Econ 196 Seminar: Quantitative Long-Run Global Economic History

J. Bradford DeLong ::

delong@econ.berkeley.edu brad.delong@gmail.com
delong@hey.com :: +1-925-708-0467

<https://bcourses.berkeley.edu/courses/1551896>

<https://braddelong.substack.com/t/quantitative-long-run-global-economic>

<https://braddelong.substack.com/p/quantitative-long-run-global-economic>

<https://substack.com/home/post/p-184363836>

- Tu 1-3 Evans 560
- Th 1-3 Zoom
- additional mandatory office hours to be arranged

Course Welcome Email:
<https://bcourses.berkeley.edu/courses/1551896/discussion_topics/7205916?is_announcement=true>

==========

==========

==========

==========

Econ 196 Seminar: Quantitative Long-Run Global Economic History

- WEEK 1: Jan 20: Overview
- WEEK 2: Jan 27: Human evolution, effective population size, & high patriarchy.
- WEEK 3: Feb 3: Poor agrarian-age Malthusian societies
- WEEK 4: Feb 10: My wife will be having rotator-cuff surgery
- WEEK 5: Feb 17: Societies of domination: land & seaborne agrarian-age empires.
- WEEK 6: Feb 24: The post-1500 Imperial-Commercial Age:
- WEEK 7: Mar 3: 1775-1875 in the "Dover Circle": The British Industrial Revolution:
- WEEK 8: Mar 10: The Post-1870 growth Explosion: quantifying the prosperity hockey stock.
- WEEK 9: Mar 17: Reversals of Fortune: colonial origins & cross-nation inequality.
- WEEK 10: Mar 31: The development of underdevelopment
- WEEK 11: Apr 7: The current spread of wealth across nations.
- WEEK 12: Apr 14: Prospects for further growth acceleration (or deceleration)
- WEEK 13: Apr 21: Schumpeterian unbalanced growth and political economy
- WEEK 14: Apr 28: Hicks Lecture…
- WEEK 15: May 5: The Fermi paradox: & the future of humanity

---

Econ 196 Seminar: Quantitative Long-Run Global Economic History

- Grading…
- Assignments…

Introductory Digression: Latin: *Ars*—skill. *Liber*—free.

Artes liberales—liberal arts—originally: the skills appropriate to a free person

What was a "free man", back in the old days when universities used Latin as their *lingua franca*?:

- One not bound to onerous obligation.
- Not fixed in place to obey the will of another—not a slave, or a serf.
- But also not a military-judicial aristocrat near the top of some society-of-domination hierarchy.
    - Almost always an (at least somewhat) inherited position.
- And not a possessor of guild- or craft-right, or an inheritor of property.
- Not somebody with ample societal power; conversely, not somebody with negative societal power.
- A man—almost invariably a man—who has to make their way in the world by their wits.

What are these skills appropriate for a free man?

What are these *artes liberales*, these *liberal arts*?

For someone who has to make their way in the world by their wits?

- What are the skills a university should teach such a person?
  - In the old days: *trivium*: logic, grammar, rhetoric: how to construct arguments, how to write, how to persuade in person.
  - In the old days: *quadrivium*: arithmetic (algebra), geometry (surveying), astrology (astronomy), harmony (music).
  - In the old days: preprofessional: theology, law, medicine.
  - Other subjects? History, commerce, accounting, finance, natural philosophy, moral philosophy?: folded into the ten departments.
  - Plus: learning to write a *fine chancery hand*
- What is the thread connecting all these?
  - They are ways of accessing the store of human knowledge.

The real ASI—the real Anthology Super-Intelligence

Not an "Artificial Super-Intelligence", but rather the collective creation of human minds...

...binding time and space across 5000 years and across the entire globe.

- None of us is smart enough to think through anything from first principles...
- We are a profoundly, profoundly *cultural* species—and we and our ancestors have been for more than a million years...

- As Isaac Newton said: "If I have seen farther than other men, it is because I stand on the shoulders of giants!"...
- And if Newton recognized this, how dare any of us not?...
- Hence the way to live a good life as a free man—the liberal arts we need—are how to access the storehouse that is "the best of all that has been said and thought"—and written down...
- Thus we can enrich our lives...
- Thus we can persuade employers that we will enrich them, and so they will pay us...
- The task of education, hence, is to train each of us to be a good front-end interface node for society to the real ASI—the Anthology Super-Intelligence of humanity's collective mind...

The real ASI—the real Anthology Super-Intelligence

.

.

.

The task of education, hence, is to train each of us to be a good front-end interface node for society to the real ASI—the Anthology Super-Intelligence of humanity's collective mind...

.

.

.

What happens to us if we find ourselves in a situation...

...in which we have to think things through from nearly first principles?

Briefly: we starve to death in short order:

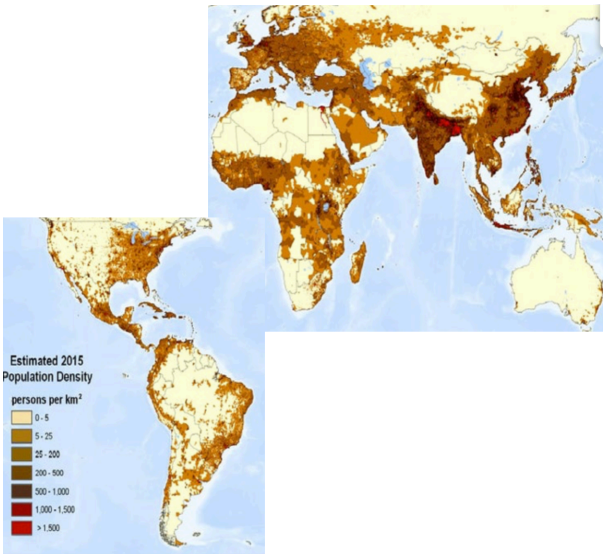
- Shlock TV show: "Naked & Afraid" <https://braddelong.substack.com/p/does-each-of-us-have-a-big-enough>
- Melissa Miller after 21 days in the Ecuadorian Amazon:
  - Not a city mouse: *an outdoors educator and nature-preserve naturalist* in Michigan.

    - Drops 17 pounds—a 2800 cal/day metabolic deficit; given her BMR of 1500 cal/day, quite a feat.
    - Her partner Chance Davis, ex-US Army Ranger, dropped 32 pounds—he was burning muscle at 1200 cal/lb, rather than fat at 3500 cal/lb.
- Before her next appearance on the show, she gained 16 fat pounds—enough to keep her going for 37 days at her BMR (or 19 at a marathon-training pace).

.

The Human Economy

Population & production *per capita*



Population:

- We have censuses, and guesstimates.

Production per capita: Necessities, conveniences & luxuries, & cultural goods

- Conceptually the question is easy: what multiple/fraction of average income at time X would make you indifferent between living then and living at the average income at time Y?
- But implementation? Hah!
- Necessities—we have, not nailed, but under controi
- Conveniences & luxuries: much harder, as problems of valuation become absurdly difficult.
- Cultural goods—positive & negative—simply impossible.

- Inequality & domination...

Other Quantities of Great Interest

Technology & inequality

Human technological capability:

- Our ability to manipulate nature and (productively) organize ourselves
- Grows as the pace of growth of production per capita plus half the pace of growth of population
  - Why not plus growth rate of population? Assumes workers not productive
  - Why not just the growth of income per capita? Assumes resources not important

Inequality:

- Within societies: a big upward leap around the year -3000
  - Bronze, writing
  - The horse, the wheel, high patriarchy
  - & full-fledged societies of domination
  - Thereafter surprisingly little action, at least not until 1800
  - Then a rise (“Gilded Age”), fall, and rise again (“Second GIlded Age”)
- Between societies: post-1800 inequality explosion

In [ ]:
```
# declare that there will be a dictionary called "pop", and assign the numbe

pop = {}
pop[2025] = 8250

pop
```

What Did I Just Do Here?

- I typed a bunch of symbols on my keyboard, pressed "return", and my doing so *changed the world*
- Who understands how my doing this changed the world?
- Who is confident that they could understand this if they were given half an hour (or less) to recall/learn something of Python?
- Why didn't I do this in Excel?

Five Origins:

- Alan Turing—extarodinary need for economy of expression
- Subsequent laziness: "Computer: use the Python computer language interpreter construct for me an empty data structure—something that I can usefully think of as a place to put "boxes", each of which will have a label, a key; and each of which will "contain" a string of symbols, a value. This data structure will be of type "dict" that the Python interpreter will then be able to manipulate in certain prescribed ways. The identifier "dict" is a mnemonic, short for "dictionary", as all that will matter is that there are a bunch of key-value pairs."
- Hellenistic philosophy on the power of words and symbols:
  - In the beginning was the Word, and the Word was with God, and the Word was divine.
  - The Word was in the beginning with God.
  - All things were made by the Word; and without the Word was not any thing made that was made.
  - In the Word was life; and the life was the light of men.
  - And that light shineth in darkness;
  - And the darkness comprehended it not.

The Last Two Origins: J.R.R. Tolkien (1):

- But though they had brought wood and kindlings by the advice of Boromir, it passed the skill of Elf or even Dwarf to strike a flame that would hold amid the swirling wind or catch in the wet fuel. At last reluctantly Gandalf himself took a hand. Picking up a faggot he held it aloft for a moment, and then with a word of command, *naur an edraith ammen*! he thrust the end of his staff into the midst of it. At once a great spout of green and blue flame sprang out, and the wood flared and sputtered. ‘If there are any to see, then I at least am revealed to them,’ he said. ‘I have written Gandalf is here in signs that all can read from Rivendell to the mouths of Anduin’...

The Last Two Origins: J.R.R. Tolkien (2):

- Something came into the chamber—I felt it through the door, and the orcs themselves were afraid and fell silent. It laid hold of the iron ring, and then it perceived me and my spell. ‘What it was I cannot guess, but I have never felt such a challenge. The counter-spell was terrible. It nearly broke me. For an instant the door left my control and began to open! I had to speak a word of Command. That proved too great a strain. The door burst in pieces.

Something dark as a cloud was blocking out all the light inside, and I was thrown backwards down the stairs. All the wall gave way, and the roof of the chamber as well, I think...

The Last Two Origins: The "Wizard Book":

- **Abelson, Harold, & Gerald Jay Sussman (with Julie Sussman)**. 1985. *The Structure and Interpretation of Computer Programs*. Cambridge, MA: MIT Press. <https://mitpress.mit.edu/9780262510363/structure-and-interpretation-of-computer-programs/>.
- We area abouit to study the idea of a computational process. Computational processes are abstract beings that inhabit computers. As they evolve, processes manipulate other abstract things called *data*. The evolution of a process is directed by a pa□ern of rules called a *program*. People create programs to direct processes. In effect, we conjure the spirits of the computer with our *spells*.
- A computational process is indeed much like a sorcerer's idea of a spirit. It cannot be seen or touched. It is not composed of ma□er at all. However, it is very real. It can perform intellectual work. It can answer questions. It can affect the world by disbursing money at a bank or by controlling a robot arm in a factory. The programs we use to conjure processes are like a sorcerer's spells. They are carefully composed from symbolic expressions in arcane and esoteric programming languages that prescribe the tasks we want our processes to perform.... Like the sorcerer's apprentice, novice programmers must learn to understand and to anticipate the consequences of their conjuring. Even small errors (usually called bugs or glitches) in programs can have complex and unanticipated consequences. Fortunately, learning to program is considerably less dangerous than learning sorcery, because the spirits we deal with are conveniently contained in a secure way...

In [ ]:
```
# fill in population guesstimates back as long as we have governments that c

pop[2000] = 6170 # relying on <http://ourworldindata.org> <https://ourworldi
pop[1975] = 4070
pop[1950] = 2530
pop[1925] = 2000
pop[1900] = 1650
pop[1875] = 1389
pop[1825] = 1100
pop[1775] = 800
pop[1700] = 610
pop[1600] = 500
pop[1500] = 450
```

```
pop
```

In [ ]:
```
# now we really are largely guessing—informed guessing, but guessing...

pop[1300] = 430
pop[800] = 200
pop[200] = 260
pop[-600] = 150
pop[-1500] = 91
pop[-3000] = 45
pop[-5000] = 19
```

In [ ]:
```
# before -5000? Oyyyyy...

pop[-8000] = 5 # <https://www.prb.org/articles/how-many-people-have-ever-liv
pop[-13000] = 4
pop[-23000] = 3
pop[-48000] = 2
pop[-73000] = 0.01 # <https://en.wikipedia.org/wiki/Prehistoric_demography>
```

In [ ]:
```
type(pop)
```

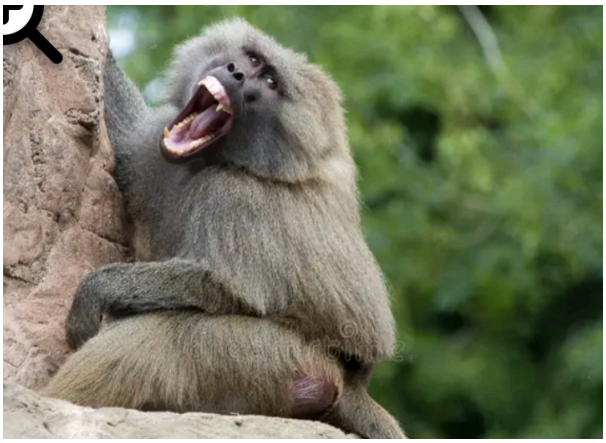
In [ ]:
```
pop
```

In [ ]:
```
import numpy as np
import pandas as pd # get libraries

human_history_df = pd.DataFrame(
                    list(pop.items()),
                    columns=['year', 'population']
                    )

human_history_df
```

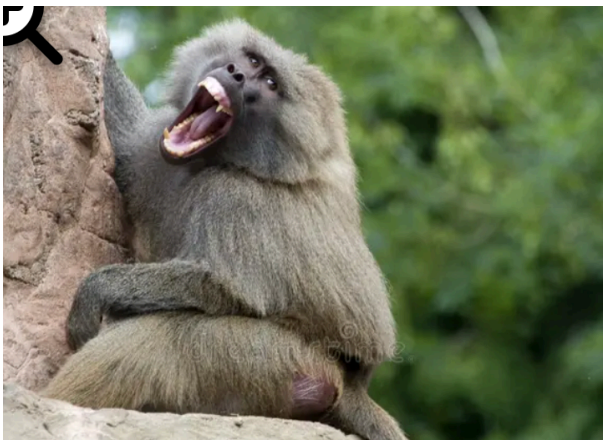
A Word About This "pop[-73000] = 0.01" Millions—That Is, 10,000 People



- The gene-science people will tell us this:
- It really looks like the overwhelming majority of us derive the overwhelming proportion of our ancestry from about 100 groups of 100 or so wandering around East-Central Africa some 75000 years ago.
- But what if we ask them a different question: how many hominins alive 75000 years ago have living descendants today?

- They will run and hide...
- Pin them down, and they will say genes are "flowing' within and between;
  - East-African *homines sapientes sapientes* population (100,000?)
  - South-African *homines sapientes sapientes* population (Khoesan: 75,000?)
  - Eurasian Central Asia and further west *homines sapientes neandertalenses* population (75000?)
  - Eurasian Central Asia and further east *homines sapientes denisovenses* population (50,000?)
  - African & Eurasian "ghost hominins" (100,000)
  - For a total of 400,000 as a **VERY ROUGH AND NOT WELL MOTIVATED GUESS**
  - By 50,000 years ago, primarily East-African *homines sapientes sapientes* lineages have spread throughout Africa and are spreading over Eurasia, and the entire population of us *homines sapientes* are shifting to primarily their genes (but with admixtures)
- Thus you could say: pop[-73000] = 0.5
- That is the group of people who have left living descendants today, and their cousins—all those whom are ancestors found attractive enough to mate with and raise children with.

A View from Outer Space 75000 Years Ago



- And then there is the larger group of *hominids*: perhaps 2,000,000 chimpanzees, bonobos, gorillas and orangutans back then in the year -73000
- We were not a terribly numerous and thus Darwinianly successful group among the great apes, back then
- Indeed: in biomass terms, baboons alone probably had all of us great-ape *hominids* beaten back then.
- Humans as *dominating* our natural environment in any real sense is quite a new thing—a 50,000-year-old thing...

In [ ]:
```
human_history_df['income'] = np.nan
human_history_df
```

In [ ]:
```
human_history_df.loc[human_history_df["year"] == 2025, "income"] = 18000

human_history_df
```

In [ ]:
```
human_history_df.loc[human_history_df["year"] == 2000, "income"] = 12000
human_history_df.loc[human_history_df["year"] == 1975, "income"] = 8900
human_history_df.loc[human_history_df["year"] == 1950, "income"] = 5800
human_history_df.loc[human_history_df["year"] == 1925, "income"] = 4400
human_history_df.loc[human_history_df["year"] == 1900, "income"] = 3400
human_history_df.loc[human_history_df["year"] == 1875, "income"] = 2200
human_history_df.loc[human_history_df["year"] == 1825, "income"] = 1800

human_history_df
```

In [ ]:
```
human_history_df.loc[human_history_df["year"] == 1775, "income"] = 1600
human_history_df.loc[human_history_df["year"] == 1700, "income"] = 1525
human_history_df.loc[human_history_df["year"] == 1600, "income"] = 1450
human_history_df.loc[human_history_df["year"] == 1500, "income"] = 1400
human_history_df.loc[human_history_df["year"] == 1300, 'income'] = 1200
human_history_df.loc[human_history_df["year"] == 800, 'income'] = 1100
human_history_df.loc[human_history_df["year"] == 200, 'income'] = 1100

human_history_df
```

In [ ]:
```
human_history_df.loc[human_history_df["year"] == -600, 'income'] = 1100
human_history_df.loc[human_history_df["year"] == -1500, 'income'] = 1100
human_history_df.loc[human_history_df["year"] == -3000, 'income'] = 1100
human_history_df.loc[human_history_df["year"] == -5000, 'income'] = 1100
human_history_df.loc[human_history_df["year"] == -8000, 'income'] = 1500
human_history_df.loc[human_history_df["year"] == -13000, 'income'] = 1500
human_history_df.loc[human_history_df["year"] == -23000, 'income'] = 1500
human_history_df.loc[human_history_df["year"] == -48000, 'income'] = 1500
human_history_df.loc[human_history_df["year"] == -73000, 'income'] = 1500

human_history_df
```

In [ ]:
```
# let us now start drawing graphs

import matplotlib.pyplot as plt

ax = human_history_df.plot(
    x="year",
    y="population",
    kind="line",
    legend=True,                          # show legend
    label="World Population (millions)"  # legend label
)

ax.set_title("World Population Over Time")   # title
ax.set_xlabel("Year")
ax.set_ylabel("Population (millions)")

plt.tight_layout()
plt.show()
```

In [ ]:
```
# a logarithmic scale: equal heights are equal multiples, and constant growt

ax = human_history_df.plot(
    x="year",
    y="population",
    kind="line",
    legend=True,                          # show legend
    label="World Population (millions)"  # legend label
)

ax.set_title("World Population Over Time: Log Scale")   # title
ax.set_xlabel("Year")
ax.set_ylabel("Population (millions)")
ax.set_yscale("log")

plt.tight_layout()
plt.show()
```

In [ ]:
```
# since the last "out of africa"

ax = human_history_df.plot(
    x="year",
    y="population",
    kind="line",
    legend=True,                          # show legend
    label="World Population (millions)"  # legend label
)

ax.set_title("World Population Over Time Since -50000: Log Scale")   # title
ax.set_xlabel("Year")
ax.set_ylabel("Population (millions)")
ax.set_yscale("log")
ax.set_xlim(left=-50000)
ax.set_xlim(right=2050)
ax.set_ylim(bottom = 1)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# since agriculture

ax = human_history_df.plot(
    x="year",
    y="population",
    kind="line",
    legend=True,                          # show legend
    label="World Population (millions)"  # legend label
)

ax.set_title("World Population Over Time Since -10000: Log Scale")   # title
ax.set_xlabel("Year")
ax.set_ylabel("Population (millions)")
ax.set_yscale("log")
ax.set_xlim(left = -10000)
ax.set_xlim(right=2050)
```

```
ax.set_ylim(bottom = 4)
ax.set_ylim(top = 10000)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# switching back to the level-scale: a hockey stick

ax = human_history_df.plot(
    x="year",
    y="population",
    kind="line",
    legend=True,                          # show legend
    label="World Population (millions)"  # legend label
)

ax.set_title("World Population Hockey Stick Since -10000")   # title
ax.set_xlabel("Year")
ax.set_ylabel("Population (millions)")
ax.set_xlim(left = -10000)
ax.set_xlim(right=2050)
ax.set_ylim(bottom = 4)
ax.set_ylim(top = 10000)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# now let us visualize income per capita, sort of

ax = human_history_df.plot(
    x="year",
    y="income",
    kind="line",
    legend=True,                          # show legend
    label="World GDP per Capita"  # legend label
)

ax.set_title("World GDP per Capita Over Time")   # title
ax.set_xlabel("Year")
ax.set_ylabel("2025-Value Dollars")
ax.set_ylim(top = 20000)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# log scale and better tick marks

import matplotlib.ticker as mticker

ax = human_history_df.plot(
    x="year",
    y="income",
    kind="line",
    legend=True,                          # show legend
    label="World GDP per Capita"  # legend label
```

```
)

ax.set_title("World GDP per Capita Over Time: Log Scale")   # title
ax.set_xlabel("Year")
ax.set_ylabel("2025-Value Dollars")
ax.set_yscale("log", base=2)
ax.yaxis.set_major_formatter(mticker.ScalarFormatter())  # <-- this is the k
ax.yaxis.set_major_formatter(mticker.ScalarFormatter())  # <-- this is the k
ax.set_ylim(top = 20000)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# how extraordinary recent the "blade" part of the productivity and standard

ax = human_history_df.plot(
    x="year",
    y="income",
    kind="line",
    legend=True,                          # show legend
    label="World GDP per Capita"  # legend label
)

ax.set_title("World GDP per Capita Hockey Stick: Log Scale")   # title
ax.set_xlabel("Year")
ax.set_ylabel("2025-Value Dollars")
ax.set_yscale("log", base=2)
ax.yaxis.set_major_formatter(mticker.ScalarFormatter())  # <-- this is the k
ax.yaxis.set_major_formatter(mticker.ScalarFormatter())  # <-- this is the k
ax.set_ylim(top = 20000)
ax.set_xlim(left = -10000)
ax.set_xlim(right = 2050)

plt.tight_layout()
plt.show()
```

In [ ]:
```
ax = human_history_df.plot(
    x="year",
    y="income",
    kind="line",
    legend=True,                          # show legend
    label="World GDP per Capita"  # legend label
)

ax.set_title("World GDP per Capita Since 1: Log Scale")   # title
ax.set_xlabel("Year")
ax.set_ylabel("2025-Value Dollars")
ax.set_yscale("log", base=2)
ax.yaxis.set_major_formatter(mticker.ScalarFormatter())  # <-- this is the k
ax.yaxis.set_major_formatter(mticker.ScalarFormatter())  # <-- this is the k
ax.set_ylim(top = 20000)
ax.set_xlim(left = 0)
ax.set_xlim(right = 2050)
```

```
plt.tight_layout()
plt.show()
```

In [ ]:
```
# time to look at growth rates

import numpy as np

# 1. Make sure there is no index/column ambiguity and you're sorted by year
human_history_df = human_history_df.reset_index(drop=True)
human_history_df = human_history_df.sort_values("year")

# 2. Year gaps
human_history_df["year_diff"] = human_history_df["year"].diff()

# 3. Annual *log* growth rates
human_history_df["pop_growth"] = (
    np.log(human_history_df["population"])
    - np.log(human_history_df["population"].shift(1))
) / human_history_df["year_diff"]

human_history_df["income_growth"] = (
    np.log(human_history_df["income"])
    - np.log(human_history_df["income"].shift(1))
) / human_history_df["year_diff"]

human_history_df
```

In [ ]:
```
# calculating "technology"—growth rate equal to income growth plus half of p

human_history_df["tech_growth"] = (
    human_history_df["income_growth"] + 0.5 * human_history_df["pop_growth"]
)

human_history_df
```

In [ ]:
```
#constructing a level index for "technology"

human_history_df = human_history_df.sort_values("year").reset_index(drop=Tru

human_history_df["ln_tech"] = np.nan

human_history_df.loc[human_history_df["year"] == -73000, "ln_tech"] = 0.0

# build ln_tech forward from the first non-NaN
base_idx = human_history_df["ln_tech"].first_valid_index()
mask_after_base = human_history_df.index > base_idx

human_history_df.loc[mask_after_base, "ln_tech"] = (
    human_history_df.loc[mask_after_base, "tech_growth"]
    * human_history_df.loc[mask_after_base, "year_diff"]
).cumsum()

human_history_df
```

In [ ]:
```
# 4. exponentiate to get a tech index (base = 1 at year -73000)

human_history_df["tech_index"] = np.exp(human_history_df["ln_tech"])

# find the tech_index value at 1875
base_value_1875 = human_history_df.loc[human_history_df["year"] == 1875, "te

# rescale the entire index so that tech_index(1875) = 1
human_history_df["tech_index"] = human_history_df["tech_index"] / base_value

human_history_df
```

In [ ]:
```
# population, income, growth rate hockey-stick

ax = human_history_df.plot(
    x="year",
    y=["pop_growth", "income_growth", "tech_growth"],
    kind="line",
    legend=True
)

ax.set_title("Annual Log Growth Rates: Population, Income, Technology")
ax.set_xlabel("Year")
ax.set_ylabel("Annual log growth rate")

plt.tight_layout()
plt.show()
```

In [ ]:
```
# looking since agriculture

ax = human_history_df.plot(
    x="year",
    y=["pop_growth", "income_growth", "tech_growth"],
    kind="line",
    legend=True
)

ax.set_title("Annual Log Growth Rates Since Agriculture: Population, Income,
ax.set_xlabel("Year")
ax.set_ylabel("Annual log growth rate")

ax.set_xlim(left = -10000)
ax.set_xlim(right = 2050)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# looking since bronze & writing...

ax = human_history_df.plot(
    x="year",
    y=["pop_growth", "income_growth", "tech_growth"],
    kind="line",
    legend=True
```

```
)

ax.set_title("Annual Log Growth Rates Since Bronze & Writing: Population, Ir
ax.set_xlabel("Year")
ax.set_ylabel("Annual log growth rate")

ax.set_xlim(left = -3000)
ax.set_xlim(right = 2050)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# since year 1

ax = human_history_df.plot(
    x="year",
    y=["pop_growth", "income_growth", "tech_growth"],
    kind="line",
    legend=True
)

ax.set_title("Annual Log Growth Rates Since Year 1: Population, Income, Tech
ax.set_xlabel("Year")
ax.set_ylabel("Annual log growth rate")

ax.set_xlim(left = 0)
ax.set_xlim(right = 2050)

plt.tight_layout()
plt.show()
```

In [ ]:
```
# select rows from -1500 on, and only the columns you care about

table = human_history_df.loc[
    7:,
    ["year", "population", "income", "tech_index",
     "pop_growth", "income_growth", "tech_growth"]
]

table
```

In [ ]:
```
# select an even smaller set of rows...

years_to_keep = [-1500, 800, 1600, 1775, 1875, 2025]

subset = human_history_df.loc[
    human_history_df["year"].isin(years_to_keep),
    ["year", "population", "income", "tech_index",
     "pop_growth", "income_growth", "tech_growth"]
]

subset
```

Comprehending the MacroStructrure of Post-Bronze Post-Writing Human History

- As of

year	population	income	tech_index	pop_growth	income_growth	tech_growth
-1500	91.0	1100.0	0.127979	0.000469	0.000000	0.000235
800	200.0	1100.0	0.189729	-0.000437	0.000000	-0.000219
1600	500.0	1450.0	0.395439	0.001054	0.000351	0.000878
1775	800.0	1600.0	0.551939	0.003615	0.000640	0.002448
1875	1389.0	2200.0	1.000000	0.004665	0.004013	0.006346
2025	8250.0	18000.0	19.940024	0.011621	0.016219	0.022029

  -1500: Ancient: 0.024%/year—2.4% per century...
- As of -800: The Late-Antiquity Pause: -0.022%/year—**agrarian-era antiquity technology growth is not guaranteed**
- As of 800-1600: Post-800 has shifted to a higher rate of growth than characteristic of "antiquity": 8%/century
- As of 1600-1775: Commercial-Imperial era has been in full swing for a while: 25%/century
- As of 1775-1875: The "Dover Circle" Industrial Revolution has been a thing: 90%/century
- As of 1875-2025: MEG: 2%/year; 7.4-fold/century
  - **Memo**: necessities, conveniences & luxuries, cultural goods, inequality—within & between societies—& domination

In [ ]:

In [ ]:

In [ ]:

In [ ]:

<http://datahub.berkeley.edu>

> How do you set up a berkeley datahub account if you have never had one before?

1. Activate your CalNet ID first: Make sure your CalNet ID is active and you can log into campus systems (e.g. bCourses). If CalNet isn’t working yet, DataHub won’t either.

2. Confirm you can log into bCourses: DataHub authentication runs through bCourses. If you can't get into bCourses with your `@berkeley.edu` credentials, that has to be fixed first (via bCourses/CalNet support).

3. Go to DataHub and log in once: In a browser (Chrome/Firefox/Safari), go to .

□ Click the login button if prompted.

□ You'll be redirected to CalNet/bCourses; log in with your CalNet ID.

□ The first successful login automatically creates your DataHub home directory and "account."

4. Authorize access when bCourses asks: The first time, bCourses will likely pop up something like "DataHub is requesting access to your account." Click to allow; this is how the hub knows who you are and which courses you're in.

5. Verify the environment: Once it finishes spinning up, you should land in JupyterLab on DataHub (as in the page you're looking at). At that point you're "set up":

□ You should see a small home directory in the file browser.

□ Opening a notebook or a course nbgitpuller link will now work under your account.

==========

Or if you have your system set up on your own computer, eventually you will get to a point where you issue the command:

```
python3 -m jupyter lab
```