An introduction to statistical inference

Ole Christian Lingjærde

Biomedical Research Group

Terminology

Variable:

These are what we observe or measure

Dependent variable (response):

Outcome of interest

Independent variable:

The outcome is modeled as depending on these

May or may not be under our control

Example

- DV: subject got the influenza (yes/no)
- IV: subject was vaccinated

The independent variable is here under our control

- DV: expression of gene TST2 (continuous)
- IV: SNP allele at locus X (AA, Aa or aa)

The independent variable is here not under our control

Types of data

Nominal	Named categories with no order
↓	
Ordinal	Ordered categories
↓	
Interval	Equal intervals, arbitrary zero point
↓	
Ratio	Meaningful zero point

Examples

Nominal:

Did/did not receive treatment

Ordinal:

Stage I, II, III, IV cancer

Interval:

IQ score (possibly)

Ratio:

Height and weight

Measures of location

Mean:

The most common measure of central tendency
For ratio data and interval data

Median:

Half of the data points fall on each side of it
Also applicable to ordinal data

Mode:

The value corresponding to the distribution peak
Also applicable to nominal data

Measures of dispersion

Range:

Difference between the highest and lowest values

Interquartile range (IQR):

Range of the middle 50% of the data
(Difference between 75th and 25th percentile)

Median absolute deviation (MAD):

The median of the numbers $|x_i - m|$ where m is the median of the observations $x_1, \ldots, x_n$

Variance and standard deviation

Guidelines for use

Data	Location (central tendency)	Dispersion
Nominal	Mode	-
Ordinal	Mode Median	Range Interquartile range
Interval	Mode Median Mean	Range Interquartile range Median absolute deviation Standard deviation
Ratio	Mode Median Mean	Range Interquartile range Median absolute deviation Standard deviation

Statistical inference

Population:

The collection of subjects that we would like to draw conclusions about.

Sample:

The subcollection considered in the study

Statistical inference:

Draw sample-based conclusions about the population, controlling for the probability of making false claims.

Statistical tests (the idea)

1) A population has individuals with an observable feature X that follows $X \sim F(\theta)$. We seek if (say) $\theta = 0$ is violated.

2) We obtain X-values $X_1,...X_N$ on a random sample.

3) A test statistic $Z = Z(X_1,...X_N)$ is defined. The observed Z is denoted z_{obs}. Large $|z_{obs}|$ supports violations.

4) Calculate the probability that $|Z| \geq |z_{obs}|$ (= p-value)

5) Conclude that $\theta = 0$ is violated if p-value is small.

Example

A population has individuals with an observable feature X that follows X ~ F(θ). We seek if some condition, say $\theta = 0$, is violated.

Example: We observe feature X in n randomly sampled individuals and assume that

$$X_1, \ldots, X_n \sim \text{i.i.d. } N(\mu, \sigma^2)$$

where the variance is assumed to be equal to 1. We seek to investigate if

$$\text{H}_0 : \quad \mu = 0$$

is violated.

Step 1

Step 2

Step 3

Step 4

Step 5

Example

We obtain X-values $X_1,\ldots X_N$ on a random sample.

Step 1
Step 2
Step 3
Step 4
Step 5

Observations:

-0.1694	0.2534	1.3868	1.7235
1.6444	2.1598	0.9932	1.1155
0.2808	1.2175	-1.2761	-0.0229
-0.4444	-0.0036	-2.2036	-0.1624
-0.7595	1.0500	-0.4378	-0.9326

Example

A test statistic Z = Z(X_1,...X_N) is defined. The observed Z is denoted z_{obs}. Large |z_{obs}| supports violations of the condition $\theta = 0$.

$$Z = \frac{\bar{X}}{1/\sqrt{20}}$$

$$\bar{X} = \frac{1}{20}\sum_{i=1}^{20} X_i$$

Step 1

Step 2

Step 3

Step 4

Step 5

Example

Calculate the probability that $|Z| \geq |z_{obs}|$ (= p-value)

$$z_{\text{obs}} = 1.210$$

$$Pr(|Z| \geq |z_{\text{obs}}|) = 2 \cdot Pr(Z < -1.210) = 0.226$$

Step 1
Step 2
Step 3
Step 4
Step 5

Example

Conclude that $\theta = 0$ is violated if p-value is small.

The p-value is large in this case (compared to 0.05 or 0.01) and we do not conclude that the expected value is different from zero.

Note: we do not conclude that the expected value is zero.

Step 1

Step 2

Step 3

Step 4

Step 5

One-sample location tests

Purpose:

Compare the location parameter of a population to a known constant value

Examples:

One-sample z-test

One-sample t-test

One-sample Wilcoxon signed ranks test

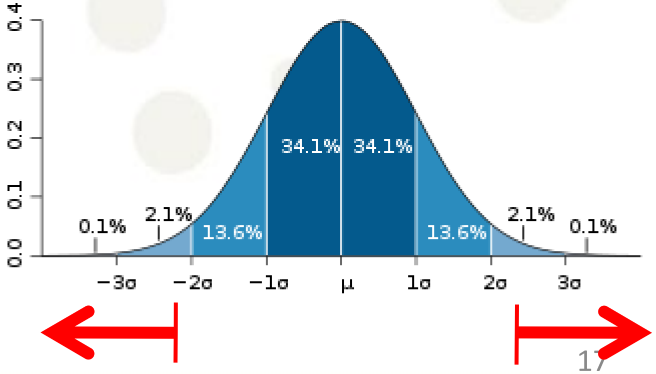
The one-sample z-test

Sample: $X_1, \ldots, X_n \sim \text{i.i.d. } N(\mu, \sigma^2)$ (σ known)

Null hypothesis (H_0) : $\mu = \mu_0$

Test statistic: $z = \dfrac{\bar{x} - \mu_0}{\sigma / \sqrt{n}}$

Reject H_0 if: $|z| > z_{\alpha/2}$



The one-sample t-test

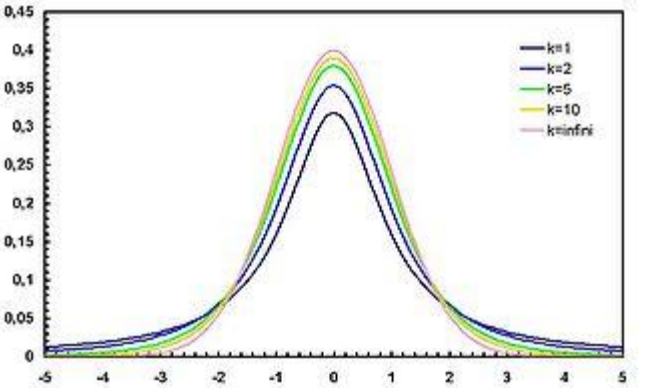
Sample: $X_1, \ldots, X_n \sim \text{i.i.d. } N(\mu, \sigma^2)$ (σ unknown)

Null hypothesis (H_0) : $\mu = \mu_0$

Test statistic: $t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$

Reject H_0 if: $|t| > t_{n-1,\alpha/2}$

Student's t-distribution is more heavy-tailed than the normal distribution. It approaches the normal distribution as the degrees of freedom increases:



One-sample Wilcoxon signed rank test

This is an alternative to the one-sample t-test.

It tests whether the median of the observations is equal to a specified value μ_0 .

It is a nonparametric test – there are no assumptions for the distribution of the measurement except that the probability distribution be symmetric.

Algorithm: rank the differences $d_i = x_i - \mu_0$, ignoring signs. Find the sum W of the ranks associated with positive d_i . A simple transformation of W is approximately N(0,1) and a Z-test may be applied.

Comparing the distribution of a sample to a theoretical distribution

A few examples are:

- Pearson's chi-square goodness-of-fit test: test the null hypothesis that the sampling distribution is equal to a given theoretical distribution

- Shapiro-Wilk: tests the null hypothesis that data come from a normal distribution

Comparing the mean of two groups

A common task is to compare the mean in two groups of (unmatched) individuals.

The easiest approach is the two-sample unpaired t-test, which utilizes the test statistic

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{(n_1-1)s_1^2 + (n_2-1)s_2^2}{n_1+n_2-2} \times \left(\frac{1}{n_1} + \frac{1}{n_2}\right)}} \sim t_{n_1+n_2-2}$$

Comparing distributions of two groups

Similar principle as used to compare a sampling distribution to a theoretical distribution.

Examples:

Mann-Whitney U-test (Wilcoxon rank sum test)

Kolmogorov-Smirnov test

Comparing more than two groups

To compare means in more than two groups, a one-way ANOVA is a very useful tool.

The basic idea in one-way ANOVA is to look at the ratio between

- The total squared distance between group means
- The variability within groups

The larger the ratio, the more evidence there is of a difference in the means of the groups. An F-test can be applied.

BUT we don’t see what groups differ from each other.

For this, we need to perform a post-hoc multiple comparison.

Observation is selection

Explanation:

An observation is interesting only in so far as it is representative of the population we are interested in.

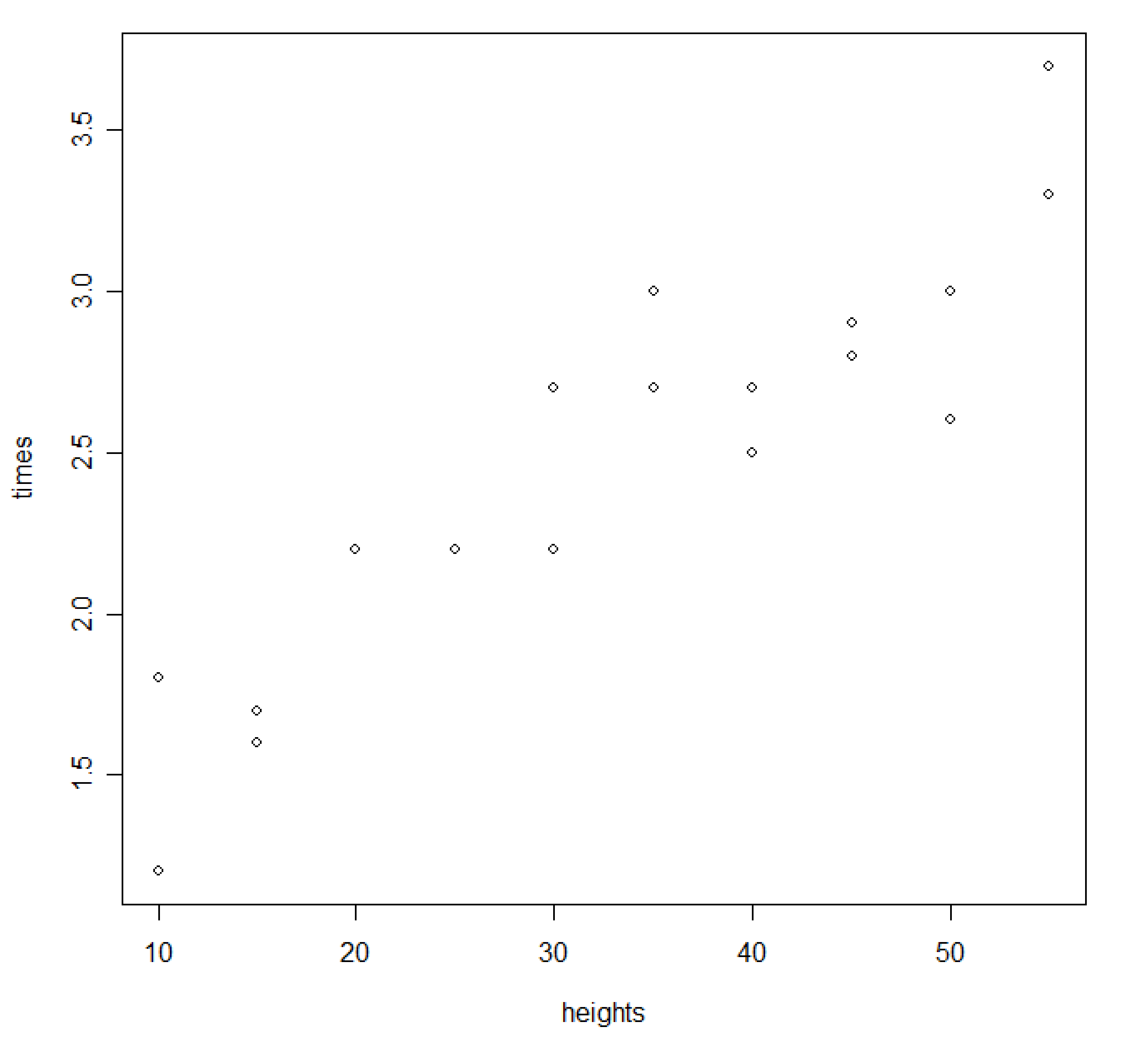
Invalid selection is the primary threat to valid inference.

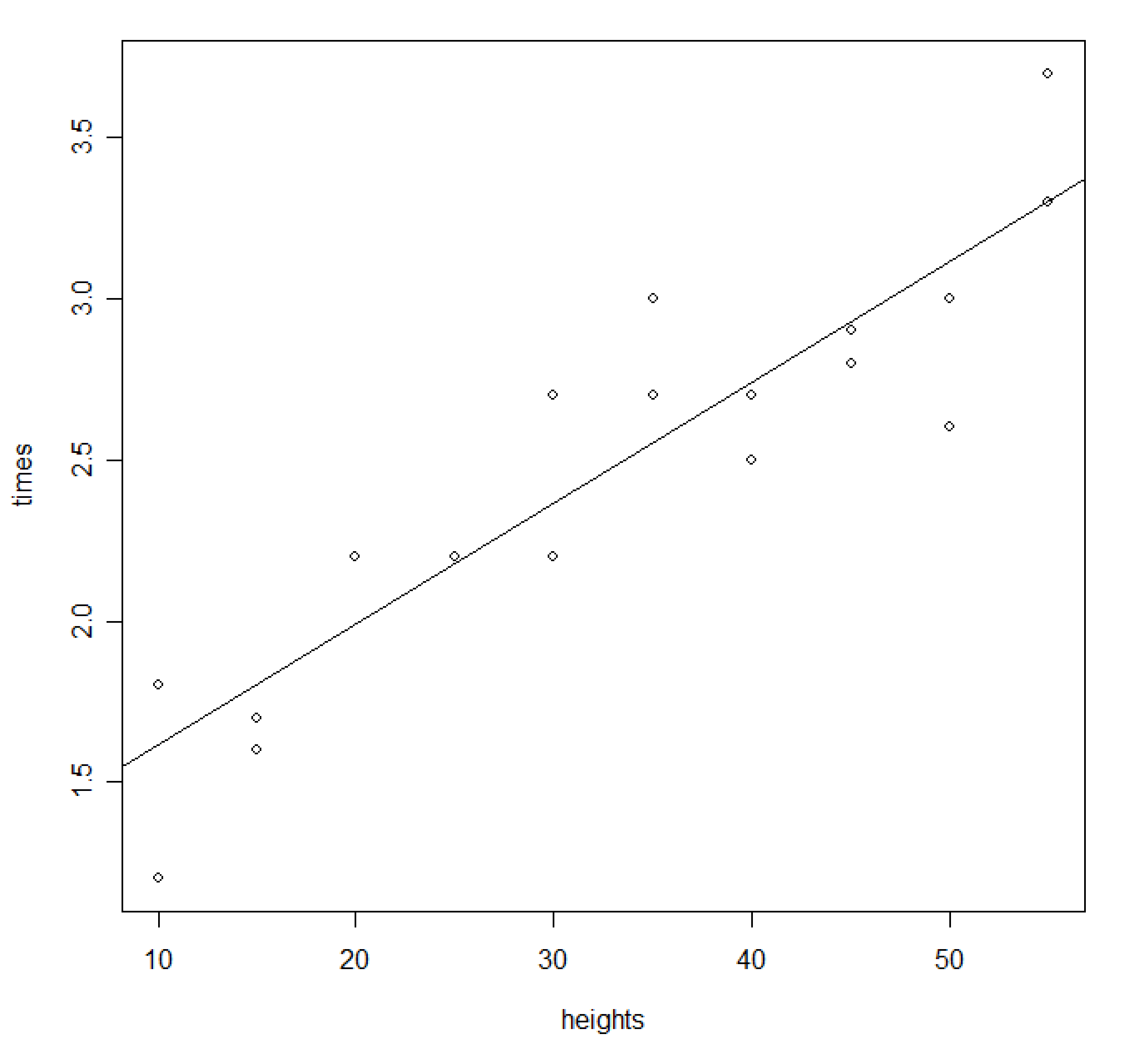
Example: Vulnerability analysis of planes returning from bombing missions during World War II.

Models are usually wrong

Explanation:

Models are theoretical constructs, not reality. This must always be remembered when interpreting significant effects.


times
heights
1.5
2.0
2.5
3.0
3.5
10
20
30
40
50


3.5
3.0
2.5
2.0
1.5
times
10
20
30
40
50
heights

$$\text{Time} = 1.243 + 0.0375 \cdot \text{Height}$$

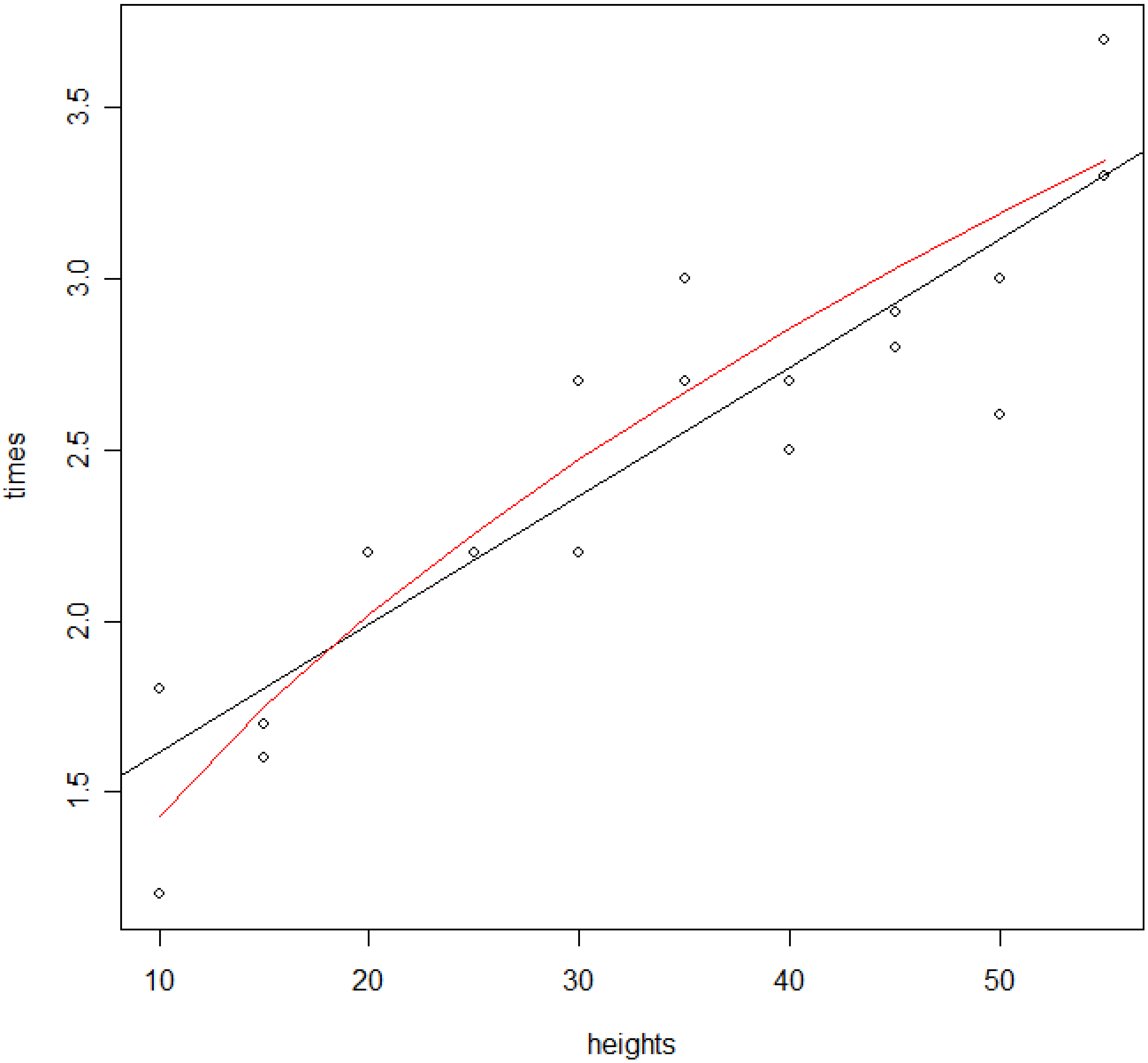
This model completely misses the point!

From Newton’s second law we know that

$$s = \frac{1}{2}gt^2$$

where s = distance traveled, t = time, g = 9.81 m/s^2.
Thus:

$$\text{Time} = \sqrt{\frac{2}{g} \cdot \text{Height}}$$


times
1.5
2.0
2.5
3.0
3.5
10
20
30
40
50
heights

Even though the intercept is highly significant in the linear model, it has no physical meaning:

$$\text{Time} = 1.243 + 0.0375 \cdot \text{Height}$$

→ Time = 1.243 at Height = 0

$$\text{Time} = \sqrt{\frac{2}{g} \cdot \text{Height}}$$

→ Time = 0 at Height = 0

Conclusion:

Even though an effect is highly significant in a model, it may not correspond to a real effect!

Also, extrapolations are very dangerous.

Statistical significance says nothing about the actual magnitude of the effect

For sample sizes ≥ 20 a point estimate ± two standard errors has roughly 95% coverage for a wide variety of distributions

Explanation:

While the coverage rule is derived from normal distribution assumptions, it is remarkably robust to distributional changes.

Estimates of correlation must be handled carefully in regression sampling schemes

Explanation:

In regression sampling, the researcher chooses the values of X.

The correlation coefficient r is dependent on the choice of values of X.

Statistical models of small effects are very sensitive to assumptions

How data are organized in R

```
# Single values:
x <- 3.5
y <- TRUE
z <- pizza

# Vectors:
x <- c(3.5, 1.2, 4.1)
y <- c(TRUE, TRUE, FALSE)
z <- c(A, small, vector)

# Matrices:
x <- matrix(0, nrow=3, ncol=4)
y <- matrix(TRUE, nrow=3, ncol=5)

# Data frames:
data <- data.frame(matrix(0, nrow=3, ncol=4))

# Lists:
x <- list(a = test, b = c(1,2,3), c = TRUE)
```

Creating vectors in R

```
# Specified values
x <- c(1, 3, 5, 7)
x
 [1] 1 3 5 7

# All values in a range
x <- 1:10
x
 [1]  1  2  3  4  5  6  7  8  9 10

# All values in a range, arbitrary step length
x <- seq(0, 1, by=0.10)
x
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

# Several identical values
x <- rep(0, 7)
x
 [1] 0 0 0 0 0 0 0
```

Many functions in R work on vectors

R is a vectorized language: functions that work on single values, also work on vectors by application to each component of the vector.

```
x <- c(1, 2, 3)
y <- c(4, 5, 6)
z <- x + y
z
[1] 5 7 9

x <- c(1, 2, 4, 8, 16, 32, 64)
y <- log2(x)
y
[1] 0 1 2 3 4 5 6

x <- c(1, 3, 5, 7, 9)
y <- x < mean(x)
y
[1]  TRUE  TRUE FALSE FALSE FALSE
```

Other things to do with vectors in R

```
# Subscripting a vector
x <- seq(0, 2, by=0.1)
x[2:6]
 [1] 0.1 0.2 0.3 0.4 0.5

# Negative subscripting of a vector
x <- c(1, 2, 4, 8, 16, 32, 64)
x[-c(1,2,3)]
 [1]  8 16 32 64

# Selecting the subset that satisfies a condition
x <- c(-1, 2, -3, 4, -5, 6, -7, 8)
x[x > 0]
 [1] 2 4 6 8

# Sorting the elements of a vector
x <- rnorm(4, mean=0, sd=1)
sort(x)
[1] -1.0849764 -1.0133422 -0.6469750  0.4340475
```

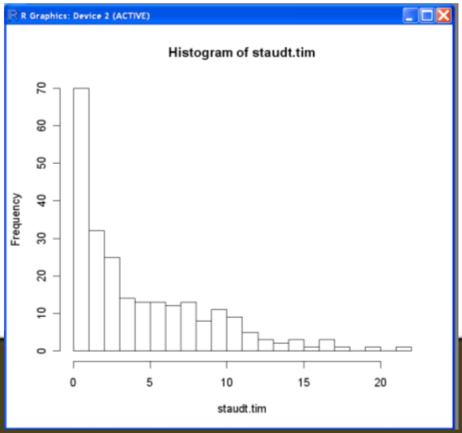
Basic summaries of vectors in R

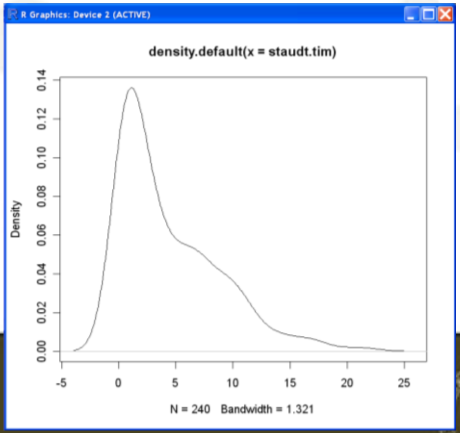
To compute the quartiles of a numeric vector:

```
summary(vec)
   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  0.000    0.900    2.800    4.411    7.100   21.800
```

To show the distribution of the elements in a vector:

```
hist(vec, nclass=20)
plot(density(vec))
```




R Graphics: Device 2 (ACTIVE)
density.default(x = staudt.tim)
Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
-5
0
5
10
15
20
25
N = 240   Bandwidth = 1.321

Data frames in R

Data frames are similar to data sheets. In particular:

- All values in a column are of the same type (e.g. numeric)
- Every column has a name (need not be unique)
- Every row has a name (need not be unique)
- Elements can be referred to by indexing

```
data <- data.frame(matrix(1:6,ncol=3))
data
  X1 X2 X3
1  1  3  5
2  2  4  6
names(data) <- c(status, time, grade)
data
  status time grade
1      1    3     5
2      2    4     6
data$time
[1] 3 4
data[1,2]
[1] 3
```

Importing data into R

The most important functions for reading data are:

- scan()
  Used to read a sequence of data elements. This is a very general input method and you *could* decide to use only this one. However, in many cases the function below is easier to use.

- read.table()
  Used to read a data sheet (stored as a text file) into a data frame in R. You will probably most often want to use this one.

read.table()

```
# Reading a table with no header line and white space delimiters
data <- read.table(mydata.txt)

# Reading a table with a header line
data <- read.table(mydata.txt, header=TRUE)

# Reading a table with header and tab-delimited elements
data <- read.table(mydata.txt, header=TRUE, sep=\t)
```

Note: a column in the file with one or more character elements will be read into R as a factor. Factors are interpreted in a special way by many functions in R.

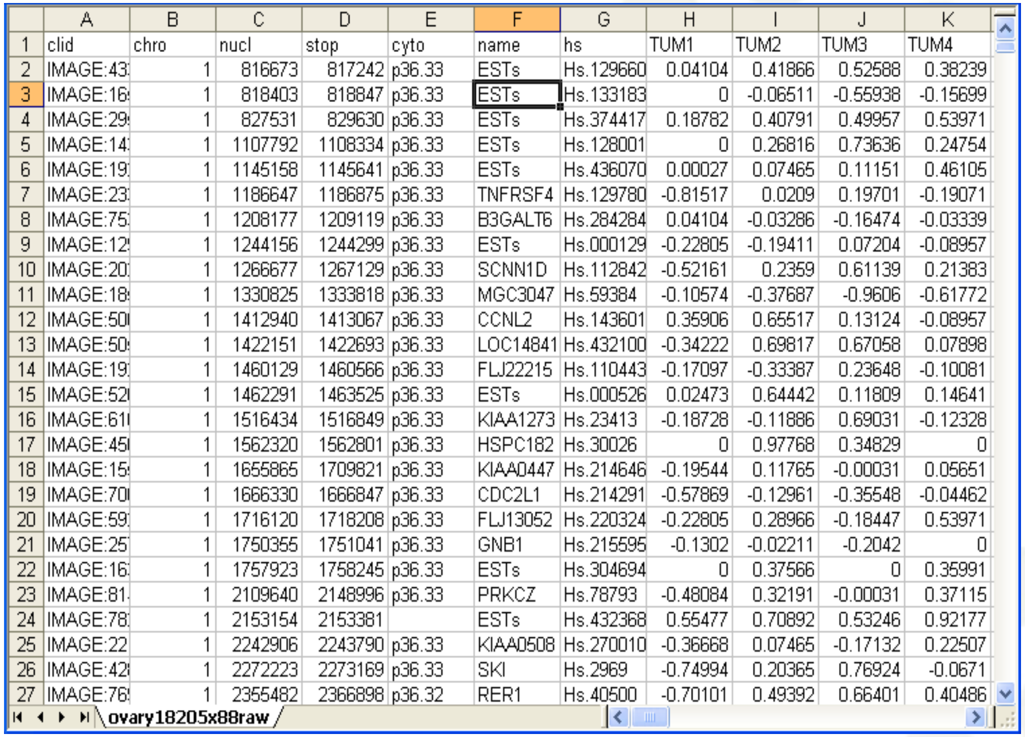
Note2: failing to declare that a header line is present is likely to lead to a data frame with more factors than you intended.

The general form of read.table()

```
read.table(file,
          header = FALSE,
          sep = ,
          quote = \',
          dec = .,
          row.names,
          col.names,
          as.is = !stringsAsFactors,
          na.strings = NA,
          colClasses = NA,
          nrows = -1,
          skip = 0,
..
          check.names = TRUE,
          fill = !blank.lines.skip,
          strip.white = FALSE,
          blank.lines.skip = TRUE,
          comment.char = #,
          allowEscapes = FALSE,
          flush = FALSE,
          stringsAsFactors = default.stringsAsFactors(),
          encoding = unknown)
```

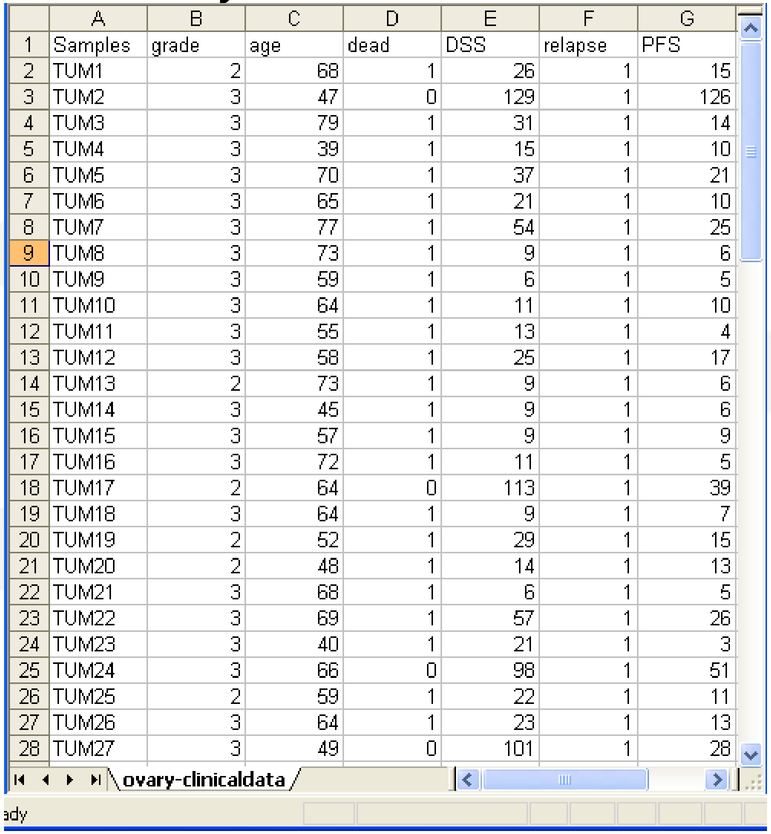
Example 1: use of read.table()

ovary17058x88raw.txt



	A	B	C	D	E	F	G	H	I	J	K
1	clid	chro	nucl	stop	cyto	name	hs	TUM1	TUM2	TUM3	TUM4
2	IMAGE:43	1	816673	817242	p36.33	ESTs	Hs.129660	0.04104	0.41866	0.52588	0.38239
3	IMAGE:16	1	818403	818847	p36.33	ESTs	Hs.133183	0	-0.06511	-0.55938	-0.15699
4	IMAGE:29	1	827531	829630	p36.33	ESTs	Hs.374417	0.18782	0.40791	0.49957	0.53971
5	IMAGE:14	1	1107792	1108334	p36.33	ESTs	Hs.128001	0	0.26816	0.73636	0.24754
6	IMAGE:19	1	1145158	1145641	p36.33	ESTs	Hs.436070	0.00027	0.07465	0.11151	0.46105
7	IMAGE:23	1	1186647	1186875	p36.33	TNFRSF4	Hs.129780	-0.81517	0.0209	0.19701	-0.19071
8	IMAGE:75	1	1208177	1209119	p36.33	B3GALT6	Hs.284284	0.04104	-0.03286	-0.16474	-0.03339
9	IMAGE:12	1	1244156	1244299	p36.33	ESTs	Hs.000129	-0.22805	-0.19411	0.07204	-0.08957
10	IMAGE:20	1	1266677	1267129	p36.33	SCNN1D	Hs.112842	-0.52161	0.2359	0.61139	0.21383
11	IMAGE:18	1	1330825	1333818	p36.33	MGC3047	Hs.59384	-0.10574	-0.37687	-0.9606	-0.61772
12	IMAGE:50	1	1412940	1413067	p36.33	CCNL2	Hs.143601	0.35906	0.65517	0.13124	-0.08957
13	IMAGE:50	1	1422151	1422693	p36.33	LOC14841	Hs.432100	-0.34222	0.69817	0.67058	0.07898
14	IMAGE:19	1	1460129	1460566	p36.33	FLJ22215	Hs.110443	-0.17097	-0.33387	0.23648	-0.10081
15	IMAGE:52	1	1462291	1463525	p36.33	ESTs	Hs.000526	0.02473	0.64442	0.11809	0.14641
16	IMAGE:61	1	1516434	1516849	p36.33	KIAA1273	Hs.23413	-0.18728	-0.11886	0.69031	-0.12328
17	IMAGE:45	1	1562320	1562801	p36.33	HSPC182	Hs.30026	0	0.97768	0.34829	0
18	IMAGE:15	1	1655865	1709821	p36.33	KIAA0447	Hs.214646	-0.19544	0.11765	-0.00031	0.05651
19	IMAGE:70	1	1666330	1666847	p36.33	CDC2L1	Hs.214291	-0.57869	-0.12961	-0.35548	-0.04462
20	IMAGE:59	1	1716120	1718208	p36.33	FLJ13052	Hs.220324	-0.22805	0.28966	-0.18447	0.53971
21	IMAGE:25	1	1750355	1751041	p36.33	GNB1	Hs.215595	-0.1302	-0.02211	-0.2042	0
22	IMAGE:16	1	1757923	1758245	p36.33	ESTs	Hs.304694	0	0.37566	0	0.35991
23	IMAGE:81	1	2109640	2148996	p36.33	PRKCZ	Hs.78793	-0.48084	0.32191	-0.00031	0.37115
24	IMAGE:78	1	2153154	2153381		ESTs	Hs.432368	0.55477	0.70892	0.53246	0.92177
25	IMAGE:22	1	2242906	2243790	p36.33	KIAA0508	Hs.270010	-0.36668	0.07465	-0.17132	0.22507
26	IMAGE:42	1	2272223	2273169	p36.33	SKI	Hs.2969	-0.74994	0.20365	0.76924	-0.0671
27	IMAGE:76	1	2355482	2366898	p36.32	RER1	Hs.40500	-0.70101	0.49392	0.66401	0.40486

ovary-clinicaldata.txt

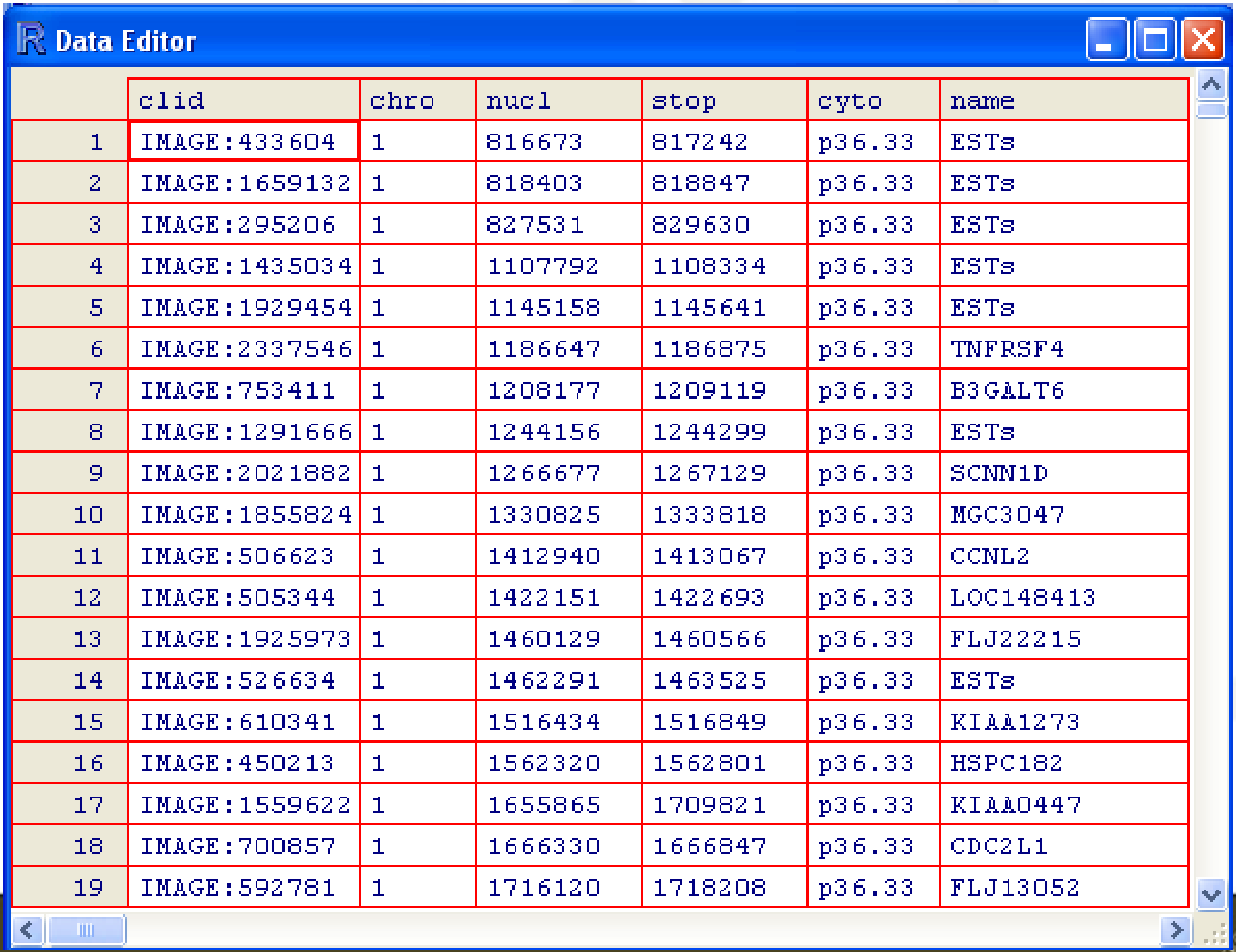


	A	B	C	D	E	F	G
1	Samples	grade	age	dead	DSS	relapse	PFS
2	TUM1	2	68	1	26	1	15
3	TUM2	3	47	0	129	1	126
4	TUM3	3	79	1	31	1	14
5	TUM4	3	39	1	15	1	10
6	TUM5	3	70	1	37	1	21
7	TUM6	3	65	1	21	1	10
8	TUM7	3	77	1	54	1	25
9	TUM8	3	73	1	9	1	6
10	TUM9	3	59	1	6	1	5
11	TUM10	3	64	1	11	1	10
12	TUM11	3	55	1	13	1	4
13	TUM12	3	58	1	25	1	17
14	TUM13	2	73	1	9	1	6
15	TUM14	3	45	1	9	1	6
16	TUM15	3	57	1	9	1	9
17	TUM16	3	72	1	11	1	5
18	TUM17	2	64	0	113	1	39
19	TUM18	3	64	1	9	1	7
20	TUM19	2	52	1	29	1	15
21	TUM20	2	48	1	14	1	13
22	TUM21	3	68	1	6	1	5
23	TUM22	3	69	1	57	1	26
24	TUM23	3	40	1	21	1	3
25	TUM24	3	66	0	98	1	51
26	TUM25	2	59	1	22	1	11
27	TUM26	3	64	1	23	1	13
28	TUM27	3	49	0	101	1	28

```
# Set directory
setwd('C:/Ole Chr/DNR/R Course 2007')

# Read both tables
cgh <- read.table(ovary17058x88raw.txt, header=T, sep=\t)
clin <- read.table(ovary-clinicaldata.txt, header=T, sep=\t)

# What are the columns of the clinical data table?
names(clin)
[1] Samples grade age dead DSS relapse PFS

# What are the dimensions of the data tables?
dim(cgh)
[1] 17058    95
dim(clin)
[1] 88  7
```

Tabular view of a data frame

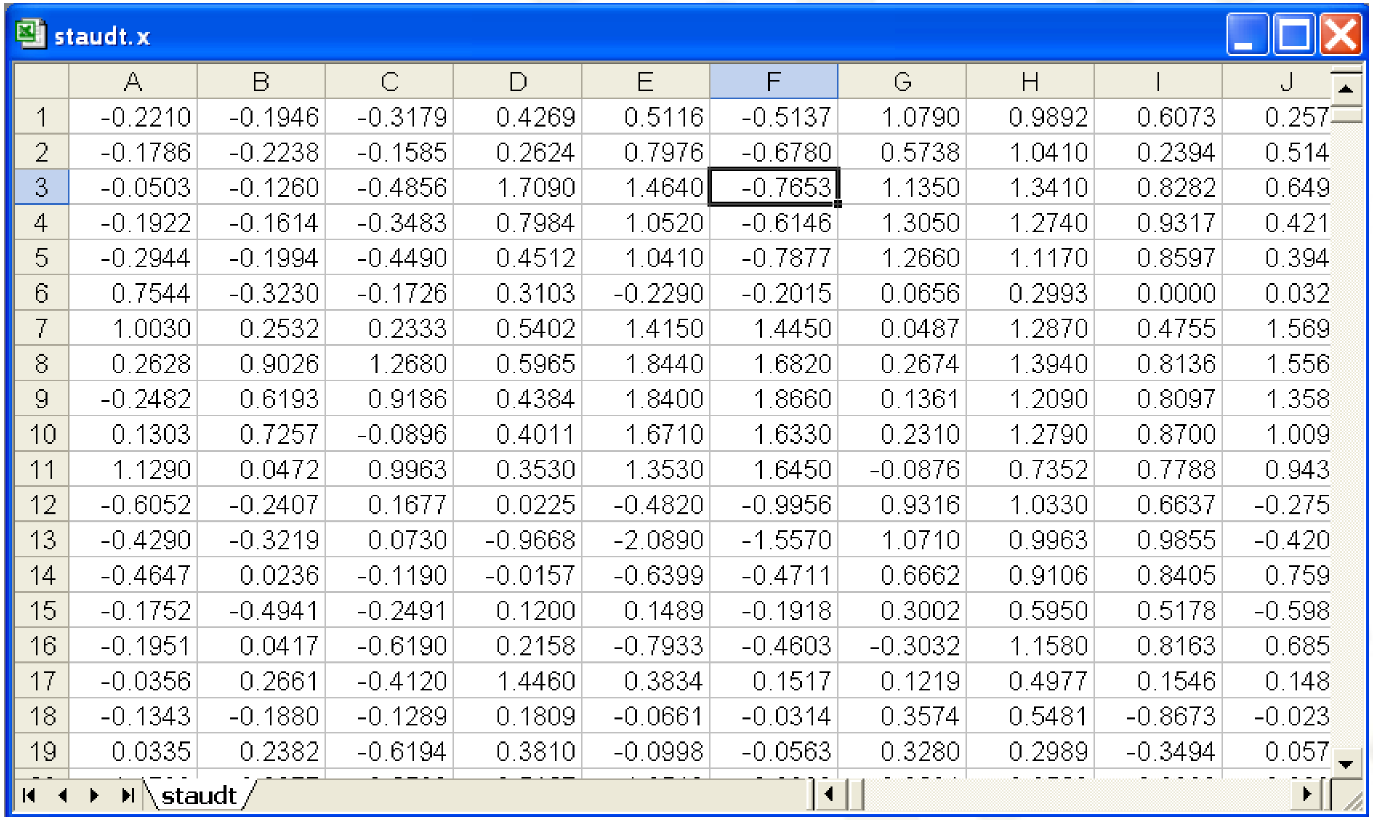
```
fix(cgh)
```



	clid	chro	nucl	stop	cyto	name
1	IMAGE:433604	1	816673	817242	p36.33	ESTs
2	IMAGE:1659132	1	818403	818847	p36.33	ESTs
3	IMAGE:295206	1	827531	829630	p36.33	ESTs
4	IMAGE:1435034	1	1107792	1108334	p36.33	ESTs
5	IMAGE:1929454	1	1145158	1145641	p36.33	ESTs
6	IMAGE:2337546	1	1186647	1186875	p36.33	TNFRSF4
7	IMAGE:753411	1	1208177	1209119	p36.33	B3GALT6
8	IMAGE:1291666	1	1244156	1244299	p36.33	ESTs
9	IMAGE:2021882	1	1266677	1267129	p36.33	SCNN1D
10	IMAGE:1855824	1	1330825	1333818	p36.33	MGC3047
11	IMAGE:506623	1	1412940	1413067	p36.33	CCNL2
12	IMAGE:505344	1	1422151	1422693	p36.33	LOC148413
13	IMAGE:1925973	1	1460129	1460566	p36.33	FLJ22215
14	IMAGE:526634	1	1462291	1463525	p36.33	ESTs
15	IMAGE:610341	1	1516434	1516849	p36.33	KIAA1273
16	IMAGE:450213	1	1562320	1562801	p36.33	HSPC182
17	IMAGE:1559622	1	1655865	1709821	p36.33	KIAA0447
18	IMAGE:700857	1	1666330	1666847	p36.33	CDC2L1
19	IMAGE:592781	1	1716120	1718208	p36.33	FLJ13052

