Problem Set 8

Due on November 7, 2023, 10am (but PCE due Nov 6)

To successfully complete this problem set, please follow these steps:

1. Download the contents of the assignment Dropbox link in a folder (directory) on your computer designated for this problem set only. Follow the directory structure, e.g., putting the `data` folder containing the datasets as a subdirectory of the project directory.

2. Insert your answers in the yellow boxes using Microsoft Word, and prepare a single `.R` script for what you produce. Save the word document as a`.PDF`.

3. Please submit the PDF to the designated `PS-XX: pdf` link and your R Script to the `PS-XX: R` link.

(1) Your name:

Nikhilla Bhuvana Sundar

(2) Group members, if any:

(3) Compliance with the Academic Code on problem set[1] (sign with an X below)

X

[1] You may use the same code from classmates, Ed Discussion Board, instructors, and generative AI. However, you must hand in your own unique written work and code in all cases. Any wholesale copy/paste of another's work is plagiarism. In other words, you can work with your classmate(s), sitting side-by-side and going through the problem set question-by-question, or use generative AI to provide potential code for you, but you must each type your own answers and your own code.

Gradescope will have the following packages installed:

```
library(tidyverse)
library(rsample)
library(glmnet)
library(glmnetUtils)
library(glue)
library(gt)
```

This problem set works through the following case:

> Improving Worker Safety in the Era of Machine Learning (A). Harvard Business School Case 618-019 (2018).
> https://www.hbs.edu/faculty/Pages/item.aspx?num=53417

which you can purchase for $4.95 for PLSC438/536 using this link with a student login: https://hbsp.harvard.edu/import/1111604. You should also watch the following documentary for background by Monday's class. This somewhat dated documentary merely serves to help you visualize OSHA a bit better.

> PBS documentary (2008), A Dangerous Business, Revisited. (55min, be warned there is some disturbing imagery around 13:00 - 15:00)

The case PDF includes a description of the data and the variables in the Appendix. Finally, please submit pre-class exercise by

Please find the dataset for this problem set enclosed. We can read it as a CSV:

```
osha <- read_csv("data/osha.csv")
```

The data contain the following outcome variables. In this problem set, we will be only modeling the variable `injury_rate`.

Variable name	Description
`injuries`	Number of days away from work (DAFW) injuries workplace had in year t
`injury_rate`	Injury rate (DAFW per 100 FTE) in year t, with DAFW = Days Away from Work
`high_injury_rate`	Injury rate (DAFW per 100 FTE) in year t is 3 or higher

Problem 1

This problem compares the in-sample fit of three OLS models of varying complexity. The setup of the kitchen model will be useful for Problem 2's LASSO regression.

1.1

Estimate a linear regression model that predicts `injury_rate` with only the following variables listed in the case.

Variable name	Description
`estab_age_netsmm1`	Workplace age in years
`exp_imp_netsmm1`	Workplace is either an exporter or importer
`exporter_netsmm1`	Workplace is an exporter
`foreign_owned_netsmm1`	Workplace is owned by a foreign entity
`govt_contractor_netsmm1`	Workplace is a government contractor
`headquarters_netsmm1`	Workplace is the headquarters of its firm
`importer_netsmm1`	Workplace is an importer
`public_netsmm1`	Workplace is publicly owned
`standalone_netsmm1`	Workplace is a single-unit firm
`num_yrs_prev_on_sst`	Years establishment has been on SST list before this year

Report the (a) total sums of squares of this data and the (b) sum of squared residuals from the model. And using these two numbers, report the (c) R-squared of this regression.

- TSS: 69766.64
- SSR: 69139.68
- R-squared: 0.008

1.2

Fit what the case calls the "simple model" and the "kitchen sink model" regression for the injury rate (`injury_rate`) on the entire dataset, and in the yellow box **summarize the R-squared of these three regressions in a sentence or two**.

- The *simple model* uses data on inspections or has received some complaints in the past denoted in the Appendix Table of the case. For convenience, below is the R formula representation of these variables that can be used directly in R:

```
form_simpl <- injury_rate ~ has_tmin1_odi + any_insp_prior +
  any_complaint_tmin13 + num_nonfat_comp_insp_cy_tc99mm1 +
  initial_pen_cy_mzmm1 + ln_initial_pen_cy_mzmm1
```

- The *kitchen-sink model* should include all the available variables in the data EXCEPT for the workplace identifier (`id`) and the outcome (`injuries`, `injury_rate`, and `high_injury_rate`). Create this formula object, again with the outcome `injury_rate`.

RS_first: 0.008

RS_simple: 0.007

RS_kitchensink: 0.427

Kitchen sink model has the highest R squared value as it over fits the data points and seems to include the maximum number of variables in the model. The first and simple model seem to have more or less the same R squared value and explain about 8% of the variations in the outcomes.

Problem 2

Now we will evaluate the out of sample predictive accuracy of three models, including a LASSO regression.

2.1

Randomly split the data 80-20:

- Use the 80 percent as the training dataset and assign it to an R object called `osha_train`.
- Use the remaining 20 percent as the test dataset and assign it to an object called `osha_test`.

Here, please use the function `rsample::initial_split`, and use the seed `06510` with `set.seed()` as shown in the screencast. Setting a seed and running the command together with initial_split ensures that the random assignment is reproduced exactly each time.

No response needed here (do this in your R code)

2.2

The case considers two types of predictive modeling within OSHA's Approach 3 (Highest predictive injury rate). We consider a third LASSO model in addition to the two mentioned in the case:

1. An OLS regression with the simple model (see problem 1.2)
2. An OLS regression with the kitchen sink model (see problem 1.2)
3. A linear LASSO regression with the same variables as the kitchen sink model but using a penalty term chosen to minimize the MSE within cross-validation.

Fit these three models on the training data and **evaluate the in-sample RMSE in the *training* set and the out-of-sample RMSE in the *test* set.** Your final table will therefore consist of six numbers (3 models, in-sample vs. out-of-sample). Please fill in the table below or show R output that follows this format.

Creating this table will require multiple steps, and there are multiple correct ways to process the data and summarize the results. See the appendix for some guidance, including tutorial videos. Aim for conciseness and readability of your code: This exercise can be done in about 50 lines (including chunk headers with comments, tables commands, and spaces between code chunks).

	RMSE	
	Training Dataset	Test Dataset

Simple Model	3.34	3.20
Kitchen-Sink Model	2.53	2.52
LASSO Model	2.55	2.49

2.3

Out of the 180 variables considered by the LASSO, how many eventually stay in the model? That is, how many variables are not zeroed out by the LASSO? Consider the case when the penalty term is set at the MSE-minimizing value.

89 variables are not zeroed out by LASSO

2.4

In Monday’s class, we will discuss the OSHA case: both the statistics underlying the LASSO regression, the policy considerations when using machine learning, and the use of prediction methods in social science research. Please answer the following questions as a Pre-class Exercise by Monday **November 6, 10am**.

https://yale.instructure.com/courses/79832/quizzes/58804

2.5: Optional

There are a few ways to extend this analysis, that you may consider as a starting point for your final project replication if you choose to explore this topic. A modest extension is to explore if your findings are robust to different random seeds and different fractions to split the data. Another is to consider models other than the LASSO. Some of these are listed below.

If you try out any of these extensions before class, please email it to shiro.kuriwaki@yale.edu.

4. A **null** model as a benchmark, computed with only the intercept (no covariates) in OLS. The formula syntax for this is `injury_rate ~ 1` where `1` represents an intercept that is implicitly assumed when there are covariates.
5. A **ridge** model similar to the `glmnet`’s LASSO. In fact, this is simple as setting the `alpha` argument to `0` instead of `1` in `cv.glmnet`.

6. Another version of the LASSO (`hdm::rlasso`) where the penalty term is chosen by the data (Chernozhukov et al.). We will call this **post**-LASSO. The name comes from the fact that variables selected by this LASSO are later (post) run as OLS.
7. A regression tree, which we'll call **forest**. The function `randomForest::randomForest` is one of the standard package.
8. A model of your choosing that you think will achieve the best out of sample prediction accuracy.

N/A

Problem 3

In other applications, the outcome of interest is a probability of a binary event happening, rather than a continuous number like the injury rate. In this problem, we will follow the same exercise 2.2 but replace the continuous outcome with a binary outcome.

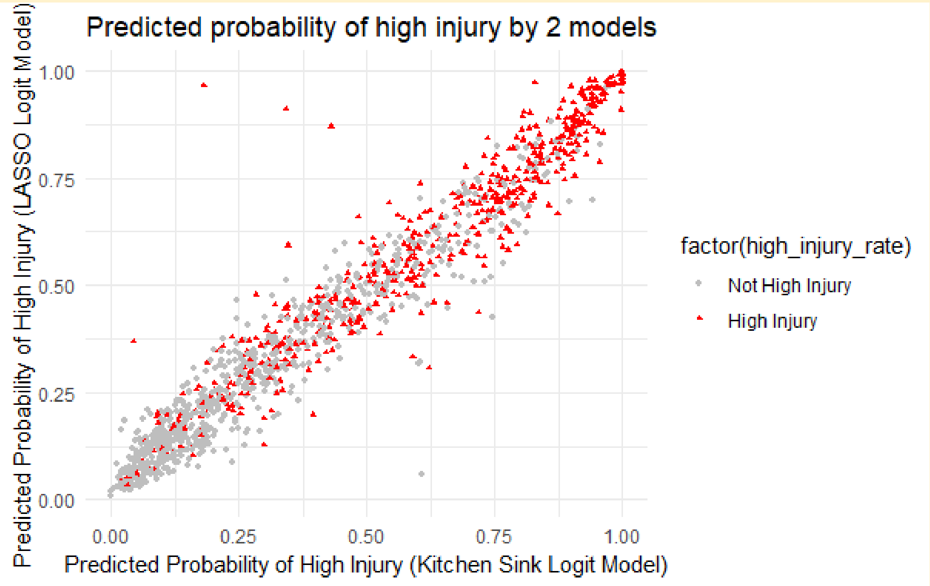
Fit the simple, kitchen sink, and LASSO specification on the same 80% training dataset. But this time, use the binary `high_injury_rate` as the outcome instead of the continuous outcome, and use a logit model instead of an OLS.

Then present the predicted *probability* of high injury for each observation in the test data ($\hat{\pi}_i$) in the following **scatterplot**:

- The x-axis should present the predicted probability of high injury as specified by the kitchen sink logit model, $\hat{\pi}_i^{\text{kitch}}$
- The y-axis should present the predicted probability of high injury as specified by the kitchen sink logit model, $\hat{\pi}_i^{\text{LASSO}}$
- The points should be colored and shaped by their value of `high_injury_rate`. Just as an example, points that are not high injury cases should be gray circles and points that are high injury cases should be red triangles.[2]

Finally, briefly **summarize** what you find about the out of sample accuracy of the two models from this figure.

[2] Using both color and shape to distinguish a variable helps with colorblind readers or grayscale print outs. (https://clauswilke.com/dataviz/redundant-coding.html)



Both the models seem to predict the outcome in more or less similar manner with most of the data points falling along the 45-degree line. However, there are outliers in prediction especially for the LASSO model and the data points seem to deviate from the 45 degree line.

Problem 4

Please give us an update on your thinking about the final project (due Dec 19), touching on all or some of the following:

- Are you currently leaning towards a replication of a class paper or an independent analysis for your final project?
    - If you are considering choosing a paper provided in class, which one(s) are you leaning towards?
    - If an independent analysis, have you received permission from me (the professor)? If you have, please describe it here again so I can check with my notes.

Currently, I am leaning towards a replication of a class paper for final project. I am most likely to go ahead with the one on instrumental variables by Carnegie and Marinov.

I am thinking of pairing up with Adaolisa from class to work on this project. However, we will meet up with Prof Shiro on Tuesday to finalize this.

Your responses will help the teaching team coordinate teaching efforts. You should feel free to use the “Private Post” feature on Ed / email / office hours to make progress.

Appendix

For the use of LASSO, refer to the lecture material or the following screencasts for LASSO on a different dataset:

1. setup and pre-processing: https://vimeo.com/878731294
2. fitting and understanding cv.glmnet: https://vimeo.com/878731749
3. making predictions: https://vimeo.com/371322903.

You may also find this manual by the authors of `cv.glmnet` to be a more comprehensive written reference: https://glmnet.stanford.edu/articles/glmnet.html.

The following suggestions are more specific to the data and case at hand. Differences with the screencasts are highlighted in bold.

1.1

For simple one-shot operations like this that do not use dataframes, it is easier to use base-R notation to extract columns of data. The `$` notation applied to a dataframe extracts the column as a vector, so `osha$injury_rate` is the vector of the `injury_rate` variable.

1.2

Typing out 100+ variables in a formula is unwieldy, so we will need to use some code to construct the formula for the kitchen-sink regression. We recommend using the approach used in PS-06 problem 2.2 that uses `colnames` and `glue`.

Before collapsing together all variable names in the OSHA data, we first need to remove the specified variables. This can be done several ways. For example, suppose you wanted all column names in the dataset `data` except for the variables `A`, `B`, and `C`:

- Remove the columns by `select` and then extract the column names as a vector: `data |> select(-c(A, B, C)) |> colnames()`. The column names can be unquoted because they are variable names that exist in `data` that are used within `tidyverse` functions.
- Alternatively, you can extract the column names as a vector first, and then use a base-R command for vectors, `setdiff`, to remove those names. `setdiff(colnames(data), c("A", "B", "C"))`. `setdiff` is the command for giving the difference in two sets. Note how the column names are now in quotes here because this is not tidyverse.

The `all_of()` function in `dplyr()` does the opposite, it takes a vector of column names in characters and gives all columns that match that vector.

2.2

The videos use the `glmnetUtils` function to abbreviate some preparation and make your code more concise.

- The package `glmnetUtils` provides some convenient shortcuts and only requires the users to input a formula and dataframe. It will then construct the matrices itself internally.
- To use this, install and load `glmnetUtils`. This loads a function `cv.glmnet.formula` that is equivalent to `cv.glmnet` except that its first argument defaults to a formula rather than a vector or matrix. In fact, once `glmnetUtils` is loaded, `cv.glmnet` (without

.formula explicitly stated) can be used with formula input to mean cv.glmnet.formula. These abbreviations in R are called "methods" or more specifically S3 methods and allow generic functions like summary and predict apply to various types of input.

- For more examples on using glmnetUtils, see the package author's manual at https://cran.r-project.org/web/packages/glmnetUtils/vignettes/intro.html.

without using glmnetUtils

- See the 2019 version of my LASSO screencast, where I did not rely on glmnetUtils: https://vimeo.com/371322839

Predictions and RMSE

- Add predictions to the training and test datasets as shown in class and the third screencast.
- With the predictions attached as a variable a dataframe, you can compute the RMSE by the summarize function just like you computed sample averages. For example, data |> summarize(MAD = mean(abs(A - B))) summarizes the mean absolute deviation between the variables A and B.
- Recall that the RMSE for the length-n observed outcome vector y and the corresponding predicted values $\hat{y}$ is defined as:

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}.$$

- Good code avoids duplication, because duplication increases the risk of errors. We could compute the RMSE six times for each of the summary statistics, but we can avoid duplicating essentially the same code six times by storing the data in "long" form. Suppose there were 50 data rows in the training set, 50 in the test set, and three models. A long dataset with 300 rows with the variables set (training or test), model (each of the three modes), and a single prediction column and a single outcome column, we could compute six RMSEs at once by group_by(set, model) and then executing the summarize function.
- To format the table, you may want to use pivot_wider and gt as you have in previous problem sets.

2.3

- Extract the coefficients of the LASSO object as in the second screencast, and then evaluate whether its value is 0.

Problem 3

- `lm()` is only for OLS, so for a logit regression use `glm()`. The `glm()` is a generalized linear model which can fit several class of models, of which OLS (the default) is one and logit is another. The class of model is set by the `family` argument, and `family = binomial` represents a logit.

- `cv.glmnet` and `glmnet` includes several class of models, including logit. It adopts the same convention as `glm` so `family = binomial` will estimate a logit model.

- `predict` will also take a glm fit object, but there are two types of predictions for a logit model. The unbounded predictions (e.g., -10, 0, 10) or the predictions transformed to a probability (e.g., 0.01, 0.5, 0.999) wit the logistic function $f(x) = e^x/(1 + e^x)$ where x is the unbounded prediction. Setting the argument `type = "response"` in the predict function will estimate the probability (i.e., the output of *f(x)*).