ST502 Project

Halid Kopanski and Justin Feathers

Part I

Framingham Heart Study

The Framingham data set contains the diastolic blood pressure of 300 smokers and nonsmokers. For this study we will assume the following:

H_0: μ_{ns} - $\mu_{\text{s}} = 0$

H_A: μ_{ns} - $\mu_{\text{s}} \neq 0$

The null hypothesis being there is no difference in the blood pressure between smokers and non smokers. We believe that there is a difference and this paper will prove if we have enough evidence to reject the null hypothesis.

Normality of Data

Plotting the density of total data (see Appendix I: Figure 1A), we can that it closely follows a normal distribution, Density plots of the split data exhibit similar forms (see Appendix I: Figure 1B). We can also calculate the kurtosis and the skew of the data, which are 3.812, 0.88 for kurtosis and skew respectively, and see that they are quite close to values typically found in normally distributed data. Additionally, the calculated values of sample standard deviation, $\frac{IQR}{1.35}$, and $\frac{MAD}{0.675}$ are 22.89, 22.22, and 21.48. The similarities between the three values indicate that the data is not influenced by outliers. Therefore, we will assume the data and the subsequent split data to be normal.

Statistical Analysis

In the first analysis we will be assuming equal variances, thereby allowing us to use pooled sample variance (see Appendix I: Equation 1).

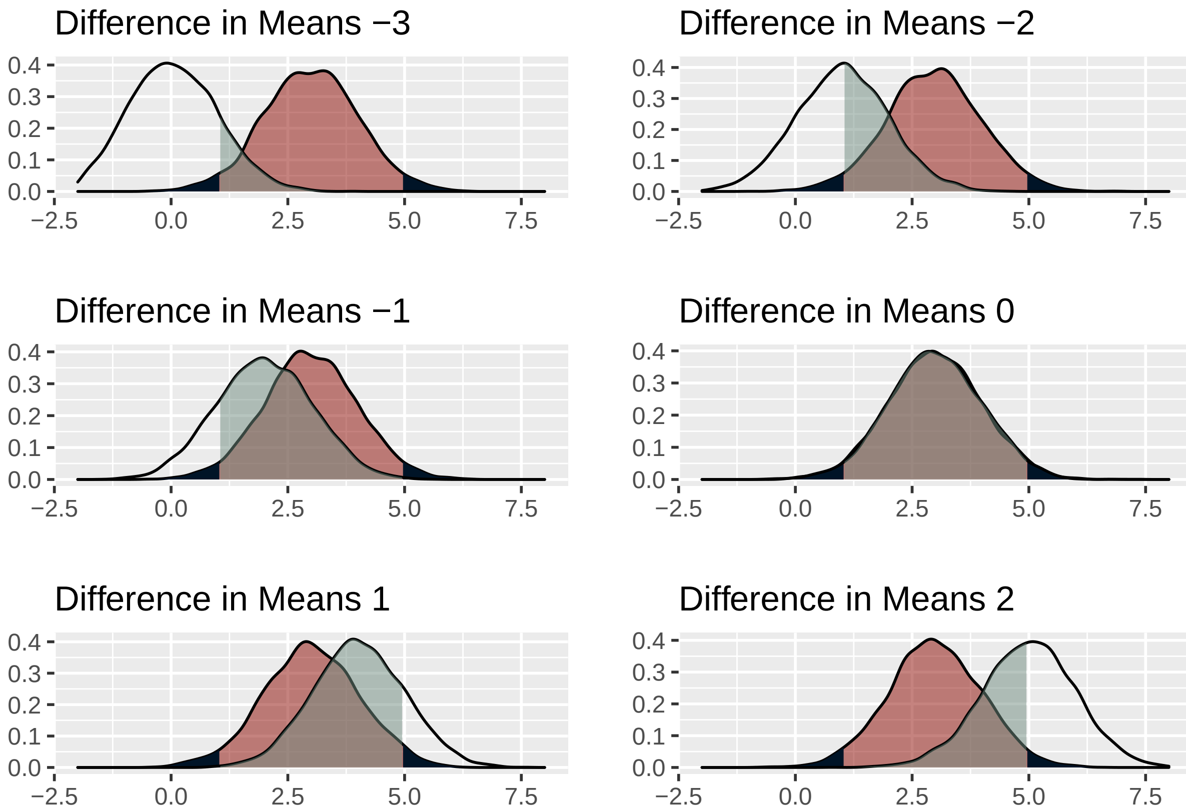
The pooled sample variance of the data is 510 using a degree of freedom value 298. In this case, we reject the null hypothesis because the calculated p value, 0.0041, is smaller than the chosen α value of 0.05. In terms of t values, our observed t value of -3.04 is smaller than the t value, -1.97 for a two sided α of 0.05.

When computing the observed t value using the assumption that the population variances are not equivalent (variance smoker is 352.2 and variance nonsmoker is 562.1), a value of -2.9 is obtained. Comparing that the two sided α of -1.96 with 693 degrees of freedom (as computed using the Satterthwaite Approximation see Appendix I: Equation 4). The observed t value is less than the chosen α value of 0.05. In case, there is sufficient evidence to reject the null hypothesis in favor of the alternative.

The observed 95% confidence intervals are -15.08 to -3.23 for pooled sample variance and -15.36 to -2.95 for non pooled sample variance. It can be seen that 0 does not fall into the 95% confidence interval in either case. Therefore the null hypothesis can be rejected.

In all three case, there is sufficient evidence to reject the null hypothesis and to support the alternative at an α level of 0.05.

Part II



As the alternate and null distribution get closer, it can be seen that β increases (power decreases).

Below, various scenarios were simluted. . . . . .

Appendix I: Equations and Figures

Figures

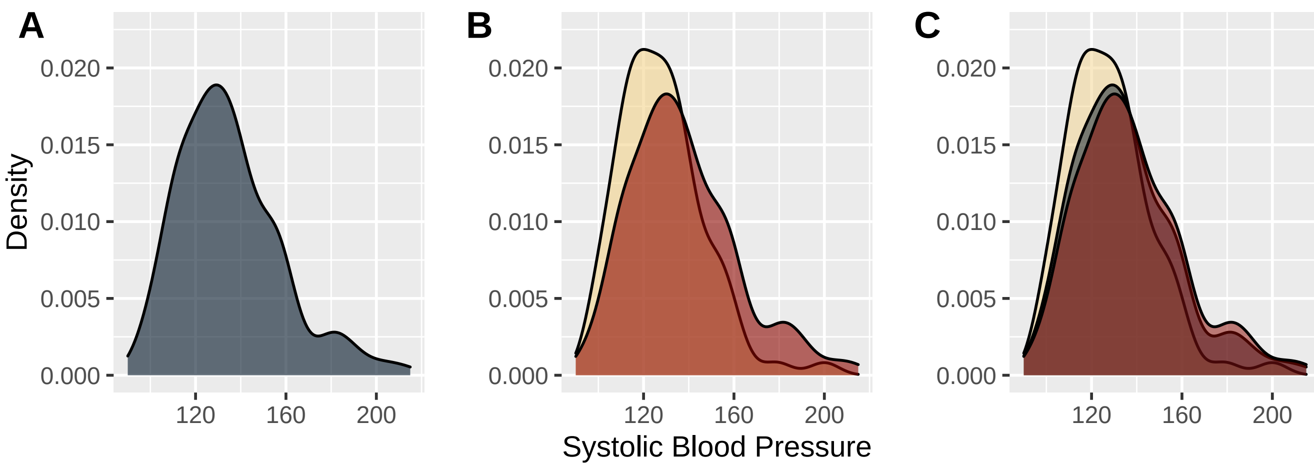


Figure 1: Data plots

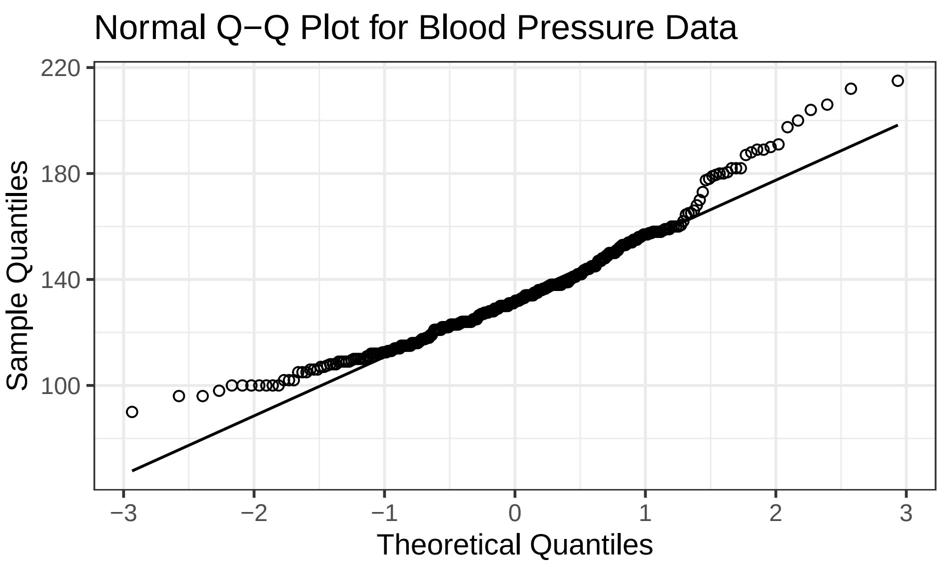


Figure 2: Q-Q Plot

Equations:

Equation 1: Pooled Sample Variance:

$$S_p^2 = \frac{(n_{ns} - 1)S_{ns}^2 + (n_s - 1)S_s^2}{n_{ns} + n_s - 2}$$

Equation 2: Observed T Statistic (pooled variance):

$$t_{obs} = \frac{\mu_{ns} - \mu_s}{S_p\sqrt{\frac{1}{n_{ns}} + \frac{1}{n_s}}}$$

Equation 3: Observed T Statistic (distinct variance):

$$t_{obs} = \frac{\mu_{ns} - \mu_s}{\sqrt{\frac{S_{ns}^2}{n_{ns}} + \frac{S_s^2}{n_s}}}$$

Equation 4: Satterthwaite Approximation:

$$\nu = \frac{\left(\frac{S_{ns}^2}{n_{ns}} + \frac{S_s^2}{n_s}\right)^2}{\frac{\frac{S_{ns}^2}{n_{ns}}}{n_{ns}-1} + \frac{\frac{S_s^2}{n_s}}{n_s-1}}$$

Appendix II: Results

Data Mean: 134.935
Data Sample Variance: 524.0852258

Smoker Mean: 128.0666667
Smoker Variance: 352.2117117

Nonsmoker Mean: 137.2244444
Nonsmoker Variance: 562.1447123

Difference in mean: -9.1577778

Pooled Sample Variance: 510.0136987
Nonpooled Sample Variance: 9.9936838

CI Pooled: -15.08, -3.23
CI Nonpooled: -15.36, -2.95

Appendix III: Part II Results

Test#	μ_0	σ_0	n_0	μ_a	σ_a	n_a	Test Results
1	0	1	10	5	1	10	Test Results 1: 1000
2	4	1	10	5	1	10	Test Results 2: 574
3	5	1	10	5	1	10	Test Results 3: 50
4	6	1	10	5	1	10	Test Results 4: 565
5	10	1	10	5	1	10	Test Results 5: 1000
6	5	1	10	5	1	10	Test Results 6: 39
7	5	1	10	5	1	10	Test Results 7: 55
8	5	1	10	5	1	10	Test Results 8: 61
9	6	1	10	5	1	10	Test Results 9: 570
10	6	1	10	5	1	10	Test Results 10: 262
11	6	1	10	5	1	10	Test Results 11: 186
12	6	1	10	5	1	10	Test Results 12: 570
13	6	1	30	5	1	10	Test Results 13: 764
14	6	1	70	5	1	10	Test Results 14: 821
15	6	1	100	5	1	10	Test Results 15: 861
16	6	1	10	5	1	30	Test Results 16: 743
17	6	1	30	5	1	30	Test Results 17: 971
18	6	1	70	5	1	30	Test Results 18: 993
19	6	1	100	5	1	30	Test Results 19: 997
20	6	1	10	5	1	70	Test Results 20: 838
21	6	1	30	5	1	70	Test Results 21: 997
22	6	1	70	5	1	70	Test Results 22: 1000
23	6	1	100	5	1	70	Test Results 23: 1000
24	6	1	10	5	1	100	Test Results 24: 830
25	6	1	30	5	1	100	Test Results 25: 996
26	6	1	70	5	1	100	Test Results 26: 1000
27	6	1	100	5	1	100	Test Results 27: 1000
28	6	1	30	5	1	100	Test Results 28: 999
29	6	4	30	5	1	100	Test Results 29: 899
30	6	9	30	5	1	100	Test Results 30: 721