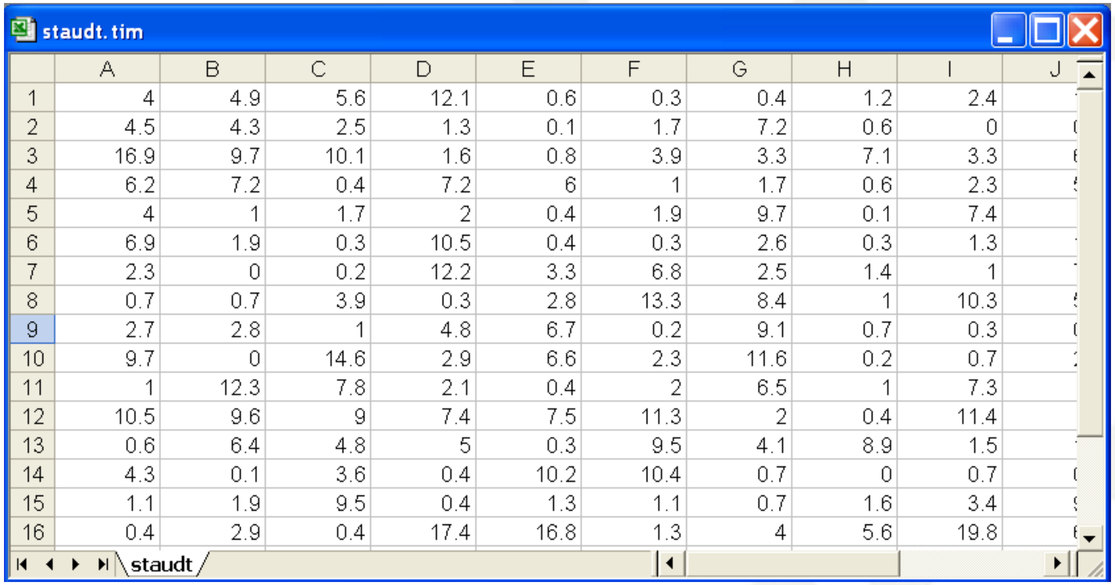
Example 2: use of scan()

Expression data (staudt.x):



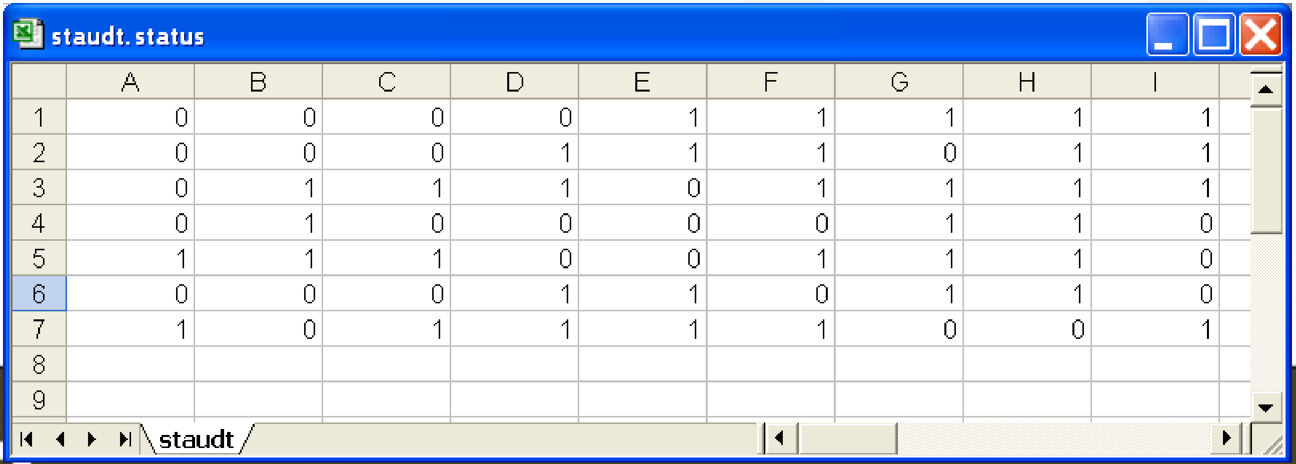
	A	B	C	D	E	F	G	H	I	J
1	-0.2210	-0.1946	-0.3179	0.4269	0.5116	-0.5137	1.0790	0.9892	0.6073	0.257
2	-0.1786	-0.2238	-0.1585	0.2624	0.7976	-0.6780	0.5738	1.0410	0.2394	0.514
3	-0.0503	-0.1260	-0.4856	1.7090	1.4640	-0.7653	1.1350	1.3410	0.8282	0.649
4	-0.1922	-0.1614	-0.3483	0.7984	1.0520	-0.6146	1.3050	1.2740	0.9317	0.421
5	-0.2944	-0.1994	-0.4490	0.4512	1.0410	-0.7877	1.2660	1.1170	0.8597	0.394
6	0.7544	-0.3230	-0.1726	0.3103	-0.2290	-0.2015	0.0656	0.2993	0.0000	0.032
7	1.0030	0.2532	0.2333	0.5402	1.4150	1.4450	0.0487	1.2870	0.4755	1.569
8	0.2628	0.9026	1.2680	0.5965	1.8440	1.6820	0.2674	1.3940	0.8136	1.556
9	-0.2482	0.6193	0.9186	0.4384	1.8400	1.8660	0.1361	1.2090	0.8097	1.358
10	0.1303	0.7257	-0.0896	0.4011	1.6710	1.6330	0.2310	1.2790	0.8700	1.009
11	1.1290	0.0472	0.9963	0.3530	1.3530	1.6450	-0.0876	0.7352	0.7788	0.943
12	-0.6052	-0.2407	0.1677	0.0225	-0.4820	-0.9956	0.9316	1.0330	0.6637	-0.275
13	-0.4290	-0.3219	0.0730	-0.9668	-2.0890	-1.5570	1.0710	0.9963	0.9855	-0.420
14	-0.4647	0.0236	-0.1190	-0.0157	-0.6399	-0.4711	0.6662	0.9106	0.8405	0.759
15	-0.1752	-0.4941	-0.2491	0.1200	0.1489	-0.1918	0.3002	0.5950	0.5178	-0.598
16	-0.1951	0.0417	-0.6190	0.2158	-0.7933	-0.4603	-0.3032	1.1580	0.8163	0.685
17	-0.0356	0.2661	-0.4120	1.4460	0.3834	0.1517	0.1219	0.4977	0.1546	0.148
18	-0.1343	-0.1880	-0.1289	0.1809	-0.0661	-0.0314	0.3574	0.5481	-0.8673	-0.023
19	0.0335	0.2382	-0.6194	0.3810	-0.0998	-0.0563	0.3280	0.2989	-0.3494	0.057

Survival data (staudt.tim, staudt.status):



staudt.tim

	A	B	C	D	E	F	G	H	I
1	4	4.9	5.6	12.1	0.6	0.3	0.4	1.2	2.4
2	4.5	4.3	2.5	1.3	0.1	1.7	7.2	0.6	0
3	16.9	9.7	10.1	1.6	0.8	3.9	3.3	7.1	3.3
4	6.2	7.2	0.4	7.2	6	1	1.7	0.6	2.3
5	4	1	1.7	2	0.4	1.9	9.7	0.1	7.4
6	6.9	1.9	0.3	10.5	0.4	0.3	2.6	0.3	1.3
7	2.3	0	0.2	12.2	3.3	6.8	2.5	1.4	1
8	0.7	0.7	3.9	0.3	2.8	13.3	8.4	1	10.3
9	2.7	2.8	1	4.8	6.7	0.2	9.1	0.7	0.3
10	9.7	0	14.6	2.9	6.6	2.3	11.6	0.2	0.7
11	1	12.3	7.8	2.1	0.4	2	6.5	1	7.3
12	10.5	9.6	9	7.4	7.5	11.3	2	0.4	11.4
13	0.6	6.4	4.8	5	0.3	9.5	4.1	8.9	1.5
14	4.3	0.1	3.6	0.4	10.2	10.4	0.7	0	0.7
15	1.1	1.9	9.5	0.4	1.3	1.1	0.7	1.6	3.4
16	0.4	2.9	0.4	17.4	16.8	1.3	4	5.6	19.8



staudt.status

	A	B	C	D	E	F	G	H	I
1	0	0	0	0	1	1	1	1	1
2	0	0	0	1	1	1	0	1	1
3	0	1	1	1	0	1	1	1	1
4	0	1	0	0	0	0	1	1	0
5	1	1	1	0	0	1	1	1	0
6	0	0	0	1	1	0	1	1	0
7	1	0	1	1	1	1	0	0	1
8									
9									

Loading the data into R using scan()

```
setwd('C:/Ole Chr/DNR/R Course 2007')
getwd()
[1] C:/Ole Chr/DNR/R Course 2007

x <- matrix(scan(staudt.x), ncol=240, byrow=TRUE)
Read 1775760 items

death <- scan(staudt.tim)
Read 240 items

status <- scan(staudt.status)
Read 240 items

genenames <- paste(Gene, 1:nrow(staudt.x), sep= )
```

Looking at a data set in R

```
# The columns of the clinical table of the ovarian data
names(clin)
[1] Samples grade age dead DSS relapse PFS

# What is the distribution of grades?
table(clin$grade)
 1  2  3
 6 22 60

# What is the proportion of censored survival times (DSS)?
sum(clin$dead==0) / nrow(clin)
[1] 0.2840909

# The proportion of patients with relapse for which death is observed
sum(clin$dead[clin$relapse==1]==1) / sum(clin$relapse==1)
[1] 0.8181818
```

Grouped data

Suppose patients are divided into two groups. For the sake of the argument, let us do this now by splitting patients into two groups based on tumor grade:

```
# Extract gene data
expr <- cgh[,8:95]

# Define two logical vectors that define the groups
g1 <- clin$grade < 3
g2 <- !g1

# Extract cgh data for each group
cgh1 <- cgh[, g1]     # Select data for the patients in group 1
cgh2 <- cgh[, g2]     # Select data for the patients in group 2
```

Fold change

Select genes with $|\bar{x}_1 - \bar{x}_2| > \log k$

```
compare <- function(x, g) {
   abs(mean(x[g]) - mean(x[!g]))
}

k <- 2      # Number of folds
absdist <- apply(cgh, 1, compare, g1)
cgh$clid[absdist > log2(2)]

[1] Gene 4131
```

In this case, only one gene had a two-fold change in expression between the two groups.

Two-sample t-test

Normal samples, equal group variances.

```
mytest <- function(x,g) {
  t.test(x[g], x[!g], var.equal=TRUE)$p.value
}

pvalues <- apply(cgh, 1, mytest, g1)
fdrvalues <- p.adjust(pvalues, method=BH)
cgh$clid[fdrvalues < 0.1]

[1] Gene 31   Gene 32   Gene 50 .....
```

Welch's test

Normal samples, unequal group variances

```
test <- function(x,g) {
  t.test(x[g], x[!g])$p.value
}
pvalues <- apply(x, 1, test, group1)
fdrvalues <- p.adjust(pvalues, method=BH)
genenames[fdrvalues < 0.1]
```

Wilcoxon rank sum test

For non-normal data where the distributions of the two groups are identical except for a location effect. Can be used in a wide range of situations, but are less powerful than parametric counterparts. With small sample sizes, it is hard to get small p-values.

```
test <- function(x,g) {
  wilcox.test(x[g], x[!g], var.equal=TRUE)$p.value
}
pvalues <- apply(x, 1, test, group1)
fdrvalues <- p.adjust(pvalues, method=BH)
genenames[fdrvalues < 0.1]
```

SAM t-test

For small sample sizes, the t statistic tends to be highly correlated with the s.e. term in the denominator. Thus low-variance genes are more easily picked up than high-variance genes.

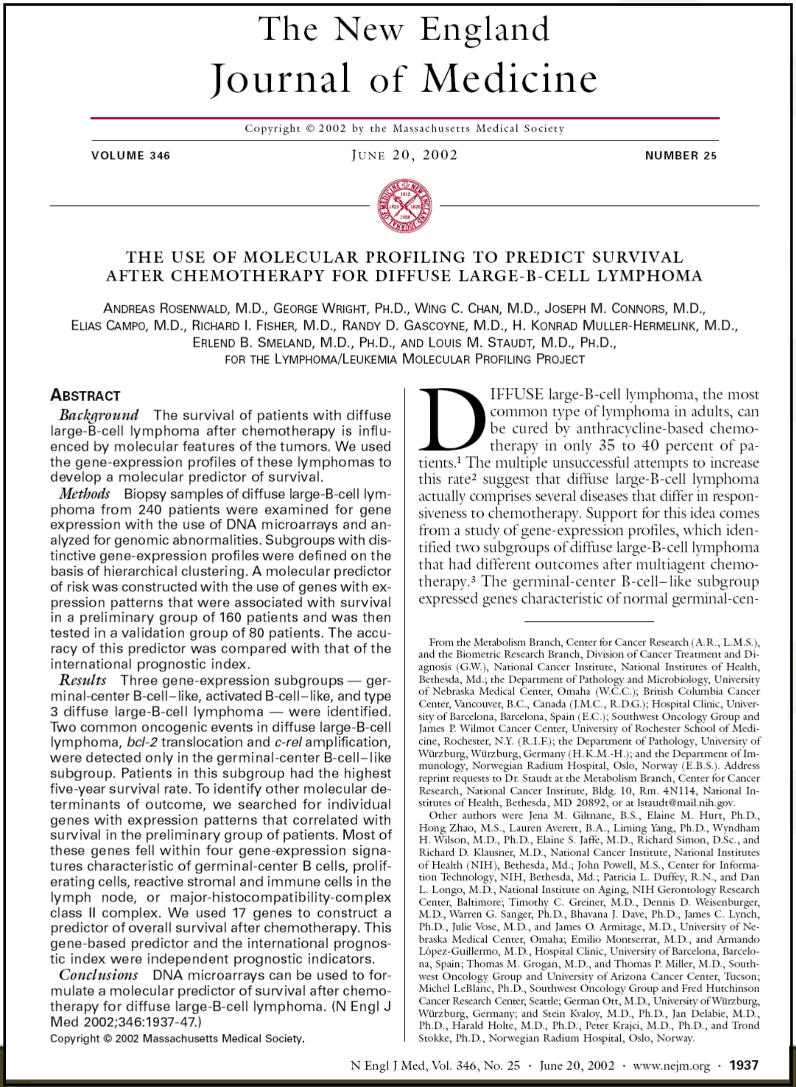
In a SAM t-test, a small *fudge factor* is added to the denominator of the t statistic. That reduces the undesirable phenomenon above.

The statistic no longer has a t-distribution under the null hypothesis, so a permutation procedure is used to obtain the significance.

SAM t-test

```
install.packages(samr)
library(samr)
group <- ifelse(group1, 1, 2)
geneid <- as.character(1:nrow(x))
data <- list(x=staudt.x, y=group, geneid=geneid,
                        genenames=genenames, logged2=TRUE)
ans <- samr(data, resp.type=Two class unpaired)
delta.table <- samr.compute.delta.table(ans)
siggenes.table <- samr.compute.siggenes.table(ans, 3, data, delta.table)
samr.plot(ans, 3)
```

The Rosenwald lymphoma data



The New England
Journal of Medicine

Copyright © 2002 by the Massachusetts Medical Society

VOLUME 346 June 20, 2002 NUMBER 25

THE USE OF MOLECULAR PROFILING TO PREDICT SURVIVAL AFTER CHEMOTHERAPY FOR DIFFUSE LARGE-B-CELL LYMPHOMA

ANDREAS ROSENWALD, M.D., GEORGE WRIGHT, PH.D., WING C. CHAN, M.D., JOSEPH M. CONNORS, M.D.,
ELIAS CAMPO, M.D., RICHARD I. FISHER, M.D., RANDY D. GASCOYNE, M.D., H. KONRAD MULLER-HERMELINK, M.D.,
ERLEND B. SMELAND, M.D., PH.D., AND LOUIS M. STAUDT, M.D., PH.D.,
FOR THE LYMPHOMA/LEUKEMIA MOLECULAR PROFILING PROJECT

ABSTRACT

Background The survival of patients with diffuse large-B-cell lymphoma after chemotherapy is influenced by molecular features of the tumors. We used the gene-expression profiles of these lymphomas to develop a molecular predictor of survival.

Methods Biopsy samples of diffuse large-B-cell lymphoma from 240 patients were examined for gene expression with the use of DNA microarrays and analyzed for genomic abnormalities. Subgroups with distinctive gene-expression profiles were defined on the basis of hierarchical clustering. A molecular predictor of risk was constructed with the use of genes with expression patterns that were associated with survival in a preliminary group of 160 patients and was then tested in a validation group of 80 patients. The accuracy of this predictor was compared with that of the international prognostic index.

Results Three gene-expression subgroups — germinal-center B-cell–like, activated B-cell–like, and type 3 diffuse large-B-cell lymphoma — were identified. Two common oncogenic events in diffuse large-B-cell lymphoma, *bcl-2* translocation and *c-rel* amplification, were detected only in the germinal-center B-cell–like subgroup. Patients in this subgroup had the highest five-year survival rate. To identify other molecular determinants of outcome, we searched for individual genes with expression patterns that correlated with survival in the preliminary group of patients. Most of these genes fell within four gene-expression signatures characteristic of germinal-center B cells, proliferating cells, reactive stromal and immune cells in the lymph node, or major-histocompatibility-complex class II complex. We used 17 genes to construct a predictor of overall survival after chemotherapy. This gene-based predictor and the international prognostic index were independent prognostic indicators.

Conclusions DNA microarrays can be used to formulate a molecular predictor of survival after chemotherapy for diffuse large-B-cell lymphoma. (N Engl J Med 2002;346:1937-47.)

DIFFUSE large-B-cell lymphoma, the most common type of lymphoma in adults, can be cured by anthracycline-based chemotherapy in only 35 to 40 percent of patients.[1] The multiple unsuccessful attempts to increase this rate[2] suggest that diffuse large-B-cell lymphoma actually comprises several diseases that differ in responsiveness to chemotherapy. Support for this idea comes from a study of gene-expression profiles, which identified two subgroups of diffuse large-B-cell lymphoma that had different outcomes after multiagent chemotherapy.[3] The germinal-center B-cell–like subgroup expressed genes characteristic of normal germinal-cen-

From the Metabolism Branch, Center for Cancer Research (A.R., L.M.S.), and the Biometric Research Branch, Division of Cancer Treatment and Diagnosis (G.W.), National Cancer Institute, National Institutes of Health, Bethesda, Md.; the Department of Pathology and Microbiology, University of Nebraska Medical Center, Omaha (W.C.C.); British Columbia Cancer Center, Vancouver, B.C., Canada (J.M.C., R.D.G.); Hospital Clinic, University of Barcelona, Barcelona, Spain (E.C.); Southwest Oncology Group and James P. Wilmot Cancer Center, University of Rochester School of Medicine, Rochester, N.Y. (R.I.F.); the Department of Pathology, University of Würzburg, Würzburg, Germany (H.K.M.-H.); and the Department of Immunology, Norwegian Radium Hospital, Oslo, Norway (E.B.S.). Address reprint requests to Dr. Staudt at the Metabolism Branch, Center for Cancer Research, National Cancer Institute, Bldg. 10, Rm. 4N114, National Institutes of Health, Bethesda, MD 20892, or at lstaudt@mail.nih.gov.

Other authors were Jena M. Giltnane, B.S., Elaine M. Hurt, Ph.D., Hong Zhao, M.S., Lauren Averett, B.A., Liming Yang, Ph.D., Wyndham H. Wilson, M.D., Ph.D., Elaine S. Jaffe, M.D., Richard Simon, D.Sc., and Richard D. Klausner, M.D., National Cancer Institute, National Institutes of Health (NIH), Bethesda, Md.; John Powell, M.S., Center for Information Technology, NIH, Bethesda, Md.; Patricia L. Duffey, R.N., and Dan L. Longo, M.D., National Institute on Aging, NIH Gerontology Research Center, Baltimore; Timothy C. Greiner, M.D., Dennis D. Weisenburger, M.D., Warren G. Sanger, Ph.D., Bhavana J. Dave, Ph.D., James C. Lynch, Ph.D., Julie Vose, M.D., and James O. Armitage, M.D., University of Nebraska Medical Center, Omaha; Emilio Montserrat, M.D., and Armando López-Guillermo, M.D., Hospital Clinic, University of Barcelona, Barcelona, Spain; Thomas M. Grogan, M.D., and Thomas P. Miller, M.D., Southwest Oncology Group and University of Arizona Cancer Center, Tucson; Michel LeBlanc, Ph.D., Southwest Oncology Group and Fred Hutchinson Cancer Research Center, Seattle; German Ott, M.D., University of Würzburg, Würzburg, Germany; and Stein Kvaloy, M.D., Ph.D., Jan Delabie, M.D., Ph.D., Harald Holte, M.D., Ph.D., Peter Krajci, M.D., Ph.D., and Trond Stokke, Ph.D., Norwegian Radium Hospital, Oslo, Norway.

N Engl J Med, Vol. 346, No. 25 · June 20, 2002 · www.nejm.org · **1937**

Survival study of patients with diffuse large-B-cell lymphoma (DLBCL) after chemotherapy.

- Biopsies from 240 patients
- Expression data (7399 genes)
- Survival times
- Censoring status

Survival analysis

```
library(survival)

# Plot Kaplan-Meier curve:
plot(survfit(Surv(death, status)))

# Plot K-M curves for each group separately:
plot(survfit(Surv(death, status) ~ group))

# Logrank test for difference between groups:
survdiff(Surv(death, status)~group)
```

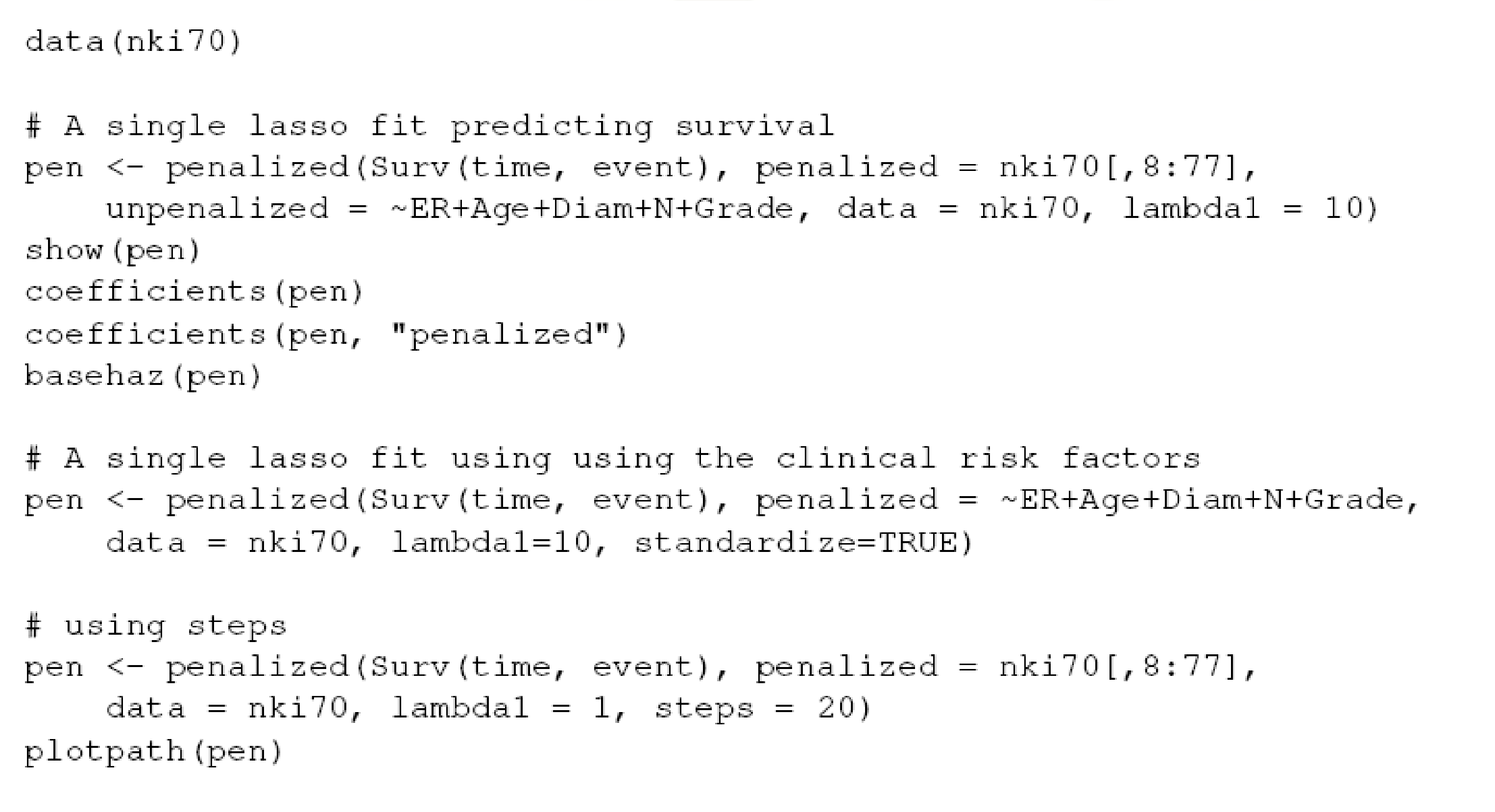
```
Call:
survdiff(formula = Surv(staudt.tim, staudt.status) ~ group)

          N Observed Expected (O-E)^2/E (O-E)^2/V
group=1 119      108     35.7     146.8       245
group=2 121       30    102.3      51.1       245

Chisq= 245  on 1 degrees of freedom, p= 0
```

SAM Cox score test

```
# Load SAM library
library(samr)

# Set up data set
data <- list(x=x, death=death, status=status, geneid=geneid,
           genenames=genenames, logged2=TRUE)

# Run SAM
ans <- samr(data, resp.type=Two class unpaired)

# Compute and view delta table
delta.table <- samr.compute.delta.table(ans)
fix(delta.table)

# Having decided on a delta value, identify significant genes
signif <- samr.compute.siggenes.table(ans, 3, data, delta.table)
fix(signif)
```

The Lasso

LASSO = Least Absolute Shrinkage and Selection Operator



```
data(nki70)

# A single lasso fit predicting survival
pen <- penalized(Surv(time, event), penalized = nki70[,8:77],
    unpenalized = ~ER+Age+Diam+N+Grade, data = nki70, lambda1 = 10)
show(pen)
coefficients(pen)
coefficients(pen, "penalized")
basehaz(pen)

# A single lasso fit using using the clinical risk factors
pen <- penalized(Surv(time, event), penalized = ~ER+Age+Diam+N+Grade,
    data = nki70, lambda1=10, standardize=TRUE)

# using steps
pen <- penalized(Surv(time, event), penalized = nki70[,8:77],
    data = nki70, lambda1 = 1, steps = 20)
plotpath(pen)
```