

Score	Stage	Concrete Examples	Notebook Examples
0.1	<i>Understanding the data</i>	Just code (1-2 unorganized cells, no output) Just code presented non-linearly [pimental, wenskovitch, wang, rule, chattopadhyay, head] or chronologically [kery-story]. Each code cell may serve multiple purposes, duplicated [koenzen] and are not constructed to separate functionally different code [rule-exploration]. We do not expect that these lines correspond to a single "logical step" in the analysis [kery-story]. Most code cells are either unrelated or loosely related to one another [rehman]. No annotations can be found, either in the form of code comments or markdown text.	https://github.com/oliverlee/bikesim/blob/3fbf33a714fccb466d0890412654d722b810c42d/scripts/plot_group_data.ipynb
0.2	<i>Iterative data wrangling</i>	Just code, some output (lots of unorganized and disjoint cells, code organized within multiple cells, some output) These notebooks contain signs of development. Some notebook code cells may contain 1-2 lines of code for the sake of fast iteration [kery-story, chattopadyay]. Code may have lots of issues like "key errors", "syntax errors" or naming errors [wang-assessing]. Notebooks maybe full of empty cells separated sections of functionally different code [kallen]. Code cells maybe executed out of order [pimental, wang, kallen, chattopadhyay]. Authors may leverage code duplication for the sake of maintaining a chronological structure [kery-story, koenzen, kallen, pimental, rule, chattopadhyay]. Most code cells are either unrelated or loosely related to one another [rule, rehman-understanding]. Code output are unlabeled, not detailed (lacking legends, headers or other hits to ease understanding), and they are leveraged for the sake of exploration. Minimal to no hints about the code cell dependencies are given through code comments or markdown text.	https://github.com/JedrzejMosiezn/R67-data-analysis/blob/master/_jupyter/fft_analysis_ver02.ipynb
0.3	<i>Defining goals using iterative data exploration</i>	Lots of code and lots of messy output (some disjoint cells are grouped by functionality, text headers and code comments are used to label groups) Code cells are grouped into sections for the sake of reuse, maintainability and readability [rule-reproduce, rule, dong, pimental, koenzen, chattopadhyay, head-messes]. Sections of code cells may be create a cohesive story of the analysis step but these sections do not relate to provide a clearly picture of an entire workflow [pimental, rule, dong]. Code maybe commented out [dong, chattopadhyay]. Code comments are used to label tested	https://github.com/scopatz/fc-deploy-opt/blob/281752b48fc9fcb534037834b9b8c45359ef2c41/scratch.ipynb

		code, or logical steps [rule, dong] . Code output are liberally leveraged for the sake of exploration and "interwoven" with different sections of code for the sake of tracking analysis [kery-story] . However, output is still "messy".	
0.4	<i>Beginning goal-oriented exploratory analysis</i>	Code interwoven with some of visual output (text headers and code comments are used to label and annotate analysis) Code is grouped by logical steps within individual code cells for the sake of maintainability and readability [kery-story, rule-reproduce]. Code cells are mostly interrelated and create at least a one cohesive story [rehman-understanding, kallen]. For example, authors may group code into functions or designate the top code cell for importing libraries [kery-story, pimental]. Authors mostly favor plots or visualizations over text output due to the amount which can be verified and communicated through those types of output [chattopadhyay]. Some subset of code content fit into a pipeline or flow of analysis---they are dependent [rule-reproduce, guo, rule]. There are remaining signs of disjoint iteration. For example, a small subset of this notebook could contain sections of dead-end iterations, buggy code, duplicated code or commented out code [wang, dong].	https://github.com/xingliuUT/battery_planner/blob/73d7eff1bd10575c23bd0ad29ed13199902613e3/nn_model.ipynb
0.5	<i>Exploratory analysis with clear goals</i>	Code interwoven with lots of visual output (text headers and code comments are used to label groups) The primary mechanisms for communicating the workflow is code and visual output [kery-story]. Code is grouped by logical steps in individual code cell [kery-story, rule]. Code and code output are well structured, labeled and formatted for the sake of readability [rule]. All notebook content highlight a critical decision point in the workflow [kery-story]. These notebooks lack messy [head], dead-end iteration, buggy code, or commented out code [rule, dong]. Code comments and text headers hint do not at the notebook's narrative---they label and separate sections [rule, wang]. The narrative primary communicates, through code, the methods being taken by the analyst.	https://github.com/b BradleyMwest/Renishaw-Reni-Awesome/blob/7dd818170656728eefbac648246b58f7f0093a38/PL_Raman_Analysis.ipynb
0.6	<i>Analytical steps are communicated</i>	Code and code output are interwoven with text headers, and code comments for the sake of outlining the logical steps which were taken. The primary mechanisms of forming a narrative in the notebook are code, code comments and markdown text [kery-story]. In these notebooks, we observe a heavier use	https://github.com/michailkoupparis/MatrixTensorFactorization/blob/03bfcca394eeea3c9865ee11b6b6b0f7ac05955b/testPoission.ipynb

		of code comments and text in order to track, and label individual analytical steps [wang, kery]. All components of the notebooks primarily seek to communicate the methods being followed by the authors [rule, wang, kery]. These notebooks explicitly communicate the methods being taken by the analyst.	
0.7	<i>Some insights of analysis tracked and communicated</i>	Analysts are also using text to briefly annotate their code with insights from individual logical steps. The mechanisms for forming a narrative in these notebooks are code, code output, code comments, text headers and a few text annotations. The text annotations or code output serve to provide an explanation of the results observed from each method [wang, rule]. The reader has an easier time of understanding the findings of the work but, not how this fits into the methodological narrative [rule-reproduce].	https://github.com/nmchaves/tissue-specific-expression/blob/cad372622f66213fba0b3e4296d1fa47f0af6955/GO_prediction/analyze_1_tissue_results_pca.ipynb
0.8	<i>Some motivations and insights of analysis tracked and communicated</i>	Analysts are explaining their motivations behind individual analytical steps, and more thoroughly annotating their logical steps with insights. The analyst is more thoroughly explaining their motivations, methods, and their insights using text [rule-reproduce, wang]. Though these elements might not perfectly match the narrative formed with code and code output, they do a sufficient job in communicating findings for an audience with prior knowledge [rule, chattopadhyay].	https://github.com/Azhag/Bayesian-visual-working-memory/blob/5a0a8c990b3398944a898f9dfe5824c0de0203c5/Experiments/dua_lrecall_fitmixturemodel/model_fitting_interactive_try.ipynb
0.9	<i>Motivations and insights of analysis being clearly communicated</i>	Analysts introduce their analytical reasoning behind the work overall. In addition, they are illustrating links between logical steps using text in the form of headers, insights and motivations. Together the text forms a narrative of the methods and the results. Text, code and code output are well interwoven to form parallel narratives of the methods, motivations, and the results of the analysis (the complete workflow) [kery-story, rule-reproduce]. However, a lack of accessibility to the dataset, an explanation of things like specialized APIs, or field specific terminology makes the notebook less accessible to a wider audience [rule, chattopadhyay]. In general, the implications of the work are not explained [rule, rule-reproduce, wang].	https://github.com/jakecarlson1/machine-learning-labs/blob/00bce5151b21473dc3525f90e0eddf76d67cf64a/Lab8/Lab%20%20-%20Recurrent%20Neural%20Networks.ipynb
1	<i>Analysis workflow being communicated to a wide audience</i>	Text may provide instructions on how to interact with the notebook, provide context behind the work, motivations on the methodology, insights from individual logical steps	https://github.com/RhysZH/GA-DAT9-Projects1/blob/98b57a24d4a42d38f70d

	and insights from the entire exercise [rule-reproduce]. If code and code output are present, they align with the narrative being outlined by the text. These notebooks are obviously created to reach a wide audience [kery-story, rule, wang]. Text, code and code output are well interwoven to form parallel narratives of the methods, and the results of the analysis (the complete analytical workflow) [rule-reproduce, rule, wang, kery-story]. They contain detailed explanations of things like specialized APIs, the dataset, and field specific terminology. They may also contain concluding text regarding the implications of the work [wang, rule-reproduce].	4ddba5ad23d7f9f9a512/R_Morgan_Project4_Files/R_Morgan_Project4_Presentation.ipynb
--	---	---