Hw4

Grady

2020/11/2

- **1a.**

```
paste('apple','banana')
```

```
## [1] "apple banana"
```

```
paste0('apple','banana')
```

```
## [1] "applebanana"
```

```
x <- c('a','b','c')
paste(x,collapse = ', ')
```

```
## [1] "a, b, c"
```

- **1b.**

```
presidents <- c("Clinton", "Bush", "Reagan", "Carter", "Ford")
first.letters <- substr(presidents,1,1)
first.letters
```

```
## [1] "C" "B" "R" "C" "F"
```

- **1c.**

```
first.letters.scrambled <- sample(first.letters)
substr(presidents,1,1) = first.letters.scrambled
presidents
```

```
## [1] "Flinton" "Bush"    "Ceagan"  "Rarter"  "Cord"
```

- **1d.**

```
phrase <- "Give me a break"
phrase<-gsub("Give","Provide",phrase)
phrase<-gsub("break","kit",phrase)
phrase
```

```
## [1] "Provide me a kit"
```

- **2a.**

```r
trump.words <- scan('trump.txt',what = 'c')
trump.words.new <- tolower(trump.words)
trump.words.new <- gsub(pattern='[[:punct:]]',replacement = '',trump.words.new)
```

- **2b.**

```r
trump.words.new <- trump.words.new[-which(nchar(trump.words.new)==0)]
trump.wordtab.new <-as.data.frame(table(trump.words.new))
```

- **2c.**

```r
length(trump.words)-length(trump.words.new)
```

```
## [1] 34
```

```r
trump.wordtab <-as.data.frame(table(trump.words))
nrow(trump.wordtab)-nrow(trump.wordtab.new)
```

```
## [1] 322
```

- **2d.**

```r
astrump.wordtab.sorted.new <- trump.wordtab.new[order(trump.wordtab.new$Freq,decreasing = T),]
astrump.wordtab.sorted.new[1:25,]
```

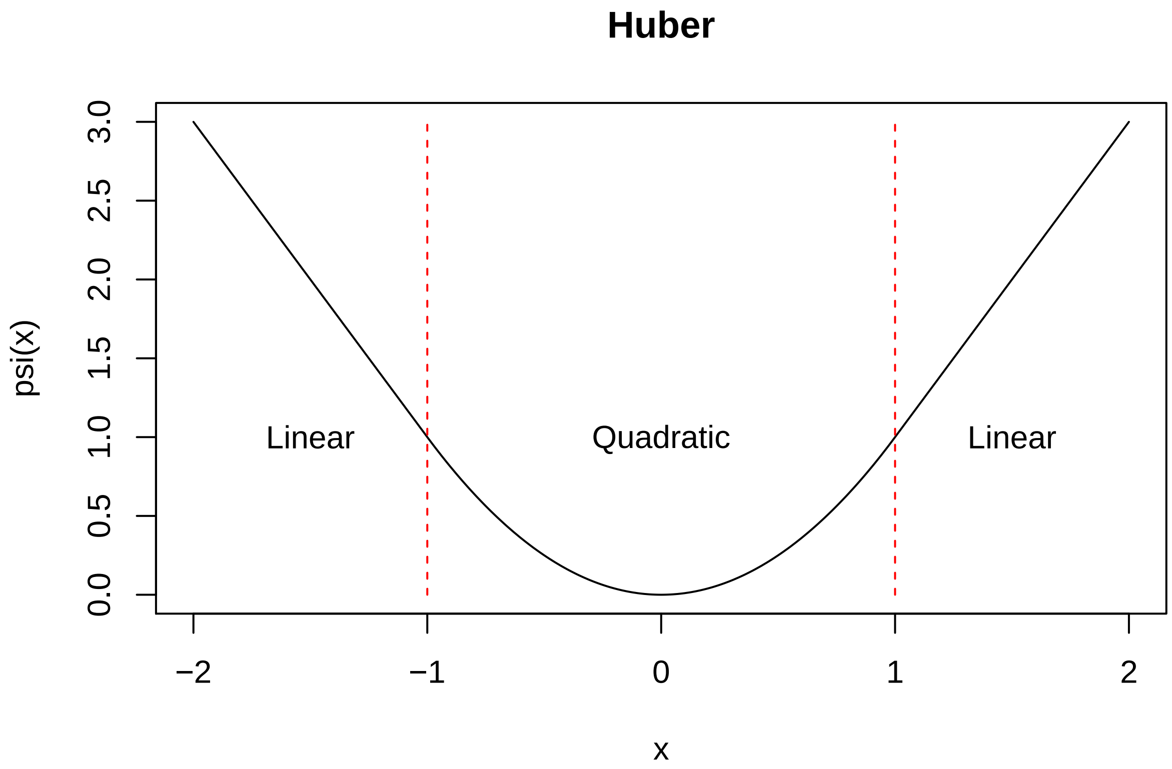
```
##      trump.words.new Freq
## 1121             the  203
## 91               and  150
## 1146              to  130
## 784               of  128
## 809              our   94
## 1244            will   87
## 568               in   79
## 550                i   65
## 1222              we   60
## 512             have   58
## 31                 a   54
## 1119            that   52
## 454              for   46
## 598               is   42
## 101              are   39
## 748               my   35
## 1133            this   33
## 789               on   30
## 1123           their   28
## 127               be   27
## 176               by   26
## 1217             was   26
## 84           america   25
## 259          country   25
## 467             from   25
```

