Data Science for Business Leaders

Lecture Notes (currently under revision)

Prof. Dr. Stephan Huber

May 31, 2024

Table of contents

List of figures

List of tables

Preface

About Data Science for Business Leaders

This course provides an overview of the tools and methods that data science offers to business leaders. Of course, we only scratch the surface on many topics. Many topics require more in-depth analysis to fully grasp the underlying principles and their benefits. In my opinion, business leaders often don't understand every detail of the decisions they make. They can't do that and often they don't need to. However, they are good at identifying and prioritizing the key factors that can drive success. They have a sense of the essential elements that have a significant impact. Ultimately, leaders lead. They recognize, cultivate and manage the expertise of their employees and teammates. They orchestrate the knowledge around them, they lead their teams effectively and help them avoid critical missteps.

About the cover and the logo of the notes

I realize that having a visually appealing cover and logo does not directly help you become an exceptional leader. The images themselves do not teach you how to master the tools of data science or apply its methods effectively. However, I do believe that attractive visual elements appeal to many students and individuals. Therefore, as a teacher, I think it is wise to occasionally incorporate these elements into my notes. This approach may encourage you to continue reading and studying the material. By the way, it only took me a couple of minutes to create these two images using ChatGPT.

About the notes

A PDF version of these notes is available here..

Please note that while the PDF contains the same content, it has not been optimized for PDF format. Therefore, some parts may not appear as intended.

- These notes aims to support my lecture at the HS Fresenius but are incomplete and no substitute for taking actively part in class.
- I hope you find this book helpful. Any feedback is both welcome and appreciated.
- This is work in progress so please check for updates regularly.
- These notes offer a curated collection of explanations, exercises, and tips to facilitate learning R without causing unnecessary frustration. However, these notes don't aim to rival comprehensive textbooks.
- These notes are published under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License. This means it can be reused, remixed, retained, revised and redistributed as long as appropriate credit is given to the authors. If you remix, or modify the original version of this open textbook, you must redistribute all versions of this open textbook under the same license.
- I host the notes in a GitHub repo.

Structure of these notes

About the author

Contact:

Prof. Dr. Stephan Huber
Hochschule Fresenius für Wirtschaft & Medien GmbH
Im MediaPark 4c
50670 Cologne
Office: 4e OG-3
Telefon: +49 221 973199-523
Mail: stephan.huber@hs-fresenius.de
Private homepage: www.hubchev.github.io
Github: https://github.com/hubchev

Figure 1: Prof. Dr. Stephan Huber

I am a Professor of *International Economics and Data Science* at HS Fresenius, holding a Diploma in Economics from the University of Regensburg and a Doctoral Degree (summa cum laude) from the University of Trier. I completed postgraduate studies at the Interdisciplinary Graduate Center of Excellence at the Institute for Labor Law and Industrial Relations in the

European Union (IAAEU) in Trier. Prior to my current position, I worked as a research assistant to Prof. Dr. Dr. h.c. Joachim Möller at the University of Regensburg, a post-doc at the Leibniz Institute for East and Southeast European Studies (IOS) in Regensburg, and a freelancer at Charles University in Prague.

Throughout my career, I have also worked as a lecturer at various institutions, including the TU Munich, the University of Regensburg, Saarland University, and the Universities of Applied Sciences in Frankfurt and Augsburg. Additionally, I have had the opportunity to teach abroad for the University of Cordoba in Spain, the University of Perugia in Italy, and the Petra Christian University in Surabaya, Indonesia. My published work can be found in international journals such as the Canadian Journal of Economics and the Stata Journal. For more information on my work, please visit my private homepage at hubchev.github.io.

I was always fascinated by data and statistics. For example, in 1992 I could name all soccer players in Germany's first division including how many goals they scored. Later, in 2003 I joined the introductory statistics course of Daniel Rösch. I learned among others that probabilities often play a role when analyzing data. I continued my data science journey with Harry Haupt's *Introductory Econometrics* course, where I studied the infamous Jeffrey M. Wooldridge [2002] textbook. It got me hooked and so I took all the courses Rolf Tschernig offered at his chair of Econometrics, where I became a tutor at the University of Regensburg and a research assistant of Joachim Möller. Despite everything we did had to do with how to make sense out of data, we never actually used the term *data science* which is also absent in the more 850 pages long textbook by Wooldridge [2002]. The book also remains silent about *machine learning* or *artificial intelligence*. These terms became popular only after I graduated. The *Harvard Business Review* article by Davenport and Patil [2012] who claimed that data scientist is "The Sexiest Job of the 21st Century" may have boosted the popularity.

The term "data scientist" has become remarkably popular, and many people are eager to adopt this title. Although I am a professor of data science, my professional identity is more like that of an applied, empirically-oriented international economist. My hesitation to adopt the title "data scientist" also stems from the deep respect I have developed through my interactions with econometricians and statisticians. Considering their in-depth expertise, I feel like a passionate amateur.

Ultimately, I poke around in data to find something interesting. Much like my ten-year-old younger self who analyzed soccer statistics to gain a deeper understanding of the sport. The only thing that has changed since then is that I know more promising methods and can efficiently use tools for data processing and data analysis.

To students

I'm not a business leader myself. I'm a professor, and like most of my peers, I don't run a big company nor do I hold a top position in a multinational corporation. Actually, I believe that leading a company successfully full-time is not compatible with being a full-time professor, and vice versa. Despite this, in my role as a professor and study program director, I do have certain responsibilities and occasionally lead people, particularly students. This, however, doesn't let me feel like I am a business leader. Is this necessary to teach you a course entitled "Data Science for Business Leaders"? I don't think so: Firstly, if you want to become a business leader, do not pick a role model that works in academics like I do. Secondly, I believe I can provide you with valuable insights into data science to support you on the journey of becoming a business leader. And thirdly, this course is not entitled "I am a business leader who teaches you data science". Thus, this course is "for" business leaders and "for" you and not about me or my personality.

Enjoy.

1 (Extended) Syllabus

Some time ago, I wrote the syllabus of this course, see Section 1.1. that was successfully accredited by the state. Since the syllabus is quite short, I thought about an extended version to help students familiarize themselves with the buzzwords I used in the original version. As a data scientist who is supposed to teach you how to use the tools of data science in business, I wondered how I could write it without devoting too many resources to it, that is, being economical. I thought it would be a good idea and a fun exercise to let ChatGPT from OpenAI [2024] write it for me. ChatGPT uses a large language model to generate text based on a vast corpus of text data. I provided ChatGPT with specific prompts to craft the extended abstract, which you can find in Section 1.2. The final extended syllabus, actually written by ChatGPT, is available in Section 1.3.

1.1 Syllabus

Scope and Nature of Data Science

- Defining data science as an academic discipline (informatics, computer science, mathematics, statistics, econometrics, social science)
- Importance of data science in businesses

Emerging Trends in a Data-Driven Business Environment

- Evolution of computers, computing, and data processing
- Business intelligence (performance marketing, etc.)
- Artificial intelligence, machine learning, deep learning, and algorithms
- Big data
- Internet of things, cloud computing, blockchain
- Industry 4.0 and remote working

Data Science Process in Business

- Workflows and data science life cycles (OSEMN, CRISP-DM, Kanban, TDSP, …)
- Types of data science roles (data engineer, data analyst, machine learning engineer, business intelligence analyst, database administrator, data product manager, …)

Data Literacy

- Conceptual framework (knowledge and understanding of data and applications of data)
- Data collection (identify, collect, and assess data)
- Data management (organize, clean, convert, curate, and preserve data)
- Data evaluation (plan, conduct, evaluate, and assess data analyses)
- Data application (share, reflect, and evaluate results of analyses and compare them with other findings considering ethical issues and scientific standards)

Overview of Data Science Methods

- Data exploration and data mining
- Supervised and unsupervised learning
- Regression and classification
- Predictive analysis
- Causal analysis

Introduction to Data Scientific Tools

- Writing and publishing reports (Markdown, Quarto)
- Collaborating in teams using a version control system (git)
- Overview of programming languages (R, Python, SQL, …)
- Overview of no-code and low-code tools for data science (makeML, PyCaret, Rapidminer, KNIME, etc.)
- Development environments (Unix-like systems, containers, APIs, Jupyter, Rstudio, etc.)

