Project 2

Project Objectives

This is a group project (groups of two, see the project 2 assignment page) that involves creating predictive models and automating Markdown reports. Once you've completed the project you will also create a blog post linking to your analyses.

Project Work

The first step is for the first group member to create a github repo and add the second group member as a collaborator. The second group member then needs to accept the membership. This gives everyone access to push changes up to the repository. All project work should be done within this repo so we can track your activity.

Each time you go to work on the project, you should pull down any of the latest changes using `git pull`. You should then upload any changes you've made via the usual workflow done previously. There may occasionally be merge conflicts that have to be dealt with. This can be done with the `Git` tab in RStudio. Let us know if you are having issues with conflicts that you can't resolve!

Repo Setting

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). Be sure to choose the master or main branch as the one to use if you have choices there.

In the README.md file for the repo (which doesn't need to be created from a .Rmd file this time, just use the one you initialize into the repo), give a brief description of the purpose of the repo. You should also state all the packages required to run the analyses you do.

You'll be automating the creation of seven documents (one for each day of the week). Each should be rendered as a `github_document` from a single .Rmd file. In the README.md file you should create links to each of the seven documents you will create (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)`.

In the README.md you should also make a note of all packages required to run your analysis and include (in code text) the code used to automate the process (i.e. the `render` function you used).

So in the end your README.md file should have a brief description of the purpose of the repo, a list of R packages used, links to the generated analyses, and the code used to create the analyses from a single .Rmd file.

Blog

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

- 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 site as well as the github.com repo site**

Topic

Ok, so that is the set up the final product(s). What are you actually doing? You'll read in and analyze the day.csv bike sharing data set. You should read more about the data set at the website. We'll summarize the data and then try to predict the number of users using predictive models.

Report

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

- 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). Your target variables will of course be the `casual` and/or `registered` variables in some way (perhaps the sum of them, that is up to you).

You should also mention the purpose of your analysis and the methods you'll use to model the response. You'll describe those in more detail later.

This section should be done by the 'second' group member.

Data

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

You'll 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.

This section should be done by whoever can get to it first.

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).

As you will automate this same analysis across other data, you can't describe the trends you see in the graph (unless you want to try to automate that!). Instead, you should describe the purpose of each summary statistic/plot and what the reader may be able to determine from it. For instance, if you create a scatterplot with the number of casual users on the y-axis and the month on the x-axis, you might say something like 'We can inspect the trend of users across months using this plot. There may be a seasonal effect present.'

Each group member is responsible for producing some summary statistics (means, sds, contingency tables, etc.) and for producing at least three graphs of the data.

Modeling

Once you have your training data set and have explored the data a bit, we are ready to fit some models.

The goal is to create models for predicting the number of users in some way. Each group member should contribute a linear regression model and an ensemble tree model. The first group member should fit a random forest model and the second group member should fit a boosted tree model. Both models should be chosen using cross-validation.

Prior to the models fit using linear regression, the first group member should provide a short but thorough explanation of the idea of a linear regression model.

Prior to each ensemble model, you should provide a short but reasonably thorough explanation of the ensemble model you are using (so one for each group member).

Comparison

All four of the models should be compared on the test set and a winner declared (this should be automated to be correct across all the created documents).

This can be done by either/both group members.

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` variable. You'll end up with seven total outputted documents.

This can be done by either/both group members.

Submission

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

Group Issues

Please notify me ASAP of any group member issues.

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	20	Worth either 0, 5,. . . , or 20
Modeling, selection, & discussion	30	Worth either 0, 5, . . . , 30
Test set prediction	5	Worth either 0 or 5
Automation	20	Worth either 0, 5, . . . ,20
Blog post and repo setup	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