

# Exam 1

Steven Chiou

## Instructions

- Due date: **Tuesday, November 13.**
- This is an 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 total 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  $H_0 : S_0(t) = S_1(t)$ . Do this *without a software package* (6 pts), and verify the calculation with *survdif* (4 pts).

## Solution:

We will first load the *tidyverse* package (but not the *survival* package):

```
> library(tidyverse)
```

The following codes calculate the necessary quantities to calculate  $Q$ :

```
> attach(whas100)
> dat <- tibble(Time = unique(sort(lenfol[fstat > 0]))) %>%
+   mutate(ni = colSums(outer(lenfol, Time, ">=")),
+          di = colSums(outer(lenfol, Time, "==")),
+          ni.1 = colSums(outer(lenfol[gender == 1], Time, ">=")),
+          di.1 = colSums(outer(lenfol[gender == 1], Time, "==")),
+          ni.0 = ni - ni.1, di.0 = di - di.1,
+          E = ni.1 * di / ni, V = ni.1 * ni.0 * di * (ni - di) / ni / ni / (ni - 1))
> detach(whas100)
```

Compute  $Q$ :

```
> with(dat, sum(di.1 - E)^2 / sum(V))
```

```
[1] 3.971377
```

2. There are many ways to form a basis for survival curve comparison. Here are some:
- The numerator term in  $Q$  without the square:

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

- The largest distance between the two curves:

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

- The difference between the median survival times:

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

- 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).

**Solution:**

Taking advantage of the `dat` created in #1, these quantity can be easily computed with the following codes:

a.

```
> with(dat, sum(di.1 - E))
```

```
[1] 6.62
```

b.

```
> dat <- dat %>% mutate(km0 = cumprod((ni.0 - di.0) / ni.0),
+                        km1 = cumprod((ni.1 - di.1) / ni.1))
> max(abs(dat$km0 - dat$km1))
```

```
[1] 0.3853606
```

c.

```
> with(dat, Time[which.max(km1 <= .5)] - Time[which.max(km0 <= .5)])
```

```
[1] -818
```

d.

```
> with(dat, sum(-diff(rev(Time)) * rev(km1 - km0)[-1]))
```

```
[1] -428.7407
```

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 permuted statistics  $D_{1i}^*$ ,  $D_{2i}^*$ ,  $D_{3i}^*$  and  $D_{4i}^*$  for  $i = 1, \dots, 5000$ . Create a histogram for these permuted 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 permuted statistics less than or equal to the observed statistic.

### Solution:

We first prepare a function that returns the 4 statistics in #2:

```
> #' @param obs is the observed survival time
> #' @param event is the censoring indicator
> #' @param x is the categorical covariate; taking values of 0 and 1
> getD <- function(obs, event, x) {
+   dat0 <- tibble(Time = unique(sort(obs[event > 0]))) %>%
+     mutate(ni = colSums(outer(obs, Time, ">=")),
+            di = colSums(outer(obs, Time, "==")),
+            ni.1 = colSums(outer(obs[x > 0], Time, ">=")),
+            di.1 = colSums(outer(obs[x > 0], Time, "==")),
+            ni.0 = ni - ni.1, di.0 = di - di.1,
+            E = ni.1 * di / ni, V = ni.1 * ni.0 * di * (ni - di) / ni / ni / (ni - 1),
+            km0 = pmax(0, cumprod((ni.0 - di.0) / ni.0), na.rm = TRUE),
+            km1 = pmax(0, cumprod((ni.1 - di.1) / ni.1), na.rm = TRUE))
+   attach(dat0)
+   d1 <- sum(di.1 - E)
+   d2 <- max(abs(km0 - km1))
+   d3 <- Time[which.max(km1 <= .5)] - Time[which.max(km0 <= .5)]
+   d4 <- sum(-diff(rev(Time)) * rev(km1 - km0)[-1])
+   detach(dat0)
+   c(d1 = d1, d2 = d2, d3 = d3, d4 = d4)
+ }
```

Check the function with `whas100` to confirm it returns values in #2:

```
> (d0 <- with(whas100, getD(lenfol, fstat, gender)))
```

d1	d2	d3	d4
6.6199997	0.3853606	-818.0000000	-428.7407041

a. The following codes perform the permutation statistics based on 5000 permutation:

```
> set.seed(123)
> system.time(
+   permd <- t(replicate(5000, with(whas100, getD(lenfol, fstat, sample(gender)))))
+ )
```

user	system	elapsed
20.632	0.020	20.674

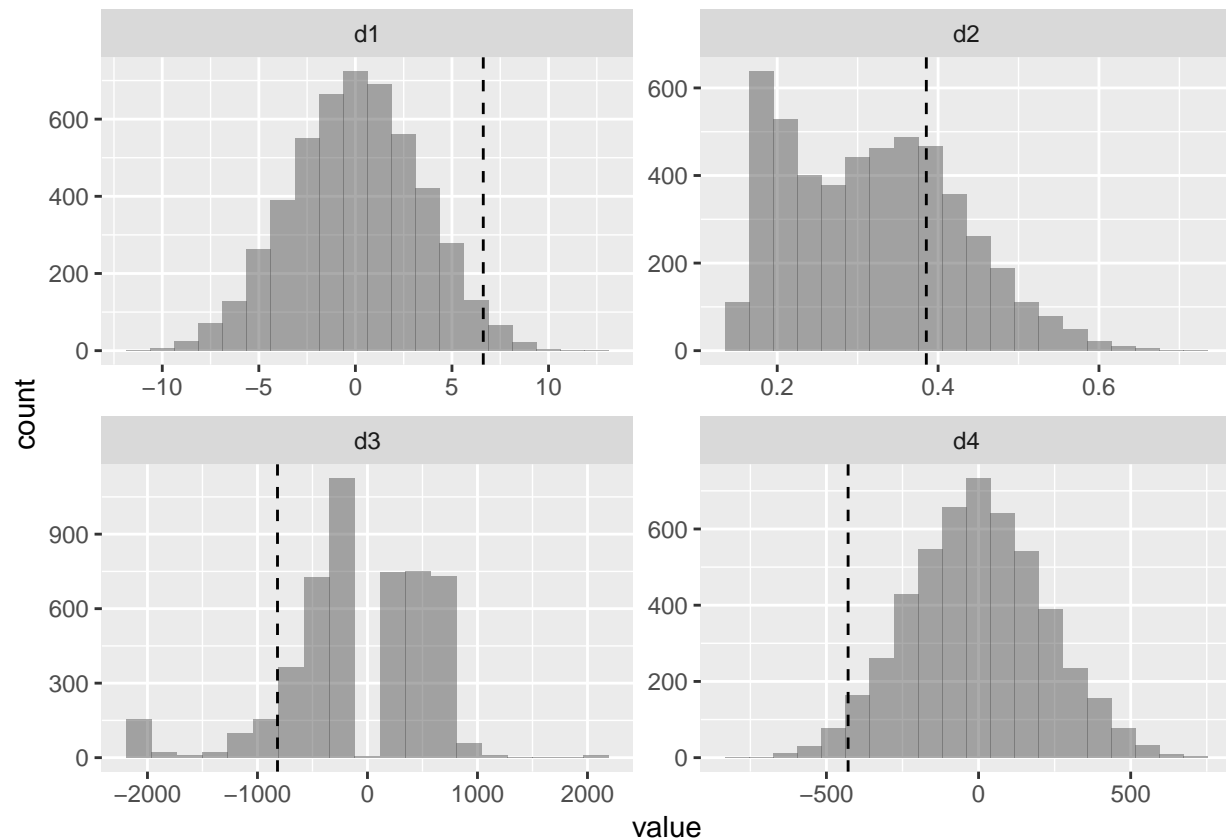
The following codes print the summary and produce the histogram

```
> summary(permd)
```

d1	d2	d3
Min. : -11.78196	Min. : 0.1439	Min. : -2195.00
1st Qu.: -2.25294	1st Qu.: 0.2229	1st Qu.: -395.00
Median : 0.05511	Median : 0.3154	Median : -136.00
Mean : 0.02653	Mean : 0.3189	Mean : -81.82
3rd Qu.: 2.31044	3rd Qu.: 0.3931	3rd Qu.: 390.00
Max. : 11.97221	Max. : 0.7143	Max. : 2195.00

d4
Min. : -763.848
1st Qu.: -154.421
Median : -4.830
Mean : -4.286
3rd Qu.: 146.719
Max. : 743.986

```
> ggdat <- gather(as.tibble(permd), "d", factor_key = TRUE) %>% mutate(d0 = unlist(d0[d]))
> ggplot(ggdat, aes(value)) +
+   geom_histogram(alpha = .5, bins = 20) + geom_vline(aes(xintercept = d0), lty = I(2)) +
+   facet_wrap(d ~., scale = "free")
```



b.

The  $p$ -values are

```
> sapply(1:4, function(x) 2 * min(sum(d0[[x]] < permd[,x]), sum(d0[[x]] > permd[,x])) / 5000)
```

```
[1] 0.0460 0.5532 0.1748 0.0532
```

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

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

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

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  $\text{sgn}(\cdot)$  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

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

- (5 pts) Compute  $U$  for the WHAS100 dataset.
- (5 pts) Create a histogram for these permuted statistics and print the summary. Obtain a permutation  $p$ -value based on 5000 permutations.

**Solution:**

We will take a similar approach as in #3; first write a function to compute  $U$  then use it for permutation. The following function can be used to compute  $U$ :

```
> getU <- function(obs, event, x) {
+   t1 <- obs[x > 0]
+   d1 <- event[x > 0]
+   t0 <- obs[x == 0]
+   d0 <- event[x == 0]
+   sum(outer(t0, t1[d1 > 0], ">")) - sum(outer(t0[d0 > 0], t1, "<"))
+ }
```

- For the whas100 data:

```
> (U0 <- with(whas100, getU(lenfol, fstat, gender)))
```

```
[1] 459
```

- Perform permutation test, calculate  $p$ -value and produce histogram:

```
> set.seed(123)
> permU <- replicate(5000, with(whas100, getU(lenfol, fstat, sample(gender))))
> summary(permU)
```

```
      Min.   1st Qu.   Median     Mean   3rd Qu.    Max.
-916.000 -166.000    2.000    3.146  177.000  896.000
```

```
> 2 * min(sum(U0 < permU), sum(U0 > permU)) / 5000
```

```
[1] 0.0644
```

```
> ggplot() + aes(permU) + geom_histogram(alpha = .5, bins = 20) +
+   geom_vline(aes(xintercept = U0), lty = I(2))
```

