a02_1

October 24, 2025

1 1. Dataset Statistics

Report of Jonas Ortner: 2265527 Marmee Pandya: 1963521

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import scipy

%load_ext autoreload
%autoreload 2

from a02_helper import *
from a02_functions import normalize_data
```

[2]:
```
# look some dataset statistics
scipy.stats.describe(X)
```

[2]:
```
DescribeResult(nobs=3065, minmax=(array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 1., 1., 1.]), array([4.5400e+00, 1.4280e+01, 5.1000e+00,
4.2810e+01, 9.0900e+00,
       3.5700e+00, 7.2700e+00, 1.1110e+01, 3.3300e+00, 1.8180e+01,
       2.0000e+00, 9.6700e+00, 5.5500e+00, 5.5500e+00, 2.8600e+00,
       1.0160e+01, 7.1400e+00, 9.0900e+00, 1.8750e+01, 6.3200e+00,
       1.1110e+01, 1.7100e+01, 5.4500e+00, 9.0900e+00, 2.0000e+01,
       1.4280e+01, 3.3330e+01, 4.7600e+00, 1.4280e+01, 4.7600e+00,
       4.7600e+00, 4.7600e+00, 1.8180e+01, 4.7600e+00, 2.0000e+01,
       7.6900e+00, 6.8900e+00, 7.4000e+00, 9.7500e+00, 4.7600e+00,
       7.1400e+00, 1.4280e+01, 3.5700e+00, 2.0000e+01, 2.1420e+01,
       1.6700e+01, 2.1200e+00, 1.0000e+01, 4.3850e+00, 9.7520e+00,
       4.0810e+00, 3.2478e+01, 6.0030e+00, 1.9829e+01, 1.1025e+03,
       9.9890e+03, 1.5841e+04])), mean=array([1.10818923e-01, 2.28486134e-01,
2.74153344e-01, 6.29690049e-02,
       3.17787928e-01, 9.57553018e-02, 1.13546493e-01, 1.07216966e-01,
       8.89233279e-02, 2.41719413e-01, 5.81305057e-02, 5.37432300e-01,
       9.26231648e-02, 4.96639478e-02, 5.07210440e-02, 2.35334421e-01,
```

1.47197390e-01, 1.86600326e-01, 1.66121044e+00, 7.63066884e-02,
8.19592170e-01, 1.22727569e-01, 1.02006525e-01, 8.90799347e-02,
5.29800979e-01, 2.62071778e-01, 7.71507341e-01, 1.14323002e-01,
1.09487765e-01, 9.92952692e-02, 6.28156607e-02, 4.90342577e-02,
9.27471452e-02, 4.96019576e-02, 1.02156607e-01, 9.93050571e-02,
1.43285481e-01, 1.24274062e-02, 7.55921697e-02, 6.60456770e-02,
4.63360522e-02, 1.32176183e-01, 4.88580750e-02, 7.11876020e-02,
