R data frame

Armand Tossou

9/3/2021

Sample public data set (data frame), built into R

- The `state.x77` data set.
- The `USPersonalExpenditure` data set.
- The `women` data set.
- The `WorldPhones` data set.

```
state.x77
```

```
##                Population Income Illiteracy Life Exp Murder HS Grad Frost
## Alabama              3615   3624        2.1    69.05   15.1    41.3    20
## Alaska                365   6315        1.5    69.31   11.3    66.7   152
## Arizona              2212   4530        1.8    70.55    7.8    58.1    15
## Arkansas             2110   3378        1.9    70.66   10.1    39.9    65
## California          21198   5114        1.1    71.71   10.3    62.6    20
## Colorado             2541   4884        0.7    72.06    6.8    63.9   166
## Connecticut          3100   5348        1.1    72.48    3.1    56.0   139
## Delaware              579   4809        0.9    70.06    6.2    54.6   103
## Florida              8277   4815        1.3    70.66   10.7    52.6    11
## Georgia              4931   4091        2.0    68.54   13.9    40.6    60
## Hawaii                868   4963        1.9    73.60    6.2    61.9     0
## Idaho                 813   4119        0.6    71.87    5.3    59.5   126
## Illinois            11197   5107        0.9    70.14   10.3    52.6   127
## Indiana              5313   4458        0.7    70.88    7.1    52.9   122
## Iowa                 2861   4628        0.5    72.56    2.3    59.0   140
## Kansas               2280   4669        0.6    72.58    4.5    59.9   114
## Kentucky             3387   3712        1.6    70.10   10.6    38.5    95
## Louisiana            3806   3545        2.8    68.76   13.2    42.2    12
## Maine                1058   3694        0.7    70.39    2.7    54.7   161
## Maryland             4122   5299        0.9    70.22    8.5    52.3   101
## Massachusetts        5814   4755        1.1    71.83    3.3    58.5   103
## Michigan             9111   4751        0.9    70.63   11.1    52.8   125
## Minnesota            3921   4675        0.6    72.96    2.3    57.6   160
## Mississippi          2341   3098        2.4    68.09   12.5    41.0    50
## Missouri             4767   4254        0.8    70.69    9.3    48.8   108
## Montana               746   4347        0.6    70.56    5.0    59.2   155
## Nebraska             1544   4508        0.6    72.60    2.9    59.3   139
## Nevada                590   5149        0.5    69.03   11.5    65.2   188
## New Hampshire         812   4281        0.7    71.23    3.3    57.6   174
## New Jersey           7333   5237        1.1    70.93    5.2    52.5   115
## New Mexico           1144   3601        2.2    70.32    9.7    55.2   120
```

```
## New York            18076   4903        1.4    70.55   10.9    52.7    82
## North Carolina       5441   3875        1.8    69.21   11.1    38.5    80
## North Dakota          637   5087        0.8    72.78    1.4    50.3   186
## Ohio                10735   4561        0.8    70.82    7.4    53.2   124
## Oklahoma             2715   3983        1.1    71.42    6.4    51.6    82
## Oregon               2284   4660        0.6    72.13    4.2    60.0    44
## Pennsylvania        11860   4449        1.0    70.43    6.1    50.2   126
## Rhode Island          931   4558        1.3    71.90    2.4    46.4   127
## South Carolina       2816   3635        2.3    67.96   11.6    37.8    65
## South Dakota          681   4167        0.5    72.08    1.7    53.3   172
## Tennessee            4173   3821        1.7    70.11   11.0    41.8    70
## Texas               12237   4188        2.2    70.90   12.2    47.4    35
## Utah                 1203   4022        0.6    72.90    4.5    67.3   137
## Vermont               472   3907        0.6    71.64    5.5    57.1   168
## Virginia             4981   4701        1.4    70.08    9.5    47.8    85
## Washington           3559   4864        0.6    71.72    4.3    63.5    32
## West Virginia        1799   3617        1.4    69.48    6.7    41.6   100
## Wisconsin            4589   4468        0.7    72.48    3.0    54.5   149
## Wyoming               376   4566        0.6    70.29    6.9    62.9   173
##                    Area
## Alabama           50708
## Alaska           566432
## Arizona          113417
## Arkansas          51945
## California       156361
## Colorado         103766
## Connecticut        4862
## Delaware           1982
## Florida           54090
## Georgia           58073
## Hawaii             6425
## Idaho             82677
## Illinois          55748
## Indiana           36097
## Iowa              55941
## Kansas            81787
## Kentucky          39650
## Louisiana         44930
## Maine             30920
## Maryland           9891
## Massachusetts      7826
## Michigan          56817
## Minnesota         79289
## Mississippi       47296
## Missouri          68995
## Montana          145587
## Nebraska          76483
## Nevada           109889
## New Hampshire      9027
## New Jersey         7521
## New Mexico       121412
## New York          47831
## North Carolina    48798
## North Dakota      69273
```

