ECON 762 Assignment 4

Question 1

a. Robustness

Robustness is the ability to resist the injection of bad data values. An estimator should respond to these small number of bad data values with smoothness. That is to say, it should only respond gradually to the injection of bad data values or small changes in the model. A robust estimator should have broad application for it to be consider useful.

b. Masking

When outliers exist, using a non-robust method may result in the outliers going undetected due to the interaction of the method with the outliers.

c. Sensitivity Curve and Influence Curve

Sensitivity curve displays whether an estimator is bounded when an outlier is inputted into the sample data. It displays the bias of the statistic when "bad data" is injected into the sample. The outlier can be an arbitrary number within a range of $\pm\infty$ and will help us determine the robustness of an estimator. An unbounded sensitivity curve is undesired as it reveals a non-robust estimator.

The influence function of an estimator is the asymptotic verson of the sensitivity curve. It is the approximation of an estimator as $n \to \infty$ when a small fraction of the sample , ϵ, are identical outliers. A contamination neighbourhood is a convex combination of the distribution used to produce the the asymptotic value of the estimator and a point mass at x_0. The influence function is the limit of the fraction of the convex distribution that is based on this point mass, as we approach the fraction from the right (ie $\epsilon > 0$).

$$IF_{\hat{\theta}}(x_0, F) = \lim_{\epsilon \downarrow 0} \frac{\hat{\theta}_\infty\left((1-\epsilon)F + \epsilon\delta_{x_0}\right) - \hat{\theta}_\infty(F)}{\epsilon}$$

It may be conisdered the limit version of the sensitivity curve.

d. Finite-Sample Breakdown Point

The point at which including one additional 'corrupted' data point into a sample will result in the bias between the corrupted and uncorrupted samples go to infinity.

X is our sample and X' is our corrupted sample when replacing m data points with arbitratary values (including $\pm\infty$ to ecompass extreme values). If our estimator $\hat{\theta} = f(X)$ then our finite-sampling breakdown point is:

$$\min\left\{\frac{m}{n}; \text{bias}(m; f(X'), X) \text{ is infinite}\right\}$$

When m=1, our breakdown point would be zero. Injecting one data point with an arbitrary value would have a large effect on $\hat{\theta}$ and cause the estimator to `break down`. The highest possible breakdown point is is one less than half the sample size or a breakdown point of 50%.

e. Huber (1964) M-Estimator

Where estimating the median through the minimizing the absolute values provides for a robust (bounded) estimator it is not efficient due to discontunity problems at $\mu = 0$.The variance of the median is much higher

than that of the mean and especially if no outliers present. As a result, the M-Estimator was proposed to combat this problem. It has the functional form:

$$\rho_c(\mu) = \begin{cases} \mu^2 & \text{if } |\mu| \leq c \\ 2c|\mu| - c^2 & \text{if } |\mu| > c \end{cases} .$$

Where c will be the resistence parameter that is specified by the user. As $c \to 0$, the efficiency of the estimator decreases. The rule-of-thumb is $c = 1.345 \times s$ where s a robust measure of scale. This would allow the efficiency is almost as good as mean (905%) while also being robust.

Question 2

a) The finite sample breakdown point for the inter-quartile range would be $m = n/4 - 1$ and will have a breakdown point of 25% (expressed in the limit $n \to \infty$). Due to the ranked order nature of calculating the interquartile range, values that are $\pm\infty$ will fall outside of the interquartile range. As the inter-quartile range is the 75% quartile minues the 25% quartile, injecting a quarter of the sample with bad data will breach at least one point in the inter-quartile range, causing the bias to go to infinity. The interquartile range will have a breakdown point of 25%. Intuitively, it make senses because if we inject m values that are less than 25% of the sample, they might all end up in the 25% quartile or above the 75% quartile, avoiding any bad data in the interquartile. A breakdown point of 25% would mean at least one bad data point would end up in the interquartile range.

b) Where ranked order of the data points is used to calculate the interquartile range, MAD_n is calculated by:

$$MAD_n = \frac{Med|x_i - Med(x_i)|}{0.675}$$

We center all of the data points in a sample and center them about the medium of the sample. By finding the medium and dividing by 0.675, if our data sample is drawn from a Guassian distribution, MAD_n will equal the standard deviation of the distribution. Where the interquartile range is based on the rank order of the sample, MAD_n is based around the deviation around the median. This allows for the scale to be robust against injections of bad data and has a breakdown point of 50%.

c) Rousseeuq and Croux's Q_n is based on distance similar to interquartile range. It is a more efficient robust scale estimator than MAD_n, which suffers a drawback of discountinity with it's influence function. If we look at the location framework for the sample median we see that it will suffer from this same discountinuity:

$$median|x_i - x_j|; i < j$$

This will have a desired breakdownpoint of 50% but still suffers from inefficiency.Q_n is a more efficient estimator of scale than the location framework produces while still have a 50% breakdown point:

