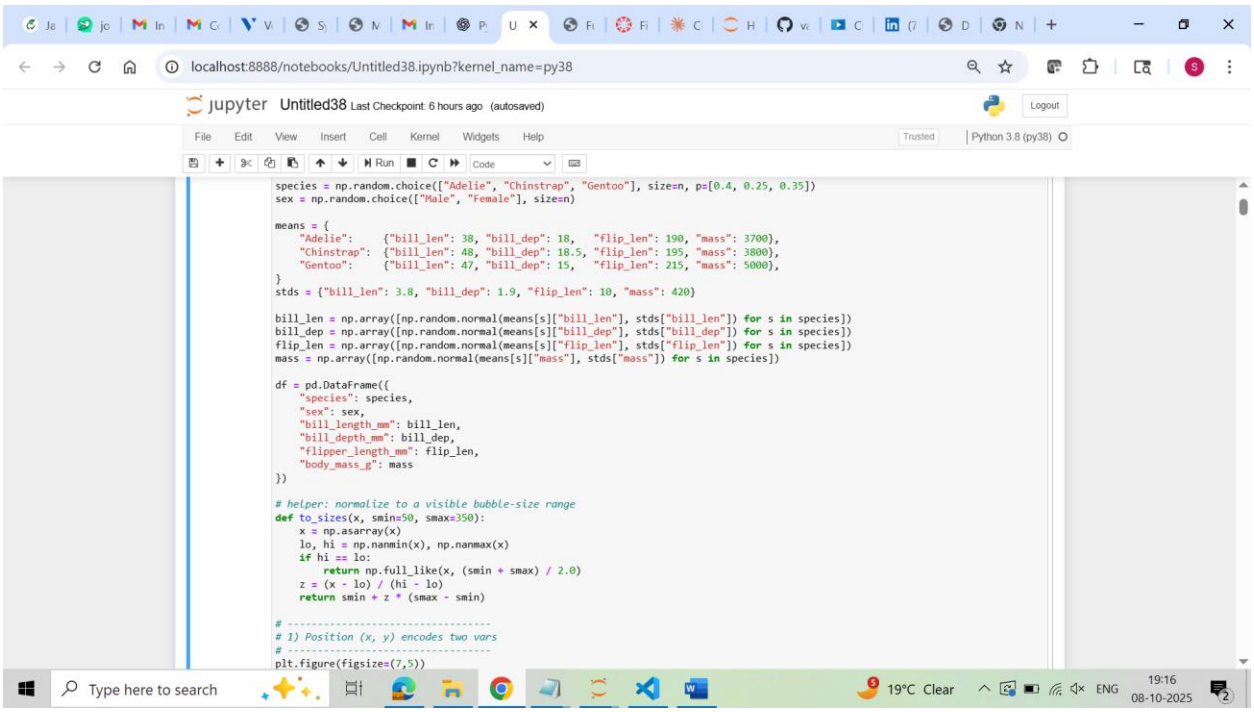
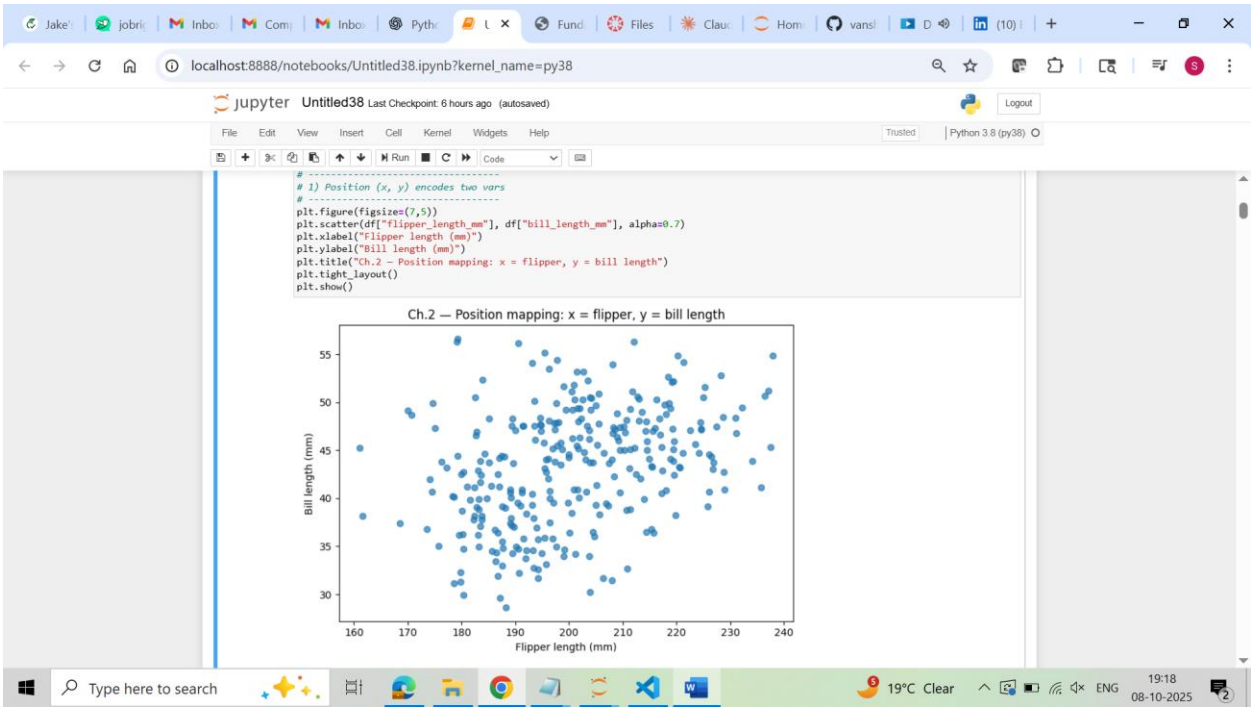
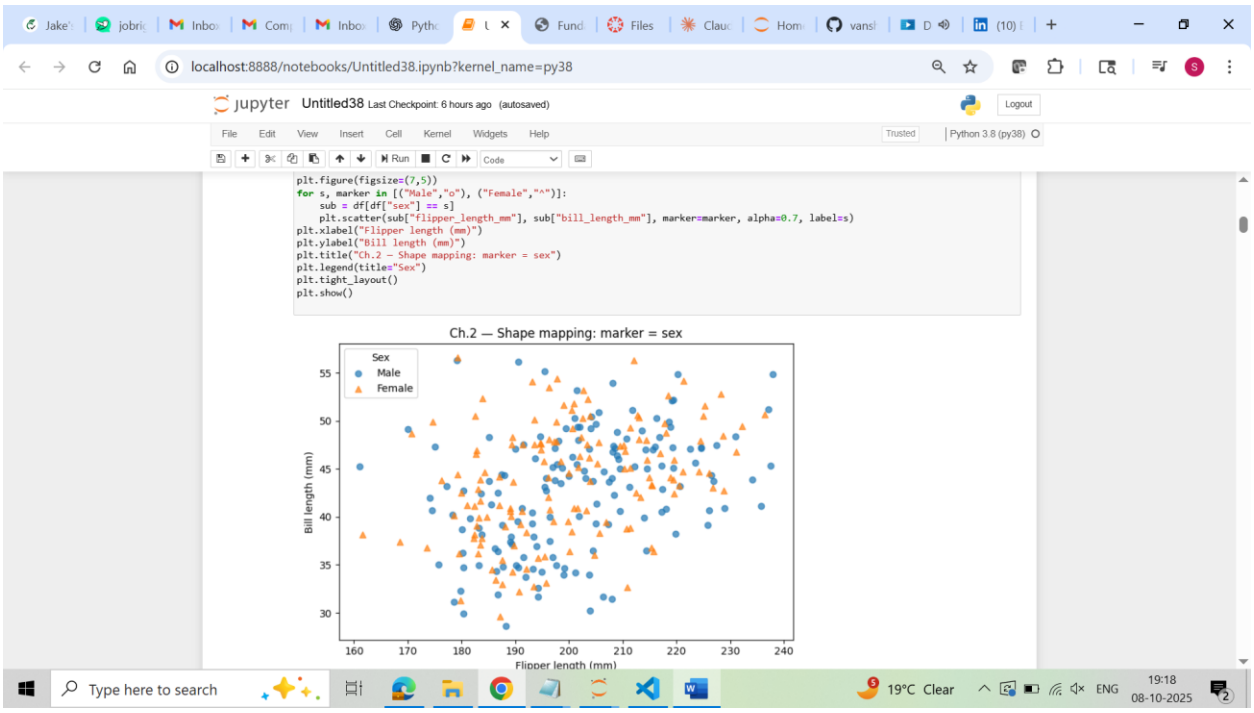
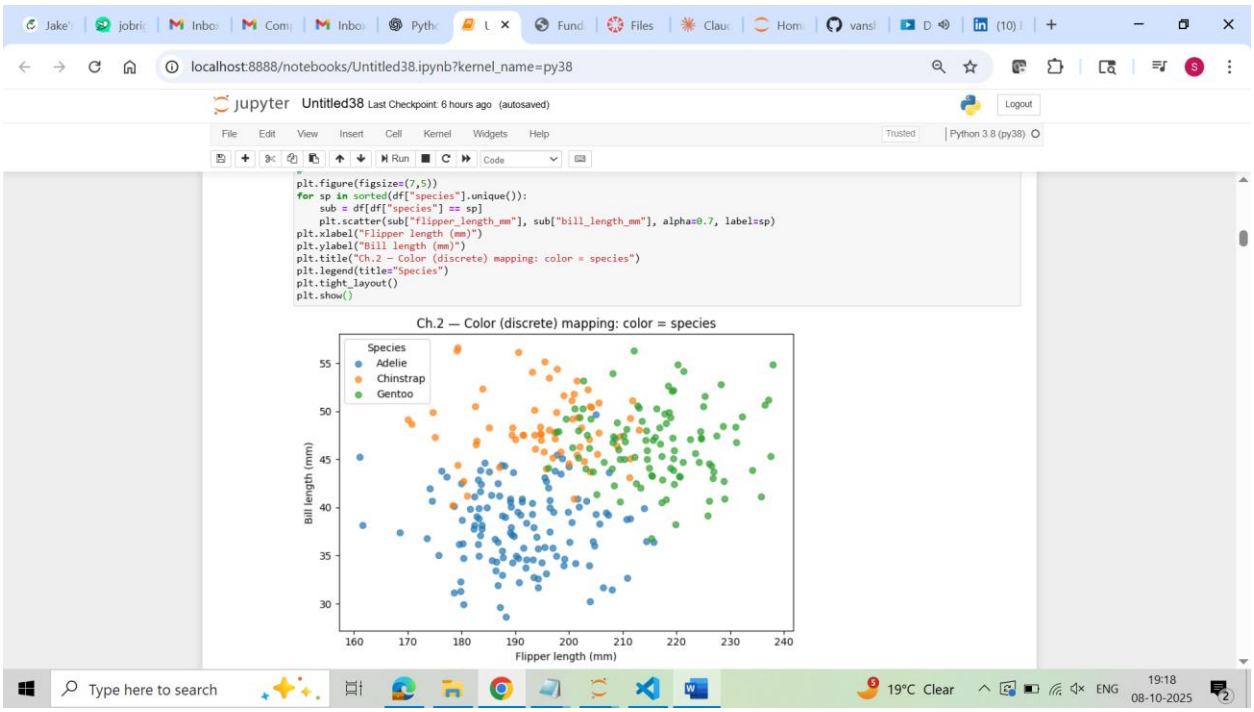
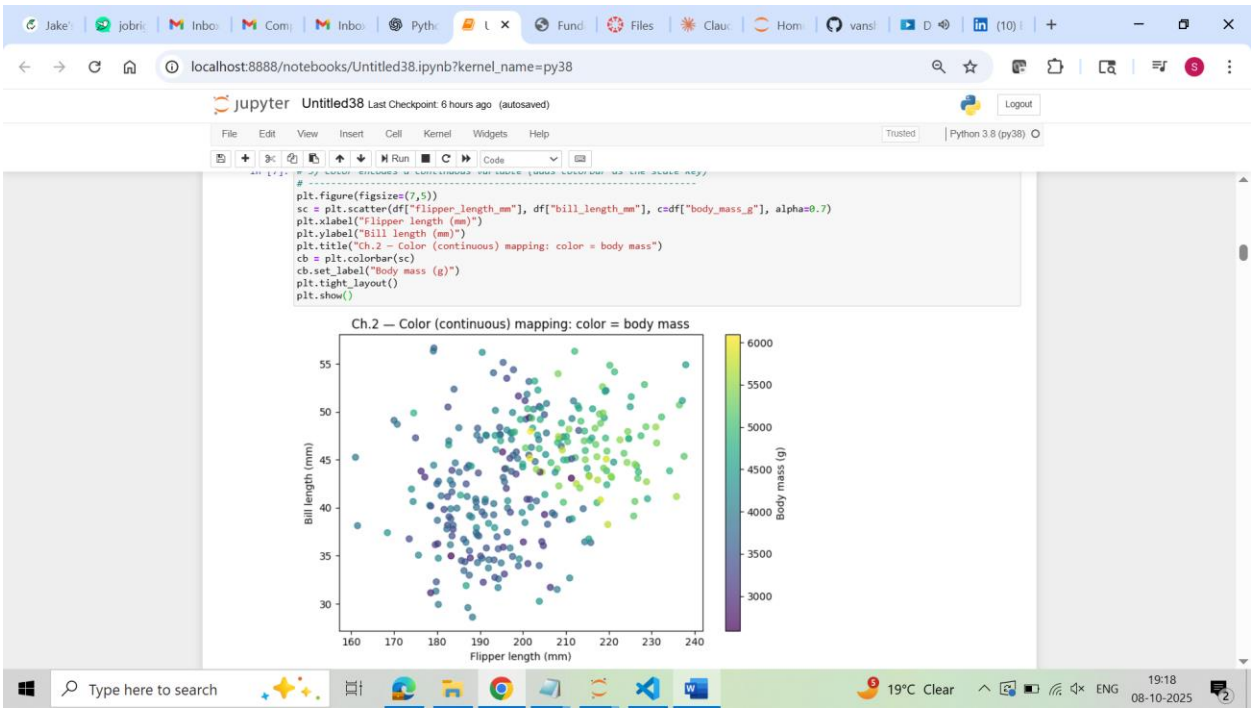
CHAPTER 2

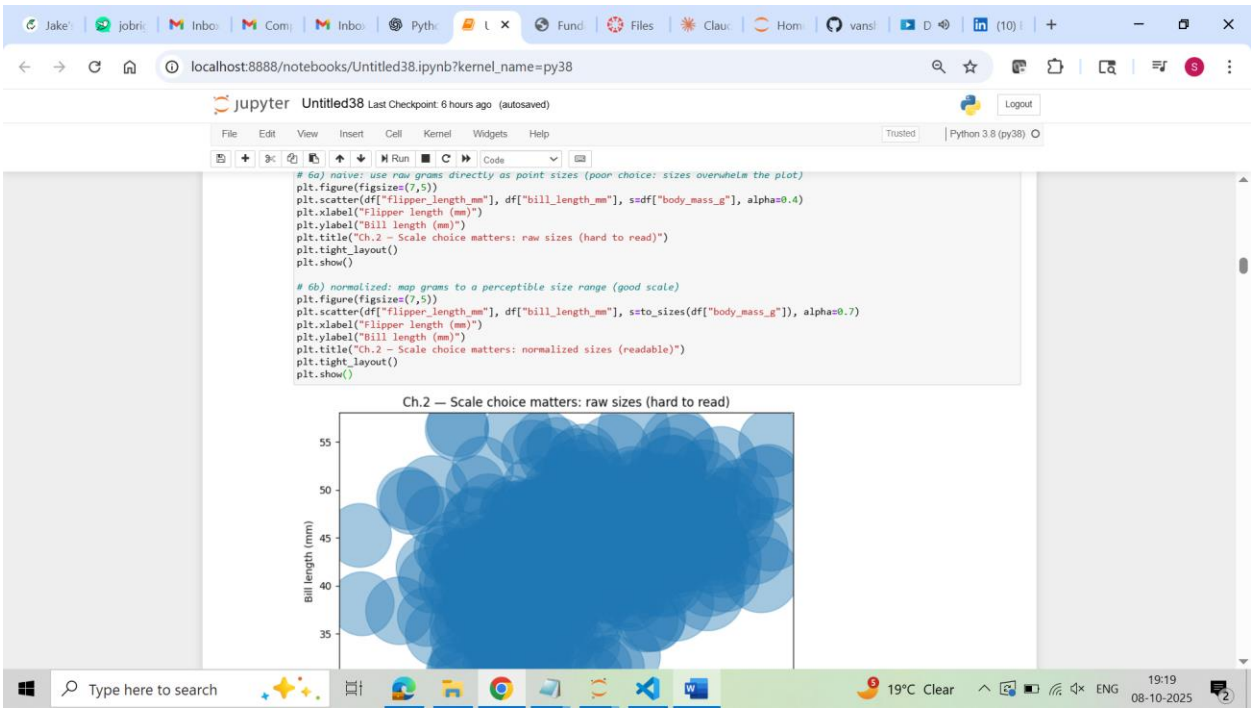

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
species = np.random.choice(["Adelie", "Chinstrap", "Gentoo"], size=n, p=[0.4, 0.25, 0.35])
sex = np.random.choice(["Male", "Female"], size=n)
means = {
"Adelie": {"bill_len": 38, "bill_dep": 18, "flip_len": 190, "mass": 3700},
"Chinstrap": {"bill_len": 48, "bill_dep": 18.5, "flip_len": 195, "mass": 3800},
"Gentoo": {"bill_len": 47, "bill_dep": 15, "flip_len": 215, "mass": 5000},
}
stds = {"bill_len": 3.8, "bill_dep": 1.9, "flip_len": 10, "mass": 420}
bill_len = np.array([np.random.normal(means[s]["bill_len"], stds["bill_len"]) for s in species])
bill_dep = np.array([np.random.normal(means[s]["bill_dep"], stds["bill_dep"]) for s in species])
flip_len = np.array([np.random.normal(means[s]["flip_len"], stds["flip_len"]) for s in species])
mass = np.array([np.random.normal(means[s]["mass"], stds["mass"]) for s in species])
df = pd.DataFrame({
"species": species,
"sex": sex,
"bill_length_mm": bill_len,
"bill_depth_mm": bill_dep,
"flipper_length_mm": flip_len,
"body_mass_g": mass
})
# helper: normalize to a visible bubble-size range
def to_sizes(x, smin=50, smax=350):
x = np.asarray(x)
lo, hi = np.nanmin(x), np.nanmax(x)
if hi == lo:
return np.full_like(x, (smin + smax) / 2.0)
z = (x - lo) / (hi - lo)
return smin + z * (smax - smin)
# ----------------------------------
# 1) Position (x, y) encodes two vars
# ----------------------------------
plt.figure(figsize=(7,5))
Type here to search
19°C Clear
ENG
19:16
08-10-2025

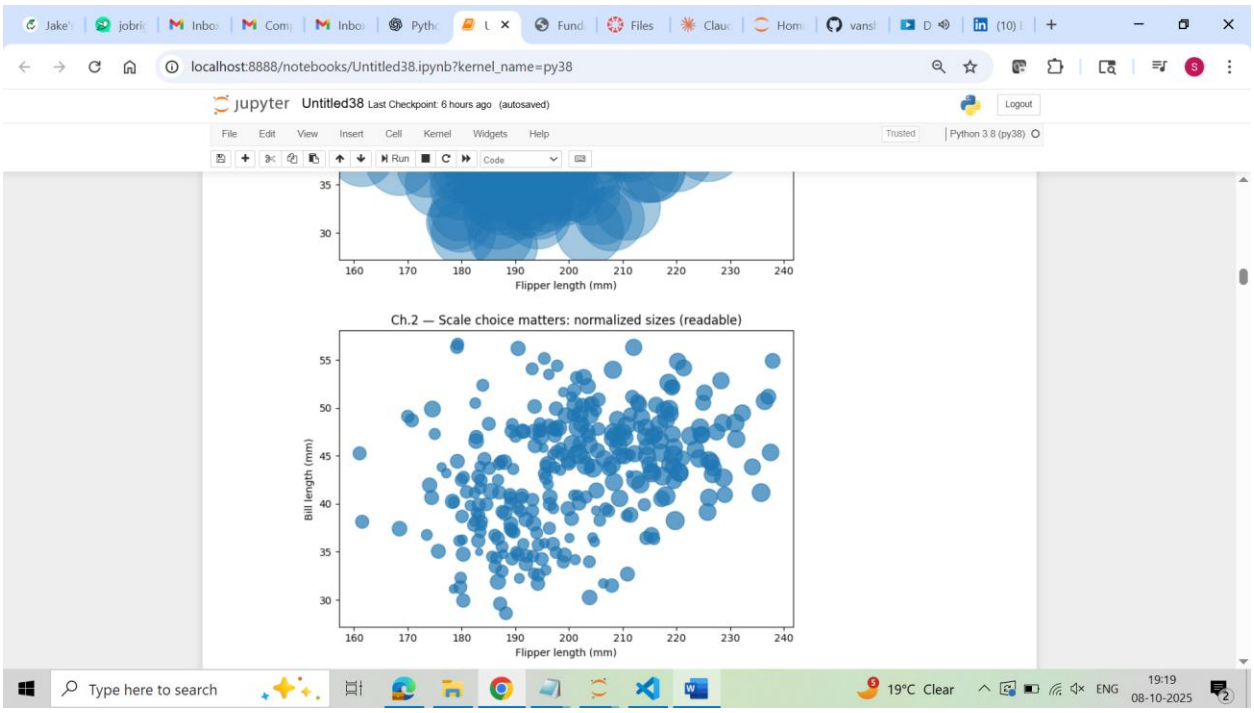

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
# ----------------------------------
# 1) Position (x, y) encodes two vars
# ----------------------------------
plt.figure(figsize=(7,5))
plt.scatter(df["flipper_length_mm"], df["bill_length_mm"], alpha=0.7)
plt.xlabel("Flipper length (mm)")
plt.ylabel("Bill length (mm)")
plt.title("Ch.2 – Position mapping: x = flipper, y = bill length")
plt.tight_layout()
plt.show()
Ch.2 — Position mapping: x = flipper, y = bill length
Bill length (mm)
Flipper length (mm)
55
50
45
40
35
30
160
170
180
190
200
210
220
230
240
Type here to search
19°C Clear
ENG
19:18
08-10-2025

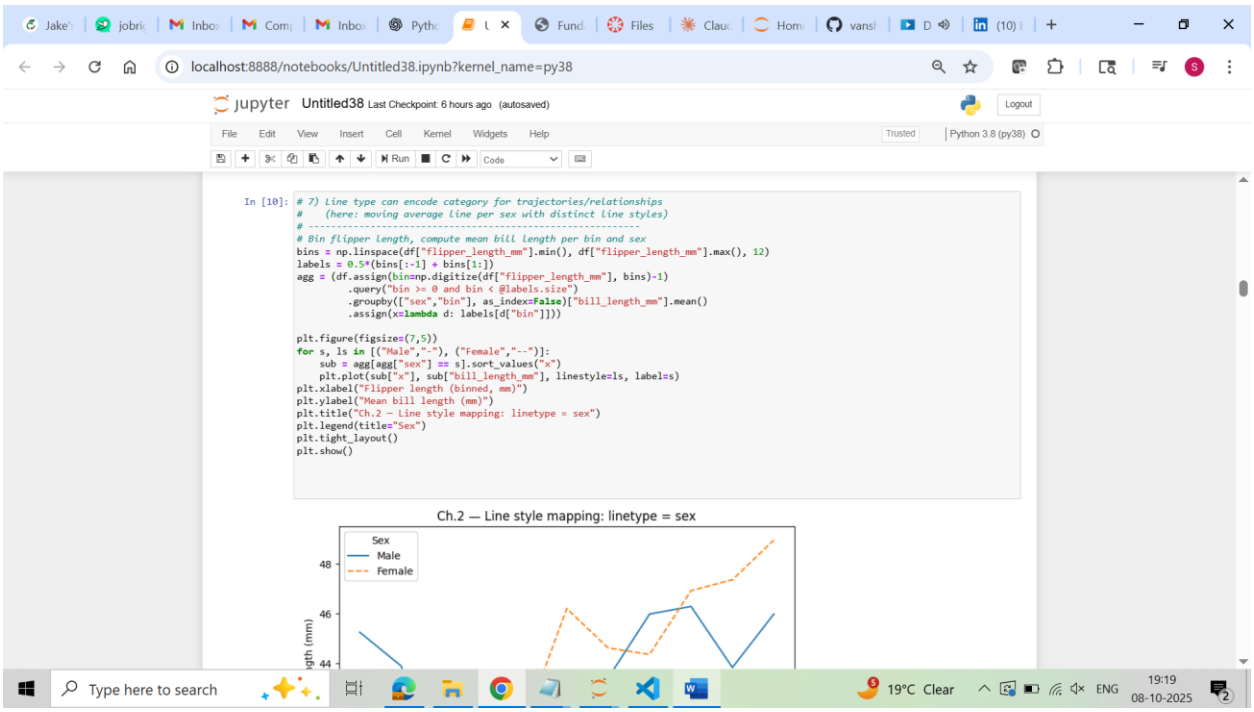

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
plt.figure(figsize=(7,5))
for s, marker in [("Male","o"), ("Female","^")]:
sub = df[df["sex"] == s]
plt.scatter(sub["flipper_length_mm"], sub["bill_length_mm"], marker=marker, alpha=0.7, label=s)
plt.xlabel("Flipper length (mm)")
plt.ylabel("Bill length (mm)")
plt.title("Ch.2 — Shape mapping: marker = sex")
plt.legend(title="Sex")
plt.tight_layout()
plt.show()
Ch.2 — Shape mapping: marker = sex
Sex
Male
Female
Bill length (mm)
Flipper length (mm)
Type here to search
19°C Clear
ENG
19:18
08-10-2025

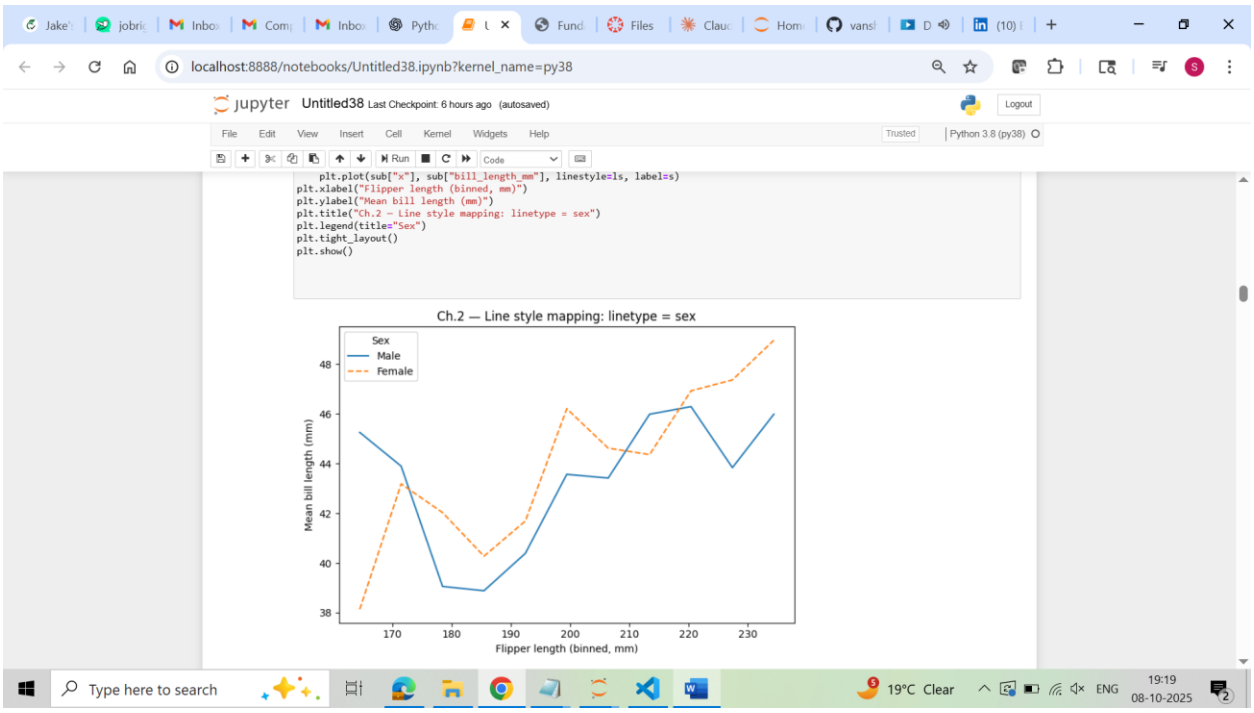

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
plt.figure(figsize=(7,5))
for sp in sorted(df["species"].unique()):
sub = df[df["species"] == sp]
plt.scatter(sub["flipper_length_mm"], sub["bill_length_mm"], alpha=0.7, label=sp)
plt.xlabel("Flipper length (mm)")
plt.ylabel("Bill length (mm)")
plt.title("Ch.2 — Color (discrete) mapping: color = species")
plt.legend(title="Species")
plt.tight_layout()
plt.show()
Ch.2 — Color (discrete) mapping: color = species
Species
Adelie
Chinstrap
Gentoo
Bill length (mm)
Flipper length (mm)
Type here to search
19°C Clear
ENG
19:18
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
# ----------
plt.figure(figsize=(7,5))
sc = plt.scatter(df["flipper_length_mm"], df["bill_length_mm"], c=df["body_mass_g"], alpha=0.7)
plt.xlabel("Flipper length (mm)")
plt.ylabel("Bill length (mm)")
plt.title("Ch.2 — Color (continuous) mapping: color = body mass")
cb = plt.colorbar(sc)
cb.set_label("Body mass (g)")
plt.tight_layout()
plt.show()
Ch.2 — Color (continuous) mapping: color = body mass
Bill length (mm)
Flipper length (mm)
Body mass (g)
55
50
45
40
35
30
160
170
180
190
200
210
220
230
240
6000
5500
5000
4500
4000
3500
3000
Type here to search
19°C Clear
ENG
19:18
08-10-2025

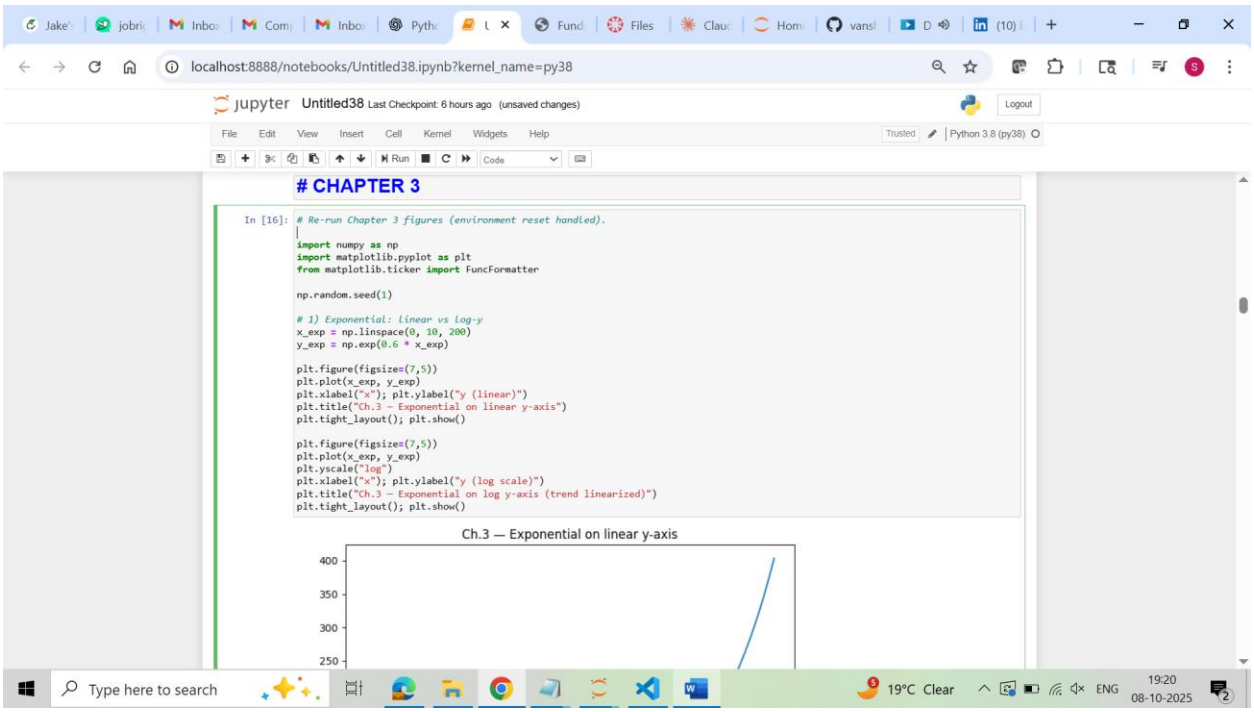

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
# 6a) naive: use raw grams directly as point sizes (poor choice: sizes overwhelm the plot)
plt.figure(figsize=(7,5))
plt.scatter(df["flipper_length_mm"], df["bill_length_mm"], s=df["body_mass_g"], alpha=0.4)
plt.xlabel("Flipper length (mm)")
plt.ylabel("Bill length (mm)")
plt.title("Ch.2 — Scale choice matters: raw sizes (hard to read)")
plt.tight_layout()
plt.show()
# 6b) normalized: map grams to a perceptible size range (good scale)
plt.figure(figsize=(7,5))
plt.scatter(df["flipper_length_mm"], df["bill_length_mm"], s=to_sizes(df["body_mass_g"]), alpha=0.7)
plt.xlabel("Flipper length (mm)")
plt.ylabel("Bill length (mm)")
plt.title("Ch.2 — Scale choice matters: normalized sizes (readable)")
plt.tight_layout()
plt.show()
Ch.2 — Scale choice matters: raw sizes (hard to read)
Bill length (mm)
55
50
45
40
35
Type here to search
19°C Clear
ENG
19:19
08-10-2025

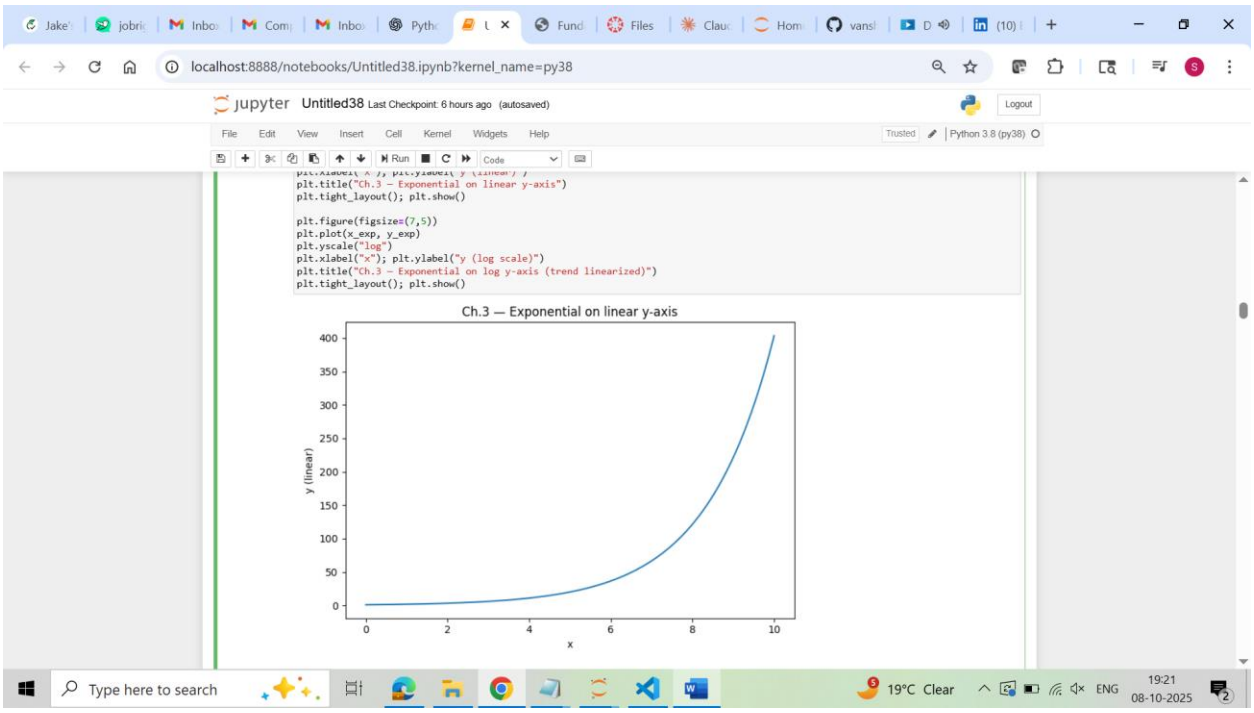

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
Flipper length (mm)
Ch.2 — Scale choice matters: normalized sizes (readable)
Bill length (mm)
Flipper length (mm)
Type here to search
19°C Clear
ENG
19:19
08-10-2025

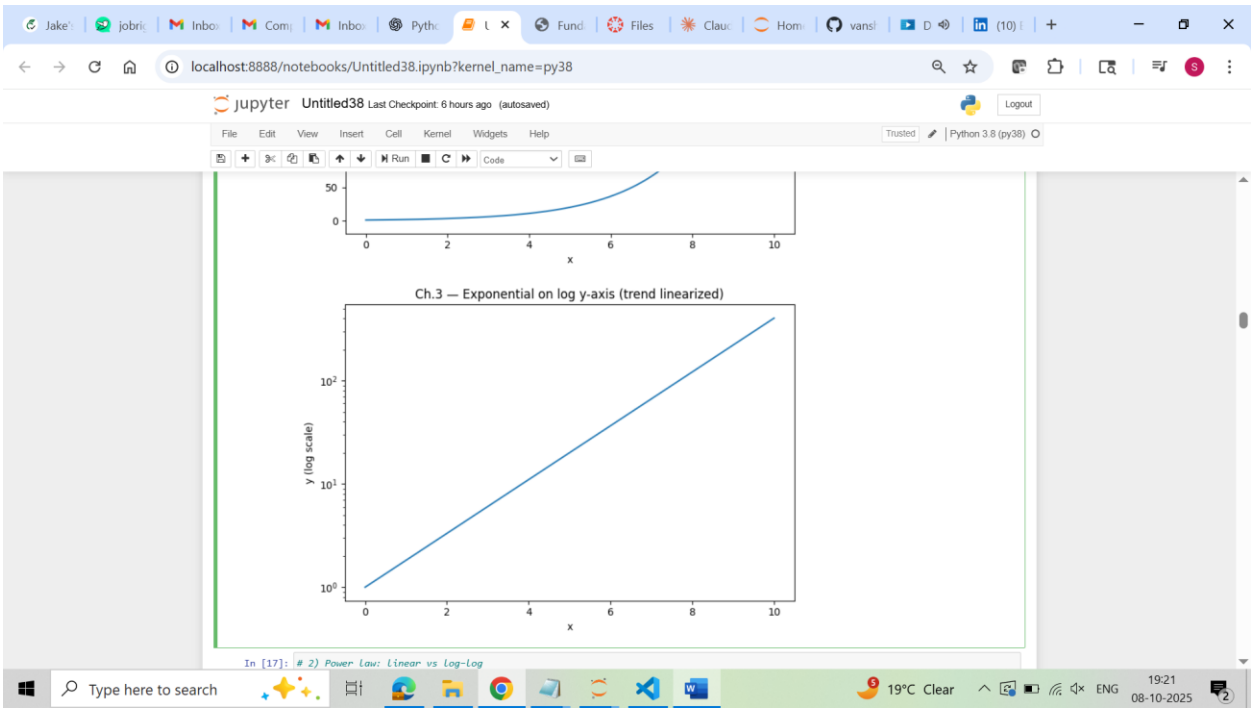

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
In [10]:
# 7) Line type can encode category for trajectories/relationships
# (here: moving average line per sex with distinct line styles)
# --
# Bin flipper length, compute mean bill length per bin and sex
bins = np.linspace(df["flipper_length_mm"].min(), df["flipper_length_mm"].max(), 12)
labels = 0.5*(bins[:-1] + bins[1:])
agg = (df.assign(bin=np.digitize(df["flipper_length_mm"], bins)-1)
.query("bin >= 0 and bin < @labels.size")
.groupby(["sex","bin"], as_index=False)["bill_length_mm"].mean()
.assign(x=lambda d: labels[d["bin"]]))
plt.figure(figsize=(7,5))
for s, ls in [("Male","-"), ("Female","--")]:
sub = agg[agg["sex"] == s].sort_values("x")
plt.plot(sub["x"], sub["bill_length_mm"], linestyle=ls, label=s)
plt.xlabel("Flipper length (binned, mm)")
plt.ylabel("Mean bill length (mm)")
plt.title("Ch.2 — Line style mapping: linetype = sex")
plt.legend(title="Sex")
plt.tight_layout()
plt.show()
Ch.2 — Line style mapping: linetype = sex
Sex
Male
Female
Type here to search
19°C Clear
ENG
19:19
08-10-2025

