Assignment 2, Spring 2021

Tuan Nguyen

01/30/2021

A few important notes

Option 1 for submitting your assignment: *This method is actually preferred. This is an RMarkdown document. Did you know you can open this document in RStudio, edit it by adding your answers and code, and then knit it to a pdf? To submit your answers to this assignment, simply knit this file as a pdf and submit it as a pdf on Forum. All of your code must be included in the resulting pdf file, i.e., don't set echo = FALSE in any of your code chunks. This is a cheat sheet for using Rmarkdown. If you have questions about RMarkdown, please post them on Piazza. Try knitting this document in your RStudio. You should be able to get a pdf file. At any step, you can try knitting the document and recreate a pdf. If you get an error, you might have incomplete code.*

Option 2 for submitting your assignment: *If you are not comfortable with RMarkdown, you can also choose the Google Doc version of this assignment, make a copy of it and edit the Google doc (include your code, figures, results, and explanations) and at the end download your Google Doc as a pdf and submit the pdf file.*

Note: *Either way (if you use Rmd and knit as pdf OR if you use Google Doc and download as pdf) you should make sure you put your name on top of the document.*

Note: *The first time you run this document you may get an error that some packages don't exist. If you don't have the packages listed on top of this document, install them first and you won't get those errors.*

Note: *Don't change seed in the document. The function* `set.seed()` *has already been set at the beginning of this document to 928. Changing the see again to a different number will make your results not replicable.*

QUESTION 1: Data Generating Example

STEP 1 Create a set of 1000 outcome observations using a data-generating process (DGP) that incorporates two variables and a stochastic component (of your choice). In other words, create two independent variables, a vector of noise, and a dependent variable (outcome) that relates to the independent variables and the noise with a formula you choose.

```r
# Your code here
#create 1000 observations for each variable
#create independent variable called finished_acad_task and time_social
time_social <- sample(c(0:8),size=1000,replace=TRUE)
finished_acad_task <- sample(c(0:10),size = 1000, replace = TRUE)
#chosen formula
productivty_level <- 10*finished_acad_task - 2*time_social + 2.5*rnorm(1000,0,0.33)
#create dataframe.
df <- data.frame(time_social,finished_acad_task,productivty_level)
```

STEP 2 Tell a 2-3 sentence story about the data generating process you coded up above. What is it about and what each component means?

I used a sample built-in function to randomly select different values from vector c(1:8), and specified the number of outcome observations with size = 1000.Similarly, I created 1000 outcome observations for the number of finished academic tasks. The dependent variable is the productivity_level, which can be measured in scores.

STEP 3 Fit a regression model of the outcome on the two independent variables and see if the coefficients you find are similar to the ones in your DGP.

```
# Your code here
fit <- lm(productivty_level~time_social+finished_acad_task,data =df )
fit
```

```
##
## Call:
## lm(formula = productivty_level ~ time_social + finished_acad_task,
##     data = df)
##
## Coefficients:
##        (Intercept)         time_social  finished_acad_task
##            -0.0866             -1.9887             10.0107
```

STEP 4 Use the simulation-based approach covered in class (the arm library, etc.) to find the computational 95% confidence interval of your coefficients and report them here. Set the number of simulations to 10,000.

```
library(arm)
sim_fit <- sim(fit, n.sims = 10000)
coef_time_on_social <-coef(sim_fit)[, 2]
coef_on_acad_task <-coef(sim_fit)[, 3]
conf_int_social <- quantile(coef_time_on_social,probs = c(0.025, 0.975))
conf_int_acad_task <- quantile(coef_on_acad_task,probs = c(0.025, 0.975))

# Your code here
#CI for time on social media
conf_int_social
```

```
##      2.5%     97.5%
## -2.009561 -1.967977
```

```
#CI for the number of academic tasks
conf_int_acad_task
```

```
##      2.5%     97.5%
##  9.994466 10.026935
```

STEP 5 Now, estimate the 95% confidence interval for the predicted outcome when your first variable is equal to 1 and the second variable is equal to -2 using the simulated coefficients you found in Step 4.

```
# Your code here
df1 = data.frame(coef(sim_fit))
colnames(df1) = c("intercept", "time_social", "finished_acad_task")
df1$pred <- df1$intercept + df1$time_social * 1 + df1$finished_acad_task * -2
conf_int_pred <- quantile(df1$pred,probs = c(0.025, 0.975))
conf_int_pred
```

