eda la1

aniruddh singh ,animesh anand

14 May 2023

```r
options(max.print = 10)
```

installing some required packages

```r
library(datasets)
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
library(gcookbook)

library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

loading the libraries

```r
search()
```

```
##  [1] ".GlobalEnv"        "package:MASS"      "package:gcookbook"
##  [4] "package:dplyr"     "package:stats"     "package:graphics"
##  [7] "package:grDevices" "package:utils"     "package:datasets"
## [10] "package:methods"
##  [ reached getOption("max.print") -- omitted 2 entries ]
```

searching all the current available packages

```r
Boston
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

getting idea of the dataset given

```r
head(Boston)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 6 rows ]
```

showing first few columns of the dataset

```r
str(Boston)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

the structure of our dataset

```r
housing <- Boston
```

assing housing the boston dataset

```r
numberOfNA <- length(which(is.na(housing)==T))
if(numberOfNA>0) {
  housing <- housing[complete.cases(housing),]
}
```

Checking for NA and missing values and removing them

```r
dim(housing)
```

```
## [1] 506  14
```

checking the dimensions of our dataframe

```r
nrow(housing)
```

```
## [1] 506
```

number of rows

```r
ncol(housing)
```

```
## [1] 14
```

no of columns

```r
names(housing)
```

```
##  [1] "crim"  "zn"    "indus" "chas"  "nox"   "rm"    "age"   "dis"   "rad"
## [10] "tax"
##  [ reached getOption("max.print") -- omitted 4 entries ]
```

show names of columns

```r
colnames(housing)
```

```
##  [1] "crim"  "zn"    "indus" "chas"  "nox"   "rm"    "age"   "dis"   "rad"
## [10] "tax"
##  [ reached getOption("max.print") -- omitted 4 entries ]
```

again colnames

```r
tail(housing)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 6 rows ]
```

last few rows of dataset

```r
subset(housing)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

extract a subset based on a condition

```r
filter(housing, age > 50)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 359 rows ]
```

filter values based on a conditon

```r
arrange(housing, crim)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

sorts by the given column

```r
mutate(housing,crim2 = crim * 0.5)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv crim2
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

adds a new column crim 2 with values half of the original column

```r
group_by(housing,chas)
```

```
## # A tibble: 506 × 14
## # Groups:   chas [2]
##       crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio black
##      <dbl> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <int> <dbl>   <dbl> <dbl>
##  1 0.00632  18    2.31     0 0.538  6.58  65.2  4.09     1   296    15.3  397.
##  2 0.0273    0    7.07     0 0.469  6.42  78.9  4.97     2   242    17.8  397.
##  3 0.0273    0    7.07     0 0.469  7.18  61.1  4.97     2   242    17.8  393.
##  4 0.0324    0    2.18     0 0.458  7.00  45.8  6.06     3   222    18.7  395.
##  5 0.0690    0    2.18     0 0.458  7.15  54.2  6.06     3   222    18.7  397.
##  6 0.0298    0    2.18     0 0.458  6.43  58.7  6.06     3   222    18.7  394.
##  7 0.0883   12.5  7.87     0 0.524  6.01  66.6  5.56     5   311    15.2  396.
##  8 0.145    12.5  7.87     0 0.524  6.17  96.1  5.95     5   311    15.2  397.
##  9 0.211    12.5  7.87     0 0.524  5.63 100    6.08     5   311    15.2  387.
## 10 0.170    12.5  7.87     0 0.524  6.00  85.9  6.59     5   311    15.2  387.
## # ℹ 496 more rows
## # ℹ 2 more variables: lstat <dbl>, medv <dbl>
```

group by column by one column

```r
group_by(housing,chas & rad)
```

```
## # A tibble: 506 × 15
## # Groups:   chas & rad [2]
##       crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio black
##      <dbl> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <dbl> <int> <dbl>   <dbl> <dbl>
##  1 0.00632  18    2.31     0 0.538  6.58  65.2  4.09     1   296    15.3  397.
##  2 0.0273    0    7.07     0 0.469  6.42  78.9  4.97     2   242    17.8  397.
##  3 0.0273    0    7.07     0 0.469  7.18  61.1  4.97     2   242    17.8  393.
##  4 0.0324    0    2.18     0 0.458  7.00  45.8  6.06     3   222    18.7  395.
##  5 0.0690    0    2.18     0 0.458  7.15  54.2  6.06     3   222    18.7  397.
##  6 0.0298    0    2.18     0 0.458  6.43  58.7  6.06     3   222    18.7  394.
##  7 0.0883   12.5  7.87     0 0.524  6.01  66.6  5.56     5   311    15.2  396.
##  8 0.145    12.5  7.87     0 0.524  6.17  96.1  5.95     5   311    15.2  397.
##  9 0.211    12.5  7.87     0 0.524  5.63 100    6.08     5   311    15.2  387.
## 10 0.170    12.5  7.87     0 0.524  6.00  85.9  6.59     5   311    15.2  387.
## # ℹ 496 more rows
## # ℹ 3 more variables: lstat <dbl>, medv <dbl>, `chas & rad` <lgl>
```

