Quantitative Cyber-Security

Colorado State University
Yashwant K Malaiya

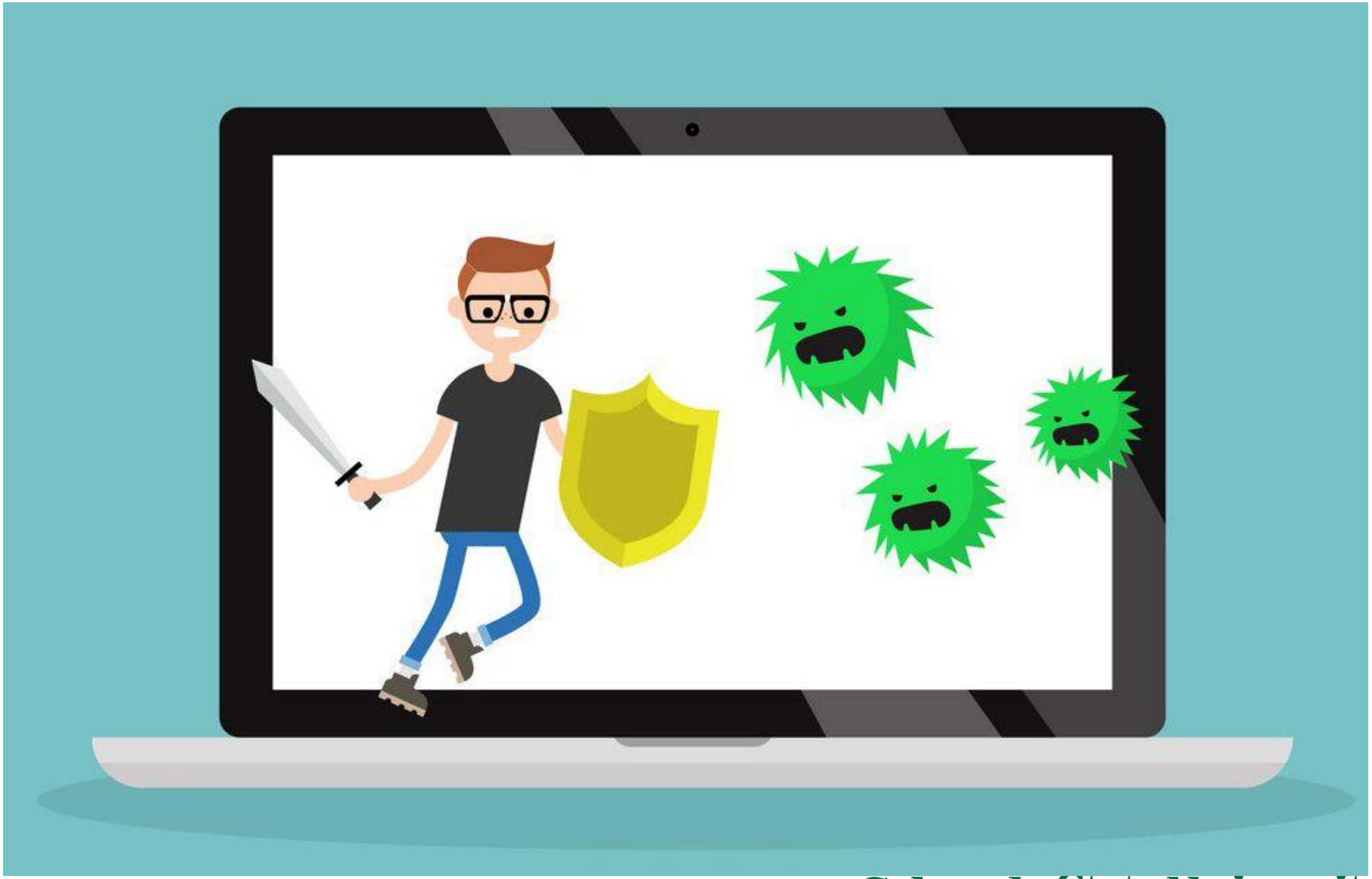
CS559

Intrusion Detection, Modeling, Regression



CSU Cybersecurity Center
Computer Science Dept

Intrusion Detection



Intrusion Detection

- Intrusion: Unauthorized act of bypassing the security mechanisms of a system.
- Intrusion Detection System (IDS): A software/hardware system that gathers and analyzes information to identify possible intrusions
  - from various areas within a computer (Host-based) **HIDS**
    - Monitors the characteristics of a single host for suspicious activity
  - Traffic on a a network (network based) **NIDS**
    - Monitors network traffic and analyzes network, transport, and application protocols to identify suspicious activity
  - Hybrid
- IDS components:
  - "Sensors" - collect data
  - Analyzers - determine if intrusion has occurred
  - User interface - view output or control system behavior

IDS Detection Approaches

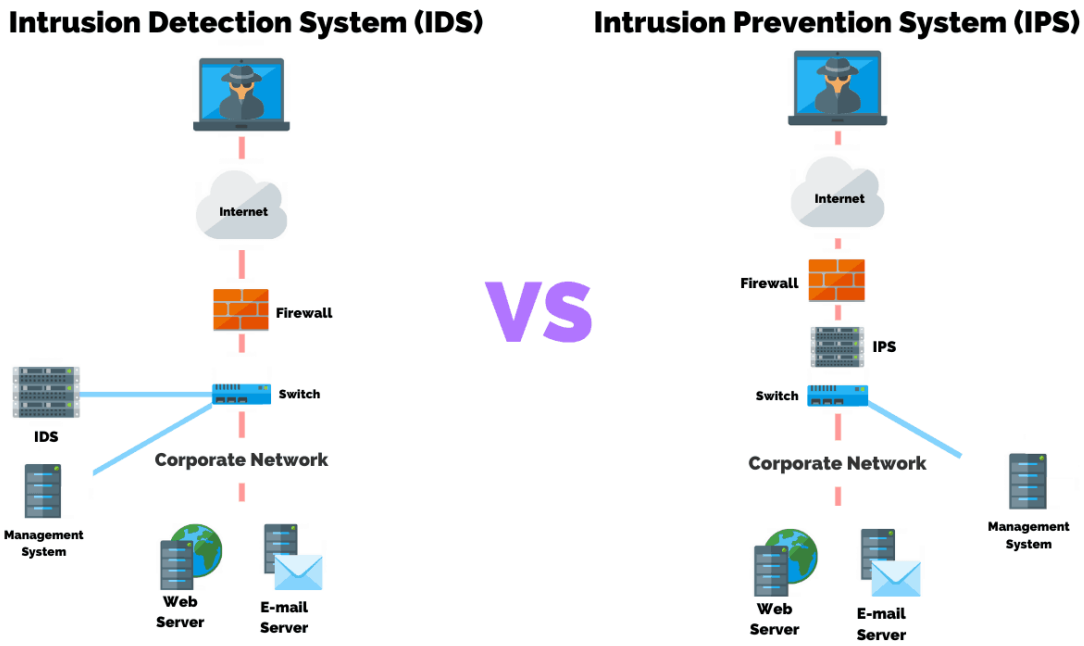
Approaches

- **Signature based detection**: Does it match known bad behavior?
  - Match a large collection of known patterns of malicious data against data on a system or in transit over a network
  - Low false positives but Ineffective for a new exploit.
- **Anomaly detection**: Is this the normal behavior?
  - Collection of data about the behavior of legitimate users, use AI
  - Does the current behavior resemble that of a legitimate user?
- Reputation-based :
  - blocks traffic from IP addresses/domains associated with malicious or suspicious activity.
- Stateful protocol analysis:
  - protocol behavior, for example, a denial-of-service (DoS) attack by detecting a single IP address making many simultaneous TCP connection requests in a short period.

IDS and IPS, Firewals

- IPS (intrusion prevention systems)
  - monitors network traffic for suspicious activity, like an IDS,
  - *intercepts* threats in real time by automatically terminating connections or triggering other security tools.
  - Because IPSs are meant to stop cyberattacks, they're placed inline, all traffic has to pass through the IPS before it can reach the rest of the network.
  - Often **IDS and IPS are combined** in a single intrusion detection and prevention system (IDPS)
- Firewalls face outside the network and act as barriers by using predefined rulesets to allow or disallow traffic.
- IDSs often sit near firewalls and help catch anything that slips past them.
- Some firewalls, especially next-generation firewalls, have built-in IDS and IPS functions.

IDS vs IPS



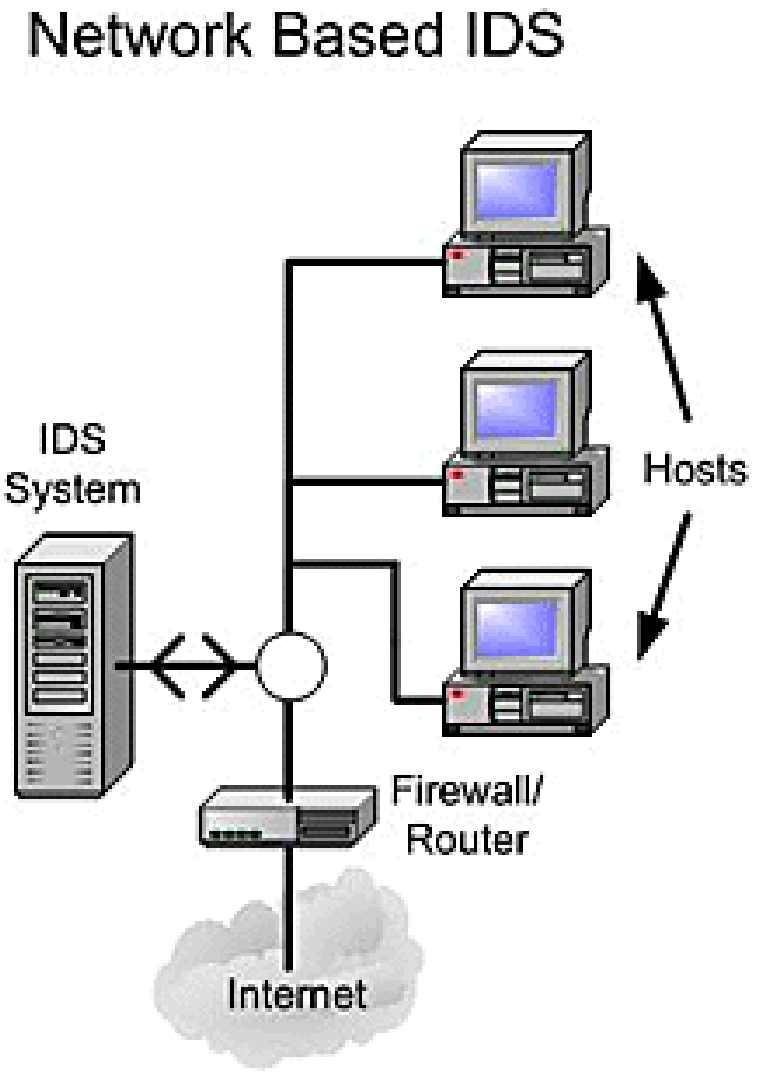
The IDS is designed to **detect** anomalous activity by observing the network traffic patterns. If detected, it generates an alert. The IDS is not placed directly in line with the network traffic.

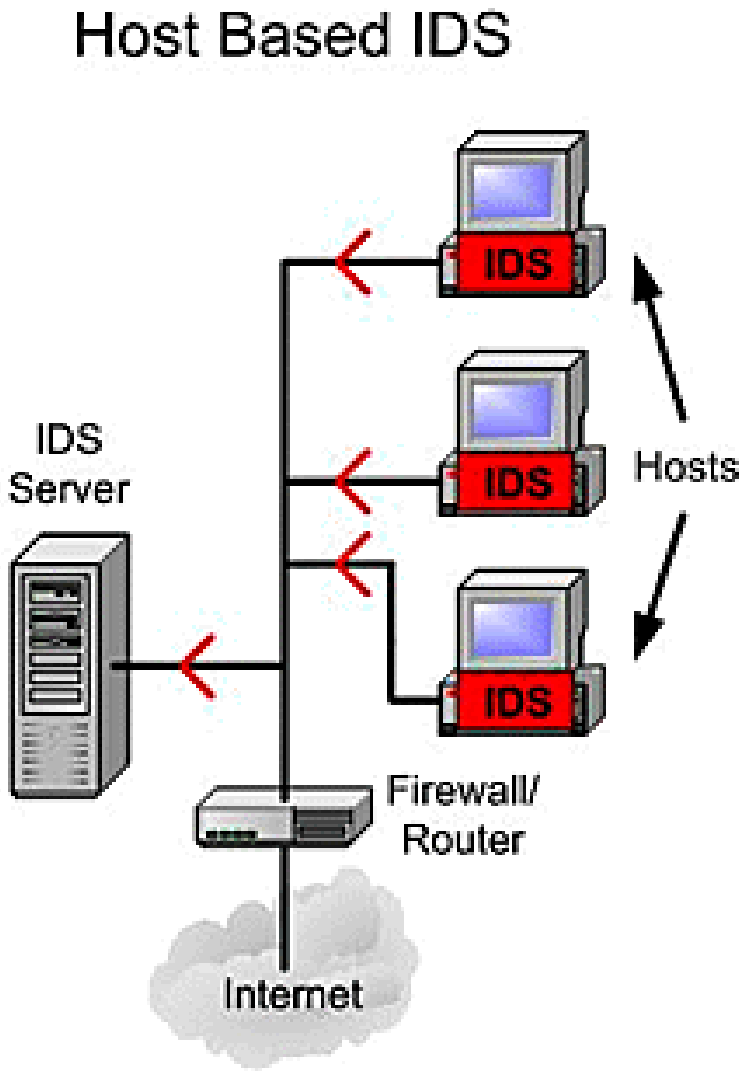
An IPS examines network traffic flows in order to **detect and prevent** vulnerability exploits. The IPS often sits directly behind the firewall.

https://purplesec.us/intrusion-detection-vs-intrusion-prevention-systems/

Colorado State University

Network based vs Host based IDS





https://en.bmstu.wiki/index.php?title=HIDS_(Host-Based_Intrusion_Detection_System)&mobileaction=toggle_view_mobile

IDS Examples

- Antivirus: looks for signatures of known threats
- SNORT: a multi-mode packet analysis tool
  - Sniffer, Packet Logger, Forensic Data Analysis too, Network Intrusion Detection System
  - Rules form "signatures"
    - Modular detection elements are combined to form these signatures
    - Wide range of detection capabilities
      - Stealth scans, OS fingerprinting, buffer overflows, back doors, CGI exploits, etc.
    - Rules system is very flexible, and creation of new rules is relatively simple.
    - bad-traffic.rules, exploit.rules, scan.rules, smtp.rules, smtp.rules, backdoor.rules shellcode.rules ….
  - SnortML uses machine learning (March 2024)
  - Incorporated into Cisco Secure Firewall (Dec 2024)

Example study

Performance comparison of intrusion detection systems and application of machine learning to Snort system

- Shah and Isaac, 2017
- Two open source NIDS Snort and Suricata compared, with specific algorithms
- Normal and malicious traffic, different protocols
- Positive = TP+FN, Negative = FP+TN
- FPR = FP/(FP+TN), FNR = FN/(FN+TP)

	Disease +	Disease -
Test +ve	TP	FP
Test –ve	FN	TN

Malicious traffic accuracy (%) measurements at 10 Gbps during 12 h.

Malicious Traffic	First 4 h			
	Snort FPR	Snort FNR	Suricata FPR	Suricata FNR
SSH	8.0	0.0	7.0	0.0
DoS/DDoS	3.0	2.0	10.0	0.0
FTP	12.0	0.0	11.0	0.0
HTTP	5.0	0.0	8.0	0.0
ICMP	20.0	0.0	22.0	0.0
ARP	7.0	0.0	10.0	12.0
Scan	1.0	4.0	5.0	3.0
Total	56.0	6.0	73.0	15.0

Industry Comparison Matrix

Intrusion Detection System	Primary Type	Average True Positive Rate (TPR)	Average False Negative Rate (FNR)		Main Operational Constraint
Snort (Cisco)	Network (NIDS)	98.50%	1.50%	Open source, widely used	Packet dropping under extreme traffic spikes.
Suricata (OISF)	Network (NIDS)	97.40%	2.60%	Open source, widely used	Demands higher baseline memory and CPU resources.
Zeek	Network Behavioral	96.40%	3.60%	Open source	Requires custom script tailoring to prevent alert fatigue.
OSSEC	Host (HIDS)	98.00%	2.00%	Open Source	Limited visibility over broad raw network traffic lanes.
Commercial NGFWs (Check Point, Palo Alto)	Hybrid / Integrated	99.70%	0.30%	65% to 70% of total enterprise market spending	High license cost and cloud ecosystem lock-in.

Based on multiple sources

Quantitative Security

Colorado State University
Yashwant K Malaiya

CS 599
Modeling and Regression



CSU CyberCenter
***Course Funding Program* – 2019**

Modeling & Regression

Model vs Reality

- Reality is complex, often too complex
- A model represents the actual behavior of interest with acceptable accuracy
  - System design & evaluation of alternatives
  - System simulation
  - Predictions
- When there are multiple models, how do you choose one?
  - Familiarity, convenience?
  - Goodness of fit?
  - Predictive capabilities?

Quantitative models

- **Derived from first principles**:
  - Arguments are actual things measured
  - Example: monthly mortgage payment

$$c = \frac{rP}{1-(1+r)^{-N}}$$

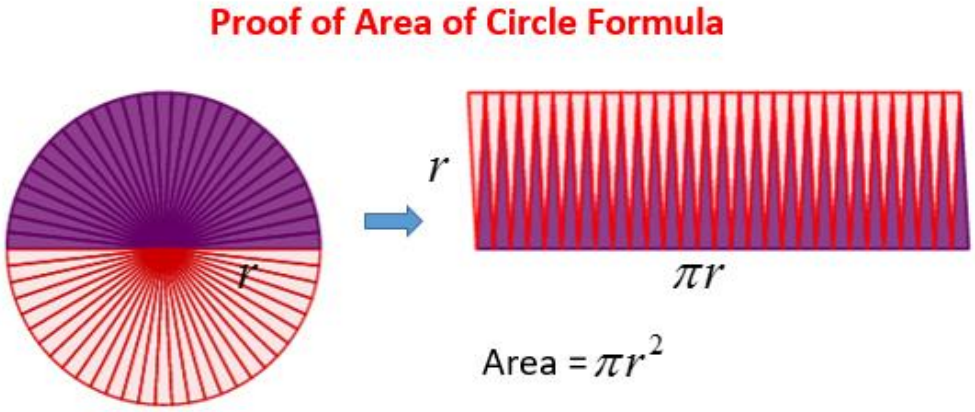
- **Empirical**
  - Arguments are just parameters
  - Example Ideal body weight in kg = a+b (height in in. - c)

    Men: a=52 kg, b= 1.9, c = 60 in.
- **Combined**
  - Empirical with interpretation of parameters
  - From first principles, adjusted to fit

14

Derived using first principles

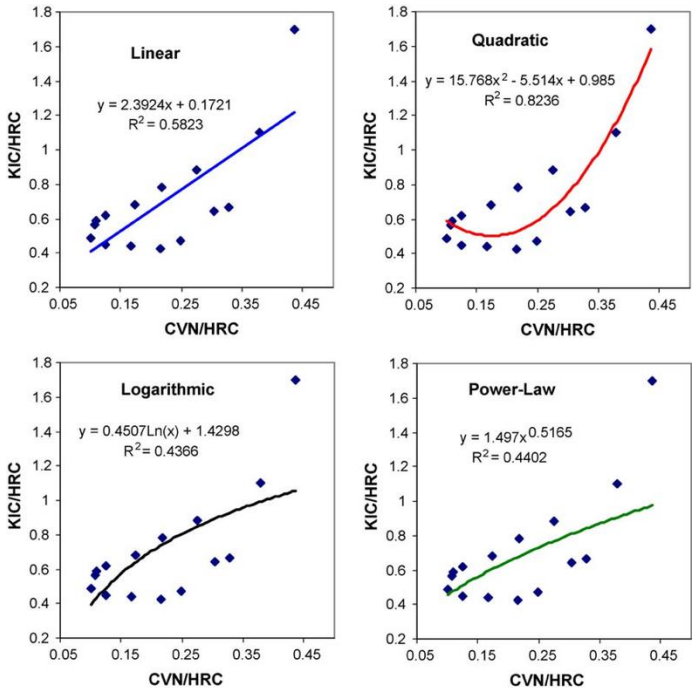
- Obtain model
  - Understand how things work
  - Use approximations
  - Derive formulas
- Validate:
  - Get real data
  - See if it fits. If not,
    - Make adjustments
    - Get alternative models
- Use the model
  - Plug in values to estimate parameters
  - Make projections



15

Empirical models

- Look at data
- See if it resembles a function
  - Linear, quadratic, logarithmic, exponential..
  - Involving 1, 2 or more parameters
- See if it fits
  - If not try something more complex
- If it fits, see if an interpretation of the parameters is possible
  - Not necessary but will be good.