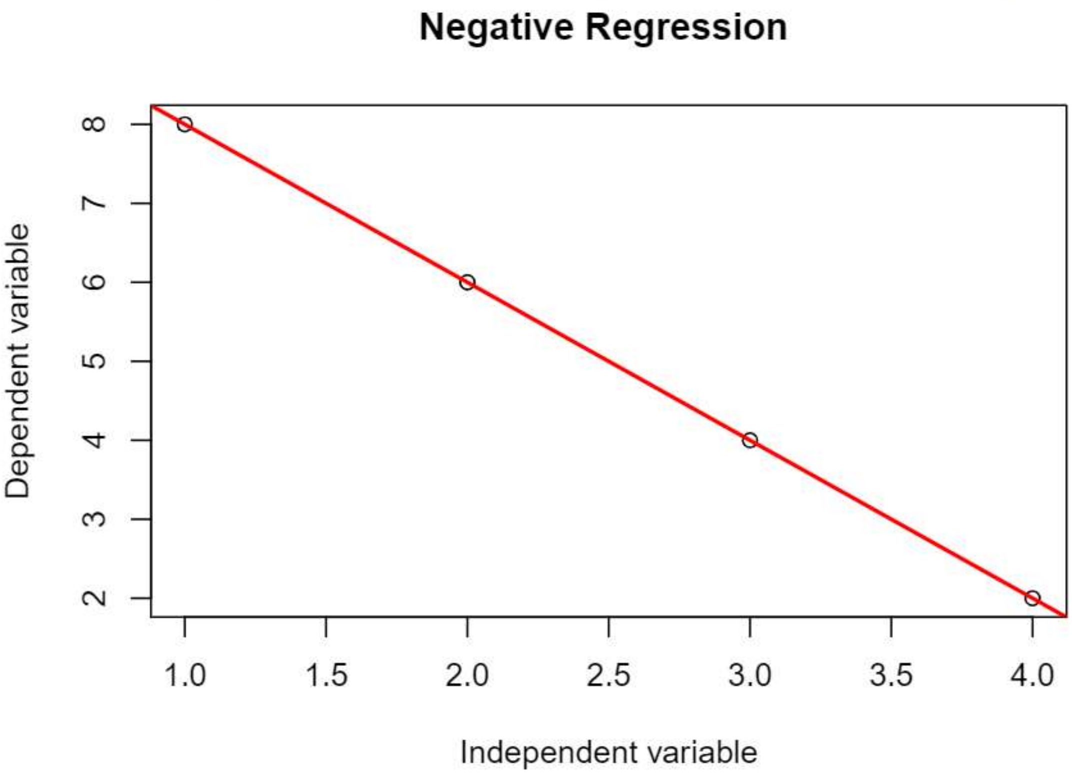
```
##      2.5%     97.5%
## -22.23417 -21.96151
```

QUESTION 2: Outliers

Imagine that you want to create a data visualization that illustrates the sensitivity of regression to outlier data points. So, you want to create two figures:

One figure that shows a regression line fit to a 2-dimensional (x and y) scatterplot, such that the regression line clearly has a negative slope.

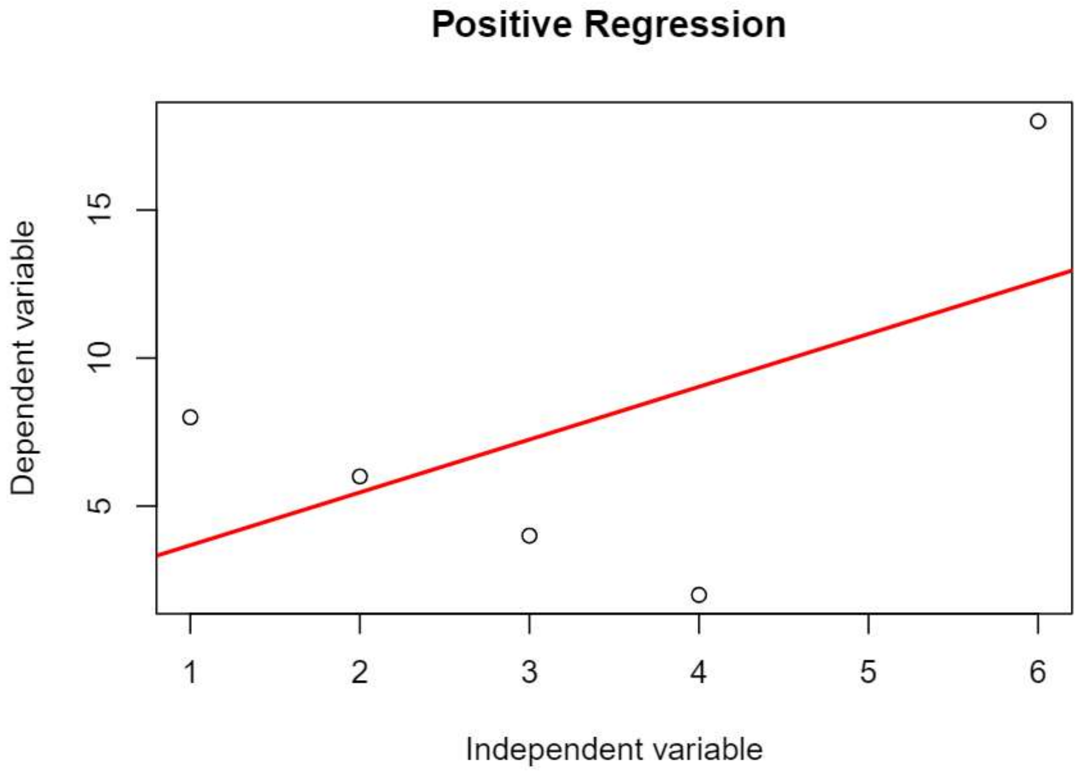
```
c1 <- c(1,2,3,4)
c2 <- c(8,6,4,2)
lm1 <- lm(c2~c1)
plot(c1, c2,xlab="Independent variable",ylab = "Dependent variable", main ="Negative Regression ") + ab
```



```
## integer(0)
```

And, another figure that shows a regression line with a positive slope fit to a scatter plot of the same data plus one additional outlier data point. This one data point is what changes the sign of the regression line's slope from negative to positive.

```
# Your code here!
c3 <- c(1,2,3,4,6)
c4 <- c(8,6,4,2,18)
lm1 <- lm(c4~c3)
plot(c3, c4,xlab="Independent variable",ylab = "Dependent variable", main ="Positive Regression ") + ab
```



```
## integer(0)
```

Be sure to label the axes and the title the figures appropriately. Include a brief paragraph that explains the number of observations and how you created the data set and the outlier. ##....For the first figure: I created two vectors of numbers. First vector contains increasing numbers, and the second vector contains decreasing numbers. To change the slope of the regression line, I added one outlier to the dataset. The outlier is (6,18), which has a complete different trend than other data points.The higher value of independent variable, the higher value it becomes for dependent variable. The total observations in the first and second set is 4. The total observations for the 3rd and 4th set is 5.

QUESTION 3: Simulation methods

You were hired by the local animal shelter at Austin, Texas to perform some data analysis. They are particularly interested in predicting how long an animal stays in a shelter given the information they have upon shelter admission. To focus on your task, you'll be only looking at dogs.

They identified the following variables as relevant to the prediction:

- `Intake Type`: dogs can come into the shelter in multiple ways. They are interested in differentiating between three main ones: *strays* (most common: animals found without an owner by animal control), *owner surrenders* (animals brought to the shelter by their owners for various reasons), and *public assistance* (animals surrender by a person who is not their owner). The other type of intake is euthanasa request (made by the owner), which you should ignore.
- `Intake Condition`: there are three conditions the shelter wants to mark — dogs identified as *Injured*, *Sick*, and *Nursing/Pregnant*. There are other conditions inserted in this field, but for dogs, they are not common.
- `Age` (use `age_upon_intake_years`): how old was the dog, in years, when arriving at the shelter.
- `Breed`: this is a fairly unreliable field, since people are not always successful at identifying the actual genetic breed of a dog. Nevertheless, you should try to look at a crude division: those identified as

purebred compared to mixed breeds.

The variable containing the length of day, which you want to predict, is `time_in_shelter_days`.

STEP 1 Download the dataset from here. Make sure you use `aac_intakes_outcomes.csv` which contains the joint intake and outcome data for all animals.

Then, perform the necessary preprocessing to be able to fit the model as required:

```
# Your code here
#1. Remove `NA`s or empty entries
data <-read.csv("aac_intakes_outcome.csv")
#important columns
data.columns <- c(23,26,27,31,41)
#v <- c("intake_type","breed","age_upon_intake_.years","intake_condition")
for(i in data.columns)
  {
  which_values_are_missing <- which(as.character(data[, i]) == "")
  data[which_values_are_missing, i] <- NA
}
new_data = na.omit(data)
#2. Remove all entries except for dogs
new_data_1 <- new_data[new_data$animal_type=="Dog",]

#3. Use the `janitor` package and the `clean_names` function to convert all column names to all lowerca
library(janitor)
new_data_1 <- clean_names(new_data_1)
colnames(new_data_1)
```

```
##  [1] "age_upon_outcome"           "animal_id_outcome"
##  [3] "date_of_birth"              "outcome_subtype"
##  [5] "outcome_type"               "sex_upon_outcome"
##  [7] "age_upon_outcome_days"      "age_upon_outcome_years"
##  [9] "age_upon_outcome_age_group" "outcome_datetime"
## [11] "outcome_month"              "outcome_year"
## [13] "outcome_monthyear"          "outcome_weekday"
## [15] "outcome_hour"               "outcome_number"
## [17] "dob_year"                   "dob_month"
## [19] "dob_monthyear"              "age_upon_intake"
## [21] "animal_id_intake"           "animal_type"
## [23] "breed"                      "color"
## [25] "found_location"             "intake_condition"
## [27] "intake_type"                "sex_upon_intake"
## [29] "count"                      "age_upon_intake_days"
## [31] "age_upon_intake_years"      "age_upon_intake_age_group"
## [33] "intake_datetime"            "intake_month"
## [35] "intake_year"                "intake_monthyear"
## [37] "intake_weekday"             "intake_hour"
## [39] "intake_number"              "time_in_shelter"
## [41] "time_in_shelter_days"
```

```
#4. Remove rows with the *Euthanasia Request* intake type. Make sure your `intake_type` column used for
new_data_2 <-new_data_1[new_data_1$intake_type!="Euthanasia Request",]

#5. Create a new `intake_condition` feature that satisfies the shelter's interests: dogs marked *normal
#duplicate the intake_type column
```

```
new_data_2$intake_type <- as.factor(new_data_2$intake_type)
new_data_2$new_intake_condition <- new_data_2$intake_condition
#for loop to convert rows with specific values
for(i in 1:length(new_data_2$new_intake_condition))
  {
  if (new_data_2$new_intake_condition[i] == "Normal"|new_data_2$new_intake_condition[i] == "Sick"|new_d
    {
    next
  } else if (new_data_2$new_intake_condition[i] == "Nursing"|new_data_2$new_intake_condition[i] == "Pre
     new_data_2$new_intake_condition[i] <- "Nurspreg"
  } else{
      new_data_2$new_intake_condition[i]<- "Other"
    }

}
```

