**#WHAT IS STATISTIC**

-Measure of Center : to see the best option between mean and median, look at the skew of the graph. If left or right skew then choose median. If normal distribution used mean.

-Mean and Median :

# Filter for Belgium

belgium\_consumption <- food\_consumption %>%

  filter(country == "Belgium")

# Filter for USA

usa\_consumption <- food\_consumption %>%

  filter(country == "USA")

# Calculate mean and median consumption in Belgium

mean(belgium\_consumption$consumption)

median(belgium\_consumption$consumption)

# Calculate mean and median consumption in USA

mean(usa\_consumption$consumption)

median(usa\_consumption$consumption)

food\_consumption %>%

  # Filter for Belgium and USA

  filter(country %in% c("Belgium", "USA")) %>%

  # Group by country

  group\_by(country) %>%

  # Get mean\_consumption and median\_consumption

  summarise(mean\_consumption = mean(consumption),

      median\_consumption = median(consumption))

-mean vs median

food\_consumption %>%

  # Filter for rice food category

  filter(food\_category == "rice") %>%

  # Create histogram of co2\_emission

  ggplot(aes(co2\_emission)) +geom\_histogram()

food\_consumption %>%

  # Filter for rice food category

  filter(food\_category == "rice") %>%

  # Create histogram of co2\_emission

  ggplot(aes(co2\_emission)) +

    geom\_histogram()

food\_consumption %>%

  # Filter for rice food category

  filter(food\_category == "rice") %>%

  # Get mean\_co2 and median\_co2

  summarize(mean\_co2 = mean(co2\_emission),

            median\_co2 = median(co2\_emission))

-Measure of Spread

-Variances : Average distance from each data point to the data’s mean

**-how to calculate variance :**

**\*dists <- msleep$sleep\_total – mean(msleep$sleep\_total) [deviate each data point with mean of total]**

**\*square\_dists <- (dists)^2**

**\*sum\_sq\_dists <- sum(square\_dists)**

**\*variance = sum\_sq\_dists/(number of sample-1)**

-The higher variance, the more spread out the data

-Calculate variance using one step :

**\*var(msleep$sleep\_total)**

-Standard Deviation

**\*sqrt(var(msleep$sleep\_total)**

**\*sd(msleep$sleep\_total)**

-Mean Absolute Deviation

**\*dists <- msleep$sleep\_total – mean(msleep$sleep\_total)**

**\*mean(abs(dists))**

-Quartiles :

**\*quantile(msleep$sleep\_total)** -> showing quartile of each data

**\*quantile(msleep$sleep\_total, probs = seq(0,1,0.2))** ->showing quartile form 0%-100% with jump every 20%

-IQR (Interquartile range) : Height of the box in boxplot

**\*quantile(msleep$sleep\_total, 0.75) –quantile(msleep$sleep\_total, 0.25)**

-Outliers : data point that is substantially different from the others

**If data <Q1-1.5 x IQR or if data > Q3 + 1.5 x IQR**

-Finding Outlier :

**\*iqr <- quantile(msleep$bodywt, 0.75) – quantile(msleep$bodywt, 0.25)**

**\*lower\_threshold <- quantile(msleep$bodywt, 0.25) – 1.5\*iqr**

**\*upper\_threshold <- quantile(msleep$bodywt, 0.75) + 1.5\*iqr**

**\*msleep %>% filter(bodywt < lower\_threshold | bodywt > upper\_threshold) %>%**

**select(name, vore, sleep\_total, bodywt)**

-Quartiles, Quantile, Quintiles :

# Calculate the quartiles of co2\_emission

quantile(food\_consumption$co2\_emission)

# Calculate the quintiles of co2\_emission

quantile(food\_consumption$co2\_emission, probs = seq(0,1,0.2))

# Calculate the deciles of co2\_emission

quantile(food\_consumption$co2\_emission, probs = seq(0,1,0.1))

-Varaince and Standard Deviation :

# Calculate variance and sd of co2\_emission for each food\_category

food\_consumption %>%

  group\_by(food\_category) %>%

  summarize(var\_co2 = var(co2\_emission),

     sd\_co2 = sd(co2\_emission))

# Plot food\_consumption with co2\_emission on x-axis

ggplot(food\_consumption, aes(co2\_emission)) +

  # Create a histogram

  geom\_histogram() +

  # Create a separate sub-graph for each food\_category

  facet\_wrap(~ food\_category)

-Finding Outlier using IQR :

# Calculate total co2\_emission per country: emissions\_by\_country

emissions\_by\_country <- food\_consumption %>%

  group\_by(country) %>%

  summarize(total\_emission = sum(co2\_emission))

# Compute the first and third quantiles and IQR of total\_emission

q1 <- quantile(emissions\_by\_country$total\_emission, 0.25)

q3 <- quantile(emissions\_by\_country$total\_emission, 0.75)

iqr <- q3 - q1

# Calculate the lower and upper cutoffs for outliers

lower <- q1 - 1.5 \* iqr

upper <- q3 + 1.5 \* iqr

# Filter emissions\_by\_country to find outliers

emissions\_by\_country %>%

  filter(total\_emission<lower | total\_emission>upper)

**#RANDOM NUMBERS AND PROBABILITY**

-What are the chances

-Sampling from a data frame

**\*sales\_count %>%  
\*sample\_n (1) -> Get the random result**

-Setting the random seed : to get the same result when we take the sample

**\*set.seed(100) -> random urutan ke 100**

**\*sales\_count %>%**

**\*sample\_n(1)** -> the result will always be same due to of set.seed

-Sampling with replacement : Sample will be back to drawing after the sample choose

**\*sales\_count %>%**

**\*sample\_n(2, replace = TRUE)** - > If there are 2 lottery

If there are 5 lottery

**\*sample(sales\_team, 5, replace = TRUE)**

-Sampling without replacement : Sample won’t be back to drawing after the sample choose

-Independent Events : two events are independent if the probability of the second event isn’t affected by the outcome of the first event.

-Dependent Events : two events are dependent if the probability of the second event is affected by the outcome of the first event

-sampling without replacement = each pick is dependent

-sampling with replacement = each pick is independent

-Calculating Probabilities

# Calculate probability of picking a deal with each product

amir\_deals %>%

  count(product) %>%

  mutate(prob = n/sum(n))

-Discrete Distribution : Uniform discrete distribution

-Visualizing the sample of dice

**\*ggplot(rolls\_10, aes(n)) + geom\_histogram(bins = 6)**

-Law of large number : as the size of your sample increases, the sample mean will approach the expected value.

-Creating Probability distribution :

# Create a histogram of group\_size

ggplot(restaurant\_groups, aes(group\_size)) +

  geom\_histogram(bins = 5)

# Create probability distribution

size\_distribution <- restaurant\_groups %>%

  count(group\_size) %>%

  mutate(probability = n / sum(n))

# Calculate expected group size

expected\_val <- sum(size\_distribution$group\_size\*          size\_distribution$probability)

expected\_val

# Create probability distribution

size\_distribution <- restaurant\_groups %>%

  count(group\_size) %>%

  mutate(probability = n / sum(n))

# Calculate probability of picking group of 4 or more

size\_distribution %>%

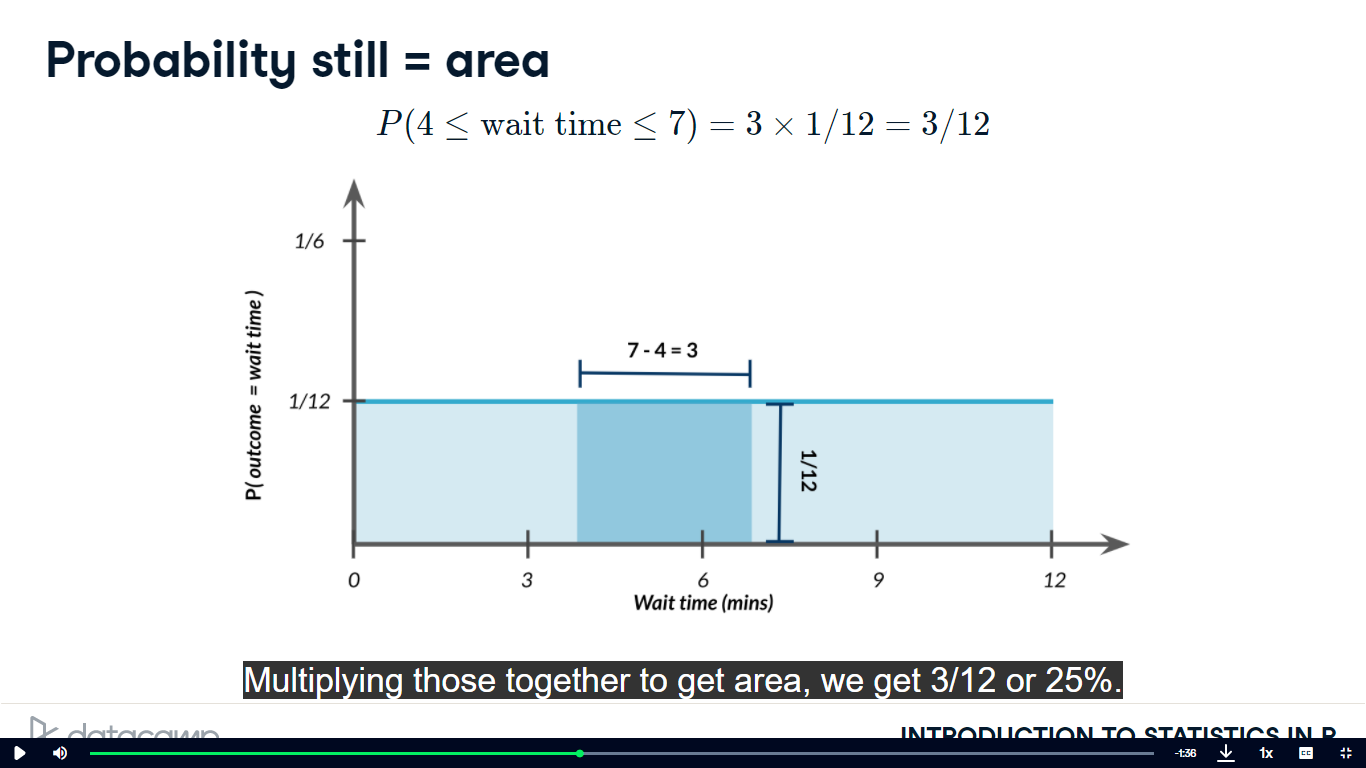
  # Filter for groups of 4 or larger

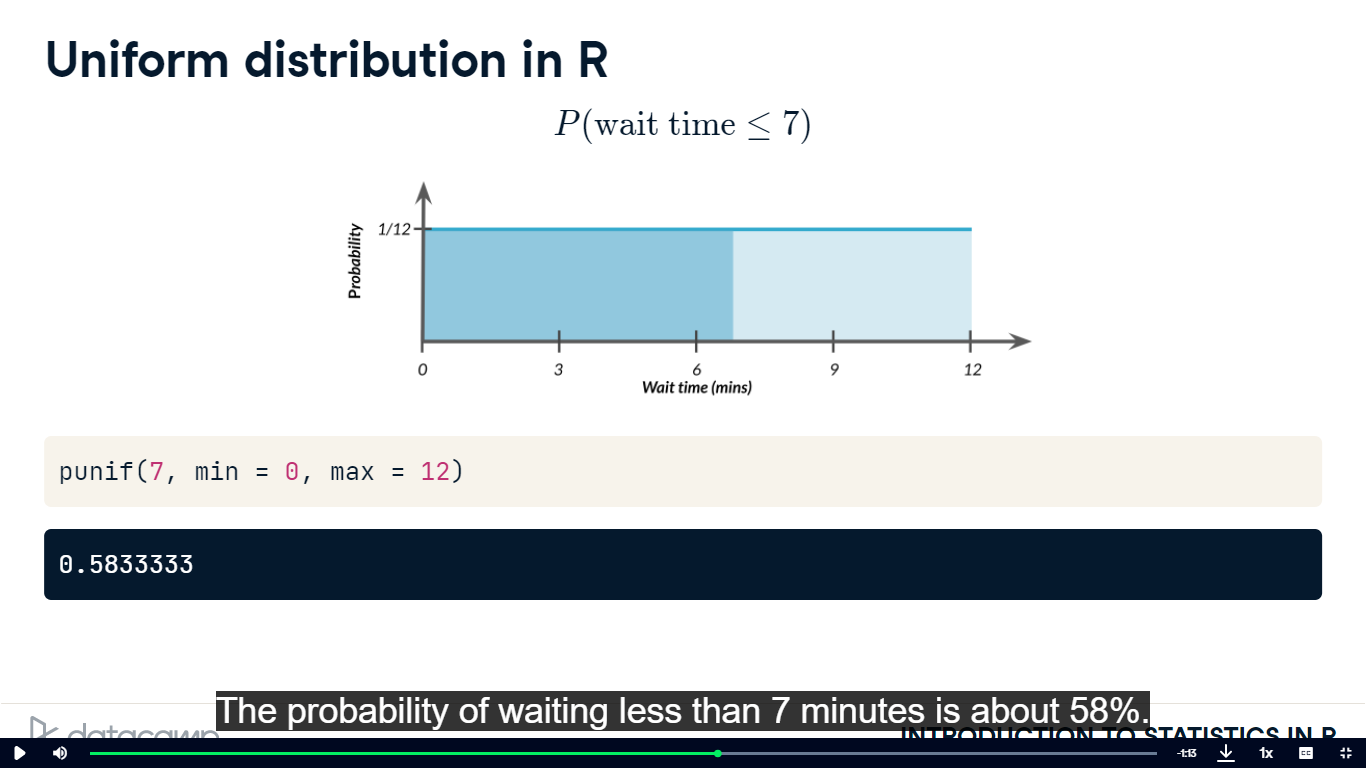
  filter(group\_size>=4) %>%

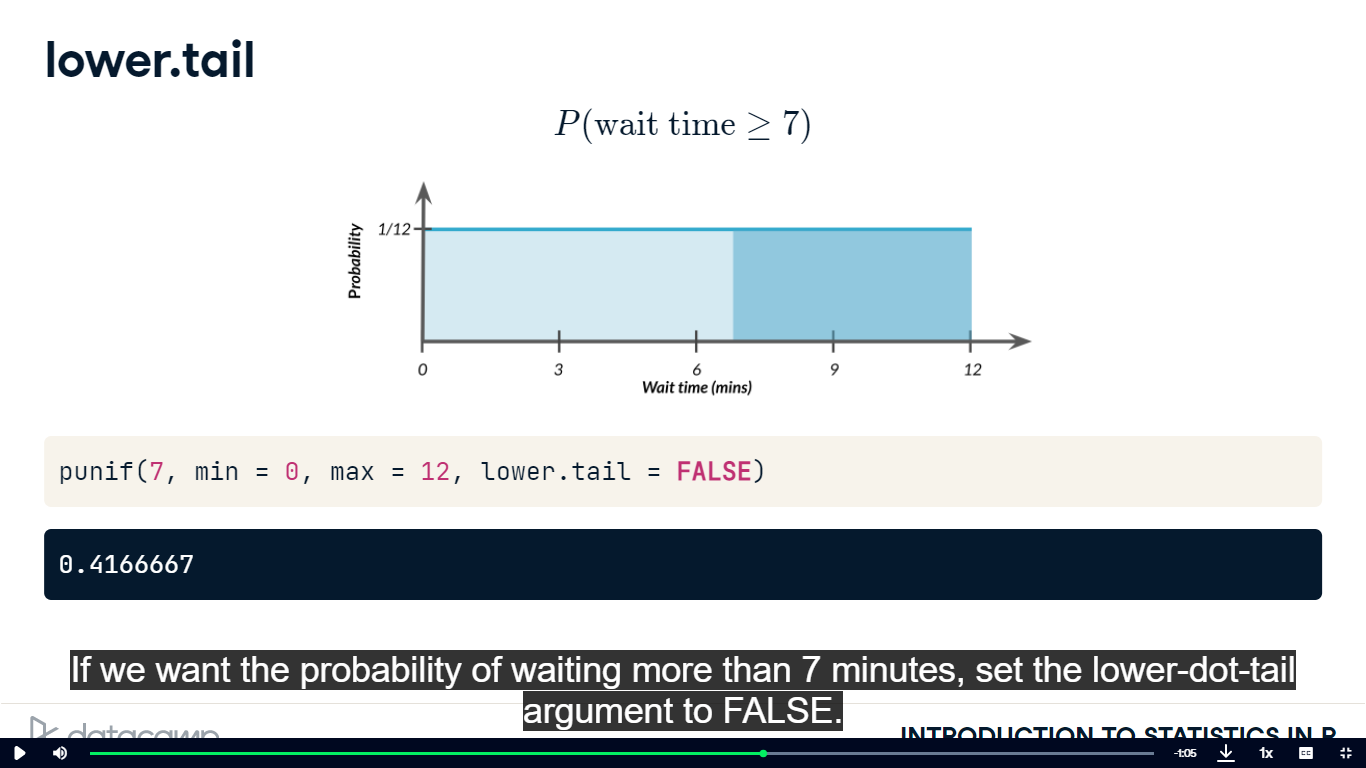
  # Calculate prob\_4\_or\_more by taking sum of probabilities

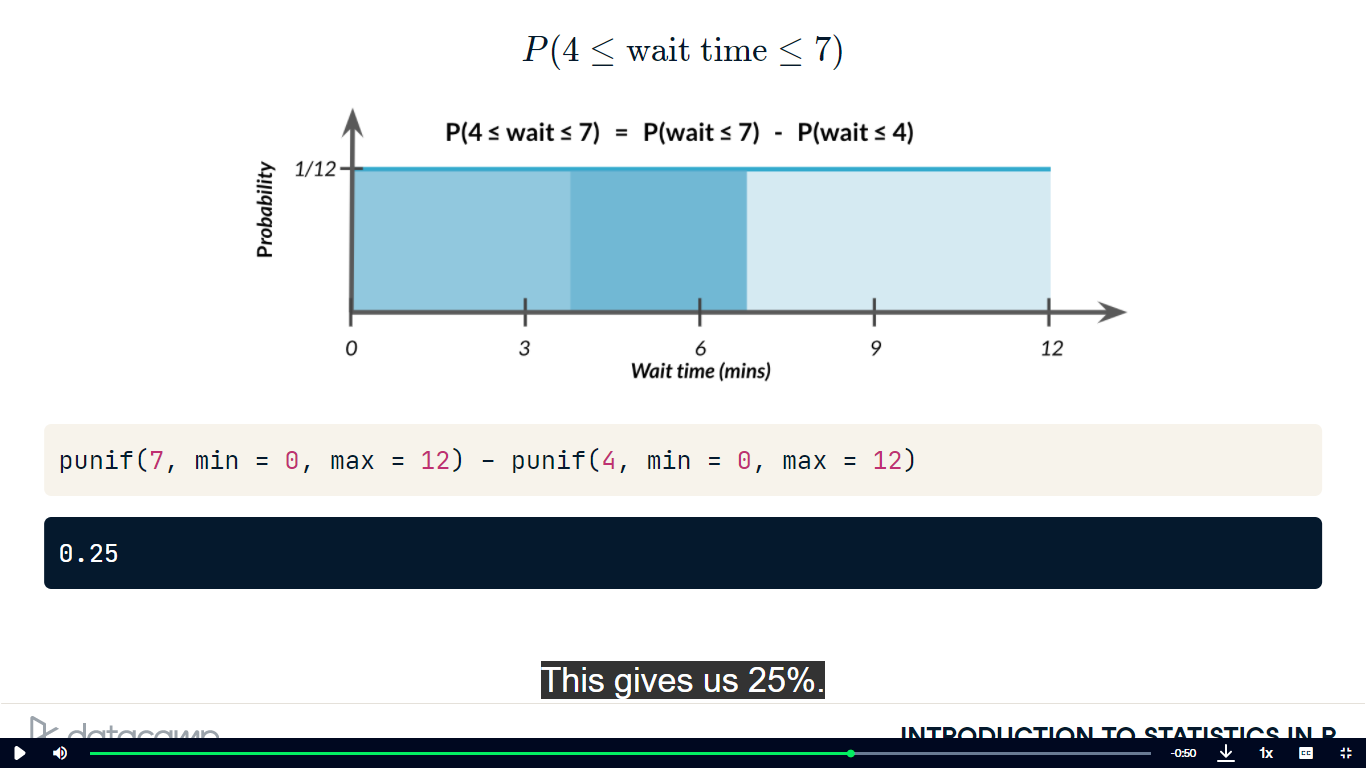
  summarise(prob\_4\_or\_more = sum(probability))

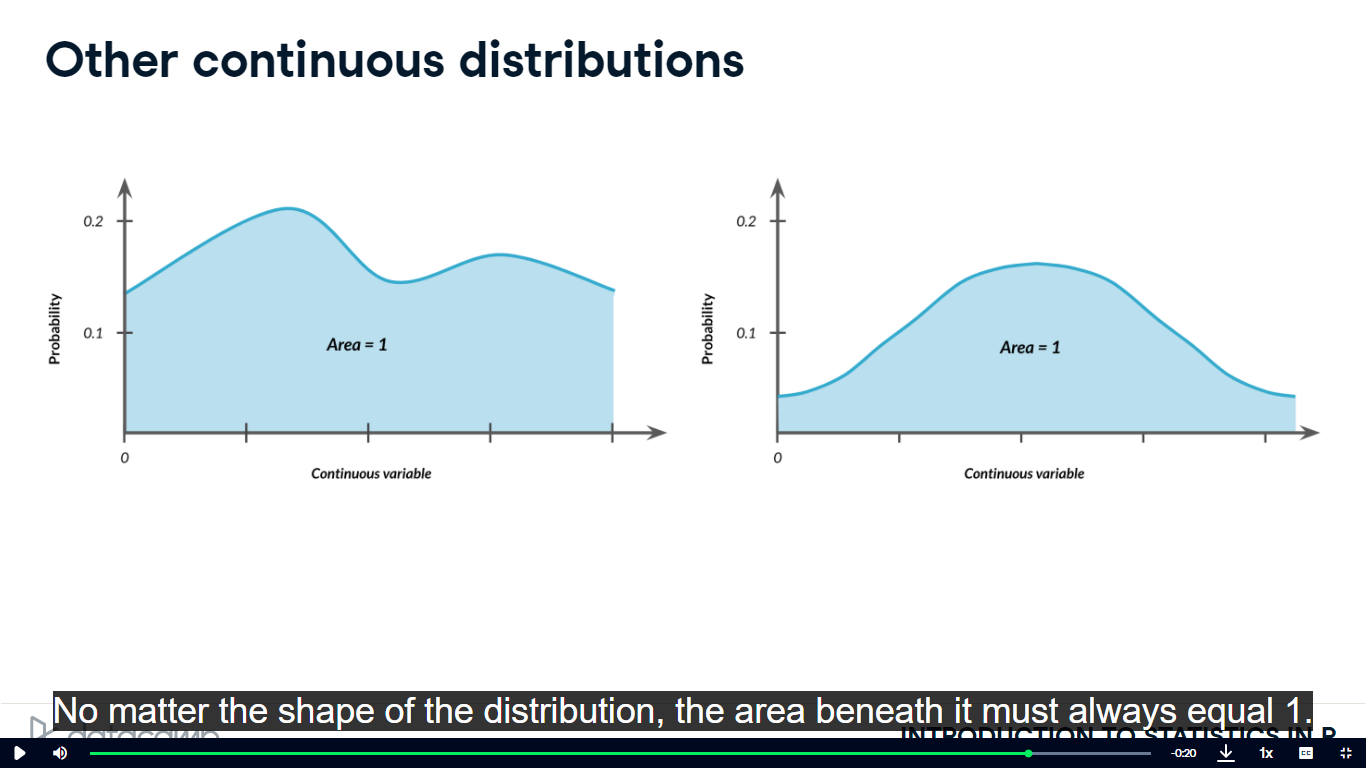
-Continous distribution

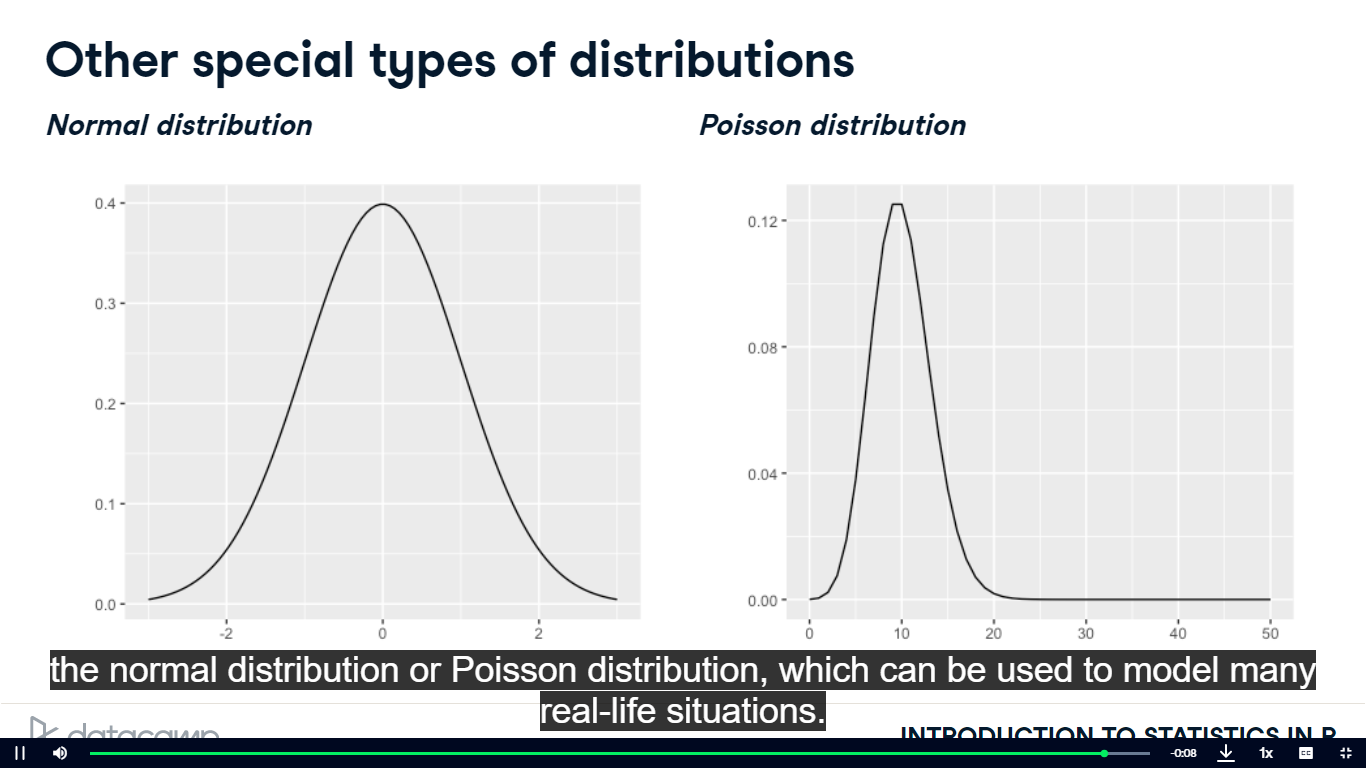












-Data Backups

# Min and max wait times for back-up that happens every 30 min

min <- 0

max <- 30

# Calculate probability of waiting less than 5 mins

prob\_less\_than\_5 <- punif(5, min = 0, max = 30)

prob\_less\_than\_5

# Min and max wait times for back-up that happens every 30 min

min <- 0

max <- 30

# Calculate probability of waiting more than 5 mins

prob\_greater\_than\_5 <- punif(5, min = 0, max = 30, lower.tail = FALSE)

prob\_greater\_than\_5

# Min and max wait times for back-up that happens every 30 min

min <- 0

max <- 30

# Calculate probability of waiting 10-20 mins

prob\_between\_10\_and\_20 <- punif(20, min = 0, max = 30) - punif(10, min = 0, max = 30)

prob\_between\_10\_and\_20

-Simulating wait times

# Set random seed to 334

set.seed(334)

# Generate 1000 wait times between 0 and 30 mins, save in time column

wait\_times %>%

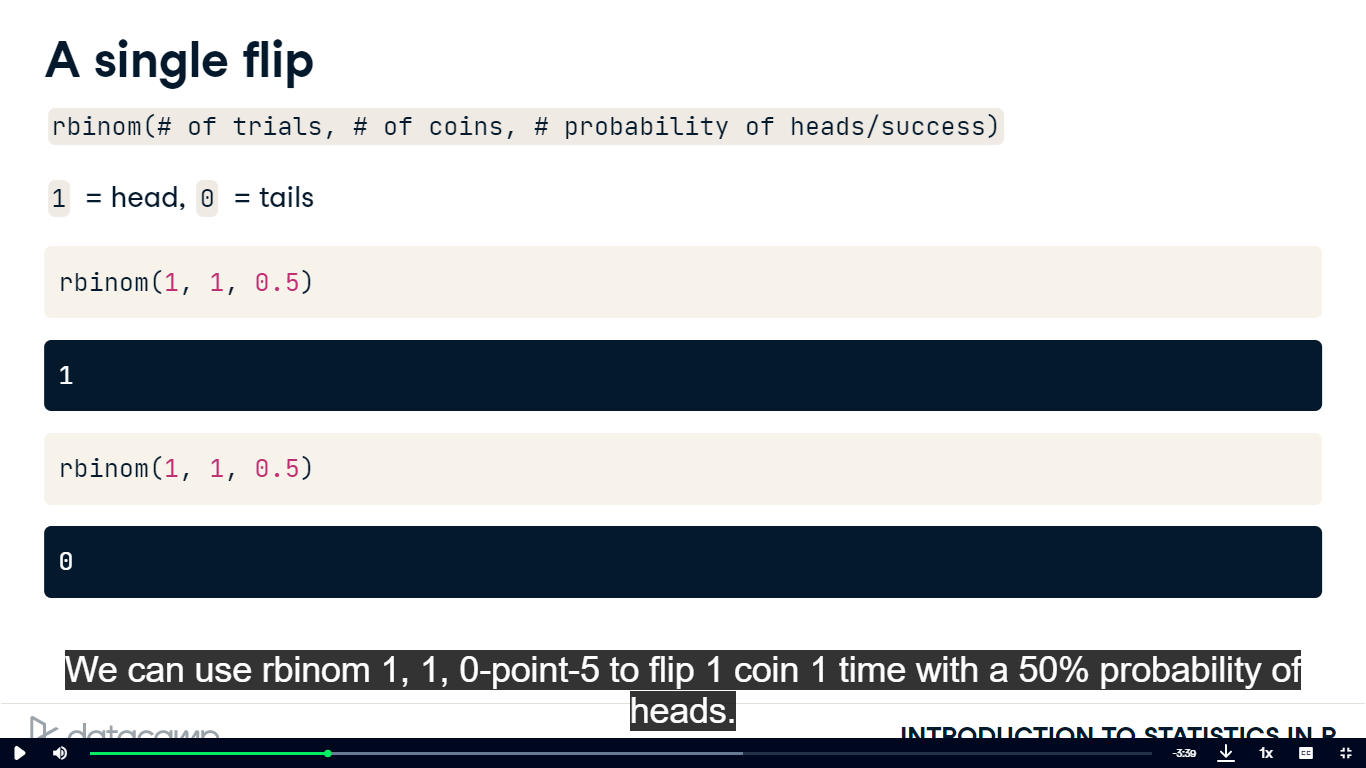
  mutate(time = runif(1000, min = 0, max = 30)) %>%

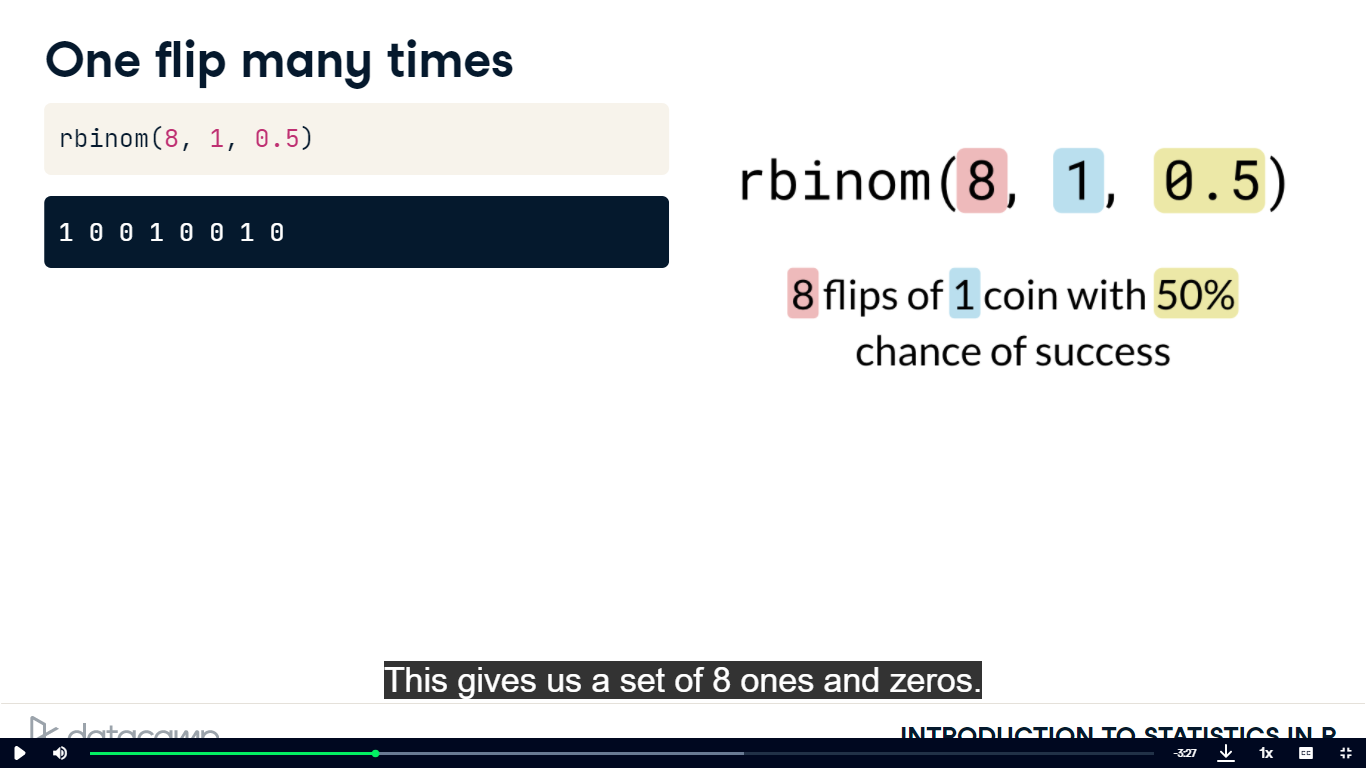
  # Create a histogram of simulated times

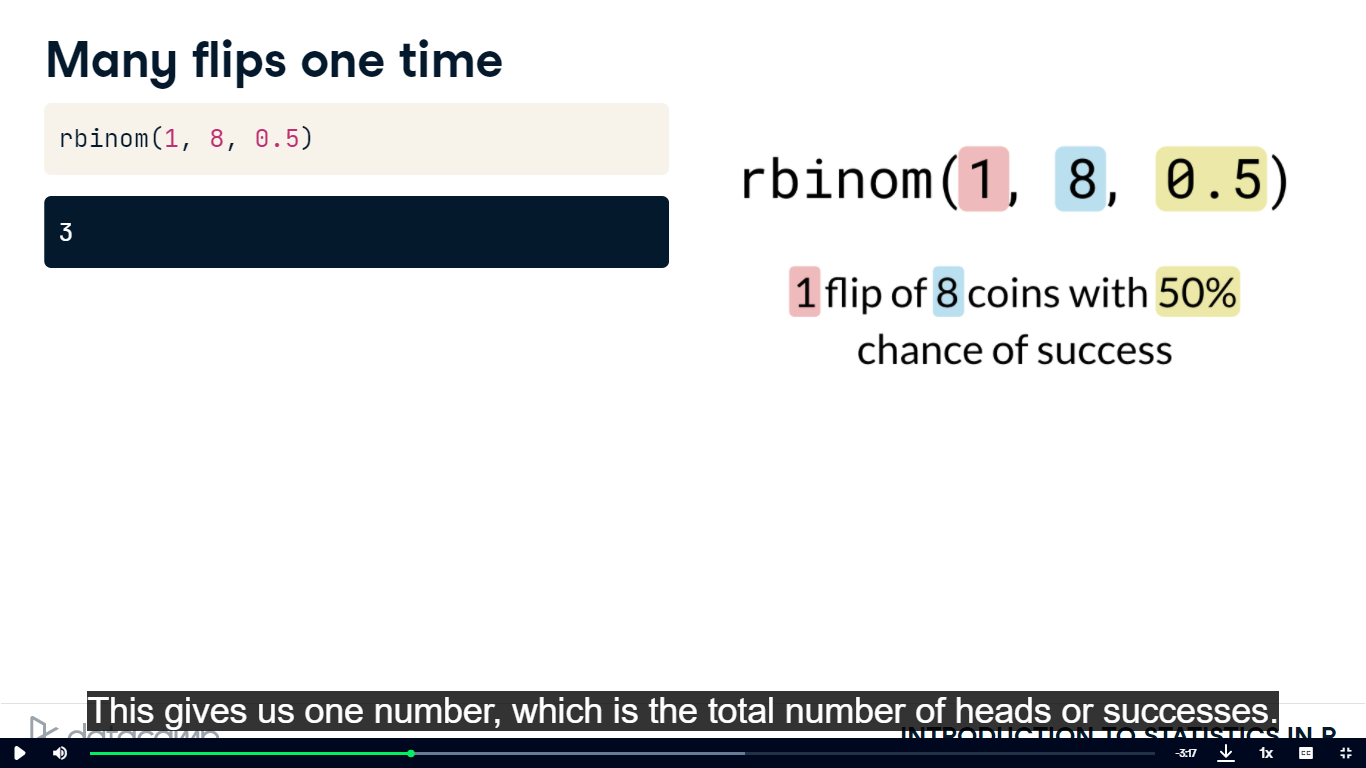
  ggplot(aes(time)) +

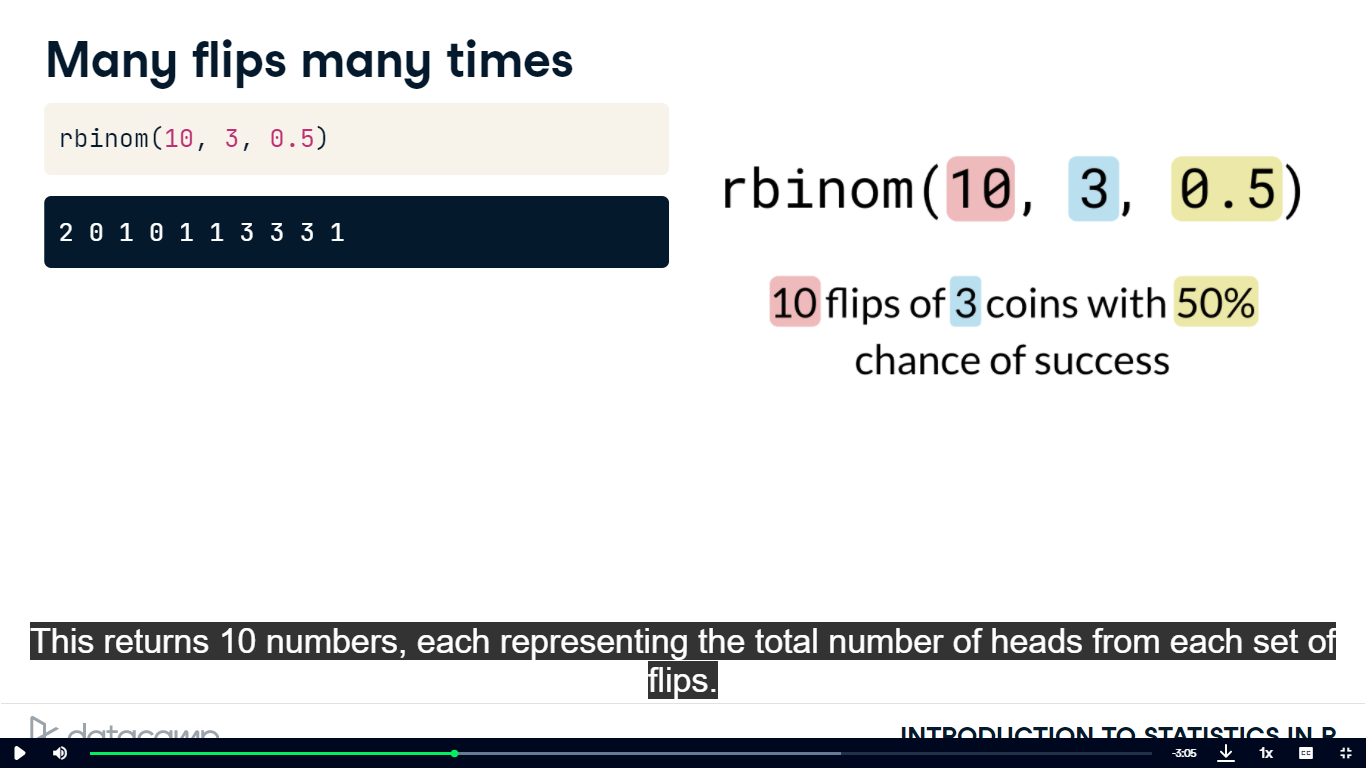
  geom\_histogram(bins=30)

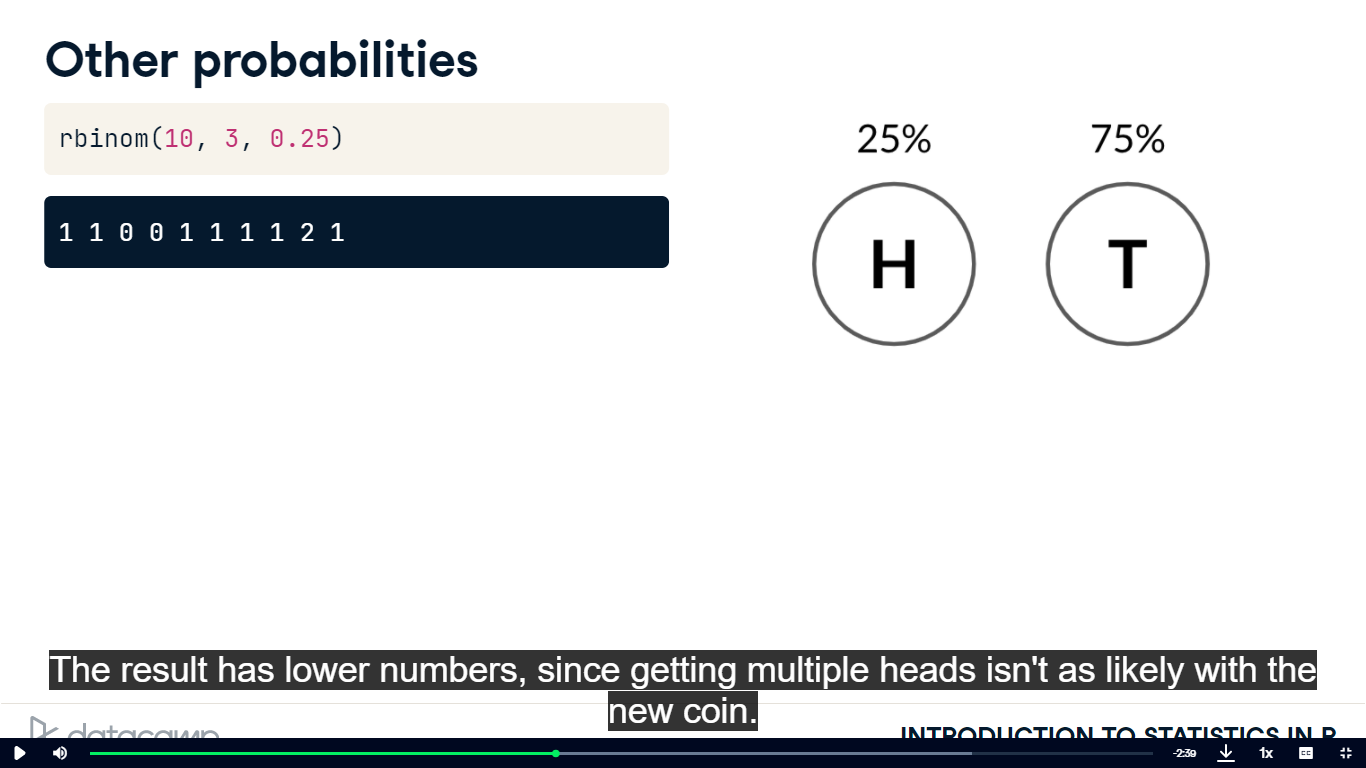
-The Binomial Distribution

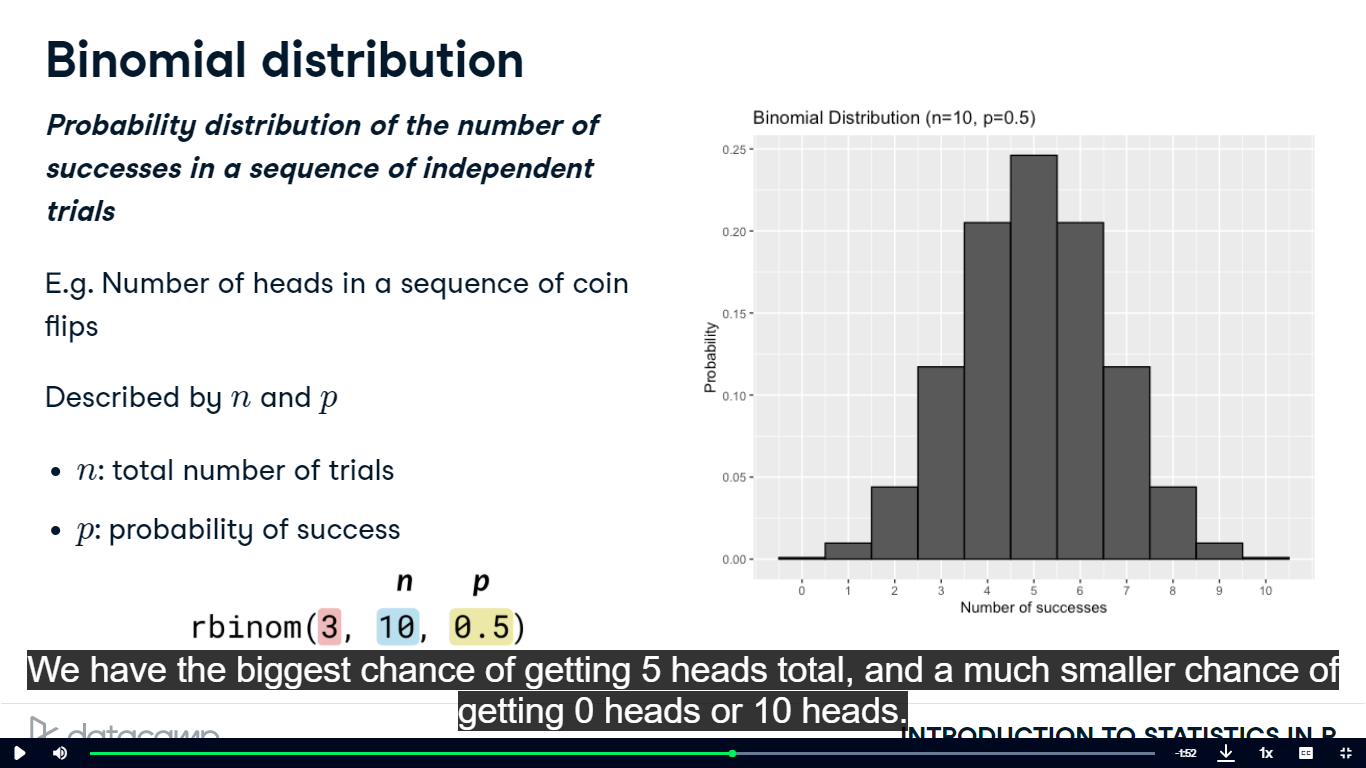


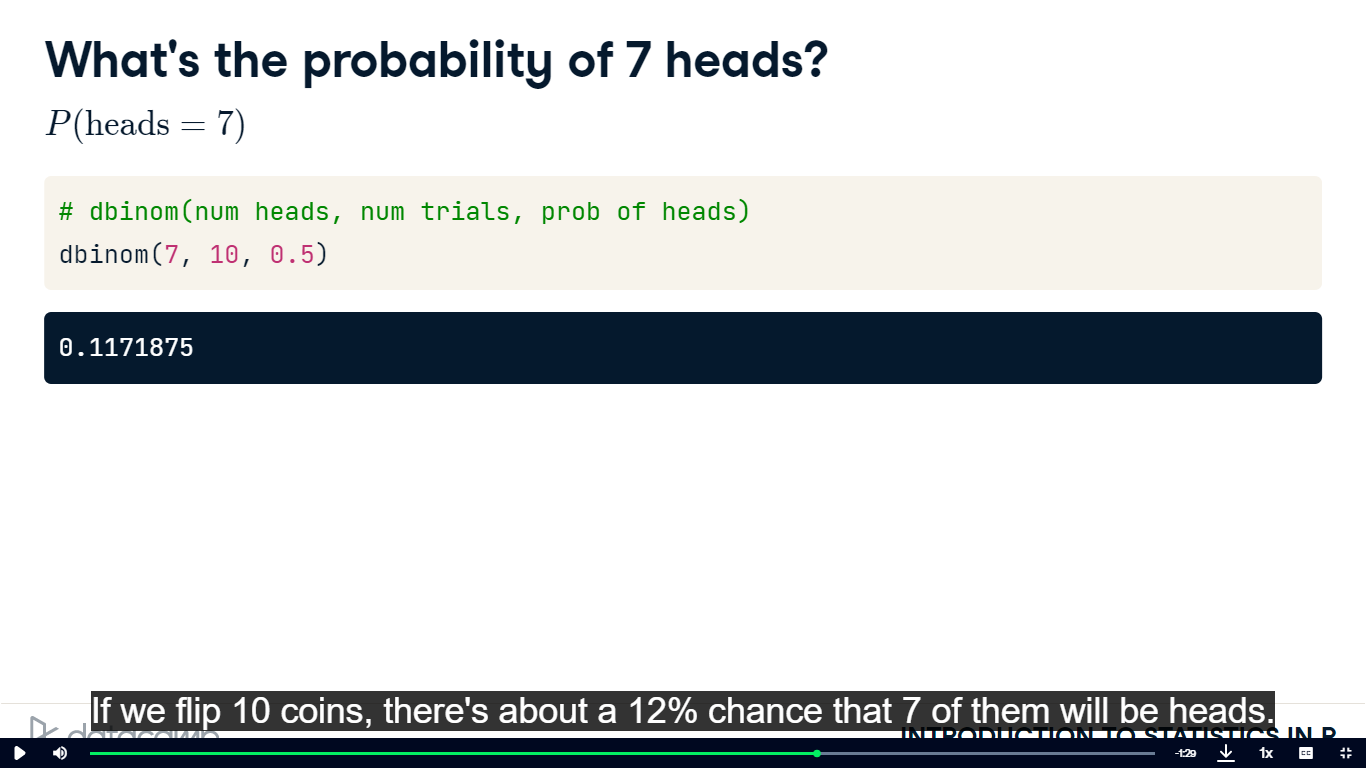


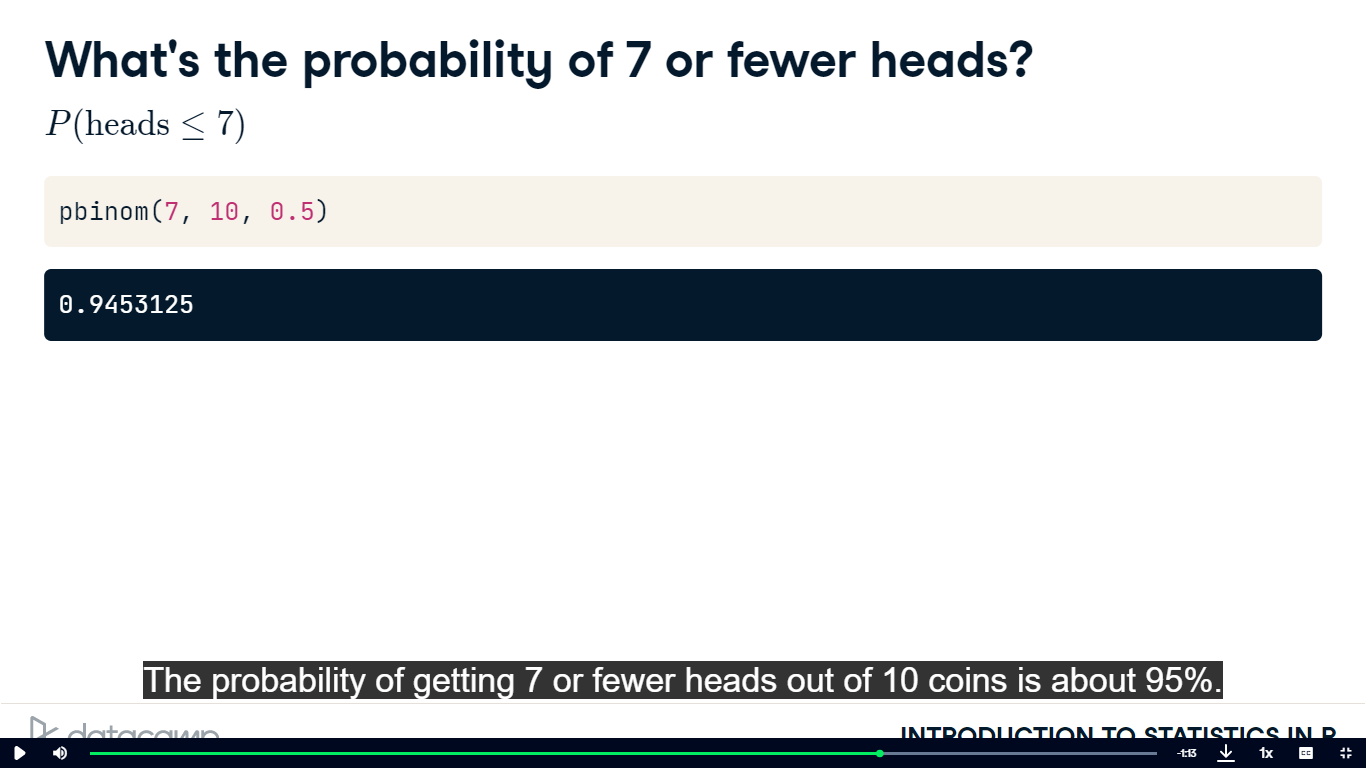


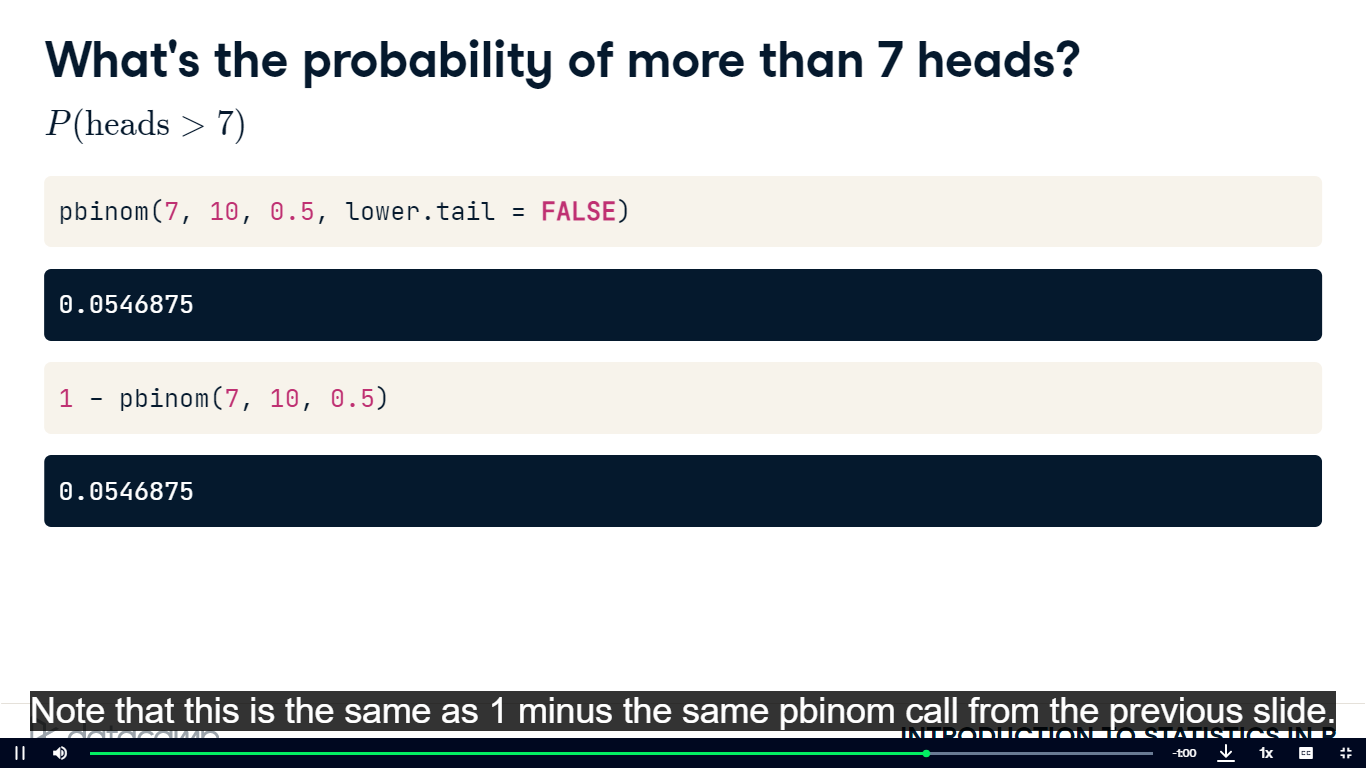


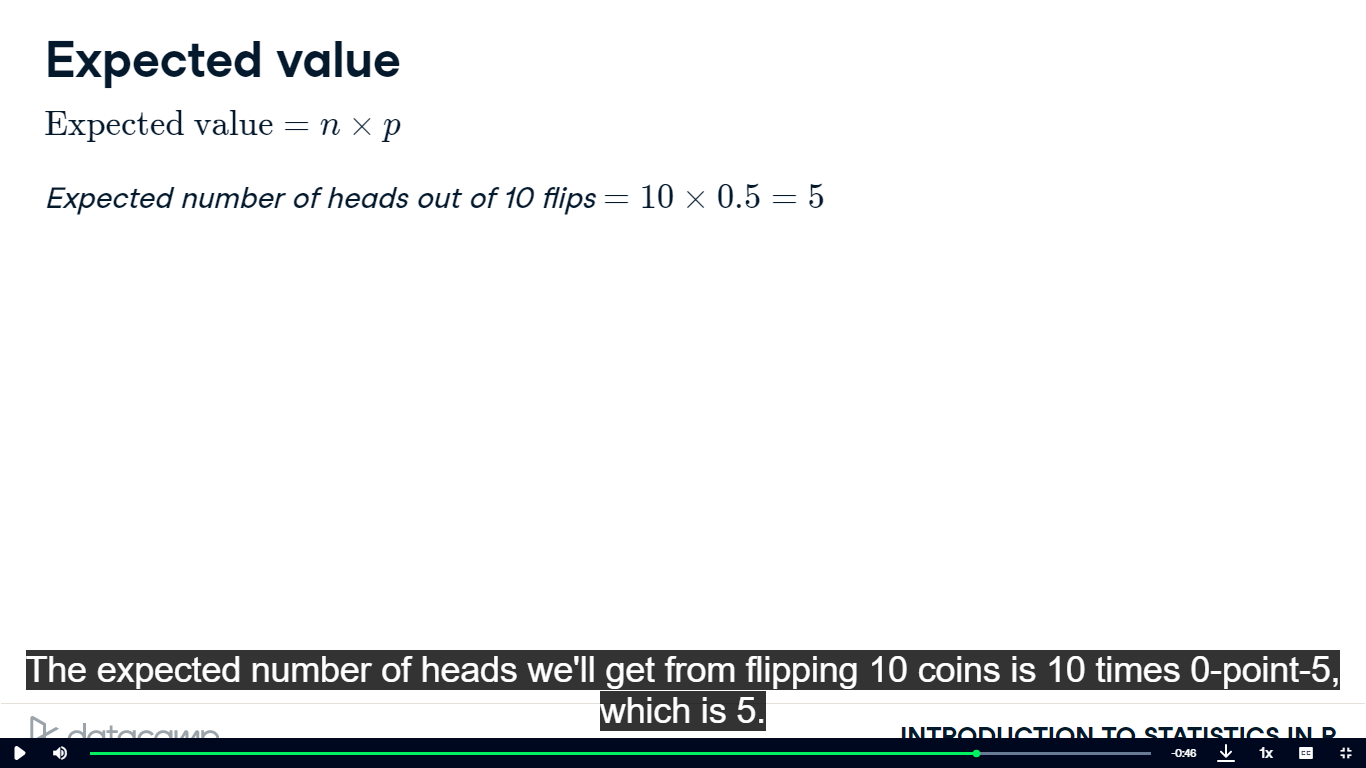


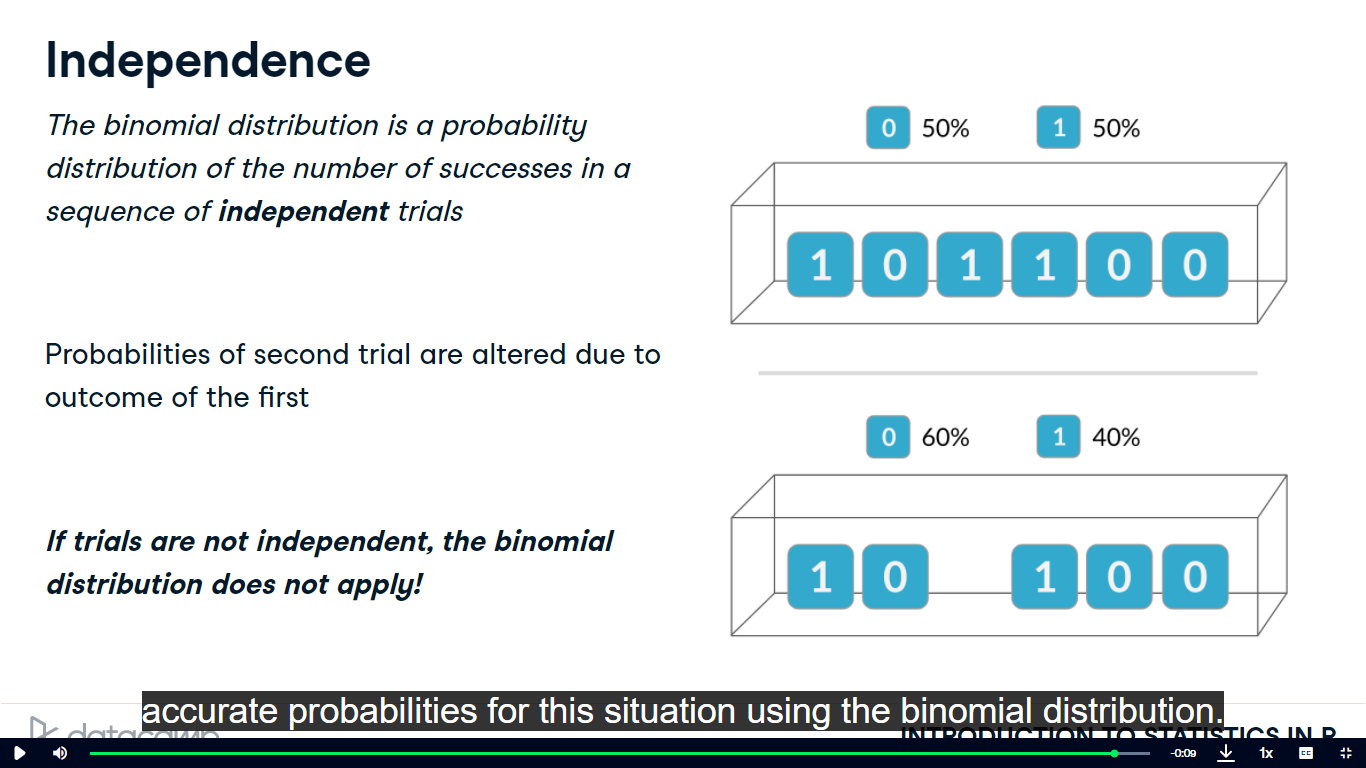












-Simulating Sales Deal

# Set random seed to 10

set.seed(10)

# Simulate a single deal

rbinom(1,1,0.3)

# Set random seed to 10

set.seed(10)

# Simulate 1 week of 3 deals

rbinom(1,3,0.3)

# Set random seed to 10

set.seed(10)

# Simulate 52 weeks of 3 deals

deals <- rbinom(52,3,0.3)

# Calculate mean deals won per week

mean(deals)

-Calculating Binomial Probabilities

# Probability of closing 3 out of 3 deals

dbinom(3,3,0.3)

# Probability of closing <= 1 deal out of 3 deals

pbinom(1,3,0.3)

# Probability of closing > 1 deal out of 3 deals

pbinom(1,3,0.3, lower.tail=FALSE)

-How many sales will be won

# Expected number won with 30% win rate

won\_30pct <- 3 \* 0.3

won\_30pct

# Expected number won with 25% win rate

won\_25pct <- 3\*0.25

won\_25pct

# Expected number won with 35% win rate

won\_35pct <- 3\*0.35

won\_35pct