Exam 1

Xinyu Zhang

Instructions

- Due date: **Tuesday, November 13**.
- This is a open resource exam, but you are not allowed to ask post exam questions online.
- You are not allowed to collaborate with classmates and/or people outside of class.
- Please circle or highlight your final answer.
- The toal possible point is 80.

Violation of this agreement will result in an **F** on this exam and it will be averaged in as a 0%.

---

1. Use the complete *WHAS100* dataset and gender as the group indicator, compute the log-rank statistic, Q, presented as Equation (9) on page 60 of note 2 with $\omega_i = 1$. Use Q to compute a p-value to test the null hypothesis of $H_o : S_0(t) = S_1(t)$. Do this **without a software package** (6 pts), and verify the calculation with `survdiff` (4 pts).

```
> km <- survfit(Surv(lenfol, fstat) ~ 1, data = whas100)
> km.gp1=with(whas100 %>% filter(gender==1),survfit(Surv(lenfol, fstat) ~ 1))
> km.gp0=with(whas100 %>% filter(gender==0),survfit(Surv(lenfol, fstat) ~ 1))
>
> km.df=data.frame(time=km$time,n.risk=km$n.risk)
> km.df.gp1=data.frame(time=km.gp1$time,d1.event=km.gp1$n.event,c1.censor=km.gp1$n.censor)
> km.df.gp0=data.frame(time=km.gp0$time,d0.event=km.gp0$n.event,c0.censor=km.gp0$n.censor)
>
> km.gender=left_join(km.df,km.df.gp1,by="time") %>%
+   mutate(d1.event=replace_na(d1.event,0),c1.censor=replace_na(c1.censor,0)) %>%
+   mutate(n1.risk=35-head(c(0,cumsum(d1.event+c1.censor)),-1),s1=cumprod(1-d1.event/n1.risk)) %>%
+   left_join(km.df.gp0,by="time") %>%
+   mutate(d0.event=replace_na(d0.event,0),c0.censor=replace_na(c0.censor,0)) %>%
+   mutate(n0.risk=65-head(c(0,cumsum(d0.event+c0.censor)),-1),s0=cumprod(1-d0.event/n0.risk)) %>%
+   mutate(s1=replace_na(s1,0),s0=replace_na(s0,0))
>
> head(km.gender)
```

```
  time n.risk d1.event c1.censor n1.risk        s1 d0.event c0.censor
1    6    100        0         0      35 1.0000000        2         0
2   14     98        1         0      35 0.9714286        0         0
3   44     97        0         0      34 0.9714286        1         0
4   62     96        1         0      34 0.9428571        0         0
5   89     95        1         0      33 0.9142857        0         0
6   98     94        1         0      32 0.8857143        0         0
  n0.risk        s0
1      65 0.9692308
2      63 0.9692308
3      63 0.9538462
4      62 0.9538462
5      62 0.9538462
6      62 0.9538462
```

We put the Kaplan-Meier survival function of group 1(gender==1) and group 0(gender==0) in the same table.

```
> km.gender %>%
+   mutate(e.d=n1.risk*(d1.event+d0.event)/n.risk,
+     var.d=(n1.risk*n0.risk*(d1.event+d0.event)*(n.risk-d1.event-d0.event))/n.risk^2/(n.risk-1)) %>%
+   summarise(q.numerator=(sum(d1.event-e.d))^2, q.denominator=sum(var.d[!is.na(var.d)])) %>%
+   mutate(q=q.numerator/q.denominator)
```

```
  q.numerator q.denominator        q
1     43.8244      11.03506 3.971377
```

```
> survdiff(Surv(lenfol, fstat) ~ gender, data = whas100)
```

```
Call:
survdiff(formula = Surv(lenfol, fstat) ~ gender, data = whas100)

          N Observed Expected (O-E)^2/E (O-E)^2/V
gender=0 65       28     34.6      1.27      3.97
gender=1 35       23     16.4      2.68      3.97

 Chisq= 4  on 1 degrees of freedom, p= 0.0463
```

We see that the q calculated by both methods are the same(3.9714).

