ExploratoryDataAnalysis

Prof. Eric A. Suess

September 11, 2019

Today we will discuss Exploratory Data Analysis (EDA).

This is the process of exploring your data using visualization and transformations and modeling (will discuss modeling more later).

library(tidyverse)

Lets take a look at the *diamonds* data set and the variable carat.

diamonds

## # A tibble: 53,940 x 10
## carat cut color clarity depth table price x y z
## <dbl> <ord> <ord> <ord> <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23 Ideal E SI2 61.5 55 326 3.95 3.98 2.43
## 2 0.21 Premium E SI1 59.8 61 326 3.89 3.84 2.31
## 3 0.23 Good E VS1 56.9 65 327 4.05 4.07 2.31
## 4 0.290 Premium I VS2 62.4 58 334 4.2 4.23 2.63
## 5 0.31 Good J SI2 63.3 58 335 4.34 4.35 2.75
## 6 0.24 Very Good J VVS2 62.8 57 336 3.94 3.96 2.48
## 7 0.24 Very Good I VVS1 62.3 57 336 3.95 3.98 2.47
## 8 0.26 Very Good H SI1 61.9 55 337 4.07 4.11 2.53
## 9 0.22 Fair E VS2 65.1 61 337 3.87 3.78 2.49
## 10 0.23 Very Good H VS1 59.4 61 338 4 4.05 2.39
## # ... with 53,930 more rows

ggplot(data = diamonds) +
 geom\_histogram(mapping = aes(x = carat), binwidth = 0.5)



diamonds %>%
 count(cut\_width(carat, 0.5))

## # A tibble: 11 x 2
## `cut\_width(carat, 0.5)` n
## <fct> <int>
## 1 [-0.25,0.25] 785
## 2 (0.25,0.75] 29498
## 3 (0.75,1.25] 15977
## 4 (1.25,1.75] 5313
## 5 (1.75,2.25] 2002
## 6 (2.25,2.75] 322
## 7 (2.75,3.25] 32
## 8 (3.25,3.75] 5
## 9 (3.75,4.25] 4
## 10 (4.25,4.75] 1
## 11 (4.75,5.25] 1

Looking at the smaller diamonds.

smaller <- diamonds %>%
 filter(carat < 3)

diamonds %>% ggplot(mapping = aes(x = carat)) +
 geom\_histogram(binwidth = 0.1)



Look at carat by cut.

smaller %>% ggplot(mapping = aes(x = carat, colour = cut)) +
 geom\_freqpoly(binwidth = 0.1)



Looking for *typical values*.

smaller %>% ggplot(mapping = aes(x = carat)) +
 geom\_histogram(binwidth = 0.01)



Looking for *unusual values*. Lets look at the *y* variable.

diamonds %>% ggplot(mapping = aes(x = y)) +
 geom\_histogram(binwidth = 0.5)



Are there outliers?

diamonds %>% ggplot(mapping = aes(x = y)) +
 geom\_histogram(binwidth = 0.5) +
 coord\_cartesian(ylim = c(0, 50))



Lets find the outliers.

unusual <- diamonds %>%
 filter(y < 3 | y > 20) %>%
 select(price, x, y, z) %>%
 arrange(y)
unusual

## # A tibble: 9 x 4
## price x y z
## <int> <dbl> <dbl> <dbl>
## 1 5139 0 0 0
## 2 6381 0 0 0
## 3 12800 0 0 0
## 4 15686 0 0 0
## 5 18034 0 0 0
## 6 2130 0 0 0
## 7 2130 0 0 0
## 8 2075 5.15 31.8 5.12
## 9 12210 8.09 58.9 8.06

Remove outliers.

diamonds2 <- diamonds %>%
 filter(between(y, 3, 20))

Better to convert them to **NA**, which means not available.

diamonds2 <- diamonds %>%
 mutate(y = ifelse(y < 3 | y > 20, NA, y))

Scatterplots.

diamonds2 %>% ggplot(mapping = aes(x = x, y = y)) +
 geom\_point()

## Warning: Removed 9 rows containing missing values (geom\_point).



ggplot(data = diamonds2, mapping = aes(x = x, y = y)) +
 geom\_point(na.rm = TRUE)



Categorical variable. cut

diamonds %>% ggplot(mapping = aes(x = cut)) +
 geom\_bar()



Continuous variable. price

diamonds %>% ggplot(mapping = aes(x = price, y = ..density..)) +
 geom\_freqpoly(mapping = aes(colour = cut), binwidth = 500)



Putting them together in one plot.

diamonds %>% ggplot(mapping = aes(x = cut, y = price)) +
 geom\_boxplot()



For a different data set. mpg

mpg %>% ggplot(mapping = aes(x = class, y = hwy)) +
 geom\_boxplot()



Re-order.

mpg %>% ggplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy)) +
 geom\_boxplot()



Flip.

mpg %>% ggplot(mapping = aes(x = reorder(class, hwy, FUN = median), y = hwy)) +
 geom\_boxplot() +
 coord\_flip()

