```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
pip install statsmodels
```

```
Requirement already satisfied: statsmodels in /usr/local/lib/python3.10/dist-packages (0.14.1)
Requirement already satisfied: numpy<2,>=1.18 in /usr/local/lib/python3.10/dist-packages (from statsmodels) (1.23.5)
Requirement already satisfied: scipy!=1.9.2,>=1.4 in /usr/local/lib/python3.10/dist-packages (from statsmodels) (1.11.4)
Requirement already satisfied: pandas!=2.1.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from statsmodels) (1.5.3)
Requirement already satisfied: patsy>=0.5.4 in /usr/local/lib/python3.10/dist-packages (from statsmodels) (0.5.6)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from statsmodels) (23.2)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas!=2.1.0,>=1.0->statsmodels)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas!=2.1.0,>=1.0->statsmodels) (2023.3.p
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from patsy>=0.5.4->statsmodels) (1.16.0)
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

```
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/1.01. Simple linear regression.csv")
```

```
data
```

	SAT	GPA
0	1714	2.40
1	1664	2.52
2	1760	2.54
3	1685	2.74
4	1693	2.83
...	...	...
79	1936	3.71
80	1810	3.71
81	1987	3.73
82	1962	3.76
83	2050	3.81

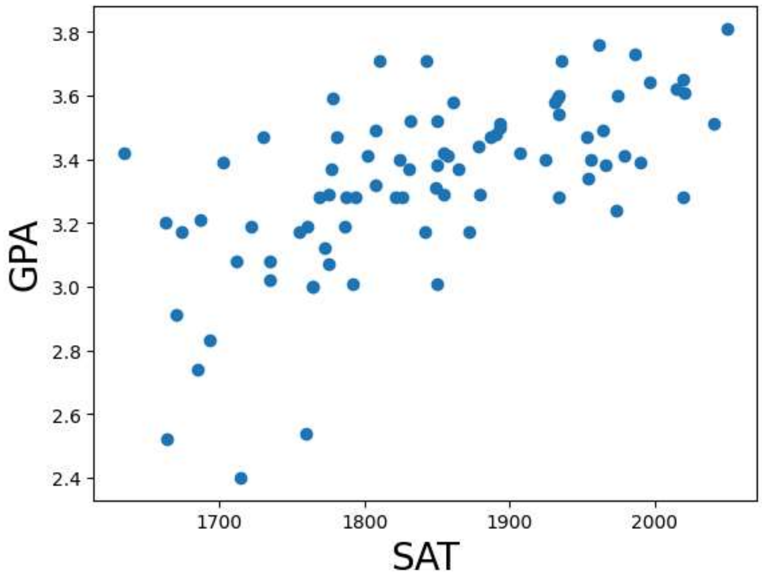
84 rows × 2 columns

```
data.describe()
```

	SAT	GPA
count	84.000000	84.000000
mean	1845.273810	3.330238
std	104.530661	0.271617
min	1634.000000	2.400000
25%	1772.000000	3.190000
50%	1846.000000	3.380000
75%	1934.000000	3.502500
max	2050.000000	3.810000

```
y = data['GPA']
x1 = data['SAT']
```

```
plt.scatter(x1,y)
plt.xlabel('SAT', fontsize = 20)
plt.ylabel('GPA', fontsize = 20)
plt.show()
```

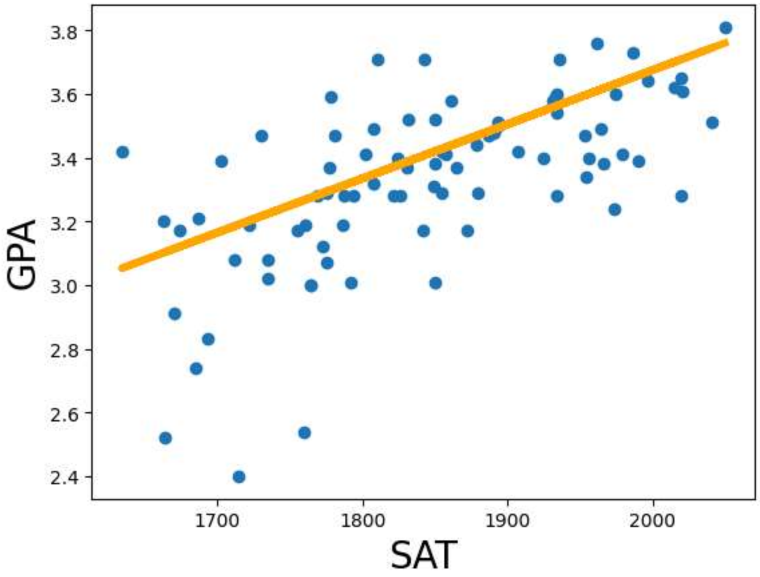


```
x = sm.add_constant(x1)
results = sm.OLS(y,x).fit()
```

```
results.summary()
```

```
                    OLS Regression Results
      Dep. Variable:  GPA                      R-squared:  0.406
             Model:   OLS                 Adj. R-squared:  0.399
            Method:   Least Squares          F-statistic:  56.05
              Date:   Sat, 20 Jan 2024 Prob (F-statistic): 7.20e-11
              Time:   03:50:39              Log-Likelihood: 12.672
  No. Observations:   84                             AIC:  -21.34
       Df Residuals:  82                             BIC:  -16.48
          Df Model:   1
   Covariance Type:   nonrobust
          coef   std err   t    P>|t|  [0.025 0.975]
  const  0.2750  0.409   0.673  0.503  -0.538  1.088
   SAT   0.0017  0.000   7.487  0.000  0.001   0.002
      Omnibus:       12.839  Durbin-Watson:     0.950
  Prob(Omnibus):     0.002   Jarque-Bera (JB):  16.155
         Skew:       -0.722       Prob(JB):     0.000310
      Kurtosis:      4.590       Cond. No.      3.29e+04


  Notes:
  [1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
  [2] The condition number is large, 3.29e+04. This might indicate that there are
  strong multicollinearity or other numerical problems.
```

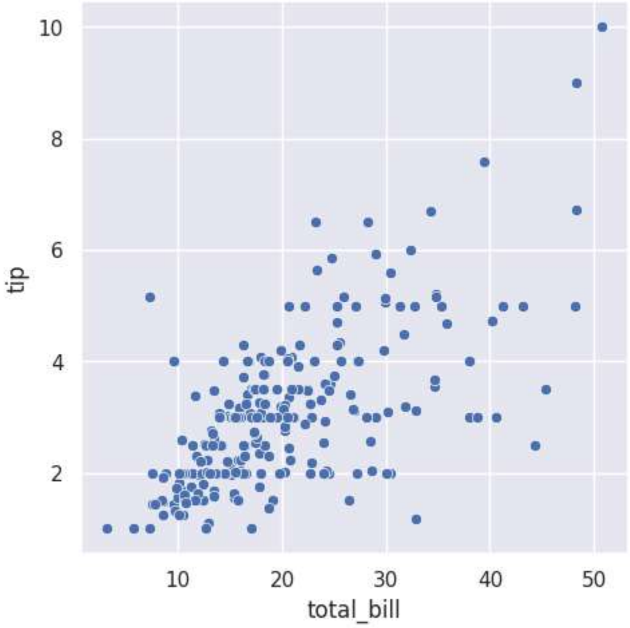
```
plt.scatter(x1,y)
yhat = 0.0017*x1 + 0.275
x2 = np.array(x1)
y2 = np.array(yhat)
fig = plt.plot(x2,y2,lw = 4, c = 'orange', label = 'regression line')
plt.xlabel('SAT', fontsize = 20)
plt.ylabel('GPA', fontsize = 20)
plt.show()
```



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_theme(style="darkgrid")
```

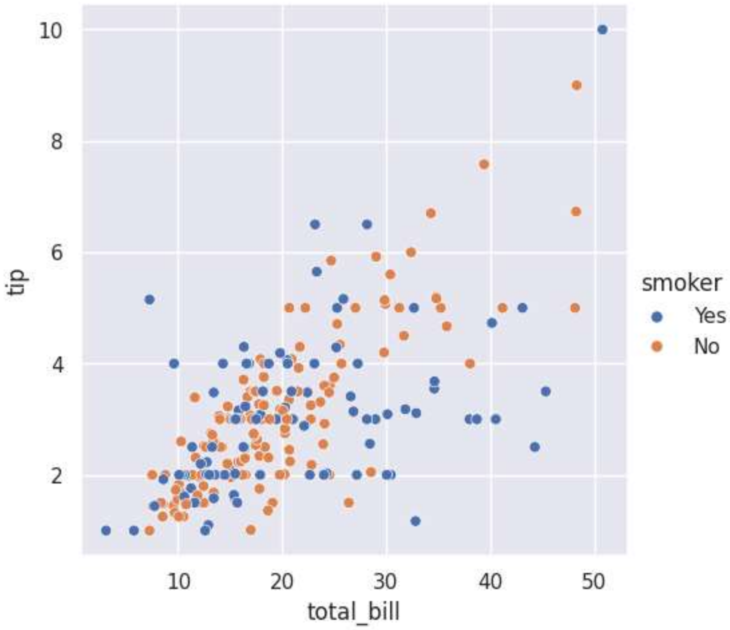
```
tips = sns.load_dataset("tips")
sns.relplot(data=tips, x="total_bill", y="tip")
```

```
<seaborn.axisgrid.FacetGrid at 0x7992c2701db0>
```



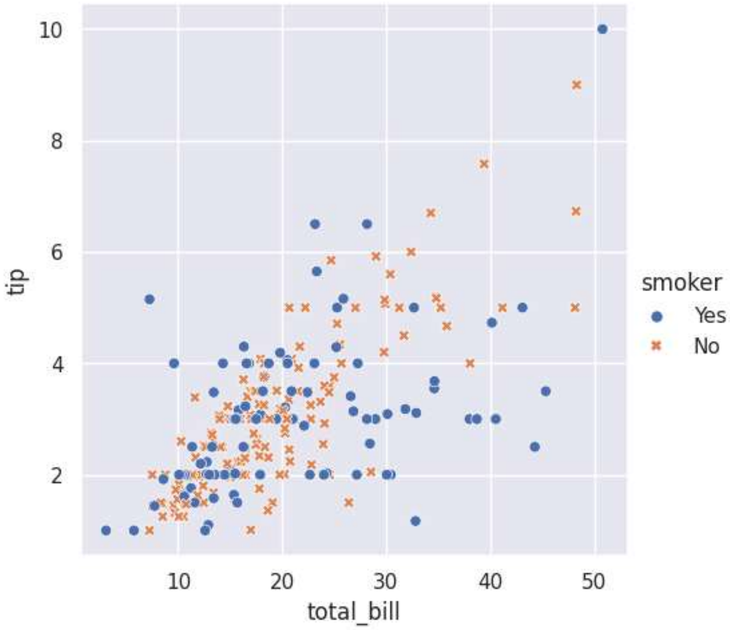
```
sns.relplot(data=tips, x="total_bill", y="tip", hue="smoker")
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bfe91510>
```



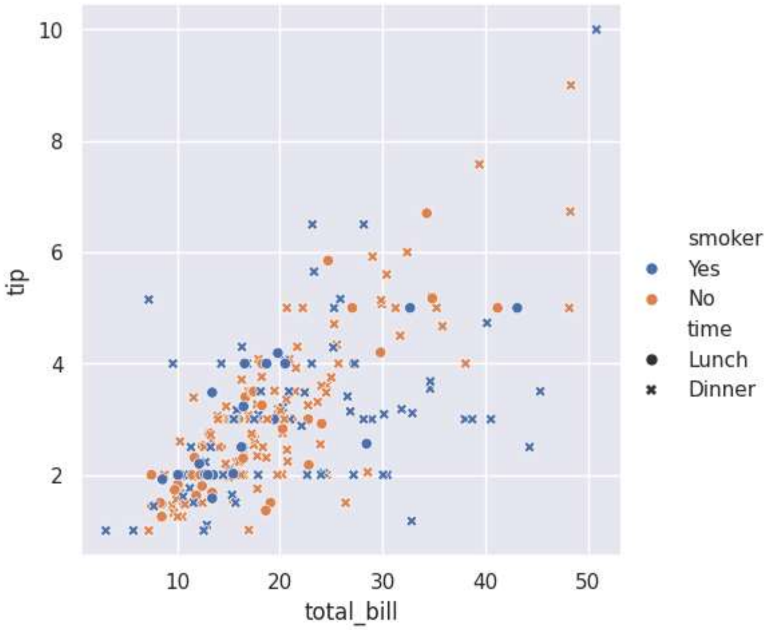
```
sns.relplot(
    data=tips,
    x="total_bill", y="tip", hue="smoker", style="smoker"
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992c2a9a500>
```



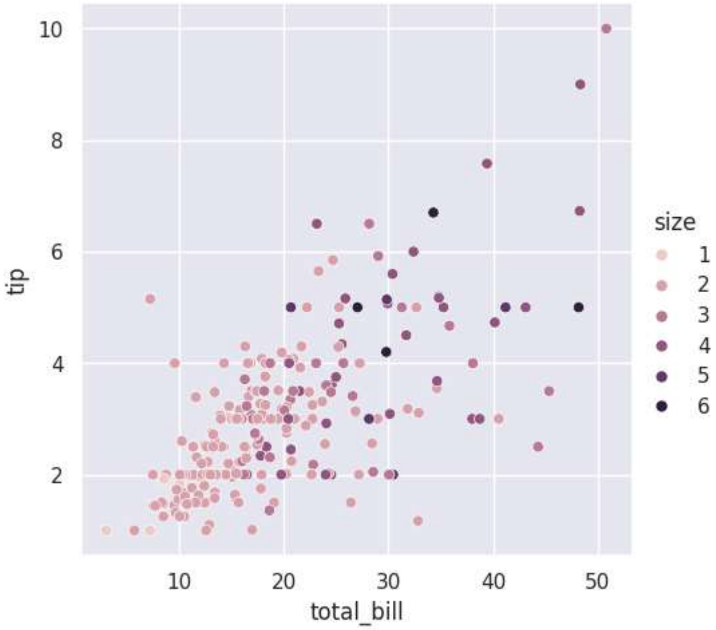
```
sns.relplot(
    data=tips,
    x="total_bill", y="tip", hue="smoker", style="time",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992c01675b0>
```



