1

Introduction

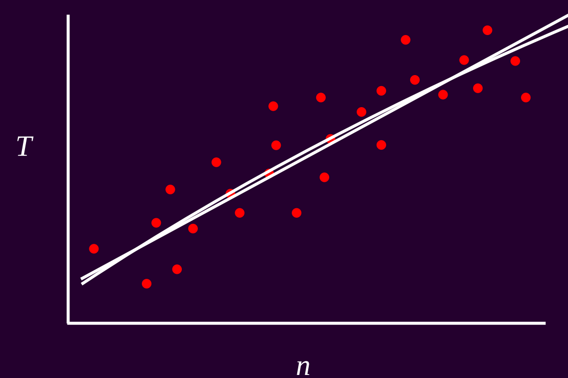
- In this topic, we will
  - Consider how we need to analyze and find the polynomial growth of data an^b or $O(n^b)$
  - Walk through an example
  - Consider how we need to analyze and find exponential growth or decay of data ae^{bn}
  - Walk through an example

2

2

Fitting a polynomial

- Recall that you can fit a polynomial to data



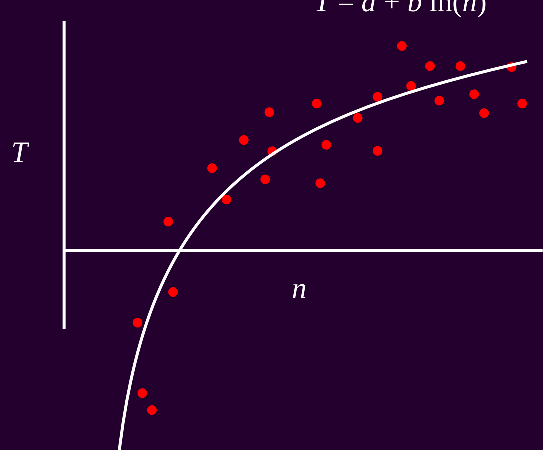
3

3

Fitting linear combinations of functions

- You can also fit any linear combination of functions:

$T = a + b\ \ln(n)$



$$V = \begin{pmatrix} 1 & \ln(n_1) \\ 1 & \ln(n_2) \\ 1 & \ln(n_3) \\ \vdots & \vdots \\ 1 & \ln(n_N) \end{pmatrix} \quad \mathbf{T} = \begin{pmatrix} T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_N \end{pmatrix}$$

$$V^{\mathrm{T}} V \begin{pmatrix} a \\ b \end{pmatrix} = V^{\mathrm{T}} \mathbf{T}$$

4

4

Other forms of data

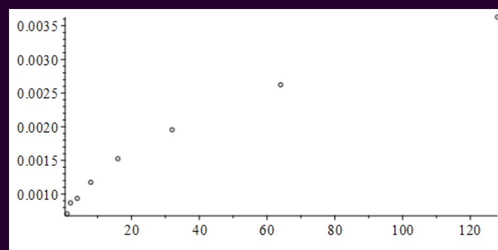
- Consider these situations:
  - Your run-time shows polynomial growth with respect to capacity:
    $T(n) = an^b + o(n^b)$
  - Your data is growing or decaying exponentially over time:
    $y(t) = a\, e^{bt}$
- It is important to note:
  - Neither of these is in the form $a_1 x + a_0$
  - In the first case, if $y = a\, x^b$, then
    $$\ln(a\, x^b) = \ln(a) + b \ln(x)$$
  - In the second case, if $y = a\, e^{bx}$, then
    $$\ln(y) = \ln(a\, e^{bx}) = \ln(a) + bx$$

Polynomial growth

- Polynomial growth is described by any function that is $O(n^d)$
  - Suppose you've just implemented a bi-parental heap that contains n items
    - You know that certain operations are $O\left(\sqrt{n}\right)$
  - You just authored the Karatsuba algorithm for multiplying two n-digit integers
    - You know the run time must be $O\left(n^{\log_2(3)}\right) \approx O\left(n^{1.585}\right)$

