line plot with multifacets

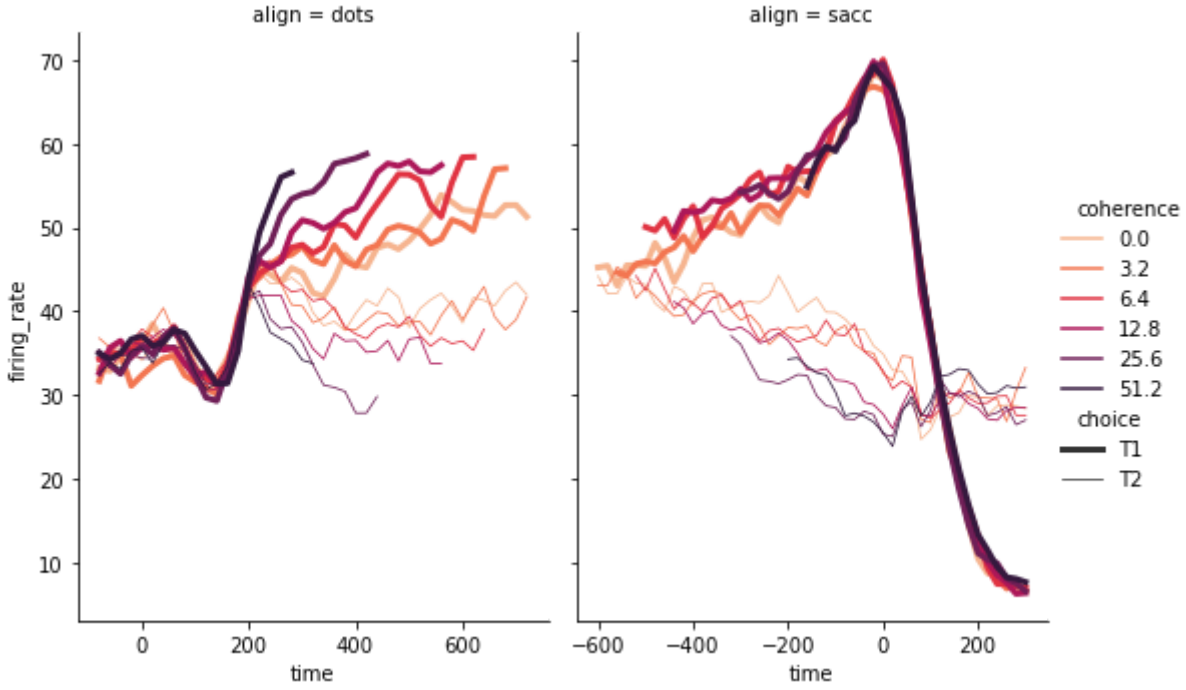
In [1]:
```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
nuqta =sns.load_dataset("dots")
nuqta.head()
```

Out[1]:

	align	choice	time	coherence	firing_rate
0	dots	T1	-80	0.0	33.189967
1	dots	T1	-80	3.2	31.691726
2	dots	T1	-80	6.4	34.279840
3	dots	T1	-80	12.8	32.631874
4	dots	T1	-80	25.6	35.060487