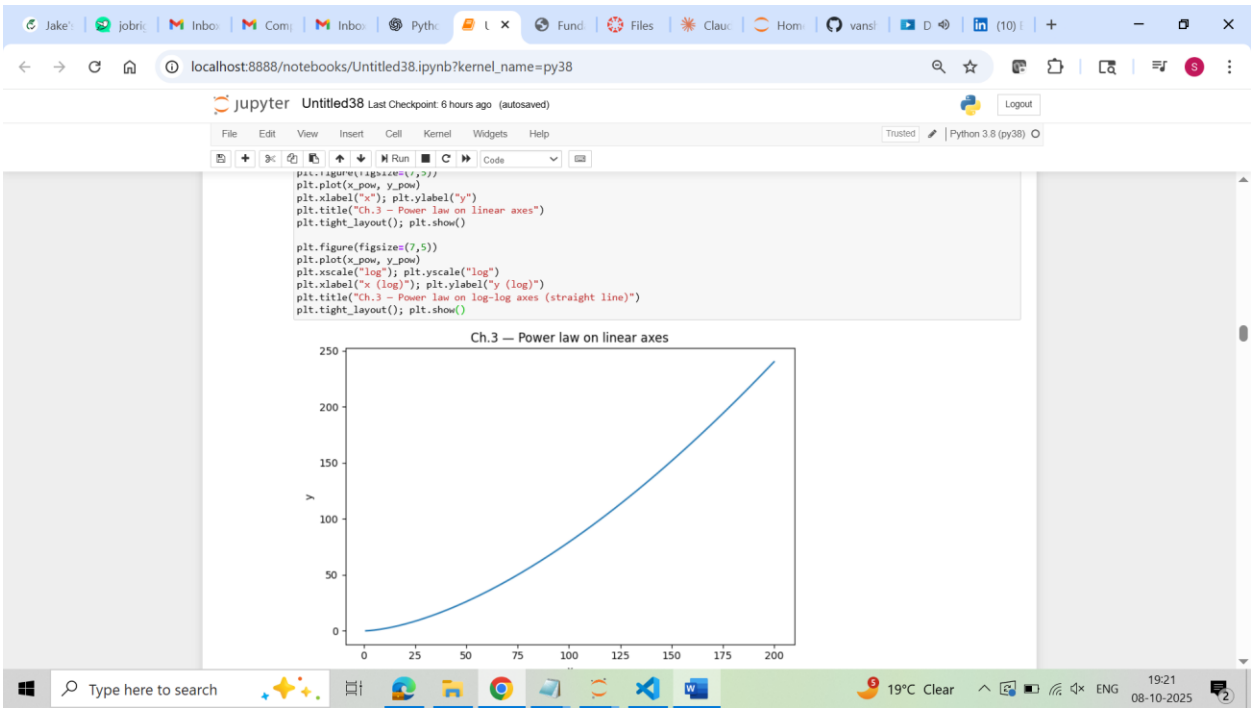

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
plt.plot(sub["x"], sub["bill_length_mm"], linestyle=ls, label=s)
plt.xlabel("Flipper length (binned, mm)")
plt.ylabel("Mean bill length (mm)")
plt.title("Ch.2 — Line style mapping: linetype = sex")
plt.legend(title="Sex")
plt.tight_layout()
plt.show()
Ch.2 — Line style mapping: linetype = sex
Sex
Male
Female
Mean bill length (mm)
Flipper length (binned, mm)
Type here to search
19°C Clear
ENG
19:19
08-10-2025

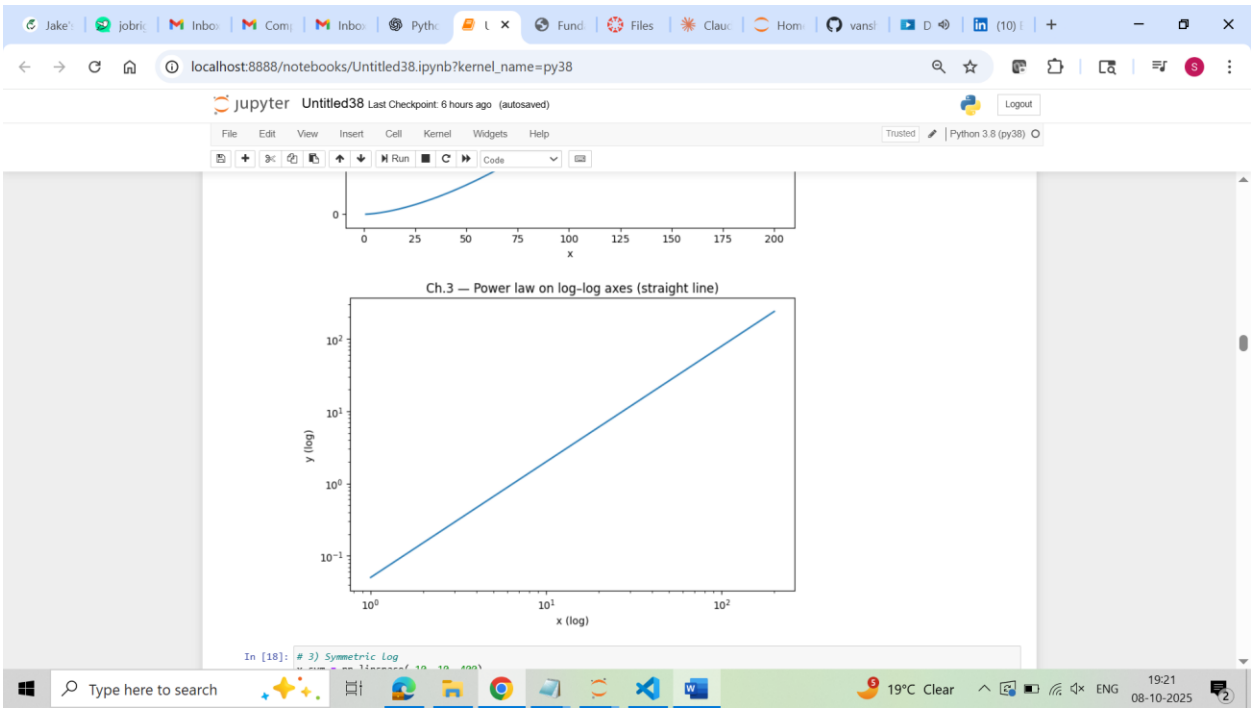
CHAPTER 3


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
CHAPTER 3
In [16]:
# Re-run Chapter 3 figures (environment reset handled).
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
np.random.seed(1)
# 1) Exponential: linear vs log-y
x_exp = np.linspace(0, 10, 200)
y_exp = np.exp(0.6 * x_exp)
plt.figure(figsize=(7,5))
plt.plot(x_exp, y_exp)
plt.xlabel("x"); plt.ylabel("y (linear)")
plt.title("Ch.3 — Exponential on linear y-axis")
plt.tight_layout(); plt.show()
plt.figure(figsize=(7,5))
plt.plot(x_exp, y_exp)
plt.yscale("log")
plt.xlabel("x"); plt.ylabel("y (log scale)")
plt.title("Ch.3 — Exponential on log y-axis (trend linearized)")
plt.tight_layout(); plt.show()
Ch.3 — Exponential on linear y-axis
Type here to search
19°C Clear
ENG
19:20
08-10-2025

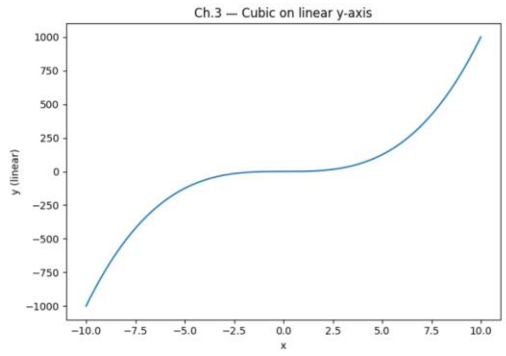

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
plt.title("Ch.3 — Exponential on linear y-axis")
plt.tight_layout(); plt.show()
plt.figure(figsize=(7,5))
plt.plot(x_exp, y_exp)
plt.yscale("log")
plt.xlabel("x"); plt.ylabel("y (log scale)")
plt.title("Ch.3 — Exponential on log y-axis (trend linearized)")
plt.tight_layout(); plt.show()
Ch.3 — Exponential on linear y-axis
y (linear)
x
Type here to search
19°C Clear
ENG
19:21
08-10-2025

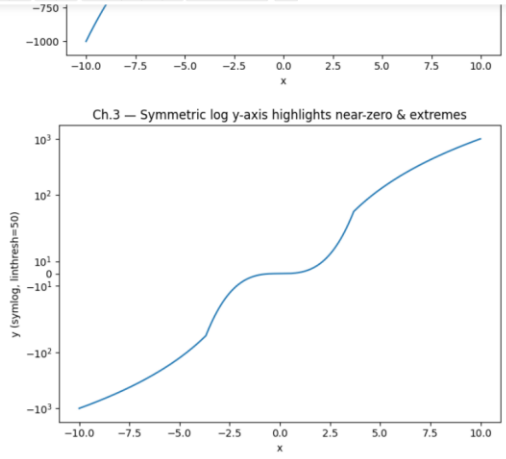

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
Ch.3 — Exponential on log y-axis (trend linearized)
y (log scale)
x
In [17]: # 2) Power law: linear vs log-log
Type here to search
19°C Clear
ENG
19:21
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
plt.plot(x_pow, y_pow)
plt.xlabel("x"); plt.ylabel("y")
plt.title("Ch.3 — Power law on linear axes")
plt.tight_layout(); plt.show()
plt.figure(figsize=(7,5))
plt.plot(x_pow, y_pow)
plt.xscale("log"); plt.yscale("log")
plt.xlabel("x (log)"); plt.ylabel("y (log)")
plt.title("Ch.3 — Power law on log-log axes (straight line)")
plt.tight_layout(); plt.show()
Ch.3 — Power law on linear axes
y
x
0
50
100
150
200
250
0
25
50
75
100
125
150
175
200
Type here to search
19°C Clear
ENG
19:21
08-10-2025

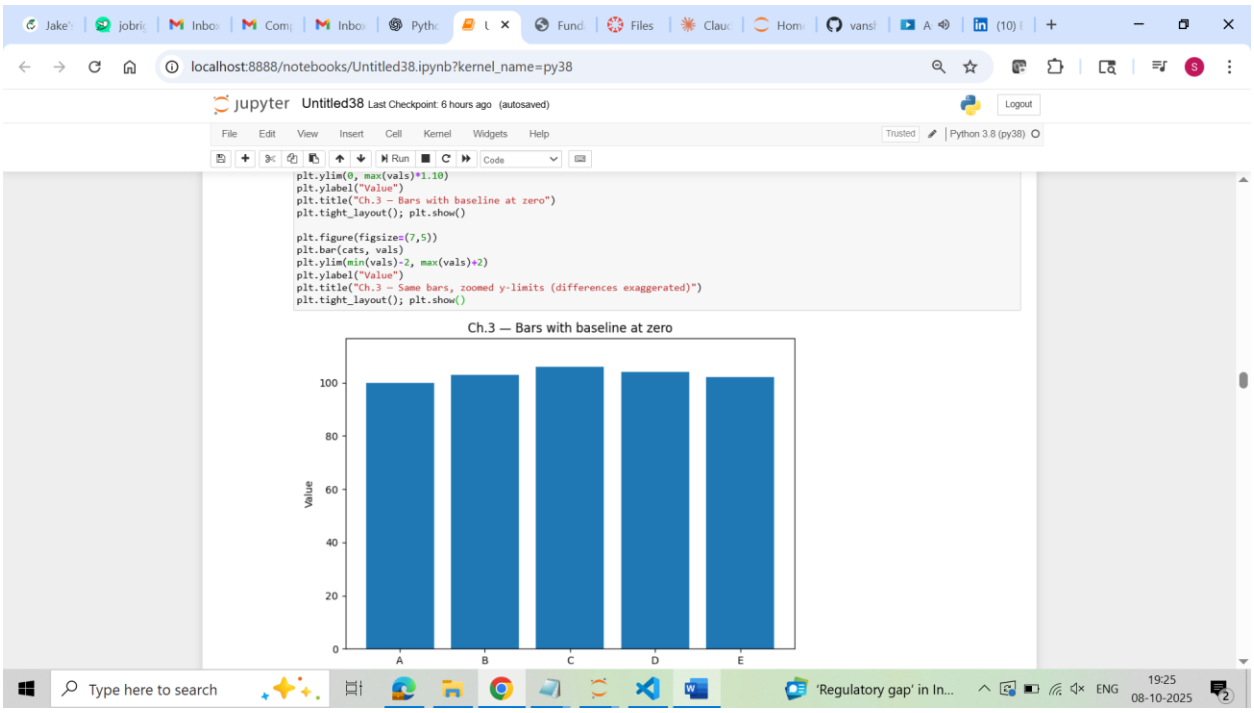

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
0
0
25
50
75
100
125
150
175
200
x
Ch.3 — Power law on log-log axes (straight line)
y (log)
x (log)
10^2
10^1
10^0
10^{-1}
10^0
10^1
10^2
In [18]: # 3) Symmetric Log
Type here to search
19°C Clear
ENG
19:21
08-10-2025

