**MSDS 6306: Doing Data Science**

**Case Study 02**

**Due: One hour before April 24-Live Session**

**Description**: DDSAnalytics is an analytics company that specializes in talent management solutions for Fortune 1000 companies. Talent management is defined as the iterative process of developing and retaining employees. It may include workforce planning, employee training programs, identifying high-potential employees and reducing/preventing voluntary employee turnover (attrition). To gain a competitive edge over its competition, DDSAnalytics is planning to leverage data science for talent management. The executive leadership has identified predicting employee turnover as its first application of data science for talent management. Before the business green lights the project, they have tasked your data science team to conduct an analysis of existing employee data.

Your team has been given a dataset (CaseStudy2Data.zip) to conduct exploratory data analysis (EDA) to determine factors that lead to attrition. You should identify (at least) the top three factors that contribute to turnover. There may or may not be a need to create derived attributes/variables. The business is also interested in learning about any job role specific trends that may exist in the data set (e.g., “Data Scientists have the highest job satisfaction”). You can also provide any other interesting trends and observations from your analysis. **The analysis should be backed up by robust experimentation and (where applicable) appropriate visualization**. Experiments and analysis must be conducted in R. If needed, consult with the business owner to get clarifications on the dataset or business goals.

**Deliverables:**

This is a group project so it’s the responsibility of the group members to collaborate accordingly. Team work is important.

The due date for submission is 1 hour before live session 15. During live session 15, the group will have up to 20 minutes to present their case study report. The goal is to communicate the findings of the project in a clear, concise and scientific manner.

Create a GitHub repository with an RMarkdown file containing an executive summary, introduction to the project, all supporting code and analysis, and the group presentation.

Submit a link to the GitHub repository via the space provided for the Case Study 02 page in 2DS.

Your Presentation should be in this outline

* Title
  + Authors all listed
* Presentation Outline
  + Business Objectives
  + Data Sourced
  + Methodology
  + Evaluation/Results
  + Summary
* Business Objectives
* Data Source
  + Where you got it
  + Basic Statistics
* Methodology
  + Steps
  + Workflow
* Evaluation/Results
  + Tell me the percentages
  + Show me graphs with explanations
  + The top three factors that contribute to turnover
  + Tell me about any job role specific trends that may exist in the data set
  + Provide any other interesting trends and observations from your analysis
  + Other things to consider?
* Summary
  + Insights
  + Recommendations
  + Improvements
* Questions

Have your presentation as a PDF ready to present on April 24.

NO CODE should be present in your presentation. You code should be in your GitHub and RMarkdown File.

Remember, no late assignments will be accepted, as this is a final exam. It’s been a pleasure teaching you this semester! Congratulations on making it through your first semester in the Masters of Data Science program. Good luck!

# Remember:

A GitHub repository and peer-grading email.

1. Introduction to the project. The introduction should not reference a project, persay. No part of this should be informal. It should be appealing-looking as well.
2. The introduction needs to be written as if you are presenting the work to someone who has given you the data to analyze and wants to understand the result. Pretend it’s a presentation for a client. This may take some imagination of whom your client might be. **If it sounds like a student presentation, that is not acceptable**.
3. As you do your ‘sales pitch’, briefly explain what your code is doing (both procedurally and for the customer). The explanations should appear as a sentence or two before or after each code chunk. Even though you will not be hiding the code chunks (so that I can see the code), you need to pretend that the client cannot see them.
4. R code for answers concerning the tasks below. Make it in RMarkdown file format and always include echo=TRUE and include=TRUE for charts. **Keep the .md file so I can readily see everything on GitHub! Your .md file should mirror the .RMD file.**
5. Give clear, explicit answers to the questions. Just the code to answer the questions is not enough, even if the code is correct and gives the correct answer. You must state the answer in a complete sentence outside the code chunk.
6. Once you’re finished, be sure to frame a conclusion to the project. The presentation does not stop when you’re done with the questions. Find a way to wrap it up: summarize your findings from this exercise. Again: **the file must be readable in GitHub. In other words, dont forget to keep the md file!!**
7. You should expand your repository with at least this RMarkdown file, the input file(s), and a README.md that describes the purpose of the project and codebook. The repo can be structured how you like, but it should make sense and be easily navigated. If things are not clear from the root directory, you will lose points.

You’re working on this as a group. I expect that you both will do equal work. Both of you will need to push, add, commit, and pull the GitHub. If I do not see equal effort (and remember, GitHub tracks commits!), I reserve the right to penalize either or both students. Part of your grade will also be contingent on your partner’s evaluations of your effort. This is a collaborative project, so take it seriously and plan ahead with your teammate

# Tasks

*Tip:* I recommend in the code block commenting which number the code answers. It will help me find your answers more readily. Something as simple as # 3B helps me.

1. Formulate your Repo
   1. The client wants this to be reproducible and know exactly what you did. There needs to be an informative Readme, complete with several sections, as referenced in Live Session. Give contact information, sessionInfo, and the objective of the repo at least.
   2. You have two large data sets, and they will need its own Codebook, formatted in an approachable way. Make sure you describe peculiarities of the data by variable and what needs transforming. However, do not make it too long either.
   3. Create a file structure that is accessible and transparent. Document it in the root directory, ideally in the Readme.
2. Clean your Raw Data
   1. Read the csv into R and take a look at the data set. Output how many rows and columns the data.frame is.
   2. The column names are either too much or not enough. Change the column names so that they do not have spaces, underscores, slashes, and the like. All column names should be under 12 characters. Make sure you’re updating your codebook with information on the tidied data set as well.
   3. Some columns are, due to Qualtrics, malfunctioning.
   4. Make sure your columns are the proper data types (i.e., numeric, character, etc.). If they are incorrect, convert them.
3. Preliminary Analysis
   1. Remove all observations where the participant is under age 18. No further analysis of underage individuals is permitted by your client. Remove any other age outliers as you see fit, but be sure to tell what you’re doing and why.
   2. Please provide (in pretty-fied table format or similar), descriptive statistics on at least 7 variables (age, Income, etc.). Create a simple histogram for *two* of them. Comment on the shape of the distribution in your markdown.
   3. Give the frequencies (in table format or similar) for Gender, Education, and Occupation. They can be separate tables, if that’s your choice.
   4. Give the counts (again, pretty table) of management positions.
4. Deeper Analysis and Visualization
   1. *Note:* You should make all of these appealing looking. Remember to include things like a clean, informative title, axis labels that are in plain English, and readable axis values that do not overlap.
   2. Create barcharts in ggplot or similar The bars should be in descending order, Use any color palette of your choice other than the default.
   3. Is there a relationship between Age and Income? Create a scatterplot and make an assessment of whether there is a relationship. Color each point based on the Gender of the participant. You’re welcome to use lm() or similar functions to back up your claims.
   4. What about Life Satisfaction? Create another scatterplot. Is there a discernible relationship there to what?