Homework3

Steven Black

2/18/2019

Use the mod_diamonds dataset attached to this assignment. The dataset is a modified version of the diamonds data you have seen before in that some of the price information is not available (NA) and the price ranges vary based on the quality of cuts.

1. Load the mod_diamonds data into R.

```
mod_diamonds <- read.table("mod_diammonds.txt", header = TRUE, sep = ",")
head(mod_diamonds)
```

```
##   carat       cut color clarity depth table price    x    y    z
## 1  0.23     Ideal     E     SI2  61.5    55 14262 3.95 3.98 2.43
## 2  0.21   Premium     E     SI1  59.8    61  9329 3.89 3.84 2.31
## 3  0.23      Good     E     VS1  56.9    65    NA 4.05 4.07 2.31
## 4  0.29   Premium     I     VS2  62.4    58  9472 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58  5453 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57    NA 3.94 3.96 2.48
##          V11
## 1 0.87172410
## 2 0.25750686
## 3 0.05680521
## 4 0.56571462
## 5 0.50120086
## 6 0.07153610
```

2. Write a script in R to show the number of records whose price information is not available.

```
no_price_rows <- mod_diamonds[is.na(mod_diamonds$price),]
paste("There are", length(no_price_rows), "records where price information is not available. ")
```

```
## [1] "There are 11 records where price information is not available. "
```

3. Find the average price of all diamonds without counting those where the price is NA.

```
mean(mod_diamonds$price, na.rm = TRUE)
```

```
## [1] 9414.753
```

4. Find the average price for each type of cut.

```r
cut_price <- mod_diamonds[, c("cut", "price")]
aggregate(cut_price[,2], list(cut_price$cut), mean, na.rm = TRUE)
```

```
##     Group.1         x
## 1      Fair  2998.482
## 2      Good  4999.898
## 3     Ideal 12506.258
## 4   Premium  9006.758
## 5 Very Good  7003.122
```

5. Write a script to replace the prices where the values are not available (NA) with the appropriate mean price value. State and justify which average price you will replace the NAs with.

```r
head(cut_price)
```

```
##         cut price
## 1     Ideal 14262
## 2   Premium  9329
## 3      Good    NA
## 4   Premium  9472
## 5      Good  5453
## 6 Very Good    NA
```

```r
# install.packages("dplyr")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
cut_price_replace_na <- cut_price %>%
  group_by(cut) %>%
  mutate_if(is.numeric, function(value) ifelse(is.na(value), mean(value, na.rm = TRUE), value))
```

```
## `mutate_if()` ignored the following grouping variables:
## Column `cut`
```

```r
head(cut_price_replace_na)
```

```
## # A tibble: 6 x 2
## # Groups:   cut [4]
##   cut        price
##   <fct>      <dbl>
## 1 Ideal     14262
## 2 Premium    9329
## 3 Good       5000.
## 4 Premium    9472
## 5 Good       5453
## 6 Very Good  7003.
```

As shown above the values which were previously `NA` have been replaced by the average for that group. Replacing missing values by the average is a data engineering technique which can be used to make previously unusable data amenable to data analysis and machine learning. Replacing all the missing values by the mean for all the groups would work but replacing them by the mean for their specific group yields a more accurate representation of what the data would look like if these values were not missing.