

R portfolio 2

March 17, 2020

0 big picture

The grade for each project includes 10 points for code which is clean and stylish. That means no errors. No dead ends. Include indentation, comments, and white space as appropriate. Format consistently. Observe clear and consistent naming conventions. Follow all instructions exactly. Some of these things are best accomplished after your code is working. Sometimes you will have starter code, and it's fine to use it, but make it look like your own work.

1 more investors

Investors A and B live in Alphaville, a small town with exactly one golf course and one Starbucks. They are strangers.

The data.frames **inv.dat** and **mystats** referred to below are exactly the same as in Rportfl. So you already have code to build **mystats** with columns **xbar**, **ssd**, and **nobs**. Most of the steps below ask you to add new columns to **mystats**.

Clean and stylish code is worth [10 points](#).

Investor A

Investor A has $n = 16$ investments in her portfolio. Each day she looks at the returns x_1, x_2, \dots, x_{16} on these investments. She knows they are from a normal population, but she doesn't know μ or σ . Each day she performs a hypothesis test at **significance level** $\alpha = .10$. She tests $H_0 : \mu = 5$ versus $H_A : \mu \neq 5$.

If each day's data REALLY are from $N(\mu = 5, \sigma)$, then her investment portfolio is well-balanced and likely to make money. She doesn't need to buy or sell anything. If she has evidence that $\mu \neq 5$, then her portfolio needs adjustment.

Her decision rule looks like this:

- When the data **reject** $H_0 : \mu = 5$, I go to Starbucks and adjust my portfolio. Sometimes this takes all day.
- When the data fail to reject, I can relax and hold that portfolio for another day. These are good days to play golf.

Use **inv.dat** to compute each of the following vectors and **add them to the data.frame** **mystats**. Each vector will have 20 elements.

1. (10 points) Compute the test statistic:

$$tstat = \frac{(\bar{x} - \mu_0)}{(s/\sqrt{n})}$$

This number can be negative or positive depending on \bar{x} . Very small or very large values are evidence against H_0 .

2. How small is very small? How large is very large? The boundary values for 'very small' and 'very large' are called the **critical values**. They are given by:

$$\pm t_{\alpha/2, n-1} = \underline{\hspace{10em}}$$

These don't change day to day since they depend only on α and whether the test is two-tailed or one-tailed. Let `tcrit1` be a 20-vector containing the negative critical value, and let `tcrit2` contain the positive critical value.

3. (10 points) Perform the hypothesis test. If `tstat` is **outside** the critical values, Investor A **rejects** H_0 . If `tstat` is **between** the critical values, she **accepts**. Use this rule with `ifelse()` to create vector `test1` populated with the values ACCEPT or REJECT.
4. (10 points) The p-value represents the conditional probability of getting a value of the test statistic more contradictory to H_0 than `tstat` when H_0 is true. Compute the vector of p-values like this:

$$pval = P(T_{n-1} \leq -|tstat|) + P(T_{n-1} \geq |tstat|) = 2*pt(-abs(tstat), nobs-1)$$

5. (10 points) Use the p-value to perform the hypothesis test a different way. Compare the p-value to the significance level α . If $pval \leq \alpha$, reject H_0 . Otherwise, accept H_0 . Use this rule with `ifelse()` to create vector `test2` populated with the values ACCEPT or REJECT. It should be identical to `test1`.
6. Finally, create vector `dec.A` and populate it with the values GOLF or STARBUCKS. The data.frame `mystats` should now be 20x10.

Investor B

Investor B's portfolio is similar to Investor A's. Every day he looks at the same data x_1, x_2, \dots, x_{16} , and he also believes the data are from a normal distribution. Like investor A, he is interested in testing $H_0 : \mu = 5$. His risk tolerance is similar to hers, so he uses the same α for his decisions.

Unfortunately, Investor B doesn't remember how to perform a hypothesis test, so he bases his decisions on confidence intervals. His decision rule looks like this:

- When my confidence interval **does not contain** 5, I **reject** $H_0 : \mu = 5$, head for Starbucks, and adjust my portfolio. Sometimes it's hard to find a table.
- When my confidence interval for μ **contains** 5, I **accept** $H_0 : \mu = 5$. That means I can relax and hold that portfolio for another day. These are good days to play golf.

(20 points) To determine Investor B's decisions, compute the following vectors and add them to `mystats`.

7. `lim1` = the lower confidence limit
8. `lim2` = the upper confidence limit
9. `test3` = ACCEPT or REJECT
10. `dec.B` = GOLF or STARBUCKS

The data.frame `mystats` should now be 20x14.

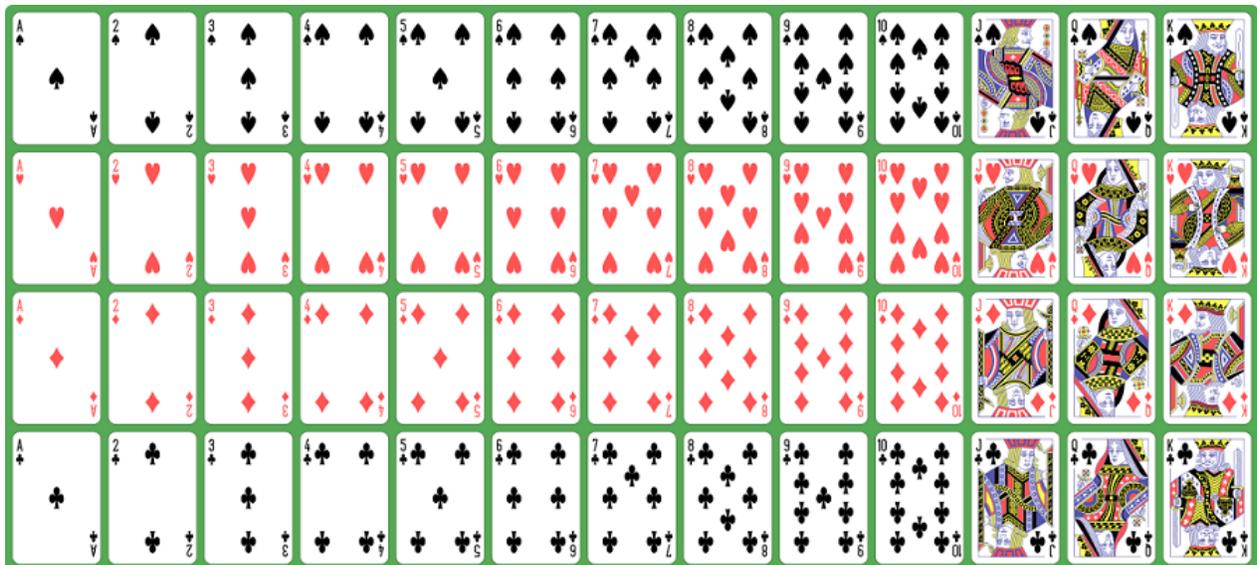
(15 points) Do you think A and B will ever meet? Explain.

A-level instructions:

(10 points) Suppose the true value of μ is 5. That is the same as assuming that H_0 is in fact true. Add columns `grade.A` and `grade.B` to `mystats`. Assuming H_0 is true, grade each investor's decisions as CORRECT or INCORRECT.

(15 points) How many of investor A's decisions were INCORRECT? What type of errors are these? Divide by 20 to get her empirical error rate and compare this number to her theoretical error rate.

(10 points) Now suppose the true value of μ is 5.25. Under this assumption H_0 is false. How would this assumption change the values in columns `grade.A` and `grade.B`?



2 deck52

Build a data.frame to represent a standard deck of 52 playing cards. It should have 52 rows and one column for each characteristic of a playing card: `color` \in {"black", "red"}, `suit`

$\in \{\text{"clubs"}, \text{"spades"}, \text{"diamonds"}, \text{"hearts"}\}$, $\text{name} \in \{\text{"ace"}, \text{"2"}, \text{"3"}, \dots, \text{"king"}\}$,
 $\text{rank} \in \{1, 2, 3, \dots, 12, 13\}$, $\text{value} \in \{1, 2, 3, \dots, 9, 10, 10, 10, 10\}$.

Instructions:

1. Clean. Stylish. (10 points)
2. (25 points) Build a vector with 52 elements for each column: `color`, `suit`, `name`, `rank`, `value`. Useful commands are `c()` and `rep()`.

Use `data.frame()` to combine the columns into a single data.frame called `deck52`. Confirm that your data.frame has 52 rows and 5 columns. Run `names(mydeck) <- toupper(names(deck52))` to convert the column names to upper case.

Consider selecting a card at random from a standard deck. The sample space Ω can be represented by `deck52`. Each row of `deck52` represents a possible outcome and has probability $= 1/52$. An event is any subset of Ω , so any event corresponds to a set of rows in `deck52`. Below we define some events associated with this experiment.

Let B = the event that the selected card is black.

Let R = the event that the selected card is red.

Let D = the event that SUIT of the selected card is diamonds, and let H, S, C define the same event for the other three suits.

Let F = the event that card is a face card = {jack, queen, king}.

Let A = the event that the card is an ace.

Let ODD = the event that the VALUE of the card is odd.

3. (10 points) For each event above, extract and display the corresponding rows of `deck52`. Use the `subset()` function. For the first one there will two lines of code:

```
B <- subset(deck52, COLOR == 'black') # extracting the event B
B                                     # displaying B
```

Hints: Type `help(Logic)` or `help("&")` to see help page for `&` and other logical operators. Type `help(Comparison)` or `help("==")` for `<`, `>` and other comparison operators. Type `help(Arithmetic)` or `help("+")` for arithmetic operators like `+` and `*`. We need `%%` for this assignment. Note the difference between `%%` and `%/%`.

4. (25 points) Calculate the probabilities below using R. The first one is an example.

(a) $P(H) =$

It helps to work it out with a pencil first: $P(H) = \frac{13 \text{ hearts}}{52 \text{ cards}} = \frac{1}{4}$.

How do we get 13/52 in R?

To get the 13 use `dim(H)[1]` or `nrow(H)`. We already have event H from above, so this one is easier. On some you have to extract a new subset before counting the rows.

To get the 52 use `dim(deck52)[1]` or `nrow(deck52)`.

Then divide 13/52 to get the answer .25.

(b) $P(F) =$

(c) $P(ODD) =$

(d) $P(EVEN) =$

This is one minus previous answer, but don't solve it that way. Extract the rows where value is even, count them, and divide by 52.

(e) $P(ODD^C) =$

Same as previous problem, but solve it explicitly: extract the rows that are NOT odd, count, divide.

(f) $P(D) =$

(g) $P(D \cap R) =$

Intersection means AND which is & in R.

(h) $P(D|R) =$

(i) $P(D \cup R) =$

Union means OR which is | in R. Note that this is unrelated to the | used in conditional probability.

(j) $P(F|EVEN) =$

5. (10 points) Package your results for part 4 in a 10x3 data.frame called `myprobs`.

Run `data.frame(question = '4(a)', probability = 'P(H)', answer = .25)` to see the first row as an example. Attach new rows with `rbind()`. When the data.frame is complete, run `myprobs` to display it. Don't use `View()`.

A-level instructions: Repeat parts 3 and 4 using `sqldf()` for subsetting and counting. Package and display these new results in a new data.frame called `myprobsSQL`. It's good to be lazy while building this data.frame. (10 + 20 = 30 points)