Inference for Regression: Test for Linear Association Between Two Quantitative Variables

Thomas Scofield

May 04, 2020

The Model Utility Test

There are things you can do whenever you have bivariate quantitative data, such as

- produce a scatterplot of the data.
- calculate the (sample) correlation r.
- find the slope b_1 and intercept b_0 (both *sample statistics*) of the least squares regression line.

As we discussed in Chapter 2, the correlation is not always *meaningful*. But, in this chapter, we will assume that it is—that we have variables X and Y where the average response value $\mu_Y(x)$ at any particular value of X is given linearly as

$$\mu_Y(X) = \beta_0 + \beta_1 X, \qquad \text{— true line}$$

b_0 estimates β_0; b_1 " β_1

making it meaningful to discuss the true correlation ρ.

An association between X and Y exists if the true slope $\beta_1 \neq 0$ or, equivalently, if the true correlation $\rho \neq 0$. Otherwise the variables are independent, meaning X has no value in predicting Y. To conduct a test, called the **model utility test**, of

① $\mathbf{H}_0\colon\ \beta_1 = 0 \quad \text{vs.} \quad \mathbf{H}_a\colon\ \beta_1 \neq 0,$

or equivalent stated as

② $\mathbf{H}_0\colon\ \rho = 0 \quad \text{vs.} \quad \mathbf{H}_a\colon\ \rho \neq 0,$

r estimates ρ

we will need sample data, producing a sample slope b_1 or sample correlation r.

Scatterplots; calculating b_1, r in R

The data set found at [http://scofield.site/teaching/data/csv/heartDiseaseDeathsAndWine.csv][http://scofield.site/teaching/data/csv/heartDiseaseDeathsAndWine.csv] contains a variable `winealc` that measures wine consumption (measured in liters per person per year) in various countries and another variable `hddeaths` which measures heart disease mortality rates (deaths per thousand). We import this data, view a scatterplot, and calculate the sample values.

chosen at time of import

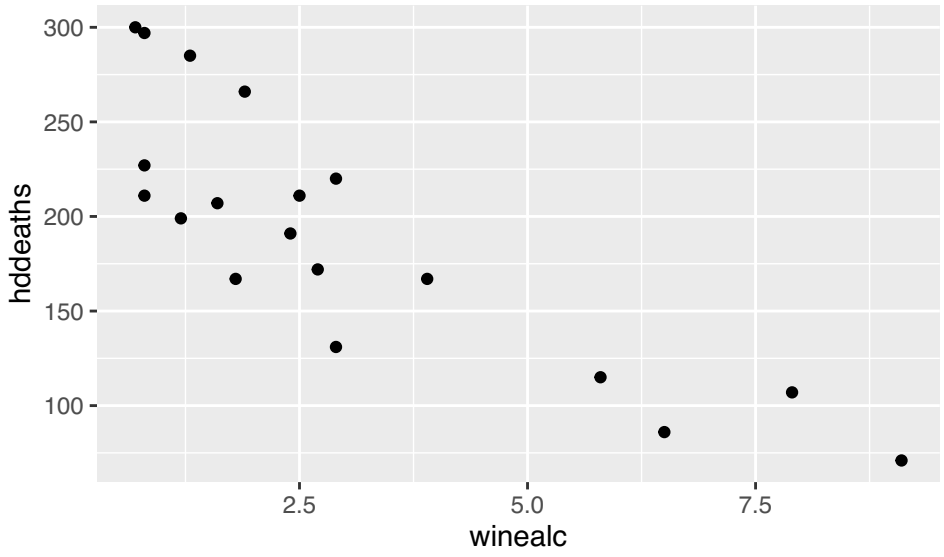
```
hdAndWine <- read.csv("http://scofield.site/teaching/data/csv/heartDiseaseDeathsAndWine.csv")
head(hdAndWine)
```

```
##       country winealc hddeaths
## 1   Australia     2.5      211
## 2 Netherlands     1.8      167
## 3     Austria     3.9      167
## 4 New Zealand     1.9      266
## 5     Belgium     2.9      131
## 6      Norway     0.8      227
```

bivariate quantitative data

In making a scatterplot, we must decide which variable (between `winealc` and `hddeaths`) to consider explanatory, placing it on the horizontal axis. Either could serve in that role, but in most discussions involving these variables, it is the alcohol consumption that people generally adopt as explanatory. So, it appears on the right side of the tilde in the command

```
gf_point(hddeaths ~ winealc, data=hdAndWine)
```



