AI assistants

SAAVEDRA ERWIN
DUDIC MATEJA
KRYLOVA ALENA
MARINGER KELVIN

Introduction

- Dataset about daily AI assistant usage
- Focuses on **when**, **how**, and **for what purposes** users interact with AI
- *Dataset Structure:* **300 rows and 8 columns**

Packages used

- Pandas
- NumPy
- Seaborn
- Matplotlib

Data Overview

Categorical features:

- `device`
- `usage_category`
- `assistant_model`

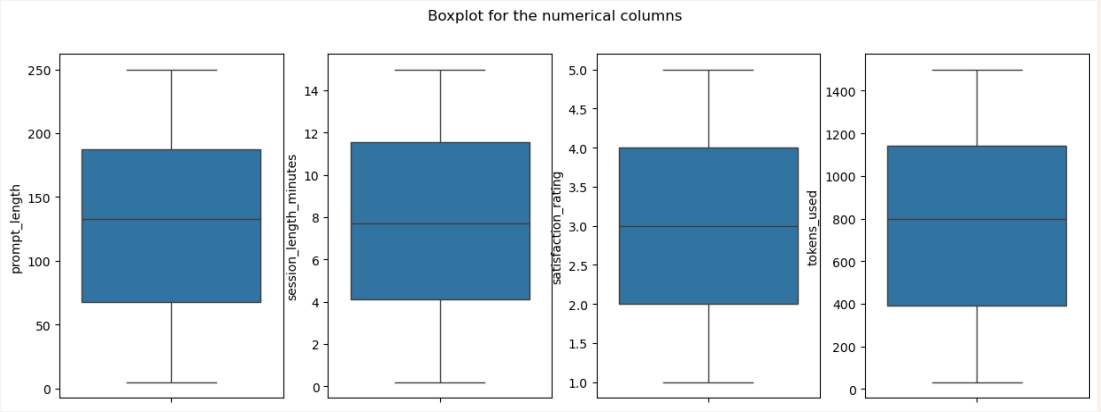
Numerical features:

- `prompt_length`
- `session_length_minutes`
- `tokens_used`
- `satisfaction_rating`

Additional feature: timestamp
The dataset combines categorical and numerical features, allowing analysis of usage patterns and session characteristics.

Data Quality check

- The dataset was checked for missing values and outliers
- No missing values were detected in any of the features
- Boxplots showed no significant outliers in numerical values.
- Frequency analysis of categorical features revealed no unusual or rare values.



Data Preprocessing

Tasks:

1. Handle missing values and outliers appropriately.
2. Create a new column timeOfDay, which bins the timestamp into morning, afternoon, evening, and night.
3. Create a new column year, which is derived from the timestamp column.
4. Convert data types where needed (e.g., categorical columns).

Missing values and outliers

There were no outliers neither missing data , everything was surprisal clean, so I had to do nothing, For this task.

Create column timeOfDay And column Year

```
RAW_data["timeOfDay"] = RAW_data["timestamp"].apply(lambda x:timeOfDay(int(x[11:-6])
)
RAW_data["year"] = RAW_data["timestamp"].apply(lambda x:int(x[0:4]))
```

```
def timeOfDay(hour:int):
    if 5 <= hour <= 11:
        return "morning"
    elif 12 <= hour <= 17:
        return "afternoon"
    elif 18 <= hour <= 22:
        return "evening"
    elif hour <0 or hour > 24:
        raise Exception("invalid hour")
    else:
        return "Night"
```

	timeOfDay	year	timestamp
0	Night	2025	2025-02-20 03:29:00
1	evening	2025	2025-01-08 18:28:00
2	afternoon	2025	2025-01-12 17:56:00
3	morning	2025	2025-01-04 09:11:00
4	evening	2025	2025-02-14 19:59:00

Convert to the correct datatypes. Code

```
RAW_data["timestamp"] = pd.to_datetime(RAW_data["timestamp"])
RAW_data["device"] = RAW_data["device"].astype("category")
RAW_data["assistant_model"] = RAW_data["assistant_model"].astype("category")
RAW_data["timeOfDay"] = RAW_data["timeOfDay"].astype("category")
RAW_data["usage_category"] = RAW_data["usage_category"].astype("category")
```

Convert to the correct datatypes. Visual

Before

```
timestamp                 object
device                    object
usage_category            object
prompt_length              int64
session_length_minutes   float64
satisfaction_rating        int64
assistant_model           object
tokens_used                int64
timeOfDay                 object
year                       int64
dtype: object
```

After

```
timestamp           datetime64[ns]
device                    category
usage_category            category
prompt_length                int64
session_length_minutes     float64
satisfaction_rating          int64
assistant_model           category
tokens_used                  int64
timeOfDay                 category
year                         int64
dtype: object
```

Data Analysis #1

- How many different AI assistants are in the dataset? Show counts and percentages.
- What is the average session length per assistant? Are there noticeable differences?
- Are specific assistants used more often for certain tasks (e.g., education)?
- Which type of tasks has the longest average prompt size and use time?
- Is there a pattern between the task type and the time of day it was used?
- Have some AI assistants become more popular over time?

How many different AI assistants are in the dataset?

```
assistant = RAW_data['assistant_model'] #selecting the column assistant_model
count = assistant.value_counts() #counting how many times each assistant appears
percentage = ((count/RAW_data['assistant_model'].count())*100).round(2) #calculating the
 percentage of each assistant occurrence
```

```
different_assistants = pd.DataFrame({'count': count, 'percentage': percentage}) #making a
 dataframe with the results
print(different_assistants)
```

There is 5 different AI assistants

GPT-4 occurs the most in the dataset

assistant_model	count	percentage
GPT-4o	79	26.33
o1	59	19.67
GPT-5	56	18.67
Mini	55	18.33
GPT-5.1	51	17.00

What is the average session length per assistant? Are there noticeable differences?

```
average_session_length = RAW_data.groupby('assistant_model')['session_length_minutes']
 .mean().round(2)
average_session_length_output = pd.DataFrame({'Average length': average_session_length})
print(average_session_length_output)
```

There is no significant differences in average session length per assistant

GPT-5 has the longest average session length at 8.15 minutes

O1 has the shortest average session length at 7.18 minutes

assistant_model	Average length
GPT-4o	7.76
GPT-5	8.15
GPT-5.1	8.12
Mini	7.58
o1	7.18

Are specific assistants used more often for certain tasks?

```
#3. Usage category per assistant model
usage_per_assistant = pd.pivot_table(RAW_data, index='assistant_model',
 columns='usage_category', aggfunc='count', values='timestamp')
print(usage_per_assistant)
```

It is noticeable that o1 is used more for education and writing than other tasks

```
usage_category   Coding  Daily Tasks  Education  Entertainment  Productivity  \
assistant_model
GPT-4o                8            9         14             14            15
GPT-5                 7            8         10              4            11
GPT-5.1               7           11          6             10             5
Mini                  9            4          8              7             9
o1                    5            4         16              5             6

usage_category   Research  Writing
assistant_model
GPT-4o                 10        9
GPT-5                   6       10
GPT-5.1                 8        4
Mini                   10        8
o1                      8       15
```

Which type of tasks has the longest average prompt size and use time?

```
longest_avg_prompt = RAW_data.groupby('usage_category')['prompt_length'].mean().round(2)
print(longest_avg_prompt)
```

```
longest_avg_time = RAW_data.groupby('usage_category')['session_length_minutes'].mean()
 .round(2)
print(longest_avg_time)
```

usage_category	
Coding	8.51
Daily Tasks	7.66
Education	8.23
Entertainment	7.01
Productivity	8.42
Research	7.30
Writing	7.04

Average session length

The longest average prompt length is in Research with 141.26 characters

The longest average session length is in Coding with 8.51 minutes

usage_category	
Coding	118.56
Daily Tasks	121.42
Education	123.57
Entertainment	122.52
Productivity	140.52
Research	141.26
Writing	133.20

Average prompt length

Is there a pattern between the task type and the time of day it was used?

```
usage_category_per_timeOfDay = pd.pivot_table(RAW_data, index='usage_category',
 columns='timeOfDay', aggfunc='count', values='timestamp')
print(usage_category_per_timeOfDay)
```

Most categories have a specific time of day during which they are clearly less used.

Education is used the least at night

Writing is used the least during the evening

timeOfDay	Night	afternoon	evening	morning
usage_category				
Coding	12	6	8	10
Daily Tasks	7	7	9	13
Education	18	17	7	12
Entertainment	7	6	9	18
Productivity	12	7	16	11
Research	10	12	12	8
Writing	14	14	4	14

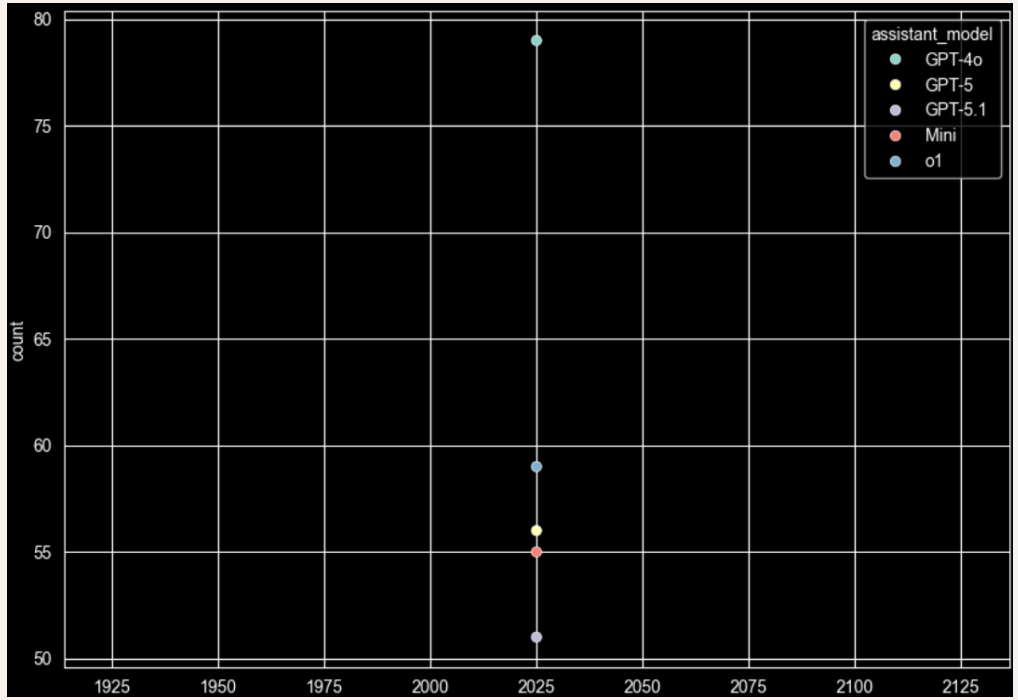
Have some AI assistants become more popular over time?

```
assistant_model_per_year = pd.pivot_table(RAW_data, index='assistant_model',
 columns='year', aggfunc='count', values='timestamp')
print(assistant_model_per_year)
```

```
usage_per_year = (RAW_data.groupby(['assistant_model', 'year']).size().reset_index
 (name='count'))
plt.figure(figsize=(10, 7))
sea.scatterplot(data=usage_per_year , x='year', y='count', hue='assistant_model')
```

