Lecture 4: Noise?

Last time

We started by taking up our discussion of developing a boxplot for more than a single variable, a graphic to summarize the shape of a 2-dimensional point cloud

We then examined tools for viewing (continuous) data in 2 or more dimensions, spending some time with projections and linked displays

We ended with some material for your (first) homework assignment -- The subject of graphics will not end here, however, in that we'll also examine spatial (map-based) data as well as text as data later in the term

Today

We will give you a little tour of R in an attempt to better prepare you for doing something really innovative with the Registrar's data! We'll talk a little about the history of R and then operations on vectors (we'll do something like this every other lecture or so)

Then, we'll look at inference, examining a simple randomized controlled trial that comes up in A/B testing -- We'll take a small historical detour and talk about the first such trial (which came up in a medical application)

If there's time, we'll talk a bit about random number generation!

A brief and abridged history of statistical computing...

Statistical Computing in the 1960's

A number of statistical systems and programs already existed; BMD and P-Stat were in current use and GenStat was being developed

These systems grew out of **specific application areas** and tended to offer **pre-packaged analyses**

At the time, most statistics researchers would not be directly involved in analyzing data; programmers (read graduate students) would do the grubby work when necessary

The systems like BMD and SAS, for example, and PStat, to some extent, and GenStat's another good example, **all grew up in environments where statisticians were required to do some fairly committed routine analysis**. So BMD, of course comes from a biomedical field; SAS from several areas but medical, again; and GenStat comes from an agricultural background.

Now in all those situations, the statistics groups, amongst other duties, were expected to be doing kind of analysis to order. You know, the data would come along from a experiment, or a clinical trial, or other sources, and as part of the job of the statisticians to produce analysis. **Now often the analysis that they produced were relatively predetermined, or at least that's how it worked out.**

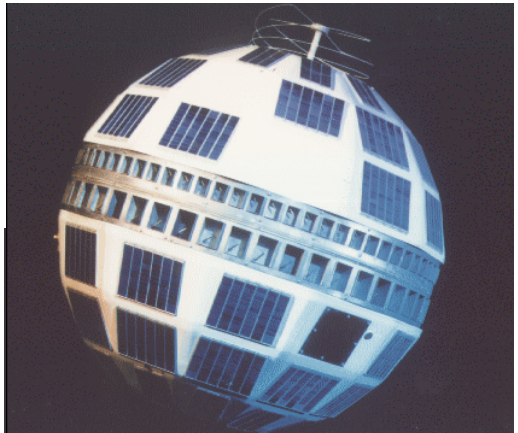
Interview with John Chambers, 2002

The mid 1960's at Bell Labs

Statistics Research at Bell Labs tackled large-scale data analysis projects with teams of researchers and "programmers"

Unlike much of the statistical computing of the day, this kind of work was not well suited to pre-packaged programs

Even then, AT&T was creating large-scale applications; data from TelStar, an early (1960s) communications satellite, involved tens of thousands of observations



Launched by NASA aboard a Delta rocket from Cape Canaveral on July 10, 1962, Telstar was the first privately sponsored space launch. A medium-altitude satellite, Telstar was placed in an elliptical orbit (completed once every 2 hours and 37 minutes), revolving at a 45 degree angle above the equator. Because of this, its transmission availability for transatlantic signals was only 20 minutes in each orbit.

Telstar relayed its first television pictures (of a flag outside its ground station in Andover, Maine) on the date of its launch. Almost two weeks later, on July 23, it relayed the first live transatlantic television signal. During that evening it also dealt with the first telephone call transmitted through space and successfully transmitted faxes, data, and both live and taped television, including the first live transmission of television across an ocean (to Pleumeur-Bodou, in France). John F. Kennedy, then President of the United States, gave a live transatlantic press conference via Telstar.

The mid 1960's at Bell Labs

During this period, John Tukey was also starting to formulate the beginnings of Exploratory Data Analysis

"Exploratory analysis is detective work -- numerical detective work -- or counting detective work -- or graphical detective work"

EDA is an approach to data analysis that postpones the usual assumptions about what kind of model the data follow with the more direct approach of allowing the data itself to reveal its underlying structure and model. EDA is not a mere collection of techniques; **EDA is a philosophy as to how we dissect a data set**; what we look for; how we look; and how we interpret.

Most EDA techniques are **graphical** in nature with a few quantitative techniques. The reason for the heavy reliance on graphics is that by its very nature **the main role of EDA is to open-mindedly explore**, and graphics gives the analysts unparalleled power to do so, enticing the data to reveal its structural secrets, and being always ready to gain some new, often unsuspected, insight into the data. In combination with the natural pattern-recognition capabilities that we all possess, graphics provides, of course, unparalleled power to carry this out.

Taken from `www.itl.nist.gov`

The mid 1960's at Bell Labs

> "Today, software and hardware together provide far more powerful factories than most statisticians realize, factories that many of today's most able young people find exciting and worth learning about on their own. Their interest can help us greatly, if statistics starts to make much more nearly adequate use of the computer. **However, if we fail to expand our uses, their interest in computers can cost us many of our best recruits, and set us back many years**."

The technical tools of Statistics, Nov 1964

The mid 1960's at Bell Labs

In previous decades, computers had matured significantly, but the access a user might have to these systems was limited; recall that in 1964, Bell Labs partnered with MIT and GE to create Multics (for Multiplexed Information and Computing Service)

> "Such systems must run continuously and reliably 7 days a week, 24 hours a day in a way similar to telephone or power systems, and must be capable of meeting wide service demands: from multiple man-machine interaction to the sequential processing of absentee-user jobs; from the use of the system with dedicated languages and subsystems to the programming of the system itself"

The mid 1960's at Bell Labs

While Bell Labs dropped out of the project in 1969, it did spark a lot of interest among researchers throughout the lab; John Chambers had just joined the lab and was starting to think about larger computing issues

He and a small group of statisticians began to consider how this kind of computing platform might benefit the practice of statistics

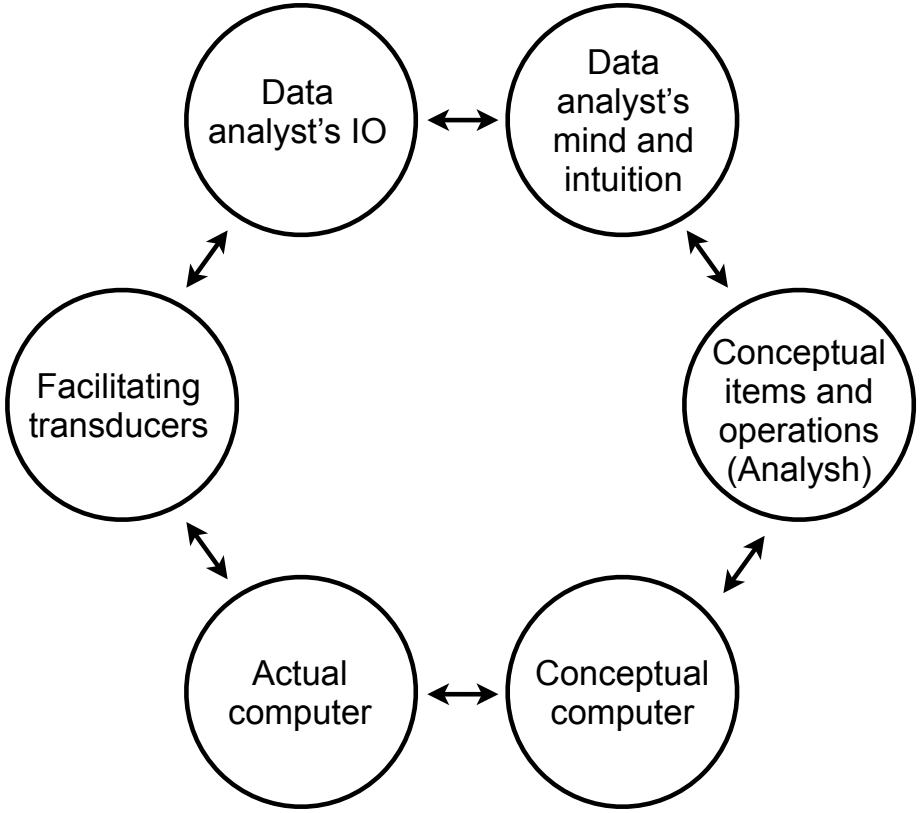
What should the computer do for us?

The mid 1960's at Bell Labs

What should the computer do for us?

In answering this question, the Bell Labs group considered the necessary components of a system that would "more naturally express what we do and allow us to interactively *program* analyses"

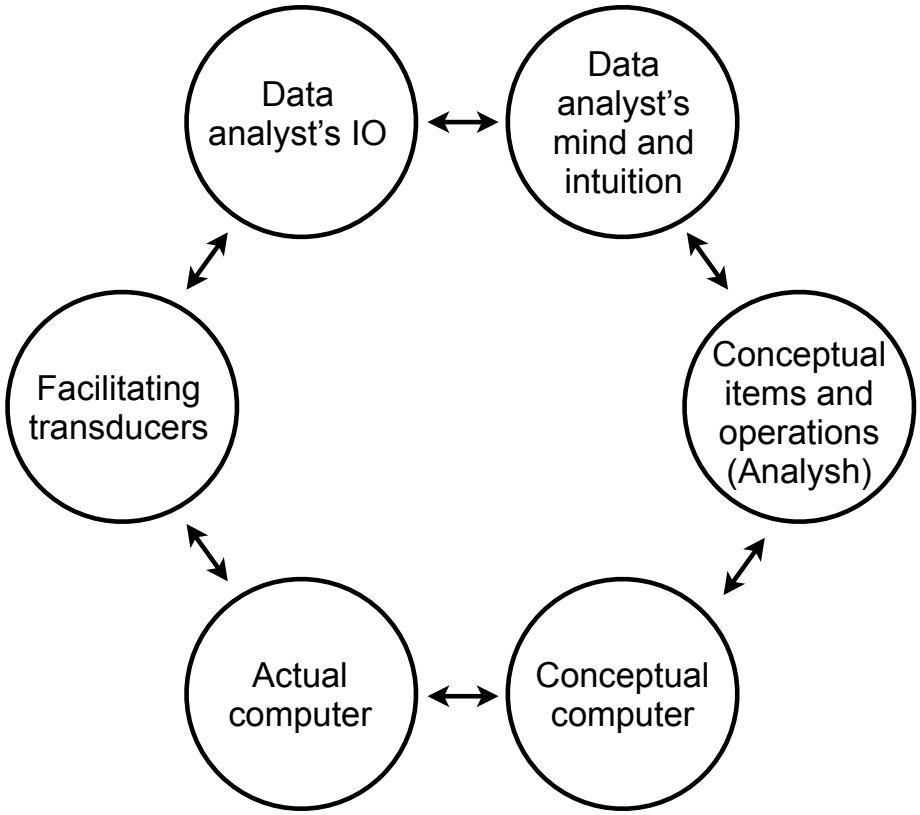
Tukey produced a memo that outlined the basic concepts; memos were the emails of the mid 1960s



Adapted from Chambers (2000)

Follow the arrows clockwise from the Mind and Intuition block. Tukey's notion is that data analysts have an arsenal of operations applicable to data, which they describe to themselves and to each other in a combination of mathematics and (English) words, for which he coins the term Analysh. These descriptions can be made into algorithms (my term, not his) -- specific computational methods, but not yet realized for an actual computer (hence the conceptual computer). Then a further mapping implements the algorithm, and running it produces output for the data analyst. The output, of course, stimulates further ideas and the cycle continues. (The facilitating transducers I interpret to mean software that allows information to be translated back and forth between internal machine form and forms that humans can write or look at -- a transducer, in general, converts energy from one form to another. So parsers and formatting software would be examples.)

Taken from Chambers (2000)



Adapted from Chambers (2000)

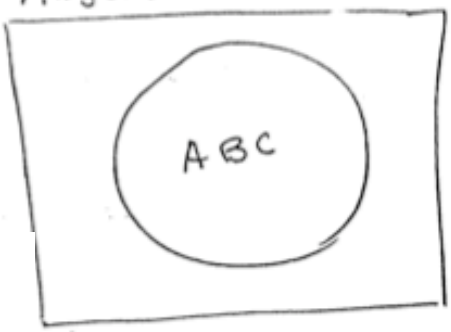
The mid 1960's at Bell Labs

This group met again in the summer of 1965; Martin Wilk opened the meeting with the following

> "What do we want? We want to have easy, flexible, availability of basic or higher level operations, with convenient data manipulation, bookkeeping and IO capacity. We want to be able easily to modify data, output formats, small and large programs and to do this and more with a standard language adapted to statistical usage."

Unfortunately, the plans developed by this group remained just that, plans; it wasn't until a decade later that a second group met, and on **May 5, 1976, work started on what would become "S"**

JHC ① Algorithm Interface 5/5/76

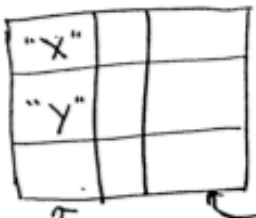


ABC: general (FORTRAN) algorithm

XABC: FORTRAN subroutine to provide interface between ABC & Language and/or utility programs

XABC (INSTR, OUTSTR)

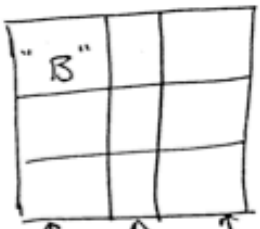
Input INSTR →



Pointers/Values

Argument Names or Blank

OUTSTR →



Pointers / Values

Types (Modes)

Result Names

Note: Names are meaningful to Algorithm, not necessarily to Language

The development of S

The image on the previous page is a scan of the first graphic produced at the May 5 meeting, and according to Chambers:

> “The upper portion depicts the concept of an interface between a proposed user-level language and an ‘algorithm,’ which meant a Fortran-callable subroutine. The lower portion of the figure shows diagrammatically a hierarchical, list-like structure for data, the direct forerunner of lists with named components, structures with attributes and eventually classes with slots... ***The two portions of the sketch in fact lead to the themes of function calls and objects, in retrospect.***

The development of S

By the end of 1976, John Chambers and Rick Becker had produced a working version of their sketch

The program was initially referred to as “the system” locally, and attempts to come up with new names yielded lots of “unacceptable” candidates, but all of which had the letter “S” in common; and so, given the precedent of the recently-developed C language...

Unix and S

Becker and Chambers made the most of the Unix project happening in parallel at the labs at the same time; by creating **a Unix version of S** (S Version 2), **their system became portable** (or at least it could go anywhere Unix could)

At that point, **AT&T started licensing both Unix and S**, with both university groups and “third-party” resellers in mind; this meant others could contribute to the development of the software

The development of S

About a decade after the first version of S, a complete revision of the language took place (S3, described in the “blue book”) which gave rise to an extensive modeling effort (described in the “white book”); for most of you, this version of S will most closely resemble R (to come)

One more decade and another version (S4, described in Chambers’ book “Computing with Data”) that brought with it a new class and method system

In 1998 Chambers won the ACM (Association for Computing Machinery) Software System Award; **S has “forever altered the way people analyze, visualize and manipulate data”**

And finally...

Ross Ihaka and Robert Gentleman (both at the University of Auckland at the time) wrote a reduced version of S for “teaching purposes”

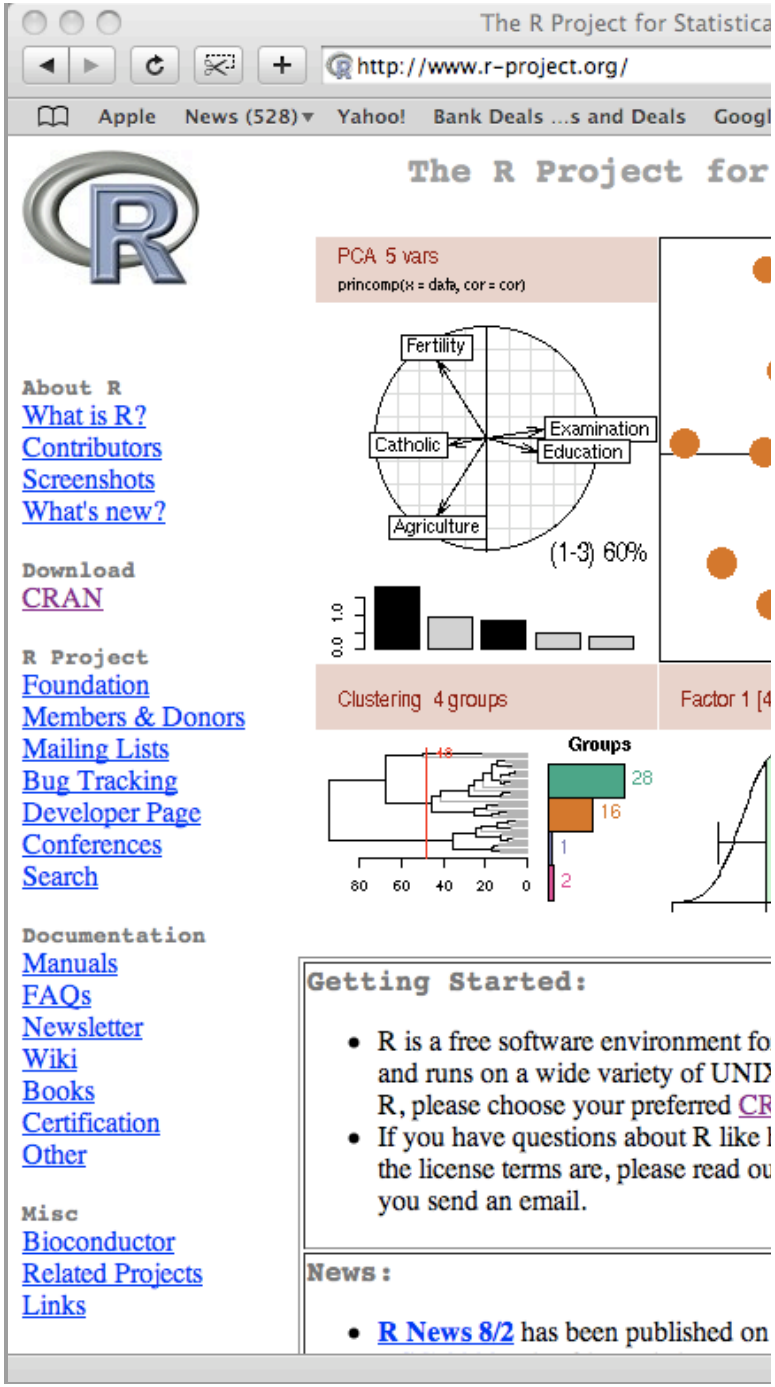
In 1995 Ross and Robert were persuaded to release the code for R under the GPL (the General Public License; it allows you to use it freely, distributed it, and even sell it, as long as the receiver has the same rights and the source code is freely available)

R is now administered by a core group of about a dozen people; but many more contribute code, and many many more program in R

The R environment

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

- *an effective data handling and storage facility,*
- *a suite of operators for calculations on arrays, in particular matrices,*
- *a large, coherent, integrated collection of intermediate tools for data analysis,*
- *graphical facilities for data analysis and display either on-screen or on hardcopy, and*
- *a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.*

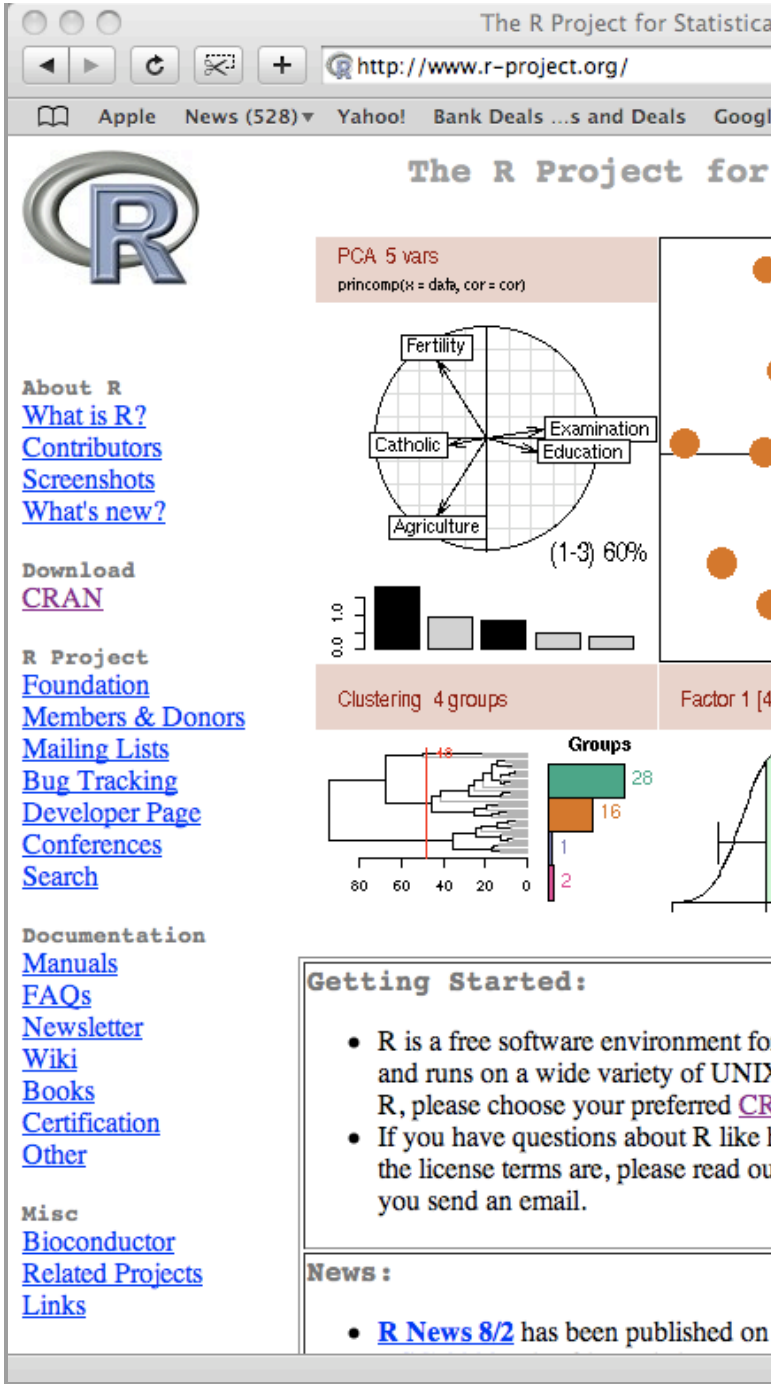


The R environment

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made.

Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages.



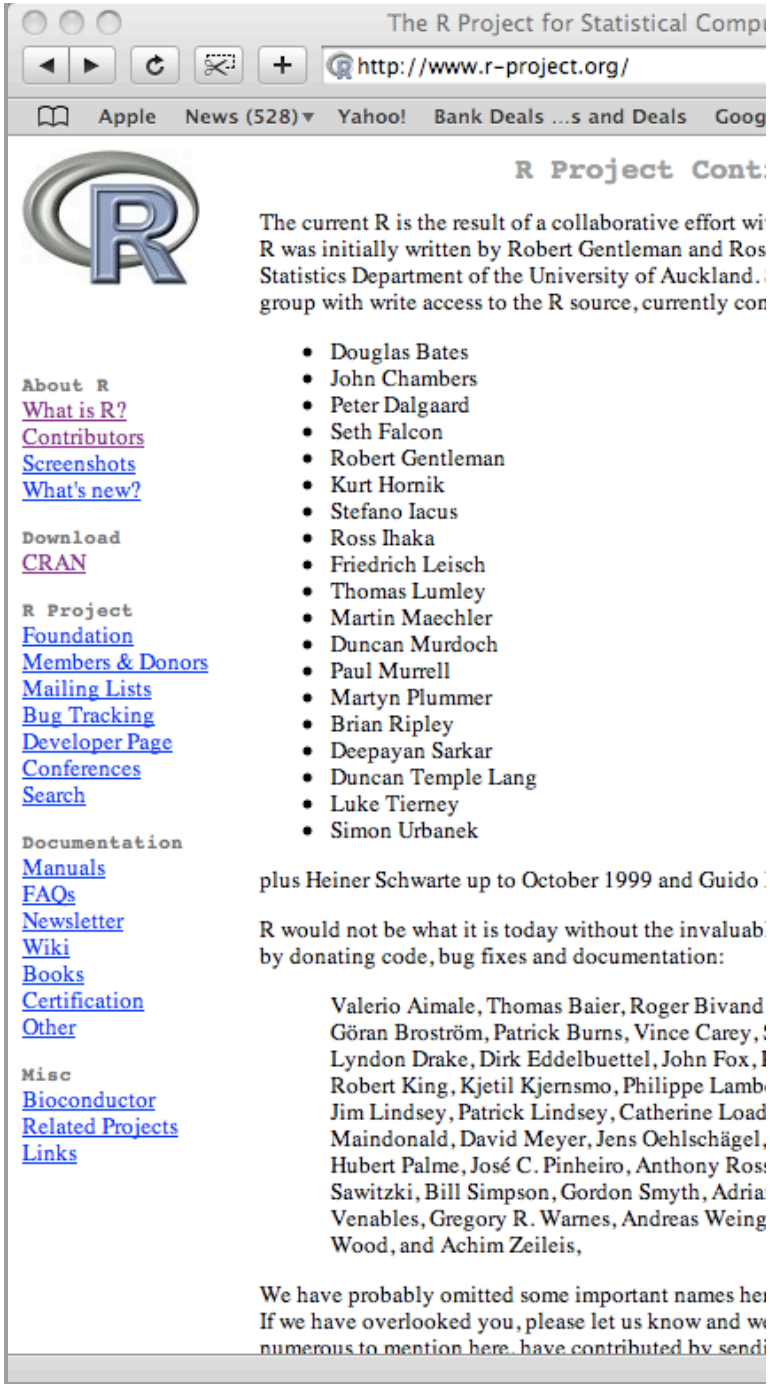
The R environment

While there is a fair bit of difference between how R and S are implemented, one of the most visible for users involves how code is shared

> *The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.*

In my mind, the ease with which packages can be published, accessed (installed) and auditioned is one of the real innovations of R

As an aside, what do (nearly) all of the names on the right have in common?



From the top...

At it's most basic level, R can be thought of as a bulky calculator; it can perform basic arithmetic and has a number of built-in functions

```
> 5+7             #addition
[1] 12

> 0.2*7.58        #multiplicaiton
[1] 1.516

> cos(2)
[1] -0.4161468

> cos(exp(4))     #compound
[1] -0.3706617

> sqrt(25)
[1] 5
```

In each case, R prints the result of the numeric operation to your screen; each output is prefaced by "`[1]`"; we will have more to say about why that is shortly.

For the moment, the important thing to realize is that you are entering mathematical expressions, having them evaluated and then R is printing the result to your screen.

Learning R

Learning R is a mix of remembering functions and knowing how to get help when you don't; R has a pretty extensive (and largely usable) help system that can answer questions about its functionality

For details on a function you can type

```
> ?cos
```

```
> help("cos")
```

```
> help.search("cos")
```

More function calls; `help` takes a "string" argument, the name of the function we'd like help on, while `help.search` performs fuzzy or regular expression matching for the string you provided

Trigonometric Functions

Description:

These functions give the obvious trigonometric functions. They respectively compute the cosine, sine, tangent, arc-cosine, arc-sine, arc-tangent, and the two-argument arc-tangent.

Usage:

```
cos(x)
sin(x)
tan(x)
acos(x)
asin(x)
atan(x)
atan2(y, x)
```

Arguments:

x, y: numeric vector

Details:

The arc-tangent of two arguments 'atan2(y,x)' returns the angle between the x-axis and the vector from the origin to (x,y), i.e., for positive arguments 'atan2(y,x) == atan(y/x)'.

Angles are in radians, not degrees (i.e., a right angle is pi/2).

All except 'atan2' are generic functions: methods can be defined for them individually or via the 'Math' group generic.

References:

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) _The New S Language_. Wadsworth & Brooks/Cole.

One, two, three, four....

Arithmetic operations in R can be combined using familiar *infix notation**; operators are written between operands (as in 2+2), as opposed to prefix (+ 2 2) or postfix (2 2 +)

Parentheses are used to clarify precedence and operators are assigned orders as well (for example, exponentiation is evaluated before multiplication, which comes before addition)

```
> ((12^2)/3)+3
[1] 51

> 12^2/3+3
[1] 51
```

* The notation is certainly familiar, the description may not be!

Catching the result

So far, we have answered the siren call of the R prompt with simple expressions that are evaluated and the result is simply printed to the screen; typically, we want to store the output of an expression

```
> x <- 5+7

> y <- sqrt(10)

> z <- exp(y) + x^2
```

In each case, we have calculated some *value* and *assigned* it to a *variable*

Catching the result

Variable names can contain letters, digits, “.” and “_”; although they cannot begin with a digit or “_”, and if they start with a “.”, the second character can’t be a digit

It is possible, of course, to get around these restrictions if you have some burning need to (and for most of you, this need will never materialize); enclosing your name in **backtick quotes** will tell the R evaluator (the program catching and processing your expressions) that the string is a (variable) name

```
> `2008vote` <- 100+3

> `more a sentence than, say, a variable` <- sqrt(200)
```

Creating variables

Variables in R have both a name and a value; when we refer to variables by name, R simply prints their value

```
> x <- 5+7
> x
[1] 12

> y <- sqrt(10)
> y
[1] 3.162278

> z <- exp(y) + x^2
> z
[1] 167.6243
```

When we refer to these variables in later expressions, R will substitute their value

Creating variables

It is likely that many of you have had a bush with either R or something like R and none of this is earth-shattering; you can make assignments and there are some rules about what you can name things

Note, however, that R uses “copying” semantics (you might also read that R is a “pass by value” language); this means when we assign the value of one variable to another, it is not “linked” to the original variable

```
> x <- 5
> y <- x
> x <- 10

> x
[1] 10

> y
[1] 5
```

Creating variables

In addition to “<-” there are several kinds of assignment operators

```
> y = sqrt(10)

> x <- 5*7

> 5*7 -> x

> x <<- 5*7  # we'll get to this later
```

According to Chambers, the first three are synonymous; the only thing that distinguishes them is the “grammar” associated with their application (for clarity, many prefer the consistent use of “<-”)

Historically, the arrows came first (bi-directional assignments were useful in an age before command line editing) and the “=” followed in later versions of the language

Objects

The variables we have created (the names we have assigned) all refer to *objects*; as we will see, **everything in R is an object** (sound familiar?)

While working in R, the system accumulates all of your objects in your *workspace*; you can view the contents of your workspace with the function called `objects()` or, out of deference to its UNIX roots, `ls()`

```
> x <- 5+7

> y <- sqrt(10)

> z <- exp(y) + x^2 ; w <- 100     # semi-colons
                                   # separate commands
> objects()
[1] "w" "x" "y" "z"
```

Removing objects

We can remove objects with the functions `rm()` or `remove()`

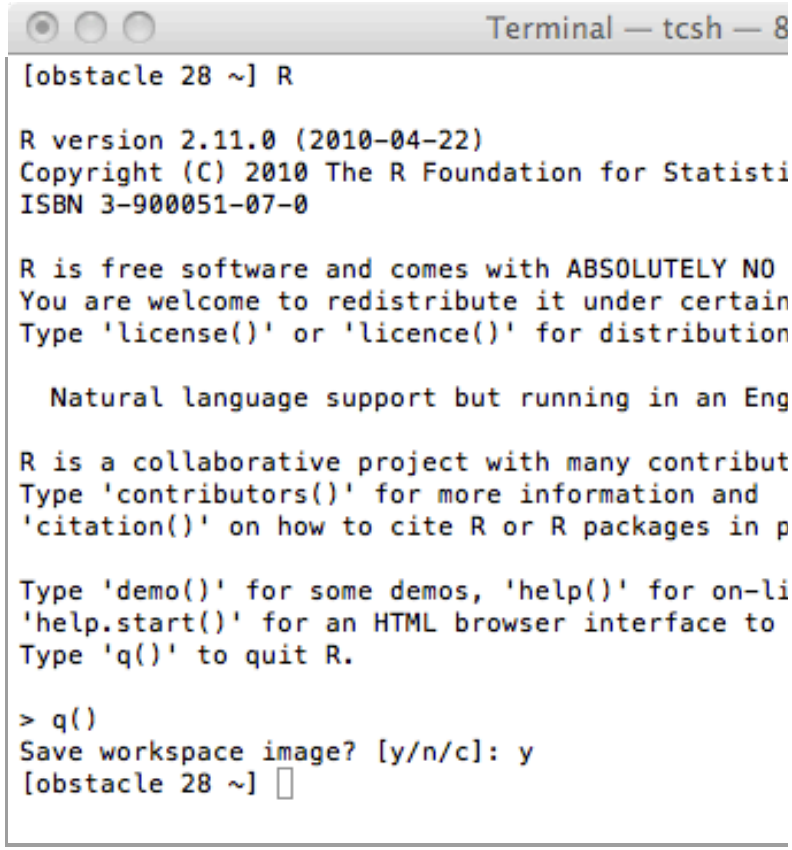
```
> objects()
[1] “w” “x” “y” “z”

> remove(”x”)
> objects()
[1] “w” “y” “z”

> rm(”y”,”z”)
> objects()
[1] “w”
```

A hasty retreat...

Before we know enough to be dangerous, let's exit out of R; or rather, quit R

This is done with the command `q()`; notice that like `help()` and `cos()`, `q()` is another function

You will be asked whether you want to save your workspace; for now answer "`y`"; you will then be returned to the UNIX prompt

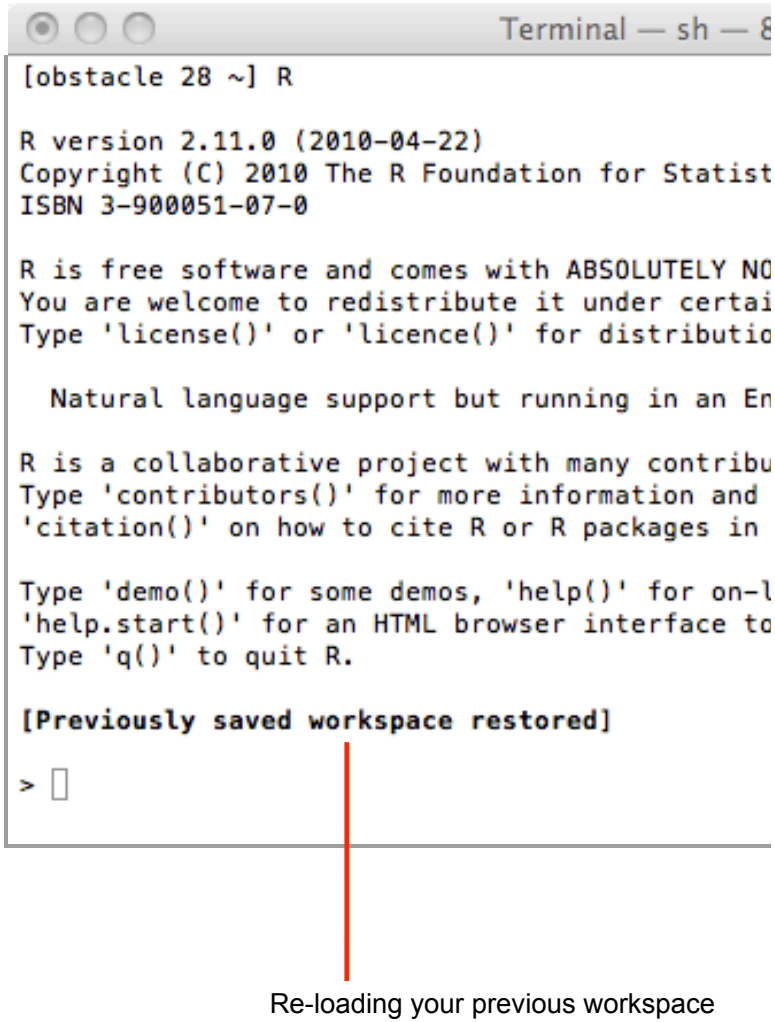


The R session

And now, let's get back into R; what do you notice?

Each time we start R, we create a new R *session;* during a session we will run a series of commands, and create a number of objects

We might read in data, fit a statistical model, view some graphical diagnostics of the fit, maybe run simulations, define functions, the works



Your workspace

As mentioned previously, R stores objects in your *workspace*

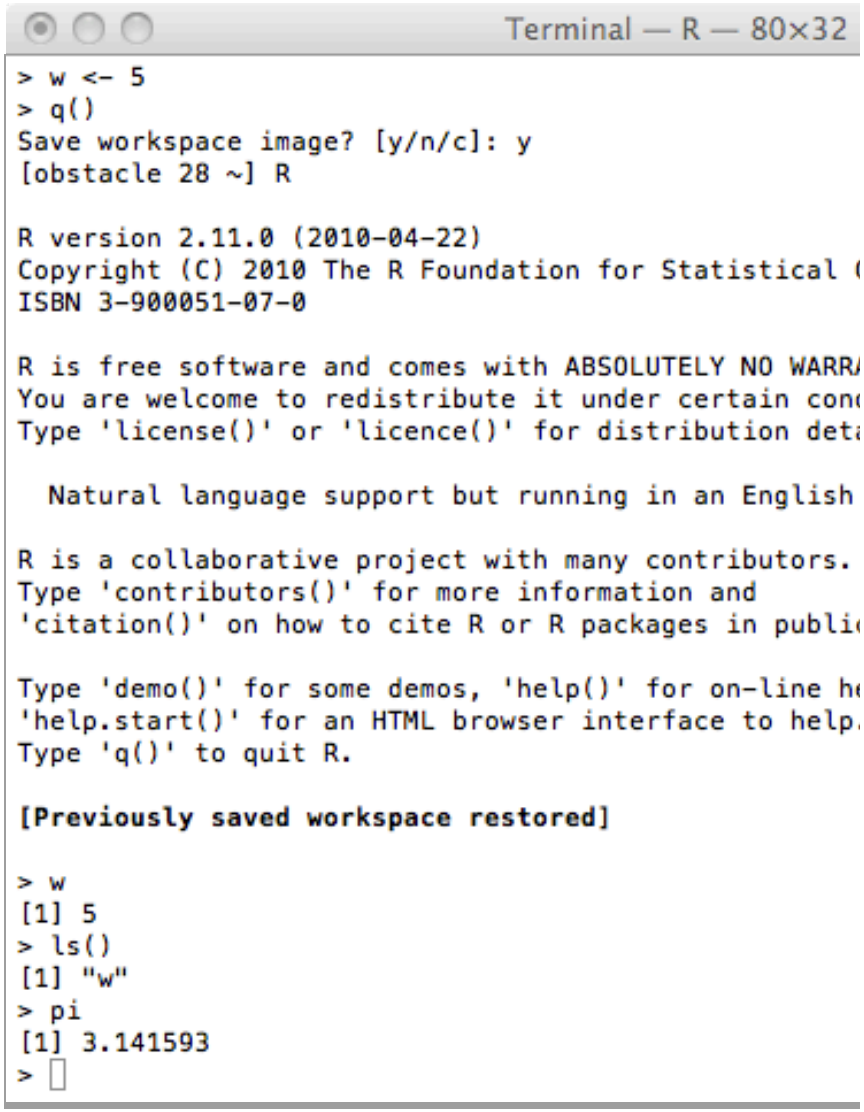
When you end your R session, you are asked if you want to save the objects you created; if you do, they are saved in a file called `.RData` that is **stored in the directory where you started R** (typed the command `R`)

When you start and R session from a directory where you have previously saved your workspace, it will restore it for you -- You can (and are encouraged to) have different directories devoted to different projects; each will have its own `.RData`

A bit more about your workspace

Technically, your workspace is only one of several locations where R can find data and functions, or, as we heard last time *objects*

Where did R find `pi`? (For that matter, where has it been finding `cos()` and `help()`?)



A bit more about your R workspace

Your workspace heads a list of places where R looks when it needs to associate a name with a value; we can see this list with the command

```
> search()

 [1] ".GlobalEnv"        "package:stats"  "package:graphics"
 [4] "package:grDevices" "package:utils"  "package:datasets"
 [7] "package:methods"   "Autoloads"      "package:base"
```

Technically, **your workspace is an "environment"** in R (**an environment is** just a collection of bindings, **a mapping between names and values**); when we end our session this workspace disappears (again, we will explain the "`[1]`" and "`[4]`" shortly)

When we ask R for an object, it runs through this list in order and inquires whether each one has the relevant variable

We refer to this list as your *search path*

A bit more about your R workspace

We can find the location of an object as follows

```
> find(pi)
[1] "package:base"
```

```
> pi = 1
> find(pi)
[1] ".GlobalEnv"   "package:base"
```

When we make an assignment, we've told R to ignore the version in the other package

A bit more about your R workspace

If you are curious, you can see what R is holding in these different workspaces

```
> objects("package:base")
```

```
  [1] "AIC"                   "ARMAacf"               "ARMAtoMA"
  [4] "Box.test"              "C"                     "D"
  [7] "Gamma"                 "HoltWinters"           "IQR"
 [10] "KalmanForecast"        "KalmanLike"            "KalmanRun"
 [13] "KalmanSmooth"          "NLSstAsymptotic"       "NLSstClosestX"
 [16] "NLSstLfAsymptote"      "NLSstRtAsymptote"      "PP.test"
 [19] "SSD"                   "SSasymp"               "SSasympOff"
 [22] "SSasympOrig"           "SSbiexp"               "SSfol"
 [25] "SSfpl"                 "SSgompertz"            "SSlogis"
 [28] "SSmicmen"              "SSweibull"             "StructTS"
 [31] "TukeyHSD"              "TukeyHSD.aov"          "acf"
 [34] "acf2AR"                "add.scope"             "add1"
 [37] "addmargins"            "aggregate"             "aggregate.data.frame"
  .                            .                       .
  .                            .                       .
  .                            .                       .
```

Finding things

We saw commands like `which` and `where` that search through a similar kind of path in Unix, looking for executable files (applications)

Naming and resolving names is an important activity and one that we have talked about in general -- Many of these ideas are common to R, Unix, Python...

Sidenote: Transcripts of your sessions

In each directory where you've run R, you will also find a file called `.RHistory`; R uses this file to maintain a list of your previous commands

Commands are accumulated across sessions, so you can recover information on possibly dimly remembered analyses; You can retrieve and edit previously typed commands (from the current and any previously saved sessions) with the up- and down-arrow keys

In general, R provides a series of functions that help you collect, tag, search and manage your history; in part, this is because the designers of how the language see you moving from casual analysis (tool user) to programmer (tool maker)

```
> help("history")
```

savehistory package:utils R Documentation

Load or Save or Display the Commands History

Description:

Load or save or display the commands history.

Usage:

```
loadhistory(file = ".Rhistory")
savehistory(file = ".Rhistory")

history(max.show = 25, reverse = FALSE, pattern, ...)

timestamp(stamp = date(),
          prefix = "##------ ", suffix = " ------##",
          quiet = FALSE)
```

Arguments:

file	The name of the file in which to save the history, or from which to load it. The path is relative to the current working directory.
max.show	The maximum number of lines to show. 'Inf' will give all of the currently available history.
reverse	logical. If true, the lines are shown in reverse order. Note: this is not useful when there are continuation lines.
pattern	A character string to be matched against the lines of the history
...	Arguments to be passed to 'grep' when doing the matching.
stamp	A value or vector of values to be written into the history.
prefix	A prefix to apply to each line.
suffix	A suffix to apply to each line.

Saving workspace

While on the topic of tracking history, you can save the contents of your workspace at any time with the function calls

```
> save(x,y,file="mywork.Rda")
> save.image()  # shorthand, save everything to .RData
```

When you quit `q()` out of R, `save.image` is called implicitly and you are prompted to save your work

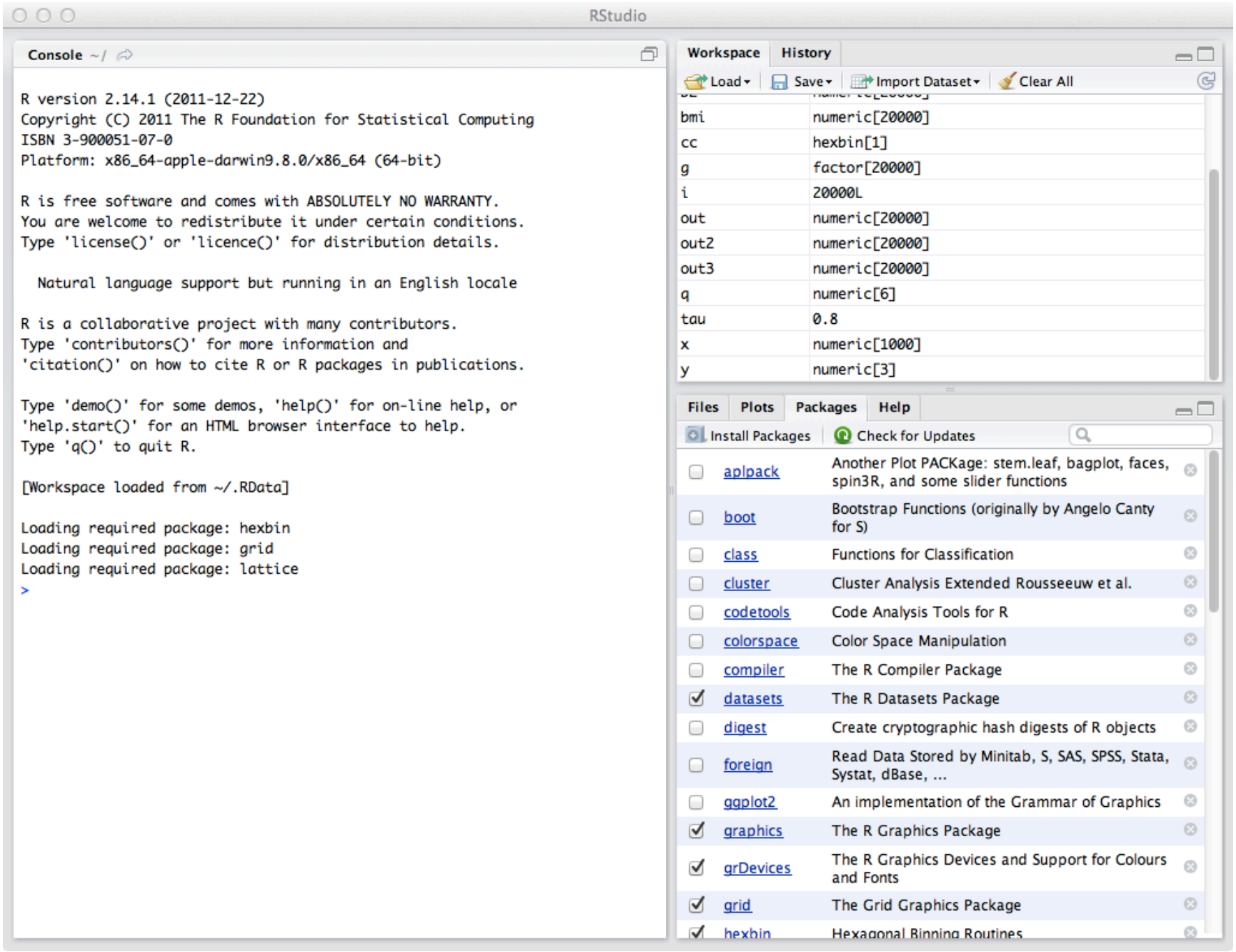
Caution: While R is running, any objects you create live only in your computer's memory; if the program dies for some reason, your work will be lost*

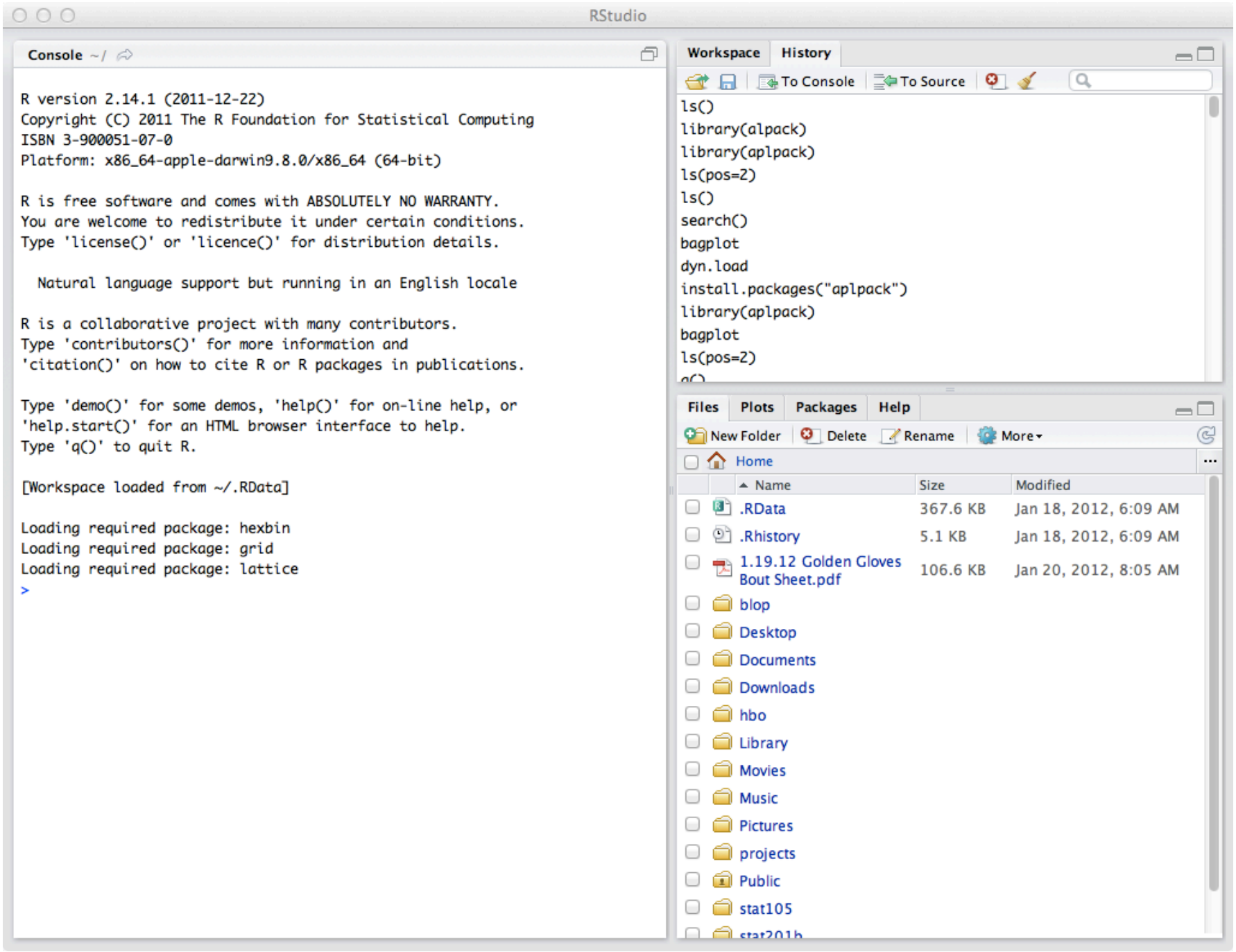
*Moral of the story: Saving your work periodically is not a bad idea!

RStudio

We've recommended that you download RStudio and do your work in it -- As an interface, you'll find many of the capabilities described here surfaced as buttons or panes in the GUI

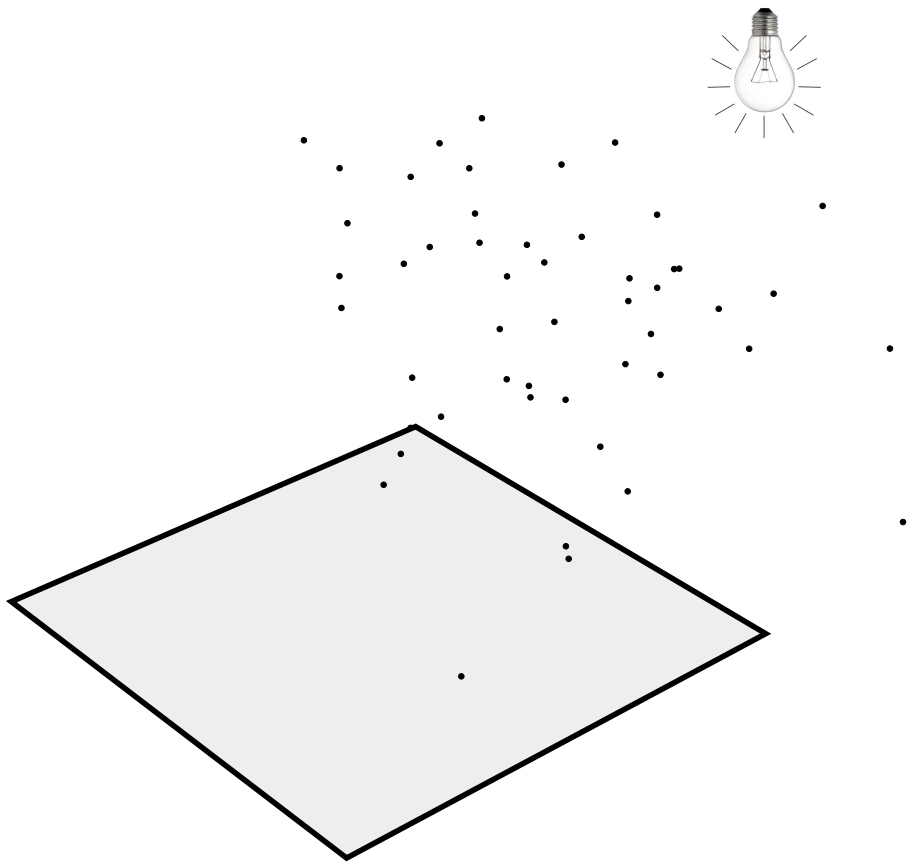
There are tabs to examine your history, to cycle through plots, to examine and install packages, together with a central pane with a running command line...


RStudio
Console ~/
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
Loading required package: hexbin
Loading required package: grid
Loading required package: lattice
>
Workspace
History
Load
Save
Import Dataset
Clear All
bmi numeric[20000]
cc hexbin[1]
g factor[20000]
i 20000L
out numeric[20000]
out2 numeric[20000]
out3 numeric[20000]
q numeric[6]
tau 0.8
x numeric[1000]
y numeric[3]
Files
Plots
Packages
Help
Install Packages
Check for Updates
aplpack Another Plot PACKage: stem.leaf, bagplot, faces, spin3R, and some slider functions
boot Bootstrap Functions (originally by Angelo Canty for S)
class Functions for Classification
cluster Cluster Analysis Extended Rousseeuw et al.
codetools Code Analysis Tools for R
colorspace Color Space Manipulation
compiler The R Compiler Package
datasets The R Datasets Package
digest Create cryptographic hash digests of R objects
foreign Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
ggplot2 An implementation of the Grammar of Graphics
graphics The R Graphics Package
grDevices The R Graphics Devices and Support for Colours and Fonts
grid The Grid Graphics Package
hexbin Hexagonal Binning Routines


RStudio
Console ~/
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-apple-darwin9.8.0/x86_64 (64-bit)
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
[Workspace loaded from ~/.RData]
Loading required package: hexbin
Loading required package: grid
Loading required package: lattice
>
Workspace
History
To Console
To Source
ls()
library(alpack)
library(aplpack)
ls(pos=2)
ls()
search()
bagplot
dyn.load
install.packages("aplpack")
library(aplpack)
bagplot
ls(pos=2)
Files
Plots
Packages
Help
New Folder
Delete
Rename
More
Home
Name
Size
Modified
.RData
367.6 KB
Jan 18, 2012, 6:09 AM
.Rhistory
5.1 KB
Jan 18, 2012, 6:09 AM
1.19.12 Golden Gloves Bout Sheet.pdf
106.6 KB
Jan 20, 2012, 8:05 AM
blop
Desktop
Documents
Downloads
hbo
Library
Movies
Music
Pictures
projects
Public
stat105

From last time

We examined two-dimensional projections of a data set that are not “axis-aligned” as in the scatterplot matrix -- We can considered casting high-dimensional shadows of the data when viewed from different angles



GGobi data visualization sys

www.ggobi.org

Overview | Learn | Blog | Foundation | Packages | Publications | Download | Support

GGobi

Good pictures force the unexpected upon us

News: **Hack-at-it 2010**

Download GGobi for Windows, Mac and Linux

Introduction

GGobi is an open source visualization program for exploring high-dimensional data. It provides highly dynamic and interactive graphics such as tours, as well as familiar graphics such as the scatterplot, barchart and parallel coordinates plots. Plots are interactive and linked with brushing and identification.

GGobi is fully documented in the GGobi book: "Interactive and Dynamic Graphics for Data Analysis".

If you are interested in how GGobi came to be, you can read more about it on our history page.

Features

- Need to look up cases with low or high values on some variables (price, weight,...) and show how they behave in terms of other variables? → brush in linked plots.

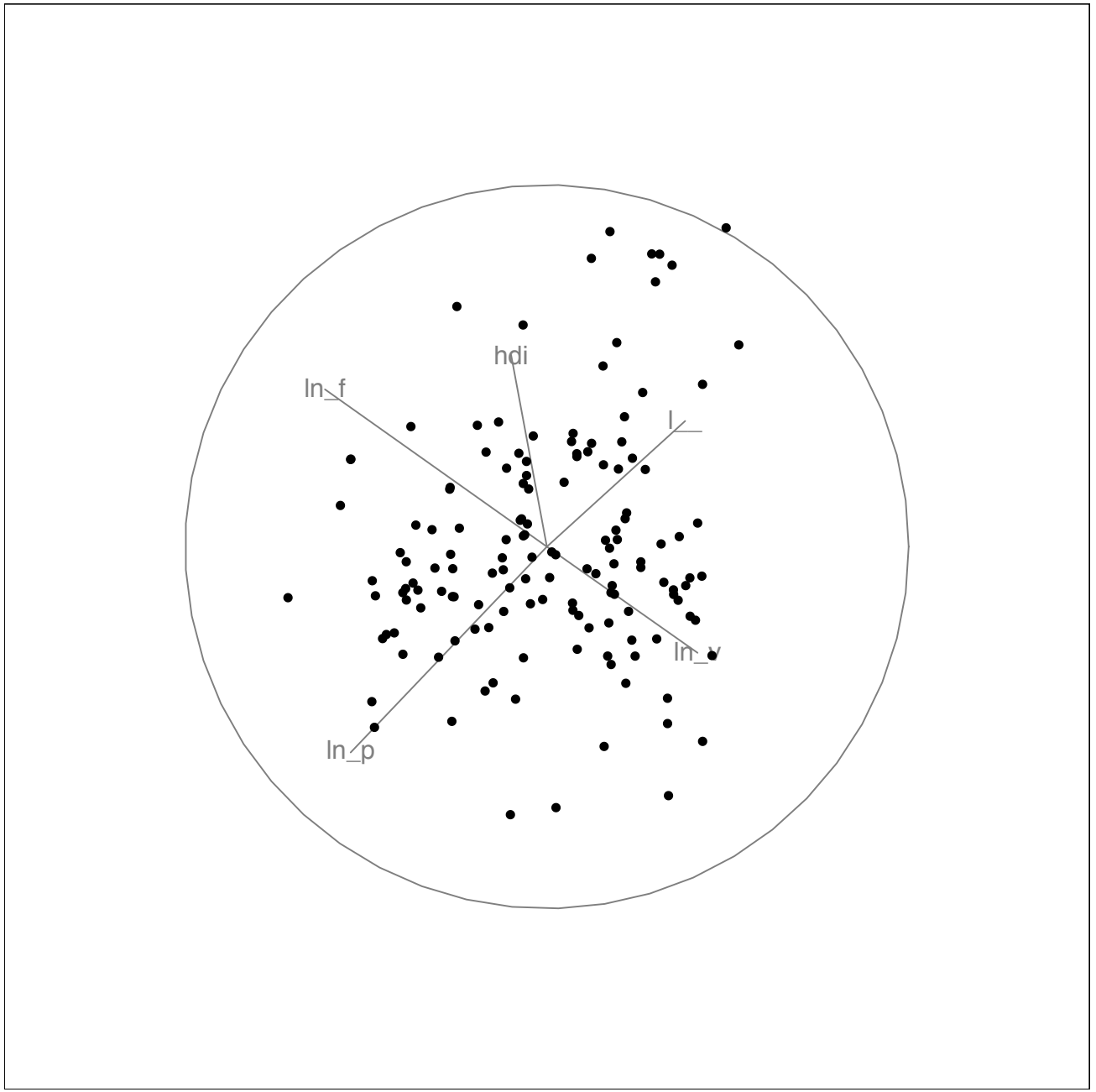
The R package `tourr`

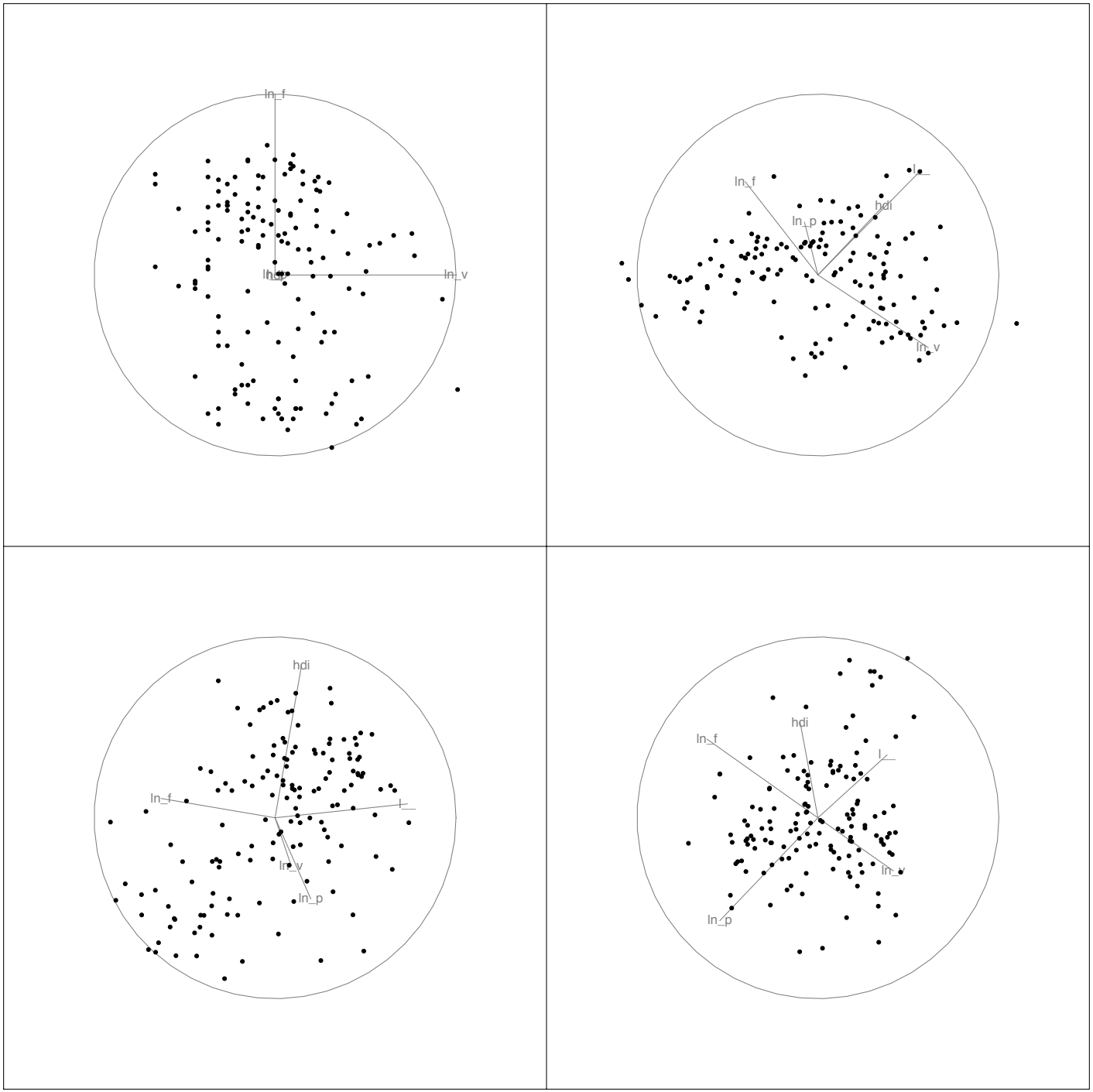
R uses a package mechanism to extend its functionality -- For example, we can examine the grand tour (and a host of other tours!) with the package tourr

```
> install.packages("tourr")
> library(tourr)
```

These lines (or their equivalent in the “packages” RStudio pane) will reach out and install the code for `tourr`, something you only have to do once -- Then, whenever you want to take a tour in R, you invoke the `library()` command

On the next page we show some stills from the R-based tour...


hdi
ln_f
l
ln_y
ln_p


ln_f
ln_v
hdi
ln_f
l_
hdi
ln_p
ln_v
hdi
ln_f
l_
ln_v
ln_p
hdi
ln_f
l_
ln_v
ln_p

The evolution of data in S and R

As we've seen, S has gone through a series of changes as it evolved from its first form in the 70s; during this process, data structures and the organization of computations on these structures changed

The earliest and most basic data structure in R is a vector and the function `is.vector()` will tell us if an object is a vector -- This structure can hold only one kind of data and `typeof()` will tell us which type it is

Data structures in R

Over the next few lectures, we'll examine the basic data types in R -- We'll start today, however with vectors and illustrate how even these simple constructions allow us to do quite a lot with data easily

vectors: ordered collections of primitive elements

matrices and arrays: rectangular collections of primitive elements having dimension 2,...

lists: a kind of generic vector, where elements can be of mixed (not-necessarily primitive) type

data frames: two-dimensional data tables

factors: categorical variables

Vectors

Vectors are a primitive type of object in R; a vector is just a collection of values grouped in a single container

The basic types of values are

```
numeric: real numbers
complex: complex numbers
integer: integers
character: character strings
logical: elements are TRUE or FALSE
raw: elements are raw bytes
```

Vectors can also contain the `NA` symbol to indicate a missing value

Aside from this special symbol, **vectors hold data of the same type** -- We refer to this type as the *mode* of the vector

Working with vectors

When displaying a vector, R lists the elements from left to right, using multiple rows depending on the “width” of your display; each new row includes the index of the value starting that row

This explains why we saw the “`[1]`” in front of each variable’s value we printed in our previous session; R is interpreting single values as vectors of length 1

Creating vectors

The simplest way to create a vector is to simply concatenate a series of primitive elements using the function `c(...)`

```
> x <- c(1,2,3)  # numeric elements; when you type
                 # values, they are stored as numeric

> x <- c("brown","whitman")  # characters

> x <- c(TRUE,FALSE,TRUE,TRUE) # logical
```

Creating vectors

Integer vectors are often used to create indices into other vectors (or more complex things like matrices); R has a number of simple functions that return integer vectors

```
> x <- 1:100      # integer elements
> class(x)
[1] "integer"

> x <- rep(1,5)
> x
[1] 1 1 1 1 1

> x <- 4:6
> y <- rep(x,1:3)
> y
[1] 4 5 5 6 6 6
```

Creating vectors

In addition to `rep`, the function `seq` is also frequently used for creating vectors (this time of class `numeric`)

```
> x <- seq(0,3,len=100)  # creates a sequence of 100
                         # evenly spaced values
                         # between 0 and 3

> x <- seq(0,3,by=1)     # another option
> class(x)
[1] “numeric”

> x
[1] 0 1 2 3
```

Creating vectors

You can also add names to the elements of a vector either when you create it

```
> x <- c(”a” = 1, “b” = 2, “c” = 3)  # adding names

> names(x)
[1] "a" "b" "c"
```

Or by using an assignment with the function `names()`

```
> x <- 1:26
> names(x) = letters     # remember this syntax!
```

Creating vectors

Note that the default printing method for a vector changes when you assign it names

```
> x <- 1:26
> x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
[16] 16 17 18 19 20 21 22 23 24 25 26

> names(x) <- letters
> x
 a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
 q  r  s  t  u  v  w  x  y  z
17 18 19 20 21 22 23 24 25 26
```

Vectorized operations

Since the basic object in R is a vector, many operations in the language **work on entire vectors**

For example, each variable below is a vector with length 100

```
> x <- runif(100)
> z <- sqrt(x)
> y <- x + 2*z + rep(c(1,2),50)
```

The recycle rule

What happens if we add two vectors of different lengths?

R replicates or recycles the shorter vector to have the same length as the longer one

```
> c(1,2,3) + 3
[1] 4 5 6

> c(100,200,300,400)+c(1,10)
[1] 101 210 301 410

> c(100,200,300,400)+c(1,2,3)
[1] 101 202 303 401

Warning message:
longer object length
        is not a multiple of shorter object length in:
c(100, 200, 300, 400) + c(1, 2, 3)
```

Coercion

Since a vector can only hold data of one kind, R will *coerce* elements to a common type

```
> x <- c(1:3,”a”)
> x
[1] "1" "2" "3" "a"

> class(x)
[1] "character"
```

This might also happen when you try to perform an arithmetic operation

```
> x <- c(TRUE,FALSE,TRUE,TRUE)
> x + 3
[1] 4 3 4 4
```

Working with vectors

R lets you perform simple operations like sorting and finding just the unique values of a vector

```
> sort(somenums)
[1] 0.1256883 0.1337630 0.3177054 0.3380988
[5] 0.4516134 0.5265388 0.5496565 0.6209405
[9] 0.8841781 0.9857302

> unique(c(TRUE,TRUE,FALSE,FALSE))
[1]  TRUE FALSE

> x <- c(1,3,1,1,4,5)
> duplicated(x)
[1] FALSE FALSE  TRUE  TRUE FALSE FALSE
```

Subsetting

R has 5 rules for selecting certain elements from a vector or a matrix -- They all use the operator “[”

- Indexing by position
- Indexing by exclusion
- Indexing by name
- Indexing with a logical mask
- Empty subsetting

Subsetting

Indexing vectors by position

```
> x <- letters
> x[1]
[1] "a"

> x[c(10,11)]
[1] "j" “k”

> x[c(10,11,43)]
[1] "j" “k” NA

> x[0]
character(0)

> x[c(0,10,11)]
[1] "j" “k”
```

Subsetting

Exclusion of elements in a vector by position

```
> letters[-c(10,11)]
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "l" "m"
[12] "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
[23] "y" "z"

> letters[c(-10,11,43)]
Error: only 0's may mix with negative subscripts
```

Subsetting

Indexing a vector by name

```
> x <- 1:3
> names(x) <- c(”a”,”b”,”c”)

> x[”a”]
a
1

> x[”a”,”b”]
Error in x["a", "b"] : incorrect number of dimensions


> x[-”a”]
Error in -"a" : Invalid argument to unary operator
```

Subsetting

Indexing a vector with **a logical mask**

```
> x <- 1:5
> x[c(TRUE,TRUE,FALSE,FALSE,FALSE)]
[1] 1 2

> x >= 3   # remember the recycle rule!
[1] FALSE FALSE TRUE TRUE TRUE

> x[x >= 3]
[1] 3 4 5

> x[x >= 2 & x < 4]  # combining w/ logical operators
[1] 2 3
```

Logical operators

These will appear both in subsetting as well as to control program flow (with loops, if-statements, and so on) -- they include operators like `<`, `>`, `<=`, `>=`, `==` and `!=`

On vectors, they apply elementwise (vectorized) and return a vector of logicals (`TRUE` or `FALSE` depending on whether the element satisfied the condition or not)

You can combine them with `&` for AND, and | for OR; the function `any(x)` returns `TRUE` if any element in `x` is true and `all(x)` returns `TRUE` if all the elements in `x` are true

Subsetting

And finally, empty subsetting

```
> x <- 1:5
> x[ ]
[1] 1 2 3 4 5
```

Subsetting

Uh, why the empty subsetting rule?

Notationally, this makes sense when you consider the subsetting operations for matrices which operate on rows and columns -- It also comes up when we talk about assignments for vectors...

Assignments for vectors

We can **modify specific elements in a vector** using our subsetting rules

```
> x <- 1:5
> x[c(2,4)] = 0
> x
[1] 1 0 3 0 5

> x[x < 2] <- 10
> x
[1] 10 10  3 10  5

> x[] <- 5
> x
[1] 5 5 5 5 5
```

Vectors and the Registrar

Let’s now have a look at how these ideas might help us with the registrar’s data --
Each column in the data set is a vector (or a factor)

The registrar

The registrar maintains **a record of every class you take**; in addition to what class, it publishes a catalog of when classes meet and how many people were enrolled

On the next page, we present a few lines from a data file we will eventually consider in lab; it is was provided by the registrar (at a cost of $85) and contains the schedules for every student on campus last quarter*

In all, we have 162380 separate rows in this table, each corresponding to a different student and a single class with 31981 total students; What can we learn from these data? And, more importantly, how?

*Note that the identification number in this table is not your student ID, or even part of it, but a random number generated to replace your real ID

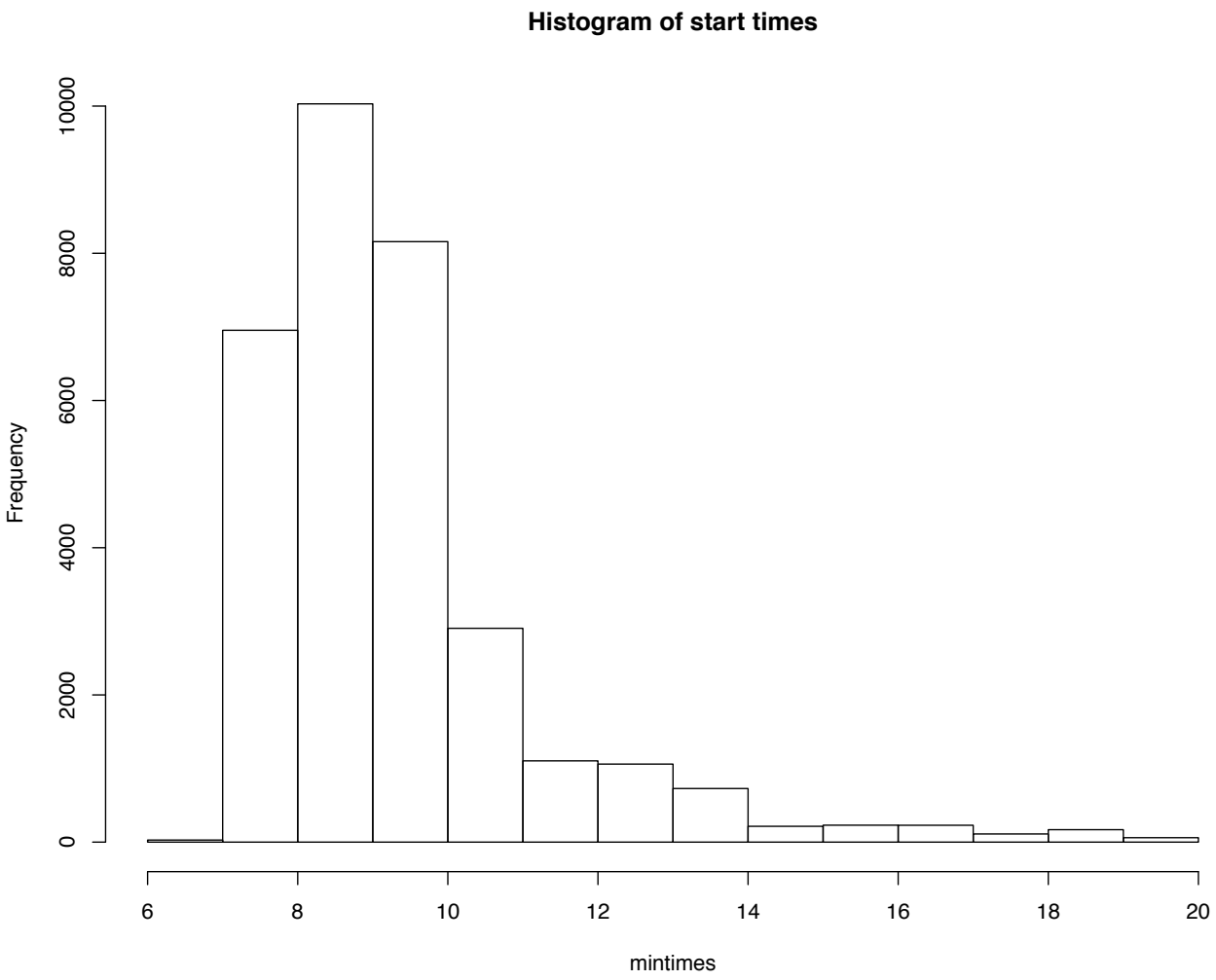
	id	subj_area_cd	cat_sort	MEET_BLDG_CD	meet_rm_sort	strt_time	end_time	DAYS_OF_WK_CD	career_cd
1	816640632	ANTHRO	0009	HAINES	00314	10:00:00	10:50:00	M	U
2	816640632	ANTHRO	0009	FOWLER	A00103B	11:00:00	12:15:00	TR	U
3	816640632	GEOG	0005	HAINES	00039	13:00:00	14:15:00	MW	U
4	816640632	ENGCOMP	0003	HUMANTS	A00046	09:30:00	10:45:00	TR	U
5	816640632	GEOG	0005	BUNCHE	A00170	11:00:00	12:50:00	M	U
6	816643648	MGMT	0403	GOLD	B00313	09:30:00	12:45:00	S	G
7	816643648	MGMT	0405	GOLD	B00313	14:00:00	17:15:00	S	G
8	816577472	COMM ST	0187	PUB AFF	01222	09:30:00	10:45:00	TR	U
9	816577472	COMM ST	0168	ROYCE	00362	17:00:00	19:50:00	M	U
10	816577472	COMM ST	0133	DODD	00175	10:00:00	10:50:00	MWF	U
12	806029941	EDUC	0491	KAUFMAN	00153	17:00:00	19:50:00	W	G
13	806029941	EDUC	0330D	FIELD		08:00:00	14:50:00	MTWRF	G
14	821748664	ANTHRO	0007	HAINES	00039	09:00:00	09:50:00	MWF	U
15	821748664	SPAN	0120	FOWLER	A00139	15:30:00	16:50:00	MW	U
16	821748664	SPAN	0120	HUMANTS	A00046	11:00:00	11:50:00	R	U
17	821748664	WOM STD	0107C M	HAINES	A00025	14:00:00	15:50:00	TR	U
18	821748664	ANTHRO	0007	HAINES	00350	12:00:00	12:50:00	R	U
19	820969784	ENGR	0180	BOELTER	02444	18:00:00	18:50:00	M	U
20	820969784	EL ENGR	0115AL	ENGR IV	18132	12:00:00	15:50:00	T	U
21	820969784	EL ENGR	0115A	ROLFE	01200	08:00:00	09:50:00	MW	U
22	820969784	EL ENGR	0115A	BOELTER	05280	09:00:00	09:50:00	F	U
23	820969784	STATS	0105	PAB	02434	15:00:00	15:50:00	R	U
24	820969784	STATS	0105	FRANZ	02258A	12:00:00	12:50:00	MWF	U
25	820969784	ENGR	0180	BOELTER	02444	16:00:00	17:50:00	MW	U
26	821030697	GEOG	0005	HAINES	00039	13:00:00	14:15:00	MW	U

```
# create a character vector of start times
> start <- as.character(reg$strt_time)
> head(start)
[1] "10:00:00" "11:00:00" "13:00:00" "09:30:00" "11:00:00" "09:30:00"

# pull out just the hours by looking at the first two characters
# note that this works on all the data -- vectorized operations!
> hours <- substr(start,1,2)
> head(hours)
[1] "10" "11" "13" "09" "11" "09"

# we don't want characters! let's make this a vector of integers
> hours <- as.integer(substr(start,1,2))
> head(hours)
[1] 10 11 13  9 11  9

# and do the same thing for minutes and seconds
> mins <- as.integer(substr(start,4,5))
> secs <- as.integer(substr(start,7,8))
> head(secs)
[1] 0 0 0 0 0 0

# hmm, not many classes have seconds specifying a start time!
> all(secs==0)
[1] TRUE
```

```
# now divide to turn these into fractional hours (past midnight)
> times <- (hours*60*60+mins*60+secs)/(60*60)
> head(times)
[1] 10.0 11.0 13.0  9.5 11.0  9.5

# create a new structure (a list) that holds all the data for
# a single student
> timesbyid <- split(times,reg$id)
> head(timesbyid)

$`805307175`
[1] 10  9

$`805310261`
[1] 13.5 17.5 13.5

$`805314438`
[1] 18.50000 19.16667

$`805315160`
[1]  8 12

$`805315434`
[1] 18

$`805355311`
[1] 12
```

```
# now, operate on this new list, “applying” a function to the data
# for each student -- here we look at the function min()

> mintimes <- sapply(timesbyid,min)

> is.vector(mintimes)
[1] TRUE

> length(mintimes)
[1] 31981

# the earliest someone starts?
> min(mintimes)
[1] 6

> sum(mintimes<=6)
[1] 13

> sum(mintimes<=7)
[1] 28

> sum(mintimes<=8)
[1] 6982
> hist(mintimes)
```

```
# now, operate on this new list, “applying” a function to the data
# for each student -- here we look at the function min()

> mintimes <- sapply(timesbyid,min)

> is.vector(mintimes)
[1] TRUE

> length(mintimes)
[1] 31981

# the earliest someone starts?
> min(mintimes)
[1] 6

> sum(mintimes<=6)
[1] 13

> sum(mintimes<=7)
[1] 28

> sum(mintimes<=8)
[1] 6982

> hist(mintimes)
```


Histogram of start times
Frequency
0
2000
4000
6000
8000
10000
6
8
10
12
14
16
18
20
mintimes

```
# now find those early risers!
> mintimes[mintimes==6]
815923238 815973022 816640891 816829288 816901547 817099145 817099697 817675913
        6         6         6         6         6         6         6         6
817679513 817684838 819143241 819929173 820773548
        6         6         6         6         6

# and have a look at the course schedule...
> subset(reg,id==815923238)

             id subj_area course building    room strt_time end_time
80425 815923238      CHEM  0014A       MS   05217  13:00:00 13:50:00
80426 815923238   MUS HST   0063      SMB   01200  11:00:00 12:50:00
80427 815923238   MIL SCI  0000Z      SAC  00120P  06:00:00 08:50:00
80428 815923238   MUS HST   0063    ROYCE   00164  10:00:00 10:50:00
80429 815923238     PSYCH   0010    FRANZ   01178  14:00:00 15:50:00
80430 815923238   MIL SCI   0012  KAUFMAN   00101  08:00:00 08:50:00
80431 815923238      CHEM  0014A  WGYOUNG CS00024  12:00:00 12:50:00
      days_of_week level
80425            W     U
80426           TR     U
80427            R     U
80428            F     U
80429           TR     U
80430            W     U
80431          MWF     U
```

Data frames

The registrar's data set is read into R as a data frame -- It's like R's equivalent of a spread sheet, having tabular structure

We're using the `subset()` function on the previous page, but as with Matlab, you can subset a data frame (or a matrix) using [-notation

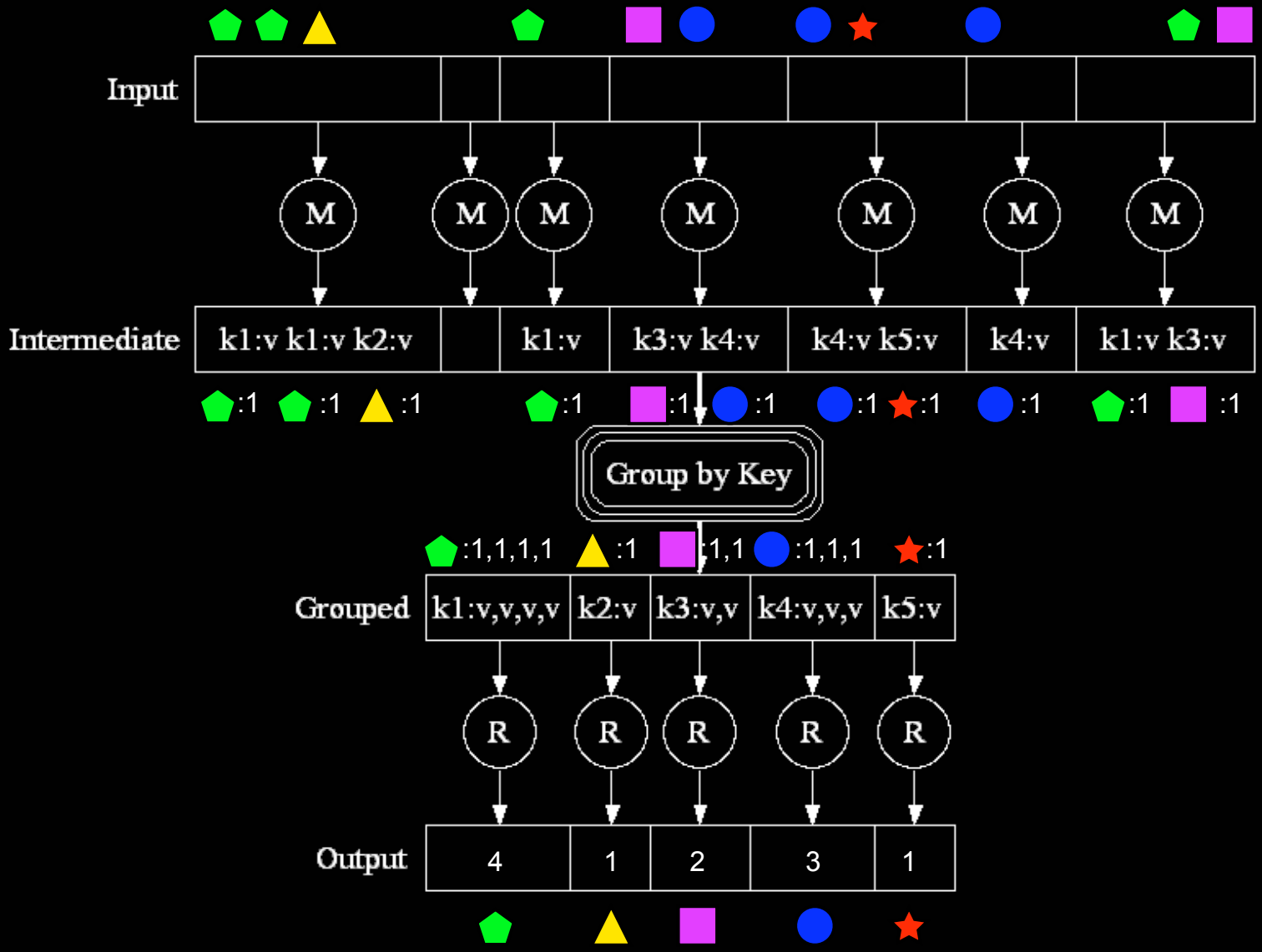
```
> reg[reg$id==815923238,]
```

All the same subsetting rules for vectors apply with expressions before the comma indexing rows and expressions after, columns

Map Reduce

The apply operations we performed on the previous page are very simple versions of a now-popular distributed computing paradigm known as Map Reduce -- The split and apply metaphor is (ignoring the distribution piece that is really the soul of the enterprise) essentially a map and reduce

On the next page we have a small cartoon illustrating Map Reduce -- We only really bring this up for context...


Input
M
M
M
M
M
M
M
Intermediate
k1:v k1:v k2:v
k1:v
k3:v k4:v
k4:v k5:v
k4:v
k1:v k3:v
:1 :1 :1 :1 :1 :1 :1 :1 :1 :1 :1
Group by Key
:1,1,1,1 :1 :1,1 :1,1,1 :1
Grouped
k1:v,v,v,v
k2:v
k3:v,v
k4:v,v,v
k5:v
R
R
R
R
R
Output
4
1
2
3
1

Inference

With the CDC BRFSS data we saw our first example of a survey -- **A random sample of the adult U.S. population** was asked a series of questions about their habits (um, health-related habits)

As we mentioned, we are often not just interested with the people who were surveyed, but instead what we might learn about the larger population -- **This is the process of statistical inference**

Today, we're going to look at one particular kind of study design and see how we might use the resulting data to draw conclusions

Types of studies

In the health and life sciences, we are faced with two kinds of studies that differ in terms the conditions under which data are collected

- In an **experimental study,** we impose some **change or treatment** and measure the result or response

- In an **observational study,** we simply **observe and measure something that has taken place or is taking place** (while trying not to cause any changes by our presence)

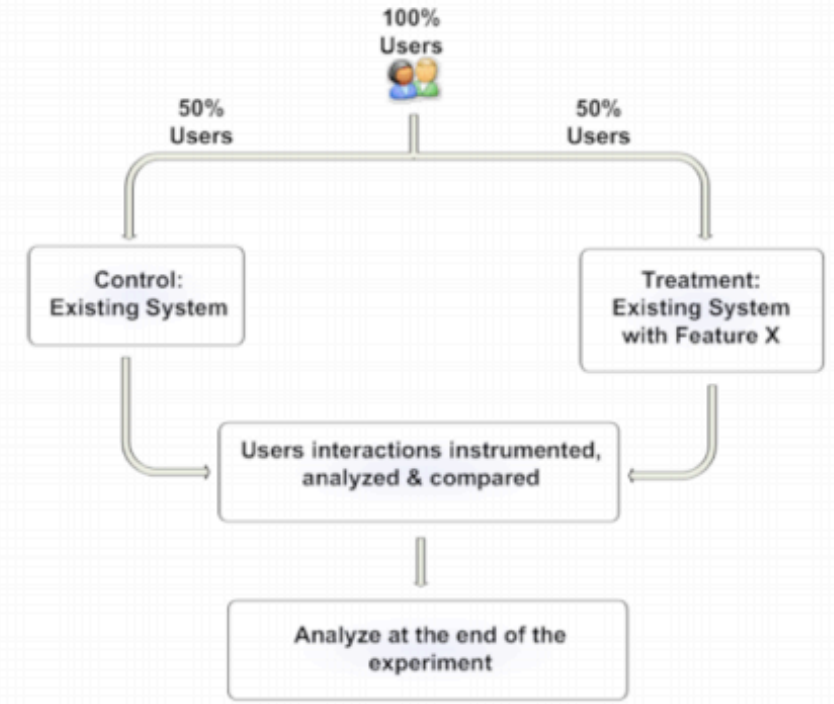
The kinds of inference you can make will depend on the type of study you conduct, **as well as its overall design or program for how data are to be collected** -- These kinds of considerations will lead to a range of (admittedly more technical) questions you should ask of a data set

In the last two lectures, we've talked about two data sets, the CDC Behavioral Risk Surveillance System and a list of courses from the registrar -- What kinds of "studies" do these represent

A/B testing

In an A/B test, visitors are presented with two different versions of the site (in some cases we literally have a treatment and control); sometimes the differences have to do with **the content on the page**, while in other cases they have to do with **the location of or visual arrangement of the content**

The outcomes that are measured depend on the site's overall objectives; an informational site might be interested in **the number of pages visitors read**, while commerce sites would be concerned with the **number of transactions made**

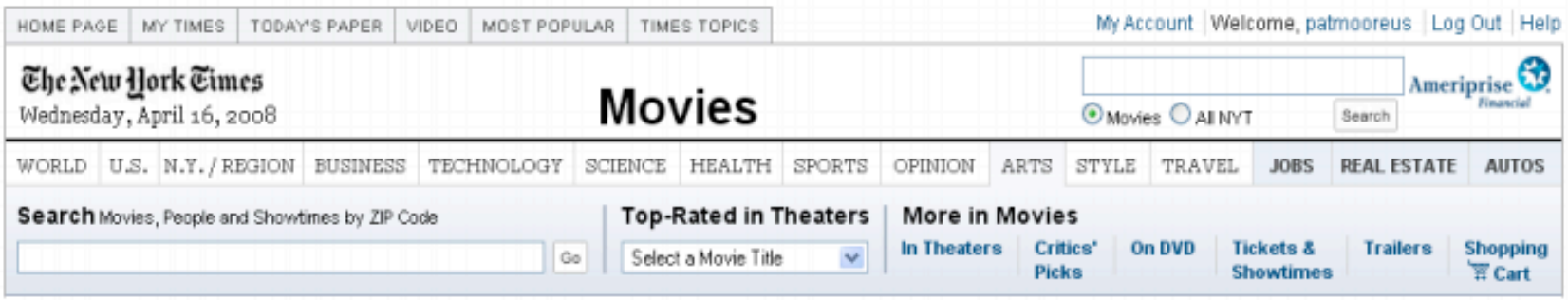


Kohavi, Henne and Sommerfield, *"Practical Guide to Controlled Experiments on the Web: Listen to Your Customers not to the HiPPO,"* KDD 2007

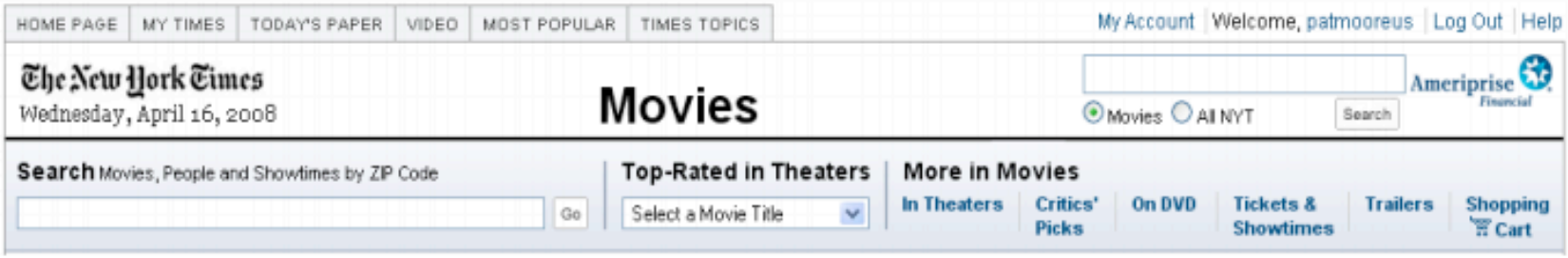
An example

Here is a simple experiment conducted by the New York Times web site (`nytimes.com`) in 2008 -- What is the difference between these two pages and what differences in visits might you be interested in comparing?

A: Control



B: Test

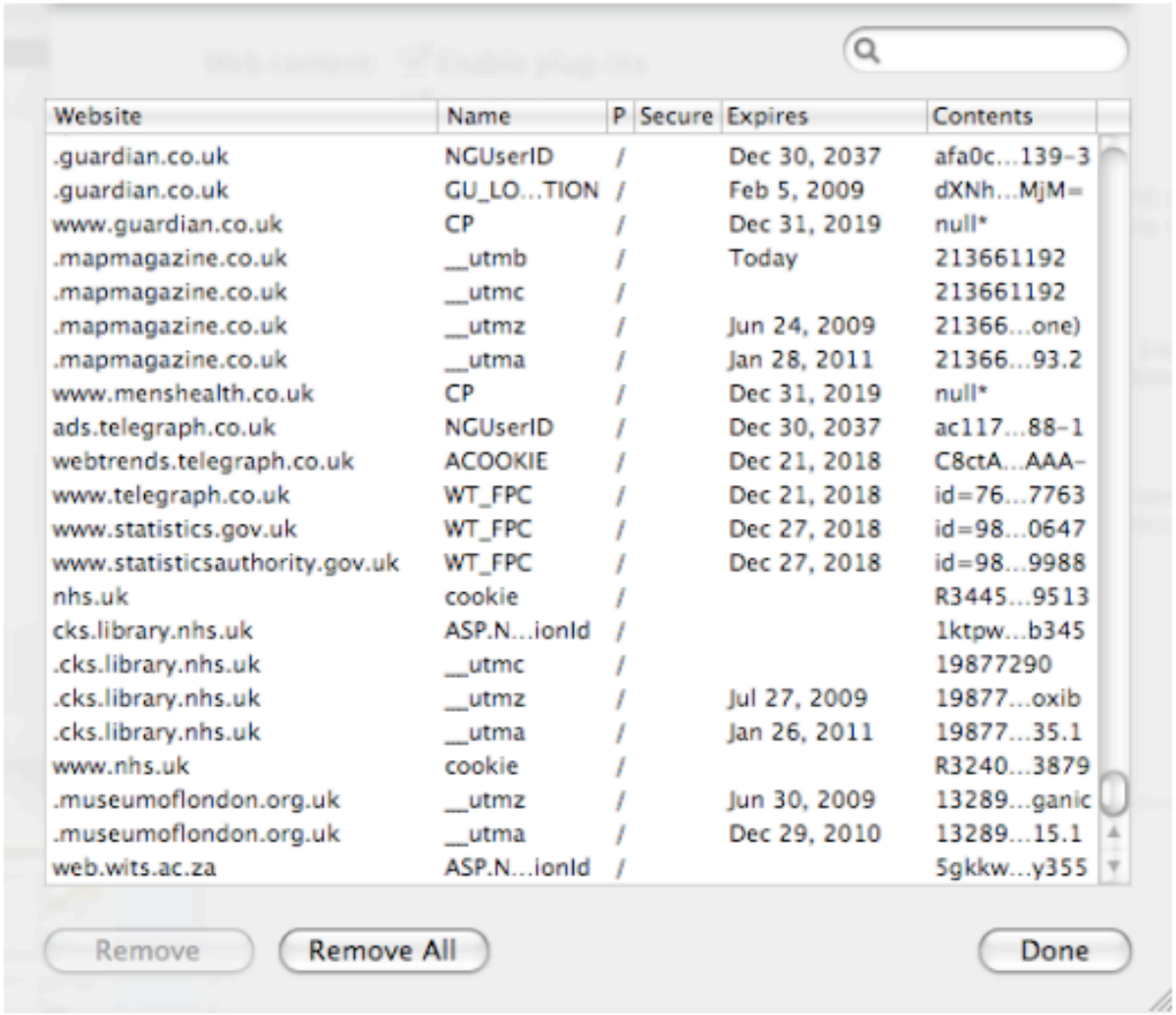


A/B testing

At a technical level (technical meaning from the standpoint of web servers), an A/B test means that visitors are assigned at random to group A or B (or C or D or E, depending on how many changes are being tested at one time)

Visitors assigned to group B, for example, will **see one version of the site consistently during the experimental period**; in the case of then Times example on the previous slide, it means that whenever a visitor in group B requests pages from The Movie Section, those pages are missing the middle navigation bar

What information does a web site need to know to make that happen? How is that information gathered?



Website	Name	P	Secure	Expires	Contents
.guardian.co.uk	NGUserID	/		Dec 30, 2037	afa0c…139-3
.guardian.co.uk	GU_LO…TION	/		Feb 5, 2009	dXNh…MjM=
www.guardian.co.uk	CP	/		Dec 31, 2019	null*
.mapmagazine.co.uk	__utmb	/		Today	213661192
.mapmagazine.co.uk	__utmc	/			213661192
.mapmagazine.co.uk	__utmz	/		Jun 24, 2009	21366…one)
.mapmagazine.co.uk	__utma	/		Jan 28, 2011	21366…93.2
www.menshealth.co.uk	CP	/		Dec 31, 2019	null*
ads.telegraph.co.uk	NGUserID	/		Dec 30, 2037	ac117…88-1
webtrends.telegraph.co.uk	ACOOKIE	/		Dec 21, 2018	C8ctA…AAA-
www.telegraph.co.uk	WT_FPC	/		Dec 21, 2018	id=76…7763
www.statistics.gov.uk	WT_FPC	/		Dec 27, 2018	id=98…0647
www.statisticsauthority.gov.uk	WT_FPC	/		Dec 27, 2018	id=98…9988
nhs.uk	cookie	/			R3445…9513
cks.library.nhs.uk	ASP.N…ionId	/			1ktpw…b345
.cks.library.nhs.uk	__utmc	/			19877290
.cks.library.nhs.uk	__utmz	/		Jul 27, 2009	19877…oxib
.cks.library.nhs.uk	__utma	/		Jan 26, 2011	19877…35.1
www.nhs.uk	cookie	/			R3240…3879
.museumoflondon.org.uk	__utmz	/		Jun 30, 2009	13289…ganic
.museumoflondon.org.uk	__utma	/		Dec 29, 2010	13289…15.1
web.wits.ac.za	ASP.N…ionId	/			5gkkw…y355

Remove | Remove All | Done

... the cookie jar

Cookies

Cookies can be set by the web sites you visit and are often used to "save state" during your session or between sessions -- This is how, for example, a site can implement a "shopping cart" or personalize a site based on your preferences or, as in this case, track your activities on the site

In this case, the cookie is set so that it can tell the site **which version of the site the visitor should see**

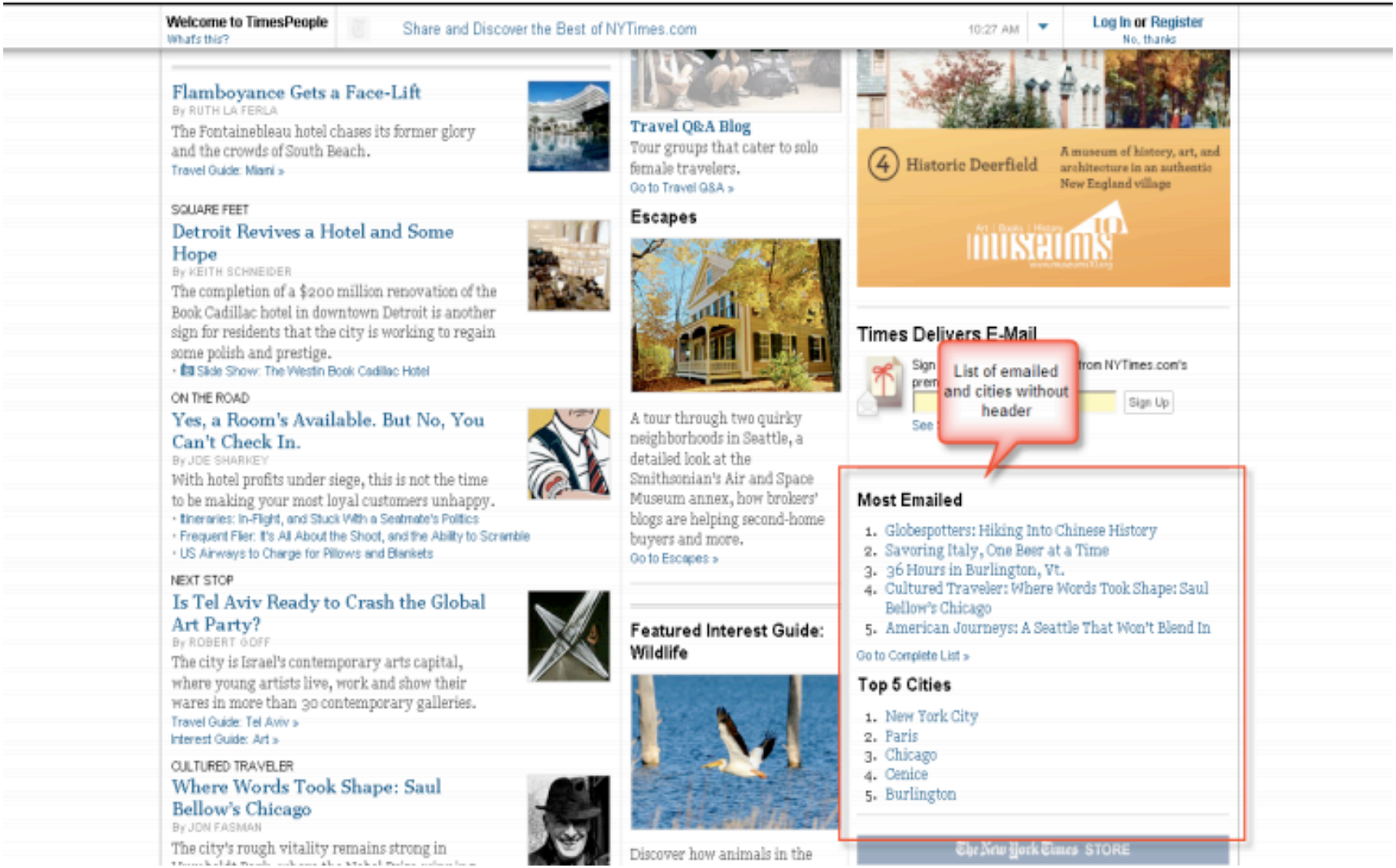
An experiment at `nytimes.com`

We will now consider a more recent example of an A/B test for **The Travel Section** of `nytimes.com` (we'll save the movie test for lab or your midterm or...)

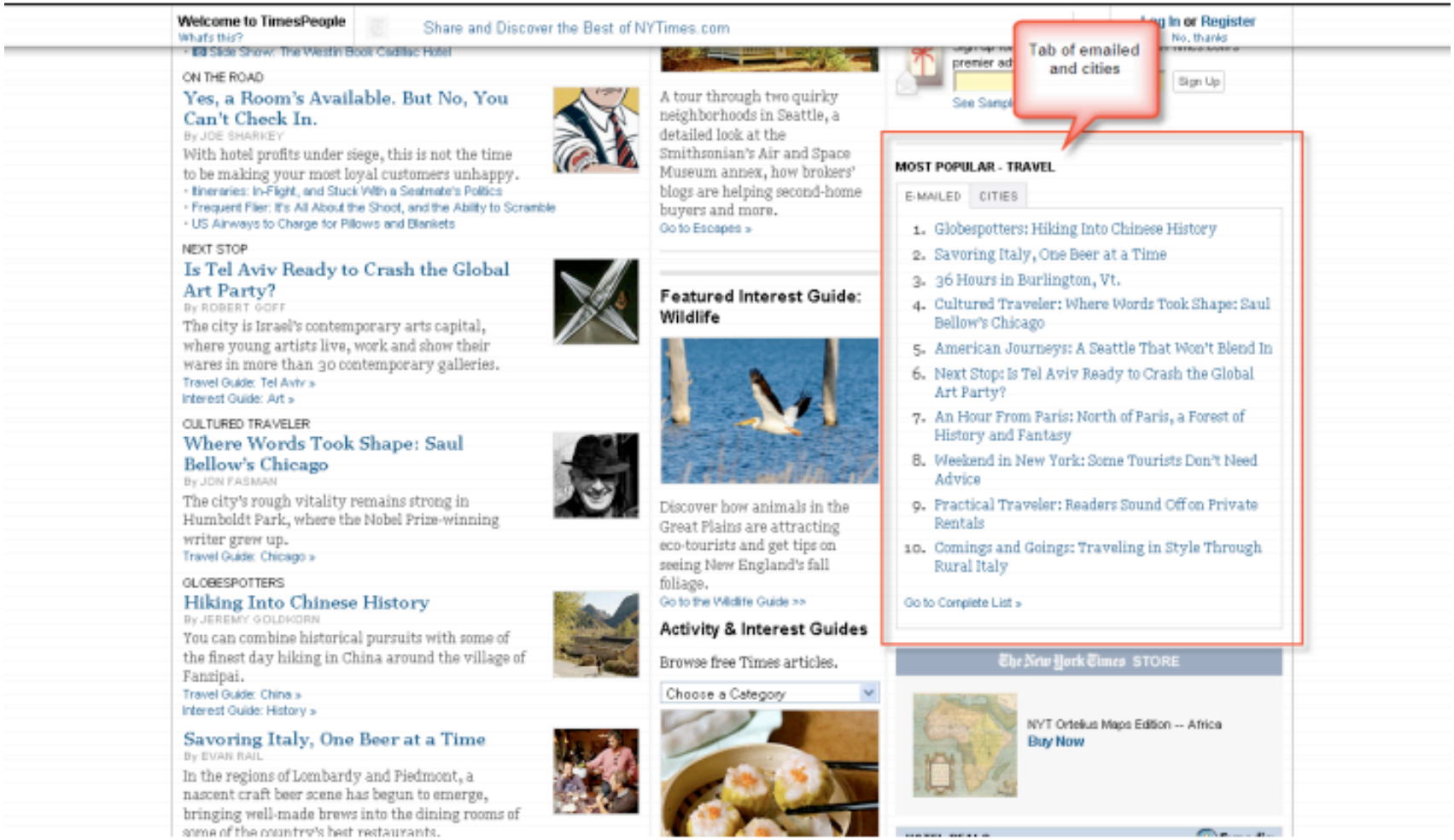
On the next two slides, we present samples of the A and B pages; the changes applied to all pages in The Travel Section, so **as a visitor browsed the site, they would consistently see either A or B**

Have a look at the two designs -- What differences do you see in terms of layout and content? What questions might the Times ask about how visitors react to these two options?

List: Variation 10858



Tabs: Variation 10859



An experiment at nytimes.com

The unit of observation during this study is a visit -- Imagine you are browsing the web and you land on `nytimes.com`, you browse a bit and then you leave to explore another site, maybe `wsj.com`

Your time at `nytimes.com` is referred to as a visit -- **The variables associated with a visit summarize your activities** and include when you arrived and how long you stayed

An experiment at `nytimes.com`

The Travel Section experiment lasted **five weeks**; the first visitor during the experiment arrived on Tuesday November 18 of 2008 at 8:40 am PST 2008, and the experiment closed with a final visit on Monday December 29 at about 8 pm PST

During that period, **nearly 200K visits were recorded as part of the experiment** (the experiment involved just a fraction of their overall traffic)

Given the nature of the change between A and B, and some of the objectives you wanted to examine, what sort of information would we like to collect about each visit? About each visitor?

The variables

Recall from last time the variables we have at our disposal; in all data were collected from about 130K users over a period of 6 weeks at the end of 2008

User_ID A unique number for each visitor

UserVisit_ID A unique number for each visit

StartTime_SSE Unix time for the start of the visit

StartTime_English A more human readable version of the time

VisitLength The number of seconds the visitor was reading Travel Section pages

Variation The version of the page they received

RefererUrl The page they clicked on (if any) to get to the page

EntryPageUrl The first page on `nytimes.com` they visited

Pageviews The number of page views to in the Travel Section

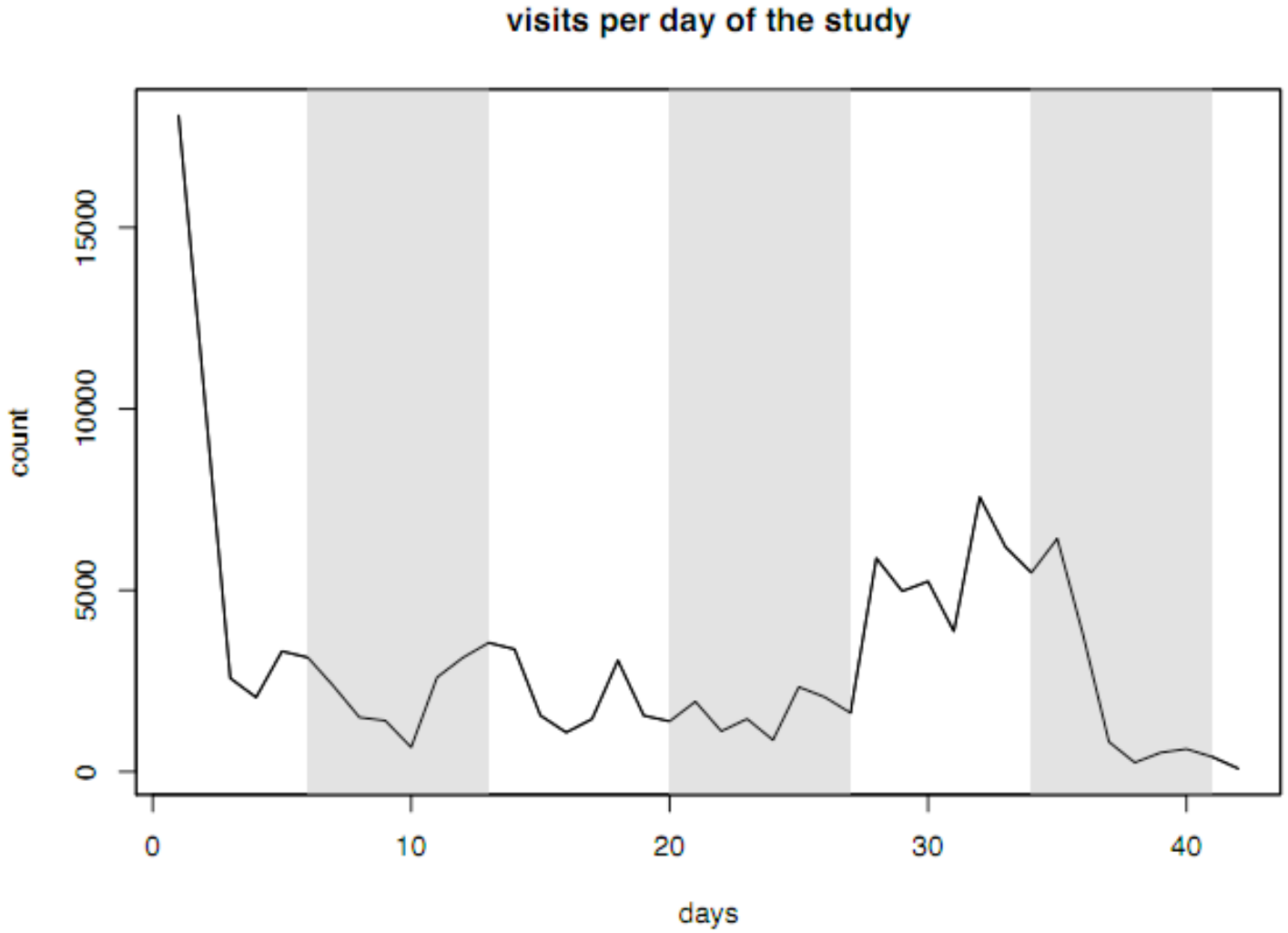
TotalVisits The total number of visits to the site

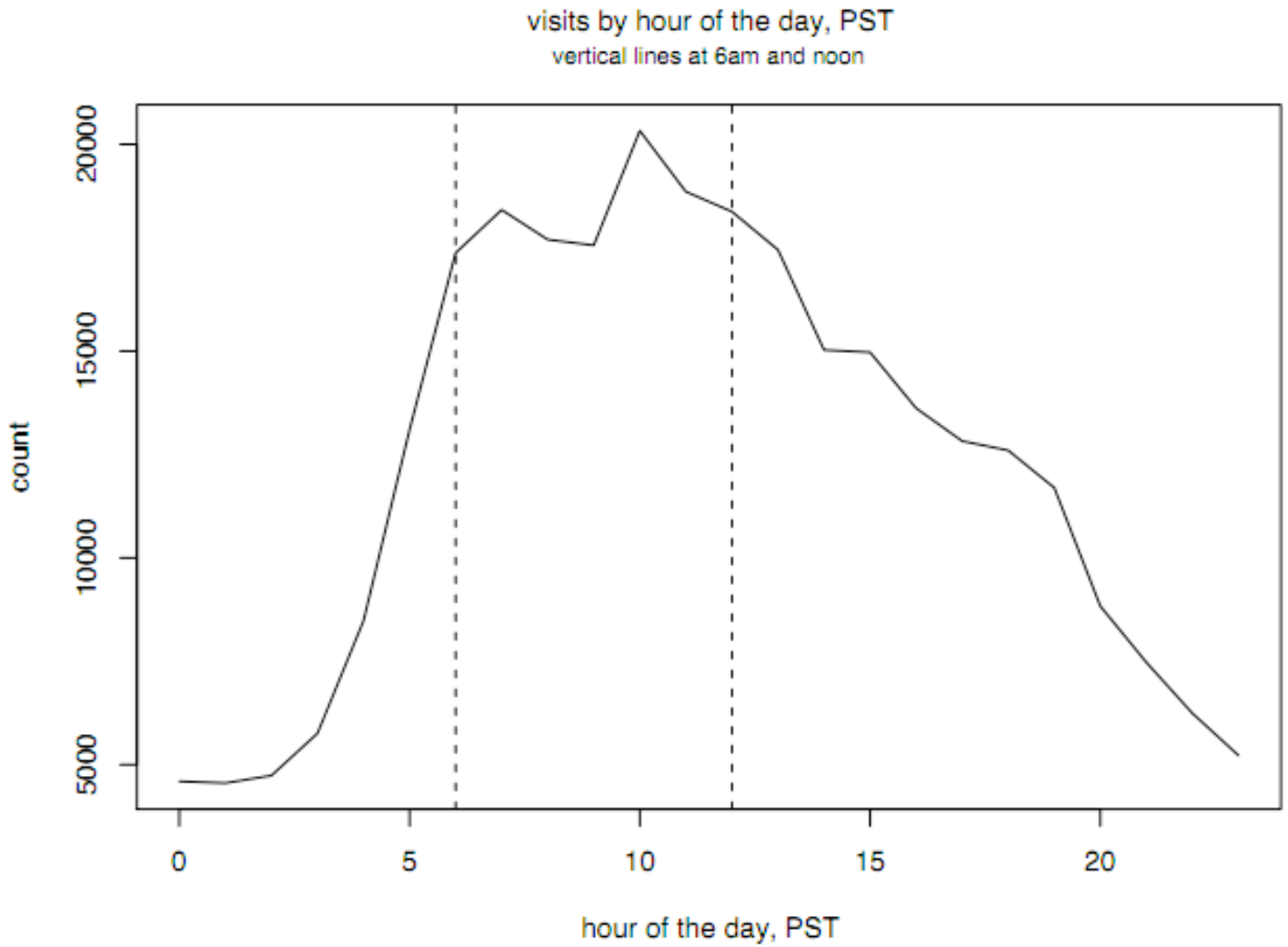
TimeSinceFirstVisit `(days)` How long it had been since their last visit

UserAgent Their browser

TotalClicks How many times did they click on the "most popular" field

IfClicked 0/1 did they click on the "most popular field" at least once


visits per day of the study
count
0
5000
10000
15000
0
10
20
30
40
days


visits by hour of the day, PST
vertical lines at 6am and noon
count
5000
10000
15000
20000
0
5
10
15
20
hour of the day, PST

Two entries from the data set

Here we have two visits, one of the first during the experiment (beginning at 8:40 am on November 18, PST) and one toward the end (starting at 6:15 am on December 23, PST)

How did they get to the site?

How long did they stay?

How many pages did they view?

Did they select anything from the “most popular” list?

* in case you were wondering... Google

User_ID 304431020
UserVisit_ID 374523135
StartTime_SSE 1227026436
StartTime_English Tue, Nov 18, 2008 - 16:40:36 (GMT)
VisitLength 23
Variation Tabs
RefererUrl http://www.google.com.jm/search?hl=en&q=clothes+in+paris
EntryPageUrl http://travel.nytimes.com/2006/05/21/travel/21forage.html
Pageviews 1
TotalVisits 1
TimeSinceFirstVisit (days) 40.99
UserAgent Mozilla/5.0 (Windows;U;Windows NT 5.0;en-US;rv:1.9.0.1) Firefox/3.0.1
TotalClicks 0
IfClicked 0

User_ID 248002503
UserVisit_ID 387427203
StartTime_SSE 1230041601
StartTime_English Tue, Dec 23, 2008 - 14:13:21 (GMT)
VisitLength 1297
Variation List
RefererUrl http://www.google.ro/search?hl=ro&q=Milan+Central+Station+grocery+stores
EntryPageUrl http://travel.nytimes.com/2007/06/17/travel/17hours.html
Pageviews 76
TotalVisits 4
TimeSinceFirstVisit (days) 136.31
UserAgent Mozilla/4.0 (compatible;MSIE 7.0;Windows NT 5.1;.NET CLR 1.1.4322;...
TotalClicks 1
IfClicked 1

"http://www.google.com/search?q=**tea+india**&sourceid=navclient-ff&ie=UTF-8&rlz=1B3GGGL_enUS242US

"http://www.google.com/search?hl=en&q=**fusion+menu**"

"http://www.google.com/search?q=**frequent+flier+mile+tracker**&ie=utf-8&oe=utf-8&aq=t&rls=org.moz

"http://www.google.com/search?q=**taos+ski+valley**&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:en-US:o

"http://www.google.com/search?hl=en&q=**Coco+Plum+Caba%C3%B1as+panama**"

"http://www.google.co.uk/search?hl=en&rlz=1T4ADBF_en-GBGB261GB261&q=d**rug+runners**&meta="

"http://www.google.ie/search?hl=en&q=**argan+oil+psoriasis**&start=30&sa=N"

"http://www.google.co.uk/search?hl=en&sa=X&oi=spell&resnum=0&ct=result&cd=1&q=**Port+of+Spain+tr**

"http://www.google.com/search?sourceid=navclient&ie=UTF-8&rls=GGLD,GGLD:2005-15,GGLD:en&**q=new+**

"http://www.google.com/search?q=**quaintest+towns+in+the+united+states**&btnG=Search&hl=en&sa=2"

"http://www.google.com/search?hl=en&q=**st.gotthard+pass+weather+december**&aq=f&oq="

"http://www.google.com/search?q=**seeko%27o+hotel+bordeaux**&rls=com.microsoft:*&ie=UTF-8&oe=UTF-8

"http://www.google.ca/search?q=**bed+and+breakfast+mont+tremblant+dog+sledding+packages**&start=20

"http://www.google.com/search?hl=en&q=**PALMYRA+NY**&btnG=Search"

"http://www.google.com/news?hl=en&tab=wn&nolr=1&q=**santa+barbara+fire+map**&btnG=Search"

"http://www.google.com/search?hl=en&q=**36+hours+in+milwaukee**&aq=f&oq="

"http://www.google.com/search?q=**new+york+city**&rls=com.microsoft:en-us:IE-Address&ie=UTF-8&oe=U

"http://www.google.com/ig/gmailmax?hl=en&mid=23"

"http://www.google.com/search?hl=en&rls=com.microsoft:en-us:IE-SearchBox&rlz=1I7RNWE&q=**outward**

"http://www.google.com/search?hl=it&lr=&client=firefox-a&channel=s&rls=org.mozilla:it:official

"http://www.google.co.in/search?hl=en&q=**sightseeing+around+england**&start=10&sa=N"

"http://news.google.com/news?q=**santa+f**e,+NM&rls=com.microsoft:en-us:IE-SearchBox&ie=UTF-8&oe=U

"http://www.google.com/search?hl=en&q=**week+end+in+new+york+last+minute**&aq=f&oq="

north america highest peaks climbing
biggest ship
lake little big wolf new york
hot springs in the northern california
twin tip skiis youth
black people and boston
biltmore neighborhood phoenix
new york times berlin
vermont ny
killington or okemo
compiegne june 22 photo
new york city tour
trips to mekong delta
how to curl reed for basket weaving
bicycle night ride around central park
small paragraph about marrakech
careyes mexico
best historic sites in the u.s
berlin rent a glass of wine
restauramts in italy
brasilia red light district
hyatt layoffs
travel time nyc providence
amsterdam
indian restaurant in costa rica
arizona desert survival courses
bolivian death road replacement
ny times prague
where to eat in new york near public library
introduction to the city of san cristobal
remote cabins ny
36 hours in buenos aires
pizza dough mario
bull wrestling
holiday markets new york city
ski resorts in new york
jordan travel guide
strip clubs in dubai
verbier switzerland
colonial days homes and buildings in new york
cinque terra italy
the chesterfield inn
nytimes mit museum
blood mountain loop trail
rock climbing in red rock canyon
charles de gaulle airport day room
star boys skansen
restaurants in paris
puebla mexico
huatulco cultures
nytimes travel
washington d c time
toronto new york flights
ho chi minh trail
outward bound children
ski underwear
sightseeing around england
hiking in napa valley
santa fe, nm
week end in new york last minute
2008 galapagos new york times
new york times san diego 36 hours
1 week on hawaii - what to see
santo domingo
argan oil
champlain map route
selling my chula vista condotel
new york city
currency exchange locations
state farm rental car insurance
donald m kendall sculpture gardens
toulouse art
best restaruants mexico city
ski resort near new york city

total: 35,936 words

cnt	word
1333	new
1174	york
1152	in
721	times
701	to
563	travel
355	best
352	the
309	ny
304	hotel
272	of
248	ski
248	city
226	restaurants
205	hours
193	36
190	nytimes
174	and
170	paris
166	restaurant
161	nyc
158	for
147	mexico
139	exchange
138	san
134	where
129	go
125	guide
124	from
122	italy
118	on
108	currency
107	de
105	hotels
103	winter
102	aires
101	buenos
95	a
92	gordon
89	oil
89	london
88	park
83	cinque
82	nyt
81	time
78	santa
77	resort
75	map
75	cartagena
74	india
72	skiing
72	how
71	terre
71	sightseeing
70	la
66	beach
65	what
65	reviews
64	argan
62	cruise
60	review
60	mountain
60	food
60	dubai
60	barcelona
59	places
59	do
59	at
58	stay
58	art
57	tour
57	is
56	place
55	shanghai
55	christmas
54	day
53	rental
53	house
53	colombia
52	street
52	resorts
52	france
51	ramsay
51	martin
51	kate
50	valley
50	st
50	middleton
49	south
49	los
48	spain
48	seattle
48	green
48	grand
48	bar
47	rio
47	kids
47	caribbean
47	car
46	national
46	lines
45	introduction
45	fe
44	with
44	top
44	socotra
44	ramsey
44	francisco
44	country
42	town
41	trip
41	chi
40	tokyo
40	ho
40	flight
39	minh
39	england
39	cheap
39	brazil
39	boston
38	puerto
37	tours
37	taos
37	puebla
37	one
37	island
36	shopping
36	itinerary
36	america
36	2008
35	st.
35	sea
35	portland
35	miami
35	club
35	bus
35	about
34	trail
34	small
34	near
34	montreal
34	bars
33	world
33	rome
33	indian
33	europe
32	steak
32	spa
32	rv
32	real
32	panama
32	market
32	islands
32	flights
32	berlin
31	towns
31	skis
31	or
31	night
31	le
31	istanbul
31	chicago
30	west
30	tremblant
30	rico
30	nude
30	museum
30	eat
30	central
30	beer
29	weekend
29	week
29	twin
29	las
29	lake
29	juan
29	estate
29	airlines
28	when
28	tayrona
28	state
28	ranch
28	locations
28	hot
27	road
27	peru
27	moroccan
27	long
27	el
27	bangkok
27	airline
26	riviera
26	rate
26	old
26	naples
26	music
26	madrid
26	janeiro
26	california
26	by
25	years
25	wine
25	united
25	tips
25	jose
25	inn
25	french
25	children
25	big

A slight adjustment

To remove the effect of frequent, repeat visitors, the Times chooses to look at **only a user's first visit to the site during the experimental period**; this reduces our data set from about 200K to 130K

One of the first questions we can ask of these data uses tools we've developed for "dichotomous responses"; that is, results that **we can display in a two-by-two table**

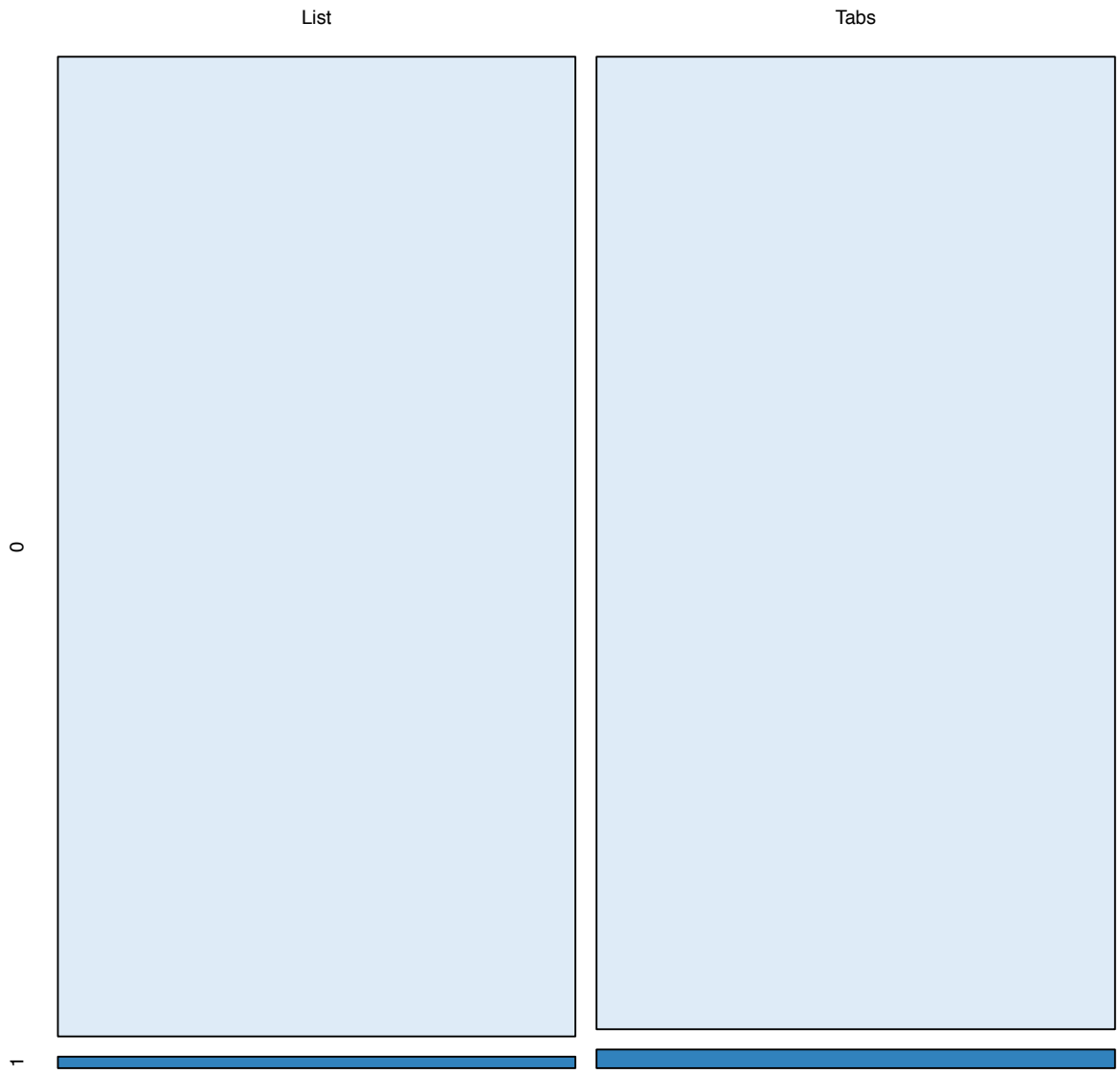
Do the different designs attract reader's interest differently? Put another way, are visitors equally likely to click on the two different representations for the "most popular" lists

A first look

Below we have a few R commands to read in the Times Travel Section data and to make a simple two-by-two table of (a portion of) the experimental results

```
> source("http://www.stat.ucla.edu/~cocteau/stat105/data/nyt.R")
> dim(nyt)
[1] 132027     13
> names(nyt)
 [1] "User_ID"             "UserVisit_ID"        "StartTime_SSE"
 [4] "StartTime_English"   "VisitLength"         "Variation"
 [7] "RefererUrl"          "EntryPageUrl"        "Pageviews"
[10] "TotalVisits"         "TimeSinceFirstVisit" "UserAgent"
[13] "IfClicked"

> table(nyt$Variation,nyt$IfClicked)
           0     1
  List 65181   766
  Tabs 64836  1244

> 766/(766+65181)
[1] 0.01161539
> 1244/(1244+64836)
[1] 0.01882567
> (1244/(1244+64836))/(766/(766+65181))
[1] 1.620752
```

It seems like the tabbed display is more popular (although both are fairly unused features) -- Is this all we can say? How can we make sure this wasn't just random, the result of our assignment into treatment and control, but an actual effect?