STEP 2 Run a linear regression that models `time_in_shelter_days` as a function of `intake_type`, `intake_condition_new` (created in Step 1), and `age_upon_intake_years`.

```
# Your code here!
lm1 <- lm(time_in_shelter_days~intake_type+new_intake_condition+age_upon_intake_years, data = new_data_
summary(lm1)
```

```
##
## Call:
## lm(formula = time_in_shelter_days ~ intake_type + new_intake_condition +
##     age_upon_intake_years, data = new_data_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -34.45  -13.34   -8.98   -3.96 1589.07
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   27.80835    1.17903  23.586  < 2e-16 ***
## intake_typePublic Assist      -4.99086    0.83916  -5.947 2.74e-09 ***
## intake_typeStray              -5.52506    0.53230 -10.380  < 2e-16 ***
## new_intake_conditionNormal    -9.41483    1.07657  -8.745  < 2e-16 ***
## new_intake_conditionNurspreg   3.37302    2.08328   1.619   0.1054
## new_intake_conditionOther     -6.78091    2.64234  -2.566   0.0103 *
## new_intake_conditionSick     -11.36625    1.78506  -6.367 1.94e-10 ***
## age_upon_intake_years          0.53189    0.07152   7.437 1.05e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.36 on 45176 degrees of freedom
## Multiple R-squared:  0.00631,    Adjusted R-squared:  0.006156
## F-statistic: 40.98 on 7 and 45176 DF,  p-value: < 2.2e-16
```

STEP 3 Report coefficients and R-squared. Which coefficients are statistically significant? Interpret what they mean in terms of the feature they are associated with and the outcome.

Then calculate R-squared by hand (as in, computing the quantities that make up R-squared rather than using the built-in one) and show that you get the same or nearly the same answer as the summary `lm` command.

Write out hand calculations here.

As shown in the table, we can see that except for new_intake_conditionNurspreg, all other varaibles are significantly important. Variable new_intake_conditionNurspreg is not significant is because its t-value > 0.05. The negative slope of intake_type such as Public Assit and Stray are negative, which means that there are some negative correlation between intake_type versus time_in_shelter_days. In constrast, age_upon_intake_years has a positive relationship with time_in_shelter_days.

In general, the main purpose of linear regression is its ability to show impact of different variables on subject of interest. Categorial variables take on binary values such as 0 and 1, which is useful for calculation. The value of R-squared is 0.00631.

```
avg_time <- mean(new_data_2$time_in_shelter_days)
pred_time <- predict(lm1,data=new_data_2)

r1 <- c()
r2 <- c()
for (i in 1:nrow(new_data_2)){
  r1 <-  c(r1,(new_data_2$time_in_shelter_days[i] - pred_time[i])^2)
  r2 <-  c(r2,(new_data_2$time_in_shelter_days[i] - avg_time)^2)
}

r_squared <- (1- sum(r1)/sum(r2))
r_squared
```

```
## [1] 0.006309819
```

The result above gives the same result as the original calculation. R-squared seems to small, which suggests that the model did a bad job in terms of fitting the data. Therefore, we can further suggest other improvisations to the model. #### STEP 4

The shelter is specifically interested in the effect of age upon intake on the length of stay. Set all predictors at the following values: `intake type` = stray and `intake_condition` = normal. Use them to create a data visualization that shows the 95% confidence interval of the expected values of `time_in_shelter_days` as `age_upon_intake_years` varies between the following values: 0.1, 0.5, 1, 2, 3, 5, 7, 10, 12, 15, 18, 20.

Follow this procedure using a simulation-based approach:

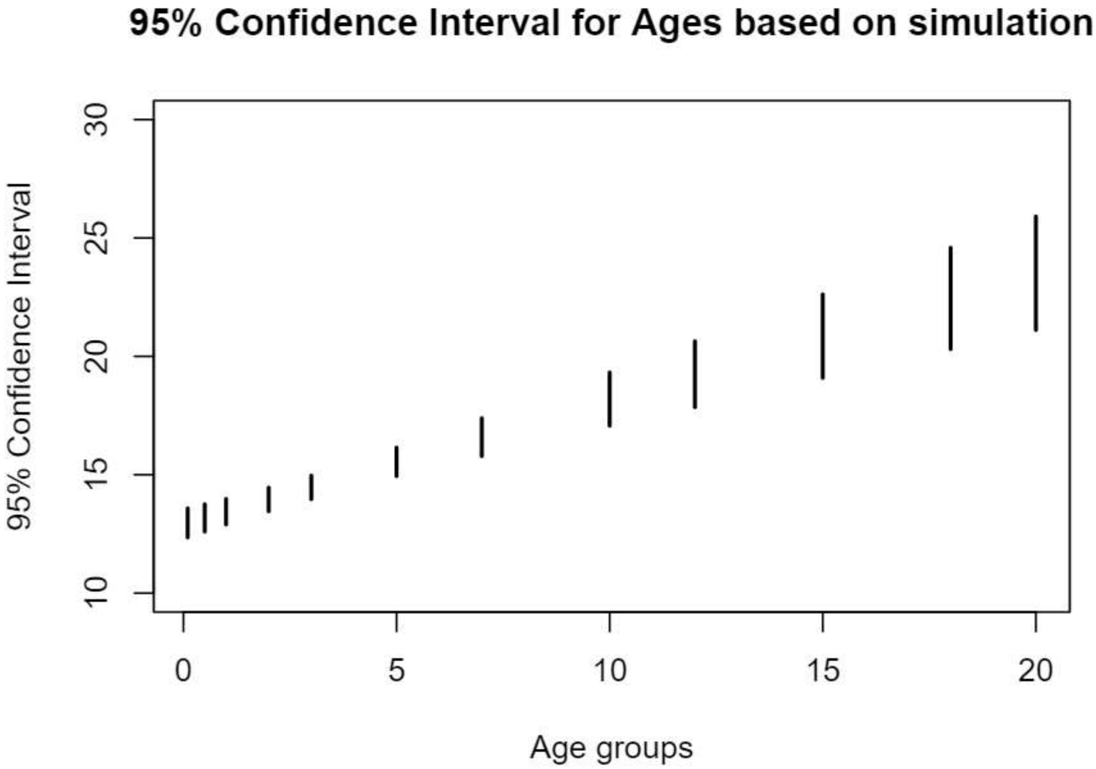
1. Generate 1,000 sets of coefficients from your regression model.
2. Generate 1,000 predictions with each value of `age`.
3. Obtain 95% confidence intervals from these predictions.
4. Plot your results.

Be sure to include axes labels and figure titles.

```
# Your code here!
library(arm)
new_data_3 <- new_data_2
new_data_3$intake_type <- "stray"
new_data_3$intake_condition<-"normal"
#lm2 <- lm(time_in_shelter_days ~ age_upon_intake_years, data = new_data_2)
sim_fit01 <- sim(lm1,n.sims=1000)
ages <- c(0.1, 0.5, 1, 2, 3, 5, 7, 10, 12, 15, 18, 20)
storage_matrix <- matrix(NA,nrow=1000,ncol =length(ages))

get_predicted <- function(coef,age) {
  return(coef[1]+coef[3]*1+coef[4]*1+coef[8]*age)
```

```
}
for (m in 1:length(ages)){
  for(i in 1:1000){
    storage_matrix[i,m] <- get_predicted(sim_fit01@coef[i, ],ages[m])
  }
}
conf.interval <- apply(storage_matrix,2,quantile,probs = c(0.025,0.975))
plot(x =ages, y = ages,type  = "n",ylim = c(10,30),xlab ="Age groups", ylab ="95% Confidence Interval",
for(i in 1:length(ages)) {
  segments(
    x= ages[i],
    y0 = conf.interval[1,i],
    x1 =ages[i],
    y1 = conf.interval[2,i],
    lwd =2)
}
```



STEP 5 Write a short paragraph with your reflections on this exercise:

1. What are the top 1-2 insights that the shelter can learn from your regression results and data visualization? Firstly, the dog's age when taking in correlates with the time he/she stays in shelter. It means the the older the dog when it enters the shelter, the longer it would stay. Based on the model, the prediction for length of stay of dogs is dependent on different variables such as input_type, input_condition, and its age. The prediction seems vague, and the length of stay can be estimated by using other variables. Looking at the data and summary of lm1, we can see that length of stay is negatively correlated if the dog's condition is identified as normal, other, and sick, except for nursing or pregnant.
2. How does the prediction for length of stay changes with age, and how confident is that prediction? The older the dog, the longer it takes him to stay in shelter. In addition, the level of uncertainty

when predicting the length of stays for dog changes with different ages. The older the dog, the higher uncertainty shown in the model. ## QUESTION 4: Different regressions in an RCT on the same set of data

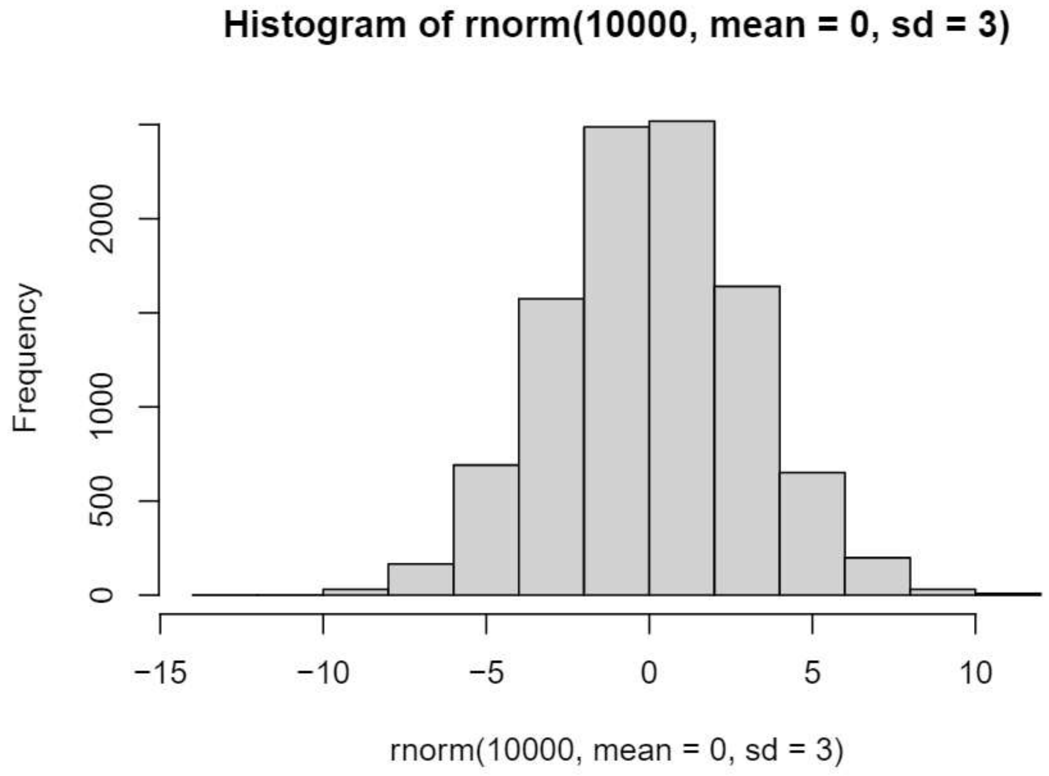
```r
# creating the following simulated toy data set
set.seed(1234)

# ed1 is the value of spend on education prior to the experiment
ed1 <- round(runif(10000, min = 8, max = 20))
mean(ed1)
```

```
## [1] 14.0035
```

```r
# treat is an indicator of RANDOM ASSIGNMENT to treatment vs. control (RCT)
# 5000 control units and 5000 treated units
treat <- c(rep(0, 5000), rep(1, 5000))

# here's what the error term looks like
hist(rnorm(10000, mean = 0, sd = 3))
```

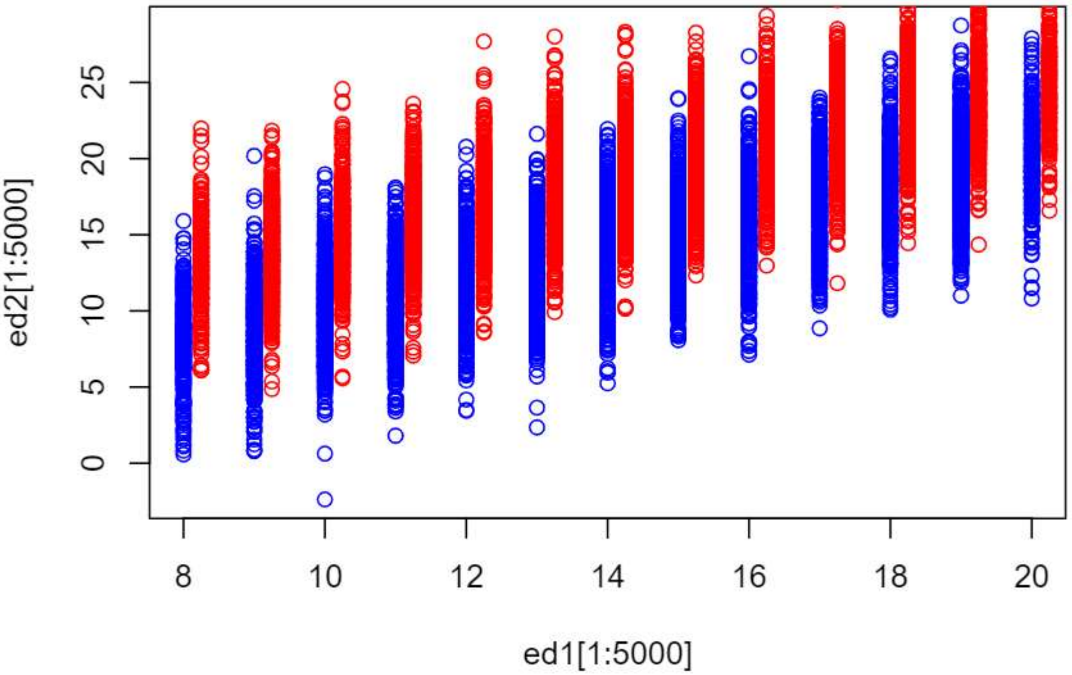


```r
error_term <- rnorm(10000, mean = 0, sd = 3)

# let us assume the data-generating process below (no intercept)
ed2 <- 1*ed1 + 5*treat + error_term

# the 'gain score' or 'diff' in educ spend across periods
ed3 <- ed2 - ed1

# plot the data
plot(ed1[1:5000], ed2[1:5000], col = "blue")
points(ed1[5001:10000] + 0.25, ed2[5001:10000], col = "red")
```



```
# confirm the avg tmt effect is as expected
mean(ed2[5001:10000] - mean(ed2[1:5000]))
```

```
## [1] 5.034747
```

```
# consider various regressions--note: here, we know correct model and the right answer
lm1 <- lm(ed2 ~ treat)
lm2 <- lm(ed2 ~ treat + ed1)
lm3 <- lm(ed3 ~ treat)
lm4 <- lm(ed3 ~ treat + ed2)
lm5 <- lm(ed2 ~ treat + ed1 + I(treat*ed1))
lm3
```

```
##
## Call:
## lm(formula = ed3 ~ treat)
##
## Coefficients:
## (Intercept)        treat
##    -0.05906      5.05655
```

```
lm5
```

```
##
## Call:
## lm(formula = ed2 ~ treat + ed1 + I(treat * ed1))
##
## Coefficients:
##    (Intercept)           treat             ed1  I(treat * ed1)
##       -0.50860         5.34179         1.03208        -0.02034
```

1. Answer the questions below after running this code
2. Which regression specification(s) correctly identifies the data generating process?

ed2 <- 1*ed1 + 5*treat + error_term is the data generation process. Therefore, based on the model, we can see that lm2 correctly identify the data generating process. In addition, lm5 also shows its similar form except that it included the interaction between ed1 and treat.

3. Which regression specification(s) reliably estimate(s) the treatment effect? Treatment effect is the difference in potentials outcomes compared between two groups, which are treatment and control group. ed3- is the treatment effect in this case. lm3 is the most reliable model to estimate the treatment effect.
4. What does the coefficient in the interaction term in lm5 imply? Is that implication accurate? The coefficient in the interaction term in lm5 implies that treat and ed1 are dependent on each other. However, it raises concern because the implication seems unreasonable. As shown in lm5, the interaction term is almost 0. Therefore, using the interaction term does not help the model.
5. Why does lm1 deliver the intercept that it does? As we see, the code uses runif() with min =8 and max = 20 to create ed1. Therefore, it has an uniform distribution of values in range of 8 10 20. lm1's intercept is the mean of ed1, approximates around 14. With the formula of lm1 is ed2 = 13.995 + 5.035*treat. lm1 does not include variable ed1. In the end, it minimizes rss by taking intercept as the mean of ed1.
6. Which is your favorite regression specification, and why, given whatever it is that you are trying to learn from the regression (which is up to you!)? Which regression specification do you deem the worst, and why? lm2 seems to be my favorite because it mostly produces the close result we want. lm4 is my least favorite because its variables are dependent on each other. In addition, no interaction term is included in the model, making it the most likely model to produce biased result.

QUESTION 5: A Classification problem

Beyond the time spent in the shelter, another important prediction question for the shelter is the outcome of the animal. Shelters typically measure themselves according to their 'live release rate': the fraction of animals with a live outcome out of all animals that arrived at the shelter. In this section, you'll use similar features to predict a dog's outcome, rather than the time spent at the shelter.

For that purpose, you'll have to create a new binary feature called `live_release`. We shall define live release as having all but the following `outcome_type` values: Died, Disposal, Euthanasia, Missing.

You should maintain the same exclusions as the previous sections, since euthansia requests may (although this varies between shelters) not count towards the live release rate. If you're wondering why, it's because some euthanasia requests don't actually have to end with euthanasia. Check how many of the 180 euthansia requests for dogs in this dataset were turned around.

STEP 1 Create a `live_release` column based on the specification above. Then, choose any other feature in the dataset that you think might be interesting to include in the model, i.e., that you want to see whether it has any association with a dog's live release chances. This may definitely include some additional feature engineering on your behalf (which can be as simple as choosing something like 'black_color' using the color feature). List your features here.

```
new_data_2$live_release <- new_data_2$outcome_type
for ( i in 1: length(new_data_2$live_release)) {
  if(new_data_2$outcome_type[i]=="Missing"|new_data_2$outcome_type[i]=="Died"|new_data_2$outcome_type[i
    new_data_2$live_release[i]<- 0
  } else{
    new_data_2$live_release[i] <- 1
  }
}
new_data_2$live_release = as.numeric(new_data_2$live_release)
#new_data_2$intake_type = as.factor(new_data_2$intake_type)
#new_data_2$intake_condition = as.factor(new_data_2$intake_condition)
```

```
#new_data_2$breed = as.factor(new_data_2$breed)
#head(new_data_2$live_release)
#Features include color
#There are many relevant features such as breed, age_upon_intake, and sex_upon_intake.
```

STEP 2 This time, we want to also be able to see how good our predictions are on unseen data. While there are several variations of the k-fold cross-validation method, let's stick with the simplest one where we just split randomly the dataset into a training and a testing (aka validation) set.

Randomly select 80% of the data to be put in a training set and leave the rest for a test set.

```
# Your code here!
test_ind <- sample(1:nrow(new_data_2), round(nrow(new_data_2)/5), replace = FALSE)
test_set <- new_data_2[test_ind,]
train_set <- new_data_2[-test_ind,]
```

STEP 3 Using your training set (only!), run a logistic regression, modeling `live_release` as a function of `intake_type`, `intake_condition`, and `age_upon_intake_years`, and your other features. Report and interpret the regression coefficient and 95% confidence intervals for `age_upon_intake`.

```
unique(new_data_2$intake_type)
```

```
## [1] Stray           Public Assist   Owner Surrender
## Levels: Owner Surrender Public Assist Stray
```

```
glm1<- glm(live_release~intake_type+intake_condition+age_upon_intake_years+sex_upon_intake,family = "bi
summary(glm1)
```

```
##
## Call:
## glm(formula = live_release ~ intake_type + intake_condition +
##     age_upon_intake_years + sex_upon_intake, family = "binomial",
##     data = train_set)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1676   0.1663   0.1839   0.2480   1.4106
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  14.682863 882.743658   0.017  0.98673
## intake_typePublic Assist      0.447231   0.107012   4.179 2.92e-05 ***
## intake_typeStray              1.122457   0.071199  15.765  < 2e-16 ***
## intake_conditionFeral        11.412673 308.506257   0.037  0.97049
## intake_conditionInjured      -1.697678   0.258930  -6.557 5.51e-11 ***
## intake_conditionNormal        0.720775   0.252981   2.849  0.00438 **
## intake_conditionNursing       0.270051   0.365231   0.739  0.45967
## intake_conditionOther        -0.716730   0.454048  -1.579  0.11444
## intake_conditionPregnant     11.519334 173.108833   0.067  0.94694
## intake_conditionSick         -1.174912   0.270803  -4.339 1.43e-05 ***
## age_upon_intake_years        -0.119653   0.008865 -13.497  < 2e-16 ***
## sex_upon_intakeIntact Female -12.222052 882.743619  -0.014  0.98895
## sex_upon_intakeIntact Male   -12.214970 882.743619  -0.014  0.98896
## sex_upon_intakeNeutered Male -11.833725 882.743618  -0.013  0.98930
## sex_upon_intakeSpayed Female -11.466612 882.743619  -0.013  0.98964
## sex_upon_intakeUnknown       -13.643957 882.743647  -0.015  0.98767
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 10539  on 36146  degrees of freedom
## Residual deviance:  9176  on 36131  degrees of freedom
## AIC: 9208
##
## Number of Fisher Scoring iterations: 13
```

As we see, two variables that have the biggest influence on the response variable are intake_conditionFeral and intake_conditionPregnant. Despite that, they are both statistically insignificant, meaking them unreliable. As for the rest, they have small coefficient, approximately 0.

```
confint(glm1,"age_upon_intake_years", level=0.95)
```

```
##      2.5 %     97.5 %
## -0.1369066 -0.1021476
```

The 95% CI means that if we randomly repeat splitting the training/testing data set, 95% of the time the result we see would show that the coefficient for age_upon_intake_year in the range of -0.1372816 and -0.1030268 #### STEP 4

Use the logistic regression model to predict the live release outcomes on the test set. Start by using 0.5 as a threshold, and show your confusion matrix.

```
# Your code here!
#release_outcome_test = predict(glm1,new_data=test_set,type="response")
release_outcome_test = predict(glm1, newdata = test_set, type = "response")

test_pred <- rep(0, nrow(test_set))
test_pred[release_outcome_test >0.5]  =1
conf <- table(test_pred,test_set$live_release)

#test_probs = predict(glm1, newdata = test_set, type = "response")
#length(test_probs)
#max()

accuracy_level <- (conf[2,2]+conf[1,1])/sum(conf)
cat("Accuracy of prediction with 0.5 threshold is",accuracy_level)
```

```
## Accuracy of prediction with 0.5 threshold is 0.96459
```

```
conf
```

```
##
## test_pred    0    1
##         0    6    7
##         1  313 8711
```

STEP 5 Is it possible that another threshold will be better for this model? Try a few different ones and show one that results in better prediction performance (if none do, show one that gives a worse result). Justify your choice, explaining why the different error types made by changing the threshold is preferable to your previous result.

```
# Your code here!
test_pred <- rep(0, nrow(test_set))
test_pred[release_outcome_test >0.43]  =1
table(test_pred,test_set$live_release)
```

```
##
## test_pred    0    1
##         0    1    0
##         1  318 8718
```

```
#I tried different threshold nd with 0.43, I got the highest accuracy because it increases the accuracy

est_acc <- function(threshold) {
  predict_acc <- release_outcome_test > threshold
  conf <- table(predict_acc,test_set$live_release)
  accuracy <- (conf[1,1]+conf[2,2])/sum(conf)
  return(accuracy)
}

threshold_level <- seq(from = 0.4,to= 0.9, by =0.01)
max_acc = 0
max_thres = 0
for (threshold in threshold_level){
  cur_acc <- est_acc(threshold)
  if (cur_acc >max_acc){
    max_acc <- cur_acc
    max_thres <- threshold
  }
}
cat(max_acc,max_thres)
```

```
## 0.9648113 0.4
```

In this case, the threshold is 0.6, which shows the maximum accuracy obtained. It is preferable to previous results because one of type I/II error decreased and overall the accuracy increased.

STEP 6: Bonus Question Write code or use code from existing packages that you come across to create the ROC curve for this classification problem!

```
# Your code here!
```

STEP 7 Lastly, write a short summary (1-2 paragraphs) with your reflections on the exercise, including:

1. Reasoning for the choices you made in your feature selection. The first reason I choose sex is because it sex in data set is categorized as 5 types. I did not choose breed because there are so many types of breed and it causes the model to run very slowly. Another reason is just the curiosity about sexual identity affecting the release rate. I read several articles on sexism and this seems like a good topic to invest my time into analyzing.

2. What other variables might have been interesting to look at which are not available in the data. Other interesting variable can be the trendy preference of people adopting dog in different time. For instance, at specific period of time, people might have different trend/preference for adopting specific types of dogs. In addition, I was thinking about the economic situation versus claims about joblessness which can correlate with dog's release rate. For instance, there might be a upward trend of dogs being adopted when the economic around Texas area is booming or vice versa.

3. How good are your model's predictions on the test set? What might explain it (that predictions are or are not highly accurate) and how could it be improved? The model is pretty good at predicting, which might due to the fact that there is training and testing test have many commonality. In addition, three variables namely intake_type, intake_condition and age_upon_intake_years are highly skewed. In addition, for sex_upon_intake, the variable has small coefficient, implying that it does not have strong correlation/affect on the outcome.

End of Assignment

Final Steps

Before finalizing your project you'll want to be sure there are **comments in your code chunks** and **text outside of your code chunks** to explain what you're doing in each code chunk. These explanations are incredibly helpful for someone who doesn't code or someone unfamiliar to your project.

You have two options for submission:

1. You can complete this .rmd file, knit it to pdf and submit the resulting .pdf file on Forum.
2. You can complete the Google Doc version of this assignment, include your code, graphs, results, and your explanations wherever necessary and download the Google Doc as a pdf file and submit the pdf file on Forum. If you choose this method, you need to make sure you will provide a link to an .R script file where your code can be found (you can host your code on Github or Google Drive). Note that links to Google Docs are not accepted as your final submission.

Knitting your R Markdown Document

Last but not least, you'll want to **Knit your .Rmd document into a pdf document**. If you get an error, take a look at what the error says and edit your .Rmd document. Then, try to Knit again! Troubleshooting these error messages will teach you a lot about coding in R. If you get any error that doesn't make sense to you, post it on Piazza.

Good Luck! The Teaching Team