STA130 Rstudio Homework

Problem Set 7

[Student Name] ([Student Number]), with Josh Speagle & Scott Schwartz

Instructions

Complete the exercises in this `.Rmd` file and submit your `.Rmd` and knitted `.pdf` output through Quercus by 11:59 pm E.T. on Thursday, March 16.

```
library(tidyverse)
```

Question 1: Multivariate Linear Regression and Mario Kart

In this question, you will revisit the Mario Kart data we looked at in this week's class. This data set contains eBay sales of the game Mario Kart for Nintendo Wii in October 2009 and is available in the `openintro` R package. We have provided a local csv copy, which we will load in below.

```
# load in data
mariokart <- read_csv("mariokart.csv")
glimpse(mariokart)
```

```
## Rows: 143
## Columns: 13
## $ ...1          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ id            <dbl> 1.50377e+11, 2.60483e+11, 3.20432e+11, 2.80405e+11, 1.70~
## $ duration      <dbl> 3, 7, 3, 3, 1, 3, 1, 1, 3, 7, 1, 1, 1, 1, 7, 7, 3, 3, 1,~
## $ n_bids        <dbl> 20, 13, 16, 18, 20, 19, 13, 15, 29, 8, 15, 15, 13, 16, 6~
## $ cond          <chr> "new", "used", "new", "new", "new", "new", "used", "new"~
## $ start_pr      <dbl> 0.99, 0.99, 0.99, 0.99, 0.01, 0.99, 0.01, 1.00, 0.99, 19~
## $ ship_pr       <dbl> 4.00, 3.99, 3.50, 0.00, 0.00, 4.00, 0.00, 2.99, 4.00, 4.~
## $ total_pr      <dbl> 51.55, 37.04, 45.50, 44.00, 71.00, 45.00, 37.02, 53.99, ~
## $ ship_sp       <chr> "standard", "firstClass", "firstClass", "standard", "med~
## $ seller_rating <dbl> 1580, 365, 998, 7, 820, 270144, 7284, 4858, 27, 201, 485~
## $ stock_photo   <chr> "yes", "yes", "no", "yes", "yes", "yes", "yes", "yes", "~
## $ wheels        <dbl> 1, 1, 1, 1, 2, 0, 0, 2, 1, 1, 2, 2, 2, 2, 1, 0, 1, 1, 2,~
## $ title         <chr> "~~ Wii MARIO KART &amp; WHEEL ~ NINTENDO Wii ~ BRAND NE~
```

Based on documentation in the data set, there are a handful of very high-priced items that were actually bundles of several games/items rather than just Mario Kart. Let's now filter these out.

```
# filter out bundles
mariokart2 <-
  mariokart %>%
  filter(total_pr < 100)
glimpse(mariokart2)
```

```
## Rows: 141
## Columns: 13
```

```
## $ ...1           <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
## $ id             <dbl> 1.50377e+11, 2.60483e+11, 3.20432e+11, 2.80405e+11, 1.70~
## $ duration       <dbl> 3, 7, 3, 3, 1, 3, 1, 1, 3, 7, 1, 1, 1, 1, 7, 7, 3, 3, 1,~
## $ n_bids         <dbl> 20, 13, 16, 18, 20, 19, 13, 15, 29, 8, 15, 15, 13, 16, 6~
## $ cond           <chr> "new", "used", "new", "new", "new", "new", "used", "new"~
## $ start_pr       <dbl> 0.99, 0.99, 0.99, 0.99, 0.01, 0.99, 0.01, 1.00, 0.99, 19~
## $ ship_pr        <dbl> 4.00, 3.99, 3.50, 0.00, 0.00, 4.00, 0.00, 2.99, 4.00, 4.~
## $ total_pr       <dbl> 51.55, 37.04, 45.50, 44.00, 71.00, 45.00, 37.02, 53.99, ~
## $ ship_sp        <chr> "standard", "firstClass", "firstClass", "standard", "med~
## $ seller_rating <dbl> 1580, 365, 998, 7, 820, 270144, 7284, 4858, 27, 201, 485~
## $ stock_photo    <chr> "yes", "yes", "no", "yes", "yes", "yes", "yes", "yes", "~
## $ wheels         <dbl> 1, 1, 1, 1, 2, 0, 0, 2, 1, 1, 2, 2, 2, 2, 1, 0, 1, 1, 2,~
## $ title          <chr> "~~ Wii MARIO KART &amp; WHEEL ~ NINTENDO Wii ~ BRAND NE~
```

(a) Sellers on eBay have the option to include a stock photo as the illustration of the product for sale. Does this choice affect the selling price? Carry out a **univariate (single-variable) linear regression analysis** and predict the mean selling price of the `total_pr` variable for sellers who do and do not use stock photos (`stock_photo`).

Hint: Your code from Question 4d in HW6 might be helpful here.

```
mean(predict(lm(total_pr ~ stock_photo, data = mariokart2)))
```

```
## [1] 47.43191
```

47.43191

(b) Sellers are rated by buyers on eBay, captured in the variable `seller_rating`. To simplify our analysis, we will categorize sellers by whether their rating is "low", "medium", or "high". Using `mutate()` and `case_when()`, create a new variable called `seller_rating_tier` that is "low" if `seller_rating <= 200`, "medium" if `200 < seller_rating <= 4500`, and "high" if `seller_rating > 4500`. Then, carry out a **linear regression analysis** to predict `total_pr` for the "low", "medium", and "high" levels of the new `seller_rating_tier` variable.

Hint: The syntax `lm(y ~ x)` will still work even if `x` is a multi-valued categorical explanatory variable.

```
mariokart3 <- mutate(mariokart2, seller_rating_tier = case_when(
seller_rating <= 200 ~ "low",
seller_rating > 200 & seller_rating <= 4500 ~ "medium",
seller_rating > 4500 ~ "high",
TRUE ~ "F"
))
mariokart3
```

```
## # A tibble: 141 x 14
##      ...1          id durat~1 n_bids cond  start~2 ship_pr total~3 ship_sp selle~4
##     <dbl>       <dbl>   <dbl>  <dbl> <chr>   <dbl>   <dbl>   <dbl> <chr>     <dbl>
##  1      1     1.50e11       3     20 new      0.99    4       51.6 standa~    1580
##  2      2     2.60e11       7     13 used     0.99    3.99    37.0 firstC~     365
##  3      3     3.20e11       3     16 new      0.99    3.5     45.5 firstC~     998
##  4      4     2.80e11       3     18 new      0.99    0       44   standa~       7
##  5      5     1.70e11       1     20 new      0.01    0       71   media       820
##  6      6     3.60e11       3     19 new      0.99    4       45   standa~  270144
##  7      7     1.20e11       1     13 used     0.01    0       37.0 standa~    7284
##  8      8     3.00e11       1     15 new      1       2.99    54.0 upsGro~    4858
##  9      9     2.00e11       3     29 used     0.99    4       47   priori~      27
```

```
## 10    10    3.30e11       7      8 used    20.0     4       50   firstC~     201
## # ... with 131 more rows, 4 more variables: stock_photo <chr>, wheels <dbl>,
## #   title <chr>, seller_rating_tier <chr>, and abbreviated variable names
## #   1: duration, 2: start_pr, 3: total_pr, 4: seller_rating
```

```
mean(predict(lm(mariokart3$total_pr ~ mariokart3$seller_rating_tier)))
```

```
## [1] 47.43191
```

```
summary(lm(mariokart3$total_pr ~ mariokart3$seller_rating_tier))
```

```
##
## Call:
## lm(formula = mariokart3$total_pr ~ mariokart3$seller_rating_tier)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.7898  -6.6688   0.2812   4.7402  25.2302
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           49.770      1.301  38.268   <2e-16 ***
## mariokart3$seller_rating_tierlow      -4.118      1.892  -2.177   0.0312 *
## mariokart3$seller_rating_tiermedium   -3.051      1.821  -1.676   0.0961 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.011 on 138 degrees of freedom
## Multiple R-squared:  0.03647,    Adjusted R-squared:  0.02251
## F-statistic: 2.612 on 2 and 138 DF,  p-value: 0.07703
```

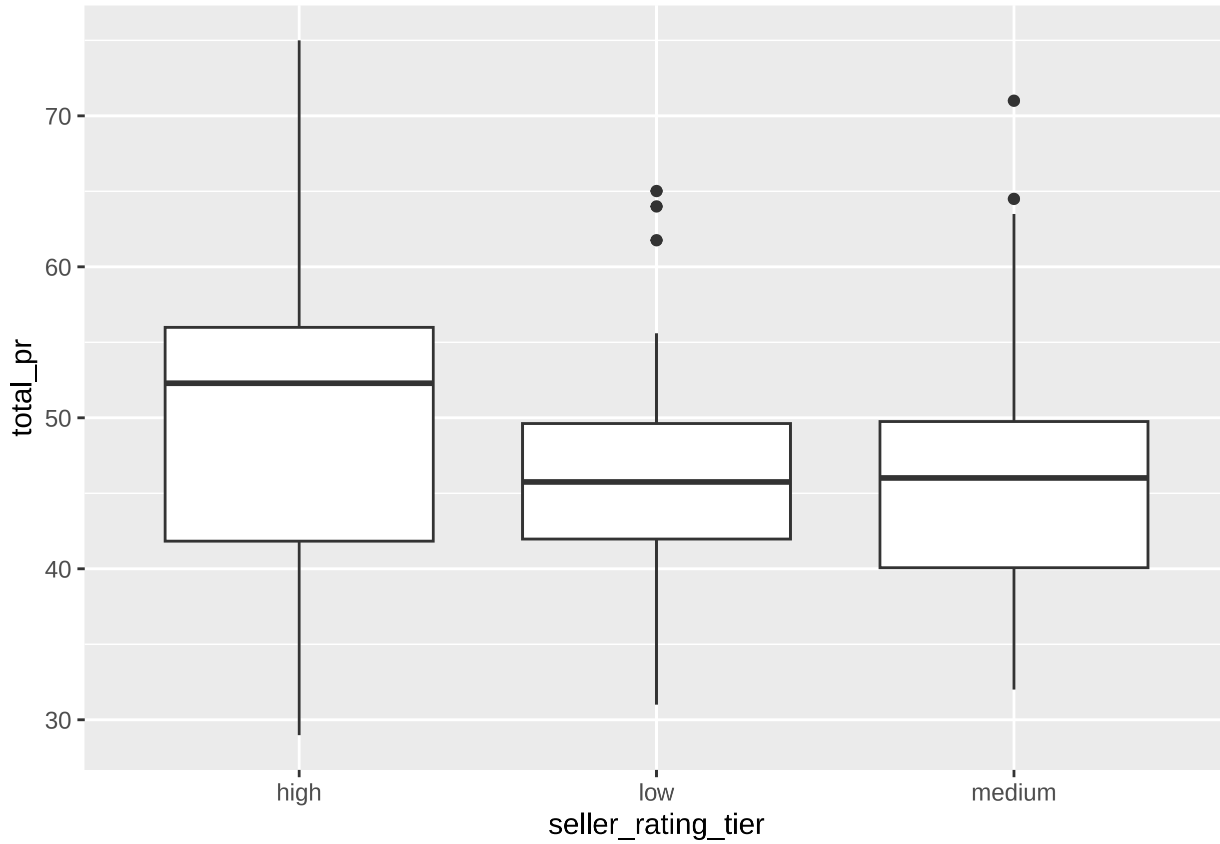
47.43191

How many indicator variables are in the model? Describe these indicator variables. Which seller rating group is `lm()` treating as the baseline category?

There are 2, condition and stock photo. The baseline category is high

(c) Create **boxplots** of `total_pr` for each category of seller based on `seller_rating_tier`.

```
mariokart3 %>%
  ggplot(aes(x = seller_rating_tier, y = total_pr)) + geom_boxplot()
```



Is this visualization consistent with your estimates from above? Why or why not might this be the case?

> *Pretty consistent as the ordering by price makes sense here. The outliers are a bit strange, however as I don't understand how that can happen here.*

(d) Now, perform an appropriate **multivariate regression analysis** including **interaction terms** to examine whether `seller_rating_tier` has an effect on the relationship between `total_pr` and `duration`.

