Statistical Data Analysis

N. Sairam

sairam@cse.sastra.edu

School of Computing, SASTRA University, Thanjavur.

Unit-II

Statistical Models

- Statistical models rely on probabilistic forms of description that have wide application over all areas of science
- They often consist of a deterministic component, with a random component added that is designed to account for residual variation
- The choice of model is crucial in formal analysis
- The choice may be influenced by previous experience with comparable data, by subject area knowledge, and by cautious use of what may emerge from exploratory analysis

Noise Component

- Statistical models combine deterministic and random components
- The random component is often called noise or error and the deterministic component is sometimes thought of as the signal
- It may be helpful to think of the statistical error as the "rough", and of the model prediction as the "smooth"
- Both noise and error are technical terms
- Use of the word error does not imply that there have been mistakes in the collection of the data, though mistakes can of course contribute to making the variability unnecessarily large

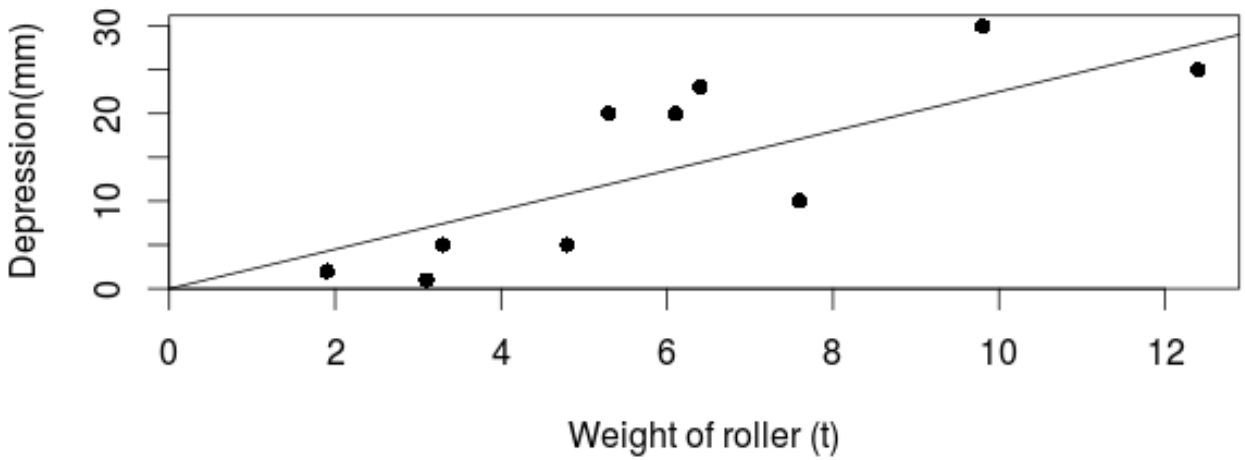
Example

- Let us consider a data from an experiment where different weights of roller were rolled over different parts of a lawn, and the depression noted
- The data seem broadly consistent with the assumption of a signal by which depression is proportional to roller weight, as implied by the solid line in figure
- Variation about this signal is reflected in variation in the values for depression / weight

Example

	weight (t)	depression (mm)	depression / weight
1	1.9	2	1.1
2	3.1	1	0.3
3	3.3	5	1.5
4	4.8	5	1.0
5	5.3	20	3.8
6	6.1	20	3.3
7	6.4	23	3.6
8	7.6	10	1.3
9	9.8	30	3.1
10	12.4	25	2.0

Example



Probability and Sampling Distributions

- Probabilities are defined as relative frequencies, and to be more exact as limits of relative frequencies
- When an experiment is conducted, such as tossing coins, rolling a die, sampling for estimating the proportion of defective units, several outcomes or events occur with certain probabilities
- These events or outcomes may be regarded as a variable which takes different values and each value is associated with a probability
- The values of this variable depends on chance or probability. Such a variable is called a random variable

Types of Random Variables

- Discrete Random Variable
  - Random variables which take a finite number of values or to be more specific those which do not take all values in any particular range are called discrete random variables
  - For example, when 20 coins are tossed,the number of heads obtained is a discrete random variable and it takes values 0,1,...,20
  - These are finite number of values and in this range, the variable does not take values such as 2.8, 5.7 or any number other than a whole number
- Continuous Random Variable
  - In contrast to discrete variable, a variable is continuous if it can assume all values of a continuous scale
  - Measurements of time, length and temperature are on a continuous scale and these may be regarded as examples of continuous variables

Discrete Vs Continuous Random Variables

- A basic difference between these two types of variables is that for a discrete variable, the probability of it taking any particular value is defined
- For continuous variable, the probability is defined only for an interval or range
- The frequency distribution of a discrete random variable is graphically represented as a histogram, and the areas of the rectangles are proportional to the class frequencies
- In continuous variable, the frequency distribution is represented as a smooth curve

Frequency Distributions

1. Observed frequency distributions
   - Observed frequency distributions are based on observations and experimentation
2. Theoretical or Expected frequency distributions
   - Deduce mathematically what the frequency distributions of certain populations should be
   - Such distributions as are expected from on the basis of previous experience or theoretical considerations

Binomial Distribution

- Binomial distribution is a probability distribution expressing the probability of one set of dichotomous alternatives i.e. success or failure
- More precisely, the binomial distribution refers to a sequence of events which posses the following properties:
  1. An experiment is performed under same conditions for a fixed number of trials say, n.
  2. In each trial, there are only two possible outcomes of the experiment "success" or "failure"
  3. The probability of a success denoted by p remains constant from trial to trial
  4. The trials are independent i.e. the outcomes of any trial or sequence of trials do not affect the outcomes of subsequent trials

Binomial Distribution

- Consider a sequence of n independent trials
- If we are interested in the probability of x successes from n trials, then we get a binomial distribution where x takes the values from 0,1,...,n

Definition

A random variable X is said to follow a binomial distribution with parameters n and p if its probability function is given by

$$P[X = x] = \binom{n}{x} p^x q^{n-x}, \; x = 0, 1, 2, ..., n \qquad (1)$$

Binomial Distribution

- The probability of success are the successive terms of the binomial expansion $(q+p)^n$
- The probable frequencies of the various outcomes in N sets of n trials are $N(q+p)^n$
- The frequencies obtained by this expression are known as expected or theoretical frequencies
- On the other hand, the frequencies actually obtained by making experiments are called observed frequencies
- Generally, there is some difference between the observed and expected frequencies but the difference becomes smaller and smaller as N increases

Constants of the Binomial Distribution

- The various constants of the binomial distribution are as follows:
  - Mean = np, Variance = npq. Here mean > variance
  - First moment $\mu_1 = 0$, Second moment $\mu_2 =$ npq
  - Third moment $\mu_3 =$ npq(q-p), Fourth moment $\mu_4 = 3n^2p^2q^2 +$ npq (1-6pq)
  - $\beta_1 = \frac{(q-p)^2}{npq}$, $\gamma_1 = \frac{(q-p)}{\sqrt{npq}}$
  - $\beta_2 = 3 + \frac{1-6pq}{npq}$, $\gamma_2 = \frac{1-6pq}{npq}$

Properties of Binomial Distribution

1. The shape and location of the binomial distribution changes as p changes for a given n or as n changes for a given p. As p increases for a fixed n, the binomial distribution shifts to the right.
2. The mode of the binomial distribution is equal to the value of x which has the largest probability. The mean and mode are equal if np is an integer.

Properties Contd..

3. As n increase for a fixed p, the binomial distribution moves to right, flattens and spreads out. When p and q are equal, the distribution is symmetrical, for p and q may be interchanged without altering the value of any term, and consequently terms equidistant from the two ends of the series are equal. If p and q are unequal, the distribution is skewed. If p is less than 1/2, the distribution is positively skewed and when p is more than 1/2, the distribution is negatively skewed.
4. If n is large and if neither p nor q is too close to zero, the binomial distribution can be closely approximated by a normal distribution with standardized variable given by $Z=\frac{X-np}{\sqrt{npq}}$

Importance of Binomial Distribution

- The binomial probability distribution is a discrete probability distribution that is useful in describing an enormous variety of real life events
- For example, an experimenter wants to know the probability of obtaining diseased trees in a random sample of 10 trees if 10 percent of the trees are diseased
- The answer can be obtained from the binomial probability distribution

Example

- The incidence of disease in a forest is such that 20% of the trees in the forest have the chance of being infected. What is the probability that out of six trees selected, 4 or more will have the symptoms of the disease?

Example

- The incidence of disease in a forest is such that 20% of the trees in the forest have the chance of being infected. What is the probability that out of six trees selected, 4 or more will have the symptoms of the disease?
- Solution: Refer BB
- Fitting a Binomial Distribution:
  1. Evaluate mean of the given distribution and then determine the values of p and q
  2. Expand the binomial $(q+p)^n$
  3. The number of terms in the expanded binomial is equal to one more than n
  4. Multiply each term of the expanded binomial by N (the total frequency) for obtaining the expected frequency in each category

Poisson Distribution

- The Poisson distribution is a discrete probability distribution for the counts of events that occur randomly in a given interval of time (or space)
- If we let X = The number of events in a given interval, Then, if the mean number of events per interval is λ
- The probability of observing x events in a given interval is given by

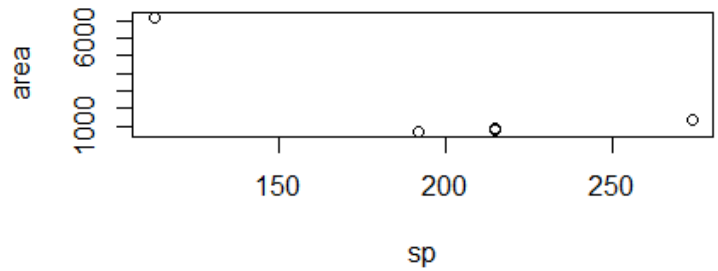
$$P(X = x) = \frac{e^{-\lambda}\lambda^{x}}{x!}, x = 0, 1, 2, 3, ... \quad (2)$$

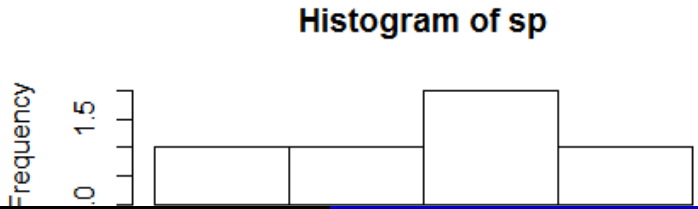
- Note: A Poisson random variable can take on any positive integer value. In contrast, the Binomial distribution always has a finite upper limit

Answers for the I Monthly Test Questions

1. `area<-c(694,905,802,1366,7167)` (0.5 M)
   `sp<-c(192.0,215.0,215.0,274.0,112.7)`(0.5 M)

   `plot(sp,area)`(2 M)
   `hist(sp)`(2 M)





Answers for the I Monthly Test Questions

2.
```
x<-seq(1,100)[0.5M]
cutoff<-55    [0.5 M]
sum<-0
for(i in x)
{if (i>cutoff)
    sum<-sum+1}[2 M]
prop<-sum/100[0.5 M]
print(prop)[0.5 M]
Tracing [1 M]
```

Answers for the I Monthly Test Questions

3. mosaic()(2 M). mosaic(table), where table is a contingency table in array form. mosaic(formula, data=) where formula is a standard R formula, and data specifies either a data frame or table (3 M)
4. addmargins() function to add marginal sums to these tables(3 M). useNA=ifany(2 M)
5. Noise is considered to be a random component(2 M). Explanation (3 M)

Answers for the I Monthly Test Questions

6. (a) 2,3 and 5 will be repeated four, three and two times respectively (2 M)
   (b)c(rep(4,4),rep(3,4),rep(2,4)) (2 M)
   (c)rep(c(3,1,1,5,7),length.out=50) (3 M)
   (d)rep(c(3,1,1,5,7),each=4) (3 M)
7. (a) Refer class notes(6 M)
   (b)cor() returns the correlation between two data sets and cor.test() returns a confidence interval, and tests for no association (2+2 M)
8. Refer class notes

Binomial Distribution in R

- dbinom():We can use the function dbinom() to determine probabilities of having 0, 1 or 2 heads in two coin tosses: dbinom(0:2, size=2, prob=0.5)

```
[1] 0.25 0.50 0.25
```

- pbinom():The function pbinom() can be used to determine cumulative probabilities. To find the probability of atmost 4 infected out of 6 will be pbinom(q=4, size=6, prob=0.2)

```
[1] 0.9984
```

- qbinom():The function qbinom() goes in the other direction, from cumulative probabilities to numbers of events. It is used to compute quantiles, a generalization of the familiar term percentiles. qbinom(p = 0.70, size = 4, prob = 0.5)

```
[1] 3
```

Poisson Distribution

Definition

A Random Variable X is said to follow a Poisson Distribution with parameter λ if its probability function is given by $P[X=x]=\frac{e^{-\lambda}\lambda^{x}}{x!}$, x=0,1,2,... and e=2.7183

Constants of the Poisson Distribution

- The various constants of the Poisson distribution are:
  1. Mean $= \lambda$, Variance $= \lambda$, [Here mean = variance]
  2. First moment $\mu_1 = 0$, Second moment $\mu_2 = \lambda$
  3. Third moment $\mu_3 = \lambda$, Fourth moment $\mu_4 = \lambda + 3\lambda^2$
  4. $\beta_1 = \frac{1}{\lambda}$, $\gamma_1 = \frac{1}{\sqrt{(\lambda)}}$
  5. $\beta_2 = 3 + \frac{1}{\lambda}$, $\gamma_2 = \frac{1}{\lambda}$

Properties of Poisson Distribution

1. As λ increases, the distribution shifts to the right, i.e. the distribution is always a skewed distribution.
2. Mode: When λ is not an integer then unique mode i.e. m=[λ]. When is an integer then bimodal i.e. m= λ and m= λ-1.
3. Poisson distribution tends to normal distribution as λ becomes large

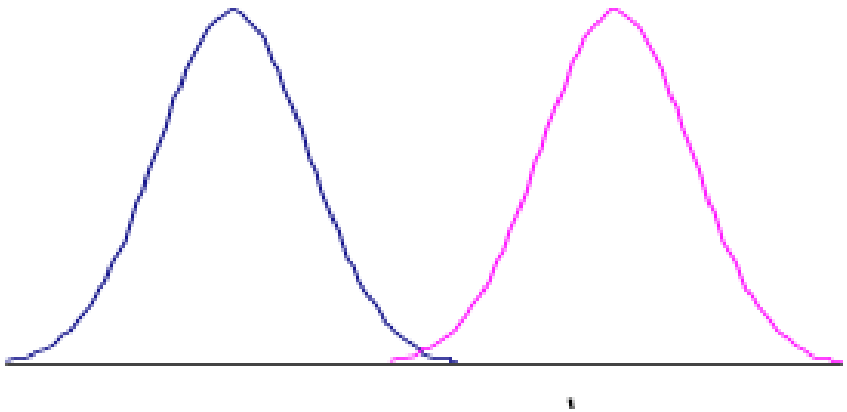
Poisson Distribution

- dpois(x = 4, lambda = 1.8) o/p: [1] 0.07230173
- ppois(): Cumulative poisson distribution. ppois(q = 0:1, lambda = 1.8) O/P: [1] 0.1652989 0.4628369

Continuous Probability Distribution

- The normal distribution is "probably" the most important distribution in Statistics
- It is a probability distribution of a continuous random variable and is often used to model the distribution of discrete random variable as well as the distribution of other continuous random variables
- The basic form of normal distribution is that of a bell, it has single mode and is symmetric about its central values
- The flexibility of using normal distribution is due to the fact that the curve may be centered over any number on the real line and it may be made flat or peaked to correspond to the amount of dispersion in the values of random variable
- The versatility in using the normal distribution as probability distribution model is depicted in Fig.

Normal Distribution



Normal Distribution

- Many quantitative characteristics have distribution similar in form to the normal distributions bell shape
- For example height and weight of people, the IQ of people, height of trees, length of leaves etc. are typically the type of measurements that produces a random variable that can be successfully approximated by normal random variable
- The values of random variables are produced by a measuring process and measurements tend to cluster symmetrically about a central value

Normal Distribution

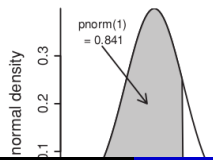
Definition

A random variate X, with mean μ and variance σ^2, is said to have normal distribution, if its probability density function is given by

$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{\frac{-(x-\mu)^2}{2\sigma^2}}$, $-\infty < x < \infty$, $-\infty < \mu < \infty$, $\sigma > 0$

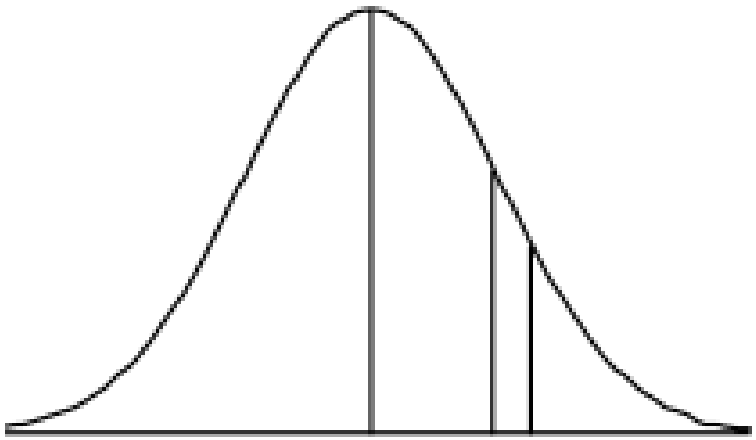
Definition (Standard Normal Distribution)

If X is a normal random variable with mean μ and standard deviation σ, then $\frac{(x-\mu)}{\sigma}$ is a standard normal variate with zero mean and standard deviation. The probability density function of standard normal variable Z is $f(z)=\frac{1}{\sqrt{2\pi}}e^{\frac{-x^2}{2}}$



Area under the Normal Curve

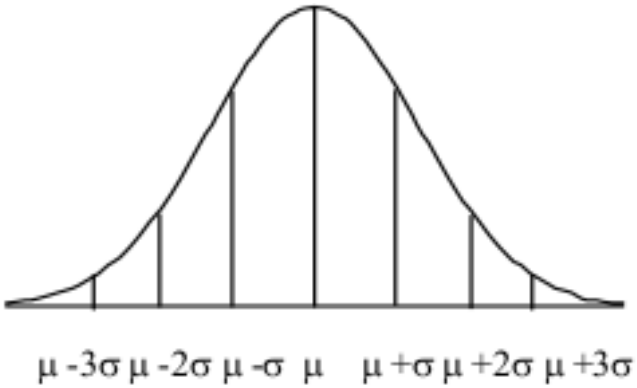
- For normal variable X, P(a $<$ X $<$ b) = Area under y = f (x) from X = a to X = b as shown in Fig



- The probability that X is between a and b (b $>$ a) can be determined by computing the probability that Z is between (a - μ) / σ and (b - μ) / σ

Area under the normal curve

- Let us consider the area under the normal curve



- The area under normal curve is distributed as follows:
  1. μ - σ and $\mu + \sigma$ covers 68.26% of area
  2. μ - 2σ and $\mu + 2\sigma$ covers 95.44% of area
  3. μ - 3σ and $\mu + 3\sigma$ covers 99.73% of area

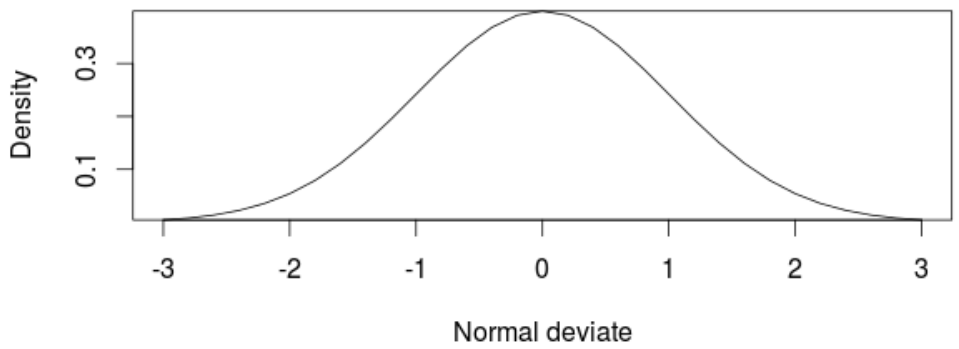
Properties of Normal Distribution

1. The normal curve is symmetrical about the mean x=μ
2. The height of normal curve is at its maximum at the mean. Hence the mean and mode of normal distribution coincides. Also the number of observations below the mean in a normal distribution is equal to the number of observations above the mean. Hence mean and median of normal distribution coincides. Thus for normal distribution mean = median = mode
3. The normal curve is uni modal at x =μ
4. The point of inflexion occurs at $\mu \pm \sigma$
5. The first and third quartiles are equidistant from the median

Plotting a Normal Distribution in R

```
## Plot the normal density, in the range -3 to 3
z <- pretty(c(-3,3), 30) # Find  30 equally spaced points
ht <- dnorm(z)
# By default: mean=0, standard deviation=1
plot(z, ht, type="l",xlab="Normal deviate",
ylab="Density",yaxs="i")
# yaxs="i" locates the axes at the limits of the data
```

Plotting Contd..



pnorm() and qnorm()

- pnorm():The function pnorm() calculates the cumulative probability, i.e., the area under the curve up to the specified ordinate or x-value
- Example:There is a probability of 0.841 that a normal deviate is less than 1- pnorm(1.0)
- Output:0.8413447
- qnorm():used to compute the normal quantiles. For example, the 90th percentile is 1.28
- Example: qnorm(.9) Output:1.281552

Simulation of random numbers and random samples

- In a simulation, repeated random samples are taken from a specified distribution
- Statistics, estimates that are derived from one or other model, can then be calculated for each successive sample
- Information is obtained on variation under repeated sampling
- This allows a check on results predicted by statistical theory
- Or it may provide guidance when theoretical results are not available or are of uncertain relevance
- Simulation of discrete and continuous random variables
- Random sampling from finite populations

Simulation of random numbers and random samples

- It is undesirable to use the same random number seed in two or more successive calls to a function that uses the random number generator
- However, users will sometimes, for purposes of checking a calculation, wish to repeat calculations with the same sequence of random numbers as was generated in an earlier call
- set.seed(): Sets the seed for the random number generator. The seed for the random number generator is stored in the workspace in a hidden variable (.Random.seed) that changes whenever there has been a call to the random number generator. This ensures that any new simulation will be independent of earlier simulations

Simulation of random numbers and random samples

- When the workspace is saved, .Random.seed is stored as part of the workspace
- This ensures that, when the workspace is loaded again, the seed will be restored to its value when the workspace was last saved
- Any new simulations will then be independent of those prior to the save
- In order to take advantage of this feature, be sure to save the workspace at the end of each session

Sampling from discrete distributions

- Values can be simulated from any of many different distributions
- Binomial, Poisson, and normal samples are simulated
- Example: Simulate a random sequence of 10 binary digits (0s or 1s) from a population with a specified proportion of 1s, here 50% (i.e., a Bernoulli distribution):

```
set.seed(23286)
 rbinom(10, size=1, p=.5)
 Output: [1] 0 0 0 0 1 0 1 0 1 1
```

- Example 2:To generate the numbers of daughters in a simulated sample of 25 four-child families, assuming that males and females are equally likely

```
rbinom(25, size=4, prob=0.5)
Output:[1] 4 1 1 2 3 3 4 3 1 2 4 0 2 2 3 2 4 1 1
2 2 2 3 2 3
```

Sampling from discrete distributions

- Simulate the number of raisins in 20 raisin buns, where the expected number of raisins per bun is 3:

```
set.seed(9388)
rpois(20, 3)
Output:[1] 3 2 4 2 2 3 5 2 4 3 0 2 3 3 7 3 4 4 1 3
```

Sampling from the normal and other continuous distributions

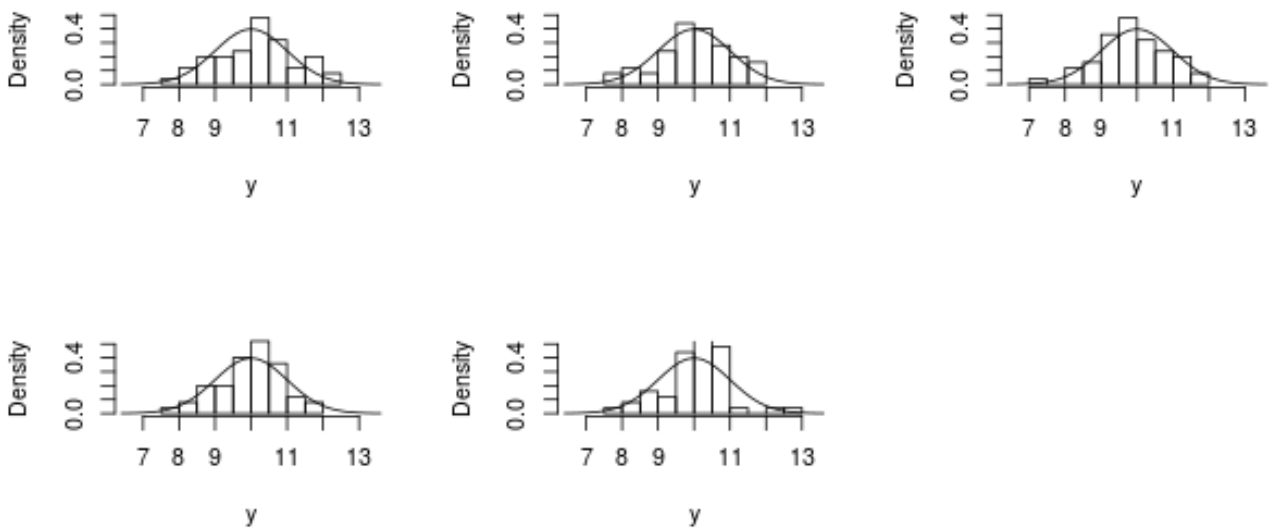
- The function rnorm() generates random deviates from the normal distribution
- Example: To generate 10 random values from a standard normal distribution

```
options(digits=2)
rnorm(10)
Output: [1]  2.11  0.65  0.55 -1.87  0.18  0.26
-1.44 -0.80  0.95 -0.34
```

Another Example

```
set.seed (21)
# Use to reproduce the data in the figure
par(mfrow=c(2,3))
x <- pretty(c(6.5,13.5), 40)
for(i in 1:5){
  y <- rnorm(50, mean=10, sd=1)
  hist(y, prob=TRUE, xlim=c(6.5,13.5), ylim=c(0,0.5),
   main="")
  lines(x, dnorm(x,10,1))
}
par(mfrow=c(1,1))
```

Output of the Previous Example



Uniform and Exponential Random Numbers

```
runif(n = 20, min=0, max=1)
Output:[1] 0.16 0.33 0.61 0.34 0.95 0.24 0.93 0.49 0.59
 0.38 0.21 0.85 0.53 0.46 0.95
[16] 0.47 0.12 0.72 0.97 0.53
rexp(n=10, rate=3)
Output:[1] 0.440 0.153 0.094 0.054 0.324 0.024 0.045
 0.311 0.576 0.149
```

Simulation of Regression Data

- A sample of n observations can be simulated from the regression model by using

$$y = b_0 + b_1 x + \epsilon \tag{3}$$

- where ϵ is normally distributed with standard deviation σ
- Let us assume n $= 8$, the intercept to be 2, the slope to be 3, and σ to be 2.5 in our simulation, and we use a fixed equally spaced design for the predictor values:

Example in R

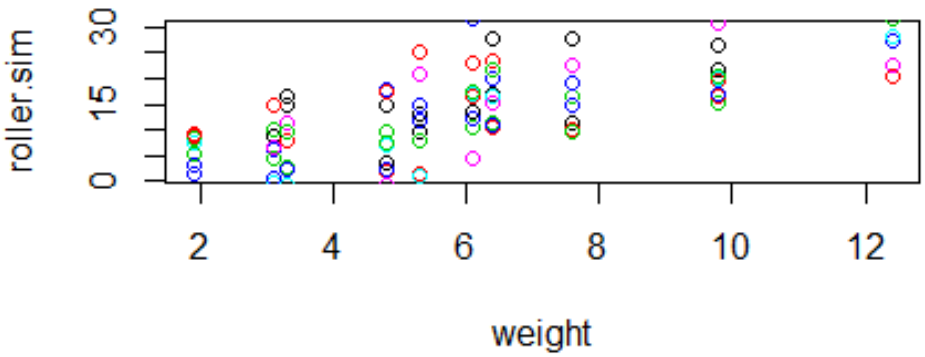
```
reg<-function(n){
options(digits=3)
  x<-seq(1,n)
  sigma<-2.5
  b0<-2
  b1<-3
  error<-rnorm(n,sd=sigma)
  y<-b0+b1*x+error
 return(t(data.frame(x,y)))
}
Function Call and Output:
> t<-reg(5)
> print(t)
  [,1] [,2] [,3] [,4] [,5]
```

simulate()

- simulate()→ Repeatedly simulate data from a fitted model, then re-fitting to each new set of simulated data
- Use: A check on variation under such repeated simulation
- Example: 10 simulations based on the model that was fitted to the roller data

```
library(DAAG)
attach(roller)
roller.lm <- lm(depression  weight, data=roller)
roller.sim <- simulate(roller.lm, nsim=10)
with(roller, matplot(weight, roller.sim, pch=1, ylim=range
                     (depression)))
```

Output



Simulation of the sampling distribution of the mean

- The sampling distribution of the mean is the distribution of the means of repeated random samples of size n
- The standard deviation of this sampling distribution has the name standard error of the mean (SEM)
- If the population mean is μ and the standard deviation is σ , then SEM $= \frac{s}{\sqrt{n}}$
- The Central Limit Theorem implies that, for large enough n, this sampling distribution will closely approx- imate the normal
- The sample size n needed so that the normal is a good approximation will depend on the distribution of the population from which samples are taken

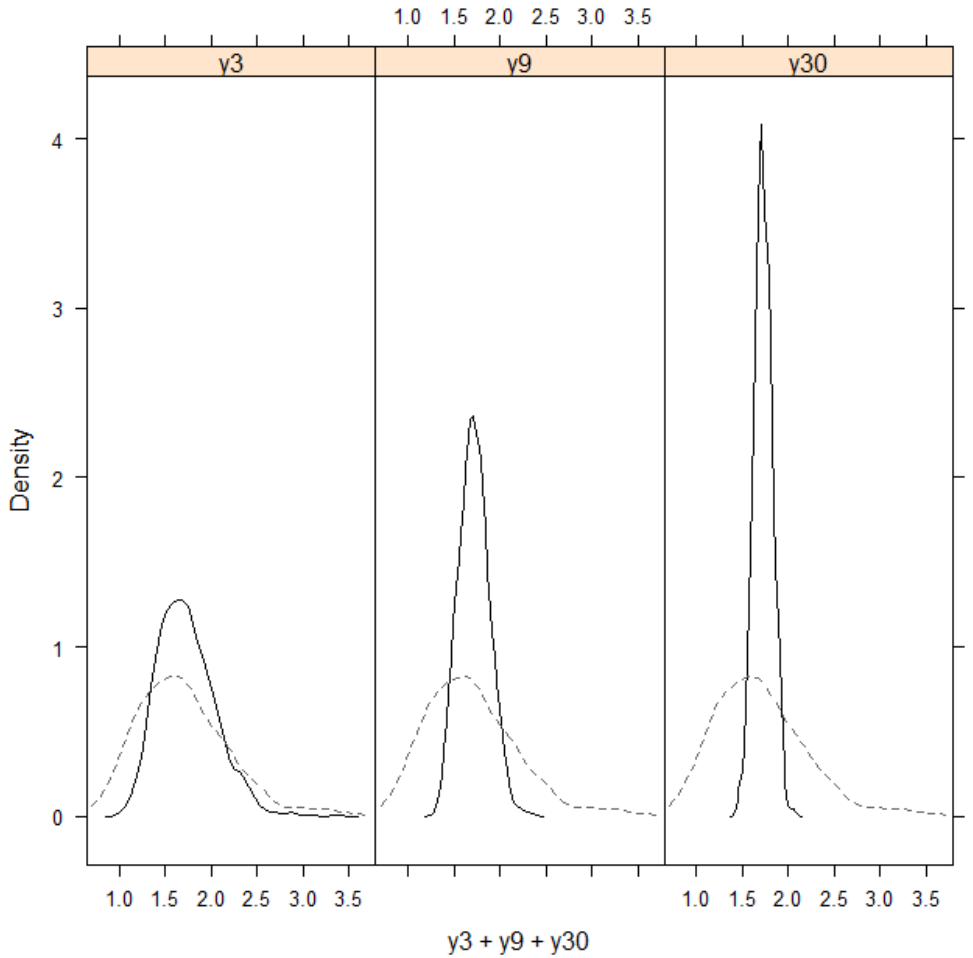
R Code

```
library(lattice)
## Function to generate n sample values; skew population
sampvals <- function(n)exp(rnorm(n, mean = 0.5, sd = 0.3))
## Means across rows of a dimension
 nsamp x sampsize matrix of
## sample values gives nsamp means of
samples of size sampsize.
samplingDist <- function(sampsize=3,
nsamp=1000, FUN=mean)
    apply(matrix(sampvals(sampsize*nsamp),
    ncol=sampsize), 1, FUN)
size <- c(3,10,30)
```

R code

```
## Simulate means of samples of 3, 9 and 30;
 place in dataframe
df <- data.frame(y3=samplingDist(sampsize=size[1]),
                 y9=samplingDist(sampsize=size[2]),
                 y30=samplingDist(sampsize=size[3]))
y <- samplingDist(sampsize=1)
densityplot(~y3+y9+y30, data = df, outer=TRUE,
 layout = c(3,1),
                  plot.points = FALSE,
                   panel = function(x, ...) {
                    panel.densityplot(x, ...,
                    col = "black")
                    panel.densityplot(y, col = "gray40",
                     lty = 2, ...)
                  })
```

Output of the Previous Program



Sampling from finite populations

- Sample()-Function to generate a simple random sample from a given set of numbers
- Example: Names on an electoral roll are numbered from 1 to 9384. To obtain a random sample of 15 individuals the R code is

```
> sample(1:9384, 15, replace=FALSE)
Output: [1] 2905  145 6638 8557  833 4933 1127
7879 3485  966  910 3597 6049 3920 3567
```

Sampling from finite populations

- Example:To randomly assign 10 plants (labeled from 1 to 10, inclusive) to one of two equal-sized groups, control and treatment

```
split(sample(seq(1:10)),
 rep(c("Control","Treatment"), 5))
Output:
$Control
[1] 6 5 9 8 2

$Treatment
[1]  4 10  3  7  1
sample(1:10, replace=TRUE)
 [1] 9 4 5 2 2 3 6 3 7 4
```

Example

1. Use y ¡- rnorm(100) to generate a random sample of size 100 from a normal distribution.
   1. Calculate the mean and standard deviation of y.
   2. Use a loop to repeat the above calculation 25 times. Store the 25 means in a vector named av. Calculate the standard deviation of the values in av.
2. To simulate samples from normal populations having different means and standard deviations, the mean and sd arguments can be used in rnorm(). Simulate a random sample of size 20 from a normal population having a mean of 100 and a standard deviation of 10

Example

3. The function pexp(x, rate=r) can be used to compute the probability that an exponential variable is less than x. Suppose the time between accidents at an intersection can be modeled by an exponential distribution with a rate of 0.05 per day. Find the probability that the next accident will occur during the next three weeks
4. Use the function rexp() to simulate 100 exponential random numbers with rate 0.2. Obtain a density plot for the observations. Find the sample mean of the observations. Compare with the population mean (the mean for an exponential population is 1/rate)

Common Model Assumptions

- Common model assumptions are normality, independence of the elements of the error term, and homogeneity of variance (i.e., the standard deviations of all measurements are the same)
- If certain assumptions fail to hold, a statistical method may be invalid
- Other assumptions may not be as important; we say that the method used is robust against those assumptions

Random sampling assumptions independence

- A data analyst has a sample of values that will be used as a window into a wider population
- Ideally, data should be gathered in such a way that the independence assumption is guaranteed
- Failure of the independence assumption is a common reason for wrong statistical inferences
- Detecting failure of the independence assumption is often difficult
- Tests for independence are at best an occasionally useful guide

Checks for normality

- Many data analysis methods rest on the assumption that the data are normally distributed
- Real data are unlikely to be exactly normally distributed
- Broadly, gross departures from normality are a cause for concern
- Small departures are of no consequence
- Check especially for data that are skew
- Check also for data that take a small number of discrete values, perhaps as a result of excessive rounding

The normal probability plot

- A better tool for assessing normality is the normal probability (or quantilequantile) plot
- The data values are sorted, then plotted against the ordered values that might be expected if the data really were from a normal distribution
- If the data are from a normal distribution, the plot should approximate a straight line
- Example: The normal probability plots for the same five sets of 50 normally distributed values

The Normal Probability Plot

- R Code: qreference(m=50, seed=21, nrep=5, nrows=1)
- Output:

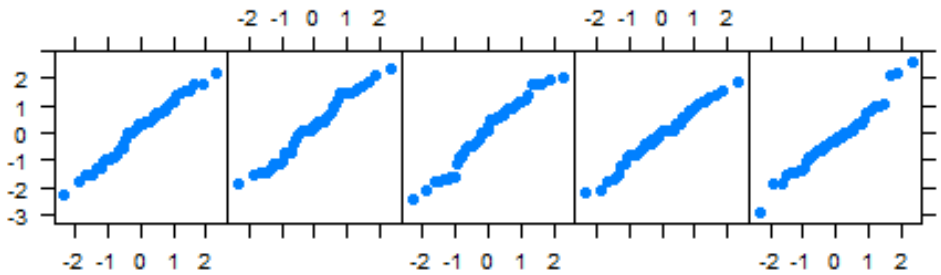


Figure: Normal probability plots for the same random normal data

The Normal Probability Plot

- The figure given in the previous slide help the data analyst to calibrate the eye, to get a feel for the nature and extent of departures from linearity that are to be expected in random normal samples of the specified size, here 50
- It is useful to repeat the process several times
- Such plots give a standard against which to compare the normal probability plot for the sample

Example

- Use the function rexp() to simulate 100 exponential random numbers with rate 0.2. Obtain a density plot for the observations. Find the sample mean of the observations. Compare with the population mean (the mean for an exponential population is 1/rate)
- The function pexp(x, rate=r) can be used to compute the probability that an exponential variable is less than x. Suppose the time between accidents at an intersection can be modeled by an exponential distribution with a rate of 0.05 per day. Find the probability that the next accident will occur during the next three weeks

Fitting a line to data

- How accurate is the line?
  - In the model $y_i = \alpha + \beta x_i + \epsilon_i$, the assumptions are that given x_i , the response y_i is from a normal distribution with mean $\alpha + \beta x_i$, and that the y_i are sampled independently
  - Equivalently, the ϵ_i are independently and identically distributed as normal variables with mean 0 and variance σ^2
  - With different assumptions (e.g., a sequential correlation between successive data points), the standard errors will be different

Summary information lawn roller example

- we use the R model formula depression weight to supply the model information to the function lm()
- Use summary() to display the output
- R Code:

```
library(DAAG)
roller.lm <- lm(depression~weight,data=roller)
print(summary(roller.lm))
```

Summary Contd..

```
Output:
Call:
lm(formula = depression ~ weight, data = roller)

Residuals:
   Min     1Q Median     3Q    Max
-8.180 -5.580 -1.346  5.920  8.020

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.0871     4.7543  -0.439  0.67227
weight        2.6667     0.7002   3.808  0.00518 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Summary Contd..

```
Residual standard error: 6.735 on 8 degrees of freedom
Multiple R-squared:  0.6445,Adjusted R-squared:  0.6001
F-statistic:  14.5 on 1 and 8 DF,  p-value: 0.005175
```

Summary Contd..

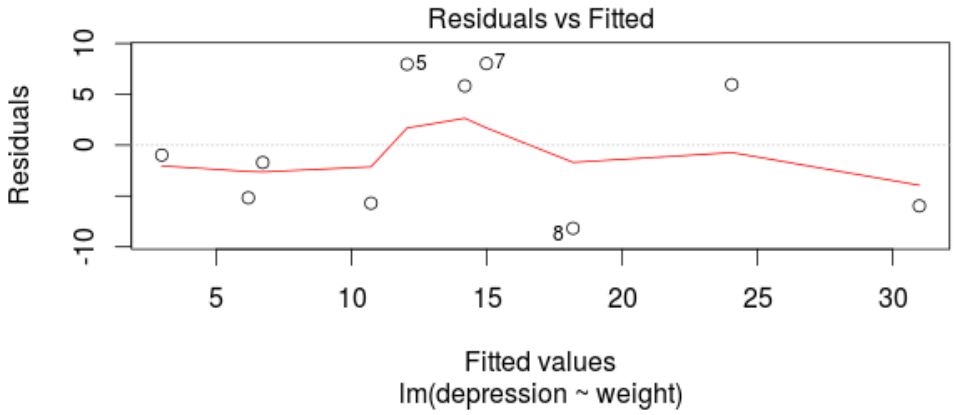
- The intercept of the fitted line is a $= 2.09$ (SE $= 4.75$), while the estimated slope is b $= 2.67$ (SE $= 0.70$)
- The p-value for the slope (testing the null hypothesis that β $=$ true slope $= 0$) is small, consistent with the evident linear trend
- The p-value for the intercept (testing $\alpha = 0$) is 0.67, i.e., the difference from zero may well be random sampling error
- Thus, consistently with the intuition that depression should be proportional to weight

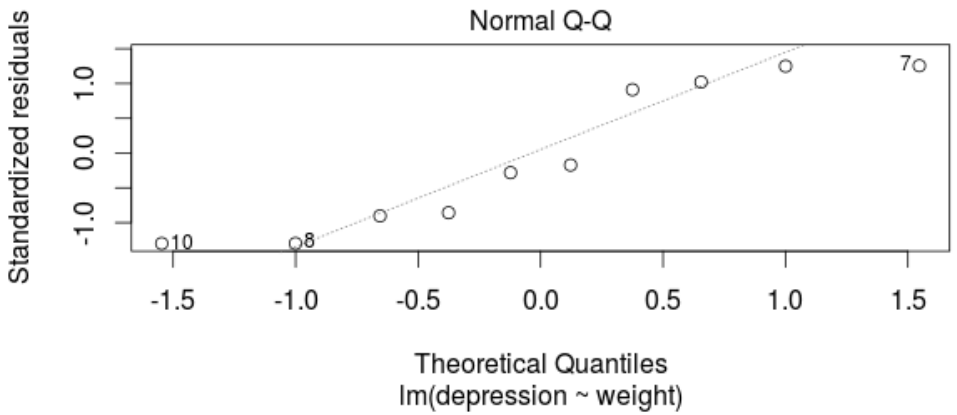
Residual Plots

- The residuals provide information about the noise term in the model, and allow limited checks on model assumptions
- However in such a small data set, departures from assumptions will be hard to detect
- Two common checks, both available by using the plot() function with an lm object, are:
  1. A plot of residuals versus fitted values. This allows a visual check for any pattern in the residuals that might suggest a curve rather than a line.
  2. A normal probability plot of residuals. If residuals are from a normal distribution points should lie, to within statistical error, close to a line

```
library(DAAG)
plot(roller.lm, which = 1:2)
```

Residual Plots Output





Residual Plots Output

- In Figure 1, there is a suggestion of clustering in the residuals, but no clear indication that there should be a curve rather than a line
- For interpreting the normal probability plot in Figure 2, the eye needs a reference standard. It is useful to compare a plot such as Figure 2, against a number of independent plots from computer-generated normal data with the same number of observations

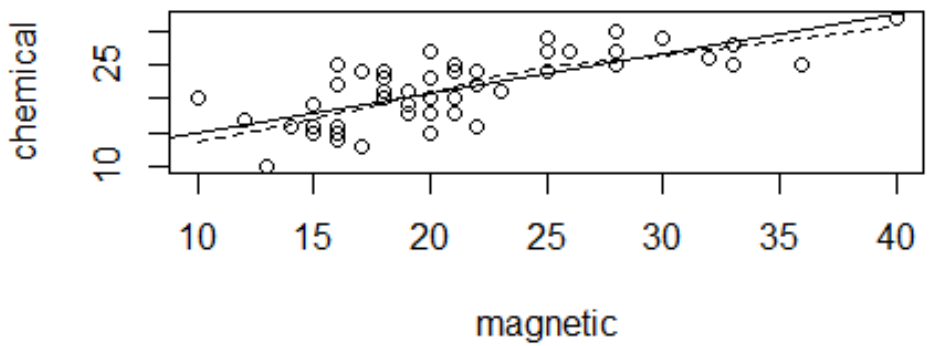
A pattern in the residuals

- Consider an example where there is an evident pattern in residuals from a straight line regression
- The data compare two methods for measuring the iron content in slag - a magnetic method and a chemical method
- The chemical method requires greater effort and is presumably expensive, while the magnetic method is quicker and easier

Pattern 1

```
library(DAAG)
plot(chemical~magnetic, data=ironslag)
ironslag.lm <- lm(chemical~magnetic, data=ironslag)
abline(ironslag.lm)
with(ironslag, lines(lowess(chemical~ magnetic, f=.9),
 lty=2))
```

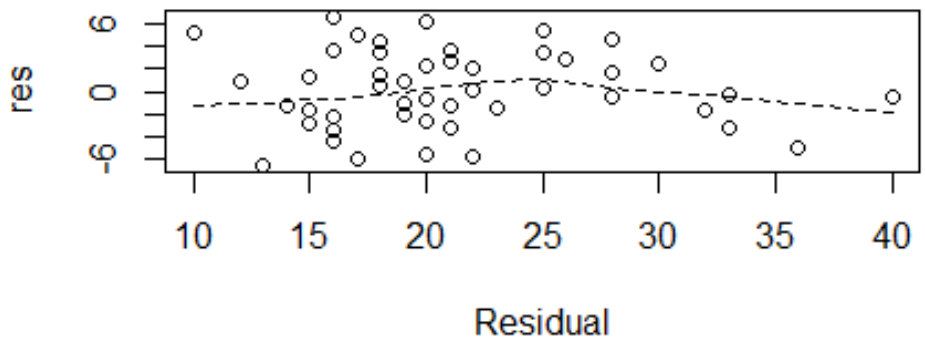
Pattern 1 - Output



- Pattern1 output suggests that the straight line model is wrong. The smooth curve (shown with a dashed line) gives a better indication of the pattern in the data

Pattern 2

```
res <- residuals(ironslag.lm)
plot(res~magnetic, xlab="Residual", data=ironslag)
with(ironslag, lines(lowess(res~magnetic, f=.9),
 lty=2))
```



- Output of Pattern2 shows the residuals from the straight line

Anova(One Way Analysis of Variance) Table

- Introduction
  - Analysis of Variance (ANOVA) is a hypothesis-testing technique used to test the equality of two or more population (or treatment) means by examining the variances of samples that are taken
  - ANOVA allows one to determine whether the differ ences between the samples are simply due to random error (sampling errors) or whether there are systematic treatment effects that causes the mean in one group to differ from the mean in another
  - Most of the time ANOVA is used to compare the equality of three or more means, however when the means from two samples are compared using ANOVA it is equivalent to using a t-test to compare the means of independent samples

Anova Contd..

- ANOVA is based on comparing the variance (or variation) between the data samples to variation within each particular sample
- If the between variation is much larger than the within variation, the means of different samples will not be equal
- If the between and within variations are approximately the same size, then there will be no significant difference between sample means
- Assumptions of Anova:
  1. All populations involved follow a normal distribution.
  2. All populations have the same variance (or standard deviation)
  3. The samples are randomly select ed and independent of one another

Anova

- Since ANOVA assumes the populations involve d follow a normal distribution, ANOVA falls into a category of hypothesis tests known as parametric tests
- If the populations involved did not follow a normal distribution, an ANOVA test could not be used to examine the equality of the sample means
- Instead, one would have to use a non-parametric test (or distribution-free test), which is a more general form of hypothesis testing that does not rely on distributional assumptions

Outliers, influence, and robust regression

- Some key terms in linear regression

Definition (Residual)

The difference between the predicted value (based on the regression equation) and the actual, observed value

Definition (Outlier)

In linear regression, an outlier is an observation with large residual. In other words, it is an observation whose dependent-variable value is unusual given its value on the predictor variables. An outlier may indicate a sample peculiarity or may indicate a data entry error or other problem

Outliers, influence, and robust regression

Definition (Leverage)

An observation with an extreme value on a predictor variable is a point with high leverage. Leverage is a measure of how far an independent variable deviates from its mean. High leverage points can have a great amount of effect on the estimate of regression coefficients

Definition (Influence)

An observation is said to be influential if removing the observation substantially changes the estimate of the regression coefficients. Influence can be thought of as the product of leverage and outlierness

Outliers, influence, and robust regression

Definition (Cook's Distance or Cook's D)

A measure that combines the information of leverage and residual of the observation. It measures the extent to which the line would change if the point were omitted

Definition (Robust Regression)

Robust regression is an alternative to least squares regression when data are contaminated with outliers or influential observations, and it can also be used for the purpose of detecting influential observations

Definition (Resistant Regression)

Methods to ensure that outliers do not contribute to the regression fit. For both robust and resistant methods, it is important that

Uses of Robust Regression

- Robust regression can be used in any situation in which you would use least squares regression
- When fitting a least squares regression, we might find some outliers or high leverage data points
- These data points are not data entry errors, neither they are from a different population than most of our data
- There no compelling reason to exclude them from the analysis
- Robust regression might be a good strategy since it is a compromise between excluding these points entirely from the analysis and including all the data points and treating all them equally in OLS regression
- The idea of robust regression is to weigh the observations differently based on how well behaved these observations are
- Roughly speaking, it is a form of weighted and re-weighted least squares regression

Packages required for Robust Regression

- The rlm command in the MASS package command implements several versions of robust regression

Example

For the data analysis below, we will use the crime dataset that appears in Statistical Methods for Social Sciences, Third Edition by Alan Agresti and Barbara Finlay (Prentice Hall, 1997). The variables are state id (sid), state name (state), violent crimes per 100,000 people (crime), murders per 1,000,000 (murder), the percent of the population living in metropolitan areas (pctmetro), the percent of the population that is white (pctwhite), percent of population with a high school education or above (pcths), percent of population living under poverty line (poverty), and percent of population that are single parents (single). It has 51 observations. We are going to use poverty and single to predict crime.

R Code for reading the data

Click Here to View the data

Using Robust Regression Analysis

- Let us begin by running an OLS regression and doing some diagnostics
- We will begin by running an OLS regression and looking at diagnostic plots examining residuals, fitted values, Cook's distance, and leverage-Click Here to View the graph
- From these plots, we can identify observations 9, 25, and 51 as possibly problematic to our model
- We can look at these observations to see which states they represent
- For results Click Here to View the results
- DC, Florida and Mississippi have either high leverage or large residuals
- We can display the observations that have relatively large values of Cook's D
- For the results Click Here to View the results

Using Robust Regression Analysis

- Probably we drop DC to begin with since it is not even a state
- Include it in the analysis just to show that it has large Cook's D and demonstrate how it will be handled by rlm
- Look at the residuals
- Generate a new variable called absr1, which is the absolute value of the residuals (because the sign of the residual doesn't matter)
- Print the ten observations with the highest absolute residual values
- Click Here to View the results

Computing Robust Regression Analysis

- Robust regression is done by iterated re-weighted least squares (IRLS)
- The command for running robust regression is rlm in the MASS package
- There are several weighting functions that can be used for IRLS
- Let us first use the Huber weights in this example
- Let us look at the final weights created by the IRLS process
- Click Here to View the r code and the results

Computing Robust Regression Analysis

- Roughly, as the absolute residual goes down, the weight goes up
- In other words, cases with a large residuals tend to be down-weighted
- This output shows us that the observation for Mississippi will be down-weighted the most
- Florida will also be substantially down-weighted
- All observations not shown above have a weight of 1
- In OLS regression, all cases have a weight of 1
- Hence, the more cases in the robust regression that have a weight close to one, the closer the results of the OLS and robust regressions
- let's run the same model, but using the bisquare weighting function
- Again, we can look at the weights

Computing Robust Regression Analysis

- Click Here to View the r code and the results
- We can see that the weight given to Mississippi is dramatically lower using the bisquare weighting function than the Huber weighting function and the parameter estimates from these two different weighting methods differ
- When comparing the results of a regular OLS regression and a robust regression, if the results are very different, we will most likely want to use the results from the robust regression
- Large differences suggest that the model parameters are being highly influenced by outliers
- Different functions have advantages and drawbacks
- Huber weights can have difficulties with severe outliers, and bisquare weights can have difficulties converging or may yield multiple solutions

Analysis of the results

- The results from the two analyses are fairly different, especially with respect to the coefficients of single and the constant (intercept)
- Normally we are not interested in the constant, if you had centered one or both of the predictor variables, the constant would be useful
- On the other hand, you will notice that poverty is not statistically significant in either analysis, whereas single is significant in both analyses

Exercise

- For each of the data sets elastic1 and elastic2, determine the regression of stretch on distance. In each case determine
  1. fitted values and standard errors of fitted values and
  2. the R^2 statistic. Compare the two sets of results. What is the key difference between the two sets of data?
- Use the robust regression function rlm() from the MASS package to fit lines to the data in elastic1 and elastic2. Compare the results with those from use of lm(). Compare regression coefficients, standard errors of coefficients, and plots of residuals against fitted values.

Standard Errors and Confidence Intervals

- Confidence Intervals and Tests for slope:
  - A 95% confidence interval for the regression slope is b $\pm$ $t_{.975}$ SE_b where $t_{.975}$ is the 97.5% point of the t-distribution with n-2 degrees of freedom, and SE_b is the standard error of b
  - Let us consider the calculation for the roller data
  - Click Here to View the r code and calculation
  - From the second row of the Coefficients table in the summary output, the slope estimate is 2.67, with SE = 0.70
  - The t-critical value for a 95% confidence interval on 10 - 2 = 8 degrees of freedom is $t_{.975}$ = 2.30
  - Therefore, the 95% confidence interval is: 2.67 $\pm$ 2.3 x 0.7 = (1.1, 4.3)

SEs and confidence intervals for predicted values

- There are two types of predictions: (i)prediction of points on the line, and (ii) prediction of a new data value
- The SE estimates of predictions for new data values take account both of uncertainty in the line and of the variation of individual points about the line
- Thus the SE for prediction of a new data value is larger than that for prediction of points on the line
- The column headed SE.OBS indicates the precision with which new observations can be predicted
- For determining SE.OBS, there are two sources of uncertainty: the standard error for the fitted value (estimated at 3.6 in row 1) and the noise standard error (estimated at 6.74) associated with a new observation
- Click Here to View the r code and calculation

Assessing the Predictive Accuracy

- Training/test sets and cross-validation

Definition (Training Set)

The data set used for developing the model is called the training set

Definition (Test Set)

The data set used for predictions is called a test set

Assessing the Predictive Accuracy

Definition (Cross Validation)

- Cross-validation extends the training/test set approach
- The data are divided into k sets (or folds), where k is typically in the range 3 to 10
- Each of the k sets becomes in turn the test set, with the remaining data forming the training set
- The predictive accuracy assessments from the k folds are combined to give a measure of the predictive performance of the model
- This may be done for several different measures of predictive performance

Cross Validation Example

- Let us consider a small data set with a three fold cross validation
- Data on floor area and sale price for 15 houses in a suburb of Canberra, in 1999
- Rows of data have been numbered from 1 to 15
- For demonstrating cross-validation, we use a random number sampling system to divide the data up into three equal groups
- Click Here to View the r code and the results
- Rerunning the calculations will of course lead to a different division into three groups

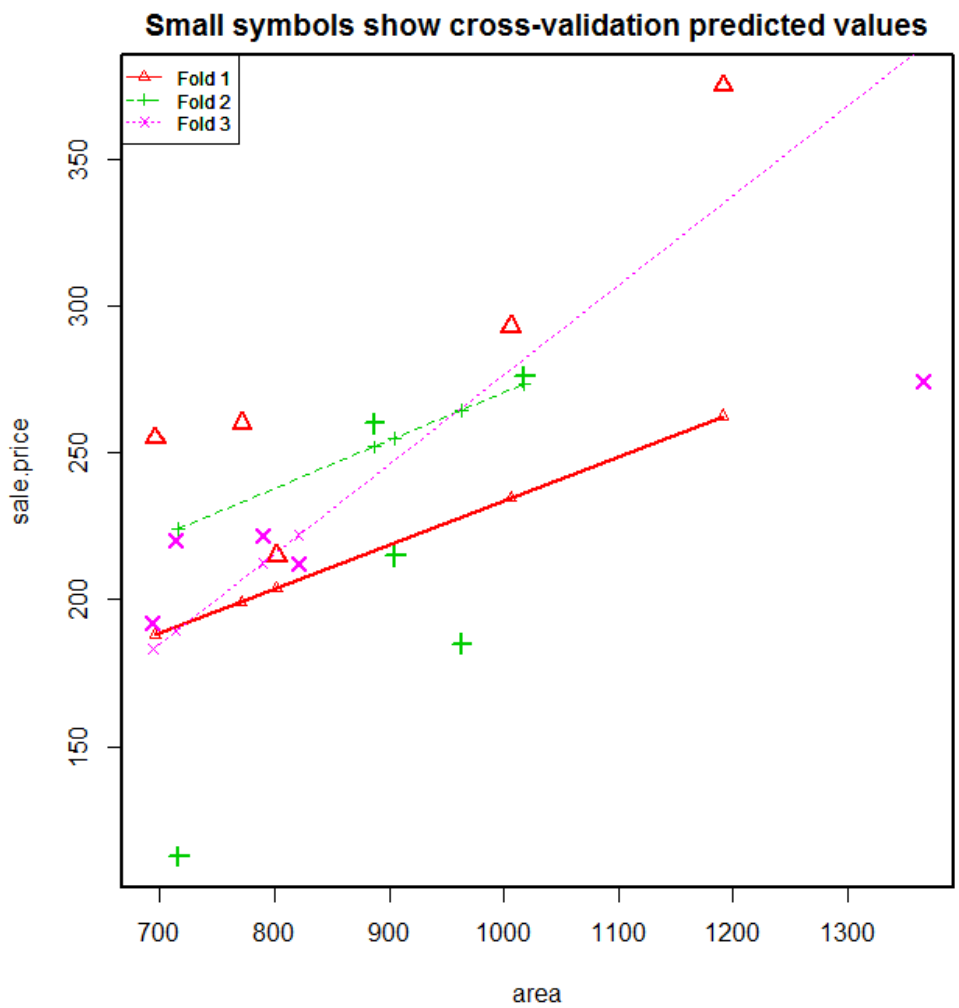
Cross Validation Example

- At the first pass (fold 1) the first set of rows will be set aside as the test data, with remaining rows making up the training data
- Each such division between training data and test data is known as a fold
- At the second pass (fold 2) the second set of rows will be set aside as the test data, while at the third pass (fold 3) the third set of rows will be set aside as the test data
- A crucial point is that at each pass the data that are used for testing are separate from the data used for prediction
- Click Here to View the r code and the results
- To obtain the estimate of the error mean square, take the total of the sums of squares and divide by 15. This gives $s^2 = (17719 + 15269 + 28127)/15 = 4074$

Cross Validation Example

- We have an estimate of the error mean square when we use only two-thirds of the data
- Thus we expect the cross-validated error to be larger than the error if all the data could be used
- We can reduce the error by doing 10-fold rather than threefold crossvalidation
- Contrast $s^2 = 4074$ with the estimate $s^2 = 2321$ that we obtained from the model-based estimate in the regression output for the total data

Cross Validation Example



Logarithmic and other transformations

- Logarithmic:
  - This is often the right transformation for size measurements (linear, surface, volume or weight) of biological organisms
  - Some data may be too skewed even for a logarithmic transformation
  - For example, counts of insects on leaves character
- Square root or Cube root:
  - These are milder than the logarithmic transformation
  - If linear measurements on insects are normally distributed, then we might expect the cube root of weight to be approximately normally distributed
  - The square root is useful for data for counts of "rare events"
  - Example: y^2, $y^{0.5}$, y^3
- Note: If the ratio of largest to smallest data value is greater than 10, and especially if it is more than 100, then the logarithmic transformation should be tried

General power transformations

- For $\lambda \neq 0$, the power transformation replaces a value y by y^{λ}
- The logarithmic transformation corresponds to λ=0
- To make this connection, a location and scale correction is needed
- The Transformation is:

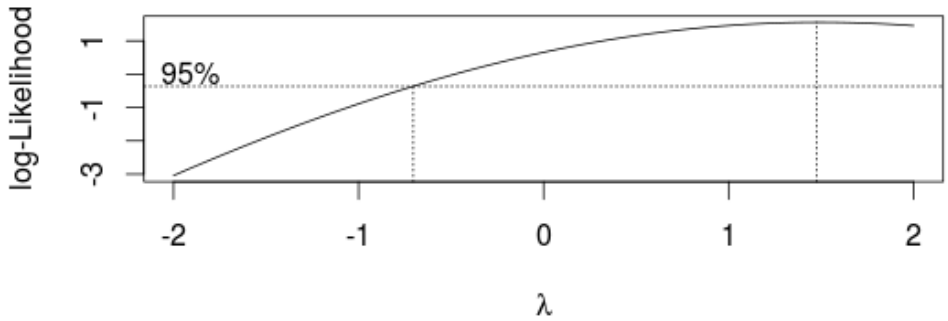
$$y(\lambda) = \frac{y^{\lambda} - 1}{\lambda} \; if \; \lambda \neq 0 \tag{4}$$

$$y(\lambda) = log(y) \; if \; \lambda = 0 \tag{5}$$

- If the small values of a variable need to be spread, make λ smaller and if the large values of a variable need to be spread, make λ larger
- This is called the BoxCox transformation
- The function boxcox() (MASS), whose syntax is similar to that of lm(), can be used to obtain data-driven estimates of λ

boxcox()

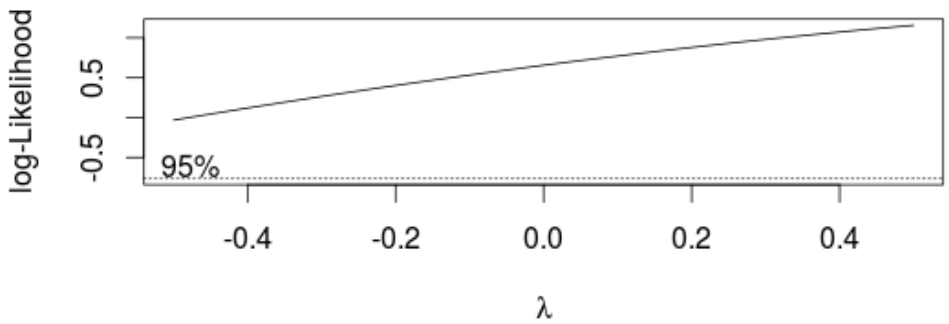
- The boxcox function is very easy to use: just specify the model formula, and the default options take care of everything else
- b<-boxcox(Min.passengers No.of.cars,data=Cars93.summary)
- output:



boxcox()

- To find the best value of lambda : which.max(b$y)
- Output: [1] 87
- To find the lambda value: (lambda¡-bc$x[which.max(bc$y)])
- Output:[1] 1.474747
- We can zoom in to get a more accurate estimate by specifying our own, non-default, range of lambda values
- Example:b<- boxcox(Min.passengers No.of.cars,data=Cars93.summary, lambda=seq(-0.5,0.5,0.01))

Output



Exercises

- Calculate volumes (volume) and page areas (area) for the books on which information is given in the data frame oddbooks (DAAG). (a)Plot log(weight) against log(volume), and fit a regression line. (b)Plot log(weight) against log(area), and again fit a regression line. (c)Which of the lines (a) and (b) gives the better fit?
- Use boxcox() for the data set pressure

Exercises

- Write a function which simulates simple linear regression data from the model y = 2 + 3x + e where the noise terms are independent normal random variables with mean 0 and variance 1. Using the function, simulate two samples of size 10. Consider two designs: first, assume that the x-values are independent uniform variates on the interval [1, 1]; second, assume that half of the x-values are 1s, and the remainder are 1s. In each case, compute slope estimates, standard error estimates, and estimates of the noise standard deviation.

Size and shape data allometric growth

- The logarithmic transformation is commonly important for morphometric data, i.e., for data on the size and shape of organisms
- The allometric growth equation:

$$\textit{The allometric growth equation is } y = ax^b \tag{6}$$

$$\log y = \log a + b \log x \tag{7}$$

$$Y = A + bX \tag{8}$$

$$\textit{where } Y = \log y,\ A = \log a,\ \textit{and } X = \log x \tag{9}$$

- Thus, we have an equation that can be fitted by linear regression methods, allowing prediction of values of Y given a value for X
- If b = 1, then the two organs (e.g., heart and body weight) grow at the same rate

Example

- R Code:

```
summary(cfseal.lm <- lm(log(heart)~log(weight),
data=cfseal))
Output:
Call:
lm(formula = log(heart) ~ log(weight), data = cfseal)

Residuals:
     Min       1Q   Median       3Q      Max
-0.31487 -0.09182  0.00002  0.11685  0.32051
```

Output

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.20434    0.21131   5.699 4.12e-06 ***
log(weight)  1.12615    0.05467  20.597  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.1796 on 28 degrees of freedom
Multiple R-squared:  0.9381,Adjusted R-squared:  0.9359
F-statistic: 424.3 on 1 and 28 DF,  p-value: < 2.2e-16
```

- The estimate of the exponent b (= 1.126) differs from 1.0 by 2.3 (= 0.126/0.0547) times its standard error
- Thus for these data, the relative rate of increase seems slightly greater for heart weight than for body weight

The model matrix in regression

- In straight line regression, the model or X matrix has two columns - a column of 1s and a column that holds values of the explanatory variable x
- The straight line model is

$$y = a + bx \tag{10}$$

- which we can write as

$$y = 1 * a + x * b \tag{11}$$

Model Matrix in regression

```
roller.lm<-lm(weight~depression,data=roller)
model.matrix(roller.lm)
Output:
(Intercept) depression
1             1           2
2             1           1
3             1           5
4             1           5
5             1          20
6             1          20
7             1          23
8             1          10
9             1          30
10            1          25
```

No Intercept Model

- $\hat{y}$=bx. In this case the model matrix has only a single column, containing the values of x

Bayesian Regression Estimation using MCMC package

- Markov Chain Monte Carlo (MCMC) simulation technique to generate successive parameter estimates
- The simulation process must be allowed to burn in, i.e., run for long enough that the posterior distribution reaches a steady state that is independent of the starting values of parameters
- The MCMCpack package has the function MCMCregress(), with a similar syntax to lm(), that can be used for regression calculations
- The following is intended as a straightforward demonstration of the methodology, albeit for an example where use of the function lm() might in practice be preferable
- The default is to assume independent uniform priors for the regression coefficients, to allow the simulation to run for 10000 iterations, and to take the first 1000 iterations as burn-in

Example R Code

```
library(MCMCpack)
> roller.mcmc <- MCMCregress(depression~weight,data=roller)
> summary(roller.mcmc)
```

Output

```
Iterations = 1001:11000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 10000

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

                Mean      SD Naive SE Time-series SE
(Intercept) -1.999  5.4859 0.054859       0.053161
weight       2.653  0.8123 0.008123       0.007652
sigma2      60.468 40.6103 0.406103       0.522642
```

Output

```
2. Quantiles for each variable:

                2.5%    25%    50%    75%   97.5%
(Intercept) -12.796 -5.378 -1.994  1.291   9.249
weight        1.009  2.167  2.658  3.156   4.257
sigma2       21.011 35.404 49.390 71.418 166.234
```