localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3.8 (py38)

Run Markdown

CHAPTER 6

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Create synthetic dataset with 100 records
np.random.seed(42)
data = pd.DataFrame({
    'Category': np.random.choice(['Electronics', 'Clothing', 'Home Decor', 'Books', 'Toys'], 100),
    'Region': np.random.choice(['West', 'East', 'North', 'South'], 100),
    'Sales': np.random.randint(20, 150, 100)
})
```

In [4]:
```
# 1. Bar Plot - Total sales by category
plt.figure(figsize=(8,5))
total_sales_by_cat = data.groupby('Category')['Sales'].sum().sort_values()
total_sales_by_cat.plot(kind='bar', color='skyblue')
plt.title('Total Sales by Category')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.show()
```

Type here to search 20°C ENG 19:02 28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

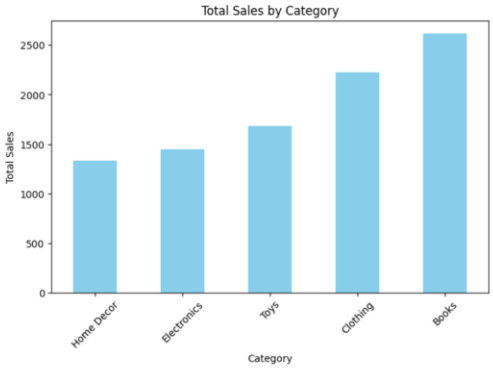
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3.8 (py38)

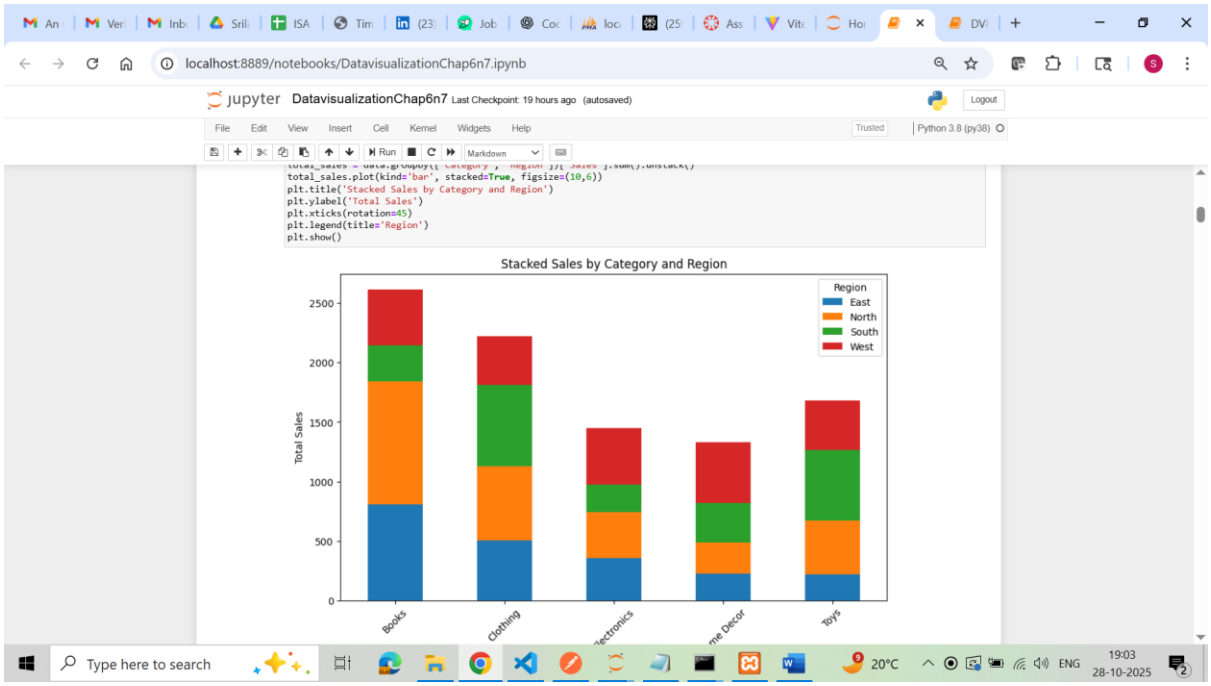
Run Markdown



In [ ]:

In [6]:
```
# 3. Stacked Bar Plot - Total sales by category stacked by region
```

Type here to search 20°C ENG 19:02 28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Python 3.8 (py38)
total_sales.plot(kind='bar', stacked=True, figsize=(10,6))
plt.title('Stacked Sales by Category and Region')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.legend(title='Region')
plt.show()
Stacked Sales by Category and Region
Region
East
North
South
West
Total Sales
0
500
1000
1500
2000
2500
Books
Clothing
Toys
Type here to search
20°C
ENG
19:03
28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

Logout

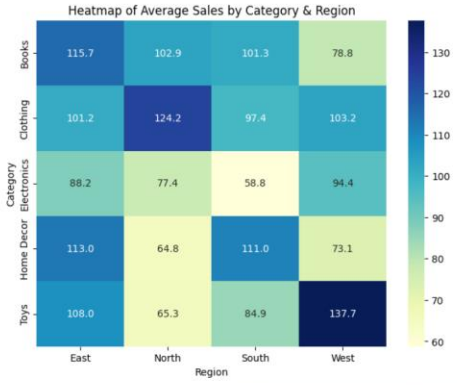
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

In [10]:

```
# 5. Heatmap - Average sales by category and region
pivot_data = data.pivot_table(index='Category', columns='Region', values='Sales', aggfunc='mean')
plt.figure(figsize=(8,6))
sns.heatmap(pivot_data, annot=True, cmap='YlGnBu', fmt=".1f")
plt.title('Heatmap of Average Sales by Category & Region')
plt.show()
```



Type here to search

20°C ENG 19:03 28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

Logout

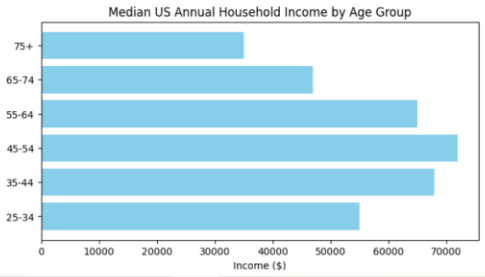
File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

In [11]:

```
# 1. Simple horizontal bar plot ordered by income
plt.figure(figsize=(8,4))
plt.barh(data_income['Age Group'], data_income['Median Income'], color='skyblue')
plt.title('Median US Annual Household Income by Age Group')
plt.xlabel('Income ($)')
plt.show()

# Dataset 2: Median Income by Age Group and Race
data_income_race = pd.DataFrame({
    'Age Group': np.repeat(['25-34', '35-44', '45-54', '55-64', '65-74', '75+'], 4),
    'Race': ['Asian', 'White', 'Hispanic', 'Black'] * 6,
    'Median Income': [68000, 55000, 45000, 43000, 78000, 62000, 47000, 46000, 82000, 70000, 52000, 49000,
                      76000, 68000, 53000, 50000, 67000, 55000, 48000, 45000, 54000, 45000, 40000, 39000]
})
```



Type here to search

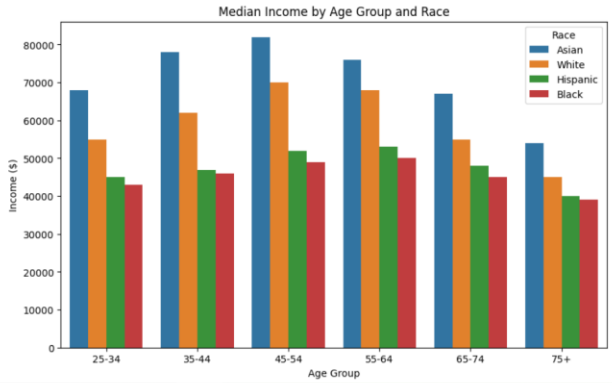
20°C ENG 19:03 28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```
plt.figure(figsize=(10,6))
sns.barplot(x='Age Group', y='Median Income', hue='Race', data=data_income_race)
plt.title('Median Income by Age Group and Race')
plt.ylabel('Income ($)')
plt.show()
```



Type here to search

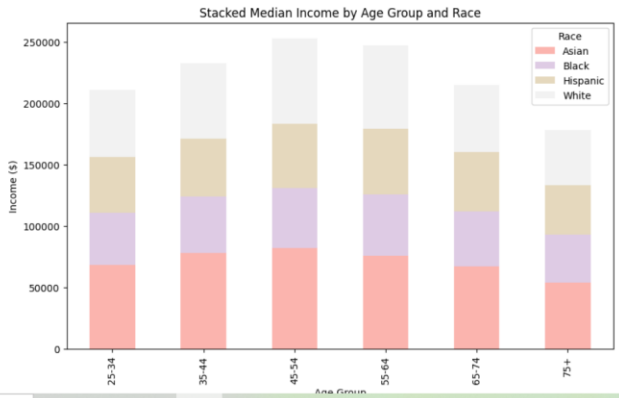
Live ENG 19:07 28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

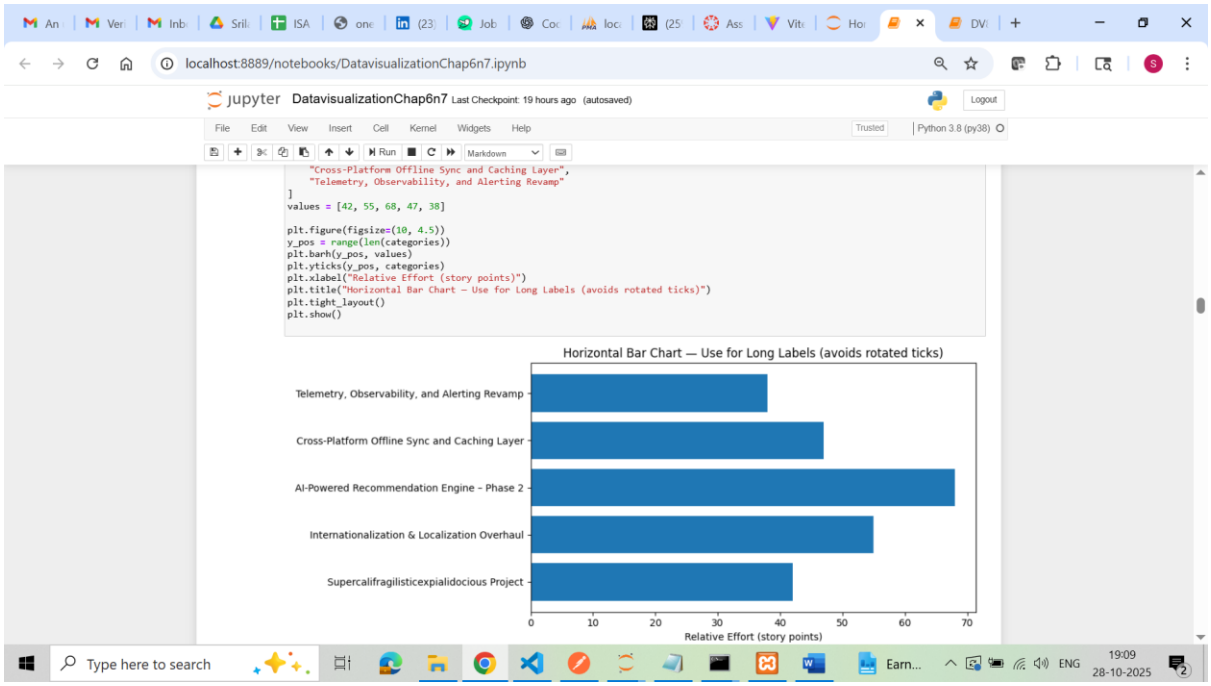
File Edit View Insert Cell Kernel Widgets Help

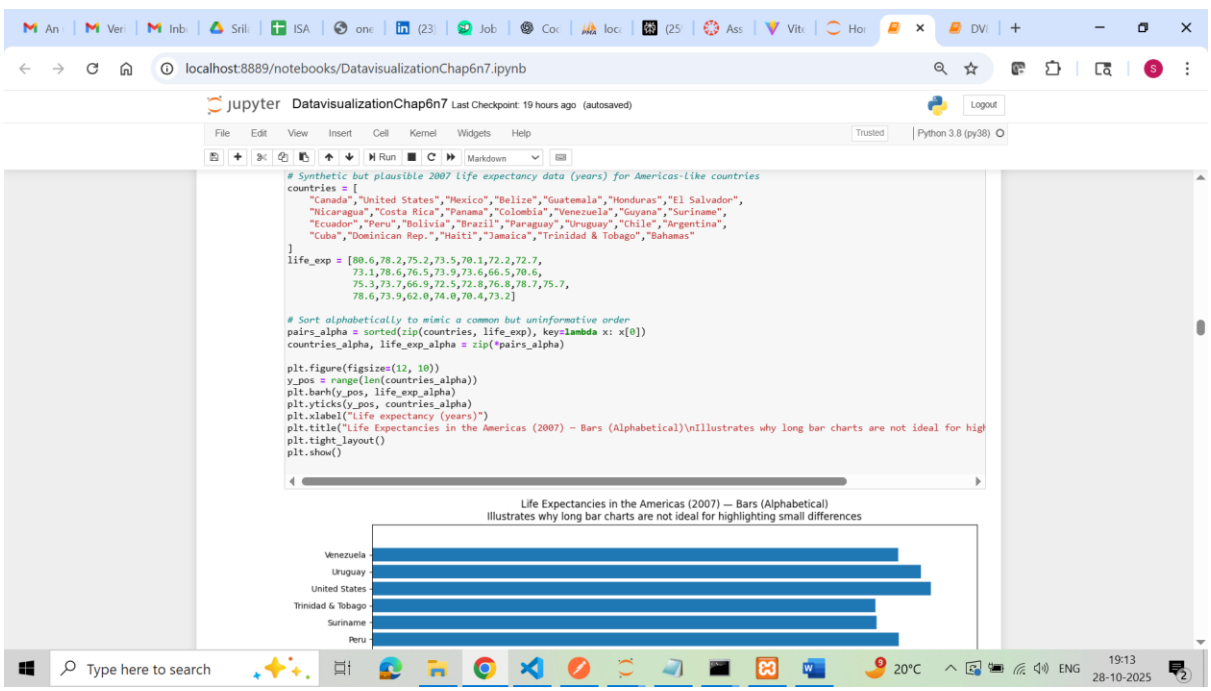
```
income_pivot = data_income_race.pivot(index='Age Group', columns='Race', values='Median Income')
income_pivot.plot(kind='bar', stacked=True, figsize=(10,6), colormap='Pastel1')
plt.title('Stacked Median Income by Age Group and Race')
plt.ylabel('Income ($)')
plt.show()
```

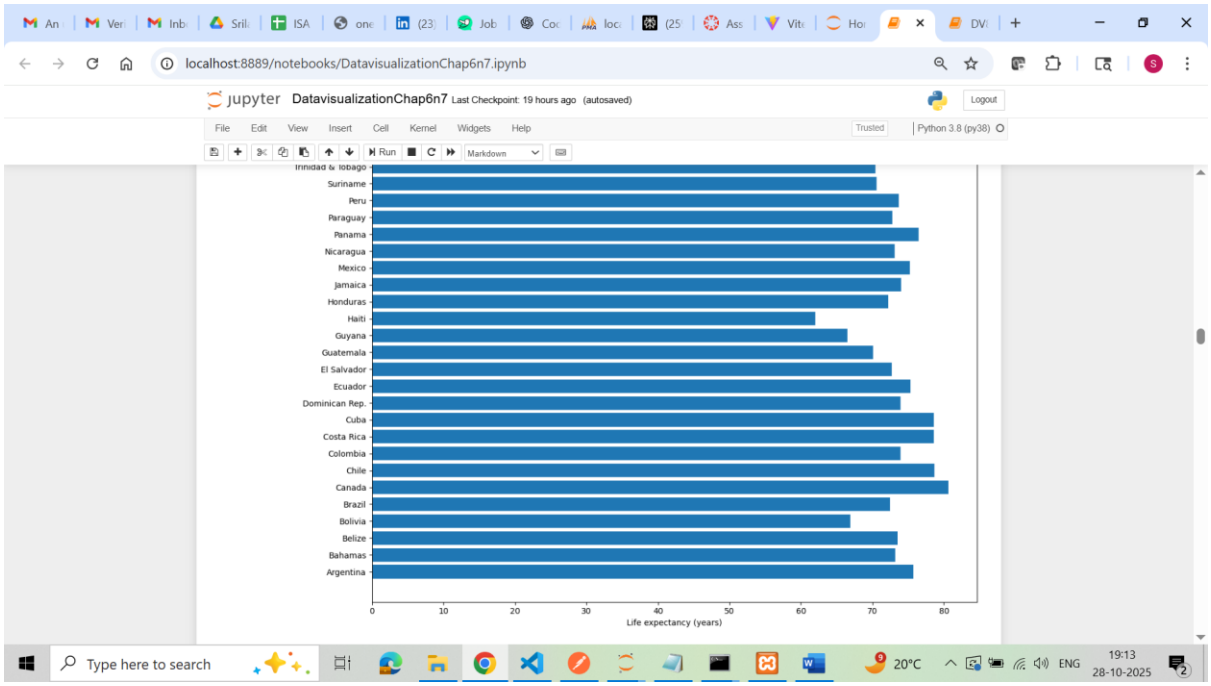


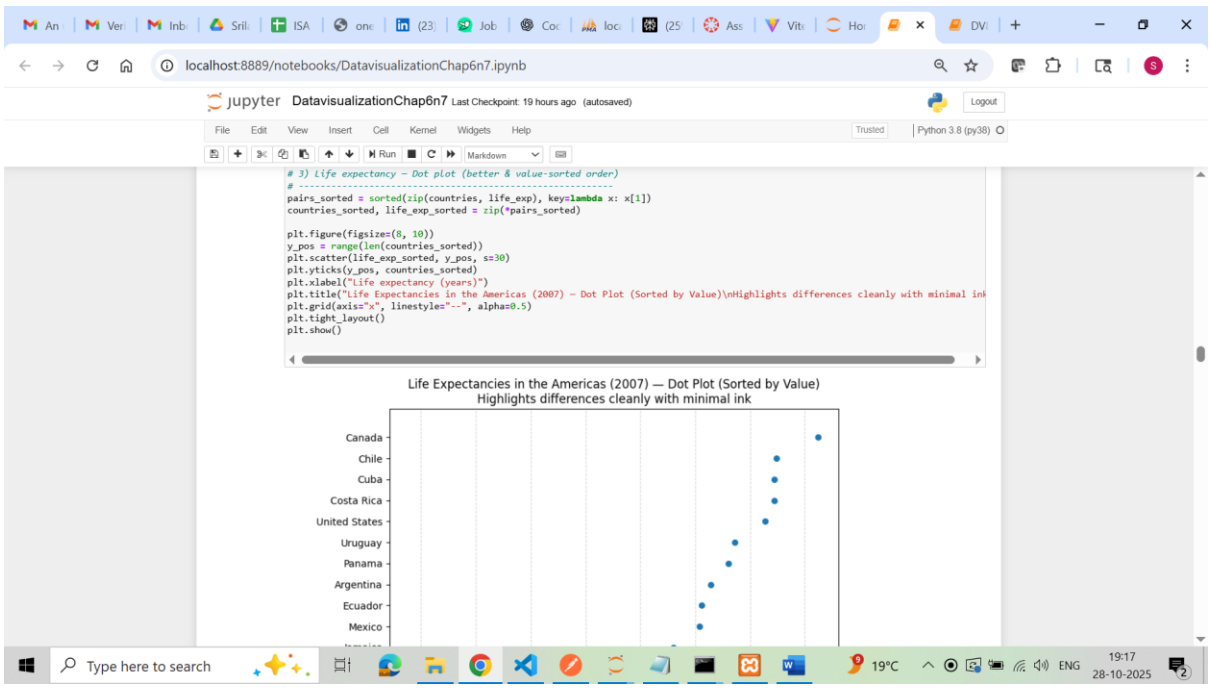
Type here to search

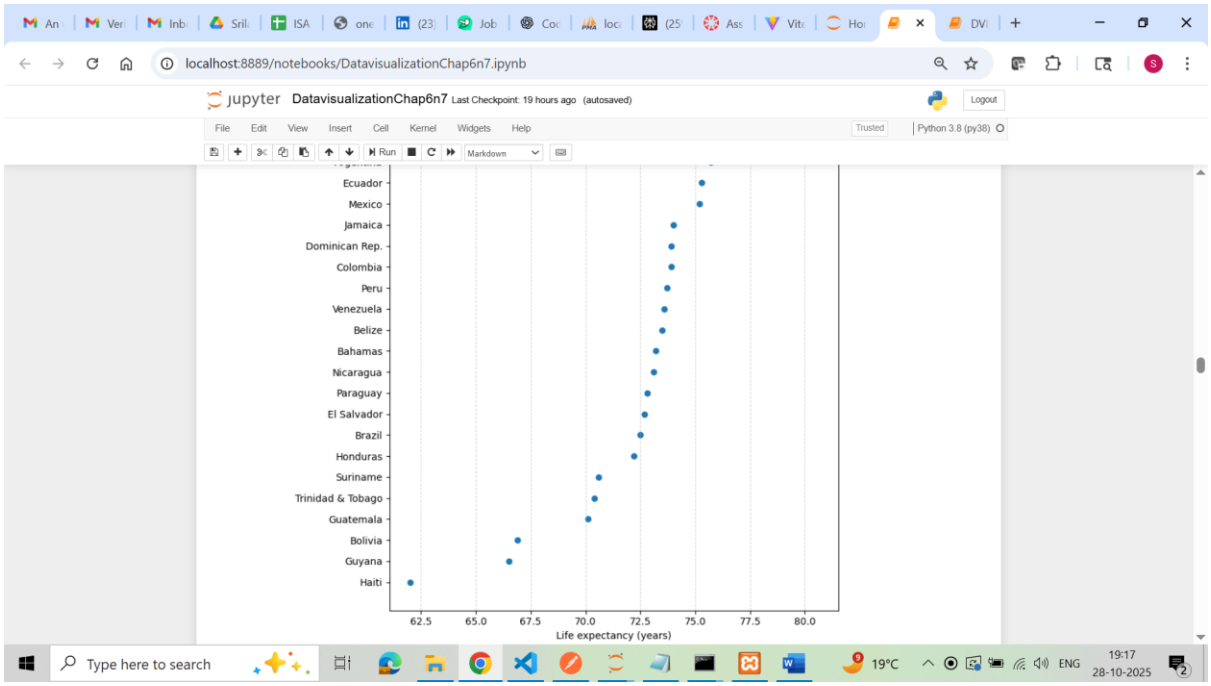
Live ENG 19:07 28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
"Cross-Platform Offline Sync and Caching Layer",
"Telemetry, Observability, and Alerting Revamp"
]
values = [42, 55, 68, 47, 38]
plt.figure(figsize=(10, 4.5))
y_pos = range(len(categories))
plt.barh(y_pos, values)
plt.yticks(y_pos, categories)
plt.xlabel("Relative Effort (story points)")
plt.title("Horizontal Bar Chart — Use for Long Labels (avoids rotated ticks)")
plt.tight_layout()
plt.show()
Horizontal Bar Chart — Use for Long Labels (avoids rotated ticks)
Telemetry, Observability, and Alerting Revamp
Cross-Platform Offline Sync and Caching Layer
AI-Powered Recommendation Engine – Phase 2
Internationalization & Localization Overhaul
Supercalifragilisticexpialidocious Project
0
10
20
30
40
50
60
70
Relative Effort (story points)
Type here to search
Earn...
ENG
19:09
28-10-2025

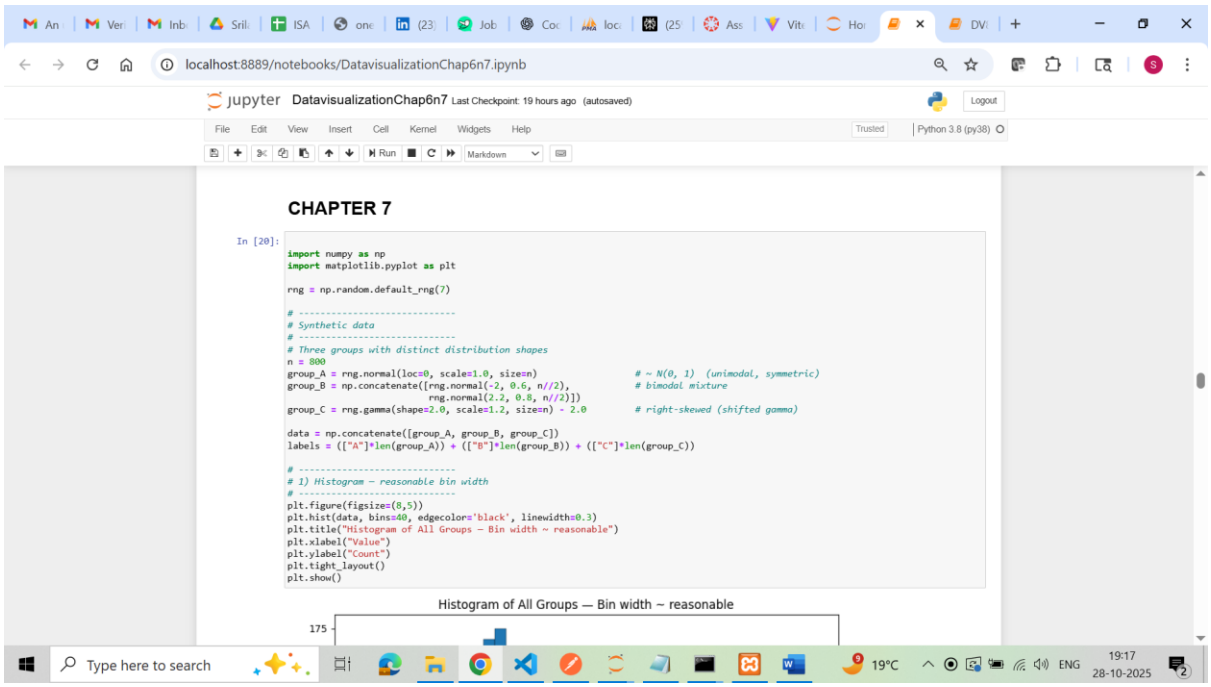

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
# Synthetic but plausible 2007 life expectancy data (years) for Americas-like countries
countries = [
"Canada","United States","Mexico","Belize","Guatemala","Honduras","El Salvador",
"Nicaragua","Costa Rica","Panama","Colombia","Venezuela","Guyana","Suriname",
"Ecuador","Peru","Bolivia","Brazil","Paraguay","Uruguay","Chile","Argentina",
"Cuba","Dominican Rep.","Haiti","Jamaica","Trinidad & Tobago","Bahamas"
]
life_exp = [80.6,78.2,75.2,73.5,70.1,72.2,72.7,
73.1,78.6,76.5,73.9,73.6,66.5,70.6,
75.3,73.7,66.9,72.5,72.8,76.8,78.7,75.7,
78.6,73.9,62.0,74.0,70.4,73.2]
# Sort alphabetically to mimic a common but uninformative order
pairs_alpha = sorted(zip(countries, life_exp), key=lambda x: x[0])
countries_alpha, life_exp_alpha = zip(*pairs_alpha)
plt.figure(figsize=(12, 10))
y_pos = range(len(countries_alpha))
plt.barh(y_pos, life_exp_alpha)
plt.yticks(y_pos, countries_alpha)
plt.xlabel("Life expectancy (years)")
plt.title("Life Expectancies in the Americas (2007) – Bars (Alphabetical)\nIllustrates why long bar charts are not ideal for high
plt.tight_layout()
plt.show()
Life Expectancies in the Americas (2007) — Bars (Alphabetical)
Illustrates why long bar charts are not ideal for highlighting small differences
Venezuela
Uruguay
United States
Trinidad & Tobago
Suriname
Peru
Type here to search
20°C
ENG
19:13
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
Trinidad & Tobago
Suriname
Peru
Paraguay
Panama
Nicaragua
Mexico
Jamaica
Honduras
Haiti
Guyana
Guatemala
El Salvador
Ecuador
Dominican Rep.
Cuba
Costa Rica
Colombia
Chile
Canada
Brazil
Bolivia
Belize
Bahamas
Argentina
0
10
20
30
40
50
60
70
80
Life expectancy (years)
Type here to search
20°C
ENG
19:13
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
# 3) Life expectancy — Dot plot (better & value-sorted order)
pairs_sorted = sorted(zip(countries, life_exp), key=lambda x: x[1])
countries_sorted, life_exp_sorted = zip(*pairs_sorted)
plt.figure(figsize=(8, 10))
y_pos = range(len(countries_sorted))
plt.scatter(life_exp_sorted, y_pos, s=30)
plt.yticks(y_pos, countries_sorted)
plt.xlabel("Life expectancy (years)")
plt.title("Life Expectancies in the Americas (2007) — Dot Plot (Sorted by Value)\nHighlights differences cleanly with minimal ink
plt.grid(axis="x", linestyle="--", alpha=0.5)
plt.tight_layout()
plt.show()
Life Expectancies in the Americas (2007) — Dot Plot (Sorted by Value)
Highlights differences cleanly with minimal ink
Canada
Chile
Cuba
Costa Rica
United States
Uruguay
Panama
Argentina
Ecuador
Mexico
Type here to search
19°C
ENG
19:17
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Ecuador
Mexico
Jamaica
Dominican Rep.
Colombia
Peru
Venezuela
Belize
Bahamas
Nicaragua
Paraguay
El Salvador
Brazil
Honduras
Suriname
Trinidad & Tobago
Guatemala
Bolivia
Guyana
Haiti
62.5
65.0
67.5
70.0
72.5
75.0
77.5
80.0
Life expectancy (years)
Type here to search
19:17
28-10-2025

CHAPTER 7


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
CHAPTER 7
In [20]:
import numpy as np
import matplotlib.pyplot as plt
rng = np.random.default_rng(7)
# ----------------------------
# Synthetic data
# ----------------------------
# Three groups with distinct distribution shapes
n = 800
group_A = rng.normal(loc=0, scale=1.0, size=n) # ~ N(0, 1) (unimodal, symmetric)
group_B = np.concatenate([rng.normal(-2, 0.6, n//2), # bimodal mixture
rng.normal(2.2, 0.8, n//2)])
group_C = rng.gamma(shape=2.0, scale=1.2, size=n) - 2.0 # right-skewed (shifted gamma)
data = np.concatenate([group_A, group_B, group_C])
labels = (["A"]*len(group_A)) + (["B"]*len(group_B)) + (["C"]*len(group_C))
# ----------------------------
# 1) Histogram – reasonable bin width
# ----------------------------
plt.figure(figsize=(8,5))
plt.hist(data, bins=40, edgecolor='black', linewidth=0.3)
plt.title("Histogram of All Groups – Bin width ~ reasonable")
plt.xlabel("Value")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
Histogram of All Groups — Bin width ~ reasonable
175
Type here to search
19:17
28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

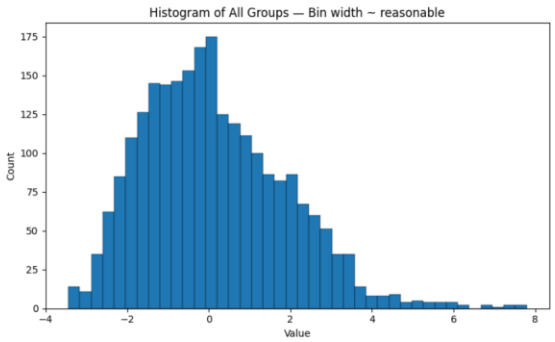
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3.8 (py38)

Run Markdown

```
plt.figure(figsize=(8,5))
plt.hist(data, bins=40, edgecolor='black', linewidth=0.3)
plt.title("Histogram of All Groups — Bin width ~ reasonable")
plt.xlabel("Value")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```



In [21]:

Type here to search

19°C ENG 19:17 28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

Logout

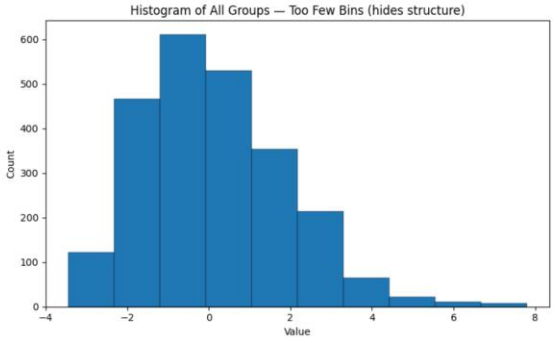
File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3.8 (py38)

Run Markdown

In [21]:

```
# 2) Histogram — too few bins (over-smoothing)
# ------------------------------
plt.figure(figsize=(8,5))
plt.hist(data, bins=10, edgecolor='black', linewidth=0.3)
plt.title("Histogram of All Groups — Too Few Bins (hides structure)")
plt.xlabel("Value")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```



Type here to search

19°C ENG 19:17 28-10-2025

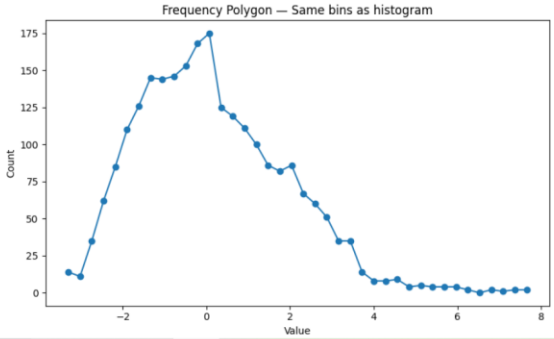
localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

In [22]:
```
# 3) Frequency polygon (line version of histogram)
# ------------------------------
counts, bin_edges = np.histogram(data, bins=40)
bin_centers = 0.5 * (bin_edges[1:] + bin_edges[:-1])
plt.figure(figsize=(8,5))
plt.plot(bin_centers, counts, marker='o', linestyle='-')
plt.title("Frequency Polygon – Same bins as histogram")
plt.xlabel("Value")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
```



Type here to search 19°C ENG 19:18 28-10-2025

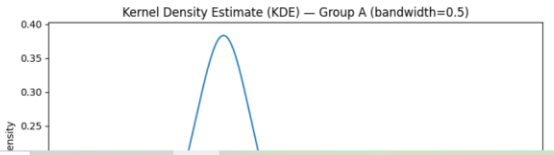
localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

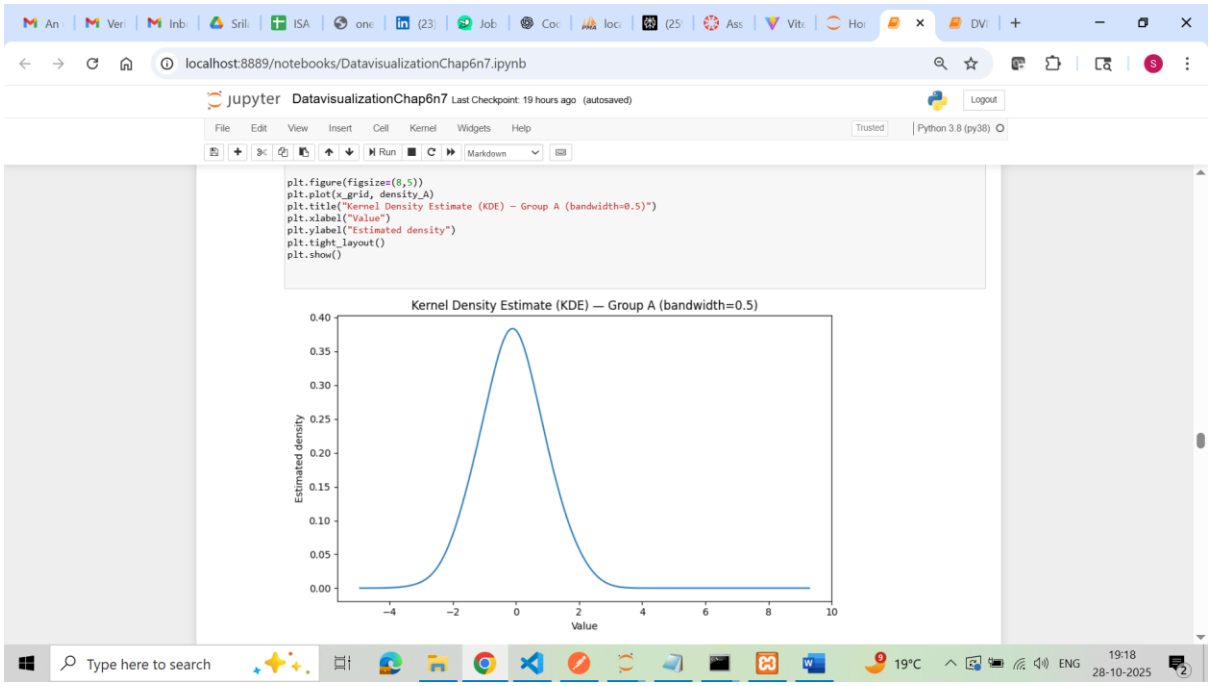
File Edit View Insert Cell Kernel Widgets Help

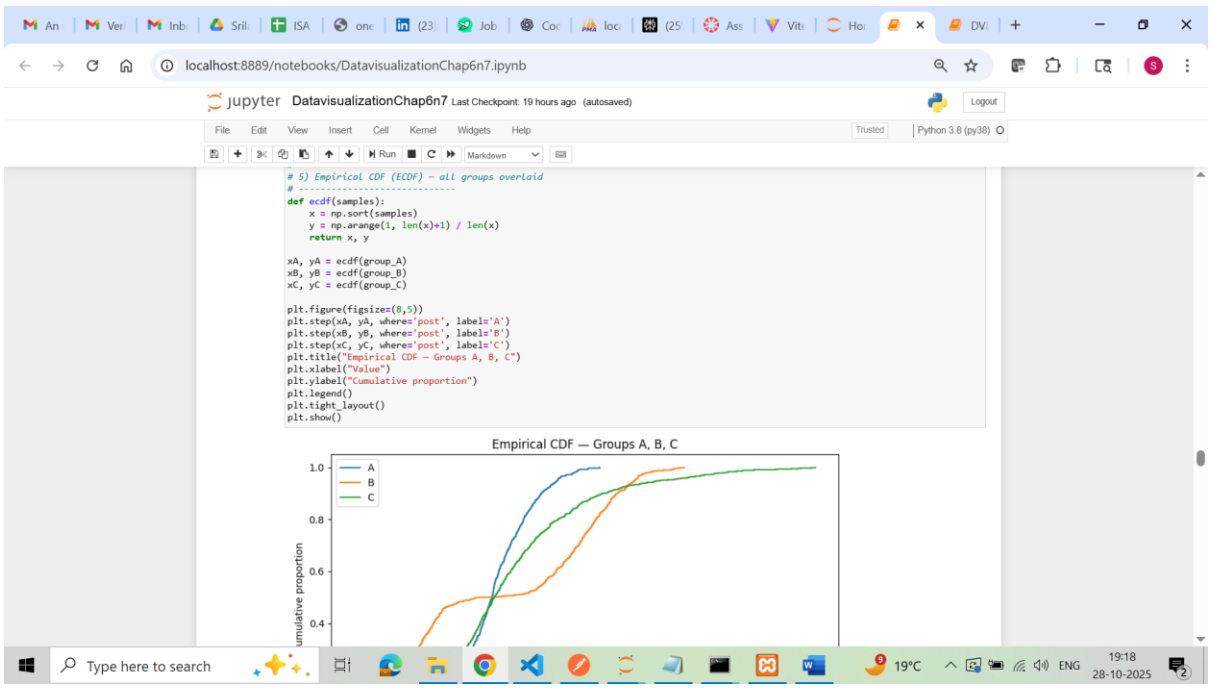
Trusted | Python 3.8 (py38)

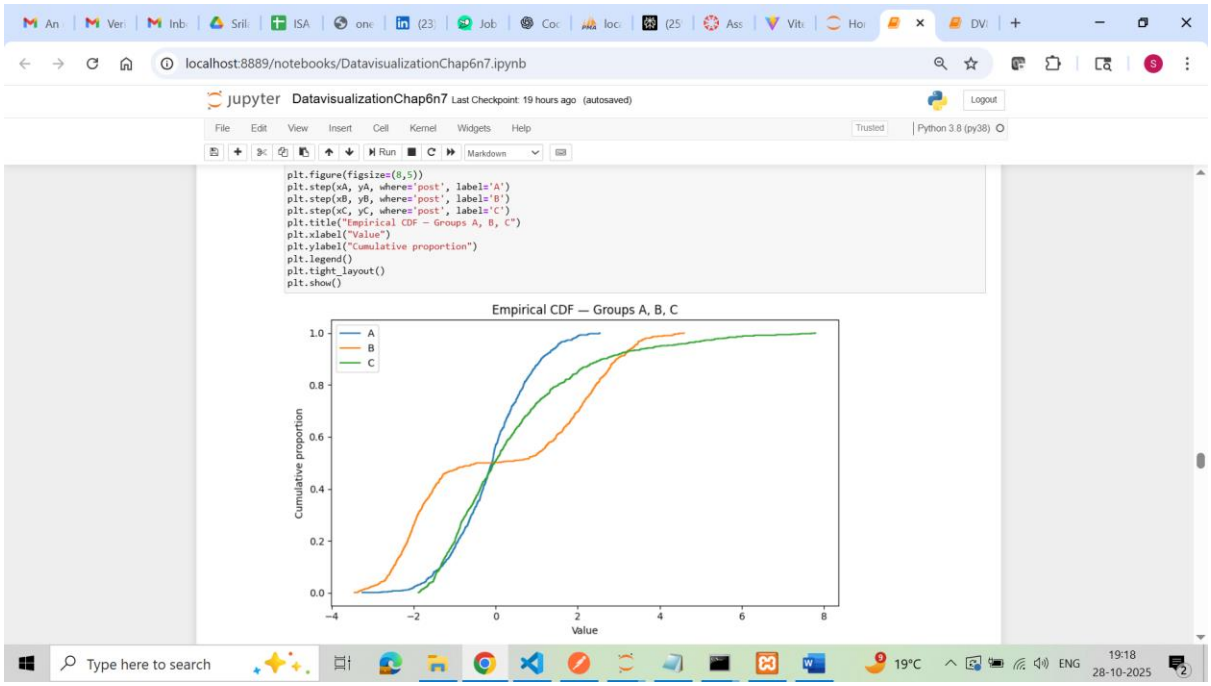
In [23]:
```
# ------------------------------
# 4) Simple Gaussian KDE for a single group (A)
# ------------------------------
def kde_gaussian(x, samples, bandwidth):
    # Gaussian kernel density estimate evaluated at x
    x = x[:, None]  # column
    diff = (x - samples[None, :]) / bandwidth
    kernel_vals = np.exp(-0.5 * diff**2) / np.sqrt(2*np.pi)
    # Average and scale by bandwidth
    return kernel_vals.mean(axis=1) / bandwidth

x_grid = np.linspace(min(data)-1.5, max(data)+1.5, 600)
bandwidth = 0.5  # try a moderate bandwidth
density_A = kde_gaussian(x_grid, group_A, bandwidth)

plt.figure(figsize=(8,5))
plt.plot(x_grid, density_A)
plt.title("Kernel Density Estimate (KDE) – Group A (bandwidth=0.5)")
plt.xlabel("Value")
plt.ylabel("Estimated density")
plt.tight_layout()
plt.show()
```

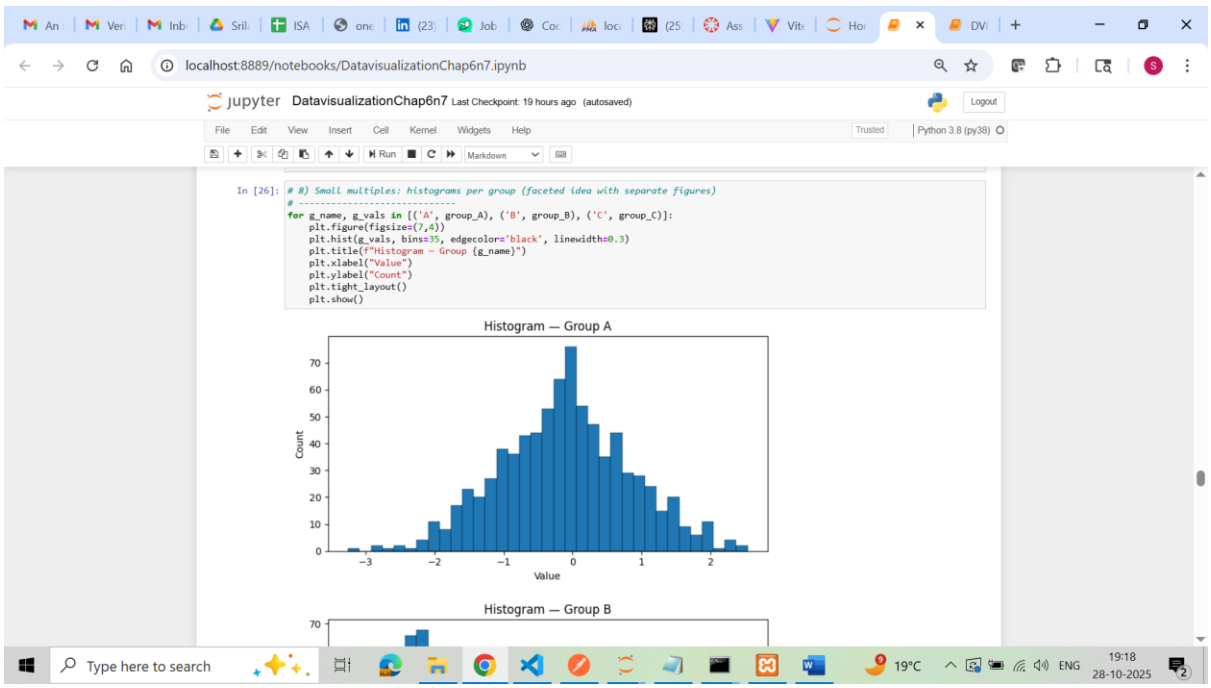


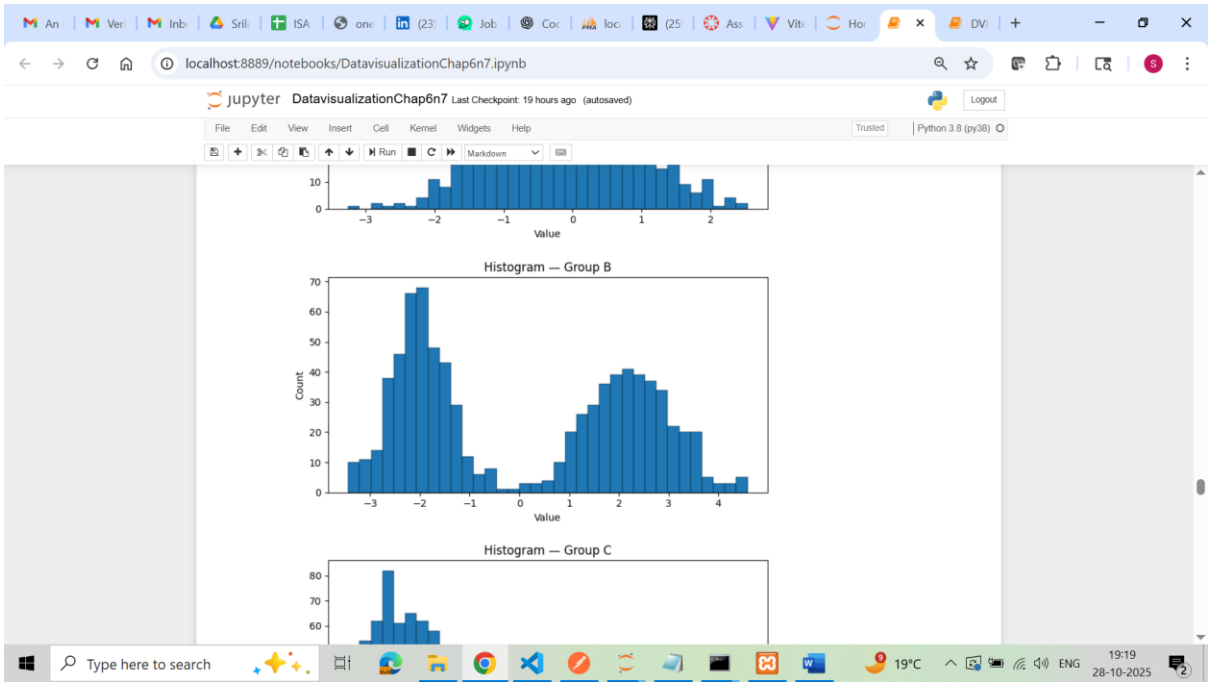
Type here to search 19°C ENG 19:18 28-10-2025

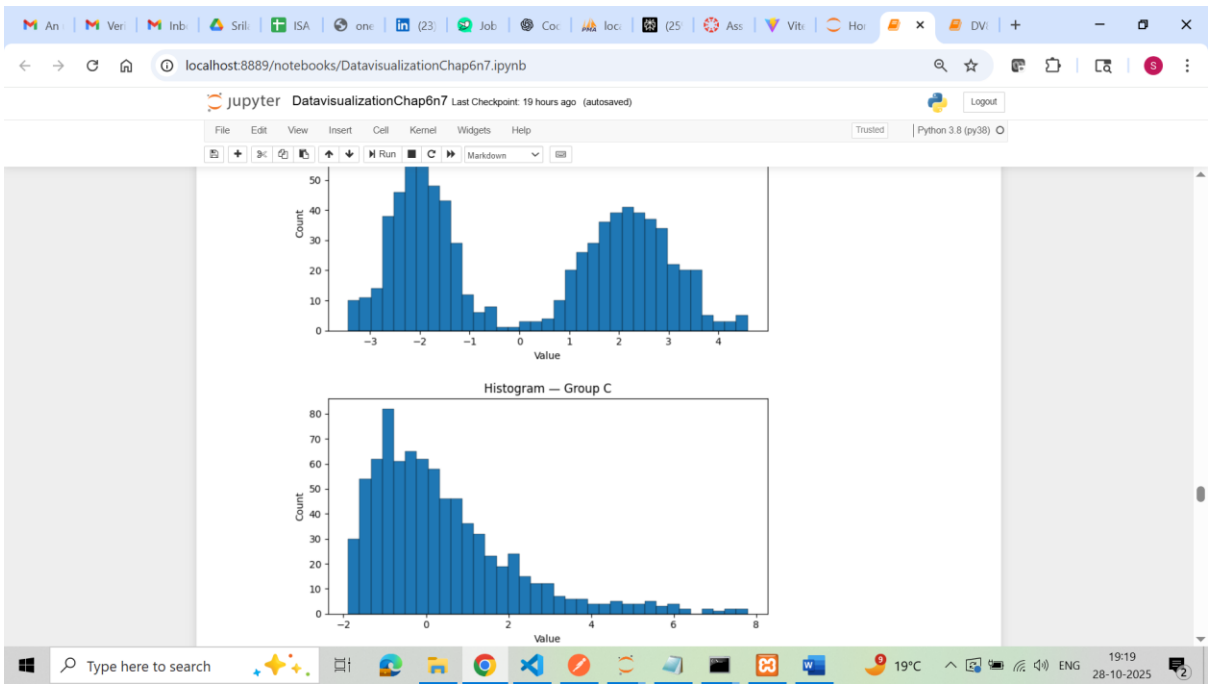

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
plt.figure(figsize=(8,5))
plt.plot(x_grid, density_A)
plt.title("Kernel Density Estimate (KDE) — Group A (bandwidth=0.5)")
plt.xlabel("Value")
plt.ylabel("Estimated density")
plt.tight_layout()
plt.show()
Kernel Density Estimate (KDE) — Group A (bandwidth=0.5)
Estimated density
Value
Type here to search
19°C
ENG
19:18
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
# 5) Empirical CDF (ECDF) — all groups overlaid
# ------------------------------
def ecdf(samples):
x = np.sort(samples)
y = np.arange(1, len(x)+1) / len(x)
return x, y
xA, yA = ecdf(group_A)
xB, yB = ecdf(group_B)
xC, yC = ecdf(group_C)
plt.figure(figsize=(8,5))
plt.step(xA, yA, where='post', label='A')
plt.step(xB, yB, where='post', label='B')
plt.step(xC, yC, where='post', label='C')
plt.title("Empirical CDF — Groups A, B, C")
plt.xlabel("Value")
plt.ylabel("Cumulative proportion")
plt.legend()
plt.tight_layout()
plt.show()
Empirical CDF — Groups A, B, C
A
B
C
Cumulative proportion
Type here to search
19°C
ENG
19:18
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
plt.figure(figsize=(8,5))
plt.step(xA, yA, where='post', label='A')
plt.step(xB, yB, where='post', label='B')
plt.step(xC, yC, where='post', label='C')
plt.title("Empirical CDF — Groups A, B, C")
plt.xlabel("Value")
plt.ylabel("Cumulative proportion")
plt.legend()
plt.tight_layout()
plt.show()
Empirical CDF — Groups A, B, C
A
B
C
Cumulative proportion
Value
Type here to search
19°C
ENG
19:18
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
In [26]:
# 8) Small multiples: histograms per group (faceted idea with separate figures)
# ------------------------------
for g_name, g_vals in [('A', group_A), ('B', group_B), ('C', group_C)]:
plt.figure(figsize=(7,4))
plt.hist(g_vals, bins=35, edgecolor='black', linewidth=0.3)
plt.title(f"Histogram — Group {g_name}")
plt.xlabel("Value")
plt.ylabel("Count")
plt.tight_layout()
plt.show()
Histogram — Group A
Count
Value
Histogram — Group B
Type here to search
19°C
ENG
19:18
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
Value
Histogram — Group B
Count
Value
Histogram — Group C
Type here to search
19°C
ENG
19:19
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
Count
Value
Histogram — Group C
Count
Value
Type here to search
19°C
ENG
19:19
28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

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

rng = np.random.default_rng(42)

# ------------------------------
# Two-group dataset: session duration (minutes)
# ------------------------------
# New users: generally shorter sessions with a long right tail
n_new = 600
session_new = np.concatenate([
    rng.normal(8, 3, size=500),     # main mass around 8 min
    rng.exponential(6, size=100)    # right tail
])
session_new = session_new[session_new > 0]

# Returning users: longer/more variable sessions
n_ret = 1000
session_ret = np.concatenate([
    rng.normal(16, 5, size=800),
    rng.normal(28, 6, size=200)     # a second hump for power users
])
session_ret = session_ret[session_ret > 0]

# Common bins so comparisons are fair
bins = np.linspace(0, max(session_new.max(), session_ret.max()), 30)

# ------------------------------
# A) BAD: Stacked histogram (easily misread)
# ------------------------------
plt.figure(figsize=(9,5))
plt.hist([session_new, session_ret], bins=bins, stacked=True, label=['New','Returning'])
plt.xlabel("Session duration (min)")
plt.ylabel("Count")
plt.title("BAD: Stacked histogram of session durations (New vs Returning)")
plt.legend()
plt.tight_layout()
plt.show()
```

Type here to search

19°C ENG 19:19 28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

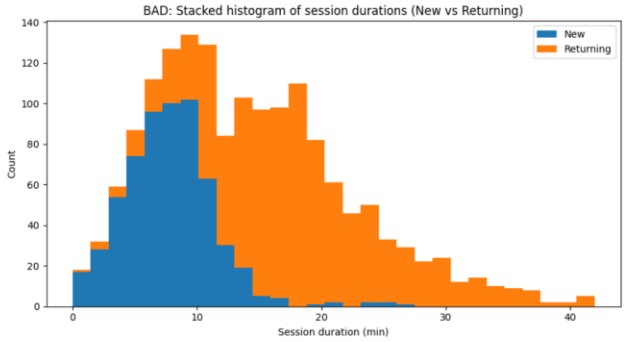
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

```
plt.figure(figsize=(9,5))
plt.hist([session_new, session_ret], bins=bins, stacked=True, label=['New','Returning'])
plt.xlabel("Session duration (min)")
plt.ylabel("Count")
plt.title("BAD: Stacked histogram of session durations (New vs Returning)")
plt.legend()
plt.tight_layout()
plt.show()
```

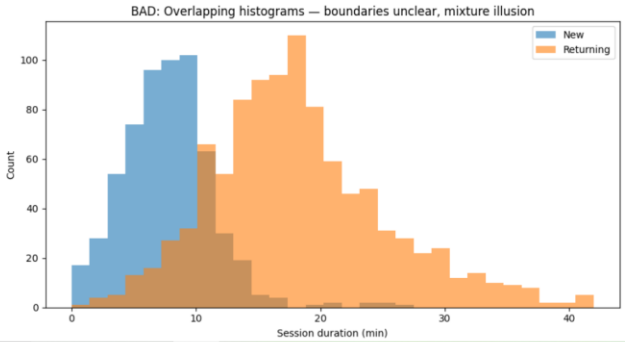



```
In [28]: # B) BAD: Overlapping histograms (alpha) — looks like 3 groups
```

Type here to search

Tem... ENG 19:20 28-10-2025

```
plt.figure(figsize=(9,5))
plt.hist(session_new, bins=bins, alpha=0.6, label='New')
plt.hist(session_ret, bins=bins, alpha=0.6, label='Returning')
plt.xlabel("Session duration (min)")
plt.ylabel("Count")
plt.title("BAD: Overlapping histograms — boundaries unclear, mixture illusion")
plt.legend()
plt.tight_layout()
plt.show()
```



In [30]:
```
# C) Better: KDEs scaled by group size (area equals group count)
# ----------------------------
def kde_gaussian(x, samples, bandwidth):
    x = x[:, None]
    z = (x - samples[None, :]) / bandwidth
    return np.exp(-0.5 * z**2).mean(axis=1) / (np.sqrt(2*np.pi) * bandwidth)

# Grid for KDEs
x_grid = np.linspace(0, bins[-1], 600)

# Choose bandwidth heuristically (Silverman-ish, simplified)
def silverman_bw(samples):
    s = np.std(samples, ddof=1)
    n = len(samples)
    return 1.06 * s * n ** (-1/5)

bw_new = silverman_bw(session_new)
bw_ret = silverman_bw(session_ret)

dens_new = kde_gaussian(x_grid, session_new, bw_new)
dens_ret = kde_gaussian(x_grid, session_ret, bw_ret)
# Scale KDE so integral approximates group size. (Riemann sum with grid step)
dx = x_grid[1] - x_grid[0]
scaled_new = dens_new * (len(session_new) / (dens_new.sum() * dx))
scaled_ret = dens_ret * (len(session_ret) / (dens_ret.sum() * dx))

plt.figure(figsize=(9,5))
plt.plot(x_grid, scaled_new, label=f"New (n={len(session_new)})")
plt.plot(x_grid, scaled_ret, label=f"Returning (n={len(session_ret)})")
plt.xlabel("Session duration (min)")
plt.ylabel("Scaled density (area = group size)")
plt.title("KDEs scaled by group size — clearer two-group comparison")
plt.legend()
plt.tight_layout()
plt.show()
```

KDEs scaled by group size — clearer two-group comparison

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

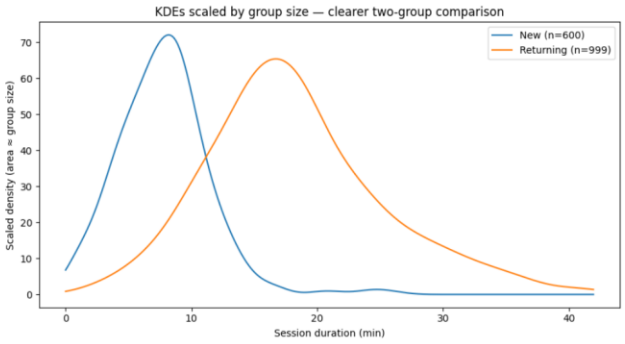
Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

```
plt.plot(x_grid, scaled_new, label=f"New (n={len(session_new)})")
plt.plot(x_grid, scaled_ret, label=f"Returning (n={len(session_ret)})")
plt.xlabel("Session duration (min)")
plt.ylabel("Scaled density (area = group size)")
plt.title("KDEs scaled by group size — clearer two-group comparison")
plt.legend()
plt.tight_layout()
plt.show()
```



Type here to search

Tem...

ENG 19:22 28-10-2025

localhost:8889/notebooks/DatavisualizationChap6n7.ipynb

jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

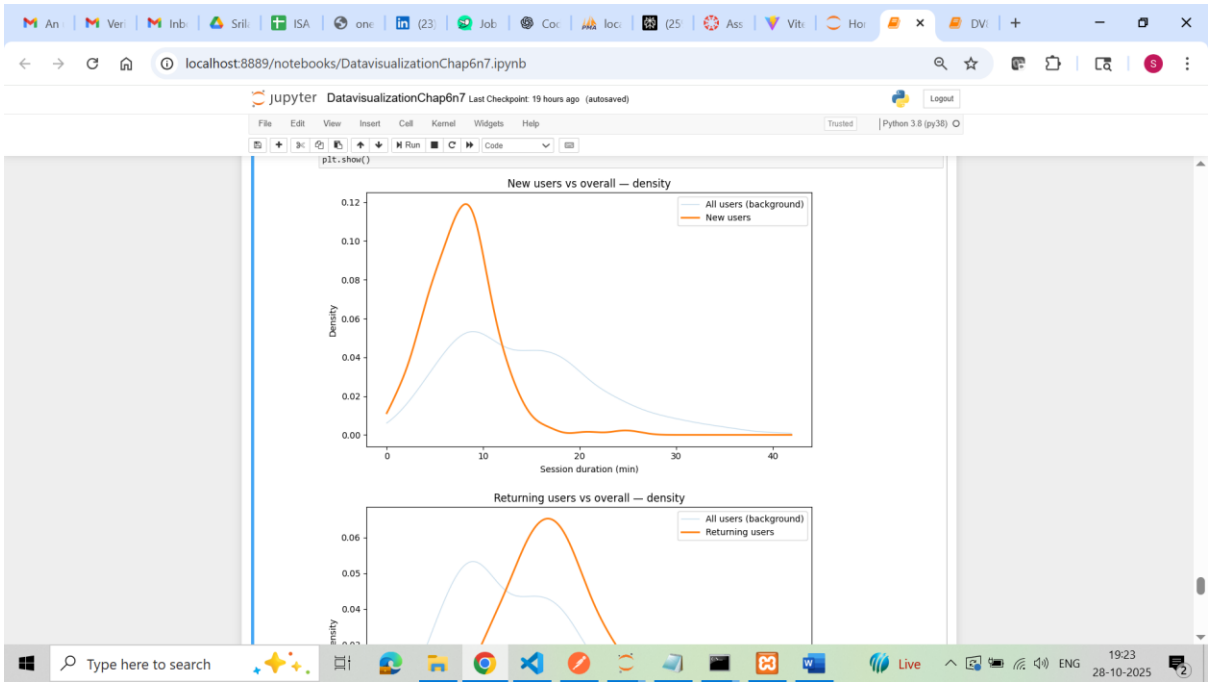
Run | Code

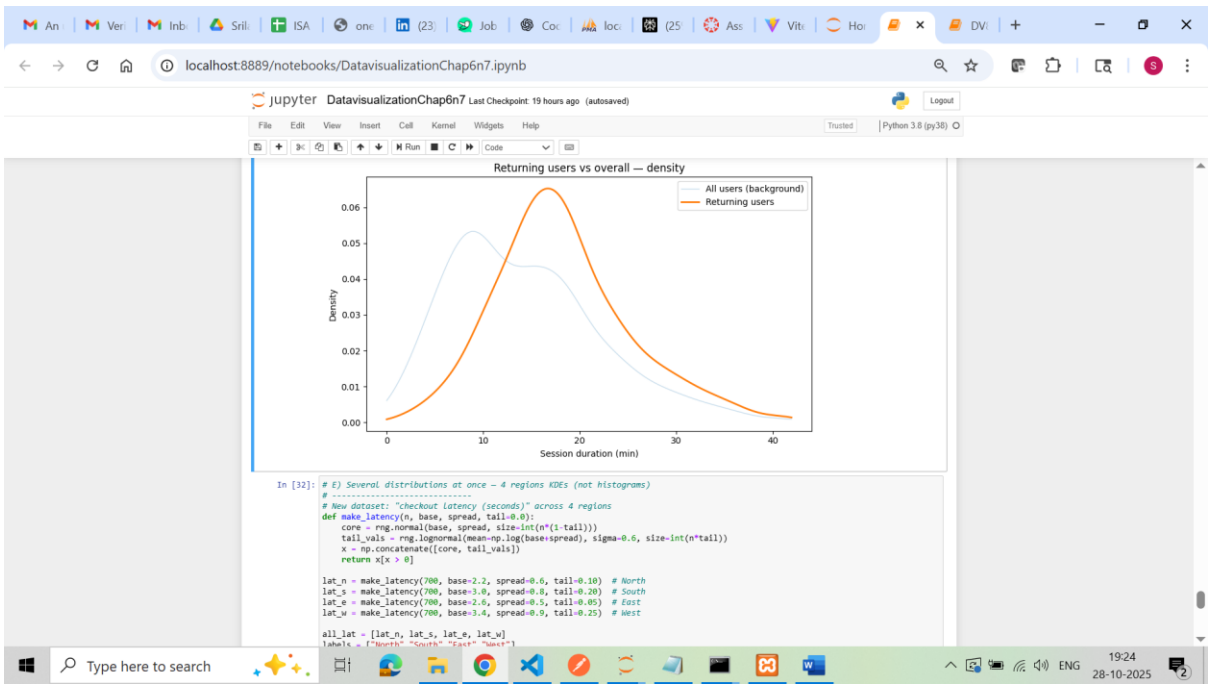
```
# ----------------------------
# D) Separate panels idea (two separate figures, no subplots):
#    Show each group's density with overall density faintly in background
# ----------------------------
# Overall density
bw_all = silverman_bw(np.concatenate([session_new, session_ret]))
dens_all = kde_gaussian(x_grid, np.concatenate([session_new, session_ret]), bw_all)

# New users panel
plt.figure(figsize=(8,5))
plt.plot(x_grid, dens_all, linewidth=1, alpha=0.25, label="All users (background)")
plt.plot(x_grid, dens_new, linewidth=2, label="New users")
plt.xlabel("Session duration (min)")
plt.ylabel("Density")
plt.title("New users vs overall — density")
plt.legend()
plt.tight_layout()
plt.show()

# Returning users panel
plt.figure(figsize=(8,5))
plt.plot(x_grid, dens_all, linewidth=1, alpha=0.25, label="All users (background)")
plt.plot(x_grid, dens_ret, linewidth=2, label="Returning users")
plt.xlabel("Session duration (min)")
plt.ylabel("Density")
plt.title("Returning users vs overall — density")
plt.legend()
plt.tight_layout()
plt.show()
```

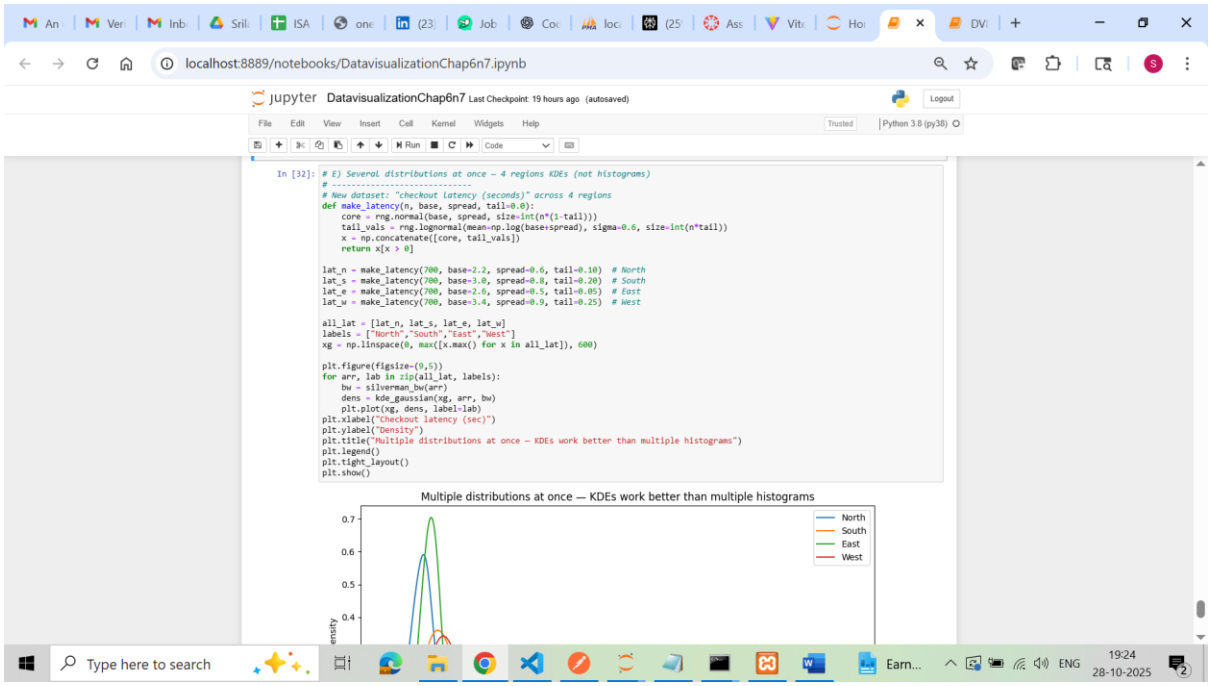
Type here to search

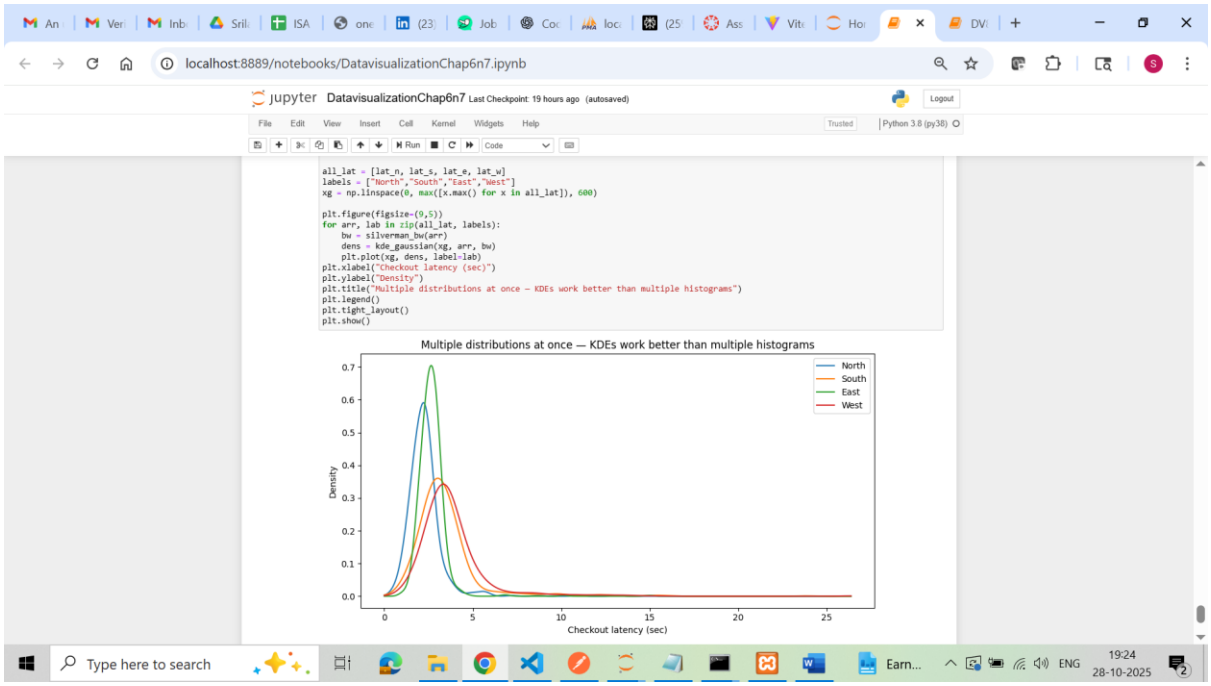
Live

ENG 19:22 28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
plt.show()
New users vs overall — density
All users (background)
New users
Density
Session duration (min)
Returning users vs overall — density
All users (background)
Returning users
Type here to search
ENG
19:23
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
Returning users vs overall — density
All users (background)
Returning users
Density
Session duration (min)
In [32]:
# E) Several distributions at once — 4 regions KDEs (not histograms)
# ------------------------------
# New dataset: "checkout latency (seconds)" across 4 regions
def make_latency(n, base, spread, tail=0.0):
core = rng.normal(base, spread, size=int(n*(1-tail)))
tail_vals = rng.lognormal(mean=np.log(base+spread), sigma=0.6, size=int(n*tail))
x = np.concatenate([core, tail_vals])
return x[x > 0]
lat_n = make_latency(700, base=2.2, spread=0.6, tail=0.10) # North
lat_s = make_latency(700, base=3.0, spread=0.8, tail=0.20) # South
lat_e = make_latency(700, base=2.6, spread=0.5, tail=0.05) # East
lat_w = make_latency(700, base=3.4, spread=0.9, tail=0.25) # West
all_lat = [lat_n, lat_s, lat_e, lat_w]
Type here to search
ENG
19:24
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
In [32]:
# E) Several distributions at once — 4 regions KDEs (not histograms)
# ----------------------------
# New dataset: "checkout latency (seconds)" across 4 regions
def make_latency(n, base, spread, tail=0.0):
core = rng.normal(base, spread, size=int(n*(1-tail)))
tail_vals = rng.lognormal(mean=np.log(base+spread), sigma=0.6, size=int(n*tail))
x = np.concatenate([core, tail_vals])
return x[x > 0]
lat_n = make_latency(700, base=2.2, spread=0.6, tail=0.10) # North
lat_s = make_latency(700, base=3.0, spread=0.8, tail=0.20) # South
lat_e = make_latency(700, base=2.6, spread=0.5, tail=0.05) # East
lat_w = make_latency(700, base=3.4, spread=0.9, tail=0.25) # West
all_lat = [lat_n, lat_s, lat_e, lat_w]
labels = ["North","South","East","West"]
xg = np.linspace(0, max([x.max() for x in all_lat]), 600)
plt.figure(figsize=(9,5))
for arr, lab in zip(all_lat, labels):
bw = silverman_bw(arr)
dens = kde_gaussian(xg, arr, bw)
plt.plot(xg, dens, label=lab)
plt.xlabel("Checkout latency (sec)")
plt.ylabel("Density")
plt.title("Multiple distributions at once — KDEs work better than multiple histograms")
plt.legend()
plt.tight_layout()
plt.show()
Multiple distributions at once — KDEs work better than multiple histograms
North
South
East
West
0.7
0.6
0.5
0.4
Type here to search
Earn...
ENG
19:24
28-10-2025


localhost:8889/notebooks/DatavisualizationChap6n7.ipynb
jupyter DatavisualizationChap6n7 Last Checkpoint: 19 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
all_lat = [lat_n, lat_s, lat_e, lat_w]
labels = ["North","South","East","West"]
xg = np.linspace(0, max([x.max() for x in all_lat]), 600)
plt.figure(figsize=(9,5))
for arr, lab in zip(all_lat, labels):
bw = silverman_bw(arr)
dens = kde_gaussian(xg, arr, bw)
plt.plot(xg, dens, label=lab)
plt.xlabel("Checkout latency (sec)")
plt.ylabel("Density")
plt.title("Multiple distributions at once — KDEs work better than multiple histograms")
plt.legend()
plt.tight_layout()
plt.show()
Multiple distributions at once — KDEs work better than multiple histograms
North
South
East
West
Density
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
5
10
15
20
25
Checkout latency (sec)
Type here to search
Earn...
ENG
19:24
28-10-2025

CHAPTER 8

localhost:8889/notebooks/DV8.ipynb

jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

Run | Markdown

CHAPTER 8

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

rng = np.random.default_rng(42)

# ---------------------------------------------------------
# Dataset setup
# ---------------------------------------------------------
scores = rng.normal(75, 10, 50)          # Exam scores ~ Normal
scores = np.clip(scores, 0, 100)

population = np.exp(rng.normal(10, 1.2, 3000))   # Right-skewed (Log-normal)
word_counts = np.random.zipf(a=2.0, size=5000)   # Very heavy tail

# ---------------------------------------------------------
# 8.1 Empirical Cumulative Distribution Function (ECDF)
# ---------------------------------------------------------
def ecdf(data):
    x = np.sort(data)
    y = np.arange(1, len(x)+1) / len(x)
    return x, y

x_scores, y_scores = ecdf(scores)

plt.figure(figsize=(7,5))
plt.step(x_scores, y_scores, where='post')
plt.xlabel("Score")
plt.ylabel("Cumulative proportion")
plt.title("ECDF of Student Exam Scores")
plt.tight_layout()
plt.show()

# Descending ECDF
plt.figure(figsize=(7,5))
plt.step(x_scores, 1 - y_scores, where='post')
plt.xlabel("Score")
plt.ylabel("Fraction ≥ Score")
plt.title("Descending ECDF (Complementary CDF)")
plt.tight_layout()
plt.show()
```

Type here to search

19°C ENG 19:29 28-10-2025

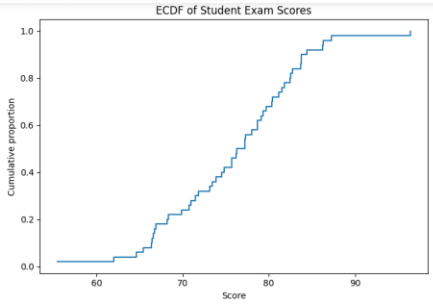
localhost:8889/notebooks/DV8.ipynb

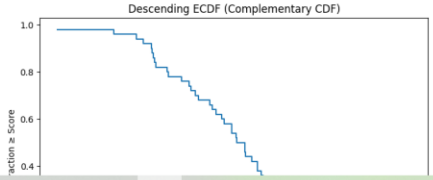
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

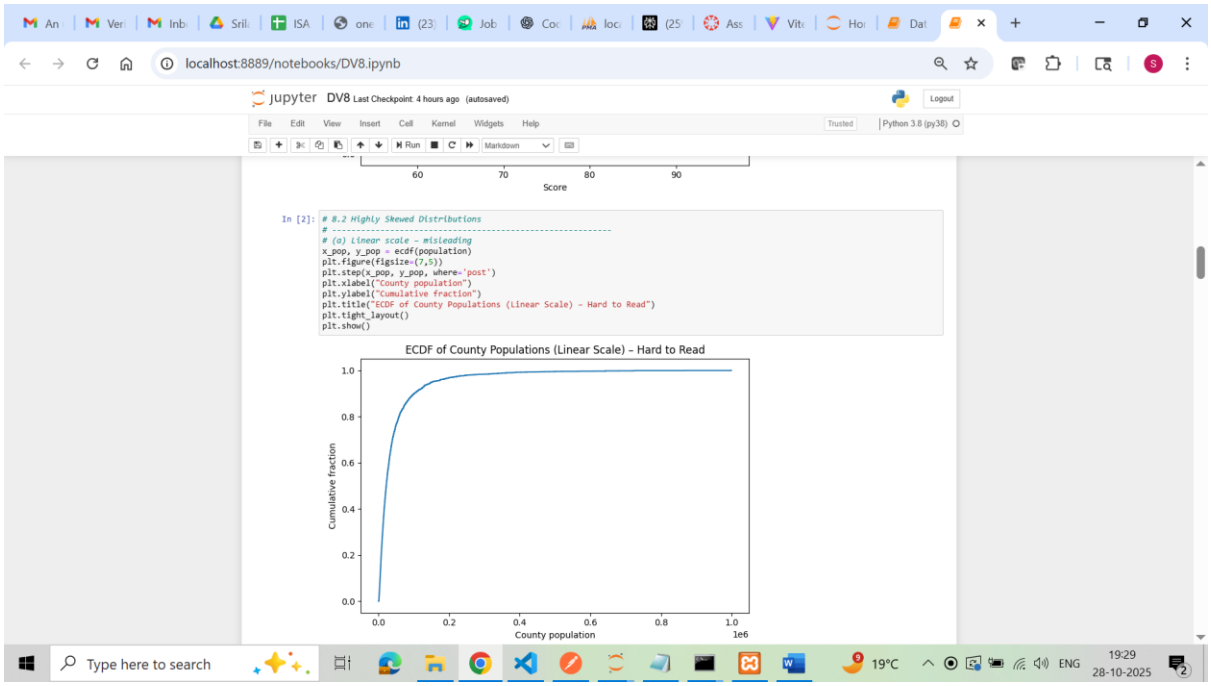
Run | Markdown

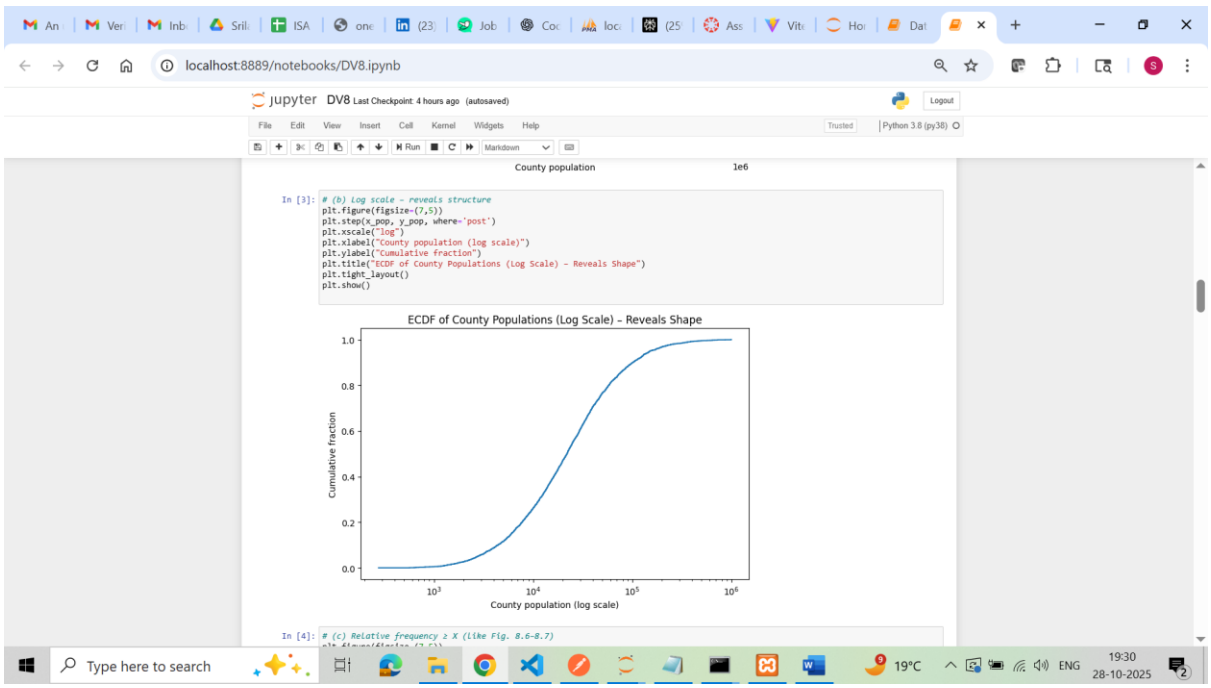


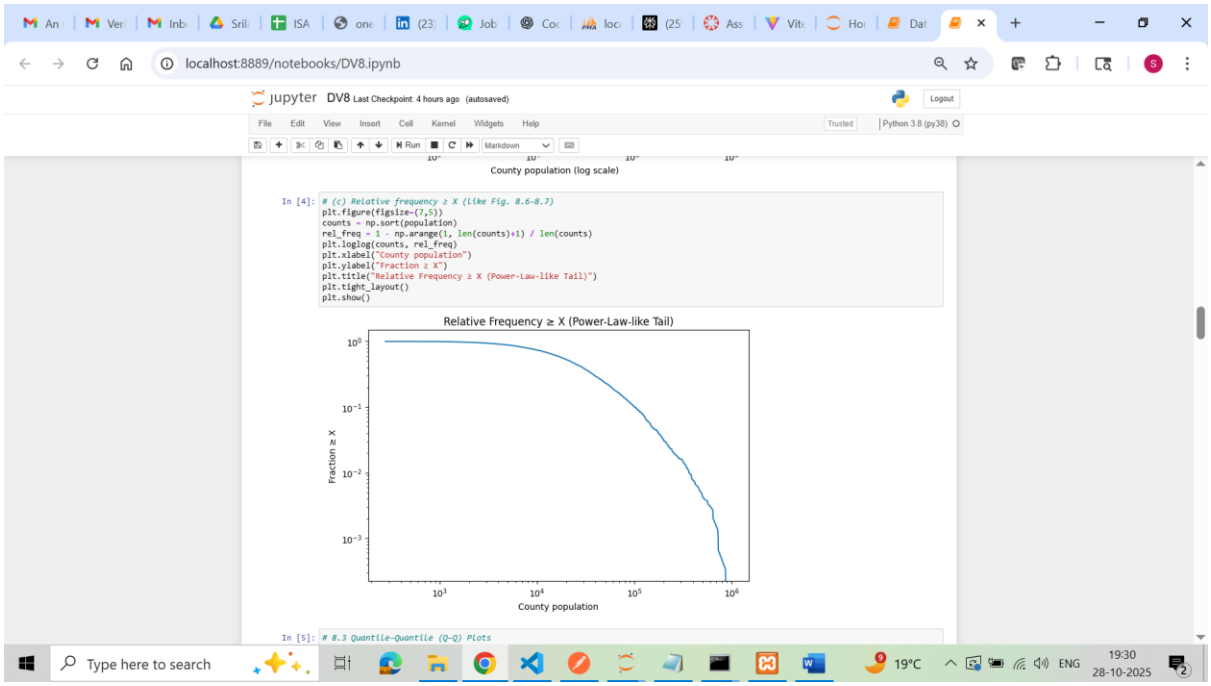


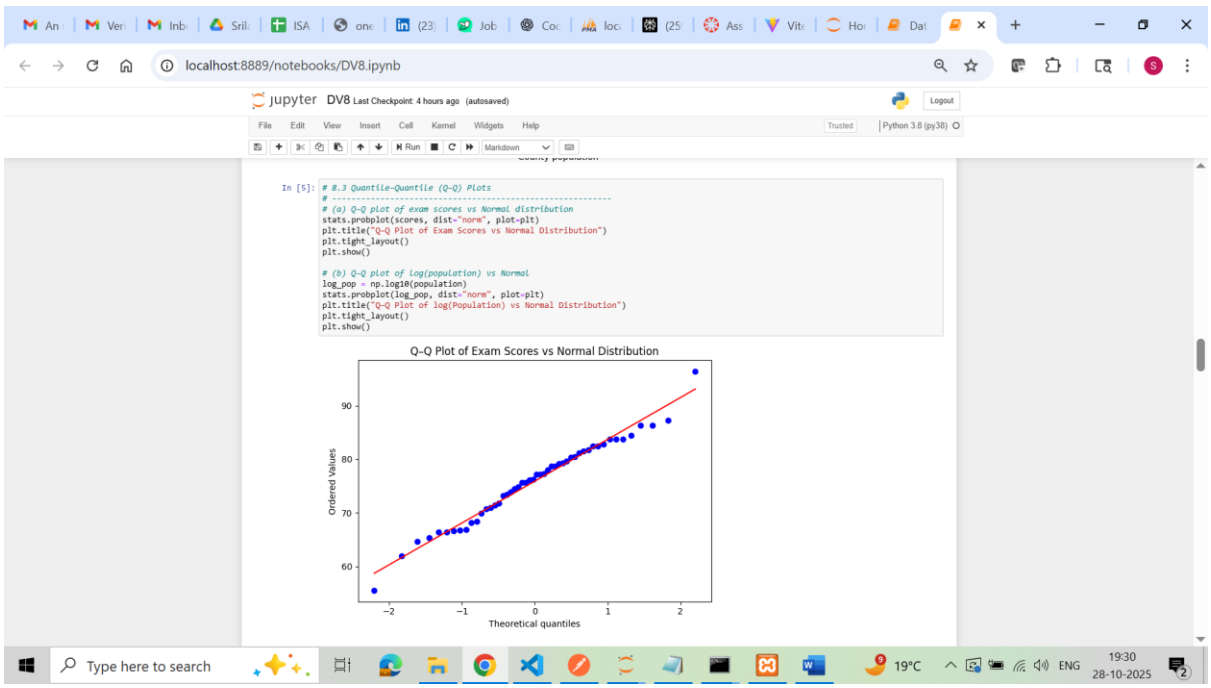
Type here to search

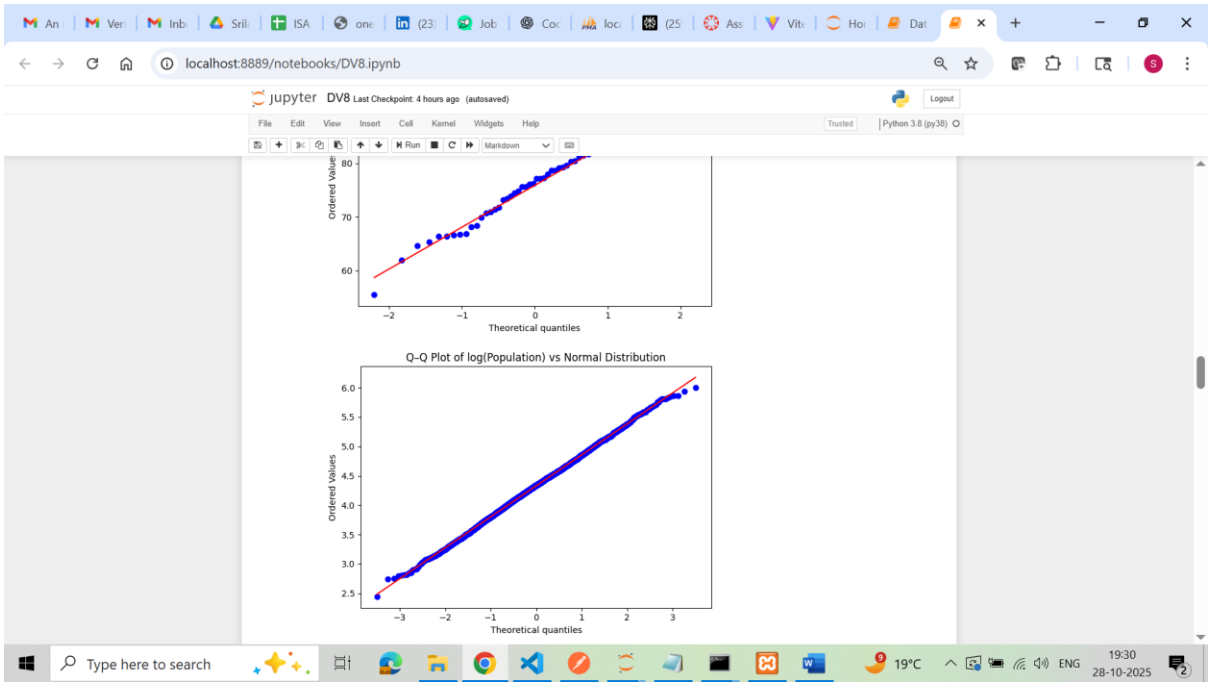
19°C ENG 19:29 28-10-2025


localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
60 70 80 90
Score
In [2]:
8.2 Highly Skewed Distributions
--
(a) Linear scale – misleading
x_pop, y_pop = ecdf(population)
plt.figure(figsize=(7,5))
plt.step(x_pop, y_pop, where='post')
plt.xlabel("County population")
plt.ylabel("Cumulative fraction")
plt.title("ECDF of County Populations (Linear Scale) – Hard to Read")
plt.tight_layout()
plt.show()
ECDF of County Populations (Linear Scale) – Hard to Read
Cumulative fraction
0.0 0.2 0.4 0.6 0.8 1.0
0.0 0.2 0.4 0.6 0.8 1.0
County population
1e6
Type here to search
19°C
ENG
19:29
28-10-2025

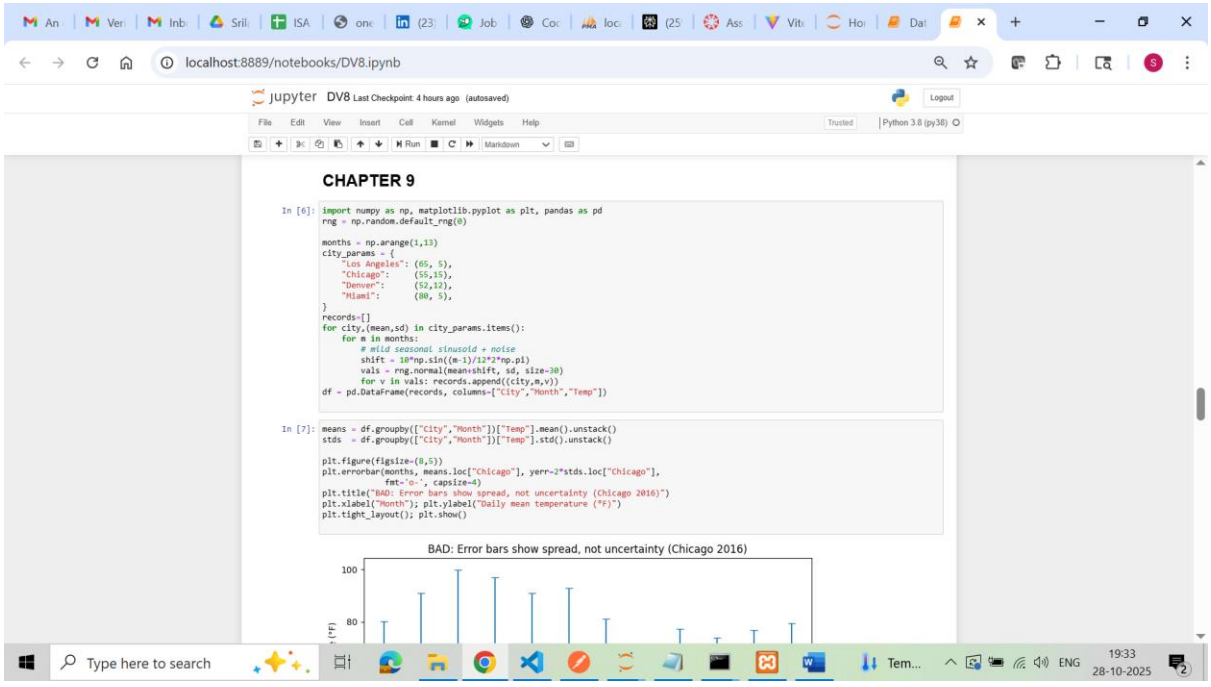

localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
County population
1e6
In [3]:
# (b) Log scale – reveals structure
plt.figure(figsize=(7,5))
plt.step(x_pop, y_pop, where='post')
plt.xscale("log")
plt.xlabel("County population (log scale)")
plt.ylabel("Cumulative fraction")
plt.title("ECDF of County Populations (Log Scale) – Reveals Shape")
plt.tight_layout()
plt.show()
ECDF of County Populations (Log Scale) – Reveals Shape
