homework_A01 - Answers

Dan Swart, CPA (ret) Claude Sonnet 4.5

January 9, 2026

Table of contents

A1.

To study honesty, behavioral scientists have used an experiment called the Random Allocation Game (RAG). In a RAG, participants are given a sin- gle coin. Participants flip the coin, and if the result is heads, they win a small cash prize (like 10 Euros). Participants flip the coin in private—the experi- menter cannot see or verify the result, and participants know this. While it is impossible to know if any individual participant honestly obeys the result of the coin flip, in the aggregate the proportion of prize claims pro- vides information about the frequency of honesty in the sample. For exam- ple, if everyone claims the prize, then probably a lot of them are liars. Suppose 10 participants play a RAG and 8 of them claim the prize. Using the "garden of forking data" approach, how many ways are there to realize this sample (8 out of 10), if all participants are honest? How many ways, if 5 of the participants are honest? Can you figure out the number of honest par- ticipants that maximizes the number of ways to realize the observed sample (8 out of 10)?

Garden of Forking Data Analysis

Conceptual Framework

- Honest participants claim only if heads (p=0.5, two paths),
- Dishonest participants always claim (p=1.0, one path).

The "ways" count is about enumerating paths through possibility space.

Step 1: Define the Parameter Space

Create a tibble with all possible values of honest participants (hon = 0 to 10).

```
df <- tibble::tibble(hon = 0:10)

# check result
print(df)
```

```
# A tibble: 11 × 1
     hon
   <int>
 1     0
 2     1
 3     2
 4     3
 5     4
 6     5
 7     6
```

```
 8      7
 9      8
10      9
11     10
```

Step 2: Calculate Derived Quantities

For each value of hon, create new columns for:

- Number of dishonest participants: 10 - hon
- Guaranteed claims from dishonest: 10 - hon (they always claim)
- Required claims from honest participants to produce sample: 8 - (10 - hon) = hon - 2

```
df2 <- dplyr::mutate(df,
  dis = 10 - hon,
  dis_claims = 10 - hon,
  hon_claims = hon - 2
)

# check result
print(df2)
```

```
# A tibble: 11 × 4
    hon   dis dis_claims hon_claims
  <int> <dbl>      <dbl>      <dbl>
 1      0    10         10         -2
 2      1     9          9         -1
 3      2     8          8          0
 4      3     7          7          1
 5      4     6          6          2
 6      5     5          5          3
 7      6     4          4          4
 8      7     3          3          5
```

```
 9      8      2           2           6
10      9      1           1           7
11     10      0           0           8
```

Step 3: Identify Valid Scenarios

Get the scenarios where the required honest claims is feasible (between 0 and hon inclusive). If hon < 2, you can't get down to only 8 claims.

```
valid_scenarios <- dplyr::filter(df2,
              hon_claims >= 0)

# check result
print(valid_scenarios)
```

```
# A tibble: 9 × 4
    hon   dis dis_claims hon_claims
  <int> <dbl>      <dbl>      <dbl>
1     2     8          8          0
2     3     7          7          1
3     4     6          6          2
4     5     5          5          3
5     6     4          4          4
6     7     3          3          5
7     8     2          2          6
8     9     1          1          7
9    10     0          0          8
```

Step 4: Count the Ways

For valid scenarios, calculate combinations using the binomial coefficient: choosing (hon-2) people from hon honest participants to get heads.

```r
poss_ways <- dplyr::mutate(valid_scenarios,
                           ways = base::choose(n = hon, k
= hon - 2)
)

# check result
print(poss_ways)
```

```
# A tibble: 9 × 5
    hon   dis dis_claims hon_claims  ways
  <int> <dbl>      <dbl>      <dbl> <dbl>
1     2     8          8          0     1
2     3     7          7          1     3
3     4     6          6          2     6
4     5     5          5          3    10
5     6     4          4          4    15
6     7     3          3          5    21
7     8     2          2          6    28
8     9     1          1          7    36
9    10     0          0          8    45
```

Step 5: Answer the Three Questions

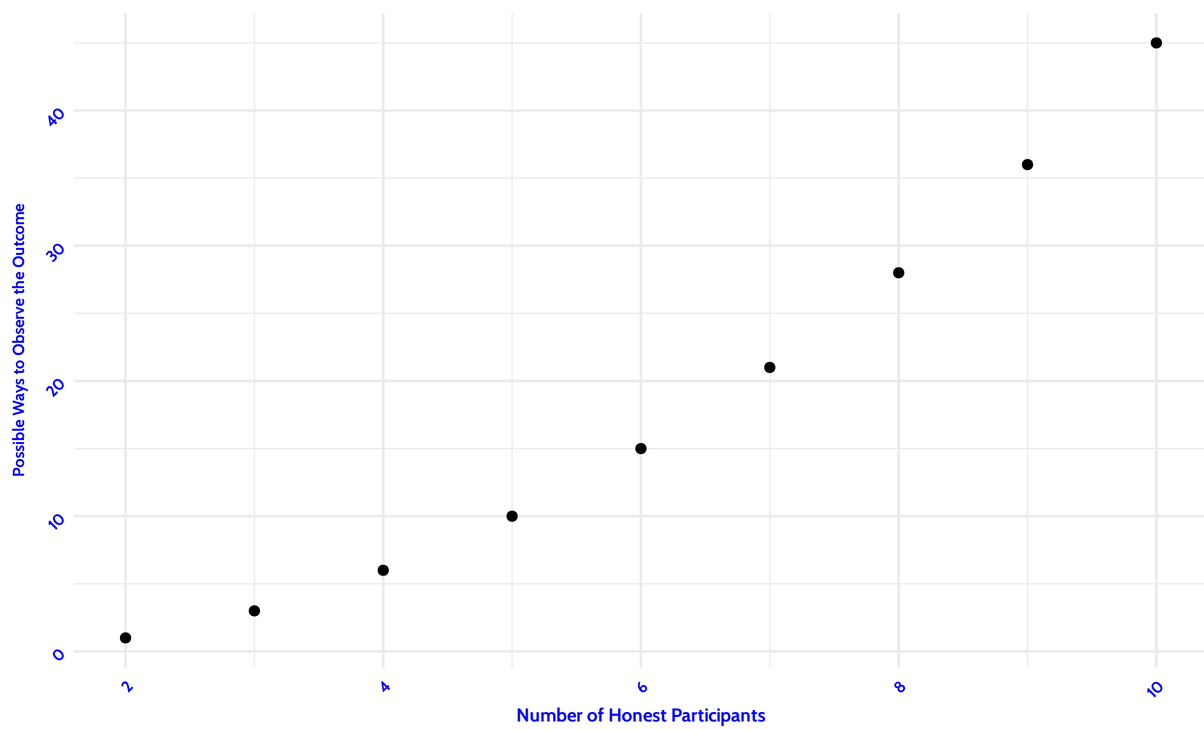
1. All honest (h=10): Pull that specific row
2. Five honest (h=5): Pull that specific row
3. Maximum: Find which value of honest people produces the most ways

```r
all_hon <- dplyr::filter(poss_ways,
                         hon == 10)
five_hon <- dplyr::filter(poss_ways,
                          hon == 5)
max <- dplyr::slice_max(poss_ways,
                        ways)
```

1. Ways for 8 out of 10 claims = 45
2. Ways for 5 out of 10 claims = 10
3. Which # of hons produces the most ways = 10

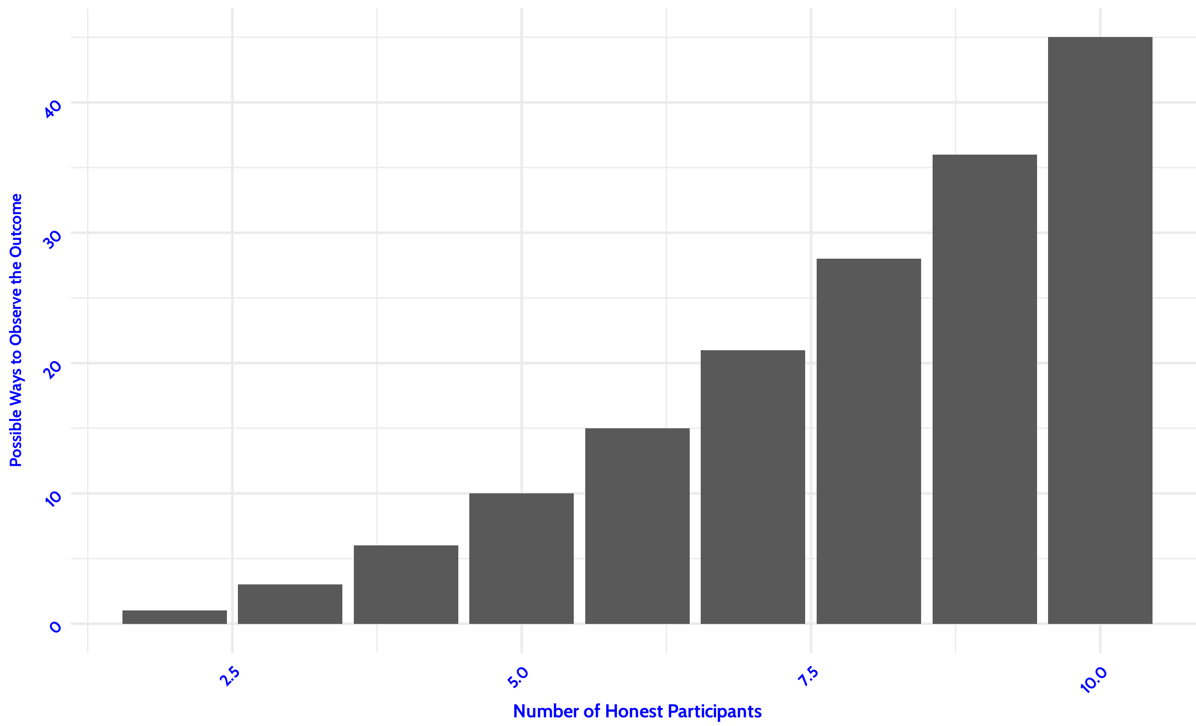
Create a simple visualization showing ways vs. number of honest participants.

```
plot <- ggplot2::ggplot(poss_ways, ggplot2::aes(x = hon, y
= ways)) +
  ggplot2::geom_point() +
  ggplot2::labs(x = "Number of Honest Participants",
                y = "Possible Ways to Observe the Outcome"
                )

plot
```




```
plot2 <- ggplot2::ggplot(poss_ways, ggplot2::aes(x = hon,
y = ways)) +
```

```
  ggplot2::geom_col() +
  ggplot2::labs(x = "Number of Honest Participants",
                y = "Possible Ways to Observe the Outcome"
                )

plot2
```



Algebraic Solution

The binomial coefficient formula:

(nk) = n!/(k!(n−k)!)

For this specific case, base::choose(n=hon, k=hon - 2):

hon!/(hon−2)!⋅2!

Which simplifies to:

hon⋅((hon−1)/2)

Verify with above results:

hon hon × ((hon-1)/2) = ways

2 2 × (1/2) = 1 1 ✓ 5 5 × (4/2) = 10 10 ✓ 10 10 × (9/2) = 45 45 ✓