HW6

Daniel Koslovsky

2024-12-07

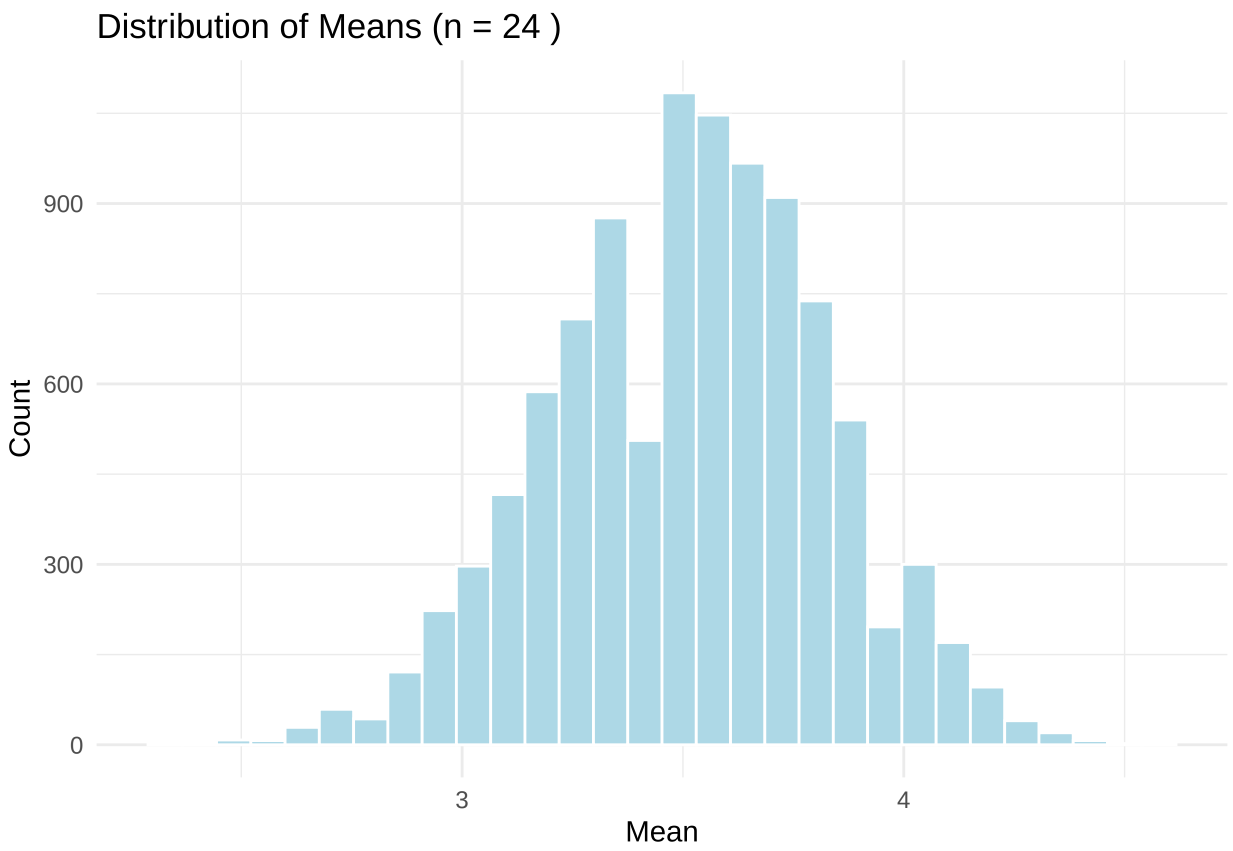
This submission is my work alone and complies with the 31202 integrity policy. Add you initials to indicate your agreement: DK

