Question 1

implement a FA function with parameters: the dataset and the desired numbers of factors. These part is a bit like the PCA function implemented in the last homework.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

```
import numpy as np
def manual_fa(X, num_factors):
    # Compute the covariance matrix
    cov_mx = np.cov(X.T)

    # Compute the eigenvectors and eigenvalues
    eigvals, eigvecs = np.linalg.eig(cov_mx)

    # Sort the eigenvectors and eigenvalues in descending order
    sort_idx = np.argsort(eigvals)[::-1]
    eigvals = np.real(eigvals[sort_idx])
    eigvecs = np.real(eigvecs[:, sort_idx])

    # Compute the loading matrix A
    loading_matrix = eigvecs[:, :num_factors]

    # Compute the factor matrix F
    factor_matrix = np.dot(X,loading_matrix)

    # Compute the communality factor of (h_i)^2
    communality = np.sum(loading_matrix**2, axis=1)

    # Compute the uniqueness vector of ψ_i
    uniqueness = 1 - communality

    # Compute the vector of the proportions of total variance
contributed by the ith factor
    prop_var = eigvals[:num_factors] / np.sum(eigvals)

    return loading_matrix, factor_matrix, communality, uniqueness,
prop_var

# Just a test case.

from sklearn.datasets import load_iris
iris = load_iris()
X = iris.data
loading_matrix, factor_matrix, communality, uniqueness, prop_va =
```

```
manual_fa(X,2)
print("Loading Matrix:\n", loading_matrix)
print("\nfactor_matrix:\n", factor_matrix)
print("\ncommunality:\n", communality)
print("\nuniqueness:\n", uniqueness)
print("\nprop_va:\n", prop_va)
```

```
Loading Matrix:
 [[ 0.36138659 -0.65658877]
 [-0.08452251 -0.73016143]
 [ 0.85667061  0.17337266]
 [ 0.3582892   0.07548102]]

factor_matrix:
 [[ 2.81823951 -5.64634982]
 [ 2.78822345 -5.14995135]
 [ 2.61337456 -5.18200315]
 [ 2.75702228 -5.0086536 ]
 [ 2.7736486  -5.65370709]
 [ 3.2215055  -6.06828303]
 [ 2.68182738 -5.23749119]
 [ 2.87622016 -5.49033754]
 [ 2.6159824  -4.74864082]
 [ 2.82960933 -5.21317833]
 [ 2.99541804 -5.97202148]
 [ 2.8896099  -5.34168252]
 [ 2.71625587 -5.09184058]
 [ 2.27856139 -4.81555799]
 [ 2.85761474 -6.50571721]
 [ 3.1163261  -6.66501491]
 [ 2.87883726 -6.13763209]
 [ 2.85406843 -5.63880172]
 [ 3.30254481 -6.19979162]
 [ 2.91437873 -5.84051289]
 [ 3.19210892 -5.71829851]
 [ 2.9586599  -5.75994864]
 [ 2.28642572 -5.46042065]
 [ 3.19963195 -5.42566143]
 [ 3.14661108 -5.28967072]
 [ 2.99569623 -5.1809357 ]
 [ 3.03354506 -5.45790407]
 [ 2.94004523 -5.69467143]
 [ 2.86283042 -5.63899256]
 [ 2.87037575 -5.12999135]
 [ 2.91496666 -5.12263409]
 [ 3.09243264 -5.73787684]
 [ 2.8535028  -6.1403164 ]
 [ 2.90362838 -6.42009834]
 [ 2.86543825 -5.20563023]
 [ 2.63612348 -5.39631705]
```

```
[ 2.87712708 -5.9263226 ]
[ 2.70168102 -5.59559631]
[ 2.52186309 -4.83899423]
[ 2.91235882 -5.55599641]
[ 2.73226271 -5.59048011]
[ 2.65299643 -4.385992  ]
[ 2.50495859 -4.98502652]
[ 3.09675065 -5.51582401]
[ 3.29287589 -5.76361572]
[ 2.78791371 -5.07674437]
