R

Alejandra E. Medina Rivera
Licenciatura en Ciencias Genómicas.
Centro de Ciencias Genómicas, UNAM

Cuernavaca, Mexico
Feb, 2010

Descriptive statistics

1. Summary statistics for a single group
2. Graphical display of distributions
3. Summary statistics by groups
4. Graphics for grouped data
5. Tables
6. Graphical display of tables
7. Correlation
8. Combinatorics
9. ROC curves

Descriptive statistics

Descriptive statistics describe the main features of a collection of data quantitatively. Descriptive statistics are distinguished from inferential statistics (or inductive statistics), in that descriptive statistics aim to summarize a data set quantitatively without employing a probabilistic formulation, rather than use the data to make inferences about the population that the data are thought to represent.

Summary statistics for a single group

It is easy to calculate simple summary statistics with R.

```
> x <- rnorm(50)
> mean(x)

[1] -0.09122392

> sd(x)

[1] 1.083760

> var(x)

[1] 1.174535

> median(x)

[1] -0.1730651
```

Summary statistics for a single group

Empirical quantiles may be obtained with the function `quantile`

```
> quantile(x)
```

```
        0%        25%        50%        75%
-2.5543238 -0.7904138 -0.1730651  0.6154567
      100%
 2.5729806
```

What do the quantiles mean?
It is also possible to obtain other quantiles, this is done by adding an argument containing the desire percentage points.

```
> pvec <- seq(0, 1, 0.1)
> quantile(x, pvec)
```

Summary statistics for a single group

```
        0%        10%        20%        30%
-2.5543238 -1.4780513 -1.0762851 -0.6074869
       40%        50%        60%        70%
-0.4221129 -0.1730651  0.1112403  0.4620028
       80%        90%       100%
 0.9145444  1.1044188  2.5729806
```

Summary statistics for a single group

But as you already know there is a function that will calculate most of this summary statistics.

```
> library(ISwR)
> data(juul)
> summary(juul)

      age             menarche
 Min.   : 0.170   Min.   :  1.000
 1st Qu.: 9.053   1st Qu.:  1.000
 Median :12.560   Median :  1.000
 Mean   :15.095   Mean   :  1.476
 3rd Qu.:16.855   3rd Qu.:  2.000
 Max.   :83.000   Max.   :  2.000
 NA's   : 5.000  NA's   :635.000
      sex              igf1
 Min.   :1.000   Min.   : 25.0
 1st Qu.:1.000   1st Qu.:202.2
 Median :2.000   Median :313.5
 Mean   :1.534   Mean   :340.2
 3rd Qu.:2.000   3rd Qu.:462.8
 Max.   :2.000   Max.   :915.0
```

Summary statistics for a single group

```
NA's   :5.000   NA's   :321.0
    tanner           testvol
Min.   :  1.000   Min.   :  1.000
1st Qu.:  1.000   1st Qu.:  1.000
Median :  2.000   Median :  3.000
Mean   :  2.640   Mean   :  7.896
3rd Qu.:  5.000   3rd Qu.: 15.000
Max.   :  5.000   Max.   : 30.000
NA's   :240.000   NA's   :859.000
```

Although as you can see, this has a big mistake, since all the variables where interpreted as quantitive, some where qualitative.

Summary statistics for a single group

```
> juul$sex <- factor(juul$sex, labels = c("M",
+     "F"))
> juul$menarche <- factor(juul$menarche,
+     labels = c("No", "Yes"))
> juul$tanner <- factor(juul$tanner,
+     labels = c("I", "II", "III", "IV",
+         "V"))
> summary(juul)
```

```
      age          menarche      sex
 Min.   : 0.170   No  :369    M   :621
 1st Qu.: 9.053   Yes :335    F   :713
 Median :12.560   NA's:635    NA's:  5
 Mean   :15.095
 3rd Qu.:16.855
```

