Problem1

Yiming Gao (yimingg2)

2017/2/22

Problem 1: Two-Sample T-test vs Wilcoxon Ranked Sum Test

(a)

Type I error rate

$X \sim N(0,1)$, $Y \sim N(-\Delta, 1)$.

$$t = \frac{\bar{X} - \bar{Y}}{s_p\sqrt{2/n}}$$

where

$$s_p^2 = \frac{(n-1)s_x^2 + (n-1)s_y^2}{2n-2}.$$

We can either write our own function or use R function `t.test`, which give same results.

```r
# draw samples
nvals = c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
m = 10000
alpha = 0.05
type1 = matrix(0, length(nvals), 1)
decisions = numeric(m)

function_a1 <- function(nvals){

  for (i in 1: length(nvals)){
    n = nvals[i]

    for (j in 1:m){
      X = rnorm(n)
      Y = rnorm(n)
      # alternative: X has larger value than Y
      P.value = t.test(X, Y, alternative = "greater", var.equal = TRUE)$p.value
      decisions[j] = (P.value < alpha)
    }
  type1[i, 1] = mean(decisions)
    }
  return(type1)
}

set.seed(27)
result1 = as.data.frame(function_a1(nvals))
rownames(result1) = c("n = 10","n = 20","n = 30","n = 40","n = 50",
                      "n = 60","n = 70","n = 80","n = 90","n = 100")
colnames(result1) = "Type I error rate"
result1
```

```
##          Type I error rate
## n = 10             0.0484
## n = 20             0.0479
## n = 30             0.0498
## n = 40             0.0500
## n = 50             0.0489
## n = 60             0.0494
## n = 70             0.0464
## n = 80             0.0548
## n = 90             0.0511
## n = 100            0.0473
```

We can know that when $\Delta = 0$, sample size $n = 30$ is enough to control `Type 1 error rate`.

Power

```
alpha = 0.05
deltavals = c(0:9)/10
m = 100
nvals = c(10, 20, 30, 40, 50, 60, 70, 80, 90, 100)
powermatrix1 <- powermatrix2 <- powermatrix3 <- matrix(0, 10, 10)

function_a2 <- function(n, delta, alpha){
  decisions_t <- decisions_ex <- decisions_asy <- numeric(m)
  for (j in 1:m){
    X = rnorm(n, 0, 1)
    Y = rnorm(n, -delta, 1)
    # test decision is 1(reject) or 0
    P.value1 = t.test(X, Y, alternative = "greater", var.equal = TRUE)$p.value
    #P.value2 = wilcox.test(X, Y, alternative = "greater", exact = TRUE)$p.value
    P.value3 = wilcox.test(X, Y, alternative = "greater", exact = FALSE)$p.value
    decisions_t[j] = (P.value1 < alpha)
    #decisions_ex[j] = (P.value2 < alpha)
    decisions_asy[j] = (P.value3 < alpha)
  }
  c(mean(decisions_t), mean(decisions_ex), mean(decisions_asy))
}

set.seed(27)
for (i in 1:10){
  n = nvals[i]

  for (j in 1:10){
    delta = deltavals[j]
    powermatrix1[i,j] = function_a2(n, delta, alpha)[1]
    #powermatrix2[i,j] = function_a2(n, delta, alpha)
    powermatrix3[i,j] = function_a2(n, delta, alpha)[3]
  }
}

colnames(powermatrix3) = c("0","0.1","0.2","0.3","0.4","0.5","0.6","0.7","0.8","0.9")
rownames(powermatrix3) = c("n = 10","n = 20","n = 30","n = 40","n = 50",
                           "n = 60","n = 70","n = 80","n = 90","n = 100")
as.data.frame(powermatrix3)
```

```
##             0   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9
```

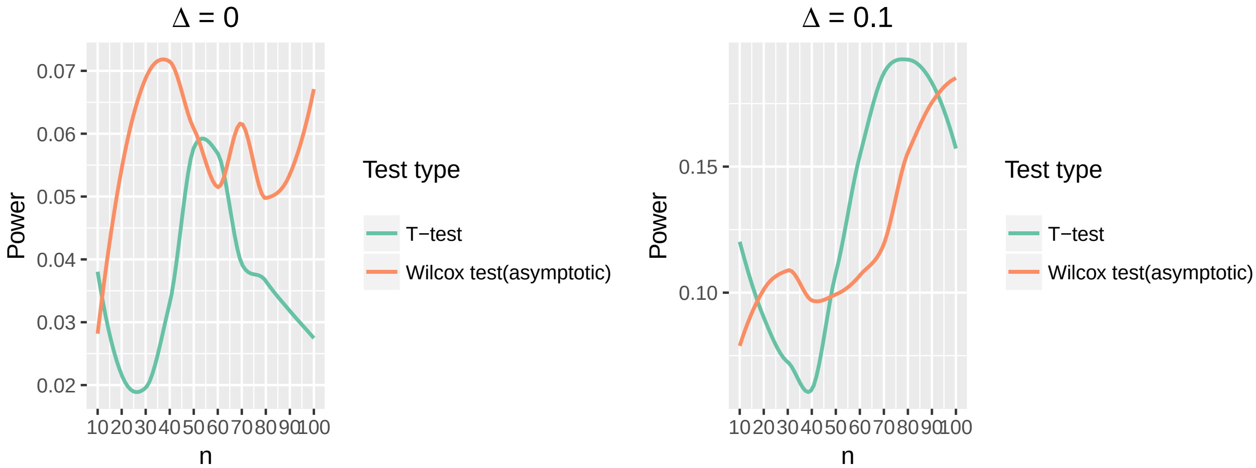
```
## n = 10  0.03 0.07 0.10 0.10 0.19 0.20 0.33 0.36 0.48 0.61
## n = 20  0.05 0.12 0.09 0.24 0.28 0.45 0.53 0.60 0.81 0.90
## n = 30  0.07 0.11 0.17 0.37 0.40 0.52 0.68 0.81 0.93 0.96
## n = 40  0.08 0.10 0.22 0.33 0.47 0.73 0.82 0.97 1.00 0.99
## n = 50  0.05 0.09 0.26 0.48 0.67 0.76 0.93 0.97 0.98 1.00
## n = 60  0.06 0.12 0.32 0.54 0.66 0.87 0.93 0.97 0.99 1.00
## n = 70  0.05 0.11 0.27 0.54 0.71 0.89 0.97 0.98 1.00 1.00
## n = 80  0.07 0.15 0.34 0.59 0.84 0.91 0.95 0.98 1.00 1.00
## n = 90  0.02 0.21 0.40 0.59 0.82 0.93 0.98 1.00 1.00 1.00
## n = 100 0.08 0.17 0.42 0.67 0.86 0.93 1.00 0.99 1.00 1.00
```

Suppose we want to have a sample size that guarantees the power is at least 0.8 if the true value of Δ is 0.5 or greater. It looks like the sample size needed to achieve our goal is somewhere between 50 and 60.

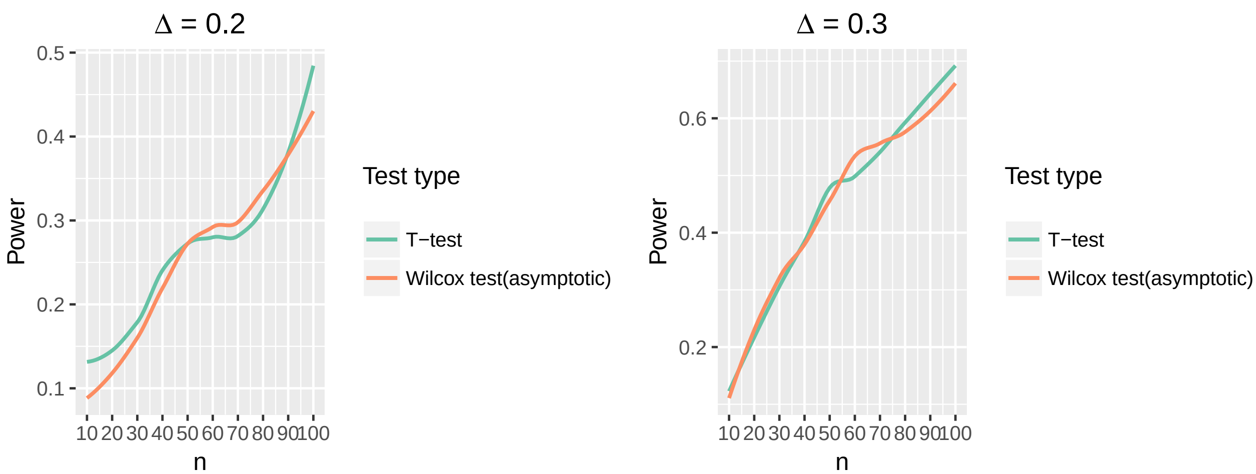
Then we create the power curves for different sample sizes from 20 to 100, with `step = 20`.

```r
data0 = data.frame("n" = nvals, "T" = powermatrix1[ ,1], "W_asy" = powermatrix3[, 1])
data1 = data.frame("n" = nvals, "T" = powermatrix1[ ,2], "W_asy" = powermatrix3[, 2])
data2 = data.frame("n" = nvals, "T" = powermatrix1[ ,3], "W_asy" = powermatrix3[, 3])
data3 = data.frame("n" = nvals, "T" = powermatrix1[ ,4], "W_asy" = powermatrix3[, 4])
data4 = data.frame("n" = nvals, "T" = powermatrix1[ ,5], "W_asy" = powermatrix3[, 5])
data5 = data.frame("n" = nvals, "T" = powermatrix1[ ,6], "W_asy" = powermatrix3[, 6])
data6 = data.frame("n" = nvals, "T" = powermatrix1[ ,7], "W_asy" = powermatrix3[, 7])
data7 = data.frame("n" = nvals, "T" = powermatrix1[ ,8], "W_asy" = powermatrix3[, 8])
data8 = data.frame("n" = nvals, "T" = powermatrix1[ ,9], "W_asy" = powermatrix3[, 9])
data9 = data.frame("n" = nvals, "T" = powermatrix1[ ,10], "W_asy" = powermatrix3[, 10])

library(ggplot2)
plot0 <- ggplot(data0, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot1 <- ggplot(data1, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.1")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot2 <- ggplot(data2, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.2")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot3 <- ggplot(data3, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.3")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")
```

```r
plot4 <- ggplot(data4, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.4")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot5 <- ggplot(data5, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.5")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot6 <- ggplot(data6, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.6")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot7 <- ggplot(data7, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.7")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot8 <- ggplot(data8, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.8")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")

plot9 <- ggplot(data9, aes(n)) + scale_x_continuous(breaks = seq(10, 100, 10), limits = c(10, 100)) +
  geom_smooth(aes(y = T, colour = "T-test"), se = FALSE, size = 0.8) +
  geom_smooth(aes(y = W_asy, colour = "Wilcox test(asymptotic)"), se = FALSE, size = 0.8) +
  labs(title = expression(paste(Delta," = 0.9")),
       x = "n", y = "Power", color = "Test type\n") +
  theme(plot.title = element_text(hjust = 0.5)) + scale_colour_brewer(palette = "Set2")


#multiplot(plot1,plot2,plot3, plot4, plot5, plot6, plot7, plot8, plot9,cols = 2)
library(gridExtra)
grid.arrange(plot0, plot1, ncol = 2)
```




```r
grid.arrange(plot2, plot3, ncol = 2)
```



- From the plot, we know that the power approaches to 1 **faster** and faster with **increasing sample size**.
- For a fixed sample size (one line in the graph), the larger Δ is, the greater power it has.