教育部教學實踐研究計畫成果報告
Project Report for MOE Teaching Practice Research Program

計畫編號/Project Number：PMS1122277

學門專案分類/Division：數理學門

計畫年度：112 年度一年期

執行期間/Funding Period：2023.08.01 – 2024.07.31

# 使用成對學生教學法進行數據資料分析課程

(商業電腦套裝軟體/ Computer Tools for Business)

計畫主持人(Principal Investigator)：鍾冬川

執行機構及系所(Institution/Department/Program)：逢甲大學統計學系

成果報告公開日期：■立即公開□延後公開(於 2023 年 9 月 30 日公開)

繳交報告日期(Report Submission Date)：2024 年 9 月 10 日

# 使用成對學生教學法進行數據資料分析課程

## 本文 (Content)

### 1. 研究動機與目的 (Research Motive and Purpose)

The motivation for this teaching and learning research proposal is to hopefully improve both teacher teaching skills and students' programming skills. On the teacher's side, he/she shall collect new information (here the new data analytic skills) and organize it into the course material, he/she also needs to execute the teaching strategy and observe how the students response. From the students' side, one would anticipate that they would become more interested in participating in the class. In particular, with more interactions with peers, they could feel more belonging and gain more feedback from each other. That would make their study in college more interesting and get them more involved. The purpose of this teaching and learning research proposal is to enhance learning motivation. With more students having a stronger desire to learn, the course would provide solid and rigorous training for future data scientists.

### 2. 研究問題 (Research Question)

Currently, we identify that when teaching both programs in a computer lab where the teacher tries to offer the onsite programming skill training, most students seem not to be involved in developing the scripts and enjoy it. Rather, for lab time where students are asked to solve a similar problem, they prefer to use the developed script and modify it to solve the problems assigned by the teacher. Under this circumstance, students do solve the problem and give the correct solution, however, it remains ineffective for students to acquire good programming skills. Based on that programming is an essential skill for data analytics in the big data era (Saeli et al. 2011), this proposal aims to use teaching strategies for improving students' learning effect.
Below we first provide the current course material (current course materials were developed and written in HTML format and are uploaded online for direct access)

## Python Programming

The first 9 weeks is designed for teaching the python programming language, the main text used is Python Programming for the Absolute Beginner, 3rd Edition(Dawson 2010). Some basic commands like how to use a keyboard, and install and use the software are introduced. Later, we introduce how to use python to create games such as tic tac toe, guess my number, word jumble. This game-oriented teaching has been studied to be effective in developing computational thinking (Allsop 2019) where computer game designs was also included.

The main courses material are divided into three subsections: 1. What’ s new, 2. What’ s due and 3. Class notes.

R programming (week 10 to week 18)

The follow up 9 weeks (week 10 – week 18) is the R programming where the main textbook is (Anderson et al. 2017) that various topic such as the discrete probability distributions (binomial, Poisson) and continuous probability distributions (uniform, normal, exponential, chi-square, and F).
The hypothesis tests for the population mean confidence interval, comparing multiple proportions, a test of independence and goodness of fit, simple linear regression, coefficient of determination, and correlation coefficient are incorporated by various statistics packages ggplot2 (Wickham 2016) for graphical Visualization, MASS (Venables and Ripley 2002) for computing the statistics.

R Shiny MySQL

*Computer Tools for Business*

with R **R Programming**

What's New? What's Due? Class Notes

The main courses material are divided into three subsections: 1. What’ s new, 2. What’ s due, and 3. Class notes.

## 文獻探討 (Literature Review)

Teaching data analytics using computing tools requires tremendous effort in programming. In the past, the lecturer adopts the conventional teaching methodology of writing up the script in class simultaneously to show students how the teacher develops the skill.

However, this teaching method requires more improvement to enhance students' learning. This proposal aims to teach coding by adopting the pair/group coding mechanism where students are paired/grouped to deal with the problems and propose their solutions.

In addition, the classmates in other groups would also investigate the validity of scripts from other groups and then give their feedback for further discussion and response.