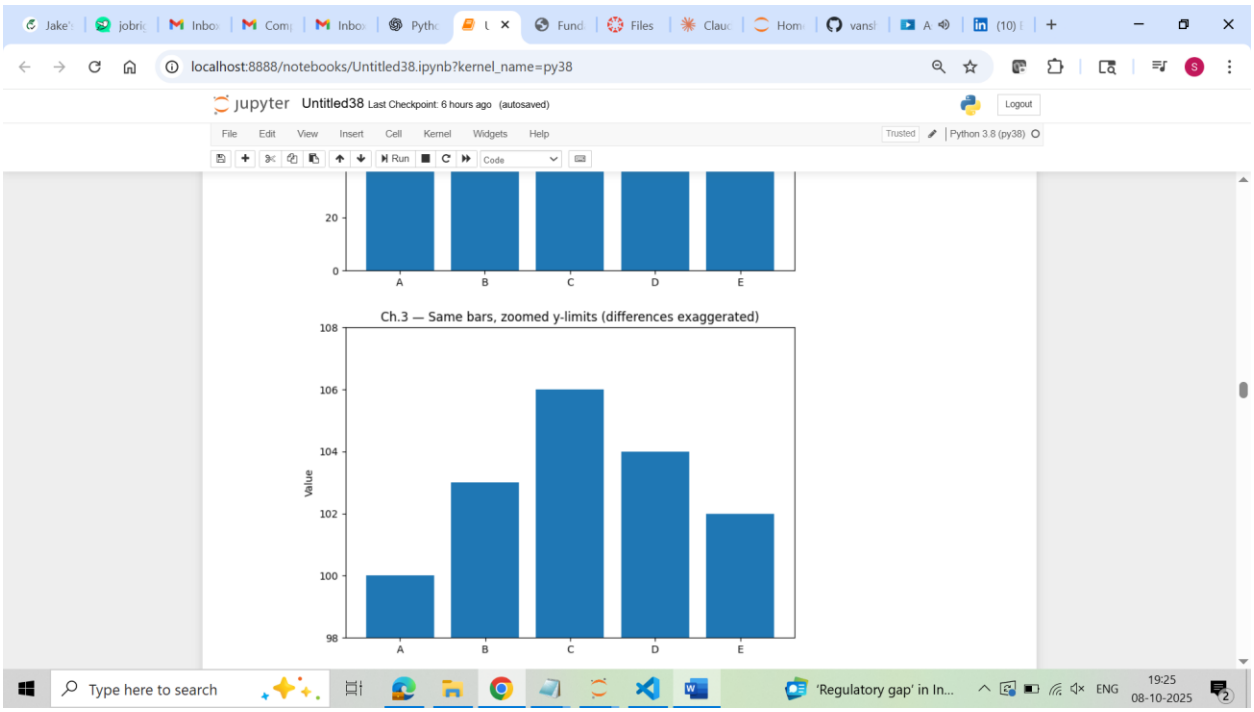
```
plt.xlabel("x"); plt.ylabel("y (linear)")
plt.title("Ch.3 — Cubic on linear y-axis")
plt.tight_layout(); plt.show()

plt.figure(figsize=(7,5))
plt.plot(x_sym, y_sym)
plt.yscale("symlog", linthresh=50)
plt.xlabel("x"); plt.ylabel("y (symlog, linthresh=50)")
plt.title("Ch.3 — Symmetric log y-axis highlights near-zero & extremes")
plt.tight_layout(); plt.show()
```

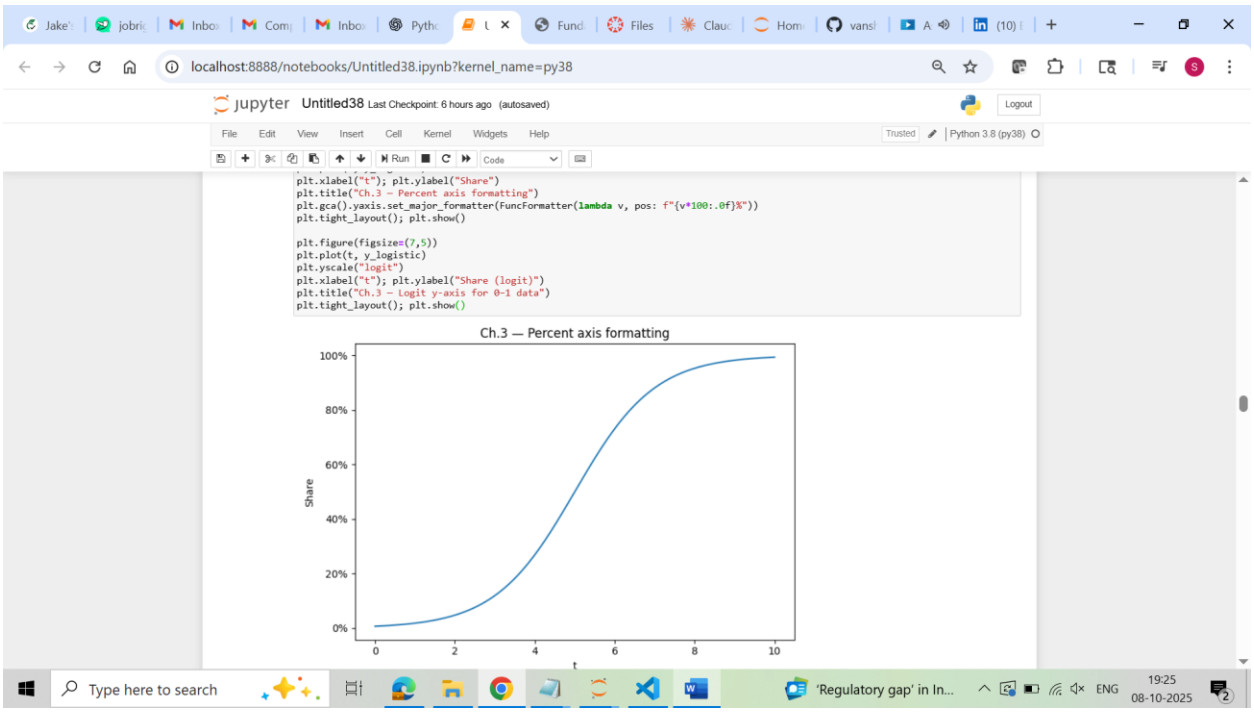


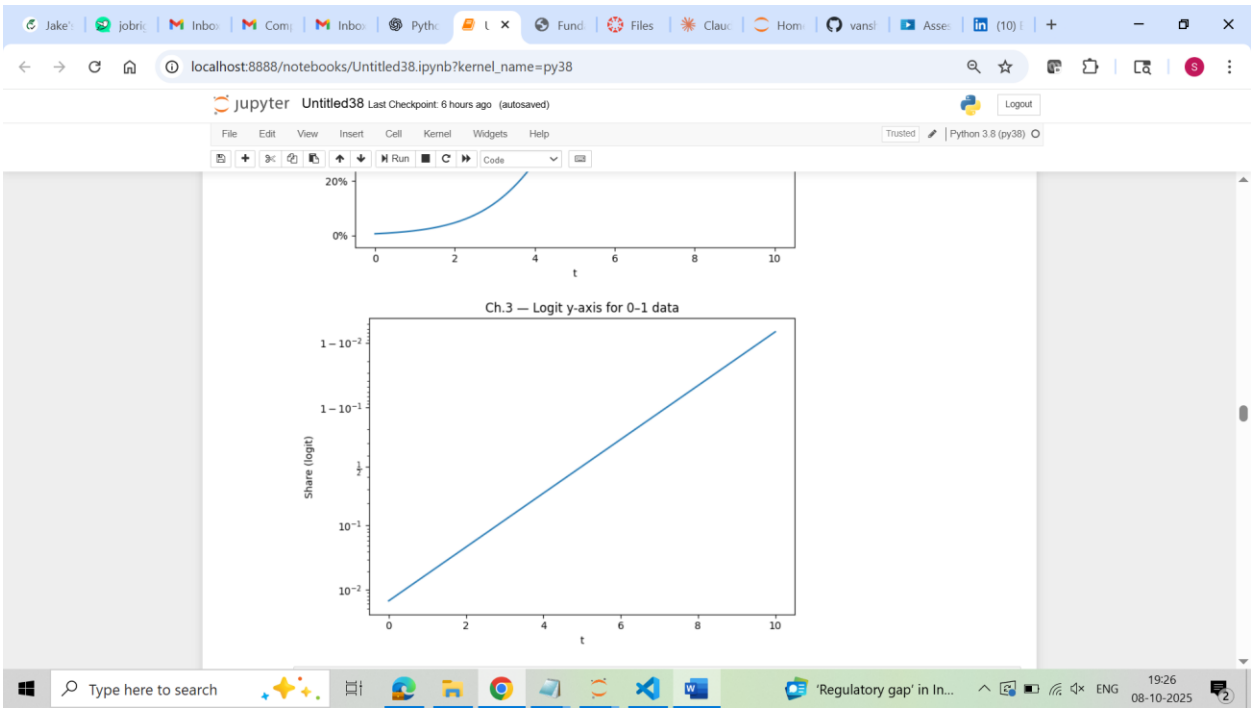


In [ ]:

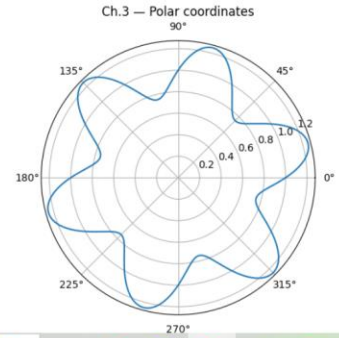

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
plt.ylim(0, max(vals)*1.10)
plt.ylabel("Value")
plt.title("Ch.3 — Bars with baseline at zero")
plt.tight_layout(); plt.show()
plt.figure(figsize=(7,5))
plt.bar(cats, vals)
plt.ylim(min(vals)-2, max(vals)+2)
plt.ylabel("Value")
plt.title("Ch.3 — Same bars, zoomed y-limits (differences exaggerated)")
plt.tight_layout(); plt.show()
Ch.3 — Bars with baseline at zero
Value
0
20
40
60
80
100
A
B
C
D
E
Type here to search
'Regulatory gap' in In...
ENG
19:25
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
0
20
A
B
C
D
E
Ch.3 — Same bars, zoomed y-limits (differences exaggerated)
Value
98
100
102
104
106
108
A
B
C
D
E
Type here to search
'Regulatory gap' in In...
ENG
19:25
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
plt.xlabel("t"); plt.ylabel("Share")
plt.title("Ch.3 — Percent axis formatting")
plt.gca().yaxis.set_major_formatter(FuncFormatter(lambda v, pos: f"{v*100:.0f}%"))
plt.tight_layout(); plt.show()
plt.figure(figsize=(7,5))
plt.plot(t, y_logistic)
plt.yscale("logit")
plt.xlabel("t"); plt.ylabel("Share (logit)")
plt.title("Ch.3 — Logit y-axis for 0–1 data")
plt.tight_layout(); plt.show()
Ch.3 — Percent axis formatting
100%
80%
60%
40%
20%
0%
Share
0
2
4
6
8
10
t
Type here to search
'Regulatory gap' in In...
ENG
19:25
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
20%
0%
0
2
4
6
8
10
t
Ch.3 — Logit y-axis for 0–1 data
1 − 10⁻²
1 − 10⁻¹
10⁻¹
10⁻²
Share (logit)
0
2
4
6
8
10
t
Type here to search
'Regulatory gap' in In...
ENG
19:26
08-10-2025

```
y_c = 0.5*x_c + 10
def c2f(c): return c*9/5 + 32
def f2c(f): return (f - 32)*5/9

fig, ax = plt.subplots(figsize=(7,5))
ax.plot(x_c, y_c)
ax.set_xlabel("Temperature (°C)")
ax.set_ylabel("Some response (°C units)")
ax.set_title("Ch.3 — Secondary x-axis (°F) via exact transform")
secax = ax.secondary_xaxis("top", functions=(c2f, f2c))
secax.set_xlabel("Temperature (°F)")
plt.tight_layout(); plt.show()
```



