A Online Appendix: Additional Examples

A.1 Claude 2's Referee Report on This Paper

The following reproduces Claude 2's editorial opinion on the first working paper version of this paper (Korinek, 2023):

Chat (Claude 2): Can you draft a referee report for this paper for the Journal of Economic Literature?
[Attachment: w30957.pdf]

Here is a draft referee report for the paper "Large Language Models and Cognitive Automation for Economic Research":

To the editors of the Journal of Economic Literature:

I have completed my review of the paper "Large Language Models and Cognitive Automation for Economic Research" by Anton Korinek.

This paper provides a timely and useful overview of how large language models (LLMs) like ChatGPT can be leveraged to automate and enhance various research tasks for economists. The paper is well-written and clearly presented. I recommend it for publication after minor revisions.

Summary:

The paper categorizes potential applications of LLMs into six domains - ideation, writing, background research, coding, data analysis, and mathematical derivations. For each domain, the author provides general guidance on prompt engineering as well as specific examples to showcase the capabilities and limitations of LLMs. Based on experiments using the GPT-3 model, the author assigns a usefulness rating to each application ranging from experimental (inconsistent results requiring oversight) to highly useful (will likely save researcher time).

The paper concludes by predicting how LLMs may affect economic research productivity in the short and long run. In the short run, the author argues LLMs will act as useful assistants to automate micro-tasks, increasing researcher productivity. In the long run, the author speculates LLMs may produce superior economic models, potentially displacing human economists. The paper ends by proposing new research questions economists should explore regarding the impacts of cognitive automation.

Comments:

The paper succeeds in its aim of providing an accessible introduction and framework for economists to begin leveraging LLMs in their work. The catalog of potential applications is comprehensive and the examples are illuminating. Prompt engineering is crucially important to effectively using LLMs, and the author helpfully includes prompts for replicability.

I have only minor recommendations to further strengthen the paper:

- Given the speed of developments in this area, mentioning the timing of experiments and capabilities of the specific LLM used would help frame the generalizability of the results.
- The usefulness ratings are subjective. Surveying researchers who have used LLMs may produce more robust assessments.
- Additional guidance on the factors that contribute to inconsistent vs. highly useful capabilities could better set expectations.
- Exploring collaborative applications where LLMs complement humans would enrich the long-run outlook.

Overall, I believe this paper makes a valuable contribution in a rapidly evolving area. It will serve as a helpful guide to researchers looking to exploit LLMs to improve their productivity. I recommend publication after minor revisions.

Please let me know if I can provide any clarification or additional feedback.

Sincerely,
[Your name here]

It is well known that Claude 2 is programmed to be friendly and upbeat. To evaluate whether it would provide an equally positive assessment for other journals, I continued with the following prompt:

Chat (Claude 2) ctd.: Would this paper be suitable for the American Economic Review?

Based on my review, I do not think this paper would be suitable for publication in the American Economic Review in its current form. Here is my assessment:

The American Economic Review is one of the most prestigious and selective economics journals, publishing theoretical and empirical research that makes significant advancements in economic analysis and knowledge.

While this paper provides a useful and timely overview of how economists can apply large language models in their research, it is more expository and demonstrative rather than presenting novel research or analysis. As such, it may not rise to the level of contributing major new economic insights demanded by AER.

Some factors to consider regarding AER suitability:

- The paper catalogs potential LLM applications but does not rigorously evaluate their impact on economic research or productivity. An empirical analysis demonstrating and quantifying the benefits would strengthen it.

- There is limited theoretical or conceptual economic analysis. Expanding on implications for economic theory and modeling could improve fit for AER.

- The mathematical derivations section is quite limited in coverage. More technical treatment would be expected for AER.

- Long-run predictions about LLMs surpassing human abilities are interesting but speculative. Linking these to formal economic growth models would make them more rigorous.

With additional empirical analysis, technical depth, and formal economic modeling, this line of research could potentially yield papers suitable for AER. However, in its current form as an applications overview, the paper may not satisfy AER's standards for novelty and rigor. Journals focused on applications like the Journal of Economic Perspectives may be better fits.

I hope these comments are useful. Please let me know if you would like me to expand on my assessment of the paper's suitability for AER. I'm happy to provide additional feedback.

A.2 Simulating the Solow Model

The following prompt uses ChatGPT Advanced Data Analysis to simulate the Solow model:

Chat (ChatGPT Advanced Data Analysis): Simulate the Solow growth model and plot the results in a four chart.

Alright, let's simulate the Solow growth model. To simplify things, let's consider the Solow-Swan model without technological progress. The model describes the evolution of capital (K), labor (L), and output (Y) in an economy. The basic equation for the change in capital stock over time is given by:

$$\frac{dK}{dt} = sY - \delta K$$

where:

- s is the savings rate

- Y is the total output, which is given by a production function, usually a Cobb-Douglas type: $Y = K^\alpha L^{1-\alpha}$,

- δ is the depreciation rate.

Assumptions for the model:

- Labor grows at a constant rate n, i.e., $L(t) = L(0)e^{nt}$,
- The savings rate s, the depreciation rate δ, and the labor growth rate n are all constants.

To get started, we need to specify the values of these parameters. For example, we can use the following:

- $s = 0.4$,
- $\delta = 0.1$,
- $n = 0.02$,
- $\alpha = 0.33$,
- initial capital stock $K(0) = 1$,
- initial labor $L(0) = 1$.

To simulate the model, we'll use the Euler method for integrating the differential equations. This method approximates the value of the function at a future time as the value at the current time plus the derivative of the function at the current time multiplied by the time step.
Then, we will plot the following four charts:

1. Capital over time,
2. Labor over time,
3. Output over time,
4. Capital per worker over time.

Does that sound good?

Chat (ChatGPT Advanced Data Analysis) ctd.: Go ahead.

```
import numpy as np
import matplotlib.pyplot as plt


# Model parameters
```

```
s = 0.4
delta = 0.1
n = 0.02
alpha = 0.33
K0 = 1
L0 = 1
T = 100 # Total simulation time

# Time grid
dt = 0.01
times = np.arange(0, T, dt)

# Initialize arrays to store the values of K, L, Y over time
K = np.zeros(len(times))
L = np.zeros(len(times))
Y = np.zeros(len(times))

# Initial values
K[0] = K0
L[0] = L0
Y[0] = K0**alpha * L0**(1-alpha)

# Euler method
for t in range(len(times)-1):
    dK = s*Y[t] - delta*K[t]
    dL = n*L[t]
    K[t+1] = K[t] + dK*dt
    L[t+1] = L[t] + dL*dt
    Y[t+1] = K[t+1]**alpha * L[t+1]**(1-alpha)

# Capital per worker
K_L = K / L

# Create plots
fig, axs = plt.subplots(2, 2, figsize=(14, 10))

# Plot capital over time
axs[0, 0].plot(times, K)
axs[0, 0].set_title('Capital over time')
axs[0, 0].set_xlabel('Time')
axs[0, 0].set_ylabel('Capital')
```

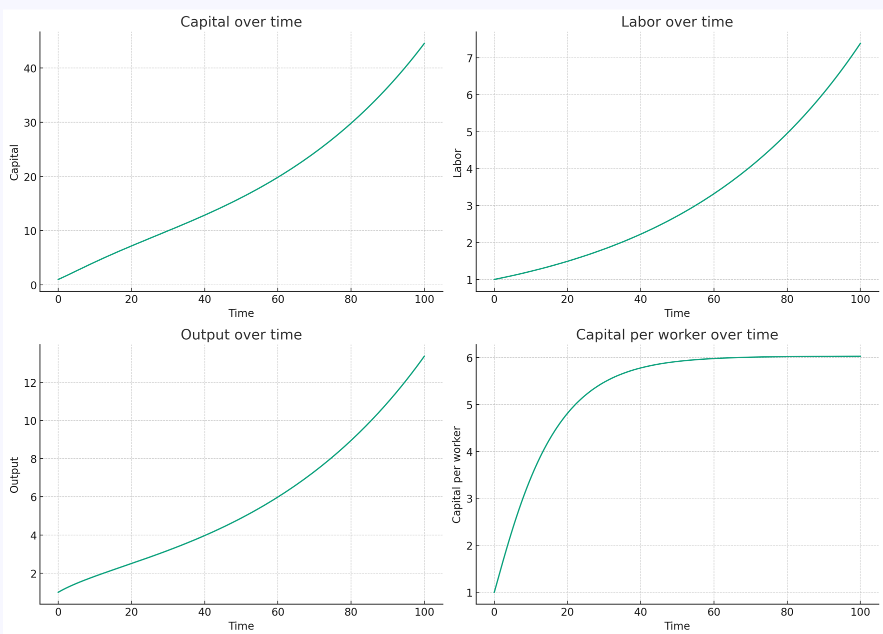
```
# Plot labor over time
axs[0, 1].plot(times, L)
axs[0, 1].set_title('Labor over time')
axs[0, 1].set_xlabel('Time')
axs[0, 1].set_ylabel('Labor')

# Plot output over time
axs[1, 0].plot(times, Y)
axs[1, 0].set_title('Output over time')
axs[1, 0].set_xlabel('Time')
axs[1, 0].set_ylabel('Output')

# Plot capital per worker over time
axs[1, 1].plot(times, K_L)
axs[1, 1].set_title('Capital per worker over time')
axs[1, 1].set_xlabel('Time')
axs[1, 1].set_ylabel('Capital per worker')

# Make the layout tight
plt.tight_layout()

plt.show()
```



Here are the four plots for the Solow growth model:

1. **Capital over time:** The capital stock increases over time and approaches a steady state. This steady state is the point at which investment (saving) equals depreciation, so the capital stock no longer changes.
   (Author's note: although the numerical simulation is correct, this point of the description is clearly wrong)

2. **Labor over time:** Labor grows exponentially at a constant rate, reflecting the assumption of constant population growth in the Solow model.

3. **Output over time:** Output also increases over time and approaches a steady state. This is because output is a function of capital and labor, which are both increasing. However, due to diminishing returns to capital, the growth of output slows down over time.

4. **Capital per worker over time:** Capital per worker also increases over time and approaches a steady state. This is the point at which the economy achieves balanced growth in the Solow model. At the steady state, output per worker and capital per worker grow at the same rate, which is the rate of technological progress. In this simplified version of the model without technological progress, this growth rate is zero, so output per worker and capital per worker are constant at the steady state.

These results illustrate the key predictions of the Solow growth model: in the long run, an economy approaches a steady state level of capital (both total and per worker), and this steady state level is determined by the savings rate, the depreciation rate, and the rate of population growth. Changes in these parameters can lead to a higher or lower steady state, but they cannot sustain long-run economic growth. According to the model, long-run economic growth can only be sustained by technological progress.

A.3 Creating figures

Since Advanced Data Analysis is not connected to the Internet, I started by asking GPT-4 to write code that downloads stock price data and saves it in a CSV file:

Chat (GPT-4): Write code to download adjusted close of GOOG, META and MSFT as well as the S&P500 index from 7/1/2014 to 6/30/2023 and save it in a CSV file.

To download the adjusted close of GOOG, META (previously FB), and MSFT as well as the S&P500 index, we can use the 'yfinance' library in Python. This library allows us to download stock data from Yahoo Finance.

Please note that the end date you provided is in the future (6/30/2023), so the code will download data up to the current date.

Here is the Python code:

```
import yfinance as yf
import pandas as pd

# Define the ticker list
tickers_list = ['GOOG', 'META', 'MSFT', '^GSPC']

# Fetch the data
data = yf.download(tickers_list, start="2014-07-01", end="2023-06-30")['Adj
Close']

# Print first 5 rows of the data
print(data.head())

# Save data to a CSV file
data.to_csv('stock_data.csv')
```

This code will download the adjusted close prices for the specified stocks and the S&P500 index from July 1, 2014, to June 30, 2023, and save it in a CSV file named 'stock_data.csv'.

Please note that you need to install the 'yfinance' library before running this code. You can install it using pip:

```
pip install yfinance
```

Also, please note that the ticker symbol for Facebook has changed to 'META' due to the company's recent rebranding. If you want to download data for Facebook before the name change, you should use 'FB' instead of 'META'.