1.2 How I let ChatGPT wrote the extended syllabus

Here are the seven prompts to ChatGPT 4.0:

1. Act as a professor of data science. Write lecture notes for management students. The first chapter of the notes should contain the following:

Scope and nature of data science

- Defining data science as an academic discipline (informatics, computer science, mathematics,statistics, econometrics, social science)
- Importance of data science in businesses

2. The chapter of the lecture notes should contain the following:

Emerging Trends in a Data-Driven Business Environment

- Evolution of computers, computing, and data processing
- Business Intelligence (Performance Marketing, etc.)
- Artificial intelligence, machine learning, deep learning, and algorithms
- Big data
- Internet of things, cloud computing, blockchain
- Industry 4.0 and remote working

3. The third chapter of the lecture notes should contain the following:

Data science process in business

- Workflows and Data science life cycles (OSEMN, CRISP-DM, Kanban, TDSP, …)
- Types of data science roles (data engineer, data analyst, machine learning engineer, business intelligence analyst, database administrator, data product manager, …)

4. Do the same for the fourth section, which contains:

Data literacy

- Conceptual framework (knowledge and understanding of data and applications of data)
- Data collection (identify, collect, and assess data)
- Data management (organize, clean, convert, curate, and preserve Kdata)

- Data evaluation (plan, conduct, evaluate, and assess data analyses)
- Data application (share, reflect, and evaluate results of analyses and compare them with other findings considering ethical issues and scientific standards)

5. And the next section:

Overview of data science methods

- Data exploration and data mining
- Supervised and unsupervised learning
- Regression and classification
- Predictive analysis
- Causal analysis

6. And the final section:

Introduction to data scientific tools

- Writing and publishing reports (Markdown, Quarto)
- Collaborating in teams using a version control system (git)
- Overview on Programming languages (R, Phyton, SQL, …)
- Overview on no-code and low-code tools for data science (makeML, PyCaret, Rapidminer, KNIME, etc.)
- Development environments (Unix-like systems, containers, APIs, Jupyter, Rstudio, etc.)

7. I am not happy with the third section. In particular the section "Workflows and Data Science Life Cycles". Can you re-write it? Consider the learning objective a bit better, that is, summarize the strengths, weaknesses and use-cases of data science processes and workflows and demonstrate the ability to justify and explain tools and methods used along the process.

I manually cut some parts of the last answer and pasted it into the respective section.

I assembled everything by copy and paste it step-by-step into Quarto and render it. That was all I did to create the notes.

Exercise 1.1. What do think about the extended syllabus and my approach
What do you think about the extended syllabus? Is it well-written? Does it include a lot of details or errors? What aspects do you think we should work on manually? How could my prompts be improved to yield better results?

1.3 The ChatGPT generated extended syllabus

The following text was generated as described in Section 1.2 using OpenAI [2024].

1.3.1 Scope and Nature of Data Science

Welcome to the introductory chapter on Data Science, designed specifically for management students. In this chapter, we will explore the multifaceted discipline of data science, understanding its definitions, scope, and the pivotal role it plays in the business world today.

1.3.1.1 Defining Data Science

Data Science is an interdisciplinary field that uses scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data. It is a confluence of several disciplines including informatics, computer science, mathematics, statistics, econometrics, and social science. This integration allows for a comprehensive approach to solving complex problems, making informed decisions, and generating actionable insights through data analysis.

1.3.1.1.1 Informatics

It is the study of computational systems and the way humans interact with these systems. In data science, informatics plays a critical role in the management and processing of data.

1.3.1.1.2 Computer Science

This foundational pillar focuses on the development of algorithms and software that are used to process, store, and analyze data.

1.3.1.1.3 Mathematics and Statistics

Mathematics provides the theoretical foundation for algorithms and models, while statistics is crucial for understanding and interpreting data. Together, they enable data scientists to perform quantitative analysis and predictive modeling.

1.3.1.1.4 Econometrics

Econometrics applies statistical methods to economic data, enabling the analysis of economic relationships. In data science, econometrics helps in forecasting economic trends and evaluating economic policies through data.

1.3.1.1.5 Social Science

The incorporation of social science principles allows data scientists to understand human behavior and societal trends, which is invaluable for businesses looking to market their products or services effectively.

1.3.1.2 Importance of Data Science in Businesses

Data Science has become a cornerstone in the business world, offering unprecedented opportunities for growth, efficiency, and innovation. Here's how:

1.3.1.2.1 Data-Driven Decision Making

Businesses can make more informed decisions by analyzing vast amounts of data. This leads to better strategic planning, operational efficiency, and competitive advantage.

1.3.1.2.2 Customer Insights

Through data analysis, companies can gain deep insights into customer behavior, preferences, and trends. This information is critical for tailoring marketing strategies, improving customer experiences, and boosting customer loyalty.

1.3.1.2.3 Risk Management

Data science enables businesses to identify and mitigate risks by analyzing historical data, detecting patterns, and predicting future outcomes. This is particularly useful in finance, insurance, and cybersecurity.

1.3.1.2.4 Innovation and Product Development

By understanding market needs and customer feedback through data analysis, businesses can innovate and develop new products and services that meet or exceed customer expectations.

1.3.1.2.5 Operational Efficiency

Data science can optimize business operations, reduce costs, and improve service delivery through the analysis of operational data, leading to streamlined processes and better resource management.

1.3.1.2.6 Competitive Edge

In a data-driven world, businesses that leverage data science effectively can stay ahead of the curve, identifying trends and opportunities before their competitors.

In conclusion, data science is an essential academic discipline that integrates various fields to analyze and interpret complex data. Its importance in the business sector cannot be overstated, as it enables organizations to make informed decisions, understand their customers, manage risks, foster innovation, improve operations, and maintain a competitive edge in the market. As we delve deeper into this course, we will explore the methodologies, tools, and applications of data science in greater detail, equipping you with the knowledge and skills to leverage data science in your future management roles.

1.3.2 Emerging Trends in a Data-Driven Business Environment

In this chapter, we explore the dynamic landscape of a data-driven business environment. The rapid evolution of technology and its integration into business processes have led to the emergence of new trends that are shaping the future of industries across the globe. Let's dive into these trends and understand their significance.

1.3.2.1 Evolution of Computers, Computing, and Data Processing

The journey from the first generation of computers to the modern era of quantum computing marks a significant evolution in technology. Initially, computers were large, expensive, and limited in their capabilities. Over the decades, advancements in semiconductor technology, the invention of the microprocessor, and the development of personal computers transformed computing into an accessible and essential tool for businesses. Today, cloud computing and edge computing have further revolutionized data processing, allowing for more efficient data storage, access, and analysis.

1.3.2.2 Business Intelligence

Business Intelligence (BI) refers to the use of data analysis in business to support decision-making processes. BI tools analyze historical and current data to provide actionable insights, helping businesses to improve their performance. Performance marketing, a subset of BI, focuses on analyzing marketing campaigns in real time to optimize marketing strategies and expenditures for better ROI.

1.3.2.3 Artificial Intelligence, Machine Learning, Deep Learning, and Algorithms

Artificial Intelligence (AI) and its subsets, Machine Learning (ML) and Deep Learning (DL), are at the forefront of technological innovation. AI involves creating systems capable of performing tasks that typically require human intelligence. ML and DL are about teaching computers to learn from data, improving their accuracy over time without being explicitly programmed. These technologies are transforming business operations, from customer service automation and predictive analytics to personalized marketing and decision-making processes.

1.3.2.4 Big Data

Big Data refers to the vast volumes of data generated every second from various sources like social media, business transactions, and IoT devices. The ability to process and analyze this data has unlocked new opportunities for businesses to gain insights into customer behavior, market trends, and operational efficiency. Big data analytics is now a crucial tool for strategic planning and competitive analysis.

1.3.2.5 Internet of Things, Cloud Computing, Blockchain

- **Internet of Things (IoT)**: IoT technology connects everyday devices to the internet, enabling them to send and receive data. This interconnectivity offers businesses real-time insights into their operations, asset tracking, and supply chain management.
- **Cloud Computing**: Cloud computing provides businesses with scalable computing resources over the internet, facilitating remote data storage, processing, and management. It supports flexibility, reduces IT costs, and enhances collaboration.
- **Blockchain**: Blockchain technology offers a secure, decentralized platform for transactions. It's particularly valuable for enhancing transparency, security, and efficiency in business operations, supply chain management, and financial services.

1.3.2.6 Industry 4.0 and Remote Working

Industry 4.0, also known as the fourth industrial revolution, integrates digital technologies into manufacturing and industry, including IoT, AI, and robotics. It represents a shift towards smart, automated production processes and data exchange. Coupled with the rise of remote working, Industry 4.0 technologies enable businesses to operate more flexibly, with teams collaborating effectively from various locations, leveraging digital tools and platforms for communication and project management.

In conclusion, the evolution of technology and its integration into business practices have brought about significant changes in the way companies operate. From the way data is processed and analyzed to the automation of manufacturing processes and the flexibility of remote working, these emerging trends are shaping the future of a data-driven business environment. As we move forward, understanding and adapting to these trends will be crucial for businesses looking to thrive in the digital age.

1.3.3 Data Science Process in Business

This chapter delves into the structured approach behind the application of data science in business settings. We will explore various data science workflows and life cycles that guide the process from raw data to actionable insights. Additionally, we will outline the different roles within a data science team and their contributions to this process.

1.3.3.1 Workflows and Data Science Life Cycles

Data science projects in business environments follow structured workflows and life cycles to ensure that the analysis is efficient, reproducible, and scalable. Several frameworks guide these processes, each with its strengths and applications.

1.3.3.1.1 OSEMN Framework

OSEMN (Obtain, Scrub, Explore, Model, iNterpret) is a streamlined approach to data science projects:

1. **Obtain**: Acquiring the data from various sources.
2. **Scrub**: Cleaning the data to ensure it is accurate and usable.
3. **Explore**: Analyzing the data to find patterns and relationships.
4. **Model**: Applying statistical models to predict or classify data.
5. **Interpret**: Drawing conclusions and making recommendations based on the model's results.

- **Strengths**: The OSEMN (Obtain, Scrub, Explore, Model, iNterpret) framework is straightforward and easy to understand, making it accessible for teams of all skill levels. It covers the essential steps of a data science project in a logical sequence.
- **Weaknesses**: Its simplicity may overlook the complexity of certain stages, such as model validation or deployment.
- **Use-Cases**: Ideal for small to medium-sized projects where the primary goal is to gain insights from data through exploration and modeling.

1.3.3.1.2 CRISP-DM

CRISP-DM stands for Cross-Industry Standard Process for Data Mining. It's a comprehensive framework that includes six phases:

1. **Business Understanding**: Define the project objectives and requirements.
2. **Data Understanding**: Collect and explore the data.
3. **Data Preparation**: Clean and preprocess the data.
4. **Modeling**: Select and apply modeling techniques.
5. **Evaluation**: Assess the model's performance.
6. **Deployment**: Implement the model in a real-world setting.

- **Strengths**: CRISP-DM (Cross-Industry Standard Process for Data Mining) is industry-agnostic and provides a detailed structure that includes understanding the business problem and deploying the solution. It encourages iterative learning and refinement.
- **Weaknesses**: Can be perceived as too rigid for projects requiring rapid development and deployment. The model doesn't explicitly address the updating or maintenance of deployed solutions.
- **Use-Cases**: Suitable for projects that require close alignment with business objectives and thorough consideration of deployment strategies.

1.3.3.1.3 Kanban

Kanban is a lean method to manage and improve work across human systems. In data science, it helps in visualizing work, limiting work-in-progress, and maximizing efficiency.

- **Strengths**: Kanban is highly flexible and promotes continuous delivery. It allows teams to adapt quickly to changes and prioritize tasks effectively.
- **Weaknesses**: Without strict stages or phases, projects might lack direction or oversight, potentially leading to inefficiencies.
- **Use-Cases**: Best for dynamic environments where priorities shift frequently and teams must remain agile to respond to business needs.

1.3.3.1.4 TDSP (Team Data Science Process)

TDSP is a standardized approach to data science projects that helps teams to improve quality and efficiency. It includes:

- **Strengths**: TDSP offers a structured approach with a strong emphasis on standardized documentation and project management methodologies, facilitating collaboration and scalability.
- **Weaknesses**: Its comprehensive nature might introduce overhead and slow down smaller projects.
- **Use-Cases**: Ideal for larger teams working on complex projects that require coordination across different roles and departments.

1.3.3.2 Types of Data Science Roles

In a business environment, a data science team might consist of various specialized roles, each contributing uniquely to the data science process.

1.3.3.2.1 Data Engineer

Focuses on the design, construction, and maintenance of the systems that data analysts and data scientists use for their work. They ensure that data flows smoothly from source to database to analytics.

1.3.3.2.2 Data Analyst

Works on processing and performing statistical analysis on existing datasets. They interpret the data to help the business make more informed decisions.

1.3.3.2.3 Machine Learning Engineer

Develops algorithms and predictive models to solve specific business problems using machine learning techniques.

1.3.3.2.4 Business Intelligence Analyst

Analyzes data to provide insights that help businesses with strategic planning. They use BI tools to convert data into understandable reports and dashboards.

1.3.3.2.5 Database Administrator

Responsible for managing, backing up, and ensuring the availability of the data stored in an organization's databases.

1.3.3.2.6 Data Product Manager

Oversees the development of data-driven products or services, ensuring that they meet the users' needs and the business objectives.

In summary, the data science process in business involves a structured approach to turning data into actionable insights. This process is supported by various frameworks and relies on the collaboration of professionals in specialized roles. Understanding these aspects of data science is crucial for anyone looking to leverage this discipline in a business context.

1.3.4 Data Literacy

Data literacy is the ability to read, understand, create, and communicate data as information. It encompasses a broad range of skills necessary for effectively working with data, from the initial stages of data collection to the final stages of analyzing and sharing findings. In this chapter, we will break down the conceptual framework of data literacy and explore its various components in detail.

1.3.4.1 Conceptual Framework

At the heart of data literacy is a deep knowledge and understanding of how data can be used to make decisions, solve problems, and communicate ideas. This conceptual framework involves:

- **Understanding the nature of data**: Recognizing different types of data (quantitative vs. qualitative) and their sources.
- **Comprehending the applications of data**: Knowing how data can be used in various contexts to derive insights and inform decisions.

1.3.4.2 Data Collection

The first step in the data lifecycle involves identifying, collecting, and assessing data:

- **Identify**: Determining the data needed to answer a question or solve a problem.
- **Collect**: Gathering data from various sources, whether they are existing datasets or new data collected through surveys, experiments, or observations.
- **Assess**: Evaluating the quality of the data, including its relevance, accuracy, and completeness.

1.3.4.3 Data Management

Once data is collected, it must be managed effectively:

- **Organize**: Arranging data in a structured format that facilitates analysis.
- **Clean**: Removing errors or inconsistencies in the data.
- **Convert**: Transforming data into a format suitable for analysis.
- **Curate**: Selecting, annotating, and maintaining valuable data for current and future use.
- **Preserve**: Ensuring that data remains accessible and usable over time.

1.3.4.4 Data Evaluation

Evaluation is critical to understanding what the data signifies:

- **Plan**: Designing a methodology for analyzing the data.
- **Conduct**: Performing the analysis using appropriate statistical methods and tools.
- **Evaluate**: Assessing the quality and reliability of the analysis.
- **Assess**: Interpreting the results in the context of the research question or business problem.

1.3.4.5 Data Application

The final step involves applying the insights gained from data analysis:

- **Share**: Communicating findings to stakeholders through reports, presentations, or visualizations.
- **Reflect**: Considering the implications of the results and how they can inform future actions.
- **Evaluate results**: Comparing findings with those from other studies or data analyses to draw broader conclusions.

- **Ethical considerations**: Ensuring that the use of data respects privacy, confidentiality, and ethical standards.
- **Scientific standards**: Adhering to rigorous standards of validity, reliability, and objectivity in data handling and analysis.

In summary, data literacy is a comprehensive set of skills that enable individuals to navigate the complex world of data from collection to application. By understanding and applying the concepts outlined in this chapter, individuals can enhance their ability to make informed decisions, solve problems, and communicate effectively using data.

1.3.5 Overview of Data Science Methods

Data science encompasses a wide array of methods and techniques for analyzing data, drawing insights, and making predictions. This chapter provides an overview of some core data science methods, including data exploration, data mining, machine learning approaches, and various types of analyses.

1.3.5.1 Data Exploration and Data Mining

1.3.5.1.1 Data Exploration

Data exploration involves analyzing data sets to find initial patterns, characteristics, and points of interest without making any prior assumptions. It typically includes summarizing the main characteristics of the data through visualizations and statistics.

1.3.5.1.2 Data Mining

Data mining is the process of discovering patterns and knowledge from large amounts of data. The data sources can include databases, data warehouses, the internet, and other sources. Data mining techniques include clustering, classification, regression, and association rule learning.

1.3.5.2 Supervised and Unsupervised Learning

1.3.5.2.1 Supervised Learning

Supervised learning is a type of machine learning where the algorithm learns from labeled training data, helping to predict outcomes for unforeseen data. It is divided into two main categories: regression and classification.

1.3.5.2.2 Unsupervised Learning

Unsupervised learning involves training on data without labeled responses. The system tries to learn the patterns and the structure from the data without any supervision. Common unsupervised learning methods include clustering and dimensionality reduction.

1.3.5.3 Regression and Classification

1.3.5.3.1 Regression

Regression methods are used to predict a continuous outcome variable based on one or more predictor variables. The goal is to find the relationship between variables and forecast an outcome. Linear regression is one of the most basic types of regression analysis.

1.3.5.3.2 Classification

Classification methods are used to predict or identify the category to which a new observation belongs. Examples include spam detection in email service providers and customer churn prediction.

1.3.5.4 Predictive Analysis

Predictive analysis uses statistical algorithms and machine learning techniques to identify the likelihood of future outcomes based on historical data. It's used in various fields, including finance, healthcare, marketing, and weather forecasting, to make more informed decisions.

1.3.5.5 Causal Analysis

Causal analysis seeks to identify and understand the cause-and-effect relationships between variables. Unlike correlation, which merely indicates that two variables move together, causation establishes that a change in one variable is responsible for a change in another.

In conclusion, these methods and techniques form the backbone of data science, enabling professionals to extract valuable insights, make predictions, and inform decision-making processes. Understanding these methods is crucial for anyone looking to delve into data science or apply its principles in their field.

1.3.6 Introduction to Data Scientific Tools

The practice of data science requires not only a solid understanding of theories and methodologies but also proficiency in a variety of tools and technologies. This chapter introduces essential tools for writing and publishing reports, collaborating in teams, programming, as well as no-code and low-code platforms, and development environments.

1.3.6.1 Writing and Publishing Reports

1.3.6.1.1 Markdown

Markdown is a lightweight markup language with plain-text formatting syntax. Its simplicity and ease of conversion to HTML and other formats make it an ideal choice for writing and publishing reports, documentation, and articles.

1.3.6.1.2 Quarto

Quarto is an open-source scientific and technical publishing system built on Pandoc. It enables users to create dynamic and reproducible reports and articles that can include executable code from various programming languages, such as R and Python.

1.3.6.2 Collaborating in Teams Using a Version Control System

1.3.6.2.1 Git

Git is a distributed version control system that enables multiple developers to work together on the same project efficiently. It tracks changes in source code during software development, supporting collaboration and fostering code integrity.

1.3.6.3 Overview of Programming Languages

1.3.6.3.1 R

R is a programming language and free software environment for statistical computing and graphics, widely used among statisticians and data miners.

1.3.6.3.2 Python

Python is a high-level, interpreted programming language known for its simplicity and versatility. It has a wide range of libraries for data analysis, machine learning, and data visualization, making it a popular choice in data science.

1.3.6.3.3 SQL

SQL (Structured Query Language) is the standard language for managing and manipulating relational databases. It allows users to query, update, and manage data.

1.3.6.4 Overview of No-Code and Low-Code Tools for Data Science

1.3.6.4.1 makeML

A no-code platform for machine learning, makeML simplifies the process of training and deploying ML models without writing extensive code.

1.3.6.4.2 PyCaret

PyCaret is a low-code machine learning library in Python that automates machine learning workflows. It enables data scientists to perform end-to-end experiments quickly and efficiently.

1.3.6.4.3 Rapidminer

Rapidminer is a data science platform that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics.

1.3.6.4.4 KNIME

KNIME is an open-source, graphical workbench for the entire analysis process: data access, data transformation, initial investigation, powerful predictive analytics, visualization, and reporting.

1.3.6.5 Development Environments

1.3.6.5.1 Unix-like Systems

Unix-like operating systems, including Linux and macOS, provide powerful tools and environments for software development and data science.

1.3.6.5.2 Containers

Containers, such as Docker, allow for the packaging of applications and their dependencies in a virtual container that can run on any Linux server, enabling easy deployment and scalability.

1.3.6.5.3 APIs

Application Programming Interfaces (APIs) enable different software applications to communicate with each other, facilitating data exchange and integration.

1.3.6.5.4 Jupyter

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text.

1.3.6.5.5 RStudio

RStudio is an integrated development environment (IDE) for R. It provides a user-friendly interface for coding, debugging, and visualizing data.

In summary, the array of tools and technologies available to data scientists is broad and varied, catering to different aspects of the data science workflow. From data manipulation and analysis to collaboration and report writing, mastering these tools is essential for effective data science practice.

2 Methods of data science

2.1 Identification

Read "How Successful Leaders Think" by Roger Martin [2007] and the chapter "Identification" of "Quantitative Methods" by Huber [2024a]. Discuss the concepts introduced by Martin [2007] critically:

- Does he provide evidence for his ideas to work?
- Is there a proof that his suggestions can yield success?
- Is there some evidence about whether his ideas are superior to alternative causes of action?
- What can we learn from the article?
- Does his argumentation fulfill highest academic standards?
- What is his identification strategy with respect to the *causes of effects* and the *effects of causes*?
- Martin [2007, p. 81] speculates:

  "At some point, integrative thinking will no longer be just a tacit skill (cultivated knowingly or not) in the heads of a select few."

- If teachers in business schools would have followed his ideas of integrative thinkers being more successful, almost 20 years later, this should be the dominant way to think as a business leader. Is that the case? And if so, can you still gain some competitive advantage by thinking that way?

3 How to use R for data science

The programming language R is one of the major tools to do data science. I wrote some lecture notes on *How to use R for data science* [Huber, 2024b].

Please read these notes.

> **Tip**
>
> We have discussed the following chapters 1, 2, 3, 5, 6, 7, 8 and some exercises of chapter 9. In particular, we have put some emphasise on chapter 6, sections 7.2-7.4, and chapter 8. Moreover, we have explicitely studied the exercises *Consumer prices over time*, *Load the Stata dataset "auto" using R*, *Convergence*,

4 Data Pitfalls

4.1 Chapter 1: Epistemic Errors

4.1.1 Learning Objectives

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

- Understand the concept of epistemic errors and their impact on data analysis.
- Identify and describe common types of epistemic errors.
- Apply strategies to avoid epistemic errors in your data work.

4.1.2 Epistemic Errors: How We Think About Data

Epistemic errors are mistakes in the way we understand and conceptualize data. These errors stem from our cognitive biases, misunderstandings, and incorrect assumptions about the nature of data and the reality it represents. Recognizing and addressing these errors is crucial for accurate data analysis and effective decision-making.

4.1.2.1 Types of Epistemic Errors

1. **The Data-Reality Gap**
    - **Definition**: The difference between the data we collect and the reality it is supposed to represent.
    - **Example**: A survey on customer satisfaction that only includes responses from a self-selected group of highly engaged customers may not accurately reflect the overall customer base.
    - **Avoidance Strategy**: Ensure that your data collection methods are representative and unbiased. Validate your data against external benchmarks or additional data sources.
2. **All Too Human Data**
    - **Definition**: Errors introduced by human biases or inaccuracies during data collection and interpretation.
    - **Example**: A researcher's personal bias influencing the design of a study or the interpretation of results.
    - **Avoidance Strategy**: Implement rigorous protocols for data collection and analysis. Use double-blind studies and peer reviews to minimize bias.
3. **Inconsistent Ratings**
    - **Definition**: Variability in data collection methods that leads to inconsistent or unreliable data.

- **Example**: Different evaluators rating the same product using different criteria or standards.
- **Avoidance Strategy**: Standardize data collection processes and provide training for all individuals involved in data collection to ensure consistency.

4. **The Black Swan Pitfall**
    - **Definition**: The failure to account for rare, high-impact events that fall outside the realm of regular expectations.
    - **Example**: Financial models that did not predict the 2008 financial crisis due to the unexpected nature of the events that led to it.
    - **Avoidance Strategy**: Consider a wide range of possible outcomes in your models and incorporate stress testing to understand the impact of rare events.
5. **Falsifiability and the God Pitfall**
    - **Definition**: The tendency to accept hypotheses that cannot be tested or disproven.
    - **Example**: Assuming that a correlation between two variables implies causation without being able to test alternative explanations.
    - **Avoidance Strategy**: Ensure that your hypotheses are testable and that you actively seek out potential falsifications. Use control groups and randomized experiments to validate causal relationships.

4.1.2.2 Avoiding Epistemic Errors

To avoid epistemic errors, it is essential to critically assess your assumptions, methodologies, and interpretations. Here are some strategies:

- **Critical Thinking**: Regularly question your assumptions and seek alternative explanations for your findings.
- **Methodological Rigor**: Use standardized and validated methods for data collection and analysis.
- **Peer Review**: Engage with peers to review and critique your work, providing a fresh perspective and identifying potential biases.
- **Continuous Learning**: Stay updated with the latest research and best practices in your field to avoid outdated or incorrect methodologies.

By understanding and addressing epistemic errors, you can significantly improve the reliability and accuracy of your data analyses, leading to better decision-making and more trustworthy insights.

4.2 Chapter 7: Design Dangers

4.2.1 Summary of Pitfalls

In "Avoiding Data Pitfalls," Chapter 7 addresses design dangers, which refer to mistakes made in the aesthetic and functional design of data presentations. These pitfalls can significantly impact the effectiveness and clarity of data visualizations. Understanding these dangers helps in creating visualizations that are both accurate and easy to understand.

- **Overcomplicated Designs**

  - **Description**: Creating visualizations that are too complex with excessive data and multiple visual elements.
  - **Impact**: Leads to confusion and misinterpretation.
  - **Solution**: Strive for simplicity and clarity in your designs to effectively communicate key insights.

- **Misleading Visuals**
  - **Description**: Using visual elements that distort the data, such as inappropriate scaling of axes, 3D effects, or incorrect chart types.
  - **Impact**: Causes misunderstandings and incorrect conclusions.
  - **Solution**: Ensure that visual elements accurately represent the data and avoid using effects that can mislead the viewer.

- **Inconsistent Design Elements**
  - **Description**: Varying color schemes, fonts, and other design elements within a single visualization or across related visualizations.
  - **Impact**: Distracts and confuses the viewer, reducing the coherence of the presentation.
  - **Solution**: Maintain consistency in visual design to create a cohesive and understandable narrative.

- **Ignoring Audience Needs**
  - **Description**: Failing to consider the level of expertise and specific needs of the audience when designing visualizations.
  - **Impact**: Results in visualizations that are inaccessible or not meaningful to the intended audience.
  - **Solution**: Tailor visualizations to the audience's level of understanding and their specific requirements.

- **Lack of Focus**
  - **Description**: Including too much information without highlighting the most important aspects.
  - **Impact**: Overwhelms the viewer and obscures the main message.
  - **Solution**: Focus on the key insights and avoid unnecessary details that do not contribute to the main message.

By recognizing and addressing these design dangers, you can improve the clarity, accuracy, and effectiveness of your data visualizations, making them more valuable for decision-making and communication.

4.3 Chapter 1: Epistemic Errors

4.3.1 Learning Objectives

By the end of this chapter, you should be able to understand the concept of epistemic errors and their impact on data analysis, identify and describe common types of epistemic errors, and apply strategies to avoid epistemic errors in your data work.

4.3.2 Epistemic Errors: How We Think About Data

Epistemic errors occur when there are mistakes in our understanding and conceptualization of data. These errors arise from cognitive biases, misunderstandings, and incorrect assumptions about the nature of data and the reality it represents. Recognizing and addressing these errors is crucial for accurate data analysis and effective decision-making.

One significant type of epistemic error is the data-reality gap, which refers to the difference between the data we collect and the reality it is supposed to represent. For instance, a survey on customer satisfaction that only includes responses from a self-selected group of highly engaged customers may not accurately reflect the overall customer base. To avoid this pitfall, it is essential to ensure that your data collection methods are representative and unbiased, and to validate your data against external benchmarks or additional data sources.

Another common epistemic error involves the influence of human biases during data collection and interpretation. Known as the all too human data error, this occurs when personal biases or inaccuracies affect the data. An example would be a researcher's personal bias influencing the design of a study or the interpretation of its results. To mitigate this, implement rigorous protocols for data collection and analysis, and consider using double-blind studies and peer reviews to minimize bias.

Inconsistent ratings can also lead to epistemic errors. This happens when there is variability in data collection methods, resulting in inconsistent or unreliable data. For example, different evaluators might rate the same product using different criteria or standards. To avoid this issue, standardize data collection processes and provide training for all individuals involved in data collection to ensure consistency.

The black swan pitfall refers to the failure to account for rare, high-impact events that fall outside regular expectations. Financial models that did not predict the 2008 financial crisis due to the unexpected nature of the events that led to it are an example of this error. To prevent such pitfalls, consider a wide range of possible outcomes in your models and incorporate stress testing to understand the impact of rare events.

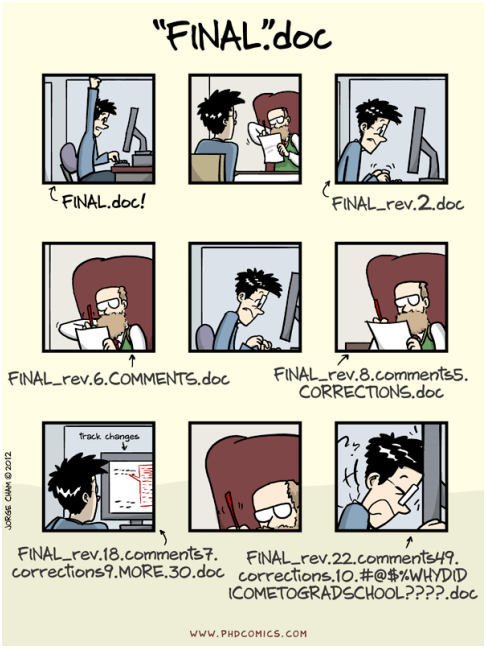
Falsifiability and the God pitfall involve the tendency to accept hypotheses that cannot be tested or disproven. This error might occur when assuming that a correlation between two variables implies causation without the ability to test alternative explanations. To avoid this, ensure that your hypotheses are testable and that you actively seek out potential falsifications. Use control groups and randomized experiments to validate causal relationships.

To avoid epistemic errors, critically assess your assumptions, methodologies, and interpretations. Engage in critical thinking by regularly questioning your assumptions and seeking alternative explanations for your findings. Employ methodological rigor by using standardized and validated methods for data collection and analysis. Engage with peers to review and critique your work, providing a fresh perspective and identifying potential biases. Finally, stay updated with the latest research and best practices in your field to avoid outdated or incorrect methodologies.

Understanding and addressing epistemic errors can significantly improve the reliability and accuracy of your data analyses, leading to better decision-making and more trustworthy insights.

5 Collaborating with Git and GitHub

Figure 5.1: The FINAL.doc problem



Source: phdcomics.com

5.1 Introduction

Git is open-source software for version control. It allows developers to track and manage changes to their codebase and files. Users can access a comprehensive history of their project and revert to previous versions of their data if necessary. It helps to overcome the *FINAL.doc problem* depicted in Figure 5.1.

Figure 5.2: GitHub is big

100+ million	4+ million	420+ million	90%
Developers	Organizations	Repositories	Fortune 100

Source: https://github.com/about as of April 2024

GitHub is an incredibly popular (see statistics in Figure 5.2) online platform that implements Git's capabilities by providing a web interface for collaboration.