[ 2.96421687 -5.83072372]
[ 2.66290296 -5.09900701]
[ 2.95927938 -5.9063626 ]
[ 2.79900535 -5.43465866]
[ 6.78719082 -6.01211305]
[ 6.43485366 -5.64528622]
[ 6.96666745 -5.83121539]
[ 5.68568285 -4.49899357]
[ 6.59046839 -5.40154325]
[ 6.14403422 -4.90870571]
[ 6.5974258  -5.61042085]
[ 4.75324246 -4.32206162]
[ 6.54649696 -5.55531448]
[ 5.49361973 -4.60387067]
[ 4.99452425 -4.06098139]
[ 6.01406369 -5.22297134]
[ 5.76734164 -4.77691611]
[ 6.48729964 -5.20213472]
[ 5.32843976 -5.07209837]
[ 6.43022591 -5.79413207]
[ 6.16264889 -4.97398291]
[ 5.73847013 -4.99334181]
[ 6.44709886 -4.78380703]
[ 5.54759211 -4.7431182 ]
[ 6.61864831 -5.24233572]
[ 5.86025355 -5.25802755]
[ 6.80054901 -4.99916527]
[ 6.42409406 -5.14421478]
[ 6.21721846 -5.47600852]
[ 6.40253951 -5.65545705]
[ 6.83438957 -5.57139345]
[ 7.06016729 -5.59444802]
[ 6.31565578 -5.16360228]
[ 5.19678135 -4.95869039]
[ 5.43423864 -4.62178045]
[ 5.31274266 -4.64666581]
[ 5.63879384 -5.01292014]
[ 6.88239157 -4.90599829]
[ 6.09037158 -4.84266516]
[ 6.30922345 -5.52113489]
```

```
[ 6.72305602 -5.73457217]
[ 6.31746037 -4.95491552]
[ 5.74832281 -5.05842818]
[ 5.66877835 -4.64502585]
[ 5.96716542 -4.65624103]
[ 6.39318033 -5.29248813]
[ 5.73291316 -4.92256673]
[ 4.79783337 -4.31470435]
[ 5.85934663 -4.82204248]
[ 5.83429961 -5.11429789]
[ 5.87858078 -5.03373365]
[ 6.14494114 -5.34469077]
[ 4.59589527 -4.57085921]
[ 5.80136597 -4.97805477]
[ 8.03355786 -5.31710347]
[ 6.91760101 -4.75203623]
[ 8.11904115 -5.67085573]
[ 7.47389619 -5.14722467]
[ 7.85237105 -5.28669163]
[ 8.89940387 -5.87778925]
[ 6.02359738 -4.13419385]
[ 8.4349522  -5.68245258]
[ 7.82359395 -5.08312107]
[ 8.4191161  -6.10974453]
[ 7.16413929 -5.56918098]
[ 7.30576709 -5.11131496]
[ 7.66795693 -5.54322816]
[ 6.84852871 -4.55013423]
[ 7.08829336 -4.78731186]
[ 7.40682151 -5.44620327]
[ 7.45205419 -5.36889584]
[ 8.9894205  -6.50269191]
[ 9.29801055 -5.58427555]
[ 6.80315685 -4.56580294]
[ 7.93018305 -5.70514859]
[ 6.70136624 -4.72086105]
[ 9.00228517 -5.78762668]
[ 6.89113126 -5.12255325]
[ 7.77779564 -5.66194318]
[ 8.11645561 -5.88785393]
[ 6.76087329 -5.14724778]
[ 6.79349719 -5.21028393]
[ 7.62597386 -5.1172231 ]
[ 7.89036815 -5.79159238]
[ 8.34403791 -5.70222174]
[ 8.73303879 -6.70111766]
[ 7.66180278 -5.109675  ]
[ 6.94652637 -5.18353917]
[ 7.28365994 -4.8270509 ]
[ 8.57886506 -6.01503825]
```

```
 [ 7.64660845 -5.46701678]
 [ 7.40746328 -5.3762531 ]
 [ 6.67169147 -5.16196231]
 [ 7.60997628 -5.69924045]
 [ 7.81651984 -5.51060386]
 [ 7.42463293 -5.73615604]
 [ 6.91760101 -4.75203623]
 [ 8.06537851 -5.60481518]
 [ 7.92111132 -5.63175077]
 [ 7.44647493 -5.51448488]
 [ 7.02953175 -4.95163559]
 [ 7.26671085 -5.40581143]
 [ 7.40330675 -5.44358054]
 [ 6.89255399 -5.04429164]]

communality:
 [0.56170908 0.54027978 0.76394261 0.13406853]

uniqueness:
 [0.43829092 0.45972022 0.23605739 0.86593147]

prop_va:
 [0.92461872 0.05306648]
```

question b

On the AutoMPG dataset, compare with the PCA results in HW05.