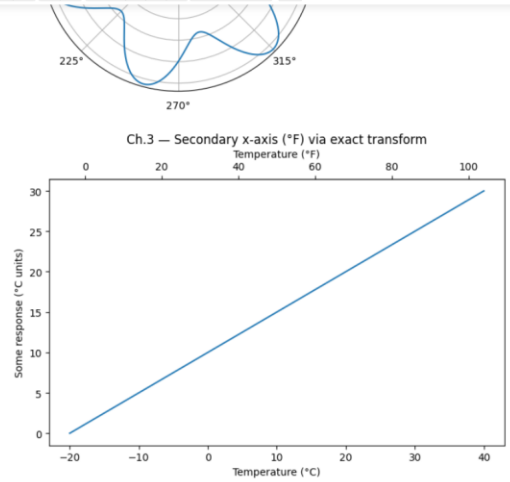
localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)



localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

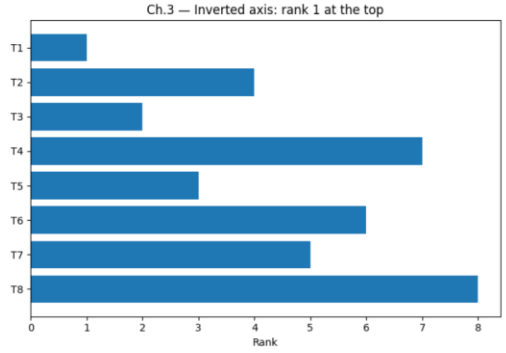
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run Code

In [23]:
```
# 9) Inverted axis (rank: 1 at top)
teams = ["T1","T2","T3","T4","T5","T6","T7","T8"]
ranks = np.array([1,4,2,7,3,6,5,8])

plt.figure(figsize=(7,5))
plt.barh(teams, ranks)
plt.gca().invert_yaxis()
plt.xlabel("Rank")
plt.title("Ch.3 — Inverted axis: rank 1 at the top")
plt.tight_layout(); plt.show()
```



Type here to search

19°C Clear ENG 19:33 08-10-2025

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run Code

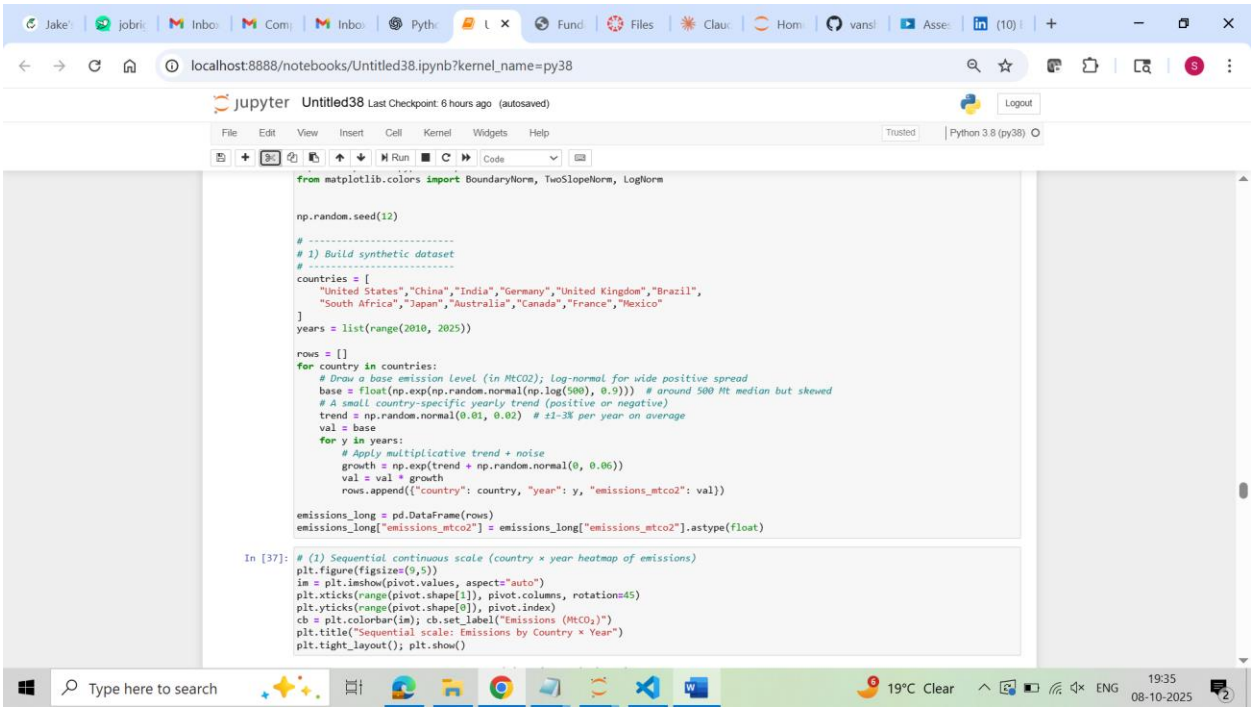
```
import matplotlib.pyplot as plt
from matplotlib.ticker import FuncFormatter
import pandas as pd

# ---- sample data (country ~ approx capital coords) ----
data = [
    {"country":"United States", "lat":38.9,  "lon":-77.0,  "lifeExp":78},
    {"country":"Canada",        "lat":45.4,  "lon":-75.7,  "lifeExp":82},
    {"country":"Mexico",        "lat":19.4,  "lon":-99.1,  "lifeExp":75},
    {"country":"Brazil",        "lat":-15.8, "lon":-47.9,  "lifeExp":76},
    {"country":"UK",            "lat":51.5,  "lon":-0.1,   "lifeExp":81},
    {"country":"Germany",       "lat":52.5,  "lon":13.4,   "lifeExp":81},
    {"country":"Egypt",         "lat":30.0,  "lon":31.2,   "lifeExp":71},
    {"country":"South Africa",  "lat":-25.7, "lon":28.2,   "lifeExp":64},
    {"country":"India",         "lat":28.6,  "lon":77.2,   "lifeExp":69},
    {"country":"China",         "lat":39.9,  "lon":116.4,  "lifeExp":77},
    {"country":"Japan",         "lat":35.7,  "lon":139.7,  "lifeExp":84},
    {"country":"Australia",     "lat":-35.3, "lon":149.1,  "lifeExp":83},
]
df = pd.DataFrame(data)

# degree formatters for nice ticks
def fmt_lon(x, pos=None): return f"{abs(x):.0f}°{'E' if x>=0 else 'W'}"
def fmt_lat(y, pos=None): return f"{abs(y):.0f}°{'N' if y>=0 else 'S'}"

plt.figure(figsize=(12,6))
sc = plt.scatter(df["lon"], df["lat"],
                 c=df["lifeExp"], cmap="viridis", s=120, alpha=0.9)

# light graticule every 30°
for lon in range(-180, 181, 30): plt.axvline(lon, lw=0.5, alpha=0.25)
for lat in range(-90,  91, 30): plt.axhline(lat, lw=0.5, alpha=0.25)

# annotate
for _, r in df.iterrows():
    plt.text(r["lon"]+2, r["lat"]+2, r["country"], fontsize=8)

plt.xlim(-180, 180); plt.ylim(-90, 90)
plt.gca().xaxis.set_major_formatter(FuncFormatter(fmt_lon))
plt.gca().yaxis.set_major_formatter(FuncFormatter(fmt_lat))
```

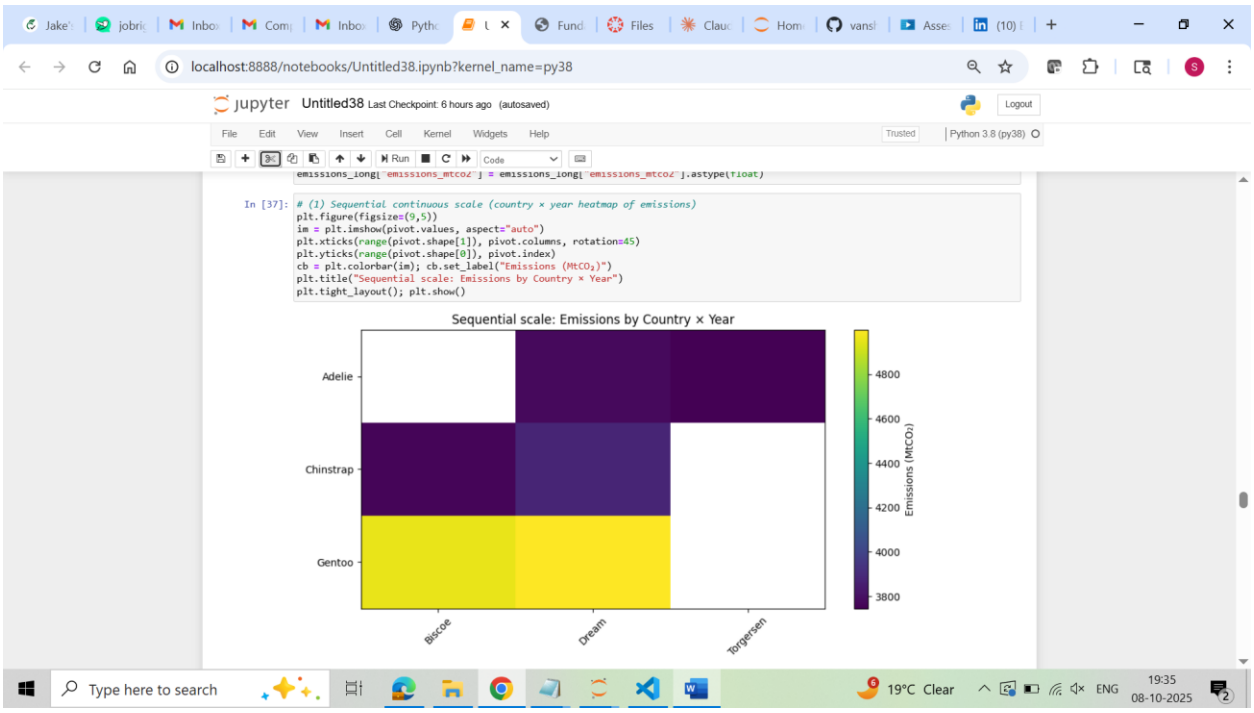
Type here to search

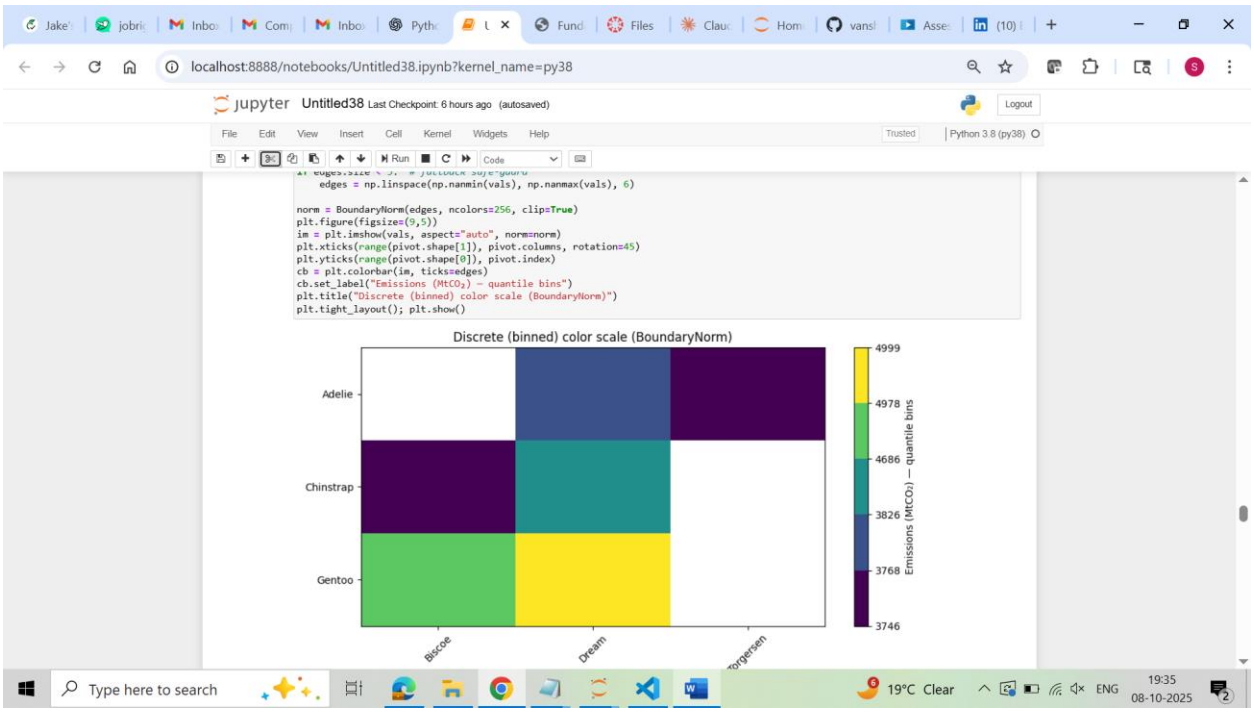
19°C Clear ENG 19:34 08-10-2025

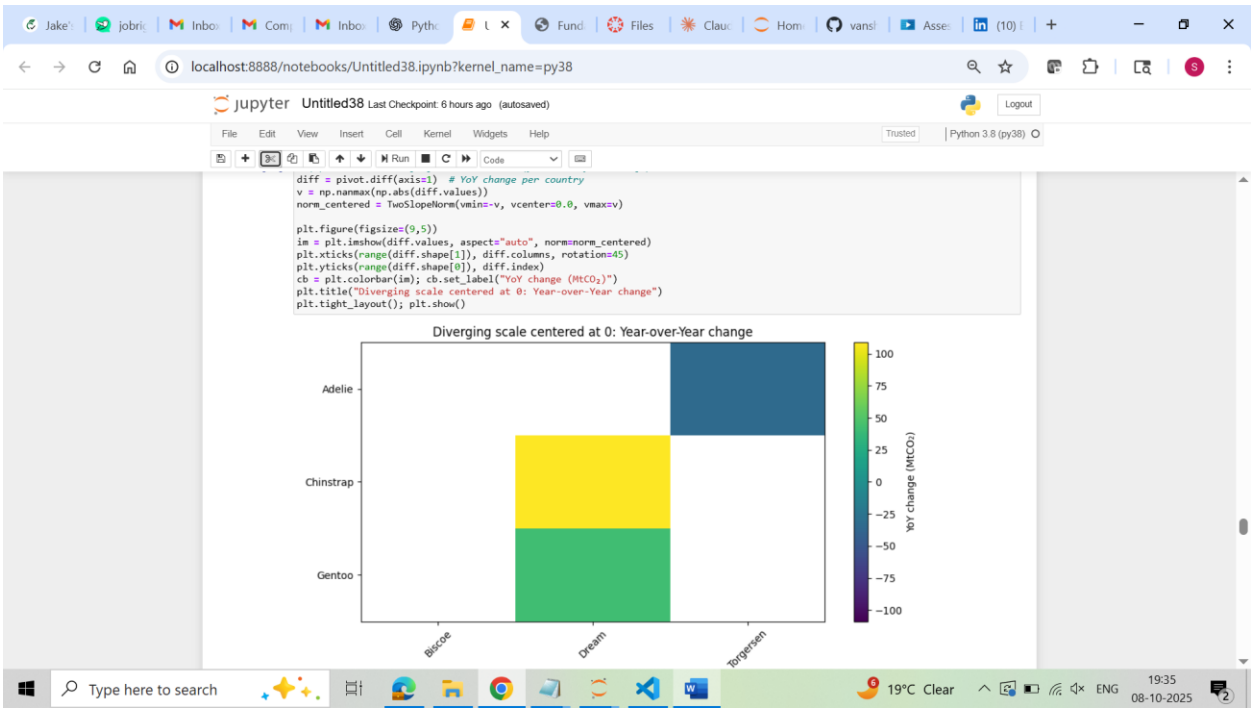

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
plt.gca().yaxis.set_major_formatter(FuncFormatter(fmt_lat))
plt.xlabel("Longitude"); plt.ylabel("Latitude")
plt.title("Life Expectancy (years) – Bubble Map (no installs)")
cb = plt.colorbar(sc); cb.set_label("Life expectancy (years)")
plt.tight_layout(); plt.show()
Life Expectancy (years) — Bubble Map (no installs)
Canada
United States
UK
Germany
China
Japan
Egypt
India
Mexico
Brazil
South Africa
Australia
80°N
60°N
40°N
20°N
0°N
20°S
40°S
60°S
80°S
Latitude
150°W
100°W
50°W
0°E
50°E
100°E
150°E
Longitude
82.5
80.0
77.5
75.0
72.5
70.0
67.5
65.0
Life expectancy (years)
Type here to search
19°C Clear
ENG
19:34
08-10-2025

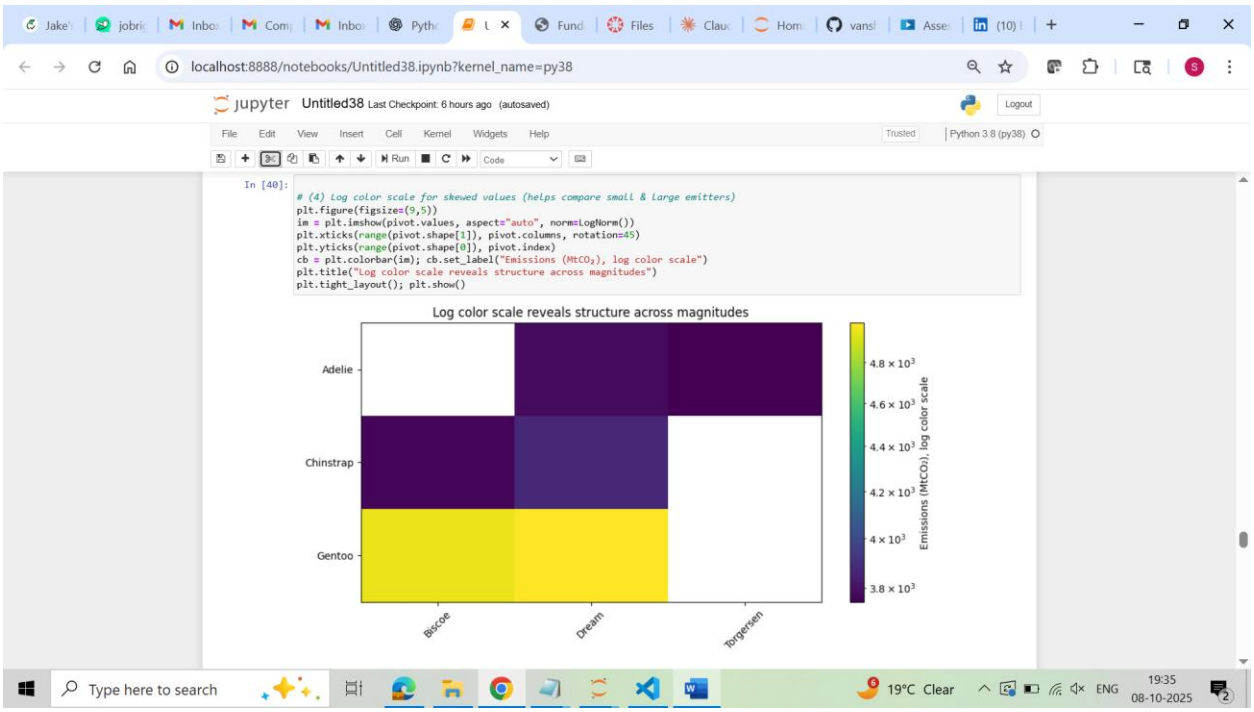
CHAPTER 4


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Code
from matplotlib.colors import BoundaryNorm, TwoSlopeNorm, LogNorm
np.random.seed(12)
# -------------------------
# 1) Build synthetic dataset
# -------------------------
countries = [
"United States","China","India","Germany","United Kingdom","Brazil",
"South Africa","Japan","Australia","Canada","France","Mexico"
]
years = list(range(2010, 2025))
rows = []
for country in countries:
# Draw a base emission level (in MtCO2); log-normal for wide positive spread
base = float(np.exp(np.random.normal(np.log(500), 0.9))) # around 500 Mt median but skewed
# A small country-specific yearly trend (positive or negative)
trend = np.random.normal(0.01, 0.02) # ±1-3% per year on average
val = base
for y in years:
# Apply multiplicative trend + noise
growth = np.exp(trend + np.random.normal(0, 0.06))
val = val * growth
rows.append({"country": country, "year": y, "emissions_mtco2": val})
emissions_long = pd.DataFrame(rows)
emissions_long["emissions_mtco2"] = emissions_long["emissions_mtco2"].astype(float)
In [37]:
# (1) Sequential continuous scale (country × year heatmap of emissions)
plt.figure(figsize=(9,5))
im = plt.imshow(pivot.values, aspect="auto")
plt.xticks(range(pivot.shape[1]), pivot.columns, rotation=45)
plt.yticks(range(pivot.shape[0]), pivot.index)
cb = plt.colorbar(im); cb.set_label("Emissions (MtCO2)")
plt.title("Sequential scale: Emissions by Country × Year")
plt.tight_layout(); plt.show()
Type here to search
19°C Clear
ENG
19:35
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
In [37]:
# (1) Sequential continuous scale (country × year heatmap of emissions)
plt.figure(figsize=(9,5))
im = plt.imshow(pivot.values, aspect="auto")
plt.xticks(range(pivot.shape[1]), pivot.columns, rotation=45)
plt.yticks(range(pivot.shape[0]), pivot.index)
cb = plt.colorbar(im); cb.set_label("Emissions (MtCO₂)")
plt.title("Sequential scale: Emissions by Country × Year")
plt.tight_layout(); plt.show()
Sequential scale: Emissions by Country × Year
Adelie
Chinstrap
Gentoo
Biscoe
Dream
Torgersen
4800
4600
4400
4200
4000
3800
Emissions (MtCO₂)
Type here to search
19°C Clear
ENG
19:35
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
edges = np.linspace(np.nanmin(vals), np.nanmax(vals), 6)
norm = BoundaryNorm(edges, ncolors=256, clip=True)
plt.figure(figsize=(9,5))
im = plt.imshow(vals, aspect="auto", norm=norm)
plt.xticks(range(pivot.shape[1]), pivot.columns, rotation=45)
plt.yticks(range(pivot.shape[0]), pivot.index)
cb = plt.colorbar(im, ticks=edges)
cb.set_label("Emissions (MtCO₂) – quantile bins")
plt.title("Discrete (binned) color scale (BoundaryNorm)")
plt.tight_layout(); plt.show()
Discrete (binned) color scale (BoundaryNorm)
Adelie
Chinstrap
Gentoo
Biscoe
Dream
Torgersen
4999
4978
4686
3826
3768
3746
Emissions (MtCO₂) – quantile bins
Type here to search
19°C Clear
ENG
19:35
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
diff = pivot.diff(axis=1) # YoY change per country
v = np.nanmax(np.abs(diff.values))
norm_centered = TwoSlopeNorm(vmin=-v, vcenter=0.0, vmax=v)
plt.figure(figsize=(9,5))
im = plt.imshow(diff.values, aspect="auto", norm=norm_centered)
plt.xticks(range(diff.shape[1]), diff.columns, rotation=45)
plt.yticks(range(diff.shape[0]), diff.index)
cb = plt.colorbar(im); cb.set_label("YoY change (MtCO₂)")
plt.title("Diverging scale centered at 0: Year-over-Year change")
plt.tight_layout(); plt.show()
Diverging scale centered at 0: Year-over-Year change
Adelie
Chinstrap
Gentoo
Biscoe
Dream
Torgersen
100
75
50
25
0
−25
−50
−75
−100
YoY change (MtCO₂)
Type here to search
19°C Clear
ENG
19:35
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Code
In [40]:
# (4) Log color scale for skewed values (helps compare small & large emitters)
plt.figure(figsize=(9,5))
im = plt.imshow(pivot.values, aspect="auto", norm=LogNorm())
plt.xticks(range(pivot.shape[1]), pivot.columns, rotation=45)
plt.yticks(range(pivot.shape[0]), pivot.index)
cb = plt.colorbar(im); cb.set_label("Emissions (MtCO₂), log color scale")
plt.title("Log color scale reveals structure across magnitudes")
plt.tight_layout(); plt.show()
Log color scale reveals structure across magnitudes
Adelie
Chinstrap
Gentoo
Biscoe
Dream
Torgersen
4.8 × 10³
4.6 × 10³
4.4 × 10³
4.2 × 10³
4 × 10³
3.8 × 10³
Emissions (MtCO₂), log color scale
Type here to search
19°C Clear
ENG
19:35
08-10-2025

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

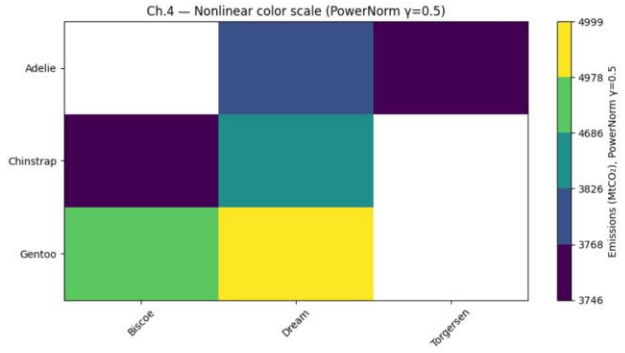
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run Code

In [43]:
```
# 8) Nonlinear PowerNorm (gamma < 1 brightens darker mid-range)
gamma = 0.5

plt.figure(figsize=(9,5))
im = plt.imshow(vals, aspect="auto", norm=norm)
plt.xticks(range(pivot.shape[1]), pivot.columns, rotation=45)
plt.yticks(range(pivot.shape[0]), pivot.index)
cb = plt.colorbar(im); cb.set_label(f"Emissions (MtCO₂), PowerNorm γ={gamma}")
plt.title(f"Ch.4 — Nonlinear color scale (PowerNorm γ={gamma})")
plt.tight_layout(); plt.show()
```



Type here to search

19°C Clear ENG 19:36 08-10-2025

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

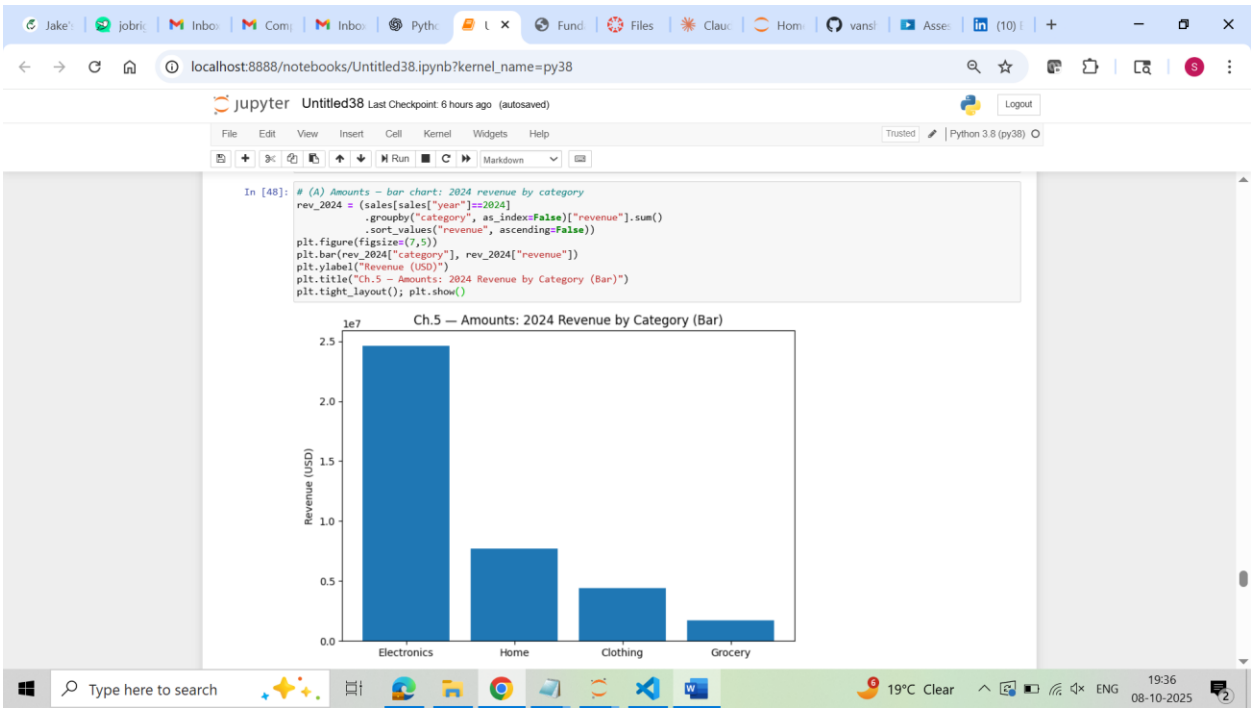
Trusted | Python 3.8 (py38)

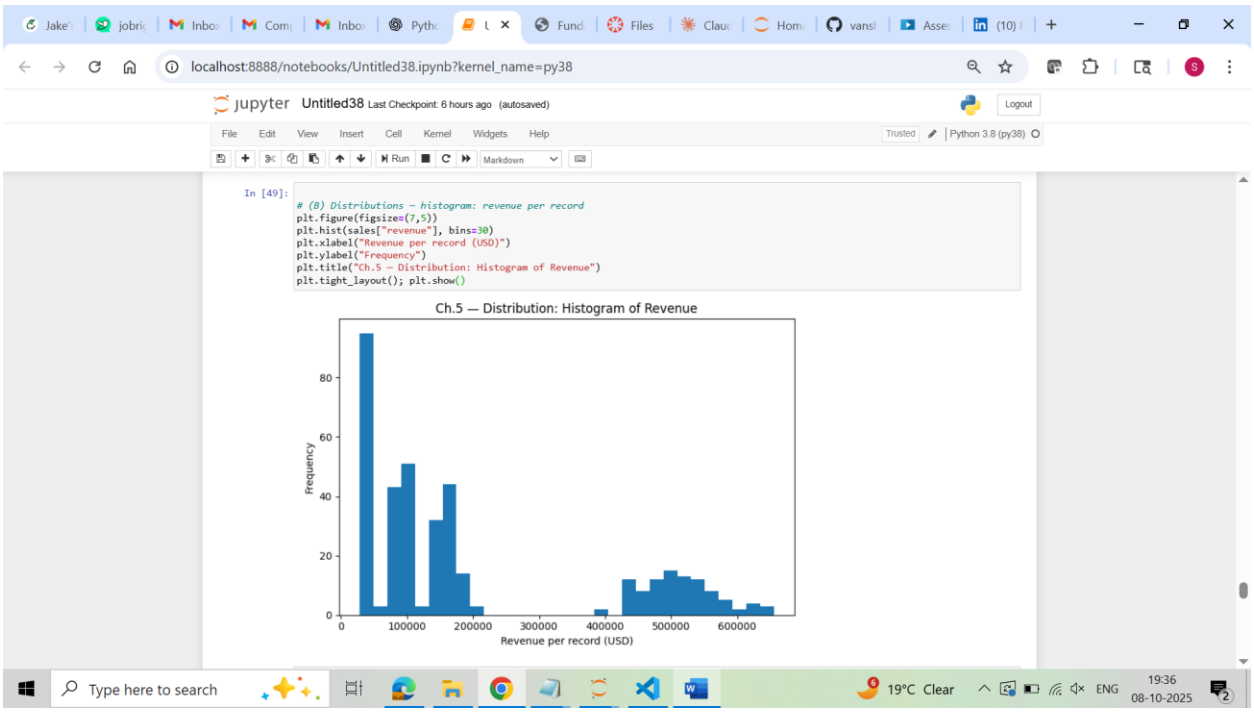
Run Markdown

```
# ------------------------
# 1) Create a submit-ready dataset
# ------------------------
months = pd.period_range("2023-01", "2024-12", freq="M").to_timestamp()
categories = ["Electronics", "Clothing", "Grocery", "Home"]
regions = ["North", "South", "East", "West"]

cat_base_units = {"Electronics": 950, "Clothing": 1600, "Grocery": 4200, "Home": 1200}
cat_base_price = {"Electronics": 520.0, "Clothing": 55.0, "Grocery": 8.5, "Home": 130.0}
region_mult = {"North": 1.05, "South": 0.95, "East": 1.00, "West": 1.00}

rows = []
for dt in months:
    # simple seasonality: back-to-school (Aug/Sep) + holiday (Nov/Dec) lift
    m = dt.month
    season = 1.0 + 0.08*np.sin((m-1)/12*2*np.pi) + (0.18 if m in [11,12] else 0.0) + (0.06 if m in [8,9] else 0.0)
    for cat in categories:
        for reg in regions:
            units = cat_base_units[cat] * region_mult[reg] * season * np.exp(np.random.normal(0, 0.08))
            price = cat_base_price[cat] * np.exp(np.random.normal(0, 0.03))
            revenue = units * price
            marketing = 0.10*revenue * np.exp(np.random.normal(0, 0.25))
            rows.append({
                "date": dt,
                "year": dt.year,
                "month": dt.month,
                "region": reg,
                "category": cat,
                "units_sold": round(units, 0),
                "price_per_unit": round(price, 2),
                "revenue": revenue,
                "marketing_spend": marketing
            })

sales = pd.DataFrame(rows)
sales["revenue"] = sales["revenue"].astype(float)
sales["marketing_spend"] = sales["marketing_spend"].astype(float)
```

Type here to search

19°C Clear ENG 19:36 08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Markdown
In [48]:
# (A) Amounts — bar chart: 2024 revenue by category
rev_2024 = (sales[sales["year"]==2024]
.groupby("category", as_index=False)["revenue"].sum()
.sort_values("revenue", ascending=False))
plt.figure(figsize=(7,5))
plt.bar(rev_2024["category"], rev_2024["revenue"])
plt.ylabel("Revenue (USD)")
plt.title("Ch.5 — Amounts: 2024 Revenue by Category (Bar)")
plt.tight_layout(); plt.show()
Ch.5 — Amounts: 2024 Revenue by Category (Bar)
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Revenue (USD)
Electronics
Home
Clothing
Grocery
Type here to search
19°C Clear
ENG
19:36
08-10-2025


localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38
jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3.8 (py38)
Run Markdown
In [49]:
# (B) Distributions — histogram: revenue per record
plt.figure(figsize=(7,5))
plt.hist(sales["revenue"], bins=30)
plt.xlabel("Revenue per record (USD)")
plt.ylabel("Frequency")
plt.title("Ch.5 — Distribution: Histogram of Revenue")
plt.tight_layout(); plt.show()
Ch.5 — Distribution: Histogram of Revenue
80
60
40
20
0
Frequency
0
100000
200000
300000
400000
500000
600000
Revenue per record (USD)
Type here to search
19°C Clear
ENG
19:36
08-10-2025

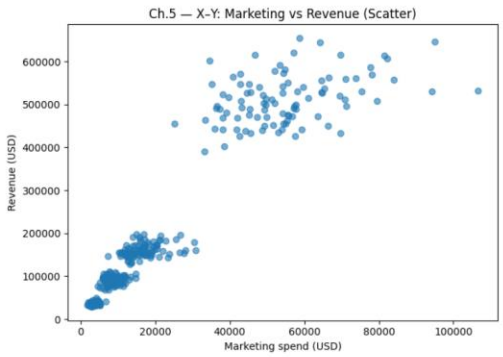
localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

In [50]:
```
# (C) X-Y relationships – scatter: marketing vs revenue
plt.figure(figsize=(7,5))
plt.scatter(sales["marketing_spend"], sales["revenue"], alpha=0.6)
plt.xlabel("Marketing spend (USD)")
plt.ylabel("Revenue (USD)")
plt.title("Ch.5 – X-Y: Marketing vs Revenue (Scatter)")
plt.tight_layout(); plt.show()
```



Type here to search 19°C Clear ENG 19:37 08-10-2025

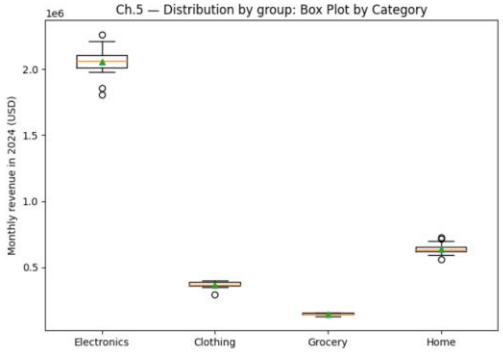
localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

In [51]:
```
# (D) Distributions by group – box plot: monthly revenue per category (2024)
rev_month_cat = (sales[sales["year"]==2024]
                 .groupby(["category","month"], as_index=False)["revenue"].sum())
groups = [rev_month_cat[rev_month_cat["category"]==c]["revenue"].values for c in categories]
plt.figure(figsize=(7,5))
plt.boxplot(groups, labels=categories, showmeans=True)
plt.ylabel("Monthly revenue in 2024 (USD)")
plt.title("Ch.5 – Distribution by group: Box Plot by Category")
plt.tight_layout(); plt.show()
```



Type here to search 19°C Clear ENG 19:37 08-10-2025

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

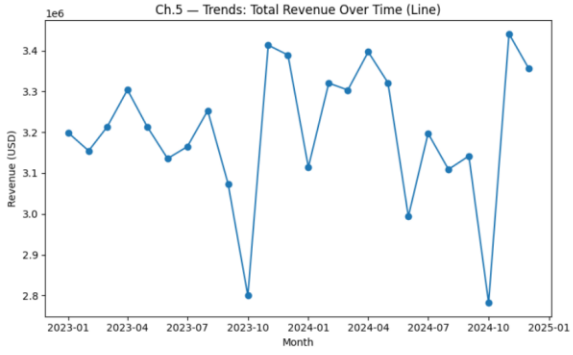
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

In [52]:
```
# (E) Trends — line chart: total monthly revenue
rev_month = sales.groupby("date", as_index=False)["revenue"].sum()
plt.figure(figsize=(8,5))
plt.plot(rev_month["date"], rev_month["revenue"], marker="o")
plt.xlabel("Month")
plt.ylabel("Revenue (USD)")
plt.title("Ch.5 — Trends: Total Revenue Over Time (Line)")
plt.tight_layout(); plt.show()
```



Type here to search

19°C Clear ENG 19:37 08-10-2025

localhost:8888/notebooks/Untitled38.ipynb?kernel_name=py38

jupyter Untitled38 Last Checkpoint: 6 hours ago (autosaved)

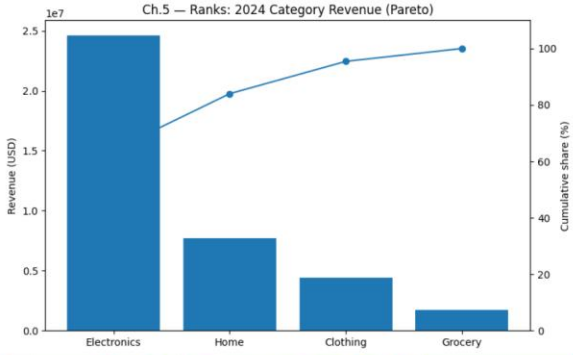
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

```
fig, ax1 = plt.subplots(figsize=(8,5))
ax1.bar(rev_share["category"], rev_share["revenue"])
ax1.set_ylabel("Revenue (USD)")
ax1.set_title("Ch.5 — Ranks: 2024 Category Revenue (Pareto)")

ax2 = ax1.twinx()
ax2.plot(rev_share["category"], rev_share["cum_share"]*100, marker="o")
ax2.set_ylabel("Cumulative share (%)")
ax2.set_ylim(0, 110)
plt.tight_layout(); plt.show()
```



Type here to search

19°C Clear ENG 19:37 08-10-2025