Creativity and Depth in Open-Ended Projects

Oct 20, 2019

Abstract

Programming is an essential component of a growing number of introductory statistics courses. Many introductory statistics courses use R to put concepts introduced in the course into practice, and these courses often serve as many students' first introduction to R and programming. While some faculty choose to teach R using base R syntax, others introduce R using packages from the tidyverse. Although some have strong opinions on how to start with R, the literature on evidence-based comparative studies on learning R with base R compared to the tidyverse syntax is lacking. We analyzed 205 final projects in an introductory statistics course taught between the 2013-2016 academic years, evaluating each project on creativity, depth, and multivariate visualizations. We found that students introduced to R with packages from the tidyverse scored higher, on average, on these metrics. Based on these findings, we created resources designed for future use in introductory statistics instruction.

Keywords: Introductory statistics, Education, Tidyverse, Base R, Creativity, Depth, Multivariate visualizations

IRB Approval number: 2019-0130

Background

Statistics has been undergoing a rapid change from its past. Statistics has morphed from a more theoretical science to one that has birthed the practice of data science, which relies more and more on programming tools to apply statistical processes to often large quantities of data. In the age of big data, programming and analytical skills have become much more desired, and as a result, the data science field has grown dramatically, (going to insert something like this link: (https://www.forbes.com/sites/quora/2017/10/25/why-data-science-is-such-a-hot-career-right-now/#1c2e9d8e106b).

But when and how should students be taught how to develop these talents?

Luckily, the Guidelines for Assessment and Instruction in Statistics Education provide the answer to the first part of the question *when* by recommending that students should use programming, when accessible, to explore real-world examples of the concepts
covered in the classroom, even in introductory courses.

In total, the GAISE set quality and scope standards for introductory statistics education and provide a set of general recommendations and specific learning goals for such courses (Carver et al. 2016). The most recent GAISE report includes updates concerning the use of real datasets and technology, in the hopes of implementing more realistic data scenarios in the classroom.

This paper seeks to help answer the second part of the question above: *how*. Since introductory statistics focuses primarily on the usage of different data analytics tools, the class is most popularly taught in R (maybe this link, might want something more edu-based: https://stackoverflow.blog/2017/10/10/impressive-growth-r/). When using R, or R Studio, as a course supplement, there are two prevailing and competing techniques for beginners. Students are instructed to work with either a relatively new suite of packages called the tidyverse or base R commands that have been in use for far longer. Proponents of base R claim that it is easier to use in creating specific visualizations, and that because students will already be accustomed to writing larger code chunks when plotting, they will not be phased when needing to write extra lines to make the visualization suitable for release (Leek 2016). Conversely, tidyverse supporters believe that if the role of coding within an introductory statistics course is to both convince students of R's importance and to supplement their course material, the tidyverse syntax encourages students to discover insights for provided datasets as soon as they start, helping to complete both objectives (D Robinson, n.d.). They also cite instances where students created interesting visualizations with seemingly advanced attributes in the tidyverse syntax that they would not be able to reproduce in base R after just a few hours of instruction (David Robinson 2014).

The tidyverse has been fashioned based on its primary author's experiences tackling real-world problems and as a teacher, and are a suite of packages to analyze and manipulate datasets within a tidy framework. A tidy dataset lists each variable as a separate column and every observation as its own row and is required as an input for many popular base R functions, such as `lm()`, the command for generating linear regressions. Within each package, a few commands can be generalized to pertain to many tasks, often making it simple to grasp

(Wickham and others 2014). As the tidyverse has become more integrated into the data science community, additional packages, such as mosaic, have been created to help beginning coders develop advanced insights alongside the tidyverse (Pruim, Kaplan, and Horton 2017).

One of the main recommendations of the GAISE report is to give students practice with developing and using statistical thinking, and the report suggests open-ended projects with multiple variables as a way to help students appreciate the role of statistics in everyday life. In this paper, we analyzed student work in an open-ended final project in the primary introductory statistics course offered at the university: STA 101 - Data Analysis and Statistical Inference.

STA 101 is the primary introductory statistics course at Duke University, where students are assumed to have entered the class without a statistical background. The majority of students enrolled in the course do not plan on matriculating in future statistics curricula, thus focusing the course on applications of statistics, as students learn how to relate STA 101's concepts to their future works. This course introduces students to the discipline of statistics as a science of understanding and analyzing data. Throughout the semester, students learn how to effectively make use of data in the face of uncertainty: how to collect data, how to analyze data, and how to use data to make inferences and conclusions about real world phenomena.

The course is flipped and team-based. Students are placed in teams at the beginning of the semester based on data collected from them on an informational survey as well as a pre-test that helps evaluate their background and previous exposure to statistics. In these teams, students complete readiness assessments (one per each of the seven learning units), weekly computational labs, in-class application exercises, and an end of semester project, which is the focus of this study. The team-based pedagogy aims to encourage students to continuously debate and discuss ideas, thus bolster or modifying their current knowledge to better comprehend the topics introduced (Brame and Director 2016). The students stay in the same teams throughout the semester to build up their team dynamics in anticipation of a higher-stakes final project at the end of the semester.

A crucial component of the course is the weekly computational labs. The objective of these labs is to give students hands on experience with data analysis using modern statistical software. The labs also provide the students with the tools they will need to complete the analysis successfully. The course introduces R as the statistical programming language, and students interface with R via the integrated development environment RStudio. For a large majority of the students in the class, this is their first exposure to programming. The introduction to this challenging aspect of the course is eased by removing the need for software installation and setup and by using tools like R Markdown that allow students to effortlessly generate reproducible data analysis reports (Çetinkaya-Rundel and Rundel 2018).

At the university, the programming aspect of STA 101 classes has been taught in either base R or the tidyverse syntax, creating an optimal platform to diagnose the direct effects of the syntax. Through the analysis of R code from STA 101 final group projects, we hope to uncover if one programming syntax encourages a more advanced level of creativity while simultaneously adhering to the GAISE recommendations and guidelines.

Despite the rising popularity of the tidyverse in teaching beginning statistical programming,

just one study has attempted to determine the differences in effectiveness in employing the tidyverse syntax and base R for introductory R users. The authors analyzed the difference in visualization quality between base R plots and `ggplot2`, a package within the tidyverse, for beginning R programmers. The study concluded that the visualizations generated using `ggplot()` compared to base R's `plot()` were often easier to understand when creating a complex multivariate visualization. Additionally, students were found to be more likely to uncover insights involving more complex relationships when using `ggplot2`, where students could use `facet_grid()` instead of needing the two `for()` loops when employing the most concise base R method (Myint et al. 2019). Although the evaluation favored visualizations crafted using `ggplot2`, it focused on the output of the code rather than the code quality. This study concerns the code itself, and whether STA 101 students at the university from 2013-2016 were encouraged to be more creative and produce higher-quality final projects based on the syntax, and then provides educational resources catering to these results. After making conclusions based on the observational study, we implemented these changes by building upon the resources for a library in R and updating student labs. Instead of attempting a form of statistical modelling, we decided to create educational materials because of our desire to contribute to the current introductory statistics curriculum.

