```
> source('Rallfun-v33')
> # USE library(WRS)
```

Wilcox Solutions

# MODERN STATISTICS FOR THE SOCIAL AND BEHAVIORAL SCIENCES: ANSWERS TO THE EXERCISES

ABSTRACT

This document provides more detailed solutions to all of the exercises

Wilcox (in press). Modern Statistics for the Social and Behavioral Sciences, 2nd Ed. New York: Chapman & Hall/CRC press.

For many of the solutions, details about how to solve the problems using R are included.

CHAPTER 2.

**1**.

```
> a=c(21,36,42,24,25,36,35,49,32)
> mean(a)

[1] 33.33333

> mean(a,tr=.2) # Or could use tmean(a)

[1] 32.85714

> median(a)

[1] 35
```

**2**.

```
> a=c(21,36,42,24,25,36,35,200,32)
> mean(a)

[1] 50.11111

> mean(a,tr=.2) # Or tmean(a) can be used.

[1] 32.85714

> median(a)
```

```
[1] 35
```

Sample mean is not resistant to outliers, its value can be substantially altered by even a single value.

**3**. The resistance of the 20% trimmed mean is 0.2, meaning that a change in more than 20% of the observations is required to generate a large change in its value.

In this case n=9, so $9 \times 0.2 = 1.8$, rounded down to the nearest integer is 1. This means that a large change in the 20% trimmed mean value requires altering at least 2 observations.

**4**. The resistance of the Median is 0.5, meaning that a change in more than 50% of the observations is required in order to make its value arbitrary large or small.

In this case n=9, so $9 \times 0.5 = 4.5$. This means that a large change in the median requires altering at least 5 observations.

**5**.

```
> b=c(6,3,2,7,6,5,8,9,8,11)
> a=c(21,36,42,24,25,36,35,49,32)
> mean(a, tr=.1) # Or use tmean(a,tr=.1)

[1] 33.33333

>  b=c(6,3,2,7,6,5,8,9,8,11)
>  mean(b)

[1] 6.5

> mean(b,tr=.2) #Or use tmean(b)

[1] 6.666667

> median(b)

[1] 6.5
```

**6**.

```
> c=c(250,220,281,247,230,209,240,160,370,274,210,204,243,251,190,200,130,150,177,475,
> mean(c)

[1] 229.1724

> mean(c,tr=.2)

[1] 220.7895
```

**7**. $f_1 = 5$, $f_2 = 8$, $f_3 = 20$, $f_4 = 32$, $f_5 = 23$ Using R:

```
> fx=c(5,8,20,32,23)
> x=c(1:5)
> n=sum(fx)
> n

[1] 88

> xbar=sum(x*fx)/n
> xbar  # the sample mean

[1] 3.681818
```

**8**.

```
> fx=c(12,18,15,10,8,5)
> n=sum(fx)
> n

[1] 68

> x=c(1:6)
> xbar=sum(x*fx)/n #The sample mean
> xbar

[1] 2.985294
```

**9**.

```
> d=c(21,36,42,24,25,36,35,49,32)
> var(d)

[1] 81
```

winvar(d)
returns
51.36111

**10**.

d=c(21,36,42,24,25,36,35,102,32)
winvar(d)
returns
51.36111

The Winsorized variance remained the same.

**11**.

Yes, because we shift the extreme values closer to the mean. This reduces the dispersion in the data. The mean squared distances from the mean decreases accordingly.

**12**. The variance has a sample breakdown point of 1/n, so a single observation can render its value arbitrarily large or small.

**13**.

The sample breakdown point of the 20% Winsorized variance is 0.2. In the case of n=25, this would be $25 \times 0.2 = 5$.

So, we need at to change at least 6 observation to render the Winsorized variance arbitrarily large.

**14**.

```
> e=c(6,3,2,7,6,5,8,9,8,11)
> var(e)

[1] 7.388889
```

winvar(e)
returns
1.822222

**15**.

```
> x=c(250,220,281,247,230,209,240,160,370,274,210,204,243,
+ 251,190,200,130,150,177,475,221,350,224,163,272,236,200,171,98)
> var(x)

[1] 5584.933
```

winvar(x)
returns
1375.606

**16**.

e=c(6,3,2,7,6,5,8,9,8,11)

idealf(e)

returns:

```
$ql
[1] 4.833333

$qu
[1] 8.083333
```

IQR=8.08-4.83=3.25

**17**. x=c(250,220,281,247,230,209,240,160,370,274,210,204,
243,251,190,200,130,150,177,475,221,350,224,163,272,236,200,171,98)

out(x)

returns:

```
$out.val
[1] 370 475 350  98
```

**18**.

The sample variance is

$$s^2 = \frac{n}{n-1}\sum (x - \bar{X})^2 \frac{f_x}{n}.$$

**19**.

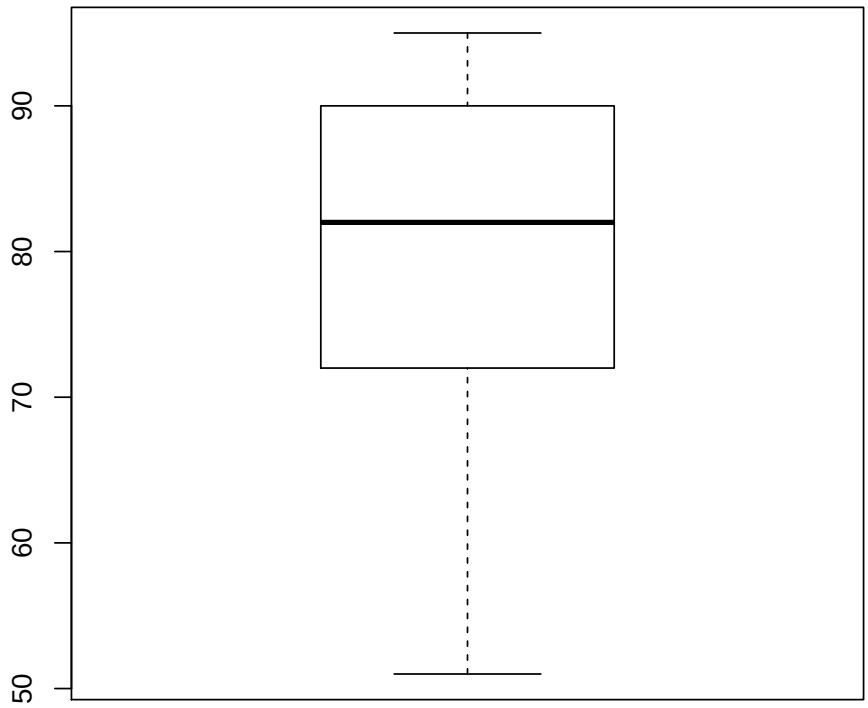
```
> x=c(0:5)
> fxn=c(.1,.2,.25,.29,.12,.04) #relative frequencies
> xbar=sum(x*fxn)
> sum((x-xbar)^2*fxn)

[1] 1.6675

>
> # This ignores the n/(n-1) term, which is
> #                     approximately eaual to one.
```

**20**.

```
> x=c(0:4)
> fxn=c(2.,.4,.2,.15,.05) #relative frequencies
> xbar=sum(x*fxn)
> sum((x-xbar)^2*fxn)

[1] 5.032
```

**21**.

```
> x=c(90,76,90,64,86,51,72,90,95,78)
> boxplot(x)
```




The command
outbox(x)
returns:

```
$out.val
numeric(0)

$out.id
NULL

$keep
 [1]  1  2  3  4  5  6  7  8  9 10

$n
[1] 10

$n.out
[1] 0
```

```
$cl
[1] 43.33333

$cu
[1] 118
```

which indicates that there are no outliers.

The command
out(x)
applies the MAD-median rule and returns:

```
$n
[1] 10

$n.out
[1] 1

$out.val
[1] 51

$out.id
[1] 6

$keep
[1]  1  2  3  4  5  7  8  9 10

$dis
 [1] 0.6744908 0.5058681 0.6744908 1.5176042 0.3372454 2.6136517 0.8431134
 [8] 0.6744908 1.0960475 0.3372454

$crit
[1] 2.241403
```

indicating one outlier, stored in $x[6]$ and equal to 51.

**22**.

(a) Two outliers are found.

(b) No outliers are found because now the number of outliers inflates the upper quartile, which in turn inflates the interquartile range.

**23**.

The boxplot has a sample break down point of 25%. The number of outliers it detects does not exceed 25% of the sample. For example, when we had 3 outliers with n=10, all outliers disappeared in Exercise 22.

**24**.

```
> m=c(0,0.12,.16,.19,.33,.36,.38,.46,.47,.60,.61,.61,.66,.67,.68,.69,
+     .75,.77,.81,.81,.82,.87,.87,.87,.91,.96,.97,.98,.98,1.02,
+     1.06,1.08,1.08,1.11,1.12,1.12,1.13,1.2,1.2,1.32,1.33,1.35,
+     1.38,1.38,1.41,1.44,1.46,1.51,1.58,1.62,1.66,1.68,1.68,
+     1.70,1.78,1.82,1.89,1.93,1.94,2.05,2.09,2.16,2.25,2.76,3.05)
```

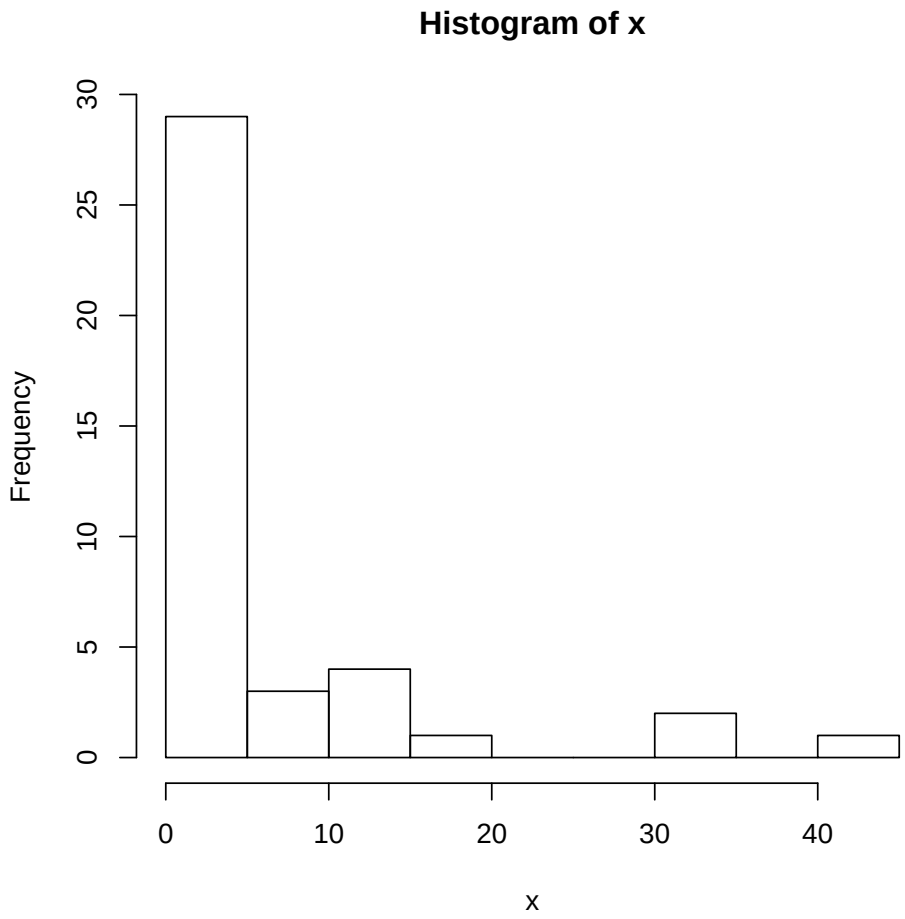
outbox(m)
indicates one outlier equal to 3.05

**25**.

The upper and lower quartiles in Figure 2.2 are approximately 125 and 50, respectively, so x is declared an outlier when
$x > 125 + 1.5(125 - 50)$
$x < 50 - 1.5(125 - 50)$.

**26**.

```
> x=c(0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
+   0,  0,  0,  1,  2,  2,
+ 2,  3,  3,  3,  6,  8,  9, 11, 11, 11, 12, 18, 32, 32, 41)
> hist(x)
```



A portion of the output from the command
outbox(x):

```
outbox(x)
$out.val
[1] 18 32 32 41

$out.id
[1] 37 38 39 40
```

The command
out(x)
returns:

```
 $n
[1] 40

$n.out
[1] 18

$out.val
 [1]  1  2  2  2  3  3  3  6  8  9 11 11 11 12 18 32 32 41
```

So all values not equal to zero are flagged as outliers.

CHAPTER 3.

**1**. Using R:

```
> x=c(0,1)  # x
> px=c(.7,.3)  #p(x)
> mu=sum(x*px) #population mean
> mu

[1] 0.3

> V=sum((x-mu)^2*px)  #population variance
> V

[1] 0.21
```

So,

$$P(X \leq \mu) = P(X \leq .3) = .7$$

**3**.

```
> x=c(1:5)  # x
> px=c(.15, 0.2, 0.3, 0.2, 0.15)  #p(x)
> mu=sum(x*px) #population mean
> mu

[1] 3

> V=sum((x-mu)^2*px)  #population variance
> V

[1] 1.6
```

So,

$$P(X \leq \mu) = P(X \leq 3) = .15 + .2 + .3 = .65$$

**4**. Would expect that this probability function has the smaller variance because again the mean is 3, but the likelihood of getting the most extreme values is smaller compared to the probability function in Exercise 3.

```
> x=c(1:5)  # x
> px=c(.1, 0.25, 0.3, 0.25, 0.1)  #p(x)
> mu=sum(x*px) #population mean
> mu

[1] 3

> V=sum((x-mu)^2*px)  #population variance
> V

[1] 1.3
```

**5**. Larger, because again the mean is 3 but the more extreme values (0 and 5) are more likely to occur.

**6**

```
> x=c(1:5)  # x
> px=c(.2, 0.2, 0.2, 0.2, 0.2)  #p(x)
> mu=sum(x*px)
> V=sum((x-mu)^2)*px
> Z=(x-mu)/sqrt(V)
> sum(Z*px)  #mean of standardized values

[1] 0

> sum((Z-0)^2*px) #Variance of standardized values

[1] 1
```

**7**.
(a) P(low income|age<30)=0.09/0.3=0.3
(b) P(age<30)=0.03+0.18+0.09=0.3
(c) P(high income|age<30)=0.03/0.3=0.1
(d) P(low income|age<30)=0.108/(0.018+0.108+0.054=0.18)=0.6

**8**.

Age and income are independent.

For example, we verify that
P(age=30-50|high income)=P(age=30-50)
P(age=30-50|high income)=0.052/(0.03+0.052+0.018)=0.52
P(age=30-50)=0.052+0.312+0.156=0.52.

**9**.
(a) P(yes)=(757+496)/3398=0.368
(b) P(yes|1)=757/(757+1071)=0.414
(c) P(1|yes)=757/(757+496)=0.60
(d) Not independent: P(yes|0)=496/(496+1074)=0.315 $\neq$ P(yes)=0.368
(e) P(yes and 1)= 757/3398=0.22
P(no and 0)= 1074/3398=0.31
0.22+0.31=0.53 or 1831/3398=0.53
(f) 1-P(yes and 1)=1-0.22=0.78
(g) P(yes or 1)=(757+1071+1074)/3398=0.854.

**10**.

The .1 quantile, say x, satisfies

$$P(X \leq x) = (x-1)\frac{1}{4-1} = .1,$$

which is the rectangular area under the probability density function; $x-1$ is the length of the base and 4-1=3 is the height. Solving for x yields x=1.3
For the median

$$P(X \leq x) = (x-1)\frac{1}{4-1} = .5,$$

so $x = 2.5$.
For the .9 quantile,

$$P(X \leq x) = (x-1)\frac{1}{4-1} = .9$$

yielding $x = 3.7$.

**11**.

Here we have a uniform distribution, the sample space extends from -3 to 2, which has length 5.
(a) Taken from the lower bound (-3),the value 1 extends over 4/5th of the sample sample space, so cumulative probability is 0.8. That is, the probability is (1-(-3))=4, the length of the base, times 1/5, the height of the rectangle corresponding to this probability function.
(b)The distance of -1.5 from -3 is 1.5, the height is 1/5, so the probability is 1.5/5=0.3.
(c) $P(X < 0) = .6$ so $P(X > 0) = 1 - 0.6 = 0.4$.

(d) $P(X < -1.2) = 1.8/5 = 0.36$, $P(X < 1) = .8$ so $P(-1.2 < X < 1) = 0.8 - 0.36 = 0.44$.
(e) $P(x = 1) = 0$. The area of a line is zero. (The base of the rectangle extends from 1 to 1.)

**12**. The median is the value $x$ such that

$$(x - (-3))\frac{1}{5} = .5$$

So the median is $x = -.5$ In a similar manner, the .25 quantile is the value $x$ such that

$$(x - (-3))\frac{1}{5} = .25,$$

which is $-1.75$. For the .75 quantile,

$$(x - (-3))\frac{1}{5} = .75,$$

$x = 1.5$.

**13**. This is a uniform distribution between -1 and 1.
For $P(X \leq c) = 0.9$, $(c - (-1))/2 = 0.9$. $c = 0.8$.
For $P(X \leq c) = 0.95$, $(c - (-1))/2 = 0.95$. $c = 0.9$.
For $P(X \geq c) = 0.99$, $(c - (-1))/2 = 0.01$. $c = -0.98$.
Note the reverse sign.

**14**. When computing any probability, the answer is based on the area of a rectangle having height 1/2.
(a) the distance from $-c$ to $c$ is 2c. So the area under the curve is 2c (1/2)=.9. That is, $c = .9$
(b) Proceed as in (a), $c = .95$
(c) Proceed as in (a), $c = .99$

**15**.
(a) Zero. The distance from 12 to 12 is zero. That is, the area of a line is zero.
(b) (5-0)(1/20)=.25
(c) (20-10)(1/20)=.5.

**16**.
(a) Zero, the area of a line is zero.
(b) For $P(X10)$, (10-0)/60=1/6
(c)For $P(X20)$, (20-0)/60=1/3. 1-1/3=2/3
(d) For $P(10X < 20)$, (20-10)/60=1/6.

**17**. $(c - 0)(1/60) = .8$, so $c = 48$.

**18**.
(a) $P(Z1.5) = 1 - 0.933 = 0.067$. (b) $P(Z2.5) = 0.006$
(c) $P(Z < 2.5) = 0.006$
For continuous variables, there is no distinction between $<$ and $\leq$.
(d) $P(-1Z1) = P(Z - 1) - P(Z1) = 0.841 - 0.158 = 0.683$.

**19**.
(a) $P(Z \leq 0.5) = 0.691$
(b) $P(Z > -1.25) = 1 - 0.105 = 0.895$

(c) $P(-1.2 \leq Z \leq 1.2) = P(\leq -1.2) - P(Z \leq 1.2) = 0.884 - 0.115 = 0.769$
(d) $P(-1.8 \leq Z \leq 1.8) = P(Z \leq -1.8) - P(Z \leq 1.8) = 0.964 - 0.0359 = 0.928$

**20**.
(a) $P(Z \leq -0.5) = 0.308$
(b) $P(Z > 1.2) = 0.884$
(c) $P(Z < 2.1) = 1 - 0.982 = 0.018$
(d) $P(-0.28 \leq Z \leq 0.28) = P(Z \leq -0.28) - P(Z \leq 2.8) = 0.61 - 0.389 = 0.221.$

**21**.
(a) For $P(Z \leq c) = 0.0099$, $c = -2.33$
(b) For $P(Z \leq c) = 0.9732$, $c = 1.93$
(c) $P(Z > c) = 0.5691$ equals $P(Z < c) = 1 - 0.5691. c = -0.175$
For $P(-c < Z < c) = 0.2358$. $P(z < c) = (1 + 0.2358)/2 = 0.6179$. $c = 0.3$.

**22**.
$P(Z > c) = 0.0764$ equals $P(Z < c) = 1 - 0.0764$. $c = 1.43$
$P(Z > c) = 0.5040$ equals $P(Z < c) = 1 - 0.5040$. $c = -0.01$
$P(-c < Z < c) = 0.9108$. $P(z < c) = (1 + 0.9108)/2 = 0.9554$, $c = 1.7$
$P(-c < Z < c) = 0.8$. $P(z < c) = (1 + 0.8/2) = 0.9$. $c = 1.285$.

**23**.
(a) Z=(40-50)/9=-1.111, P(Z $\leq$ -1.111)=0.134.
(b) Z=(55-50)/9=0.555, P(Z $\leq$ 0.555)=0.71.
(c) Z=(60-50)/9=1.111, 1-P(Z $\leq$ 1.111)=1-0.866=.134.
(d) P(Z $\leq$ 1.111)-P(Z $\leq$ -1.111)=0.866-0.134=0.732.

**24**.
(a) Z=(22-20)/9=0.2222, P(z$le$0.222)=0.587
(b) Z=(17-20)/9=-0.3333, 1-P(z$\leq$-0.333)=0.631
(c) Z=(15-20)/9=-0.5555, 1-P(z$\leq$-0.555)=0.711
(d) Z=(38-20)/9=2 Z=(2-20)/9=-2 P(z$\leq$ 2)-P(z$\leq$ -2)=0.977-0.022=0.955

**25**.
(a) Z=(0.25-0.75)/0.5=-1, P(z$\leq$-1)=0.158
(b) Z=(0.9-0.75)/0.5=0.3, 1-P(z$\leq$ 0.3)=0.382
(c) Z=(0.5-0.75)/0.5=-0.5, Z=(1-0.75)/0.5=0.5
P(z$\leq$0.5)-P(z$\leq$-0.5)=0.691-0.308=0.383
(d) Z=(0.25-0.75)/0.5=-1, Z=(1.25-0.75)/0.5=1
P(z$\leq$ 1)-P(z $\leq$ -1)=0.841-0.158=0.683.

**26**. The question asks: how many standard deviations (c) above and below the mean capture the middle 95% of the distribution? c will correspond to the 0.975 quantile and $-c$ to 0.025. From Table 1. c=1.96.
Alternative solution: standardize, that is compute a Z score.

$$P(\mu - c\sigma \leq X \leq \mu + c\sigma) = P(-c \leq Z \leq c) = .95$$

so $c = 19.6$.

**27**. Proceeding as in the answer to Exercise 26,

$$P(\mu - c\sigma \leq X \leq \mu + c\sigma) = P(-c \leq Z \leq c) = .80,$$

so $c$ is the .9 quantile, which is 1.28.

**28**. Z=(78-68)/10=1
1-P(Z<1)=1-0.84=0.16.

**29**. The top 5% correspond to the 0.95 quantile. From Table 1, z=1.64

$$c = \mu + 1.64\sigma$$

so $c = 68 + 101.645 = 84.45$.

**30**.
The standard score of 62 is z=(62-58)/3=1.333
From Table 1, P(Z<1.333)=0.908
Hence P(Z>1.333)=1-0.908=0.092.

**31**.First step is to calculate the standardized scores for the interval:
Z=(115000-100000)/10000=1.5
Z=(85000-100000)/10000=-1.5
From Table 1: P(Z<1.5)-P(Z<-1.5)=0.933-0.066=0.867.

**32**. We are looking for the probability of observing X>0 (not losing money), which is 3 standard deviations from the mean. From Table 1:
P(Z<3)=0.998. Hence P(Z>3)=1-0.998=0.002.

**33**. The goal is to compute the probability of observing a salary between 1 standard deviation above and below the mean (40000-60000). From Table 1:
P(Z<1)-P(Z<-1)=0.841-0.158=0.683.

**34**. We are asked to compute the probability of observing a salary between 2 SDs above and below the mean (350-550). From Table 1:
P(Z<2)-P(Z<-2)=0.977-0.022=0.955.

