Chapter Name

Exploring Data

Exploring and Understanding Data

- After collecting data and loading it into R's data structures, the next step in the machine learning process involves examining the data in detail.
- It is during this step that you will begin to explore the data's features and examples, and realize the peculiarities that make your data unique.
- The better you understand your data, the better you will be able to match a machine learning model to your learning problem.

The **usedcars.csv** Dataset

☞ The best way to learn the process of data exploration is with an example.

- We will explore the usedcars.csv dataset, which contains actual data about used cars recently advertised for sale on a popular U.S. website.
- Following along with the examples, we have to be sure that this file has been downloaded and saved to our R working directory.
- Since the dataset is stored in the CSV form, we can use the read.csv() function to load the data into an R data frame:

```
> usedcars <- read.csv("usedcars.csv", stringsAsFactors = FALSE)
```

☞ Given the usedcars data frame, we will now assume the role of a data scientist who has the task of understanding the used car data.

- Although data exploration is a fluid process, the steps can be imagined as a sort of investigation in which questions about the data are answered.
- The exact questions may vary across projects, but the types of questions are always similar.

Exploring the Structure of Data (1/2)

☞ One of the first questions to ask in an investigation of a new dataset should be about how the dataset is organized.

- If you are fortunate, your source will provide a data dictionary, which is a document that describes the dataset's features. Assuming that the used car data does not come with this documentation, we'll need to create one on our own.
- The str() function provides a method to display the construction of R data structures such as data frames, vectors, or lists. It can be used to create the basic outline for our data dictionary:

```
> str(usedcars)
'data.frame':   150 obs. of  6 variables:
 $ year       : int  2011 2011 2011 2011 2012 2010 2011 2010 2011 2010 ...
 $ model      : chr  "SEL" "SEL" "SEL" "SEL" ...
 $ price      : int  21992 20995 19995 17809 17500 17495 17000 16995 16995 16995 ...
 $ mileage    : int  7413 10926 7351 11613 8367 25125 27393 21026 32655 36116 ...
 $ color      : chr  "Yellow" "Gray" "Silver" "Gray" ...
 $ transmission: chr  "AUTO" "AUTO" "AUTO" "AUTO" ...
```

Exploring the Structure of Data (2/2)

☞ Using such a simple command as str(usedcars), we learn a wealth of information about the dataset.

- The statement 150 obs. informs us that the data includes 150 observations, which is just another way of saying that the dataset contains 150 records or examples.
- The 6 variables statement refers to the six features that were recorded in the data.

☞ Looking at the line for the feature called color, we can note some additional details:

```
$ color      : chr  "Yellow" "Gray" "Silver" "Gray" ...
```

- After the variable's name, the chr label tells us that the feature is character type. In this dataset, three of the variables are character while three are noted as int, which indicates integer type.
- Although the usedcars dataset includes only character and integer variables, you are also likely to encounter num or numeric type while using noninteger data. Any factors would be listed as factor type.
- Following each variable's type, R presents a sequence of the first few feature values.

Exploring Numeric Variables (1/2)

☞ To investigate the numeric variables in the used car data, we will employ a common set of measurements to describe values known as summary statistics.

☞ The summary() function displays several common summary statistics. Let's take a look at a single feature, year:

```
> summary(usedcars$year)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   2000    2008    2009    2009    2010    2012
```

- Even if you aren't already familiar with summary statistics, you may be able to guess some of them from the heading before the summary() output.
- Ignoring the meaning of the values for now, the fact that we see numbers such as 2000, 2008, and 2009 could lead us to believe that the year variable indicates the year of manufacture rather than the year the advertisement was posted, since we know the vehicles were recently listed for sale.

Exploring Numeric Variables (2/2)

☞ We can also use the summary() function to obtain summary statistics for several numeric variables at the same time:

```
> summary(usedcars[c("price", "mileage")])
     price          mileage
 Min.   : 3800   Min.   :  4867
 1st Qu.:10995   1st Qu.: 27200
 Median :13592   Median : 36385
 Mean   :12962   Mean   : 44261
 3rd Qu.:14904   3rd Qu.: 55125
 Max.   :21992   Max.   :151479
```

- The six summary statistics that the summary() function provides are simple, yet powerful tools to investigate data. They can be divided into two types: measures of center and measures of spread.

Measuring the Central Tendency – Mean (1/2)

Measures of central tendency are a class of statistics used to identify a value that falls in the middle of a set of data.

☞ You most likely are already familiar with one common measure of center: the average.

- In common use, when something is deemed average, it falls somewhere between the extreme ends of the scale.
- An average student might have marks falling in the middle of his or her classmates; an average weight is neither unusually light nor heavy. An average item is typical and not too unlike the others in the group. You might think of it as an exemplar by which all the others are judged.

☞ In statistics, the average is also known as the mean, which is a measurement defined as the sum of all values divided by the number of values.

- For example, to calculate the mean income in a group of three people with incomes of $36,000, $44,000, and $56,000, use the following command:

```
> (36000 + 44000 + 56000) / 3
[1] 45333.33
```

Measuring the Central Tendency – Mean (2/2)

- R also provides a mean() function, which calculates the mean for a vector of numbers:

```
> mean(c(36000, 44000, 56000))
[1] 45333.33
```

  - The mean income of this group of people is about $45,333. Conceptually, this can be imagined as the income each person would have, if the total amount of income were divided equally across every person.

- Recall that the preceding summary() output listed mean values for the price and mileage variables.
  - The means suggest that the typical used car in this dataset was listed at a price of $12,962 and had an odometer reading of 44,261.
  - What does this tell us about our data?

Since the average price is relatively low, we might expect that the dataset contains economy class cars. Of course, the data can also include late-model luxury cars with high mileage, but the relatively low mean mileage statistic doesn't provide evidence to support this hypothesis. On the other hand, it doesn't provide evidence to ignore the possibility either. We'll need to keep this in mind as we examine the data further.

Measuring the Central Tendency – Median (1/2)

- Another commonly used measure of central tendency is the median, which is the value that occurs halfway through an ordered list of values.
  - As with the mean, R provides a median() function, which we can apply to our salary data, as shown in the following example:

```
> median(c(36000, 44000, 56000))
[1] 44000
```

    Because the middle value is 44000, the median income is $44,000.
  - At the first glance, it seems like the median and mean are very similar measures. Certainly, the mean value of $45,333 and the median value of $44,000 are not very different.
  - Why have two measures of central tendency?

The reason is due to the fact that the mean and median are affected differently by the values falling at the far ends of the range. In particular, the mean is highly sensitive to outliers, or values that are atypically high or low in relation to the majority of data. Because the mean is sensitive to outliers, it is more likely to be shifted higher or lower by a small number of extreme values.

Measuring the Central Tendency – Median (2/2)

☞ Recall again the reported median values in the summary() output for the used car dataset.

- Although the mean and median price are fairly similar (differing by approximately five percent), there is a much larger difference between the mean and median for mileage.

For mileage, the mean of 44,261 is approximately 20 percent more than the median of 36,385. Since the mean is more sensitive to extreme values than the median, the fact that the mean is much higher than the median might lead us to suspect that there are some used cars in the dataset with extremely high mileage values. To investigate this further, we'll need to add additional summary statistics to our analysis.

Measuring Spread

☞ Measuring the mean and median provides one way to quickly summarize the values, but these measures of center tell us little about whether or not there is diversity in the measurements.

☞ To measure the diversity, we need to employ another type of summary statistics that is concerned with the spread of data, or how tightly or loosely the values are spaced.

☞ Knowing about the spread provides a sense of the data's highs and lows and whether most values are like or unlike the mean and median.

The Five-Number Summary

- The five-number summary is a set of five statistics that roughly depict the spread of a feature's values.
- All five of the statistics are included in the output of the summary() function. Written in order, they are:
  1. Minimum (Min.)
  2. First quartile, or Q1 (1st Qu.)
  3. Median, or Q2 (Median)
  4. Third quartile, or Q3 (3rd Qu.)
  5. Maximum (Max.)

- As you would expect, minimum and maximum are the most extreme feature values, indicating the smallest and largest values, respectively. R provides the min() and max() functions to calculate these values on a vector of data.
- The span between the minimum and maximum value is known as the range. In R, the range() function returns both the minimum and maximum value.

Quartiles and Quantiles

- The first and third quartiles—Q1 and Q3—refer to the value below or above which one quarter of the values are found.
  - Along with the (Q2) median, the quartiles divide a dataset into four portions, each with the same number of values.
- Quartiles are a special case of a type of statistics called quantiles, which are numbers that divide data into equally sized quantities.
  - In addition to quartiles, commonly used quantiles include tertiles (three parts), quintiles (five parts), deciles (10 parts), and percentiles (100 parts).
- Percentiles are often used to describe the ranking of a value;
  - for instance, a student whose test score was ranked at the 99th percentile performed better than, or equal to, 99 percent of the other test takers.

Interquartile Range (IQR)

☞ The middle 50 percent of data between the first and third quartiles is of particular interest because it in itself is a simple measure of spread.

- The difference between Q1 and Q3 is known as the Interquartile Range (IQR), and it can be calculated with the IQR() function:

```
> IQR(usedcars$price)
[1] 3909.5
```

- We could have also calculated this value by hand from the summary() output for the usedcars$price variable by computing 14904 – 10995 = 3909.
- The small difference between our calculation and the IQR() output is due to the fact that R automatically rounds the summary() output.

The **quantile()** Function

☞ The quantile() function provides a robust tool to identify quantiles for a set of values.

- By default, the quantile() function returns the five-number summary. Applying the function to the used car data results in the same statistics as done earlier:

```
> quantile(usedcars$price)
   0%   25%   50%   75%  100%
3800.0 10995.0 13591.5 14904.5 21992.0
```

- If we specify an additional probs parameter using a vector denoting cut points, we can obtain arbitrary quantiles, such as the 1st and 99th percentiles:

```
> quantile(usedcars$price, probs = c(0.01, 0.99))
    1%    99%
5428.69 20505.00
```

- The seq() function is used to generate vectors of evenly-spaced values. This makes it easy to obtain other slices of data, such as the quintiles (five groups), as shown in the following command:

```
> quantile(usedcars$price, seq(from = 0, to = 1, by = 0.20))
   0%   20%   40%   60%   80%  100%
3800.0 10759.4 12993.8 13992.0 14999.0 21992.0
```

Discussing the **summary()** Output (1/2)

- Equipped with an understanding of the five-number summary, we can re-examine the used car summary() output.
  - On the price variable, the minimum was $3,800 and the maximum was $21,992.
  - Interestingly, the difference between the minimum and Q1 is about $7,000, as is the difference between Q3 and the maximum; yet, the difference from Q1 to the median to Q3 is roughly $2,000.
  - This suggests that the lower and upper 25 percent of values are more widely dispersed than the middle 50 percent of values, which seem to be more tightly grouped around the center.
  - We see a similar trend with the mileage variable, which is not unsurprising.
  - As you will learn later in this chapter, this pattern of spread is common enough that it has been called a "normal" distribution of data.

Discussing the **summary()** Output (2/2)

- The spread of the mileage variable also exhibits another interesting property:
  - the difference between Q3 and the maximum value is far greater than that between the minimum value and Q1.
  - In other words, the larger values are far more spread out than the smaller values.
- This finding explains why the mean value is much greater than the median.
  - Because the mean is sensitive to extreme values, it is pulled higher, while the median stays relatively in the same place.
  - This is an important property, which becomes more apparent when the data is presented visually.

Visualizing Numeric Values – Boxplots (1/3)

☞ Visualizing numeric variables can be helpful in diagnosing data problems.

- A common visualization of the five-number summary is boxplot, also known as a box-and-whiskers plot.
- The boxplot displays the center and spread of a numeric variable in a format that allows you to quickly obtain a sense of the range and skew of a variable or compare it to other variables.

☞ Let's take a look at a boxplot for the used car price and mileage data.

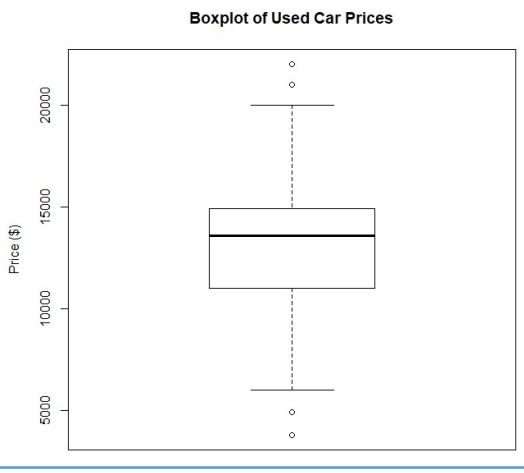
- To obtain a boxplot for a variable, we will use the boxplot() function. We will also specify a pair of extra parameters, main and ylab, to add a title to the figure and label the *y* axis (the vertical axis), respectively.
- The commands to create the price and mileage boxplots are:

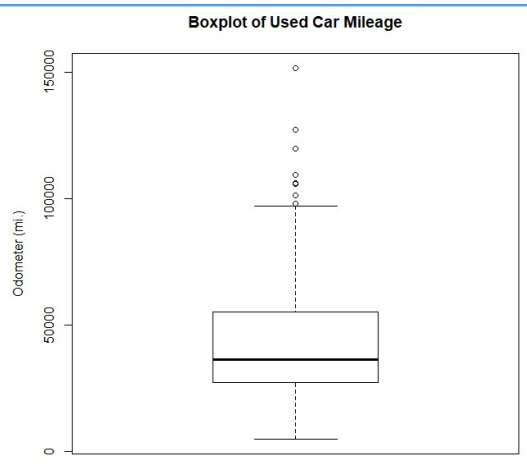
```
> boxplot(usedcars$price, main="Boxplot of Used Car Prices", ylab="Price ($)")
```

```
> boxplot(usedcars$mileage, main="Boxplot of Used Car Mileage", ylab="Odometer (mi.)")
```


Visualizing Numeric Values – Boxplots (2/3)

```
> boxplot(usedcars$price, main="Boxplot of Used Car Prices", ylab="Price ($)")
```






```
> boxplot(usedcars$mileage, main="Boxplot of Used Car Mileage", ylab="Odometer (mi.)")
```

Visualizing Numeric Values – Boxplots (3/3)

- The plot depicts the five-number summary values using the horizontal lines and dots.
  - The horizontal lines forming the box in the middle of each figure represent Q1, Q2 (the median), and Q3 while reading the plot from the bottom to the top.
  - The median is denoted by the dark line, which lines up with $13,592 on the vertical axis for price and 36,385 mi. on the vertical axis for mileage.
- The minimum and maximum values can be illustrated using the whiskers that extend below and above the box;
  - however, a widely used convention only allows the whiskers to extend to a minimum or maximum of 1.5 times the IQR below Q1 or above Q3. Any values that fall beyond this threshold are considered outliers and are denoted as circles or dots.
  - For example, recall that the IQR for the price variable was 3,909 with a Q1 of 10,995 and a Q3 of 14,904. An outlier is therefore any value that is less than *10995 - 1.5 * 3909= 5131.5* or greater than *14904 + 1.5 * 3909 = 20767.5*. The plot shows two such outliers on both the high and low ends.
  - On the mileage boxplot, there are no outliers on the low end and thus, the bottom whisker extends to the minimum value, 4,867. On the high end, we see several outliers beyond the 100,000 mile mark. These outliers are responsible for our earlier finding, which noted that the mean value was much greater than the median.

Visualizing Numeric Values – Histograms (1/3)

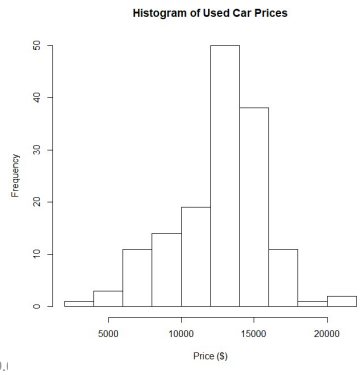
- A histogram is another way to graphically depict the spread of a numeric variable.
  - It is similar to a boxplot in a way that it divides the variable's values into a predefined number of portions or bins that act as containers for values.
- Their similarities end there, however.
  - On one hand, a boxplot requires that each of the four portions of data must contain the same number of values, and widens or narrows the bins as needed.
  - On the other hand, a histogram uses any number of bins of an identical width, but allows the bins to contain different number of values.

Visualizing Numeric Values – Histograms (2/3)

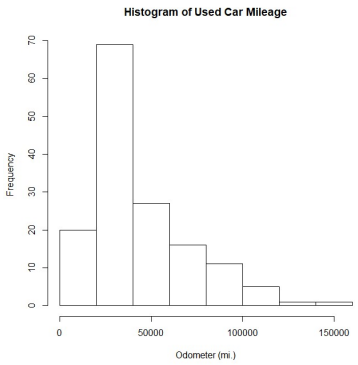
☞ We can create a histogram for the used car price and mileage data using the hist() function.

- The commands to create the histograms and the resulting diagrams are:

```
> hist(usedcars$price, main = "Histogram of Used Car Prices",
+ xlab = "Price ($)")
```




```
> hist(usedcars$mileage, main = "Histogram of Used Car Mileage",
+ xlab = "Odometer (mi.)")
```



Visualizing Numeric Values – Histograms (3/3)

☞ The histogram is composed of a series of bars with heights indicating the count, or frequency of values falling within each of the equal width bins partitioning the values.

☞ The vertical lines that separate the bars, as labeled on the horizontal axis, indicate the start and end points of the range of values for the bin.

☞ On the price histogram:

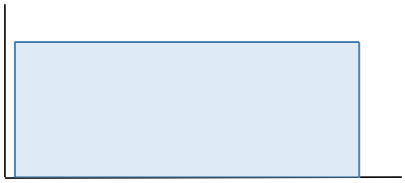
- Each of the 10 bars spans an interval of $2,000, beginning at $2,000 and ending at $22,000.
- The tallest bar at the center of the figure covers the $12,000 to $14,000 range and has a frequency of 50.
- Since we know that our data includes 150 cars, we know that one-third of all the cars are priced from $12,000 to $14,000.
- Nearly 90 cars—more than half—are priced from $12,000 to $16,000.

Uniform and Normal Distributions

☞ Histograms, boxplots, and statistics describing the center and spread provide ways to examine the distribution of a variable's values.

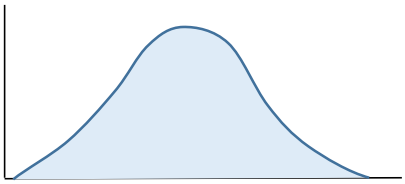
☞ A variable's distribution describes how likely a value is to fall within various ranges.

- A uniform distribution is easy to detect with a histogram, because the bars are approximately the same height.



Uniform Distribution

- Many real-world phenomena generate data that can be described by the normal (*bell-shaped*) distribution.



Normal Distribution

- Therefore, the normal distribution's properties have been studied in great detail.

Measuring Spread – Variance and Standard Deviation (1/2)

☞ Distributions allow us to characterize a large number of values using a smaller number of parameters.

☞ The normal distribution, which describes many types of real-world data, can be defined with just two: center and spread.

- The center of normal distribution is defined by its mean value, which we have used earlier.
- The spread is measured by a statistic called the standard deviation.

☞ In order to calculate the standard deviation, we must first obtain the variance, which is defined as the average of the squared differences between each value and the mean value.

- In mathematical notation, the variance of a set of n values of x is defined by the following formula:

$$\mathrm{Var(X)} = \sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

- The Greek letter μ denotes the mean of the values, and the variance itself is denoted by σ^2 (*sigma* squared).

Measuring Spread – Variance and Standard Deviation (2/2)

☞The standard deviation is the square root of the variance, and is denoted by *sigma*, as shown in the following formula:

$$\text{StdDev(X)} = \sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

☞The var() and sd() functions can be used to obtain the variance and standard deviation in R.

- For example, computing the variance and standard deviation on our price and mileage variables, we find:

```
> var(usedcars$price)
[1] 9749892
```

```
> var(usedcars$mileage)
[1] 728033954
```

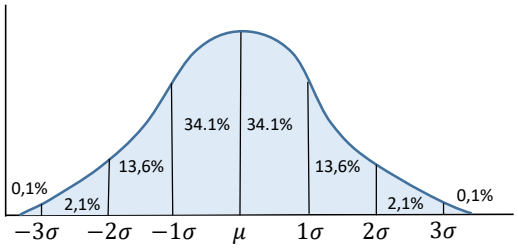
```
> sd(usedcars$price)
[1] 3122.482
```

```
> sd(usedcars$mileage)
[1] 26982.1
```


The 68-95-99.7 Rule

☞The standard deviation can be used to quickly estimate how extreme a given value is under the assumption that it came from a normal distribution.

- The 68-95-99.7 rule states that 68 percent of the values in a normal distribution fall within one standard deviation of the mean, while 95 percent and 99.7 percent of the values fall within two and three standard deviations, respectively.
- This is illustrated in the explanatory diagram.



Normal Distribution

- Applying this information to the used car data, we know that since the mean and standard deviation of price were $12,962 and $3,122, respectively, assuming that the prices are normally distributed, approximately 68 percent of cars in our data were advertised at prices between $12,962 - $3,122 = $9,840 and $12,962 + $3,122 = $16,804.

Exploring Categorical Variables (1/3)

- If you recall, the used car dataset had three categorical variables: model, color, and transmission.
  - Because we used the stringsAsFactors = FALSE parameter while loading the data, R has left them as the character (chr) type vectors rather than automatically converting them into factor type.
- Additionally, we might consider treating the year variable as categorical;
  - Although it has been loaded as a numeric (int) type vector, each year is a category that could apply to multiple cars.
- In contrast to numeric data, categorical data is typically examined using tables rather than summary statistics.
- A table that presents a single categorical variable is known as a one-way table.
  - The R table() function can be used to generate one-way tables for our used car data.

Exploring Categorical Variables (2/3)

```
> table(usedcars$year)
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012
   3    1    1    1    3    2    6   11   14   42   49   16    1
```

```
> table(usedcars$model)
 SE SEL SES
 78  23  49
```

```
> table(usedcars$color)
Black  Blue  Gold  Gray Green   Red Silver White Yellow
   35    17     1    16     5    25    32    16      3
```

- The table() output lists the categories of the nominal variable and a count of the number of values falling into this category.
  - Since we know that there are 150 used cars in the dataset, we can determine that roughly one-third of all the cars were manufactured in the year 2010, given that 49/150 = 0.327.

Exploring Categorical Variables (3/3)

- R can also perform the calculation of table proportions directly, by using the prop.table() command on a table produced by the table() function:

```
> model_table <- table(usedcars$model)
> prop.table(model_table)
    SE     SEL     SES
0.5200000 0.1533333 0.3266667
```

- The results of prop.table() can be combined with other R functions to transform the output.
  - Suppose that we would like to display the results in percentages with a single decimal place. We can do this by multiplying the proportions by 100, then using the round() function while specifying digits = 1, as shown in the following example:

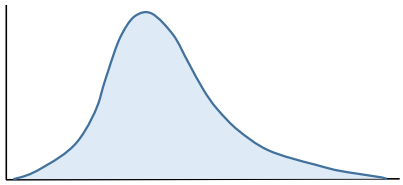
```
> color_table <- table(usedcars$color)
> color_pct <- prop.table(color_table) * 100
> round(color_pct, digits = 1)
 Black  Blue  Gold  Gray  Green   Red Silver  White Yellow
  23.3  11.3   0.7  10.7   3.3  16.7  21.3  10.7   2.0
```


Measuring the Central Tendency – The Mode

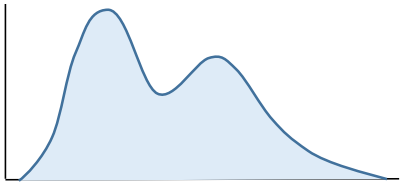
- In statistics terms, the mode of a feature is the value occurring most often.
  - Like the mean and median, the mode is another measure of central tendency.
- It is often used for categorical data, since the mean and median are not defined for nominal variables.
  - For example, in the used car data, the mode of the year variable is 2010, while the modes for the model and color variables are SE and Black, respectively.
  - A variable may have more than one mode; a variable with a single mode is unimodal, while a variable with two modes is bimodal. Data having multiple modes is more generally called multimodal.
- Although you might suspect that you could use the mode() function, R uses this to obtain the type of variable (as in numeric, list, and so on) rather than the statistical mode.
  - Instead, to find the statistical mode, simply look at the table output of the category with the greatest number of values.

The Mode for Numeric Data

- Thinking about modes as common values allows us to apply the concept of statistical mode to the numeric data.
  - Strictly speaking, it would be unlikely to have a mode for a continuous variable, since no two values are likely to repeat. Yet, if we think about modes as the highest bars on a histogram, we can discuss the modes of variables such as price and mileage.
  - It can be helpful to consider mode while exploring the numeric data, particularly to examine whether or not the data is multimodal.



Unimodal Distribution



Bimodal Distribution

Exploring Relationships Between Variables

- So far, we have examined variables one at a time, calculating only univariate statistics. During our investigation, we raised questions that we were unable to answer at that time:
  - Does the price data imply that we are examining only economy-class cars or are there also luxury cars with high mileage?
  - Do relationships between the model and color data provide insight into the types of cars we are examining?
- These type of questions can be addressed by looking at bivariate relationships, which consider the relationship between two variables.
- Relationships of more than two variables are called multivariate relationships.
- Let's begin with the bivariate case.

Visualizing Relationships - Scatterplots

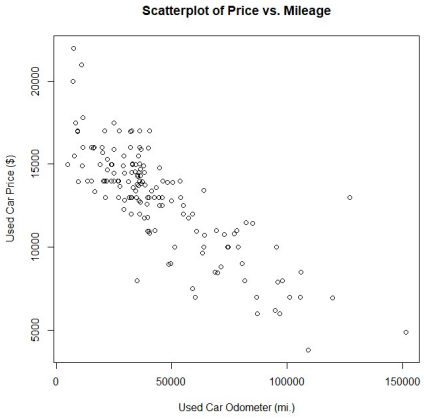
- A scatterplot is a diagram that visualizes a bivariate relationship.
  - It is a two-dimensional figure in which dots are drawn on a coordinate plane using the values of one feature to provide the horizontal x coordinates and the values of another feature to provide the vertical y coordinates.
  - Patterns in the placement of dots reveal the underlying associations between the two features.
- To answer our question about the relationship between price and mileage, we will examine a scatterplot.
  - We'll use the plot() function along with the main, xlab and ylab parameters used previously to label the diagram.