```
import pandas as pd
df = pd.read_csv('/content/drive/MyDrive/autompg.csv')
df.replace('?', np.nan, inplace=True)
df.dropna(inplace=True)
autompg_variables = df.drop(['mpg','car name'], axis=1)
autompg_mpg = df['mpg']
autompg_variables = pd.DataFrame(autompg_variables, dtype=float)

loading_matrix, factor_matrix, communality, uniqueness, prop_va =
manual_fa(autompg_variables, 2)
print("Loading Matrix:\n", loading_matrix)
print("\nfactor_matrix:\n", factor_matrix)
print("\ncommunality:\n", communality)
print("\nuniqueness:\n", uniqueness)
print("\nprop_va:\n", prop_va)
```

```
Loading Matrix:
 [[ 1.79262233e-03 -1.33244815e-02]
 [ 1.14341275e-01 -9.45778439e-01]
 [ 3.89670355e-02 -2.98248416e-01]
 [ 9.92673415e-01  1.20752748e-01]
 [-1.35283460e-03  3.48258394e-02]
 [-1.33684138e-03  2.38516836e-02]
```

```
 [-5.51538021e-04  3.24298106e-03]]

factor_matrix:
 [[3518.40010719   95.97553085]
 [3712.29658003   67.67321996]
 [3452.93676257   71.36098697]
 [3448.35661164   84.27445271]
 [3463.62306262   91.02829896]
 [4365.86985482   61.3100678 ]
 [4382.4917687    32.63910157]
 [4339.00454868   42.28221344]
 [4453.27940475   38.81035204]
 [3873.69819151   41.239576  ]
 [3587.21916265   19.22119334]
 [3627.5785002    68.35712512]
 [3784.93363611   32.99955947]
 [3124.08970266 -122.87757815]
 [2371.13541624  153.36739271]
 [2838.48088064  128.62752043]
 [2780.10542452  119.960833  ]
 [2594.12155643  100.03068451]
 [2128.80689669  141.34800859]
 [1834.32405843  118.45809332]
 [2668.27984995  194.9003615 ]
 [2427.83081942  167.51630893]
 [2373.08153446  162.32547852]
 [2235.7665555   123.67850684]
 [2654.75648146  106.81631269]
 [4630.62985069  154.72411346]
 [4386.73495867  180.49901592]
 [4394.34045279  167.78524666]
 [4739.50616626  228.53395903]
 [2128.80555985  141.37186028]
 [2266.81815654  116.31609874]
 [2228.19046054  135.96802277]
 [2645.02335491   70.88680157]
 [3443.5165178   173.30903567]
 [3336.9861389   137.87301445]
 [3309.71635228  138.19167123]
 [3294.22838595  149.94616366]
 [4224.51404869  130.02290277]
 [4483.75317992  110.5260346 ]
 [4169.56171848  126.06694291]
 [4108.09717367  151.15130432]
 [4969.40685997  184.40262548]
 [4763.49157123  146.08696467]
 [5154.79973175  192.17230548]
 [2973.98210198   82.93839241]
 [2409.05698667  139.19485644]
 [3290.33116483  132.18022235]
```

```
[3147.91193854  118.47414739]
[2220.92854446  129.16774461]
[2124.1084922   121.93964796]
[2070.45008641  157.17307596]
[2062.78566354  145.61067559]
[1770.54594813  129.86979833]
[1611.98976405  108.37576063]
[1833.87761593  114.13347567]
[1953.69324408  131.48574049]
[2277.82076518  142.08175064]
[2124.57668469  142.90637771]
[2250.55922323  166.81968848]
[2409.75638005  133.86764956]
[2226.87986602  130.00317739]
[4289.03648379  137.8956831 ]
[4405.32996691  101.02783208]
[4146.80942358  155.90192611]
[4144.74422269  123.05456297]
[3685.60356046  113.16465004]
[4656.11613571   93.6699154 ]
[4514.9743227   168.46203267]
[4469.50618081  161.41616416]
[4442.64271237  101.03937045]
[2324.60195502  188.3765424 ]
[3903.99035883  139.76508054]
[4108.04273609  167.82001848]
[4302.42198365  193.30361689]
[4089.23368912  148.91567962]
[2929.60092589  208.50028148]
[2509.28519674  168.40145504]
[2974.16862162  222.63102594]
[2186.51305613  155.25126992]
[2394.64234949  150.39297896]
[2285.79910856  159.36871715]
[2505.02997953  162.36160554]
[2162.35815075  146.95244125]
[2099.0213149   137.84278118]
[4116.69829034  113.96089823]
[3685.60222362  113.18850172]
[4004.34985684  109.38404288]
[4052.15207449  163.74028245]
[3791.43235716  112.661468  ]
[4972.38727134  135.21379092]
[4482.77565393  118.04736128]
[4377.22330004  149.84331571]
[4248.05942219  168.27738397]
[4758.89801042   93.51924569]
[4975.42025745  102.43267862]
[3840.88852609   70.74344535]
