Assignment 2

Mitheysh Asokan, Jason (Eungjoo) Kim

Question 1: Nonlinear Regression

1.1 Process your data

Dataset chosen: diamonds

Choose the input and output variables.

```
diamonds <- read.csv('diamonds.csv')
diamonds <- select(diamonds,carat:price)

dim(diamonds)
```

```
## [1] 5000    7
```

```
head(diamonds)
```

```
##   carat       cut color clarity depth table price
## 1  0.77   Premium     E     SI1  60.4    58  2975
## 2  1.51      Fair     F      I1  67.8    59  3734
## 3  0.71   Premium     D     SI1  61.7    56  2863
## 4  0.90 Very Good     H     SI1  62.3    63  3387
## 5  1.00      Fair     I     SI1  67.9    62  2856
## 6  0.92 Very Good     J     SI1  62.6    58  3170
```

Remove all NAs from your data

```
diamonds <- na.omit(diamonds)

dim(diamonds)
```

```
## [1] 5000    7
```

Downsample if your data is larger than 5000 rows:

```
diamonds <- diamonds[diamonds$price > 2500,]

dim(diamonds)
```

```
## [1] 4460    7
```

If there are numeric variables that were supposed to be categorical, convert them to categorical

```
diamonds$cut <- as.factor(diamonds$cut)
diamonds$color <- as.factor(diamonds$color)
diamonds$clarity <- as.factor(diamonds$clarity)
```

1.2 train/Test Split

Split your data into train and test using 80/20 ratio. Print number of observations each dataset contains.

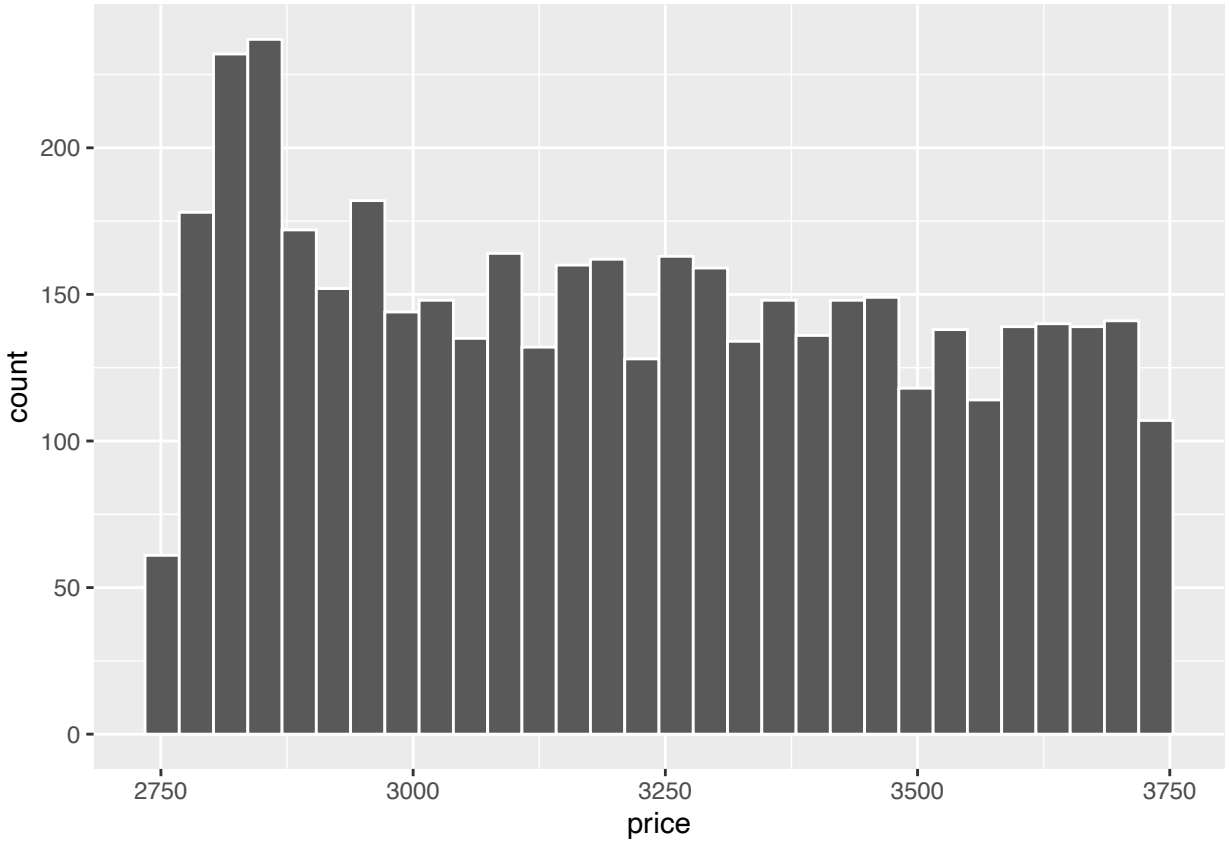
```
set.seed(100)

train_size <- floor(0.8 * nrow(diamonds))
train_inds <- sample(1:nrow(diamonds), size = train_size)
test_inds  <- setdiff(1:nrow(diamonds), train_inds)

train <- diamonds[ train_inds , ]
test  <- diamonds[ test_inds , ]

cat('train size:', nrow(train), '\ntest size:', nrow(test))
```

```
## train size: 3568
## test size: 892
```

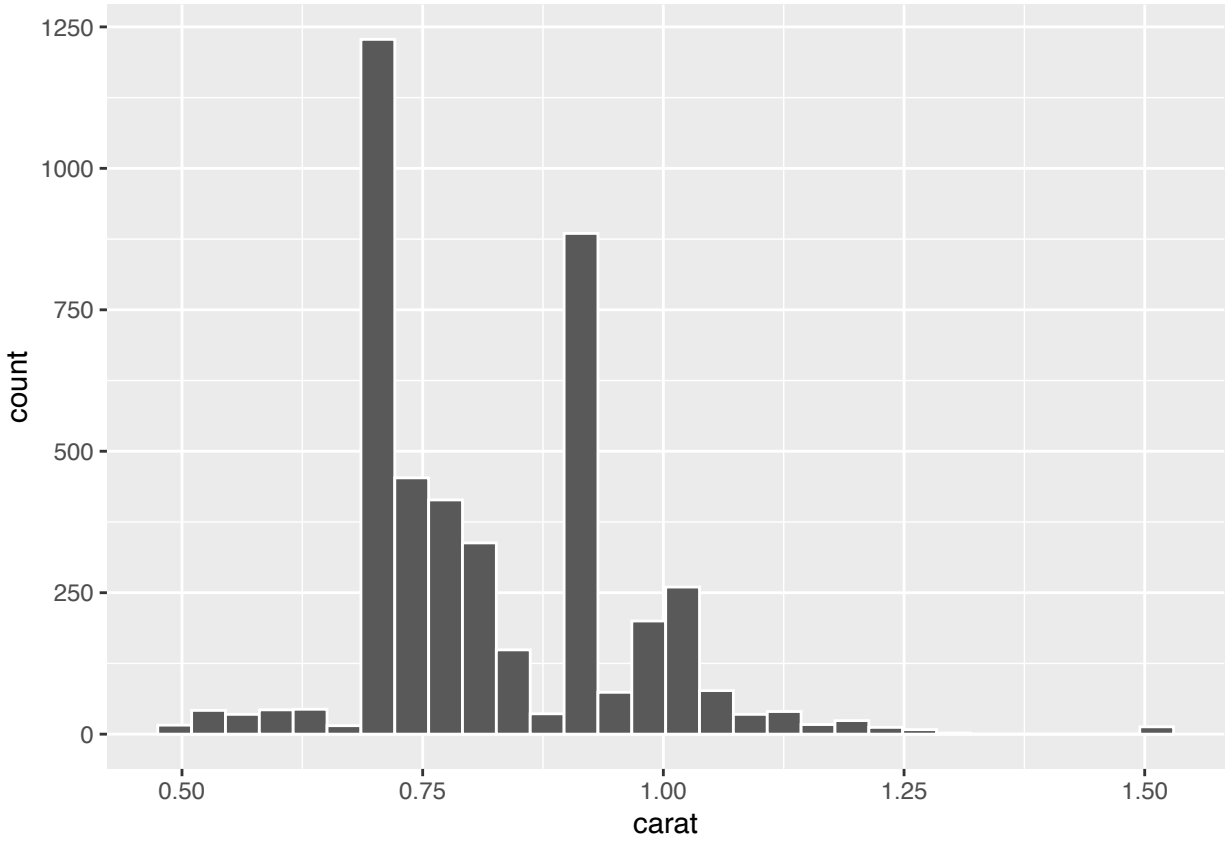
1.3 Visualize the data

Draw histogram of one of the numeric input variable and the output variable you selected.

```
ggplot(data = diamonds) +
  geom_histogram(aes(x = price), color = 'white')
```

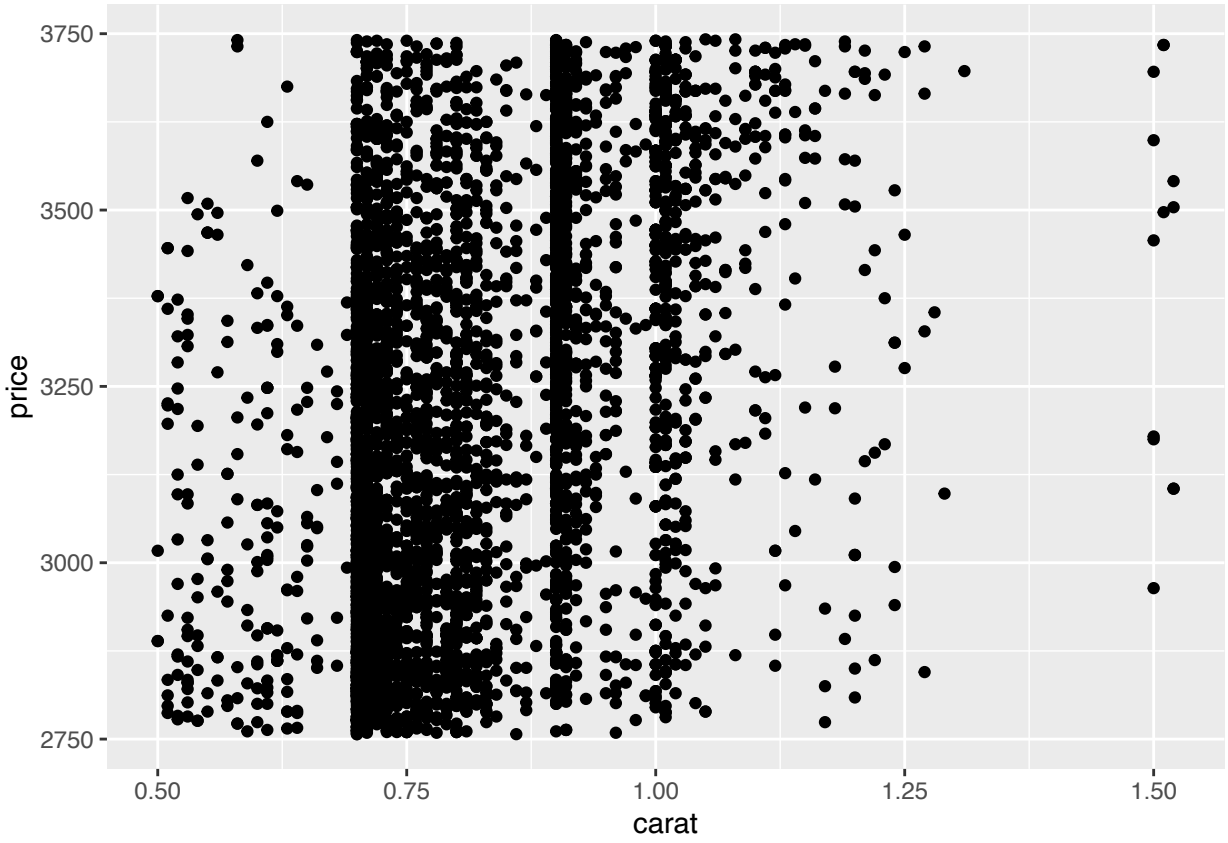



```
ggplot(data = diamonds) +
  geom_histogram(aes(x = carat), color = 'white')
```



Draw scatter plot between one of your numeric inputs and the output variable.

```
ggplot(data = diamonds) +
  geom_point(aes(x = carat, y = price))
```



Discuss whether the plot indicate a relation, if it is linear, if there are outliers?

The plot definitely hints at some minor relations. According to the diagram, the price of the diamond is increasing with respect to the carat. This is a true statement when compared to real diamond markets, where the carat weight of the diamond tends to partial influence the price of the diamond.

It is hard to definitely claim if the relationship is linear, but an upward sloping linear trend can be slightly observed.

Based on the data used, there doesn't seem to be any outliers.

1.4 Fit 4 models

One simple linear regression,

```
fit1 <- lm(price ~ carat, data = train)
```

One multilinear regression with all the variables you selected.

```
fit2 <- lm(price ~ . , data = train)
```

One polynomial regression using one input variable and one output variable.

```
fit3 <- lm(price ~ poly(carat,2), data = train)
```

One Locally Weighted Regression using one input numeric input variable and one output variable.

```
fit4 <- loess(price ~ depth, data = train)
```

Calculate each model's RMSE on the train set. Which one performed the best and which did worse? Rank the models based on their training error.

