HW2

Jinhong Du
15338039

Content

1 Problem

By stochastic simulation, compare the power of independent two-sample t-test and two-sample Wilcoxon Rank Sum Test in different data distributions.

Suppose that we have the following data

$$X_1, \cdots, X_{n_1} \sim F(x)$$
$$Y_1, \cdots, Y_{n_2} \sim G(x)$$

Hypothesis test:

$$H_0 : F(x) = G(x) \qquad\qquad H_a : F(x) = G(x + \Delta)$$

2 Independent two-sample t-test

If X, Y comes from independent normal distribution with equal varianes. For equal or unequal sample sizes, and equal variance, the t statistic is given by

$$t = \frac{\overline{X} - \overline{Y}}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

where

$$s = \sqrt{\frac{(n_1 - 1)s_X^2 + (n_2 - 1)s_Y^2}{n_1 + n_2 - 2}}$$

If X, Y are not normal distributed, when the sample size is big enough, X, Y will be approximated normal distributed and we can use this test.

3 Wilcoxon Rank Sum Test

After we discard the treatment labels and rank the observations, the statistic

$$\begin{aligned}
W &= \sum_{i=1}^{n_1} rank(X_i) \\
&= \sum_{i=1}^{n_1} \left(\sum_{j=1}^{n_1} \mathbb{1}_{\{X_j \le X_i\}} + \sum_{j=1}^{n_2} \mathbb{1}_{\{Y_j \le X_i\}} \right) \\
&= \sum_{i=1}^{n_1} \sum_{j=1}^{n_1} \mathbb{1}_{\{X_j \le X_i\}} + \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbb{1}_{\{Y_j \le X_i\}} \\
&= \frac{n_1(n_1 + 1)}{2} + \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \mathbb{1}_{\{Y_j \le X_i\}}
\end{aligned}$$

where $U = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \mathbb{1}_{\{Y_j \le X_i\}}$ is often called Mann-Whitney Statistic.

Since under H_0, W has an exact distribution with no simple expression. But we have

$$\mathbb{E}W \xlongequal{H_0} \frac{n_1(n_1+n_2+1)}{2}$$
$$VarW \xlongequal{H_0} \frac{n_1n_2(n_1+n_2+1)}{12}$$

For large $n = \min\{n_1, n_2\}$, $W \dot\sim N\left(\frac{n_1(n_1+n_2+1)}{2}, \frac{n_1n_2(n_1+n_2+1)}{12}\right)$.

There is a correction term necessary for ties

$$VarW_+ \xlongequal{H_0} \frac{n_1n_2(n_1+n_2+1)}{12} - \sum_{i=1}^{g} \frac{n_1n_2(t_i^3-t_i)}{12(n_1+n_2)(n_1+n_2+1)}$$

where t_i refers to the number of differences with the same absolute value in the ith tied group and g is the number of tied groups. If the number of ties is small (and especially if there are no large tie bands) ties can be ignored when doing calculations by hand.

4 Estimator of Power

Sample $\mathbf{X}$ and $\mathbf{Y}$ from two different distribution repectively and carry out the t-test(or Wilcoxon Rank Sum Test) for N times. Suppose that there are N_α tests that H_0 is rejected, then the power of the t-test(or Wilcoxon Rank Sum Test) can be estimated by $\frac{N_\alpha}{N}$.

5 Simulation

5.1 Normalilty with unequal mean and equal unkonwn variance

The theoretic power of two-side t test is given by

$$Power = \mathbb{P}\left(|t_{2n-2}(\Delta)| \ge t_{2n-2,1-\frac{\alpha}{2}} \middle| \Delta = \frac{\mu_2-\mu_1}{s\sqrt{\frac{2}{n}}}\right)$$

We use the above method to estimate power of the data sampled from following distributions in Figure 1.

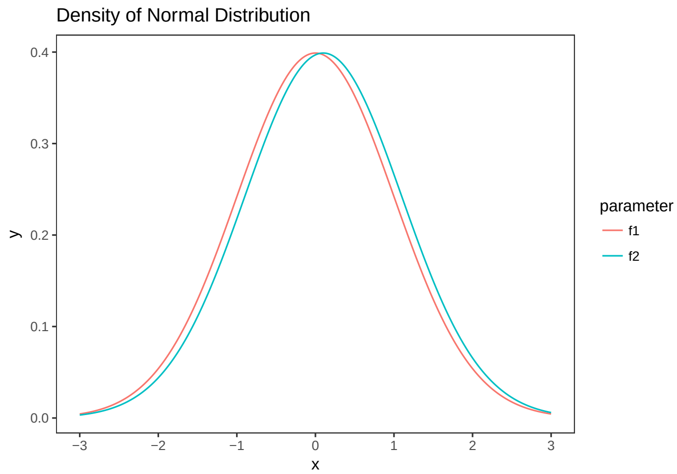

Figure 1: Normal Distribution

In this case, $X \sim N(0,1)$ and $Y - 0.1 \sim N(0,1)$. The result is as following,

n	Theoretic $Power_t$	$Power_t$	$Power_W$	$Power_t - Power_W$	$Power_W/Power_t$
10	0.0551613	0.0556	0.0584	-0.0028	1.0503597
25	0.0638669	0.0645	0.0615	0.003	0.9534884
50	0.078524	0.0791	0.0771	0.002	0.9747155
100	0.1083718	0.1073	0.1058	0.0015	0.9860205
200	0.1694809	0.174	0.1669	0.0071	0.9591954
5000	0.9988154	0.9989	0.999	-10×10^{-5}	1.0001001

Table 1: Normal Distribution

5.2 Non-Normal Distribution

We also use some non-normal distributions to simulate including

1. Symmetric distributions: Uniform distribution and Weibull distribution($scale = 1,\ shape = 5$).
2. Nonsymmetric distribution: Exponential distribution($rate = 1$) and Gamma Distribution($shape = 3,\ rate = 1$).

Then we shift their location for 0.1 and begin the simulations. The results of these four cases are given at Table 2, Table 3, Table 4 and Table 5.

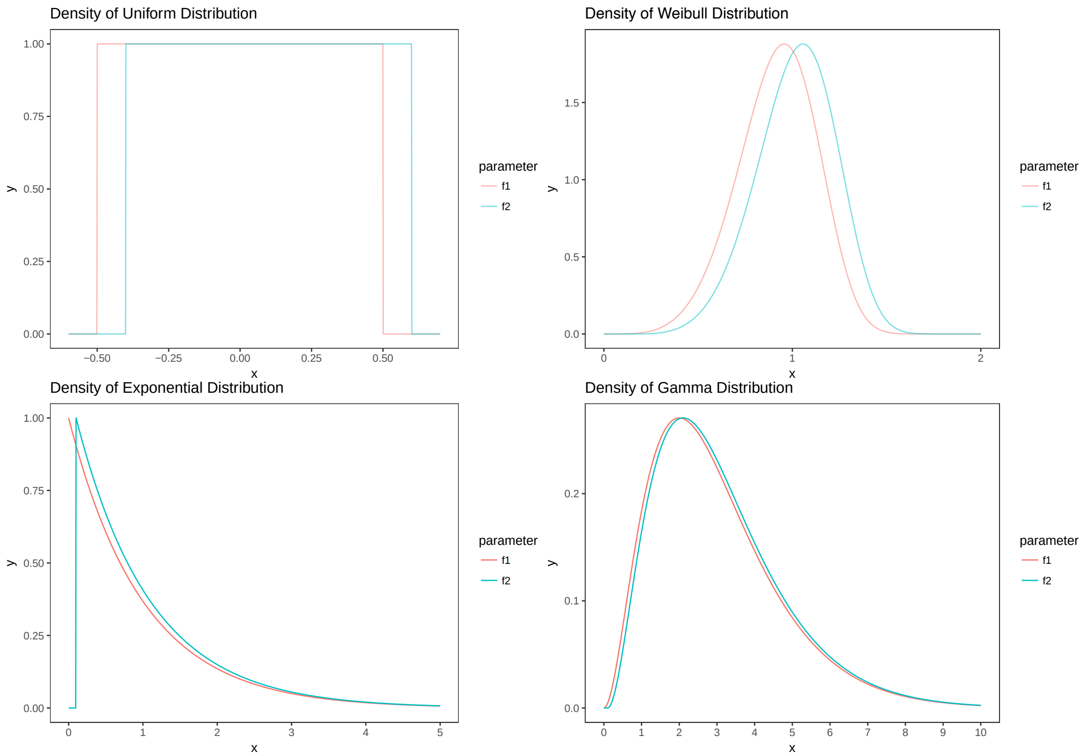

Figure 2: Non-Normal Distributions

n	$Power_t$	$Power_W$	$Power_t - Power_W$	$Power_W/Power_t$
10	0.107	0.1067	3×10^{-4}	0.9971963
25	0.2157	0.2028	0.0129	0.9401947
50	0.4042	0.3828	0.0214	0.9470559
100	0.6772	0.6419	0.0353	0.9478736
500	0.9305	0.9086	0.0219	0.9764643
5000	1	1	0	1

Table 2: Uniform Distribution

n	$Power_t$	$Power_W$	$Power_t - Power_W$	$Power_W/Power_t$
10	0.1704	0.1673	0.0031	0.9818075
25	0.379	0.3618	0.0172	0.9546174
50	0.6457	0.6207	0.025	0.9612823
100	0.9201	0.9056	0.0145	0.9842408
500	0.9972	0.9957	0.0015	0.9984958
5000	1	1	0	1

n	$Power_t$	$Power_W$	$Power_t - Power_W$	$Power_W / Power_t$

Table 3: Weibull Distribution

n	$Power_t$	$Power_W$	$Power_t - Power_W$	$Power_W / Power_t$
10	0.0545	0.0727	−0.0182	1.333945
25	0.0602	0.0815	−0.0213	1.3538206
50	0.0789	0.1291	−0.0502	1.6362484
100	0.1125	0.2101	−0.0976	1.8675556
200	0.1778	0.3791	−0.2013	2.132171
5000	0.999	1	−0.001	1.001001

Table 4: Expontinial Distribution

n	$Power_t$	$Power_W$	$Power_t - Power_W$	$Power_W / Power_t$
10	0.0489	0.0513	−0.0024	1.0490798
25	0.051	0.0549	−0.0039	1.0764706
50	0.0597	0.0606	-9×10^{-4}	1.0150754
100	0.0661	0.0723	−0.0062	1.0937973
200	0.0886	0.0988	−0.0102	1.1151242
5000	0.8169	0.8963	−0.0794	1.0971967

Table 5: Gamma Distribution

6 Conclusion

1. From Table 1 we know that under the normality with equal unkonwn variance, both $Power_t$ and $Power_W$ is close to the real power of t test. However, $Power_t$ is a little bigger than $Power_W$.

2. As the sample size n increases, both $Power_t$ and $Power_W$ will increase.

3. From Table 2 and Table 3, if the population distribution is symmetric, like uniform distribution, or approximated symmetric, then $Power_t$ is a little bigger than $Power_W$ when n is small and $Power_t \approx Power_W$ when n is large.

4. From Table 4 and Table 5, if the population distribution is nonsymmetric, like exponential distribution, then $Power_W$ may be larger than $Power_t$ when n is small and $Power_W$ is much bigger than $Power_W$ when n is large.

7 *R* code

7.1 Normal Distribution

```
library(ggplot2)
library(reshape2)

set.seed(0)
N <- 10000
n_list <- c(10,25,50,100,200,5000) # number in each group
alpha <- 0.05
mu1 <- 0
mu2 <- 0.1
sigma <- 1

x <- seq(-3, 3, length.out = 1000)
result <- data.frame(f1 = dnorm(x,mean = mu1),
                     f2 = dnorm(x,mean = mu2),
                     x = x)
print(ggplot(data = melt(result,id = 'x',variable.name="parameter") ,
       aes(x=x, y=value, colour=parameter)) +
    geom_line()+
    scale_x_continuous(breaks=seq(-3, 3, 1))+
    theme_bw() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
    labs(title="Density of Normal Distribution", x = 'x', y = 'y'))

Power_real_list <- c(1:6)
Power_real <- c(1:6)
for (i in 1:6) {
    n <- n_list[i]
    delta <- abs(mu1 - mu2)
    C <- qt(1 - alpha/2, 2 * n - 2)
    se <- sigma * sqrt( 1/n + 1/n )
    Power_real <- 1 - pt(C, 2*n-2, ncp=delta/se) + pt(-C, 2*n-2, ncp=delta/se)
    # Alternative function
    # Power_real <- power.t.test(n=n,delta = abs(mu1-mu2)/sigma)$power
    Power_real_list[i] <- Power_real
}
```

```
Power_t_list <- c(1:6)
Power_W_list <- c(1:6)
for (i in 1:6) {
    n <- n_list[i]
    Power_t <- 0
    Power_W <- 0
    for (i_ in c(1:N)) {
        X <- rnorm(n, mu1, sigma)
        Y <- rnorm(n, mu2, sigma)
        result_t <- t.test(X, Y, alt = "t", var.equal = T)
        result_W <- wilcox.test(X, Y, alt = "t", exact = F, corr = F)
        if (result_t$p.value < alpha) {
            Power_t <- Power_t + 1
        }
        if (result_W$p.value < alpha) {
            Power_W <- Power_W + 1
        }
    }
    Power_t <- Power_t / N
    Power_W <- Power_W / N
    Power_t_list[i] <-Power_t
    Power_W_list[i] <-Power_W
}
```

7.2 Uniform Distribution

```
set.seed(0)
N <- 10000
n_list <- c(10,25,50,100,200,5000) # number in each group
alpha <- 0.05
mu1 <- 0
mu2 <- 0.1

x <- seq(-1/2-0.1, 1/2+0.2, length.out = 1000)
result <- data.frame(f1 = ifelse(x<1/2 & x>-1/2,1,0),
                     f2 = ifelse(x>0.1-1/2 & x<1/2+0.1,1,0),
                     x = x)
print(ggplot(data = melt(result,id = 'x',variable.name="parameter") ,
       aes(x=x, y=value, colour=parameter)) +
```

```
    geom_line(alpha = 0.5)+
    theme_bw() +
    scale_x_continuous(breaks=seq(-1/2, 1/2, length.out = 5))+
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
    labs(title="Density of Uniform Distribution", x = 'x', y = 'y'))

Power_t_list <- c(1:6)
Power_W_list <- c(1:6)
for (i in 1:6) {
    n <- n_list[i]
    Power_t <- 0
    Power_W <- 0
    for (i_ in c(1:N)) {
        X <- runif(n, mu1-1/2, mu1 + 1/2)
        Y <- runif(n, mu2-1/2, mu2 + 1/2)
        result_t <- t.test(X, Y, alt = "t", var.equal = T)
        result_W <- wilcox.test(X, Y, alt = "t", exact = F, corr = F)
        if (result_t$p.value < alpha) {
            Power_t <- Power_t + 1
        }
        if (result_W$p.value < alpha) {
            Power_W <- Power_W + 1
        }
    }
    Power_t <- Power_t / N
    Power_W <- Power_W / N
    Power_t_list[i] <-Power_t
    Power_W_list[i] <-Power_W
}
```

7.3 Weibull Distribution

```
set.seed(0)
N <- 10000
n_list <- c(10,25,50,100,200,5000) # number in each group
alpha <- 0.05

x <- seq(0, 2, length.out = 1000)
result <- data.frame(f1 = dweibull(x,shape = 5,scale = 1),
```

```
                        f2 = dweibull(x-0.1,shape = 5,scale = 1),
                        x = x)
print(ggplot(data = melt(result,id = 'x',variable.name="parameter") ,
    aes(x=x, y=value, colour=parameter)) +
    geom_line(alpha = 0.5)+
    theme_bw() +
    scale_x_continuous(breaks = seq(0, 2, 1))+
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
    labs(title="Density of Logistic Distribution", x = 'x', y = 'y'))

Power_t_list <- c(1:6)
Power_W_list <- c(1:6)
for (i in 1:6) {
    n <- n_list[i]
    Power_t <- 0
    Power_W <- 0
    for (i_ in c(1:N)) {
        X <- rweibull(n,shape = 5,scale = 1)
        Y <- rweibull(n,shape = 5,scale = 1) + 0.1
        result_t <- t.test(X, Y, alt = "t", var.equal = T)
        result_W <- wilcox.test(X, Y, alt = "t", exact = F, corr = F)
        if (result_t$p.value < alpha) {
            Power_t <- Power_t + 1
        }
        if (result_W$p.value < alpha) {
            Power_W <- Power_W + 1
        }
    }
    Power_t <- Power_t / N
    Power_W <- Power_W / N
    Power_t_list[i] <-Power_t
    Power_W_list[i] <-Power_W
}
```

7.4 Exponential Distribution

```
set.seed(0)
N <- 10000
n_list <- c(10,25,50,100,200,5000) # number in each group
```

```
alpha <- 0.05
mu <- 1

x <- seq(0, 5, length.out = 1000)
result <- data.frame(f1 = dexp(x, 1/mu),
                     f2 = dexp(x-0.1, 1/mu),
                     x = x)
print(ggplot(data = melt(result,id = 'x',variable.name="parameter") ,
       aes(x=x, y=value, colour=parameter)) +
    geom_line()+
    scale_x_continuous(breaks=seq(0, 5, 1))+
    theme_bw() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
    labs(title="Density of Exponential Distribution", x = 'x', y = 'y'))

Power_t_list <- c(1:6)
Power_W_list <- c(1:6)
for (i in 1:6) {
    n <- n_list[i]
    Power_t <- 0
    Power_W <- 0
    for (i_ in c(1:N)) {
        X <- rexp(n, 1/mu)
        Y <- rexp(n, 1/mu) + 0.1
        result_t <- t.test(X, Y, alt = "t", var.equal = T)
        result_W <- wilcox.test(X, Y, alt = "t", exact = F, corr = F)
        if (result_t$p.value < alpha) {
            Power_t <- Power_t + 1
        }
        if (result_W$p.value < alpha) {
            Power_W <- Power_W + 1
        }
    }
    Power_t <- Power_t / N
    Power_W <- Power_W / N
    Power_t_list[i] <-Power_t
    Power_W_list[i] <-Power_W
}
```

7.5 Gamma Distribution

```
set.seed(0)
N <- 10000
n_list <- c(10,25,50,100,200,5000) # number in each group
alpha <- 0.05
shape <- 3
rate <- 1

x <- seq(0, 10, length.out = 1000)
result <- data.frame(f1 = dgamma(x, shape, rate),
                     f2 = dgamma(x-0.1, shape, rate),
                     x = x)
print(ggplot(data = melt(result,id = 'x',variable.name="parameter") ,
       aes(x=x, y=value, colour=parameter)) +
    geom_line()+
    scale_x_continuous(breaks=seq(0, 10, 1))+
    theme_bw() +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
    labs(title="Density of Gamma Distribution", x = 'x', y = 'y'))

Power_t_list <- c(1:6)
Power_W_list <- c(1:6)
for (i in 1:6) {
    n <- n_list[i]
    Power_t <- 0
    Power_W <- 0
    for (i_ in c(1:N)) {
        X <- rgamma(n, shape, rate)
        Y <- rgamma(n, shape, rate) + 0.1
        result_t <- t.test(X, Y, alt = "t", var.equal = T)
        result_W <- wilcox.test(X, Y, alt = "t", exact = F, corr = F)
        if (result_t$p.value < alpha) {
            Power_t <- Power_t + 1
        }
        if (result_W$p.value < alpha) {
            Power_W <- Power_W + 1
        }
    }
```

```
        Power_t <- Power_t / N
        Power_W <- Power_W / N
        Power_t_list[i] <-Power_t
        Power_W_list[i] <-Power_W
    }
```