STA104_hw4.R

pumad

2021-12-04

```
# Problem 1.1

before = c(250,50,80, 55,188)
after = c(240,48,72,47,230)

diff = before-after
diff
```

```
## [1]  10   2   8   8 -42
```

```
t.test(before, after, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  before and after
## t = -0.28307, df = 4, p-value = 0.7912
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -30.26296  24.66296
## sample estimates:
## mean of the differences
##                    -2.8
```

```
b = 5000
absdiff=abs(diff)
diffobs=mean(diff)
replicates=length(diff)
d=c()
p=c()
for( i in 1:b){
  permut=rbinom(replicates, 1, .5)
  positive=1*permut
  negative=(-1)*(1-permut)
  signed=positive+negative
  d[i]=mean(signed*absdiff)
  p[i]=(d[i]>=diffobs) +0
}

pvalue=sum(p)/b
pvalue
```

```
## [1] 0.5212
```

```
# Problem 1.2

b = 5000
absdiff=abs(diff)
diffobs=mean(diff)
replicates=length(diff)
d=c()
finalans = c()
for( i in 1:b){
  permut=rbinom(replicates, 1, .5)
  positive=1*permut
  negative=(-1)*(1-permut)
  signed=positive+negative

  a=c()
  for(j in (signed*rank(absdiff))){
    if(j>0){
      a[j]=j
    }
    ans=sum(a,na.rm = TRUE)
    d[i] = ans
  }
  finalans[i] = (d[i]>=10)+0
}
pvalue=sum(finalans)/b
pvalue
```

```
## [1] 0.228
```

```
# Problem 2

##Wilcoxon's signed-rank statistic
ybefore = c(89,90,87,98,120,85,97,110)
yafter = c(76,101,84,86,105,84,93,115)

wilcox.test(ybefore, yafter, alternative='two.sided', paired=TRUE)
```

```
##
##  Wilcoxon signed rank exact test
##
## data:  ybefore and yafter
## V = 27, p-value = 0.25
## alternative hypothesis: true location shift is not equal to 0
```

```
#Normal approximation
t.test(ybefore, yafter, paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  ybefore and yafter
## t = 1.2408, df = 7, p-value = 0.2547
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.623065 11.623065
## sample estimates:
## mean of the differences
##                       4
```

```
# Problem 3.1

x = c(57, 65, 70, 78)
y =c(120, 145, 153, 162)

b=10000
obs = cor(x, y)*sd(y)/sd(x)

obs
```

```
## [1] 1.978541
```

```
ans = c()
d=c()
for(i in 1:b){
  permuty = sample(y)
  ans[i]=cor(x, permuty) * sd(permuty)/sd(x)
  d[i]=(ans[i]>=obs) +0
}
pvalue = sum(d)/b

pvalue
```

```
## [1] 0.0418
```

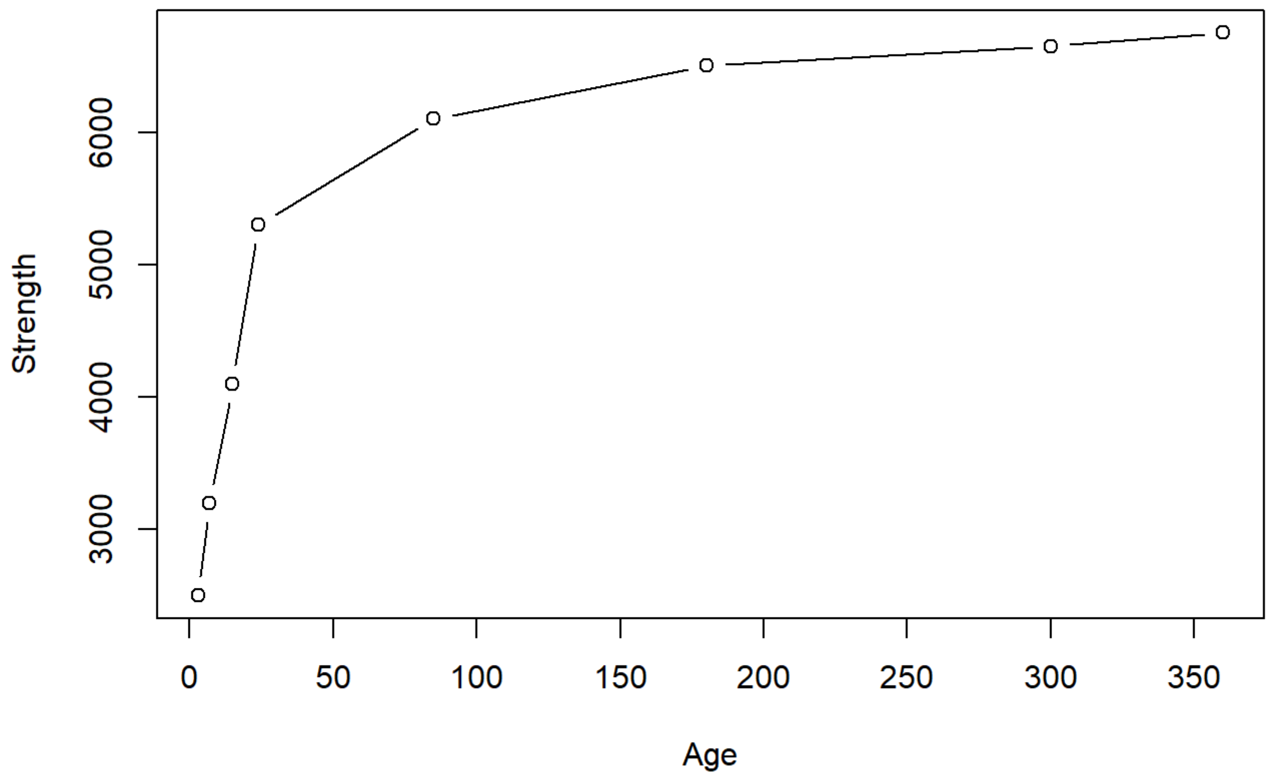
```
# Problem 3.2

##Spearman's r
rx = rank(x)
ry = rank(y)

spearobs=cor(rx,ry)
rs=c()
p=c()
for(i in 1:10000){
  rs[i]=cor(rx,sample(ry))
  p[i]=(rs[i]>=spearobs)+0
}
pvalue=sum(p)/10000
pvalue
```

```
## [1] 0.0428
```

```
##Kendall's t

obs = cor.test(x,y,method="k")$p.value
b = 10000
d = c()
p=c()
for( i in 1:b){
  d[i] = cor.test(sample(x),sample(y),method="k")$p.value
  p[i] = sum(d[i] >= obs)
}
pvalue = sum(p[i])/b

pvalue
```

```
## [1] 1e-04
```

```
# Problem 4

## 4.a
Age = c(3,7,15,24,85,180, 300, 360)
Strength = c(2500,3200,4100,5300,6100,6500,6650,6750)

plot(Age,Strength, type="b", col="black", lwd=1, pch=1, xlab="Age", ylab="Strength")
```




```
# Pearson's correlation
cor(Age, Strength)
```

```
## [1] 0.7999108
```

```
# Spearman's correlation
cor.test(Age,Strength,method="s")
```

```
##
##  Spearman's rank correlation rho
##
## data:  Age and Strength
## S = 0, p-value = 4.96e-05
## alternative hypothesis: true rho is not equal to 0
## sample estimates:
## rho
##   1
```

```
# Kendall's tau
cor.test(Age,Strength,method="k")
```

```
##
##  Kendall's rank correlation tau
##
## data:  Age and Strength
## T = 28, p-value = 4.96e-05
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
## tau
##   1
```

```
# Problem 5

df = data.frame("Nearby"=c(4,7), "Not_Nearby"=c(3,2),
                row.names = c("Low","Hight"),
                stringsAsFactors = FALSE)


fisher.test(df)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  df
## p-value = 0.5962
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0236709 5.2363497
## sample estimates:
## odds ratio
##  0.4057565
```

```
# Problem 6

df = data.frame("Missed_Second"=c(5,3), "Made_second"=c(14,8),
                row.names = c("Missed_first","Made_first"),
                stringsAsFactors = FALSE)

#(a)
mcnemar.test(matrix(c(5,3,14,8),2,2))
```

```
##
##  McNemar's Chi-squared test with continuity correction
##
## data:  matrix(c(5, 3, 14, 8), 2, 2)
## McNemar's chi-squared = 5.8824, df = 1, p-value = 0.01529
```

```
#(b)
chisq.test(matrix(c(5,3,14,8),2,2))$expected
```

```
## Warning in chisq.test(matrix(c(5, 3, 14, 8), 2, 2)): Chi-squared approximation
## may be incorrect
```

```
##          [,1]      [,2]
## [1,] 5.066667 13.933333
## [2,] 2.933333  8.066667
```