- **3a.**

```
geo.mean <- function(x,y=1){
  for (i in 1:length(x)) {
    y <- y*x[i]
  }
  a <- length(x)
  return(y^(1/a))
}
geo.mean.two <- function(x,i=1,y=1){
  while(i <= length(x)){
    y <- x[i]*y
    i <- i+1
  }
  a <- length(x)
  return(y^(1/a))
}
```

- **3b.**

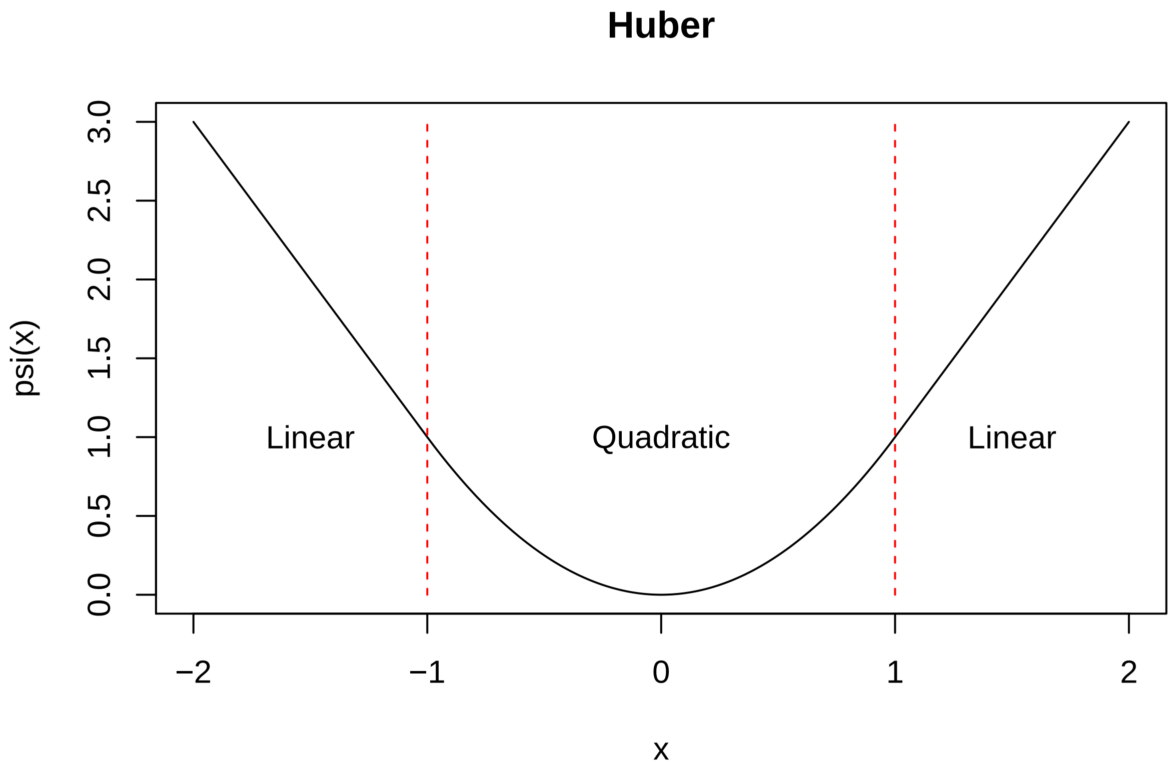
```
huber <- function(a){
  x <- seq(-2,2,0.01)
  y <- ifelse(abs(x) <= a,x^2,2*a*abs(x)-a^2)
  plot(x,y,type = ’l’,main = ’Huber’,ylab = "psi(x)")
  text(0,1,’Quadratic’)
  b <- a + 0.5
  text(b,1,’Linear’)
  text(-b,1,’Linear’)
  lines(x=c(a,a),y=c(0,3),lty=2,col=’red’)
  lines(x=c(-a,-a),y=c(0,3),lty=2,col=’red’)
}
huber(1)
```



- **3c.**

```
huber <- function(a){
  x <- seq(-2,2,0.01)
  y <- ifelse(abs(x) <= a,x^2,2*a*abs(x)-a^2)
  plot(x,y,type = 'l',main = 'Huber',ylab = "psi(x)")
  text(0,1,'Quadratic')
  b <- a + 0.5
  text(b,1,'Linear')
  text(-b,1,'Linear')
  mtext('Invented by the great Swiss statistician Peter Huber!',side = 3,outer = T,line = -1)
  lines(x=c(a,a),y=c(0,3),lty=2,col='red')
  lines(x=c(-a,-a),y=c(0,3),lty=2,col='red')
}
huber(1)
```

Invented by the great Swiss statistician Peter Huber!



- **3d.**

```
library(imager)
```

```
## Warning: package 'imager' was built under R version 4.0.3
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'imager'
```

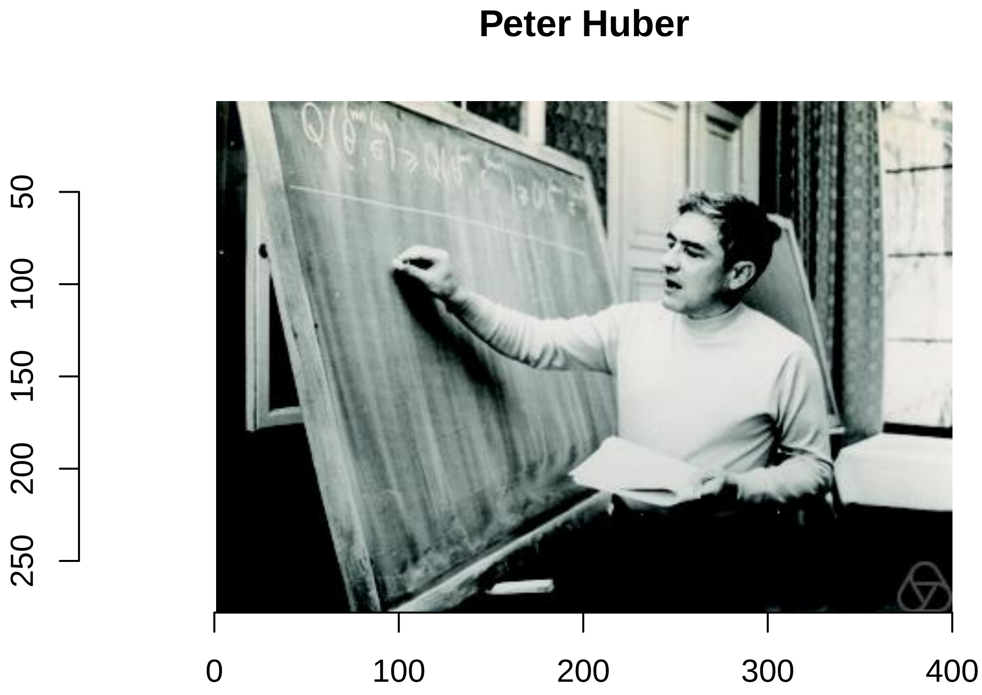
```
## The following object is masked from 'package:magrittr':
##
##     add
```

