Thought leaders in data science and analytics:

Data Modeling

James G. Shanahan[1]

[1]***Independent Consultant***

EMAIL: James_DOT_Shanahan_AT_gmail_DOT_com

I 296A UC Berkeley

Lecture 2, Wednesday January 25, 2012

Outline

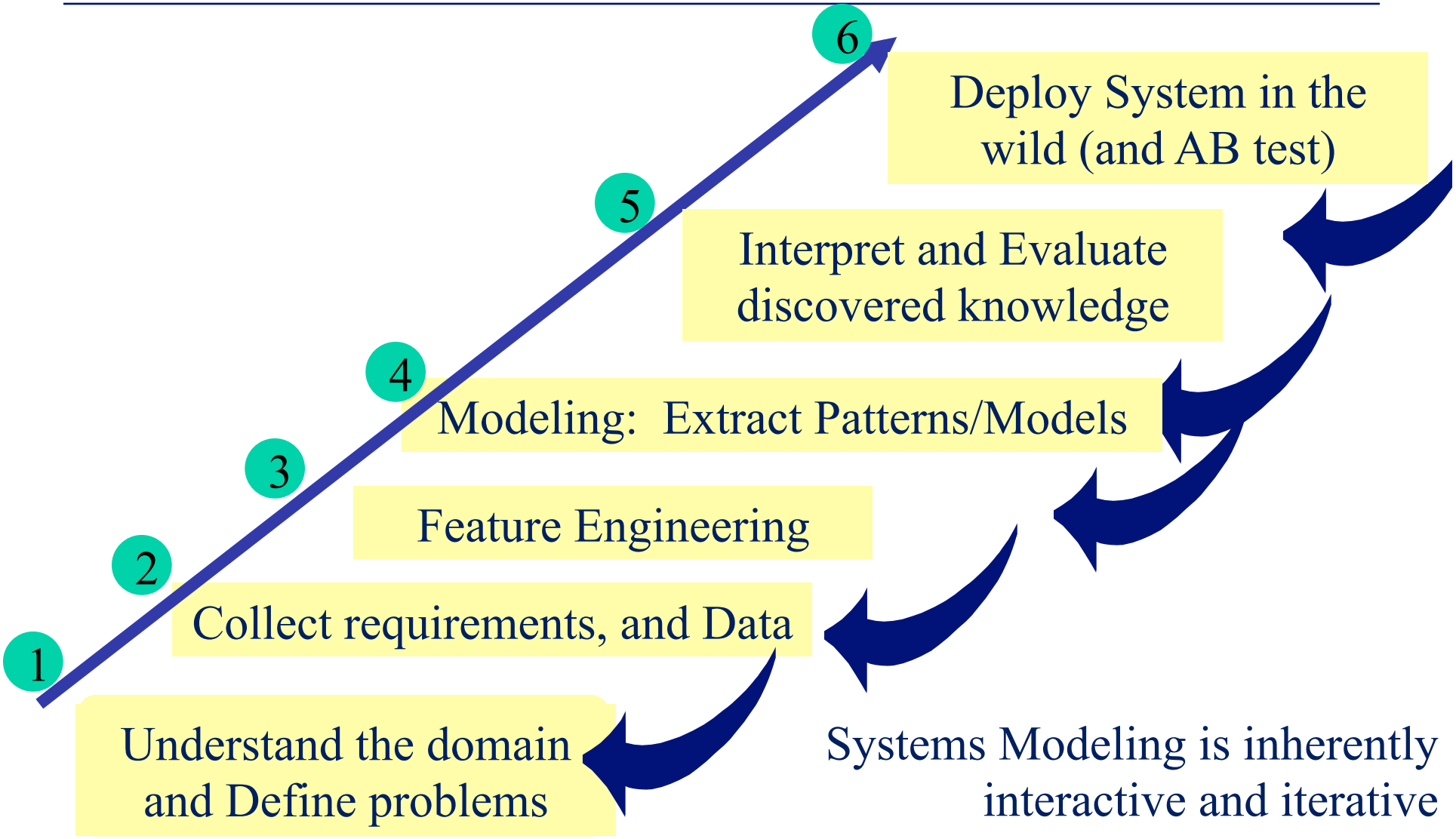
- Data Modeling 6-step process
- Machine learning
- Linear regression
- R Statistical programming environment

Data Analytics

- Analytics leverages advanced techniques from the disciplines of computer science, statistics, operations research, economics, and mathematics to solve problems in engineering and business departments.
  - Modern day analytic systems goes beyond the traditional business intelligence tasks of querying, reporting, OLAP, and alert tools.
- They generally involve ingesting massive amounts of domain-specific information (e.g., think in terms of tens of billions of webpages), data logging of user behavioral data (petabytes) , and analyzing these and other sources of information ultimately leading to the automatic construction of decision systems that operate in the 100-millesecond range.