On the other hand, in the current big data era, there are many established data science companies that provide many resources and advocate the newest technique in data analysis. Concerning the contents of the school courses shall also meet the need for the future career requirement, this proposal aims to develop notes implement and integrate useful and appropriate online resources developed by the companies into the current curriculum. In these courses, the lectures are provided mainly for teaching two popular programming languages: Python (Van Rossum and Drake Jr 1995) and R (R Core Team 2022). In the first half semester (week 1 to week 9), teaching programming using the python language is planned, and the python is executed under the integrated platform Anaconda ("Anaconda Software Distribution" 2020) where the Spyder (Raybaut 2009) program is used for developing script (see below Figure 1 for a demonstrative example).

We hope that by adopting the novel methods (paired/group coding) and course material, students would benefit enough to gain sufficient programming training to meet the prerequisite for their study in the program and improve their programming skill by learning the latest data science techniques and information. Hopefully, with rigorous training, students would be competitive in their oversea study and the future job market.

教學設計與規劃 (Teaching Planning)

| Week | Content | Evalution | Tool for Evaluation |
|---|---|---|---|
| 1 | Python: Ch1. Getting Started | lab quiz/homework | Explain the **paired/group coding and feedback** system |
| 2 | Python: Ch2. Types, Variables, and Simple I/O | lab quiz/homework | group feedback and evaluation |
| 3 | Python: Ch3. Branching, While Loops, and Programming Planning | lab quiz/homework | group feedback and evaluation |
| 4 | Python: Ch4. For Loops, Strings, and Tuples | lab quiz/homework | group feedback and evaluation |
| 5 | Python: Ch5. Lists ad Dictionaries | lab quiz/homework | group feedback and evaluation |
| 6 | Python: Ch6. Functions | lab quiz/homework | group feedback and evaluation |
| 7 | Python: Modules and Data Libraries | lab quiz/homework | group feedback and evaluation |
| 8 | Review | | |
| 9 | Midterm: Python | midterm | |
| 10 | R: Part I – Getting Started | lab quiz/homework | group feedback and evaluation |
| 11 | R: Part II — Basic Methods– Graphs I: bar chart, pie chart | lab quiz/homework | group feedback and evaluation |
| 12 | R: Part II — Basic Methods– Graphs II: cross table, scatter plot, boxplot | lab quiz/homework | group feedback and evaluation |
| 13 | R: Part II – Basic Methods– Statistics I: Discrete and continuous probability distributions | lab quiz/homework | group feedback and evaluation |
| 14 | R: Part II – Basic Methods– Statistics II: Test of population means | lab quiz/homework | group feedback and evaluation |

| Week | Content | Evalution | Tool for Evaluation |
|---|---|---|---|
| 15 | R: Part III — Function and Intermediate Methods I: Test of Independence | lab quiz/homework | group feedback and evaluation |
| 16 | R: Part III — Function and Intermediate Methods II: Simple linear regression | lab quiz/homework | group feedback and evaluation |
| 17 | Review | | |
| 18 | Final Exam | | |

研究設計與執行方法 (Research Methodology)

The research participants are freshman students in the Spring semester international school of technology and management in the business Analytics program and plan to go to San Jose University USA or other universities abroad after their two years of study at Feng-Chia University. The research sites are the computer labs where the teacher conducts the lectures by showing the coding to students. To collect the novel teaching, we will group/pair students in the first-week lectures. If necessary we can rotate the group/pairs for the need of being more compatible with each other or to reach a better working atmosphere.

To examine the effectiveness of the novel teaching methods (pair coding) and answer your research questions (do pair coding to improve the ability of programming skill). The following method are described

Pair/triple/group coding:

We adopt the idea from Pair programming which is an agile software development technique in which two programmers work together at one workstation. But being more lenient for those in a group that is willing to help each other, so would allow more flexibility that would help in learning programming in a group.

While one student mainly writes code, the others students would serve as the observers or navigators who would review each line of the code as it is typed in. We are aware that although pair programming could be helpful for attaining high quality and correctness on complex programming tasks, it would also increase the development effort (cost) significantly (Hannay et al. 2009). However, concerning that the benefit of pairing is greatest on tasks that the programmers do not fully understand before they begin: that is, challenging tasks that call for creativity and sophistication, and for novices as compared to experts (Lui and Chan 2006). On simple tasks, which the pair already fully understands, pairing results in a net drop in productivity (Lui and Chan 2006; Arisholm et al. 2007). And productivity can also drop when the pairing is used without sufficient availability of a mentor to coach them (Williams and Kessler 2003). It may reduce the code development time but also risks reducing the quality of the program.

Given the above pros and cons, we remain positive and would like to make an attempt to implement the pair coding strategy in our course. We believe that with the group feedback (see section 5.2 below), the negative effects would be mainly remediated.

Group feedback and validation.

Below we list the step-by-step procedure for the next step group feedback.

- Step 1. Ask one group to provide the script to another group which can rerun the script and tries to reproduce the same results.

By doing so, the students can, on one hand, investigate the technical sound of their peers scripts, on the other they can learn from others thoughts of computer thinking. This would mutually benefit each other.

- Step 2. then to give question on which script needs more explanation
- Step 3. the group shall answer the feedback question and then propose their solution to those questions.

## 教學暨研究成果 (Teaching and Research Outcomes)

### (1) 教學過程與成果

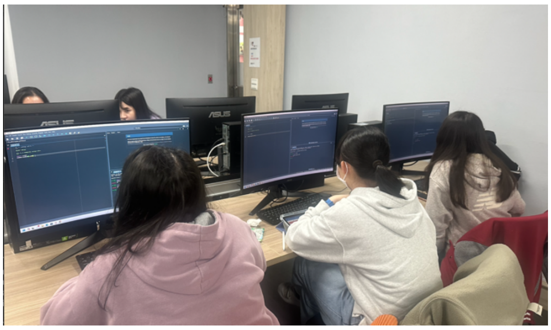
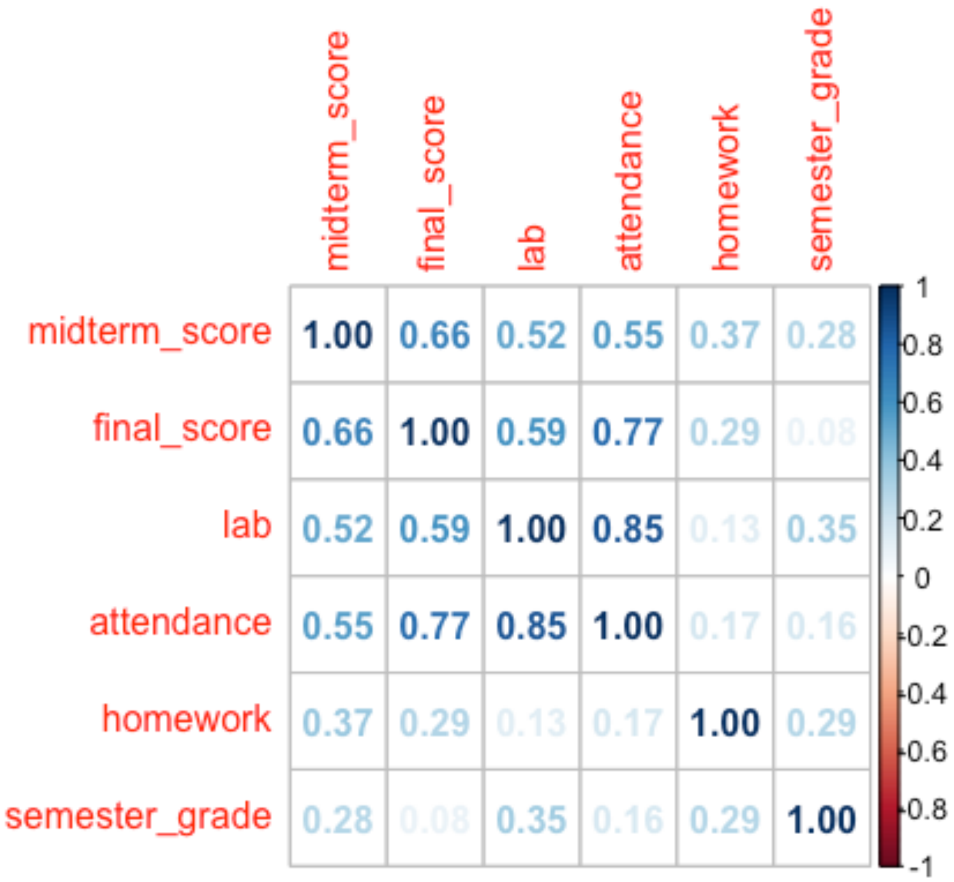


Figure 1: Students in class with coding lesson

Data Collection and analysis

This is the course for the data analysis, as shown in the above statement the data are mainly from the statistics textbook (Anderson et al. 2017) where we would like to use descriptive statistics (graphical visualization and tabular presentation) as well as inferential statistics (hypothesis testing, a test of the population mean, independent test, ANOVA, and linear regression).

For each week, as usual, we will introduce the lecture in the first two hours, on the 3rd hour, is the lab time for students to practice the work. With the newly developed course material, we hope students can learn the latest knowledge in data analysis. In addition, after the new courses and novel pair coding teaching are implemented, we would correct the feedback from the midterm evaluation. We perform the summary statistics and visualization (boxplots, violin plot, scatter plots, correlation matrix–in heat map with number)



| | midterm_score | final_score | lab | attendance | homework | semester_grade |
|---|---|---|---|---|---|---|
| midterm_score | 1.00 | 0.66 | 0.52 | 0.55 | 0.37 | 0.28 |
| final_score | 0.66 | 1.00 | 0.59 | 0.77 | 0.29 | 0.08 |
| lab | 0.52 | 0.59 | 1.00 | 0.85 | 0.13 | 0.35 |
| attendance | 0.55 | 0.77 | 0.85 | 1.00 | 0.17 | 0.16 |
| homework | 0.37 | 0.29 | 0.13 | 0.17 | 1.00 | 0.29 |
| semester_grade | 0.28 | 0.08 | 0.35 | 0.16 | 0.29 | 1.00 |

Figure 2: Correlation among the variables of interest in this study

We combine all score from 2018, 2019, 2020,2021, 2022 and 2023. And then perform the linear regression analysis pm the midterm, final, lab attendance homework correlation matrix displaying the relationships between different variables related to student performance, including midterm scores, final scores, lab performance, attendance, homework, and semester grades. The matrix shows Pearson correlation coefficients, where values close to 1 indicate a strong positive correlation, and values close to -1 indicate a strong negative correlation.

Key observations include:

- The highest positive correlation is between attendance and lab performance (0.85).
- Final scores are moderately correlated with midterm scores (0.66) and attendance (0.77).
- Semester grades have the weakest correlation with most variables, particularly with midterm scores (0.28) and final scores (0.08), suggesting that semester grades may be influenced by factors beyond these individual assessments.

### *Multiple Regression*

```
We use regression to search the potential factors for the
performance of the semester grade.
semester_grade ~ midterm_score + final_score + homework + lab
+ attendance
```

We find that the intercept is significant with an estimate of 62.90, indicating the baseline semester grade when all predictors are zero.
The Lab (Estimate: 2.30, $p < 0.001$) and homework (Estimate: 1.34, $p = 0.00106$) are significant positive predictors of semester grades.
Attendance has a negative and significant effect on semester grades (Estimate: -1.11, $p = 0.01533$).
Midterm score is marginally significant (Estimate: 0.096, $p = 0.05412$).
Final score is not a significant predictor ($p = 0.10806$).
The model explains approximately 28.2% of the variance in semester grades (Multiple R-squared = 0.2821, Adjusted R-squared = 0.2572). The overall model is significant with an F-statistic of 11.32 ($p < 0.001$).
This suggests that while the predictors collectively contribute to explaining semester grades, other unmeasured factors may also play a role.

### *Testing for 2024 performance*

#### *Midterm and Final Scores Mean Differences*

The paired t-test was conducted to compare the midterm and final scores for the year 2024. The results indicated a statistically significant difference between the two sets of scores, with a t-value of -3.8059, degrees of freedom (df) of 20, and a p-value of 0.001107. The mean difference between the midterm and final scores was -10.33, with a 95% confidence interval ranging from -15.997 to -4.670. These results suggest that there is a significant decrease from the midterm scores to the final semester grades after performing the pair coding.

### (2) 教師教學反思

Teaching in this course is exciting for me. Every week, before the class, I review the content and prepare notes for the lecture. Students arrive on time and work on the lecture problem set with me. Sometimes we work on the homework together before moving on to a new topic. It is a really good experience for students to participate in the pair coding activities. On one hand, students become aware that their peers might have different understandings; on the other hand, students get opportunities to work closely together, allowing them to become acquainted easily, learn together, and share the workload.

