390R EDA2 notes
11/3/2020

We've talked about analyzing variation in one variable. We want to know extreme values, clusters, missing values…

When you have missing values or outliers you can

1. Remove those observations using the filter function to get rid of certain values.
   A problem you might have is that you override the original data. One way to recover the original data is rm(data)
2. (Recommended) You can replace the missing/outliers explicitly with an na value in r studio. ifelse() function.
   ifelse(true or false,value you want to have when the first argument = true, value you want to have when the first argument = false)
   x<- TRUE
   ifelse(x,10,1) outputs 10

You can use ifelse with functions
Example: ifelse(x>5,3*x,0) so if the x value is larger than 5 it will multiple by 3, if not then 0
How to use it to get rid of unusual values:
diamonds%>%mutate(y= ifelse(y<3 | y>20, NA, y))

Other times you want to understand what makes observations with missing values different to observations with recorded values. For example, in flights data, missing values in the dep_time variable indicate that the flight was cancelled. So you might want to compare the scheduled departure times for cancelled and non-cancelled times.

First, we can create a cancelled indicator to tell us if the flight was cancelled. We will also use a date type for the time variable to make a formal date time format for the function.
flights%>%mutate(cancelled = is.na(dep_time), time = hms::hms(hour = hour, minute = minute))
Then we can make a histogram
ggplot(mapping = aes(time)) +
  geom_histogram(mapping = aes(fill = cancelled), binwidth = 20) this will set binwidth to 20 seconds
The problem with this is that we can't see the difference between the cancelled and non cancelled flights. This is a problem when you display a distribution of a continuous variable split by a categorical variable. Initially the histogram works well when displaying one continuous variable but when you want to split by a categorical variable.
Instead, we can use geom_freqpoly. Same as histogram but instead of bars it uses lines to represent counts.
While this plot is better, it would be better to find the proportions of cancelled flights. We can do this with stat = "density"
geom_freqpoly(mapping = aes(color = cancelled), stat = "density")
This makes a density plot. Changes counts in y axis to density. It's the proportion for continuous variables. For discrete variables, the sum of the proportions across different categories is 1. So the area under each curve is 1.
One benefit to the density function is that we can ignore the difference in the total number of flights because all proportions are divided by total number of counts in each type.

Second type of variation: Covariation

Covariation describes the behavior between variables. Covariation is the tendency for the values of two or more variables to vary together in a related way. The best way to spot covariation is to visualize the relationship between two or more variables.

Examples: Is the height of a person related to their weight?

What is the relationship between two or more variables.

Is one variable positively associated with another variable?

Depending on the types of variables you can draw different graphs.

3 categories

1 categorical variable and 1 continuous variable.

1 way is to use histograms, another way is freqpoly.

Use diamonds data set to compare cut (categorical) with price (Continuous).

```
ggplot(diamonds)+
  geom_histogram(aes(x=price,fill = cut))
```

We can add in a density function to see the proportion

```
ggplot(diamonds)+
  geom_histogram(aes(x=price,fill = cut), stat=”density”)
```

Conclusion: The ideal diamonds have a larger proportion than others in the lower price range.

Another way to display 1 categorical variable and 1 continuous variable is a boxplot.

```
ggplot(diamonds)+
  geom_boxplot(aes(y=price,x = cut))
```

We can reorder the x variable to display the plots by the median.

```
geom_boxplot(aes(y=price,x =fct_reorder(cut,price))))
```

The boxplot doesn’t show each variable distribution, only the summary stats like mean and quartiles.

IF you want to know more details you can use geom_violin to show the distributions of each variable.

Question: Use boxplot to improve the visualization of the scheduled departure times of cancelled vs. non-cancelled flights.

```
flights %>%
  mutate(
    cancelled = is.na(dep_time),
   time = hms::hms(hour = hour, minute = minute)
   ) %>%
  ggplot() +
    geom_boxplot(aes(y=time, x = cancelled))
```

Now we can see that cancelled flights has a larger median than non cancelled meaning the scheduled dep time are on average later than the non cancelled flights. Same conclusion we got from freqpoly, but this one isn’t as clear as we can see the distributions.

Next category

2 categorical variables

Barchart is popular for this

Lets look at cut and color variables.

```
ggplot(diamonds)+geom_bar(aes(x=color,fill=cut))
```

Can we use a different order, can we switch the variables? You should think of which way will produce a better, simpler barchart.

You can use position adjustments to make it easier to see.

ggplot(diamonds)+geom_bar(aes(x=color,fill=cut),position=”dodge”) which make the bars with separated colors side by side so it’s easier to get the counts of the bars compared to the default setting.

ggplot(diamonds)+geom_bar(aes(x=color,fill=cut),position=”fill”) Will rescale the height of the bars to 1 so you can see the proportion of each category.

Geom_tile is useful for displaying relationships between categorical variables. It’s like a heat map for variables.

```
diamonds%>%group_by(color,cut)%>%summarize(n=n())
%>%ggplot()+geom_tile(aes(x=color,y=cut,fill=n))
```

Divides the graph into a grid, each color depends on n, the counts of the diamonds corresponding to each cut.

We can show the distribution of cut within color, calculate a new variable prop which is the proportion of each cut within a color

```
diamonds %>%
  count(color, cut) %>%
  group_by(color) %>%
  mutate(prop = n / sum(n)) %>%
  ggplot(mapping = aes(x = color, y = cut)) +
  geom_tile(mapping = aes(fill = prop))
```

The color is corresponding to the proportion instead of the counts.

Category 3

2 continuous variables

Carat and price

If we want to display the relationship between these variables we can use a scatterplot

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

You could change a continuous variable to a categorical variable and use one of the other options.

Change carat variable into categorical and use a boxplot to draw relationship between price

```
ggplot(diamonds)+geom_boxplot(aes(x=carat,y=price,group=cut_width(carat,0.1)))
```

We cut the width of the boxes into 0.1 carat segments and drew the boxplot for each carat value.

Conclusion: Middle lines are in increasing order, as the size increases the price increases.