CSI 2300: Introduction to Data Science

Lecture 07: Exploratory Data Analysis

Today's Topics

Histograms

- Basics
- Common Syntax
- Modality
- Skewness

Boxplots

- Basics
- Common Syntax
- Comparing two distributions

Scatterplots

- Basics
- Common Syntax
- Investigating the relationship between two variables

A Short History Lesson

Famous statistician John Tukey defined 'data analysis' as: "Procedures for analyzing data, techniques for interpreting the results of such procedures, ways of planning the gathering of data to make its analysis easier, more precise or more accurate, and all the machinery and results of (mathematical) statistics which apply to analyzing data" [1]. Though Tukey didn't create the idea of exploratory data analysis (EDA), its original concepts could be traced back to an assortment of different scholars, he did write one of the preeminent books on the subject [2]. Tukey hoped future statisticians would explore beyond the traditional modeling and hypothesis testing that was so prevalent during his early career. There is so much more information, new hypothesis that can be formed, and questions asked when you are able to explore the data visually. This backing of exploratory data analysis led to the development of statistical computing programs such as 'S' which is the precursor to 'R.'

[1] Tukey, J. W. (1962). The future of data analysis. The annals of mathematical statistics, 33(1), 1-67.

[2] Tukey, John W. (1977). Exploratory Data Analysis. Pearson.

Histograms

The idea behind a histogram is to divide the range of data (the range is the maximum value minus the minimum) into equal size bins and then categorize each observation into each of these bins. Then, draw bars such that the height of each bar represents how many observations are in that bin. The heights can be counts or proportions of observations. The width of each bin can impact the look of the histogram.

The syntax for histograms in R is fairly basic:

```
hist(v,main,xlab,xlim,ylim,breaks,col,border)
```

Where each of the parameters within the '( )', outside of 'v', are predominantly optional for the call and used to help augment the overall visual display of the data. Here is a breakdown of the individual parameters:

- **v** is a vector containing numeric values used in histogram.
- **main** indicates title of the chart.
- **col** is used to set color of the bars.
- **border** is used to set border color of each bar.
- **xlab** is used to give description of x-axis.
- **xlim** is used to specify the range of values on the x-axis.
- **ylim** is used to specify the range of values on the y-axis.
- **breaks** is used to set the number of 'bins' in the histogram.

In the following code chunk, we will be creating some simple data and plotting three different histograms where the only differences will be the number of bins in each of the plats.

```
set.seed(77)

nn <- 1000
data <- c(rnorm(nn, 6, 2), rnorm(nn, 12, 1))

par(mfrow=c(1, 3))
hist(data, breaks=4, main="", xlab="4 Bins")
hist(data, main="", xlab="Default Bins")
hist(data, breaks=100, main="", xlab="100 Bins")
```

Important Features in Histograms

- **Min/Max/Range:** From a histogram, you can easily see the lowest value, the highest value, and the range, which is the difference between the highest and lowest values.
- **Modality:** How many peaks a histogram has. Unimodal for one, bimodal for two, etc.

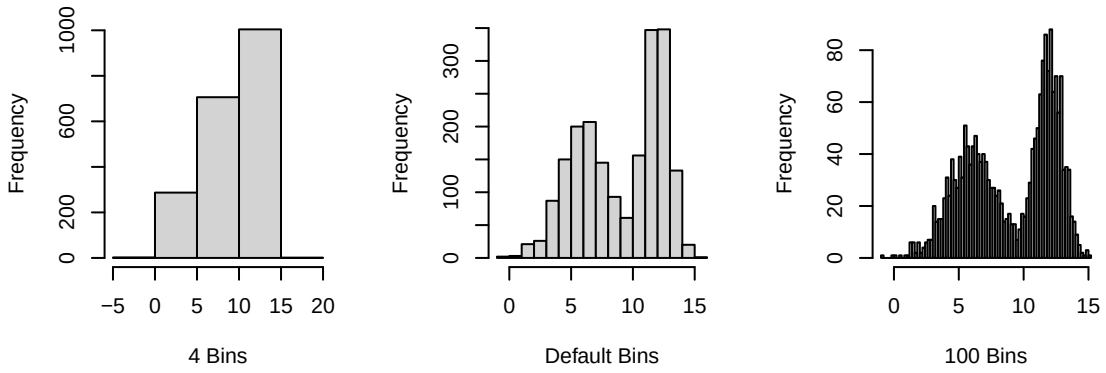


Figure 1: The same data are plotted in each of these histograms; only the bin width is differs.

- **Symmetry:** If a line were drawn vertically down the middle, does the left side match the right side?
- **Skewness:** If the data has a long tail on the left (skewed left) or on the right (skewed right). Can you think of any variables that may be skewed one way or the other?
- **Kurtosis:** In the upper and lower tails (the left and right sides of the histogram), the kurtosis is a measure of how much area is in the tail. A heavy-tailed distribution has a lot of area, and a light-tailed distribution has very little area in the tails. This is often difficult to see with your eyes.

How would you characterize these histograms?

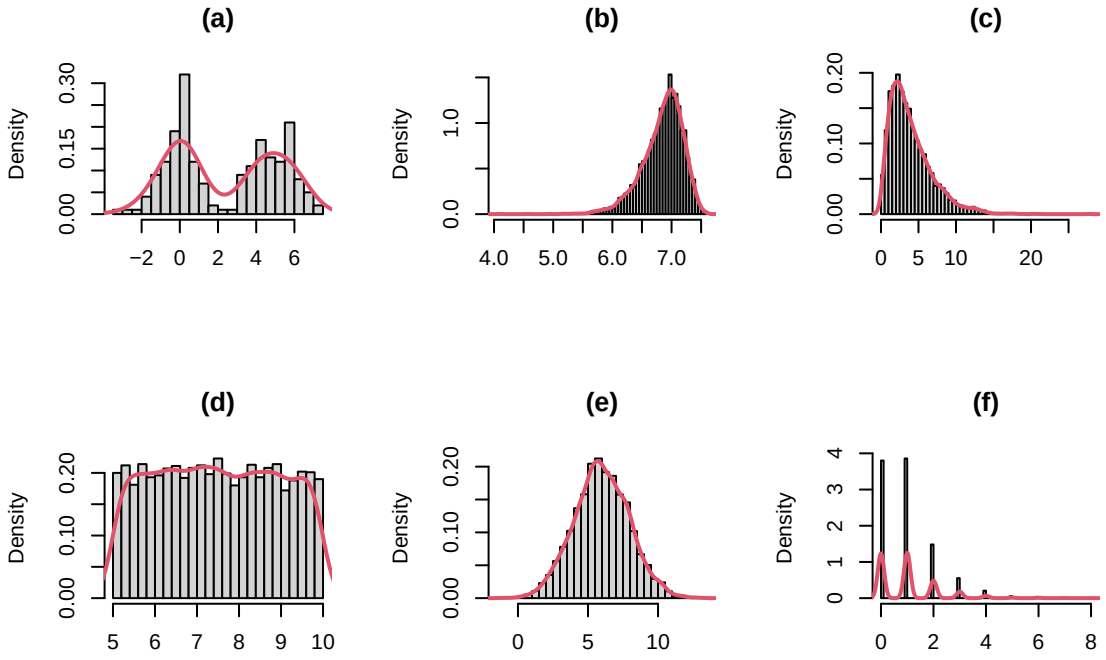


Figure 2: Histograms with different shapes and features.

Example, Boulder Home Sales: Boulder County tracks the real estate sales by year, and the data can be downloaded here: https://www.bouldercounty.org/property-and-land/assessor/sales/recent/ Plot the histogram of sales prices in 2019 and in 2020. What pattern do you see?

```
sales2019 <- read.csv(file="dat/boulder-2019-residential_sales.csv",
                      header = T, stringsAsFactors = F)
sales2020 <- read.csv(file="dat/boulder-2020-residential_sales.csv",
                      header = T, stringsAsFactors = F)

colnames(sales2019)
str(sales2019)
hist(sales2019$SALE_PRICE)
# Error in hist.default(sales2019$SALE_PRICE): 'x' must be numeric

#A little bit of wrangling to get rid of the commas and dollar signs
sales2019$SALE_PRICE <- gsub(",", "", sales2019$SALE_PRICE)
sales2019$SALE_PRICE <- as.numeric(gsub("\\$", "", sales2019$SALE_PRICE))
sales2020$SALE_PRICE <- gsub(",", "", sales2020$SALE_PRICE)
sales2020$SALE_PRICE <- as.numeric(gsub("\\$", "", sales2020$SALE_PRICE))

hist(sales2019$SALE_PRICE, breaks="FD")
summary((sales2019$SALE_PRICE))
hist(sales2019$SALE_PRICE/10000, breaks="FD", xlim=c(0,100))

hist(sales2020$SALE_PRICE, breaks="FD")
summary((sales2020$SALE_PRICE))
hist(sales2020$SALE_PRICE/10000, breaks="FD", xlim=c(0,100))
```

Boxplots

A boxplot is a graphical representation of how well the data is distributed within a data set. It divides the inputed data set into three quartiles (25%, 50%, 75%), with the central 50% of data plotted within a "box." The top of the box is the 75th quantile (the value that separates the bottom 75% of observation from the top 25% of observations), and the bottom of the box is the 25th quantile (the value that separates the bottom 25% of observation from the top 75% of observations)

- The median is plotted as a line through the middle of the box.
- The "whiskers" of the plot extend to no more than a certain multiplier of the range from the 25th and 75th quantiles.
- Potential "outliers" are flagged with a special point that is beyond the end of the

whiskers. These are observations whose values are unusual with respect to the rest of the observations.

- Can be plotted vertically or horizontally.

The basic syntax for boxplot is:

```
boxplot(x, data, notch, varwidth, names, main)
```

Where in the parameters identified correspond to the following options:

- **x** is a vector or a formula.
- **data** is the data frame.
- **notch** is a logical value. Set as TRUE to draw a notch.
- **varwidth** is a logical value. Set as true to draw width of the box proportionate to the sample size.
- **names** are the group labels which will be printed under each boxplot.
- **main** is used to give a title to the graph.

As with Histograms, Boxplots are able to be thoroughly controlled. This allows us to represent the data in whatever visual manner we deem appropriate. The following example plots the same data both horizontally and vertically.

```
par(mfrow = c(1, 2))
set.seed(77)
nn <- 1000
data <- c(rnorm(nn, 6, 2), rnorm(nn, 12, 1))
boxplot(c(data, 23), pch=8, ylim=c(-5, 25), horizontal = T)
boxplot(c(data, 23), pch=8, ylim=c(-5, 25))
```

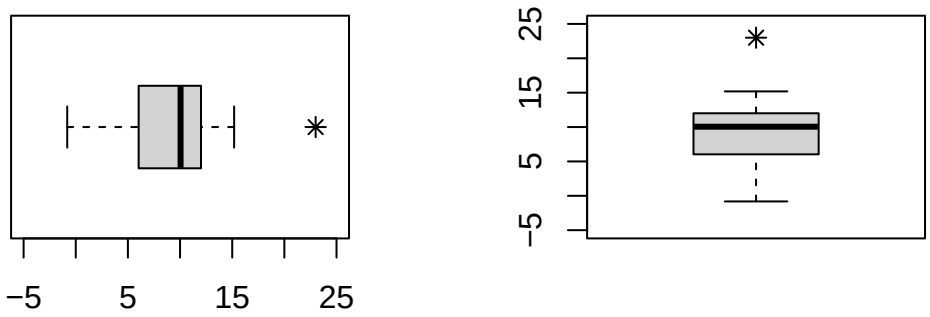


Figure 3: Boxplots plotted either vertically or horizontally.

Important Features in Boxplots

- **Center/Spread Comparison:** Compare the centers and spreads of the variables plotted by a particular feature of interest. For example, you could plot GPA by major

or gas mileage by make & model of car.

- **Outliers:** How many outliers does a variable have, and in which direction are they (high or low)?
- **Symmetry:** Symmetric distributions have medians in the center of the box with whiskers that extend approximately the same distance on either side.

Example, Boulder Home Sales: Plot the side-by-side boxplots of home sale price in 2019 versus 2020. What do you see?

```
n2019 <- nrow(sales2019)
n2020 <- nrow(sales2020)

year <- c(rep(2019, n2019), rep(2020, n2020))
sales <- c(sales2019$SALE_PRICE, sales2020$SALE_PRICE)
boxplot(sales ~ year)
boxplot(sales ~ year, ylim = c(0, 2000000))
```

Scatterplots

Scatterplots are a plot of ordered pairs, and ordered pairs are generally two quantitative variables measured on the same individual.

The basic syntax for a scatterplot call is:

```
plot(x, y, main, xlab, ylab, xlim, ylim, axes)
```

Where the parameters of the call correspond to: * **x** is the data set whose values are the horizontal coordinates.

- **y** is the data set whose values are the vertical coordinates.
- **main** is the tile of the graph.
- **xlab** is the label in the horizontal axis.
- **ylab** is the label in the vertical axis.
- **xlim** is the limits of the values of x used for plotting.
- **ylim** is the limits of the values of y used for plotting.
- **axes** indicates whether both axes should be drawn on the plot.

One of the main uses for scatterplots is to visualize two variables and try to identify any trends in their relationship to each other. In the Boulder home dataset, there are many variables. Which do you think might be "predictive" of the sales price? A good choice would

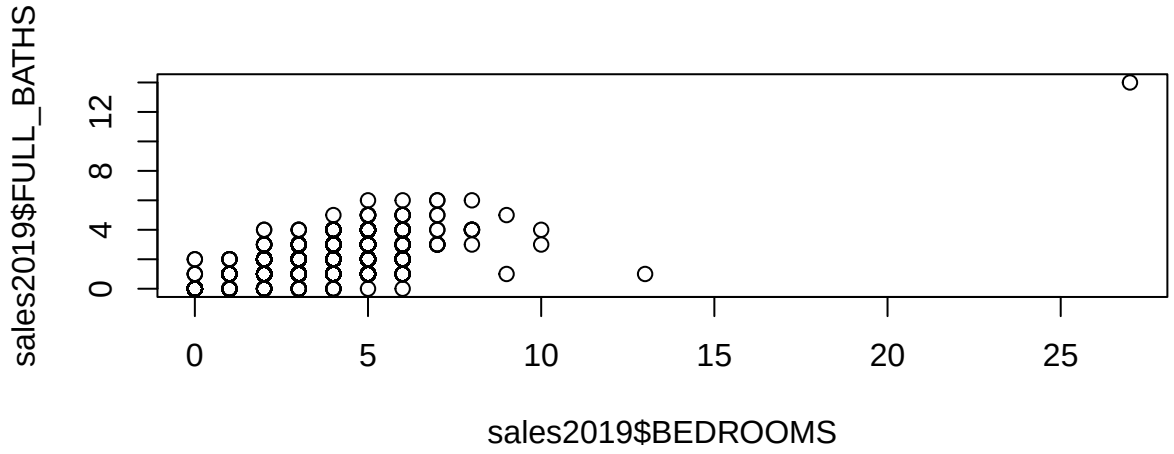
be square footage of the house. Lets plot that variable on the x-axis and the sales price on the y-axis. The following are three different versions of the scatterplot with various parameters set.

```
plot(sales2019$ABOVE_GROUND_SQFT, sales2019$SALE_PRICE)
plot(sales2019$ABOVE_GROUND_SQFT, sales2019$SALE_PRICE, pch=19,
     col = rgb(0,0,0,.1))
plot(sales2019$ABOVE_GROUND_SQFT, sales2019$SALE_PRICE, pch=19,
     col = rgb(0,0,0,.1), ylim=c(0,5000000))
```

In scatterplots, you are also looking for the following:

- **Shape:** The shape of the relationship between the two variables, generally do you see something that could follow a straight line (linear), or is there curvature present?
- **Outliers:** Look for observations that are very different from the others in either the x-direction, the y-direction, or both.

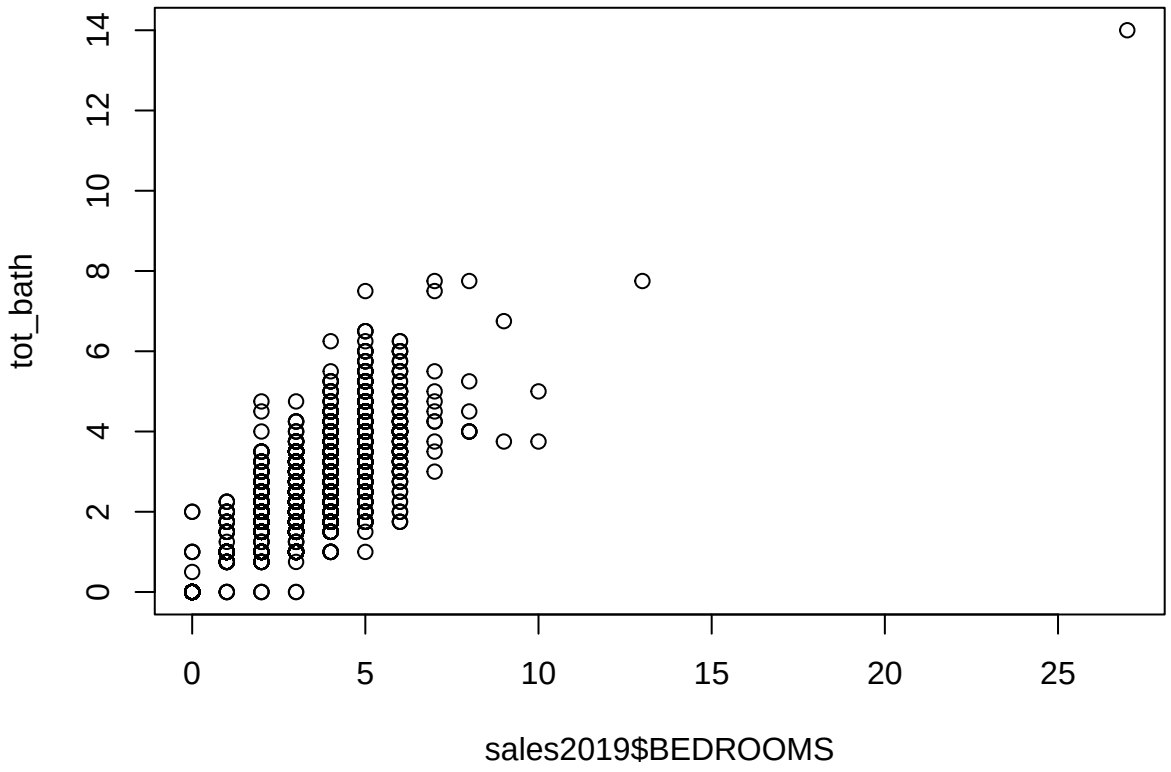
```
plot(sales2019$BEDROOMS, sales2019$FULL_BATHS)
```



But this doesn't tell the whole story since our data has a little more information regarding the bathroom situations. Lets combine the values of these three variables and replot.

```
tot_bath <- sales2019$FULL_BATHS + (0.75 * sales2019$THREE_QTR_BATHS) +
        (0.5 * sales2019$HALF_BATHS)

plot(sales2019$BEDROOMS, tot_bath)
```



Let's try and compare the sales price with the total number of bathrooms. What do we see here?

```
plot(sales2019$SALE_PRICE, tot_bath, xlim = c(0, 3500000), ylim = c(0, 8))
```