using various benchmark methods to forecast the training set and compare the results on the test set. Which method did best?
    d. Check the residuals of your preferred method. Do they resemble white noise?

3.8 Further reading

- Ord et al. (2017) provides further discussion of simple benchmark forecasting methods.
- A review of forecast evaluation methods is given in Hyndman & Koehler (2006), looking at the strengths and weaknesses of different approaches. This is the paper that introduced the MASE as a general-purpose forecast accuracy measure.

Bibliography

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, *22*(4), 679–688. [DOI]

Ord, J. K., Fildes, R., & Kourentzes, N. (2017). *Principles of business forecasting* (2nd ed.). Wessex Press Publishing Co. [Amazon]

Chapter 4 Judgmental forecasts

Forecasting using judgement is common in practice. In many cases, judgmental forecasting is the only option, such as when there is a complete lack of historical data, or when a new product is being launched, or when a new competitor enters the market, or during completely new and unique market conditions. For example, in December 2012, the Australian government was the first in the world to pass legislation that banned the use of company logos on cigarette packets, and required all cigarette packets to be a dark green colour. Judgement must be applied in order to forecast the effect of such a policy, as there are no historical precedents.

There are also situations where the data are incomplete, or only become available after some delay. For example, central banks include judgement when forecasting the current level of economic activity, a procedure known as nowcasting, as GDP is only available on a quarterly basis.

Research in this area[4] has shown that the accuracy of judgmental forecasting improves when the forecaster has (i) important domain knowledge, and (ii) more timely, up-to-date information. A judgmental approach can be quick to adjust to such changes, information or events.

Over the years, the acceptance of judgmental forecasting as a science has increased, as has the recognition of its need. More importantly, the quality of judgmental forecasts has also improved, as a direct result of recognising that improvements in judgmental forecasting can be achieved by implementing well-structured and systematic approaches. It is important to recognise that judgmental forecasting is subjective and comes with limitations. However, implementing systematic and well-structured approaches can confine these limitations and markedly improve forecast accuracy.

There are three general settings in which judgmental forecasting is used: (i) there are no available data, so that statistical methods are not applicable and judgmental forecasting is the only feasible approach; (ii) data are available, statistical forecasts are generated, and these are then adjusted using judgement; and (iii) data are available and statistical and judgmental forecasts are generated independently and then combined. We should clarify that when data are available, applying statistical

methods (such as those discussed in other chapters of this book), is preferable and should always be used as a starting point. Statistical forecasts are generally superior to generating forecasts using only judgement. For the majority of the chapter, we focus on the first setting where no data are available, and in the last section we discuss the judgmental adjustment of statistical forecasts. We discuss combining forecasts in Section 12.4.

Bibliography

Lawrence, M., Goodwin, P., O'Connor, M., & Önkal, D. (2006). Judgmental forecasting: A review of progress over the last 25 years. *International Journal of Forecasting*, *22*(3), 493–518. [DOI]

4. Lawrence, Goodwin, O'Connor, & Önkal (2006)↩

4.1 Beware of limitations

Judgmental forecasts are subjective, and therefore do not come free of bias or limitations.

Judgmental forecasts can be inconsistent. Unlike statistical forecasts, which can be generated by the same mathematical formulas every time, judgmental forecasts depend heavily on human cognition, and are vulnerable to its limitations. For example, a limited memory may render recent events more important than they actually are and may ignore momentous events from the more distant past; or a limited attention span may result in important information being missed; or a misunderstanding of causal relationships may lead to erroneous inferences. Furthermore, human judgement can vary due to the effect of psychological factors. One can imagine a manager who is in a positive frame of mind one day, generating forecasts that may tend to be somewhat optimistic, and in a negative frame of mind another day, generating somewhat less optimistic forecasts.

Judgement can be clouded by personal or political agendas, where targets and forecasts (as defined in Chapter 1) are not segregated. For example, if a sales manager knows that the forecasts she generates will be used to set sales expectations (targets), she may tend to set these low in order to show a good performance (i.e., exceed the expected targets). Even in cases where targets and forecasts are well segregated, judgement may be plagued by optimism or wishful thinking. For example, it would be highly unlikely that a team working towards launching a new product would forecast its failure. As we will discuss later, this optimism can be accentuated in a group meeting setting. "Beware of the enthusiasm of your marketing and sales colleagues"[5].

Another undesirable property which is commonly seen in judgmental forecasting is the effect of anchoring. In this case, the subsequent forecasts tend to converge or be close to an initial familiar reference point. For example, it is common to take the last observed value as a reference point. The forecaster is influenced unduly by prior information, and therefore gives this more weight in the forecasting process. Anchoring may lead to conservatism and undervaluing new and more current information, and thereby create a systematic bias.

Bibliography

Fildes, R., & Goodwin, P. (2007b). Good and bad judgment in forecasting: Lessons from four companies. *Foresight: The International Journal of Applied Forecasting*, (8), 5–10.

5. Fildes & Goodwin (2007b)↩

4.2 Key principles

Using a systematic and well structured approach in judgmental forecasting helps to reduce the adverse effects of the limitations of judgmental forecasting, some of which we listed in the previous section. Whether this approach involves one individual or many, the following principles should be followed.

Set the forecasting task clearly and concisely

Care is needed when setting the forecasting challenges and expressing the forecasting tasks. It is important that everyone be clear about what the task is. All definitions should be clear and comprehensive, avoiding ambiguous and vague expressions. Also, it is important to avoid incorporating emotive terms and irrelevant information that may distract the forecaster. In the Delphi method that follows (see Section 4.3), it may sometimes be useful to conduct a preliminary round of information gathering before setting the forecasting task.

Implement a systematic approach

Forecast accuracy and consistency can be improved by using a systematic approach to judgmental forecasting involving checklists of categories of information which are relevant to the forecasting task. For example, it is helpful to identify what information is important and how this information is to be weighted. When forecasting the demand for a new product, what factors should we account for and how should we account for them? Should it be the price, the quality and/or quantity of the competition, the economic environment at the time, the target population of the product? It is worthwhile to devote significant effort and resources to put together decision rules that will lead to the best possible systematic approach.

Document and justify

Formalising and documenting the decision rules and assumptions implemented in the systematic approach can promote consistency, as the same rules can be implemented repeatedly. Also, requesting a forecaster to document and justify their

forecasts leads to accountability, which can lead to reduced bias. Furthermore, formal documentation aids significantly in the systematic evaluation process that is suggested next.

Systematically evaluate forecasts

Systematically monitoring the forecasting process can identify unforeseen irregularities. In particular, keep records of forecasts and use them to obtain feedback when the corresponding observations become available. Although you may do your best as a forecaster, the environment you operate in is dynamic. Changes occur, and you need to monitor these in order to evaluate the decision rules and assumptions. Feedback and evaluation help forecasters learn and improve their forecast accuracy.

Segregate forecasters and users

Forecast accuracy may be impeded if the forecasting task is carried out by users of the forecasts, such as those responsible for implementing plans of action about which the forecast is concerned. We should clarify again here (as in Section 1.2), that forecasting is about predicting the future as accurately as possible, given all of the information available, including historical data and knowledge of any future events that may impact the forecasts. Forecasters and users should be clearly segregated. A classic case is that of a new product being launched. The forecast should be a reasonable estimate of the sales volume of a new product, which may differ considerably from what management expects or hopes the sales will be in order to meet company financial objectives. In this case, a forecaster may be delivering a reality check to the user.

It is important that forecasters communicate forecasts to potential users thoroughly. As we will see in Section 4.7, users may feel distant and disconnected from forecasts, and may not have full confidence in them. Explaining and clarifying the process and justifying the basic assumptions that led to the forecasts will provide some assurance to users.

The way in which forecasts may then be used and implemented will clearly depend on managerial decision making. For example, management may decide to adjust a forecast upwards (be over-optimistic), as the forecast may be used to guide purchasing and stock keeping levels. Such a decision may be taken after a cost-

benefit analysis reveals that the cost of holding excess stock is much lower than that of lost sales. This type of adjustment should be part of setting goals or planning supply, rather than part of the forecasting process. In contrast, if forecasts are used as targets, they may be set low so that they can be exceeded more easily. Again, setting targets is different from producing forecasts, and the two should not be confused.

The example that follows comes from our experience in industry. It exemplifies two contrasting styles of judgmental forecasting — one that adheres to the principles we have just presented and one that does not.

Example: Pharmaceutical Benefits Scheme (PBS)

The Australian government subsidises the cost of a wide range of prescription medicines as part of the PBS. Each subsidised medicine falls into one of four categories: concession copayments, concession safety net, general copayments, and general safety net. Each person with a concession card makes a concession copayment per PBS medicine ($5.80)[6], until they reach a set threshold amount labelled the concession safety net ($348). For the rest of the financial year, all PBS-listed medicines are free. Each general patient makes a general copayment per PBS medicine ($35.40) until the general safety net amount is reached ($1,363.30). For the rest of the financial year, they contribute a small amount per PBS-listed medicine ($5.80). The PBS forecasting process uses 84 groups of PBS-listed medicines, and produces forecasts of the medicine volume and the total expenditure for each group and for each of the four PBS categories, a total of 672 series. This forecasting process aids in setting the government budget allocated to the PBS, which is over $7 billion per year, or approximately 1% of GDP.

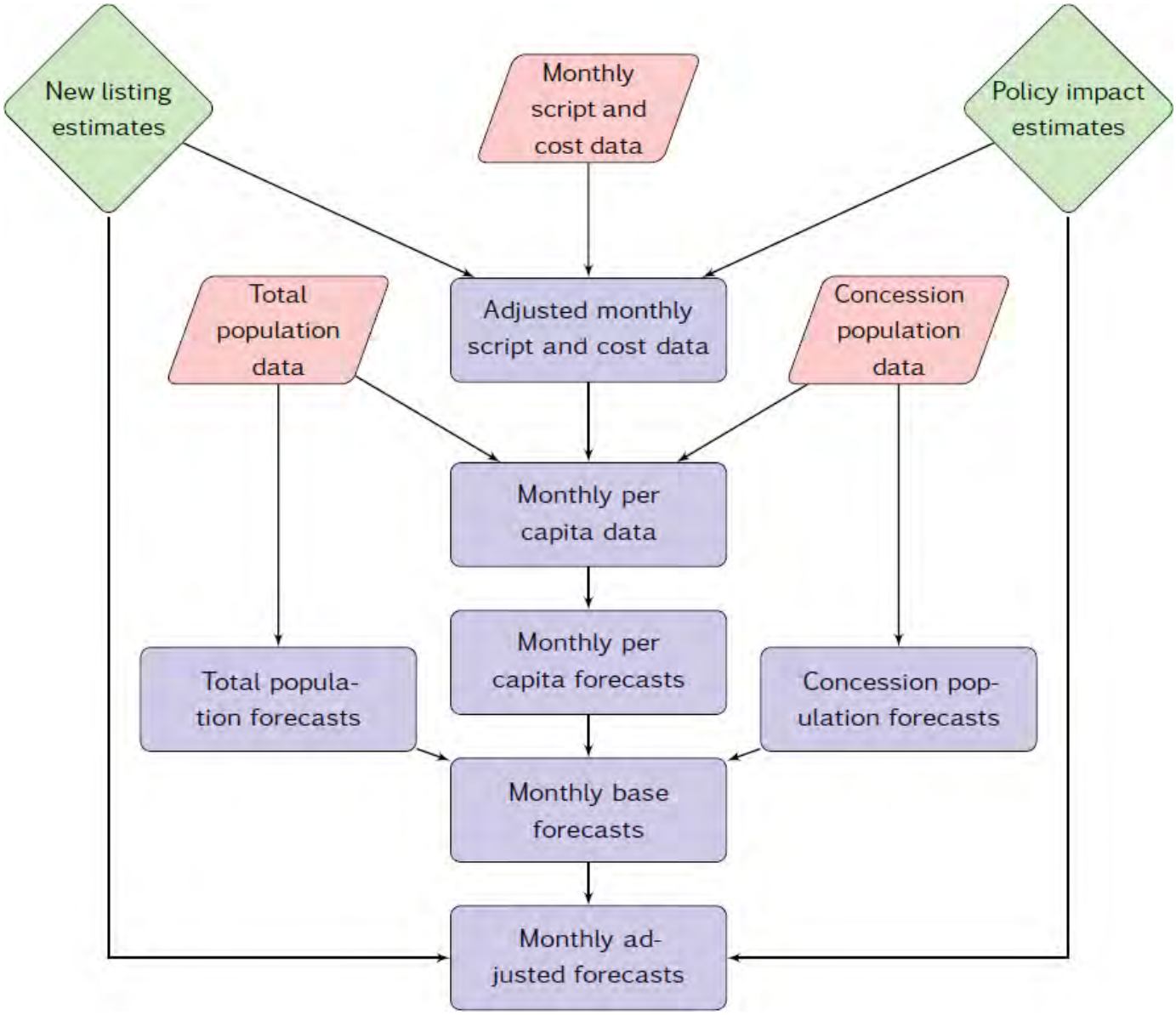


Figure 4.1: Process for producing PBS forecasts.

Figure 4.1 summarises the forecasting process. Judgmental forecasts are generated for new listings of medicines and for estimating the impact of new policies. These are shown by the green items. The pink items indicate the data used which were obtained from various government departments and associated authorities. The blue items show things that are calculated from the data provided. There were judgmental adjustments to the data to take account of new listings and new policies, and there were also judgmental adjustments to the forecasts. Because of the changing size of both the concession population and the total population, forecasts are produced on a per-capita basis, and then multiplied by the forecast population to obtain forecasts of total volume and expenditure per month.

One of us (Hyndman) was asked to evaluate the forecasting process a few years ago. We found that using judgement for new listings and new policy impacts gave better forecasts than using a statistical model alone. However, we also found that the

forecasting accuracy and consistency could be improved through a more structured and systematic process, especially for policy impacts.

Forecasting new listings: Companies who apply for their medicine to be added to the PBS are asked to submit detailed forecasts for various aspects of the medicine, such as projected patient numbers, market share of the new medicine, substitution effects, etc. The Pharmaceutical Benefits Advisory Committee provides guidelines describing a highly structured and systematic approach for generating these forecasts, and requires careful documentation for each step of the process. This structured process helps to reduce the likelihood and effects of deliberate self-serving biases. Two detailed evaluation rounds of the company forecasts are implemented by a sub-committee, one before the medicine is added to the PBS and one after it is added. Finally, comparisons of observations versus forecasts for some selected new listings are performed, 12 months and 24 months after the listings, and the results are sent back to the companies for comment.

Policy impact forecasts: In contrast to the highly structured process used for new listings, there were no systematic procedures for policy impact forecasts. On many occasions, forecasts of policy impacts were calculated by a small team, and were often heavily reliant on the work of one person. The forecasts were not usually subject to a formal review process. There were no guidelines for how to construct judgmental forecasts for policy impacts, and there was often a lack of adequate documentation about how these forecasts were obtained, the assumptions underlying them, etc.

Consequently, we recommended several changes:

- that guidelines for forecasting new policy impacts be developed, to encourage a more systematic and structured forecasting approach;
- that the forecast methodology be documented in each case, including all assumptions made in forming the forecasts;
- that new policy forecasts be made by at least two people from different areas of the organisation;
- that a review of forecasts be conducted one year after the implementation of each new policy by a review committee, especially for new policies that have a significant annual projected cost or saving. The review committee should include those involved in generating the forecasts, but also others.

These recommendations reflect the principles outlined in this section.

6. These are Australian dollar amounts published by the Australian government for 2012.↩

4.3 The Delphi method

The Delphi method was invented by Olaf Helmer and Norman Dalkey of the Rand Corporation in the 1950s for the purpose of addressing a specific military problem. The method relies on the key assumption that forecasts from a group are generally more accurate than those from individuals. The aim of the Delphi method is to construct consensus forecasts from a group of experts in a structured iterative manner. A facilitator is appointed in order to implement and manage the process. The Delphi method generally involves the following stages:

1. A panel of experts is assembled.
2. Forecasting tasks/challenges are set and distributed to the experts.
3. Experts return initial forecasts and justifications. These are compiled and summarised in order to provide feedback.
4. Feedback is provided to the experts, who now review their forecasts in light of the feedback. This step may be iterated until a satisfactory level of consensus is reached.
5. Final forecasts are constructed by aggregating the experts' forecasts.

Each stage of the Delphi method comes with its own challenges. In what follows, we provide some suggestions and discussions about each one of these.[7]

Experts and anonymity

The first challenge of the facilitator is to identify a group of experts who can contribute to the forecasting task. The usual suggestion is somewhere between 5 and 20 experts with diverse expertise. Experts submit forecasts and also provide detailed qualitative justifications for these.

A key feature of the Delphi method is that the participating experts remain anonymous at all times. This means that the experts cannot be influenced by political and social pressures in their forecasts. Furthermore, all experts are given an equal say and all are held accountable for their forecasts. This avoids the situation where a group meeting is held and some members do not contribute, while others dominate. It also prevents members exerting undue influence based on seniority or personality. There have been suggestions that even something as simple as the

seating arrangements in a group setting can influence the group dynamics. Furthermore, there is ample evidence that a group meeting setting promotes enthusiasm and influences individual judgement, leading to optimism and overconfidence.[8]

A by-product of anonymity is that the experts do not need to meet as a group in a physical location. An important advantage of this is that it increases the likelihood of gathering experts with diverse skills and expertise from varying locations. Furthermore, it makes the process cost-effective by eliminating the expense and inconvenience of travel, and it makes it flexible, as the experts only have to meet a common deadline for submitting forecasts, rather than having to set a common meeting time.

Setting the forecasting task in a Delphi

In a Delphi setting, it may be useful to conduct a preliminary round of information gathering from the experts before setting the forecasting tasks. Alternatively, as experts submit their initial forecasts and justifications, valuable information which is not shared between all experts can be identified by the facilitator when compiling the feedback.

Feedback

Feedback to the experts should include summary statistics of the forecasts and outlines of qualitative justifications. Numerical data summaries and graphical representations can be used to summarise the experts' forecasts.

As the feedback is controlled by the facilitator, there may be scope to direct attention and information from the experts to areas where it is most required. For example, the facilitator may direct the experts' attention to responses that fall outside the interquartile range, and the qualitative justification for such forecasts.

Iteration

The process of the experts submitting forecasts, receiving feedback, and reviewing their forecasts in light of the feedback, is repeated until a satisfactory level of consensus between the experts is reached. Satisfactory consensus does not mean complete convergence in the forecast value; it simply means that the variability of

the responses has decreased to a satisfactory level. Usually two or three rounds are sufficient. Experts are more likely to drop out as the number of iterations increases, so too many rounds should be avoided.

Final forecasts

The final forecasts are usually constructed by giving equal weight to all of the experts' forecasts. However, the facilitator should keep in mind the possibility of extreme values which can distort the final forecast.

Limitations and variations

Applying the Delphi method can be time consuming. In a group meeting, final forecasts can possibly be reached in hours or even minutes — something which is almost impossible to do in a Delphi setting. If it is taking a long time to reach a consensus in a Delphi setting, the panel may lose interest and cohesiveness.

In a group setting, personal interactions can lead to quicker and better clarifications of qualitative justifications. A variation of the Delphi method which is often applied is the "estimate-talk-estimate" method, where the experts can interact between iterations, although the forecast submissions can still remain anonymous. A disadvantage of this variation is the possibility of the loudest person exerting undue influence.

The facilitator

The role of the facilitator is of the utmost importance. The facilitator is largely responsible for the design and administration of the Delphi process. The facilitator is also responsible for providing feedback to the experts and generating the final forecasts. In this role, the facilitator needs to be experienced enough to recognise areas that may need more attention, and to direct the experts' attention to these. Also, as there is no face-to-face interaction between the experts, the facilitator is responsible for disseminating important information. The efficiency and effectiveness of the facilitator can dramatically increase the probability of a successful Delphi method in a judgmental forecasting setting.

Bibliography

Buehler, R., Messervey, D., & Griffin, D. (2005). Collaborative planning and prediction: Does group discussion affect optimistic biases in time estimation? *Organizational Behavior and Human Decision Processes*, *97*(1), 47–63. [DOI]

Rowe, G. (2007). A guide to Delphi. *Foresight: The International Journal of Applied Forecasting*, (8), 11–16.

Rowe, G., & Wright, G. (1999). The Delphi technique as a forecasting tool: Issues and analysis. *International Journal of Forecasting*, *15*(4), 353–375. [DOI]

7. For further reading, refer to: Rowe (2007); Rowe & Wright (1999)↩

8. Buehler, Messervey, & Griffin (2005)↩

4.4 Forecasting by analogy

A useful judgmental approach which is often implemented in practice is forecasting by analogy. A common example is the pricing of a house through an appraisal process. An appraiser estimates the market value of a house by comparing it to similar properties that have sold in the area. The degree of similarity depends on the attributes considered. With house appraisals, attributes such as land size, dwelling size, numbers of bedrooms and bathrooms, and garage space are usually considered.

Even thinking and discussing analogous products or situations can generate useful (and sometimes crucial) information. We illustrate this point with the following example.[9]

Example: Designing a high school curriculum

A small group of academics and teachers were assigned the task of developing a curriculum for teaching judgement and decision making under uncertainty for high schools in Israel. Each group member was asked to forecast how long it would take for the curriculum to be completed. Responses ranged between 18 and 30 months. One of the group members who was an expert in curriculum design was asked to consider analogous curricula developments around the world. He concluded that 40% of analogous groups he considered never completed the task. The rest took between 7 to 10 years. The Israel project was completed in 8 years.

Obviously, forecasting by analogy comes with challenges. We should aspire to base forecasts on multiple analogies rather than a single analogy, which may create biases. However, these may be challenging to identify. Similarly, we should aspire to consider multiple attributes. Identifying or even comparing these may not always be straightforward. As always, we suggest performing these comparisons and the forecasting process using a systematic approach. Developing a detailed scoring mechanism to rank attributes and record the process of ranking will always be useful.

A structured analogy

Alternatively, a structured approach comprising a panel of experts can be implemented, as was proposed by Green & Armstrong (2007). The concept is similar to that of a Delphi; however, the forecasting task is completed by considering analogies. First, a facilitator is appointed. Then the structured approach involves the following steps.

1. A panel of experts who are likely to have experience with analogous situations is assembled.
2. Tasks/challenges are set and distributed to the experts.
3. Experts identify and describe as many analogies as they can, and generate forecasts based on each analogy.
4. Experts list similarities and differences of each analogy to the target situation, then rate the similarity of each analogy to the target situation on a scale.
5. Forecasts are derived by the facilitator using a set rule. This can be a weighted average, where the weights can be guided by the ranking scores of each analogy by the experts.

As with the Delphi approach, anonymity of the experts may be an advantage in not suppressing creativity, but could hinder collaboration. Green and Armstrong found no gain in collaboration between the experts in their results. A key finding was that experts with multiple analogies (more than two), and who had direct experience with the analogies, generated the most accurate forecasts.

Bibliography

Green, K. C., & Armstrong, J. S. (2007). Structured analogies for forecasting. *International Journal of Forecasting*, *23*(3), 365–376. [DOI]

Kahneman, D., & Lovallo, D. (1993). Timid choices and bold forecasts: A cognitive perspective on risk taking. *Management Science*, *39*(1), 17–31. [DOI]

9. This example is extracted from Kahneman & Lovallo (1993)↩

4.5 Scenario forecasting

A fundamentally different approach to judgmental forecasting is scenario-based forecasting. The aim of this approach is to generate forecasts based on plausible scenarios. In contrast to the two previous approaches (Delphi and forecasting by analogy) where the resulting forecast is intended to be a likely outcome, each scenario-based forecast may have a low probability of occurrence. The scenarios are generated by considering all possible factors or drivers, their relative impacts, the interactions between them, and the targets to be forecast.

Building forecasts based on scenarios allows a wide range of possible forecasts to be generated and some extremes to be identified. For example it is usual for "best", "middle" and "worst" case scenarios to be presented, although many other scenarios will be generated. Thinking about and documenting these contrasting extremes can lead to early contingency planning.

With scenario forecasting, decision makers often participate in the generation of scenarios. While this may lead to some biases, it can ease the communication of the scenario-based forecasts, and lead to a better understanding of the results.

4.6 New product forecasting

The definition of a new product can vary. It may be an entirely new product which has been launched, a variation of an existing product ("new and improved"), a change in the pricing scheme of an existing product, or even an existing product entering a new market.

Judgmental forecasting is usually the only available method for new product forecasting, as historical data are unavailable. The approaches we have already outlined (Delphi, forecasting by analogy and scenario forecasting) are all applicable when forecasting the demand for a new product.

Other methods which are more specific to the situation are also available. We briefly describe three such methods which are commonly applied in practice. These methods are less structured than those already discussed, and are likely to lead to more biased forecasts as a result.

Sales force composite

In this approach, forecasts for each outlet/branch/store of a company are generated by salespeople, and are then aggregated. This usually involves sales managers forecasting the demand for the outlet they manage. Salespeople are usually closest to the interaction between customers and products, and often develop an intuition about customer purchasing intentions. They bring this valuable experience and expertise to the forecast.

However, having salespeople generate forecasts violates the key principle of segregating forecasters and users, which can create biases in many directions. It is common for the performance of a salesperson to be evaluated against the sales forecasts or expectations set beforehand. In this case, the salesperson acting as a forecaster may introduce some self-serving bias by generating low forecasts. On the other hand, one can imagine an enthusiastic salesperson, full of optimism, generating high forecasts.

Moreover a successful salesperson is not necessarily a successful nor well-informed forecaster. A large proportion of salespeople will have no or limited formal training in forecasting. Finally, salespeople will feel customer displeasure at first hand if, for

example, the product runs out or is not introduced in their store. Such interactions will cloud their judgement.

Executive opinion

In contrast to the sales force composite, this approach involves staff at the top of the managerial structure generating aggregate forecasts. Such forecasts are usually generated in a group meeting, where executives contribute information from their own area of the company. Having executives from different functional areas of the company promotes great skill and knowledge diversity in the group.

This process carries all of the advantages and disadvantages of a group meeting setting which we discussed earlier. In this setting, it is important to justify and document the forecasting process. That is, executives need to be held accountable in order to reduce the biases generated by the group meeting setting. There may also be scope to apply variations to a Delphi approach in this setting; for example, the estimate-talk-estimate process described earlier.

Customer intentions

Customer intentions can be used to forecast the demand for a new product or for a variation on an existing product. Questionnaires are filled in by customers on their intentions to buy the product. A structured questionnaire is used, asking customers to rate the likelihood of them purchasing the product on a scale; for example, highly likely, likely, possible, unlikely, highly unlikely.

Survey design challenges, such as collecting a representative sample, applying a time- and cost-effective method, and dealing with non-responses, need to be addressed.[10]

Furthermore, in this survey setting we must keep in mind the relationship between purchase intention and purchase behaviour. Customers do not always do what they say they will. Many studies have found a positive correlation between purchase intentions and purchase behaviour; however, the strength of these correlations varies substantially. The factors driving this variation include the timings of data collection and product launch, the definition of "new" for the product, and the type of industry. Behavioural theory tells us that intentions predict behaviour if the intentions are measured just before the behaviour.[11] The time between intention and behaviour will vary depending on whether it is a completely new product or a

variation on an existing product. Also, the correlation between intention and behaviour is found to be stronger for variations on existing and familiar products than for completely new products.

Whichever method of new product forecasting is used, it is important to thoroughly document the forecasts made, and the reasoning behind them, in order to be able to evaluate them when data become available.

Bibliography

Groves, R. M., Fowler, F. J., Couper, M. P., Lepkowski, J. M., Singer, E., & Tourangeau, R. (2009). *Survey methodology* (2nd ed). John Wiley & Sons. [Amazon]

Randall, D. M., & Wolff, J. A. (1994). The time interval in the intention-behaviour relationship: Meta-analysis. *British Journal of Social Psychology*, *33*(4), 405–418. [DOI]

10. Groves et al. (2009)↩

11. Randall & Wolff (1994)↩

4.7 Judgmental adjustments

In this final section, we consider the situation where historical data are available and are used to generate statistical forecasts. It is common for practitioners to then apply judgmental adjustments to these forecasts. These adjustments can potentially provide all of the advantages of judgmental forecasting which have been discussed earlier in this chapter. For example, they provide an avenue for incorporating factors that may not be accounted for in the statistical model, such as promotions, large sporting events, holidays, or recent events that are not yet reflected in the data. However, these advantages come to fruition only when the right conditions are present. Judgmental adjustments, like judgmental forecasts, come with biases and limitations, and we must implement methodical strategies in order to minimise them.

Use adjustments sparingly

Practitioners adjust much more often than they should, and many times for the wrong reasons. By adjusting statistical forecasts, users of forecasts create a feeling of ownership and credibility. Users often do not understand or appreciate the mechanisms that generate the statistical forecasts (as they will usually have no training in this area). By implementing judgmental adjustments, users feel that they have contributed to and completed the forecasts, and they can now relate their own intuition and interpretations to these. The forecasts have become their own.

Judgmental adjustments should not aim to correct for a systematic pattern in the data that is thought to have been missed by the statistical model. This has been proven to be ineffective, as forecasters tend to read non-existent patterns in noisy series. Statistical models are much better at taking account of data patterns, and judgmental adjustments only hinder accuracy.

Judgmental adjustments are most effective when there is significant additional information at hand or strong evidence of the need for an adjustment. We should only adjust when we have important extra information which is not incorporated in the statistical model. Hence, adjustments seem to be most accurate when they are large in size. Small adjustments (especially in the positive direction promoting the illusion of optimism) have been found to hinder accuracy, and should be avoided.

Apply a structured approach

Using a structured and systematic approach will improve the accuracy of judgmental adjustments. Following the key principles outlined in Section 4.2 is vital. In particular, having to document and justify adjustments will make it more challenging to override the statistical forecasts, and will guard against unnecessary adjustments.

It is common for adjustments to be implemented by a panel (see the example that follows). Using a Delphi setting carries great advantages. However, if adjustments are implemented in a group meeting, it is wise to consider the forecasts of key markets or products first, as panel members will get tired during this process. Fewer adjustments tend to be made as the meeting goes on through the day.

Example: Tourism Forecasting Committee (TFC)

Tourism Australia publishes forecasts for all aspects of Australian tourism twice a year. The published forecasts are generated by the TFC, an independent body which comprises experts from various government and industry sectors; for example, the Australian Commonwealth Treasury, airline companies, consulting firms, banking sector companies, and tourism bodies.

The forecasting methodology applied is an iterative process. First, model-based statistical forecasts are generated by the forecasting unit within Tourism Australia, then judgmental adjustments are made to these in two rounds. In the first round, the TFC Technical Committee[12] (comprising senior researchers, economists and independent advisers) adjusts the model-based forecasts. In the second and final round, the TFC (comprising industry and government experts) makes final adjustments. In both rounds, adjustments are made by consensus.

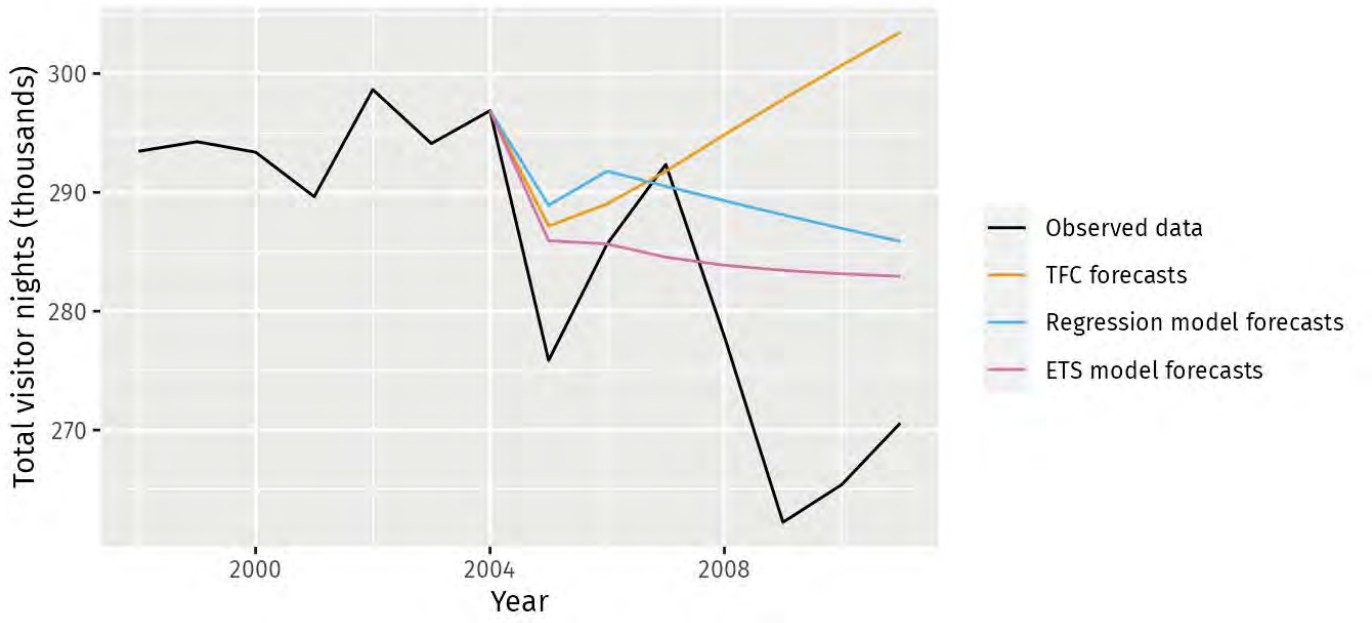


Figure 4.2: Long run annual forecasts for domestic visitor nights for Australia. We study regression models in Chapter 5, and ETS (ExponenTial Smoothing) models in Chapter 7.

In 2008, we[13] analysed forecasts for Australian domestic tourism. We concluded that the published TFC forecasts were optimistic, especially for the long-run, and we proposed alternative model-based forecasts. We now have access to observed data up to and including 2011. In Figure 4.2, we plot the published forecasts against the actual data. We can see that the published TFC forecasts have continued to be optimistic.

What can we learn from this example? Although the TFC clearly states in its methodology that it produces 'forecasts' rather than 'targets', could this be a case where these have been confused? Are the forecasters and users sufficiently well-segregated in this process? Could the iterative process itself be improved? Could the adjustment process in the meetings be improved? Could it be that the group meetings have promoted optimism? Could it be that domestic tourism should have been considered earlier in the day?

Bibliography

Athanasopoulos, G., & Hyndman, R. J. (2008). Modelling and forecasting Australian domestic tourism. *Tourism Management*, *29*(1), 19–31. [DOI]

12. GA was an observer on this technical committee for a few years.↩

4.8 Further reading

Many forecasting textbooks ignore judgmental forecasting altogether. Here are three which do cover it in some detail.

- Chapter 11 of Ord et al. (2017) provides an excellent review of some of the same topics as this chapter, but also includes using judgement to assessing forecast uncertainty, and forecasting using prediction markets.
- Goodwin & Wright (2009) is a book-length treatment of the use of judgement in decision making by two of the leading researchers in the field.
- Kahn (2006) covers techniques for new product forecasting, where judgmental methods play an important role.

There have been some helpful survey papers on judgmental forecasting published in the last 20 years. We have found these three particularly helpful.

- Fildes & Goodwin (2007b)
- Fildes & Goodwin (2007a)
- Harvey (2001)

Some helpful papers on individual judgmental forecasting methods are listed in the table below.

Forecasting Method	Recommended papers
Delphi	Rowe & Wright (1999)
	Rowe (2007)
Adjustments	Sanders et al. (2005)
	Eroglu & Croxton (2010)
	Franses & Legerstee (2013)
Analogy	Green & Armstrong (2007)
Scenarios	Önkal, Sayım, & Gönül (2013)
Customer intentions	Morwitz, Steckel, & Gupta (2007)

Bibliography

Eroglu, C., & Croxton, K. L. (2010). Biases in judgmental adjustments of statistical forecasts: The role of individual differences. *International Journal of Forecasting*, *26*(1), 116–133. [DOI]

Fildes, R., & Goodwin, P. (2007a). Against your better judgment? How organizations can improve their use of management judgment in forecasting. *Interfaces*, *37*(6), 570–576. [DOI]

Fildes, R., & Goodwin, P. (2007b). Good and bad judgment in forecasting: Lessons from four companies. *Foresight: The International Journal of Applied Forecasting*, (8), 5–10.

Franses, P. H., & Legerstee, R. (2013). Do statistical forecasting models for SKU-level data benefit from including past expert knowledge? *International Journal of Forecasting*, *29*(1), 80–87. [DOI]

Goodwin, P., & Wright, G. (2009). *Decision analysis for management judgment* (4th ed). Chichester: John Wiley & Sons. [Amazon]

Green, K. C., & Armstrong, J. S. (2007). Structured analogies for forecasting. *International Journal of Forecasting*, *23*(3), 365–376. [DOI]

Harvey, N. (2001). Improving judgment in forecasting. In J. S. Armstrong (Ed.), *Principles of forecasting: A handbook for researchers and practitioners* (pp. 59–80). Boston, MA: Kluwer Academic Publishers. [DOI]

Kahn, K. B. (2006). *New product forecasting: An applied approach*. M.E. Sharp. [Amazon]

Morwitz, V. G., Steckel, J. H., & Gupta, A. (2007). When do purchase intentions predict sales? *International Journal of Forecasting*, *23*(3), 347–364. [DOI]

Önkal, D., Sayım, K. Z., & Gönül, M. S. (2013). Scenarios as channels of forecast advice. *Technological Forecasting and Social Change*, *80*(4), 772–788. [DOI]

Ord, J. K., Fildes, R., & Kourentzes, N. (2017). *Principles of business forecasting* (2nd ed.). Wessex Press Publishing Co. [Amazon]

Rowe, G. (2007). A guide to Delphi. *Foresight: The International Journal of Applied Forecasting*, (8), 11–16.

Rowe, G., & Wright, G. (1999). The Delphi technique as a forecasting tool: Issues and analysis. *International Journal of Forecasting*, *15*(4), 353–375. [DOI]

Sanders, N., Goodwin, P., Önkal, D., Gönül, M. S., Harvey, N., Lee, A., & Kjolso, L. (2005). When and how should statistical forecasts be judgmentally adjusted? *Foresight: The International Journal of Applied Forecasting*, *1*(1), 5–23. http://www.forecastpro.com/Trends/pdf/Nada%20Sanders%20Judgmental%20Adjustments%20to%20Statistical%20Forecasts%20July%202008.pdf

Chapter 5 Time series regression models

In this chapter we discuss regression models. The basic concept is that we forecast the time series of interest y assuming that it has a linear relationship with other time series x.

For example, we might wish to forecast monthly sales y using total advertising spend x as a predictor. Or we might forecast daily electricity demand y using temperature x_1 and the day of week x_2 as predictors.

The **forecast variable** y is sometimes also called the regressand, dependent or explained variable. The **predictor variables** x are sometimes also called the regressors, independent or explanatory variables. In this book we will always refer to them as the "forecast" variable and "predictor" variables.

5.1 The linear model

Simple linear regression

In the simplest case, the regression model allows for a linear relationship between the forecast variable y and a single predictor variable x:

$$y_t = \beta_0 + \beta_1 x_t + \varepsilon_t.$$

An artificial example of data from such a model is shown in Figure 5.1. The coefficients β_0 and β_1 denote the intercept and the slope of the line respectively. The intercept β_0 represents the predicted value of y when $x = 0$. The slope β_1 represents the average predicted change in y resulting from a one unit increase in x.

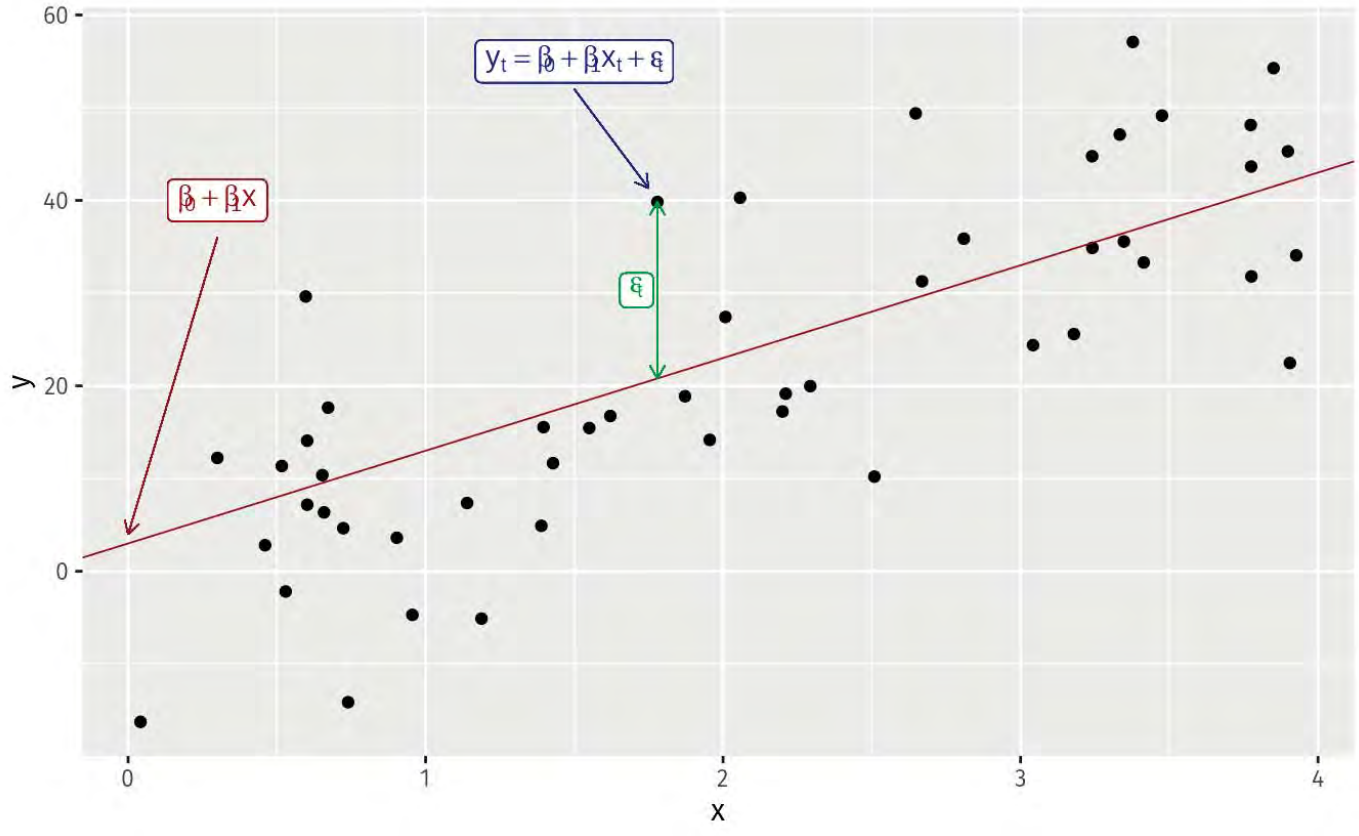


Figure 5.1: An example of data from a simple linear regression model.

Notice that the observations do not lie on the straight line but are scattered around it. We can think of each observation y_t as consisting of the systematic or explained part of the model, $\beta_0 + \beta_1 x_t$, and the random "error", ε_t. The "error" term does not imply a mistake, but a deviation from the underlying straight line model. It captures anything that may affect y_t other than x_t.

Example: US consumption expenditure

Figure 5.2 shows time series of quarterly percentage changes (growth rates) of real personal consumption expenditure, y, and real personal disposable income, x, for the US from 1970 Q1 to 2016 Q3.

```
autoplot(uschange[,c("Consumption","Income")]) +
  ylab("% change") + xlab("Year")
```

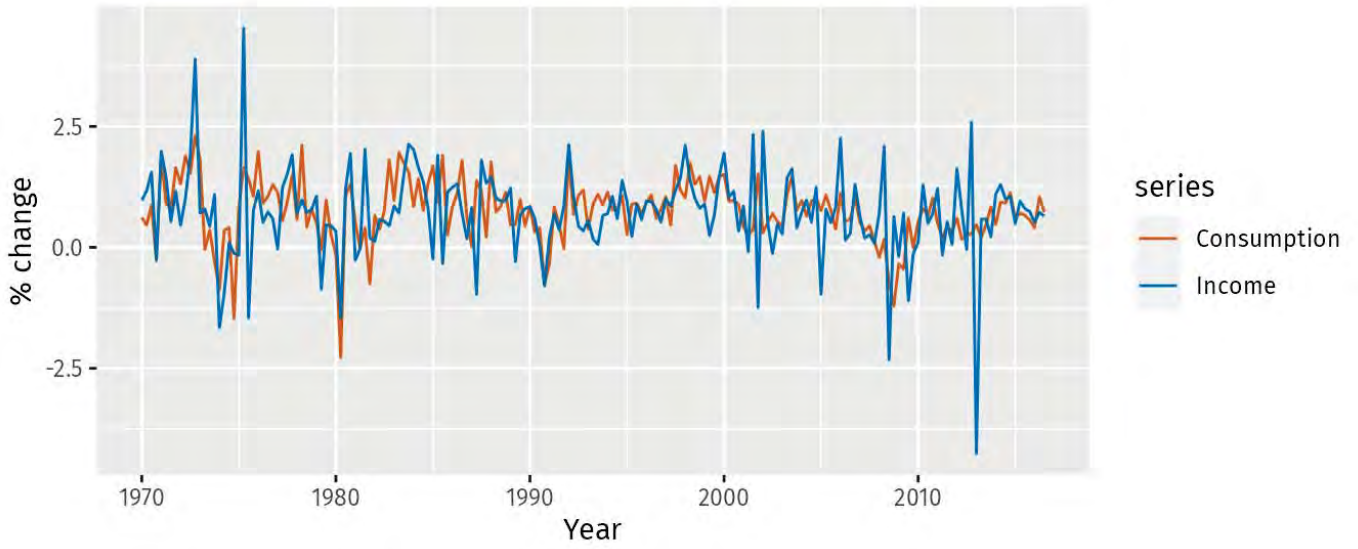


Figure 5.2: Percentage changes in personal consumption expenditure and personal income for the US.

A scatter plot of consumption changes against income changes is shown in Figure 5.3 along with the estimated regression line

$$\hat{y}_t = 0.55 + 0.28x_t.$$

(We put a "hat" above y to indicate that this is the value of y predicted by the model.)

```
uschange %>%
  as.data.frame() %>%
  ggplot(aes(x=Income, y=Consumption)) +
    ylab("Consumption (quarterly % change)") +
    xlab("Income (quarterly % change)") +
    geom_point() +
    geom_smooth(method="lm", se=FALSE)
#> `geom_smooth()` using formula = 'y ~ x'
```

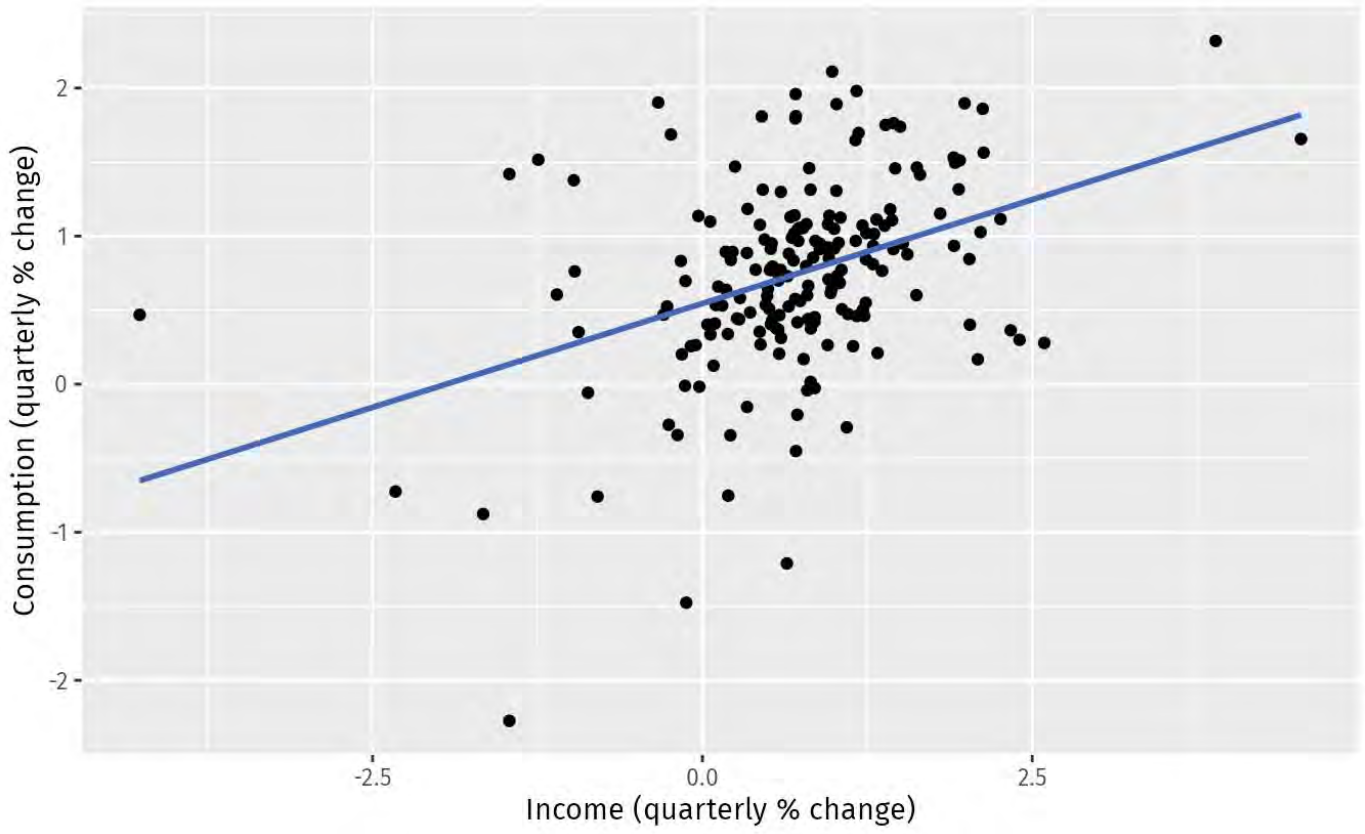


Figure 5.3: Scatterplot of quarterly changes in consumption expenditure versus quarterly changes in personal income and the fitted regression line.

The equation is estimated in R using the `tslm()` function:

```
tslm(Consumption ~ Income, data=uschange)
#>
#> Call:
#> tslm(formula = Consumption ~ Income, data = uschange)
#>
#> Coefficients:
#> (Intercept)       Income
#>       0.545        0.281
```

We will discuss how `tslm()` computes the coefficients in Section 5.2.

The fitted line has a positive slope, reflecting the positive relationship between income and consumption. The slope coefficient shows that a one unit increase in x (a 1 percentage point increase in personal disposable income) results on average in 0.28 units increase in y (an average increase of 0.28 percentage points in personal consumption expenditure). Alternatively the estimated equation shows that a value of 1 for x (the percentage increase in personal disposable income) will result in a forecast value of $0.55 + 0.28 \times 1 = 0.83$ for y (the percentage increase in personal consumption expenditure).

The interpretation of the intercept requires that a value of $x = 0$ makes sense. In this case when $x = 0$ (i.e., when there is no change in personal disposable income since the last quarter) the predicted value of y is 0.55 (i.e., an average increase in personal consumption expenditure of 0.55%). Even when $x = 0$ does not make sense, the intercept is an important part of the model. Without it, the slope coefficient can be distorted unnecessarily. The intercept should always be included unless the requirement is to force the regression line "through the origin". In what follows we assume that an intercept is always included in the model.

Multiple linear regression

When there are two or more predictor variables, the model is called a **multiple regression model**. The general form of a multiple regression model is

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \cdots + \beta_k x_{k,t} + \varepsilon_t, \tag{5.1}$$

where y is the variable to be forecast and $x_1, \ldots, x_k$ are the k predictor variables. Each of the predictor variables must be numerical. The coefficients $\beta_1, \ldots, \beta_k$ measure the effect of each predictor after taking into account the effects of all the other predictors in the model. Thus, the coefficients measure the *marginal effects* of the predictor variables.

Example: US consumption expenditure

Figure 5.4 shows additional predictors that may be useful for forecasting US consumption expenditure. These are quarterly percentage changes in industrial production and personal savings, and quarterly changes in the unemployment rate (as this is already a percentage). Building a multiple linear regression model can potentially generate more accurate forecasts as we expect consumption expenditure to not only depend on personal income but on other predictors as well.

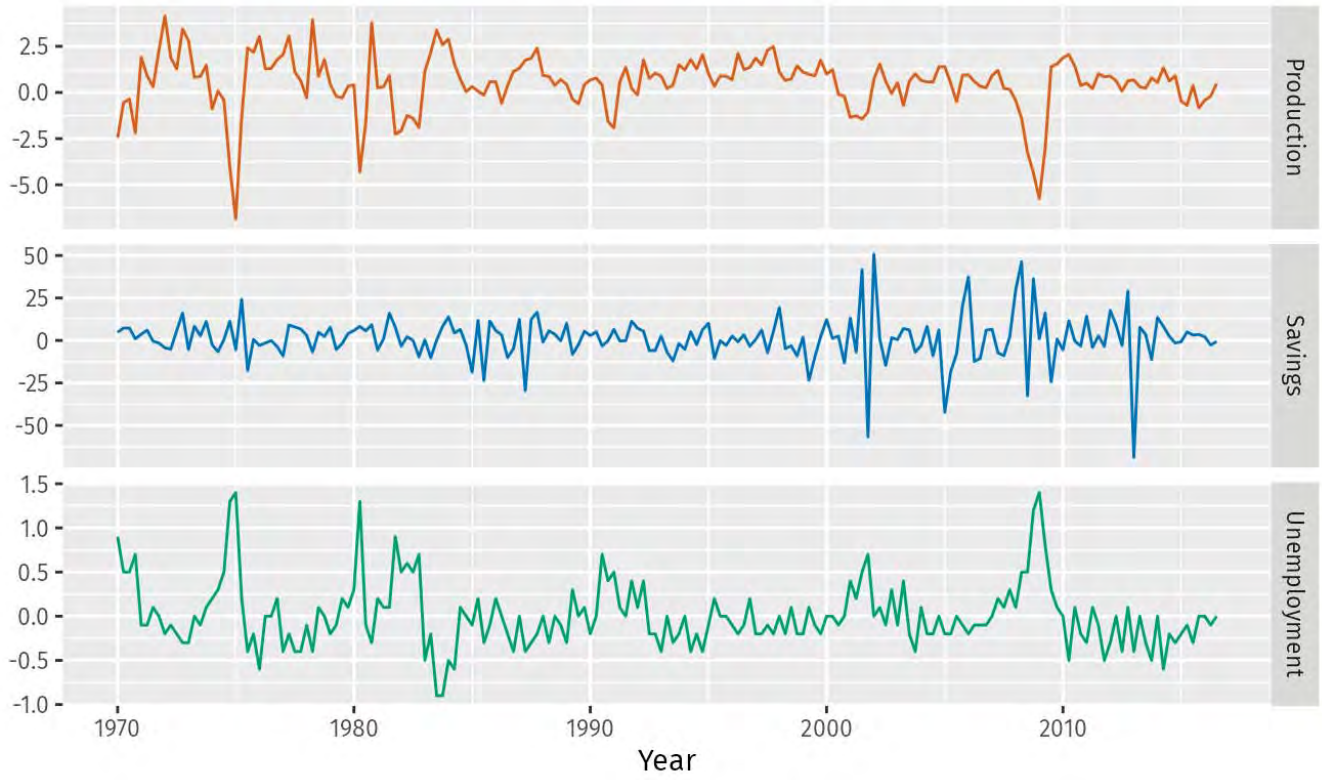


Figure 5.4: Quarterly percentage changes in industrial production and personal savings and quarterly changes in the unemployment rate for the US over the period 1970Q1-2016Q3.

Figure 5.5 is a scatterplot matrix of five variables. The first column shows the relationships between the forecast variable (consumption) and each of the predictors. The scatterplots show positive relationships with income and industrial production, and negative relationships with savings and unemployment. The strength of these relationships are shown by the correlation coefficients across the first row. The remaining scatterplots and correlation coefficients show the relationships between the predictors.

```
uschange %>%
  as.data.frame() %>%
  GGally::ggpairs()
```

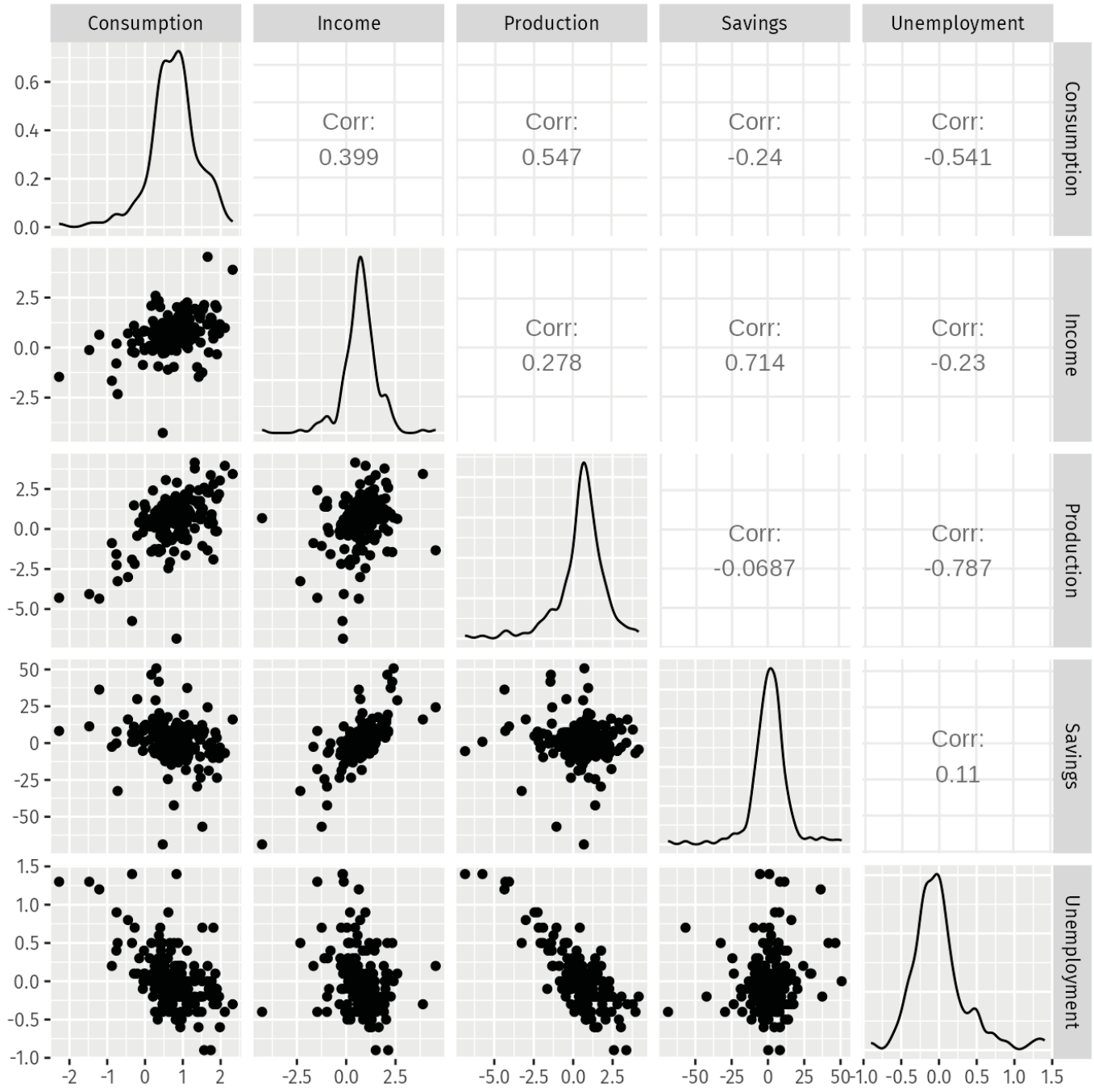


Figure 5.5: A scatterplot matrix of US consumption expenditure and the four predictors.

Assumptions

When we use a linear regression model, we are implicitly making some assumptions about the variables in Equation (5.1).

First, we assume that the model is a reasonable approximation to reality; that is, the relationship between the forecast variable and the predictor variables satisfies this linear equation.

Second, we make the following assumptions about the errors $(\varepsilon_1, \ldots, \varepsilon_T)$:

- they have mean zero; otherwise the forecasts will be systematically biased.

- they are not autocorrelated; otherwise the forecasts will be inefficient, as there is more information in the data that can be exploited.
- they are unrelated to the predictor variables; otherwise there would be more information that should be included in the systematic part of the model.

It is also useful to have the errors being normally distributed with a constant variance σ^2 in order to easily produce prediction intervals.

Another important assumption in the linear regression model is that each predictor x is not a random variable. If we were performing a controlled experiment in a laboratory, we could control the values of each x (so they would not be random) and observe the resulting values of y. With observational data (including most data in business and economics), it is not possible to control the value of x, we simply observe it. Hence we make this an assumption.

5.2 Least squares estimation

In practice, of course, we have a collection of observations but we do not know the values of the coefficients $\beta_0, \beta_1, \dots, \beta_k$. These need to be estimated from the data.

The least squares principle provides a way of choosing the coefficients effectively by minimising the sum of the squared errors. That is, we choose the values of $\beta_0, \beta_1, \dots, \beta_k$ that minimise

$$\sum_{t=1}^{T} \varepsilon_t^2 = \sum_{t=1}^{T} (y_t - \beta_0 - \beta_1 x_{1,t} - \beta_2 x_{2,t} - \cdots - \beta_k x_{k,t})^2.$$

This is called **least squares** estimation because it gives the least value for the sum of squared errors. Finding the best estimates of the coefficients is often called "fitting" the model to the data, or sometimes "learning" or "training" the model. The line shown in Figure 5.3 was obtained in this way.

When we refer to the *estimated* coefficients, we will use the notation $\hat{\beta}_0, \dots, \hat{\beta}_k$. The equations for these will be given in Section 5.7.

The `tslm()` function fits a linear regression model to time series data. It is similar to the `lm()` function which is widely used for linear models, but `tslm()` provides additional facilities for handling time series.

Example: US consumption expenditure

A multiple linear regression model for US consumption is

$$y_t = \beta_0 + \beta_1 x_{1,t} + \beta_2 x_{2,t} + \beta_3 x_{3,t} + \beta_4 x_{4,t} + \varepsilon_t,$$

where y is the percentage change in real personal consumption expenditure, x_1 is the percentage change in real personal disposable income, x_2 is the percentage change in industrial production, x_3 is the percentage change in personal savings and x_4 is the change in the unemployment rate.

The following output provides information about the fitted model. The first column of `Coefficients` gives an estimate of each β coefficient and the second column gives its standard error (i.e., the standard deviation which would be obtained from

repeatedly estimating the β coefficients on similar data sets). The standard error gives a measure of the uncertainty in the estimated β coefficient.

```
fit.consMR <- tslm(
  Consumption ~ Income + Production + Unemployment + Savings,
  data=uschange)
summary(fit.consMR)
#>
#> Call:
#> tslm(formula = Consumption ~ Income + Production + Unemployment +
#>     Savings, data = uschange)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -0.8830 -0.1764 -0.0368  0.1525  1.2055
#>
#> 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
#>
#> Residual standard error: 0.329 on 182 degrees of freedom
#> Multiple R-squared:  0.754,  Adjusted R-squared:  0.749
#> F-statistic:  139 on 4 and 182 DF,  p-value: <2e-16
```

For forecasting purposes, the final two columns are of limited interest. The "t value" is the ratio of an estimated β coefficient to its standard error and the last column gives the p-value: the probability of the estimated β coefficient being as large as it is if there was no real relationship between consumption and the corresponding predictor. This is useful when studying the effect of each predictor, but is not particularly useful for forecasting.

Fitted values

Predictions of y can be obtained by using the estimated coefficients in the regression equation and setting the error term to zero. In general we write,

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t} + \hat{\beta}_2 x_{2,t} + \cdots + \hat{\beta}_k x_{k,t}. \tag{5.2}$$

Plugging in the values of $x_{1,t}, \ldots, x_{k,t}$ for $t = 1, \ldots, T$ returns predictions of y_t within the training-sample, referred to as *fitted values*. Note that these are predictions of the data used to estimate the model, not genuine forecasts of future values of y.

The following plots show the actual values compared to the fitted values for the percentage change in the US consumption expenditure series. The time plot in Figure 5.6 shows that the fitted values follow the actual data fairly closely. This is verified by the strong positive relationship shown by the scatterplot in Figure 5.7.

```
autoplot(uschange[,'Consumption'], series="Data") +
  autolayer(fitted(fit.consMR), series="Fitted") +
  xlab("Year") + ylab("") +
  ggtitle("Percent change in US consumption expenditure") +
  guides(colour=guide_legend(title=" "))
```

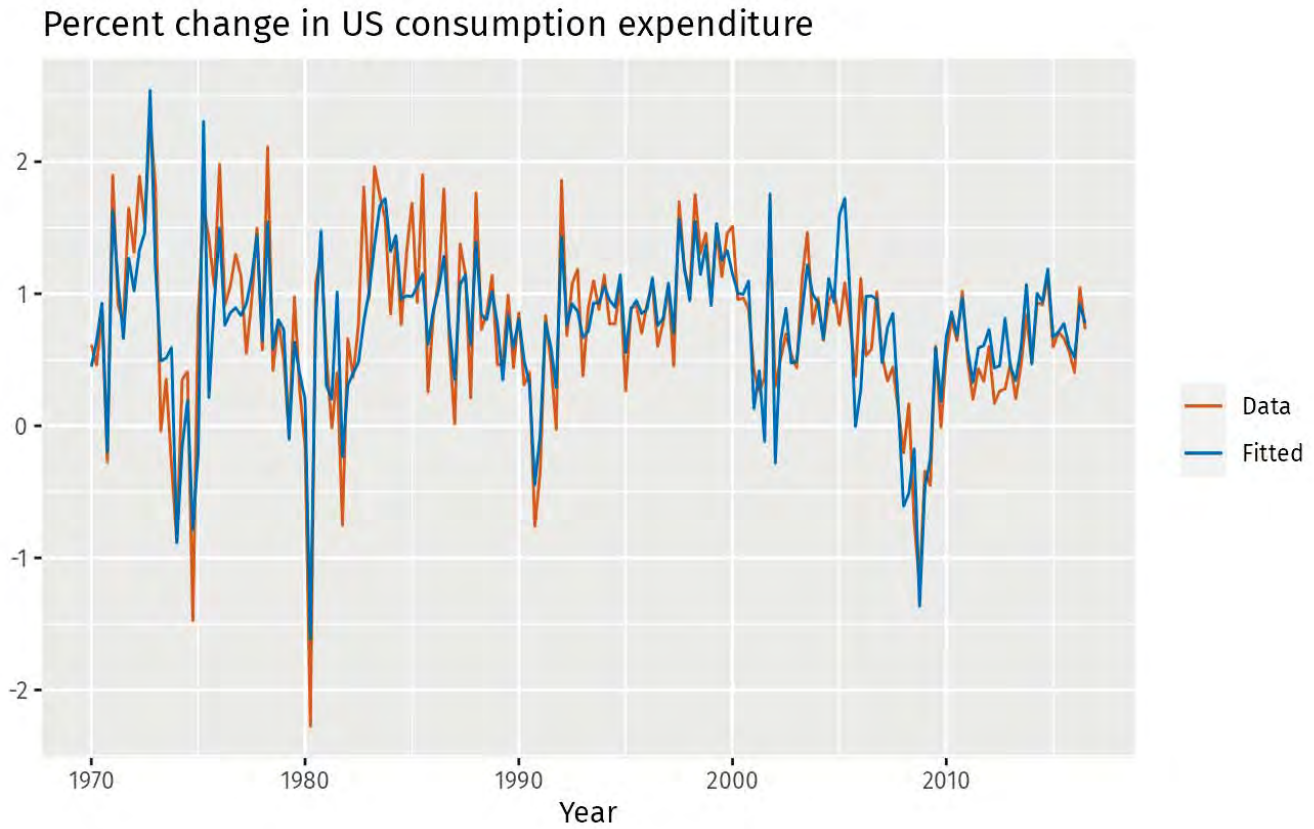


Figure 5.6: Time plot of actual US consumption expenditure and predicted US consumption expenditure.

```
cbind(Data = uschange[,"Consumption"],
      Fitted = fitted(fit.consMR)) %>%
  as.data.frame() %>%
  ggplot(aes(x=Data, y=Fitted)) +
    geom_point() +
    ylab("Fitted (predicted values)") +
    xlab("Data (actual values)") +
    ggtitle("Percent change in US consumption expenditure") +
    geom_abline(intercept=0, slope=1)
```

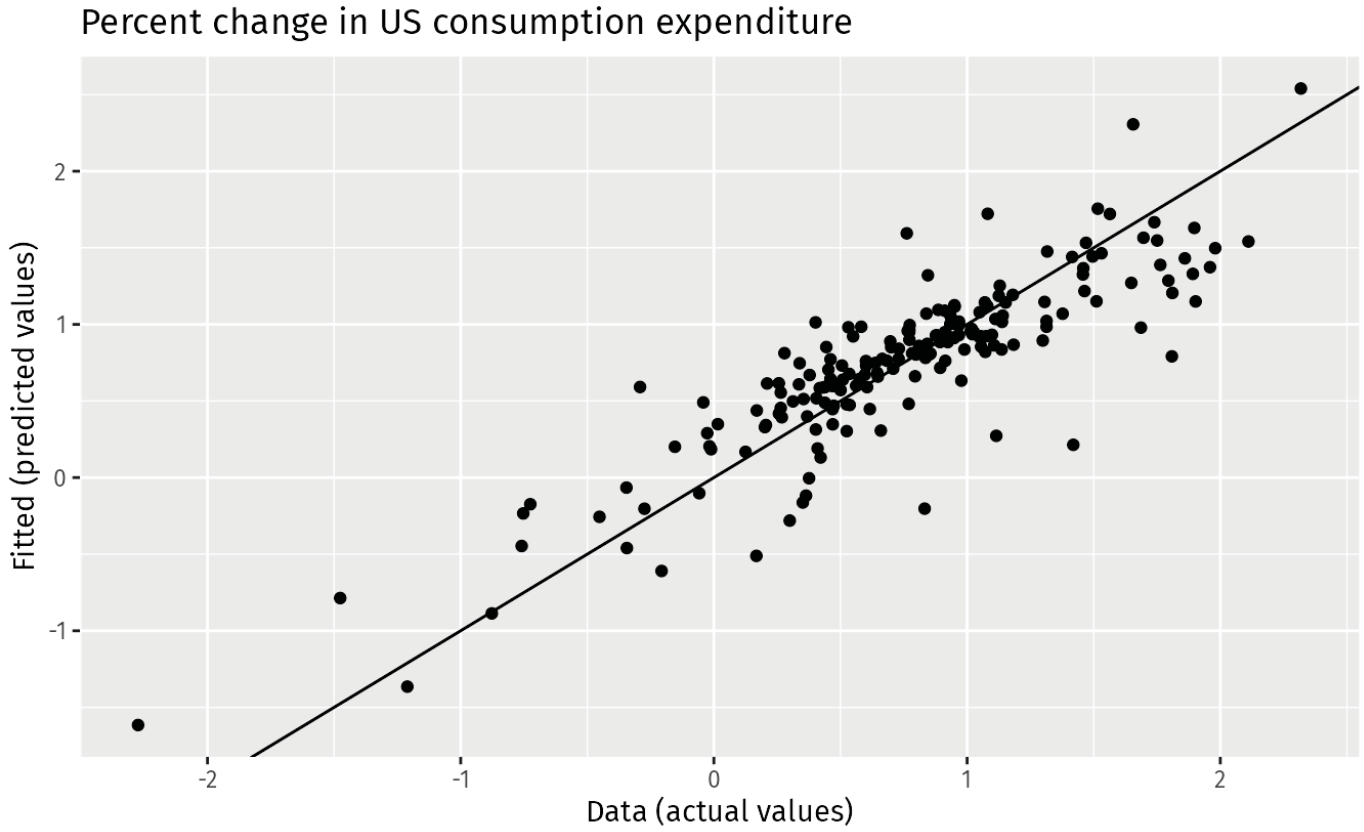


Figure 5.7: Actual US consumption expenditure plotted against predicted US consumption expenditure.

Goodness-of-fit

A common way to summarise how well a linear regression model fits the data is via the coefficient of determination, or R^2. This can be calculated as the square of the correlation between the observed y values and the predicted $\hat{y}$ values. Alternatively, it can also be calculated as,

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

where the summations are over all observations. Thus, it reflects the proportion of variation in the forecast variable that is accounted for (or explained) by the regression model.

In simple linear regression, the value of R^2 is also equal to the square of the correlation between y and x (provided an intercept has been included).

If the predictions are close to the actual values, we would expect R^2 to be close to 1. On the other hand, if the predictions are unrelated to the actual values, then $R^2 = 0$ (again, assuming there is an intercept). In all cases, R^2 lies between 0 and 1.

The R^2 value is used frequently, though often incorrectly, in forecasting. The value of R^2 will never decrease when adding an extra predictor to the model and this can lead to over-fitting. There are no set rules for what is a good R^2 value, and typical values of R^2 depend on the type of data used. Validating a model's forecasting performance on the test data is much better than measuring the R^2 value on the training data.

Example: US consumption expenditure

Figure 5.7 plots the actual consumption expenditure values versus the fitted values. The correlation between these variables is $r = 0.868$ hence $R^2 = 0.754$ (shown in the output above). In this case model does an excellent job as it explains 75.4% of the variation in the consumption data. Compare that to the R^2 value of 0.16 obtained from the simple regression with the same data set in Section 5.1. Adding the three extra predictors has allowed a lot more of the variation in the consumption data to be explained.

Standard error of the regression

Another measure of how well the model has fitted the data is the standard deviation of the residuals, which is often known as the "residual standard error". This is shown in the above output with the value 0.329.

It is calculated using

$$\hat{\sigma}_e = \sqrt{\frac{1}{T-k-1}\sum_{t=1}^{T} e_t^2}, \tag{5.3}$$

where k is the number of predictors in the model. Notice that we divide by $T-k-1$ because we have estimated $k+1$ parameters (the intercept and a coefficient for each predictor variable) in computing the residuals.

The standard error is related to the size of the average error that the model produces. We can compare this error to the sample mean of y or with the standard deviation of y to gain some perspective on the accuracy of the model.

The standard error will be used when generating prediction intervals, discussed in Section 5.6.

5.3 Evaluating the regression model

The differences between the observed y values and the corresponding fitted $\hat{y}$ values are the training-set errors or "residuals" defined as,

$$\begin{aligned} e_t &= y_t - \hat{y}_t \\ &= y_t - \hat{\beta}_0 - \hat{\beta}_1 x_{1,t} - \hat{\beta}_2 x_{2,t} - \cdots - \hat{\beta}_k x_{k,t} \end{aligned}$$

for $t = 1, \ldots, T$. Each residual is the unpredictable component of the associated observation.

The residuals have some useful properties including the following two:

$$\sum_{t=1}^{T} e_t = 0 \quad \text{and} \quad \sum_{t=1}^{T} x_{k,t} e_t = 0 \qquad \text{for all } k.$$

As a result of these properties, it is clear that the average of the residuals is zero, and that the correlation between the residuals and the observations for the predictor variable is also zero. (This is not necessarily true when the intercept is omitted from the model.)

After selecting the regression variables and fitting a regression model, it is necessary to plot the residuals to check that the assumptions of the model have been satisfied. There are a series of plots that should be produced in order to check different aspects of the fitted model and the underlying assumptions. We will now discuss each of them in turn.

ACF plot of residuals

With time series data, it is highly likely that the value of a variable observed in the current time period will be similar to its value in the previous period, or even the period before that, and so on. Therefore when fitting a regression model to time series data, it is common to find autocorrelation in the residuals. In this case, the estimated model violates the assumption of no autocorrelation in the errors, and our forecasts may be inefficient — there is some information left over which should be accounted for in the model in order to obtain better forecasts. The forecasts from a

model with autocorrelated errors are still unbiased, and so they are not "wrong", but they will usually have larger prediction intervals than they need to. Therefore we should always look at an ACF plot of the residuals.

Another useful test of autocorrelation in the residuals designed to take account for the regression model is the **Breusch-Godfrey** test, also referred to as the LM (Lagrange Multiplier) test for serial correlation. It is used to test the joint hypothesis that there is no autocorrelation in the residuals up to a certain specified order. A small p-value indicates there is significant autocorrelation remaining in the residuals.

The Breusch-Godfrey test is similar to the Ljung-Box test, but it is specifically designed for use with regression models.

Histogram of residuals

It is always a good idea to check whether the residuals are normally distributed. As we explained earlier, this is not essential for forecasting, but it does make the calculation of prediction intervals much easier.

Example

Using the `checkresiduals()` function introduced in Section 3.3, we can obtain all the useful residual diagnostics mentioned above.

```
checkresiduals(fit.consMR)
```

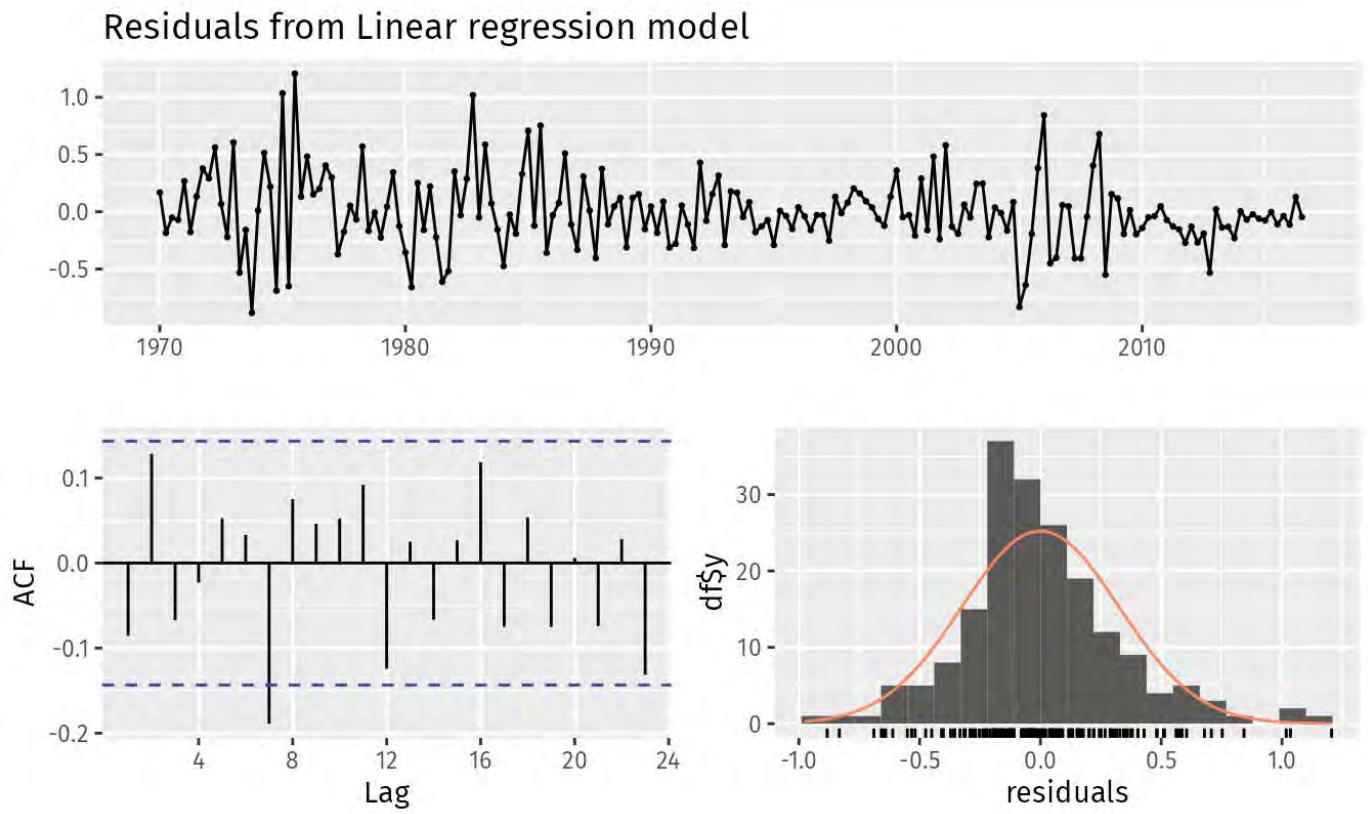


Figure 5.8: Analysing the residuals from a regression model for US quarterly consumption.

```
#>
#>  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
```

Figure 5.8 shows a time plot, the ACF and the histogram of the residuals from the multiple regression model fitted to the US quarterly consumption data, as well as the Breusch-Godfrey test for jointly testing up to 8th order autocorrelation. (The `checkresiduals()` function will use the Breusch-Godfrey test for regression models, but the Ljung-Box test otherwise.)

The time plot shows some changing variation over time, but is otherwise relatively unremarkable. This heteroscedasticity will potentially make the prediction interval coverage inaccurate.

The histogram shows that the residuals seem to be slightly skewed, which may also affect the coverage probability of the prediction intervals.

The autocorrelation plot shows a significant spike at lag 7, but it is not quite enough for the Breusch-Godfrey to be significant at the 5% level. In any case, the autocorrelation is not particularly large, and at lag 7 it is unlikely to have any

noticeable impact on the forecasts or the prediction intervals. In Chapter 9 we discuss dynamic regression models used for better capturing information left in the residuals.

Residual plots against predictors

We would expect the residuals to be randomly scattered without showing any systematic patterns. A simple and quick way to check this is to examine scatterplots of the residuals against each of the predictor variables. If these scatterplots show a pattern, then the relationship may be nonlinear and the model will need to be modified accordingly. See Section 5.8 for a discussion of nonlinear regression.

It is also necessary to plot the residuals against any predictors that are *not* in the model. If any of these show a pattern, then the corresponding predictor may need to be added to the model (possibly in a nonlinear form).

Example

The residuals from the multiple regression model for forecasting US consumption plotted against each predictor in Figure 5.9 seem to be randomly scattered. Therefore we are satisfied with these in this case.

```
df <- as.data.frame(uschange)
df[,"Residuals"]  <- as.numeric(residuals(fit.consMR))
p1 <- ggplot(df, aes(x=Income, y=Residuals)) +
  geom_point()
p2 <- ggplot(df, aes(x=Production, y=Residuals)) +
  geom_point()
p3 <- ggplot(df, aes(x=Savings, y=Residuals)) +
  geom_point()
p4 <- ggplot(df, aes(x=Unemployment, y=Residuals)) +
  geom_point()
gridExtra::grid.arrange(p1, p2, p3, p4, nrow=2)
```

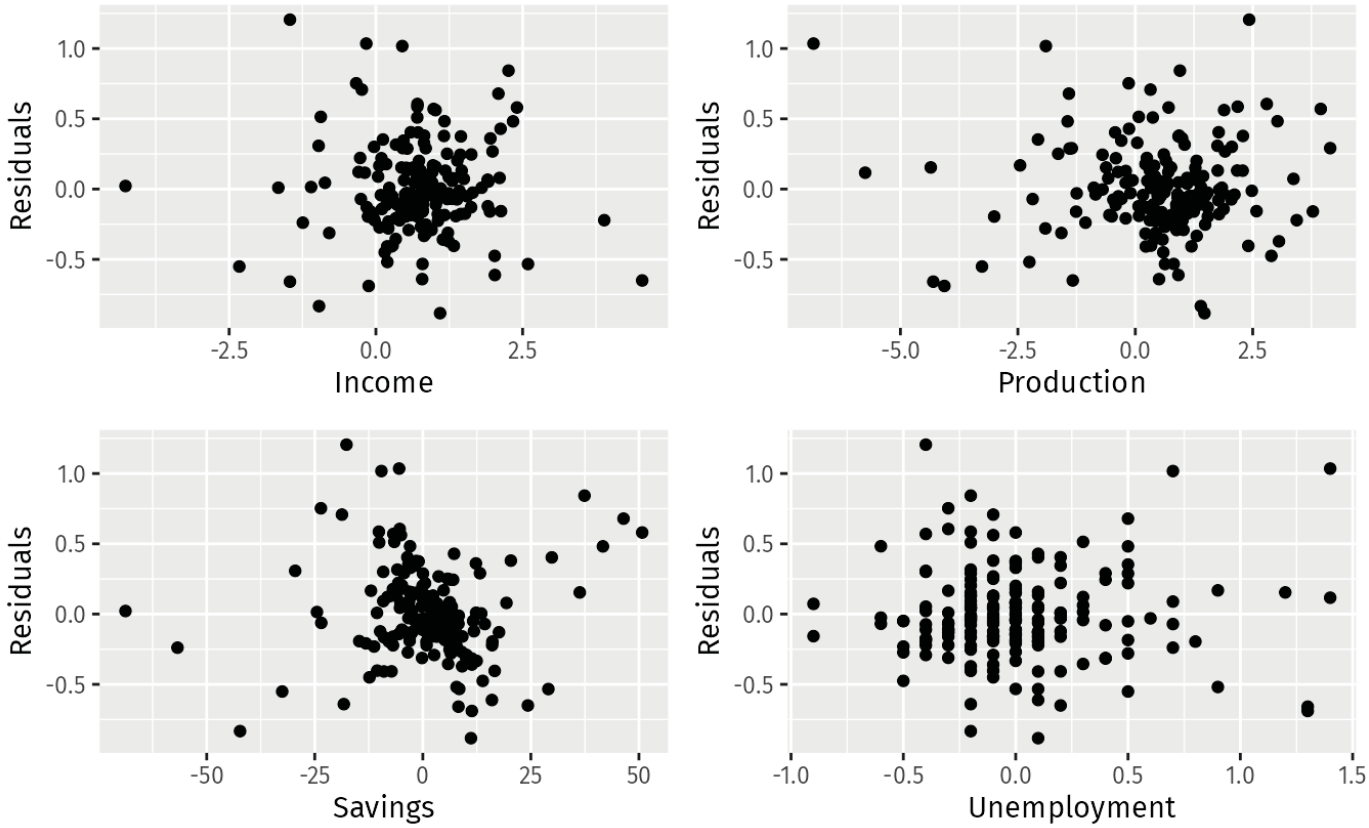


Figure 5.9: Scatterplots of residuals versus each predictor.

Residual plots against fitted values

A plot of the residuals against the fitted values should also show no pattern. If a pattern is observed, there may be "heteroscedasticity" in the errors which means that the variance of the residuals may not be constant. If this problem occurs, a transformation of the forecast variable such as a logarithm or square root may be required (see Section 3.2.)

Example

Continuing the previous example, Figure 5.10 shows the residuals plotted against the fitted values. The random scatter suggests the errors are homoscedastic.

```
cbind(Fitted = fitted(fit.consMR),
      Residuals=residuals(fit.consMR)) %>%
  as.data.frame() %>%
  ggplot(aes(x=Fitted, y=Residuals)) + geom_point()
```

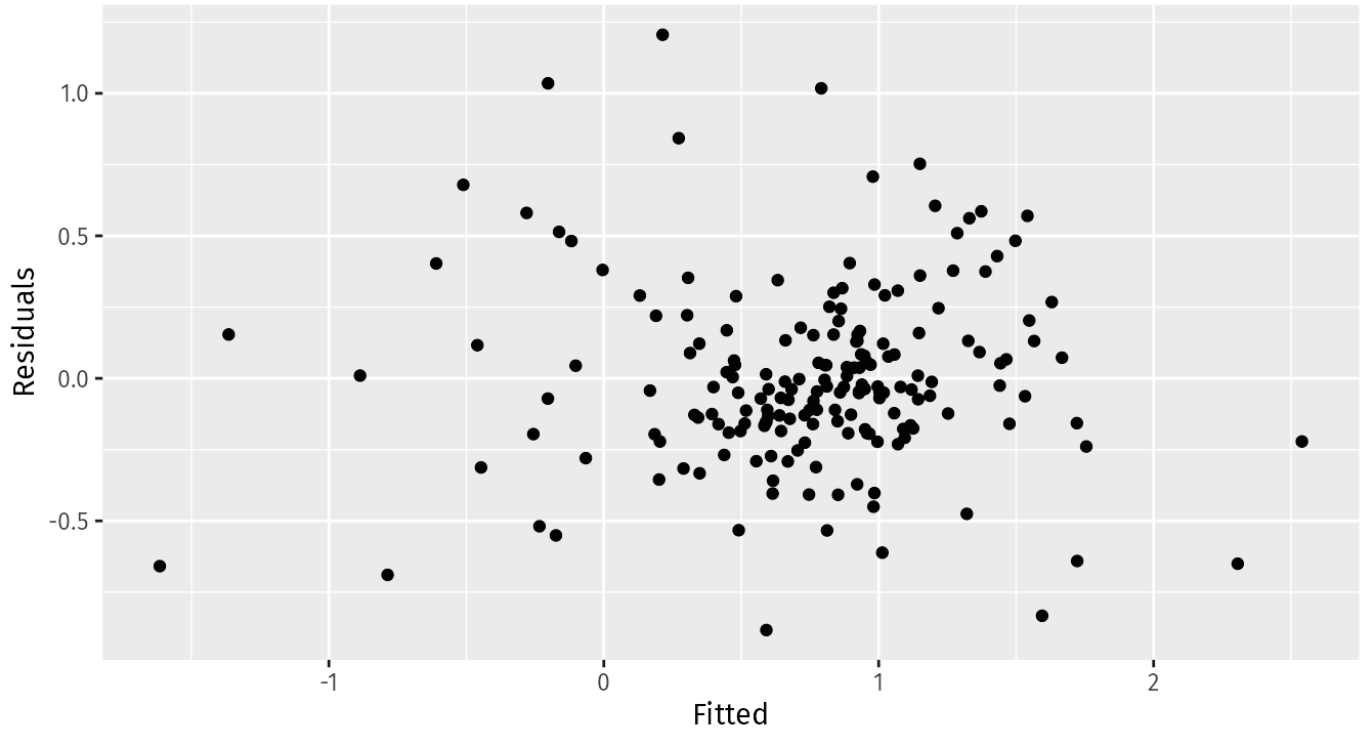


Figure 5.10: Scatterplots of residuals versus fitted values.

Outliers and influential observations

Observations that take extreme values compared to the majority of the data are called **outliers**. Observations that have a large influence on the estimated coefficients of a regression model are called **influential observations**. Usually, influential observations are also outliers that are extreme in the x direction.

There are formal methods for detecting outliers and influential observations that are beyond the scope of this textbook. As we suggested at the beginning of Chapter 2, becoming familiar with your data prior to performing any analysis is of vital importance. A scatter plot of y against each x is always a useful starting point in regression analysis, and often helps to identify unusual observations.

One source of outliers is incorrect data entry. Simple descriptive statistics of your data can identify minima and maxima that are not sensible. If such an observation is identified, and it has been recorded incorrectly, it should be corrected or removed from the sample immediately.

Outliers also occur when some observations are simply different. In this case it may not be wise for these observations to be removed. If an observation has been identified as a likely outlier, it is important to study it and analyse the possible

reasons behind it. The decision to remove or retain an observation can be a challenging one (especially when outliers are influential observations). It is wise to report results both with and without the removal of such observations.

Example

Figure 5.11 highlights the effect of a single outlier when regressing US consumption on income (the example introduced in Section 5.1). In the left panel the outlier is only extreme in the direction of y, as the percentage change in consumption has been incorrectly recorded as -4%. The red line is the regression line fitted to the data which includes the outlier, compared to the black line which is the line fitted to the data without the outlier. In the right panel the outlier now is also extreme in the direction of x with the 4% decrease in consumption corresponding to a 6% increase in income. In this case the outlier is extremely influential as the red line now deviates substantially from the black line.

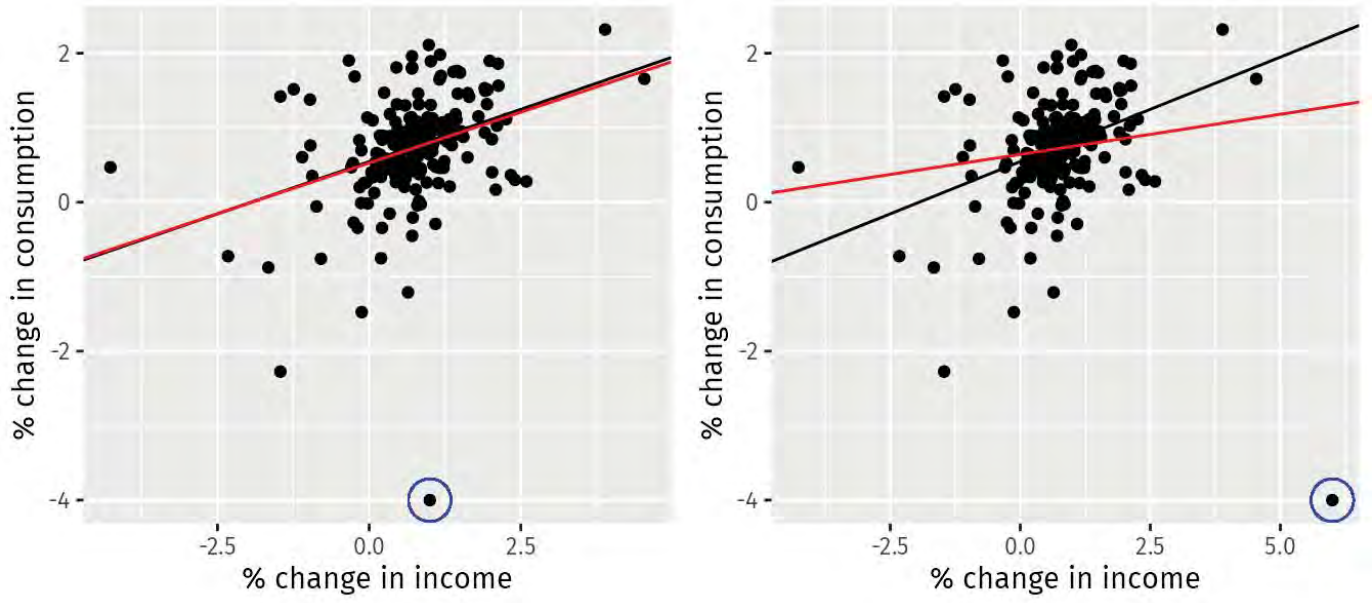


Figure 5.11: The effect of outliers and influential observations on regression

Spurious regression

More often than not, time series data are "non-stationary"; that is, the values of the time series do not fluctuate around a constant mean or with a constant variance. We will deal with time series stationarity in more detail in Chapter 8, but here we need to address the effect that non-stationary data can have on regression models.

For example, consider the two variables plotted in Figure 5.12. These appear to be related simply because they both trend upwards in the same manner. However, air passenger traffic in Australia has nothing to do with rice production in Guinea.

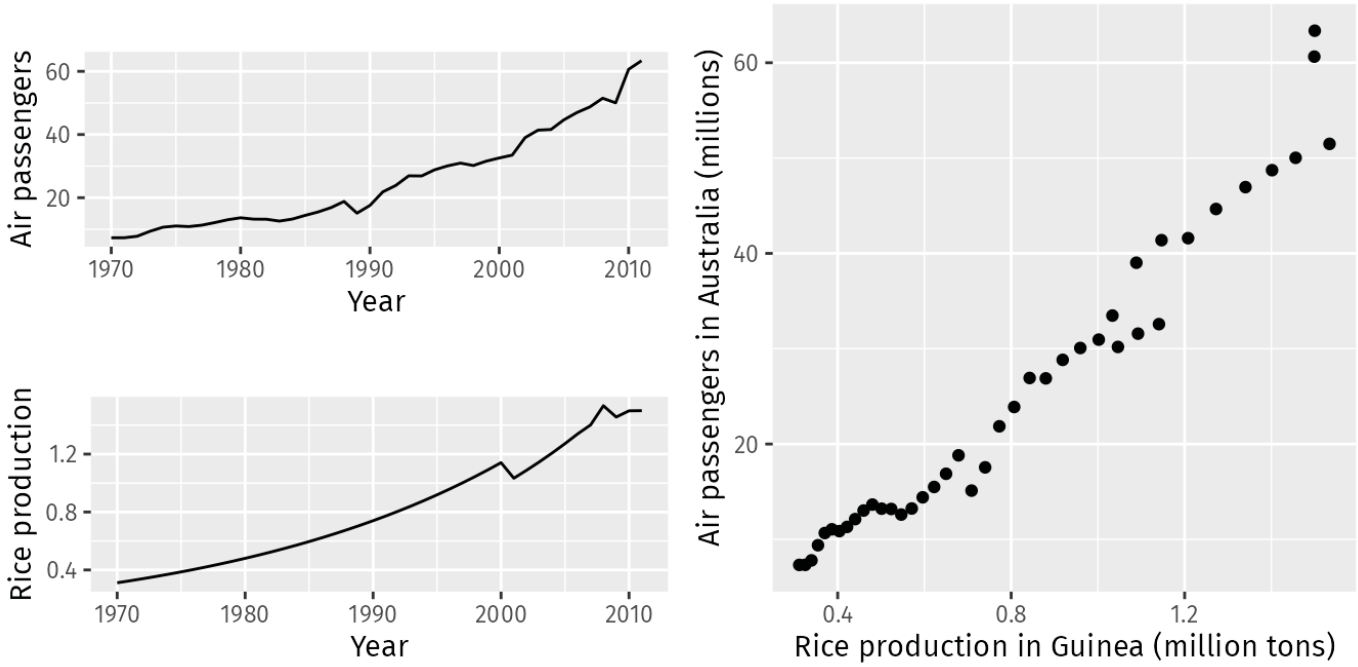


Figure 5.12: Trending time series data can appear to be related, as shown in this example where air passengers in Australia are regressed against rice production in Guinea.

Regressing non-stationary time series can lead to spurious regressions. The output of regressing Australian air passengers on rice production in Guinea is shown in Figure 5.13. High R^2 and high residual autocorrelation can be signs of spurious regression. Notice these features in the output below. We discuss the issues surrounding non-stationary data and spurious regressions in more detail in Chapter 9.

Cases of spurious regression might appear to give reasonable short-term forecasts, but they will generally not continue to work into the future.

```r
aussies <- window(ausair, end=2011)
fit <- tslm(aussies ~ guinearice)
summary(fit)
#>
#> Call:
#> tslm(formula = aussies ~ guinearice)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -5.945 -1.892 -0.327  1.862 10.421
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    -7.49       1.20   -6.23  2.3e-07 ***
#> guinearice     40.29       1.34   30.13  < 2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 3.24 on 40 degrees of freedom
#> Multiple R-squared:  0.958,  Adjusted R-squared:  0.957
#> F-statistic:  908 on 1 and 40 DF,  p-value: <2e-16

checkresiduals(fit)
```

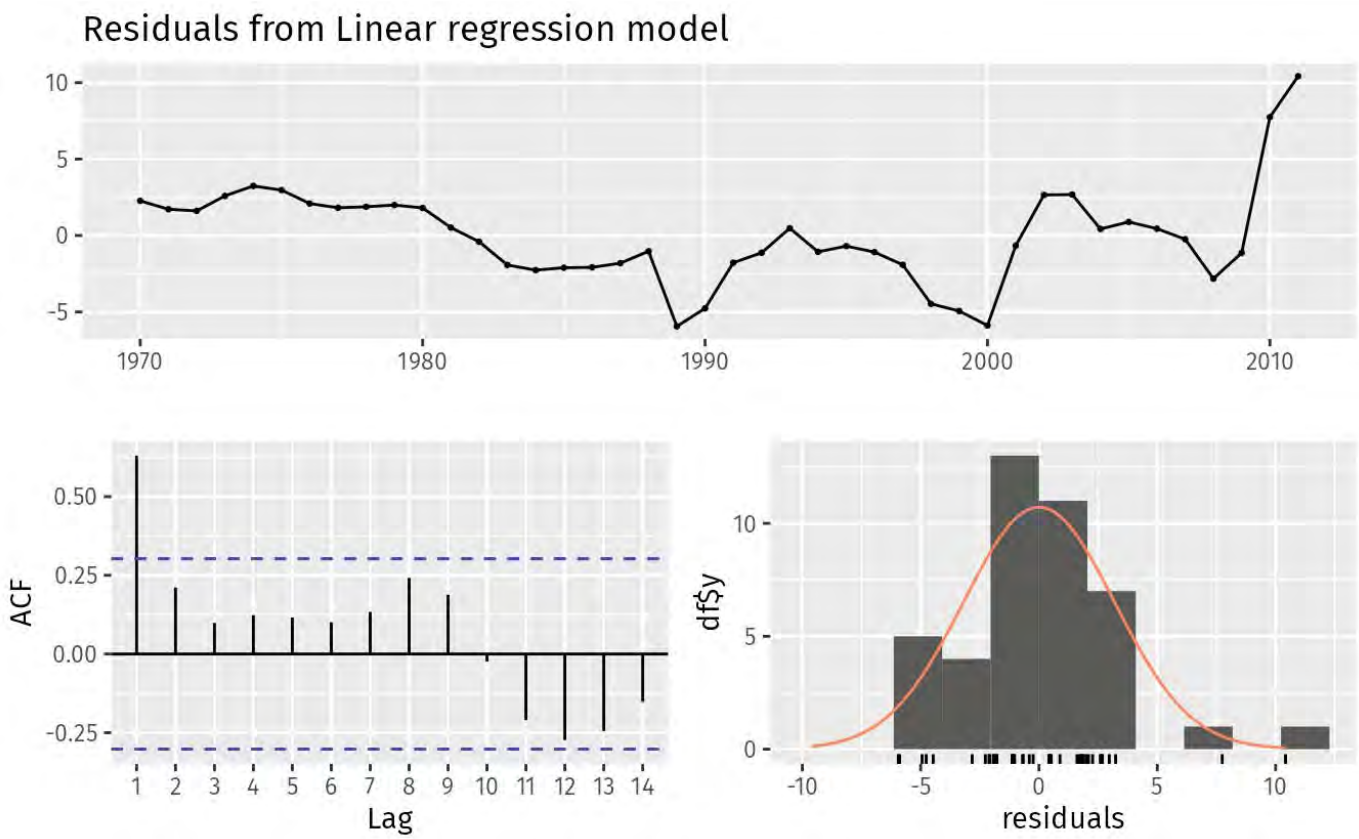


Figure 5.13: Residuals from a spurious regression.

```
#>
#>  Breusch-Godfrey test for serial correlation of order up to 8
#>
#> data:  Residuals from Linear regression model
#> LM test = 29, df = 8, p-value = 3e-04
```

5.4 Some useful predictors

There are several useful predictors that occur frequently when using regression for time series data.

Trend

It is common for time series data to be trending. A linear trend can be modelled by simply using $x_{1,t} = t$ as a predictor,

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

where $t = 1, \dots, T$. A trend variable can be specified in the `tslm()` function using the `trend` predictor. In Section 5.8 we discuss how we can also model a nonlinear trends.

Dummy variables

So far, we have assumed that each predictor takes numerical values. But what about when a predictor is a categorical variable taking only two values (e.g., "yes" and "no")? Such a variable might arise, for example, when forecasting daily sales and you want to take account of whether the day is a **public holiday** or not. So the predictor takes value "yes" on a public holiday, and "no" otherwise.

This situation can still be handled within the framework of multiple regression models by creating a "dummy variable" which takes value 1 corresponding to "yes" and 0 corresponding to "no". A dummy variable is also known as an "indicator variable".

A dummy variable can also be used to account for an **outlier** in the data. Rather than omit the outlier, a dummy variable removes its effect. In this case, the dummy variable takes value 1 for that observation and 0 everywhere else. An example is the case where a special event has occurred. For example when forecasting tourist arrivals to Brazil, we will need to account for the effect of the Rio de Janeiro summer Olympics in 2016.

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). `tslm()` will automatically handle this case if you specify a factor variable as a predictor. There is usually no need to manually create the corresponding dummy variables.

Seasonal dummy variables

Suppose that we are forecasting daily data and we want to account for the day of the week as a predictor. Then the following dummy variables can be created.

	$d_{1,t}$	$d_{2,t}$	$d_{3,t}$	$d_{4,t}$	$d_{5,t}$	$d_{6,t}$
Monday	1	0	0	0	0	0
Tuesday	0	1	0	0	0	0
Wednesday	0	0	1	0	0	0
Thursday	0	0	0	1	0	0
Friday	0	0	0	0	1	0
Saturday	0	0	0	0	0	1
Sunday	0	0	0	0	0	0
Monday	1	0	0	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮

Notice that only six dummy variables are needed to code seven categories. That is because the seventh category (in this case Sunday) is captured by the intercept, and is specified when the dummy variables are all set to zero.

Many beginners will try to add a seventh dummy variable for the seventh category. This is known as the "dummy variable trap", because it will cause the regression to fail. There will be one too many parameters to estimate when an intercept is also included. The general rule is to use one fewer dummy variables than categories. So for quarterly data, use three dummy variables; for monthly data, use 11 dummy variables; and for daily data, use six dummy variables, and so on.

The interpretation of each of the coefficients associated with the dummy variables is that it is *a measure of the effect of that category relative to the omitted category*. In the above example, the coefficient of $d_{1,t}$ associated with Monday will measure the effect of Monday on the forecast variable compared to the effect of Sunday. An example of interpreting estimated dummy variable coefficients capturing the quarterly seasonality of Australian beer production follows.

The `tslm()` function will automatically handle this situation if you specify the predictor `season`.

Example: Australian quarterly beer production

Recall the Australian quarterly beer production data shown again in Figure 5.14.

```
beer2 <- window(ausbeer, start=1992)
autoplot(beer2) + xlab("Year") + ylab("Megalitres")
```

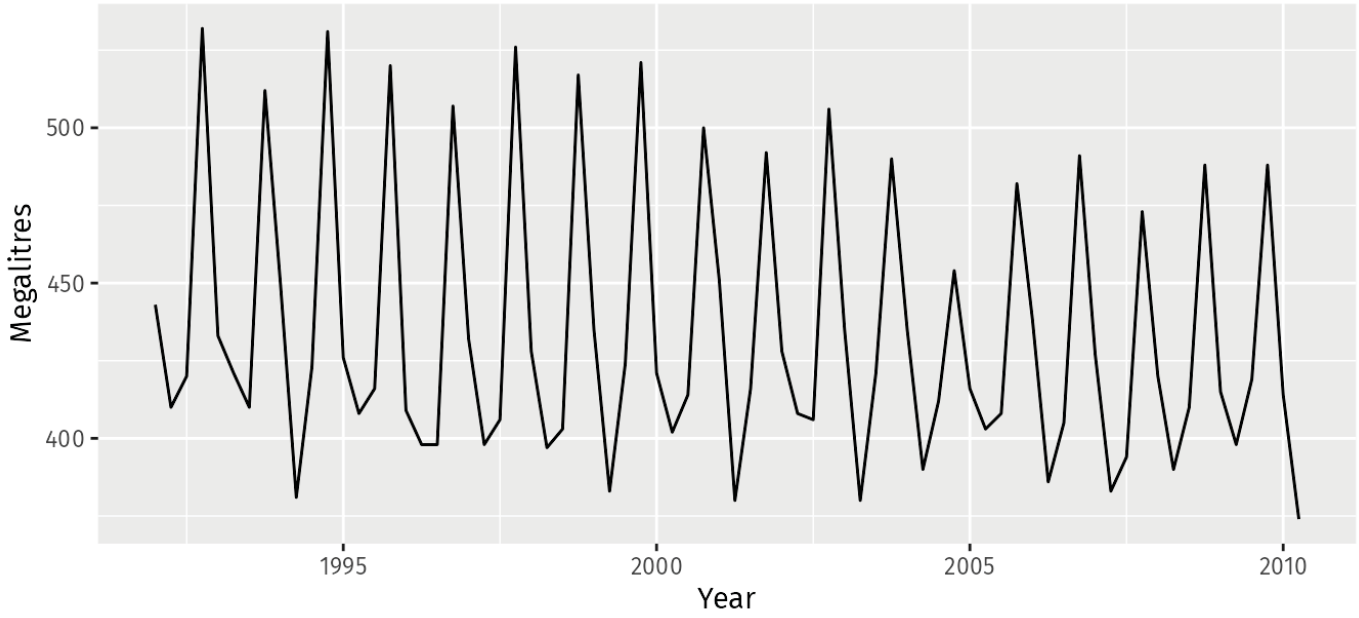


Figure 5.14: Australian quarterly beer production.

We want to forecast the value of future beer production. We can model this data using a regression model with a linear trend and quarterly dummy variables,

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

where $d_{i,t} = 1$ if t is in quarter i and 0 otherwise. The first quarter variable has been omitted, so the coefficients associated with the other quarters are measures of the difference between those quarters and the first quarter.

```
fit.beer <- tslm(beer2 ~ trend + season)
summary(fit.beer)
#>
#> Call:
#> tslm(formula = beer2 ~ trend + season)
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -42.90  -7.60  -0.46   7.99  21.79
#>
#> 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
#>
#> 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
```

Note that `trend` and `season` are not objects in the R workspace; they are created automatically by `tslm()` when specified in this way.

There is an average downward trend of -0.34 megalitres per quarter. On average, the second quarter has production of 34.7 megalitres lower than the first quarter, the third quarter has production of 17.8 megalitres lower than the first quarter, and the fourth quarter has production of 72.8 megalitres higher than the first quarter.

```
autoplot(beer2, series="Data") +
  autolayer(fitted(fit.beer), series="Fitted") +
  xlab("Year") + ylab("Megalitres") +
  ggtitle("Quarterly Beer Production")
```

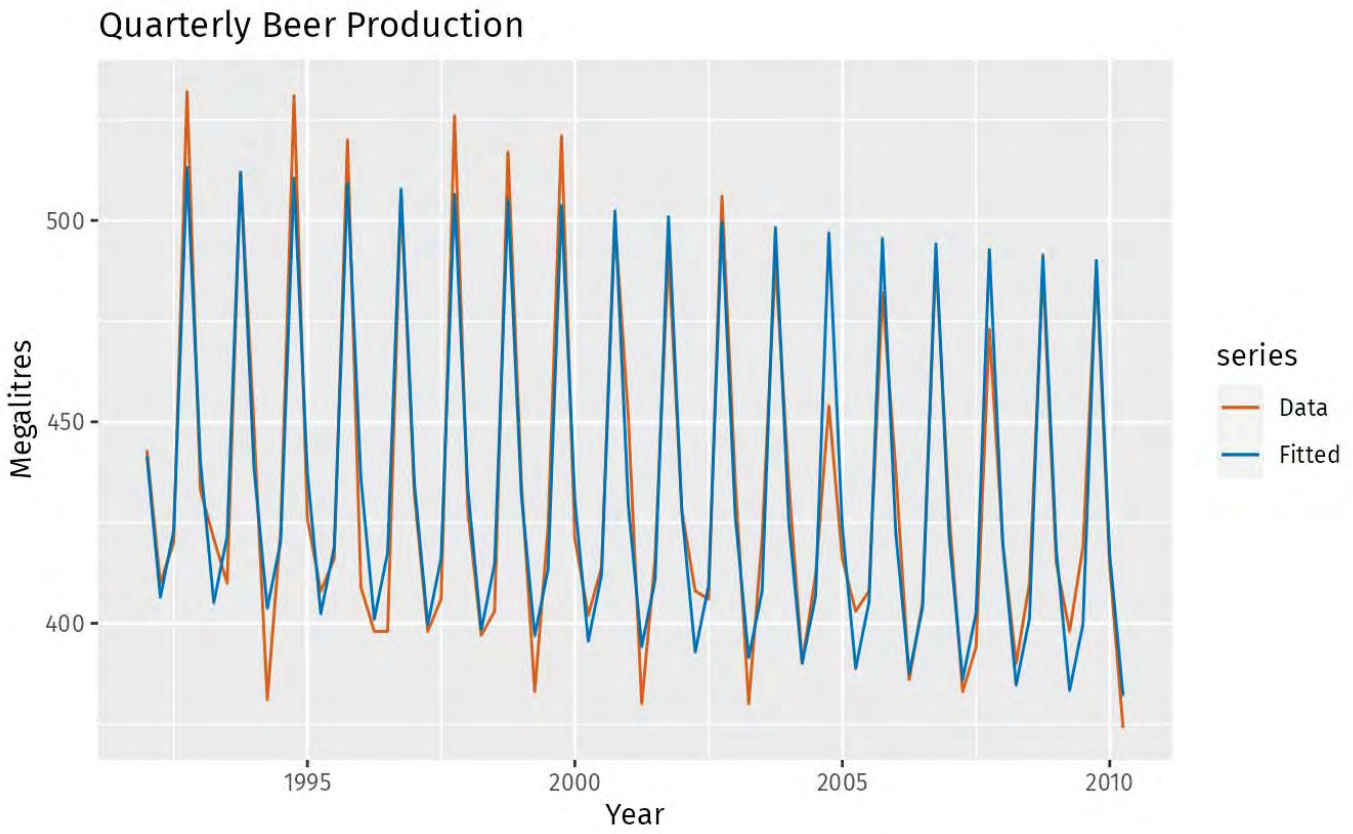


Figure 5.15: Time plot of beer production and predicted beer production.

```
cbind(Data=beer2, Fitted=fitted(fit.beer)) %>%
  as.data.frame() %>%
  ggplot(aes(x = Data, y = Fitted,
             colour = as.factor(cycle(beer2)))) +
    geom_point() +
    ylab("Fitted") + xlab("Actual values") +
    ggtitle("Quarterly beer production") +
    scale_colour_brewer(palette="Dark2", name="Quarter") +
    geom_abline(intercept=0, slope=1)
```

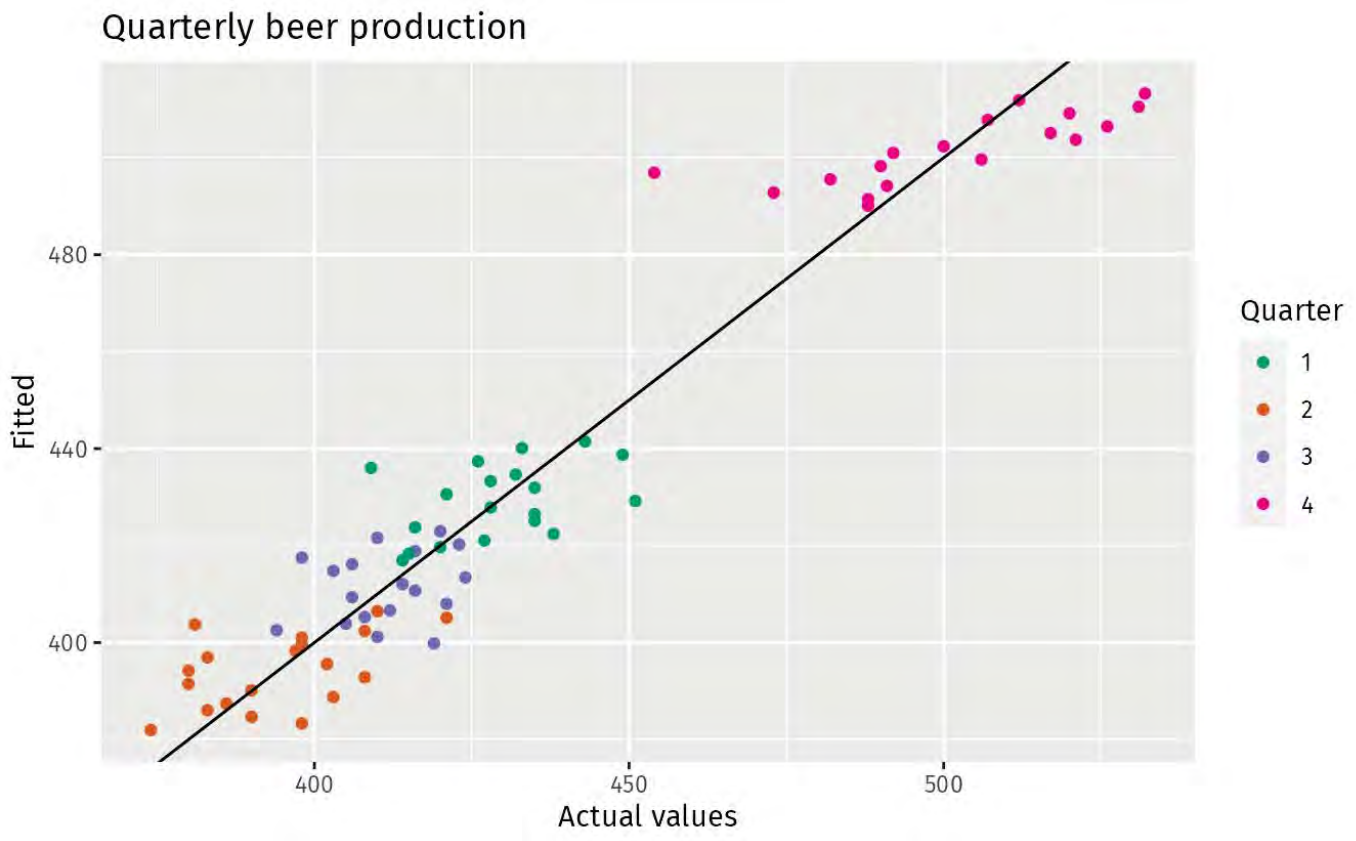


Figure 5.16: Actual beer production plotted against predicted beer production.

Intervention variables

It is often necessary to model interventions that may have affected the variable to be forecast. For example, competitor activity, advertising expenditure, industrial action, and so on, can all have an effect.

When the effect lasts only for one period, we use a "spike" variable. This is a dummy variable that takes value one in the period of the intervention and zero elsewhere. A spike variable is equivalent to a dummy variable for handling an outlier.

Other interventions have an immediate and permanent effect. If an intervention causes a level shift (i.e., the value of the series changes suddenly and permanently from the time of intervention), then we use a "step" variable. A step variable takes value zero before the intervention and one from the time of intervention onward.

Another form of permanent effect is a change of slope. Here the intervention is handled using a piecewise linear trend; a trend that bends at the time of intervention and hence is nonlinear. We will discuss this in Section 5.8.

Trading days

The number of trading days in a month can vary considerably and can have a substantial effect on sales data. To allow for this, the number of trading days in each month can be included as a predictor.

For monthly or quarterly data, the `bizdays()` function will compute the number of trading days in each period.

An alternative that allows for the effects of different days of the week has the following predictors:

$$\begin{aligned} x_1 &= \text{number of Mondays in month;} \\ x_2 &= \text{number of Tuesdays in month;} \\ &\vdots \\ x_7 &= \text{number of Sundays in month.} \end{aligned}$$

Distributed lags

It is often useful to include advertising expenditure as a predictor. However, since the effect of advertising can last beyond the actual campaign, we need to include lagged values of advertising expenditure. Thus, the following predictors may be used.

$$\begin{aligned} x_1 &= \text{advertising for previous month;} \\ x_2 &= \text{advertising for two months previously;} \\ &\vdots \\ x_m &= \text{advertising for } m \text{ months previously.} \end{aligned}$$

It is common to require the coefficients to decrease as the lag increases, although this is beyond the scope of this book.

Easter

Easter differs from most holidays because it is not held on the same date each year, and its effect can last for several days. In this case, a dummy variable can be used with value one where the holiday falls in the particular time period and zero otherwise.

With monthly data, if Easter falls in March then the dummy variable takes value 1 in March, and if it falls in April the dummy variable takes value 1 in April. When Easter starts in March and finishes in April, the dummy variable is split proportionally between months.

The `easter()` function will compute the dummy variable for you.

Fourier series

An alternative to using seasonal dummy variables, especially for long seasonal periods, is to use Fourier terms. Jean-Baptiste Fourier was a French mathematician, born in the 1700s, who showed that a series of sine and cosine terms of the right frequencies can approximate any periodic function. We can use them for seasonal patterns.

If m is the seasonal period, then the first few Fourier terms are given by

$$x_{1,t} = \sin\left(\tfrac{2\pi t}{m}\right), x_{2,t} = \cos\left(\tfrac{2\pi t}{m}\right), x_{3,t} = \sin\left(\tfrac{4\pi t}{m}\right),$$

$$x_{4,t} = \cos\left(\tfrac{4\pi t}{m}\right), x_{5,t} = \sin\left(\tfrac{6\pi t}{m}\right), x_{6,t} = \cos\left(\tfrac{6\pi t}{m}\right),$$

and so on. If we have monthly seasonality, and we use the first 11 of these predictor variables, then we will get exactly the same forecasts as using 11 dummy variables.

With Fourier terms, we often need fewer predictors than with dummy variables, especially when m is large. This makes them useful for weekly data, for example, where $m \approx 52$. For short seasonal periods (e.g., quarterly data), there is little advantage in using Fourier terms over seasonal dummy variables.

These Fourier terms are produced using the `fourier()` function. For example, the Australian beer data can be modelled like this.

```
fourier.beer <- tslm(beer2 ~ trend + fourier(beer2, K=2))
summary(fourier.beer)
#>
#> Call:
#> tslm(formula = beer2 ~ trend + fourier(beer2, K = 2))
#>
#> Residuals:
#>    Min     1Q Median     3Q    Max
#> -42.90  -7.60  -0.46   7.99  21.79
#>
#> 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(beer2, K = 2)S1-4  8.9108     2.0112    4.43  3.4e-05 ***
#> fourier(beer2, K = 2)C1-4 53.7281     2.0112   26.71  < 2e-16 ***
#> fourier(beer2, 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
```

The first argument to `fourier()` allows it to identify the seasonal period m and the length of the predictors to return. The second argument `K` specifies how many pairs of sin and cos terms to include. The maximum allowed is $K = m/2$ where m is the seasonal period. Because we have used the maximum here, the results are identical to those obtained when using seasonal dummy variables.

If only the first two Fourier terms are used ($x_{1,t}$ and $x_{2,t}$), the seasonal pattern will follow a simple sine wave. A regression model containing Fourier terms is often called a **harmonic regression** because the successive Fourier terms represent harmonics of the first two Fourier terms.

5.5 Selecting predictors

When there are many possible predictors, we need some strategy for selecting the best predictors to use in a regression model.

A common approach that is *not recommended* is to plot the forecast variable against a particular predictor and if there is no noticeable relationship, drop that predictor from the model. This is invalid because it is not always possible to see the relationship from a scatterplot, especially when the effects of other predictors have not been accounted for.

Another common approach which is also invalid is to do a multiple linear regression on all the predictors and disregard all variables whose p-values are greater than 0.05. To start with, statistical significance does not always indicate predictive value. Even if forecasting is not the goal, this is not a good strategy because the p-values can be misleading when two or more predictors are correlated with each other (see Section 5.9).

Instead, we will use a measure of predictive accuracy. Five such measures are introduced in this section. They can be calculated using the `CV()` function, here applied to the model for US consumption:

```
CV(fit.consMR)
#>        CV       AIC      AICc       BIC     AdjR2
#>    0.1163 -409.2980 -408.8314 -389.9114    0.7486
```

We compare these values against the corresponding values from other models. For the CV, AIC, AICc and BIC measures, we want to find the model with the lowest value; for Adjusted R^2, we seek the model with the highest value.

Adjusted R^2

Computer output for a regression will always give the R^2 value, discussed in Section 5.2. However, it is not a good measure of the predictive ability of a model. It measures how well the model fits the historical data, but not how well the model will forecast future data.

In addition, 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. For these reasons, forecasters should not use R^2 to determine whether a model will give good predictions, as it will lead to overfitting.

An equivalent idea is to select the model which gives the minimum sum of squared errors (SSE), given by

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

Minimising the SSE is equivalent to maximising R^2 and will always choose the model with the most variables, and so is not a valid way of selecting predictors.

An alternative which is designed to overcome these problems is the adjusted R^2 (also called "R-bar-squared"):

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

where T is the number of observations and k is the number of predictors. This is an improvement on R^2, as it will no longer increase with each added predictor. Using this measure, the best model will be the one with the largest value of $\bar{R}^2$. Maximising $\bar{R}^2$ is equivalent to minimising the standard error $\hat{\sigma}_e$ given in Equation (5.3).

Maximising $\bar{R}^2$ works quite well as a method of selecting predictors, although it does tend to err on the side of selecting too many predictors.

Cross-validation

Time series cross-validation was introduced in Section 3.4 as a general tool for determining the predictive ability of a model. For regression models, it is also possible to use classical leave-one-out cross-validation to selection predictors (Bergmeir, Hyndman, & Koo, 2018). This is faster and makes more efficient use of the data. The procedure uses the following steps:

1. Remove observation t from the data set, and fit the model using the remaining data. Then compute the error ($e_t^* = y_t - \hat{y}_t$) for the omitted observation. (This is not the same as the residual because the tth observation was not used in estimating the value of $\hat{y}_t$.)
2. Repeat step 1 for $t = 1, \dots, T$.

3. Compute the MSE from $e_1^*, \dots, e_T^*$. We shall call this the **CV**.

Although this looks like a time-consuming procedure, there are fast methods of calculating CV, so that it takes no longer than fitting one model to the full data set. The equation for computing CV efficiently is given in Section 5.7. Under this criterion, the best model is the one with the smallest value of CV.

Akaike's Information Criterion

A closely-related method is Akaike's Information Criterion, which we define as

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

where T is the number of observations used for estimation and k is the number of predictors in the model. Different computer packages use slightly different definitions for the AIC, although they should all lead to the same model being selected. The $k+2$ part of the equation occurs because there are $k+2$ parameters in the model: the k coefficients for the predictors, the intercept and the variance of the residuals. The idea here is to penalise the fit of the model (SSE) with the number of parameters that need to be estimated.

The model with the minimum value of the AIC is often the best model for forecasting. For large values of T, minimising the AIC is equivalent to minimising the CV value.

Corrected Akaike's Information Criterion

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,

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

As with the AIC, the AICc should be minimised.

Schwarz's Bayesian Information Criterion

A related measure is Schwarz's Bayesian Information Criterion (usually abbreviated to BIC, SBIC or SC):

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

As with the AIC, minimising the BIC is intended to give the best model. The model chosen by the BIC is either the same as that chosen by the AIC, or one with fewer terms. This is because the BIC penalises the number of parameters more heavily than the AIC. For large values of T, minimising BIC is similar to leave-v-out cross-validation when $v = T[1 - 1/(\log(T) - 1)]$.

Which measure should we use?

While $\bar{R}^2$ is widely used, and has been around longer than the other measures, its tendency to select too many predictor variables makes it less suitable for forecasting.

Many statisticians like to use the BIC because it has the feature that if there is a true underlying model, the BIC will select that model given enough data. However, in reality, there is rarely, if ever, a true underlying model, and even if there was a true underlying model, selecting that model will not necessarily give the best forecasts (because the parameter estimates may not be accurate).

Consequently, we recommend that one of the AICc, AIC, or CV statistics be used, each of which has forecasting as their objective. If the value of T is large enough, they will all lead to the same model. In most of the examples in this book, we use the AICc value to select the forecasting model.

Example: US consumption

In the multiple regression example for forecasting US consumption we considered four predictors. With four predictors, there are $2^4 = 16$ possible models. Now we can check if all four predictors are actually useful, or whether we can drop one or more of them. All 16 models were fitted and the results are summarised in Table 5.1. A "1" indicates that the predictor was included in the model, and a "0" means that the predictor was not included in the model. Hence the first row shows the measures of predictive accuracy for a model including all four predictors.

The results have been sorted according to the AICc. Therefore the best models are given at the top of the table, and the worst at the bottom of the table.

Table 5.1: All 16 possible models for forecasting US consumption with 4 predictors.

Income	Production	Savings	Unemployment	CV	AIC	AICc	BIC	AdjR2
1	1	1	1	0.116	-409.3	-408.8	-389.9	0.749
1	0	1	1	0.116	-408.1	-407.8	-391.9	0.746
1	1	1	0	0.118	-407.5	-407.1	-391.3	0.745
1	0	1	0	0.129	-388.7	-388.5	-375.8	0.716
1	1	0	1	0.278	-243.2	-242.8	-227.0	0.386
1	0	0	1	0.283	-237.9	-237.7	-225.0	0.365
1	1	0	0	0.289	-236.1	-235.9	-223.2	0.359
0	1	1	1	0.293	-234.4	-234.0	-218.2	0.356
0	1	1	0	0.300	-228.9	-228.7	-216.0	0.334
0	1	0	1	0.303	-226.3	-226.1	-213.4	0.324
0	0	1	1	0.306	-224.6	-224.4	-211.7	0.318
0	1	0	0	0.314	-219.6	-219.5	-209.9	0.296
0	0	0	1	0.314	-217.7	-217.5	-208.0	0.288
1	0	0	0	0.372	-185.4	-185.3	-175.7	0.154
0	0	1	0	0.414	-164.1	-164.0	-154.4	0.052
0	0	0	0	0.432	-155.1	-155.0	-148.6	0.000

The best model contains all four predictors. However, a closer look at the results reveals some interesting features. There is clear separation between the models in the first four rows and the ones below. This indicates that Income and Savings are both more important variables than Production and Unemployment. Also, the first two rows have almost identical values of CV, AIC and AICc. So we could possibly drop the Production variable and get similar forecasts. Note that Production and Unemployment are highly (negatively) correlated, as shown in Figure 5.5, so most of the predictive information in Production is also contained in the Unemployment variable.

Best subset regression

Where possible, all potential regression models should be fitted (as was done in the example above) and the best model should be selected based on one of the measures discussed. This is known as “best subsets” regression or “all possible subsets” regression.

Stepwise regression

If there are a large number of predictors, it is not possible to fit all possible models. For example, 40 predictors leads to $2^{40} > 1$ trillion possible models! Consequently, a strategy is required to limit the number of models to be explored.

An approach that works quite well is *backwards stepwise regression*:

- Start with the model containing all potential predictors.
- Remove one predictor at a time. Keep the model if it improves the measure of predictive accuracy.
- Iterate until no further improvement.

If the number of potential predictors is too large, then the backwards stepwise regression will not work and *forward stepwise regression* can be used instead. This procedure starts with a model that includes only the intercept. Predictors are added one at a time, and the one that most improves the measure of predictive accuracy is retained in the model. The procedure is repeated until no further improvement can be achieved.

Alternatively for either the backward or forward direction, a starting model can be one that includes a subset of potential predictors. In this case, an extra step needs to be included. For the backwards procedure we should also consider adding a predictor with each step, and for the forward procedure we should also consider dropping a predictor with each step. These are referred to as *hybrid* procedures.

It is important to realise that any stepwise approach is not guaranteed to lead to the best possible model, but it almost always leads to a good model. For further details see James, Witten, Hastie, & Tibshirani (2014).

Beware of inference after selecting predictors

We do not discuss statistical inference of the predictors in this book (e.g., looking at p-values associated with each predictor). If you do wish to look at the statistical significance of the predictors, beware that *any* procedure involving selecting predictors first will invalidate the assumptions behind the p-values. The procedures we recommend for selecting predictors are helpful when the model is used for forecasting; they are not helpful if you wish to study the effect of any predictor on the forecast variable.

Bibliography

Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, *120*, 70–83. [DOI]

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: With applications in R*. New York: Springer. [Amazon]

5.6 Forecasting with regression

Recall that predictions of y can be obtained using

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1,t} + \hat{\beta}_2 x_{2,t} + \cdots + \hat{\beta}_k x_{k,t},$$

which comprises the estimated coefficients and ignores the error in the regression equation. Plugging in the values of the predictor variables $x_{1,t}, \ldots, x_{k,t}$ for $t = 1, \ldots, T$ returned the fitted (training-sample) values of y. What we are interested in here, however, is forecasting *future* values of y.

Ex-ante versus ex-post forecasts

When using regression models for time series data, we need to distinguish between the different types of forecasts that can be produced, depending on what is assumed to be known when the forecasts are computed.

Ex-ante forecasts are those that are made using only the information that is available in advance. For example, ex-ante forecasts for the percentage change in US consumption for quarters following the end of the sample, should only use information that was available *up to and including* 2016 Q3. These are genuine forecasts, made in advance using whatever information is available at the time. Therefore in order to generate ex-ante forecasts, the model requires forecasts of the predictors. To obtain these we can use one of the simple methods introduced in Section 3.1 or more sophisticated pure time series approaches that follow in Chapters 7 and 8. Alternatively, forecasts from some other source, such as a government agency, may be available and can be used.

Ex-post forecasts are those that are made using later information on the predictors. For example, ex-post forecasts of consumption may use the actual observations of the predictors, once these have been observed. These are not genuine forecasts, but are useful for studying the behaviour of forecasting models.

The model from which ex-post forecasts are produced should not be estimated using data from the forecast period. That is, ex-post forecasts can assume knowledge of the predictor variables (the x variables), but should not assume knowledge of the data that are to be forecast (the y variable).

A comparative evaluation of ex-ante forecasts and ex-post forecasts can help to separate out the sources of forecast uncertainty. This will show whether forecast errors have arisen due to poor forecasts of the predictor or due to a poor forecasting model.

Example: Australian quarterly beer production

Normally, we cannot use actual future values of the predictor variables when producing ex-ante forecasts because their values will not be known in advance. However, the special predictors introduced in Section 5.4 are all known in advance, as they are based on calendar variables (e.g., seasonal dummy variables or public holiday indicators) or deterministic functions of time (e.g. time trend). In such cases, there is no difference between ex-ante and ex-post forecasts.

```
beer2 <- window(ausbeer, start=1992)
fit.beer <- tslm(beer2 ~ trend + season)
fcast <- forecast(fit.beer)
autoplot(fcast) +
  ggtitle("Forecasts of beer production using regression") +
  xlab("Year") + ylab("megalitres")
```

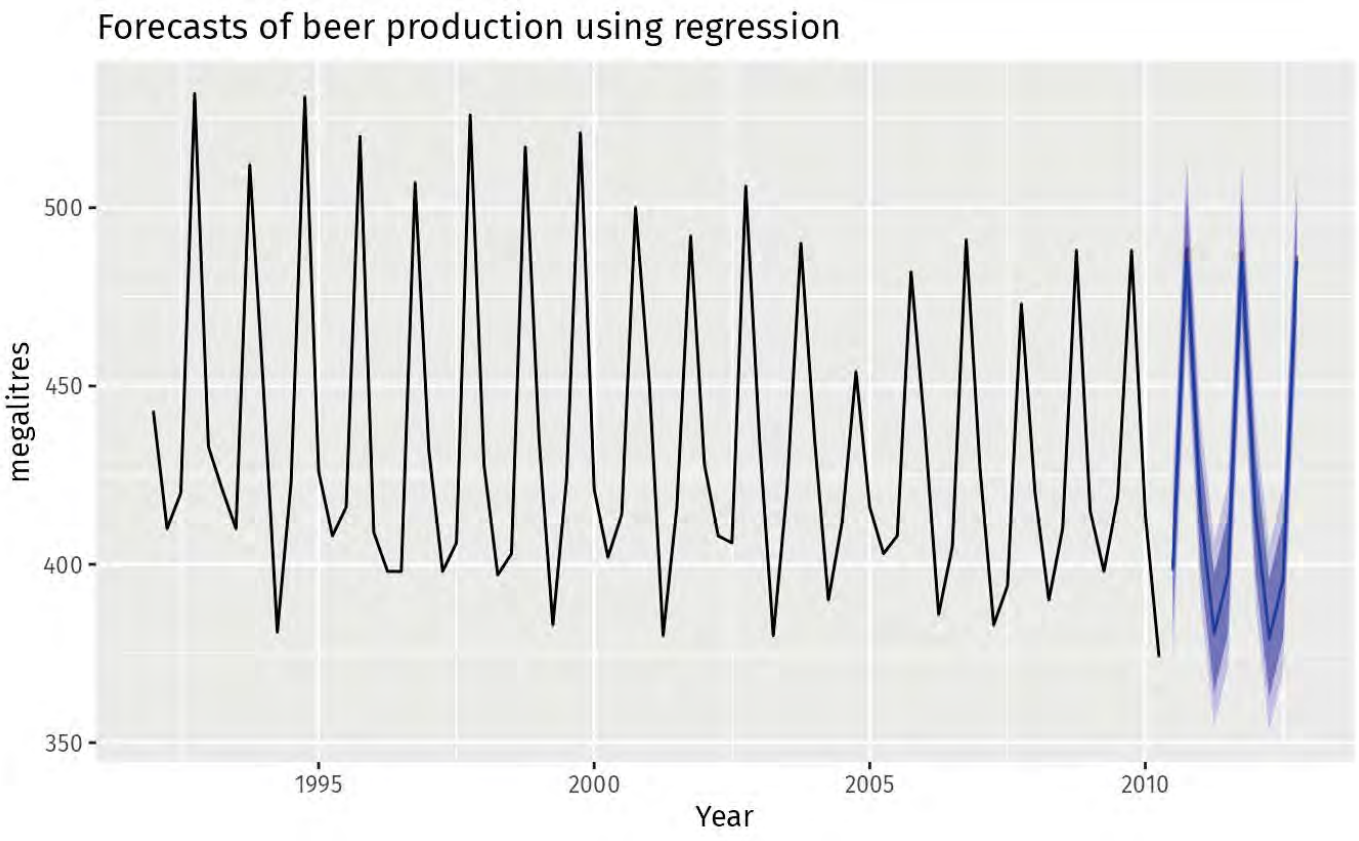


Figure 5.17: Forecasts from the regression model for beer production. The dark shaded region shows 80% prediction intervals and the light shaded region shows 95% prediction intervals.

Scenario based forecasting

In this setting, the forecaster assumes possible scenarios for the predictor variables that are of interest. For example, a US policy maker may be interested in comparing the predicted change in consumption when there is a constant growth of 1% and 0.5% respectively for income and savings with no change in the employment rate, versus a respective decline of 1% and 0.5%, for each of the four quarters following the end of the sample. The resulting forecasts are calculated below and shown in Figure 5.18. We should note that prediction intervals for scenario based forecasts do not include the uncertainty associated with the future values of the predictor variables. They assume that the values of the predictors are known in advance.

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

autoplot(uschange[, 1]) +
  ylab("% change in US consumption") +
  autolayer(fcast.up, PI = TRUE, series = "increase") +
  autolayer(fcast.down, PI = TRUE, series = "decrease") +
  guides(colour = guide_legend(title = "Scenario"))
```

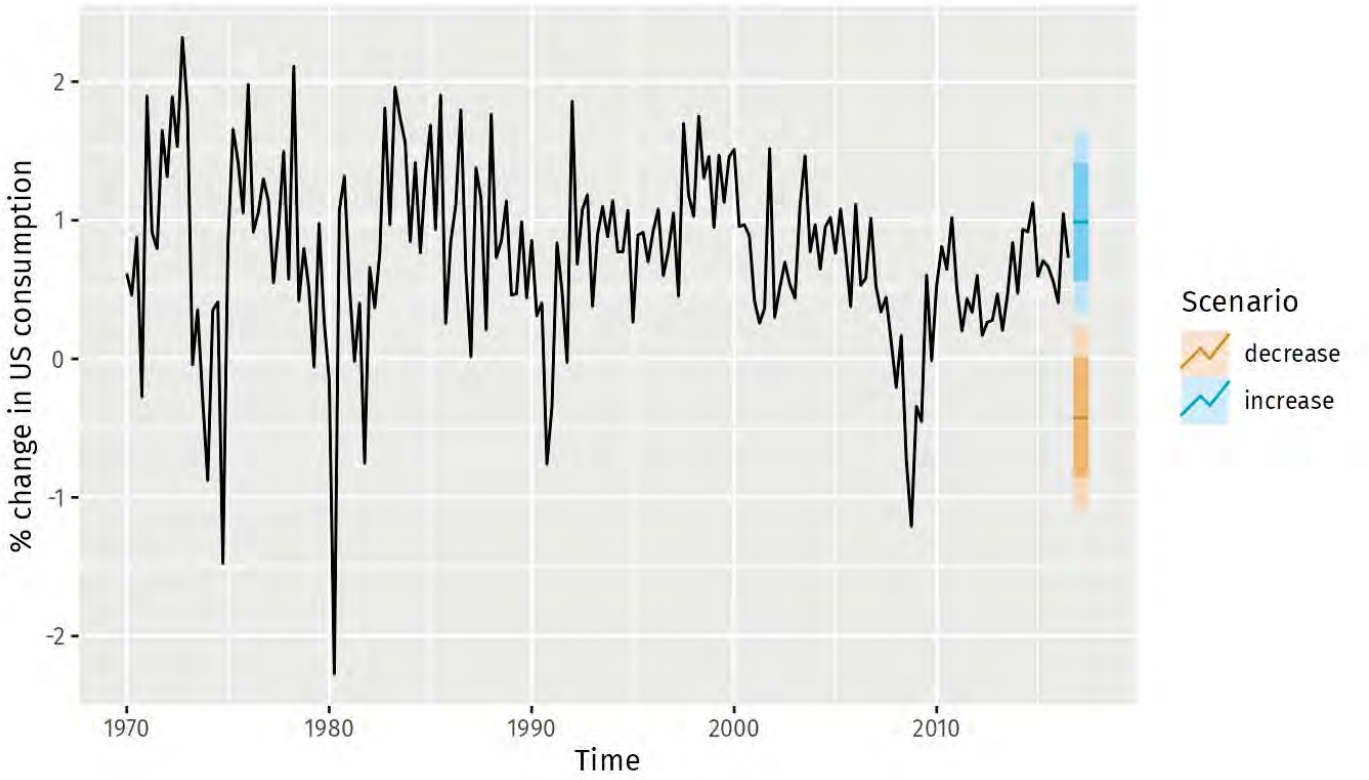


Figure 5.18: Forecasting percentage changes in personal consumption expenditure for the US under scenario based forecasting.

Building a predictive regression model

The great advantage of regression models is that they can be used to capture important relationships between the forecast variable of interest and the predictor variables. A major challenge however, is that in order to generate ex-ante forecasts, the model requires future values of each predictor. If scenario based forecasting is of interest then these models are extremely useful. However, if ex-ante forecasting is the main focus, obtaining forecasts of the predictors can be challenging (in many cases generating forecasts for the predictor variables can be more challenging than forecasting directly the forecast variable without using predictors).

An alternative formulation is to use as predictors their lagged values. Assuming that we are interested in generating a h-step ahead forecast we write

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

for $h = 1, 2 \ldots$. The predictor set is formed by values of the xs that are observed h time periods prior to observing y. Therefore when the estimated model is projected into the future, i.e., beyond the end of the sample T, all predictor values are available.

Including lagged values of the predictors does not only make the model operational for easily generating forecasts, it also makes it intuitively appealing. For example, the effect of a policy change with the aim of increasing production may not have an instantaneous effect on consumption expenditure. It is most likely that this will happen with a lagging effect. We touched upon this in Section 5.4 when briefly introducing distributed lags as predictors. Several directions for generalising regression models to better incorporate the rich dynamics observed in time series are discussed in Section 9.

Prediction intervals

With each forecast for the change in consumption in Figure 5.18, 95% and 80% prediction intervals are also included. The general formulation of how to calculate prediction intervals for multiple regression models is presented in Section 5.7. As this involves some advanced matrix algebra we present here the case for calculating prediction intervals for a simple regression, where a forecast can be generated using the equation,

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x.$$

Assuming that the regression errors are normally distributed, an approximate 95% prediction interval associated with this forecast is given by

$$\hat{y} \pm 1.96\hat{\sigma}_e\sqrt{1 + \frac{1}{T} + \frac{(x-\bar{x})^2}{(T-1)s_x^2}}, \tag{5.4}$$

where T is the total number of observations, $\bar{x}$ is the mean of the observed x values, s_x is the standard deviation of the observed x values and $\hat{\sigma}_e$ is the standard error of the regression given by Equation (5.3). Similarly, an 80% prediction interval can be obtained by replacing 1.96 by 1.28. Other prediction intervals can be obtained by replacing the 1.96 with the appropriate value given in Table 3.1. If R is used to obtain prediction intervals, more exact calculations are obtained (especially for small values of T) than what is given by Equation (5.4).

Equation (5.4) shows that the prediction interval is wider when x is far from $\bar{x}$. That is, we are more certain about our forecasts when considering values of the predictor variable close to its sample mean.

Example

The estimated simple regression line in the US consumption example is

$$\hat{y}_t = 0.55 + 0.28x_t.$$

Assuming that for the next four quarters, personal income will increase by its historical mean value of $\bar{x} = 0.72\%$, consumption is forecast to increase by 0.75% and the corresponding 95% and 80% prediction intervals are $[-0.45, 1.94]$ and $[-0.03, 1.52]$ respectively (calculated using R). If we assume an extreme increase of 5% in income, then the prediction intervals are considerably wider as shown in Figure 5.19.

