NYC DATA SCIENCE ACADEMY

R Data Analysis Project Rubric

Use this rubric as a guide to completing a successful R Data Analysis project. These are the criteria the instructors will use to evaluate your project.

	Excellent (A) **5 points** Exceeds expectations	**Good (B)** **4 points** Meets expectations	**Fair (C)** **3 points** Meets lowest acceptable standards	**Poor (F)** **1 point** Doesn't meet acceptable standards
Project Submission Criterion related to timely submission and presentation of the project. **Weight: 15%**	Project presented on time, presentation shared correctly, code is hosted on GitHub repository, and there is strong evidence of version control.	Project presented on time, presentation shared correctly, code is hosted on GitHub repository, but there is no evidence of version control.	Project presented on time, but presentation not shared, and code not hosted on GitHub repository.	Project not presented on time, no presentation shared, and code not hosted on GitHub repository.
Purpose Criterion related to the purpose of the project. **Weight: 20%**	Objective of the problem statement is clearly defined and demonstrates a value proposition. The data chosen is relevant to answering the question asked.	Objective of the problem statement is defined and loosely demonstrates a value proposition. The data chosen is relevant to answering the question asked.	Objective of the problem statement is vaguely defined and does not demonstrate a value proposition. The data chosen is somewhat relevant to answering the question asked.	No objective of the problem statement is defined. The data chosen is not relevant to the project goals.
Tools and Methods Criterion related to tools and methods used in the project. **Weight: 20%**	Demonstrates correct usage of tools and methods learned in the coursework and selects those appropriate to solve tasks at hand. (For Shiny App) User experience is taken into consideration when designing the app (user manual/glossary given where appropriate). Appropriate widgets are effectively, and sometimes novelly, implemented to allow the user to interact with the app.	Demonstrates correct usage of tools and methods learned in the coursework and selects those which are somewhat appropriate to solve tasks at hand. (For Shiny App) Some evidence that user experience is taken into consideration when designing the app. Appropriate widgets are implemented to allow the user to interact with the app.	Usage of tools and methods learned in the coursework is often incorrect or misunderstood. (For Shiny App) Limited evidence that user experience is taken into consideration when designing the app. Limited number of widgets, sometimes of inappropriate type, are implemented to allow the user to interact with the app.	Limited to no usage of tools and methods learned in the coursework and it is mostly incorrect or misunderstood. (For Shiny App) App does not function as intended or takes prohibitively long to load content. Limited or no interactivity implemented.

	Excellent (A) **5 points** Exceeds expectations	**Good (B)** **4 points** Meets expectations	**Fair (C)** **3 points** Meets lowest acceptable standards	**Poor (F)** **1 point** Doesn't meet acceptable standards
Outcome Criterion related to the outcome of the project. **Weight: 15%**	Project achieves stated objectives and evidence-based actionable steps for future recommendations are given. (For Shiny App) Application functionality serves the stated purpose and demonstrates value for the target audience.	Project somewhat achieves stated objectives and limited actionable steps for future recommendations are given. (For Shiny App) Application functionality somewhat serves the stated purpose and demonstrates value for the target audience.	Project fails to achieve stated objectives, but some actionable steps for future recommendations are given. (For Shiny App) Application functionality somewhat serves the stated purpose but demonstrated value for the target audience is not clear.	Project fails to achieve stated objectives and no actionable steps for future recommendations are given. (For Shiny App) Application functionality does not serve the stated purpose and there is little to no demonstrated value for the target audience.
Presentation Criterion related to the presentation of the project. **Weight: 30%**	The student uses the app, a prepared slide deck, or a combination of both to effectively present their project goals, analyses, and outcomes in a logical and easy-to-follow way. Visualizations used effectively convey the intended messages. Presentation clearly indicates that time was dedicated for development and practice of the presentation.	The student uses the app, a prepared slide deck, or a combination of both to effectively present their project goals, analyses, and outcomes in a logical way. Visualizations used mostly convey the intended messages. Presentation indicates that some time was dedicated for development and practice of the presentation.	The student uses the app, a prepared slide deck, or a combination of both to present their project goals, analyses, and outcomes in a way that is difficult to follow. Visualizations used have difficulty conveying the intended messages. Presentation indicates that minimal time was dedicated for development and practice of the presentation.	The student presents their project goals, analyses, and outcomes in a way that is hard to follow. Few to no visualizations used. Presentation indicates that little to no time was dedicated for development and practice of the presentation.

Note: This criterion may be used at the instructor's discretion to alter the overall grade of the project. These categories are designed to help guide students to follow common best practices and produce high quality code.

	Excellent (A) **5 points** Exceeds expectations	**Good (B)** **4 points** Meets expectations	**Fair (C)** **3 points** Meets lowest acceptable standards	**Poor (F)** **1 point** Doesn't meet acceptable standards
Code (For Discretionary Consideration) Criterion related to the usage of proper coding techniques and practices.	Code is organized into scripts, each with a clear purpose, and are given meaningful file names. Sufficient commenting/docstring is used to describe the functionality of the code. In Jupyter Notebooks, markdown is used to partition the code into logical sections. Code is modular, robust, efficient and demonstrates an understanding of best practices (such as using helper functions). Code has no syntax errors and follows the standard formatting style.	Code is organized into scripts, each with a vague purpose. Some commenting is used to describe the functionality of the code. In Jupyter Notebooks, markdown is used occasionally to partition the code into sections. Code is somewhat robust and efficient but has room for improvement. Code may have minor syntax errors and mostly follows standard formatting style.	Code is not organized into scripts. Minimal commenting is used to describe the functionality of the code. In Jupyter Notebooks, markdown is not used to partition the code into sections. Code is repetitive and inefficient. Code has several syntax errors and often ignores standard formatting style.	Code does not run successfully, and no commenting is used to describe the functionality of the code. Code is riddled with syntax errors and completely ignores standard formatting style.
Shiny App Organization (For Discretionary Consideration) Criterion related to the proper organization of project files.	Shiny code is organized into three separate partitions or scripts (global, ui, server). Code is organized in a way that is easy to interpret (correct indentation, code sections are logically partitioned and ordered). Project directory and files (images, data, etc.) are organized in a logical way.	Shiny code is organized into three separate partitions or scripts (global, ui, server). Code is somewhat organized (mostly correct indentation, code sections are logically partitioned and ordered). Project directory and files (images, data, etc.) are organized in a logical way.	All Shiny code is contained in a single script. Code is loosely organized (often incorrect indentation, code sections have a somewhat logical order). Project directory and files (images, data, etc.) are organized into one subfolder.	All Shiny code is contained in a single script. Code is not organized (incorrect indentation, code sections have no logical order). Project directory and files (images, data, etc.) are all located in the root project folder with no subfolder structure.