Summary Statistics

Descriptive statistics summarize the data at hand.	Inferential statistics uses sample data to make inferences or conclusions about a larger population.
Continuous numeric data can be measured. Discrete numeric data is is usually count data like number of pets.	Nominal categorical data does not have any inherent ordering such as gender or marital status. Ordinal does have an ordering.
Mean, Median, and Mode are the typical measures of center. Mean is sensitive to outliers so use median when data is skewed. But always note the distribution and explain why you chose one measure over another.	Variance is the average, squared distance of each data point to the data's mean. For sample variance, divide the sum of squared distances by number of data points - 1.
M.A.D is the mean absolute deviation of distances to the mean.	Standard deviation is the square root of variance.
Quartiles split the data into four equal parts. 0-25-50-75-100. Thus, the second quartile is median. You can use boxplots to visualize quartiles.	Quantiles can split the data into n pieces as it is a generalized version of quartiles.
Interquartile range is the distance between the 75th and 25th percentile.	Outliers are "substantially" different data points from others. *data* < *q1-1.5*IQR* or *data* > *q3+1.5*IQR.*

Calculating summary stats in R

```
# Using food consumption data to show how to use dplyr verbs and calculate a column's summary stats.
# Calculate Belgium's and USA's "typical" food consumption and its spread.
food_consumption %>%
  filter(country %in% c('Belgium', 'USA')) %>%
  group_by(country) %>%
  summarize(mean_consumption = mean(consumption),
            median_consumption = median(consumption)
            sd_consumption = sd(consumption))
# Make a histogram to compare the distribution of rice's carbon footprint. A great way to understand how skewed is the variable.
food_consumption %>%
  # Filter for rice food category
  filter(food_category == "rice") %>%
  # Create histogram of co2_emission
  ggplot(aes(co2_emission)) +
    geom_histogram()
# Calculate the quartiles of co2 emission
quantile(food_consumption$co2_emission)
# If you want to split the data into n pieces. This is equivalent of splitting the data into n+1 quantiles.
quantile(food_consumption$co2_emission, probs = seq(0, 1, 1/n).
# Calculate variance and sd of co2_emission for each food_category
food_consumption %>%
  group_by(food_category) %>%
```

Calculating summary stats in R (cont)

```
  summarize(var_co2 = var(co2_emission),
     sd_co2 = sd(co2_emission))
# Plot food_consumption with co2_emission on x-axis
ggplot(data = food_consumption, aes(co2_emission)) +
  # Create a histogram
  geom_histogram() +
  # Create a separate sub-graph for each food_category
  facet_wrap(~ food_category)
```

Random Numbers and probability

p(event) = # ways event can happen / total # of possible outcomes.	Sampling can be done with our without replacement.
Two events are independent if the p(second event) is not affected by the outcome of first event.	A probability distribution describes the probability of each outcome in a scenario.
The expected value is the mean of a probability distribution.	Discrete random variables can take on discrete outcomes. Thus, they have a discrete probability distribution.
A bernouli trial is an independent trial with only two possible outcomes, a success or a failure.	A binomial distribution is a probability distribution of the number of successes in a sequence of n bernoulli trials. Described by two parameters: number of trials (n) and pr(success) (p).
The expected value of a binomial distribution is n * p.	Ensure that the trials are independent to use the binomial distribution.

- When sampling with replacement, you are ensuring that p(event) stays the same in different trials. In other words, each pick is independent.
- Expected value is calculated by multiplying each value a random variable can take by its probability. and summing those products.
- Uniform distribution is when all outcomes have the same probability.

Sampling and Distributions in R

```
# Randomly select n observations with or without replacement
df %>% sample_n(# of obvs to sample, replace=TRUE or FALSE).
# Say you assume that the probability distribution of a random variable (wait time for ex.) is uniform,
where it takes a min value and a max value. Then, the probability that this variable will take on a value
less than x can be calculated as:
punif(x, min, max)
# To generate 1000 wait times between min and max.
runif(1000, min, max).
# Binomial distribution ------------------------------------
rbinom(# of trials, # of coins, pr(success))
rbinom(1, 1, 0.5) # To simulate a single coin flip
rbinom(8, 1, 0.5) # Eight flips of one coin.
rbinom(1, 8, 0.5) # 1 flip of eight coins. Gives us the total # of successes.
dbinom(# of successes, # of trial, pr(success)).
dbinom(7, 10, 0.5) # The chances of getting 7 successes when you flip 10 coins.
pbinom(7, 10, 0.5) # Chances of getting 7 successes or less when you flip 10 coins.
pbinom(7, 10, 0.5, lower.tail = FALSE) # Chances of getting more than 7 successes when you flip 10 coins.
```

More distributions and the CLT

The Normal distribution is a cotinuous distribution that is symmetrical and has an area beaneath the curve is 1.	It is described by its mean and standard deviation. The standard normal distribution has a mean of 0 and an sd of 1.
Regardless of the shape of the distribution you're taking sample means from, the central limit theorem will apply if the sampling distribution contains enough sample means.	The sampling distribution is a distribution of a sampling statistic obtained by randomly sampling from a larger population.
To determine what kind of distribution a variable follows, plot its histogram.	The sampling distribution of a statistic becomes closer to normal distribution as the number of trials increase. This is known as the CLT, and the sample must be random and independent.
A Poisson process is when events happen at a certain, and a known, rate but completely at random.	For example, we know that there are 2 earthquakes every month in a certain area, but the timing of the earthquake is completely random.
Thus, the poisson distribution shows us the probability of some # of events happening over a fixed period of time.	The poisson distribution is described by lambda which is the average number of events per time interval.
The exponential distribution allows us to calculate the probability of time between poisson events; Probability of more than 1 day between pet adoptions. It is a continuous distribution and uses the same lambda value.	The expected value of an exponential distribution is 1/lambda. This is the rate.
(Student's) t-distribution has a similar shape as the normal distribution but has fatter tails.	Degrees of freedom (df) affect the t-distribution's tail thickness.
Variables that follow a log-normal distribution have a logarithm that is normally distributed.	There are lots of others.

-The peak of Poisson distribution is at its lambda.
-Because we are counting the # of events, the Poisson distribution is a discrete distribution. Thus, we can use dpois(), and other probability functions we have seen so far.
-Lower df = thicker tails and higher sd.

More distributions and the CLT in R

```
# Say you're a salesman and each deal you worked on was worth different amount of money. You tracked every deal you worked on, and the amount column follows a normal distribution with a mean of $5000 and sd of $2000.
# Pr(deal < $7500):
pnorm(7500, mean=5000, sd=2000)
# Pr(deal > 1000)
pnorm(1000, mean=5000, sd=2000, lower.tail=FALSE)
# Pr(deal between 3000 and 7000)
pnorm(7000, mean=5000, sd=2000) - pnorm(3000, mean=5000, sd=2000)
# How much money will 75% of your deals will be worth more than?
qnorm(0.75, mean=5000, sd=2000)
# Simulate 36 deals.
rnorm(36, mean=5000, sd=2000)
CLT in action----------------------------------------
# Say you also tracked how many users used the product you sold in num_users column. The CLT, in this case, says that the sampling distribution of the average number of users approaches the normal distribution as you take more samples.
```

More distributions and the CLT in R (cont)

```
# Set seed to 104
set.seed(104)
# Sample 20 num_users from amir_deals and take mean
sample(amir_deals$num_users, size = 20, replace = TRUE) %>%
  mean()
# Repeat the above 100 times
sample_means <- replicate(100, sample(amir_deals$num_users, size = 20, replace = TRUE) %>% mean())
# Create data frame for plotting
samples <- data.frame(mean = sample_means)
# Histogram of sample means
samples %>% ggplot(aes(x=mean)) + geom_histogram(bins=10)
```

Correlation and Experimental Design

The correlation coefficient quantifies a linear relationship between two variables. Its magnitude corresponds to strength of relationship.	The number is between -1 and 1, and the sign corresponds to the relationship's direction.
The most common measure of correlation is the Pearson product-moment correlation (r).	Don't just calculate r blindly. Visualize the relationship first.
Sometimes , you must transform one or both variables to make a relationship linear and then calculate r.	The transformation choice will depend on your data.
And as always, correlation does not imply causation. You must always think of confounding or hidden variables.	Experiments try to understand what is the effect of the treatment of the response.
In a randomized control trial, participants are randomly assigned by researchers to treatment or contol group.	In observational studies, participants are not randomly assigned to groups. Thus, they establis causation.
In a longitudinal study, participants are followed over a period of time to examine the treatment's effect.	In a cross-sectional study, data is collected from a single snapshot of time.

- Measures the strength of only linear relationship.
- Use a scatterplot and add a linear trend line to see a relationship between two variables.
- Other transformations include taking square root, taking reciprocal, Box-Cox transformation, etc.

Correlation and design in R

```
# Make a scatter plot to view a bi-variate relationship
df %>% ggplot(aes(x=col_1, y=col_2)) + geom_point() +
 geom_smooth(method='lm', se=FALSE (usually)).
# Measure the correlation between two data frame columns
cor(df$col_1, df$col_2)
# Transform the x variable to log.
df %>% mutate(log_x = log(col_x)) %>% # Natural log by default
 ggplot(aes(x=log_x, y=col_y)) + geom_point() +
 geom_smooth(method='lm', se=FALSE).
```