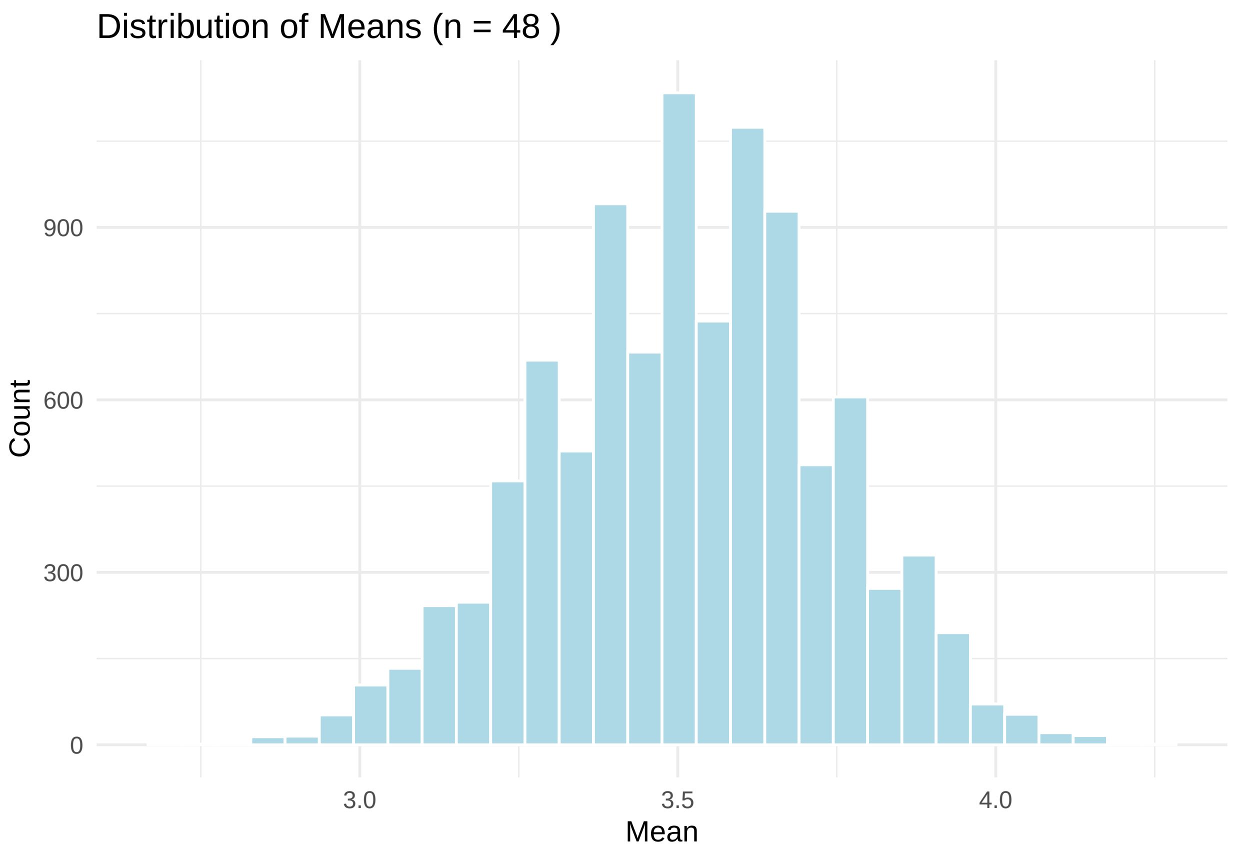
HW6

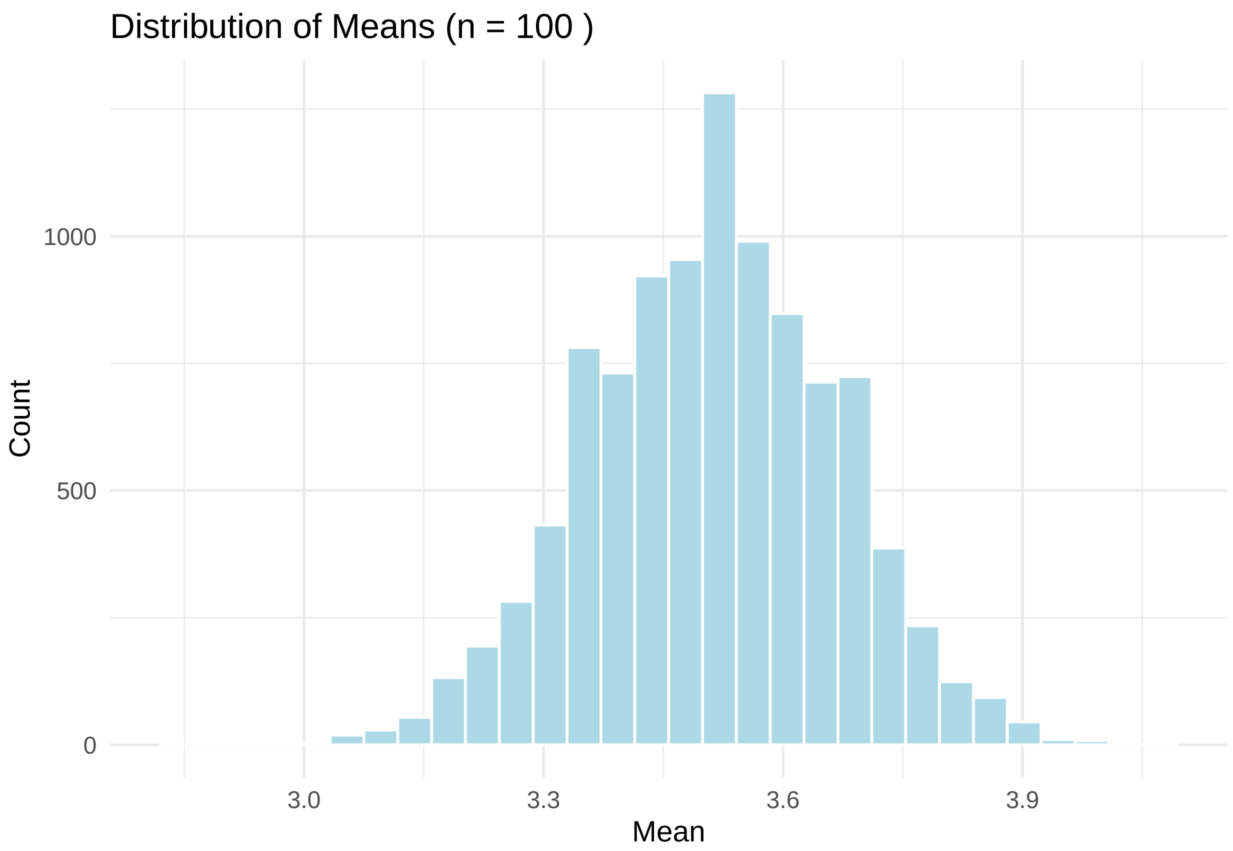
```
library(tidyverse)
```

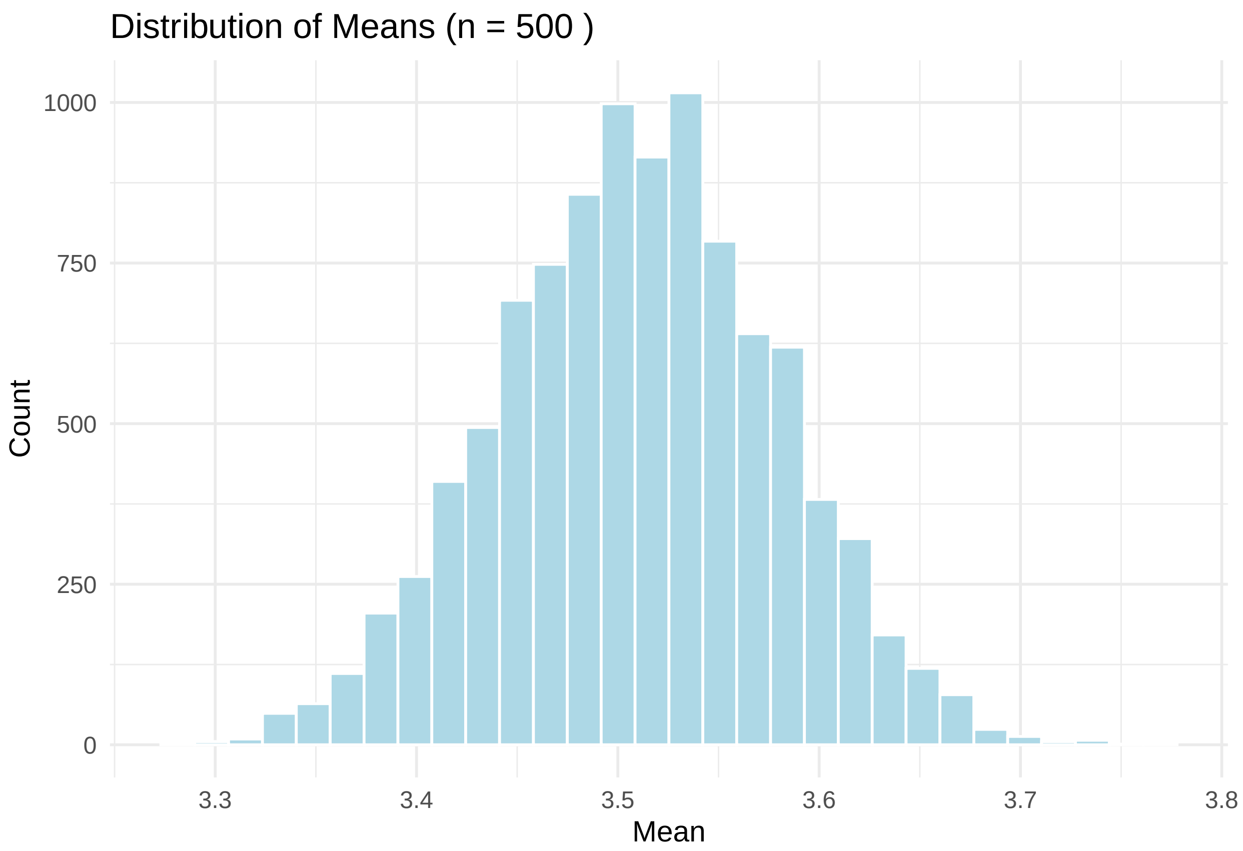
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

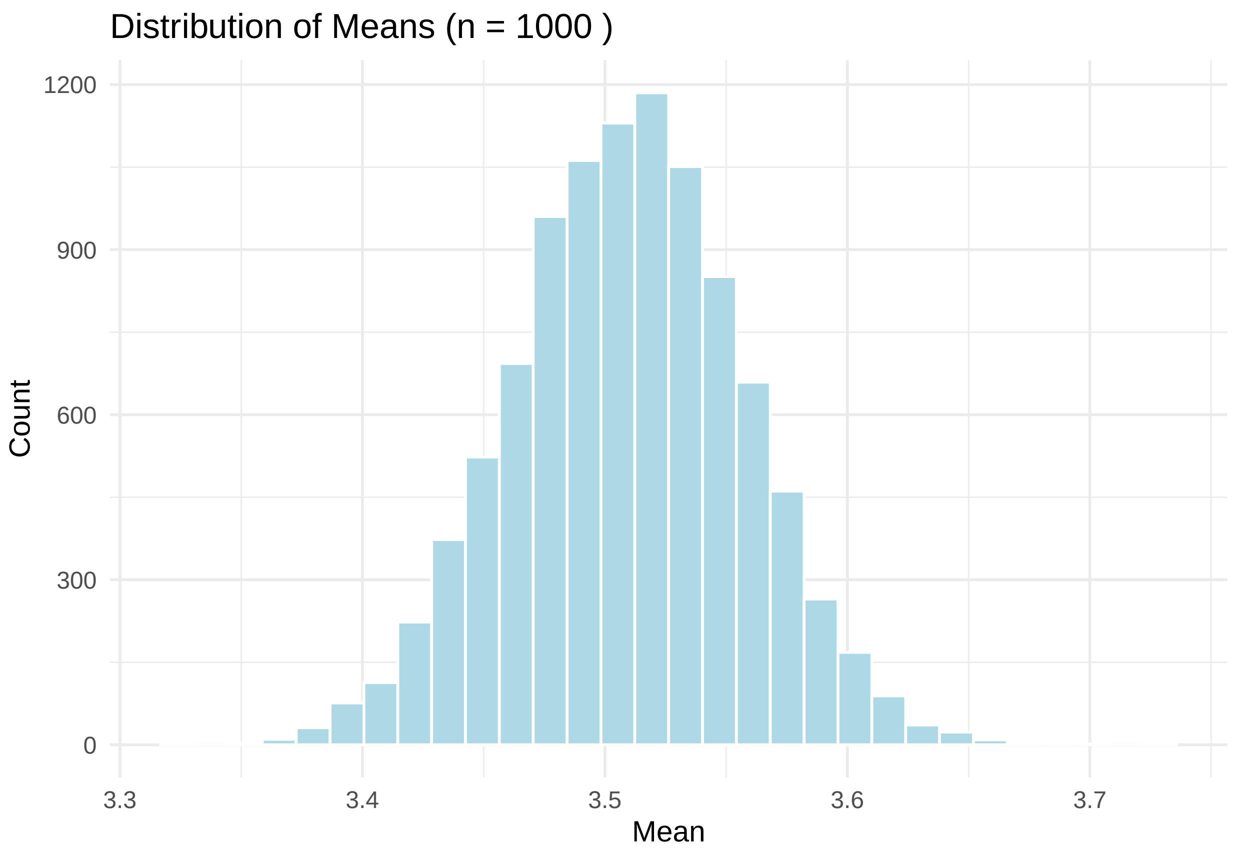
```
library(tinytex)

# Question 1
set.seed(312101)

true_mean <- 1 * 0.2 + 2 * 0.1 + 3 * 0.03 + 4 * 0.33 + 5 * 0.34
alpha <- 2.069  # critical value at 5% two-tailed test, df = 23 (approx.)
N <- 10000  # Number of simulations per sample size

sample_sizes <- c(24, 48, 100, 500, 1000, 10000)

all_results <- lapply(sample_sizes, function(n) {
    # Repeat the simulation N times for the given sample
    # size n
    results <- replicate(N, {
        # Sample data for the given n
        n_sample <- sample(x = c(1, 2, 3, 4, 5), size = n, prob = c(0.2,
            0.1, 0.03, 0.33, 0.34), replace = TRUE)

        # Compute sample statistics
        mean_n <- mean(n_sample)
        se_n <- sd(n_sample)/sqrt(n)
        z_score_n <- (mean_n - true_mean)/se_n
        h_test_n <- ifelse(abs(z_score_n) > alpha, "Reject H0",
            "Fail to Reject H0")
        left_or_right_tail_n <- ifelse(mean_n - true_mean > 0,
            "Right Tail", "Left Tail")

        c(sample_size = n, mean = mean_n, se = se_n, z_score = z_score_n,
            decision = h_test_n, tail = left_or_right_tail_n)
    })

    results_df <- as.data.frame(t(results))

    # Convert columns to appropriate types
    results_df$sample_size <- as.numeric(results_df$sample_size)
    results_df$mean <- as.numeric(results_df$mean)
    results_df$se <- as.numeric(results_df$se)
    results_df$z_score <- as.numeric(results_df$z_score)

    # Count how many left and right tail rejections
    left_tail_count <- results_df %>%
        filter(decision == "Reject H0", tail == "Left Tail") %>%
        nrow()

    right_tail_count <- results_df %>%
        filter(decision == "Reject H0", tail == "Right Tail") %>%
        nrow()

    # Run binomial tests
    left_binomial_test <- binom.test(x = left_tail_count, n = N,
        p = 0.025, alternative = "two.sided")
    right_binomial_test <- binom.test(x = right_tail_count, n = N,
        p = 0.025, alternative = "two.sided")

    # Create a ggplot histogram of the means for this
    # sample size
    p <- ggplot(results_df, aes(x = mean)) + geom_histogram(fill = "lightblue",
        color = "white", bins = 30) + labs(title = paste("Distribution of Means (n =",
        n, ")"), x = "Mean", y = "Count") + theme_minimal()

    print(p)

    # Return results for this sample size
    list(sample_size = n, results_df = results_df, left_tail_count = left_tail_count,
        right_tail_count = right_tail_count, left_binomial_test = left_binomial_test,
        right_binomial_test = right_binomial_test)
})
```