3.06590538e-01, 1.79794454e-01, 5.28874388e-03, 3.13768352e-02,
3.79543230e-02, 1.38396411e-01, 1.81830343e-02, 2.65470799e-01,
7.91275693e-02, 5.34218597e-02, 4.90062936e+00, 5.26750408e+01,
2.82203915e+02]), variance=array([1.07094140e-01, 1.88742036e+00,
2.34317437e-01, 1.78161723e+00,
4.40325719e-01, 6.79193461e-02, 1.39844435e-01, 1.72001423e-01,
6.97247542e-02, 4.69800274e-01, 3.58302179e-02, 7.59167719e-01,
9.28365241e-02, 8.26118648e-02, 7.00470321e-02, 4.29393369e-01,
2.00636301e-01, 2.92991898e-01, 3.18992370e+00, 1.65626303e-01,
1.44315254e+00, 1.01505046e+00, 1.19749530e-01, 1.43862796e-01,
2.45800502e+00, 7.38036013e-01, 1.13920029e+01, 2.31010973e-01,
4.31507668e-01, 1.90528093e-01, 1.24671084e-01, 1.07425177e-01,
2.95159161e-01, 1.07745599e-01, 3.08154062e-01, 1.67896547e-01,
1.85791650e-01, 4.34829439e-02, 1.42525114e-01, 1.16865102e-01,
1.50361473e-01, 6.09903912e-01, 5.73945833e-02, 3.19259425e-01,
1.01935877e+00, 8.17471270e-01, 4.63438951e-03, 7.50333517e-02,
5.54612799e-02, 7.77968333e-02, 1.48045497e-02, 7.59181612e-01,
6.74541224e-02, 2.69600271e-01, 7.42311765e+02, 4.86573219e+04,
3.68952901e+05]), skewness=array([5.92257918, 9.5555492 , 2.94110789,
27.15035267, 4.22000271,
4.55490419, 6.21454549, 10.63604439, 4.44795353, 9.63368819,
5.1601559 , 3.12797362, 7.99555783, 10.07103212, 6.44051978,
5.9017492 , 5.71193665, 5.63845456, 1.6918398 , 8.05102821,
2.36131511, 9.70708774, 5.74851972, 13.62929854, 5.51200726,
5.77490458, 5.72163481, 5.84582426, 11.30526457, 6.67894971,
8.78006633, 10.35563132, 16.1291286 , 10.31146394, 17.98980105,
7.86085564, 5.29526945, 27.69555992, 10.51869112, 9.12514394,
12.60532735, 9.42688905, 7.88762618, 19.69945392, 9.63372543,
8.97501221, 18.94255005, 20.98217881, 14.12336521, 16.36382061,
21.32440567, 21.32959254, 10.88427173, 26.25786993, 27.34951229,
31.14016596, 9.80477376]), kurtosis=array([51.71558405,   93.89016173,
13.18839908, 785.40163828,
28.69487647,   31.20576951,   66.53150801, 198.68010939,