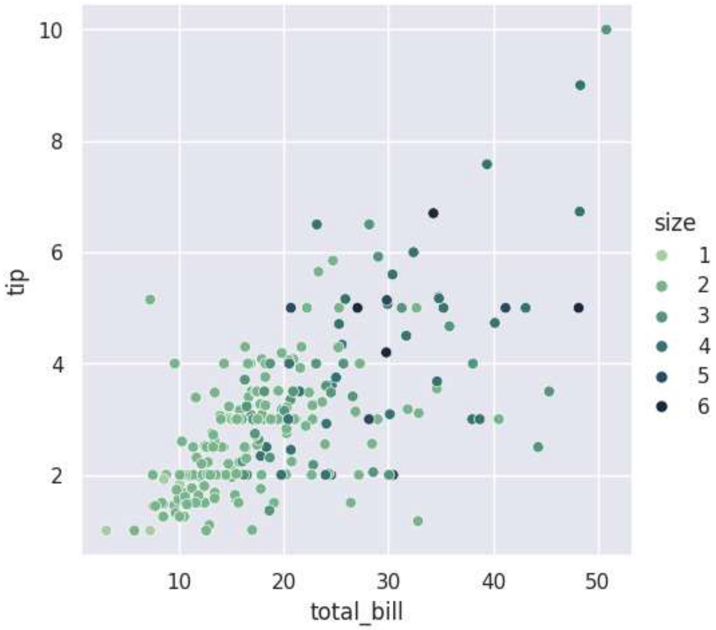
```
sns.relplot(
    data=tips, x="total_bill", y="tip", hue="size",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bff55a20>
```



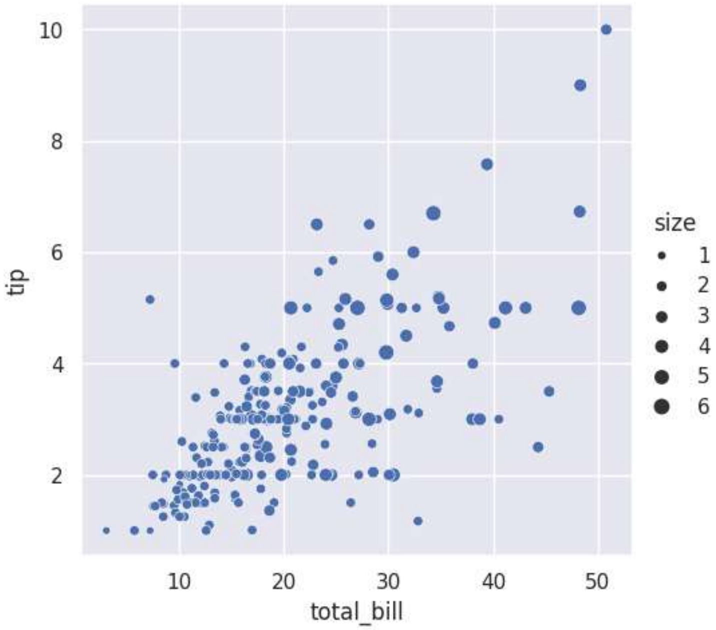
```
sns.relplot(
    data=tips,
    x="total_bill", y="tip",
    hue="size", palette="ch:r=-.5,l=.75"
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bff55300>
```



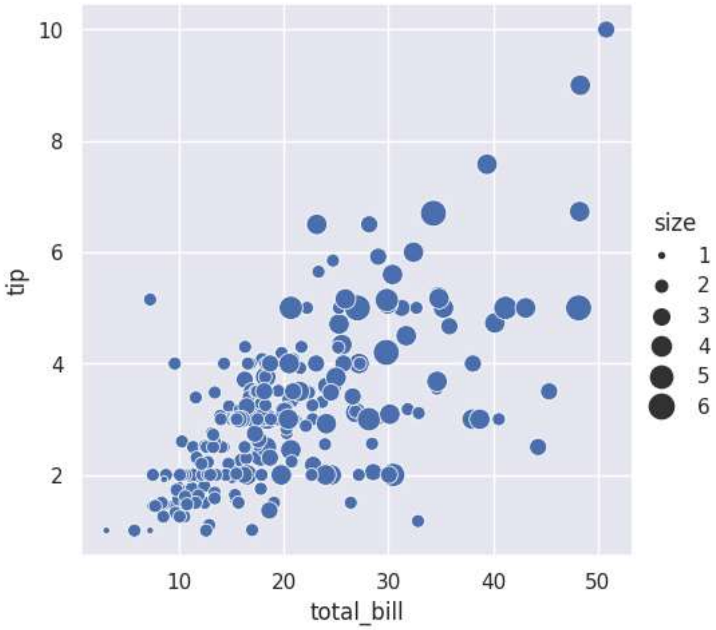
```
sns.relplot(data=tips, x="total_bill", y="tip", size="size")
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bfe910c0>
```



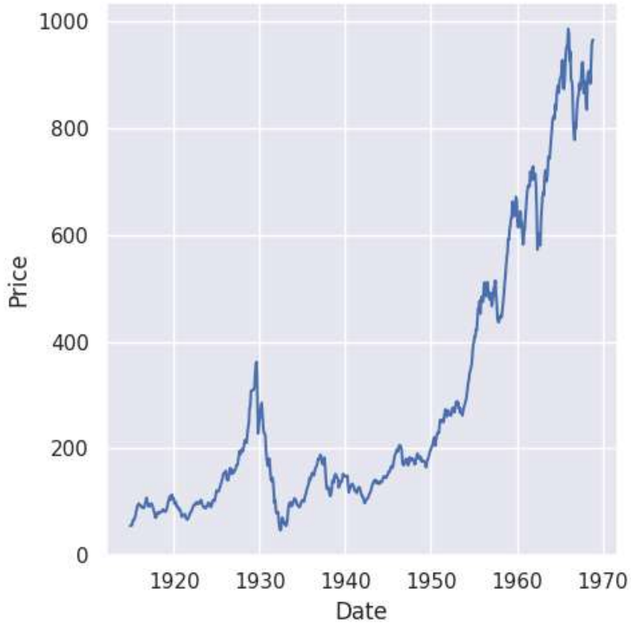
```
sns.relplot(
    data=tips, x="total_bill", y="tip",
    size="size", sizes=(15, 200)
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be45f3a0>
```



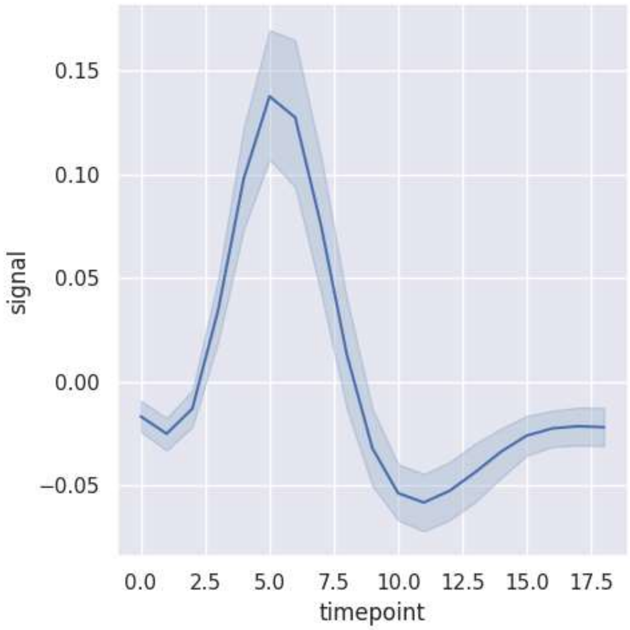
```
dowjones = sns.load_dataset("dowjones")
sns.relplot(data=dowjones, x="Date", y="Price", kind="line")
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be30e3b0>
```



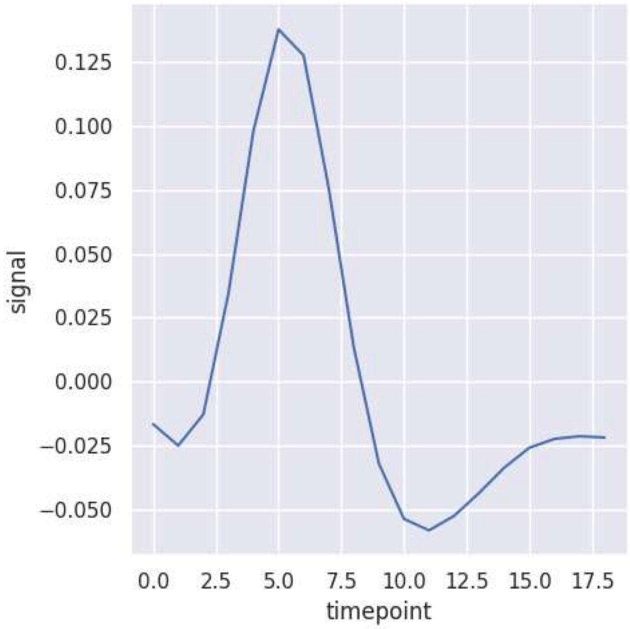
```
fmri = sns.load_dataset("fmri")
sns.relplot(data=fmri, x="timepoint", y="signal", kind="line")
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be1adb40>
```



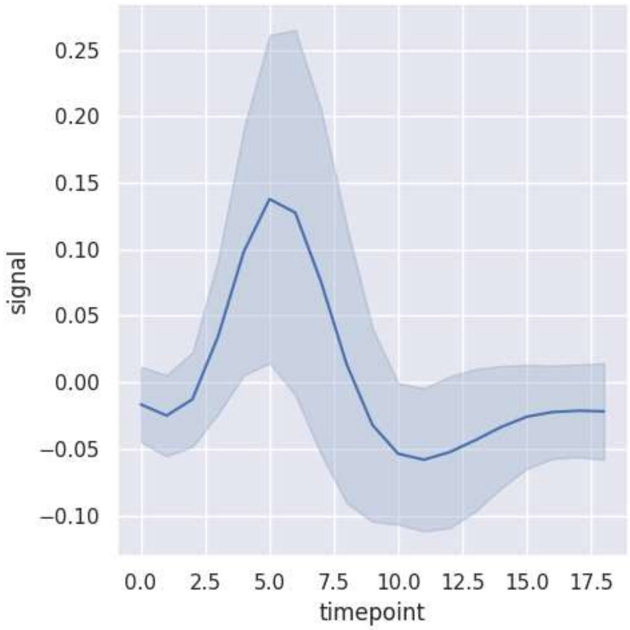
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal", errorbar=None,
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be2023e0>
```



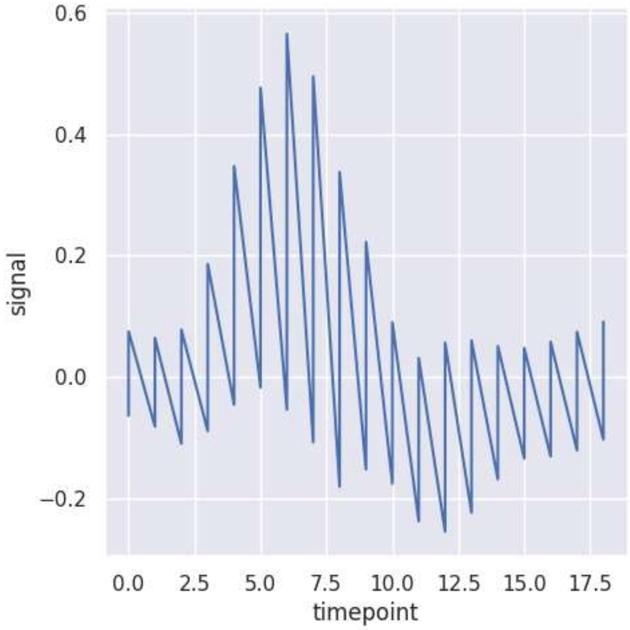
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal", errorbar="sd",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be30f9d0>
```



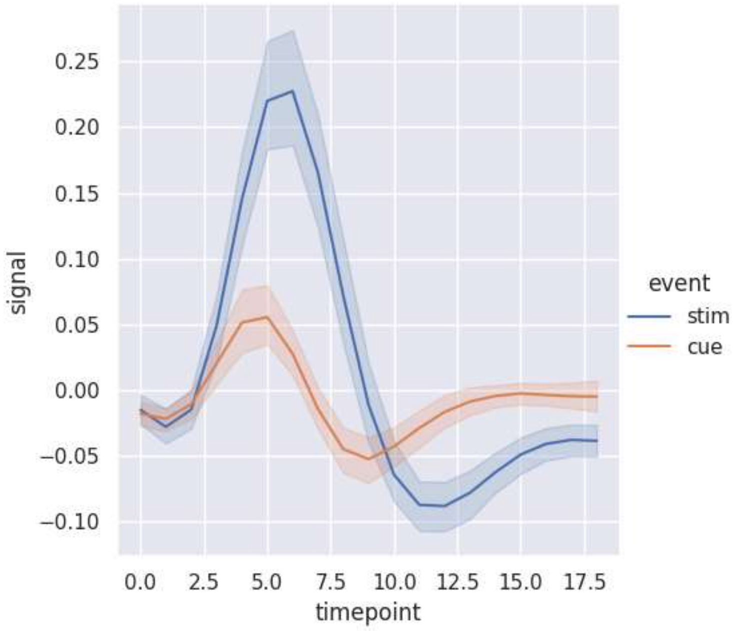
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal",
    estimator=None,
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be0ee7a0>
```



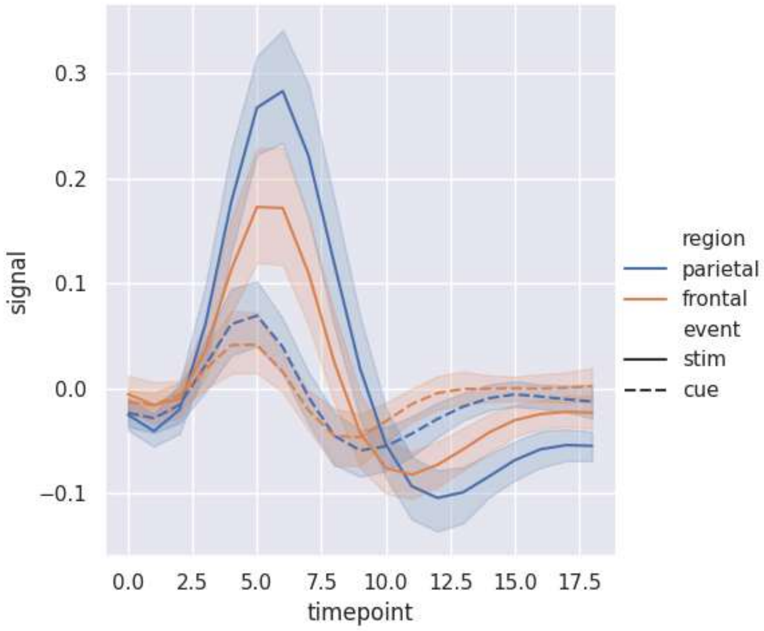
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal", hue="event",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992c00272b0>
```



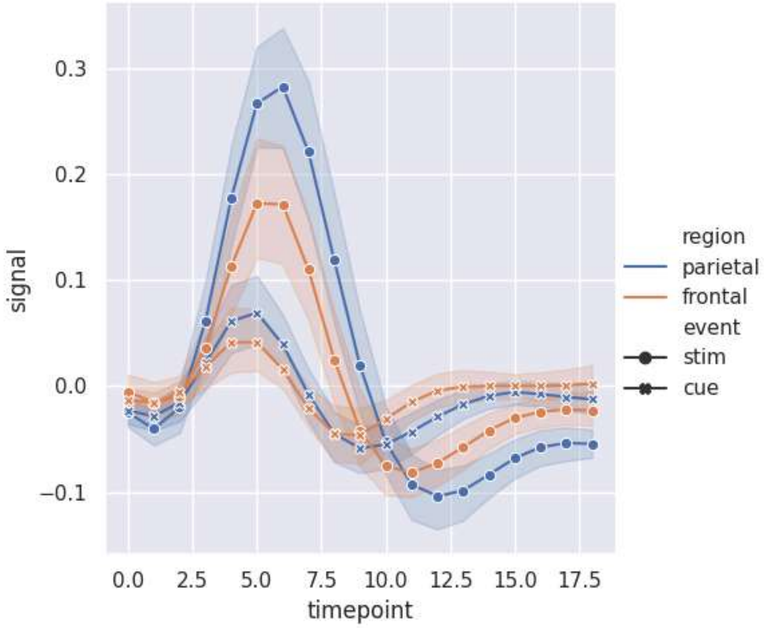
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal",
    hue="region", style="event",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be05f970>
```



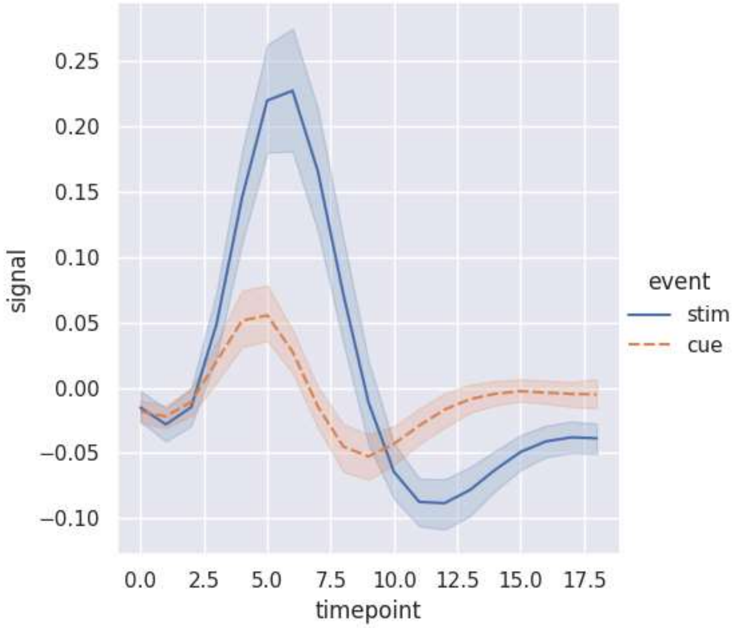
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal", hue="region", style="event",
    dashes=False, markers=True,
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be037850>
```



