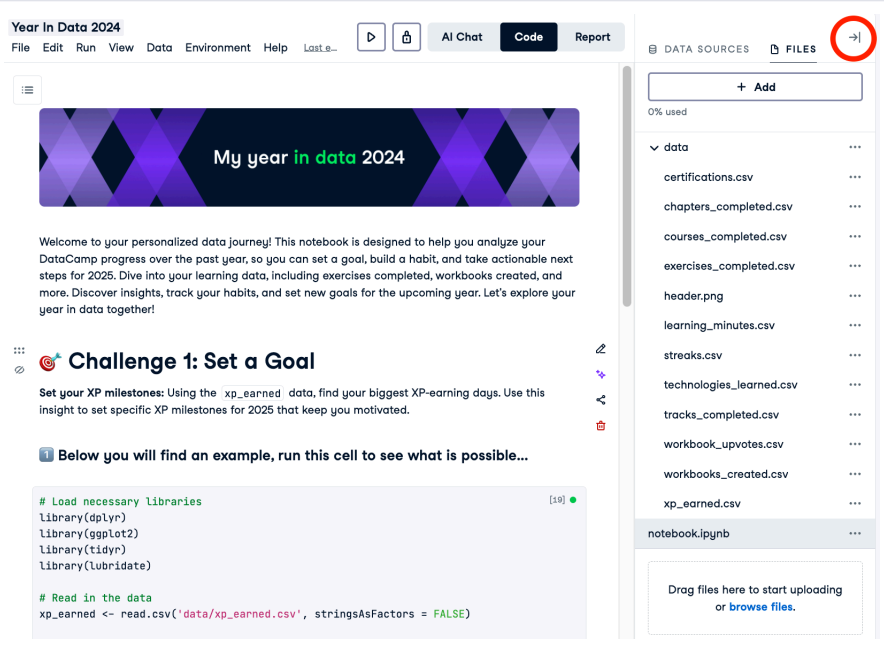
DATACAMP YEAR IN DATA 2024

datalab

Welcome to your personalized data journey! This notebook is designed to help you analyze your DataCamp progress over the past year, so you can set a goal, build a habit, and take actionable next steps for 2025. Dive into your learning data, including exercises completed, workbooks created, and more. Discover insights, track your habits, and set new goals for the upcoming year. Let's explore your year in data together!

🧭 Get familiar and explore your data by opening the context panel

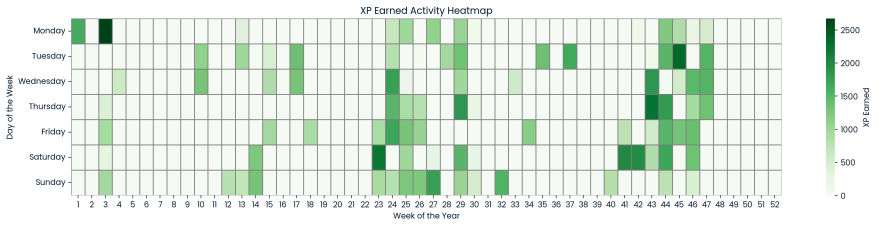


🎯 Challenge 1: Set a Goal

Set your XP milestones: Using the `xp_earned` data, find your biggest XP-earning days. Use this insight to set specific XP milestones for 2025 that keep you motivated.

1️⃣ Below you will find an example, run this cell to see what is possible...

