**Cruise Line Exploratory Data Analysis**

 For my cruise line EDA, I though it would be most interesting to subset the data by each of the cruise line companies. This way, I can get a good picture of how the liners compare to one another across all the given categories. First, I used the command ‘names (cruise)’ to get an idea of all the headers in the file. The console responded with the following: "ship", "cline", "age", “tonnage", "passengers", "length", "cabins", "passdens", "crew". Then, I decided to get a list of all the different cruise lines in the file, so I used the command ‘unique(cruise$cline); and this gave me the following results: "Azamara", "Carnival", "Celebrity", "Costa", "Crystal", "Cunard", "Disney", "Holland American”, "MSC", "Norwegian", "Oceania", "Orient", "P&O", "Princess", “Regent Seven Seas", "Royal Caribbean", "Seabourn", "Silversea,", "Star" and "Windstar". Now that I was aware of all the different categories, I felt it was time to subset the data by each of the cruise lines. I used the command ‘cruise line abbreviation= subset(cruise, cruise$cline==”cruise line name”)’. What I found here that was interesting was that the original data file had excess spaces behind the cruise lines’ names, so that meant that I could not type the cruise line names into the command manually, rather, I had to copy and paste the cruise line names into the command because it would read an error if the command didn’t have the right number of spaces behind the cruise line name. At this point I realized that it might be a waste of time to have to go through and give individual commands for each of the 20 subsets, so I decided not to use any of data frames that I had subsetted in the commands. However, it was still helpful to be able to go through the data and simply look at the information as it was sorted by cruise line. Perhaps it wasn’t entirely a waste of time. Shortly thereafter, I decided to approach the analysis much like the way the instructional video on YouTube suggests. I had already got a look at the headers, so I decided to use the command ‘summary(cruise)’. This gave me an array of information, like the minimums and maxes for each column as well as means and medians for each value. However, I soon felt the urge to know which cruise liner held the title for heaviest, longest and/or oldest ship. For this, I had to use two commands for each category. I used the commands ‘cruise$ship[which.max(cruise$tonnage)] and cruise$cline[which.max(cruise$tonnage)]. This informed me that the heaviest ship in the data set was the Royal Caribbean’s ‘Oasis’ at 220,000 tons. This is well over double the median weight of Royal’s ships which is roughly 91,000 tons (I did find a use for all that subsetted data!). Not surprisingly, the ‘Oasis’ is also the longest ship at 1,182 ft. This is substantially above the industry median of 855.5 ft and the Royal Caribbean’s median of 962 ft. Finally, I found that the oldest ship was the Orient’s “Marco Polo” at 48 years old…almost twice as old as me! For all of these commands, I used ‘cruise$ship/cline[which.max(cruise$tonnage/length/age’. Next, I decided to produce a graph which shows the passenger density by each of the ship names. #The only two things I wasn’t pleased with were the values that showed up in black and the fact that I couldn’t figure out how to get the data sorted in such a way that the barplot displayed them from least to greatest, left to right. I tried the ‘sort’ command, but this always took the variable from the dataframe and turned it into a vector instead of just sorting the data and keeping it in a data frame. As for the values that show up in black, as you can see from the code, I told it to only display values above 60, but it shows these lower values anyway AND I told it to only use the colors “purple”, “red” and “green’, but it still shows these values in black that are below my cutoff.# For this I used the command ‘barplot(cruise$passdens, cruise$passdens>60, names.arg =cruise$ship, col = c("purple", "red", "green"), las=2, cex.names = 0.6, main= "Cruise Line Passenger Density by Ship", ylim = c(40,70))’



Next, I wanted to check out some correlations within the data. First, I installed “Hmisc” so that I could use the command ‘rcorr’. #For some reason, it would not let me run the rcorr(as.matrix()) command; the error message read “NA/NaN/Inf in foreign function call (arg 1)”.# Interestingly, I found that the amount of passengers on the ship is not significantly correlated with passenger density. After testing multiple different correlation possibilities (since it would not let me run the matrix all at once), I did finally find some significant, strong correlations. One was that tonnage and length of the ship had a very strong correlation at 0.92 and a p value of 0. Cabins and length were correlated at 0.89 also with a p value of 0. Tonnage and passengers were correlated at 0.95 with a p value of 0. Passengers and cabins were correlated at 0.98 with a p value of 0.

Lastly, I used the corrgram command to make a correlogram of the cruise data.



Though I was unable to get the ‘as.matrix command to work., this command almost serves the same purpose just without the respective p values. We can see here that tonnage, length, cabins, and passengers all seem to have a strong correlation with one another in this data set. Interestingly, age seems to have roughly a 0.5 correlation with almost every variable. I suppose that this means that tonnage, length, crew, cabins and passenger capacity increase somewhat as the age of the ship goes down. The command I used for this correlogram reads as follows: ‘corrgram(cruise, order=TRUE, lower.panel = panel.pie, upper.panel = panel.ellipse, text.panel = panel.txt, main= "Cruise Line Correlogram")’.