- Examples of such problem domains include websearch, online advertising, media management, and healthcare.

6 Steps to Data Modeling in Practice



Machine Learning in one slide

- **Machine learning, a branch of artificial intelligence, is a scientific discipline that is concerned with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases.**
- **A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables.**
- **A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too large to be covered by the set of observed examples (training data).**
- **Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases. Machine learning, like all subjects in artificial intelligence, require cross-disciplinary proficiency in several areas, such as probability theory, statistics, pattern recognition, cognitive science, data mining, adaptive control, computational neuroscience and theoretical computer science.**

[WikiPedia]

What is the Learning Problem?

Learning = Improving with experience at some task

- **Improve over Task T**
- **with respect to performance measure P**
- **based on experience E**

Types of Learning

- Supervised learning - Generates a function that maps inputs to desired outputs. For example, in a classification problem, the learner approximates a function mapping a vector into classes by looking at input-output examples of the function.
- Unsupervised learning - Models a set of inputs: like clustering
- Semi-supervised learning - Combines both labeled and unlabeled examples to generate an appropriate function or classifier.
- Reinforcement learning - Learns how to act given an observation of the world. Every action has some impact in the environment, and the environment provides feedback in the form of rewards that guides the learning algorithm.
  Transduction - Tries to predict new outputs based on training inputs, training outputs, and test inputs.

Supervised Learning :Regression

- **Regression**
  - Linear Regression
- **Classification**
  - Logistic Regression
  - ?lm
- **Generalized Linear Models (GLMs)**
  - Broader family of models (that subsume Linear Regression and logistic regress and more
  - In R checkout ?glm()

Parametric Approaches vs. Non-parametric
Convex/Concave
Discriminative versus generative

Terminology: linear regression

W_i are the model coefficients

$$y = W_0 + W_1X_1 + W_2X_2 + \ldots + W_nX_n$$

y

Predicted

Response variable

Outcome variable

Dependent

X_i's

Predictor variables

Explanatory variables

Covariables

Independent variables

Y-intercept/threshold

Terminology: linear regression

W_i are the model coefficients

$$y = W_0 + W_1X_1 + W_2X_2 + \ldots + W_nX_n$$

y

Predicted

Response variable

Outcome variable

Dependent

X_i's

Predictor variables

Explanatory variables

Covariables

Independent variables

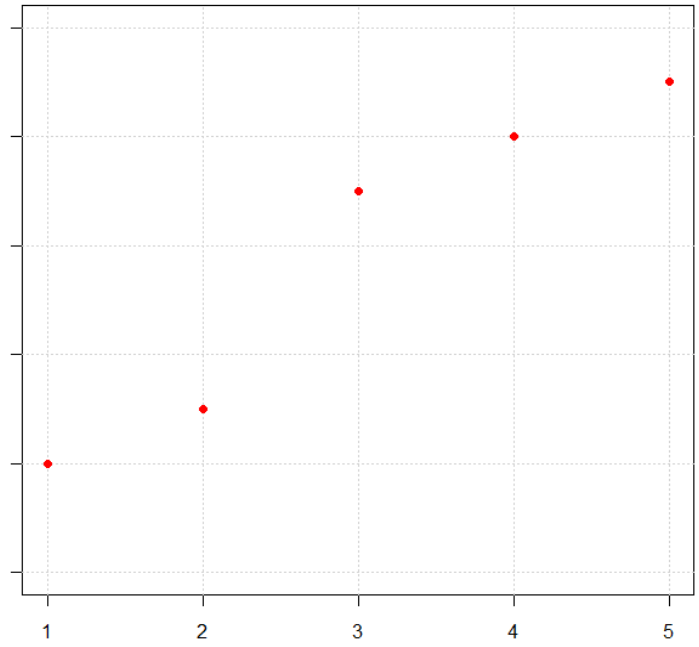
Y-intercept/threshold

Pr(Click): Advertising Problem

- **Predict Pr(Click|dwellTimeOnWebpage)**
  - at the times 1, 2, 3, 4, and 5 seconds after loading the page.
- **Graph each data point with time on the *x*-axis and CTR on the *y*-axis. Your data should follow a straight line.**
- **Use locator() to input data**
- **Find the equation of this line.**

