.

import matplotlib.pyplot as plt

import numpy as np

import pandas as pd

def main():

y\_lin = lambda c,x: c[0]\*np.exp(c[1]\*x)

s\_lin = lambda cs: f"{cs[0]:.4f}·e^({cs[1]:+.4f}x)"

xs\_yy\_offset = 1950

h = 5

xs = np.arange(0,31,h) # [1950,1980]

ys = np.array([53.05,73.04,98.31,139.78,193.48,260.20,320.39]) # b

ck = np.polyfit(xs,np.log(ys),deg=1) # returns [c2,k=ln(c1)]

cs = np.array([np.exp(ck[1]),ck[0]]) # reordered,translated [ck]

print(f"\ncoeffs: [c,k] = {ck[1]:.4f},{ck[0]:.4f} → y(x) = {cs[0]:.4f}·e^({cs[1]:.4f}x)\n")

xs\_dis = np.linspace(min(xs),max(xs),(len(xs)-1)\*h+1) # display interval

ys\_lin = y\_lin(cs,xs\_dis) # linearized model over interval

yi\_lin = y\_lin(cs,xs) # linearized model at data x[i]

for x in xs\_dis:

y = y\_lin(cs,x)

# pandas dataframe # only for 4 data points so "xs"

ps = np.full((xs.size,4),np.nan) # cols = t,T,model,error

ps[:,0] = xs + xs\_yy\_offset

ps[:,1] = ys

ps[:,2] = yi\_lin

ps[:,3] = ps[:,1] - ps[:,2]

nCol = ["yy","cars","model","error"] # pandas can use latex, i guess?

df = pd.DataFrame(data=ps,columns=nCol) # populate dataframe

df["error"] = df["error"].apply("{:.4f}".format) # format column, error

df["yy"] = df["yy"].apply("{:.0f}".format) # format column, time

display(df)

# plot, data

plt.scatter(xs+xs\_yy\_offset,ys,c="C0",marker="o",label="some data")

plt.scatter(xs+xs\_yy\_offset,yi\_lin,c="C1",marker="\*")

plt.plot(xs\_dis+xs\_yy\_offset,ys\_lin,c="C1",linestyle=":",label=s\_lin(cs))

# plot, style

plt.title("\nso many cars, so little time\n")

plt.legend(bbox\_to\_anchor=(1.02,1),loc='upper left',borderaxespad=0)

plt.xlabel("year")

plt.ylabel("cars (M)")

# plot, display

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

main()