

Lecture 3

1. EDA (EXPLORATORY DATA ANALYSIS by J.Tukey).

Learn how to explore data and find valuable information, structures and relationships among variables. Find the structure of the majority of the data but also detect exceptional observations, rare events. Read th article “The future of Data Analysis” J W Tukey (1952), Annals of Probability & Statistics.

2. STEM AND LEAF DISPLAYS (Tukey).

A number has about 3 significant digits. Split numbers into STEM + LEAF

22.5 22 . 5
0.00000917 0.0000091 7

If we find a	they can be
set of numbers	decomposed
such as	common + stem + leaf + noise
0.000007884482	0.0000078 84 4 82
0.000007888237	0.0000078 88 2 37
0.000007884282	0.0000078 84 2 82
0.000007883434	0.0000078 83 4 34
0.000007884150	0.0000078 84 1 50

- **ROUNDING:**

- ❖ Rounding is important: 84 5, 88 2, 84 3 , 83 4
- ❖ Round 5's to the even number. 84 1 50 rounds to 84 2

- **Constructing a STEM AND LEAF diagram.**

Put the stems in one column and the leaves are added in rows:

Stems: 83 to 88

83 4

84 235

85

86

87

88 2

GROUP LINES BY TWO DIGITS:

0-1 2-3 4-5 6-7 8-9

82 4

84 235

86

88 2

But this is not well specified so we have to round one more digit

8 3

8 444

8

8 8

Or we group lines by 5 digits: 0-4 5-9

8 3444

8 8

- **R example :** In R we use the *stem* function:

```
> data(faithful)
> attach(faithful)
> stem(waiting, scale=1)
The decimal point is 1 digit(s) to the right of the |
 4 | 3
 4 | 55566666777788899999
 5 | 00000111111222223333334444444444
 5 | 55555666677788889999999
 6 | 00000022223334444
 6 | 555667899
 7 | 0000111112333333444444
 7 | 55555556666666677777777777888888888888888999999999
 8 | 00000000111111111111222222222223333333333334444444444
 8 | 5555566666677888888999
 9 | 00000012334
 9 | 6
```

- **SAS example**

In SAS this is part of PROC UNIVARIATE.

THIS IS AN EXAMPLE OF WHAT SAS DOES.

```
OPTIONS PS=55 LS=80;
DATA CRIME;
INFILE 'crime.dat';
INPUT MURDER RAPE ROBBERY ASSAULT BURGLARY LARCENY
      AUTOTHFT REGION $;

run;
proc univariate plot;
var murder;
run;
```

Stem Leaf	#	Boxplot
15 3	1	
14 6	1	
13		
12 27	2	
11 156778	6	
10 17	2	
9 24456	5	+-----+
8 488	3	
7 79	2	
6 2269	4	*--+-*
5 3779	4	
4 6688	4	
3 022455568	9	+-----+
2 000	3	
1 359	3	
0 5	1	

- **Depth, Ranks, Order of data values**

DEPTH OF A VALUE = MINIMUM OF ASCENDING AND DESCENDING RANKS.

WE MAY ADD DEPTH TO THE STEM AND LEAF DIAGRAM

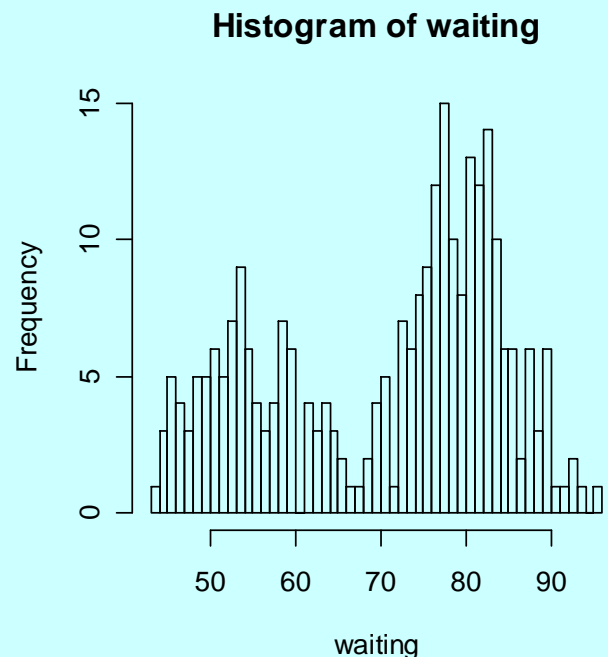
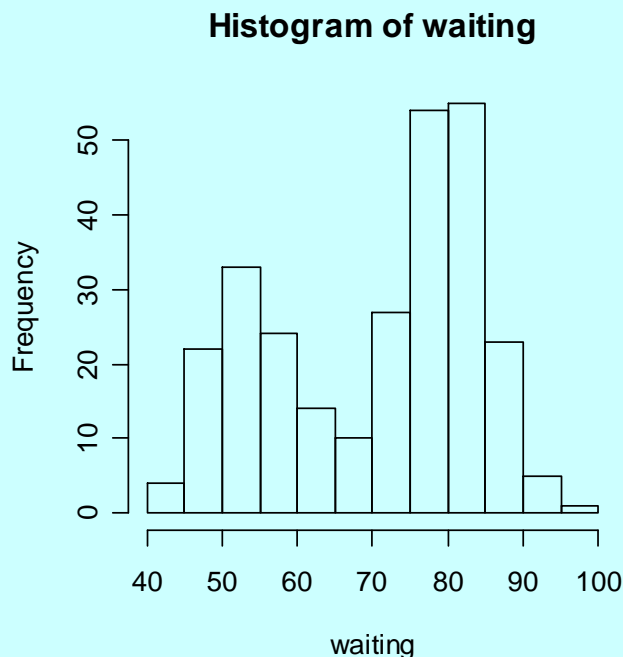
```
> depth = function(x) pmin( rank(x), length(x) +1-rank(x) )
```

```
> cbind(waiting,depth=depth(waiting), rank=rank(waiting))
```

```
      waiting depth rank
[1,]      79  97.5 175.5
[2,]      54  49.0  49.0
[3,]      74 123.5 123.5 ...
```

3. Histograms

- > `par(mfrow=c(1,2))`
- > `hist(waiting)`
- > `hist(waiting,50)`



For large samples letter values are hard to read but histograms are suitable.

- **CHOOSING THE NUMBER OF LINES OF A STEM & LEAF OR A HISTOGRAM**

$$L = [10 * \log_{10}(n)]$$

where $[x]$ is the integer part of x .

EXAMPLE: If $n = 25$ $L = [10 * \log_{10}(25)] = [10 * 1.39794] = 13$

- **CLASS WIDTH: Round(RANGE / L)**

The range for ROBBERY IS 435.7 so $435.7 / 13 = 33.5$

We choose width=50.

The range for MURDER IS 13.3 so $13.3 / 13 = 1.02$

We choose width= 1 or 2.

- **OTHER RULES**

Sturges: $L = [1 + \log_2(n)]$

RootN: $L = [2 \sqrt[n]{n}]$

Table 1. comparing Log_{10} , RootN and Sturges rules.

N	LOG ₁₀	ROOTN	STURGES
10	10.00	6.32	4.32
20	13.01	8.94	5.32
30	14.77	10.95	5.90
40	16.02	12.64	6.32
50	16.98	14.14	6.64
60	17.78	15.49	6.90
70	18.45	16.73	7.12
80	19.03	17.88	7.32
90	19.54	18.97	7.49
100	20.00	20.00	7.64
200	23.01	28.28	8.64
300	24.77	34.64	9.22
400	26.02	40.00	9.64
500	26.98	44.72	9.96

N	LOG ₁₀	ROOTN	STURGES
16	12.04	8.00	5.00
32	15.05	11.31	6.00
64	18.06	16.00	7.00
128	21.07	22.62	8.00
256	24.08	32.00	9.00

DOANE: CORRECTION FOR SKEWNESS

$\text{LOG}_2(1 + \text{skewness} / \sigma(\text{skewness}))$

$$\text{skewness} = \mu_3 / \sigma^3$$

$$\sigma(\text{skewness}) = \sqrt{\frac{6(n-2)}{(n+1)(n+3)}}$$

- **ANOTHER APPROACH: RULES FOR INTERVAL WIDTH: h**

Scotts Rule: $h = \frac{3.49 \times s}{\sqrt[3]{n}}$

where s is a resistant estimate of σ

Freedman & Diaconis $h = 1.66 \times s \times \sqrt[3]{\text{Log}_e(n)/n}$

Freedman & Diaconis $h^* = \frac{2 \times IQR}{\sqrt[3]{n}}$

Table 2. comparing Freedman & Diaconis h Vs Scott's rule (times s).

n	F-D	SCOTT
10	1.017	1.619
20	0.881	1.285
30	0.803	1.123
40	0.749	1.020
50	0.709	0.947
60	0.678	0.891
70	0.652	0.846
80	0.630	0.809
90	0.611	0.778
100	0.595	0.751
200	0.494	0.596
300	0.443	0.521
400	0.409	0.473
500	0.384	0.439
N	F-D	SCOTT
16	0.925	1.385
32	0.791	1.099
64	0.667	0.872
128	0.557	0.692
256	0.462	0.549

4. LETTER VALUES

Start with a batch of data X_1, \dots, X_n

Order Statistics: SORT IT and it becomes $X_{(1)}, \dots, X_{(n)}$

DEFINE Ranks: Upward and Downward for each observation.

DEFINE depth: Min of Upward and Downward rank.

- **MEDIAN**

$$\text{Median} = \begin{cases} (X_{(k)} + X_{(k+1)})/2 & \text{if } n = 2k \\ X_{(k+1)} & \text{if } n = 2k + 1 \end{cases}$$

This definition has some bad properties.

- **FOURTHS**

$$\text{depth(Fourth)} = ([\text{depth(Median)}] + 1)/2$$

- **5-NUMBER SUMMARIES**

Min Fourth Median Fourth Max

- **LETTER VALUES**

$$\text{depth(Eight)} = ([\text{depth(FOURTH)}] + 1)/2$$

AND SO ON.

$$([\text{Prev depth}] + 1)/2$$

Letters as Tags : M F E D C B A Z Y X

- **MID VALUES = (UPPER LV + LOWER LV)/2**
- **SPREAD = UPPER LV - LOWER LV**

Example: Fourth Spread = (Upper Fourth - Lower Fourth)

- **NICE FORM LETTER VALUE DISPLAYS**

M	Depth of the Median	Median			
F	Depth of Fourth	Lower Fourth	Upper Fourth	Mid	Spread
1		Lower Extreme	Upper Extreme	Mid	Spread

Figure 1. Letter values for the Standard Normal Distribution

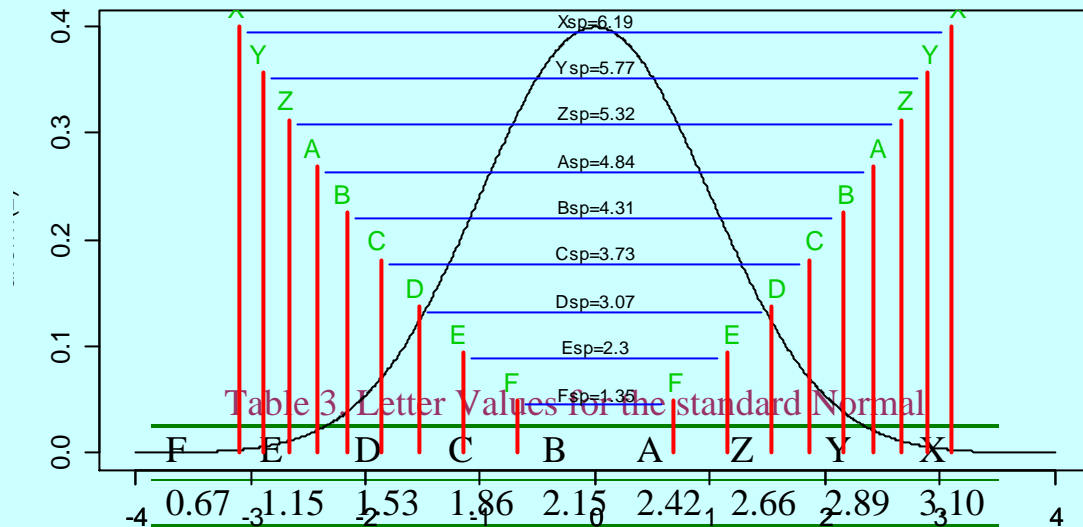
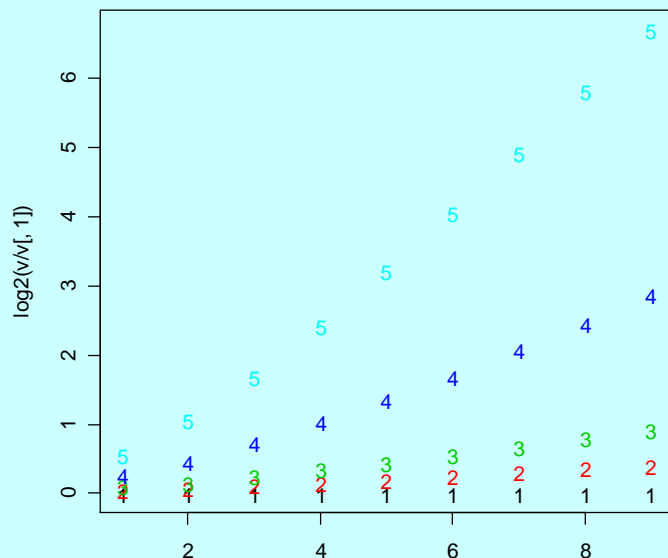


Table 4. Letter Spreads for the T-10, T5, T-2 and T-1 and Ratios by std Normal spreads

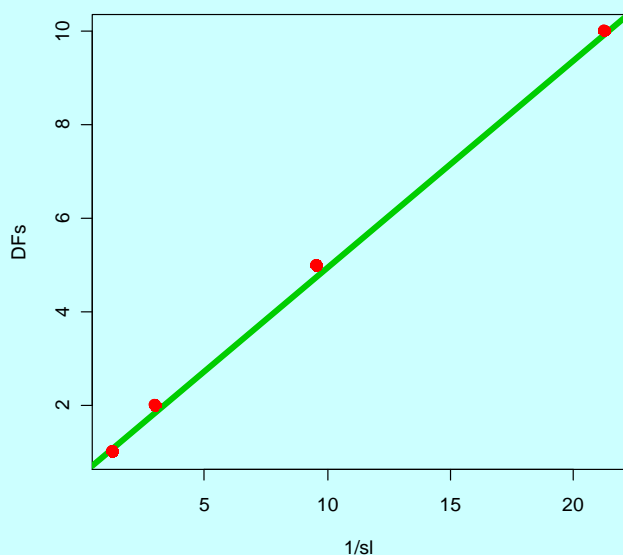
	NORMAL	T-10	RAT	T-5	RAT	T-2	RAT	T-1	RAT
F_{SPREAD}	1.35	1.40	1.04	1.45	1.08	1.63	1.21	2.00	1.48
E_{SPREAD}	2.30	2.44	1.06	2.60	1.13	3.21	1.39	4.83	2.10
D_{SPREAD}	3.07	3.35	1.09	3.68	1.20	5.11	1.67	10.05	3.28
C_{SPREAD}	3.73	4.19	1.13	4.78	1.28	7.62	2.05	20.31	5.45
B_{SPREAD}	4.31	5.01	1.16	5.93	1.38	11.05	2.56	40.71	9.45
A_{SPREAD}	4.84	5.82	1.20	7.19	1.49	15.81	3.27	81.47	16.85
Z_{SPREAD}	5.32	6.63	1.25	8.57	1.61	22.49	4.23	162.97	30.63
Y_{SPREAD}	5.77	7.46	1.29	10.12	1.75	31.91	5.53	325.95	56.48
X_{SPREAD}	6.19	8.32	1.34	11.85	1.91	45.19	7.29	651.90	105.24

Figure 2. $\log_2(\text{spread ratios of } T \text{ with } df=10,5,2,1 \text{ over Normal spreads})$ Vs $-\log_2(\text{Tail Prob})$



Estimated slopes: 0.047 0.1046 0.328 0.782

Figure 3. DF approximation: $1/2 + 1/(2.25 \cdot \text{slope})$



R-CODE

```

probs= 2^-(2:10) # tail prob for 9 letter values
u = -qnorm(probs) # Letter values for Standard Normal
x = c(t(cbind(-u,-u,NA,u,u,NA))) # Lines for the graph
yy = seq(0.05,0.4,length=9) # Lines for the graph
y = c(t(cbind(0,yy,NA,0,yy,NA))) # Lines for the graph
x1 = c(t(cbind(-u+0.07,u-0.09,NA))) # Lines for the graph
y1 = c(t(cbind(yy-0.005,yy-0.005,NA))) # Lines for the graph
# Graph commands start here
par(cex=0.8) # Set character size at 80%
    
```

```

plot(z <- ((-400):400)/100,dnorm(z),type="l",xlab="") #Plots
# the bell curve
lines(x,y,lwd=2,col=2) # draws vertical lines
L=c("F","E","D","C","B","A","Z","Y","X") # vector of letters
text(-u-.05,yy+0.02,L,col=3) # Draws letters
text(u+.03,yy+0.02,L,col=3) # Draws letters
lines(x1,y1,col=4) # Draw lines
par(cex=0.6) # lower character size
text(0,yy+0.006,paste(L,"sp=",round(2*u,2),sep=""))
#text over lines

v = qnorm(probs) # normal quantiles
v = -2*cbind(v,qt(probs,df=10),qt(probs,df=5),
qt(probs,df=2),qt(probs,df=1)) # combine it with t quantiles
# for df= 10,5,2,1

matplot(1:9,log2(v/v[,1])) # Plot all the ratios
lsfit(1:9,log2(v/v[,1])[,2])$coef[2] ->s1 # calculate
lsfit(1:9,log2(v/v[,1])[,3])$coef[2] ->s1[2] # individual
lsfit(1:9,log2(v/v[,1])[,4])$coef[2] ->s1[3] # slopes for
lsfit(1:9,log2(v/v[,1])[,5])$coef[2] ->s1[4] # all lines

lsfit(1/s1,c(10,5,2,1))$coef # line fit
plot(1/s1,c(10,5,2,1),pch=16,col=2,ylab="DFs") # graph of
abline(0.5, 1/2.25,col=3,lwd=2) # fit
round(1/2 + 1/(s1*2.25)) # final estimate of df's from line

```

LETTER VALUE DISPLAYS USING SAS: “PROC IML” CODE

We write a program in SAS PROC IML. This is more like a programming language.

```

/* This is just an example of PROC IML*/
OPTIONS LS=80 PS=55;
PROC IML;
START LETVAL;
/* DATA IS IN X, M IS THE NUMBER OF LETTER VALUES*/
M = 5;
N = NROW(X);
/* SORT X */
A = X;
X[RANK(X),]= A ;
/* SET THE TABLE OF LETTER VALUES */
LV = REPEAT(0,M,4);

/* D IS THE LENGTH OF THE SUBSAMPLE FOR THE LV */
D = N;
DO I=1 TO M;

```

```

        IF(D > 0) THEN DO;

/* CREATE THE RIGHT SUBSET */
        B = X[1:D,];
/* SORT THE RIGHT SUBSET */
        A = B;
        B[RANK(B),]=A;
/* CREATE THE LEFT SUBSET */
        C = X[(N-D+1):N,];
/* SORT THE LEFT SUBSET */
        A = C;
        C[RANK(C),]=A;
/* THE MEDIANS*/
        DD = INT((D+1)/2);
        IF( D = (2*DD -1)) THEN DO;
            LV[I,1] = B[DD];
            LV[I,3] = C[DD];
        END;
        ELSE DO;
            LV[I,1] = (B[DD]+B[DD+1])/2;
            LV[I,3] = (C[DD]+C[DD+1])/2;
        END;
/*CALCULATE THE MID VALUES*/
        LV[I, 2] = (LV[I, 1] + LV[I, 3])/2;
/*CALCULATE THE SPREAD*/
        LV[I, 4] = LV[I, 3] - LV[I, 1];
        D = DD;
    END;
END;
FINISH;

/*DEFINE X*/
X = {42, 37, 37, 28, 18, 18, 19, 20, 15, 14, 14, 13, 11, 12, 8, 7, 8, 8,
9, 15, 15};

/* RUN THE CODE */
RUN LETVAL;

/* PRINT THE TABLE */
R = {"M" "F" "E" "D" "C"} ;
C = {"Lower" "Mid" "Upper" "Spread"};
PRINT LV[ ROWNAME=R COLNAME=C];
QUIT;

```

• **OUTPUT FROM SAS:**

LV	Lower	Mid	Upper	Spread
M	77.5	77.5	77.5	0
F	7	68.5	130	123
E	3	81.5	160	157
D	2	84.75	167.5	165.5
C	1	88	175	174

LETTER VALUE DISPLAYS USING R: LETTER VALUE FUNCTION.

```
letval <- function(x, k = 4) {
  LV <- c("M", "F", "E", "D", "C", "B", "A", "Z", "Y", "X", "W")
  out <- array(NA, c(k, 4))
  lx <- rx <- sort(x)
  dimnames(out) <- list(LV[1:k], c("LOWER", "UPPER", "MID", "SPREAD"))
  for(i in 1:k) {
    out[i, 1:2] <- c(median(lx), median(rx))
    nn <- (length(lx) + 1)/2
    lx <- lx[1:nn]
    rx <- rev(rev(rx)[1:nn])
  }
  out[, 3] <- (out[, 1] + out[, 2])/2
  out[, 4] <- out[, 2] - out[, 1]
  out
}
```

A more complicated version:

```
letval2 <- function(x, k = 4) {
  LV <- c("M", "F", "E", "D", "C", "B", "A", "Z", "Y", "X", "W")
  out <- array(NA, c(k, 6))
  lx <- rx <- sort(x)
  dimnames(out) <- list(LV[1:k], c("LOWER", "UPPER", "DEPTH", "MID", "SPREAD", "TAIL"))
  for(i in 1:k) {
    out[i, 1:2] <- c(median(lx), median(rx))
    nn <- (length(lx) + 1)/2
    lx <- lx[1:nn]
    rx <- rev(rev(rx)[1:nn])
    out[i, 3] <- nn
  }
  out[, 4] <- (out[, 1] + out[, 2])/2
  out[, 5] <- out[, 2] - out[, 1]
  out[, 6] <- c(0, out[-1, 5])/2/qnorm(1 - 1/2^(2:k))
  out
}
```

• OUTPUT FROM R:

```
> data(stackloss)
> letval2(stack.loss, 7)
```

	LOWER	UPPER	DEPTH	MID	SPREAD	TAIL
M	15.0	15.0	11.0	15.00	0.0	0.000000
F	11.0	19.0	6.0	15.00	8.0	5.930409
E	8.0	32.5	3.5	20.25	24.5	10.648939
D	8.0	37.0	2.0	22.50	29.0	9.451669
C	7.5	39.5	1.5	23.50	32.0	8.589535
B	7.0	42.0	1.0	24.50	35.0	8.124892
A	7.0	42.0	1.0	24.50	35.0	7.238706

5. HOMEWORK:

1. Run the letter value program in SAS and in R for the following data.

```
stack.loss = c(42, 37, 37, 28, 18, 18, 19, 20, 15, 14, 14, 13, 11, 12, 8, 7, 8, 8, 9, 15, 15)
```

```
airflow=scan()  
80 80 75 62 62 62 62 62 58 58 58 58 58 58 50 50 50 50 50 56 70
```

```
Water Temp  
27 27 25 24 22 23 24 24 23 18 18 17 18 19 18 18 19 19 20 20 20
```

```
Acid Conc.  
89 88 90 87 87 87 93 93 87 80 89 88 82 93 89 86 72 79 80 82 91
```

2. Change the SAS program so it will do a proper letter value table for M more than 5 letter values.
3. For a log-normal distribution and a chi-square with $df=1,3,10$ find the degrees of freedom of corresponding t-distribution with similar tail. Use the code posted above and make small modifications. First you need to generate a sample from the distribution, then work with the `letval2` function and use the last column with the ratios as input for the regression. The slope of the regression is the input for the equation

This is an example for the log normal distribution:

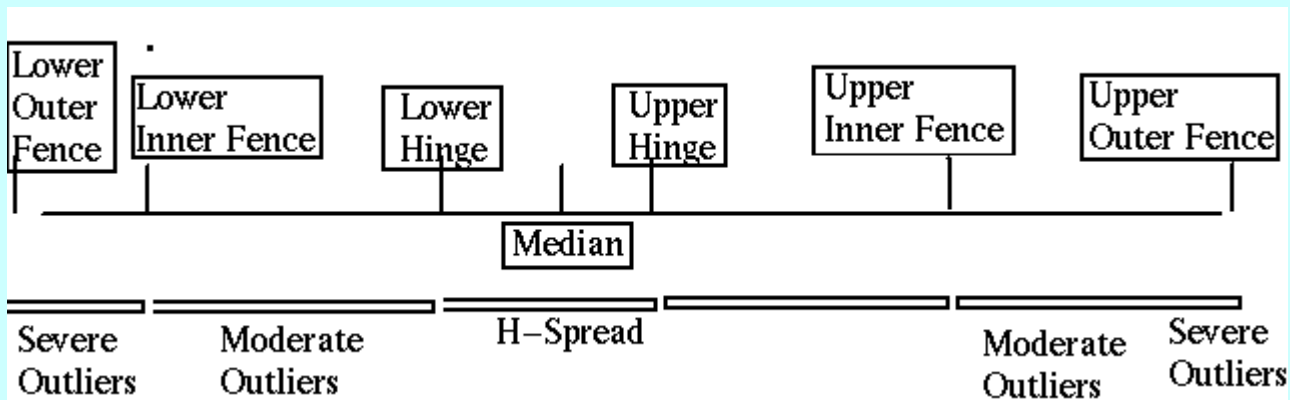
```
x = exp(rnorm(1000))  
xlet = letval2(x,9)  
(beta <- lsfit(1:(nrow(xlet)-1), log(xlet[-1,6])))$coef[2])  
0.1267003  
.5 + 1/(3.25*beta)  
2.928506
```

Lecture 4

I. Box Plots and batch comparisons

1. BOX PLOTS

- **MEDIAN:** Center of the data
- **HINGES:**
More or less like quartiles. In fact you do not need to learn this definition if you do not wish to. Just use the definition of quartile.
- **HSPREAD = upper hinge - lower hinge. (Similar to IQR)**
- **INNER FENCES:**
 - Lower inner fence = lower hinge - 1.5 Hspread
 - Upper inner fence = upper hinge + 1.5 Hspread
- **OUTER FENCES :**
 - Lower outer fence = lower hinge - 3 Hspread
 - Upper outer fence = Upper hinge + 3 Hspread
- **ADJACENT VALUS**
closest point to the inner fences toward the median.



2. Example:

data 6, 9, 12, 13, 13, 15, 15, 17, 27

depth 1, 2, 3, 4, 5, 4, 3, 2, 1

H-spread = 15 - 12 = 3

Inner fences = 12 - 3 * 1.5, 15 + 3 * 1.5 = 7.5, 19.5

outer fences = 3, 24

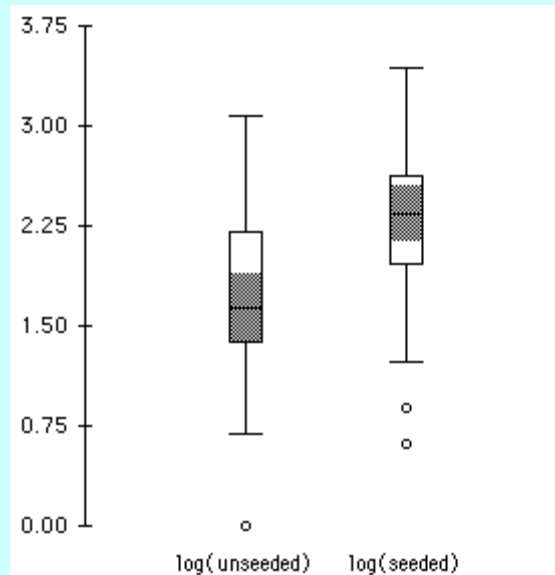
adjacent values = 9, 17

Example: Clouds were randomly seeded or not with silver nitrate. Rainfall amounts were recorded from the clouds. The purpose of the experiment was to determine if cloud seeding increases rainfall. The rainfall distributions are more nearly symmetric after a log transformation. The log transformation also makes the variance of the two groups more nearly equal.

After a log transformation, a pooled t-test may be appropriate. (Without a transformation it is neither appropriate (failing both the normality and equal variance assumptions) nor significant at .05.) Without transforming, a Mann-Whitney U test would be appropriate.

A boxplot of rainfall for the two groups of clouds is helpful.

Image: Side by side boxplots of the two logged variables.

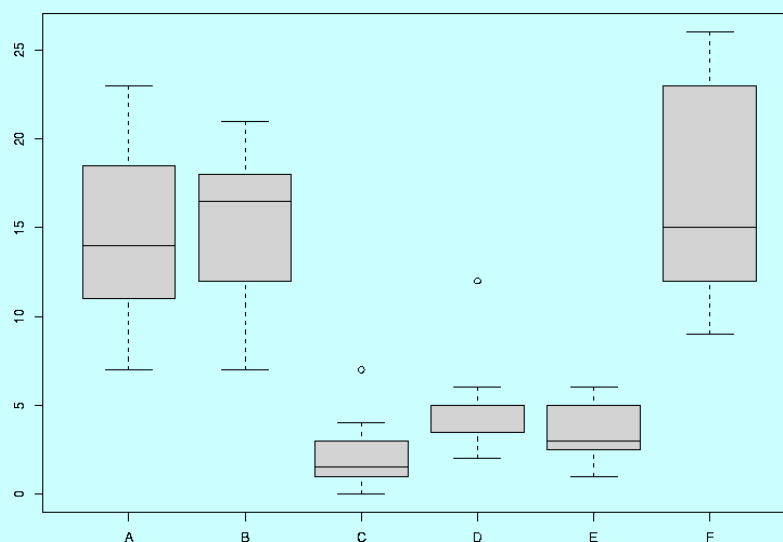


3. Example on R:

The function boxplot produces boxplots in R

Data: Count of insects after spraying by spray factor.

```
data(InsectSprays)  
boxplot(count~spray,data=InsectSprays,col="lightgray")
```



4. SAS PROGRAM USING PROC GPLOT:

In SAS there are several ways to get boxplots

- a. Use Proc Boxplot;
- b. Interactive data analysis menu
- c. Use proc gplot with the symbol option

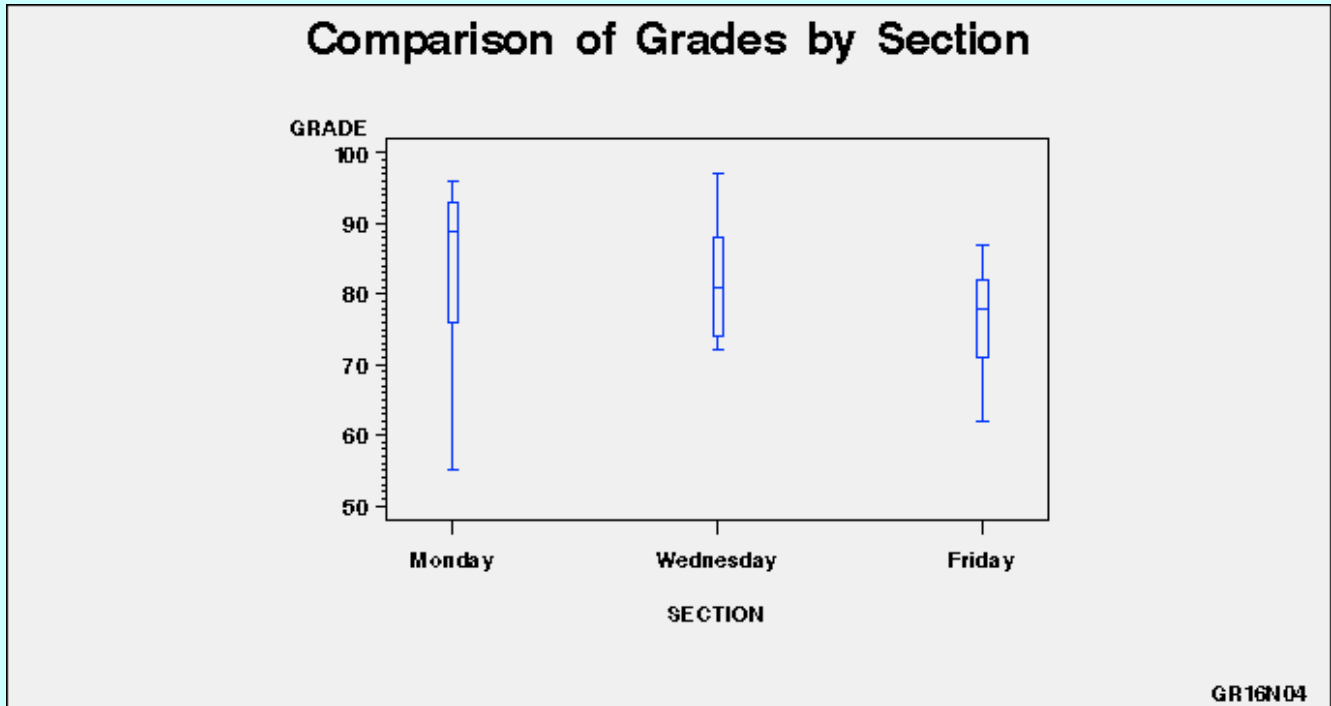
This is code from SAS that generates a boxplot using option c.

```
/******  
/*          S A S   S A M P L E   L I B R A R Y          */  
/*          */  
/*    NAME: GR16N04          */  
/*    TITLE: Creating a Box Plot - GR16N04          */  
/*    PRODUCT: GRAPH          */  
/*    SYSTEM: ALL          */  
/*    KEYS: graphics symbol gplot interpol box reference */  
/*    PROCS: GPLOT          */  
/*    DATA: INTERNAL          */  
/*          */  
/*    REF: SAS/GRAPH REFERENCE GUIDE          */  
/*    MISC:          */  
/*          */  
/******  
  
/* set the graphics environment */  
goptions reset=global gunit=pct border  
        ftext=swissb htitle=6 htext=3;  
/* create the data set GRADES */  
data grades;  
    input section $ grade @@;  
    cards;  
A 74 A 89 A 91 A 76 A 87 A 93 A 93 A 96 A 55  
B 72 B 72 B 84 B 81 B 97 B 78 B 88 B 90 B 74  
C 62 C 74 C 71 C 87 C 68 C 78 C 80 C 85 C 82  
;  
run;  
  
/* define title and footnote */  
title 'Comparison of Grades by Section';  
footnote j=r 'GR16N04  ';  
/* define symbol characteristics */  
symbol interpol=boxt10 /* box plot          */  
        cv=red /* plot symbol color          */  
        co=blue /* box and whisker color */  
        width=6 /* line width          */  
        value=square /* plot symbol          */  
        height=4; /* symbol height          */  
/* define axis characteristics */  
axis1 value=('Monday' 'Wednesday' 'Friday')  
        offset=(5,5)  
        length=50;  
/* generate plot */  
proc gplot data=grades;  
    plot grade*section / haxis=axis1  
                        vaxis=50 to 100 by 10  
                        frame
```

```
des='GR16N04-1';
```

```
run;  
quit;
```

Image: Boxplots of Grades By Section



Go back to the Cloud seed data and see if it is of without the use of log transformation

```
/*  
*****  
/* SAS SAMPLE LIBRARY */  
/* NAME: GR16N04 */  
/* TITLE: Creating a Box Plot - GR16N04 */  
/* PRODUCT: GRAPH */  
/* SYSTEM: ALL */  
/* KEYS: graphics symbol gplot interpol box reference */  
/* PROCS: GPLOT */  
/* DATA: INTERNAL */  
/* REF: SAS/GRAPH REFERENCE GUIDE */  
*****  
/* set the graphics environment */  
goptions reset=global gunit=pct border  
ftext=swissb htitle=6 htext=3;  
/* create the data set GRADES */  
data clouds;  
input Clouds $ Rain ;  
cards;  
U 1202.6  
S 2745.6  
U 830.1  
S 1697.8  
U 372.4  
S 1656.0  
U 345.5  
S 978.0
```

```

U 321.2
S 703.4
U 244.3
S 489.1
U 163.0
S 430.0
U 147.8
S 334.1
U 95.0
S 302.8
U 87.0
S 274.7
U 81.2
S 274.7
U 68.5
S 255.0
U 47.3
S 242.5
U 41.1
S 200.7
U 36.6
S 198.6
U 29.0
S 129.6
U 28.6
S 119.0
U 26.3
S 118.3
U 26.1
S 115.3
U 24.4
S 92.4
U 21.7
S 40.6
U 17.3
S 32.7
U 11.5
S 31.4
U 4.9
S 17.5
U 4.9
S 7.7
U 1.0
S 4.1
;
run;
proc print;
  /* define title and footnote */
  title 'Comparison of Rain Level for seeded and unseeded clouds';
  footnote j=r 'GR16N04  ';

  /* define symbol characteristics */
  symbol interpol=boxt10 /* box plot */
          cv=red /* plot symbol color */
          co=blue /* box and whisker color */
          width=6 /* line width */
          value=square /* plot symbol */
          height=4; /* symbol height */

  /* define axis characteristics */
  axis1 value=('Unsided Cloud' 'Seeded Cloud')

```

```

offset=(15,15)
length=50;

/* generate plot */
proc gplot data=clouds;
  plot rain*clouds/ haxis=axis1
                    vaxis=50 to 100 by 10
                    frame
                    des='GR16N04-1';
run;
quit;

```

Image: Boxplots of Rain by Seeded

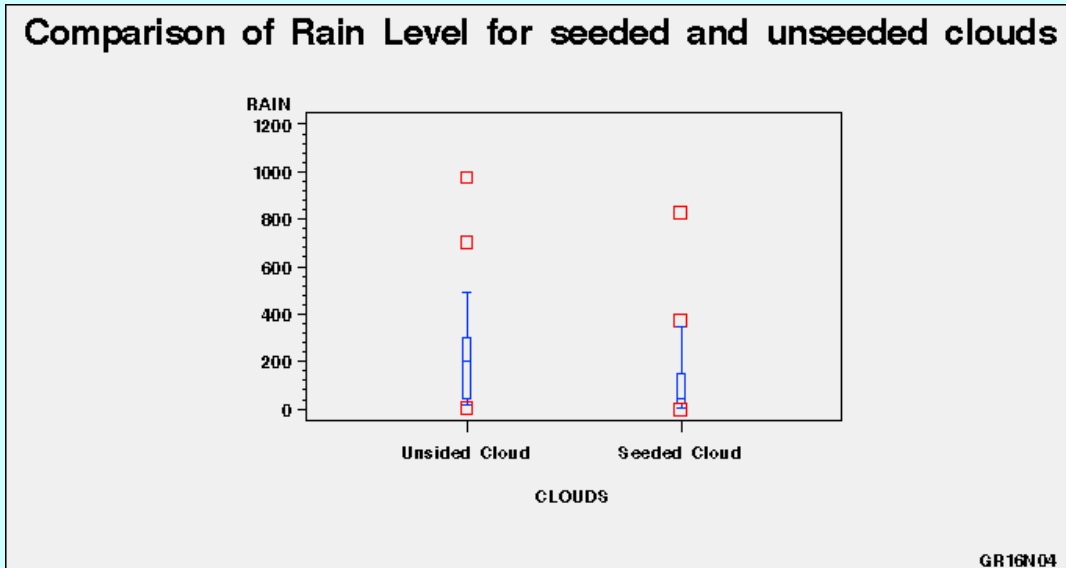
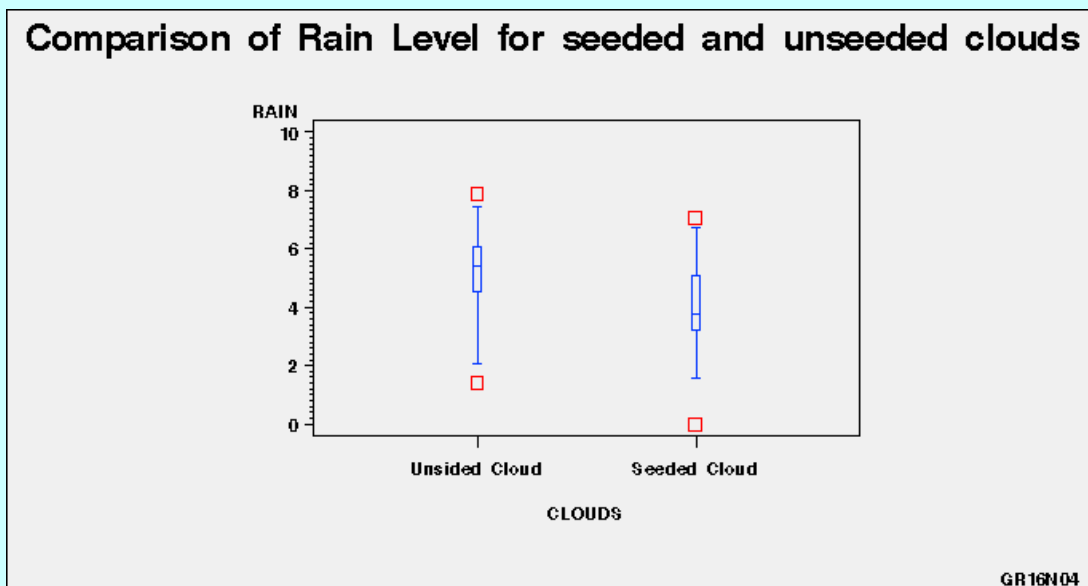


Image: Boxplots of Log(Rain) by Seeded



5. SAS PROGRAM USING PROC BOXPLOT:

```
data clouds;
  set clouds ;
  if Clouds eq 'U' then cld = 1;
  if Clouds eq 'S' then cld = 2;
run;

proc sort data=clouds; by cld;
proc boxplot;
plot rain*cld;
run;
```

6. Homework II

1. Using the above Boxplot SAS code in PROC IML, generate your own code to do a boxplot of the square root of rain, and of the $1/\sqrt{\text{rain}}$. This will show the effect of other transformations that could be used to represent this data.
2. Do the same with the PROC BOXPLOT code. Which one do you prefer and why?
3. Repeat the boxplot using the R code and compare the graph with the ones produced by SAS .

II. THE USE OF POWER TRANSFORMATIONS

1. Definition of Power Transformation:

$$\begin{aligned} \text{if } p = 0 & \quad \log(x) \\ \text{if } p > 0 & \quad x^p \\ \text{if } p < 0 & \quad -x^p \end{aligned}$$

2. SPREAD VS LEVEL PLOT

Compare several datasets with boxplots.

Problem: Sometimes the boxplots are hard to compare because the spread of measurements may depend on the level. That is large values have large spread and small values have small spread.

Solution: Use power transformations to make the scales of the samples more comparable.

2.1 Method for calculating p for the power transformation:

- We will call d_F to the fourths spread.
- Use equation $d_F = c \text{ Median}^b$
- Then take logs
- $\log(d_F) = c' + b \log(\text{Median})$
- and calculate b using linear regression.
- Then the desired power is $p=1-b$

2.2 SAS Example:

EXAMPLE CODE IN SAS

OUTPUT FILE

2.3 R:

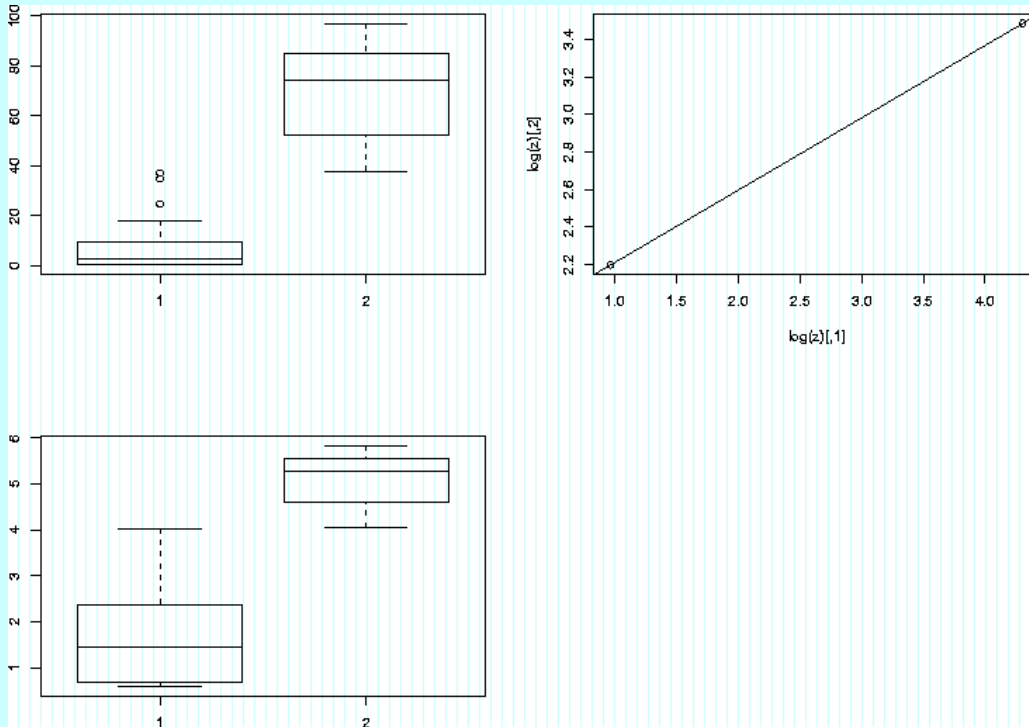
```

SprVsLevel = function(u1,u2,...) {
  x = list(u1,u2,...)
  z <- sapply(x, function(z) {u=letval2(z);c(u[1,1],u[2,5])})
  z <- t(z)
  b <- lsfit(log(z[, 1]), log(z[, 2]))$coef
  boxplot(x)
  plot(log(z))
  abline(b)
  lz <- sapply(x, function(u, uu) u^uu, uu=1-b[2])
  if( is.matrix(lz))
  boxplot(data.frame(lz)) else boxplot(lz)
  1 - b[2]
}

letval2 = function(x, k = 4) {
  LV <- c("M", "F", "E", "D", "C", "B", "A", "Z", "Y", "X", "W")
  out <- array(NA, c(k, 6))
  lx <- rx <- sort(x)
  dimnames(out)=list(LV[1:k],c("LOWER","UPPER","DEPTH","MID","SPREAD","TAIL"))
  for(i in 1:k) {
    out[i, 1:2] <- c(median(lx), median(rx))
    nn <- (length(lx) + 1)/2
    lx <- lx[1:nn]
    rx <- rev(rev(rx)[1:nn])
    out[i, 3] <- nn
  }
  out[, 4] <- (out[, 1] + out[, 2])/2
  out[, 5] <- out[, 2] - out[, 1]
  out[, 6] <- c(0, out[-1, 5]/2/qnorm(1 - 1/2^(2:k)))
  out
}
x<-c(0.250,0.375,0.250,0.250,0.250,0.250,1.250,3.250,2.000,1.375,34.750,
0.625,37.125,12.000,18.125,5.000,10.000,5.250,4.875,0.375,7.000,1.000,0.250,9.3
75,7.125,24.750)
y<-c(85.000,68.875,97.000,83.500,65.625,52.500,51.000,77.950,85.125,66.625,
87.625,94.750,58.875,49.650,72.250,48.625,69.250,82.750,52.125,89.500,65.125,8
0.375,47.875,86.125,79.375,37.875,76.000,49.250,86.125,88.000)
par(mfrow=c(2,2))
  SprVsLevel(x,y)

```

0.6144924



3. TRANSFORMING FOR SYMMETRY

Problem: We have a sample of observations that shows moderate to strong skewness.

Solution: Use power transformations to make the sample more symmetric.

3.1 Method for calculating p for the power transformation:

For several letter values define

$$Y = ((X_U - M)^2 - (M - X_L)^2) / (4M)$$
$$Z = (X_U + X_L) / 2 - M$$

- Plot Y VS Z
- b is the slope of Y vs Z
- If the result is linear then $p = 1 - b$ is the power for the transformation:
- $T(X) = X^p$ or maybe $k X^p$

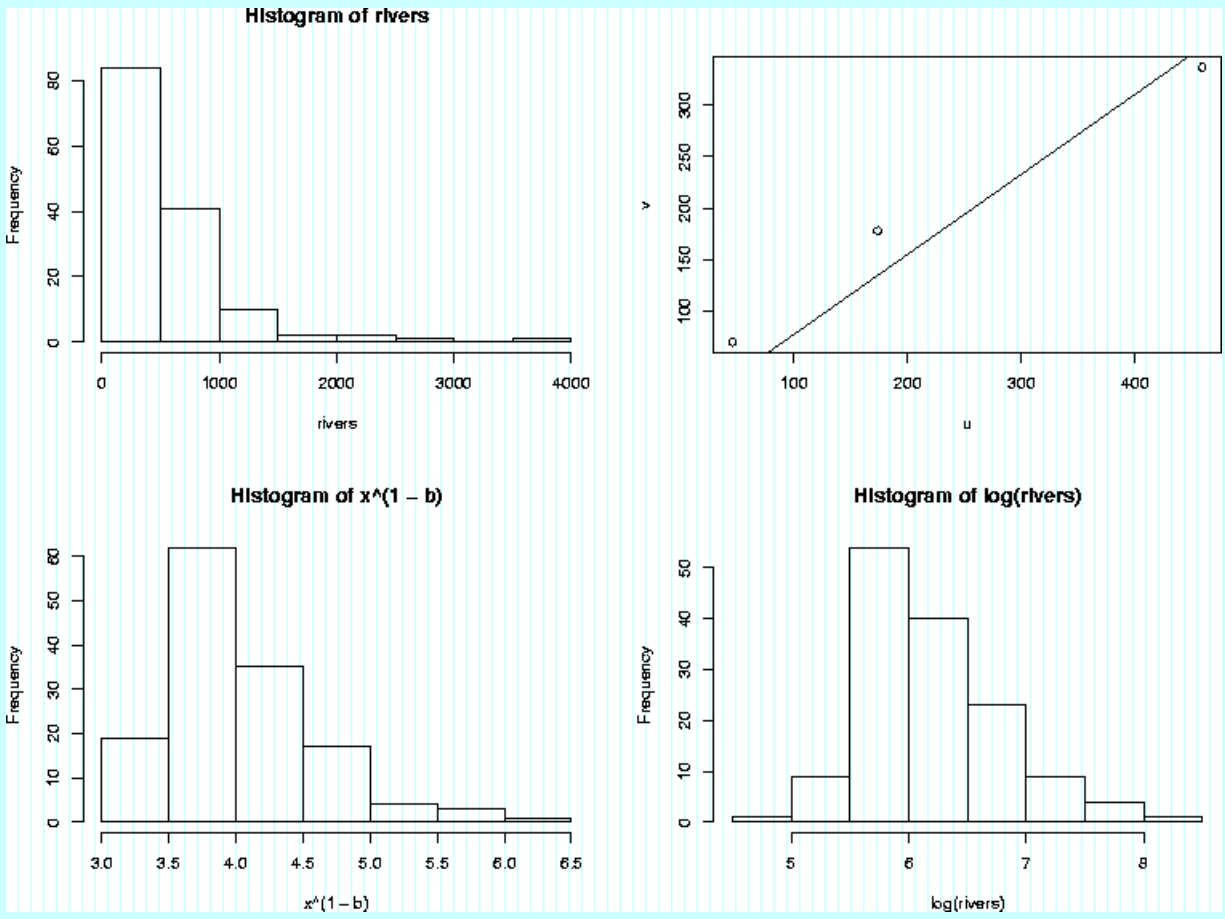
3.2 R:

```
letval2 <- function(x, k = 4)
{
```

```

LV <- c("M", "F", "E", "D", "C", "B", "A", "Z", "Y", "X", "W")
out <- array(NA, c(k, 6))
lx <- rx <- sort(x)
dimnames(out) <- list(LV[1:k], c("LOWER", "UPPER", "DEPTH", "MID", "SPREAD", "TAIL"))
for(i in 1:k) {
  out[i, 1:2] <- c(median(lx), median(rx))
  nn <- (length(lx) + 1)/2
  lx <- lx[1:nn]
  rx <- rev(rev(rx)[1:nn])
  out[i, 3] <- nn
}
out[, 4] <- (out[, 1] + out[, 2])/2
out[, 5] <- out[, 2] - out[, 1]
out[, 6] <- c(0, out[-1, 5]/2/qnorm(1 - 1/2^(2:k)))
out
}
sym <- function(x, k = 7) {
  z <- letval2(x, k)
M <- z[1,1]
  u <- ((z[-1,2]-M)^2 + (z[-1,1]-M)^2)/(4*M)
  v <- z[-1, 4] - M
  plot(u, v)
  b <- lsfit(u, v, int = F)$coef
  abline(0, b)
  hist(x^(1 - b))
  1 - b
}
par(mfrow=c(2,2))
data(rivers)
hist(rivers)
sym(rivers,4)
hist(log(rivers))

```



7. HOMEWORK II (cont'd).

4. Using the SAS code below please compute the letter value displays for the following variables: *air flow*, *water temperature*, *acid concentration*

5. We know that the spread and level of the chisquare family are related. Let x , y , z be three samples of size 20 from 3 chi-square distributions with parameters 1, 3, and 5 respectively. (in R: $x=rchisq(20,1)$; $y=rchisq(20,3)$; $z=rchisq(20,5)$). Please calculate the spread Vs level graph. What does it look like?

```

/* This is just an example of PROC IML*/
OPTIONS LS=80 PS=55;
PROC IML;
START LETVAL;
/* DATA IS IN X, M IS THE NUMBER OF LETTER VALUES*/
M = 7;
qnorm = {0.00, 1.35, 2.30, 3.07, 3.73, 4.31, 4.84, 5.32};
N = NROW(X);
/* SORT X */

```

```

A = X;
X[RANK(X),]= A ;
/* SET THE TABLE OF LETTER VALUES */
LV = REPEAT(0,M,5);
/* D IS THE LENGTH OF THE SUBSAMPLE FOR THE LV */
D = N;
DO I=1 TO M;
    IF(D > 0) THEN DO;
/* CREATE THE RIGHT SUBSET */
        B = X[1:D,];
/* SORT THE RIGHT SUBSET IS REMOVED HERE */
/* CREATE THE LEFT SUBSET */
        C = X[(N-D+1):N,];
/* SORT THE LEFT SUBSET IS REMOVED HERE */
/* THE MEDIANS*/
        DD = INT((D+1)/2);
        IF( D = (2*DD -1)) THEN DO;
            LV[I,1] = B[DD];
            LV[I,3] = C[DD];
        END;
        ELSE DO;
            LV[I,1] = (B[DD]+B[DD+1])/2;
            LV[I,3] = (C[DD]+C[DD+1])/2;
        END;
/*CALCULATE THE MID VALUES*/
        LV[I, 2] = (LV[I, 1] + LV[I, 3])/2;
/*CALCULATE THE SPREAD*/
        LV[I, 4] = LV[I, 3] - LV[I, 1];
        if I > 1 then LV[I, 5] = LV[I, 4]/qnorm[I] ;
        D = DD;
    END;
END;
FINISH;

RN = {"M" "F" "E" "D" "C" "B" "A" "Z"} ;
CN = {"LOWER" "MID" "UPPER" "SPREAD","SP_RATIO"};

X={42,37,37,28,18,18,19,20,15,14,14,13,11,12,8,7,8,8,9,15,15};
title 'Stack Loss data';          /* RUN THE CODE */
RUN LETVAL;                       /* PRINT THE TABLE */
PRINT LV[ ROWNAME=RN COLNAME=CN];

X={80,80,75,62,62,62,62,62,58,58,58,58,58,58,50,50,50,50,50,56,70};
title 'AIR FLOW';                 /* RUN THE CODE */
RUN LETVAL;                       /* PRINT THE TABLE */
PRINT LV[ ROWNAME=RN COLNAME=CN];

X={27,27,25,24,22,23,24,24,23,18,18,17,18,19,18,18,19,19,20,20,20};
title 'Water Temperature';       /* RUN THE CODE */
RUN LETVAL;                       /* PRINT THE TABLE */
PRINT LV[ ROWNAME=RN COLNAME=CN];

X={89,88,90,87,87,87,93,93,87,80,89,88,82,93,89,86,72,79,80,82,91};
title 'Acid Concentration';      /* RUN THE CODE */
RUN LETVAL;                       /* PRINT THE TABLE */
PRINT LV[ ROWNAME=RN COLNAME=CN];
QUIT; AIR FLOW Water Temperature Acid Concentration

```

Lecture 5

Two way ANOVA

ONE-WAY ANOVA:

Linear Model: $x_{gi} = \mu_g + \varepsilon_{gi}$, $g=1, \dots, k$ $i=1, \dots, n_g$

Assumptions: Errors ε_{gi} are (i) independent (ii) normally distributed (iii) with equal variances.

Fit: Data = group effect + residual

Hypothesis Testing: All means are equal Vs some are not equal. ANOVA

(a) $H_0 : \mu_1 = \mu_2 = \dots = \mu_k$

(b) $H_a : \mu_i \neq \mu_j$

LINEAR MODEL FOR TWO WAY TABLE:

This is a typical dataset where we have a response and two factors.

Example:

	B1	B2	B3
clarion	32.7	32.3	31.5
clinton	32.1	29.7	29.1
knox	35.7	35.9	33.1
o'neill	36.0	34.2	31.2
compost	31.8	28.0	29.2
wabash	38.2	37.8	31.9
webster	32.5	31.1	29.7

Fitted model: Data = Tot Mean + Row Effect + Col Eff + Residual

	B1	B2	B3	row effect
clarion	-1.052	-0.024	1.076	-0.390
clinton	0.214	-0.757	0.543	-2.257
knox	-0.786	0.843	-0.057	2.343
o'neill	0.614	0.243	-0.857	1.243
compost	0.548	-1.824	1.276	-2.890
wabash	0.648	1.676	-2.324	3.410
webster	-0.186	-0.157	0.343	-1.457
col eff	1.586	0.157	-1.743	32.557

SAS DOES IT IN A DIFFERENT WAY, IT SETS THE EFFECT FOR ONE OF THE GROUPS AS ZERO.

		ESTIMATE	TVALUE	PVALUE	STD ERROR
INTERCEPT		29.3571	B 34.96	0.0001	0.839703
TYPE					
clarion		1.0667	B 1.02	0.3285	1.047294
clinton		-0.800	B -0.76	0.4597	1.047294
knox		3.800	B 3.63	0.0035	1.047294
o'neill		2.700	B 2.58	0.0242	1.047294
compost		-1.433	B -1.37	0.1962	1.047294
wabash		4.867	B 4.65	0.0006	1.047294
webster		0.000	B .	.	.
BLOCK	1	3.3285	B 4.85	0.0004	0.685615
	2	1.900	B 2.77	0.0169	0.685615
	3	0.000	B .	.	.

COMPUTE ANOVA TABLE:

In order to test the significance of row effects or column effects.

Source	DF	Type I SS	F Value	Pr > F
TYPE	6	103.15142	10.45	0.0004
BLOCK	2	39.03714	11.86	0.0014

Source	DF	Type III SS	F Value	Pr > F
TYPE	6	103.15142	10.45	0.0004
BLOCK	2	39.03714	11.86	0.0014

MULTIPLE COMPARISONS:

Once we find that a factor is significant then we need to explore the difference between the corresponding factor levels. We do that using a multiple comparison procedure such as Tukey or Duncan.

Another useful tool is to use *contrasts* if we are interested in testing only a few comparisons or in some linear combinations of the parameters.

This is the SAS code and output file

```
options ps=50 ls=70;
*-----snapdragon experiment-----*
| as reported by stenstrom, 1940, an experiment was |
| undertaken to investigate how snapdragons grew in |
| various soils. each soil type was used in three   |
| blocks.                                           |
*-----*
data plants;
  input type $ @;
  do block=1 to 3;
    input stemleng @;
    output;
  end;
cards;
clarion 32.7 32.3 31.5
clinton 32.1 29.7 29.1
knox    35.7 35.9 33.1
o'neill 36.0 34.2 31.2
compost 31.8 28.0 29.2
wabash  38.2 37.8 31.9
webster 32.5 31.1 29.7
```

```

;
proc glm;
  class type block;
  model stemleng=type block; run;
proc glm order=data;
  class type block;
  model stemleng=type block / solution;
  means type / bon duncan tukey;

*-type-order---clrn-cltn-knox-oneil-cpst-wbsh-wstr;
contrast 'compost v others' type -1 -1 -1 -1 6 -1 -1;
contrast 'knox vs oneill' type 0 0 1 -1 0 0 0;
run;

```

[OUTPUT: Output file from "glm.sas"](#)

INCOMPLETE DESIGNS

```

120 6559 1240    6   71  237   40  689  165  855
202  233  165   62    5  385   40   74   25   36
   15   22   34  129   32   54   23   48   10    1

```

This is the data but with some missing observations:

```

              B1   B2   B3
clarion    32.7 32.3  NA
clinton    32.1 29.7 29.1
knox       35.7 35.9 33.1
o'neill    NA  34.2 31.2
compost    31.8 28.0 29.2
wabash     38.2 37.8 31.9
webster    32.5  NA  29.7

```

Row Effects :

```

clarion  clinton  knox  o'neill  compost  wabash  webster
0.05238095 -2.147619 2.452381 0.252381 -2.780952 3.519048 -1.34761

```

Column Effects:

```

      B1      B2      B3
1.427778 0.3111111 -1.738889

```

Main Effect: 32.44762

So what is SAS going to do?

	ESTIMATE	TVALUE	PVALUE	STD ERROR
INTERCEPT	29.372 B	27.73	0.0001	1.0591
TYPE				
clarion	0.281 B	0.20	0.8491	1.4390
clinton	-0.969 B	-0.76	0.4682	1.2804
knox	3.630 B	2.84	0.0196	1.2804
o'neill	2.209 B	1.54	0.1591	1.4390
compost	-1.603 B	-1.25	0.2421	1.2804
wabash	4.696 B	3.67	0.0052	1.2804
webster	0.000 B	.	.	.
BLOCK 1				
1	3.455 B	4.16	0.0025	0.8308
2	2.236 B	2.69	0.0247	0.8308
3	0.000 B	.	.	.

We look at the ANOVA table and we see that order matters.

TYPE BEFORE BLOCK

```

Source  DF      Type I SS  F Value  Pr > F

```

TYPE	6	95.93611111	8.42	0.0028
BLOCK	2	33.76848485	8.89	0.0074
Source	DF	Type III SS	F Value	Pr > F
TYPE	6	98.19681818	8.62	0.0026
BLOCK	2	33.76848485	8.89	0.0074

BLOCK BEFORE TYPE

Source	DF	Type I SS	F Value	Pr > F
BLOCK	2	31.50777778	8.30	0.0091
TYPE	6	98.19681818	8.62	0.0026
Source	DF	Type III SS	F Value	Pr > F
BLOCK	2	33.76848485	8.89	0.0074
TYPE	6	98.19681818	8.62	0.0026

THE BASIC STATS

Source	DF	Sum of Squares	F Value	Pr > F
Model	8	129.70459596	8.54	0.0021
Error	9	17.08484848		
Total	17	146.78944444		

R-Square	C.V.	STEMLENG Mean
0.883610	4.238642	32.5055556

Lecture 6

Power transformations:

$$\begin{aligned} \text{if } p > 0 & \quad x^p \\ \text{if } p = 0 & \quad \log(x) \\ \text{if } p < 0 & \quad -x^p \end{aligned}$$

In a more formal way

$$\begin{aligned} \text{if } p = 0 & \quad \log(x) \\ \text{Otherwise} & \quad (x^p - 1)/p \end{aligned}$$

A MORE GENERAL TRANSFORMATION: $A + B X^P$ CAN BE USED TO CHANGE THE LOCATION AND SCALE BUT NOT THE SHAPE.

ANOTHER MORE: $A + B (X - C)^P$

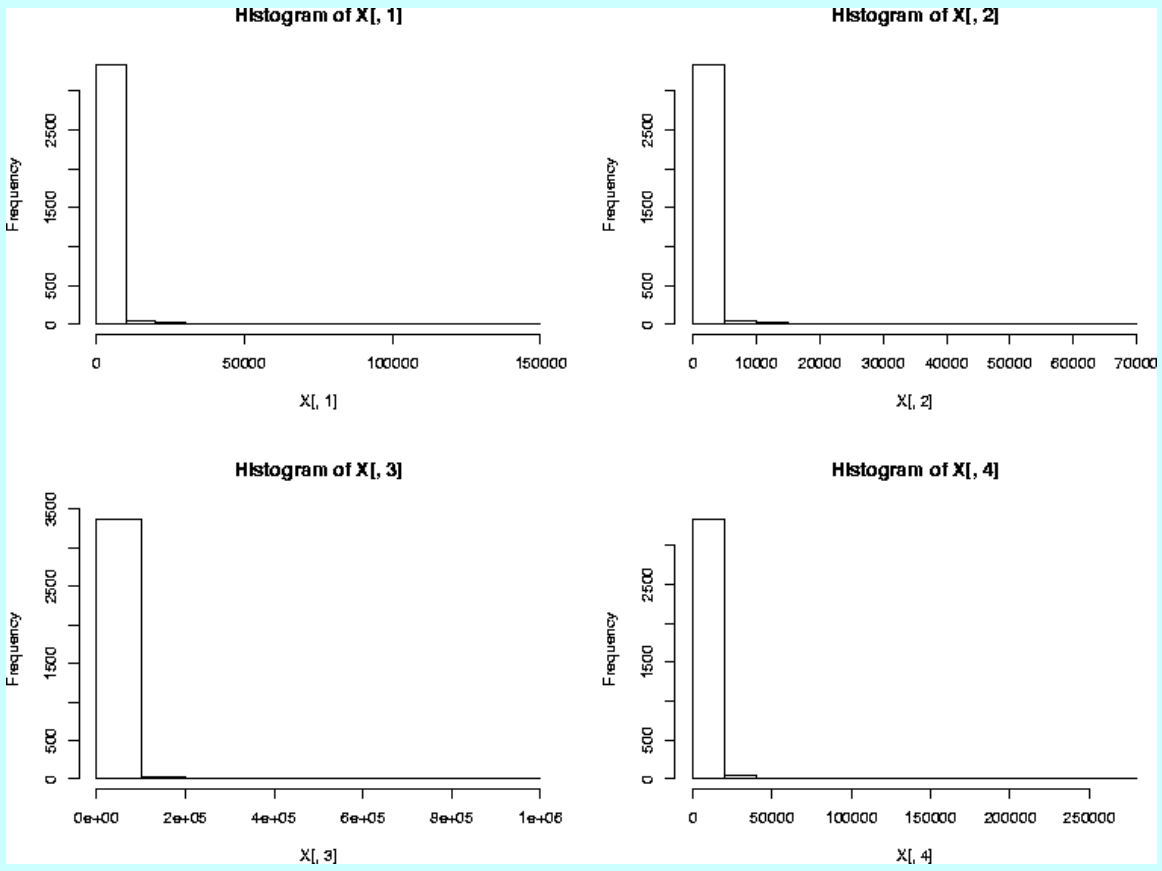
For several letter values define

$$\begin{aligned} Y &= ((X_U - M)^2 - (M - X_L)^2) / (4M) \\ Z &= (X_U + X_L) / 2 - M \end{aligned}$$

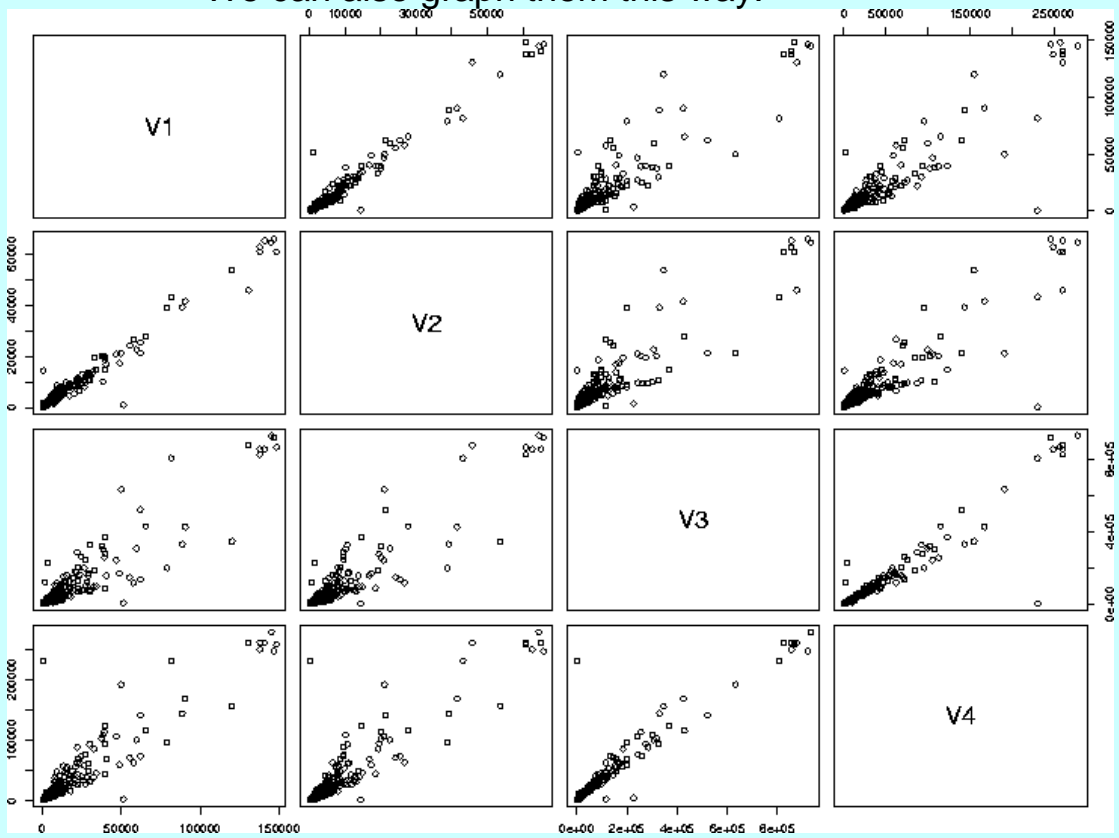
- Plot Y VS Z
- b is the slope of Y vs Z
- If the result is linear then $p = 1 - b$ is the power for the transformation:
- $T(X) = X^p$ or maybe $k X^p$

EXAMPLE DNA MICRO ARRAY DATA:

DNA expression data is perhaps the most important tool in drug development and drug discovery. We have here an experiment with four arrays and we want to find out if they differ from other similar sets.



We can also graph them this way:



Rweb code:
pairs(X)

```

par(mfrow=c(2,2))
hist(X[,1])
hist(X[,2])
hist(X[,3])
hist(X[,4])

```

The data is in :

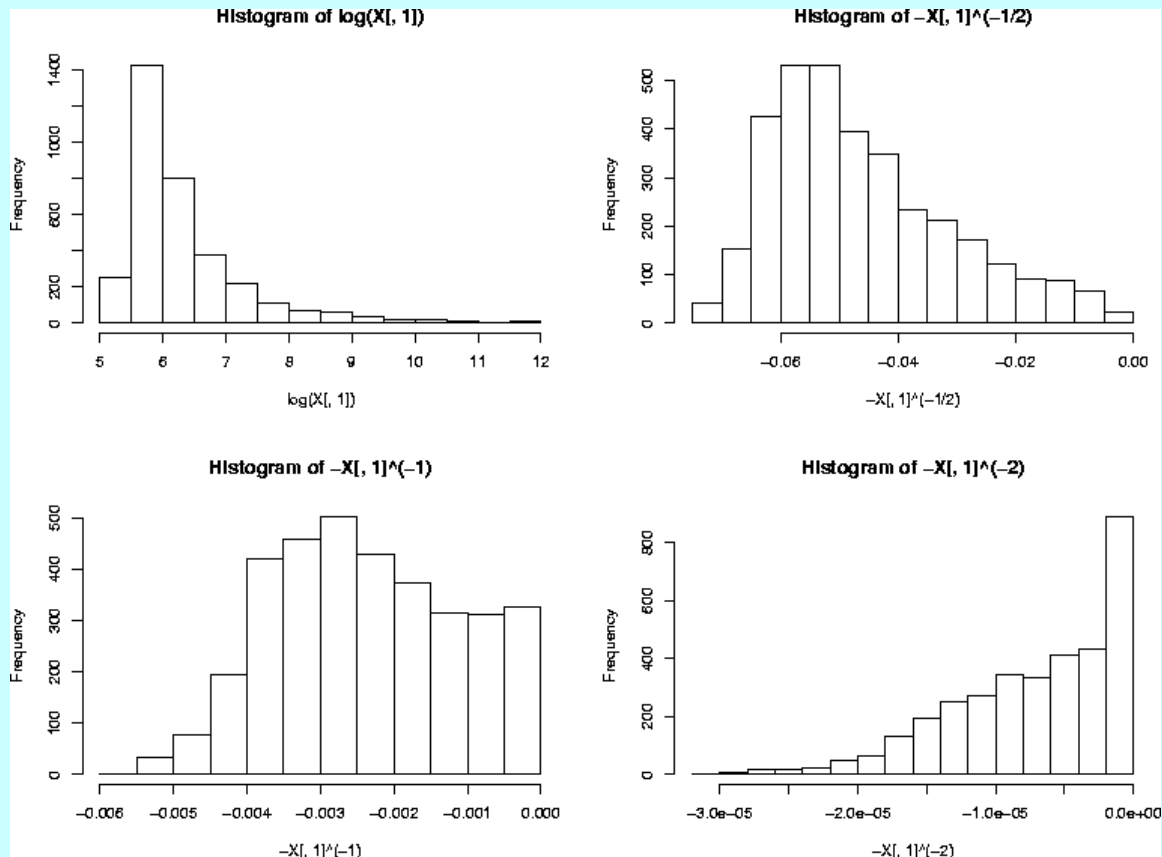
<http://www.rci.rutgers.edu/~cabrera/486/micro.txt>

Take the first one and try to find a power transformation

```

hist(log(X[,1]),title="log")
hist(- X[,1]^(-1/2),title="p^-1/2")
hist(- X[,1]^(-1),title="p^-1")
hist(- X[,1]^(-2),title="p^-2")
#

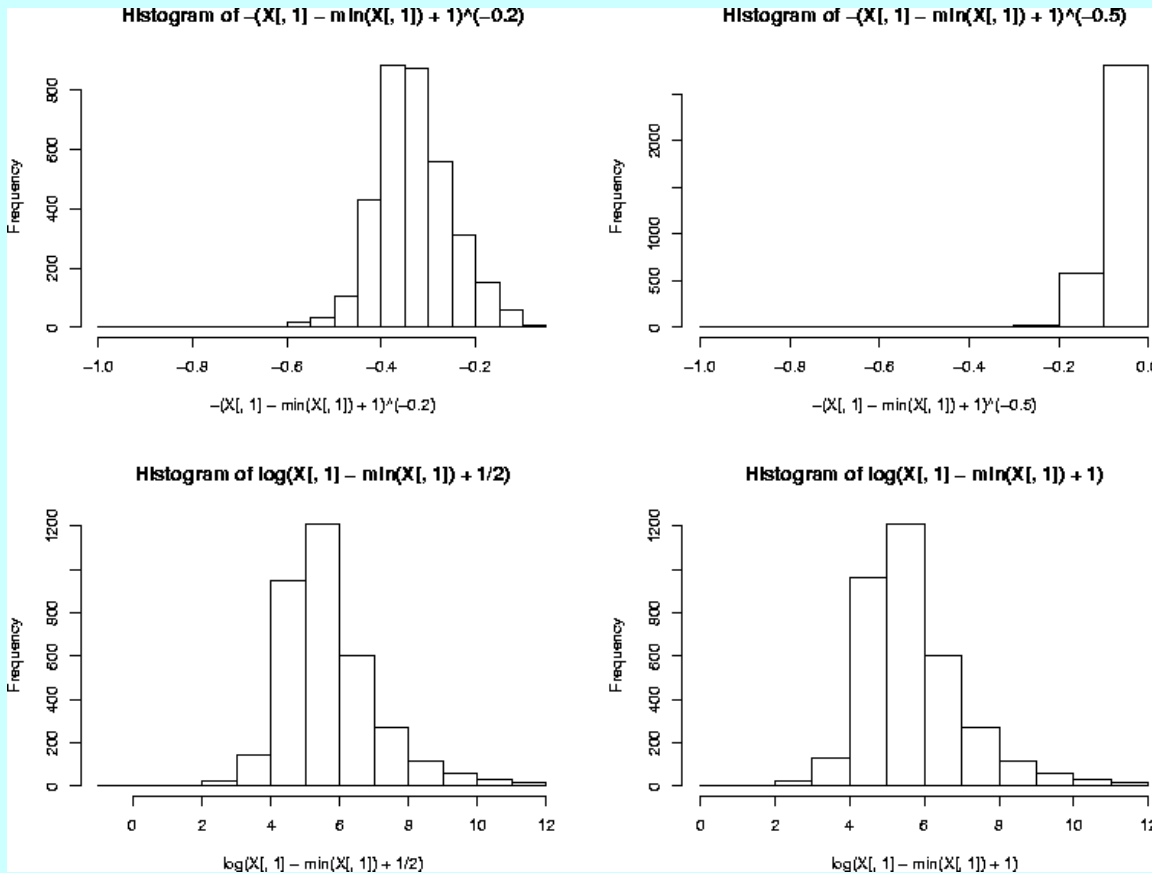
```



```

hist(-(X[,1]-min(X[,1])+1)^(-0.2),title="-0.2")
hist(-(X[,1]-min(X[,1])+1)^(-0.5),title="-0.5")
hist(log(X[,1]-min(X[,1])+1/2),title="log")
hist(log(X[,1]-min(X[,1])+1),title="log")

```



Resistant Lines

By now you should know that least-squares is a non-robust procedure and when you fit a line to data you should always check for the presence of outliers. One alternative is to use a method for fitting lines that is robust. There is a multitude of robust procedures. The most intuitive is “least median squares” but there are others like the “three group line”, median slopes and other.

Least Median S – Least Quantile S- Least Trimmed Means S

These methods minimize the median/quantile/trimmed mean of the square residuals:

$$\text{Median}\{(y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2\}$$

$$\text{Quantile}_p\{(y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2\}$$

$$\text{TrimMean}\{(y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2\}$$

Least Squares estimators minimize:
$$\sum_{i=1}^n (y_i - \beta_0 - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2$$

The exact computation of the LMS, LQS and LTS estimators requires solving combinatorial problems that are n^p hard or $O(C_n^{p+1})$. Instead of finding the optimal solution we get an approximated solution by sampling. One such algorithm considers many thousands of subgroups of $p+1$ points chosen at random and evaluates the LMS (or

other) criterion for the hyperplane determined by the group of points. The final estimate is not the optimal one but it might be close enough if the simulation is sufficiently large.

Algorithm for Tukey's three group line:

1. Sort the data by x .
2. Divide the data into 3 equal groups by x .
3. Calculate the centers (median) of each group:

$$(x_L, y_L), (x_M, y_M), (x_R, y_R)$$

4. Define

$$b_0 = (y_R - y_L) / (x_R - x_L)$$

5. Define

$$a_0 = (y_L - b_0 x_L + y_M - b_0 x_M + y_R - b_0 x_R) / 3$$

Compare this with the Least Squares Line, which is clearly not resistant because each observation influences the value of the slope and intercept.

SAS PROGRAM

```
options ps=60 ls=80;
GOPTIONS GUNIT=PCT CBACK=BLACK;

proc iml;
start median;
a =u;
u[rank(u),] = a;
d = nrow(u);
DD = INT((D+1)/2);
IF( D = (2*DD -1)) THEN mu = u[DD];
      ELSE mu = (u[DD]+u[DD+1])/2;
finish;

START r1;
A = y;
y[RANK(X),]= A ;
A = X;
X[RANK(X),]= A ;
n = nrow(x);
m = int(n/3);
u = x[1:m,];
run median;
x1 = mu;
```

```

u = y[1:m,];
run median;
yl = mu;

u = x[(m+1):(n-m),];
run median;
xm = mu;

u = y[(m+1):(n-m),];
run median;
ym = mu;

u = x[(n-m+1):n,];
run median;
xr = mu;

u = y[(n-m+1):n,];
run median;
yr = mu;

b = (yr-yl)/(xr-xl);
a = (yl-b*xl + ym-b*xm + yr-b*xr)/3;
estimate = a||b;
cn = {"Intercept" "Slope"};
print estimate[OLNAME=CN];
finish;

xbox={0 100 100 0};
ybox={0 0 100 100};
year=do(71,86,1); /* initialize YEAR */
price={123.75 128.00 139.75 /* initialize PRICE */
       155.50 139.750 151.500
       150.575 149.125 159.500
       152.375 147.000 134.125
       138.750 123.625 127.125
       125.50};
y = price`;
x = year`;
run rl;
quit;
run;

```

OUTPUT:

```

ESTIMATE Intercept      Slope
          229.40076    -1.147727

```

OUTLIERS:

Suppose that now we change one observation by error:

```

price={123.75 128.00 139.75 /* initialize PRICE */
       155.50 139.750 151.500
       150.575 149.125 159.500
       152.375 147.000 134.125
       138.750 123.625 1027.125 125.50};

```

Then this is what happens to the LS estimator

```
ESTIMATE Intercept      Slope
          181.77955 -0.511364
```

```
data a;
input  year price;
cards;
  71    123.75
  72     128
  73    139.75
  74    155.5
  75    139.75
  76    151.5
  77    150.575
  78    149.125
  79    159.5
  80    152.375
  81     147
  82    134.125
  83    138.75
  84    123.625
  85   1027.125
  86     125.5
;
proc reg; model price = year; run;
```

OUTPUT:

Dependent Variable: PRICE

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	94566.42825	94566.42825	2.058	0.1733
Error	14	643200.39284	45942.88520		
C Total	15	737766.82109			

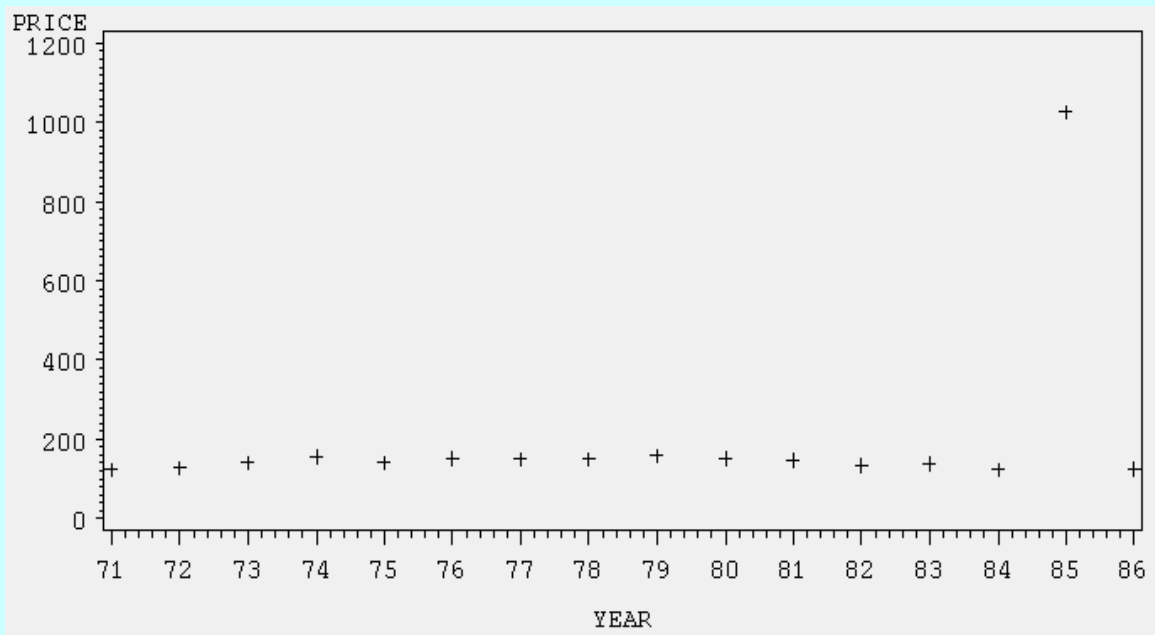
Root MSE	214.34291	R-square	0.1282
Dep Mean	196.62188	Adj R-sq	0.0659
C.V.	109.01275		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > T
INTERCEP	1	-1112.556103	914.08557756	-1.217	0.2437
YEAR	1	16.677426	11.62437667	1.435	0.1733

To see the outlier you can do a graph of the data:

```
proc gplot data=a;
plot price*year / vaxis=0 to 1200 by 200 frame;
run;
```



These are the Least squares estimators without the outlier:

```
INTERCEPT    181.8557
YEAR           -0.52846
```

Notice that they are very similar to what you got from the resistant line.

Fitting nonlinear equations to data.

We have two examples that show the way to fit this kind of models

Kepler's Law

```
# Kepler's data
# Period^2=k*Distance^3
# Distance  Period
#    36      88
#   67.25  224.7
#    93     365.3
#  141.75  687
#  483.8  4332.1

d <- c(36 ,67.25 ,93 , 141.75 , 483.8) # Distance
p <- c( 88, 224.7 ,365.3 ,687 ,4332.1 ) # Period
plot(d,p,xlab="Distance",ylab="Period",pch="o")
# Plot the data. The graph is inconclusive about linearity of relationship.
# Plot residuals to check the goodness of the fit. Residuals show pattern.
rr <- lsfit(d,p)
plot(d,rr$resid,xlab="residuals-distance",ylab="period",pch="o")
# Plot log-log to see if the relationship is a power law
dl <- log(d); pl <- log(p)
```

```

plot(dl,pl,xlab="log-distance",ylab="log-period",pch="o")
abline(lsfit(dl,pl))
rr <- lsfit(dl,pl) # Plot residuals to check the goodness of the fit
plot(dl,rr$resid,xlab="residuals-log-distance",ylab="log-period",pch="o")
dp <- d^3 ; pp <- p^2 # Try several version of Keplers law
plot(dp,pp,xlab="cube-distance",ylab="square-period",pch="o")
abline(lsfit(dp,pp))
rr <- lsfit(dp,pp)
plot(dp,rr$resid,xlab="residuals-cube-distance",ylab="square-period",pch="o")
dq <- d^(3/2)
plot(dq,p,xlab="distance^3/2",ylab="period",pch="o")
abline(lsfit(dq,p))
rr <- lsfit(dq,p)
plot(dq,rr$resid,xlab="residuals-distance^3/2",ylab="period",pch="o")

```

=====

Y VS X

	Coef	Std Err	t Value
Intercept	-471.7008	119.6464	-3.942456
x2	9.80239	0.5161943	18.98972

Residual Standard Error = 188.642 Multiple R-Square = 0.99175
N = 5 F Value = 360.61 on 1, 3 df

=====

log(Y) VS log(X)

	Coef	Std Err	t Value
Intercept	-0.896615	6.807606e-4	-1317.078
x2	1.499643	1.426618e-4	10511.87

Residual Standard Error = 2.77212e-4 Multiple R-Square = 1
N = 5 F Value = 110499512 on 1, 3 df

=====

Y^2 VS X^3

	Coef	Std Err	t Value
Intercept	44.72135	4.34842e1	1.028451e0
x2	0.165729	8.58359e-7	1.930766e5

Residual Standard Error = 86.1907 Multiple R-Square = 1
N = 5 F Value = 3.727856e10 on 1, 3 df

Y VS X^3/2

	Coef	Std Err	t Value
Intercept	0.1130044	5.930967e-2	1.905328e0
x2	0.4070870	1.224807e-5	3.323684e4

Residual Standard Error = 0.1082272 Multiple R-Square = 1
N = 5 F Value = 1.104687e9 on 1, 3 df

Boyle's Law P x V = cte

V	P						
29.750	1.0	5.625	5.0	3.000	10.0	1.500	24.0
19.125	1.5	4.875	6.0	2.625	12.0	1.375	28.0
14.375	2.0	4.250	7.0	2.250	14.0	1.750	20.0
9.500	3.0	3.750	8.0	2.000	16.0	1.250	32.0
7.125	4.0	3.375	9.0	1.875	18.0		

```

boyle <- data.frame(matrix(scan(),ncol=2,byrow=T))
names(boyle) <- c("V","P")
attach(boyle)
par(mfrow=c(2,2),cex=1.07,mar=c(4,4,0.2,0.2),mgp=c(2,1,0))
plot(P,V,pch=16,col=2) # Plot 1
plot(log(P),log(V),pch=16,col=2) # Plot 2
lsfit(log(P),log(V))$coef
[1] 3.2717004 -0.9152016
# the slope is close to -1.
Pm1 <- 1/P
rm1 <- lsfit(Pm1,V)
plot(rm1$resid,V,pch=16,col=2) # Plot 3
# See U-pattern. It shows a possible quadratic fit.
rm2 <- lsfit(cbind(Pm1,Pm1^2),V)
plot(rm2$resid,V,pch=16,col=2) # Plot 4
# Residuals look very good!!
# We have obtained the fit:  $V = a + b 1/P + c (1/P)^2$  where  $a$  ,  $b$ 
and  $c$  are
rm2$coef
[1] 0.4127083 26.0636703 3.2543654
# But the polynomial in the fit equation looks like a taylor
expanssion
# of something, but what?
# A possibility will be to expand  $1/(P-d)$  as a
# polynomial on  $1/P$  near around zero. We get
#  $1/(P-d) = 1/P + d (1/P)^2 + d^2 (1/P)^3 + O( (1/P)^4 )$ 
# So , fit (1) suggests a possible fit ( Fit 2)
#  $y = a_1 + b_1 1/(P -d)$ 
# where  $a_1 = a = 0.41$  ,  $d = c/b = 0.12$ , and  $b_1=26.06$ 
#
nls(formula= V~ a1 + b1/(P-d),data=boyle, start=c(a1=0,b1=1,d=0))
Nonlinear regression model
model: V ~ a1 + b1/(P - d)
data: boyle
      a1      b1      d
0.4016758 26.2392374 0.1055646
residual sum-of-squares: 0.05783763
# We conclude d= 0.1055646
par(mfrow=c(2,2))
d <- 0.1055646
rmd1 <- lsfit(1/(P-d),V)
d <- 0.12
rmd2 <- lsfit(1/(P-d),V)
plot(rmd1$resid,P)
plot(rmd2$resid,P)
boxplot(rmd1$resid,rm2$resid,rmd2$resid,rm1$resid)

```

Multiple Linear Regression procedures

This is the standard model in many statistical problems. The data contains several predictors and one or more responses.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \dots + \beta_p x_p + \text{Error}$$

We calculate the least squares estimators of the regression parameters. We define fitted values and residuals the usual way:

$$y_i = b_0 + b_1 x_{i1} + \dots + b_p x_{ip} + r_i$$

In practice we have a group of predictors that are candidates for the regression equation but we need to determine which ones are appropriate.

There may be collinearities among the predictors

Most datasets contain a number of errors, bad observation or simply unusual observations which we call **OUTLIERS** and **LEVERAGE POINTS**. Regression diagnostics must be used to find out if there are any outliers on the dataset.

Example:

This example comes from Urban Housing in a US city in 1997. The cases correspond to urban units each of the size of a neighborhood. In each unit we observe a response that gives the increase in property taxes for the residential housing units in that neighborhood. The predictors are several demographic variables describing the type unit plus four variables given expenditures in transportation and in roads. The question is to try to assess the impact of expenditures in transportation and roads on property values. This dataset like many other contains a number of outlier and leverage points.

- First we plot the data and we run a regression to check if there are any outliers.
- The graphs are concentrated in the four variables measuring expenditure, but in practice one should plot the response vs all of the predictors.
- The options PR in the model statement produce an output containing the residuals and Cook's D statistic. You must eyeball the numbers and find the corresponding outliers.

```
libname mylib spss 'Reg.por';  
options linesize=70 pagesize=55;
```

```
data a;  
format IN80BO IN80C IN90C INBBID INBP90 INCENP INCODE INCR80 INCR90  
INEMP INMAIN INMAIN2 INMAIN3 INNV INO INREST INT79 INT80 INT90 INV90  
INVAL SEV79 SEV80 SEVR80 SFAMV VAPT79 VAPT80 VCOM79 VCOM80 VFRM79  
VFRM80 VIND79 VIND80 VRES79 VRES80 VTOT79 VTOT80 VVAL79 VVAL80  
comma8. NAMEV1 $CHAR200.;
```

```

set mylib._first_;
if _N_ < 100;
run;

proc plot;
plot ravlchm*(roadacc roadcap transacc transcap);
run;

proc reg data=a;
model ravlchm = hhinc black hslsds setr vac indexr pstu
        roadacc roadcap transacc transcap/P R COLLIN;

```

[Output file from "reg.sas"](#)

Then we find that observations 14 and 13 are outliers so we may omit them for now. We repeat the regression again and we find that #2 is also an outlier.

```

proc reg data=a;
model ravlchm = hhinc black hslsds setr vac indexr pstu
        roadacc roadcap transacc transcap/P R COLLIN;
output out=b p=pred r=res;

proc plot data=b;
plot res*pred res*(roadacc roadcap transacc transcap);
run;

proc reg data=a;
model ravlchm = hhinc black hslsds setr vac indexr pstu
        roadacc roadcap transacc transcap/selection=adjrsq;

proc reg data=a;
model ravlchm = hhinc black hslsds vac indexr roadacc
        roadcap transacc transcap;

```

[Output file from "reg1.sas"](#)

Then we repeat and we find new outliers and so on

[Output file from "reg2.sas"](#)

On the other hand the robust procedure yields the output

```

Intercept      HHINC      BLACK      HSLDS      SETR      VAC
-72.54527 0.00780025 -0.1025825 0.1485328 -36.75587 0.1651357
      INDEXR      PSTU      ROADACC
-0.001151889 4.031584 5.032116
      ROADCAP      TRANSACC      TRANSCAP
48.80187 1.165922 166.9255

```