p8160_hw4_jsg2145

Jared Garfinkel

4/19/2020

```
blue <- c(4,69,87,35,39,79,31,79,65,95,68,
          62,70,80,84,79,66,75,59,77,36,86,
          39,85,74,72,69,85,85,72)

red <- c(62,80,82,83,0,81,28,69,48,90,63,
        77,0,55,83,85,54,72,58,68,88,83,78,
        30,58,45,78,64,87,65)

acui <- data.frame(str = c(rep(0,20),
                           rep(1,10)),
                   red,
                   blue)
```

Part 1i

```
one_sui_red = acui %>%
  filter(str == 1) %>%
  dplyr::select(red)

one_sui_blue = acui %>%
  filter(str == 1) %>%
  dplyr::select(blue) %>%
  as.tibble()
```

```
teststat <- function(x = pull(one_sui_red, red), y = pull(one_sui_blue, blue)) {
  return((mean(y) - mean(x))/(var(x)/length(x) + var(y)/length(y)))
}

boottest <- function(x = pull(one_sui_red, red), y = pull(one_sui_blue, blue), nboot = 200) {
  combmean <- mean(c(x, y))
  # The mean of the combined sample
  teststatvec <- rep(NA, nboot)
  adjx <- x - mean(x) + combmean
  # The adjusted X's will have mean=combmean
  adjy <- y - mean(y) + combmean
  # The adjusted X's will have mean=combmean
  for(b in 1:nboot) {
    teststatvec[b] <- teststat(sample(adjx, replace=T),
                               sample(adjy, replace=T))
  }
  return(list(bootpval = sum(teststat(x, y) < teststatvec)/nboot,
              teststatvec = teststatvec))
}
```

```
teststat()
```

```
## [1] 0.03824062
```

```
boottest()
```

```
## $bootpval
## [1] 0.35
##
## $teststatvec
##   [1]  0.100795325  0.111401747 -0.222080408  0.213713268 -0.008319852
##   [6] -0.242649214  0.007269056 -0.250437522 -0.143217080 -0.183024558
##  [11]  0.063603491 -0.149536692 -0.009406191  0.200101747  0.084366363
##  [16] -0.090596426 -0.214536928  0.104471282  0.126080520  0.147806739
##  [21]  0.073257150 -0.080855680 -0.036805763 -0.101570167  0.132294310
##  [26] -0.195337621  0.093728846  0.031054115 -0.048024759 -0.102992013
##  [31] -0.138798268  0.204807692  0.088060749  0.056997198  0.025081864
##  [36]  0.147883125 -0.008226503 -0.036347727  0.162327478 -0.014472944
##  [41] -0.188838224 -0.158613373  0.114693032  0.130782272 -0.005682356
##  [46]  0.001628075  0.359927653  0.234473901  0.011727233  0.144021336
##  [51] -0.014934248 -0.098497962  0.209295220  0.112251882  0.387031841
##  [56] -0.059330015  0.001691729 -0.070242145  0.226742417 -0.048983099
##  [61] -0.030562866 -0.145731681 -0.011311980  0.145937318  0.002814919
##  [66] -0.160329341  0.058247041 -0.089002764 -0.225350335 -0.209133971
##  [71] -0.249390987 -0.031170630 -0.134408146 -0.160415042  0.198978894
##  [76] -0.146603255  0.097265108  0.145481675 -0.082226062 -0.097588402
##  [81] -0.012445220 -0.024366974 -0.127209747  0.084552213 -0.112350200
##  [86] -0.023591431  0.136686615 -0.196543866  0.046471049  0.133473305
##  [91]  0.034907398  0.033754947  0.049687571  0.047912766 -0.068302257
##  [96] -0.148263753 -0.075764864 -0.024293793 -0.160176747 -0.185535131
## [101]  0.080222570  0.063705539 -0.013444675 -0.028226931 -0.044160168
## [106]  0.038545961 -0.014110374 -0.163324104 -0.045233125 -0.137107120
## [111] -0.092479260 -0.024301144 -0.094535584 -0.006401024 -0.014968249
## [116] -0.064376471  0.016911617 -0.214616979  0.142898762  0.062921479
## [121]  0.373475396 -0.081402711 -0.188553162  0.062191510 -0.134840189
## [126]  0.010254908  0.153428800 -0.045875651 -0.146711530 -0.298752732
## [131] -0.238259821  0.148173012  0.093986014 -0.243385973 -0.024857587
## [136]  0.242373590 -0.560786486  0.001403706 -0.149251256 -0.064308248
## [141] -0.125871192  0.002199709  0.028088792 -0.004327235  0.016475973
## [146] -0.370132374 -0.178832880 -0.207017544 -0.073262869  0.133981154
## [151]  0.157057521 -0.157489036 -0.169230769  0.069861779  0.024657534
## [156] -0.085804887 -0.044356219  0.343444418 -0.135126681 -0.159315533
## [161]  0.239736290  0.184675746  0.008011679 -0.059586618 -0.198171244
## [166] -0.012650141  0.079353833  0.016674386 -0.171407011 -0.074293387
## [171] -0.137784926 -0.197210311  0.265468654 -0.075817850  0.015989132
## [176]  0.072322930  0.073692992 -0.020611259  0.118648612 -0.213569747
## [181]  0.081846456 -0.026045574 -0.088264683 -0.131931166 -0.095281307
## [186]  0.023725965  0.118286364  0.090291712  0.054336468  0.506231790
## [191] -0.066145590 -0.311943950  0.138886275  0.135322341  0.077672459
## [196] -0.027790254  0.192609115 -0.044859813  0.119002103  0.314463296
```

```r
paired_sample_df = acui %>%
  filter(str == 0) %>%
  mutate(diff = red - blue) %>%
  as.tibble()
```

```r
set.seed(22)
teststat_paired <- function(x = pull(paired_sample_df, diff)) {
    return(mean(x)/(sqrt(var(x)/length(x))))
}

set.seed(22)
boottest_paired <- function(diff = pull(paired_sample_df, diff), nboot=200) {
  teststatvec <- rep(NA, nboot)
  adjdiff <- diff - mean(diff)
  for (b in 1:nboot) {
    teststatvec[b] <- teststat_paired(sample(adjdiff, replace = TRUE))
  }
  return(list(bootpval = sum(teststat_paired(diff) < teststatvec)/nboot,
              teststatvec = teststatvec))
}

set.seed(22)
boottest_paired()
```

```
## $bootpval
## [1] 0.695
##
## $teststatvec
##   [1]  0.957576295 -0.022370624 -1.054028451 -0.577034097 -1.450720899
##   [6] -0.024816519 -1.575152645 -2.633822752  1.704612003  0.509158683
##  [11]  0.460948346  2.008182888  0.804934588  0.183665916 -0.471302304
##  [16]  1.929950206  0.603356075 -0.111346951  0.236425088  0.229227082
##  [21] -0.401614992  0.117134139  1.127811764  1.426999536  1.065242616
##  [26]  0.774747804  0.611731503  0.028367292  1.381058734 -0.760918630
##  [31] -0.064634824  0.094946239  0.804864586  1.002675784  1.186610143
##  [36] -1.008347582  0.521854522  1.059073864 -0.499745056  1.299448399
##  [41] -0.542208213  1.017752975 -0.696005030  1.772556269  1.000724861
##  [46] -0.403687267 -0.446486363 -2.057092240 -0.298502613  0.808856226
##  [51] -0.288606164  1.809229192  1.958058034 -0.226639457 -0.233602846
##  [56] -1.470461329 -0.289839230 -0.582855294  0.068433857 -0.639584446
##  [61] -1.223342562 -1.138381055 -1.362016269  1.421283168  1.421950719
##  [66] -1.520017904  0.261891689 -2.449482898  0.744534627  1.150952309
##  [71] -0.308618136 -0.963926888  0.415438583 -0.417544209 -1.137254035
##  [76]  0.013577609 -0.062931022  0.546642067  0.048736707 -1.019026230
##  [81]  1.945891686 -0.806507704  0.672517057  0.983576888  0.411939767
##  [86] -1.348939701 -1.150283861  1.526998439  0.104940775 -0.267298285
##  [91]  0.529578144  1.987731620  0.249083767  0.200580367 -0.622611817
##  [96]  0.498845149  1.349254182  0.769015255 -2.441879724 -0.656052588
## [101] -2.917162029 -0.589972236 -1.219204662 -0.601381705 -0.129826419
## [106] -0.487663204  0.670140307 -0.778887054 -1.157142457 -0.521200111
## [111]  0.081959431 -0.489478707  0.632514859 -1.798208231 -0.078549346
## [116]  1.764412282 -0.596498778 -0.601223252 -0.457398095 -0.882080897
## [121]  0.164897017 -0.161453006  1.234431085  2.079115497 -0.040360175
```

```
## [126]  0.395743976  0.660860384 -0.836636175  0.249253035 -0.243086004
## [131] -0.329610582 -0.253815053 -0.579681157 -1.223256486  0.308478310
## [136] -0.884898575 -2.176591360 -0.163157204 -0.263735306  0.273042753
## [141] -1.107561811  1.404944033  1.447835819 -1.876965146  1.374055732
## [146]  0.705149696  0.336866349  0.433902936 -0.634370429 -1.309157240
## [151]  1.478370845  1.394084380 -2.554712189 -0.579740873 -1.159310197
## [156]  1.428679920  0.522239182 -0.255569713 -0.240859834 -0.459193831
## [161]  0.198899537 -1.364305781 -1.434015553  0.485703852 -0.621613350
## [166]  1.274019551 -1.852633436 -1.611185892  0.608796185  0.386794494
## [171]  0.896587474  0.238547059 -0.821349464 -0.074099713 -0.972724478
## [176]  0.718012491  0.744798761 -0.744069480  1.125333848 -1.057597961
## [181] -0.395479635  1.571920665  0.048382062 -0.455542561  0.466081777
## [186] -1.177513018 -0.254874413  1.828480885 -0.390746821  0.685769900
## [191] -0.185499111  0.008983621  1.476246004 -0.582439083  0.123076973
## [196]  2.388174014  0.467490284  1.284166659 -0.663028022 -1.086267524
```

Problem 1b

```
np = function (size = 200, func) {
  boottest_boot = func()$teststatvec %>%
    sample(size = 200, replace = TRUE)
  n = sum(boottest_boot > func()$bootpval)
  np = n/size
  return(np)
}

treffect_lower_paired = qnorm(.025, mean = mean(sample(paired_sample_df$diff, size = length(paired_sampl
treffect_upper_paired = qnorm(.025, mean = mean(sample(paired_sample_df$diff, size = length(paired_sampl

c(treffect_lower_paired, treffect_upper_paired)
```

```
## [1]  2.990036 -2.309964
```

The p-value for the paired samples is 0.24 and the p-value for the two-sample data is 0.035, which shows a difference for different types of patients.

The confidence interval for the effect size in paired data is (2.99, -2.31), which is not what would be expected based on the p-values.

in parallel

```
set.seed(22)
nCores <- detectCores() # detect numbers of available cores
cl = makeCluster(nCores)
cl
```

```
## socket cluster with 8 nodes on host 'localhost'
```

```
system.time({
res2 = boottest_paired()
})
```

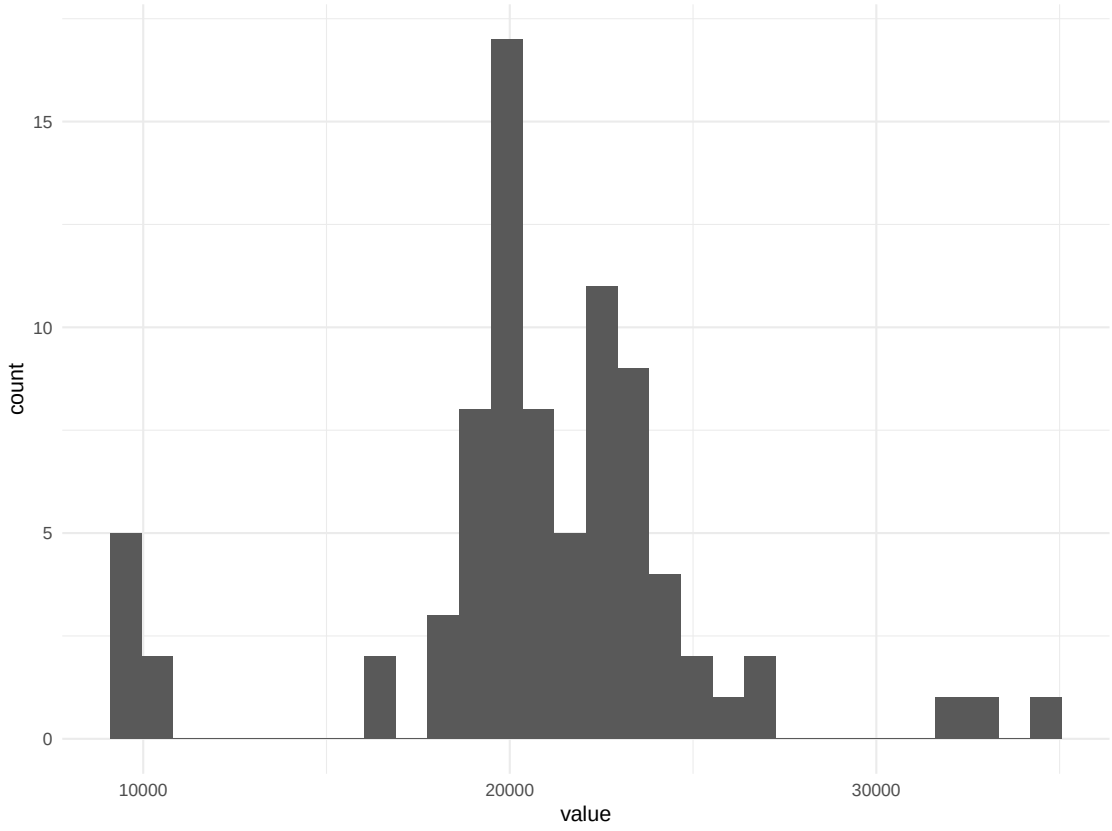
```
##    user  system elapsed
##       0       0       0
```

```
stopCluster(cl)
```

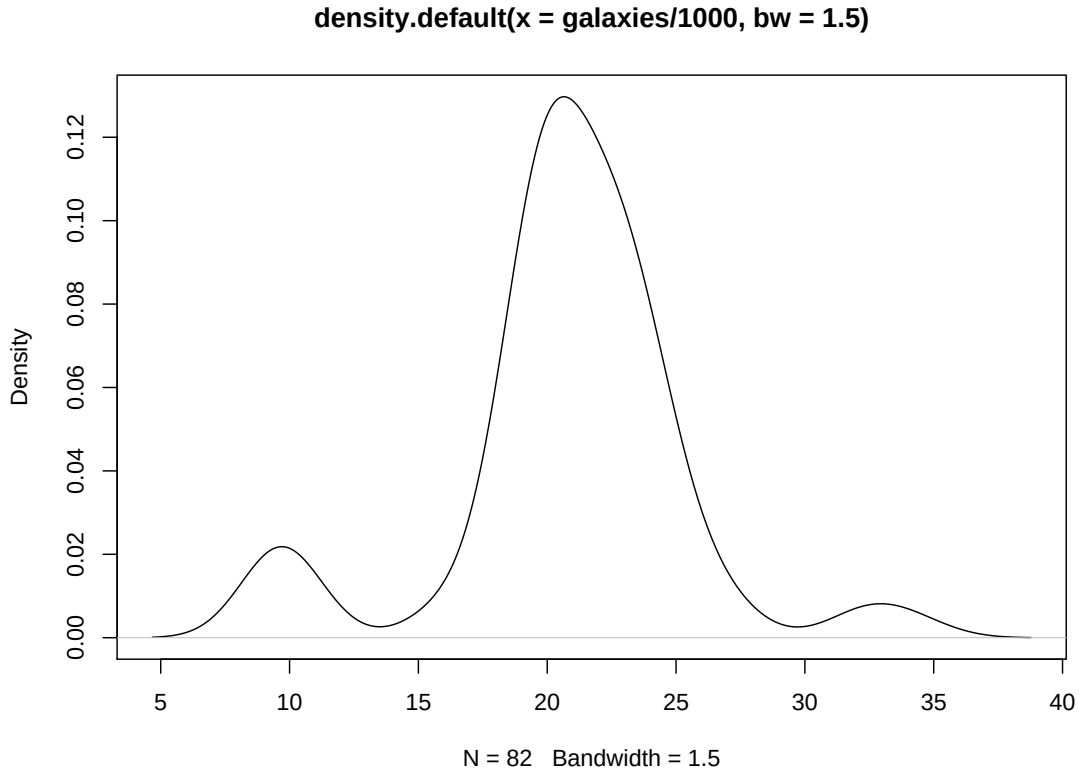
Problem 2

```
data(galaxies)
galaxies %>%
  as_tibble() %>%
  ggplot(aes(x = value)) +
  geom_histogram()
```

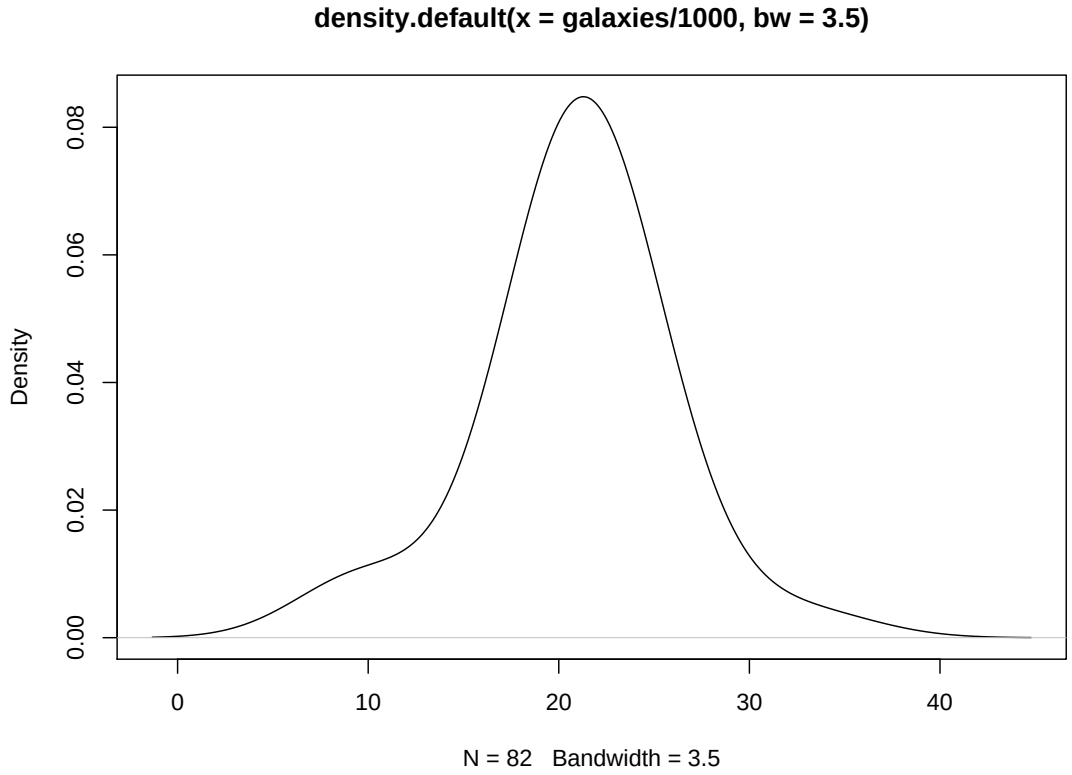
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
plot(density(galaxies/1000, bw=1.5))
```




```
plot(density(galaxies/1000, bw=3.5))
```




```
#calculate the number of modes in the density
den <- density(galaxies/1000, bw=1.5)
den.s <- smooth.spline(den$x, den$y, all.knots = TRUE, spar = 0.8)
s.1 <- predict(den.s, den.s$x, deriv=1)
nmodes <- length(rle(den.sign <- sign(s.1$y))$values)/2
```

```
unif_dens = function(u){
  u = runif(82)
}

data = galaxies

x = rep(NA, 82)
for (i in 1:82) {
  x[i] = unif_dens(u)[[i]] %*% data[[i]]
}

h = (4 * min(sd(data), IQR(data))/(3*200))^(1/5)

kernel = function(x) {
  mean(unif_dens((x - data)/h))/h
}

kernel(x)
```

```
## [1] 0.2613537
```

```
kpdf = function(x) {
  sapply(x, kernel)
}

kpdf(x)
```

```
##  [1] 0.3052535 0.2904226 0.2335151 0.2551579 0.2429064 0.2685964 0.2361574
##  [8] 0.2394938 0.2825924 0.2802298 0.2708697 0.2496031 0.2575829 0.2727521
## [15] 0.2709521 0.2174171 0.2261209 0.2794149 0.2751300 0.2712368 0.2665116
## [22] 0.2971769 0.2816363 0.2699804 0.2377779 0.2655621 0.2413035 0.2437198
## [29] 0.2670400 0.2933334 0.2862980 0.2624176 0.2645550 0.2819202 0.2655556
## [36] 0.2793238 0.2593900 0.2589355 0.2578916 0.2742556 0.2446879 0.2467353
## [43] 0.2693671 0.2592562 0.2508607 0.2653156 0.2656719 0.2766899 0.2556559
## [50] 0.2687743 0.2530010 0.2517961 0.2666387 0.2662923 0.2830011 0.2692714
## [57] 0.2859677 0.2491910 0.2586846 0.2570340 0.2783758 0.2529073 0.2790781
## [64] 0.3028477 0.2775777 0.2732184 0.2826819 0.2810434 0.3095742 0.2784450
## [71] 0.2713188 0.2658945 0.2668400 0.2842546 0.2559046 0.2554585 0.2302489
## [78] 0.2740709 0.2565015 0.2725445 0.2365751 0.2502229
```

```
plot(x,unif_dens(x), type="l", col="red")
par(new=T)
plot(x,kpdf(x),type="l",ylim=c(0,0.23),xlab="",ylab="",axes=F)
```