#	x	y%
1	1	2
.	2	3
.	3	7
.	4	8
m	5	9

Temperature Dataset



X are features, aka variables, continuous, discrete, ordinal ($X \varepsilon \Re^n$)

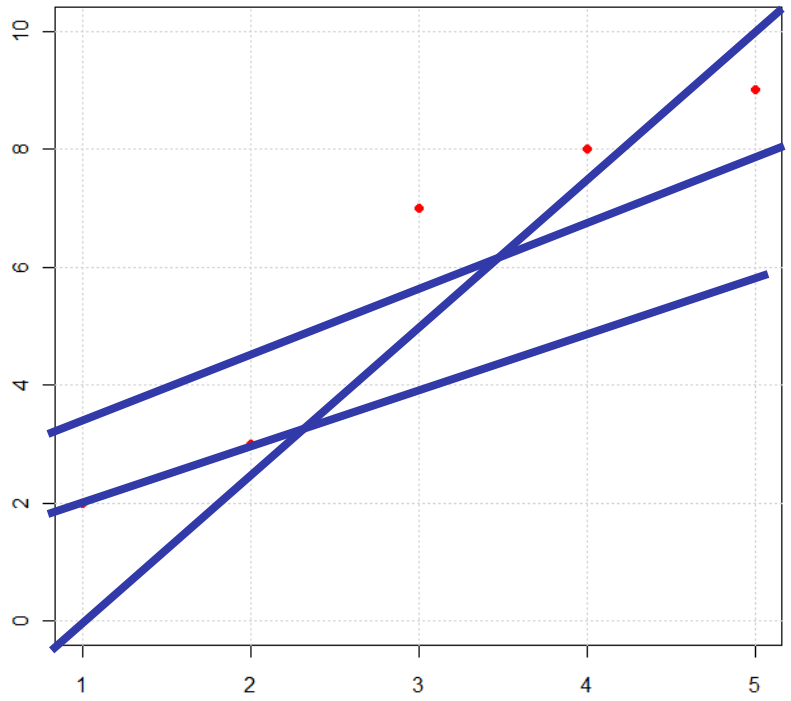
Generate Your Own Data

```
# You can generate data by clicking on a plot.
# Create data that illustrates the effect of varying 'f' and 'iter' in 'lowess'.
example.generateYourOwnData = function(){

  #change range of X and Y and experiment
  plot( c(0,1), c(0,1), type = 'n')
  xy <- locator(type = "p")      # create your own data set by clicking on the
                                 # left mouse button, then with the right mouse button
                                 # to finish. With a Macintosh cntrl-click outside the
                                 # plot to finish.
data1=data.frame(x=xy$x, y=xy$y)   #PLOT LINE
abline(coef(lm(y~x, data=data1)), col="red")

lines(lowess(xy, f = 2/3, iter = 3))  # here I've used the defaults
                                      # for f and iter,
                                      # experiment with other values
```

Least Square Fit Approximations

Suppose we want to fit the data set.

#	x	y
1	1	2
.	2	3
.	3	7
.	4	8
m	5	9

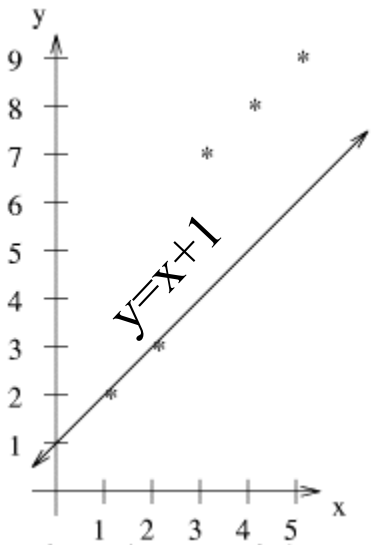


We would like to find the best straight line to fit the data?

Fit a line based on…

- **If we assume that the first two points are correct and choose the line that goes through them, we get the line *y* = 1 + *x*.**
- **If we substitute our points (x-values) into this equation, we get the following chart.**
- **How good is this line?**
  - The sum of the squares of the errors is 27.

$$y = mx + b$$
$$m = \frac{y_2 - y_1}{x_2 - x_1}.$$



x	y	predicted y	error	$(\text{error})^2$
1	2	2	0	0
2	3	3	0	0
3	7	4	3	9
4	8	5	3	9
5	9	6	3	9

SSE = 27

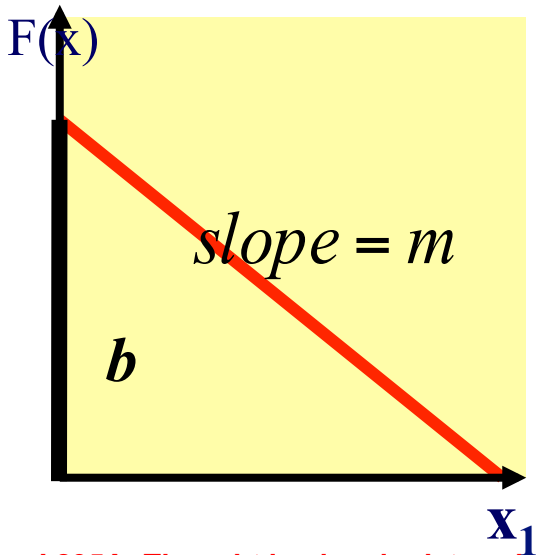
Do you think that we can do better than this?

Linear Model More Generally

- E.g., y=mx+b can be more generally seen a function of the form
- Here the W's are the parameters (also called weights) parametering the space of linear function mapping from X → Y=F(x)

$$y = f(x_0, x_1) = w_0 x_0 + w_1 x_1$$

$$= \sum_{i=1}^{n} w_i x_i = W^T X$$



Sometimes use θ instead of W

$$y = f(x_0, x_1) = \sum_{i=1}^{n} \theta_i x_i = \theta^T X$$

#	X0	x1	y
1	1	1	2
.	1	2	3
.	1	3	7
.	1	4	8
m	1	5	9

Linear Model: Ordinary Least Squares

Measuring Quality

- **How do we pick, or learn, the parameters W (aka θ)?**
- **One reasonable method seems to be to make f(x) close to y, at least for the training examples.**
- **To formalize, let's define a function that measures, for each possible model/hypothesis, W, how close f_θ (x^i)'s are to the corresponding y^i 's:**

$$J(W) = \sum_{i=1}^{m} \left| WX^i - y^i \right|$$

This error minimization is going to have problems?

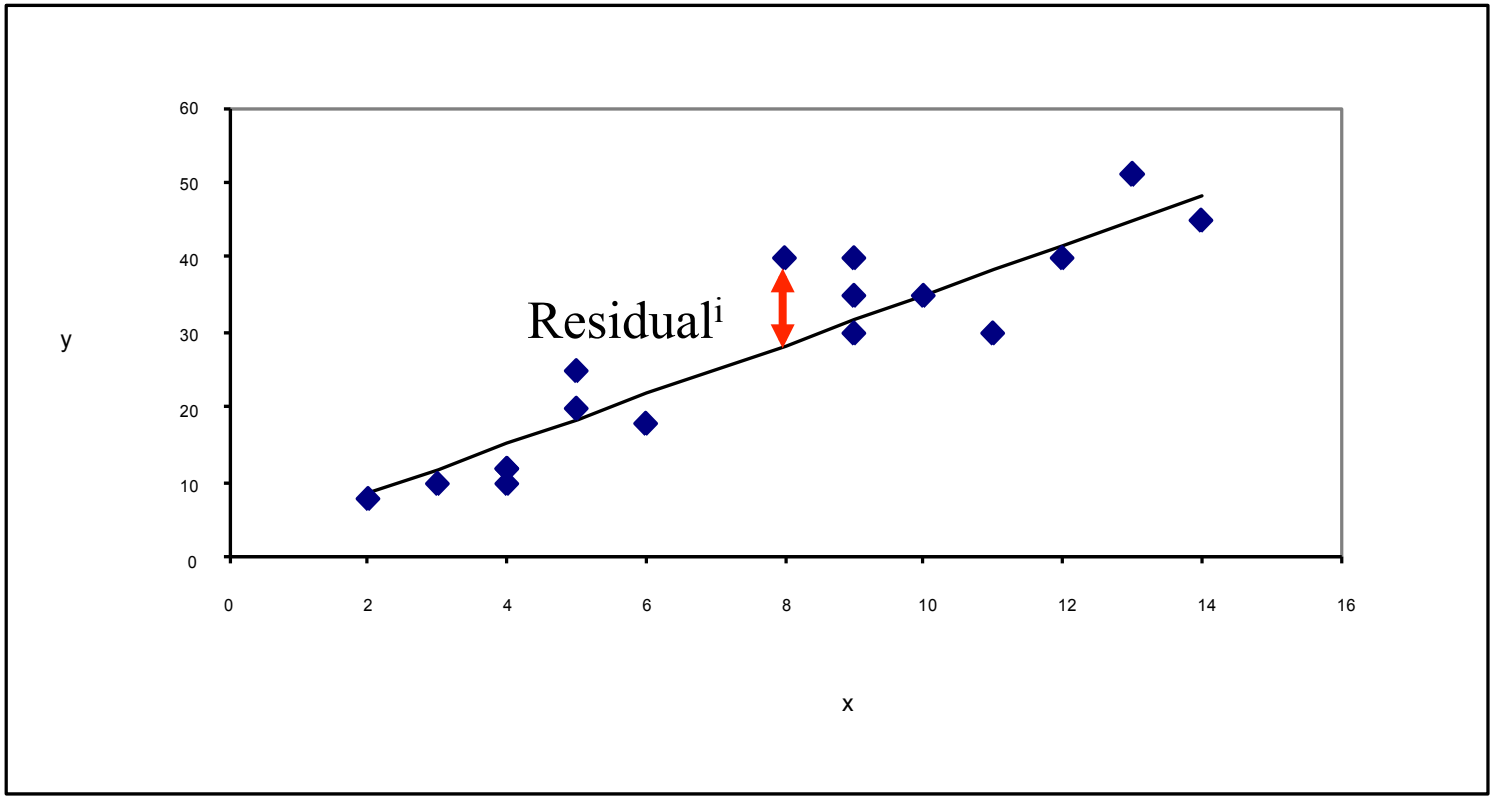
$$J(W) = \frac{1}{2} \sum_{i=1}^{m} \left(WX^i - y^i \right)^2$$

Residual sum of squares

- **Sum of squared error**
- **AKA Residual Sum of Squares (Residual squared)**

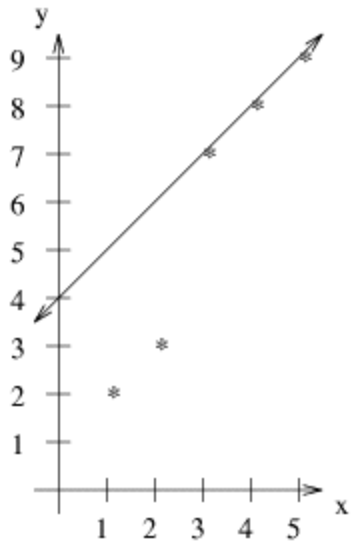
Residual

$$\text{Residual}^{\text{i}} = \left(WX^{i} - y^{i}\right)$$



Which Line is it anyway?

- **Select another two points and build a line**
- **If we choose the line that goes through the points when *x* = 3 and 4, we get the line *y* = 4 + *x*. Will we get a better fit? Let's look at it.**



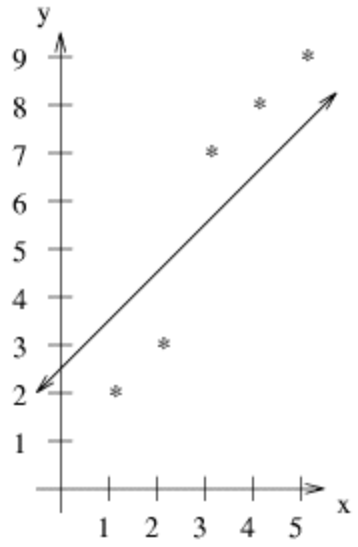
x	y	predicted y	error	$(\text{error})^2$
1	2	5	−3	9
2	3	6	−3	9
3	7	7	0	0
4	8	8	0	0
5	9	9	0	0

SSE = 18. Getting better but can we do better?

Can we do better than guesswork?

- **Let's try the line that is half way between these two lines. The equation would be *y* = 2.5 + *x*.**

- **Is there a more scientific or efficient way than guessing at which line would give the best fit.**
  - Surely there is a methodical way to determine the best fit line. Let's think about what we want.



x	y	predicted y	error	$(\text{error})^2$
1	2	3.5	−1.5	2.25
2	3	4.5	−1.5	2.25
3	7	5.5	1.5	2.25
4	8	6.5	1.5	2.25
5	9	7.5	1.5	2.25

SSE = 11.25. Getting better but can we do better?

Hypothesis Space of Linear Models

- **Here the W's are the parameters (also called weights) parameterizing the space of linear function mapping from X → Y = f(X)**
- **Augment Training Data with dummy intercept variable (simplifies notation and modeling)**

#	X0	x1	y
1	1	1	2
.	1	2	3
.	1	3	7
.	1	4	8
m	1	5	9

$$y = f(x_0, x_1) = w_0 x_0 + w_1 x_1$$

$$= \sum_{i=1}^{n} w_i x_i = W^T X$$

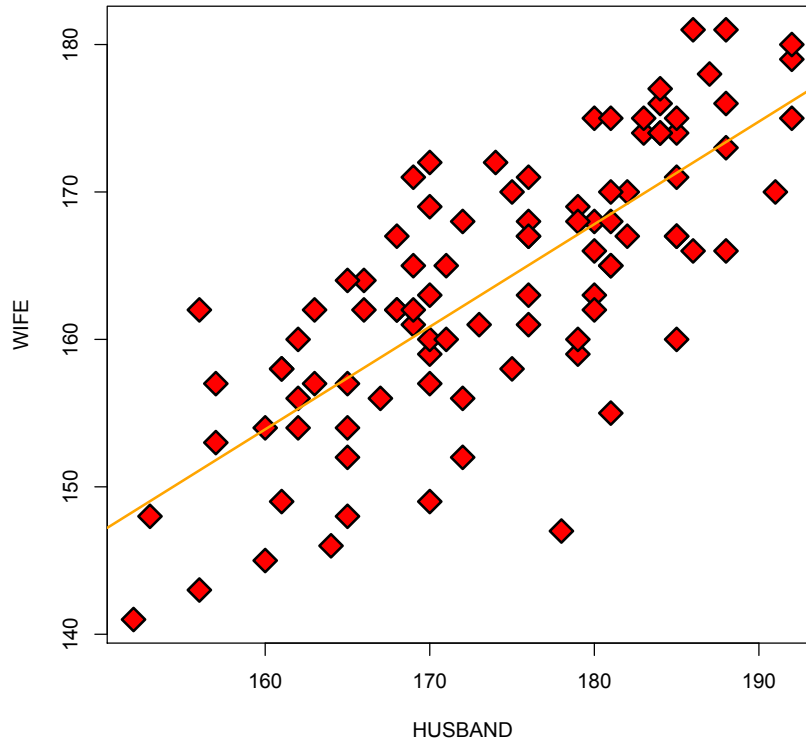
Sometimes use θ instead of W

$$y = f(x_0, x_1) = \sum_{i=1}^{n} \theta_i x_i = \theta^T X$$

Linear Regression in R

The following code is available at
http://www-stat.stanford.edu/~jtaylo/courses/stats203/R/introduction/introduction.R.html

```
heights.table <- read.table('http://www-stat.stanford.edu/~jtaylo/courses/stats203/data/heights.table', header=T, sep=',')
attach(heights.table)

# wife's height vs. husband's height
plot(heights.table, pch=23, bg='red', cex=2, lwd=2)

# Fit model

wife.lm <- lm(WIFE ~ HUSBAND)
print(summary(wife.lm))

# with fitted line
plot(heights.table, pch=23, bg='red', cex=2, lwd=2)
abline(wife.lm$coef, lwd=2, col='orange')
```



For more background see:
http://en.wikipedia.org/wiki/Linear_regression

R Example: Simple Linear Regression

- ### Download the data and tell R where to find the variables by attaching it

```
heights.table <- read.table('http://www-stat.stanford.edu/~jtaylo/courses/stats203/data/heights.table',
header=T, sep=',')
attach(heights.table)

# wife's height vs. husband's height
plot(heights.table, pch=23, bg='red', cex=2, lwd=2)

# Fit model
wife.lm <- lm(WIFE ~ HUSBAND)
print(summary(wife.lm))

# with fitted line
plot(heights.table, pch=23, bg='red', cex=2, lwd=2)
abline(wife.lm$coef, lwd=2, col='orange')

### Some other aspects of R

# Take a look at the variable names
names(heights.table)
# Estimate beta.1 using S_xx and S_yx

num <- cov(HUSBAND, WIFE) # = S_xx / (n-1)
den <- var(HUSBAND) # = S_yx / (n-1)
print(num/den)
# Get predicted values (Y.hat)

wife.hat <- predict(wife.lm)
# Two different ways of getting residuals
wife.resid1 <- WIFE - predict(wife.lm)
wife.resid2 <- resid(wife.lm)

# Computing sample variance by hand
```

http://www-stat.stanford.edu/~jtaylo/courses/stats203/R/introduction/introduction.R.html

Residuals

- # Get predicted values (Y.hat)

  wife.hat <- predict(wife.lm)

  # Two different ways of getting residuals

  wife.resid1 <- WIFE - predict(wife.lm)
  wife.resid2 <- resid(wife.lm)

  # Computing sample variance by hand

  husband.var <- sum((HUSBAND - mean(HUSBAND))^2) / (length(HUSBAND) - 1)
  print(c(var(HUSBAND), husband.var))

  # Estimating sigma.sq

  S2 <- sum(resid(wife.lm)^2) / wife.lm$df
  print(sqrt(S2))
  print(sqrt(sum(resid(wife.lm)^2) / (length(WIFE) - 2)))
  print(summary(wife.lm)$sigma)

  # What else is in summary(wife.lm)?

  print(names(summary(wife.lm)))

Linear Regression in R : WWW

- R Homepage
- R Download Page
- Using R in Statistics

- **Dataframes, distributions etc. in R**

- http://msenux.redwoods.edu/math/R/

- Linear Regression in R

R

- The S statistical programming language and computing environment has become the de-facto standard among machine learners, statisticians, operation research (kitchen sink, gateway).
- The S language has two implementations: the commercial product S-PLUS, and the free, open-source R.
- Both are available for Windows and Unix/Linux systems; R, in addition, runs on Macintoshes.
- This lecture series will use R.

R: A History (from 1997 –

- In computing, R is a programming language and software environment for general purpose statistical and analytics computing and graphics.
- It is an implementation of the S programming language with lexical scoping semantics inspired by Scheme.
- R was created by Ross Ihaka and Robert Gentleman[2] at the University of Auckland, New Zealand, and is now developed by the *R Development Core Team*.
- It is named partly after the first names of the first two R authors (Robert Gentleman and Ross Ihaka), and partly as a play on the name of S.[3]
- The R language has become a de facto standard among statisticians/engineers for the development of statistical and engineering software, and is widely used for statistical software development and data analysis.

[Wikipedia]

Scripting languages

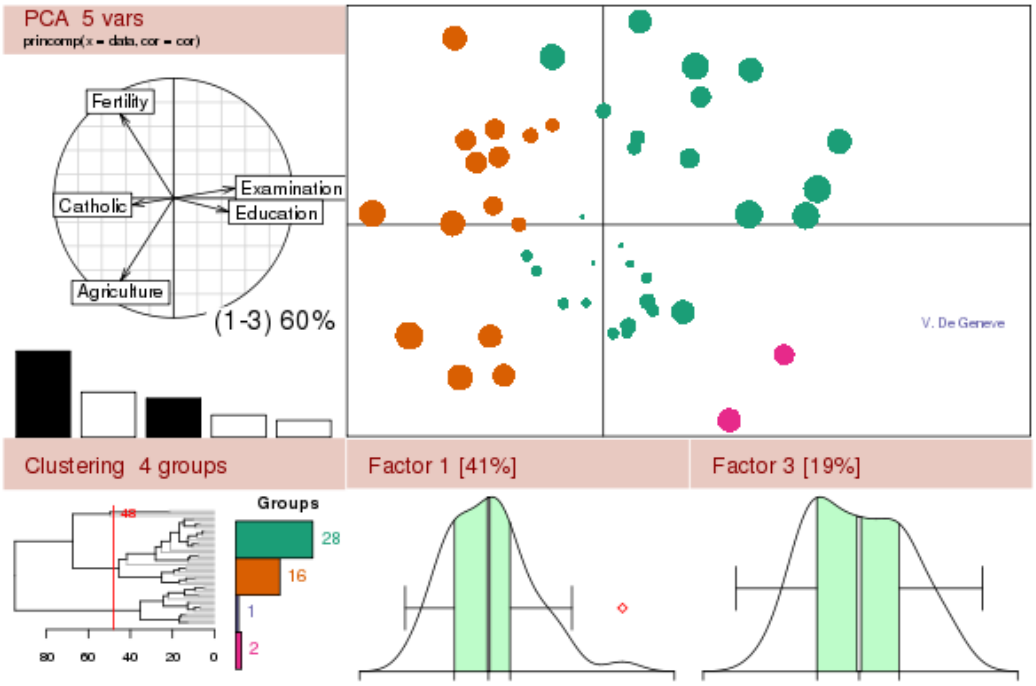
- **R has its own language**
  - R functionality has been made accessible from several scripting languages. E.g.,
    - Python (by the RPy[17] interface package)
    - Perl (by the Statistics::R[18] module).
- **Packages:**
  - Optimization packages are available
  - It can also be used as a general matrix calculation toolbox with comparable benchmark results to GNU Octave and its proprietary counterpart, MATLAB
  - An RWeka[9] interface has been added to the popular data mining software Weka which allows the capability to read/ write into the arff data format thus allowing the usage of data mining capabilities in Weka and statistical in R.