[3127.84234545  134.99219087]
```

```
[3286.35373899  131.84939224]
[2953.73805466  108.59308723]
[3030.77109627  104.34267812]
[2908.95600613  137.28993354]
[1948.47681417  132.43362736]
[5011.86787823  182.47822789]
[4922.19906636  166.36726595]
[4667.58793957  172.89137857]
[4512.97049434  160.63258987]
[2798.88235482   89.72083263]
[2276.70513718  159.56843944]
[2402.10492267  138.41470349]
[2377.46741696  159.36359637]
[2119.83712553  165.61306681]
[2310.22142329  140.53816004]
[2475.67472631  122.14458641]
[2263.00527118  156.21049227]
[4097.66115781  120.73480124]
[4301.25914375   71.6404122 ]
[1862.88795414  148.89173996]
[2158.26285324  130.73966252]
[2579.55320028  179.00617134]
[2865.07446426  200.71003035]
[3416.20383571   66.96469033]
[2658.52248922  176.13771674]
[2808.91874033  157.16632551]
[3684.09022244   59.75180615]
[3105.50332895  161.24024234]
[2910.05908757  103.30381798]
[3343.92881302  138.8420775 ]
[1947.23779625  143.15186551]
[2449.9949084   159.00973965]
[1833.07765705  137.61842821]
[2542.19082048  154.48288286]
[3785.6684825   192.57705056]
[3639.06319145  164.07100521]
[3616.23632857  194.42639479]
[4150.54298321  174.80649807]
[4710.33212368  193.8240954 ]
[4466.44758939  194.83201448]
[4643.89896458  234.89026573]
[4266.30942295  183.99201461]
[2217.06683519  152.80228228]
[1960.1478371   144.59651783]
[2297.16691098  164.95062374]
[1647.5189512   114.02928518]
[2000.07305011  147.55813792]
[2122.53232494  151.3308441 ]
[2105.65497252  149.3161162 ]
[2245.61880613  141.33751731]
```

```
[2488.15118138  160.37392511]
[2389.34054689  161.09990361]
[1996.87607702  149.08178244]
[3269.40297611  155.2726085 ]
[3466.22249418  152.19244932]
[3438.12763565  158.94845203]
[3166.13714931  125.8099602 ]
[4686.05837395  136.7462441 ]
[4453.0342137   164.04681439]
[4507.14450971  199.84155469]
[4668.67626346  188.3922242 ]
[3908.95577068  222.94235809]
[3901.01006767  205.16921775]
[3736.3424964   175.96345208]
[3786.5817666   218.28926797]
[3047.32336386  117.92001739]
[3231.54011933  110.49899828]
[3185.2371557    60.66975914]
[2168.99122177  150.34759786]
[2638.79054113  163.83376382]
[2922.96250512  104.89745539]
[2591.93802622  159.70186548]
[2701.1531219   173.12322372]
[2219.65385738  164.45435346]
[2543.62250881  168.17536082]
[2985.35995451  171.88513905]
[1935.71367586  130.13025121]
[3217.39548606  143.77833168]
[2690.99880561  190.47423627]
[2952.3681451   219.66070751]
[2940.96345784  214.19634733]
[2669.63473469  176.00504535]
[1794.20066621  117.23293898]
[2461.41647349  172.99283198]
[2220.03650266  136.55715366]
[2572.63362979  153.00981135]
[2252.64346442  158.42862628]
[2200.53336068  147.9180626 ]
[4224.34251887  180.91774366]
[4201.40179043  162.62132109]
[3972.30124545  157.30938831]
[4230.07009251  133.82598932]
[3238.8244103   150.0409874 ]
[3360.99980465  139.36427091]
[3015.8417039   152.74242771]
[3092.31648728  128.60823256]
[2031.71067893  152.36756483]
[2161.56385755  153.26007976]
[1935.71206846  130.16106807]
[1794.19946465  117.25330808]
```

```
[3653.75878607  200.59573569]
[3579.31972589  174.3297035 ]
[3651.05297706  173.19203531]
[3202.69265381  115.57501307]
[1825.37470765  109.64841107]
[1987.74771358  141.39061126]
[2153.10656916  148.45333591]
[2564.90733386  158.08423684]
[3145.6434872   229.31130384]
[3953.23316623  132.44009914]
[3263.06695813  257.65081607]
[2930.46531563  176.33559812]
[3815.78328245  268.91577002]
[4394.83888862  146.32063752]
[4070.85631793  117.51120624]
[3881.134791    145.1475166 ]
[3769.58750227  110.12870135]
[2043.74988779  136.40947999]
[2154.90411296  133.68284006]
[1822.79999826  130.79638943]
[2300.72203768  136.04082919]
[1943.07634365  135.9736241 ]
[3891.99109974  138.98076961]
[4064.15431732  213.9413566 ]
[4151.5710007   158.12315552]