Appendix IV: Code

```
knitr::opts_chunk$set(echo = TRUE)

#Use required packages
library(tidyverse) #for plots and data manipulation
library(cowplot) #aligning plots
library(gridExtra)
library(scales)

df_data <- read_csv("framingham_data.csv") # Read in data
df_data$index <- seq(nrow(df_data)) # Add an index column

#df_data %>% summary # Summarize Data

# Split data into smoker and nonsmoker
df_smoker <- df_data %>% filter(currentSmoker == 1)
df_nonsmoker <- df_data %>% filter(currentSmoker == 0)

#Create a sample variance function to ensure proper calculation
sample_variance <- function(x, sampling = TRUE){
    if (sampling == TRUE){
        sum((x - mean(x))^2) / (length(x) - 1)
    } else if(sampling == FALSE) {
        sum((x - mean(x))^2) / (length(x))
    }
}
#Create pooled sample variance function
f_pooled_variance <- function(x, y){
    ((length(x) - 1) * sample_variance(x) +
     (length(y) - 1) * sample_variance(y)) /
    (length(x) + length(y) - 2)
}

# Skewness function
skew_function <- function(x){
    mean((x - mean(x))^3) / sqrt(sample_variance(x))^3
}

# kurtosis function
kurt_function <- function(x){
    mean((x - mean(x))^4) / sqrt(sample_variance(x))^4
}

# Create a Satterthawaite Approximation Function

satterth <- function(s1, s2, n1, n2){
    term1 <- s1/n1
    term2 <- s2/n2
    nu <- (term1 + term2)^2 / ((term1/(n1 - 1)) + (term2/(n2 - 1)))
    return(floor(nu))
}


#Plot and compare split data

#options(repr.plot.width = 6, repr.plot.height = 4, repr.plot.res = 150)
```

```
plot_colors <- c("#001427","#708d81","#f4d58d","#bf0603","#8d0801")
y_limits <- c(0, 0.0225)

total_data <- ggplot(df_data) + geom_density(aes(sysBP),
                                              fill = plot_colors[1],
                                              alpha = 0.6) +
                       ylim(y_limits) + ylab("Density") + xlab("")

sep_data <- ggplot() + geom_density(data = df_smoker, aes(sysBP),
                                    fill = plot_colors[3], alpha = 0.6) +
                       geom_density(data = df_nonsmoker, aes(sysBP),
                                    fill = plot_colors[5], alpha = 0.6) +
                       ylim(y_limits) + ylab("") + xlab("Systolic Blood Pressure")

plot_3 <- ggplot() + geom_density(data = df_smoker, aes(sysBP),
                                    fill = plot_colors[3], alpha = 0.5) +
                       geom_density(data = df_data, aes(sysBP),
                                    fill = plot_colors[1], alpha = 0.5) +
                       geom_density(data = df_nonsmoker, aes(sysBP),
                                    fill = plot_colors[5], alpha = 0.5) +
                       ylim(y_limits) + ylab("") + xlab("")

#plot_grid(total_data, sep_data, plot_3, align = 'vh',
#          hjust = -1,nrow = 1, ncol = 3)

data_kurtosis <- kurt_function(df_data$sysBP)