I executed the code in python on my local computer.[18] Then I uploaded this CSV file in ChatGPT Advanced Data Analysis and prompted it as follows:

Chat (ChatGPT Advanced Data Analysis): Can you display a graph that shows the performance of these securities normalized to start at 100?
[Uploaded stock_data.csv]

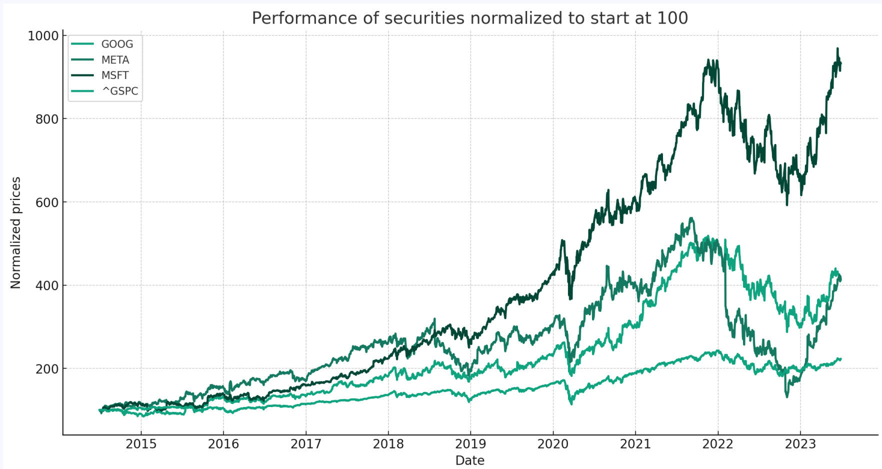
[Code generated and executed]
The dataset contains the closing prices for the securities Google (GOOG), Meta (META), Microsoft (MSFT), and the S&P 500 index (^GSPC) over a certain period of time.

Let's normalize the prices to start at 100 and plot the resulting performance of these securities. This normalization will allow us to compare the performance of the securities relative to their initial prices. To normalize the data, we'll divide

[18] ChatGPT Plus can also provide very helpful advice on questions such as "How do I install python on my computer?"

all prices by their initial price and then multiply by 100.

[Code generated and executed]
[Code generated and executed]



Here is the graph showing the performance of the securities Google (GOOG), Meta (META), Microsoft (MSFT), and the S&P 500 index (^GSPC), all normalized to start at 100. The x-axis represents time (the dates), and the y-axis represents the normalized prices.

From the graph, we can see how each of these securities has performed relative to its starting point over time. This type of comparison can be helpful when assessing the relative performance of different securities or indices.

Chat (ChatGPT Advanced Data Analysis) ctd.: Calculate the beta for each of the stocks and add it to the legend. Change the colors to make them easier to distinguish. Use the title "One-Decade Performance of Three Big Tech Companies".

The beta of a stock is a measure of the stock's volatility in relation to the overall market. The overall market has a beta of 1.0, and individual stocks are ranked according to how much they deviate from the market.

A stock that swings more than the market over time has a beta above 1.0. If a stock moves less than the market, the stock's beta is less than 1.0. High-beta stocks are supposed to be riskier but provide higher return potential; low-beta stocks pose less risk but also lower returns.

Let's calculate the beta for each of these stocks over the given period. We can calculate beta using the formula:

$$\beta = \frac{Cov(r_E, r_M)}{Var(r_M)}$$

where:

- r_E is the return of the stock
- r_M is the return of the market
- $Cov(r_E, r_M)$ is the covariance between the stock and market returns
- $Var(r_M)$ is the variance of the market returns

For the market return, we'll use the S&P 500 index (^GSPC). We'll calculate daily returns for each stock and the market, then calculate the beta for each stock.

After that, I'll recreate the graph with the new colors and the calculated betas included in the legend. The colors will be chosen to provide good contrast and clear visibility.

[Code generated and executed]

