

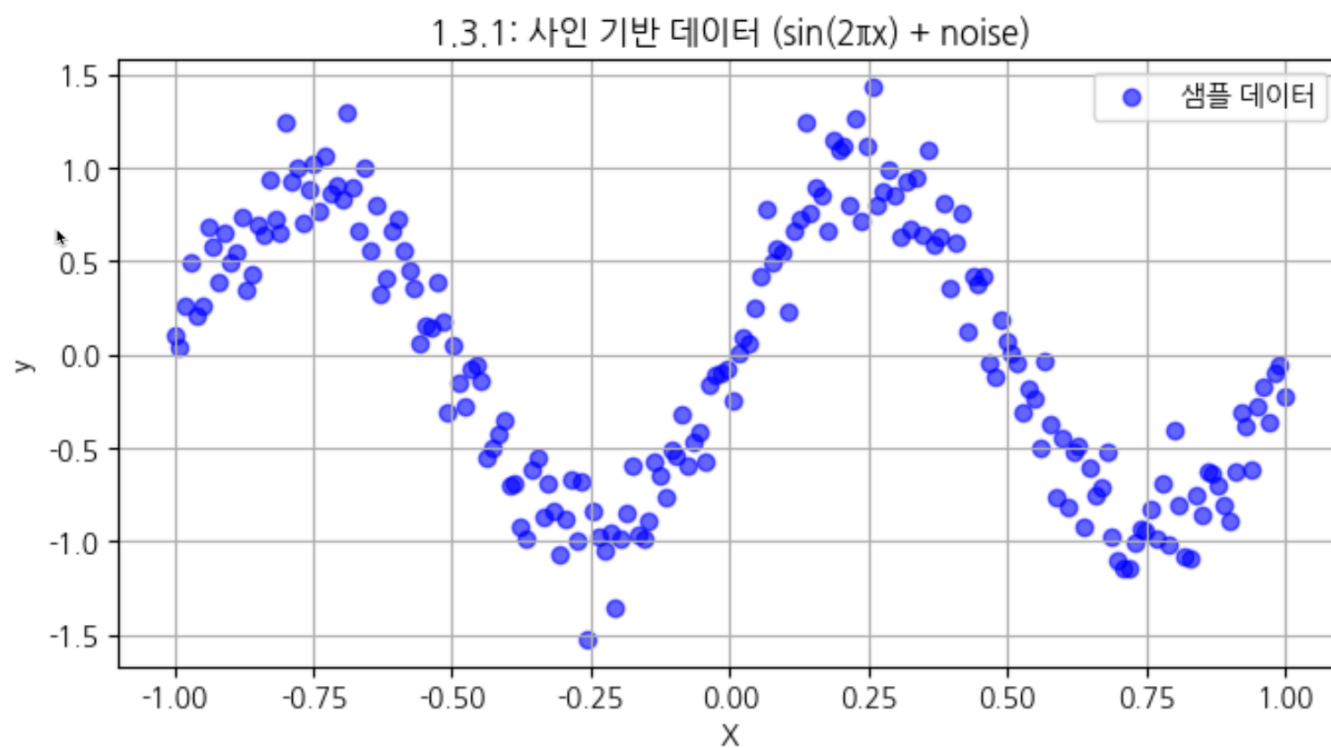
1.3 코드

```
# 1.3.1: 사인 함수 기반 고차 다항 데이터 생성
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split

np.random.seed(42)
# 입력은 [-1,1] 균등 분포
X = np.linspace(-1, 1, 200).reshape(-1, 1)
# 실제 함수는  $\sin(2\pi x)$ , 여기에 약간의 노이즈 추가
y = np.sin(2 * np.pi * X).ravel() + 0.2 * np.random.randn(200)

# (훈련/테스트 분할 사용하지 않음)
# X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

plt.figure(figsize=(8, 4))
plt.scatter(X, y, color='blue', alpha=0.6, label='샘플 데이터')
plt.title("1.3.1: 사인 기반 데이터 ( $\sin(2\pi x)$  + noise)")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```



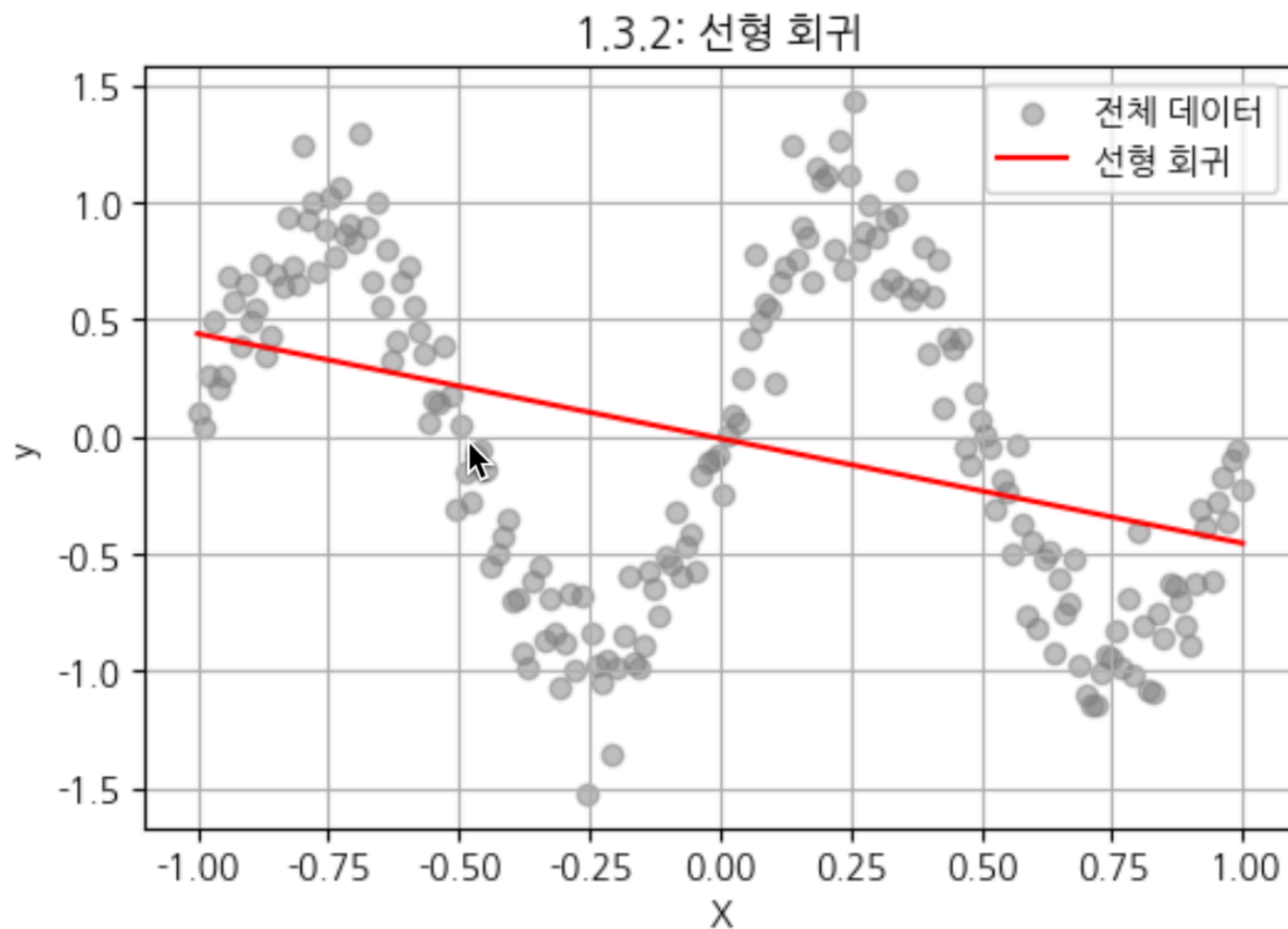
```
# 1.3.2: 선형 회귀
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression()
lin_reg.fit(X, y)

# 정렬된 X 값에 대해 예측
X_sorted = np.sort(X, axis=0)
y_lin = lin_reg.predict(X_sorted)

# 시각화
plt.figure(figsize=(6, 4))
plt.scatter(X, y, color='gray', alpha=0.5, label='전체 데이터')
```

```
plt.plot(X_sorted, y_lin, color='red', label='선형 회귀')
plt.title("1.3.2: 선형 회귀")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```



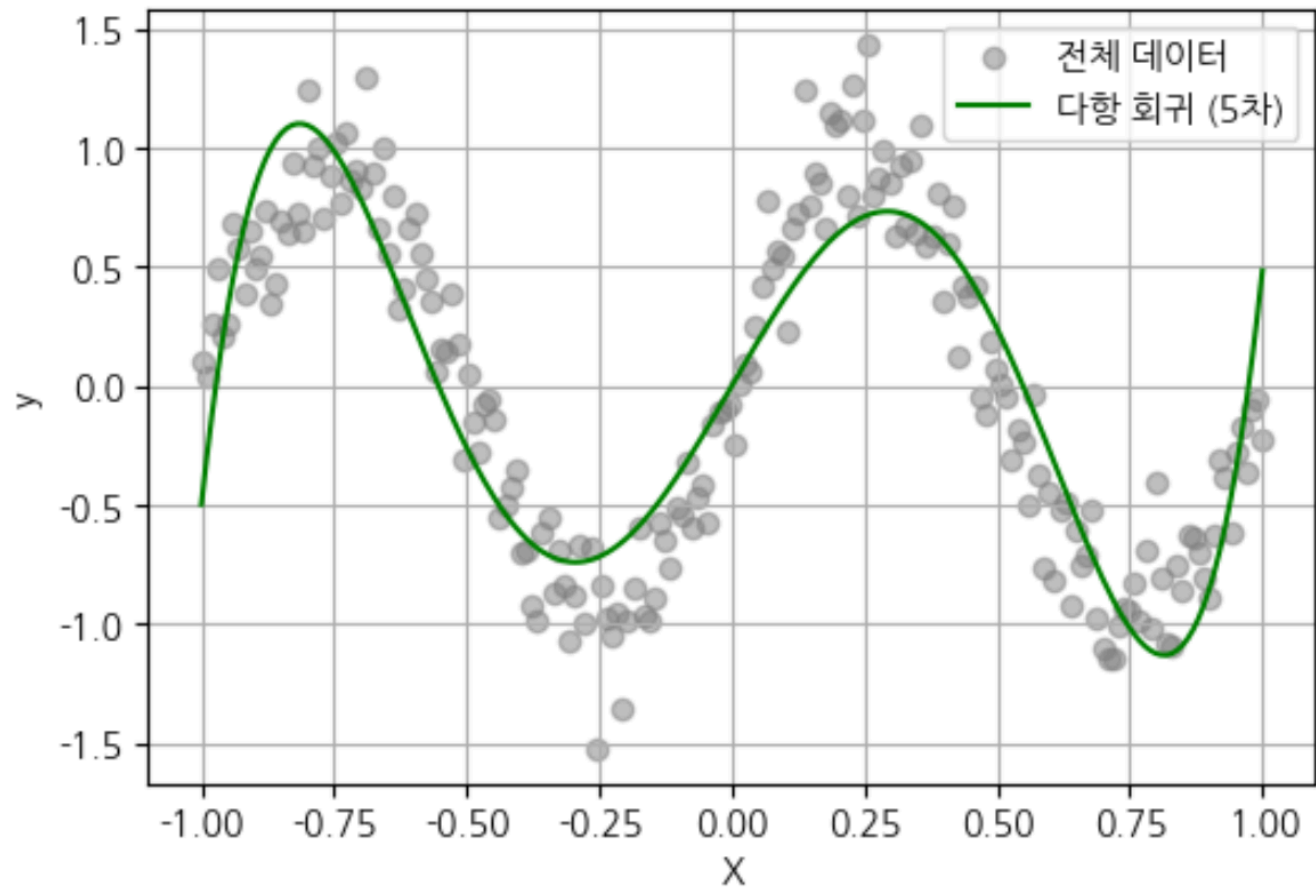
```
# 1.3.3: 다항 회귀 (5차)
from sklearn.preprocessing import PolynomialFeatures

poly = PolynomialFeatures(degree=5)
X_poly = poly.fit_transform(X)
X_sorted_poly = poly.transform(X_sorted)

lin_poly = LinearRegression()
lin_poly.fit(X_poly, y)
y_poly = lin_poly.predict(X_sorted_poly)

# 시각화
plt.figure(figsize=(6, 4))
plt.scatter(X, y, color='gray', alpha=0.5, label='전체 데이터')
plt.plot(X_sorted, y_poly, color='green', label='다항 회귀 (5차)')
plt.title("1.3.3: 다항 회귀")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```

1.3.3: 다항 회귀



1.3.4: 릿지 회귀

```
from sklearn.linear_model import Ridge
```

```
ridge = Ridge(alpha=1.0)
```

```
ridge.fit(X_poly, y)
```

```
y_ridge = ridge.predict(X_sorted_poly)
```

```
# 시각화
```

```
plt.figure(figsize=(6, 4))
```

```
plt.scatter(X, y, color='gray', alpha=0.5, label='전체 데이터')
```

```
plt.plot(X_sorted, y_ridge, color='blue', label='릿지 회귀 (L2)')
```

```
plt.title("1.3.4: 릿지 회귀")
```

```
plt.xlabel("X")
```

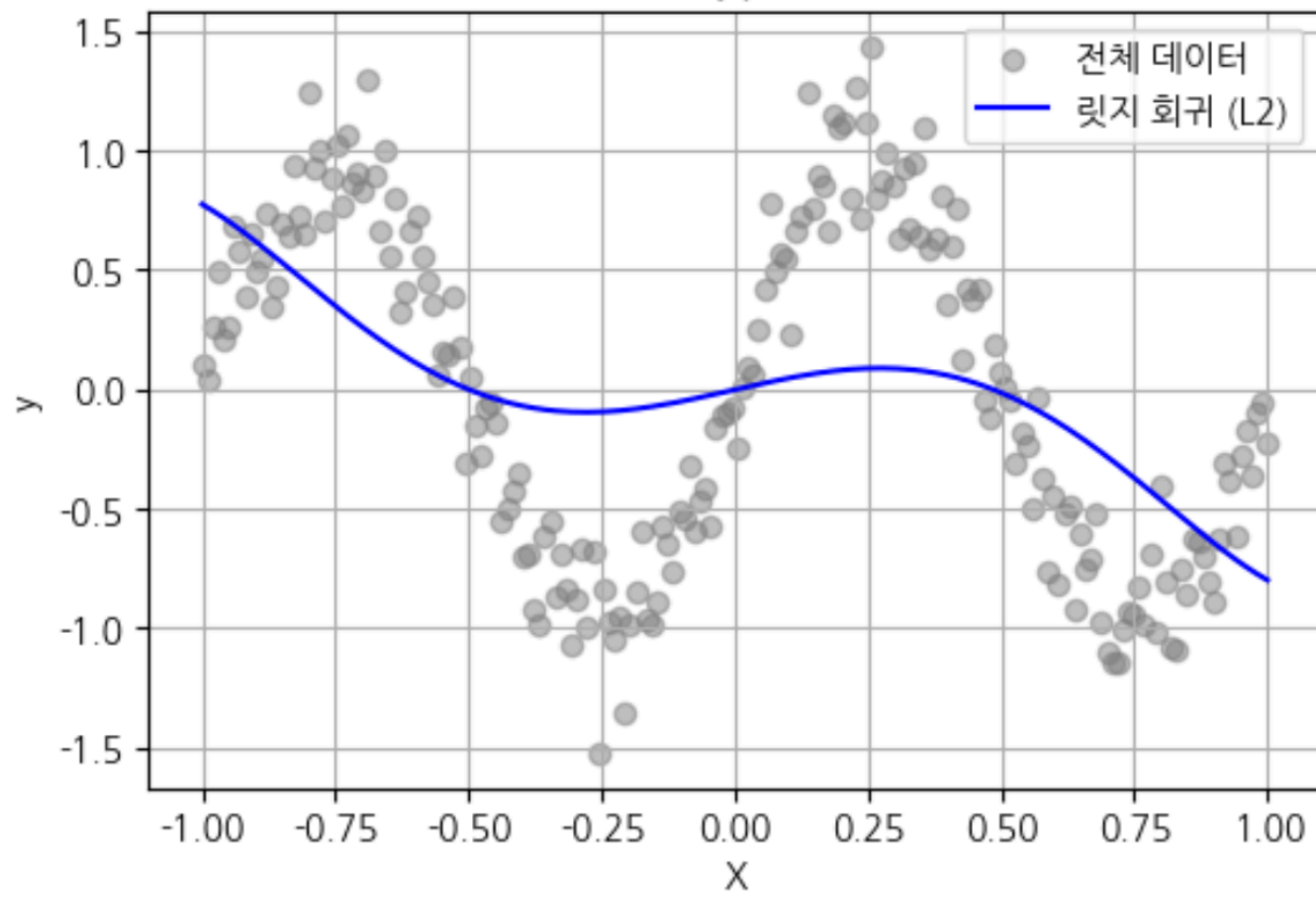
```
plt.ylabel("y")
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```

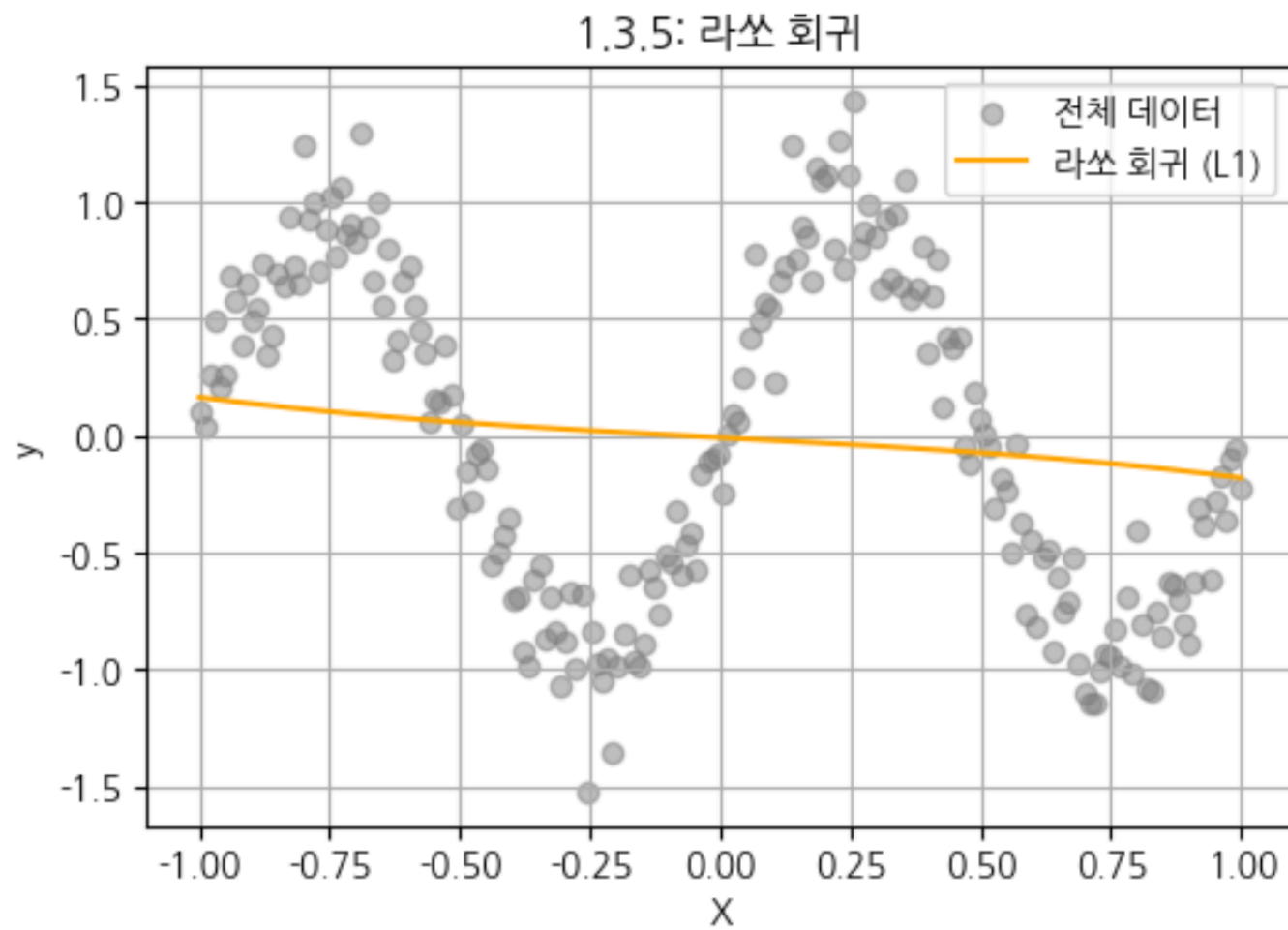
1.3.4: 릿지 회귀



```
# 1.3.5: 라쏘 회귀
from sklearn.linear_model import Lasso

lasso = Lasso(alpha=0.1, max_iter=10000)
lasso.fit(X_poly, y)
y_lasso = lasso.predict(X_sorted_poly)

# 시각화
plt.figure(figsize=(6, 4))
plt.scatter(X, y, color='gray', alpha=0.5, label='전체 데이터')
plt.plot(X_sorted, y_lasso, color='orange', label='라쏘 회귀 (L1)')
plt.title("1.3.5: 라쏘 회귀")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```



1.3.6: 네 가지 회귀 모델 통합 비교

```
plt.figure(figsize=(10, 6))
plt.scatter(X, y, color='gray', alpha=0.5, label='전체 데이터')
plt.plot(X_sorted, y_lin, color='red', label='선형 회귀')
plt.plot(X_sorted, y_poly, color='green', linestyle='--', label='다항 회귀 (5차)')
plt.plot(X_sorted, y_ridge, color='blue', linestyle='--', label='릿지 회귀')
plt.plot(X_sorted, y_lasso, color='orange', linestyle='-', label='라쏘 회귀')
plt.title("1.3.6: 회귀 모델 통합 비교")
plt.xlabel("X")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```

