APPLIED MARKETING ANALYTICS USING PYTHON

Gokhan Yildirim & Raoul Kübler

Sage

1 Oliver's Yard
55 City Road
London EC1Y 1SP

2455 Teller Road
Thousand Oaks
California 91320

Unit No 323-333, Third Floor, F-Block
International Trade Tower
Nehru Place, New Delhi – 110 019

8 Marina View Suite 43-053
Asia Square Tower 1
Singapore 018960

Editor: Amy Mitchell
Editorial Assistant: Maura Mary Joseph
Production editor: Ian Antcliff
Marketing manager: Elena Asplen
Cover design: Francis Kenney
Typeset by: C&M Digitals (P) Ltd, Chennai, India
Printed in the UK

Library of Congress Control Number: 2024941339

British Library Cataloguing in Publication data

A catalogue record for this book is available from
the British Library

ISBN: 978-1-5296-8416-2
ISBN: 978-1-5296-8417-9

CONTENTS

ACKNOWLEDGEMENTS

I warmly thank my industry partners for their input on producing practical, relevant and rigorous content for the book. I am also grateful to my students. With them in mind, I tried to be clear when illustrating the key technical concepts.

I am particularly indebted to my co-authors who stimulated me and brought new ideas to my attention.

I also wish to thank my colleagues at Imperial College London for their useful suggestions on the creation of this book. Special thanks go to Britney Wang whose help with the software applications was enormous.

Finally, I would like to express my sincere appreciation to my wife, Asli, my children, Aras and Nehir, my parents and my extended family for their constant support, encouragement and sacrifices during the creation of this book.

Gokhan Yildirim

This one is for you, Dad. Thank you for teaching me how to write. Even though you've gone, you're still helping me find the right words. Farewell, old sailor!

Raoul Kübler

Finally, we wish to thank Matthew Waters, Charlotte Hanson and Nicola Marshall at Sage for all the excellent service they provided during the creation of this book. In addition, we sincerely thank our student assistants Rhianne Gonsalves and Senih Yilmaz for their great help with creating the Jupyter files for this book.

ABOUT THE AUTHORS

Dr. Gokhan Yildirim is an associate professor of marketing at Imperial College Business School where he teaches on the full-time MBA, executive MBA and MSc Business Analytics programmes. Gokhan's research is at the intersection of marketing effectiveness, metrics and models. He quantifies how marketing actions impact offline and online consumer behaviour, and how changes in consumer attitudes in turn drive company performance. Specifically, his research concerns the short- and long-term effectiveness of digital and non-digital marketing activities, cross-channel marketing resource allocation, and consumer attitudinal metrics for guiding marketing decisions. He uses applied time-series econometrics and machine learning tools to offer managerial insights in these areas. Gokhan's academic work has appeared in leading journals of the field such as the *Journal of Marketing*, *Marketing Science*, *Journal of the Academy of Marketing Science* and the *International Journal of Research in Marketing*. He is the recipient of prestigious research awards such as an Amazon award in advertising and the ISMS-MSI Gary Lilien Practice Prize award for outstanding implementation of marketing science concepts and methods.

Dr. Raoul Kübler is an associate professor of marketing at ESSEC Business School in Cergy/Paris where he teaches on the *Financial Times* top 10 ranked Master in Management programme of ESSEC's Grande École, as well as on the school's global MBA and executive MBA. Before joining ESSEC he worked from 2018 until 2022 as a junior professor at the University of Münster, and from 2012 until 2018 an assistant professor of marketing at Özyeğin University in Istanbul, Turkey. In his research, he examines how marketers can leverage user-generated content and social media data in combination with machine learning and artificial intelligence to derive better marketing decisions. His research focus is largely guided by his close collaboration with marketing professionals. He has consulted leading international companies such as TetraPak, Rausch AG Switzerland, Dr. Wolff Group, PepsiCo Turkey, Sisecam Turkey, Garanti Bank, GfK and OD Yachting on digital marketing issues. Dr. Kübler's research is frequently published in leading international marketing and business journals such as the *Journal of Marketing*, the *Journal of the Academy of Marketing Science*, the *Journal of Retailing*, the *Journal of Interactive Marketing* and the *Journal of Business Research*. Furthermore, Raoul has been ranked since 2018 within the top 10% of most successful business scholars in Germany, Switzerland and Austria according to A+ publications, of which some have been awarded by institutions such as the Academy of Marketing Science, the Marketing Science Institute, Marketing EDGE and the European Marketing Academy.

OVERVIEW

The marketing landscape is changing at an astonishing rate as a result of new emerging technologies, granular data availability and ever-growing analytics tools. Indeed, today's marketing is no longer just an art; it requires the use of data and quantitative approaches to support strategic decisions in rapidly evolving offline and online environments. With the help of the latest artificial intelligence tools as well as other statistical and econometric models, marketers can now strengthen their decision-making with more accurate information and insights derived from the rich available data.

In combination with 'soft' skills (e.g. creativity, communication and teamwork), coding, machine learning tools, big data analytics and insight generation are now becoming the new 'must-have' skill set for marketing graduates and practitioners. Taking a very hands-on approach with real-world datasets and cases, this textbook aims to help instructors, students and practitioners explore a range of marketing phenomena using various applied analytics tools and frameworks. The textbook's practical orientation together with computer-based case studies and real-life examples will allow users to identify potential customers, boost their experience through improved targeting, turn customer engagement into sales, and derive useful and actionable managerial insights.

This textbook ties the theoretical marketing frameworks to real-world cases through computer-based applications and software code. It is particularly recommended to instructors who teach marketing analytics courses at bachelor's and postgraduate (MSc and MBA) levels and would like to adopt computer-supported case-based delivery in their teaching. Furthermore, we recommend the book to managers who are keen to extend their knowledge in the domain and gain a better understanding of how to set up and manage an analytics project, where to get data from, and how to use specific analytical models to answer contemporary marketing questions. The book provides such practitioners with easy-to-use tools and techniques that allow them to make evidence-based marketing decisions.

The structure and flow of the book reflect the traditional marketing strategy development process as mandated by the market-oriented leadership school. First the reader is introduced to the necessary processes and tools to identify and target suitable customer segments (Chapter 2). Then, the reader learns how to adapt the marketing activities to the targeted segments (Chapter 3) and evaluate the contribution of digital marketing touchpoints to consumer responses (Chapter 4) to increase customer acquisition. After the acquisition stage, marketers need to keep customers engaged to increase sales and turn one-time shoppers into loyal customers. The reader continues by learning how to benefit from user-generated data (Chapter 5) to gain more customer insights by enriching existing – commonly survey-based – marketing metrics with information obtained from online user chatter. Meanwhile, brand equity building may play an important role; to understand brand health and strategically guide brand management, marketers need to identify the relevant key performance indicators for their brand (Chapter 6). To refine information from the mined data, the reader learns how to use state-of-the-art text mining techniques (Chapter 7). As customer acquisition – and especially reacquisition – is among the most expensive and difficult tasks in marketing, managers are well advised to also monitor customer satisfaction and predict customer churn, to prevent customers from leaving the company early on (Chapter 8). Having established a strong and healthy brand, marketers need to be able to predict demand for future periods with the help of time-series models that account for marketing activities as well as market trends (Chapter 9). As an advanced market research technique, image-mining tools are introduced to turn user-generated visual data into meaningful information that can subsequently be fed into classic marketing analytics models (Chapter 10). The book closes with a discussion of data quality, data storage and data management questions as well as a profound ethical discussion of the potential impact of analytics on the various stakeholders and society at large (Chapter 11).

PRAISE FOR *APPLIED MARKETING ANALYTICS USING R*

There are good books on marketing principles, on analytical models, and on statistical software, but not on the combination of these three areas. This is where *Applied Marketing Analytics Using R* breaks new ground and offers exceptional value to the practice of marketing model building. The marketing decision areas are carefully selected, the modelling principles are well explained, and the case studies offer relevant applications of the Python software modules. I recommend this book with enthusiasm!

Dominique M. Hanssens, Distinguished Research Professor of Marketing, UCLA Anderson School of Management, USA

Kuebler and Yildirim manage to mix tried-and-true marketing models with recent advances in machine learning to offer a coherent, practical, and down-to-earth toolbox for data-driven marketers. A must-have for modern marketing managers.

Arnaud De Bruyn, Ph.D., Professor of Marketing, ESSEC Business School, Author of *Principles of Marketing Engineering and Analytics*

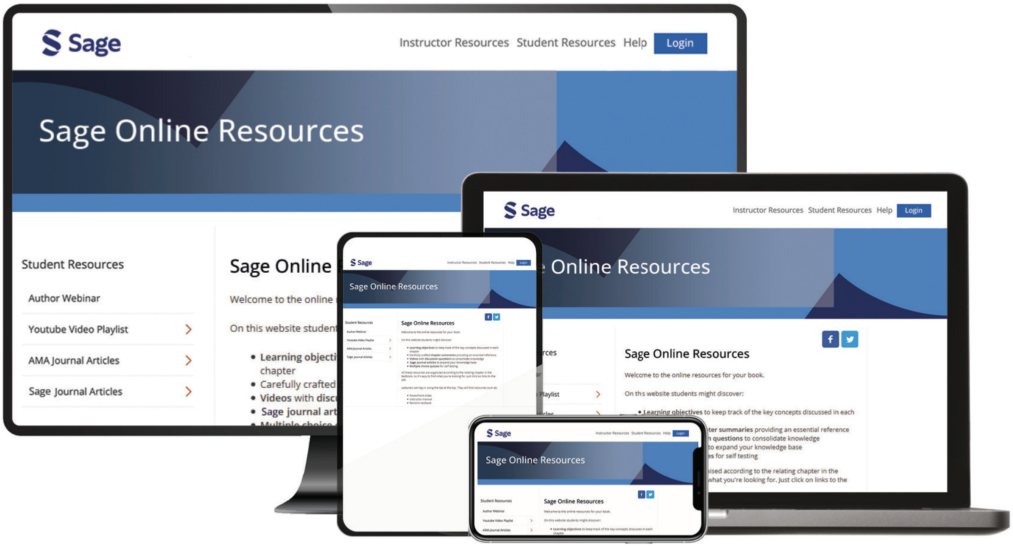
This book brings a much-needed practical perspective to scientifically sophisticated marketing analytics. The authors Gokhan and Raoul truly represent the best of both worlds, being both accomplished marketing academics and practical data scientists. They start each chapter with a case study ranging from US banks to EU skincare, and UK airlines to Turkish kitchens and Finnish game developers. I love the natural flow of the book chapters, following the market orientation structure of segmentation, targeting, positioning and marketing mix modeling. At the same time, the authors demonstrate the value of adding the latest tools in attribution, online chatter and image mining. They explain every step both in the marketing strategy process and in the software installation and implementation. As to the latter, the R exercises give you hands-on experience in the latest in marketing analytics, which helps you optimize your decisions and shine in the marketplace.

Koen Pauwels, Distinguished Professor of Marketing at Northeastern University, Boston, and co-director of its Digital, Analytics, Technology and Automation (DATA) Initiative

Applied Marketing Analytics Using R is an exceptional resource for individuals eager to achieve business success and students seeking an extensive exploration of marketing analytics and Python. Unlike many purely academic books, Yildirim from Imperial College and Kübler from ESSEC seamlessly blend a rigorous academic perspective with a practical approach to solving real-world marketing problems. This comprehensive guide takes you through the entire A to Z process of marketing analytics, covering everything from fundamental data sets and visualization techniques to advanced statistical modelling and its business implications. The inclusion of insightful case studies further enhances the practicality of the book, offering valuable applications of marketing analytics. By delving into this book, marketing researchers can elevate their skills and expertise, making it an indispensable resource for anyone serious about pursuing analytics in the field of marketing. Overall, this book makes a significant contribution to the field and is highly recommended!

Shuba Srinivasan, Norman and Adele Barron Professor in Management, Professor, Marketing, Questrom School of Business, Boston University, USA

ONLINE RESOURCES



This textbook is accompanied by online resources to aid teaching and support learning. To access these resources, visit: **https://study.sagepub.com/yildirimpython**. Please note that lecturers will require a Sage account to access the lecturer resources. An account can be created via the above link.

FOR LECTURERS

- **PowerPoints** that can be downloaded and adapted to suit individual teaching needs
- A **Teaching Guide** providing practical guidance and support and additional materials for lecturers using this textbook in their teaching
- A **Testbank** that can be used for both formative and summative student assessments

FOR STUDENTS

- **Datasets**, **software code** and **solutions** (Jupyter Notebooks, HTML files) that can be downloaded and used alongside exercises in the textbook

3

MARKETING MIX MODELLING

Chapter Contents

LEARNING OBJECTIVES

At the end of this chapter, you should be able to:

- develop a sound understanding of how marketing mix modelling works;
- build a marketing mix model using multiple regression;
- apply marketing mix modelling to a real-world dataset to measure return on marketing investments and inform optimal marketing budget allocation decisions; and
- explain the drawbacks of marketing mix modelling and how they can be addressed by other potential modelling approaches.

This chapter focuses on marketing mix modelling (MMM). MMM is a tool widely used to assess the impact of marketing mix decisions (e.g. advertising, promotions, distribution, price and salesforce) on performance metrics (e.g. web traffic, sales, revenues and profits). It allows marketing managers to guide their resource allocation strategies by measuring the contribution of their marketing efforts to business performance outcomes.

The chapter begins with a description of a case study on a marketing mix management problem faced by *FourTex*, an apparel brand. It introduces the methodological approach to address the managerial questions outlined in the case study. Then, it provides a Python application of MMM to FourTex. The software application demonstrates how to:

- assess the contribution of different advertising channels to web traffic performance;
- perform return on marketing investment analysis; and
- allocate marketing budget optimally based on estimated elasticities.

Finally, we describe the limitations of the MMM and discuss some potential alternative approaches.

Case Study

FourTex

FourTex is an apparel brand. Its products are targeted at the mass market with a focus on casual apparel for men and women. The brand's advertising is prominent and extensive, and is present on multiple channels such as television, radio, paid search and social media.

Recently, digital marketing tools have extended FourTex's reach among potential customers. The marketing director, Hannah Schmidt, was so proud of her team's achievements. She was getting ready for a meeting with the management board to explain how successful their previous marketing campaigns had been, and hoping to get an additional budget to further improve the brand's online store visits and sales leads performance metrics.

The meeting did not go as planned. In a difficult conversation, the chief executive officer of the company said: 'Mrs. Schmidt, you always ask for more money, but can rarely explain how much incremental value this money will generate'. As she left the meeting, Mrs. Schmidt felt that she was under enormous pressure to demonstrate the value of her marketing decisions.

Next day, she pulled some data from the company's database. Table 3.1 provides a brief description of the FourTex dataset.

Table 3.1 FourTex's marketing mix dataset

Marketing mix variable	Description	Channel
Google_AdWords	Cost of Google AdWords campaigns	Online
Facebook	Cost of sponsored ads delivered on Facebook	Online
TV	Cost of TV advertising	Offline
Radio	Cost of radio advertising	Offline
Traffic	Total number of visits to the website	Online

Being very keen to demonstrate the value of marketing, Mrs. Schmidt asked her analytics team to assess the impact of their Google AdWords, Facebook, TV and radio ads on website traffic performance. She did not want to include the sales performance metric in the analysis because her campaigns last year aimed to increase conversion to the website rather than sales outcomes. Therefore, she thought that the relevant key performance indicator was *website traffic*.

Finally, Mrs. Schmidt prepared a checklist for the analytics team. She sought answers to the following questions:

- Which marketing mix instrument really drives the performance outcome (i.e. website traffic)?
- What is the return on marketing investment?
- Should I keep pushing on with Google AdWords and Facebook ads? Should I stop advertising on TV and radio channels?

How can Mrs. Schmidt demonstrate the impact of her marketing mix decisions to the management board?

To address Mrs. Schmidt's questions, we build a marketing mix model that gauges the effectiveness of her marketing mix decisions and estimates the contribution of each advertising vehicle to website traffic performance. To this end, we briefly introduce the modelling approach and then proceed to the Python application.

MARKETING MIX MODELLING APPROACH

Mrs. Schmidt would like to quantify the contribution of her marketing decisions to the website traffic outcome. To help Mrs. Schmidt solve this problem, we utilize the MMM approach. In building a marketing mix model, we typically employ the multiple regression toolkit. The purpose of multiple regression analysis is to predict what the outcome variable (e.g. sales, profits, traffic) will be for a given value of input (e.g. advertising, promotion). In statistics, the outcome variable is called the *dependent variable*, whereas input variables are called *independent variables*.

Model Formulation

More technically, suppose that we have multiple independent variables, x_1, x_2,…,$x_{p,}$ in the model and want to predict the outcome, y. We observe these variables over time ($t = 1,…,T$). We can write the multiple linear regression model in the following form:

$$y_t = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \ldots + \beta_p x_{pt} + \epsilon_t, \tag{3.1}$$

where

- β_0 is the intercept,
- $\beta_1, \beta_2, \ldots, \beta_p$ are the coefficients (also known as slopes) for the independent variables $x_1, x_2 \ldots, xp$,
- $\in_t$ is the residual term.

Furthermore, we assume that the residuals are uncorrelated (independent), and follow a normal distribution with zero mean and constant variance.

Using equation (3.1), the expected value of y conditional on a set of values of X, $x_1, x_2, \ldots, x_p$, is given by

$$E(y \mid X) = \beta_0 + \beta_1 x_{1t} + \beta_2 x_{2t} + \ldots + \beta_p x_{pt}, \tag{3.2}$$

Once we estimate the intercept and slopes, the estimated regression function will be as follows:

$$\hat{y}_t = \hat{\beta}_0 + \hat{\beta}_1 x_{1t} + \hat{\beta}_2 x_{2t} + \ldots + \hat{\beta}_p x_{pt}. \tag{3.3}$$

From equation (3.3), the predicted residuals can be computed as

$$\hat{\epsilon}_t = y_t - \hat{y}_t. \tag{3.4}$$

In fact, the residuals can be thought of as prediction errors of our model estimates.

Coefficient Interpretation

How can we interpret the estimated parameters? The intercept $\hat{\beta}_0$ captures baseline sales, that is the estimated value of y (e.g. sales) when $\mathbf{X} = \mathbf{0}$ (i.e. no marketing effort). Each slope represents the estimated change in y per unit change in x_i. If the slope is positive, we say that there is a positive (negative) relationship: That is, when x_1 increases (decreases) by one unit, then y is expected to change by β_1. Note that this interpretation is for the model with variables where no data transformation has been applied. We illustrate the coefficient interpretation of a model with log-transformed data in more depth in our software application.

Model Estimation

To estimate the intercept and slope parameters, we use the ordinary least squares (OLS) estimation technique, which minimizes the sum of squared residuals of the model.

In matrix notation, the model in equation (3.1) can be expressed as

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_T \end{bmatrix} = \begin{bmatrix} 1 & \ldots & x_{1p} \\ 1 & \ldots & x_{2p} \\ \vdots & \ddots & \vdots \\ 1 & \ldots & x_{Tp} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_T \end{bmatrix} \tag{3.5}$$

We can reduce the above matrix–vector notation to a compact form as follows:

$$y = \boldsymbol{X}\beta + \epsilon \tag{3.6}$$

Then, we obtain the parameters of the model by minimizing the sum of squared residuals (*SSR*):

$$\hat{\beta} = \arg\min_{\beta} SSR(\beta), \tag{3.7}$$

where

$$SSR(\beta) \;=\; \| y - \boldsymbol{X}\beta^2 \| \;= \sum_{t=1}^{T}\left(y_t - \beta_0 - \sum_{j=1}^{p} x_{tj}\beta_j \right)^2$$

Taking the partial derivatives with respect to β yields, the OLS estimates

$$\hat{\beta} = \left(\boldsymbol{X}'\boldsymbol{X}\right)^{-1}\boldsymbol{X}'y \tag{3.8}$$

Next, we turn to the FourTex case study and implement the marketing mix model with Python.

MODEL APPLICATION WITH PYTHON

Before we roll out the analyses, we should make sure that all our files are organized and our Python environment is set up. We encourage the reader to follow the instructions below and to consult the Jupyter Notebook and HTML files that are available on the website of this book.

Preparation and Set-Up

Begin with the following steps:

- Create a folder on your computer and name the folder (e.g. *fourtex*).
- Download the data to the folder you just created.
- Open your JupyterLab and launch a new ipynb file from the '*File*' tab. Name your Jupyter Notebook (ipynb) file (e.g. *mmm_fourtex*) and save it to your folder.

To set the working directory, please run the following code. Ensure that you change the path to yours.

```
import os
os.chdir('C:/Users/gyildiri/Downloads/chapter 3')
```

We are now ready to load the dataset to Python and perform some exploratory data analysis to get a feel for the data.

Exploratory Data Analysis

To load the data into our Python environment and execute the rest of the codes, first we need to import the necessary libraries: `import numpy as np`

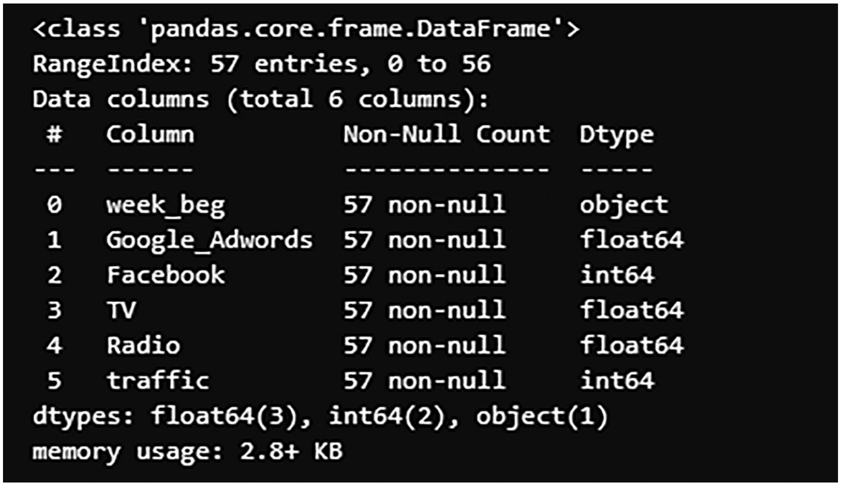
```
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
```

The following code loads the data and summarizes its main features.

```
# Load the data

data = pd.read_csv(r'C:\Users\gyildiri\Downloads\chapter 3\
data_fourtex.csv')#('/content/data_fourtex.csv')
data.info()
```

When you run the code, the output in Figure 3.1 will appear. The dataset has six variables and 57 time-series observations for each variable. In the output, we also see the variable names and data type (e.g. float, integer or object).



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 57 entries, 0 to 56
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   week_beg        57 non-null     object
 1   Google_Adwords  57 non-null     float64
 2   Facebook        57 non-null     int64
 3   TV              57 non-null     float64
 4   Radio           57 non-null     float64
 5   traffic         57 non-null     int64
dtypes: float64(3), int64(2), object(1)
memory usage: 2.8+ KB
```

Figure 3.1 FourTex dataset

Now, if you check the variables using the right-click and variable inspector options, you will see that the dataset is loaded. You can click on the new tab to observe the data.

Next, we would like to extract each of the variables to the environment tab: Python categorizes these variables as float, int and object. Python recognizes the variables we want to work with. However, it does not recognize them as time-series variables. Therefore, we should have them recognized as time-series data by using the *pd.to_datetime* function.

```
# Convert week_beg to datetime
data['week_beg'] = pd.to_datetime(data['week_beg'],
format='%d/%m/%Y')
```

```
# Creating time series for each column with a weekly frequency

data.set_index('week_beg', inplace=True)
data.index = pd.DatetimeIndex(data.index.values,
freq=data.index.inferred_freq)
```

Note that the dataset runs on a weekly basis. We set the frequency of the data to 52 weeks. However, not all the years have 52 weeks. Normally, the year has 365.25/7 = 52.18 weeks, on average. This allows for a leap year every fourth year. Therefore, some years may have 53 weeks. This is not an issue for our data as we have 57 observations spanning 2 years. None of them covers 53 weeks.

Data Plots

To get a feel for the data patterns, we need to perform a visual inspection through time-series plots. As an example, we may want to sum up online spending variables to find the total online spending, then do the same for offline spending variables.

```
# Summing up the online and offline totals
data['online_total'] = data['Google_Adwords'] +
data['Facebook']
data['offline_total'] = data['TV'] + data['Radio']
```

Next, we plot the time-series data on total online spending, total offline spending and traffic. The following codes produce the time-series plots in Figure 3.2.

```
# Plotting the traffic data
plt.figure(figsize=(10, 6))
plt.plot(data['traffic'], color="blue",
label="Online Traffic")
plt.title("Online traffic")
plt.xlabel("Date")
plt.ylabel("Online traffic")
plt.legend()
plt.show()

# Plotting the Online spending data
plt.figure(figsize=(10, 6))
plt.plot(data['online_total'], color="green",
label="Online spending")
plt.title("Online spending")
plt.xlabel("Date")
plt.ylabel("Online spending")
plt.legend()
plt.show()
```

```
# Plotting the Offline spending data
plt.figure(figsize=(10, 6))
plt.plot(data['offline_total'],
color="red", label="Offline spending")
plt.title("Offline spending")
plt.xlabel("Date")
plt.ylabel("Offline spending")
plt.legend()
plt.show()
```

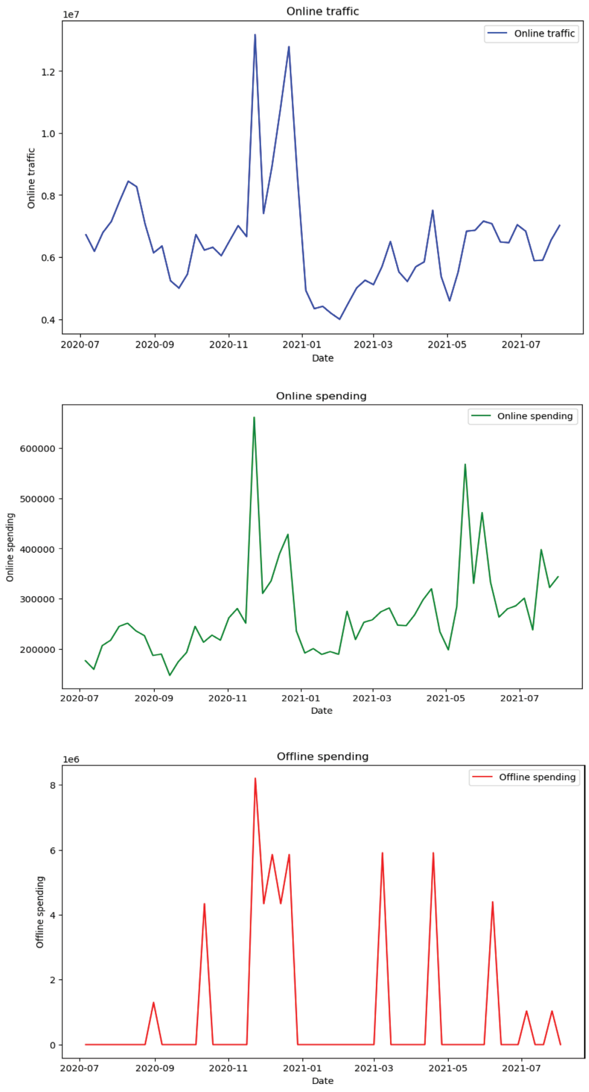


Figure 3.2 Time-series plots

Looking at the plots in Figure 3.2, what do you observe? Do increases and decreases in online traffic coincide with online and offline advertising spending? Do you see any seasonality or trend patterns?

In addition to simple time-series plot analysis, we may want to explore how much the brand spent on online and offline ads over the data period. The Python code for this is given below, and the corresponding pie chart output is shown in Figure 3.3.

```
# Calculating media spending share
sum_online = data['online_total'].sum()
sum_offline = data['offline_total'].sum()
total_spend = sum_online + sum_offline

online_share = sum_online / total_spend
offline_share = sum_offline / total_spend

# Pie-Chart for Media Spending Share
slices = [online_share, offline_share]
labels = ['Online', 'Offline']
colors = plt.cm.rainbow(np.linspace(0, 1, len(labels)))

plt.figure(figsize=(6, 6))
plt.pie(slices, labels=labels, colors=colors, autopct='%1.1f%%',
startangle=140)
plt.title("Ad spending share")
plt.show()
```

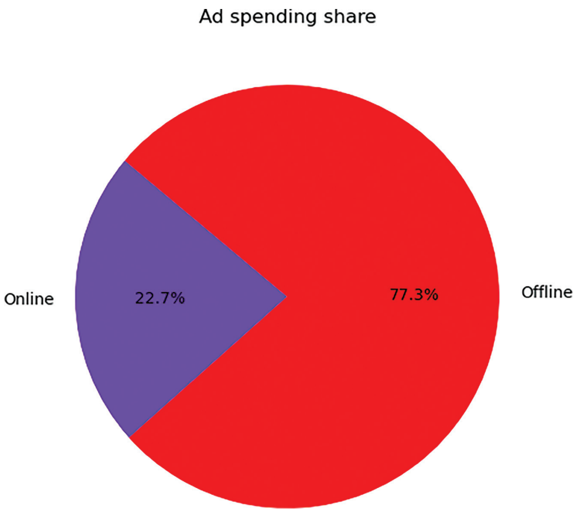


Figure 3.3 Ad spending share

We see from the pie chart that FourTex spent 77% of its budget on offline ads, while 23% of the budget went on online ads.

Marketing Mix Modelling

Having explored the main features of the data, we are ready to investigate the drivers of web traffic performance. We will develop a multiple regression model that uses the `traffic` data as dependent variable (i.e. outcome or response variable) and `Google_AdWords`, `Facebook`, `TV` and `Radio` as independent variables (i.e. predictors).

Diminishing Returns

At this point, an important decision we need to make is what functional form to use in the model. Shall we assume a linear or a nonlinear relationship? The marketing literature suggests that the relationship between advertising and performance variables mostly follows a diminishing returns pattern (Hanssens et al., 2001, 2014), as illustrated in Figure 3.4.

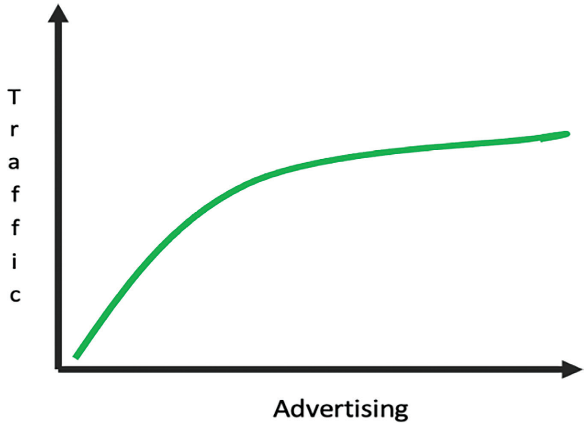


Figure 3.4 Diminishing returns

This plot tells us that initially spending more and more money on advertising is beneficial, but after a certain point, the additional value gained from an extra spending will be very small. How can we introduce this type of nonlinearity into the model? A typical approach is to use a log–log model specification.[1] The log–log regression model suggests that log transformation is performed on both sides of the equation.

[1] A semi-log model can also be used to allow for a diminishing returns pattern. The semi-log model suggests that log transformation is performed for the independent variable(s) but not for the dependent variable. Another alternative approach would be to take the square root of the independent variable(s). To decide which transformation suits best, one can try all alternative specifications and compare the model fit statistics.

Turning to our data application, with the log–log specification our marketing mix model becomes

$$\ln(\mathit{Traffic}_t) = \beta_0 + \beta_1 \ln(\mathit{Traffic}_{t\text{-}1}) + \beta_2 \ln(\mathit{Adwords}_t) + \beta_3 \ln(\mathit{Facebook}_t) + \beta_4 \ln(\mathit{TV}_t) + \beta_5 \ln(\mathit{Radio}_t) + \epsilon_t \tag{3.9}$$

where ln stands for the natural logarithm.

> Did you notice that we added the lagged traffic variable as an additional predictor to our model? Do you think including the first lagged traffic variable in the model makes sense? Why?

To estimate the log–log model in equation (3.9), we need to log-transform our variables. The following code chunk shows how to do so.

```
# Log transformation of the variables
data['ln_google_adwords'] = np.log(data['Google_Adwords'] + 1)
data['ln_facebook'] = np.log(data['Facebook'] + 1)
data['ln_tv'] = np.log(data['TV'] + 1)
data['ln_radio'] = np.log(data['Radio'] + 1)
data['ln_traffic'] = np.log(data['traffic'] + 1)
```

Did you notice that we added +1 to the original variables? The reason for doing this is that some variables include zero observations, and the logarithm of zero (ln 0) is undefined. Therefore, we should add a small number to be able to take the logarithm.

We have all the variables to be used in the model, except the first lagged traffic variable, $\ln \mathit{Traffic}_{t\text{-}1}$. The following code chunk creates this variable.

```
# Creating lagged traffic variable (1 lag)
data['L1_ln_traffic'] = data['ln_traffic'].shift(1)
```

Finally, using the following code, we estimate our model.

```
# Dropping NA values created by lag
data_lagged = data.dropna()

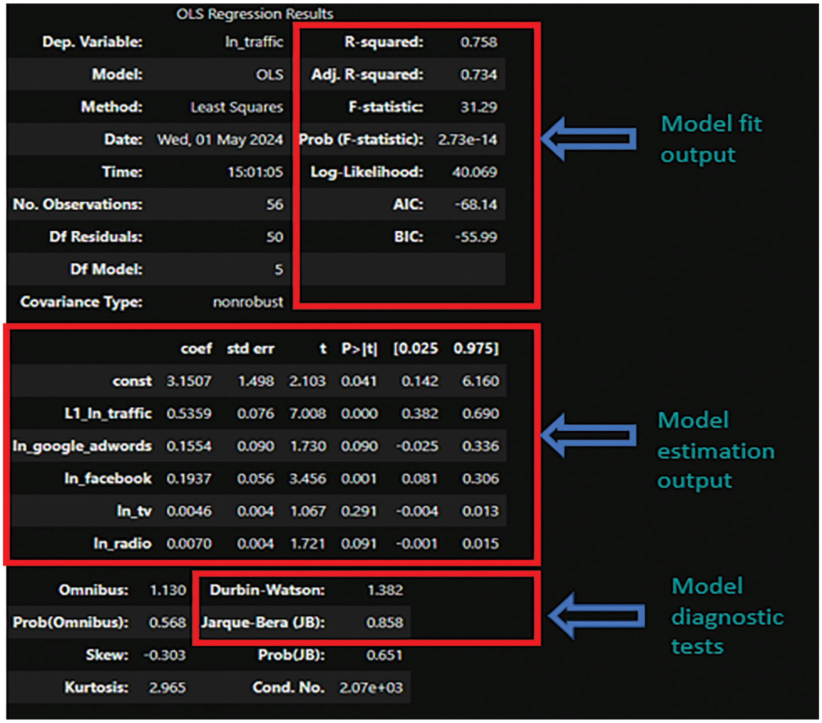
# Preparing the data for regression
X = data_lagged[['L1_ln_traffic', 'ln_google_adwords',
'ln_facebook', 'ln_tv', 'ln_radio']]
y = data_lagged['ln_traffic']

# Fit a linear regression model
regression_model = LinearRegression()
regression_model.fit(X, y)
fitted_traffic = regression_model.predict(X)
```

```
# Summary of the regression model
model_summary = sm.OLS(y, sm.add_constant(X)).fit().summary()
model_summary
```

Model Output

The Python output of our model is shown in Figure 3.5. We take a closer look at the model output and learn about its key components.



```
                         OLS Regression Results
    Dep. Variable:          ln_traffic        R-squared:     0.758
           Model:                  OLS   Adj. R-squared:     0.734
          Method:        Least Squares      F-statistic:     31.29
            Date:     Wed, 01 May 2024 Prob (F-statistic): 2.73e-14
            Time:             15:01:05   Log-Likelihood:    40.069
No. Observations:                   56              AIC:    -68.14
    Df Residuals:                   50              BIC:    -55.99
        Df Model:                    5
 Covariance Type:            nonrobust

                    coef  std err      t  P>|t|  [0.025  0.975]
            const  3.1507   1.498  2.103  0.041   0.142   6.160
     L1_ln_traffic 0.5359   0.076  7.008  0.000   0.382   0.690
ln_google_adwords  0.1554   0.090  1.730  0.090  -0.025   0.336
      ln_facebook  0.1937   0.056  3.456  0.001   0.081   0.306
            ln_tv  0.0046   0.004  1.067  0.291  -0.004   0.013
         ln_radio  0.0070   0.004  1.721  0.091  -0.001   0.015

       Omnibus:  1.130   Durbin-Watson:    1.382
Prob(Omnibus):  0.568 Jarque-Bera (JB):   0.858
          Skew: -0.303         Prob(JB):   0.651
      Kurtosis:  2.965        Cond. No. 2.07e+03
```

Figure 3.5 Marketing mix model output

Residuals: Recall that residuals represent the 'unexplained' part of the model, that is the impact of other factors that are not explicitly included in the model. The descriptive statistics (e.g. min, max) of residuals are reported at the top of the model output.

Coefficient estimates: We can write down the estimated coefficients of this marketing mix model as an equation:

$$\ln(Traffic_t) = 3.151 + 0.536\ln(Traffic_{t-1}) + 0.155\ln(Adwords_t) + 0.194\ln(Facebook_t) + 0.005\ln(TV_t) + 0.007\ln(Radio_t) \tag{3.10}$$

In Figure 3.5, the standard error of each coefficient shows at what precision level we have estimated that particular coefficient; that is, it represents the uncertainty surrounding that coefficient. The *t*-value is obtained by dividing the coefficient by its standard error. The *p*-value, `Pr(>|t|)`, is computed based on the *t*-value and helps us understand whether the coefficient is statistically significant. For example, lagged traffic is a strong indicator of the next period's traffic as it is highly significant (the *p*-value is close to zero). The effect of Facebook ads is significant at the 0.1% level while Google AdWords and radio effects are significant at the 10% level. Finally, the effect of TV ads is not statistically significant.

How to Interpret the Estimated Coefficients

We start with the interpretation of the autoregressive coefficient. The estimated effect size is 0.536. If web traffic performance gains a certain momentum today, we would expect that it will carry over into future periods with an attrition rate of 0.464 (1 – 0.536). This implies that some FourTex customers make repeat web visits because the brand or advertising has gained a place in their memories, while others stop visiting after some time because their ad or site memory decays rather quickly.

Next, we look at the advertising media effects. Since this is a log–log model output, the estimated coefficients of the advertising variables can be interpreted as elasticities: the expected percentage change in the response variable with respect to a percentage change in a predictor variable, holding other predictors of the model constant.

To understand this better, let us focus on the effect of Google AdWords. Take two values of Google AdWords at two consecutive periods: $A1$ and $A2$. Holding the other variables fixed in equation (3.10), we obtain the following:

$$lnTraffic(A2) - lnTraffic(A1) = 0.155\ (lnAdwords(A2) - lnAdwords(A1)). \tag{3.11}$$

Using the properties of logarithms, we can simplify equation (3.11) as follows:

$$\ln\left(\frac{Traffic(A2)}{Traffic(A1)}\right) = 0.155 \ln\left(\frac{Adwords(A2)}{Adwords(A1)}\right). \tag{3.12}$$

Simplifying the equation in (3.12) further, we get

$$\frac{Traffic(A2)}{Traffic(A1)} = \left(\frac{Adwords(A2)}{Adwords(A1)}\right)^{0.155} \tag{3.13}$$

This result suggests that as long as the ratio of the two Google AdWords spending levels, $Adwords(A2)\,/\,Adwords(A1)$, stays the same, the expected ratio of the response variable, $Traffic(A2)\,/\,Traffic(A1)$, stays the same. In other words, percentage increases in Google AdWords lead to constant percentage changes in traffic. For example, when we increase Google AdWords by 10%, we expect a roughly 1.5% increase in Traffic ($1.10^{0.155} = 1.015$).

What is your interpretation of the estimated coefficients of the other advertising variables, Facebook, TV and radio?

R-squared: *R*-squared, also known as coefficient of determination, measures the proportion of variation in the response variable explained by the independent variables. In our case, the model we built explains 76% of the variation in logged traffic (Multiple *R*-squared in the regression output).

Adjusted *R*-squared: It is possible to increase *R*-squared by adding more and more variables to the regression equation. However, the model's explanatory power can be increased just by chance when we add more variables to the model. To see whether a new variable will improve the model, the adjusted *R*-squared value should be checked. From the regression output, we see that the adjusted *R*-squared is 0.73. That means that our model did not suffer much from adding more variables (76% for *R*-squared compared to 73% for adjusted *R*-squared).

F-test: We use the *F*-test to assess whether a linear regression with predictor variables is favoured over a simple average value of the response variable. The computed *F*-statistic is 31.29, with a *p*-value of 2.73×10^{-14}. This is a very small number, far less than the threshold of 0.05 for a 95% level of statistical significance. Thus, we can conclude that the model we estimated is favoured over a simple mean of the traffic variable.

The Durbin–Watson (DW) statistic tests for autocorrelation in the residuals of a regression model, with values ranging from 0 to 4. A value of 2.0 suggests no autocorrelation. Values below 2 indicate positive autocorrelation, while values above 2 indicate negative autocorrelation. In our model, the DW statistic is found to be 1.38, indicating slight positive autocorrelation. In such cases, one may consider adding further lags to the lagged dependent variable and retest the autocorrelation in the residuals.

The Jarque–Bera test is a statistical test that checks for the normality of the distribution of residuals in a dataset. Specifically, it tests the null hypothesis that the data are normally distributed, examining both the skewness (asymmetry) and the kurtosis (tailedness) against what would be expected from a normal distribution. A lower value, such as .858, typically suggests that the deviation from normality is small.

Model Fit

Having estimated the parameters, we can obtain the model fit plot to see visually whether our model captures the patterns in the traffic data. Running the following code chunk gives us the model fit plot in Figure 3.6.

```
# Plotting the actual vs fitted logged traffic data
plt.figure(figsize=(10, 6))
plt.plot(data_lagged.index, data_lagged['ln_traffic'], label='Logged
Traffic Data', color='blue', lw=2)
plt.plot(data_lagged.index, fitted_traffic, label='Fitted',
color='red', linestyle='--')
plt.title('Web Traffic (Actual vs Fitted)')
plt.xlabel('Date')
plt.ylabel('Logged Traffic')
plt.legend()
plt.show()
```

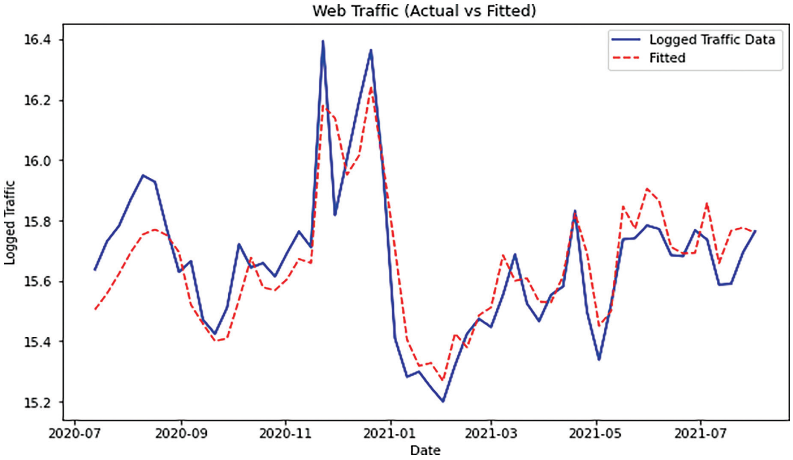


Figure 3.6 Model fit plot

Do you think the model predicted well the web traffic performance of FourTex?

Model Diagnostics

Having estimated our marketing mix model, it is usually good practice to perform model diagnostic checks on the estimated residuals (Esteban-Bravo et al., 2017; Franses, 2005). In the marketing mix model above, we assume that residuals are uncorrelated (i.e. independent), have zero mean and constant variance. If the model passes these diagnostics, we conclude that the model is not misspecified and can be used to make statistical inferences and predictions.

The model output above provides two diagnostic tests: the Durbin–Watson (DW) statistic and the Jarque–Bera test. The Durbin–Watson (DW) statistic tests for autocorrelation in the residuals of a regression model, with values ranging from 0 to 4. A value of 2.0 suggests no autocorrelation. Values below 2 indicate positive autocorrelation, while values above 2 indicate negative autocorrelation. In our model, the DW statistic is found to be 1.38, indicating slight positive autocorrelation. In such cases, one may consider adding further lags to the lagged dependent variable and retesting the autocorrelation in the residuals.

The Jarque–Bera test is a statistical test that checks for the normality of the distribution of residuals in a dataset. Specifically, it tests the null hypothesis that the data are normally distributed, examining both the skewness (asymmetry) and the kurtosis (tailedness) against what would be expected from a normal distribution. A lower value, such as 0.858, typically suggests that the deviation from normality is small.

For a detailed review on how to conduct model diagnostic tests, see Chapter 9 on demand forecasting.

Next, we turn our attention to the following questions that are central to the case study:

- What drives the web traffic performance?
- What is the traffic return on marketing investment?
- What is the optimal budget allocation?

What Drives Web Traffic Performance?

What is the contribution of marketing to web traffic performance? How much traffic was generated thanks to TV, radio, Facebook and Google AdWords? To see this, first we need to convert the elasticities to unit effects using the following formula:

$$\theta_i = \beta_i \frac{\bar{y}}{\bar{x}_i}, i = \{AdWords, Facebook, TV, Radio\}, \tag{3.14}$$

where θ_i denotes the unit effect for advertising media *i*, β_i is the estimated elasticity for media *i*, $\bar{y}$ is the baseline (average) traffic, and $\overline{x}_i$ is the baseline advertising for media *i*. Readers interested in learning how to derive this formula are referred to the appendix to this chapter.

For example, for Google AdWords, equation (3.14) can be expressed as follows:

$$\theta_{AdWords} = \beta_{AdWords} \frac{Baseline\ Traffic}{Baseline\ AdWords} \tag{3.15}$$

The following code chunk retrieves the coefficients (elasticities) from our log–log model output and then computes the unit effects using equation (3.14).

```
# Retrieve each model coefficient
coefficients = regression_model.coef_
beta_adwords = coefficients[1] # corresponding to ln_google_adwords
beta_facebook = coefficients[2]# corresponding to ln_facebook
beta_tv = coefficients[3]# corresponding to ln_tv
beta_radio = coefficients[4]# corresponding to ln_radio

# Calculate the baseline (average) traffic
average_traffic = data['traffic'].mean()

# Calculate the baseline (average) advertising
spending for each media
average_adwords = data['Google_Adwords'].mean()
average_facebook = data['Facebook'].mean()
average_tv = data['TV'].mean()
average_radio = data['Radio'].mean()

# Calculate the unit effects
theta_adwords = beta_adwords * (average_traffic / average_adwords)
theta_facebook = beta_facebook * (average_traffic / average_facebook)
```

```
theta_tv = beta_tv * (average_traffic / average_tv)
theta_radio = beta_radio * (average_traffic / average_radio)
```

Next, we compute the contribution of each advertising media to the overall traffic performance using the formula

$$Contribution_i = \theta_i \sum_{t=1}^{T} x_t^i,\ i = \{AdWords, Facebook, TV, Radio\}, \tag{3.16}$$

where θ_i indicates the estimated unit effect for advertising media i, and x_t^i is the spending of advertising media i at time t.

For example, for Google AdWords, we multiply the estimated unit effect of Google AdWords, $\theta_{AdWords}$, (6.44 from equation (3.15)) by the sum of Google AdWords spending. We compute the contribution of the other advertising media in the same way. The following code chunk performs this task.

```
# Calculate total spending for each media channel
sum_adwords = data['Google_Adwords'].sum()
sum_facebook = data['Facebook'].sum()
sum_tv = data['TV'].sum()
sum_radio = data['Radio'].sum()

# Calculate each media's contribution to traffic
adwords_contribution = theta_adwords * sum_adwords
facebook_contribution = theta_facebook * sum_facebook
tv_contribution = theta_tv * sum_tv
radio_contribution = theta_radio * sum_radio

# Print the contribution of Adwords to traffic
print("Adwords Contribution to Traffic:", adwords_contribution)
print("Facebook Contribution to Traffic:", facebook_contribution)
print("TV Contribution to Traffic:", tv_contribution)
print("Radio Contribution to Traffic:", radio_contribution)
```

We can show these media contributions graphically as well. We use barplots for this. To obtain the barplots, we need to import two libraries called matplotlib and seaborn. If you have not installed them already in your computer, you can just use:

```
! pip install 'library name'
```

```
ex: ! pip install seaborn
```

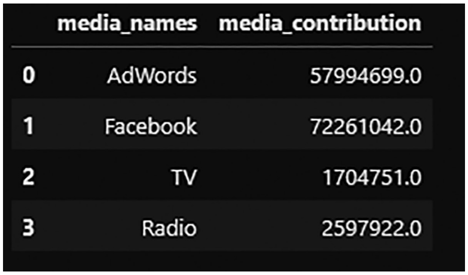
Just create a new chunk and run the updated version of this installation snippet.

We need to plug in the necessary data for the barplot. The code chunk below generates the input for the barplot.

```
import matplotlib.pyplot as plt
import seaborn as sns

# Bar plot information
media_contribution = [adwords_contribution, facebook_contribution,
tv_contribution, radio_contribution]
media_contribution = np.round(media_contribution, decimals=0)
media_names = ["AdWords", "Facebook", "TV", "Radio"]
df = pd.DataFrame({'media_names': media_names, 'media_contribution':
media_contribution})
df.head()
```

The first and second lines of the code above produce the *media contribution* column using the calculated contribution of each media. The third line of the code provides the media names together, and the fourth line of the code creates a *data frame* called *df*. Finally, the last line *df.head()* shows the data in Jupyter. The output of the code is shown in Figure 3.7.



	media_names	media_contribution
0	AdWords	57994699.0
1	Facebook	72261042.0
2	TV	1704751.0
3	Radio	2597922.0

Figure 3.7 Media contribution data for the barplot

We will use the output of the above code to create a barplot of the web traffic contribution of each media:

```
# Bar plot for media contribution to traffic
plt.figure(figsize=(8, 6))
sns.barplot(x='media_names', y='media_contribution', data=df,
palette=['Red', 'Orange', 'Blue', 'Green'])
plt.title("Contribution to Traffic")
plt.xlabel("Media")
plt.ylabel("Contribution")
plt.show()
```

Running the code above generates the barplot in Figure 3.8. This barplot suggests that most of the traffic is driven by Facebook and Google AdWords campaigns, with Facebook being the leading contributor. Radio and TV contribute very little.

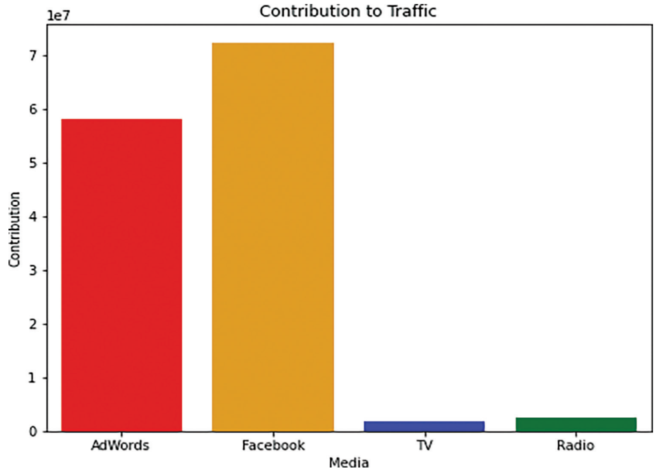
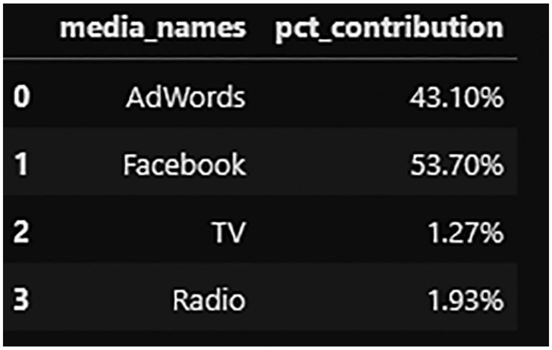


Figure 3.8 Contribution of marketing to web traffic performance

To convey the contributions more effectively, especially when the figures are substantial, we can calculate and present the contributions of each driver as a percentage. This method enables a clearer and more concise display of information, enhancing comprehensibility for the audience. Next, we compute the media contributions in percentage terms. Running the following code chunk provides the output in Figure 3.9.

~~~~
# Calculate each media's contribution as a percentage
allmedia_contribution = sum(media_contribution)
pct_contribution = [x / allmedia_contribution for x in
media_contribution]
pct_contribution = pd.Series(pct_contribution).apply(lambda x:
f"{x:.2%}")

df2 = pd.DataFrame({'media_names': media_names, 'pct_contribution':
pct_contribution})
df2
```
~~~~



	media_names	pct_contribution
0	AdWords	43.10%
1	Facebook	53.70%
2	TV	1.27%
3	Radio	1.93%

Figure 3.9 Data for the barplot with percentages

Finally, the following Python code generates the barplot with media contribution in percentage terms.

```
# Bar plot for media contribution to traffic in percentage
plt.figure(figsize=(8, 6))

# Use the numeric values for y-axis
df2['pct_contribution'] = [float(i[:-1]) for i in (df2['pct_contribution'].
tolist())]
sns.barplot(x='media_names', y=df2['pct_contribution'].astype(float),
data=df2, palette=['Orange', 'Red', 'Blue', 'Green'])

# Adding labels manually
for index, value in enumerate(df2['pct_contribution']):
    plt.text(index, value, f"{value/100:.2%}",
color='black', ha="center")

plt.title("Contribution to Traffic in %")
plt.xlabel("Media")
plt.ylabel("Contribution (%)")
plt.show()
```

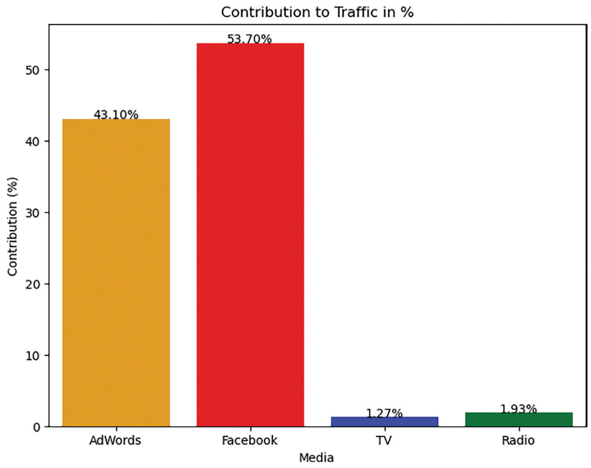


Figure 3.10 Barplot with percentages

The plot in Figure 3.10 tells us that 53.7% of the web traffic is driven by Facebook, 43.1% is driven by Google AdWords, 1.9% by radio and 1.3% by TV.

Return on Marketing Investment

Does performance increase most with a £1 reduction in TV ads or by increasing social media ads by £1? Financially oriented marketing executives are very often concerned about the return on marketing investment. That is, they would like to know how much they earn with respect to how much they spend. Usually, the return metric is sales, revenues or profits. However, it can also be something non-financial, such as customer engagement, web traffic or store traffic. For FourTex, we focus on the traffic return on marketing investment (TROMI).

The first input we need is the cost data. We obtain it using the following code chunk.

```
# Calculate the cost of each media
cost_adwords = sum_adwords
cost_facebook = sum_facebook
cost_tv = sum_tv
cost_radio = sum_radio
cost_total = cost_adwords + cost_facebook + cost_tv + cost_radio

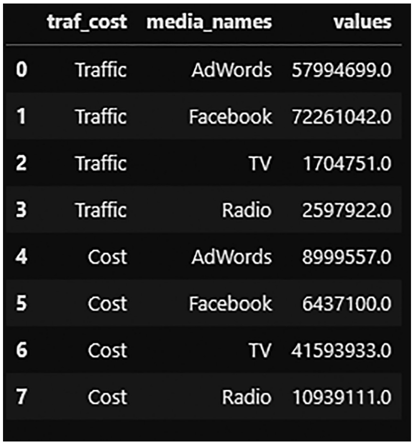
cost = [cost_adwords, cost_facebook, cost_tv, cost_radio]
cost = np.round(cost, decimals=0)
```

We run the following code to get the input for the barplot of traffic contribution and cost incurred (see Figure 3.11).

```
# Convert media_contribution and cost to lists if they aren't already
media_contribution_list = list(media_contribution)
cost_list = list(cost)

# Now, create the DataFrame
df3 = pd.DataFrame({
    'traf_cost': ['Traffic']*4 + ['Cost']*4,
    'media_names': media_names * 2,
    'values': media_contribution_list + cost_list
})

# Let's check the dataframe
df3.head(n=10)
```



```
   traf_cost  media_names       values
0    Traffic      AdWords   57994699.0
1    Traffic     Facebook   72261042.0
2    Traffic           TV    1704751.0
3    Traffic        Radio    2597922.0
4       Cost      AdWords    8999557.0
5       Cost     Facebook    6437100.0
6       Cost           TV   41593933.0
7       Cost        Radio   10939111.0
```

Figure 3.11 Input for traffic versus cost plot

Next, we run the following code chunk to see the traffic return and cost data together in a barplot (see Figure 3.12).

```
# Bar plot for Traffic vs. Media Cost
# Below there is an easier version this is just for being fancy
# Bar plot for Traffic vs. Media Cost
plt.figure(figsize=(10, 6))
sns.barplot(
    x='media_names',
    y='values',
    hue='traf_cost',
    data=df3,
    palette=['turquoise', 'salmon'], # Turquoise and salmon colors
    edgecolor='black' # Black outline for each bar
)
plt.title("Traffic vs. Media Cost")
plt.xlabel("Media")
plt.ylabel("Traffic and Cost")
plt.legend(title='traf_cost')
plt.show()

# Simpler version of the graph
# plt.figure(figsize=(10, 6))
# sns.barplot(x='media_names', y='values', hue='traf_cost',
data=df3, palette=['#1f77b4', '#ff7f0e']) # Adjusted colors
to match the first plot
# plt.title("Traffic vs. Media Cost")
# plt.xlabel("Media")
# plt.ylabel("Traffic and Cost")
# plt.legend(title='traf_cost')
# plt.show()
```

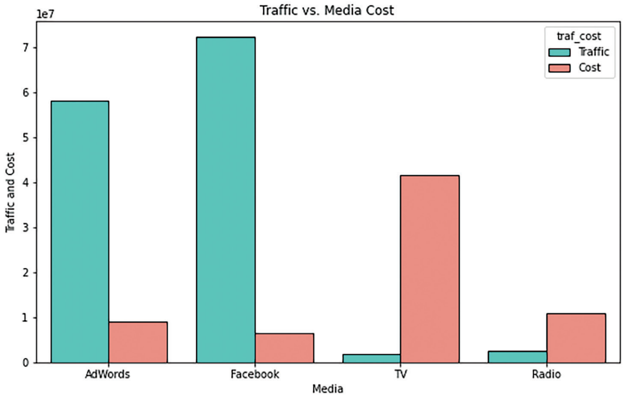


Figure 3.12 Plot of traffic against media cost

Have a look at the barplot in Figure 3.12. What is your conclusion about traffic generated and cost incurred? Which media returns the most and which the least?

