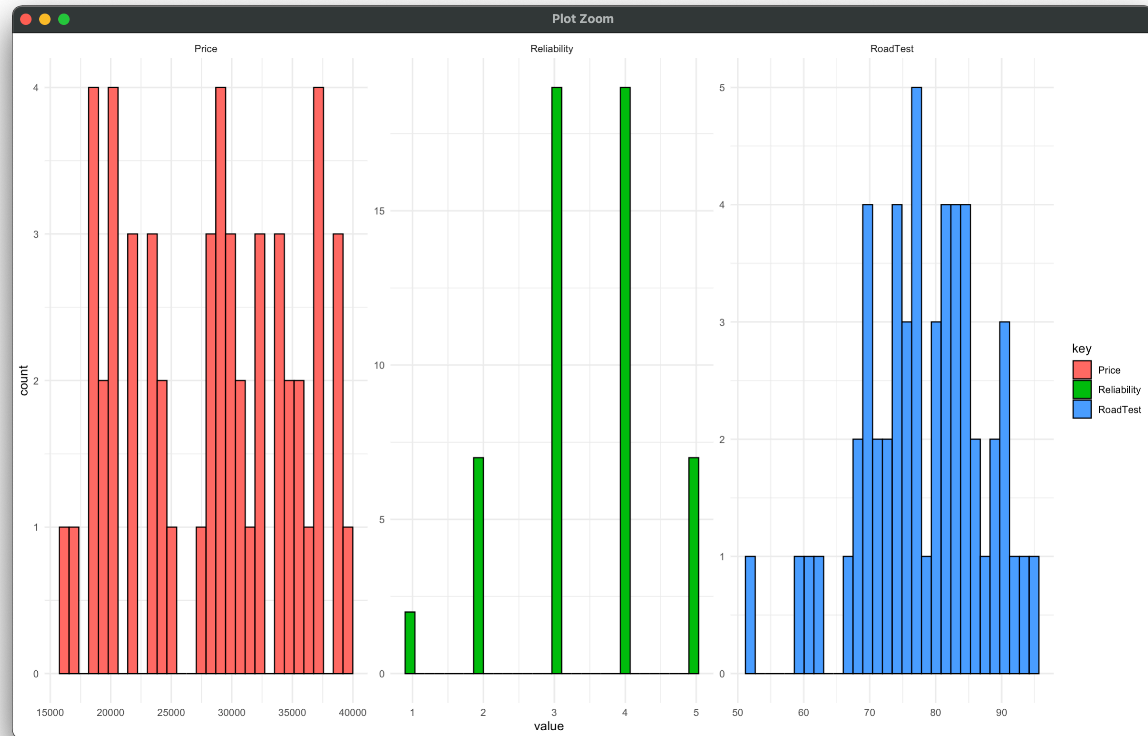
1. From R:
    1. All code from your R script (Code should be presented single-spaced in a <u>fixed-width font</u>. Adjust the font size so that no lines of code extend to the next line in the document)

```
# YUEH-TING WU
# MIS 545 Section 02
# Lab07WuY.R
# In this R programming, importing a csv file and generate a k-nearest model to
# know which k value has more higher predictive accuracy.

# Install the tidyverse package
# install.packages("tidyverse")

# Load the tidyverse and class libraries
library(tidyverse)
library(class)

# Set the working directory to Lab07 folder
setwd("~/MIS 545/Lab07")

# Read SedanSize.csv into a tibble called sedanSize
sedanSize <- read_csv(file = "SedanSize.csv",
                      col_types = "cfnii",
                      col_names = TRUE)

# Display sedanSize in the console
print(sedanSize)

# Display the structure of sedanSize in the console
str(sedanSize)

# Display the summary of sedanSize in the console
summary(sedanSize)

# Remove the MakeModel feature from the tibble
sedanSize <- sedanSize %>% select(-MakeModel)

# Separate the tibble into two.
# One called sedanSize with 3 variables (Price, RoadTest, and Reliability), and
# another called sedanSizeLabels with 1 variable (SedanSize).
sedanSizeLabels <- sedanSize %>% select(SedanSize)
sedanSize <- sedanSize %>% select(-SedanSize)

# Recreate the displayAllHistograms() function
displayAllHistograms <- function(tibbleDataset) {
  tibbleDataset %>%
```

```
    keep(is.numeric) %>%
    gather() %>%
    ggplot() + geom_histogram(mapping = aes(x=value, fill=key),
                             color = "black") +
    facet_wrap (~key, scales = "free") +
    theme_minimal ()
}

# Call the displayAllHistograms() function, passing in sedanSize as an argument.
displayAllHistograms(sedanSize)

# Randomly split the dataset into sedanSizeTraining (75% of records) and
# sedanSizeTesting (25% of records) using 517 as the random seed.
set.seed(517)
# Create a vector of 75% randomly sample rows from the orighinal dataset
sampleSet <- sample(nrow(sedanSize),
                    round(nrow(sedanSize) * 0.75),
                    replace = FALSE)

# Put the records from the 75% sample into sedanSizeTraining (3 variables) and
# sedanSizeTrainingLabels (1 variable).
sedanSizeTraining <- sedanSize[sampleSet, ]
sedanSizeTrainingLabels <- sedanSizeLabels[sampleSet, ]

# Put the remaining 25% of records into sedanSizeTesting (3 variables) and
# sedanSizeTestingLabels (1 variable).
sedanSizeTesting <- sedanSize[-sampleSet, ]
sedanSizeTestingLabels <- sedanSizeLabels[-sampleSet, ]

# Generate the k-nearest neighbors model using sedanSizeTraining as the train
# argument, sedanSizeTesting as the test argument.
sedanSizePrediction <- knn(train = sedanSizeTraining,
                           test = sedanSizeTesting,
                           cl = sedanSizeTrainingLabels$SedanSize,
                           k = 7)

# Display the sedanSizePrediction from the testing dataset on the console
print(sedanSizePrediction)

# Display the summary of the sedanSizePrediction from the testing dataset
summary(sedanSizePrediction)

# Evaluate the model by forming a confusion matrix
sedanSizeConfusionMatrix <- table(sedanSizeTestingLabels$SedanSize,
                                  sedanSizePrediction)
```

```r
# Display the confusion matrix on the console
print(sedanSizeConfusionMatrix)

# Calculate the model predictive accuracy and store it into a variable called
# predictiveAccuracy.
predictiveAccuracy <- sum(diag(sedanSizeConfusionMatrix)) /
  nrow(sedanSizeTesting)

# Display the predictiveAccuracy on the console
print(predictiveAccuracy)

# Create a matrix of k-values with their predictive accuracy (the matrix will be
# empty and have 2 column and 0 rows).
# And store the matrix into an object called kValueMatrix
kValuematrix <- matrix(data = NA,
                       ncol = 2,
                       nrow = 0)

# Assign column names of "k value" and "Predictive accuracy" to the kValueMatrix
colnames(kValuematrix) <- c("k value", "Predictive accuracy")

# Loop through odd values of k from 1 up to the number of records in the
# training dataset.
for (kValue in 1:nrow(sedanSizeTraining))
  {
  # Only calculate predictive accuracy if the k value is odd
  if(kValue %% 2 != 0) {
    # Generate the model
    sedanSizePrediction <- knn(train = sedanSizeTraining,
                               test = sedanSizeTesting,
                               cl = sedanSizeTrainingLabels$SedanSize,
                               k = kValue)

    # Generate the confusion matrix
    sedanSizeConfusionMatrix <- table(sedanSizeTestingLabels$SedanSize,
                                      sedanSizePrediction)

    # Calculate the predictive accuracy
    predictiveAccuracy <- sum(diag(sedanSizeConfusionMatrix)) /
      nrow(sedanSizeTesting)

    # Add a new row to the kValueMatrix
    kValuematrix <- rbind(kValuematrix, c(kValue, predictiveAccuracy))
  }
}
print(kValuematrix)
```

2. The histogram plots



3. The k-value / predictive accuracy matrix

	k value	Predictive accuracy
1	1	0.8571429
2	3	0.8571429
3	5	0.8571429
4	7	0.8571429
5	9	0.8571429
6	11	0.8571429
7	13	0.7857143
8	15	0.8571429
9	17	0.5714286
10	19	0.5714286
11	21	0.5714286
12	23	0.5714286
13	25	0.5714286
14	27	0.5714286
15	29	0.5714286
16	31	0.5714286
17	33	0.2857143
18	35	0.5000000
19	37	0.4285714
20	39	0.2857143

2. Answer the following question in a sentence: What would be the best value for k given this data? Why?

The best values of k are 1, 3, 5, 7, 9, 11, 15 because taking seven nearest neighbors to get the highest accuracy.

3. Answer the following question in a sentence: How could an automobile manufacturer take advantage of this model?

Through this model, an automobile manufacturer can make a more accurate prediction.