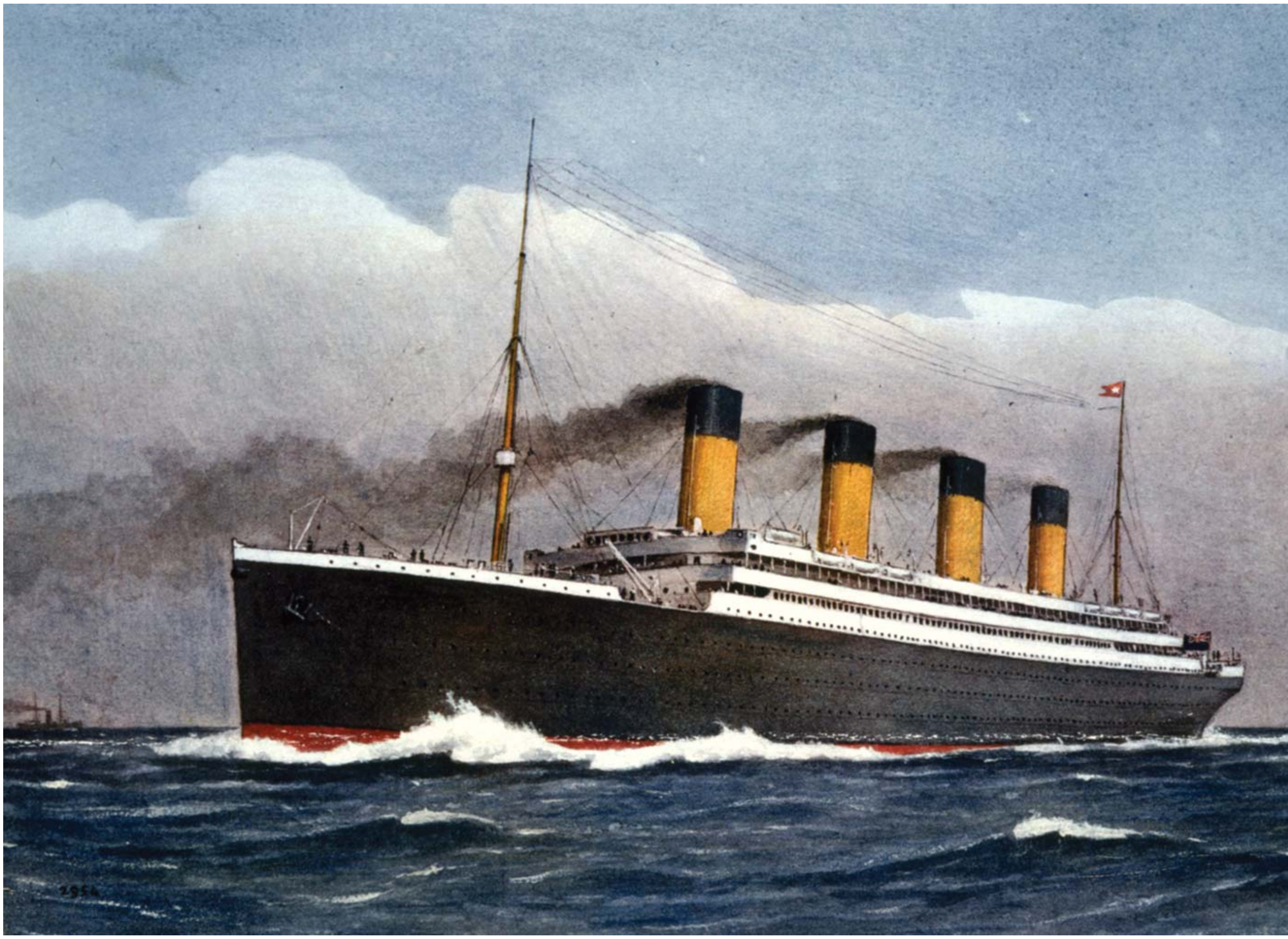
Lecture 3.1 - Practicing Confidence Intervals

Student

2025-04-10

Table of contents



Interpretation steps

Let's practice with a dataset of the passengers of the Titanic, from the dataset `titanic_survival.csv`.

Setting things up

1. Create new variable called `survivedindicator`; assign it a value of 0 if the passenger did not survive and 1 if the passenger did survive using the `case_when` verb (information on how to do that can be found here or here) . Find the proportion

that survived in the entire dataset (you can do this by using the `table()` command: `table(titanic$survivedindicator)`.

Random matters

Now let's sample from the dataset. We can sample whether or not they survived by running the following code (note: you need the library `dplyr` loaded for this code):

```
titanic_n50 <- titanic_survival %>%
  slice_sample(survivedindicator, n=50)
```

Create a sample of size 50 and 200.

2. Calculate by hand the standard deviation and 95% confidence interval for `survivedindicator` of your sample of 50 and 200.
3. Interpret these confidence intervals
4. Find the proportion that survived of the entire dataset – was it inside or outside the standard error of your confidence intervals for 50 and 200? Why was it inside or outside?
5. If you sampled many times, how many sample proportions would be inside or outside of your confidence interval you just created?

Sampling distributions

Let's now add the following command to the `setup` block (copy and paste the entire part and then run your `setup` code block again:

```
prop.multiple.samples <- function(n, numsamples, variable) {
     meanvector <- c()
     meanonesample <- 0
     for (i in 1:numsamples) {
       meanonesample <- mean(sample(variable, n, replace=TRUE))
         meanvector[i] <- meanonesample
     }
     meanvector
}
```

This defines a new function in R called `prop.multiple.samples()`. It takes as its arguments the sample size (`n`), the number of samples (`numsamples`) and the variable from which you would like to create a sampling distribution. Once you have created this function, you can use it as follows:

```
titanic_n50_s100 <-prop.multiple.samples(50, 100, titanic_survival$survivedindicator)
```

This takes 100 samples of $n = 50$ from the variable `titanic_survival$survivedindicator`. In practical terms, this line of code draws 50 people at random from the dataset 100 times and then calculate the proportion in each sample of the number of people surviving.

This process creates a pseudo sampling distribution.

Observing the sampling distribution

6. Make a histogram using `ggplot` of the results of `titanic_n50_s100`. What does this histogram show? Interpret this carefully.
7. Calculate the `sd()` of `titanic_n50_s100` - what is this quantity indicate? What calculation should it be equal to? Why?
8. If you increased the number of items sampled (`n` or the first entry in `prop.multiple.samples`) what do you think will happen to your histogram? How about the `sd()` you calculated?. How about if you increased `numsamples` instead?

Comparing the distribution

9. Increase the `n` and `numsamples` separately (try values like 200 and 500 and 10000). How does the shape and distribution of the histogram change? Did it match your expectations? Why or why not?

Extra activity - modeling

While it is more common to use a non-linear link function (logistic regression) to model an outcome variable with a 0 or 1 outcome, in this case please try to use linear regression make a model that predicts what factors are most important in predicting survival on the Titanic.