Summary statistics for a single group

```
Max.   :83.000
NA's   : 5.000
     igf1        tanner
Min.   : 25.0   I   :515
1st Qu.:202.2   II  :103
Median :313.5   III : 72
Mean   :340.2   IV  : 81
3rd Qu.:462.8   V   :328
Max.   :915.0   NA's:240
NA's   :321.0
   testvol
Min.   :  1.000
1st Qu.:  1.000
Median :  3.000
Mean   :  7.896
```

Summary statistics for a single group

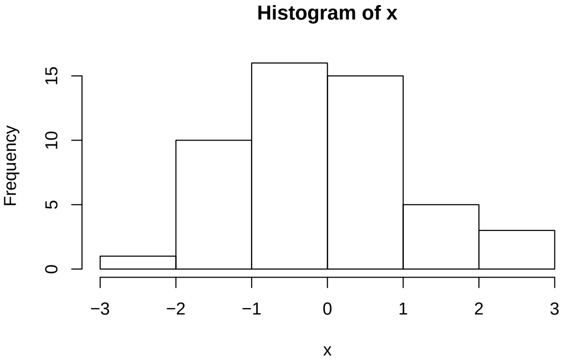
```
3rd Qu.: 15.000
Max.   : 30.000
NA's   :859.000
```

Histograms

You can get a reasonable impression of the shape of a distribution by drawing a histogram, this is, a count of how many observations fall with specified divisions (”bins”) if the *x*-axis

```
> hist(x)
```

Histograms



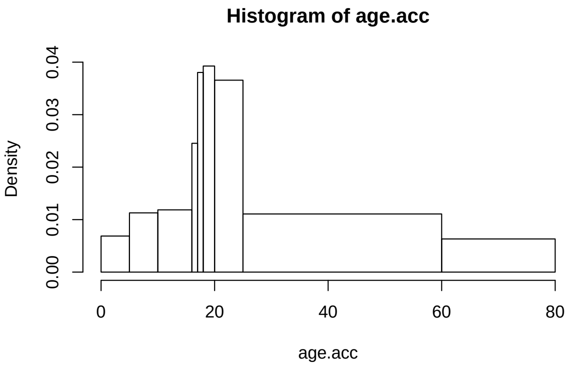
By specifying `breaks`$= n$, you get approximately n bars in the histogram since the algorithm tries to create pretty cut points.

Histograms

Althought you can have full control of the position of the breaks if you specify a vector rather than a number.

```
> mid.age <- c(2.5, 7.5, 13, 16.5, 17.5,
+      19, 22.5, 44.5, 70.5)
> acc.count <- c(28, 46, 58, 20, 31,
+      64, 149, 316, 103)
> age.acc <- rep(mid.age, acc.count)
> brk <- c(0, 5, 10, 16, 17, 18, 20,
+      25, 60, 80)
> hist(age.acc, breaks = brk)
```

Histograms



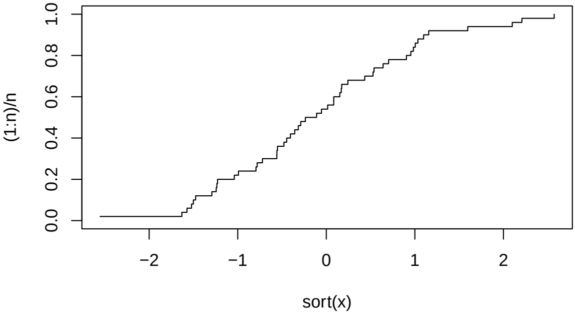
Which is the main difference between the histogram with *n* breakpoints and the one where we selected specific breaks? Why this is important?

Empirical cumulative distribution

The empirical cumulative distribution function is defined as the fraction of data smaller than or equal to x.

```
> n <- length(x)
> plot(sort(x), (1:n)/n, type = "s",
+      ylim = c(0, 1))
```

Empirical cumulative distribution

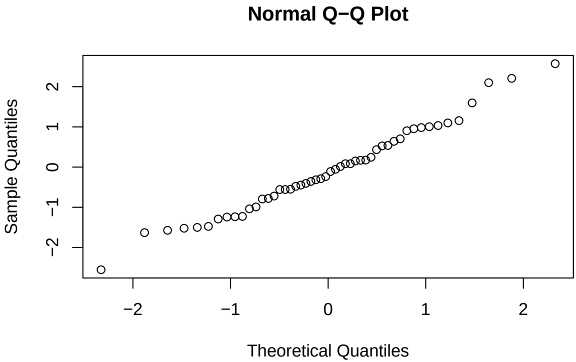


Q-Q plots

- One propose of calculating the empirical cumulative distribution function is to see whether data ca be assumed normally distributed.
- For a better assessment, you might plot the k'th smallest observation against the expected value of the k'th smallest observation out of n in a standard normal distribution.
- The point is that in this way you would expect to obtain a straight line if the data come from a normal distribution.
- We already know how to compare two data sets using qq plots, but `R`, has functions to compare data with distributions

```
> qqnorm(x)
```

Q-Q plots


Normal Q−Q Plot
Sample Quantiles
−2
−1
0
1
2
Theoretical Quantiles
−2
−1
0
1
2

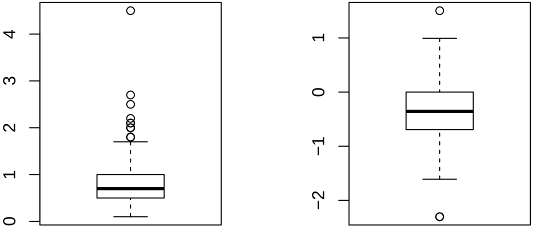
Boxplots

A boxplot, or more descriptively a “box-and-whiskers”plot. is a graphical summary of a distribution

- The box in the middle indicates “hinges.ªnd medina.
- The lines (“whiskers”) show the largest/smallest observation that falls within a distance of 1.5 times the box size from the nearest hinge.
- If any observation fall farther away, the additional points are considered “extreme”values and are shown separately

```
> data(IgM)
> par(mfrow = c(1, 2))
> boxplot(IgM)
> boxplot(log(IgM))
> par(mfrow = c(1, 1))
```

Boxplots



Summary statistics by groups

When dealing with grouped data, you will often want to have various summary statistics computed within groups.

```
> data(red.cell.folate)
> tapply(red.cell.folate$folate, red.cell.folate$vent
+      mean)

N2O+O2,24h  N2O+O2,op      O2,24h
  316.6250   256.4444    278.0000

> tapply(red.cell.folate$folate, red.cell.folate$vent
+      sd)

N2O+O2,24h  N2O+O2,op      O2,24h
  58.71709   37.12180    33.75648

> tapply(red.cell.folate$folate, red.cell.folate$vent
+      length)
```

Summary statistics by groups

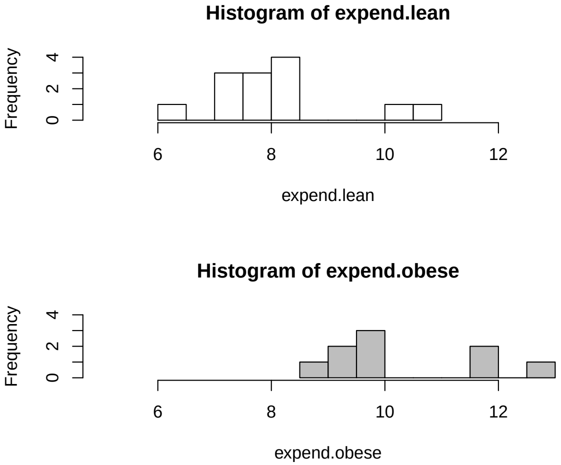
```
N2O+O2,24h  N2O+O2,op     O2,24h
         8          9          5
```

Histograms

In dealing with grouped data it is important to be able not only to create plots for each group but also to be able to compare the plots between groups.

```
> data(energy)
> expend.lean <- energy$expend[energy$stature ==
+     "lean"]
> expend.obese <- energy$expend[energy$stature ==
+     "obese"]
> par(mfrow = c(2, 1))
> hist(expend.lean, breaks = 10, xlim = c(5,
+     13), ylim = c(0, 4), col = "white")
> hist(expend.obese, breaks = 10, xlim = c(5,
+     13), ylim = c(0, 4), col = "grey")
> par(mfrow = c(1, 1))
```

Histograms

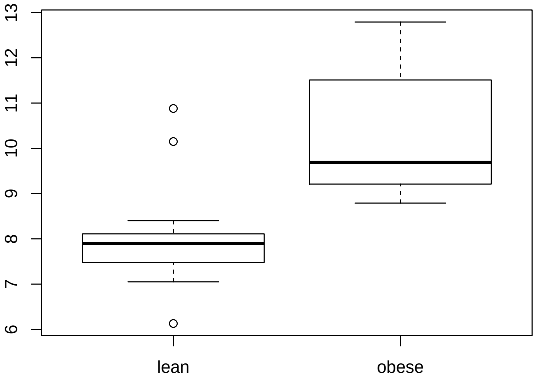

Histogram of expend.lean
Frequency
0
2
4
6
8
10
12
expend.lean
Histogram of expend.obese
Frequency
0
2
4
6
8
10
12
expend.obese

Parallel boxplots

You might want a set of boxplots from several groups in the same frame. `boxplot` can handle this, both when data are given in the form of separate vectors from each group and when data are in one long vector and a is classified with a factor.

```
> boxplot(energy$expend ~ energy$stature)
```

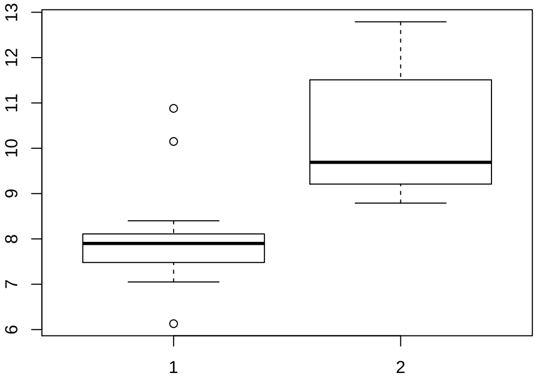
Parallel boxplots



Parallel boxplots

```
> boxplot(expend.lean, expend.obese)
```

Parallel boxplots

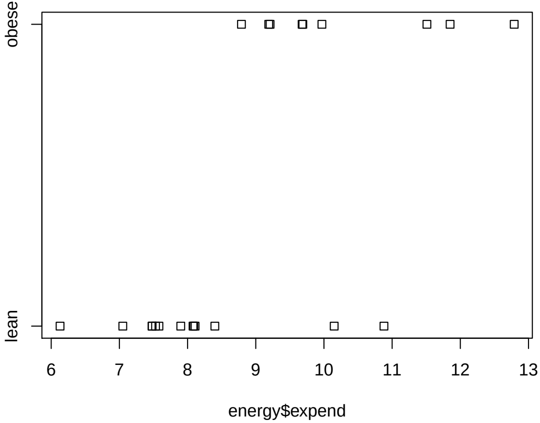


Stripcharts

On the pervious boxplot you can see that since the interquartile range is quiet a bit larger in one group than in the other one of the boxplots looks fatter.
For small data set it is recommended to plot the raw data on a dot diagram.

```
> stripchart(energy$expend ~ energy$stature)
```

Stripcharts



Generating Tables

The common case is that you have a data-frame with diferent variables, in this case you can obtain a table out from the data using the commands `table()`, `xtable()` and `ftable()`. The `table()` function is the basic one.

```
> table(juul$sex)

  M   F
621 713

> table(juul$sex, juul$menarche)

     No Yes
  M   0   0
  F 369 335

> table(juul$menarche, juul$tanner)
```

Generating Tables

```
        I  II III  IV   V
  No  221  43  32  14   2
  Yes   1   1   5  26 202

> table(juul$menarche, juul$tanner,
+       juul$sex)

, ,  = M

        I  II III  IV   V
  No    0   0   0   0   0
  Yes   0   0   0   0   0

, ,  = F
```

Generating Tables

```
        I   II III  IV   V
No    221   43  32  14   2
Yes     1    1   5  26 202
```

Marginal Tables and relative frequency

The common case is that you have a data-frame with diferent variables, in this case you can obtain a table out from the data using the commands `table()`, `xtable()` and `ftable()`. The `table()` function is the basic one.

```
> tanner.sex <- table(juul$tanner, juul$sex)
> margin.table(tanner.sex, 1)

  I   II III  IV   V
515 103  72  81 328

> margin.table(tanner.sex, 2)

  M   F
545 554
```

Relative frequencies in a table are generally expressed as proportions of the row or column totals.

Marginal Tables and relative frequency

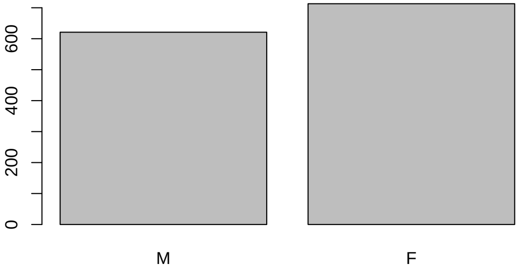
```
> prop.table(tanner.sex, 1)

            M         F
  I   0.5650485 0.4349515
  II  0.5339806 0.4660194
  III 0.4722222 0.5277778
  IV  0.5061728 0.4938272
  V   0.3780488 0.6219512
```

Bar plots

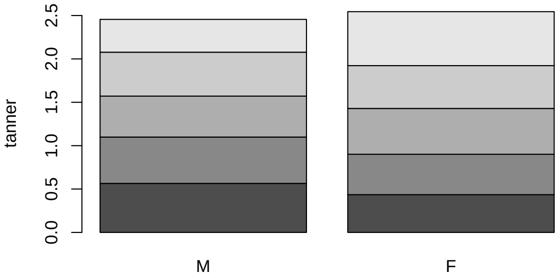
Tables can be the input of the barplot() function we already know.

```
> barplot(table(juul$sex))
```



Bar plots

```
> barplot(prop.table(tanner.sex, 1),
+       ylab = "tanner")
```

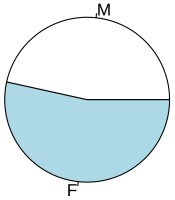


Pie charts

- Pie charts are traditionally scored upon statistics because they are often used to make trivial data look impressive and are dificult to decode for the human mind.
- They very rarely contain information that couldn't better be displayed as a bar plot.
- Even thought R can draw pretty pie charts.

```
> pie(table(juul$sex))
```

Pie charts



Person correlation

- In R to obtain the pearson soeficient correaltion between two variables is easy.

```
> data(thuesen)
> cor(thuesen$blood.glucose, thuesen$short.velocity,
+     use = "complete.obs")
[1] 0.4167546
```

Arrangements

an arrangement is a list of elements in a specific order

- Sampling with replacement Imagine we want to sample the 4 nucleotides for creating an oligonucleotide of length 7, so we can get any of the 4 nucleotides at each position

```
> n <- 4
> n * n
[1] 16
> n^7
[1] 16384
```

So we can get 16384 diferent oligonucleotides of length 7

Arrangements

- Permutation of the elements of a set: the factorial Imagine we want to have all the oligonucleotides of size 4 that contain all the four nucleotides: ATCG, TACG,ACGT, etc ...

  Intuitively, the generating process is quite simple: we will enumerate all the possible ways to rank the x elements of a set ($x = 4$). For this, we will first select a single element in the set, and place it on the top of the ordered list. For this first step, there are x possible choices (each letter of the considered alphabet). As soon as the first element has been drawn, it is excluded from the set (since we want no more than one occurrence of each nucleotide).

```
> n * (n - 1) * (n - 2) * (n - 3)
[1] 24
```

Arrangements

```
> factorial(n)
[1] 24
```

- Ordered selections without replacement The 6,000 genes of a genome were sorted according to their level of expression, as measured with a microarray. The 15 top genes were selected. How many possible selections are there, if we consider that the order of the selection matters?
  In a set of size n, there are n possible choices for the first element, n?1 choices left for the second element, . . . , and n?14 choices for the 15th element. Thus, for a selection of $x = 15$ elements among $n = 6000$ gened, the number of possibilities is $N =$6000 $\cdot$ 5999 $\cdot$ 5998 $\cdot$ . . . $\cdot$(6000 $-$ 14)= 4,62E56.

Arrangements

$$A_n^x = \frac{n!}{(n-x)!} \tag{1}$$

Ordereless selections without replacement (combinations) The 6,000 genes of a genome were sorted according to their level of expression, measured with an oligonucleotide microarray. The 15 top genes were selected. How many distinct sets would be possible, if one does not take into account the order of the selection?

$$C_n^x = \frac{n!}{x!(n-x)!} \tag{2}$$

```
> choose(6000, 15)
```

```
[1] 3.533156e+44
```

ROC curves

- The real-valued output of scoring classifiers is turned into a binary class decision by choosing a cutoff.
- As no cutoff is optimal according to all possible performance criteria, cutoff choice involves a trade-off among different measures.
- Typically a trade-off between a pair of criteria (eg. sensitivity versus specificity) is visualized as a cutoff-parametrizied curve.
- Receiver operating characteristic (ROC) curves are one of the most popular graphs
- A variety of libraries are available for these tasks: ROCR, ROC, nonbinROC. Remember you can always create your own fucntions. if needed.
- We are going to explore only the `ROCR` package.

ROCR

- First we have to install and load the package

```
> install.packages("ROCR")
> library(ROCR)
```

- The data for today comes from a 10-fold cross-validation set of predictions and corresponding class labels from a study on predicting HIV coreceptor usage from the sequence of the viral envelope protein.

```
gdata: read.xls support for 'XLS'
gdata: (Excel 97-2004) files ENABLED.

gdata: Unable to load perl libaries
gdata: needed by read.xls()
gdata: to support 'XLSX' (Excel 2007+)
gdata: files.
```

ROCR

```
gdata: Run the function
gdata: 'installXLSXsupport()'
gdata: to automatically download and
gdata: install the perl
gdata: libaries needed to support Excel
gdata: XLS and XLSX formats.
```

```
> data(ROCR.hiv)
```

- Then we are going to create a prediction data structure, so for one experiment we will take our values for the predictions and the labels of classification.

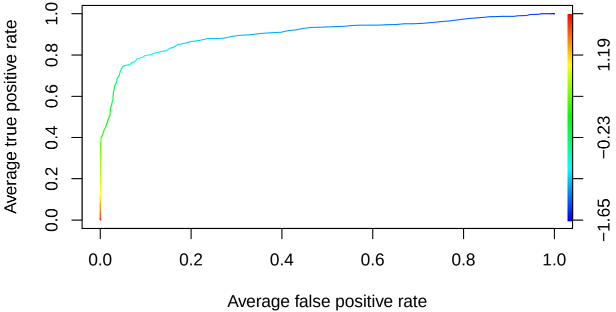
```
> pred <- prediction(ROCR.hiv$hiv.svm$predictions
+      ROCR.hiv$hiv.svm$labels)
```

ROCR

- What we actually want now is to measure the performance of the method of classification,

```
> perf <- performance(pred, "tpr", "fpr")
> plot(perf, avg = "threshold", colorize = TRUE)
```

ROCR


Average true positive rate
0.0
0.2
0.4
0.6
0.8
1.0
Average false positive rate
0.0
0.2
0.4
0.6
0.8
1.0
1.19
−0.23
−1.65