While you can use Git and GitHub independently, most developers integrate it with GitHub for enhanced project management and collaboration. This combination helps maintain local and remote copies of a project, facilitating teamwork and data backup as GitHub is sort of a

backup as data loss at your local machine do not matter if you have a remote version saved on GitHub.

Git and GitHub support simultaneous multi-user access, unlike systems that are optimized for single-user like Dropbox.

5.2 Install Git

To install the version control system Git, follow the instructions here.

Figure 5.3: Memorizing six git commands

Source: DEV Community on GitHub

Familiarize yourself with Git by using the resources available here. Specifically, work through the resources listed in the box below. Although Git may appear complex, it is generally not too challenging for most users. Many people use Git primarily to track their work and to host and share files conveniently with just a handful of commands. While Git is a robust system with many capabilities, you don't need to memorize all the commands (see Figure 5.3). In fact, you typically use only a few basic ones as shown in Table 5.1.

In the upcoming sections, I will demonstrate some use cases both in the terminal Section 5.3 and within RStudio Section 5.4. In Section 5.5, I show how to contribute to a repository using Git and GitHub.

Learning resources

Plenty books and tutorial exist that introduce Git and GitHub. I'd like to highlight the following sources:

- The book comprehensive book *Happy Git and GitHub for the useR* by Bryan
- The much shorter book [*Version Control with Git and GitHub*] by Halbritter and Telford
- The online tutorial *How to Use Git/GitHub with R* of David Keyes who explains

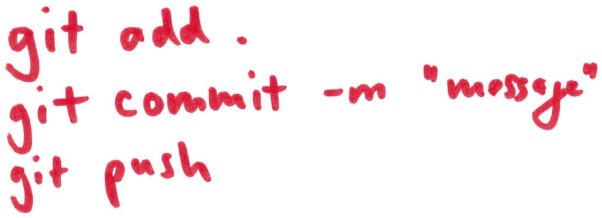
in short videos how to setup Git and GitHub in RStudio using among others the `usethis` package.

Table 5.1: Most important git commands

Git Command	Description
`git init`	Initialize a new Git repository in the current directory.
`git clone <url>`	Clone a repository from a remote URL to your local machine.
`git add <file>`	Add a specific file to the staging area in preparation for committing.
`git add .`	Add all changed files in the current directory to the staging area.
`git commit -m "message"`	Commit the staged changes to the repository with a descriptive message.
`git status`	Display the status of the working directory and staging area.
`git push <remote> <branch>`	Push committed changes in your local branch to the remote repository.
`git pull <remote> <branch>`	Pull changes from the remote repository into your current branch and merge them.
`git branch <name>`	Create a new branch with the specified name.
`git checkout <branch>`	Switch to another branch and update the working directory.
`git merge <branch>`	Merge a specified branch into the current branch.

5.3 Using Git from the terminal

Figure 5.4: Three git commands you really need



This tutorial will guide you through the basic Git operations using the Bash command line, commonly referred to as the terminal. Essentially, it focuses on the three Git commands illustrated in Figure 5.4.

5.3.1 Configuring Git

Before you start using Git, you need to configure your Git environment. Set your username and email address with these commands:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

5.3.2 Initializing a Repository

To create a new Git repository, use the git init command in the directory you want to version control:

```
cd /path/to/a/directory
mkdir my_project
cd my_project
git init
```

In case you are not familiar with using the terminal please consider Table 5.2 where I introduce the most basic commands that we use. For example, with `cd` you can change your directory and with `mkdir` you create a new directory. If you are not familiar with the file system of your computer please read the section *Navigating the file system* of Huber [2024b]. With `git init` you initialize the directory to be a git repository. This will create a hidden folder "`.git`" in which Git keeps track of all your changes.

Most common bash commands

Table 5.2: Most common bash commands

Bash Command (macOS/Linux)	Windows Command Prompt Equivalent	Description
`pwd`	`cd`	Prints the current directory's path.
`ls`	`dir`	Lists all files and directories in the current directory.
`cd`	`cd`	Changes the directory.
`mkdir`	`mkdir`	Creates a new directory.
`rmdir`	`rmdir`	Removes an empty directory.
`touch`	`copy nul`	Creates a new empty file or updates an existing file's timestamp.
`rm`	`del` or `erase`	Removes files. `rmdir /s` is used for directories.
`cp`	`copy`	Copies files or directories.
`mv`	`move`	Moves or renames files or directories.
`echo`	`echo`	Displays a line of text/string.
`cat`	`type`	Concatenates and displays the content of files.
`grep`	`find` or `findstr`	Searches for patterns in files.

5.3.3 Staging Changes

To track changes in your repository, you need to stage them using the git add command. To stage a single file:

```
git add filename.txt
```

To stage all changes in the directory:

```
git add .
```

5.3.4 Committing Changes

After staging, you can commit it to the repository. A commit records changes to the repository and must include a message describing what changed:

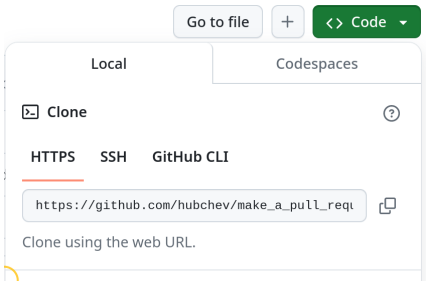
```
git commit -m "A message"
```

5.3.5 Pushing Changes

To share your commits with others or store them in a remote repository (GitHub), use `git push`. A prerequisite here is that you need to be connected to a remote repo. Therefore, you must add a remote repository by copying the URL of the GitHub repo as shown in Figure 5.5. Then you can add the remote repository and push it to the repo with these lines of code:

```
git remote add origin https://github.com/username/repository.git
git push -u origin main
```

Figure 5.5: Copy the https URL of your repo



5.3.6 Undo changes

With `git reset` and `git revert` you can go back in time and undo specific changes, respectively. For example, with

```
git log
git reset --hard <commit_id_hash>
```

you can view the commit history and find the hash identifier of the commit to move the HEAD pointer to that commit. This effectively removes all commits after commit you choose from the current branch's history. Be cautious when using git reset –hard as it discards all changes made after the specified commit. Make sure you have backups or are certain you want to discard these changes before proceeding.

With

```
git revert <commit_id_hash>
```

you revert the changes introduced by that commit only. It will create a new commit that undoes the changes made in commit chosen while keeping the other commits that may have followed the chosen commit intact. It's a safer approach compared to `git reset --hard`, as it preserves the commit history and allows you to selectively undo changes without affecting the rest of the commits.

5.4 Using Git from RStudio

Integrating Git with RStudio enhances your project management by utilizing version control directly within the IDE. Here's how you can set up and use Git in RStudio using R code.

5.4.1 Set up Git in RStudio

First, ensure the `usethis` package is installed and loaded:

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(usethis)
```

Configure your Git settings in RStudio:

```
use_git_config(user.name = "Your Name", user.email = "Your@email.com")
```

You can change the configuration of your user name and email using the `edit_git_config()` function.

Start a new project in RStudio, which will also initialize a Git repository:

```
create_project("~/Music/")
use_git()
```

After restarting RStudio, you will notice a Git tab in the top right panel, indicating that Git is now active for your project.

5.4.2 Connecting RStudio Projects with GitHub repositories

To connect your RStudio project with GitHub, you need a Personal Access Token (PAT) on GitHub. If you haven't one already, you can use the `create_github_token()` function from `usethis` package, and store the PAT securely with `gitcreds_set` from the `gitcreds` package:

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(usethis gitcreds)
create_github_token()
gitcreds::gitcreds_set()
```

Now, the procedure depends on whether the project has bin initialized on your local machine and you want to create a repo on GitHub, or the repo already exists on GitHub and you want to connect that remote repo with your local PC. Both ways are described below.

5.4.2.1 Project exists on RStudio first

After initializing Git in your project, use the `use_github()` function from `usethis` to create a new GitHub repository and connect it directly:

```
use_github()
```

This creates a repo on your GitHub account.