Cumulative fraction
0.0 0.2 0.4 0.6 0.8 1.0
10^3 10^4 10^5 10^6
County population (log scale)
In [4]:
(c) Relative frequency ≥ X (like Fig. 8.6-8.7)
Type here to search
19°C
ENG
19:30
28-10-2025

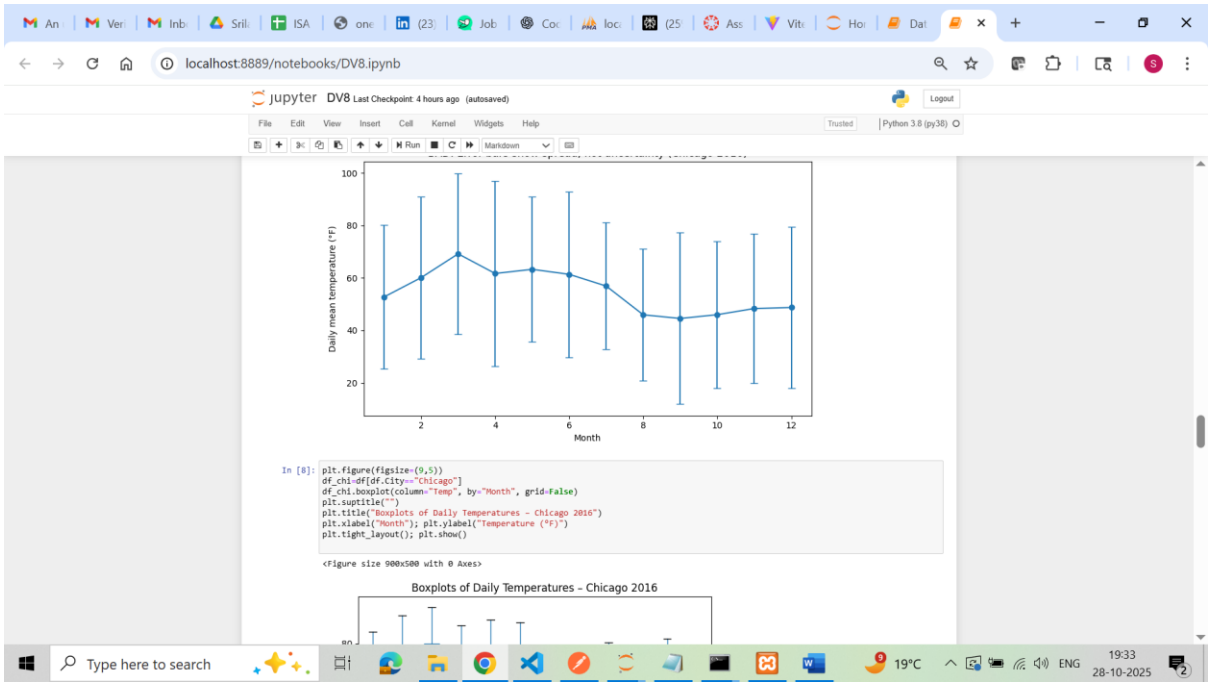

localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Logout
Run
Markdown
County population (log scale)
In [4]:
# (c) Relative frequency ≥ X (like Fig. 8.6-8.7)
plt.figure(figsize=(7,5))
counts = np.sort(population)
rel_freq = 1 - np.arange(1, len(counts)+1) / len(counts)
plt.loglog(counts, rel_freq)
plt.xlabel("County population")
plt.ylabel("Fraction ≥ X")
plt.title("Relative Frequency ≥ X (Power-Law-like Tail)")
plt.tight_layout()
plt.show()
Relative Frequency ≥ X (Power-Law-like Tail)
Fraction ≥ X
County population
In [5]:
8.3 Quantile-Quantile (Q-Q) Plots
Type here to search
19°C
ENG
19:30
28-10-2025

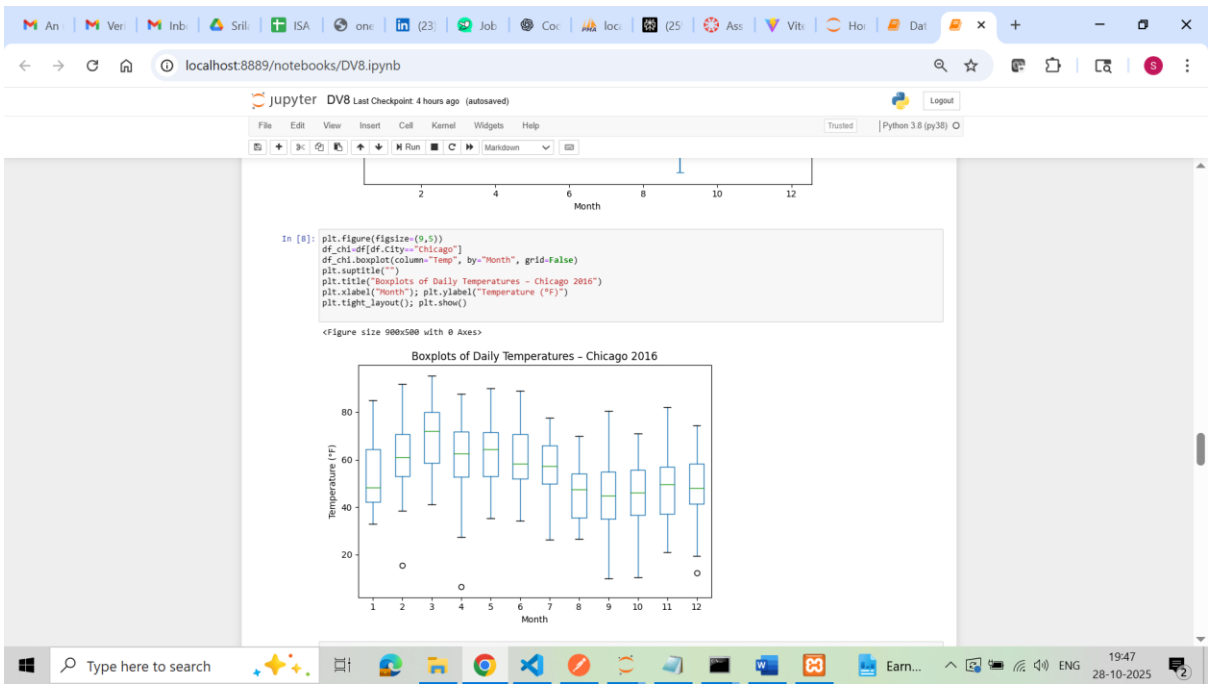

localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Logout
Run
Markdown
In [5]:
# 8.3 Quantile-Quantile (Q-Q) Plots
# --
# (a) Q-Q plot of exam scores vs Normal distribution
stats.probplot(scores, dist="norm", plot=plt)
plt.title("Q-Q Plot of Exam Scores vs Normal Distribution")
plt.tight_layout()
plt.show()
# (b) Q-Q plot of log(population) vs Normal
log_pop = np.log10(population)
stats.probplot(log_pop, dist="norm", plot=plt)
plt.title("Q-Q Plot of log(Population) vs Normal Distribution")
plt.tight_layout()
plt.show()
Q-Q Plot of Exam Scores vs Normal Distribution
Ordered Values
Theoretical quantiles
Type here to search
19°C
ENG
19:30
28-10-2025

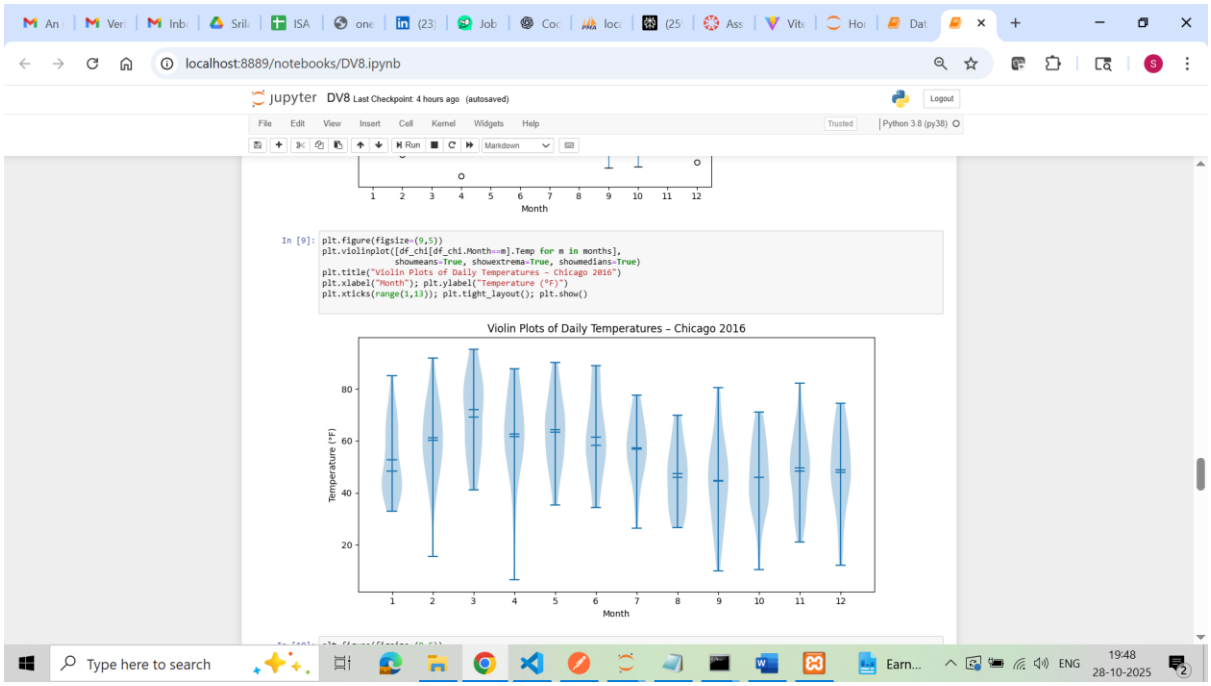

localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
Ordered Values
Theoretical quantiles
Q-Q Plot of log(Population) vs Normal Distribution
Ordered Values
Theoretical quantiles
Type here to search
19°C
ENG
19:30
28-10-2025

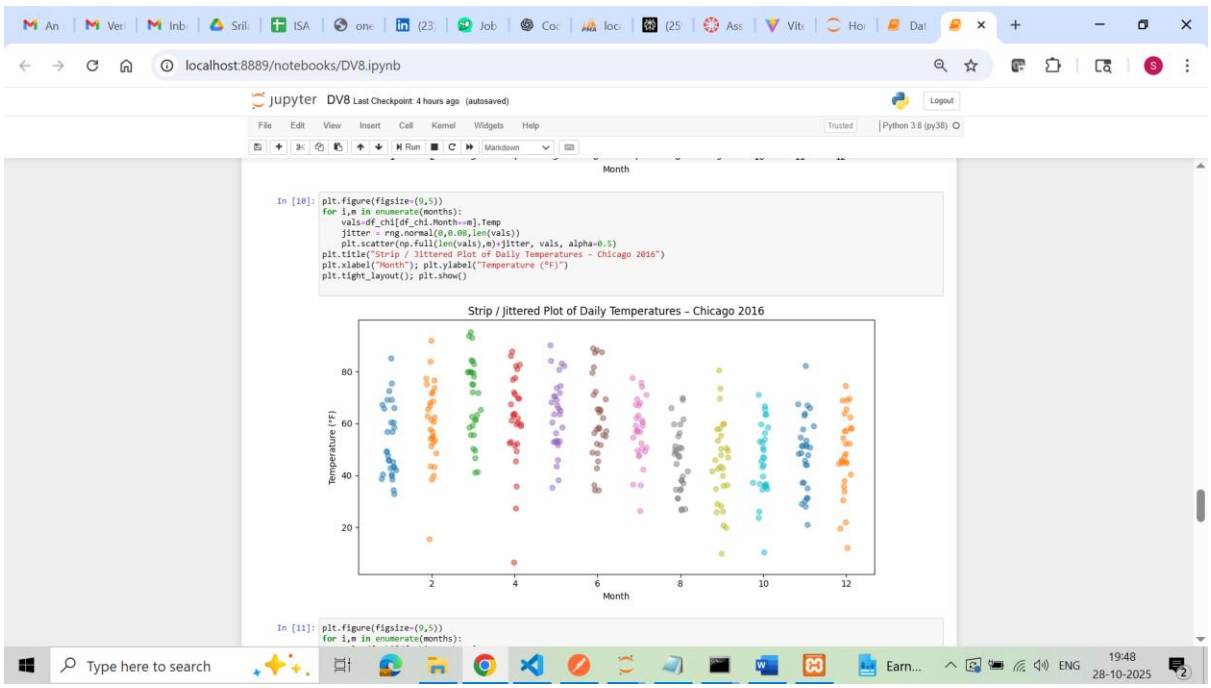
CHAPTER 9


localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
CHAPTER 9
In [6]:
import numpy as np, matplotlib.pyplot as plt, pandas as pd
rng = np.random.default_rng(0)
months = np.arange(1,13)
city_params = {
"Los Angeles": (65, 5),
"Chicago": (55,15),
"Denver": (52,12),
"Miami": (80, 5),
}
records=[]
for city,(mean,sd) in city_params.items():
for m in months:
# mild seasonal sinusoid + noise
shift = 10*np.sin((m-1)/12*2*np.pi)
vals = rng.normal(mean+shift, sd, size=30)
for v in vals: records.append((city,m,v))
df = pd.DataFrame(records, columns=["City","Month","Temp"])
In [7]:
means = df.groupby(["City","Month"])["Temp"].mean().unstack()
stds = df.groupby(["City","Month"])["Temp"].std().unstack()
plt.figure(figsize=(8,5))
plt.errorbar(months, means.loc["Chicago"], yerr=2*stds.loc["Chicago"],
fmt='o-', capsize=4)
plt.title("BAD: Error bars show spread, not uncertainty (Chicago 2016)")
plt.xlabel("Month"); plt.ylabel("Daily mean temperature (°F)")
plt.tight_layout(); plt.show()
BAD: Error bars show spread, not uncertainty (Chicago 2016)
Type here to search
ENG
19:33
28-10-2025


localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
Daily mean temperature (°F)
Month
In [8]:
plt.figure(figsize=(9,5))
df_chi=df[df.City=="Chicago"]
df_chi.boxplot(column="Temp", by="Month", grid=False)
plt.suptitle("")
plt.title("Boxplots of Daily Temperatures - Chicago 2016")
plt.xlabel("Month"); plt.ylabel("Temperature (°F)")
plt.tight_layout(); plt.show()
<Figure size 900x500 with 0 Axes>
Boxplots of Daily Temperatures - Chicago 2016
Type here to search
19°C
ENG
19:33
28-10-2025


localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Run
Markdown
Month
In [8]:
plt.figure(figsize=(9,5))
df_chi=df[df.City=="Chicago"]
df_chi.boxplot(column="Temp", by="Month", grid=False)
plt.suptitle("")
plt.title("Boxplots of Daily Temperatures - Chicago 2016")
plt.xlabel("Month"); plt.ylabel("Temperature (°F)")
plt.tight_layout(); plt.show()
<Figure size 900x500 with 0 Axes>
Boxplots of Daily Temperatures - Chicago 2016
Temperature (°F)
Month
Type here to search
Earn...
ENG
19:47
28-10-2025


localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
Month
In [9]:
plt.figure(figsize=(9,5))
plt.violinplot([df_chi[df_chi.Month==m].Temp for m in months],
showmeans=True, showextrema=True, showmedians=True)
plt.title("Violin Plots of Daily Temperatures – Chicago 2016")
plt.xlabel("Month"); plt.ylabel("Temperature (°F)")
plt.xticks(range(1,13)); plt.tight_layout(); plt.show()
Violin Plots of Daily Temperatures – Chicago 2016
Temperature (°F)
Month
Type here to search
ENG
19:48
28-10-2025


localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
Month
In [10]:
plt.figure(figsize=(9,5))
for i,m in enumerate(months):
vals=df_chi[df_chi.Month==m].Temp
jitter = rng.normal(0,0.08,len(vals))
plt.scatter(np.full(len(vals),m)+jitter, vals, alpha=0.5)
plt.title("Strip / Jittered Plot of Daily Temperatures – Chicago 2016")
plt.xlabel("Month"); plt.ylabel("Temperature (°F)")
plt.tight_layout(); plt.show()
Strip / Jittered Plot of Daily Temperatures – Chicago 2016
Temperature (°F)
Month
In [11]:
Type here to search
ENG
19:48
28-10-2025

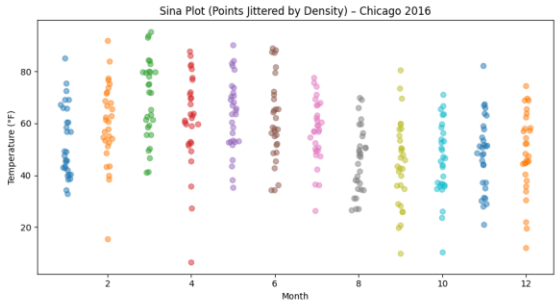
localhost:8889/notebooks/DV8.ipynb

jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3.8 (py38)

```
In [11]: plt.figure(figsize=(9,5))
for i,m in enumerate(months):
    vals=df_chi[df_chi.Month==m].Temp
    # horizontal jitter proportional to local density
    density,grid=np.histogram(vals,bins=20)
    probs=density[np.digitize(vals,grid[:-1])-1]
    xjit=rng.normal(0,0.08*probs/probs.max(),len(vals))
    plt.scatter(np.full(len(vals),m)+xjit, vals, alpha=0.5)
plt.title("Sina Plot (Points Jittered by Density) – Chicago 2016")
plt.xlabel("Month"); plt.ylabel("Temperature (°F)")
plt.tight_layout(); plt.show()
```




```
In [12]: import matplotlib.cm as cm
```

Type here to search

Earn... ENG 19:48 28-10-2025

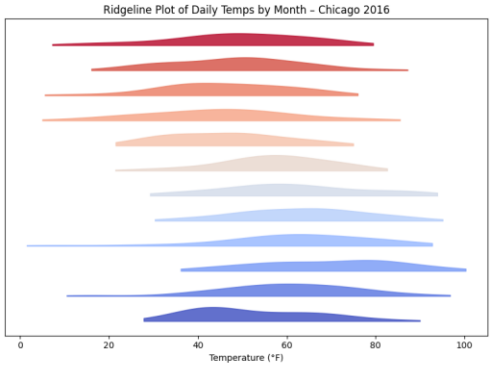
localhost:8889/notebooks/DV8.ipynb

jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

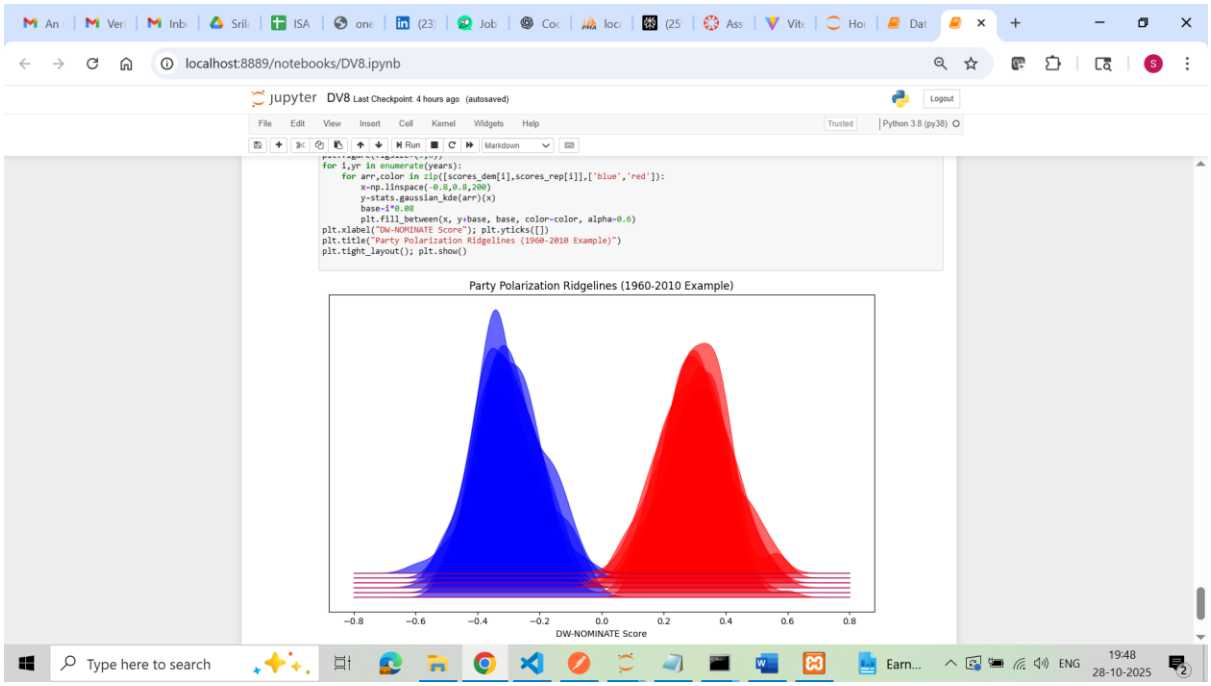
Trusted | Python 3.8 (py38)

```
colors=cm.coolwarm(np.linspace(0,1,12))
for i,m in enumerate(months):
    vals=df_chi[df_chi.Month==m].Temp
    kde_x=np.linspace(min(vals)-5,max(vals)+5,200)
    kde_y=stats.gaussian_kde(vals)(kde_x)
    plt.fill_between(kde_x, kde_y+i*0.05, i*0.05, color=colors[i], alpha=0.8)
plt.yticks([]); plt.xlabel("Temperature (°F)")
plt.title("Ridgeline Plot of Daily Temps by Month – Chicago 2016")
plt.tight_layout(); plt.show()
```



Type here to search

Earn... ENG 19:48 28-10-2025


localhost:8889/notebooks/DV8.ipynb
jupyter DV8 Last Checkpoint: 4 hours ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3.8 (py38)
Logout
Run
Markdown
for i,yr in enumerate(years):
for arr,color in zip([scores_dem[i],scores_rep[i]],['blue','red']):
x=np.linspace(-0.8,0.8,200)
y=stats.gaussian_kde(arr)(x)
base=i*0.08
plt.fill_between(x, y+base, base, color=color, alpha=0.6)
plt.xlabel("DW-NOMINATE Score"); plt.yticks([])
plt.title("Party Polarization Ridgelines (1960-2010 Example)")
plt.tight_layout(); plt.show()
Party Polarization Ridgelines (1960-2010 Example)
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
DW-NOMINATE Score
Type here to search
Earn...
ENG
19:48
28-10-2025