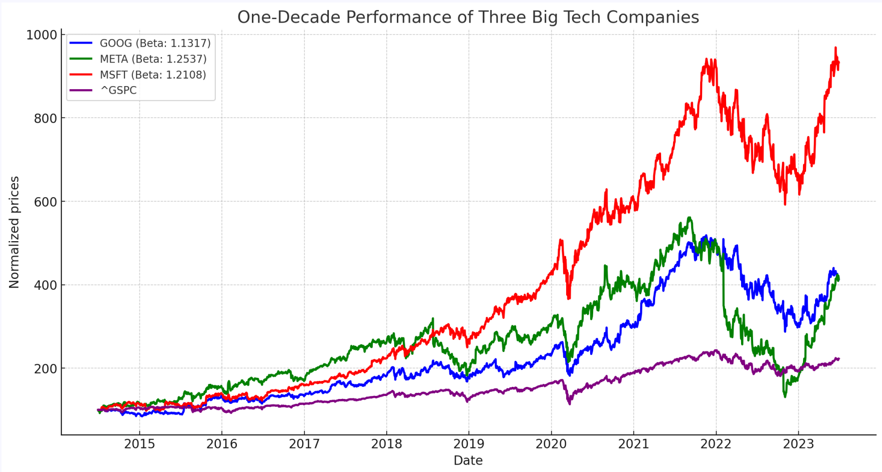
Here are the beta values for the stocks:

- Google (GOOG): 1.1317
- Meta (META): 1.2537
- Microsoft (MSFT): 1.2108

These values indicate that all three stocks are generally more volatile than the market (S&P 500), since their beta values are greater than 1.

Now let's update the graph with the new title, colors, and the beta values in the legend.

[Code generated and executed]

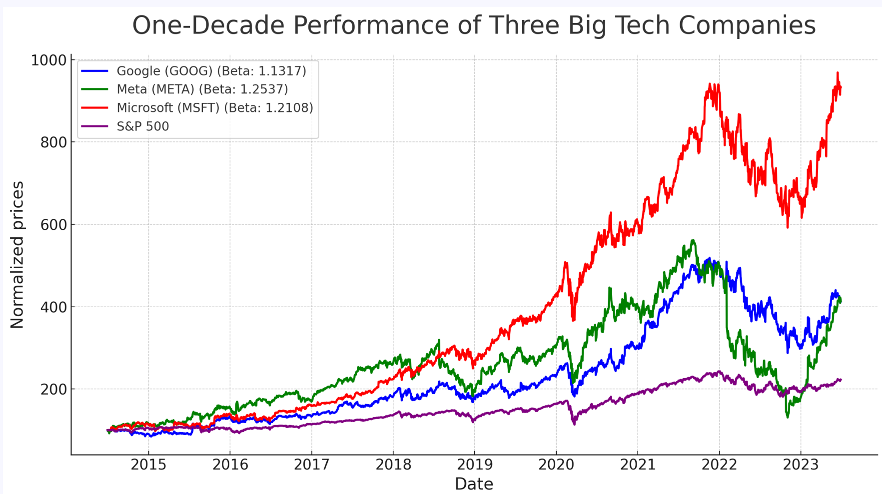


Here is the updated graph with the new title, colors, and beta values included in the legend.

The graph represents the one-decade performance of the three big tech companies: Google (GOOG), Meta (META), and Microsoft (MSFT), as well as the S&P 500 index (^GSPC), all normalized to start at 100. The beta values for the stocks, which measure their volatility in relation to the overall market, are included in the legend.

Chat (ChatGPT Advanced Data Analysis) ctd.: Double the size of the title, legend, axes labels and ticks. Label "^GSPC" in the legend as "S&P 500".

[Code generated and executed]



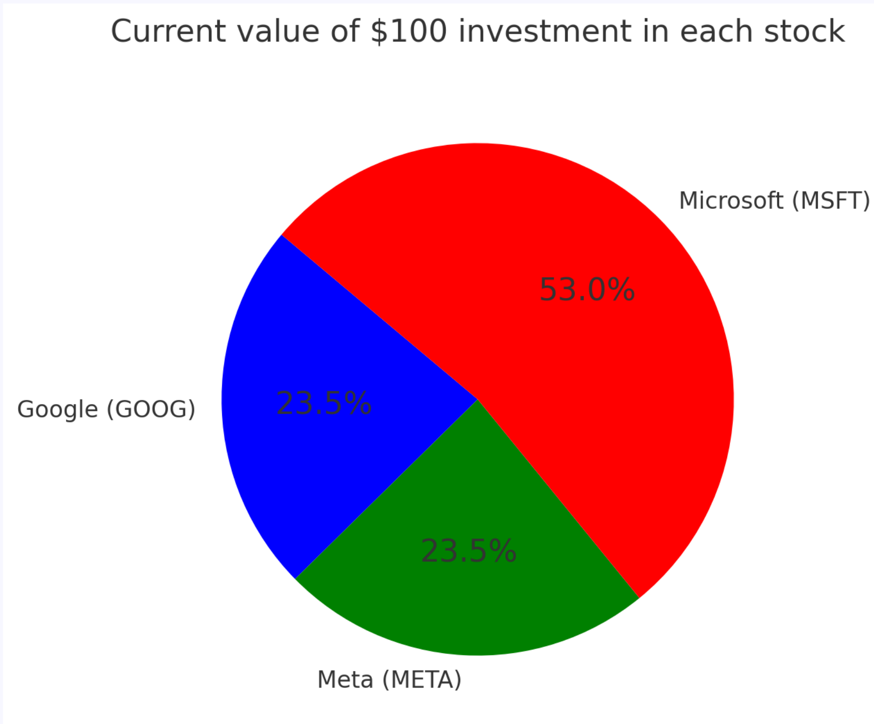
Here is the updated graph with the requested changes:

- The title, legend, axes labels, and ticks have all been doubled in size.
- The "^GSPC" label in the legend has been changed to "S&P 500".

This graph shows the one-decade performance of the three big tech companies: Google (GOOG), Meta (META), and Microsoft (MSFT), as well as the S&P 500 index, all normalized to start at 100. The beta values for the stocks are included in the legend.

Chat (ChatGPT Advanced Data Analysis) ctd.: Assume I put $100 in each of the three stocks at the start. Can you show a pie chart that shows how much I would have in each stock now?

[Code generated and executed]

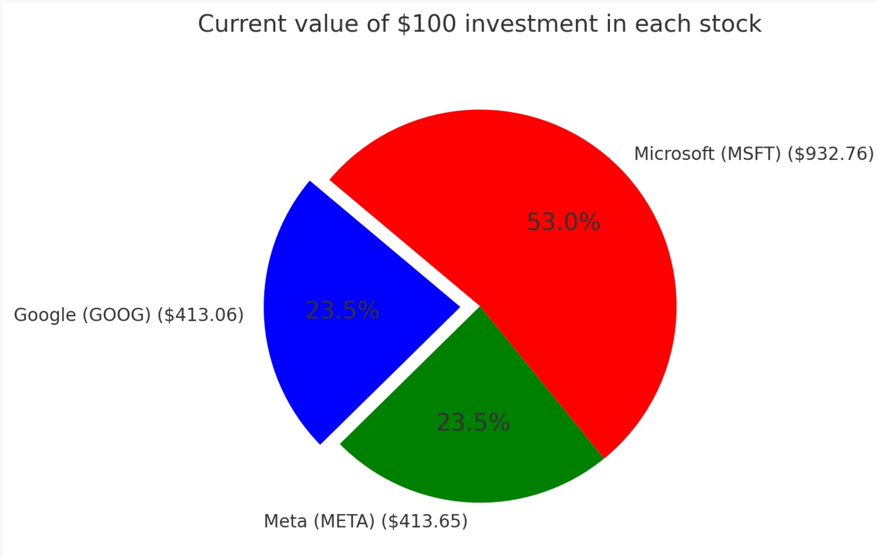


Here is a pie chart that shows the current value of a $100 investment made in each of the three big tech stocks (Google, Meta, and Microsoft) at the start of the period represented in the data.

The pie chart is divided into three sections, each representing one of the stocks. The size of each section corresponds to the current value of the initial $100 investment in that stock, expressed as a percentage of the total value of all three investments. The percentages are also labeled on the chart for clarity.

Chat (ChatGPT Advanced Data Analysis) ctd.: Can you label each position with its current dollar value in addition to the percentage? Can you make one of the pieces pop out of the pie?

[Code generated and executed]



Here is the updated pie chart. Each slice now includes both the current dollar value and the percentage of the total portfolio that it represents.

The first slice, representing the Google stock, is "exploded" or offset from the pie to highlight it. This visual effect can be useful for emphasizing a particular piece of a pie chart. In this case, it emphasizes that Google has the highest value of the three stocks based on an initial $100 investment.

(Author's note: the figure is correct, but ChatGPT's statement that Google has the highest value is false - this serves as a reminder that Advanced Data Analysis cannot "see" the figures it creates.)