Statistics Primer With R

And a Review of Functions

Outline

Basics of Statistics!

- Descriptive Measures
- Randomnes
- Probability Distributions
- Inference

Looking forward: Regression and Modeling!

Defining The Terms

Data - The information we gather from experiments and surveys

Statistics - The art and science of learning from data! More technically, a statistic is any summary of data. Thus, statistics is the study of those summaries.

Statistical / Data Science Process:

1. Formulate Statistical Question
2. Collect Data
3. Analyze Data
4. Interpret Results

Types of Statistics

Descriptive Statistics: methods for summarizing the collected data. Summaries consist of graphs and numbers such as averages and percentages.

Inferential Statistics: methods of making decisions or predictions about a population based on data obtained from a sample of that population.

Descriptive Measures

In Stats I, you should have learned about a bunch of statistics (summaries of data)! R has built-in functions for nearly all of them!

Center: `mean()` `median()` No default mode

Spread: `sd()` `var()` `range()` `IQR()` `mad()`

Percentiles: `quantile()`

Your Turn!

Even though there is a built-in IQR function, I want you to create your own IQR function called **`iqr()`**!

Use the template below as a starting point:

```
iqr <- function(data) {
}
```

Check: `iqr(1:100)`

```
49.5
```

Randomness

Randomness is key to Statistics!

Random Variable - numerical variable whose outcome is not known until after the event happens.

Situation: Flipping a fair coin 10 times

RV: Counting the number of heads out of 10 flips

Characterizing the behavior of those random variables through probability is of paramount importance in statistics!

Your Turn!

I want you to make a function that simulates flipping a fair coin for a specified number of times

Hint: The key function you want to use is `sample()`. Check the function out, it is pretty cool!

Template:

```
flips <- function(n) {

}
```

Probability Distributions

The way in which we quantify randomness is with a probability distribution.

Probability Distribution - describes all the possible outcomes of the random variable and their associated probabilities

Common families of distributions have names like Normal/ Gaussian, binomial, Poisson, Chi-Squared, t, F, etc.

They also have common functional forms. For instance the density function of the Normal distribution is

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

Probability Distributions

R has almost any probability distribution that you can think of!

There are 4 different “characteristics” of a distribution that we are interested in.

Probability density/mass functions (pdf, pmf) with **d__()**

`dnorm()` `dt()` `dchisq()` etc.

Cumulative distribution functions (cdf) with **p__()**

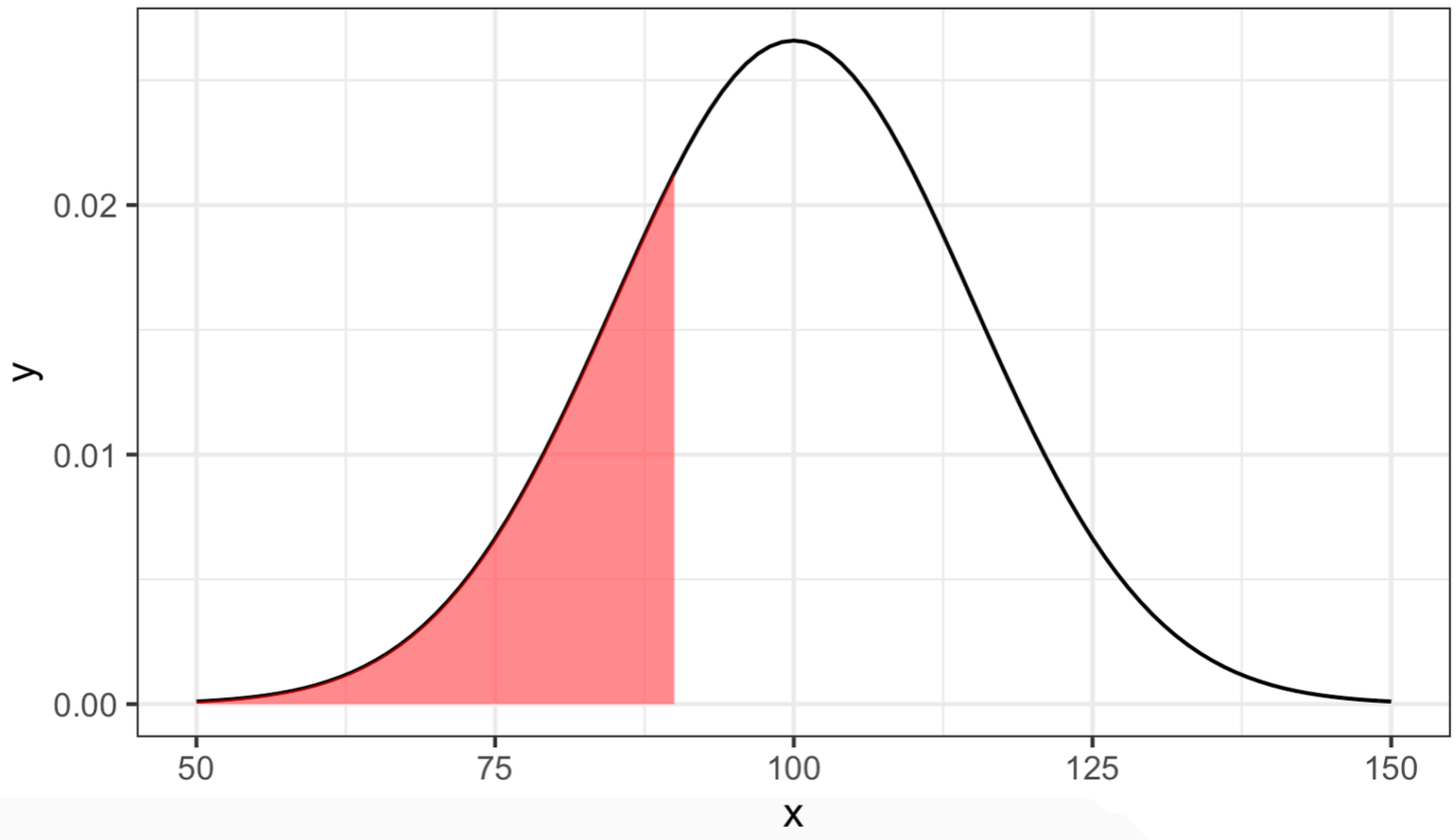
Inverse CDF’s (quantile function) with **q__()**

Random samples from a distribution with **r__()**

Probability Distributions

Examples: Suppose that $X \sim N(100,15)$. Find $P(X < 90)$

```
ggplot(data = data.frame(x = c(50, 150)), aes(x)) +
  stat_function(fun = dnorm, args = list(mean = 100, sd = 15)) +
  geom_area(stat = "function", fun = dnorm, fill = "red", xlim = c(50, 90),
            args = list(mean = 100, sd = 15), alpha = 0.5)
```

