Workshop on Quantitative Methods in Linguistics (WoQuMeL)

School of Languages and Linguistics | Jadavpur University

Outline

- Objectives
- To learn statistical methods for quantitative analyses of linguistic data
- Approach empirical questions in linguistics from a model-theoretic approach
- Models, similar to theories, make predictions, but not always.
- We tweak them till we arrive at a model whose unpredictable aspects are within acceptable bounds

What we will be following

- Quantitative methods in Linguistics (Johnson 2008)
- Code and datsets related to Johnson (2008)
  - Code and data
- Statistics for Linguistics: An Introduction Using R (Winter 2020)

Topics that we will cover

- Descriptive statistics
  - mean
- Distributions
- Models
- Data visualization
- Summary Stats
- Linear Models
- Correlations
- Multiple Regressions

Working with R

- Basic R functions and packages
- Designing and building the statistical components of experiments
- Writing code and debugging

This document

- We are writing R code and associated content in Quarto
- Markdown flavor syntax
- Weaving r code and text in the same document

What statistic are and what they are not

- Statistical analyses lend validity
- We perform tests that allow us to either accept or reject the null hypothesis
- They give us a means to uncover causal relationships
- They are, however, not magic wands
- Each test and set of analyses are specific to the conditions, variables, nature and distribution of the data; so we decide first before we conduct the experiment what tests to perform NOT after

Statistical environment

- R because it is:

  1. a powerful statistics package, good at reading data, wide range of statistical tests and techniques, good graphics, very flexible
  2. a usable package available for many platforms (PC, Mac, Unix, Linux….) programmable user community for support 3.it is noncommercial - distributed under the GNU "copyleft", maintained by a community of users, upgrades happen because the users need improvements, not because the company needs more money.

- Where: [R project page]
- How:

  1. Go to the R project page,
  2. click the CRAN link to see the download servers on the Comprehensive R Archive Network,
  3. choose a download server near your location,
  4. choose your platform (Windows, Linux, Mac)

Describing data

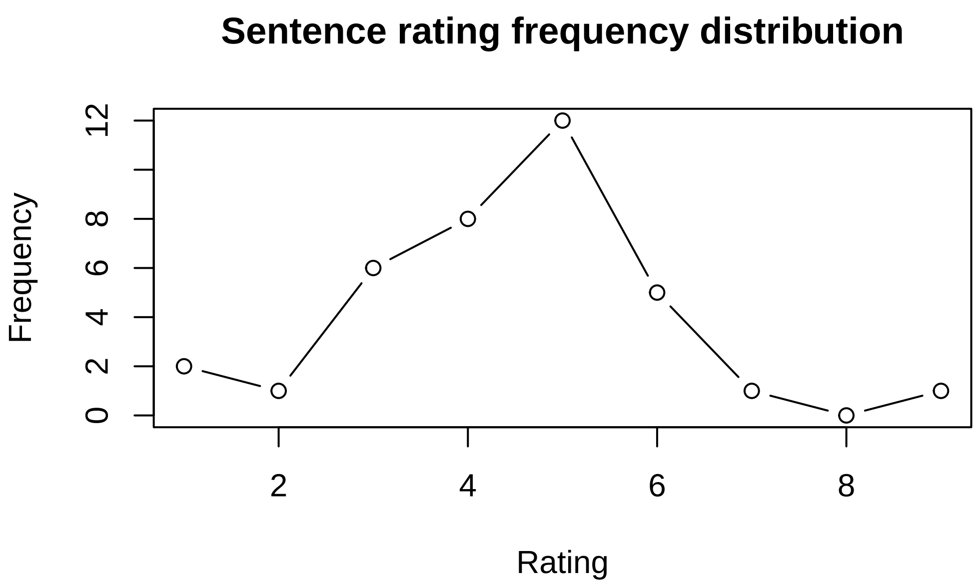
- Let's say we ask 36 people to score a sentence on a grammaticality scale So that a score of 1 means that it sounds pretty ungrammatical, and 10 sounds perfectly OK. A simple way of generating data in R

```
x=round(rnorm(36,4.5,2))
```

- *rnorm* needs some arguments: N, mean and the SD
- How many people gave the sentence a rating of "1"?
- How many rated it a "2"? When we answer these questions for all of the possible ratings we have the values that make up the *frequency distribution* of our sentence grammaticality ratings

Getting the frequency distribution

```
data = c(2,1,6,8,12,5,1,0,1)#c function to catenate individual values together
rating = c(1,2,3,4,5,6,7,8,9)
plot(rating,data,type = "b", main="Sentence rating frequency distribution",
     xlab = "Rating", ylab = "Frequency")
```



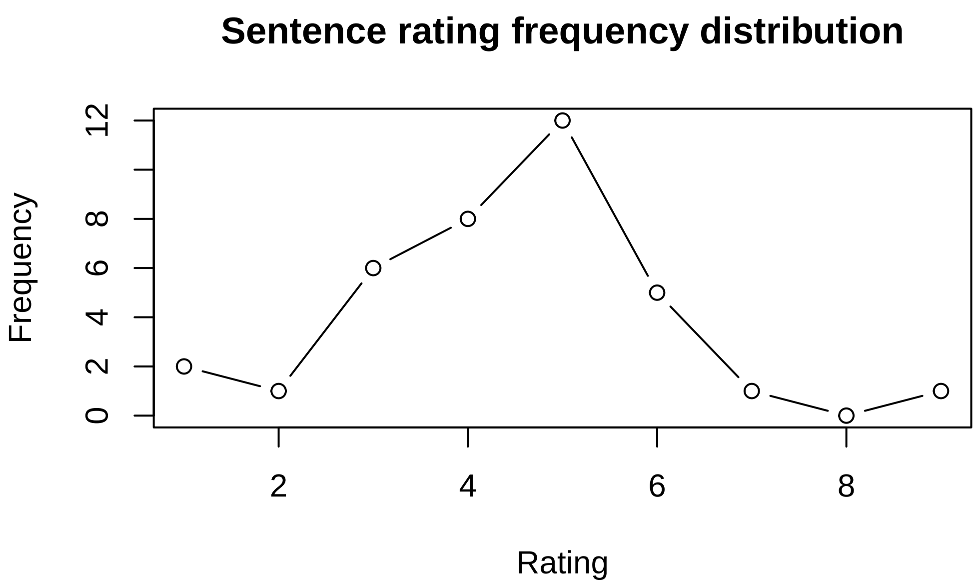
- Here we defined two vectors, data and rating, data is the frequency data of ratings, and rating refers to a vector of the rating scale
- How many people gave a particular sentence the rating of 5? Or how frequently was the rating 5 given?

What is a vector?

- Container vector
  - Ordered collection of numbers with no other structure
  - The length of a vector is the number of elements in the container.
- Operations are applied componentwise.
  - Given two vectors x and y of equal length, x*y would be the vector whose nth component is the product of the nth components of x and y.
  - log(x) would be the vector whose nth component is the logarithm of the nth component of x.

How informative are frequency distributions?

```
plot(rating,data,type = "b", main="Sentence rating frequency distribution",
     xlab = "Rating", ylab = "Frequency")
```

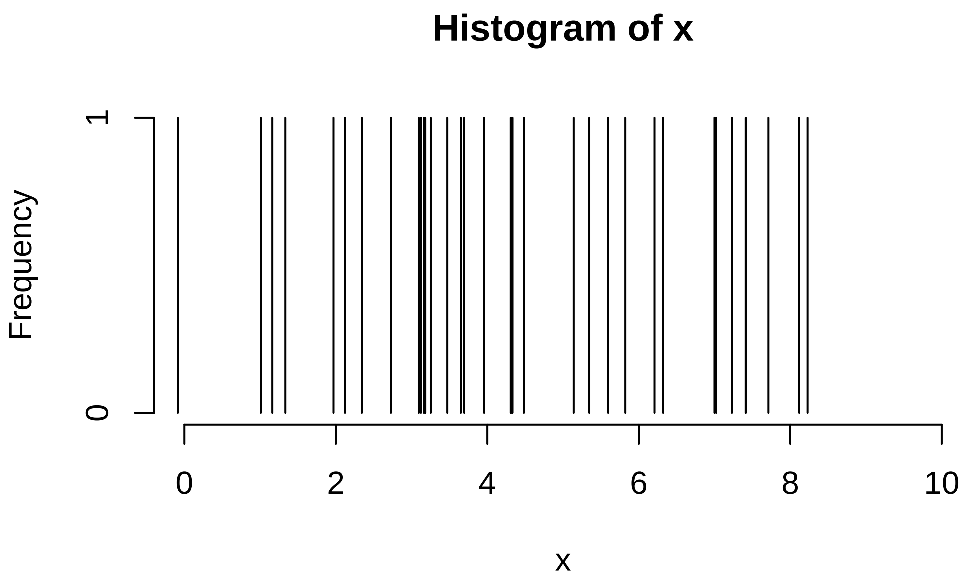


- Plotting rating and data gives us the frequency distribution
- Majority of subjects (12) rated the sentence to be 5 on the scale
- Few people rated the sentence to be absolutely ungrammatical rating of 1 (2) and absolutely grammatical rating of 9 (1)
- A lot many subjects rated the sentence to be 5 than 1 or 9
- This suggests that the frequency of ratings is crowded around the average rating of 4.5

Changing the granularity of the rating scale

- The rating scale we used forces the subject to rate in integers
- Imagine a situation were subjects are given the freedom to use decimals to rate
- If so, then: no two ratings are ever going to be the same; each subject will have a rating that is different from the other, and will have a frequency of 1

```
x=rnorm(36,4.5,2)
hist(x, breaks=300000,xlim=c(0,10))
```



- If we quantize this difference and put individual ratings in intervals, say between 0 and 1, 1 and 2, and 2 and 3, again we will get a distribution similar to the first one

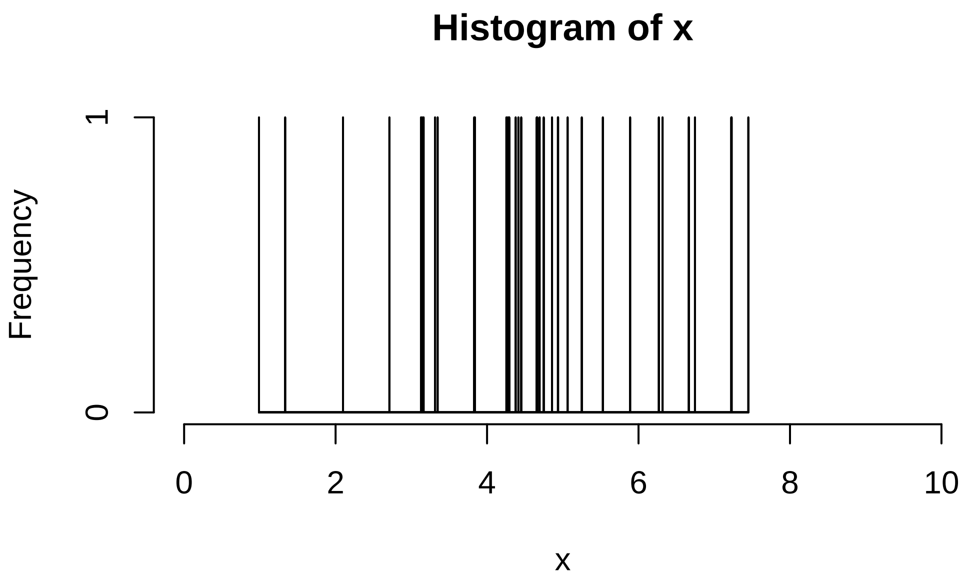
Frequency distribution in R

- How did we generate these plots and distributions
- First we defined a vector using the function, rnorm

```
x = rnorm(36, 4.5, 2)
#notice that this is different from round(rnorm(36,4.5,2)) where we had asked for rounded/in
```

- We defined a vector, x, with 36 values, a mean of 4.5 and standard deviation of 2.
- So decimal ratings would be ok
- Then we made two histograms
  - First with:

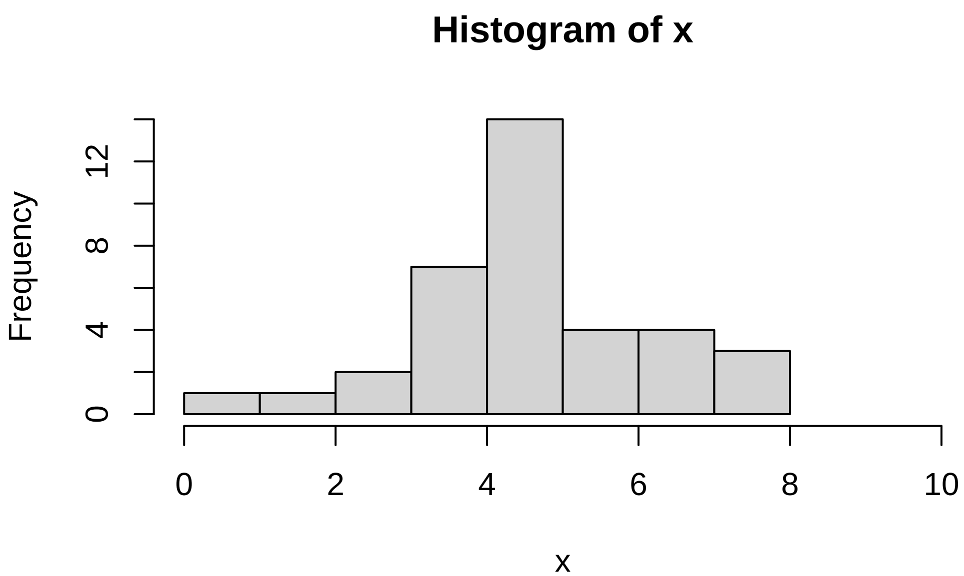
```r
hist(x,breaks=30000, xlim = c(0,10))
```




```r
#here we want to plot a histogram where the width of the cells/bins is very small
```

- Second with:

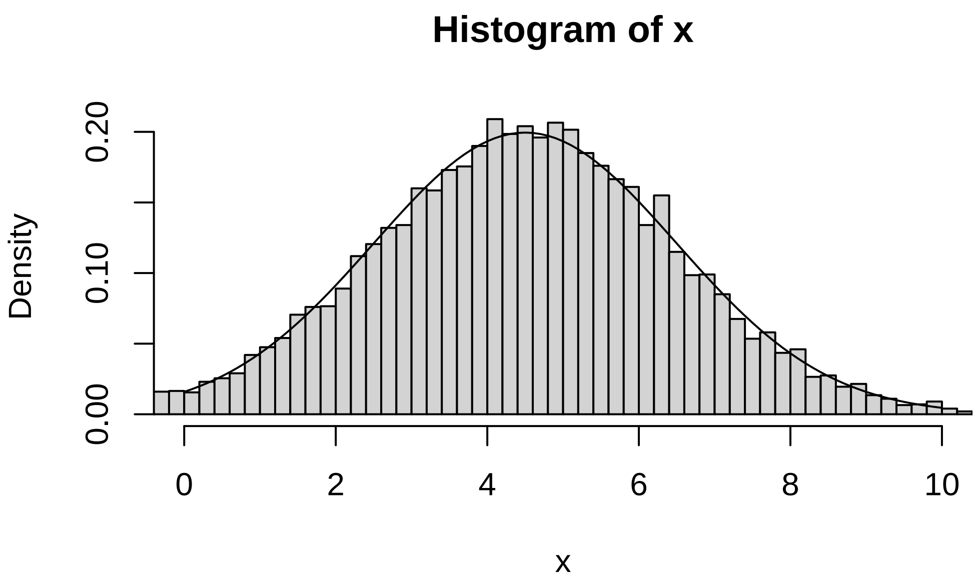
```r
hist(x, xlim = c(0,10))#here we want to plot a histogram where the width of the cells/bins is
```



Theoretical fequency distributions

- Suppose we could draw from an infinite data set
- The larger our data set - more detailed a representation of the frequency distribution
- If we keep collecting sentence grammaticality data for the same sentence, so that instead of ratings from 36 people we have ratings from 10,000 people
- With a histogram that has 1000 bars in it, we see that ratings near 4.5 are more common than those at the edges of the rating scale
- Adding observations up to infinity and reducing the size of the bars in the histogram of the frequency distribution
- Intervals between bars is vanishingly small - i.e. we end up with a continuous curve, almost
- Plotting the normal distribution curve on the frequency distribution

```
x = rnorm(10000, 4.5, 2)
hist(x,breaks=100,freq=FALSE,xlim = c(0,10))
plot(function(x)dnorm(x, mean=4.5, sd=2), 0,10, add=TRUE)
```



Adding the normal curve

- Why the excellent fit between the "observed" and the theoretical distributions?
- The data is generated by random selection
  - rnorm() - observations from the theoretical normal distribution dnorm()
- The "normal distribution" is an very useful theoretical function because…

1. Let's assume that there is an underlying property that we are trying to measure like - grammaticality, or
   - typical duration, or
   - amount of processing time
2. Assume that there is some source of random error that makes it difficult for us to get to this underlying property

- If so, then we can think that - the "true" value of the underlying property we want to measure –
  - Must be at the center of the frequency distribution that we observe in our measurements
  - And, the distribution (we observe) is caused by error - with (the probability of) bigger errors being less likely than smaller errors

