MEASURES OF SPREAD AND CORRELATION

Applied math for data science (DSC 482/MTH 445) Fall 2022

Table of Contents

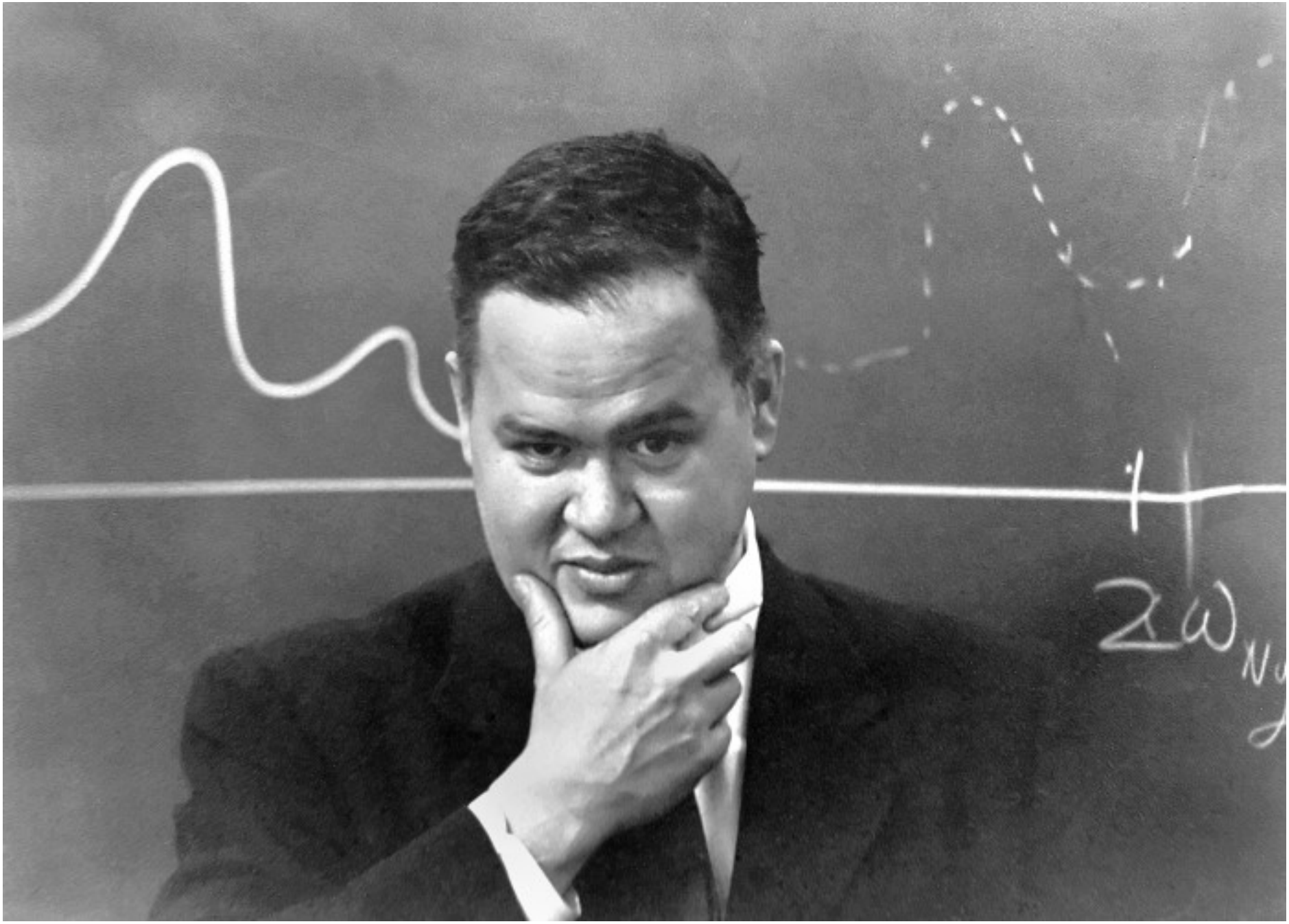

2ω

Figure 1: John Wilder Tukey (1915-2000)

1. [X] Quantiles, percentiles, and the five-number-summary
2. [X] Measures of spread: variance, standard deviation and IQR
3. [ ] Covariance and correlation
4. [ ] Outliers

1 Prequel: why distributions?

- Statistics are derived from experimental observations, which form values of variables stored in column vectors of data frames
- As soon as you observe 1 value, you have a *distribution*. The more values, the more interesting the distribution (for classification)

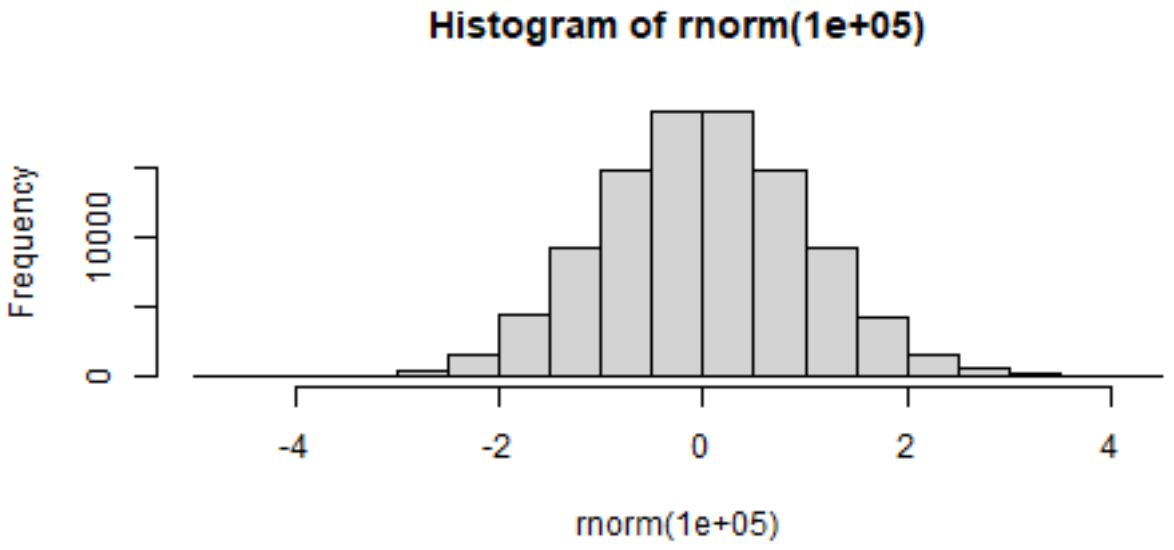
- Example: a "normal" distribution (also Gaussian or Bell curve)