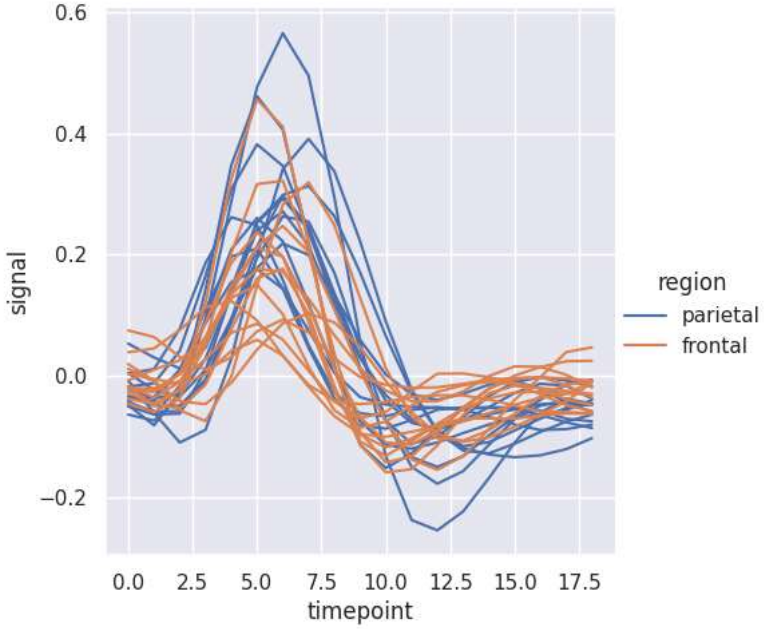
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal", hue="event", style="event",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bddaf5b0>
```



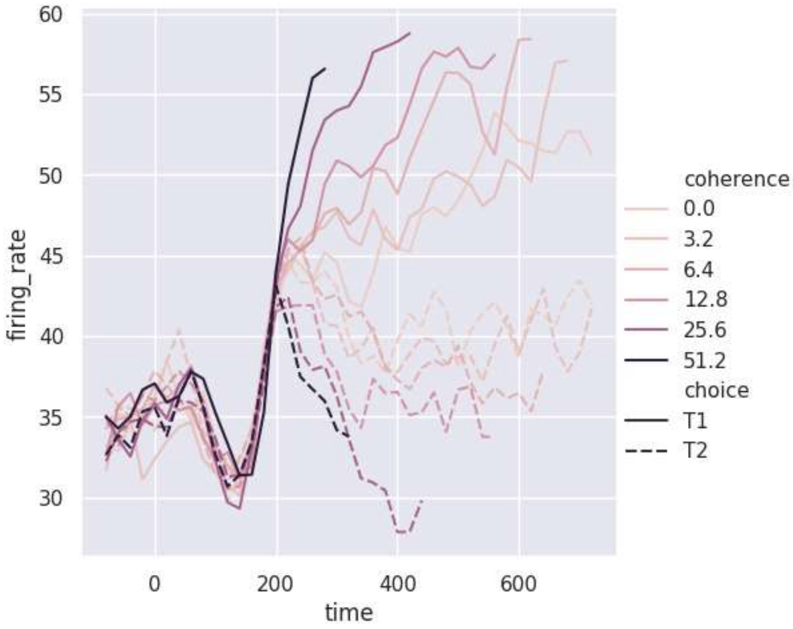
```
sns.relplot(
    data=fmri.query("event == 'stim'"), kind="line",
    x="timepoint", y="signal", hue="region",
    units="subject", estimator=None,
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bdc6cf40>
```



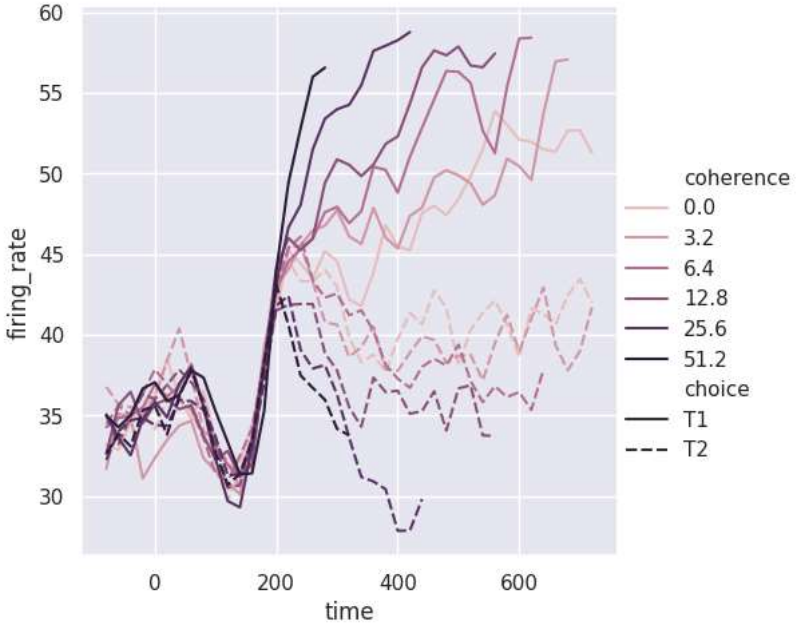
```
dots = sns.load_dataset("dots").query("align == 'dots'")
sns.relplot(
    data=dots, kind="line",
    x="time", y="firing_rate",
    hue="coherence", style="choice",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bdd32ec0>
```



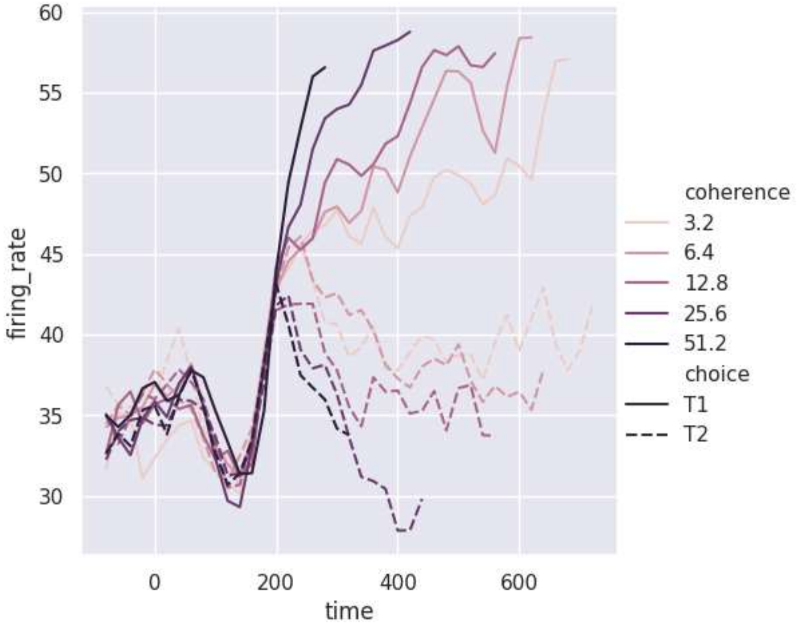
```
palette = sns.cubehelix_palette(light=.8, n_colors=6)
sns.relplot(
    data=dots, kind="line",
    x="time", y="firing_rate",
    hue="coherence", style="choice", palette=palette,
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bde03e20>
```



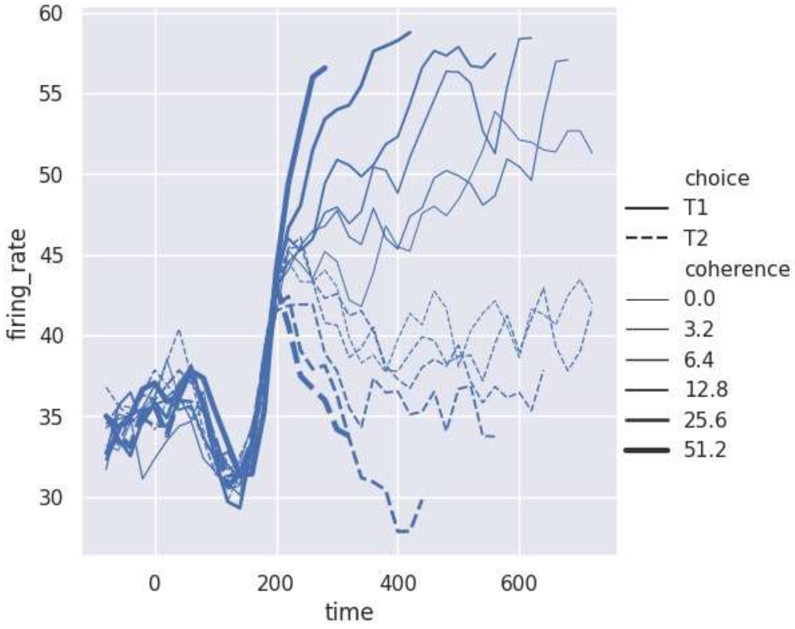
```
from matplotlib.colors import LogNorm
palette = sns.cubehelix_palette(light=.7, n_colors=6)
sns.relplot(
    data=dots.query("coherence > 0"), kind="line",
    x="time", y="firing_rate",
    hue="coherence", style="choice",
    hue_norm=LogNorm(),
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992be05f1c0>
```



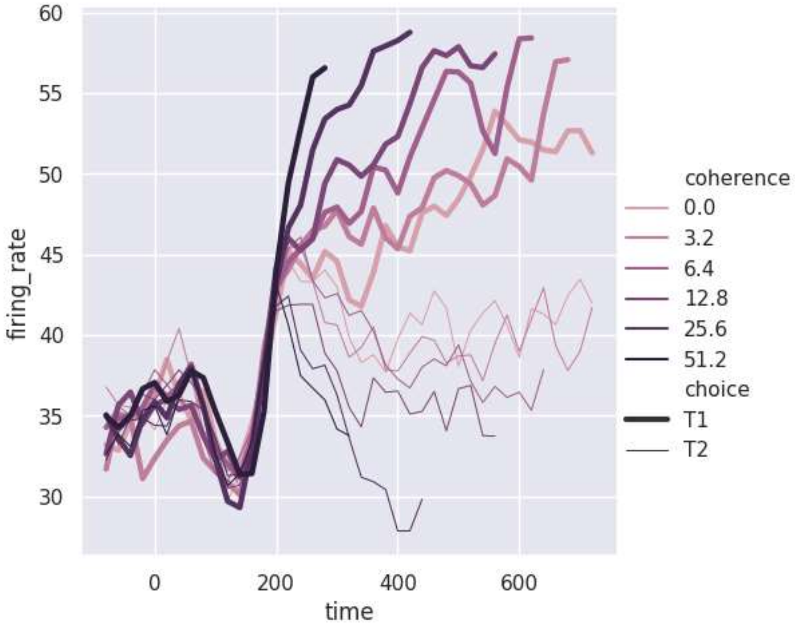
```
sns.relplot(
    data=dots, kind="line",
    x="time", y="firing_rate",
    size="coherence", style="choice",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bd9ea350>
```



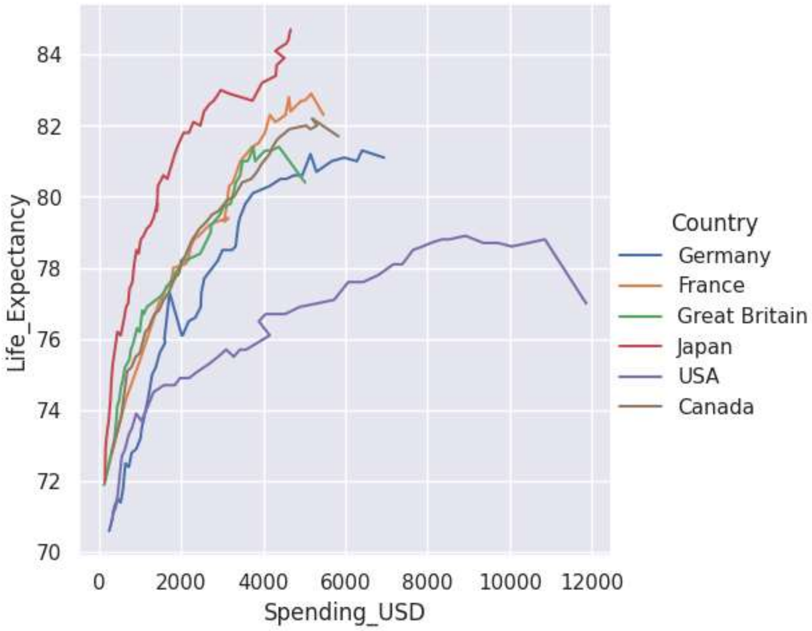
```
sns.relplot(
    data=dots, kind="line",
    x="time", y="firing_rate",
    hue="coherence", size="choice", palette=palette,
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bd7bb580>
```



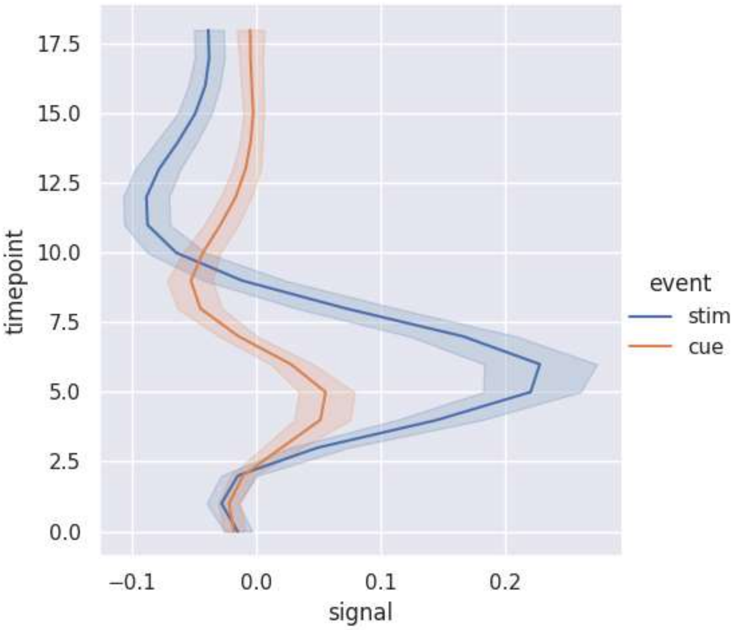
```
healthexp = sns.load_dataset("healthexp").sort_values("Year")
sns.relplot(
    data=healthexp, kind="line",
    x="Spending_USD", y="Life_Expectancy", hue="Country",
    sort=False
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bd69f700>
```



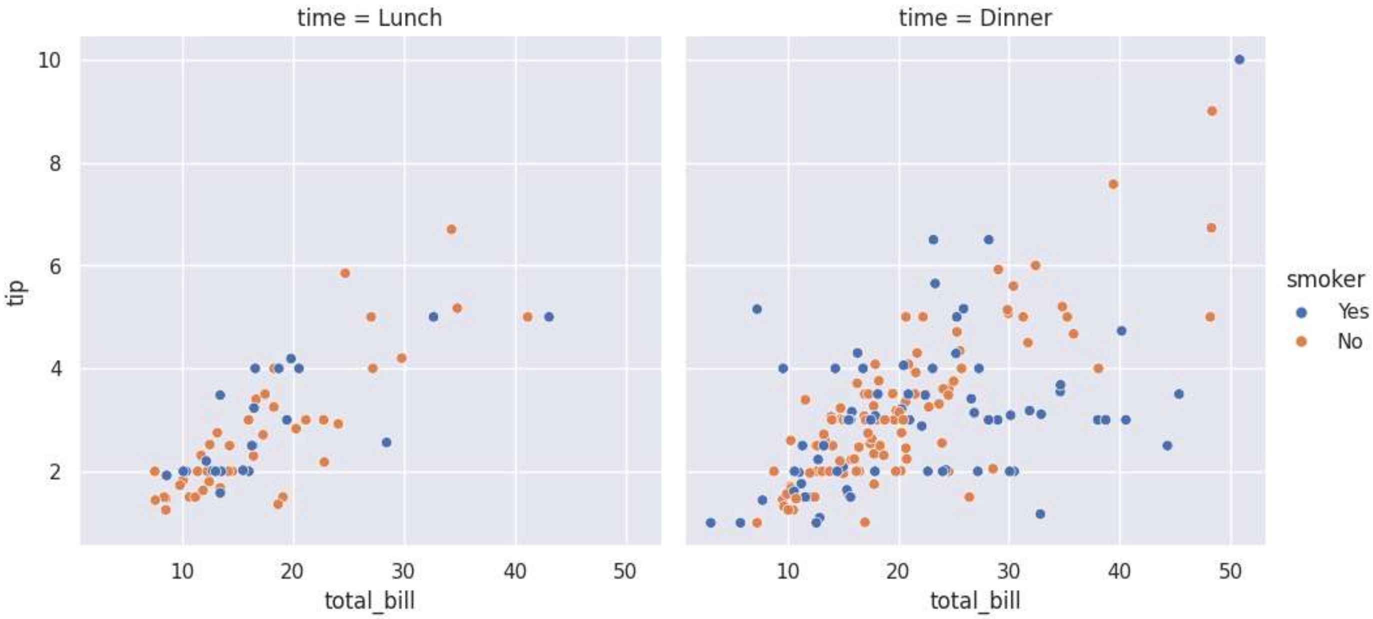
```
sns.relplot(
    data=fmri, kind="line",
     x="signal", y="timepoint", hue="event",
    orient="y",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bd69f6a0>
```



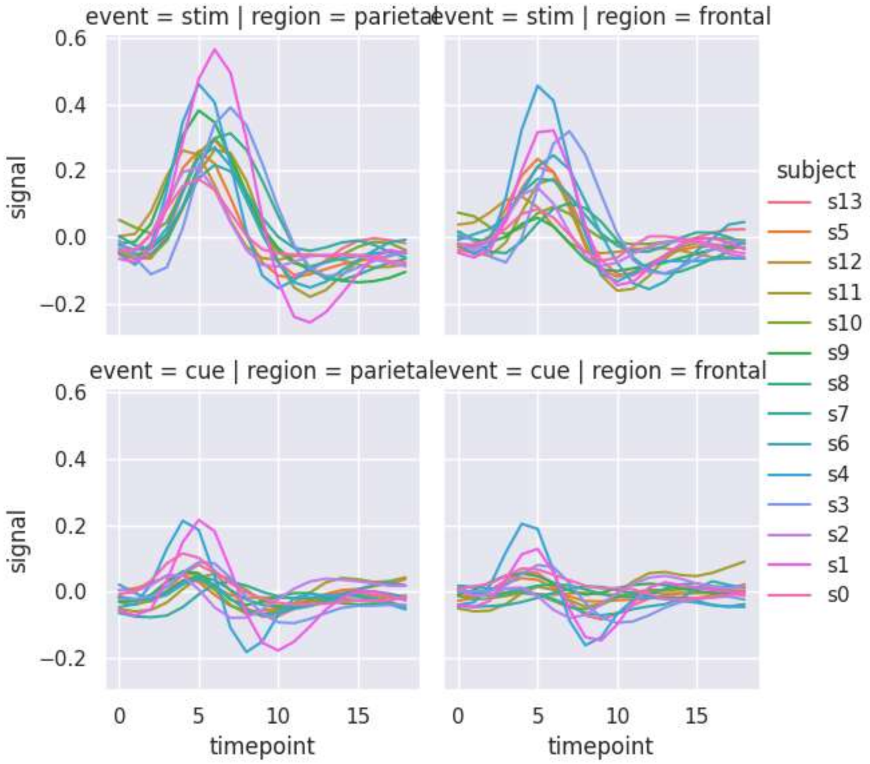
```
sns.relplot(
    data=tips,
    x="total_bill", y="tip", hue="smoker", col="time",
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bd5f7fd0>
```



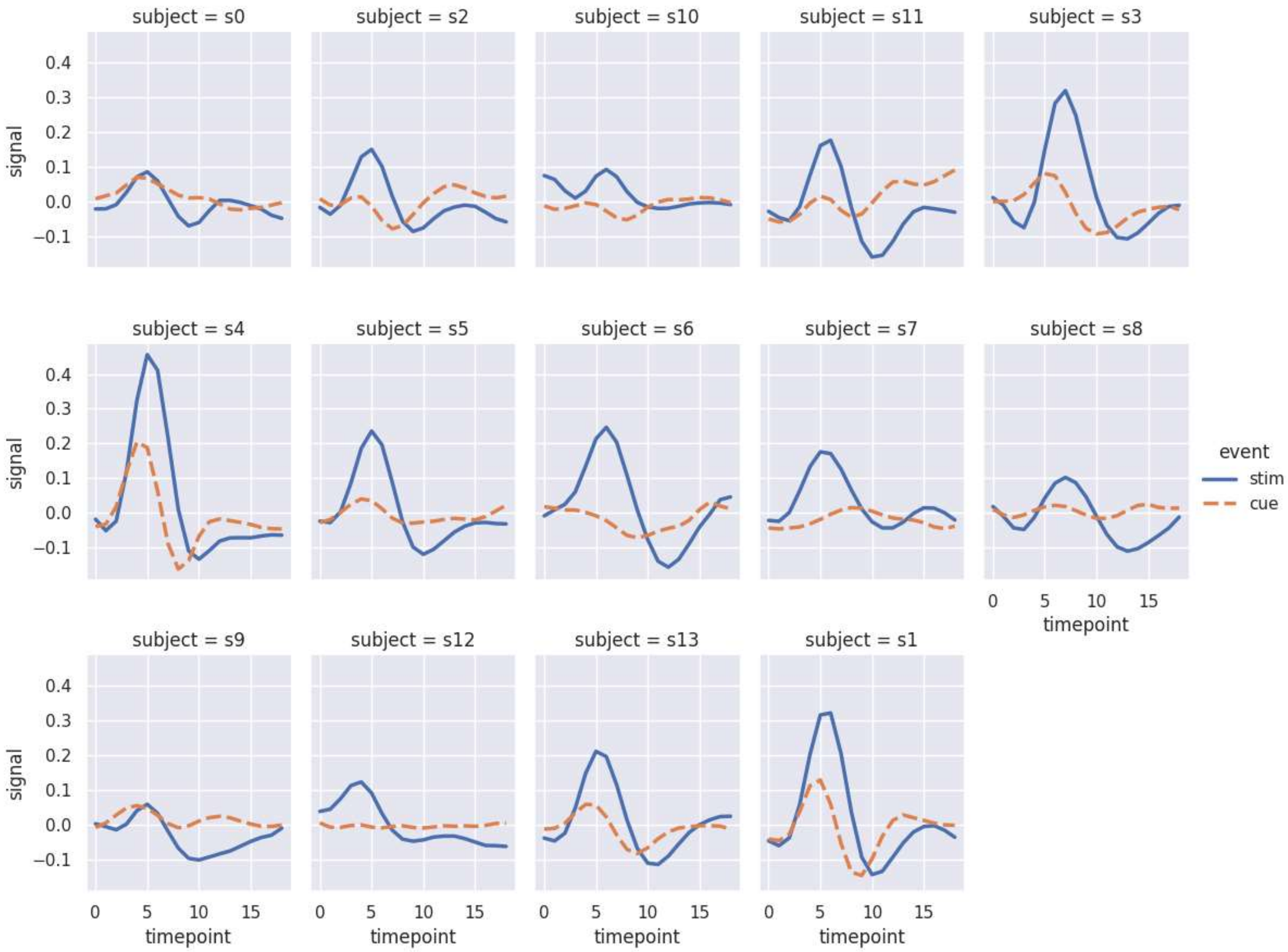
```
sns.relplot(
    data=fmri, kind="line",
    x="timepoint", y="signal", hue="subject",
    col="region", row="event", height=3,
    estimator=None
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bd4fb790>
```