16

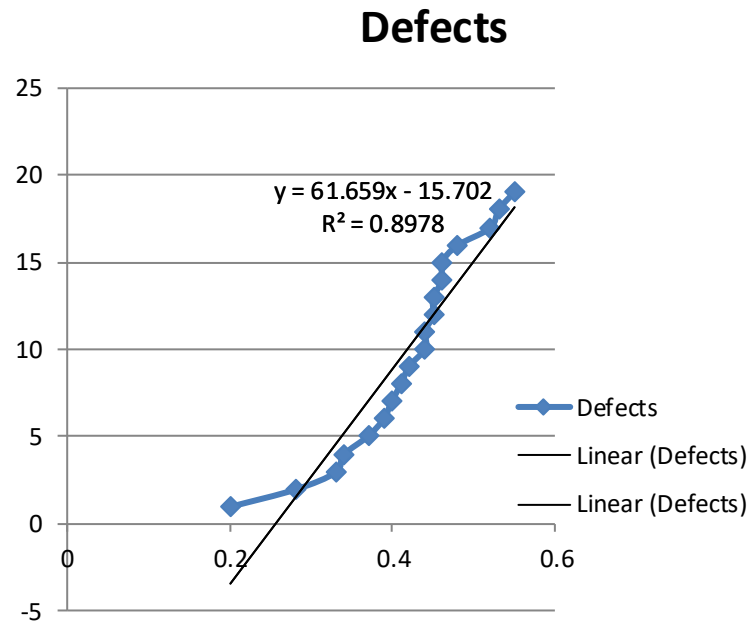
Curve fitting

- Get x-axis and y-axis numbers (x_i, y_i).
- Draw a scatter plot.
- Fit (i.e. estimate parameters):
  - Use formulas for parameters
  - linest( ) or logest( ) functions in excel
  - Select plot, rc, add trendline, select display options in excel
  - Use Solver for general fitting for excel
  - Goodness of fit measures: R^2
- Note: High goodness of fit does not imply high predictive capability

17

Example

Branch Cov	Defects
0.2	1
0.28	2
0.33	3
0.34	4
0.37	5
0.39	6
0.4	7
0.41	8
0.42	9
0.44	10
0.44	11
0.45	12
0.45	13
0.46	14
0.46	15
0.48	16
0.52	17
0.53	18
0.55	19



Open Excel sheet CurveFittingExamples.xlsx

18

Curve Fitting

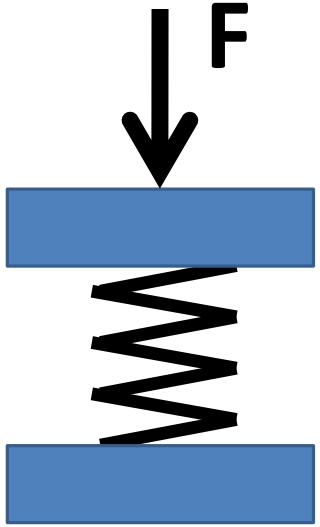
From

Engineering Computation: An Introduction Using MATLAB and Excel, Ch 5

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

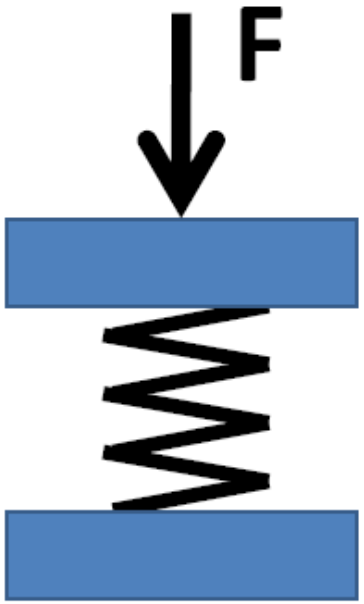
- Often, we have data points and we want to find an equation that “fits” the data
- Simplest equation is that of a straight line

- A spring is placed between two flat plates and force is slowly applied to the upper plate, causing the spring to compress. When the force reaches a pre-load value of 25 N, the location of the upper plate is recorded. As the upper plate continues to move, the distance that the top plate moves, *d*, and the force required to move the plate, *F*, are recorded. As soon as the upper plate has moved 10 mm, the test is ended.



Curve Fitting Example

- Data from the test is shown here. The slope of the load-displacement curve is called the spring constant.

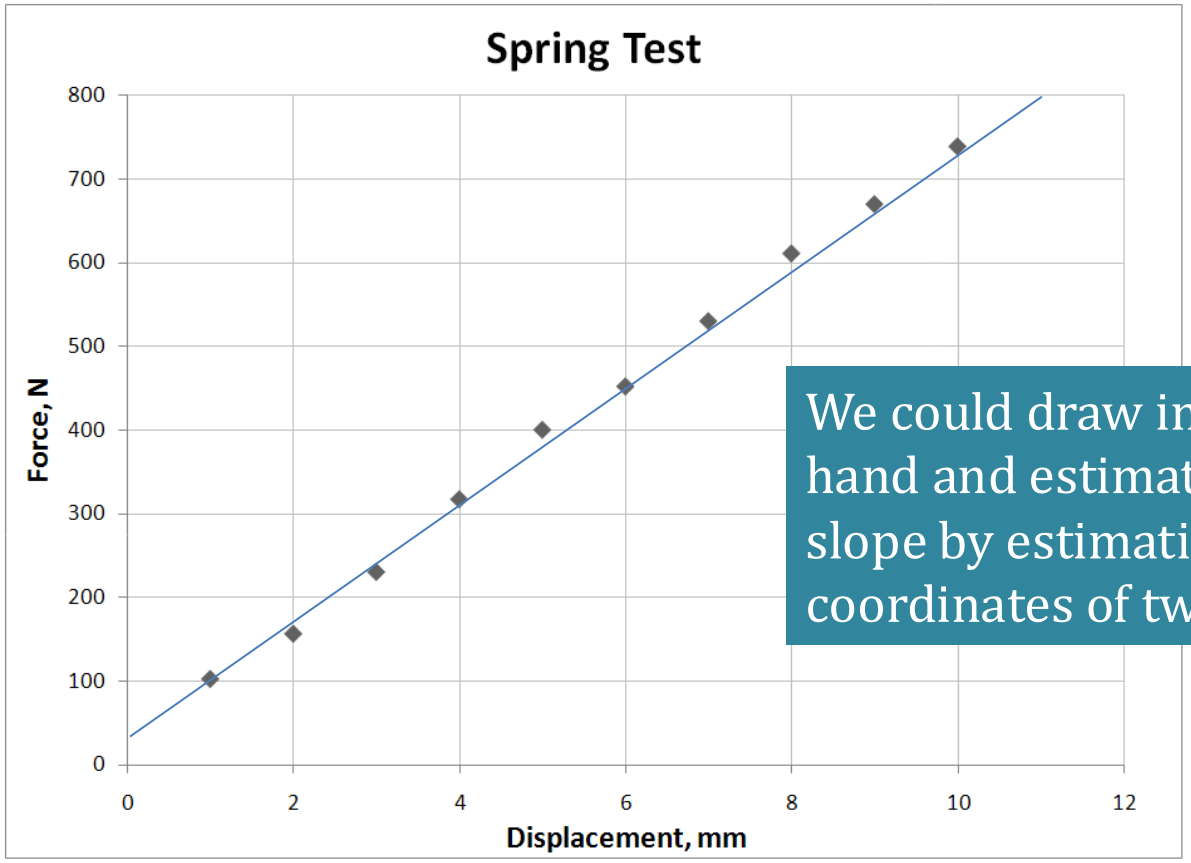


d, mm	F, N
1	102
2	156
3	230
4	317
5	400
6	452
7	530
8	611
9	670
10	739

Using Excel

- Try data in CurveFittingExampleswithSolver.xlsx

- Data from a test – how do we find the slope?



We could draw in a line by hand and estimate its slope by estimating the coordinates of two points

Curve Fitting Example

- How do we find the line that *best* fits the data?
- Find the equation of the line that minimizes the sum of the *squares* of the differences between the values predicted from the equation and the actual data values
- Why do we square the differences? Because we are interested in the magnitudes of the differences. If one point is above the line and other is below it, we don't want these difference to cancel each other

- In Excel, we can choose to have the equation of the best-fit line (the trendline) displayed in the form

$$y = mx + b$$

where m = the slope of the line

and b = the intercept of the line with the y-axis

$$y = 72.28x + 23.13$$

- In this example, the slope of the line (the stiffness of the spring) equals 72.3 N/mm
- When the deflection (x) equals zero, the force (y) equals 23.1 pounds. This is consistent with our nominal pre-load of 25 pounds

Correlation Coefficient

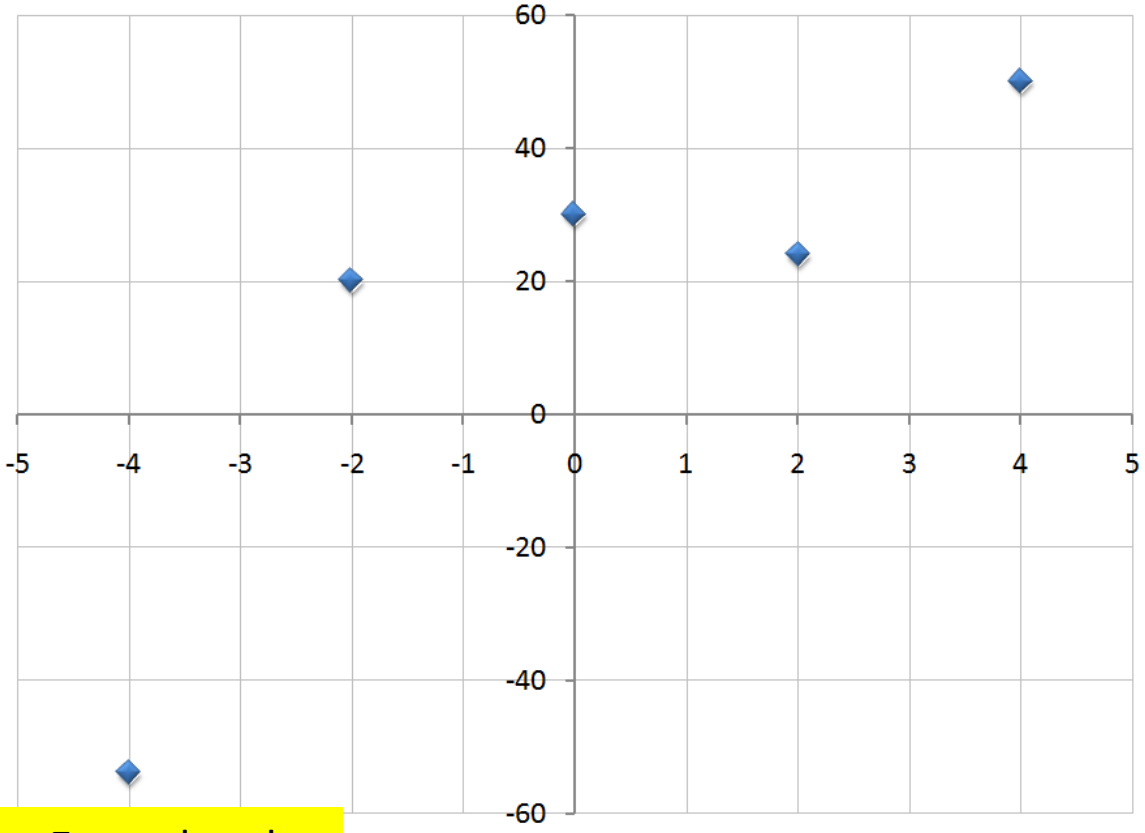
- The *correlation coefficient* (R^2) is a measure of how well the trendline fits the data
- A value of one represents a perfect fit
- In our example, the line fit is very good

$$y = 72.28x + 23.13$$
$$R^2 = 0.998$$

Curve Fitting Example

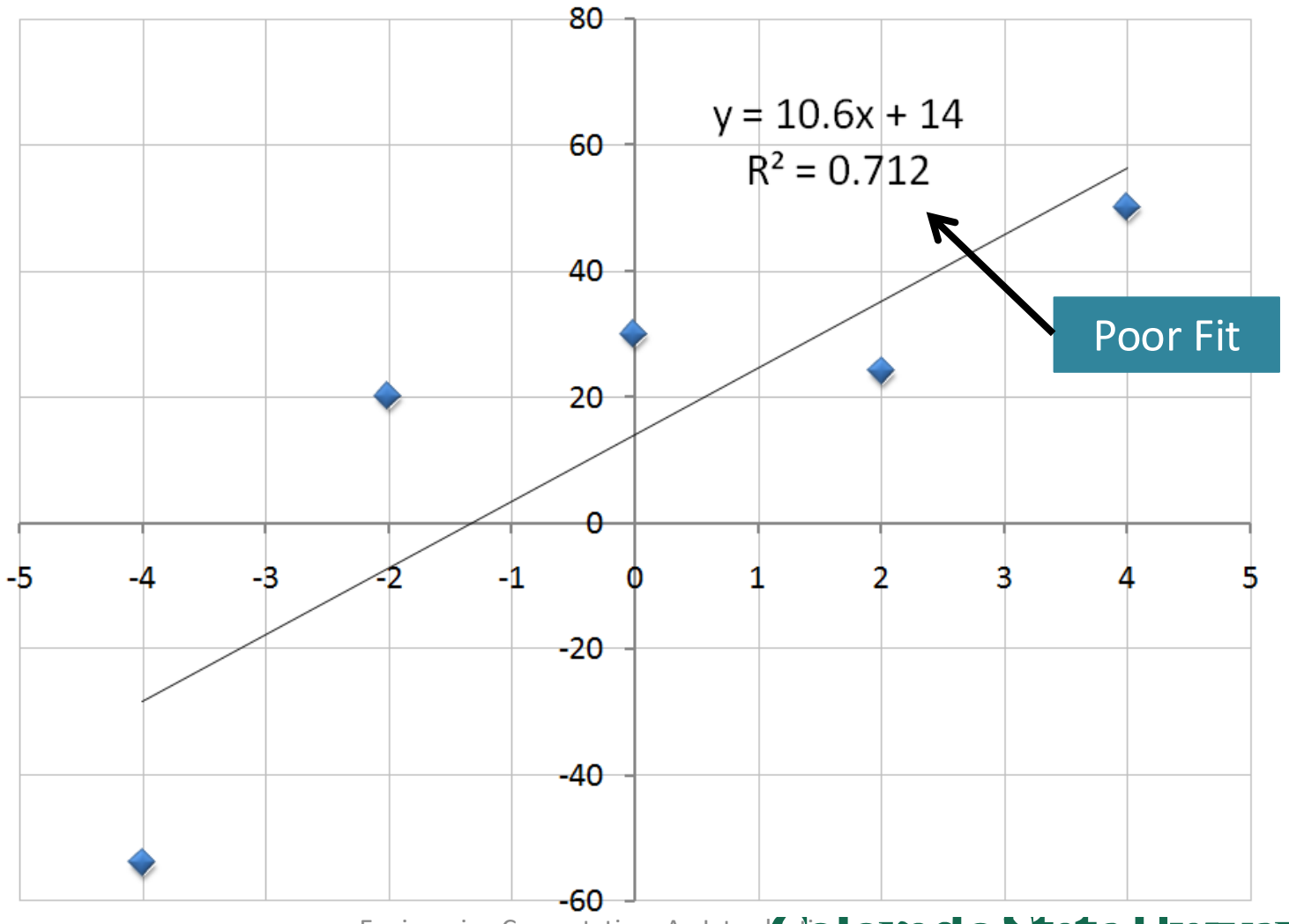
- Consider these five data points:

x	y
-4	-54
-2	20
0	30
2	24
4	50

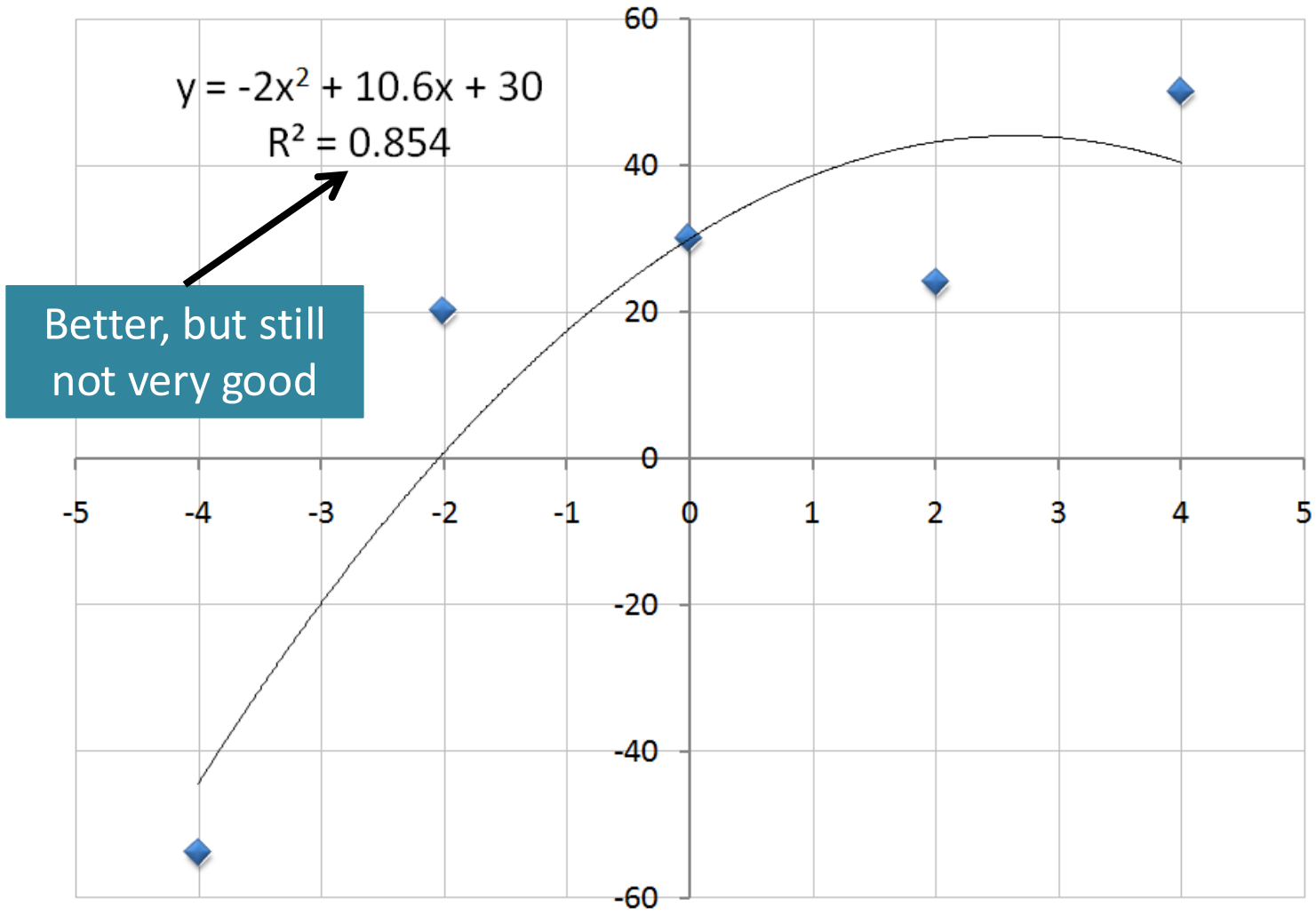


Open Excel sheet CurveFittingExamples.xlsx

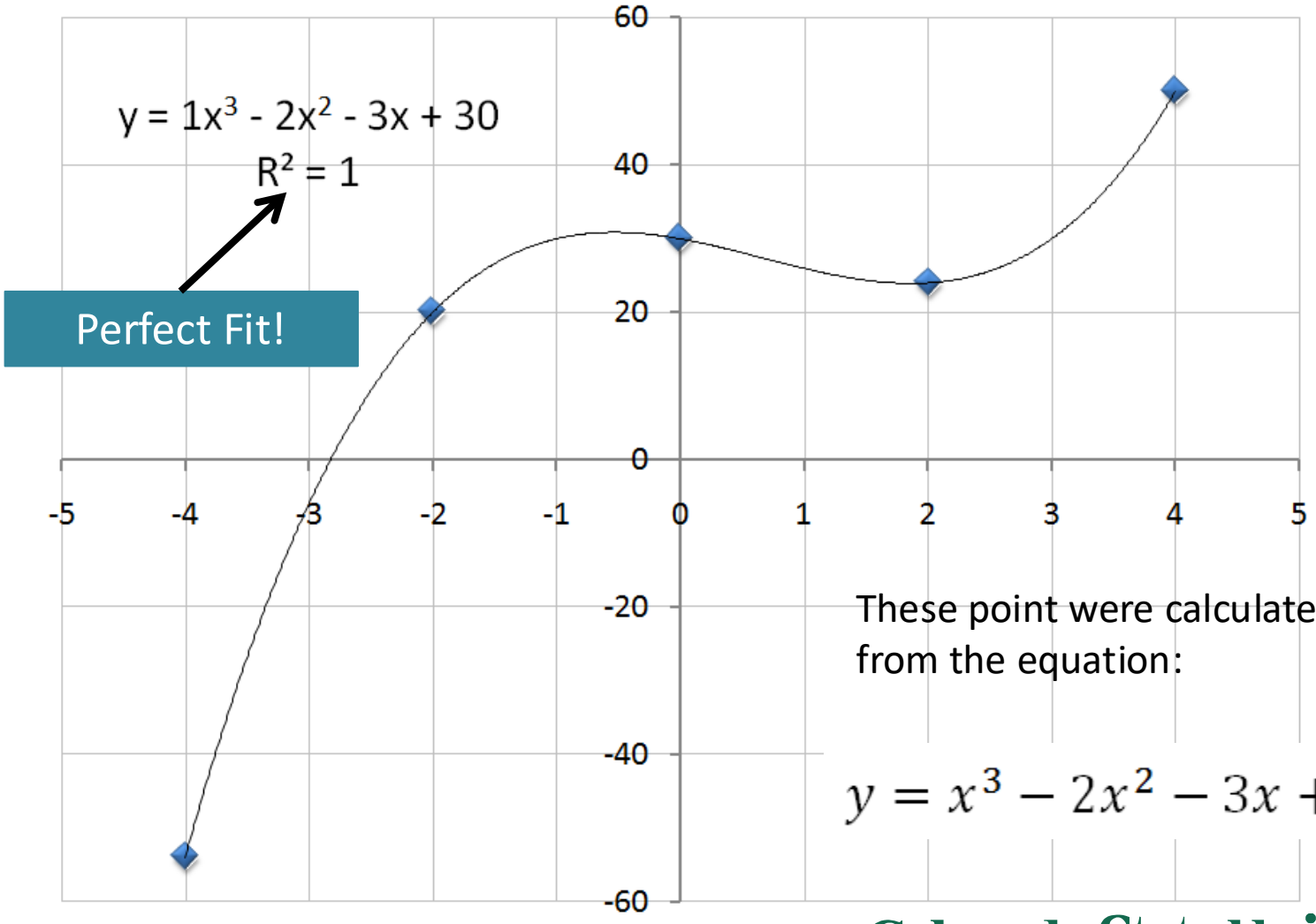
Linear Curve Fit



Try Second–Order Polynomial Fit



Third-Order Polynomial Fit



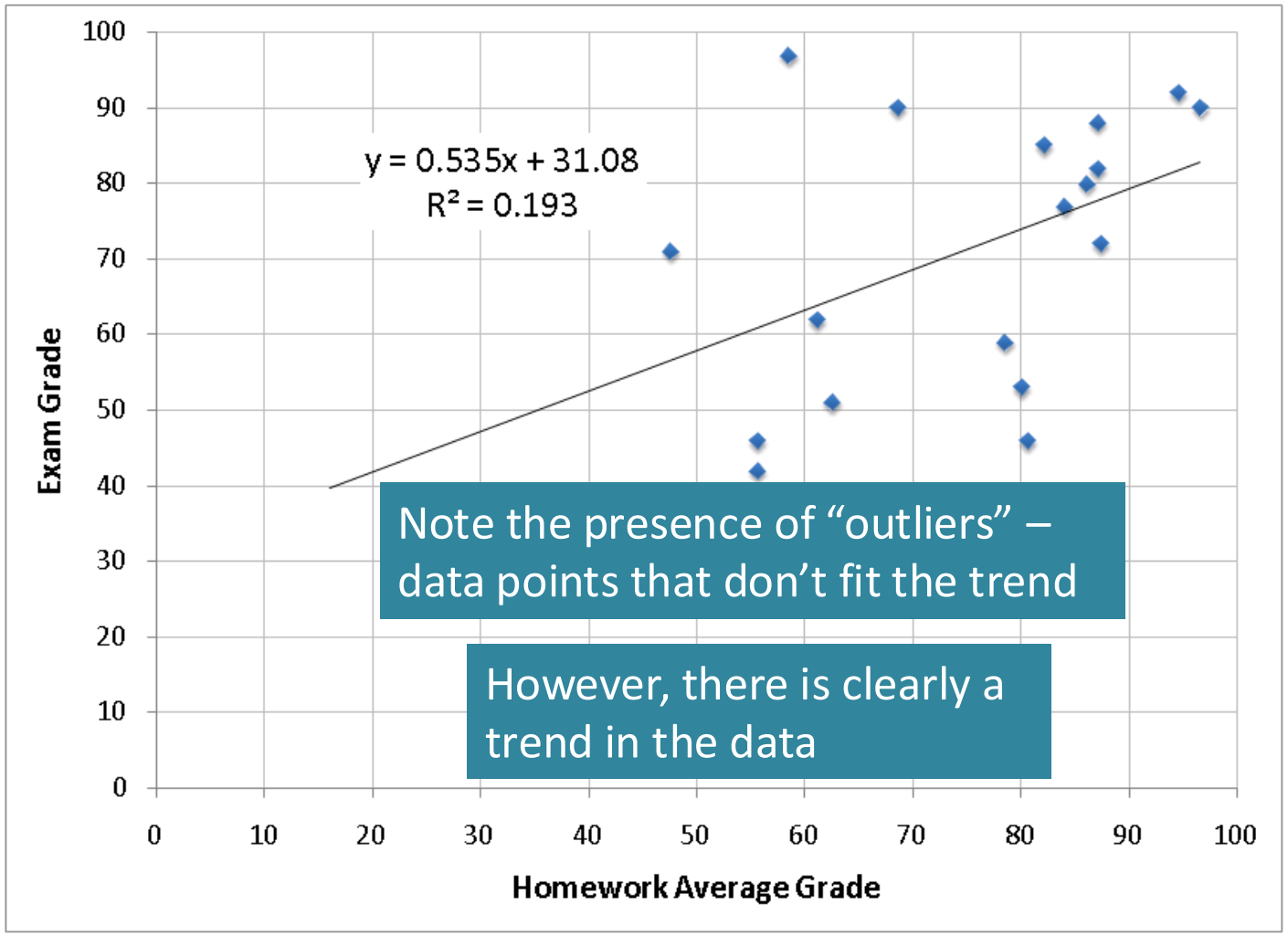
These point were calculated from the equation:

$$y = x^3 - 2x^2 - 3x + 30$$

Correlation Coefficient

- A “good” fit to data is relative
- In the case of the spring example, the data should fit a mathematical model, and so an R^2 value of close to one is expected
- For other cases, a much lower R^2 value is expected
- Consider a comparison of final exam scores in a class vs. homework averages
- We would expect that students who do well on HW will generally do well on the final exam, but there will be exceptions

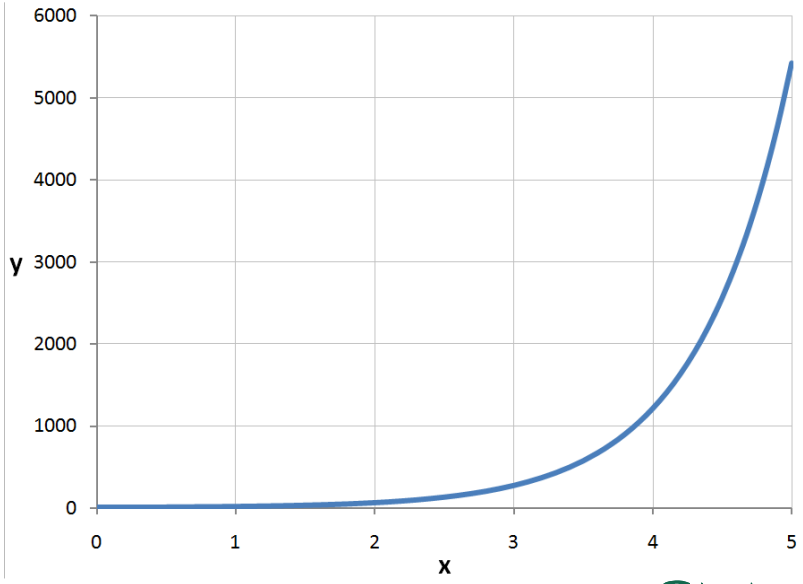
Exam vs. HW Example



Exponential and Power Equations

- Consider this equation:

- Here is a graph of the equation: $y = 3e^{1.5x}$



$$y = 3e^{1.5x}$$

$$\ln(y) = \ln(3e^{1.5x})$$

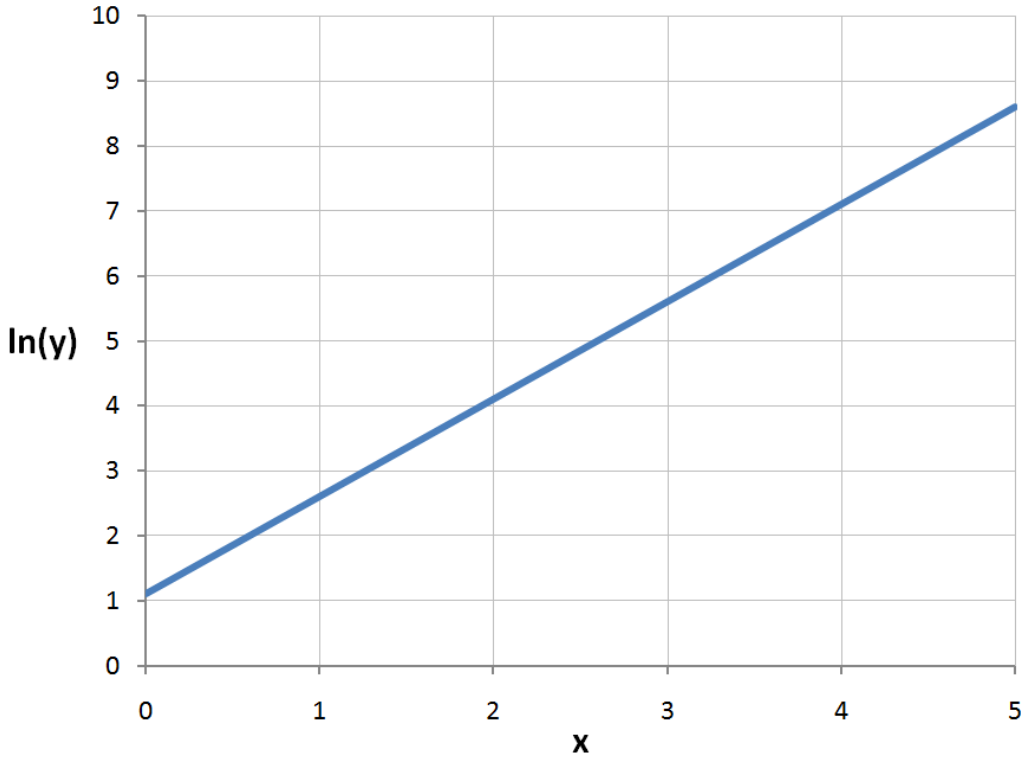
$$\ln(y) = \ln(3) + \ln(e^{1.5x})$$

$$\ln(y) = 1.5x + \ln(3)$$

If we plot the $\ln(y)$ instead of y, then we have the equation of a straight line

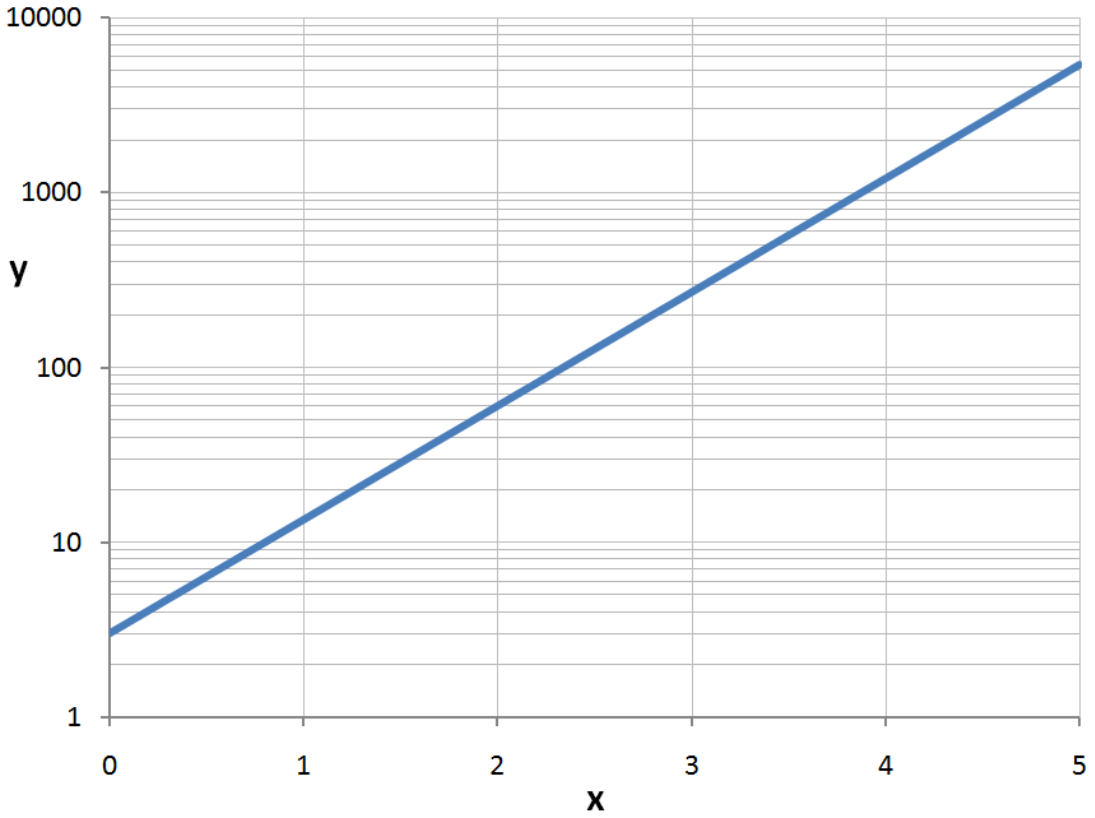
Exponential Equation

- This plot is not particularly useful, since it requires us to read the ln of the dependent variable *y*, instead of *y* itself



Exponential Equation

- A better way is to display the *y* values on a *logarithmic scale*:



Exponential Equation

- We call this a *semi-log* plot since one of the axes is logarithmic
- Note that we have used a base-10 scale: remember that a logarithm can be converted to a logarithm of another base by multiplying by a constant:

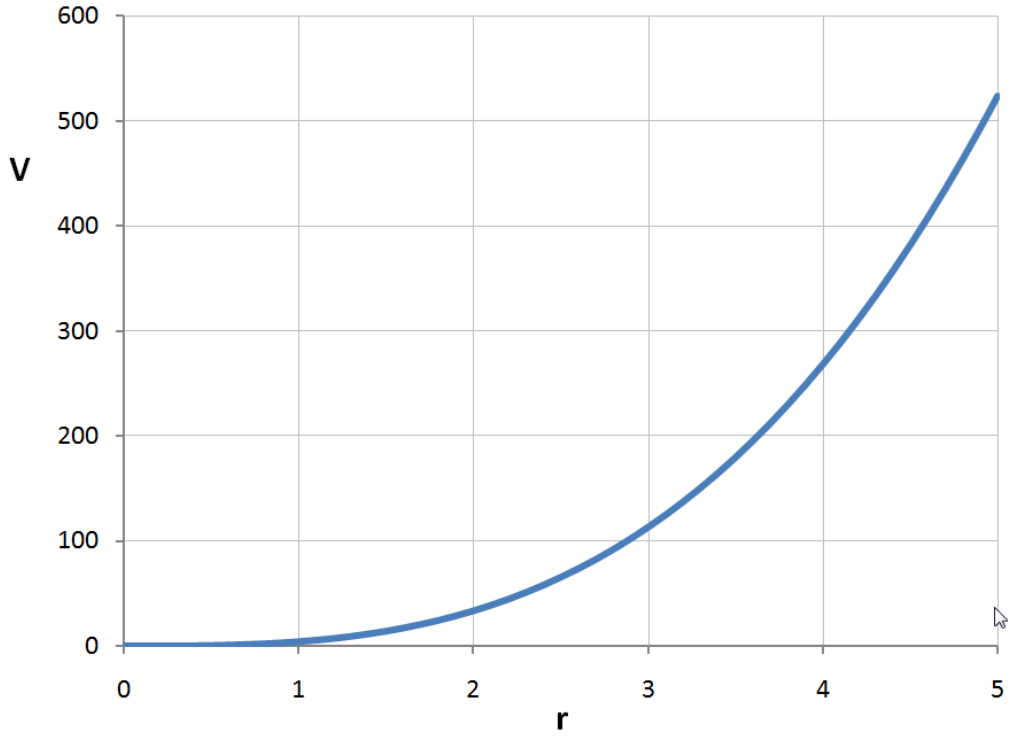
$$\log_a(x) = \log_b(x) * \log_a(b)$$

- Therefore, the equation will produce a straight line on semi-log axes regardless of the base of the logarithmic scale. Base 10 normally used – easiest to read
- Logarithmic scales can be specified in Excel and MATLAB

- Consider the equation for the volume of a sphere:

$$V = \frac{4}{3}\pi r^3$$

Plot:



Power Equations

$$V = \frac{4}{3}\pi r^3$$

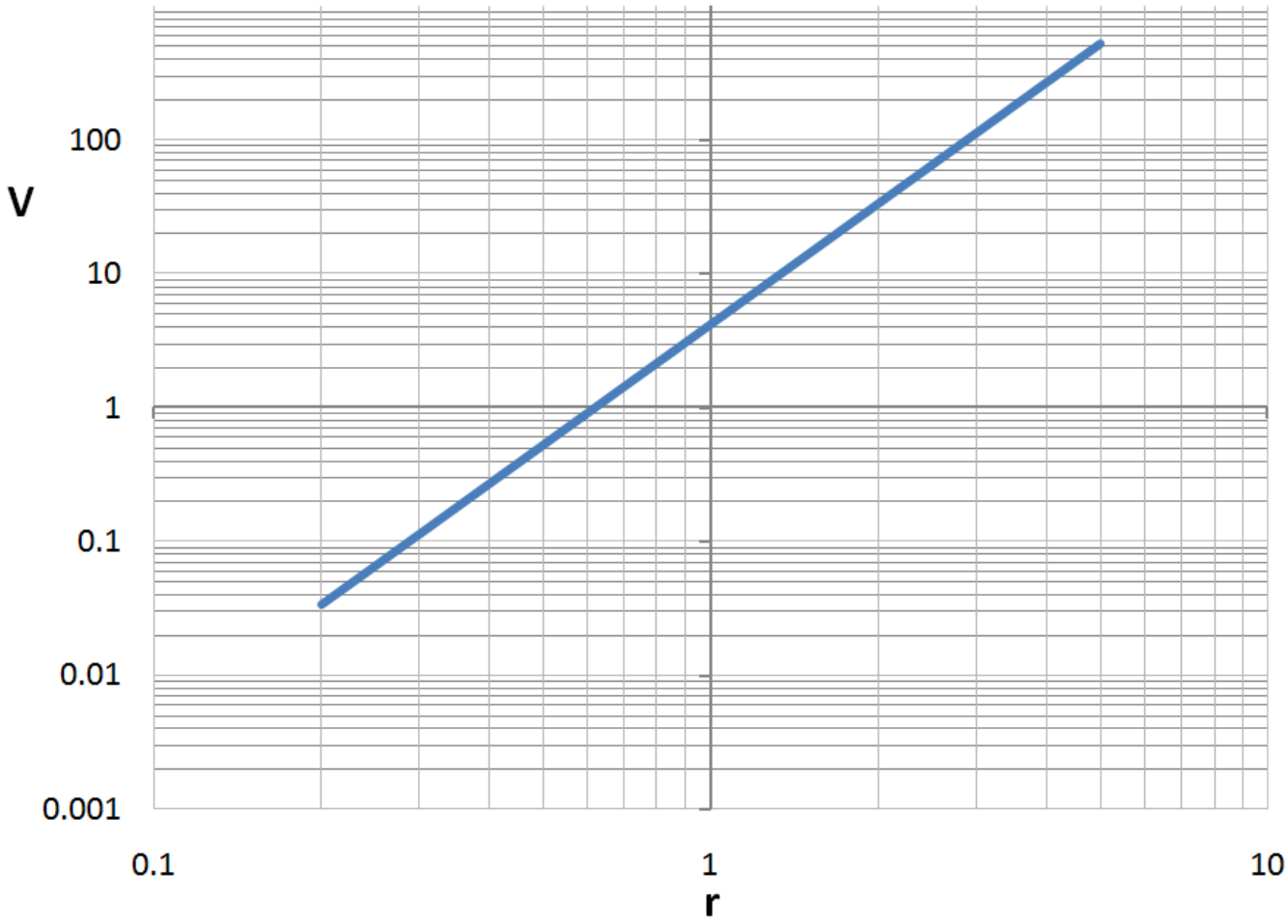
$$\log(V) = \log\left(\frac{4}{3}\pi r^3\right)$$

$$\log(v) = \log\left(\frac{4}{3}\pi\right) + \log(r^3)$$

$$\log(V) = 3 * \log(r) + \log\left(\frac{4}{3}\pi\right)$$

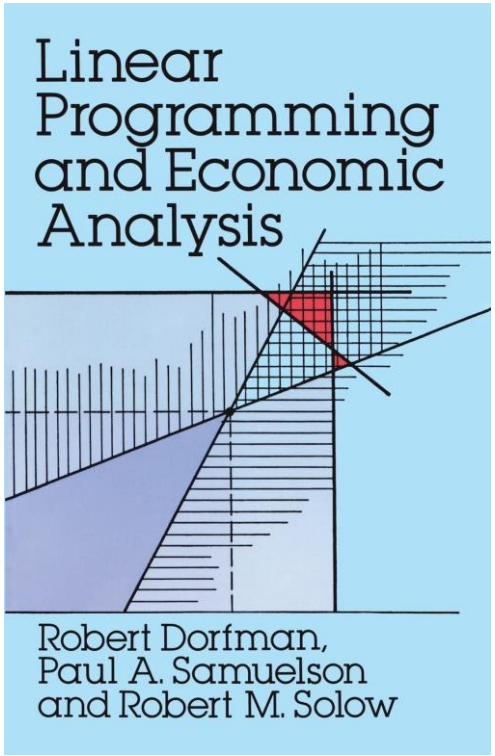
If we plot the log(y) *and* log(x), then we have the equation of a straight line

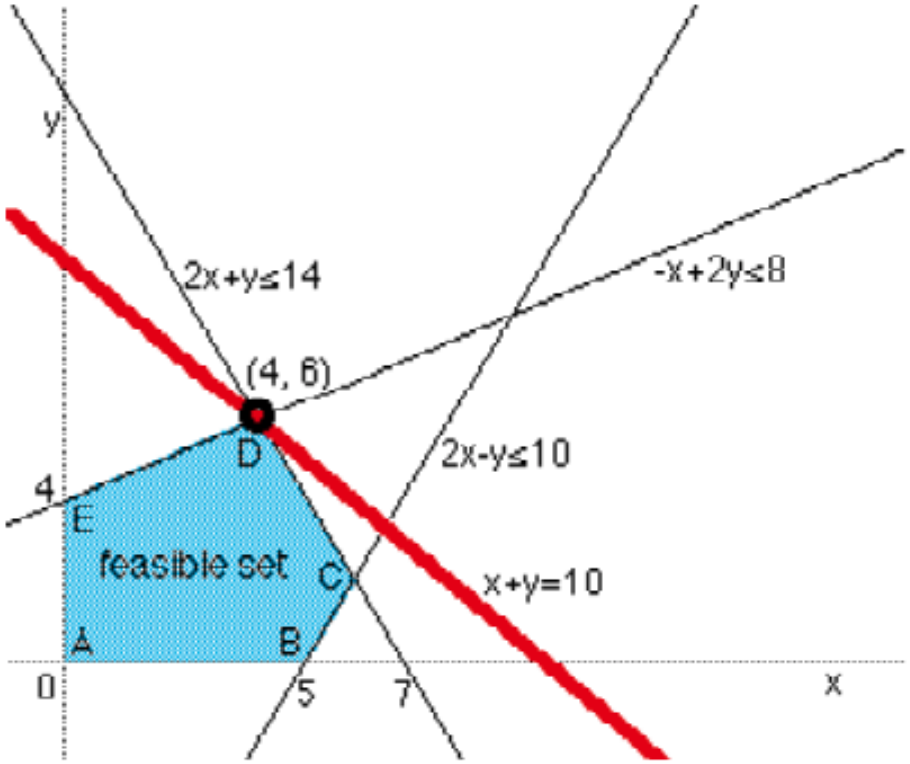
- This is called a *log-log* plot, since both axes are logarithmic



Optimization with constraints

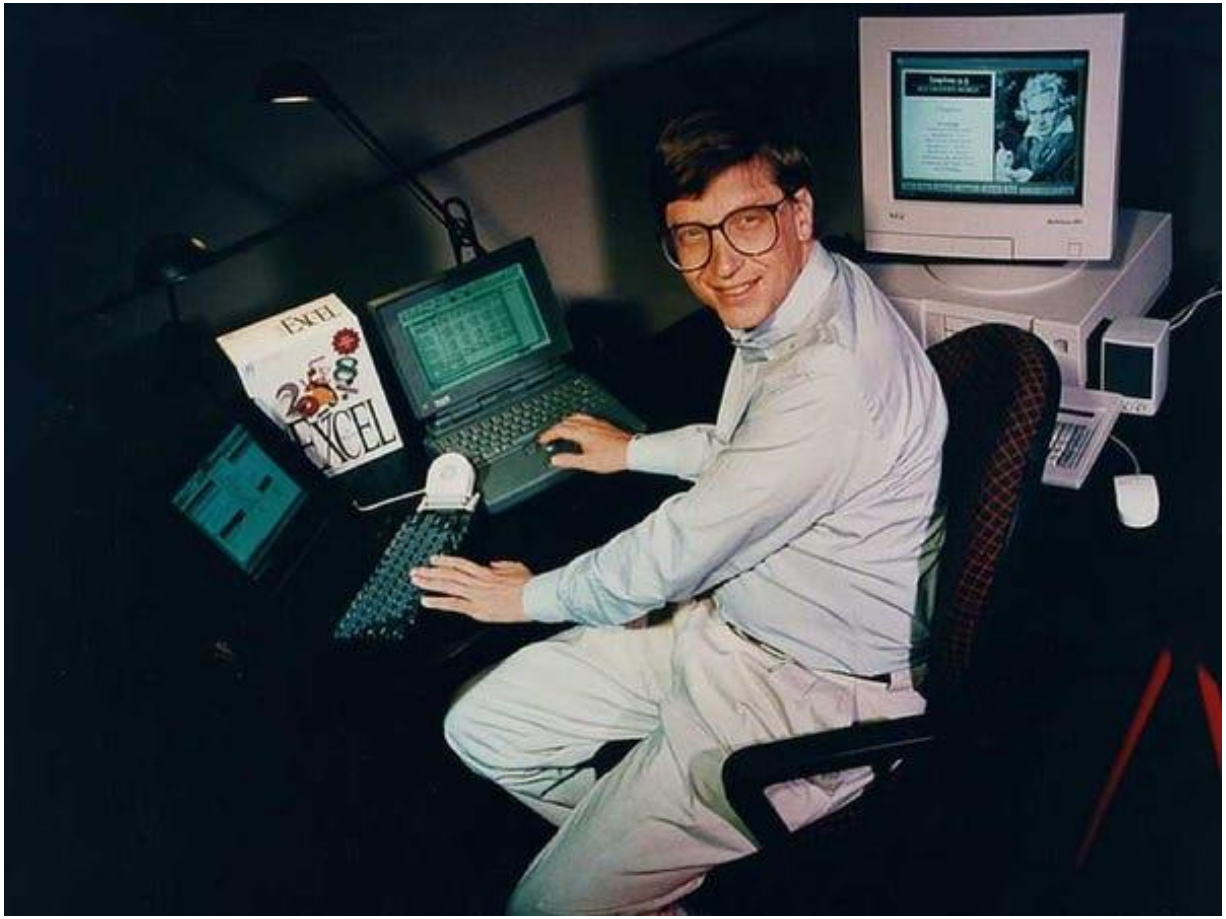
Maximize x+y with the given constraints.





