Chapter 10: Interactive Notebook for Students

Ram Gopal, Dan Philps, and Tillman Weyde

Summer 2022

Contents

Load functions to compute p-value

```
p_rtail = function(sampdist,tstat)
  {
  temp = density(sampdist)
  df = data.frame(temp$x, temp$y)
  formula1 = df$temp.x<tstat
  df1 = df[formula1,]
  plot(df, col = "red", type = "h")
  points(df1, col = "green", type = "h")
  pvalue = length(sampdist[sampdist>tstat])/(length(sampdist))
  return(pvalue)
  }

p_ltail = function(sampdist,tstat)
  {
  temp = density(sampdist)
  df = data.frame(temp$x, temp$y)
  formula1 = df$temp.x>tstat
  df1 = df[formula1,]
  plot(df, col = "red", type = "h")
  points(df1, col = "green", type = "h")
  pvalue = length(sampdist[sampdist<tstat])/(length(sampdist))
  return(pvalue)
  }

p_2tail = function(sampdist,tstat)
  {
hyp = mean(sampdist)
```

```
cutoff1 = hyp - abs(tstat-hyp)
  cutoff2 = hyp + abs(tstat-hyp)
  temp = density(sampdist)
  df = data.frame(temp$x, temp$y)
  formula1 = df$temp.x<cutoff1 | df$temp.x>cutoff2
  df1 = df[formula1,]
  plot(df, col = "green", type = "h")
  points(df1, col = "red", type = "h")
  pvalue = length(sampdist[sampdist<cutoff1 |  sampdist>cutoff2])/(length(sampdist))
  return(pvalue)
  }
```

COVID-19 in Nursing Homes

In this lesson, we will explore the practical case of investigating how COVID-19 infections affected nursing homes. Our study will be carried out through the practical application of some of the statistical tests studied in the previous chapters and it will demonstrate how we can transition from theory to practice.

For our analysis, we downloaded the data on infections as of July 14, 2020. The broad question we want to investigate is whether infections among staff is independent of infections among patients.

Let us take a peek at the data.

```
CA.COVID <- read.csv("../../data/CA COVID.csv")
str(CA.COVID)
```

```
## 'data.frame':    1223 obs. of  14 variables:
##  $ County                                : chr  "Alameda" "Alameda" "Alameda" "Alameda" ...
##  $ FACILITY.NAME                         : chr  "ALAMEDA COUNTY MEDICAL CENTER D/P SNF" "ALAMEDA HEALTH
##  $ COUNTY                                : chr  "ALAMEDA" "ALAMEDA" "ALAMEDA" "ALAMEDA" ...
##  $ FACILITY.ID                           : int  140000321 20000043 630011864 140000686 630013891 200000
##  $ AVAILABLE.BEDS                        : int  7 54 1 3 2 10 7 6 7 34 ...
##  $ AVAILABLE.BEDS.CAPABLE.OF.ISOLATION: int  4 8 1 1 0 0 6 6 1 0 ...
##  $ NEW.CONFIRMED.POSITIVE.RESIDENTS    : chr  "0" "0" "0" "0" ...
##  $ CURRENT.ACTIVE.CASES.RESIDENTS      : chr  "0" "0" "0" "0" ...
##  $ CUMULATIVE.POSITIVE.RESIDENTS       : chr  "86" "88" "<11" "<11" ...
##  $ COVID.RELATED.RESIDENT.DEATHS       : chr  "0" "<11" "0" "0" ...
##  $ NEW.CONFIRMED.POSITIVE.HCW          : chr  "0" "0" "0" "0" ...
##  $ CURRENT.ACTIVE.HCW                  : chr  "<11" "0" "0" "0" ...
##  $ CUMULATIVE.POSITIVE.HCW             : chr  "76" "55" "<11" "13" ...
##  $ COVID.RELATED.HCW.DEATHS            : chr  "0" "0" "0" "0" ...
```

```
size = CA.COVID$AVAILABLE.BEDS
patients_I = CA.COVID$CUMULATIVE.POSITIVE.RESIDENTS
patients_D = CA.COVID$COVID.RELATED.RESIDENT.DEATHS
staff_I = CA.COVID$CUMULATIVE.POSITIVE.HCW
staff_D = CA.COVID$COVID.RELATED.HCW.DEATHS
df = data.frame(size,patients_I, staff_I,patients_D,staff_D)
head(df)
```

```
##   size patients_I staff_I patients_D staff_D
## 1    7         86      76          0       0
```

```
## 2   54         88      55        <11       0
## 3    1        <11     <11          0       0
## 4    3        <11      13          0       0
## 5    2          0     <11          0       0
## 6   10        <11     <11          0       0
```

We are interested in the total number of patients and staff infected.

```
table(df$staff_D)
```

```
##
##  <11    0
##  166 1057
```

Most nursing homes fortunately did not experience any death of their staff and when they did it always was 10 or less deaths.

How many nursing homes had more than 0 infections amongst their staff?

```
nrow(df[!staff_I=="0",])
```

```
## [1] 1201
```

Hypothesis: 15% of nursing homes that had staff infections experienced death amongst their staff.

```
prop.test(166,1201,p=0.15)
```

```
##
##  1-sample proportions test with continuity correction
##
## data:  166 out of 1201, null probability 0.15
## X-squared = 1.2, df = 1, p-value = 0.3
## alternative hypothesis: true p is not equal to 0.15
## 95 percent confidence interval:
##  0.1195 0.1593
## sample estimates:
##      p
## 0.1382
```

For the next analysis, we will take the following steps:

1. Create a new dataframe including only the nursing homes that had staff infections.
2. Create a new variable staff_D_1 coded as none if there are no staff deaths and as some otherwise.
3. Create a new variable staff_I_1 coded as small if the value is <11 and as large otherwise.
4. Conduct a chi-square test with null hypothesis that the level of infections (small or large) is unrelated to the level of deaths (none or some).

```r
df1 = df[!staff_I=="0",]
staff_D_1 = ifelse(df1$staff_D == 0, "none", "small" )
staff_I_1 = ifelse(df1$staff_I == "<11", "small", "large" )
chisq.test(staff_I_1,staff_D_1)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  staff_I_1 and staff_D_1
## X-squared = 4.1, df = 1, p-value = 0.04
```

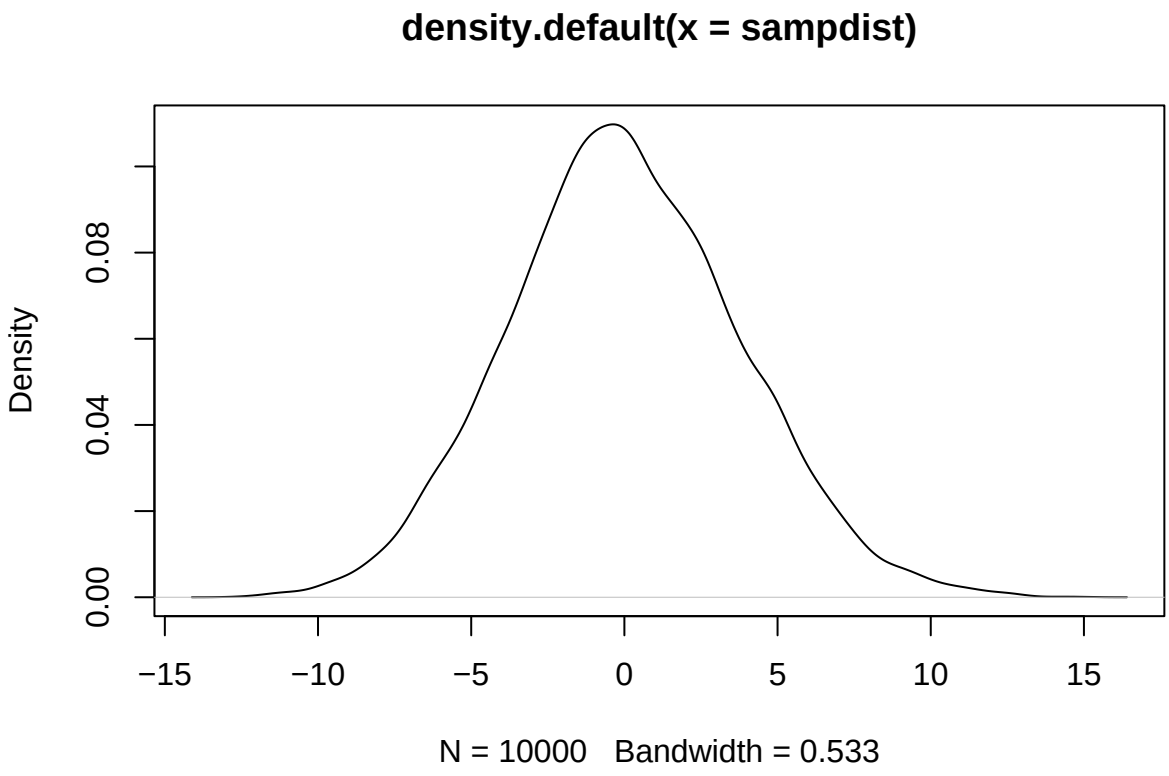
```r
staff_I_2 = ifelse(df1$staff_I=="<11",5,df1$staff_I)
staff_I_2 = as.numeric(staff_I_2)
summary(staff_I_2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     5.0    38.0    61.0    68.5    91.0   556.0
```

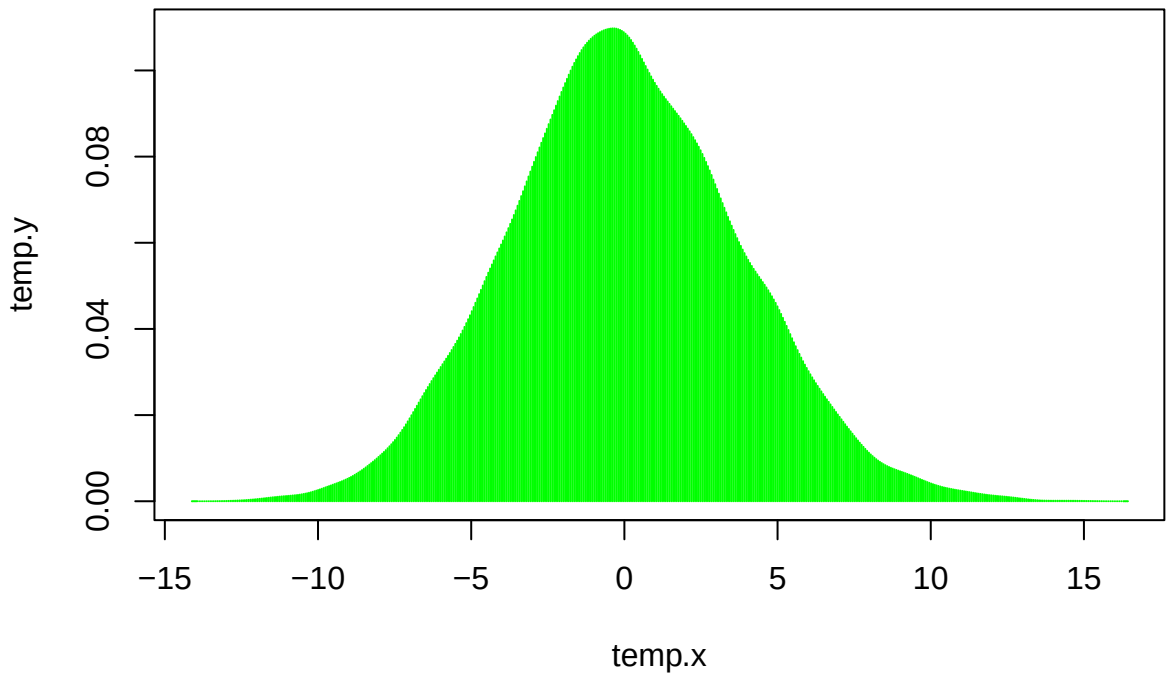
```r
table(staff_D_1)
```

```
## staff_D_1
##  none small
##  1035   166
```

```r
set.seed(87654321)
f1 = function(){
  s1 = sample(staff_I_2)
  control1 = s1[1:1035]
  treatment1 = s1[1036:length(s1)]
  return(mean(treatment1)-mean(control1))
}
sampdist = replicate(10000, f1())
plot(density(sampdist))
```




```
t1= mean(staff_I_2[staff_D_1=="none"])
t2 = mean(staff_I_2[staff_D_1=="small"])
tstat = t2-t1
p_2tail(sampdist,tstat)
```

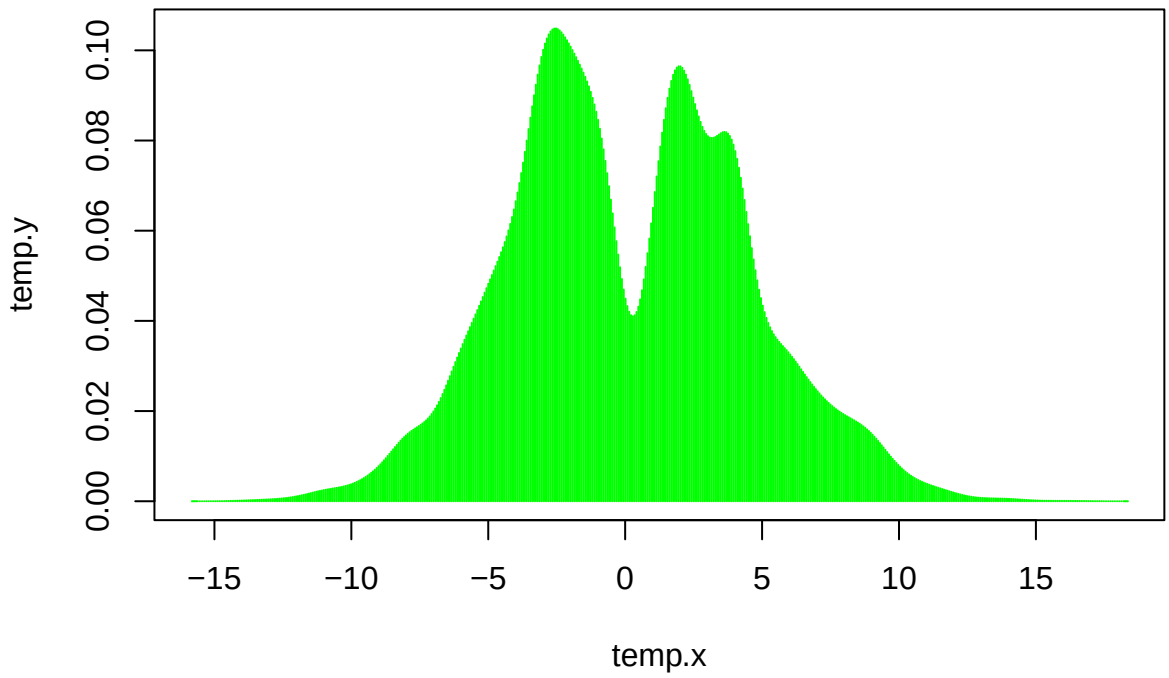



```
## [1] 0
```

Test with Median values

```
set.seed(87654321)
f1 = function(){
  s1 = sample(staff_I_2)
  control1 = s1[1:1035]
  treatment1 = s1[1036:length(s1)]
  return((median(treatment1)-median(control1)))
}
sampdist = replicate(10000, f1())

t1= median(staff_I_2[staff_D_1=="none"])
t2 = median(staff_I_2[staff_D_1=="small"])
tstat = (t2-t1)
p_2tail(sampdist,tstat)
```




```
## [1] 0
```

Patient and Staff Infections

```
staff_I_3 = ifelse(df$staff_I=="<11",5,df$staff_I)
staff_I_3 = as.numeric(staff_I_3)
patient_I_3 = ifelse(df$patients_I=="<11",5,df$patients_I)
patient_I_3 = as.numeric(patient_I_3)

summary(staff_I_3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    36.0    59.0    67.3    90.0   556.0
```

```
summary(patient_I_3)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0    28.0    62.0    69.9   101.0   334.0
```

Are the number of infections reasonably normal?

```
shapiro.test(staff_I_3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  staff_I_3
## W = 0.88, p-value <0.0000000000000002
```

```
shapiro.test(patient_I_3)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  patient_I_3
## W = 0.93, p-value <0.0000000000000002
```

```
length(staff_I_3)
```

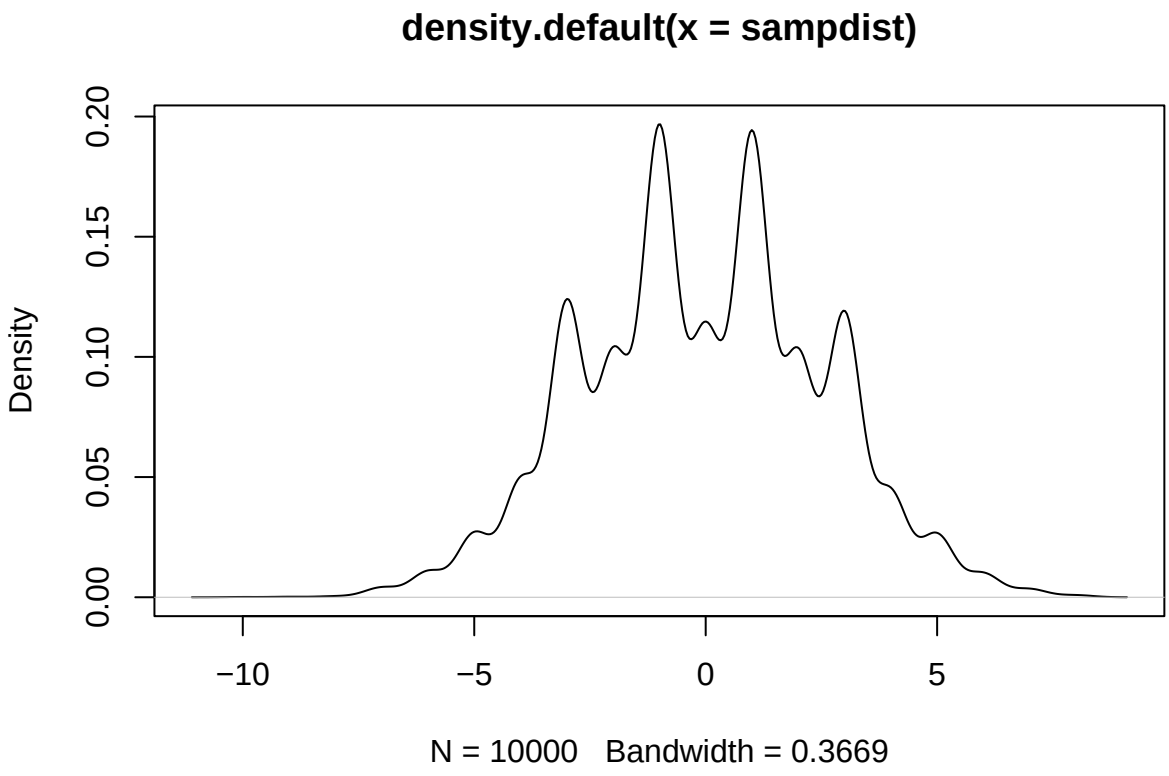
```
## [1] 1223
```

```
length(patient_I_3)
```

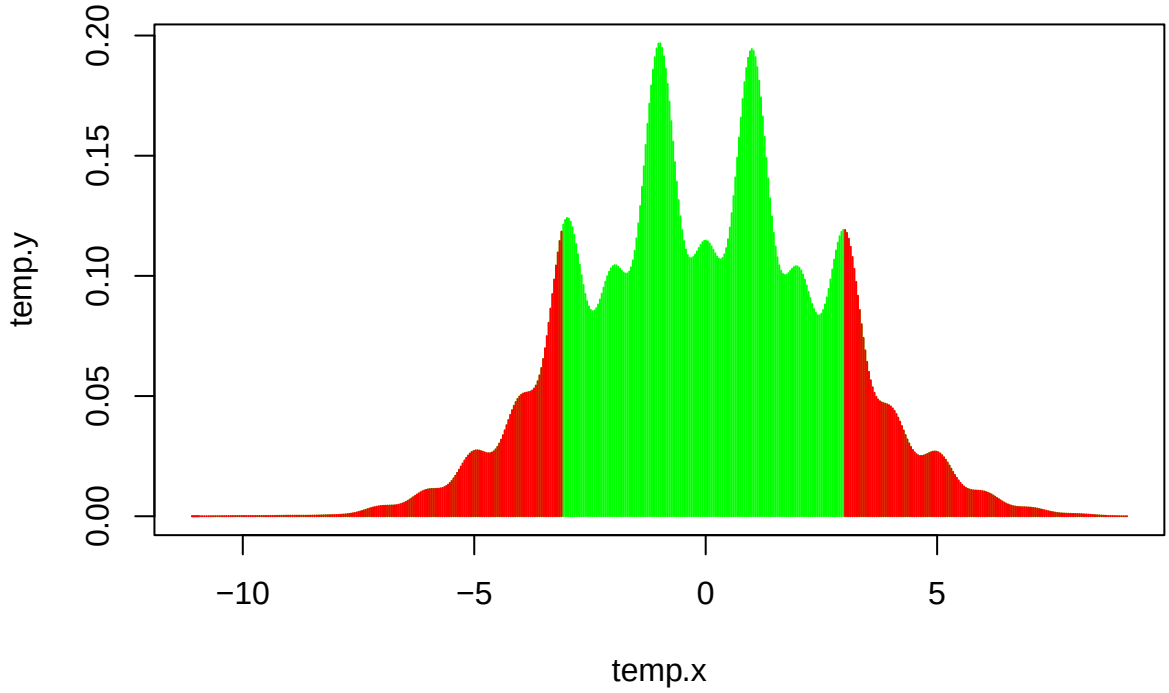
```
## [1] 1223
```

Null hypothesis: the median value of the cumulative infections in a nursing home is the same for both the staff and the patients. The following code performs the nonparametric median test:

```
set.seed(87654321)
n = length(patient_I_3)
f1 = function(){
  pool = c(staff_I_3,patient_I_3)
  s1 = sample(pool)
  control1 = s1[1:n]
  treatment1 = s1[(n+1):(2*n)]
  return(median(treatment1)-median(control1))
}
sampdist = replicate(10000, f1())
plot(density(sampdist))
```



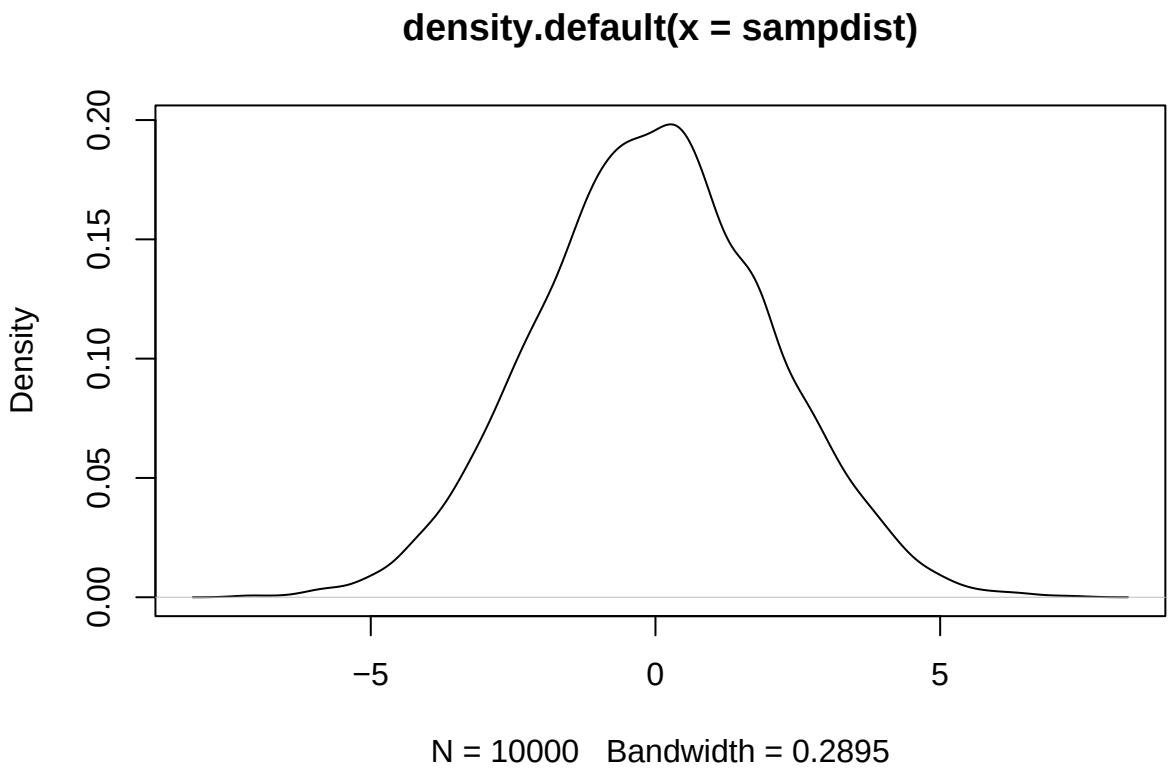

```
tstat = median(patient_I_3)-median(staff_I_3)
p_2tail(sampdist,tstat)
```



