Семинар 3

[1]:
```
import numpy as np
import scipy.linalg as sl
import sklearn.decomposition as sk
```

Функция для расчета QR разложения методом Грама-Шмидта.

[2]:
```
def QR(A: np.ndarray) -> tuple[np.ndarray, np.ndarray]:
    r, c = A.shape

    # assert r ≤ c

    Q = np.empty((r, min(r, c)))
    u = np.empty(r)

    for i in range(min(r, c)):
        u[:] = A[:, i]
        for j in range(i):
            u[:] -= (A[:, i] @ Q[:, j]) * Q[:, j]  # Внимание
        Q[:, i] = u / np.linalg.norm(u)

    R = Q.T @ A

    return Q, R
```

Функция для расчета QR методом отражений Хаусхольдера.

[3]:
```
def QRH(A):
    r, c = A.shape

    M = np.copy(A)
    Q = np.eye(r)

    e = np.zeros((r, 1))
    e[0, 0] = 1

    v = np.zeros((r, 1))

    for i in range(min(r, c)):
        v[:, :] = M[:, [0]] - np.linalg.norm(M[:, [0]]) * e[:
↪, [0]]
```

```
        v = v / sl.norm(v)

        # Матрица Хаусхольдера
        H = np.eye(r - i) - 2 * v @ v.T

        # Расширяем матрицу H и умножаем ее слева на Q
        Q = Q @ sl.block_diag(np.eye(i), H)

        # Домножаем слева и отбрасываем первый столбец и
строку
        M = (H @ M)[1:, 1:]

        # Укорачиваем e и v
        e = e[:-1, :]
        v = v[:-1, :]

    R = Q.T @ A

    return Q, R
```

Создадим матрицу 10×5, заполненную случайными целыми числами $\in [0, 50)$.

[4]:
```
X = np.random.randint(50, size=50).reshape((10, 5))
```

[5]:
```
X
```

[5]:
```
array([[43, 12,  5, 32, 46],
       [16, 47,  7, 42, 20],
       [38,  4, 16, 32, 40],
       [23,  2,  7, 33, 11],
       [28, 35, 27, 34,  3],
       [20, 17, 37, 23,  3],
       [14,  0, 20, 12, 36],
       [37, 14, 16,  9, 27],
       [34, 35, 33,  7, 42],
       [36, 17, 46, 22, 44]], dtype=int32)
```

Сравним результаты применения разных методов QR разложения.

[6]:
```
q, r = np.linalg.qr(X)
```

[7]:
```
q1, r1 = QR(X)
```

[8]:
```
q2, r2 = QRH(X)
```

[9]:
```
q
```

[9]:
```
array([[-4.46393506e-01,  2.20889267e-01, -4.71822176e-01,
        -1.47927626e-08,  1.16892442e-01],
       [-1.66099909e-01, -7.23472058e-01, -3.25148144e-01,
```

```
        -3.16633977e-01,  3.27313964e-01],
       [-3.94487285e-01,  3.20304747e-01, -1.11785652e-01,
        -2.17829776e-01,  9.70858981e-02],
       [-2.38768620e-01,  2.01843921e-01, -1.26154124e-01,
        -4.59064846e-01, -1.99455978e-01],
       [-2.90674841e-01, -3.71102542e-01,  6.31264716e-02,
        -1.52320374e-01, -4.81704714e-01],
       [-2.07624887e-01, -1.13543330e-01,  4.98689731e-01,
        -2.47870114e-01, -3.39814395e-01],
       [-1.45337421e-01,  1.45920417e-01,  2.89306472e-01,
        -1.69626528e-01,  5.59939966e-01],
       [-3.84106040e-01,  1.20469467e-01, -1.62601814e-01,
         4.38575556e-01, -2.96506063e-01],
       [-3.52962307e-01, -3.08565220e-01,  1.28549742e-01,
         5.79352911e-01,  1.78246540e-01],
       [-3.73724796e-01,  5.32228617e-02,  5.13724249e-01,
        -1.18328654e-02,  2.21945201e-01]])
```

[10]:
```
q1
```

[10]:
```
array([[ 4.46393506e-01, -2.20889267e-01, -4.71822176e-01,
         1.47927620e-08,  1.16892442e-01],
       [ 1.66099909e-01,  7.23472058e-01, -3.25148144e-01,
         3.16633977e-01,  3.27313964e-01],
       [ 3.94487285e-01, -3.20304747e-01, -1.11785652e-01,
         2.17829776e-01,  9.70858981e-02],
       [ 2.38768620e-01, -2.01843921e-01, -1.26154124e-01,
         4.59064846e-01, -1.99455978e-01],
       [ 2.90674841e-01,  3.71102542e-01,  6.31264716e-02,
         1.52320374e-01, -4.81704714e-01],
       [ 2.07624887e-01,  1.13543330e-01,  4.98689731e-01,
         2.47870114e-01, -3.39814395e-01],
       [ 1.45337421e-01, -1.45920417e-01,  2.89306472e-01,
         1.69626528e-01,  5.59939966e-01],
       [ 3.84106040e-01, -1.20469467e-01, -1.62601814e-01,
        -4.38575556e-01, -2.96506063e-01],
       [ 3.52962307e-01,  3.08565220e-01,  1.28549742e-01,
        -5.79352911e-01,  1.78246540e-01],
       [ 3.73724796e-01, -5.32228617e-02,  5.13724249e-01,
         1.18328654e-02,  2.21945201e-01]])
```

[11]:
```
q2[:, :5]
```

[11]:
```
array([[ 4.46393506e-01, -2.20889267e-01, -4.71822176e-01,
         1.47927625e-08,  1.16892442e-01],
       [ 1.66099909e-01,  7.23472058e-01, -3.25148144e-01,
         3.16633977e-01,  3.27313964e-01],
       [ 3.94487285e-01, -3.20304747e-01, -1.11785652e-01,
         2.17829776e-01,  9.70858981e-02],
       [ 2.38768620e-01, -2.01843921e-01, -1.26154124e-01,
```

```
         4.59064846e-01, -1.99455978e-01],
       [ 2.90674841e-01,  3.71102542e-01,  6.31264716e-02,
         1.52320374e-01, -4.81704714e-01],
       [ 2.07624887e-01,  1.13543330e-01,  4.98689731e-01,
         2.47870114e-01, -3.39814395e-01],
       [ 1.45337421e-01, -1.45920417e-01,  2.89306472e-01,
         1.69626528e-01,  5.59939966e-01],
       [ 3.84106040e-01, -1.20469467e-01, -1.62601814e-01,
        -4.38575556e-01, -2.96506063e-01],
       [ 3.52962307e-01,  3.08565220e-01,  1.28549742e-01,
        -5.79352911e-01,  1.78246540e-01],
       [ 3.73724796e-01, -5.32228617e-02,  5.13724249e-01,
         1.18328654e-02,  2.21945201e-01]])
```

Проведем полное SVD разложение.

[12]:
```
U, S, Vt = sl.svd(X, True)
```

Проверим свойства матриц U и V.

[13]:
```
np.round(U.T @ U, 3)
```

[13]:
```
array([[ 1.,  0.,  0., -0.,  0.,  0., -0.,  0.,  0., -0.],
       [ 0.,  1., -0., -0., -0.,  0.,  0., -0., -0.,  0.],
       [ 0., -0.,  1.,  0.,  0., -0., -0., -0., -0., -0.],
       [-0., -0.,  0.,  1.,  0.,  0., -0., -0., -0., -0.],
       [ 0., -0.,  0.,  0.,  1., -0.,  0., -0., -0., -0.],
       [ 0.,  0., -0.,  0., -0.,  1., -0., -0., -0.,  0.],
       [-0.,  0., -0., -0.,  0., -0.,  1.,  0., -0.,  0.],
       [ 0., -0., -0., -0., -0., -0.,  0.,  1.,  0.,  0.],
       [ 0., -0., -0., -0., -0., -0., -0.,  0.,  1., -0.],
       [-0.,  0., -0., -0., -0.,  0.,  0.,  0., -0.,  1.]])
```

[14]:
```
np.round(U @ U.T, 3)
```

[14]:
```
array([[ 1.,  0., -0.,  0., -0., -0., -0., -0., -0., -0.],
       [ 0.,  1., -0.,  0.,  0., -0., -0.,  0., -0., -0.],
       [-0., -0.,  1., -0., -0., -0.,  0., -0., -0.,  0.],
       [ 0.,  0., -0.,  1.,  0., -0., -0., -0., -0., -0.],
       [-0.,  0., -0.,  0.,  1., -0., -0., -0., -0., -0.],
       [-0., -0., -0., -0., -0.,  1., -0., -0.,  0.,  0.],
       [-0., -0.,  0., -0., -0., -0.,  1.,  0.,  0.,  0.],
       [-0.,  0., -0., -0., -0., -0.,  0.,  1., -0., -0.],
       [-0., -0., -0., -0., -0.,  0.,  0., -0.,  1.,  0.],
       [-0., -0.,  0., -0., -0.,  0.,  0., -0.,  0.,  1.]])
```

[15]:
```
np.round(Vt.T @ Vt, 3)
```

[15]:
```
array([[ 1.,  0.,  0.,  0., -0.],
       [ 0.,  1.,  0., -0.,  0.],
       [ 0.,  0.,  1., -0., -0.],
```

```
       [ 0., -0., -0.,  1.,  0.],
       [-0.,  0., -0.,  0.,  1.]])
```

[16]:
```
np.round(Vt @ Vt.T, 3)
```

[16]:
```
array([[ 1., -0., -0.,  0., -0.],
       [-0.,  1., -0.,  0., -0.],
       [-0., -0.,  1.,  0., -0.],
       [ 0.,  0.,  0.,  1.,  0.],
       [-0., -0., -0.,  0.,  1.]])
```

[17]:
```
S2, U2 = sl.eig(X @ X.T)
```

[18]:
```
U[:, :5]
```

[18]:
```
array([[-0.37504703,  0.26051046,  0.47806314,  0.11514321,
        0.18442056],
       [-0.3140202 , -0.55734954,  0.31549097,  0.45870302,
       -0.38068833],
       [-0.35140761,  0.25824752,  0.29779016, -0.2188735 ,
       -0.02869124],
       [-0.20044133, -0.07428678,  0.36804894, -0.42704075,
        0.02589251],
       [-0.30194944, -0.53420599, -0.08245567, -0.17798561,
        0.25769657],
       [-0.23874754, -0.29267022, -0.34041616, -0.47188664,
       -0.0197139 ],
       [-0.22037364,  0.32391337, -0.07230314, -0.03448758,
       -0.55544428],
       [-0.27382075,  0.16846077, -0.05716748,  0.09917508,
        0.5935431 ],
       [-0.38161369,  0.08675302, -0.41988659,  0.50342252,
        0.14438654],
       [-0.42393609,  0.19346471, -0.37456557, -0.16507262,
       -0.26667669]])
```

[19]:
```
np.abs(U2[:, :5])
```

[19]:
```
array([[0.37504703, 0.26051046, 0.47806314, 0.11514321, 0.
       18442056],
       [0.3140202 , 0.55734954, 0.31549097, 0.45870302, 0.
       38068833],
       [0.35140761, 0.25824752, 0.29779016, 0.2188735 , 0.
       02869124],
       [0.20044133, 0.07428678, 0.36804894, 0.42704075, 0.
       02589251],
       [0.30194944, 0.53420599, 0.08245567, 0.17798561, 0.
       25769657],
```

```
       [0.23874754, 0.29267022, 0.34041616, 0.47188664, 0.
↪0197139 ],
       [0.22037364, 0.32391337, 0.07230314, 0.03448758, 0.
↪55544428],
       [0.27382075, 0.16846077, 0.05716748, 0.09917508, 0.
↪5935431 ],
       [0.38161369, 0.08675302, 0.41988659, 0.50342252, 0.
↪14438654],
       [0.42393609, 0.19346471, 0.37456557, 0.16507262, 0.
↪26667669]])
```

[20]:
```
S3, V3 = sl.eig(X.T @ X)
```

[21]:
```
Vt.T
```

[21]:
```
array([[-0.5302893 ,  0.17643842,  0.11560333, -0.16916295,
↪ 0.80354372],
       [-0.33972058, -0.63925729, -0.17284378,  0.6629983 ,
↪ 0.08061219],
       [-0.38991472, -0.05680738, -0.78052203, -0.43090124,
↪-0.22326872],
       [-0.42918422, -0.41078386,  0.58070673, -0.4202132 ,
↪-0.36504532],
       [-0.51686614,  0.62309641,  0.10161673,  0.41177924,
↪-0.40584745]])
```

[22]:
```
V3
```

[22]:
```
array([[ 0.5302893 ,  0.17643842, -0.80354372,  0.11560333,
↪-0.16916295],
       [ 0.33972058, -0.63925729, -0.08061219, -0.17284378,
↪ 0.6629983 ],
       [ 0.38991472, -0.05680738,  0.22326872, -0.78052203,
↪-0.43090124],
       [ 0.42918422, -0.41078386,  0.36504532,  0.58070673,
↪-0.4202132 ],
       [ 0.51686614,  0.62309641,  0.40584745,  0.10161673,
↪ 0.41177924]])
```

Проведем PCA.

[23]:
```
pca = sk.PCA().fit(X)
```

[24]:
```
def ourPCA(X):
    r, c = X.shape
    Mm = np.mean(X, axis=0)
    B = X - np.ones((r, 1)) @ Mm.reshape((1, c))
    V = np.linalg.svd(B)[2]
    return V
```

[25]:
```
pca.components_
```

[25]:
```
array([[ 0.30684194, -0.4536664 ,  0.07111556, -0.33171965,
        0.76481336],
       [ 0.04863283,  0.55202639,  0.74836001, -0.35455131,
        0.08457168],
       [ 0.25201221,  0.65996083, -0.39369646,  0.34478741,
        0.47651474],
       [ 0.57618607, -0.21166722,  0.43808048,  0.6461885 ,
       -0.11718582],
       [ 0.71272879,  0.09540627, -0.29662849, -0.47730233,
       -0.4087901 ]])
```

[26]:
```
ourPCA(X)
```

[26]:
```
array([[-0.30684194,  0.4536664 , -0.07111556,  0.33171965,
       -0.76481336],
       [-0.04863283, -0.55202639, -0.74836001,  0.35455131,
       -0.08457168],
       [-0.25201221, -0.65996083,  0.39369646, -0.34478741,
       -0.47651474],
       [ 0.57618607, -0.21166722,  0.43808048,  0.6461885 ,
       -0.11718582],
       [ 0.71272879,  0.09540627, -0.29662849, -0.47730233,
       -0.4087901 ]])
```