There is only one year in the dataset and GPT-4 is the most used model

year	2025
assistant_model	
GPT-4o	79
GPT-5	56
GPT-5.1	51
Mini	55
o1	59

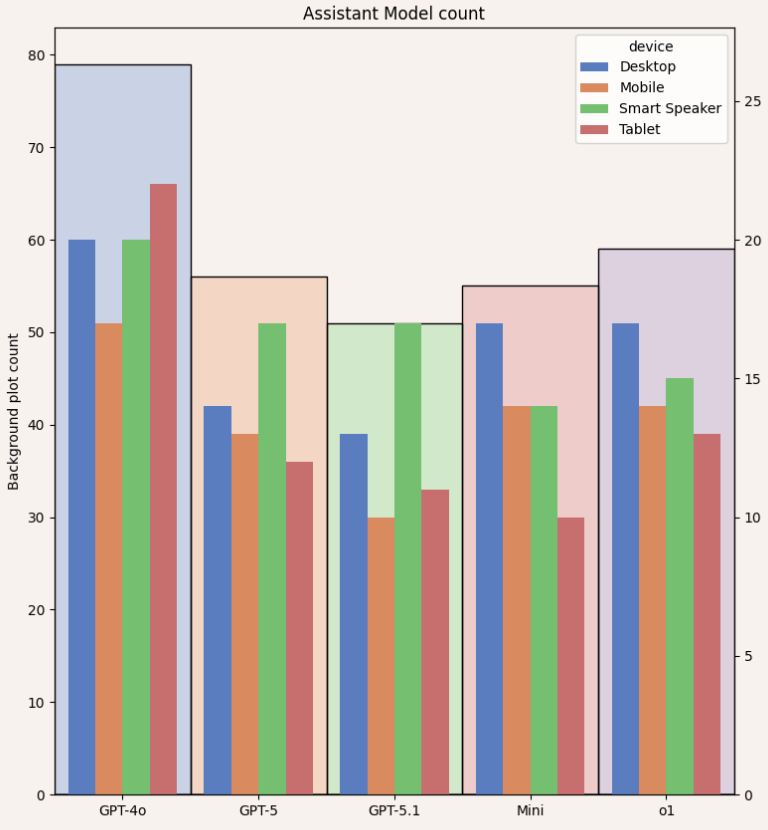


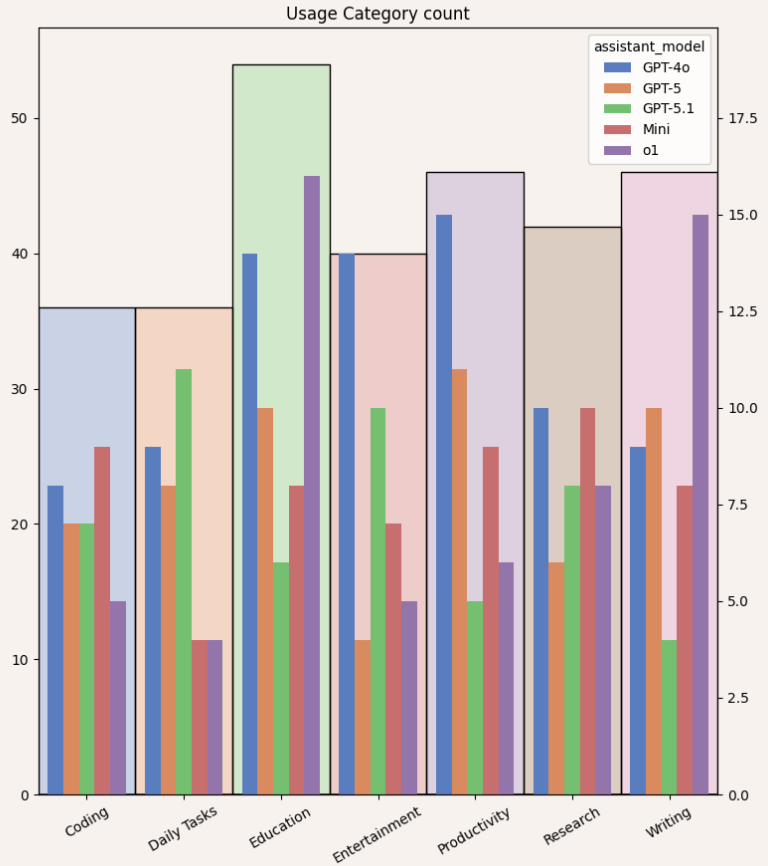
Data Analysis #2

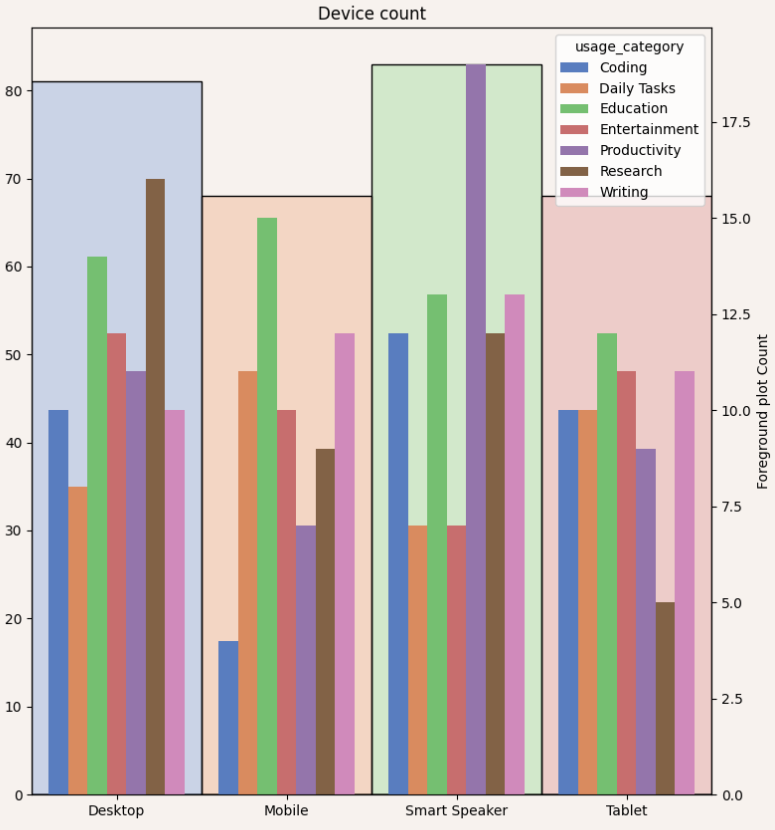
- Visualizations
- Are there any features that clearly differentiate device types?
- Use pair plots to explore feature relationships, color-coded by the assistant model.
- Correlation Analysis
- Time-Based Analysis.

Visualization - Histograms

Histograms (∑300 entries)

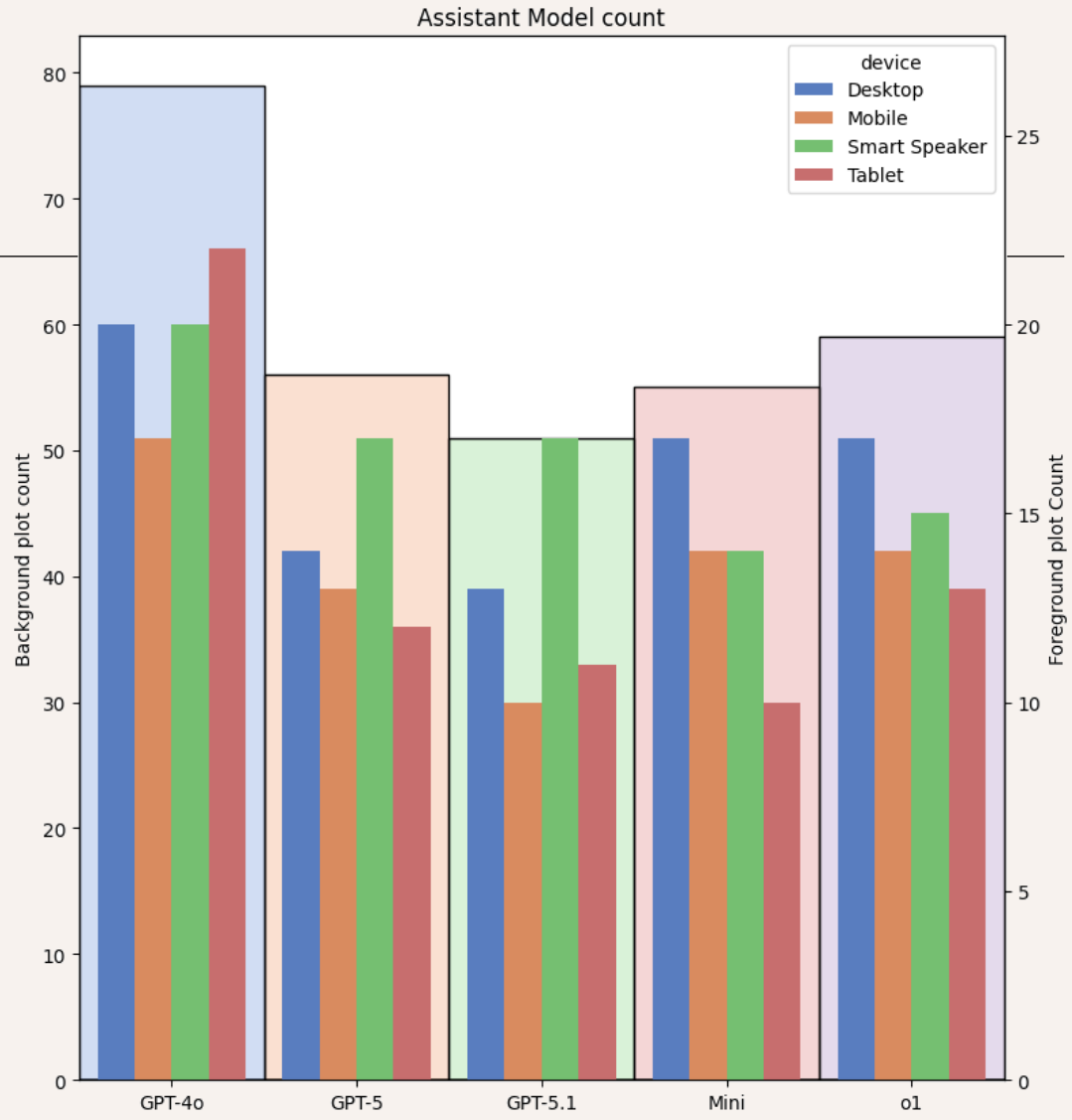






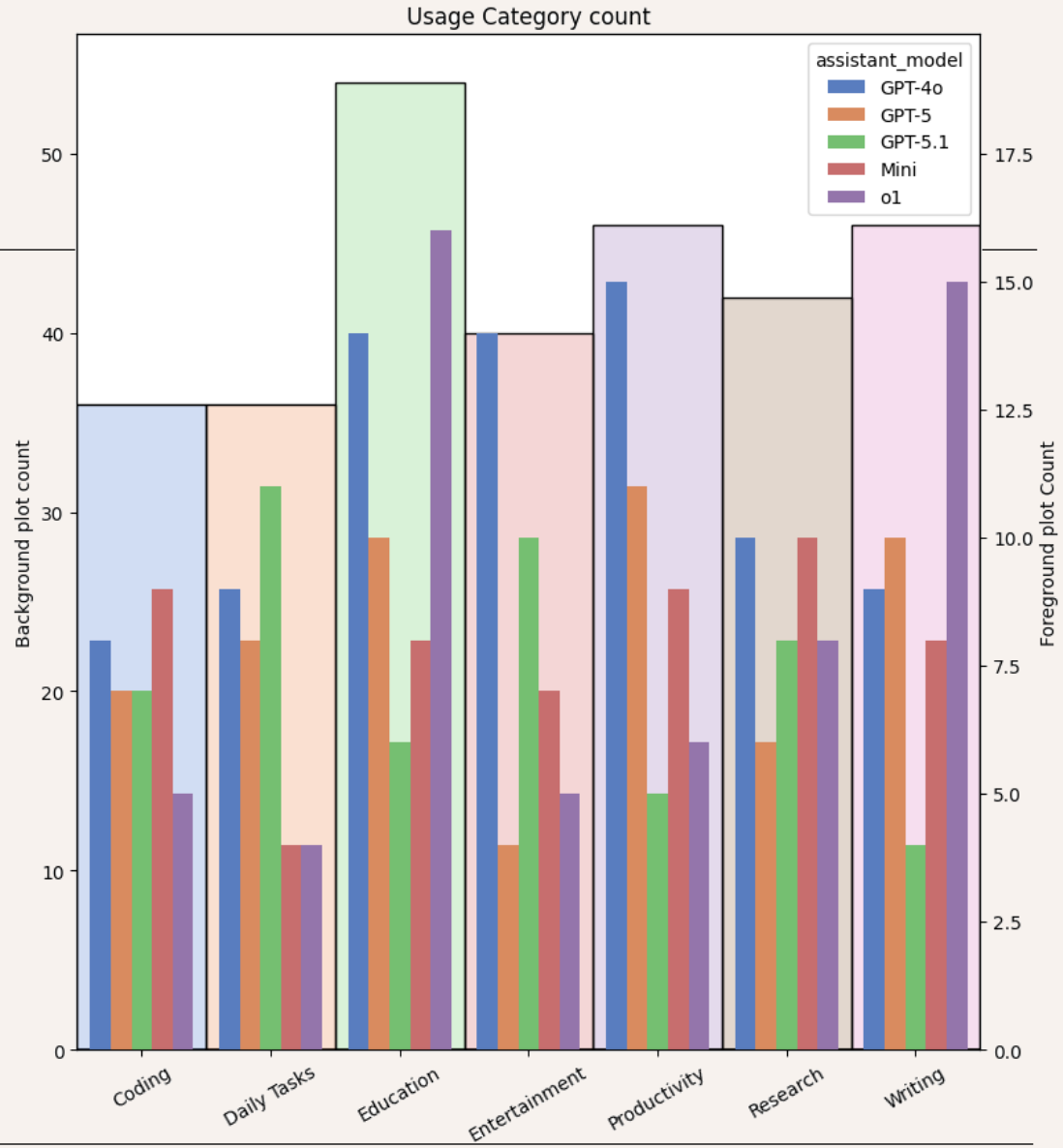
Visualization

- GPT-4o is the most used model
- GPT-5.1 is often used on smart speakers



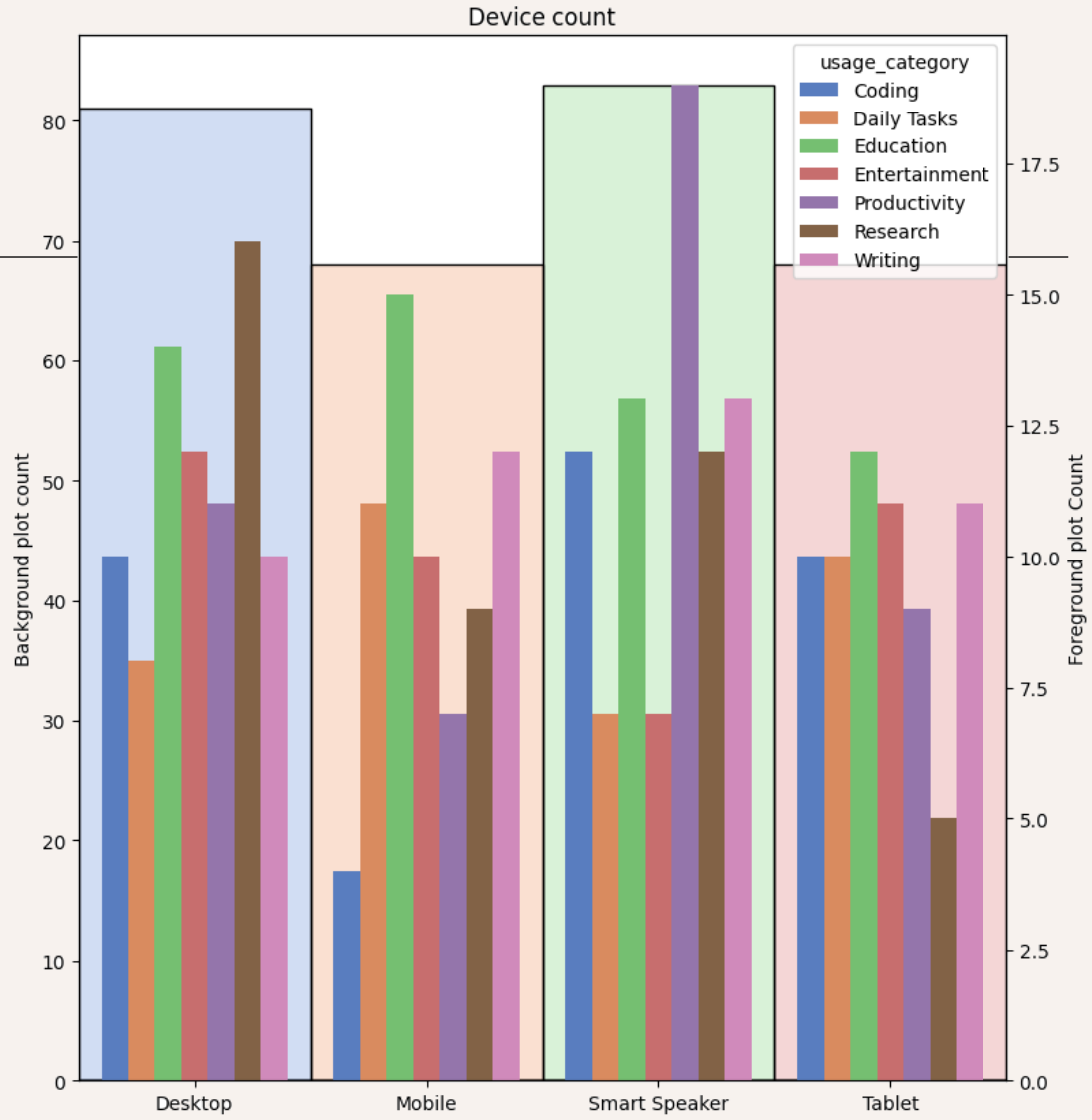
Visualization

- AI assistants are often used for Education
- o1 is often used for writing, not for productivity



Visualization

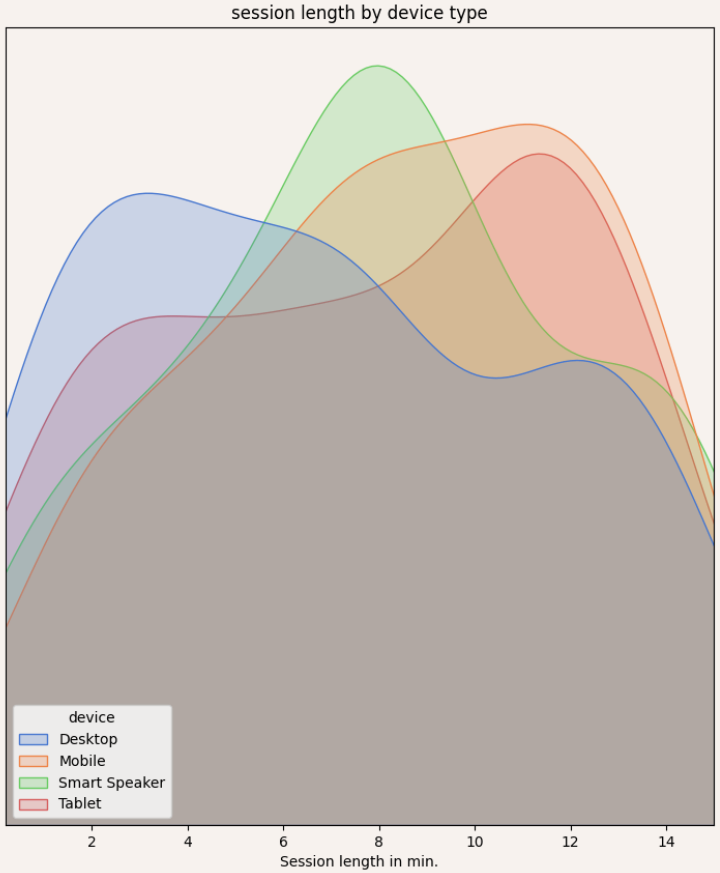
- Desktops and smart speakers are often used
- Smart Speakers are mostly used for productivity
- -> Smart Speakers are also used for coding

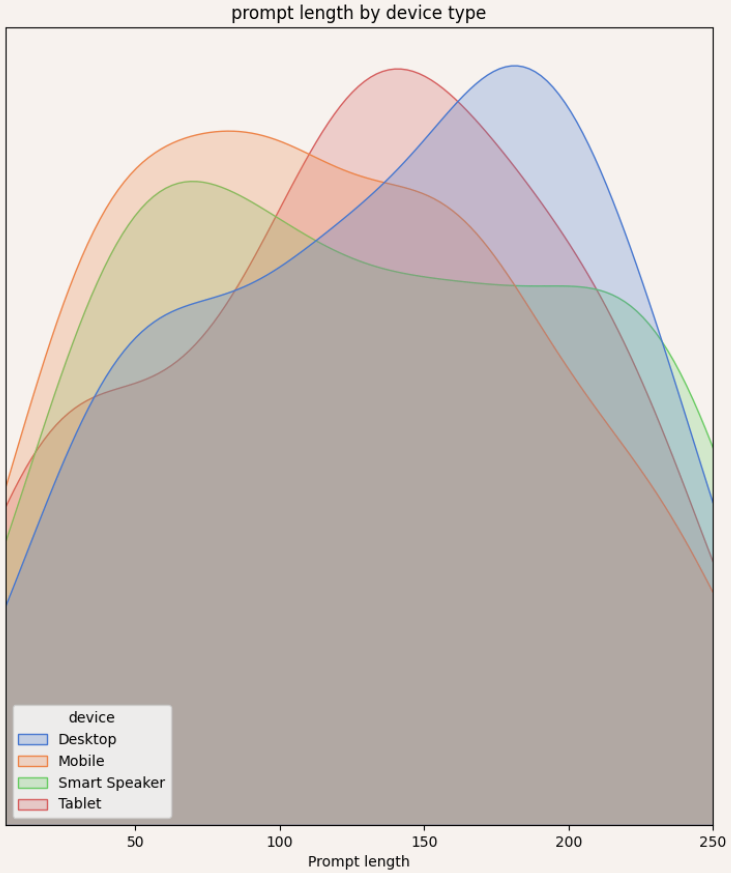


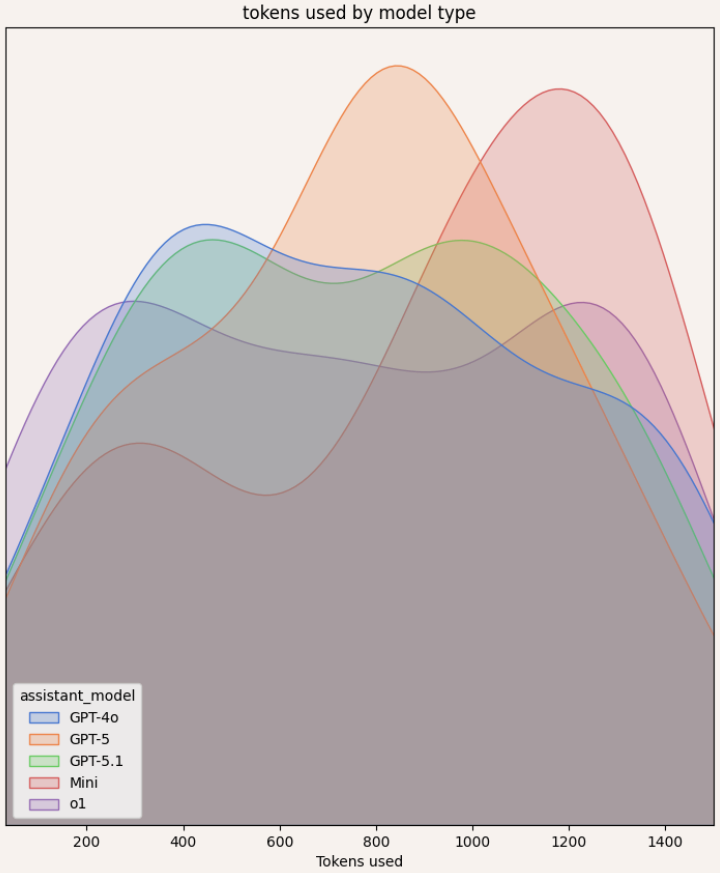
Coding on Smart Speakers

Visualization – Distribution

KDE Plots

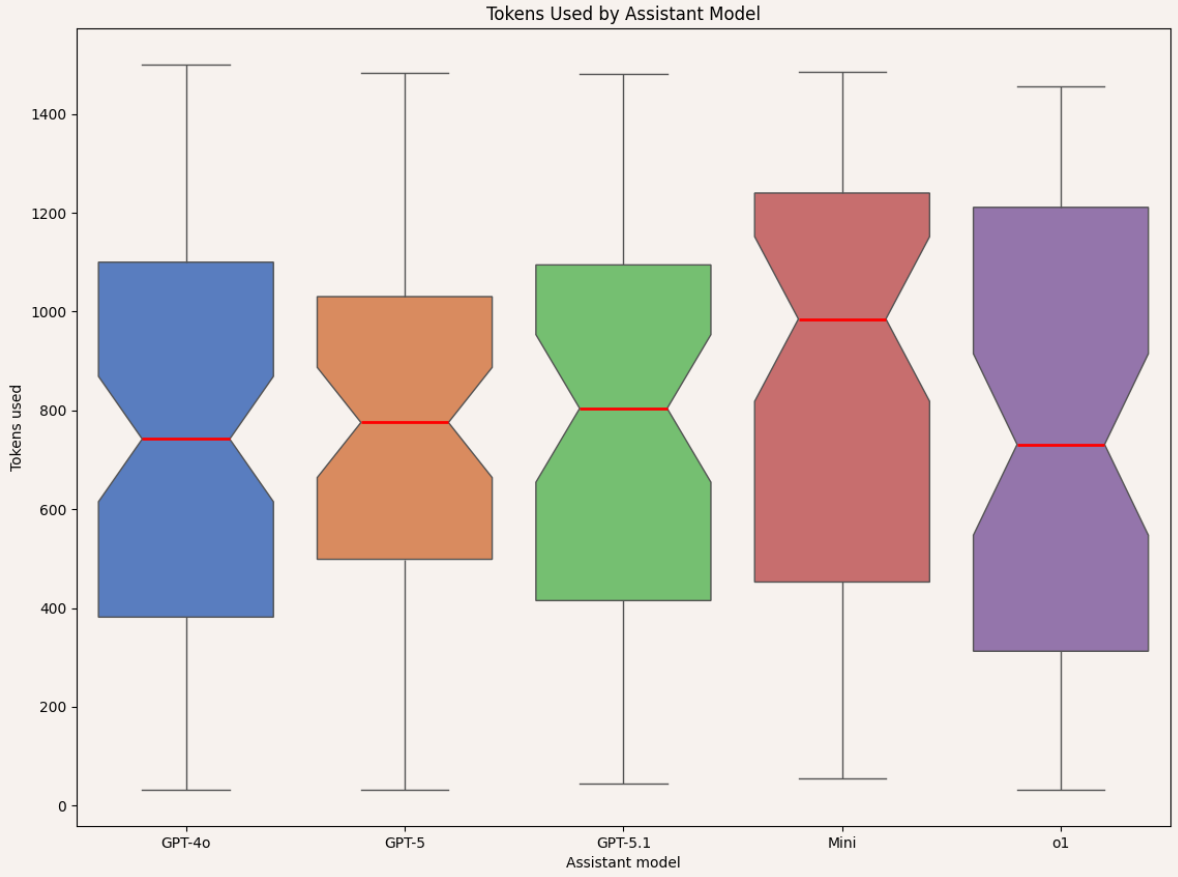


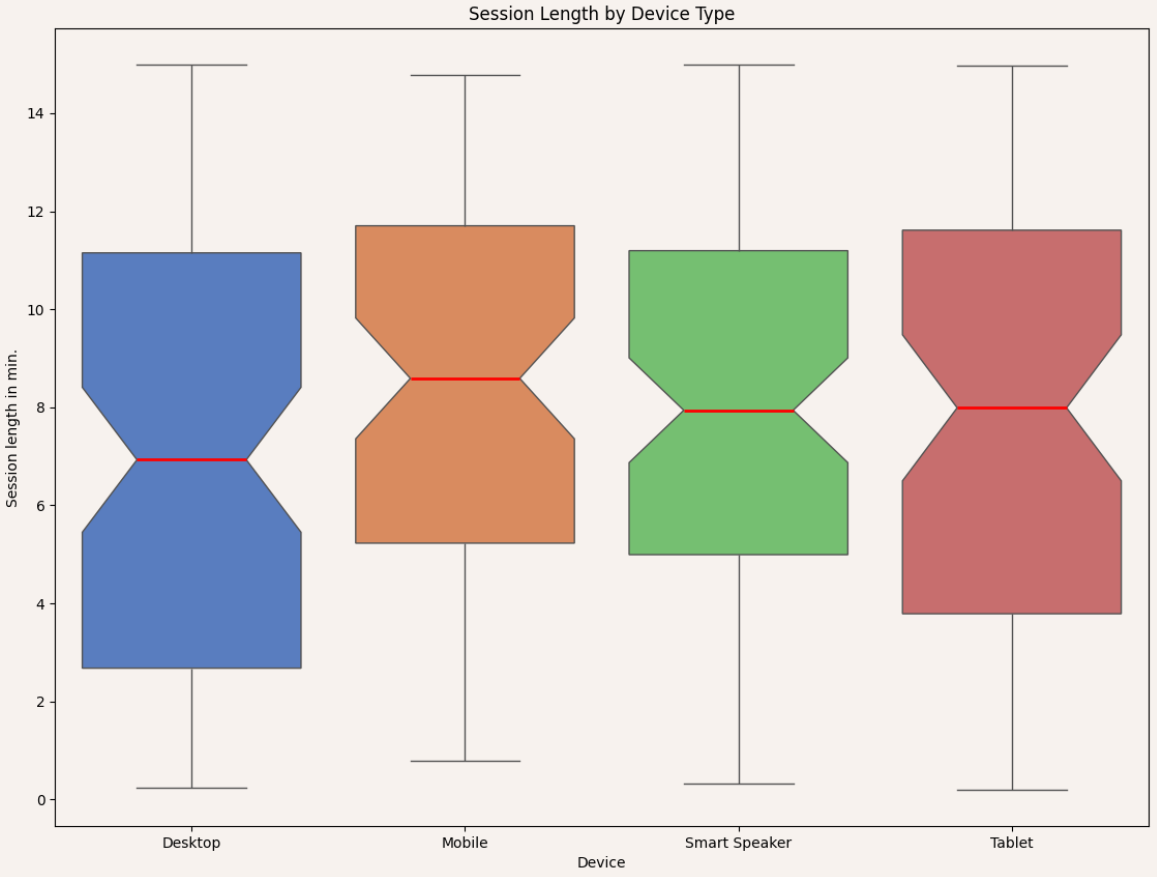




Visualization – Box Plots

Box Plots





Are there any features that clearly differentiate device types?

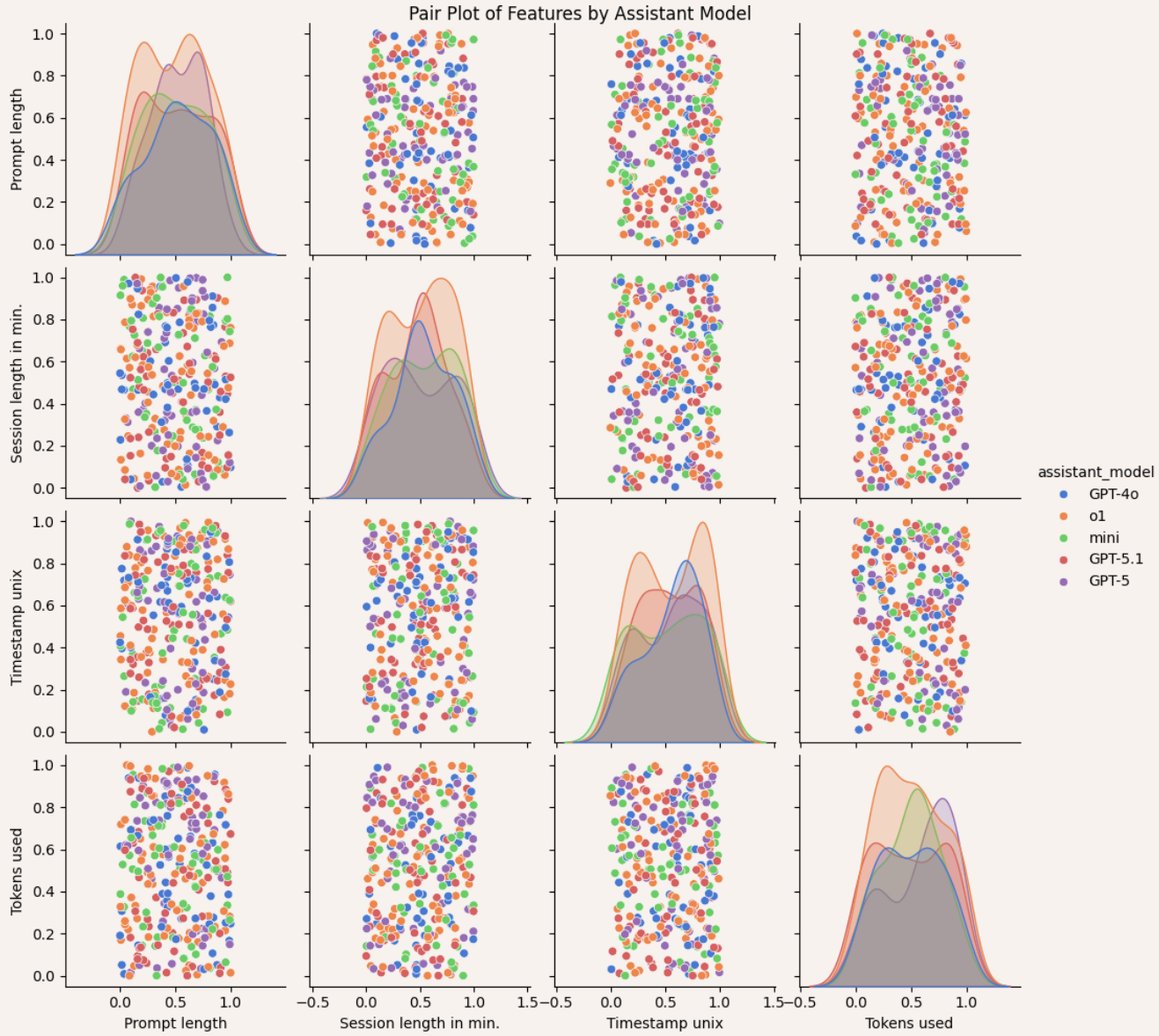
- **No.**

- Best features for device-type estimation:
  - Usage-category
  - Prompt-length
  - Session length
- -> ONLY USEFULL TOGETHER

Pair Plots

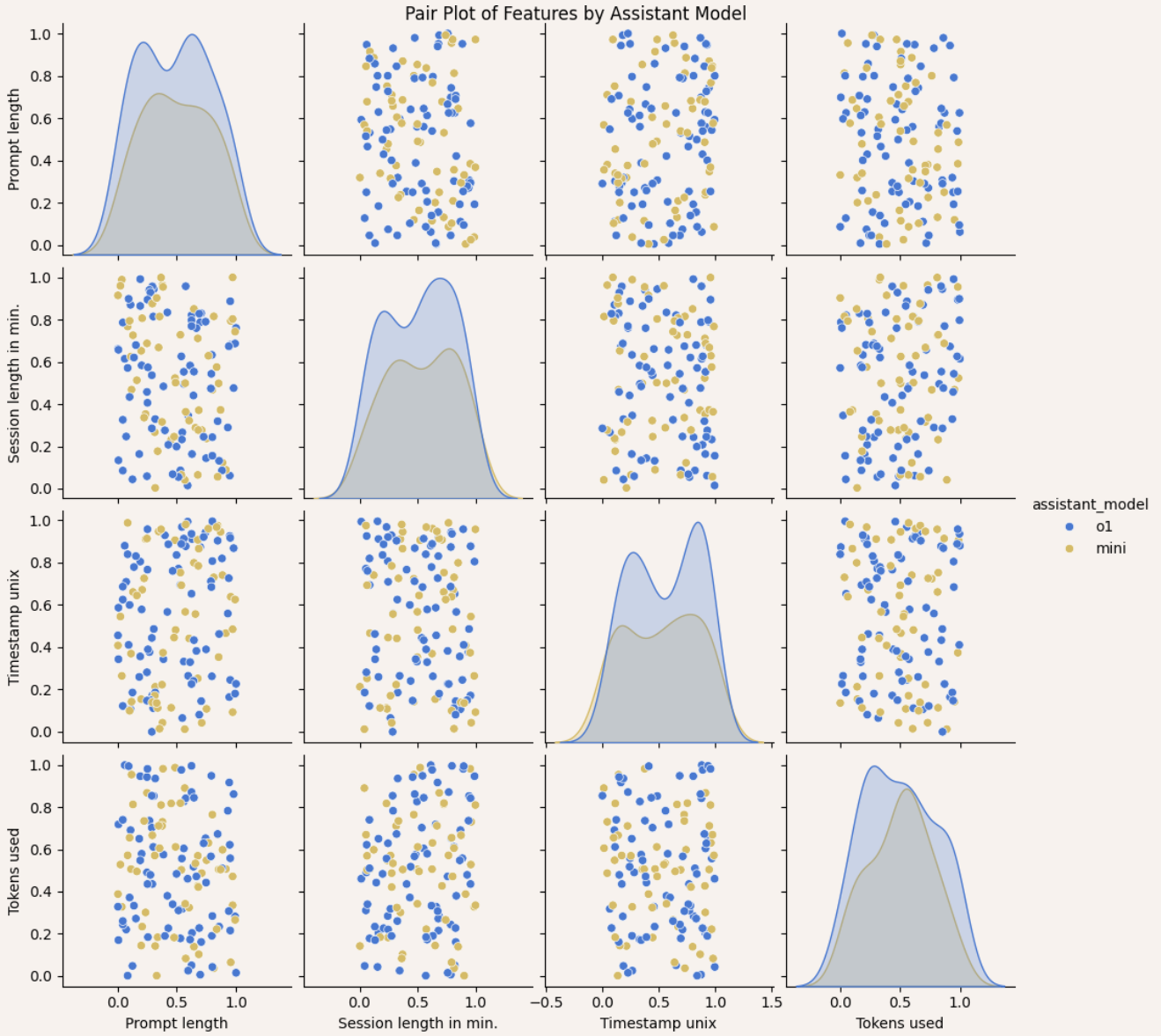
…

What?



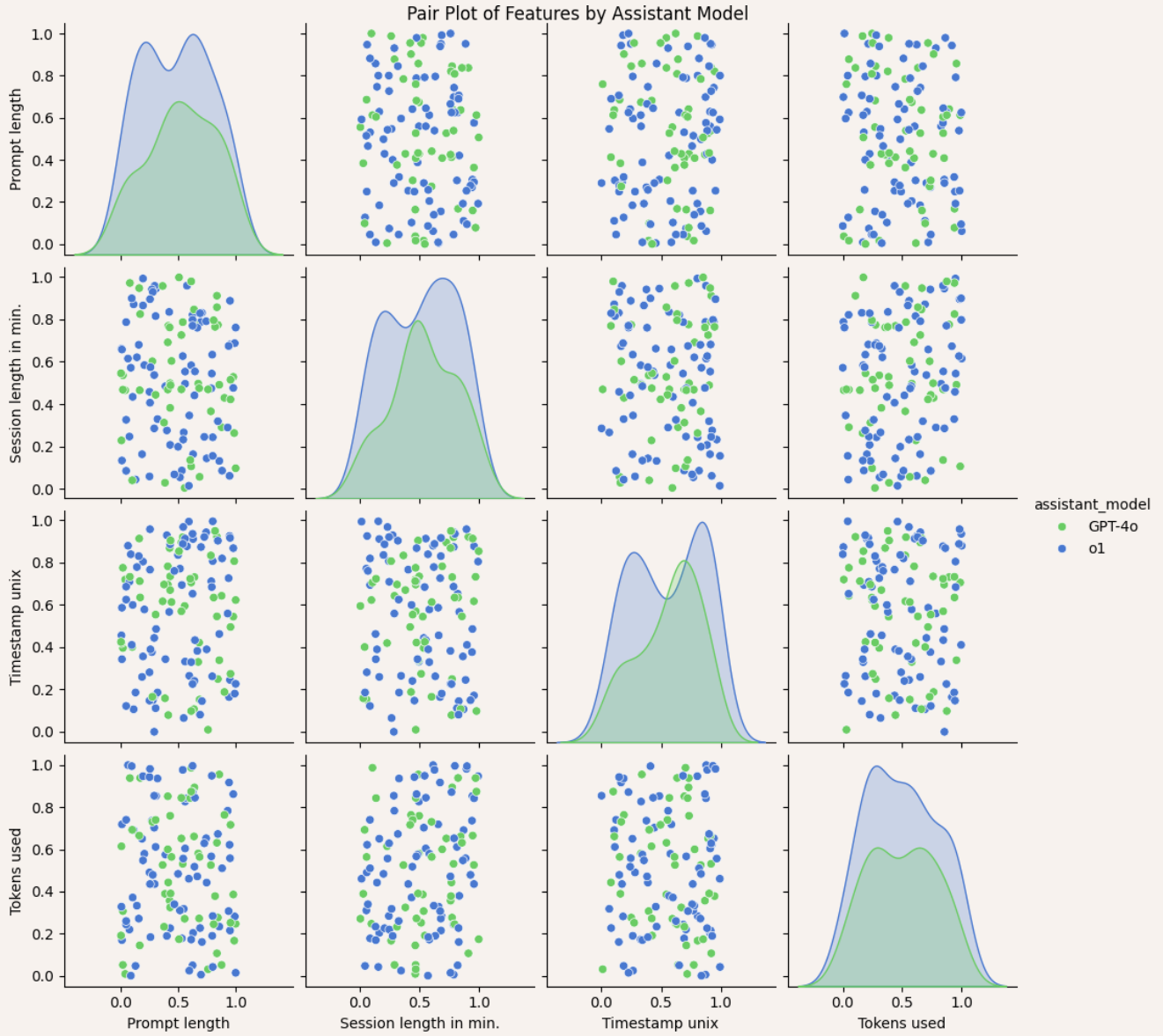
Pair Plots

...


