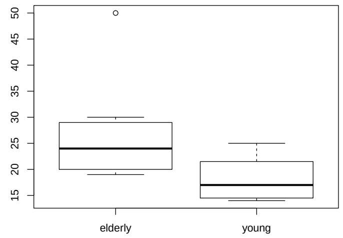
STAT 621: Nonparametric Statistics
Take Home Midterm Solutions

Problem 1 a) First a boxplot. Distributions are fairly similar and symmetric although there is one elderly outlier. Because of the small sample size, a distribution-free procedure seems necessary.

```
balance=read.table("balance.txt",header=T)
boxplot(FB˜Age, data=balance)
```



We can assume that elderly and young are independent samples. I'll use the Wilcoxon-Mann-Whitney test to compare medians (permutation test is also a valid choice). Let Δ be the difference in the median FB balance for elderly vs. young subjects. We test

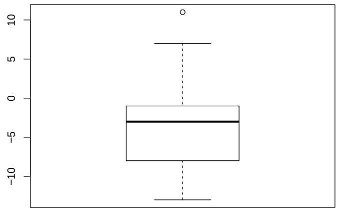
$$H_0 : \Delta = 0 \qquad \text{vs.} \qquad H_A : \Delta \neq 0$$

Below I separate the data into young and elderly and run the test. The Mann-Whitney U test statistic is 59, giving an approximate p-value of 0.02988 (note there are ties so this isn't exact). We'd reject the null and conclude that there is a difference in median FB balance.

```
FBelderly=balance$FB[1:9]
FByoung=balance$FB[10:17]
wilcox.test(FBelderly,FByoung,alt="two.sided",paired=F)

       Wilcoxon rank sum test with continuity correction

     data:  FBelderly and FByoung
     W = 59, p-value = 0.02988
     alternative hypothesis: true location shift is not equal to 0

     Warning message:
     In wilcox.test.default(FBelderly, FByoung, alt = "two.sided", paired = F) :
       cannot compute exact p-value with ties
```

b) Within an individual, the two balance measurements are paired. First a boxplot of the difference Z =SS-FB. The distribution looks a little skewed-right, with one large outlier. Even though the sample size is larger here it's still pretty small and assuming normality seems questionable.

```
boxplot(SS-FB)
```



For paired data, the options are the Sign test or Wilcox test (or permutation test of course). The Wilcox test requires that the distribution of differences is symmetric, so the Sign test might be a better choice. I'll run both below. With θ=median(Z) we are testing

$$H_0 : \theta = 0 \qquad \text{vs.} \qquad H_A : \theta \neq 0$$

The Wilcox procedure reports a test statistic of $W = 111.5$ and approximate (ties) p-value of $p = .02596$. The Sign test gives $S = 13$ and $p = .02127$. Both procedures result in rejection of the null. We conclude that median balance ability is different for FB vs. SS.

```
wilcox.test(FB,SS,alt="two.sided",paired=T)

     Wilcoxon signed rank test with continuity correction

    data:  FB and SS
    V = 111.5, p-value = 0.02596
    alternative hypothesis: true location shift is not equal to 0

    Warning messages:
    1: In wilcox.test.default(FB, SS, alt = "two.sided", paired = T) :
     cannot compute exact p-value with ties
    2: In wilcox.test.default(FB, SS, alt = "two.sided", paired = T) :
     cannot compute exact p-value with zeroes

library(BSDA)
SIGN.test(FB,SS,alt="two.sided")

    Dependent-samples Sign-Test

    data:  FB and SS
    S = 13, p-value = 0.02127
    alternative hypothesis: true median difference is not equal to 0
    (more output omitted)
```

Problem 2

a) Data are paired. Let μ_d be the mean difference in aggressive incidents between moon days and other days. We want to test

$$H_0 : \mu_d = 0 \qquad \text{vs.} \qquad H_A : \mu_d \neq 0$$

A reasonable test statistic would be the difference in the sample means, $\overline{Y}_d = \overline{Y}_{moon} - \overline{Y}_{other}$. Under the null, each patient's scores could have been observed on moon days or other days – this designation wouldn't matter. So the assignment of *moon* and *other* to each patient's score is random. To get the null distribution of $\overline{Y}_d$ we would:

- Enumerate all possible ways we could assign *moon* and *other* to each patient's scores. Note this is equivalent to assigning a $+$ or $-$ to each difference. There would be 2^{15} ways to do this.
- For each of these assignments, compute $\overline{Y}_d$.
- Each assignment is equally likely under H_0. The null distribution would list the unique values of $\overline{Y}_d$ and the proportion of times it took each unique value.
- Use the null distribution to find the p-value, the prob. that $\overline{Y}_d$ is greater than the absloute value of the observed statistic.

b) The ranodmization test is done below. Since the test stat is the mean difference, I note first that switching the labels just changes the sign of a patient's difference (moon-other). So I use a binomial to randomly identify the sign, then calculate $\overline{Y}_d$. Repeating this many times and summarizing gives the randomization dist.

Result: The p-value is approximately zero. Reject the null and conclude that there is a difference in mean aggressiveness between moon days and other days, at least for dementia patients.

```
attach(aggressive)
nreps=5000
ybar.d=rep(NA,nreps)
a=rep(1,15)
diffs=moon-other
for (i in 1:nreps)
{
  temp=rbinom(15,1,.5)
  diff.sign=a*temp+a*(temp-1)
  ybar.d[i]=mean(diffs*diff.sign)
}
#  table(round(ybar.d,2))/nreps   #  too many possible values
hist(ybar.d)
# calc the p-value
mean(moon-other)     #  observed ybar.d = 2.434
    [1] 2.434
sum((ybar.d>2.43))/nreps   # pval approx 0
    [1] 2e-04
```

c) Here we resample the data **paris**, with replacement, to create new bootstrap data sets. On each we compute $\overline{Y}_d$. This gives a sample of B values of $\overline{Y}_d$ from which we derive the bootstrap estimate of samping distribution. We find the CI using the quantiles of this distribution. Code is below. I found a 95% confidence interval to be $(1.69,\ 3.14)$.

```
nreps=5000
ybar.d=rep(NA,nreps)
diffs=moon-other
for (i in 1:nreps)
{
  temp=sample(1:15,15, replace=T)
  ybar.d[i]=mean(diffs[temp])
}
hist(ybar.d)                        # dist very symmetric
quantile(ybar.d, c(.025,.975))
       2.5%    97.5%
   1.693983 3.142700
```

Problem 3 **a)** If the dispersion of Y is larger, then the Y-values will be more spread out than the X-values. So **large** values of G will support H_A.

b) Smallest $G = n - 1$ when Y-values are all grouped together. Largest $G = m + n - 1$ when the largest and smallest values in the combined sample are Ys.

c) There are $\binom{m+n}{n}$ ways to order the Xs and Ys. So the probability of any ordering is $1/\binom{m+n}{n}$.

d) We have $m + n = 7$ and $n = 3$ so $G = 2, \ldots, 6$. There are $\binom{7}{3} = 35$ possible sequences. Think of this as having 7 slots that need to be filled in with 4-Xs and 3-Ys.

- $G = 2$: when the min and max ranks of the Ys are either (1,3), (2,4), (3,5), (4,6) or (5,7). Note this means that the ranks of the remaining Y is in the single place between the min and max ranks. So a total of 5 ways to have $G = 2$. $P(G = 2) = 5/35$.
- $G = 3$: when min and max Y-ranks are (1,4), (2,5), (3,6) or (4,7). In each case, the middle Y can be in two possible positions between the min and max Y-ranks. So a total of 8 ways to have $G = 3$, and $P(G = 3) = 8/35$.
- $G = 4$: Min and max Y-ranks must be (1,5), (2,6) or (3,7). Here for each there are 3 possible rankings for the middle Y, for a total of 9 sequences that will give $G = 4$. $P(G = 4) = 9/35$.
- $G = 5$: Min and max Y-ranks must be (1,6) or (2,7). For both there are 4 positions where the middle Y can fall. So $G = 5$ in 8 ways for $P(G = 5) = 8/35$.
- $G = 6$: Only one way, if min and max Y-ranks are (1,7), leaving 5 spots for the middle Y. So 5 ways for $P(G = 6) = 5/35$.

In summary,

g	2	3	4	5	6
$P(G = g)$	5/25	8/35	9/35	8/35	5/35

e) Drawing a sketch may help. For any g, there are $(m + n) - g$ ways to choose the min and max Y-ranks. Once these are chosen we can choose the ranks for the remaining Ys in $\binom{g-1}{n-2}$ ways. So,

$$P(G = g) = \frac{(m + n - g)\binom{g-1}{n-2}}{\binom{m+n}{n}}.$$

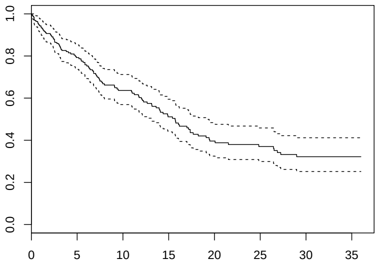
f) The observed value of G is $G = 7 - 1 = 6$. We reject the null if this is unusually large. So find $p = P(G \geq 6) = 5/35 = 0.143$. Fail to reject the null at $\alpha = 0.05$. We can't conclude that the Y dispersion is larger than the X dispersion.

Problem 4

a) Some summaries: 106 prisoners sent back to jail before 3 years; 26 stayed out the whole time; 62 left study before 3 years.

b) Plot and R code below. Comment....

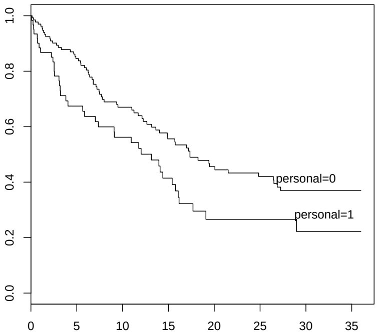
```
x=file.choose()
crime=read.table(x,header=T)

library(survival)
S.crime=Surv(crime$months, crime$delta)
a=survfit(S.crime~1)
plot(a)
summary(a)  # please don't turn in
```



c) Reject the null; p=.017. Conclude that survival curves are different for the two sets of prisoners. In particular, those not convicted of personal crimes have a better survial.

```
a=survdiff(S.crime~crime$personal, rho=0)
plot(survfit(S.crime~crime$personal))
a
```

```
                         N Observed Expected (O-E)^2/E (O-E)^2/V
      crime$personal=0 133       67     77.8      1.50       5.7
      crime$personal=1  61       39     28.2      4.14       5.7

 Chisq= 5.7  on 1 degrees of freedom, p= 0.017
```



d) The Cox PH model reaches the same conclusion. The test for the coefficient on the explanatory variable `personal` is significant $p = 0.0181$.

```
crime.ph1=coxph(S.crime~personal, data=crime)
summary(crime.ph1)


               coef exp(coef) se(coef)     z Pr(>|z|)
    personal 0.4787    1.6140   0.2025 2.364   0.0181 *
---

              exp(coef) exp(-coef) lower .95 upper .95
    personal     1.614     0.6196     1.085       2.4

  MORE OMITTED
```

e) Looks like all the terms are significant!

```
summary(crime.ph2)
Call:
coxph(formula = S.crime ~ personal + property + ctr.age, data = crime)

  n= 194, number of events= 106

             coef exp(coef) se(coef)      z Pr(>|z|)
personal  0.56915   1.76677  0.20521  2.774  0.00555 **
property  0.93577   2.54919  0.35088  2.667  0.00765 **
ctr.age  -0.06671   0.93546  0.01678 -3.976    7e-05 ***
---
```

Problem 5

Normal Sims: I wrote the funciton `fun1`, which appears at the end of these solutions, to do the simulations. For each θ value, I compute the power of both tests for a number of different sample sizes. I use the output to find the sample size that delivers the desired power for each test, at least approximately. The ratio of the two sample sizes is then my estimate of efficiencey. I rerun the function and compute efficiency for each different theta (I could of added another loop but this seemed simpler). Here’s an example of running the function for $\theta = 1$:

```
n=3:30
fun1(n,shift=1,nreps=500)
```

My efficiencies are summarized below; answers will vary. I find that the t-test is a bit more efficient than Wicox for Normal data.

```
# -------------------------------------------------------------------------
#      Power=.5                                        Power=.8
#  theta    n_t    n_U    eff=n_u/n_t             n_t     n_U     eff=n_u/n_t
#  .25      122   130         1.07                256     268      1.05
#  .5       30     34         1.13                 63     68       1.08
#  .75      14     16         1.14                 27     29       1.07
#  1.0      9      10         1.11                 17     17       1.0
# ----------------------------------------------------------------------
```

T Sims: Same idea here, just generate data from a $t(3)$ distribution. See my function `fun2` attached. My results are below. Here Wilcoxon is clearly more efficient when data are $t(3)$ distributed, which has much heavier tails than the Normal.

```
#----------------------------------------------------------------------------
#      Power=.5                                        Power=.8
#  theta    n_t    n_U    eff=n_u/n_t             n_t     n_U     eff=n_u/n_t
#  .25     346    200        .58                  710    385      .54
#   .5     74      53      .72                    180    104      .58
#  .75     35      25      .71                     74      49      .66
# 1.0      19      15      .79                     43      28      .65
# -----------------------------------------------------------------------
```

```
 fun1=function(n,shift,nreps) {
   power.t=rep(NA,length(n))
   power.U=rep(NA,length(n))
   for (j in 1:length(n))
     {
       pval.t=rep(NA,nreps)
       pval.U=rep(NA,nreps)
       for(i in 1:nreps)
         {
           X=rnorm(n[j])
           Y=rnorm(n[j])+shift
           pval.t[i]=t.test(X,Y,alt="two.sided",var.equal=T)$p.value
           pval.U[i]=wilcox.test(X,Y,alt="two.sided",paired=F)$p.value
           }
       power.t[j]=sum(pval.t<.05)/nreps
       power.U[j]=sum(pval.U<.05)/nreps
       }
   data.frame(n,power.t,power.U)
 }

fun2=function(n,shift,nreps) {
  power.t=rep(NA,length(n))
  power.U=rep(NA,length(n))
  for (j in 1:length(n))
  {
    pval.t=rep(NA,nreps)
    pval.U=rep(NA,nreps)
    for(i in 1:nreps)
    {
      X=rt(n[j],3)
      Y=rt(n[j],3)+shift
      pval.t[i]=t.test(X,Y,alt="two.sided",var.equal=T)$p.value
      pval.U[i]=wilcox.test(X,Y,alt="two.sided",paired=F)$p.value
    }
    power.t[j]=sum(pval.t<.05)/nreps
    power.U[j]=sum(pval.U<.05)/nreps
  }
  data.frame(n,power.t,power.U)
}
```