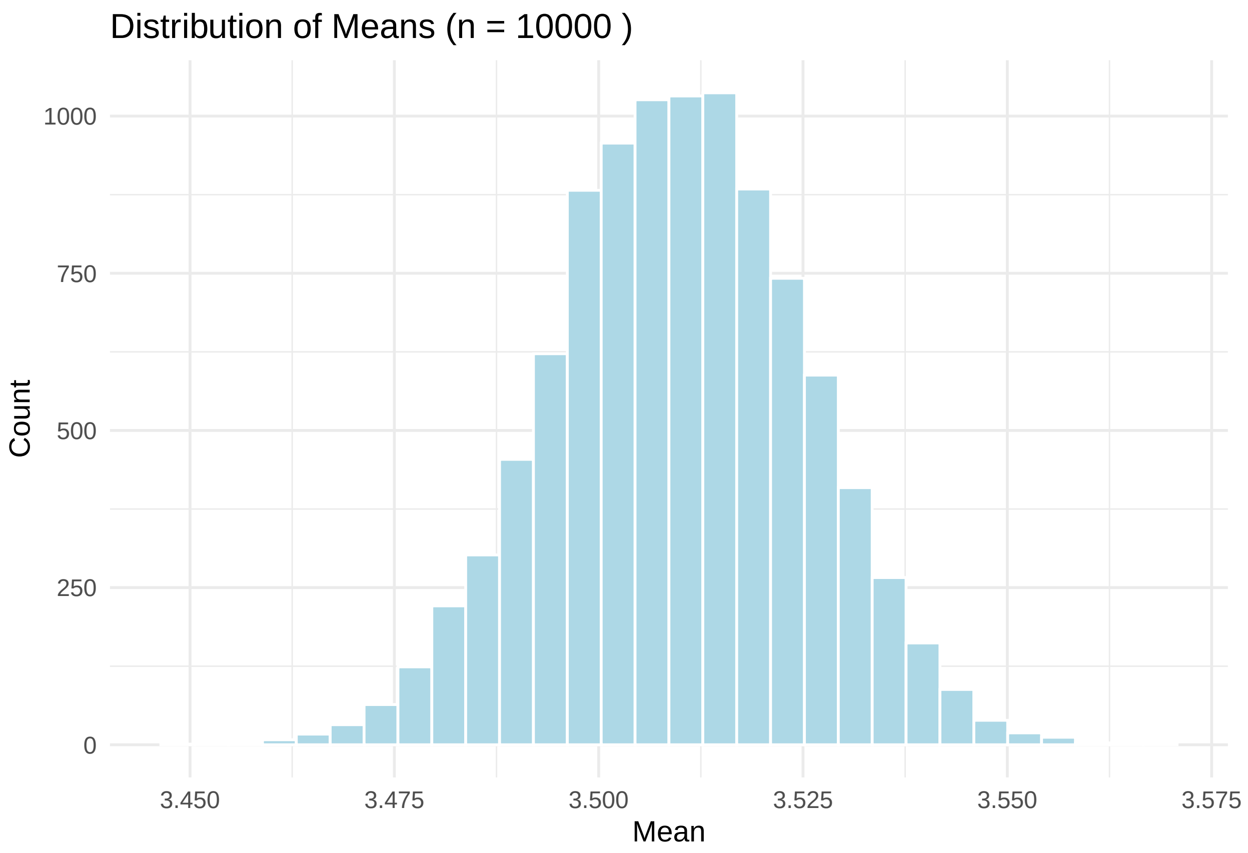

Distribution of Means (n = 24)
Count
900
600
300
0
3
4
Mean


Distribution of Means (n = 48)
Count
900
600
300
0
3.0
3.5
4.0
Mean


Distribution of Means (n = 100)
Count
1000
500
0
3.0
3.3
3.6
3.9
Mean


Distribution of Means (n = 500)
Count
1000
750
500
250
0
3.3
3.4
3.5
3.6
3.7
3.8
Mean


Distribution of Means (n = 1000)
Count
1200
900
600
300
0
3.3
3.4
3.5
3.6
3.7
Mean



```
# Initialize a matrix to store 'Passed'/'Not Passed'
# results
res_matrix <- matrix(NA, nrow = 2, ncol = length(sample_sizes))
rownames(res_matrix) <- c("Left Tail", "Right Tail")
colnames(res_matrix) <- sample_sizes

# Fill the matrix based on the binomial test p-values
for (i in seq_along(sample_sizes)) {
    left_pval <- all_results[[i]]$left_binomial_test$p.value
    right_pval <- all_results[[i]]$right_binomial_test$p.value

    # Check significance at alpha = 0.05
    res_matrix["Left Tail", i] <- ifelse(left_pval < 0.05, "Passed",
        "Not Passed")
    res_matrix["Right Tail", i] <- ifelse(right_pval < 0.05,
        "Passed", "Not Passed")
}


final_df <- as.data.frame(res_matrix)
print(final_df)
```

```
##                24     48        100    500   1000  10000
## Left Tail  Passed Passed     Passed Passed Passed Passed
## Right Tail Passed Passed Not Passed Passed Passed Passed
```

The normal apporoximation works well for most sample sizes, especially for sample sizes > 500 observations

```
pset5_simulated_data <- read.csv("pset5_simulated_data.csv")

contingency_table <- table(pset5_simulated_data$treatment, pset5_simulated_data$disease)

# a - exact test
fisher_test <- fisher.test(contingency_table)
print(fisher_test)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  contingency_table
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.2311482 0.2732357
## sample estimates:
## odds ratio
##   0.251402
```

```
# b - asymptotic test
chi2_test <- chisq.test(contingency_table)
print(chi2_test)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  contingency_table
## X-squared = 1201.1, df = 1, p-value < 2.2e-16
```

Both the exact test and asymptotic test conclude that the treatment did reduce the faction of individuals contracting the disease.

```
# Question 3 a
t_statistic <- (20.4 - 0)/(314.5/sqrt(1600))
five_percent_level_test <- ifelse(t_statistic > 1.645, "Reject H0",
    "Fail to Reject H0")
print(five_percent_level_test)
```

```
## [1] "Reject H0"
```

```
one_percent_level_test <- ifelse(t_statistic > 2.326, "Reject H0",
    "Fail to Reject H0")
print(one_percent_level_test)
```

```
## [1] "Reject H0"
```

```
# b
normal_distribution_sample_n <- c(500, 750, 10000)
normal_samples <- vector("list", length(normal_distribution_sample_n))
mean <- numeric(length(normal_distribution_sample_n))
sd <- numeric(length(normal_distribution_sample_n))
set.seed(312101)

for (n in seq_along(normal_distribution_sample_n)) {
    normal_samples[[n]] <- rnorm(normal_distribution_sample_n[[n]],
        15, 5)
    mean[[n]] <- mean(normal_samples[[n]])
    sd[[n]] <- sd(normal_samples[[n]])
}

results <- data.frame(sample_size = normal_distribution_sample_n,
    mean = mean, sd = sd)

print(results)
```

```
##   sample_size     mean       sd
## 1         500 14.77710 5.185977
## 2         750 14.90023 5.018389
## 3       10000 14.92428 4.998991
```

```
t_statistic_500 <- (results$mean[1] - 0)/(results$sd[1]/sqrt(results$sample_size[1]))

test_at_5_percent <- ifelse(t_statistic_500 > 1.645, "Reject H0",
    "Fail to Reject H0")
print(test_at_5_percent)
```

```
## [1] "Reject H0"
```

```
test_at_1_percent <- ifelse(t_statistic_500 > 2.326, "Reject H0",
    "Fail to Reject H0")
print(test_at_1_percent)
```

```
## [1] "Reject H0"
```

a) The t-statistic is is greater than the critical value at 5% of 1.645 and the critical value at the 1% level of 2.326. So we can reject H0 that there was no increase in the ridership.

b) The t-statistic is is greater than the critical value at 5% of 1.645 and the critical value at the 1% level of 2.326. So we can reject H0 that there was no increase in the ridership.

c) The mean for Yi, ranges from 14.777 to 14.924 depending on the sample size. This means that the predicted average change in ridership per Chicagoan would be almost 15 riders per person. This is a large change and thus practically significant. We are able to reject H0 at the 1% significance level so the change is also statistically significant.

```
# Question 4
matching <- read.csv("matching.csv")

# a
summary(matching)
```

```
##       fips             x              lhmed90          frsing90
##  Min.   : 1001   Min.   :-1.8550   Min.   : 9.616   Min.   :0.0000
##  1st Qu.:19045   1st Qu.:-0.1260   1st Qu.:10.513   1st Qu.:0.1471
##  Median :29212   Median : 0.1873   Median :10.719   Median :0.1805
##  Mean   :30672   Mean   : 0.2286   Mean   :10.778   Mean   :0.1889
##  3rd Qu.:46008   3rd Qu.: 0.5587   3rd Qu.:11.000   3rd Qu.:0.2204
##  Max.   :56045   Max.   : 2.0325   Max.   :13.097   Max.   :0.5886
##                  NA’s   :260       NA’s   :1
##     lpop90          treat              pop
##  Min.   : 4.673   Min.   :0.0000   Min.   :    107
##  1st Qu.: 9.254   1st Qu.:0.0000   1st Qu.:  10445
##  Median :10.012   Median :1.0000   Median :  22295
##  Mean   :10.138   Mean   :0.5959   Mean   :  79509
##  3rd Qu.:10.916   3rd Qu.:1.0000   3rd Qu.:  55049
##  Max.   :15.997   Max.   :1.0000   Max.   :8863166
##  NA’s   :1                         NA’s   :1
```

```
# b
housing_indicator_model <- lm(x ~ treat + frsing90 + lpop90,
    data = matching)
summary(housing_indicator_model)
```

```
##
## Call:
## lm(formula = x ~ treat + frsing90 + lpop90, data = matching)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.01474 -0.25019 -0.01298  0.23326  1.77694
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.256183   0.061126   36.91  < 2e-16 ***
## treat       -0.101673   0.015787   -6.44 1.39e-10 ***
## frsing90    -5.680112   0.108429  -52.39  < 2e-16 ***
## lpop90      -0.083677   0.006399  -13.08  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3516 on 2844 degrees of freedom
##   (260 observations deleted due to missingness)
## Multiple R-squared:   0.56,  Adjusted R-squared:  0.5596
## F-statistic:  1207 on 3 and 2844 DF,  p-value: < 2.2e-16
```

frsing90 is the fraction of female headed households in within the county in the 1990 census. Its coefficient can be interpreted as the average difference in impact on the percentile income of people who were in the 25th percentile of income when they were adolescents in counties with all male headed households versus all female headed households. Going from a county with all female headed households to one with all male headed households is expected to reduce where someone ends up in the income distribution by 5.68 percentage points.

```
housing_price_model <- lm(x ~ lhmed90 + frsing90 + lpop90, data = matching)
summary(housing_price_model)
```

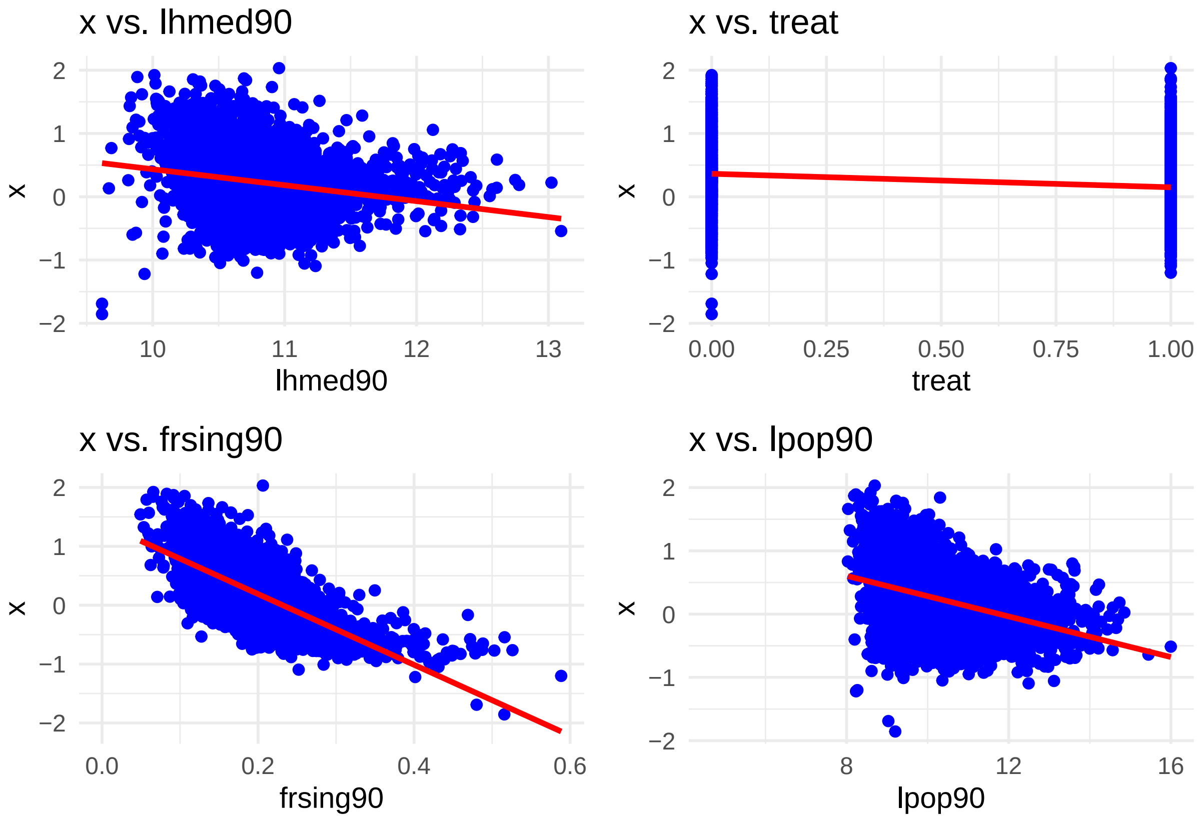
```
##
## Call:
## lm(formula = x ~ lhmed90 + frsing90 + lpop90, data = matching)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.05688 -0.25652 -0.01548  0.23205  1.73645
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.90860    0.18538  15.690  < 2e-16 ***
## lhmed90     -0.06039    0.02063  -2.927  0.00345 **
## frsing90    -5.65052    0.10940 -51.650  < 2e-16 ***
## lpop90      -0.09034    0.00735 -12.291  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3536 on 2844 degrees of freedom
##   (260 observations deleted due to missingness)
## Multiple R-squared:  0.555,  Adjusted R-squared:  0.5545
## F-statistic:  1182 on 3 and 2844 DF,  p-value: < 2.2e-16
```

The coefficient on lhmed90 can be interpreted as the average change in the percentile income of people who were in the 25th percentile of income when they were adolescents from a 1 percentage point change in median house price in a person's county. A 1 percentage point change in median house price is expected to reduce where someone is on the income distribution by .06%. This is similar, but smaller than the effect of treat var used in the first model, which measured the effect of whether the median house price in a county was above or below $100k.

```r
library(patchwork)
p1 <- ggplot(matching, aes(x = lhmed90, y = x)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "x vs. lhmed90", x = "lhmed90", y = "x") +
  theme_minimal()

p2 <- ggplot(matching, aes(x = treat, y = x)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "x vs. treat", x = "treat", y = "x") +
  theme_minimal()

p3 <- ggplot(matching, aes(x = frsing90, y = x)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "x vs. frsing90", x = "frsing90", y = "x") +
  theme_minimal()

p4 <- ggplot(matching, aes(x = lpop90, y = x)) +
  geom_point(color = "blue") +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "x vs. lpop90", x = "lpop90", y = "x") +
  theme_minimal()
```

```r
# Combine the four plots into a 2x2 grid using patchwork
final_plot <- (p1 + p2) / (p3 + p4)

# Print the combined figure
print(final_plot)
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 260 rows containing non-finite outside the scale range
## (`stat_smooth()`).

## Warning: Removed 260 rows containing missing values or values outside the scale range
## (`geom_point()`).

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 260 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 260 rows containing missing values or values outside the scale range
## (`geom_point()`).

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 260 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 260 rows containing missing values or values outside the scale range
## (`geom_point()`).

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 260 rows containing non-finite outside the scale range
## (`stat_smooth()`).
## Removed 260 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


x vs. lhmed90
x
2
1
0
−1
−2
10
11
12
13
lhmed90
x vs. treat
x
2
1
0
−1
−2
0.00
0.25
0.50
0.75
1.00
treat
x vs. frsing90
x
2
1
0
−1
−2
0.0
0.2
0.4
0.6
frsing90
x vs. lpop90
x
2
1
0
−1
−2
8
12
16
lpop90