[4303.01979062  196.48780379]
[3526.96719283  158.1287586 ]
[3430.29522263  166.13569284]
[3632.91130752  198.08377966]
[3531.45943811  162.40205052]
[4241.72822203   79.70030166]
[4181.02404427  123.34075435]
[4346.3471128    89.43516452]
[4349.07036449  149.29411999]
[1939.80047122  121.55118935]
[2740.50182436  164.14786403]
[2262.29687282  161.82267643]
[2754.17340915  176.0575045 ]
[2049.51422543  138.56651384]
[2074.1192162   135.46130307]
[1984.03901556  130.33520768]
[2187.96936599  151.72544613]
[2814.73583881  175.17657486]
[2598.95836038  168.94616756]
[2713.38788671  222.25435892]
[1982.4905682   142.82059714]
[1800.47228001  107.29613323]
[1981.27663045  152.90705346]
[2067.15674852  151.15425585]
[1799.43428012  115.78221044]
```

```
[3374.24969228  129.92015767]
[3749.3427722   110.71597323]
[3583.6837725   106.20854284]
[3539.48484987  179.52244628]
[3158.33605529  165.90319179]
[2969.34168297  145.85907173]
[2719.3900769   172.13932807]
[3434.37596418  173.93966873]
[3216.39853155  143.73609915]
[3385.62507025  160.68736242]
[3073.57117395  158.56945357]
[3623.37155405  193.95244554]
[3419.07793963  134.27582269]
[3440.32241229   84.08649884]
[3452.49011113  150.5578041 ]
[3221.36014072   62.07806831]
[4091.81442381  152.39308437]
[2152.94643529  149.63999637]
[2560.14957249  156.370515  ]
[2300.41662324  138.58239307]
[2228.47280523  149.91861284]
[2515.47845714  150.95753682]
[2746.69705358  155.00074228]
[2854.53886455  179.00874838]
[2404.6470612   151.26839682]
[2828.14015122  189.46774153]
[3140.39000946  189.98129445]
[2792.71925133  191.12655805]
[3408.72059144  220.27516606]
[1988.24476452  137.28057521]
[2133.09172864  147.23490994]
[3252.00303393  141.41317856]
[2994.15393469  148.98532414]
[2888.14065015  192.75731609]
[3270.99287968  150.43617784]
[3365.27797038  162.50744836]
[3851.69306073  138.77308422]
[3737.15633286  127.9524502 ]
[3971.42491457  106.69123792]
[3843.4478689   123.77222978]
[4374.00344598  151.53086035]
[4069.85396264  117.49107478]
[3613.87554581  147.81308075]
[3958.03175982   92.78199459]
[1923.72087328  129.42415499]
[1972.77554303  140.13365221]
[1915.17392434  117.03107317]
[2667.27138995  186.46738966]
[3527.93712605  232.73876565]
[3916.20128213  105.01911313]
```

```
[3185.38387839  233.37202203]
[3428.05698854  142.78371196]
[2198.49818946  147.76585056]
[2148.86221891  141.78741706]
[2017.32609408  146.65260769]
[2127.36872476  152.90778647]
[2671.08994573  155.14637818]
[2600.13906714  117.63625558]
[2704.36761113  130.37101551]
[2557.9289644   141.2830525 ]
[2142.3379739   145.91398864]
[1965.96881073  138.09151444]
[2118.27947901  144.83003087]
[2016.45021303  145.51241584]
[2679.02931978  156.15366477]
[2868.28476275  190.39397347]
[3001.64330931  195.52368104]
[3385.34059358  171.10521034]
[2185.97770882  151.61469961]
[2709.84399055  176.18839905]
[2539.89419443  173.56546062]
[2433.23694608  156.31265456]
[2263.55467482  151.38619114]
[2106.7814645   156.55315471]
[2801.29452922  161.60974738]
[2106.66536454  157.54420674]
[2081.7549654   154.95054052]
[2329.92061337  185.20837931]
[2944.70664992  224.33980543]
[3245.36284316  237.00066258]
[1849.34156379  119.68926514]
[2142.86058683  149.78292383]
[1843.95320127  122.51729613]
[2912.91935075  155.3658607 ]
[2414.05012154  198.50553215]
[2498.94096338  162.63823391]
[2288.1378327   156.30822004]
[2490.34320066  150.37008459]
[2636.10935919  152.77115618]
[2622.1054063   143.77533375]
[2728.98734349  134.91833192]
[2386.11628007  137.59353365]
[1753.30926016  122.38314283]
[1873.46604053  128.4396342 ]
[1758.58032612  120.47099219]
[2063.44564739  140.13959046]
[1972.65284617  141.27340255]
[2047.44665721  147.36835591]
[1983.38712871  135.79310629]
[2213.11054793  151.77178799]
```

```
 [2043.63185391  137.31304658]
 [2376.17136003  177.92193359]
 [2188.7227454   145.4511016 ]
 [2208.84304117  145.68655942]
 [2347.92841368  161.72974778]
 [2613.22150412  175.79980122]
 [2632.17143584  185.145427  ]
 [3225.44479708  235.4123659 ]
 [3156.26382253  224.31504378]
 [2902.3661838   158.99595056]
 [2929.78512292  182.27445602]
 [3420.55946441  163.49402158]
 [3741.69925565   89.95577518]
 [3063.75678925  156.55266623]
 [3468.53182934  182.69040071]
 [2602.02004925  184.97623355]
 [2636.7649716   189.16775391]
 [2393.56079673  159.56243503]
 [2572.12754535  182.12998849]
 [2525.08502748  154.63073022]
 [2735.60700148  163.13651356]
 [2863.47688995  188.58571551]
 [1980.25850338  120.15518453]
 [2023.08975222  140.72368476]
 [1968.49352612  134.06138809]
 [2123.76887434  140.92092714]
 [2123.23773729  145.54418449]
 [2161.20090636  123.50390715]
 [2203.87820197  145.11012995]
 [2243.50139785  150.56928554]
 [1963.49470938  133.66532559]
 [1963.49376239  133.68970367]
 [1993.27328841  137.32969905]
 [2948.28374728  154.07389894]
 [3026.05754199   93.39564368]
 [2587.36036466  139.57632182]
 [2845.00131059   91.90069547]
 [2665.55771929  159.37847976]
 [2371.22470672  135.80957669]
 [2949.03273259  189.07397639]
 [2788.79366391  181.2908892 ]
 [2127.37492922  152.71966976]
 [2296.7760946   126.70436438]
 [2622.43889726  182.47447091]
 [2716.74434921  194.02487588]]

communality:
 [1.80755303e-04 9.07570783e-01 9.04705474e-02 9.99981734e-01
 1.21466925e-03 5.70689956e-04 1.08211203e-05]
```

```
uniqueness:
 [9.99819245e-01 9.24292173e-02 9.09529453e-01 1.82657685e-05
 9.98785331e-01 9.99429310e-01 9.99989179e-01]

prop_va:
 [0.99756151 0.0020628 ]
```

We can see that the structure of two model are different. The FA use X=FA, and the PCA use Z=XV.

Question 2:

Transpose the ORL face dataset to be a data matrix of 2576x400. Perform the factore matrix. We take n=400 and see how many variables are needed to get the desired explanation.

```python
from PIL import Image

def edge(datas):
    criteria = []
    element = 0
    for i, nums in enumerate(datas):
        element += nums
        if element >= 0.5 and len(criteria) == 0:
            criteria.append(i+1)
        if element >= 0.6 and len(criteria) == 1:
            criteria.append(i+1)
        if element >= 0.7 and len(criteria) == 2:
            criteria.append(i+1)
        if element >= 0.8 and len(criteria) == 3:
            criteria.append(i+1)
        if element >= 0.9 and len(criteria) == 4:
            criteria.append(i+1)
            break
    return criteria

ORLface_data = []
for i in range(1, 41):
    for j in range(1,11):
        image_dir = f"/content/drive/MyDrive/DA/ORL Faces/{i}_{j}.png"
        img = Image.open(image_dir)
        img_array = np.asarray(img)
        ORLface_data.append(img_array.flatten())
ORLface_data = np.array(ORLface_data)

loading_matrix, factor_matrix, communality, uniqueness, prop_va =
manual_fa(ORLface_data, 400)
print("Loading Matrix:\n", loading_matrix)
print("\nfactor_matrix:\n", factor_matrix)
print("\ncommunality:\n", communality)
```

```python
print("\nuniqueness:\n", uniqueness)
print("\nprop_va:\n", prop_va)

criteria = edge(prop_va)
print("50% explanable:", criteria[0], " principal components")
print("60% explanable:", criteria[1], " principal components")
print("70% explanable:", criteria[2], " principal components")
print("80% explanable:", criteria[3], " principal components")
print("90% explanable:", criteria[4], " principal components")
```

```
Loading Matrix:
 [[ 4.18418191e-03  2.92082392e-02  3.98717037e-02 ...  1.08463577e-02
  -3.88942168e-03 -2.43641364e-03]
 [ 4.34579327e-03  2.92119864e-02  3.97740019e-02 ...  9.40164343e-03
  -1.01740124e-03 -2.45015920e-03]
 [ 3.69863744e-03  2.93623995e-02  4.00714546e-02 ...  1.27420187e-02
   4.05565911e-03  5.34254516e-03]
 ...
