**Seaborn을 활용하면 시각화가 쉬워 집니다(사실 외울게 많습니다...ㅠㅠ)**[**¶**](#gjdgxs)

먼저 터미널이나 명령 프롬포트에서 **pip install seaborn**을 입력하여 seaborn 모듈을 설치 합니다. 좀더 예쁘게 시각화 할 수 있도록 도와줍니다

먼저 seaborn을 사용하지 않고 matplot만 사용하여 그래프를 그려 보겠습니다

In [5]:

**import** **matplotlib.pyplot** **as** **plt**  
**import** **numpy** **as** **np**  
%**matplotlib** inline  
  
x = np.linspace(0, 14, 100)  
  
y1 = np.sin(x)  
y2 = 2\*np.sin(x+0.5)  
y3 = 3\*np.sin(x+1.0)  
y4 = 4\*np.sin(x+1.5)  
  
plt.figure(figsize=(10,6))  
plt.plot(x, y1, x,y2, x,y3, x,y4)  
plt.show()

예쁘긴 하지만 이번엔 seaborn을 활용해서 여러가지 스타일을 지정해 보겠습니다. seaborn을 사용 할 때 반드시 matplotlib 모듈도 같이 import 되어 있어야 합니다

In [7]:

**import** **seaborn** **as** **sns**  
  
sns.set\_style('whitegrid') *# seaborn의 whitegrid 스타일 지정하기*  
  
plt.figure(figsize=(10,6))  
plt.plot(x,y1, x,y2, x,y3, x,y4)  
plt.show()

Seaborn에 있는 몇가지 연습 데이터셋을 사용 해 보도록 하겠습니다. 먼저 요일별 점심, 저녁, 흡연 여부와 식사 금액과 팁을 정리한 데이터 입니다

In [8]:

**import** **matplotlib.pyplot** **as** **plt**  
**import** **numpy** **as** **np**  
**import** **seaborn** **as** **sns**  
  
sns.set\_style('whitegrid')  
%**matplotlib** inline  
  
tips = sns.load\_dataset('tips') *# seaborn 에서 기본적으로 제공하는 tips 데이터셋 불러오기*  
tips.head()

Out[8]:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **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 |

In [10]:

plt.figure(figsize=(8,6))  
sns.boxplot(x='day', y='total\_bill', data=tips) *# 박스 그래프 로 보기. x축은 day(요일), y축은 전체 금액(total\_bill)*  
plt.show()

위 데이터는 단순히 요일별 매출을 시각화 해본 그래프 입니다. 이것만으로도 충분히 시각화가 쉬워지지만,

**hue**라는 옵션을 이용하면 또다른 데이터를 이용해서 데이터를 구분 해 볼 수 있습니다.

In [12]:

plt.figure(figsize=(8,6))  
sns.boxplot(x='day', y='total\_bill', hue='smoker', data=tips, palette='Set3') *# 흡연자(smoker)로 추가 구분하기*  
plt.show()

위의 그래프를 확인 해보면 흡연자가 더 결제범위가 큰 것이 확인됩니다

이번엔 약간 다른 스타일의 그래프를 그려 보겠습니다. 결제 금액과 종업원에게 주는 팁에 대한 회귀 분석 시각화 입니다.

lmplot을 활용하면 간단하게 데이터 분석이 가능합니다.

**적절한 상관계수를 확인 할 때 사용하기 좋은 lmplot**[**¶**](#30j0zll)

In [14]:

sns.set\_style('darkgrid') *# darkgrid 스타일의 그래프 그리기*  
sns.lmplot(x='total\_bill', y='tip', data=tips, size=7) *# x축 결제금액, y축 종업원에게 주는 팁*  
plt.show()

회기 분석은 간단하게 말해서 예전에 그려봤던 상관계수라고 보시면 될 것 같습니다.

일전에 상관계수를 데이터와 데이터 사이에 상관 정도를 구해서 분석의 용도로 활용 했는데, 회기 분석은 그 상관관계를 구하는 조금 더 정확한 방법 이라고 보시면 될 것 같습니다.

즉 위의 그래프에서 제시하는 결제금액에 따른 적절한 팁 액수를 확인 할 수 있다는 것입니다.

아무튼 지금은 total\_bill과 tip은 양의 상관 관계를 갖는다 라고 이야기 합니다.( total\_bill이 많아 질수록 tip이 많아지네요)

또한 유효한 범위까지 그림으로 그려주는 것이 확인 됩니다. hue를 이용해서 흡연자로 비교해 보겠습니다

In [15]:

sns.lmplot(x='total\_bill', y='tip', hue='smoker', data=tips, palette='Set1', size=7) *#흡연자 데이터로 비교하기*  
plt.show()

lmplot 시각화도 hue옵션을 가질 수 있고 palette 옵션을 이용해 색상도 지정해 주었습니다.

**다음은 연도 및 월별 항공기 승객수를 기록한 데이터 입니다**[**¶**](#1fob9te)

In [20]:

flights = sns.load\_dataset("flights")  
flights.head()

Out[20]:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **year** | **month** | **passengers** |
| **0** | 1949 | January | 112 |
| **1** | 1949 | February | 118 |
| **2** | 1949 | March | 132 |
| **3** | 1949 | April | 129 |
| **4** | 1949 | May | 121 |

연도 및 월별 항공기 승객수로 구분하기 위해 pivot\_table을 활용 하겠습니다.

In [21]:

flights = flights.pivot("month", "year", "passengers") *#month별, year컬럼, passengers 데이터*  
flights.head()

Out[21]:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **year** | **1949** | **1950** | **1951** | **1952** | **1953** | **1954** | **1955** | **1956** | **1957** | **1958** | **1959** | **1960** |
| **month** |  |  |  |  |  |  |  |  |  |  |  |  |
| **January** | 112 | 115 | 145 | 171 | 196 | 204 | 242 | 284 | 315 | 340 | 360 | 417 |
| **February** | 118 | 126 | 150 | 180 | 196 | 188 | 233 | 277 | 301 | 318 | 342 | 391 |
| **March** | 132 | 141 | 178 | 193 | 236 | 235 | 267 | 317 | 356 | 362 | 406 | 419 |
| **April** | 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 |

**값별 수치를 색상으로! heatmap을 사용하여 데이터 확인하기**[**¶**](#3znysh7)

heatmap을 활용하면 수치 별 시각화를 정말 쉽게 확인 할 수 있습니다. 일전 데이터를 보고 pivot을 활용한다면 heatmap으로 데이터를 확인해보세요!

In [29]:

plt.figure(figsize=(10,8))  
sns.heatmap(flights, annot=**True**, fmt="d")  
plt.show()

**각 항목별 비교로 확인해 보기 pairplot**[**¶**](#2et92p0)

머신러닝에서 많이 사용되는 아이리스 꽃에 대한 데이터 입니다. 각각의 데이터가 의미 하는 바는 다음과 같습니다.

* Species : 붓꽃의 종. setosa, versicolor, virginica 세 가지 값 중 하나
* Sepal.Width : 꽃받침의 너비
* Sepal.Length : 꽃받침의 길이
* Petal.Width : 꽃잎의 너비
* Petal.Length : 꽃잎의 길이

In [30]:

sns.set(style='ticks')  
iris = sns.load\_dataset('iris')  
iris.head(10)

Out[30]:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **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 |
| **5** | 5.4 | 3.9 | 1.7 | 0.4 | setosa |
| **6** | 4.6 | 3.4 | 1.4 | 0.3 | setosa |
| **7** | 5.0 | 3.4 | 1.5 | 0.2 | setosa |
| **8** | 4.4 | 2.9 | 1.4 | 0.2 | setosa |
| **9** | 4.9 | 3.1 | 1.5 | 0.1 | setosa |

In [31]:

sns.pairplot(iris, hue='species') *# 꽃 종류별 데이터 비교 분석 하기*  
plt.show()

아주 간단하게 데이터 분석이 가능해집니다. 아이리스 예제는 인공지능 머신러닝에서 매우 중요하게 다뤄지는 예제 입니다.

# **이제 본격적으로 범죄 데이터를 시각화 해보겠습니다.**[**¶**](#tyjcwt)

In [ ]: