Statistical Data Analysis

N. Sairam

sairam@cse.sastra.edu

School of Computing, SASTRA University, Thanjavur.

Unit-III

Multiple Linear Regression

- In straight line regression, a response variable y is regressed on a single explanatory variable x
- Multiple linear regression generalizes this methodology to allow multiple explanatory or predictor variables The
- Accurate Prediction is our focus

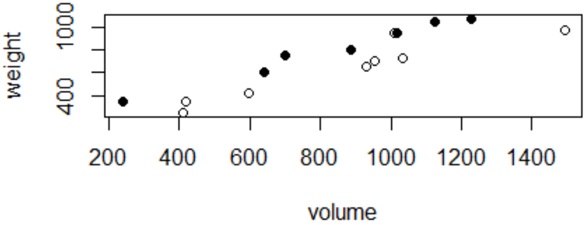
Basic Idea: Example

- Let us consider the book weight example that has two x-variables in the regression equation
- Explanatory variables are the volume of the book ignoring the covers, and the total area of the front and back covers
- weight of book $= b_0 + b_1$ x volume $+ b_2$ x area of covers

R Code

```
lot(weight   volume, data=allbacks, pch=c(16,1)
[unclass(cover)])
# unclass(cover) gives the integer codes that
 identify levels
with(allbacks, text(weight   volume, labels=paste(1:15),
pos=c(2,4)[unclass(cover)]))
```

Graph



Summary of the Regression Model

```
summary(allbacks.lm <- lm(weight~volume+area, data=allbacks
Output:
Call:
lm(formula = weight ~ volume + area,
data = allbacks)

Residuals:
    Min       1Q  Median       3Q      Max
-104.06  -30.02  -15.46   16.76   212.30

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 22.41342   58.40247   0.384 0.707858
volume       0.70821    0.06107  11.597 7.07e-08 ***
area         0.46843    0.10195   4.595 0.000616 ***
---
```

Summary of the Regression Model

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 77.66 on 12 degrees of freedom
Multiple R-squared:  0.9285,Adjusted R-squared:  0.9166
F-statistic: 77.89 on 2 and 12 DF,  p-value: 1.339e-07
```

Analysis of the Results

- The coefficient estimates are $b_0 = 22.4$, $b_1 = 0.708$, and $b_2 = 0.468$
- Standard errors and p-values are provided for each estimate
- The p-value for the intercept suggests that it cannot be distinguished from 0
- The p-value for volume tests $b_1 = 0$, in the equation that has both volume and area as explanatory variables
- The estimate of the noise standard deviation (the residual standard error) is 77.7
- There are now 15-3 = 12 degrees of freedom for the residual

Analysis of the Results

- The null hypothesis for this test is that all coefficients (other than the intercept) are 0
- Here, we reject this hypothesis and conclude that the equation does have explanatory power
- Confidence Interval for the volume: 0.708±qt(0.975, 12)*0.0611
- Output: 0.708±2.178813*0.0611=0.575 to 0.841
- anova(allbacks.lm)
- model.matrix(allbacks.lm)

Analysis of Anova Table

- This table gives the contribution of volume after fitting the overall mean, then the contribution of area after fitting both the overall mean and volume
- The p-value for area in the anova table must agree with that in the main regression output, since both these p-values test the contribution of area after including volume in the model
- The p-values for volume will differ if there is a correlation between volume and area
- Command to compute correlation:with(allbacks, cor(volume,area))
- Here, the correlation of volume with area is 0.0015

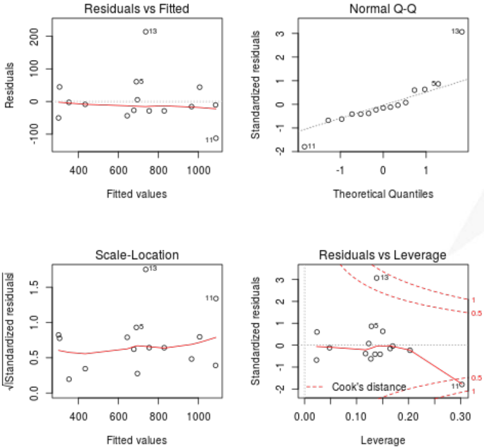
Analysis of model.matrix() Results

- Predicted values are given by multiplying the first column by b_0 (=22.4), the second by b_1 (=0.708), the third by b_2 (=0.468), and adding
- Omission of the Intercept Term:
  - allbacks.lm0 <- lm(weight -1+volume+area, data=allbacks)
  - summary(allbacks.lm0)
  - The regression coefficients now have smaller standard errors
  - The reason is that, in the model that included the intercept, there was a substantial negative correlation between the estimate of the intercept and the coefficient estimates
  - The reduction in standard error is greater for the coefficient of volume, where the correlation was -0.88, than for area, where the correlation was -0.32. Correlations between estimates can be obtained by setting corr=TRUE in the call to summary()

Diagnostic Plots

- Let us consider the following code:

```
par(mfrow=c(2,2));plot(allbacks.lm0);
dev.copy(png,'31.png');dev.off()
```



Explanation of the Diagnostic Plots

- The residual for observation 13 is large
- The observation 13 lies outside the 0.5 contour of Cook's distance, well out towards the contour for a Cook's distance of 1.
- It is a (somewhat) influential point

What happens if we omit observation 13?

- R code: allbacks.lm13 <- lm(weight -1+volume+area, data=allbacks[-13,])
- summary(allbacks.lm13)

Ouput of the above code

```
Call:
lm(formula = weight ~ -1 + volume + area,
 data = allbacks[-13, ])

Residuals:
    Min      1Q  Median      3Q     Max
-61.721 -25.291   3.429  31.244  58.856

Coefficients:
       Estimate Std. Error t value Pr(>|t|)
volume  0.69485    0.01629   42.65 1.79e-14 ***
area    0.55390    0.05269   10.51 2.08e-07 ***
---
```

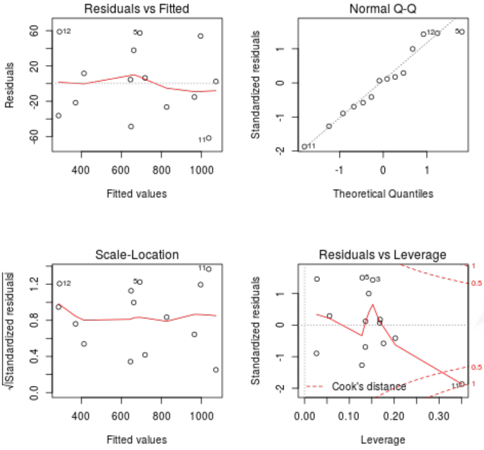
Output of the above code

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 41.02 on 12 degrees of freedom
Multiple R-squared:  0.9973,Adjusted R-squared:  0.9969
F-statistic:  2252 on 2 and 12 DF,  p-value: 3.521e-16
```

- The residual standard error is substantially smaller (41 instead of 75.1) in the absence of observation 13
- Observation 11 now has a Cooks distance that is close to 1, but does not stand out in the plot of residuals

Results


Residuals vs Fitted
Residuals
60
20
-20
-60
400
600
800
1000
Fitted values
12
5
11
Normal Q-Q
Standardized residuals
1
0
-1
-2
-1
0
1
Theoretical Quantiles
12
5
11
Scale-Location
√|Standardized residuals|
1.2
0.8
0.4
0.0
400
600
800
1000
Fitted values
12
5
11
Residuals vs Leverage
Standardized residuals
1
0
-1
-2
0.00
0.10
0.20
0.30
Leverage
Cook's distance
1
0.5
0.5
1
5
3
11

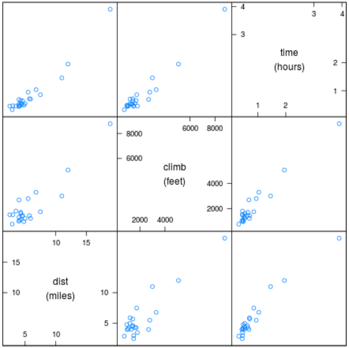
Interpretation of Model Coefficients

- To Understand the interpretation of model coefficients then it is important to fit a model whose coefficients are open to the relevant interpretations
- Different formulations of the regression model, or different models, may serve different explanatory purposes
- Predictive accuracy is in any case a consideration, and is often the main interest

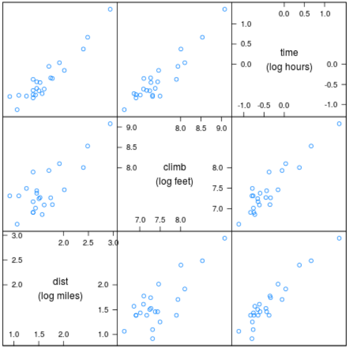
Example 1

- Let us consider the data set nihills (DAAG), that gives the distances (dist), heights climbed (climb), male record times (time), and female record times (timef), for Northern Irish hill races
- Let us begin with scatter plot matrices, both for the untransformed data and for the log transformed data
- Let us limit our attention to Male
- The diagonal panels give the x-variable names for all plots in the column above or below, and the y-variable names for all plots in the row to the left or right
- Note that the vertical axis labels alternate between the axis on the extreme left and the axis on the extreme right, and similarly for the horizontal axis labels

Scatter Plot



Scatter Plot Matrix



Scatter Plot Matrix

Investigation of Taking Logarithms

- The range of values of time is large (3.9:0.32, i.e., $>$10:1), and similarly for dist and climb. The times are bunched up towards zero, with a long tail. In such instances, use of a logarithmic transformation is likely to lead to a more symmetric distribution
- One point in particular has a time that is more than twice that of the next largest time. The values of dist and climb similarly stand out as much larger than for other points. In a regression that uses the untransformed variables, this point will have a much greater say in determining the regression equation than any other point. In the terminology, it has large leverage. Even after taking logarithms, its leverage remains large, but not quite so dominating

Investigation of Taking Logarithms

- It can be expected that time will increase more than linearly at very long times, and similarly for climb, as physiological demands on the human athlete move closer to limits of human endurance
- Such relationship as is evident between the explanatory variables (dist and climb) is more nearly linear on the logarithmic scale
- Additionally, use of a logarithmic scale may help stabilize the variance

Fitting the Equation

$$log(time) = a + b_1 log(dist) + b_2 log(climb) \quad (1)$$

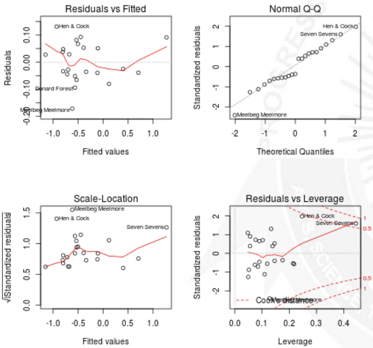
- Equivalent to Power Relationship

$$time = A(dist)^{b_1}(climb)^{b_2} \quad (2)$$

- where a = log(A)

R Code for fitting the model

```
nihills.lm<-lm(log(time)
~log(dist)
+log(climb),data=nihills)
plot(nihills.lm)
```



Summary

```
summary(nihills.lm)$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) -4.9611313 0.27387193 -18.11479 7.085048e-14
log(dist)    0.6813596 0.05517831  12.34832 8.186381e-11
log(climb)   0.4657575 0.04530181  10.28121 1.980592e-09
```

Interpreting the coefficients

- The estimated equation is log(time) = -4.96+0.68 x log(dist)+0.47x log(climb)
- Exponentiating both sides of this equation, and noting exp(-4.96) = 0.0070, gives time = 0.00070 x dist$^{0.68}$ x climb$^{0.47}$
- This equation implies that for a given height of climb, the time taken is smaller for the second three miles than for the first three miles

A meaningful coefficient for logdist

- The coefficient for logdist will be more meaningful if we regress on logdist and log(climb/dist)
- R Code:

```
>lognihills <- log(nihills)
> names(lognihills) <- paste("log", names(nihills),
 sep="")
> lognihills$logGrad <- with(nihills, log(climb/dist))
> nihillsG.lm <- lm(logtime   logdist + logGrad,
 data=lognihills)
>nihillsG.lm <- lm(logtime˜ logdist + logGrad,
 data=lognihills)
> summary(nihillsG.lm)$coef
```

Output for the R Code

```
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) -4.9611313 0.27387193 -18.11479 7.085048e-14
logdist      1.1471170 0.03459867  33.15494 5.896354e-19
logGrad      0.4657575 0.04530181  10.28121 1.980592e-09
```

Analysis of the above code

- The coefficient of logdist is now, greater than 1

  ```
  cor(lognihills$logdist,lognihills$logGrad)
  [1] -0.06529222
  cor(lognihills$logdist,lognihills$logclimb)
  [1] 0.780067
  ```

- The correlation between logdist and logGradient is 0.065, negligible relative to the correlation of 0.78 between logdist and logclimb

```
nihills.lm<-lm(logtime~logdist,data=lognihills)
> summary(nihills.lm)
```

Analysis of the above code

```
 summary(nihills.lm)$coeff
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) -2.210125 0.14284610 -15.47207 5.910392e-13
logdist      1.123892 0.08446864  13.30543 1.060019e-11
```

- Because the correlation between logdist and logGradient is so small, the coefficient of logdist (=1.124) in the regression on logdist alone is almost identical to the coefficient of logdist (=1.147) in the regression on logdist and logGradient
- The standard error of the coefficient of logdist is smaller - 0.035 as against 0.045 - when the second explanatory variable is logGradient rather than logclimb
- Note that the predicted values do not change
- The models nihills.lm nihillsG.lm are different mathematical formulations of the same underlying model

Scatter Plot Matrix

```
library(lattice); library(DAAG)
splom(  nihills[, c("dist","climb","time")],
 cex.labels=1.2,
varnames=c("dist\n(miles)","climb\n(feet)",
 "time\n(hours)"))
## Panel B: log transformed data
splom(  log(nihills[, c("dist","climb","time")]),
 cex.labels=1.2,
varnames=c("dist\n(log miles)", "climb\n(log feet)",
"time\n(log hours)"))
```

Exercise

- In the data set cement (MASS package), examine the dependence of y (amount of heat produced) on x1, x2, x3 and x4 (which are proportions of four constituents). Begin by examining the scatterplot matrix. As the explanatory variables are proportions, do they require transformation, perhaps by taking log(x/(100 x))?

Plots that show the contribution of individual terms

- For simplicity, the discussion will assume just two explanatory variables, x_1 and x_2, with the intention of showing the contribution of each in turn to the model
- The fitting of a regression model makes it possible to write:

$$
\begin{aligned}
y &= b_0 + b_1 x_1 + b_2 x_2 + e && (3)\\
&= \hat{y} + e && (4)
\end{aligned}
$$

- Another way to write the model that is to be fitted is:

$$y - \bar{y} = a + b_1(x_1 - \bar{x_1}) + b_2(x_2 - \bar{x_2}) + e \qquad (5)$$

Plots that show the contribution of individual terms

- For fitting the model in this form:
  - The observations are y-$\bar{y}$, with mean zero
  - The first explanatory variable is x_1-$\bar{x_1}$, with mean zero, and the first term in the model is $b_1(x_1 - \bar{x_1})$, with mean zero
  - The second explanatory variable is x_2-$\bar{x_2}$, with mean zero, and the first term in the model is) $b_2(x_2 - \bar{x_2})$, with mean zero
- The residuals e are exactly the same as before, and have mean zero
- The fitted model can then be written:

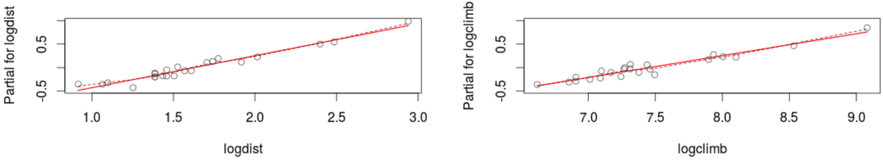
$$
\begin{aligned}
y = \bar{y} + b_1(x_1 - \bar{x_1}) + b_2(x_2 - \bar{x_2}) + e \qquad (6) \\
= \bar{y} + t_1 + t_2 + e \qquad (7)
\end{aligned}
$$

- Splits the response value y into three parts - an overall mean $\bar{y}$, a term that is due to x_1, a term that is due to x_2, and a residual e
- Moreover, the values of t_1 and t_2 sum,in each case, to zero

- The predict() function has an option (type="terms") that gives t_1 and t_2
- yterms <- predict(nihills.lm, type="terms")
- The first column of yterms has the values of $t_1 = b_1$ $(x_1$ - $\bar{x_1})$, while the second has the values of t_2
- Values in both these columns sum to zero

Partial Residual Plot

- The solid lines of the component plus residual plot in the figure given below show the contributions of the individual terms to the model
- The solid line in the left panel shows a plot of b_1 $(x_1 - \bar{x}_1)$ against x_1 , while the solid line in the right panel shows a plot of b_2 $(x_2 - \bar{x}_2)$ against x_2

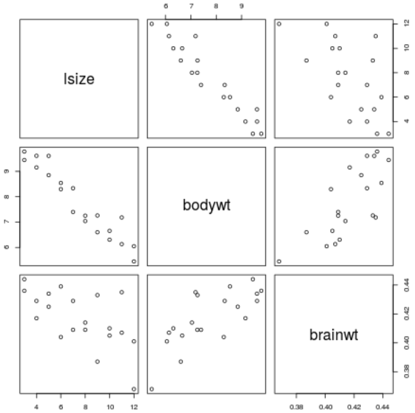


Analysis of the graph

- The lines can be obtained directly with the termplot() command
- The plotted points are the partial residuals, for the respective term
- The vector $t_1+e = \hat{y}-t_2$ holds the partial residuals for x_1 given x_2, i.e., they account for that part of the response that is not explained by the term in x_2
- The vector t_2+e holds the partial residuals for x_2 given x_1

Mouse Brain Weight Example

- The litters data frame (DAAG library) has observations on brain weight, body weight, and litter size of 20 mice



Mouse Brain Weight Example

- The explanatory variables lsize and bodywt are strongly correlated(From the graph)
- Regression of brainwt on lsize: summary(lm(brainwt lsize,data = litters))$coef

```
                Estimate   Std. Error   t value    Pr(>|t|)
(Intercept) 0.447000000 0.009624762 46.442707 3.391193e-20
lsize      -0.004033333 0.001198423 -3.365534 3.444524e-03
```

Mouse Brain Weight Example

- Regression of brainwt on lsize and bodywt

```
summary(lm(brainwt~lsize+bodywt,data = litters))$coef
               Estimate  Std. Error  t value    Pr(>|t|)
(Intercept) 0.178246962 0.075322590 2.366448 0.030097278
lsize       0.006690331 0.003132075 2.136070 0.047513226
bodywt      0.024306344 0.006778653 3.585719 0.002278441
```

Interpretation of the results

- In the first regression, variation in brainwt is being explained only with lsize, regardless of bodywt
- No adjustment has been made for the fact that bodywt increases as lsize decreases: individuals having small values of lsize have brainwt values corresponding to large values of bodywt, while individuals with large values of lsize have brainwt values corresponding to low bodywt values
- In the multiple regression, the coefficient for lsize is a measure of the change in brainwt with lsize, when bodywt is held constant
- For any particular value of bodywt, brainwt increases with lsize

A strategy for fitting multiple regression models

- Careful graphical scrutiny of the explanatory variables is an essential first step
- This may lead to any or all of:
  - Transformation of some or all variables.
  - Replacement of existing variables by newly constructed variables that are a better summary of the information. For example, we might want to replace variables x_1 and x_2 by the new variables $x_1 + x_2$ and x_1-x_2.
  - Omission of some variables.

Why are linear relationships between explanatory variables preferable?

- The following are reasons for restricting attention to transformations, where available, that lead to scatterplots in which relationships between explanatory variables are approximately linear
  1. If relationships between explanatory variables are non-linear, diagnostic plots may be misleading
  2. Approximately linear relationships ensure that all explanatory variables have similar distributions, preferably distributions that are not asymmetric to an extent that gives the smallest or largest points undue leverage
  3. If relationships are linear, it is useful to check the plots of explanatory variables against the response variable for indications of the relationship with the dependent variable

Note

1. Often, logarithmic or other standard forms of transformation give more symmetric distributions, lead to scatterplots where the relationships appear more nearly linear, and make it straightforward to identify a regression equation that has good predictive power

Steps to be followed to fit a Multiple Regression Model

1. Examine the distribution of each of the explanatory variables, and of the dependent variable. Look for any instances where distributions are highly skew, or where there are outlying values
2. Examine the scatterplot matrix involving all the explanatory variables. Look first for evidence of non-linearity in the plots of explanatory variables against each other. Look for values that appear as outliers in any of the pairwise scatterplots
3. Note the ranges of each of the explanatory variables
4. Find out the accuracy of each of the explanatory variables measured
5. If some pairwise plots show evidence of non-linearity, consider use of transformation(s) to give more nearly linear relationships
6. Where the distribution is skew, consider transformations that may lead to a more symmetric distribution

Steps to be followed to fit a Multiple Regression Model

7. Look for pairs of explanatory variables that are so highly correlated that they appear to give the same information

- Checks should include:
    - Plot residuals against the fitted values. Check the patterns of the residuals and for the fanning in or out residuals as the fitted values change
    - Examine the Cook's distance statistics. If it is helpful, examine the standardized versions of the drop-1 coefficients directly using dfbetas(). If it is necessary delete the influential data points and refit the model
    - For each explanatory variable, construct a component plus residual plot, to check whether any of the explanatory variables require transformation

Example

- Database:hills2000 (Scotland)

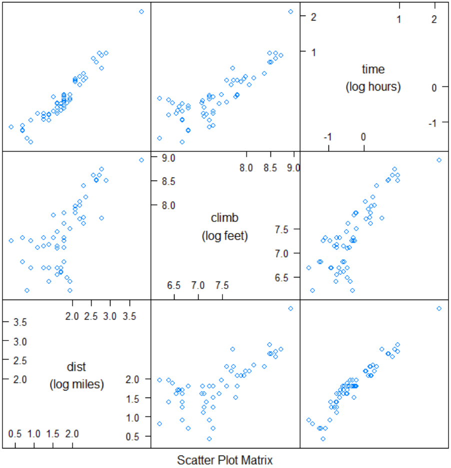
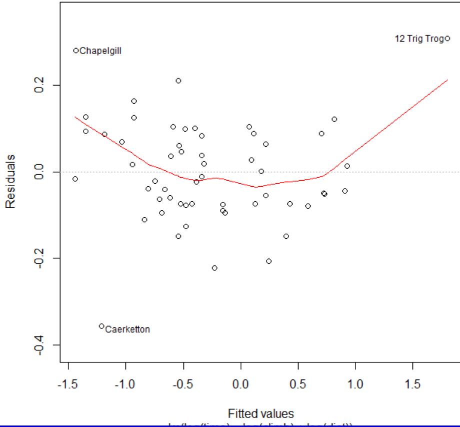


Figure: Scatter Plot for hills2000 data frame

Linear Modeling for the hills2000 data

- We will include Caerketton during our initial analysis from races2000 data

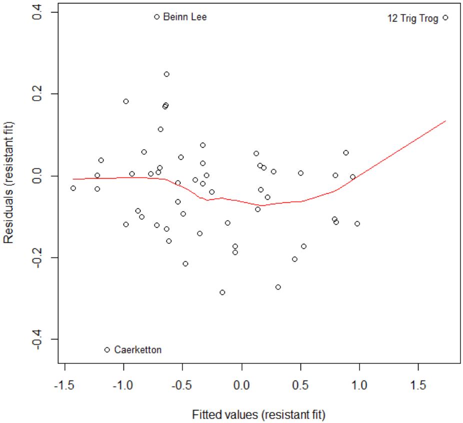
```
lhills2k.lm <- lm(log(time)
  log(climb) + log(dist),
data = hills2000)
plot(lhills2k.lm,
 caption="", which=1)
```



Linear Modeling for the hills2000 data

```
library(MASS)# lqs() is in the MASS package
lhills2k.lqs <- lqs(log(time)
~log(climb) + log(dist),data = hills2000)
reres <- residuals(lhills2k.lqs)
refit <- fitted(lhills2k.lqs)
big3 <- which(abs(reres) >= sort(abs(reres),
 decreasing=TRUE)[3])
plot(reres~refit, xlab="Fitted values (resistant fit)",
     ylab="Residuals (resistant fit)")
lines(lowess(reres  refit), col=2)
text(reres[big3] ~ refit[big3],
 labels=rownames(hills2000)[big3],
     pos=4-2*(refit[big3] > mean(refit)), cex=0.8)
```

Linear Modeling for the hills2000 data



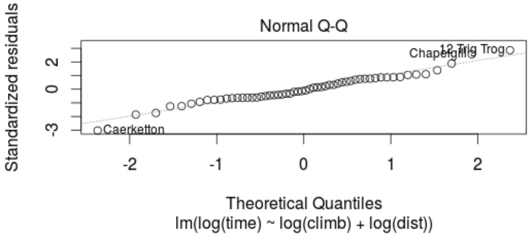
Linear Modeling for the hills2000 data Contd..

- The figures in the previous two slides shows residuals from a least squares (lm) fit and from a resistant lqs fit, in both cases plotted against fitted values
- Resistant fits completely ignore the effects of large residuals
- By default, even if almost half the observations are outliers, the effect on the fitted model would be small

Linear Modeling for the hills2000 data Contd..

- Caerketton shows up rather clearly as an outlier, in both graphs
- Its residual in graph1 is -0.356(i.e 0.35)
- The predicted log(time) is denoted as log($\hat{time}$)
- Then log(time)-log($\hat{time}$)=-0.356
- Thus log($\frac{time}{\hat{time}}$)=-0.356 $\implies$ $\frac{time}{\hat{time}}$=0.7
- Thus the time given for this race is 70% of that predicted by the regression equation, a very large difference
- The standardized difference is -3
- This can be viewed by plot(lhills2k.lm, which=2)

Viewing the standardized difference



Linear Modeling for the hills2000 data contd..

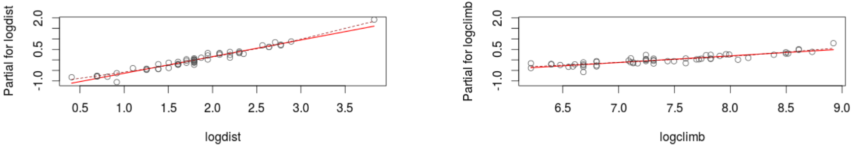
- If the model is correct and residuals are approximately normally distributed, a residual of this magnitude will occur about 2.6 times in 1000 residuals
- 2 * pnorm(-3) Output: [1] 0.002699796
- The resistant fit in the graph suggests that Beinn Lee and 12 Trig Trog are also outliers. These outliers may be a result of non-linearity
- termplot() can be used to check whether log(dist) and/or log(climb) should enter the model non-linearly

The contribution of the separate terms

- In order to interpret the plot easily, it is better to transform the variables before entering them into the model

```
lhills2k <- log(hills2000[, c("dist", "climb", "time")])
names(lhills2k) <- paste("log", names(lhills2k), sep="")
lhills2k.lm <- lm(logtime~logdist+logclimb, data=lhills2k)
termplot(lhills2k.lm, partial.resid=TRUE,
smooth=panel.smooth,
         col.res="gray30")
```

Result of the termplot()



- There is a small but clear departure from linearity, most evident for logdist
- The linear model does however give a good general summary of the indications in the data
- If the shortest and longest of the races are left out, it appears entirely adequate
- It is necessary to model the departure from linearity before proceeding further with checking residuals
- A quadratic term in logdist will work

A resistant fit that has a polynomial term in logdist

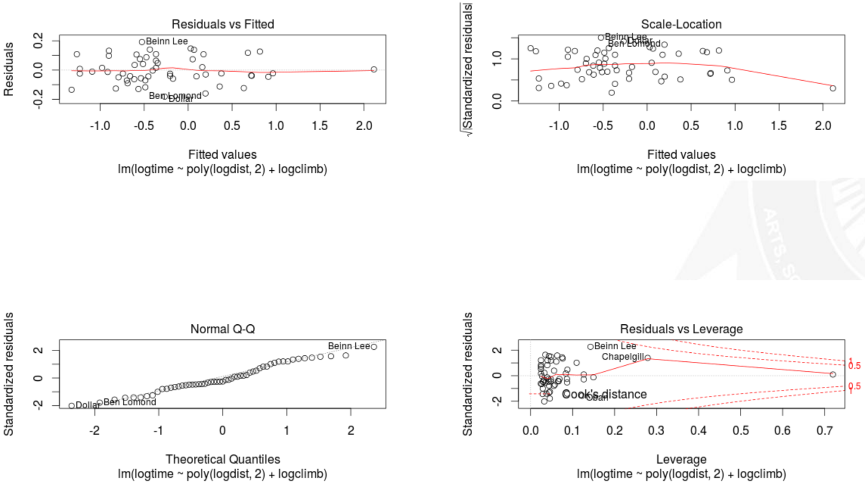
```
 reres2 <- residuals(lqs(logtime~poly(logdist,2)
+logclimb, data=lhills2k))
print(reres2[order(abs(reres2), decreasing=TRUE)[1:4]])
Output:
Caerketton  Beinn Lee Ardoch Rig  Cornalees
-0.4108673  0.3211900  0.2067721  0.1955449
```

- Caerketton is now the only large residual. In the sequel, we therefore omit Caerketton

Refining the model

```
lhills2k.lm2 <- lm(logtime~poly(logdist,2)+logclimb,
data=lhills2k[-42, ])
plot(lhills2k.lm2)
```

Plots of the refined model



What happens if we do not transform?

- If we avoid transformation and do not allow for increasing variability for the longer races we find several outlying observations, with the race that has the longest time highly influential

Problems with many explanatory variables

1. An informed guess as to what variables/factors are likely to be important. An extension of this approach classifies explanatory variables into a small number of groups according to an assessment of scientific importance. Fit the most important variables first, then add the next set of variables as a group, checking whether the fit improves from the inclusion of all variables in the new group
2. Interaction effects are sometimes modeled by including pairwise multiples of explanatory variables, e.g., x_1 x x_2 as well as x_1 and x_2

Problems with many explanatory variables

3. Principal components analysis is one of several methods that may be able to identify a small number of components, i.e., combinations of the explanatory variables, that together account for most of the variation in the explanatory variables. In favorable circumstances, one or more of the first few principal components will prove to be useful explanatory variables, or may suggest useful simple forms of summary of the original variables. In unfavorable circumstances, the components will prove irrelevant!
4. Discriminant analysis can sometimes be used to identify a summary variable

Answers for the II Monthly Test Question Paper

1. Discrete RV: 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 (1 Mark)
   Continuous RV:a variable is continuous if it can assume all values of a continuous scale (1 Mark)
   Differences:
   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. (1 Mark)For discrete the graph is a histogram while for a continuous rv the graph is smooth curve. (1 Mark)

Answers for the II Monthly Test Question Paper

2. Properties: (Any FOUR 4 X 1 = 4 Marks)
   1. The normal curve is symmetrical about the mean x=μ
   2. The height of normal curve is at its maximum at the mean
   3. The normal curve is uni modal at x =μ
   4. The point of inflexion occurs at $\pm\sigma$
   5. The first and third quartiles are equidistant from the median
3. Residual: The difference between the predicted value (based on the regression equation) and the actual, observed value (1 Mark) outlier:In linear regression, an outlier is an observation with large residual.(1 Mark) Leverage:Leverage is a measure of how far an independent variable deviates from its mean. (1 Mark) Cook's D:It measures the extent to which the line would change if the point were omitted. (1 Mark)

Answers for the II Monthly Test Question Paper

4. (a) Two formulae (2 X 1 = 2 Marks) (b)graph and graphviz (1 Mark) (c)MCMCregress(), with a similar syntax to lm(), that can be used for regression (1 Mark)
5. unclass(cover): gives the integer codes that identify levels calculations.(2 Marks) a¡-lm(y -1+x, data=d) (2 Marks)
6. (a) mean=0.44 (1 Mark), table (5 Marks)

```
  n   x           a
1 0 211 0.283376025
2 1  90 0.062342726
3 2  19 0.009143600
4 3   5 0.001005796
5 4   0 0.000000000
```

(b)(4 Marks)

```
n<-c(0,1,2,3,4)
x<-c(211,90,19,5,0)
N<-sum(x)

lambda<-sum(n*x)/N
print(lambda)
for(i in 0:4)
{
  a[i]=exp(-lambda)*lambda^i/factorial(i)
}
print(a)
d<-data.frame(n,x,a)
print(d)
```

Answers for the II Monthly Test Question Paper

7. (a) (8 Marks) (b) R code (2 Marks) Refer Class Notes
8. (8+2 Marks)

```
e<-rnorm(10)
x<-runif(10,min=-1,max=1)
y<-2+3*x+e
d1<-data.frame(x,y,e)
x1<-c(rep(-1,5),rep(1,5))
y1<-2+3*x1+e
d2<-data.frame(x1,y1,e)
print(d1)
print(d2)
#without intercept
y2<-3*x+e
d3<-data.frame(x,y2,e)
print(d3)
```

Multicollinearity

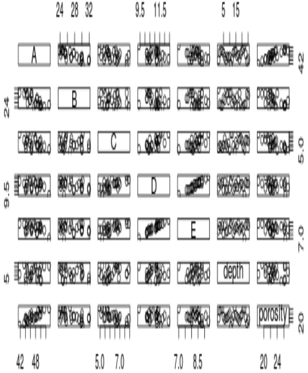
- Some explanatory variables may be linearly related to combinations of one or more of the other explanatory variables
- Technically, this is known as multicollinearity
- For each multi-collinear relationship, there is one redundant variable
- How to avoid more extreme effects of multicollinearity?
  - Careful thinking about the background science
  - Careful initial scrutiny of the data and
  - Removal of variables whose effect is already accounted for by other variables

Example

- The data set Coxite, in the compositions package, has the mineral compositions of 25 rock specimens of coxite type
- Each composition consists of the percentage by weight of five minerals, the depth of location, and porosity
- The names of the minerals are abbreviated to A = albite, B = blandite, C = cornite, D = daubite, and E = endite
- The analysis that follows is a relatively crude use of these data

Scatter Plot Matrix

```
data(Coxite)
coxite <- as.data.frame(Coxite)
pairs(coxite)
```



A Linear Model

```
coxiteAll.lm <- lm(porosity~A+B+C+D+E+depth, data=coxite)
summary(coxiteAll.lm)
```

Output

```
Output:
Call:
lm(formula = porosity ~ A + B + C + D + E + depth, data =

Residuals:
     Min       1Q   Median       3Q      Max
-0.93042 -0.46984  0.02421  0.35219  1.18217
```

Output

```
Coefficients: (1 not defined because of singularities)
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -217.74660  253.44389  -0.859    0.401
A              2.64863    2.48255   1.067    0.299
B              2.19150    2.60148   0.842    0.410
C              0.21132    2.22714   0.095    0.925
D              4.94922    4.67204   1.059    0.303
E                   NA         NA      NA       NA
depth          0.01448    0.03329   0.435    0.668

Residual standard error: 0.6494 on 19 degrees of freedom
Multiple R-squared:  0.9355,Adjusted R-squared:  0.9186
F-statistic: 55.13 on 5 and 19 DF,  p-value: 1.185e-10
```

Note

- The variable E, because it is a linear combination of earlier variables, adds no information additional to those variables. Effectively, its coefficient has been set to zero
- None of the individual coefficients comes anywhere near the usual standards of statistical significance
- The overall regression fit, with a p-value of 1.18×10^{-10} , is highly significant
- Pointwise confidence bounds can be obtained thus: hat <- predict(coxiteAll.lm, interval="confidence", level=0.95)

Pointwise Confidence Bounds

```
        fit      lwr      upr
1  22.35377 21.55079 23.15675
2  24.88085 24.21571 25.54598
3  25.74781 25.09407 26.40154
4  26.16992 25.46261 26.87724
5  23.43656 22.79668 24.07643
6  21.67586 21.16465 22.18707
7  20.81783 20.16378 21.47187
8  22.93042 22.43819 23.42265
9  23.29712 22.55150 24.04274
10 23.58893 22.91217 24.26569
11 18.08165 17.36894 18.79436
12 20.66938 19.99506 21.34371
13 23.57699 23.13436 24.01963
14 26.10297 25.48422 26.72173
15 24.09866 23.65937 24.53794
16 22.52641 21.87774 23.17507
17 20.88399 20.11536 21.65261
18 20.05628 19.46561 20.64695
19 22.18044 21.69818 22.66270
20 23.95788 23.43339 24.48238
21 25.89225 25.23359 26.55090
22 26.37182 25.47439 27.26925
23 22.95658 22.38832 23.52484
24 19.96984 19.07501 20.86466
25 21.27579 20.41019 22.14139
```

The variance inflation factor

- The variance inflation factor (VIF) measures the effect of correlation with other variables in increasing the standard error of a regression coefficient
- If x$_j$, with values x$_{ij}$ (i=1,...,n) is the only variable in a straight line regression model, and b$_j$ is the estimated coefficient then:

$$var[b_j] = \frac{\sigma^2}{s_{jj}} \; where \; s_{jj} = \sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2 \tag{8}$$

- σ^2 is the variance of the error term in the model
- When further terms are included in the regression model, this variance is inflated, as a multiple of σ^2 , by the variance inflation factor
- VIF depends only on the model matrix. It does not reflect changes in the residual variance

How to obtain VIF?

```
vif(lm(porosity~A+B+C+D+depth, data=coxite))
Output:
        A         B         C         D     depth
2717.8000 2485.0000  192.5900  566.1400    3.4166
```

- Given the size of these factors, it is unsurprising that none of the individual coefficients can be estimated meaningfully

How to proceed?

- Try a model that uses those variables that, individually, correlate most strongly with porosity

```
cor(coxite$porosity, coxite)
Output:
          A          B          C          D      E

[1,] 0.8690284 -0.5511044 -0.7233127 -0.3199149 -0.4075
         depth              porosity
[1,]    -0.1467961              1
```

Another Model

```
summary(coxiteABC.lm <- lm(porosity~A+B+C, data=coxite))
Output:
Call:
lm(formula = porosity ~ A + B + C, data = coxite)

Residuals:
     Min       1Q   Median       3Q      Max
-0.98137 -0.37455  0.02294  0.41742  1.27272

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 53.30463   13.22186   4.032 0.000603 ***
A           -0.01246    0.15580  -0.080 0.937003
B           -0.58668    0.15134  -3.876 0.000873 ***
C           -2.21880    0.33886  -6.548 1.74e-06 ***
---
```

Output Contd..

```
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.6425 on 21 degrees of freedom
Multiple R-squared:  0.9302,Adjusted R-squared:  0.9203
F-statistic: 93.35 on 3 and 21 DF,  p-value: 2.639e-12
```

Again VIF?

```
vif(coxiteABC.lm)
Output:
      A        B        C
10.9360  8.5924  4.5551
```

Another Simplified Model

```
 summary(coxiteBC.lm <- lm(porosity~B+C, data=coxite))

Call:
lm(formula = porosity ~ B + C, data = coxite)

Residuals:
     Min       1Q   Median       3Q      Max
-0.98353 -0.37851 -0.00347  0.41783  1.26453
```

Output

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 52.25713    1.78312   29.31  < 2e-16 ***
B           -0.57531    0.05078  -11.33 1.19e-10 ***
C           -2.19490    0.15617  -14.05 1.81e-12 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.6278 on 22 degrees of freedom
Multiple R-squared:  0.9302,Adjusted R-squared:  0.9239
F-statistic: 146.6 on 2 and 22 DF,  p-value: 1.909e-13
```

Comparison of two models using AIC

- Using the AIC statistic to compare this model with the model that used all six explanatory variables

```
AIC(coxiteAll.lm, coxiteBC.lm)
Output:
             df       AIC
coxiteAll.lm  7 56.49973
coxiteBC.lm   4 52.47331
```

- Akaike's Information Criterion is a criterion for selecting among nested econometric models
- The simpler model wins

Remedies for Multicollinearity

- Careful initial choice of variables, based on scientific knowledge and careful scrutiny of relevant exploratory plots of explanatory variables, will often avert the problem
- Occasionally, it may be possible to find or collect additional data that will reduce correlations among the explanatory variables
- Ridge regression is one of several approaches that may be used to alleviate the effects of multicollinearity, in inflating coefficients and their standard errors
- Use of the function lm.ridge() requires the user to choose the tuning parameter lambda

Errors in x

- The discussion so far has been assumed: either that the explanatory variables are measured with negligible error, or that the interest is in the regression relationship given the observed values of explanatory variables
- This subsection draws attention to the effect that errors in the explanatory variables can have on regression slope
- Discussion is mainly on the relatively simple "classical" errors in x model
- With a single explanatory variable, the effect under the classical "errors in x" model is to reduce the expected magnitude of the slope, that is, the slope is attenuated
- Furthermore, the estimated slope is less likely to be distinguishable from statistical noise

Simulations of the effect of measurement error

- Suppose that the underlying regression relationship that is of interest is:

$$y_i = \alpha \ + \ \beta \ x_i \ + \ \epsilon_i \ var[\epsilon_i] = \sigma^2 \ (i = 1, 2, ..., n) \tag{9}$$

$$s_x = \sqrt{(}\sum_{i=1}^{n}(x_i - \bar{x})^2/(n-1)) \tag{10}$$

- be the standard deviation of the values that are measured without error
- Take the measured values as w$_i$=x$_i$+η_i, where var[η_i]=s$_x^2$ τ^2
- The η_i are assumed independent of the ϵ_i
- If $\tau = 0.4$ (the added error has a variance that is 40% of s$_x$), the effect on the slope is modest. If $\tau = 2$, the attenuation is severe

Simulations of the effect of measurement error

- An estimate of the attenuation in the slope is, to a close approximation:

$$\lambda = \frac{1}{1+\tau^2} \tag{11}$$

- Here, λ has the name reliability ratio

Two explanatory variables

- Consider first the case where one predictor is measured with error, and others without error. The coefficient of the variable that is measured with error is attenuated, as in the single variable case
- The coefficients of other variables may be reversed in sign, or show an effect when there is none
- Suppose that y $=\beta_1 x_1 + \beta_2 x_2 + \epsilon$
- If w_1 is unbiased for x_1 and the measurement error η is independent of x_1 and x_2 , then least squares regression with explanatory variables w_1 and x_2 yields an estimate of $\lambda \ \beta_1$, where if ρ is the correlation between x_1 and x_2:

$$\lambda = \frac{1-\rho^2}{1-\rho^2+\tau^2} \tag{12}$$

Two Explanatory Variables

- A new feature is the bias in the least squares estimate of β_2
- The naive least squares estimator estimates

$$\beta_2 + \beta_1(1 - \lambda)\gamma_{12}, \text{ where } \gamma_{12} = \rho\frac{s_1}{s_2} \tag{13}$$

- Here, γ_{12} is the coefficient of x_2 in the least squares regression of x_1 on x_2, $s_1 = SD[x_1]$ and $s_2 = SD[x_2]$
- The estimate of β_2 may be substantially different from zero, even though $\beta_2 = 0$
- Where $\beta_2 = 0$, the least squares estimate can be reversed in sign from β_2
- Some of the effect of x_1 is transferred to the estimate of the effect of x_2

Two explanatory variables, one measured without error a simulation

- The function errorsINseveral() simulates a model where there are two continuous variables x_1 and x_2 . The default choice of arguments has β_1=1.5, β_2=0,ρ=-0.5, s_1=s_2=2, τ=1.5,var[ϵ]=0.25
- Measurement error variances are x_1:s_1^2 τ^2, x_2=0. Then λ=0.25, γ_{12}=-0.5
- and the expected value for the naive least squares estimator of β_2 is

$$\beta_2+\beta_1(1-\lambda)\gamma_{12} \ = \ 0 + 1.5x0.75x(-0.5) \ = \ -0.5675 \quad (14)$$

Simulation Results

```
errorsINseveral()
                          Intercept    b1     b2 SE(Int) SE
Values for simulation         2.496 1.500  0.000      NA
Estimates: no error in x1     2.713 1.498 -0.015   0.190  0
LS Estimates: error in x1    23.087 0.389 -0.544   0.589  0
Theoretical attenuation of b1                Theoretical b2
                       0.2500                       -0.5625
```

Time Series Models

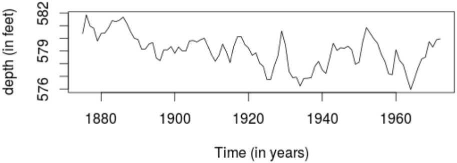
- Time series models are very useful models when you have serially correlated data
- Most of business houses work on time series data to analyze sales number for the next year, website traffic, competition position and much more
- However, it is also one of the areas, which many analysts do not understand

Time Series Models

- A time series is a sequence of observations that have been recorded over time
- Almost invariably, observations that are close together in time are more strongly correlated than observations that are widely separated
- The independence assumption of previous discussion is, in general, no longer valid
- The analyses will use functions in the stats package, which is a recommended package, included with binary distributions. Additionally, there will be use of the forecast package
- Non-linear time series (ARCH and GARCH models) will require access to the tseries package

Time series some basic ideas

- The time series object LakeHuron (datasets) has annual depth measurements at a specific site on Lake Huron
- The discussion of sequential dependence, and of the use of ARIMA-type models of this dependence, will use these data for illustrative purposes
- Preliminary graphical explorations
  - plot(LakeHuron, ylab="depth (in feet)", xlab = "Time (in years)")

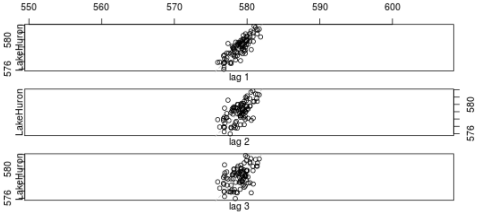


Lag Plots

- Lag plots may give a useful visual impression of the dependence
- Suppose that our observations are x_1, x_2, . . . , x_n . Then the lag 1 plot plots x_i against x_{i1} (i = 2, . . . , n), thus:
  y-value : x_2 x_3 .... x_n
  lag 1(x-axis): x_1 x_2..... x_{n-1}
- For a lag 2 plot, x_i is plotted against x_{i-2} (i = 3, . . . ,n), and so on for higher lags
- The first four lag plots for the Lake Huron data is shown in the next slide

Lag Plots Contd..

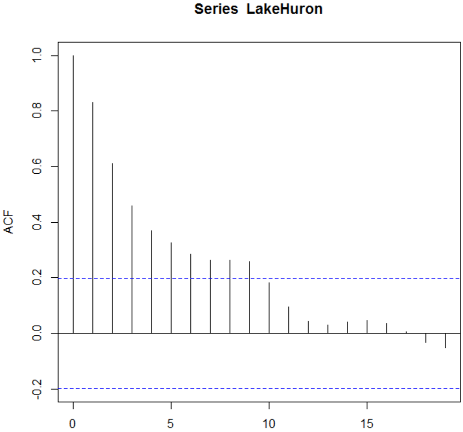
- lag.plot(LakeHuron, lags=3, do.lines=FALSE)



- The scatter of points about a straight line in the first lag plot suggests that the dependence between consecutive observations is linear
- The second, third, and fourth lag plots show increasing scatter
- As might be expected, points separated by two or three lags are successively less dependent
- Note that the slopes, and therefore the correlations, are all positive

The autocorrelation and partial autocorrelation function

- Self correlation
- The autocorrelation function (ACF), which gives the autocorrelations at all possible lags
- acf(LakeHuron)

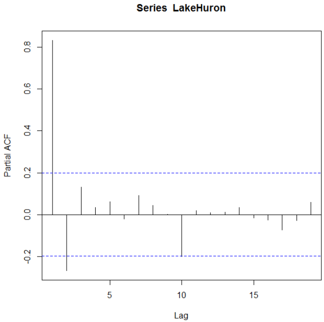


About Autocorrelation Contd..

- The autocorrelation at lag 0 is included by default; this always takes the value 1, since it is the correlation between the data and themselves
- Interest is in the autocorrelations at lag 1 and later lags
- As inferred from the lag plots, the largest autocorrelation is at lag 1 (sometimes called the serial correlation), with successively smaller autocorrelations at lags 2, 3, ...
- There is no obvious linear association among observations separated by lags of more than about 10

Partial Autocorrelation

- The partial autocorrelation at a particular lag measures the strength of linear correlation between observations separated by that lag, after adjusting for correlations between observations separated by fewer lags
- pacf(LakeHuron)



Effects of Autocorrelation

- Autocorrelation in a time series complicates the estimation of such quantities as the standard error of the mean
- There must be appropriate modeling of the dependence structure

Autoregressive models

- Autoregressive (AR) models move beyond attention to correlation structure to the modeling of regressions relating successive observations
- AR(1)
  - The autoregressive model of order 1 (AR(1)) for a time series X_1, X_2, . . . has the recursive formula $X_t = \mu + \alpha(X_{t1}$ - $\mu) +$ ϵ_t, t = 1, 2, . . . where μ and α are parameters
  - Usually, α takes values in the interval (-1, 1); this is the so-called stationary case
  - Nonstationarity, in the sense used here, implies that the properties of the series are changing with time
  - The mean may be changing with time, and/or the variances and covariances may depend on the time lag
  - Any such nonstationarity must be removed or modeled
  - The error term ϵ_t is the familiar independent noise term with constant variance σ^2
  - The distribution of X_0 is assumed fixed and will not be of immediate concern

AR(1) Contd..

- For the AR(1) model, the ACF at lag i is α^{i} , for i = 1, 2, ... If $\alpha = 0.8$, then the observed autocorrelations should be 0.8, 0.64, 0.512, 0.410, 0.328, . . . , a geometrically decaying pattern
- To gain some appreciation for the importance of models like the AR(1) model, we consider estimation of the standard error for the estimate of the mean μ
- Under the AR(1)model, a large-sample approximation to the standard error for the mean is:

$$\frac{\sigma}{\sqrt{n}}\frac{1}{(1-\alpha)} \qquad (15)$$

AR(1) Contd..

- There are several alternative methods for estimating for the parameter α
- The method of moments estimator uses the autocorrelation at lag 1, here equal to 0.8319
- The maximum likelihood estimator, equal to 0.8376, is an alternative

AR(1) Contd..

```
LH.yw <- ar(x = LakeHuron, order.max = 1, method = "yw")
# autocorrelation estimate
# order.max = 1 for the AR(1) model
print(LH.yw$ar)
# autocorrelation estimate of alpha
## Maximum likelihood estimate
LH.mle <- ar(x = LakeHuron, order.max = 1, method = "mle")
print(LH.mle$ar)
# maximum likelihood estimate of alpha
print(LH.mle$x.mean)
# estimated series mean
print(LH.mle$var.pred)
# estimated innovation variance
```

AR(1) Contd..

```
[1] 0.8319112
[1] 0.837546
[1] 579.1141
[1] 0.5092867
```

The general AR(p) model

- It is possible to include regression terms for X$_t$ against observations at greater lags than one
- The autoregressive model of order p (the AR(p) model) regresses X$_t$ against X$_{t-1}$, X$_{t-2}$, . . . ,X$_{t-p}$:

$$X_t = \mu + \alpha_1(X_{t-1} - \mu) + ... + \alpha_p(X_{t-p} - \mu) + \epsilon_t, t = 1, 2, ..., \tag{16}$$

- where α_1, α_2,...,α_p are additional parameters that would need to be estimated
- The parameter α_i is the partial autocorrelation at lag i
- Assuming an AR process, how large should p be, i.e., how many AR parameters are required? The function ar(), in the stats package, can be used to estimate the AR order

General AR(p) model

```
ar(LakeHuron, method="mle")

Call:
ar(x = LakeHuron, method = "mle")

Coefficients:
      1        2
 1.0437  -0.2496

Order selected 2  sigma^2 estimated as  0.4788
ar(LakeHuron, method="mle")
# AIC is used by default if
# order.max is not specified
```

Autoregressive moving average models

- In a moving average (MA) process of order q, the error term is the sum of an innovation ϵ_t that is specific to that term, and a linear combination of earlier innovations ϵ_{t-1},ϵ_{t-2},...,ϵ_{t-q}. The equation is

$$X_t = \mu_t + \epsilon_t + b_1\epsilon_{t-1} + \ + b_q\epsilon_{t-q} \tag{17}$$

- where ϵ_1, ϵ_2, ... , ϵ_q are independent random normal variables, all with mean 0
- The autocorrelation between terms that are more than q time units apart is zero
- Moving average terms can be helpful for modeling autocorrelation functions that are spiky, or that have a sharp cutoff

Auto Regressive Moving Average Model

- An autoregressive moving average (ARMA) model is an extension of an autoregressive model to include "moving average" terms
- Autoregressive integrated moving average(ARIMA) models allow even greater flexibility. (The model (AR or MA or ARMA) is applied, not to the series itself, but to a differenced series, where the differencing process may be repeated more than once.)
- There are three parameters: the order p of the autoregressive component, the order d of differencing (in our example 0), and the order q of the moving average component

Auto Regressive Moving Average Model

- Differencing of order 1 removes a linear trend
- Differencing of order 2 removes a quadratic trend
- The downside of differencing is that it complicates the correlation structure
- Thus, it turns an uncorrelated series into an MA series of order 1
- Differencing can be done explicitly prior to analysis
- However it is easiest to let the R time series modeling functions do any differencing that is required internally, then reversing the process following the fitting of an ARMA model to the differenced series

Automatic Model Selection

- The function auto.arima() from the forecast package uses the AIC in a model selection process that can proceed pretty much automatically
- This takes some of the inevitable arbitrariness from the selection process
- Fortunately, in applications such as forecasting or the regression example in the next section, what is important is to account for the major part of the correlation structure
- A search for finesse in the detail may be pointless, with scant practical consequence
- The algorithm looks for the optimum AR order p, the optimum order of differencing, and the optimum MA order q
- Additionally, there is provision for seasonal terms and for "drift"
- Drift implies that there is a constant term in a model that has $d > 0$

Automatic Model Selection

- An exhaustive (non-stepwise) search can be very slow
- The auto.arima() default is a stepwise search
- At each iteration, the search is limited to a parameter space that is "close" to the parameter space of the current model
- Values of AR and MA parameters are allowed to change by at most one from their current values
- There are similar restrictions on the search space for seasonal and other parameters

Example

- library(forecast)
- auto.arima(LakeHuron)
- Output:

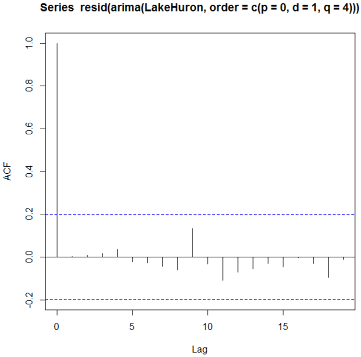
```
Series: LakeHuron
ARIMA(0,1,4) with drift

Coefficients:
         ma1      ma2      ma3      ma4    drift
      0.0584  -0.3158  -0.3035  -0.2349  -0.0205
s.e.  0.1031   0.1058   0.1100   0.1299   0.0167

sigma^2 estimated as 0.4806:  log likelihood=-101.09
AIC=214.18   AICc=215.12   BIC=229.63
```

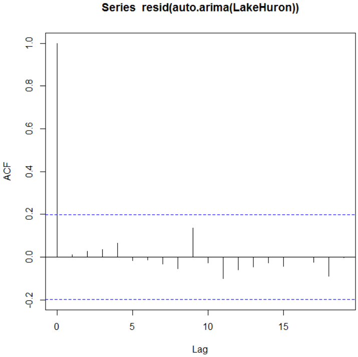
Checking with auto correlation

```
## Check that model removes
 most of the correlation
 structure
acf(resid(arima(LakeHuron,
 order=c(p=1, d=1, q=2))))
```



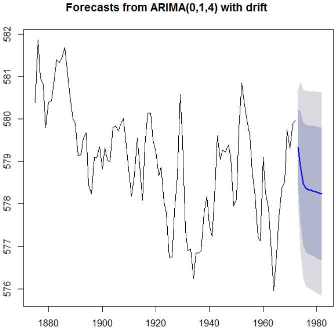
Checking with auto correlation

```
## The following achieves
the same effect like the
previous figure, for these data
acf(resid(auto.arima(LakeHuron)
```



Time Series Forecast

```
LH.arima <- auto.arima(LakeHuron)
fcast <- forecast(LH.arima)
plot(fcast)
```



Non Linear Time Series

- In the ARMA models so far considered, the error structures have been constructed from linear combinations of the innovations, and the innovations have been i.i.d. normal random variables
- Such models are unable to capture the behavior of series where the variance fluctuates widely with time, as happens for many financial and economic time series
- ARCH (autoregressive conditionally heteroscedastic) and GARCH (generalized ARCH) models have been developed to meet these requirements
- The principal idea behind a GARCH model is that there is an underlying (or hidden) process which governs the variance of the noise term while ensuring that these noise terms at different times remain uncorrelated

Model with normal ARCH(1) errors

- The error term at the current time step is normally distributed with mean 0 and with a variance linearly related to the square of the error at the previous time step
- In other words, squares of noise terms form an autoregressive process of order 1
- An AR(1) process with ARCH(1) errors is given by

$$X_t = \mu + \alpha(X_{t-1} - \mu) + \epsilon_t \tag{18}$$

- where ϵ_t is normal with mean 0 and variance $\sigma_t^2 = \alpha_0 + \alpha\epsilon_{t-1}^2$
- The error terms ϵ_t are uncorrelated, while their squares have serial correlations with the squares of historical values of the error term

GARCH Models

- GARCH models are an extension of ARCH models
- In a GARCH model of order (p, q), σ_t^2 is the sum of two terms: (1) a linear function both of the previous p squares of earlier errors, as for an ARCH model of order p and (2) a linear function of the variances of the previous q error terms previous q error terms
- The function garch() allows for estimation of the mean and the underlying process parameters for a given time series by maximum likelihood, assuming normality

Simulation of ARCH(1) Process

- α_0=0.25,α_1=0.95

```
x <- numeric(999) # x will contain the ARCH(1) errors
x0 <- rnorm(1)
for (i in 1:999){
x0 <- rnorm(1, sd=sqrt(.25 + .95*x0^2))
x[i] <- x0
}
print(garch(x, order = c(0, 1), trace=FALSE))
```

Output

```
Coefficient(s):
    a0      a1
0.2315  0.8906
```