**35**. Z=(260-230)/25=1.2 From Table 1:
P(Z<1.2)=0.885. Hence P(Z>1.2)=1-0.885=0.115.

**36**. Z=(20000-14000)/3500=1.714 From Table 1:
P(Z<1.714)=0.956. Hence P(Z>1.714)=1-0.956=0.044.

**37**. Yes. The median is robust against shifts in the tail. The mean is sensitive to tail changes because extreme values have a large impact on the mean but not the median.

**38**. When the tails become slightly heavier, they introduce more values that are far from the mean. Because the variance is the expected value of the squared distance from the mean, the effect of these extreme values on the variance is disproportionately large.

**39**. No. Because the standard deviation no longer corresponds to a known probability density function. Using the normal PDF can give poor probability coverage when non-normal distributions are considered.

**40**. No, if it is heavy tailed, the standard deviation will be inflated, giving larger probability coverage than expected.

**41**. Yes, two skewed distributions in the opposite direction can have the same mean and variance, but little overlap.

**42**. Yes, if it is heavy tailed.

**43**.

$$\mu = 0.2 \times 1 + 0.4 \times 2 + 0.3 \times 3 + 0.1 \times 4 = 2.3.$$

$$\sigma^2 = 0.2 \times 1.3^2 + \times 0.3^2 + 0.3 \times 0.7^2 + 0.1 \times 1.7^2 = 0.81.$$

$$\sigma = \sqrt{0.81} = 0.9$$

2.3 ± 0.9=(1.4,3.1) $P(\mu - \sigma \leq X \leq \mu + \sigma) = 0.7$.

**44**. P=0.75, q=0.25 Can't use Table 2. Use the binomial probability function or can use R. Using R,
All 5 lose money

```
> dbinom(5,5,.75)

[1] 0.2373047
```

All 5 make money,

```
> dbinom(0,5,.75)

[1] 0.0009765625
```

Exactly four make money, which means only one loses money:

```
> dbinom(1,5,.75)

[1] 0.01464844
```

At least two lose money, which is one minus the probability that one or more lose money:

```
> 1-pbinom(1,5,.75)

[1] 0.984375
```

**45**.
(a) From Table 2, with P=0.4, n=25 and at most 10 successes, $P(X \leq 10) = 0.586$.
(b) 11 at most successes, $P(X \leq 11) = 0.732$
(c) At least 10 successes equals 1 minus at most 9 successes: $1 - P(X \leq 9) = 1 - 0.425 = 0.575$.
(d) 1 minus at most 8 successes. That is, $1 - P(X \leq 8) = 1 - 0.274 = 0.726$.

**46**.
$\mu = np = 0.4(25) = 10$
VAR(X)=$npq = 25(0.4)(0.6) = 6$
$E(\hat{p}) = p = 0.4$
VAR($\hat{p}$) = $pq/n$=0.4(0.6)/25=0.0096.

**47**.
Let $D$ represent the event that someone has the illness and let $Y$ indicate that the test indicates that someone does indeed have the illness. The notation $D$ means that someone does not have the illness and $Y$ means that the test indicates no illness. The goal is to determine $P(D|Y)$. We are given that $P(Y|D) = 0.9$, $P(\ Y|\ D) = 0.95$ and $P(D) = 0.02$. Note that $P(Y|\ D) = 1 - 0.95 = 0.05$. Bayes' theorem states that $P(D|Y) = P(Y|D)P(D)/P(Y)$, so

we first must determine $P(Y)$, which is the sum of two mutually exclusive events: $P(Y) = P(Y\text{and D}) + \text{P(Yand D)}$. But $P(Y\text{and D}) = \text{P(Y|D)P(D)} = 0.02(0.9) = .018$. Similarly, $P(Y\text{and D}) = \text{P(Y| D)P( D)} = .05(0.98) = 0.49$, so $P(Y) = 0.018 + 0.049 = 0.067$. So Bayes' theorem yields $P(D|Y) = 0.9(0.02)/0.067 = 0.27$.

CHAPTER 4

**1**. The 95% CI represents a range of values that contains a population parameter with a 0.95 probability. This range is determined by the values that correspond to the 0.025 and 0.975 quartiles of the sampling distribution of the test statistic.

**2**. $c$ is the 1-$\alpha/2$ quantile of the standard normal distribution. From Table 1 or R function qnorm:
For CI of 0.8, the 0.9 quantile is 1.281
For CI of 0.92, the 0.96 quantile is 1.750
For CI of 0.98, the 0.99 quantile is 2.326

**3**. We that $1 - \alpha/2 = .975$, so from Table 1, $c = 1.96$
$45 \pm 1.96\frac{5}{\sqrt{25}} = (43.04, 46.96)$.

**4**. Using R:

```
> 45-qnorm(.995)*5/sqrt(25)

[1] 42.42417

> 45+qnorm(.995)*5/sqrt(25)

[1] 47.57583
```

That is, the .99 confidence interval is (42.42417, 47.57583).

**5**. We have that $1 - \alpha/2 = .975$, the variance is given, so from Table 1, $c = 1.96$. (If the variance is not known, used Table 4, Student's t.) The .95 confidence interval is
$1150 \pm \frac{25}{\sqrt{36}} = (1141.84, 1158.16)$
The 95% CI for $\mu$ does not contain 1200, so the claim seems unreasonable.

**6**.
(a) $65 \pm \frac{22}{\sqrt{12}} = (77.44, 52.55)$
(b) $185 \pm \frac{10}{\sqrt{22}} = (189.17, 180.82)$
(c) Using R:

```
> 19-qnorm(.975)*30/sqrt(50)

[1] 10.68458

> 19+qnorm(.975)*30/sqrt(50)

[1] 27.31542
```

**7**.Random sampling requires:
1. That all observations are sampled from the same distribution. That is, the same probability function applies for every observation.

2. That the sampled observations are independent, meaning that the probability of sampling a given observation does not alter the probabilities associated with another sample.

**8**. The variance of the sampling distribution is given by $\sigma^2/n$, so in this case it is 8/10.

**9**. The population mean and variance can be computed using R as follows:

```
> x=c(1:4)
> px=c(0.2,   0.1,   0.5,   0.2)
> mu=sum(x*px)  # Population mean
> sigsq=sum((x-mu)^2*px) # Population variance
> mu

[1] 2.7

> sigsq

[1] 1.01
```

**10**. The expected value of the sample mean equals the population mean, so if you average 1000 sample means the grand average should be approximately equal $\mu$, in this case, 2.7. So, $E(\bar{X}) = 2.7$ and the variance of $\bar{X}$ is 1.01/12=.084.

**11**. Based on the same principle as in Exercise 10, the expected value of the sample variance equals to the population variance, so if you average 1000 sample variances, the result should be approximately equal $\sigma^2$, which in this case is 1.01.

**12**.

```
> x=c(2,6,10,1,15,22,11,29)
> s.sq=var(x) # sample variance
> s.sq

[1] 94.28571
```

So the variance of the sample mean is estimated by $s^2/n = 94.286/8 = 11.786$ and the standard error is estimated to be $\sqrt{11.785} = 3.43$.

**13**. The estimate of $\mu$ in this case would be based on a single observation = 32.

With a single observation, it is not possible to estimate the standard error because there is no variance in the sample.

As the sample size increases, the variance of the sampling distribution (the squared standard error) decreases. Note that n is in the denominator of the standard error. Lower variance in the sampling distribution means a smaller standard error, roughly meaning less error when estimating the population mean with the sample mean.

**14**.

```
> b=c(450,12,52,80,600,93,43,59,1000,102,98,43)
> n=12
> var(b)  #sample variance

[1] 93663.52

> var(b)/n # estimate of squared standard error.

[1] 7805.293
```

**15**.
b=c(450,12,52,80,600,93,43,59,1000,102,98,43)
> out(b)
$out.val
450 600 1000
These outliers substantially inflate the estimate of the standard error because they inflate the sample variance.

**16**.

```
> c=c(6,3,34,21,34,65,23,54,23)
> n=9
> var(c) # sample variance

[1] 413.9444

> var(c)/n # estimate of the squared standard error.

[1] 45.99383
```

**17**. No. An accurate estimate of the standard error requires independence among sampled observations.

**18**. The variance of the mixed normal is 10.9, so the squared standard error for a sample of 25 would be 10.9/25=0.436, compared to 1/25=0.04 when sampling from a standard normal distribution.

This means that under small departures from normality, the standard error can inflate more than 10 fold. The inflation greatly increases error, and the length of CIs.

**19**. When sampling from a non-normal distribution, the sampling distribution of the sample mean is not necessarily determined by the mean and variance. In particular, the sampling distribution is not exactly normal and more importantly, confidence intervals based on Student's T distribution can be highly inaccurate.

**20**. Determine Z and consult Table 1, or use R. The R function pnorm(x,mean,sd) determines $P(X \leq x)$. By default, a standard normal distribution is assumed if the arguments mean and sd are not specified.

For $P(\bar{X} \leq 29)$,

```
> pnorm(29,30,2/sqrt(16))
```

```
[1] 0.02275013
```

For $P(\bar{X} > 30.1)$, this is the same as 1-$P(\bar{X} < 30.1)$,

```
> 1-pnorm(30.5,30,2/sqrt(16))

[1] 0.1586553
```

$P(29 < \bar{X} < 31)$ is

```
> pnorm(31,30,2/sqrt(16))-pnorm(29,30,2/sqrt(16))

[1] 0.9544997
```

**21**. The mean is 5 and the standard error is 5/5=1. So using pnorm as done in Exercise 20,
(a)

```
> pnorm(4,5,1)

[1] 0.1586553
```

(b)

```
> 1- pnorm(7,5,1)

[1] 0.02275013
```

(c)

```
> pnorm(7,5,1)-pnorm(3,5,1)

[1] 0.9544997
```

**22**. $Z = \sqrt{n}(\bar{X} - \mu)/\sigma = \sqrt{16}(95000 - 100000)/10000 = -2$. From Table 1, the probability is .0227. Using R:

```
> pnorm(95000,100000,10000/sqrt(16))

[1] 0.02275013
```

**23**. Compute Z scores for each value and consult Table 1. Or use R:

```
> pnorm(97500,100000,10000/sqrt(16))

[1] 0.1586553

> pnorm(102500,100000,10000/sqrt(16))
```

```
[1] 0.8413447
```

**24**. Compute Z scores for each value and consult Table 1. Or use R.

```
> pnorm(700,750,100/sqrt(9))-pnorm(800,750,100/sqrt(9))

[1] -0.8663856
```

**25**. (a) $Z = \sqrt{16}(34 - 36)/5 = -1.6$, so $P(\bar{X} < 34) = P(Z < -1.6) = .055$ Using R:

```
> pnorm(34,36,5/4)

[1] 0.05479929
```

(b)

```
> pnorm(37,36,5/4)

[1] 0.7881446
```

(c)

```
> 1-pnorm(33,36,5/4)

[1] 0.9918025
```

(d)

```
> pnorm(37,36,5/4)-pnorm(34,36,5/4)

[1] 0.7333453
```

**26**. (a) $Z = \sqrt{25}(24 - 25)/3 = -1.666$, so $P(\bar{X} < 24) = P(Z < -1.666) = .047$. Using R

```
> pnorm(24,25,3/5)

[1] 0.04779035
```

(b)

```
> pnorm(26,25,3/5)

[1] 0.9522096
```

(c)

```
> 1-pnorm(24,25,3/5)

[1] 0.9522096
```

(d)

```
> pnorm(26,25,3/5)-pnorm(24,25,3/5)

[1] 0.9044193
```

**27**. Heavy tailed distributions generally yield long CI for the mean because their large variance inflates the SE. Central limit theorem does not remedy this problem.

**28**.Light tailed, symmetric distributions result in relatively accurate probability coverage even with small sample sizes. Central limit theorem works relatively well in this case.

**29**. $c$ is the 1-$\alpha/2$ quantile of a T distribution with n-1 degrees of freedom. Here, look up $c$ from Table 4, 0.975 quantile with 9 df; $c$=2.262157. The R function qt(p,df) computes the pth quantile.
(a) Using R, the ends of the confidence intervals are:

```
> 26-qt(.975,9)*9/sqrt(10)

[1] 19.56179

> 26+qt(.975,9)*9/sqrt(10)

[1] 32.43821
```

(b) Now df=18-1=17.

```
> 132-qt(.975,17)*20/sqrt(18)

[1] 122.0542

> 132+qt(.975,17)*20/sqrt(18)

[1] 141.9458
```

(c)

```
> 52-qt(.975,24)*12/sqrt(25)

[1] 47.04664

> 52+qt(.975,24)*12/sqrt(25)

[1] 56.95336
```

**30**. $c$ is the $1-\alpha/2 = .995$ quantile of a T distribution with n-1 degrees of freedom. (a)

```
> 26-qt(.995,9)*9/sqrt(10)

[1] 16.75081

> 26+qt(.995,9)*9/sqrt(10)
```

```
[1] 35.24919
```

(b)

```
> 132-qt(.995,17)*20/sqrt(18)

[1] 118.3376

> 132+qt(.995,17)*20/sqrt(18)

[1] 145.6624
```

(c)

```
> 52-qt(.995,24)*12/sqrt(25)

[1] 45.28735

> 52+qt(.995,24)*12/sqrt(25)

[1] 58.71265
```

**31**.

```
> x=c(77,87,88,114,151,210,219,246,253,262,296,299,306,376,428,515,666,1310,2611)
> t.test(x)

        One Sample t-test

data:  x
t = 3.2848, df = 18, p-value = 0.004117
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 161.5030 734.7075
sample estimates:
mean of x
 448.1053
```

**32**.

```
> y=c(5,12,23,24,18,9,18,11,36,15)
> t.test(y)

        One Sample t-test

data:  y
t = 6.042, df = 9, p-value = 0.0001924
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 10.69766 23.50234
sample estimates:
mean of x
     17.1
```

**33**. Heavy tailed distributions have two negative consequences when using T. First, they alter the cumulative probabilities of the T distribution. For symmetric distributions, the result is that the probability coverage, when computing a CI, is larger than intended. For example, when computing a 0.95 CI will, the actual probability coverage can exceed 0.99.

Second, they inflate the SE, due the larger frequency of extreme values in the tails, which leads to a relatively long CI.

When distributions are skewed, T becomes skewed, off centered (the mean value of $T$ is not equal to zero due to the dependency that is created between the mean and SD); its quantiles do not correspond to the quantiles in Table 4. This results in highly inaccurate probability coverage for CIs.

**34**. When the variance is not known but estimated using the available data, the actual distribution of $T$ can differ markedly from Students t distribution, which in turn can result in inaccurate confidence intervals. Practical concerns can occur with a sample size as large as 200.

**35**. (a) The number of observations trimmed from each tail is $g = .2(24)$ rounded down to the nearest integer, which is 4. So the total number trimmed is $2g = 8$ and the degrees of freedom are 24-8-1=15. Using R, the ends of the confidence intervals are as follows:

```
> 52-qt(.975,15)*sqrt(12)/(.6*sqrt(24))

[1] 49.48806

> 52+qt(.975,15)*sqrt(12)/(.6*sqrt(24))

[1] 54.51194
```

(b) DF=21

```
> 10-qt(.975,21)*sqrt(30)/(.6*sqrt(36))

[1] 6.835968

> 10+qt(.975,21)*sqrt(30)/(.6*sqrt(36))

[1] 13.16403
```

(c)

```
> 16-qt(.975,7)*sqrt(9)/(.6*sqrt(12))

[1] 12.58696

> 16+qt(.975,7)*sqrt(9)/(.6*sqrt(12))

[1] 19.41304
```

**36**. (a) Proceed as in Exercise 35. The number of observations trimmed from each tail is $g = .2(24)$ rounded down to the nearest integer, which is 4. So the total number trimmed is $2g = 8$ and the degrees of freedom are 24-8-1=15. Using R, the ends of the confidence intervals are as follows:

```
> 52-qt(.995,15)*sqrt(12)/(.6*sqrt(24))

[1] 48.52727

> 52+qt(.995,15)*sqrt(12)/(.6*sqrt(24))

[1] 55.47273
```

(b) DF=21

```
> 10-qt(.995,21)*sqrt(30)/(.6*sqrt(36))

[1] 5.692224

> 10+qt(.995,21)*sqrt(30)/(.6*sqrt(36))

[1] 14.30778
```

(c)

```
> 16-qt(.995,7)*sqrt(9)/(.6*sqrt(12))

[1] 10.94893

> 16+qt(.995,7)*sqrt(9)/(.6*sqrt(12))

[1] 21.05107
```

**37**. x=c(77,87,88,114,151,210,219,246,253,262,296,299,306,376,428,515,666,1310,2611) The R function trimci(x) returns
$ci
160.3913 404.9933.

**38**. Using a 20% trimmed mean, the length of the confidence interval is 244.6. Using the mean it is 573.2, which is 2.34 times longer. The mean has a larger standard error resulting in a longer CI.

**39**.

```
> m=c(56,106,174,207,219,237,313,365,458,497,515,529,557,615,625,645,973,1065,3215)
> t.test(m)
```

```
        One Sample t-test

data:  m
t = 3.7891, df = 18, p-value = 0.001344
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 266.6441 930.3033
sample estimates:
mean of x
 598.4737
```

```
> library(WRS)
> trimci(m)
```

```
[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
$ci
[1] 293.5976 595.9409

$estimate
[1] 444.7692

$test.stat
[1] 6.410388

$se
[1] 69.38258

$p.value
[1] 3.35059e-05

$n
[1] 19
```

```
> out(m)
```

```
$n
[1] 19

$n.out
[1] 1

$out.val
[1] 3215
```

```
$out.id
[1] 19

$keep
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18

$dis
 [1] 1.61657840 1.43329286 1.18402454 1.06305609 1.01906756 0.95308477
 [7] 0.67449076 0.48387381 0.14296272 0.00000000 0.06598279 0.11730274
[13] 0.21994264 0.43255386 0.46921096 0.54252518 1.74487827 2.08212365
[19] 9.96340154

$crit
[1] 2.241403
```

So the confidence interval based on the mean is (266.6441, 930.3033) For a 20% trimmed mean it is (293.5976, 595.9409). Using the MAD-median rule, the value 3215 is flagged as an outlier. The CI based on a 20% trimmed mean is far shorter than the CI for the mean because the outlier (3213) inflates the SE. In the case of the trimmed mean, the outlier is trimmed.

**40**. Under normality, the sample mean has the smallest standard error. So it is the only candidate for being ideal. But as we have seen, other estimators have a smaller standard error than the mean in other situations, so an optimal estimator does not exist.

**41**. No, because what often appears to be normal is not normal. In addition, there are robust estimators that compare relatively well (although not as well) to the mean under normality but perform far better in situations under small departures from normality.

**42**.

```
> c=c(250,220,281,247,230,209,240,160,370,274,210,204,243,251,190,200,130,150,177,475,
> t.test(c)

        One Sample t-test

data:  c
t = 16.514, df = 28, p-value = 5.759e-16
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 200.7457 257.5991
sample estimates:
mean of x
 229.1724

> library(WRS)
> trimci(c)
```

```
[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
$ci
[1] 196.6734 244.9056

$estimate
[1] 220.7895

$test.stat
[1] 19.23453

$se
[1] 11.47881

$p.value
[1] 1.887379e-13

$n
[1] 29
```

**43**.

```
> c=c(250,220,281,247,230,209,240,160,370,274,
+ 210,204,243,251,190,200,130,150,177,475,221,350,224,
+ 163,272,236,200,171,98)
> library(WRS)
> out(c)

$n
[1] 29

$n.out
[1] 4

$out.val
[1] 370 475 350  98

$out.id
[1]  9 20 22 29

$keep
 [1]  1  2  3  4  5  6  7  8 10 11 12 13 14 15 16 17 18 19 21 23 24 25 26 27 28

$dis
 [1] 0.65200773 0.02248303 1.34898152 0.58455866 0.20234723 0.26979630
```

```
 [7] 0.42717748 1.37146454 3.34997077 1.19160034 0.24731328 0.38221143
[13] 0.49462656 0.67449076 0.69697378 0.47214353 2.04595530 1.59629480
[19] 0.98925311 5.71068843 0.00000000 2.90031027 0.06744908 1.30401547
[25] 1.14663429 0.33724538 0.47214353 1.12415127 2.76541211

$crit
[1] 2.241403
```

Four values are flagged as outliers, so it is not surprising that the length of the confidence interval based on a 20% trimmed mean is shorter.

**44**. Even if the two measures are identical, outliers can largely inflate the CI based on mean, rendering the outcome less informative.

**45**. In this case we have 16 successes in 16 trials.

```
> binomci(16,16,alpha=.01)

$phat
[1] 1

$ci
[1] 0.7498942 1.0000000

$n
[1] 16
```

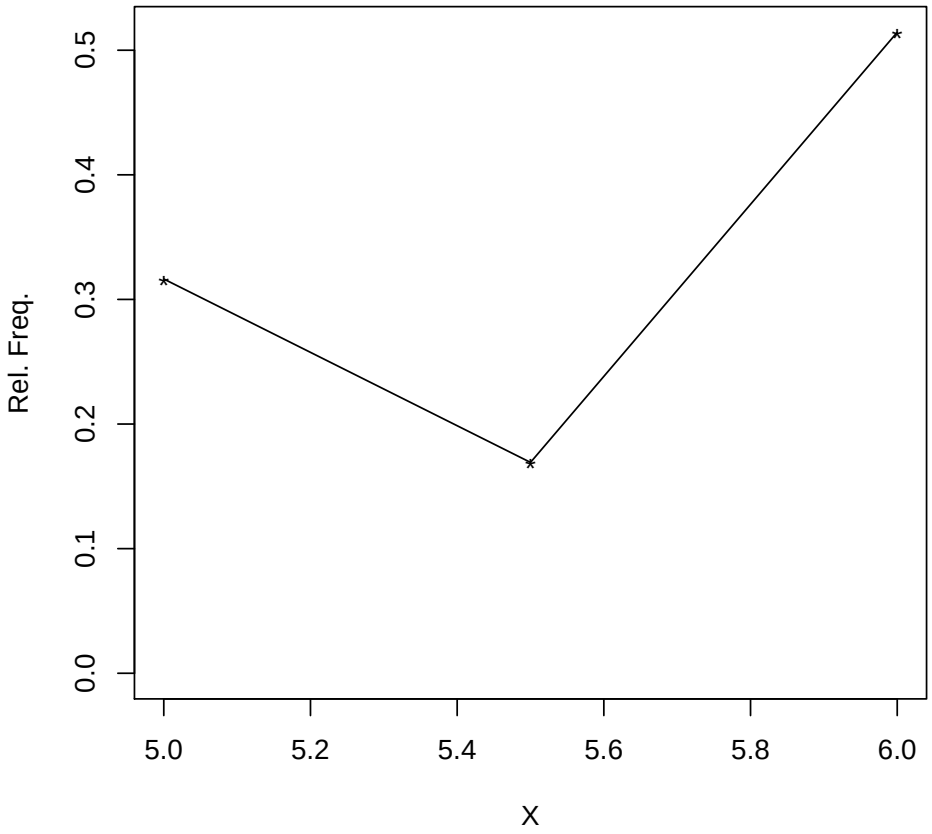
**46**. In this case we have 0 successes in 200000 trials. Using R:

```
> binomci(0,200000)

$phat
[1] 0

$ci
[1] 0.000000e+00 1.497855e-05

$n
[1] 2e+05
```

**47**.

```
> val=NULL # initialize an R variable where results will be stored
> set.seed(2) #Set the seed of the random number generator so that the
> #         reader can duplicate the results.
> for(i in 1:5000){
+ x=rbinom(20,6,0.9)
+ val[i]=median(x)
+ }
> splot(val)
```

```
$n
[1] 5000

$frequencies
[1] 1582  846 2572

> # could have also used the command plot(table(val))
```




CHAPTER 5

**1**. Because $\sigma$ is given, compute $Z$ and use Table 1.
For $H_0$: $\mu > 80$, the critical value is in the left tail of a standard normal distribution.

```
> Z=sqrt(10)*(78-80)/5
> Z

[1] -1.264911
```

With $\alpha = .05$, the critical value is $c = -1.645$. So fail to reject because $Z > -1.645$.

**2**. Because $\sigma$ is given, compute $Z$ and use Table 1 or could determine the critical value with the R function qnorm. Testing for equality, so there are two critical values.

```
> Z=sqrt(10)*(78-80)/5
> Z

[1] -1.264911

> qnorm(.025) #lower critical value

[1] -1.959964

> qnorm(.975) # Upper critical value.

[1] 1.959964
```

Fail to reject because $Z$ has a value between the two critical values. Or in terms of Tukey's three decision rule, make no decision about whether the population mean is greater than or less than 80.

**3**.

```
> 78-1.96*5/sqrt(10)

[1] 74.90097

> 78+1.96*5/sqrt(10)

[1] 81.09903
```

This interval contains the hypothesized value, so fail to reject.

**4**. The value of the test statistic is $Z = -1.265$. With $H_0$: $\mu > 80$, reject if $Z$ is sufficiently small. Using $-1.265$ as the critical value, the p-value is $P(Z \leq -1.265)$. Computing this probability (the p-value) using R:

```
> pnorm(-1.265)

[1] 0.1029357
```

**5**. This is a two tailed test, so compute a p-value using Equation (5.4) on p. 185. $|Z| = 1.265$, so the p-value is

```
> 2*(1-pnorm(1.265))

[1] 0.2058713
```

**6**.

```
> Z=sqrt(49)*(120-130)/5
> Z

[1] -14
```

The critical value is $-1.645$, reject.

**7**. Now the critical values are $-1.96$ and 1.96. $Z$ is smaller than the lower critical value, so reject.

**8**.

```
> 120-1.96*5/sqrt(49)

[1] 118.6

> 120+1.96*5/sqrt(49)

[1] 121.4
```

The confidence interval does not contain the hypothesized value, so reject.

**9**. Yes, because the sample mean is consistent with the null hypothesis so fail to reject.

**10**. $H_0$: $\mu < 232$.

```
> Z=sqrt(25)*(240-232)/4
> Z

[1] 10

> qnorm(.95) #critical value

[1] 1.644854
```

Reject

**11**. Testing for exact equality.

```
> Z=sqrt(20)*(565-546)/40
> Z

[1] 2.124265
```

The critical values are $-1.964$ and 1.96, reject.

**12**. Confidence intervals indicate whether a hypothesis should be rejected and it simultaneously indicates a range of values that is likely to contain the population mean. Hypothesis testing, from the point of view of Tukey's three decision rule, reflects the strength of the empirical evidence that a decision can be made about whether the population mean is less than or greater than the null value. But note that when working with means, both methods depend crucially on the normality assumption.

**13**. This is case 2 on p. 188.
$1 - \beta = P(Z \leq -2.33 - \sqrt{25}(56 - 60)/5) = P(Z \leq 1.67) = .9525$

Note, if $\sigma$ is not known and is estimated with $s$, power can be determined using R:

```
> power.t.test(25,4/5,type="one.sample",alternative="one.sided",sig.level=0.01)
```

```
     One-sample t test power calculation

              n = 25
          delta = 0.8
             sd = 1
      sig.level = 0.01
          power = 0.9254881
    alternative = one.sided
```

**14**. This is case 1, p. 188.
$1-\beta = P(Z \geq 1.96\sqrt{36}(103-100)/8) = P(Z \leq 1.67) = .614$
Using R:

```
> 1-pnorm(1.96-6*(103-100)/8)

[1] 0.6140919
```

**15**. This is case 3, p. 188.
$1-\beta = P(Z \leq -4.06) + P(Z \geq -.14) = .556$.

**16**. The sample size is too small, so failure to reject can be the result of insufficient power.

**17**. The critical value is $-1.645$. $1-\beta = P(Z \leq -1.645 - \sqrt{10}(46-48)/5) = P(Z \leq -0.38) = .35$

**18**. $1-\beta = P(Z \leq -1.645 - \sqrt{20}(46-48)/5) = P(Z \leq 0.143) = .556$
$1-\beta = P(Z \leq -1.645 - \sqrt{30}(46-48)/5) = P(Z \leq 0.54) = .705$
$1-\beta = P(Z \leq -1.645 - \sqrt{40}(46-48)/5) = P(Z \leq 0.884) = .81$

**19**. Increase $\alpha$, but this will also increases Type I error probability.

**20**.Given $s$, not $\sigma$, so use $T$, not $Z$.

```
> T=sqrt(25)*(44-42)/10  # (a) test statistic, fail to reject
> T

[1] 1

> qt(.975,25) # critical value, fail to reject

[1] 2.059539

> T=sqrt(25)*(43-42)/10  # (b) test statistic
> T

[1] 0.5

> qt(.975,24) # critical value

[1] 2.063899
```

```
> T=sqrt(25)*(43-42)/2  # (c) test statistic
> T

[1] 2.5

> qt(.975,25) # critical value,  reject

[1] 2.059539
```

**21**. Power depends on the sample variance; larger variance lower power.

**22**.

```
> T=sqrt(16)*(44-42)/10  # (a) test statistic, fail to reject
> T

[1] 0.8

> qt(.95,15) # critical value, fail to reject

[1] 1.75305

> T=sqrt(16)*(43-42)/10  # (b) test statistic
> T

[1] 0.4

> qt(.95,15) # critical value

[1] 1.75305

> T=sqrt(16)*(43-42)/2  # (c) test statistic
> T

[1] 2

> qt(.95,15) # critical value,  reject

[1] 1.75305
```

**23**. The sample means are consistent with the null hypothesis, so we can fail to reject without performing any calculations.

**24**.

```
> T=sqrt(10)*(46.4-45)/11.27
> T

[1] 0.3928295

> qt(.975,9)
```

```
[1] 2.262157

> # OR
> x=c(38, 44, 62, 72, 43, 40,  43, 42, 39, 41)
> t.test(x,mu=45)

        One Sample t-test

data:  x
t = 0.39295, df = 9, p-value = 0.7035
alternative hypothesis: true mean is not equal to 45
95 percent confidence interval:
 38.34045 54.45955
sample estimates:
mean of x
     46.4
```

**25**.

```
> T=sqrt(100)*(9.79-10.5)/2.72
> T

[1] -2.610294

> qt(.025,99)

[1] -1.984217
```

**26**.

(a)
$T_t = \frac{.6\sqrt{16}(44-42)}{9} = 0.53$
DF=16-6-1=9. Critical value is

```
> qt(.975,9)

[1] 2.262157
```

$|T_t| < 2.262$, fail to reject.
(b)
$T_t = \frac{.6\sqrt{16}(43-42)}{9} = 0.26$
DF=16-6-1=9. Critical value is

```
> qt(.975,11)

[1] 2.200985
```

$|T_t| < 2.262$, fail to reject.
(c)
$T_t = \frac{.6\sqrt{16}(43-42)}{3} = 0.8$
DF=16-6-1=9. Critical value is

```
> qt(.975,9)

[1] 2.262157
```

$|T_t| < 2.262$, fail to reject.

**27**.
(a)
$T_t = \frac{.6\sqrt{16}(44-42)}{9} = 0.53$
DF=16-6-1=9. Critical value is

```
> qt(.95,9)

[1] 1.833113
```

$|T_t| < 1.83$, fail to reject.
(b)
$T_t = \frac{.6\sqrt{16}(43-42)}{9} = 0.26$
DF=16-6-1=9. Critical value is

```
> qt(.95,9)

[1] 1.833113
```

$|T_t| < 1.83$, fail to reject.
(c)
$T_t = \frac{.6\sqrt{16}(43-42)}{3} = 0.8$
DF=16-6-1=9. Critical value is

```
> qt(.975,9)

[1] 2.262157
```

$|T_t| < 1.83$, fail to reject.

**28**. $T_t = \frac{.6\sqrt{10}(42.17-45)}{1.73} = -3.1$
DF=10-4-1=5. Critical value is

```
> qt(.975,5)

[1] 2.570582
```

$|T_t| > 2.57$, reject.

**29**
$T_t = \frac{.6\sqrt{25}(5.1-4.8)}{7} = 0.129$
DF=25-10-1=14. Critical value is

```
> qt(.995,14)

[1] 2.976843
```

$|T_t| < 2.97$, fail to reject.

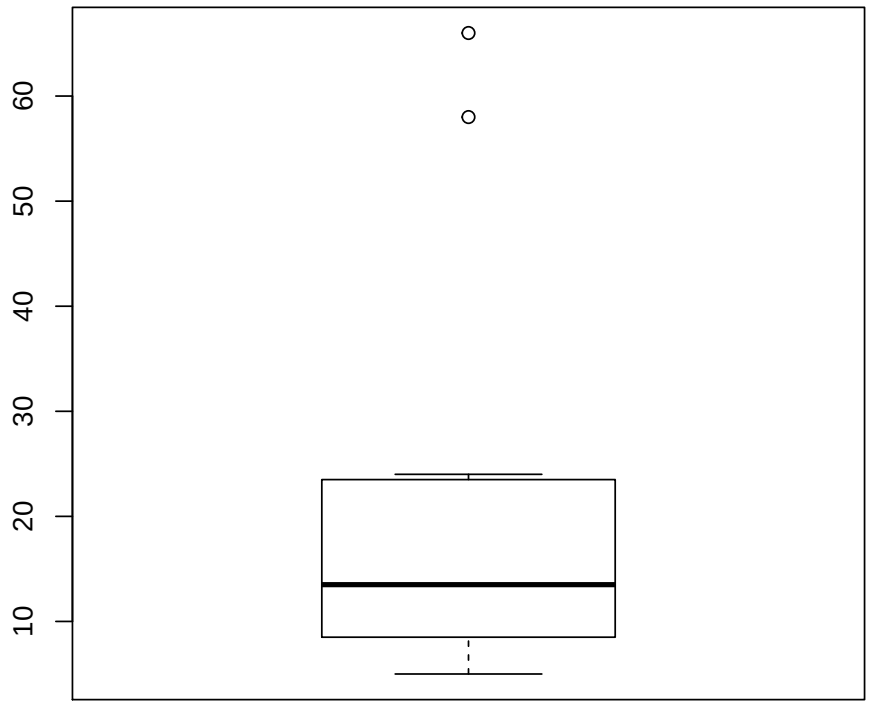
**30**. Using R:

```
> x=c(7.6,8.1,9.6,10.2,10.7,12.3,13.4,13.9,14.6,15.2) # store
> #      the bootstrap sample means in x
> pstar=mean(x<8.5) # proportion less than the hypothesized value
> pstar  # 20% are less than the hypothesized value.

[1] 0.2

> p.value=2*min(c(pstar,1-pstar))
> p.value

[1] 0.4

> xsort=sort(x)  # sort the values. A 0.8 confidence interval is based
>         # on the middle 80% of the sorted values. With 10
>         # bootstrap values, the confidence interval is
> c(xsort[2],xsort[9])

[1]  8.1 14.6
```

**31**.

```
> x=c(5,12,23,24,6,58,9,18,11,66,15,8)
> trimci(x)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
$ci
[1]  7.157829 22.842171
```

```
$estimate
[1] 15

$test.stat
[1] 4.522901

$se
[1] 3.316456

$p.value
[1] 0.002722697

$n
[1] 12

> trimci(x,tr=0)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
$ci
[1]  8.504757 33.995243

$estimate
[1] 21.25

$test.stat
[1] 3.669678

$se
[1] 5.790699

$p.value
[1] 0.003691702

$n
[1] 12

> boxplot(x)
```




The boxplot indicates outliers, suggesting that the standard error of the mean will be larger than the standard error of a 20% trimmed mean.

**32**.

```
> x=c(5,12,23,24,6,58,9,18,11,66,15,8)
> trimcibt(x,side=FALSE,tr=0)

[1] "NOTE: p.value is computed only when side=T"
$estimate
[1] 21.25

$ci
[1] 12.24982 52.55845

$test.stat
[1] 3.669678

$p.value
[1] NA
```

```
$n
[1] 12
```

Using side=TRUE gives a different result.

**33**. The 20% trimmed mean because skewness and outliers impact the T distribution in a manner that changes its cumulative probabilities (and quantiles). Trimmed means, based in bootstrap methods (trimcibt), perform much better when dealing skewed distributions, and they deal with outliers. The bootstrap method estimates well the sampling distribution of T. However, if there is some substantive reason for wanting to make inferences about the population mean, for skewed distributions methods based on a trimmed mean are unsatisfactory.

**34**.

```
> y=c(7.6,8.1,9.6,10.2,10.7,12.3,13.4,13.9,14.6,15.2)
> trimcibt(y, side=FALSE)

[1] "NOTE: p.value is computed only when side=T"
$estimate
[1] 11.68333

$ci
[1]  7.834137 15.807540

$test.stat
[1] 11.36358

$p.value
[1] NA

$n
[1] 10
```

**35**.

```
> x=c(5,12,23,24,6,58,9,18,11,66,15,8)
> trimpb(x)

[1] "The p-value returned by this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
$ci
[1]  9.75 31.50

$p.value
[1] 0
```

**36**. This is a difficult question. With sufficiently heavy-tailed distributions, the percentile bootstrap is preferable. With a light-tailed distribution, the symmetric bootstrap t is better. A boxplot provides some indication of which situation we have at hand but it is not definitive. Because of the outliers, it seems that a percentile bootstrap method is best. Generally, to avoid having the actual Type I error substantially smaller than the nominal level, a percentile bootstrap method is preferable with the understanding that the actual level might be a bit higher than intended. See Table 7.2, p. 281.

**37**.

```
> x=c(2,4,6,7,8,9,7,10,12,15,8,9,13,19,5,2,100,200,300,400)
> onesampb(x)

$ci
[1]  7.425758 19.806385

$n
[1] 20

$estimate
[1] 10.3303

$p.value
[1] 0
```

**38**.

```
> x=c(2,4,6,7,8,9,7,10,12,15,8,9,13,19,5,2,100,200,300,400)
> trimpb(x)

[1] "The p-value returned by this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
$ci
[1]  7.25000 63.33333

$p.value
[1] 0
```

The sample size is $n = 20$, there are four large outliers, so in some instances the percentage of outliers contained in bootstrap sample will exceed 20% resulting in a relatively high estimate. In contrast, an M-estimator can handle more outliers that occur in a bootstrap sample.

**39**.

```
> x=c(2,4,6,7,8,9,7,10,12,15,8,9,13,19,5,2,100,200,300,400)
> trimpb(x,tr=0.3)
```

```
[1] "The p-value returned by this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
$ci
[1]  7.125 35.500

$p.value
[1] 0
```

A larger amount of trimming can handle more outliers that might occur in a bootstrap sample.

**40**.

```
> x=c(2,4,6,7,8,9,7,10,12,15,8,9,13,19,5,2,100,200,300,400)
> trimpb(x,tr=0.4)

[1] "The p-value returned by this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
$ci
[1]  7.0 14.5

$p.value
[1] 0
```

A larger amount of trimming can handle more outliers that might occur in a bootstrap sample.

**41**. There are tied (duplicated) values, so the estimate of the standard error might be highly inaccurate.

**42**.

```
> x=c(2,4,6,7,8,9,7,10,12,15,8,9,13,19,5,2,100,200,300,400)
> sint(x)

[1]  7.00000 14.52953
```

In some situations the confidence for the median can be shorter.

CHAPTER 6

**1**.

```
> x=c(5,8,9,7,14)
> y=c(3,1,6,7,19)
> lsfit(x,y)$coef

Intercept         X
-8.477876  1.823009
```

**2**.

```
> x=c(5,8,9,7,14)
> y=c(3,1,6,7,19)
> yhat=1.8*x-8.5
> res=y-yhat
> sum(res^2)

[1] 46.85
```

**3**.

```
> x=c(5,8,9,7,14)
> y=c(3,1,6,7,19)
> yhat=2*x-9
> res=y-yhat
> sum(res^2)

[1] 53
```

The sum of the squared residuals should be larger than 47 because least squares regression is designed so as to minimize the sum of the squared residuals.

**4**.

$b_1 = 0.6\sqrt{\frac{25}{12}} = 0.866$.

**5**.

```
> a=c(3,104,50,9,68,29,74,11,18,39,0,56,54,77,14,32,34,13,96,84,5,4,18,76,34,14,9,28,7
> b=c(0,5,0,0,0,6,0,1,1,2,17,0,3,6,4,2,4,2,0,0,13,9,1,4,2,0,4,0,4,6,4,4,1,6,6,13,3,1,0
> lsfit(a,b)$coef

  Intercept           X
 4.58061839 -0.04051423
```

**6**. The data are stored in the file cancer_rate_dat.txt on the author's web page.

```
> cancer=read.table('cancer_rate_dat.txt',sep='&',header=TRUE)
> lsfit(cancer[,3],cancer[,2])$coef

  Intercept           X
39.99094634 -0.03565283
```

The slope is negative suggesting that as daily calories increases, average breast cancer rates decline.

**7**.

```
> x=c(500,530,590,660,610,700,570,640)
> y=c(2.3,3.1,2.6,3.0,2.4,3.3,2.6,3.5)
> lsfit(x,y)$coef
```

```
  Intercept           X
0.484615385 0.003942308
```

**8**.

```
> x=c(500,530,590,660,610,700,570,640)
> y=c(2.3,3.1,2.6,3.0,2.4,3.3,2.6,3.5)
> cor(x,y)^2

[1] 0.3618685
```

This means that SAT accounts for about 36% of the variance in GPA based on least squares regression, assuming the regression line is straight. This gives an indication of the strength of the association

**9**.

```
> x=c(40,41,42,43,44,45,46)
> y=c(1.62,1.63,1.90,2.64,2.05,2.13,1.94)
> lsfit(x,y)$coef

  Intercept           X
-1.25321429  0.07535714
```

**10**.

```
> cancer=read.table('cancer_rate_dat.txt',sep='&',header=TRUE)
> fit=lsfit(cancer[,3],cancer[,2])$coef
> 600*fit[2]+fit[1]

       X
18.59925
```

A concern is extrapolation. The value 600 is greater than any of the daily calorie measures used in the study.

**11**.

```
> mou=c(63.3,60.1,53.6,58.8,67.5,62.5)
> time=c(241.5,249.8,246.1,232.4,237.2,238.4)
> library(WRS)
> wincor(mou,time,tr=0) #The current WRS version  of this function

$cor
[1] -0.3662634

$cov
[1] -10.842

$p.value
```

```
[1] 0.4751718

$n
[1] 6

>                    #reports the p-value as significance level.
```

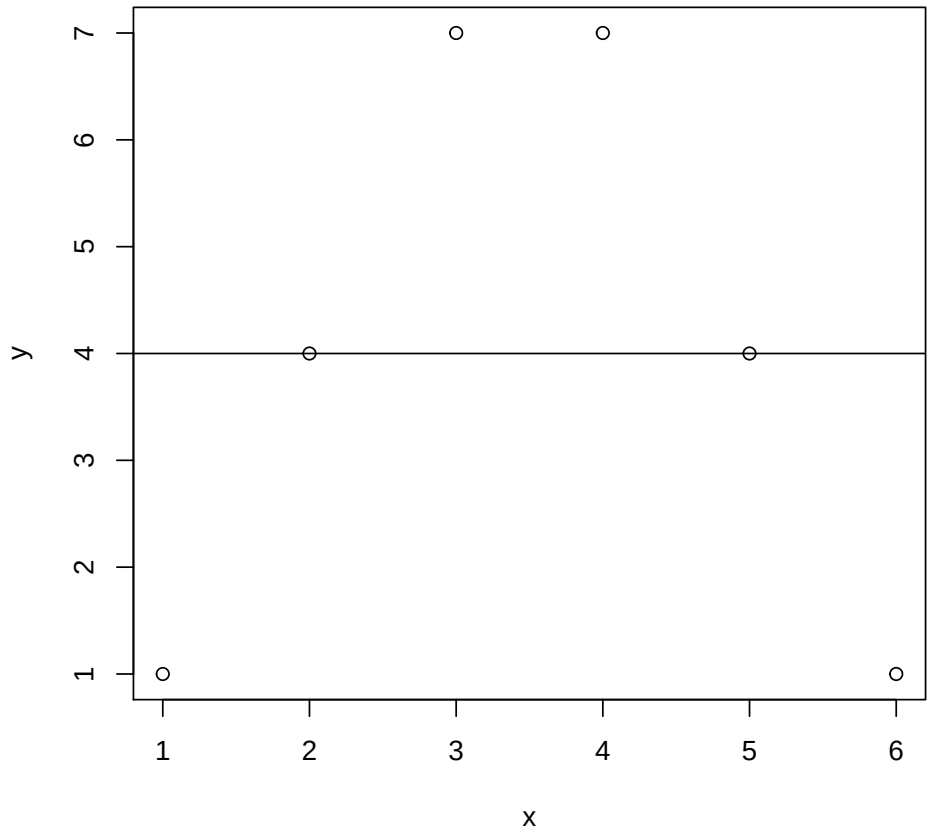
Alternatively,
$T = -0.366\sqrt{\frac{6-2}{1-(-0.366)^2}} = 0.787$. DF=4 and the critical value is

```
> qt(.975,4)

[1] 2.776445
```

Because $|T|$ is less than the critical value, fail to reject.

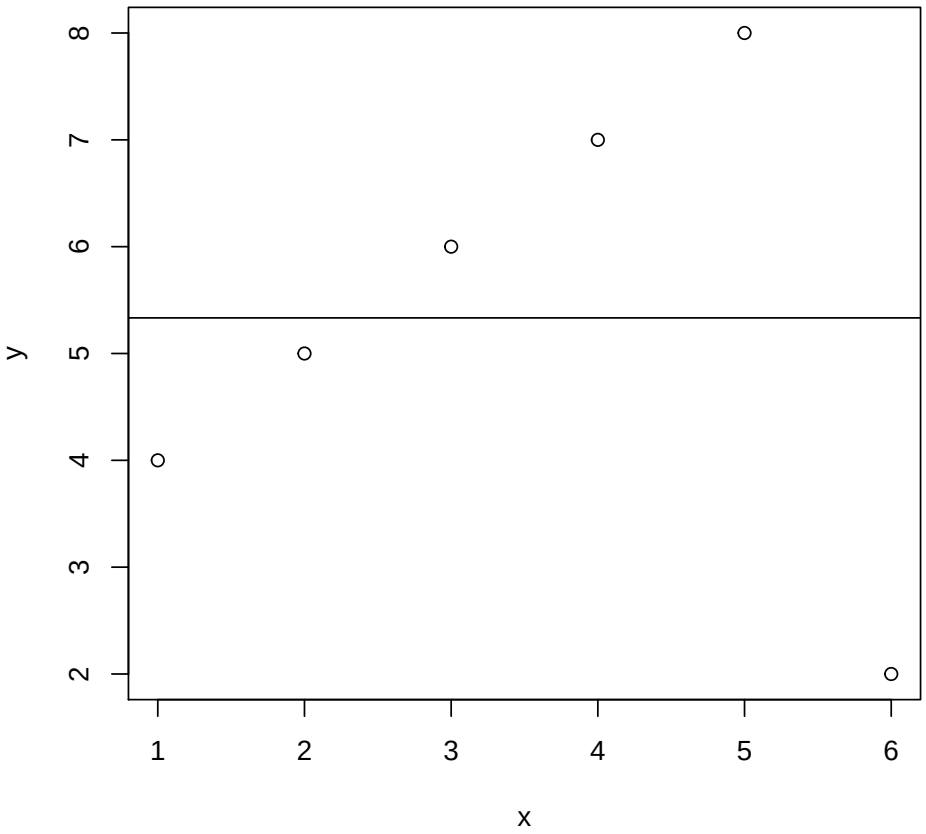
**12**.

```
> x=c(1,2,3,4,5,6)
> y=c(1,4,7,7,4,1)
> coef=lsfit(x,y)$coef
> coef

    Intercept             X
 4.000000e+00 -5.838669e-16

> plot(x,y)
> abline(coef[1],coef[2])
```




**13**.

```
> x=c(1:6)
> y=c(4:8,2)
> coef=lsfit(x,y)$coef
> coef

    Intercept             X
 5.333333e+00 -6.369458e-16

> plot(x,y)
> abline(coef[1],coef[2])
```




**14**. The nature of the relationship between two variables can vary with the predictor value. In other words, the association between Y and X can change as a function of X values. Extrapolating beyond the data range, therefore, can be problematic.

**15**.

```
> diab=read.table('diabetes_sockett_dat.txt',header=TRUE)
> cor(diab$age,diab$pep)

[1] 0.388361

> library(WRS)
> hc4test(diab$age,diab$pep)

$n
[1] 43

$n.keep
[1] 43

$test
```

```
[1] 4.51634

$p.value
[1] 0.03357257

$coef
 Intercept          X
4.15498509 0.06721893
```

**16**.

```
> diab=read.table('diabetes_sockett_dat.txt',header=TRUE)
> cor(diab$age,diab$pep)

[1] 0.388361

> flag=diab$age<7
> lsfit(diab$age[flag],diab$pep[flag])$coef

Intercept         X
3.5148814 0.2474008

> lsfit(diab$age[!flag],diab$pep[!flag])$coef

  Intercept           X
4.797163370 0.009422997
```

The results suggest that there is a positive association up to about the age of 7, but for older children the strength of the association is much weaker.

**17**.

```
> home=read.table('home_sales_dat.txt',header=TRUE)
> lsfit(home$size,home$price)$coef

 Intercept          X
38.1921217  0.2153008
```

The intercept is 38,192, the estimated cost of a house on an empty lot, which is much less than the cost of any realistic cost. Extrapolation should be avoided.

**18**. Using R:

```
> lot=c(18200,12900,10060,14500,76670,22800,10880,10880,23090,10875,3498,42689,17790,3
> price=c(510,690,365,592,1125,850,363,559,860,695,182,860,1050,675,859,435,555,525,80
> ols(lot/1000,price)$coef

(Intercept)           x
  436.83368    11.04288
```

**19**. Yes, because the magnitude of Pearson's correlation depends on the magnitude of the residuals.

**20**.

```
> x=c(18,20,35,16,12)
> y=c(36,29,48,64,18)
> ols(x,y)

$n
[1] 5

$n.keep
[1] 5

$summary
              Estimate Std. Error   t value  Pr(>|t|)
(Intercept) 25.3283679  23.774217 1.0653713 0.3648449
x            0.6768135   1.096856 0.6170485 0.5808715

$coef
(Intercept)           x
 25.3283679   0.6768135

$F.test
    value
0.3807488

$Ftest.p.value
    value
0.5808715

$F.test.degrees.of.freedom
numdf dendf
    1     3

$R.squared
[1] 0.1126226

$residuals
NULL

> wincor(x,y,tr=0)

$cor
[1] 0.3355929
```

```
$cov
[1] 52.25

$p.value
[1] 0.5808715

$n
[1] 5
```

Both analyses agree, both fail to reject, but there might an association that is not readily detected using least squares regression. And even when the assumptions are true, there are power considerations with a small sample size.

**21**.

```
> x=c(12.2,41,5.4,13,22.6,35.9,7.2,5.2,55,2.4,6.8,29.6,58.7)
> y=c(1.8,7.8,0.9,2.6,4.1,6.4,1.3,0.9,9.1,0.7,1.5,4.7,8.2)
> temp=ols(x,y)
> df=temp$n-2  #degrees of freedom
> df

[1] 11

> temp$summary

             Estimate  Std. Error   t value     Pr(>|t|)
(Intercept) 0.3269323 0.248122843  1.317623 2.144131e-01
x           0.1550843 0.008413901 18.431919 1.280856e-09

> temp$summar[2,1]-qt(.975,df)*temp$summary[2,2]

[1] 0.1365655

> temp$summar[2,1]+qt(.975,df)*temp$summary[2,2]

[1] 0.1736032
```

So the 0.95 confidence interval is (0.1366, 0.1736). The R command
olshomci(x,y)
is an easier way of computing a confidence interval for the slope.

**22**.

```
> x=scan(file='read_table6_7_x_dat.txt',sep=',')
> y=scan(file='read_table6_7_y_dat.txt',sep=',')
> # These files are stored on the author's web page.
> temp=ols(x,y)
> df=temp$n-2  #degrees of freedom
> df
```

```
[1] 75

> temp$summary

               Estimate Std. Error    t value     Pr(>|t|)
(Intercept) 97.95728197 4.73432147 20.6908809 9.985891e-33
x           -0.02136595 0.07096758 -0.3010664 7.641969e-01

> temp$summar[2,1]-qt(.975,df)*temp$summary[2,2]

[1] -0.1627406

> temp$summar[2,1]+qt(.975,df)*temp$summary[2,2]

[1] 0.1200087
```

The R command
olshomci(x,y)
is an easier way of computing a confidence interval for the slope.

**23**.

```
> x=scan(file='read_table6_7_x_dat.txt',sep=',')
> y=scan(file='read_table6_7_y_dat.txt',sep=',')
> # These files are stored on the author's web page.
> khomreg(x,y)

$test
          [,1]
[1,] 0.3083691

$p.value
          [,1]
[1,] 0.5786827
```

But it is unclear whether power is sufficiently high to detect heteroscedasticity that might be a practical concern.

**24**.

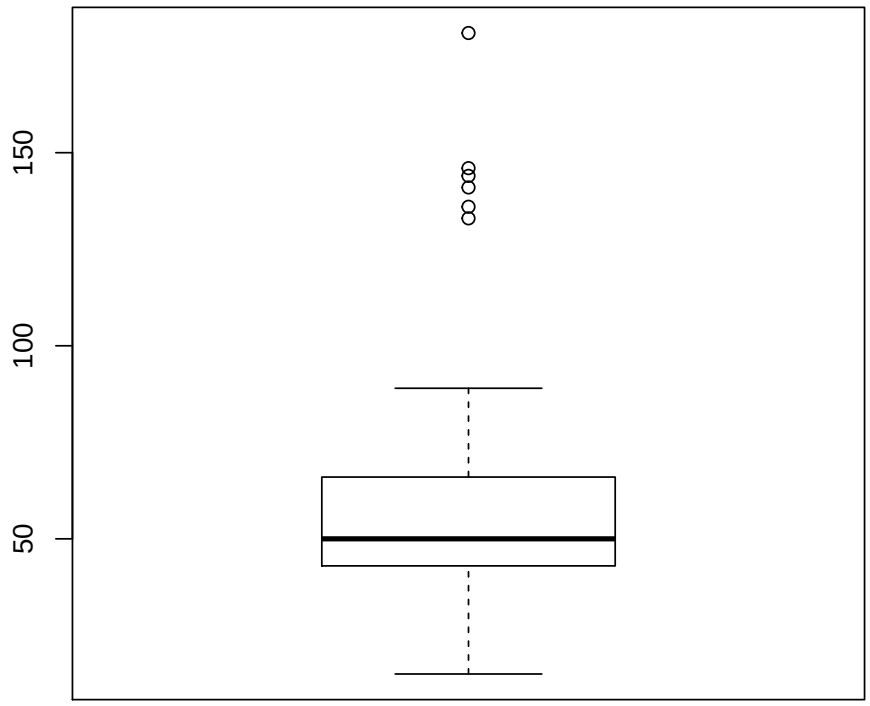
```
> x=scan(file='read_table6_7_x_dat.txt',sep=',')
> y=scan(file='read_table6_7_y_dat.txt',sep=',')
> # These files are stored on the author's web page.
> ols(y,x)

$n
[1] 77

$n.keep
```

```
[1] 77

$summary
               Estimate Std. Error    t value    Pr(>|t|)
(Intercept) 65.46175413 18.4508380  3.5479014 0.000673844
x           -0.05649584  0.1876524 -0.3010664 0.764196940

$coef
(Intercept)           x
65.46175413 -0.05649584

$F.test
     value
0.09064099

$Ftest.p.value
    value
0.7641969

$F.test.degrees.of.freedom
numdf dendf
    1    75

$R.squared
[1] 0.001207088

$residuals
NULL
```

*> rqfit(y,x) #assumes package quantreg has been installed.*

```
$coef
(Intercept)           x
 95.2000000  -0.4333333

$ci
               lower bd    upper bd
(Intercept) 64.4610733 105.9727355
x           -0.5505706  -0.1450298

$residuals
[1] NA
```

*> boxplot(x)*




There are outliers among the dependent variable x, which can destroy power when using least squares and they can have a substantial impact on the estimate of the slope. Note that the OLS estimate of the slope is $-0.056$ while rqfit estimates the slope to be $-0.551$, which is about tens times larger. The outliers have less of an impact when using rqfit.

**25**.

```
> library(MASS)
> ols(leuk[,1],leuk[,3])

$n
[1] 33

$n.keep
[1] 33

$summary
                 Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) 53.8899623928 1.027986e+01  5.242286 1.072131e-05
x           -0.0004461206 2.296306e-04 -1.942775 6.117379e-02
```

```
$coef
  (Intercept)             x
53.8899623928 -0.0004461206

$F.test
   value
3.774373

$Ftest.p.value
     value
0.06117379

$F.test.degrees.of.freedom
numdf dendf
    1    31

$R.squared
[1] 0.1085389

$residuals
NULL

> olshc4(leuk[,1],leuk[,3])

$n
[1] 33

$n.keep
[1] 33

$ci
            Coef.     Estimates      ci.lower      ci.upper      p-value
(Intercept)     0 53.8899623928 30.5619402421  7.721798e+01 4.902827e-05
Slope           1 -0.0004461206 -0.0008776261 -1.461508e-05 4.315956e-02
               Std.Error
(Intercept) 1.143803e+01
Slope       2.115728e-04

$cov
               [,1]          [,2]
[1,] 130.828599434 -1.844167e-03
[2,]  -0.001844167  4.476304e-08
```

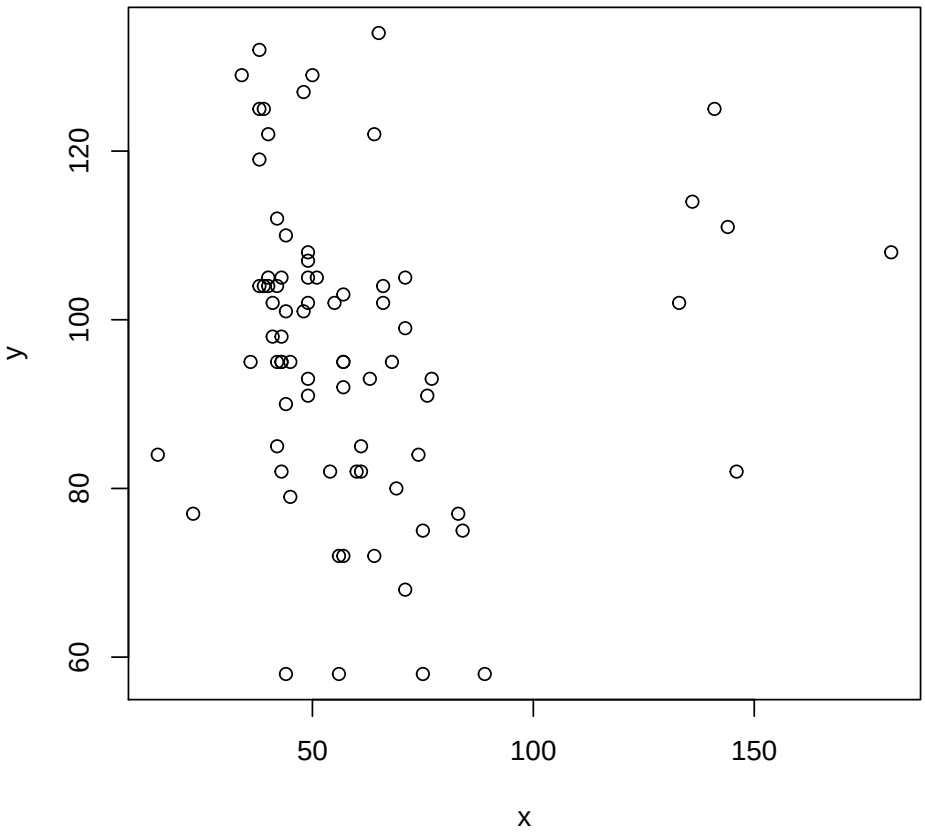
olshc4 rejects at the.05 level, ols does not. Note that the estimated standard error is

smaller using olshc4.

**26**. The methods might be relatively insensitive (have low power) based on the nature of the association. Outliers can mask a true association.

**27**.

```
> x=scan(file='read_table6_7_x_dat.txt',sep=',')
> y=scan(file='read_table6_7_y_dat.txt',sep=',')
> plot(x,y)
> flag=x<125
> lsfitci(x[flag],y[flag])

[1] "Taking bootstrap samples; please wait"
$intercept.ci
[1] 104.3009 137.8797

$slope.ci
           [,1]       [,2]
[1,] -0.7979083 -0.1467203

$crit.level
[1] NA

$p.values
[1] NA
```




Outliers among the independent variable can mask as association among the bulk the data.

**28**.

```
> x=scan(file='read_table6_7_x_dat.txt',sep=',')
> y=scan(file='read_table6_7_y_dat.txt',sep=',')
> flag=x<125
> pcorb(x[flag],y[flag])

$r
[1] -0.3868361

$ci
[1] -0.6403710 -0.1176394
```

**29**. The data are stored in the file cancer_rate_dat.txt on the author's web page.

```
> cancer=read.table('cancer_rate_dat.txt',sep='&',header=TRUE)
> lsfitci(cancer[,3],cancer[,2])
```

```
[1] "Taking bootstrap samples; please wait"
$intercept.ci
[1] 35.78134 44.68923

$slope.ci
            [,1]        [,2]
[1,] -0.04809108 -0.02457128

$crit.level
[1] NA

$p.values
[1] NA
```

In some situations, classic methods give very similar results to those obtained with more modern techniques.

**30**.

```
> x=scan('hubble_distance_dat.txt')
> y=scan('hubble_recession_dat.txt')
> lsfitci(x,y)

[1] "Taking bootstrap samples; please wait"
$intercept.ci
[1] -191.9351  108.0508

$slope.ci
         [,1]    [,2]
[1,] 305.8974 642.884

$crit.level
[1] NA

$p.values
[1] NA
```

CHAPTER 7

**1**.

$T = \frac{15-12}{\sqrt{13.14\left(\frac{1}{20}+\frac{1}{10}\right)}} = 2.14$

$\nu = 10 + 20 - 2 = 28$

Using R:

```
> T=(15-12)/sqrt(13.14*(1/20+1/10))
> T
```

```
[1] 2.136869

> qt(.975,28) # The critical value

[1] 2.048407
```

$|T| > c$, the critical value, so reject.

**2**.
$s_p^2 = \frac{(20-1)4+(30-1)16}{20+30-2} = 11.25$
Using R:

```
> top=(20-1)*4+(30-1)*16
> bot=20+30-2
> top/bot

[1] 11.25
```

**3**. From Exercise 2, $s_p^2 = 11.25$, so
$T = \frac{45-36}{\sqrt{11.25(1/20+1/30)}} = 9.295$
$\nu = 20 + 30 - 2 = 48$
Using R:

```
> T=(45-36)/sqrt(11.25*(1/20+1/30))
> T

[1] 9.29516

> qt(.975,38) #Critical value

[1] 2.024394
```

$|T| > c$, the critical value, reject.

**4**.

```
> W=(45-35)/sqrt(4/20+16/30)
> W

[1] 11.67748

> q1=4/20
> q2=16/30
> df=(q1+q2)^2/(q1^2/15+q2^2/29)
> df

[1] 43.10811

> qt(.975,df)
```

```
[1] 2.016546
```

$|W| > c$, the critical value, reject.

**5**. Welch appears to have more power in this case simply because the statistic $W$ is substantially larger than $T$. (It can be shown that Welch has a smaller p-value.)

**6**.

```
> s.sq.p=((20-1)*25+(20-1)*25)/(20+20-2)
> T=(86-80)/sqrt(s.sq.p*(1/20+1/20))
> T

[1] 3.794733

> DF=20+20-2
> qt(.975,DF)

[1] 2.024394
```

Reject

**7**.

```
> W=(86-80)/sqrt(25/20+25/20)
> W

[1] 3.794733

> q1=25/20
> q2=25/20
> df=(q1+q2)^2/(q1^2/19+q2^2/19)
> df

[1] 38

> qt(.975,df)

[1] 2.024394
```

**8**. When the sample variances are approximately equal, Welch and T give very similar results.

**9**.

```
> n1=24
> n2=16
> g1=floor(.2*n1) # .2n1 rounded down
> g2=floor(.2*n2)
> h1=n1-2*g1  # number of observations left after trimming
> h2=n2-2*g2
> d1=(n1-1)*25/(h1*(h1-1))
> d2=(n2-1)*36/(h2*(h2-1))
> d1
```

```
[1] 2.395833

> d2

[1] 6

> Ty=(42-36)/sqrt(d1+d2)
> sqrt(d1+d2)  # the estimate of the standard error (used in Exercise 10)

[1] 2.897556

> Ty   # test statistic

[1] 2.07071

> df=(d1+d2)^2/((d1^2/(h1-1))+d2^2/(h2-1))
> df

[1] 16.08381

> qt(.975,df)  # critical value

[1] 2.119008
```

$|T_y| = 2.07$, which is less than the critical value, fail to reject.

**10**.

From Exercise 9, the standard error is 2.8976 and the degrees of freedom are 16.0838. So for $\alpha = 0.99$, the critical value and confidence interval are:

```
> crit=qt(.995,16.0838)
> crit

[1] 2.918772

> (42-36)-crit*2.8976

[1] -2.457433

> (42-36)+crit*2.8976

[1] 14.45743
```

**11**.

```
> xb1=10
> xb2=5
> n1=16
> n2=16
> s1sq=21
> s2sq=29
> q1=s1sq/n1
> q2=s2sq/n2
> df=(q1+q2)^2/(q1^2/(n1-1)+q2^2/(n2-1))
> crit=qt(.975,df)
> (xb1-xb2)-crit*sqrt(q1+q2)

[1] 1.385859

> (xb1-xb2)+crit*sqrt(q1+q2)

[1] 8.614141
```

Interval does not contain zero, reject. Or based on Tukey's three decision rule, assuming normality, decide that the first group has the larger population mean. But this must be tempered with the fact that methods based on means can result in poor control over the Type I error probability.

**12**.

```
> xb1=10
> xb2=5
> n1=16
> n2=16
> s1sq=21
> s2sq=29
> sqse=(n1-1)*s1sq+(n2-1)*s2sq #estimate of assumed commone variance
> sqse=sqrt(sqse*(1/n1+1/n2))
> T.value=(xb1-xb2)/sqrt(sqse) # test statistics, T
> df=n1+n2-2
> crit=qt(.975,df)
> crit

[1] 2.042272

> (xb1-xb2)-crit*sqrt(sqse)

[1] -1.354867

> (xb1-xb2)+crit*sqrt(sqse)

[1] 11.35487
```

The interval contains zero, fail to reject. Or based on Tukey's three decision rule, make no decision about which group has the larger population mean.

**13**.

```
> X=c(132,204,603,50,125,90,185,134)
> Y=c(92,-42,121,63,182,101,294,36)
> t.test(X,Y)  # or could use

        Welch Two Sample t-test

data:  X and Y
t = 1.1922, df = 11.193, p-value = 0.2579
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -71.17601 240.17601
sample estimates:
mean of x mean of y
  190.375   105.875

> yuen(X,Y,tr=0)

$n1
[1] 8

$n2
[1] 8

$est.1
[1] 190.375

$est.2
[1] 105.875

$ci
[1] -71.17601 240.17601

$p.value
[1] 0.2578553

$dif
[1] 84.5

$se
[1] 70.87876

$teststat
```

```
[1] 1.192177

$crit
[1] 2.19637

$df
[1] 11.19272
```

**14.**

```
> X=c(132,204,603,50,125,90,185,134)
> Y=c(92,-42,121,63,182,101,294,36)
> yuen(X,Y)

$n1
[1] 8

$n2
[1] 8

$est.1
[1] 145

$est.2
[1] 99.16667

$ci
[1] -34.77795 126.44461

$p.value
[1] 0.2325326

$dif
[1] 45.83333

$se
[1] 35.96011

$teststat
[1] 1.274561

$crit
[1] 2.241686

$df
[1] 9.572932
```

**15**. Not necessarily, power can be low. You may still reject with other methods that are sensitive to different features of the data.

**16**.

```
> A=c(11.1,12.2,15.5,17.6,13.0,7.5,9.1,6.6,9.5,18.0,12.6)
> B=c(18.6,14.1,13.8,12.1,34.1,12.0,14.1,14.5,12.6,12.5,19.8,13.4,
+ 16.8,14.1,12.9)
> t.test(A,B)

        Welch Two Sample t-test

data:  A and B
t = -1.966, df = 23.925, p-value = 0.061
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -7.4407184  0.1813244
sample estimates:
mean of x mean of y
 12.06364  15.69333
```

**17**.

```
> A=c(11.1,12.2,15.5,17.6,13.0,7.5,9.1,6.6,9.5,18.0,12.6)
> B=c(18.6,14.1,13.8,12.1,34.1,12.0,14.1,14.5,12.6,12.5,19.8,13.4,16.8,14.1,12.9)
> yuen(A,B)

$n1
[1] 11

$n2
[1] 15

$est.1
[1] 11.85714

$est.2
[1] 14.03333

$ci
[1] -5.511854  1.159473

$p.value
[1] 0.1762253

$dif
[1] -2.17619
```

```
$se
[1] 1.492984

$teststat
[1] 1.457612

$crit
[1] 2.234226

$df
[1] 9.802916
```

**18**.

```
> A=c(11.1,12.2,15.5,17.6,13.0,7.5,9.1,6.6,9.5,18.0,12.6)
> B=c(18.6,14.1,13.8,12.1,34.1,12.0,14.1,14.5,12.6,12.5,19.8,13.4,16.8,14.1,12.9)
> akp.effect(A,B,tr=0)

[1] -0.7363244
```

**19**.

```
> sq.p=(41-1)*.05^2+(41-1)*.06^2 #numerator of estimate of commona variance
> df=41+41-2
> sq.p=sq.p/(41+41-2) # estimate of common variance.\
> T.value=(1.12-1.09)/sqrt(sq.p*(1/41+1/41))
> T.value

[1] 2.459508

> qt(.975,df)

[1] 1.990063
```

Reject.

**20**. When the groups differ, the probability coverage for the CI using T can be inaccurate, if standard assumptions are not met.

**21**.

```
> x=c(41,38.4,24.4,25.9,21.9,18.3,13.1,27.3,28.5,-16.9,26,17.4,21.8,15.4,27.4,19.2,22.
> y=c(10.1,6.1,20.4,7.3,14.3,15.15,-9.9,6.8,28.2,17.9,-9,-12.9,14,6.6,12.1,15.7,39.9,-
> n1=length(x)
> n2=length(y)
> n1

[1] 23
```

```
> n2

[1] 22

> top=(n1-1)*var(x)+(n2-1)*var(y)
> bot=n1+n2-2
> top/bot

[1] 235.9172
```

**22**.

```
> (22.4-11)/sqrt(235.9)

[1] 0.7422341

> qt(.975,43) # from previous exercise, df=22+23-2

[1] 2.016692
```

**23**.

```
> x=c(41,38.4,24.4,25.9,21.9,18.3,13.1,27.3,28.5,-16.9,26,17.4,21.8,15.4,27.4,
+ 19.2,22.4, 17.7,26,29.4,21.4,26.6,22.7)
> y=c(10.1,6.1,20.4,7.3,14.3,15.15,-9.9,6.8,28.2,17.9,-9,-12.9,14,6.6,12.1,15.7,39.9,
+ -15.9,54.6,-14.7,44.1,-9)
> t.test(x,y,var.equal=TRUE)

        Two Sample t-test

data:  x and y
t = 2.4913, df = 43, p-value = 0.01667
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  2.173751 20.648581
sample estimates:
mean of x mean of y
 22.40435  10.99318
```

A concern is that the probability coverage of the confidence interval might be inaccurate.

**24**. The distributions probably differ, suggesting that population means differ. But from the point of view of Tukey's three decision rule, there is concern about making any decision about which group has the largest population mean.

**25**.

```
> rats=read.table('rat_data.txt') #The file can be downloaded from the
> #    author's web page.
> yuen(rats[,1],rats[,2],tr=0)
```

```
$n1
[1] 23

$n2
[1] 22

$est.1
[1] 22.40435

$est.2
[1] 11.00909

$ci
[1]  1.964178 20.826336

$p.value
[1] 0.01938477

$dif
[1] 11.39526

$se
[1] 4.635055

$teststat
[1] 2.458495

$crit
[1] 2.034729

$df
[1] 32.90912
```

**26**. Power might not be high enough to detect a practical difference in variances.

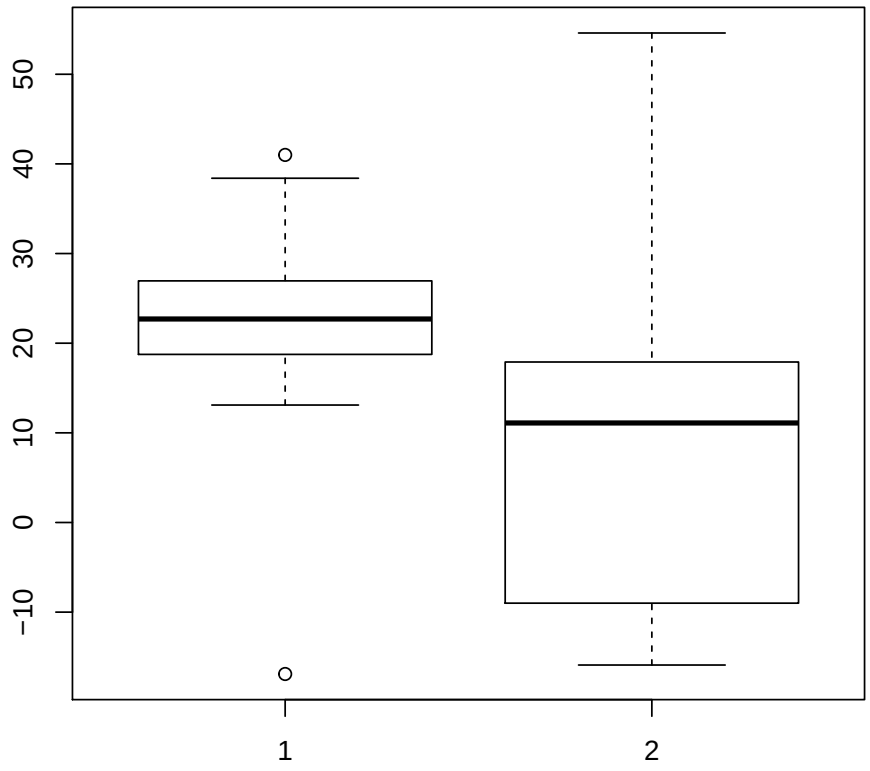
**27**.

```
> rats=read.table('rat_data.txt') #The file can be downloaded from the
> #    author's web page.
> yuen(rats[,1],rats[,2])

$n1
[1] 23

$n2
[1] 22
```

```
$est.1
[1] 23.26667

$est.2
[1] 9.2

$ci
[1]  5.282977 22.850357

$p.value
[1] 0.00374359

$dif
[1] 14.06667

$se
[1] 4.136851

$teststat
[1] 3.400332

$crit
[1] 2.12328

$df
[1] 15.69311
```

**28**.

```
> rats=read.table('rat_data.txt') #The file can be downloaded from the
> #    author's web page.
> boxplot(rats[,1],rats[,2])
```




The boxplots suggest that one distribution is reasonably symmetric while the other is not. Differences in skewness can result in inaccurate confidence intervals.

**29**.

```
> g1=c(77,87,88,114,151,210,219,246,253,262,296,299,306,376,428,515,666,1310,2611)
> g2=c(59,106,174,207,219,237,313,365,458,497,515,529,557,615,625,645,973,1065,3215)
> yuenbt(g1,g2,tr=0)

$ci
[1] -547.2861  246.2334

$test.stat
[1] -0.7213309

$p.value
[1] 0.524207

$est.1
[1] 448.1053
```

```
$est.2
[1] 598.6316

$est.dif
[1] -150.5263

$n1
[1] 19

$n2
[1] 19
```

**30**.

```
> g1=c(77,87,88,114,151,210,219,246,253,262,296,299,306,376,428,515,
+      666,1310,2611)
> g2=c(59,106,174,207,219,237,313,365,458,497,515,529,557,615,625,
+      645,973,1065,3215)
> comvar2(g1,g2)

[1] "Taking bootstrap samples. Please wait."
$ci
[1] -1124937.6   753191.4

$vardif
[1] -120219.8
```

**31**. The median will have a smaller se and therefore possibly more power when there are many outliers: for example, when dealing with heavy-tailed distribution. But for skewed distribution, it is possible that the population medians are approximately equal when other measures of location are not.

**32**.

```
> g1=c(77,87,88,114,151,210,219,246,253,262,296,299,306,376,428,515,666,1310,2611)
> g2=c(59,106,174,207,219,237,313,365,458,497,515,529,557,615,625,645,973,1065,3215)
> pb2gen(g1,g2,est=bivar)

$est.1
[1] 25481.9

$est.2
[1] 83567.29

$est.dif
[1] -58085.39
```

```
$ci
[1] -154718.19   50452.43

$p.value
[1] 0.191

$sq.se
[1] 3661815907

$n1
[1] 19

$n2
[1] 19
```

**33**. Discarding outliers that are in fact valid, the remaining values are dependent. Consequently, using a standard method based on means results in an erroneous estimate of standard error.

**34**.

```
> g1=c(1,2,1,1,1,1,1,1,1,1,2,4,1,1)
> g2=c(3,3,4,3,1,2,3,1,1,5,4)
> cidv2(g1,g2)  # CLiff's method

$n1
[1] 14

$n2
[1] 11

$d.hat
[1] -0.5779221

$d.ci
[1] -0.8271426 -0.1387524

$p.value
[1] 0.011

$p.hat
[1] 0.788961

$p.ci
[1] 0.5693762 0.9135713
```

```
$summary.dvals
        P(X<Y)    P(X=Y)     P(X>Y)
[1,] 0.6688312 0.2402597 0.09090909

> wmw(g1,g2)   #Wilcoxon--Mann-Whitney

$p.value
[1] 0.01484463

$adj.p.value
[1] 0.0118334

$p.hat
[1] 0.788961

> ks(g1,g2)

$test
[1] 0.5649351

$critval
[1] 0.5471947

$p.value
[1] 0.02024947
```

There are tied values suggesting that the Kolmogorov–Smirnov test might have relatively low power.

**35**.

```
> a=c(-25,-24,-22,-22,-21,-18,-18,-18,-18,-17,-16,
+     -14,-14,-13,-13,-13,-13,-9,-8,-7,-5,1,3,7)
> b=c(-21,-18,-16,-16,-16,-14,-13,-13,-12,-11,
+     -11,-11,-9,-9,-9,-9,-7,-6,-3,-2,3,10)
> wmw(a,b)

$p.value
[1] 0.05573169

$adj.p.value
[1] 0.05376009

$p.hat
[1] 0.6647727

> cidv2(a,b)
```

```
$n1
[1] 24

$n2
[1] 22

$d.hat
[1] -0.3295455

$d.ci
[1] -0.60380331  0.01447369

$p.value
[1] 0.061

$p.hat
[1] 0.6647727

$p.ci
[1] 0.4927632 0.8019017

$summary.dvals
        P(X<Y)     P(X=Y) P(X>Y)
[1,] 0.6420455 0.04545455 0.3125

> ks(a,b)

$test
[1] 0.344697

$critval
[1] 0.4008614

$p.value
[1] 0.09196079

> bmp(a,b)

$test.stat
[1] 2.003187

$phat
[1] 0.6647727

$dhat
[1] -0.3295455
```

```
$p.value
[1] 0.05223305

$ci.p
[1] 0.4983267 0.8312187

$df
[1] 38.50105
```

The WMW test is in fact testing the hypothesis that the groups have identical distributions. It is sensitive to differences that can be missed by other methods.

**36**.

```
> A=c(1.96,2.24,1.71,2.41,1.62,1.93)
> B=c(2.11,2.43,2.07,2.71,2.50,2.84,2.88)
> wmw(A,B)

$p.value
[1] 0.01515844

$adj.p.value
[1] 0.008684649

$p.hat
[1] 0.9047619

> cidv2(A,B)

$n1
[1] 6

$n2
[1] 7

$d.hat
[1] -0.8095238

$d.ci
[1] -0.9664164 -0.2130135

$p.value
[1] 0.01

$p.hat
[1] 0.9047619
```

```
$p.ci
[1] 0.6065067 0.9832082

$summary.dvals
        P(X<Y) P(X=Y)    P(X>Y)
[1,] 0.9047619      0 0.0952381
```

The estimate of the probability is $p.hat=0.9

**37**.

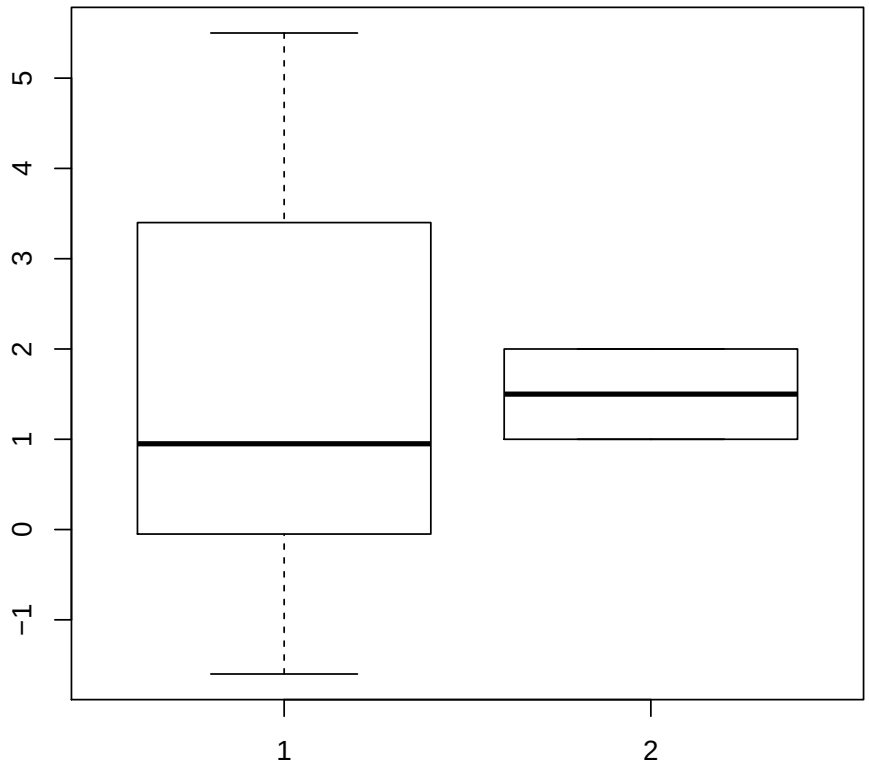
```
> A=c(1.96,2.24,1.71,2.41,1.62,1.93)
> B=c(2.11,2.43,2.07,2.71,2.50,2.84,2.88)
> bmp(A,B)

$test.stat
[1] 4.702908

$phat
[1] 0.9047619

$dhat
[1] -0.8095238

$p.value
[1] 0.0006558994

$ci.p
[1] 0.7152154 1.0943084

$df
[1] 10.94515
```

**38**.

```
>  #fac2list is a better and more
> #        recent function than selby. Could also use the
> #         the function split
>
> z=fac2list(sleep[,1],sleep[,2])

[1] "Group Levels:"
[1] 1 2

> boxplot(sleep[,1],sleep[,2])
> yuen(z[[1]],z[[2]])
```

```
$n1
[1] 10

$n2
[1] 10

$est.1
[1] 0.5333333

$est.2
[1] 2.2

$ci
[1] -4.0306400  0.6973066

$p.value
[1] 0.1433783

$dif
[1] -1.666667

$se
[1] 1.030857

$teststat
[1] 1.616777

$crit
[1] 2.293211

$df
[1] 8.264709

> yuen(z[[1]],z[[2]],tr=0)

$n1
[1] 10

$n2
[1] 10

$est.1
[1] 0.75

$est.2
```

```
[1] 2.33

$ci
[1] -3.3654832  0.2054832

$p.value
[1] 0.07939414

$dif
[1] -1.58

$se
[1] 0.849091

$teststat
[1] 1.860813

$crit
[1] 2.102817

$df
[1] 17.77647
```




No outliers are detected suggesting that the mean will have a smaller standard error, which in turn suggests that power might be higher using a mean. But differences in power also depend of skewness.

CHAPTER 8

**1**. The data can be downloaded from the author's web page.

```
> cork=read.table('corkall_dat.txt',header=TRUE)
> t.test(cork$E,cork$S,var.equal = TRUE)

        Two Sample t-test

data:  cork$E and cork$S
t = -0.77577, df = 54, p-value = 0.4413
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -12.545268   5.545268
sample estimates:
```

```
mean of x mean of y
 46.17857  49.67857
```

**2**.

```
> cork=read.table('corkall_dat.txt',header=TRUE)
> trimci(cork$E-cork$S)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
$ci
[1] -7.30965260 -0.02368073

$estimate
[1] -3.666667

$test.stat
[1] -2.12353

$se
[1] 1.726685

$p.value
[1] 0.04868652

$n
[1] 28
```

**3**.

```
> cork=read.table('corkall_dat.txt',header=TRUE)
> yuend(cork$E,cork$S)

$ci
[1] -8.657220  1.101665

$p.value
[1] 0.1207526

$est1
[1] 43.61111

$est2
[1] 47.38889
```

```
$dif
[1] -3.777778

$se
[1] 2.312734

$teststat
[1] -1.633468

$n
[1] 28

$df
[1] 17
```

**4**. The difference between the marginal trimmed means is, in general, not equal to the trimmed mean of the difference scores.

**5**. Yes. Again the difference between the marginal trimmed means is, in general, not equal to the trimmed mean of the difference scores. So power can be higher or lower using the marginal trimmed means rather than he trimmed mean of the difference scores.

**6**.

```
> cork=read.table('corkall_dat.txt',header=TRUE)
> trimcibt(cork$E-cork$S,tr=0)

$estimate
[1] -3.5

$ci
[1] -7.6260961  0.6260961

$test.stat
[1] -1.751449

$p.value
[1] 0.0918197

$n
[1] 28
```

**7**.

```
> cork=read.table('corkall_dat.txt',header=TRUE)
> trimcibt(cork$E-cork$S)

$estimate
[1] -3.666667
```

```
$ci
[1] -7.0710545 -0.2622789

$test.stat
[1] -2.12353

$p.value
[1] 0.03672788

$n
[1] 28
```

**8**.

```
> g1=c(10, 14,15, 18, 20, 29, 30, 40)
> g2=c(40, 8, 15, 20, 10, 8, 2, 3)
> wilcox.test(g1,g2,paired = TRUE)

        Wilcoxon signed rank test with continuity correction

data:  g1 and g2
V = 21, p-value = 0.2719
alternative hypothesis: true location shift is not equal to 0

> signt(g1,g2)

$Prob_x_less_than_y
[1] 0.2857143

$ci
[1] 0.07563576 0.64764872

$n
[1] 8

$N
[1] 7

$p.value
[1] 0.27
```

**9**.

```
> g1=c(86, 71, 77, 68, 91, 72, 77, 91, 70, 71, 88, 87)
> g2=c(88, 77, 76, 64, 96, 72, 65, 90, 65, 80, 81, 72)
> wilcox.test(g1,g2,paired = TRUE)
```

```
        Wilcoxon signed rank test with continuity correction

data:  g1 and g2
V = 41.5, p-value = 0.4765
alternative hypothesis: true location shift is not equal to 0
```

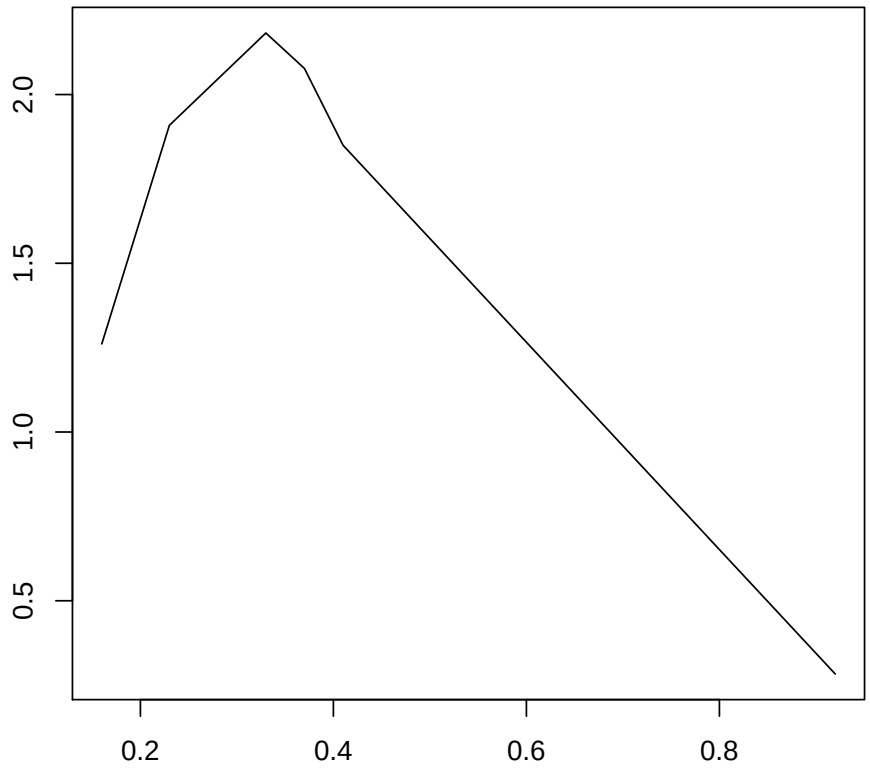
**10**.

```
> z=fac2list(Indometh[,3],Indometh[,2])

[1] "Group Levels:"
 [1]  1  2  3  4  5  6  7  8  9 10 11

> t.test(z[[2]],z[[3]],paired = TRUE)

        Paired t-test

data:  z[[2]] and z[[3]]
t = 3.668, df = 5, p-value = 0.01447
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1206732 0.6859935
sample estimates:
mean of the differences
              0.4033333
```

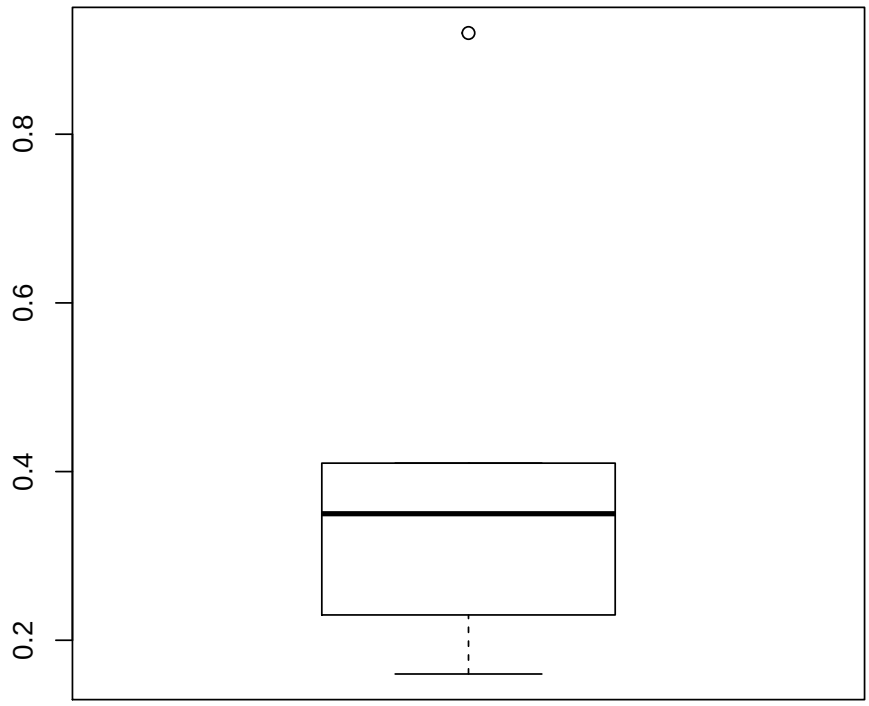
**11**.

```
> z=fac2list(Indometh[,3],Indometh[,2])

[1] "Group Levels:"
 [1]  1  2  3  4  5  6  7  8  9 10 11

> akerd(z[[2]]-z[[3]])

[1] "Done"

> trimcibt(z[[2]]-z[[3]])

$estimate
[1] 0.335

$ci
[1] 0.1267734 0.5432266

$test.stat
[1] 5.935775
```

```
$p.value
[1] 0.02337229

$n
[1] 6

> trimcibt(z[[2]]-z[[3]],tr=0)

$estimate
[1] 0.4033333

$ci
[1] 0.009749199 0.796917467

$test.stat
[1] 3.668014

$p.value
[1] 0.05008347

$n
[1] 6
```




```
> boxplot(z[[2]]-z[[3]])
```




**12**. Evidently the link to this data set is no longer available. So the data are stored on the author's web page.

```
> scent=read.table('scent_dat.txt',header=TRUE)
> trimci(scent[,7]-scent[,10],tr=0)

[1] "The p-value returned by the this function is based on the"
[1] "null value specified by the argument null.value, which defaults to 0"
[1] "To get a measure of effect size using a Winsorized measure of scale,  use trimciv2"
$ci
[1] -7.863156  4.644109

$estimate
[1] -1.609524

$test.stat
[1] -0.5368732

$se
```

```
[1] 2.997959

$p.value
[1] 0.5972777

$n
[1] 21
```

**13**.

```
> scent=read.table('scent_dat.txt',header=TRUE)
> yuend(scent[,7],scent[,10])

$ci
[1] -12.485669   3.193361

$p.value
[1] 0.2209227

$est1
[1] 51.21538

$est2
[1] 55.86154

$dif
[1] -4.646154

$se
[1] 3.598067

$teststat
[1] -1.291292

$n
[1] 21

$df
[1] 12
```

CHAPTER 9

**1**. Because there are equal sample sizes, the estimate of the assumed common variance is

```
> (6.214+3.982+2.214)/3
```

```
[1] 4.136667
```

**2**.

```
> x=read.table("CH9ex1.txt")
> anova1(x)

$F.test
[1] 6.053237

$p.value
[1] 0.00839879

$df1
[1] 2

$df2
[1] 21

$MSBG
[1] 25.04167

$MSWG
[1] 4.136905
```

**3**.

```
> x=read.table("CH9ex1.txt")
> t1way(x,tr=0)

$TEST
[1] 7.774918

$nu1
[1] 2

$nu2
[1] 13.40326

$n
[1] 8 8 8

$p.value
[1] 0.005733349
```

**4**.

```
> x=read.table("CH9ex1.txt")
> t1way(x)

$TEST
[1] 6.044463

$nu1
[1] 2

$nu2
[1] 9.661829

$n
[1] 8 8 8

$p.value
[1] 0.01983871
```

5.

```
> x=list()
> x[[1]]=c(15,17,22)
> x[[2]]=c(9,12,15)
> x[[3]]=c(17,20,23)
> x[[4]]=c(13,12,17)
> anova1(x)

$F.test
[1] 4.210526

$p.value
[1] 0.04615848

$df1
[1] 3

$df2
[1] 8

$MSBG
[1] 40

$MSWG
[1] 9.5

> v.vec=lapply(x,var)
> v.values=list2matrix(v.vec)
```

```
> MSWG=mean(v.values)
> MSWG

[1] 9.5
```

**6**. Roughly, tests of the assumption that there is homoscedasticity might not have enough power to detect situations where violating this assumption is a practical concern.

**7**. If the null hypothesis is true, MSBG estimates the assumed common variance. Otherwise it does not. But failing to reject does not mean the null hypothesis should be excepted.

**8**.

```
> x=list()
> x[[1]]=c(9,10,15)
> x[[2]]=c(16,8,13)
> x[[3]]=c(7,6,9)
> anova1(x)

$F.test
[1] 2.172414

$p.value
[1] 0.195112

$df1
[1] 2

$df2
[1] 6

$MSBG
[1] 21

$MSWG
[1] 9.666667
```

**9**. MSBG estimates

$$\sigma_p^2 + \frac{n\sum(\mu_j - \bar{\mu})^2}{J-1}$$

So for the situation at hand, the value estimated by MSBG can be determined with the following R commands:

```
> mus=c(3:7)
> 2+10*sum((mus-mean(mus))^2)/4

[1] 27
```

**10**.

```
> x=list()
> x[[1]]=c(10,11,12,9,8,7)
> x[[2]]=c(10,66,15,32,22,51)
> x[[3]]=c(1,12,42,31,55,19)
> anova1(x)

$F.test
[1] 2.960933

$p.value
[1] 0.08244822

$df1
[1] 2

$df2
[1] 15

$MSBG
[1] 867.3889

$MSWG
[1] 292.9444

> t1way(x,tr=0)

$TEST
[1] 5.013846

$nu1
[1] 2

$nu2
[1] 6.772878

$n
[1] 6 6 6

$p.value
[1] 0.04611879
```

**11**.

$$delta = a^2(J-1)/J$$

```
> delta=0.2^2*3/4
> x=read.table('skin_dat.txt',header=T,sep='&')
> bdanova1(x,delta=delta)
```

```
$N
[1] 110  22  40  38

$d
[1] 0.001524799

$crit
[1] 10.88554
```

**12**.

```
> x=read.table('skin_dat.txt',header=T,sep='&')
> pbadepth(x,est=mom)

$p.value
[1] 0.3665

$psihat
[1]  0.26505  0.26505 -0.46463  0.00000 -0.72968 -0.72968

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1

$n
[1] 10 10 10 10

> pbadepth(x,est=tmean)

$p.value
[1] 0.064

$psihat
[1] -0.03357333  0.07834500 -0.19335500  0.11191833 -0.15978167 -0.27170000

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1

$n
[1] 10 10 10 10
```

Even among robust estimators, which estimator is used can make a substantial difference.

**13**.

```
> x=read.table('skin_dat.txt',header=T,sep='&')
> pbadepth(x,est=mom,op=3,MC=T)

$p.value
[1] 0.905

$psihat
[1]  0.26505  0.26505 -0.46463  0.00000 -0.72968 -0.72968

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1

$n
[1] 10 10 10 10

> pbadepth(x,est=tmean,op=3,MC=T)

$p.value
[1] 0.0505

$psihat
[1] -0.03357333  0.07834500 -0.19335500  0.11191833 -0.15978167 -0.27170000

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1

$n
[1] 10 10 10 10
```

**14**.

```
> x=read.table('ch9_ex14_dat.txt',header=T)
> x=fac2list(x[,3],x[,2])

[1] "Group Levels:"
[1] 1 2 3
```

```
> t1way(x,tr=0)

$TEST
[1] 0.08116164

$nu1
[1] 2

$nu2
[1] 18.55514

$n
[1] 10 10 11

$p.value
[1] 0.9223701

> t1way(x)

$TEST
[1] 0.2490255

$nu1
[1] 2

$nu2
[1] 8.758583

$n
[1] 10 10 11

$p.value
[1] 0.7848971
```

**15**.

```
> x=read.table('ch9_table9_6_dat.txt',header = T,sep='&')
> t1way(x,tr=0)

$TEST
[1] 0.09829264

$nu1
[1] 4

$nu2
```

```
[1] 54.88754

$n
[1] 23 23 23 23 23

$p.value
[1] 0.9825719
```

**16**.

```
> x=read.table('ch9_table9_6_dat.txt',header = T,sep='&')
> t1way(x)

$TEST
[1] 16.20457

$nu1
[1] 4

$nu2
[1] 34.47712

$n
[1] 23 23 23 23 23

$p.value
[1] 1.475302e-07
```

**17**. As noted in the text, using pbadepth with default settings results in the error message:

```
Error in solve.default(cov, ...) :
  system is computationally singular:
   reciprocal condition number = 2.15333e-18
```

**18**.

```
> x=read.table('ch9_table9_6_dat.txt',header = T,sep='&')
> pbadepth(x,MC=T,op=3)

$p.value
[1] 0.004

$psihat
 [1]  0.31357324 -1.97102280 -1.84392956 -1.90082071 -2.28459604 -2.15750280
 [7] -2.21439396  0.12709324  0.07020209 -0.05689115
```

```
$con
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    1    1    0    0    0    0    0     0
[2,]   -1    0    0    0    1    1    1    0    0     0
[3,]    0   -1    0    0   -1    0    0    1    1     0
[4,]    0    0   -1    0    0   -1    0   -1    0     1
[5,]    0    0    0   -1    0    0   -1    0   -1    -1

$n
[1] 23 23 23 23 23
```

CHAPTER 10

**1**.

Main effects Factor A:

$$H_0 : \mu_1 + \mu_2 + \mu_3 + \mu_4 = \mu_5 + \mu_6 + \mu_7 + \mu_8.$$

Main effects Factor B:

$$H_0 : \mu_1 + \mu_5 = \mu_2 + \mu_6 = \mu_3 + \mu_7 = \mu_4 + \mu_8$$

No Interactions:

$$H_0 : \mu_1 - \mu_2 = \mu_5 - \mu_6,$$

$$H_0 : \mu_1 - \mu_3 = \mu_5 - \mu_7,$$

$$H_0 : \mu_1 - \mu_4 = \mu_5 - \mu_8,$$

$$H_0 : \mu_2 - \mu_3 = \mu_6 - \mu_7,$$

$$H_0 : \mu_2 - \mu_4 = \mu_6 - \mu_8,$$

$$H_0 : \mu_3 - \mu_4 = \mu_7 - \mu_8.$$

**2**.

Main effect Factor A: $110 + 70 \neq 80 + 40$
Main effect Factor B $110 + 80 \neq 70 + 40$
No interaction: $110 - 70 = 80 - 40$.

**3**.

There is an interaction: $10 - 20 \neq 40 - 10$ It is disordinal.

**4**.

Row 1: 10, 20, 30,
Row 2: 20, 30, 40, Row 3: 30, 40, 50.

**5**. Method 2 is more effective, ignoring gender. The estimate is that method 1 is better than method 2 for females, but the empirical evidence that this is indeed the case is weak. Or in terms of Tukey's three decision rule, make no decision about any interaction.

**6**. Now there is stronger empirical evidence that method 2 is better for males, but that method 1 is better for females.

**7**. No, more needs to be done to establish that a disordinal interaction exists. It is necessary to reject $H_0 : \mu_1 = \mu_2$ as well as $H_0 : \mu_3 = \mu_4$.

**8**. Ignoring abuse, hostility is higher when there is retaliation. Ignoring retaliation, hostility is higher when there is abuse. The decrease in hostility, going from Insult to Apology, is greater for no retaliation versus retaliation.

**9**.

```
> x=read.table('Snedecor_dat.txt',header=T,sep='&')
> pbad2way(2,2,x,est=median)

[1] "Taking bootstrap samples. Please wait."
$sig.levelA
[1] 0.277

$sig.levelB
[1] 0.059

$sig.levelAB
[1] 0.1315

$conA
     [,1]
[1,]    1
[2,]    1
[3,]   -1
[4,]   -1

$conB
     [,1]
[1,]    1
[2,]   -1
[3,]    1
[4,]   -1

$conAB
     [,1]
[1,]    1
[2,]   -1
[3,]   -1
[4,]    1
```

**10**.

```
> x=read.table('CRCch10_Ex10.txt',header=T,sep='&')
> z=fac2list(x[,3],x[,1:2])
```

```
[1] "Group Levels:"
     [,1] [,2]
[1,]    1    1
[2,]    1    2
[3,]    1    3
[4,]    2    1
[5,]    2    2
[6,]    2    3
```

```
> t2way(2,3,z,tr=0)
```

```
$Qa
[1] 0.8350773

$A.p.value
[1] 0.372

$df.A
[1] 1

$Qb
[1] 0.1050177

$B.p.value
[1] 0.952

$df.B
[1] 2

$Qab
[1] 1.998881

$AB.p.value
[1] 0.408

$df.AB
[1] 2

$means
         [,1] [,2]   [,3]
[1,] 35.80000 23.2 27.200
[2,] 29.66667 39.8 33.875
```

**11**.

```
> x=read.table('CRCch10_Ex10.txt',header=T,sep='&')
> z=fac2list(x[,3],x[,1:2])
```

```
[1] "Group Levels:"
     [,1] [,2]
[1,]    1    1
[2,]    1    2
[3,]    1    3
[4,]    2    1
[5,]    2    2
[6,]    2    3
```

```
> t2way(2,3,z,tr=0.1)
```

```
$Qa
[1] 0.8350773

$A.p.value
[1] 0.372

$df.A
[1] 1

$Qb
[1] 0.1050177

$B.p.value
[1] 0.952

$df.B
[1] 2

$Qab
[1] 1.998881

$AB.p.value
[1] 0.408

$df.AB
[1] 2

$means
         [,1] [,2]   [,3]
[1,] 35.80000 23.2 27.200
[2,] 29.66667 39.8 33.875
```

**12**.

```
> x=read.table('CRCch10_Ex10.txt',header=T,sep='&')
> z=fac2list(x[,3],x[,1:2])
```

```
[1] "Group Levels:"
     [,1] [,2]
[1,]    1    1
[2,]    1    2
[3,]    1    3
[4,]    2    1
[5,]    2    2
[6,]    2    3

> t2way(2,3,z,tr=0.2)

$Qa
[1] 2.634293

$A.p.value
[1] 0.145

$df.A
[1] 1

$Qb
[1] 0.1590158

$B.p.value
[1] 0.932

$df.B
[1] 2

$Qab
[1] 2.567354

$AB.p.value
[1] 0.365

$df.AB
[1] 2

$means
         [,1]     [,2]     [,3]
[1,] 29.33333 15.33333 26.66667
[2,] 28.50000 38.00000 33.00000
```

One possible reason is that boxplots indicate that there are outliers in the first two groups. Boxplots also indicate skewed distributions. Yet another possible reason is that the standard

errors of the means are all larger than the corresponding standard error of the 20% trimmed means.

**13**.

```
> plasma=read.table("plasma_dat.txt",skip=15)
> z=fac2list(plasma[,14],plasma[,2:3])

[1] "Group Levels:"
     [,1] [,2]
[1,]    1    1
[2,]    1    2
[3,]    1    3
[4,]    2    1
[5,]    2    2
[6,]    2    3

> pbad2way(2,3,z,est=mom)

[1] "Taking bootstrap samples. Please wait."
$sig.levelA
[1] 0.5045

$sig.levelB
[1] 0.513

$sig.levelAB
[1] 0.5465

$conA
     [,1]
[1,]    1
[2,]    1
[3,]    1
[4,]   -1
[5,]   -1
[6,]   -1

$conB
     [,1] [,2] [,3]
[1,]    1    1    0
[2,]   -1    0    1
[3,]    0   -1   -1
[4,]    1    1    0
[5,]   -1    0    1
[6,]    0   -1   -1
```

```
$conAB
     [,1] [,2] [,3]
[1,]    1    1    0
[2,]   -1    0    1
[3,]    0   -1   -1
[4,]   -1   -1    0
[5,]    1    0   -1
[6,]    0    1    1
```

In contrast to a 20% trimmed mean, for Factor B, fail to reject at the 0.05 level.

CHAPTER 11

**1**.

```
> w=read.table('eegall.dat',skip=2)
> bwtrim(2,4,w)

$Qa
[1] 0.7739295

$Qa.p.value
         [,1]
[1,] 0.390786

$Qb
[1] 34.7582

$Qb.p.value
             [,1]
[1,] 8.781447e-07

$Qab
[1] 2.361414

$Qab.p.value
          [,1]
[1,] 0.1148291

>
```

**2**.

```
> w=read.table('eegall.dat',skip=2)
> sppba(2,4,w,avg=FALSE)
```

```
[1] "As of Oct. 2014 the argument est defaults to tmean"
[1] "Taking bootstrap samples. Please wait."
$p.value
[1] 0.116

$psihat
[1] -0.517 -0.050  0.178 -0.012

$con
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1
[5,]   -1    0    0    0
[6,]    0   -1    0    0
[7,]    0    0   -1    0
[8,]    0    0    0   -1
```

**3**.

```
> w=read.table('eegall.dat',skip=2)
> sppba(2,4,w,est=onestep,avg=TRUE)

[1] "As of Oct. 2014 the argument est defaults to tmean"
[1] "Taking bootstrap samples. Please wait."
$p.value
[1] 0.38

$psihat
[1] -0.1244383

$con
     [,1]
[1,]    1
[2,]   -1
```

**4**. Significant results can depend crucially on which method is used. This illustrates the need to replicate studies in a manner that takes different perspective into account.

**5**.

```
> w=read.table('eegall.dat',skip=2)
> sppbi(2,4,w,est=tmean)

[1] "As of Oct. 2014, argument est defaults to tmean"
[1] "Taking bootstrap samples. Please wait."
```

```
$p.value
[1] 0.042

$psihat
[1] -0.393 -0.850 -0.405 -0.326  0.040  0.211

$con
      [,1] [,2] [,3] [,4] [,5] [,6]
 [1,]    1    0    0    0    0    0
 [2,]    0    1    0    0    0    0
 [3,]    0    0    1    0    0    0
 [4,]    0    0    0    1    0    0
 [5,]    0    0    0    0    1    0
 [6,]    0    0    0    0    0    1
 [7,]   -1    0    0    0    0    0
 [8,]    0   -1    0    0    0    0
 [9,]    0    0   -1    0    0    0
[10,]    0    0    0   -1    0    0
[11,]    0    0    0    0   -1    0
[12,]    0    0    0    0    0   -1
```

*> sppbi(2,4,w,est=onestep)*

```
[1] "As of Oct. 2014, argument est defaults to tmean"
[1] "Taking bootstrap samples. Please wait."
$p.value
[1] 0.388

$psihat
[1] -0.3983086 -0.7261643 -0.4215113 -0.3795496  0.1348505  0.1866788

$con
      [,1] [,2] [,3] [,4] [,5] [,6]
 [1,]    1    0    0    0    0    0
 [2,]    0    1    0    0    0    0
 [3,]    0    0    1    0    0    0
 [4,]    0    0    0    1    0    0
 [5,]    0    0    0    0    1    0
 [6,]    0    0    0    0    0    1
 [7,]   -1    0    0    0    0    0
 [8,]    0   -1    0    0    0    0
 [9,]    0    0   -1    0    0    0
[10,]    0    0    0   -1    0    0
[11,]    0    0    0    0   -1    0
[12,]    0    0    0    0    0   -1
```

CHAPTER 12

**1**. Controlling the probability of one or more Type I errors.

**2**.

a) With equal sample sizes, MSWG=(5+6+4+10+15)/5=11.6. So $T = |15-10|/\sqrt{11.6(1/20+1/20)} = 4.64$ $\nu = 100 - 5 = 95$, reject. Using R:

```
> MSWG=(5+6+4+10+15)/5
> T_test=(10-15)/sqrt(MSWG*(1/20+1/20))
> DF=100-5
> crit.val=qt(.975,DF)
```

b) $T = |15 - 10|/\sqrt{11.6(1/20 + 1/20)/2} = 6.565$, the critical value is $q = 3.9$, reject. Using R:

```
> MSWG=(5+6+4+10+15)/5
> T_test=(10-15)/sqrt(MSWG*(1/20+1/20))
```

(c) $W = (15 - 10)/\sqrt{4/20 + 9/20} = 6.2$, $\hat{\nu} = 33$, $c = 2.99$, reject.

(d) $(15 - 10)/\sqrt{.5(4/20 + 9/20)} = 8.77$, $q = 4.1$, reject.

(e) $f = 2.47$, $S = \sqrt{4(2.47)(11.6)(1/20 + 1/20)} = 3.39$, reject.

**3**. MSWG=8. (a) $T = |20\text{-}12|/\sqrt{8(1/10 + 1/10)} = 6.325$ $\nu = 50 - 5 = 45$, reject.

(b) $T = |20 - 12|/\sqrt{8(1/10 + 1/10)/2} = 8.94$, $q = 4.01$, reject.

(c) $W = (20 - 12)/\sqrt{5/10 + 6/10} = 7.63$, $\hat{\nu} = 37.7$, $c = 2.96$, reject.

(d) $(20 - 12)/\sqrt{.5(5/10 + 6/10)} = 10.79$, $q = 4.06$, reject.

(e) $f = 2.58$, $S = \sqrt{4(2.58)(8)(1/10 + 1/10)} = 4.06$, reject.

(f) $f = 2.62$, $A = 11.3$, reject.

**4**. Reject if the p−value is less than or equal to $0.05/6 = 00.0083$. So the fourth test is significant.

**5**. Put the p-values in descending order yielding: 0.400, 0.150, 0.100, 0.070, 0.010, 0.001. Referring to Table 12.4, only the smallest p-value is less than the corresponding critical p-value, which is 0.01020. So reject the fourth hypothesis only.

**6**. None are rejected.

**7**. All are rejected.

**8**. MSWG goes up and eventually you will no longer reject when comparing groups 1 and 2.

**9**. The same problem as in Exercise 8 occurs.

**10**. The sample variances are given in Table 9.1: $s_1^2 = 0.1676608$, $s_2^2 = 0.032679$, $s_3^2 = 0.0600529$, $s_4^2 = 0.0567414$

```
> sv=c(0.1676608,0.032679,0.0600529,0.0567414) # sample variances
> m=0.5/2 # margina of error
> d=(m/4.3)^2
> floor(sv/d)+1

[1] 50 10 18 17
```

So, $N_1 = 50$, $N_2 = 11$, $N_3 = 18$ and $N_4 = 17$ are the required samples. Using R:

```
> x=read.table('skin_dat.txt',header=TRUE,sep='&')
> tamhane(x,cil=.5,crit=4.3)

$n.vec
[1] 50 11 18 17

$ci.mat
[1] NA
```

**11**.

```
> source('hochberg_chk.tex') # Why is this command here?
> #  answer: updated version of hochberg not yet added to WRS
>
> x=read.table('skin_dat.txt',header=TRUE,sep='&')
> hochberg(x,cil=.5,crit=4.3,tr=0)  # Using means.

[1] "If using the tables of Studentized range distribution,"
[1] "the degrees of freedom are: 9"
[1] "Need an additional  40  observations for group 1"
[1] "Need an additional  0  observations for group 2"
[1] "Need an additional  8  observations for group 3"
[1] "Need an additional  7  observations for group 4"
$ci.mat
NULL

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1
```

```
> hochberg(x,cil=.5,crit=4.3,tr=0.2) #Using 20% trimmed means

[1] "If using the tables of Studentized range distribution,"
[1] "the degrees of freedom are: 5"
[1] "Need an additional  5  observations for group 1"
[1] "Need an additional  12  observations for group 2"
[1] "Need an additional  0  observations for group 3"
[1] "Need an additional  0  observations for group 4"
$ci.mat
NULL

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1
```

**12**.

```
> x=read.table('skin_dat.txt',header=TRUE,sep='&')
> lincon(x)

[1] "Note: confidence intervals are adjusted to control FWE"
[1] "But p-values are not adjusted to control FWE"
[1] "Adjusted p-values can be computed with the R function p.adjusted"
$n
[1] 10 10 10 10

$test
     Group Group      test     crit         se       df
[1,]     1     2 0.2909503 3.217890 0.11539200 9.627086
[2,]     1     3 0.8991534 3.306229 0.08713196 8.568838
[3,]     1     4 2.2214991 3.307845 0.08703807 8.550559
[4,]     2     3 1.1074715 3.406642 0.10105753 7.609956
[5,]     2     4 1.5823636 3.408688 0.10097658 7.593710
[6,]     3     4 4.0624997 3.190010 0.06688001 9.999866

$psihat
     Group Group      psihat   ci.lower    ci.upper     p.value
[1,]     1     2 -0.03357333 -0.4048921  0.33774542 0.777260838
[2,]     1     3  0.07834500 -0.2097332  0.36642319 0.393136653
[3,]     1     4 -0.19335500 -0.4812635  0.09455346 0.054949443
[4,]     2     3  0.11191833 -0.2323484  0.45618510 0.301863181
[5,]     2     4 -0.15978167 -0.5039794  0.18441602 0.154235781
[6,]     3     4 -0.27170000 -0.4850479 -0.05835214 0.002277467
```

**13**.

```
> x=read.table('skin_dat.txt',header=TRUE,sep='&')
> msmed(x)

[1] "Warning: Group 1 has tied values. Might want to used medpb"
[1] "WARNING: tied values detected."
[1] "Estimate of standard error might be highly inaccurate, even with n large"
[1] "Warning: Group 2 has tied values. Might want to used medpb"
[1] "WARNING: tied values detected."
[1] "Estimate of standard error might be highly inaccurate, even with n large"
[1] "Warning: Group 3 has tied values. Might want to used medpb"
[1] "WARNING: tied values detected."
[1] "Estimate of standard error might be highly inaccurate, even with n large"
$test
     Group Group      test crit        se   p.value
[1,]     1     2 0.1443384 2.63 0.2742167 0.8852624
[2,]     1     3 0.4313943 2.63 0.3025886 0.6662748
[3,]     1     4 0.5795942 2.63 0.3019699 0.5623190
[4,]     2     3 0.9624726 2.63 0.1767479 0.3360452
[5,]     2     4 0.7709186 2.63 0.1756865 0.4409375
[6,]     3     4 1.4059718 2.63 0.2173265 0.1600434

$psihat
     Group Group    psihat   ci.lower  ci.upper
[1,]     1     2 -0.039580 -0.7607699 0.6816099
[2,]     1     3  0.130535 -0.6652731 0.9263431
[3,]     1     4 -0.175020 -0.9692008 0.6191608
[4,]     2     3  0.170115 -0.2947319 0.6349619
[5,]     2     4 -0.135440 -0.5974955 0.3266155
[6,]     3     4 -0.305555 -0.8771238 0.2660138

> medpb(x)

$output
     con.num    psihat p.value  p.crit  ci.lower  ci.upper
[1,]       1 -0.039580  0.8405 0.05000 -0.348410  0.389345
[2,]       2  0.130535  0.4655 0.02500 -0.204745  0.442295
[3,]       3 -0.175020  0.1165 0.01020 -0.582645  0.150560
[4,]       4  0.170115  0.4175 0.01690 -0.213720  0.369515
[5,]       5 -0.135440  0.2090 0.01270 -0.582645  0.103590
[6,]       6 -0.305555  0.0035 0.00851 -0.672385 -0.040115

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
```

```
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1

$num.sig
[1] 1
```

So using a method designed for tied values, now groups 3 and 4 differ significantly.

**14**.

```
> x=read.table('skin_dat.txt',header=TRUE,sep='&')
> apply(x,2,bootse,est=hd)

  No.Schiz. Schizotypal Schiz..Neg. Schiz..Pos.
 0.10873670  0.08713257  0.06145651  0.07776561

> apply(x,2,trimse)

  No.Schiz. Schizotypal Schiz..Neg. Schiz..Pos.
 0.07036502  0.08589388  0.04558919  0.04542282
```

For all four groups,the standard error is smallest when using a 20% trimmed mean.

**15**.

```
> x=read.table('skin_dat.txt',header=TRUE,sep='&')
> rmmcppb(x)

[1] "dif=T, so analysis is done on difference scores"
$output
     con.num      psihat p.value      p.crit   ci.lower   ci.upper
[1,]       1  0.02696063      NA 0.050000000 -0.2617753         NA
[2,]       2  0.08984625      NA 0.025000000 -0.2566880 0.58240646
[3,]       3 -0.15881856      NA 0.016666667 -0.3178867 0.11650100
[4,]       4  0.11328189      NA 0.012500000 -0.2831895         NA
[5,]       5 -0.20488428      NA 0.010000000 -0.4726536 0.11961500
[6,]       6 -0.25383611      NA 0.008333333 -0.5939699 0.04523077

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1

$num.sig
[1] NA
```

```
> rmmcppb(x,est=mom)

[1] "dif=T, so analysis is done on difference scores"
$output
     con.num   psihat p.value      p.crit ci.lower ci.upper
[1,]       1  0.26505  0.9895 0.050000000 -0.34841  1.02297
[2,]       2  0.26505  0.7930 0.025000000 -0.79480  1.34099
[3,]       3 -0.46463  0.6240 0.012500000 -0.46463  0.46567
[4,]       4  0.00000  0.7785 0.016666667 -0.79480  0.44428
[5,]       5 -0.72968  0.5970 0.010000000 -0.72968  0.12777
[6,]       6 -0.72968  0.3770 0.008333333 -0.87532  0.44577

$con
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    1    1    0    0    0
[2,]   -1    0    0    1    1    0
[3,]    0   -1    0   -1    0    1
[4,]    0    0   -1    0   -1   -1

$num.sig
[1] 0
```

When using a one-step M-estimator, bootstrap samples can have MAD=0, which in turn means dividing by zero when computing the one-step M-estimator.

**16**.

```
> x=read.table('skull_data.txt')
> z=fac2list(x[,1],x[,5])

[1] "Group Levels:"
[1] 1 2 3 4 5

> lincon(z,tr=0)

[1] "Note: confidence intervals are adjusted to control FWE"
[1] "But p-values are not adjusted to control FWE"
[1] "Adjusted p-values can be computed with the R function p.adjusted"
$n
[1] 30 30 30 30 30

$test
      Group Group      test     crit        se       df
 [1,]     1     2 0.7789241 2.903874 1.2838221 57.76223
 [2,]     1     3 2.7390141 2.917553 1.1317941 51.04095
 [3,]     1     4 3.5070561 2.910587 1.1785763 54.25569
```

```
 [4,]     1     5 3.5471112 2.903632 1.3532139 57.89699
 [5,]     2     3 1.9371418 2.913550 1.0840714 52.84016
 [6,]     2     4 2.7659438 2.907667 1.1328261 55.72730
 [7,]     2     5 2.8929011 2.904612 1.3135603 57.35491
 [8,]     3     4 1.0796425 2.904888 0.9571069 57.20364
 [9,]     3     5 1.4587040 2.920426 1.1654181 49.82304
[10,]     4     5 0.5505538 2.912855 1.2109019 53.16578

$psihat
      Group Group     psihat  ci.lower    ci.upper      p.value
 [1,]     1     2 -1.0000000 -4.728058  2.72805781 0.4392039462
 [2,]     1     3 -3.1000000 -6.402069  0.20206887 0.0084662393
 [3,]     1     4 -4.1333333 -7.563683 -0.70298384 0.0009181630
 [4,]     1     5 -4.8000000 -8.729236 -0.87076449 0.0007796297
 [5,]     2     3 -2.1000000 -5.258496  1.05849612 0.0580821056
 [6,]     2     4 -3.1333333 -6.427215  0.16054796 0.0076845001
 [7,]     2     5 -3.8000000 -7.615383  0.01538268 0.0053865918
 [8,]     3     4 -1.0333333 -3.813622  1.74695545 0.2848325993
 [9,]     3     5 -1.7000000 -5.103517  1.70351741 0.1509224471
[10,]     4     5 -0.6666667 -4.193848  2.86051459 0.5842456082
```

*> tmcppb(z)*

```
$output
      con.num     psihat p.value  p.crit  ci.lower    ci.upper
 [1,]       1 -0.7777778 0.53250 0.02500 -4.055556  2.61111111
 [2,]       2 -3.6666667 0.00550 0.00851 -6.833333 -0.05555556
 [3,]       3 -3.8888889 0.00425 0.00639 -7.888889 -0.11111111
 [4,]       4 -4.9444444 0.00125 0.00511 -9.000000 -0.94444444
 [5,]       5 -2.8888889 0.00625 0.01020 -5.500000  0.00000000
 [6,]       6 -3.1111111 0.00550 0.00730 -6.388889  0.00000000
 [7,]       7 -4.1666667 0.00150 0.00568 -7.722222 -0.50000000
 [8,]       8 -0.2222222 0.80125 0.05000 -3.777778  2.83333333
 [9,]       9 -1.2777778 0.29350 0.01270 -4.833333  2.38888889
[10,]      10 -1.0555556 0.45450 0.01690 -4.611111  2.88888889

$con
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    1    1    1    0    0    0    0    0     0
[2,]   -1    0    0    0    1    1    1    0    0     0
[3,]    0   -1    0    0   -1    0    0    1    1     0
[4,]    0    0   -1    0    0   -1    0   -1    0     1
[5,]    0    0    0   -1    0    0   -1    0   -1    -1

$num.sig
[1] 6
```

Based on the confidence intervals, comparing means results in two significant results. Using the percentile bootstrap with a 20% trimmed mean results in six significant results.

**17**. One possibility is that the boxplot rule is missing outliers. Using the MAD-median rule, group 3 is found to have four outliers, but none are found based on the boxplot rules in Chapter 3. Comparing the standard errors of the means versus the standard errors when using a 20% trimmed mean, sometimes a 20% trimmed has a smaller standard error, but sometimes the reverse is true. Differences in skewness can affect power when comparing means even when there are no outliers.

CHAPTER 13

**1**. Other projections might find an outlier.That is, this method does not take into account the overall structure of the data.

**2**.

The commands

```
sk=read.table('skull_data.txt')
z=mat2list(sk[,1:4],sk[,5])
smean2(z[[1]][,1:2],z[[5]][,1:2],MC=TRUE)
```

return p.value=0

**3**.

```
> set.seed(3)
> x=rmul(100,p=5,mar.fun=ghdist,h=.2,rho=0.7)
> x[,1]=rlnorm(100)
> regpca(x)

Importance of components:
                          Comp.1    Comp.2     Comp.3     Comp.4    Comp.5
Standard deviation     1.7586307 1.0130181 0.58867094 0.57353529 0.4533608
Proportion of Variance 0.6185564 0.2052411 0.06930669 0.06578855 0.0411072
Cumulative Proportion  0.6185564 0.8237976 0.89310425 0.95889280 1.0000000

Loadings:
     Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
[1,]         0.982         0.112 -0.138
[2,] -0.487 -0.109 -0.823  0.195 -0.190
[3,] -0.517  0.136        -0.264  0.802
[4,] -0.505         0.279 -0.612 -0.541
[5,] -0.490         0.490  0.711
```

**4**.

```
> set.seed(3)
> x=rmul(100,p=5,mar.fun=ghdist,h=.2,rho=0.7)
> x[,1]=rlnorm(100)
> y=regpca(x,SCORES=T)
> cor(y)

              Comp.1        Comp.2        Comp.3        Comp.4        Comp.5
Comp.1  1.000000e+00 -6.472480e-17 -1.398659e-16  2.437291e-16 -6.172381e-16
Comp.2 -6.472480e-17  1.000000e+00 -4.268471e-17 -1.142471e-16  1.046479e-16
Comp.3 -1.398659e-16 -4.268471e-17  1.000000e+00  1.143805e-15 -1.019519e-15
Comp.4  2.437291e-16 -1.142471e-16  1.143805e-15  1.000000e+00  1.550826e-15
Comp.5 -6.172381e-16  1.046479e-16 -1.019519e-15  1.550826e-15  1.000000e+00
```

**5**.

Both functions return the following results:

```
$n
[1] 100

$n.out
[1] 13

$out.id
 [1]   6 20  26  42  50  59  64  74  79  82  92  95 100
```

This is not surprising because there are only five variables.

The command
outproad(x,pr=FALSE)
returns the same results.

**6**.

The commands
y=mat2grp(plasma,2)
smean2(y[[1]][,11:12],y[[2]][,11:12],MC=TRUE)
return a p-value equal to 0.316.

CHAPTER 14

**1**.

The function returns NA because there no values for the independent variable that are close to 250. The value 250 is larger than any of the values that are available and the function does not extrapolate.

**2**.

```
> lake=read.table('lake_dat.txt',skip=4,header=T)
> lplot(lake[,3],lake[,2])
```

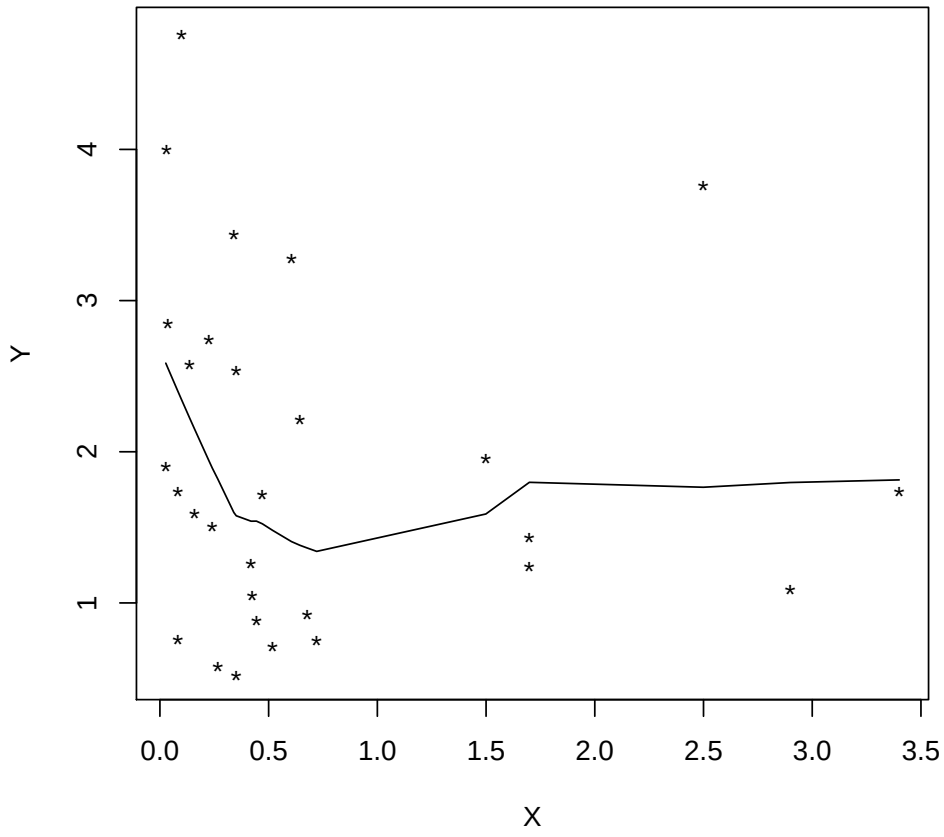
```
$Strength.Assoc
[1] 0.3894342

$Explanatory.power
[1] 0.151659

$yhat.values
NULL

$n
[1] 29

$n.keep
[1] 29
```

*> regci(lake[,3],lake[,2],xout=F)*

```
[1] "Duplicate values detected; tshdreg might have more power than tsreg"
[1] "Taking bootstrap samples. Please wait."
$regci
             ci.low     ci.up   Estimate      S.E.   p-value
Intercept  1.191617 2.9182781  2.0473646 0.4786042 0.0000000
Slope 1   -1.572917 0.1763668 -0.4693141 0.4795832 0.1886477

$n
[1] 29

$n.keep
[1] 29
```

*> regci(lake[,3],lake[,2],xout=T)*

```
[1] "Default for outfun is now outpro, not out"
[1] "Taking bootstrap samples. Please wait."
$regci
             ci.low       ci.up  Estimate      S.E.    p-value
Intercept  1.616149  3.62878641  2.117727 0.5516302 0.00000000
Slope 1   -4.548287 -0.07661966 -1.883117 1.0403851 0.02671119

$n
[1] 29

$n.keep
[1] 23
```




Removing leverage points can substantially impact the test of a zero slope even when using a robust regression estimator.

**3**.

```
> lake=read.table('lake_dat.txt',skip=4,header=T)
> lplot(lake[,1],lake[,2])

$Strength.Assoc
[1] 0.7847085

$Explanatory.power
[1] 0.6157674

$yhat.values
NULL

$n
[1] 29
```

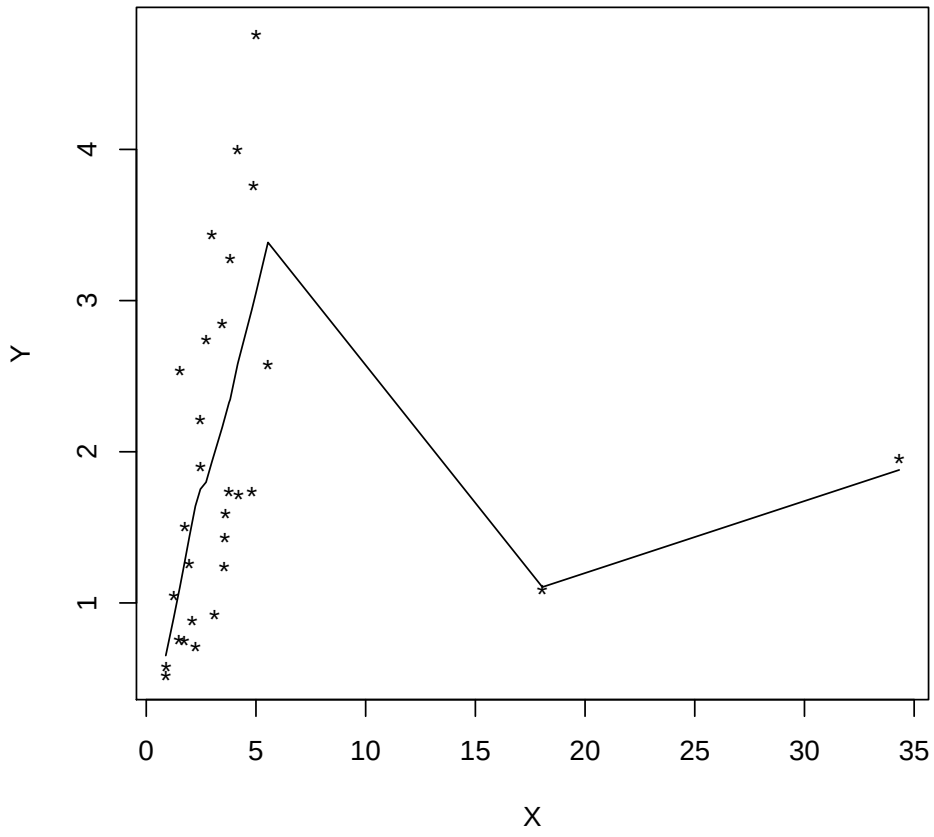
```
$n.keep
[1] 29
```

```
> regci(lake[,1],lake[,2],xout=F)
```

```
[1] "Duplicate values detected; tshdreg might have more power than tsreg"
[1] "Taking bootstrap samples. Please wait."
$regci
               ci.low     ci.up  Estimate      S.E.   p-value
Intercept -0.23454695 1.4803754 0.3481192 0.3972529 0.2103506
Slope 1    0.05342432 0.8591481 0.3702308 0.1804933 0.0000000

$n
[1] 29

$n.keep
[1] 29
```

```
> regci(lake[,1],lake[,2],xout=T)
```

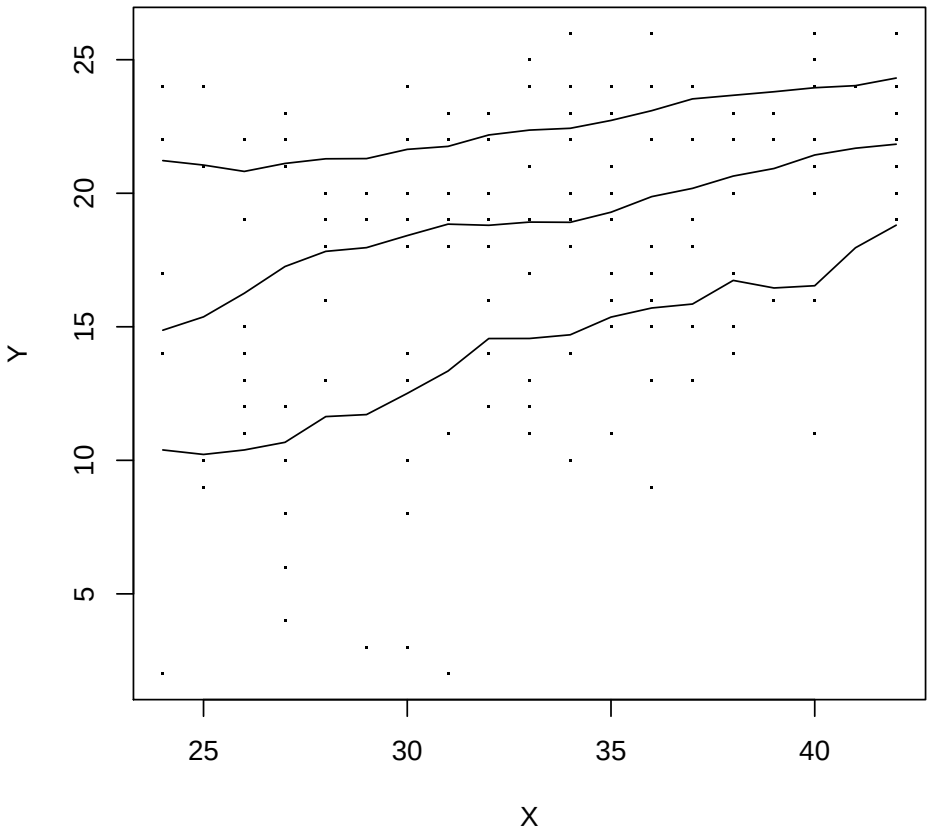
```
[1] "Default for outfun is now outpro, not out"
[1] "Duplicate values detected; tshdreg might have more power than tsreg"
[1] "Taking bootstrap samples. Please wait."
$regci
              ci.low     ci.up  Estimate      S.E.   p-value
Intercept -0.4475665 0.8010914 0.1406369 0.3078001 0.5475793
Slope 1    0.3099490 0.9498264 0.4921830 0.1796407 0.0000000

$n
[1] 29

$n.keep
[1] 27
```




In this case, there are obvious leverage points, they impact the estimate somewhat, but in both cases, reject at the 0.001 level.

**4**.

```
> A3=read.table('A3_dat.txt',header=T)
> qhdsm(A3$MAPAGLOB,A3$LSIZ,xout=T,q=c(.2,.5,.8))

[1] "DONE"
```




The R function khomreg assumes that the regression lines are straight, which appears to be reasonable approximation in this particular case.

**5**.

```
> A3=read.table('A3_dat.txt',header=T)
> qregci(A3$MAPAGLOB,A3$LSIZ,xout=T,q=.8)

[1] "Default for argument outfun is now outpro"
[1] "Taking bootstrap samples. Please wait."
$test
[1] 1.880628

$se.val
[1] 0.1063475

$crit.val
[1] 1.857601

$crit.fwe
```

```
[1] 1.857601

$slope.est
[1] 0.2

$ci
     Quantile  ci.lower  ci.upper
[1,]      0.8 0.0024488 0.3975512
```

**6**.

```
> B3=read.table('B3_dat.txt',header=T)
> qhdsm(B3$MAPAGLOB,B3$LSIZ,xout=T,q=c(.2,.5,.8))

[1] "DONE"
```

There is a hint of curvature for the .2 and .8 quantiles, but it is not very striking

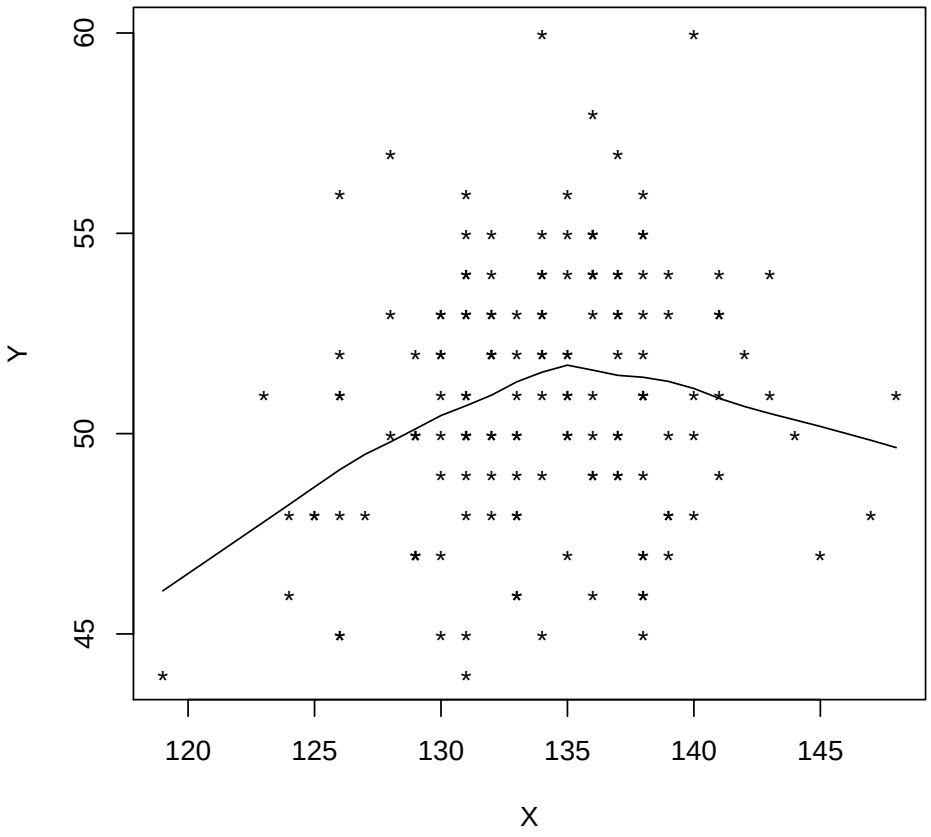
**7**.

```
> B3=read.table('B3_dat.txt',header=T)
> qregci(B3$MAPAGLOB,B3$LSIZ,xout=T,q=.8)

[1] "Default for argument outfun is now outpro"
[1] "Taking bootstrap samples. Please wait."
$test
[1] 3.818372

$se.val
[1] 0.06235518

$crit.val
[1] 1.842089

$crit.fwe
[1] 1.842089

$slope.est
[1] 0.2380952

$ci
     Quantile  ci.lower ci.upper
[1,]      0.8 0.1232314 0.352959
```

Consistent with the results in Exercise 5, reject at the 0.05 level.

**8**.

```
> z=read.table('skull_data.txt')
> lplot(z[,1],z[,4])

$Strength.Assoc
[1] 0.1745025

$Explanatory.power
[1] 0.03045113

$yhat.values
NULL

$n
[1] 150

$n.keep
[1] 150

> linchk(z[,1],z[,4],sp=135)

[1] "Splitting data using predictor 1"
$n
[1] 90 60

$output
     Parameter     ci.lower   ci.upper     p.value   Group 1 Group 2
[1,]         0 -133.1969697 -19.642857 0.008347245 7.0000000   86.25
[2,]         1    0.1428571   0.969697 0.010016694 0.3333333   -0.25
```

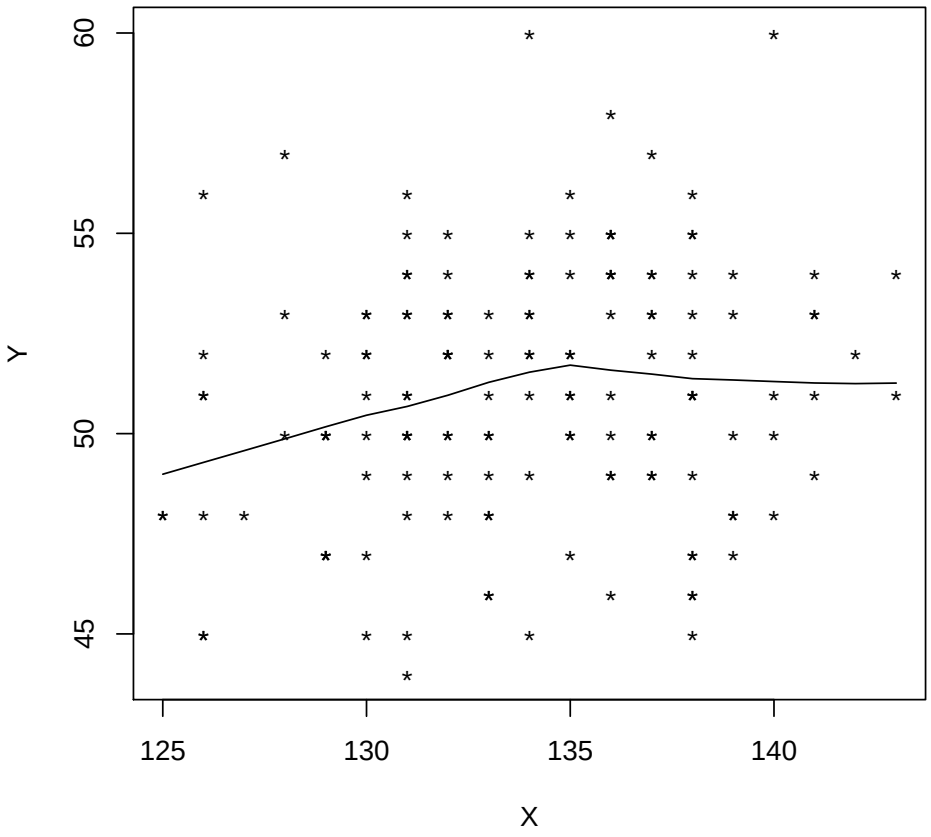



**9**.

Leverage points might impact the results

```
> z=read.table('skull_data.txt')
> lplot(z[,1],z[,4],xout=T)
```

```
$Strength.Assoc
[1] 0.1616033

$Explanatory.power
[1] 0.02611562

$yhat.values
NULL

$n
[1] 150

$n.keep
[1] 142
```




Here are the results when using the command linchk(z[,1],z[,4],sp=135,xout=T)

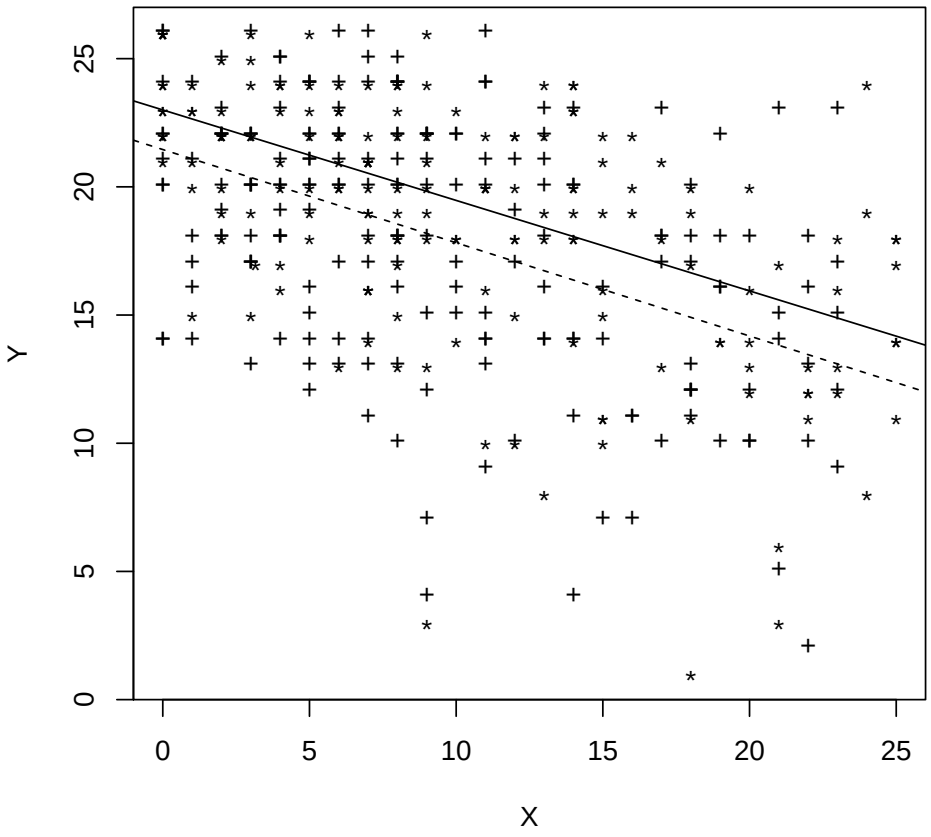
```
$n
[1] 86 53

$output
     Parameter ci.lower ci.upper    p.value    Group 1    Group 2
[1,]         0   -191.6      0.0 0.05509182 13.2857143 97.6666667
[2,]         1      0.0      1.4 0.06176962  0.2857143 -0.3333333
```

Now `linchk` no longer rejects at the .05 level.

**10**.

```
> A3=read.table('A3_dat.txt',header=T)
> B1=read.table('B1_dat.txt',header=T)
> ancJN(A3$CESD,A3$LSIZ,B1$CESD,B1$LSIZ,xout=T)

$n
[1] 186 227
```

```
$intercept.slope.group1
 Intercept
23.0000000 -0.3529412

$intercept.slope.group2
 Intercept
21.4545455 -0.3636364

$output
         X     Est1     Est2      DIF     TEST        se     ci.low    ci.hi
[1,]  0.00 23.00000 21.45455 1.545455 2.238139 0.6905089 -0.2291533 3.320062
[2,]  7.25 20.44118 18.81818 1.622995 2.742830 0.5917227  0.1022674 3.143722
[3,] 14.50 17.88235 16.18182 1.700535 2.242834 0.7582081 -0.2480600 3.649130
[4,] 21.75 15.32353 13.54545 1.778075 1.657319 1.0728625 -0.9791817 4.535331
[5,] 29.00 12.76471 10.90909 1.855615 1.287071 1.4417347 -1.8496431 5.560873
         p.value
[1,] 0.025212020
[2,] 0.006091223
[3,] 0.024907533
[4,] 0.097455091
[5,] 0.198069518
```



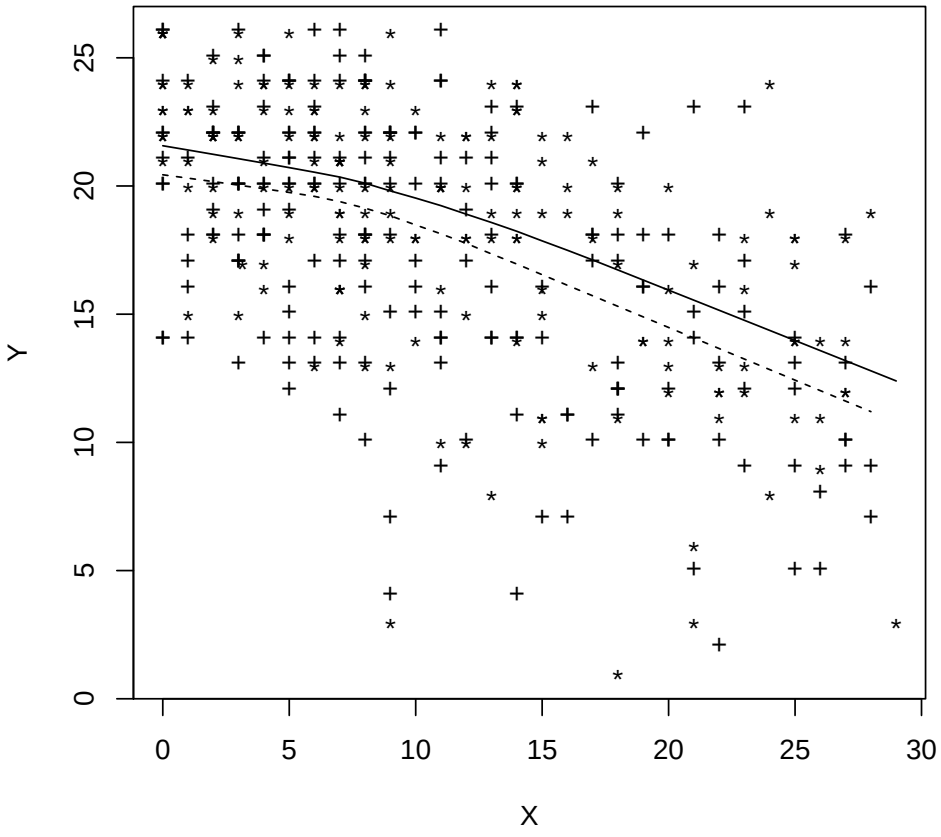

**11**.

```
> A3=read.table('A3_dat.txt',header=T)
> B1=read.table('B1_dat.txt',header=T)
> res=ancova(A3$CESD,A3$LSIZ,B1$CESD,B1$LSIZ,xout=T)$output

[1] "NOTE: Confidence intervals are adjusted to control the probability"
[1] "of at least one Type I error."
[1] "But p-values are not"

> res

      X  n1  n2       DIF     TEST        se     ci.low    ci.hi    p.value
[1,]  0  70  76 1.2867495 2.190970 0.5872967 -0.2495942 2.823093 0.03117602
[2,]  5 107 130 0.8692308 1.695792 0.5125809 -0.4557004 2.194162 0.09215378
[3,]  9 116 126 0.9266917 1.575100 0.5883384 -0.5946989 2.448082 0.11754650
[4,] 16  78  72 1.5378788 1.634208 0.9410543 -0.9207061 3.996464 0.10573964
[5,] 29  26  20 2.6250000 1.957756 1.3408207 -1.0811391 6.331139 0.06163288
     crit.val
```

```
[1,] 2.615958
[2,] 2.584824
[3,] 2.585911
[4,] 2.612586
[5,] 2.764083
```

```
> p.adjust(res[,9])
```

```
[1] 0.1558801 0.2764613 0.2764613 0.2764613 0.2465315
```

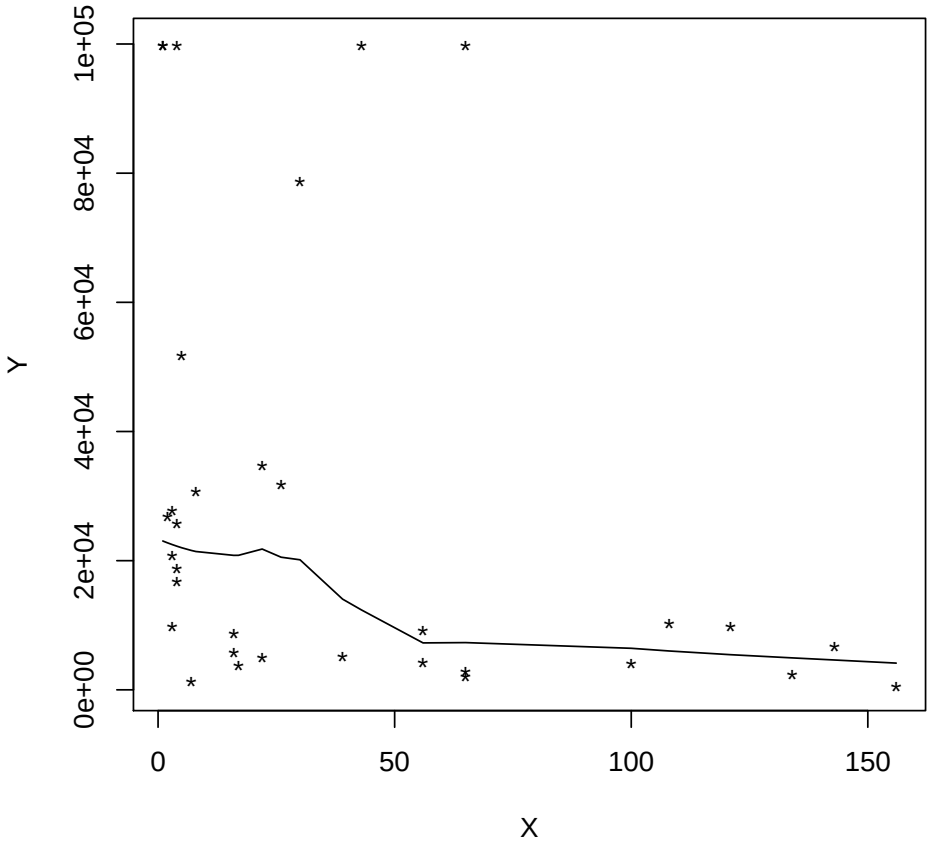



**12**.

```
> lplot(leuk[,3],leuk[,1])
```

```
$Strength.Assoc
[1] 0.5452813

$Explanatory.power
[1] 0.2973316
```

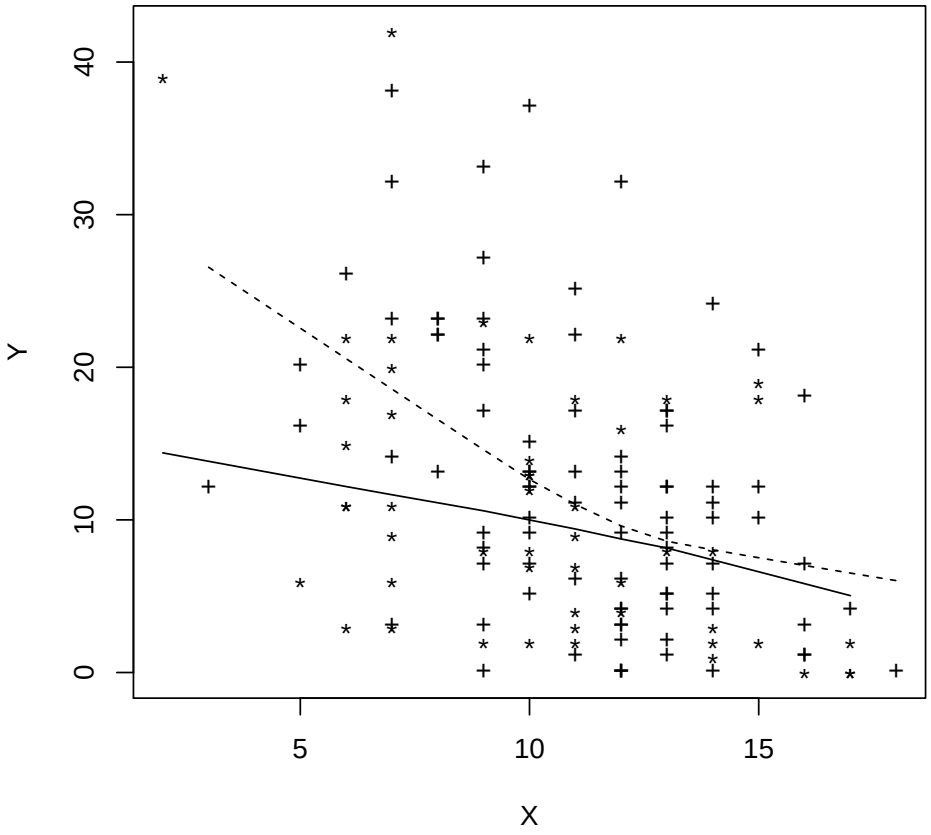
```
$yhat.values
NULL

$n
[1] 33

$n.keep
[1] 33

> wincor(leuk[,3],leuk[,1],tr=0)

$cor
[1] -0.3294525

$cov
[1] -530668.4

$p.value
[1] 0.06117379

$n
[1] 33
```




**13**.

```
> B3=read.table('B3_dat.txt',header=T)
> flag=B3$BK_SEX==1
> ancova(B3$PEOP[flag],B3$CESD[flag],B3$PEOP[!flag],B3$CESD[!flag])

[1] "NOTE: Confidence intervals are adjusted to control the probability"
[1] "of at least one Type I error."
[1] "But p-values are not"
$output
      X n1 n2        DIF      TEST       se     ci.low      ci.hi     p.value
[1,]  6 27 13 -8.3529412 3.1002120 2.694313 -15.899294 -0.8065884 0.005345770
[2,]  7 33 24 -7.8928571 3.0408807 2.595583 -15.009489 -0.7762254 0.005083267
[3,] 10 42 48 -1.5641026 0.7236809 2.161315  -7.310921  4.1827158 0.472387064
[4,] 12 32 54 -0.2705882 0.1380573 1.959970  -5.637438  5.0962618 0.891164842
[5,] 16 17 18 -0.8030303 0.2548275 3.151271  -9.881079  8.2750179 0.802020200
     crit.val
[1,] 2.800845
[2,] 2.741824
```

```
[3,] 2.658945
[4,] 2.738230
[5,] 2.880758
```




CHAPTER 15

**1**.

There are 35 participants in the high and yes category among the 200 participants.

```
> binomci(35,200)

$phat
[1] 0.175

$ci
[1] 0.1261386 0.2348873

$n
[1] 200
```

**2**.

```
> mat=matrix(c(35,80,42,43),nrow=2,ncol=2)
> contab(mat)

$delta
[1] -0.19

$CI
[1] -0.29499079 -0.08500921

$p.value
[1] 0.0003897739

> mcnemar.test(mat)

        McNemar's Chi-squared test with continuity correction

data:  mat
McNemar's chi-squared = 11.221, df = 1, p-value = 0.0008086
```

So both methods have p-values less than 0.001.

**3**.

```
> mat=matrix(c(35,80,42,43),nrow=2,ncol=2)
> chi.test.ind(mat)

$test.stat
[1] 7.433703

$p.value
[1] 0.006401347
```

**4**.

The odds ratio is

```
> (35*43)/(80*42)

[1] 0.4479167
```

The odds for the first row is

```
> (35/200)/(42/200)

[1] 0.8333333
```

the ratio of the estimated probabilities.

For the second row, the estimate is

```
> (80/200)/(43/200)

[1] 1.860465
```

The odds ratio indicates that the high group, the odds of YES is 45% of the odds for the low group.

**5**.

There are 140 participants on the diagonal. So

```
> binomci(140,320)

$phat
[1] 0.4375

$ci
[1] 0.3847509 0.4937832

$n
[1] 320
```

**6**.

```
> mat=matrix(c(30,50,10,50,70,20,20,30,40),nrow=3,ncol=3)
> Ckappa(mat)

$kappa
[1] 0.128593

$weighted.kappa
[1] 0.1538983
```

So the proportion of agreement, adjusted for chance agreement, is estimated to be 0.13.

**7**.

```
> x=c(40,50,10)
> chisq.test(x,p=c(0.5,0.3,0.2))

        Chi-squared test for given probabilities

data:  x
X-squared = 20.333, df = 2, p-value = 3.843e-05
```

**8**.

Three tests are performed, so compute each confidence at the 0.05/3 level.

```
> binomci(40,100,alpha=.05/3)
```

```
$phat
[1] 0.4

$ci
[1] 0.2895451 0.5244989

$n
[1] 100

> binomci(50,100,alpha=.05/3)

$phat
[1] 0.5

$ci
[1] 0.3850497 0.6225419

$n
[1] 100

> binomci(10,100,alpha=.05/3)

$phat
[1] 0.1

$ci
[1] 0.04164681 0.19426851

$n
[1] 100
```

The second and third confidence intervals do not contain the hypothesized value. So reject and by Tukey's three decision rule, decide that the second probability is greater than 0.3 and that the third probability is less than 0.2.

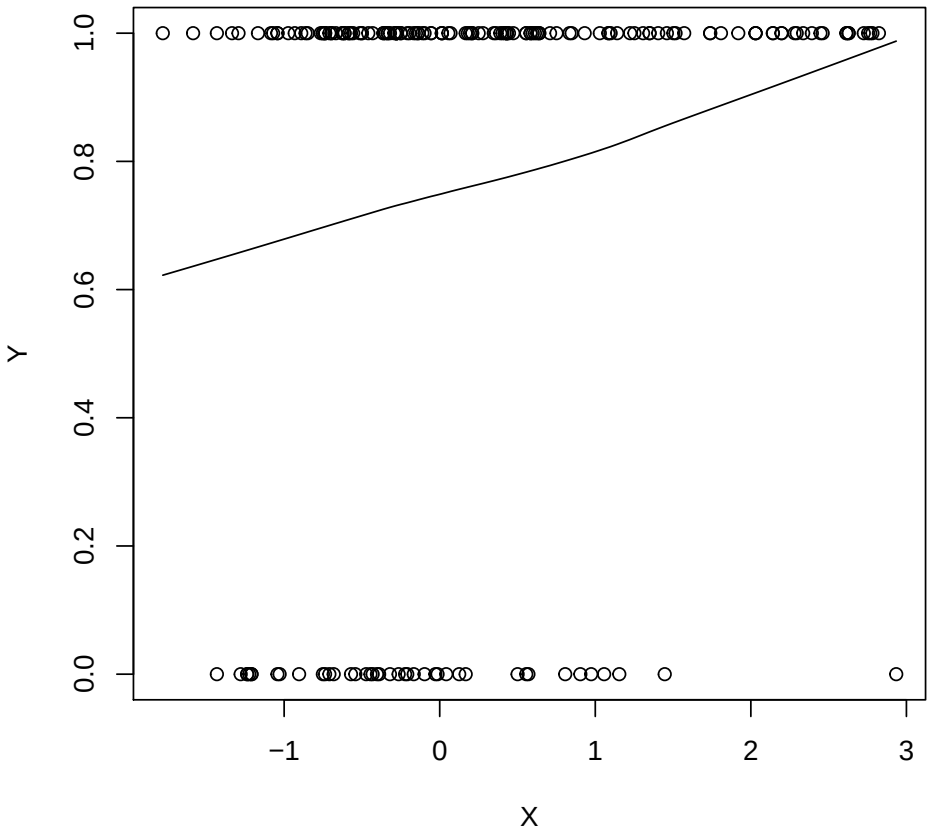
**9**.

Odds for Type A personality are 8/67. For Type B they are 5/20, so the odds ratio is (8/67)/(5/20)=0.4776119,

**10**.

```
> plasma=read.table("plasma_dat.txt",skip=15)
> flag=plasma[,3]==2
> y=plasma[!flag,3]==1 # The R function logreg takes 0-1 or T-F data only
> logrsm(plasma[!flag,13],y,xout=T)

$output
[1] "Done"
```

```
> logreg(plasma[!flag,13],y,xout=T,plotit=T)

               Estimate  Std. Error   z value    Pr(>|z|)
(Intercept) 0.135119915 0.374504128 0.3607969 0.718251307
x           0.007958051 0.002690124 2.9582470 0.003093941
```




So the plot suggests a monotonic increasing association. That is, a standard logistic regression model assumes monotonic associations, so using a standard logistic regression model seems reasonable based on this requirement. Testing the hypothesis of no association, $H_0$: $\beta_1 = 0$, the a p-value is equal to 0.003