We can get the coefficients (intercept b_0 and slope b_1) of the best-fit line for the data via the command

```
lm(hddeaths ~ winealc, data=hdAndWine)
```

```
##
## Call:
## lm(formula = hddeaths ~ winealc, data = hdAndWine)
##
## Coefficients:
## (Intercept)     winealc
##      260.56      -22.97
```

b_0 b_1 Both are sample statistics

Note that, by adding `$coefficients`, the output is less "wordy",

```
lm(hddeaths ~ winealc, data=hdAndWine)$coefficients
```

```
## (Intercept)     winealc
##   260.56338   -22.96877
```

and by altering this to '$coefficients[2]$ we obtain just the slope.

```
lm(hddeaths ~ winealc, data=hdAndWine)$coefficients[2]
```

```
##   winealc
## -22.96877
```

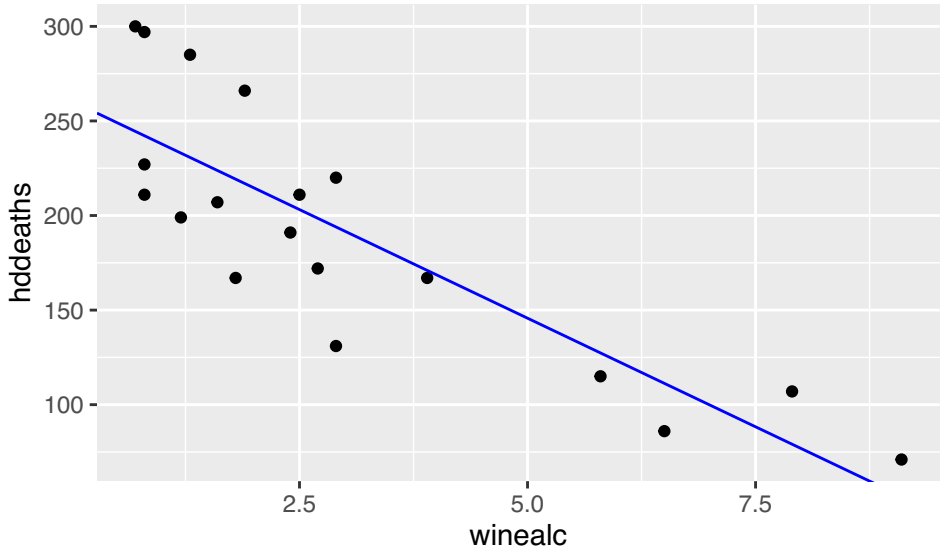
test statistic — produced b_1

Note, ending in $coefficients[1] would produce b_0

This last version, isolating the response to *slope* only, will be helpful in generating randomization distributions for b_1.

We can overlay the best-fit line by "piping" the scatterplot to the `gf_abline()` command with specified slope and intercept:

```
gf_point(hddeaths ~ winealc, data=hdAndWine) %>%
  gf_abline(slope = ~-22.97, intercept = ~260.56, color="blue")
```



To calculate the sample correlation, instead, we change `lm()` to `cor()`:

```
cor(hddeaths ~ winealc, data=hdAndWine)
```

```
## [1] -0.8428127
```

— sample correlation r, estimates true correlation ρ.

Question

Would we get the same slope and intercept if we exchanged the roles of the variables, in this case making `hddeaths` the explanatory variable? Would we get the same correlation?

Randomization

Randomization distributions are meant to simulate the null distribution—what sort of values we expect, and how frequently, out of our sample statistic when the null hypothesis (no association between the quantitative variables) holds. We simulate it by shuffling one of the variables.

Relevant if $H_0: \beta_1 = 0$, $H_a: \beta_1 \neq 0$

→ **Randomization distribution for b_1**: In the context of our *wine-and-heart-disease-deaths* data, one randomization statistic b_1 arises from

```
lm(hddeaths ~ shuffle(winealc), data=hdAndWine)$coefficients[2]
```

Draw w/out replacement

match

```
## shuffle(winealc)
##         8.703485
```

We get an approximate P-value when we generate lots of these randomization statistics, locate our test statistic (the slope for the original data), and determining how often something that extreme (or more so) occurs:

```
manyb1s <- do(5000) * lm(hddeaths ~ shuffle(winealc), data=hdAndWine)$coefficients[2]
head(manyb1s)
```

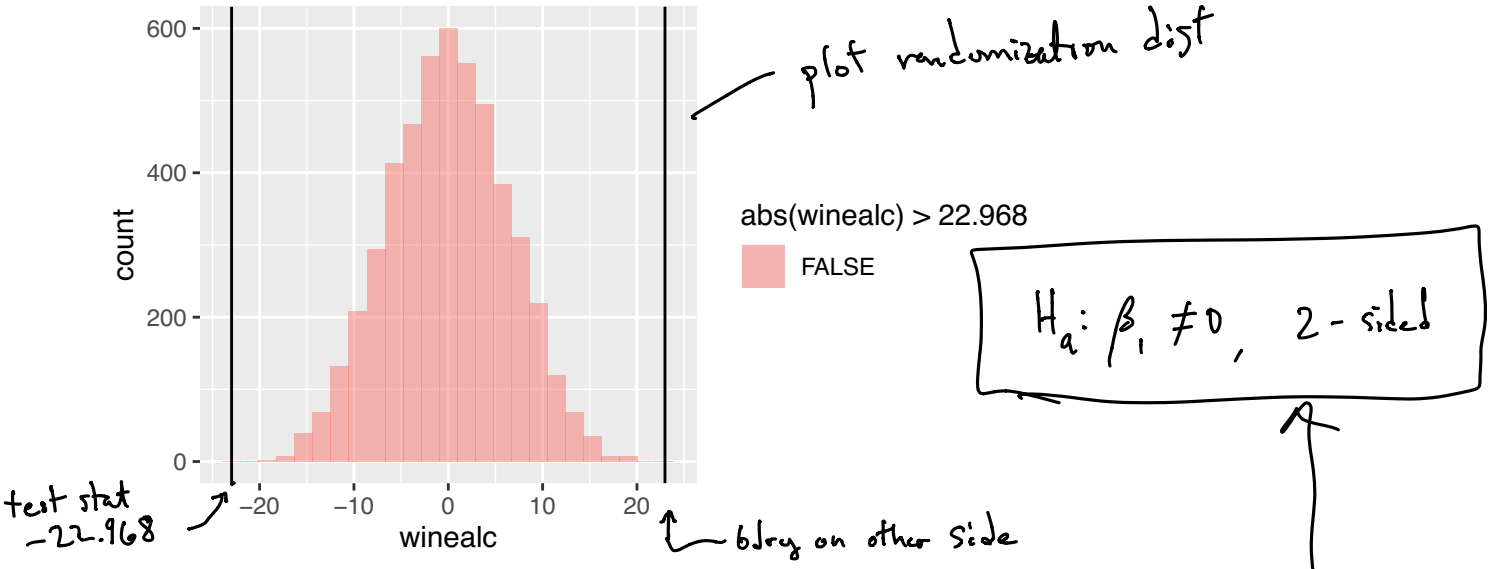
```
##      winealc
## 1  2.1430627
## 2 -3.0308334
## 3  4.2475489
## 4  0.5924815
## 5 -2.2687730
```

— All produced under assumption of "no association"

```
## 6  3.6019144
```

This randomization distribution with the test statistic can be visualized in a manner like we have used before:

```
gf_histogram(~winealc, data=manyb1s, fill = ~abs(winealc)>22.968) %>%
  gf_vline(xintercept = ~22.968) %>%
  gf_vline(xintercept = ~-22.968)
```



This graph makes it appear that occurrences of $b_1 = 22.96877$, or even further distant from 0, are extremely rare. Indeed, if we count how often it happened in our 5000 tries, we have

```
2 * nrow( filter(manyb1s, winealc < -22.968) ) / 5000
```

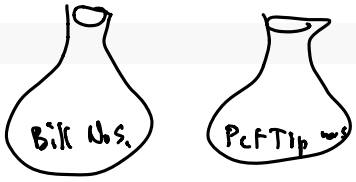
```
## [1] 0
```
— approx. P-value 0 — reject H_0 in favor of this

Randomization distribution for r: Consider another dataset, the **RestaurantTips** data from the `Lock5withR` package. The question here is whether there is an association between a bill for the meal at a restaurant, and the tip as a percentage-of-the-bill-as-tip (variable name `PctTip`). There are other ways to state this question of association. In the text, the Locks state it (roughly) as, "Is `Bill` an effective predictor of the size of the tip as a percentage of the bill?" The null hypothesis says, "no, it isn't", or $\rho = 0$.

To generate a randomization distribution for r under this null hypothesis, we start with the command that generates the r from the original data:

```
cor(PctTip ~ Bill, data=RestaurantTips)
```

```
## [1] 0.1352976
```
— test statistic r



That is our test statistic.

The generation of a single randomization statistic r comes from shuffling one of the variables:

```
cor(PctTip ~ shuffle(Bill), data=RestaurantTips)
```

```
## [1] -0.01835389
```
— one randomization stat (r)

If we do this often, we get a randomization distribution for r. We locate our test statistic on this distribution and use it as a boundary to determine the P-value.

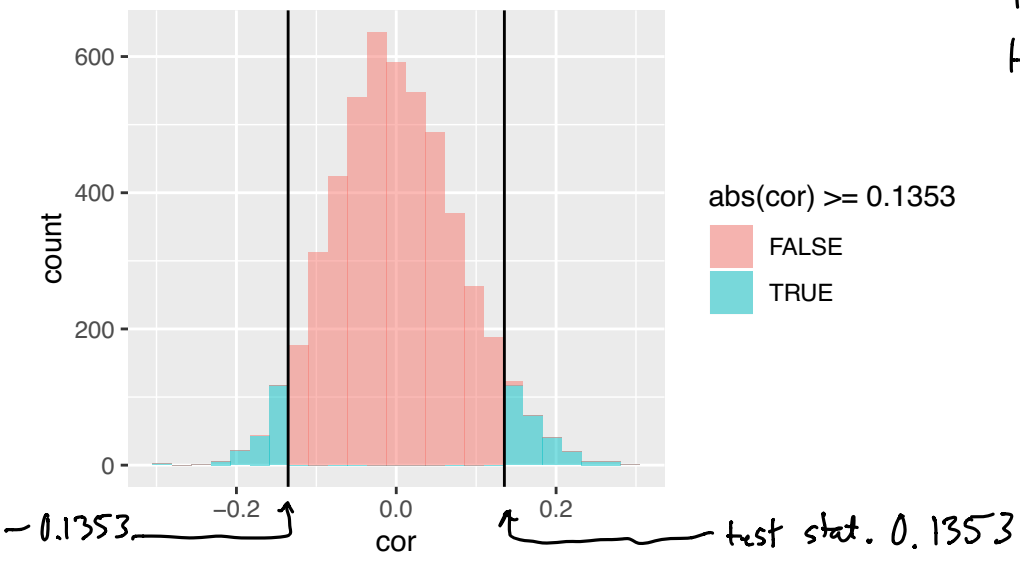
```
manyCors <- do(5000) * cor(PctTip ~ shuffle(Bill), data=RestaurantTips)
head(manyCors)
```

```
##           cor
```

```
## 1  0.120342237
## 2 -0.041139911
## 3 -0.020694148
## 4 -0.084496966
## 5  0.199104121
## 6  0.003735002
```

```
gf_histogram(~cor, data=manyCors, fill= ~abs(cor) >= 0.1353) %>%
  gf_vline(xintercept = ~0.1353) %>%
  gf_vline(xintercept = ~-0.1353)
```

→ $H_0: \rho = 0$
$H_a: \rho \neq 0$ (2 sided)




```
nrow( filter(manyCors, abs(cor) >= 0.13529) ) / 5000
```

```
## [1] 0.091
```
← Approx. P-value

This P-value is not smaller than $\alpha = 0.05$, so at that level we fail to reject the null hypothesis, that the true correlation ρ (and the true slope β_1) is zero. That is, if someone tends to think that patrons of restaurants tip the same percentage regardless of the overall tab, we have not found here evidence sufficient to conclude otherwise.

Exercise

Throughout the discussion above, it has been implied that it didn't matter which test statistic you randomized, r or b_1. Convince yourself that this is the case by playing with randomization distributions and P-values for bivariate quantitative data in StatKey. The confirmation that "it does not matter" would be that, when working with a fixed dataset, when you generate a P-value corresponding to your test statistic r, it is roughly the same as the P-value corresponding to your test b_1.

Model Utility Test without simulation/randomization: Simple Linear Regression model

People have been conducting the Model Utility Test for a long while, since before computers were on every desktop, before it was feasible to generate randomization distributions, when only hand calculations were possible. Upon request, R will generate the kind of results those hand calculations produced. Naturally the `lm()` command is used, but so is `summary()`. In the case of the *PctTip-and-Bill* data, the request goes like this:

Seen above

```
summary( lm(PctTip ~ Bill, data=RestaurantTips) )
```

```
##
## Call:
## lm(formula = PctTip ~ Bill, data = RestaurantTips)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9927 -2.3096 -0.6455  1.4679 25.5335
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 15.50965    0.73956   20.97   <2e-16 ***
## Bill         0.04881    0.02871    1.70   0.0911 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.36 on 155 degrees of freedom
## Multiple R-squared:  0.01831,    Adjusted R-squared:  0.01197
## F-statistic:  2.89 on 1 and 155 DF,  p-value: 0.09112
```

$H_0: \beta_1 = 0$

Standardizing

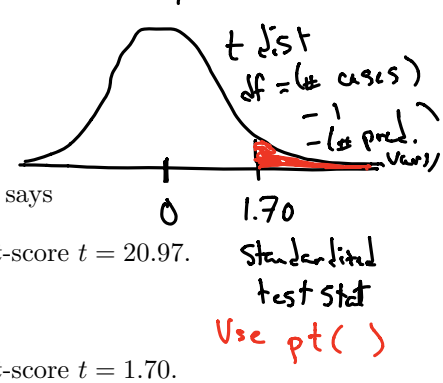
$$\text{z or t} = \frac{(\text{sample stat.}) - (\text{hyp. value})}{(\text{Std. error})}$$

$$t = \frac{b_1 - 0}{SE_{b_1}} = \frac{0.04881}{0.02871} = 1.70$$

b_0, SE_{b_0}, b_1, SE_{b_1}

← not interesting

← used for inference on β_1



Use pt( )

Notice the result contains information about both coefficients. The `(Intercept)` row says

> Our point estimate b_0 for β_0 is 15.50965, has $\text{SE}_{b_0} = 0.73956$, and standardized t-score $t = 20.97$.

The other row, the one about *slope*, is always of greater interest to us. It says

> Our point estimate b_1 for β_1 is 0.04881, has $\text{SE}_{b_1} = 0.02871$, and standardized t-score $t = 1.70$.

We see that the process our forbears devised in lieu of randomization led them to deal, again, with t-distributions. Specifically, since in the Model Utility Test we hypothesize that $\beta_1 = 0$, standardizing leads to the reported t-score:

$$t = \frac{0.04881 - 0}{0.02871} = 1.700.$$

To get the resulting P-value the way they (and this command) did, we find how often a t-score is as extreme or more so than this one—i.e., compute the tail area and double it. We get the tail area using a t-distribution, but with how many degrees of freedom? If there are n cases in the dataset, regression computes degrees of freedom in this way:

$$\text{df} = n - 1 - (\text{number of predictor variables}).$$

What is implied here is that it is possible to consider more than 1 explanatory/predictor variable. When we do, it is called **multiple regression**, a topic we will not cover in this course. Since we are considering only 1 predictor variable, the number of degrees of freedom is $\text{df} = n - 2$. So, in the case of *PctTip-as-predicted-by-Bill* data, which has $n = 157$ cases, R determined its P-value above by doing what the following command does:

```
2 * (1 - pt(1.7, df=155))
```

```
## [1] 0.09113667
```

gives shaded area above

Question:

Our forbears also developed the formula for the standardized t-statistic of the sample correlation r to be

$$t = \frac{r\sqrt{n-2}}{\sqrt{1-r^2}},$$

= same as t-score for b_1

(= 1.7 again for our Restaurant Tips data.)

which also has $n-2$ degrees of freedom when there is one predictor variable. What is the resulting P-value, and what hypotheses would we be testing?

Caution about using results based on a theoretical t-distribution: As in the previous chapters, beginning with Chapter 5, whenever we have turned to a theoretical distribution (a normal distribution, a t distribution, a chi-square distribution, or an F distribution) to compute a P-value, there have been conditions which validate the approach and, in the absence of such, leave us in some doubt about the conclusions. The same is true with the results above.

What is assumed may be described like this: no matter where you look X-wise, data points are centered on the (ideal) regression line with the same spread. Some pictures from the textbook:

~Good

~Bad

~Bad

~Bad