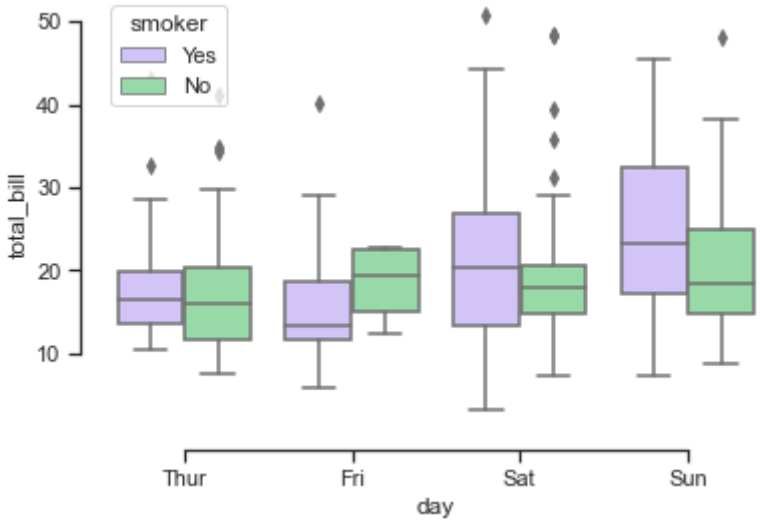
In [2]:
```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

kashti = sns.load_dataset("titanic")

#defining a color pelette
p = sns.color_palette('rocket_r')

#plot liner plot

sns.relplot(
    data=nuqta,
            x="time", y="firing_rate",
            hue="coherence",size="choice", col="align",
            kind="line", size_order=["T1","T2"],palette=p,
            height=5, aspect=.75,facet_kws=dict(sharex=False),
)
```

Out[2]:
```
<seaborn.axisgrid.FacetGrid at 0x1e77cc911c0>
```



In [3]:

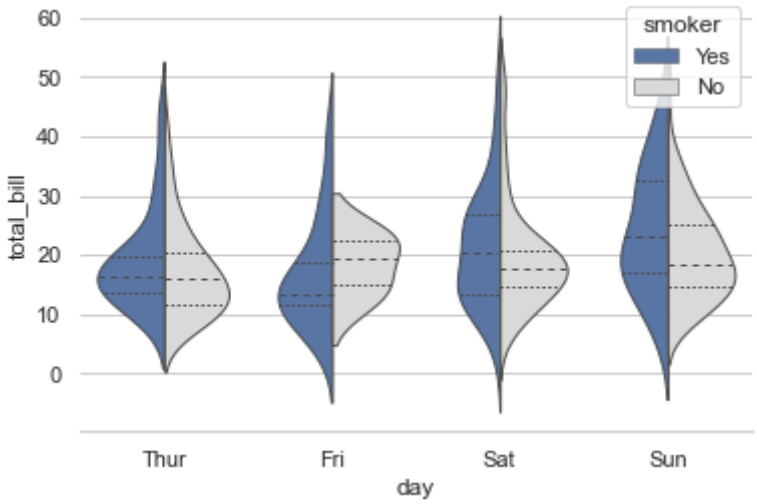
```
import seaborn as sns
sns.set_theme(style="ticks", palette="pastel")
#load the example tips dataset
tips = sns.load_dataset("tips")

#draw as nested boxplot to show biils by day and time
sns.boxplot(
            x="day", y="total_bill",
            hue="smoker",palette=["m", "g"],
            data=tips)
sns.despine(offset=10 ,trim=True)
```



In [6]:

```
import seaborn as sns
sns.set_theme(style="whitegrid")

#load the example tips dataset
tips = sns.load_dataset("tips")

#draw as nested violineplot and split the violins for easier comparsion
sns.violinplot(data=tips, x="day", y="total_bill", hue="smoker",
               split=True, inner="quart", linewidth=1,
```

```
                palette={"Yes":"b","No": ".85"})
sns.despine(left=True)
```

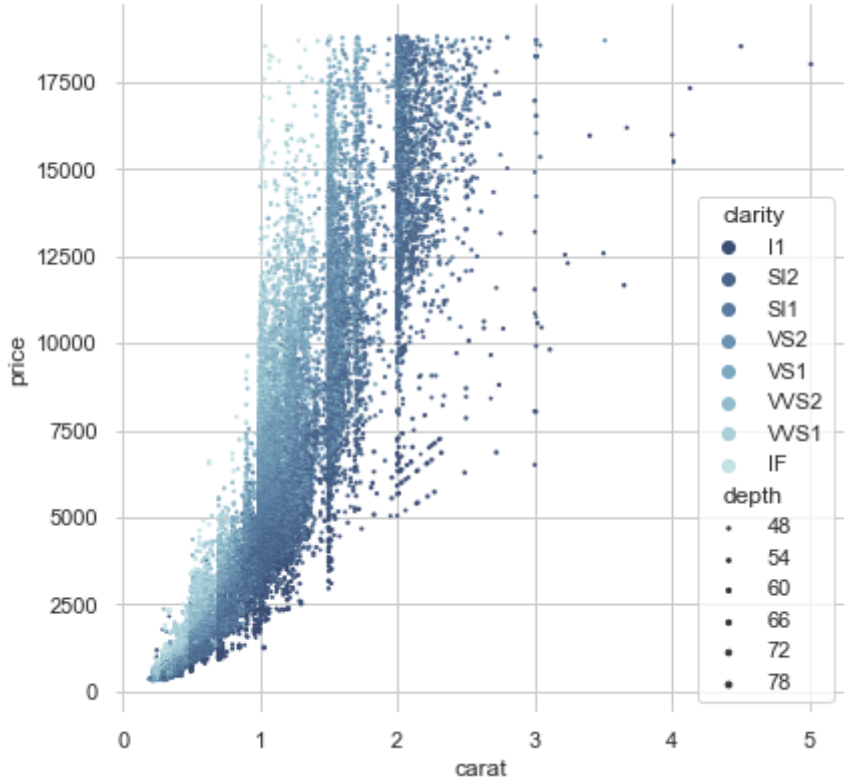


In [11]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.set_theme(style="whitegrid")

#load the example tips dataset
diamonds = sns.load_dataset("diamonds")

#draw a scatter plot white assigning point color and sizes to different
# variable in the dataset
f,ax=plt.subplots(figsize=(6.5, 6.5))
sns.despine(f,left=True, bottom=True)
clarity_ranking = ["I1","SI2","SI1","VS2","VS1","VVS2","VVS1","IF"]
sns.scatterplot(x="carat", y="price",
                hue="clarity", size="depth",
                palette="ch:r=-.2,d=.3_r",
                hue_order=clarity_ranking,
                sizes=(1, 8),linewidth=0,
                data=diamonds, ax=ax)
```

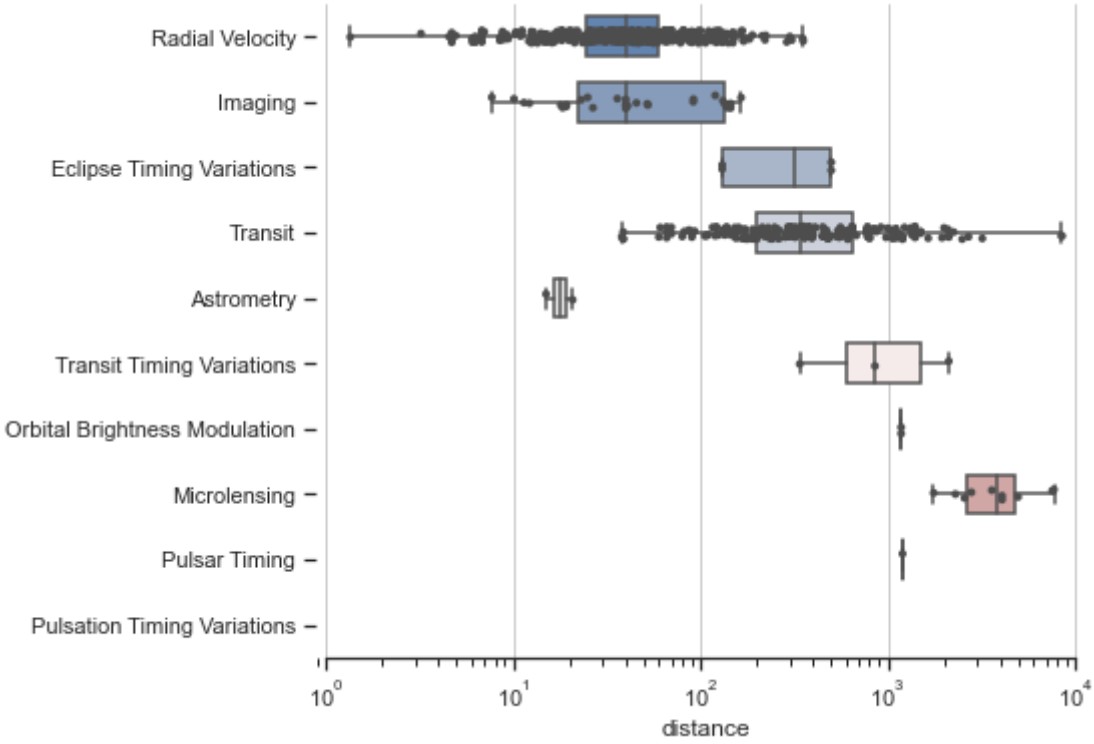
Out[11]:

```
<AxesSubplot:xlabel='carat', ylabel='price'>
```



In [14]:

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.set_theme(style="ticks")

# initialize the figure with a logarithumic x axis
f, ax = plt.subplots(figsize=(7,6))
ax.set_xscale("log")

#load the example tips dataset
planets = sns.load_dataset("planets")

#plot the orbital period with horizantal boxes
sns.boxplot(x="distance",y="method", data=planets,
            whis=[0,100],width=.6, palette="vlag")

# Add in point to show each observation
sns.stripplot(x="distance",y="method",data=planets,
              size=4, color=".3",linewidth=0)

# tweak the visual presentation
ax.xaxis.grid(True)
ax.set(ylabel="")
sns.despine(trim=True, left=True)
```



In [16]:
```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
nuqta =sns.load_dataset("dots")
nuqta.head()
```

Out[16]:

	align	choice	time	coherence	firing_rate
0	dots	T1	-80	0.0	33.189967
1	dots	T1	-80	3.2	31.691726
2	dots	T1	-80	6.4	34.279840
3	dots	T1	-80	12.8	32.631874
4	dots	T1	-80	25.6	35.060487

In [19]:
```
diamonds = sns.load_dataset("diamonds")
diamonds.head()
```

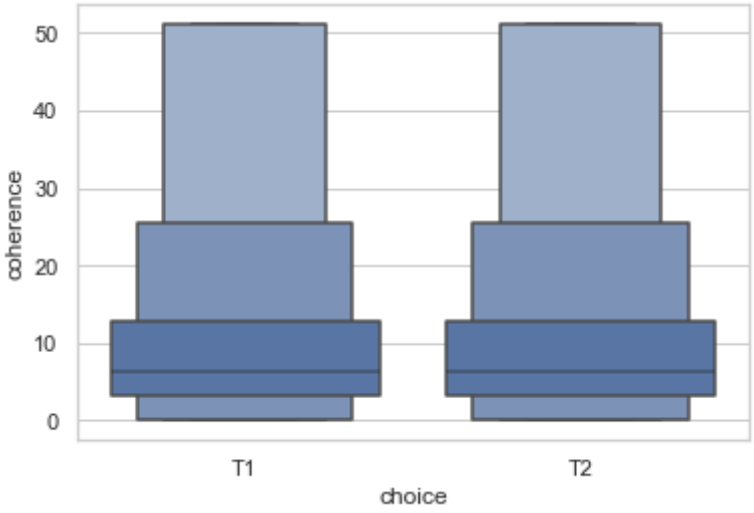
Out[19]:

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

In [21]:
```
import seaborn as sns
sns.set_theme(style="whitegrid")
```

```
diamonds = sns.load_dataset("diamonds")
#clarity_ranking = ["I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF"]

sns.boxenplot(x="choice", y="coherence",
              color="b",
              scale="linear", data=nuqta)
```

Out[21]: <AxesSubplot:xlabel='choice', ylabel='coherence'>



In [4]:

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9104/736536530.py in <module>
      2 import matplotlib.pyplot as plt
      3
----> 4 import plotly.graph_objects as sns
      5
      6 import pandas as pd

ModuleNotFoundError: No module named 'plotly'
```

In [ ]: