Module 2 Worksheet

Prepared by: Katherine Daignault and the STA302H1 Fall 2025 Teaching Team

Worksheet Information

Goal of the worksheet:

The Module 2 worksheet is an opportunity to practice fitting multiple linear models in R and interpreting the coefficients when using different types of predictors. By completing this worksheet, each student will be developing the skills to achieve the following weekly learning objectives:

- Correctly interpret the estimated coefficients of a multiple linear model in the context of the dataset.
- Apply multiple linear models on various datasets using R statistical software.
- Differentiate the relationships modelled using qualitative predictors, interactions between predictors, and continuous predictors.

This worksheet, in addition to the remainder of the class time, is **important practice for completing questions on the term test and final exam, and for your final project**.

Preparation assumed:

For hybrid sections: As part of the flipped design of the course, it is assumed that each student is attending this lecture having completed the following pre-class preparation:

- Watched the Module 2 Videos, attempted the Pre-Class Quiz, and accessed the code provided (Guided Practice)
- (Optionally) attempted the corresponding LearnR module

For in person sections:

- Please complete this worksheet after attending your in-person lectures for each week. If you did not attend class, please review the annotated slides posted on Quercus which will be posted on the Module 1 Quercus Page.
- It is also recommended that you attempt the corresponding LearnR module, or have it open as a reference.
- Additional R/Coding resources are linked here.

How to complete this worksheet:

- Students may work in groups of 2-3 if desired. However **each student** must submit their worksheet to MarkUs to receive their completion credit. It is recommended that each student work on their **own copy** of the assignment.
- All the code and course knowledge needed to complete this worksheet has been provided in the pre-class materials. It may help to have these open while working on this document.
- Follow the instructions provided in each question to complete the code.
- DO NOT change the names of the variables that store your final answers.
- When in doubt about a question in the worksheet or your code, ask a TA or the instructor during office hours or on the discussion board.

Steps for submitting to MarkUs:

1. Go to MarkUs and log in using your UofT credentials.
2. Select Worksheet 2 from the assignment list.
3. Under Submissions, upload your Rmd file and select the file name from the list.

4. Go to Automated Testing and select Run Tests to check your worksheet answers.
5. You can submit as many times as you want, but only your latest submission before the deadline will be counted.
6. It is recommended you submit your file to MarkUs after you complete each activity to check your answers before moving on. You can submit multiple times to check your work, as your autograding tokens regenerate over time.

What to do if a test fails on MarkUs

1. Don't panic. Your work won't really be graded until the deadline, so start early to make sure you have lots of time to resolve issues before the deadline.
2. Read the message to get hints about what the problem is. For example "variable X not present" means that you may have a typo in your variable name.
3. Double check the instructions for each question to ensure you are entering an answer in the correct format.
4. Search on the discussion board to see if other classmates have encountered a similar error (and if not, consider posting a screenshot of the error message).
5. Come to TA or instructor office hours with your issue.

The due date for MarkUs Worksheet 2 is Tuesday, September 16, at 11:59pm

Activity 1 - Explore

1a) Load the data and create subgroups

Load in the dataset `MichelinNY.csv` into R. This dataset contains customer ratings (out of 30) on the `Food`, `Decor`, and `Service` characteristics of 164 restaurants, as well as whether or not the restaurant has a Michelin star (`InMichelin`). The response will by the `Price` of a dinner at these restaurants.

```
# run this chunk to load the data
nyc <- read.csv(file="MichelinNY.csv", header=T)
```

In Activity 1b), we will look at differences between Michelin and non-Michelin star restaurants, so we will split the dataset in two, with each half representing either the Michelin or non-Michelin star restaurants. To do this, we will only take the rows of `nyc` that correspond to `nyc$InMichelin == 1` (the Michelin star restaurants) and `nyc$InMichelin == 0` (the non-Michelin star restaurants).

NOTE: you'll want to treat the dataset like a matrix where we can talk about its dimension using the format [rows, columns], so to isolate only certain rows of the data, you would write nyc[nyc$InMichelin == 1,]

```
# Create a smaller dataset containing only the rows for Michelin stars
star <- NULL

# Create a smaller dataset containing only the rows of non-Michelin stars
nostar <- NULL
```

1b) Plot differences between Michelin stars and no stars

Now, you will check whether there are differences between star and no star restaurants, focusing on differences in the relationship between Price (response) and each numerical predictor (Food, Decor, and Service). The code to create the plots is already prepared. All you have to do is run it.

The blue datapoints correspond to Michelin star restaurants, and the black points are non-Michelin star restaurants.

Looking at the plots that are produced from running the above code chunk, do you think an interaction term might be needed to model the relationships present? If so, which numerical predictor should form an interaction term with `InMichelin`?

Enter one of the following numerical options: 1 for `Food`, 2 for `Decor` or 3 for `Service`

```
interaction_needed <- NULL
```

Question: What did you see in the above plots that led you to select that predictor?

TYPE YOUR ANSWER BELOW:

Activity 2 - Fitting a categorical-numerical interaction

2a) Fit your model from 1b)

We will assume that all ratings (Food, Decor and Service) are important for estimating the Price of a menu item at these restaurants. Fit a model using the `lm()` function that uses `Price` as the response, `Food`, `Decor` and `Service` as numerical predictors, `InMichelin` as a main effect term and include your chosen interaction term. You should use the syntax `variable1:variable2` to specify your interaction term.

```
# fit your model with all 4 predictors and your chosen interaction term
model2a <- NULL

# display your model coefficients
```

Which coefficient corresponds to the average change in Price of a non-Michelin star restaurant with fixed ratings for Decor and Service and a 1-rating increase in its Food rating? You should enter a numerical answer, rounded to 1 decimal place.

```
# enter the numerical value of the coefficient, rounded to 1 decimal
model2a_interpretation <- NULL
```

Question: What is the interpretation of the coefficient of the InMichelin main-effect term?

TYPE YOUR ANSWER BELOW:

2b) Making predictions

In order to see how R performs least squares with multiple predictors, we can extract the design matrix $\mathbf{X}$ from `model2a`. To do this, we can use the function `model.matrix()` and we just have to give it the name of the model we wish to look at (in this case `model2a`).

```
# extract the model matrix (i.e. design matrix X) for model 2a
Xmat_model2a <- NULL

# look at the top few rows of this matrix
head(Xmat_model2a)
```

```
## NULL
```

We see that we told R to create a new column for the interaction term by telling it which two predictors in our dataset to multiply together. This is helpful for using `model2a` to make a prediction.

Making a prediction occurs in 2 steps:

- first create a new small dataset `pred_data1` that contains the predictor values we wish to use in our prediction. We would create this dataset using the code `data.frame(Food = __, Decor = __, Service = __, InMichelin = __)` where we replace the underlined gaps with the values we wish to use.
- next, we use the `predict()` function which requires us to tell it which model to use for predictions, and which dataset holds the values of the predictors for prediction. The syntax for this function is `predict(model, newdata = dataset)`.

Predict the average Price of a dish at a non-Michelin star restaurant with a Food rating of 25, a Service rating of 20 and a Decor rating of 15 using `model2a`.

```
# create your dataframe storing predictor values we want to predict for
pred_data1 <- NULL

# make your prediction (write your predict function in place of NULL)
model2a_prediction1 <- NULL
```

Do the same again to create a small dataset `pred_data2` that contains the same predictors with the same values for Food, Service, and Decor, but now we consider a prediction for a restaurant with a Michelin star. Write the `predict()` function in place of `NULL` in `model2a_prediction2`.

```
# create your dataframe storing predictor values we want to predict for
pred_data2 <- NULL
```

```
# make your prediction (write your predict function in place of NULL)
model2a_prediction2 <- NULL
```

Question: Are your predictions consistent with the patterns you observed in the scatterplots in Activity 1? Why?

TYPE YOUR ANSWER BELOW:

Activity 3 - Removing InMichelin

3a) Remove the interaction

Interactions complicate interpretation of coefficients so it is sometimes worth checking if they are really needed. Create a new model `model3a` using the `lm()` function that uses Price as the response, and only Food, Decor, Service and InMichelin as main effects (i.e., no interaction).

Display the coefficients, then write the numerical value of the slope of Food, rounded to 1 decimal place.

```
# create a main effect model without the interaction
model3a <- NULL

# display coefficients


# write the numerical value of the slope of Food, rounded to 1 decimal place.
model3a_food_slope <- NULL
```

Question: Compare the slope of Food from model3a to the main effect of Food and interaction from model2a. What is the difference in what these two models tell you about the conditional effect of Food ratings on Price? (feel free to comment on the magnitude and direction of the effects from both models, as well as what the interaction term was doing originally).

TYPE YOUR ANSWER BELOW:

3b) Remove InMichelin altogether

Let's fit a final model `model3b` where we only use the numerical predictors in our model (i.e., Food, Decor, and Service). As before, the response will be Price.

Display the coefficients, then write the numerical value of the slope of Food, rounded to 1 decimal place.

```
# create a model without InMichelin
model3b <- NULL

# display coefficients


# write the numerical value of the slope of Food, rounded to 1 decimal place.
model3b_food_slope <- NULL
```

If you compare the estimated coefficients from all the models, you should notice that without the interaction term `model3a` and `model3b` have quite similar estimates.

Question: Looking at the estimates from all the models and the plots in Activity 1, explain why InMichelin seems to only make a substantial difference to the estimates when we include it as both a main effect and an interaction.

TYPE YOUR ANSWER BELOW:

END OF WORKSHEET - BE SURE TO SUBMIT YOUR WORKSHEET ON MARKUS TO RECEIVE COMPLETION CREDIT