Stat 145, Thu 22-Apr-2021 -- Thu 22-Apr-2021
Biostatistics
Spring 2021

Thursday, April 22nd 2021

Wk 12, Th
Topic:: Regression inference, randomization in R

Some good data sets for linear regression
- spruces: http://scofield.site/teaching/data/csv/hesterberg/Spruce.csv
  lm(Di.change ~ Ht.change, data = spruces)
- hdAndWine: .../teaching/data/csv/heartDiseaseDeathsAndWine.csv
- Lock sets
  MalevolentUnivormsNFL
  Hurricanes (double check it is the Lock5 one by this name)

$\hat{y} = a + bx$ (Chapter 2)

Some preliminaries:
- Lock Chapter 9 has a different pair of names for slope and intercept
  no longer 'a' for intercept, b for slope (as in Chapter 2)
  now it is 'b0' for intercept, b1 for slope (sample values)
    corresponding names for parameter values: beta0, beta1

$\hat{y} = b_0 + b_1 x$

True line
$\beta_0 + \beta_1 x$

- Model utility test gives focus to beta1 (or rho), ignores beta0
  why is beta0 not as important?
  interpolation vs. extrapolation

- extracting specific info from R commands such as
  chisq.test()
  [redacted]
  anova()
  lm()

  use names() to see what info is available
    lm(...)$coefficients
    lm(...)$coefficients[2]

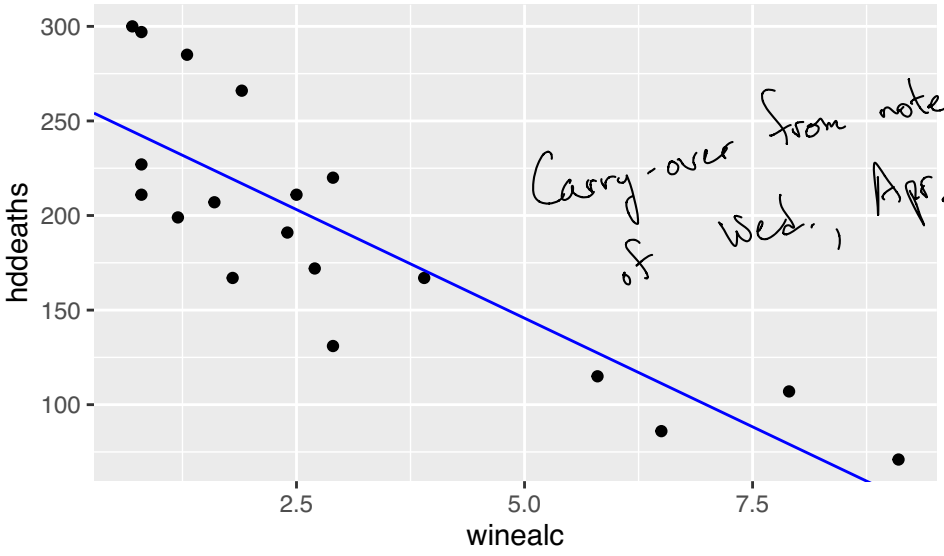
    anova(lm(Ants ~ Filling, data=SandwichAnts))["F value"]
    anova(lm(Ants ~ Filling, data=SandwichAnts))["F value"][1,1]

Model Utility Test tests hypotheses

$$H_0: \rho = 0 \quad \text{vs.} \quad H_a: \rho \neq 0$$

or, equivalently

$$H_0: \beta_1 = 0 \quad \text{vs.} \quad H_a: \beta_1 \neq 0$$



Carry-over from notes of Wed., Apr. 21

It is simpler, and achieves the same thing as above, to pipe the scatterplot to `gf_lm()`.

```
gf_point(hddeaths ~ winealc, data=hdAndWine) %>% gf_lm(color="blue")
```

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

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

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

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.

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]
```

```
## shuffle(winealc)
##        -9.047053
```

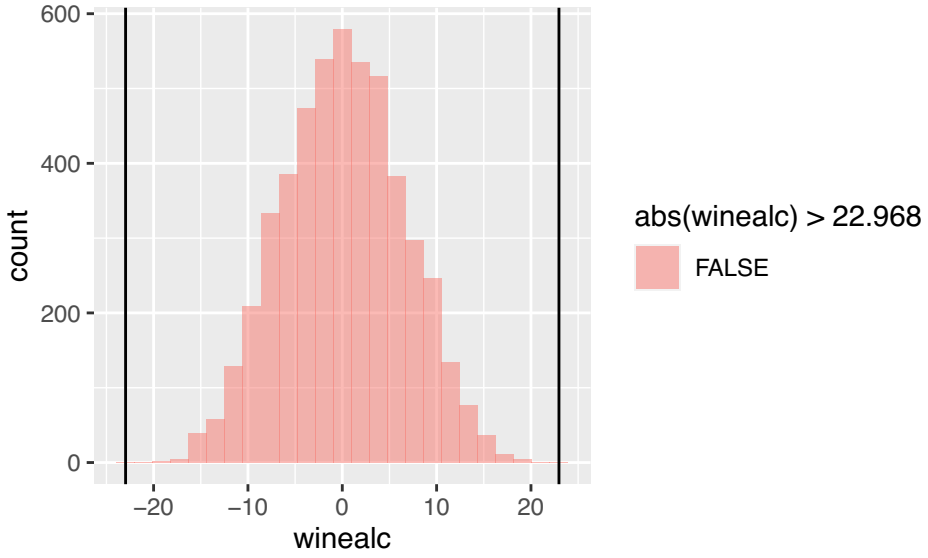
We get an approximate P-value when we generate lots of these randomization statistics, locate our test statistic (the slope for the original data), and determining how often something that extreme (or more so) occurs:

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

```
##      winealc
## 1   7.390166
## 2   6.858311
## 3   4.810274
## 4 -13.061982
## 5  -3.448026
## 6  -7.589083
```

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
```

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
```

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)
```

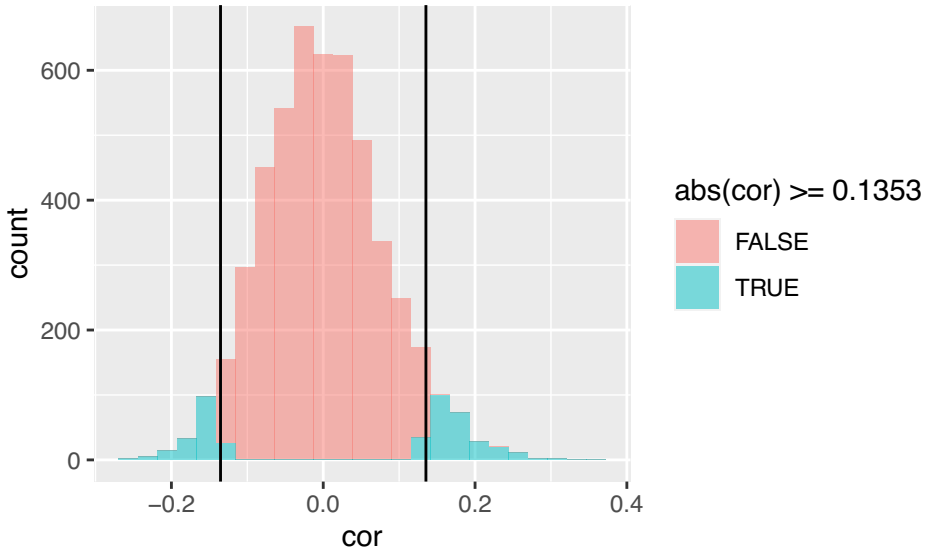
```
## [1] 0.02614511
```

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 -0.05075822
## 2  0.01201975
## 3 -0.07974203
## 4 -0.14990215
## 5  0.10869358
## 6  0.02972937
```

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




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

```
## [1] 0.0902
```

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.