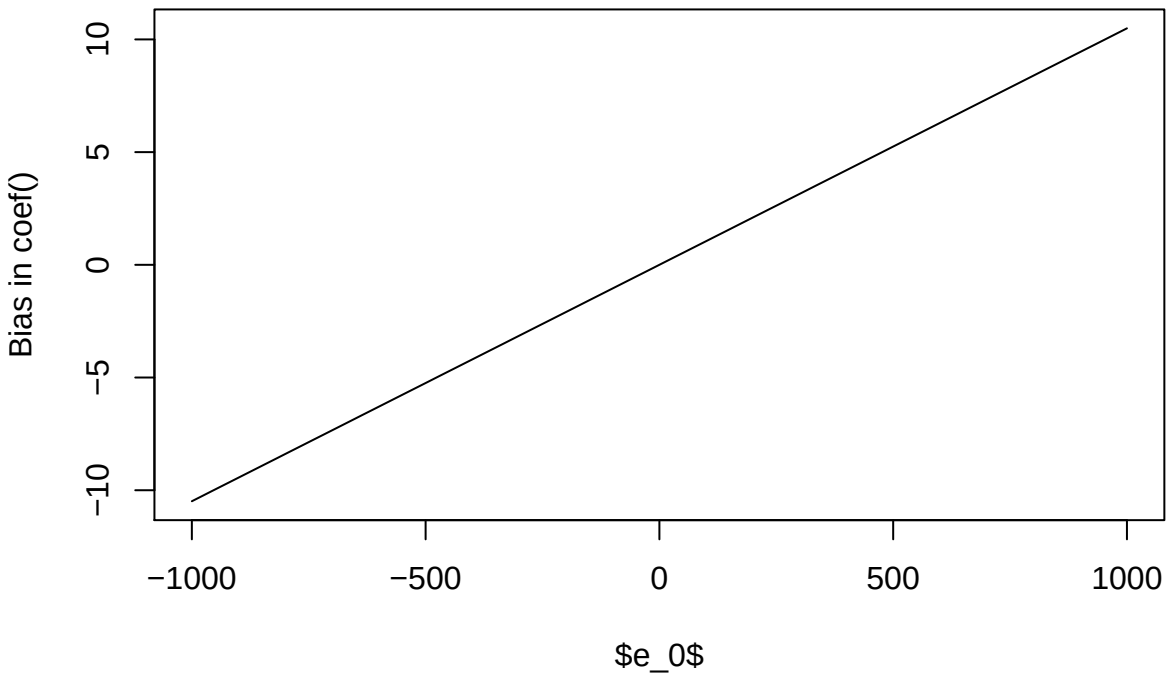
$$Q_n = d|x_i - x_j|; i < j_{(k)}$$

Where d is a constant factor and k denotes the kth order statistic.

Question 3

a)

```
set.seed(42)
n <- 100
x <- runif(n)
beta1 <- 1
beta2 <- 2
epsilon <- rnorm(n,sd=.1)
y <- beta1 + beta2*x + epsilon
model <- lm(y~x)
coef.sample<-coef(model)[2]

n.e0<-100
e0 <- seq(-1000,1000,length=n.e0)
sc.coef <- numeric(n.e0)
for(i in 1:n.e0) {
  e.augmented <- c(epsilon[1:99],e0[i])
  y <- beta1 + beta2*x + e.augmented
  sc.coef[i] <- coef(lm(y~x))[2]-coef.sample
}

## Sensitivity curve for the sample coef
plot(e0, sc.coef,
     ylab="Bias in coef()",
     xlab="$e_0$",
     type="l")
```



b) The key feature of the sensitivity curve is to help determine the robustness of an estimation method. We wish to have boundness in the sensitivity curve to determine if the estimator is robust and can handle injection of arbtirary data values. The OLS estimator has a sensitivity curve that is linear when injecting $x0$ over the pre-determined interval. This shows that our bias will go to infinity if we inject a data value of infinity. The OLS estimator is a non-robust estimator.

We should care because outliers can effect our estimator by large magnitudes and producing a method where there is little confidence in the estimator.

Question 4

a)

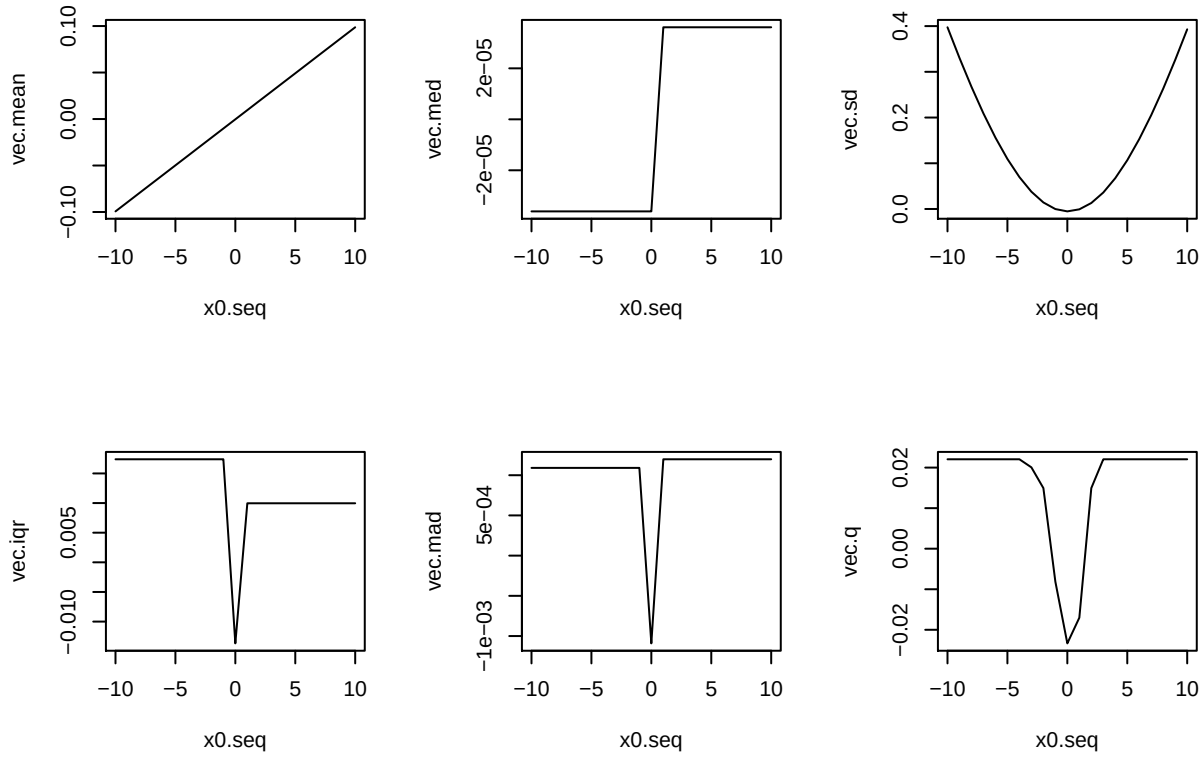
```
set.seed(42)
require(robustbase)
```

```
## Loading required package: robustbase
```

```
n <- 100
x<-rnorm(n)
mean.x<-mean(x)
med.x<-median(x)
sd.x<-sd(x)
iqr.x<-IQR(x)
mad.x<-mad(x)
q.x<-Qn(x)
```

b)

```r
set.seed(42)
require(robustbase)
n <- 100
x<-rnorm(n)
x0.seq<-seq(from=-10,to=10,by=1)

mean.x<-mean(x)
med.x<-median(x)
sd.x<-sd(x)
iqr.x<-IQR(x)
mad.x<-mad(x)
q.x<-Qn(x)

vec.mean<-numeric()
vec.med<-numeric()
vec.sd<-numeric()
vec.iqr<-numeric()
vec.mad<-numeric()
vec.q<-numeric()

for (i in 1:length(x0.seq)){
  x[101]<-x0.seq[i]
  vec.mean[i]<-mean(x)-mean.x
  vec.med[i]<-median(x)-med.x
  vec.sd[i]<-sd(x)-sd.x
  vec.iqr[i]<-IQR(x)-iqr.x
  vec.mad[i]<-mad(x)-mad.x
  vec.q[i]<-Qn(x)-q.x
}

par(mfrow=c(2,3))
plot(x0.seq,vec.mean,type="l")
plot(x0.seq,vec.med,type="l")
plot(x0.seq,vec.sd,type="l")
plot(x0.seq,vec.iqr,type="l")
plot(x0.seq,vec.mad,type="l")
plot(x0.seq,vec.q,type="l")
```



The median, interquartile range, MAD_n and Q_n appear to the be bounded.

c)

```
set.seed(42)
require(robustbase)
n <- 20
x<-rnorm(n)
mean.x<-mean(x)
med.x<-median(x)
sd.x<-sd(x)
iqr.x<-IQR(x)
mad.x<-mad(x)
q.x<-Qn(x)

x0<-1000

vec.mean<-numeric()
vec.med<-numeric()
vec.sd<- numeric()
vec.iqr<-numeric()
vec.mad<-numeric()
vec.q<-numeric()

for (i in 1:10){
  x[1:i]<-x0
  vec.mean[i]<-mean(x)
```

```
  vec.med[i]<-median(x)
  vec.sd[i]<-sd(x)
  vec.iqr[i]<-IQR(x)
  vec.mad[i]<-mad(x)
  vec.q[i]<-Qn(x)
}

values<-data.frame(seq(1,10,by=1),vec.mean,vec.med,vec.sd,vec.iqr,vec.mad,vec.q)

colnames(values)<-c("M","Mean","Median","Standard Deviation","IQR","MAD","Q")

knitr::kable(values)
```

M	Mean	Median	Standard Deviation	IQR	MAD	Q
1	50.12337	0.1502072	223.5814	1.588835	0.8900408	1.363505
2	100.15161	0.3836984	307.7443	1.537654	1.1780166	1.446579
3	150.13345	0.5185655	366.2922	1.807936	1.1893167	1.711573
4	200.10181	0.5201094	410.3411	2.255167	1.3312079	2.343159
5	250.08159	0.9704100	444.2151	251.884672	1.8561127	2.580050
6	300.08690	1.3124915	470.1057	1000.169688	2.3632827	2.611880
7	350.01132	1.3124915	489.3535	1000.169688	3.1861790	2.580711
8	400.01606	1.6692685	502.6127	1000.169688	5.3135381	2.580050
9	449.91514	1.8033794	510.4978	1000.169688	6.4520388	2.071679
10	499.91827	501.1433227	513.0743	1000.169688	739.6049098	1.717336

From looking at the table, the non-robust method values are much more sensitive to influence by the mass points. We can also see from the table, the breakdown points for the estimators with Q_n being the only one to remain robust when influence exerted on half the sample.

d)

```
set.seed(42)
require(robustbase)
n <- 20
x<-rnorm(n)

x0<-1000

for (i in 1:5){
  x[1:i]<-x0
  t.t<-(x-mean(x))/sd(x)
  print(subset(x,abs(t.t)>3.0))
  t.r<-(x-median(x))/mad(x)
  print(subset(x,abs(t.r)>3.0))
}
```

```
## [1] 1000
## [1] 1000.000000   -2.656455
## numeric(0)
## [1] 1000 1000
## numeric(0)
## [1] 1000 1000 1000
## numeric(0)
```

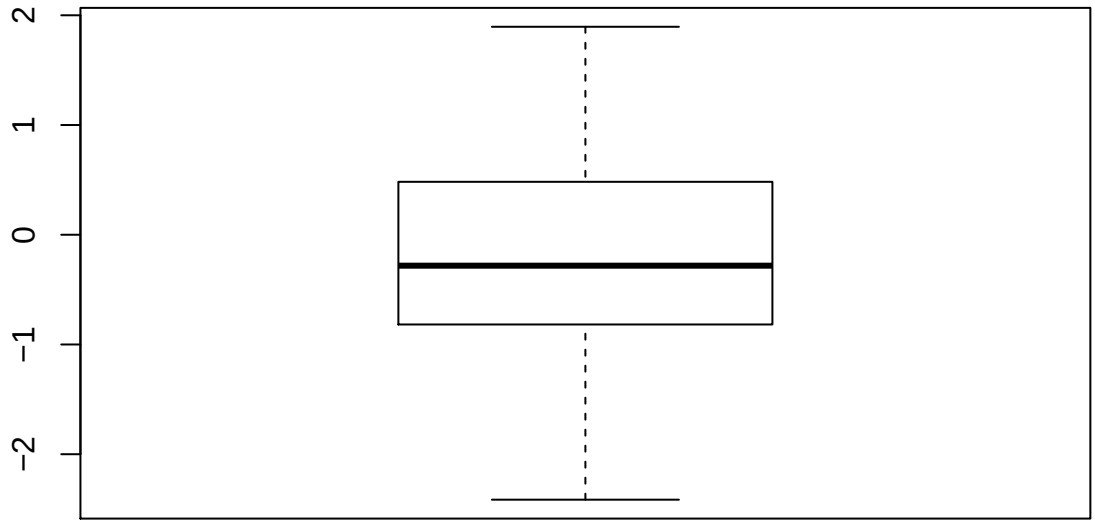
```
## [1] 1000 1000 1000 1000
## numeric(0)
## [1] 1000 1000 1000 1000 1000
```

We can see that the non-robust three-sigma edit rule results in the value -2.656 being masked when compared to the robust method. Once we start increasing the number of point mass values in the sample, all of the point masses are masked when using the non-robust method for $m \geq 2$. We can see that the non-robust method will resulting in masking as we inject bad data values into the sample.

Question 5

a) A box and whisker plot is a method that defines what points would be considered outliers and displays the median, interquartile range, the boundary for outliers and the outliers themselves. It defines

```
n <- 20
x<-rnorm(n)

boxplot(x)
```

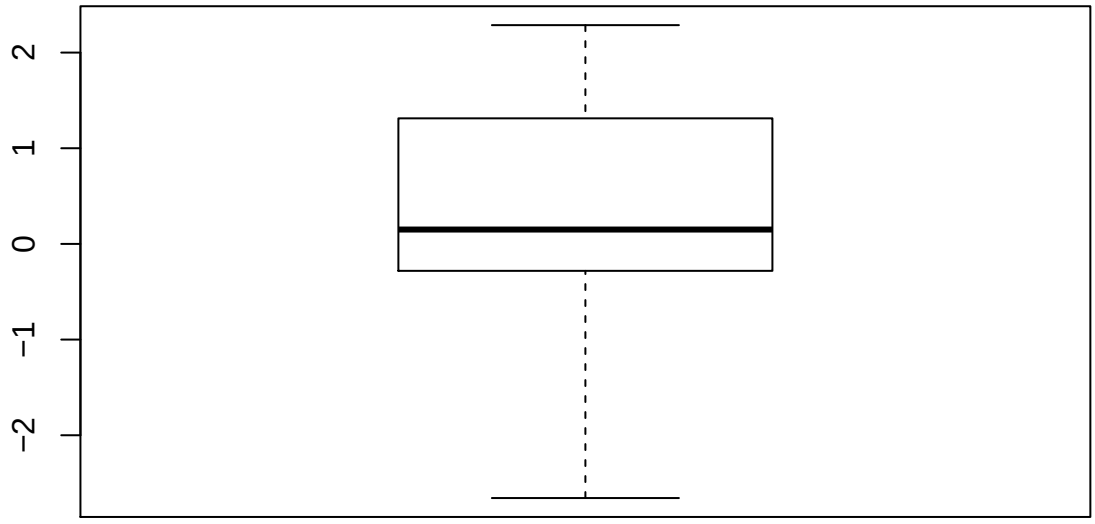


The thick line is the median of the distribution while the top and bottom of the "box" are quartiles Q_1 and Q_1. The "whiskers" are the farthest points that are not outliers. Here they are defined as values within $(3/2) \times Q$. The points are data values that lie outside of the whiskers and values that are the same would be placed next to each other horizontally.

b) The advantage for using a box and whisker plot is that it is a robust measurement of scale as it relies on the median values of the sample. Large data values that are

c)

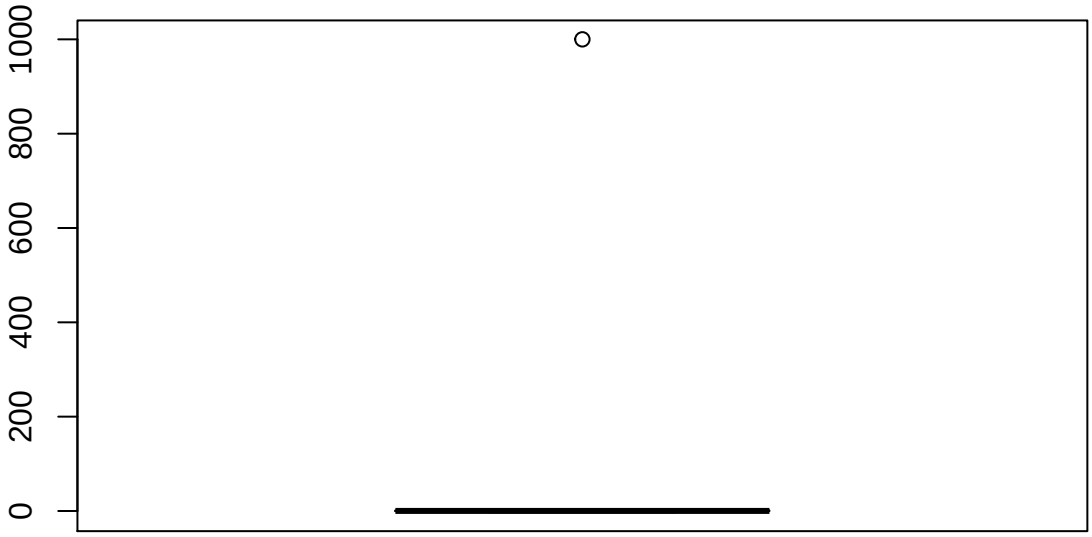
```
set.seed(42)
require(robustbase)
n <- 20
x<-rnorm(n)

x0<-1000
bp<-boxplot(x)
```



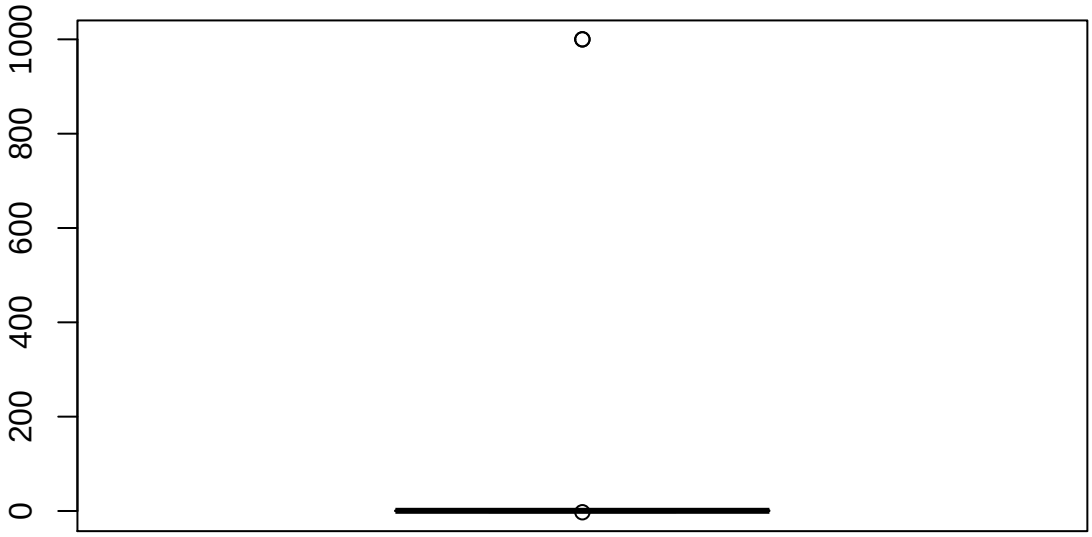

```
print(bp$out)
```

```
## numeric(0)
```

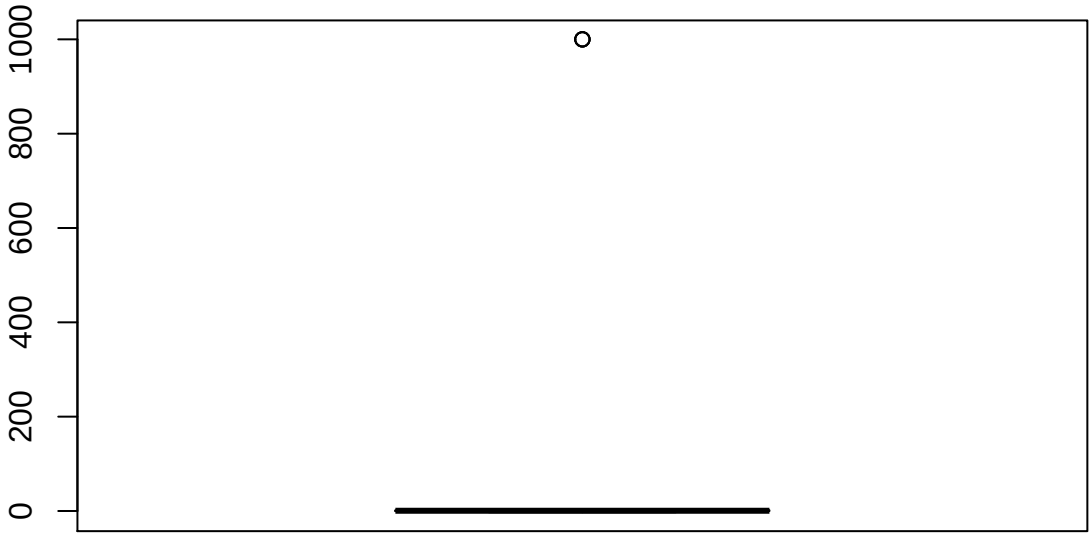
```
for (i in 1:5){
  x[1:i]<-x0
  bp<-boxplot(x)
  print(bp$out)
}
```



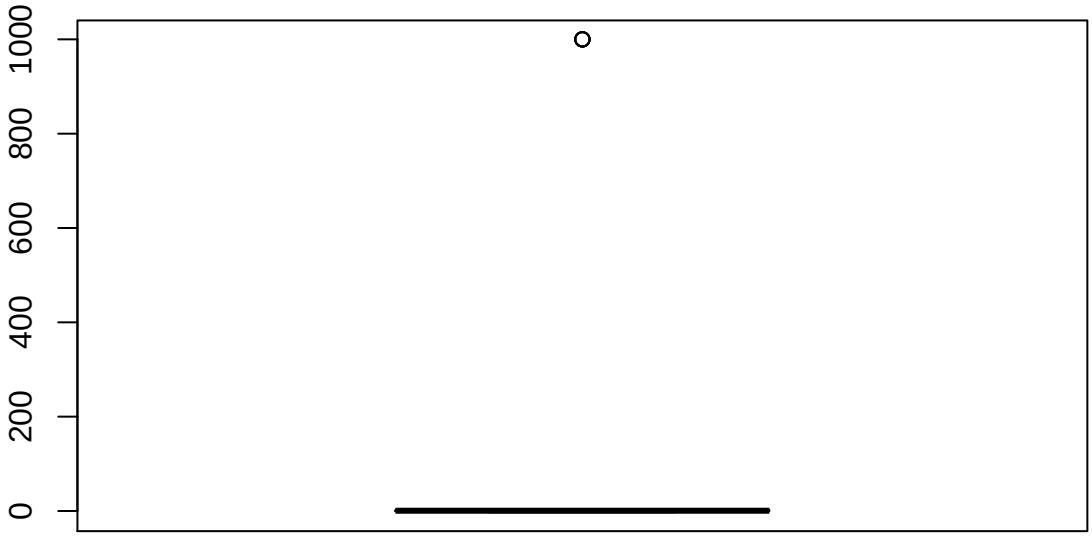

```
## [1] 1000
```



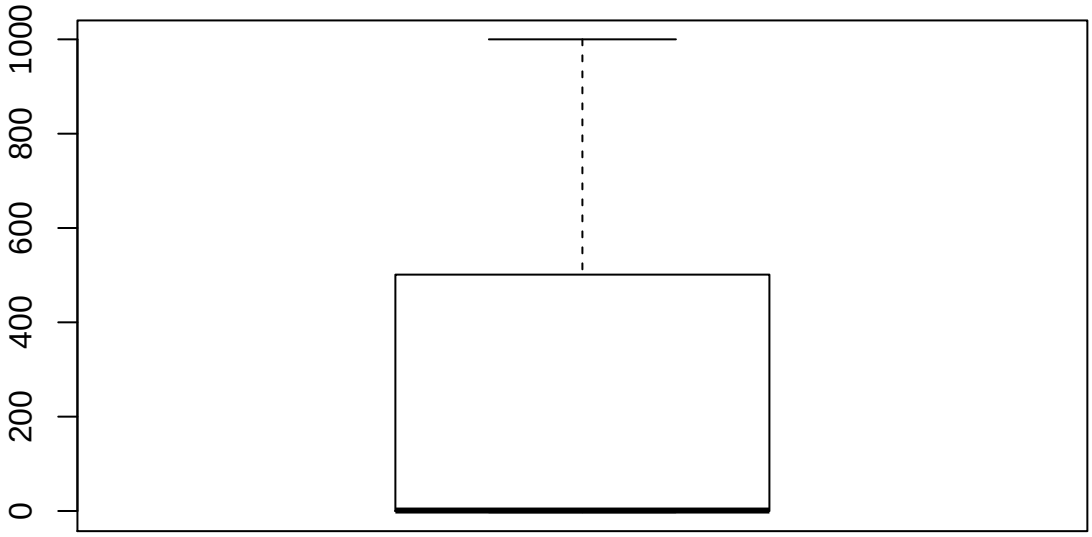

```
## [1] 1000.000000 1000.000000   -2.656455
```




```
## [1] 1000 1000 1000
```




```
## [1] 1000 1000 1000 1000
```




```
## numeric(0)
```

Question 6

The hat matrix H gives the fitted value of Y: $\hat{Y} = HY$. Our X is a $n \times k$ matrix.

$$\begin{aligned} H &= X(X'X)^{-1}X' \\ HY &= X(X'X)^{-1}X'Y \\ HY &= X\hat{\beta} \\ HY &= \hat{Y} \end{aligned}$$

The trace of the H matrix requires knowledge of the properties for finding the trace of a matrix:

$$tr(ABC) = tr(BAC) = tr(CBA)$$

The trace of a matrix is the sum of the diagonals.

$$\begin{aligned} tr(X(X'X)^{-1}X') &= tr((X'X)^{-1}X'X) \\ tr(I_k) &= k \end{aligned}$$

An element of an H matrix is defined as h_{ij}. Since the trace of a matrix is the sum of diagonals, we can show that $\bar{h} = k/n$.

$$tr(H) = \sum_{i=1}^{k} h_{ii} = k$$
$$\frac{1}{n}\sum_{i=1}^{n} h_{ii} = \frac{1}{n}\bar{h} = \frac{k}{n}$$

A hat matrix is symmetric and idempotent. We know that the property of an idempotent is that the a diagonal matrix is the squared sum of columns of the row the element is on. As a result of this property, this value will be ≥ 0:

$$h_{ii} = \sum_{j=1}^{n} h_{ij}^2 \geq 0$$

By decomposing the summation into diagonal and non-diagonal terms and dividing by h_{ii}, we can show that each diagonal matrix is ≤ 0:

$$\frac{h_{ii}}{h_{ii}} = \frac{1}{h_{ii}}(h_{ii}^2 + \sum_{j=1, j\neq i}^{n} h_{ij}^2)$$
$$1 = h_{ii} + \frac{\sum_{j=1, j\neq i}^{n} h_{ij}^2}{h_{ii}}$$
$$h_{ii} = 1 - \frac{\sum_{j=1, j\neq i}^{n} h_{ij}^2}{h_{ii}}$$
$$h_{ii} \leq 1$$

Therefore, $0 \leq h_{ii} \leq 1$.

b) We know that $X'X = \sum_{i=1}^{n} x_i x_i'$ and that $X_{(-t)}X_{-t}'$ subtracts the t^{th} observation from $X'X$ matrix. Therefore:

$$X_{(-t)}'X_{(-t)} = X'X - x_t x_t'$$

The Sherman-Morrison-Woodbury formula states $(A - UV')^{-1} = A^{-1} + A^{-1}U\left(I_m - V'A^{-1}U\right)^{-1}V'A^{-1}$ where A is a $p \times p$ matrix and U and V are $p \times m$ matricies. From our equation above, we can state that $A = X'X$ and $U, V = x_t$ where m=1. We can use the Sherman-Morrison-Woodbury formula to solve for inverse of our equation:

$$(X_{(-t)}X'_(-t))^{-1} = (X'X - x_t x'_t)^{-1}$$
$$= (X'X)^{-1} + (X'X)^{-1}x_t(1 - x'_t(X'X)^{-1}x_t)^{-1}x'_t(X'X)^{-1}$$

We know that $h_{ii} = x'_i(X'X)^{-1}x_i$.

$$(X_{(-t)}X'_{(-t)})^{-1} = (X'X)^{-1} + \frac{(X'X)^{-1}x_t x'_t(X'X)^{-1}}{(1 - h_{tt})}$$

c) Similarily, $X'Y = \sum_{i=1}^{n} x'_i y_i$. We can find $X'_{(-t)}Y_{(-t)}$ by subtracting the t^{th} observation $x_t y'_t$:

$$X'_{(-t)}Y_{(-t)} = X'Y - x'_t y_t$$

d) Using the formula for OLS, we can estimate coefficients for data missing an observation

$$\hat{\beta}_{(-t)} = (X'_{(-t)}X_(-t))^{-1}X'_{(-t)}Y_{(-t)}$$

We can replace the elements with the values we previously determined.

$$\hat{\beta}_{(-t)} = \left((X'X)^{-1} + \frac{(X'X)^{-1}x_t x'_t(X'X)^{-1}}{(1 - h_{tt})}\right)(X'Y - x'_t y_t)$$
$$= (X'X)^{-1}X'Y + \frac{(X'X)^{-1}x_t x'_t(X'X)^{-1}}{(1 - h_{tt})}X'Y - (X'X)^{-1}x_t y_t - \frac{(X'X)^{-1}x_t x'_t(X'X)^{-1}}{(1 - h_{tt})}x_t y_t$$

We know that $h_{ii} = x'_i(X'X)^{-1}x_i$ so we can simplify our expression.

$$= \hat{\beta} + \frac{(X'X)^{-1}x_t x'_t\hat{\beta}}{(1 - h_{tt})} - \frac{(1 - h_{tt})(X'X)^{-1}x_t y_t}{(1 - h_{tt})} - \frac{(X'X)^{-1}x_t h_{tt} y_t}{(1 - h_{tt})}$$

The fitted values of an observation can be placed into equation ($\hat{y} = x'_t\hat{\beta}$) and we factor our common terms:

$$
\begin{aligned}
&= \hat{\beta} + \frac{(X'X)^{-1}x_t}{(1-h_{tt})}(\hat{y}_t - (1-h_{tt})y_t - h_{tt}y_t) \\
&= \hat{\beta} + \frac{(X'X)^{-1}x_t}{(1-h_{tt})}(\hat{y}_t - y_t) \\
\hat{\beta}_{(-t)} &= \hat{\beta} + \frac{(X'X)^{-1}x_t\hat{\epsilon}_t}{(1-h_{tt})} \\
\hat{\beta}_{(-t)} - \hat{\beta} &= \frac{(X'X)^{-1}x_t\hat{\epsilon}_t}{(1-h_{tt})}
\end{aligned}
$$