Calculating Descriptive Statistics in R

Ariel Felices

5/13/2022

Every student of statistics should know about the different branches of statistics in order to correctly understand statistics from a more holistic point of view. Often times, the kind of job or work one is involved in hides the other aspect of statistics, but it’s very important to know and have an overall idea behind statistical analysis in order to fully appreciate its importance and beauty. Yes, statistics is beautiful. The two main branches of statistics are descriptive statistics and inferential statistics. Both of these are employed in scientific analysis of data, and both are equally important for students of statistics.

What is descriptive statistics? Descriptive statistics deals with the presentation and collection of data. This is usually the first part of a statistical analysis. It is usually not as simple as it sounds. And the statistician needs to be aware of designing experiment, choosing the right focus group, and avoiding biases that are so easy to crip in experiment. So, that is descriptive statistics– it talks about description, presentation and collection of data.

What about inferential statistics? Inferential statistics as the name suggests, involves drawing the right conclusions from statistical analysis that has been performed using descriptive statistics. In the end, it is the inferences that make studies important and this aspect is dealt with in inferential statistics.

## Task One: Getting Started

In this task, we will learn to set and check our current working directory.

### 1.1. Set and get the working directory.

setwd(dir = "C:/Users/user/Documents/R\_PROJECTS/Calculating Descriptive Statistics in R/Files")

### 1.2. Get the working directory

getwd()

## [1] "C:/Users/user/Documents/R\_PROJECTS/Calculating Descriptive Statistics in R/Files"

## Task Two: Import data set and explore

In this task, we will import two data sets and explore them.

### 2.1 Import the msleep.csv and mpg.csv data sets

df <- read.csv(file = "msleep.csv")

df1 <- read.csv("mpg.csv")

### 2.2: View and check the dimension of the data sets

# View(df)

dim(df)

## [1] 83 12

dim(df1)

## [1] 234 12

### 2.3: Take a peek at df using the head and tail functions

head(df)

## X.1 X name genus vore order conservation  
## 1 1 1 Cheetah Acinonyx 1 Carnivora 4  
## 2 2 2 Owl monkey Aotus 4 Primates 4  
## 3 3 3 Mountain beaver Aplodontia 2 Rodentia 5  
## 4 4 4 Greater short-tailed shrew Blarina 4 Soricomorpha 4  
## 5 5 5 Cow Bos 2 Artiodactyla 2  
## 6 6 6 Three-toed sloth Bradypus 2 Pilosa 4  
## Total.Sleep.Time sleep\_rem awake brainwt bodywt  
## 1 12.1 1.87541 11.9 0.2815814 50.000  
## 2 17.0 1.80000 7.0 0.0155000 0.480  
## 3 14.4 2.40000 9.6 0.2815814 1.350  
## 4 14.9 2.30000 9.1 0.0002900 0.019  
## 5 4.0 0.70000 20.0 0.4230000 600.000  
## 6 14.4 2.20000 9.6 0.2815814 3.850

tail(df)

## X.1 X name genus vore order conservation  
## 78 78 78 Tenrec Tenrec 4 Afrosoricida 4  
## 79 79 79 Tree shrew Tupaia 4 Scandentia 4  
## 80 80 80 Bottle-nosed dolphin Tursiops 1 Cetacea 4  
## 81 81 81 Genet Genetta 1 Carnivora 4  
## 82 82 82 Arctic fox Vulpes 1 Carnivora 4  
## 83 83 83 Red fox Vulpes 1 Carnivora 4  
## Total.Sleep.Time sleep\_rem awake brainwt bodywt  
## 78 15.6 2.30000 8.4 0.0026000 0.900  
## 79 8.9 2.60000 15.1 0.0025000 0.104  
## 80 5.2 1.87541 18.8 0.2815814 173.330  
## 81 6.3 1.30000 17.7 0.0175000 2.000  
## 82 12.5 1.87541 11.5 0.0445000 3.380  
## 83 9.8 2.40000 14.2 0.0504000 4.230

### 2.4: Check the internal structure of the data frame

str(df)

