Lecture 18 : Least squares

Tuesday, October 27, 2015 9:30 AM

Admin:

Natural problems for solving linear systems:

1. Solve $Ax = b$ for x

TODAY

2. Assuming $Ax = b$ has a solution, find the shortest solution x.
3. "Least squares": If $Ax = b$ has no solution ($b \notin R(A)$), find an x that minimizes $\|Ax - b\|$.
4. "Compressed sensing": Find the sparsest x such that $Ax = b$.

Simple application of SVD:

Theorem: For any matrix A,

- $\operatorname{rank}(A^T A) = \operatorname{rank}(A)$
- $R(A^T A) = R(A^T)$
- $N(A^T A) = N(A)$.

Proof: We have seen this already, but it is much easier to prove using the SVD.

Let the SVD of A be

$$A = \sum_i \lambda_i \vec{u}_i \vec{v}_i^T \qquad \Rightarrow \operatorname{rank}(A) = \#\{i \mid \lambda_i > 0\}$$

$$\Rightarrow \quad A^T A = \left(\sum_i \lambda_i v_i u_i^T\right)\left(\sum_j \lambda_j u_j v_j^T\right)$$

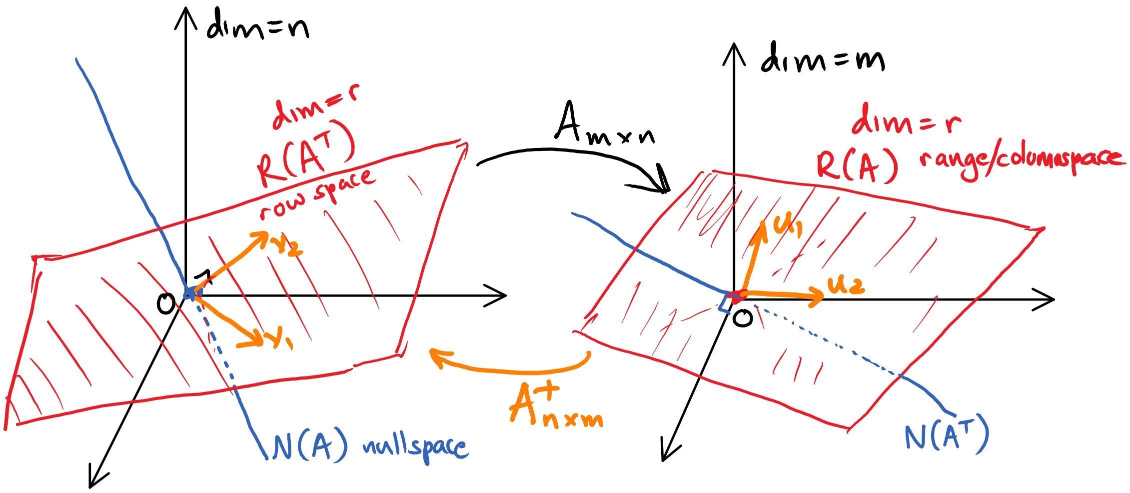
$$= \sum_i \lambda_i^2 v_i v_i^T \qquad \text{since } u_i \cdot u_j = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

This is ~~a~~ SVD for $A^T A$ $\Rightarrow \operatorname{rank}(A^T A) = \#\{i \mid \lambda_i^2 > 0\}$

$$\Rightarrow \operatorname{rank}(A^T A) = \operatorname{rank}(A). \quad \checkmark$$

The other statements are all similar (exercises). $\square$

Definition: Matrix pseudoinverse



The pseudoinverse of A, denoted A^+, is the unique linear operator $\mathbb{R}^m \to \mathbb{R}^n$ with

plus sign; not to be confused with adjoint ($\dagger$) or transpose (T)

- $N(A^+) = R(A)^\perp = N(A^T)$
- $R(A^+) = R(A^T) = N(A)^\perp$
- $A^+A = P_{R(A^T)}$

If the SVD of A is

$$A = \sum_i \lambda_i \vec{u}_i \vec{v}_i^T,$$

then A^+ is given by

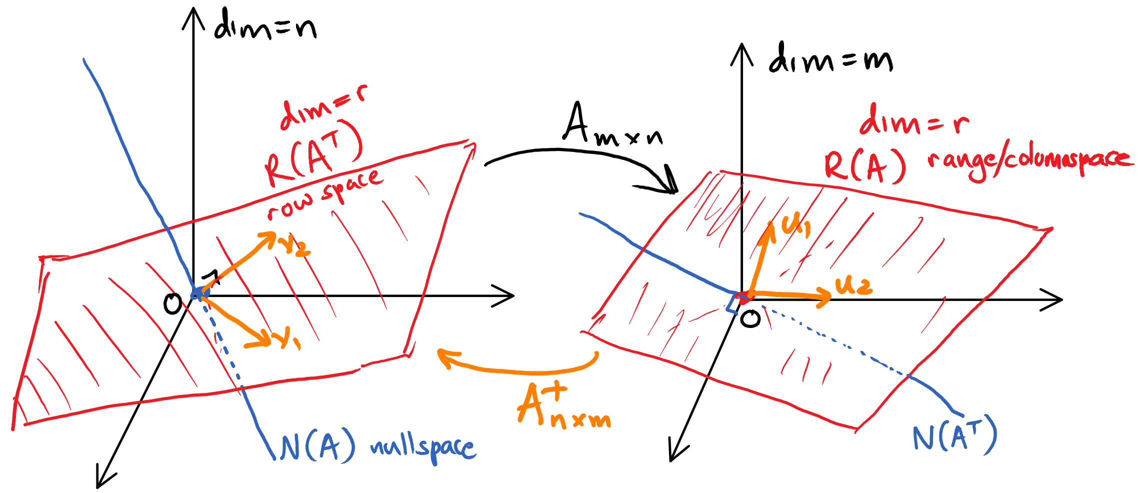
$$\boxed{A^+ = \sum_{i:\lambda_i>0} \frac{1}{\lambda_i} \vec{v}_i \vec{u}_i^T}$$

Properties of the (Moore-Penrose) pseudoinverse:

$$A = \sum_i \lambda_i \vec{u}_i \vec{v}_i^T \qquad A^+ = \sum_{i:\lambda_i>0} \frac{1}{\lambda_i} \vec{v}_i \vec{u}_i^T$$

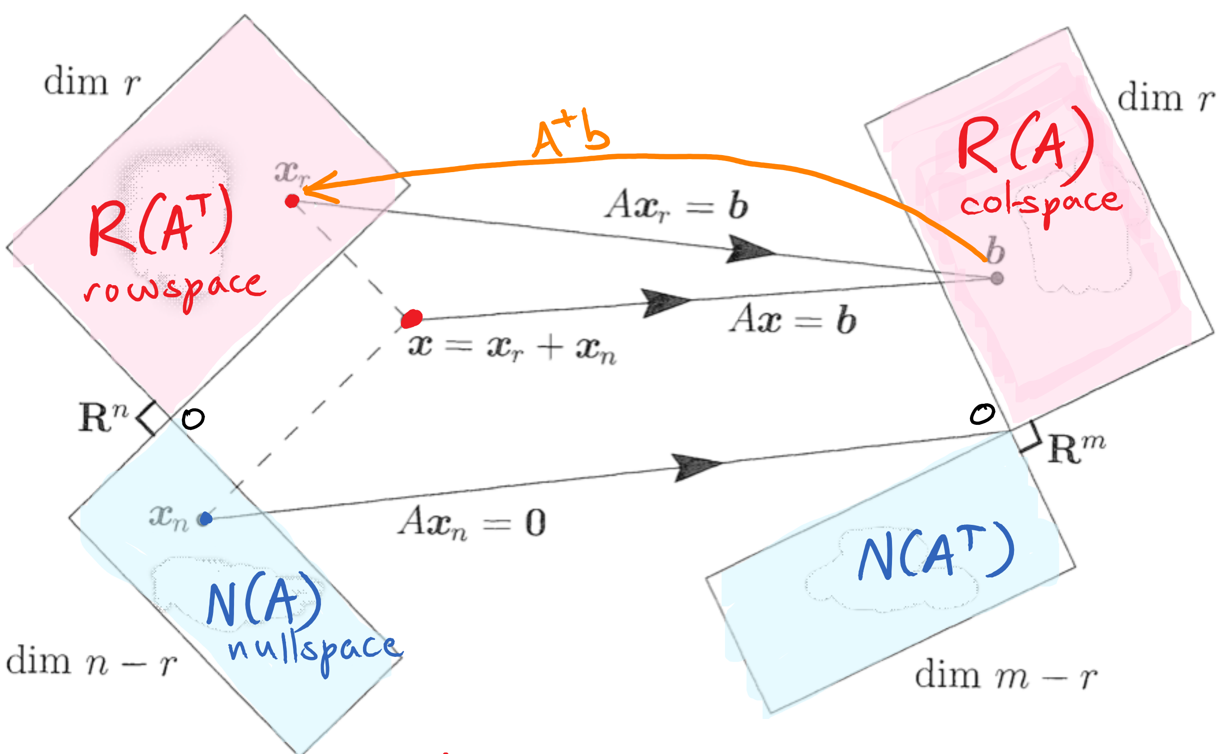
- $A A^+ = \sum_{i:\lambda_i>0} \vec{u}_i \vec{u}_i^T$ = Projection onto $R(A)$
- $A^+ A = \sum_{i:\lambda_i>0} \vec{v}_i \vec{v}_i^T$ = Projection onto $R(A^T)$

$\Rightarrow$ (for example) $A A^+ A = A P_{R(A)} = P_{R(A^T)} A = A$



• If A is invertible/nonsingular, then $A^+ = A^{-1}$.

Corollary: If $Ax = b$ has multiple solutions, ie., $N(A) \neq \{\vec{0}\}$, then $x = A^+ b$ is the shortest solution.



$N(A) = R(A^T)^\perp$ $\qquad$ $N(A^T) = R(A)^\perp$

Why?

Any solution $x \in A^+ b + N(A)$

particular solution

set of solutions to the homogeneous equations $Ax=0$

Since $A^+b \in R(A^T)$, which is $\perp$ to $N(A)$,

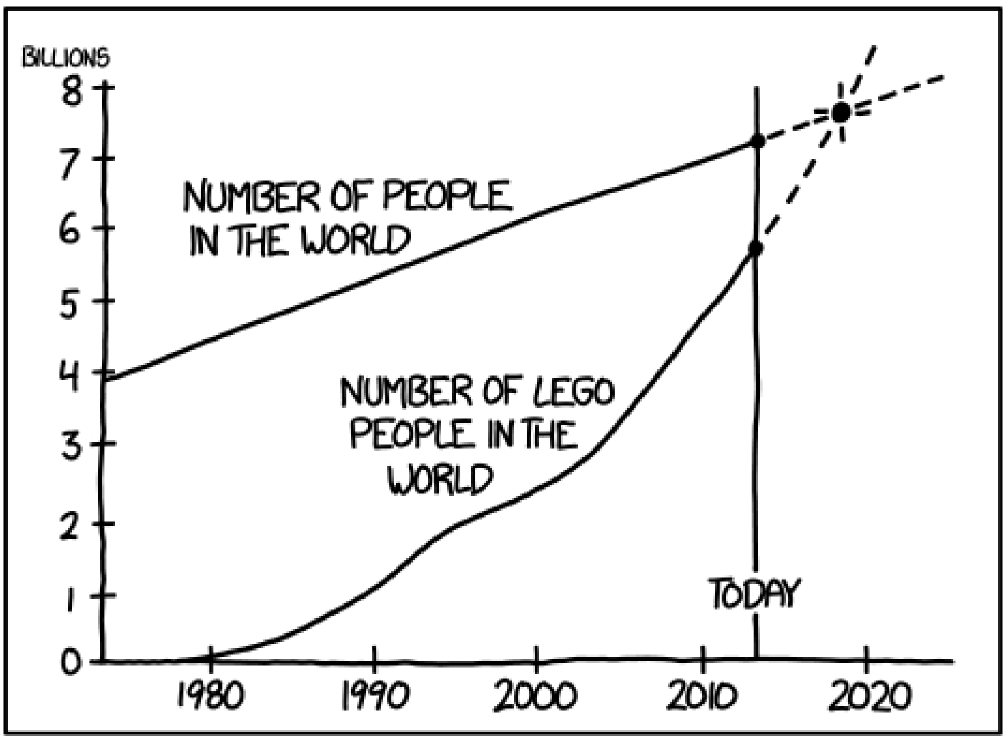
$$\|x\|^2 = \|A^+b\|^2 + \|\text{its component in } N(A)\|^2$$

which is minimal if the component in $N(A)$ is 0,

i.e., $x = A^+b$. ✓ $\square$

"Least Squares" Fitting & Applications to Data Analysis

(Reading: Meyer 4.6, 5.13-14; Strang 3.3)



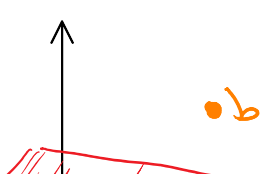
BY 2019, HUMANS WILL BE OUTNUMBERED.

[xkcd.com/1281]

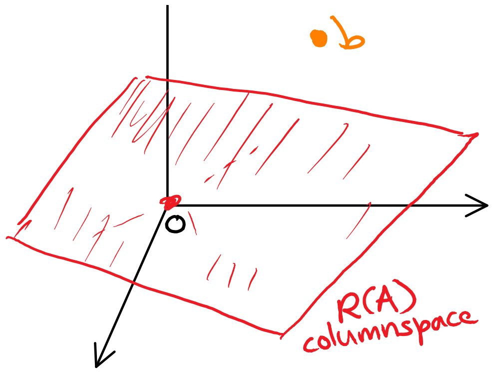
Typical problem: System of equations

$$A\vec{x} = \vec{b}$$

but $\vec{b} \notin R(A)$!



$\Rightarrow$ there is no solution $\vec{x}$



$\Rightarrow$ there is no solution $\vec{x}$

$\Rightarrow$ Instead, find x to minimize $\|Ax-b\|$

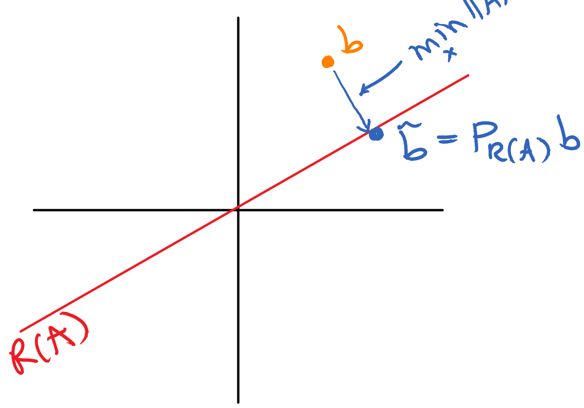
Example: $\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$

For $b_3 \neq 0$, there is no solution, but $x_1 = b_1$ and $x_2 = b_2$ comes closest:

$$\|Ax - b\| = |b_3|.$$

Geometrically:
Project b into $R(A)$, solve

$$Ax = \tilde{b} = P_{R(A)} b$$



Example:

$$\underbrace{\begin{pmatrix} 1 & -1 \\ 2 & 3 \\ 0 & 0 \end{pmatrix}}_{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \underbrace{\begin{pmatrix} 0 \\ 5 \\ 10 \end{pmatrix}}_{b}$$

There is no solution, since $(Ax)_3 = 0$ always.

But $R(A) = \text{Span}\left\{ \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 3 \\ 0 \end{pmatrix} \right\}$

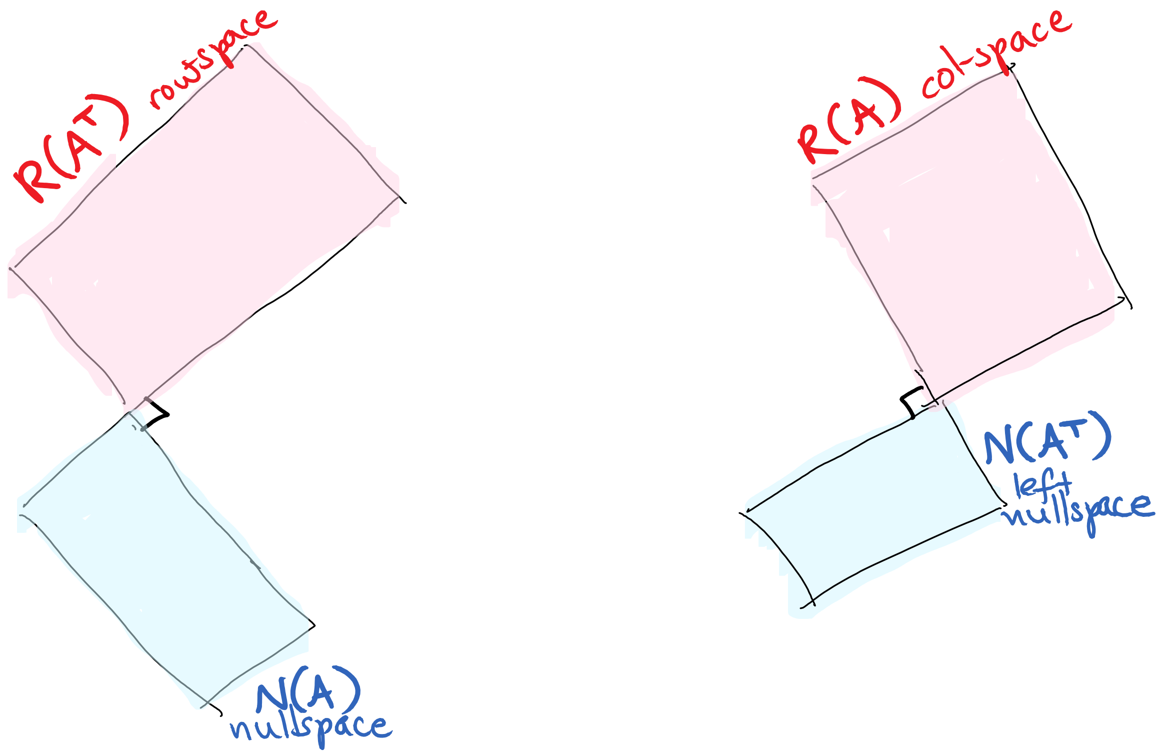
$= xy\text{-plane in } \mathbb{R}^3$

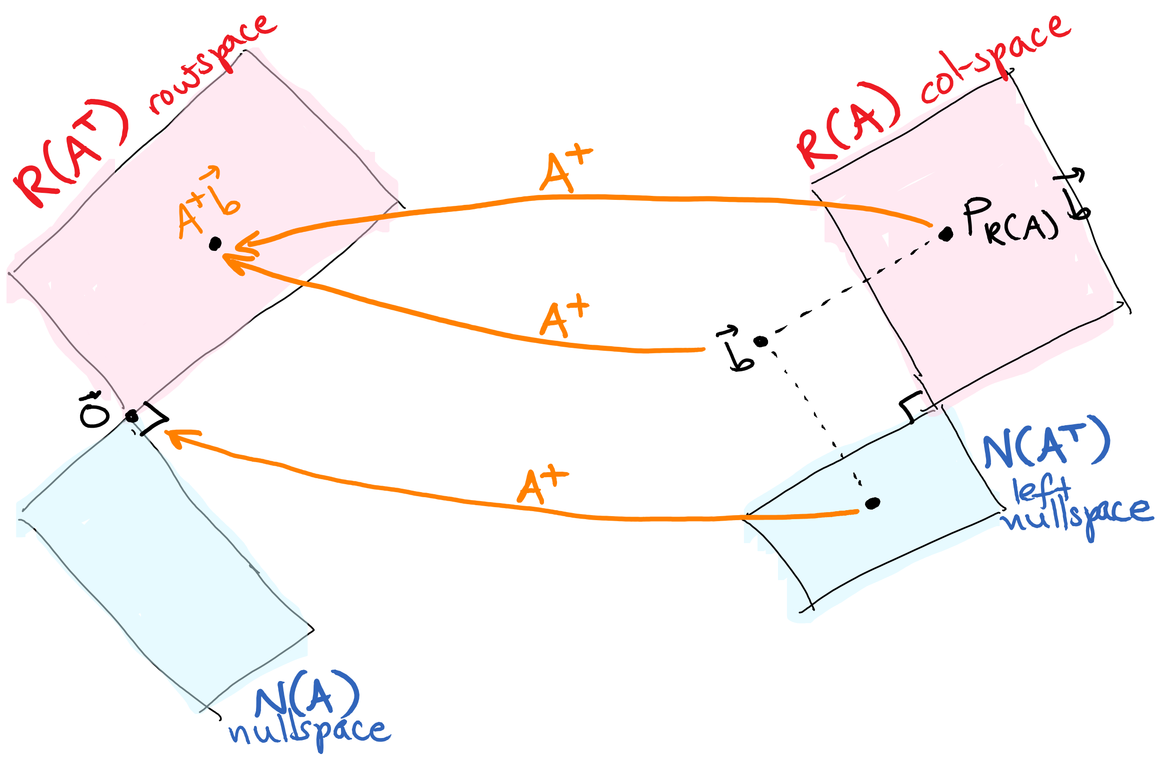
$\Rightarrow$ min-error solution solves

$$A\vec{x} = P_{R(A)} \vec{b} = \begin{pmatrix} 0 \\ 5 \\ 0 \end{pmatrix}$$

$$\Rightarrow \vec{x} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} . \checkmark$$

Another approach...





$$\boxed{\Rightarrow x = A^+ b \text{ minimizes } \|Ax-b\|}$$

Algebraically: By the SVD,

$$A = \sum_{i:\lambda_i>0} \lambda_i u_i v_i^T$$

$$\Rightarrow R(A) = \mathrm{Span}(\{\vec{u}_i \mid \lambda_i > 0\})$$

$$(\text{and } R(A^T) = \mathrm{Span}(\{\vec{v}_i \mid \lambda_i > 0\}))$$

Since the $\{u_i\}$ are orthonormal, they form a **basis** for $R(A)$

$$\Rightarrow P_{R(A)} = \sum_{i:\lambda_i>0} u_i u_i^T$$

$\Rightarrow$ We want to solve

$$Ax = P_{R(A)} b$$

$$\sum_{i:\lambda_i>0} \lambda_i \vec{u}_i (v_i \cdot x) = \sum_{i:\lambda_i>0} \vec{u}_i (u_i \cdot b)$$

$$\Rightarrow v_i \cdot x = \begin{cases} \frac{1}{\lambda_i} u_i \cdot b & \text{if } \lambda_i > 0 \\ 0 & \text{if } \lambda_i = 0 \end{cases}$$

$$\boxed{\Rightarrow \vec{x} = \sum_{i:\lambda_i>0} \frac{1}{\lambda_i}(u_i \cdot b)\vec{v}_i = \left(\sum_{i:\lambda_i>0} \frac{1}{\lambda_i} v_i u_i^T\right)\vec{b}} = A^+ \vec{b}$$

$$\boxed{\Rightarrow x = A^+ b \text{ minimizes } \|Ax-b\|}$$

Example:

```
>> A = [1 -1; 2 3; 0 0]

A =

     1    -1
     2     3
     0     0

>> format long e
>> [U,D,V] = svd(A)
```

left singular vectors

```
U =

  -8.980559531591714e-02    9.959593139531120e-01                        0
   9.959593139531121e-01    8.980559531591698e-02                        0
                       0                        0    1.000000000000000e+00


D =

   3.618033988749895e+00                        0
                       0    1.381966011250106e+00
                       0                        0
```

singular values

right singular vectors

```
V =

   5.257311121191336e-01    8.506508083520399e-01
   8.506508083520399e-01   -5.257311121191336e-01

>> Apseudoinv = V * [1/3.618033988749895 0 0; 0 1/1.381966011250106 0] * U'

Apseudoinv =

   5.999999999999995e-01    1.999999999999999e-01                        0
  -3.999999999999997e-01    2.000000000000000e-01                        0
```

$$= \begin{pmatrix} .6 & .2 & 0 \\ -.4 & .2 & 0 \end{pmatrix} = \frac{1}{5}\begin{pmatrix} 3 & 1 & 0 \\ -2 & 1 & 0 \end{pmatrix}$$

```
>> Apseudoinv * [0; 5; 10]

ans =

   9.999999999999996e-01
   1.000000000000000e+00
```

$= \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ ✓

```
>> format short e
>> Apseudoinv2 = pinv(A)
```

Matlab's built-in pseudo-inverse function

```
Apseudoinv2 =

   6.0000e-01   2.0000e-01            0
```

same as we got above!

Matlabs built in ...

```
Apseudoinv2 =

   6.0000e-01   2.0000e-01            0
  -4.0000e-01   2.0000e-01            0
```

same as we got above!

```
>> A * Apseudoinv

ans =

   1.0000e+00  -1.3878e-16            0
            0   1.0000e+00            0
            0            0            0
```

Observe:

AA^+ = projection onto xy-plane

```
>> Apseudoinv * A

ans =

   1.0000e+00   1.1102e-16
   3.8858e-16   1.0000e+00
```

$A^+A = I$ on $\mathbb{R}^2$

Application: Linear Regression

Example: Find the equation for the line $y = a + bx$ that goes through the points $(2, 5)$ and $(4, 11)$.

Answer: $a + 2 \cdot b = 5$, $a + 4 \cdot b = 11$ $\Longleftrightarrow \begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \end{pmatrix}$

```
>> [1 2; 1 4] \ [5; 11]

ans =

    -1
     3
```

$\Rightarrow \boxed{y = -1 + 3x}$

Observe: $\begin{pmatrix} 1 & 2 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 5 \\ 11 \end{pmatrix}$

all 1s, to account for the affine shift a

x-values

y-values

Example: Find the equation for the line $y = a + bx$ that goes through the points

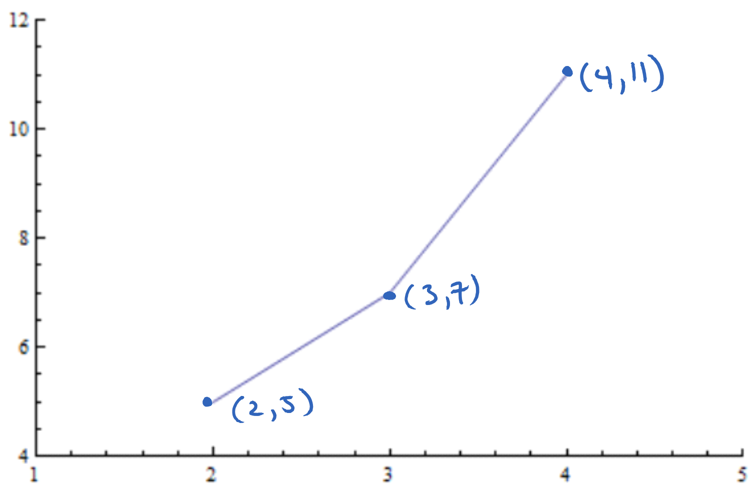
$(2,5)$

$(3,7)$

$(4,11)$.

Answer: There is no such line!

```
ListPlot[{{2, 5}, {3, 7}, {4, 11}},
 PlotRange → {{1, 5}, {4, 12}}, Joined → True]
```



Let's try setting it up anyway...

$$\begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 5 \\ 7 \\ 11 \end{pmatrix}$$

... an infeasible set of equations

But we can minimize the squared error with

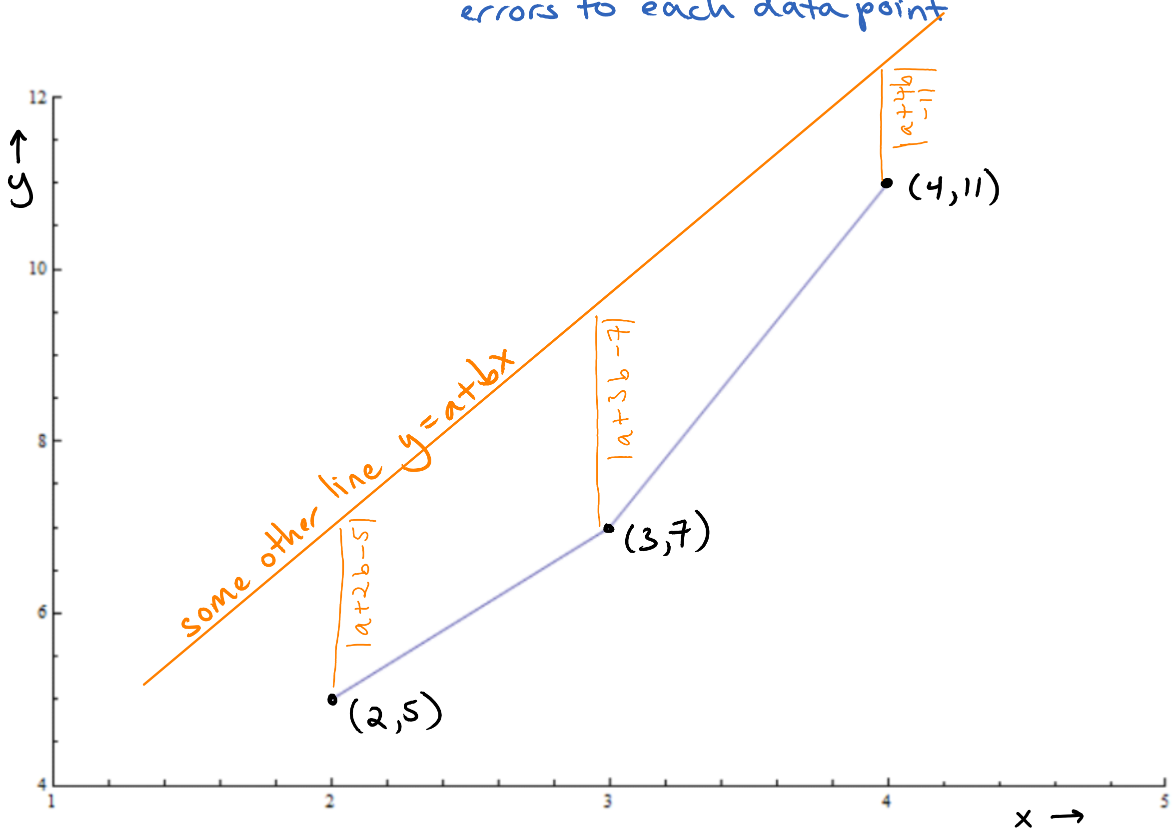
```
>> pinv([1 2; 1 3; 1 4]) * [5;7;11]

ans =

   -1.3333
    3.0000
```

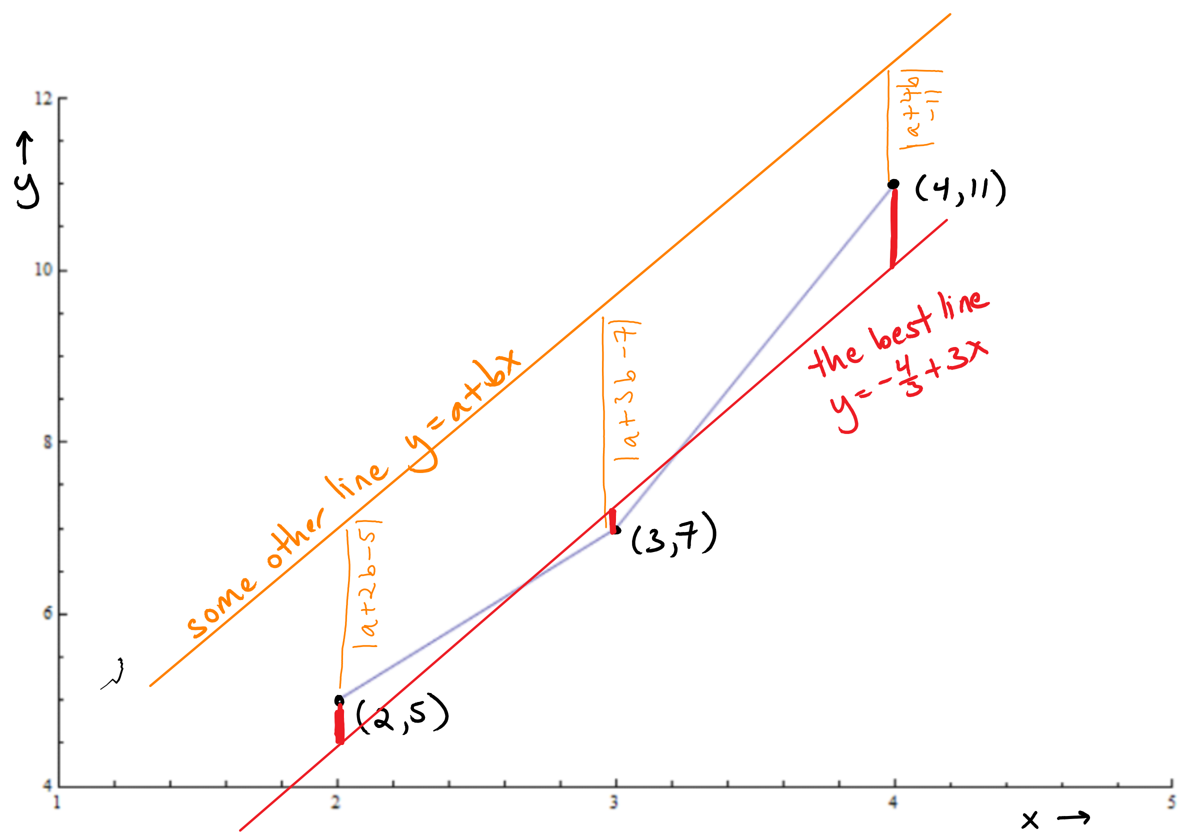
$\Rightarrow (a,b) = (-\frac{4}{3}, 3)$ minimizes $\left\| \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} - \begin{pmatrix} 5 \\ 7 \\ 11 \end{pmatrix} \right\|^2$

$$= |a + 2b - 5|^2$$
$$+ |a + 3b - 7|^2$$

$+ |a+4b-11|^2$

the sum of the squared errors to each data point



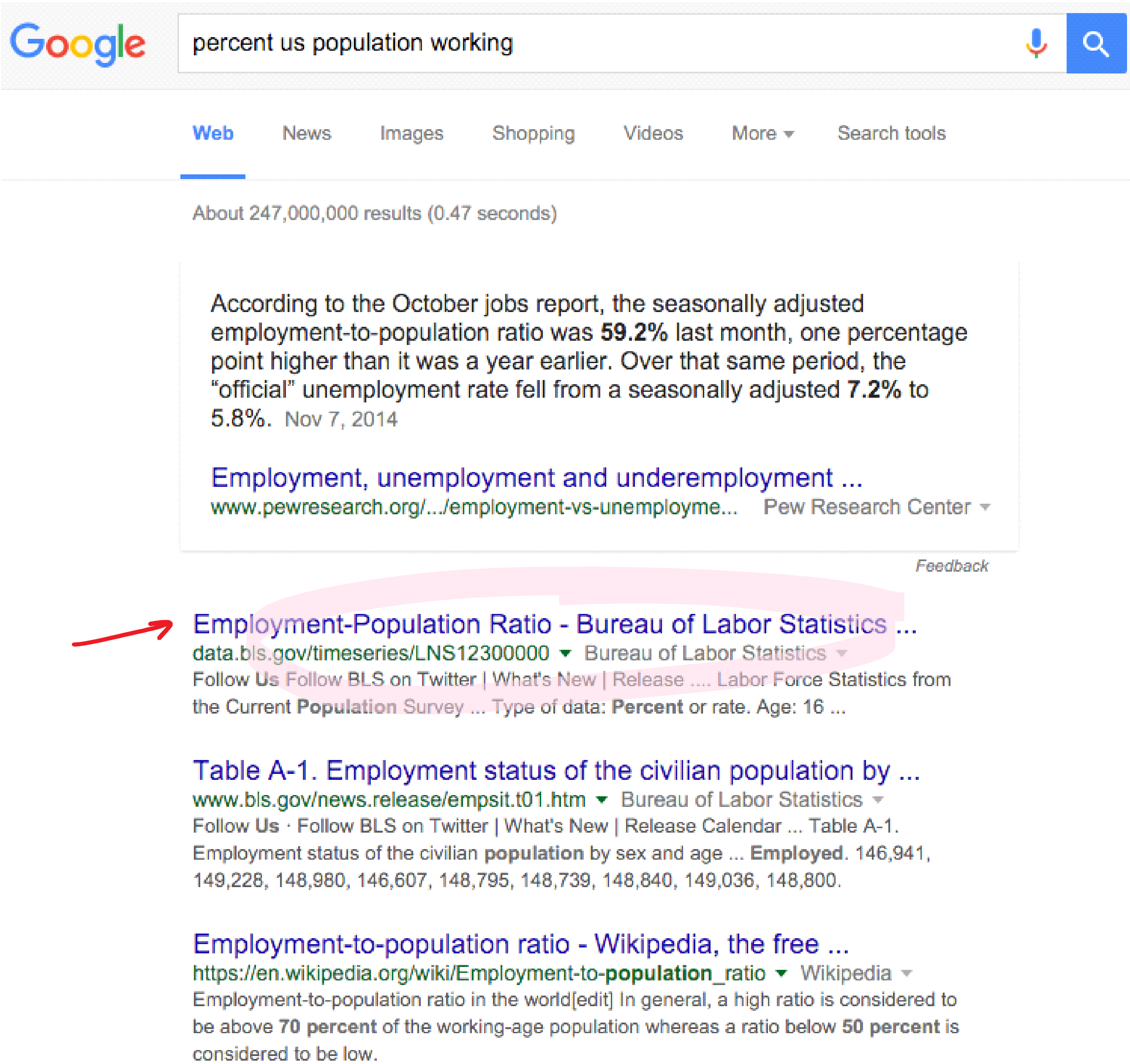


Example:

Predict what percent of Americans 16 or older will be employed in 2050.

Answer:

1. Get the data.



percent us population working

Web News Images Shopping Videos More Search tools

About 247,000,000 results (0.47 seconds)

According to the October jobs report, the seasonally adjusted employment-to-population ratio was **59.2%** last month, one percentage point higher than it was a year earlier. Over that same period, the "official" unemployment rate fell from a seasonally adjusted **7.2%** to 5.8%. Nov 7, 2014

Employment, unemployment and underemployment ...
www.pewresearch.org/.../employment-vs-unemployme... Pew Research Center

Feedback

Employment-Population Ratio - Bureau of Labor Statistics ...
data.bls.gov/timeseries/LNS12300000 Bureau of Labor Statistics
Follow **Us** Follow BLS on Twitter | What's New | Release .... Labor Force Statistics from the Current **Population** Survey ... Type of data: **Percent** or rate. Age: 16 ...

Table A-1. Employment status of the civilian population by ...
www.bls.gov/news.release/empsit.t01.htm Bureau of Labor Statistics
Follow **Us** · Follow BLS on Twitter | What's New | Release Calendar ... Table A-1. Employment status of the civilian **population** by sex and age ... **Employed**. 146,941, 149,228, 148,980, 146,607, 148,795, 148,739, 148,840, 149,036, 148,800.

Employment-to-population ratio - Wikipedia, the free ...
https://en.wikipedia.org/wiki/Employment-to-**population**_ratio Wikipedia
Employment-to-population ratio in the world[edit] In general, a high ratio is considered to be above **70 percent** of the working-age population whereas a ratio below **50 percent** is considered to be low.

UNITED STATES DEPARTMENT OF LABOR

BUREAU OF LABOR STATISTICS

Home | Subjects | Data Tools | Publications | Economic Releases | Students | Beta

Databases, Tables & Calculators by Subject

Change Output Options: From: 2005 To: 2015 GO

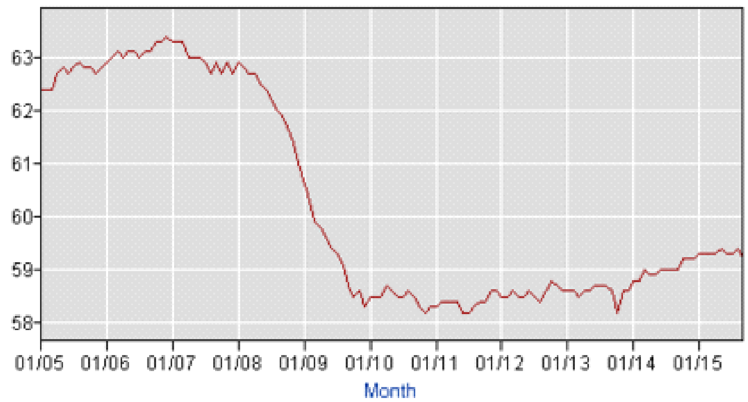
☑ include graphs ☐ include annual averages

1948

Data extracted on: October 27, 2015 (10:44:11 AM)

Labor Force Statistics from the Current Population Survey

```
Series Id:           LNS12300000
Seasonally Adjusted
Series title:        (Seas) Employment-Population Ratio
Labor force status:  Employment-population ratio
Type of data:        Percent or rate
Age:                 16 years and over
```



Download: xlsx

Year	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2005	62.4	62.4	62.4	62.7	62.8	62.7	62.8	62.9	62.8	62.8	62.7	62.8
2006	62.9	63.0	63.1	63.0	63.1	63.1	63.0	63.1	63.1	63.3	63.3	63.4
2007	63.3	63.3	63.3	63.0	63.0	63.0	62.9	62.7	62.9	62.7	62.9	62.7
2008	62.9	62.8	62.7	62.7	62.5	62.4	62.2	62.0	61.9	61.7	61.4	61.0
2009	60.6	60.3	59.9	59.8	59.6	59.4	59.3	59.1	58.7	58.5	58.6	58.3
2010	58.5	58.5	58.5	58.7	58.6	58.5	58.5	58.6	58.5	58.3	58.2	58.3
2011	58.3	58.4	58.4	58.4	58.4	58.2	58.2	58.3	58.4	58.4	58.6	58.6
2012	58.5	58.5	58.6	58.5	58.5	58.6	58.5	58.4	58.6	58.8	58.7	58.6

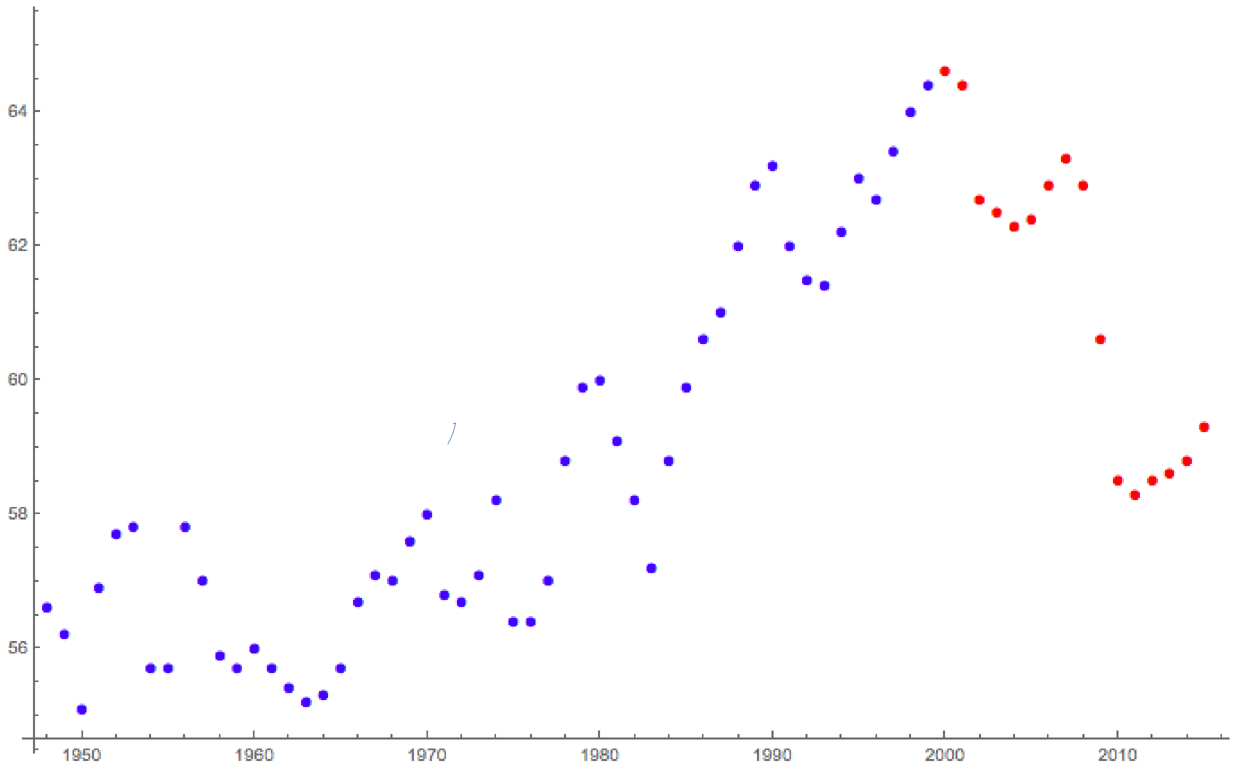
② Load the data into Mathematica / Matlab, and set up the matrices

```
data = Import["employment-population.csv"]
plot1 = ListPlot[data[[ ;; -17]], PlotMarkers → {Automatic, Small}, PlotStyle → Blue];
plot2 = ListPlot[data[[-16 ;;]], PlotMarkers → {Automatic, Small}, PlotStyle → Red];
plot = Show[plot1, plot2, PlotRange → {{1948, 2015}, {55, 65}}]
```

```
{{1948, 56.6}, {1949, 56.2}, {1950, 55.1}, {1951, 56.9}, {1952, 57.7}, {1953, 57.8}, {1954, 55.7}, {1955, 55.7}, {1956, 57.8}, {1957, 57.
 {1958, 55.9}, {1959, 55.7}, {1960, 56.}, {1961, 55.7}, {1962, 55.4}, {1963, 55.2}, {1964, 55.3}, {1965, 55.7}, {1966, 56.7}, {1967, 57.1
 {1968, 57.}, {1969, 57.6}, {1970, 58.}, {1971, 56.8}, {1972, 56.7}, {1973, 57.1}, {1974, 58.2}, {1975, 56.4}, {1976, 56.4}, {1977, 57.},
 {1978, 58.8}, {1979, 59.9}, {1980, 60.}, {1981, 59.1}, {1982, 58.2}, {1983, 57.2}, {1984, 58.8}, {1985, 59.9}, {1986, 60.6}, {1987, 61.}
 {1988, 62.}, {1989, 62.9}, {1990, 63.2}, {1991, 62.}, {1992, 61.5}, {1993, 61.4}, {1994, 62.2}, {1995, 63.}, {1996, 62.7}, {1997, 63.4},
 {1998, 64.}, {1999, 64.4}, {2000, 64.6}, {2001, 64.4}, {2002, 62.7}, {2003, 62.5}, {2004, 62.3}, {2005, 62.4}, {2006, 62.9},
 {2007, 63.3}, {2008, 62.9}, {2009, 60.6}, {2010, 58.5}, {2011, 58.3}, {2012, 58.5}, {2013, 58.6}, {2014, 58.8}, {2015, 59.3}}
```

```
data = Import["employment-population.csv"]
plot1 = ListPlot[data[[ ;; -17]], PlotMarkers → {Automatic, Small}, PlotStyle → Blue];
plot2 = ListPlot[data[[-16 ;;]], PlotMarkers → {Automatic, Small}, PlotStyle → Red];
plot = Show[plot1, plot2, PlotRange → {{1948, 2015}, {55, 65}}]
```

```
{{1948, 56.6}, {1949, 56.2}, {1950, 55.1}, {1951, 56.9}, {1952, 57.7}, {1953, 57.8}, {1954, 55.7}, {1955, 55.7}, {1956, 57.8}, {1957, 57.
 {1958, 55.9}, {1959, 55.7}, {1960, 56.}, {1961, 55.7}, {1962, 55.4}, {1963, 55.2}, {1964, 55.3}, {1965, 55.7}, {1966, 56.7}, {1967, 57.1
 {1968, 57.}, {1969, 57.6}, {1970, 58.}, {1971, 56.8}, {1972, 56.7}, {1973, 57.1}, {1974, 58.2}, {1975, 56.4}, {1976, 56.4}, {1977, 57.},
 {1978, 58.8}, {1979, 59.9}, {1980, 60.}, {1981, 59.1}, {1982, 58.2}, {1983, 57.2}, {1984, 58.8}, {1985, 59.9}, {1986, 60.6}, {1987, 61.}
 {1988, 62.}, {1989, 62.9}, {1990, 63.2}, {1991, 62.}, {1992, 61.5}, {1993, 61.4}, {1994, 62.2}, {1995, 63.}, {1996, 62.7}, {1997, 63.4},
 {1998, 64.}, {1999, 64.4}, {2000, 64.6}, {2001, 64.4}, {2002, 62.7}, {2003, 62.5}, {2004, 62.3}, {2005, 62.4}, {2006, 62.9},
 {2007, 63.3}, {2008, 62.9}, {2009, 60.6}, {2010, 58.5}, {2011, 58.3}, {2012, 58.5}, {2013, 58.6}, {2014, 58.8}, {2015, 59.3}}
```




```
A = data[[-16 ;;, 1]];
A = {1, #} & /@ A
b = data[[-16 ;;, 2]]
```

insert 1's in each row

```
{{1, 2000}, {1, 2001}, {1, 2002}, {1, 2003}, {1, 2004}, {1, 2005}, {1, 2006}, {1, 2007},
 {1, 2008}, {1, 2009}, {1, 2010}, {1, 2011}, {1, 2012}, {1, 2013}, {1, 2014}, {1, 2015}}

{64.6, 64.4, 62.7, 62.5, 62.3, 62.4, 62.9, 63.3, 62.9, 60.6, 58.5, 58.3, 58.5, 58.6, 58.8, 59.3}
```

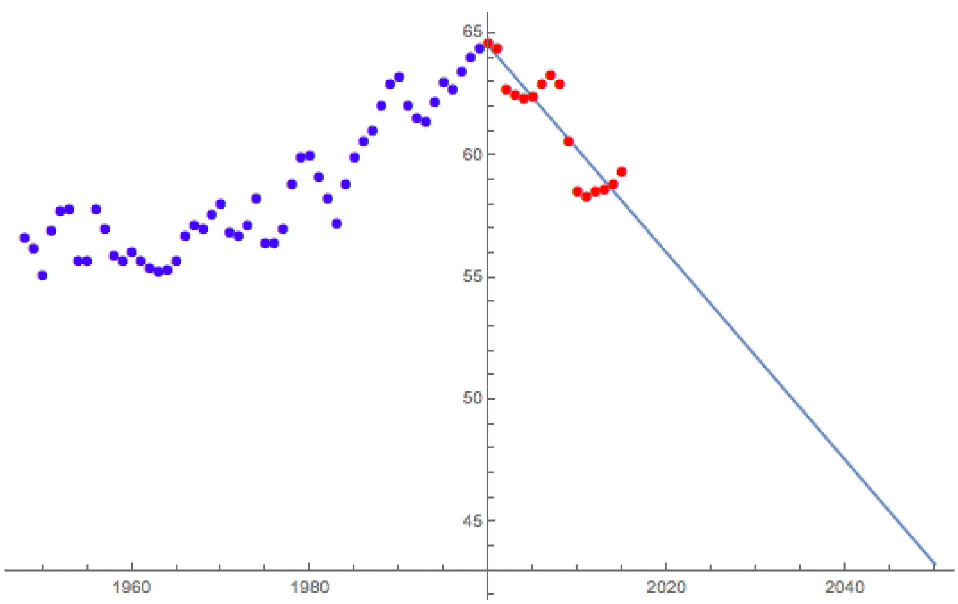
③ Compute the best-fitting line

```
{intercept, slope} = PseudoInverse[A].b

{912.113, -0.423824}
```

④ Evaluate it, and get the answer!

```
plot3 = Plot[intercept + slope x, {x, 2000, 2050}];
plot4 = Show[plot3, plot, PlotRange → {{2000, 2015}, Automatic}]
```


```
Show[plot4, PlotRange → {{1948, 2050}, Automatic}]
```




```
intercept + slope 2050

intercept + slope 1900

intercept + slope 2200
43.275
106.849
-20.2985
```

43.275 in 2050

106.849 in 1900

-20.2985 in 2200

Extrapolations often don't work well. :)

LET'S GENERALIZE!

