

# Homework 1: Simulation-Based Data Analysis

BEE 4850/5850, Fall 2024

**Name:** Anthony Nicolaides

**ID:** ajn68

## Due Date

Friday, 2/2/24, 9:00pm

## Overview

## Instructions

The goal of this homework assignment is to introduce you to simulation-based data analysis.

- Problem 1 asks you to explore whether a difference between data collected from two groups might be statistically meaningful or the result of noise. This problem repeats the analysis from [Statistics Without The Agonizing Pain](#) by John Rauser (which is a neat watch!).
- Problem 2 asks you to evaluate an interview method for finding the level of cheating on a test to determine whether cheating was relatively high or low. This problem was adapted from [Bayesian Methods for Hackers](#).
- Problem 3 asks you to assess the quality of fit of a normal distribution to realizations from a Galton Board simulation.
- Problem 4 (**graded only for graduate students**) asks you to simulate outcomes from the *Price Is Right* Showcase game to identify a bidding strategy.

## Load Environment

The following code loads the environment and makes sure all needed packages are installed. This should be at the start of most Julia scripts.

```
In [ ]: import Pkg
        Pkg.activate(@__DIR__)
        Pkg.instantiate()
```

**Activating** project at `~/Documents/BEE5850/hw/hw01`

The following packages are included in the environment (to help you find other similar packages in other languages). The code below loads these packages for use in the subsequent notebook (the desired functionality for each package is commented next to the package).

```
In [ ]: using Random # random number generation and seed-setting
using DataFrames # tabular data structure
using CSVFiles # reads/writes .csv files
using Distributions # interface to work with probability distributions
using Plots # plotting library
using StatsBase # statistical quantities like mean, median, etc
using StatsPlots # some additional statistical plotting tools
```

## Problems

Each problem in this assignment is worth 10 points.

### Problem 1

The underlying question we would like to address is: what is the influence of drinking beer on the likelihood of being bitten by mosquitoes? There is a mechanistic reason why this might occur: mosquitoes are attracted by changes in body temperature and released CO<sub>2</sub>, and it might be that drinking beer induces these changes. We'll analyze this question using (synthetic) data which separates an experimental population into two groups, one which drank beer and the other which drank only water.

First, we'll load data for the number of bites reported by the participants who drank beer. This is in a comma-delimited file, `data/bites.csv` (which is grossly overkill for this assignment). Each row contains two columns: the group ( `beer` and `water` ) the person belonged to and the number of times that person was bitten.

In Julia, we can do this using `CSVFiles.jl`, which will read in the `.csv` file into a `DataFrame`, which is a typical data structure for tabular data (and equivalent to a Pandas `DataFrame` in Python or a `dataframe` in R).

```
In [ ]: data = DataFrame(load("data/bites.csv")) # load data into DataFrame

# print data variable (semi-colon suppresses echoed output in Julia, which i
@show data;
```

```
data = 43x2 DataFrame
```

Row	group String	bites Int64
1	beer	27
2	beer	20
3	beer	21
4	beer	26
5	beer	27
6	beer	31
7	beer	24
8	beer	21
9	beer	20
10	beer	19
11	beer	23
12	beer	24
13	beer	28
14	beer	19
15	beer	24
16	beer	29
17	beer	18
18	beer	20
19	beer	17
20	beer	31
21	beer	20
22	beer	25
23	beer	28
24	beer	21
25	beer	27
26	water	21
27	water	22
28	water	15
29	water	12
30	water	21
31	water	16
32	water	19
33	water	15
34	water	22
35	water	24
36	water	19
37	water	23
38	water	13
39	water	22
40	water	20
41	water	24
42	water	18
43	water	20

How can we tell if there's a meaningful difference between the two groups? Naively, we might just look at the differences in group means.

## Broadcasting

The subsetting operations in the below code use `.==`, which “broadcasts” the element-wise comparison operator `==` across every element. The decimal in front of `==` indicates that this should be used element-wise (every pair of elements compared for equality, returning a vector of `true` or `false` values); otherwise Julia would try to just check for vector equality (returning a single `true` or `false` value).

Broadcasting is a very specific feature of Julia, so this syntax would look different in a different programming language.

```
In [ ]: # split data into vectors of bites for each group
beer = data[data.group.=="beer", :bites]
water = data[data.group.=="water", :bites]

observed_difference = mean(beer) - mean(water)
@show observed_difference;
```

```
observed_difference = 4.3777777777778
```

This tells us that, on average, the participants in the experiment who drank beer were bitten approximately 4.4 more times than the participants who drank water! Does that seem like a meaningful difference, or could it be the result of random chance?

We will use a *simulation* approach to address this question, as follows.

- Suppose someone is skeptical of the idea that drinking beer could result in a higher attraction to mosquitoes, and therefore more bites. To this skeptic, the two datasets are really just different samples from the same underlying population of people getting bitten by mosquitoes, rather than two different populations with different propensities for being bitten. This is the skeptic’s *hypothesis*, versus our hypothesis that drinking beer changes body temperature and CO<sub>2</sub> release sufficiently to attract mosquitoes.
- If the skeptic’s hypothesis is true, then we can “shuffle” all of the measurements between the two datasets and re-compute the differences in the means. After repeating this procedure a large number of times, we would obtain a distribution of the differences in means under the assumption that the skeptic’s hypothesis is true.
- Comparing our experimentally-observed difference to this distribution, we can then evaluate the consistency of the skeptic’s hypothesis with the experimental results.

### Why Do We Call This A Simulation-Based Approach?

This is a simulation-based approach because the “shuffling” is a non-parametric way of generating new samples from the underlying distribution (more on this later in the semester).

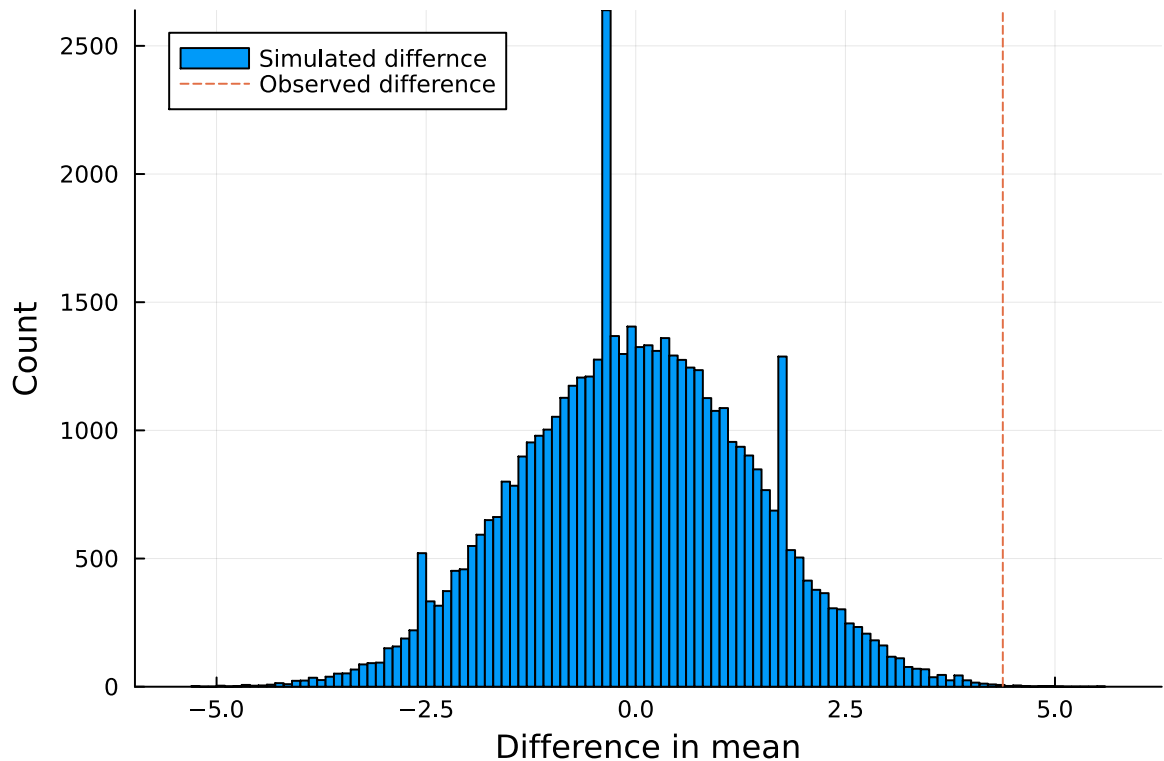
The alternative to this approach is to use a statistical test, such as a [t-test](#), which [may have other assumptions](#) which may not be appropriate for this setting, particularly given the seemingly small sample sizes.

**In this problem:**

- Conduct the above procedure to generate 50,000 simulated datasets under the skeptic's hypothesis.
- Plot a histogram of the results and add a dashed vertical line to show the experimental difference (if you are using Julia, feel free to look at the [Making Plots with Julia tutorial](#) on the class website).
- Draw conclusions about the plausibility of the skeptic's hypothesis that there is no difference between groups. Feel free to use any quantitative or qualitative assessments of your simulations and the observed difference.

```
In [ ]: all_bites = copy(data.bites)
        group_water_end = sum(data.group .== "water")
        group_beer_start = group_water_end + 1
        differences = []
        n = 50000 # simulations
        for i in 1:n
            shuffle!(all_bites) # shuffle values and assign them to the groups
            group_water = all_bites[1:group_water_end]
            group_beer = all_bites[group_beer_start:end]
            push!(differences, mean(group_beer) - mean(group_water))
        end
```

```
In [ ]: histogram(differences, label="Simulated difference")
        vline!([observed_difference],
                label="Observed difference",
                linestyle=:dash)
        xlabel!("Difference in mean")
        ylabel!("Count")
```



I believe you can conclude that the skeptic's hypothesis is not plausible. If the skeptic's hypothesis was plausible, then the observed difference would have been closer to the mean of the simulated differences, which is about 0. However, the observed difference is actually on the very end of the tail of the simulated differences distribution. Knowing this, it is unlikely the observed difference is due to change, and therefore the original conclusion of the observed difference can be kept.

## Problem 2

You are trying to detect if how prevalent cheating was on an exam. You are skeptical of the efficacy of just asking the students if they cheated. You are also concerned about privacy — your goal is not to punish individual students, but to see if there are systemic problems that need to be addressed. Someone proposes the following interview procedure, which the class agrees to participate in:

Each student flips a fair coin, with the results hidden from the interviewer. The student answers honestly if the coin comes up heads. Otherwise, if the coin comes up tails, the student flips the coin again, and answers "I did cheat" if heads, and "I did not cheat", if tails.

We have a hypothesis that cheating was not prevalent, and the proportion of cheaters was no more than 5% of the class; in other words, we expect 5 "true" cheaters out of a class of 100 students. Our TA is more jaded and thinks that cheating was more rampant, and that 30% of the class cheated. The proposed interview procedure is noisy: the interviewer does not know if an admission means that the student cheated, or the result

of a heads. However, it gives us a data-generating process that we can model and analyze for consistency with our hypothesis and that of the TA.

**In this problem:**

- Derive and code a simulation model for the above interview procedure given the “true” probability of cheating  $p$ .
- Simulate your model (for a class of 100 students) 50,000 times under the your hypothesis and the TA's hypothesis, and plot the two resulting datasets.
- If you received 31 “Yes, I cheated” responses while interviewing your class, what could you conclude? Feel free to use any qualitative or quantitative assessments to justify your conclusions.
- How useful do you think the interview procedure is to identify systemic teaching? What changes to the design might you make?

```
In [ ]: function interview(class_size, hypothesis)
    class_cheaters = 0
    for i in 1:class_size
        # Flip a fair coin
        coin_flip1 = rand(0:1)
        if coin_flip1 == 0 # Heads
            #tell the truth
            honesty = rand(0:0.01:1)
            if honesty <= hypothesis
                class_cheaters = class_cheaters + 1
            end
        else # tails
            coin_flip2 = rand(0:1)
            if coin_flip2 == 0 # Heads
                class_cheaters = class_cheaters + 1 # says yes to cheating
            else # tails
                #not cheater
            end
        end
    end
    return class_cheaters
end
```

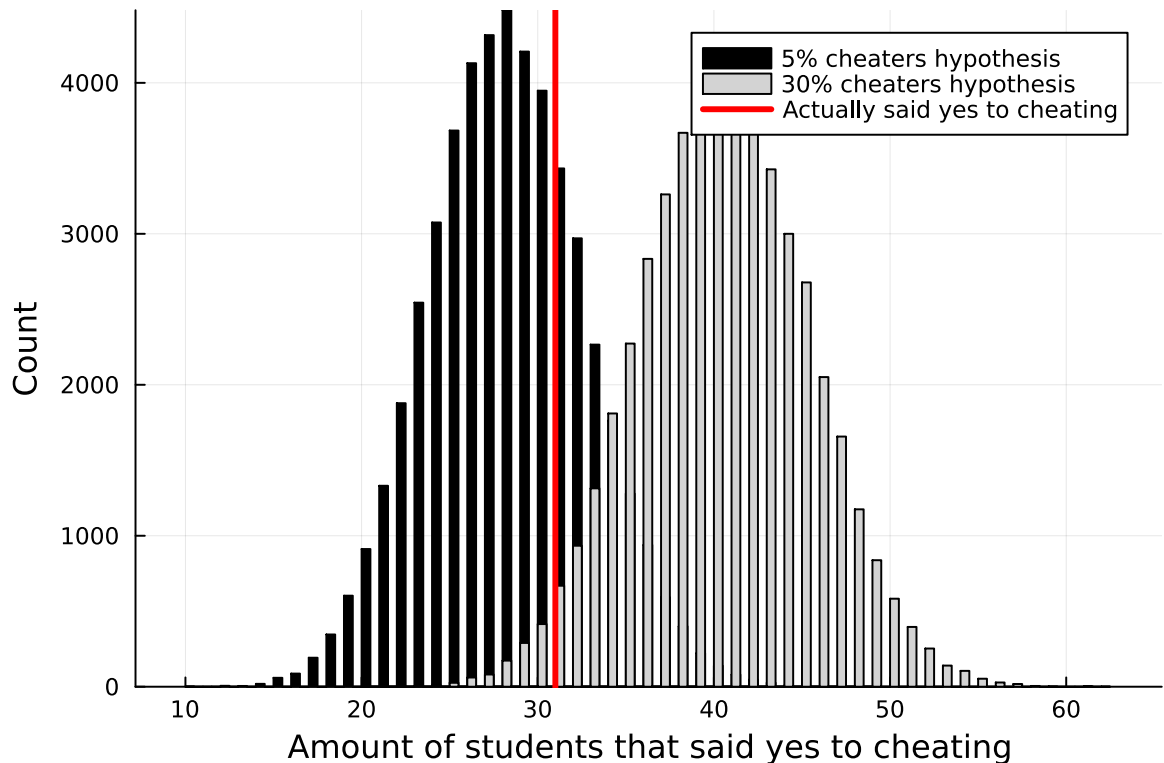
interview (generic function with 1 method)

```
In [ ]: class_size = 100
n = 50000
p_hypothesis = 0.05
TA_hypothesis = 0.3

simulation_p = []
simulation_TA = []
Random.seed!(10)
for i in 1:n
    simulation_p = push!(simulation_p, interview(class_size, p_hypothesis))
    simulation_TA = push!(simulation_TA, interview(class_size, TA_hypothesis))
end
```

```
In [ ]: histogram(simulation_p, color="black",
                  label="5% cheaters hypothesis")
histogram!(simulation_TA, color="light gray",
            label="30% cheaters hypothesis")
vline!([31], color="red", linewidth=3,
        label="Actually said yes to cheating")

xlabel!("Amount of students that said yes to cheating")
ylabel!("Count")
```



I believe the hypothesis of there being 5 cheaters out of 100 students more plausible. Since we know 31 students said yes to cheating, which every hypothesis that has 31 students admitting to cheating being more likely is the more plausible one. The 5% cheaters hypothesis (seen in black) has 31 students say yes about 3200 out of the 50000 simulations that were ran, whereas the 30% cheaters hypothesis (seen in gray) has 31 students say yes about 600 out of the 50000 simulations ran (eyeball-ing). This means the 5% cheaters hypothesis is more plausible.

I think the interview procedure is a simple format that can yield some useful results of the amount of cheaters because it is anonymous and has a way to minimize the amount of liars. I do think the interview should be conducted multiple times to the class to reduce variability in the amount of "Yes, I cheated" responses.

### Problem 3 (10 points)



Suppose that we ask a class of students to walk individually from one end of a field to the other and count the number of steps that they take in order to cover the distance. More than likely, each student will get a somewhat different answer. After each student has paced off the field, we then make a histogram of the number of steps taken by the individual students. For a sufficiently large number of students, this histogram would resemble a normal distribution, due to the [Central Limit Theorem](#).

A **Galton board** (named after [Sir Francis Galton](#), a pioneering statistician[1], shown below, is a physical device that demonstrates the Central Limit Theorem. A Galton board has rows of pins arranged in a triangular shape, with each row of pins offset horizontally relative to the rows above and below it. As illustrated in [Figure 1](#), the top row has one pin, the second row has two pins, and so forth. If you've watched *The Price is Right*, you've seen this as "Plinko".

[1] And, notably, a leading eugenicist, which is unfortunately a recurring theme with leading early statisticians...)

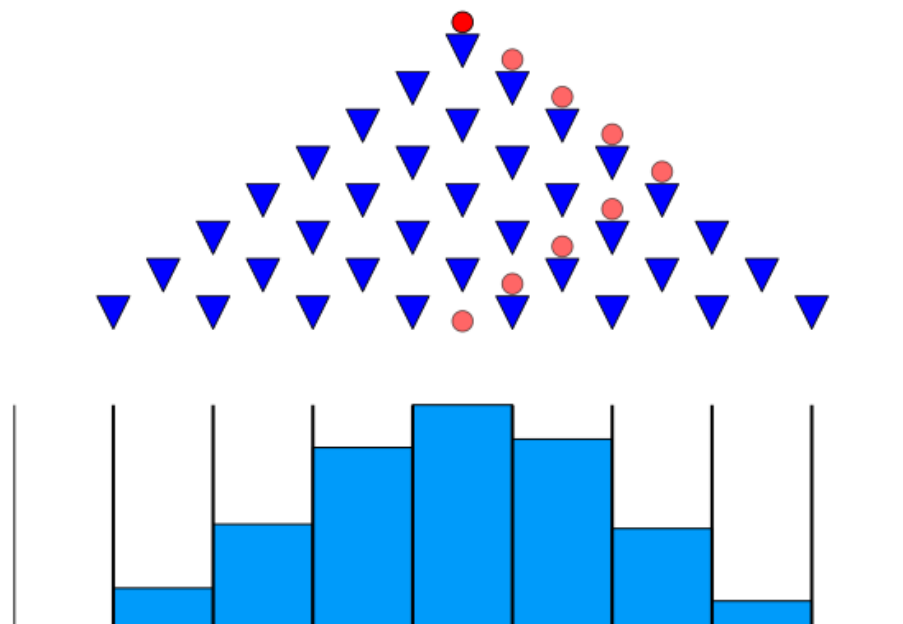


Figure 1: Galton Board simulation. The top shows a schematic of how the Galton board works and the bottom shows a histogram of the landing positions after 200 simulations.

If a ball is dropped into the Galton board, it falls either to the right or the left when it bounces off the top pin. The ball then falls all the way to the bottom of the board, moving slightly to the right or left as it passes through each row of pins. Bins at the bottom of the board capture the ball and record its final position. If this experiment is repeated with many balls, the number of balls in each bin resembles a normal distribution, which is expected due to the Central Limit Theorem.

Your goal in this exercise is to explore how well a normal distribution fits the outcomes of repeated Galton Board trials as the number increases using quantile-quantile plots. Assessing the appropriateness of a probability model for a data set is a key part of exploratory analysis; we will return to this theme repeatedly. Or, to put it another way, it's time for you to play Plinko!

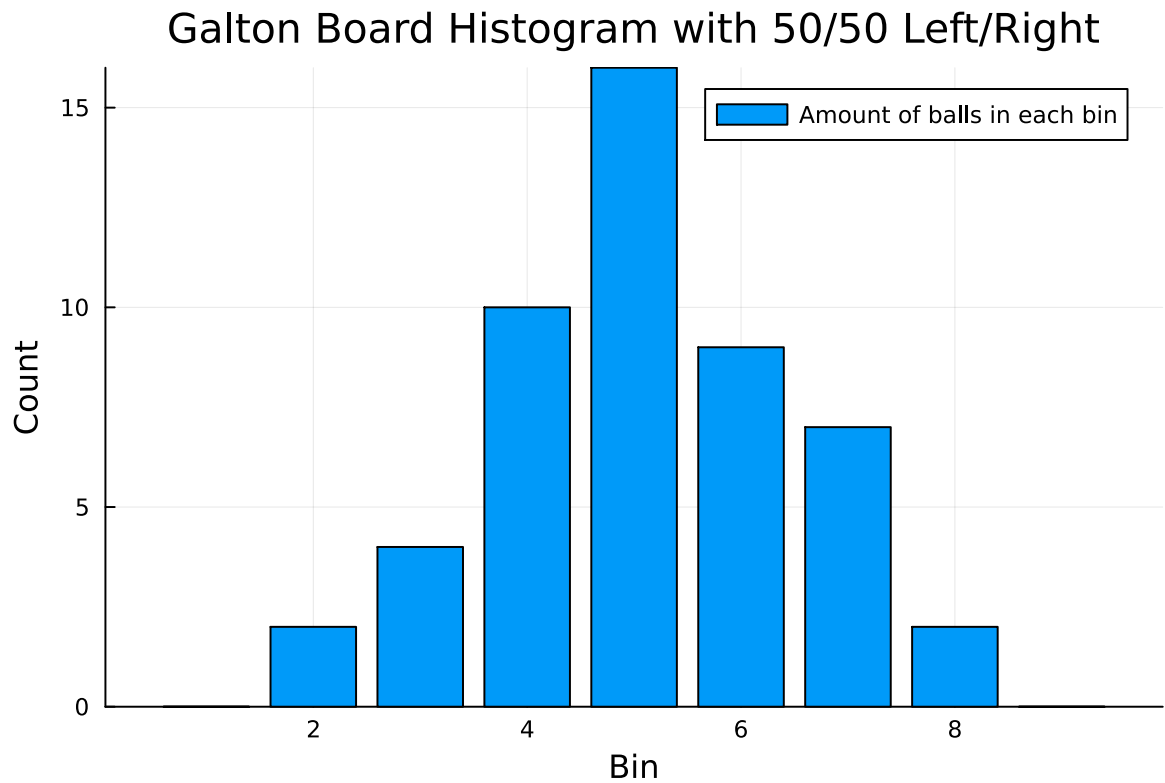
### In this problem:

- Write a function `galton_sim` which simulates `n` Galton board trials (assume the board has 8 board rows, as in the image above) and returns a vector with the number of balls which fall into each bin. You can assume (for now) that the board is fair, e.g. that the probability of a left or right bounce is 0.5; you may want to make this probability a function parameter so you can change it later.
- Run your simulation for a sample of 50 balls. Create a histogram of the results, with each bar corresponding to one bin. Make sure you use a random seed for reproducibility, and label your axes!
- Each Galton board trial can be represented as a realization from a [binomial distribution](#). But as we noted above, by the Central Limit Theorem, the distribution of a large enough number of trials should be approximately normal. Use a [quantile-quantile \(Q-Q\) plot](#) to compare a fitted normal distribution with your simulation results. How well does a normal distribution fit the data?
- Repeat your simulation experiment with 250 trials and compare to a normal distribution. Does it describe the empirical distribution better?
- If the probability of a left bounce is 70%, what does this do to the fit of a normal distribution? What other distribution might you use if not a normal and why?

```
In [ ]: function galton_sim(n, p)
    rows = 8
    bin_array = zeros(rows + 1)
    for ball in 1:n
        bounce_order = []
        for i in 1:rows
            bounce_direction = rand(0:0.01:1)
            if bounce_direction <= p # left
                push!(bounce_order, -1)
            else # right
                push!(bounce_order, 1)
            end
        end
        #
        #           bins:  1  2  3  4  5  6  7  8  9
        # sum(bounce_order) gives: -8 -6 -4 -2 0 +2 +4 +6 +8
        # so convert sum(bounce_order)/2 + 5 = bin
        # used div to return an integer
        bin_array[div(sum(bounce_order), 2)+5] += 1
    end
    return bin_array
end
```

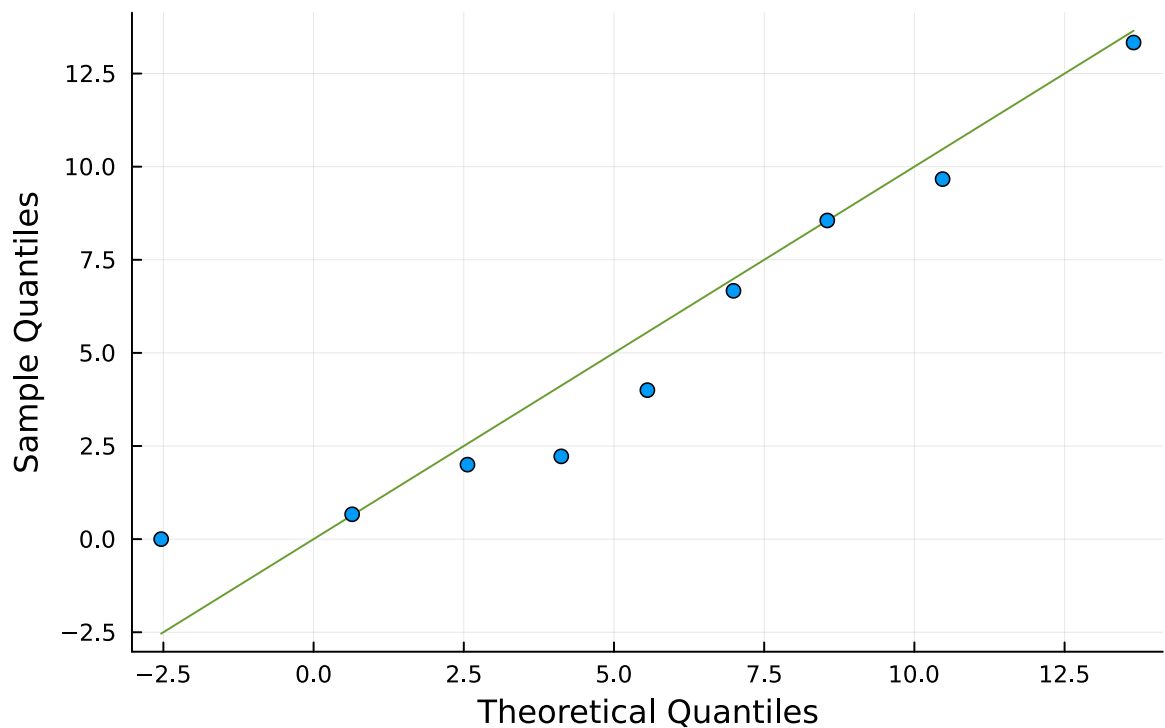
galton\_sim (generic function with 1 method)

```
In [ ]: Random.seed!(10)
results = galton_sim(50, 0.5)
bar(results, label="Amount of balls in each bin",
      xlabel="Bin", ylabel="Count")
title!("Galton Board Histogram with 50/50 Left/Right")
```



```
In [ ]: fitted_results = fit(Normal, results)
qqplot(fitted_results, results, label="QQ Plot")
xlabel!("Theoretical Quantiles")
ylabel!("Sample Quantiles")
title!("QQ-Plot with 1 Trial")
```

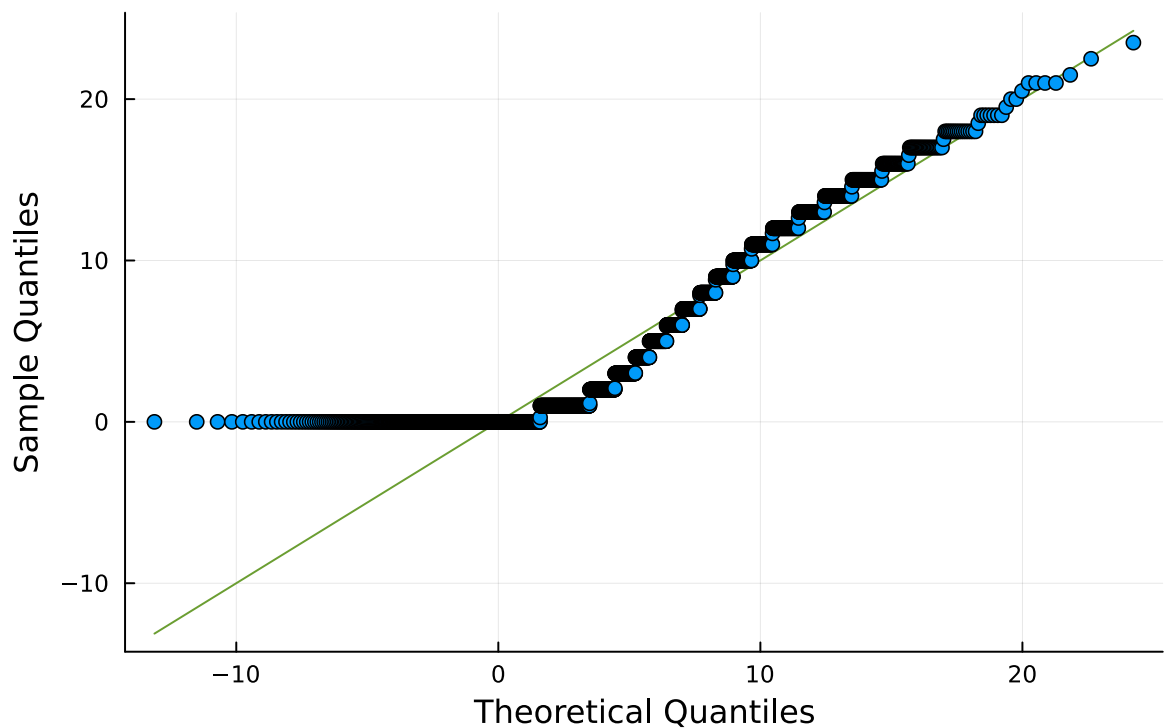
## QQ-Plot with 1 Trial



The QQ-plot seem to vaguely follow a straight line, made with a small bit of a left tail. This would suggest that the simulation results does not strictly follow a normal distribution. However, with such a low amount of trials, it's difficult to made any definitive conclusions.

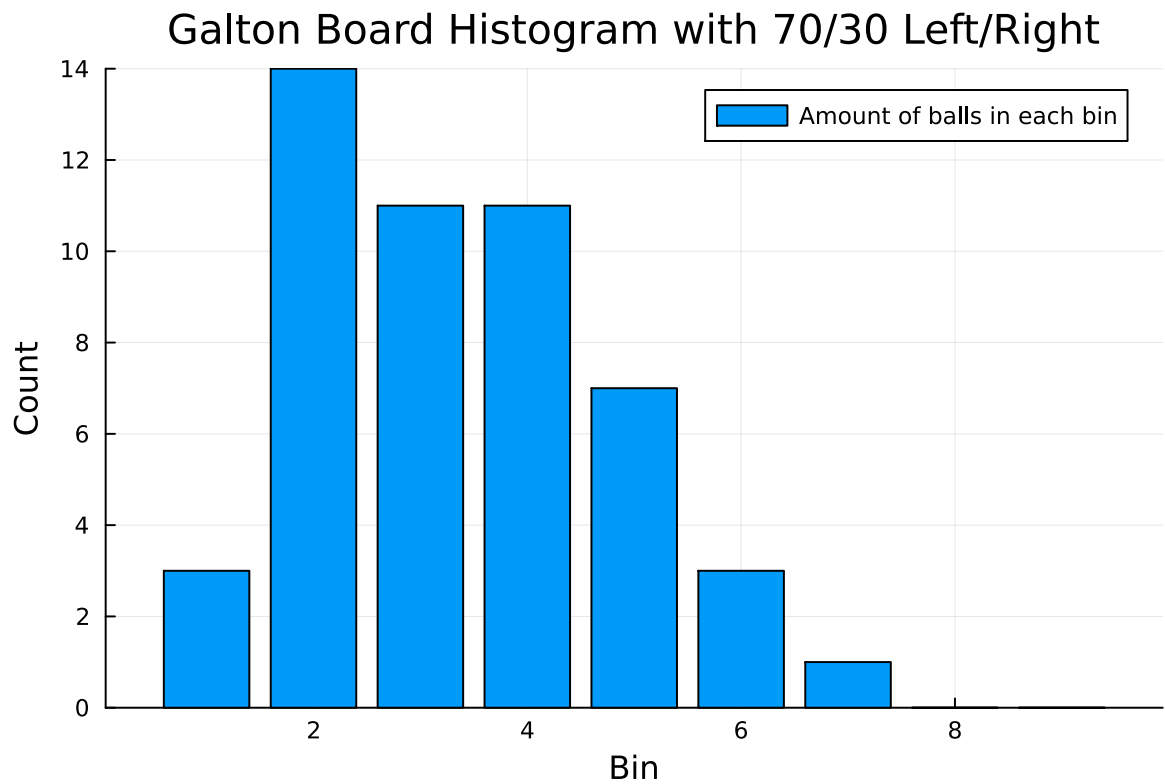
```
In [ ]: results = Vector{Float64}{}
for i in 1:250
    Random.seed!(i)
    append!(results, galton_sim(50, 0.5))
end
fitted_results = fit(Normal, results)
qqplot(fitted_results, results, label="QQ Plot")
xlabel!("Theoretical Quantiles")
ylabel!("Sample Quantiles")
title!("QQ-Plot with 250 Trials")
```

## QQ-Plot with 250 Trials



The increased number of trials shows that the simulation results do follow a normal distribution. However, because you can not have a negative amount of balls in a bin, the simulation results get stopped at 0 balls. This results in a large displacement of values, as seen in the left tail of the plot.

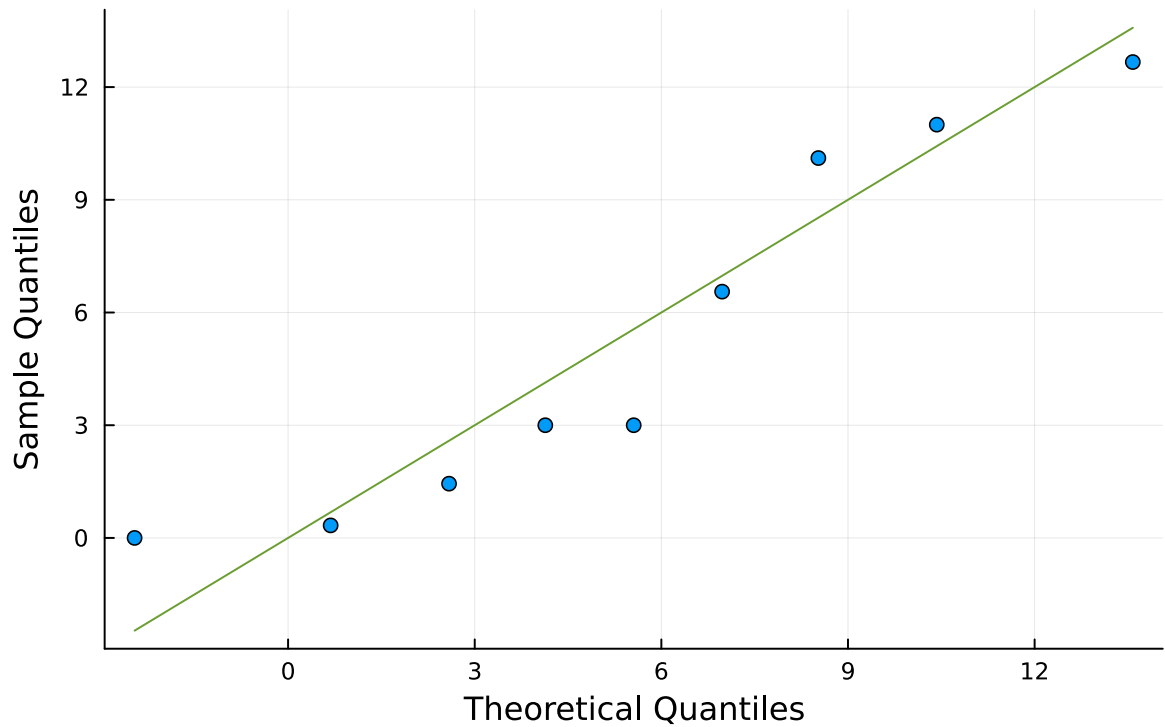
```
In [ ]: Random.seed!(10)
         results = galton_sim(50, 0.7)
         bar(results, label="Amount of balls in each bin",
             xlabel="Bin", ylabel="Count")
         title!("Galton Board Histogram with 70/30 Left/Right")
```



The simulation results now show a right skew due to more balls being in the left bins.

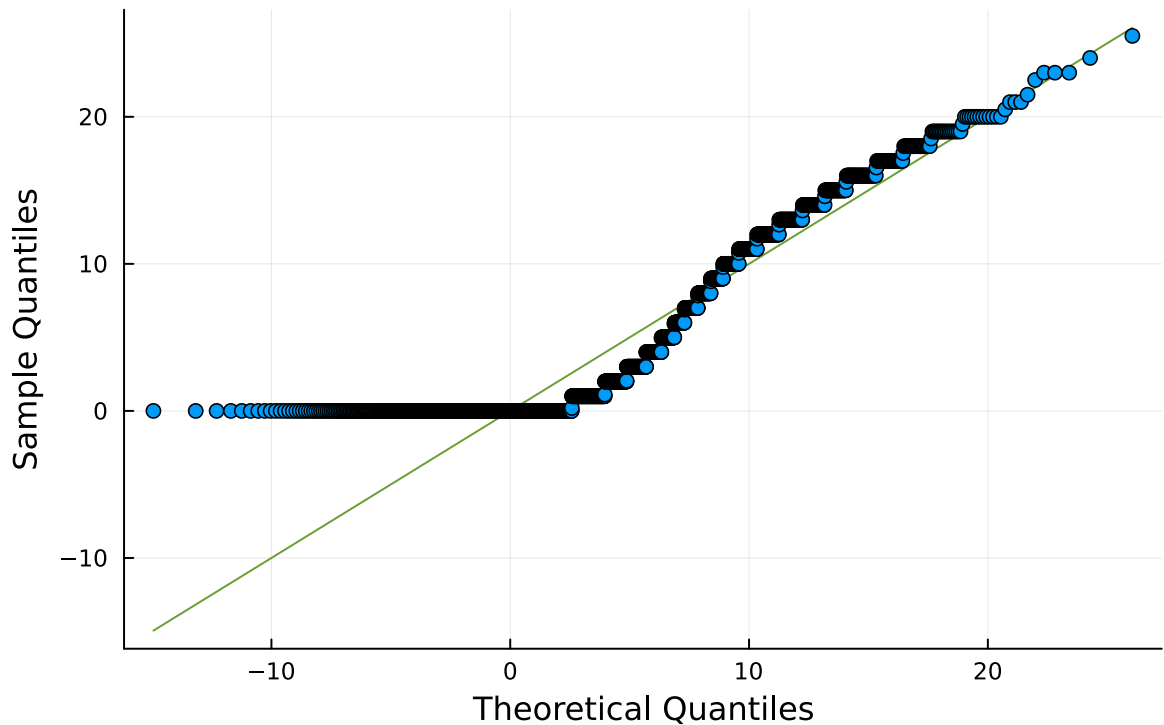
```
In [ ]: fitted_results = fit(Normal, results)
qqplot(fitted_results, results, label="QQ Plot")
xlabel!("Theoretical Quantiles")
ylabel!("Sample Quantiles")
title!("QQ-Plot of 1 Trial")
```

QQ-Plot of 1 Trial



```
In [ ]: results = Vector{Float64}{}
for i in 1:250
    Random.seed!(i)
    append!(results, galton_sim(50, 0.7))
end
fitted_results = fit(Normal, results)
qqplot(fitted_results, results, label="QQ Plot")
xlabel!("Theoretical Quantiles")
ylabel!("Sample Quantiles")
title!("QQ-Plot with 250 Trials")
```

## QQ-Plot with 250 Trials



The qq-plot does show a subtle right skew as the plot makes a dip below the middle of the diagonal line. This is appropriate results as the probability of the balls landing in the left bins was increased. So the distribution would now have a right skew to it.

### Problem 4

**GRADED FOR 5850 STUDENTS ONLY**

Your mastery of the Central Limit Theorem has led you to win your game of Plinko, and it's time for the [Showcases](#). This is the final round of an episode of *The Price is Right*, matching the two big winners from the episode. Each contestant is shown a "showcase" of prizes, which are usually some combination of a trip, a motor vehicle, some furniture, and maybe some other stuff. They then each have to make a bid on the retail price of the showcase. The rules are:

- an overbid is an automatic loss;
- the contest who gets closest to the retail price wins their showcase;
- if a contestant gets within \$250 of the retail price and is closer than their opponent, they win both showcases.

Thanks to [exhaustive statistics kept by game show fans](#), in Season 51 of *The Price is Right* showcase values had the following properties:

- median: \$33,0481;
- lowest-value: \$20,432



- highest-value: \$72,409

### In this problem:

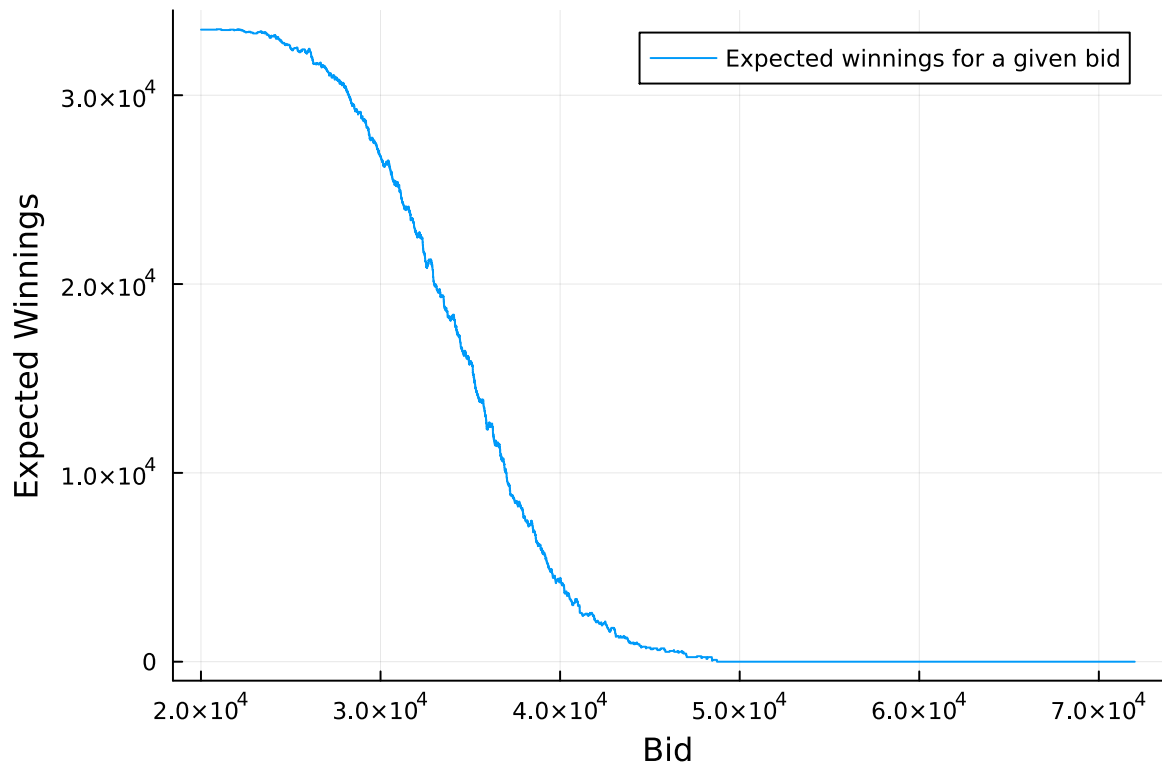
- Write down a model which encodes the Showcase rules as a function of the showcase value and your bid. You can assume that your wagering is independent of your opponent.
- Select and fit a distribution to the above statistics (you have some freedom to pick a distribution, but make sure you justify it).
- Using 1,000 samples from your price distribution in your model, plot the expected winnings for bids from 20,000 *through* 72,000.
- Find the bid which maximizes your expected winnings. If you were playing *The Price Is Right*, is this the strategy you would adopt, or are there other considerations you would take into account which were not included in this model?

```
In [ ]: function showcase(bid)
    Random.seed!(10)
    showcase_dist = truncated(Normal(33481, 5000), lower=20432, upper=72409)
    # Using a normal distribution as it is assumed there is a large number
    # of prizes over the course of the whole season
    # StDev of 5000 seemed appropriate.
    expected_winnings = 0
    for _ in 1:1000
        value = rand(showcase_dist)
        if bid <= value # do not overbid
            if (value - bid) <= 250
                expected_winnings = (value * 2) + expected_winnings
            else
                expected_winnings = (value) + expected_winnings
            end
        end
    end
    return expected_winnings / 1000
end
```

showcase (generic function with 1 method)

```
In [ ]: expected_winnings = zeros(length(20000:72000), 2)
    for (i, bid) in enumerate(20000:72000)
        expected_winnings[i, 1] = bid
        expected_winnings[i, 2] = showcase(bid)
    end
```

```
In [ ]: plot(expected_winnings[:, 1], expected_winnings[:, 2],
    label="Expected winnings for a given bid")
xlabel!("Bid")
ylabel!("Expected Winnings")
```



```
In [ ]: expected_winnings[argmax(expected_winnings[:, 2]), 1]
```

22069.0

The bid that gives the largest expected winnings is 22069 USD. One of the limitations of this model is that it does not consider the opponent's bid, which should play a factor in your deciding bid. Therefore, 22069 USD would be my strategic bid as I would need to consider what the opponent's bid is (e.g. if the opponent bids 22070, then I need to reconsider bidding 22069).