

STA 380, Part 2: Exercises

Due: by end of the working day (5:00 PM US Central time) on Monday, August 14.

Prepare your report on the problems below using RMarkdown or Jupyter (or similar, e.g. Quarto) so that they are fully reproducible, carefully integrating visual and numerical evidence with prose. You may work solo, or in groups of 4 or fewer people. You can self-organize groups via Canvas.

Note: the option to submit as a group is intended to give you an incentive to get to know some of your classmates. The idea is for y'all to work together on *all* the problems and to learn from each other, not to divide up the individual problems.

Submit via Canvas under the "Assignments" tab. **You should submit a link to your GitHub repo where the final report has been stored.** You are also welcome to use R or Python in any combination (e.g. all in one language, or maybe R for some problems and Jupyter/Python for others.) No matter how you do it, please include a `.README.md` file in your GitHub repo linking clearly to your solutions. We shouldn't have to guess where your answers are based on the link you send.

If you are submitting as a group, make sure you include everyone's names in this `.README.md` file.

Notes: - Do not knit to `.html`, which won't render properly on GitHub.

- Do not directly e-mail the instructors/TA directly with your reports. We will ignore any e-mailed submissions.
- For any mathematical expressions in your report, you can use LaTeX syntax, which I encourage you to learn anyway. Alternatively, you can just handwrite the math, snap a photo, and include the image in the final report. This is a simple, low-overhead option.
- Submissions that are late, but received within 24 hours, will receive a 20% penalty. Submissions more than 24 hours late will not be considered and will receive a grade of zero.
- You are welcome to use large language models like ChatGPT to help you code.

Grading criteria:

- Did you make an honest, concerted attempt at the problem?
- Did you address all parts of the question?
- Did you include enough detail on what you actually did so that a well-informed reader could understand your analysis in detail? (You won't receive full credit if it's not clear what steps you actually took in your analysis.)
- Did you include properly annotated figures/tables where appropriate?
- Did you write up your solution professionally, with an actual narrative flow (good), or did you just copy and paste a bunch of code without much in the way of explanation (bad)? Submissions with lots of "raw," unadorned code and little in the way of explanation will receive very poor grades.
- Did you use sensible analysis techniques to answer a given question?
- Did you make any significant technical mistakes?

Probability practice

Part A. Visitors to your website are asked to answer a single survey question before they get access to the content on the page. Among all of the users, there are two categories: Random Clicker (RC), and Truthful Clicker (TC). There are two possible answers to the survey: yes and no. Random clickers would click either one with equal probability. You are also giving the information that the expected fraction of random clickers is 0.3. After a trial period, you get the following survey results: 65% said Yes and 35% said No. What fraction of people who are truthful clickers answered yes? Hint: use the rule of total probability.

Part B. Imagine a medical test for a disease with the following two attributes:

- The sensitivity is about 0.993. That is, if someone has the disease, there is a probability of 0.993 that they will test positive.
- The specificity is about 0.9999. This means that if someone doesn't have the disease, there is probability of 0.9999 that they will test negative.
- In the general population, incidence of the disease is reasonably rare: about 0.0025% of all people have it (or 0.000025 as a decimal probability).

Suppose someone tests positive. What is the probability that they have the disease?

Wrangling the Billboard Top 100

Consider the data in `billboard.csv` (`../data/billboard.csv`) containing every song to appear on the weekly Billboard Top 100 (<https://www.billboard.com/charts/hot-100/>) chart since 1958, up through the middle of 2021. Each row of this data corresponds to a single song in a single week. For our purposes, the relevant columns here are:

- `performer`: who performed the song
- `song`: the title of the song
- `year`: year (1958 to 2021)
- `week`: chart week of that year (1, 2, etc)
- `week_position`: what position that song occupied that week on the Billboard top 100 chart.

Use your skills in data wrangling and plotting to answer the following three questions.

Part A: Make a table of the top 10 most popular songs since 1958, as measured by the *total number of weeks that a song spent on the Billboard Top 100*. Note that these data end in week 22 of 2021, so the most popular songs of 2021 will not have up-to-the-minute data; please send our apologies to The Weeknd.

Your table should have **10 rows** and **3 columns**: `performer`, `song`, and `count`, where `count` represents the number of weeks that song appeared in the Billboard Top 100. Make sure the entries are sorted in descending order of the `count` variable, so that the more popular songs appear at the top of the table. Give your table a short caption describing what is shown in the table.

(Note: you'll want to use both `performer` and `song` in any `group_by` operations, to account for the fact that multiple unique songs can share the same title.)

Part B: Is the “musical diversity” of the Billboard Top 100 changing over time? Let's find out. We'll measure the musical diversity of given year as *the number of unique songs that appeared in the Billboard Top 100 that year*. Make a line graph that plots this measure of musical diversity over the years. The x axis should show the year, while the y axis should show the number of unique songs appearing at any position on the Billboard Top 100 chart in any week that year. For this part, please filter the data set so that it excludes the years 1958 and 2021, since we do not have complete data on either of those years. Give the figure an informative caption in which you explain what is shown in the figure and comment on any interesting trends you see.

There are number of ways to accomplish the data wrangling here. For example, you could use two distinct sets of data-wrangling steps. The first set of steps would get you a table that counts the number of times that a given song appears on the Top 100 in a given year. The second set of steps operate on the result of the first set of steps; it would count the number of unique songs that appeared on the Top 100 in each year, *irrespective of how many times* it had appeared.

Part C: Let's define a "ten-week hit" as a single song that appeared on the Billboard Top 100 for at least ten weeks. There are 19 artists in U.S. musical history since 1958 who have had *at least 30 songs* that were "ten-week hits." Make a bar plot for these 19 artists, showing how many ten-week hits each one had in their musical career. Give the plot an informative caption in which you explain what is shown.

Notes:

1. You might find this easier to accomplish in two distinct sets of data wrangling steps.
2. Make sure that the individuals names of the artists are readable in your plot, and that they're not all jumbled together. If you find that your plot isn't readable with vertical bars, you can add a `coord_flip()` layer to your plot to make the bars (and labels) run horizontally instead.
3. By default a bar plot will order the artists in alphabetical order. This is acceptable to turn in. But if you'd like to order them according to some other variable, you can use the `fct_reorder` function, described in this blog post (<https://datavizpyr.com/re-ordering-bars-in-barplot-in-r/>). This is optional.

Visual story telling part 1: green buildings

The case

Over the past decade, both investors and the general public have paid increasingly close attention to the benefits of environmentally conscious buildings. There are both ethical and economic forces at work here. In commercial real estate, issues of eco-friendliness are intimately tied up with ordinary decisions about how to allocate capital. In this context, the decision to invest in eco-friendly buildings could pay off in at least four ways.

1. Every building has the obvious list of recurring costs: water, climate control, lighting, waste disposal, and so forth. Almost by definition, these costs are lower in green buildings.
2. Green buildings are often associated with better indoor environments—the kind that are full of sunlight, natural materials, and various other humane touches. Such environments, in turn, might result in higher employee productivity and lower absenteeism, and might therefore be more coveted by potential tenants. The financial impact of this factor, however, is rather hard to quantify *ex ante*; you cannot simply ask an engineer in the same way that you could ask a question such as, "How much are these solar panels likely to save on the power bill?"
3. Green buildings make for good PR. They send a signal about social responsibility and ecological awareness, and might therefore command a premium from potential tenants who want their customers to associate them with these values. It is widely believed that a good corporate image may enable a firm to charge premium prices, to hire better talent, and to attract socially conscious investors.
4. Finally, sustainable buildings might have longer economically valuable lives. For one thing, they are expected to last longer, in a direct physical sense. (One of the core concepts of the green-building movement is "life-cycle analysis," which accounts for the high front-end environmental impact of acquiring materials and constructing a new building in the first place.) Moreover, green buildings may also be less susceptible to market risk—in particular, the risk that energy prices will spike, driving away tenants into the arms of bolder, greener investors.

Of course, much of this is mere conjecture. At the end of the day, tenants may or may not be willing to pay a premium for rental space in green buildings. We can only find out by carefully examining data on the commercial real-estate market.

The file `greenbuildings.csv` (`../data/greenbuildings.csv`) contains data on 7,894 commercial rental properties from across the United States. Of these, 685 properties have been awarded either LEED or EnergyStar certification as a green building. You can easily find out more about these rating systems on the web, e.g. at www.usgbc.org. The

basic idea is that a commercial property can receive a green certification if its energy efficiency, carbon footprint, site selection, and building materials meet certain environmental benchmarks, as certified by outside engineers.

A group of real estate economists constructed the data in the following way. Of the 1,360 green-certified buildings listed as of December 2007 on the LEED or EnergyStar websites, current information about building characteristics and monthly rents were available for 685 of them. In order to provide a control population, each of these 685 buildings was matched to a cluster of nearby commercial buildings in the CoStar database. Each small cluster contains one green-certified building, and all non-rated buildings within a quarter-mile radius of the certified building. On average, each of the 685 clusters contains roughly 12 buildings, for a total of 7,894 data points.

The columns of the data set are coded as follows:

- CS.PropertyID: the building's unique identifier in the CoStar database.
- cluster: an identifier for the building cluster, with each cluster containing one green-certified building and at least one other non-green-certified building within a quarter-mile radius of the cluster center.
- size: the total square footage of available rental space in the building.
- empl.gr: the year-on-year growth rate in employment in the building's geographic region.
- Rent: the rent charged to tenants in the building, in dollars per square foot per calendar year.
- leasing.rate: a measure of occupancy; the fraction of the building's available space currently under lease.
- stories: the height of the building in stories.
- age: the age of the building in years.
- renovated: whether the building has undergone substantial renovations during its lifetime.
- class.a, class.b: indicators for two classes of building quality (the third is Class C). These are relative classifications within a specific market. Class A buildings are generally the highest-quality properties in a given market. Class B buildings are a notch down, but still of reasonable quality. Class C buildings are the least desirable properties in a given market.
- green.rating: an indicator for whether the building is either LEED- or EnergyStar-certified.
- LEED, Energystar: indicators for the two specific kinds of green certifications.
- net: an indicator as to whether the rent is quoted on a "net contract" basis. Tenants with net-rental contracts pay their own utility costs, which are otherwise included in the quoted rental price.
- amenities: an indicator of whether at least one of the following amenities is available on-site: bank, convenience store, dry cleaner, restaurant, retail shops, fitness center.
- cd.total.07: number of cooling degree days in the building's region in 2007. A degree day is a measure of demand for energy; higher values mean greater demand. Cooling degree days are measured relative to a baseline outdoor temperature, below which a building needs no cooling.
- hd.total07: number of heating degree days in the building's region in 2007. Heating degree days are also measured relative to a baseline outdoor temperature, above which a building needs no heating.
- total.dd.07: the total number of degree days (either heating or cooling) in the building's region in 2007.
- Precipitation: annual precipitation in inches in the building's geographic region.
- Gas.Costs: a measure of how much natural gas costs in the building's geographic region.
- Electricity.Costs: a measure of how much electricity costs in the building's geographic region.
- cluster.rent: a measure of average rent per square-foot per calendar year in the building's local market.

The goal

An Austin real-estate developer is interested in the possible economic impact of "going green" in her latest project: a new 15-story mixed-use building on East Cesar Chavez, just across I-35 from downtown. Will investing in a green building be worth it, from an economic perspective? The baseline construction costs are \$100 million, with a 5% expected premium for green certification.

The developer has had someone on her staff, who's been described to her as a "total Excel guru from his undergrad statistics course," run some numbers on this data set and make a preliminary recommendation. Here's how this person described his process.

I began by cleaning the data a little bit. In particular, I noticed that a handful of the buildings in the data set had very low occupancy rates (less than 10% of available space occupied). I decided to remove these buildings from consideration, on the theory that these buildings might have something weird going on with them, and could potentially distort the analysis. Once I scrubbed these low-occupancy buildings from the data set, I looked at the green buildings and non-green buildings separately. The median market rent in the non-green buildings was \$25 per square foot per year, while the median market rent in the green buildings was \$27.60 per square foot per year: about \$2.60 more per square foot. (I used the median rather than the mean, because there were still some outliers in the data, and the median is a lot more robust to outliers.) Because our building would be 250,000 square feet, this would translate into an additional $\$250000 \times 2.6 = \650000 of extra revenue per year if we build the green building.

Our expected baseline construction costs are \$100 million, with a 5% expected premium for green certification. Thus we should expect to spend an extra \$5 million on the green building. Based on the extra revenue we would make, we would recuperate these costs in $\$5000000 / \$650000 = 7.7$ years. Even if our occupancy rate were only 90%, we would still recuperate the costs in a little over 8 years. Thus from year 9 onwards, we would be making an extra \$650,000 per year in profit. Since the building will be earning rents for 30 years or more, it seems like a good financial move to build the green building.

The developer listened to this recommendation, understood the analysis, and still felt unconvinced. She has therefore asked you to revisit the report, so that she can get a second opinion.

Do you agree with the conclusions of her on-staff stats guru? If so, point to evidence supporting his case. If not, explain specifically where and why the analysis goes wrong, and how it can be improved. Do you see the possibility of confounding variables for the relationship between rent and green status? If so, provide evidence for confounding, and see if you can also make a picture that visually shows how we might "adjust" for such a confounder. *Tell your story in pictures, with appropriate introductory and supporting text.*

Note: this is intended as an exercise in visual and numerical story-telling. Your approach should rely on pictures and/or tables, not a regression model. Tell a story understandable to a non-technical audience. Keep it concise.

Visual story telling part 2: Capital Metro data

The file `capmetro_UT.csv` (`../data/capmetro_UT.csv`) contains data from Austin's own Capital Metro bus network, including shuttles to, from, and around the UT campus. These data track ridership on buses in the UT area. Ridership is measured by an optical scanner that counts how many people embark and alight the bus at each stop. Each row in the data set corresponds to a 15-minute period between the hours of 6 AM and 10 PM, each and every day, from September through November 2018. The variables are:

- *timestamp*: the beginning of the 15-minute window for that row of data
- *boarding*: how many people got on board any Capital Metro bus on the UT campus in the specific 15 minute window
- *alighting*: how many people got off ("alit") any Capital Metro bus on the UT campus in the specific 15 minute window
- *day_of_week* and *weekend*: Monday, Tuesday, etc, as well as an indicator for whether it's a weekend.
- *temperature*: temperature at that time in degrees F
- *hour_of_day*: on 24-hour time, so 6 for 6 AM, 13 for 1 PM, 14 for 2 PM, etc.
- *month*: July through December

Your task is to create a figure, or set of related figures, that tell an interesting story about Capital Metro ridership patterns around the UT-Austin campus during the semester in question. Provide a clear annotation/caption for each figure, but the figure(s) should be more or less stand-alone, in that you shouldn't need many, many paragraphs to convey its meaning. Rather, the figure together with a concise caption should speak for itself as far as possible.

You have broad freedom to look at any variables you'd like here – try to find that sweet spot where you're showing genuinely interesting relationships among more than just two variables, but where the resulting figure or set of figures doesn't become overwhelming/confusing. (Faceting/panel plots might be especially useful here.)

Clustering and dimensionality reduction

The data in `wine.csv` (`../data/wine.csv`) contains information on 11 chemical properties of 6500 different bottles of *vinho verde* wine from northern Portugal. In addition, two other variables about each wine are recorded: - whether the wine is red or white
- the quality of the wine, as judged on a 1-10 scale by a panel of certified wine snobs.

Run PCA, tSNE, and any clustering algorithm of your choice on the 11 chemical properties (or suitable transformations thereof) and summarize your results. Which dimensionality reduction technique makes the most sense to you for this data? Convince yourself (and me) that your chosen approach is easily capable of distinguishing the reds from the whites, using only the "unsupervised" information contained in the data on chemical properties. Does your unsupervised technique also seem capable of distinguishing the higher from the lower quality wines? Present appropriate numerical and/or visual evidence to support your conclusions.

To clarify: I'm not asking you to run a supervised learning algorithms. Rather, I'm asking you to see whether the differences in the labels (red/white and quality score) emerge naturally from applying an unsupervised technique to the chemical properties. This should be straightforward to assess using plots.

Market segmentation

Consider the data in `social_marketing.csv` (`../data/social_marketing.csv`). This was data collected in the course of a market-research study using followers of the Twitter account of a large consumer brand that shall remain nameless—let's call it "NutrientH2O" just to have a label. The goal here was for NutrientH2O to understand its

social-media audience a little bit better, so that it could hone its messaging a little more sharply.

A bit of background on the data collection: the advertising firm who runs NutrientH2O's online-advertising campaigns took a sample of the brand's Twitter followers. They collected every Twitter post ("tweet") by each of those followers over a seven-day period in June 2014. Every post was examined by a human annotator contracted through Amazon's Mechanical Turk (<https://www.mturk.com/mturk/welcome>) service. Each tweet was categorized based on its content using a pre-specified scheme of 36 different categories, each representing a broad area of interest (e.g. politics, sports, family, etc.) Annotators were allowed to classify a post as belonging to more than one category. For example, a hypothetical post such as "I'm really excited to see grandpa go wreck shop in his geriatric soccer league this Sunday!" might be categorized as both "family" and "sports." You get the picture.

Each row of `social_marketing.csv` (`../data/social_marketing.csv`) represents one user, labeled by a random (anonymous, unique) 9-digit alphanumeric code. Each column represents an interest, which are labeled along the top of the data file. The entries are the number of posts by a given user that fell into the given category. Two interests of note here are "spam" (i.e. unsolicited advertising) and "adult" (posts that are pornographic, salacious, or explicitly sexual). There are a lot of spam and pornography "bots" on Twitter (<http://mashable.com/2013/11/08/twitter-spambots/>); while these have been filtered out of the data set to some extent, there will certainly be some that slip through. There's also an "uncategorized" label. Annotators were told to use this sparingly, but it's there to capture posts that don't fit at all into any of the listed interest categories. (A lot of annotators may used the "chatter" category for this as well.) Keep in mind as you examine the data that you cannot expect perfect annotations of all posts. Some annotators might have simply been asleep at the wheel some, or even all, of the time! Thus there is some inevitable error and noisiness in the annotation process.

Your task to is analyze this data as you see fit, and to prepare a concise report for NutrientH2O that identifies any interesting market segments that appear to stand out in their social-media audience. You have complete freedom in deciding how to pre-process the data and how to define "market segment." (Is it a group of correlated interests? A cluster? A latent factor? Etc.) Just use the data to come up with some interesting, well-supported insights about the audience, and be clear about what you did.

The Reuters corpus

Revisit the Reuters C50 text corpus that we briefly explored in class. Your task is simple: tell an interesting story, anchored in some analytical tools we have learned in this class, using this data. For example:

- you could cluster authors or documents and tell a story about what you find.
- you could look for common factors using PCA.
- you could train a predictive model and assess its accuracy, constructing features for each document that maximize performance.
- you could do anything else that strikes you as interesting with this data.

