Week 4 - Homework

STAT 420, Summer 2017, Dalpiaz

Directions

- Be sure to remove this section if you use this `.Rmd` file as a template.
- You may leave the questions in your final document.

Exercise 1 (Using `lm`)

For this exercise we will use the data stored in `nutrition.csv`. It contains the nutritional values per serving size for a large variety of foods as calculated by the USDA. It is a cleaned version totaling 5,138 observations and is current as of September 2015.

The variables in the dataset are:

- `ID`
- `Desc` - Short description of food
- `Water` - in grams
- `Calories`
- `Protein` - in grams
- `Fat` - in grams
- `Carbs` - Carbohydrates, in grams
- `Fiber` - in grams
- `Sugar` - in grams
- `Calcium` - in milligrams
- `Potassium` - in milligrams
- `Sodium` - in milligrams
- `VitaminC` - Vitamin C, in milligrams
- `Chol` - Cholesterol, in milligrams
- `Portion` - Description of standard serving size used in analysis

(a) Fit the following multiple linear regression model in `R`. Use `Calories` as the response and `Carbs`, `Fat`, and `Protein` as predictors.

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \epsilon_i.$$

Here,

- Y_i is `Calories`.
- x_{i1} is `Carbs`.
- x_{i2} is `Fat`.
- x_{i3} is `Protein`.

Use an F-test to test the significance of the regression. Report the following:

- The null and alternative hypotheses
- The value of the test statistic
- The p-value of the test
- A statistical decision at $\alpha = 0.01$
- A conclusion in the context of the problem

When reporting these, you should explicitly state them in your document, not assume that a reader will find and interpret them from a large block of `R` output.

(b) Output only the estimated regression coefficients. Interpret all $\hat{\beta}_j$ coefficients in the context of the problem.

(c) Use your model to predict the number of `Calories` in a Big Mac. According to McDonald's publicized nutrition facts, the Big Mac contains 47g of carbohydrates, 28g of fat, and 25g of protein.

(d) Calculate the standard deviation, s_y, for the observed values in the Calories variable. Report the value of s_e from your multiple regression model. Interpret both estimates in the context of this problem.

(e) Report the value of R^2 for the model. Interpret its meaning in the context of the problem.

(f) Calculate a 90% confidence interval for β_2. Give an interpretation of the interval in the context of the problem.

(g) Calculate a 95% confidence interval for β_0. Give an interpretation of the interval in the context of the problem.

(h) Use a 99% confidence interval to estimate the mean Calorie content of a small order of McDonald's french fries that has 30g of carbohydrates, 11g of fat, and 2g of protein. Interpret the interval in context.

(i) Use a 90% prediction interval to predict the Calorie content of new healthy menu item that has 11g of carbohydrates, 1.5g of fat, and 1g of protein. Interpret the interval in context.

Exercise 2 (More `lm`)

For this exercise we will use the data stored in `goalies_cleaned.csv`. It contains career data for 462 players in the National Hockey League who played goaltender at some point up to and including the 2014-2015 season. The variables in the dataset are:

- `W` - Wins
- `GA` - Goals Against
- `SA` - Shots Against
- `SV` - Saves
- `SV_PCT` - Save Percentage
- `GAA` - Goals Against Average
- `SO` - Shutouts
- `MIN` - Minutes
- `PIM` - Penalties in Minutes

For this exercise we will consider three models, each with Wins as the response. The predictors for these models are:

- Model 1: Goals Against, Shots Against, Saves
- Model 2: Goals Against, Shots Against, Saves, Minutes, Penalties in Minutes
- Model 3: All Available

(a) Use an F-test to compares models 1 and 2. Report the following:

- The null hypothesis
- The value of the test statistic
- The p-value of the test
- A statistical decision at $\alpha = 0.01$
- The model you prefer

(b) Use an F-test to compare model 3 to your preferred model from part **(a)**. Report the following:

- The null hypothesis
- The value of the test statistic

- The p-value of the test
- A statistical decision at $\alpha = 0.01$
- The model you prefer

(c) Use a t-test to test $H_0 : \beta_{\text{SA}} = 0$ vs $H_1 : \beta_{\text{SA}} \neq 0$ for the model you preferred in part **(b)**. Report the following:

- The value of the test statistic
- The p-value of the test
- A statistical decision at $\alpha = 0.01$

Exercise 3 (Regression without `lm`)

For this exercise we will once again use the data stored in `goalies_cleaned.csv`. The goal of this exercise is to fit a model with `W` as the response and the remaining variables as predictors.

(a) Obtain the estimated regression coefficients **without** the use of `lm()` or any other built-in functions for regression. That is, you should use only matrix operations. Store the results in a vector `beta_hat_no_lm`. To ensure this is a vector, you may need to use `as.vector()`. Return this vector as well as the results of `sum(beta_hat_no_lm)`.

(b) Obtain the estimated regression coefficients **with** the use of `lm()`. Store the results in a vector `beta_hat_lm`. To ensure this is a vector, you may need to use `as.vector()`. Return this vector as well as the results of `sum(beta_hat_lm)`.

(c) Use the `all.equal()` function to verify that the results are the same. You may need to remove the names of one of the vectors. The `as.vector()` function will do this as a side effect, or you can directly use `unname()`.

(d) Calculate s_e without the use of `lm()`. That is, continue with your results from **(a)** and perform additional matrix operations to obtain the result. Output this result. Also, verify that this result is the same as the result obtained from `lm()`.

(e) Calculate R^2 without the use of `lm()`. That is, continue with your results from **(a)** and **(d)**, and perform additional operations to obtain the result. Output this result. Also, verify that this result is the same as the result obtained from `lm()`.

Exercise 4 (Regression for Prediction)

For this exercise use the `Boston` dataset from the `MASS` package. Use `?Boston` to learn about the dataset. The goal of this exercise is to find a model that is useful for **predicting** the response `medv`.

When evaluating a model for prediction, we often look at RMSE. However, if we both fit the model with all the data as well as evaluate RMSE using all the data, we're essentially cheating. We'd like to use RMSE as a measure of how well the model will predict on *unseen* data. If you haven't already noticed, the way we had been using RMSE resulted in RMSE decreasing as models became larger.

To correct for this, we will only use a portion of the data to fit the model, and then we will use leftover data to evaluate the model. We will call these datasets **train** (for fitting) and **test** (for evaluating). The definition of RMSE will stay the same

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

where

- y_i are the actual values of the response for the given data
- $\hat{y}_i$ are the predicted values using the fitted model and the predictors from the data

However, we will now evaluate it on both the **train** set and the **test** set separately. So each model you fit will have a **train** RMSE and a **test** RMSE. When calculating **test** RMSE, the predicted values will be found by predicting the response using the **test** data with the model fit using the **train** data. ***Test*** *data should never be used to fit a model.*

- Train RMSE: Model fit with train data. Evaluate on **train** data.
- Test RMSE: Model fit with train data. Evaluate on **test** data.

Set a seed of `42`, and then split the `Boston` data into two datasets, one called `train_data` and one called `test_data`. The `train_data` dataframe should contain 250 randomly chosen observations. `test_data` will contain the remaining observations. Hint: consider the following code:

```
library(MASS)
set.seed(42)
train_index = sample(1:nrow(Boston), 250)
```

Fit a total of five models using the training data.

- One must use all possible predictors.
- One must use only `tax` as a predictor.
- The remaining three you can pick to be anything you like. One of these should be the best of the five for predicting the response.

For each model report the **train** and **test** RMSE. Arrange your results in a well-formatted markdown table. Argue that one of your models is the best for predicting the response.

Exercise 5 (Simulating Multiple Regression)

For this exercise we will simulate data from the following model:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \epsilon_i$$

Where $\epsilon_i \sim N(0, \sigma^2)$. Also, the parameters are known to be:

- $\beta_0 = 1$
- $\beta_1 = 2.5$
- $\beta_2 = 0$
- $\beta_3 = 4$
- $\beta_4 = 1$
- $\sigma^2 = 16$

We will use samples of size `n = 20`.

We will verify the distribution of $\hat{\beta}_1$ as well as investigate some hypothesis tests.

(a) We will first generate the X matrix and data frame that will be used throughout the exercise. Create the following 9 variables:

- `x0`: a vector of length `n` that contains all `1`
- `x1`: a vector of length `n` that is randomly drawn from a uniform distribution between `0` and `5`
- `x2`: a vector of length `n` that is randomly drawn from a uniform distribution between `0` and `10`
- `x3`: a vector of length `n` that is randomly drawn from a normal distribution with a mean of `0` and a standard deviation of `1`
- `x4`: a vector of length `n` that is randomly drawn from a normal distribution with a mean of `0` and a standard deviation of `2`

- `X`: a matrix that contains `x0`, `x1`, `x2`, `x3`, `x4` as its columns
- `C`: the C matrix that is defined as $(X^\top X)^{-1}$
- `y`: a vector of length `n` that contains all `0`
- `sim_data`: a data frame that stores `y` and the **four** predictor variables. `y` is currently a placeholder that we will update during the simulation

Report the diagonal of `C` as well as the 10th row of `sim_data`. For this exercise we will use the seed `1337`. Generate the above variables in the order listed after running the code below to set a seed.

```
set.seed(1337)
sample_size = 20
```

(b) Create three vectors of length `2000` that will store results from the simulation in part **(c)**. Call them `beta_hat_1`, `beta_2_pval`, and `beta_3_pval`.

(c) Simulate 2000 samples of size `n = 20` from the model above. Each time update the `y` value of `sim_data`. Then use `lm()` to fit a multiple regression model. Each time store:

- The value of $\hat{\beta}_1$ in `beta_hat_1`
- The p-value for the two-sided test of $\beta_2 = 0$ in `beta_2_pval`
- The p-value for the two-sided test of $\beta_3 = 0$ in `beta_3_pval`

(d) Based on the known values of X, what is the true distribution of $\hat{\beta}_1$?

(e) Calculate the mean and variance of `beta_hat_1`. Are they close to what we would expect? Plot a histogram of `beta_hat_1`. Add a curve for the true distribution of $\hat{\beta}_1$. Does the curve seem to match the histogram?

(f) What proportion of the p-values stored in `beta_3_pval` are less than 0.05? Is this what you would expect?

(g) What proportion of the p-values stored in `beta_2_pval` are less than 0.05? Is this what you would expect?