5.4.2.2 Project exists on GitHub first

Alternatively, suppose you have created a repository on GitHub first, then start a new project in RStudio using the version control option, specifying your new repository's URL. Just click `File > New Project > Version Control` and then link the GitHub repo by putting the URL into the respective box of the menu. See Figure 5.5 how to get the URL of a repo.

5.5 Make a contribution using Git and GitHub

This is a guide for beginners on how to make a contribution using Git and GitHub. If you are looking to make your first contribution, follow the steps below.

> **Watch**
>
> this video where I do all the following steps in real time. It takes about 15 minutes.

1. Create an account on GitHub

It is for free and should just take some minutes.

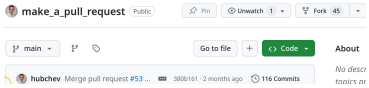
2. Install Git

Here is a tutorial on how to set up Git.

3. Fork this repository

Click on the fork button (see Figure 5.6) on the top of this page: https://github.com/hubchev/make_a_pull_request. This will create a copy of this repository in your account.

Figure 5.6: Fork the repo



4. Clone the forked repository

Go to your GitHub account, open the forked repository, click on the code button and then click the *copy to clipboard* icon, see Figure 5.5.

Then, open a terminal and run the following git command:

```
git clone "url you just copied"
```

where "url you just copied" (without the quotation marks) is the url to this repository (your fork of this project). See the previous steps to obtain the url.

For example:

```
git clone https://github.com/this-is-you/make_a_pull_request.git
```

where `this-is-you` is your GitHub username. Here you're copying the contents of the first-contributions repository on GitHub to your computer.

5. Create a branch

Change to the repository directory on your computer (if you are not already there):

```
cd make_a_pull_request
```

Now create a branch using the `git switch` command:

```
git switch -c your-new-branch-name
```

For example:

```
git switch -c add-Stephan-Huber
```

6. Make changes.

Now open the`I_am_a_data_scientist.md` file in a text editor. (You find this file in the repository.) Add your name, your GitHub account and the project you are working on. You can put it anywhere in between. Now, save the file.

If you go to the project directory and execute the command `git status`, you'll see there are changes.

7. Add changes (staging). Add those changes to the branch you just created using the `git add` command:

```
git add .
```

8. Commit changes. Now commit those changes using the `git commit` command:

```
git commit -m "Add your-name to the list"
```

replacing `your-name` with your name.

9. Use Git Bash. Open Git Bash and set your email and your nickname on GitHub:

```
git config --global user.name "FIRST_NAME LAST_NAME"
git config --global user.email "MY_NAME@example.com"
```

10. Push changes to GitHub.

Push your changes using the command `git push`:

```
git push -u origin your-new-branch-name
```

replacing `your-new-branch-name` with the name of the branch you created earlier.

If you get any errors while pushing that refers to authentication failed something, go to GitHub's tutorial on generating and configuring an SSH key to your account. Alternatively, you can watch this YouTube tutorial

11. Submit your changes for review on GitHub.

If you go to your repository on GitHub, you'll see a `Compare & pull request` button. Click on that button.

Now submit the pull request.

Soon I'll be merging all your changes into the main branch of this project. You will get a notification email once the changes have been merged.

Congrats! You just completed the standard *fork -> clone -> edit -> pull request* workflow that you'll often encounter as a contributor!

6 Markdown and Quarto

Verbal and non-verbal communication is important in business. This section is about writing and publishing texts, leaving out things like body language and writing skills. I will introduce some applications (Markdown, RMarkdown, Quarto) that data scientists often use to write and publish their work. I will also discuss the version control system *git* and the online platform *GitHub*, which can be used in combination with Quarto and Markdown to create, store, manage and share files. These tools are the backbone of most data science collaborations. Once you master these tools, they can significantly enhance your efficiency and make your presentations more impactful, even if you are not directly involved in the field of data science.

Quarto, a modern documentation system, is an excellent choice for writing, especially for projects that require rigorous data analysis, visualization, and reproducibility. This tutorial will guide you through producing various forms of text with Quarto. You can write reports, articles, theses, books, websites and many more with Quarto.

Step 1: Learn Markdown

Markdown is a lightweight markup language with plain-text formatting syntax. It's an essential skill for using Quarto effectively. Start by learning enough Markdown to structure your thesis, including headings, lists, links, and code blocks.

You can learn Markdown (not R Markdown!) in 10 minutes. Just go to www.markdowntutorial.com and work through the interactive lessons.

Step 2: Learn Quarto

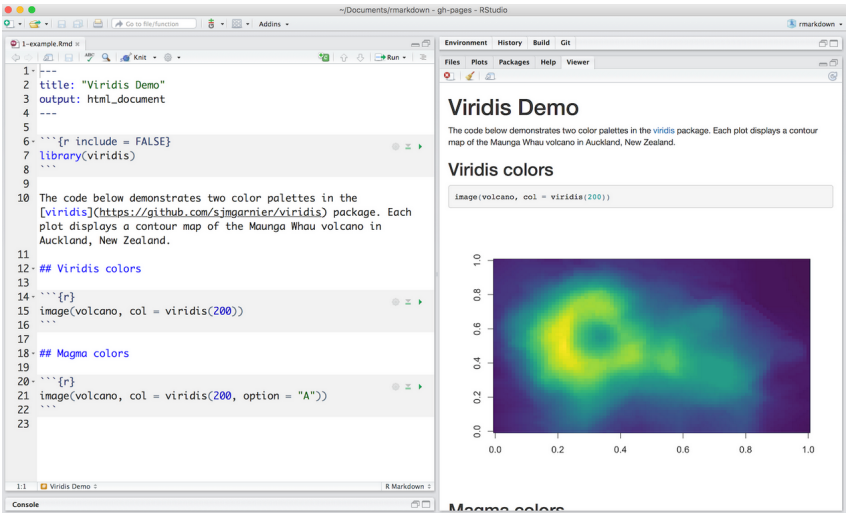
Read Telford [2023]: Enough Markdown to Write a Thesis. This resource covers the basics and some advanced Markdown features that are useful for academic writing.

More extensive resources on how to do things with Quarto can be found at quarto.org.

> **Quarto and R markdown**
>
> Quarto is a relatively new tool. It can be considered the successor to R Markdown, as it is built upon R Markdown. Consequently, almost all R Markdown documents are compatible with Quarto. However, Quarto includes several improvements over R Markdown that enhance its ease of use. For a detailed description of all the differences and similarities between the two, you can read this article. For an introduction to R Markdown see Chapter 7.

7 Write with R Markdown

Figure 7.1: Example of an R Markdown file

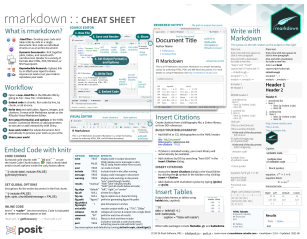


R Markdown provides an authoring framework for data science. You can use a single R Markdown file to transcript your work, run code, and generate high quality reports, books, websites, articles, theses, blogs, and many more (see Figure 7.1).

In contrast to Quarto (see Chapter 6), which is the more recent format, R Markdown is around for some time and hence there are uncountable resources to learn it. For example:

- The R Markdown Cheatsheet (see Figure 7.2) from Posit offers an overview on the most important features of R Markdown.

Figure 7.2: R Markdown Cheatsheet from Posit



- The book *R Markdown Cookbook* by Xie et al. [2020] (see Figure 7.3) offers an introduction. The online version of the book is regularly updated and free of costs.

- The book *R Markdown: The Definitive Guide* by Xie et al. [2018] offers a comprehensive introduction. The online version of the book is regularly updated and free of costs.

Please watch the video What is R Markdown? and then study the R Markdown tutorial from RStudio.