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
o1
mini

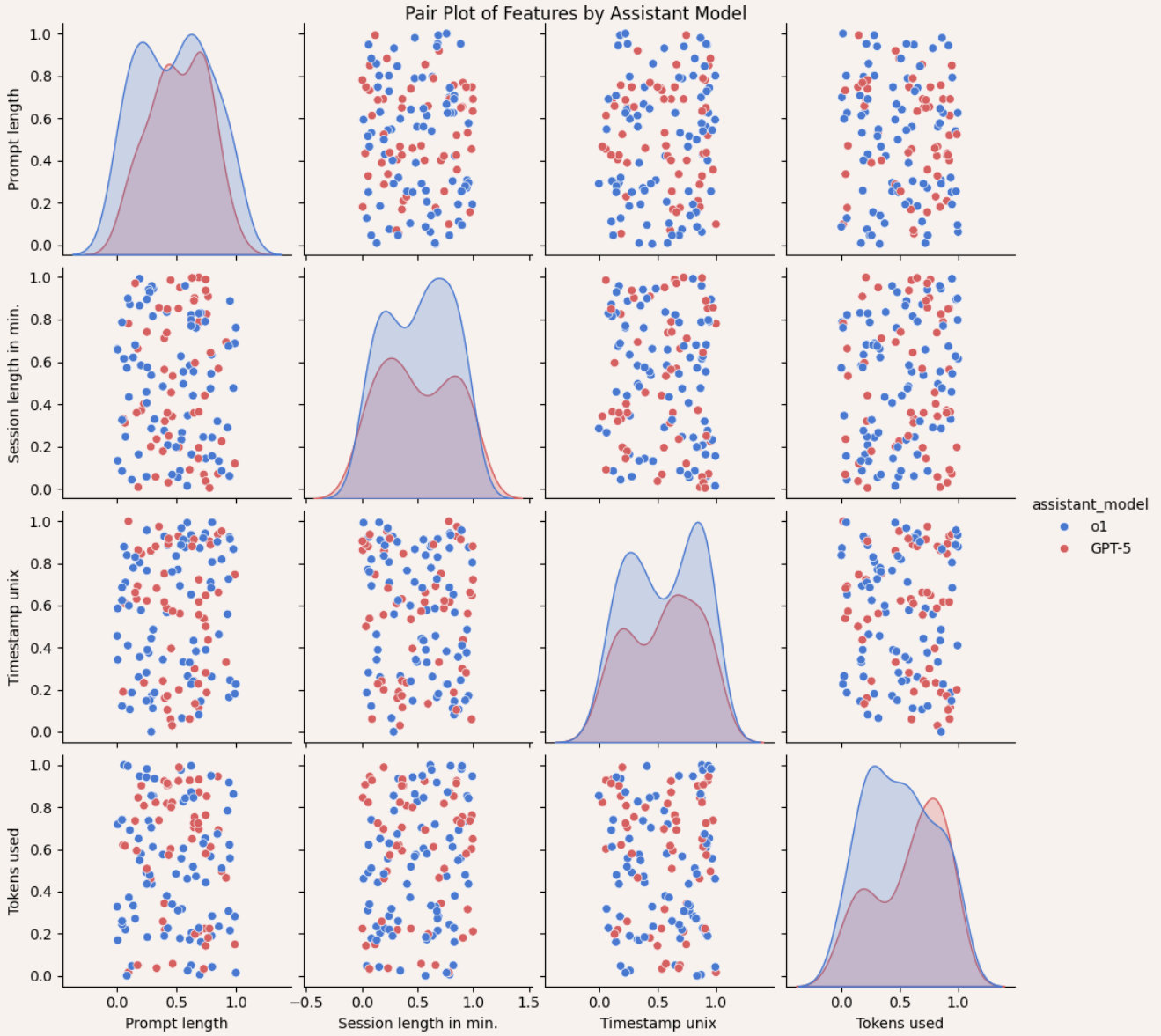
Pair Plots

....


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
GPT-4o
o1

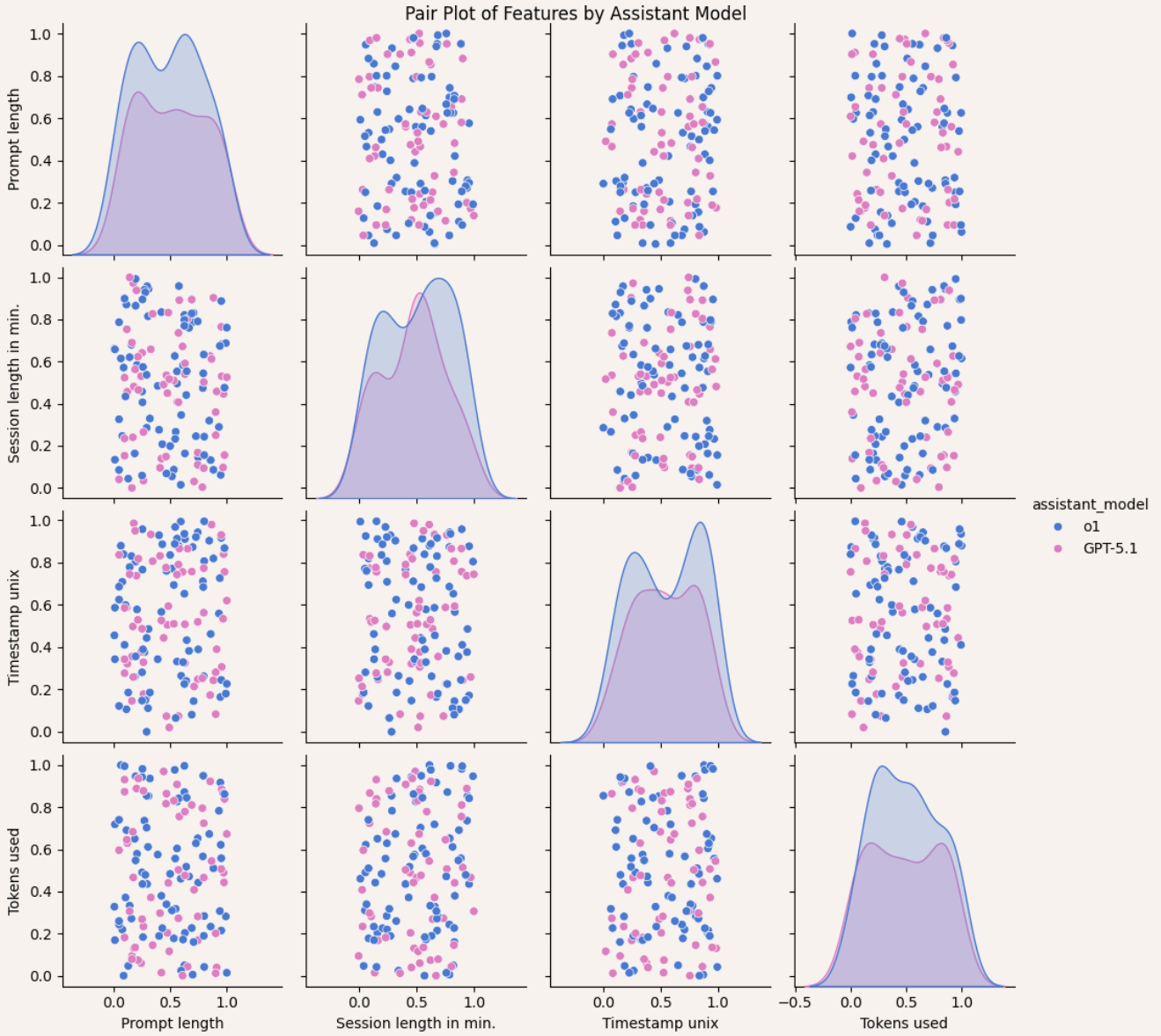
Pair Plots

.....


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
o1
GPT-5

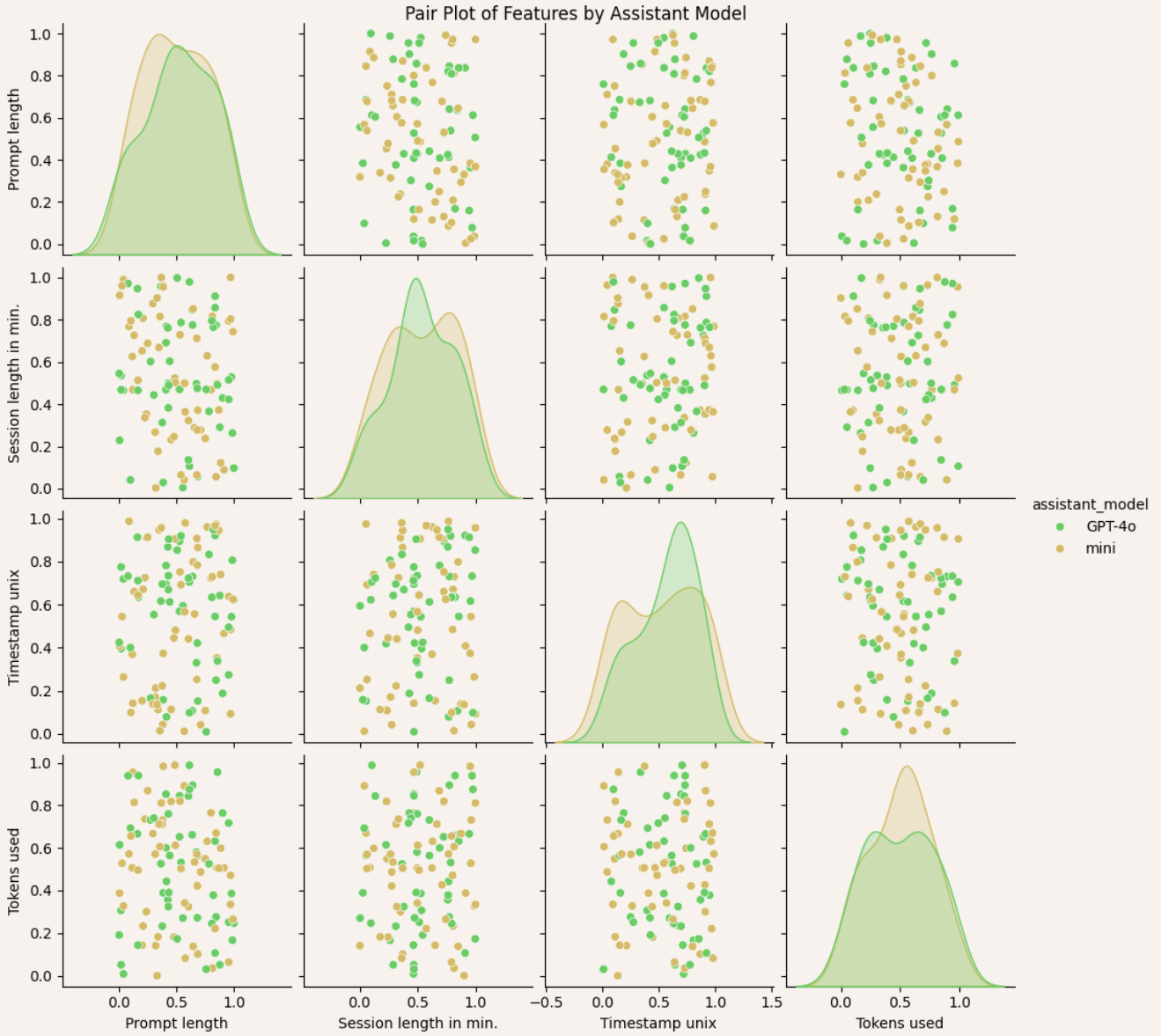
Pair Plots

.....


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
o1
GPT-5.1

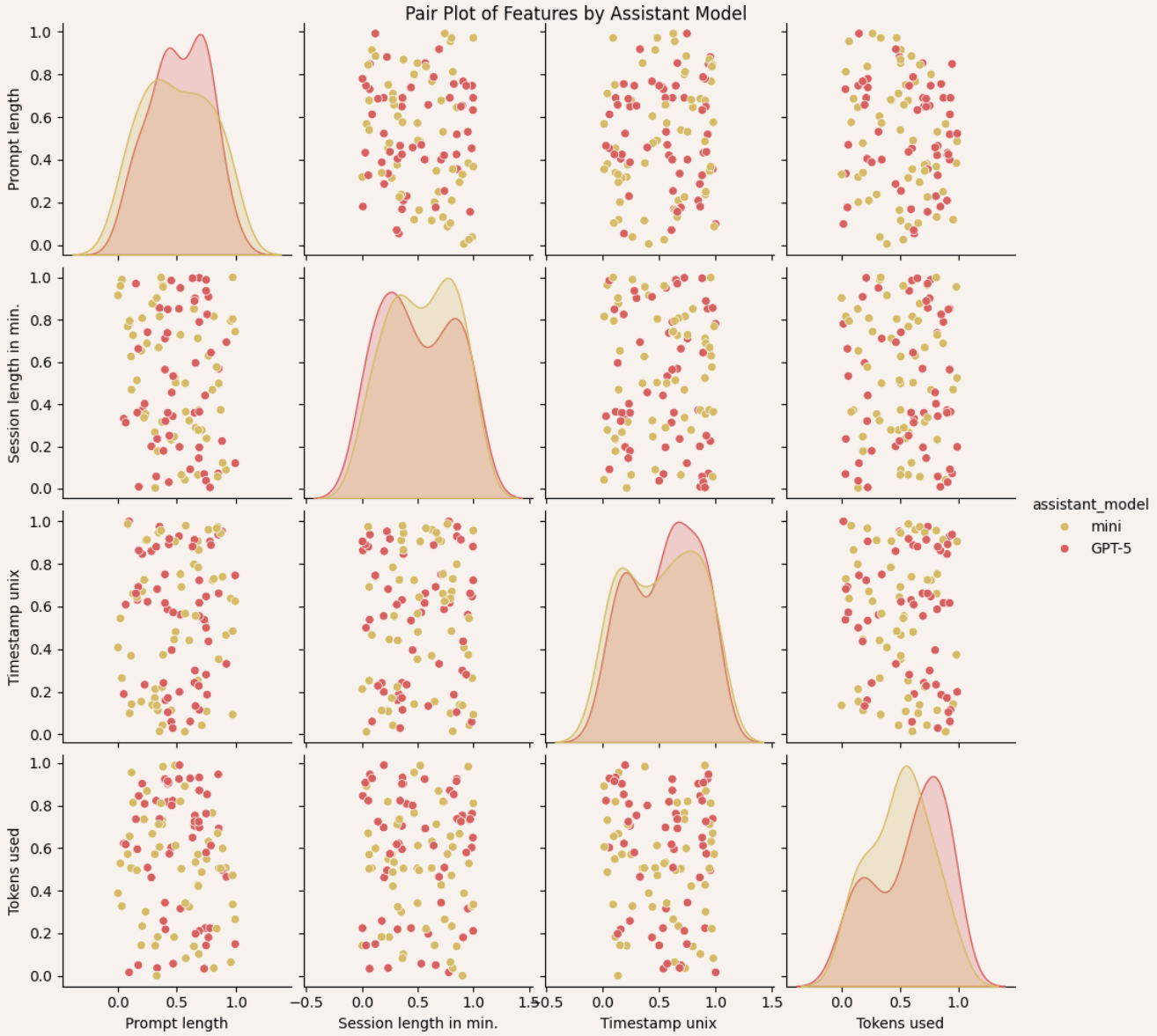
Pair Plots

.....


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
GPT-4o
mini

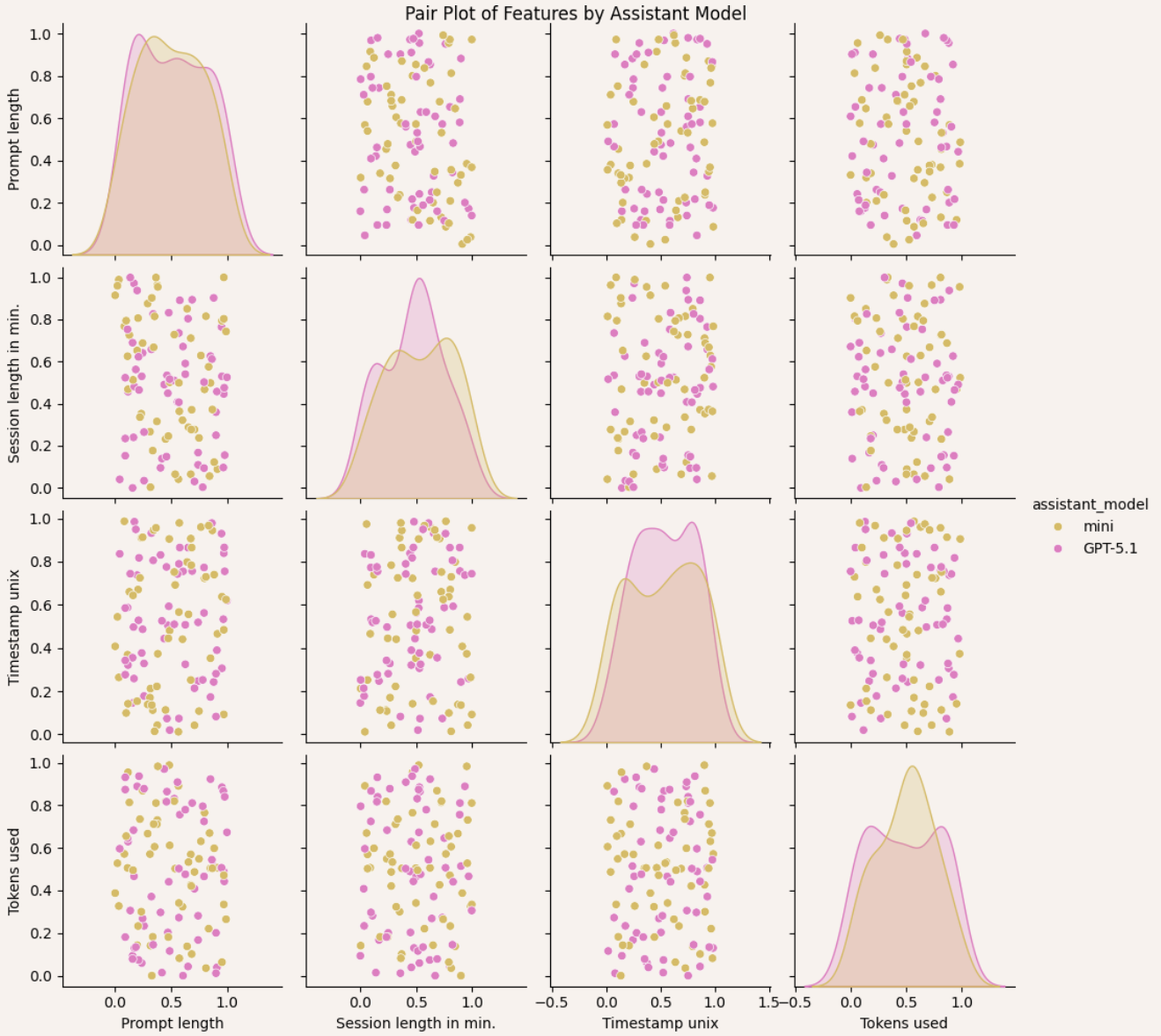
Pair Plots

……..


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
mini
GPT-5

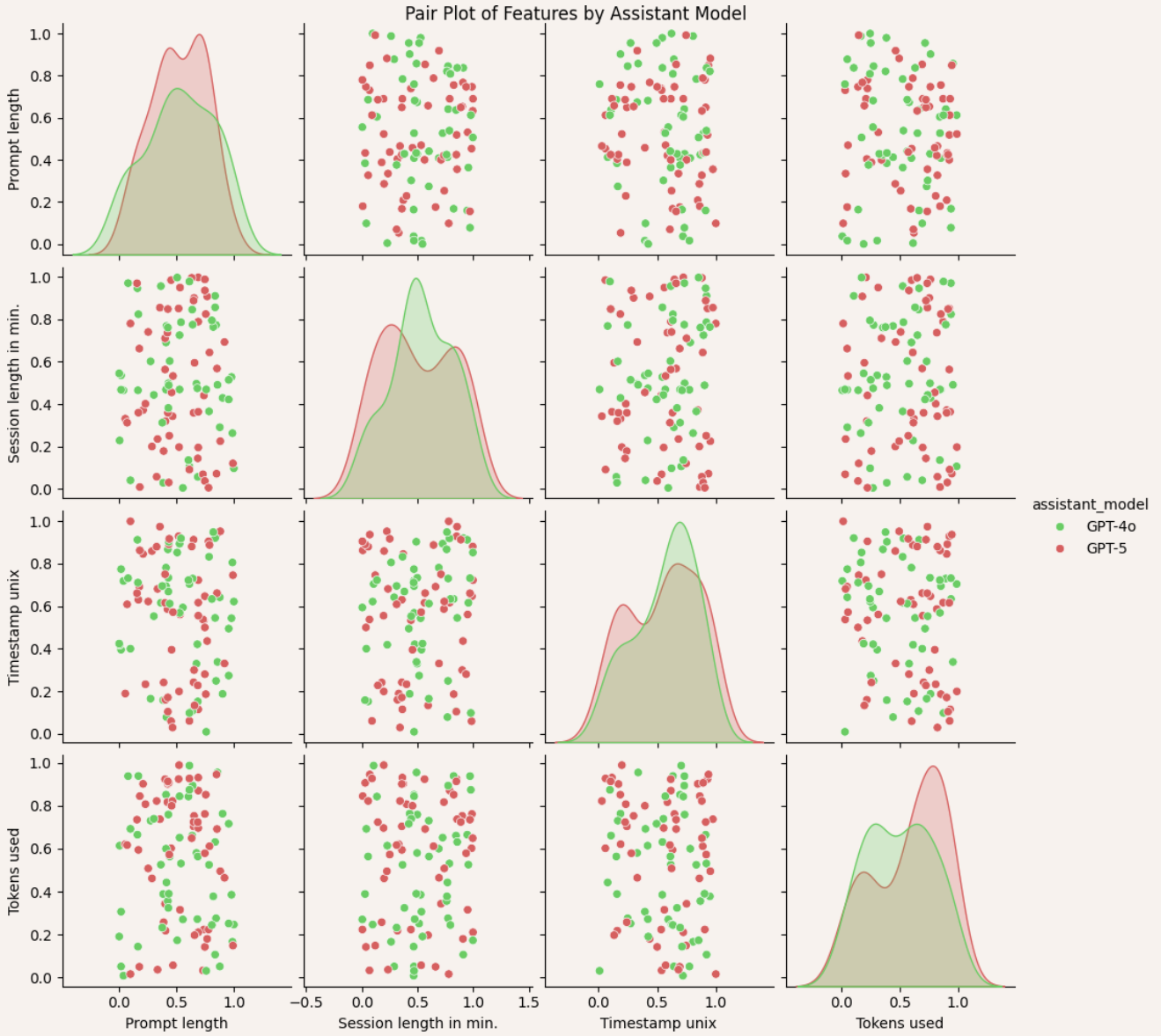
Pair Plots

.....


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
mini
GPT-5.1

Pair Plots

.....


Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
GPT-4o
GPT-5

Pair Plots

.....

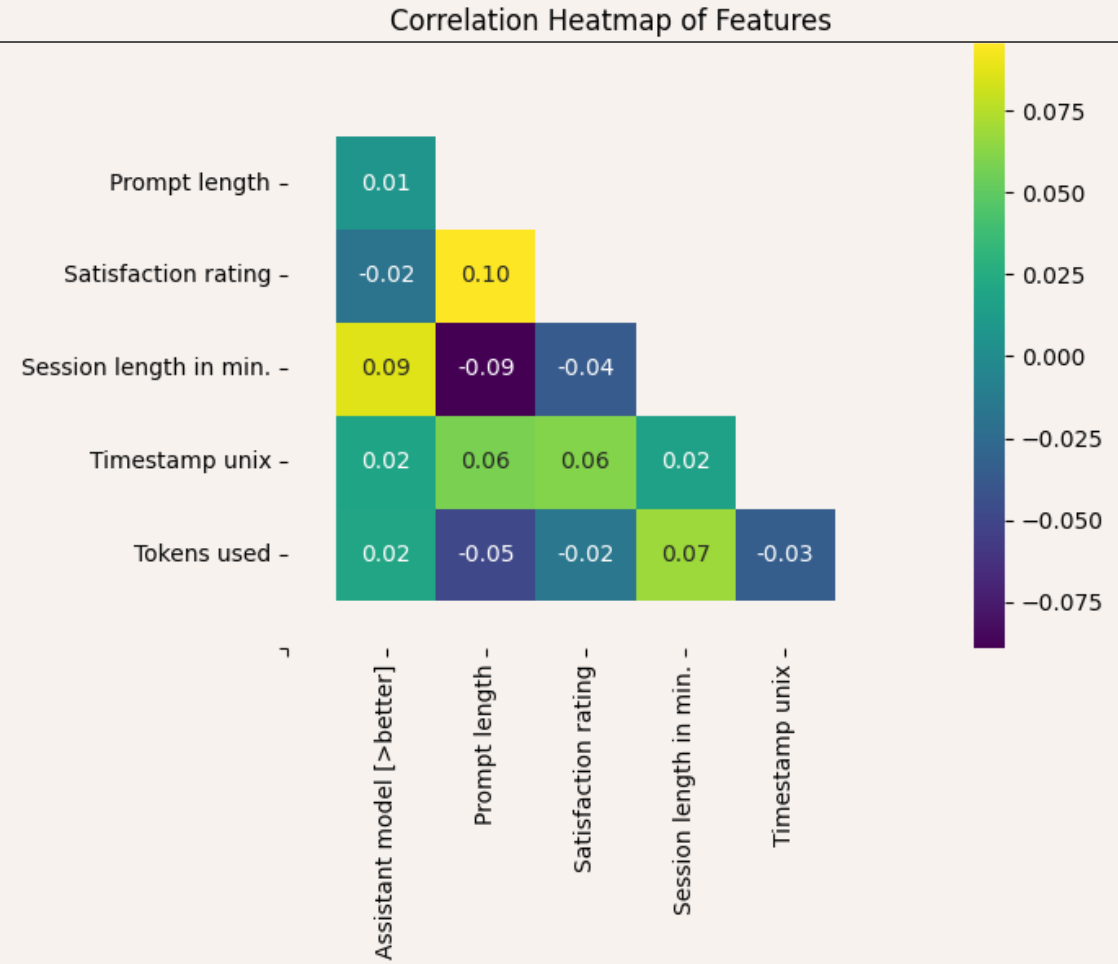

Pair Plot of Features by Assistant Model
Prompt length
Session length in min.
Timestamp unix
Tokens used
assistant_model
GPT-4o
GPT-5.1

Pair Plots

.....

There are no clear Groupings.

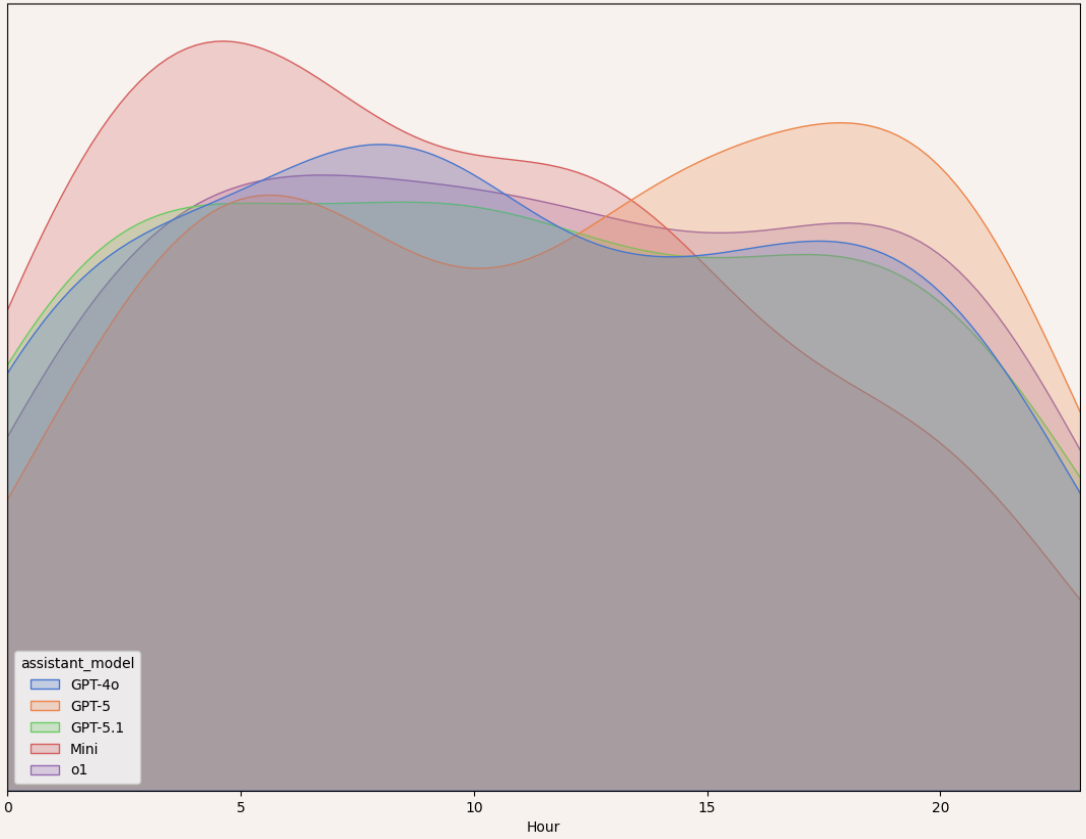
Correlation Analysis



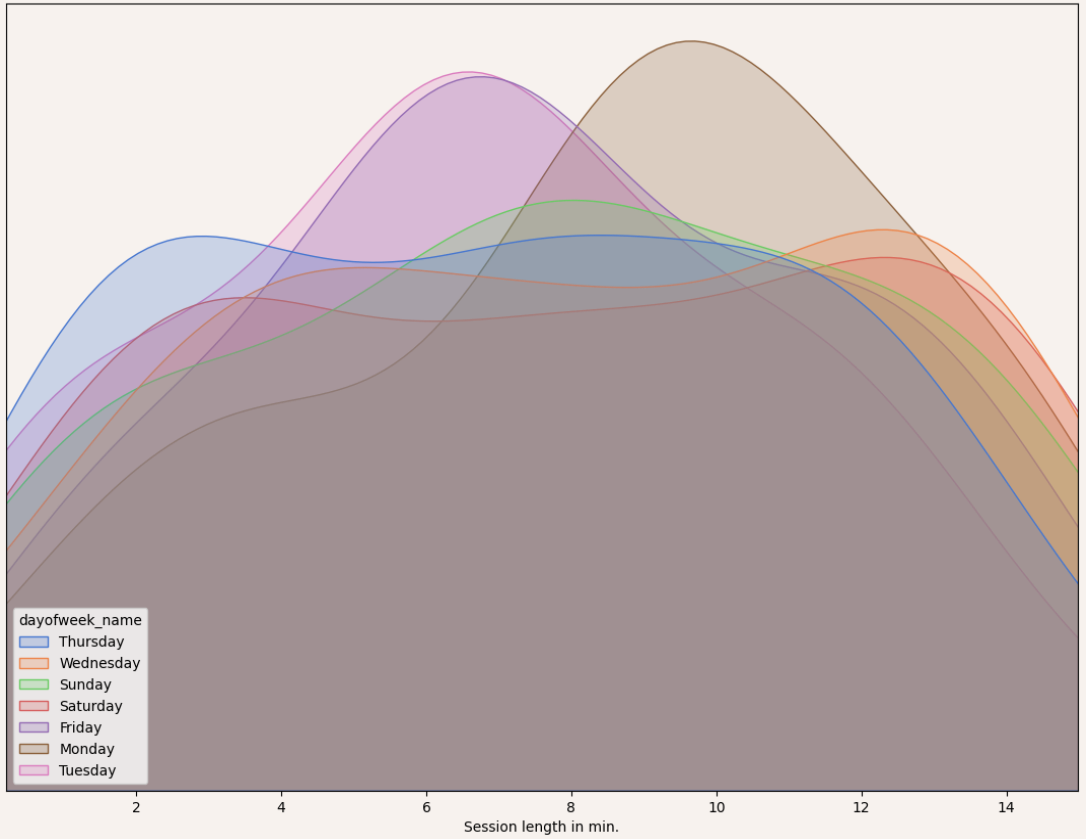
Time – Based Analysis

Time-Based Analysis via Histograms

session hour by assistant model over the day

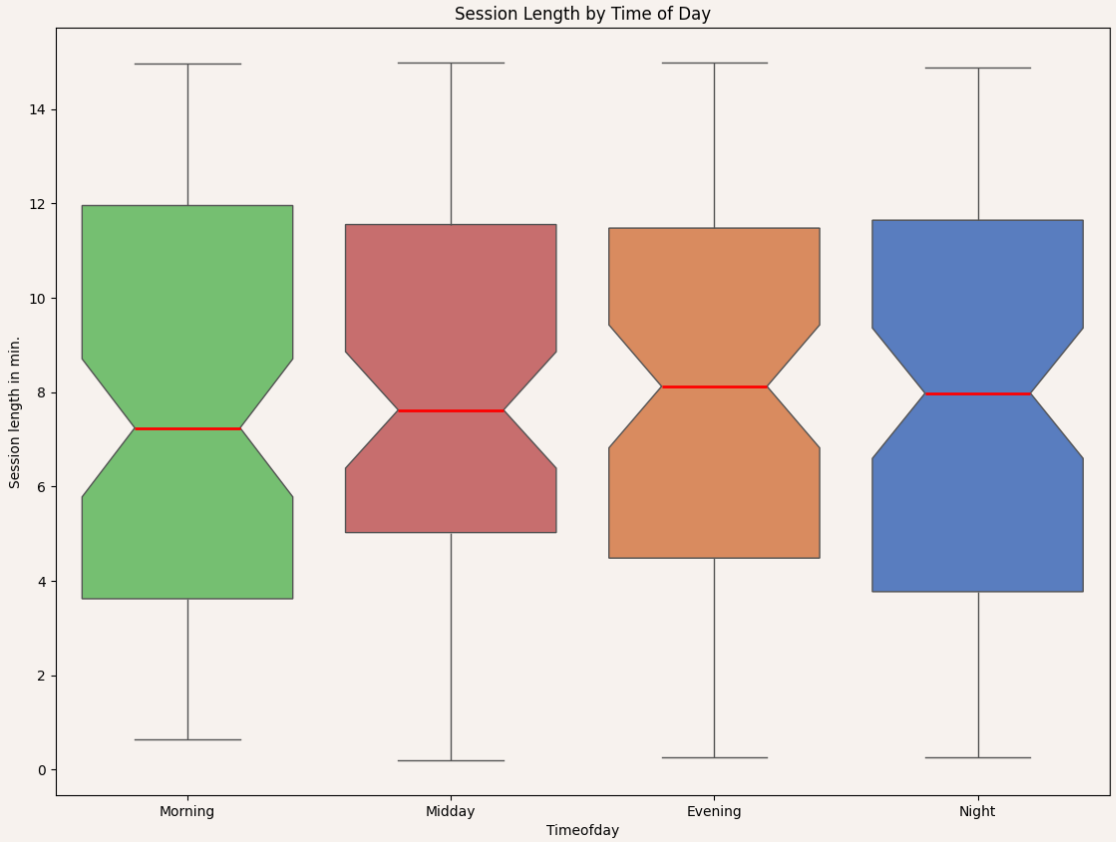


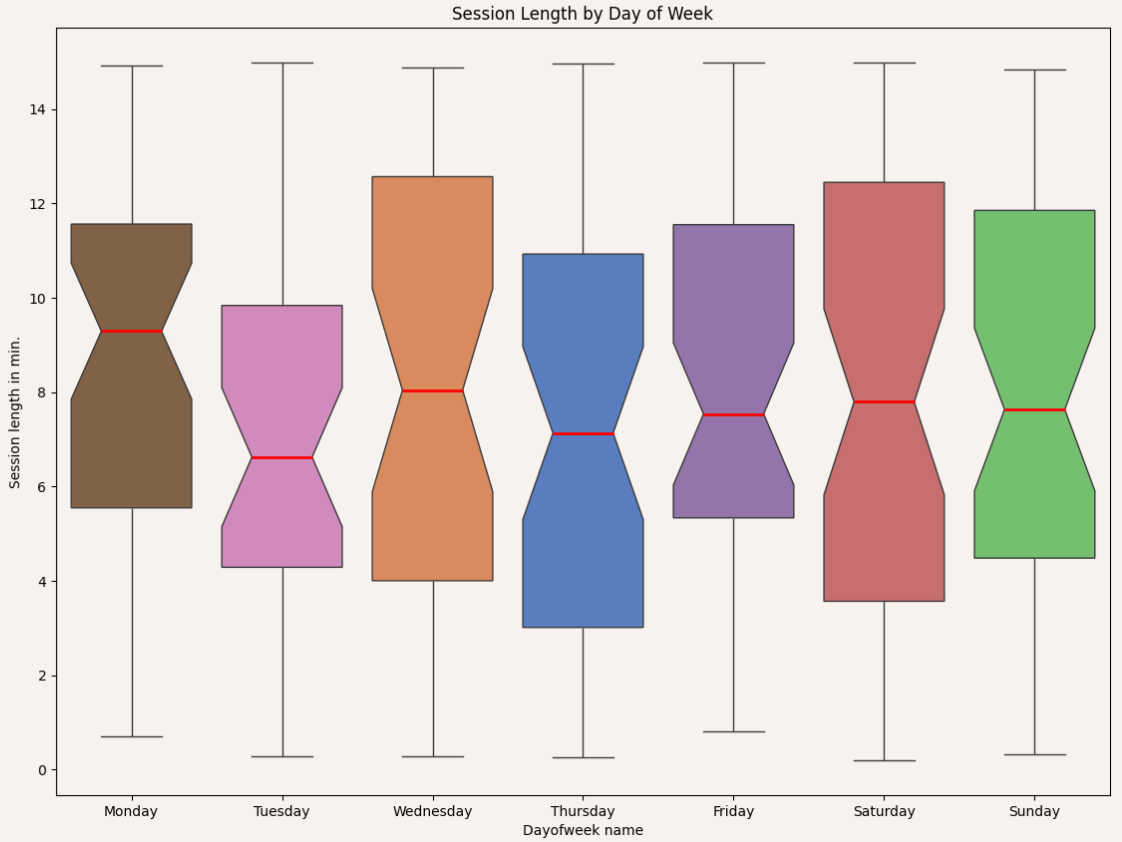
session length by day of week



Time – Based Analysis

Time-Based Analysis via Boxplots





Additional insights

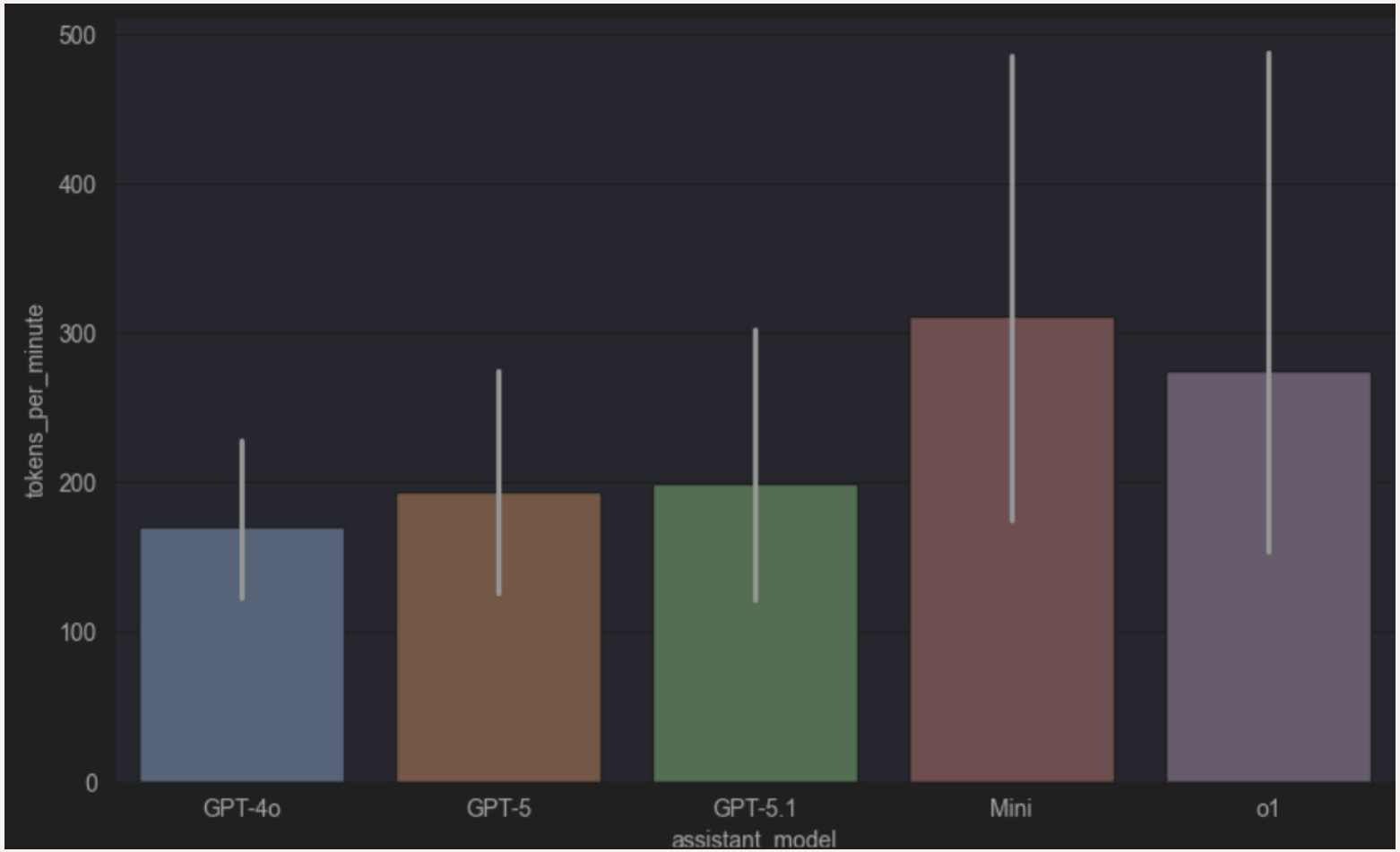
1. Analysis of the timeperiod of the database.
2. Analysis of the tokens used per minute per model
3. Analysis of the relation between Device and Satisfaction rating
4. Analysis of the correlation between weekday and time of day
5. Analysis of the Average session length per usage category.

