Project 2

Project Objectives

This project involves creating predictive models and automating Markdown reports. You will create a blog post linking to your analyses. There is a second part to this project that will be done with a partner assigned by Dr. Post after the initial due date.

Project Work

All project work should be done in a github repo. Ideally, you have connected RStudio with github and can work from the command line within RStudio. All major updates should be made through github so we can track your activity.

- You will create a repo with analysis documents along with a blog post
- After the due date you will fork your partner's repo, make edits, and do a pull request (see below for details - this second part will be due on Thursday, October 22nd)

Other than when you are creating your repo and getting it linked up to RStudio, your repo should remain private until the day the project is due.

Repo

Create a repo on github for your project. On your project repo you should go into the settings and enable github pages (feel free to select a theme too!). This will make it so your repo can be accessed like your blog (username.github.io/repo-name).

When you knit your .Rmd file, use the output type `output: rmarkdown::github_document`. This will create a .md file which will automatically be rendered by github when used appropriately (you'll have seven of these in the end as you'll automate your analysis for each weekday).

In the README.md file for the repo, give a brief description of the purpose of the repo and create links to each sub-document (Monday's analysis, Tuesday's analysis, etc.). Links can be made to the sub-documents using relative paths. For instance, if you have all of the outputted .md files in the main directory you would just use markdown linking:

- The analysis for `[Monday is available here](MondayAnalysis.md)`.

Of course, this supports the use of folders as well if you output the files into separate folders.

- You should also make a note of all packages required to run your analysis here.
- You should include the code used to automate the process (i.e. the `render` function you used) here as well.

Blog

Once you've completed the above tasks you should write a brief blog post outlining your project and linking to the username.github.io/repo-name site. You should then also reflect on the process you went through for this project. Discuss things like:

- what would you do differently?
- what was the most difficult part for you?
- what are your big take-aways from this project?
- **In your blog post, provide a link to your github pages repo**

Topic

If you are in the partner A group (see the homework link for partner info), you'll read in and analyze an online news popularity data set.

If you are in the partner B group, you'll read in and analyze a bike sharing data set.

You can read more about the data sets at the websites.

Report

Recommendation: At first, consider just using the 'Monday' data. Once you have all of the below steps done for that data, then automate it to work with any chosen day of the week data.

- All code chunks should be shown unless they are setup code chunks.

Introduction section

You should have an introduction section that briefly describes the data and the variables you have to work with (no need to discuss all of them, just the ones you want to use). **If you are analyzing the bike share data, do not use the `casual` and `registered` variables to do any modeling!**

You should also mention the purpose of your analysis and the methods you'll use (no need to detail them here) for analysis.

Data

When reading in your data, you should use a relative path.

You should randomly sample from the (Monday) data in order to form a training (use 70% of the data) and test set (use 30% of the data). You should set the seed to make your work reproducible.

Summarizations

You should produce some basic (but meaningful) summary statistics and plots about the training data you are working with (especially as it relates to your response). The general things that the plots describe should be explained but, since we are going to automate things, there is no need to try and explain particular trends in the plots you see (unless you want to try and automate that too!).

Modeling

Once you have your training data set, we are ready to fit some models.

For the online news data set, the goal is to create models for predicting the `shares` variable. For the bike sharing data, the goal is to create models for predicting the `cnt` variable.

You will create two models initially (and one when you collaborate later):

- a (not ensemble) tree-based model chosen using leave one out cross validation

- a boosted tree model chosen using cross-validation

There should be text describing the type of model you are fitting, your fitting process, and the final chosen model (this last part is to be automated so I don't expect you to explicitly interpret that model, but you should be able to display something about the final model chosen on the training data).

The two models should be compared on the test set with a brief discussion.

Automation

Once you've completed the above for Monday, adapt the code so that you can use a parameter in your build process. You should be able to automatically generate an analysis report for each `weekday_is_*` or `weekday` variable (depending on your data set). You'll end up with seven total outputted documents.

All of this above is due on the original due date for the project.

Submission

In the project submission, you should simply put a link to your blog post (which will have a link to your github pages, which will have a link to your github repo).

Second portion of project

The second part is a brief collaboration (hopefully this won't take more than an hour or two). You'll be notified of your partner on Saturday after the due date.

You should fork your partner's repo, add in the brief changes below, rerun the analysis, and submit a pull request. This pull request must be done by Thursday, October 22nd. The owner of the repo should then accept the pull request sometime on Friday if possible.

Secondary analysis

Add in a linear regression model of your choice (no need to do variable selection, just pick a model) to the modeling section. Add in the predictions of the model on the test set.

Rubric for Grading (total = 100 points)

Item	Points	Notes
Introduction	10	Worth either 0, 5, or 10
Data split	5	Worth either 0 or 5
Summarizations & discussions	15	Worth either 0, 5, 10, or 15
Modeling, selection, & discussion	30	Worth either 0, 5, . . . , 30
Test set prediction	5	Worth either 0 or 5
Automation	15	Worth either 0, 5, . . . , 15
Blog post and repo setup	10	Worth either 0, 5, or 10
Secondary analysis	10	Worth either 0, 5, or 10

Notes on grading:

- For each item in the rubric, your grade will be lowered one level for each each error (syntax, logical, or other) in the code and for each required item that is missing or lacking a description.

- **If your work was not completed and documented using your github repo you will lose 50 points on the project.**

- You should use Good Programming Practices when coding (see wolfware). If you do not follow GPP you can lose up to 25 points on the project.

- You should use appropriate markdown options/formatting (you can lose up to 20 points) for not doing so