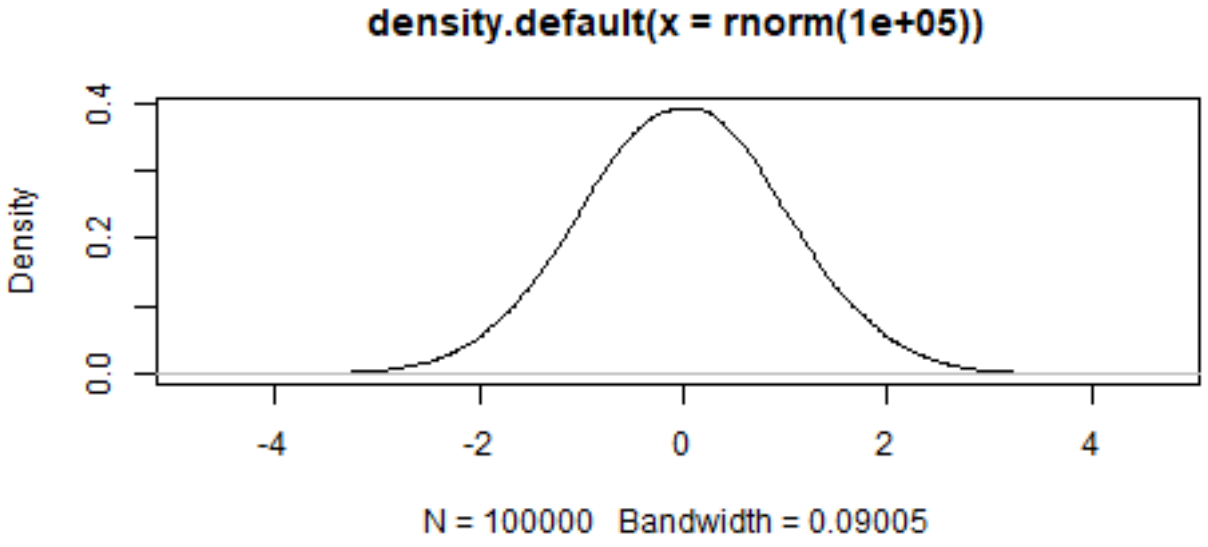
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Figure 2: Bell curve with sample mean and standard deviation

- This can be reproduced in R with pseudo-random numbers from `rnorm`.

```
par(mfrow=c(2,1))
hist(rnorm(1e+05))
plot(density(rnorm(1e+05)))
```


Histogram of rnorm(1e+05)
Frequency
0
10000
-4
-2
0
2
4
rnorm(1e+05)

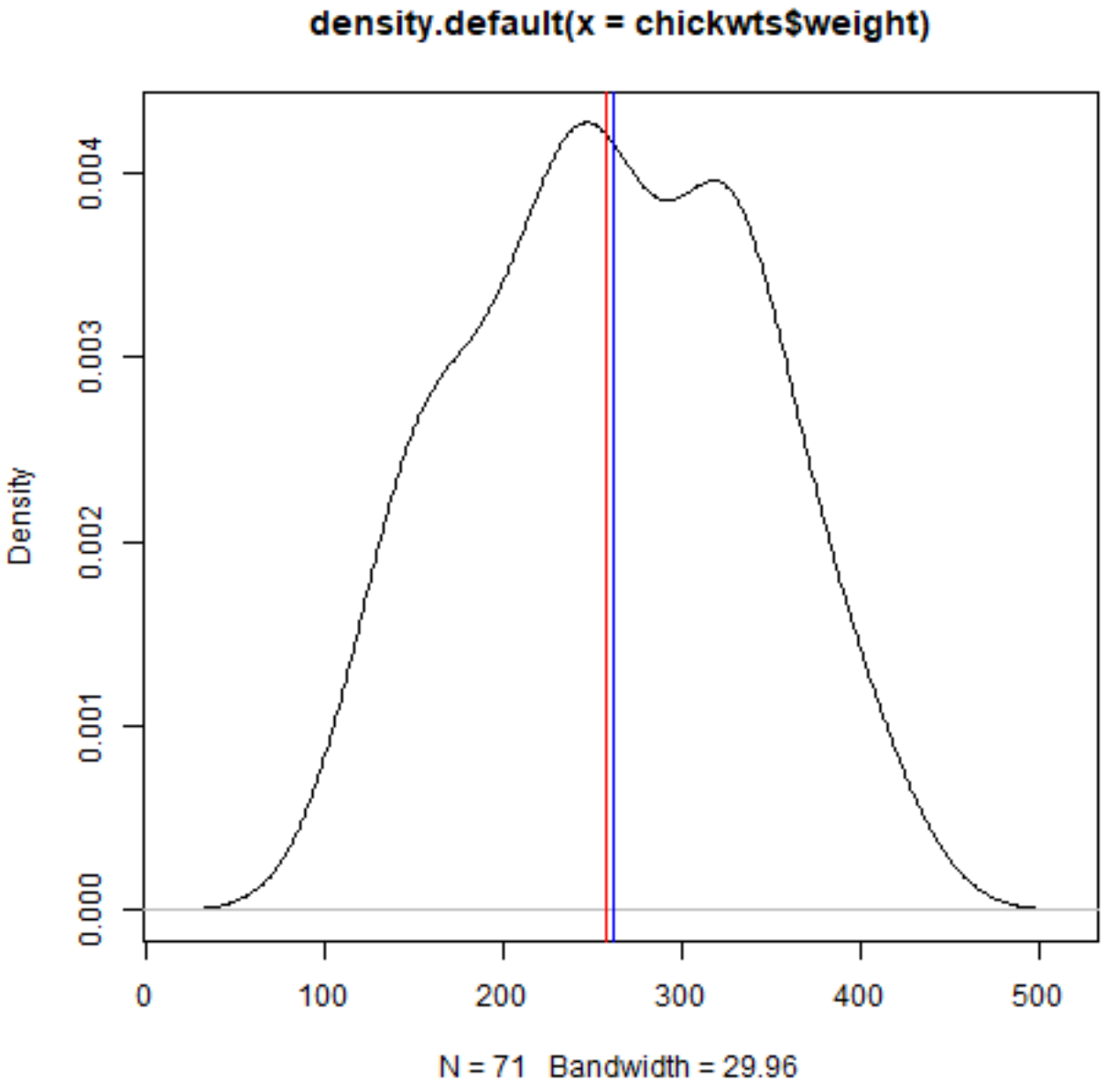

density.default(x = rnorm(1e+05))
Density
0.0
0.2
0.4
-4
-2
0
2
4
N = 100000 Bandwidth = 0.09005

The distribution as a pattern exhibits measures of centrality (like μ), and measures of spread (like σ).

2 Quantiles, percentiles and five-number-summary

- A *quantile* is a value that characterizes a distribution: it characterizes the rank of an observation
- The *median* or middle magnitude is the 0.5th quantile: half of all measurements lie below (and above) it

```
plot(density(chickwts$weight))
abline(v=median(chickwts$weight), col="red")
abline(v=mean(chickwts$weight), col="blue")
```


density.default(x = chickwts$weight)
Density
0.000
0.001
0.002
0.003
0.004
0
100
200
300
400
500
N = 71 Bandwidth = 29.96

- Quantiles can be expressed as *percentiles*, on a scale of [0,100] instead of [0,1]: the p-th quantile is the 100*p-th percentile
- All algorithms to identify quantiles/percentiles sort observations, and then compute a weighted average to find p numerically

3 Quantiles by hand

Do it by hand! What is the 0th / 0.25th / 0.5th / .75th and 1th quantile of `foo <- c(100, 498.5, 22, 0, -33, 100.66, -2)`

```
foo <- c(100, 498.5, 22, 0, -33, 100.66, -2)
foo
```

1. Which number is smaller than any other (0th quantile/percentile - 0 numbers lie below it). **Minimum of the data**.

   ```
   min(foo)
   ```

2. The 25th percentile splits off the lowest 25% from the highest 75%. This number is the middle number between the smallest number (minimum, 0th quantile) and the median (0.5th quantile), or the **median of the lower half of the data**.

   ```
   foo_s <- sort(foo)  # sort vector from smallest to largest
   foo_s
   median(foo_s) # compute median: (-2+1)/2 = -1
   median(c(-33,-2,0,22)) # 0.25th quartile
   ```

3. The 0.5th quartile or 50th percentile: half of the values are smaller than it - **median of the data**.

```
median(foo)
```

4. The 0.75th quartile or 75% percentile: 3/4 of all values are smaller than it. This number is the middle number between the largest number (maximum), or the **median of the upper half of the data**.

```
foo_s <- sort(foo)  # sort vector from smallest to largest
foo_s
median(foo_s) # compute median: (-2+1)/2 = -1
median(c(22,100,100.66,498.50)) # 75th percentile
```

5. The 1.00th quartile or 100% percentile: all values are smaller than it. This number is the **maximum value of the data**.

```
max(foo)
```

Compare the result with the `quantile` algorithm:

```
q <- quantile(foo)
names(q)<-NULL
identical(q,c(-33,-1,22,100.33,498.5))
```

4 Quantiles with `quantile`

- Using our old friend `xdata <- c(2,4.4,3,3,2,2.2,2,4)`, we can use `quantile` to compute the 0th to 4th quantile.

```
xdata <- c(2,4.4,3,3,2,2.2,2,4)
xdata
sort(xdata)
quantile(xdata)
```

- With `quantile`, we can also compute other quantiles, like the 0.8th quantile (or 80th percentile): 80% of all values are smaller than it:

```
quantile(xdata, prob=0.8)
```

- `[ ]`

  Does `quantile` allow removing `NA` values?

```
quantile(c(xdata, NA), prob=0.8, na.rm=TRUE)
```

- `quantile` is a generic function and can take multiple input formats

```
methods(quantile)
```

- `quantile` can also handle probability vectors.

```
quantile(xdata, prob=c(0, .25, 0.75, 1))
```

- `[ ]`

  What happens if you choose `prob > 1`

```
quantile(xdata, prob=1.5)
```

- `quantile` supports **nine** different algorithms. The `help(quantile)` reveals that different statistical programming languages (S, SPSS, SAS) use different algorithms.

5 Quantiles and summary with functions

- `quantile(x,prob=c(0,0.25,0.5,0.75,1)` is the 5-number summary consists of:
  1. the minimum (0th quantile/percentile) or minimum
  2. the 1st/lower quartile (0.25th quantile/25th percentile)
  3. the 2nd quartile or median (0.5th quantile/50th percentile)
  4. the 3rd or upper quartile (0.75th quantile/75th percentile)
  5. the 4th quartile (1st quantile/100th percentile) or maximum

- This summary is also computed by `summary`

```
summary(xdata)
```

6 Practice: `quantile` and `summary`

1. Compute the lower and upper quartiles (25th and 75th percentile or 0.25th and 0.75th quantile) of the weights of the chicks in the built-in `chickwts` data frame.

```
quantile(chickwts$weight, prob=c(0.25,0.75))
```

2. What do these results mean?

   25% of the chicks weights lie at or below 204.5 grams, and 75% of the chick weights lie at or below 323.5 grams.

3. Compute the five-number summary and the sample mean of the magnitude of the seismic events off the coast of Fiji that occurred at a depth of less than 400 km, using the built-in `quakes` data frame.

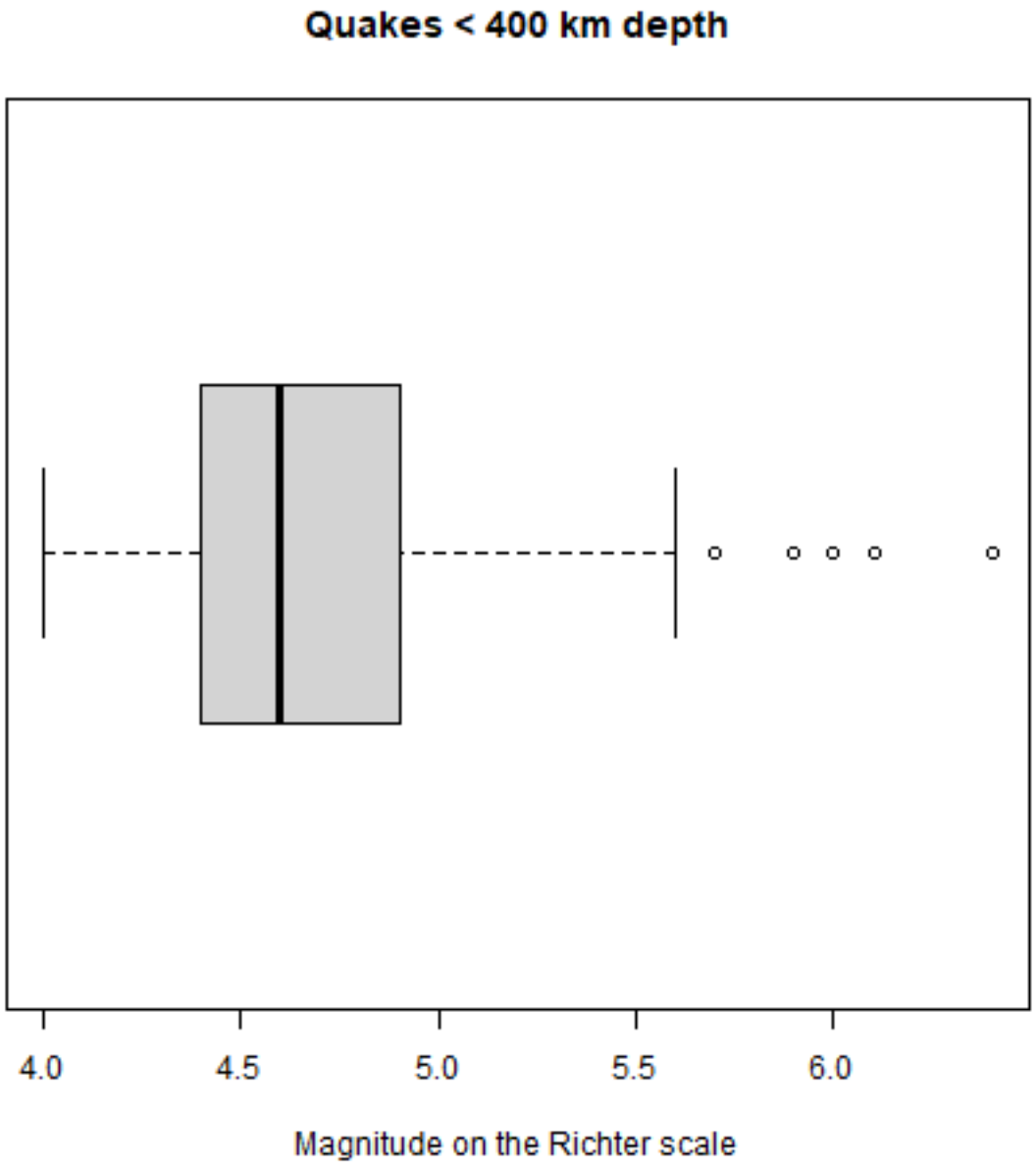
```
summary(quakes$mag[quakes$depth<400])
```

4. What do these results mean?

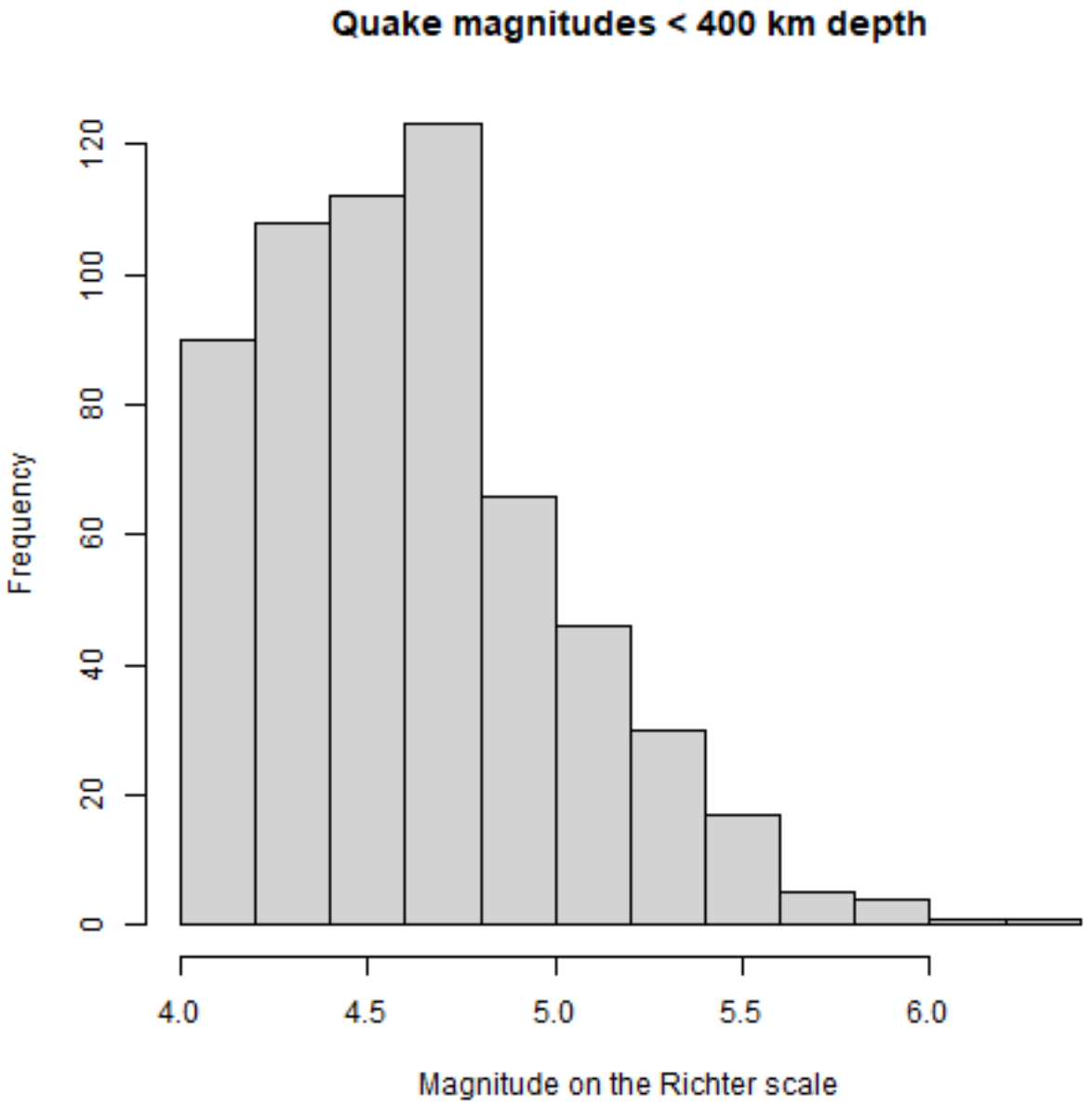
   Most of the quakes below that depth of 400km lie around 4.6 on the Richter scale. The maximum is much further away from the upper quartile than the minimum is from the lower quartile. This suggests that the distribution of quake magnitude vs. depth is skewed. More specifically, it's skewed to the right - i.e. it stretches out more positively from the center to the right. The mean is dragged up by this skewedness.

```
index <- quakes$depth<400
y <- quakes$mag[index]
boxplot(y,
        data=quakes,
        xlab="Magnitude on the Richter scale",
```

```
        main="Quakes < 400 km depth",
        horizontal=TRUE)
```


Quakes < 400 km depth
4.0
4.5
5.0
5.5
6.0
Magnitude on the Richter scale

```
hist(y,
     main="Quakes < 400 km depth",
     xlab="Magnitude on the Richter scale")
```


Quake magnitudes < 400 km depth
Frequency
0
20
40
60
80
100
120
4.0
4.5
5.0
5.5
6.0
Magnitude on the Richter scale

7 Spread: variance, standard deviation and IQR

- Measures of centrality indicate where your observations are *massed*, but they say nothing about the degree of *dispersion* or *spread*
- The measures of spread include: variance, standard deviation, and IQR

8 Example: same centrality, different spread

- Define two vectors of hypothetical observations, `xdata` and `ydata`

```
(xdata <- c(2, 4.4, 3, 3, 2, 2.2, 2, 4))
(ydata <- c(1, 4.4, 1, 3, 2, 2.2, 2, 7))
```

- These vectors have the same arithmetic mean

```
mean(xdata)
mean(ydata)
```

- Let's plot the vectors on top of one another using some base R plotting functions: `plot`, `abline`, `text`, `points`, and `jitter`.
- Remember that, for plots in Org-mode, you need additional header arguments after the `#+begin_src R` - to store a graph in `plot.png`:

  `:results graphics file :file plot.png`

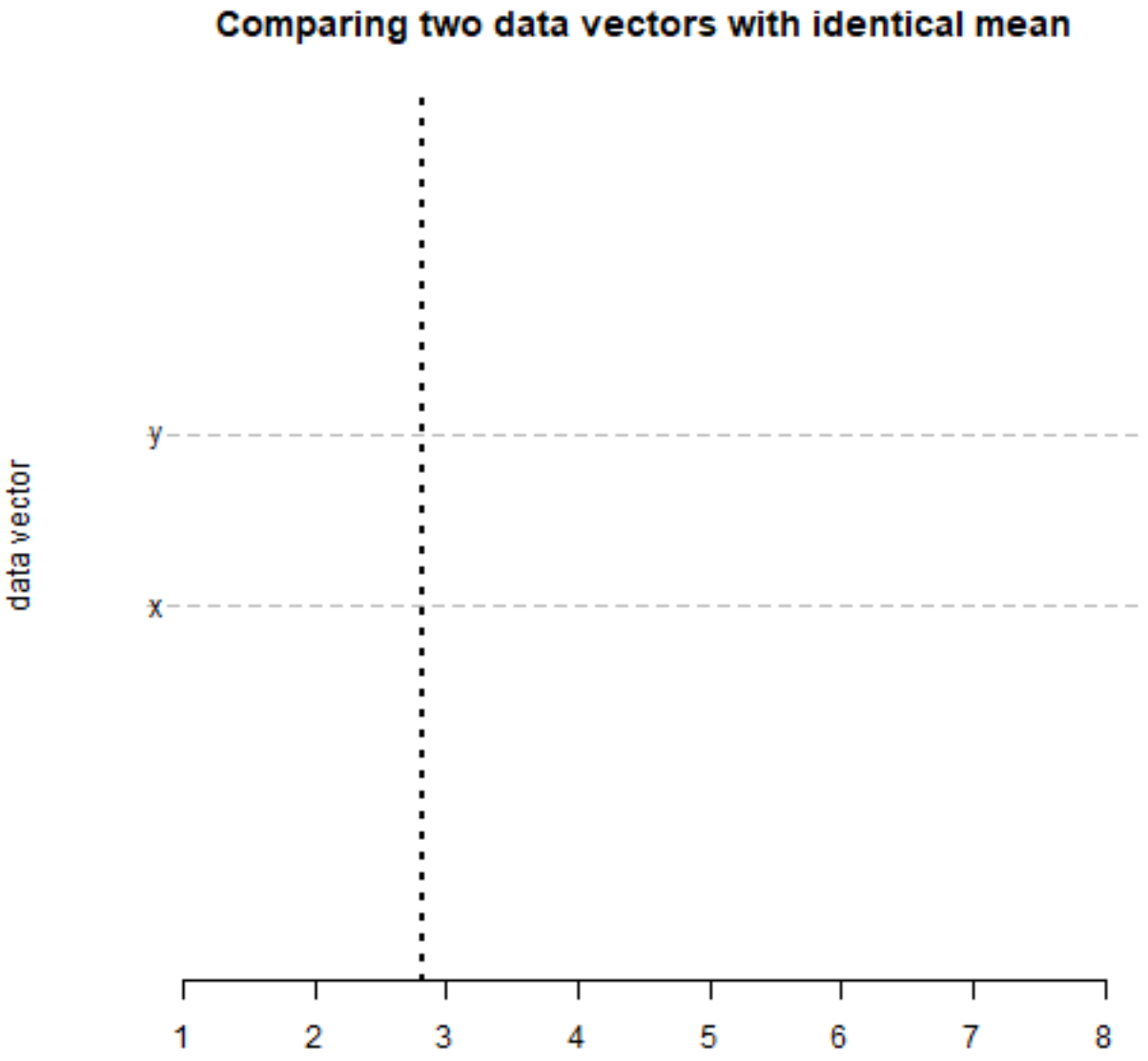
- The first code block only plots some guiding lines and labels

```
plot(x=xdata,    # data to plot - one vector only
     type="n",   # don't plot anything actually (n='nothing')
     xlab="",    # empty x-axis label
     ylab="data vector", # y-axis label
     yaxt="n",   # suppress y-axis
     bty="n",    # remove box around plot
     main="Comparing two data vectors with identical mean"
     )

abline(h=c(3,3.5), # draw horizontal line at y=3 and y=3.5
       lty=2,      # draw a dashed line
       col="gray"  # draw a gray line
       )

abline(v=2.825, # draw vertical line at x=2.825 (the mean)
       lwd=2,   # draw a thick line
       lty=3    # draw a dotted line
       )

text(x=c(0.8,0.8),     # location of text boxes
     y=c(3,3.5),
     labels=c("x","y")
     )
```


Comparing two data vectors with identical mean
data vector
y
x
1
2
3
4
5
6
7
8

- The second code block contains all of the above and plots the points in "jittered" fashion, because some of the vector data are identical:

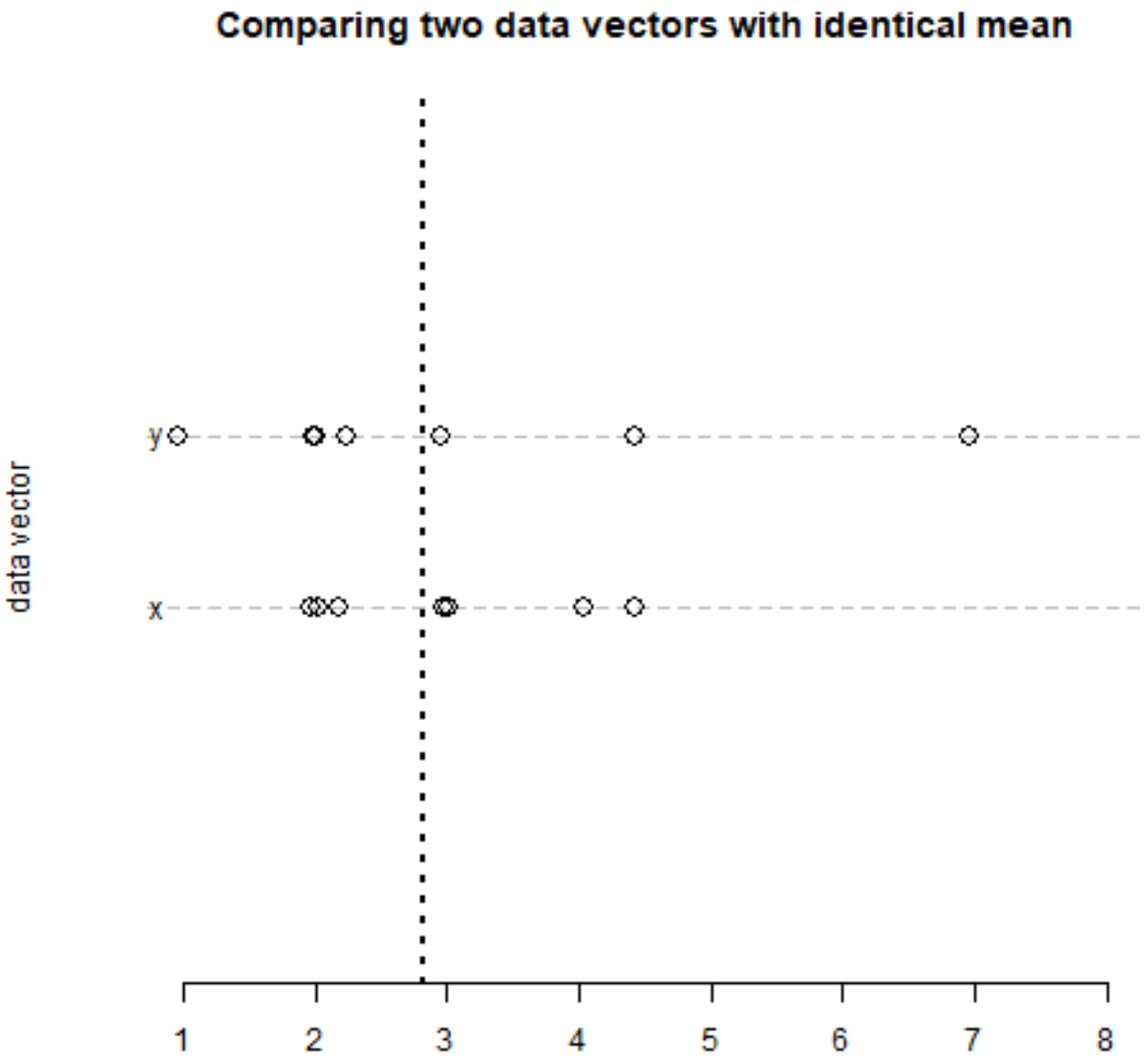
```
plot(x=xdata,    # data to plot - one vector only
     type="n",   # don't plot anything actually (n='nothing')
     xlab="",    # empty x-axis label
     ylab="data vector", # y-axis label
     yaxt="n",   # suppress y-axis
     bty="n",    # remove box around plot
     main="Comparing two data vectors with identical mean"
     )

abline(h=c(3,3.5), # draw horizontal line at y=3 and y=3.5
       lty=2,      # draw a dashed line
       col="gray"  # draw a gray line
       )

abline(v=2.825, # draw vertical line at x=2.825 (the mean)
       lwd=2,   # draw a thick line
       lty=3    # draw a dotted line
       )

text(x=c(0.8,0.8),     # location of text boxes
     y=c(3,3.5),
     labels=c("x","y")
     )

points(                       # draw points
  jitter(c(xdata,ydata)),     # jitter the data points
  c(rep(3, length(xdata)),    # plot xdata over lower line
    rep(3.5, length(ydata))), # plot ydata over upper line
  cex=1.5                     # scale point size by 1.2
)
```


Comparing two data vectors with identical mean
data vector
y
x
1
2
3
4
5
6
7
8

- The observations in `ydata` (upper horizontal line) are more spread out around the measure of centrality (dotted vertical line) than the observations in `xdata`.
- To quantify these differences, you need exact measures of spread like variance, standard deviation, and interquartile range (IQR)

9 Variance

- The *sample variance* measures the degree of the spread of numeric observations around their arithmetic mean via average squared distance

- Shown here for a set of n numeric measurements $x = \{x_1, x_2, \ldots, x_n\}$

$$s_x^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \ldots + (x_n - \bar{x})^2}{n-1} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

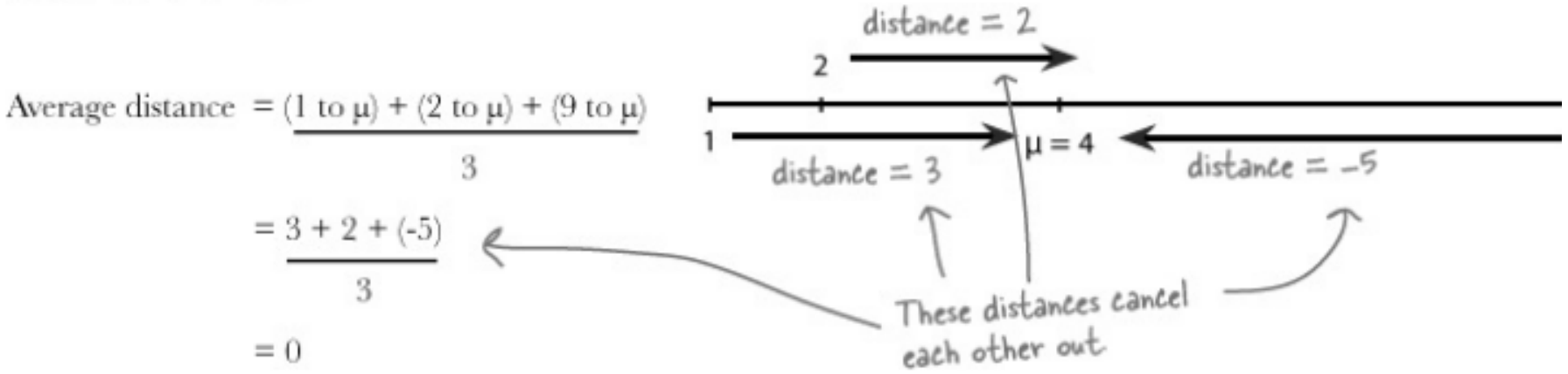
- For example, for `xdata`

$$\frac{(2-2.825)^2+(4.4-2.825)^2+\ldots+(4-2.825)^2}{7}$$
$$=\frac{(-0.825)^2+(1.575)^2+\ldots+(1.175)^2}{7}$$
$$=\frac{6.355}{7}=0.908$$

- Why do we use the square to measure average distance? Because the average distance of values from the mean is always 0.
- The arithmetic/sample average of 1,2,9 is 4.

```
mean(c(1,2,9))
```

- Positive and negative distances cancel each other:



- Why is the denominator for the variance `n-1` and not `n`?

  The use of `n-1` vs. `n` in the denominator of the formula for the variance and, consequentially, for the standard deviation, is the subject of much discussion among statisticians. It is called Bessel correction to correct the bias in the estimation of population variance. At the same time, it increases the mean squared error in the same estimations. With `n-1`, you are not exactly calculating the average squared distance, but it is approached as the sample size `n` increases. It is used when calculating spread for samples rather than for (ideal) populations - so in practice it is mostly used.

10 Standard deviation (`sd`)

- Thinking about "average of the distance from the mean squared" is not very intuitive. To fix that, we can take the square root.

- The *standard deviation* is the square root of the variance

$$s_x = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

- For example, for `xdata <- c(2, 4.4, 3, 3, 2, 2.2, 2, 4)`

$$\sqrt{0.908} = 0.953$$

- Roughly speaking, 0.953 represents the average distance of each observation from the mean.
- `[X]`

  Is it better if the standard deviation is large or small?

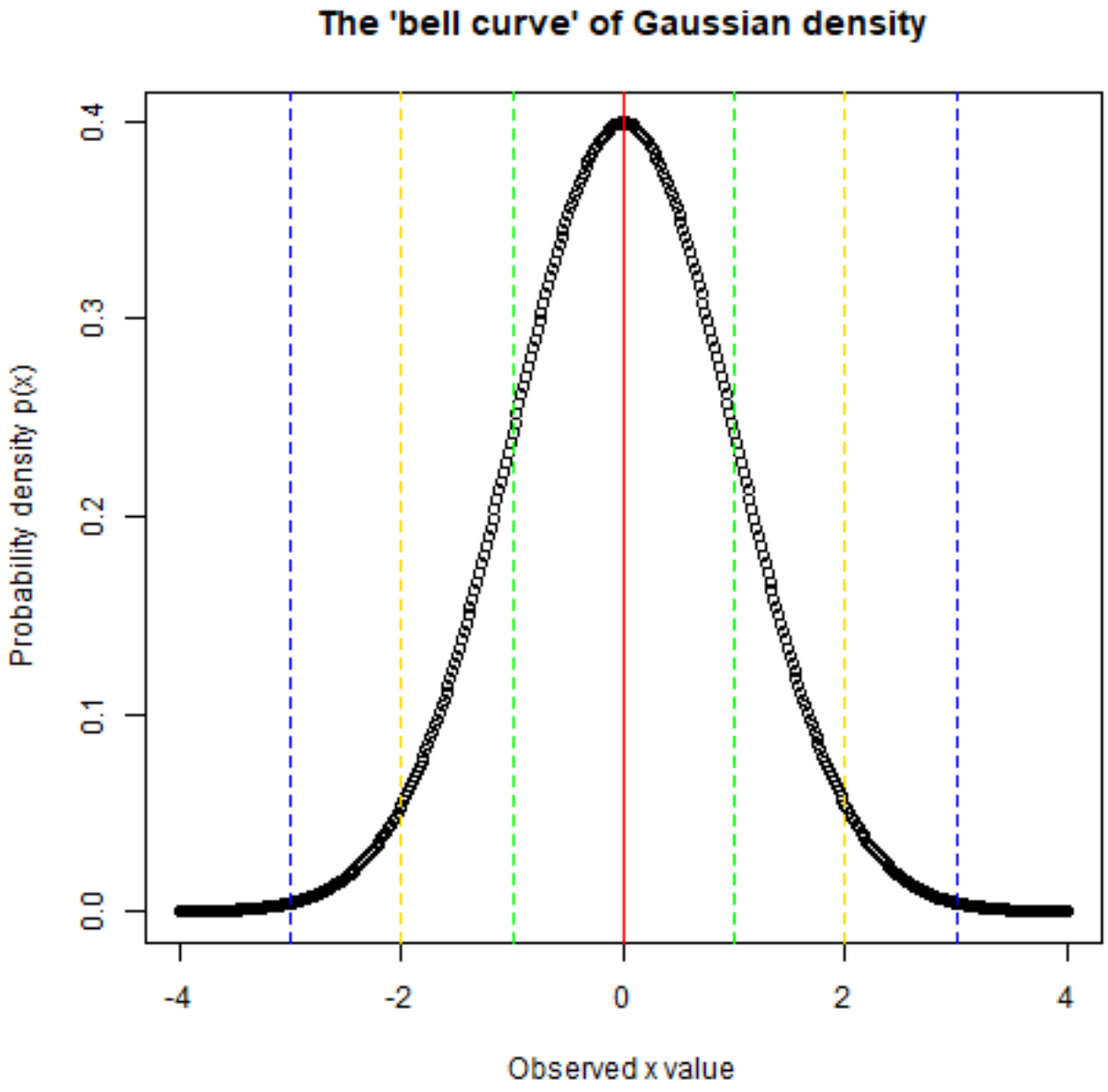
  It depends. If you manufacture machine parts, you want σ to be small so that you can be sure that all pieces are about the same. If you're looking at wages in a large company, I'll naturally be large, because there is a large spread between low and high earners.

- Interesting: the conflict between model-centric vs. data-centric data science (ref. Andrew Ng, Stanford U)

11 Ideal and real spread

- For numerical data, we expect values to conform to the normal or Gaussian distribution described by a "bell curve".
- The plot shows a 'bell curve' of Gaussian density generated from random points.

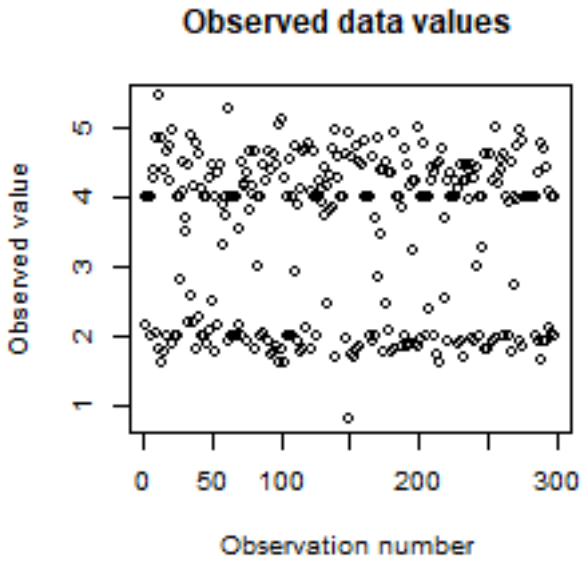
```
par(mfrow=c(1,1))
x <- seq(from=-4,to=4,by=0.02)
y <- dnorm(x, mean = 0, sd = 1)
plot(x,y, type="p", pch=1,
     main="The 'bell curve' of Gaussian density",
     xlab="Observed x value",
     ylab="Probability density p(x)")
abline(v=mean(x), col="red", lty=1)
abline(v=1, col="green", lty=2)
abline(v=-1, col="green", lty=2)
abline(v=2, col="gold", lty=2)
abline(v=-2, col="gold", lty=2)
abline(v=3, col="blue", lty=2)
abline(v=-3, col="blue", lty=2)
```

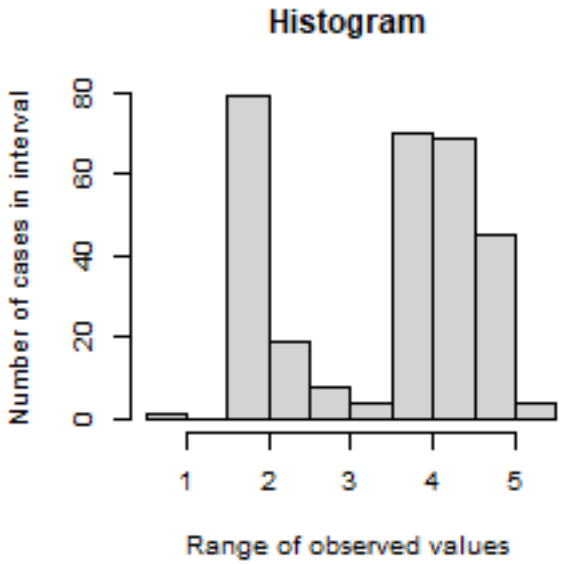

The 'bell curve' of Gaussian density
Probability density p(x)
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4
Observed x value

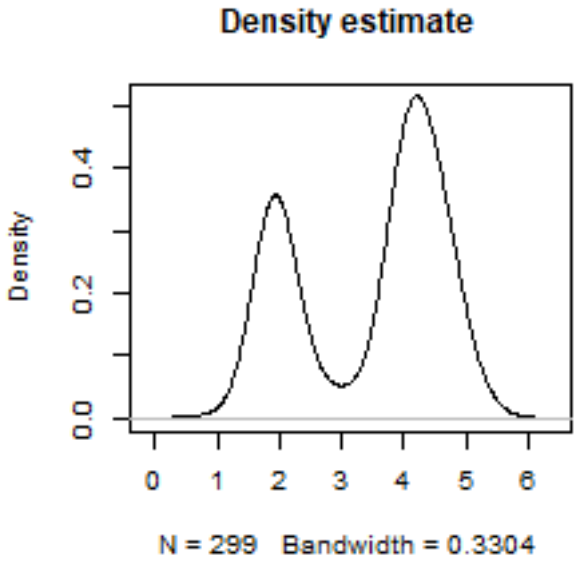
- One use of this distribution: when comparing standard deviation and mean across different data sets, we have to transform each set of data into a more generic distribution with a mean of 0 and a standard deviation of 1.
- Base R has many different tools to characterize such data, e.g.
    1. Scatterplot
    2. Histogram of random numbers
    3. Density estimate
    4. Normal QQ-plot

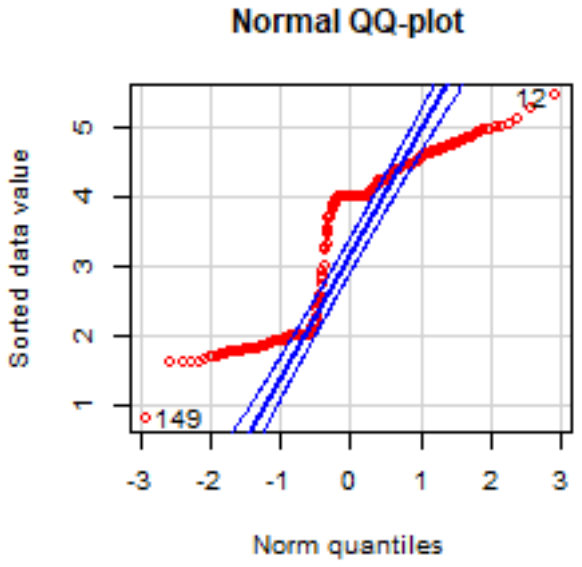
- Here is how this looks like for real data, the `geyser` data set of 299 eruptions of the Old Faithful geyser in Yellowstone National Park, WY.

```
library("car")
library("MASS")
par(mfrow=c(2,2))
x <- geyser$duration
plot(x, type="p", pch=1,
     main="Observed data values",
     xlab="Observation number",
     ylab="Observed value")
hist(x,
     main="Histogram",
     xlab="Range of observed values",
     ylab="Number of cases in interval")
plot(density(x),
     main="Density estimate")
qqPlot(x, col="red",
       main="Normal QQ-plot",
       xlab="Norm quantiles",
       ylab="Sorted data value")
```


Observed data values
Observed value
Observation number
0
50
100
200
300
1
2
3
4
5


Histogram
Number of cases in interval
Range of observed values
0
20
40
60
80
1
2
3
4
5


Density estimate
Density
N = 299 Bandwidth = 0.3304
0.0
0.2
0.4
0
1
2
3
4
5
6


Normal QQ-plot
Sorted data value
Norm quantiles
149
12
1
2
3
4
5
-3
-2
-1
0
1
2
3

12 Interquartile Range (IQR)

- The interquartile range (IQR) is not computed with respect to the sample mean
- It measures the "middle 50%" of the data - the range of values that lie within a 25% quartile on either side of the median

- It is computed as the difference between the upper and lower quartiles of the data. For example for `xdata` these values were:

```
xdata <- c(2, 4.4, 3, 3, 2, 2.2, 2, 4)
quantile(xdata, prob=c(0.25,0.75))
```

- If $Q_X()$ denotes the quartile function, then $IQR_X = Q_X(0.75)-Q_X(0.25)$, and $IQR_{xdata}=3.25-2.00=1.25$

13 R functions

- Let's compute spread stats in R:

```
var(xdata)   # variance
sd(xdata)    # standard deviation
IQR(xdata)   # interquartile range
```

- Confirm the definitory relationship between variance and standard deviation numerically:

```
identical(sqrt(var(xdata)),sd(xdata))
```

- Let's confirm the definition of the IQR numerically:

```
as.numeric(quantile(xdata,0.75)-quantile(xdata,0.25))
```

- Note that `as.numeric` strips away the percentile annotations of `quartile` results
- Compute standard variation and IQR for `ydata`:

```
sd(ydata)
IQR(ydata)
```

- Confirm that `ydata` are more spread out than `xdata`

```
paste("Are ydata more spread out than xdata?")
paste("Standard deviation: ", sd(ydata) > sd(xdata))
paste("Interquartile range: ",IQR(ydata) > IQR(xdata))
```

14 Practice: chick weights and quakes

- Let's return to chicks and quakes
- We computed the mean `weight` of all chicks in the `chickwts` data set

```
weights <- chickwts$weight
mean(weights)
```

- How far is the weight of each chick on average away from the mean?

```
sd(weights)
```

- Technically, this value is the square root of a function of the squared distances of all observations in the sample
- We computed the five-point summary of the magnitudes of some of the earthquakes in the `quakes` data set

```
magnitudes <- quakes$mag[quakes$depth < 400]
summary(magnitudes)
```

- What is the width, in units of the Richter scale, of the middle 50% of these observations?

```
IQR(magnitudes)
```

15 Glossary: concepts

TERM	MEANING
Quantile	
Quartile	
Percentile	
Tukey's 5-point summary	

TERM	MEANING
Interquartile Range (IQR)	
Range	
Lower/upper quartile	
Variance	
Standard deviation	

16 Glossary: code

CODE	MEANING
`quantile(x,prob=seq(0,1,0.25),na.rm=FALSE)`	Quantile
`summary`	5-Point-Summary
`as.numeric`	Conversion to numeric
`IQR`	Interquartile range
`var`	Variance
`sd`	Standard deviation

17 References

- Davies TD (2016). Book of R. NoStarch Press. URL: nostarch.com

- Udias A (1996). The Jesuit Contribution to Seismology. URL: seismosoc.org

Author: MARCUS BIRKENKRAHE
Created: 2022-10-13 Thu 20:45