Data

The dataset was manually compiled from 205 STA 101 final group projects spanning the 2013-2016 academic years. 13 data entries were discarded since the complete project submissions could not be recovered. The projects' contents are located on the university's Sakai sites of seven different classes, six of which were taught by the advisor of this project. However, the professor does not cover much of the R learning over the course of the semester, since it occurs within the labs, which are coordinated by teaching assistants. Although there may be some slight instructor effects on coding abilities, the teaching assistants are the primary R educators, and they experience a lot of turnover each semester. Therefore, some sections may have stronger grasps of coding for the final project, but they are likely to be minimal since teaching assistants are apprised of the upcoming week's content via weekly meetings with the professor. Regardless, some discrepancies may be due to teaching assistants and are noted in the Results section.

The dataset constructed in this paper primarily focuses on actions taken by the student groups within the univariate and bivariate portions of the final projects. In doing so, the dataset contains variables summarizing relative creativity measures, as well as the projects' depth and level of multivariate visualizations. Variable explanations will be available in the Methods chapter.

Final Project

The computing labs are designed primarily to prepare students for the open-ended data analysis project at the end of the semester. In this project, all student groups are given a

single dataset with many variables, and are asked to explore and model the data and make conclusion using statistical inference methods.

Due to the flexibility promoted by the project assignment, student groups are encouraged to display aspects of creativity in their analyses, whether it is focusing on Warner Bros. Entertainment Inc. movies, or assembling an indicator variable tracking if a member of the film received a nomination for best actor or actress. Here, creativity often stems from the exploratory data analysis process, where groups can uncover interesting aspects to further scrutinize before beginning the analysis portion of the project.

The final project simulates a complete data analysis using relevant techniques covered throughout the STA 101 curriculum. Student groups complete a hypothetical task in an R script or R Markdown document where they work at a movie studio and their boss assigns two goals: the boss wants to learn about the attributes that make a movie popular and something new about movies.

The final assignment contains five components—an introduction, a univariate analysis, a bivariate analysis, a multiple regression for predicting audience scores, and a conclusion. The regression section is relatively constant from a code standpoint amongst groups in determining an optimal regression. However, in the univariate and bivariate analyses, which are generally utilized for exploratory data analysis (EDA), student groups have the flexibility to explore a variety of facets of the data, often separating final projects from one another.

As a check for potential biases, we also analyzed the final project assignment documents for each STA 101 sections examined in this study. Overall, there were no significant changes in the STA 101 final project assignment document over the course of the 2013-2016 academic years. However, there were small alterations that may have influenced the project submissions. For final projects completed in 2014 onward, student groups were placed in a situation at Paramount Pictures at a hypothetical job, whereas for Fall 2013 students, they were assigned the task at hand without the additional setting. Fall 2013 student groups were also not tasked with a prediction component, thus rendering a covariate tracked in this study inconsequential.

Perhaps most importantly, though, the Fall 2013 final project assignment document features a difference in the grading section, where the professor details the factors important in grading the final projects. In the Fall 2013 semester, one section is titled "Creativity and Critical Thought," compared to the document for more recent semesters, where the section is titled, "Critical Thought". Although it is a minute wording discrepancy, the stating of the direct importance of creativity for project grades may have provided an extra incentive for Fall 2013 project groups when conducting their EDAs.

The final project dataset has remained nearly the same for each class, and it tracks a random sample of American movies released since 1970. The dataset contains between 25 and 32 variables summarizing the movies' general characteristics such as run time, genre, and production studio, award trackers such as best picture, best actor, best actress and best director indicator variables, as well as data from an online film review website (Rotten Tomatoes) and an online movie database (IMDB). Although variables such as the producing studio, the month and day of the week of both the theater and DVD releases, IMDB rating (out of 10) and audience rating on Rotten Tomatoes were not included in the original 2013

dataset, student groups were provided with a sufficient amount of potential variables to analyze. The final project dataset's most recent codebook is available in the Appendix as part of the STA 101 Spring 2016 final project assignment. The next section will describe how the final projects were coded.

Methods

Since 98 percent of the projects were submitted either as an R script or R Markdown document, each final project submission was directly analyzed on the downloaded document for each group. The projects were examined and scored for 16 variables as either a 0, if the attribute was missing, or 1, if it was present. The remaining three covariates identified the student projects by grade, index, and class.

For the scoring mechanism, initially, we inputted the grades received for the assignment prior to coding the indicator variables. However, after the first two classes were coded, we recognized that the grades might provide a potential source of error, so the grades were subsequently inputted for all future projects after the entire submission was scored. In an attempt to fully expel any bias, we reconfirmed the variable coding while ignoring the grades received, and complex scoring situations were confirmed via collaboration between Mr. Feder and Dr. Çetinkaya-Rundel.

The indices corresponding to each project were created due to privacy concerns, as the student group and the names of each student were removed from the submission document. For the dataset utilized in this thesis, projects are solely identified by their assigned index. A separate local dataset serves as a link between the final projects and their indices.

The variables and the first few observations for each variable compiled in this project are available below.

```
#> Observations: 193
#> Variables: 22
#> $ index              <chr> "1", "2", "3", "4", "5", "7", "8", "9", "10...
#> $ grade              <dbl> 87.1, 89.2, 80.2, 87.2, 80.4, 90.2, 83.0, 8...
#> $ sem                <chr> "Fall 2014", "Fall 2014", "Fall 2014", "Fal...
#> $ r_rmd              <chr> ".r", ".r", ".r", ".r", ".r", ".r", ".r", "...
#> $ tidyverse          <chr> "base R", "base R", "base R", "base R", "ba...
#> $ create_new_var     <chr> "no", "yes", "no", "yes", "no", "no", "yes"...
#> $ change_var         <chr> "no", "no", "no", "no", "no", "no", "no", "...
#> $ sub_analysis       <chr> "yes", "yes", "yes", "yes", "yes", "no", "y...
#> $ sub_data           <chr> "no", "yes", "no", "yes", "no", "yes", "no"...
#> $ viz_mult_make      <chr> "no", "no", "no", "no", "no", "no", "no", "...
#> $ viz_mult_interpret <chr> "no", "no", "no", "no", "no", "no", "no", "...
#> $ eda_theme          <chr> "yes", "yes", "yes", "no", "no", "no", "yes...
#> $ rel_data           <chr> "no", "yes", "no", "yes", "no", "yes", "no"...
#> $ slr_fit            <chr> "yes", "yes", "yes", "no", "no", "no", "no"...
```

```
#> $ mlr_fit           <chr> "yes", "yes", "yes", "yes", "yes", "yes", "...
#> $ mlr_check_cond    <chr> "yes", "yes", "no", "yes", "yes", "yes", "y...
#> $ prediction        <chr> "yes", "yes", "yes", "yes", "yes", "yes", "...
#> $ ht                <chr> "yes", "yes", "no", "yes", "yes", "yes", "y...
#> $ ht_check_cond     <chr> "yes", "yes", "no", "no", "no", "no", "no",...
#> $ creative          <dbl> 1, 3, 1, 3, 1, 1, 2, 1, 2, 1, 1, 1, 2, 1, 1...
#> $ theme             <dbl> 1, 2, 1, 1, 0, 1, 1, 1, 2, 2, 1, 2, 0, 0, 2...
#> $ multiviz          <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0...
```

The final project code was not compiled to confirm that the code ran as it was designed, since it was an arduous task to determine the package version that the student groups utilized, as some commands are now defunct or have different arguments in the corresponding current version. Because of this decision, this analysis operates under the assumption that the code produced the desired results in each project and did not require further debugging, unless there was a clear typo in the code.

However, the contents of the student project code were analyzed for clarity, as well as creativeness, depth, and multivariate visualization effectiveness using combinations of the 16 indicator variables. Examples will be provided for code that scored a one for distinct covariates, but not before we provide an in-depth description of the creativity, depth and multivariate visualization effectiveness metrics.

1. `index`: Project ID
2. `grade`: Score on final project
3. `sem`: Semester course taken
4. `r_rmd`: Was the submission an R script (with prose of the project turned in as a Word document) or was the submission an R Markdown file?
5. `tidyverse`: Project used `"tidyverse"` or `"base R"` syntax
6. `create_new_var`: Students created a new variable based on existing variables, `"yes"` or `"no"`
7. `change_var`: Students changed existing variables, `"yes"` or `"no"`
8. `sub_analysis`: Students performed a subgroup analysis, `"yes"` or `"no"`
9. `sub_data`: Students used data subsets for the entire project, `"yes"` or `"no"`
10. `viz_mult_make`: Students employed visualizations with at least three variables, `"yes"` or `"no"`
11. `viz_mult_interpret`: Students properly interpreted their 3+ variable visualization, `"yes"` or `"no"`
12. `eda_theme`: Students used a consistent theme throughout their project, `"yes"` or `"no"`
13. `rel_data`: Students supplemented their project theme with relevant data, `"yes"` or `"no"`
14. `slr_fit`: Students fitted a simple linear regression, `"yes"` or `"no"`
15. `mlr_fit`: Students fitted a multiple linear regression, `"yes"` or `"no"`
16. `mlr_check_cond`: Students properly checked the conditions for their multiple linear regression, `"yes"` or `"no"`
17. `prediction`: Students used their multiple linear regression to predict a movie's audience score, `"yes"` or `"no"`

18. `ht`: Students performed a hypothesis test, `"yes"` or `"no"`
19. `ht_check_cond`: Students correctly checked the conditions for their hypothesis test, `"yes"` or `"no"`

Creativity

The creativity metric encapsulated anything the student groups coded that was not directly specified in the instructions but still contributed to their projects. The metric's possible scores ranged from 0 to 4, as a project was scored with a single point for each of the following:

1. Creation of new variable(s) based on existing variables
2. Transformation of existing variable(s)
3. Existence of a subgroup analysis
4. The use of a subset of the dataset for all steps of the project

For projects using the tidyverse, groups were still given scores of 1s if they satisfied these conditions in base R. While rare, two groups in labs taught in the tidyverse created or transformed covariates using base R, which was likely due to alternative resources, such as Stack Overflow, that prioritized base R solutions. Now, though, as the tidyverse's popularity continues to grow, more online resources are incorporating and promoting solutions in the tidyverse syntax.

Creation of New Variable(s)

The creation of new variable(s) is defined as any data manipulation throughout the EDA process where student groups compose a previously non-existing covariate. As an example, one group created a new variable tracking if a movie had won any of the following awards: best picture, best actor, best actress, or best director, and coded it as "yes" or "no". In order to score a 1, the student group also had to utilize the new variable within an aspect of their analysis, which eliminated groups that created unnecessary covariates. However, a score of 1 would be valid if the group did not use the variable in the inference or regression sections, but explored the covariate in their EDA.

Transformation of Existing Variable(s)

Although related to the above covariate, the transformation of existing variable(s) did not qualify as creating new variable(s), and vice versa. In this situation, a student group would score a 1 if they mutated a variable already existing within the dataset, oftentimes to highlight certain cases. For instance, some project groups changed `mpaa_rating` to either "R" or "Other," if the movie was not rated R. Similar to the requirements for scoring a 1 for the creation of new variable(s) covariate, the mutation was required to be employed usefully, as groups would have to provide at least a cursory analysis of the newly-transformed variable to score a 1.

A distinction between scoring a 1 for this covariate and 1 for subsetting the dataset or conducting a subgroup analysis is that filtering the dataset for entries that cover a portion of levels within a specific variable would qualify as a part of either a subgroup analysis or data subset, but not this covariate. Also, converting a variable to a factor that could be potentially read in as one when loading the dataset did not qualify as a mutation of an existing variable.

Existence of Subgroup Analysis

Projects that received a one analyzed portions of the data during their EDA process. Student groups could use an assortment of commands to satisfy a score of a 1, such as the creation of a box plot, five-number summary of a specific variable, or a subset with a corresponding numerical or graphical analysis. As an example, a project receiving a 1 for this category could have analyzed how the audience ratings for R rated movies compared to that of PG-13 movies in their bivariate analysis portion of the assignment.

Use of a Data Subset for Project's Entirety

Although the use of a data subset for the project's entirety covariate may seem similar to the one above, this variable received a 1 for a different aspect of the final group projects. Here, student groups are not just using the provided movies dataset for their EDA, inference, and regression—they are intentionally focusing on a few characteristics of the movie dataset. Student projects were not required to employ the same subsetted data throughout the entire analysis, but they did have to analyze related aspects of the movies dataset to qualify for a 1. For example, one student group scrutinized solely PG-13 rated movies for their final project, while another used the PG-13 rated movies subset for the EDA, PG-13 movies released after 2000 for their inference portion, and then the same PG-13 rated movies subset utilized in the EDA process to complete their regression analysis.

Depth

The depth metric measured the scope of the analysis, both in terms of the statistical methods utilized and storytelling. Since the GAISE advises instructors to focus on students' comprehension of important basic concepts rather than covering a multitude of topics in less detail, the depth metric qualified the student groups' understanding of the subjects addressed in the course. The metric ranged from 0-2 and was scored with 1 point for each of the following:

1. Presence of consistent theme throughout the project
2. Use of relevant data

Consistent Theme

In the world of data science, storytelling is such an important aspect, just as storytelling is designed to be for the STA 101 project. An impressive final project requires a story: a leading question, initial findings, subsequent analyses, and conclusion, all formed around a specific theme. Although this covariate's scoring was subjective, the requirements for final projects to score a 1 were similar clarity-wise to those defining the creativity metric. To receive a 1, student groups linked the steps in their project, often opting to focus on a few aspects within the entire movie dataset. For instance, analyzing the impact of movie ratings on audience scores would qualify as a sufficient theme, but merely inspecting an assortment of different predictors with minimal reasoning would not register as 1.

Presence of Relevant Data

Another subjective variable, the presence of relevant data was formed to complement the consistent theme covariate. To receive a 1 for this variable, student groups were required to sufficiently use R to create insights surrounding their chosen theme(s). The covariate addressed the issue that a project may have an interesting theme but lacks the analysis and coding quality to supplement it. For example, if a group was analyzing how comedy movie run time related to its audience scores but did not univariably plot the distribution of comedy run times, it would not score a 1. If the majority of the code could be employed to support the final project, the project group received a 1.

Multivariate Visualization Effectiveness

The multivariate visualization effectiveness metric accounted for both the presence and the insights derived from visualizations with at least three variables. Especially when using a movies dataset with many binary variables, visualizations with at least three variables and their subsequent interpretations could springboard important discoveries. Also, the GAISE highlights the significance of teaching students how to interpret multivariate visualizations. "When students leave an introductory course, they will likely encounter situations within their own fields of study in which multiple variables relate to one another in intricate ways. We should prepare our students for challenging questions that require investigating and exploring relationships among more than two variables (Carver et al. 2016)." The metric ranged from 0-2 and was scored with 1 point for each of the following:

1. Presence of a visualization with 3+ variables
2. Interpretation of the multivariate visualization

Although the two variables that constitute the multivariate visualization effectiveness metric are related, as a project could not score a 1 for the interpretation if it did not contain a multivariate visualization, the presence of the visualization did not imply a useful interpretation.

Presence of a Visualization with 3+ Variables

The presence of a visualization with at least three variables is an objective variable simple to determine when dissecting final project submissions. In creating these plots, student groups nearly always utilized colors to display the third variable along with two numerical ones on the x- and y-axes. To receive a score of a 1, projects were required to produce a graphical output with at least three aesthetics. For instance, some student groups created scatter plots between the critic and audience scores, with different colors corresponding to movies that won best picture at the Academy Awards that year.

Interpretation of Multivariate Visualization

The interpretation of a multivariate visualization is not a completely objective variable. A project containing an incorrect or insufficient interpretation of its multivariate visualization would not receive a score of a 1. By sufficient, the student group did not need to address every aspect of the visualization, but was required to discuss a key finding. Otherwise, the student group would receive a score of a 0. Next, we will discuss the results, as well as the specific covariates comprising each of these three metrics.

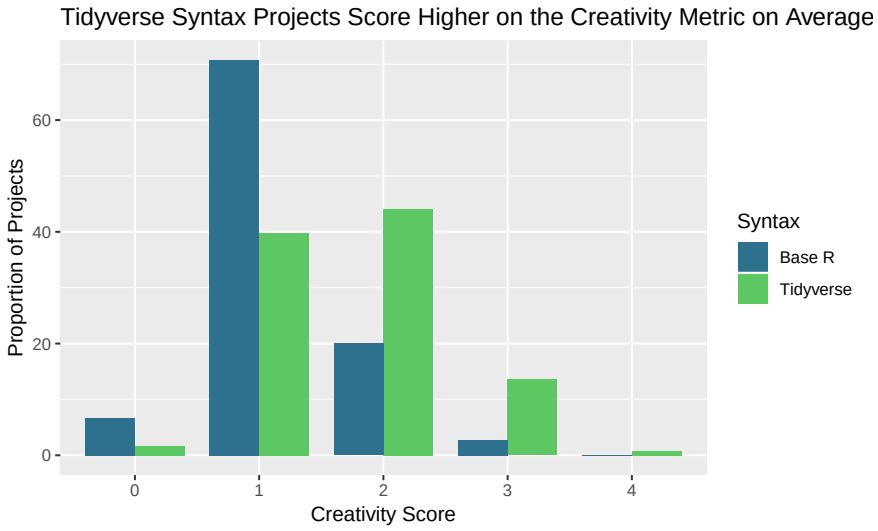
Results

Both numerically and graphically, the student projects using the tidyverse syntax scored higher on all three of the developed metrics. Tidyverse projects were much more prevalent in the upper levels of the three variables, and their means and standard deviations reflected these findings.

Creativity Metric

Despite there being only 75 base R projects recorded to the 118 tidyverse projects, more base R projects scored a 0 or 1 on the creativity metric than those using the tidyverse syntax. Overall, there was a single project that scored a perfect 4 out of the 193 projects, and the majority of the projects scored a 1 or 2 in the creativity metric.

However, within the tidyverse projects, more than half (58.5 percent) registered at least a 2 on creativity, compared to just 24.4 percent of base R projects.



The average creativity scores and corresponding standard deviations for projects utilizing the two syntaxes are available below as well:

Syntax	Mean	Standard Deviation
Base R	1.2	0.6
Tidyverse	1.7	0.8

When dissected by semester, the creativity score distributions did not notably vary for sections taught in base R. However, the two sections—both of which employed the tidyverse syntax—from the Spring 2016 semester fared significantly better judged by their score breakdown than the other tidyverse courses. Without those two semesters of tidyverse projects, the tidyverse projects still boasted a higher score distribution than those of base R. Still, the discrepancy for these two sections may be attributed to advanced instruction from the teaching assistants that encouraged groups to satisfy the requirements for higher creativity scores. The following subsections will compare base R and tidyverse student projects for each of the four variables combined to form the creativity metric, as well as propose potential reasons for the outcomes.

Creation of New Variable(s)

Out of the four variables that form the creativity metric, the starkest difference between projects employing the two syntaxes was within the creation of new variable(s) covariate. Half of all final projects using the tidyverse syntax featured a creation of a new variable, whereas less than a quarter of base R projects did.

Syntax	Creation of New Variable(s)	N	Percentage
base R	yes	18	24.0%
tidyverse	yes	59	50.0%

This difference may be due to the ease in utilizing the tidyverse's `mutate()` function, which allows users to create new variables using a single command. In base R, students are instructed to use `$` notation, which may be harder to grasp due to issues with `$` and its usage in other base R tasks such as variable selection.

Transformation of Existing Variables

While more than 75 project groups created new variables, far less opted to mutate existing ones in the provided dataset. Because the counts for both base R and tidyverse projects that satisfied this variable were lower, the resulting proportions were also much smaller. Regardless, there were more projects using the tidyverse syntax that mutated existing variables.

Syntax	Transformation of Existing Variables	N	Percentage
base R	yes	2	2.7%
tidyverse	yes	9	7.6%

The difference in proportions was not nearly as significant as it was for the creation of new variables and can be attributed to two factors. First, with the option to create their own variables, groups may not have found transforming existing variables as appealing. Second, the mechanism to do so does not differ dramatically between the two syntaxes. For comparison, an example of changing an existing studio variable to "Warner Bros. Studios" or "Other" is listed in both base R and the tidyverse syntax:

Base R:

```
movies$studio <- ifelse(movies$studio == "Warner Bros. Pictures",
                        "Warner Bros. Pictures", "Other")
```

Tidyverse:

```
movies <- movies %>%
  mutate(studio = (if_else(studio == "Warner Bros. Pictures",
                           "Warner Bros. Pictures", "Other")))
```

Existence of Subgroup Analysis

Out of all the covariates forming the creativity metric, the subgroup analysis was the most popular one satisfied for both base R and tidyverse projects. 97.5 percent of tidyverse projects performed a type of subgroup analysis, while more than 86 of base R projects did.

Syntax	Subgroup Analysis	N	Percentage
base R	yes	65	86.7%
tidyverse	yes	115	97.5%

The popularity of this aspect within group projects may be explained by the copious ways student groups could perform a subgroup analysis. Subgroup analyses may potentially be easier to perform in the tidyverse due to the presence of the `group_by()` command, which is often one of the most popular functions for beginning R programmers using the tidyverse syntax. In contrast, base R's `by()` function is not nearly as widely used.