I also find that teaching in this way improves the interaction between me and the students. Sometimes we do team recordings and have the coding process recorded for students to review, which some find very useful.1

學生學習回饋

1. 鍾冬川老師是我在國際學院雙學位學程的專任老師，教授我商業程式套件 Python 和 R 語言的程式設計，本學期開始使用成對教學把我們分成二人一組，一起檢查程式，我發現有所幫助。

2. 在課堂上，鍾老師每週都會準備豐富的教材，一步步指導我們這些學生學習程式語言設計，也會跟我們成對分組後所學習的問題做檢查。

3. 鍾老師對教學很熱誠， 每週準備作業讓我們知道自己對 Python 和 R 語言了解多少，並且透過專題報告，實作在課堂上學到的知識， 鍾老師也會為我們細心地點評，以讓我們在之後的報告上更進一步。

4. 當學生遇到問題時， 鍾老師也會馬上解決我們的疑惑，就算遇到 ios 和 windows 作業系統不一樣導致程式出錯時，老師也不辭辛勞地幫我們尋找答案，只為了我們能跟上他的教學腳步，而不是對這程式語言感到挫折、灰心。

5. 有時有小組學生有程式方面的問題無法一一處理完，老師會幫忙他們哪段程式出錯需要修改，對程式設計有更深入的了解。

## 參考文獻 (References)

Adie, Lenore, Fabienne van der Kleij, and Joy Cumming. 2018. "The Development and Application of Coding Frameworks to Explore Dialogic Feedback Interactions and Self-Regulated Learning." British Educational Research Journal 44 (4): 704–23.

Allsop, Yasemin. 2019. "Assessing Computational Thinking Process Using a Multiple Evaluation Approach." International Journal of Child-Computer Interaction 19: 30–55.

"Anaconda Software Distribution." 2020. Anaconda Documentation. Anaconda Inc. https://docs.anaconda.com/.

Anderson, David R, Dennis J Sweeney, Thomas A Williams, Jeffrey D Camm, and James J Cochran. 2017. Essentials of Statistics for Business and Economics. Cengage Learning.

Arisholm, Erik, Hans Gallis, Tore Dyba, and Dag IK Sjoberg. 2007. "Evaluating Pair Programming with Respect to System Complexity and Programmer Expertise." IEEE Transactions on Software Engineering 33 (2): 65–86.

Bers, Marina Umaschi. 2019. "Coding as Another Language: A Pedagogical Approach for Teaching Computer Science in Early Childhood." Journal of Computers in Education 6 (4): 499–528.

Dawson, Michael. 2010. Python Programming for the Absolute Beginner. Course Technology Boston, MA.

Faja, Silvana. 2014. "Evaluating Effectiveness of Pair Programming as a Teaching Tool in Programming Courses." Information Systems Education Journal 12 (6): 36.

Fröberg, Andreas, and Suzanne Lundvall. 2022. "Sustainable Development Perspectives in Physical Education Teacher Education Course Syllabi: An Analysis of Learning Outcomes." Sustainability 14 (10): 5955.

Hannay, Jo E, Tore Dybå, Erik Arisholm, and Dag IK Sjøberg. 2009. "The Effectiveness of Pair Programming: A Meta-Analysis." Information and Software Technology 51 (7): 1110–22.

Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, et al. 2020. "Array Programming with NumPy." Nature 585 (7825): 357–62.

https://doi.org/10.1038/s41586-020-2649-2.

Henry, Lionel, and Hadley Wickham. 2020. Purrr: Functional Programming Tools. https://CRAN.R-project.org/package=purrr.

Kong, Siu-Cheung, Ming Lai, and Daner Sun. 2020. "Teacher Development in Computational Thinking: Design and Learning Outcomes of Programming Concepts, Practices and Pedagogy." Computers & Education 151: 103872.

Krekel, Holger, Bruno Oliveira, Ronny Pfannschmidt, Floris Bruynooghe, Brianna Laugher, and Florian Bruhin. 2004. "Pytest x.y." https://github.com/pytest-dev/pytest.