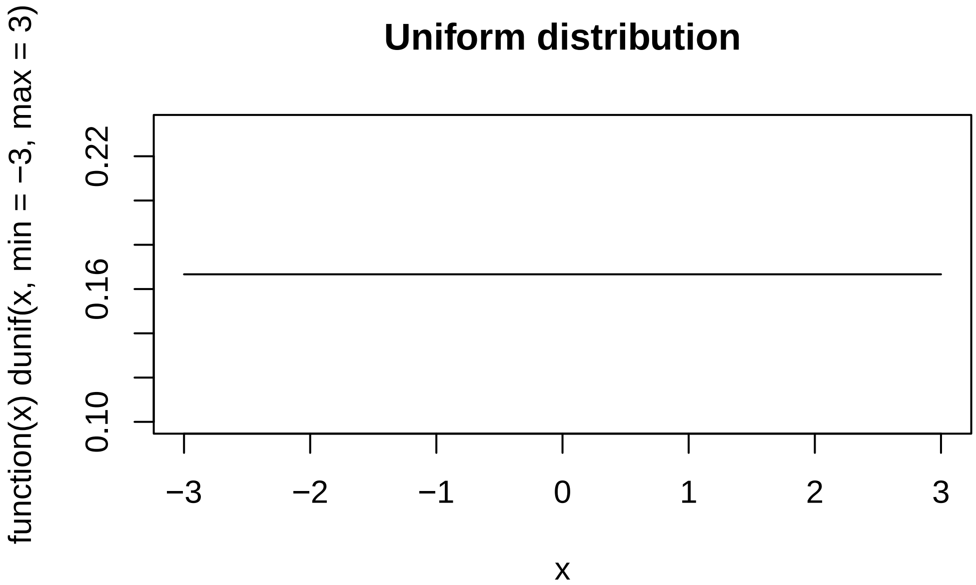
The Normal Distribution

- The normal distribution is described by the normal curve, or the bell-shaped curve
- It is an exponential function of the mean value (μ "mew") and the variance (σ "sigma")
- The sum of the area under the curve, fx is 1
- Derived from just two numbers, the mean value and a measure of how variable the data are
- The area under the curving equalling to 1, is also useful to go from frequency distributions to probability densities
- This is related to hypothesis testing
  - $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2 / 2\sigma^2}$
- *e* is Euler's constant

Type of distributions

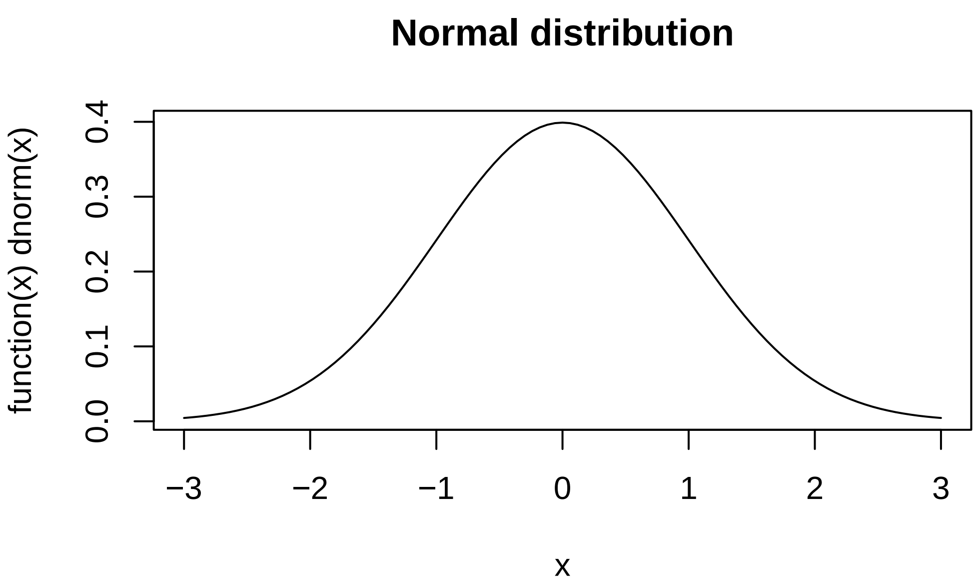
- Uniform distribution: Every outcome is equally likely
  - Six sides of a dice - equal likelihood that either side will be rolled

```
uni=plot(function(x)dunif(x,min=-3,max=3), -3,3, main="Uniform distribution")
```



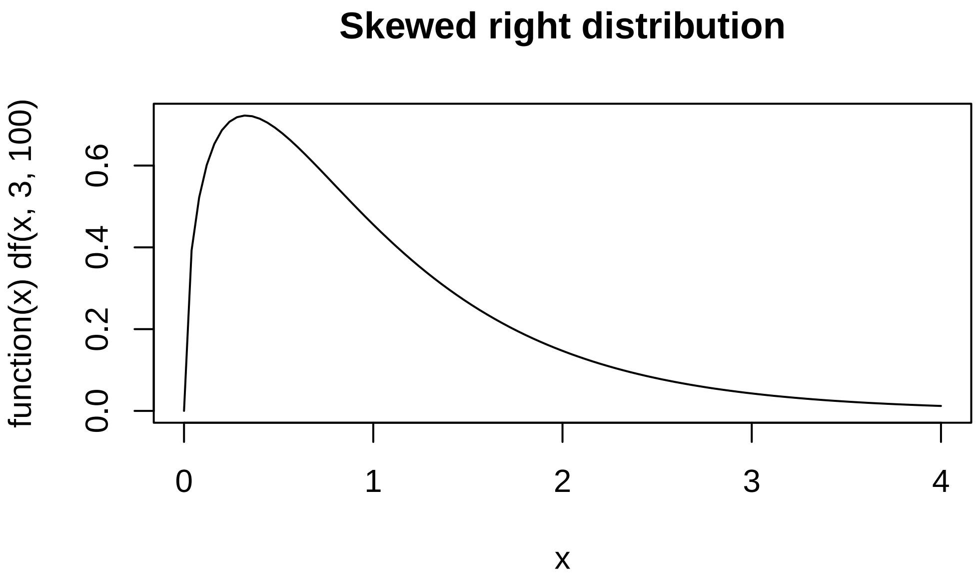
- Normal, bell-shaped distribution, measurements congregate around a typical value and values become less and less likely as they deviate from the central value

```
norm=plot(function(x)dnorm(x), -3,3, main="Normal distribution")
```



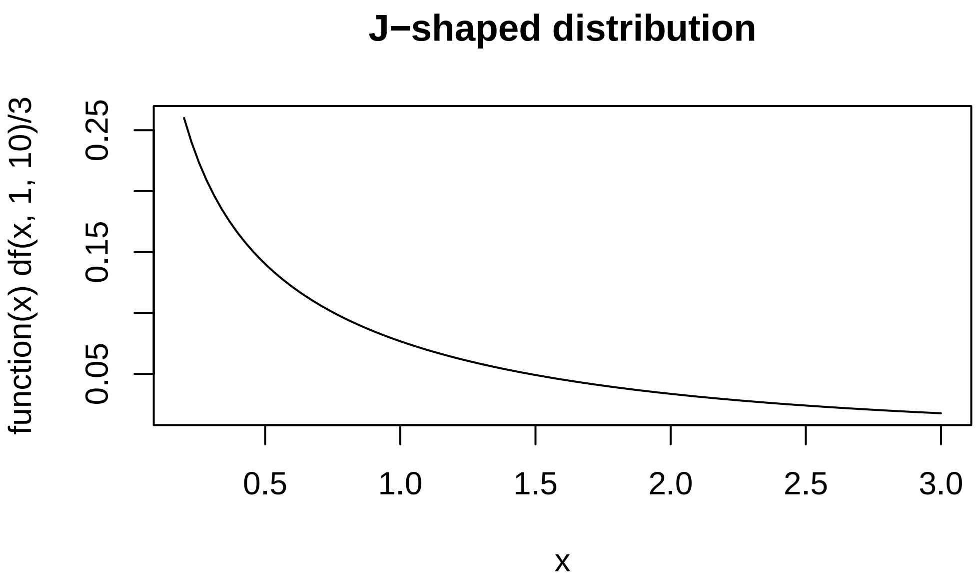
- Skewed right: Skewed frequency distributions
  - percentage data and reaction time data
  - Mean is no longer 'central' to the distribution, or extreme values (from one end of the scale and less from the other) dominate the distribution

```
skewed=plot(function(x)df(x, 3, 100),0,4, main="Skewed right distribution")
```



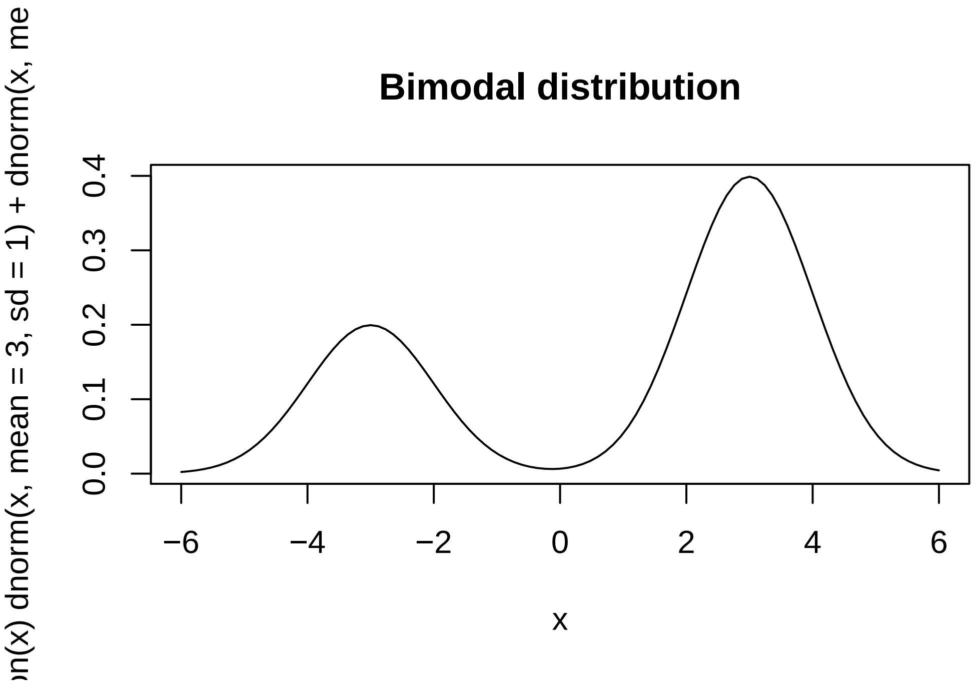
- The J-shaped distribution is a special kind of skewed distribution
  - Most observations come from the end of the measurement scale
  - Most speech errors counts per utterance will have a speech error count of 0

```
j=plot(function(x)df(x, 1, 10)/3,0.2,3, main="J-shaped distribution")
```



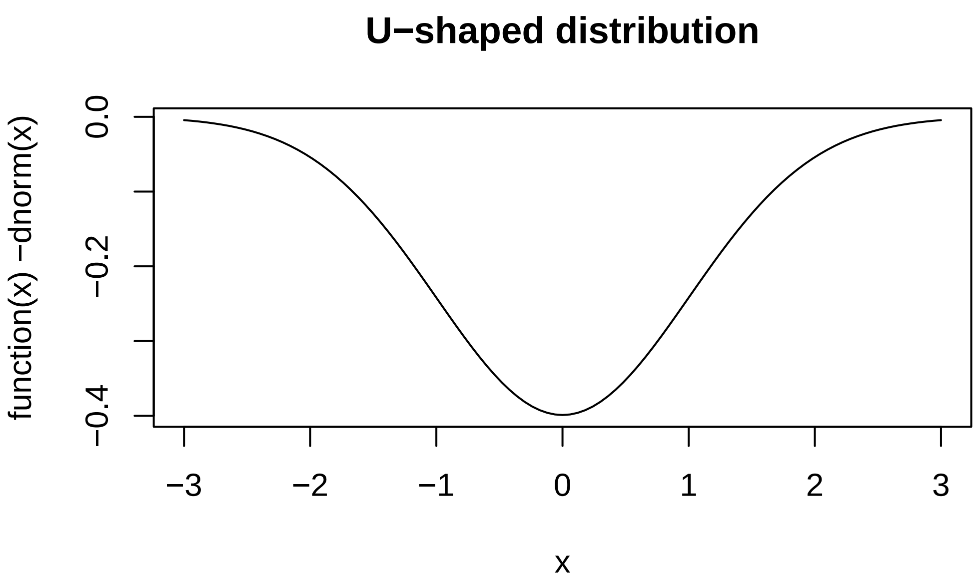
- Bimodal distribution is a frequency distribution where clearly two modalities are involved. For instance
  - f_0 (or pitch) for men and women

```
bimodal=plot(function(x)dnorm(x, mean=3, sd=1)+dnorm(x, mean=-3, sd=1)/2,-6,6,
     main="Bimodal distribution")
```



- U shaped distributions result out of polarization where subjects may take drastically one view or the other

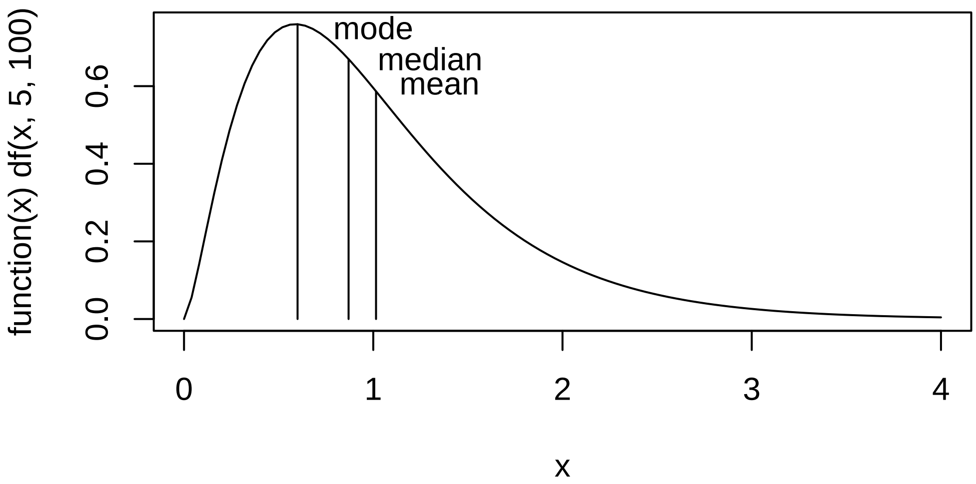
```
u=plot(function(x)-dnorm(x),-3,3,
     main="U-shaped distribution")
```



Measures of central tendency

```
plot(function(x)df(x,5,100),0,4, main="Measures of central tendency")
lines(x=c(0.6,0.6),y=c(0,df(0.6,5,100)))
skew.data <- rf(10000,5,100)
lines(
  x=c(mean(skew.data), mean(skew.data)),
  y=c(0,df(mean(skew.data),5,100)))
lines(
  x=c(median(skew.data),median(skew.data)),
  y=c(0,df(median(skew.data),5,100)))
text(1,0.75,labels="mode")
text(1.3, 0.67,labels="median")
text(1.35,0.6,labels="mean")
```

Measures of central tendency



- Normal distribution - the central 'values' (from our samples) have the highest probability of being part of the population
- What are these?
- The most frequently occurring value - *mode* - the tip of the frequency distribution. In the skewed distribution, the mode is 0.6
- The central value, that is in an ordered dataset of the values, the one in the middle is the *median*; aka, the center of gravity
- Arithmetic *mean*, or the sum of values divided by the total number of values, n
- *Least squares estimate of central tendency*

  1. take the difference between the mean and each value in our data set
  2. square these differences and
  3. add them up

- We will get a value that will be smaller than what we would get if we took the median or any other estimate of the "mid-point" of the data set

Weighted means

- Means represent the least squared estimate of the central tendency; say of ratings
- What if we also asked each subject to rate their ratings of grammaticality with a weight, wi
- This way those ratings with a higher weight will give a better estimate of the central tendency; confidence values
- The weights represent the confidence each rater has on her particular rating
- Sample mean = $\bar{x} = \frac{\sum_{i=0}^{n} x_i}{n}$
- Weighted mean = $\bar{x} = \frac{\sum_{i=0}^{n} w_i x_i}{\sum_{i=0}^{n} w_i}$

- Population variance = $\sigma^2 = \sum \frac{(x_i-\mu)^2}{N}$
- Sample variance = $s^2 = \sum \frac{(x_i-\bar{x})^2}{n-1}$

Measures of dispersion

- The mean absolute deviation measures the absolute difference between the mean and each observation
- Absolute deviation could be one measure of difference, where absolute values of the difference for each x_i and sample mean, $\bar{x}$ could be added
- We don't because the mean is the least squares estimator of central tendency
  - so a measure of deviation that uses squared deviations is more comparable to the mean
  - Sum of the squared deviations, $d^2 = \sum_{i=0}^{n}(x_i - \bar{x})^2$
- Variance
  - We square the deviations before averaging them
  - We have definitions for variance of a population and for a sample drawn from a larger population
  - Notice that sample variance, s^2 is calculated by dividing the sum of the squared deviations by n-1 and not n

Why n-1

- Generalize about the process but we only have access to the samples
- Relationship between scores, std. deviation and error
- Accurately talk about the population
  - when we only have access to samples we divide by n-1
  - Taking (n-1) as the denominator in the definition of s^2, sample variance, because $\bar{x}$ is not μ
  - Sample mean $\bar{x}$ is only an estimate of μ, derived from the x_i, so in trying to measure variance we have to keep in mind that our estimate of the central tendency $\bar{x}$ is probably wrong to a certain extent
- The mean of the underlying process (population) we don't know
- The mean of the n points we do, this however contains an error due to statistical noise
- Effect of the error is reduction in the calculated value of s^2
- To make up for this, n is replaced by n-1
- *If n is large, the difference doesn't matter*
- *If n is small, this replacement provides a more accurate estimate of the standard deviation of the underlying process*

Standard deviation

- Variance is the average squared deviation - the differences are squared
- To get to the original unit of deviation we take the square root of the variance; sample and population
- Aka, the RMS (root mean square) sample standard deviation

  1. first square the difference
  2. then take the mean and then
  3. square root of that

- Sample standard deviation
  - $s = \sqrt{\sum \frac{(x_i - \bar{x})^2}{n-1}}$
- Area under the normal distribution is equal to 1
- Measures of the central tendency in terms of $\bar{x}$ (sample mean) and also the sample standard deviation, s
- Normal distribution can be defined for any mean value μ, and any standard deviation σ
- This distribution is also used to calculate probabilities, where the total area under the curve is equal to 1
- That means that the area under any portion of the curve is equal to some proportion of 1
- This happens, when the mean of the bell-shaped distribution is 0 and the standard deviation is 1
  - $f_x = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$

Distributions

- Throwing a six sided dice 20 times
- Let's note down all the 20 outcomes
- Drawing from a uniform distribution
- Sample distribution
- For every outcome count the number of times it appears

Z-score and normalization

- Two things to remember:

  1. Since the area under the normal distribution curve is 1, we can state the probability (area under the curve) of finding a value larger than any value of x, smaller than any value of x, or between any two values of x; relating individual scores to the normal distribution

2. Since, we can approximate our data with a normal distribution - we can state these probabilities for our data given the mean and standard deviation; under the assumption that our data are normally distributed

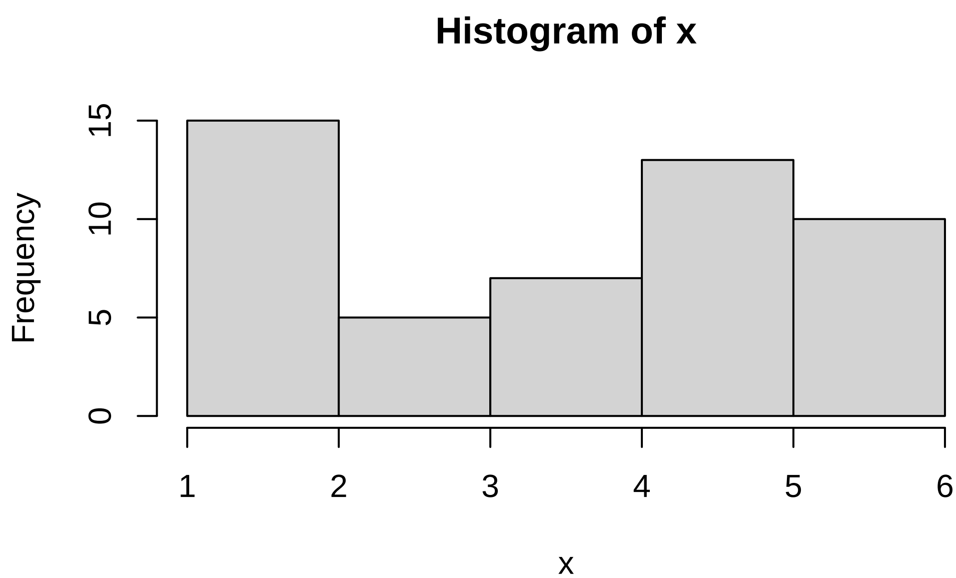
- Relate the frequency distribution of our data to the normal distribution because we know the mean and standard deviation of both
- Key is to be able to express any value in a data set in terms of its distance in standard deviations from the mean
- z-score normalization, $z_i = \frac{x_i - \bar{x}}{s}$

```
#------- shade.tails -------------------
# draw probability density functions of t with critical regions shaded.
#   by default the function draws the 95% confidence interval on the normal
#   distribution.
#
# Input parameters
#   crit - the critical value of t (always a positive number)
#   df - degrees of freedom of the t distribution
#   tail - "upper", "lower" or "both"
#   xlim - the x axis range is -xlim to +xlim

shade.tails <- function(crit=1.96, df = 10000, tail = "both",xlim=3.5)
{

curve(dt(x,df),-xlim,xlim,ylab="Density",xlab="t")

ylow = dt(xlim,df)
pcrit = pt(crit,df)
caption = paste(signif(1-pcrit,3))

if (tail == "both" | tail == "lower") {
    xx <- seq(-xlim,-crit,0.05)
    yy <- dt(xx,df)
    polygon(c(xx,-crit,-xlim),c(yy,ylow,ylow),density=20,angle = -45)
    text(-crit-0.7,dt(crit,df)+0.02,caption)
}
if (tail =="both" | tail == "upper") {
    xx2 <- seq(crit,xlim,0.05)
    yy2 <- dt(xx2,df)
    polygon(c(xx2,xlim,crit),c(yy2,ylow,ylow),density=20,angle = 45)
    text(crit+0.7,dt(crit,df)+0.02,caption)
}
}
```

Sampling from a uniform distribution

- Storing outputs of functions in vectors
- Here, x, is a vector that stores the outout of the function *sample*
-

```
x <- sample(1:6,50,TRUE)
hist(x)
```



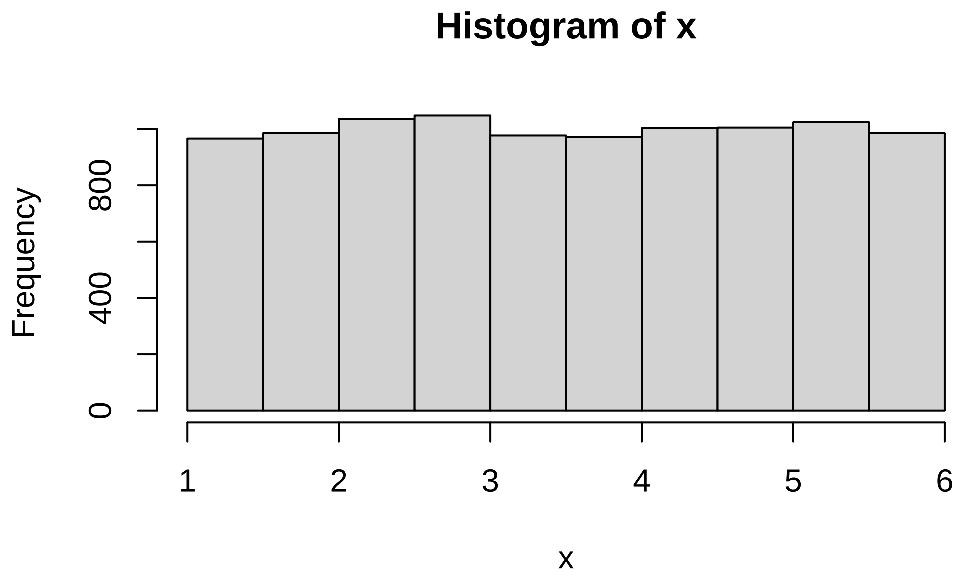

```
x
```

```
 [1] 5 4 2 2 5 4 4 5 6 2 6 4 5 4 1 2 6 5 1 2 6 6 5 6 2 1 6 4 6 3 1 6 5 5 3 1 5 6
[39] 1 2 5 3 3 2 1 3 5 4 5 5
```

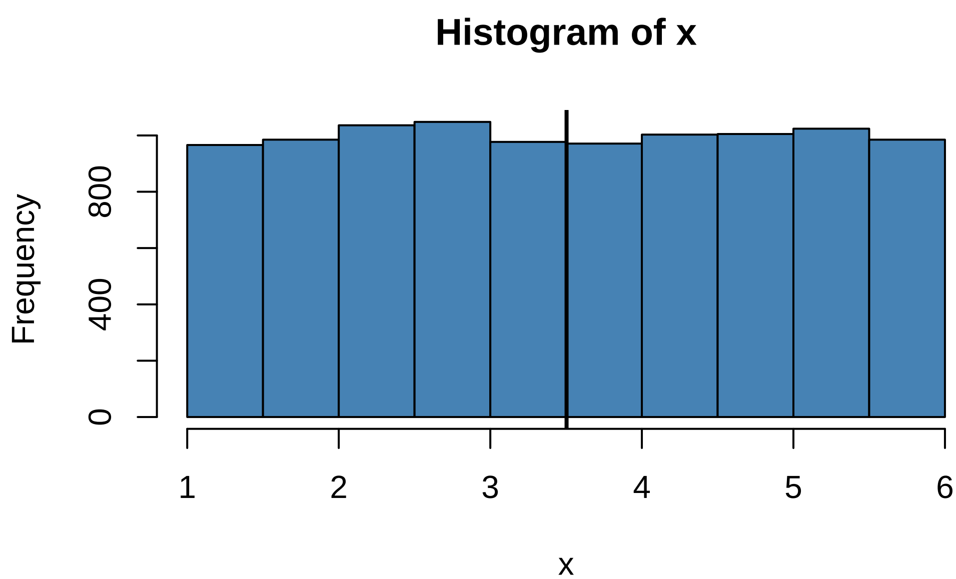
- Every time we run this code chunk the out of the sampling will change

---

```
x <- runif(10000, min = 1, max = 6)
hist(x)
```

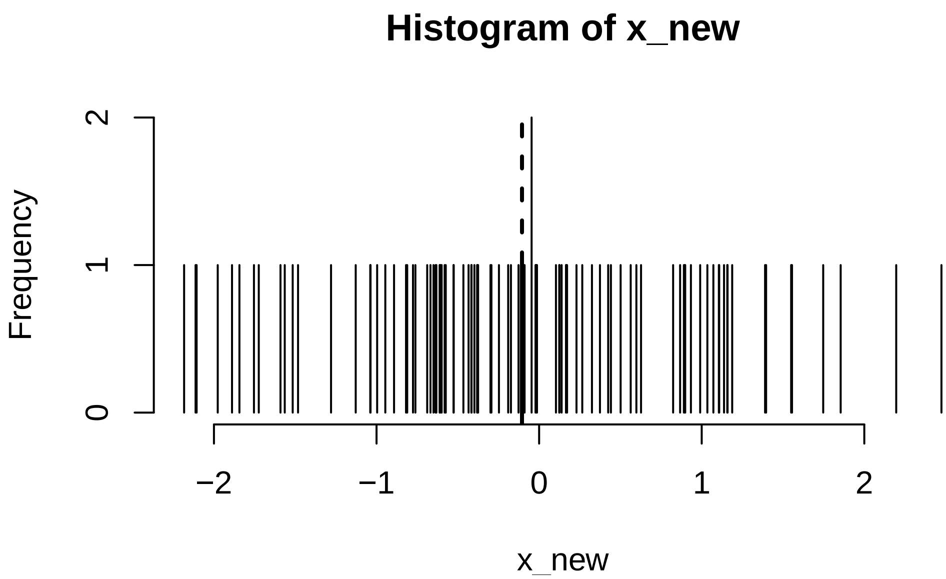


```
hist(x, col = 'steelblue')
abline(v = mean(x), lty = 1, lwd = 2)
```



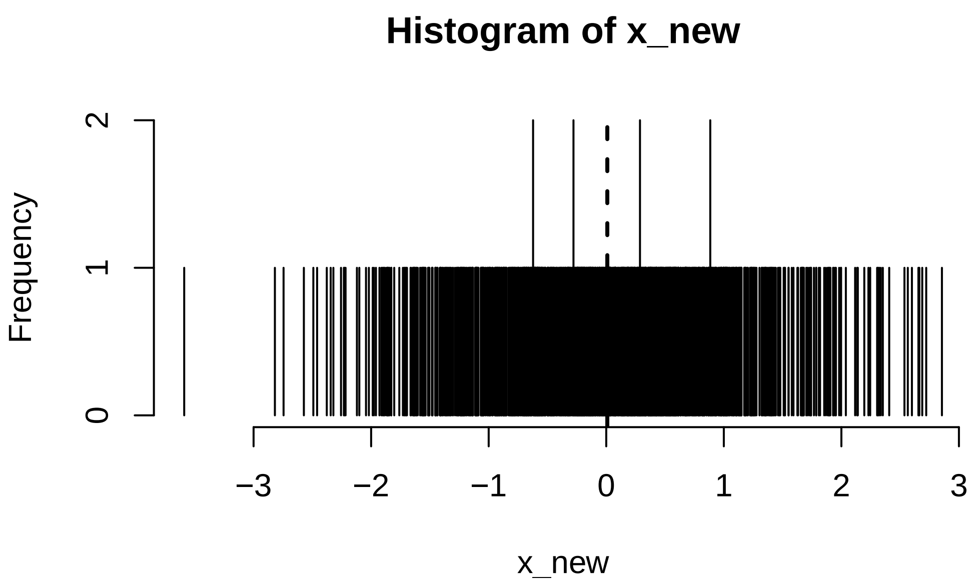
Uniform Distribution

```
x_new <- rnorm(100)
hist(x_new, breaks=100000,col = 'steelblue')
abline(v = mean(x_new), lty = 2, lwd = 2)
```



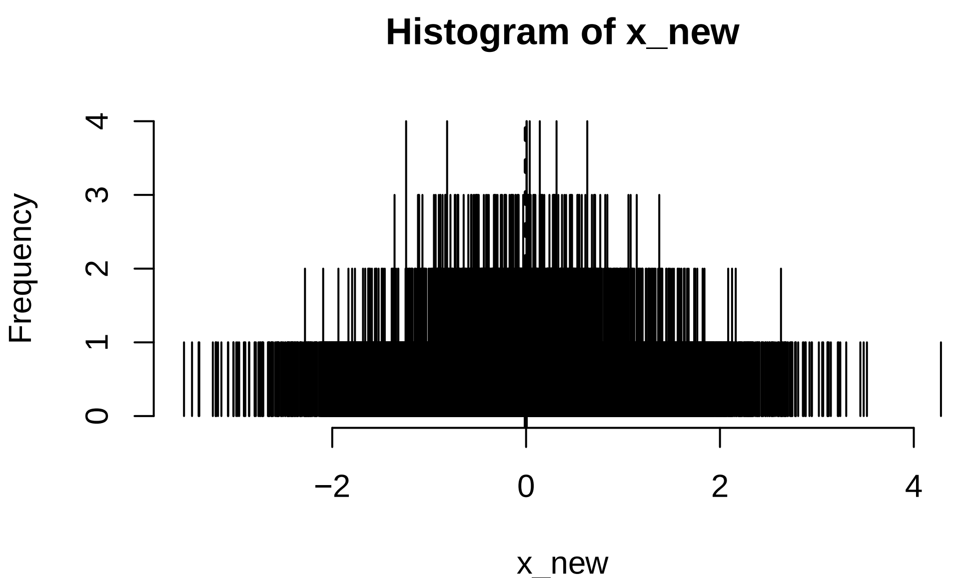
Still Uniform Distribution

```
x_new <- rnorm(1000)
hist(x_new, breaks=100000,col = 'steelblue')
abline(v = mean(x_new), lty = 2, lwd = 2)
```



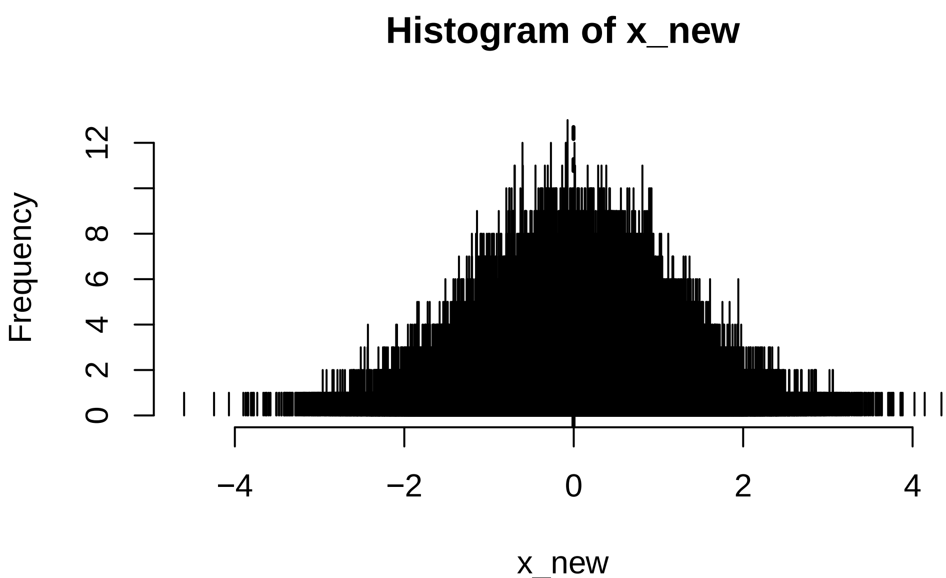
Normal or Gaussian Distribution

```
x_new <- rnorm(10000)
hist(x_new, breaks=100000,col = 'steelblue')
abline(v = mean(x_new), lty = 2, lwd = 2)
```



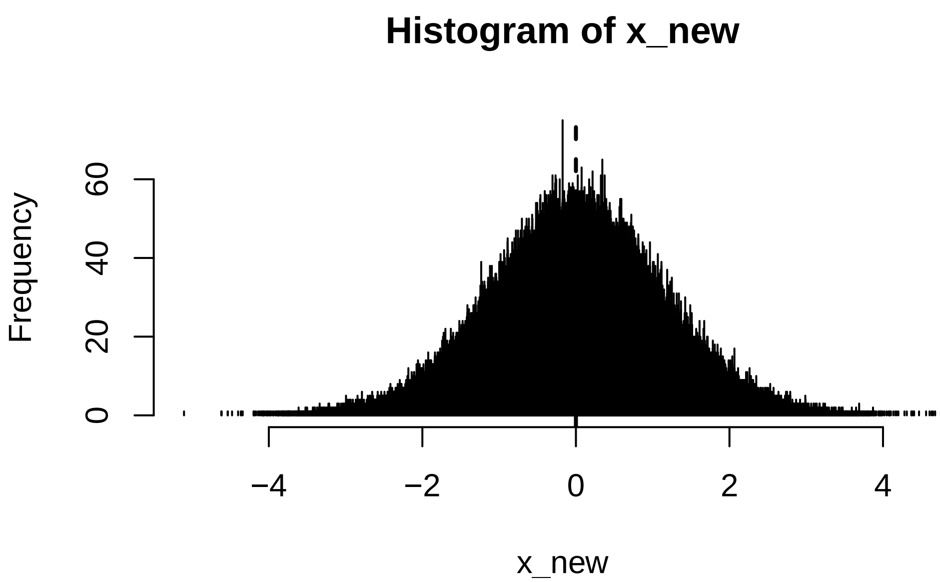
Increasing sampling in a normal or Gaussian Distribution

```
x_new <- rnorm(100000)
hist(x_new, breaks=100000,col = 'steelblue')
abline(v = mean(x_new), lty = 2, lwd = 2)
```



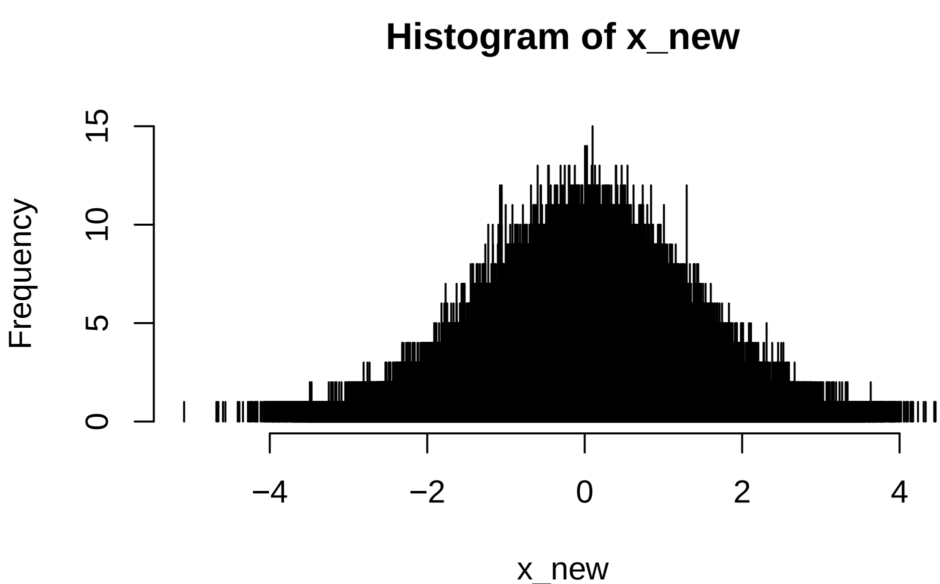
Further increasing sampling in a normal or Gaussian Distribution

```
x_new <- rnorm(1000000)
hist(x_new, breaks=100000,col = 'steelblue')
abline(v = mean(x_new), lty = 2, lwd = 2)
```



Increasing breaks now in a normal or Gaussian Distribution

```
x_new <- rnorm(1000000)
hist(x_new, breaks=1000000,col = 'steelblue')
```



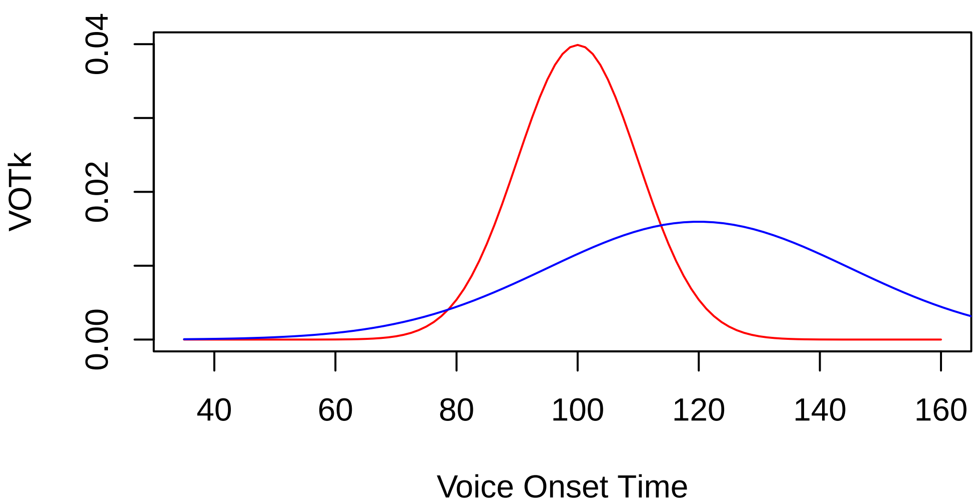
Getting some invariant parts of the sample: mean and standard deviation

- Sum of x $\sum x_i$
  - $\sum x_i^2$
  - $\sum x_i y_i$
- Mean of x $\frac{1}{n}\sum_{i=i}^{n} x_i$
- $Standard Deviation$

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

  σ is the population parameter
- Variance = σ^2

```
VOTk <- function(x) dnorm(x, mean = 100, sd = 10)
VOTp <- function(x) dnorm(x, mean = 120, sd = 25)
myYLim <- c(0, 0.04)
myXlim <- c(0,140)
plot(VOTk, from = 35, to = 160, ylim = myYLim, col="red",
     xlab="Voice Onset Time", myXlim)
plot(VOTp, from = 35, to = 200, add = TRUE, col="blue", ylim = myYLim, xlim=myXlim)
```



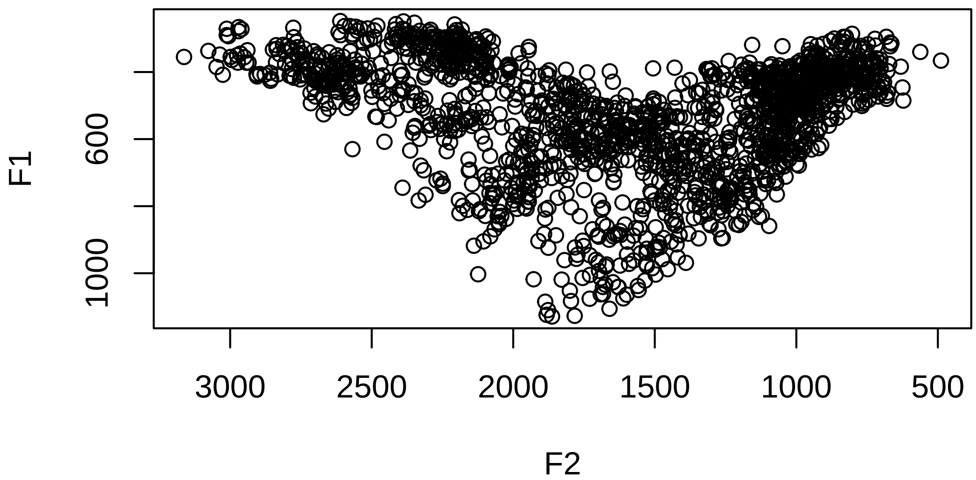
Plotting Vowels using PhonR

```r
library(phonR)
#(indo)
#head(indo)
summary(indo)
```

```
     subj             gender   vowel         f1              f2      
 Length:1725        f:867    a:349   Min.   : 248.0   Min.   : 489  
 Class :character   m:858    e:335   1st Qu.: 402.0   1st Qu.:1055  
 Mode  :character            i:348   Median : 493.0   Median :1509  
                             o:346   Mean   : 531.1   Mean   :1594  
                             u:347   3rd Qu.: 632.0   3rd Qu.:2097  
                                     Max.   :1129.0   Max.   :3163  
```

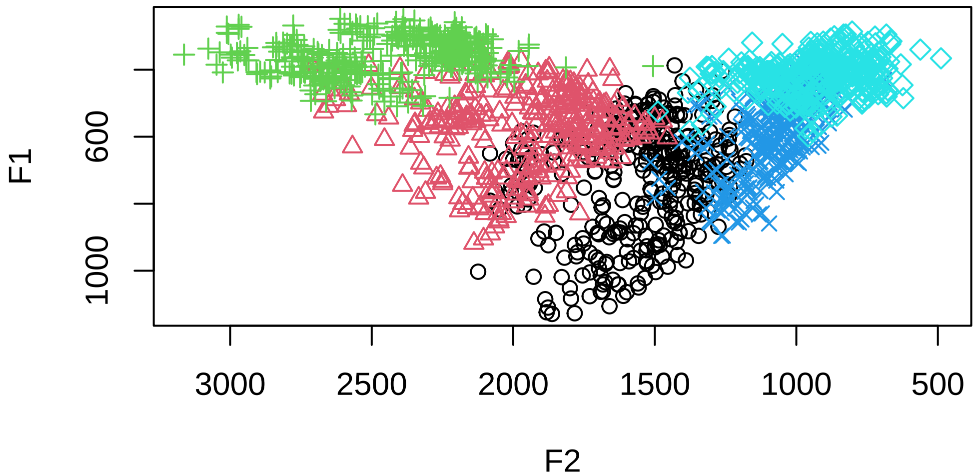
Plotting Vowels using PhonR

```r
with(indo, plotVowels(f1, f2))
```



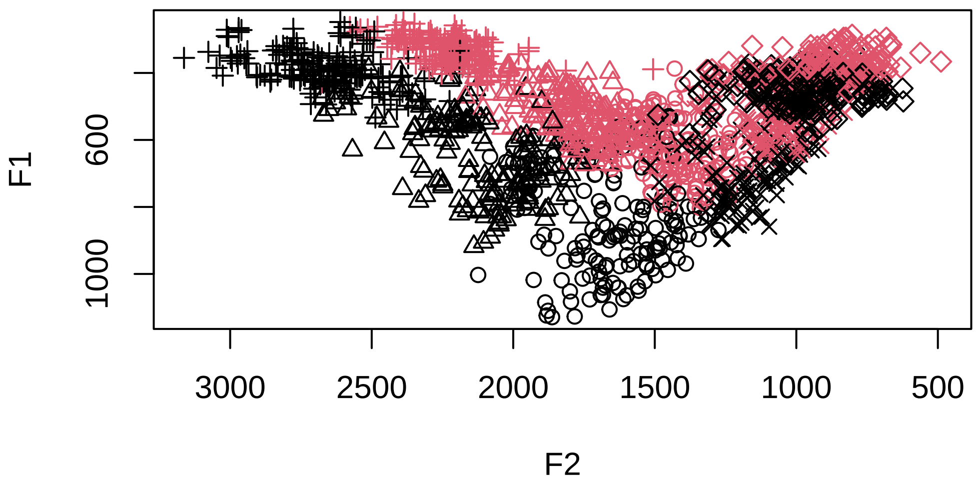
Plotting Vowels using PhonR

```r
with(indo, plotVowels(f1, f2, var.sty.by = vowel, var.col.by = vowel))
```



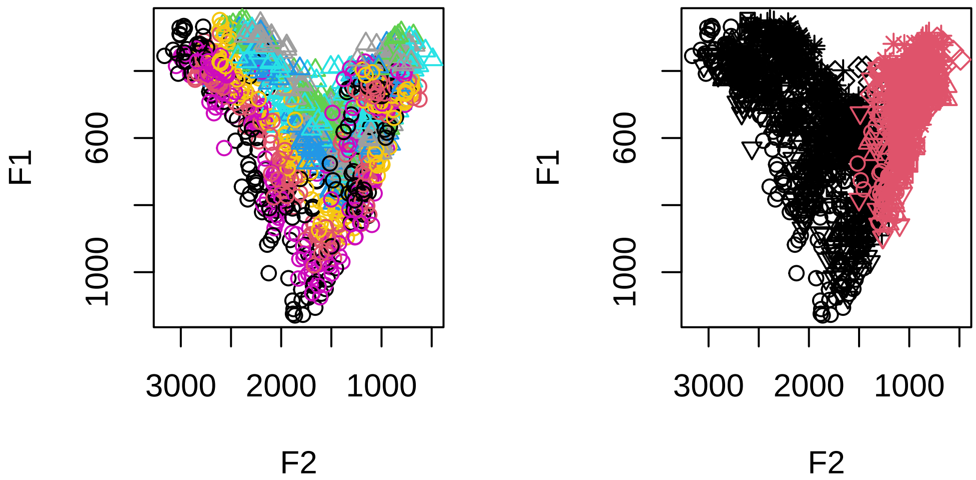
Plotting Vowels using PhonR

```r
with(indo, plotVowels(f1, f2, var.sty.by = vowel, var.col.by = gender))
```



Plotting Vowels using PhonR

```
par(mfrow = c(1, 2))
rounded <- ifelse(indo$vowel %in% c("o", "u"), "round", "unround")
with(indo, plotVowels(f1, f2, var.sty.by = gender, var.col.by = subj))
with(indo, plotVowels(f1, f2, var.sty.by = subj, var.col.by = rounded))
```



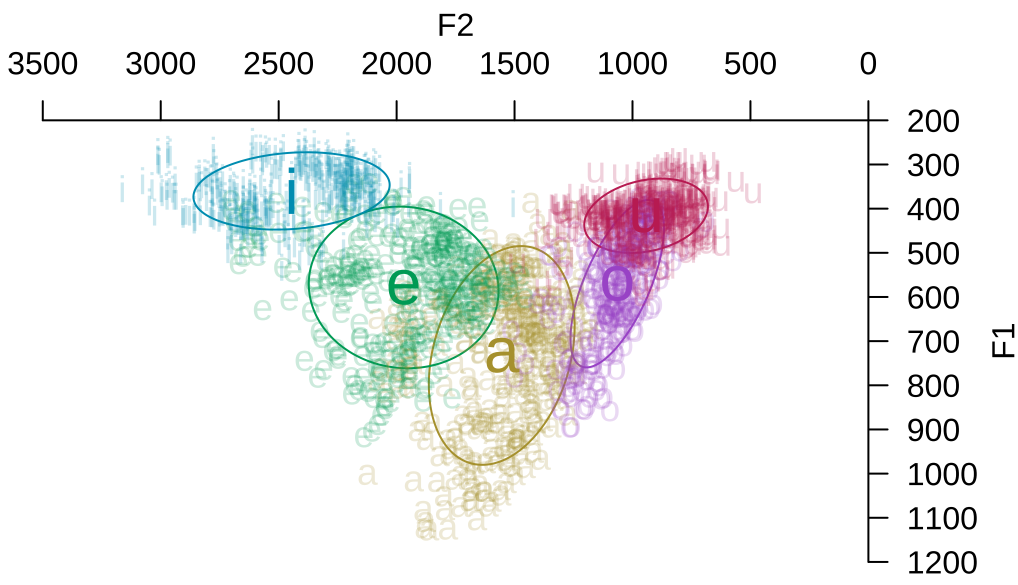
Calculating vowel space areas

```
poly.area <- with(indo, vowelMeansPolygonArea(f1, f2, vowel, poly.order = c("i",
    "e", "a", "o", "u"), group = subj))
hull.area <- with(indo, convexHullArea(f1, f2, group = subj))
rbind(poly.area, hull.area)
```

```
                F02      F04     F08      F09      M01      M02      M03
poly.area  485051.4 337364.0  434816 302064.9 197746.1 229501.7 215713.3
hull.area 1254575.0 866109.5 1020835 751327.0 517212.5 666246.0 477518.5
               M04
poly.area 177131.1
hull.area 568364.0
```

Ellipses, polygons, and hulls

```
#par(mfrow = c(2, 2))
with(indo, plotVowels(f1, f2, vowel, plot.tokens = TRUE, pch.tokens = vowel, cex.tokens = 1.
    alpha.tokens = 0.2, plot.means = TRUE, pch.means = vowel, cex.means = 2, var.col.by = vo
    ellipse.line = TRUE, pretty = TRUE))
```



Normalizing data

- Speaker vocal tracts are variable - different lengths and cross-sections
- Implies variable resonances
- $F_n = \frac{(2n-1)c}{4L}$, for a tube that is open at one end and closed in the other

Minimizing variation

- In order to minimize the variation brought about by the variable vocal tract parameters, often we do a type of normalization that we call z-score normalization

---

Z-Score Normalization

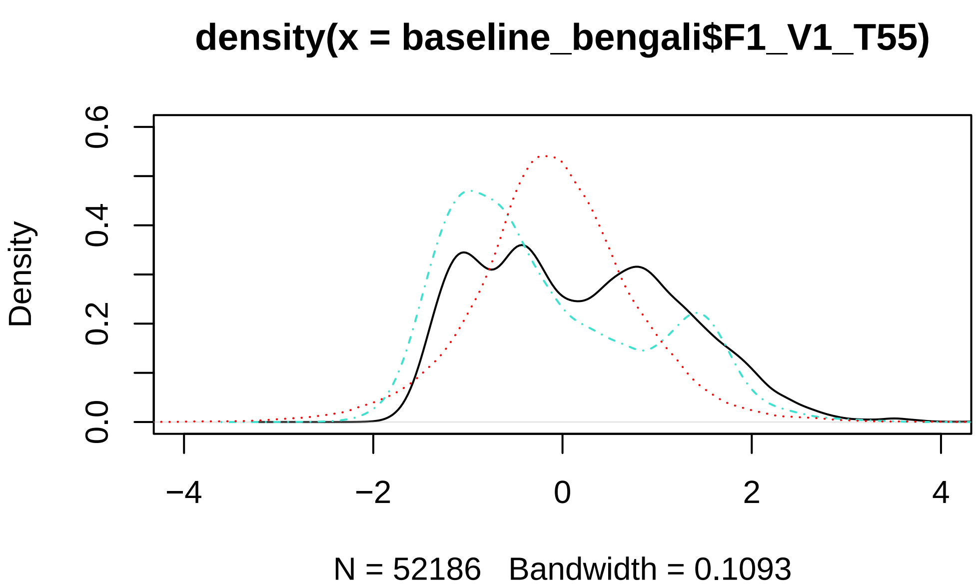
- This serves two purposes
  1. Allows us to reduce individual differences (between subjects)
  2. Makes data comparable
- Z-Score normalization
- $z = \frac{x_i - \overline{x}}{\sigma}$
- Where $\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2}$

Error in estimating population parameters

- Two major sources of errors

1. The underlying distribution
2. The number of samples

- $SE = \frac{\sigma}{\sqrt{n}}$



References

Johnson, K. 2008. *Quantitative Methods in Linguistics*. Wiley. https://books.google.co.in/books?id=kJJpAAAAMAAJ.

Winter, B. 2020. *Statistics for Linguists: An Introduction Using r*. Routledge. https://books.google.co.in/books?id=IXhpxQEACAAJ.