Describe clearly what question you are trying to answer, what models you are using, how you pre-processed the data, and so forth. Make sure you include at least *one* really interesting plot (although more than one might be necessary, depending on your question and approach.)

Format your write-up in the following sections, some of which might be quite short:

- Question: What question(s) are you trying to answer?
 - Approach: What approach/statistical tool did you use to answer the questions?
 - Results: What evidence/results did your approach provide to answer the questions? (E.g. any numbers, tables, figures as appropriate.)
 - Conclusion: What are your conclusions about your questions?
- Provide a written interpretation of your results, understandable to stakeholders who might plausibly take an interest in this data set.

Regarding the data itself: In the C50train directory, you have 50 articles from each of 50 different authors (one author per directory). Then in the C50test directory, you have another 50 articles from each of those same 50 authors (again, one author per directory). This train/test split is obviously intended for building predictive models, but to repeat, you need not do that on this problem. You can tell any story you want using any methods you want. Just make it compelling!

Note: if you try to build a predictive model, you will need to figure out a way to deal with words in the test set that you never saw in the training set. This is a nontrivial aspect of the modeling exercise. (E.g. you might simply ignore those new words.)

This question will be graded according to three criteria:

1. the overall “interesting-ness” of your question and analysis.
2. the clarity of your description. We will be asking ourselves: could your analysis be reproduced by a competent data scientist based on what you’ve said? (That’s good.) Or would that person have to wade into the code in order to understand what, precisely, you’ve done? (That’s bad.)
3. technical correctness (i.e. did you make any mistakes in execution or interpretation?)

Association rule mining

Revisit the notes on association rule mining and the R example on music playlists: `playlists.R` (`./R/playlists.R`) and `playlists.csv` (`./data/playlists.csv`). Then use the data on grocery purchases in `groceries.txt` (`./data/groceries.txt`) and find some interesting association rules for these shopping baskets. The data file is a list of shopping baskets: one person’s basket for each row, with multiple items per row separated by commas. Pick your own thresholds for lift and confidence; just be clear what these thresholds are and say why you picked them. Do your discovered item sets make sense? Present your discoveries in an interesting and visually appealing way.

Notes: - This is an exercise in visual and numerical story-telling. Do be clear in your description of what you’ve done, but keep the focus on the data, the figures, and the insights your analysis has drawn from the data, rather than technical details.

- The data file is a list of baskets: one row per basket, with multiple items per row separated by commas. You’ll have to cobble together your own code for processing this into the format expected by the “arules” package. This is not intrinsically all that hard, but it is the kind of data-wrangling wrinkle you’ll encounter frequently on real problems, where your software package expects data in one format and the data comes in a different format. Figuring out how to bridge that gap is part of the assignment, and so we won’t be giving tips on this front.

Image classification with neural networks

In this problem, you will train a neural network to classify satellite images. In the `data/EuroSAT_RGB` (https://github.com/jgscott/STA380/tree/master/data/EuroSAT_RGB) directory, you will find 11 subdirectories, each corresponding to a different class of land or land use: e.g. industrial, crops, rivers, forest, etc. Within each subdirectory, you will find examples in .jpg format of each type. (Thus the name of the directory in which the image lives is the class label.)

Your job is to set up a neural network that can classify the images as accurately as possible. Use an 80/20 train test split. Summarize your model and its accuracy in any way you see fit, but make you include *at a minimum* the following elements:

- overall test-set accuracy, measured however you think is appropriate
- show some of the example images from the test set, together with your model’s predicted classes.
- a confusion matrix showing the performance of the model on the set test, i.e. a table that cross-tabulates each test set example by (actual class, predicted class).

I strongly recommend the use of PyTorch in a Jupyter notebook for this problem; look into PyTorch's `ImageFolder` data set class, which will streamline things considerably.