Lui, Kim Man, and Keith CC Chan. 2006. "Pair Programming Productivity: Novice–Novice Vs. Expert–Expert." International Journal of Human-Computer Studies 64 (9): 915–25.

Miller, Jodie. 2019. "STEM Education in the Primary Years to Support Mathematical Thinking: Using Coding to Identify Mathematical Structures and Patterns." Zdm 51 (6): 915–27.

Mosleh, Mohsen, and David G Rand. 2022. "Measuring Exposure to Misinformation from Political Elites on Twitter." Nature Communications 13 (1): 1–9.

Pipere, Anita, Marika Veisson, and Ilga Salīte. 2015. "Developing Research in Teacher Education for Sustainability: UN DESD via the Journal of Teacher Education for Sustainability." Journal of Teacher Education for Sustainability 17 (2): 5–43.

Popat, Shahira, and Louise Starkey. 2019. "Learning to Code or Coding to Learn? A Systematic Review." Computers & Education 128: 365–76.

"Posit: The Open-Source Data Science Company." n.d. https://posit.co/.

Prasad, Alvin, Kaylash Chaudhary, and Bibhya Sharma. 2022. "Programming Skills: Visualization, Interaction, Home Language and Problem Solving." Education and Information Technologies 27 (3): 3197–3223.

R Core Team. 2022. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Raybaut, Pierre. 2009. "Spyder-Documentation." Available Online at:

Pythonhosted. Org.

RStudio Team. 2020. RStudio: Integrated Development Environment for r. Boston, MA: RStudio, PBC. http://www.rstudio.com/.

Saeli, Mara, Jacob Perrenet, Wim MG Jochems, and Bert Zwaneveld. 2011. “Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective.” Informatics in Education 10 (1): 73–88.

Salleh, Norsaremah, Emilia Mendes, and John Grundy. 2010. “Empirical Studies of Pair Programming for CS/SE Teaching in Higher Education: A Systematic Literature Review.” IEEE Transactions on Software Engineering 37 (4): 509–25.

Schneider, Daniel, and Kristen Harknett. 2022. “What’s to Like? Facebook as a Tool for Survey Data Collection.” Sociological Methods & Research 51 (1): 108–40.

Sentance, Sue, Jane Waite, and Maria Kallia. 2019. “Teaching Computer Programming with PRIMM: A Sociocultural Perspective.” Computer Science Education 29 (2-3): 136–76.

Sharp, Jason H, and Guido Lang. 2018. “Agile in Teaching and Learning: Conceptual Framework and Research Agenda.” Journal of Information Systems Education 29 (2): 45–52.

team, The pandas development. 2020. Pandas-Dev/Pandas: Pandas (version latest). Zenodo. https://doi.org/10.5281/zenodo.3509134.

Thompson, Maris, and Alfred Schademan. 2019. “Gaining Fluency: Five Practices That Mediate Effective Co-Teaching Between Pre-Service and Mentor Teachers.” Teaching and Teacher Education 86: 102903.

Threekunprapa, Arinchaya, and Pratchayapong Yasri. 2020. “Unplugged Coding Using Flowblocks for Promoting Computational Thinking and Programming Among Secondary School Students.” International Journal of Instruction 13 (3): 207–22.

Ushey, Kevin, JJ Allaire, and Yuan Tang. 2022. Reticulate: Interface to ’Python’. https://CRAN.R-project.org/package=reticulate.

Van Rossum, Guido, and Fred L Drake Jr. 1995. Python Reference Manual. Centrum voor Wiskunde en Informatica Amsterdam.

Venables, W. N., and B. D. Ripley. 2002. Modern Applied Statistics with s.

Fourth. New York: Springer. https://www.stats.ox.ac.uk/pub/MASS4/.

Virtanen, Pauli, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, et al. 2020. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python.” Nature Methods 17: 261–72. https://doi.org/10.1038/s41592-019-0686-2.

Wickham, Hadley. 2016. Ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D’Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. “Welcome to the tidyverse.” Journal of Open Source Software 4 (43): 1686. https://doi.org/10.21105/joss.01686.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2022. Dplyr: A Grammar of Data Manipulation. https://CRAN.R-project.org/package=dplyr.

Williams, Laurie, and Robert R Kessler. 2003. Pair Programming Illuminated. Addison-Wesley Professional.