28.29530115, 185.40607771,   34.48800593,   15.18712484,
109.66544541, 138.05561341,   44.19188958,   55.62892   ,
47.49151277,   52.75647121,   6.32523058,   77.87379384,
8.48736408, 103.7022867 ,   49.37553046, 272.09125904,
42.43992409,   49.41302953,   33.63974328,   39.86629858,
166.19735746,   53.12216402,   91.72439904, 124.79234055,
433.42661801, 123.97955409, 555.16708959,   86.72460731,

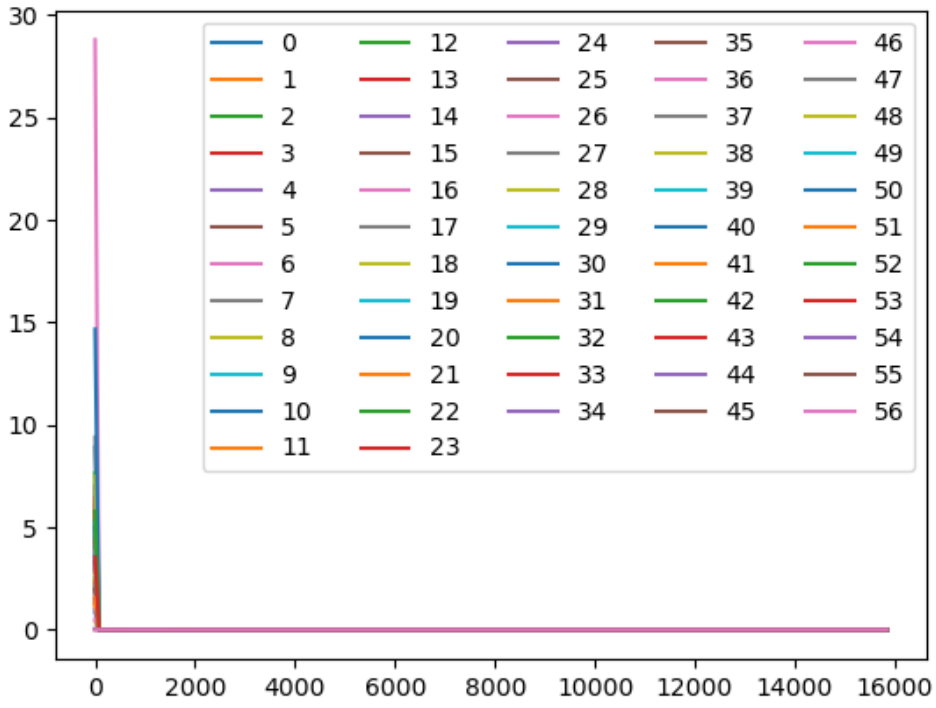
```
 43.92486688,  865.39968623,  181.33012173,  100.87592785,
189.11563172,  111.21705016,   81.96093958,  567.75150773,
147.5283386 ,  107.79164424,  445.8361165 ,  634.57001982,
228.75884956,  499.07842266,  588.19774644,  688.05527222,
184.31757803,  851.48819158,  954.59095344, 1348.49464105,
183.78053905]))
```

[3]:
```
scipy.stats.describe(y)
```

[3]:
```
DescribeResult(nobs=3065, minmax=(0, 1), mean=0.39738988580750406,
variance=0.23954932085067235, skewness=0.41936632478193103,
kurtosis=-1.824131885638896)
```

[18]:
```
# plot the distribution of all features
nextplot()
densities = [scipy.stats.gaussian_kde(X[:, j]) for j in range(D)]
xs = np.linspace(0, np.max(X), 200)
for j in range(D):
    plt.plot(xs, densities[j](xs), label=j)
plt.legend(ncol=5)
```

[18]:
```
<matplotlib.legend.Legend at 0xffff7c03bf10>
```



[5]:
```
# this plots is not really helpful; go now explore further
# YOUR CODE HERE
```

[7]:
```
# Let's compute z-scores; create two new variables Xz and Xtestz by completing␣
 ↪the
# `normalize` function in `a02_functions.py`. Once you implemented this␣
 ↪function, Xz and
# Xtestz will be automatically provided to you in subsequent notebooks.
Xz, Xtestz = normalize_data(X, Xtest)
assert Xz.shape == X.shape
assert Xtestz.shape == Xtest.shape
```

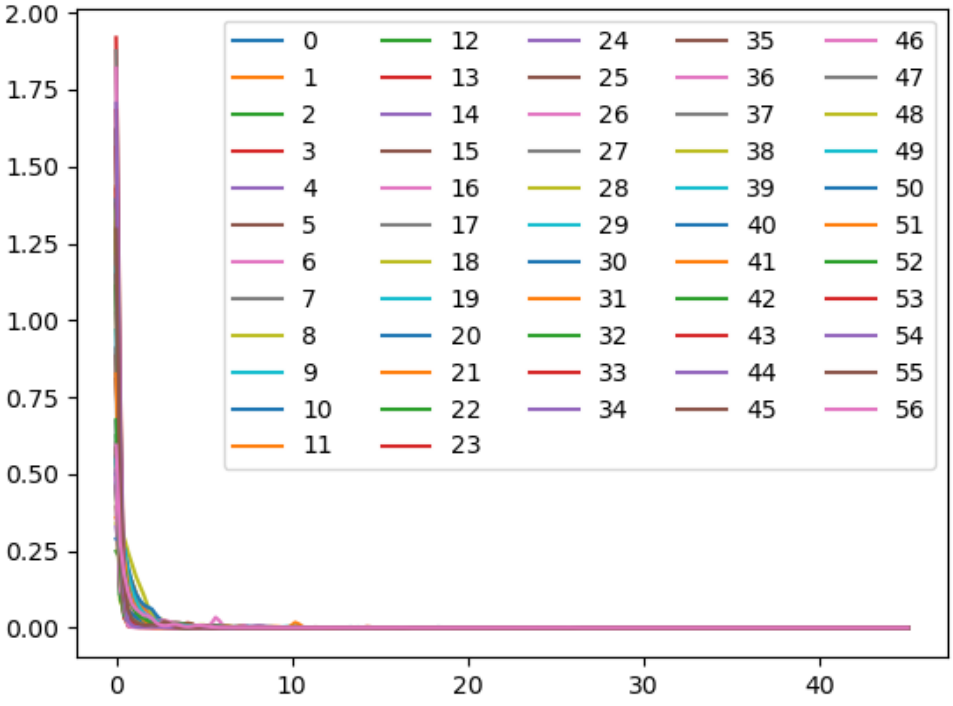
[ ]:
```
# Let's check.
print('mean:', np.mean(Xz, axis=0))  # should be all 0
print(f'var: {np.var(Xz, axis=0)}')  # should be all 1
print(f'mean: {np.mean(Xtestz, axis=0)}')  # what do you get here?
print(f'var: {np.var(Xtestz, axis=0)}')

print(f'sum: {np.sum(Xz**3)}')  # should be: 1925261.15
```

```
mean: [ 1.85459768e-17  9.27298839e-18 -5.56379304e-17 -9.27298839e-18
  5.56379304e-17  3.70919536e-17  0.00000000e+00 -7.41839072e-17
  5.56379304e-17  0.00000000e+00 -1.85459768e-17 -2.43415945e-17
 -4.63649420e-17  1.85459768e-17  1.85459768e-17  3.70919536e-17
 -3.70919536e-17 -9.27298839e-17 -1.66913791e-16  9.27298839e-18
  1.85459768e-17  9.27298839e-18 -5.56379304e-17 -1.85459768e-17
 -6.49109188e-17 -3.70919536e-17 -1.85459768e-17  1.85459768e-17
 -2.78189652e-17  4.63649420e-17 -1.85459768e-17  5.56379304e-17
  0.00000000e+00 -1.85459768e-17  3.70919536e-17  1.85459768e-17
 -9.27298839e-18  4.63649420e-18  1.85459768e-17  9.27298839e-18
  2.31824710e-17 -2.78189652e-17 -9.27298839e-18  4.63649420e-18
 -9.27298839e-18 -9.27298839e-18  1.39094826e-17 -2.78189652e-17
 -3.70919536e-17 -6.49109188e-17  4.63649420e-18  3.70919536e-17
 -3.70919536e-17  9.27298839e-18 -9.27298839e-18  9.27298839e-18
 -7.41839072e-17]
var: [1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1.
 1. 1. 1. 1. 1. 1. 1. 1. 1.]
mean: [-5.73600192e-02 -3.37389835e-02  4.02481250e-02  5.51233798e-03
 -2.51229644e-02  1.67364997e-03  5.29785531e-03 -1.38875040e-02
  1.29802458e-02 -1.00804532e-02  2.68026912e-02  1.46804853e-02
  1.28455840e-02  9.34193448e-02 -1.71666713e-02  6.17841473e-02
 -3.08405298e-02 -1.02710095e-02  1.49139906e-03  6.82438979e-02
 -2.45179646e-02 -4.53675036e-03 -3.12737328e-03  4.09841941e-02
  3.76515934e-02  1.15494599e-02 -3.73018154e-03  6.55839018e-02
 -4.82178216e-02  2.44089391e-02  1.64408852e-02 -1.81514851e-02
```

```
 2.47142980e-02 -1.61248615e-02  1.75684573e-02 -1.33686432e-02
-4.40153254e-02  1.11212504e-02  2.40959269e-02 -1.06211719e-02
-2.06246544e-02  6.23149655e-04 -3.45073187e-02  4.24615929e-02
-1.59254291e-02  9.77429328e-05  6.85319587e-03  5.38462415e-03
 7.89156240e-03  6.81007462e-03 -2.97234292e-02  1.23785037e-02
-3.82610483e-02 -5.29891640e-02  3.19860888e-02 -6.82149671e-03
 5.35333143e-03]
var: [0.61068019 0.64746339 1.25293677 1.2774661  1.08119249 1.31173762
 1.28697678 0.80611698 1.33973062 0.65533893 1.40034314 0.93450565
 0.92877323 2.0728468  0.86981179 2.75968123 0.94816223 0.88879741
 0.96502082 2.70171906 0.99741759 1.1098788  1.07414603 2.08336518
 1.40816544 1.19772845 0.9862879  1.76326753 0.44704368 1.28342341
 1.91457064 1.01476883 1.14073258 1.02208023 0.75850361 0.89687605
 0.89454052 1.35876298 1.97554069 1.14319113 0.60370645 0.89279613
 0.61835224 1.633395   1.01236044 1.04674566 1.76525404 1.2642542
 1.20646248 0.81912474 0.42556335 0.62984245 0.68863812 0.05099329
 2.06687781 0.34306778 0.98979083]
sum: 1925261.1560010156
```

[13]:
```
# plot the distribution of all features
nextplot()
densities = [scipy.stats.gaussian_kde(Xz[:, j]) for j in range(D)]
xs = np.linspace(0, np.max(Xz), 200)
for j in range(D):
    plt.plot(xs, densities[j](xs), label=j)
plt.legend(ncol=5)
```

[13]:
```
<matplotlib.legend.Legend at 0xffff7ed7a750>
```


2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
10
20
30
40
0 1 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23
24 25 26 27 28 29 30 31 32 33 34
35 36 37 38 39 40 41 42 43 44 45
46 47 48 49 50 51 52 53 54 55 56

[ ]: