Theoretical foundations of the analysis of large data sets - report 4

Łukasz Rębisz

2023-12-16

Exercise 1

Let's consider a low dimensial setup $n = 20$ for:

$$X_1, ..., X_n \sim N(\mu_i, 1)$$

in the following situations:

a) $\mu_1 = 1.2\sqrt{2\log n}, \mu_2, ..., \mu_n = 0;$
b) $\mu_1 = ... = \mu_5 = 1.02\sqrt{2\log \frac{n}{10}}, \mu_6, ..., \mu_n = 0;$
c) $\mu_i = \sqrt{2\log \frac{20}{i}}, i = 1, ..., 10, \mu_{11}, ..., \mu_n = 0.$

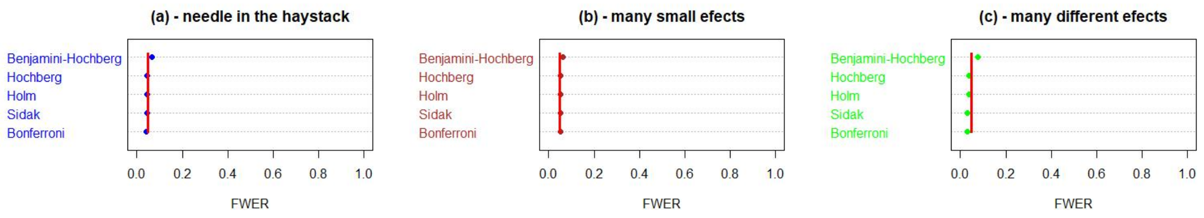
In each case we will compare **FWER, FDR** and **Power** of the following procedures:

- Bonferroni,
- Sidak's procedure with $\alpha_n = 1 - (1 - \alpha)^{1/n}$,
- Holm,
- Hochberg,
- Benjamini-Hochberg.

Implementation:

```
alpha <- 0.05

tests <- function(n, k, p_values){
    FWER <- rep(0, 5)
    FDP <- rep(0, 5)
    Power <- rep(0, 5)
    # Bonferroni
    BF_decision <- p_values <= alpha/n
    FWER[1] <- sum(BF_decision[(k+1):n])>=1
    FDP[1] <- sum(BF_decision[(k+1):n])/max(1, sum(BF_decision))
    Power[1] <- sum(BF_decision[1:k])/k

    # Sidak
    Sidak_decision <- p_values <= 1 - (1-alpha)^(1/n)
    FWER[2] <- sum(Sidak_decision[(k+1):n])>=1
    FDP[2] <- sum(Sidak_decision[(k+1):n])/max(1, sum(Sidak_decision))
    Power[2] <- sum(Sidak_decision[1:k])/k

    # Holm
    p_values_ordered <- sort(p_values, index.return=TRUE)
    p_values_ordered_val <- p_values_ordered$x
    p_values_ordered_indx <- p_values_ordered$ix
    i <- 1:n
    min_p_val_FALSE <- min(which((p_values_ordered_val > alpha/(n-i+1))))
    Holm_decision <- rep(0, n)
    if(min_p_val_FALSE >= 2){
    indx_rejection <- rep(0, n)
    indx_rejection[1:(min_p_val_FALSE-1)] <- 1
    Holm_decision[p_values_ordered_indx[indx_rejection == 1]] <- 1
    }
    FWER[3] <- sum(Holm_decision[(k+1):n])>=1
    FDP[3] <- sum(Holm_decision[(k+1):n])/max(1, sum(Holm_decision))
    Power[3] <- sum(Holm_decision[1:k])/k

    # Hochberg
    j <- n
    while(p_values_ordered_val[j] > alpha/(n-j+1)){
      j <- j - 1
      if(j <= 0) break
    }
    indx_rejection <- rep(0, n)
    if(j >= 1) indx_rejection[1:j] <- 1
    Hochberg_decision <- rep(0, n)
    Hochberg_decision[p_values_ordered_indx[indx_rejection == 1]] <- 1
    FWER[4] <- sum(Hochberg_decision[(k+1):n])>=1
    FDP[4] <- sum(Hochberg_decision[(k+1):n])/max(1, sum(Hochberg_decision))
    Power[4] <- sum(Hochberg_decision[1:k])/k

    # Benjamini - Hochberg
    i_0 <- max(which(p_values_ordered_val <= i*(alpha/n)))
    indx_rejection <- rep(0, n)
    if(i_0 >= 1) indx_rejection[1:i_0] <- 1
    BH_decision <- rep(0, n)
    BH_decision[p_values_ordered_indx[indx_rejection == 1]] <- 1
    FWER[5] <- sum(BH_decision[(k+1):n])>=1
    FDP[5] <- sum(BH_decision[(k+1):n])/max(1, sum(BH_decision))
    Power[5] <- sum(BH_decision[1:k])/k

    return(c(FWER, FDP, Power))
}

ex_1_20 <- function(){
  n <- 20

  # a
  vec_x <- c(rnorm(1, mean = 1.2*sqrt(2*log(n))), rnorm(n-1))
  p_values <- 2*(1-pnorm(abs(vec_x)))
  results_a <- tests(n, 1, p_values)

  # b
  vec_x <- c(rnorm(5, mean = 1.02*sqrt(2*log(n/10))), rnorm(n-5))
  p_values <- 2*(1-pnorm(abs(vec_x)))
  results_b <- tests(n, 5, p_values)

  # c
  vec_x <- c(rnorm(10, mean = sqrt(2*log(20/(1:10)))), rnorm(n-10))
  p_values <- 2*(1-pnorm(abs(vec_x)))
  results_c <- tests(n, 10, p_values)

  return(c(results_a, results_b, results_c))
}
```