 [ 1.53271003e-02 -1.27914953e-02  2.41267237e-02 ... -3.11265398e-02
  -2.33792628e-02 -2.19128626e-06]
 [ 1.33515940e-02 -1.78069997e-02  2.89114944e-02 ... -1.43982079e-03
  -4.05094260e-03  2.61679055e-03]
 [ 1.38193377e-02 -1.75111834e-02  3.08289732e-02 ... -2.35419739e-03
  -4.65625113e-02 -7.79406816e-03]]

factor_matrix:
 [[-3735.25330174  3663.54695712   761.4230817  ...   -15.44696504
     32.79484107    10.95171292]
 [-4020.91757202  3539.56387202   398.21358833 ...   -14.493715
     28.63463334    10.95171292]
 [-4201.5410653   2920.14627233   298.77109127 ...    -8.79447361
     31.76165768    10.95171292]
 ...
 [-4339.84011576  3889.9586488    488.35937662 ...   -10.55792251
     30.33874686    10.95171292]
 [-4618.60206838  4054.94613442   121.88426316 ...    -6.52740545
     28.16675215    10.95171292]
 [-4184.84387619  3653.11249044   746.73153846 ...   -15.23761258
     28.07977455    10.95171292]]

communality:
 [0.04367717 0.03707898 0.0410595  ... 0.23481048 0.21846609
0.24361547]

uniqueness:
 [0.95632283 0.96292102 0.9589405  ... 0.76518952 0.78153391
0.75638453]

prop_va:
 [1.87166976e-01 1.36991737e-01 7.25151882e-02 5.91353234e-02
```

5.41577962e-02	3.55799201e-02	2.58114751e-02	2.46087008e-02
2.06763355e-02	1.89634592e-02	1.51035765e-02	1.46901473e-02
1.17120662e-02	1.10960257e-02	1.01470442e-02	9.49254659e-03
9.08642298e-03	8.71692121e-03	7.62451171e-03	7.18196305e-03
6.89473181e-03	6.62338845e-03	6.08294048e-03	5.56952482e-03
5.44672720e-03	5.01839357e-03	4.90268374e-03	4.85857277e-03
4.53852438e-03	4.41693873e-03	4.30960519e-03	4.02032884e-03
3.97989405e-03	3.82763871e-03	3.72423522e-03	3.50839081e-03
3.41460546e-03	3.35659372e-03	3.16588723e-03	2.99127509e-03
2.93549888e-03	2.86920867e-03	2.79085774e-03	2.74853383e-03
2.61556413e-03	2.53969984e-03	2.50963275e-03	2.42690473e-03
2.40341348e-03	2.32714174e-03	2.28969785e-03	2.12752835e-03
2.07420534e-03	2.07037068e-03	2.03197694e-03	1.94762992e-03
1.91939862e-03	1.88601335e-03	1.82784190e-03	1.79435399e-03
1.75263679e-03	1.70687597e-03	1.67031461e-03	1.61999760e-03
1.56815209e-03	1.54401167e-03	1.50579786e-03	1.46628926e-03
1.43069616e-03	1.39858747e-03	1.38167165e-03	1.33385796e-03
1.32021295e-03	1.30045584e-03	1.28221922e-03	1.26744540e-03
1.26552851e-03	1.23452170e-03	1.21874792e-03	1.19265317e-03
1.17628157e-03	1.16492010e-03	1.14998815e-03	1.12392815e-03
1.09078804e-03	1.08120539e-03	1.06870472e-03	1.04075104e-03
1.02100457e-03	1.01327209e-03	1.00368742e-03	9.75710965e-04
9.59024976e-04	9.52584603e-04	9.34109333e-04	9.17643277e-04
8.98946096e-04	8.90368890e-04	8.64845611e-04	8.45556465e-04
8.38812067e-04	8.21580423e-04	8.17613011e-04	8.02030653e-04
7.77691813e-04	7.70682752e-04	7.63704629e-04	7.57768260e-04
7.42395766e-04	7.37199973e-04	7.29475848e-04	7.23809968e-04
7.11401667e-04	7.03844439e-04	6.99238028e-04	6.76665053e-04
6.68075522e-04	6.58481962e-04	6.47069048e-04	6.41235270e-04
6.37868534e-04	6.30332123e-04	6.22901494e-04	6.18033899e-04
6.13921219e-04	6.05718474e-04	5.99875068e-04	5.91114546e-04
5.84656392e-04	5.78804064e-04	5.73220570e-04	5.65188325e-04