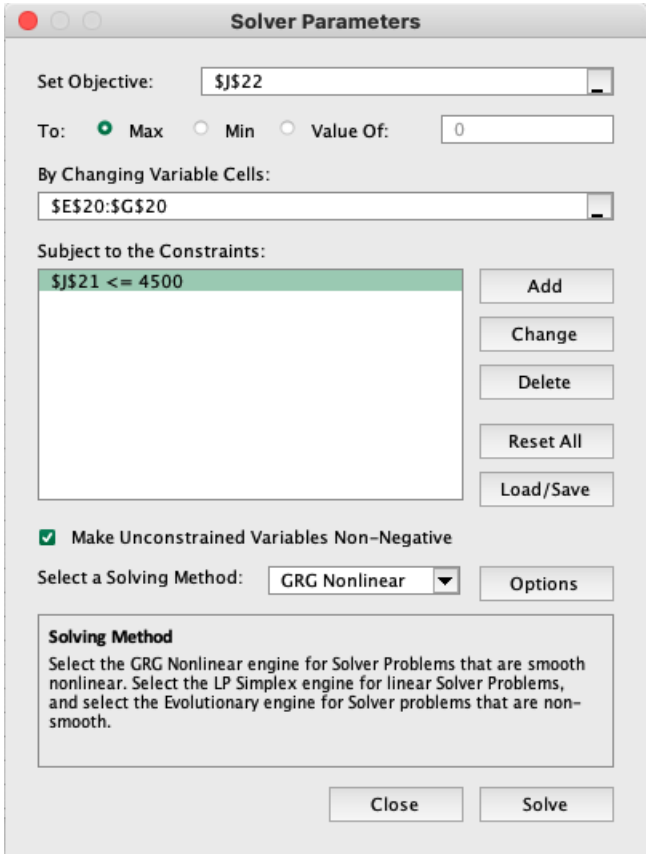
Colorado State University

Excel Magic



Colorado State University

- You need to install add-in in your excel so that it shows up in the Data menu.
- Set up an optimization problem
  - What to optimize (e.g. minimize least square error)
  - Constraints to satisfy
  - Specify initial estimates of parameters
- Iterative solution (choice of algorithms)
- Result may depend on initial values for non-linear fitting.



Colorado State University

Classic Optimization Problem

- Linear Programming, Non-Linear Programming etc.
- Specified
  - Objective function: minimize or maximize
  - Constraints: equalities, inequalities
- Generally solution is iterative
- Excel Solver algorithms
    - Simplex method is used for solving linear problems
    - GRG solver for solving smooth nonlinear problems
    - Evolution solver uses genetic algorithms
- Excel solver for curve fitting
  - Minimize error (square of differences between model and actual data points)
  - Some constraints(e.g. parameters may be required to be positive0

Initial Values

- Start with some initial values and the gradually iterate towards optimal.
- When 3 or more parameters are used, it is best to start with some good initial guesses.
- Algorithm may get stuck at a local minimum/maximum
- Repeat with diverse initial guesses.

Excel Solver

See Solver demo

Colorado State University

Quantitative Security

Colorado State University

Yashwant K Malaiya

CS559

Vulnerabilities



CSU Cybersecurity Center
Computer Science Dept

Security Holes: Types

- Software holes: Vulnerabilities
  - CVSS scores involving *exploitability* and *impact* is a type of risk measure.
- System/physical holes
- Personnel/Procedural holes:
  - e.g. Phishing
- Exploitation may involve multiple holes, perhaps of different types
  - Understanding and classify them:
    - Target 2013 breach: credentials stolen from a HVAC contractor
    - Equifax 2017 breach: vulnerability patch not applied

Defects vs vulnerabilities

- Software defects vs Vulnerabilities
  - Defect: any defect
  - Vulnerability: security related defect, perhaps 1-5% of defects
- Testing for defects or vulnerabilities:
  - Defects: Apply inputs to ensure the output is as expected
    - Or examine code to look for flaws
  - Vulnerabilities: Apply inputs to see if they cause unexpected behavior
    - Or examine code to look for flaws
- When are the discovered
  - prior to release by developer
  - Field testing: developers, users, bug-hunters

Risks due to vulnerabilites

- **Risk** due to an adverse event e_i

  $Risk_i = Likelihood_i \times Impact_i$

- $Likelihood_i$ = P{A security $hole_i$ is exploited}.

  = P{$hole_i$ present}.

  P{exploitation | $hole_i$ present}

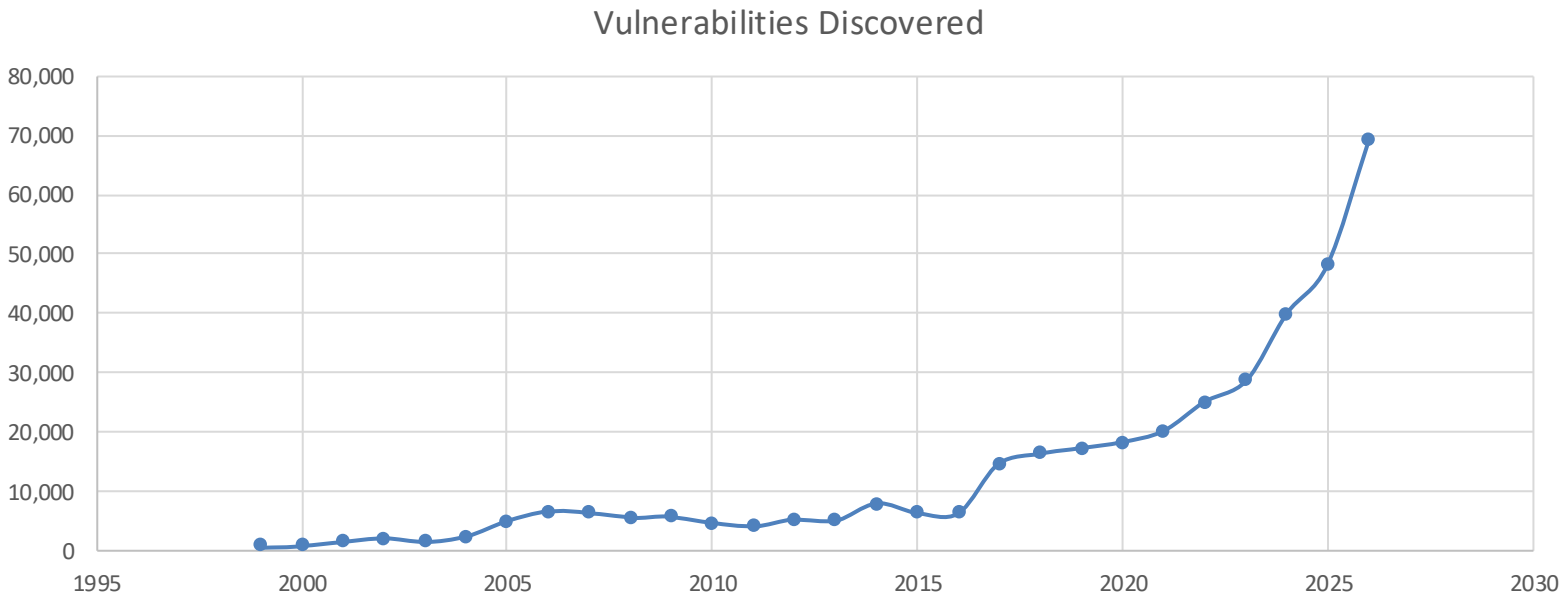
  - P{$hole_i$ present}: an internal attribute of the system.
  - P{exploitation | $hole_i$ present}: depends on circumstances outside the system.

Components of Likelihood of Exploitation

- Internal
  - Presence of a vulnerability (Vulnerability Discovery*)
  - Vulnerability not patched
- External
  - Attacker's motivation level
  - Technical capabilities, exploit availability*
  - Network access to vulnerable system
- Interface
  - Attack surface* of vulnerable system
  - Reachability* of vulnerability

We have published research related to these. See publications from 2005

Vulnerability Discovery Trend



Vulnerability Data-bases

NVD database
CVEDetails
VulnDB
ExploitDB

2026: only partial data

Why the jump in 2017?
More extensive collection by NVD?

Colorado State University