In [13]:
```python
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

In [14]:
```python
%matplotlib inline

import seaborn as sns
```
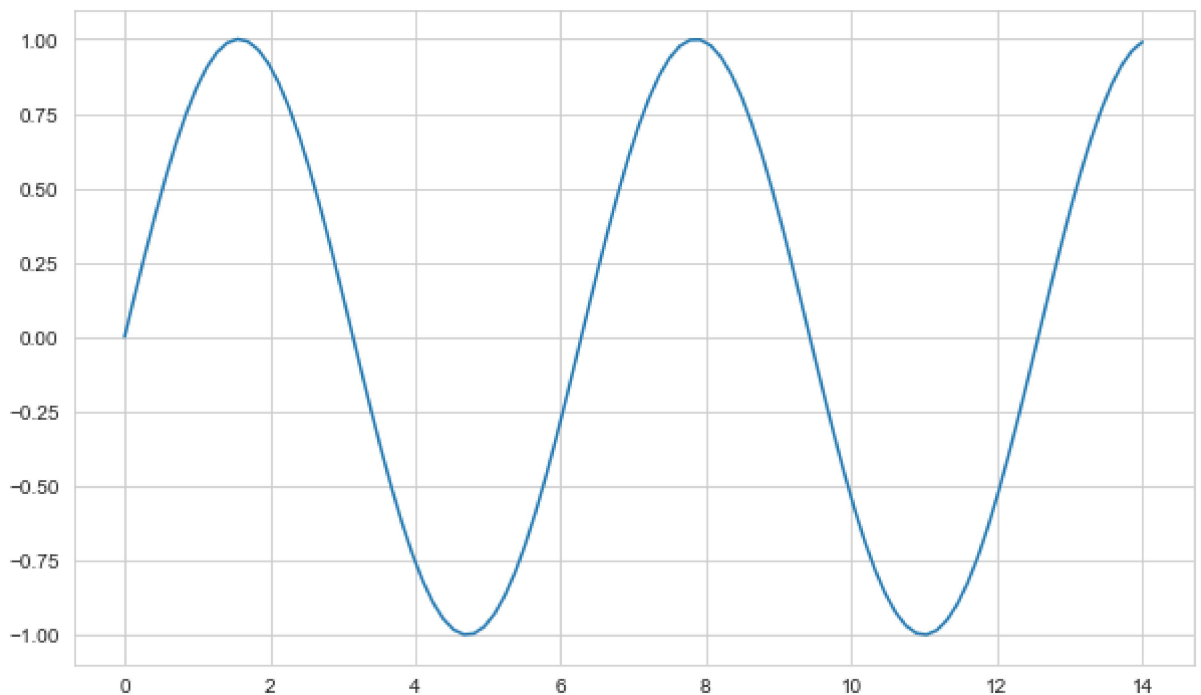
In [15]:
```python
x = np.linspace(0,14,100)
y1 = np.sin(x)
y2 = 2*np.sin(x+0.5)
y3 = 4*np.sin(x+1.0)

plt.figure(figsize=(10,6))   # 그림의 크기
plt.plot(x, y1)
```
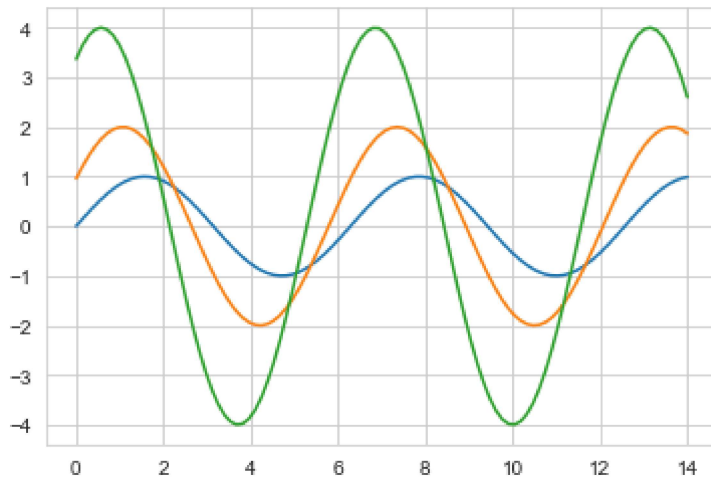
Out[15]: [<matplotlib.lines.Line2D at 0x278c5e5de20>]

In [16]:
```python
plt.plot(x, y1, x,y2, x, y3)   # 3개의 sin 그래프
```

Out[16]: [<matplotlib.lines.Line2D at 0x278c6004550>,
 <matplotlib.lines.Line2D at 0x278c6004610>,
 <matplotlib.lines.Line2D at 0x278c60046d0>]

새로운 스타일 적용

In [17]:
```
sns.set_style("whitegrid")
```

In [18]:
```
plt.plot(x, y1, x,y2, x, y3)  # 3개의 sin 그래프
```
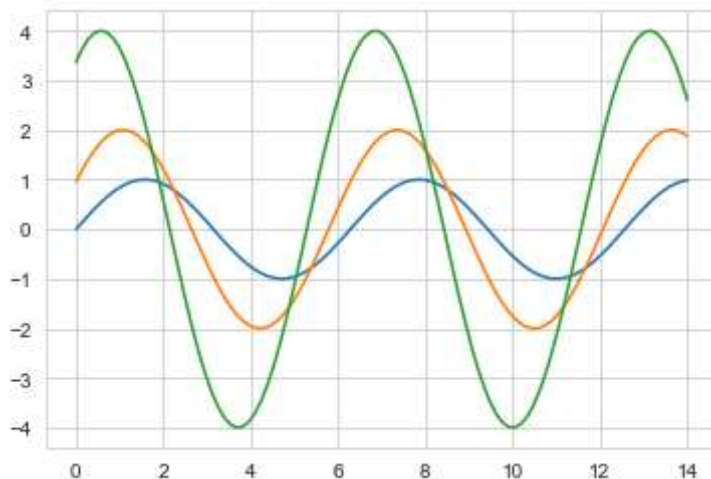
Out[18]:
```
[<matplotlib.lines.Line2D at 0x278c5ee1d60>,
 <matplotlib.lines.Line2D at 0x278c5ee1e20>,
 <matplotlib.lines.Line2D at 0x278c5ee1ee0>]
```


tip 데이터로 데이터 살펴보고 인사이트 얻기

In [19]:
```
sns.set_style("whitegrid")
```

In [20]:
```
tips = sns.load_dataset("tips")  # 인터넷이 켜져 있어야 함.
tips
```

Out[20]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2

	total_bill	tip	sex	smoker	day	time	size
4	24.59	3.61	Female	No	Sun	Dinner	4
...	...	...	...	...	...	...	...
239	29.03	5.92	Male	No	Sat	Dinner	3
240	27.18	2.00	Female	Yes	Sat	Dinner	2
241	22.67	2.00	Male	Yes	Sat	Dinner	2
242	17.82	1.75	Male	No	Sat	Dinner	2
243	18.78	3.00	Female	No	Thur	Dinner	2

244 rows × 7 columns

In [21]:
```
tips.head()  ## 앞의 데이터 조금만 살펴보기
```

Out[21]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

Boxplot을 이용한 시각화 하기

요일별 식사금액은 얼마나 될까?

In [22]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x="day", y="total_bill", data=tips)
plt.show()
```
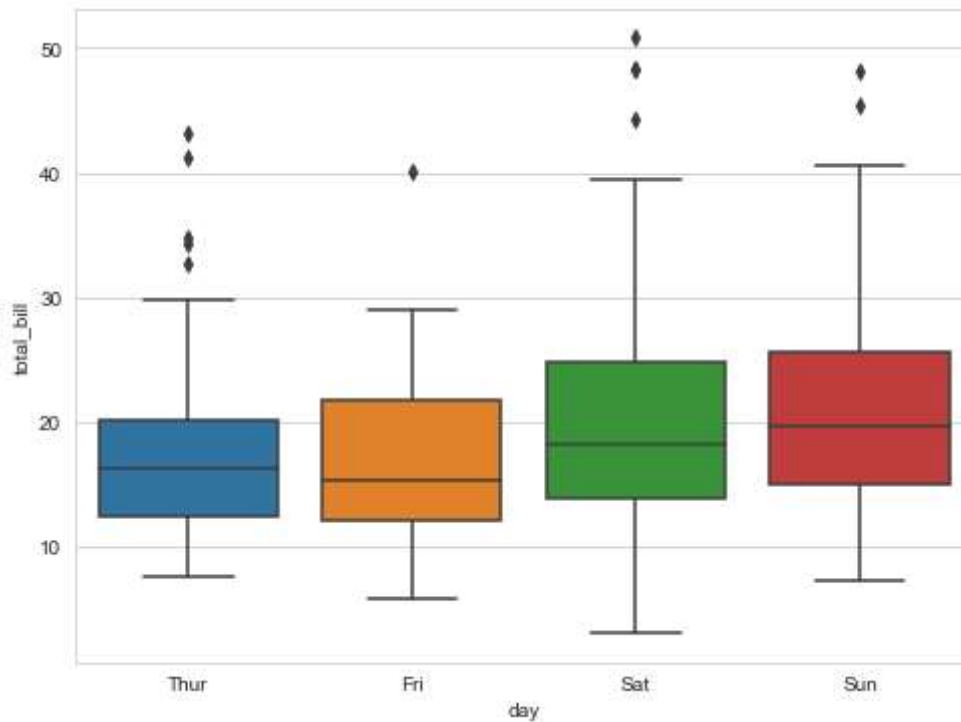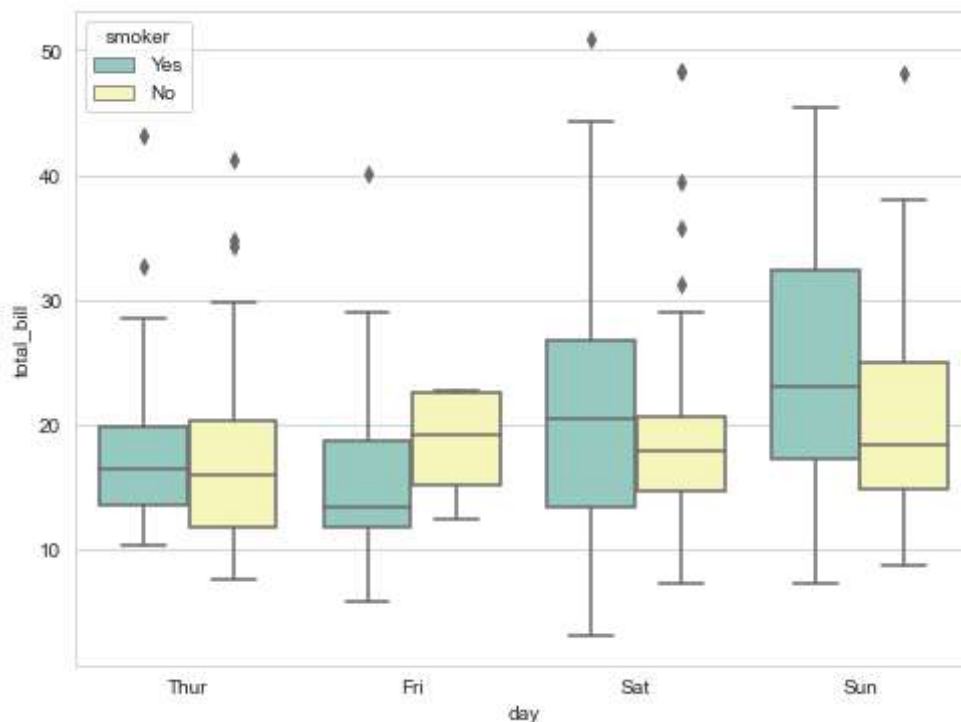
실습1. 요일별 Tip은 얼마나 될까?

```
plt.figure(figsize=(8,6))  # 사이즈
_____           # 빈칸을 채워보자.
plt.show()
```

요일별 식사 금액, 그런데 흡연자와 비흡연자를 비교해 보자.

In [24]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x="day", y="total_bill", hue="smoker", data=tips, palette="Set3")
plt.show()
```
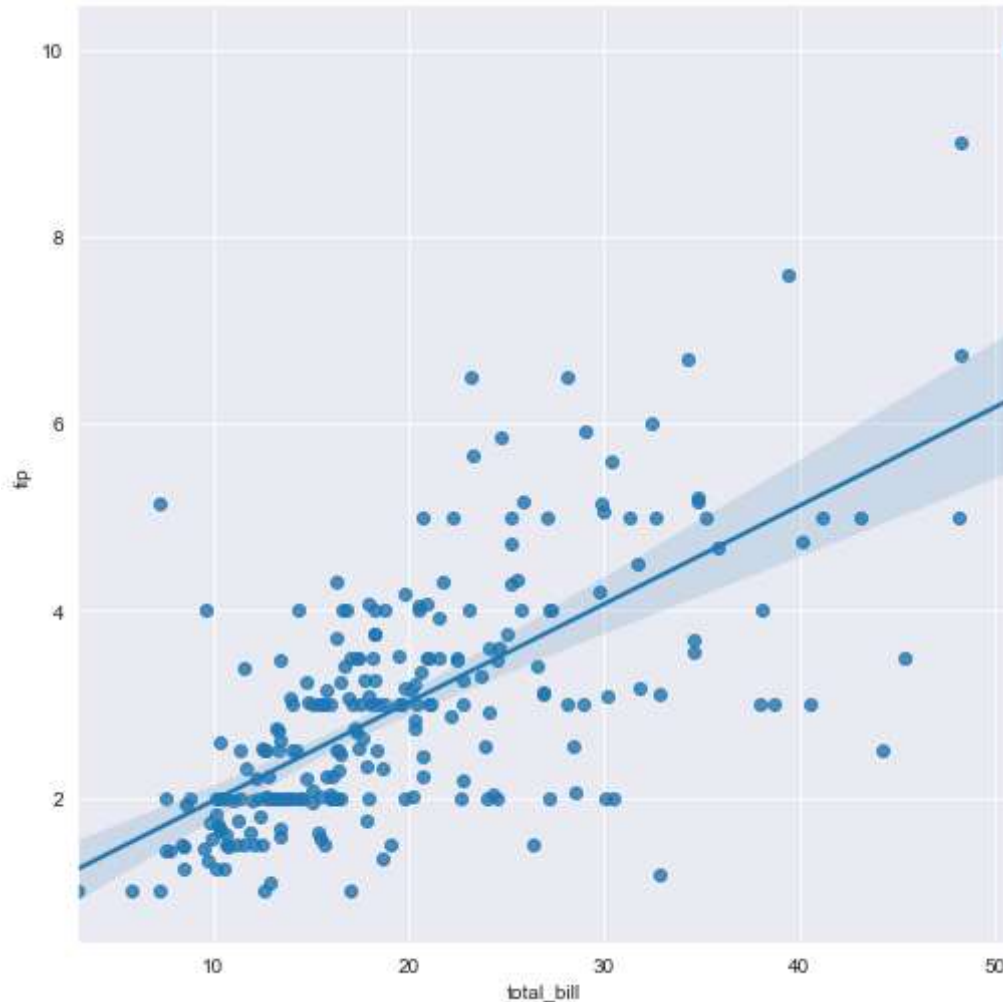

Regression(회귀선)을 그어서 대략적인 예측을 수행해보자.

식사금액과 팁의 상관관계

In [25]:
```
sns.set_style("darkgrid")
sns.lmplot(x="total_bill", y="tip", data=tips, size=7)
plt.show()
```

C:\Users\toto\anaconda3\lib\site-packages\seaborn\regression.py:580: UserWarning: The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)

그러면 담배 필 때와 안 피는 사람은?

In [26]:
```
sns.lmplot(x="total_bill", y="tip", hue="smoker", data=tips, palette="Set1", size=7)
plt.show()
```
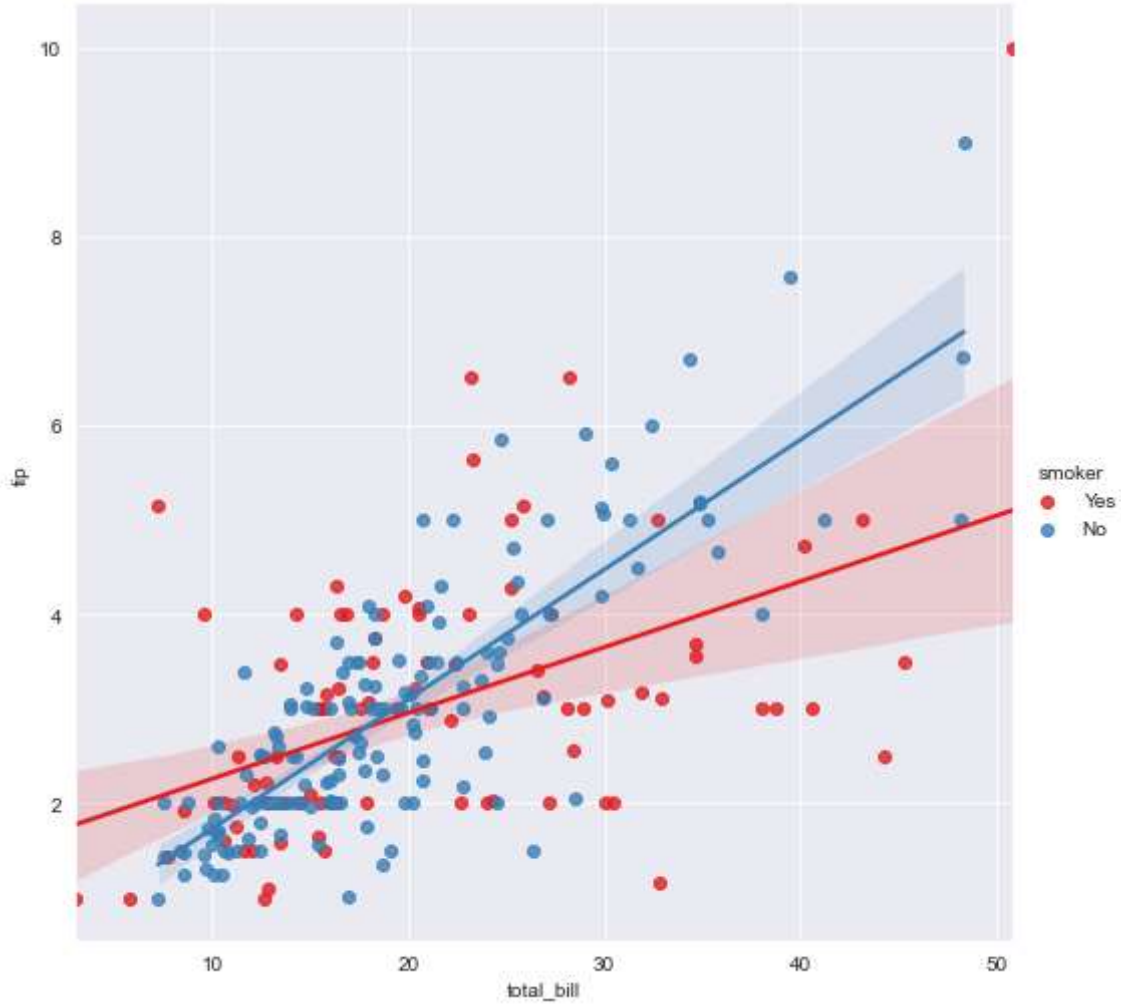
C:\Users\toto\anaconda3\lib\site-packages\seaborn\regression.py:580: UserWarning: The `size` parameter has been renamed to `height`; please update your code.
  warnings.warn(msg, UserWarning)

항공 데이터 이용

연도별   월   승객
...

```
In [27]:  fg = sns.load_dataset("flights")
          fg.head(5)
          fg
```

Out[27]:

	year	month	passengers
0	1949	Jan	112
1	1949	Feb	118
2	1949	Mar	132
3	1949	Apr	129
4	1949	May	121
...	...	...	...
139	1960	Aug	606
140	1960	Sep	508
141	1960	Oct	461
142	1960	Nov	390
143	1960	Dec	432

144 rows × 3 columns
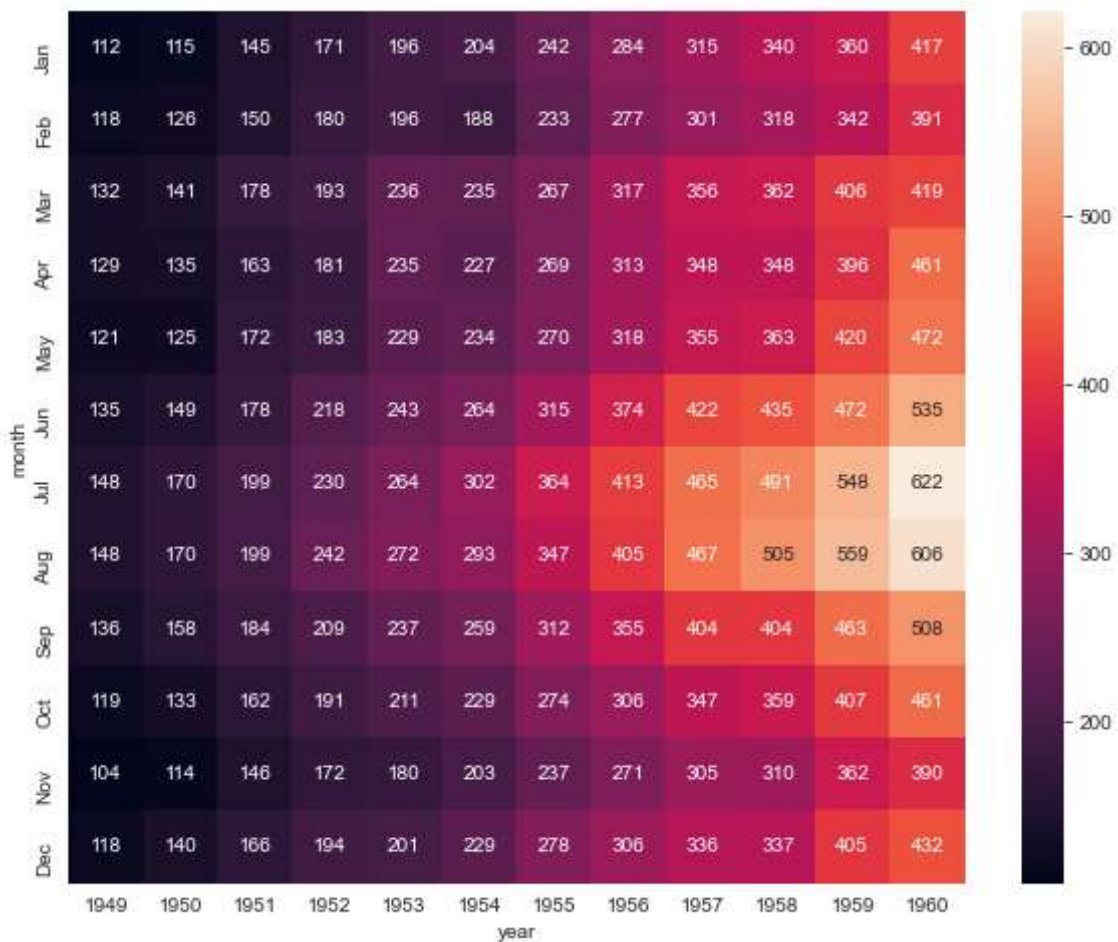
In [28]:
```python
type(fg)
```

Out[28]: pandas.core.frame.DataFrame

In [29]:
```python
fgp = fg.pivot("month", "year", "passengers")
fgp
```

Out[29]:

year	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
month												
Jan	112	115	145	171	196	204	242	284	315	340	360	417
Feb	118	126	150	180	196	188	233	277	301	318	342	391
Mar	132	141	178	193	236	235	267	317	356	362	406	419
Apr	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
Jun	135	149	178	218	243	264	315	374	422	435	472	535
Jul	148	170	199	230	264	302	364	413	465	491	548	622
Aug	148	170	199	242	272	293	347	405	467	505	559	606
Sep	136	158	184	209	237	259	312	355	404	404	463	508
Oct	119	133	162	191	211	229	274	306	347	359	407	461
Nov	104	114	146	172	180	203	237	271	305	310	362	390
Dec	118	140	166	194	201	229	278	306	336	337	405	432

In [30]:
```python
plt.figure(figsize=(10,8))
sns.heatmap(fgp, annot=True, fmt="d")
plt.show()
```

	1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960
Jan	112	115	145	171	196	204	242	284	315	340	360	417
Feb	118	126	150	180	196	188	233	277	301	318	342	391
Mar	132	141	178	193	236	235	267	317	356	362	406	419
Apr	129	135	163	181	235	227	269	313	348	348	396	461
May	121	125	172	183	229	234	270	318	355	363	420	472
Jun	135	149	178	218	243	264	315	374	422	435	472	535
Jul	148	170	199	230	264	302	364	413	465	491	548	622
Aug	148	170	199	242	272	293	347	405	467	505	559	606
Sep	136	158	184	209	237	259	312	355	404	404	463	508
Oct	119	133	162	191	211	229	274	306	347	359	407	461
Nov	104	114	146	172	180	203	237	271	305	310	362	390
Dec	118	140	166	194	201	229	278	306	336	337	405	432

iris 데이터를 살펴보기

In [31]:
```python
sns.set(style="ticks")
iris = sns.load_dataset("iris")
iris
```

Out[31]:

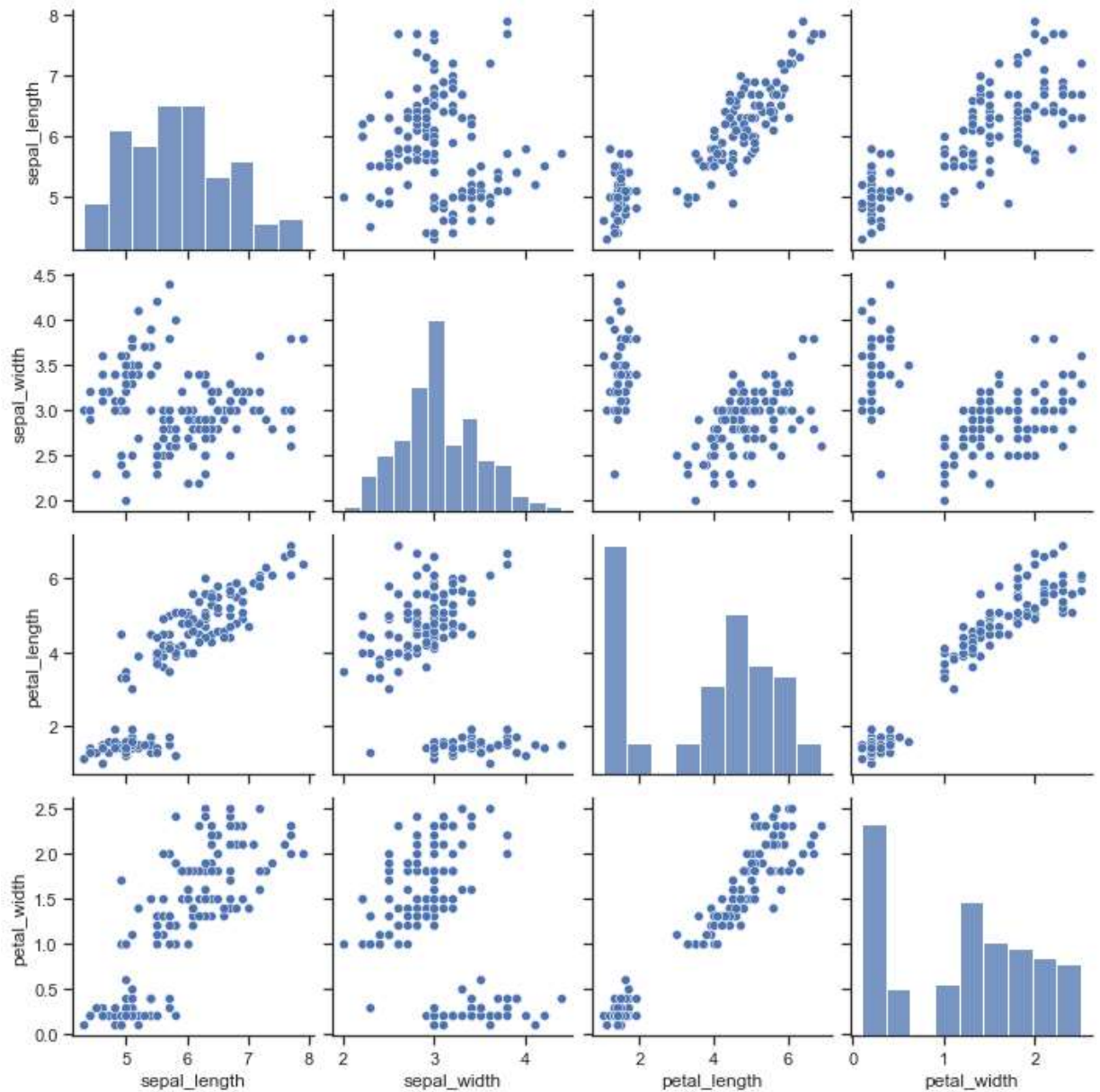|	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	virginica
146	6.3	2.5	5.0	1.9	virginica
147	6.5	3.0	5.2	2.0	virginica
148	6.2	3.4	5.4	2.3	virginica
149	5.9	3.0	5.1	1.8	virginica

150 rows × 5 columns

In [32]:
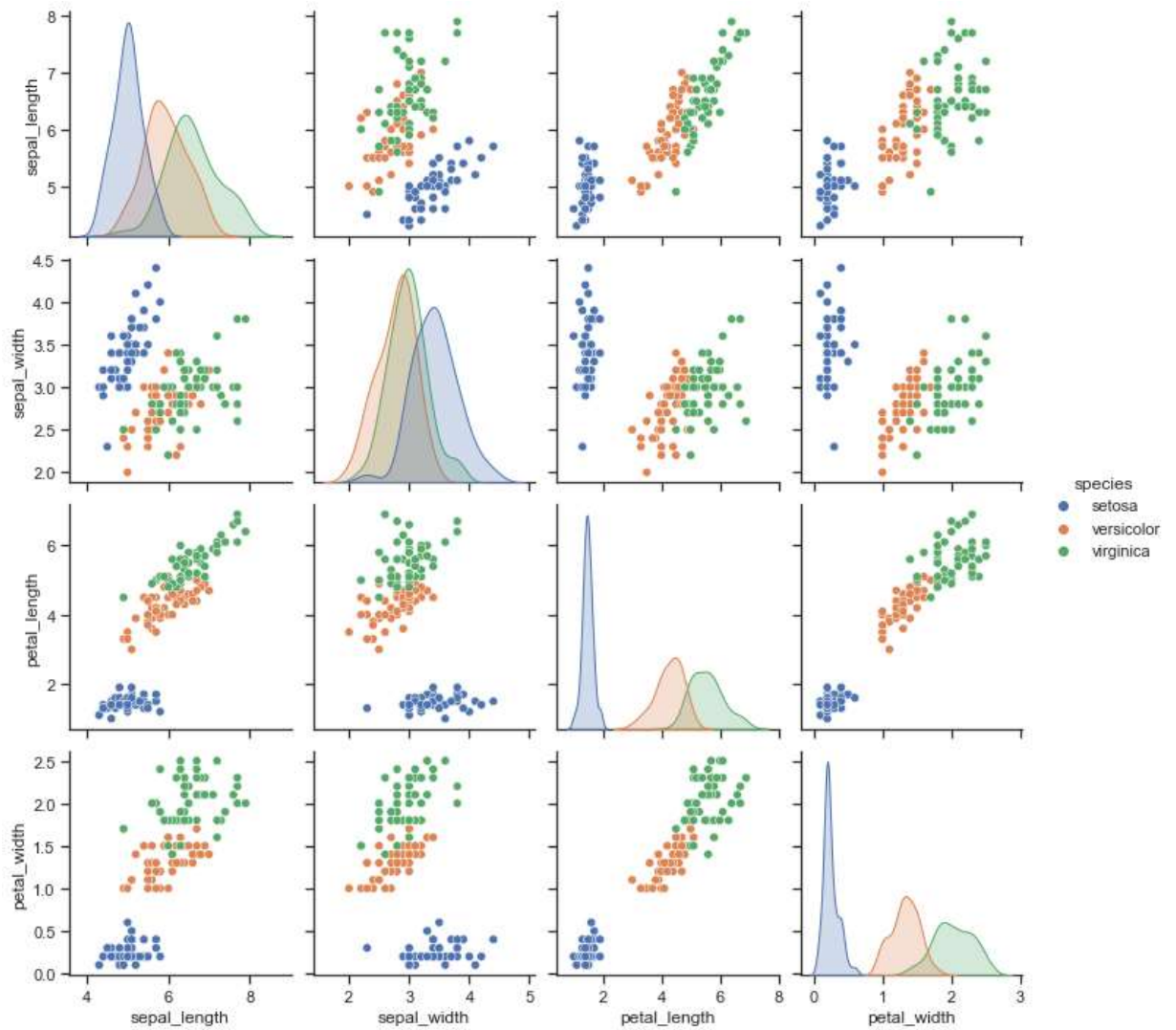
```
sns.pairplot(iris)
```

Out[32]:   <seaborn.axisgrid.PairGrid at 0x278c6329310>

In [33]:
```
sns.pairplot(iris, hue="species")
plt.show()
```

Reference

http://seaborn.pydata.org/generated/seaborn.heatmap.html

In [ ]: