Chapter-4-Figures-Code.R

Stephanie

2025-03-31

#### Chapter 4 Coding Exercises ####  
  
#### Table 4.3: Summary Statistics for the Boston Housing Data ####  
  
boston.housing.df <- read.csv("BostonHousing.csv", header = TRUE)  
head(boston.housing.df, 9)

## CRIM ZN INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO LSTAT MEDV  
## 1 0.00632 18.0 2.31 0 0.538 6.575 65.2 4.0900 1 296 15.3 4.98 24.0  
## 2 0.02731 0.0 7.07 0 0.469 6.421 78.9 4.9671 2 242 17.8 9.14 21.6  
## 3 0.02729 0.0 7.07 0 0.469 7.185 61.1 4.9671 2 242 17.8 4.03 34.7  
## 4 0.03237 0.0 2.18 0 0.458 6.998 45.8 6.0622 3 222 18.7 2.94 33.4  
## 5 0.06905 0.0 2.18 0 0.458 7.147 54.2 6.0622 3 222 18.7 5.33 36.2  
## 6 0.02985 0.0 2.18 0 0.458 6.430 58.7 6.0622 3 222 18.7 5.21 28.7  
## 7 0.08829 12.5 7.87 0 0.524 6.012 66.6 5.5605 5 311 15.2 12.43 22.9  
## 8 0.14455 12.5 7.87 0 0.524 6.172 96.1 5.9505 5 311 15.2 19.15 27.1  
## 9 0.21124 12.5 7.87 0 0.524 5.631 100.0 6.0821 5 311 15.2 29.93 16.5  
## CAT..MEDV  
## 1 0  
## 2 0  
## 3 1  
## 4 1  
## 5 1  
## 6 0  
## 7 0  
## 8 0  
## 9 0

summary(boston.housing.df)

## CRIM ZN INDUS CHAS   
## Min. : 0.00632 Min. : 0.00 Min. : 0.46 Min. :0.00000   
## 1st Qu.: 0.08205 1st Qu.: 0.00 1st Qu.: 5.19 1st Qu.:0.00000   
## Median : 0.25651 Median : 0.00 Median : 9.69 Median :0.00000   
## Mean : 3.61352 Mean : 11.36 Mean :11.14 Mean :0.06917   
## 3rd Qu.: 3.67708 3rd Qu.: 12.50 3rd Qu.:18.10 3rd Qu.:0.00000   
## Max. :88.97620 Max. :100.00 Max. :27.74 Max. :1.00000   
## NOX RM AGE DIS   
## Min. :0.3850 Min. :3.561 Min. : 2.90 Min. : 1.130   
## 1st Qu.:0.4490 1st Qu.:5.886 1st Qu.: 45.02 1st Qu.: 2.100   
## Median :0.5380 Median :6.208 Median : 77.50 Median : 3.207   
## Mean :0.5547 Mean :6.285 Mean : 68.57 Mean : 3.795   
## 3rd Qu.:0.6240 3rd Qu.:6.623 3rd Qu.: 94.08 3rd Qu.: 5.188   
## Max. :0.8710 Max. :8.780 Max. :100.00 Max. :12.127   
## RAD TAX PTRATIO LSTAT   
## Min. : 1.000 Min. :187.0 Min. :12.60 Min. : 1.73   
## 1st Qu.: 4.000 1st Qu.:279.0 1st Qu.:17.40 1st Qu.: 6.95   
## Median : 5.000 Median :330.0 Median :19.05 Median :11.36   
## Mean : 9.549 Mean :408.2 Mean :18.46 Mean :12.65   
## 3rd Qu.:24.000 3rd Qu.:666.0 3rd Qu.:20.20 3rd Qu.:16.95   
## Max. :24.000 Max. :711.0 Max. :22.00 Max. :37.97   
## MEDV CAT..MEDV   
## Min. : 5.00 Min. :0.000   
## 1st Qu.:17.02 1st Qu.:0.000   
## Median :21.20 Median :0.000   
## Mean :22.53 Mean :0.166   
## 3rd Qu.:25.00 3rd Qu.:0.000   
## Max. :50.00 Max. :1.000

# compute mean, standard dev., min, max, median, length, and missing values   
# of CRIM  
mean(boston.housing.df$CRIM) # mean

## [1] 3.613524

sd(boston.housing.df$CRIM) # standard deviation

## [1] 8.601545

min(boston.housing.df$CRIM) # minimum

## [1] 0.00632

max(boston.housing.df$CRIM) # maximum

## [1] 88.9762

median(boston.housing.df$CRIM) # median

## [1] 0.25651

length(boston.housing.df$CRIM)

## [1] 506

# find the number of missing values of variable CRIM  
sum(is.na(boston.housing.df$CRIM))

## [1] 0

# compute mean, standard dev., min, max, median, length, and missing values for   
# all variables.  
data.frame(mean = sapply(boston.housing.df, mean),   
 sd = sapply(boston.housing.df, sd),   
 min = sapply(boston.housing.df, min),   
 max = sapply(boston.housing.df, max),   
 median = sapply(boston.housing.df, median),   
 length = sapply(boston.housing.df, length),   
 miss.val = sapply(boston.housing.df, function(x)  
 sum(length(which(is.na(x))))) )

## mean sd min max median length miss.val  
## CRIM 3.61352356 8.6015451 0.00632 88.9762 0.25651 506 0  
## ZN 11.36363636 23.3224530 0.00000 100.0000 0.00000 506 0  
## INDUS 11.13677866 6.8603529 0.46000 27.7400 9.69000 506 0  
## CHAS 0.06916996 0.2539940 0.00000 1.0000 0.00000 506 0  
## NOX 0.55469506 0.1158777 0.38500 0.8710 0.53800 506 0  
## RM 6.28463439 0.7026171 3.56100 8.7800 6.20850 506 0  
## AGE 68.57490119 28.1488614 2.90000 100.0000 77.50000 506 0  
## DIS 3.79504269 2.1057101 1.12960 12.1265 3.20745 506 0  
## RAD 9.54940711 8.7072594 1.00000 24.0000 5.00000 506 0  
## TAX 408.23715415 168.5371161 187.00000 711.0000 330.00000 506 0  
## PTRATIO 18.45553360 2.1649455 12.60000 22.0000 19.05000 506 0  
## LSTAT 12.65306324 7.1410615 1.73000 37.9700 11.36000 506 0  
## MEDV 22.53280632 9.1971041 5.00000 50.0000 21.20000 506 0  
## CAT..MEDV 0.16600791 0.3724560 0.00000 1.0000 0.00000 506 0

# needed to remove all '+' to have the code run  
  
  
#### Table 4.4: Correlation Table for Boston Housing Data ####  
  
round(cor(boston.housing.df),2)

## CRIM ZN INDUS CHAS NOX RM AGE DIS RAD TAX PTRATIO  
## CRIM 1.00 -0.20 0.41 -0.06 0.42 -0.22 0.35 -0.38 0.63 0.58 0.29  
## ZN -0.20 1.00 -0.53 -0.04 -0.52 0.31 -0.57 0.66 -0.31 -0.31 -0.39  
## INDUS 0.41 -0.53 1.00 0.06 0.76 -0.39 0.64 -0.71 0.60 0.72 0.38  
## CHAS -0.06 -0.04 0.06 1.00 0.09 0.09 0.09 -0.10 -0.01 -0.04 -0.12  
## NOX 0.42 -0.52 0.76 0.09 1.00 -0.30 0.73 -0.77 0.61 0.67 0.19  
## RM -0.22 0.31 -0.39 0.09 -0.30 1.00 -0.24 0.21 -0.21 -0.29 -0.36  
## AGE 0.35 -0.57 0.64 0.09 0.73 -0.24 1.00 -0.75 0.46 0.51 0.26  
## DIS -0.38 0.66 -0.71 -0.10 -0.77 0.21 -0.75 1.00 -0.49 -0.53 -0.23  
## RAD 0.63 -0.31 0.60 -0.01 0.61 -0.21 0.46 -0.49 1.00 0.91 0.46  
## TAX 0.58 -0.31 0.72 -0.04 0.67 -0.29 0.51 -0.53 0.91 1.00 0.46  
## PTRATIO 0.29 -0.39 0.38 -0.12 0.19 -0.36 0.26 -0.23 0.46 0.46 1.00  
## LSTAT 0.46 -0.41 0.60 -0.05 0.59 -0.61 0.60 -0.50 0.49 0.54 0.37  
## MEDV -0.39 0.36 -0.48 0.18 -0.43 0.70 -0.38 0.25 -0.38 -0.47 -0.51  
## CAT..MEDV -0.15 0.37 -0.37 0.11 -0.23 0.64 -0.19 0.12 -0.20 -0.27 -0.44  
## LSTAT MEDV CAT..MEDV  
## CRIM 0.46 -0.39 -0.15  
## ZN -0.41 0.36 0.37  
## INDUS 0.60 -0.48 -0.37  
## CHAS -0.05 0.18 0.11  
## NOX 0.59 -0.43 -0.23  
## RM -0.61 0.70 0.64  
## AGE 0.60 -0.38 -0.19  
## DIS -0.50 0.25 0.12  
## RAD 0.49 -0.38 -0.20  
## TAX 0.54 -0.47 -0.27  
## PTRATIO 0.37 -0.51 -0.44  
## LSTAT 1.00 -0.74 -0.47  
## MEDV -0.74 1.00 0.79  
## CAT..MEDV -0.47 0.79 1.00

#### Table 4.5: Number of Neighborhoods that Bound the Charles River vs. Not ####  
  
boston.housing.df <- read.csv("BostonHousing.csv")  
table(boston.housing.df$CHAS)

##   
## 0 1   
## 471 35

#### Table 4.6: Average MEDV by CHAS and RM ####  
  
# create bins of size 1  
boston.housing.df$RM.bin <- .bincode(boston.housing.df$RM, c(1:9))  
  
# compute the average of MEDV by (binned) RM and CHAS  
# in aggregate() use the argument by= to define the list of aggregating variables  
# and FUN= as an aggregating function.  
aggregate(boston.housing.df$MEDV,  
 by=list(RM=boston.housing.df$RM.bin,  
 CHAS=boston.housing.df$CHAS),  
 FUN=mean)

## RM CHAS x  
## 1 3 0 25.30000  
## 2 4 0 15.40714  
## 3 5 0 17.20000  
## 4 6 0 21.76917  
## 5 7 0 35.96444  
## 6 8 0 45.70000  
## 7 5 1 22.21818  
## 8 6 1 25.91875  
## 9 7 1 44.06667  
## 10 8 1 35.95000

#### Table 4.7: Pivot Tables in R ####  
  
# use install.packages("reshape") the first time the package is used  
library(reshape)  
boston.housing.df <- read.csv("BostonHousing.csv")  
# create bins of size 1  
boston.housing.df$RM.bin <- .bincode(boston.housing.df$RM, c(1:9))  
  
# use melt() to stack a set of columns into a single column of data.  
# stack MEDV values for each combination of (binned) RM and CHAS  
mlt <- melt(boston.housing.df, id=c("RM.bin", "CHAS"),  
 measure=c("MEDV"))  
head(mlt, 5)

## RM.bin CHAS variable value  
## 1 6 0 MEDV 24.0  
## 2 6 0 MEDV 21.6  
## 3 7 0 MEDV 34.7  
## 4 6 0 MEDV 33.4  
## 5 7 0 MEDV 36.2

# use cast() to reshape data and generate pivot table  
cast(mlt, RM.bin ~ CHAS, subset=variable=="MEDV",  
 margins=c("grand\_row", "grand\_col"), mean)

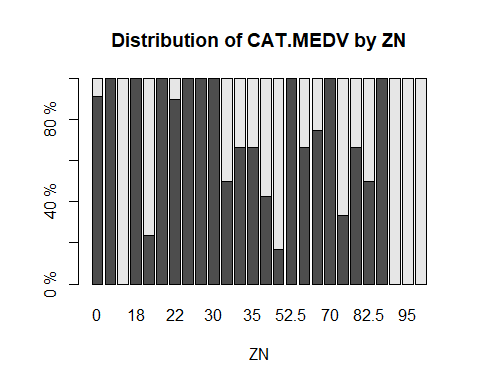
## RM.bin 0 1 (all)  
## 1 3 25.30000 NaN 25.30000  
## 2 4 15.40714 NaN 15.40714  
## 3 5 17.20000 22.21818 17.55159  
## 4 6 21.76917 25.91875 22.01599  
## 5 7 35.96444 44.06667 36.91765  
## 6 8 45.70000 35.95000 44.20000  
## 7 (all) 22.09384 28.44000 22.53281

#### Figure 4.1: Distribution of CAT.MEDV ####  
  
library(ggmap)

## Loading required package: ggplot2

## ℹ Google's Terms of Service: <https://mapsplatform.google.com>  
## Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>  
## OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>  
## ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.

boston.housing.df <- read.csv("BostonHousing.csv")  
  
tbl <- table(boston.housing.df$CAT..MEDV, boston.housing.df$ZN)  
prop.tbl <- prop.table(tbl, margin=2)  
barplot(prop.tbl, xlab="ZN", ylab="", yaxt="n",   
 main="Distribution of CAT.MEDV by ZN")  
axis(2, at=(seq(0,1, 0.2)), paste(seq(0,100,20), "%"))



#### Table 4.10: PCA on the Two Variables Calories and Rating ####  
  
cereals.df <- read.csv("Cereals.csv")  
# compute PCs on two dimensions  
pcs <- prcomp(data.frame(cereals.df$calories, cereals.df$rating))  
summary(pcs)

## Importance of components:  
## PC1 PC2  
## Standard deviation 22.3165 8.8844  
## Proportion of Variance 0.8632 0.1368  
## Cumulative Proportion 0.8632 1.0000

pcs$rot

## PC1 PC2  
## cereals.df.calories 0.8470535 0.5315077  
## cereals.df.rating -0.5315077 0.8470535

scores <- pcs$x  
head(scores, 5)

## PC1 PC2  
## [1,] -44.921528 2.1971833  
## [2,] 15.725265 -0.3824165  
## [3,] -40.149935 -5.4072123  
## [4,] -75.310772 12.9991256  
## [5,] 7.041508 -5.3576857

#### Table 4.11: PCA Output Using All 13 Numerical Variables in Breakfast Cereals Dataset ####  
  
pcs <- prcomp(na.omit(cereals.df[,-c(1:3)]))  
summary(pcs)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6  
## Standard deviation 83.7641 70.9143 22.64375 19.18148 8.42323 2.09167  
## Proportion of Variance 0.5395 0.3867 0.03943 0.02829 0.00546 0.00034  
## Cumulative Proportion 0.5395 0.9262 0.96560 0.99389 0.99935 0.99968  
## PC7 PC8 PC9 PC10 PC11 PC12 PC13  
## Standard deviation 1.69942 0.77963 0.65783 0.37043 0.1864 0.06302 5.334e-08  
## Proportion of Variance 0.00022 0.00005 0.00003 0.00001 0.0000 0.00000 0.000e+00  
## Cumulative Proportion 0.99991 0.99995 0.99999 1.00000 1.0000 1.00000 1.000e+00

#### Table 4.12: PCA Output Using All Normalized 13 Numerical Variables in the Breakfast Cereals Dataset ####  
  
pcs.cor <- prcomp(na.omit(cereals.df[,-c(1:3)]), scale. = T)  
summary(pcs.cor)

## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 1.9062 1.7743 1.3818 1.00969 0.9947 0.84974 0.81946  
## Proportion of Variance 0.2795 0.2422 0.1469 0.07842 0.0761 0.05554 0.05166  
## Cumulative Proportion 0.2795 0.5217 0.6685 0.74696 0.8231 0.87861 0.93026  
## PC8 PC9 PC10 PC11 PC12 PC13  
## Standard deviation 0.64515 0.56192 0.30301 0.25194 0.13897 1.499e-08  
## Proportion of Variance 0.03202 0.02429 0.00706 0.00488 0.00149 0.000e+00  
## Cumulative Proportion 0.96228 0.98657 0.99363 0.99851 1.00000 1.000e+00