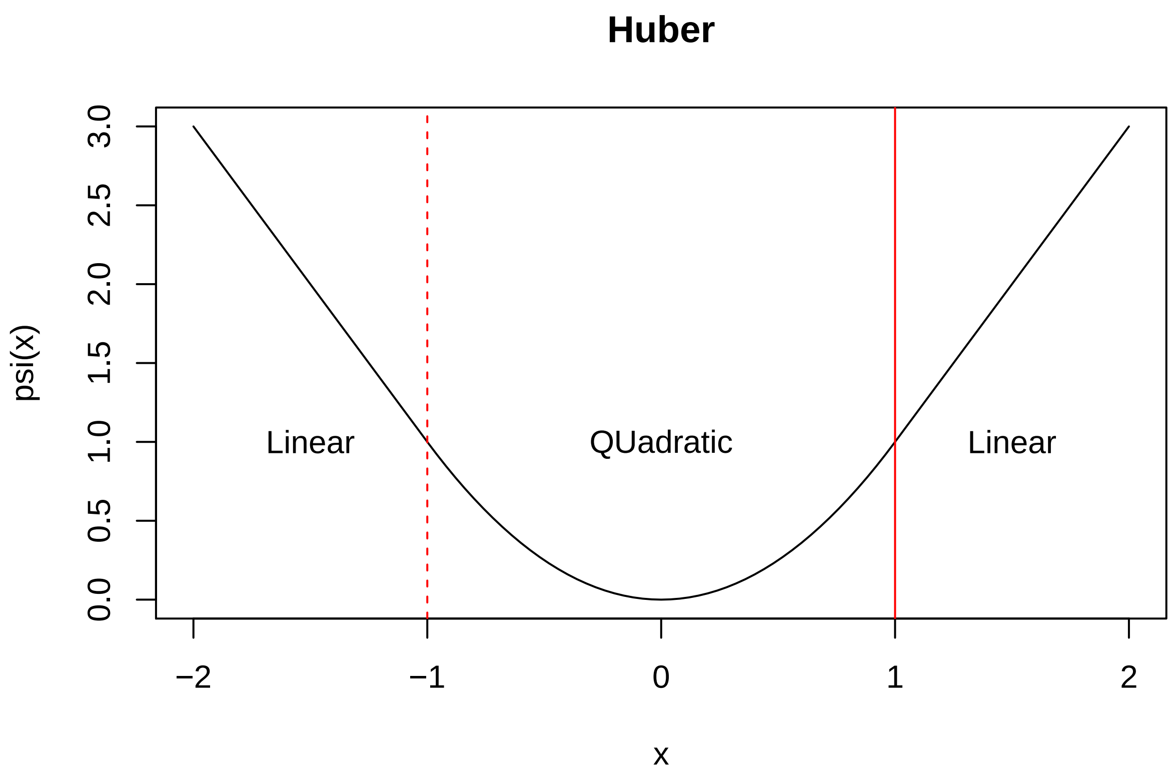
```
## The following objects are masked from 'package:stats':
##
##     convolve, spectrum
```

```
## The following object is masked from 'package:graphics':
##
##     frame
## The following object is masked from 'package:base':
##
##     save.image
```

```
huber = function(x_range = 2, a = 1) {
h = function(x) {
y = c()
for (x_ in x) {
if (abs(x_) <= a) {
y = append(y, x_ ^ 2) } else {
y = append(y, 2 * a * abs(x_) - a ^ 2) } }
return (y)
}
print("Invented by the great Swiss statistician Peter Huber!")
peter.huber.jpg = load.image("Peter_huber.jpg")
plot(peter.huber.jpg, main = "Peter Huber")
curve(h, from = -x_range, to = x_range, ylab = "psi(x)", xlab = "x", main = "Huber")
abline(v = -a, col = "red", lty = 2)
abline(v = a, col = "red", lty = 1)
text(x = -a - 0.5, y = 1, labels = "Linear")
text(x = 0, y = 1, labels = "QUadratic")
text(x = a + 0.5, y = 1, labels = "Linear") }
huber()
```

```
## [1] "Invented by the great Swiss statistician Peter Huber!"
```





- **4a.**

```
load('pros.Rda')
pros.dat.svi <- pros.dat[pros.dat$svi!=0, ]
pros.dat.svi.sd <- vector(length=ncol(pros.dat))
i = 1
while(i<=length(pros.dat.svi.sd)){
  pros.dat.svi.sd[i] <- sd(pros.dat.svi[,i])
  i <- i+1
}
pros.dat.svi.sd
```

```
## [1]  0.6707867  0.3275689  7.8715885  1.3545258  0.0000000  1.0452899  0.6015852
## [8] 25.7344498  0.9251229
```

- **4b.**

```
pros.dat.no.svi <- pros.dat[pros.dat$svi==0, ]
pros.dat.no.svi.sd <- vector(length=ncol(pros.dat))
i = 1
while(i<=length(pros.dat.svi.sd)){
  pros.dat.no.svi.sd[i] <- sd(pros.dat.no.svi[,i])
  i <- i+1
}
pros.dat.no.svi.sd
```

```
## [1]  1.0685730  0.4479291  7.3105907  1.4782007  0.0000000  1.0379398  0.7088414
## [8] 25.0667600  0.9646403
```

- **4c.**

```
for(i in 1:ncol(pros.dat)){
  pros.dat.no.svi.sd[i] <- sd(pros.dat.no.svi[,i])
  pros.dat.svi.sd[i] <- sd(pros.dat.svi[,i])
}
##pros.dat.svi.sd
pros.dat.svi.sd
```

```
## [1]  0.6707867  0.3275689  7.8715885  1.3545258  0.0000000  1.0452899  0.6015852
## [8] 25.7344498  0.9251229
```

```
##pros.dat.no.svi.sd
pros.dat.no.svi.sd
```

```
## [1]  1.0685730  0.4479291  7.3105907  1.4782007  0.0000000  1.0379398  0.7088414
## [8] 25.0667600  0.9646403
```

- **4d.**

```
##pros.dat.svi.sd
sapply(pros.dat.svi, function(x) sd(x))
```

```
##     lcavol    lweight        age       lbph        svi        lcp    gleason
##  0.6707867  0.3275689  7.8715885  1.3545258  0.0000000  1.0452899  0.6015852
##      pgg45       lpsa
## 25.7344498  0.9251229
```

```
##pros.dat.no.svi.sd
sapply(pros.dat.no.svi, function(x) sd(x))
```

```
##     lcavol    lweight        age       lbph        svi        lcp    gleason
##  1.0685730  0.4479291  7.3105907  1.4782007  0.0000000  1.0379398  0.7088414
##      pgg45       lpsa
## 25.0667600  0.9646403
```

- **5a.**

```
pros.dat.t.stat = mapply(function(x, y) {
return ((mean(x) - mean(y)) / sqrt(var(x) / length(x) + var(y) / length(y)))
}, pros.dat.svi, pros.dat.no.svi)
pros.dat.t.stat
```

```
##     lcavol    lweight        age       lbph        svi        lcp    gleason
##  8.0351008  1.8265652  1.1069463 -0.8828113        Inf  8.8354810  3.6193860
##      pgg45       lpsa
##  4.9417916  6.8578171
```

- **5b.**

```
pros.dat.df = mapply(function(x, y) {
v1 = (var(x) / length(x) + var(y) / length(y)) ^ 2
v2 = (var(x) / length(x)) ^ 2 / (length(x) - 1)
v3 = (var(y) / length(y)) ^ 2 / (length(y) - 1)
return (v1 / (v2 + v3))
}, pros.dat.svi, pros.dat.no.svi)
pros.dat.df
```

```
##   lcavol  lweight      age     lbph      svi      lcp  gleason    pgg45
## 51.17241 42.94873 30.21179 34.33734      NaN 31.75374 36.84259 31.28770
##     lpsa
## 33.02725
```

- **5c.**

```
pros.dat.p.val = pt(q=abs(pros.dat.t.stat), df=pros.dat.df, lower.tail = FALSE)
pros.dat.p.val
```

```
##       lcavol      lweight          age         lbph          svi          lcp
## 6.255198e-11 3.736044e-02 1.385266e-01 1.917386e-01          NaN 2.289876e-10
##      gleason        pgg45         lpsa
## 4.408147e-04 1.241127e-05 3.939533e-08
```

```
print(names(pros.dat.p.val[!is.nan(pros.dat.p.val)&pros.dat.p.val < 0.05]))
```

```
## [1] "lcavol"  "lweight" "lcp"     "gleason" "pgg45"   "lpsa"
```

```
print(names(pros.dat.p.val[!is.nan(pros.dat.p.val)&(pros.dat.p.val == min(pros.dat.p.val, na.rm = TRUE)
```

```
## [1] "lcavol"
```