Analysis of the time period of the database

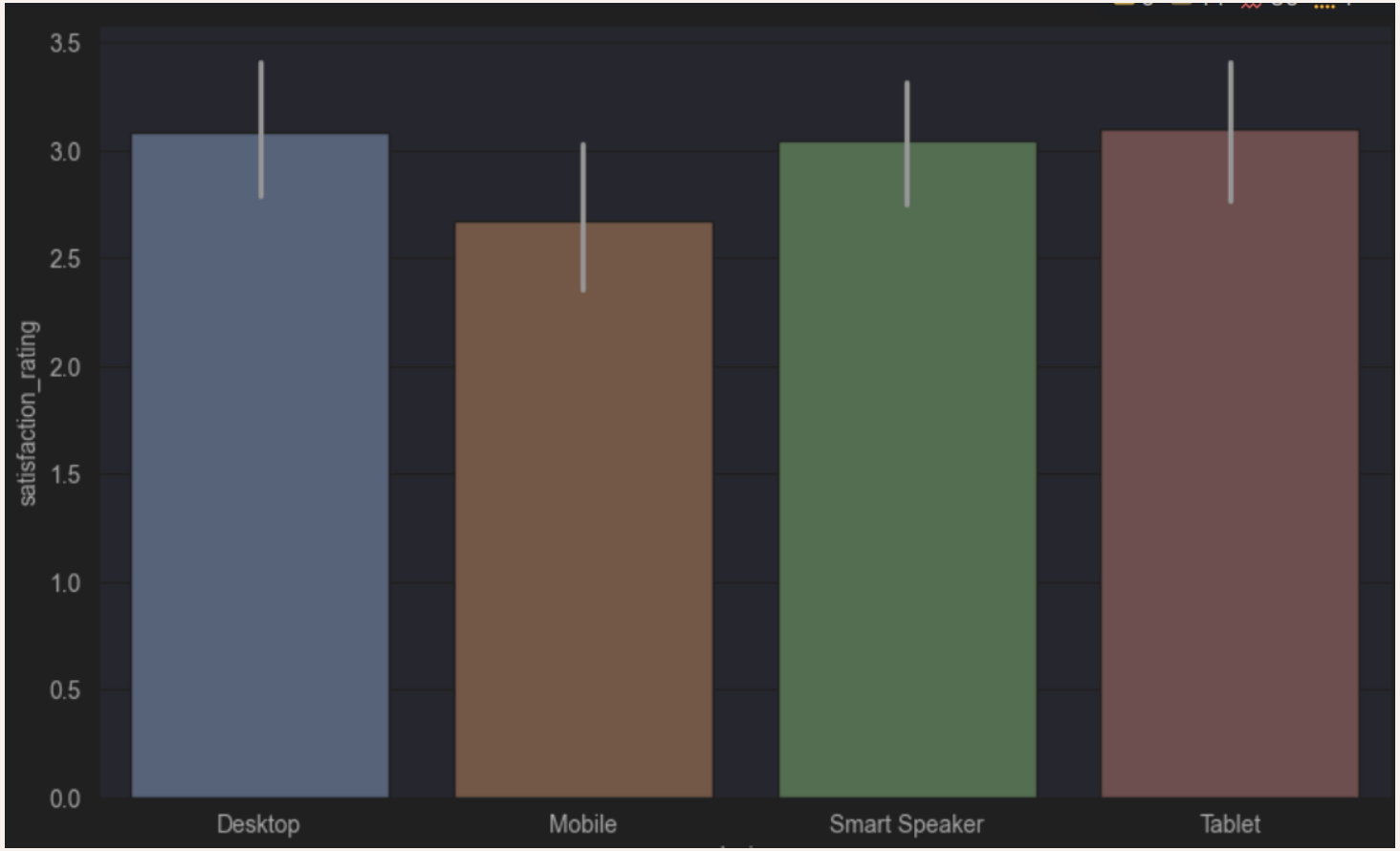
We get that the timeperiod of this database is only 69 days.

```
time_period = RAW_data['timestamp'].max() - RAW_data['timestamp'].min()
print(f"The time period of this dataset is {time_period.days} days")
```

Analysis of the tokens used per minute per model



Analysis of the relation between Device and Satisfaction rating

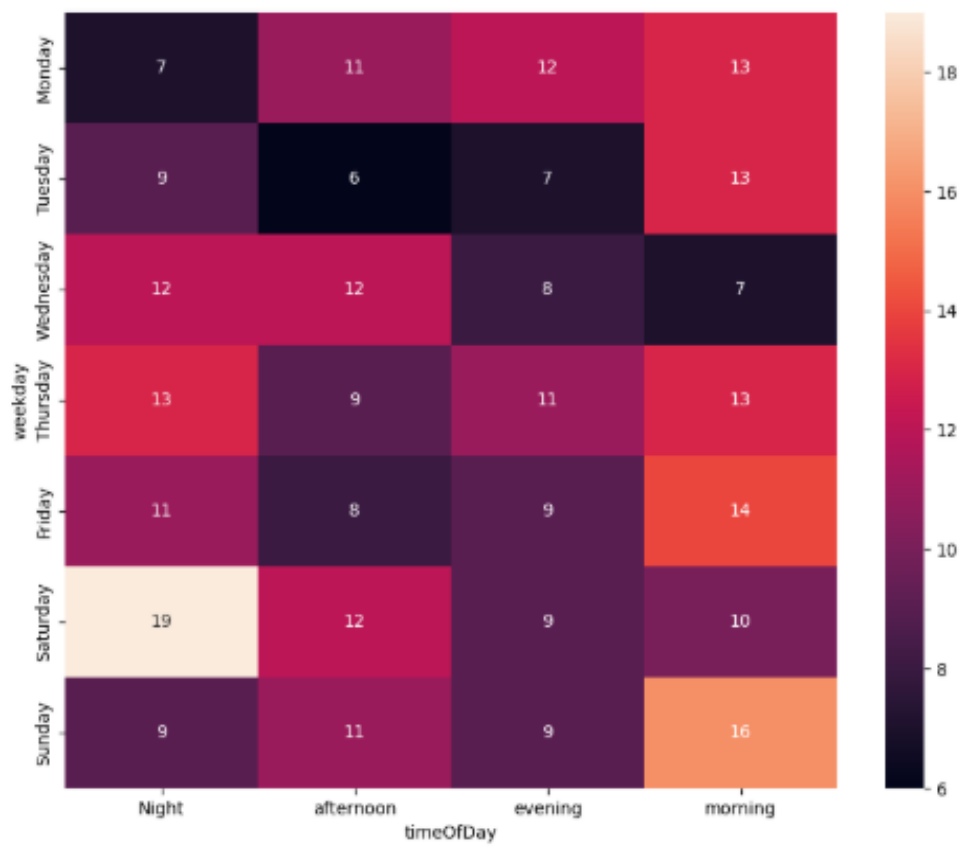


Analysis of the relation between Device and Usage category. Code

```
device_usage_category = pd.pivot_table(RAW_data, index='device', columns='usage_category', aggfunc='count
 values='timestamp')
plt.figure(figsize = (10,10))
device_usage_category.plot(kind='bar')
plt.show()
```

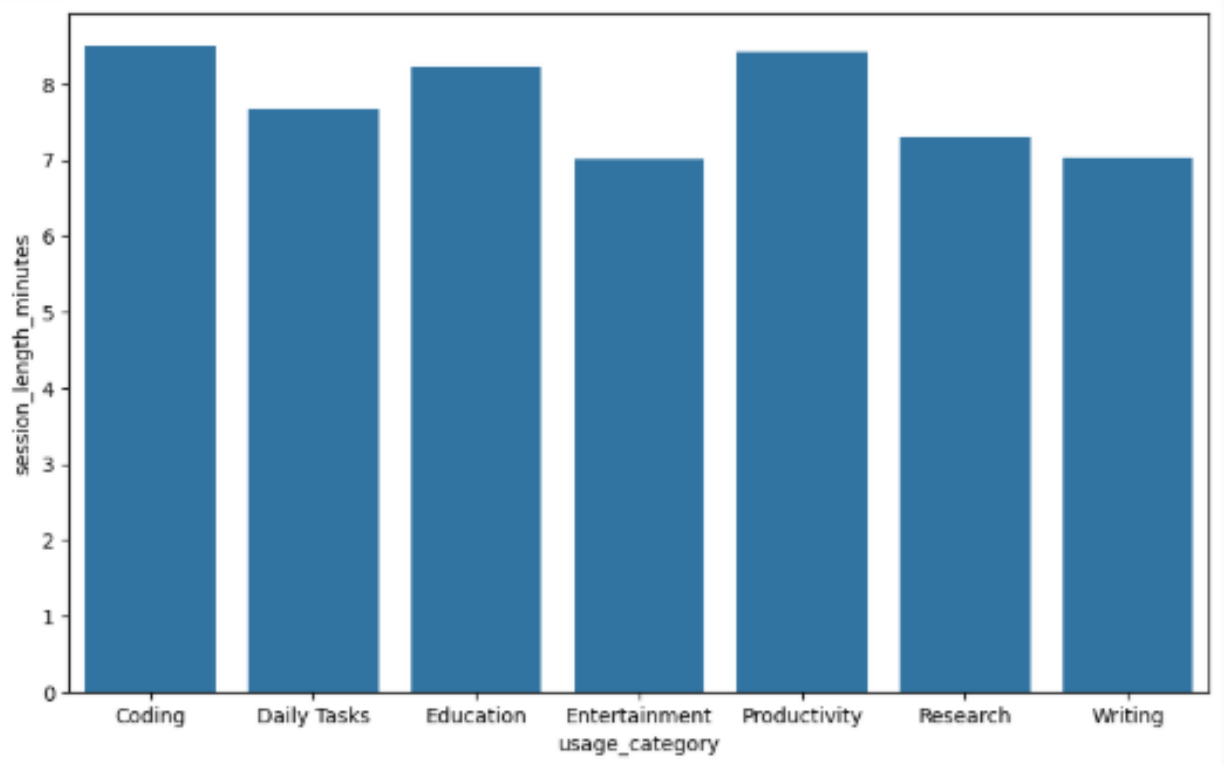
Analysis of the correlation between weekday and time of day

The heatmap shows AI assistant usage by day of the week and time of day. The number of sessions is the highest on Saturday night (19 sessions) and Sunday morning (16 sessions), while weekday afternoons, especially Tuesday and Wednesday, are less active.



Analysis of the Average session length per usage category

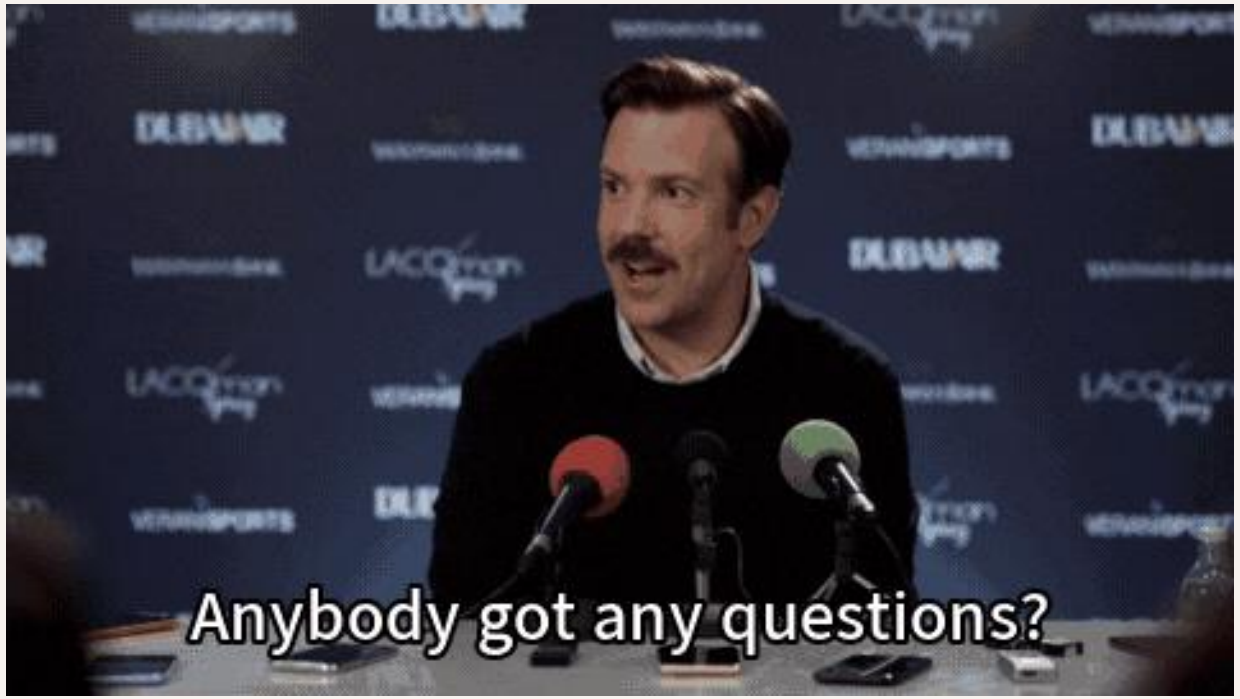
In the result we can see that coding tasks have the longest average session duration, showing that they are more time-consuming and require longer interactions with AI assistants.



Summary

- Number of Tasks completed: (460 Lines of Code/Comments)
  - Total of 27 tasks completed (22 mandatory and 5 additional tasks), each presented by the person responsible.
- Challenges:
  - The dataset is small, artificially created and hard to draw conclusions from.
- Surprising findings:
  - The use of smart speakers, especially for tasks like coding.
  - The dataset was synthetically generated. (some datasets are not marked as synthetic)
- What we learned:
  - We learned how to use GitHub repositories.
  - Read the Kaggle page first (Fully, not just skimming)

Q&A



E n D