```
sns.relplot(
    data=fmri.query("region == 'frontal'"), kind="line",
    x="timepoint", y="signal", hue="event", style="event",
    col="subject", col_wrap=5,
    height=3, aspect=.75, linewidth=2.5,
)
```

```
<seaborn.axisgrid.FacetGrid at 0x7992bd181780>
```



```
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/parenthood.csv")
```

```
data
```

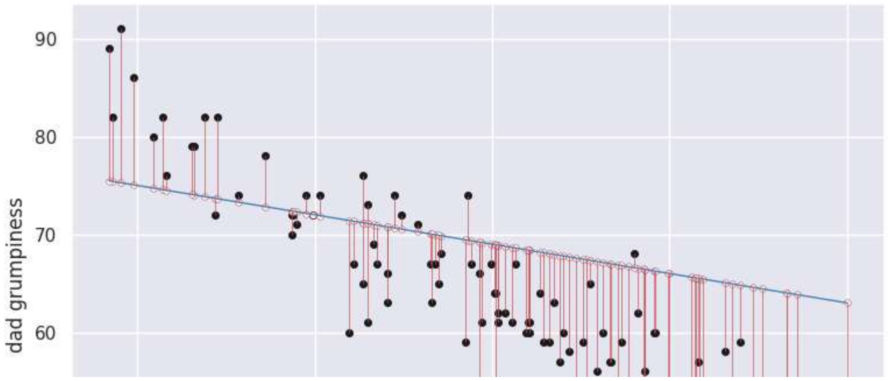
	ID	dan.sleep	baby.sleep	dan.grump	day
0	1	7.59	10.18	56	1
1	2	7.91	11.66	60	2
2	3	5.14	7.92	82	3
3	4	7.71	9.61	55	4
4	5	6.68	9.75	67	5
...	...	...	...	...	...
95	96	5.31	5.89	79	96
96	97	7.77	9.77	51	97
97	98	5.38	6.97	82	98
98	99	7.02	6.56	55	99
99	100	6.45	7.93	74	100

100 rows × 5 columns

```
import numpy as np
import matplotlib.pyplot as plt
import ipywidgets as widgets
from IPython.display import display

ph_df = pd.DataFrame({
    'dadsleep': data['dan.sleep'],
    'dadgrump': data['dan.grump']
})
@widgets.interact(slope=widgets.IntSlider(min=-20, max=20, step=1, value=0), intercept=widgets.IntSlider(min=0, max=200, step=1, value=50))
def plot_grumpiness(slope, intercept):
    predict = intercept+slope*ph_df.dadsleep
    resid = ph_df.dadgrump-predict
    fig, ax = plt.subplots(figsize=(9,6))

    x1 = np.linspace(ph_df.dadsleep.min(), ph_df.dadsleep.max(),400)
    y1 = intercept+slope*x1

    ax.plot(ph_df.dadsleep, ph_df.dadgrump,'ko',markersize=4)
    ax.plot(ph_df.dadsleep, ph_df.dadgrump-resid,'o',markersize=4,markeredgecolor='r', markeredgewidth=.4, markerfacecolor='white')

    ax.plot(x1,y1,'-',color='steelblue',linewidth=1)
    plt.xlabel("dad sleep")
    plt.ylabel("dad grumpiness")

    ax.vlines(ph_df.dadsleep, ph_df.dadgrump, ph_df.dadgrump-resid,'r',linewidth=0.5)
    plt.show()
```

slope -3

intercept 90


dad grumpiness