```
fit.cons <- tslm(Consumption ~ Income, data = uschange)
h <- 4
fcast.ave <- forecast(fit.cons,
  newdata = data.frame(
    Income = rep(mean(uschange[,"Income"]), h)))
fcast.up <- forecast(fit.cons,
  newdata = data.frame(Income = rep(5, h)))
autoplot(uschange[, "Consumption"]) +
  ylab("% change in US consumption") +
  autolayer(fcast.ave, series = "Average increase",
    PI = TRUE) +
  autolayer(fcast.up, series = "Extreme increase",
    PI = TRUE) +
  guides(colour = guide_legend(title = "Scenario"))
```

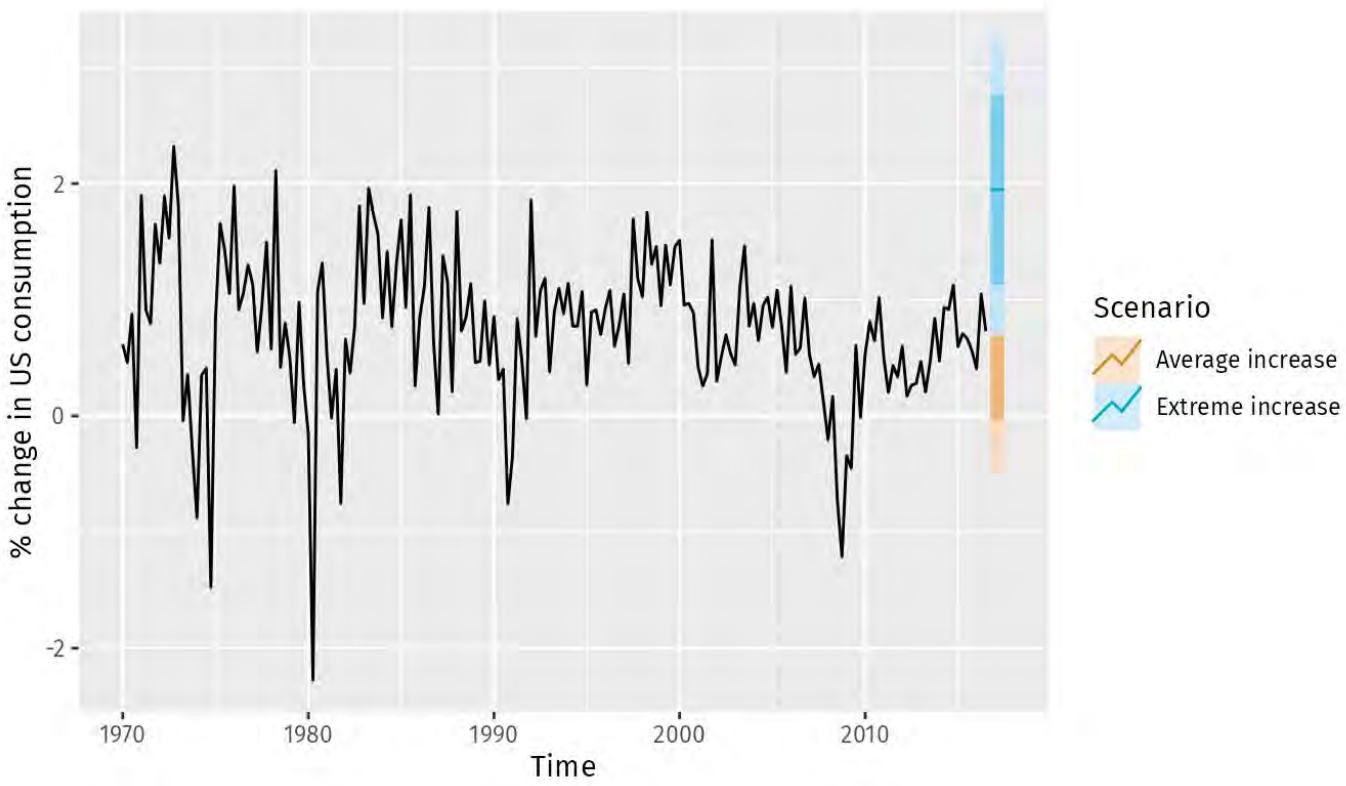


Figure 5.19: Prediction intervals if income is increased by its historical mean of 0.72% versus an extreme increase of 5%.

5.7 Matrix formulation

Warning: this is a more advanced, optional section and assumes knowledge of matrix algebra.

Recall that multiple regression model can be written as

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

where ε_t has mean zero and variance σ^2. This expresses the relationship between a single value of the forecast variable and the predictors.

It can be convenient to write this in matrix form where all the values of the forecast variable are given in a single equation. Let $\boldsymbol{y} = (y_1, \ldots, y_T)'$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_T)'$, $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_k)'$ and

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

Then

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

where $\boldsymbol{\varepsilon}$ has mean $\boldsymbol{0}$ and variance $\sigma^2\boldsymbol{I}$. Note that the $\boldsymbol{X}$ matrix has T rows reflecting the number of observations and $k+1$ columns reflecting the intercept which is represented by the column of ones plus the number of predictors.

Least squares estimation

Least squares estimation is performed by minimising the expression $\boldsymbol{\varepsilon}'\boldsymbol{\varepsilon} = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})'(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$. It can be shown that this is minimised when $\boldsymbol{\beta}$ takes the value

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

This is sometimes known as the "normal equation". The estimated coefficients require the inversion of the matrix $\boldsymbol{X}'\boldsymbol{X}$. If $\boldsymbol{X}$ is not of full column rank then matrix

$\boldsymbol{X}'\boldsymbol{X}$ is singular and the model cannot be estimated. This will occur, for example, if you fall for the "dummy variable trap", i.e., having the same number of dummy variables as there are categories of a categorical predictor, as discussed in Section 5.4.

The residual variance is estimated using

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

Fitted values and cross-validation

The normal equation shows that the fitted values can be calculated using

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

where $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$ is known as the "hat-matrix" because it is used to compute $\hat{\boldsymbol{y}}$ ("y-hat").

If the diagonal values of $\boldsymbol{H}$ are denoted by $h_1, \dots, h_T$, then the cross-validation statistic can be computed using

$$\text{CV} = \frac{1}{T}\sum_{t=1}^{T}[e_t/(1-h_t)]^2,$$

where e_t is the residual obtained from fitting the model to all T observations. Thus, it is not necessary to actually fit T separate models when computing the CV statistic.

Forecasts and prediction intervals

Let $\boldsymbol{x}^*$ be a row vector containing the values of the predictors (in the same format as $\boldsymbol{X}$) for which we want to generate a forecast . Then the forecast is given by

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

and its estimated variance is given by

$$\hat{\sigma}_e^2\left[1 + \boldsymbol{x}^*(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{x}^*)'\right].$$

A 95% prediction interval can be calculated (assuming normally distributed errors) as

$$\hat{y} \pm 1.96\hat{\sigma}_e\sqrt{1 + \boldsymbol{x}^*(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{x}^*)'}.$$

This takes into account the uncertainty due to the error term ε and the uncertainty in the coefficient estimates. However, it ignores any errors in $\boldsymbol{x}^*$. Thus, if the future values of the predictors are uncertain, then the prediction interval calculated using this expression will be too narrow.

5.8 Nonlinear regression

Although the linear relationship assumed so far in this chapter is often adequate, there are many cases in which a nonlinear functional form is more suitable. To keep things simple in this section we assume that we only have one predictor x.

The simplest way of modelling a nonlinear relationship is to transform the forecast variable y and/or the predictor variable x before estimating a regression model. While this provides a non-linear functional form, the model is still linear in the parameters. The most commonly used transformation is the (natural) logarithm (see Section 3.2).

A **log-log** functional form is specified as

$$\log y = \beta_0 + \beta_1 \log x + \varepsilon.$$

In this model, the slope β_1 can be interpreted as an elasticity: β_1 is the average percentage change in y resulting from a 1% increase in x. Other useful forms can also be specified. The **log-linear** form is specified by only transforming the forecast variable and the **linear-log** form is obtained by transforming the predictor.

Recall that in order to perform a logarithmic transformation to a variable, all of its observed values must be greater than zero. In the case that variable x contains zeros, we use the transformation $\log(x+1)$; i.e., we add one to the value of the variable and then take logarithms. This has a similar effect to taking logarithms but avoids the problem of zeros. It also has the neat side-effect of zeros on the original scale remaining zeros on the transformed scale.

There are cases for which simply transforming the data will not be adequate and a more general specification may be required. Then the model we use is

$$y = f(x) + \varepsilon$$

where f is a nonlinear function. In standard (linear) regression, $f(x) = \beta_0 + \beta_1 x$. In the specification of nonlinear regression that follows, we allow f to be a more flexible nonlinear function of x, compared to simply a logarithmic or other transformation.

One of the simplest specifications is to make f **piecewise linear**. That is, we introduce points where the slope of f can change. These points are called **knots**. This can be achieved by letting $x_{1,t} = x$ and introducing variable $x_{2,t}$ such that

$$x_{2,t} = (x - c)_+ = \begin{cases} 0 & x < c \\ (x - c) & x \geq c \end{cases}$$

The notation $(x - c)_+$ means the value $x - c$ if it is positive and 0 otherwise. This forces the slope to bend at point c. Additional bends can be included in the relationship by adding further variables of the above form.

An example of this follows by considering $x = t$ and fitting a piecewise linear trend to a time series.

Piecewise linear relationships constructed in this way are a special case of **regression splines**. In general, a linear regression spline is obtained using

$$x_1 = x \quad x_2 = (x - c_1)_+ \quad \dots \quad x_k = (x - c_{k-1})_+$$

where $c_1, \dots, c_{k-1}$ are the knots (the points at which the line can bend). Selecting the number of knots $(k - 1)$ and where they should be positioned can be difficult and somewhat arbitrary. Some automatic knot selection algorithms are available in some software, but are not yet widely used.

A smoother result can be obtained using piecewise cubics rather than piecewise lines. These are constrained to be continuous (they join up) and smooth (so that there are no sudden changes of direction, as we see with piecewise linear splines). In general, a cubic regression spline is written as

$$x_1 = x \quad x_2 = x^2 \quad x_3 = x^3 \quad x_4 = (x - c_1)_+^3 \quad \dots \quad x_k = (x - c_{k-3})_+^3.$$

Cubic splines usually give a better fit to the data. However, forecasts of y become unreliable when x is outside the range of the historical data.

Forecasting with a nonlinear trend

In Section 5.4 fitting a linear trend to a time series by setting $x = t$ was introduced. The simplest way of fitting a nonlinear trend is using quadratic or higher order trends obtained by specifying

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

However, it is not recommended that quadratic or higher order trends be used in forecasting. When they are extrapolated, the resulting forecasts are often unrealistic.

A better approach is to use the piecewise specification introduced above and fit a piecewise linear trend which bends at some point in time. We can think of this as a nonlinear trend constructed of linear pieces. If the trend bends at time τ, then it can be specified by simply replacing $x = t$ and $c = \tau$ above such that we include the predictors,

$$\begin{aligned} x_{1,t} &= t \\ x_{2,t} = (t-\tau)_+ &= \begin{cases} 0 & t < \tau \\ (t-\tau) & t \geq \tau \end{cases} \end{aligned}$$

in the model. If the associated coefficients of $x_{1,t}$ and $x_{2,t}$ are β_1 and β_2, then β_1 gives the slope of the trend before time τ, while the slope of the line after time τ is given by $\beta_1 + \beta_2$. Additional bends can be included in the relationship by adding further variables of the form $(t-\tau)_+$ where τ is the "knot" or point in time at which the line should bend.

Example: Boston marathon winning times

The top panel of Figure 5.20 shows the Boston marathon winning times (in minutes). The course was lengthened (from 24.5 miles to 26.2 miles) in 1924, which led to a jump in the winning times, so we only consider data from that date onwards. The time series shows a general downward trend as the winning times have been improving over the years. The bottom panel shows the residuals from fitting a linear trend to the data. The plot shows an obvious nonlinear pattern which has not been captured by the linear trend. There is also some heteroscedasticity, with decreasing variation over time.

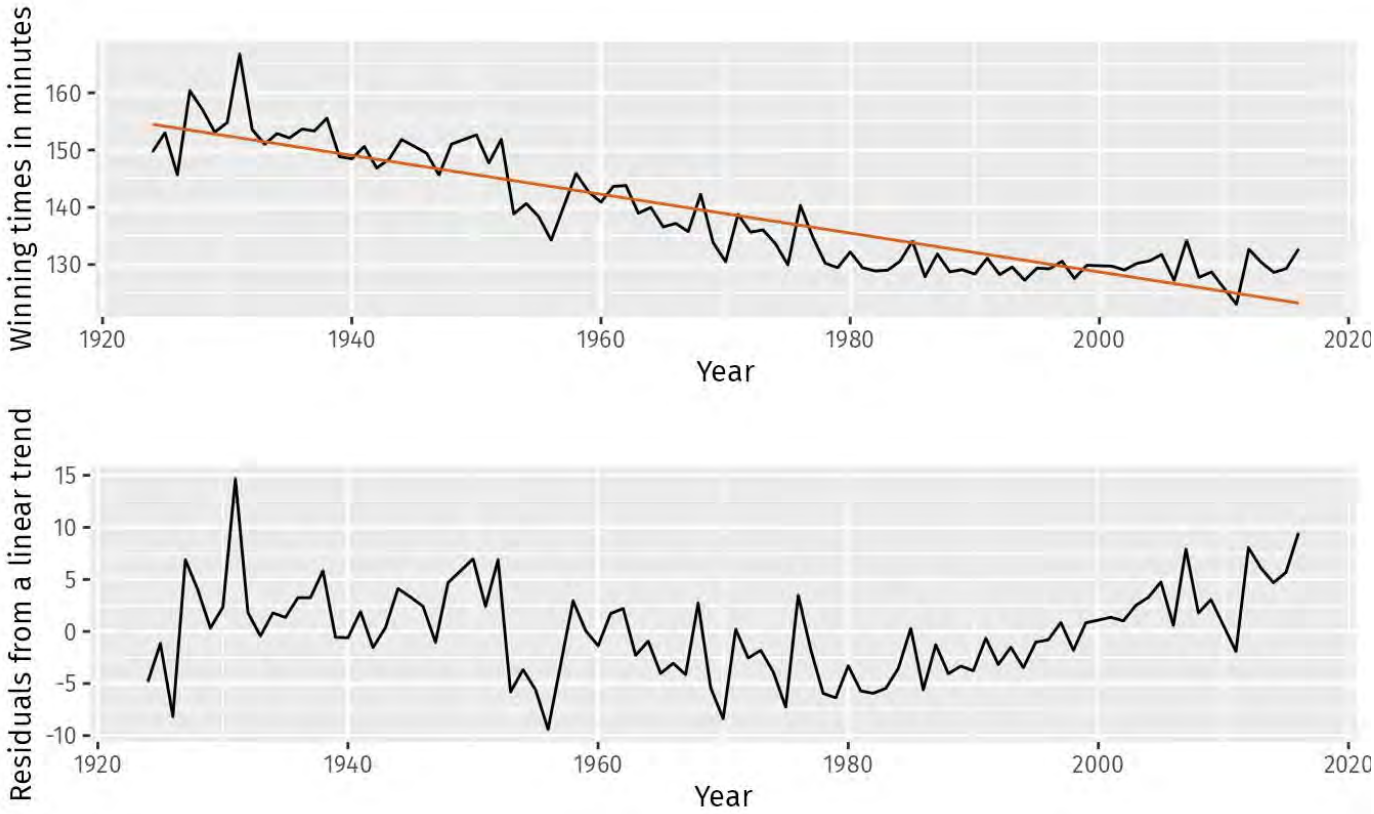


Figure 5.20: Fitting a linear trend to the Boston marathon winning times is inadequate

Fitting an exponential trend (equivalent to a log-linear regression) to the data can be achieved by transforming the y variable so that the model to be fitted is,

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

This also addresses the heteroscedasticity. The fitted exponential trend and forecasts are shown in Figure 5.21. Although the exponential trend does not seem to fit the data much better than the linear trend, it gives a more sensible projection in that the winning times will decrease in the future but at a decaying rate rather than a fixed linear rate.

The plot of winning times reveals three different periods. There is a lot of volatility in the winning times up to about 1950, with the winning times barely declining. After 1950 there is a near-linear decrease in times, followed by a flattening out after the 1980s, with the suggestion of an upturn towards the end of the sample. To account for these changes, we specify the years 1950 and 1980 as knots. We should warn here that subjective identification of knots can lead to over-fitting, which can be detrimental to the forecast performance of a model, and should be performed with caution.

```
boston_men <- window(marathon, start=1924)
h <- 10
fit.lin <- tslm(boston_men ~ trend)
fcasts.lin <- forecast(fit.lin, h = h)
fit.exp <- tslm(boston_men ~ trend, lambda = 0)
fcasts.exp <- forecast(fit.exp, h = h)

t <- time(boston_men)
t.break1 <- 1950
t.break2 <- 1980
tb1 <- ts(pmax(0, t - t.break1), start = 1924)
tb2 <- ts(pmax(0, t - t.break2), start = 1924)

fit.pw <- tslm(boston_men ~ t + tb1 + tb2)
t.new <- t[length(t)] + seq(h)
tb1.new <- tb1[length(tb1)] + seq(h)
tb2.new <- tb2[length(tb2)] + seq(h)

newdata <- cbind(t=t.new, tb1=tb1.new, tb2=tb2.new) %>%
  as.data.frame()
fcasts.pw <- forecast(fit.pw, newdata = newdata)

fit.spline <- tslm(boston_men ~ t + I(t^2) + I(t^3) +
  I(tb1^3) + I(tb2^3))
fcasts.spl <- forecast(fit.spline, newdata = newdata)

autoplot(boston_men) +
  autolayer(fitted(fit.lin), series = "Linear") +
  autolayer(fitted(fit.exp), series = "Exponential") +
  autolayer(fitted(fit.pw), series = "Piecewise") +
  autolayer(fitted(fit.spline), series = "Cubic Spline") +
  autolayer(fcasts.pw, series="Piecewise") +
  autolayer(fcasts.lin, series="Linear", PI=FALSE) +
  autolayer(fcasts.exp, series="Exponential", PI=FALSE) +
  autolayer(fcasts.spl, series="Cubic Spline", PI=FALSE) +
  xlab("Year") + ylab("Winning times in minutes") +
```

```
  ggtitle("Boston Marathon") +
  guides(colour = guide_legend(title = " "))
```

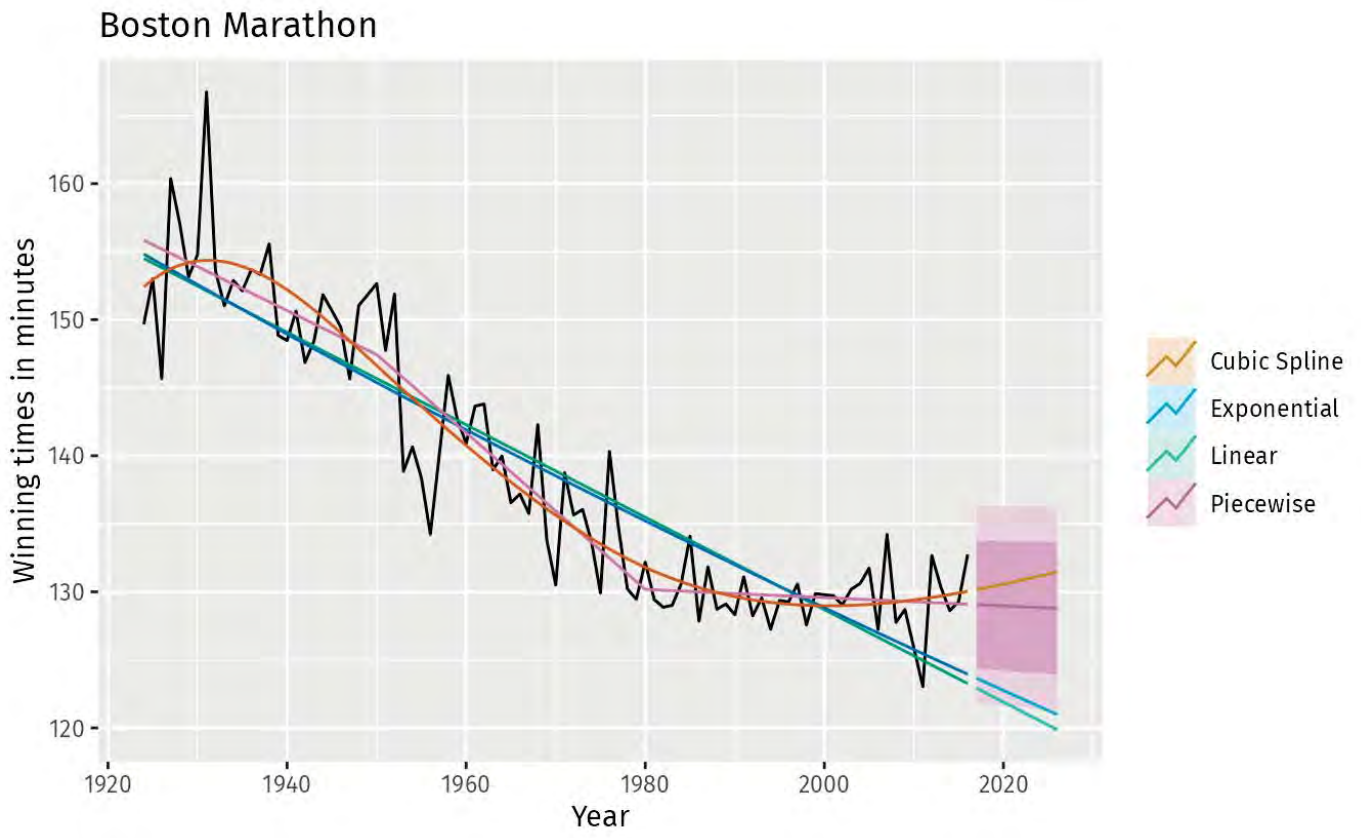


Figure 5.21: Projecting forecasts from a linear, exponential, piecewise linear trends and a cubic spline for the Boston marathon winning times

Figure 5.21 above shows the fitted lines and forecasts from linear, exponential, piecewise linear, and cubic spline trends. The best forecasts appear to come from the piecewise linear trend, while the cubic spline gives the best fit to the historical data but poor forecasts.

There is an alternative formulation of cubic splines (called **natural cubic smoothing splines**) that imposes some constraints, so the spline function is linear at the end, which usually gives much better forecasts without compromising the fit. In Figure 5.22, we have used the `splinef()` function to produce the cubic spline forecasts. This uses many more knots than we used in Figure 5.21, but the coefficients are constrained to prevent over-fitting, and the curve is linear at both ends. This has the added advantage that knot selection is not subjective. We have also used a log transformation (`lambda=0`) to handle the heteroscedasticity.

```
boston_men %>%
  splinef(lambda=0) %>%
  autoplot()
```

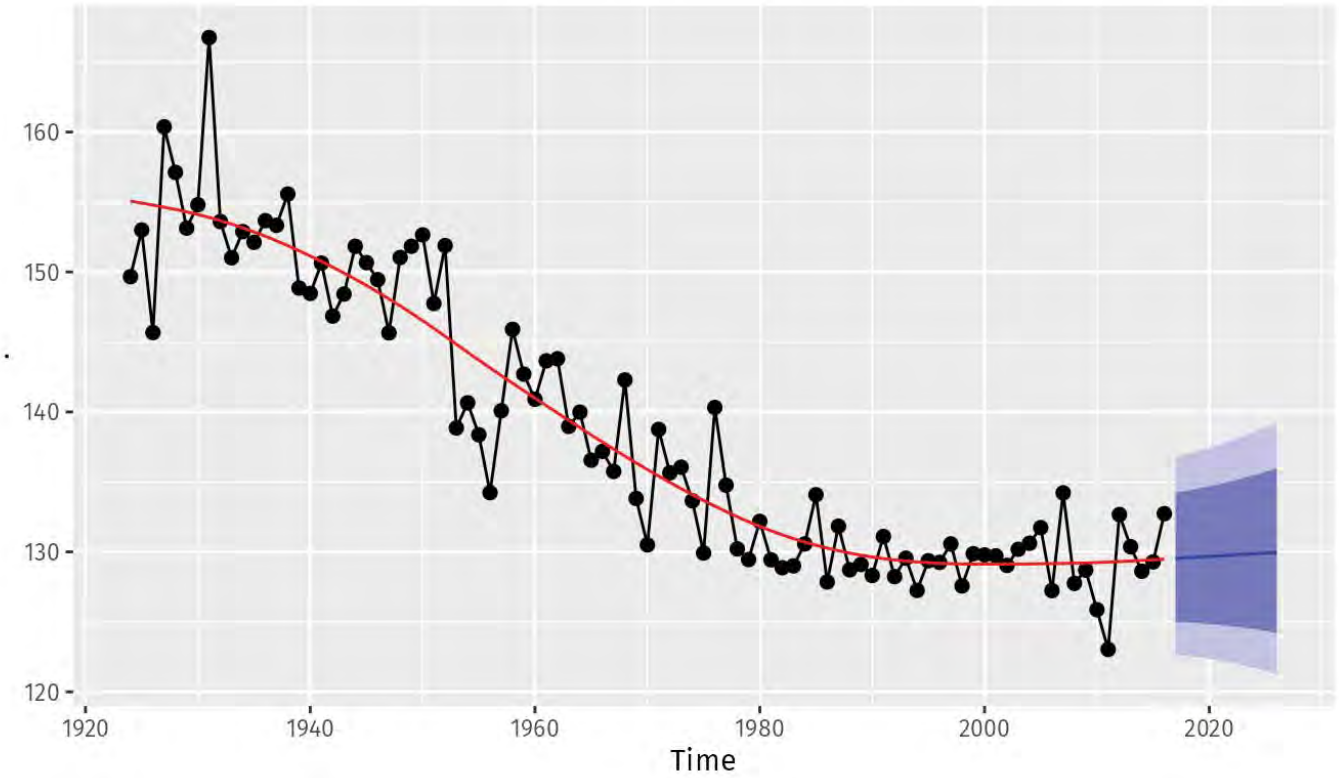


Figure 5.22: Natural cubic smoothing splines applied to the marathon data. The forecasts are a linear projection of the trend at the end of the observed data.

The residuals plotted in Figure 5.23 show that this model has captured the trend well, although there is some heteroscedasticity remaining. The wide prediction interval associated with the forecasts reflects the volatility observed in the historical winning times.

```
boston_men %>%
  splinef(lambda=0) %>%
  checkresiduals()
```

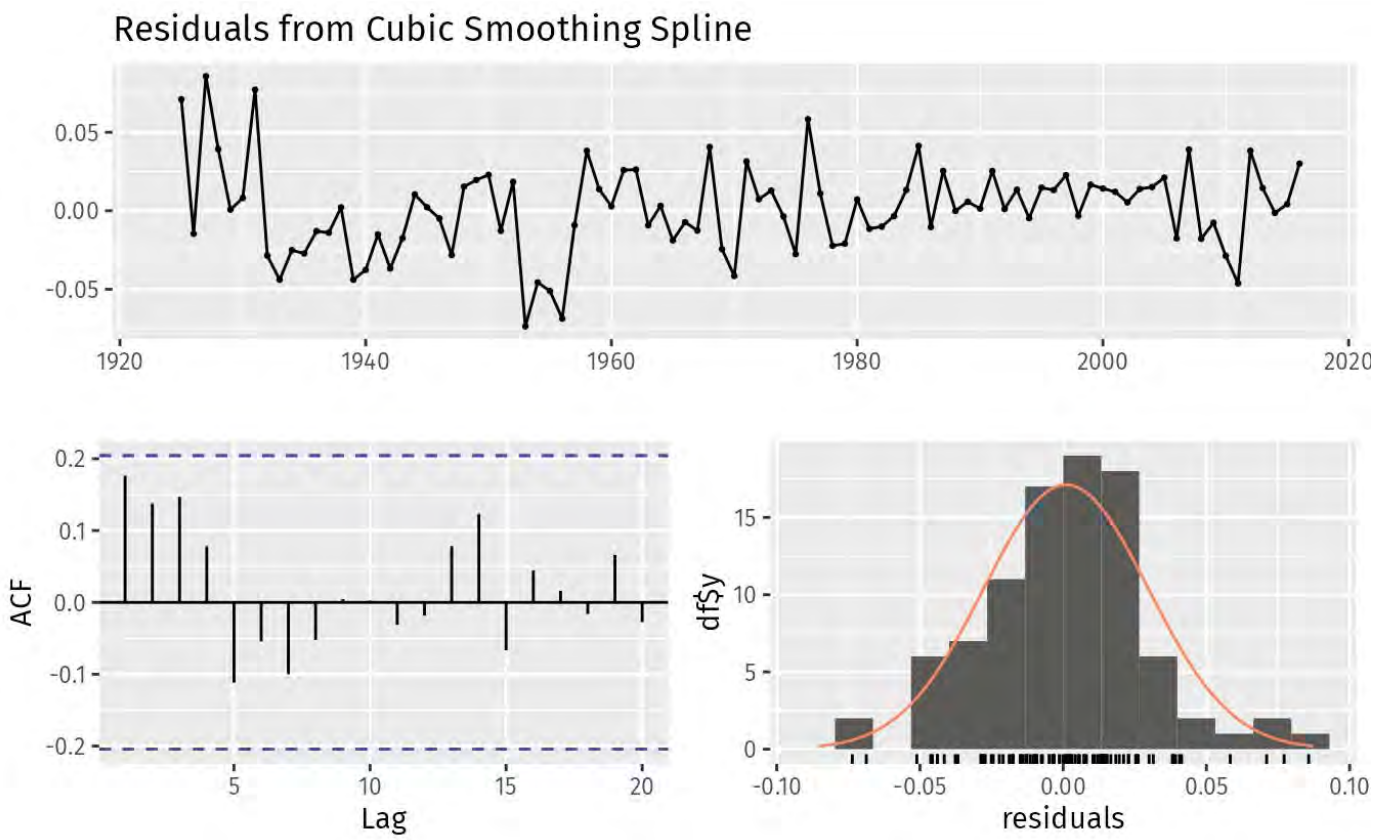


Figure 5.23: Residuals from the cubic spline trend.

```
#>
#>  Ljung-Box test
#>
#> data:  Residuals from Cubic Smoothing Spline
#> Q* = 10, df = 10, p-value = 0.4
#>
#> Model df: 0.   Total lags used: 10
```

5.9 Correlation, causation and forecasting

Correlation is not causation

It is important not to confuse correlation with causation, or causation with forecasting. A variable x may be useful for forecasting a variable y, but that does not mean x is causing y. It is possible that x *is* causing y, but it may be that y is causing x , or that the relationship between them is more complicated than simple causality.

For example, it is possible to model the number of drownings at a beach resort each month with the number of ice-creams sold in the same period. The model can give reasonable forecasts, not because ice-creams cause drownings, but because people eat more ice-creams on hot days when they are also more likely to go swimming. So the two variables (ice-cream sales and drownings) are correlated, but one is not causing the other. They are both caused by a third variable (temperature). This is an example of "confounding" — where an omitted variable causes changes in both the response variable and at least one predictor variables.

We describe a variable that is not included in our forecasting model as a **confounder** when it influences both the response variable and at least one predictor variable. Confounding makes it difficult to determine what variables are *causing* changes in other variables, but it does not necessarily make forecasting more difficult.

Similarly, it is possible to forecast if it will rain in the afternoon by observing the number of cyclists on the road in the morning. When there are fewer cyclists than usual, it is more likely to rain later in the day. The model can give reasonable forecasts, not because cyclists prevent rain, but because people are more likely to cycle when the published weather forecast is for a dry day. In this case, there is a causal relationship, but in the opposite direction to our forecasting model. The number of cyclists falls because there is rain forecast. That is, y (rainfall) is affecting x (cyclists).

It is important to understand that correlations are useful for forecasting, even when there is no causal relationship between the two variables, or when the causality runs in the opposite direction to the model, or when there is confounding.

However, often a better model is possible if a causal mechanism can be determined. A better model for drownings will probably include temperatures and visitor numbers and exclude ice-cream sales. A good forecasting model for rainfall will not include cyclists, but it will include atmospheric observations from the previous few days.

Forecasting with correlated predictors

When two or more predictors are highly correlated it is always challenging to accurately separate their individual effects. Suppose we are forecasting monthly sales of a company for 2012, using data from 2000–2011. In January 2008, a new competitor came into the market and started taking some market share. At the same time, the economy began to decline. In your forecasting model, you include both competitor activity (measured using advertising time on a local television station) and the health of the economy (measured using GDP). It will not be possible to separate the effects of these two predictors because they are highly correlated.

Having correlated predictors is not really a problem for forecasting, as we can still compute forecasts without needing to separate out the effects of the predictors. However, it becomes a problem with scenario forecasting as the scenarios should take account of the relationships between predictors. It is also a problem if some historical analysis of the contributions of various predictors is required.

Multicollinearity and forecasting

A closely related issue is **multicollinearity**, which occurs when similar information is provided by two or more of the predictor variables in a multiple regression.

It can occur when two predictors are highly correlated with each other (that is, they have a correlation coefficient close to +1 or -1). In this case, knowing the value of one of the variables tells you a lot about the value of the other variable. Hence, they are providing similar information. For example, foot size can be used to predict height, but including the size of both left and right feet in the same model is not going to make the forecasts any better, although it won't make them worse either.

Multicollinearity can also occur when a linear combination of predictors is highly correlated with another linear combination of predictors. In this case, knowing the value of the first group of predictors tells you a lot about the value of the second group of predictors. Hence, they are providing similar information.

An example of this problem is the dummy variable trap discussed in Section 5.4. Suppose you have quarterly data and use four dummy variables, d_1, d_2, d_3 and d_4. Then $d_4 = 1 - d_1 - d_2 - d_3$, so there is perfect correlation between d_4 and $d_1 + d_2 + d_3$.

In the case of perfect correlation (i.e., a correlation of +1 or -1, such as in the dummy variable trap), it is not possible to estimate the regression model.

If there is high correlation (close to but not equal to +1 or -1), then the estimation of the regression coefficients is computationally difficult. In fact, some software (notably Microsoft Excel) may give highly inaccurate estimates of the coefficients. Most reputable statistical software will use algorithms to limit the effect of multicollinearity on the coefficient estimates, but you do need to be careful. The major software packages such as R, SPSS, SAS and Stata all use estimation algorithms to avoid the problem as much as possible.

When multicollinearity is present, the uncertainty associated with individual regression coefficients will be large. This is because they are difficult to estimate. Consequently, statistical tests (e.g., t-tests) on regression coefficients are unreliable. (In forecasting we are rarely interested in such tests.) Also, it will not be possible to make accurate statements about the contribution of each separate predictor to the forecast.

Forecasts will be unreliable if the values of the future predictors are outside the range of the historical values of the predictors. For example, suppose you have fitted a regression model with predictors x_1 and x_2 which are highly correlated with each other, and suppose that the values of x_1 in the fitting data ranged between 0 and 100. Then forecasts based on $x_1 > 100$ or $x_1 < 0$ will be unreliable. It is always a little dangerous when future values of the predictors lie much outside the historical range, but it is especially problematic when multicollinearity is present.

Note that if you are using good statistical software, if you are not interested in the specific contributions of each predictor, and if the future values of your predictor variables are within their historical ranges, there is nothing to worry about — multicollinearity is not a problem except when there is perfect correlation.

5.10 Exercises

1. Daily electricity demand for Victoria, Australia, during 2014 is contained in `elecdaily`. The data for the first 20 days can be obtained as follows.

```
daily20 <- head(elecdaily,20)
```

   a. Plot the data and find the regression model for Demand with temperature as an explanatory variable. Why is there a positive relationship?

   b. Produce a residual plot. Is the model adequate? Are there any outliers or influential observations?

   c. Use the model to forecast the electricity demand that you would expect for the next day if the maximum temperature was 15° and compare it with the forecast if the with maximum temperature was 35°. Do you believe these forecasts?

   d. Give prediction intervals for your forecasts. The following R code will get you started:

```
autoplot(daily20, facets=TRUE)
daily20 %>%
  as.data.frame() %>%
  ggplot(aes(x=Temperature, y=Demand)) +
  geom_point() +
  geom_smooth(method="lm", se=FALSE)
fit <- tslm(Demand ~ Temperature, data=daily20)
checkresiduals(fit)
forecast(fit, newdata=data.frame(Temperature=c(15,35)))
```

   e. Plot Demand vs Temperature for all of the available data in `elecdaily`. What does this say about your model?

2. Data set `mens400` contains the winning times (in seconds) for the men's 400 meters final in each Olympic Games from 1896 to 2016.

   a. Plot the winning time against the year. Describe the main features of the plot.

b. Fit a regression line to the data. Obviously the winning times have been decreasing, but at what *average* rate per year?
c. Plot the residuals against the year. What does this indicate about the suitability of the fitted line?
d. Predict the winning time for the men's 400 meters final in the 2020 Olympics. Give a prediction interval for your forecasts. What assumptions have you made in these calculations?

3. Type `easter(ausbeer)` and interpret what you see.

4. An elasticity coefficient is the ratio of the percentage change in the forecast variable (y) to the percentage change in the predictor variable (x). Mathematically, the elasticity is defined as $(dy/dx) \times (x/y)$. Consider the log-log model,

$$\log y = \beta_0 + \beta_1 \log x + \varepsilon.$$

Express y as a function of x and show that the coefficient β_1 is the elasticity coefficient.

5. The data set `fancy` concerns the monthly sales figures of a shop which opened in January 1987 and sells gifts, souvenirs, and novelties. The shop is situated on the wharf at a beach resort town in Queensland, Australia. The sales volume varies with the seasonal population of tourists. There is a large influx of visitors to the town at Christmas and for the local surfing festival, held every March since 1988. Over time, the shop has expanded its premises, range of products, and staff.

a. Produce a time plot of the data and describe the patterns in the graph. Identify any unusual or unexpected fluctuations in the time series.
b. Explain why it is necessary to take logarithms of these data before fitting a model.
c. Use R to fit a regression model to the logarithms of these sales data with a linear trend, seasonal dummies and a "surfing festival" dummy variable.
d. Plot the residuals against time and against the fitted values. Do these plots reveal any problems with the model?
e. Do boxplots of the residuals for each month. Does this reveal any problems with the model?
f. What do the values of the coefficients tell you about each variable?
g. What does the Breusch-Godfrey test tell you about your model?

h. Regardless of your answers to the above questions, use your regression model to predict the monthly sales for 1994, 1995, and 1996. Produce prediction intervals for each of your forecasts.
i. Transform your predictions and intervals to obtain predictions and intervals for the raw data.
j. How could you improve these predictions by modifying the model?

6. The `gasoline` series consists of weekly data for supplies of US finished motor gasoline product, from 2 February 1991 to 20 January 2017. The units are in "million barrels per day". Consider only the data to the end of 2004.

   a. Fit a harmonic regression with trend to the data. Experiment with changing the number Fourier terms. Plot the observed gasoline and fitted values and comment on what you see.

   b. Select the appropriate number of Fourier terms to include by minimising the AICc or CV value.

   c. Check the residuals of the final model using the `checkresiduals()` function. Even though the residuals fail the correlation tests, the results are probably not severe enough to make much difference to the forecasts and prediction intervals. (Note that the correlations are relatively small, even though they are significant.)

   d. To forecast using harmonic regression, you will need to generate the future values of the Fourier terms. This can be done as follows.

```
fc <- forecast(fit, newdata=data.frame(fourier(x,K,h)))
```

   where `fit` is the fitted model using `tslm()`, `K` is the number of Fourier terms used in creating `fit`, and `h` is the forecast horizon required.

   Forecast the next year of data.

   e. Plot the forecasts along with the actual data for 2005. What do you find?

7. Data set `huron` gives the water level of Lake Huron in feet from 1875 to 1972.

   a. Plot the data and comment on its features.
   b. Fit a linear regression and compare this to a piecewise linear trend model with a knot at 1915.
   c. Generate forecasts from these two models for the period up to 1980 and comment on these.

8. *(For advanced readers following on from Section 5.7).*

Using matrix notation it was shown that if $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{e}$ has mean $\boldsymbol{0}$ and variance matrix $\sigma^2\boldsymbol{I}$, the estimated coefficients are given by $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$ and a forecast is given by $\hat{y} = \boldsymbol{x}^*\hat{\boldsymbol{\beta}} = \boldsymbol{x}^*(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$ where $\boldsymbol{x}^*$ is a row vector containing the values of the regressors for the forecast (in the same format as $\boldsymbol{X}$), and the forecast variance is given by $var(\hat{y}) = \sigma^2\left[1 + \boldsymbol{x}^*(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{x}^*)'\right]$.

Consider the simple time trend model where $y_t = \beta_0 + \beta_1 t$. Using the following results,

$$\sum_{t=1}^{T} t = \frac{1}{2}T(T+1), \quad \sum_{t=1}^{T} t^2 = \frac{1}{6}T(T+1)(2T+1)$$

derive the following expressions:

a. $\boldsymbol{X}'\boldsymbol{X} = \dfrac{1}{6}\begin{bmatrix} 6T & 3T(T+1) \\ 3T(T+1) & T(T+1)(2T+1) \end{bmatrix}$

b. $(\boldsymbol{X}'\boldsymbol{X})^{-1} = \dfrac{2}{T(T^2-1)}\begin{bmatrix} (T+1)(2T+1) & -3(T+1) \\ -3(T+1) & 6 \end{bmatrix}$

c. $\hat{\beta}_0 = \dfrac{2}{T(T-1)}\left[(2T+1)\sum_{t=1}^{T} y_t - 3\sum_{t=1}^{T} t y_t\right]$

$\hat{\beta}_1 = \dfrac{6}{T(T^2-1)}\left[2\sum_{t=1}^{T} t y_t - (T+1)\sum_{t=1}^{T} y_t\right]$

d. $\text{Var}(\hat{y}_t) = \hat{\sigma}^2\left[1 + \dfrac{2}{T(T-1)}\left(1 - 4T - 6h + 6\dfrac{(T+h)^2}{T+1}\right)\right]$

5.11 Further reading

There are countless books on regression analysis, but few with a focus on regression for time series and forecasting.

- A good general and modern book on regression is Sheather (2009).
- Another general regression text full of excellent practical advice is Harrell (2015).
- Ord et al. (2017) provides a practical coverage of regression models for time series in Chapters 7–9, with a strong emphasis on forecasting.

Bibliography

Harrell, F. E. (2015). *Regression modeling strategies: With applications to linear models, logistic and ordinal regression, and survival analysis* (2nd ed). New York, USA: Springer. [Amazon]

Ord, J. K., Fildes, R., & Kourentzes, N. (2017). *Principles of business forecasting* (2nd ed.). Wessex Press Publishing Co. [Amazon]

Sheather, S. J. (2009). *A modern approach to regression with R*. New York, USA: Springer. [Amazon]

Chapter 6 Time series decomposition

Time series data can exhibit a variety of patterns, and it is often helpful to split a time series into several components, each representing an underlying pattern category.

In Section 2.3 we discussed three types of time series patterns: trend, seasonality and cycles. When we decompose a time series into components, we usually combine the trend and cycle into a single **trend-cycle** component (sometimes called the **trend** for simplicity). Thus we think of a time series as comprising three components: a trend-cycle component, a seasonal component, and a remainder component (containing anything else in the time series).

In this chapter, we consider some common methods for extracting these components from a time series. Often this is done to help improve understanding of the time series, but it can also be used to improve forecast accuracy.

6.1 Time series components

If we assume an additive decomposition, then we can write

$$y_t = S_t + T_t + R_t,$$

where y_t is the data, S_t is the seasonal component, T_t is the trend-cycle component, and R_t is the remainder component, all at period t. Alternatively, a multiplicative decomposition would be written as

$$y_t = S_t \times T_t \times R_t.$$

The additive decomposition is the most appropriate if the magnitude of the seasonal fluctuations, or the variation around the trend-cycle, does not vary with the level of the time series. When the variation in the seasonal pattern, or the variation around the trend-cycle, appears to be proportional to the level of the time series, then a multiplicative decomposition is more appropriate. Multiplicative decompositions are common with economic time series.

An alternative to using a multiplicative decomposition is to first transform the data until the variation in the series appears to be stable over time, then use an additive decomposition. When a log transformation has been used, this is equivalent to using a multiplicative decomposition because

$$y_t = S_t \times T_t \times R_t \quad \text{is equivalent to} \quad \log y_t = \log S_t + \log T_t + \log R_t.$$

Electrical equipment manufacturing

We will look at several methods for obtaining the components S_t, T_t and R_t later in this chapter, but first, it is helpful to see an example. We will decompose the new orders index for electrical equipment shown in Figure 6.1. The data show the number of new orders for electrical equipment (computer, electronic and optical products) in the Euro area (16 countries). The data have been adjusted by working days and normalised so that a value of 100 corresponds to 2005.

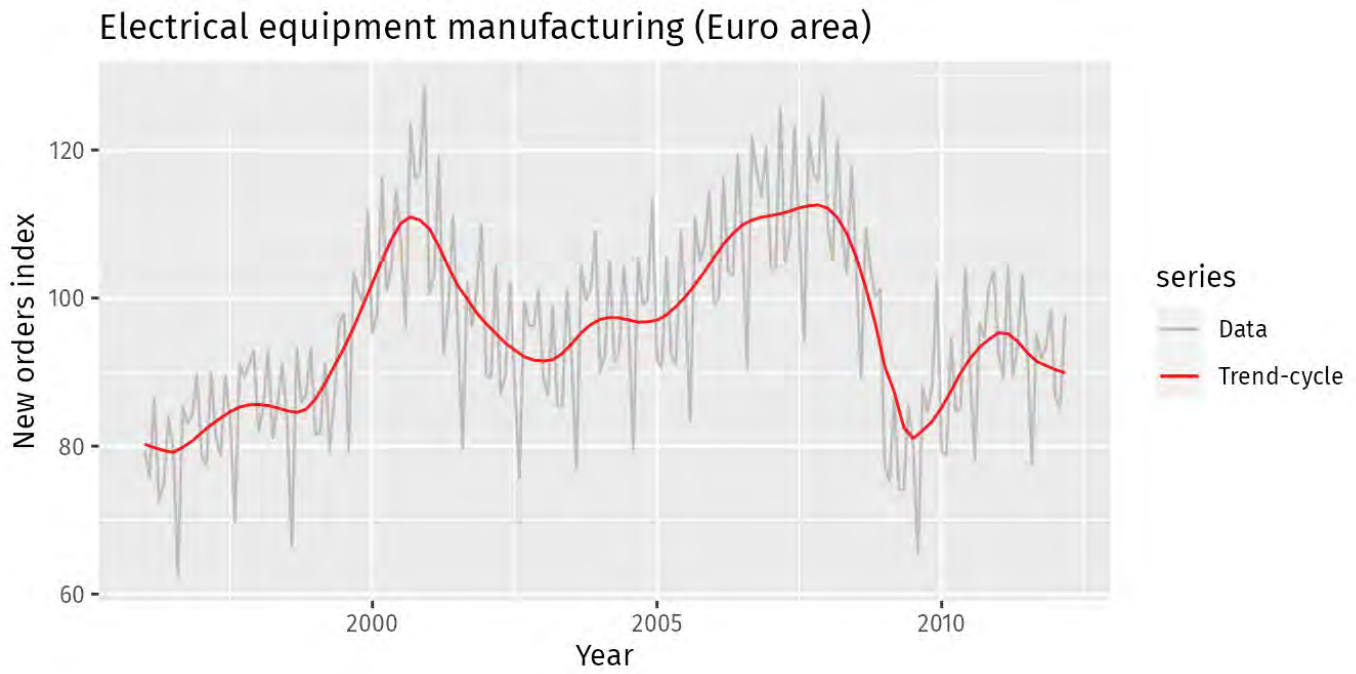


Figure 6.1: Electrical equipment orders: the trend-cycle component (red) and the raw data (grey).

Figure 6.1 shows the trend-cycle component, T_t, in red and the original data, y_t, in grey. The trend-cycle shows the overall movement in the series, ignoring the seasonality and any small random fluctuations.

Figure 6.2 shows an additive decomposition of these data. The method used for estimating components in this example is STL, which is discussed in Section 6.6.

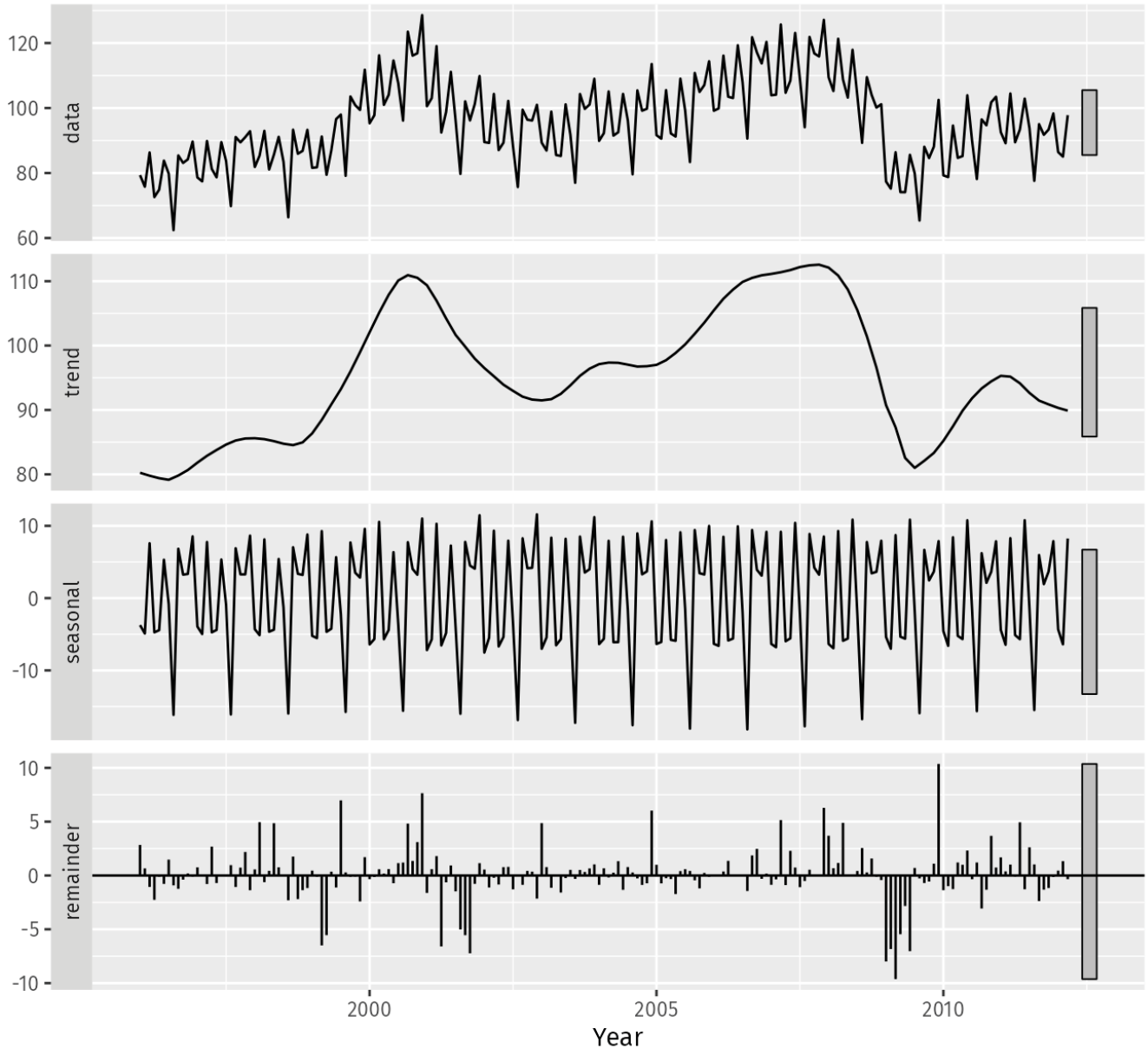


Figure 6.2: The electrical equipment orders (top) and its three additive components.

The three components are shown separately in the bottom three panels of Figure 6.2. These components can be added together to reconstruct the data shown in the top panel. Notice that the seasonal component changes slowly over time, so that any two consecutive years have similar patterns, but years far apart may have different seasonal patterns. The remainder component shown in the bottom panel is what is left over when the seasonal and trend-cycle components have been subtracted from the data.

The grey bars to the right of each panel show the relative scales of the components. Each grey bar represents the same length but because the plots are on different scales, the bars vary in size. The large grey bar in the bottom panel shows that the variation in the remainder component is small compared to the variation in the data,

which has a bar about one quarter the size. If we shrunk the bottom three panels until their bars became the same size as that in the data panel, then all the panels would be on the same scale.

Seasonally adjusted data

If the seasonal component is removed from the original data, the resulting values are the "seasonally adjusted" data. For an additive decomposition, the seasonally adjusted data are given by $y_t - S_t$, and for multiplicative data, the seasonally adjusted values are obtained using y_t / S_t.

Figure 6.3 shows the seasonally adjusted electrical equipment orders.

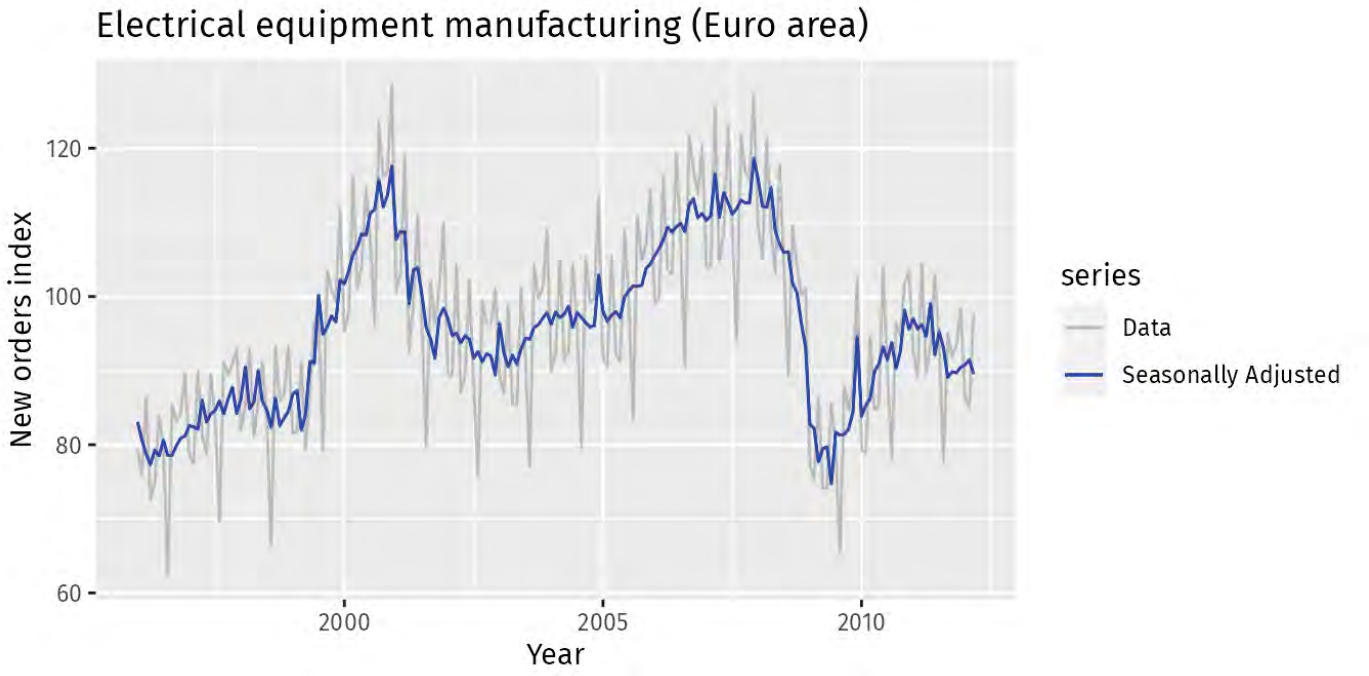


Figure 6.3: Seasonally adjusted electrical equipment orders (blue) and the original data (grey).

If the variation due to seasonality is not of primary interest, the seasonally adjusted series can be useful. For example, monthly unemployment data are usually seasonally adjusted in order to highlight variation due to the underlying state of the economy rather than the seasonal variation. An increase in unemployment due to school leavers seeking work is seasonal variation, while an increase in unemployment due to an economic recession is non-seasonal. Most economic analysts who study unemployment data are more interested in the non-seasonal variation. Consequently, employment data (and many other economic series) are usually seasonally adjusted.

Seasonally adjusted series contain the remainder component as well as the trend-cycle. Therefore, they are not “smooth”, and “downturns” or “upturns” can be misleading. If the purpose is to look for turning points in a series, and interpret any changes in direction, then it is better to use the trend-cycle component rather than the seasonally adjusted data.

6.2 Moving averages

The classical method of time series decomposition originated in the 1920s and was widely used until the 1950s. It still forms the basis of many time series decomposition methods, so it is important to understand how it works. The first step in a classical decomposition is to use a moving average method to estimate the trend-cycle, so we begin by discussing moving averages.

Moving average smoothing

A moving average of order m can be written as

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^{k} y_{t+j}, \tag{6.1}$$

where $m = 2k + 1$. That is, the estimate of the trend-cycle at time t is obtained by averaging values of the time series within k periods of t. Observations that are nearby in time are also likely to be close in value. Therefore, the average eliminates some of the randomness in the data, leaving a smooth trend-cycle component. We call this an m-**MA**, meaning a moving average of order m.

```
autoplot(elecsales) + xlab("Year") + ylab("GWh") +
  ggtitle("Annual electricity sales: South Australia")
```

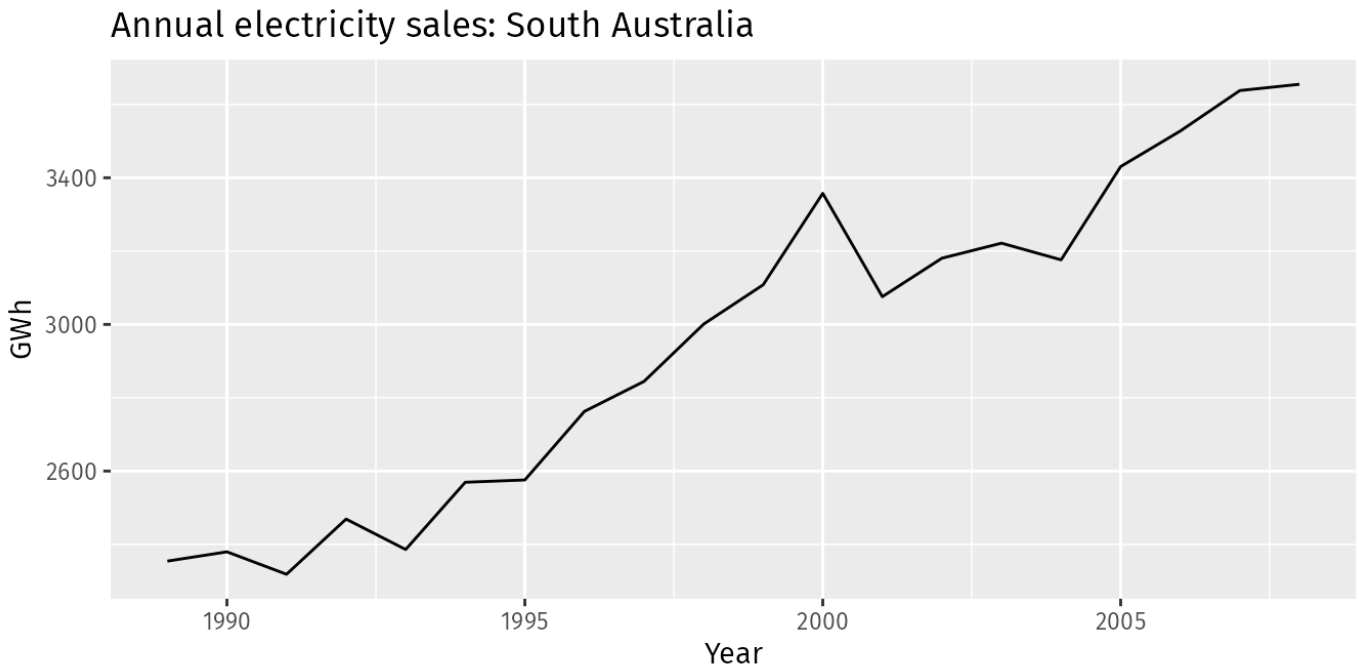


Figure 6.4: Residential electricity sales (excluding hot water) for South Australia: 1989–2008.

For example, consider Figure 6.4 which shows the volume of electricity sold to residential customers in South Australia each year from 1989 to 2008 (hot water sales have been excluded). The data are also shown in Table 6.1.

Table 6.1: Annual electricity sales to residential customers in South Australia. 1989–2008.

Year	Sales (GWh)	5-MA	
1989	2354.34		
1990	2379.71		
1991	2318.52	2381.53	
1992	2468.99	2424.56	
1993	2386.09	2463.76	
1994	2569.47	2552.60	
1995	2575.72	2627.70	
1996	2762.72	2750.62	
1997	2844.50	2858.35	
1998	3000.70	3014.70	
1999	3108.10	3077.30	
2000	3357.50	3144.52	
2001	3075.70	3188.70	
2002	3180.60	3202.32	
2003	3221.60	3216.94	
2004	3176.20	3307.30	
2005	3430.60	3398.75	
2006	3527.48	3485.43	
2007	3637.89		
2008	3655.00		

In the last column of this table, a moving average of order 5 is shown, providing an estimate of the trend-cycle. The first value in this column is the average of the first five observations (1989–1993); the second value in the 5-MA column is the average of the values for 1990–1994; and so on. Each value in the 5-MA column is the average of the observations in the five year window centred on the corresponding year. In the notation of Equation (6.1), column 5-MA contains the values of $\hat{T}_t$ with $k = 2$ and $m = 2k + 1 = 5$. This is easily computed using

```
ma(elecsales, 5)
```

There are no values for either the first two years or the last two years, because we do not have two observations on either side. Later we will use more sophisticated methods of trend-cycle estimation which do allow estimates near the endpoints.

To see what the trend-cycle estimate looks like, we plot it along with the original data in Figure 6.5.

```
autoplot(elecsales, series="Data") +
  autolayer(ma(elecsales,5), series="5-MA") +
  xlab("Year") + ylab("GWh") +
  ggtitle("Annual electricity sales: South Australia") +
  scale_colour_manual(values=c("Data"="grey50","5-MA"="red"),
                      breaks=c("Data","5-MA"))
```

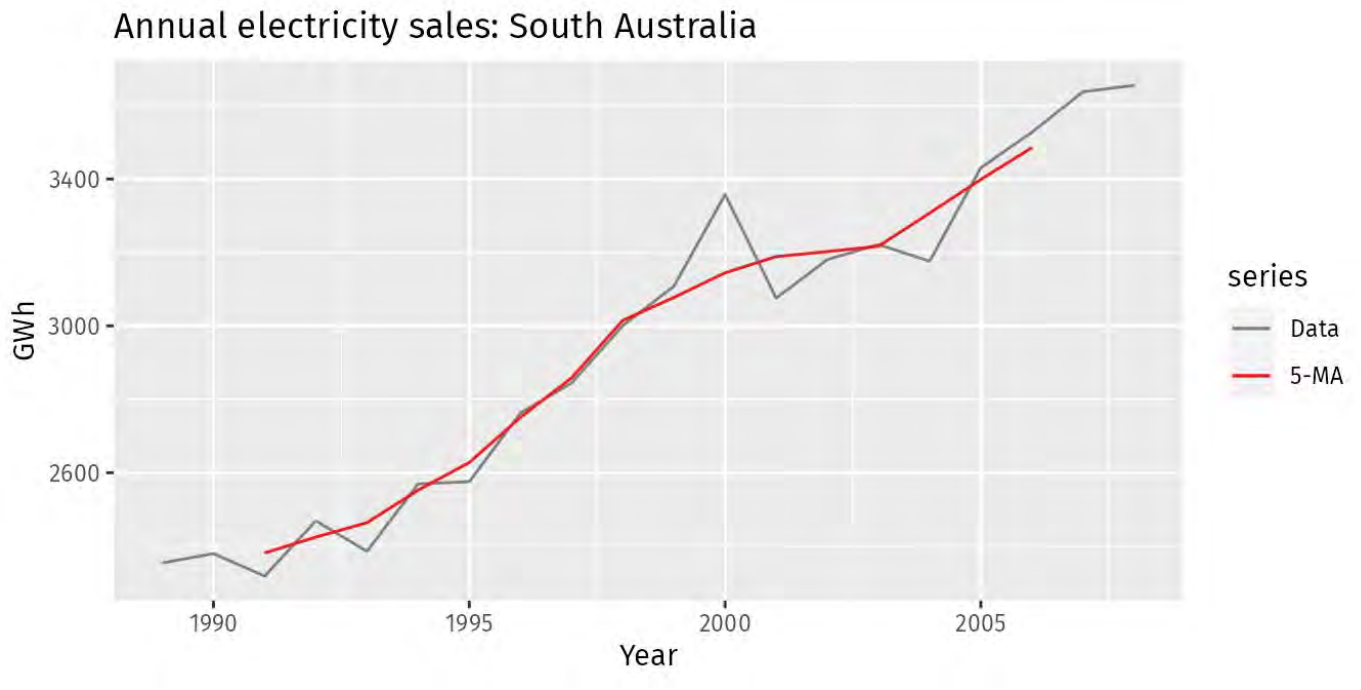


Figure 6.5: Residential electricity sales (black) along with the 5-MA estimate of the trend-cycle (red).

Notice that the trend-cycle (in red) is smoother than the original data and captures the main movement of the time series without all of the minor fluctuations. The order of the moving average determines the smoothness of the trend-cycle estimate. In general, a larger order means a smoother curve. Figure 6.6 shows the effect of changing the order of the moving average for the residential electricity sales data.

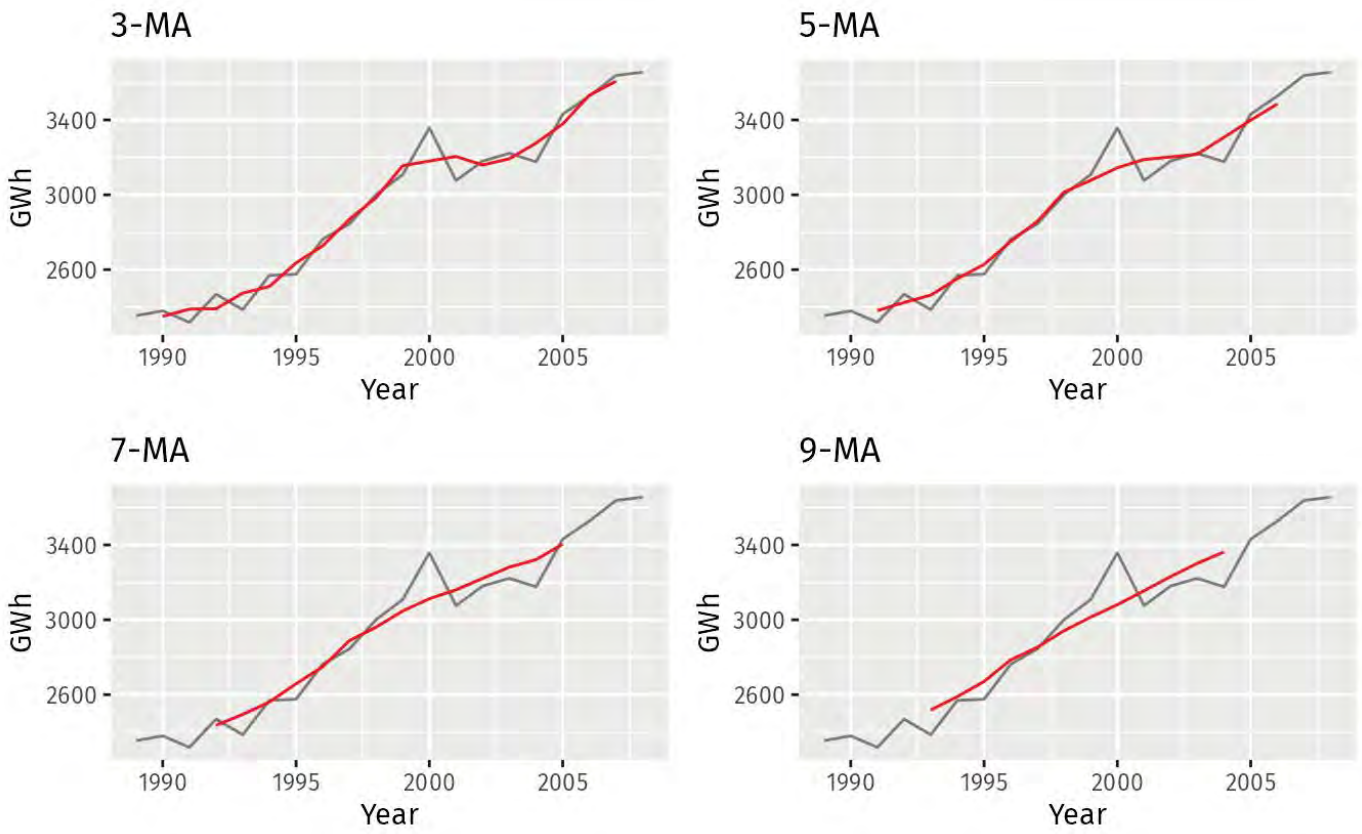


Figure 6.6: Different moving averages applied to the residential electricity sales data.

Simple moving averages such as these are usually of an odd order (e.g., 3, 5, 7, etc.). This is so they are symmetric: in a moving average of order $m = 2k + 1$, the middle observation, and k observations on either side, are averaged. But if m was even, it would no longer be symmetric.

Moving averages of moving averages

It is possible to apply a moving average to a moving average. One reason for doing this is to make an even-order moving average symmetric.

For example, we might take a moving average of order 4, and then apply another moving average of order 2 to the results. In the following table, this has been done for the first few years of the Australian quarterly beer production data.

```
beer2 <- window(ausbeer,start=1992)
ma4 <- ma(beer2, order=4, centre=FALSE)
ma2x4 <- ma(beer2, order=4, centre=TRUE)
```

Table 6.2: A moving average of order 4 applied to the quarterly beer data, followed by a moving average of order 2.

Year	Quarter	Observation	4-MA	2x4-MA
1992	Q1	443		
1992	Q2	410	451.25	
1992	Q3	420	448.75	450.00
1992	Q4	532	451.50	450.12
1993	Q1	433	449.00	450.25
1993	Q2	421	444.00	446.50
1993	Q3	410	448.00	446.00
1993	Q4	512	438.00	443.00
1994	Q1	449	441.25	439.62
1994	Q2	381	446.00	443.62
1994	Q3	423	440.25	443.12
1994	Q4	531	447.00	443.62
1995	Q1	426	445.25	446.12
1995	Q2	408	442.50	443.88
1995	Q3	416	438.25	440.38
1995	Q4	520	435.75	437.00
1996	Q1	409	431.25	433.50
1996	Q2	398	428.00	429.62
1996	Q3	398	433.75	430.88
1996	Q4	507	433.75	433.75

The notation "2×4-MA" in the last column means a 4-MA followed by a 2-MA. The values in the last column are obtained by taking a moving average of order 2 of the values in the previous column. For example, the first two values in the 4-MA column are 451.25=(443+410+420+532)/4 and 448.75=(410+420+532+433)/4. The first value in the 2x4-MA column is the average of these two: 450.00=(451.25+448.75)/2.

When a 2-MA follows a moving average of an even order (such as 4), it is called a "centred moving average of order 4". This is because the results are now symmetric. To see that this is the case, we can write the 2×4-MA as follows:

$$\begin{aligned}\hat{T}_t &= \frac{1}{2}\left[\frac{1}{4}(y_{t-2}+y_{t-1}+y_t+y_{t+1}) + \frac{1}{4}(y_{t-1}+y_t+y_{t+1}+y_{t+2})\right] \\ &= \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}.\end{aligned}$$

It is now a weighted average of observations that is symmetric. By default, the `ma()` function in R will return a centred moving average for even orders (unless `center=FALSE` is specified).

Other combinations of moving averages are also possible. For example, a 3×3-MA is often used, and consists of a moving average of order 3 followed by another moving average of order 3. In general, an even order MA should be followed by an even order MA to make it symmetric. Similarly, an odd order MA should be followed by an odd order MA.

Estimating the trend-cycle with seasonal data

The most common use of centred moving averages is for estimating the trend-cycle from seasonal data. Consider the 2×4-MA:

$$\hat{T}_t = \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}.$$

When applied to quarterly data, each quarter of the year is given equal weight as the first and last terms apply to the same quarter in consecutive years. Consequently, the seasonal variation will be averaged out and the resulting values of $\hat{T}_t$ will have little or no seasonal variation remaining. A similar effect would be obtained using a 2×8-MA or a 2×12-MA to quarterly data.

In general, a $2 \times m$-MA is equivalent to a weighted moving average of order $m + 1$ where all observations take the weight $1/m$, except for the first and last terms which take weights $1/(2m)$. So, if the seasonal period is even and of order m, we use a $2 \times m$-MA to estimate the trend-cycle. If the seasonal period is odd and of order m, we use a m-MA to estimate the trend-cycle. For example, a 2×12-MA can be used to estimate the trend-cycle of monthly data and a 7-MA can be used to estimate the trend-cycle of daily data with a weekly seasonality.

Other choices for the order of the MA will usually result in trend-cycle estimates being contaminated by the seasonality in the data.

Example: Electrical equipment manufacturing

```
autoplot(elecequip, series="Data") +
  autolayer(ma(elecequip, 12), series="12-MA") +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Electrical equipment manufacturing (Euro area)") +
  scale_colour_manual(values=c("Data"="grey","12-MA"="red"),
                      breaks=c("Data","12-MA"))
```

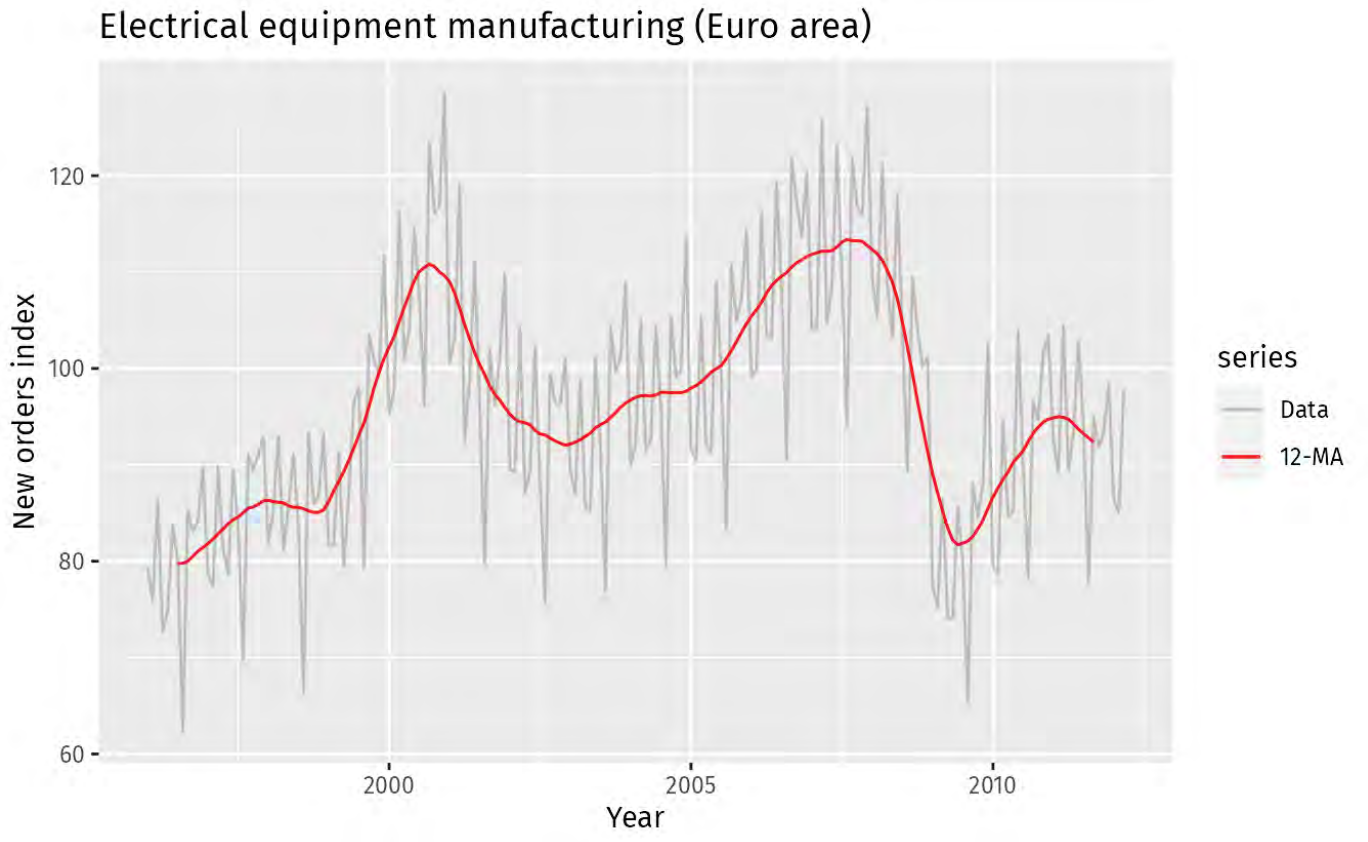


Figure 6.7: A 2x12-MA applied to the electrical equipment orders index.

Figure 6.7 shows a 2×12-MA applied to the electrical equipment orders index. Notice that the smooth line shows no seasonality; it is almost the same as the trend-cycle shown in Figure 6.1, which was estimated using a much more sophisticated method than a moving average. Any other choice for the order of the moving average (except for 24, 36, etc.) would have resulted in a smooth line that showed some seasonal fluctuations.

Weighted moving averages

Combinations of moving averages result in weighted moving averages. For example, the 2×4-MA discussed above is equivalent to a weighted 5-MA with weights given by $\left[\frac{1}{8}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{8}\right]$. In general, a weighted m-MA can be written as

$$\hat{T}_t = \sum_{j=-k}^{k} a_j y_{t+j},$$

where $k = (m-1)/2$, and the weights are given by $[a_{-k}, \dots, a_k]$. It is important that the weights all sum to one and that they are symmetric so that $a_j = a_{-j}$. The simple m-MA is a special case where all of the weights are equal to $1/m$.

A major advantage of weighted moving averages is that they yield a smoother estimate of the trend-cycle. Instead of observations entering and leaving the calculation at full weight, their weights slowly increase and then slowly decrease, resulting in a smoother curve.

6.3 Classical decomposition

The classical decomposition method originated in the 1920s. It is a relatively simple procedure, and forms the starting point for most other methods of time series decomposition. There are two forms of classical decomposition: an additive decomposition and a multiplicative decomposition. These are described below for a time series with seasonal period m (e.g., $m = 4$ for quarterly data, $m = 12$ for monthly data, $m = 7$ for daily data with a weekly pattern).

In classical decomposition, we assume that the seasonal component is constant from year to year. For multiplicative seasonality, the m values that form the seasonal component are sometimes called the "seasonal indices".

Additive decomposition

Step 1

If m is an even number, compute the trend-cycle component $\hat{T}_t$ using a $2 \times m$-MA. If m is an odd number, compute the trend-cycle component $\hat{T}_t$ using an m-MA.

Step 2

Calculate the detrended series: $y_t - \hat{T}_t$.

Step 3

To estimate the seasonal component for each season, simply average the detrended values for that season. For example, with monthly data, the seasonal component for March is the average of all the detrended March values in the data. These seasonal component values are then adjusted to ensure that they add to zero. The seasonal component is obtained by stringing together these monthly values, and then replicating the sequence for each year of data. This gives $\hat{S}_t$.

Step 4

The remainder component is calculated by subtracting the estimated seasonal and trend-cycle components: $\hat{R}_t = y_t - \hat{T}_t - \hat{S}_t$.

Multiplicative decomposition

A classical multiplicative decomposition is similar, except that the subtractions are replaced by divisions.

Step 1

If m is an even number, compute the trend-cycle component $\hat{T}_t$ using a $2 \times m$-MA. If m is an odd number, compute the trend-cycle component $\hat{T}_t$ using an m-MA.

Step 2

Calculate the detrended series: $y_t/\hat{T}_t$.

Step 3

To estimate the seasonal component for each season, simply average the detrended values for that season. For example, with monthly data, the seasonal index for March is the average of all the detrended March values in the data. These seasonal indexes are then adjusted to ensure that they add to m. The seasonal component is obtained by stringing together these monthly indexes, and then replicating the sequence for each year of data. This gives $\hat{S}_t$.

Step 4

The remainder component is calculated by dividing out the estimated seasonal and trend-cycle components: $\hat{R}_t = y_t/(\hat{T}_t\hat{S}_t)$.

Figure 6.8 shows a classical decomposition of the electrical equipment index. Compare this decomposition with that shown in Figure 6.1. The run of remainder values below 1 in 2009 suggests that there is some "leakage" of the trend-cycle component into the remainder component. The trend-cycle estimate has over-smoothed the drop in the data, and the corresponding remainder values have been affected by the poor trend-cycle estimate.

```
elecequip %>% decompose(type="multiplicative") %>%
  autoplot() + xlab("Year") +
  ggtitle("Classical multiplicative decomposition
    of electrical equipment index")
```

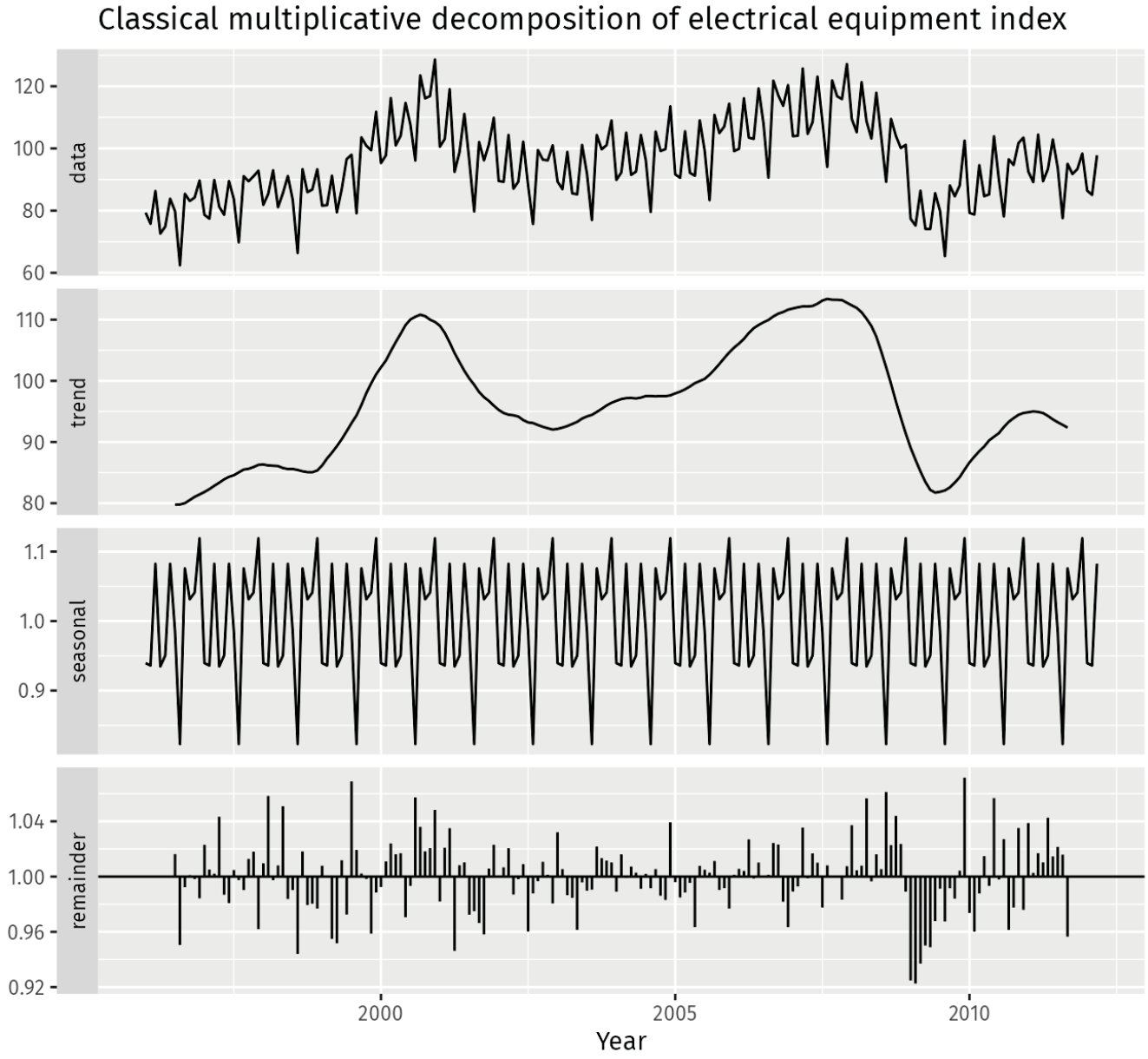


Figure 6.8: A classical multiplicative decomposition of the new orders index for electrical equipment.

Comments on classical decomposition

While classical decomposition is still widely used, it is not recommended, as there are now several much better methods. Some of the problems with classical decomposition are summarised below.

- The estimate of the trend-cycle is unavailable for the first few and last few observations. For example, if $m = 12$, there is no trend-cycle estimate for the first six or the last six observations. Consequently, there is also no estimate of the remainder component for the same time periods.
- The trend-cycle estimate tends to over-smooth rapid rises and falls in the data (as seen in the above example).

- Classical decomposition methods assume that the seasonal component repeats from year to year. For many series, this is a reasonable assumption, but for some longer series it is not. For example, electricity demand patterns have changed over time as air conditioning has become more widespread. Specifically, in many locations, the seasonal usage pattern from several decades ago had its maximum demand in winter (due to heating), while the current seasonal pattern has its maximum demand in summer (due to air conditioning). The classical decomposition methods are unable to capture these seasonal changes over time.
- Occasionally, the values of the time series in a small number of periods may be particularly unusual. For example, the monthly air passenger traffic may be affected by an industrial dispute, making the traffic during the dispute different from usual. The classical method is not robust to these kinds of unusual values.

6.4 X11 decomposition

Another popular method for decomposing quarterly and monthly data is the X11 method which originated in the US Census Bureau and Statistics Canada.

This method is based on classical decomposition, but includes many extra steps and features in order to overcome the drawbacks of classical decomposition that were discussed in the previous section. In particular, trend-cycle estimates are available for all observations including the end points, and the seasonal component is allowed to vary slowly over time. X11 also has some sophisticated methods for handling trading day variation, holiday effects and the effects of known predictors. It handles both additive and multiplicative decomposition. The process is entirely automatic and tends to be highly robust to outliers and level shifts in the time series.

The details of the X11 method are described in Dagum & Bianconcini (2016). Here we will only demonstrate how to use the automatic procedure in R.

The X11 method is available using the `seas()` function from the **seasonal** package for R.

```
library(seasonal)
elecequip %>% seas(x11="") -> fit
autoplot(fit) +
  ggtitle("X11 decomposition of electrical equipment index")
```

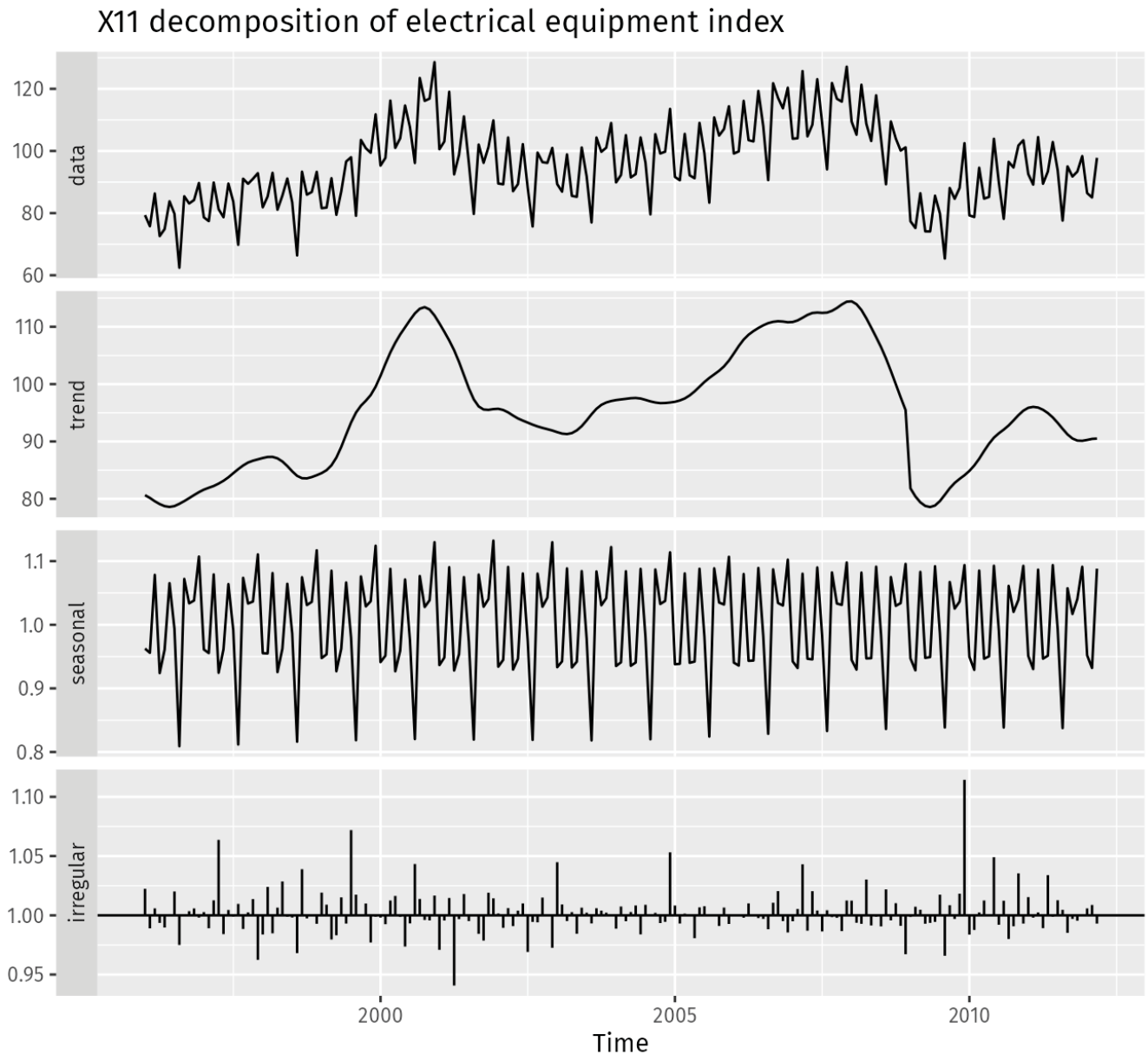


Figure 6.9: An X11 decomposition of the new orders index for electrical equipment.

Compare this decomposition with the STL decomposition shown in Figure 6.1 and the classical decomposition shown in Figure 6.8. The X11 trend-cycle has captured the sudden fall in the data in early 2009 better than either of the other two methods, and the unusual observation at the end of 2009 is now more clearly seen in the remainder component.

Given the output from the `seas()` function, `seasonal()` will extract the seasonal component, `trendcycle()` will extract the trend-cycle component, `remainder()` will extract the remainder component, and `seasadj()` will compute the seasonally adjusted time series.

For example, Figure 6.10 shows the trend-cycle component and the seasonally adjusted data, along with the original data.

```
autoplot(elecequip, series="Data") +
  autolayer(trendcycle(fit), series="Trend") +
  autolayer(seasadj(fit), series="Seasonally Adjusted") +
  xlab("Year") + ylab("New orders index") +
  ggtitle("Electrical equipment manufacturing (Euro area)") +
  scale_colour_manual(values=c("gray","blue","red"),
             breaks=c("Data","Seasonally Adjusted","Trend"))
```

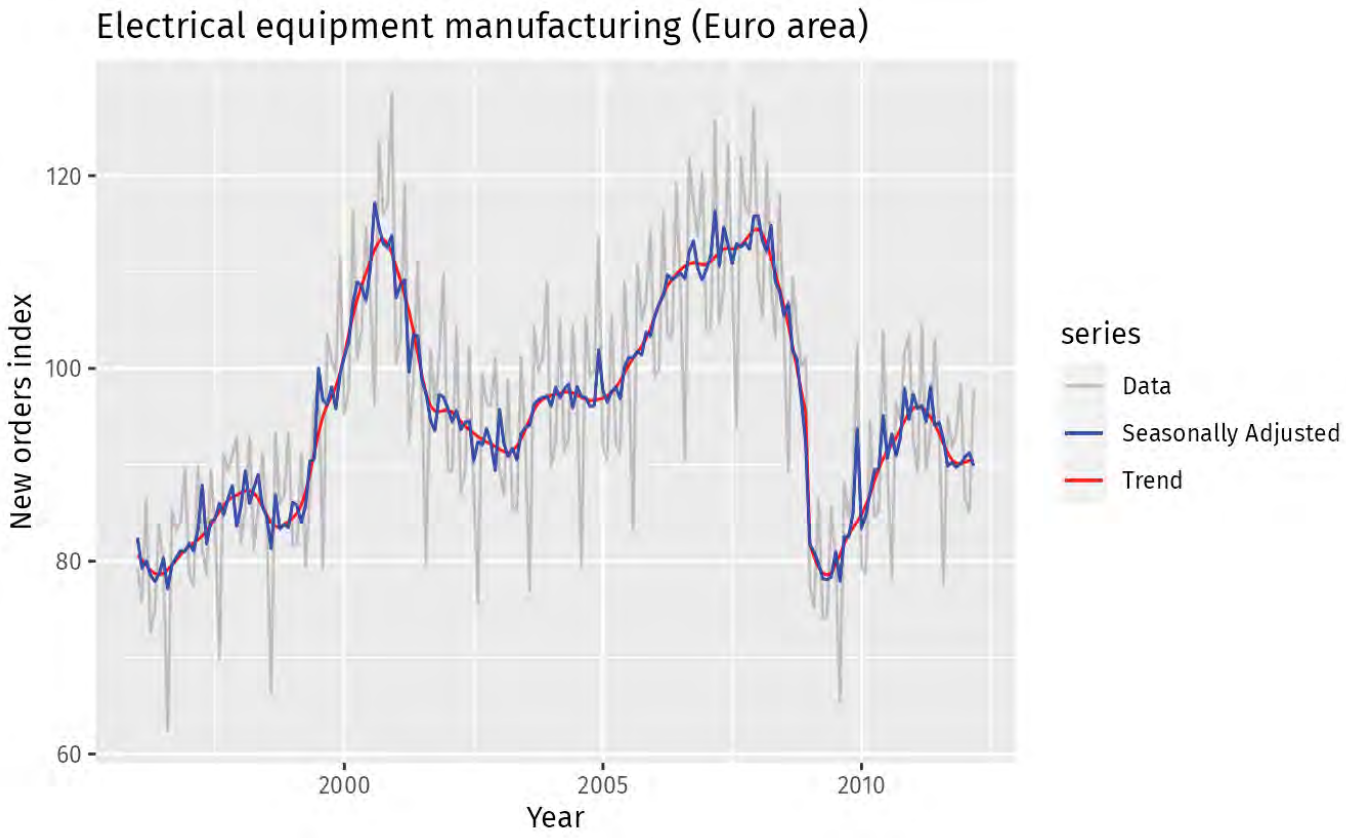


Figure 6.10: Electrical equipment orders: the original data (grey), the trend-cycle component (red) and the seasonally adjusted data (blue).

It can be useful to use seasonal plots and seasonal sub-series plots of the seasonal component. These help us to visualise the variation in the seasonal component over time. Figure 6.11 shows a seasonal sub-series plot of the seasonal component from Figure 6.9. In this case, there are only small changes over time.

```
fit %>% seasonal() %>% ggsubseriesplot() + ylab("Seasonal")
```

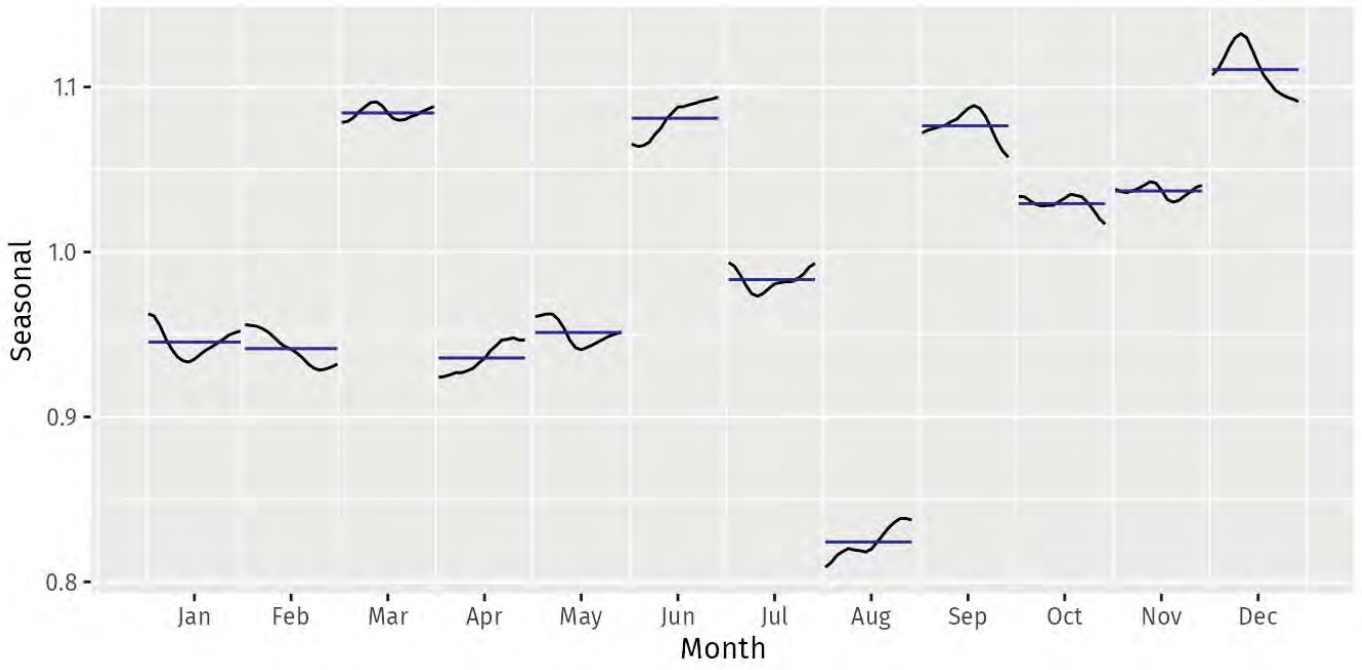


Figure 6.11: Seasonal sub-series plot of the seasonal component from the X11 decomposition of the new orders index for electrical equipment.

Bibliography

Dagum, E. B., & Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation*. Springer. [Amazon]

6.5 SEATS decomposition

"SEATS" stands for "Seasonal Extraction in ARIMA Time Series" (ARIMA models are discussed in Chapter 8). This procedure was developed at the Bank of Spain, and is now widely used by government agencies around the world. The procedure works only with quarterly and monthly data. So seasonality of other kinds, such as daily data, or hourly data, or weekly data, require an alternative approach.

The details are beyond the scope of this book. However, a complete discussion of the method is available in Dagum & Bianconcini (2016). Here we will only demonstrate how to use it via the **seasonal** package.

```
library(seasonal)
elecequip %>% seas() %>%
autoplot() +
  ggtitle("SEATS decomposition of electrical equipment index")
```

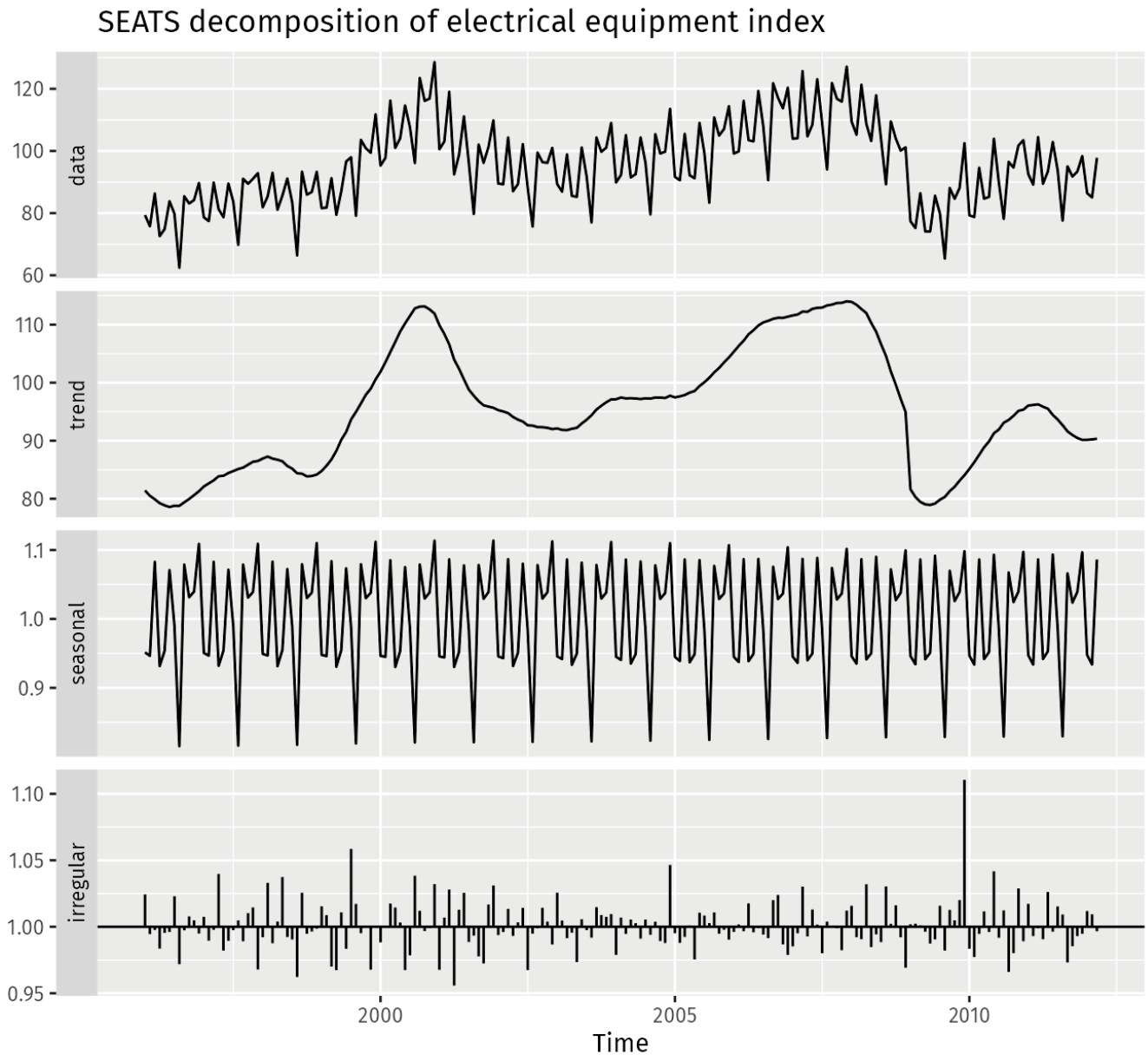


Figure 6.12: A SEATS decomposition of the new orders index for electrical equipment.

The result is quite similar to the X11 decomposition shown in Figure 6.9.

As with the X11 method, we can use the `seasonal()`, `trendcycle()` and `remainder()` functions to extract the individual components, and `seasadj()` to compute the seasonally adjusted time series.

The **seasonal** package has many options for handling variations of X11 and SEATS. See the package website for a detailed introduction to the options and features available.

Bibliography

Dagum, E. B., & Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation*. Springer. [Amazon]

6.6 STL decomposition

STL is a versatile and robust method for decomposing time series. STL is an acronym for "Seasonal and Trend decomposition using Loess", while Loess is a method for estimating nonlinear relationships. The STL method was developed by R. B. Cleveland, Cleveland, McRae, & Terpenning (1990).

STL has several advantages over the classical, SEATS and X11 decomposition methods:

- Unlike SEATS and X11, STL will handle any type of seasonality, not only monthly and quarterly data.
- The seasonal component is allowed to change over time, and the rate of change can be controlled by the user.
- The smoothness of the trend-cycle can also be controlled by the user.
- It can be robust to outliers (i.e., the user can specify a robust decomposition), so that occasional unusual observations will not affect the estimates of the trend-cycle and seasonal components. They will, however, affect the remainder component.

On the other hand, STL has some disadvantages. In particular, it does not handle trading day or calendar variation automatically, and it only provides facilities for additive decompositions.

It is possible to obtain a multiplicative decomposition by first taking logs of the data, then back-transforming the components. Decompositions between additive and multiplicative can be obtained using a Box-Cox transformation of the data with $0 < \lambda < 1$. A value of $\lambda = 0$ corresponds to the multiplicative decomposition while $\lambda = 1$ is equivalent to an additive decomposition.

The best way to begin learning how to use STL is to see some examples and experiment with the settings. Figure 6.2 showed an example of STL applied to the electrical equipment orders data. Figure 6.13 shows an alternative STL decomposition where the trend-cycle is more flexible, the seasonal component does not change over time, and the robust option has been used. Here, it is more obvious

that there has been a down-turn at the end of the series, and that the orders in 2009 were unusually low (corresponding to some large negative values in the remainder component).

```
elecequip %>%
  stl(t.window=13, s.window="periodic", robust=TRUE) %>%
  autoplot()
```

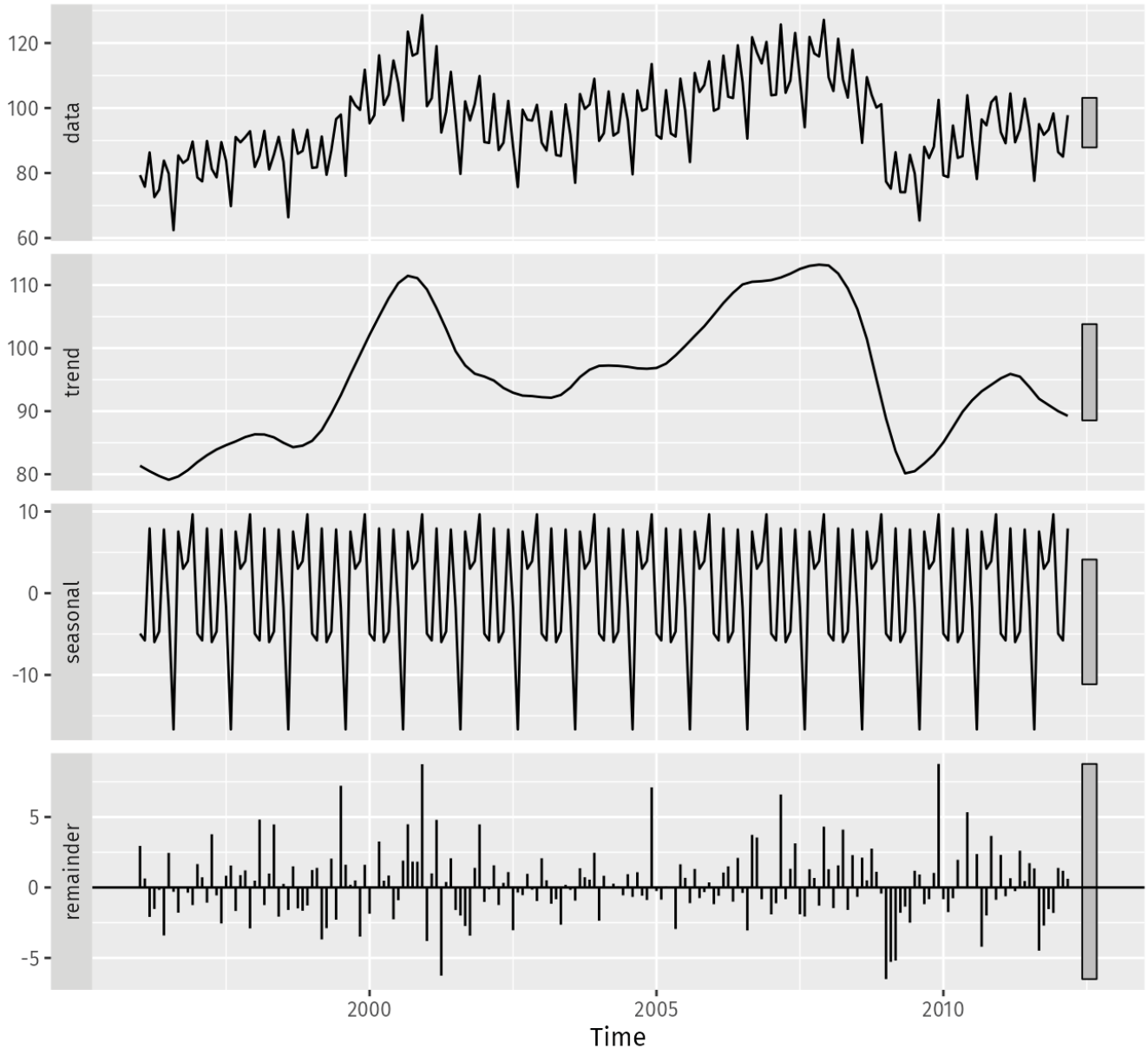


Figure 6.13: The electrical equipment orders (top) and its three additive components obtained from a robust STL decomposition with flexible trend-cycle and fixed seasonality.

The two main parameters to be chosen when using STL are the trend-cycle window (`t.window`) and the seasonal window (`s.window`). These control how rapidly the trend-cycle and seasonal components can change. Smaller values allow for more rapid changes. Both `t.window` and `s.window` should be odd numbers; `t.window` is

the number of consecutive observations to be used when estimating the trend-cycle; `s.window` is the number of consecutive years to be used in estimating each value in the seasonal component. The user must specify `s.window` as there is no default. Setting it to be infinite is equivalent to forcing the seasonal component to be periodic (i.e., identical across years). Specifying `t.window` is optional, and a default value will be used if it is omitted.

The `mstl()` function provides a convenient automated STL decomposition using `s.window=13`, and `t.window` also chosen automatically. This usually gives a good balance between overfitting the seasonality and allowing it to slowly change over time. But, as with any automated procedure, the default settings will need adjusting for some time series.

As with the other decomposition methods discussed in this book, to obtain the separate components plotted in Figure 6.8, use the `seasonal()` function for the seasonal component, the `trendcycle()` function for trend-cycle component, and the `remainder()` function for the remainder component. The `seasadj()` function can be used to compute the seasonally adjusted series.

Bibliography

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. J. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, *6*(1), 3–33. http://bit.ly/stl1990

6.7 Measuring strength of trend and seasonality

A time series decomposition can be used to measure the strength of trend and seasonality in a time series (Wang, Smith, & Hyndman, 2006). Recall that the decomposition is written as

$$y_t = T_t + S_t + R_t,$$

where T_t is the smoothed trend component, S_t is the seasonal component and R_t is a remainder component. For strongly trended data, the seasonally adjusted data should have much more variation than the remainder component. Therefore $\text{Var}(R_t)$ /$\text{Var}(T_t + R_t)$ should be relatively small. But for data with little or no trend, the two variances should be approximately the same. So we define the strength of trend as:

$$F_T = \max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(T_t + R_t)}\right).$$

This will give a measure of the strength of the trend between 0 and 1. Because the variance of the remainder might occasionally be even larger than the variance of the seasonally adjusted data, we set the minimal possible value of F_T equal to zero.

The strength of seasonality is defined similarly, but with respect to the detrended data rather than the seasonally adjusted data:

$$F_S = \max\left(0, 1 - \frac{\text{Var}(R_t)}{\text{Var}(S_t + R_t)}\right).$$

A series with seasonal strength F_S close to 0 exhibits almost no seasonality, while a series with strong seasonality will have F_S close to 1 because $\text{Var}(R_t)$ will be much smaller than $\text{Var}(S_t + R_t)$.

These measures can be useful, for example, when you have a large collection of time series, and you need to find the series with the most trend or the most seasonality.

Bibliography

Wang, X., Smith, K. A., & Hyndman, R. J. (2006). Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery*, *13*(3), 335–364.

[DOI]

6.8 Forecasting with decomposition

While decomposition is primarily useful for studying time series data, and exploring historical changes over time, it can also be used in forecasting.

Assuming an additive decomposition, the decomposed time series can be written as

$$y_t = \hat{S}_t + \hat{A}_t,$$

where $\hat{A}_t = \hat{T}_t + \hat{R}_t$ is the seasonally adjusted component. Or, if a multiplicative decomposition has been used, we can write

$$y_t = \hat{S}_t\hat{A}_t,$$

where $\hat{A}_t = \hat{T}_t\hat{R}_t$.

To forecast a decomposed time series, we forecast the seasonal component, $\hat{S}_t$, and the seasonally adjusted component $\hat{A}_t$, separately. It is usually assumed that the seasonal component is unchanging, or changing extremely slowly, so it is forecast by simply taking the last year of the estimated component. In other words, a seasonal naïve method is used for the seasonal component.

To forecast the seasonally adjusted component, any non-seasonal forecasting method may be used. For example, a random walk with drift model, or Holt's method (discussed in the next chapter), or a non-seasonal ARIMA model (discussed in Chapter 8), may be used.

Example: Electrical equipment manufacturing

```
fit <- stl(elecequip, t.window=13, s.window="periodic",
  robust=TRUE)
fit %>% seasadj() %>% naive() %>%
  autoplot() + ylab("New orders index") +
  ggtitle("Naive forecasts of seasonally adjusted data")
```

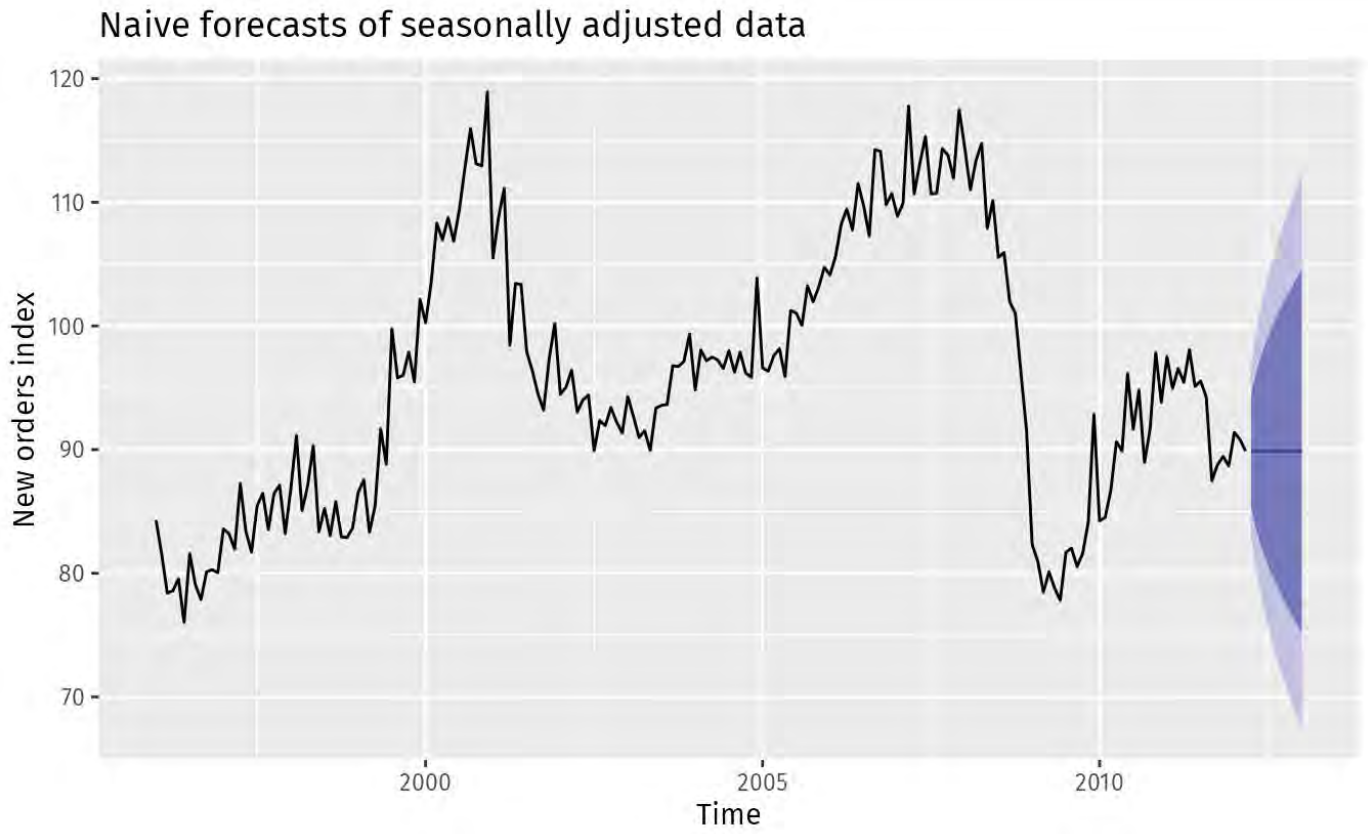


Figure 6.14: Naïve forecasts of the seasonally adjusted data obtained from an STL decomposition of the electrical equipment orders data.

Figure 6.14 shows naïve forecasts of the seasonally adjusted electrical equipment orders data. These are then "reseasonalised" by adding in the seasonal naïve forecasts of the seasonal component.

This is made easy with the `forecast()` function applied to the `stl` object. You need to specify the method being used on the seasonally adjusted data, and the function will do the reseasonalising for you. The resulting forecasts of the original data are shown in Figure 6.15.

```
fit %>% forecast(method="naive") %>%
  autoplot() + ylab("New orders index")
```

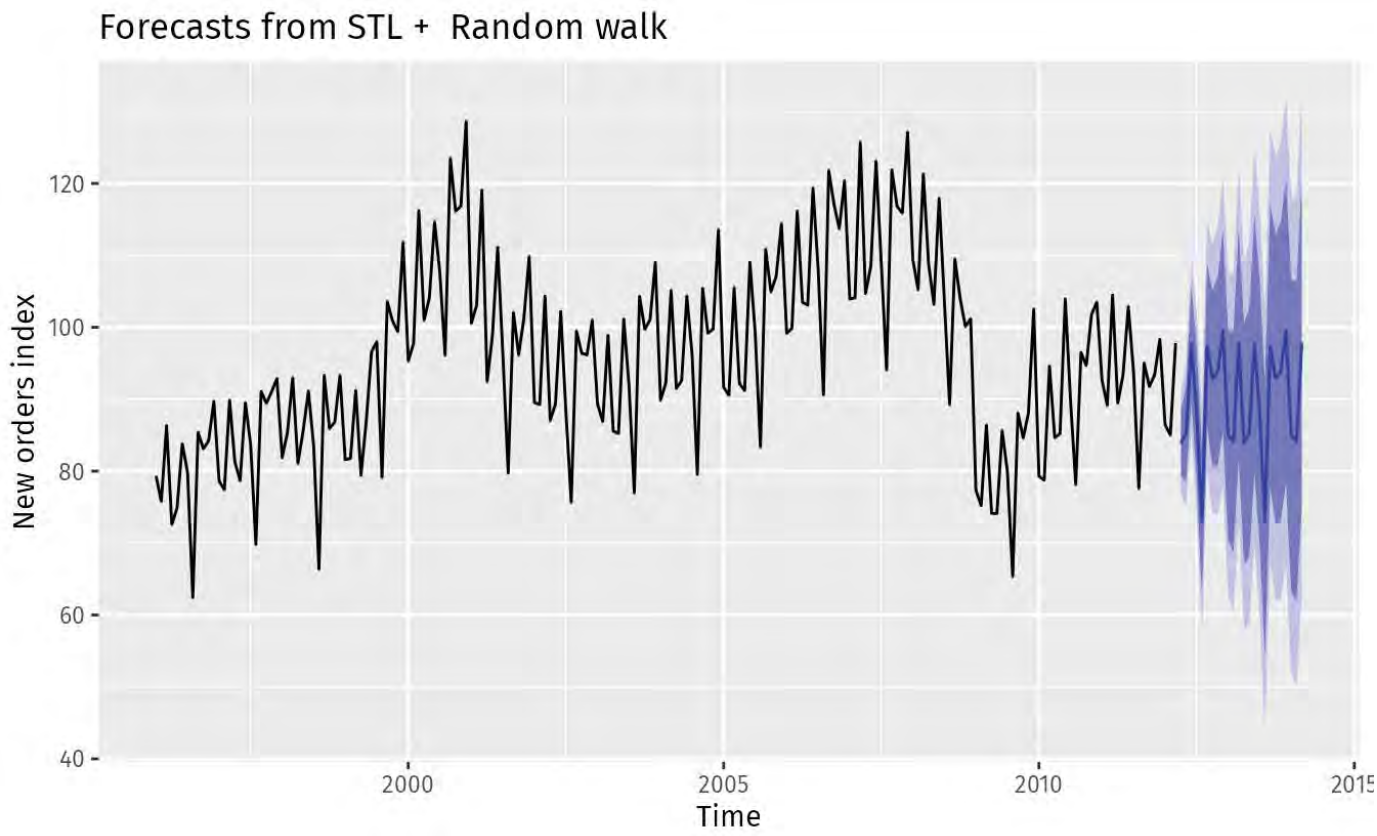


Figure 6.15: Forecasts of the electrical equipment orders data based on a naïve forecast of the seasonally adjusted data and a seasonal naïve forecast of the seasonal component, after an STL decomposition of the data.

The prediction intervals shown in this graph are constructed in the same way as the point forecasts. That is, the upper and lower limits of the prediction intervals on the seasonally adjusted data are "reseasonalised" by adding in the forecasts of the seasonal component. In this calculation, the uncertainty in the forecasts of the seasonal component has been ignored. The rationale for this choice is that the uncertainty in the seasonal component is much smaller than that for the seasonally adjusted data, and so it is a reasonable approximation to ignore it.

A short-cut approach is to use the `stlf()` function. The following code will decompose the time series using STL, forecast the seasonally adjusted series, and return the reseasonalised forecasts.

```
fcast <- stlf(elecequip, method='naive')
```

The `stlf()` function uses `mstl()` to carry out the decomposition, so there are default values for `s.window` and `t.window`.

As well as the naïve method, several other possible forecasting methods are available with `stlf()`, as described in the corresponding help file. If `method` is not specified, it will use the ETS approach (discussed in the next chapter) applied to the seasonally

adjusted series. This usually produces quite good forecasts for seasonal time series, and some companies use it routinely for all their operational forecasts.

6.9 Exercises

1. Show that a 3×5 MA is equivalent to a 7-term weighted moving average with weights of 0.067, 0.133, 0.200, 0.200, 0.200, 0.133, and 0.067.

2. The `plastics` data set consists of the monthly sales (in thousands) of product A for a plastics manufacturer for five years.

    a. Plot the time series of sales of product A. Can you identify seasonal fluctuations and/or a trend-cycle?
    b. Use a classical multiplicative decomposition to calculate the trend-cycle and seasonal indices.
    c. Do the results support the graphical interpretation from part a?
    d. Compute and plot the seasonally adjusted data.
    e. Change one observation to be an outlier (e.g., add 500 to one observation), and recompute the seasonally adjusted data. What is the effect of the outlier?
    f. Does it make any difference if the outlier is near the end rather than in the middle of the time series?

3. Recall your retail time series data (from Exercise 3 in Section 2.10). Decompose the series using X11. Does it reveal any outliers, or unusual features that you had not noticed previously?

4. Figures 6.16 and 6.17 show the result of decomposing the number of persons in the civilian labour force in Australia each month from February 1978 to August 1995.

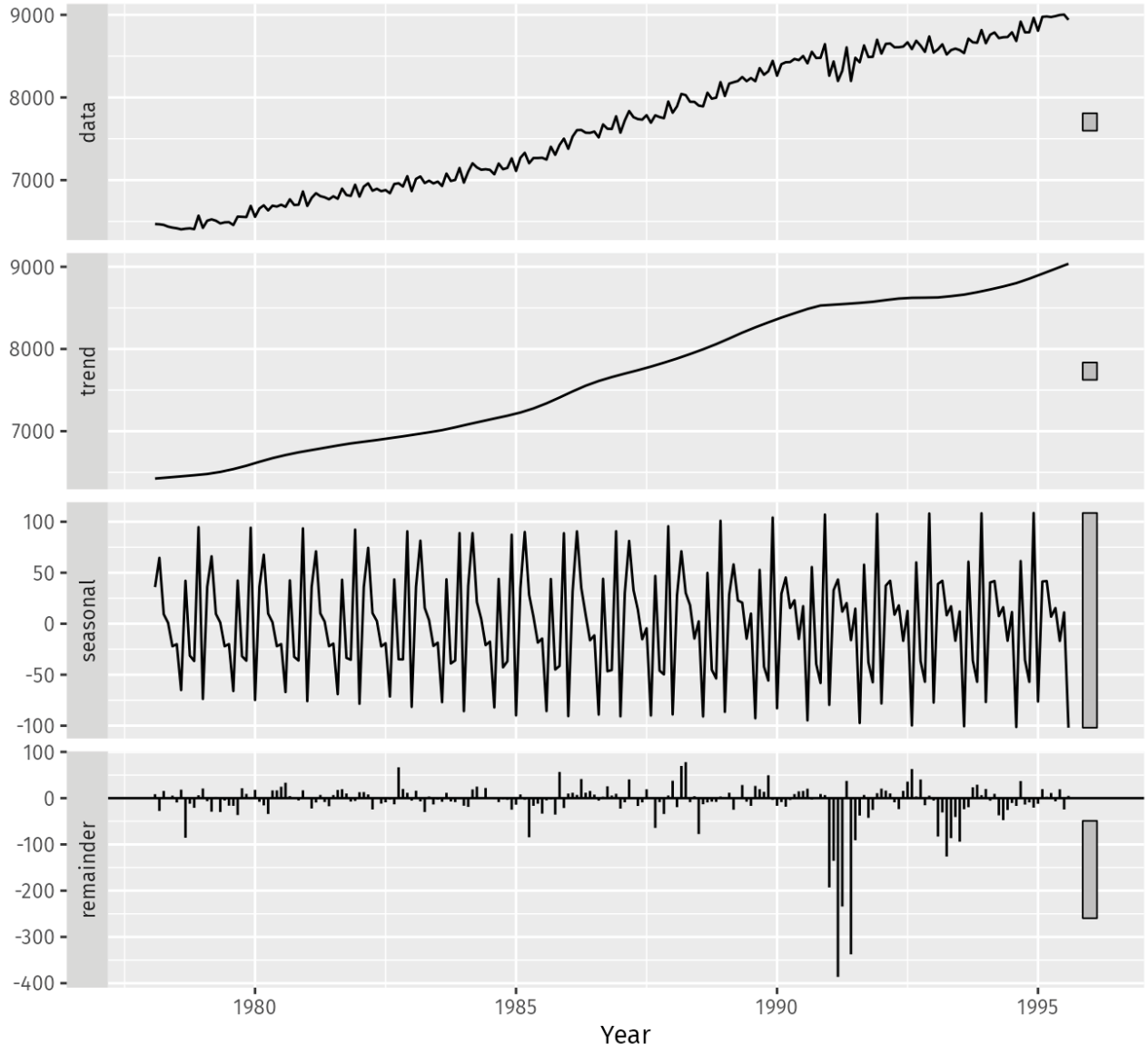


Figure 6.16: Decomposition of the number of persons in the civilian labour force in Australia each month from February 1978 to August 1995.

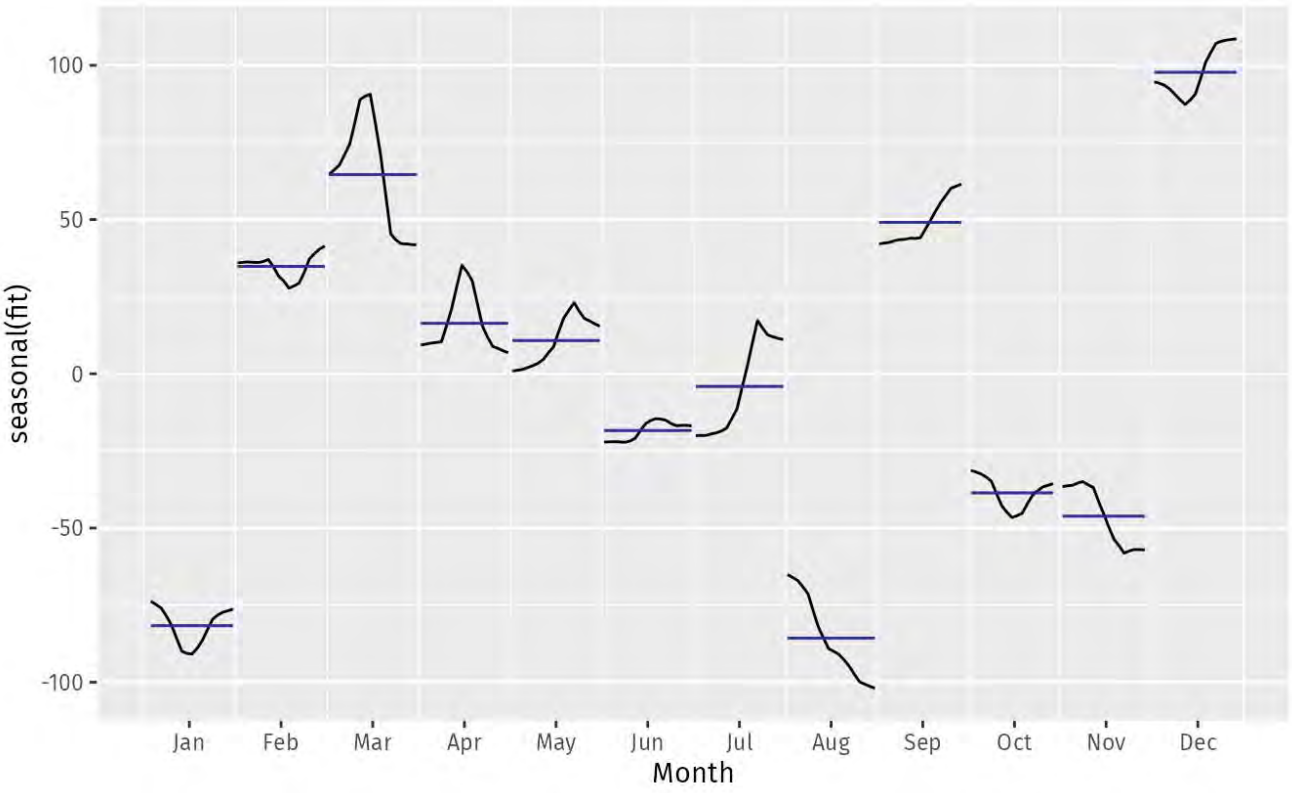


Figure 6.17: Seasonal component from the decomposition shown in the previous figure.

a. Write about 3–5 sentences describing the results of the decomposition. Pay particular attention to the scales of the graphs in making your interpretation.
b. Is the recession of 1991/1992 visible in the estimated components?

5. This exercise uses the `cangas` data (monthly Canadian gas production in billions of cubic metres, January 1960 – February 2005).

   a. Plot the data using `autoplot()`, `ggsubseriesplot()` and `ggseasonplot()` to look at the effect of the changing seasonality over time. What do you think is causing it to change so much?
   b. Do an STL decomposition of the data. You will need to choose `s.window` to allow for the changing shape of the seasonal component.
   c. Compare the results with those obtained using SEATS and X11. How are they different?

6. We will use the `bricksq` data (Australian quarterly clay brick production. 1956–1994) for this exercise.

   a. Use an STL decomposition to calculate the trend-cycle and seasonal indices. (Experiment with having fixed or changing seasonality.)
   b. Compute and plot the seasonally adjusted data.
   c. Use a naïve method to produce forecasts of the seasonally adjusted data.

d. Use `stlf()` to reseasonalise the results, giving forecasts for the original data.
e. Do the residuals look uncorrelated?
f. Repeat with a robust STL decomposition. Does it make much difference?
g. Compare forecasts from `stlf()` with those from `snaive()`, using a test set comprising the last 2 years of data. Which is better?

7. Use `stlf()` to produce forecasts of the `writing` series with either `method="naive"` or `method="rwdrift"`, whichever is most appropriate. Use the `lambda` argument if you think a Box-Cox transformation is required.

8. Use `stlf()` to produce forecasts of the `fancy` series with either `method="naive"` or `method="rwdrift"`, whichever is most appropriate. Use the `lambda` argument if you think a Box-Cox transformation is required.

6.10 Further reading

- A detailed modern discussion of SEATS and X11 decomposition methods is provided by Dagum & Bianconcini (2016).
- R. B. Cleveland et al. (1990) introduced STL, and still provides the best description of the algorithm.
- For a discussion of forecasting using STL, see Theodosiou (2011).

Bibliography

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. J. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, *6*(1), 3–33. http://bit.ly/stl1990

Dagum, E. B., & Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation*. Springer. [Amazon]

Theodosiou, M. (2011). Forecasting monthly and quarterly time series using STL decomposition. *International Journal of Forecasting*, *27*(4), 1178–1195. [DOI]

Chapter 7 Exponential smoothing

Exponential smoothing was proposed in the late 1950s (Brown, 1959; Holt, 1957; Winters, 1960), and has motivated some of the most successful forecasting methods. Forecasts produced using exponential smoothing methods are weighted averages of past observations, with the weights decaying exponentially as the observations get older. In other words, the more recent the observation the higher the associated weight. This framework generates reliable forecasts quickly and for a wide range of time series, which is a great advantage and of major importance to applications in industry.

This chapter is divided into two parts. In the first part (Sections 7.1–7.4) we present the mechanics of the most important exponential smoothing methods, and their application in forecasting time series with various characteristics. This helps us develop an intuition to how these methods work. In this setting, selecting and using a forecasting method may appear to be somewhat ad hoc. The selection of the method is generally based on recognising key components of the time series (trend and seasonal) and the way in which these enter the smoothing method (e.g., in an additive, damped or multiplicative manner).

In the second part of the chapter (Sections 7.5–7.7) we present the statistical models that underlie exponential smoothing methods. These models generate identical point forecasts to the methods discussed in the first part of the chapter, but also generate prediction intervals. Furthermore, this statistical framework allows for genuine model selection between competing models.

Bibliography

Brown, R. G. (1959). *Statistical forecasting for inventory control*. McGraw/Hill.

Holt, C. C. (1957). *Forecasting seasonals and trends by exponentially weighted averages* (O.N.R. Memorandum No. 52). Carnegie Institute of Technology, Pittsburgh USA. [DOI]

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, *6*(3), 324–342. [DOI]

7.1 Simple exponential smoothing

The simplest of the exponentially smoothing methods is naturally called **simple exponential smoothing** (SES)[14]. This method is suitable for forecasting data with no clear trend or seasonal pattern. For example, the data in Figure 7.1 do not display any clear trending behaviour or any seasonality. (There is a rise in the last few years, which might suggest a trend. We will consider whether a trended method would be better for this series later in this chapter.) We have already considered the naïve and the average as possible methods for forecasting such data (Section 3.1).

```
oildata <- window(oil, start=1996)
autoplot(oildata) +
  ylab("Oil (millions of tonnes)") + xlab("Year")
```

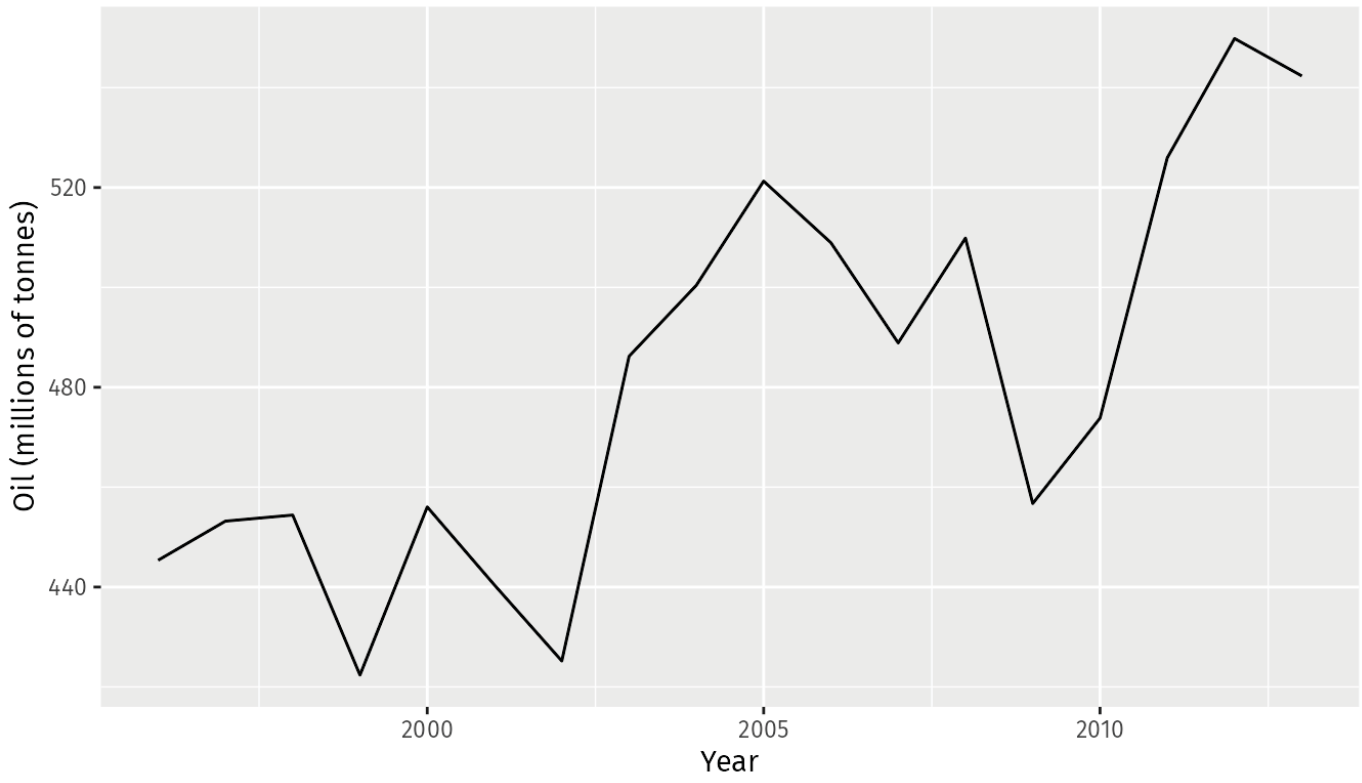


Figure 7.1: Oil production in Saudi Arabia from 1996 to 2013.

Using the naïve method, all forecasts for the future are equal to the last observed value of the series,

$$\hat{y}_{T+h|T} = y_T,$$

for $h = 1, 2, \ldots$. Hence, the naïve method assumes that the most recent observation is the only important one, and all previous observations provide no information for the future. This can be thought of as a weighted average where all of the weight is given to the last observation.

Using the average method, all future forecasts are equal to a simple average of the observed data,

$$\hat{y}_{T+h|T} = \frac{1}{T}\sum_{t=1}^{T} y_t,$$

for $h = 1, 2, \ldots$. Hence, the average method assumes that all observations are of equal importance, and gives them equal weights when generating forecasts.

We often want something between these two extremes. For example, it may be sensible to attach larger weights to more recent observations than to observations from the distant past. This is exactly the concept behind simple exponential smoothing. Forecasts are calculated using weighted averages, where the weights decrease exponentially as observations come from further in the past — the smallest weights are associated with the oldest observations:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots, \tag{7.1}$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter. The one-step-ahead forecast for time $T+1$ is a weighted average of all of the observations in the series $y_1, \ldots, y_T$. The rate at which the weights decrease is controlled by the parameter α.

The table below shows the weights attached to observations for four different values of α when forecasting using simple exponential smoothing. Note that the sum of the weights even for a small value of α will be approximately one for any reasonable sample size.

	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
y_T	0.2000	0.4000	0.6000	0.8000
y_{T-1}	0.1600	0.2400	0.2400	0.1600
y_{T-2}	0.1280	0.1440	0.0960	0.0320
y_{T-3}	0.1024	0.0864	0.0384	0.0064
y_{T-4}	0.0819	0.0518	0.0154	0.0013
y_{T-5}	0.0655	0.0311	0.0061	0.0003

For any α between 0 and 1, the weights attached to the observations decrease exponentially as we go back in time, hence the name "exponential smoothing". If α is small (i.e., close to 0), more weight is given to observations from the more distant past. If α is large (i.e., close to 1), more weight is given to the more recent observations. For the extreme case where $\alpha = 1$, $\hat{y}_{T+1|T} = y_T$, and the forecasts are equal to the naïve forecasts.

We present two equivalent forms of simple exponential smoothing, each of which leads to the forecast Equation (7.1).

Weighted average form

The forecast at time $T+1$ is equal to a weighted average between the most recent observation y_T and the previous forecast $\hat{y}_{T|T-1}$:

$$\hat{y}_{T+1|T} = \alpha y_T + (1-\alpha)\hat{y}_{T|T-1},$$

where $0 \leq \alpha \leq 1$ is the smoothing parameter. Similarly, we can write the fitted values as

$$\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1},$$

for $t = 1, \dots, T$. (Recall that fitted values are simply one-step forecasts of the training data.)

The process has to start somewhere, so we let the first fitted value at time 1 be denoted by ℓ_0 (which we will have to estimate). Then

$$\begin{aligned}
\hat{y}_{2|1} &= \alpha y_1 + (1-\alpha)\ell_0 \\
\hat{y}_{3|2} &= \alpha y_2 + (1-\alpha)\hat{y}_{2|1} \\
\hat{y}_{4|3} &= \alpha y_3 + (1-\alpha)\hat{y}_{3|2} \\
&\vdots \\
\hat{y}_{T|T-1} &= \alpha y_{T-1} + (1-\alpha)\hat{y}_{T-1|T-2} \\
\hat{y}_{T+1|T} &= \alpha y_T + (1-\alpha)\hat{y}_{T|T-1}.
\end{aligned}$$

Substituting each equation into the following equation, we obtain

$$
\begin{aligned}
\hat{y}_{3|2} &= \alpha y_2 + (1-\alpha)\left[\alpha y_1 + (1-\alpha)\ell_0\right] \\
&= \alpha y_2 + \alpha(1-\alpha)y_1 + (1-\alpha)^2\ell_0 \\
\hat{y}_{4|3} &= \alpha y_3 + (1-\alpha)[\alpha y_2 + \alpha(1-\alpha)y_1 + (1-\alpha)^2\ell_0] \\
&= \alpha y_3 + \alpha(1-\alpha)y_2 + \alpha(1-\alpha)^2 y_1 + (1-\alpha)^3\ell_0 \\
&\vdots \\
\hat{y}_{T+1|T} &= \sum_{j=0}^{T-1} \alpha(1-\alpha)^j y_{T-j} + (1-\alpha)^T \ell_0.
\end{aligned}
$$

The last term becomes tiny for large T. So, the weighted average form leads to the same forecast Equation (7.1).

Component form

An alternative representation is the component form. For simple exponential smoothing, the only component included is the level, ℓ_t. (Other methods which are considered later in this chapter may also include a trend b_t and a seasonal component s_t.) Component form representations of exponential smoothing methods comprise a forecast equation and a smoothing equation for each of the components included in the method. The component form of simple exponential smoothing is given by:

$$
\begin{aligned}
\text{Forecast equation} \qquad & \hat{y}_{t+h|t} = \ell_t \\
\text{Smoothing equation} \qquad & \ell_t = \alpha y_t + (1-\alpha)\ell_{t-1},
\end{aligned}
$$

where ℓ_t is the level (or the smoothed value) of the series at time t. Setting $h = 1$ gives the fitted values, while setting $t = T$ gives the true forecasts beyond the training data.

The forecast equation shows that the forecast value at time $t + 1$ is the estimated level at time t. The smoothing equation for the level (usually referred to as the level equation) gives the estimated level of the series at each period t.

If we replace ℓ_t with $\hat{y}_{t+1|t}$ and ℓ_{t-1} with $\hat{y}_{t|t-1}$ in the smoothing equation, we will recover the weighted average form of simple exponential smoothing.

The component form of simple exponential smoothing is not particularly useful, but it will be the easiest form to use when we start adding other components.

Flat forecasts

Simple exponential smoothing has a "flat" forecast function:

$$\hat{y}_{T+h|T} = \hat{y}_{T+1|T} = \ell_T, \qquad h = 2, 3, \ldots.$$

That is, all forecasts take the same value, equal to the last level component. Remember that these forecasts will only be suitable if the time series has no trend or seasonal component.

Optimisation

The application of every exponential smoothing method requires the smoothing parameters and the initial values to be chosen. In particular, for simple exponential smoothing, we need to select the values of α and ℓ_0. All forecasts can be computed from the data once we know those values. For the methods that follow there is usually more than one smoothing parameter and more than one initial component to be chosen.

In some cases, the smoothing parameters may be chosen in a subjective manner — the forecaster specifies the value of the smoothing parameters based on previous experience. However, a more reliable and objective way to obtain values for the unknown parameters is to estimate them from the observed data.

In Section 5.2, we estimated the coefficients of a regression model by minimising the sum of the squared residuals (usually known as SSE or "sum of squared errors"). Similarly, the unknown parameters and the initial values for any exponential smoothing method can be estimated by minimising the SSE. The residuals are specified as $e_t = y_t - \hat{y}_{t|t-1}$ for $t = 1, \ldots, T$. Hence, we find the values of the unknown parameters and the initial values that minimise

$$\text{SSE} = \sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1})^2 = \sum_{t=1}^{T} e_t^2. \tag{7.2}$$

Unlike the regression case (where we have formulas which return the values of the regression coefficients that minimise the SSE), this involves a non-linear minimisation problem, and we need to use an optimisation tool to solve it.

Example: Oil production

In this example, simple exponential smoothing is applied to forecast oil production in Saudi Arabia.

```
oildata <- window(oil, start=1996)
# Estimate parameters
fc <- ses(oildata, h=5)
# Accuracy of one-step-ahead training errors
round(accuracy(fc),2)
#>                    ME  RMSE   MAE MPE MAPE MASE  ACF1
#> Training set 6.4 28.12 22.26 1.1 4.61 0.93 -0.03
```

This gives parameter estimates $\hat{\alpha} = 0.83$ and $\hat{\ell}_0 = 446.6$, obtained by minimising SSE over periods $t = 1, 2, \ldots, 18$, subject to the restriction that $0 \leq \alpha \leq 1$.

In Table 7.1 we demonstrate the calculation using these parameters. The second last column shows the estimated level for times $t = 0$ to $t = 18$; the last few rows of the last column show the forecasts for $h = 1, 2, 3, 4, 5$.

Table 7.1: Forecasting the total oil production in millions of tonnes for Saudi Arabia using simple exponential smoothing.

Year	Time	Observation	Level	Forecast
	t	y_t	ℓ_t	$\hat{y}_{t\|t-1}$
1995	0		446.59	
1996	1	445.36	445.57	446.59
1997	2	453.20	451.93	445.57
1998	3	454.41	454.00	451.93
1999	4	422.38	427.63	454.00
2000	5	456.04	451.32	427.63
2001	6	440.39	442.20	451.32
2002	7	425.19	428.02	442.20
2003	8	486.21	476.54	428.02
2004	9	500.43	496.46	476.54
2005	10	521.28	517.15	496.46
2006	11	508.95	510.31	517.15
2007	12	488.89	492.45	510.31
2008	13	509.87	506.98	492.45
2009	14	456.72	465.07	506.98
2010	15	473.82	472.36	465.07
2011	16	525.95	517.05	472.36
2012	17	549.83	544.39	517.05
2013	18	542.34	542.68	544.39
	h			$\hat{y}_{T+h\|T}$
2014	1			542.68
2015	2			542.68
2016	3			542.68
2017	4			542.68
2018	5			542.68

The black line in Figure 7.2 is a plot of the data, which shows a changing level over time.

```
autoplot(fc) +
  autolayer(fitted(fc), series="Fitted") +
  ylab("Oil (millions of tonnes)") + xlab("Year")
```

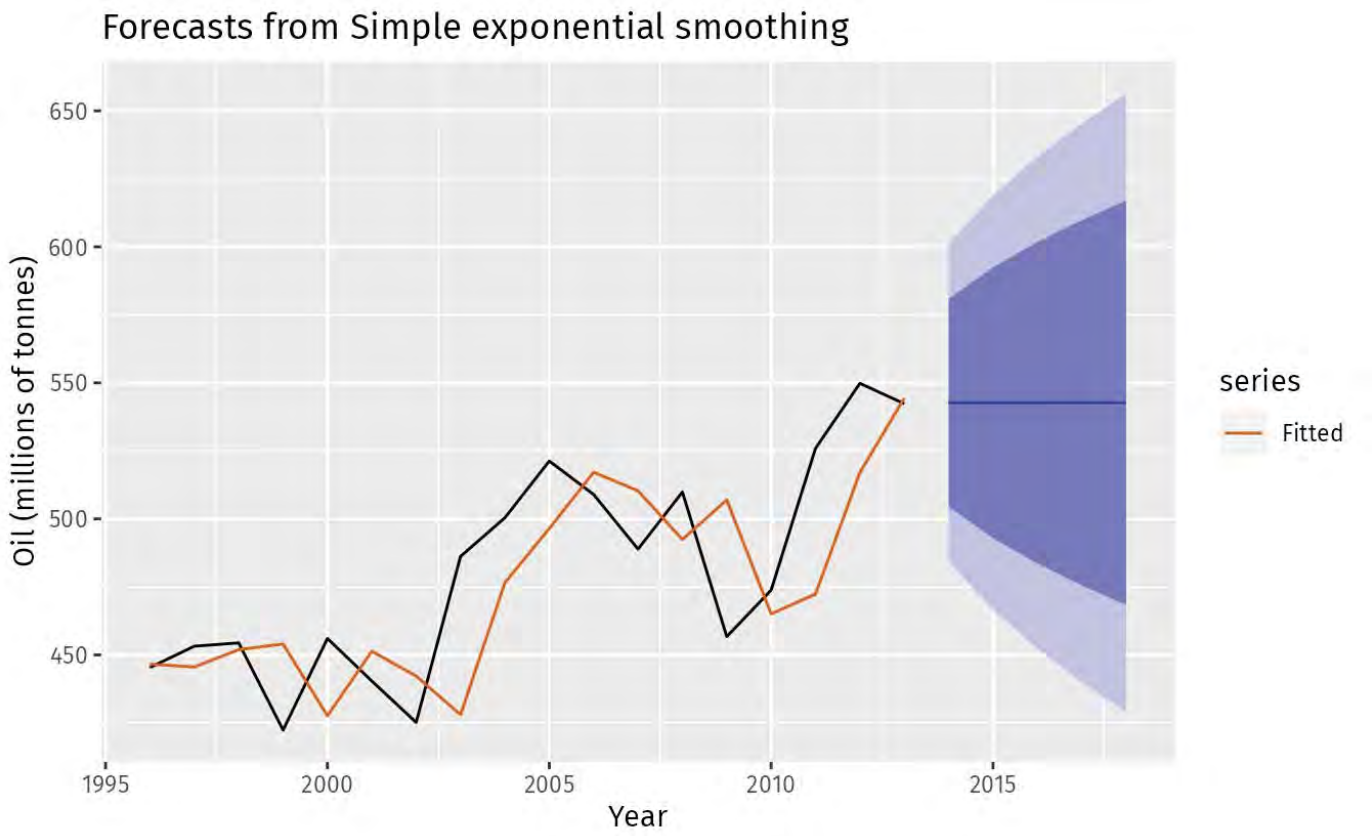


Figure 7.2: Simple exponential smoothing applied to oil production in Saudi Arabia (1996–2013).

The forecasts for the period 2014–2018 are plotted in Figure 7.2. Also plotted are one-step-ahead fitted values alongside the data over the period 1996–2013. The large value of α in this example is reflected in the large adjustment that takes place in the estimated level ℓ_t at each time. A smaller value of α would lead to smaller changes over time, and so the series of fitted values would be smoother.

The prediction intervals shown here are calculated using the methods described in Section 7.7. The prediction intervals show that there is considerable uncertainty in the future values of oil production over the five-year forecast period. So interpreting the point forecasts without accounting for the large uncertainty can be very misleading.

14. In some books it is called "single exponential smoothing".↩

7.2 Trend methods

Holt's linear trend method

Holt (1957) extended simple exponential smoothing to allow the forecasting of data with a trend. This method involves a forecast equation and two smoothing equations (one for the level and one for the trend):

$$\begin{aligned}
\text{Forecast equation} \quad & \hat{y}_{t+h|t} = \ell_t + hb_t \\
\text{Level equation} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{Trend equation} \quad & b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1},
\end{aligned}$$

where ℓ_t denotes an estimate of the level of the series at time t, b_t denotes an estimate of the trend (slope) of the series at time t, α is the smoothing parameter for the level, $0 \leq \alpha \leq 1$, and β^* is the smoothing parameter for the trend, $0 \leq \beta^* \leq 1$. (We denote this as β^* instead of β for reasons that will be explained in Section 7.5.)

As with simple exponential smoothing, the level equation here shows that ℓ_t is a weighted average of observation y_t and the one-step-ahead training forecast for time t, here given by $\ell_{t-1} + b_{t-1}$. The trend equation shows that b_t is a weighted average of the estimated trend at time t based on $\ell_t - \ell_{t-1}$ and b_{t-1}, the previous estimate of the trend.

The forecast function is no longer flat but trending. The h-step-ahead forecast is equal to the last estimated level plus h times the last estimated trend value. Hence the forecasts are a linear function of h.

Example: Air Passengers

```
air <- window(ausair, start=1990)
fc <- holt(air, h=5)
```

In Table 7.2 we demonstrate the application of Holt's method to annual passenger numbers for Australian airlines. The smoothing parameters, α and β^*, and the initial values ℓ_0 and b_0 are estimated by minimising the SSE for the one-step training errors as in Section 7.1.