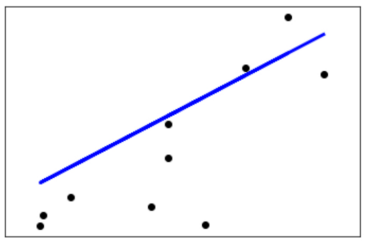
In [17]:
```
print(__doc__)

#kutuphaneleri importlanmasi
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

# diabetes datasetin yüklenmesi
diabetes = datasets.load_diabetes()


# Sadece 1 feature kullanıyorum
diabetes_X = diabetes.data[:, np.newaxis, 2]

# Datayı training/testing sets olmak üzere ikiye ayırıyorum
diabetes_X_train = diabetes_X[:-30]
diabetes_X_test = diabetes_X[-10:]

# Outputu training/testing sets olarak ayırdım
diabetes_y_train = diabetes.target[:-30]
diabetes_y_test = diabetes.target[-10:]

# Linear Regression modeli oluşturdum
regr = linear_model.LinearRegression()

# Modelimi train ettim
regr.fit(diabetes_X_train, diabetes_y_train)

# Test setimi kullanarak predit ediyorum
diabetes_y_pred = regr.predict(diabetes_X_test)

#Coefficient değerlerim
print('Coefficients: \n', regr.coef_)
#Mean squred error saptandı
print("Mean squared error: %.2f"
      % mean_squared_error(diabetes_y_test, diabetes_y_pred))
# Predicition doğruluk oranım
print('Variance score: %.2f' % r2_score(diabetes_y_test, diabetes_y_pred))

# Değerler grafikle gösterildi
plt.scatter(diabetes_X_test, diabetes_y_test,  color='black')
plt.plot(diabetes_X_test, diabetes_y_pred, color='blue', linewidth=3)

plt.xticks(())
plt.yticks(())

plt.show()
```

```
Automatically created module for IPython interactive environment
Coefficients:
 [941.43097333]
Mean squared error: 2154.74
Variance score: 0.39
```



In [10]:
```
print(diabetes.feature_names)
```

```
['age', 'sex', 'bmi', 'bp', 's1', 's2', 's3', 's4', 's5', 's6']
```

In [11]:
```
print(diabetes.data.shape)
```

```
(442, 10)
```

In [18]:
```
dib = pd.DataFrame(diabetes.data)
print(dib.describe())
```

```
                  0             1             2             3             4  \
count  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+02
mean  -3.634285e-16  1.308343e-16 -8.045349e-16  1.281655e-16 -8.835316e-17
std    4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-02
min   -1.072256e-01 -4.464164e-02 -9.027530e-02 -1.123996e-01 -1.267807e-01
25%   -3.729927e-02 -4.464164e-02 -3.422907e-02 -3.665645e-02 -3.424784e-02
50%    5.383060e-03 -4.464164e-02 -7.283766e-03 -5.670611e-03 -4.320866e-03
75%    3.807591e-02  5.068012e-02  3.124802e-02  3.564384e-02  2.835801e-02
max    1.107267e-01  5.068012e-02  1.705552e-01  1.320442e-01  1.539137e-01

                  5             6             7             8             9
count  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+02  4.420000e+02
mean   1.327024e-16 -4.574646e-16  3.777301e-16 -3.830854e-16 -3.412882e-16
std    4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-02  4.761905e-02
min   -1.156131e-01 -1.023071e-01 -7.639450e-02 -1.260974e-01 -1.377672e-01
25%   -3.035840e-02 -3.511716e-02 -3.949338e-02 -3.324879e-02 -3.317903e-02
50%   -3.819065e-03 -6.584468e-03 -2.592262e-03 -1.947634e-03 -1.077698e-03
75%    2.984439e-02  2.931150e-02  3.430886e-02  3.243323e-02  2.791705e-02
max    1.987880e-01  1.811791e-01  1.852344e-01  1.335990e-01  1.356118e-01
```