

Introducing Computation via Statistics, and *vice versa*.

Daniel T. Kaplan

Macalester College

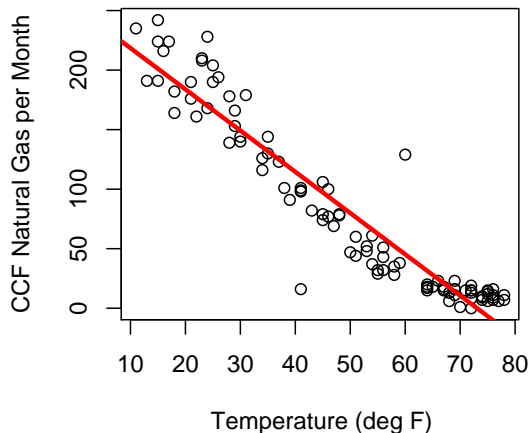
Mellon Inter-Institutional Workshop
Computing and Mathematics Across the Sciences
June 4, 2009

The Introductory Statistics Course

- Collection of data: random sampling, experiment and random assignment
- Descriptive statistics: mean, median, standard deviation, IQR, correlation, simple linear regression, scatter plots, box-plots, histograms, outliers, skewness
- Basic probability (not usually emphasized). Central limit theorem. Law of large numbers. Binomial and normal distributions.
- Inference
 - ▶ confidence intervals on sample means and proportions
 - ▶ t-tests, one-way ANOVA
 - ▶ χ^2 test on counts
 - ▶ simple linear regression
- Typically doesn't require calculus. If it does, calculus not really used: areas under curves.
- Computation (apologetically) done in Minitab, Excel, Fathom, JMP, SPSS, and occasionally (but growing!) in R.

Linear Regression

Given pairs of data (x, y) find the parameters of the best fitting relationship $y = b_0 + b_1x$



Linear Regression (cont.)

The Algorithm

- 1 Compute the means of x and $y \rightarrow \bar{x}, \bar{y}$.
- 2 Compute the standard deviations of x and $y \rightarrow s_x$ and s_y .
- 3 Compute the correlation between x and $y \rightarrow r$.
- 4 $b_1 = rs_y/s_x$
- 5 $b_0 = \bar{y} - b_1\bar{x}$

Using R (like the Lost In Space Robot)

```
> b1 = with(utils, cor(temp, ccf) * sd(ccf)/sd(temp))
> b0 = with(utils, mean(ccf) - b1 * mean(temp))
> b0

[1] 253.0982

> b1

[1] -3.464251
```

Linear Regression (cont)

A Formula for the Standard Error of b_1

- 1 Fit the line and find b_0 and b_1 and remember s_x .
- 2 Calculate $\hat{y} = b_0 + b_1x$.
- 3 Compute $s_e = \sqrt{\frac{\sum(y-\hat{y})^2}{n-2}}$
- 4 $SE(b_1) = \frac{s_e}{\sqrt{n-1}s_x}$.

Using R (like the Lost in Space Robot)

```
> yhat = with(utils, b0 + temp * b1)
> resids = with(utils, ccf - yhat)
> se = sqrt(sum(resids^2)/(nrow(utils) - 2))
> b1se = se/(sqrt(nrow(utils) - 1) * sd(utils$temp))
> b1se

[1] 0.1154627
```

Linear Regression (cont)

A Confidence Interval on b_1

- 1 Find the critical value of the t distribution for $n - 2$ degrees of freedom. For significance level α , this is the $(1 - \alpha)/2$ quantile. Almost always you use $\alpha = 0.05$.
- 2 Multiply this by the standard error to get the margin of error.

Using R (like the Robot ...)

```
> tcritical = qt(0.025, df = nrow(utils) - 2)
> tcritical * b1se
[1] -0.2291614
```

So we would write the 95% confidence interval on the slope as -3.46 ± 0.23 .

The “Black Box” Approach

One of the basic ideas in computation is that you can abstract an operation.

The Natural R Approach

```
> mod = lm(ccf ~ temp, data = utils)
> summary(mod)
```

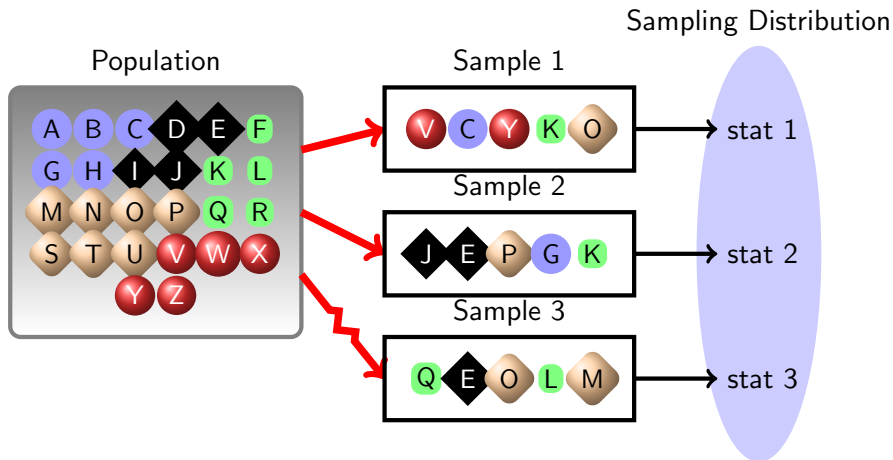
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	253.0982	6.1927	40.87	0.0000
temp	-3.4643	0.1155	-30.00	0.0000

Some people don't like this. They think it hides the “real” math in a black box. But there are some things you can do better if you adopt this approach.

Thinking Statistically about Regression

- Revealing the basic logic of the standard error.
- Allowing additional variables into the model.
... as well as some things I won't talk about today ...
- Making sure the model makes sense. Deal with the outliers and with the failure of the linear model at temperatures above 65° and the higher variance of residuals at low temperature.

The Process behind Sampling Distributions



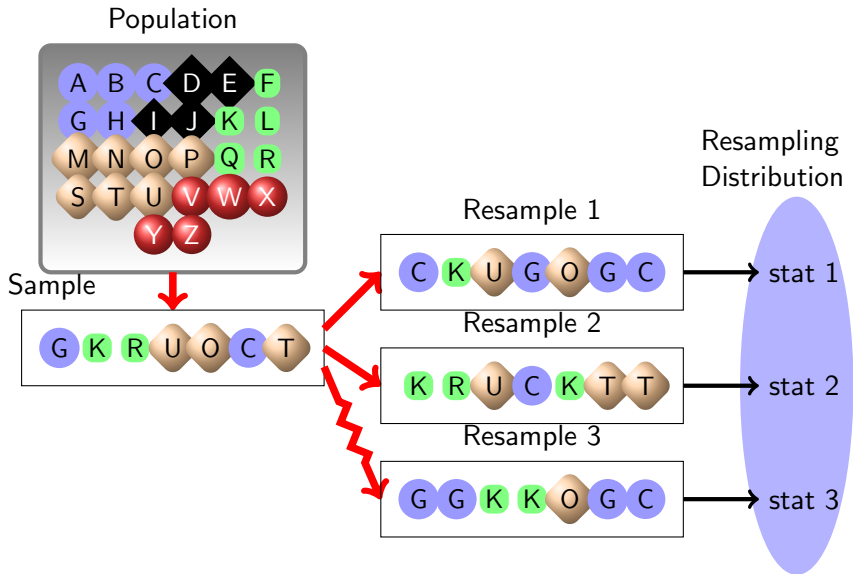
Resampling

Typically, it's impractical to draw repeated samples from the population. We acquire our sample with difficulty; repeating it is not feasible.

Instead of repeating the draws from the population, we draw instead from the sample. This is a matter of copying the entries from the sample rather than going to the field for new data.

Sampling from a sample: RE-sampling.

The Resampling Distribution



Computer Operators for Sampling and Resampling

- `resample` — selects randomly from a set with replacement.
- `shuffle` — selects randomly from a set without replacement.

```
> resample(1:6)
```

```
[1] 4 1 1 6 5 3
```

```
> resample(1:6)
```

```
[1] 6 1 2 4 6 6
```

```
> shuffle(1:6)
```

```
[1] 3 4 5 1 2 6
```

```
> resample(1:6, 20)
```

```
[1] 4 1 1 3 1 1 6 3 6 4 5 3 4 3 1 6 2 5 4 3
```

Thinking the Unthinkable (from 1979)

SIAM REVIEW
Vol. 21, No. 4, October 1979

© 1979 Society for Industrial and Applied Mathematics
0036-1445/79/2104-0002\$01.00/0

COMPUTERS AND THE THEORY OF STATISTICS: THINKING THE UNTHINKABLE*

BRADLEY EFRON†

...

The “unthinkable” mentioned in the title is simply the thought that one might be willing to perform 500,000 numerical operations in the analysis of 16 data points. Or one might be willing to perform a billion operations to analyze 500 numbers. Such statements would have seemed insane thirty years ago, when a slow and noisy fifty pound desk calculator which added, subtracted, multiplied, and divided was the most sophisticated computational aid available to most scientists. Most of the statistical theory in common use was developed under the constraint of slow and expensive computation. Now computation is fast and cheap. It is not surprising that new theory is being developed, which takes advantage of the high-speed computer. This paper consists of several examples of such theory, presented, hopefully, in a manner accessible to nonstatisticians.

Bootstrapping the Standard Error

```
> samps = do(1000) * lm(ccf ~ temp, data = resample(utils))  
> head(samps)
```

	(Intercept)	temp
1	246.9330	-3.393529
2	256.3900	-3.567407
3	246.7151	-3.349271
4	255.4658	-3.498249
5	242.8217	-3.345717
6	250.4681	-3.361495

```
> sd(samps)
```

	(Intercept)	temp
	6.5248669	0.1109608

Covariation

A common problem in statistics is to reveal the relationship between two variables: the response and an explanatory variable. Examples:

- How does natural gas usage (ccf) depend on temperature?
- Does a PSA testing regimen reduce mortality from prostate cancer?
- How are educational outcomes influenced by school expenditures?

In many or most settings, there are some other variables that are important in setting the context for the relationship. These are covariates or confounders or “lurking variables.”

Example: SATs and Confounding

[T]he 10 states with the lowest per pupil spending included four — North Dakota, South Dakota, Tennessee, Utah — among the 10 states with the top SAT scores. Only one of the 10 states with the highest per pupil expenditures — Wisconsin — was among the 10 states with the highest SAT scores. New Jersey has the highest per pupil expenditures, an astonishing \$10,561, which teachers' unions elsewhere try to use as a negotiating benchmark. New Jersey's rank regarding SAT scores? Thirty-ninth... The fact that the quality of schools... [fails to correlate] with education appropriations will have no effect on the teacher unions' insistence that money is the crucial variable. — George F. Will, (September 12, 1993), "Meaningless Money Factor," The Washington Post, C7.

Looking at the Data

State-by-state average SAT data from 1994

- expend — per pupil expenditures in thousands of dollars.
- ratio — average pupil/teacher ratio in public elementary and secondary schools.
- salary — estimated average annual salary of public elementary and secondary teachers in thousands of dollars
- sat — average total score on the SAT
- frac — percentage of all eligible students taking the SAT

Simple Regression on SAT

```
> sat = ISMdata("sat.csv")
```

Expenditures

```
> summary(lm(sat ~ expend, data = sat))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1089.2937	44.3900	24.54	0.0000
expend	-20.8922	7.3282	-2.85	0.0064

SAT goes down with higher expenditures!

Salaries

```
> summary(lm(sat ~ salary, data = sat))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1158.8588	57.6594	20.10	0.0000
salary	-5.5396	1.6324	-3.39	0.0014

SAT goes down with higher salaries!

Student/Teacher Ratio

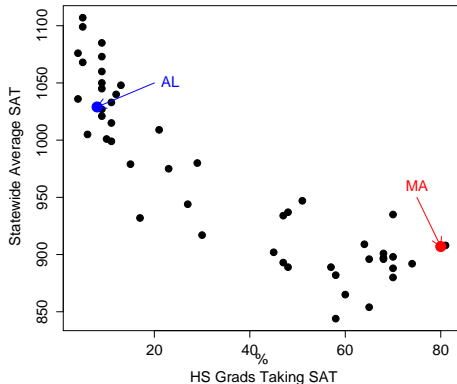
```
> summary(lm(sat ~ ratio, data = sat))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	920.6987	80.7705	11.40	0.0000
ratio	2.6825	4.7493	0.56	0.5748

No discernible relationship.

SAT Averages are Not Directly Comparable

- Statewide SAT averages — the data George Will used — are influenced by at least two factors: how good the schools are and what fraction of students take the test in each state.



Including frac as a Covariate

Expenditures

```
> summary(lm(sat ~ expend + frac, data = sat))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	993.8317	21.8332	45.52	0.0000
expend	12.2865	4.2243	2.91	0.0055
frac	-2.8509	0.2151	-13.25	0.0000

SAT goes **up** with higher expenditures!

Salaries

```
> summary(lm(sat ~ salary + frac, data = sat))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	987.9005	31.8775	30.99	0.0000
salary	2.1804	1.0291	2.12	0.0394
frac	-2.7787	0.2285	-12.16	0.0000

SAT goes **up** with higher salaries!

Student/Teacher Ratio

```
> summary(lm(sat ~ ratio + frac, data = sat))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1118.5087	39.4733	28.34	0.0000
ratio	-3.7264	2.2089	-1.69	0.0982
frac	-2.5474	0.1871	-13.62	0.0000

Computers and Multivariate Models

Fitting models with multiple explanatory variables — even hundreds of them — is not difficult. But ...

- The formulas for doing this involve matrices.
- Singularity is often an issue, particularly when categorical variables are involved.

My Attitude ...

The statistical need to incorporate covariates is absolute. So let the computer handle the routine calculation issues and teach students how to think about the process.

- Open the black box not with formulas (which are themselves opaque to most students) but with alternative representations. I use geometry.
- The computer lets students focus on the statistical rather than computational issues: collinearity, model specification and interpretation.

Taking Data Organization Seriously

Since statistics books tend to focus on examples involving one or two variables, they take a casual attitude toward organizing data: lists of numbers. If we want students to be able to deal with complicated situations, we need to give them some of the skills needed.

Relational Databases

- Few students (or faculty!) know what this term means, but it is tremendously important.
- It doesn't take long to teach students the basics of working with data organized relationally. Four basic operations: select, project, group, join.

Example: Grade Data

Three basic tables from the registrar's office:

- Grades — a case is one student registered in one course.
- Courses — a case is one course offered in one semester
- Grade-to-number — Letter to number policy for GPA.

Grades

	sid	grade	sessionID
1	S31215	A-	session3762
2	S31647	B	session2064
3	S32319	B	session3151
4	S32445	A	session3080
5	S31830	B+	session2269
6	S31344	B	session2087
7	S32112	A-	session3537
8	S31791	A-	session2386
9	S31251	C	session2240
10	S31701	B-	session2399

and so on...

Courses

	sessionID	dept	level	sem	enroll	iid
1	session2064	D	200	FA2001	10	inst162
2	session2087	i	100	FA2001	24	inst423
3	session2240	M	300	SP2002	18	inst263
4	session2269	q	200	SP2002	20	inst530
5	session2386	N	300	SP2003	18	inst278
6	session2399	W	300	FA2002	41	inst211
7	session3080	m	300	FA2003	37	inst488
8	session3151	O	200	SP2004	16	inst288
9	session3537	n	100	FA2004	12	inst497
10	session3762	U	400	SP2005	20	inst220

and so on...

GPA policy

	grade	gradepoint
1	A	4.00
2	A-	3.66
3	B+	3.33
4	B	3.00
5	B-	2.66
6	C+	2.33
7	C	2.00
8	C-	1.66
9	D+	1.33
10	D	1.00
11	D-	0.66
12	I	
13	NC	0.00
14	S	
15	AU	

Joining the Tables

```
> foo = merge(gg, ff)
> xtable(foo)
```

	sessionID	sid	grade	dept	level	sem	enroll	iid
1	session2064	S31647	B	D	200	FA2001	10	inst163
2	session2087	S31344	B	i	100	FA2001	24	inst423
3	session2240	S31251	C	M	300	SP2002	18	inst263
4	session2269	S31830	B+	q	200	SP2002	20	inst530
5	session2386	S31791	A-	N	300	SP2003	18	inst278
6	session2399	S31701	B-	W	300	FA2002	41	inst211
7	session3080	S32445	A	m	300	FA2003	37	inst488
8	session3151	S32319	B	O	200	SP2004	16	inst288
9	session3537	S32112	A-	n	100	FA2004	12	inst497
10	session3762	S31215	A-	U	400	SP2005	20	inst220

Example: Grouping Operations to Calculate GPA

```
> all = merge(grades, courses)
> all = merge(all, gp)
> all = subset(all, complete.cases(all))
```

Student GPA

```
> sgpa = with(all, group(gradepoint, sid, mean))
```

	group	results
1	S31185	2.41
2	S31188	3.02
3	S31191	3.21
4	S31194	3.36
5	S31197	3.33
6	S31200	2.19

How big is a “typical” course?

The Institutional Perspective

Average over the courses:

```
> mean(courses$enroll)
```

```
[1] 21.17006
```

```
> median(courses$enroll)
```

```
[1] 18
```

```
> prop.table(table(courses$enroll))
```

FALSE	TRUE
0.7373326	0.2626674

The Student Perspective

Average over the students-in-a-course

```
> mean(all$enroll)
```

```
[1] 23.19916
```

```
> median(all$enroll)
```

```
[1] 20
```

```
> prop.table(table(all$enroll))
```

FALSE	TRUE
0.8011911	0.1988089

GPA with a Standard Error

Standard Error of the Mean

```
> mean.se = function(x) {  
+   list(mean = mean(x), se = sd(x)/sqrt(length(x)))  
+ }
```

Student GPA with Standard Error

```
> sgpa = with(all, group(gradepoint, sid, mean.se))
```

	mean	se	group
1	2.41	0.43	S31185
2	3.02	0.25	S31188
3	3.21	0.10	S31191
4	3.36	0.17	S31194
5	3.33	0.09	S31197
6	2.19	0.26	S31200

Faculty GPA

```
> fgpa = with(all, group(gradepoint, iid, mean.se))
```

	mean	se	group
1	3.41	0.16	inst125
2	3.51	0.06	inst126
3	3.15	0.31	inst128
4	3.55	0.45	inst129

Department GPA

```
> dgpa = with(all, group(gradepoint, dept, mean.se))
```

	mean	se	group
36	3.48	0.05	p
37	3.50	0.03	q
38	3.48	0.08	s
39	3.48	0.06	t

GPA with Covariates

- The ordinary GPA is effectively the coefficient from the model $\text{gradepoint} \sim \text{student}$.
- But there are covariates: the level of the course, the department, the instructor.
- Modeling lets you take the covariates into account, effectively adjusting for different instructors.
- Done by fitting a model with approximately 1000 explanatory variables: one for each student and for each instructor. Feasible on a small laptop using Matlab, R,

An Example: Classifying Natural Language Using χ^2 I

Here are some data on the frequency of different letters in various languages. (From, A Salomaa, "Public-Key Cryptography", 2nd ed. Springer-Verlag, 1996, p. 17)

English		German		Finnish	
	%		%		%
e	12.31	e	18.46	a	12.06
t	9.59	n	11.42	i	10.59
a	8.05	i	8.02	t	9.76
o	7.94	r	7.14	n	8.64
n	7.19	s	7.04	e	8.11
i	7.18	a	5.38	s	7.83
s	6.59	t	5.22	l	5.86
r	6.03	u	5.01	o	5.54
h	5.14	d	4.94	k	5.20

An Example: Classifying Natural Language Using χ^2 II

French		Italian		Spanish	
	%		%		%
e	15.87	e	11.79	e	13.15
a	9.42	a	11.74	a	12.69
i	8.41	i	11.28	o	9.49
s	7.90	o	9.83	s	7.60
t	7.26	n	6.88	n	6.95
n	7.15	l	6.51	r	6.25
r	6.46	r	6.37	i	6.25
u	6.24	t	5.62	l	5.94
l	5.34	s	4.98	d	5.58

An Example: Classifying Natural Language Using χ^2 III

- Write a program, `whichlanguage(string, frequencies)` that takes a character string and some data structure representing the frequencies in each of the above languages — it's your choice how to represent the above tables — and calculates a χ^2 statistic for each of the languages (based on the relevant characters for that language). The program should return the name of the language whose χ^2 statistic was smallest. The program should be written in a way that makes it easy to add more languages by changing the contents of the `frequencies` argument.

Note that in order to compute the “expected” number of counts of each letter in one language, you need to multiply the length of the string by the percentage given above. Keep in mind that a percentage should be represented as a fraction, that is 5% is 0.05.

Here are some examples, using frequency information stored in a variable called `languagefreqs`. Please note that this method for classifying text is not reliable for very short phrases such as those used here.

An Example: Classifying Natural Language Using χ^2 IV

Many a true word is spoken in jest.

```
> whichlanguage('Entre broma y broma, la verdad se asoma.', ...  
  languagefreqs)  
ans: Spanish
```

Too many cooks spoil the broth.

```
> whichlanguage('Viele Koche verderben den Brei.', languagefreqs)  
ans: German
```

Hope for miracles, but don't rely on one happening.

```
> whichlanguage('Ihmetapauksiin voi toivoa mutta ala luota niihin.',  
  languagefreqs)  
ans: Finnish
```

Strike while the iron is hot.

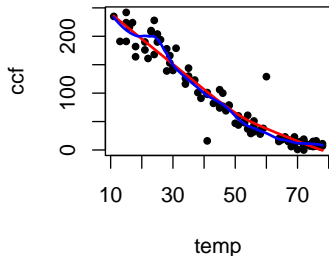
```
> whichlanguage('Il faut battre le fer pendant qu'il est chaud.', ..  
  languagefreqs)  
ans: French
```

Decoding ENIGMA



- Coding machine patented in the 1920s and extensively used by the Germans in WW II.
- Code set by the position of thumbwheels and location of patch cords (in front).
- The settings were decoded by heroic effort by the Polish, British, and US during the war.
- A good project to write a simulator and then decode it using the entropy of the output to determine when the plain text has been found. (Thumbwheels only.)

Leave-one-out Cross Validation



Which one of the curves is a better fit?

- Measure quality of fit with the sum of square residuals.
- The wiggly one is closer to the data points — smaller residuals. But does it “overfit” the data. With enough wiggleness, you can get close to the data.
- Leave each point out of the data set in turn, fit the model, then evaluate the residual for the omitted point.
- Choose the model structure that has the best fit for the left-out points.

Summary

- Computation is natural in statistics. Now that it's easy to do, we should do it.
- Statistical settings make nice applications for computing students.
- So many students see statistics and are required to take it, that it is a natural locus for teaching computing.