Software and Licenses

- **Available on Windows/Linux/Mac**
- **R is part of the GNU project.**
  - Its source code is freely available under the GNU General Public License, and pre-compiled binary versions are provided for various operating systems.
- **R uses a command line interface, though several graphical user interfaces are available.**

R

- **Intro R website**
  - http://cran.r-project.org/doc/manuals/R-intro.html#Graphics
- **Nice examples**
  - http://www.mayin.org/ajayshah/KB/R/index.html



Online Resources

- **R Site with examples (French, Naïve Bayes)**
  - http://zoonek2.free.fr/UNIX/48_R/12.html#2
- **Intro R website**
  - http://cran.r-project.org/doc/manuals/R-intro.html#Graphics
- **Nice examples**
  - http://www.mayin.org/ajayshah/KB/R/index.html
- **Steward Book website**
  - http://www.stewartcalculus.com/media/9_inside_chapters.php?subaction=showfull&id=1090822711&archive=&start_from=&ucat=2&show_cat=2
- **Taylor's page at Stanford**
  - http://www-stat.stanford.edu/~jtaylo//
- **Contour plots**
  - http://online.redwoods.cc.ca.us/instruct/darnold/MULTCALC/grad/grad.pdf
- **Fox's Book**
  - 2009, http://socserv.socsci.mcmaster.ca/jfox/Courses/R-programming/index.html
  - 2008, http://socserv.mcmaster.ca/jfox/Courses/R-course/index.html

Online Resources

- **R** http://www.r-project.org/
- **R books online**
  - http://www.math.ccu.edu.tw/~yshih/Rrefs/Rlecturenotes.pdf
  - http://www.cran.r-project.org/doc/contrib/Faraway-PRA.pdf
- **STATISTICS: AN INTRODUCTION USING R (Crawley)**
  - http://www.bio.ic.ac.uk/research/crawley/statistics/exercises.htm
- **Resources at Stanford**
  - http://www-stat.stanford.edu/~jtaylo/courses/stats191/R/logistic/flu.R
  - http://www-stat.stanford.edu/~jtaylo/courses/stats191/R/logistic/fluout.html

Installing R and an Editor

- **Rstudio http://www.rstudio.org/ (self contained environment)**
- **OR do the following:**
- **Installing an editor: EditPlus (for Windows)**
  - Useful Editor on Windows (30 temporary license)
  - http://www.brothersoft.com/download-editplus-16751.html
  - Eclipse
- **Installing R (Windows, also on Linux and Mac)**
  - Click here to download an installer EXE: http://cran.r-project.org/bin/windows/base/R-2.10.0-win32.exe

    The distribution is distributed as a 30Mb installer R-2.10.0-win32.exe.

    Just run this for a Windows-XP style installer. It contains all the R components, and you can select what want installed.

    For more details, including command-line options for the installer and how to uninstall, see the rw-FAQ (http://cran.r-project.org/bin/windows/base/rw-FAQ.html).

RStudio

http://www.rstudio.org/