Use of a Data Subset for Project's Entirety

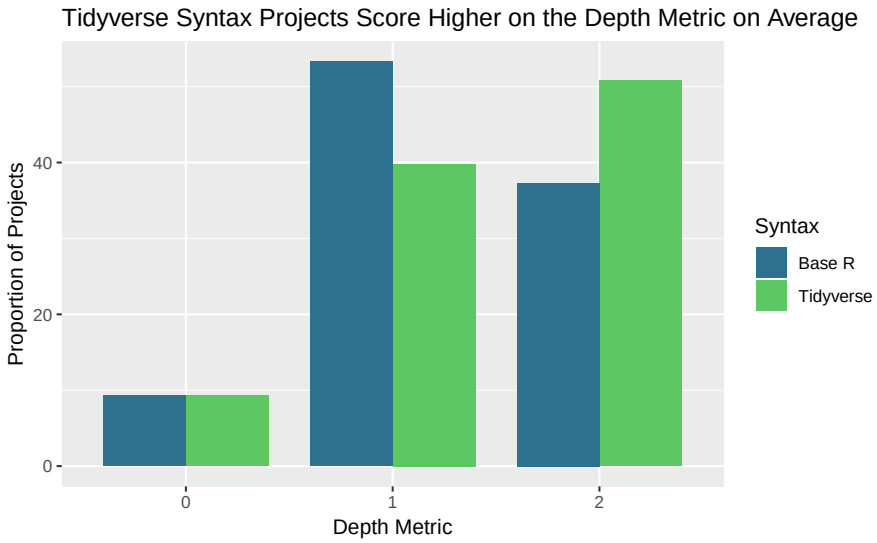
The use of a subset of the provided dataset for the entire final project was also significantly more popular amongst tidyverse projects, as more than 16 percent of student projects using the tidyverse satisfied a score of a one for this covariate, compared to less than 6 percent of all base R projects.

Syntax	Data Subset Usage	N	Percentage
base R	yes	4	5.3%
tidyverse	yes	20	16.9%

Since there is little difference between the tidyverse's `filter()` and base R's `subset()` functions, this discrepancy might not have a direct explanation. Perhaps, though, student groups using the tidyverse syntax may have been more encouraged to utilize data subsets throughout their projects than those using base R because other creativity aspects were performed more frequently in tidyverse projects, so those groups may have uncovered additional insights that led them to use data subsets that base R project groups did not utilize.

Depth Metric

Although the depth metric scores between final projects using the two syntaxes do not have the same discrepancy level as the one for the creativity metric, there is still a considerable difference between the depth metric distributions. 50.8 percent of the projects using the tidyverse syntax scored a perfect 2 in the depth metric compared to 37.3 percent of base R projects, as most base R projects scored a 0 or 1 in the depth metric.



The average depth scores and corresponding standard deviations for projects utilizing the two syntaxes are available below as well:

Syntax	Mean	Standard Deviation
Base R	1.3	0.6
Tidyverse	1.4	0.7

The results of the depth metric may be attributed to the prevalence of command chains in the tidyverse syntax that may make the project easier to code for beginners in R.

Upon inspection by class section, the depth score distributions were largely similar, sans one Spring 2016 section. Similar to how student groups from this section performed on the creativity metric, they scored much higher on average in the depth category than projects from the other classes. Still, though, like we saw when analyzing the creativity metric results, the tidyverse projects still boasted a higher score distribution than those of base R. The following subsections will contain a comparison of base R and tidyverse student projects for the two variables that were combined to form the depth metric.

Consistent Theme

Within the depth metric, the variable tracking the presence of a consistent theme showcased a larger difference in proportions between base R and tidyverse projects. Compared to base R projects, where 62.7 percent boasted a consistent theme, 74.6 percent of all tidyverse projects maintained uniformity theme-wise.

Syntax	Consistent Theme	N	Percentage
base R	yes	47	62.7%
tidyverse	yes	88	74.6%

The difference in proportions may potentially be attributed to the prevalence and popularity of `select()` and `filter()` within the tidyverse syntax, which allowed groups to choose specific columns within a data that satisfy a certain criteria, compared to base R, where student groups utilized square brackets to delineate those same criteria, which is not covered as extensively in beginning R programming.

Presence of Relevant Data

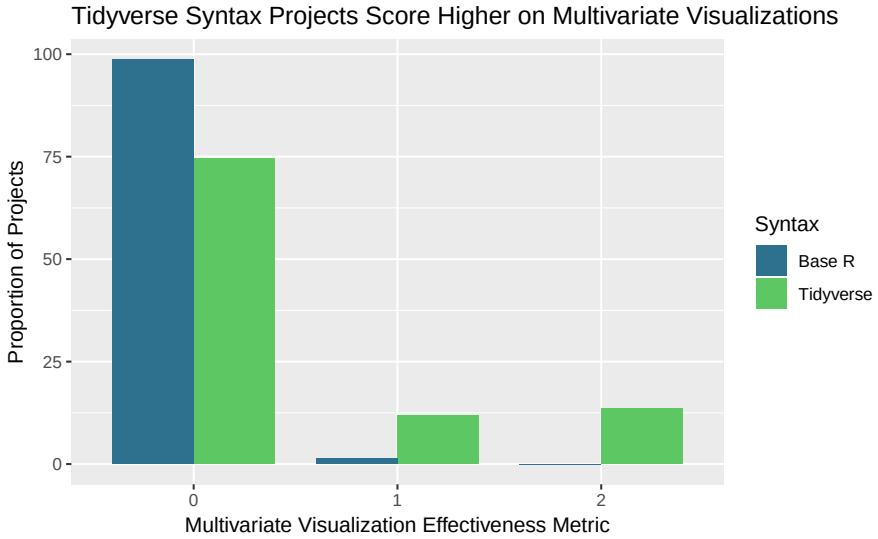
Proportions for the presence of relevant data covariate were very similar across projects using the two syntaxes, with just a 1.6 percent difference in percentages.

Syntax	Relevant Data Presence	N	Percentage
base R	yes	49	65.3%
tidyverse	yes	79	66.9%

The presence of relevant supporting data should not be greatly impacted by the particular coding syntax student groups employed, which may explain the small difference in percentages between the two syntaxes for this variable.

Multivariate Visualization Effectiveness Metric

Between projects using the two different syntaxes, the most significant difference in the three metrics occurred within the multivariate visualization effectiveness metric, where just one student group using base R created a plot containing at least three different variables. The majority of all projects did not include multivariate visualizations, but projects using the tidyverse syntax were more popular within higher scores of the metric, with 13.6 percent scoring a perfect 2. No projects in base R reached the same score.



The average multivariate visualization scores and corresponding standard deviations for projects utilizing the two syntaxes are available below as well. Although these statistics do not fully represent the score difference, there is still a distinct divide.

Syntax	Mean	Standard Deviation
Base R	0.0	0.1
Tidyverse	0.4	0.7

Although we observed a higher rate and score distribution of tidyverse projects, the trend is not consistent by semester. One of the Fall 2015 sections included 10 projects with a multivariate visualization score of at least a one, while the other did not contain a single project showcasing a multivariate plot. Similar to the reasoning described in the creativity metric section, this may be due to the emphasis and quality of instruction from the teaching assistants (which differ by class section) in regards to creating plots with at least three variables. The following subsections will contain a comparison of base R and tidyverse student groups for the two variables that were combined to form the multivariate visualization metric.