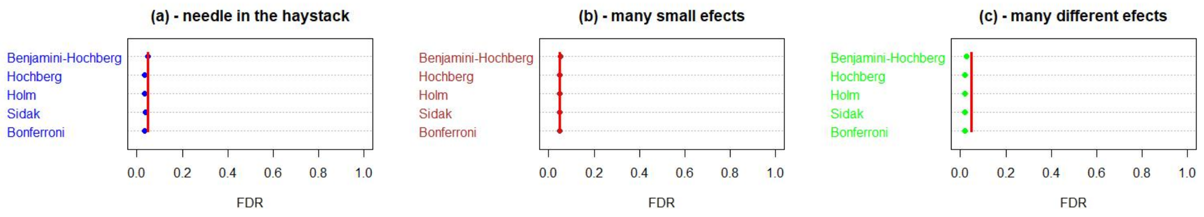
FWER, $n = 20$



The above results show us, that in a low dimensional setup:

- Benjamini-Hochberg does NOT control the Family Wise Error Rate FDER (the result is consistent with the theory),
- the other procedures control FWER at the significance level $\alpha = 0.05$ (red line) which is also consistent with the theory,
- the case (c) shows us the difference between controlling FWER by B-H procedures and the other procedures.

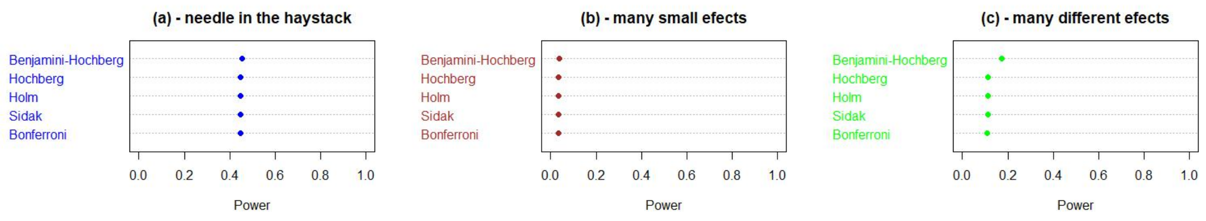
FDR, $n = 20$



We can show that:

- each of the investigating procedures controls False Discover Rate FDR at the significance rate $\alpha = 0.05$ (red line),
- this result is consistent with the theory, beacuse procedures which control FWER, control also FDR and the Benjamini-Hochberg procedure (which doesn't control FWER) is constructed to control FDR,
- the best results we get at the case (c), the worst in the case (b) - natural results, because when we have many small effects it is difficult to properly discover alternatives.

Power, $n = 20$



The above results of the power show us that:

- in case (a) each of the investigating tests has the power close to 0.5,
- in case (b) the power is very low (close to zero) for each procedure - many small effects are difficult to discover,
- case (c): the B-H procedure gets the best results, the power of the others procedures is lower; B-H doesn't control FWER so that procedure is less conservative.

Exercise 2

Let's repeat the above exercise for the large dimensional setup ($n = 5000$). In this case we will investigate the same procedures for the following situations:

a) $\mu_1 = 1.2\sqrt{2\log n}, \mu_2, ..., \mu_n = 0;$

b) $\mu_1 = ... = \mu_{100} = 1.02\sqrt{2\log\frac{n}{200}}, \mu_{101}, ..., \mu_n = 0;$

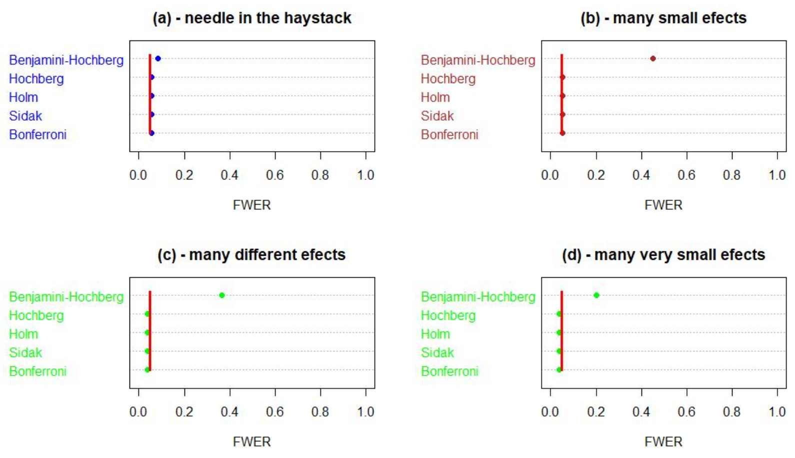
c) $\mu_1 = ... = \mu_{100} = \sqrt{2\log\frac{n}{200}}, \mu_{101}, ..., \mu_n = 0;$

d) $\mu_1 = ... = \mu_{1000} = 1.002\sqrt{2\log\frac{n}{2000}}, \mu_{1001}, ..., \mu_n = 0.$

Implementation:

```r
alpha <- 0.05

ex_1_5k <- function(){
  n <- 5000
  # a
  vec_x <- c(rnorm(1, mean = 1.2*sqrt(2*log(n))), rnorm(n-1))
  p_values <- 2*(1-pnorm(abs(vec_x)))
  results_a <- tests(n, 1, p_values)

  # b
  vec_x <- c(rnorm(100, mean = 1.02*sqrt(2*log(n/200))), rnorm(n-100))
  p_values <- 2*(1-pnorm(abs(vec_x)))
  results_b <- tests(n, 100, p_values)

  # c
  vec_x <- c(rnorm(100, mean = sqrt(2*log(n/200))), rnorm(n-100))
  p_values <- 2*(1-pnorm(abs(vec_x)))
  results_c <- tests(n, 100, p_values)

  # d
  vec_x <- c(rnorm(1000, mean = 1.002*sqrt(2*log(n/2000))), rnorm(n-1000))
  p_values <- 2*(1-pnorm(abs(vec_x)))
  results_d <- tests(n, 1000, p_values)
  return(c(results_a, results_b, results_c, results_d))
}
```

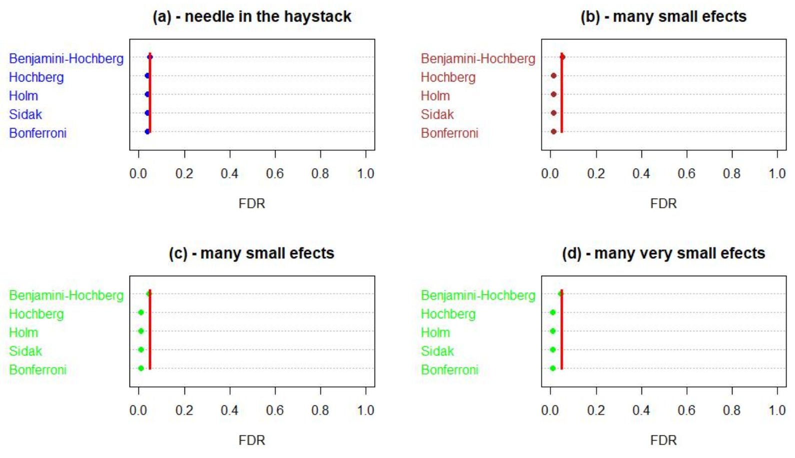
FWER, $n = 5000$



The above results show us, that in a large dimensional setup:

- Benjamini-Hochberg doesn't control FWER - the difference to the significance level $\alpha = 0.05$ is much greater then in the low dimensional setup, especially for the many small effects (b and c),
- the other procedures control FWER at the significance level $\alpha = 0.05$ (red line) which is consistent with the theory.

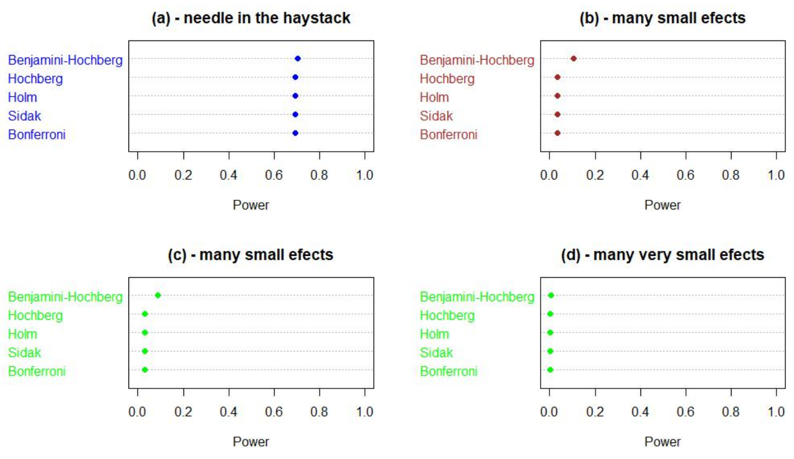
FDR, $n = 5000$



We can show that:

- each one investigating procedure controls FDR at the significance level $\alpha = 0.05$,
- procedures which control FWER are very conservative and they also control FDR (at level which is even smaller then α),
- Benjamini-Hochberg procedure constructed to control FDR at level α indeed gets FDR at the level very close to the value $\alpha = 0.05$.

Power, $n = 5000$



The above results show us that:

- case (a) - neddle in the haystack: each procedure gets the power which is close to 0.7 (for $n = 20$ the power was equal around 0.5 for each procedure),
- cases (b) and (c) - many small effects: procedures which control FWER are too conservative to get significance power which is close to zero; on the other hand, the less conservative Benjami-Hochberg procedure get better power which is close to 0.1,
- case (d) - many very small effects: effects are to small to be detected by anyone procedure: the power is close to zero.

Conclusion:

- Procedures which control FWER are very conservative. In the low dimensional setup it's not very bad: procedures aren't powerless, but in the large dimensional setup when we get many small effects those procedures are totally powerless.
- On the other hand, the Benjamini-Hochberg procedure which controls FDR, but doesn't control FWER is less conservative and so gets better power in each situation, especially in the large dimensional setup when we have many small effects.

Exercise 3

In this paragraph we will consider **two-step Fisher procedure** using:

- Bonferroni,
- chi-square test

for the first step in the following case $n \in \{20, 5000\}$ where we have:

$$X_1, ..., X_n, \quad X_i \sim N(\mu_i, 1)$$

for the following situations:

a) $\mu_1 = 1.2\sqrt{2\log n}, \mu_2, ..., \mu_n = 0;$

b) $\mu_1 = ... = \mu_5 = 1.02\sqrt{2\log \frac{n}{10}}, \mu_6, ..., \mu_n = 0;$

c) $\mu_i = \sqrt{2\log \frac{20}{i}}, i = 1, ..., 10, \mu_{11}, ..., \mu_n = 0;$

d) $\mu_1 = ... = \mu_{1000} = 1.002\sqrt{2\log\frac{n}{2000}}, \mu_{1001}, ..., \mu_n = 0$ (only for $n = 5000$).

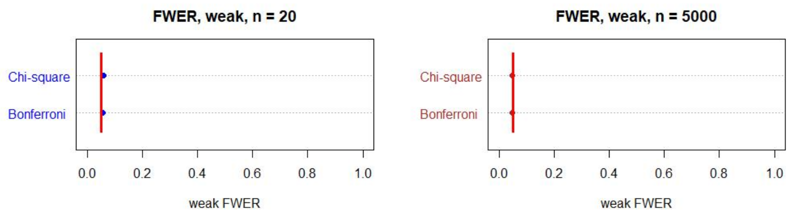
In each case we will compare FWER in the strong and weak sense, FDR and Power of the investigating tests.

Implementation:

```
M <- 1000
alpha <- 0.05

FWER_weak <- function(n){
  vec_x <- rnorm(n)
  p_values <- 2*(1-pnorm(abs(vec_x)))
  BF_decision_global <- min(p_values) <= alpha/n
  chi_square_decision_global <- sum(vec_x^2) > qchisq(1-alpha, df = n)

  FWER <- c(BF_decision_global, chi_square_decision_global)
  return(FWER)
}


FWER_strong <- function(n){
  # a
  FWER_a <- c(0, 0)
  FDR_a <- c(0, 0)
  Power_a <- c(0, 0)
  vec_x <- c(rnorm(1, mean = 1.2*sqrt(2*log(n))), rnorm(n-1))
  p_values <- 2*(1-pnorm(abs(vec_x)))

  BF_decision_global <- min(p_values) <= alpha/n
  if(BF_decision_global == 1){
    BF_decision <- p_values <= alpha
    FWER_a[1] <- sum(BF_decision[2:n])>=1
    FDR_a[1] <- sum(BF_decision[2:n])/max(1, sum(BF_decision))
    Power_a[1] <- sum(BF_decision[1])/1
  }

  chi_square_decision_global <- sum(vec_x^2) > qchisq(1-alpha, df = n)
  if(chi_square_decision_global == 1){
    chi_square_decision <- vec_x^2 > qchisq(1-alpha, df = 1)
    FWER_a[2] <- sum(chi_square_decision[2:n])>=1
    FDR_a[2] <- sum(chi_square_decision[2:n])/max(1, sum(chi_square_decision))
    Power_a[2] <- sum(chi_square_decision[1])/1
  }

  # b
  FWER_b <- c(0, 0)
  FDR_b <- c(0, 0)
  Power_b <- c(0, 0)
  vec_x <- c(rnorm(5, mean = 1.02*sqrt(2*log(n/10))), rnorm(n-5))
  p_values <- 2*(1-pnorm(abs(vec_x)))

  BF_decision_global <- min(p_values) <= alpha/n
  if(BF_decision_global == 1){
    BF_decision <- p_values <= alpha
    FWER_b[1] <- sum(BF_decision[6:n])>=1
    FDR_b[1] <- sum(BF_decision[6:n])/max(1, sum(BF_decision))
    Power_b[1] <- sum(BF_decision[1:5])/5
  }

  chi_square_decision_global <- sum(vec_x^2) > qchisq(1-alpha, df = n)
  if(chi_square_decision_global == 1){
    chi_square_decision <- vec_x^2 > qchisq(1-alpha, df = 1)
    FWER_b[2] <- sum(chi_square_decision[6:n])>=1
    FDR_b[2] <- sum(chi_square_decision[6:n])/max(1, sum(chi_square_decision))
    Power_b[2] <- sum(chi_square_decision[1:5])/5
  }
  # c
  FWER_c <- c(0, 0)
  FDR_c <- c(0, 0)
  Power_c <- c(0, 0)
  vec_x <- c(rnorm(10, mean = sqrt(2*log(20/(1:10)))), rnorm(n-10))
  p_values <- 2*(1-pnorm(abs(vec_x)))

  BF_decision_global <- min(p_values) <= alpha/n
  if(BF_decision_global == 1){
    BF_decision <- p_values <= alpha
    FWER_c[1] <- sum(BF_decision[11:n])>=1
    FDR_c[1] <- sum(BF_decision[11:n])/max(1, sum(BF_decision))
    Power_c[1] <- sum(BF_decision[1:10])/10
  }

  chi_square_decision_global <- sum(vec_x^2) > qchisq(1-alpha, df = n)
  if(chi_square_decision_global == 1){
    chi_square_decision <- vec_x^2 > qchisq(1-alpha, df = 1)
    FWER_c[2] <- sum(chi_square_decision[11:n])>=1
    FDR_c[2] <- sum(chi_square_decision[11:n])/max(1, sum(chi_square_decision))
    Power_c[2] <- sum(chi_square_decision[1:10])/10
  }

  # d
  if(n==5000){
      FWER_d <- c(0, 0)
      FDR_d <- c(0, 0)
      Power_d <- c(0, 0)
      vec_x <- c(rnorm(1000, mean = 1.002*sqrt(2*log(n/2000))), rnorm(n-1000))
      p_values <- 2*(1-pnorm(abs(vec_x)))

      BF_decision_global <- min(p_values) <= alpha/n
      if(BF_decision_global == 1){
        BF_decision <- p_values <= alpha
        FWER_d[1] <- sum(BF_decision[1001:n])>=1
        FDR_d[1] <- sum(BF_decision[1001:n])/max(1, sum(BF_decision))
        Power_d[1] <- sum(BF_decision[1:1000])/1000
      }

      chi_square_decision_global <- sum(vec_x^2) > qchisq(1-alpha, df = n)
      if(chi_square_decision_global == 1){
        chi_square_decision <- vec_x^2 > qchisq(1-alpha, df = 1)
```

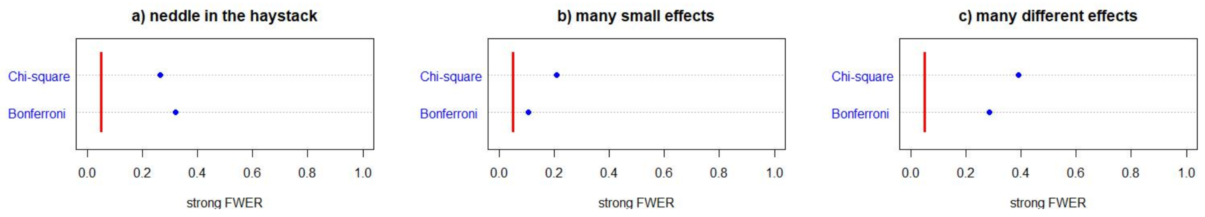
```
        FWER_d[2] <- sum(chi_square_decision[1001:n])>=1
        FDR_d[2] <- sum(chi_square_decision[1001:n])/max(1, sum(chi_square_decision))
        Power_d[2] <- sum(chi_square_decision[1:1000])/1000
      }
      return(c(FWER_a, FWER_b, FWER_c, FWER_d,
               FDR_a, FDR_b, FDR_c, FDR_d,
               Power_a, Power_b, Power_c, Power_d))
  }


  return(c(FWER_a, FWER_b, FWER_c,
           FDR_a, FDR_b, FDR_c,
           Power_a, Power_b, Power_c))
}
```

FWER in the weak sense



We can see that for the both cases both procedures control FWER in the weak sense (for the global null case) at the significance level $\alpha = 0.05$ (red line).

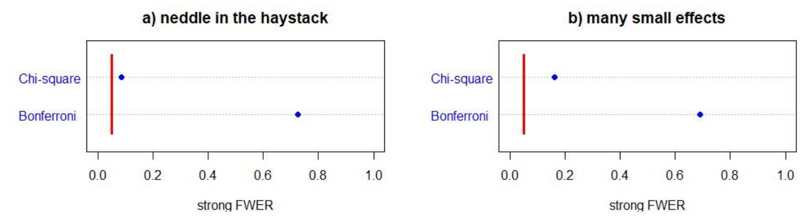
FWER in the strong sense, n = 20

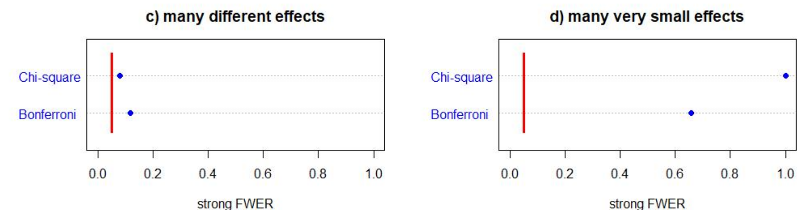


The above results show us that:

- in the low dimensional setup the both investigating procedures do NOT control FWER in the strong sense.
- The Bonferroni method is close to control strong FWER for the case (b) - many small effects.

FWER in the strong sense, n = 5000





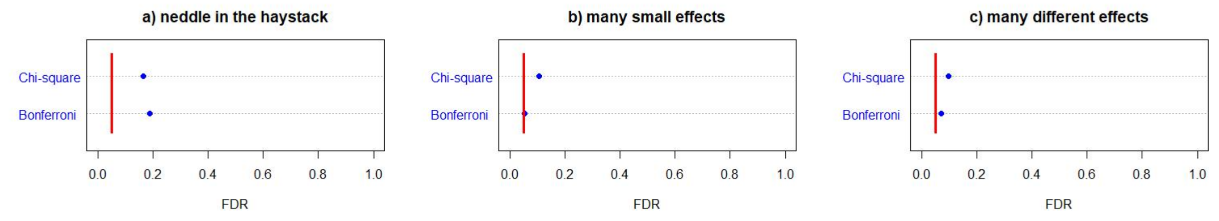
The above results show us that:

- in the large dimensional setup both investigating procedures also do NOT control FWER in the strong sense.
- The method with chi-square test is close to control strong FWER for cases a, b, c.
- The worst results we get in the case (d) - many very small effects are difficult to be corectly discovered.

Conclusion:

the two-step Fisher procedure controls FWER in the weak sense (in the global null case) and does NOT control FWER in the strong sense in any case. This result is consistent with the theory.

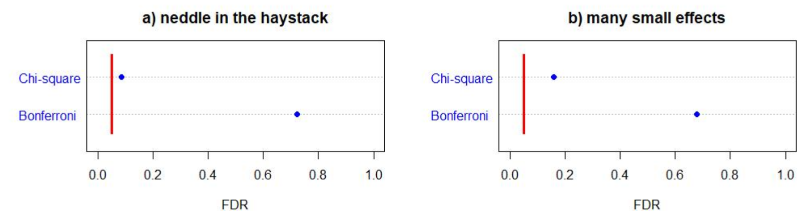
FDR, n = 20

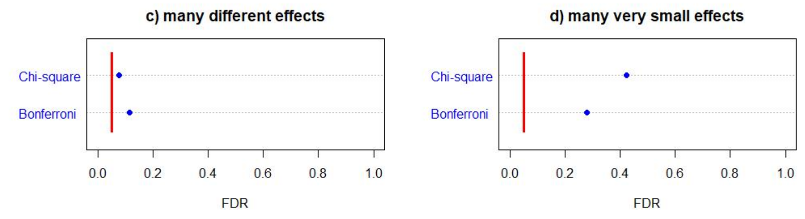


We can see that:

- only in the case (b) and using the Bonferroni test we have two-step Fisher test which controls FDR at the significance level α,
- generally: two-step Fisher test doesn't control FDR in the low dimensional setup.

FDR, n = 5000





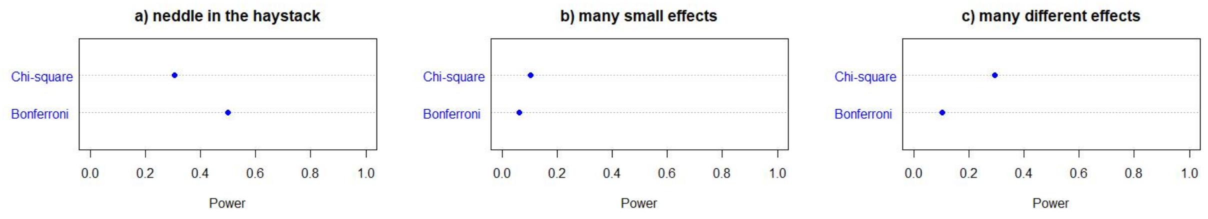
We can see that in the large dimensional setup:

- only in the case (a) and (c) and using the chi-square test we have two-step Fisher test which is close to control FDR at the significance level α,
- generally: two-step Fisher test doesn't control FDR in the large dimensional setup.

Conclusion:

Two-step Fisher test doesn't control False Discover Rate FDR.

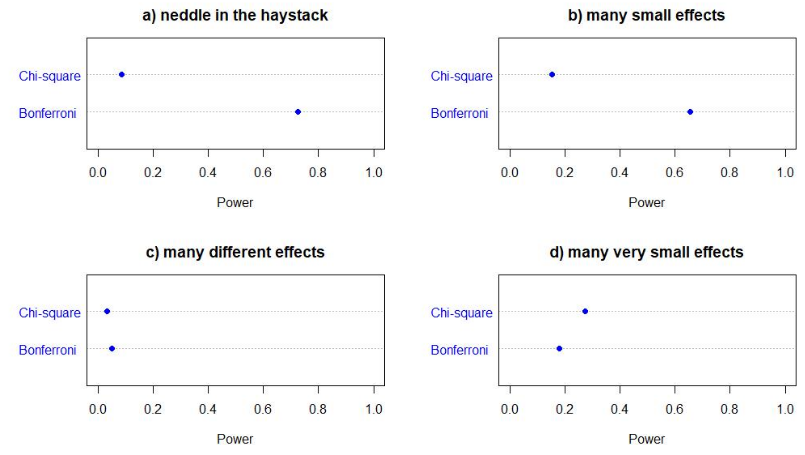
Power, n = 20



We can see that the power in the low dimensional setup for the two-step Fisher test:

- gets positive values: the test isn't powerless,
- Bonferroni method is better in the case (a) - neddle in the haystack (natural results),
- on the other hand: chi-square test is better for many small and many different effects.

Power, n = 5000



The results for the large dimensional setup are very similiar to the small n - case:

- two-step Fisher test isn't powerless,
- Bonferroni method is better for the neddle in the haystack problem and many different effects,
- for many small and very small effects the chi-square test give us the better power.

Exercise 4

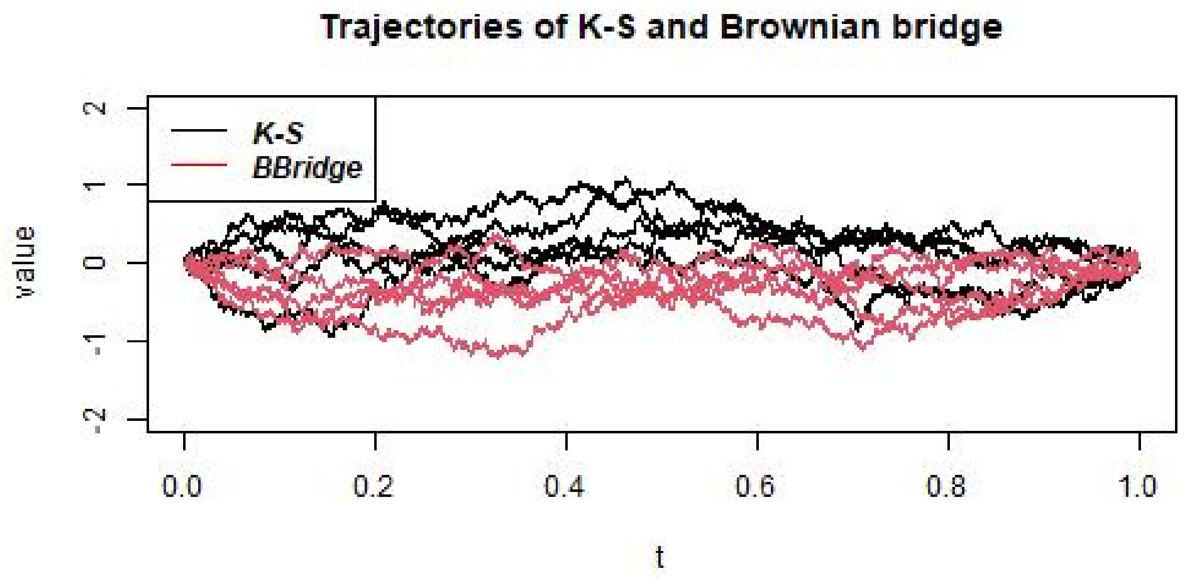
In this paragraph we will simulate (for $n = 5000$) 1000 trajectories of the empirical process (Kolmogorov-Smirnov):

$$U_n(t) = \sqrt{n}(F_n(t) - t), \quad t \in [0, 1]$$

and 1000 trajectories of the Brownian bridge $B(t), t \in [0, 1]$.

```r
n <- 5000
k <- 10000
vec_t <- seq(0, 1, 1/k)

U_function <- function(n){
  p <- runif(n)
  U <- sqrt(n)*(ecdf(p)(vec_t) - vec_t)
  return(U)
}

vec_U <- replicate(10000, U_function(n))
vec_B <- replicate(10000, BBridge(N=k))
```

For the above results we will plot 5 trajectories for each of these processes on the same graph.



The above plot shows us that trajectories for K-S empirical process and for the Brownian Bridge process are quite similiar.

The bove types of trajectories start and finish at the value 0 and get similiar values beetwen those points.

Now we will estimate the α quantile for (absolute) K-S statistics under the null hypothesis as well as α quantile of $T = \sup_{t\in[0,1]} |B(t)|$ for $\alpha \in \{0.8, 0.9, 0.95\}$ based on the above 1000 simulations.

Table 1: Quantiles

	0.8	0.9	0.95
K-S	1.060660	1.216224	1.343503
B	1.063834	1.217728	1.351726

We can see that for each investigating level α the corresponding quantiles for K-S statistics and Brownian bridge (absolute value in both cases) have very similiar values.

The above results (comparison of trajectories and quantiles) showed us empirically the following theoretical fact (under null hypothesis):

K-S statistics has asymptotically Kolmogorov distribution - the distribution of the random variable:

$$K = \sup_{t \in [0,1]} |B(t)|,$$

where $B(t)$ is the Brownian bridge.