Figure 7.3: Xie et al. [2020]: R Markdown Cookbook

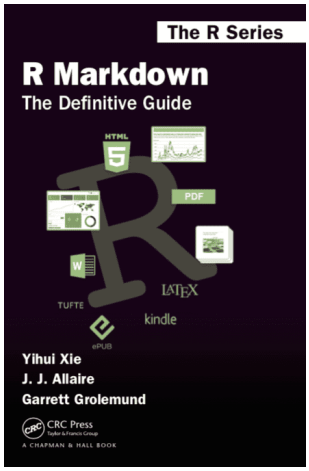
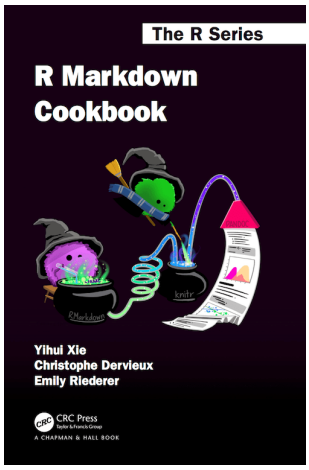


Figure 7.4: Xie et al. [2018]: R Markdown: The Definitive Guide



Working directory in R Markdown

The working directory is by default set to the directory that contains the Rmd document. In case you want to use another directory you can do so by changing the working directory with `setwd()`. However, that is not persistent in R Markdown and only works for the current code chunk. After the code chunk has been evaluated, the working directory will be restored to the directory where the Rmd file is placed.

Exercise 7.1. Start Markdown and R Markdown

a) You can learn Markdown (not R Markdown!) in 10 minutes. Just go to https://www.markdowntutorial.com and work throught the interactive lessons.
b) Now create your first R Markdown file in 3 minutes by doing the following:

- click in RStudio on *File > New File > R Markdown*
- click *OK*
- look for a button entitled *Knit* and click it
- save your file (it will be saved with .Rmd file extension)

c) Play around with the file. For example, change the output format can you create a word file or a presentation. Play around with the code chunks. Add a picture that you find somewhere online.
d) Set your working directory to the folder where you have saved your first Rmd-file. Can you come up with a way to generate different output format with just one function.

Exercise 7.2. R Markdown cite literature

a) Create a new R Markdown file (*File > New File > R Markdown*), save the file in an empty folder, and knit it.
b)

- Make a new script with *File > New File > R Script*.
- Go to https://scholar.google.de/ and search for *osrmtime*.
- Click on *"cite"* and *"BibTeX"*. Copy and paste everything that you see into your script and save the script as *lit.bib*. R Studio will ask you if you confirm the file type change. Click yes. Your *lit.bib* file should look like this:

```
@article{huber2016calculate,
  title={Calculate travel time and distance with OpenStreetMap
    data using the Open Source Routing Machine (OSRM)},
  author={Huber, Stephan and Rust, Christoph},
  journal={The Stata Journal},
  volume={16},
  number={2},
  pages={416--423},
  year={2016},
  publisher={SAGE Publications Sage CA: Los Angeles, CA}
}
```

- Add the text *"bibliography: references.bib"* to your YAML header of your R Markdown file so that it looks somehow like that:

```
---
title: "Untitled"
author: "Stephan Huber"
date: "`r Sys.Date()`"
output: html_document
bibliography: lit.bib
---
```

- Now you can cite the OSRMTIME paper with `@huber2016calculate` somewhere in the text of your R Markdown file.
- Knit the R Markdown file and you should see the paper cited and a reference list at the end of the html report.
- You can manipulate the citation style you can specify a CSL (Citation Style Language) file in the YAML header. For example the APA style can be chosen with:

```
csl: "https://www.zotero.org/styles/apa.csl"
```

Many more citation styles can be found on github.com/citation-style-language and on the Zotero Style Repository.

Exercise 7.3. Preparing APA journal articles (`papaja`)
There is an easy way to write a manuscript that follows all the APA rules using the package `papaja` written by two psychologists from Cologne. Please read their manual and consider their repository on GitHub.
Now, install and load the package:

```
install.packages("papaja")
library("papaja")
```

Then, click "File > New File > R Markdown" and choose the "APA-style manuscript" from the section "from template". Knit the R markdown template and you will have a template for a APA manuscript.
Apart from the obvious adjustments, I recommend to make at least two general adjustments: Change `classoption` to "doc" and `linenumbers` to "no".

Exercise 7.4. R Markdown template
Please follow the instructions below to access the file "23-09_ds-project-desc.Rmd" from my GitHub account:

1. Download the file from my GitHub account by clicking on the link provided here.
2. Save the file in your working directory.
3. Use the knit function to run the file, but be aware that it may not work properly at first. If you encounter any issues, troubleshooting may be required. Don't worry, error messages will usually provide guidance to help you resolve the issue. Please note that the YAML header is sensitive to spacing, so be careful when setting it up to avoid breaking the code.
4. In the project template, I have used BibTeX to cite literature. This method is excellent for automating tedious tasks such as citing papers and generating reference lists based on citation styles, saving time and reducing the likelihood of citation

> errors. The literature cited is in a separate file, which can be found on one of my GitHub repositories.

8 Create and host a website

8.1 Creating a website with Quarto

This tutorial guides you through creating a simple, yet professional-looking website using Quarto.

Step W1: Install Quarto

Ensure Quarto is installed on your system. If not, download and install it from Quarto's official website.

Step W2: Create a website

Follow the tutorial that you find here.

Step W3: Copy the `_site` directory

After you have rendered your website a directory "`_site`" appears in the project folder that contains your website. Copy all files of that directory to a directory where you want to save your website. Let's say `my_website`.

In the terminal you can do this with

```
mkdir /home/sthu/my_website/
cp -r /home/sthu/quarto_website/_site/* /home/sthu/my_website/
```

8.2 Hosting the website on GitHub

R Studio and Quarto offers you various ways to publish the website. I explain you a way that worked out well for me.

Step G1: Create a GitHub account

GitHub will host your thesis website and manage version control for your thesis project. If you don't already have a GitHub account, you'll need to create one: Sign up at GitHub.

Step G2: Create a repository

Create a repository. Name the repo with your username followed by *github.io*. You find a tutorial here.

Step G3: Obtain a personal access token

A personal access token (PAT) is required to authenticate with GitHub from Quarto and RStudio. This token allows you to push changes to your repository securely. Follow the instructions to create a personal access token on GitHub. Alternatively, you can do the following in R:

```
if (!require(pacman)) install.packages("pacman")
pacman::p_load(usethis)
create_github_token()
```

Make sure to note down your token and keep it secure. You'll use this token in RStudio and Quarto to authenticate your GitHub operations.

Step G4: Install and Learn Git

See Section 5.2.

Step G5: Upload the website to GitHub

Use the Terminal of R Studio. Go to the directory with your website that you have copied in Step W3. Then initiate a git repository on the command line, connect it to the repository created in Steph G2 on GitHub and finally push it:

```
cd /home/sthu/my_website/
echo "# test" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/test-hsf/test.git
git push -u origin main
```

Alternatively, you can clone a repository, make some changes, and then push those changes back to GitHub. Here are the Bash commands to accomplish this:

```
# Clone the repository
git clone https://github.com/your-username/your-repository.git

# Make changes, here adding a new file as an example
echo "Some content for the new file" > newfile.txt

# Add the new file to the repository
git add newfile.txt

# Commit the changes
git commit -m "Add new file"

# Push the changes back to GitHub
git push origin main
```

References

Jennifer Bryan. Happy git and github for the user. URL https://happygitwithr.com/.

Thomas H Davenport and DJ Patil. Data scientist: The sexiest job of the 21st century. *Harvard Business Review*, 90(5):70–76, 2012.

Aud Halbritter and Richard J Telford. Version control with git and github. URL https://biostats-r.github.io/biostats/github/.

Stephan Huber. Quantitative methods, 2024a. URL https://hubchev.github.io/qm/.

Stephan Huber. How to use R for data science, 2024b. URL https://hubchev.github.io/ds/.

Roger Martin. How successful leaders think. *Harvard Business Review*, 85(6):75–83, 2007.

OpenAI. Chatgpt. Large language model, 2024. URL https://chat.openai.comt.

Richard J Telford. Enough markdown to write a thesis, 9 2023. URL https://biostats-r.github.io/biostats/quarto/.

Jeffrey M. Wooldridge. *Introductory Econometrics: A Modern Approach*. South-Western, 2nd edition, 2002.

Yihui Xie, Joseph J. Allaire, and Garrett Grolemund. *R Markdown: The Definitive Guide*. Chapman and Hall/CRC, 2018.

Yihui Xie, Christophe Dervieux, and Emily Riederer. *R Markdown Cookbook*. Chapman and Hall/CRC, 2020. available at https://bookdown.org/yihui/rmarkdown-cookbook.