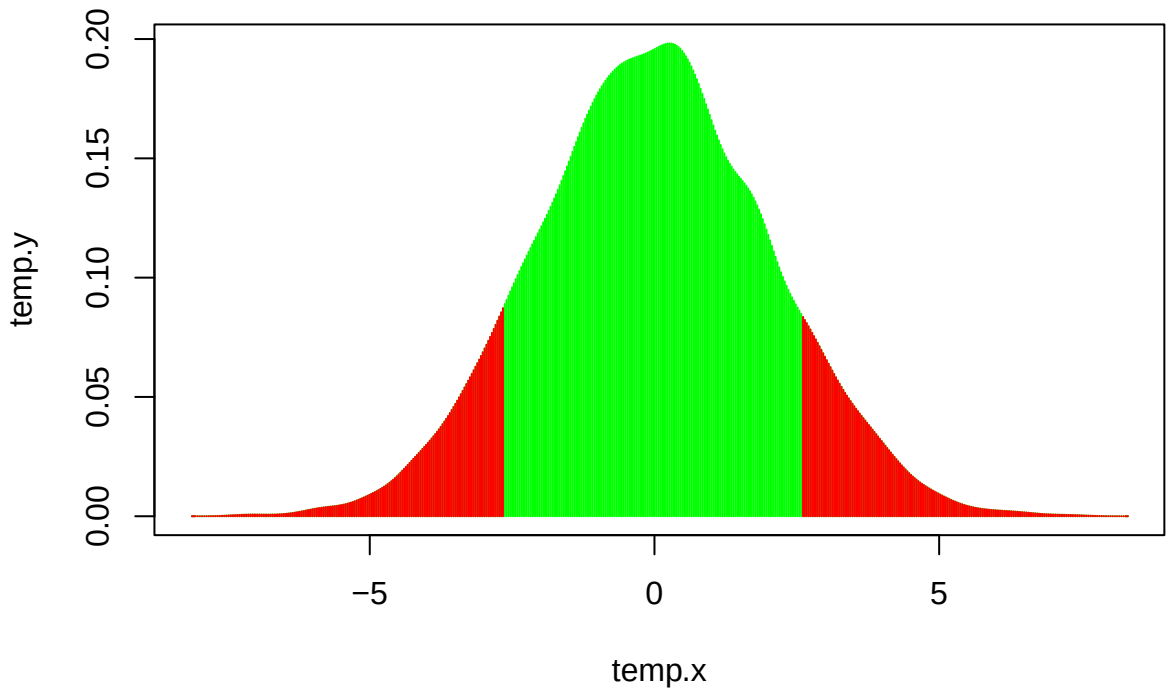

```
## [1] 0.1559
```

Repeat the above test with mean.

```
set.seed(87654321)
n = length(patient_I_3)
f1 = function(){
  pool = c(staff_I_3,patient_I_3)
  s1 = sample(pool)
  control1 = s1[1:n]
  treatment1 = s1[(n+1):(2*n)]
  return(mean(treatment1)-mean(control1))
}
sampdist = replicate(10000, f1())
plot(density(sampdist))
```




```
tstat = mean(patient_I_3)-mean(staff_I_3)
p_2tail(sampdist,tstat)
```



