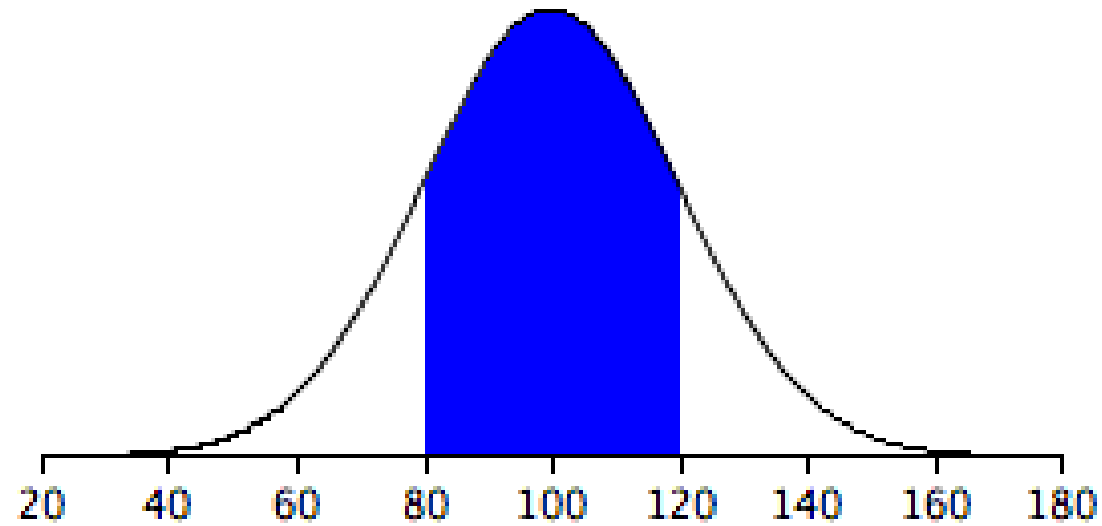


Data and sampling distributions



Overview

Very quick review

Probability functions

- Generating random numbers
- Probability density functions
- Cumulative distribution functions

Sampling distributions

Where we are in the plan for the semester

- | | | | <u>Analysis</u> | <u>R</u> |
|---|-----------|---|-----------------|----------|
| 1 | Sep 2 | Course overview, introduction to R, descriptive statistics | | |
| 2 | Sep 7-9 | Review of central statistical concepts and exploratory analysis using R | | |
| 3 | Sep 14-16 | Confidence Intervals and the bootstrap | | |
| 4 | Sep 21-23 | Review of hypothesis tests and permutation tests in R | | |
| 5 | Sep 28-30 | Parametric, non-parametric and theories of hypothesis testing | | |






resampling
methods

base R



Where we are in the plan for the semester

How would describe the pace of the class so far?

Way too slow	1 respondent	1 %	
Too slow	4 respondents	5 %	
About right	60 respondents	69 %	
Too fast	21 respondents	24 %	
Way too fast	1 respondent	1 %	

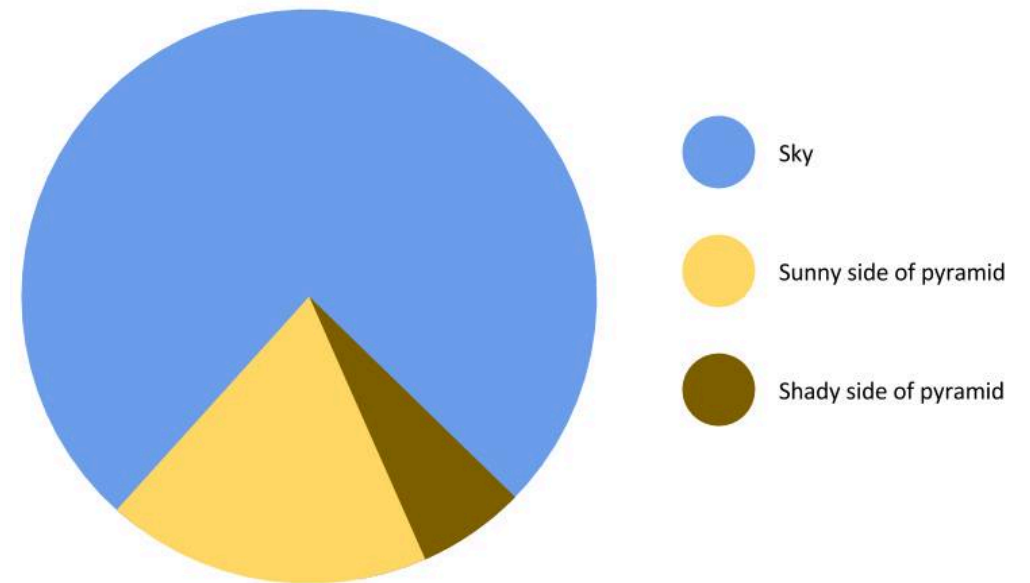
Quick review

Basics of R

```
> my_vec <- c(5, 28, 19)
> my_booleans <- c(TRUE, FALSE, TRUE)
> my_vec[my_booleans]
```

How to plot categorical data

```
> drinks_table <- table(profiles$drinks)
> barplot(drinks_table)
> pie(drinks_table)
```

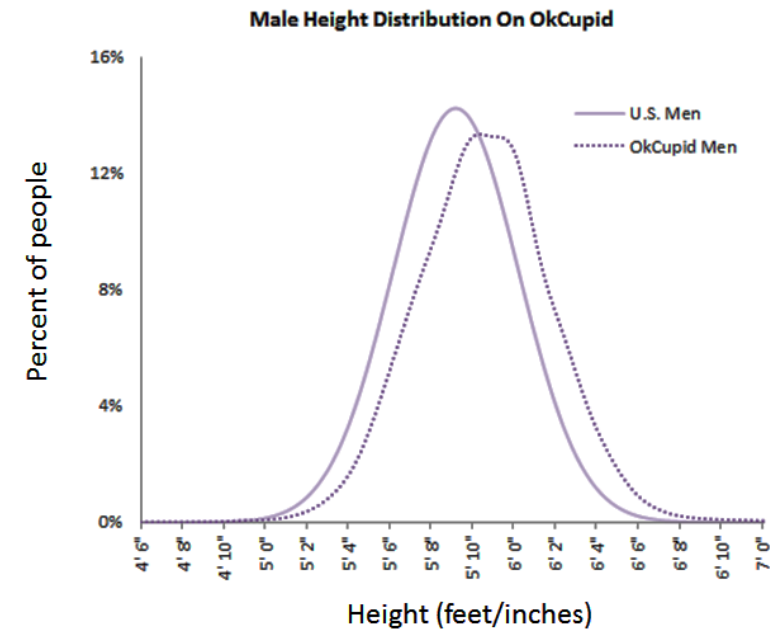
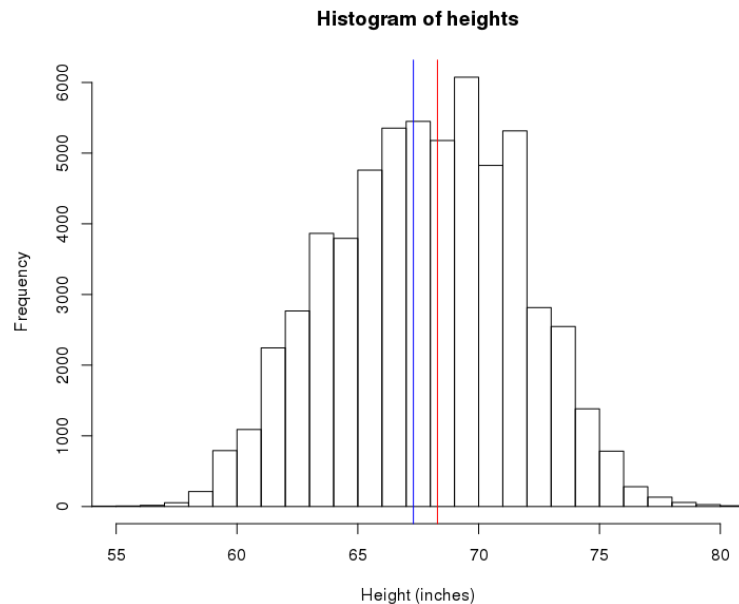


Quick review

How to plot quantitative data:

```
> hist(profiles$height)
```

```
> abline(v = 67)
```



Quick review

For loops

```
my_results <- NULL    # create an empty vector to store the results
for (i in 1:100) {
  my_results[i] <- i^2
}
```

Staying organized

It is useful to create separate folders for different homework and even for the different pieces of class code.

Be sure to set your working directory properly so that R can find the relevant files.

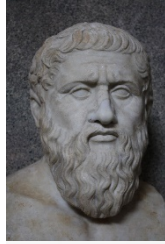


Questions?

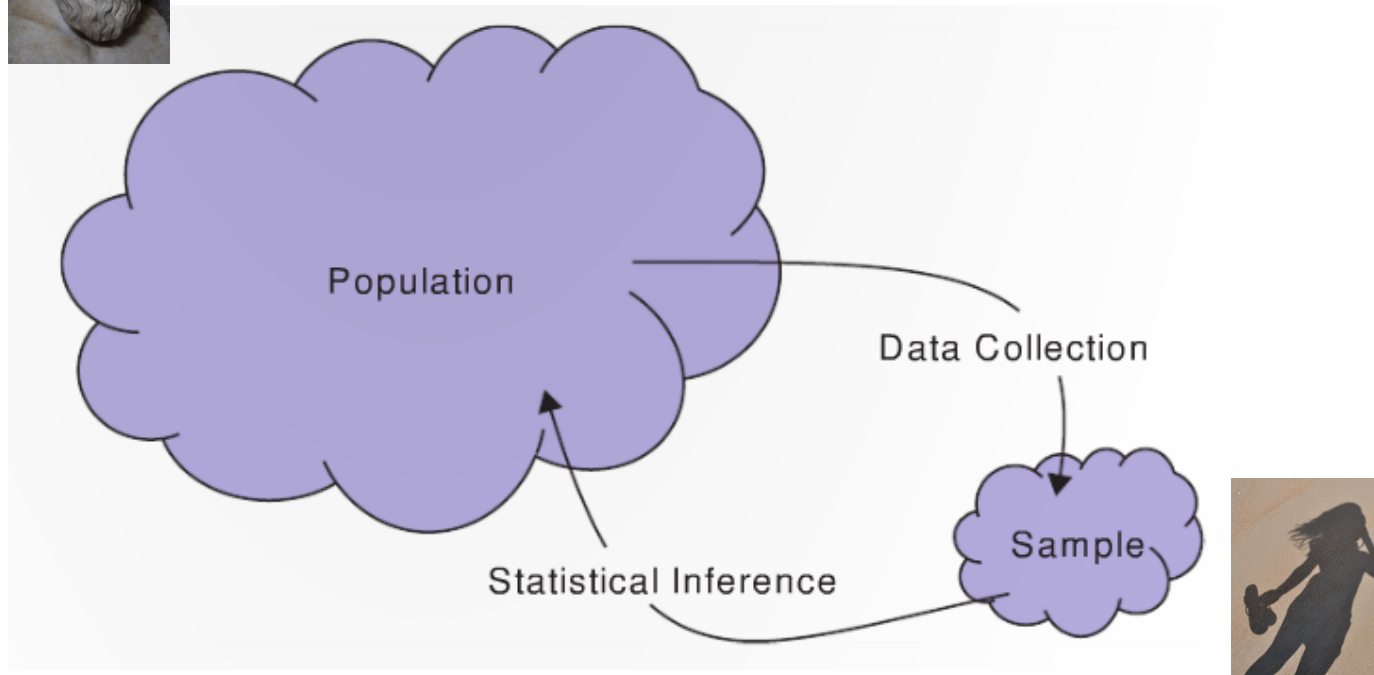


Review and extension of statistical concepts

Where does data come from?



Population: all individuals/objects of interest

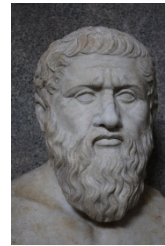


Sample: A subset of the population

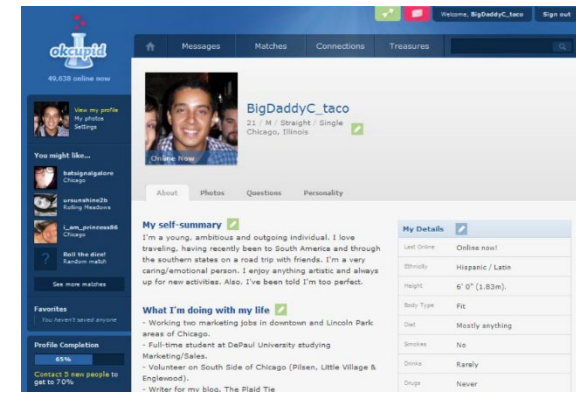
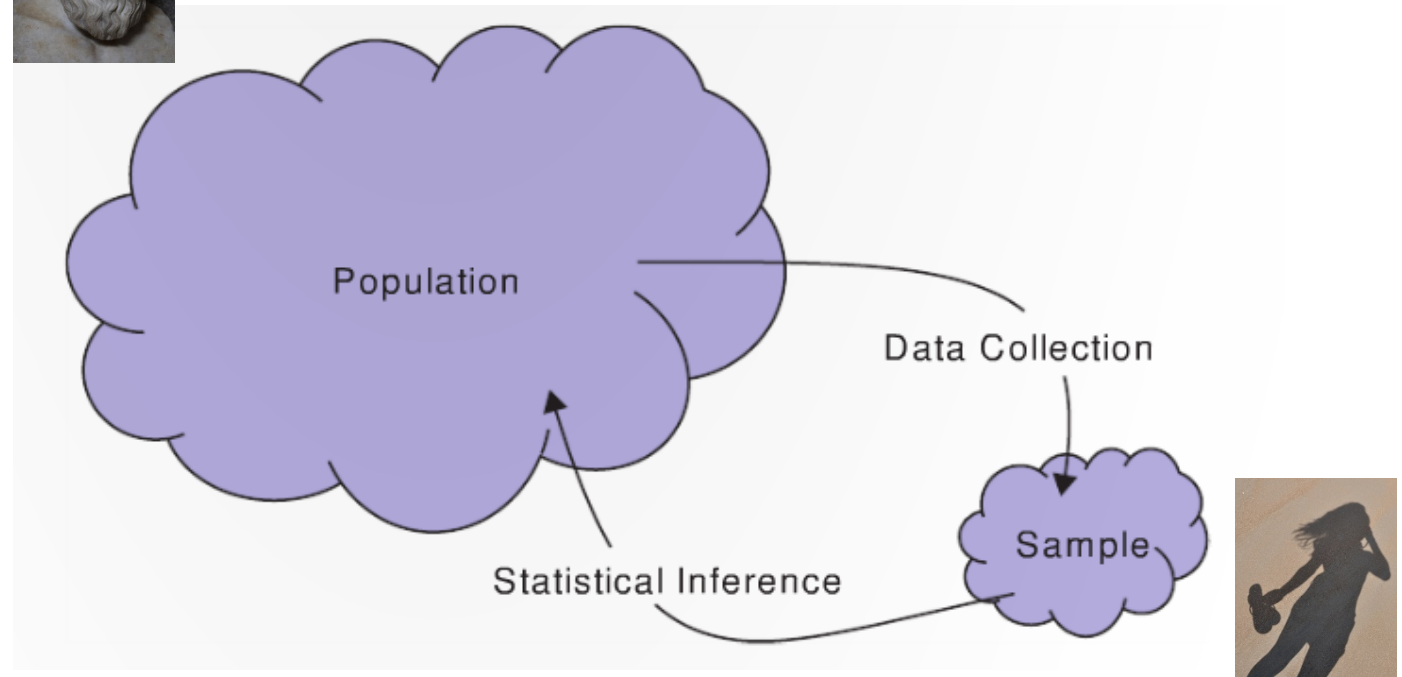
Where does data come from?

Question: Is the okcupid profiles data frame a population or a sample?

Question: If the OkCupid profiles data frame is a sample, what is the population?



Parameters: $\pi, \mu, \sigma, \rho, \beta$



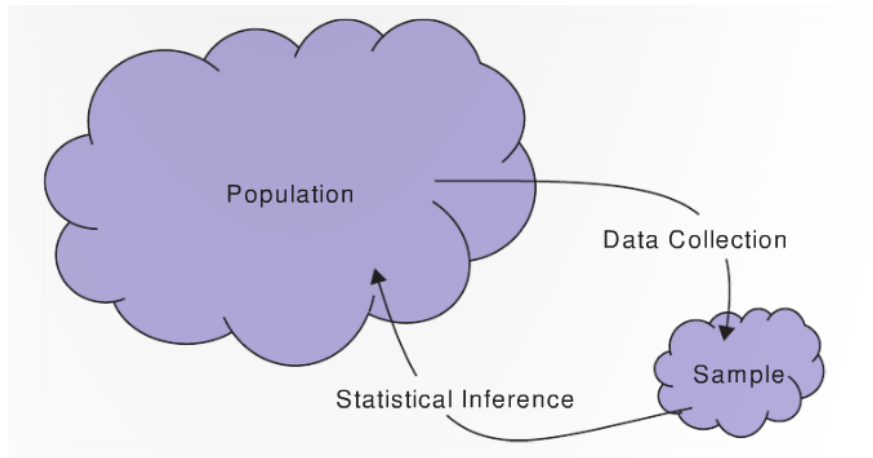
Statistics: $\hat{p}, \bar{x}, s, r, b$

How do we get sample of data?

Simple random sample: each member in the population is equally likely to be in the sample

“Random selection”

Q: Why is this good?



Questions:

- Is the OkCupid profiles data a simple random sample?
- Would we expect sampling bias from statistics computed from the OkCupid profiles?

Generating random data and probability models

To understand our data, it is often useful to be able to:

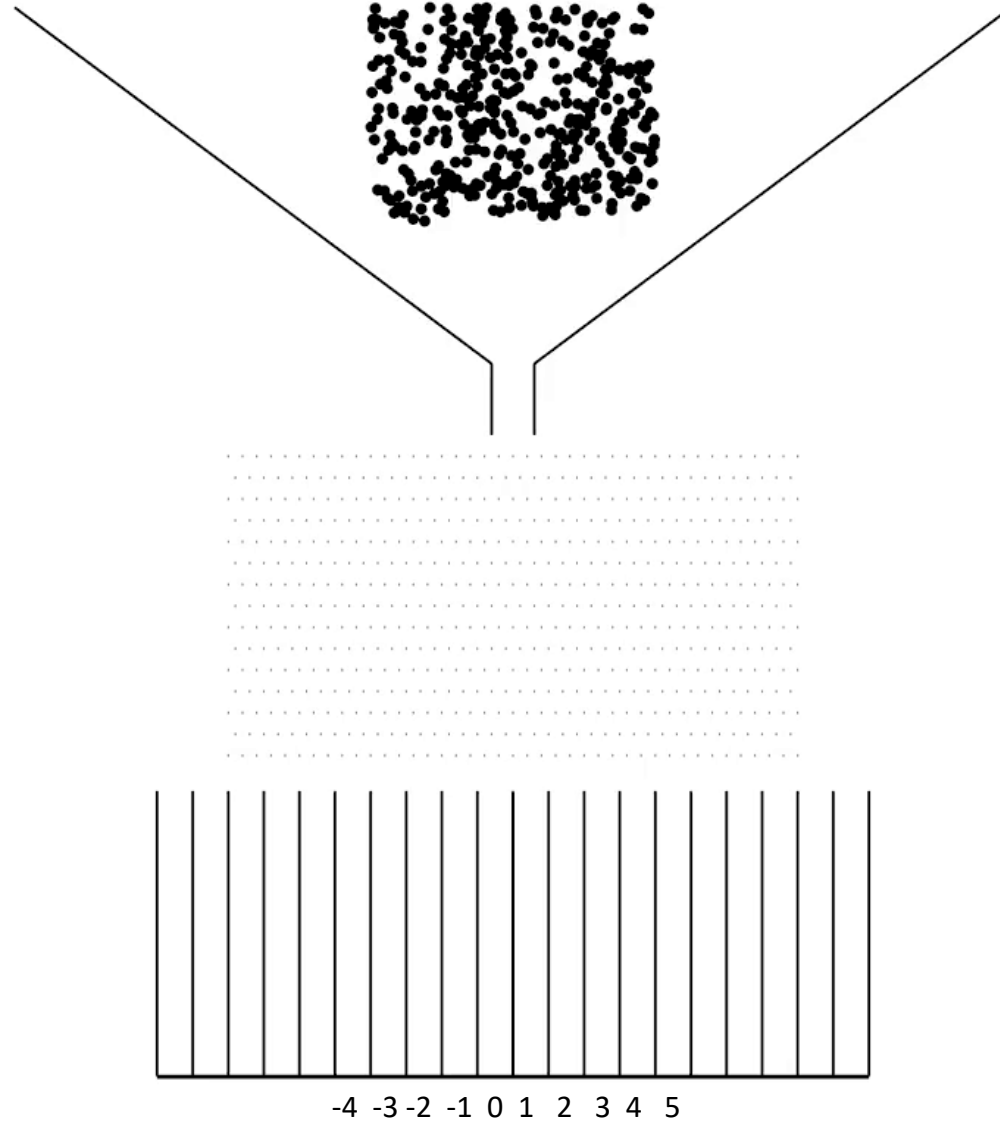
1. Simulate data in a way that replicates key properties of the data
2. Create mathematical (probability) models of our data

Generating random data and probability models

To understand our data, it is often useful to be able to:

1. Simulate data in a way that replicates key properties of the data
2. Create mathematical (probability) models of our data

Generating random data



Generating random data

R has built in functions to generate data from different distributions

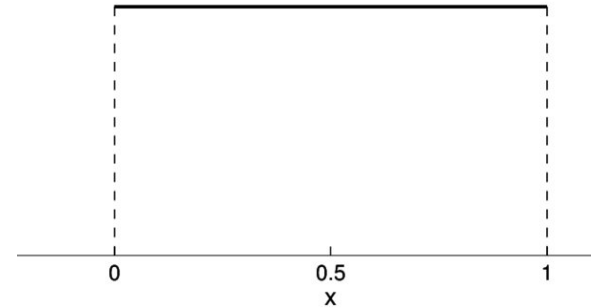
- All these functions start with the letter *r*

The uniform distribution

generate $n = 100$ points from $U(0, 1)$

```
> rand_data <- runif(100)
```

```
> hist(rand_data)
```

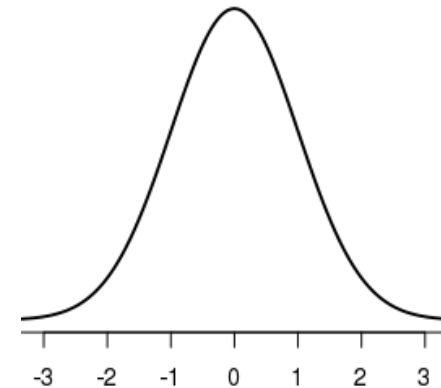


The normal distribution

generate $n = 100$ points from $N(0, 1)$

```
> rand_data <- rnorm(100)
```

```
> hist(rand_data)
```



Generating random data

R has built in functions to generate data from different distributions

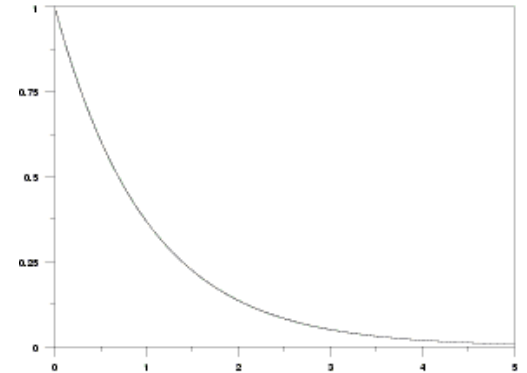
- All these functions start with the letter *r*

The exponential distribution

generate n = 100 points from `exponential($\lambda = 1$)`

> Homework 2

>

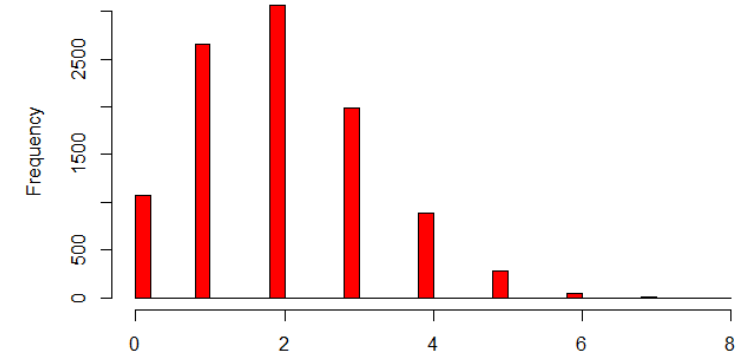


The binomial distribution

generate n = 100 points from `binomial(n = 8, $\pi = .2$)`

> `rand_data <- rbinom(100, 8, .2)`

> `hist(rand_data)`



Generating random data

If we want the same sequence of random numbers we can set the random number generating seed

```
> set.seed(123)
```

```
> runif(100)
```

Q: Why would we want the same sequence of random number?

A: Reproducibility!

Generating random data and probability models

To understand our data, it is often useful to be able to:

1. Simulate data in a way that replicates key properties of the data
2. Create mathematical (probability) models of our data

Density Curves

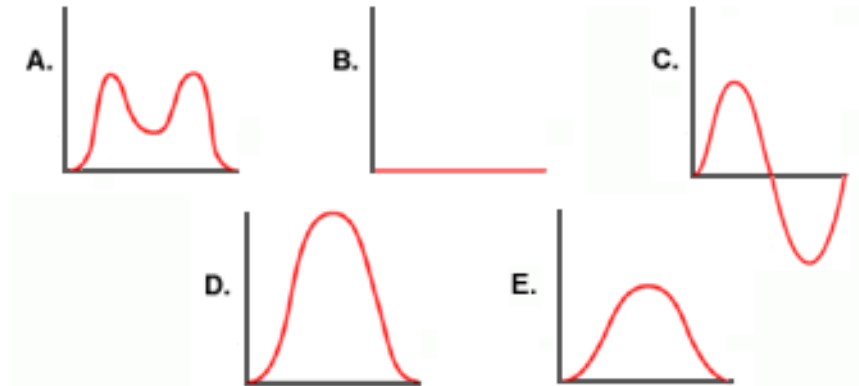
A **density curve** is a mathematical function $f(x)$ that can be used to model data

- We can imagine density curves as histograms that have:
 - Infinitely large data sample
 - With infinitely small bins sizes
 - Normalized to have an area of 1

Density curves have two defining properties:

1. The total area under the curve $f(x)$ is equal to 1
2. The curve is always ≥ 0

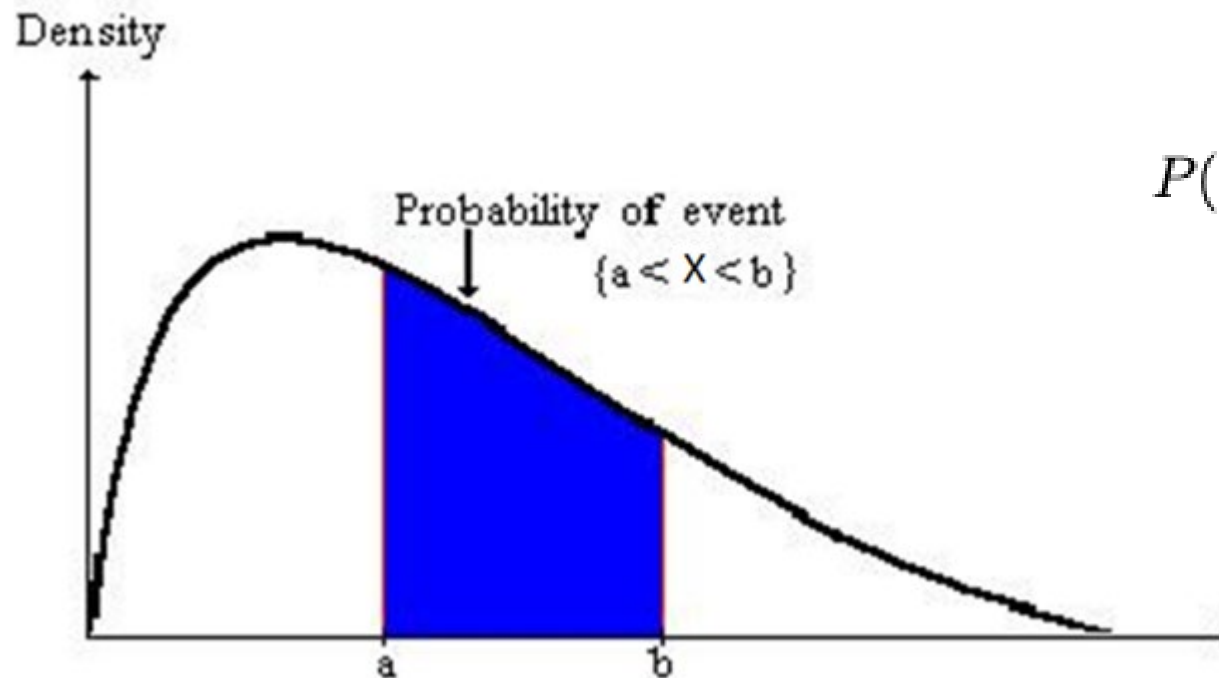
Which of these could **not** be a density curve?



Density Curves

The area under the density curve in an interval $[a, b]$ models the probability that a random number X will be in the interval

$\Pr(a < X < b)$ is the area under the curve from a to b



$$P(a < X < b) = \int_a^b f(x)dx$$

Examples of density curves

R has built in functions to create density curves

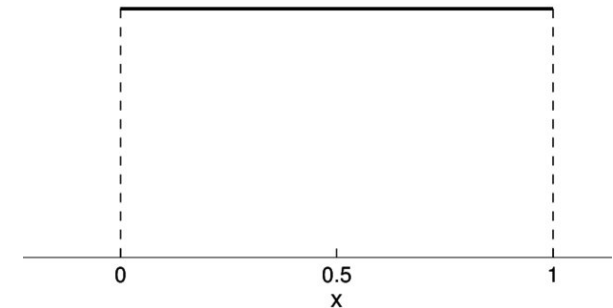
- All these functions start with the letter **d**

The uniform distribution

- (here $b = 1$, $a = 0$)

```
> x <- seq(-.2, 1.2, by = .001)
> y <- dunif(x)
> plot(x, y, type = "l")
```

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b, \\ 0 & \text{for } x < a \text{ or } x > b \end{cases}$$

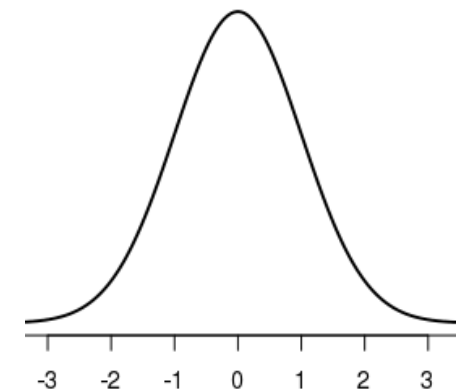


The normal distribution

- (here $\mu = 0$, $\sigma = 1$)

```
> x <- seq(-3, 3, by = .001)
> y <- dnorm(x)
> plot(x, y, type = "l")
```

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



Examples of density curves

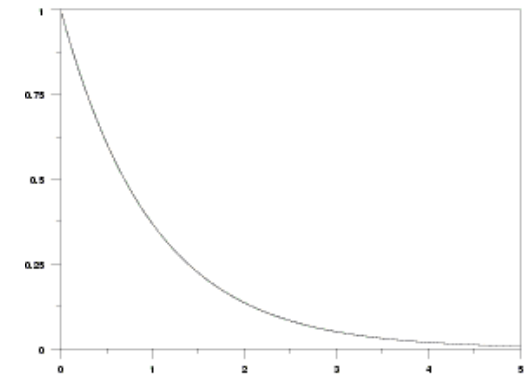
R has built in functions to create density curves

- All these functions start with the letter **d**

The exponential distribution

```
> Homework 2  
>  
>
```

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & x \geq 0, \\ 0 & x < 0. \end{cases}$$

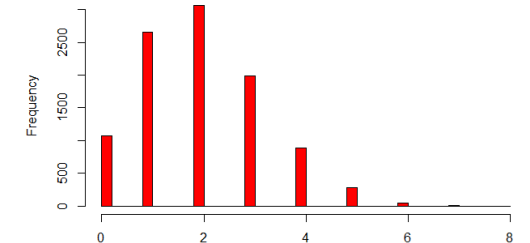


The binomial distribution

- (actually a probability mass function)

```
> x <- 0:8  
> y <- dbinom(x, 8, .2)  
> names(y) <- x  
> barplot(y)
```

$$f(k, n, p) = \binom{n}{k} p^k (1 - p)^{n-k}$$

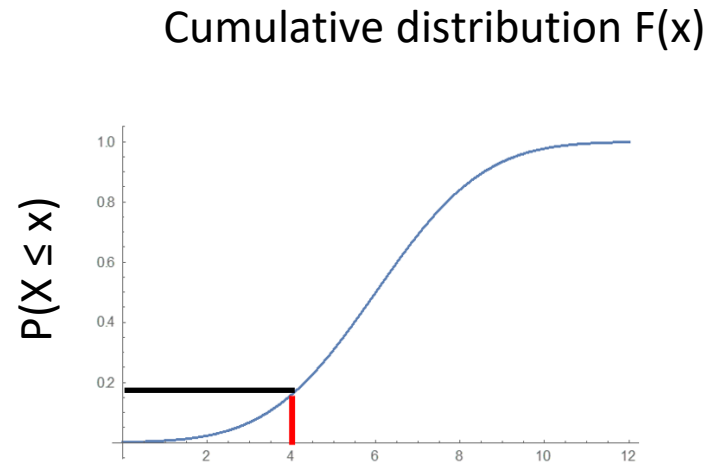
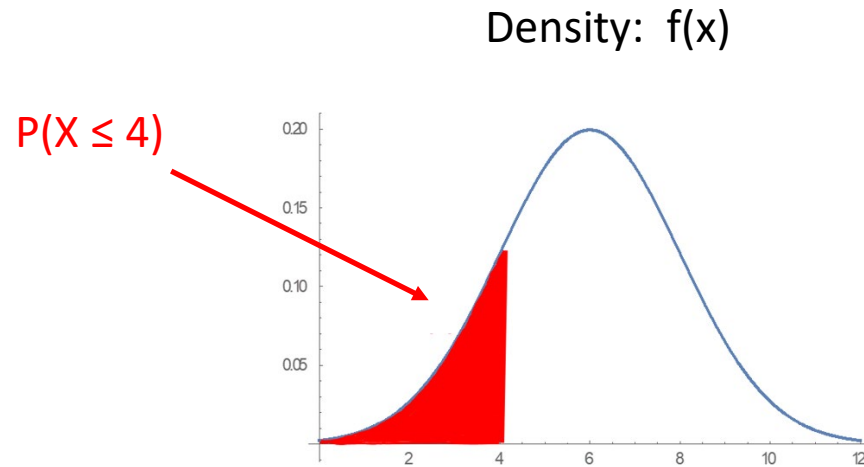


Cumulative distribution functions

Cumulative distribution functions give the probability of getting a random value X less than or equal to a value x : $P(X \leq x)$

- For example, we would write the probability of getting a random number X less than 2 as: $P(X \leq 2)$

Cumulative distribution functions are obtained by calculating the area under a probability density function



$$P(X \leq x)$$

$$= F(x)$$

$$= \int_{-\infty}^x f(x) dx$$

Examples of cumulative distributions in R

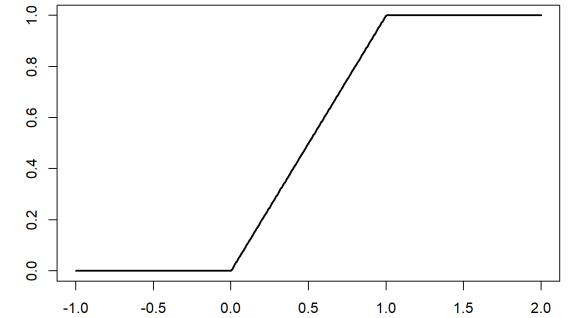
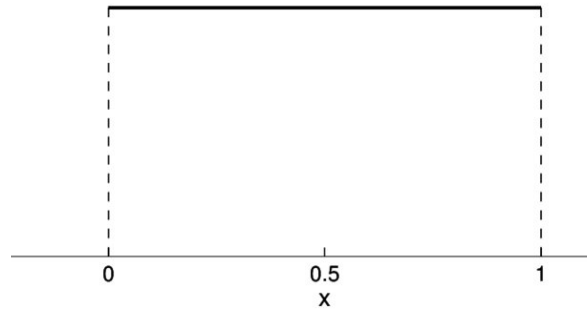
R has built in functions to get probabilities from different distributions

- All these functions start with the letter *p*

The uniform distribution

$P(X \leq .25)$

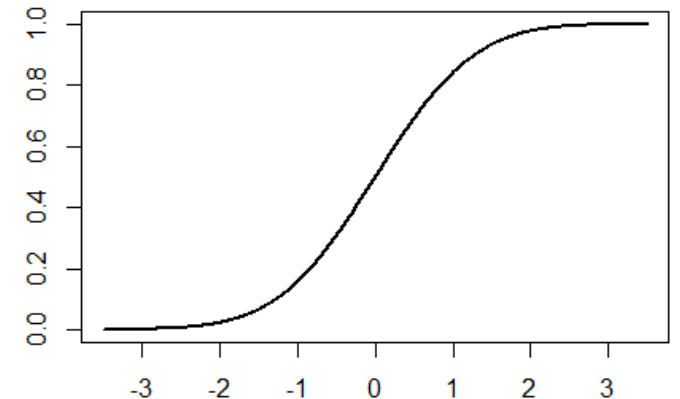
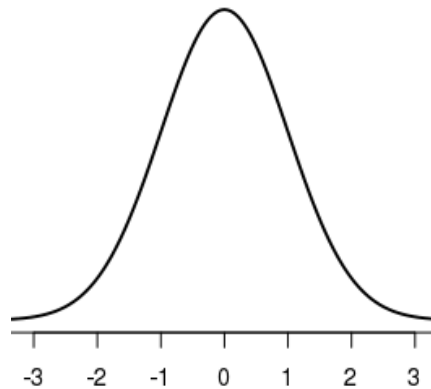
`dunif(.25)`



The normal distribution

$P(X \leq 2)$

`dnorm(2)`



Examples of cumulative distributions in R

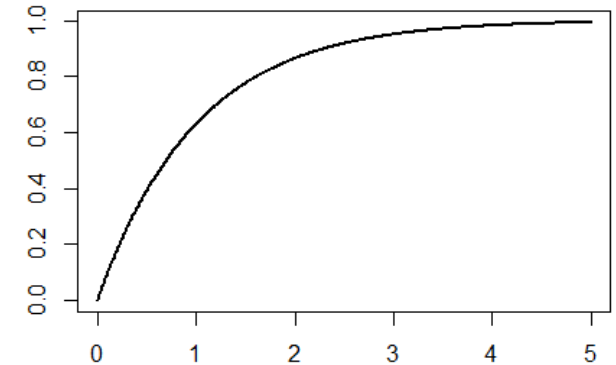
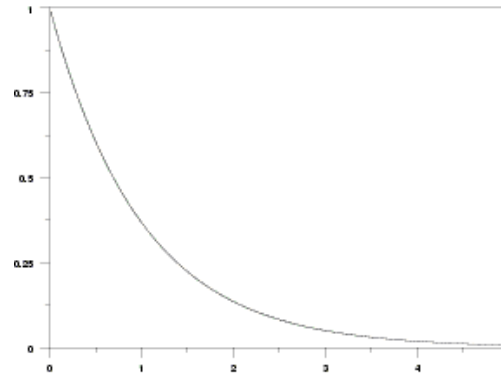
R has built in functions to get probabilities from different distributions

- All these functions start with the letter ***p***

The exponential distribution

$P(X \leq 2)$

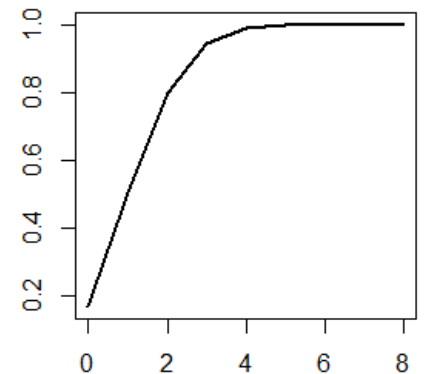
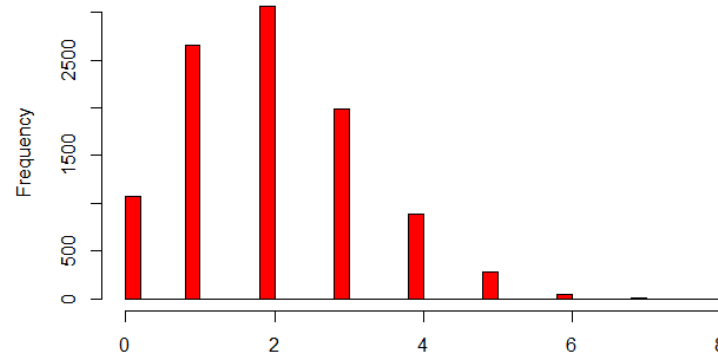
`dexp(2)`



The binomial distribution

$P(X \leq 2; n = 8, \pi = .2)$

`dbinom(2, 8, .2)`



Sampling distributions

Sample statistics

Q: What is a statistic?

The sample mean \bar{x}

(shadow of the parameter μ)

```
> rand_data <- runif(100) # generate n = 100 points from U(0, 1)  
> mean(rand_data)
```

Q: If we repeat the code above will we get the same statistic?

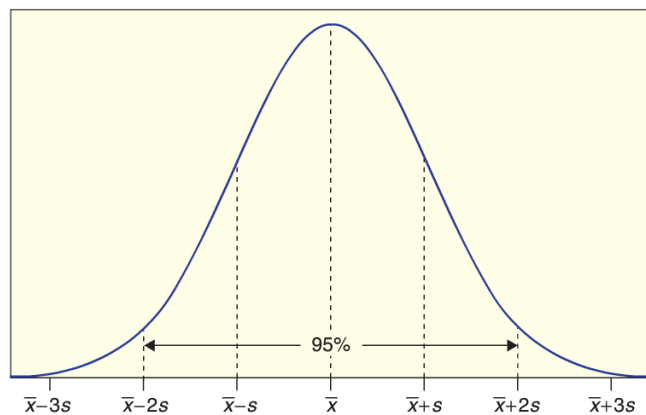
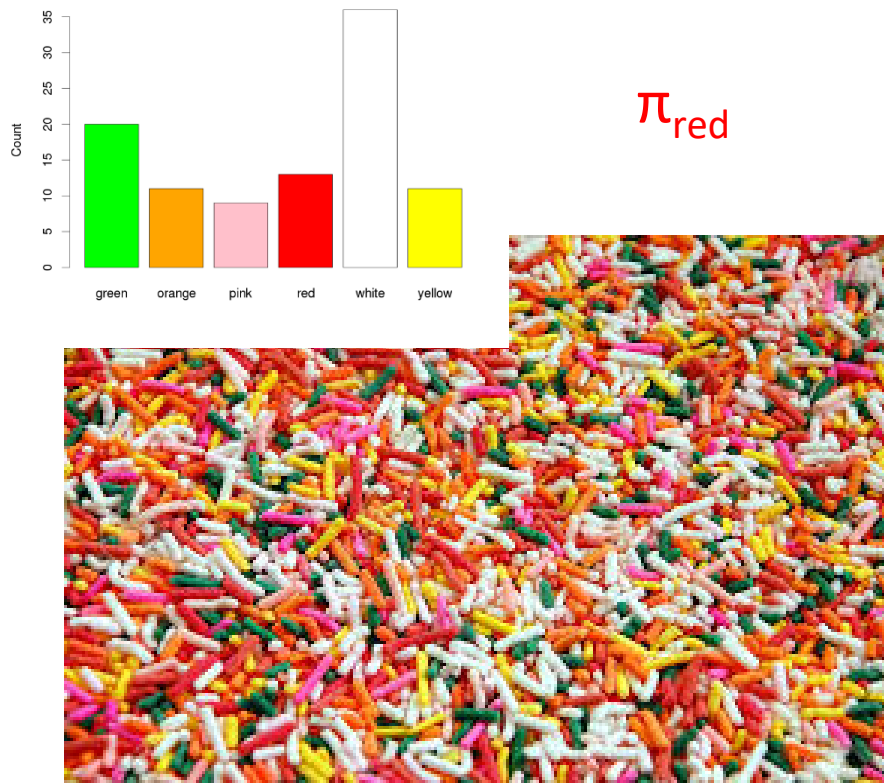
Sampling distributions

A ***sampling distribution*** is a distribution of ***statistics***

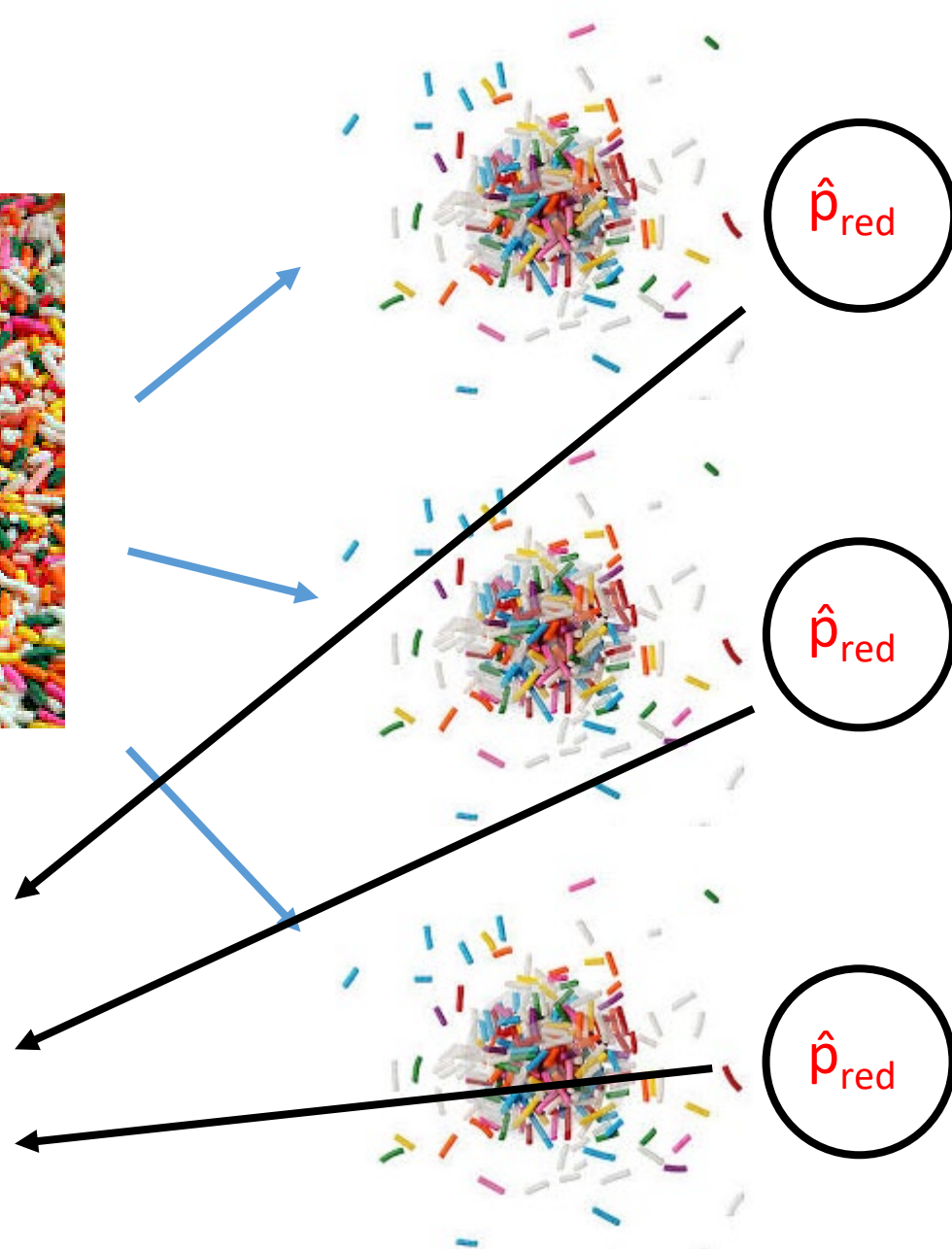
Reminder: For a *single ***categorical variable****, the main statistic of interest is the ***proportion*** (\hat{p}) in each category

- (shadow of the parameter π)

$$\hat{p} = \text{Proportion in a category} = \frac{\text{number in that category}}{\text{total number}}$$

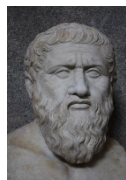


Sampling distribution!

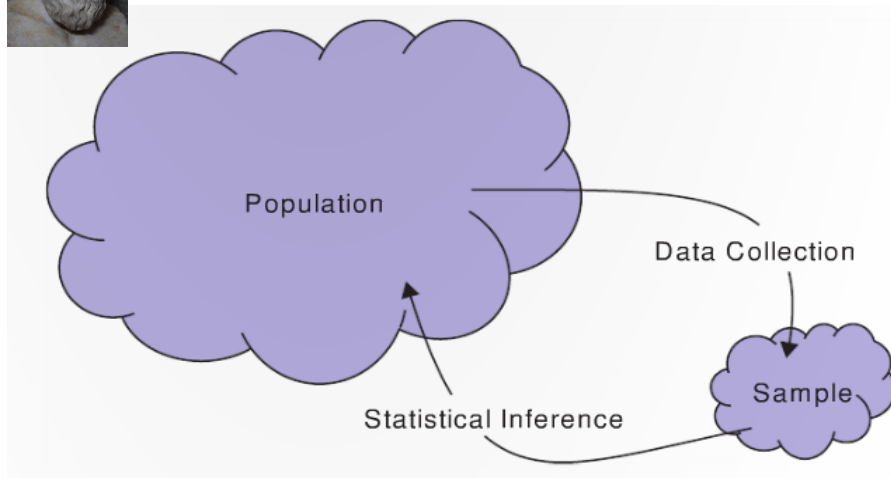


Sampling distribution

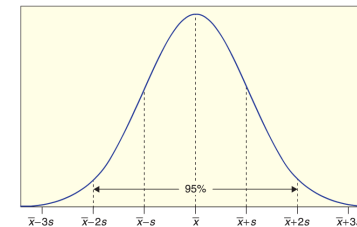
Why would we be interested in the sampling distribution?



Parameters: $\pi, \mu, \sigma, \rho, \beta$



Sampling distribution



Statistics: $\hat{p}, \bar{x}, s, r, b$

Sampling distributions

```
sampling_dist <- NULL
for (i in 1:1000) {
    rand_data <- runif(100)  # generate n = 100 points from U(0, 1)
    sampling_dist[i] <- mean(rand_data)  # save the mean
}

hist(sampling_dist)
```

Sampling distributions

Distribution of OkCupid user's heights $n = 100$

```
heights <- profiles$height
```

```
# get one random sample of heights from 100 people
```

```
height_sample <- sample(heights, 100)
```

```
# get the mean of this sample
```

```
mean(height_sample)
```

Sampling distributions

Distribution of OkCupid user's heights $n = 100$

```
sampling_dist <- NULL
for (i in 1:1000) {
  height_sample <- sample(heights, 100)  # sample 100 random heights
  sampling_dist[i] <- mean(height_sample) # save the mean
}

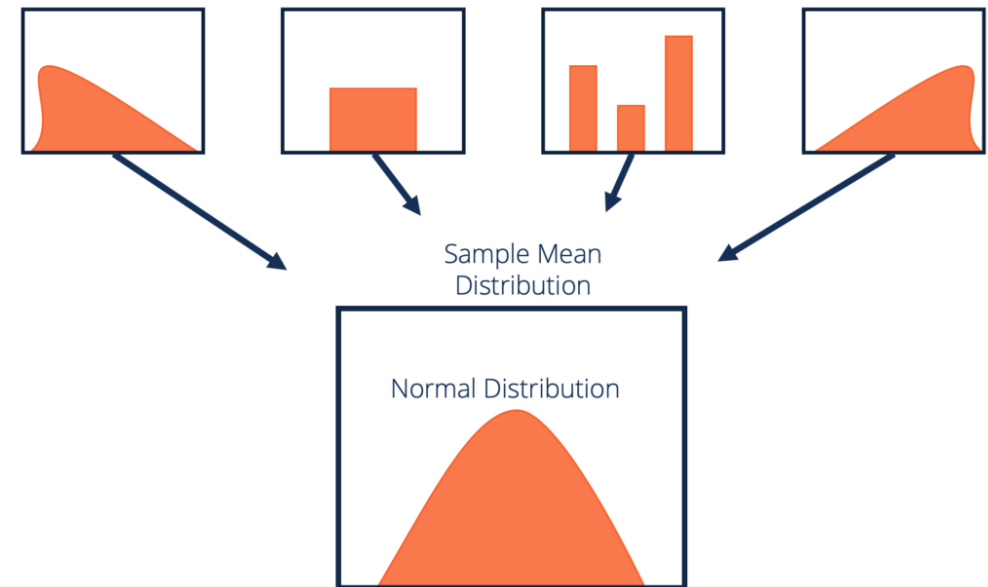
hist(sampling_dist)
```

The central limit theorem

The **central limit theorem** establishes that, in many situations, when independent random variables are summed up, their properly normalized sum tends toward a normal distribution.

Since many statistics we use are the sum of randomly data, many of our sampling distributions will be approximately normal

- You will explore this more on homework 2



Statistics: \hat{p} , \bar{x} , s , r , b