In [7]:
```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
boat = sns.load_dataset("titanic")
boat
```

Out[7]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	dec
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	Na
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	Na
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	Na
...	...	...	...	...	...	...	...	...	...	...	...	
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	Na
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	Na
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	Na

891 rows × 15 columns

In [22]:
```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
nuqta = sns.load_dataset("dots")
nuqta
```

Out[22]:

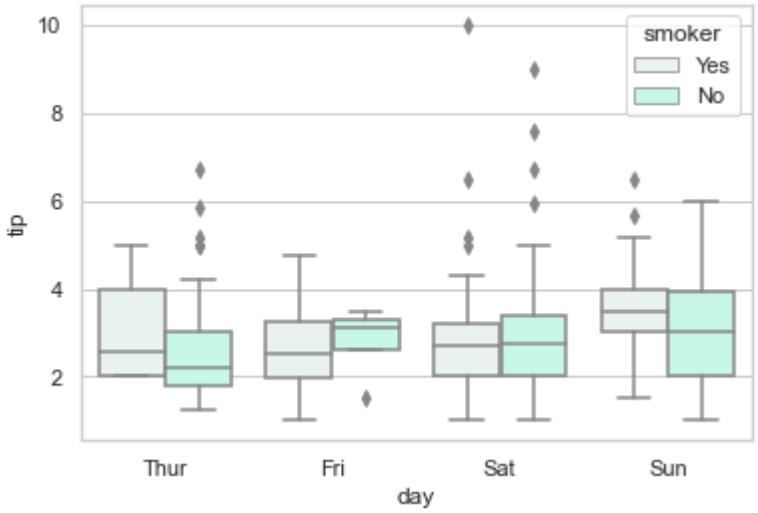
	align	choice	time	coherence	firing_rate
0	dots	T1	-80	0.0	33.189967
1	dots	T1	-80	3.2	31.691726
2	dots	T1	-80	6.4	34.279840
3	dots	T1	-80	12.8	32.631874
4	dots	T1	-80	25.6	35.060487
...	...	...	...	...	...
843	sacc	T2	300	3.2	33.281734

	align	choice	time	coherence	firing_rate
844	sacc	T2	300	6.4	27.583979
845	sacc	T2	300	12.8	28.511530
846	sacc	T2	300	25.6	27.009804
847	sacc	T2	300	51.2	30.959302

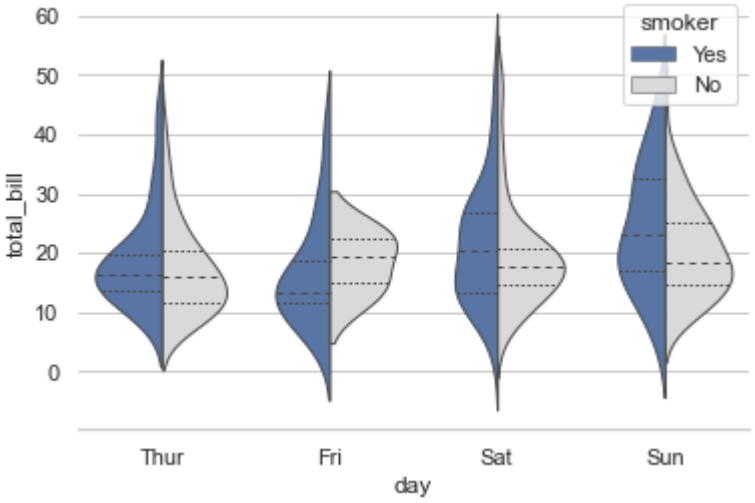
848 rows × 5 columns

In [24]:
```
#import libraries
import seaborn
#canvas (Baloon Board)
seaborn.set(style='whitegrid')
#Loading DataSets
tip = seaborn.load_dataset('tips')
seaborn.boxplot(x='day', y='tip', hue="smoker",data=tip, color= "#c0ffee" )
```

Out[24]:
```
<AxesSubplot:xlabel='day', ylabel='tip'>
```



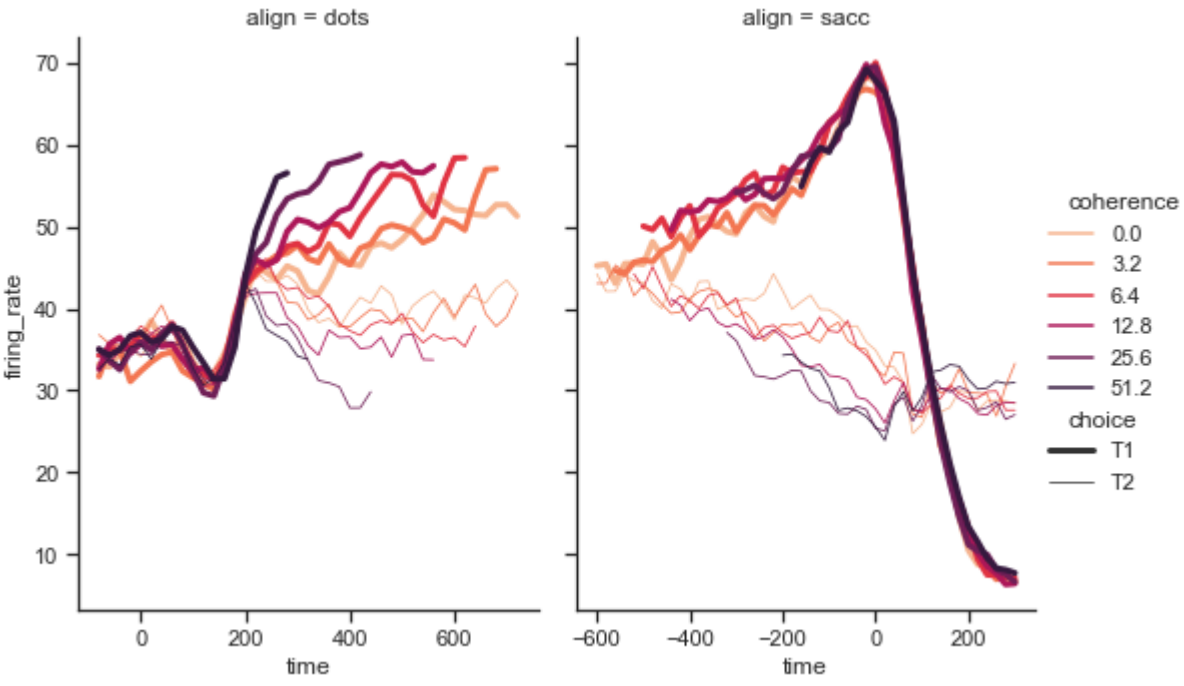
In [28]:
```
import seaborn as sns
sns.set_theme(style="whitegrid")

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

# Draw a nested violinplot and split the violins for easier comparison
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 [29]:

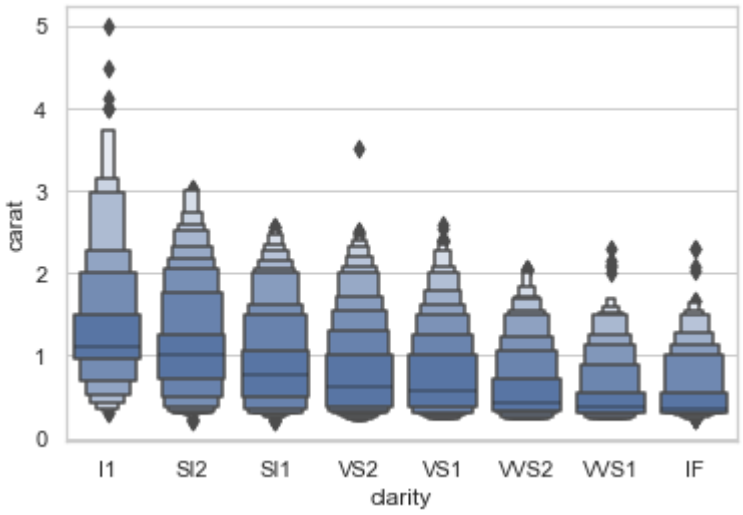
```
import seaborn as sns
sns.set_theme(style="ticks")

dots = sns.load_dataset("dots")

# Define the palette as a list to specify exact values
palette = sns.color_palette("rocket_r")

# Plot the lines on two facets
sns.relplot(
    data=dots,
    x="time", y="firing_rate",
    hue="coherence", size="choice", col="align",
    kind="line", size_order=["T1", "T2"], palette=palette,
    height=5, aspect=.75, facet_kws=dict(sharex=False),
)
```

Out[29]: <seaborn.axisgrid.FacetGrid at 0x1a346515dc0>

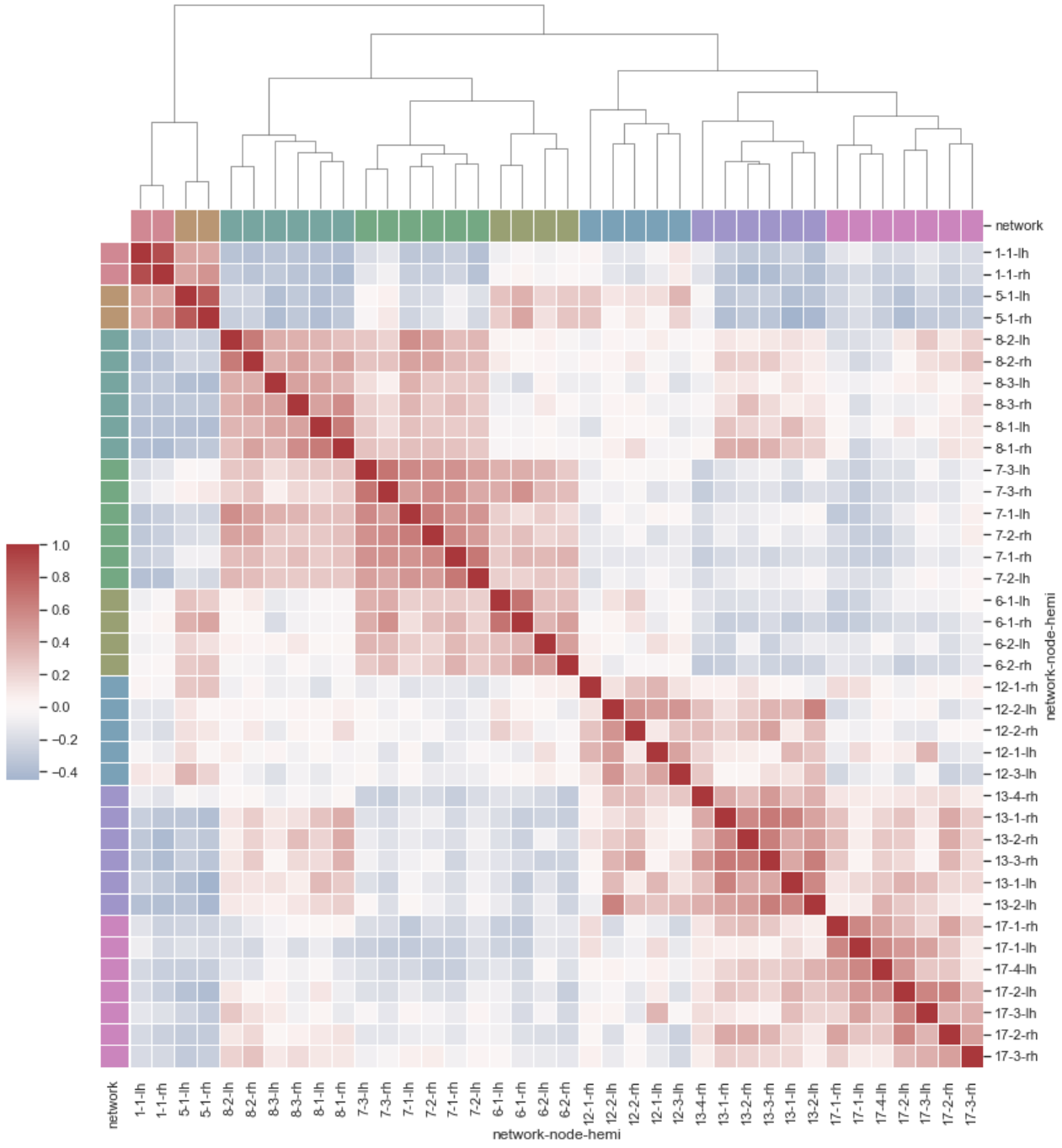


In [31]:
```
import seaborn as sns
sns.set_theme(style="whitegrid")

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

sns.boxenplot(x="clarity", y="carat",
              color="b", order=clarity_ranking,
              scale="linear", data=diamonds)
```

Out[31]:
```
<AxesSubplot:xlabel='clarity', ylabel='carat'>
```



In [32]:
```
import pandas as pd
import seaborn as sns
sns.set_theme()

# Load the brain networks example dataset
df = sns.load_dataset("brain_networks", header=[0, 1, 2], index_col=0)

# Select a subset of the networks
used_networks = [1, 5, 6, 7, 8, 12, 13, 17]
used_columns = (df.columns.get_level_values("network")
                          .astype(int)
                          .isin(used_networks))
df = df.loc[:, used_columns]

# Create a categorical palette to identify the networks
network_pal = sns.husl_palette(8, s=.45)
network_lut = dict(zip(map(str, used_networks), network_pal))

# Convert the palette to vectors that will be drawn on the side of the matrix
networks = df.columns.get_level_values("network")
network_colors = pd.Series(networks, index=df.columns).map(network_lut)

# Draw the full plot
g = sns.clustermap(df.corr(), center=0, cmap="vlag",
                   row_colors=network_colors, col_colors=network_colors,
                   dendrogram_ratio=(.1, .2),
                   cbar_pos=(.02, .32, .03, .2),
                   linewidths=.75, figsize=(12, 13))

g.ax_row_dendrogram.remove()
```

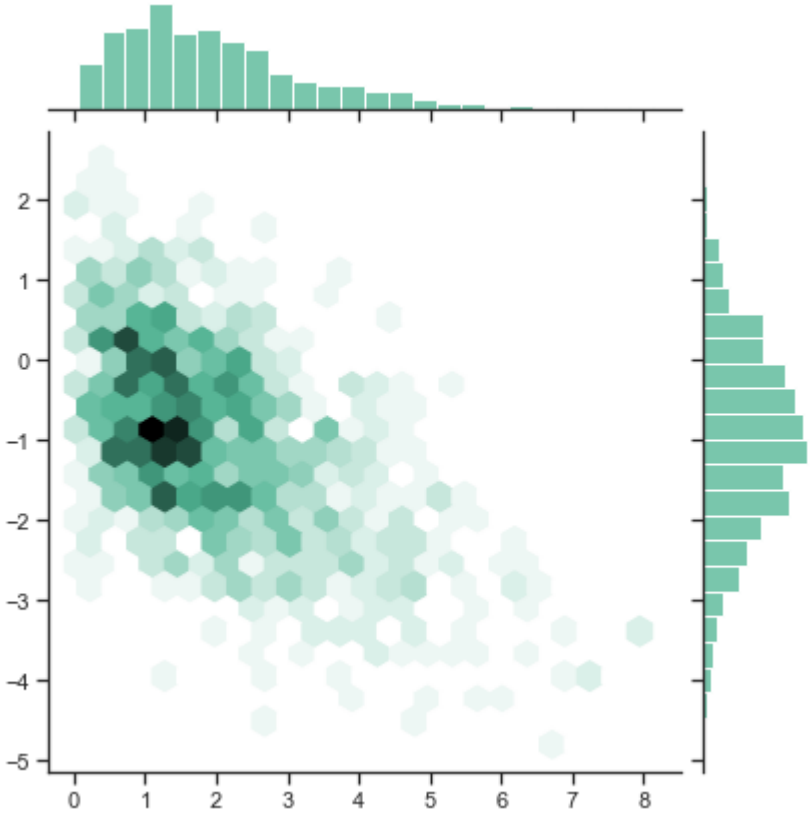


In [33]:

```
import numpy as np
import seaborn as sns
sns.set_theme(style="ticks")

rs = np.random.RandomState(11)
x = rs.gamma(2, size=1000)
y = -.5 * x + rs.normal(size=1000)

sns.jointplot(x=x, y=y, kind="hex", color="#4CB391")
```

Out[33]:

```
<seaborn.axisgrid.JointGrid at 0x1a346859790>
```

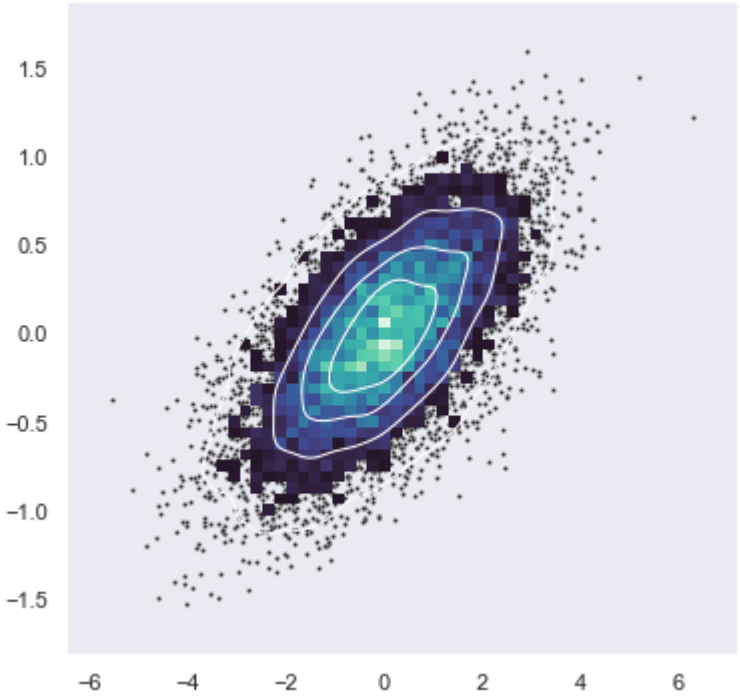


In [38]:

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

# Simulate data from a bivariate Gaussian
n = 10000
mean = [0, 0]
cov = [(2, .4), (.4, .2)]
rng = np.random.RandomState(0)
x, y = rng.multivariate_normal(mean, cov, n).T

# Draw a combo histogram and scatterplot with density contours
f, ax = plt.subplots(figsize=(6, 6))
sns.scatterplot(x=x, y=y, s=5, color=".15")
sns.histplot(x=x, y=y, bins=50, pthresh=.1, cmap="mako")
sns.kdeplot(x=x, y=y, levels=5, color="w", linewidths=1)
```

Out[38]:

```
<AxesSubplot:>
```

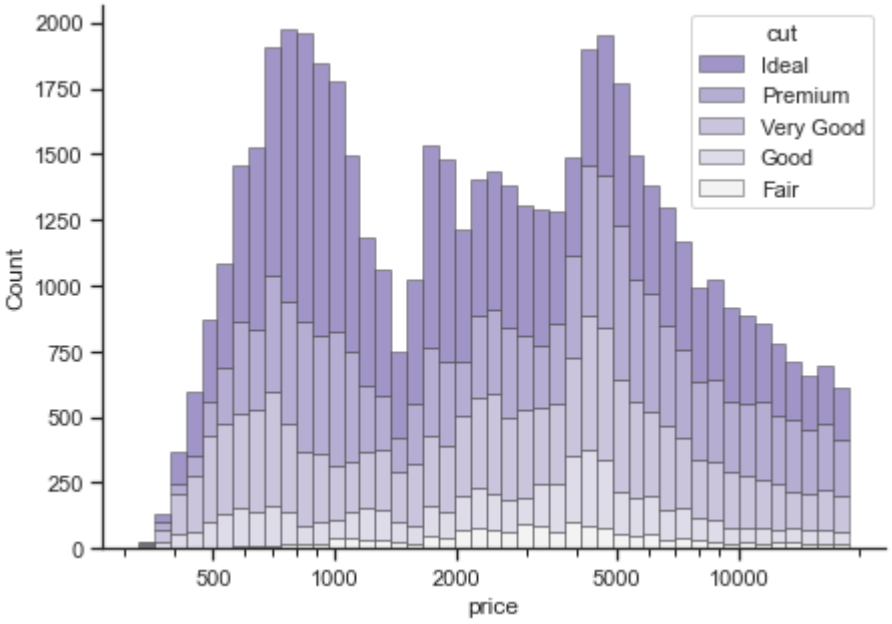


In [40]:

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

sns.set_theme(style="ticks")

diamonds = sns.load_dataset("diamonds")

f, ax = plt.subplots(figsize=(7, 5))
sns.despine(f)

sns.histplot(
    diamonds,
    x="price", hue="cut",
    multiple="stack",
    palette="light:m_r",
    edgecolor=".3",
    linewidth=.5,
    log_scale=True,
)
ax.xaxis.set_major_formatter(mpl.ticker.ScalarFormatter())
ax.set_xticks([500, 1000, 2000, 5000, 10000])
```

Out[40]:

```
[<matplotlib.axis.XTick at 0x1a347d48850>,
 <matplotlib.axis.XTick at 0x1a347d48820>,
 <matplotlib.axis.XTick at 0x1a347df1040>,
 <matplotlib.axis.XTick at 0x1a347f4de20>,
 <matplotlib.axis.XTick at 0x1a347f5f5e0>]
```


cut
Ideal
Premium
Very Good
Good
Fair
Count
2000
1750
1500
1250
1000
750
500
250
0
500
1000
2000
5000
10000
price