2. There are many ways to form a basis for survival curve comparison. Here are some:

   a. The numerator term in Q without the square:

$$D_1 = \sum_{i=1}^{D} \{d_{1i} - E(d_{1i})\}.$$

   b. The largest distance between the two curves:

$$D_2 = \max |S_1(t) - S_0(t)|.$$

   c. The difference between the median survival times:

$$D_3 = S_1^{-1}(0.5) - S_0^{-1}(0.5)$$

   d. The difference between the mean survival times:

$$D_4 = \int_0^{t_{(n)}} \{S_1(u) - S_0(u)\}\, du,$$

   where $t_{(n)}$ is the maximum observed survival time.

Compute each of the above statistic for the $WHAS100$ dataset (5 pts $\times 4$).

```
> #d1
> with(km.gender,sum(d1.event-n1.risk*(d1.event+d0.event)/n.risk))
```

```
[1] 6.62
```

```
> #d2
> with(km.gender,max(abs(s1-s0)))
```

```
[1] 0.3853606
```

```
> #d3
> with(km.gender,min(time[s1<=0.5])-min(time[s0<=0.5]))
```

```
[1] -818
```

```
> #d4
> with(km.gender,sum(head(c(1,s1),-1)*(km.gender$time-head(c(0,km.gender$time),-1))-
+                          head(c(1,s0),-1)*(km.gender$time-head(c(0,km.gender$time),-1))))
```

```
[1] -432.2089
```

D1, D2, D3 and D4 are 6.26, 0.3854, -818 and -432.2089 respectively.

3. The statistics computed in (2) do not provide meaningful interpretations when standing along. We will use a permutation approach to test for the null hypothesis of $H_o : S_0(t) = S_1(t)$ based on these statistics. The idea of a permutation test is simple. The general procedure can be summarized into the following steps:

   i. Compute the desired statistic based on the observed data; we will call this the observed statistic.
   ii. Permute the data under the null.
   iii. Compute the statistics for each possible permutation in Step ii.; we will call these permutation statistics.
   iv. Draw conclusion based on where the observed statistic stands among the permutation statistics.

   The statistics we computed in (2) are the observed statistics in Step i. If the null hypothesis of $H_o : S_0(t) = S_1(t)$ is true, then one can randomly shuffle the group indicator to generate different permutations (Step ii) and the statistics for these permutations should be similar (Step iii).

   a. (5 pts×4) Generate 5000 permutation and, for each of the permutation, compute the four statistics presented in (2). We will call the permutated statistics D^*_{1i}, D^*_{2i}, D^*_{3i} and D^*_{4i} for $i = 1, \ldots, 5000$. Create a histgram for these permutated statistics and print the `summary`.

   b. (5 pts×4) Compute the p-value based on these statistics by

$$p = 2 \cdot \frac{\min(N_1, N_2)}{5000},$$

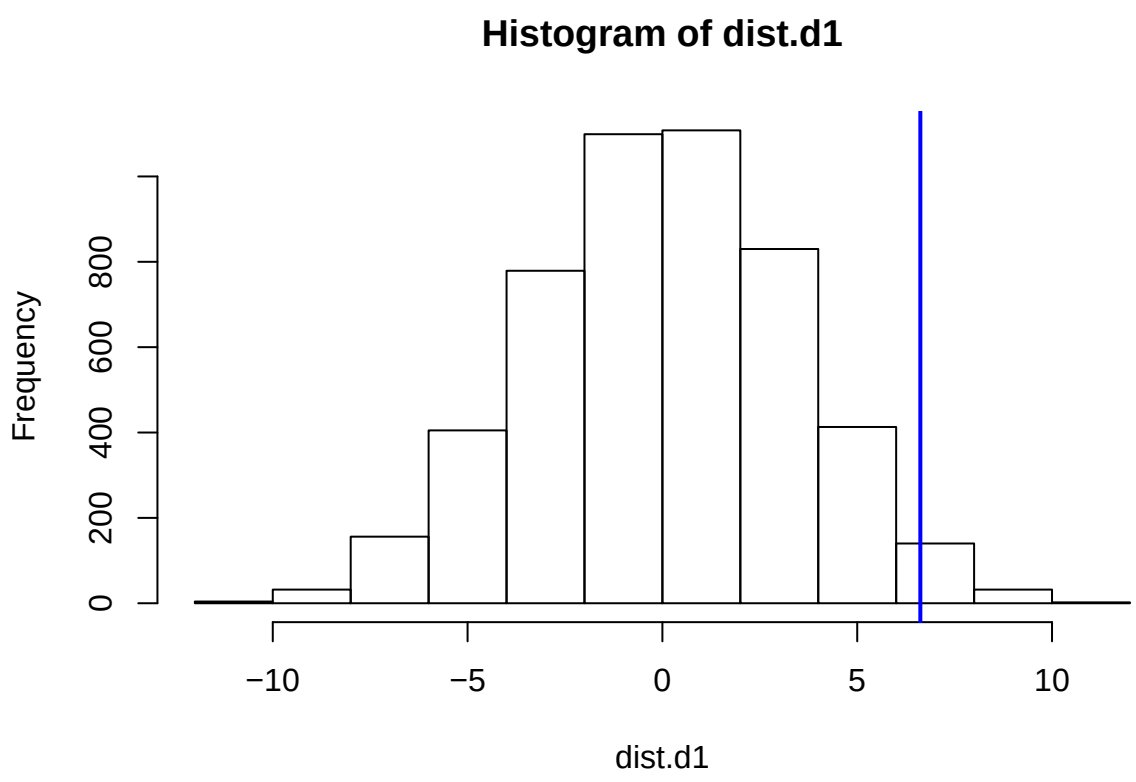
   where $N_1 = \#\{D \geq D^*\}$, $N_2 = \#\{D \leq D^*\}$, and $\#$ means the "number of", e.g., N_1 is the number of these permutated statistics less than or equal to the observed statistic.

```
> d.fun = function(data) {
+   km <- survfit(Surv(lenfol, fstat) ~ 1, data = data)
+   km.gp1=with(data %>% filter(gender==1),survfit(Surv(lenfol, fstat) ~ 1))
+   km.gp0=with(data %>% filter(gender==0),survfit(Surv(lenfol, fstat) ~ 1))
+
+   km.df=data.frame(time=km$time,n.risk=km$n.risk)
```

```
+   km.df.gp1=data.frame(time=km.gp1$time,d1.event=km.gp1$n.event,c1.censor=km.gp1$n.censor)
+   km.df.gp0=data.frame(time=km.gp0$time,d0.event=km.gp0$n.event,c0.censor=km.gp0$n.censor)
+
+   km.gender=left_join(km.df,km.df.gp1,by="time") %>%
+     mutate(d1.event=replace_na(d1.event,0),c1.censor=replace_na(c1.censor,0)) %>%
+     mutate(n1.risk=35-head(c(0,cumsum(d1.event+c1.censor)),-1),s1=cumprod(1-d1.event/n1.risk)) %>%
+     left_join(km.df.gp0,by="time") %>%
+     mutate(d0.event=replace_na(d0.event,0),c0.censor=replace_na(c0.censor,0)) %>%
+     mutate(n0.risk=65-head(c(0,cumsum(d0.event+c0.censor)),-1),s0=cumprod(1-d0.event/n0.risk)) %>%
+     mutate(s1=replace_na(s1,0),s0=replace_na(s0,0))
+
+
+   d1=with(km.gender,sum(d1.event-n1.risk*(d1.event+d0.event)/n.risk))
+   d2=with(km.gender,max(abs(s1-s0)))
+   d3=with(km.gender,min(time[s1<=0.5])-min(time[s0<=0.5]))
+   d4=with(km.gender,sum(head(c(1,s1),-1)*(km.gender$time-head(c(0,km.gender$time),-1))-
+                           head(c(1,s0),-1)*(km.gender$time-head(c(0,km.gender$time),-1))))
+
+   return(list(d1=d1,d2=d2,d3=d3,d4=d4))
+ }
>
> d.obs=d.fun(data=whas100)
>
> set.seed(123)
> dist.d.list=
+  with(whas100,
+   replicate(5000,
+    d.fun(data=data.frame(lenfol=lenfol,fstat=fstat,gender=sample(gender,length(gender),FALSE)))))
>
> dist.d.matrix=matrix(unlist(dist.d.list),ncol=4,byrow=TRUE)
```

We first define funtion d.fun which calculates D1, D2, D3 and D4 from the data. Then we sample gender without replacement and create a new dataset from it. Finally, we use d.fun to calculates D1, D2, D3 and D4. The last two steps are repeated 5000 times. dist.d.matrix(5000 by 4 matrix) holds the sample distribution for D1, D2, D3 and D4.

```
> dist.d1=dist.d.matrix[,1]
>
> hist(dist.d1)
> abline(v=d.obs$d1,col="blue",lwd = 2)
```




```
> summary(dist.d1)
```

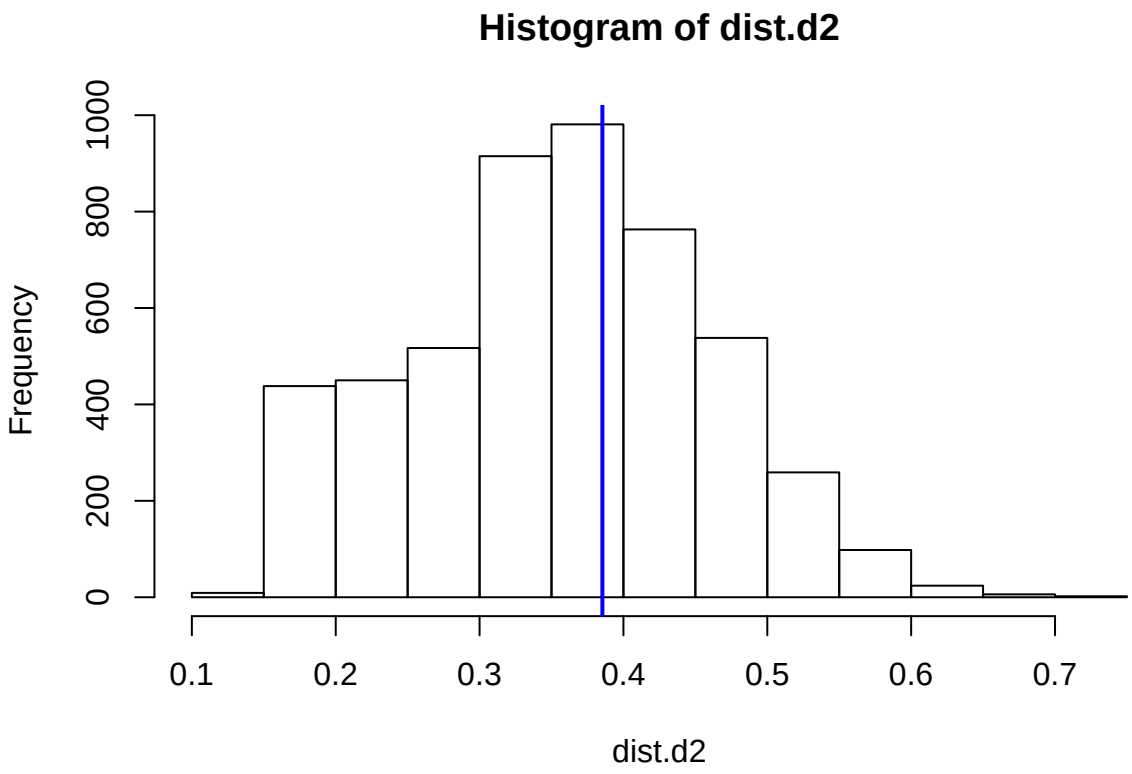
```
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-11.78196  -2.25294   0.05511   0.02653   2.31044  11.97221
```

```
> 2*min(sum(dist.d1>=d.obs$d1),sum(dist.d1<=d.obs$d1))/5000
```

```
[1] 0.046
```

P-value for D1 is 0.046.

```
> dist.d2=dist.d.matrix[,2]
>
> hist(dist.d2)
> abline(v=d.obs$d2,col="blue",lwd = 2)
```




```
> summary(dist.d2)
```

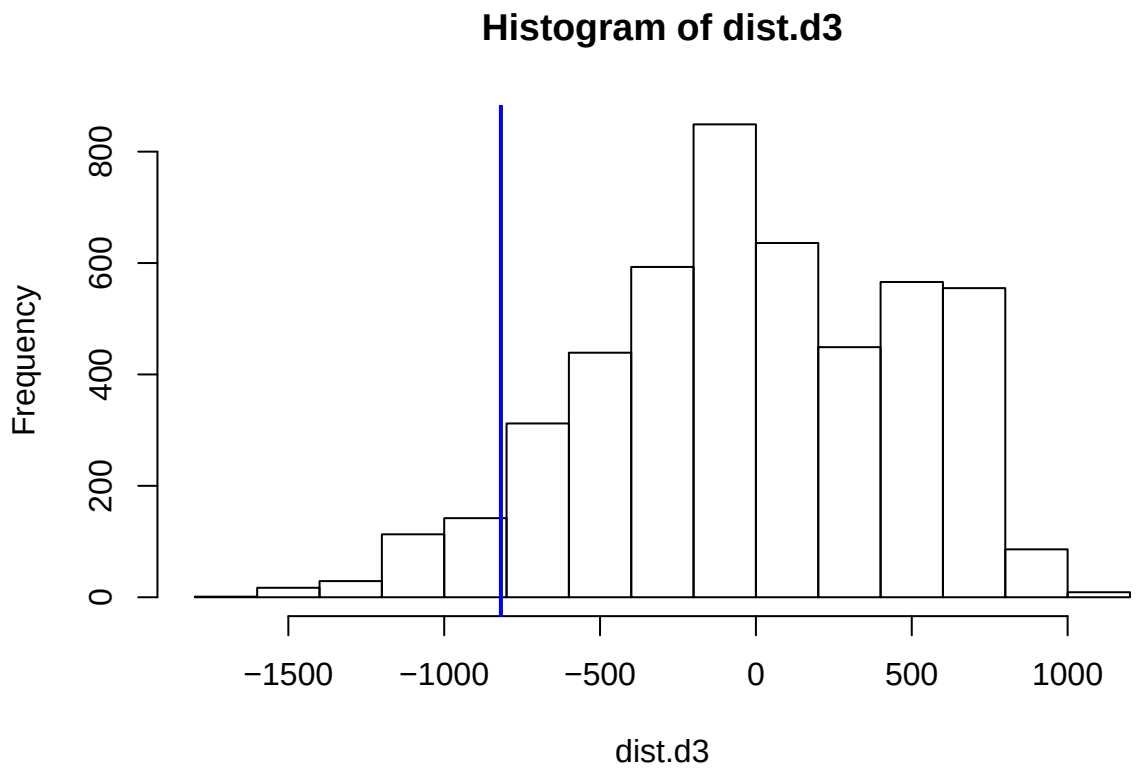
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.1439  0.2872  0.3589  0.3560  0.4283  0.7143
```

```
> 2*min(sum(dist.d2>=d.obs$d2),sum(dist.d2<=d.obs$d2))/5000
```

```
[1] 0.7888
```

P-value for D2 is 0.7888. Note that the largest survival time is a censored observation, so the permutation test doesn't center at 0.

```
> dist.d3=dist.d.matrix[,3][is.finite(dist.d.matrix[,3])]
>
> hist(dist.d3)
> abline(v=d.obs$d3,col="blue",lwd = 2)
```




```
> summary(dist.d3)
```

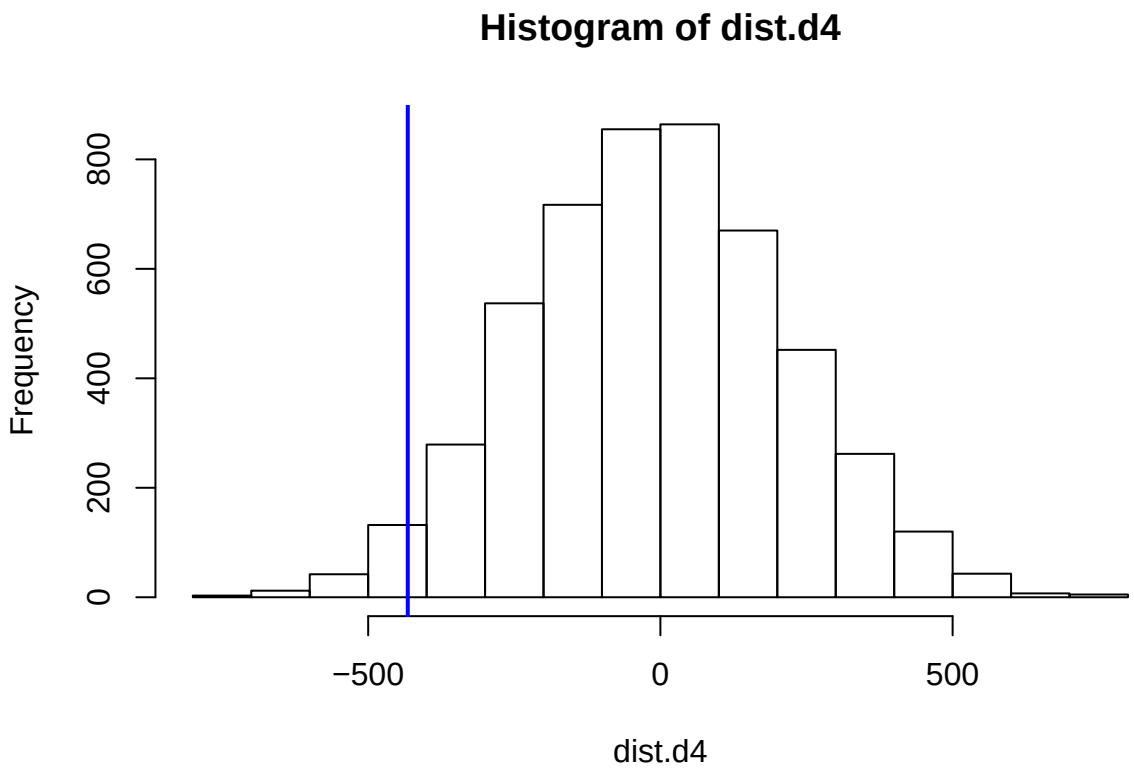
```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1662.00  -390.00  -136.00   -16.77   409.00  1133.00
```

```
> 2*min(sum(dist.d3>=d.obs$d3),sum(dist.d3<=d.obs$d3))/length(dist.d3)
```

```
[1] 0.118849
```

P-value for D3 is 0.118849. Note that some sameples do not have median survival function, so we remove those D3 and adjust the denominator when calculate the p-value.

```
> dist.d4=dist.d.matrix[,4]
>
> hist(dist.d4)
> abline(v=d.obs$d4,col="blue",lwd = 2)
```




```
> summary(dist.d4)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-763.063 -160.088   -7.595   -8.316  142.017  741.945
```

```
> 2*min(sum(dist.d4>=d.obs$d4),sum(dist.d4<=d.obs$d4))/5000
```

```
[1] 0.0548
```

P-value for D4 is 0.0548.

4. Another method to compare two survival curves is to consider a sign test. Suppose we have two groups of uncensored survival times:

   Males: $x_1, x_2, \ldots, x_{n_0}$.

   Females: $y_1, y_2, \ldots, y_{n_1}$.

   The sign test looks at the statistic

$$U = \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} \text{sgn}(x_i - y_j),$$

where sgn(·) is the sign function. In the pretense of right censoring, survival times can not be compared directly and a modified version of $U = \sum_{i=1}^{n_0} \sum_{j=1}^{n_1} U_{ij}$ is considered, where

$$U_{ij} = \begin{cases} 1 & \text{if } x_i > y_j, y_j \text{ is uncersored.} \\ -1 & \text{if } x_i < y_j, x_i \text{ is uncersored.} \\ 0 & \text{otherwise.} \end{cases}$$

a. (5 pts) Compute U for the WHAS100 dataset.
b. (5 pts) Create a histgram for these permutated statistics and print the `summary`. Obtain a permutation p-value based on 5000 permutations.

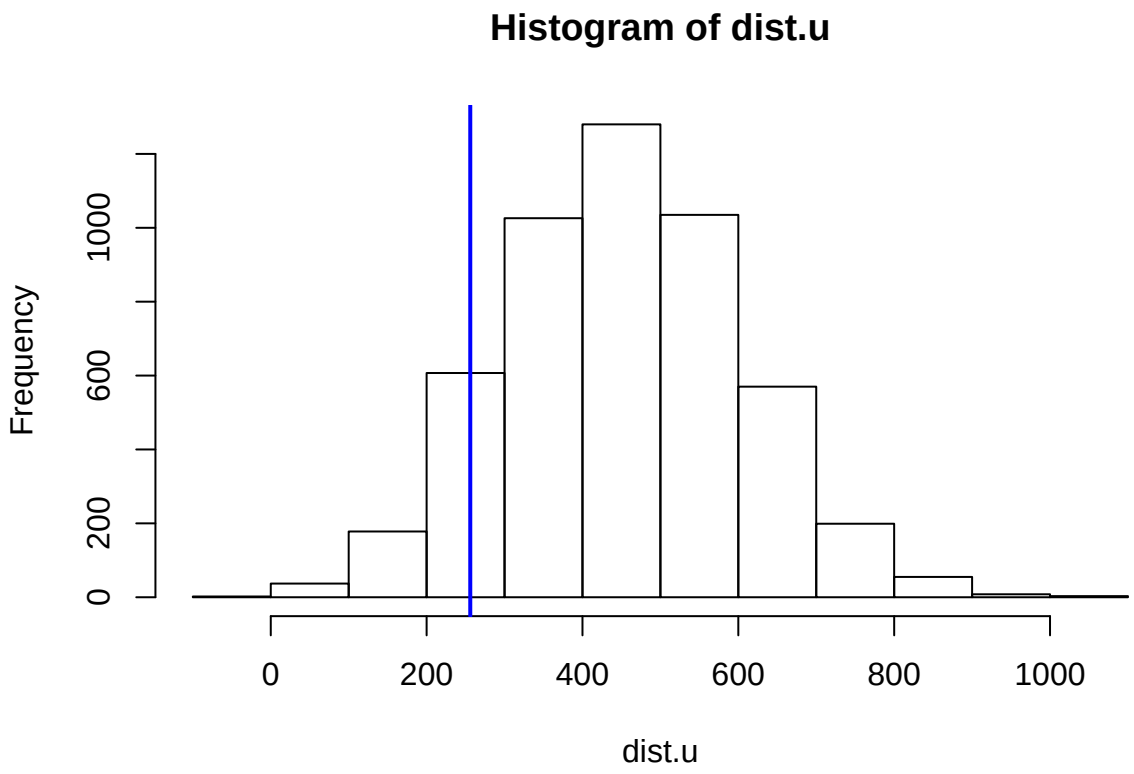
```
> grid.1u0u=with(whas100 %>% filter(fstat==1),
+                expand.grid(lenfol[gender==1],lenfol[gender==0]))
> grid.1c0u=with(whas100 %>% filter((fstat==0&gender==1)|(fstat==1&gender==0)),
+                expand.grid(lenfol[gender==1],lenfol[gender==0]))
> grid.1u0c=with(whas100 %>% filter((fstat==1&gender==0)|(fstat==0&gender==1)),
+                expand.grid(lenfol[gender==1],lenfol[gender==0]))
>
> ##u
> with(grid.1u0u,sum(Var1>Var2)-sum(Var1<Var2))+
+ with(grid.1c0u,sum(Var1>Var2))+
+ with(grid.1u0c,-sum(Var1<Var2))
```

```
[1] 256
```

U is 256.

```
> u.fun = function(data) {
+   grid.1u0u=with(data %>% filter(fstat==1),
+                  expand.grid(lenfol[gender==1],lenfol[gender==0]))
+   grid.1c0u=with(data %>% filter((fstat==0&gender==1)|(fstat==1&gender==0)),
+                  expand.grid(lenfol[gender==1],lenfol[gender==0]))
+   grid.1u0c=with(data %>% filter((fstat==1&gender==0)|(fstat==0&gender==1)),
+                  expand.grid(lenfol[gender==1],lenfol[gender==0]))
+
+   u=with(grid.1u0u,sum(Var1>Var2)-sum(Var1<Var2))+
+     with(grid.1c0u,sum(Var1>Var2))+
+     with(grid.1u0c,-sum(Var1<Var2))
+
+   return(u)
+ }
>
> u.obs=u.fun(whas100)
```

```
> set.seed(123)
> dist.u=
+  with(whas100,
+   replicate(5000,
+    u.fun(data=data.frame(lenfol=lenfol,fstat=fstat,gender=sample(gender,length(gender),FALSE)))))
>
> hist(dist.u)
> abline(v=u.obs,col="blue",lwd = 2)
```




```
> summary(dist.u)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  -41.0   346.0   453.0   452.6   555.0  1100.0
```

```
> 2*min(sum(dist.u>=u.obs),sum(dist.u<=u.obs))/5000
```

```
[1] 0.2052
```

P-value is 0.2052.