data_skew <- skew_function(df_data$sysBP)
data_IQR <- as.numeric(quantile(df_data$sysBP, probs = 0.75)) -
       as.numeric(quantile(df_data$sysBP, probs = 0.25))
data_MAD <- median(abs(df_data$sysBP - median(df_data$sysBP)))
data_samVar <- sample_variance(df_data$sysBP)

eIQR <- data_IQR / 1.35
eMAD <- data_MAD / 0.675

# Q-Q Plot

data_qqplot <-
ggplot(df_data, aes(sample = sysBP)) +
stat_qq(shape = 1) + stat_qq_line() +
ggtitle("Normal Q-Q Plot for Blood Pressure Data") +
xlab("Theoretical Quantiles") +
ylab("Sample Quantiles")

# Common values for analysis

alpha <- 0.05

mu_smoker <- mean(df_smoker$sysBP)
var_smoker <- sample_variance(df_smoker$sysBP)
n_smoker <- length(df_smoker$sysBP)

mu_nonsmoker <- mean(df_nonsmoker$sysBP)
var_nonsmoker <- sample_variance(df_nonsmoker$sysBP)
n_nonsmoker <- length(df_nonsmoker$sysBP)

# Two Sample T-test - Pooled Sample Variance - P-value
```

```r
dof_1 <- (n_smoker + n_nonsmoker - 2)

p_sample_var_1 <- f_pooled_variance(df_smoker$sysBP,
                                    df_nonsmoker$sysBP)


t_obs_1 <- (mu_smoker - mu_nonsmoker) / (sqrt(p_sample_var_1) * sqrt(1/n_smoker + 1/n_nonsmoker))

t_stat_1 <- qt(alpha / 2, dof_1)

p_value_obs_1 <- dt(t_obs_1, dof_1)

#Two Sample T-test - Difference Variance Sample Variance - P-value

dof_2 <- satterth(var_smoker, var_nonsmoker, n_smoker, n_nonsmoker)

np_sample_var_2 <- (var_nonsmoker/n_smoker + var_nonsmoker/n_nonsmoker)

t_obs_2 <- (mu_smoker - mu_nonsmoker) / (sqrt(var_nonsmoker/n_smoker + var_nonsmoker/n_nonsmoker))

t_stat_2 <- qt(alpha / 2, dof_2)

p_value_obs_2 <- dt(t_obs_2, dof_2)

# Confidence Limits

diff_mu <- mu_smoker - mu_nonsmoker

#Pooled Sample variance

CL_pooled <- t_stat_1 * (sqrt(p_sample_var_1/n_smoker + p_sample_var_1/n_nonsmoker))

#Non pooled Sample variance

CL_nonpooled <- t_stat_2 * (sqrt(var_nonsmoker/n_smoker + var_nonsmoker/n_nonsmoker))

CI_pooled <- round(c(diff_mu + CL_pooled, diff_mu - CL_pooled), 2)

CI_nonpooled <-round(c(diff_mu + CL_nonpooled, diff_mu - CL_nonpooled), 2)

options(repr.plot.width = 12, repr.plot.height = 5, repr.plot.res = 150)
set.seed(100)

null_mean <- 3
alt_means <- c(0, 1, 2, 3, 4, 5)
plot_list <- list()

#plot_colors <- c("#072ac8","#1e96fc","#a2d6f9","#fcf300","#ffc600")

for(i in 1:6){

    sim1 <- rnorm(5000, null_mean, sqrt(1))
    sim2 <- rnorm(5000, alt_means[i], sqrt(1))

    alpha1 <- qnorm(0.025, null_mean, sqrt(1))
    alpha2 <- qnorm(0.975, null_mean, sqrt(1))
```

```r
    df_set <- tibble("H0" = sim1, "HA" = sim2)

    title_string <- sprintf("Difference in Means %i", (alt_means[i] - null_mean))

    plot_list[[i]]  <-
    ggplot(data = df_set) + geom_density(aes(H0), alpha = 0.5, fill = plot_colors[5]) +
                            geom_area(
                                aes(x = stage(H0, after_scale = oob_censor(x, c(-Inf, alpha1)