Polynomial growth

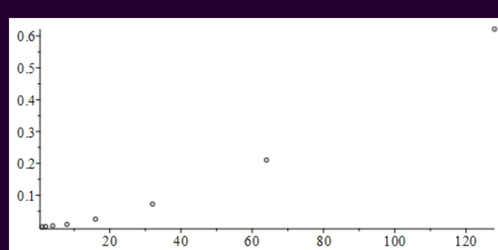
- So, you time your operation on the bi-parental heap for various values of $n = 2^k$ items contained in the container



  - It looks like $\mathrm{O}\left(\sqrt{n}\right)$ but is it $\mathrm{O}(n^{2/3})$?
  - We should find the best fitting $T(n) = an^b$ and see if $b \approx 0.5$

Polynomial growth

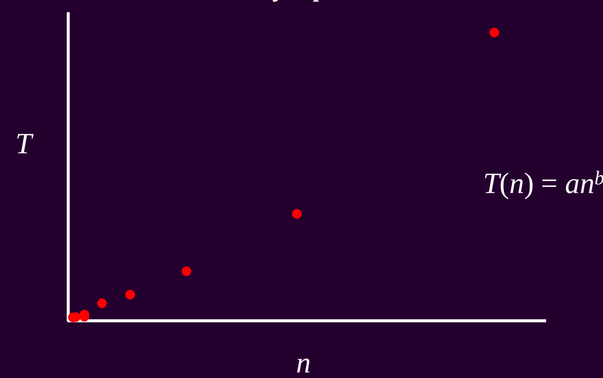
- Alternatively, you try your integer multiplication routine with some large-integer class with integers of $n = 2^m$ digits



  - It looks like $\mathrm{O}\left(n^{\log_2(3)}\right)$ but is it $\mathrm{O}(n^2)$?
  - We should find the best fitting $T(n) = an^b$ and see if $b \approx 1.585$

Polynomial growth

- Alternatively, suppose you've implemented a new algorithm and you'd like to determine the asymptotic behavior



Polynomial growth

- Suppose data has polynomial growth, so

$$T(n) = an^b + \mathrm{o}(n^b)$$

  – Therefore, for larger values of n_k, we have

$$T_k \approx a n_k^b$$

  – Thus, taking logarithms of both sides yields

$$\begin{aligned}\ln(T_k) &\approx \ln(a n_k^b) \\ &= \ln(a) + \ln(n_k^b) \\ &= \ln(a) + b\ln(n_k)\end{aligned}$$

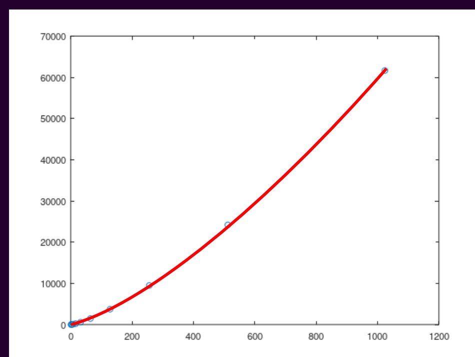
Polynomial and exponential growth

Polynomial growth

- Let's try this out:

```
>> ns = [1 2 4 8 16 32 64 128 256 512 1024];
>> Ts = 5.32*ns.^1.35;
>> plot( ns, Ts, 'o' );
```

$$T_k = 5.32 n_k^{1.35}$$



11

11

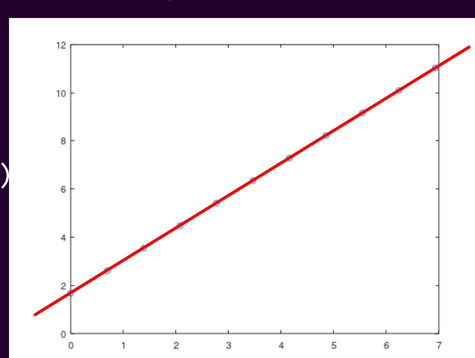
Polynomial and exponential growth

Polynomial growth

- Let's try this out:

```
>> ns = [1 2 4 8 16 32 64 128 256 512 1024];
>> Ts = 5.32*ns.^1.35;
>> plot( log( ns ), log( Ts ), 'o' );
```

  - The slope of the line is 1.35
  - The *y*-intercept is ln(5.32)
  - Recall that `log(x)` is the natural logarithm ln(*x*)
    `log2(x)`
    `log10(x)`