Instead of showing cost and return data together, we can compute the TROMI and show the results in percentages. We just need to divide the traffic contribution of each media by the cost of each media, as in the following code chunk.

```
# Calculate the traffic return for each media
roi_adwords = adwords_contribution / cost_adwords
roi_facebook = facebook_contribution / cost_facebook
roi_tv = tv_contribution / cost_tv
roi_radio = radio_contribution / cost_radio
```

Next, we get the input for the TROMI barplot via the following code chunk.

```
# TROMI Plot input
roi = [roi_adwords, roi_facebook, roi_tv, roi_radio]
roi = np.round(roi, decimals=0)

df4 = pd.DataFrame({'media_names': media_names, 'roi': roi})
df4.head()
```

Finally, we obtain the barplot for the TROMI analysis using the following code.

```
# TROMI bar plot
plt.figure(figsize=(10, 6))
sns.barplot(x='media_names', y='roi', data=df4, palette=['Red',
'Orange', 'Blue', 'Green'])
plt.title("Traffic Return on Marketing Investment")
plt.xlabel("Media")
plt.ylabel("TROMI")
plt.show()
```

In a nutshell, the barplot in Figure 3.13 suggests that, for every £1 spent on Facebook, the expected number of web visits is 11. For every £1 spent on Google AdWords campaigns, we expect six web visits to occur.

Are you surprised that the TROMI for TV and radio is zero? Why do you think that the company invested in TV and radio channels even though their traffic return is almost null?

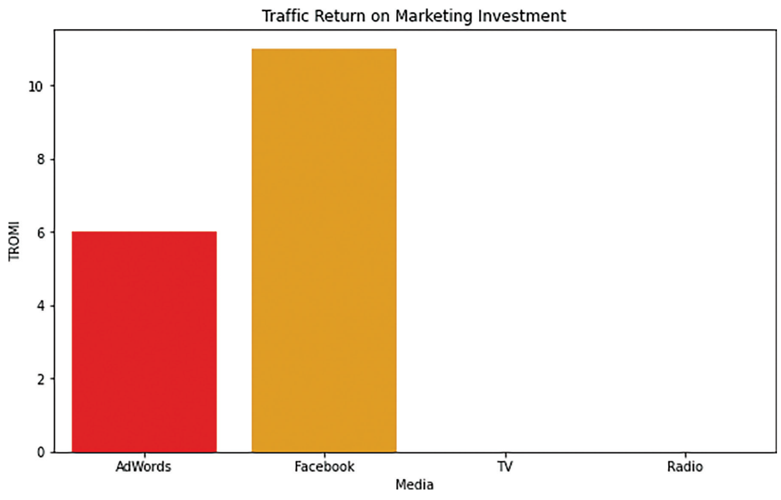


Figure 3.13 Barplot for traffic return on marketing investment

Next, we explore what Mrs. Schmidt should do with regard to her next budget allocation strategies.

Marketing Budget Allocation

In practice, marketing analysts follow two main approaches to guide their resource allocation strategies. One of them is normative decision-making based on constrained optimization models (e.g. profit maximization subject to budget constraints). The other is elasticity-based allocation. In this Python application, we allocate the marketing budget of FourTex by making use of the elasticities obtained from the log–log regression model above.

Before diving into the optimal resource allocation, let us see what the current budget allocation looks like. Running the following code yields the output in Figure 3.14.

```
# Actual Budget Spending
costshare_adwords = cost_adwords / cost_total
costshare_facebook = cost_facebook / cost_total
costshare_tv = cost_tv / cost_total
costshare_radio = cost_radio / cost_total

# Pie chart for Actual Ad Spending
slices_actual = [costshare_adwords, costshare_facebook, costshare_tv,
costshare_radio]
labels_actual = [f"{name} {share:.0%}" for name, share in zip(media_
names, slices_actual)]
colors = ['#fb0007', '#73ff08', '#21ffff', '#6b00ff']
```

```
# Adjusted color hex codes

plt.figure(figsize=(6, 6))
plt.pie(slices_actual, labels=labels_actual, colors=colors,
autopct='%1.1f%%', startangle=140)
plt.title("Actual Ad Spending")
plt.show()
```

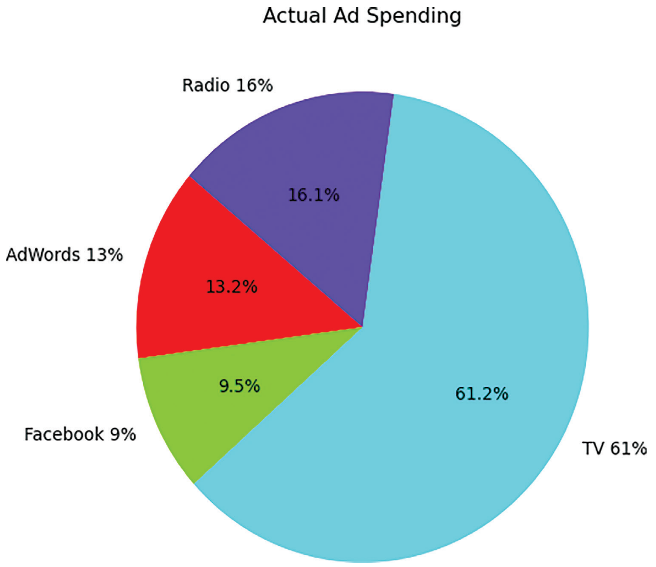


Figure 3.14 Current budget allocation of FourTex

The pie chart in Figure 3.14 tells us that 61% of the marketing budget was used for TV ads, while 16% of the budget went on radio campaigns, 13% on Google AdWords and 9% on Facebook. Given our findings on the media elasticities, how would you spend the budget? Would you spend so much money on TV ads to boost web traffic?

For the optimal allocation, we use the estimated coefficients (βs) from the log–log regression model. Recall that those coefficients are elasticities.

We calculate the optimal allocation for each media using the following formula:

$$\textit{Optimal Allocation}_i = \frac{\beta_i}{\sum_i \beta_i}, \quad i = \{AdWords, Facebook, TV, Radio\} \tag{3.17}$$

As an example, for Google AdWords, we have

$$\textit{Optimal Allocation}_{AdWords} = \frac{\beta_{AdWords}}{\beta_{AdWords} + \beta_{Facebook} + \beta_{TV} + \beta_{Radio}}. \tag{3.18}$$

Let us now do this in Python.

```
# The sum of all elasticities
beta_allmedia = beta_adwords + beta_facebook + beta_tv + beta_radio

# Optimal resource allocation
optim_adwords = beta_adwords / beta_allmedia
optim_facebook = beta_facebook / beta_allmedia
optim_tv = beta_tv / beta_allmedia
optim_radio = beta_radio / beta_allmedia
```

When we run the code chunk above, we can see the computed allocation from the *Variable Inspector* tab. Now, running the following code chunk we can get the pie chart in Figure 3.15, which shows the allocation with percentages.

```
# Pie chart for Optimal Budget Allocation
slices_optim = [optim_adwords, optim_facebook, optim_tv, optim_radio]
labels_optim = [f"{name} {share:.0%}" for name, share in zip(media_
names, slices_optim)]

plt.figure(figsize=(6, 6))
colors = ['#fb0007', '#73ff08', '#21ffff', '#6b00ff']
plt.pie(slices_optim, labels=labels_optim, colors=colors,
autopct='%1.1f%%', startangle=140)
plt.title("Optimal Budget Allocation")
plt.show()
```

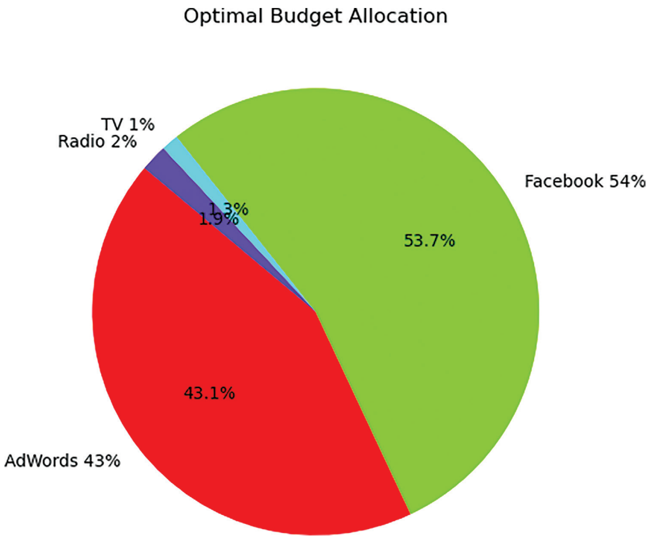


Figure 3.15 Visualization of optimal budget allocation

What is your conclusion? Is the optimal allocation different from the actual spending? How do you suggest FourTex should deploy their marketing resources to boost the web traffic performance? Would the optimal allocation be different for a different performance metric (e.g. sales)?

To summarize, through the case study on FourTex, we have explored how to:

- assess the contribution of different advertising channels to web traffic performance;
- perform return on marketing investment analysis; and
- allocate marketing budget optimally based on estimated elasticities.

Limitations

Although the marketing mix model that we have developed in this chapter is a useful tool for marketers, it has some limitations. We discuss these limitations below.

Functional form: the log–log model we developed in this chapter allows for a diminishing returns pattern. However, the relationship between website traffic performance and advertising media can be characterized by different nonlinear approximations as well (e.g. an S-shaped pattern). The decision on the exact functional form can be made by examining the scatterplots of the variables as well as the model fit statistics. Depending on the type of nonlinearity observed in the data, one can opt for different data-transformation techniques.