Scatterplot of **price** vs. **mileage** (1/2)

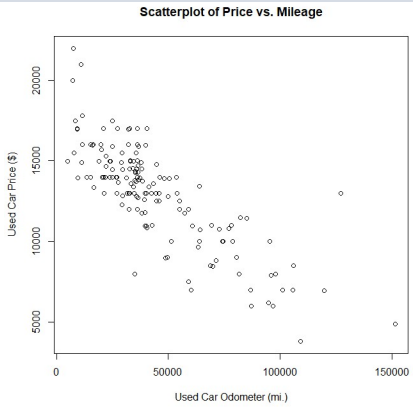
- To use plot(), we need to specify x and y vectors containing the values used to position the dots on the figure.
  - The y variable is the one that is presumed to depend on the other (and is therefore known as the dependent variable).
  - Our hypothesis is that the price depends on the odometer mileage. Therefore, we will use price as the y, or dependent, variable.
- The full command to create our scatterplot is:

```
> plot(x = usedcars$mileage, y = usedcars$price,
+ main = "Scatterplot of Price vs. Mileage",
+ xlab = "Used Car Odometer (mi.)",
+ ylab = "Used Car Price ($)")
```

  - Using the scatterplot, we notice a clear relationship between the price of a used car and the odometer reading.



Scatterplot of **price** vs. **mileage** (2/2)



- Perhaps a more interesting finding is the fact that there are very few cars that have both high price and high mileage, aside from a lone outlier at about 125,000 miles and $14,000.
  - The absence of more points like this provides evidence to support a conclusion that our data is unlikely to include any high mileage luxury cars.
  - All of the most expensive cars in the data, particularly those above $17,500, seem to have extraordinarily low mileage, implying that we could be looking at a brand new type of car retailing for about $20,000.

- The relationship we've found between car prices and mileage is known as a negative association, because it forms a pattern of dots in a line sloping downward.
  - A positive association would appear to form a line sloping upward.
  - A flat line, or a seemingly random scattering of dots, is evidence that the two variables are not associated at all. The strength of a linear association between two variables is measured by a statistic known as correlation.

Correlation Coefficient - Definition

- For pairs of variables measured on an interval or ratio scale, a correlation coefficient r can be calculated as follows:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{n\sigma_x\sigma_y}$$

x and y - variables,
x_i - individual values of x,
y_i - individual values of y,
$\bar{x}$ - the mean of the x variable,
$\bar{y}$ - the mean of the y variable,
σ_x and σ_y - the standard deviations of the variables x and y, respectively,
n - the number of observations.

- The values of r are within a range of $-1.0 \div 1.0$ and quantify the linear relationship between the variables.
- Positive numbers for r indicate a positive correlation between the pair of variables, and negative numbers indicate a negative correlation.
- A value of r close to 0 indicates little or no relationship between the variables.

Correlation Coefficient – An Example

☞ Now we are ready to calculate the correlation between car prices and mileage.

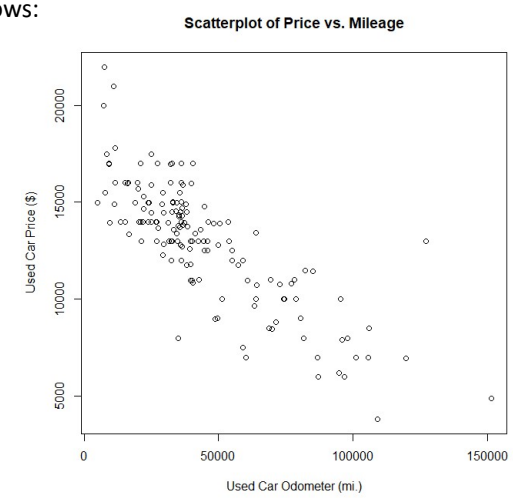
- This could be accomplished by the R function cor() as follows:

```
> cor(usedcars$mileage, usedcars$price)
[1] -0.8061494
```

- The computed correlation value fully confirms what we've already seen in the scatterplot.

☞ Keep in mind that not all associations form straight lines.

- Sometimes the dots form a *U* shape, or a *V* shape; sometimes the pattern seems to be weaker or stronger for increasing values of the x or y variable.
- Such patterns imply that the relationship between the two variables is not linear.



Two-Way Cross-Tabulations (1/4)

☞To examine a relationship between two nominal variables, a two-way cross-tabulation is used (also known as a crosstab or contingency table).

- A cross-tabulation is similar to a scatterplot in that it allows you to examine how the values of one variable vary by the values of another.
- The format is a table in which the rows are the levels of one variable, while the columns are the levels of another.
- Counts in each of the table's cells indicate the number of values falling into the particular row and column combination.

Two-Way Cross-Tabulations (2/4)

☞ To answer our earlier question about whether there is a relationship between car model and color, we will examine a crosstab.

- There are several functions to produce two-way tables in R, including table(), which we used for one-way tables.
- The CrossTable() option in the gmodels package is perhaps the most user-friendly function, as it presents the row, column, and margin percentages in a single table, saving us the trouble of combining this data ourselves.
- To install the gmodels package, type:

```
> install.packages("gmodels")
```

- After the package installs, type: library(gmodels) to load the package. You will need to do this during each R session in which you plan on using the CrossTable() function.

Two-Way Cross-Tabulations (3/4)

☞ Before proceeding with our analysis, let's simplify our project by reducing the number of levels in the color variable.

- This variable has nine levels, but we don't really need this much detail. What we are really interested in is whether or not the car's color is conservative.
- Toward this end, we'll divide the nine colors into two groups: the first group will include the conservative colors Black, Gray, Silver, and White; and the second group will include Blue, Gold, Green, Red, and Yellow.
- We will create a binary indicator variable (often called a dummy variable), indicating whether or not the car's color is conservative by our definition. Its value will be 1 if true, 0 otherwise:

```
> usedcars$conservative <-
+ usedcars$color %in% c("Black", "Gray", "Silver", "White")
```

- The %in% operator returns TRUE or FALSE for each value in the vector on the left-hand side of the operator depending on whether the value is found in the vector on the right-hand side.

Two-Way Cross-Tabulations (4/4)

- Examining the table() output for our newly created variable, we see that about two-thirds of the cars have conservative colors, while one-third do not have conservative colors:

```
> table(usedcars$conservative)
FALSE  TRUE
   51    99
```

- Now, let's look at a cross-tabulation to see how the proportion of conservatively colored cars varies by the model.
  - Since we're assuming that the model of the car dictates the choice of color, we'll treat the conservative color indicator as the dependent (y) variable.
  - The CrossTable() command is therefore:

```
> CrossTable(x = usedcars$model, y = usedcars$conservative)
```


The **CrossTable()** Output

- The preceding command results in the following table:
  - The rows in the table indicate the three models of used cars: SE, SEL, and SES (plus an additional row for the total across all models).
  - The columns indicate whether or not the car's color is conservative (plus a column totaling across both types of color).
- What we are most interested in is the row proportion for conservative cars for each model.
  - The row proportions tell us that 0.654 (65 percent) of SE cars are colored conservatively in comparison to 0.696 (70 percent) of SEL cars and 0.653 (65 percent) of SES.
  - These differences are relatively small, suggesting that there are no substantial differences in the types of colors chosen by the model of the car.

```
Cell Contents
|-------------------------|
|                       N |
| Chi-square contribution |
|           N / Row Total |
|           N / Col Total |
|         N / Table Total |
|-------------------------|

Total Observations in Table:  150

             | usedcars$conservative
usedcars$model |     FALSE |      TRUE | Row Total |
---------------|-----------|-----------|-----------|
            SE |        27 |        51 |        78 |
               |     0.009 |     0.004 |           |
               |     0.346 |     0.654 |     0.520 |
               |     0.529 |     0.515 |           |
               |     0.180 |     0.340 |           |
---------------|-----------|-----------|-----------|
           SEL |         7 |        16 |        23 |
               |     0.086 |     0.044 |           |
               |     0.304 |     0.696 |     0.153 |
               |     0.137 |     0.162 |           |
               |     0.047 |     0.107 |           |
---------------|-----------|-----------|-----------|
           SES |        17 |        32 |        49 |
               |     0.007 |     0.004 |           |
               |     0.347 |     0.653 |     0.327 |
               |     0.333 |     0.323 |           |
               |     0.113 |     0.213 |           |
---------------|-----------|-----------|-----------|
  Column Total |        51 |        99 |       150 |
               |     0.340 |     0.660 |           |
---------------|-----------|-----------|-----------|
```