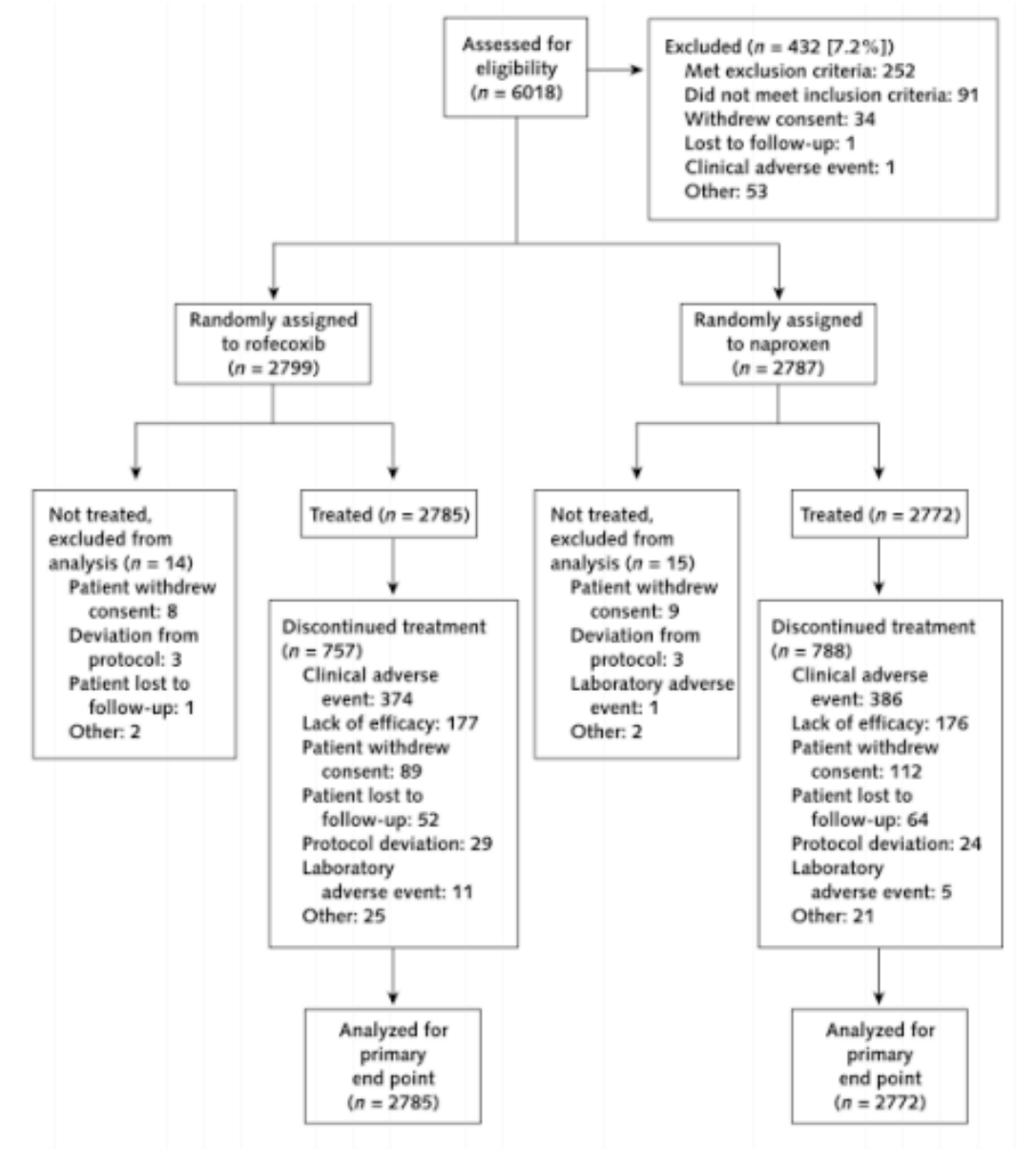

NYT Travel experiment
List
Tabs
0
1

Clinical trials

Randomized allocation into treatment and control comes up in a host of other contexts -- It has a long history in clinical trials

We'll take a slight historic detour and talk about the evolution of this design in medicine -- It provides us with a lot of intuition that we can bring back to the A/B website testing example

We'll focus initially on the very first published randomized controlled trial...



Clinical trials

A clinical trial is simply an experimental study in which two or more treatments are assigned to human subjects -- Experimental studies in all areas of biology have been greatly informed by procedures used in clinical trials

The clinical trial, however, has evolved considerably -- It was not always the "gold standard" of experimental designs -- Richard Doll (a well known epidemiologist who studied lung cancer) noted that before 1946

> *... new treatments were almost always introduced on the grounds that in the hands of professor A or in the hands of a consultant at one of the leading teaching hospitals, the results in a small series of patients (seldom more than 50) had been superior to those recorded by professor B (or some other consultant) or by the same investigator previously.* ***Under these conditions variability of outcome, chance, and the unconscious (leave alone the conscious) in the selection of patients brought about apparently important differences in the results obtained****; consequently, there were many competing new treatments*

Clinical trials

In an attempt to improve the evaluation of different treatments, **Austin Bradford Hill began advocating a more systematic approach to designing clinical trials**; like Doll, he was frustrated with the quality of research at the time, going so far as to question the ethics of the existing system

Hill was the son of a distinguished physiologist; his hope of a medical career was thwarted by the onset of tuberculosis in 1917, and instead, while an invalid, he completed a degree in economics by correspondence

In 1927 Hill moved to the London School of Hygiene and Tropical Medicine and during the 1930s he researched mainly in occupational epidemiology; his renown in medical statistics started in 1937 with the publication of his textbook, *Principles of Medical Statistics*, based on a series of articles in the Lancet

Clinical Trials

Hill's work emphasizes the **practical snags and difficulties of applying statistics** in a clinical setting rather than theoretical minutiae -- It seems that his advice, while often statistically sound, was motivated by practical concerns

In terms of clinical trials, Hill argued for **well-specified study aims or outcomes**, and the consistent use of controls -- Patients were to be divided into two groups: **the "treatment" group would receive a new drug or procedure**, while **the "control" group would be prescribed the standard therapy**

Upon completion of the trial, researchers would examine the differences between the two groups, measuring outcomes, and determine if the proposed treatment is superior to the existing therapy

With his very practical approach to clinical work, Hill took a special interest in how patients were divided into the treatment and control groups -- **Left solely to physicians, he felt there could be a problem**

What was he worried about?

Clinical Trials

To remove the subjective bias of physicians in making assignments, some clinicians (including HIII, initially) had recommended the so-called **alternation method** -- That is, as patients appear at a clinic or study center, researchers alternately assign them to treatment or control

Other similar schemes include the assignment of a patient based on his or her initials or even their birthdate -- Taking Hill's very practical stance, do these methods completely remove potential bias?

Clinical trials

In 1948, Hill published a groundbreaking study on the effectiveness of streptomycin (an antibiotic) in treating pulmonary tuberculosis; here is how he assigned patients to the treatment and control groups

Determination of whether a patient would be treated by streptomycin and bed-rest (S case) or by bed-rest alone (C case) was made by reference to a statistical series based on ***random sampling numbers drawn up for each sex*** *at each centre by Professor Bradford Hill; the details of the series were unknown to any of the investigators or to the co-ordinator and were contained in a set of sealed envelopes, each bearing on the outside only the name of the hospital and number. After acceptance of a patient by the panel, and before admission to the streptomycin centre,* ***the appropriate numbered envelope was opened at the central office: the card inside told if the patient was to be an S or C case****, and this information was then given to the medical officer of the centre. Patients were not told before admission that they were to get special treatment; C patients did not know throughout their stay in hospital that they were control patients in a special study; they were in fact treated as they would have been in the past, the sole difference being that they had been admitted to the centre more rapidly than was normal. Usually they were not in the same wards as S patients, but the same regimen was maintained."*

Medical Research Council. Streptomycin treatment of pulmonary tuberculosis. BMJ 1948; 2: 769-782.

An aside: Some history

Following the **immense success of penicillin**, there was a great deal of research activity around detecting other potential antibiotics

Also, tuberculosis was the **"most important cause of death"** of young adults in Europe and North America at the time

Considerable laboratory work and some early experiments on patients suggested that Streptomycin would be an effective treatment for pulmonary tuberculosis

The MRC randomized trial of streptomycin and its legacy: a view from the clinical front line, J. Crofton
http://jrsm.rsmjournals.com/cgi/reprint/99/10/531

Clinical Trials

The tuberculosis study was the first time **randomization of treatments** was used in a clinical trial; after its publication, Hill wrote a series of articles describing its use

> *In these articles, I had set out the need for controlled experiments in clinical medicine with groups chosen at random. At the outset, I think I pleaded that trials should be made using alternate cases. I suspect if (and its a very large IF) if that, in fact, were done strictly they would be random.* ***I deliberately left out the words "randomization" and "random sampling numbers" at that time, because I was trying to persuade the doctors to come into controlled trials in the very simplest form and I might have scared them off.*** *I think the concepts of "randomization" and "random sampling numbers" are slightly odd to the layman, or, for that matter, to the lay doctor, when it comes to statistics. I thought it would be better to get doctors to walk first, before I tried to get them to run.*
>
> *Memories of the British streptomycin trial in tuberculosis: The first randomized clinical trial*, Sir Austin Bradford Hill

Clinical Trials

Through randomization (and the blinding of the physicians), Hill achieved his goal of reducing bias by allocating “the patients to the ‘treatment’ and ‘control’ groups in such a way that the two groups are initially equivalent in all respects relevant to the inquiry” -- He writes

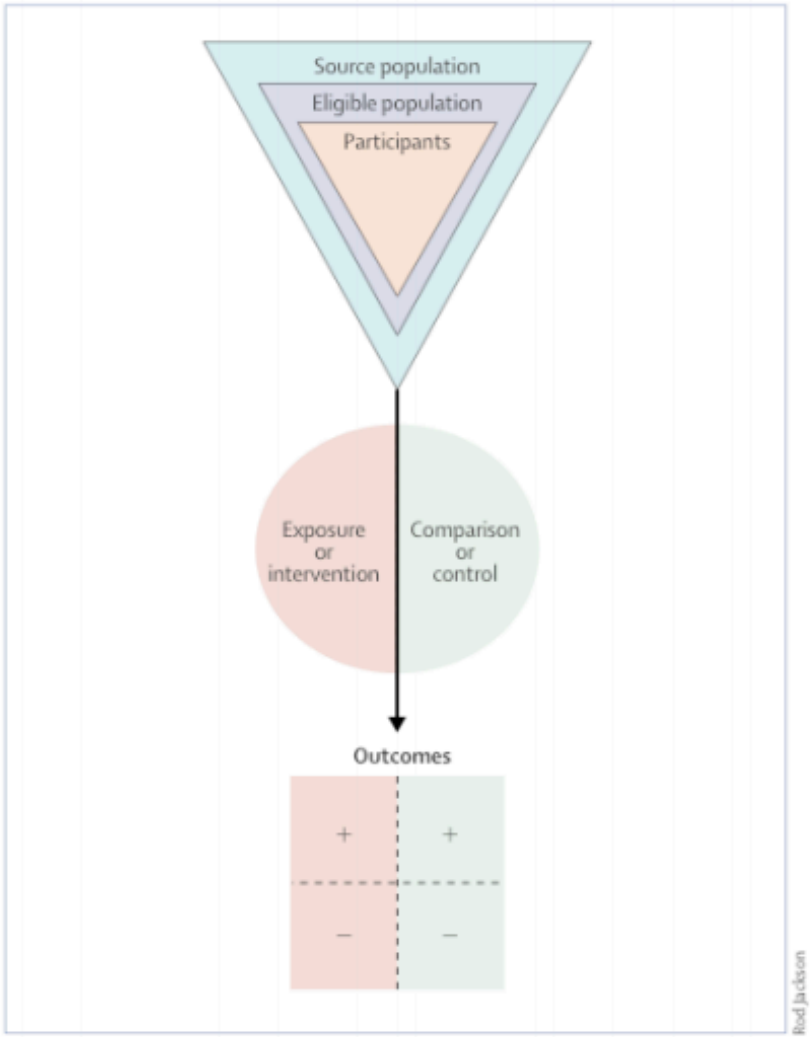
> It ensures that neither our personal idiosyncrasies (our likes or dislikes consciously or unwittingly applied) nor our lack of balanced judgement has entered into the construction of the different treatment groups—the allocation has been outside our control and the groups are therefore unbiased;
>
> ... it removes the danger, inherent in an allocation based on personal judgement, that believing we may be biased in our judgements we endeavour to allow for that bias, to exclude it, and that in doing so we may overcompensate and by thus ‘leaning over backward’ introduce a lack of balance from the other direction;
>
> ... and, having used a random allocation, the sternest critic is unable to say when we eventually dash into print that quite probably the groups were differentially biased through our predilections or through our stupidity.

Memories of the British streptomycin trial in tuberculosis: the first randomized clinical trial. *Control Clin Trials* 1990;11:77–7

Randomized controlled trials

As an experiment then, the design is straightforward: participants are assigned randomly to either receive a treatment under study or a “control,” perhaps a placebo or a standard therapy

At the end of the study, an outcome is recorded for each participant; in some cases, scientists are evaluating whether a drug helps with a particular condition, say



Fisher and randomization

It would be incorrect to suggest that the idea of randomization is due to Hill; Hill was working in the 1940's and 50's and became an advocate of randomization on fairly practical grounds (reducing bias)

In the 1920s and 1930s, R. A. Fisher was promoting the idea of randomization from a technical perspective -- To Fisher, randomization gave rise to valid statistical procedures, a point we'll see shortly

Fisher and randomization

"*The theory of estimation presupposes a process of random sampling. All our conclusions within that theory rest on this basis; without it our tests of significance would be worthless. ... In controlled experimentation it has been found not difficult to introduce explicit and objective randomisation in such a way that the tests of significance are demonstrably correct. In other cases we must still act in faith that Nature has done the randomisation for us....* ***We now recognise randomisation as a postulate necessary to the validity of our conclusions, and the modern experimenter is careful to make sure that this postulate is justified.***"

Fisher RA. *Development of the theory of experimental design*. Proceedings of the International Statistical Conferences 1947;3:434–39

Another aside: Fisher and Hill

There is, in fact, an interesting story that connects these two researchers; both were active in roughly the same time period and they were certainly aware of each other's work

They exchanged correspondence starting in 1929, **"Dear Sir"**; and then in 1931 **"Dear Fisher"** and **"Dear Bradford Hill";** and then in 1940 **"My dear Fisher"** and **"My dear Bradford Hill"**; and then by 1952 **"My dear Ron"** and **"My dear Tony"** (Hill went by Tony)

But by 1958 they were back to **"Dear Fisher"** and **"Dear Bradford Hill"** as the two (Doll, a significant co-investigator with Hill) were on opposites sides in a dispute as to **whether or not smoking caused lung cancer**

From the point of view of our discussion, one of Fisher's main criticisms of the studies suggesting that smoking caused lung cancer was the fact tha**t they were entirely observational** -- He wanted a "properly randomized experiment" (which of course would be difficult as you can't force people to start smoking)

We will speak more about causation and what you can conclude from different types of studies over the next couple of lectures

Hill's tuberculosis trial

But back to the task at hand; here is a simple summary of the patients enrolled in Hill's tuberculosis trial -- as Hill hoped, the groups seem relatively well balanced in terms of their measured "condition"

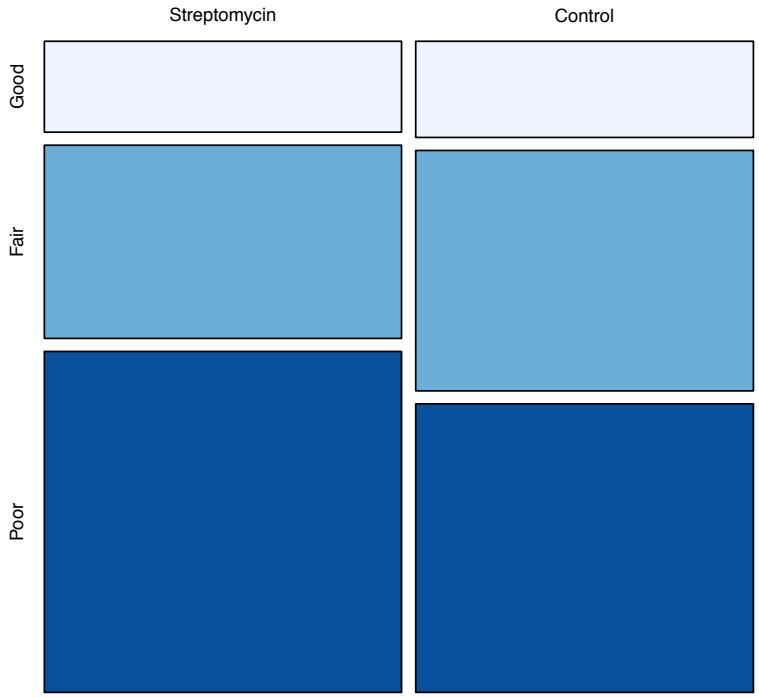
TABLE I.—*Condition on Admission*

General Condition	S Group	C Group	Max. Evening Temp. in First Week*	S Group	C Group	Sedimentation Rate	S Group	C Group
Good .	8	8	98–98·9° F. (36·7–37·15° C.)	3	4	0–10	0	0
Fair . .	17	20	99–99·9° F. (37·2–37·75° C.)	13	12	11–20	3	2
Poor . .	30	24	100–100·9° F. (37·8–38·25° C.)	15	17	21–50	16	20
			101° F. (38·3° C.) +	24	19	51 +	36	29
Total	55	52	Total	55	52	Total	55	51†

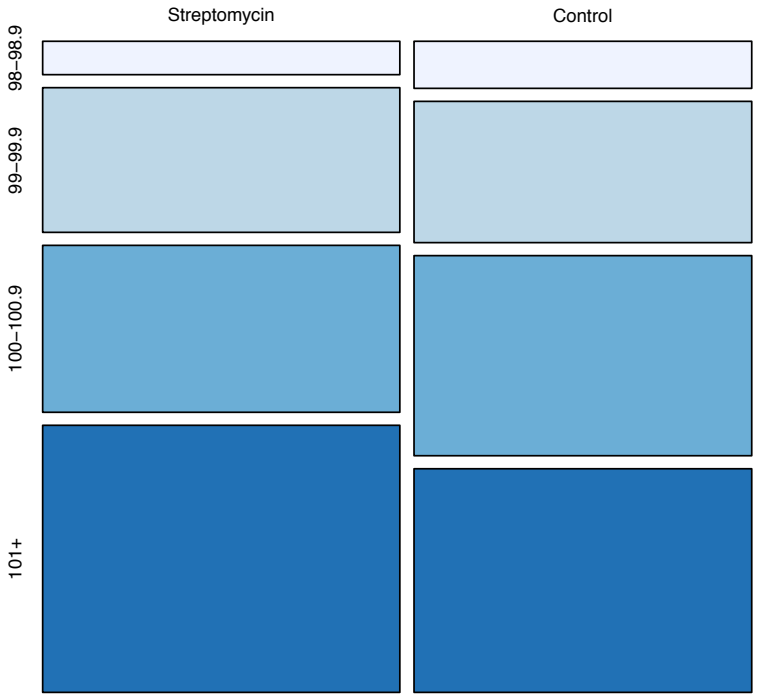
* Temperature by mouth in all but six cases. †Examination not done in one case.

Medical Research Council. Streptomycin treatment of pulmonary tuberculosis. BMJ 1948; 2: 769-782.

Condition on Admission, after Hill



Max. Evening Temp., after Hill



Hill's tuberculosis trial

And here are Hill's original results from his 1948 paper; what do we see?

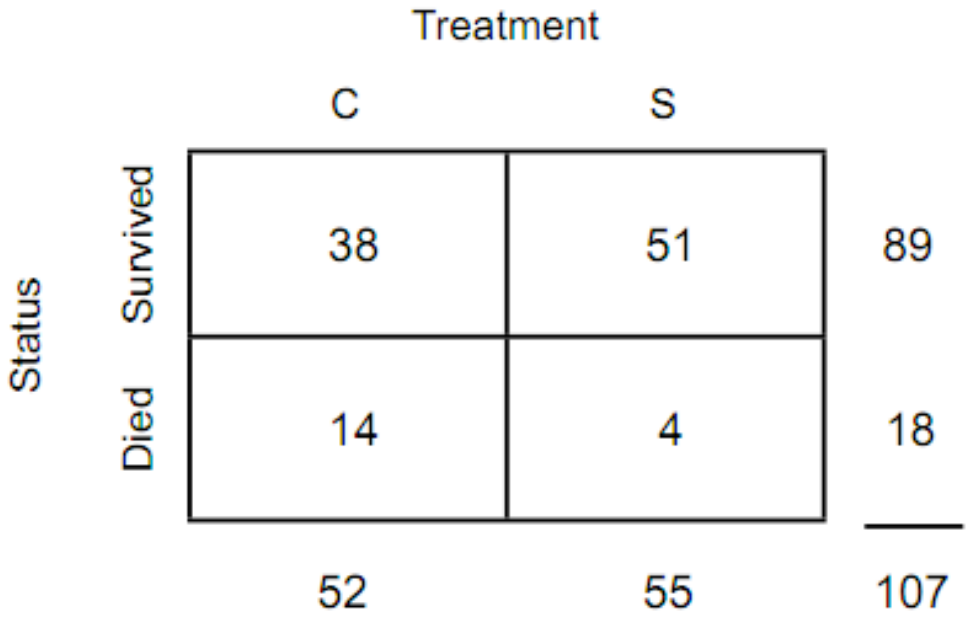
TABLE II.—*Assessment of Radiological Appearance at Six Months as Compared with Appearance on Admission*

Radiological Assessment	Streptomycin Group		Control Group	
Considerable improvement ..	28	*51%*	4	*8%*
Moderate or slight improvement	10	*18%*	13	*25%*
No material change	2	*4%*	3	*6%*
Moderate or slight deterioration	5	*9%*	12	*23%*
Considerable deterioration ..	6	*11%*	6	*11%*
Deaths	4	*7%*	14	*27%*
Total	55	*100%*	52	*100%*

Medical Research Council. Streptomycin treatment of pulmonary tuberculosis. BMJ 1948; 2: 769-782.

Some analysis with Hill's data

Here we create a 2x2 table for Hill's data; we will focus on whether or not patients survived to the end of the trial

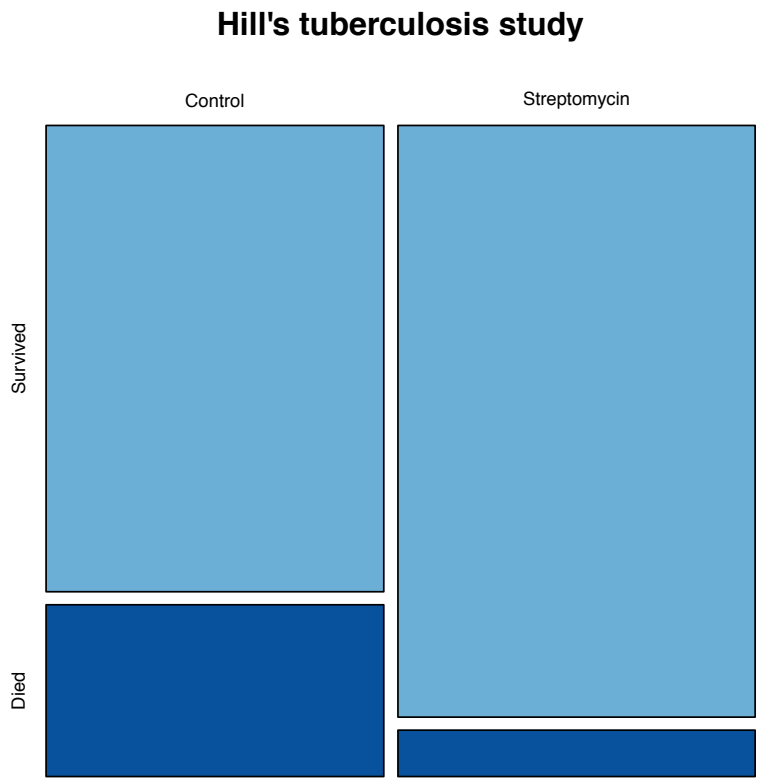


Another view

Here is a mosaic plot of Hill's tuberculosis study; it's worth taking a second look at the computations that go into this plot

Recall that the columns are sized according to the proportion of people receiving Streptomycin and the Control (slightly more received the treatment, 52 v. 55)

Then, within each column, the proportion of participants that survived is shaded yellow; we will call this **the conditional proportion who survived** given that a participant received either Streptomycin or the control



Some analysis with Hill's data

To work this out, we see that 14/52 or 27% of the patients receiving the control died; whereas 4/55 = 7.3% of those receiving Streptomycin died

What do we think?

		Treatment		
		C	S	
Status	Survived	38	51	89
	Died	14	4	18
		52	55	107

Some analysis with Hill's data

When you read about these kinds of trials in the medical literature, it is not uncommon **to work with a single figure of merit** -- Rather than look at the two conditional proportions, it is customary to look at their fraction

In this case, the ratio of the proportion of patients that died in the Streptomycin group (7.3%) to those that died in the Control group (27%) is 0.27 -- Streptomycin reduced the rate of mortality by nearly a quarter

This ratio is often called **the relative risk** -- The language comes from epidemiological studies where "treatment" is really exposure to some toxic substance and the outcome is not that you get better but that something horrible happens to you

Statistical analysis

On the face of it, things look promising for Streptomycin relative to the standard therapy, bed rest, but is that where our analysis stops?

How do we judge the size of an effect? In particular, could these results have occurred “by pure chance”?

And what is the model for chance here?

Randomized controlled trials

Below we present a simple cartoon of the steps that go into “analyzing” data from a trial such as this

1. We begin with **a null hypothesis**, a plausible statement (a model or scenario) which may explain some pattern in a given set of data made for the purposes of argument -- A good null hypothesis is a statement that would be interesting to reject

2. We then define **a test statistic**, some quantity calculated from our data that is used to evaluate how compatible the results are with those expected under the null hypothesis (if the hypothesized statement - or model or scenario - was true)

3. We then simulate values of the test statistic using the null hypothesis -- Today this will mean **simulating a series of data sets assuming the null hypothesis is true**, and for each computing the test statistic (the ensemble of simulated test statistics is often called a null distribution, but we’ll talk about this more formally when we review probability)

4. And finally, **we compare** the value of the test statistic calculated for our data and compare it to the values that were obtained by simulation -- If they are very different, we have evidence that the null hypothesis is wrong

Hill's tuberculosis study

So let's talk about each of these components in the context of Hill's randomized trial -- When testing the efficacy of a new medical procedure, **the natural null hypothesis is that it offers no improvement over the standard therapy**

Under this "model" we assume that the two treatments are the same, so that patients would have had **the same chance of survival under either** -- Put another way, **their outcome, whether they lived or died, would have been the same regardless of which group they were placed in**

Under this hypothesis, the table we see is merely the result of random assignment -- That is, 18 people would have died regardless of what group we assigned them to, and **the fact that we saw 4 in the Streptomycin group and 14 in the control group was purely the result of chance**

Hill's tuberculosis study

Therefore, under the null hypothesis, if we had chosen a different random assignment of patients, **we would still have 18 people who died and 89 who survived, but they would appear in different cells of the table**

We can simulate under this "model" pretty easily -- That is, we take the 18 people who died and the 89 who survived and we re-randomize, **assigning 52 of them to the control group and 55 to the treatment group**

Let's see what that produces...

Simulating random assignments

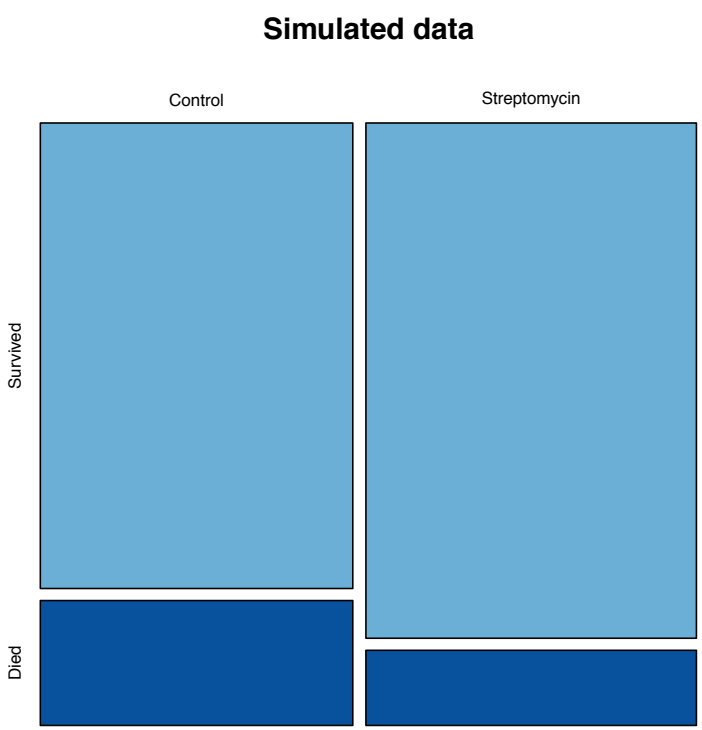
In this simulated table, we have 11/52 or 21% chance of dying under the control, and a 7/55 or 12% chance under Streptomycin; the treatment reduced the mortality rate among the participants by nearly 60%

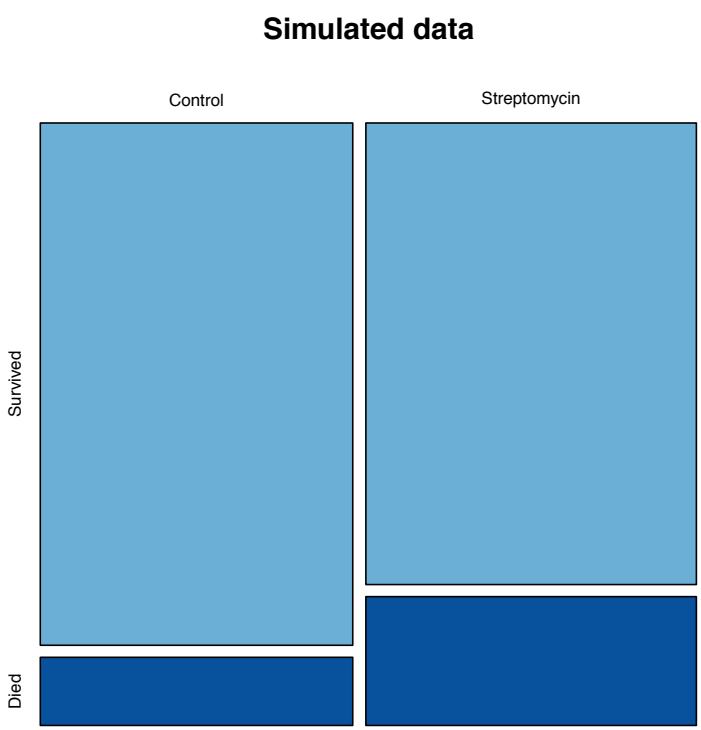
Status	Treatment C	Treatment S	
Survived	41	48	89
Died	11	7	18
	52	55	107

Simulating random assignments

In this simulated table, we have the opposite, with 6/52 or 12% chance of dying under the control, and a 12/55 or 22% chance under Streptomycin; the treatment almost doubled the mortality rate among the participants

Status	Treatment C	Treatment S	
Survived	46	43	89
Died	6	12	18
	52	55	107


Simulated data
Control
Streptomycin
Survived
Died


Simulated data
Control
Streptomycin
Survived
Died

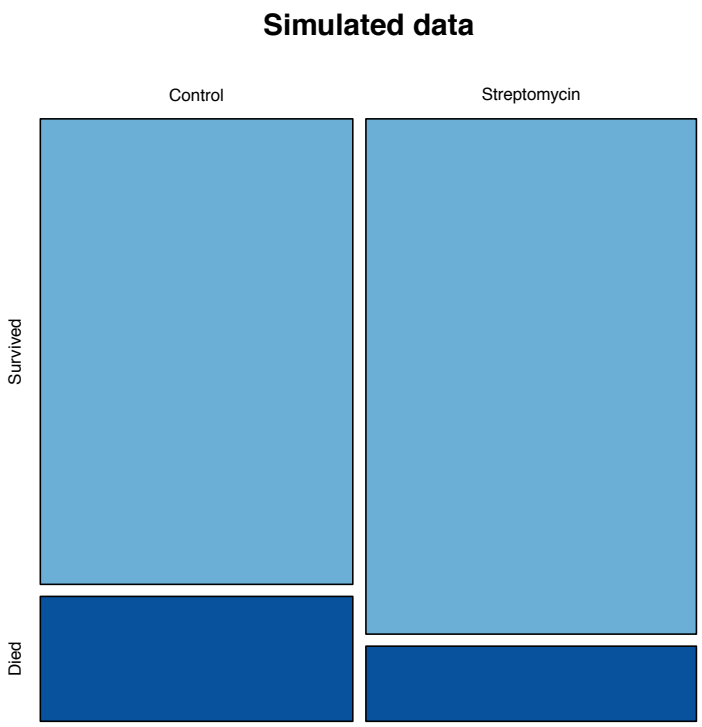
Simulating random assignments

Notice that we only need to record **one piece of information for each trial, the number of deaths under Streptomycin** -- Knowing that we know all the other entries in the table

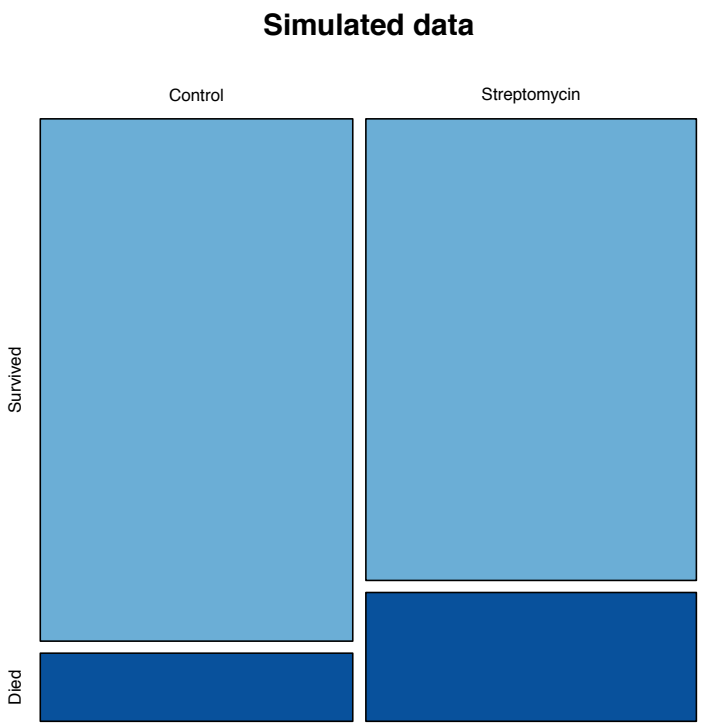
Using the language of hypothesis testing, we will take **the number of patients in the Streptomycin group that died as our test statistic**

Therefore, the question becomes, under the random assignment patients to treatments, **how common is it for us to see 4 or fewer deaths in the Streptomycin group?**

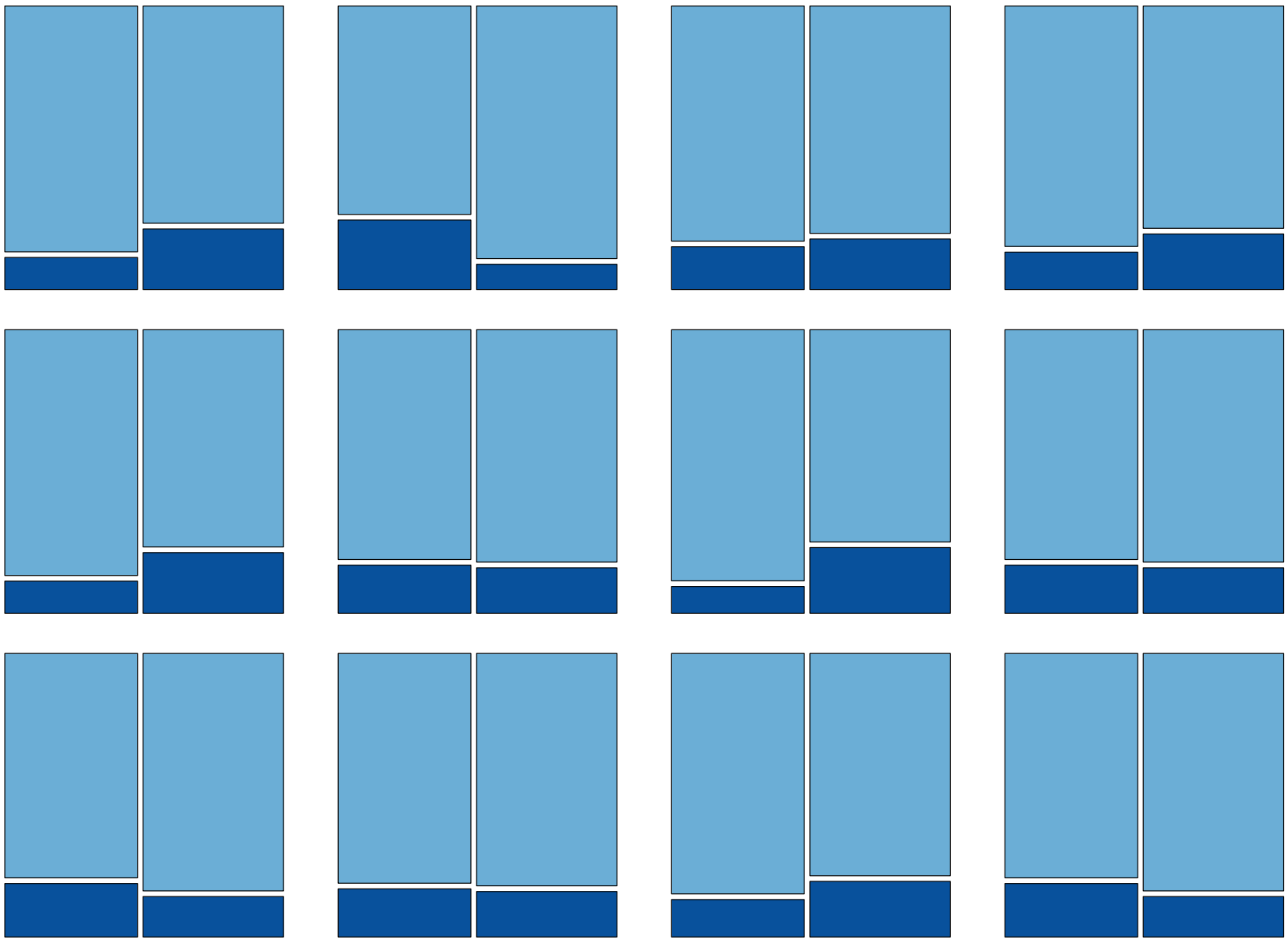
How would we figure this out?

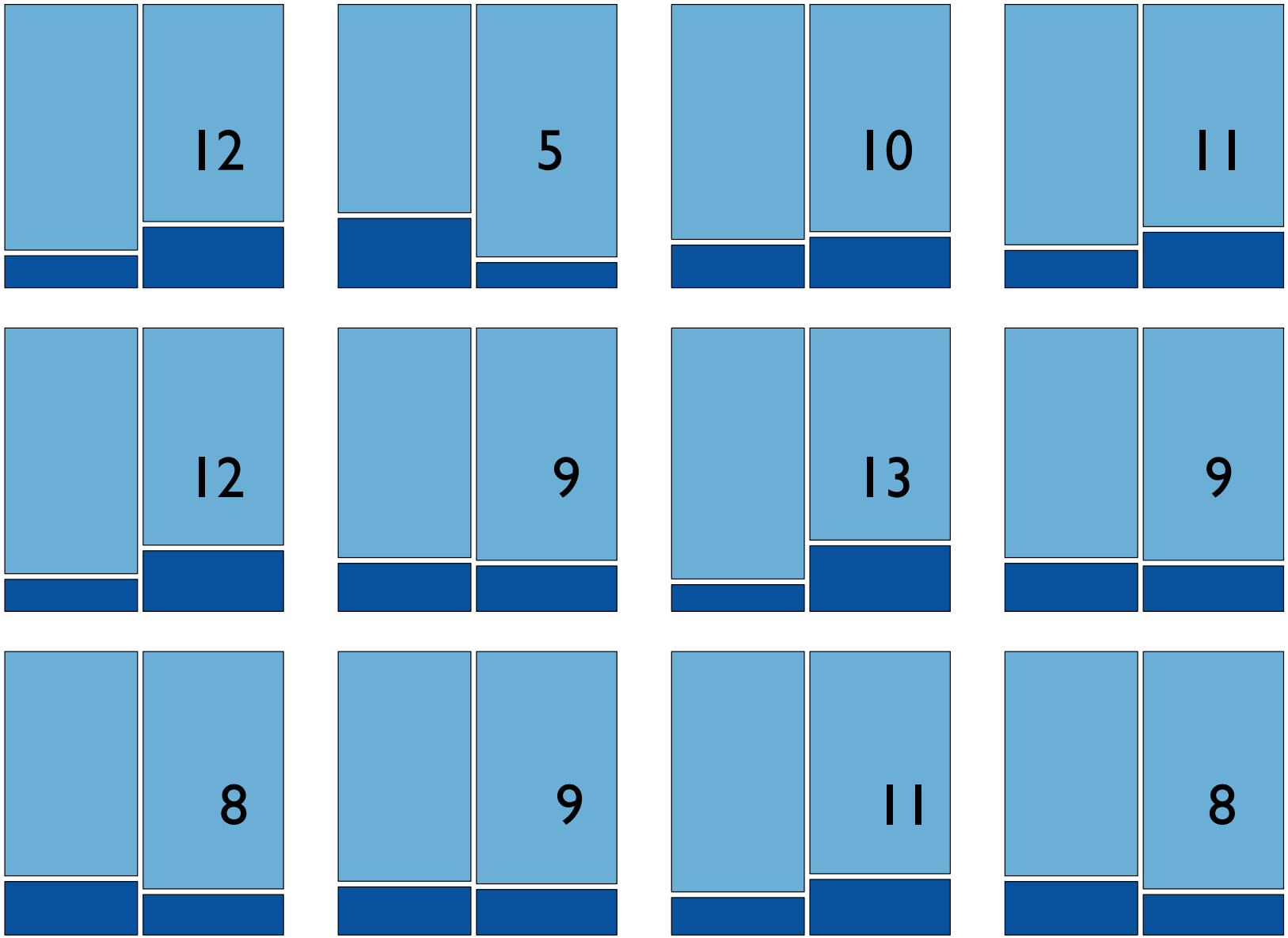


7 deaths in the Streptomycin group

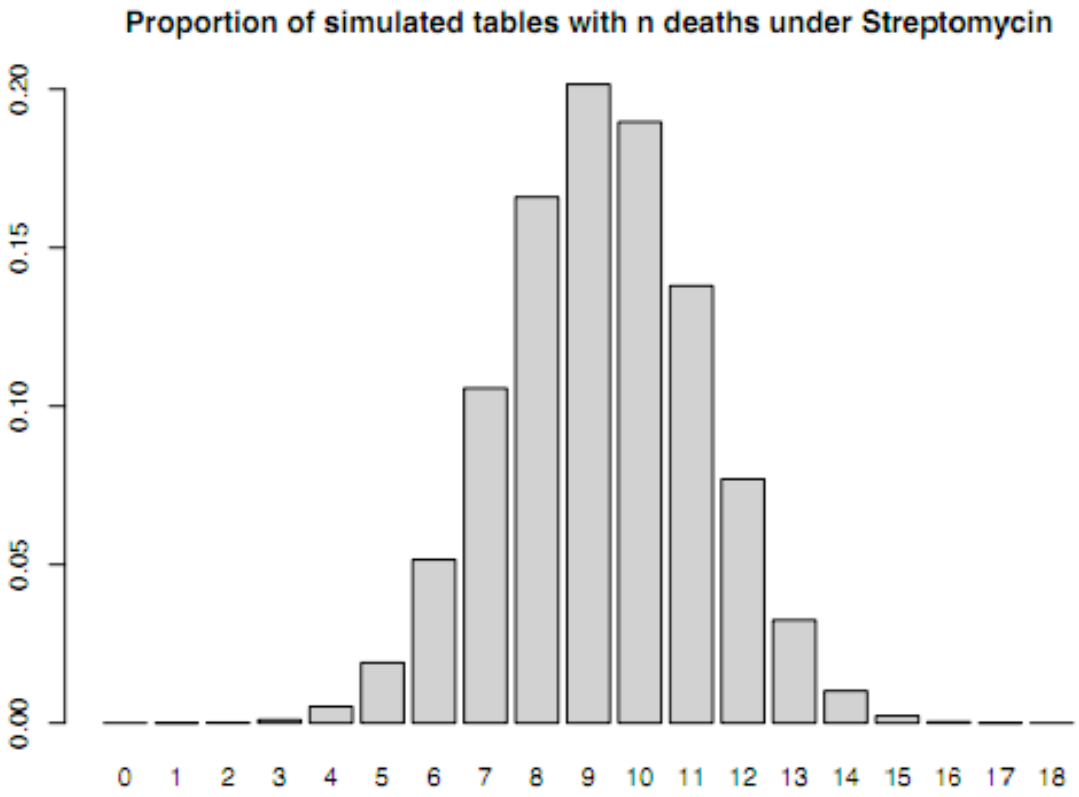


12 deaths in the Streptomycin group




12
5
10
11
12
9
13
9
8
9
11
8

12	5	10	11
12	9	13	9
8	9	11	8


Proportion of simulated tables with n deaths under Streptomycin
0.00
0.05
0.10
0.15
0.20
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

Simulating random assignments

In this plot we see that a value as small or smaller than four is fairly rare; to be precise, only 0.6% of the tables have 4 or fewer deaths in the Streptomycin group

This, then, provides us with evidence that there is something more at work here than random assignment

If we believed the null hypothesis, that there was no difference between Streptomycin and bed rest, the results HIII observed would have been extremely rare, coming up a very small fraction of the time