Synergy effects: according to Naik and Raman (2003), an advertising activity serves a dual purpose: It increases sales and enhances the effectiveness of other advertising media. For example, while consumers watch TV, they browse social mobile apps on their devices. This implies that when consumers are exposed to TV commercials, they may remember the sponsored advertisement images from social media (e.g. Facebook or X) they have recently viewed. This kind of media reinforcement suggests that one media may strengthen the impact of other media. In marketing, this effect is called *synergy*. The central idea behind synergy effects is that the combined impact of advertising media (e.g. TV ads and Facebook ads) exceeds the sum of their independent effects (i.e. $\beta_3 Facebook_t + \beta_4 TV_t$). Although our computer application did not include any synergy effects, marketers are advised to inspect some potential synergy effects in their datasets and model them by generating interaction terms (e.g. $Facebook_t \times TV_t$).

Long-term effects: the marketing literature demonstrates that the consumer response to advertising campaigns may be delayed (Hanssens, 2018). In other words, the impact of advertising on traffic performance may not be immediate. We call this a *delay or lagged effect*. Thus, an advertising campaign run today is expected to be remembered by customers over the next periods. Our marketing mix model, however, did not account for the impact over time (or dynamic impact) of advertising. In addition to modelling contemporaneous effects, one can include lagged variables in the model (e.g. $Facebook_{t-1}$, $Facebook_{t-2}$, TV_{t-1}, TV_{t-2}), and compute the long-term or cumulative effects. Autoregressive distributed-lag models can accommodate such effects (Hanssens et al., 2001; Hendry et al., 1984). Alternatively, considering the theoretical foundation of the well-established Koyck model (Franses, 2021; Koyck, 1954), one can use the estimated coefficients in equation (3.10) to compute the long-term effects. For example, the long-term effect for Facebook ads, $\varphi_{Facebook}$, is calculated as

$$\varphi_{Facebook} = \frac{\beta_3}{1-\beta_1} = \frac{0.194}{1-0.536} = 0.42. \tag{3.19}$$

If more than one lag is used for the lagged dependent variable, the denominator term should include the sum of all the autoregressive coefficients instead of β_1.

Finally, multi-equation time-series models (e.g. vector autoregressive models) can be used to infer the long-term effects of the marketing variables (Dekimpe and Hanssens, 1999). These dynamic models are flexible in capturing not only the immediate and lagged response to marketing variables but also the complex feedback loops (e.g. TV ads→Facebook ads→web traffic). The impulse-response functions derived from such dynamic system models allow managers to evaluate the wear-in, wear-out as well as short- and long-term effects of their marketing actions (Pauwels, 2018; Wang and Yildirim, 2022).

Segment-specific effects. Our marketing mix model used aggregate-level data on FourTex's marketing activities and web traffic performance. Thus, the estimated advertising effects are not segment- or individual-specific. However, some managers may be interested in exploring the responsiveness of different segments to advertising campaigns, and measuring the return on marketing investment at the segment or individual level. A practical approach to address this is to determine the distinct segments (clusters) in the dataset using clustering techniques (e.g. *k*-means algorithm, latent class segmentation), apply the marketing mix model to each segment and obtain the effects at the disaggregate level.

Intermediate metrics. While bottom-line-oriented managers typically assess marketing effectiveness using the observable metrics (e.g. web traffic, sales, profits), some marketers may use different performance metrics such as brand awareness, consideration and liking (Pauwels et al., 2013). These metrics are often considered as *intermediate* performance metrics and help managers track the state of mind of consumers (Srinivasan et al., 2016). Our model did not include such intermediate metrics as the dataset we use for FourTex did not cover such information. However, in Chapter 6 of this book, we delve into consumer attitudinal metrics and learn how they can be utilized to assess the impact of marketing.

Endogeneity. One of the assumptions that we made in our model is that there is no correlation between the advertising variables and the residuals term, that is, that advertising media are strictly exogenous. However, this assumption may sometimes fail due to the following factors:

1 *Omitted variable bias*. It is possible that advertising spending decisions are made strategically based on future web traffic expectations. For example, in the FourTex case study, Mrs. Schmidt may adjust her TV ad spending around special events in the UK, or she may decide to advertise only the successful products in her social media campaigns. When managers adapt their decisions in response to factors that are unobserved by the analyst (Papies et al., 2017), then we say that the model suffers from endogeneity and that the estimated effects are inconsistent or biased. The common approach to address this type of endogeneity in marketing mix models is the use of instrumental variables. Since this is beyond the scope of this book, we refer the interested reader to our references on this issue.
2 *Simultaneity*. When advertising media and web traffic performance variables are co-determined, an endogeneity issue arises. For example, advertising spending on Google

AdWords and Facebook may influence web traffic performance, and the brand manager may decide how much to spend on AdWords and Facebook campaigns based on the observed web traffic performance. When simultaneity-induced endogeneity occurs, the OLS estimates become biased, which in turn may result in erroneous conclusions on return on media calculations. The common approach to address this type of endogeneity is to employ multi-equation dynamic models (e.g. vector autoregressive models, vector error correction models). We refer the reader to Dekimpe and Hanssens (1999), Lütkepohl (2005) and Srinivasan (2022) for in-depth applications of these models.

CHAPTER SUMMARY

Marketing mix modelling has become one of the most frequently used analytical approaches by marketers in recent years. This approach uses aggregate-level data on marketing mix actions (e.g. advertising, promotions, distribution, price and salesforce) and performance metrics (e.g. traffic, sales, revenues, profits) and enables marketers to understand what parts of their marketing strategy generate the desired outcomes and which decisions need to be optimized.

In this chapter, we presented a case study on a marketing mix management problem faced by FourTex, an apparel brand. Then, we introduced the multiple regression toolkit to tackle the managerial questions outlined in the case study. Finally, we demonstrated the data application of the marketing mix modelling approach to FourTex, using Python Jupyter Notebook. Through the case study, we explored how to:

- assess the contribution of different advertising channels to web traffic performance;
- perform return on marketing investment analysis; and
- allocate the marketing budget optimally based on estimated elasticities.

REFERENCES

Dekimpe, M. G. and Hanssens, D. M. (1999) Sustained spending and persistent response: A new look at long-term marketing profitability. *Journal of Marketing Research*, 36, 397–412.

Esteban-Bravo, M., Vidal-Sanz, J. M. and Yildirim, G. (2017) Can retail sales volatility be curbed through marketing actions? *Marketing Science*, 36(2), 232–53.

Franses, P. H. (2005) On the use of econometric models for policy simulation in marketing. *Journal of Marketing Research*, 42, 4–14.

Franses, P. H. (2021) Marketing response and temporal aggregation. *Journal of Marketing Analytics*, 9, 111–7.

Hanssens, D. M. (2018) Return on media models. In C. Homburg, M. Klarmann and A. E. Vomberg (eds), *Handbook of Market Research*. Cham: Springer.

Hanssens, D. M., Parsons, L. J. and Schultz, R. L. (2001) *Market Response Models: Econometric and Time-Series Research*, 2nd edn. Boston: Kluwer Academic Publishers.

Hanssens, D. M., Pauwels, H. K., Srinivasan S., Vanhuele, M. and Yildirim, G. (2014) Consumer attitude metrics for guiding marketing mix decisions. *Marketing Science*, 33(4), 534–50.

Hendry, D. F., Pagan, A. R. and Sargan, J. D. (1984) Dynamic specification. In Z. Griliches and M. D. Intriligator (eds), *Handbook of Econometrics* (Vol. 2, pp. 1023–100). Amsterdam: Elsevier.

Koyck, L. M. (1954) *Distributed Lags and Investment Analysis*. Amsterdam: North-Holland.

Lütkepohl, H. (2005) *New Introduction to Multiple Time Series Analysis*. Berlin: Springer.

Naik, P. A. and Raman, K. (2003) Understanding the impact of synergy in multimedia communications. *Journal of Marketing Research*, 40(4), 375–88.

Papies, D., Ebbes, P. and van Heerde, H. J. (2017) Addressing endogeneity in marketing models. In P. S. H. Leeflang, J. E. Wieringa, T. H. A. Bijmolt and K. H. Pauwels (eds), *Advanced Methods for Modeling Markets*. Cham: Springer.

Pauwels, K. H. (2018) Modeling dynamic relations among marketing and performance metrics. *Foundations and Trends in Marketing*, 11(4), 215–301.

Pauwels, K., Erguncu, S. and Yildirim, G. (2013) Winning hearts, minds and sales: How marketing communication enters the purchase process in emerging and mature markets. International *Journal of Research in Marketing*, 30(1), 57–68.

Srinivasan, S. (2022) Modeling marketing dynamics using vector autoregressive (VAR) models. In C. Homburg, M. Klarmann and A. Vomberg (eds), *Handbook of Market Research*. Cham: Springer.

Srinivasan, S., Rutz, O. and Pauwels, K. (2016) Paths to and off purchase: Quantifying the impact of traditional marketing and online consumer activity. *Journal of the Academy of Marketing Science*, 44(4), 440–53.

Wang, W. and Yildirim, G. (2022) Applied time-series analysis in marketing. In C. Homburg, M. Klarmann and A. E. Vomberg (eds), *Handbook of Market Research*. Cham: Springer.

APPENDIX: UNIT EFFECTS AND ELASTICITIES

Here, we demonstrate how unit (marginal) effects and elasticities are interrelated. Suppose that we have the following log–log model:

$$\ln(y) = \beta_0 + \beta_1 \ln(x_1) + \varepsilon. \tag{3A.1}$$

Note that we omit the subscript t to keep the notation simple. Solving for y, we get:

$$y = e^{\beta_0 + \beta_1 \ln(x_1) + \varepsilon}. \tag{3A.2}$$

Next, we differentiate y with respect to x_1:

$$y = e^{\beta_0 + \beta_1 \ln(x_1) + \varepsilon}. \tag{3A.3}$$

Since $y = e^{\beta_0 + \beta_1 \ln(x_1) + \varepsilon}$, we can express equation (3A.3) as follows:

$$\frac{dy}{dx_1} = \frac{\beta_1}{x_1} y \tag{3A.4}$$

Rearranging the terms, we get the unit (marginal) effects

$$\frac{dy}{dx_1} = \beta_1 \frac{y}{x_1}, \tag{3A.5}$$

where y and x_1 can be replaced by average values over a range. It is straightforward to see that β_1 is elasticity:

$$\beta_1 = \frac{dy}{dx_1} \frac{x_1}{y}, \tag{3A.6}$$