```
## Ohio            40975
## Oklahoma        68782
## Oregon          96184
## Pennsylvania    44966
## Rhode Island     1049
## South Carolina  30225
## South Dakota    75955
## Tennessee       41328
## Texas          262134
## Utah            82096
## Vermont          9267
## Virginia        39780
## Washington      66570
## West Virginia   24070
## Wisconsin       54464
## Wyoming         97203
```

```
USPersonalExpenditure
```

```
##                       1940   1945  1950 1955  1960
## Food and Tobacco    22.200 44.500 59.60 73.2 86.80
## Household Operation 10.500 15.500 29.00 36.5 46.20
## Medical and Health   3.530  5.760  9.71 14.0 21.10
## Personal Care        1.040  1.980  2.45  3.4  5.40
## Private Education    0.341  0.974  1.80  2.6  3.64
```

```
women
```

```
##    height weight
## 1      58    115
## 2      59    117
## 3      60    120
## 4      61    123
## 5      62    126
## 6      63    129
## 7      64    132
## 8      65    135
## 9      66    139
## 10     67    142
## 11     68    146
## 12     69    150
## 13     70    154
## 14     71    159
## 15     72    164
```

```
WorldPhones
```

```
##      N.Amer Europe Asia S.Amer Oceania Africa Mid.Amer
## 1951  45939  21574 2876   1815    1646     89      555
## 1956  60423  29990 4708   2568    2366   1411      733
## 1957  64721  32510 5230   2695    2526   1546      773
## 1958  68484  35218 6662   2845    2691   1663      836
```

```
## 1959  71799  37598 6856   3000    2868   1769      911
## 1960  76036  40341 8220   3145    3054   1905     1008
## 1961  79831  43173 9053   3338    3224   2005     1076
```

```
data() # list of all data sets built in R

# preview the first 6 rows of the 'state.x77' data set
head(state.x77)
```

```
##            Population Income Illiteracy Life Exp Murder HS Grad Frost   Area
## Alabama          3615   3624        2.1    69.05   15.1    41.3    20  50708
## Alaska            365   6315        1.5    69.31   11.3    66.7   152 566432
## Arizona          2212   4530        1.8    70.55    7.8    58.1    15 113417
## Arkansas         2110   3378        1.9    70.66   10.1    39.9    65  51945
## California      21198   5114        1.1    71.71   10.3    62.6    20 156361
## Colorado         2541   4884        0.7    72.06    6.8    63.9   166 103766
```

```
# preview the last 6 rows of the 'state.x77' data set
tail(state.x77)
```

```
##               Population Income Illiteracy Life Exp Murder HS Grad Frost  Area
## Vermont              472   3907        0.6    71.64    5.5    57.1   168  9267
## Virginia            4981   4701        1.4    70.08    9.5    47.8    85 39780
## Washington          3559   4864        0.6    71.72    4.3    63.5    32 66570
## West Virginia       1799   3617        1.4    69.48    6.7    41.6   100 24070
## Wisconsin           4589   4468        0.7    72.48    3.0    54.5   149 54464
## Wyoming              376   4566        0.6    70.29    6.9    62.9   173 97203
```

```
# get the structure of the 'state.x77' data set
str(state.x77)
```

```
##  num [1:50, 1:8] 3615 365 2212 2110 21198 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:50] "Alabama" "Alaska" "Arizona" "Arkansas" ...
##   ..$ : chr [1:8] "Population" "Income" "Illiteracy" "Life Exp" ...
```

```
# summary statistics of variables in the 'state.x77' data set
summary(state.x77)
```

```
##    Population        Income       Illiteracy       Life Exp
##  Min.   :  365   Min.   :3098   Min.   :0.500   Min.   :67.96
##  1st Qu.: 1080   1st Qu.:3993   1st Qu.:0.625   1st Qu.:70.12
##  Median : 2838   Median :4519   Median :0.950   Median :70.67
##  Mean   : 4246   Mean   :4436   Mean   :1.170   Mean   :70.88
##  3rd Qu.: 4968   3rd Qu.:4814   3rd Qu.:1.575   3rd Qu.:71.89
##  Max.   :21198   Max.   :6315   Max.   :2.800   Max.   :73.60
##      Murder          HS Grad          Frost             Area
##  Min.   : 1.400   Min.   :37.80   Min.   :  0.00   Min.   :  1049
##  1st Qu.: 4.350   1st Qu.:48.05   1st Qu.: 66.25   1st Qu.: 36985
##  Median : 6.850   Median :53.25   Median :114.50   Median : 54277
##  Mean   : 7.378   Mean   :53.11   Mean   :104.46   Mean   : 70736
##  3rd Qu.:10.675   3rd Qu.:59.15   3rd Qu.:139.75   3rd Qu.: 81163
##  Max.   :15.100   Max.   :67.30   Max.   :188.00   Max.   :566432
```

Create our own data frame

We'll create weather data.

```
days <- c('Mon','Tue','Wed','Thu','Fri') # weekdays
temp <- c(22.2,21,23,24.3,25) # daily temperature
rain <- c(T,T,F,F,T) # whether or not it rained on a given day

# combine those matrices into a data frame
df <- data.frame(days,temp,rain)
df
```

```
##   days temp  rain
## 1  Mon 22.2  TRUE
## 2  Tue 21.0  TRUE
## 3  Wed 23.0 FALSE
## 4  Thu 24.3 FALSE
## 5  Fri 25.0  TRUE
```

```
# check the structure of the newly created data frame
str(df)
```

```
## 'data.frame':    5 obs. of  3 variables:
##  $ days: chr  "Mon" "Tue" "Wed" "Thu" ...
##  $ temp: num  22.2 21 23 24.3 25
##  $ rain: logi  TRUE TRUE FALSE FALSE TRUE
```

```
#get summary of each variable
summary(df)
```

```
##      days                temp         rain
##  Length:5           Min.   :21.0   Mode :logical
##  Class :character   1st Qu.:22.2   FALSE:2
##  Mode  :character   Median :23.0   TRUE :3
##                     Mean   :23.1
##                     3rd Qu.:24.3
##                     Max.   :25.0
```

Selecting and Indexing Data Frame Elements

```
# select everything from the first row
df[1,]
```

```
##   days temp rain
## 1  Mon 22.2 TRUE
```

```
# select everything from the first column
df[,1]
```

```
## [1] "Mon" "Tue" "Wed" "Thu" "Fri"
```

```
## We can also index using column and row labels

# select everything from the 'rain' column
df[,'rain']
```

```
## [1]  TRUE  TRUE FALSE FALSE  TRUE
```

```
class(df[,'days']) # character
```

```
## [1] "character"
```

```
# select first 5 rows in columns 'days' and 'temp'
df[1:5,c('days','temp')]
```

```
##   days temp
## 1  Mon 22.2
## 2  Tue 21.0
## 3  Wed 23.0
## 4  Thu 24.3
## 5  Fri 25.0
```

```
## grab all the values of a particular column

# using the '$' notation
df$days # R returns a vector
```

```
## [1] "Mon" "Tue" "Wed" "Thu" "Fri"
```

```
class(df$'days') # data frame
```

```
## [1] "character"
```

```
# using bracket notation
df['days'] # R returns a data frame
```

```
##   days
## 1  Mon
## 2  Tue
## 3  Wed
## 4  Thu
## 5  Fri
```

```
class(df['days']) # character
```

```
## [1] "data.frame"
```

```
## using the `subset()` function for selection

# find out days where it rained
subset(df, subset = rain==T) # one option
```

```
##   days temp rain
## 1  Mon 22.2 TRUE
## 2  Tue 21.0 TRUE
## 5  Fri 25.0 TRUE
```

```
subset(df,df$rain==T) # an alternative
```

```
##   days temp rain
## 1  Mon 22.2 TRUE
## 2  Tue 21.0 TRUE
## 5  Fri 25.0 TRUE
```

```
# grab days where the temperature was greater than 23 degrees
subset(df, subset = temp > 23)
```

```
##   days temp  rain
## 4  Thu 24.3 FALSE
## 5  Fri 25.0  TRUE
```

```
## sorting a data frame. Use the `order()` function

# sorting daily temperatures in accenting order
sorted.temp <- order(df['temp']) # this is a mask containing row indices
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
sorted.temp
```

```
## [1] 2 1 3 4 5
```

```
df[sorted.temp,] # apply mask. We sort the data frame in
```

```
##   days temp  rain
## 2  Tue 21.0  TRUE
## 1  Mon 22.2  TRUE
## 3  Wed 23.0 FALSE
## 4  Thu 24.3 FALSE
## 5  Fri 25.0  TRUE
```

```
# Now let's sort the data frame in descending order
desc.temp <- order(-df['temp'])
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
df[desc.temp,] # apply mask
```

```
##   days temp  rain
## 5  Fri 25.0  TRUE
## 4  Thu 24.3 FALSE
## 3  Wed 23.0 FALSE
## 1  Mon 22.2  TRUE
## 2  Tue 21.0  TRUE
```

Overview of Data Frame Operations

Here we'll cover the following: - Creating data frames - Importing and exporting data - Getting information about a data frame - Referencing cells - Referencing rows - Referencing columns - Adding rows - Adding columns - Setting column names - Selecting multiple columns - Dealing with missing data

Creating data frames

```
# create an empty dataframe
empty <- data.frame()

# create a dataframe from vectors

c1 <- 1:10 # numbers, 1 through 10
letters # the letters of the alphabet. They're built into R
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
c2 <- letters[1:10] # create a vector of the first 10 letters of the alphabet
c2
```

```
##  [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j"
```

```
# combine c1 and c2 into a dataframe

df <- data.frame(c1,c2) # without column names assigned
df
```

```
##    c1 c2
## 1   1  a
## 2   2  b
## 3   3  c
## 4   4  d
## 5   5  e
## 6   6  f
## 7   7  g
## 8   8  h
## 9   9  i
## 10 10  j
```

```
df <- data.frame(col.name.1=c1,col.name.2=c2) # assign each column a name
df
```

```
##    col.name.1 col.name.2
## 1           1          a
## 2           2          b
## 3           3          c
## 4           4          d
## 5           5          e
## 6           6          f
## 7           7          g
## 8           8          h
## 9           9          i
## 10         10          j
```

Importing and exporting data

```
# to write a .csv file
## this saves the last dataframe (i.e., df) to our working directory
## this also saves our index column as a separate column
write.csv(df, file = 'saved_df.csv')

#  reading a .csv file into R
df2 <- read.csv('saved_df.csv')
df2
```

```
##     X col.name.1 col.name.2
## 1   1          1          a
## 2   2          2          b
## 3   3          3          c
## 4   4          4          d
## 5   5          5          e
## 6   6          6          f
## 7   7          7          g
## 8   8          8          h
## 9   9          9          i
## 10 10         10          j
```

Getting information about a data frame

```
# find out the numbers of rows and columns
nrow(df)
```

```
## [1] 10
```

```
ncol(df)
```

```
## [1] 2
```

```r
# find out the names of rows and columns
colnames(df)
```

```
## [1] "col.name.1" "col.name.2"
```

```r
rownames(df) # these are the indices too
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```r
# get the structure of the dataframe
str(df)
```

```
## 'data.frame':    10 obs. of  2 variables:
##  $ col.name.1: int  1 2 3 4 5 6 7 8 9 10
##  $ col.name.2: chr  "a" "b" "c" "d" ...
```

```r
# get a summary of the dataframe
summary(df)
```

```
##    col.name.1     col.name.2
##  Min.   : 1.00   Length:10
##  1st Qu.: 3.25   Class :character
##  Median : 5.50   Mode  :character
##  Mean   : 5.50
##  3rd Qu.: 7.75
##  Max.   :10.00
```

Referencing cells

```r
# reference a cell in the dataframe, in row 5 and column 2
df[[5,2]] # just using row and column indices
```

```
## [1] "e"
```

```r
df[[5,'col.name.2']] # by using the column name
```

```
## [1] "e"
```

```r
# change a value in the dataframe
df[[2,'col.name.1']] <- 9999 # change value 2 at the intersection of row 2 and column 1
df
```

```
##    col.name.1 col.name.2
## 1           1          a
## 2        9999          b
## 3           3          c
## 4           4          d
## 5           5          e
## 6           6          f
## 7           7          g
## 8           8          h
## 9           9          i
## 10         10          j
```

Referencing rows

```
df[1,] # show row 1. Returns a dataframe
```

```
##   col.name.1 col.name.2
## 1          1          a
```

```
class(df[1,]) # "data.frame"
```

```
## [1] "data.frame"
```

```
as.numeric(df[1,]) # show row 1. Returns a vector instead of a dataframe
```

```
## Warning: NAs introduced by coercion
```

```
## [1]  1 NA
```

```
class(as.numeric(df[1,])) # "numeric"
```

```
## Warning: NAs introduced by coercion
```

```
## [1] "numeric"
```

Referencing columns

```
head(mtcars) # check out this built-in dataframe
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
# 4 ways to grab the 'mpg' column as a vector
mtcars$mpg
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
mtcars[,'mpg']
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
mtcars[,1] # 'mpg' is the first column
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
mtcars[['mpg']]
```

```
##  [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2 10.4
## [16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8 19.7
## [31] 15.0 21.4
```

```
# 2 ways to grab the 'mpg' column as a dataframe
mtcars['mpg']
```

```
##                      mpg
## Mazda RX4           21.0
## Mazda RX4 Wag       21.0
## Datsun 710          22.8
## Hornet 4 Drive      21.4
## Hornet Sportabout   18.7
## Valiant             18.1
## Duster 360          14.3
## Merc 240D           24.4
## Merc 230            22.8
## Merc 280            19.2
## Merc 280C           17.8
## Merc 450SE          16.4
## Merc 450SL          17.3
## Merc 450SLC         15.2
## Cadillac Fleetwood  10.4
## Lincoln Continental 10.4
## Chrysler Imperial   14.7
## Fiat 128            32.4
## Honda Civic         30.4
## Toyota Corolla      33.9
## Toyota Corona       21.5
## Dodge Challenger    15.5
## AMC Javelin         15.2
## Camaro Z28          13.3
## Pontiac Firebird    19.2
## Fiat X1-9           27.3
## Porsche 914-2       26.0
## Lotus Europa        30.4
## Ford Pantera L      15.8
## Ferrari Dino        19.7
## Maserati Bora       15.0
## Volvo 142E          21.4
```

```
mtcars[1] # same result using the index/order of the column
```

```
##                      mpg
## Mazda RX4           21.0
## Mazda RX4 Wag       21.0
## Datsun 710          22.8
## Hornet 4 Drive      21.4
## Hornet Sportabout   18.7
## Valiant             18.1
## Duster 360          14.3
## Merc 240D           24.4
## Merc 230            22.8
## Merc 280            19.2
## Merc 280C           17.8
## Merc 450SE          16.4
## Merc 450SL          17.3
## Merc 450SLC         15.2
## Cadillac Fleetwood  10.4
## Lincoln Continental 10.4
## Chrysler Imperial   14.7
## Fiat 128            32.4
## Honda Civic         30.4
## Toyota Corolla      33.9
## Toyota Corona       21.5
## Dodge Challenger    15.5
## AMC Javelin         15.2
## Camaro Z28          13.3
## Pontiac Firebird    19.2
## Fiat X1-9           27.3
## Porsche 914-2       26.0
## Lotus Europa        30.4
## Ford Pantera L      15.8
## Ferrari Dino        19.7
## Maserati Bora       15.0
## Volvo 142E          21.4
```

```
## selecting multiple columns as a dataframe
head(mtcars[c('mpg','disp')])
```

```
##                    mpg disp
## Mazda RX4         21.0  160
## Mazda RX4 Wag     21.0  160
## Datsun 710        22.8  108
## Hornet 4 Drive    21.4  258
## Hornet Sportabout 18.7  360
## Valiant           18.1  225
```

Adding rows

```
# create a new dataframe
df2 <- data.frame(col.name.1 = 2000, col.name.2 = 'new')
df2
```

```
##   col.name.1 col.name.2
```

```
## 1       2000        new
```

```
# combine this last dataframe into an existing one
dfnew <- rbind(df,df2)
dfnew[11,] # preview the 11th row in the dataframe
```

```
##    col.name.1 col.name.2
## 11       2000        new
```

Adding columns

```
# add a new column to 'df' that's twice the values in column 'col.name.1'
df$newcol <- 2*df$col.name.1
df
```

```
##    col.name.1 col.name.2 newcol
## 1           1          a      2
## 2        9999          b  19998
## 3           3          c      6
## 4           4          d      8
## 5           5          e     10
## 6           6          f     12
## 7           7          g     14
## 8           8          h     16
## 9           9          i     18
## 10         10          j     20
```

```
# alternative approach
df['newcol.copy'] <- df$newcol
df
```

```
##    col.name.1 col.name.2 newcol newcol.copy
## 1           1          a      2           2
## 2        9999          b  19998       19998
## 3           3          c      6           6
## 4           4          d      8           8
## 5           5          e     10          10
## 6           6          f     12          12
## 7           7          g     14          14
## 8           8          h     16          16
## 9           9          i     18          18
## 10         10          j     20          20
```

Setting column names

```
# retrive column names of the 'df' dataframe
colnames(df)
```

```
## [1] "col.name.1"  "col.name.2"  "newcol"      "newcol.copy"
```

```r
# rename the columns
colnames(df) <- c('1','2','3','4')
df
```

```
##       1 2     3     4
## 1     1 a     2     2
## 2  9999 b 19998 19998
## 3     3 c     6     6
## 4     4 d     8     8
## 5     5 e    10    10
## 6     6 f    12    12
## 7     7 g    14    14
## 8     8 h    16    16
## 9     9 i    18    18
## 10   10 j    20    20
```

```r
# rename a specific column in the dataframe
colnames(df)[1] <- c('NEW COL NAME')
df
```

```
##    NEW COL NAME 2     3     4
## 1             1 a     2     2
## 2          9999 b 19998 19998
## 3             3 c     6     6
## 4             4 d     8     8
## 5             5 e    10    10
## 6             6 f    12    12
## 7             7 g    14    14
## 8             8 h    16    16
## 9             9 i    18    18
## 10           10 j    20    20
```

Selecting multiple ROWS

```r
# SELECT FIRST 10 ROWS
df[1:10,]
```

```
##    NEW COL NAME 2     3     4
## 1             1 a     2     2
## 2          9999 b 19998 19998
## 3             3 c     6     6
## 4             4 d     8     8
## 5             5 e    10    10
## 6             6 f    12    12
## 7             7 g    14    14
## 8             8 h    16    16
## 9             9 i    18    18
## 10           10 j    20    20
```

```r
# using the head() method, select first 7 rows
head(df,7)
```

```
##   NEW COL NAME 2     3     4
## 1            1 a     2     2
## 2         9999 b 19998 19998
## 3            3 c     6     6
## 4            4 d     8     8
## 5            5 e    10    10
## 6            6 f    12    12
## 7            7 g    14    14
```

```r
# select everything but row 2
df[-2,]
```

```
##    NEW COL NAME 2  3  4
## 1             1 a  2  2
## 3             3 c  6  6
## 4             4 d  8  8
## 5             5 e 10 10
## 6             6 f 12 12
## 7             7 g 14 14
## 8             8 h 16 16
## 9             9 i 18 18
## 10           10 j 20 20
```

```r
## conditional selecting
head(mtcars) # use the 'mtcars' built-in dataset
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```r
# select on rows where 'mpg' is greater than 20
mtcars[mtcars$mpg > 20, ]
```

```
##                 mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Merc 240D      24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230       22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Fiat 128       32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic    30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla 33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
```

```
## Fiat X1-9      27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

```
# using multiple conditions
mtcars[ (mtcars$mpg>20 & mtcars$cyl==6), ]
```

```
##                 mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
```

```
mtcars[ (mtcars$mpg>20 & mtcars$cyl==6), c('mpg','cyl','hp')]
```

```
##                 mpg cyl  hp
## Mazda RX4      21.0   6 110
## Mazda RX4 Wag  21.0   6 110
## Hornet 4 Drive 21.4   6 110
```

```
## using the subset() function
subset(mtcars, (mpg>20 & cyl==6) )
```

```
##                 mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4      21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag  21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
```

Selecting multiple columns

```
# select multiple columns from 'mtcars'
mtcars[,c(1,2,3)] # first 3 columns
```

```
##                     mpg cyl  disp
## Mazda RX4          21.0   6 160.0
## Mazda RX4 Wag      21.0   6 160.0
## Datsun 710         22.8   4 108.0
## Hornet 4 Drive     21.4   6 258.0
## Hornet Sportabout  18.7   8 360.0
## Valiant            18.1   6 225.0
## Duster 360         14.3   8 360.0
## Merc 240D          24.4   4 146.7
## Merc 230           22.8   4 140.8
## Merc 280           19.2   6 167.6
## Merc 280C          17.8   6 167.6
## Merc 450SE         16.4   8 275.8
## Merc 450SL         17.3   8 275.8
## Merc 450SLC        15.2   8 275.8
## Cadillac Fleetwood 10.4   8 472.0
```

```
## Lincoln Continental 10.4   8 460.0
## Chrysler Imperial   14.7   8 440.0
## Fiat 128            32.4   4  78.7
## Honda Civic         30.4   4  75.7
## Toyota Corolla      33.9   4  71.1
## Toyota Corona       21.5   4 120.1
## Dodge Challenger    15.5   8 318.0
## AMC Javelin         15.2   8 304.0
## Camaro Z28          13.3   8 350.0
## Pontiac Firebird    19.2   8 400.0
## Fiat X1-9           27.3   4  79.0
## Porsche 914-2       26.0   4 120.3
## Lotus Europa        30.4   4  95.1
## Ford Pantera L      15.8   8 351.0
## Ferrari Dino        19.7   6 145.0
## Maserati Bora       15.0   8 301.0
## Volvo 142E          21.4   4 121.0
```

```
mtcars[,c('mpg','cyl','disp')]
```

```
##                     mpg cyl  disp
## Mazda RX4          21.0   6 160.0
## Mazda RX4 Wag      21.0   6 160.0
## Datsun 710         22.8   4 108.0
## Hornet 4 Drive     21.4   6 258.0
## Hornet Sportabout  18.7   8 360.0
## Valiant            18.1   6 225.0
## Duster 360         14.3   8 360.0
## Merc 240D          24.4   4 146.7
## Merc 230           22.8   4 140.8
## Merc 280           19.2   6 167.6
## Merc 280C          17.8   6 167.6
## Merc 450SE         16.4   8 275.8
## Merc 450SL         17.3   8 275.8
## Merc 450SLC        15.2   8 275.8
## Cadillac Fleetwood 10.4   8 472.0
## Lincoln Continental 10.4   8 460.0
## Chrysler Imperial  14.7   8 440.0
## Fiat 128           32.4   4  78.7
## Honda Civic        30.4   4  75.7
## Toyota Corolla     33.9   4  71.1
## Toyota Corona      21.5   4 120.1
## Dodge Challenger   15.5   8 318.0
## AMC Javelin        15.2   8 304.0
## Camaro Z28         13.3   8 350.0
## Pontiac Firebird   19.2   8 400.0
## Fiat X1-9          27.3   4  79.0
## Porsche 914-2      26.0   4 120.3
## Lotus Europa       30.4   4  95.1
## Ford Pantera L     15.8   8 351.0
## Ferrari Dino       19.7   6 145.0
## Maserati Bora      15.0   8 301.0
## Volvo 142E         21.4   4 121.0
```

Dealing with missing data

```
sum(is.na(df)) # get booleans
```

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

```
any(is.na(df)) # get a single FALSE/TRUE boolean for whether we have at leat 1 case of missing values
```

```
## [1] FALSE
```

```
sum(is.na(df)) # get total number of missing values
```

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

```
## replace all NULL values with zero
df[is.na(df)] <- 0

## replace missing values in column 'mpg' of built-in dataset 'mtcars' by the column mean
mtcars$mpg[is.na(mtcars$mpg)] <- mean(mtcars$mpg)
```

Data Frame Training Exercise

Ex 1: Recreate the following dataframe by creating vectors and using the data.frame function:

```
Ages <- c(22,25,26)
Weight <- c(150,165,120)
Sex <- c('M','M','F')

mydf <- data.frame(Ages, Weight, Sex, row.names = c('Sam','Frank','Amy'))
mydf
```

```
##       Ages Weight Sex
## Sam     22    150   M
## Frank   25    165   M
## Amy     26    120   F
```

Ex 2: Check if mtcars is a dataframe using is.data.frame()

```
is.data.frame(mtcars)
```

```
## [1] TRUE
```

Ex 3: Use as.data.frame() to convert a matrix into a dataframe:

```
mat <- matrix(1:25,nrow = 5)
df_mat <- as.data.frame(mat)
df_mat
```

```
##   V1 V2 V3 V4 V5
## 1  1  6 11 16 21
## 2  2  7 12 17 22
## 3  3  8 13 18 23
## 4  4  9 14 19 24
## 5  5 10 15 20 25
```

Ex 4: Set the built-in data frame mtcars as a variable df. We'll use this df variable for the rest of the exercises.

```
df <- mtcars
df
```

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
## Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
## Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
## Honda Civic       30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Dodge Challenger  15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28        13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird  19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
## Fiat X1-9         27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Lotus Europa      30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Ford Pantera L    15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
## Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

Ex 5: Display the first 6 rows of df

```
head(df,6)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
# df[1:6,] # alternatively
```

Ex 6: What is the average mpg value for all the cars?

```
mean(df$mpg)
```

```
## [1] 20.09062
```

Ex 7: Select the rows where all cars have 6 cylinders (cyl column)

```
df[df$cyl == 6,]
```

```
##                  mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4       21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag   21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Valiant         18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Merc 280        19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
## Merc 280C       17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## Ferrari Dino    19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
```

Ex 8: Select the columns am,gear, and carb.

```
df[ , c('am','gear','carb')]
```

```
##                    am gear carb
## Mazda RX4           1    4    4
## Mazda RX4 Wag       1    4    4
## Datsun 710          1    4    1
## Hornet 4 Drive      0    3    1
## Hornet Sportabout   0    3    2
## Valiant             0    3    1
## Duster 360          0    3    4
## Merc 240D           0    4    2
## Merc 230            0    4    2
## Merc 280            0    4    4
## Merc 280C           0    4    4
## Merc 450SE          0    3    3
## Merc 450SL          0    3    3
## Merc 450SLC         0    3    3
```

```
## Cadillac Fleetwood   0    3    4
## Lincoln Continental  0    3    4
## Chrysler Imperial    0    3    4
## Fiat 128             1    4    1
## Honda Civic          1    4    2
## Toyota Corolla       1    4    1
## Toyota Corona        0    3    1
## Dodge Challenger     0    3    2
## AMC Javelin          0    3    2
## Camaro Z28           0    3    4
## Pontiac Firebird     0    3    2
## Fiat X1-9            1    4    1
## Porsche 914-2        1    5    2
## Lotus Europa         1    5    2
## Ford Pantera L       1    5    4
## Ferrari Dino         1    5    6
## Maserati Bora        1    5    8
## Volvo 142E           1    4    2
```

```
# subset(df, select = c(am, gear, carb)) # alternative approach
```

Ex 9: Create a new column called performance, which is calculated by hp/wt.

```
df$performance <- df$hp / df$wt
head(df)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
##                   performance
## Mazda RX4            41.98473
## Mazda RX4 Wag        38.26087
## Datsun 710           40.08621
## Hornet 4 Drive       34.21462
## Hornet Sportabout    50.87209
## Valiant              30.34682
```

Ex 10: Your performance column will have several decimal place precision. Figure out how to use round() (check help(round)) to reduce this accuracy to only 2 decimal places.

```
df$performance <- round(df$hp / df$wt, 2)
head(df)
```

```
##                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
```

```
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
##                   performance
## Mazda RX4               41.98
## Mazda RX4 Wag           38.26
## Datsun 710              40.09
## Hornet 4 Drive          34.21
## Hornet Sportabout       50.87
## Valiant                 30.35
```

Ex 11: What is the mpg of the Hornet Sportabout?

```
df[['Hornet Sportabout', 'mpg']]
```

```
## [1] 18.7
```