Introduction to R: the tidyverse package

Ziv Shkedy, Hasselt University

```
## Warning: package 'mvtnorm' was built under R version 3.6.2
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  2.1.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## v purrr   0.3.4
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## Warning: package 'dslabs' was built under R version 3.6.3
```

```
## Warning: package 'NHANES' was built under R version 3.6.3
```

Introduction

The tidyverse

The tidyverse is a collection of R packages that work in data frames in a tidy format.All packages are uploaded in CRAN, and can be installed using install.packages . In this online tutorial we cover materials at an introduction level and follow closely the topics presented in Chapter 4 in the book Data Analysis and Prediction Algorithms with R by Rafael A. Irizarry.

What do we cover in this chapter ?

The chapter is developed at a beginner level, We cover few functions from the tidyverse packages and illustrate the basic concepts using different examples of the following functions.

- mutate()
- filter()
- select()
- The pipe: $\% > \%$
- summarize()
- group_by()
- arrange()
- top_n()

Our aim in this tutorial is not to teach ggplot2 . However, some functions of the package are used to visualized the main patterns in the data we used for examples. The following subcions are used:

- qplot()
- ggplot() + geom_jitter()
- ggplot() + geom_point()
- gplot() + geom_density()
- stripplot()

Online references

Materials about tidyverse are widely available online. We list below a selection that we find useful and clear.

YouTube tutorials: tidyverse in R

- For a YouTube tutorial about tidyverse in R by Mark Gingrass see YTtidyverse1.
- For a YouTube tutorial about tidyverse in R by Garreet Grolemund see YTtidyverse2a.
- For a YouTube tutorial about tidyverse in R by Garreet Grolemund see YTtidyverse2b.
- For a YouTube tutorial about tidyverse in R by Ben Stenhaug see YTtidyverse3.

Online book

Chapter 4 in the book: Data Analysis and Prediction Algorithms with R by Rafael A. Irizarry see Booktidyverse1.

Datasets

Many datasets will be used for illustration. All of them are data frames available in R. We do not focus in this chapter on the question how to read the data but rather on the question how to organize the data for the analysis.

The murders data

The murders dataset gives information about the number of gun murders in 51 US states (2010).

```
data("murders")
head(murders)
```

```
##        state abb region population total
## 1    Alabama  AL  South    4779736   135
## 2     Alaska  AK   West     710231    19
## 3    Arizona  AZ   West    6392017   232
## 4   Arkansas  AR  South    2915918    93
## 5 California  CA   West   37253956  1257
## 6   Colorado  CO   West    5029196    65
```

In total, five variables are included in the data.

```
dim(murders)
```

```
## [1] 51  5
```

The heights data

The heights dataset gives information about the self reported heights (in inches) for males and females of 1050 subjects.

```
data(heights)
dim(heights)
```

```
## [1] 1050    2
```

the first 6 subjects are shown below.

```
head(heights)
```

```
##      sex height
## 1   Male     75
## 2   Male     70
## 3   Male     68
## 4   Male     74
## 5   Male     61
## 6 Female     65
```

The NHANES data

The NHANES dataset consists of data from the US National Health and Nutrition Examination Study. Information about 76 variables is available for 10000 subjects.

```r
library(NHANES)
data(NHANES)
dim(NHANES)
```

```
## [1] 10000    76
```

variables neams are listed below.

```r
names(NHANES)
```

```
##  [1] "ID"               "SurveyYr"         "Gender"           "Age"
##  [5] "AgeDecade"        "AgeMonths"        "Race1"            "Race3"
##  [9] "Education"        "MaritalStatus"    "HHIncome"         "HHIncomeMid"
## [13] "Poverty"          "HomeRooms"        "HomeOwn"          "Work"
## [17] "Weight"           "Length"           "HeadCirc"         "Height"
## [21] "BMI"              "BMICatUnder20yrs" "BMI_WHO"          "Pulse"
## [25] "BPSysAve"         "BPDiaAve"         "BPSys1"           "BPDia1"
## [29] "BPSys2"           "BPDia2"           "BPSys3"           "BPDia3"
## [33] "Testosterone"     "DirectChol"       "TotChol"          "UrineVol1"
## [37] "UrineFlow1"       "UrineVol2"        "UrineFlow2"       "Diabetes"
## [41] "DiabetesAge"      "HealthGen"        "DaysPhysHlthBad"  "DaysMentHlthBad"
## [45] "LittleInterest"   "Depressed"        "nPregnancies"     "nBabies"
## [49] "Age1stBaby"       "SleepHrsNight"    "SleepTrouble"     "PhysActive"
## [53] "PhysActiveDays"   "TVHrsDay"         "CompHrsDay"       "TVHrsDayChild"
## [57] "CompHrsDayChild"  "Alcohol12PlusYr"  "AlcoholDay"       "AlcoholYear"
## [61] "SmokeNow"         "Smoke100"         "Smoke100n"        "SmokeAge"
## [65] "Marijuana"        "AgeFirstMarij"    "RegularMarij"     "AgeRegMarij"
## [69] "HardDrugs"        "SexEver"          "SexAge"           "SexNumPartnLife"
## [73] "SexNumPartYear"   "SameSex"          "SexOrientation"   "PregnantNow"
```

The Chicks Weights data

71 newly hatched chicks were randomly allocated into six groups, and each group was given a different feed supplement. Their weights (the response variable) in grams after six weeks are given along with feed types (the factor). The Chick Weights data is a data frame in R called chickwts .

```r
head(chickwts)
```

```
##   weight      feed
## 1    179 horsebean
## 2    160 horsebean
## 3    136 horsebean
## 4    227 horsebean
## 5    217 horsebean
## 6    168 horsebean
```

```r
dim(chickwts)
```

```
## [1] 71  2
```

The Chicken Weights data

The ChickWeight dataset is a data frame with 578 rows and 4 columns from an experiment on the effect of diet on early growth of chicks. Each chick was measuered 12 times over a period of 21 days.

```r
head(ChickWeight)
```

```
## Grouped Data: weight ~ Time | Chick
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

Note that each row in the data represents the chick weight in a specific day.

```r
dim(ChickWeight)
```

```
## [1] 578   4
```

The cars data

The cars data and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models).

```r
dim(mtcars)
```

```
## [1] 32 11
```

Data manipulation with the tidyverse package

Tidy data

Tidy data is a data format in which each row represents one measurement for one observation and columns are, as usual, the variables in the data.

The murders data

The murders is an example of a tidy data. The information for each state is given in one line. Note that in this case, each observation (=state) has information in one data line.

```
data("murders")
head(murders)
```

```
##        state abb region population total
## 1    Alabama  AL  South    4779736   135
## 2     Alaska  AK   West     710231    19
## 3    Arizona  AZ   West    6392017   232
## 4   Arkansas  AR  South    2915918    93
## 5 California  CA   West   37253956  1257
## 6   Colorado  CO   West    5029196    65
```

The murder rate by region is shown in the stripplot presented in Figure~@ref(fig:fig1b) that shows clearly that in the west, the murder rate is the lowest.

```
ggplot(murders, aes(region,population)) + geom_jitter(position = position_jitter(width = .05))
```

The ChickWeight data

In the Chicken Weight data, each observation is a chick and it was measured in 12 times points.

```
unique(ChickWeight$Time)
```

```
##  [1]  0  2  4  6  8 10 12 14 16 18 20 21
```

In the data frame, each measurement is presented in one data line. Hence, the ChickWeight is a tidy data. This implies that that the data for each observation is presented in 12 lines. Data for the first 6 time points of the first chick is listed below.

```
head(ChickWeight)
```

```
## Grouped Data: weight ~ Time | Chick
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     51    2     1    1
## 3     59    4     1    1
## 4     64    6     1    1
## 5     76    8     1    1
## 6     93   10     1    1
```

The boxplot of the chicken weights by time point, presented in Figure~@ref(fig:fig1c), shows the increasing trend of the weight over time.

```
ggplot(ChickWeight, aes(as.factor(Time),weight)) + geom_boxplot()
```

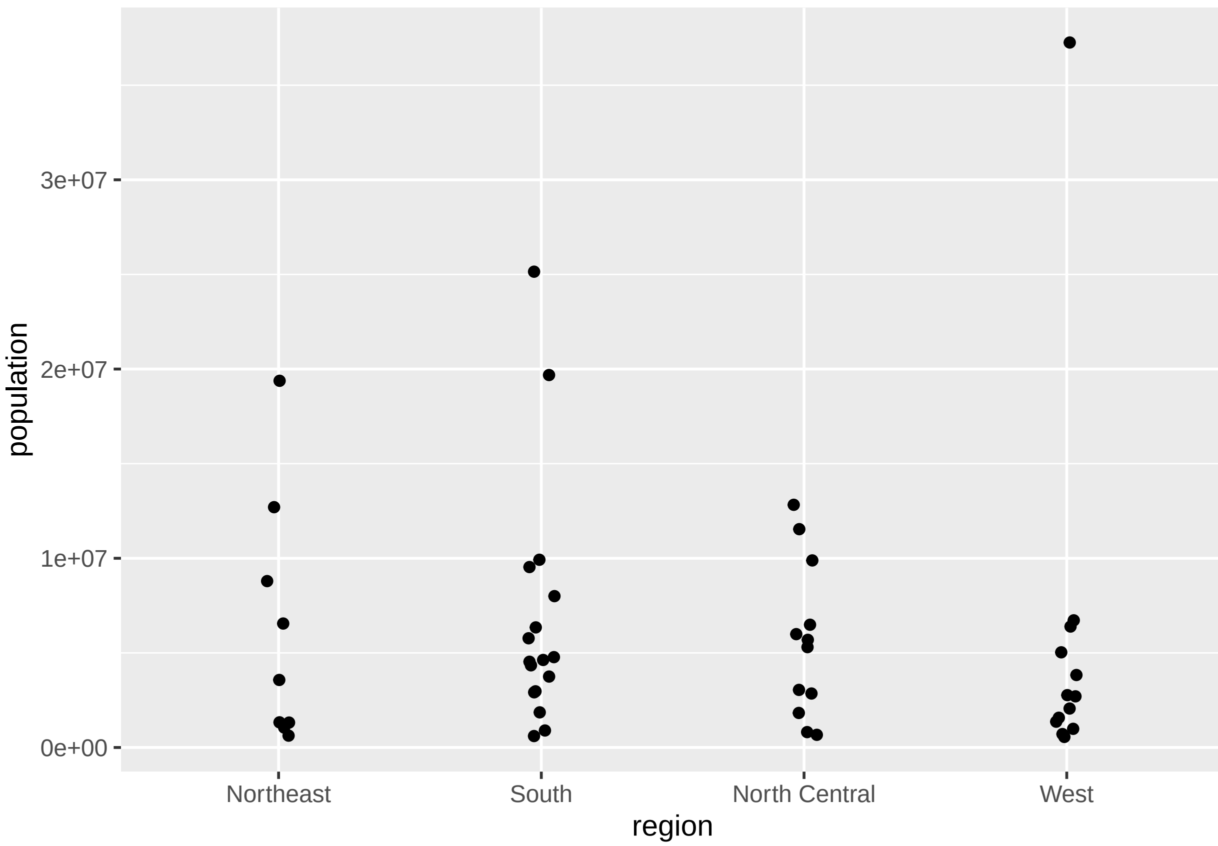

Figure 1: Dotplot using the gg2plot package

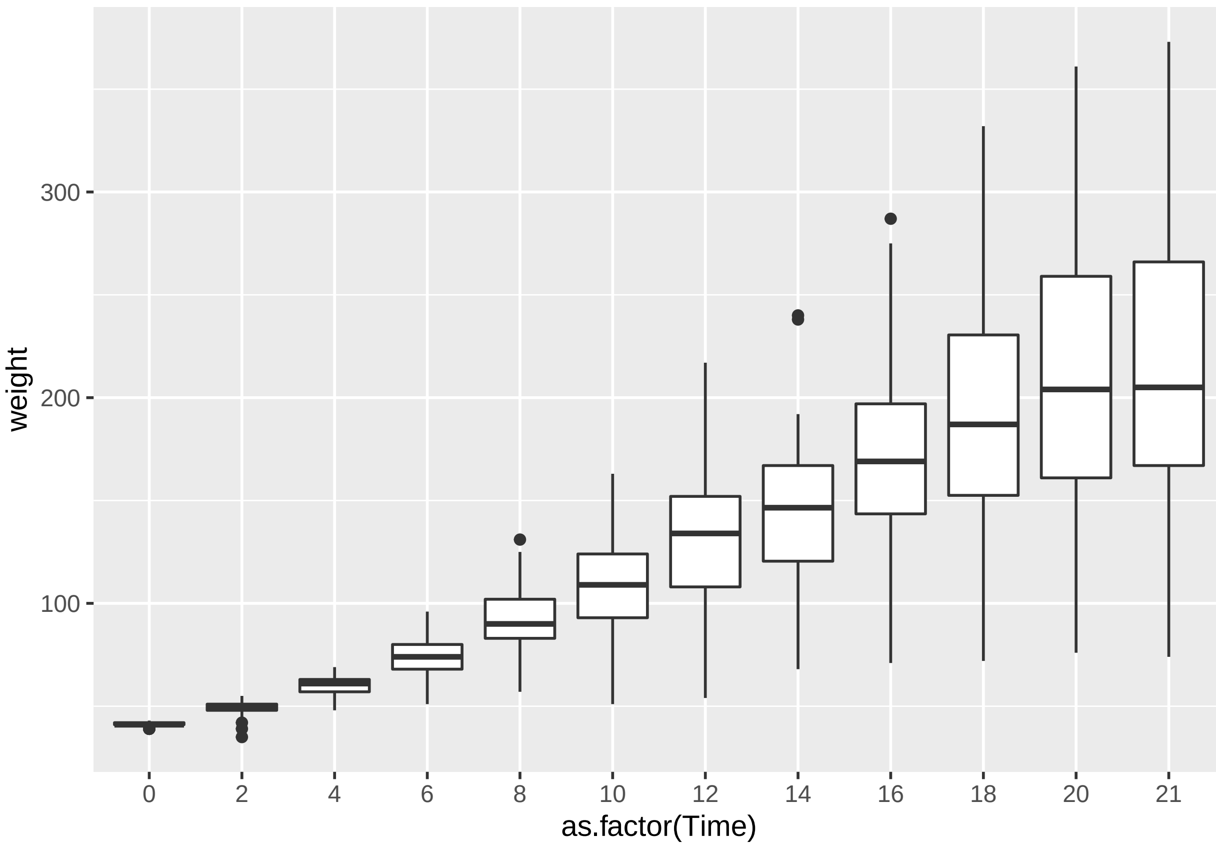

Figure 2: Boxplot for the singers data (I.1).

Adding a variable (column)

Suppose that we would like to calculate the murder rate per 100000 people that is

$$\frac{total}{population} \times 100000.$$

This can be done using the mutate function that has the grneral call of: mutate(data frame , new variable) .

The murders data

For the murders data we have

```
data("murders")
murders <- mutate(murders, rate = total / population * 100000)
```

Note that after calculating the murder rate, the murders has an extra column (=variable).

```
head(murders)
```

```
##        state abb region population total     rate
## 1    Alabama  AL  South    4779736   135 2.824424
## 2     Alaska  AK   West     710231    19 2.675186
## 3    Arizona  AZ   West    6392017   232 3.629527
## 4   Arkansas  AR  South    2915918    93 3.189390
## 5 California  CA   West   37253956  1257 3.374138
## 6   Colorado  CO   West    5029196    65 1.292453
```

The NHANES data

The BMI of a person is given by

$$BMI = \frac{weight}{height^2}.$$

To calculate the BMI in the NHANES we use

```
data("NHANES")
Data_new <- mutate(NHANES, BMI_new = Weight / (Height*Height))
```

The histogram of the BMI is shown in Figure~@ref(fig:fig1d).

```
qplot(BMI_new , data=Data_new, geom="histogram")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 366 rows containing non-finite values (stat_bin).
```

Filtering

Selection of observation for the data can be done using the function filter() .

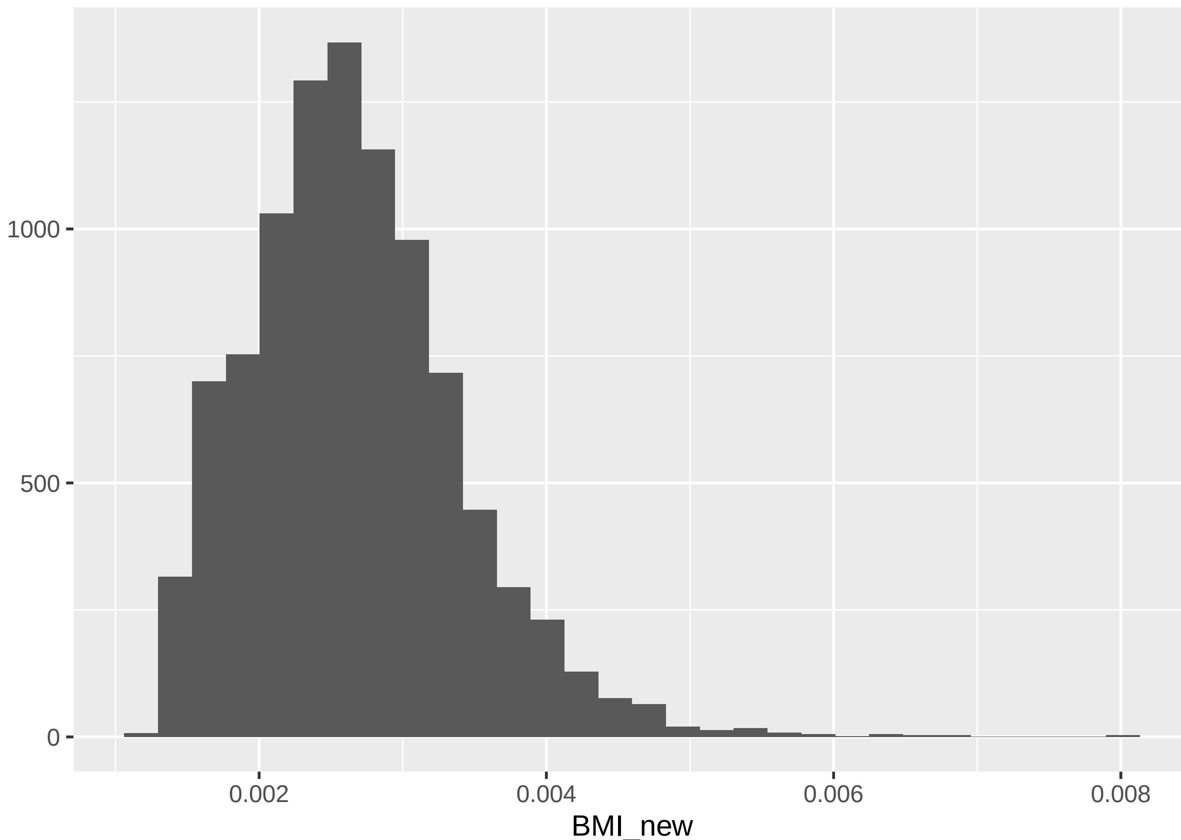

Figure 3: Histogram for the BMI.

The murders data

For murder dataset, we can select all the states with murder rate ≤ 0.71 by

```
filter(murders, rate <= 0.71)
```

```
##          state abb        region population total      rate
## 1       Hawaii  HI          West    1360301     7 0.5145920
## 2         Iowa  IA North Central    3046355    21 0.6893484
## 3 New Hampshire  NH     Northeast    1316470     5 0.3798036
## 4  North Dakota  ND North Central     672591     4 0.5947151
## 5      Vermont  VT     Northeast     625741     2 0.3196211
```

The cars data

For cars data, suppose that we would like to plot the cars' weight versus the cars mpg for cars with weight ≤ 3.

```
mtcars1<-filter(mtcars, wt <= 3)
```

The new data frame mtcars1 contains the information for all cars with weight lower than 3,

```
mtcars1
```

```
##                 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
## 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
## Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

The scaterplot in Figure~@ref(fig:fig1e) below of the weight versus the mpg can be produce using the following code.

```
ggplot(mtcars1, aes(x=wt, y=mpg)) +
  geom_point( color="#69b3a2")
```

Selecting columns

In the previous section we use the function filter() to select observations. IN this section we focus on variable selection from the data frame using the function select () .