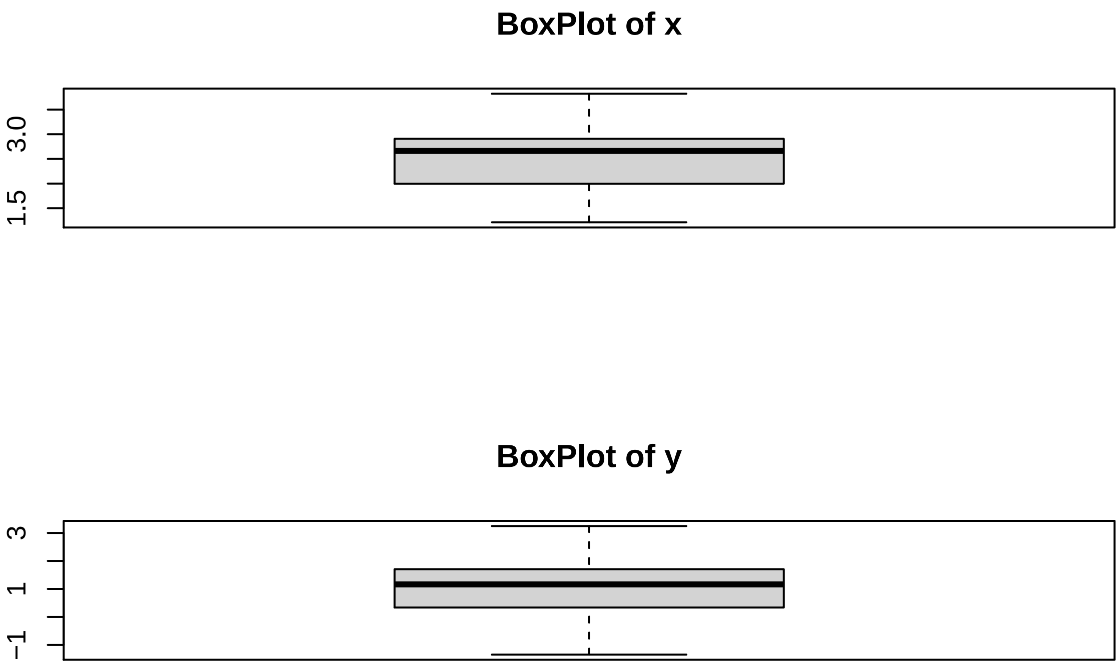
- **5d.**

```
my.t.test = function(x, y) {
statistics = (mean(x) - mean(y)) / sqrt(var(x) / length(x) + var(y) / length(y))
v1 = (var(x) / length(x) + var(y) / length(y)) ^ 2
v2 = (var(x) / length(x)) ^ 2 / (length(x) - 1)
v3 = (var(y) / length(y)) ^ 2 / (length(y) - 1)
df = v1 / (v2 + v3)
p.value = pt(q=abs(statistics), df = df, lower.tail = FALSE)
tt = c(statistics, df, p.value)
names(tt) = c("statistics", "df", "p.value")
return (tt)
}
result = mapply(FUN = my.t.test, pros.dat.svi, pros.dat.no.svi)
result
```

```
##                  lcavol     lweight        age       lbph svi          lcp
## statistics 8.035101e+00  1.82656523  1.1069463 -0.8828113 Inf 8.835481e+00
## df         5.117241e+01 42.94872991 30.2117862 34.3373409 NaN 3.175374e+01
## p.value    6.255198e-11  0.03736044  0.1385266  0.1917386 NaN 2.289876e-10
##                 gleason        pgg45         lpsa
## statistics 3.619386e+00 4.941792e+00 6.857817e+00
## df         3.684259e+01 3.128770e+01 3.302725e+01
## p.value    4.408147e-04 1.241127e-05 3.939533e-08
```

- **5e.**

```r
my.t.test = function(x, y) {
layout(matrix(c(1,1,2,2), 2,2, byrow = TRUE))
boxplot(x, main = "BoxPlot of x")
boxplot(y, main = "BoxPlot of y")
statistics = (mean(x) - mean(y)) / sqrt(var(x) / length(x) + var(y) / length(y))
v1 = (var(x) / length(x) + var(y) / length(y)) ^ 2
v2 = (var(x) / length(x)) ^ 2 / (length(x) - 1)
v3 = (var(y) / length(y)) ^ 2 / (length(y) - 1)
df = v1 / (v2 + v3)
p.value = pt(q=abs(statistics), df = df, lower.tail = FALSE)
tt = c(statistics, df, p.value)
names(tt) = c("statistics", "df", "p.value")
return (tt)
}
result = my.t.test(pros.dat.svi$lcavol, pros.dat.no.svi$lcavol)
```



```
result
```

```
##   statistics           df      p.value
## 8.035101e+00 5.117241e+01 6.255198e-11
```