## 'data.frame': 83 obs. of 12 variables:  
## $ X.1 : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ X : int 1 2 3 4 5 6 7 8 9 10 ...  
## $ name : chr "Cheetah" "Owl monkey" "Mountain beaver" "Greater short-tailed shrew" ...  
## $ genus : chr "Acinonyx" "Aotus" "Aplodontia" "Blarina" ...  
## $ vore : int 1 4 2 4 2 2 1 2 1 2 ...  
## $ order : chr "Carnivora" "Primates" "Rodentia" "Soricomorpha" ...  
## $ conservation : int 4 4 5 4 2 4 6 4 2 4 ...  
## $ Total.Sleep.Time: num 12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...  
## $ sleep\_rem : num 1.88 1.8 2.4 2.3 0.7 ...  
## $ awake : num 11.9 7 9.6 9.1 20 9.6 15.3 17 13.9 21 ...  
## $ brainwt : num 0.28158 0.0155 0.28158 0.00029 0.423 ...  
## $ bodywt : num 50 0.48 1.35 0.019 600 ...

### 2.5: Count missing values in the variables

sum(is.na(df))

## [1] 0

To check the missing value per column:

sapply(df, function(x) sum(is.na(x)))

## X.1 X name genus   
## 0 0 0 0   
## vore order conservation Total.Sleep.Time   
## 0 0 0 0   
## sleep\_rem awake brainwt bodywt   
## 0 0 0 0

### 2.6: Check the column names for the df data frame

names(df)

## [1] "X.1" "X" "name" "genus"   
## [5] "vore" "order" "conservation" "Total.Sleep.Time"  
## [9] "sleep\_rem" "awake" "brainwt" "bodywt"

colnames(df)

## [1] "X.1" "X" "name" "genus"   
## [5] "vore" "order" "conservation" "Total.Sleep.Time"  
## [9] "sleep\_rem" "awake" "brainwt" "bodywt"

## Task Three: Frequency of categorical variables

In this task, we will learn how to count the frequency and percentage of a categorical variable.

Let us view the df data frame

# View(df)

dim(df)

## [1] 83 12

### 3.1: Drop the first two columns of df

df <- df[, -c(1, 2)]

dim(df)

## [1] 83 10

names(df)

## [1] "name" "genus" "vore" "order"   
## [5] "conservation" "Total.Sleep.Time" "sleep\_rem" "awake"   
## [9] "brainwt" "bodywt"

### 3.2: Get the frequency of a categorical variable

str(df)

## 'data.frame': 83 obs. of 10 variables:  
## $ name : chr "Cheetah" "Owl monkey" "Mountain beaver" "Greater short-tailed shrew" ...  
## $ genus : chr "Acinonyx" "Aotus" "Aplodontia" "Blarina" ...  
## $ vore : int 1 4 2 4 2 2 1 2 1 2 ...  
## $ order : chr "Carnivora" "Primates" "Rodentia" "Soricomorpha" ...  
## $ conservation : int 4 4 5 4 2 4 6 4 2 4 ...  
## $ Total.Sleep.Time: num 12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...  
## $ sleep\_rem : num 1.88 1.8 2.4 2.3 0.7 ...  
## $ awake : num 11.9 7 9.6 9.1 20 9.6 15.3 17 13.9 21 ...  
## $ brainwt : num 0.28158 0.0155 0.28158 0.00029 0.423 ...  
## $ bodywt : num 50 0.48 1.35 0.019 600 ...

Get the frequency of the feature “order”

table(df$order)

##   
## Afrosoricida Artiodactyla Carnivora Cetacea Chiroptera   
## 1 6 12 3 2   
## Cingulata Didelphimorphia Diprotodontia Erinaceomorpha Hyracoidea   
## 2 2 2 2 3   
## Lagomorpha Monotremata Perissodactyla Pilosa Primates   
## 1 1 3 1 12   
## Proboscidea Rodentia Scandentia Soricomorpha   
## 2 22 1 5

### 3.3: Calculate the percentage from the frequency

total <- length(df$order)

percent <- (table(df$order)/total) \* 100  
percent

##   
## Afrosoricida Artiodactyla Carnivora Cetacea Chiroptera   
## 1.204819 7.228916 14.457831 3.614458 2.409639   
## Cingulata Didelphimorphia Diprotodontia Erinaceomorpha Hyracoidea   
## 2.409639 2.409639 2.409639 2.409639 3.614458   
## Lagomorpha Monotremata Perissodactyla Pilosa Primates   
## 1.204819 1.204819 3.614458 1.204819 14.457831   
## Proboscidea Rodentia Scandentia Soricomorpha   
## 2.409639 26.506024 1.204819 6.024096

Round to 2 decimal places

percent\_r <- round(percent, 2)  
percent\_r

##   
## Afrosoricida Artiodactyla Carnivora Cetacea Chiroptera   
## 1.20 7.23 14.46 3.61 2.41   
## Cingulata Didelphimorphia Diprotodontia Erinaceomorpha Hyracoidea   
## 2.41 2.41 2.41 2.41 3.61   
## Lagomorpha Monotremata Perissodactyla Pilosa Primates   
## 1.20 1.20 3.61 1.20 14.46   
## Proboscidea Rodentia Scandentia Soricomorpha   
## 2.41 26.51 1.20 6.02

## Task Four: Univariate statistics for univariate variables - Part I

In this task, we will perform different descriptive statistics for a quantitative variable

### Univariate statistics for univariate variables

### 4.1: Calculate the mean of total sleep time variable

mean(df$Total.Sleep.Time)

## [1] 10.43373

### 4.2: Calculate the median of total sleep time variable

median(df$Total.Sleep.Time)

## [1] 10.1

### 4.3: Assign a single variable to a new variable

sleeptime <- df$Total.Sleep.Time

### 4.4: Calculate the mode of total sleep time

which.max(sleeptime)

## [1] 43

which.max(table(sleeptime))

## 12.5   
## 43

### The spread of a quantitative variable

### 4.5: Get the minimum value for total sleep time

min(sleeptime)

## [1] 1.9

### 4.6: Get the maximum value for total sleep time

max(sleeptime)

## [1] 19.9

### 4.7: Get the range for total sleep time

range(sleeptime)

## [1] 1.9 19.9

### 4.8: Get the range difference for total sleep time

diff(range(sleeptime))

## [1] 18

### 4.9: Sort the total sleep time variable

sort(sleeptime)

## [1] 1.9 2.7 2.9 3.0 3.1 3.3 3.5 3.8 3.9 4.0 4.4 5.2 5.3 5.3 5.4  
## [16] 5.6 6.2 6.3 6.3 7.0 7.7 8.0 8.3 8.4 8.4 8.6 8.7 8.7 8.9 9.1  
## [31] 9.1 9.4 9.4 9.5 9.6 9.7 9.8 9.8 10.0 10.1 10.1 10.1 10.3 10.3 10.4  
## [46] 10.6 10.9 11.0 11.0 11.1 11.3 11.5 12.1 12.5 12.5 12.5 12.5 12.8 12.8 13.0  
## [61] 13.5 13.7 13.8 14.2 14.3 14.4 14.4 14.5 14.6 14.9 14.9 15.6 15.8 15.8 15.9  
## [76] 16.6 17.0 17.4 18.0 18.1 19.4 19.7 19.9

In descending order

sort(sleeptime, decreasing = T)

## [1] 19.9 19.7 19.4 18.1 18.0 17.4 17.0 16.6 15.9 15.8 15.8 15.6 14.9 14.9 14.6  
## [16] 14.5 14.4 14.4 14.3 14.2 13.8 13.7 13.5 13.0 12.8 12.8 12.5 12.5 12.5 12.5  
## [31] 12.1 11.5 11.3 11.1 11.0 11.0 10.9 10.6 10.4 10.3 10.3 10.1 10.1 10.1 10.0  
## [46] 9.8 9.8 9.7 9.6 9.5 9.4 9.4 9.1 9.1 8.9 8.7 8.7 8.6 8.4 8.4  
## [61] 8.3 8.0 7.7 7.0 6.3 6.3 6.2 5.6 5.4 5.3 5.3 5.2 4.4 4.0 3.9  
## [76] 3.8 3.5 3.3 3.1 3.0 2.9 2.7 1.9

## Task Five: Univariate statistics for univariate variables - Part II

In this task, we will continue to perform different descriptive statistics for a quantitative variable

### 5.1: Calculate the quartile of the total sleep time variable

Returns all the quartiles

quantile(sleeptime)

## 0% 25% 50% 75% 100%   
## 1.90 7.85 10.10 13.75 19.90

Return the 1st, 2nd and 3rd quartiles

quantile(sleeptime, c(0.25, 0.50, 0.75))

## 25% 50% 75%   
## 7.85 10.10 13.75

Return the 90th percentile

quantile(sleeptime, 0.90)

## 90%   
## 15.88

### 5.2: Calculate the inter-quartile range

IQR(sleeptime)

## [1] 5.9

### 5.3: Calculate the variance

var(sleeptime)

## [1] 19.80568

The variance measures how far a set of numbers is spread out from the average value or the mean. The variance is arguably the most commonly used measure of dispersion or measure of spread because it measures the deviation from one particular central tendency, say for example, the mean. So, like the difference between each value and the mean.