The murders data

Originaly, the murder data frame has 6 variables.

```
dim(murders)
```

```
## [1] 51  6
```

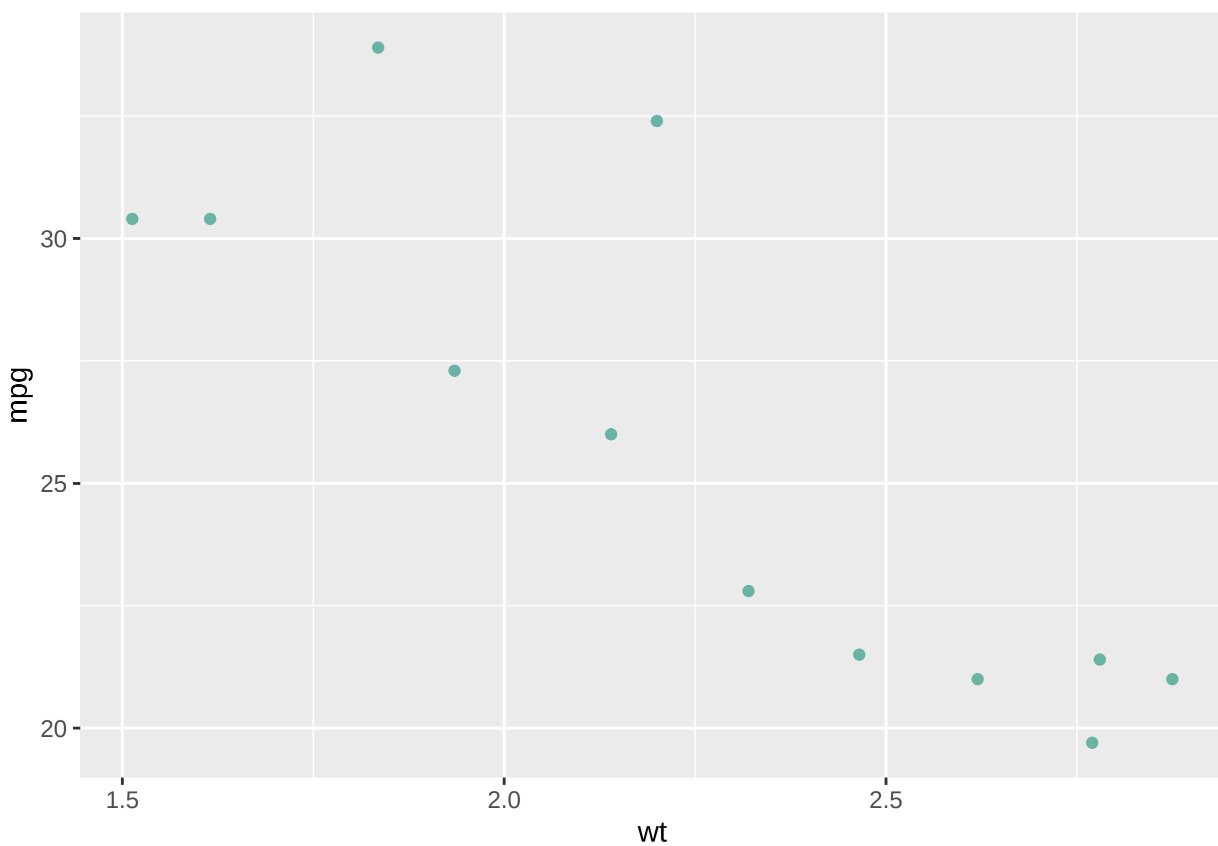

Figure 4: mile per galllon versus weight.

Originally, the murder data frame has 6 variables. We define a new data frame newdata that contains only 3 of the variables in the murder data frame.

```
newdata <- select(murders, state, region, rate)
dim(newdata)
```

```
## [1] 51  3
```

Note that we can further filter the observations, for example a selection of all observations with murder rate lower than 0.71:

```
filter(newdata, rate <= 0.71)
```

```
##            state        region      rate
## 1         Hawaii          West 0.5145920
## 2           Iowa North Central 0.6893484
## 3  New Hampshire     Northeast 0.3798036
## 4   North Dakota North Central 0.5947151
## 5        Vermont     Northeast 0.3196211
```

The NHANES data

In this example, we define a new data frame with contains 6 variables from the NHANES .

```
NHANES1 <- select(NHANES, Gender, Age, Weight, Height,BMI,Diabetes)
dim(NHANES1)
```

```
## [1] 10000     6
```

```
head(NHANES1)
```

```
## # A tibble: 6 x 6
##   Gender   Age Weight Height   BMI Diabetes
##   <fct>  <int>  <dbl>  <dbl> <dbl> <fct>
## 1 male      34   87.4   165.  32.2 No
## 2 male      34   87.4   165.  32.2 No
## 3 male      34   87.4   165.  32.2 No
## 4 male       4   17     105.  15.3 No
## 5 female    49   86.7   168.  30.6 No
## 6 male       9   29.8   133.  16.8 No
```

A density plot of the BMI by gender is shown in Figure~@ref(fig:fig1fa).

```
ggplot(data=NHANES1, aes(x=BMI, group=Gender, fill=Gender)) +
    geom_density(adjust=1.5)
```

```
## Warning: Removed 366 rows containing non-finite values (stat_density).
```

Alternatively, we can present the density in a separate panel per gender group ad shown in Figure~@ref(fig:fig1fb).

```
ggplot(data=NHANES1, aes(x=BMI, group=Gender, fill=Gender)) +
    geom_density(adjust=1.5)+
    facet_wrap(~Gender)
```

The pipe: $\% > \%$

In the previous section we use the functions filter() and select () to select a part of the dataset in two steps. In this section we use the pipe $\% > \%$ to make the selection in one step.

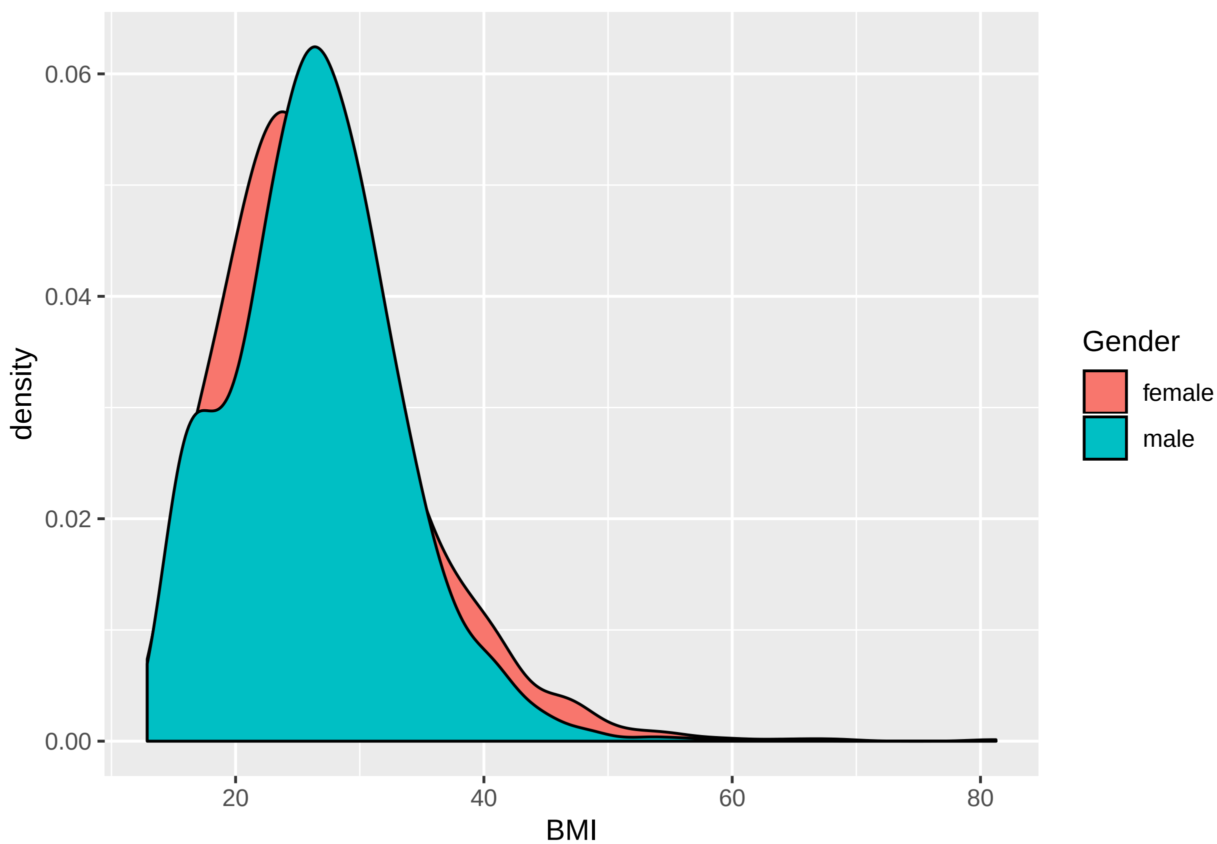

Figure 5: Density plot for the BMI by gender (I).

The murders data

We want to select from the murder data frame three variables and all the states with murder rate lower or equal to 0.71. The two selection conditions can be applied to the data frame in one step in the following way:

```
murders %>% select(state, region, rate) %>% filter(rate <= 0.71)
```

```
##           state        region      rate
## 1        Hawaii          West 0.5145920
## 2          Iowa North Central 0.6893484
## 3 New Hampshire     Northeast 0.3798036
## 4  North Dakota North Central 0.5947151
## 5       Vermont     Northeast 0.3196211
```

The region is a factor variable with four levels

```
murders$region
```

```
##  [1] South         West          West          South         West
##  [6] West          Northeast     South         South         South
## [11] South         West          West          North Central North Central
## [16] North Central North Central South         South         Northeast
## [21] South         Northeast     North Central North Central South
## [26] North Central West          North Central West          Northeast
## [31] Northeast     West          Northeast     South         North Central
## [36] North Central South         West          Northeast     Northeast
## [41] South         North Central South         South         West
## [46] Northeast     South         West          South         North Central
## [51] West
## Levels: Northeast South North Central West
```

We define a new data frame which contains the states from the Northeast and West regions

```
data1<-filter(murders, region %in% c("Northeast", "West"))
data1
```

```
##            state abb    region population total      rate
## 1         Alaska  AK      West     710231    19 2.6751860
## 2        Arizona  AZ      West    6392017   232 3.6295273
## 3     California  CA      West   37253956  1257 3.3741383
## 4       Colorado  CO      West    5029196    65 1.2924531
## 5    Connecticut  CT Northeast    3574097    97 2.7139722
## 6         Hawaii  HI      West    1360301     7 0.5145920
## 7          Idaho  ID      West    1567582    12 0.7655102
## 8          Maine  ME Northeast    1328361    11 0.8280881
## 9  Massachusetts  MA Northeast    6547629   118 1.8021791
## 10       Montana  MT      West     989415    12 1.2128379
## 11        Nevada  NV      West    2700551    84 3.1104763
## 12 New Hampshire  NH Northeast    1316470     5 0.3798036
## 13    New Jersey  NJ Northeast    8791894   246 2.7980319
## 14    New Mexico  NM      West    2059179    67 3.2537239
## 15      New York  NY Northeast   19378102   517 2.6679599
## 16        Oregon  OR      West    3831074    36 0.9396843
## 17  Pennsylvania  PA Northeast   12702379   457 3.5977513
## 18  Rhode Island  RI Northeast    1052567    16 1.5200933
## 19          Utah  UT      West    2763885    22 0.7959810
## 20       Vermont  VT Northeast     625741     2 0.3196211
```

```
## 21    Washington  WA      West    6724540    93 1.3829942
## 22       Wyoming  WY      West     563626     5 0.8871131
```

We select all states from the Northeast and West regions with mtder rate lower or equal to 1

```
filter(data1, rate <=1)
```

```
##             state abb    region population total      rate
## 1          Hawaii  HI      West    1360301     7 0.5145920
## 2           Idaho  ID      West    1567582    12 0.7655102
## 3           Maine  ME Northeast    1328361    11 0.8280881
## 4 New Hampshire  NH Northeast    1316470     5 0.3798036
## 5          Oregon  OR      West    3831074    36 0.9396843
## 6            Utah  UT      West    2763885    22 0.7959810
## 7         Vermont  VT Northeast     625741     2 0.3196211
## 8         Wyoming  WY      West     563626     5 0.8871131
```

In one step, the selection above can be implemented with the following code:

```
data2<-filter(murders, region %in% c("Northeast", "West") & rate <= 1)
print(data2)
```

```
##             state abb    region population total      rate
## 1          Hawaii  HI      West    1360301     7 0.5145920
## 2           Idaho  ID      West    1567582    12 0.7655102
## 3           Maine  ME Northeast    1328361    11 0.8280881
## 4 New Hampshire  NH Northeast    1316470     5 0.3798036
## 5          Oregon  OR      West    3831074    36 0.9396843
## 6            Utah  UT      West    2763885    22 0.7959810
## 7         Vermont  VT Northeast     625741     2 0.3196211
## 8         Wyoming  WY      West     563626     5 0.8871131
```

```
select(data2,state,region,population)
```

```
##             state    region population
## 1          Hawaii      West    1360301
## 2           Idaho      West    1567582
## 3           Maine Northeast    1328361
## 4 New Hampshire Northeast    1316470
## 5          Oregon      West    3831074
## 6            Utah      West    2763885
## 7         Vermont Northeast     625741
## 8         Wyoming      West     563626
```

The NHANES data

We select 6 variables for all female in the NHANES data frame

```
NHANES1<-NHANES %>% select(Gender, Age, Weight, Height,BMI,Diabetes) %>% filter(Gender %in% c("female"))
head(NHANES1)
```

```
## # A tibble: 6 x 6
##   Gender   Age Weight Height   BMI Diabetes
##   <fct>  <int>  <dbl>  <dbl> <dbl> <fct>
## 1 female    49   86.7   168.  30.6 No
## 2 female    45   75.7   167.  27.2 No
## 3 female    45   75.7   167.  27.2 No
```

```
## 4 female    45   75.7   167.  27.2 No
## 5 female    10   38.6   142.  19.2 No
## 6 female    58   57.5   148.  26.2 No
```

Figure~@ref(fig:fig1g) shows the distribution of the BMI by diabetes group for the feample in the NHANES data frame.

```
stripplot(Diabetes ~ jitter(BMI),
            data = NHANES1 ,
            aspect = 1, jitter = T,
            xlab="BMI", col = 1)
```

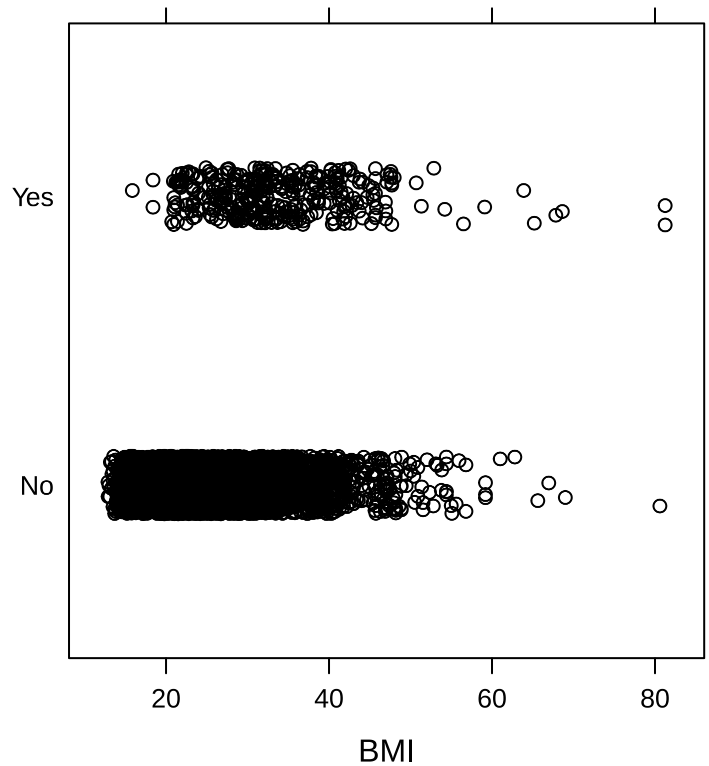

Figure 6: BMI by Diabetes group (for female).

The summarize function

The function summarize() allows us to produce summery statistics for variables in the data frame.

heights data

The height data frame gives the height by gender for 1050 individuals

```
library(dplyr)
library(dslabs)
```

```
data(heights)
dim(heights)
```

```
## [1] 1050    2
```

```
head(heights)
```

```
##      sex height
## 1   Male     75
## 2   Male     70
## 3   Male     68
## 4   Male     74
## 5   Male     61
## 6 Female     65
```

We can calculate the mean and standard deviation for female using the function summarize () . Note that we first filter the data and define and subgroup contains the data for female

```
s <- heights %>%  filter(sex == "Female") %>%
  summarize(average = mean(height), standard_deviation = sd(height))
```

The object s stores the results

```
s
```

```
##    average standard_deviation
## 1 64.93942           3.760656
```

We can define a vector that contains the results

```
c(s$average,s$standard_deviation)
```

```
## [1] 64.939424  3.760656
```

Alternatively, we can define a vector with the female heights (height.female) and calculate the mean and standard deviation for this vector.

```
height.female<-heights$height[heights$sex == "Female"]
mean(height.female)
```

```
## [1] 64.93942
```

```
sd(height.female)
```

```
## [1] 3.760656
```

The median, minimum and maximum height for female

```
heights %>%
  filter(sex == "Female") %>% summarize(median = median(height), minimum = min(height),
                                       maximum = max(height))
```

```
##     median minimum maximum
## 1 64.98031      51      79
```

These summary statistics can be also calculate can the function quantile .

```
heights %>%  filter(sex == "Female") %>%
  summarize(range = quantile(height, c(0, 0.5, 1)))
```

```
##      range
## 1 51.00000
```

```
## 2 64.98031
## 3 79.00000
```

The chicks data

To calculate the mean and standard deviatio for the chick weights we use

```
s <- chickwts %>%  summarize(average = mean(weight), standard_deviation = sd(weight))
s
```

```
##    average standard_deviation
## 1 261.3099            78.0737
```

Note that for this example we ignore the diet group.

Analysis by group

In this section we focus on an analysis in which the analysis is done across a level of a factor in the data frame. For example, the diet group in the chick data frame etc.

The heights data

The mean and standard deviation for the height by gender can be calculate by adding the function group_by(sex)

```
heights %>%
  group_by(sex) %>%
  summarize(average = mean(height), standard_deviation = sd(height))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 3
##   sex    average standard_deviation
##   <fct>    <dbl>              <dbl>
## 1 Female    64.9               3.76
## 2 Male      69.3               3.61
```

The same results can be obtained using the function tapply .

```
tapply(heights$height,heights$sex,mean)
```

```
##   Female     Male
## 64.93942 69.31475
```

```
tapply(heights$height,heights$sex,sd)
```

```
##   Female     Male
## 3.760656 3.611024
```

The murders data

The median murder rate by region using the group_by(region) and the summarize () functions

```
murders %>% group_by(region) %>%
  summarize(median_rate = median(rate))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 4 x 2
##   region        median_rate
##   <fct>               <dbl>
## 1 Northeast            1.80
## 2 South                3.40
## 3 North Central        1.97
## 4 West                 1.29
```

The median murder rate by region using the tapply() function.

```
tapply(murders$rate,murders$region,median)
```

```
##     Northeast         South North Central          West
##      1.802179      3.398069      1.971105      1.292453
```

The chicks data

Summry statistics by diet group

```
chickwts %>%
  group_by(feed) %>%
  summarize(average = mean(weight), standard_deviation = sd(weight))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 6 x 3
##   feed      average standard_deviation
##   <fct>       <dbl>              <dbl>
## 1 casein       324.               64.4
## 2 horsebean    160.               38.6
## 3 linseed      219.               52.2
## 4 meatmeal     277.               64.9
## 5 soybean      246.               54.1
## 6 sunflower    329.               48.8
```

The striptplot in Figure~@ref(fig:figh) reveals that the cash offers in the middle age group are higher than the cash offers in the young and elderly age groups.

```
stripplot(feed ~ jitter(weight),
            data = chickwts,
            aspect = 1, jitter = T,
            xlab="Chicks weight", col = 1)
```

Sorting

The murders data

We can sort the data frame by a variable x using the function arrange(x) . For the murder data frame, we sort the data by the population size

```
murders %>%
  arrange(population) %>% head()
```

```
##                   state abb        region population total        rate
## 1               Wyoming  WY          West     563626     5   0.8871131
```

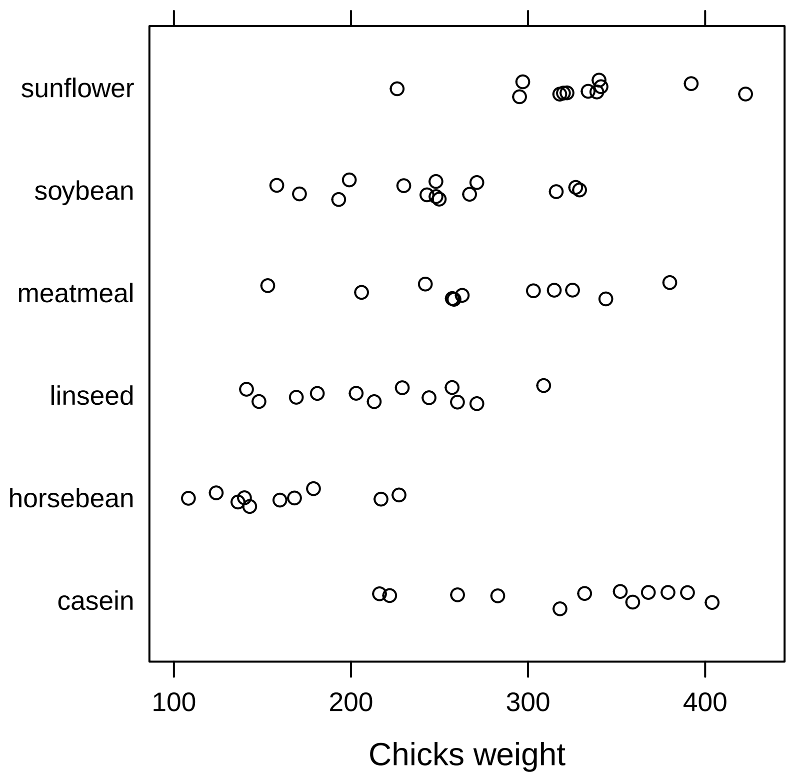

Figure 7: by diet group.

```
## 2 District of Columbia  DC         South     601723    99 16.4527532
## 3              Vermont  VT     Northeast     625741     2  0.3196211
## 4         North Dakota  ND North Central     672591     4  0.5947151
## 5               Alaska  AK          West     710231    19  2.6751860
## 6         South Dakota  SD North Central     814180     8  0.9825837
```

The same sorting can be implemented using the function order () . In this case the rows will be presented in the order of the population.

```
data1<-murders[order(murders$population),]
head(data1)
```

```
##                   state abb        region population total       rate
## 51              Wyoming  WY          West     563626     5  0.8871131
## 9  District of Columbia  DC         South     601723    99 16.4527532
## 46              Vermont  VT     Northeast     625741     2  0.3196211
## 35         North Dakota  ND North Central     672591     4  0.5947151
## 2                Alaska  AK          West     710231    19  2.6751860
## 42         South Dakota  SD North Central     814180     8  0.9825837
```

We sort the data frame by rate from the lowest to the highest rate

```
murders %>%
  arrange(rate) %>%
  head()
```

```
##           state abb        region population total      rate
## 1       Vermont  VT     Northeast     625741     2 0.3196211
## 2 New Hampshire  NH     Northeast    1316470     5 0.3798036
## 3        Hawaii  HI          West    1360301     7 0.5145920
## 4  North Dakota  ND North Central     672591     4 0.5947151
## 5          Iowa  IA North Central    3046355    21 0.6893484
## 6         Idaho  ID          West    1567582    12 0.7655102
```

We can change the order using the function desc() so the data are presented from the highest to the lowest rate in a decreasing order.

```
murders %>%
  arrange(desc(rate)) %>%
  head
```

```
##                  state abb        region population total      rate
## 1 District of Columbia  DC         South     601723    99 16.452753
## 2            Louisiana  LA         South    4533372   351  7.742581
## 3             Missouri  MO North Central    5988927   321  5.359892
## 4             Maryland  MD         South    5773552   293  5.074866
## 5       South Carolina  SC         South    4625364   207  4.475323
## 6             Delaware  DE         South     897934    38  4.231937
```

The Chicken Weight data

The first 6 observations in the chicken weight data belongs to the first chick at time point 0 to 10.

```
head(ChickWeight)
```

```
## Grouped Data: weight ~ Time | Chick
##   weight Time Chick Diet
## 1     42    0     1    1
```

```
## 2     51    2     1     1
## 3     59    4     1     1
## 4     64    6     1     1
## 5     76    8     1     1
## 6     93   10     1     1
```

We sort the data frame according to the Tim variable. After sorting, the first 6 lines in the data frame are the measurements for chock 1-6 at day 21.

```
ChickWeight %>%
  arrange(desc(Time)) %>%
  head
```

```
## Grouped Data: weight ~ Time | Chick
##   weight Time Chick Diet
## 1    205   21     1    1
## 2    215   21     2    1
## 3    202   21     3    1
## 4    157   21     4    1
## 5    223   21     5    1
## 6    157   21     6    1
```

We can reverse the order, in this case the first 6 lines are the measurements for check 1-6 at baseline (Time =0).

```
ChickWeight %>%
  arrange(Time) %>%
  head
```

```
## Grouped Data: weight ~ Time | Chick
##   weight Time Chick Diet
## 1     42    0     1    1
## 2     40    0     2    1
## 3     43    0     3    1
## 4     42    0     4    1
## 5     41    0     5    1
## 6     41    0     6    1
```

Nested sorting

The murders data

Suppose that we want to present the data in an increasing order of x across a level of a factor y . We can sort the data frame by a variable x within the factor levels using the function arrange(y,x) . For the murder data frame, we sort the data by murder rate within the region

```
murders %>%
  arrange(region, rate) %>%
  head()
```

```
##           state abb    region population total      rate
## 1       Vermont  VT Northeast     625741     2 0.3196211
## 2 New Hampshire  NH Northeast    1316470     5 0.3798036
## 3         Maine  ME Northeast    1328361    11 0.8280881
## 4  Rhode Island  RI Northeast    1052567    16 1.5200933
## 5 Massachusetts  MA Northeast    6547629   118 1.8021791
## 6      New York  NY Northeast   19378102   517 2.6679599
```

The cars data

Figure~@ref(fig:figi) shows that mpg as the number of cylinders decreases.

```
stripplot(cyl ~ jitter(mpg),
            data = mtcars,
            aspect = 1, jitter = T,
            xlab="MPG", col = 1)
```

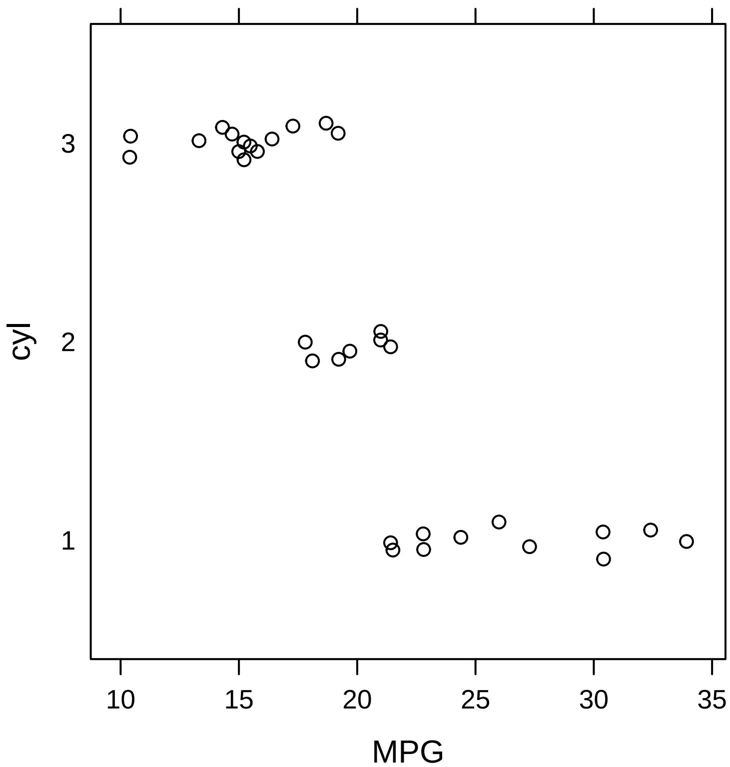

Figure 8: MPG by number of cylinders.

We can sort the cars according to their mpg (in an increasing order)by the number of cylinders

```
mtcars %>%
  arrange(cyl, mpg)
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
```

```
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
## 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
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
## 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
## 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
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    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
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
```

Top n

The murders data

To print the top n observations according to the variable x we can use the function top_n(n, x) . For the murders data, we print the top 5 states with the highest murder rate

```
murders %>% top_n(5, rate)
```

```
##                  state abb        region population total      rate
## 1 District of Columbia  DC         South     601723    99 16.452753
## 2            Louisiana  LA         South    4533372   351  7.742581
## 3             Maryland  MD         South    5773552   293  5.074866
## 4             Missouri  MO North Central    5988927   321  5.359892
## 5       South Carolina  SC         South    4625364   207  4.475323
```

The cars data

Figure~@ref(fig:figj) shows the scaterplot of the cars' weight versus the cars' mpg.

```
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point()
```

The top 4 cars, with the highest mpg are given below

```
mtcars %>% top_n(4,mpg)
```

```
##                 mpg cyl disp  hp drat    wt  qsec vs am gear carb
## 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
## Lotus Europa   30.4   4 95.1 113 3.77 1.513 16.90  1  1    5    2
```

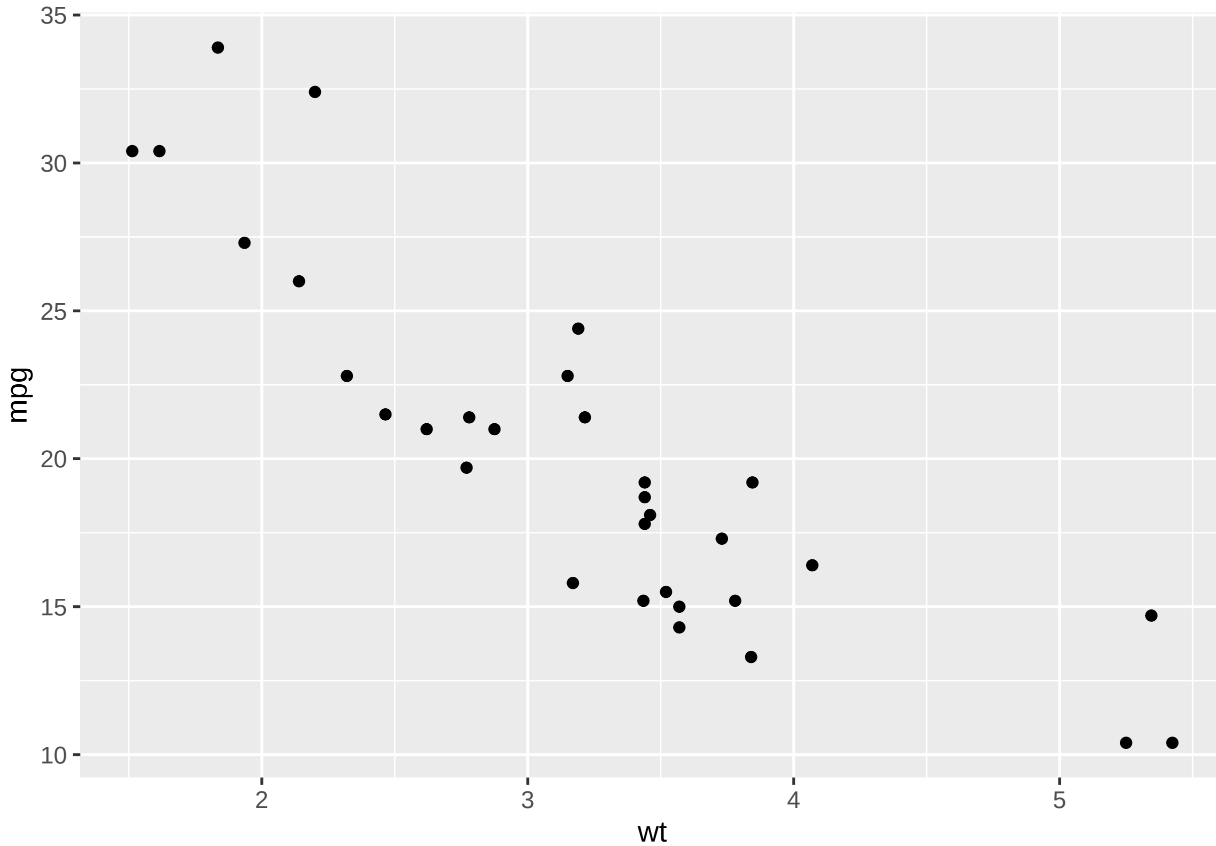

Figure 9: MPG by number of cylinders.