                                                                              )
                                                 )
                                      ),
                                stat = "density", fill = plot_colors[1]
                                 ) +
                            geom_area(
                                aes(x = stage(H0, after_scale = oob_censor(x, c(alpha2, Inf)
                                                                              )
                                                 )
                                      ),
                                stat = "density", fill = plot_colors[1]
                                     ) +
                            geom_density(aes(HA), alpha = 0.5) +
                            geom_area(
                                aes(x = stage(HA, after_scale = oob_censor(x, c(alpha1, alpha2)
                                                                              )
                                                 )
                                      ),
                                stat = "density", fill = plot_colors[2], alpha = 0.5
                                     ) +
                            xlim(-2, 8) + xlab("") + ylab("") + ggtitle(title_string)

                                }

do.call(grid.arrange, plot_list)
#Part II
set.seed(1)

alpha <- 0.05

test_function <- function (x, y, pooled = FALSE){

    mu_1 <- mean(x)
    var_1 <- sample_variance(x, sampling = TRUE)

    mu_2 <- mean(y)
    var_2 <- sample_variance(y, sampling = TRUE)

    #Calculate the pooled sample variance
    pooled_sample <- ((length(x) - 1) * var_1 + (length(y) - 1) * var_2) / (length(x) + length(y) - 2)

    #calulate the observed t statistic
    if (pooled == TRUE){

        cal_sigma <- (sqrt(pooled_sample/length(x) + pooled_sample/length(y)))

        ttest <- (mu_1 - mu_2) / cal_sigma
        dof <- length(x) + length(y) - 2 #Determine degrees of freedom

        } else {
```

```r
        cal_sigma <- (sqrt(var_1/length(x) + var_2/length(y)))

        ttest <- (mu_1 - mu_2) / cal_sigma
        dof <- satterth(var_1, var_2, length(x), length(y))
    }

    #Determine whether or not the null hypothesis can be rejected (1 = rejected, 0 = not rejected)
    verdict <- !between(ttest, qt(alpha / 2, dof), qt(1 - alpha / 2, dof))

    #Return calculated values
    return(c(mu_1, var_1, mu_2, var_2, ttest, cal_sigma, dof,verdict))
}



mu1 <- c(0, 4, 5, 6, 10)
var1 <- c(1, 4, 9)
n1 <- c(10, 30, 70, 100)

mu2 <- 5
var2 <- 1
n2 <- c(10, 30, 70, 100)

sim_test <- function(x_mu, x_var, x_n, y_mu, y_var, y_n, pooled = TRUE){

    sim_data_results <- matrix(rep(0, 8), ncol = 8)

    for (i in 1:1000){

        sim_set1 <- rnorm(x_n, x_mu, sqrt(x_var))
        sim_set2 <- rnorm(y_n, y_mu, sqrt(y_var))

        sim_data_results <- rbind(sim_data_results, test_function(sim_set1, sim_set2, pooled))

        #print(sim_data_results)
    }

    df_sim_data <- data.frame(sim_data_results[2 : nrow(sim_data_results),])
    colnames(df_sim_data) = c("Null Mean", "Null Variance", "Alternate Mean", "Alternate Variance",
                              "T statistic", "Calculated Variance", "DoF", "Null Reject")
    return(df_sim_data)
}

# HA: mean = 5, var = 1
test_results_1 <- sim_test(mu1[1], var1[1], n1[1], mu2, var2, n2[1])
test_results_2 <- sim_test(mu1[2], var1[1], n1[1], mu2, var2, n2[1])
test_results_3 <- sim_test(mu1[3], var1[1], n1[1], mu2, var2, n2[1])
test_results_4 <- sim_test(mu1[4], var1[1], n1[1], mu2, var2, n2[1])
test_results_5 <- sim_test(mu1[5], var1[1], n1[1], mu2, var2, n2[1])

test_results_6 <- sim_test(mu1[3], var1[1], n1[1], mu2, var2, n2[1])
test_results_7 <- sim_test(mu1[3], var1[2], n1[1], mu2, var2, n2[1])
test_results_8 <- sim_test(mu1[3], var1[3], n1[1], mu2, var2, n2[1])

test_results_9 <- sim_test(mu1[4], var1[1], n1[1], mu2, var2, n2[1])
test_results_10 <- sim_test(mu1[4], var1[2], n1[1], mu2, var2, n2[1])
test_results_11 <- sim_test(mu1[4], var1[3], n1[1], mu2, var2, n2[1])
```

```r
test_results_12 <- sim_test(mu1[4], var1[1], n1[1], mu2, var2, n2[1])
test_results_13 <- sim_test(mu1[4], var1[1], n1[2], mu2, var2, n2[1])
test_results_14 <- sim_test(mu1[4], var1[1], n1[3], mu2, var2, n2[1])
test_results_15 <- sim_test(mu1[4], var1[1], n1[4], mu2, var2, n2[1])

test_results_16 <- sim_test(mu1[4], var1[1], n1[1], mu2, var2, n2[2])
test_results_17 <- sim_test(mu1[4], var1[1], n1[2], mu2, var2, n2[2])
test_results_18 <- sim_test(mu1[4], var1[1], n1[3], mu2, var2, n2[2])
test_results_19 <- sim_test(mu1[4], var1[1], n1[4], mu2, var2, n2[2])

test_results_20 <- sim_test(mu1[4], var1[1], n1[1], mu2, var2, n2[3])
test_results_21 <- sim_test(mu1[4], var1[1], n1[2], mu2, var2, n2[3])
test_results_22 <- sim_test(mu1[4], var1[1], n1[3], mu2, var2, n2[3])
test_results_23 <- sim_test(mu1[4], var1[1], n1[4], mu2, var2, n2[3])

test_results_24 <- sim_test(mu1[4], var1[1], n1[1], mu2, var2, n2[4])
test_results_25 <- sim_test(mu1[4], var1[1], n1[2], mu2, var2, n2[4])
test_results_26 <- sim_test(mu1[4], var1[1], n1[3], mu2, var2, n2[4])
test_results_27 <- sim_test(mu1[4], var1[1], n1[4], mu2, var2, n2[4])

test_results_28 <- sim_test(mu1[4], var1[1], n1[2], mu2, var2, n2[4])
test_results_29 <- sim_test(mu1[4], var1[2], n1[2], mu2, var2, n2[4])
test_results_30 <- sim_test(mu1[4], var1[3], n1[2], mu2, var2, n2[4])
# Plotting
options(repr.plot.width = 6, repr.plot.height = 4, repr.plot.res = 150)

plot_grid(total_data, sep_data, plot_3, align = 'vh',
          hjust = -1,nrow = 1, ncol = 3, labels = c("A", "B", "C"))
data_qqplot + theme_bw()
```