Lecture 4: Distributions

POLS3316, Instructor: Tom Hanna, Spring 2025, University of Houston

2026-02-09

Table of contents

1 Data Assumptions

1.1 Why do we use data?

- Purpose: analyzing data for causal inference (to begin to make statements about cause and effect - inferring causes)
- Complex and uncertain data requires that we make…

1.2 Assumptions about the data

- Because the world is complex, to make sense of unknowns we make assumptions about data
- The assumptions are useful approximations even when not precisely true
- We still need to check that the real data does not seriously violate the assumptions

1.3 Data Assumptions: Random, Independent, and Identically Distributed

- Randomness and independence matter as assumptions about data
- Specifically, these are assumptions about the *Data Generating Process* or DGP
- The Data Generating Process: the way the world produces the data

1.4 The Data Generating Process

- The source of the data matters - the DGP matters
- Previously stated: Data comes from a random world
- So the DGP is random

1.5 Independence and Distribution

- Events in the data are *independent and identically distributed* - the IID assumption

1.6 Statistical Independence

- Independence is statistical independence - the outcome of one event does not affect our belief about the probability of another event
  - `If we draw a number from a hat, then flip a coin, the hat draw does not affect the v`
  - `X does not affect Y - the outcome of X does not affect our belief about the probabil`

1.7 What if X does affect Y?

If X does affect Y

- begin to infer a causal relationship
- possibly through one or more additional variables
- in some direction
- not necessarily that X causes Y
- *not quite accurate* summary "correlation does not imply causation"
- Better: *Correlation does not prove causation*

1.8 Identically Distributed

- Identically distributed: drawn from the same *probability distribution*

So...

2 Distributions

2.1 Introduction to distributions

- R has functions for at least 20 *probability distributions*
- The most important is the *normal distribution*
- This is because of the *central limit theorem*
- We will look at these in the most detail: *normal*, *binomial*, *uniform*, *poisson*
- Because these are *probability distributions* they allow us to determine the probability that events are random chance or not

2.2 Distribution examples

- The following are *histograms*
- They represent the *frequency* or simply the number count of observations for each value
- For example, if the value 4 shows 500, it means that 4 came up 500 times in the data
- The graphs were produced by generating random numbers based on the particular distribution with an R function

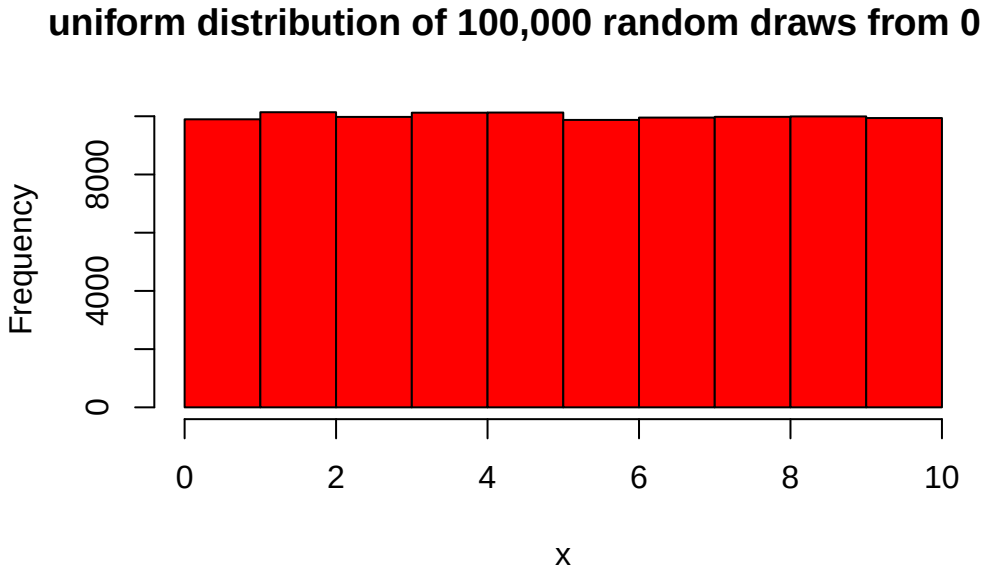
2.3 Uniform distribution

All outcomes are equally likely

2.4 Uniform distribution

All outcomes are equally likely

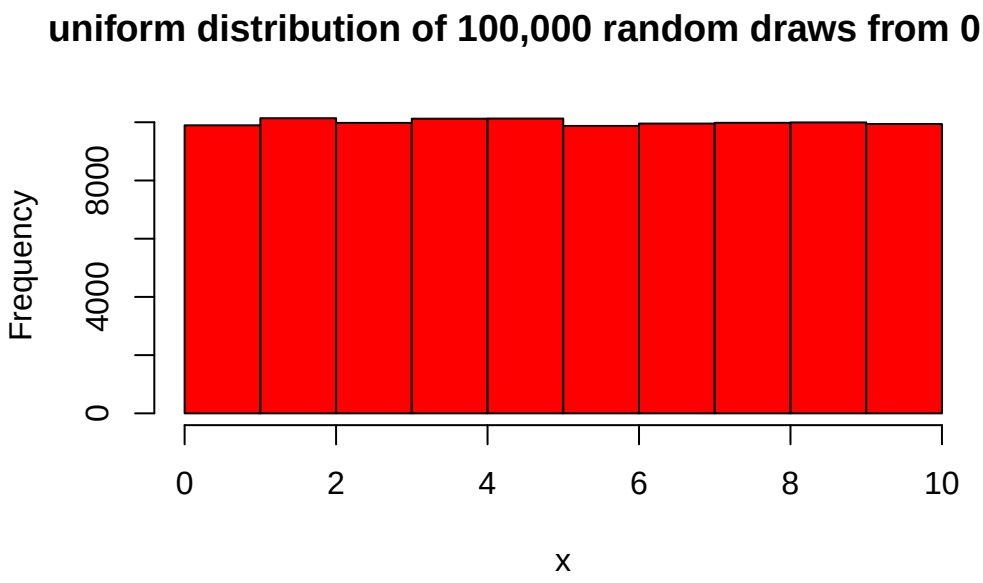
```
# Set a seed for reproducibility

set.seed(123)

rand.unif <- runif(100000, min = 0, max = 10)
hist(rand.unif, breaks = 10, freq = TRUE, main = "uniform distribution of 100,000 random dra
```



2.5 Uniform Distribution: Probability

- The probability of any frequency is 1/10
- Any deviation from a value of 10,000 is a random deviation from the expected value

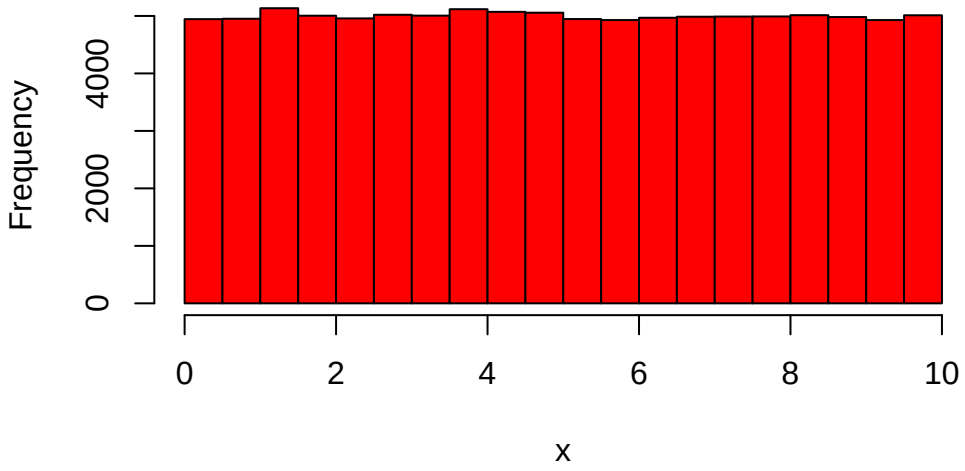
```
# Set a seed for reproducibility

set.seed(123)

rand.unif <- runif(100000, min = 0, max = 10)
hist(rand.unif, breaks = 10, freq = TRUE, main = "uniform distribution of 100,000 random draw
```



2.6 Uniform distribution: with code

All outcomes are equally likely

```
# Set a seed for reproducibility

set.seed(123)

rand.unif <- runif(100000, min = 0, max = 10)
hist(rand.unif, breaks = 20, freq = TRUE, main = "uniform distribution of 100,000 random draw
```

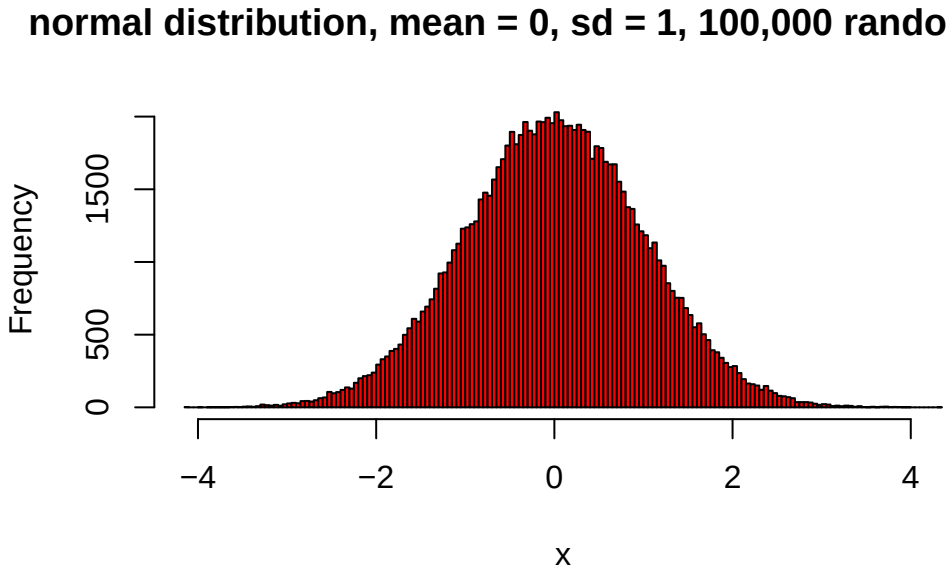


2.7 Normal Distribution

- *symmetrical around its mean with most values near the central peak*
- width is a function of the *standard deviation*
- Other names: *Gaussian distribution*, *bell curve*

2.8 Normal Distribution

```
set.seed(123)

rand.norm<- rnorm(100000)
hist(rand.norm, breaks = 200, freq = TRUE, main = "normal distribution, mean = 0, sd = 1, 10
```

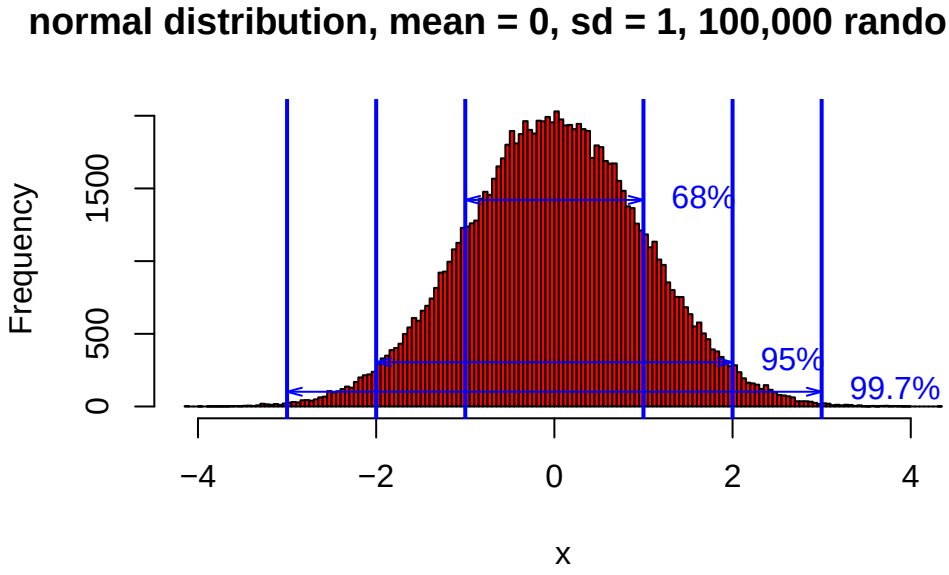


2.9 Normal Distribution: Probability

- The probability of a value is determined by how far it is from the mean in terms of standard deviations

```
set.seed(123)

# plot a normal distribution with mean = 0, sd = 1, 100,000 random draws, 200 breaks, and re
# add lines to the plot to illustrate the 68-95-99.7 rule

rand.norm<- rnorm(100000)
h <- hist(rand.norm, breaks = 200, freq = TRUE, main = "normal distribution, mean = 0, sd =
abline(v = c(-1, 1), col = "blue", lwd = 2)
abline(v = c(-2, 2), col = "blue", lwd = 2)
abline(v = c(-3, 3), col = "blue", lwd = 2)

# create labels to indicate the percentage within each range


# get max y from the histogram for label placement
ymax <- max(h$counts)

# add labels for the 68-95-99.7 rule

text(1.1, ymax*0.7, "68%",   pos = 4, col = "blue")
```

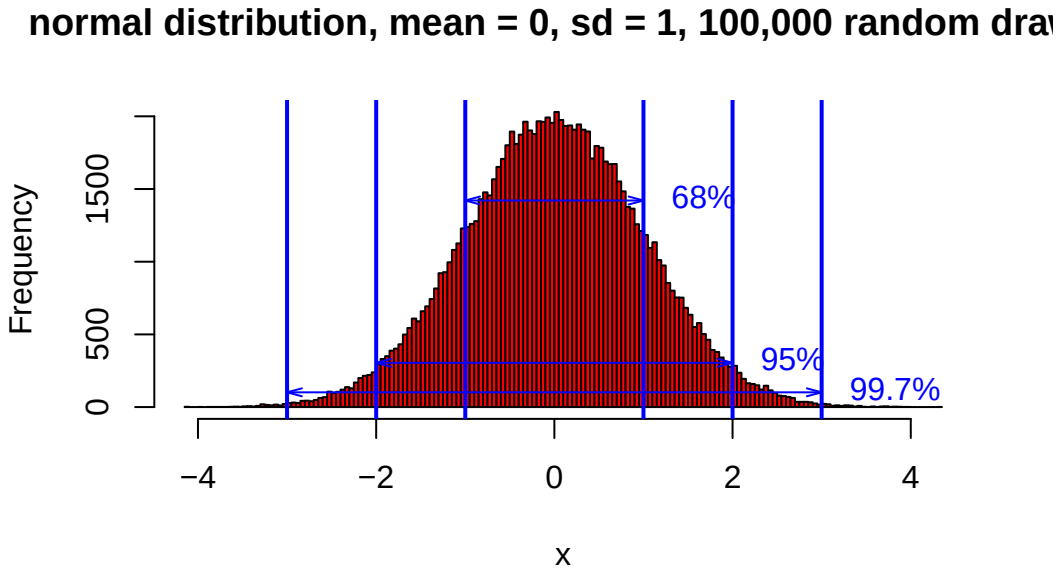
```
text(2.1, ymax*0.15, "95%",   pos = 4, col = "blue")
text(3.1, ymax*0.05, "99.7%", pos = 4, col = "blue")

# add arrows to illustrate the range of values within each standard deviation range

## arrows from +sd line to -sd line at same y-level
arrows(x0 = 1,  y0 = ymax*0.7, x1 = -1,  y1 = ymax*0.7,
       code = 3, angle = 15, length = 0.08, col = "blue")   # 68% range
arrows(x0 = 2,  y0 = ymax*0.15, x1 = -2,  y1 = ymax*0.15,
       code = 3, angle = 15, length = 0.08, col = "blue")   # 95% range
arrows(x0 = 3,  y0 = ymax*0.05, x1 = -3,  y1 = ymax*0.05,
       code = 3, angle = 15, length = 0.08, col = "blue")   # 99.7% range
```



2.10 Normal Distribution: with code and probabilities

```
set.seed(123)

# plot a normal distribution with mean = 0, sd = 1, 100,000 random draws, 200 breaks, and re
# add lines to the plot to illustrate the 68-95-99.7 rule

rand.norm<- rnorm(100000)
h <- hist(rand.norm, breaks = 200, freq = TRUE, main = "normal distribution, mean = 0, sd =
abline(v = c(-1, 1), col = "blue", lwd = 2)
```

```r
abline(v = c(-2, 2), col = "blue", lwd = 2)
abline(v = c(-3, 3), col = "blue", lwd = 2)

# create labels to indicate the percentage within each range


# get max y from the histogram for label placement
ymax <- max(h$counts)

# add labels for the 68-95-99.7 rule

text(1.1, ymax*0.7, "68%",   pos = 4, col = "blue")
text(2.1, ymax*0.15, "95%",   pos = 4, col = "blue")
text(3.1, ymax*0.05, "99.7%", pos = 4, col = "blue")

# add arrows to illustrate the range of values within each standard deviation range

## arrows from +sd line to -sd line at same y-level
arrows(x0 = 1,  y0 = ymax*0.7, x1 = -1,  y1 = ymax*0.7,
       code = 3, angle = 15, length = 0.08, col = "blue")   # 68% range
arrows(x0 = 2,  y0 = ymax*0.15, x1 = -2,  y1 = ymax*0.15,
       code = 3, angle = 15, length = 0.08, col = "blue")   # 95% range
arrows(x0 = 3,  y0 = ymax*0.05, x1 = -3,  y1 = ymax*0.05,
       code = 3, angle = 15, length = 0.08, col = "blue")   # 99.7% range
```



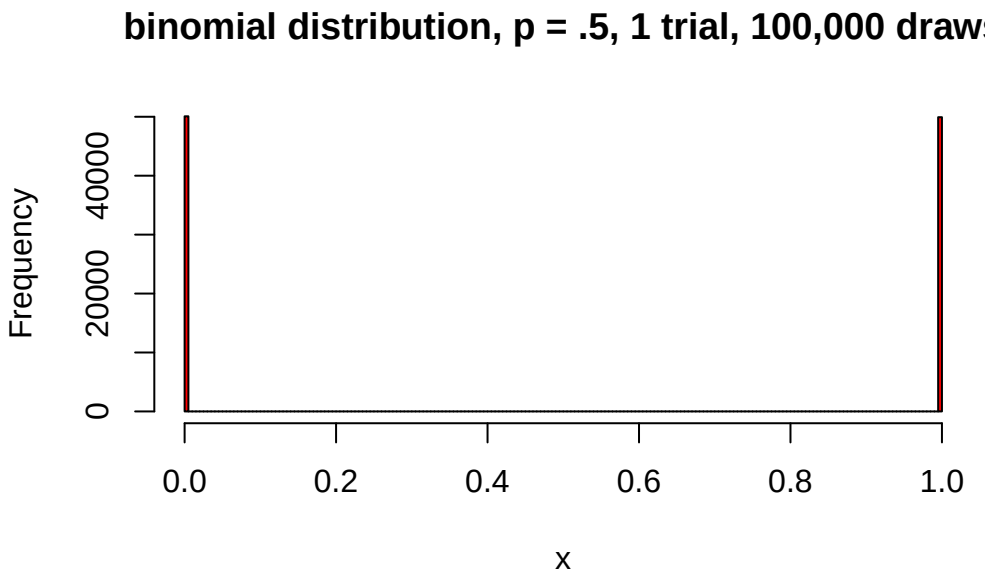
2.11 Binomial Distribution

- binary
- success/failure
- yes/no
- distribution for a number of Bernoulli trials

2.12 Binomial - Bernoulli example

- n = 1 makes this a Bernoulli distribution

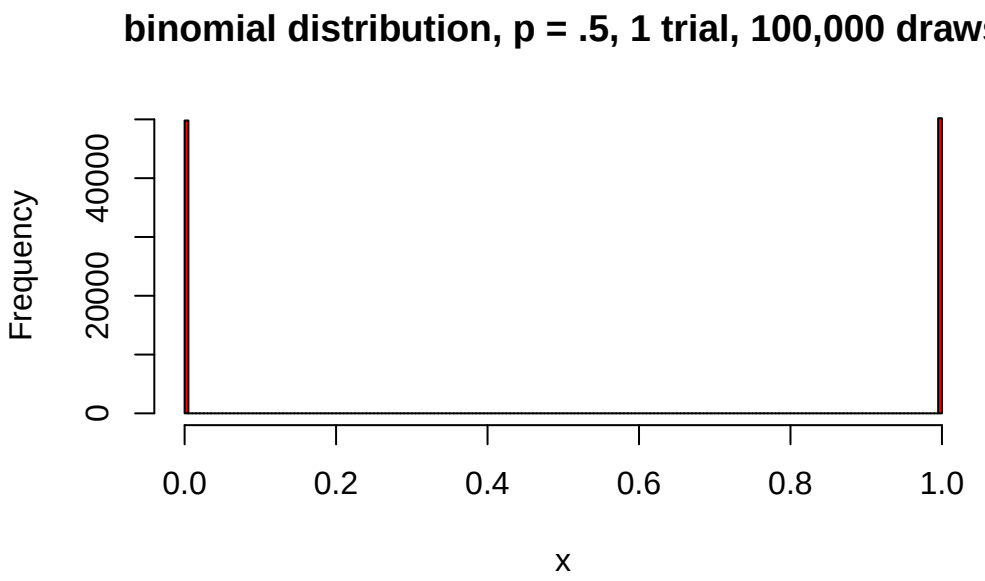
```
rand.binom<- rbinom(100000,1,.5)
hist(rand.binom, breaks = 200, freq = TRUE, main = "binomial distribution, p = .5, 1 trial, 1
```



2.13 Binomial example: with code

- n = 1 makes this a Bernoulli distribution

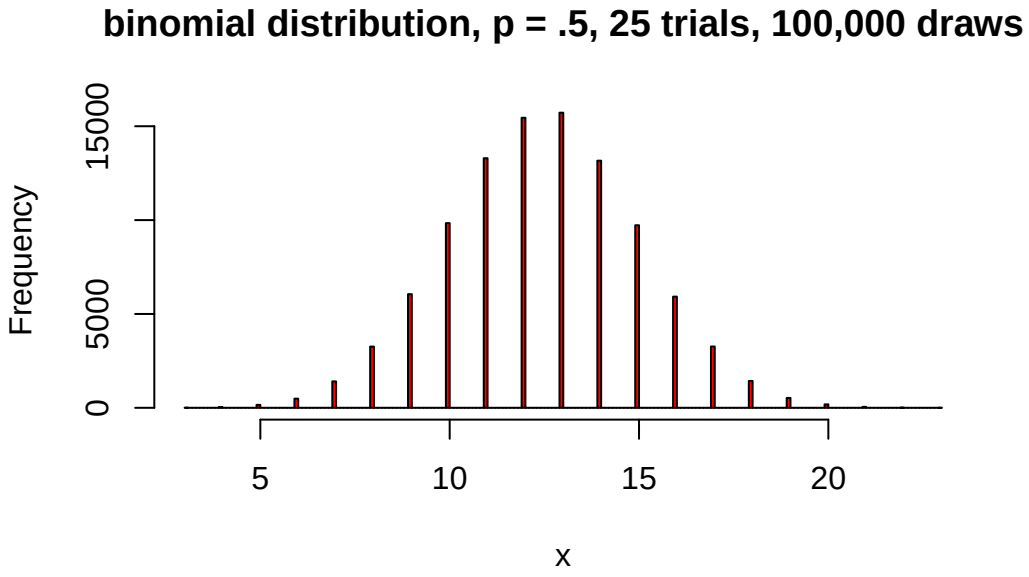
```
rand.binom<- rbinom(100000,1,.5)
hist(rand.binom, breaks = 200, freq = TRUE, main = "binomial distribution, p = .5, 1 trial, 1
```



2.14 Binomial example: with code

- trials = 25

```
rand.binom2 <- rbinom(100000,25,.5)
hist(rand.binom2, breaks = 200, freq = TRUE, main = "binomial distribution, p = .5, 25 trials
```

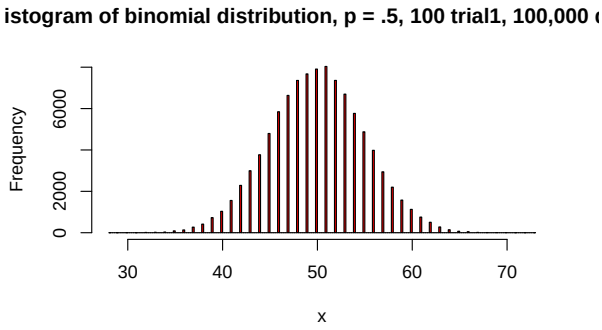


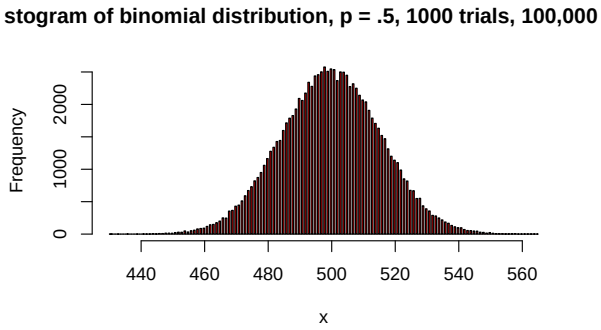
2.15 Preview of the Central Limit Theorem

What happens if we do the same thing above but do it 1,000 times and plot the counts?

2.16 Preview of the Central Limit Theorem

```
rand.binom3<- rbinom(100000,100,.5)
hist(rand.binom3, breaks = 200, freq = TRUE, main = "Histogram of binomial distribution, p =

rand.binom4<- rbinom(100000,1000,.5)
hist(rand.binom4, breaks = 200, freq = TRUE, main = "Histogram of binomial distribution, p =
```

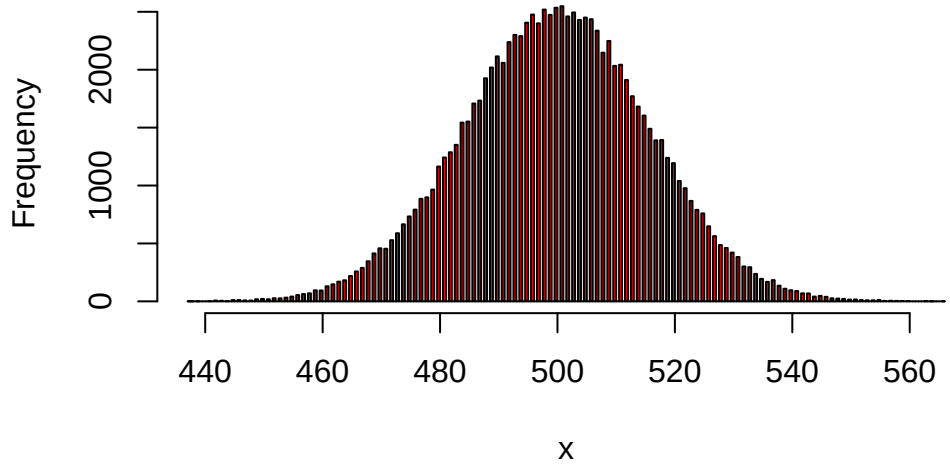




2.17 Preview of the Central Limit Theorem: code

```
rand.binom3<- rbinom(100000,1000,.5)
hist(rand.binom3, breaks = 200, freq = TRUE, main = "Histogram of binomial distribution, p =
```

stogram of binomial distribution, p = .5, 1000 trials, 100,000



2.18 Preview of the Central Limit Theorem

- For sufficiently large sample sizes, the distribution *of sample means* approximates a normal distribution
- This means with a large enough number of trials, we can apply the normal distribution to know things about measures of central tendency, measures of dispersion, and probabilities
- Sample sizes above 30
- This is just a preview

2.19 68-95-99.7 Rule

- One of the rules for normal distributions is:

The 68-95-99.7 rule

- 68% of the data is within 1 standard deviation of the mean
- 95% of the data is within 2 standard deviations of the mean
- 99.7% of the data is within 3 standard deviations of the mean

2.20 Preview of the Law of Large Numbers

- The law of large numbers tells us that if we repeat an experiment a large number of time, the *average* of the results will be close to the expected value

- This allows us to apply the **actual mean of the sample** to the **expected mean of the population**
- We can treat the *statistics* of the sample as estimates of the *parameters* of the population

2.21 Statistics and Parameters

- A *statistic* is a measure calculated from a sample of data

  - e.g., sample mean, sample variance, sample standard deviation

- A *parameter* is a measure calculated from the entire population

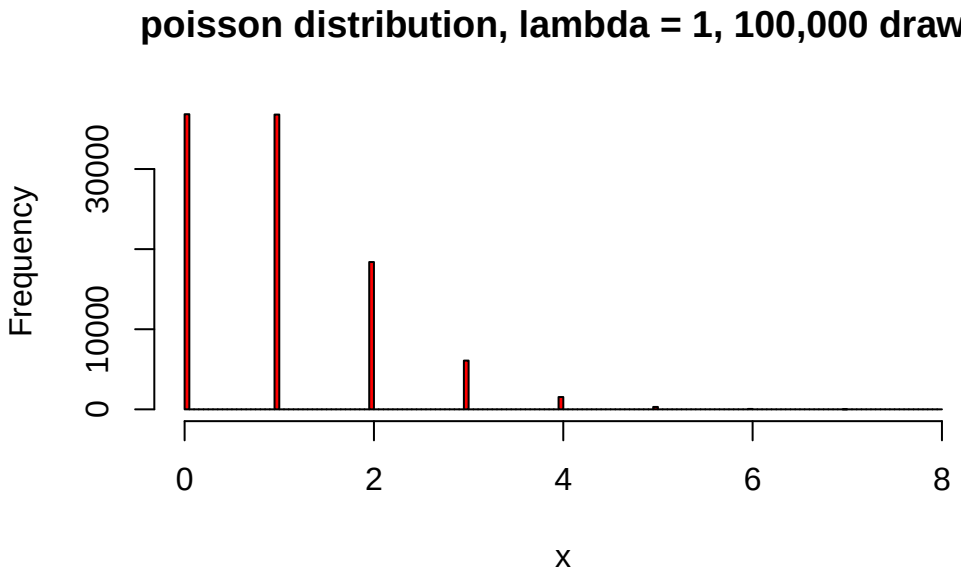
  - e.g., population mean, population variance, population standard deviation

2.22 Poisson distribution

- *Count* of number of events in a fixed time/space
- *Known* constant mean rate of occurrence
- *Independent* of time since last event

2.23 Poisson distribution

```
set.seed(123)

rand.poiss<- rpois(100000,1)
hist(rand.poiss, breaks = 200, freq = TRUE, main = "poisson distribution, lambda = 1, 100,000
```



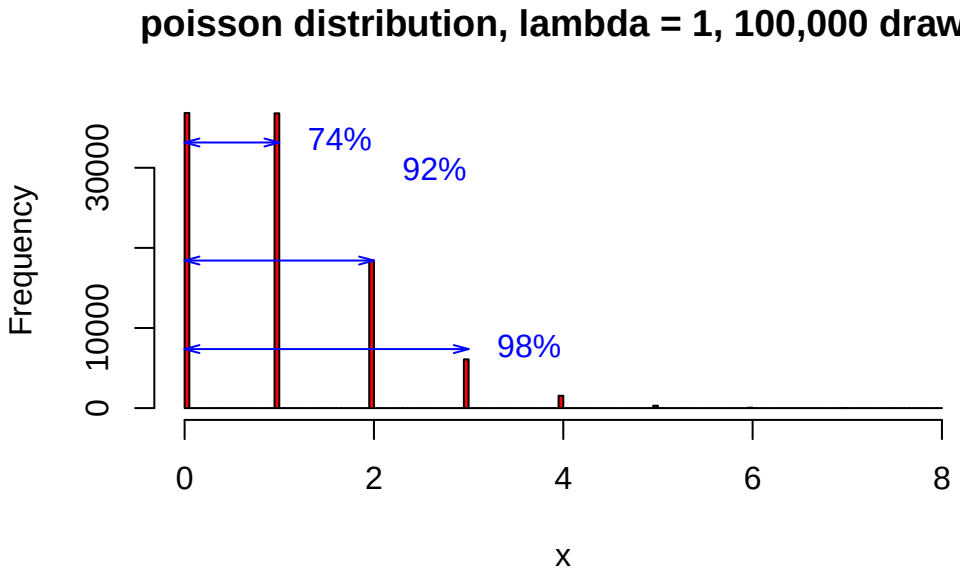
2.24 Poisson distribution: Assumptions

- The probability is different than the normal distribution
- lamda is both the mean and the variance of the distribution
- We don't generally use standar deviation with Poisson
- The probabilities in the following example are specific to this setup with lambda = 1
- This is why count variables should not strictly be handled with *Ordinary Least Squares (OLS)* regression, which is the *linear regression* technique we will be using

2.25 Poisson distribution: Probability

```
set.seed(123)

rand.poiss<- rpois(100000,1)
hp <- hist(rand.poiss, breaks = 200, freq = TRUE, main = "poisson distribution, lambda = 1, 1

# add lines to illustrate the probabilities of 0, 1, 2, and 3 events occurring

ypmax <- max(hp$counts)

# labels on the right
text(1.1, ypmax*0.9, "74%",   pos = 4, col = "blue")
text(2.1, ypmax*0.8, "92%",   pos = 4, col = "blue")
```

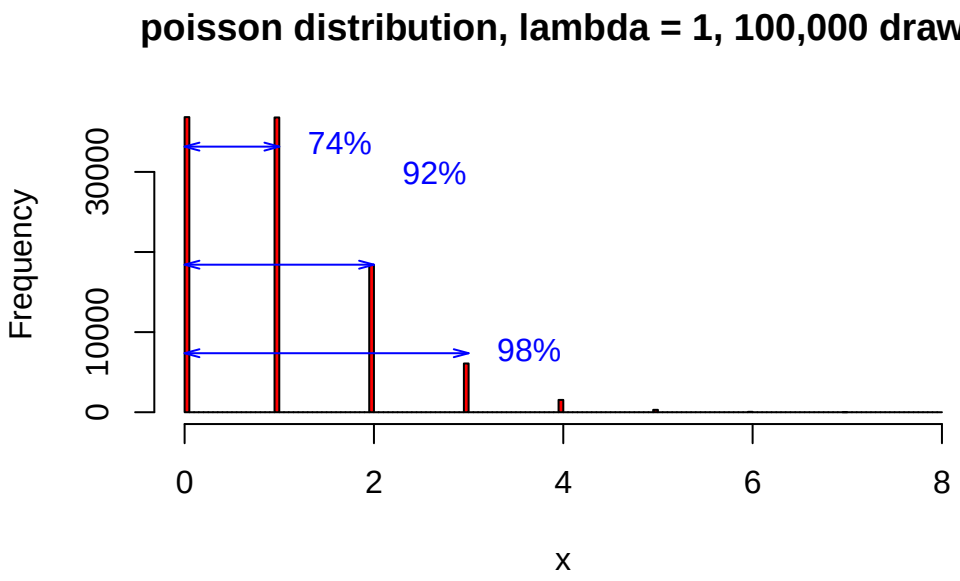
```
text(3.1, ypmax*0.2, "98%",   pos = 4, col = "blue")

# arrows spanning from 0 to each quantile line
arrows(x0 = 0.0, y0 = ypmax*0.9, x1 = 1, y1 = ypmax*0.9,
       code = 3, angle = 15, length = 0.08, col = "blue")
arrows(x0 = 0.0, y0 = ypmax*0.5, x1 = 2, y1 = ypmax*0.5,
       code = 3, angle = 15, length = 0.08, col = "blue")
arrows(x0 = 0.0, y0 = ypmax*0.2, x1 = 3, y1 = ypmax*0.2,
       code = 3, angle = 15, length = 0.08, col = "blue")
```



2.26 Poisson distribution: with code

```
set.seed(123)

rand.poiss<- rpois(100000,1)
hp <- hist(rand.poiss, breaks = 200, freq = TRUE, main = "poisson distribution, lambda = 1, 1

# add lines to illustrate the probabilities of 0, 1, 2, and 3 events occurring

ypmax <- max(hp$counts)

# labels on the right
text(1.1, ypmax*0.9, "74%",   pos = 4, col = "blue")
```

```
text(2.1, ypmax*0.8, "92%",   pos = 4, col = "blue")
text(3.1, ypmax*0.2, "98%",   pos = 4, col = "blue")

# arrows spanning from 0 to each quantile line
arrows(x0 = 0.0, y0 = ypmax*0.9, x1 = 1, y1 = ypmax*0.9,
       code = 3, angle = 15, length = 0.08, col = "blue")
arrows(x0 = 0.0, y0 = ypmax*0.5, x1 = 2, y1 = ypmax*0.5,
       code = 3, angle = 15, length = 0.08, col = "blue")
arrows(x0 = 0.0, y0 = ypmax*0.2, x1 = 3, y1 = ypmax*0.2,
       code = 3, angle = 15, length = 0.08, col = "blue")
```

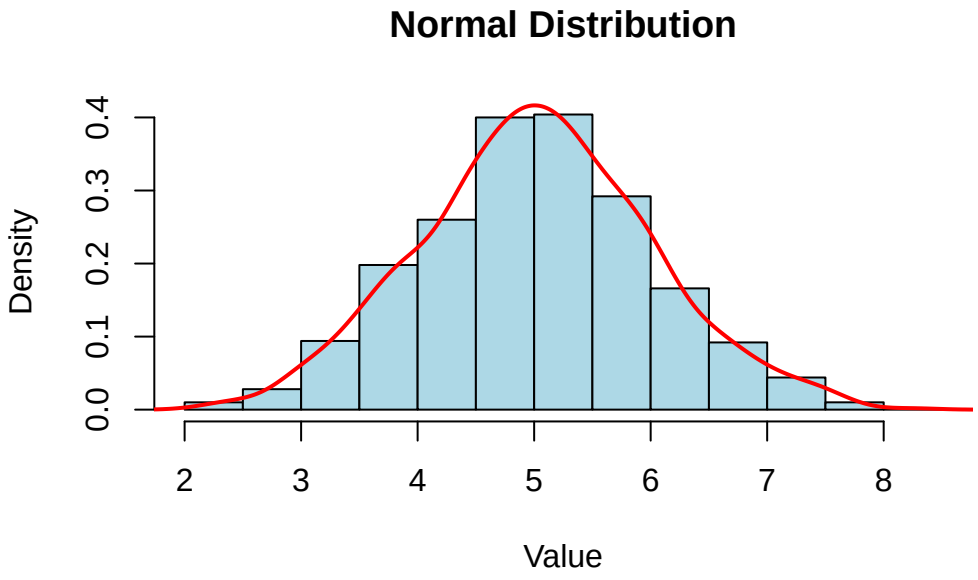


2.27 Why we can't use standard OLS regression for other DGP directly

- We base the likelihood of something being significant on the proximity to the mean
- As things get further from the mean in a normal distribution, they become less likely
- We can apply OLS to sample means of multiple trials because of the Central Limit Theorem
- For specific distributions, like Poisson, we have often have better techniques

2.28 Why we don't use standard OLS regression for other DGP: Example

```r
# Set a seed for reproducibility
set.seed(123)

# Generate data for a normal distribution
mean_normal <- 5  # Mean of the normal distribution
sd_normal <- 1    # Standard deviation of the normal distribution
normal_data <- rnorm(1000, mean = mean_normal, sd = sd_normal)

# Create a histogram for the normal distribution
hist(normal_data, probability = TRUE, main = "Normal Distribution",
     xlab = "Value", ylab = "Density", col = "lightblue")

# Overlay a density plot for the normal distribution
lines(density(normal_data), col = "red", lwd = 2)
```



2.29 Poisson vs. Normal Distribution

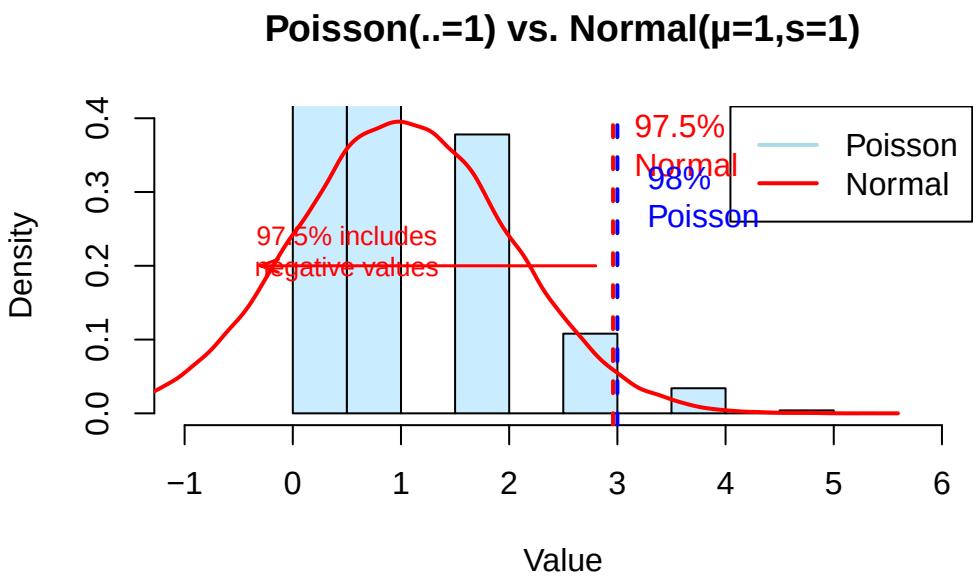
```r
# Set a seed for reproducibility
set.seed(123)

# Generate data
poisson_data <- rpois(1000, lambda = 1)
normal_data <- rnorm(100000, mean = 1, sd = 1)
```

```r
# Create histogram for Poisson (density)
h <- hist(poisson_data, probability = TRUE,
          main = "Poisson( =1) vs. Normal( =1, =1)",
          xlab = "Value", ylab = "Density", ylim = c(0, 0.4),
          col = rgb(0.7, 0.9, 1, 0.7),
          xlim = c(-1, 6))  # extend x to show negatives
```

```
Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
conversion failure on 'Poisson( =1) vs. Normal( =1, =1)' in 'mbcsToSbcs': dot
substituted for <ce>
```

```
Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
conversion failure on 'Poisson( =1) vs. Normal( =1, =1)' in 'mbcsToSbcs': dot
substituted for <bb>
```

```r
# Overlay normal density
lines(density(normal_data), col = "red", lwd = 2)

# Legend
legend("topright", legend = c("Poisson", "Normal"),
       col = c("lightblue", "red"), lty = 1, lwd = 2)

# Lines
abline(v = qnorm(0.975, 1, 1), col = "red", lwd = 2, lty = 2)  # ~2.96
abline(v = qpois(0.98, 1), col = "blue", lwd = 2, lty = 2)     # 3

# Labels
text(2.98, 0.35, "97.5%\nNormal", col = "red", pos = 4)
text(3.1,  0.28, "98%\nPoisson", col = "blue", pos = 4)

# Arrow pointing left from 97.5% line to negatives, with label
arrows(x0 = 2.8, y0 = 0.20, x1 = -0.3, y1 = 0.20,
       code = 2, angle = 20, length = 0.1, col = "red", lwd = 1.5)
text(0.5, 0.22, "97.5% includes\nnnegative values", col = "red", cex = 0.8)
```



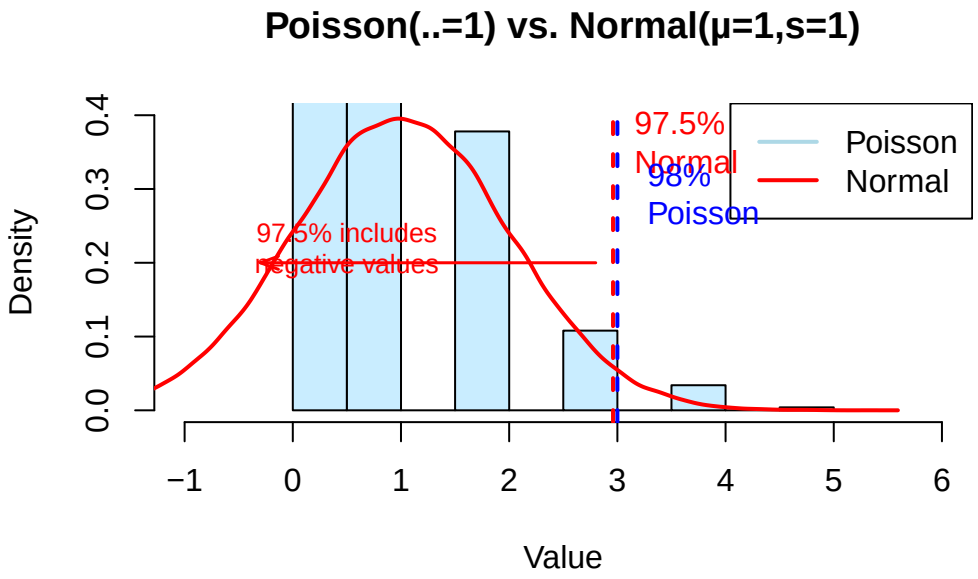
2.30 Poisson v. Normal: Code

```
# Set a seed for reproducibility
set.seed(123)

# Generate data
poisson_data <- rpois(1000, lambda = 1)
normal_data <- rnorm(100000, mean = 1, sd = 1)

# Create histogram for Poisson (density)
h <- hist(poisson_data, probability = TRUE,
          main = "Poisson( =1) vs. Normal( =1, =1)",
          xlab = "Value", ylab = "Density", ylim = c(0, 0.4),
          col = rgb(0.7, 0.9, 1, 0.7),
          xlim = c(-1, 6))  # extend x to show negatives
```

```
Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
conversion failure on 'Poisson( =1) vs. Normal( =1, =1)' in 'mbcsToSbcs': dot
substituted for <ce>
```

```
Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...):
conversion failure on 'Poisson( =1) vs. Normal( =1, =1)' in 'mbcsToSbcs': dot
substituted for <bb>
```

```r
# Overlay normal density
lines(density(normal_data), col = "red", lwd = 2)

# Legend
legend("topright", legend = c("Poisson", "Normal"),
       col = c("lightblue", "red"), lty = 1, lwd = 2)

# Lines
abline(v = qnorm(0.975, 1, 1), col = "red", lwd = 2, lty = 2)  # ~2.96
abline(v = qpois(0.98, 1), col = "blue", lwd = 2, lty = 2)     # 3

# Labels
text(2.98, 0.35, "97.5%\nNormal", col = "red", pos = 4)
text(3.1,  0.28, "98%\nPoisson", col = "blue", pos = 4)

# Arrow pointing left from 97.5% line to negatives, with label
arrows(x0 = 2.8, y0 = 0.20, x1 = -0.3, y1 = 0.20,
       code = 2, angle = 20, length = 0.1, col = "red", lwd = 1.5)
text(0.5, 0.22, "97.5% includes\nnegative values", col = "red", cex = 0.8)
```



2.31 Authorship and License

- Author: Tom Hanna
- Website: tomhanna.me

- License: This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.