Setting: m data points

$$\left(x_1^{(1)}, x_2^{(1)}, x_3^{(1)}, \dots, x_k^{(1)}, y^{(1)}\right)$$
$$\left(x_1^{(2)}, x_2^{(2)}, \dots, x_k^{(2)}, y^{(2)}\right)$$
$$\vdots$$
$$\underbrace{\left(x_1^{(m)}, x_2^{(m)}, \dots, x_k^{(m)}\right.}_{\substack{\text{components known exactly,} \\ \text{e.g., date/time}}}, \underbrace{\left.y^{(m)}\right)}_{\substack{\text{component that we} \\ \text{want to predict}}}$$

Goal: Find the best linear predictor for y,

$$a_0, a_1, a_2, \dots, a_k \in \mathbb{R}$$

to minimize total squared error.

$$\sum_{j=1}^{m} \left| a_0 + a_1 x_1^{(j)} + \dots + a_k x_k^{(j)} - y^{(j)} \right|^2$$

Answer:

$$\underbrace{\begin{pmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdots & x_k^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdots & x_k^{(2)} \\ 1 & x_1^{(3)} & x_2^{(3)} & \cdots & x_k^{(3)} \\ & & \vdots & & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdots & x_k^{(m)} \end{pmatrix}}_{= A} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix} = \underbrace{\begin{pmatrix} y^{(1)} \\ y^{(2)} \\ y^{(3)} \\ \vdots \\ y^{(m)} \end{pmatrix}}_{= \vec{y}}$$

The least-squares solution is $\underset{\substack{\uparrow \\ \text{pseudoinverse}}}{A^+}\vec{y}$.

Example:

Consider the time (T) it takes for a runner to complete a marathon (26 miles and 385 yards). Many factors such as height, weight, age, previous training, etc. can influence an athlete's performance, but experience has shown that the following three factors are particularly important:

$$x_1 = \text{Ponderal index} = \frac{\text{height (in.)}}{[\text{weight (lbs.)}]^{\frac{1}{3}}},$$

$$x_2 = \text{Miles run the previous 8 weeks},$$

$$x_3 = \text{Age (years)}.$$

A linear model hypothesizes that the time T (in minutes) is given by $T = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \varepsilon$, where ε is a random function accounting for all other factors and whose mean value is assumed to be zero. On the basis of the five observations given below, estimate the expected marathon time for a 43-year-old runner of height 74 in., weight 180 lbs., who has run 450 miles during the previous eight weeks.

T	x_1	x_2	x_3
181	13.1	619	23
193	13.5	803	42
212	13.8	207	31
221	13.1	409	38
248	12.5	482	45

What is your personal predicted mean marathon time?

Answer: Start with a sanity check: $\alpha_1 < 0$, $\alpha_2 < 0$, $\alpha_3 > 0$

① Enter the data

```
>> A = [
1 13.1 619 23 ;
1 13.5 803 42 ;
1 13.8 207 31 ;
1 13.1 409 38 ;
1 12.5 482 45
]; x1  x2  x3
>> b = [
181; 193; 212; 221; 248
];
```

times in minutes

② Find best-fit plane

```
>> alpha = pinv(A) * b

alpha =

  492.0442
  -23.4355
   -0.0761
    1.8624
```

2 miles more/week = 1 min. faster

every year older = ~2 minutes slower

③ Predict!

```
                                      x1          x2    x3
>> alpha' * [1; 74/180^(1/3); 450; 43]

ans =

  230.7209
```

predicted time = 3 hours 50 minutes

Example (continued): Same problem, but now fit a quadratic curve to x_2 (distance over previous 8 weeks) — running too much slows you down?

Answer: Want $T = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_2^2$

① Enter the data & find best-fit hyperplane

```
>> A2 = [A, A(:,3).^2];
alpha2 = pinv(A2) * b

alpha2 =

  740.7697
  -38.7075
   -0.2441
```

② Predict!

```
>> alpha2' * [1; 74/180^(1/3); 450; 43; 450^2]

ans =

  224.3594
```

= 3 hours 44 minutes

```
740.7697
-38.7075
-0.2441
1.5801
0.0002
```

```
ans =

224.3594
```
= 3 hours 44 minutes

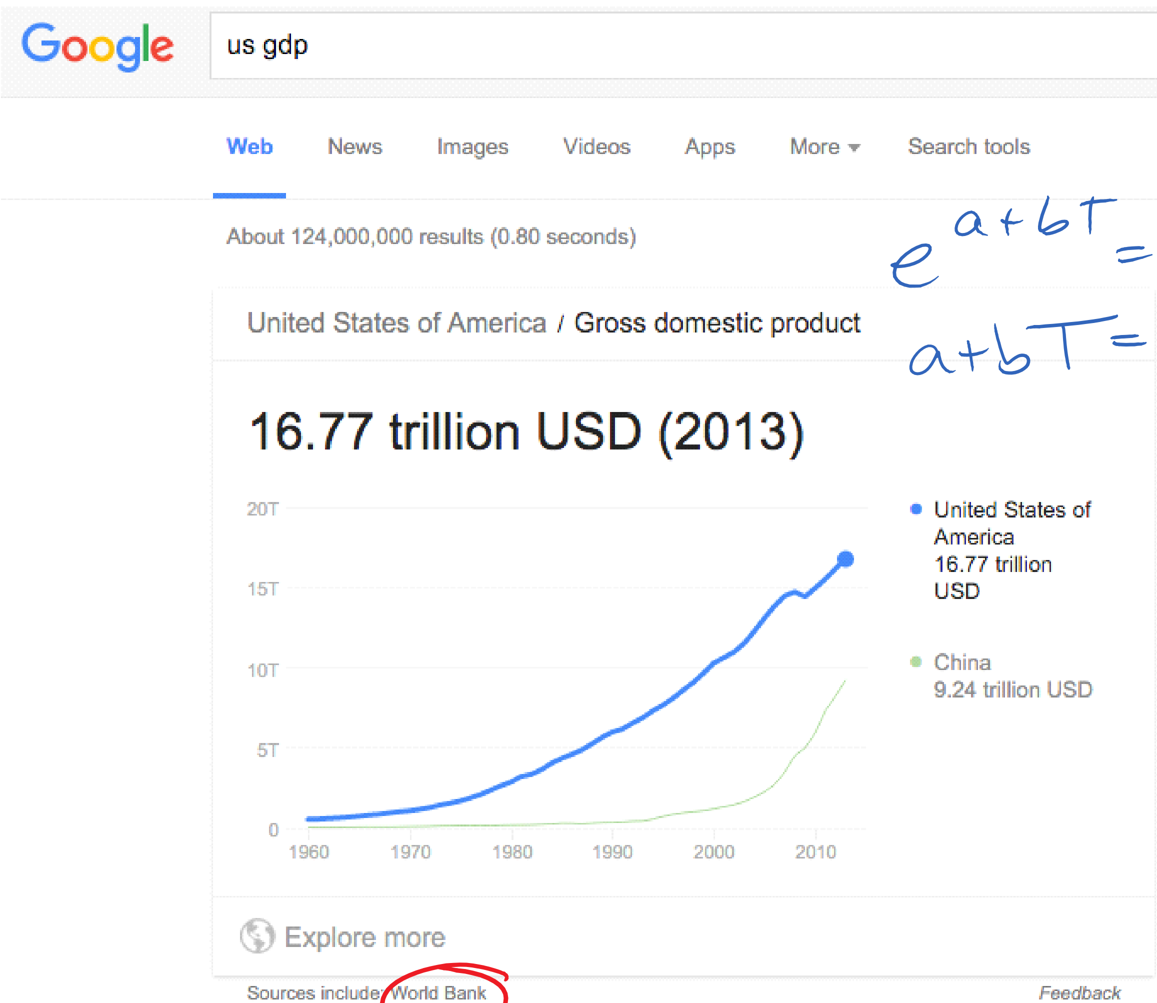
(Of course, be careful of overfitting the data by using too many parameters.)

Fitting other curves than lines:

It works just like the previous example!

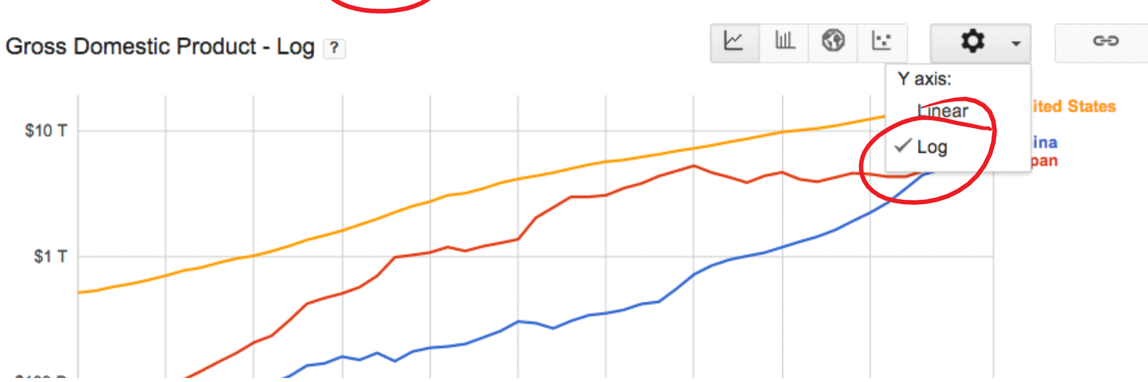
Example: Predict US gross domestic product (GDP) in 2050.

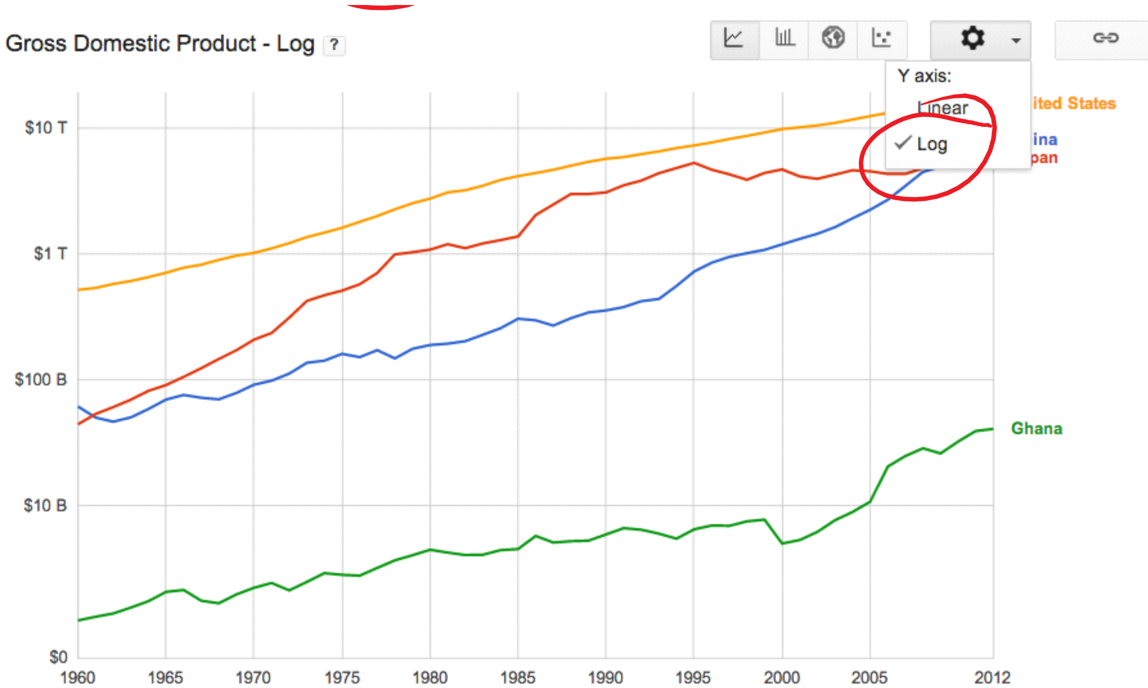
Answer:



$$e^{a+bT} = G$$

$$a + bT = \log G$$





An exponential should fit the data better than a straight line.

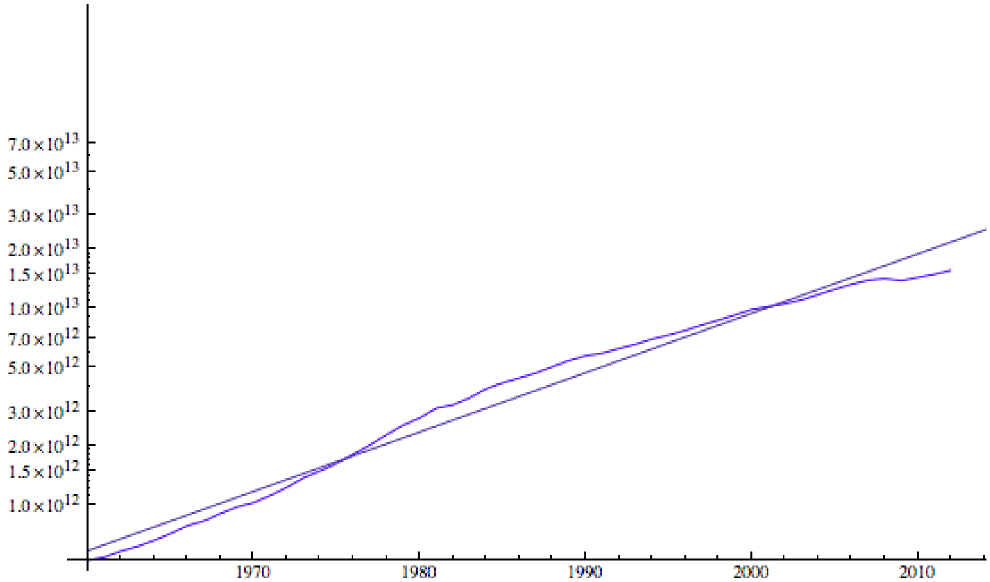
```
USGDPdata = [1960, 5.20531e11; 1961, 5.39051e11; 1962, 5.79748e11; 1963, 6.1167e11; 1964, 6.56912e11;
1965, 7.12082e11; 1966, 7.80761e11; 1967, 8.25056e11; 1968, 9.01456e11; 1969, 9.73385e11; 1970,
1.0248e12; 1971, 1.1131e12; 1972, 1.225e12; 1973, 1.3693e12; 1974, 1.4859e12; 1975, 1.6234e12; 1976,
1.8091e12; 1977, 2.0136e12; 1978, 2.276e12; 1979, 2.5435e12; 1980, 2.7675e12; 1981, 3.1038e12; 1982,
3.2277e12; 1983, 3.5069e12; 1984, 3.9004e12; 1985, 4.1848e12; 1986, 4.425e12; 1987, 4.6989e12; 1988,
5.0619e12; 1989, 5.4397e12; 1990, 5.7508e12; 1991, 5.9307e12; 1992, 6.2618e12; 1993, 6.5829e12; 1994,
6.9933e12; 1995, 7.3384e12; 1996, 7.7511e12; 1997, 8.2565e12; 1998, 8.741e12; 1999, 9.301e12; 2000,
9.8988e12; 2001, 1.02339e13; 2002, 1.05902e13; 2003, 1.10893e13; 2004, 1.17978e13; 2005, 1.25643e13;
2006, 1.33145e13; 2007, 1.39618e13; 2008, 1.42193e13; 2009, 1.38983e13; 2010, 1.44194e13; 2011,
1.49913e13; 2012, 1.56848e13];
```

```
>> years = USGDPdata(:,1);
A = [ones(length(years),1), years];
USFit = pinv(A) * log(USGDP)

USFit =

    -1.096282320582321e+02
     6.975400397258215e-02
```

$$\begin{pmatrix} 1 & 1960 \\ 1 & 1961 \\ 1 & 1962 \\ & \vdots \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} \log(5.2\times10^{11}) \\ \log(5.4\times10^{11}) \\ \log(5.8\times10^{11}) \\ \vdots \end{pmatrix}$$

```
>> exp(USFit' * [1; 2050])

ans =

     3.099636305012074e+14
```

= 30 trillion dollars



More practical ways of computing the best fit:

Meyer p.423: **Caution!** Generalized inverses are useful in formulating theoretical statements such as those above, but, just as in the case of the ordinary inverse, generalized inverses are not practical computational tools. In addition to being computationally inefficient, serious numerical problems result from the fact that $\mathbf{A}^{+}$ need not be a continuous functions of the entries of A.

① $A^T A x = A^T b$

② $\begin{pmatrix} I & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} y \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}$

[Problem 4.6.9, p.237 of Meyer]