Presence of a Visualization with 3+ Variables

Only one project coded in base R displayed a visualization with at least three variables, compared to 30 tidyverse projects. Percentage-wise, too, the tidyverse projects were more likely to contain a multivariate visualization.

Syntax	Multivariate Visualization Presence	N	Percentage
base R	yes	1	1.3%
tidyverse	yes	30	25.4%

These results may be due to the differences in base R's `plot()` function relative to the tidyverse's `ggplot()`. Whereas `plot()` requires extra commands to add graphical aesthetics beyond x- and y-variables, the `aes()` command embedded within `ggplot()` provides a simple platform for tidyverse users to employ additional aesthetics. For example, code is listed below that would satisfy a score of one when using the two syntaxes.

Base R:

```
movies.color <- rep("pink", length(movies$best_pic_nom))
movies.color[movies$best_pic_nom == "yes"] <- "blue"
plot(audience_score ~ critics_score, col = movies.color, data = movies)
legend(col = c("pink", "blue"))
```

Tidyverse:

```
ggplot(data = movies, aes(x = critics_score, y = audience_score,
                          color = best_pic_nom)) +
    geom_point()
```

Interpretation of Multivariate Visualization

No base R student group interpreted a multivariate plot, whereas 13.6 percent of groups utilizing the tidyverse syntax did. Although the percentage satisfying a score of a 1 is not very high, more than half of tidyverse projects that formed a visualization with at least three variables (16 out of 30) contained sufficient interpretations.

Syntax	Multivariate Visualization Interpretation	N	Percentage
tidyverse	yes	16	13.6%

Theoretically, the difference in proportions between projects using base R compared to the tidyverse syntax should not drastically differ. However, since only one base R project contained a multivariate plot, there was bound to be a discrepancy. In general, though, the plots should not deviate for interpretation, though some users may find more comfort interpreting complex plots generated in `ggplot2` (Myint et al. 2019). Based on the results of the three metrics, we created educational resources designed to teach the tidyverse syntax as a supplement for introductory statistics courses.

Educational Resources

After analyzing the retrospective study, we provided two resources intended for use in statistical instruction to reflect our findings in favor of the tidyverse syntax. We altered current STA 101 labs to further adhere to the GAISE manual and provided two code samples with explanations for `infer`, an R package created to perform inference tasks using a tidy framework.

Infer Vignettes

Currently, the tidyverse contains packages to clean, mutate, model and visualize data, but not to perform inference tasks. Due to the popularity and demand for the tidyverse and tidy framework, a group of professors created `infer`, which is fashioned as the tidyverse solution to inference. Like packages such as `dplyr` and `ggplot2`, `infer` relies on a few commands to execute a variety of tasks and benefits from the piping structure used throughout the tidyverse syntax.

Unfortunately, `infer` does not presently include detailed examples of how to use the package. Therefore, we wrote two vignettes—longer code samples embedded within data analysis

stories—to provide `infer` users with vivid examples and explanations for the package's primary commands.

By inference, the two primary endeavors are performing hypothesis tests and creating confidence intervals—hence a vignette for each. The vignettes utilize separate aspects of the same dataset to provide users the comfort of a consistent data source while exposing them to tasks using both numerical and categorical variables.

The complete vignettes will be released with the next version of `infer`. The link to the package is available in the Appendix, and `infer` was also introduced into the second educational resource, which is described next.

Lab Enhancements

Based on the results gleaned from analyzing the student project data, we decided to fully apply the tidyverse syntax and methods to the STA 101 labs. OpenIntro Statistics is an open-source textbook approved for use at the undergraduate level by the American Institute of Mathematics, and the labs accompanying OpenIntro textbooks are available online and are used by students at many institutions, including at the university. These labs were updated in 2016 to incorporate tidyverse syntax for data visualization and wrangling, though statistical inference still relied on base R syntax. As part of this project, we updated the code introduced in the labs to fully leverage the tidyverse ecosystem.

Since the most recent update of the labs to incorporate tidyverse syntax, recommendations around introducing data visualization with `ggplot2` have changed. Previous practice used the `qplot()` (quick plot) function, which has a simpler API than the `ggplot()` function, but is more cumbersome to produce complex multivariate visualizations with. With this update, we have completely abandoned the use of `qplot()` and replaced it with `ggplot()`, resulting in changes in associated code. Two main reasons for this change are the plethora of resources for debugging `ggplot()` as well as ease in expansion to complex visualizations.

Another major update was in the labs that focus on statistical inference. Previous versions of the labs used a custom function called `inference()` from the `oilabs` package, the R package used to supplement OpenIntro Statistics. This function, designed with the best intentions in mind for highlighting the unified nature of statistical inference across various hypothesis tests and confidence intervals introduced in introductory statistics curricula, over time morphed into a function too extensive for efficient debugging and too customized for use beyond the introductory statistics classroom. The infer package, released in 2018, was heavily inspired by the `inference()` function, but significantly improved the API for tidy statistical inference and tied it closely to how both we and the GAISE recommend introducing statistical inference in introductory statistics curricula. The goal of this package "is to perform inference using an expressive statistical grammar that coheres with the tidy design framework," as per its R documentation.

For instance, here is code for generating a two-sided hypothesis test using the two methods, with the `inference()` function listed first.

```
inference(y = y_variable, x = x_variable, data = data, statistic = "mean",
          type = "ht", null = 0, alternative = "twosided",
          method = "theoretical")
```

```
obs_diff <- data %>%
  specify(dependent ~ independent) %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
null <- data %>%
  specify(dependent ~ independent) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("yes", "no"))
null %>%
  get_p_value(obs_stat = obs_diff, direction = "two_sided")
```

And here is an example of code creating a 95 percent confidence interval using `inference()` and `infer`, respectively.

```
inference(y = response, data = data, statistic = "proportion", type = "ci",
          method = "theoretical", success = "yes")
```

```
data %>%
  specify(formula = text_ind ~ NULL, success = "yes") %>%
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "prop") %>%
  get_ci(level = .95)
```

The labs focusing on statistical inference have thus been updated with the current infer syntax in order to provide students with a workflow they can extend outside of their classroom setting alongside the tidyverse, as well as available online resources.

A few of the labs also featured datasets that had either been out of use for the past few years and will continue to be considered outmoded, were convoluted and required supplementary information, or intentionally or unintentionally promoted gender stereotypes. These datasets, and the corresponding labs, were updated with more robust and interesting material, as summarized in the table below. The datasets were analyzed prior to insertion to affirm that they would have the bandwidth to seamlessly fit the particular focus of the labs.

Previous dataset	Reason for Change	New dataset
Body Dimensions	Focuses on weight comparison across binary genders	Fast Food Nutritional Facts
Atheism and Religion	Unverified dataset	Youth Risk Behavior Surveillance System
Baseball	Required further dataset explanation	Human Freedom Index

Previous dataset	Reason for Change	New dataset
North Carolina Births	Dated and Not interesting to most undergrads	Youth Risk Behavior Surveillance System

Conclusion

Given the scope of the GAISE, whose key components in regards to statistical computing were proxied by creativity, depth, and multivariate visualization scores, we recommend teaching the tidyverse syntax as the primary syntax when using R as a supplement for introductory statistics courses. The tidyverse's success may be attributed to its consistent framework and commands throughout the syntax, potentially helping students with limited programming experience create insightful aspects of their final projects. In accordance with this hypothesis, we recommend teaching inference in R using `infer`, as it was designed specifically to fit the general tidyverse mold. As a result, we provided updated versions for the OpenIntro lab examples to align with the most recent updates within the tidyverse and `infer`, as well as further tune them to fit the GAISE. `infer` users will also have additional reference materials with novel vignettes to introduce programmers to the library's syntax.

Although the sample in the observational study is limited by solely analyzing final projects, the standardization of STA 101 classes provides an optimal baseline for comparison between the two coding syntaxes. Concerns about another potential source of bias, the project assignment document, can be assayed by the lack of wording differences since the Fall 2013 document (a base R class) and a similar creativity metric distribution for the Fall 2013 class as other base R sections. As such, since there were higher score distributions for the three metrics for projects using the tidyverse syntax, the assignment document language is not a confounding variable in this analysis.

Since we could not identify other relevant biases, we can attribute the distinctions in creativity, depth, and multivariate visualization effectiveness scores to the R syntax. Additionally, because the final projects were randomly scored by course, and then later verified, we are confident that there was minimal bias in the coding of the indicator variables. Thus, we strongly encourage instructors to teach introductory R using the tidyverse syntax,as well as `infer` for inference tasks, as we believe that the tidyverse's consistency encourages students to produce outputs in closer adherence to the GAISE.

Future Work

We encourage others to build upon this analysis in a randomized controlled experimental setting since this study is solely retrospective. Although we have attempted to eliminate all potential sources of bias, a randomized trial would remove ambiguity in a causal analysis. However, we acknowledge that there are limitations in performing a randomized experiment in a classroom setting due to the nature of student collaboration across sections and logistical

issues. Perhaps similar to the study performed by (Myint et al. 2019), a randomized experiment could be conducted through an online educational company.

There are additional aspects of the two syntaxes that we did not focus upon in the retrospective study. For beginning programmers, the readability differences between the two methods are essential to learning. Therefore, we welcome experiments dissecting the readability of the two syntaxes for individuals with varying coding backgrounds.

We believe it may be important to conduct follow up studies on the same individuals in regards to three goals mentioned in the GAISE: tracking these students' motivation to learn R while in the introductory statistics class, subsequent performance in more advanced R or other programming topics, and tracking the retention rates within the statistics and data science fields. Another study could work in supplement by having the students in this project interpret their code after a certain amount of elapsed time to determine if it can still be understood.

Finally, our work suggests the `infer` package as a natural extension to learning tidyverse syntax. However, there are no studies currently in the literature that specifically state the effectiveness of the syntax of this package in understanding and using correctly statistical inference techniques, and thus, we encourage others to conduct such a study.

Appendix

2016 Spring Semester STA 101 Final Project Assignment Document

Listed below is the final project assignment document, which includes a codebook for the movies dataset, given to students who enrolled in the course in the Spring 2016 iteration:

You and your teammates work for Paramount Pictures.

Your boss has just acquired data about how much audiences and critics like movies as well as numerous other variables about the movies.

She is interested in learning what attributes make a movie popular. She is also interested in learning something new about movies. She wants your team to figure it all out.

As part of this project you will complete exploratory data analysis (EDA), inference, modeling, and prediction. You have been introduced to some of these concepts already, and you will learn about the others later in the course.

The data can be loaded directly in R Studio.

Data

The dataset is comprised of 651 randomly sampled movies produced and released before 2016.

Some of these variables are only there for informational purposes and do not make any sense to include in a statistical analysis. It is up to you to decide which variables are meaningful and which should be omitted. For example information in the the `actor1` through `actor5` variables was used to determine whether the movie casts an actor or actress who won a best actor or actress Oscar.

You might also choose to omit certain observations or restructure some of the variables to make them suitable for answering your research questions.

When you are fitting a model you should also be careful about collinearity, as some of these variables may be dependent on each other.

Codebook

1. `title`: Title of movie
2. `title_type`: Type of movie (Documentary, Feature Film, TV Movie)
3. `genre`: Genre of movie (Action & Adventure, Comedy, Documentary, Drama, 1. Horror, Mystery & Suspense, Other)
4. `runtime`: Runtime of movie (in minutes)

5. `mpaa_rating`: MPAA rating of the movie (G, PG, PG-13, R, Unrated)
6. `studio`: Studio that produced the movie
7. `thtr_rel_year`: Year the movie is released in theaters
8. `thtr_rel_month`: Month the movie is released in theaters
9. `thtr_rel_day`: Day of the month the movie is released in theaters
10. `dvd_rel_year`: Year the movie is released on DVD
11. `dvd_rel_month`: Month the movie is released on DVD
12. `dvd_rel_day`: Day of the month the movie is released on DVD
13. `imdb_rating`: Rating on IMDB
14. `imdb_num_votes`: Number of votes on IMDB
15. `critics_rating`: Categorical variable for critics rating on Rotten Tomatoes 1. (Certified Fresh, Fresh, Rotten)
16. `critics_score`: Critics score on Rotten Tomatoes
17. `audience_rating`: Categorical variable for audience rating on Rotten Tomatoes 1. (Spilled, Upright)
18. `audience_score`: Audience score on Rotten Tomatoes (response variable)
19. `best_pic_nom`: Whether or not the movie was nominated for a best picture 1. Oscar (no, yes)
20. `best_pic_win`: Whether or not the movie won a best picture Oscar (no, yes)
21. `best_actor_win`: Whether or not one of the main actors in the movie ever won an Oscar (no, yes) – note that this is not necessarily whether the actor won an Oscar for their role in the given movie
22. `best_actress win`: Whether or not one of the main actresses in the movie ever won an Oscar (no, yes) – not that this is not necessarily whether the actresses won an Oscar for their role in the given movie
23. `best_dir_win`: Whether or not the director of the movie ever won an Oscar (no, yes) – not that this is not necessarily whether the director won an Oscar for the given movie
24. `top200_box`: Whether or not the movie is in the Top 200 Box Office list on BoxOfficeMojo (no, yes)
25. `director`: Director of the movie
26. `actor1`: First main actor/actress in the abridged cast of the movie
27. `actor2`: Second main actor/actress in the abridged cast of the movie
28. `actor3`: Third main actor/actress in the abridged cast of the movie
29. `actor4`: Fourth main actor/actress in the abridged cast of the movie
30. `actor5`: Fifth main actor/actress in the abridged cast of the movie
31. `imdb_url`: Link to IMDB page for the movie
32. `rt_url`: Link to Rotten Tomatoes page for the movie

Stages of the project

You will complete this project in two stages:

1. Stage 1: Proposal (25 points)

2. Stage 2: Poster and presentation (75 points)

The remainder of this document outlines the requirements and expectations for both stages of the project. You should read the entire document before getting started. The requirements and expectations for Stage 1 will only make sense in context of those for Stage 2.

Stage 1: Proposal (25 points)

Content

Your proposal should contain the following:

1. **Data**: (2 points) Describe how the observations in the sample are collected, and the implications of this data collection method on the scope of inference (generalizability / causality).
2. **Research questions**: (6 points) Come up with at least three research questions that you want to answer using these data. You should phrase your research questions in a way that matches up with the scope of inference your dataset allows for. Make sure that at least two of these questions involve at least three variables. You are welcomed to create new variables based on existing ones. Note that you will have the option to update / revise / change these questions for your poster at the end of the semester.
3. **EDA**: (9 points) Perform exploratory data analysis that adresses each of the three research questions you outlined above. Your EDA should contain numerical summaries and visualizations. Each R output and plot should be accompanied by a brief interpretation.
4. **Timeline**: (4 points) Sketch out a timeline for the work you will do to complete this project. Be as detailed and precise as possible. And be realistic – discuss course schedules, travel plans, etc.
5. **Teamwork**: (4 points) Describe in detail how you will divvy up the work between team members and what aspects of the project you will complete together as a team. Note that during the poster session each member needs to be able to answer questions about all aspects of the work, regardless of whether they took the lead on that section or not.

Format & length

Your proposal should be written using the R Markdown template, so that all R code, output, and plots will be automatically included in your write up.

Download the template for the proposal.

Your proposal should not exceed 5 pages (view a print preview to determined length).

Grading

Your proposal will be graded out of 25 points (as outlined above), and will make up 25% of your overall project score.

The following will result in deductions:

- Late: -1 points for each day late
- Reproducibility issues, requiring to make changes to the R Markdown file to knit the document: -3 points
- Each page over limit: -2 points per page (view print preview to confirm length)

Stage 2: Poster and presentation (75 points)

Content

1. **Introduction:** Outline your main research question(s).
2. **EDA:** Do some exploratory data analysis to tell an "interesting" story about movies. Instead of limiting yourself to relationships between just two variables, broaden the scope of your analysis and employ creative approaches that evaluate relationships between two variables while controlling for another.
3. **Inference:** Use one of your research questions (or come up with a new one depending on feedback from the proposal) that can be answered with a hypothesis test or a confidence interval, e.g. "Is there a difference in mean audience scores between genres?" or "What is the average difference in audience scores between movies that do and do not feature without oscar winner actors?" This question could be used to shed some light on your choice of the "best" linear model.
   Carry out the appropriate inference task to answer your question.
4. **Modeling:** Develop a multiple linear regression model to predict a numerical variable in the dataset.
5. **Prediction:** Pick a movie from 2015 (a new movie that is not in the sample) and do a prediction for this movie using your the model you developed (and the `predict` function in R). Also quantify the uncertainty around this prediction using an appropriate interval.
6. **Conclusion:** A brief summary of your findings from the previous sections **without** repeating your statements from earlier as well as a discussion of what you have learned about the data and your research question(s). You should also discuss any shortcomings of your current study (either due to data collection or methodology) and include ideas for possible future research.

Poster format & length

Poster: We suggest using a tri-fold poster. You can organize it however you like. You do not need to get your poster professionally printed. You can see sample posters from previous years at your professor's office.

R Markdown: All code used to generate the statistics and plots on your poster should be organized and submitted in an R Markdown document.

Download the template for the project.

There is no length limit for this document.

Presentation format & length

You will give a four minute presentation of your work. Each team member must speak during this presentation. The time limit is firm, you will be asked to stop at the end of four minutes. This is not a lot of time, therefore you must decide carefully what you will highlight during your presentation and practice to make sure you can fit everything you want to say in the time limit.

Grading

Your poster (and accompanying code) and presentation will be graded out of 75 points, and will make up 75% of your overall project score.

Grading of the project will take into account:

- Correctness: Are the procedures and explanations correct?
- Presentation: What was the quality of the presentation and poster?
- Content/Critical thought: Did your think carefully about the problem?
- Tidyness: Is your code organized well?

Your team scores will be based on the following components:

- 25 points - poster
- 20 points - presentation
- 20 points - code
- 10 points - classmates' evaluation

Submission

Online on Sakai under Assignments. These will be time stamped, and late penalty will be applied based on the time stamp. Only one submission per team required.

1. R Markdown file (.Rmd)
2. HTML output (.html)

We will download your R Markdown file and run your code to confirm reproducibility of your work. Grading will be based on the document we compile, so make sure that your R Markdown file contains everything necessary to compile your entire work.

Teamwork and grading

Team scores for both the proposal and the poster will be adjusted based on team peer evaluation data to determine each student's indivudual grade. You will be asked to fill out a survey where you rate the contribution of each team member. Filling out the survey is a prerequisite for receiving a project score.

All team members must be present at the poster session. Failure to do so will result in a 0 on the project for the absent team member.

Note that each student must complete the project and score at least 30% of total possible points on the project in order to pass this class.

Honor code

You may not discuss this project in any way with anyone outside your team, besides the professor and TAs. Failure to abide by this policy will result in a 0 for all teams involved.

Tips

This project is an opportunity to apply what you have learned about descriptive statistics, graphical methods, correlation and regression, and hypothesis testing and confidence intervals.

The goal is not to do an exhaustive data analysis i.e., do not calculate every statistic and procedure you have learned for every variable, but rather to show that you are proficient at using R at a basic level and that you are proficient at interpreting and presenting the results.

You might consider critiquing your own method, such as issues pertaining to the reliability of the data and the appropriateness of the statistical analysis you used within the context of this specific dataset.

Updated OpenIntro Labs

The Github repository containing the updated OpenIntro labs can be accessed using this link:

https://github.com/openIntrostat/oilabs-tidy

Infer Vignettes

The two new `infer` vignettes will be released with the next update to `infer`, whose website can be accessed here:

http://infer.netlify.com

References

Brame, Cynthia J, and CFT Assistant Director. 2016. "Team-Based Learning." *Vanderbilt Center for Teaching. Retrieved from Http://Cft. Vanderbilt. Edu/Guides-Sub-Pages/Team-Base d-Learning*.

Carver, Robert, Michelle Everson, John Gabrosek, Nicholas Horton, Robin Lock, Megan Mocko, Allan Rossman, et al. 2016. "Guidelines for Assessment and Instruction in Statistics Education (Gaise) College Report 2016." AMSTAT.

Çetinkaya-Rundel, Mine, and Colin Rundel. 2018. "Infrastructure and Tools for Teaching Computing Throughout the Statistical Curriculum." *The American Statistician* 72 (1). Taylor & Francis: 58–65.

Leek, Jeffrey. 2016. "Why I Don't Use Ggplot2." https://simplystatistics.org/2016/02/11/why-i-dont-use-ggplot2/.

Myint, Leslie, Aboozar Hadavand, Leah Jager, and Jeffrey Leek. 2019. "Comparison of Plotting System Outputs in Beginner Analysts." *arXiv Preprint arXiv:1903.01829*.

Pruim, Randall, Daniel T Kaplan, and Nicholas J Horton. 2017. "The Mosaic Package: Helping Students to 'Think with Data' Using R." *The R Journal* 9 (1): 77–102.

Robinson, D. n.d. "Teach the Tidyverse to Beginners (July 5, 2017)."

Robinson, David. 2014. "Don't Teach Built-in Plotting to Beginners (Teach Ggplot2)." http://varianceexplained.org/r/teach_ggplot2_to_beginners/.

Wickham, Hadley, and others. 2014. "Tidy Data." *Journal of Statistical Software* 59 (10). Foundation for Open Access Statistics: 1–23.