Physics 360 Mid Term Exam November 2019

Instructions: You will be building four models: linear regression (`lm`), KNN, LDA, and logistic regression (`glm`). Your answers should provide sufficient material (typically by copy & paste from R) so the grader can understand exactly what you did. Your answers will certainly contain summary outputs of every significant fit (including significance of variables and measures of fit quality), but additionally include relevant material related to those fits. Your answers should be your work; aid from non-human sources (e.g., Google, your textbooks) is allowed, as is aid from your instructor, but other human sources are not allowed. If you are stuck, Kirkman will try to provide direction and/or clarification, so feel free to ask! Your work is due Tuesday 19-Nov-2019 6 P.M..

1. (**linear regression**) The file `WHO19.csv` contains 193 rows (countries) with 77 variables (mostly health related). Warning: the file has many `NA` (indeed just four `complete.cases`). Here, for example, are the variables not reported for US:

   ```
   > names(d)[is.na(d[d$Country=="United States of America",])]
   [1] "literacy."
   [2] "Aid"
   [3] "External_debt_per_GNI"
   [4] "Literacy_rate_adult_female"
   [5] "Literacy_rate_adult_male"
   [6] "Literacy_rate_adult_total"
   [7] "Primary_completion_rate_total"
   [8] "Ratio_of_young_literate_females_to_males"
   ```

   Aim: explain some variable (your choice) in terms of linear regression of some other variables. Your model must contain at least three significant (here defined as $p < .1$) variables (not counting Intercept). Your model should contain at least one significant interaction term, or show evidence that you tried hard to find such a term and failed. If you become exhausted trying to meet these criteria, talk to Kirkman.

   Remark: column names that end in "`.`" had % in the file; R does not like % in column names and has replaced it with "`.`".

   Provide a concluding summary statement of every variable in your model starting: "Correlation does not imply causation, but", changing these dependent variables (report whether to increase or decrease each variable) seems like a reasonable starting point to improve (report whether to increase or decrease) the dependent variable I selected.

2. (**KNN**) The file `wine.csv` contains 178 observations of 14 variables relating to the physical/-chemical properties of various classes of Italian wine. Recall that to do KNN you need the `class` library. After reading in the file, begin by converting the `class` variable to a factor:

   ```
   d$class=as.factor(d$class)
   ```

   Aim: Using KNN, provide a way to predict the `class` of a wine based on its physical and chemical properties. With just 178 rows, classic train/test will not be used; train with the entire dataset. 14 variables is uncomfortably large from your instructors point-of-view (see *the curse of high dimensionality* in ISLR), so begin by rationally selecting just 4–5 variables which show large variations between classes. If you want you can do this by-eye: examining 13 individual boxplots for significant variation between groups. (If you adopt this approach: report in words how you interpreted each plot; no need for lots of boxplot hardcopies.) Alternatively, you might recall (`wilt.pdf` bottom p. 3) that we automatically evaluated 2-class differences using a t-test and a newly-created function. The R function `aov` (analysis of variance) provides the F statistic which is analogous to t but for differences between *multiple* groups. The command:

```
summary(aov(d[,14]~d$class))[[1]][1,4]
```

reports the F statistic for differences in column 14 (Proline) according to the 3 `class`es. Yea, it's a pretty opaque command[1]—I found it by Googling. `summary(aov(...))` is a printout like `summary(lm(...))`, the `[[1]][1,4]` simply selects exactly the F statistic from that printout. Large F means large differences between groups. It should then be easy to define a function to `return` this result for any numbered column and then `sapply` the columns `2:14` to that function. IF you adopt this approach document your commands.

Once you have selected your variable subset, for the following provide the exact commands you used:

- make a subset data.frame just including your 4–5 selected independent variables and `class`
- make a scaled data.frame of your dependent variables
- use KNN to predict the `class`
- make a confusion matrix table

Do the above for $k = 5, 7$ and both scaled and raw independent variables. Report (in words) which is 'best' and how you selected 'best'.

3. (**LDA**) Continue to use your dimensionally reduces wine data.frame. For the following provide the exact commands you used:

   - use LDA to predict the `class`; include the output (text summary).
   - make a confusion matrix table

   Do the above using both your scaled and raw selected variables. Is there any difference? Compare (in words) the LDA result to the KNN result: report which is better and why.

4. (**logistic regression**) In the `stocks` directory find files with names: X`2000.csv` where X is a stock market name of a prominent Minnesota company. Select one of those files; read it into R. You should find columns:

```
> str(d)
'data.frame': 4821 obs. of  14 variables:
 $ Date      : Factor w/ 4821 levels "2000-01-03","2000-01-04",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ Close     : num  47.2 45.3 46.6 50.4 51.4 ...
 $ Volume    : int  2173400 2713800 3699400 5975800 4101200 3863800 2357600 2868400 2244400
 $ Today     : num  -0.0371 -0.0414 0.0282 0.0744 0.0195 ...
 $ Lag1      : num  0.0153 -0.0371 -0.0414 0.0282 0.0744 ...
 $ Lag2      : num  -0.0117 0.0153 -0.0371 -0.0414 0.0282 ...
 $ AVolume   : num  2195387 2207000 2263013 2377687 2418267 ...
 $ TodayV    : num  0.53 0.199 0.266 0.381 -0.457 ...
 $ Lag1V     : num  -0.671 0.53 0.199 0.266 0.381 ...
 $ Lag2V     : num  -0.138 -0.671 0.53 0.199 0.266 ...
 $ AClose    : num  47.7 47.5 47.4 47.5 47.6 ...
 $ Lag1BullC: num  1.081 -0.528 -2.209 -0.801 2.881 ...
 $ Lag1BullV: num  -1192340 -21987 506800 1436387 3598113 ...
 $ Direction: Factor w/ 2 levels "Down","Up": 1 1 2 2 2 1 1 2 1 1 ...
```

   Aim: Using logistic regression predict the stock's `Direction` based on performance on the previous two days (the variables that start `Lag`). Remark: the `Lag1Bull` variables are somewhat fancier predictors based on closing price and volume. For convenience in plotting convert `Date` (currently a Factor) into Rs time-series variable-type

[1] No surprise, R has a package that provides a 'tidy' command to do the same thing.

```
d$Date=as.Date(d$Date)
```

Your final model should include as many *significant* (here defined as $p < .1$) variables as possible and no insignificant variables. If you can't find any significant variables, consult Kirkman or try a different stock. Provide summary output of that final model. `predict`ing with your model should produce probabilities for `Up`; let's call that vector `out`. If you always bought when `Up` was likely and sold when `Down` was predicted your daily percentage gains/loss would be (assuming no trading costs):

```
d$Today*(out>.5)-d$Today*(out<.5)
```

Explain exactly how this command works including structure/meaning/type/length of each of the four terms. Your average gain would be

```
mean(d$Today*(out>.5)-d$Today*(out<.5))
```

Almost always this result is a (small) positive average gain, but given all the fluctuations is that gain really significantly greater that zero (in terms of standard deviation of the mean)? Provide the output of a one-sample t-test to resolve this issue.

Realistically you probably would not trade if `out` was very close to .5. Make a histogram of `out`; Play around with some cuts that exclude trades when `out` is near .5 and that do not eliminate a large fraction of trading days. See if you can come up with a trading strategy whose average gain is significantly greater that zero. (If the .5 cuts were already significant, see if you can improve the p a bit.) Provide the commands (showing the cuts) that define your final strategy and the evidence (t-test) that the strategy produced significant positive gain. (Do this even if your strategy in the end actually failed to provide a significant positive average return.) This is your trained strategy.

`cumsum()` is an R function that returns the cumulative sum of a vector. Using your final strategy, plot your cumulative return vs. `Date`. (Use `pch="+"` or `pch="."` for smaller data points.) Provide hardcopy of your plot.

You may have noticed that the run of stock prices runs from Jan-2000 until Mar-2019 (when I downloaded the data). The folder `new` in `stocks` contains files X`2020.csv` which record the stock results since Mar-2019. Read in the recent data for your stock, and `predict` how your model would apply to the recent data (`out2`). Using the model and the cuts you established for the past data, find the vector of your daily returns using the recent data. Plot your cumulative returns since Mar-2019 and provide hardcopy of the results.

5. As you know the 'final exam' for this course consists of you presenting an analysis of data. Please record here an indication of what data you are thinking of analyzing. (I won't hold you to this choice if you change your mind, but procrastination is risky!)