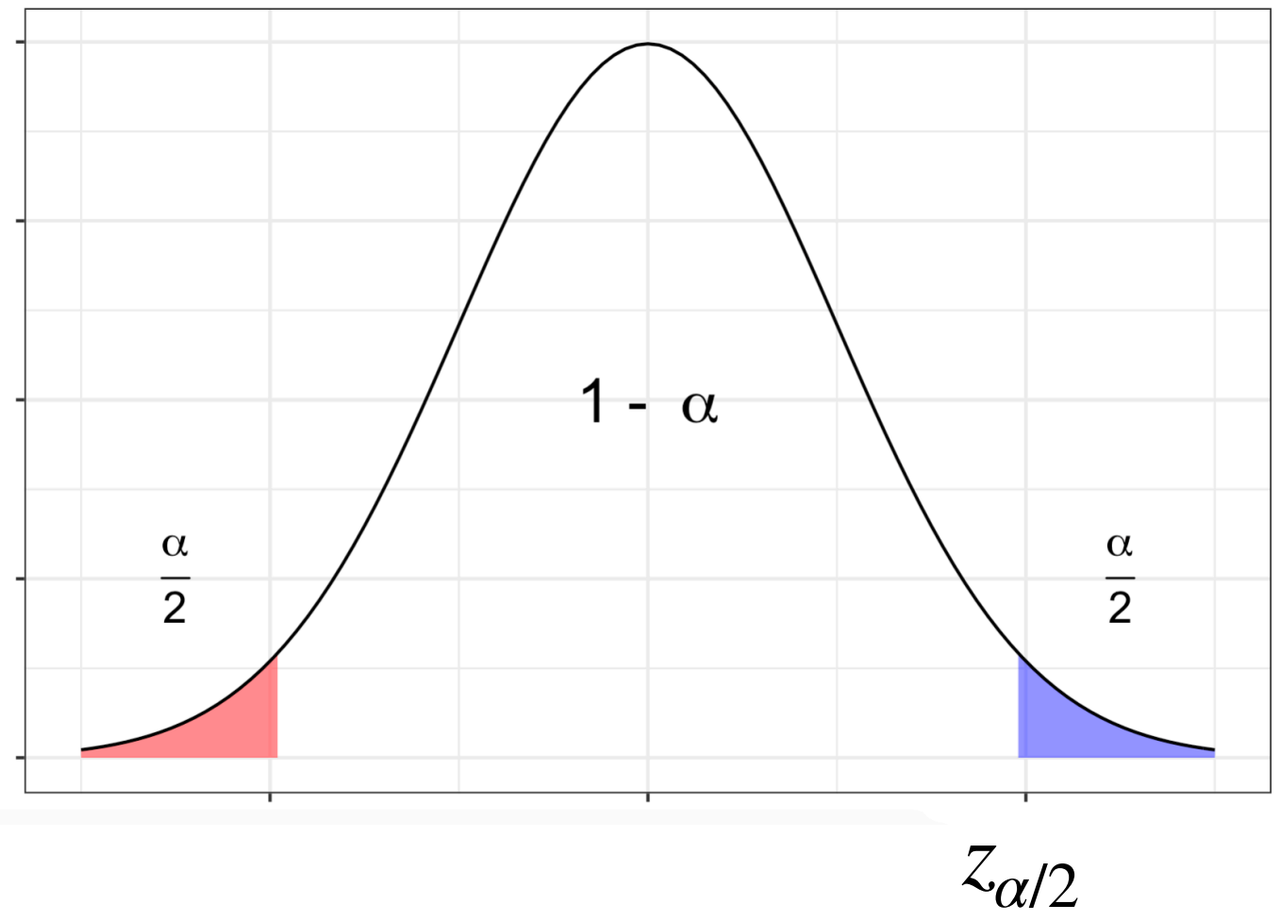


Solution: `pnorm(90, mean = 100, sd = 15)` `[1] 0.2524925`

Probability Distributions

Examples: Finding a confidence factor for a confidence interval for a population proportion.

$$\hat{p} \pm z_{\alpha/2} \times \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$



95% CI

Solution:

```
qnorm(.025, mean = 0, sd = 1, lower.tail = FALSE)
```

```
[1] 1.959964
```

Your Turn!

I want you to make your own version of dnorm() called my_dnorm().

Hints: $f(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}$ is the density function

In R, e is **exp()**

Template:

```
my_dnorm <- function(x, mean, sd) {

}
```

Check:

```
my_dnorm(0, 0, 1)
```

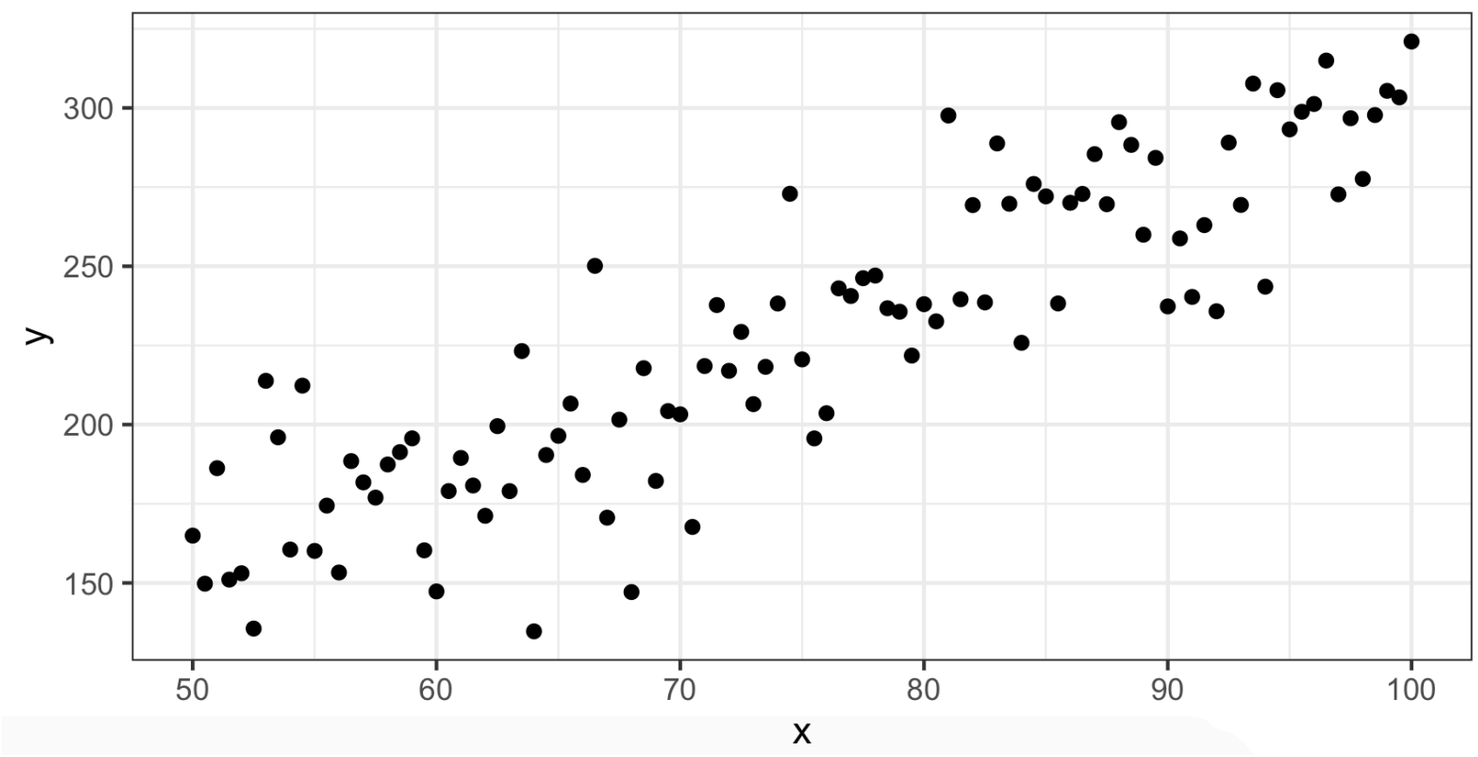
```
[1] 0.3989423
```

Probability Distributions

The r__() functions are the most important and the ones we will use the most!

Being able to simulate draws from a distribution is a crucial part of statistics and data science!

Example: How can you generate data that comes from a linear regression model?



Linear Regression

The linear regression model: $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$

Common assumption: $e_i \sim N(0, \sigma^2)$

Example:

```
data <- tibble(
    x = seq(50, 100, by = .5),
    y = 3*x + 5 + rnorm(length(x), 0, 25)
)
```

```
lm(y ~ x, data = data)
```

```
Call:
lm(formula = y ~ x, data = data)

Coefficients:
(Intercept)            x
      6.505        2.949
```

Your Turn!

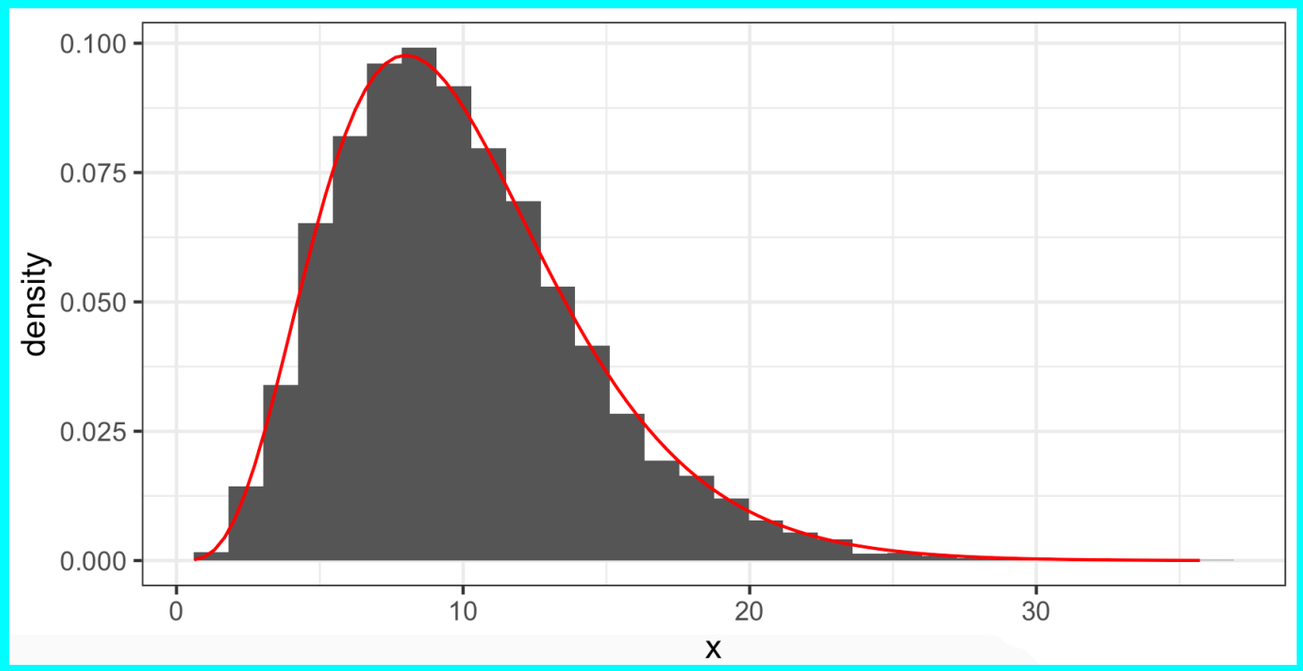
Challenge Round!

I want you to generate 10000 draws from a χ^2 (chi-squared) distribution with 10 degrees of freedom. Then overlay the density function in red!

Hints:

1. rchisq() for random generation
2. dchisq() in the stat_function() call
3. Use ..density.. in the y aesthetic when making the histogram

Check: Should look like this!



Statistical Inference

Other than prediction/classification, inference is what people do with statistics!

Two types of inference in Stats I: estimation and testing

Because R was made for statistics, it has great support for inference!

CI's and hypothesis tests for proportions: `prop.test()`

CI's and hypothesis tests for means: `t.test()`

Your Turn!

Challenge Round!

Since all of the confidence intervals are built into the test functions, there is no such thing as a t.interval. So I want you to make one called t_interval (because . are for losers)!

t interval formula: $\bar{x} \pm t_{n-1,\frac{\alpha}{2}} \times \frac{s}{\sqrt{n}}$

sample mean

df

sample std dev

Sample size

Template:

```
t_interval <- function(data, conf_level = 0.95) {

}
```