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Read in the data
xp_earned = pd.read_csv('data/xp_earned.csv', parse_dates=
['earned_at'])

# Ensure the 'date' column is in datetime format
xp_earned['date'] = pd.to_datetime(xp_earned['earned_at'])

# Create columns for week number and day of the week
xp_earned['week'] = xp_earned['earned_at'].dt.isocalendar().week
xp_earned['day'] = xp_earned['earned_at'].dt.dayofweek

# Create a DataFrame to ensure all days and weeks are represented
all_weeks = range(1, 53)  # Weeks 1 to 52
all_days = range(0, 7)    # Days 0 (Monday) to 6 (Sunday)
full_index = pd.MultiIndex.from_product([all_days, all_weeks], names=
['day', 'week'])

# Sum XP earned for each day and week, reindex to fill missing values
with 0
heatmap_data = xp_earned.pivot_table(index='day', columns='week',
values='xp_amount', aggfunc='sum', fill_value=0)
heatmap_data = heatmap_data.reindex(index=all_days, columns=all_weeks,
fill_value=0)

# Create the heatmap
plt.figure(figsize=(20, 4))
sns.heatmap(heatmap_data, cmap='Greens', linewidths=.5,
linecolor='gray', cbar_kws={'label': 'XP Earned'})

# Set the labels
plt.title('XP Earned Activity Heatmap', loc='center')
plt.xlabel('Week of the Year')
plt.ylabel('Day of the Week')
plt.yticks(ticks=np.arange(7) + 0.5, labels=['Monday', 'Tuesday',
'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'], rotation=0)

# Display the heatmap
plt.show()
```



2 Now make it your own!

```
xp_earned = pd.read_csv('data/xp_earned.csv', parse_dates=
['earned_at'])
xp_earned.head()

## complete challenge 1, be ambitious
```

	earned_at	xp
0	2024-10-24T03:30:49.000	50
1	2024-11-20T13:31:10.000	50
2	2024-11-15T13:50:18.000	50
3	2024-06-30T15:38:43.000	50
4	2024-06-12T21:58:57.000	100

Rows: 5 Expand

📅 Challenge 2: Track Your Habits

Track your learning streaks: Use the `streaks` data to find your longest streak in 2024. Reflect on the strategies that helped you stay consistent and apply them to maintain or extend streaks in 2025.

Fun fact, learners who extend their daily streak by just two days are **18x** more likely to complete a Career/Skill Track.

```
streaks = pd.read_csv('data/streaks.csv', parse_dates=
['streak_started_at', 'streak_ended_at'])
streaks.head()

## complete challenge 2, be curious
```

	streak_started_at	streak_ended_at
0	2024-06-16T19:41:16.000	2024-06-16T19:41:16.000
1	2024-07-09T20:30:50.000	2024-07-09T20:30:50.000
2	2024-03-05T22:52:08.000	2024-03-05T22:52:08.000
3	2024-08-14T18:35:51.000	2024-08-14T18:35:51.000
4	2024-07-20T19:21:01.000	2024-07-20T19:21:01.000

Rows: 5 Expand

🐾 Challenge 3: Set Next Steps

Identify your most productive days: Analyze the `learning_minutes` data to find which day(s) of the week you dedicated the most time to learning. Consider setting goals for 2025 to make these days even more productive, or identify new days to focus on learning.

```
learning_minutes = pd.read_csv('data/learning_minutes.csv',
parse_dates=['date'])
learning_minutes.head()

## complete challenge 3, be creative
```

	date	total_duration_in_minutes
0	2024-04-09T00:00:00.000	23.8994666667
1	2024-07-17T00:00:00.000	69.6842
2	2024-08-11T00:00:00.000	24.8338666667
3	2024-10-06T00:00:00.000	60.29875
4	2024-01-15T00:00:00.000	147.4764333333

Rows: 5 Expand

Additional Challenges

Here are some examples of how you can utilize your learning data in this notebook to have a more productive 2025.

🎯 Set Goals

These challenges will help you reflect on your achievements and set meaningful goals for the upcoming year.

- **Explore Your Certification Achievements:** Review the certifications earned in `certifications` and decide which certifications you want to pursue in 2025. Use this as a foundation for your learning goals.
- **Find Your Learning Peaks by Course Type:** Analyze the `courses_completed` data to identify the types of courses (beginner, intermediate, advanced) you completed most often. Decide if 2025 will focus on exploring more advanced topics or mastering foundational skills.
- **Set New Track Goals Based on Completed Tracks:** Use the `tracks_completed` data to identify complementary tracks you'd like to pursue in 2025. Focus on building depth or breadth in your learning journey.
- **Analyze Your Skill Development by Technology:** Using `technologies_learned`, reflect on your expertise across different technologies. Set 2025 goals to deepen your knowledge or branch into new areas.

🗓️ Form Habits

Learn from your data to identify habits that have worked well and those you can improve to form better learning routines in 2025.

- **Track Your Learning Streaks:** Use the `streaks` data to find your longest streak in 2024. Reflect on the strategies that helped you stay consistent and apply them to maintain or extend streaks in 2025.
- **Uncover Patterns in Learning Minutes:** Visualize your `learning_minutes` over the year to identify periods of increased or decreased study time. Adjust your schedule to create a more balanced learning routine in 2025.
- **Calculate Your Average Learning Time per Session:** From `learning_minutes`, calculate your average session duration. Decide if you want to increase, maintain, or adjust this time for optimal learning in 2025.

🐾 Take Next Steps

Use these challenges to extract actionable insights from your data and plan concrete steps for improvement.

- **Monthly XP Growth Tracking:** Use `xp_earned` to analyze monthly XP growth. Identify what contributed to high-growth months and plan to replicate these strategies consistently in 2025.
- **Evaluate Your Progress Across Courses and Chapters:** Track the number of `courses_completed` and `chapters_completed` each month. Identify productive periods and plan how to maintain or increase completion rates next year.
- **Identify Your Most Engaged Workbook Creations:** Analyze `workbooks_created` and `workbook_upvotes` to find your most successful creations. Reflect on what worked well and use these insights to create impactful projects in 2025.

- **Analyze Your Progress Across Technologies:** Review the `technologies_learned` data to identify which areas saw the most progress. Plan specific courses or tracks to strengthen your expertise in these areas.

🛠️ Your Data Toolbox

Below, you'll find detailed explanations of the data available to you in this notebook. Use this to be creative and go beyond the challenges provided. Find your own way to extract insights that can help you improve your learning habits for 2025!

Certifications

```
certifications = pd.read_csv('data/certifications.csv', parse_dates=
['certificate_granted_at'])
certifications.head()
```

Your query ran successfully but returned no results.

Column Name	Data Type	Description
certificate_granted_at	datetime64[ns, UTC]	The date when the certificate was granted
certification_name	object	The name of the certification

Chapters completed

```
chapters_completed = pd.read_csv('data/chapters_completed.csv',
parse_dates=['completed_at'])
chapters_completed
```

	completed_at	chapter_title
0	2024-06-20T00:00:00.000	Advanced Merging and Concatenating
1	2024-11-04T00:00:00.000	Uniquely identify records with key constrai
2	2024-06-08T00:00:00.000	NumPy
3	2024-04-06T00:00:00.000	Programming in PySpark RDD?s
4	2024-07-09T00:00:00.000	Visualizing Two Quantitative Variables
5	2024-01-15T00:00:00.000	Data frames
6	2024-03-26T00:00:00.000	Regular Expressions for Pattern Matching
7	2024-07-07T00:00:00.000	Introduction to Seaborn
8	2024-06-12T00:00:00.000	Slicing and Indexing DataFrames
9	2024-03-31T00:00:00.000	Introduction to Big Data analysis with Spar
10	2024-10-24T00:00:00.000	Outer Joins, Cross Joins and Self Joins
11	2024-11-16T00:00:00.000	Data Warehouse Data Modeling
12	2024-07-06T00:00:00.000	Quantitative comparisons and statistical v
13	2024-04-07T00:00:00.000	PySpark SQL & DataFrames
14	2024-11-18T00:00:00.000	Implementation and Data Prep
15	2024-04-24T00:00:00.000	Case Study: Hacker Statistics

Rows: 89 Expand

Column Name	Data Type	Description
completed_at	datetime64[ns, UTC]	The date when the chapter was completed
chapter_title	object	The title of the completed chapter

Courses completed

```
courses_completed = pd.read_csv('data/courses_completed.csv',
parse_dates=['completed_at'])
courses_completed
```

	completed_at	course_title
20	2024-01-01T00:00:00.000	Financial Modeling in Excel
5	2024-01-15T00:00:00.000	Introduction to R
21	2024-03-26T00:00:00.000	Regular Expressions in Python
0	2024-04-10T00:00:00.000	Big Data Fundamentals with PySpark
11	2024-04-24T00:00:00.000	Intermediate Python
13	2024-06-08T00:00:00.000	Introduction to Python
8	2024-06-12T00:00:00.000	Data Manipulation with pandas
12	2024-06-21T00:00:00.000	Joining Data with pandas
3	2024-06-28T00:00:00.000	Introduction to Statistics in Python
16	2024-07-07T00:00:00.000	Introduction to Data Visualization with Mat
2	2024-07-16T00:00:00.000	Introduction to Data Visualization with Sea
19	2024-07-18T00:00:00.000	Introduction to Functions in Python
17	2024-08-11T00:00:00.000	Understanding Data Engineering
10	2024-08-23T00:00:00.000	Python Toolbox
6	2024-10-06T00:00:00.000	Exploratory Data Analysis in Python
1	2024-10-12T00:00:00.000	Introduction to SQL

Rows: 22 Expand

Column Name	Data Type	Description
completed_at	datetime64[ns, UTC]	The date when the course was completed
course_title	object	The title of the completed course

Exercises completed

```
exercises_completed = pd.read_csv('data/exercises_completed.csv',
parse_dates=['completed_at'])
exercises_completed
```

	completed_at	exercise_title
0	2024-07-16T00:00:00.000	FacetGrids vs. AxesSubplots
1	2024-11-20T00:00:00.000	JSONified (2)
2	2024-04-23T00:00:00.000	Determine your next move
3	2024-07-18T00:00:00.000	Map() and lambda functions
4	2024-01-01T00:00:00.000	Two is better than one
5	2024-11-16T00:00:00.000	The OLAP data cube
6	2024-10-23T00:00:00.000	The ins and outs of INNER JOIN
7	2024-03-26T00:00:00.000	Flying home (3)
8	2024-06-21T00:00:00.000	Descriptive and inferential statistics
9	2024-11-13T00:00:00.000	Converting to 2NF (2)
10	2024-11-03T00:00:00.000	Primary keys
11	2024-07-15T00:00:00.000	Customizing point plots (2)
12	2024-03-26T00:00:00.000	Playing safe (2)
13	2024-06-23T00:00:00.000	Distribution of Amir's sales (2)
14	2024-10-31T00:00:00.000	Subquery inside FROM (1)
15	2024-11-16T00:00:00.000	Data warehouse data modeling

Rows: 1,664 Expand

Column Name	Data Type	Description
completed_at	datetime64[ns, UTC]	The date when the exercise was completed
exercise_title	object	The title of the completed exercise

Learning minutes

```
learning_minutes = pd.read_csv('data/learning_minutes.csv',
parse_dates=['date'])
print(learning_minutes)
avg_lm_2024 = learning_minutes['total_duration_in_minutes'].mean()
avg_lm_2024
```

```
                       date  total_duration_in_minutes
0   2024-04-09 00:00:00+00:00                  23.899467
1   2024-07-17 00:00:00+00:00                  69.684200
2   2024-08-11 00:00:00+00:00                  24.833867
3   2024-10-06 00:00:00+00:00                  60.298750
4   2024-01-15 00:00:00+00:00                 147.476433
..                        ...                        ...
118 2024-01-25 00:00:00+00:00                   0.925250
119 2024-06-20 00:00:00+00:00                  80.590133
```