Best: Model2

Worst: Model4

Ranking: Model2, Model3, Model1, Model4

```
sigma(fit1)
```

```
## [1] 269.3641
```

```
sigma(fit2)
```

```
## [1] 234.5879
```

```
sigma(fit3)
```

```
## [1] 268.3738
```

```
pred4 <- predict(fit4)
RMSE(pred4, train$price)
```

```
## [1] 291.9991
```

Calculate each model's RMSE on the test set. Which one performed the best and which did worse? Rank the models based on their test error.

Best: Model2

Worst: Model4

Ranking: Model2, Model3, Model1, Model4

```
pred1 <- predict(fit1, newdata=test)
pred2 <- predict(fit2, newdata=test)
pred3 <- predict(fit3, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 265.4409
```

```
RMSE(pred2, test$price)
```

```
## [1] 233.0535
```

```
RMSE(pred3, test$price)
```

```
## [1] 265.2131
```

```
RMSE(pred4, test$price)
```

```
## [1] 286.1072
```

Did the order of models change when ranked using training and test error?

No.

1.5. Cross Validation

Fit the 4 models but train using the cross validation.

```
train.control <- trainControl(method = 'cv', number = 10)

model1 <- train(price ~ carat, data = diamonds, method = 'lm', trControl = train.control)
model2 <- train(price ~ ., data = diamonds, method = 'lm', trControl = train.control)
model3 <- train(price ~ poly(carat,2), data = diamonds, method = 'lm', trControl = train.control)
model4 <- train(price ~ carat, data = diamonds, method = 'gamLoess', trControl = train.control)

pred1 <- predict(model1, newdata=test)
pred2 <- predict(model2, newdata=test)
pred3 <- predict(model3, newdata=test)
pred4 <- predict(model4, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 265.0814
```

```
RMSE(pred2, test$price)
```

```
## [1] 231.9981
```

```
RMSE(pred3, test$price)
```

```
## [1] 264.7938
```

```
RMSE(pred4, test$price)
```

```
## [1] 261.1916
```

What is the test error of each resulting model?

Best: Model2

Worst: Model1

Ranking: Model2, Model4, Model3, Model1

```
pred1 <- predict(model1, newdata=test)
pred2 <- predict(model2, newdata=test)
pred3 <- predict(model3, newdata=test)
pred4 <- predict(model4, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 265.0814
```

```
RMSE(pred2, test$price)
```

```
## [1] 231.9981
```

```
RMSE(pred3, test$price)
```

```
## [1] 264.7938
```

```
RMSE(pred4, test$price)
```

```
## [1] 261.1916
```

Did the order of models' test performances change when trained using cross validation?

Yes.

1.6. Shrinkage

Fit the first three models (exclude locally weighted model) using ridge regression.

```
x1 <- model.matrix(price ~ carat, data = diamonds)
x2 <- model.matrix(price ~ ., data = diamonds)
x3 <- model.matrix(price ~ poly(carat,2), data = diamonds)

y <- diamonds$price

fit1 <- glmnet(x1,y,alpha=0)
fit2 <- glmnet(x2,y,alpha=0)
fit3 <- glmnet(x3,y,alpha=0)
```

Calculate RMSE loss on test set.

```
pred1 <- predict(fit1, newx=x1, newdata=test)
pred2 <- predict(fit2, newx=x2, newdata=test)
pred3 <- predict(fit3, newx=x3, newdata=test)

RMSE(pred1, test$price)
```

```
## [1] 290.3445
```

```
RMSE(pred2, test$price)
```

```
## [1] 291.6528
```

```
RMSE(pred3, test$price)
```

```
## [1] 290.5159
```

Fit the first three models (exclude locally weighted model) using lasso regression.

```
fit4 <- glmnet(x1,y,alpha=1)
fit5 <- glmnet(x2,y,alpha=1)
fit6 <- glmnet(x3,y,alpha=1)
```

Calculate RMSE loss on test set.

```
pred4 <- predict(fit4, newx=x1, newdata=test)
pred5 <- predict(fit5, newx=x2, newdata=test)
pred6 <- predict(fit6, newx=x3, newdata=test)

RMSE(pred4, test$price)
```

```
## [1] 299.5081
```

```
RMSE(pred5, test$price)
```

```
## [1] 316.14
```

```
RMSE(pred6, test$price)
```

```
## [1] 300.0939
```

Which model yielded the minimum test loss? Rank the 6 models.

Minimum Test Loss: Model1

Rankings: Model1, Model3, Model2, Model4, Model6, Model5