### 5.4: Calculate the standard deviation

sqrt(var(sleeptime))

## [1] 4.450357

sd(sleeptime)

## [1] 4.450357

Standard deviation is simply the square root of the variance. In statistics, majorly, the standard deviation is a measure of the amount of variation or dispersion of the set of value. So, a low standard division indicates that the value tends to be close to the mean of the set, and while, a high standard deviation indicates that the values are spread out in a wider range.

### 5.5: Get the five number summary

fivenum(sleeptime)

## [1] 1.90 7.85 10.10 13.75 19.90

The five number summary is a set of descriptive statistics that provides information about a dataset. The summary consists of five values. The most extreme values that is the minimum and maximum value. And then the lower and the upper quartiles and then the median. This five number summary was introduced by John W. Tukey about 1970.

### 5.6: Summarize the total sleep time variable

summary(sleeptime)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.90 7.85 10.10 10.43 13.75 19.90

### Task Six: Distribution of Quantitative variable

In this task, we will learn how to check the distribution of a quantitative variable

Analyze the spread of the quantitative variable

### 6.1: Install the e1071 package

# install.packages("e1071")

### 6.2: Importing a library

library(e1071)

## Warning: package 'e1071' was built under R version 4.1.3

### 6.3: Calculate the skewness of the total sleep time variable

skewness((sleeptime))

## [1] 0.05230964

The skewness is 0.05. That looks to be close to the normal distribution.

Skewness is simply a measure of symmetry. How symmetric the particular numeric variable is. More precisely, we look at the data set if it is symmetric. Skewness for a normal distribution, which is desired is zero. So, if we check the skewness of this particular sleep time, and it gives us zero. Then, we can say that it is symmetric or it is normally distributed. However, a negative value for skewness indicates that the data is skewed to the left. And positive value indicates that the data is skewed to the right. Skewed to the left means that the left tail is long relative to the right tail. And similarly skewed right means that the right tail is long relative to the left tail. If data is multi-modal, that is it has two most reocurring value, then this may affect the sign of the skewness.

### 6.4: Calculate the kurtosis of the total sleep time variable

kurtosis(sleeptime)

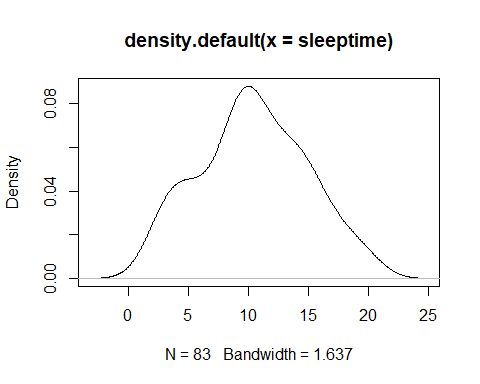
## [1] -0.7074466

Now, the kurtosis is negative. That means that there are no much outliers. It makes sense again that there are no much outliers. Another way we can do this, is to plot a density plot to see how the distribution is spread out.

Kurtosis is a measure of whether data are highly or heavily tailed, or light tailed relative to the normal distribution. This is to check the spread. How distributed– is it normally distributed. How heavy are the tails? Is it skewed? That is datasets with high kurtosis, tends to have heavy tails, that is outliers. Then data with low kurtosis tend to have light tails and lack of outliers. One particular graphical method we can use is histogram. The histogram is an effective graphical technique for showing both the skewness and kurtosis of a data set. We will consider using histogram in this particular task.

### 6.5: Plot the density of the total sleep time variable

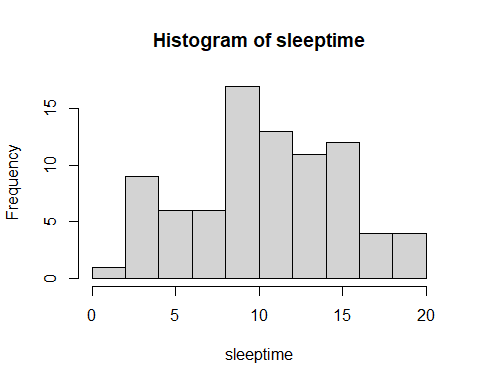
plot(density(sleeptime))



It seems to be normally distributed and the tails are light. No wonder that the kurtosis gave us a negative value. Next is, we want to plot an histogram, and let us see how it looks.

### 6.6: Plot an histogram of the total sleep time variable

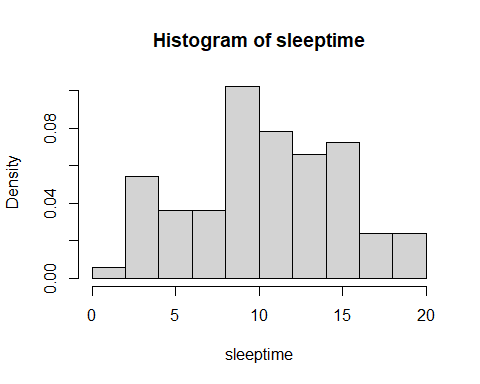
hist(sleeptime)



A histogram is simply a graphical display of data using bins of different height. In a histogram, each bar groups numbers into ranges. So, a histogram displays the shape and the spread of continuous sample data. It was first introduced by Karl Pearson, who was an English mathematician and Biostatistician. He has been credited with establishing the discipline of mathematical statistics.

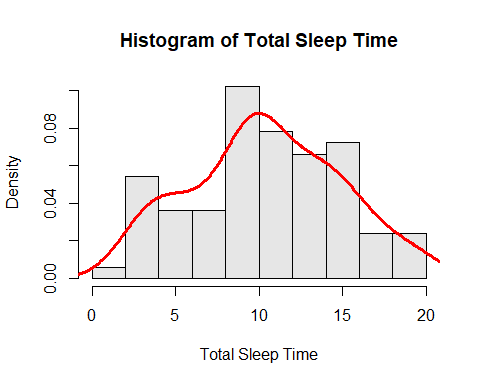
### 6.7: Plot the histogram using proportions or probabilities

hist(sleeptime, probability = TRUE)



Give the histogram a title. Add a density line on the histogram. Superimpose a normal curve.

hist(sleeptime, probability = T,  
 col = grey(0.9),  
 main = "Histogram of Total Sleep Time",  
 xlab = "Total Sleep Time")  
lines(density(sleeptime), col = "red", lw = 3)



## Task Seven: Bivariate statistics for variables

In this task, we will learn how to perform bivariate statistics for both qualitative and quantitative variables

In statistics, bivariate data is data on each of two variables where each value of one of the variables is paired with the value of another variable. Typically, it would be of interest to investigate the possible relationship or association between these two variables.

### 7.1: Bivariate statistics for two qualitative variables

View(df1) ## View the data set

table(df1$class, df1$drv)

##   
## 4 f r  
## 2seater 0 0 5  
## compact 12 35 0  
## midsize 3 38 0  
## minivan 0 11 0  
## pickup 33 0 0  
## subcompact 4 22 9  
## suv 51 0 11

So, f here means forward, uh, front wheel, r here means rear wheel, and 4 here means 4ward. So, you can see that for the 2 seater, they had five that were– that had rear wheels, alright? So, we can see how to compare two categorical variables to check their frequency.

### Bivariate statistics for two quantitative variables

What if we have two quantitative variables? What do we use to get their descriptive statistics? We use two common statistics. And that is the covariance and the correlation. The covariance is simply a measure of the directional relationship between two variables. A positive covariance means that the variables move together in the same direction. A negative covariance means that they move inversely in in different directions. For correlation in statistics, this is usually used to check relationship or association whether it is causal or not, between two random variables. In simple terms, these two measures the relationship and dependency between two variables, but covariance only measures, or indicates the direction of the linear relationship. While the correlation handles both the strength and the direction of the linear relationship. And a correlation always spans from minus one to plus one. Basically, the correlation is a function of the covariance.

### 7.2: Calculate covariances

cov(df$Total.Sleep.Time, df$sleep\_rem)

## [1] 3.237229

This gives a positive covariance. So, this means that these two particular variables, move in the same direction. So, their measure goes in the same direction. Let us see two other quantitative variables or numerical variables.

cov(df$awake, df$brainwt)

## [1] 1.04393

This gives also a positive covariance. So this makes complete sense.

### 7.3: Calculate correlation coefficients

cor(df$Total.Sleep.Time, df$sleep\_rem)

## [1] 0.6549963

And now, this says 0.65. In this case, I can say, 65%. So this tells me that there is a relationship between the total sleep and this sleep rem and that it is kind of strong. It’s in a positive direction. Now, the 65% does not just tell me the direction with they move but the strength of this relationship. Positive relationship.

cor(df$awake, df$brainwt)

## [1] 0.293224

This is 0.29. Although, there seems to be a positive correlation between awake and brainwt, however, the relationship is not so strong and this is 0.29 as compared to 0.65. I think this makes complete sense. So, covariance measures the direction of the relationship, correlation measures both direction and the strength of the relationship.

#### Bivariate statistics for both a quantitative and qualitative variable

Okay, lastly, we want to check how to check a bivariate statistics for both a quantitative and a qualitative variable. So, remember that the the sleeptime is quantitative, now we want tocheck that for the different orders. Remember, we looked at order earlier, in task two. We want to see the mean total sleep time for each of the different order of this mammal. We know a mammal has order, class and the rest of them. So, we want to see for the different orders– we want to see the mean sleep time.

### 7.4: Calculate the mean total sleep time for different orders

tapply(df$Total.Sleep.Time, df$order, mean)

## Afrosoricida Artiodactyla Carnivora Cetacea Chiroptera   
## 15.600000 4.516667 10.116667 4.500000 19.800000   
## Cingulata Didelphimorphia Diprotodontia Erinaceomorpha Hyracoidea   
## 17.750000 18.700000 12.400000 10.200000 5.666667   
## Lagomorpha Monotremata Perissodactyla Pilosa Primates   
## 8.400000 8.600000 3.466667 14.400000 10.500000   
## Proboscidea Rodentia Scandentia Soricomorpha   
## 3.600000 12.468182 8.900000 11.100000

Now we see how we can check bivariate– bivariate means two variables between a qualitative and a quantitative variable.

Okay, these names are so serious to pronounce. Say rodentia for example, the mean sleeptime is 12.46. So, what this does is, it goes through the data set and checks for each of the order, get their different sleep times, and calculates the mean. So for primates- 10.5, for carnivora 10.11, and the rest of it like that. So this really makes sense that we can check the relationship between not just two quantitative or two qualitative, but between a quantitative and a qualitative variable.

### 7.5: Calculate the summary of all variables in the df data frame

summary(df)

## name genus vore order   
## Length:83 Length:83 Min. :1.000 Length:83   
## Class :character Class :character 1st Qu.:2.000 Class :character   
## Mode :character Mode :character Median :2.000 Mode :character   
## Mean :2.313   
## 3rd Qu.:3.000   
## Max. :4.000   
## conservation Total.Sleep.Time sleep\_rem awake   
## Min. :1.000 Min. : 1.90 Min. :0.100 Min. : 4.10   
## 1st Qu.:4.000 1st Qu.: 7.85 1st Qu.:1.150 1st Qu.:10.25   
## Median :4.000 Median :10.10 Median :1.875 Median :13.90   
## Mean :3.855 Mean :10.43 Mean :1.875 Mean :13.57   
## 3rd Qu.:4.000 3rd Qu.:13.75 3rd Qu.:2.200 3rd Qu.:16.15   
## Max. :6.000 Max. :19.90 Max. :6.600 Max. :22.10   
## brainwt bodywt   
## Min. :0.00014 Min. : 0.005   
## 1st Qu.:0.00635 1st Qu.: 0.174   
## Median :0.11500 Median : 1.670   
## Mean :0.28158 Mean : 166.136   
## 3rd Qu.:0.28158 3rd Qu.: 41.750   
## Max. :5.71200 Max. :6654.000

Now, I can see that for order it counts since order is a categorical variable. For those that are numeric variables (sleep\_rem), it tells you the min, the first quartile, the median, the mean, the third quartile, the maximum. You can see for the numeric variable, it returns like the five numsum– the five number summary that we looked at, including the mean and then for the other ones, it just gives a kind of frequency of what we have.