Note that the full regression model here is:

$$\begin{aligned}\texttt{total\_pr}_i = &\ \beta_0 + \beta_1 \times \texttt{seller\_tier\_low}_i + \beta_2 \times \texttt{seller\_rating\_tier\_medium}_i + \beta_3 \times \texttt{duration}_i \\ &+ \beta_4 \times \texttt{seller\_rating\_tier\_low}_i \times \texttt{duration}_i + \beta_5 \times \texttt{seller\_rating\_tier\_medium}_i \times \texttt{duration}_i \\ &+ \epsilon_i\end{aligned}$$

Hint: The syntax for a multivariate interaction model is `lm(y ~ x1 + x2 + x1 * x2)`.

```
mariokart3$medium <- ifelse(mariokart3$seller_rating_tier == "medium", 1, 0)
mariokart3$low <- ifelse(mariokart3$seller_rating_tier == "low", 0, 1)
lm(total_pr ~ duration + seller_rating_tier+ seller_rating_tier * duration, data = mariokart3)
```

```
##
## Call:
## lm(formula = total_pr ~ duration + seller_rating_tier + seller_rating_tier *
##     duration, data = mariokart3)
##
## Coefficients:
```

```
##                       (Intercept)                         duration
##                            55.399                           -2.937
##             seller_rating_tierlow         seller_rating_tiermedium
##                            -8.186                           -2.388
##    duration:seller_rating_tierlow  duration:seller_rating_tiermedium
##                             2.620                            1.539
```

What is the equation of the fitted regression line for sellers with low ratings?

> *lm(total_pr ~ duration + seller_rating_tier0,1+ seller_rating_tier* duration, data = mariokart3)*

What is the equation of the fitted regression line for sellers with medium ratings?

> *lm(total_pr ~ duration + seller_rating_tier1,0+ seller_rating_tier*

What is the equation of the fitted regression line for sellers with high ratings?

> *lm(total_pr ~ duration + seller_rating_tier0,0+ seller_rating_tier*

(e) Produce an appropriate plot to visualize the fitted relation.

Hint: Your code from Problem 2d in HW6 might prove useful here.

```
#ggplot(mariokart3) + aes(x = (duration + seller_rating_tier+ seller_rating_tier * duration), y =total_
```

Does the seller rating tier appear to modify the association between duration and total price? Write 1-2 sentences explaining your answer.

> *REPLACE THIS TEXT WITH YOUR ANSWER*

Question 2: Predictions and Model Comparison

(a) Divide the data into **testing** and **training** data sets that include 30% and 70% of the data, respectively. Then, fit multivariate linear regression models for the total price `total_pr` using the following combinations of variables (**"features"**) as predictors with training data only:

i. `stock_photo`
ii. `stock_photo`, `duration`, and their interaction
iii. `seller_rating`
iv. `stock_photo`, `seller_rating`, and their interaction
v. `stock_photo`, `seller_rating`, `duration`, and all interaction terms

Hint: There are a number of approaches for computing the training/testing splits here. One possibility is you can random sample some fraction `X` *of the input data using the* `sample_frac()` *function and then subsequently select the remaining data that has not been sampled using the* `anti_join()` *function.*

```
set.seed(130) # use this seed to make your analysis reproducible

dataset <- mariokart3

# Divide the dataset into training and testing sets
training_indices <- sample(nrow(dataset), round(0.7 * nrow(dataset)), replace = FALSE)
train <- dataset[training_indices, ]
test <- dataset[-training_indices, ]
model1 <- lm(total_pr ~ stock_photo, data = train)
```

```r
model2 <- lm(total_pr ~ stock_photo * duration, data = train)
model3 <- lm(total_pr ~ seller_rating, data = train)
model4 <- lm(total_pr ~ stock_photo * seller_rating, data = train)
model5 <- lm(total_pr ~ stock_photo * seller_rating * duration, data = train)
```

(b) Calculate the **root-mean-square-error (RMSE)** for each of the five models from part (a) over both the training and testing datasets (10 values in total) and save the results in a tibble with columns named `model`, `rmse_train`, and `rmse_test`.

As a reminder, for a given response with observed values $y_1, \ldots, y_n$ and corresponding predicted values (from the above models) of $\hat{y}_1, \ldots, \hat{y}_n$, the RMSE is defined as:

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

Hint: You can use the syntax ***`train_linear_model %>% predict(test_data)`*** *to generate predictions to new data values. You can also store your models in a list using the syntax* ***`list(model1, model2, ...)`*** *and access them using the syntax* ***`list[[i]]`***.

```r
rmse_list <- list()

# Create a function to calculate the RMSE
rmse <- function(actual, predicted) {
  sqrt(mean((actual - predicted)^2))
}

# Model 1: stock_photo
# Train the model
model1 <- lm(total_pr ~ stock_photo, data = train)
# Calculate the RMSE for the training data
rmse_train1 <- rmse(train$total_pr, predict(model1, newdata = train))
# Calculate the RMSE for the testing data
rmse_test1 <- rmse(test$total_pr, predict(model1, newdata = test))
# Store the results in the list
rmse_list[[1]] <- c("Model 1", rmse_train1, rmse_test1)

# Model 2: stock_photo, duration, and interaction
# Train the model
model2 <- lm(total_pr ~ stock_photo * duration, data = train)
# Calculate the RMSE for the training data
rmse_train2 <- rmse(train$total_pr, predict(model2, newdata = train))
# Calculate the RMSE for the testing data
rmse_test2 <- rmse(test$total_pr, predict(model2, newdata = test))
# Store the results in the list
rmse_list[[2]] <- c("Model 2", rmse_train2, rmse_test2)

# Model 3: seller_rating
# Train the model
model3 <- lm(total_pr ~ seller_rating, data = train)
# Calculate the RMSE for the training data
rmse_train3 <- rmse(train$total_pr, predict(model3, newdata = train))
# Calculate the RMSE for the testing data
rmse_test3 <- rmse(test$total_pr, predict(model3, newdata = test))
```

```r
# Store the results in the list
rmse_list[[3]] <- c("Model 3", rmse_train3, rmse_test3)

# Model 4: stock_photo, seller_rating, and interaction
# Train the model
model4 <- lm(total_pr ~ stock_photo * seller_rating, data = train)
# Calculate the RMSE for the training data
rmse_train4 <- rmse(train$total_pr, predict(model4, newdata = train))
# Calculate the RMSE for the testing data
rmse_test4 <- rmse(test$total_pr, predict(model4, newdata = test))
# Store the results in the list
rmse_list[[4]] <- c("Model 4", rmse_train4, rmse_test4)

# Model 5: all predictors and interactions
# Train the model
model5 <- lm(total_pr ~ stock_photo * seller_rating * duration, data = train)
# Calculate the RMSE for the training data
rmse_train5 <- rmse(train$total_pr, predict(model5, newdata = train))
# Calculate the RMSE for the testing data
rmse_test5 <- rmse(test$total_pr, predict(model5, newdata = test))
# Store the results in the list
rmse_list[[5]] <- c("Model 5", rmse_train5, rmse_test5)

# Combine the results into a tibble
rmse_df <- as.data.frame(do.call(rbind, rmse_list))
names(rmse_df) <- c("model", "rmse_train", "rmse_test")
rmse_df
```

```
##     model       rmse_train        rmse_test
## 1 Model 1 9.37056559572717 7.70904420805139
## 2 Model 2   8.664983620051 7.33891654580419
## 3 Model 3 9.56425747568213 7.62567039159108
## 4 Model 4 9.32234787052574 7.66272426378562
## 5 Model 5 7.95474811878323 6.95552301095432
```

(c) Based on the results in part (b), write 1-2 sentences discussing which model would you prefer to use for future predictions and why.

> *Model 5 as it is the lowest*

(d) (Optional but strongly encouraged) Make a histogram and boxplot showcasing the distribution of the **effect sizes** over the test data for your preferred model from part (c). As a reminder, the effect size e_{ij} for object i with explanatory variable(s) $x_i \times z_i$ and coefficient $j > 0$ is defined as

$$e_{ij} = \beta_j \times (x_i \times z_i)$$

such that our linear regression model can be rewritten as

$$y_i = \beta_0 + \sum_{j=1}^{m} e_{ij} + \epsilon_i$$

```
# code you answer here
```

Write 1-2 sentences interpreting your results.

REPLACE THIS TEXT WITH YOUR ANSWER