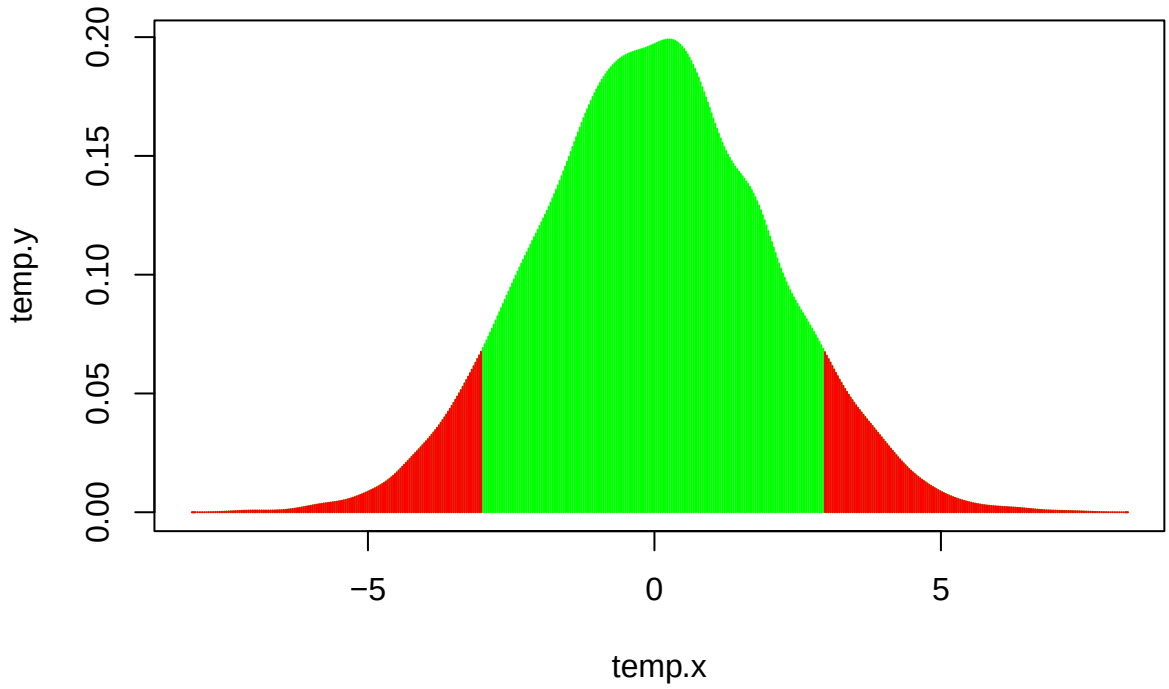

```
## [1] 0.1984
```

As one final check, instead of replacing <11 with 5, we replace it with 1 to evaluate whether the results continue to hold:

```
staff_I_3 = ifelse(df$staff_I=="<11",10,df$staff_I)
staff_I_3 = as.numeric(staff_I_3)
patient_I_3 = ifelse(df$patients_I=="<11",10,df$patients_I)
patient_I_3 = as.numeric(patient_I_3)
```

```
set.seed(87654321)
n = length(patient_I_3)
f1 = function(){
  pool = c(staff_I_3,patient_I_3)
  s1 = sample(pool)
  control1 = s1[1:n]
  treatment1 = s1[(n+1):(2*n)]
  return(mean(treatment1)-mean(control1))
}
sampdist = replicate(10000, f1())

tstat = mean(patient_I_3)-mean(staff_I_3)
p_2tail(sampdist,tstat)
```




```
## [1] 0.1408
```