MATH 3070 Lab Project 12

Prachi Aswani

21 November, 2024

Contents

Remember: I expect to see commentary either in the text, in the code with comments created using **#***, or (preferably) both!* ***Failing to do so may result in lost points!***

Problem 1 (Verzani problem 8.7)

Of the last ten times you've dropped your toast, it landed sticky-side down nine times. If these are a random sample from the Ber(p) distribution, find an 80% confidence interval for p, the probability of the stidy side landing down. (Use `binconf()` ***(Hmisc)*** *to compute the score interval.)*

```
# Load necessary library
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
# Define the observed data
successes <- 9  # Number of times toast lands sticky-side down
trials <- 10    # Total trials

# Compute the 80% confidence interval using the score method
ci <- binconf(successes, trials, alpha = 0.2,method = "wilson")

# Print results
ci
```

```
##  PointEst     Lower     Upper
##       0.9 0.7175557 0.9776856
```

Problem 2 (Verzani problem 8.10)

A survey is taken of 250 students, and a $\hat{p}$ of 0.45 is found. The same survey is repeated with 1000 students, and the same $\hat{p}$ is found. Compare the two 95% confidence intervals. What is the relationship? Is the margin of error for the second one four times smaller? If not, how much smaller is it? (Use `binom.test()` to answer this problem.)

```r
# Define observed proportions and sample sizes
p_hat <- 0.45
n1 <- 250  # First sample size
n2 <- 1000 # Second sample size

# Compute the number of successes (must be integer)
successes_n1 <- round(p_hat * n1)  # Round to the nearest integer
successes_n2 <- round(p_hat * n2)

# Compute 95% confidence intervals using binom.test()
ci_n1 <- binom.test(successes_n1, n1)$conf.int
ci_n2 <- binom.test(successes_n2, n2)$conf.int

# Print results
cat("95% CI for n = 250:", ci_n1, "\n")
```

```
## 95% CI for n = 250: 0.3852992 0.5119484
```

```r
cat("95% CI for n = 1000:", ci_n2, "\n")
```

```
## 95% CI for n = 1000: 0.4188517 0.4814435
```

```r
# Calculate margin of error ratios
moe_n1 <- diff(ci_n1) / 2
moe_n2 <- diff(ci_n2) / 2
ratio <- moe_n1 / moe_n2

cat("Margin of error for n = 1000 is", ratio, "times smaller than n = 250.\n")
```

```
## Margin of error for n = 1000 is 2.023417 times smaller than n = 250.
```

Problem 3 (Verzani problem 8.15)

*The `stud.recs` (**UsingR**) data set contains a sample of math SAT scores from some population in the variable `sat.m`. Find a 90% confidence interval for the mean math SAT score for this data. (Do not use ‘t.test(); find this confidence interval “by hand”.)*

```r
# Load necessary library
library(UsingR)
```

```
## Warning: package 'UsingR' was built under R version 4.3.3
```

```
## Loading required package: MASS
```

```
## Loading required package: HistData

## Warning: package 'HistData' was built under R version 4.3.3
```

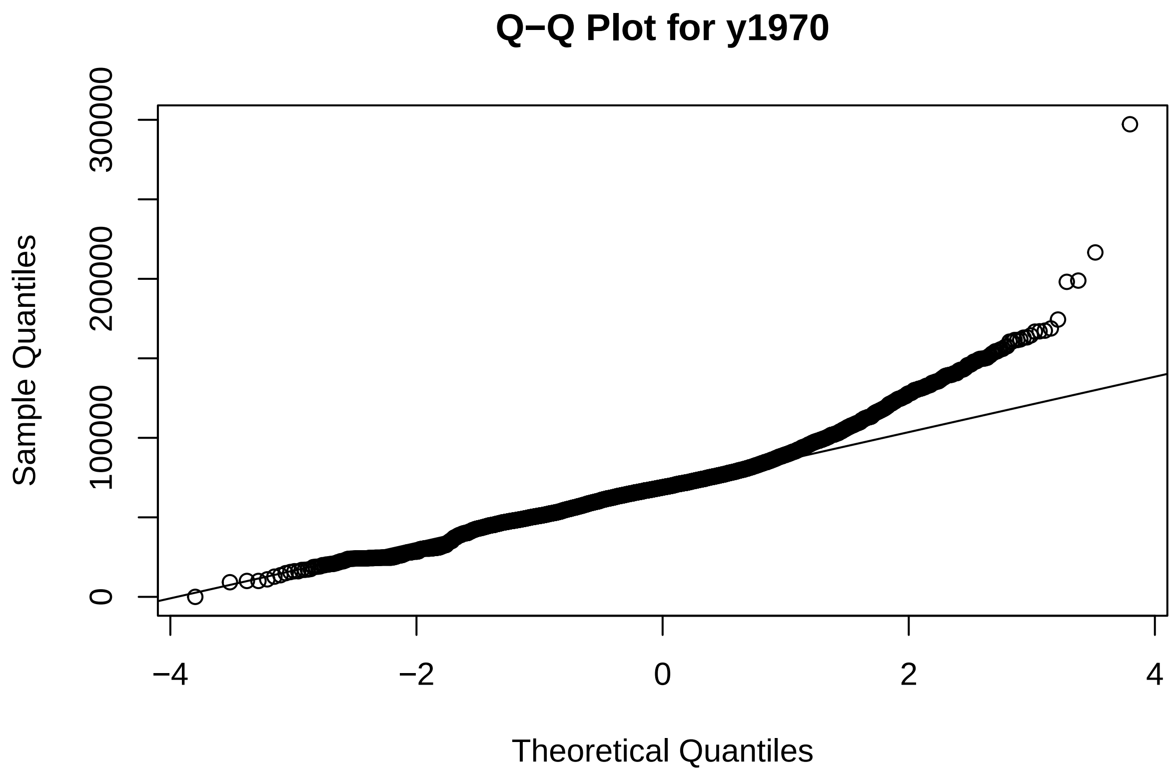
```
# Extract data
data("stud.recs")
scores <- stud.recs$sat.m

# Calculate statistics
mean_score <- mean(scores, na.rm = TRUE)
sd_score <- sd(scores, na.rm = TRUE)
n <- length(scores)

# Find critical t-value
alpha <- 0.1  # 90% confidence level
t_crit <- qt(1 - alpha/2, df = n - 1)

# Compute confidence interval
moe <- t_crit * (sd_score / sqrt(n))  # Margin of error
ci <- c(mean_score - moe, mean_score + moe)

# Print results
cat("90% Confidence Interval for Mean Math SAT Score:", ci, "\n")
```

```
## 90% Confidence Interval for Mean Math SAT Score: 476.8953 494.9797
```

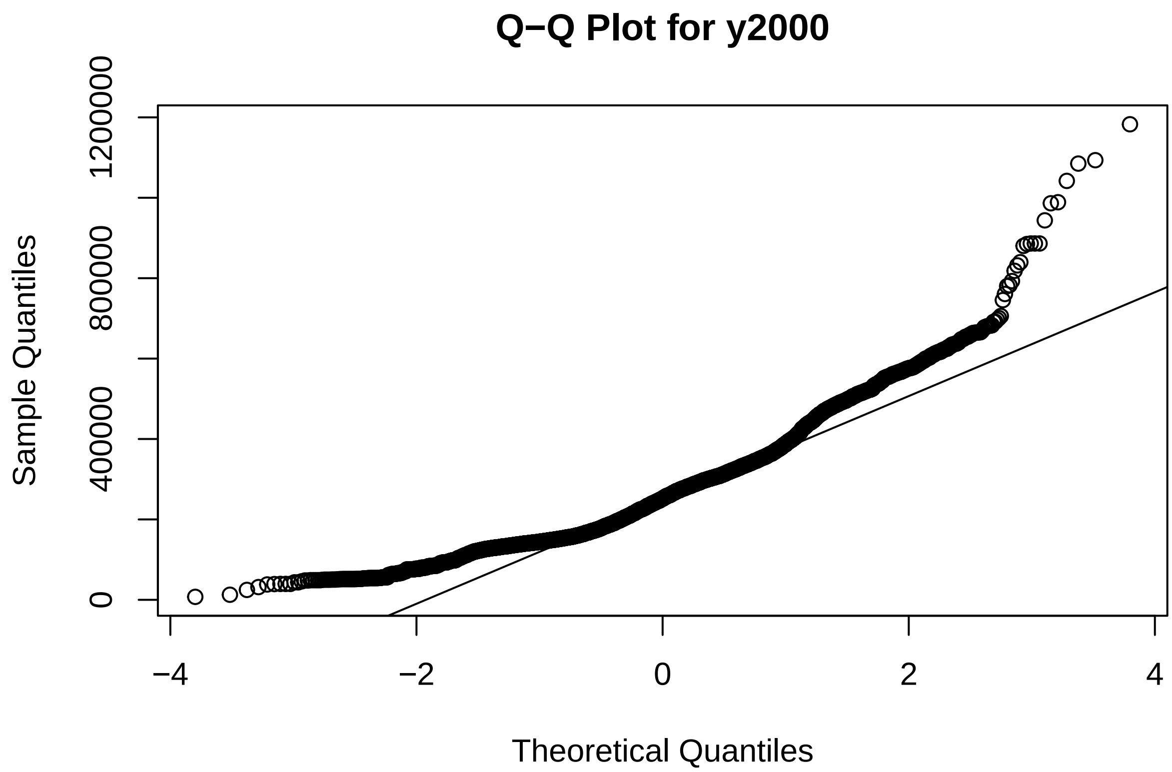
Problem 4 (Verzani problem 8.15)

For the **`homedata`** *(**UsingR**) data set find 90% confidence intervals for both variables* **`y1970`** *and* **`y2000`**, *assuming the sample represents some population. Perform one sample t-test for each variable, use* **`t.test()`**, *but first discuss whether the model assumptions are appropriate (include some check of the assumptions, like a Q-Q plot).*

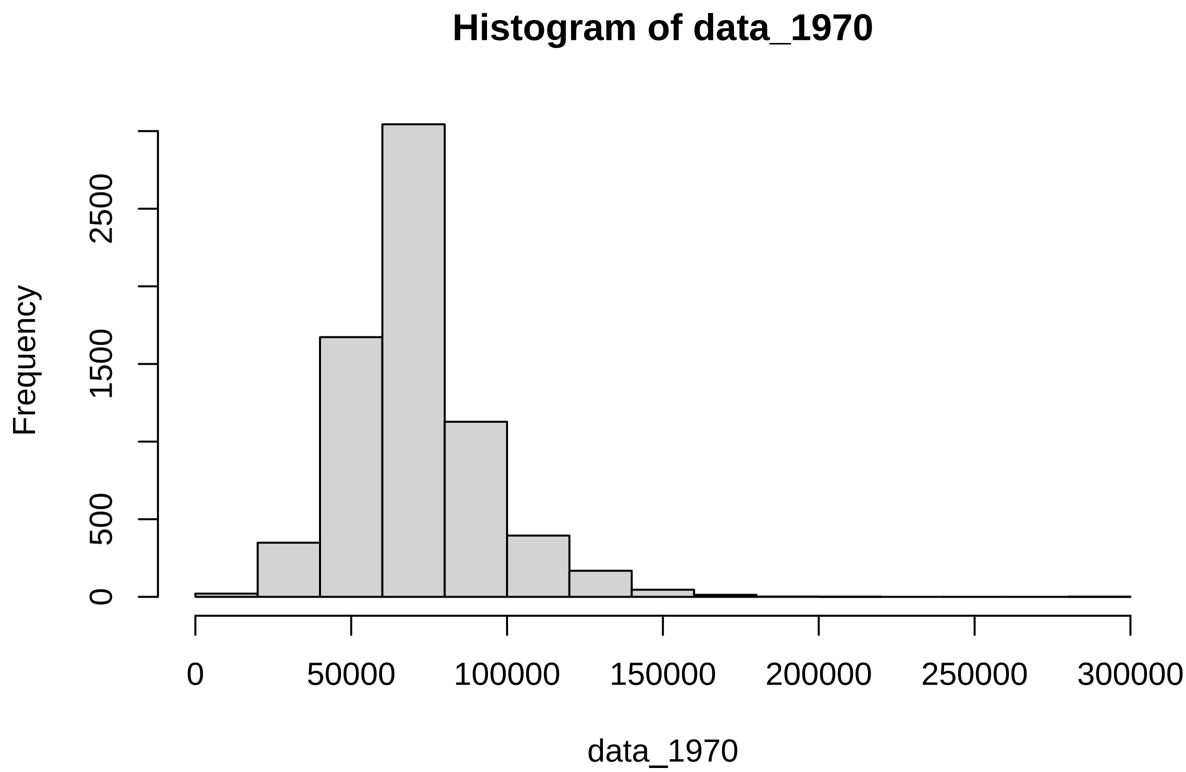
```
# Your solution here
# Load necessary library
data("homedata")

# Variables to analyze
data_1970 <- homedata$y1970
data_2000 <- homedata$y2000

# Check normality with Q-Q plots
qqnorm(data_1970, main = "Q-Q Plot for y1970")
qqline(data_1970)
```



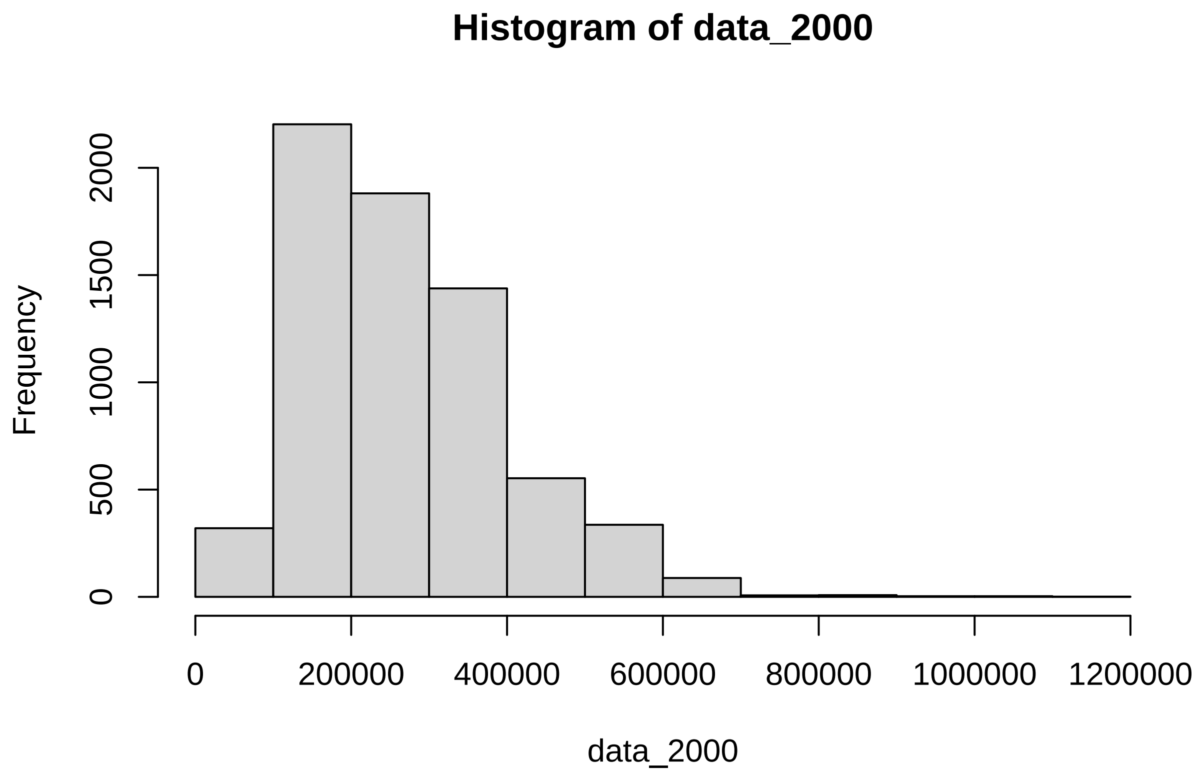

```r
qqnorm(data_2000, main = "Q-Q Plot for y2000")
qqline(data_2000)
```




```
hist(data_1970)
```




```r
hist(data_2000)
```




```r
# Perform one-sample t-tests and calculate 90% CIs
t_test_1970 <- t.test(data_1970, conf.level = 0.90)
t_test_2000 <- t.test(data_2000, conf.level = 0.90)

# Print results
cat("90% CI for y1970:", t_test_1970$conf.int, "\n")
```

```
## 90% CI for y1970: 70377.72 71264.14
```

```r
cat("90% CI for y2000:", t_test_2000$conf.int, "\n")
```

```
## 90% CI for y2000: 265769.5 270970
```