12

12

Polynomial growth

- How can we use this?
  - You have authored your algorithm and run it on various input sizes (values of n) and recorded the run time
  - Two issues:
    - Your program is not running in exactly in an^b time
      - It will be $an^b + o(n^b)$, so a log-log plot will not be a straight line
    - There will be noise in your timings

13

13

Polynomial growth

- Step 1: Collect data
  - Use exponentially growing values of $n = 2^k$ for $k = 0, 1, 2, 3, \ldots, N_{\max}$ and record the time T_k for $n = 2^k$
    - Try to get as many points as is reasonable, so sizes up to $n = 2^{16}$ or more ($N_{\max} \geq 16$)
      - This ensures lower-order terms are overwhelmed by the dominant term
    - If the run time is too large for problems of size $n = 2^{16}$, use $n = \left\lfloor \sqrt{2}^k \right\rfloor$ or even $n = \left\lfloor \sqrt[3]{2}^k \right\rfloor$
  - Gather at least two sample run times per n, as each sample will likely have some error in the timing
    - With three samples, we will have $T_{k,1}$, $T_{k,2}$ and $T_{k,3}$
    - You cannot determine error from a single sample

14

Polynomial growth

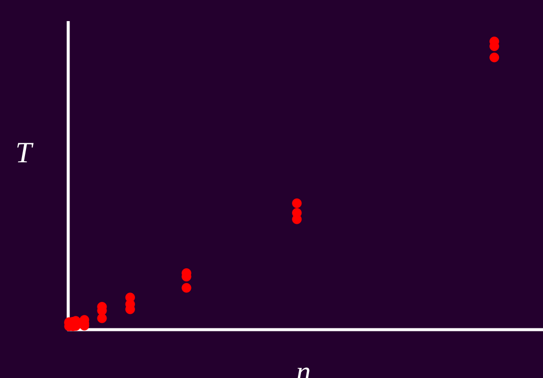
- For example, if the algorithm is Gaussian elimination and backward substitution, we may be able to deduce that the average run time is

$$T(n) = 6n^3 + 25n^2 + 2n\ln(n) + 42n + 15$$

  - If $n = 10$, the higher-order term is just twice the lower-order terms
  - If $n = 100$, the higher-order term is 23.5 times the lower-order terms
  - In the limit, the higher order term will be $0.24n$ times the lower-order terms

Polynomial growth

- For example:



Polynomial growth

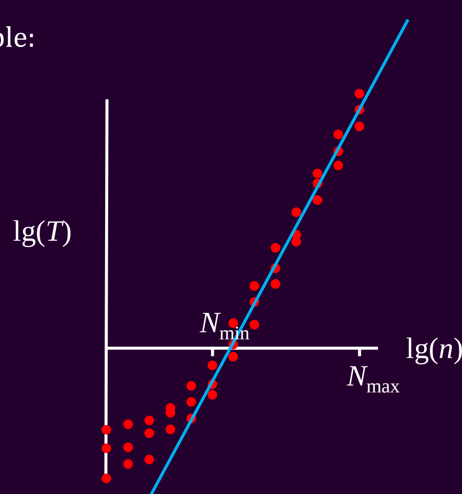
- Step 2: Analysis
  - Plot a log-log plot of $\log(2^k)$ versus $\log(T_{k,j})$
  - It doesn't matter the base of the logarithm, so long as both are the same
    - If you use a base-2 logarithm, the powers of two are integers
    - Recall that $\lg(n)$ often represents $\log_2(n)$
  - Determine visually if it appears in the limit to be a straight line
    - If it isn't, you may have exponential growth or decay!
  - Determine at which point it seems that the effect of the lower-order terms (the $o(n^b)$ component) has diminished impact

17

17

Polynomial growth

- For example:



18

18

Polynomial growth

- Step 3: Linear regression
  - On those points you determined were reasonable, preform a linear regression

$$n = 2^{N_{\min}}, \ldots, 2^{N_{\max}},$$

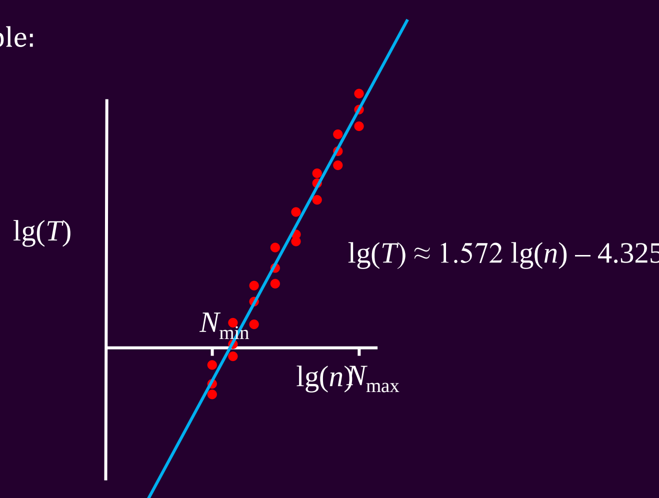
  - Solve $V^{\mathrm{T}}V\begin{pmatrix} b \\ a \end{pmatrix} = V^{\mathrm{T}}\mathbf{y}$

$$b\lg(n)+a$$

$$V = \begin{pmatrix} N_{\min} & 1 \\ N_{\min} & 1 \\ N_{\min} & 1 \\ N_{\min}+1 & 1 \\ N_{\min}+1 & 1 \\ N_{\min}+1 & 1 \\ \vdots & \vdots \\ N_{\max} & 1 \\ N_{\max} & 1 \\ N_{\max} & 1 \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} \lg\left(T_{N_{\min},1}\right) \\ \lg\left(T_{N_{\min},2}\right) \\ \lg\left(T_{N_{\min},3}\right) \\ \lg\left(T_{N_{\min}+1,1}\right) \\ \lg\left(T_{N_{\min}+1,2}\right) \\ \lg\left(T_{N_{\min}+1,3}\right) \\ \vdots \\ \lg\left(T_{N_{\max},1}\right) \\ \lg\left(T_{N_{\max},2}\right) \\ \lg\left(T_{N_{\max},3}\right) \end{pmatrix}$$

Polynomial growth

- For example:

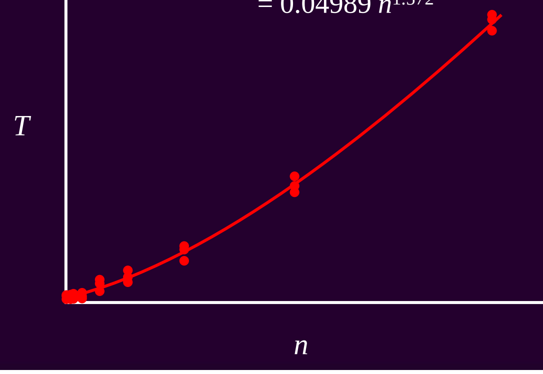


Polynomial growth

- Thus, if lg(T) ≈ 1.572 lg(n) – 4.325, it follows

$$T = 2^{\lg(T)} \approx 2^{1.572 \lg(n) - 4.325}$$
$$= 2^{1.572 \lg(n)}\, 2^{-4.325}$$
$$\approx n^{1.572}\, 0.04989$$
$$= 0.04989\, n^{1.572}$$

$$\beta^{a \log_\beta(n)} = n^a$$



Exponential growth or decay

- Exponential growth is described by any function that is $\Theta(e^{bn})$
  – This includes radioactive decay

$$m_0 e^{-0.00012097n}$$

  – It also describes exponential growth:
    - Moore's law says that the number of transistors in a dense integrated circuit doubles every at two years since 1970

$$Ce^{0.3466n}$$

    - I found a paper by Anthony Ricciardi and Rachael Ryan:

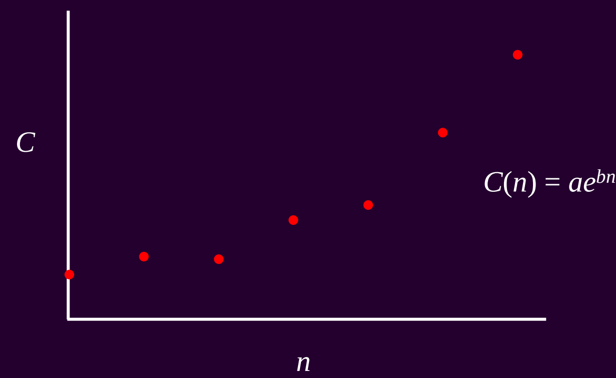
      The exponential growth of invasive species denialism

$$0.122e^{0.18n}$$

      – This is number of years since 1990

Exponential growth or decay

- Suppose you are counting the number hives of Asian giant hornets found in North America



Exponential growth or decay

- Suppose data has exponential growth or decay, so

$$C(n) = ae^{bn}$$

  - Therefore, for various values of n_k, we have

$$C_k \approx ae^{bn_k}$$

  - Thus, taking logarithms of both sides yields

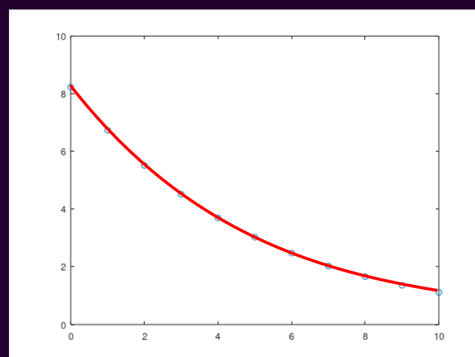
$$\begin{aligned} \ln(C_k) &\approx \ln\left(ae^{bn_k}\right) \\ &= \ln(a) + \ln\left(e^{bn_k}\right) \\ &= \ln(a) + bn_k \end{aligned}$$

Exponential growth or decay

- Let's try this out:

```
>> ns = 0:10;
>> Cs = 8.23*exp( -0.2*ns );
>> plot( ns, Cs, 'o' );
```

$$C_k = 8.23e^{-0.2n_k}$$

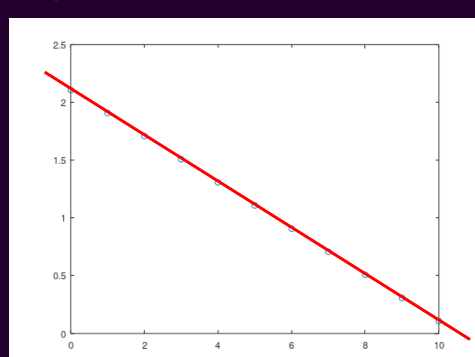


Exponential growth or decay

- Let's try this out:

```
>> ns = 0:10;
>> Cs = 8.23*exp( -0.2*ns );
>> plot( ns, log( Cs ), 'o' );
```

  - The slope of the line is –0.2
  - The *y*-intercept is ln(8.23)



Exponential growth or decay

- How can we use this?
  - You have collected your data and you understand it to be growing or decaying exponentially
  - Two issues:
    - There will be noise in your data
    - There may be other factors

27

27

Exponential growth or decay

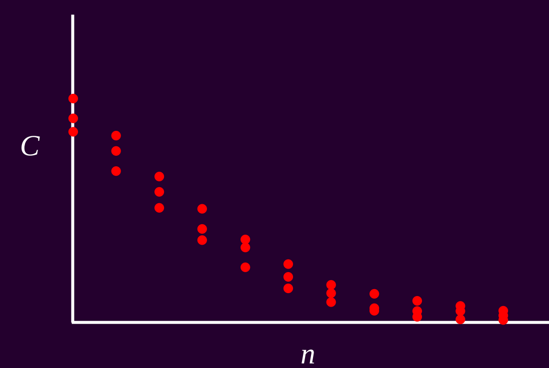
- Step 1: Collect data
  - Use different values of (n_0, C_0), (n_1, C_1), (n_2, C_2), …, (n_N, C_N),
    - Try to get as many points as is reasonable
    - Equally spaced points are common, but not required
    - Often $n_0 = 0$, but this is not required
  - If possible, gather at least two samples per n,
    as each sample will likely have some error in the count
    - With three samples, we will have $C_{k,1}$, $C_{k,2}$ and $C_{k,3}$
    - You cannot determine error from a single sample

28

28

Exponential growth or decay

- For example:

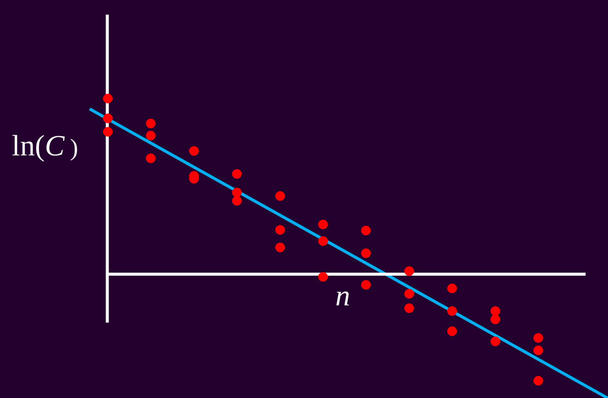


Exponential growth or decay

- Step 2: Analysis
  - Plot a semi-log plot of n_k versus ln($C_{k,j}$)
  - It doesn't matter the base of the logarithm,
    so long as we use the same base later
  - Determine visually if it appears to be a straight line
    - If it isn't, you may have polynomial growth or decay!

Exponential growth or decay

- For example:



Exponential growth or decay

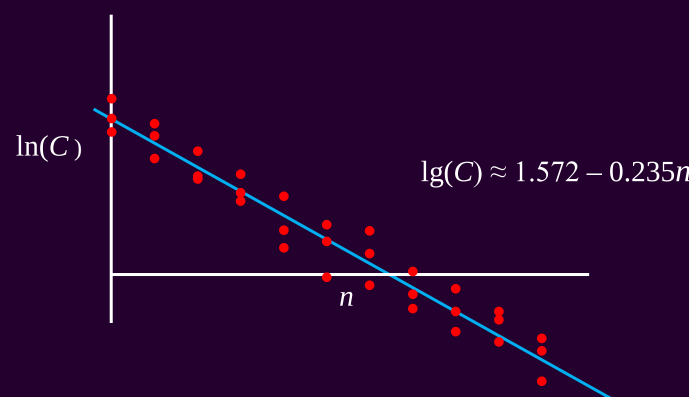
- Step 3: Linear regression
  - On those points you determined were reasonable, preform a linear regression
  - Solve $V^{\mathrm{T}} V \begin{pmatrix} a \\ b \end{pmatrix} = V^{\mathrm{T}} \mathbf{y}$

$$a + bn$$

$$V = \begin{pmatrix} 1 & n_0 \\ 1 & n_0 \\ 1 & n_0 \\ 1 & n_1 \\ 1 & n_1 \\ 1 & n_1 \\ \vdots & \vdots \\ 1 & n_N \\ 1 & n_N \\ 1 & n_N \end{pmatrix} \qquad \mathbf{y} = \begin{pmatrix} \ln(C_{0,1}) \\ \ln(C_{0,2}) \\ \ln(C_{0,3}) \\ \ln(C_{1,1}) \\ \ln(C_{1,2}) \\ \ln(C_{1,3}) \\ \vdots \\ \ln(C_{N,1}) \\ \ln(C_{N,2}) \\ \ln(C_{N,3}) \end{pmatrix}$$

Exponential growth or decay

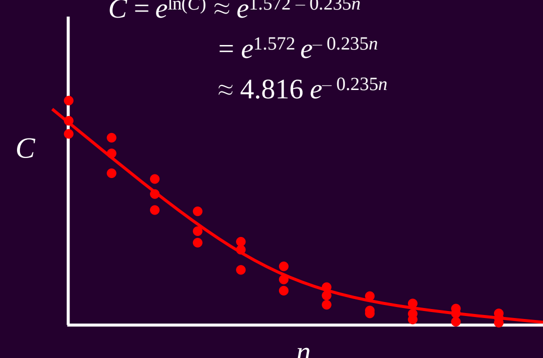
- For example:



Exponential growth or decay

- Thus, if $\ln(C) \approx 1.572 - 0.235n$, it follows

$$C = e^{\ln(C)} \approx e^{1.572 - 0.235n}$$
$$= e^{1.572}\, e^{-0.235n}$$
$$\approx 4.816\, e^{-0.235n}$$



  - We can also find the half-life by solving $e^{-0.735n} = 0.5$
  - Thus, the half life is $n_{½} = -\ln(0.5)/0.235 = 2.9496$

Exponential growth or decay

- If the data was exponentially growing:
  - The coefficient $b > 0$
  - You can calculate the doubling time by solving $e^{bn} = 2$ once you have estimated b

Polynomial-logarithmic growth

- What do we do about growth that are in these forms?

$$T = an^b \ln(n) \text{ or } T = an^b \ln^c(n)$$

  - Problem: $\ln(n) = o(n^b)$ for any $b > 0$
  - Consequently, you could do the following:

$$\ln(T) = \ln(a) + b \ln(n) + \ln(\ln(n))$$

$$\ln(T) - \ln(\ln(n)) = \ln(a) + b \ln(n)$$

$$\ln(T) = \ln(a) + b \ln(n) + c \ln(\ln(n))$$

  - This may be useful to confirm an implementation is, say, $n \ln(n)$
- These techniques may be useful if you are testing for:

$$T = a \ln^c(n)$$

  - Thus, we would match $\ln(T) = \ln(a) + c \ln(\ln(n))$

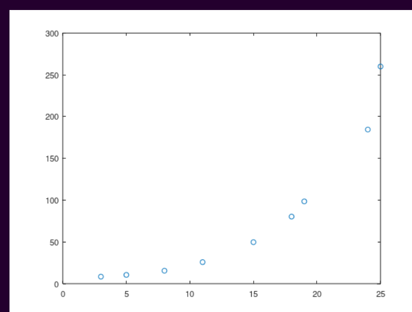
Polynomial and exponential growth

Example 1

- Suppose we have this example:

```
>> xs = [3    5    8    11    15    18    19    24    25]';
>> ys = [8.4 10.5 15.5 25.8 49.7 80.3 98.5 184.5 260.0]';
>> plot( xs, ys, 'o' );
```

  – Is it growing polynomially or exponentially?



37

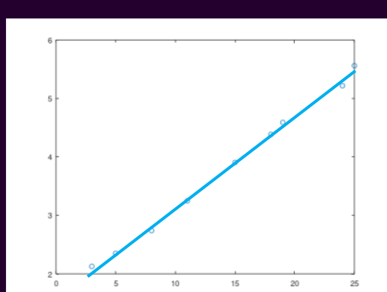
37

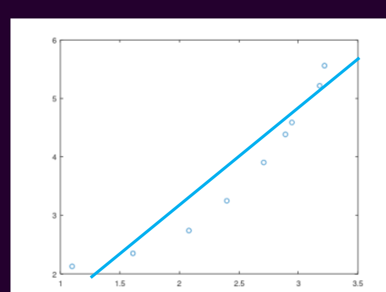
Polynomial and exponential growth

Example 1

- Let's plot both:

```
>> plot( xs, log( ys ), 'o' );
>> plot( log( xs ), log( ys ), 'o' );
```





38

38

Example 1

- We can find the best-fitting exponential curve:

```
>> V = [ones(9,1) xs];
>> a = V \ log( ys )
    a =
       1.576485560599445
       0.155801668985937
>> exp( a(1) )
    ans = 4.837923310623190
```

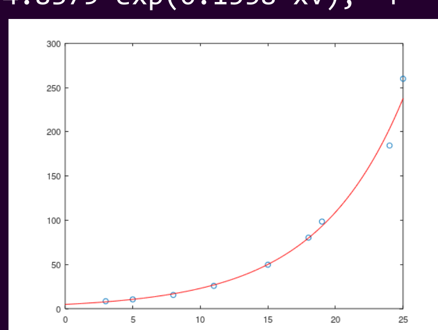
$$y(x) \approx 4.8379e^{0.1558x}$$

Example 1

- We can now plot the data and the best-fitting exponential curve:

```
>> plot( xs, ys, 'o' )
>> hold on
>> xv = 0:0.1:25;
>> plot( xv, 4.8379*exp(0.1558*xv), 'r' );
```

$$y(x) \approx 4.8379e^{0.1558x}$$

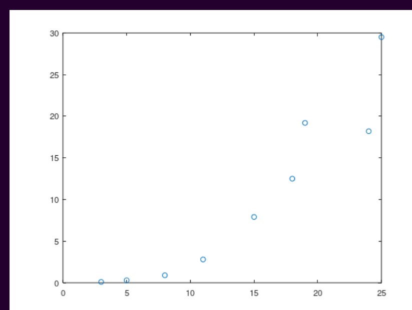


Example 2

- Suppose we have this example:

```
>> xs = [3   5   8   11   15   18   19   24   25]';
>> ys = [0.1 0.3 0.9  2.8  7.9 12.5 19.2 18.2 29.5]';
>> plot( xs, ys, 'o' );
```

  – Is it growing polynomially or exponentially?



41

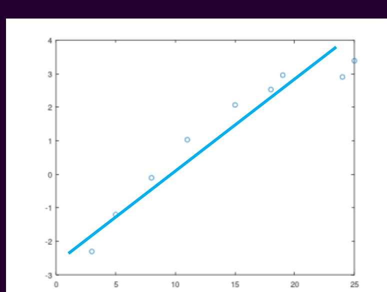
41

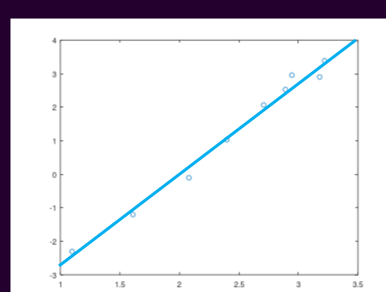
Polynomial and exponential growth

Example 2

- Let's plot both:

```
>> plot( xs, log( ys ), 'o' );
>> plot( log( xs ), log( ys ), 'o' );
```





42

42

Example 2

- We can find the best-fitting polynomial curve:

```
>> V = [ones(9,1) log( xs )];
>> a = V \ log( ys )
    a =
       -5.519325745036392
        2.753647782022680
>> exp( a(1) )
    ans = 4.008549815722325e-03
```

$$y(x) \approx 0.004009x^{2.7536}$$

43

43

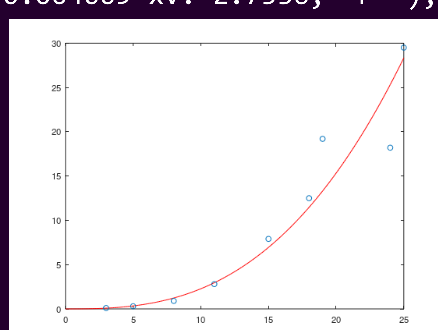
Polynomial and exponential growth

Example 2

- We can now plot the data and the best-fitting exponential curve:

```
>> plot( xs, ys, 'o' )
>> hold on
>> xv = 0:0.1:25;
>> plot( xv, 0.004009*xv.^2.7536, 'r' );
```

$$y(x) \approx 0.004009x^{2.7536}$$



44

44

Summary

- Following this topic, you now
  - Understand that you can use least-squares best-fitting linear polynomials to find both polynomial and exponential growth
  - Know the data may be transformed to being linear by:
    - Taking the logarithm of both values for polynomial growth
    - Taking the logarithm of the counts for exponential growth or decay
  - Understand that using least-squares best-fitting techniques, this gives us the best estimates of the unknown coefficients
  - Know that you can calculate the half life for exponential decay by solving $e^{bn} = ½$ and the doubling time by solving $e^{bn} = 2$

45

45

Appproximating integrals using interpolating polynomials

References

[1] https://en.wikipedia.org/wiki/Log%E2%80%93log_plot

[2] https://en.wikipedia.org/wiki/Semi-log_plot

46

46

Acknowledgments

None so far.

Colophon

These slides were prepared using the Cambria typeface. Mathematical equations use Times New Roman, and source code is presented using `Consolas`. Mathematical equations are prepared in MathType by Design Science, Inc.
Examples may be formulated and checked using Maple by Maplesoft, Inc.

The photographs of flowers and a monarch butter appearing on the title slide and accenting the top of each other slide were taken at the Royal Botanical Gardens in October of 2017 by Douglas Wilhelm Harder. Please see

https://www.rbg.ca/

for more information.

Disclaimer

These slides are provided for the ECE 204 *Numerical methods* course taught at the University of Waterloo. The material in it reflects the author's best judgment in light of the information available to them at the time of preparation. Any reliance on these course slides by any party for any other purpose are the responsibility of such parties. The authors accept no responsibility for damages, if any, suffered by any party as a result of decisions made or actions based on these course slides for any other purpose than that for which it was intended.