group by chas and rad multiple columns

```r
summarize(housing)
```

```
## data frame with 0 columns and 1 row
```

summarize the dataframe

```r
na.omit(housing)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

remove rows with missing values

```r
mean(housing$crim)
```

```
## [1] 3.613524
```

show the mean of crim column

```r
median(housing$age)
```

```
## [1] 77.5
```

calculate median age

```r
sd(housing$zn)
```

```
## [1] 23.32245
```

calculate the standard deviation

```r
var(housing$crim)
```

```
## [1] 73.98658
```

calculate the variance of the crim column

```r
max(housing$age)
```

```
## [1] 100
```

calculate the max age

```r
housing
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

colnam

```r
colnames(housing)
```

```
##  [1] "crim"  "zn"    "indus" "chas"  "nox"   "rm"    "age"   "dis"   "rad"
## [10] "tax"
##  [ reached getOption("max.print") -- omitted 4 entries ]
```

```r
dataset<- housing
```

using assignment operator

```r
unique(dataset$chas)
```

```
## [1] 0 1
```

Get the unique values in the “chas” variable.

```r
table(dataset$chas)
```

```
##
##   0   1
## 471  35
```

Show the frequency table of the “chas” variable.

```r
mean(dataset$crim)
```

```
## [1] 3.613524
```

Calculate the mean of the “crim” variable.

```r
median(dataset$zn)
```

```
## [1] 0
```

Calculate the median of the “zn” variable.

```r
sd(dataset$indus)
```

```
## [1] 6.860353
```

Calculate the standard deviation of the “indus” variable.

```r
min(dataset$dis)
```

```
## [1] 1.1296
```

Calculate the minimum value in the “dis” variable.

```r
max(dataset$rad)
```

```
## [1] 24
```

Calculate the maximum value in the “rad” variable.

```r
sum(dataset$tax)
```

```
## [1] 206568
```

Calculate the sum of the “tax” variable.

```r
quantile(dataset$ptratio, probs = c(0.25, 0.50, 0.75))
```

```
##   25%   50%   75%
## 17.40 19.05 20.20
```

Calculate the quantiles (0.25, 0.50, 0.75) of the “ptratio” variable.

```r
subset(dataset, black > 300)
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 452 rows ]
```

Subset the dataset where “black” is greater than 300.

```r
dataset[order(-dataset$lstat), ]
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

Sort the dataset in descending order of “lstat” variable.

```r
dataset$tax_per_room <- dataset$tax / dataset$rm
```

Create a new variable “tax_per_room” by dividing “tax” by “rm”.

```r
dataset$log_medv <- log(dataset$medv)
```

Add a new column “log_medv” that contains the logarithm of “medv” variable.

```r
dataset <- subset(dataset, select = -age)
```

Remove the “age” variable from the dataset.

```r
aggregate(dataset$lstat, by = list(dataset$rad), FUN = mean)
```

```
##   Group.1         x
## 1       1  7.370000
## 2       2 10.024583
## 3       3  9.076053
## 4       4 12.199091
## 5       5 10.657217
##  [ reached 'max' / getOption("max.print") -- omitted 4 rows ]
```

Calculate the average “lstat” for each unique value of “rad”.

```r
subset(dataset, crim > 5 & rm > 6)
```

```
##      crim zn indus chas nox rm dis rad tax ptratio black lstat medv
##      tax_per_room log_medv
##  [ reached 'max' / getOption("max.print") -- omitted 62 rows ]
```

Create a subset of the dataset based on multiple conditions.

```r
dataset$row_sum <- rowSums(dataset[, c("crim", "indus")])
```

Calculate the row-wise sum of variables “crim” and “indus”.

```r
dataset$lstat_normalized <- (dataset$lstat - min(dataset$lstat)) / (max(dataset$lstat) - min(dataset$lstat))
```

Normalize the “lstat” variable using min-max scaling.

```r
aggregate(dataset$medv, by = list(dataset$rad, dataset$chas), FUN = mean)
```

```
##   Group.1 Group.2        x
## 1       1       0 23.01579
## 2       2       0 26.83333
## 3       3       0 27.92778
##  [ reached 'max' / getOption("max.print") -- omitted 12 rows ]
```

Compute the mean of “medv” for each unique combination of “rad” and “chas”.

```r
cor(dataset)
```

```
##                         crim          zn       indus         chas          nox
##                           rm         dis          rad          tax    ptratio
##                        black      lstat       medv tax_per_room   log_medv
##                      row_sum lstat_normalized
##  [ reached getOption("max.print") -- omitted 17 rows ]
```

Calculate the pairwise correlation matrix for all variables.

```r
filtered_dataset <- dataset[dataset$crim > 10 | dataset$zn < 5, ]
```

Filter rows where “crim” is greater than 10 or “zn” is less than 5.

```r
sapply(dataset$crim, function(x) x^2)
```

```
##  [1] 0.0000399424 0.0007458361 0.0007447441 0.0010478169 0.0047679025
##  [6] 0.0008910225 0.0077951241 0.0208947025 0.0446223376 0.0289136016
##  [ reached getOption("max.print") -- omitted 496 entries ]
```

Apply a function to each element of a variable using sapply()

```r
dataset$chas <- as.factor(dataset$chas)
```

Convert the “chas” variable into a factor.

```r
dataset$medv <- as.character(dataset$medv)
```

Convert the “medv” variable to a character type.

```r
nrow(dataset)
```

```
## [1] 506
```

Calculate the number of observations in the dataset.

```r
dataset <- dataset[order(housing$age), ]
```

Sort the dataset based on the “age” variable in ascending order.

```r
subset(dataset, rad %in% c(1, 3, 5))
```

```
##      crim zn indus chas nox rm dis rad tax ptratio black lstat medv
##      tax_per_room log_medv row_sum lstat_normalized
##  [ reached 'max' / getOption("max.print") -- omitted 173 rows ]
```

Create a new dataset containing only rows with “rad” values 1, 3, and 5.

```r
names(dataset)[names(dataset) == "ptratio"] <- "pupil_teacher_ratio"
```

Rename the variable “ptratio” to “pupil_teacher_ratio”.

```r
aggregate(dataset$lstat, by = list(dataset$rad), FUN = median)
```

```
##   Group.1      x
## 1       1  7.200
## 2       2  7.815
## 3       3  7.645
## 4       4 10.920
## 5       5  9.680
##  [ reached 'max' / getOption("max.print") -- omitted 4 rows ]
```

Calculate the median of “lstat” for each value of “rad” using aggregate().

```r
new_dataset <- subset(dataset, select = -c(black, tax))
```

Create a new dataset excluding the variables “black” and “tax”.

```r
dataset$high_crim <- ifelse(dataset$crim > 10, 1, 0)
```

Create a binary variable “high_crim” indicating if “crim” is above 10.

```r
aggregate(dataset$rm, by = list(dataset$rad, dataset$chas), FUN = mean)
```

```
##   Group.1 Group.2        x
## 1       1       0 6.483895
## 2       2       0 6.649958
## 3       3       0 6.454556
##  [ reached 'max' / getOption("max.print") -- omitted 12 rows ]
```

Calculate the average “rm” for each unique combination of “rad” and “chas” using aggregate().

```r
dataset$medv <- cut(housing$medv, breaks = c(0, 20, 30, Inf), labels = c("low", "medium", "high"))
```

Convert the “medv” variable to a factor with three levels: low, medium, high.

```r
cor_vars <- dataset[, c("crim", "nox", "lstat")]
cor(cor_vars)
```

```
##            crim       nox     lstat
## crim  1.0000000 0.4209717 0.4556215
## nox   0.4209717 1.0000000 0.5908789
## lstat 0.4556215 0.5908789 1.0000000
```

Compute the correlation matrix for variables “crim,” “nox,” and “lstat.”

```r
filtered_dataset <- subset(dataset, indus < 10 & tax > 500)
```

Filter rows where “indus” is less than 10 and “tax” is greater than 500.

```r
dataset$dis[dataset$dis < 0] <- 0
```

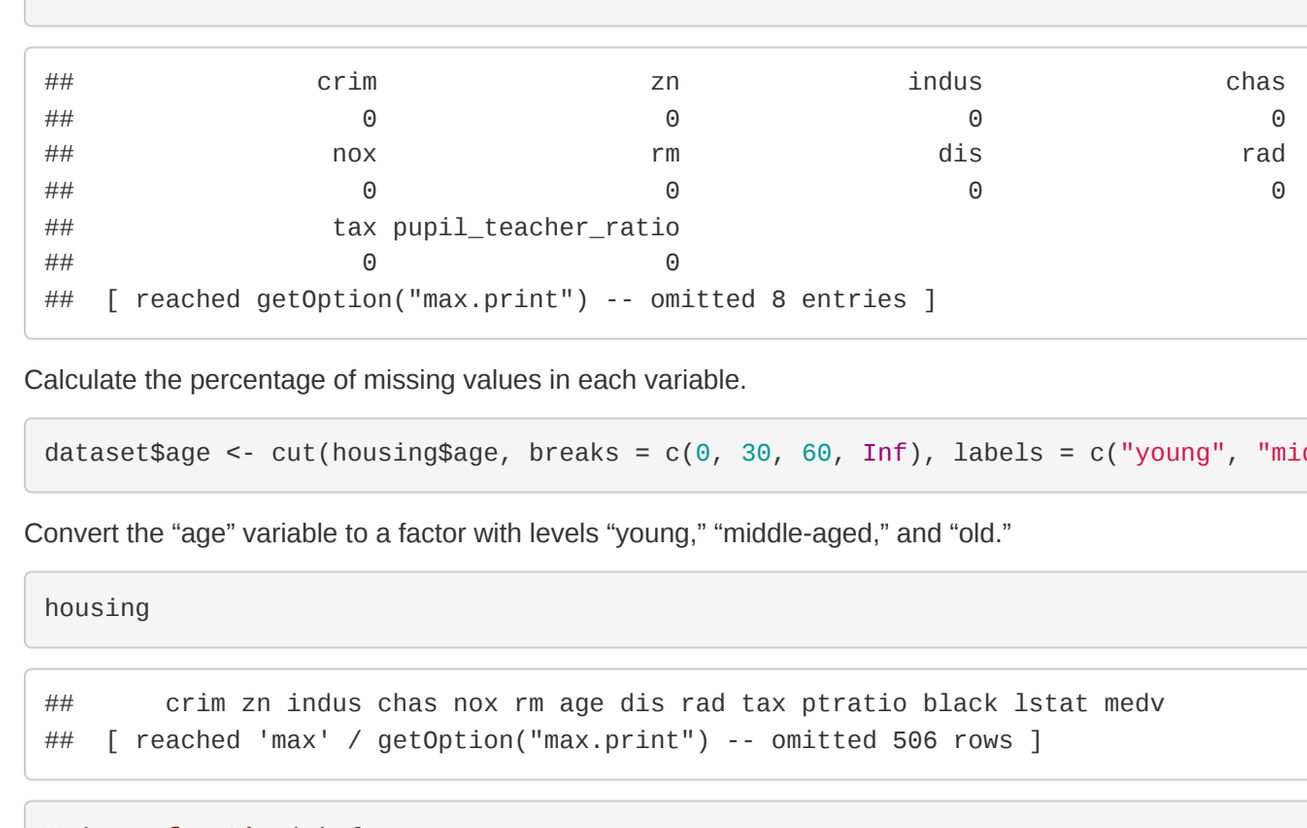
Replace negative values in the “dis” variable with 0.

```r
cumsum(dataset$lstat)
```

```
##  [1]  4.84 11.62 19.16 26.60 32.41 39.13 42.66 48.18 53.37 56.91
##  [ reached getOption("max.print") -- omitted 496 entries ]
```

Calculate the cumulative sum of “lstat” variable.

```r
plot(dataset$rm, dataset$medv, main = "Scatter Plot", xlab = "RM", ylab = "MEDV")
```



Create a scatter plot between “rm” and “medv” variables.

```r
colSums(is.na(dataset)) / nrow(dataset) * 100
```

```
##                crim                  zn               indus                chas
##                   0                   0                   0                   0
##                 nox                  rm                 dis                 rad
##                   0                   0                   0                   0
##                 tax pupil_teacher_ratio
##                   0                   0
##  [ reached getOption("max.print") -- omitted 8 entries ]
```

Calculate the percentage of missing values in each variable.

```r
dataset$age <- cut(housing$age, breaks = c(0, 30, 60, Inf), labels = c("young", "middle-aged", "old"))
```

Convert the “age” variable to a factor with levels “young,” “middle-aged,” and “old.”

```r
housing
```

```
##      crim zn indus chas nox rm age dis rad tax ptratio black lstat medv
##  [ reached 'max' / getOption("max.print") -- omitted 506 rows ]
```

```r
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
mode_rad <- Mode(dataset$rad)
```

Calculate the mode of the “rad” variable.

```r
mode_rad
```

```
## [1] 24
```

print the mode

```r
set.seed(123)  # For reproducibility
train_indices <- sample(1:nrow(dataset), 0.7 * nrow(dataset))
train_dataset <- dataset[train_indices, ]
test_dataset <- dataset[-train_indices, ]
```

Split the dataset into training and testing sets in a 70:30 ratio.

```r
dataset <- na.omit(dataset)
```

Remove rows with missing values from the dataset.

```r
dataset$chas <- as.logical(dataset$chas)
```

Convert the “chas” variable to a logical type.

```r
dist_euclidean <- sqrt(sum((dataset$rm - dataset$lstat)^2))
```

Calculate the Euclidean distance between variables “rm” and “lstat”

```r
cross_tab <- table(dataset$chas, dataset$rad)
```

Compute the cross-tabulation of “chas” and “rad” variables.

```r
library(dplyr)
housing %>%
  group_by(age) %>%
  summarize(mean_medv = mean(medv))
```

```
## # A tibble: 356 × 2
##      age mean_medv
##    <dbl>     <dbl>
##  1   2.9      26.6
##  2   6        24.1
##  3   6.2      23.4
##  4   6.5      24.7
##  5   6.6      24.8
##  6   6.8      29.6
##  7   7.8      23.2
##  8   8.4      42.8
##  9   8.9      24.8
## 10   9.8      23.7
## # ℹ 346 more rows
```

Compute the mean of “medv” for each level of “age” using dplyr package.

```r
dataset_matrix <- as.matrix(dataset)
```

Convert the dataset into a matrix.

```r
housing$age_category <- cut(housing$age, breaks = c(0, 30, 60, Inf), labels = c("young", "middle-aged", "old"))
```

Create a new variable “age_category” based on the age groups “young,” “middle-aged,” and “old”.

```r
aggregate(housing$medv, by = list(housing$age_category, housing$chas), FUN = mean)
```

```
##       Group.1 Group.2        x
## 1       young       0 26.92581
## 2 middle-aged       0 25.42807
## 3         old       0 19.78983
##  [ reached 'max' / getOption("max.print") -- omitted 3 rows ]
```

Compute the mean of “medv” for each combination of “age_category” and “chas” using aggregate().

```r
skewness <- moments::skewness(dataset$lstat)
```

Calculate the skewness of the “lstat” variable.

```r
anova_result <- aov(medv ~ age_category, data = housing)
```

Perform a one-way ANOVA to compare the means of “medv” across different levels of “age_category”.

```r
dataset$indus_tax_ratio <- dataset$indus / dataset$tax
```

Calculate the ratio of “indus” to “tax” for each observation.

```r
cummax_medv <- cummax(housing$medv)
```

Compute the cumulative maximum of the “medv” variable

```r
housing$age_squared <- housing$age^2
```

Create a new variable “age_squared” that contains the squared values of “age”.

```r
dataset$rad <- as.character(dataset$rad)
```

Convert the “rad” variable to a character type.

```r
cummin_nox <- cummin(dataset$nox)
```

Compute the cumulative minimum of the “nox” variable

```r
dataset$black <- trimws(dataset$black)
```

Remove leading and trailing whitespaces from the “black” variable.

```r
library(dplyr)
dataset %>%
  group_by(rad) %>%
  summarize(sum_lstat = sum(lstat))
```

```
## # A tibble: 9 × 2
##   rad   sum_lstat
##   <chr>     <dbl>
## 1 1          147.
## 2 2          241.
## 3 24        2455.
## 4 3          345.
## 5 4         1342.
## 6 5         1226.
## 7 6          320.
## 8 7          136.
## 9 8          191.
```

Calculate the sum of “lstat” for each level of “rad” using dplyr package.