Untitled

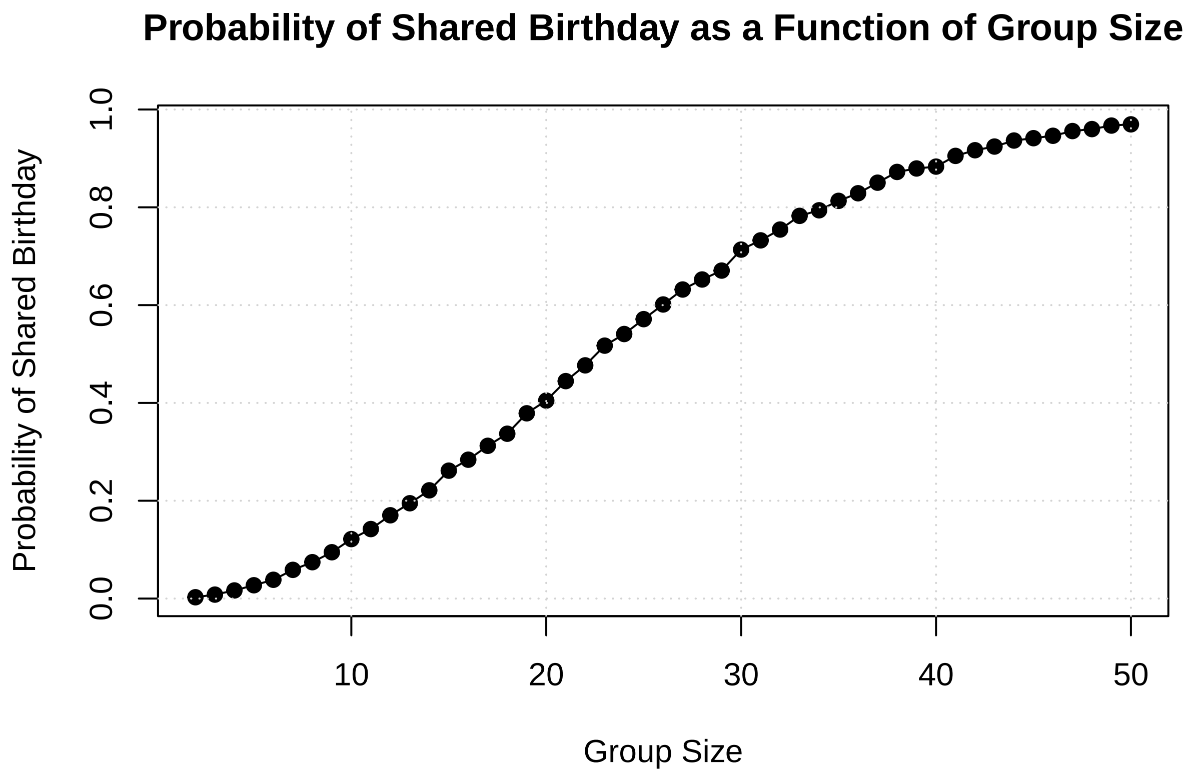
Tai Yue

2024-11-11

problem1

```r
has_shared_birthday <- function(n) {

  birthdays <- sample(1:365, n, replace = TRUE)


  return(length(birthdays) != length(unique(birthdays)))
}
```

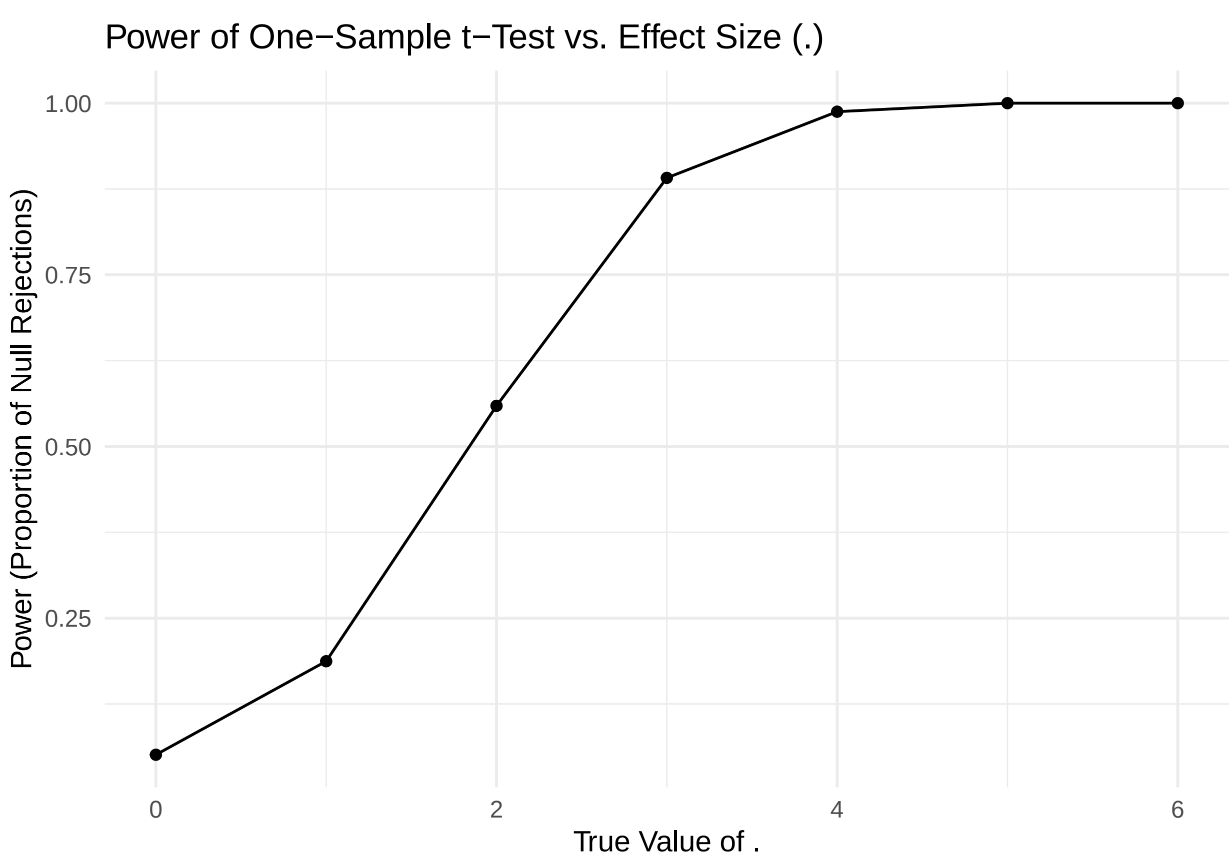
```r
has_shared_birthday <- function(n) {
  birthdays <- sample(1:365, n, replace = TRUE)
  return(length(birthdays) != length(unique(birthdays)))
}


group_sizes <- 2:50
n_simulations <- 10000
probabilities <- numeric(length(group_sizes))


for (i in seq_along(group_sizes)) {
  n <- group_sizes[i]
  duplicates <- sum(replicate(n_simulations, has_shared_birthday(n)))
  probabilities[i] <- duplicates / n_simulations
}


plot(group_sizes, probabilities, type = "o", pch = 19,
     xlab = "Group Size", ylab = "Probability of Shared Birthday",
     main = "Probability of Shared Birthday as a Function of Group Size")
grid()
```



problem2:

```
library(broom)
library(ggplot2)


n <- 30
sigma <- 5
alpha <- 0.05
mu_values <- c(0, 1, 2, 3, 4, 5, 6)
num_simulations <- 5000


results <- data.frame()

for (mu in mu_values) {

  rejections <- 0

  for (i in 1:num_simulations) {

    sample <- rnorm(n, mean = mu, sd = sigma)


    t_test_result <- t.test(sample, mu = 0)
    tidy_result <- tidy(t_test_result)
```

```
    if (tidy_result$p.value < alpha) {
      rejections <- rejections + 1
    }
  }


  power <- rejections / num_simulations


  results <- rbind(results, data.frame(mu = mu, power = power))
}


ggplot(results, aes(x = mu, y = power)) +
  geom_line() +
  geom_point() +
  labs(
    x = "True Value of  ",
    y = "Power (Proportion of Null Rejections)",
    title = "Power of One-Sample t-Test vs. Effect Size ( )"
  ) +
  theme_minimal()
```



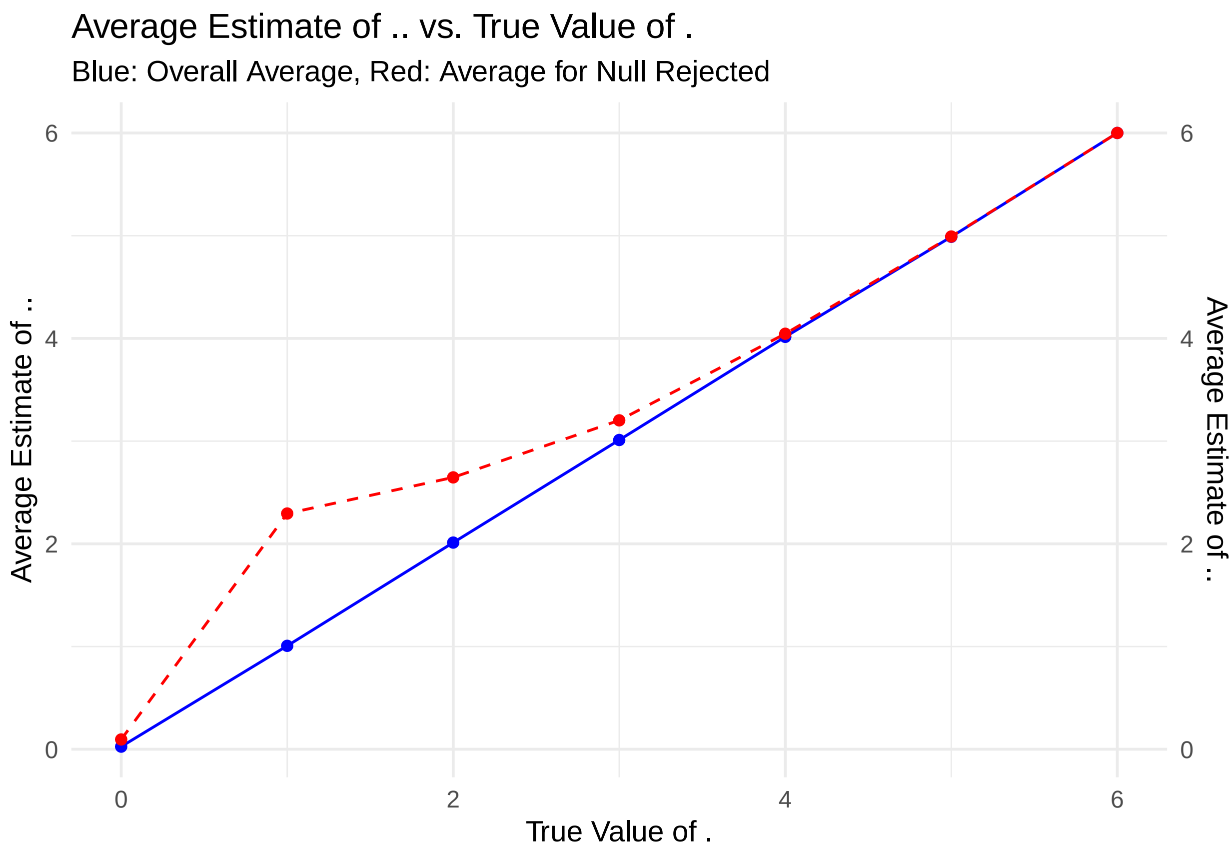
For small values of , the power is low, meaning the test often fails to reject the null hypothesis. As increases, the power rises significantly, reaching close to 1 for values around 4 and higher. This means that, with larger effect sizes, the test almost always correctly rejects the null hypothesis.The trend suggests a positive relationship between effect size and power: larger effect sizes make it easier to detect an effect, thus increasing the test's power.

```r
library(broom)
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
n <- 30
sigma <- 5
alpha <- 0.05
mu_values <- c(0, 1, 2, 3, 4, 5, 6)
num_simulations <- 5000


results <- data.frame()

for (mu in mu_values) {

  estimates <- numeric(num_simulations)
  rejections <- numeric(num_simulations)

  for (i in 1:num_simulations) {

    sample <- rnorm(n, mean = mu, sd = sigma)


    t_test_result <- t.test(sample, mu = 0)
    tidy_result <- tidy(t_test_result)


    estimates[i] <- tidy_result$estimate
    rejections[i] <- ifelse(tidy_result$p.value < alpha, 1, 0)
  }


  avg_mu_hat <- mean(estimates)


  avg_mu_hat_rejected <- mean(estimates[rejections == 1])
```

```
  results <- rbind(results, data.frame(mu = mu, avg_mu_hat = avg_mu_hat, avg_mu_hat_rejected = avg_mu_ha
}


ggplot(results, aes(x = mu)) +
  geom_line(aes(y = avg_mu_hat), color = "blue", linetype = "solid") +
  geom_point(aes(y = avg_mu_hat), color = "blue") +
  geom_line(aes(y = avg_mu_hat_rejected), color = "red", linetype = "dashed") +
  geom_point(aes(y = avg_mu_hat_rejected), color = "red") +
  labs(
    x = "True Value of  ",
    y = "Average Estimate of ^",
    title = "Average Estimate of ^ vs. True Value of  ",
    subtitle = "Blue: Overall Average, Red: Average for Null Rejected"
  ) +
  theme_minimal() +
  scale_y_continuous(sec.axis = dup_axis(name = "Average Estimate of ^"))
```



the red dashed line is very close to the true value of .The average estimate of ^ across tests where the null is rejected approximates the true value of well for larger effect sizes but overestimates it for smaller values of due to selection bias among the rejected samples.

problem3:

The raw dataset on homicides in large U.S. cities includes the following fields:

uid: A unique identifier for each homicide case. reported_date: The date the homicide was reported, formatted as an eight-digit integer victim_last: The last name of the victim. victim_first: The first name of the victim. victim_race: The race of the victim victim_age: The age of the victim. victim_sex: The sex of the victim, usually Male or Female. city: The city where the homicide occurred. state: The state where the homicide occurred. lat: The latitude of the homicide location. lon: The longitude of the homicide location. disposition: The outcome or current status of the case. city_state: A derived field combining the city and state for each record, added during the analysis for grouping purposes. unsolved: A binary variable indicating whether a homicide is unsolved

```r
library(dplyr)

homicide_data <- read.csv("homicide-data.csv", stringsAsFactors = FALSE)


homicide_data <- homicide_data %>%
  mutate(city_state = paste(city, state, sep = ", "))


homicide_data <- homicide_data %>%
  mutate(unsolved = disposition %in% c("Closed without arrest", "Open/No arrest"))


city_summary <- homicide_data %>%
  group_by(city_state) %>%
  summarise(
    total_homicides = n(),
    unsolved_homicides = sum(unsolved, na.rm = TRUE)
  )


print(city_summary)
```

```
## # A tibble: 51 x 3
##    city_state      total_homicides unsolved_homicides
##    <chr>                     <int>              <int>
##  1 Albuquerque, NM             378                146
##  2 Atlanta, GA                 973                373
##  3 Baltimore, MD              2827               1825
##  4 Baton Rouge, LA             424                196
##  5 Birmingham, AL              800                347
##  6 Boston, MA                  614                310
##  7 Buffalo, NY                 521                319
##  8 Charlotte, NC               687                206
##  9 Chicago, IL                5535               4073
## 10 Cincinnati, OH              694                309
## # i 41 more rows
```

```r
library(dplyr)
library(broom)


baltimore_data <- homicide_data %>%
  filter(city == "Baltimore", state == "MD")
```

```r
unsolved_count <- sum(baltimore_data$unsolved, na.rm = TRUE)
total_count <- nrow(baltimore_data)


prop_test_result <- prop.test(x = unsolved_count, n = total_count)


tidy_result <- broom::tidy(prop_test_result)


estimated_proportion <- tidy_result$estimate
confidence_interval <- tidy_result[c("conf.low", "conf.high")]


list(estimated_proportion = estimated_proportion, confidence_interval = confidence_interval)
```

```
## $estimated_proportion
##         p
## 0.6455607
##
## $confidence_interval
## # A tibble: 1 x 2
##   conf.low conf.high
##      <dbl>     <dbl>
## 1    0.628     0.663
```

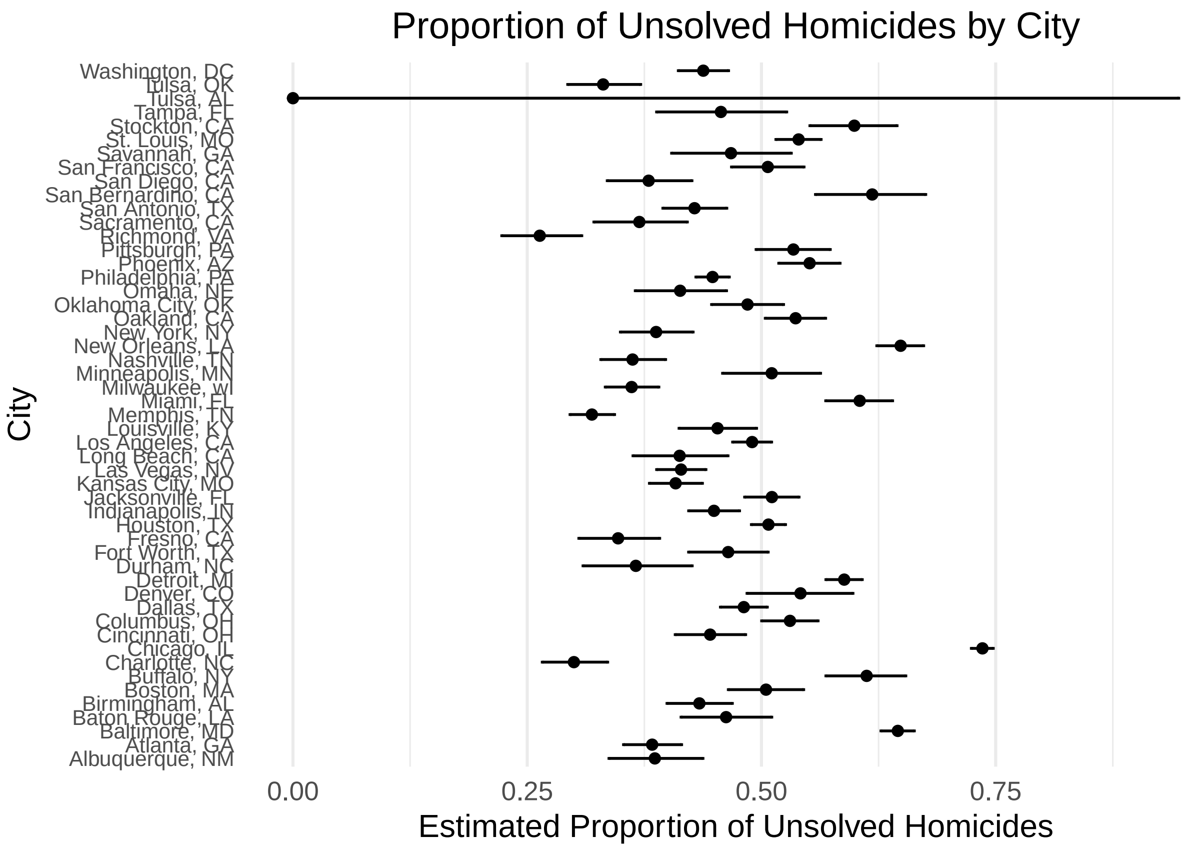
```r
library(dplyr)
library(purrr)
library(broom)
library(tidyr)


city_summary <- homicide_data %>%
  mutate(city_state = paste(city, state, sep = ", "),
         unsolved = disposition %in% c("Closed without arrest", "Open/No arrest")) %>%
  group_by(city_state) %>%
  summarise(
    unsolved_count = sum(unsolved, na.rm = TRUE),
    total_count = n()
  ) %>%
  ungroup()


prop_test_results <- city_summary %>%
  mutate(
    test_result = map2(unsolved_count, total_count, ~ prop.test(x = .x, n = .y) %>% tidy())
  ) %>%
  unnest(test_result)
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `test_result = map2(...)`.
## Caused by warning in `prop.test()`:
## ! Chi-squared approximation may be incorrect
```

```r
city_proportions <- prop_test_results %>%
  select(city_state, estimate, conf.low, conf.high)


print(city_proportions)
```

```
## # A tibble: 51 x 4
##    city_state      estimate conf.low conf.high
##    <chr>              <dbl>    <dbl>     <dbl>
##  1 Albuquerque, NM    0.386    0.337     0.438
##  2 Atlanta, GA        0.383    0.353     0.415
##  3 Baltimore, MD      0.646    0.628     0.663
##  4 Baton Rouge, LA    0.462    0.414     0.511
##  5 Birmingham, AL     0.434    0.399     0.469
##  6 Boston, MA         0.505    0.465     0.545
##  7 Buffalo, NY        0.612    0.569     0.654
##  8 Charlotte, NC      0.300    0.266     0.336
##  9 Chicago, IL        0.736    0.724     0.747
## 10 Cincinnati, OH     0.445    0.408     0.483
## # i 41 more rows
```

```r
ggplot(city_proportions, aes(x = city_state, y = estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = conf.low, ymax = conf.high), width = 0.2) +
  coord_flip() +
  labs(
    title = "Proportion of Unsolved Homicides by City",
    x = "City",
    y = "Estimated Proportion of Unsolved Homicides"
  ) +
  theme_minimal(base_size = 12) +  # Increase font size for readability
  theme(
    axis.text.y = element_text(size = 8),  # Adjust city label size
    plot.title = element_text(hjust = 0.5),  # Center the title
    panel.grid.major.y = element_blank()  # Reduce grid lines for cleaner look
  )
```


Proportion of Unsolved Homicides by City
City
Washington, DC
Tulsa, OK
Tulsa, AL
Tampa, FL
Stockton, CA
St. Louis, MO
Savannah, GA
San Francisco, CA
San Diego, CA
San Bernardino, CA
San Antonio, TX
Sacramento, CA
Richmond, VA
Pittsburgh, PA
Phoenix, AZ
Philadelphia, PA
Omaha, NE
Oklahoma City, OK
Oakland, CA
New York, NY
New Orleans, LA
Nashville, TN
Minneapolis, MN
Milwaukee, wI
Miami, FL
Memphis, TN
Louisville, KY
Los Angeles, CA
Long Beach, CA
Las Vegas, NV
Kansas City, MO
Jacksonville, FL
Indianapolis, IN
Houston, TX
Fresno, CA
Fort Worth, TX
Durham, NC
Detroit, MI
Denver, CO
Dallas, TX
Columbus, OH
Cincinnati, OH
Chicago, IL
Charlotte, NC
Buffalo, NY
Boston, MA
Birmingham, AL
Baton Rouge, LA
Baltimore, MD
Atlanta, GA
Albuquerque, NM
0.00
0.25
0.50
0.75
Estimated Proportion of Unsolved Homicides