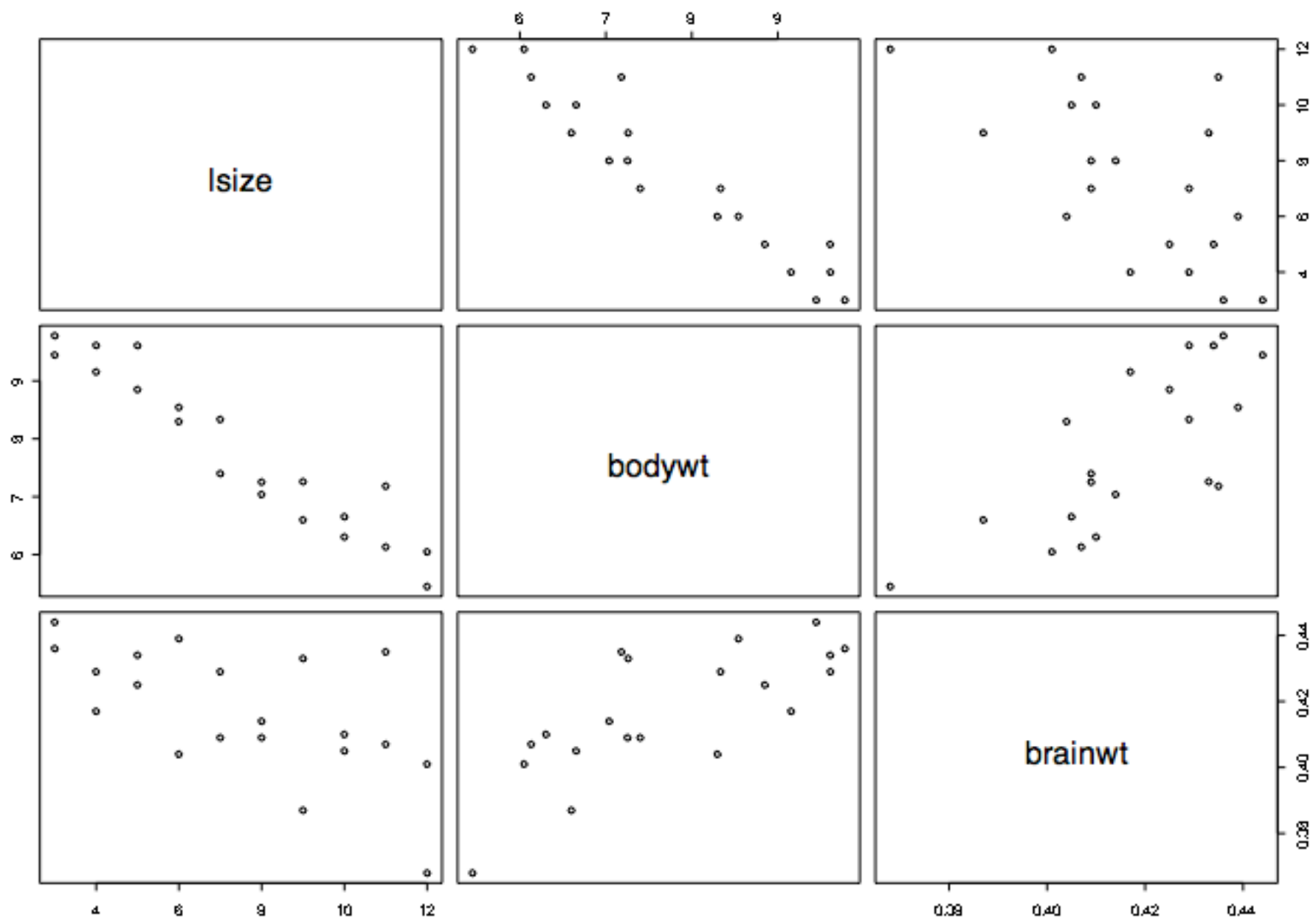


Fitting Linear Models

DAAG Chapter 6

interpretation of coefficients can be tricky...

```
> litters
  lsize bodywt brainwt
1      3  9.447   0.444
2      3  9.780   0.436
3      4  9.155   0.417
4      4  9.613   0.429
5      5  8.850   0.425
6      5  9.610   0.434
7      6  8.298   0.404
8      6  8.543   0.439
9      7  7.400   0.409
10     7  8.335   0.429
11     8  7.040   0.414
12     8  7.253   0.409
13     9  6.600   0.387
14     9  7.260   0.433
15    10  6.305   0.410
16    10  6.655   0.405
17    11  7.183   0.435
18    11  6.133   0.407
19    12  5.450   0.368
20    12  6.050   0.401
> pairs(litters)
```

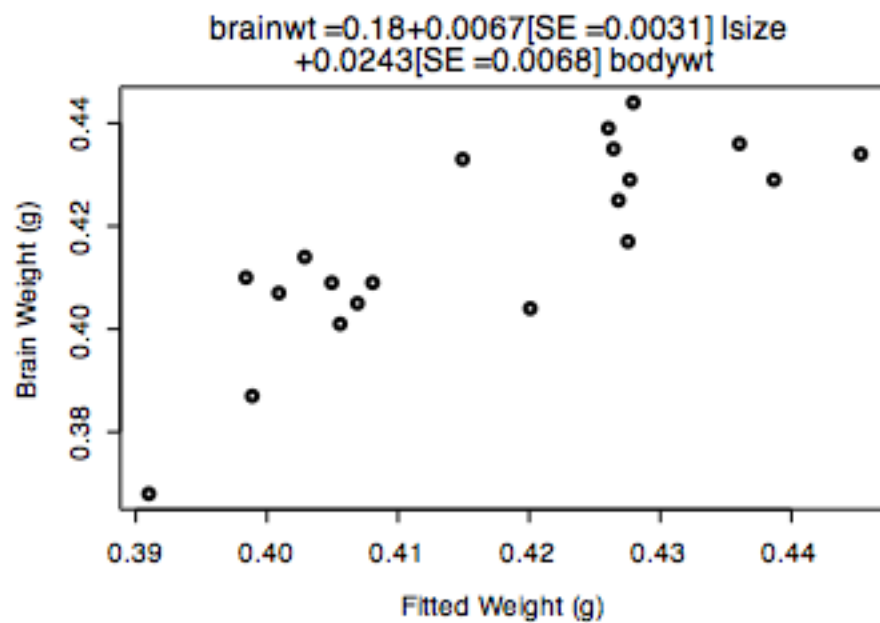
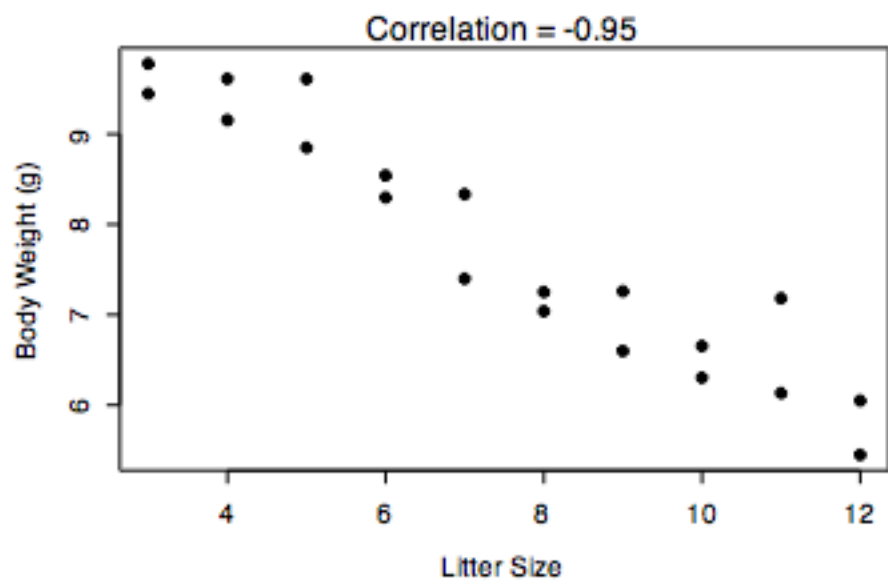
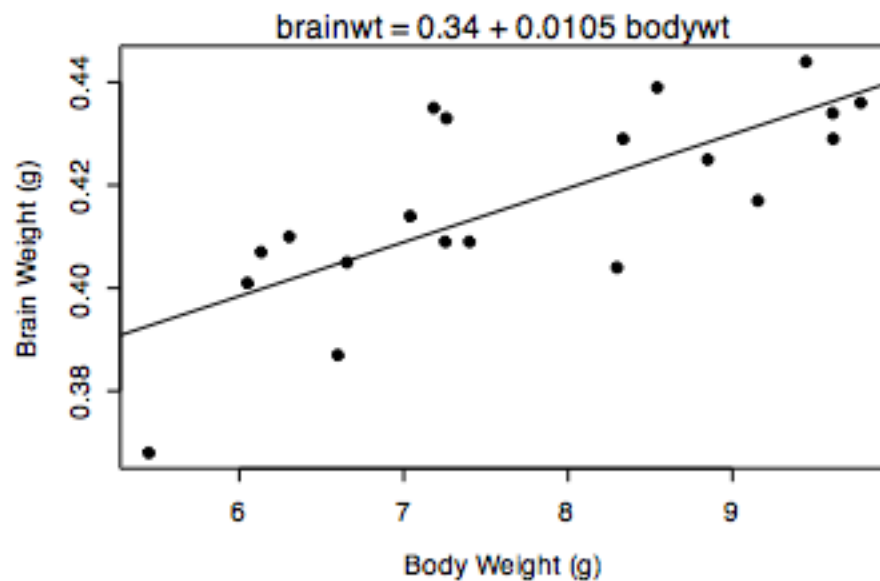
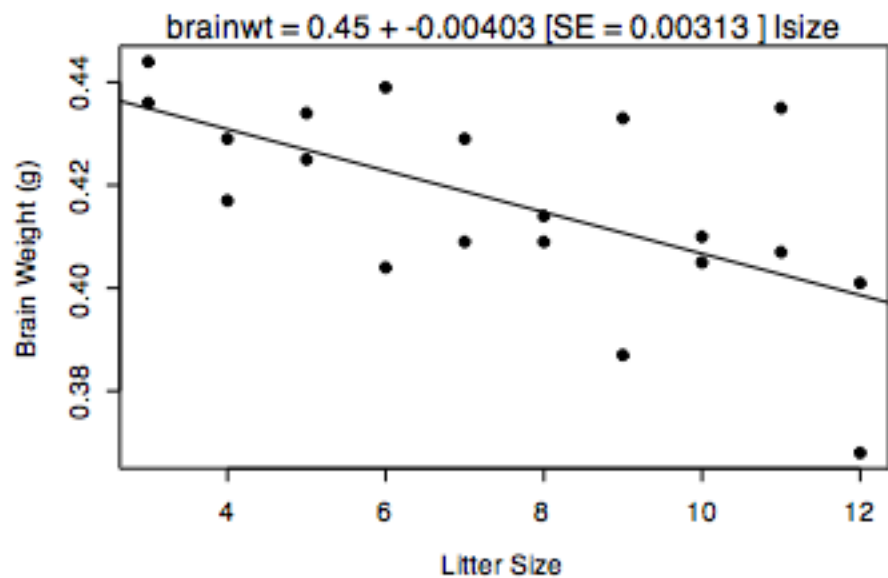


```

titl <- "Brain Weight as a Function of Litter Size and Body Weight"
plot(litters[, 1], litters[, 3], xlab = "Litter Size", ylab =
      "Brain Weight (g)", pch = 16)
u1 <- lm(brainwt ~ lsize, data = litters)
se <- summary(u1)$coef[2, 2]
abline(u1)
mtext(side = 3, line = 0.25, text = paste("brainwt =", round(u1$coef[1], 2), "+",
      round(u1$coef[2], 5), "[SE =", round(se, 5), "]", "lsize"), cex = 1.0)

plot(litters[, 2], litters[, 3], xlab = "Body Weight (g)", ylab =
      "Brain Weight (g)", pch = 16)
u2 <- lm(brainwt ~ bodywt, data = litters)
abline(u2)
mtext(side = 3, line = 0.25, text = paste("brainwt =", round(u2$coef[1], 2), "+",
      round(u2$coef[2], 4), "bodywt"), cex = 1.0)
plot(litters[, 1], litters[, 2], xlab = "Litter Size", ylab =
      "Body Weight (g)", pch = 16, mkh = 0.04)
r3 <- cor(litters[, 1], litters[, 2])
mtext(side = 3, line = 0.25, text = paste("Correlation =", round(r3, 2)
      ), cex = 1.0)
u <- lm(brainwt ~ lsize + bodywt, data = litters)
hat <- fitted(u)
plot(hat, litters[, 3], xlab = "Fitted Weight (g)", ylab =
      "Brain Weight (g)", pch = 1, lwd=2)
se <- summary(u)$coef[2, 2]
se1 <- summary(u)$coef[3, 2]
mtext(side = 3, line = 0.5,
      text = paste("brainwt =",
        round(u$coef[1], 2), "+", round(u$coef[2], 4),
        "[SE =", round(se, 4), "]" " ", "lsize \n+",
        round(u$coef[3], 4), "[SE =", round(se1, 4), "]" " ",
        "bodywt", sep=""), cex = 1.0)

```



```
> summary(u)
```

```
Call:
```

```
lm(formula = brainwt ~ lsize + bodywt, data = litters)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-0.0230005	-0.0098821	0.0004512	0.0092036	0.0180760

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	0.178247	0.075323	2.366	0.03010	*
lsize	0.006690	0.003132	2.136	0.04751	*
bodywt	0.024306	0.006779	3.586	0.00228	**

```
---
```

```
Residual standard error: 0.01195 on 17 degrees of freedom
```

```
Multiple R-Squared: 0.6505, Adjusted R-squared: 0.6094
```

```
F-statistic: 15.82 on 2 and 17 DF, p-value: 0.0001315
```

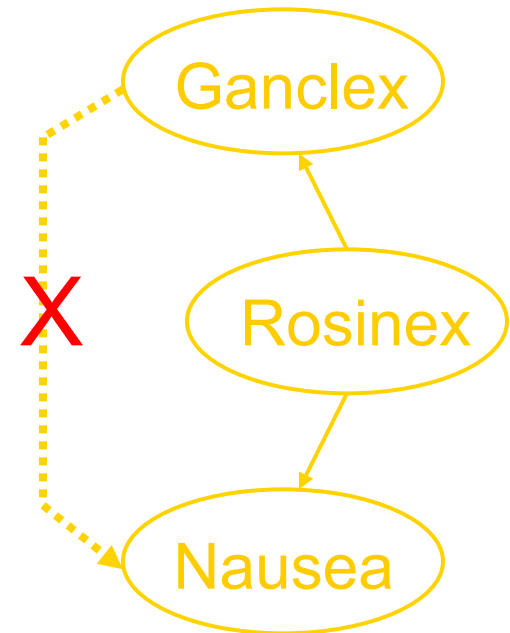
estimated amount by which $E[Y]$ increases as lsize goes up by one, holding bodywt constant

"Simpson's Paradox"

- 2 X 2 table analysis ignores effects of drug-drug association on drug-AE association



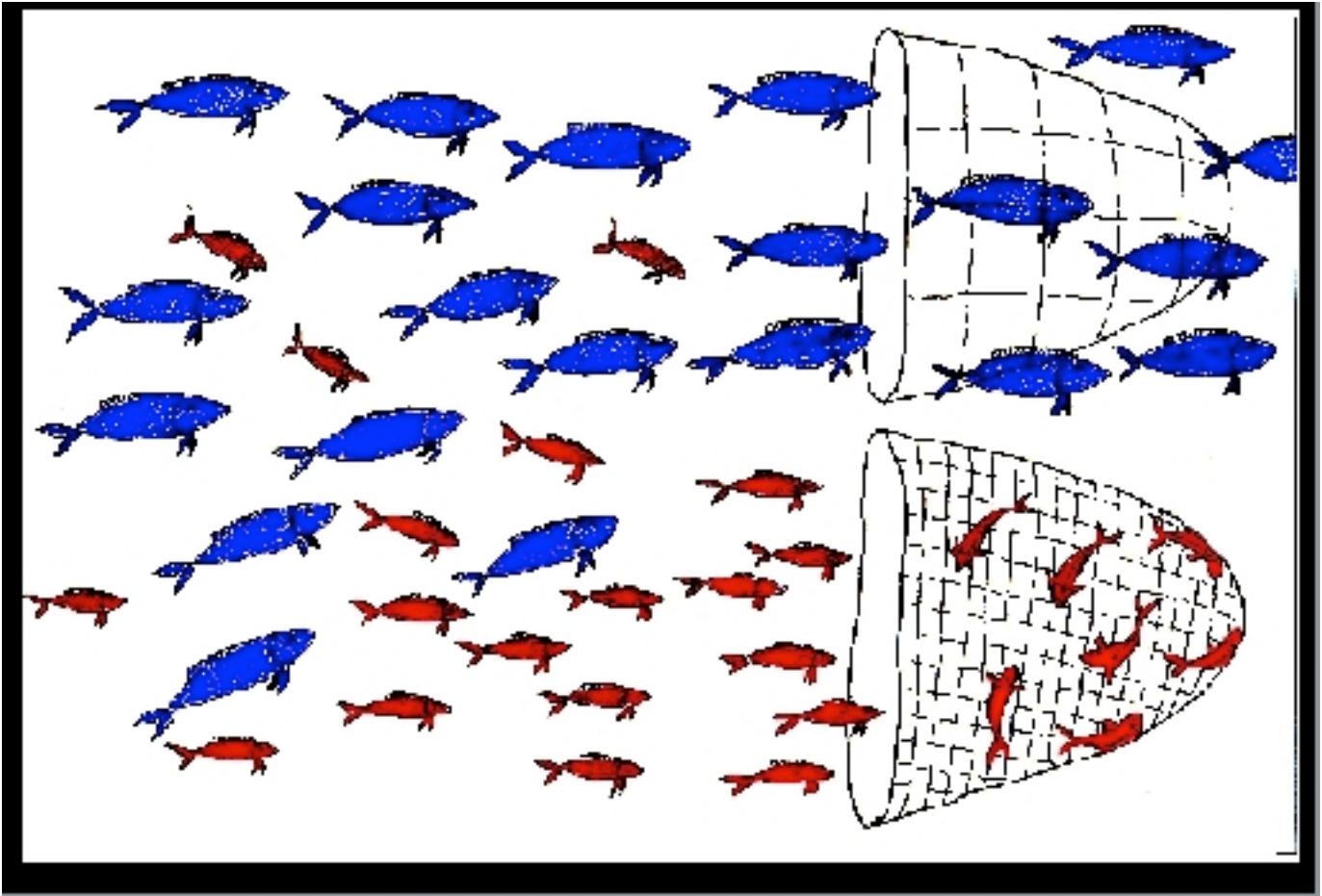
	Rosinex		No Rosinex		Total	
	No Nausea	No Nausea	No Nausea	No Nausea	No Nausea	No Nausea
Ganclex	81	9	1	9	82	18
No Ganclex	9	1	90	810	99	811
RR	1		1		4.58	



berkeley admission's data 1973

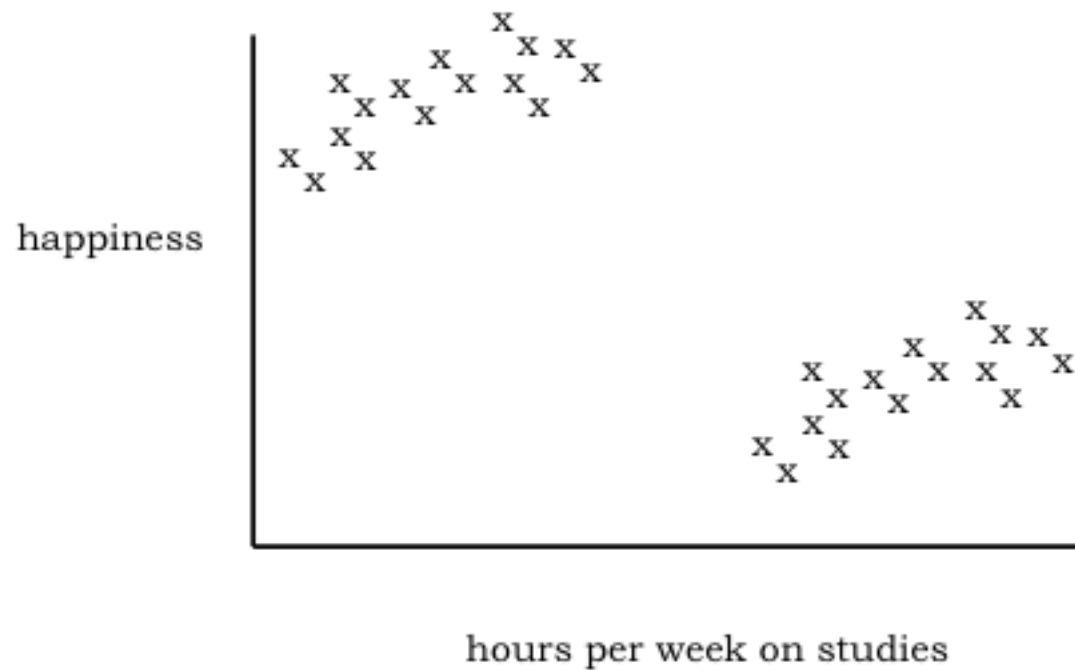
	Applicants	% admitted
Men	8442	44%
Women	4321	35%

Major	Men		Women	
	Applicants	% admitted	Applicants	% admitted
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	272	6%	341	7%

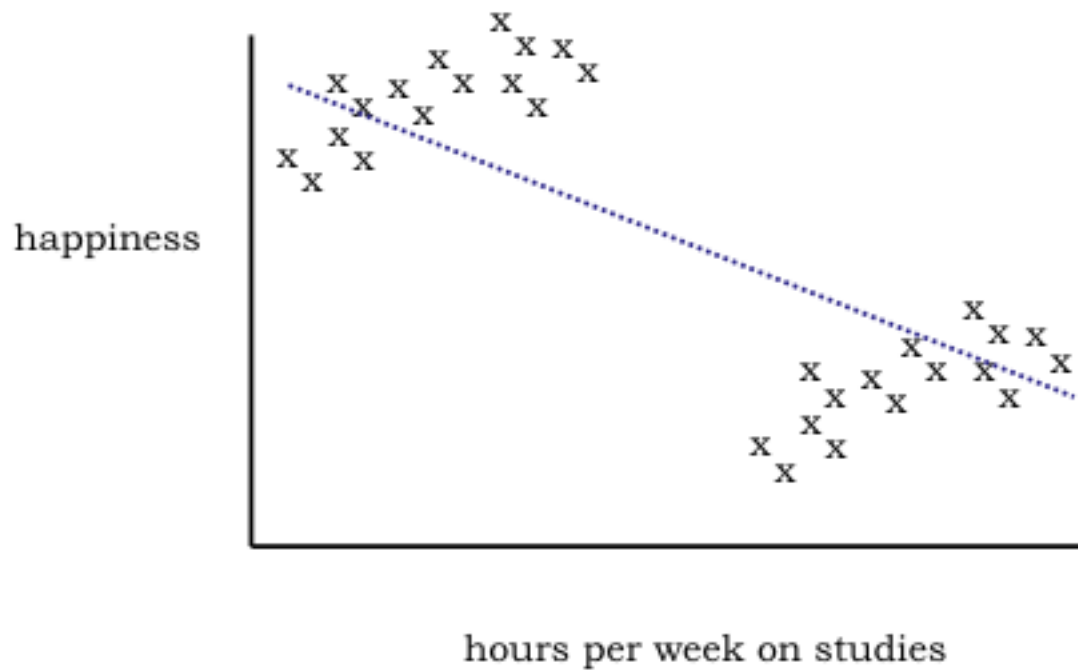


Bad Things Can Happen...

DATA

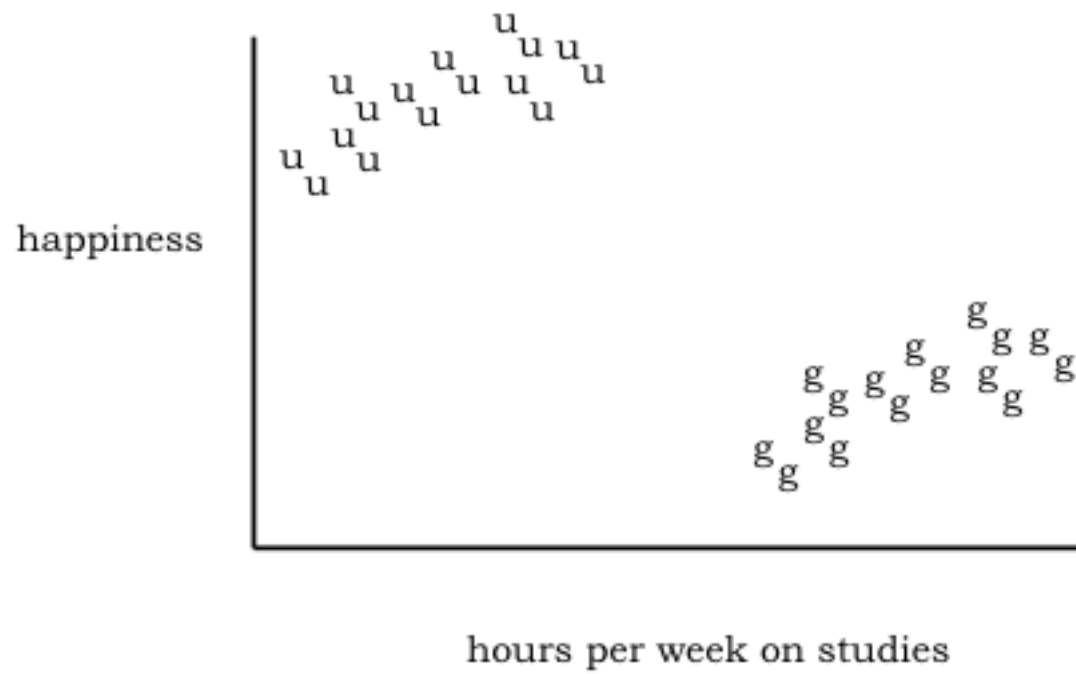


simple regression line

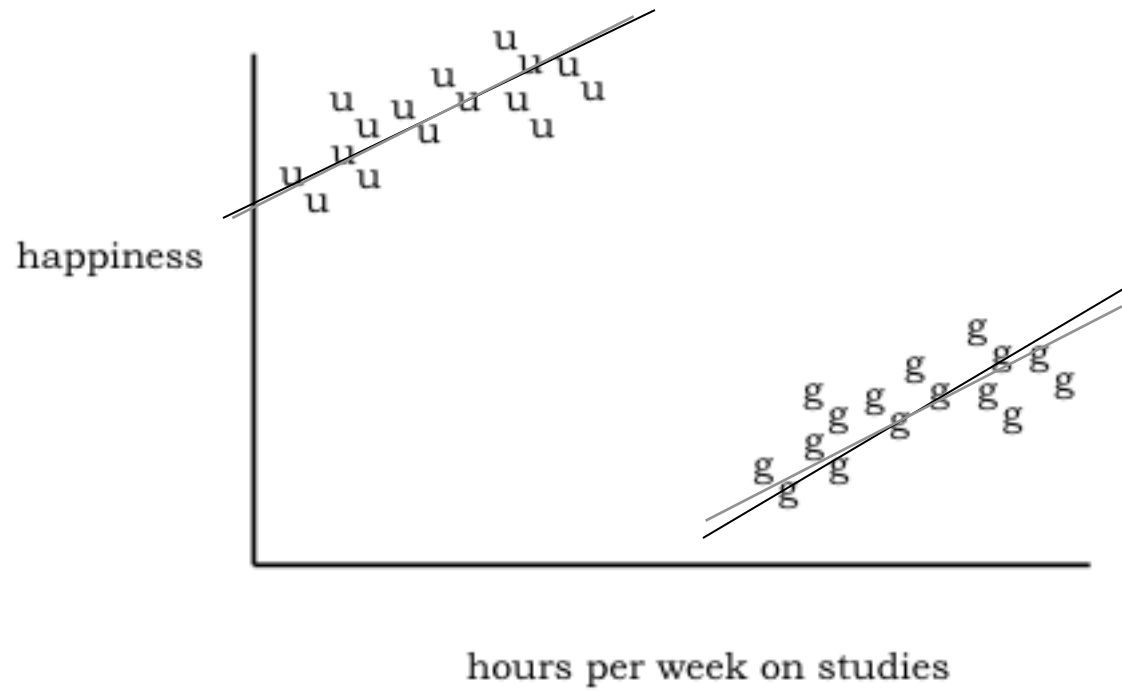


$$\text{HAP} = \beta_0 + \beta_1 \times \text{HOURS}, \beta_1 \text{ will be estimated to be negative}$$

A 2nd Look at the DATA



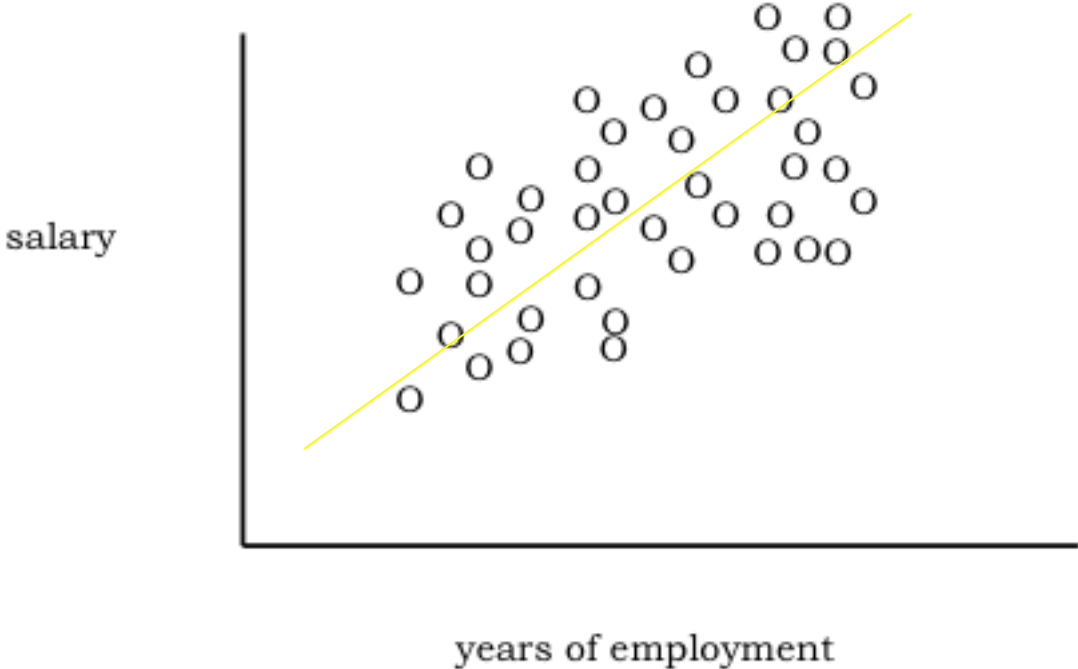
A 2nd Look at the DATA

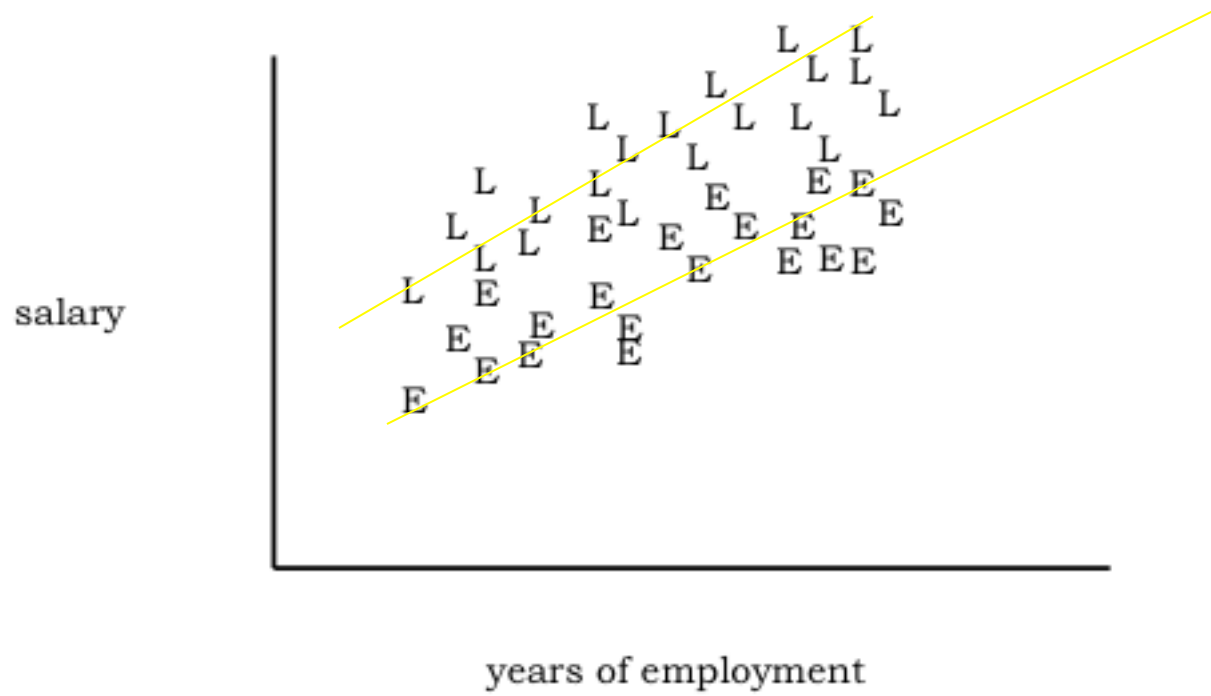


Other Odd Things Can Happen...



Other Odd Things Can Happen...





important diversion: these issues are related to "confounding"

but...

are not the same...

Confounding and Causality

- Confounding is a causal concept

Outcome	Population D		Population d	
	Drug (factual)	Not drug (counterfactual)	Drug (counterfactual)	Not drug (factual)
Y=1	30	20	30	10
Y=0	70	80	70	90
	<u>a=0.3</u>	<u>b=0.2</u>		<u>c=0.1</u>

True causal effect = $a/b = 1.5$ or $a / (1-a) \div b / (1-b) = 1.71$

Estimated causal effect = $a/c = 3$ or $a / (1-a) \div c / (1-c) = 3.86$

- “The association in the combined D+d populations is confounded for the effect in population D”

Why does this happen?

- For confounding to occur there must be some characteristics/covariates/conditions that distinguish D from d.
- However, the existence of such factors does not in and of itself imply confounding.
- For example, D could be males and d females but it could still be the case that $b=c$.

Stratification can introduce confounding

	Population D		Population d	
Outcome	Drug (actual)	Not drug (counter)	Drug (counter)	Not drug (actual)
Y=1	30	20	30	20
Y=0	70	80	70	80
	a=0.3	b=0.2		c=0.2

True causal effect = $a-b = 0.1$

Estimated causal effect = $a-c = 0.1$

No confounding

Male

	Population D		Population d	
Outcome	Drug (actual)	Not drug (counter)	Drug (counter)	Not drug (actual)
Y=1	15	2	5	5
Y=0	35	8	65	15
	a=0.3	b=0.2		c=0.25

True = $a-b = 0.1$

Estimated = $a-c = 0.05$

Confounding

Female

	Population D		Population d	
Outcome	Drug (actual)	Not drug (counter)	Drug (counter)	Not drug (actual)
Y=1	15	18	25	15
Y=0	35	72	5	65
	a=0.3	b=0.2		0.1875

True = $a-b = 0.1$

Estimated = $a-c = 0.1125$

Confounding

Non-Collapsibility without Confounding

Population D				
	Drug (factual)		Not drug (counterfactual)	
Covariate	Y=1	Y=0	Y=1	Y=0
Z=1	80	20	60	40
Z=0	40	60	20	80
Total	120	80	80	120

True causal effect | Z=1: $0.8 / 0.2 \div 0.6 / 0.4 = 2.67$

True causal effect | Z=0: $0.4 / 0.6 \div 0.2 / 0.8 = 2.67$

True causal effect ignoring Z: $0.6 / 0.4 \div 0.4 / 0.6 = 2.25$

Population d				
	Drug (factual)		Not drug (counterfactual)	
Covariate	Y=1	Y=0	Y=1	Y=0
Z=1			60	40
Z=0			20	80
Total			80	120

Estimated causal effect | Z=1: $0.8 / 0.2 \div 0.6 / 0.4 = 2.67$

Estimated causal effect | Z=0: $0.4 / 0.6 \div 0.2 / 0.8 = 2.67$

Estimated causal effect ignoring Z: $0.6 / 0.4 \div 0.4 / 0.6 = 2.25$

Collapsibility with Confounding

Population D				
	Drug (factual)		Not drug (counterfactual)	
Covariate	Y=1	Y=0	Y=1	Y=0
Z=1	80	20	60	40
Z=0	40	60	20	80
Total	120	80	80	120

True causal effect | Z=1: $0.8 / 0.2 \div 0.6 / 0.4 = 2.67$

True causal effect | Z=0: $0.4 / 0.6 \div 0.2 / 0.8 = 2.67$

True causal effect ignoring Z: $0.6 / 0.4 \div 0.4 / 0.6 = 2.25$

Population d				
	Drug (factual)		Not drug (counterfactual)	
Covariate	Y=1	Y=0	Y=1	Y=0
Z=1			60	40
Z=0			30	120
Total			90	160

Estimated causal effect | Z=1: $0.8 / 0.2 \div 0.6 / 0.4 = 2.67$

Estimated causal effect | Z=0: $0.4 / 0.6 \div 0.2 / 0.8 = 2.67$

Estimated causal effect ignoring Z: $0.6 / 0.4 \div 0.36 / 0.64 = 2.67$

the hills example

```
> hills
```

	dist	climb	time
Greenmantle	2.4	650	0.2680556
Carnethy	6.0	2500	0.8058333
Craig Dunain	6.0	900	0.5608333
Ben Rha	7.5	800	0.7600000
Ben Lomond	8.0	3070	1.0377778
Goatfell	8.0	2866	1.2202778
Bens of Jura	16.0	7500	3.4102778
Cairnpapple	6.0	800	0.6061111
Scolty	5.0	800	0.4958333
Traprain	6.0	650	0.6625000
Lairig Ghru	28.0	2100	3.2111111
Dollar	5.0	2000	0.7175000
Lomonds	9.5	2200	1.0833333
Cairn Table	6.0	500	0.7355556
Eildon Two	4.5	1500	0.4488889
Cairngorm	10.0	3000	1.2041667
Seven Hills	14.0	2200	1.6402778
Knock Hill	3.0	350	1.3108333
Black Hill	4.5	1000	0.2902778
Creag Beag	5.5	600	0.5427778

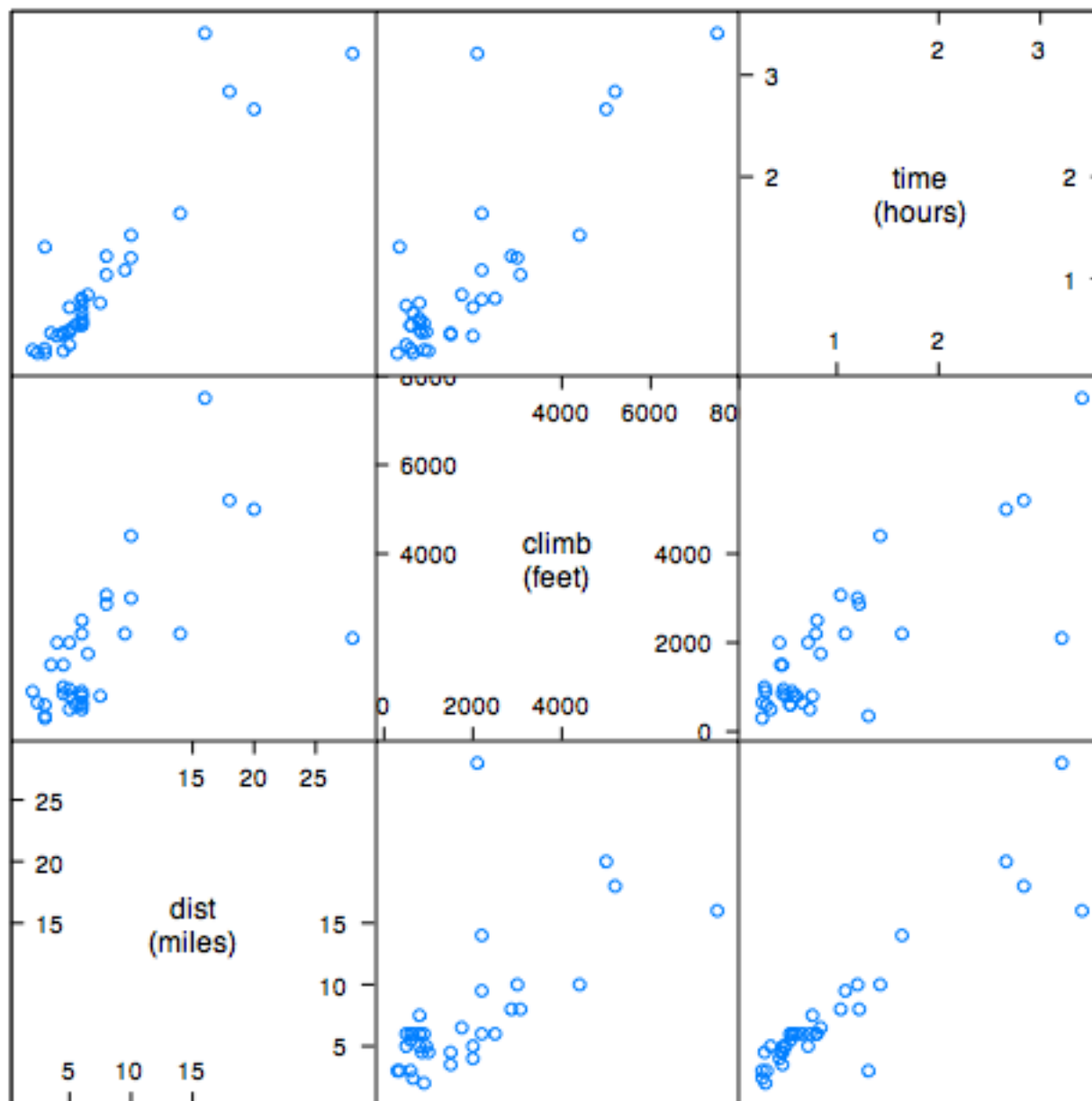
begin with scatterplot matrices...

```
library(lattice)
splom(~hills, cex.labels=1.2,
      varnames=c("dist\n(miles)", "climb\n(feet)", "time\n(hours)"))

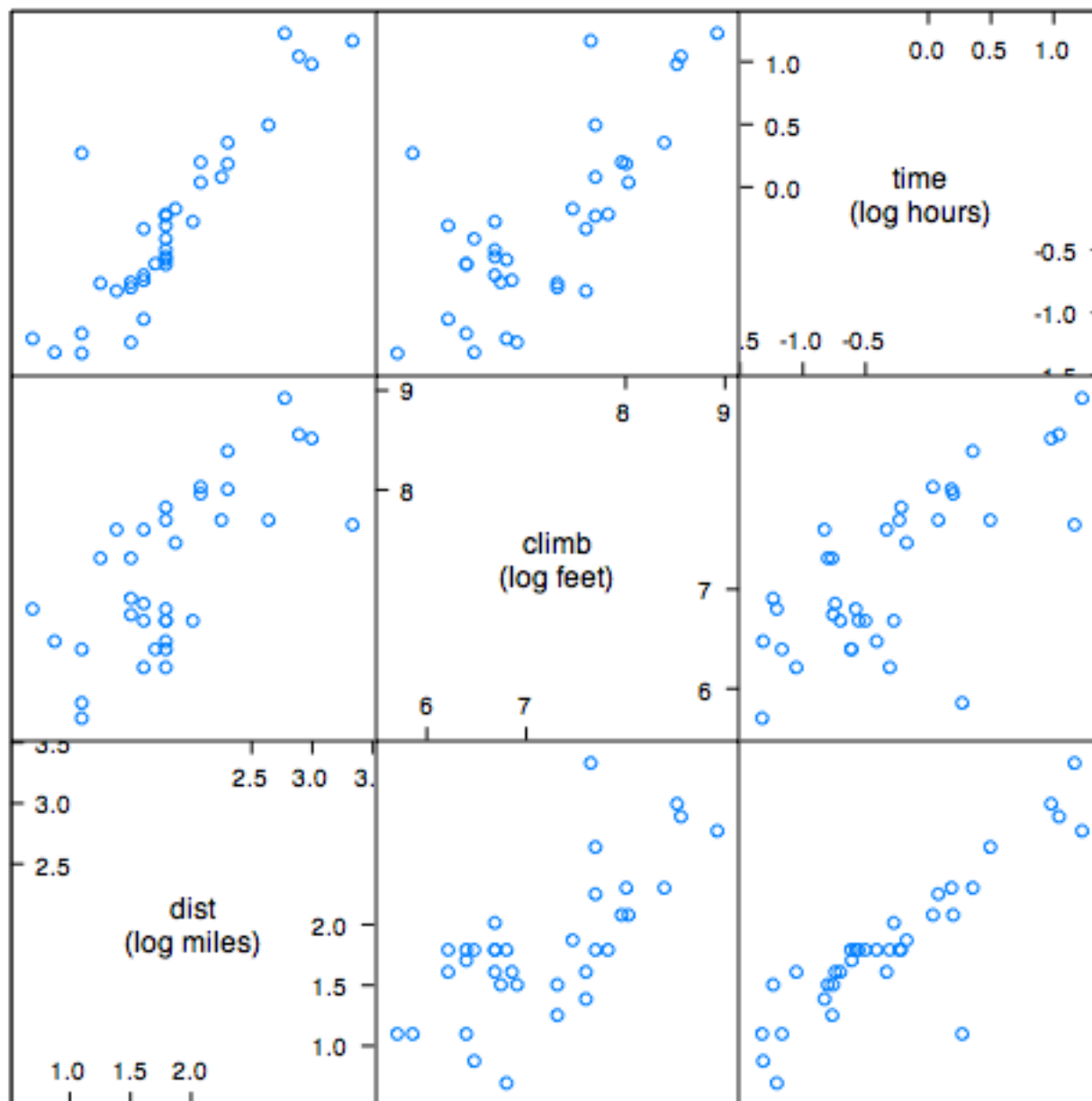
splom(~log(hills), cex.labels=1.2,
      varnames=c("dist\n(log miles)", "climb\n(log feet)",
                  "time\n(log hours)"))
```

why log?

- perhaps expect prediction error to be a fraction of the predicted time (think: prediction of 15 minutes as against a prediction of 3 hours)
- long tails
- marathon record not 26 X 4 minutes...



Scatter Plot Matrix



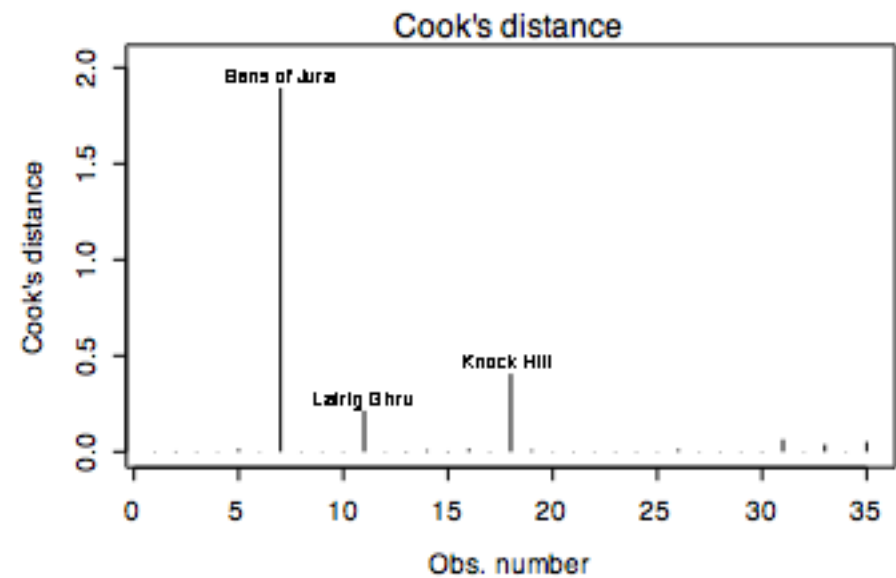
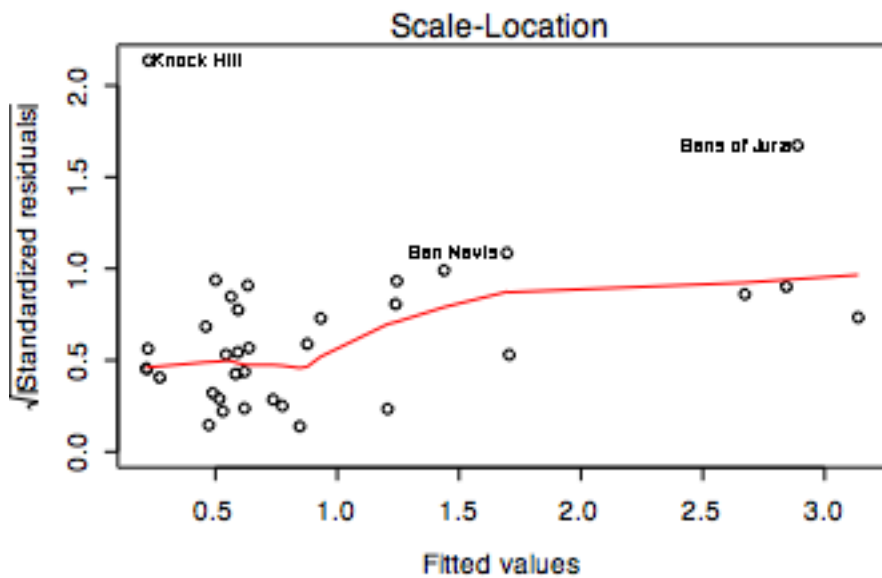
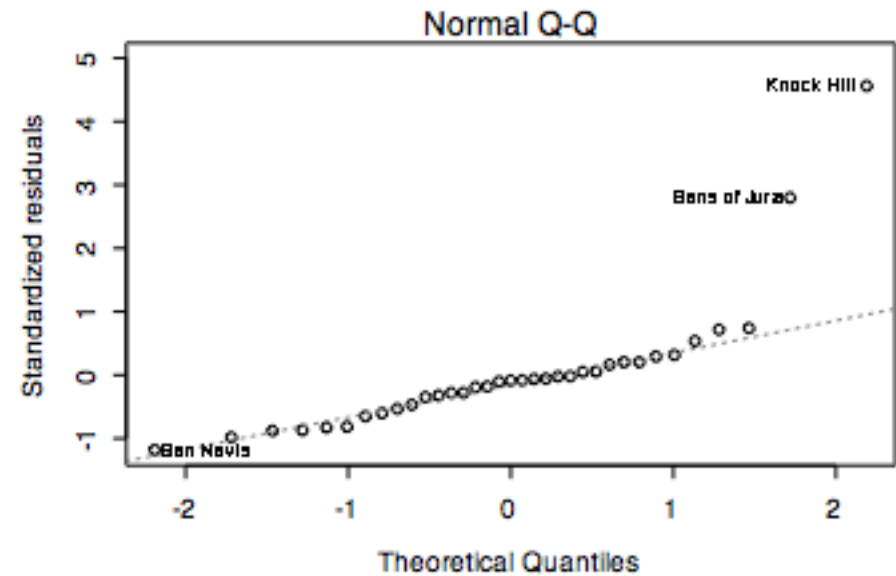
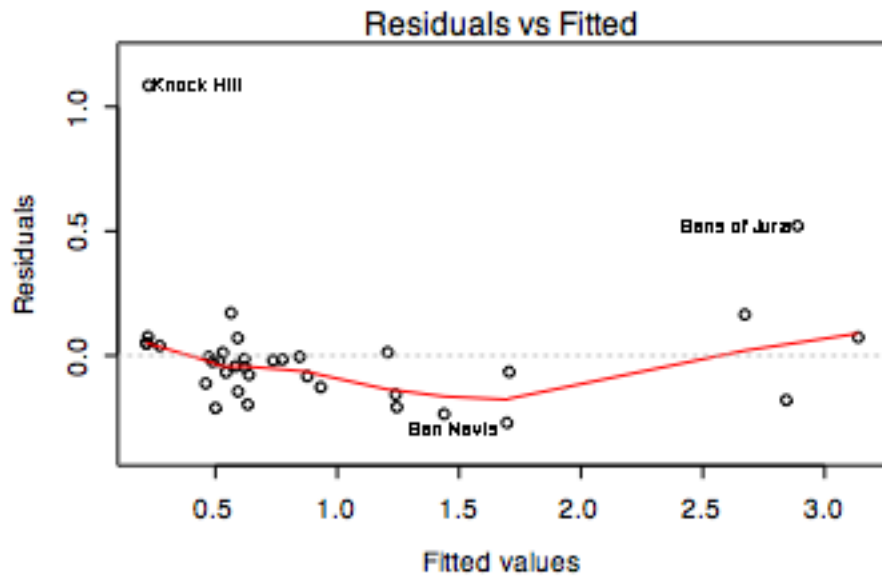
Scatter Plot Matrix

```
plot(hills$dist,hills$time)
identify(hills$dist,hills$time)
```

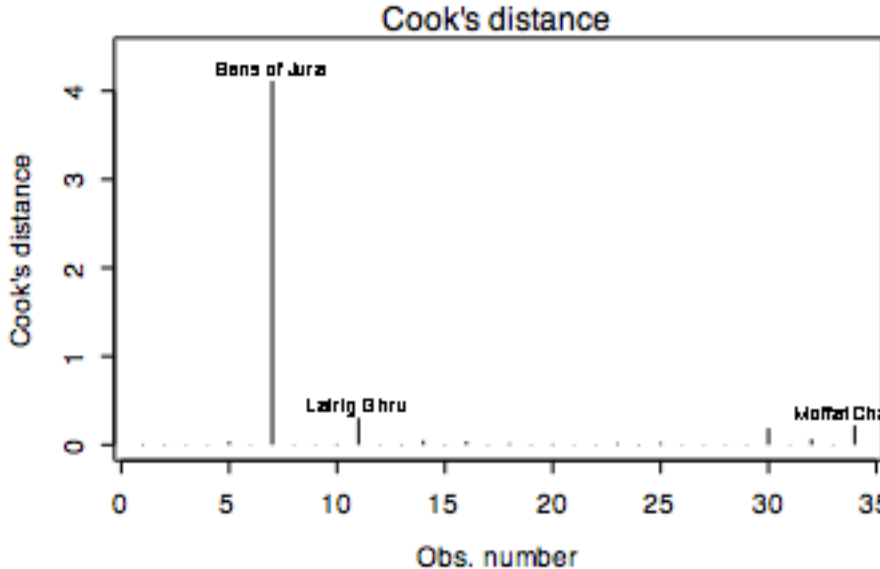
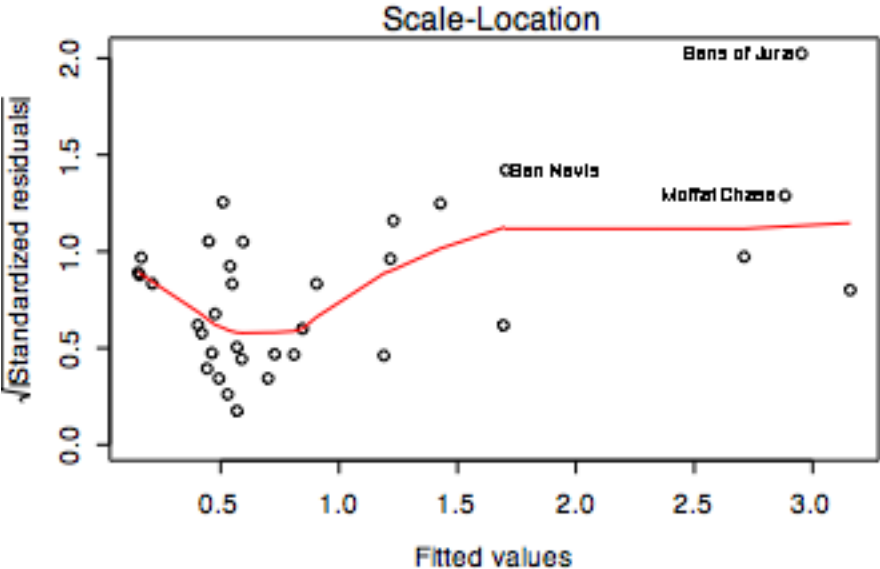
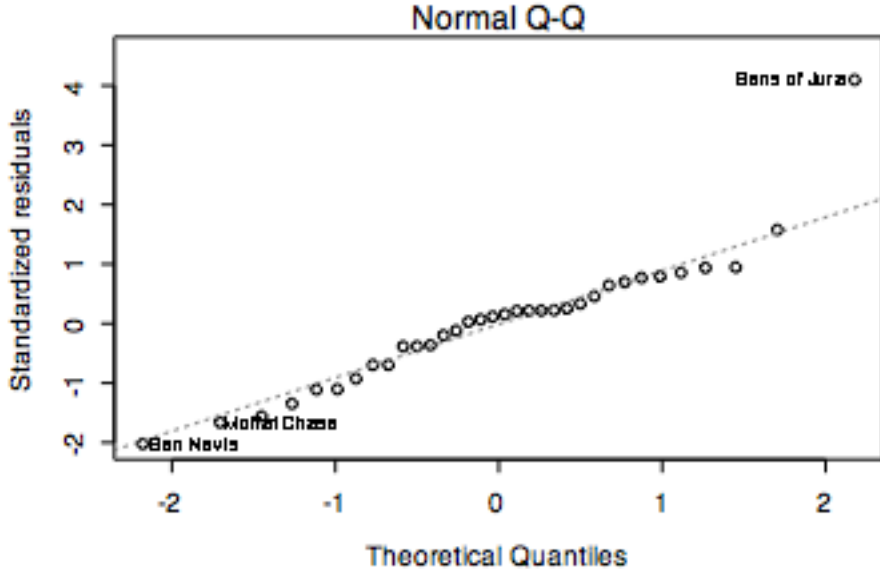
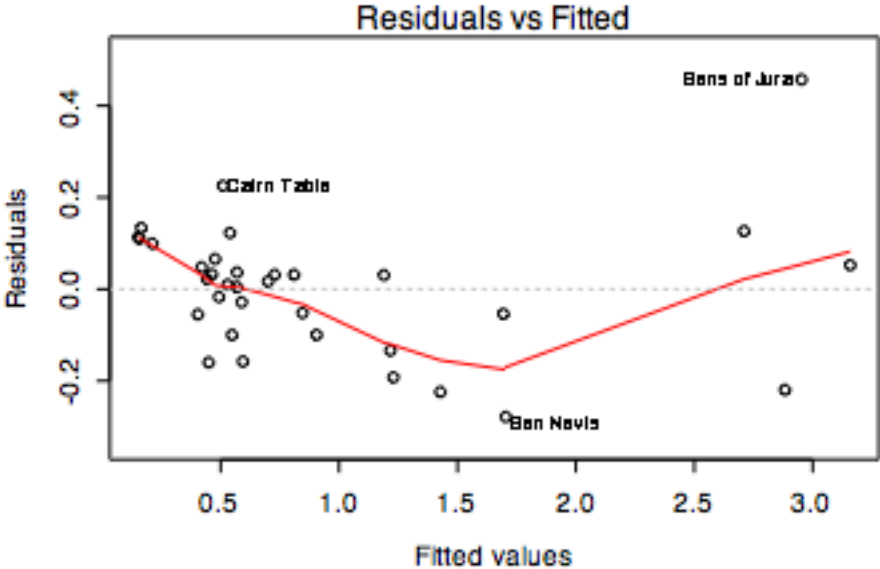
```
> hills[18,]
           dist climb      time
Knock Hill     3   350 1.310833
```

actually an error - should be 0.31!

```
plot(lm(time~dist+climb, data=hills), which=1:4)
```



```
plot(lm(time~dist+climb,data=hills,subset=-18),which=1:4)
```



how about an interaction?

```
logHills <- log(hills)
names(logHills) <- c("logDist", "logClimb", "logTime")

hillsInt.lm <- lm(logTime~logDist*logClimb,
data=logHills,subset=-18)
summary(hillsInt.lm)
par(mfrow=c(2,2))
plot(hillsInt.lm,which=1:4)
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-2.47552	0.96528	-2.565	0.0156 *
logDist	0.16854	0.49129	0.343	0.7339
logClimb	0.06724	0.13540	0.497	0.6231
logDist:logClimb	0.09928	0.06530	1.520	0.1389

↑
???

```
> summary(lm(logTime~logDist+logClimb,data=logHills,subset=-18))
```

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-3.88155	0.28263	-13.734	1.01e-14	***
logDist	0.90924	0.06500	13.989	6.16e-15	***
logClimb	0.26009	0.04839	5.375	7.33e-06	***

Stack Loss Example

```
> stackloss
  Air.Flow Water.Temp Acid.Conc. stack.loss
1      80         27         89         42
2      80         27         88         37
3      75         25         90         37
4      62         24         87         28
5      62         22         87         18
6      62         23         87         18
7      62         24         93         19
8      62         24         92         20
```

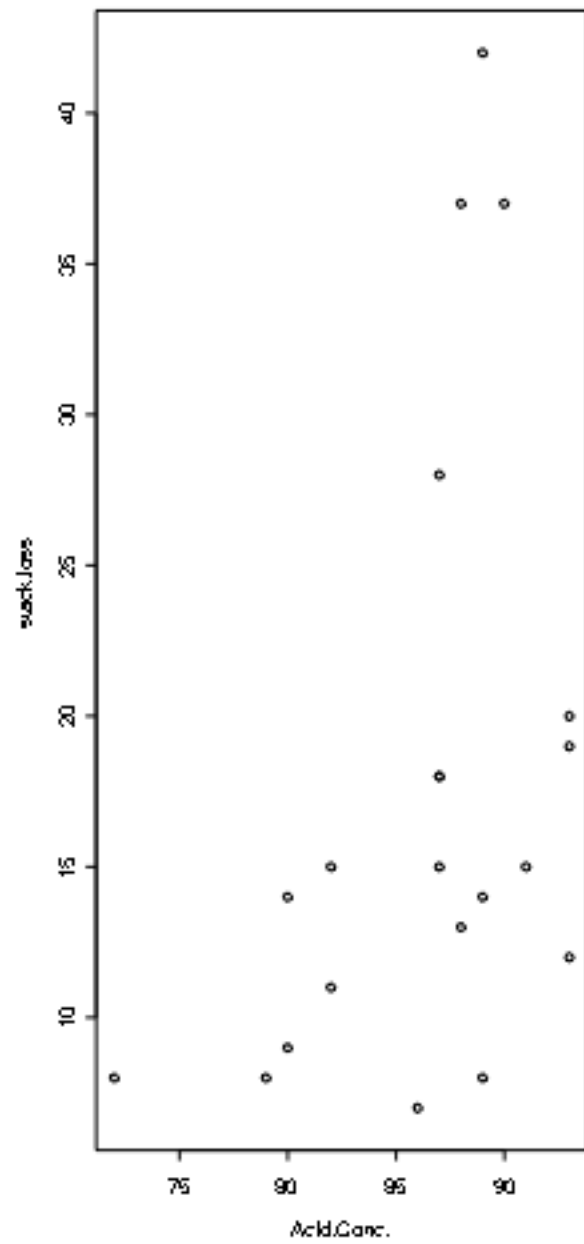
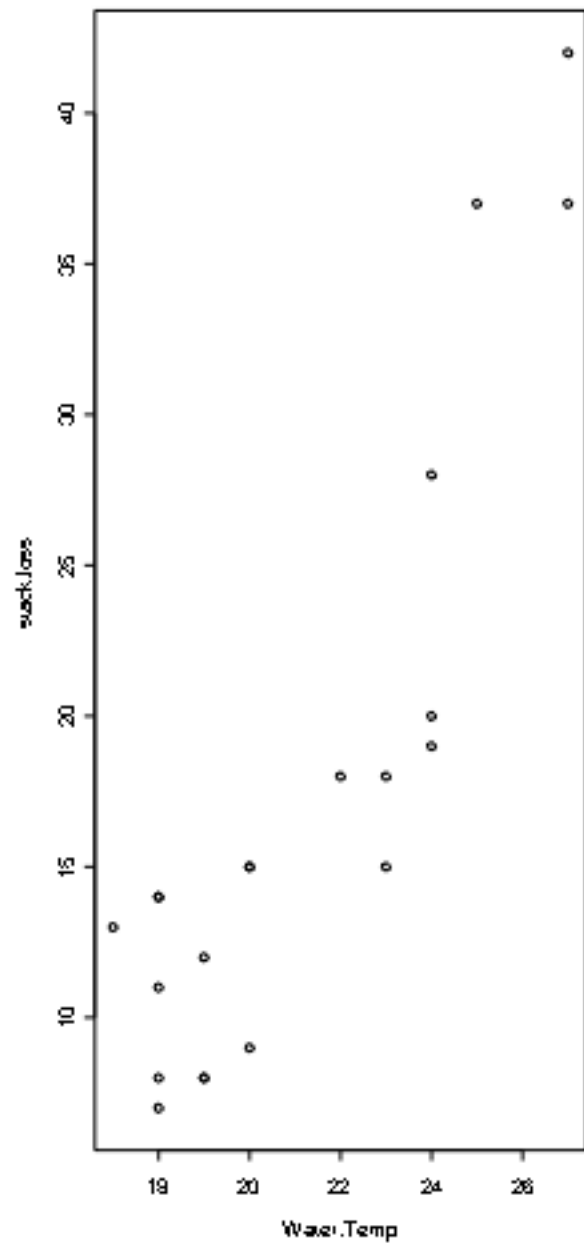
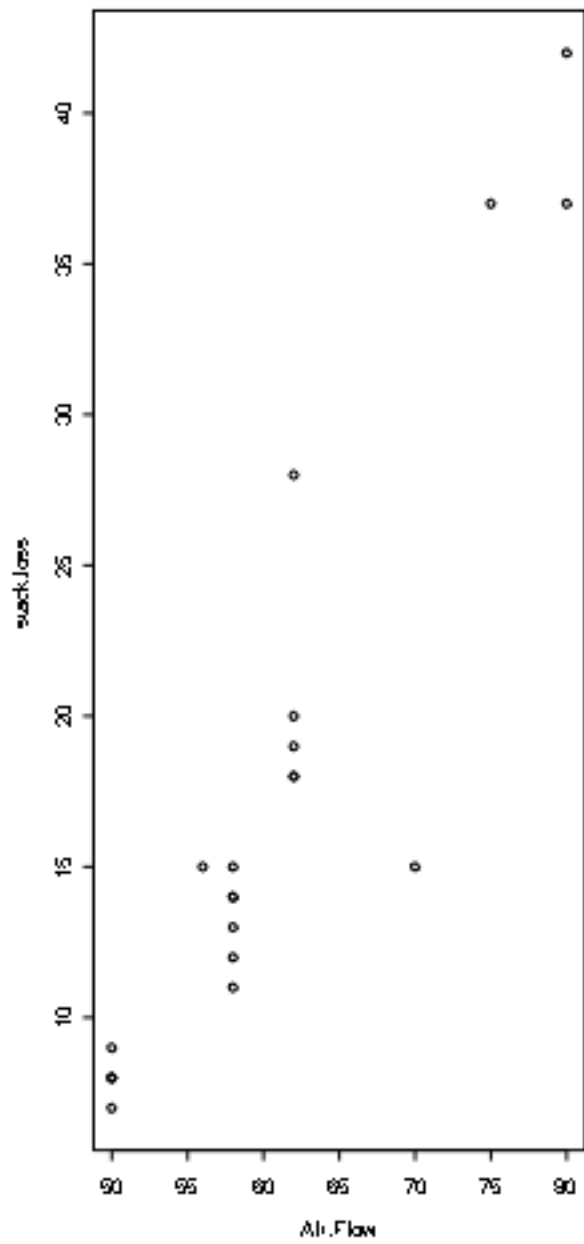
Oxidation of Ammonia to Nitric Acid on Successive Days

x1 airflow to the plant

x2 temperature of the cooling water

x3 concentration of nitric acid in the absorbing liquid

y 10 X the percentage of ingoing ammonia that is lost as unabsorbed nitric acids



```
> m1 <- lm(stack.loss~Air.Flow+Water.Temp+Acid.Conc.)
> summary(m1)
```

Call:

```
lm(formula = stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-7.2377	-1.7117	-0.4551	2.3614	5.6978

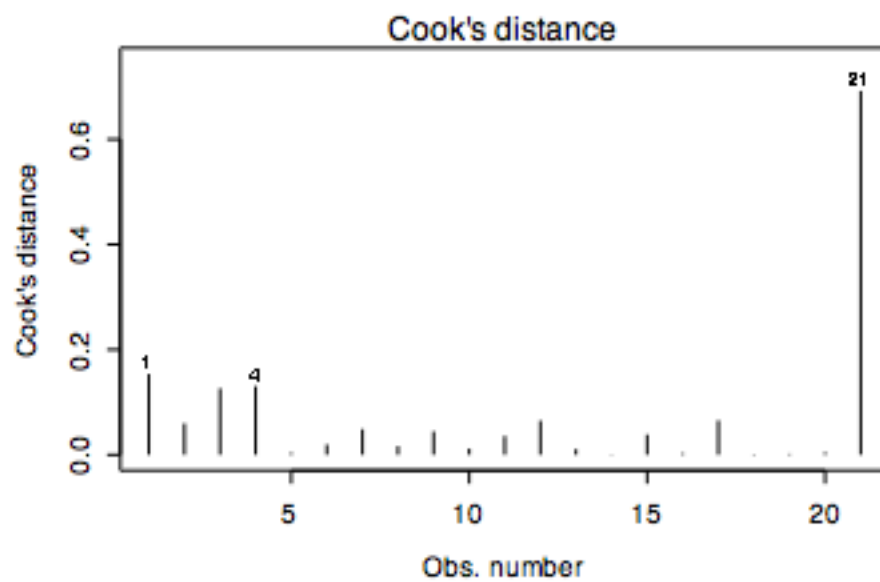
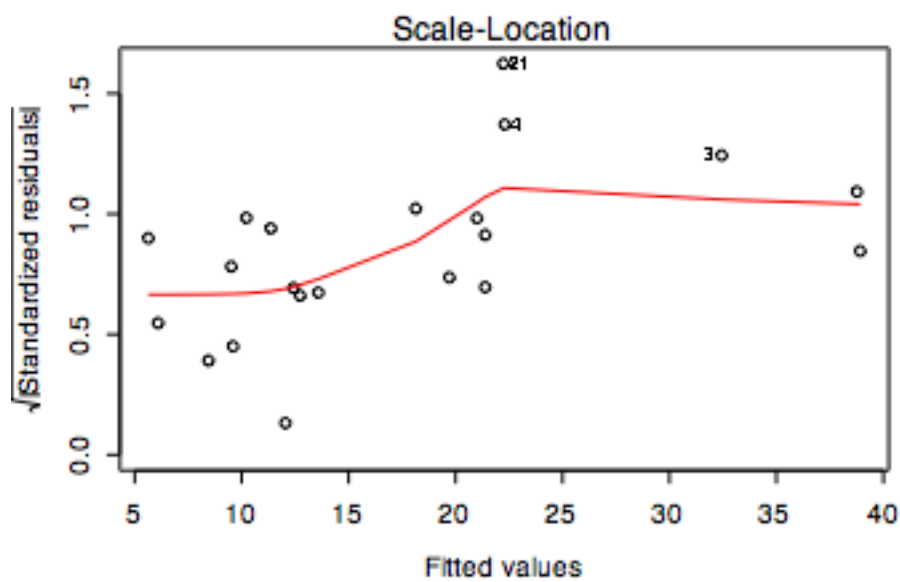
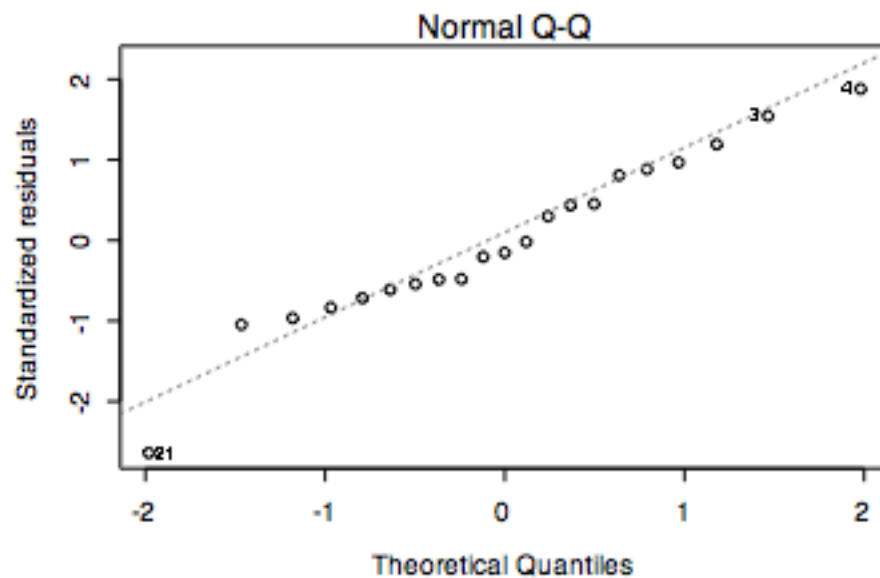
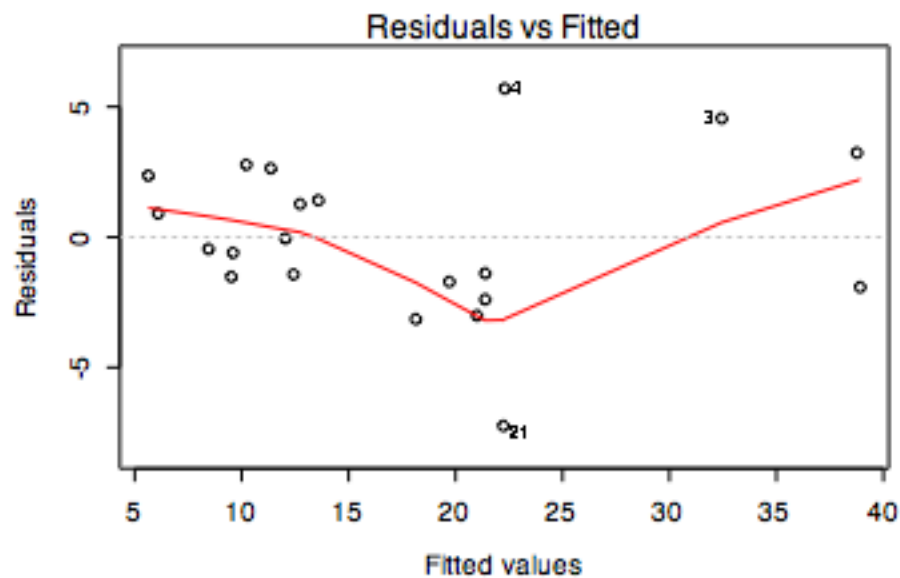
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-39.9197	11.8960	-3.356	0.00375	**
Air.Flow	0.7156	0.1349	5.307	5.8e-05	***
Water.Temp	1.2953	0.3680	3.520	0.00263	**
Acid.Conc.	-0.1521	0.1563	-0.973	0.34405	

Residual standard error: 3.243 on 17 degrees of freedom

Multiple R-Squared: 0.9136, Adjusted R-squared: 0.8983

F-statistic: 59.9 on 3 and 17 DF, p-value: 3.016e-09



Some odd patterns and outlying points. How come?

1. Some points may be entirely erroneous
2. We might not have the best functional form
3. The random error might not be normal
4. Some points may reflect changing conditions - perhaps the plant requires time to reach equilibrium after significant input changes

focus on 1 and 2 to begin with. Lets try dropping #21.

17	50	19	72	8
18	50	19	79	8
19	50	20	80	9
20	56	20	82	15
21	70	20	91	15

21 is the last day of measurement
big input change, no change in output...

```
> m2 <- lm(stack.loss~Air.Flow+Water.Temp+Acid.Conc.,subset=-21)
> summary(m2)
```

Call:

```
lm(formula = stack.loss ~ Air.Flow + Water.Temp + Acid.Conc.,
    subset = -21)
```

Residuals:

Min	1Q	Median	3Q	Max
-3.0449	-2.0578	0.1025	1.0709	6.3017

Coefficients:

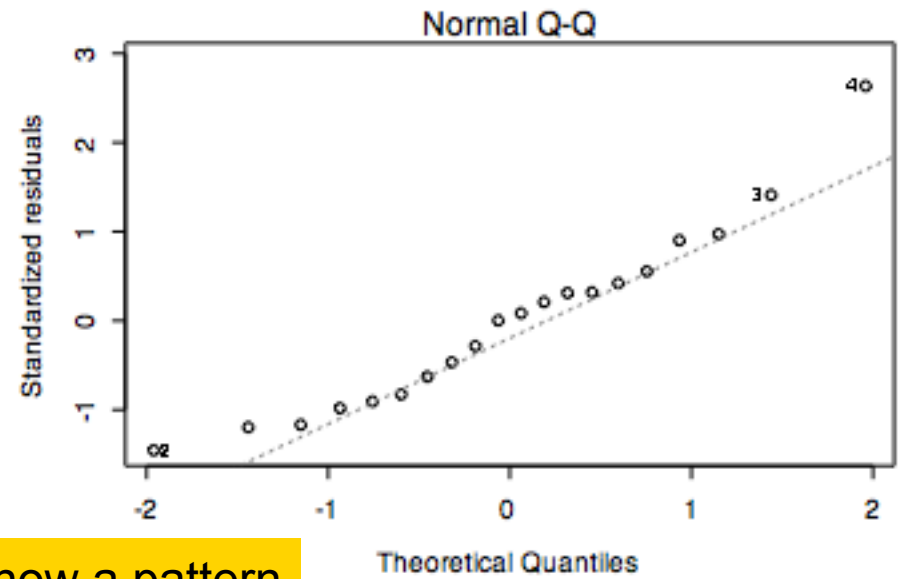
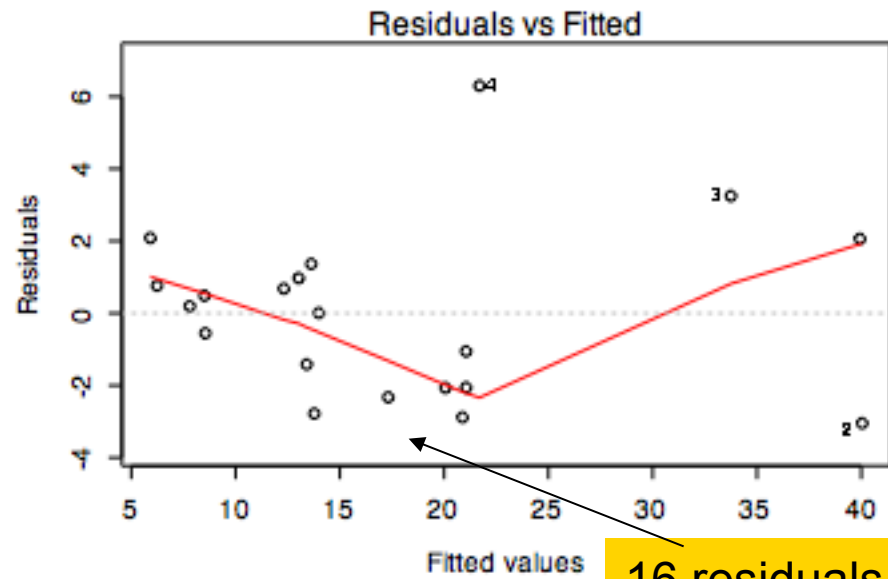
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-43.7040	9.4916	-4.605	0.000293	***
Air.Flow	0.8891	0.1188	7.481	1.31e-06	***
Water.Temp	0.8166	0.3250	2.512	0.023088	*
Acid.Conc.	-0.1071	0.1245	-0.860	0.402338	

Residual standard error: 2.569 on 16 degrees of freedom

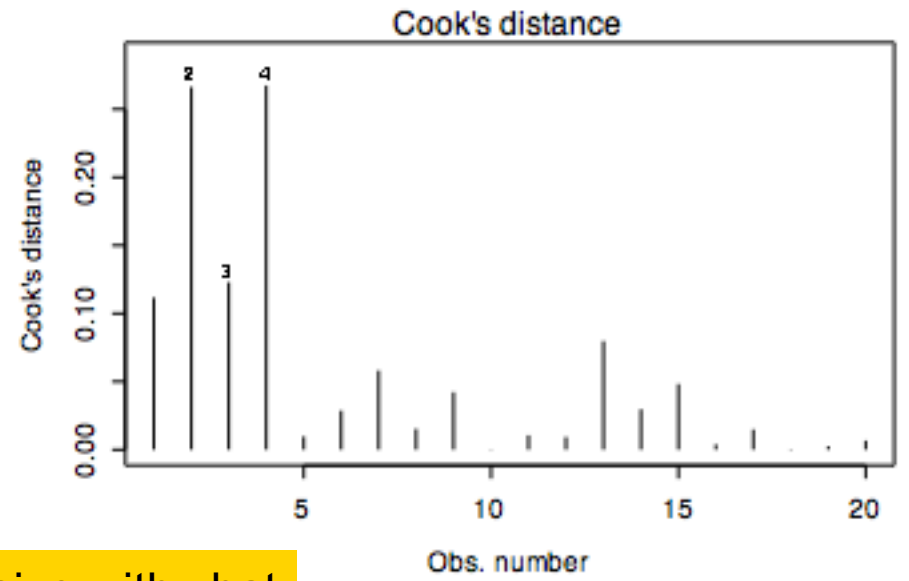
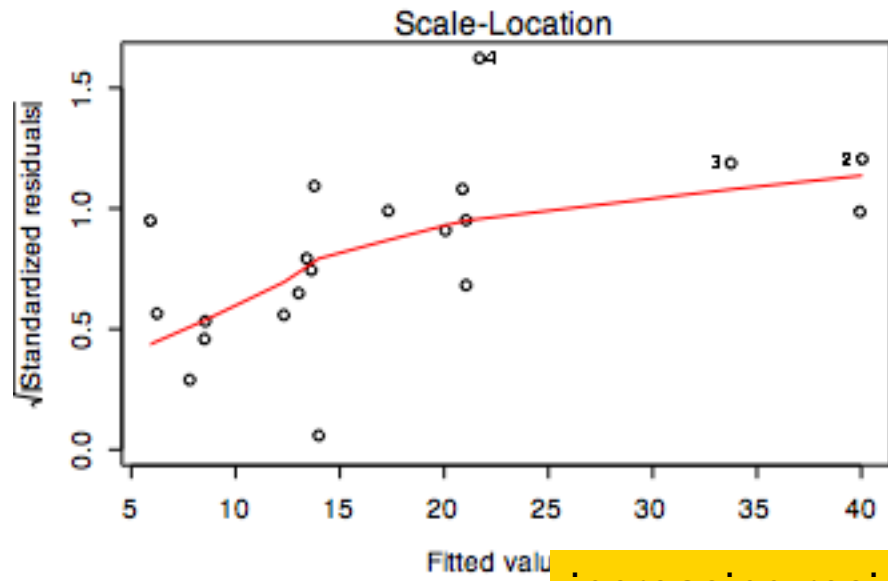
Multiple R-Squared: 0.9488, Adjusted R-squared: 0.9392

F-statistic: 98.82 on 3 and 16 DF, p-value: 1.541e-10

normal plot now a bit skew



16 residuals show a pattern



increasing resid size with yhat

try log(y)...

```
> m3 <- lm(log(stack.loss)~Air.Flow+Water.Temp+Acid.Conc.)  
> summary(m3)
```

Call:

```
lm(formula = log(stack.loss) ~ Air.Flow + Water.Temp + Acid.Conc.)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-0.29269	-0.09734	-0.03937	0.12290	0.36558

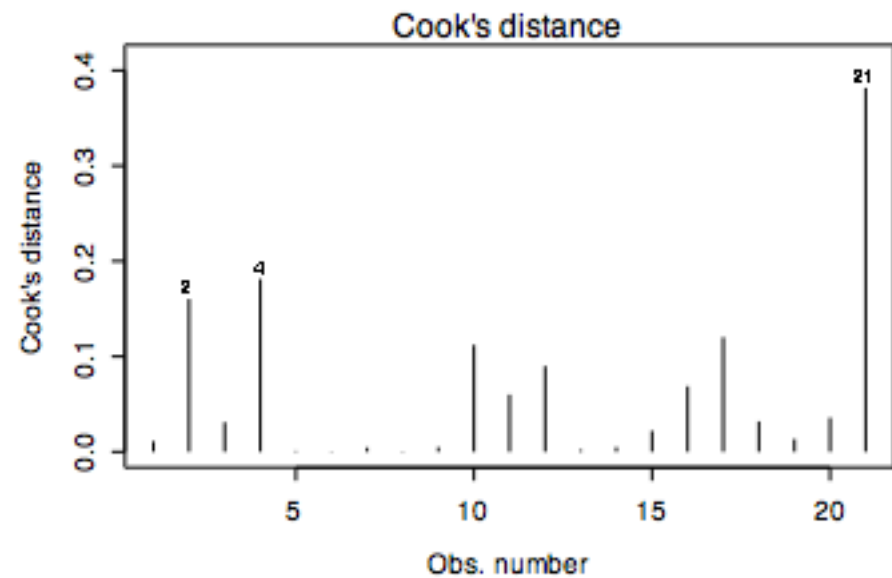
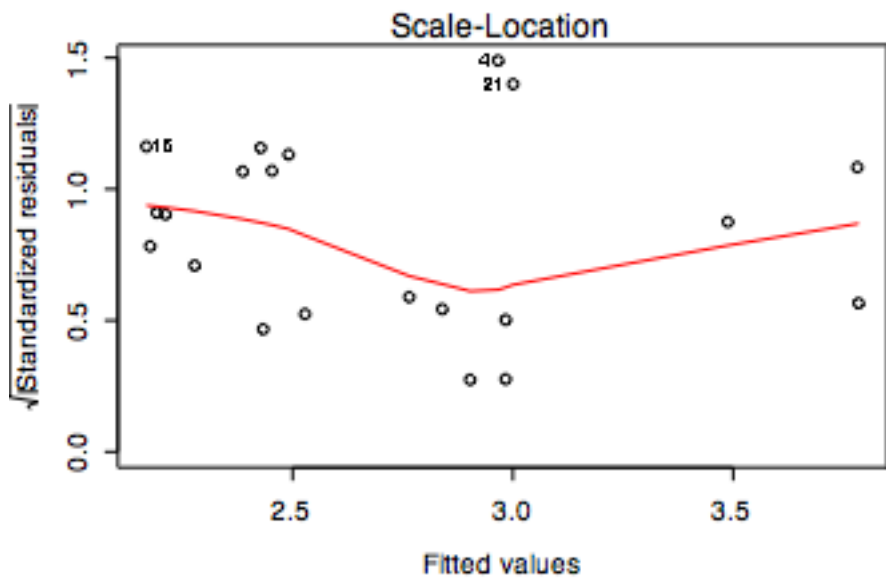
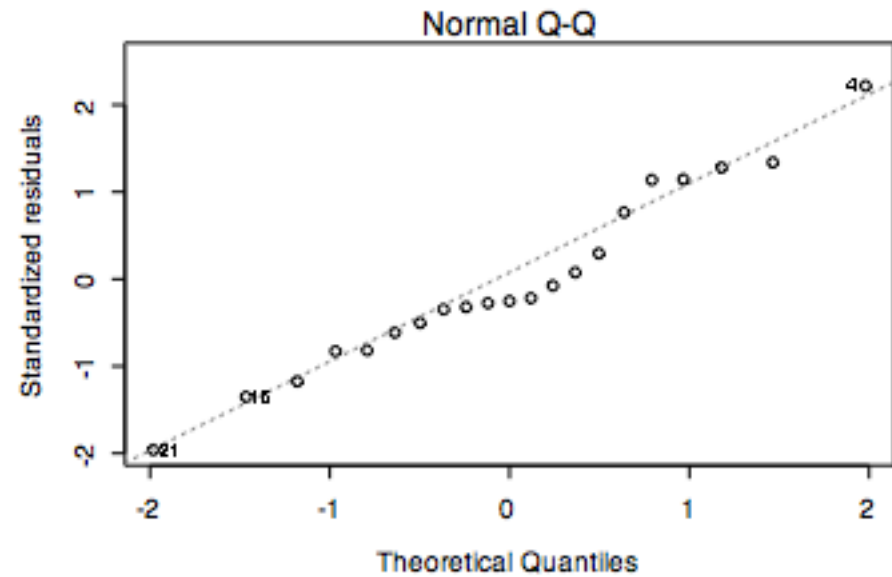
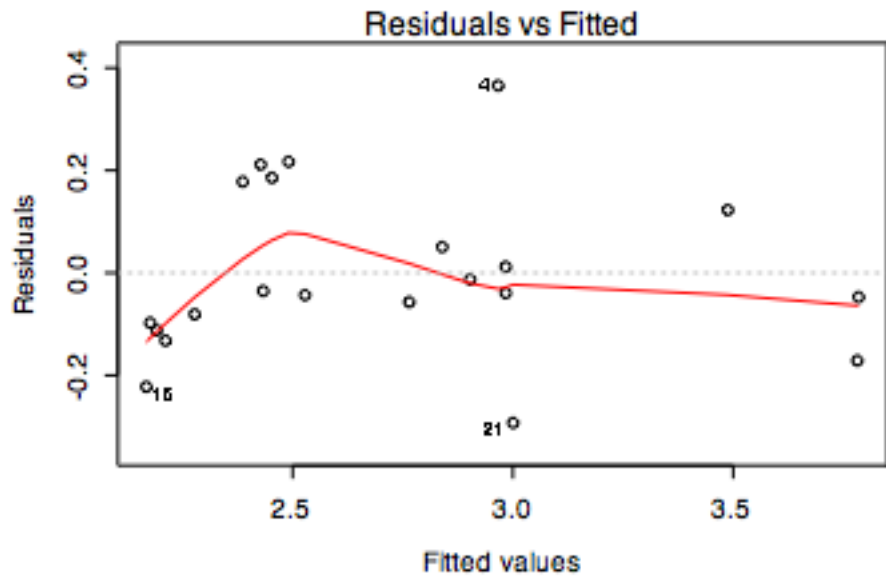
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-0.948729	0.647721	-1.465	0.161247	
Air.Flow	0.034565	0.007343	4.707	0.000203	***
Water.Temp	0.063465	0.020038	3.167	0.005632	**
Acid.Conc.	0.002864	0.008510	0.337	0.740566	

Residual standard error: 0.1766 on 17 degrees of freedom

Multiple R-Squared: 0.9033, Adjusted R-squared: 0.8862

F-statistic: 52.92 on 3 and 17 DF, p-value: 7.811e-09



R^2 is down but the diagnostic plots look pretty good...

Acid Concentration has shown negligible influence in all three models. Drop it?

Three different modeling actions under consideration:

A: Observation 21 in or out

B: y or $\log y$ as the response

C: Acid Concentration in or out

#	Obs. 21	log	acid conc.	R^2	SSE/df	Normal Plot	Residual versus yHat
1	in	y	in	0.91	10.5	21 low	1,2,3,4 outside
2	out	y	in	0.95	6.6	Curved	1,2,3,4 outside
3	in	log y	in	0.90	0.0059	OK	OK
4	in	y	out	0.91	10.5	21 low	1,2,3,4 outside
5	out	y	out	0.95	6.5	Curved	1,2,3,4 outside
6	in	log y	out	0.90	0.0056	21 high	OK
7	out	log y	in	0.92	0.0048	4 high	OK
8	out	log y	out	0.92	0.0046	4 high	OK

seems clear we should drop #21 and acid conc.

if $\log(y)$ is the "correct" response, the error should increase linearly with y . Lets check this...

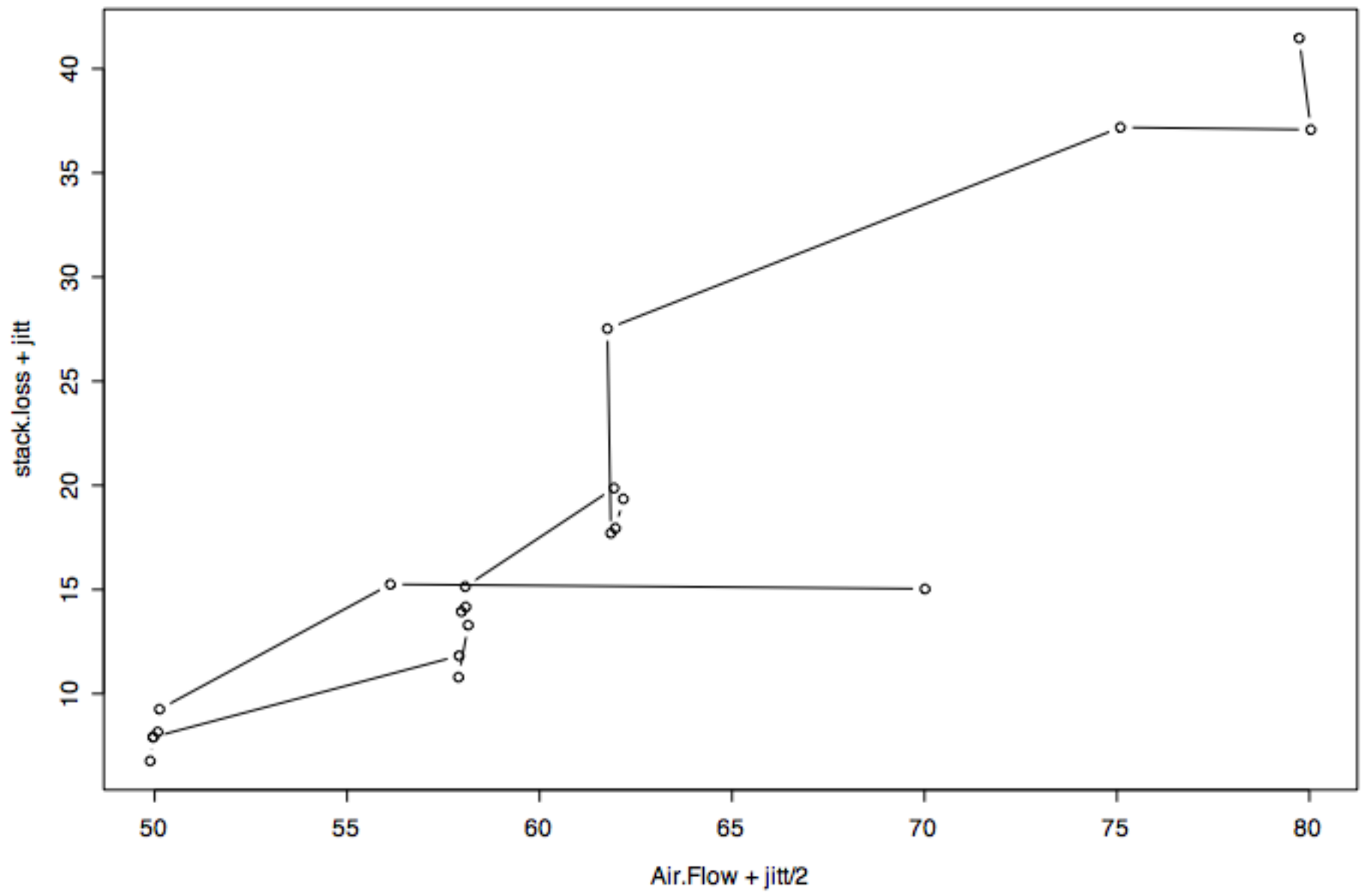
5	62	22	87	18
6	62	23	87	18
7	62	24	93	19
8	62	24	93	20
9	58	23	87	15
10	58	18	80	14
11	58	18	89	14
12	58	17	88	13
13	58	18	82	11
14	58	19	93	12
15	50	18	89	8
16	50	18	86	7
17	50	19	72	8
18	50	19	79	8
19	50	20	80	9
20	56	20	82	15

"near replicates"

Observations	SSE	df	MSE	s	Yhat	s/Y
5,6,7,8	2.8	3	0.93	1.0	21	5
10, 11,12, 13,14	6.8	4	1.70	1.3	12	11
15,16, 17,18, 19	2.0	4	0.50	0.7	8	9
Pooled	11.6	11	1.05	1.02		

no evidence that error increase with y

Also, MSE here is much smaller than, e.g., model 5 (MSE=6.5)
 Something is not right! (can use an F-test here)



it seems that when air flow exceeds 60, the plant takes about a day to come to equilibrium

"line-out" is the term used by plant operators

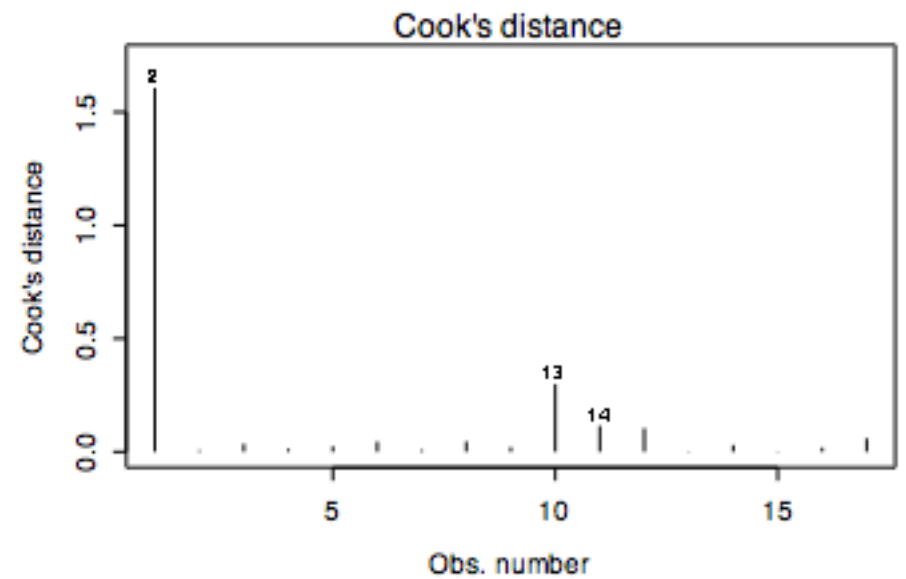
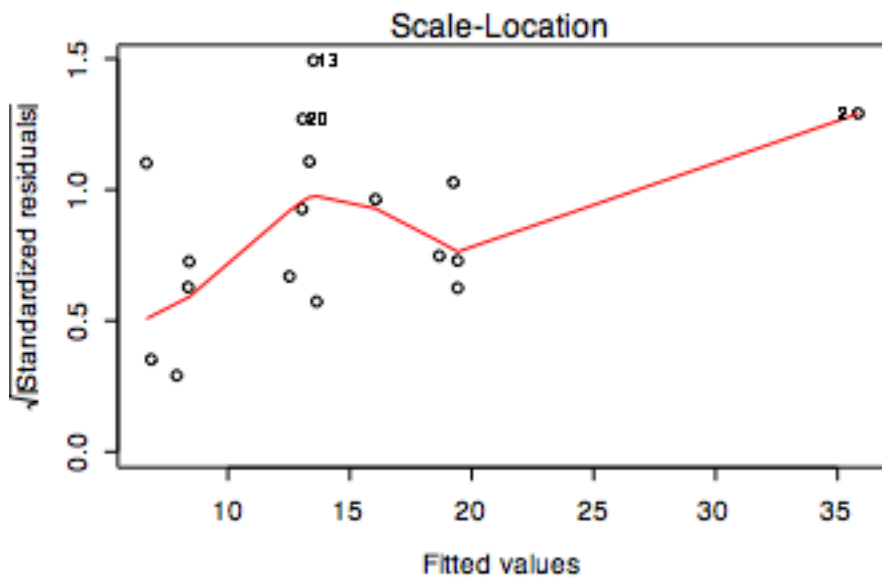
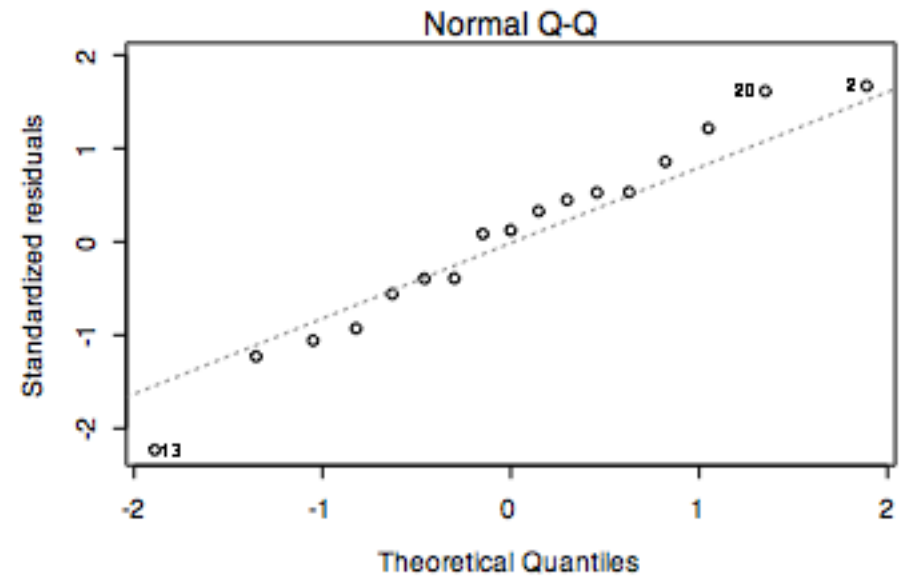
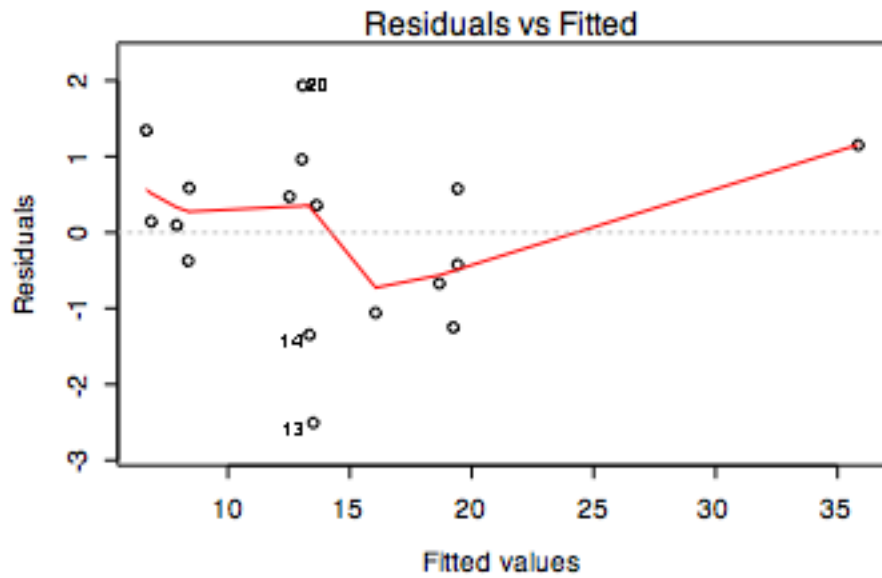
This would suggest permanently dropping points 1, 3, 4, and 21 since they correspond to transient states

Now revisit the log y and acid concentration issues...

1, 3, 4, and 21 removed

#	Obs. 21	log	acid conc.	R^2	SSE/df	Normal Plot	Residual versus yHat
9	out	y	in	0.975	1.6	20 low	curvature?
10	out	y	out	0.973	1.6	OK	curvature?
11	out	log y	in	0.92	0.0032	20 high	curvature?
12	out	log y	out	0.92	0.0031	20 high	curvature?

model 9



drop acid conc. again
try adding airflow^2

```
lm(formula = stack.loss ~ Air.Flow + Water.Temp + I(Air.Flow^2),  
    subset = c(-1, -3, -4, -21))
```

Residuals:

Min	1Q	Median	3Q	Max
-2.0177	-0.6530	-0.1252	0.5101	2.3429

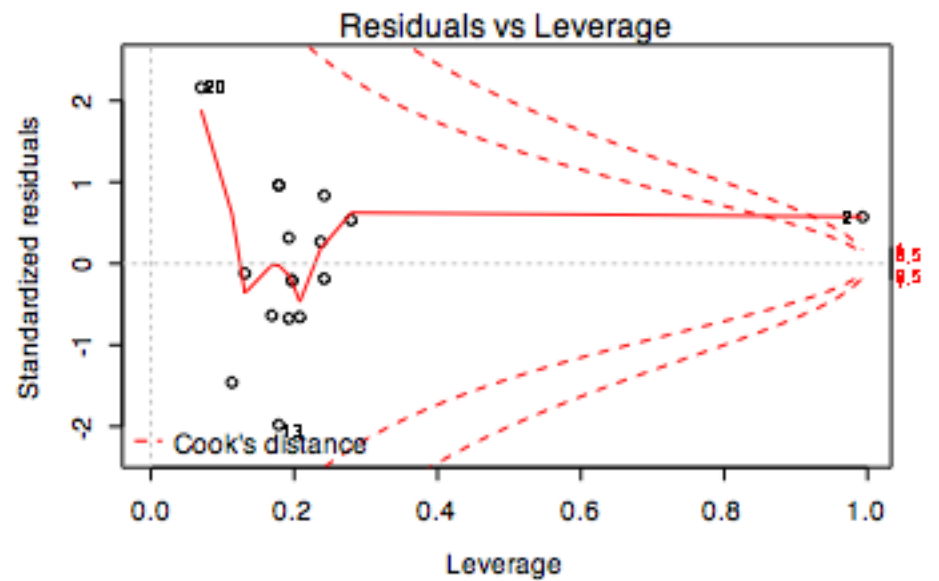
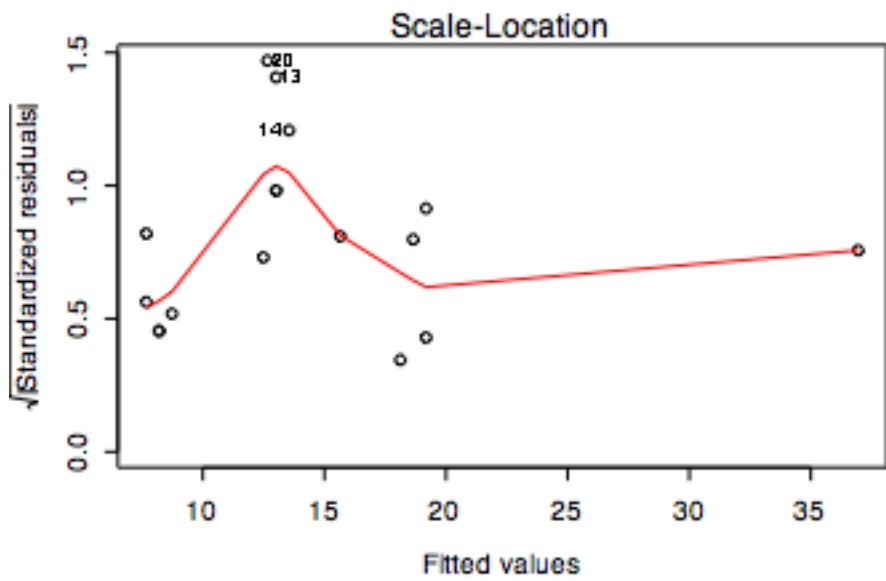
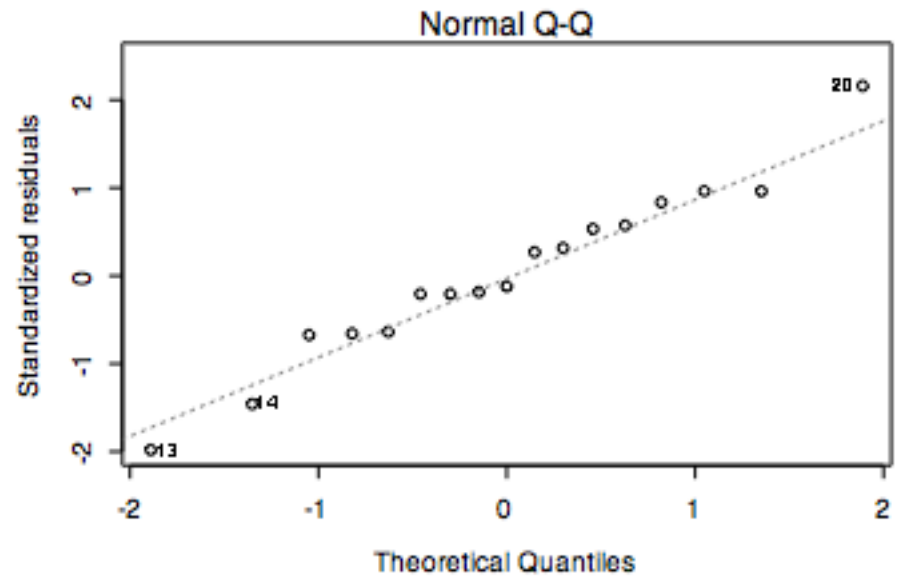
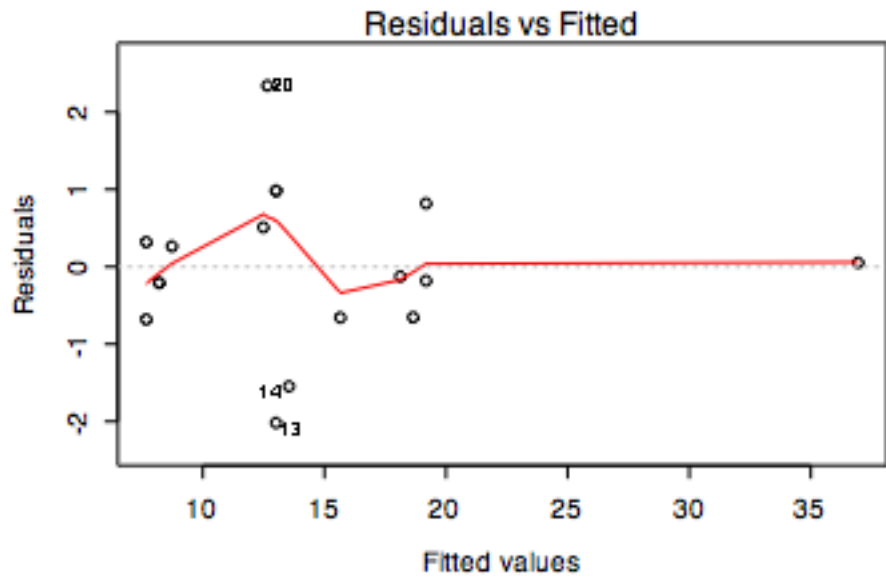
Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-15.409290	12.602668	-1.223	0.24315
Air.Flow	-0.069142	0.398419	-0.174	0.86490
Water.Temp	0.527804	0.150079	3.517	0.00379 **
I(Air.Flow^2)	0.006818	0.003178	2.145	0.05139 .

Residual standard error: 1.125 on 13 degrees of freedom

Multiple R-Squared: 0.9799, Adjusted R-squared: 0.9752

F-statistic: 210.8 on 3 and 13 DF, p-value: 2.854e-11



probably makes no sense to go further

Could try a factorial study of x_1^2 , x_2^2 , x_1x_2 , $\log y$, etc.

SSE/df is now 1.26 compared with the "minimum" 1.05

model selection in linear regression

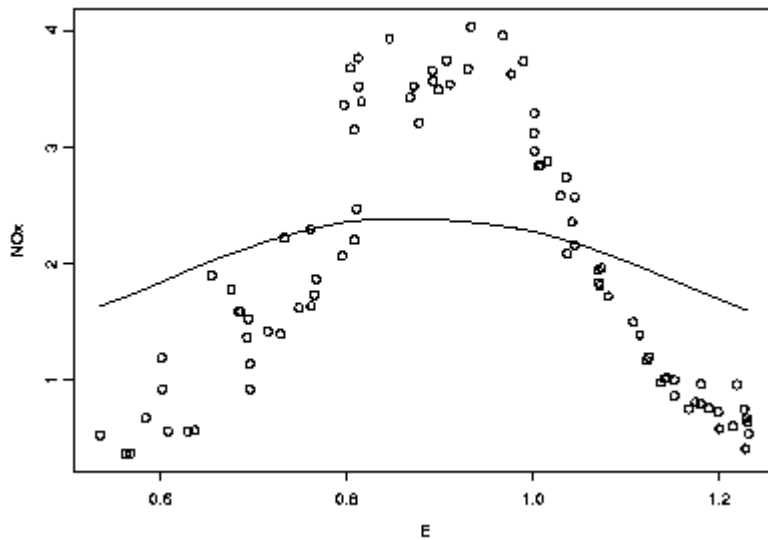
basic problem: how to choose between competing linear regression models

model too small: "underfit" the data; poor predictions;
high bias; low variance

model too big: "overfit" the data; poor predictions;
low bias; high variance

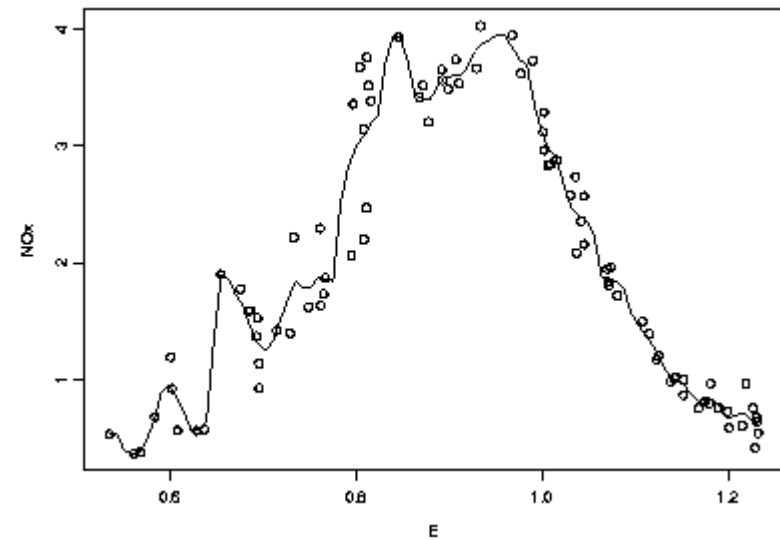
model just right: balance bias and variance to get
good predictions

Bias-Variance Tradeoff



High Bias - Low Variance

Score function should
embody the compromise



Low Bias - High Variance

“overfitting” - modeling
the random component

model selection in regression has two facets:

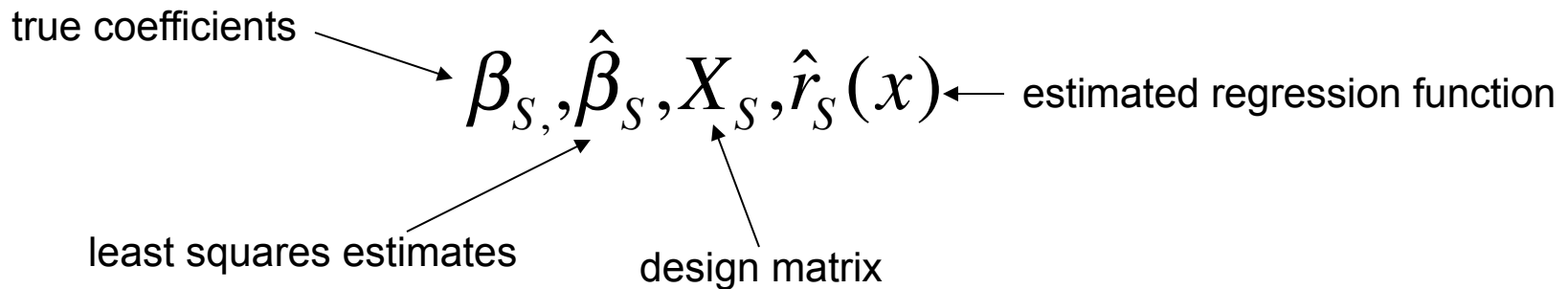
1. assign a score to each model
2. search for models with good scores

linear regression model scores

consider the problem of selecting a "good" subset of k candidate predictors

$$S \subseteq \{1, \dots, k\}$$

$$\mathcal{X}_S = \{X_j : j \in S\}$$




$$\hat{Y}_i(S) = \hat{r}_S(X_i)$$

prediction risk:

$$R(S) = \sum_{i=1}^n E_{y, Y^*} (\hat{Y}_i(S) - Y_i^*)^2$$

value of future observation at X_i



goal: pick the model that minimizes $R(S)$

training error:

$$\hat{R}_{TR}(S) = \sum_{i=1}^n (\hat{Y}_i(S) - Y_i)^2$$

bad estimate of risk!



Theorem: The training error is a downward-biased estimate of the prediction risk:

$$E\left(\hat{R}_{TR}(S)\right) < R(S)$$

tends to be
large when
the model is
large

$$\text{bias}\left(\hat{R}_{TR}(S)\right) = E_y\left(\hat{R}_{TR}(S)\right) - R(S) = -2 \sum_{i=1}^n \text{Cov}\left(\hat{Y}_i, Y_i\right)$$

for linear models with $|S|$ predictors:

$$\sum_{i=1}^n \text{Cov}\left(\hat{Y}_i, Y_i\right) = |S| \sigma_{\varepsilon}^2$$

obvious thing to do is estimate the bias and adjust!

$$C_p = \frac{\hat{R}_{TR}(S)}{1} + \frac{2|S|\hat{\sigma}_\varepsilon^2}{1}$$

often estimated from
the "full" model

how well the model fits the
training data; smaller is better

complexity penalty; bigger
model, bigger penalty

"Mallows C_p statistic"

Akaike Information Criterion is one alternative:

$$\text{AIC} = l_S - |S|$$

where l_S is the maximized log-likelihood

(very similar to C_p in normal linear regression models)

- can use cross-validation to estimate prediction risk
- for linear regression, there are short cut formulae that can compute the CV estimate from a single (full) model fit

AIC in R is multiplied by -2 (so smaller is better)

$$\text{AIC}_R = -2l_S + 2|S|$$

for linear regression with normal errors, the log likelihood is:

$$-\frac{n}{2} \log 2\pi - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \|y - X\beta\|^2$$

plugging the MLE for σ^2 :

$$-\frac{n}{2} \log 2\pi - \frac{n}{2} \log \sigma^2 - \frac{1}{2\sigma^2} \text{RSS}$$

Thus, if β is known:

$$\text{AIC} = n \log 2\pi + n \log \sigma^2 + \frac{1}{\sigma^2} \text{RSS} + 2|S|$$

↑
constant

If β is unknown:

$$\text{AIC} = n \log(\text{RSS}/n) + 2|S| + \text{const}$$

Bayesian Information Criterion is one alternative:

$$\text{BIC} = l_S - \frac{|S|}{2} \log n$$

where l_S is the maximized log-likelihood

Bayesian interpretation: suitably normalized, BIC scores can be interpreted as approximate posterior model probabilities: $P(S_j | \text{Data})$

Bayesian Criterion

$$\begin{aligned} p(M_k | D) &\propto p(D | M_k) p(M_k) \\ &= p(M_k) \int p(D | \theta_k, M_k) p(\theta_k | M_k) d\theta_k \end{aligned}$$

- Typically impossible to compute analytically
- All sorts of Monte Carlo approximations

Savage-Dickey Density Ratio

- Suppose M_0 simplifies M_1 by setting one parameter (say q_1) to some constant (typically zero)
- If $p_1(q_2 | q_1 = 0) = p_0(q_2)$ then:

$$\frac{p(\text{data} | M_0)}{p(\text{data} | M_1)} = \frac{p(q_1 = 0 | M_1, \text{data})}{p(q_1 = 0 | M_1)}$$

Laplace Method for $p(D|M)$

$$\text{let } l(\theta) = \frac{\log(L(\theta))}{n} + \frac{\log p(\theta)}{n}$$

(i.e., the log of the integrand divided by n)

$$\text{then } p(D) = \int e^{nl(\theta)} d\theta$$

Laplace's Method:

$$p(D) \approx \int \exp[nl(\tilde{\theta}) - n(\theta - \tilde{\theta})^2 / (2\sigma^2)] d\theta$$

where $\sigma^2 = -1/l''(\tilde{\theta})$ and

$\tilde{\theta}$ is the posterior mode

Laplace cont.

$$p(D) = \int \exp[nl(\tilde{\theta}) - n(\theta - \tilde{\theta})^2 / (2\sigma^2)] d\theta$$
$$\approx \sqrt{2\pi\sigma n}^{-1/2} \exp\{nl(\tilde{\theta})\}$$

- Tierney & Kadane (1986, JASA) show the approximation is $O(n^{-1})$
- Using the MLE instead of the posterior mode is also $O(n^{-1})$
- Using the expected information matrix in $\boxed{\tilde{\theta}}$ is $O(n^{-1/2})$ but convenient since often computed by standard software
- Raftery (1993) suggested approximating $\tilde{\theta}$ by a single Newton step starting at the MLE
- Note the prior is explicit in these approximations

Monte Carlo Estimates of $p(D|M)$

$$p(D) = \int p(D | \theta) p(\theta) d\theta$$

Draw iid $\theta_1, \dots, \theta_m$ from $p(\theta)$:

$$\hat{p}(D) = \frac{1}{m} \sum_{i=1}^m p(D | \theta^{(i)})$$

In practice has large variance

Monte Carlo Estimates of $p(D|M)$ (cont.)

Draw iid $\theta_1, \dots, \theta_m$ from $p(\theta | D)$:

$$\hat{p}(D) = \frac{\sum_{i=1}^m w_i p(D | \theta^{(i)})}{\sum_{i=1}^m w_i}$$

“Importance
Sampling”

$$w_i = \frac{p(\theta^{(i)})}{p(\theta^{(i)} | D)} = \frac{\cancel{p(\theta^{(i)})} p(D)}{p(D | \theta^{(i)}) \cancel{p(\theta^{(i)})}}$$

Monte Carlo Estimates of $p(D|M)$ (cont.)

$$\hat{p}(D) = \frac{\sum_{i=1}^m \frac{p(D)}{p(D|\theta^{(i)})} p(D|\theta^{(i)})}{\sum_{i=1}^m \frac{p(D)}{p(D|\theta^{(i)})}}$$
$$= \left\{ \frac{1}{m} \sum_{i=1}^m p(D|\theta^{(i)})^{-1} \right\}^{-1}$$

Newton and Raftery's "Harmonic Mean Estimator"

Unstable in practice and needs modification

$p(D|M)$ from Gibbs Sampler Output

First note the following identity (for any \mathbb{W}^*):

$$p(D) = \frac{p(D | \theta^*) p(\theta^*)}{p(\theta^* | D)}$$

$p(D | \mathbb{W}^*)$ and $p(\mathbb{W}^*)$ are usually easy to evaluate.

What about $p(\mathbb{W}^* | D)$?

Suppose we decompose \mathbb{W} into $(\mathbb{W}_1, \mathbb{W}_2)$ such that $p(\mathbb{W}_1 | D, \mathbb{W}_2)$ and $p(\mathbb{W}_2 | D, \mathbb{W}_1)$ are available in closed-form...

Chib (1995)

$p(D|M)$ from Gibbs Sampler Output

$$p(\theta_1^*, \theta_2^* | D) = p(\theta_2^* | D, \theta_1^*) p(\theta_1^* | D)$$

The Gibbs sampler gives (dependent) draws from $p(\theta_1^*, \theta_2^* | D)$ and hence marginally from $p(\theta_2^* | D)$

$$\begin{aligned} p(\theta_1^* | D) &= \int p(\theta_1^* | D, \theta_2) p(\theta_2 | D) d\theta_2 \\ &\approx \frac{1}{G} \sum_{g=1}^G p(\theta_1^* | D, \theta_2^{(g)}) \end{aligned}$$

“Rao-Blackwellization”

Bayesian Information Criterion

$$S_{BIC}(M_k) = -S_L(\hat{\theta}_k; M_k) - \frac{d_k}{2} \log n, \quad k = 1, \dots, K$$

(S_L is the negative log-likelihood)

- BIC is an $O(1)$ approximation to $p(D|M)$
- Circumvents explicit prior
- Approximation is $O(n^{-1/2})$ for a class of priors called “unit information priors.”
- No free lunch (Weakliem (1998) example)

Srole (1956): "It's hardly fair to bring a child into the world now the way things look for the future." The data are from the 1993-94 General Social Survey; respondents were given the options of agreeing or disagreeing, and the few who could not choose are excluded from the analysis. The sample of 2,266 valid responses is composed of 44.0

$$\log(n_{ij}) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \Theta x_3. \quad (4)$$

In this parameterization, x_1 is a dummy variable that is zero in row 1 and one in row 2, x_2 is a dummy variable that is zero in column 1 and one in column 2, and x_3 is a dummy variable that is one in column 2 of row 2 and zero otherwise. The maximum likelihood estimate of Θ is the logarithm of the observed odds ratio $(n_{11}n_{22})/(n_{12}n_{21})$, so another way to put the question is to ask if Θ is equal to zero.

The sample contains 412 men who agree with the statement, 583 men who disagree, 584 women who agree, and 687 women who disagree. The L^2 for the model of independence is 4.68 with one degree of

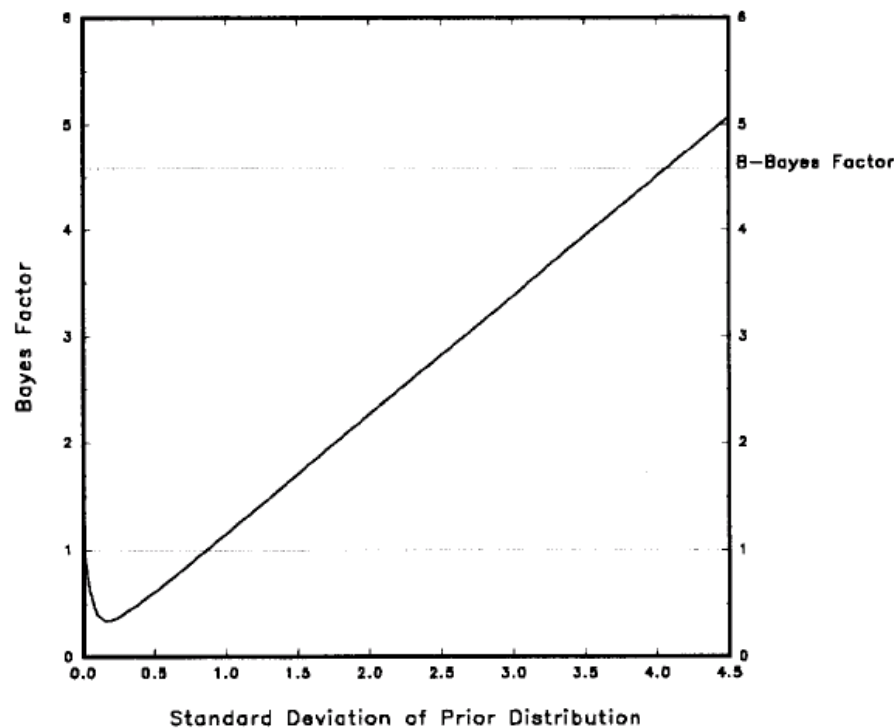


Figure 1: Bayes Factors for Model of No Association in Anomia by Gender Table:
Normal Prior Distribution With Mean Zero

about 4.0 (the exact figure is 4.07). With this prior distribution, the 95 percent range for possible values of the odds ratio would extend from $1/2,914$ to $2,914$, whereas the 50 percent range would extend from $1/14.7$ to 14.7 . In other words, adopting this prior distribution is equivalent to saying that if there is *any* association between the variables, there is a 50 percent chance that the absolute value of the odds ratio will be more than 14.7 or less than $1/14.7$. As discussed above,

Deviance Information Criterion

- Deviance is a standard measure of model fit:

$$D(y, \theta) = -2 \log p(y | \theta)$$

- Can summarize in two ways...at posterior mean or mode:

$$(1) \quad D_{\hat{\theta}}(y) = D(y, \hat{\theta}(y))$$

or by averaging over the posterior:

$$(2) \quad D_{avg}(y) = E(D(y, \theta) | y)$$

(2) will be bigger (i.e., worse) than (1)

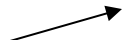

Deviance Information Criterion

$$p_D^{(1)} = D_{avg}(y) - D_{\hat{\theta}}(y)$$

is a measure of model complexity.

- In the normal linear model $p_D^{(1)}$ equals the number of parameters
- More generally $p_D^{(1)}$ equals the number of unconstrained parameters
- DIC = $D_{avg}(y) + p_D^{(1)}$
- Approximately equal to $E[D(y^{rep}, \hat{\theta}(y))]$

model search

- forward stepwise  start with the empty model and add one variable at a time greedily
- backward stepwise  start with the full model and delete one variable at a time greedily
- all-subsets
- genetic algorithms
- stochastic search

stepwise methods can get stuck at local modes

Zheng-Loh

1. fit the full model with all d predictors and let:

$$W_j = \frac{\hat{\beta}_j}{s\hat{e}(\hat{\beta}_j)}$$

2. Order the statistics in absolute value from largest to smallest:

$$|W_{(1)}| \geq |W_{(2)}| \geq \dots \geq |W_{(d)}|$$

3. Let \hat{j} be the value of j that minimizes

$$\text{RSS}(j) + j\hat{\sigma}^2 \log n$$

$\hat{\sigma}^2$ is the variance estimate from the full model

$\text{RSS}(j)$ is from the model using $x_{(1)}, \dots, x_{(j)}$

4. Choose as the final model, the regression with the terms with the largest W 's

Computing: Variable Selection via Stepwise Methods

- Efroymsen's 1960 algorithm still the most widely used

1. Enter into the (linear) regression model any variables that are to be “forced in.”

2. Find the variable from those not in the model but available for inclusion that has the largest F -to-enter value. If it is at least as great as a prespecified value, F_{in} , then add the variable to the model. Stop if no variable can be added.

3. Find that variable among those in the model, other than those forced in, that has the smallest F -to-remove value. If it is less than a prespecified value, F_{out} , then drop the variable from the model. Repeat this step until no further variables can be dropped; then go to step 2.

Efroymson

- F-to-Enter $\frac{\text{RSS}_p - \text{RSS}_{p+1}}{\text{RSS}_{p+1}/(n - p - 1)}$
 - F-to-Remove $\frac{\text{RSS}_{p-1} - \text{RSS}_p}{\text{RSS}_p/(n - p)}$
- } Distribution not even remotely like F
- Guaranteed to converge
 - Not guaranteed to converge to the right model...

Trouble

An artificial data set

<i>Observation number</i>	<i>Predictors</i>			<i>Y</i>
	<i>X₁</i>	<i>X₂</i>	<i>X₃</i>	
1	1000	1002	0	-2
2	-1000	-999	-1	-1
3	-1000	-1001	1	1
4	1000	998	0	2

- $Y = X_1 - X_2$
- Y almost orthogonal to X_1 and X_2
- Forward selection and Efroymsen pick X_3 alone

More Trouble

- Berk Example with 4 variables

Variables	<i>Highest R^2</i>
X_1	0.01
X_2, X_3	0.99
X_1, X_2, X_4	0.994

- The forward and backward sequence is $(X_1, X_1X_2, X_1X_2X_4)$
- The R^2 for X_1X_2 is 0.015

Even More Trouble

- “Detroit” example, $N=13$, $d=11$
- First variable selected in forward selection is the first variable eliminated by backward elimination
- Best subset of size 3 gives RSS of 6.8
- Forward’s best set of 3 has RSS = 21.2; Backward’s gets 23.5

Variable selection with pure noise using leaps

```
y <- rnorm(100)
xx <- matrix(rnorm(4000),ncol=40)
dimnames(xx) <- list(NULL,paste("X",1:40,sep=""))
```

```
library(leaps)
xx.subsets <- regsubsets(xx, y, method="exhaustive",
nvmax=3, nbest=1)
subvar <- summary(xx.subsets)$which[3,-1]
best3.lm <- lm(y ~ -1 + xx[, subvar])
print(summary(best3.lm, corr=FALSE))
```

```
or...bestsetNoise(m=100,n=40)
```

run this experiment ten times:

- all three significant at $p < 0.01$ 1
- all three significant at $p < 0.05$ 3
- two out of three significant at $p < 0.05$ 3
- one out of three significant at $p < 0.05$ 1

The Impact of Model Selection on Inference in Linear Regression

CLIFFORD M. HURVICH and CHIH-LING TSAI*

Table 2. Coverage Rates for Confidence Regions With Nominal
Rate $1 - \alpha$
($n = 30$, $p_o = 4$, model order p chosen by AIC)

p	$1 - \alpha = .9$
3	.000 (0/15)
4	.867 (263/303)
5	.838 (62/74)
6	.660 (35/53)
7	.600 (33/55)
OCR	.786 (393/500)

Table 5. Coverage Rates for Confidence Regions With Nominal
Rate $1 - \alpha$
($n = 30$, $p_o = 4$, model order p chosen by BIC)

p	$1 - \alpha = .9$
3	.000 (0/62)
4	.890 (331/372)
5	.833 (30/36)
6	.313 (5/16)
7	.286 (4/14)
OCR	.740 (370/500)

Bayesian Model Averaging

- If we believe that one of the candidate models generated the data, then the predictively optimal strategy is to average over all the models.
- If Q is the inferential target, Bayesian Model Averaging (BMA) computes:

$$p(Q) = \sum_{i \in I} p(Q|D, M_i)p(M_i|D)$$

- Substantial empirical evidence that BMA provides better prediction than model selection