5.54828264e-04	5.49324542e-04	5.41018661e-04	5.38277396e-04
5.30728101e-04	5.23739359e-04	5.23037633e-04	5.09879057e-04
5.06888086e-04	4.95835576e-04	4.89012683e-04	4.84900865e-04
4.81998304e-04	4.77964476e-04	4.72931165e-04	4.71137014e-04
4.64069630e-04	4.59749330e-04	4.55778752e-04	4.51134277e-04
4.45782598e-04	4.43629525e-04	4.41204688e-04	4.37462328e-04
4.33272230e-04	4.25936125e-04	4.19596922e-04	4.16877592e-04
4.10023341e-04	4.07051525e-04	4.05540451e-04	4.00485202e-04
3.98948109e-04	3.94399572e-04	3.90904504e-04	3.87769664e-04
3.85354591e-04	3.80802935e-04	3.71901974e-04	3.66689950e-04
3.63714840e-04	3.59378275e-04	3.57043532e-04	3.55517427e-04
3.52230448e-04	3.47496624e-04	3.45124406e-04	3.42008445e-04
3.40172257e-04	3.36067757e-04	3.32425529e-04	3.28885542e-04
3.23583664e-04	3.19732934e-04	3.18067938e-04	3.15875599e-04
3.09023633e-04	3.06823798e-04	3.05022495e-04	3.00638867e-04
2.99301057e-04	2.97530018e-04	2.93887210e-04	2.88334570e-04
2.85485856e-04	2.84622246e-04	2.81273883e-04	2.79222826e-04
2.76653917e-04	2.73420666e-04	2.70228697e-04	2.67891771e-04

```
 2.67672764e-04 2.63364037e-04 2.62339725e-04 2.58360530e-04
 2.57426267e-04 2.54480386e-04 2.54204181e-04 2.51678211e-04
 2.51111998e-04 2.45870456e-04 2.44001696e-04 2.41976543e-04
 2.37603667e-04 2.36209949e-04 2.35123328e-04 2.33432882e-04
 2.32111770e-04 2.27529283e-04 2.26060170e-04 2.24604899e-04
 2.23175918e-04 2.20828512e-04 2.17568478e-04 2.17371057e-04
 2.14214986e-04 2.12179764e-04 2.11816228e-04 2.10193738e-04
 2.05188298e-04 2.04322835e-04 2.03640438e-04 1.98824625e-04
 1.97657589e-04 1.93911188e-04 1.92420236e-04 1.90972020e-04
 1.89318133e-04 1.88440932e-04 1.87272997e-04 1.85840618e-04
 1.83381504e-04 1.81916927e-04 1.79995428e-04 1.79498341e-04
 1.78873718e-04 1.76353555e-04 1.75043493e-04 1.74030961e-04
 1.72528343e-04 1.70624889e-04 1.68741852e-04 1.68097648e-04
 1.66871284e-04 1.65292702e-04 1.61189506e-04 1.60546489e-04
 1.59488400e-04 1.58253898e-04 1.56588698e-04 1.55533993e-04
 1.54879117e-04 1.54188316e-04 1.52814146e-04 1.51854873e-04
 1.49878456e-04 1.48885405e-04 1.46714130e-04 1.45657170e-04
 1.44319761e-04 1.42549680e-04 1.41513201e-04 1.39742073e-04
 1.38685042e-04 1.37192216e-04 1.35829536e-04 1.34358691e-04
 1.32900205e-04 1.32102293e-04 1.31682749e-04 1.28683697e-04
 1.27913224e-04 1.26878900e-04 1.26192566e-04 1.25046865e-04
 1.24449613e-04 1.23281931e-04 1.22368208e-04 1.20580431e-04
 1.19747919e-04 1.18539019e-04 1.17423000e-04 1.16950662e-04
 1.15076921e-04 1.14404811e-04 1.13751000e-04 1.13453398e-04
 1.11360312e-04 1.11010614e-04 1.09863889e-04 1.08062207e-04
 1.07699378e-04 1.06832065e-04 1.05611019e-04 1.03755024e-04
 1.03042210e-04 1.02292609e-04 1.01428453e-04 1.00606877e-04
 9.97124303e-05 9.81938705e-05 9.72007785e-05 9.68280177e-05
 9.60940629e-05 9.54250080e-05 9.33151657e-05 9.28466985e-05
 9.22386694e-05 9.15090452e-05 9.04175404e-05 9.01460491e-05
 8.92318903e-05 8.83257227e-05 8.74777752e-05 8.67059754e-05
 8.62616016e-05 8.46162655e-05 8.38471703e-05 8.32517943e-05
 8.29109130e-05 8.24954665e-05 8.09449541e-05 8.00407434e-05
 7.93364437e-05 7.85626600e-05 7.75538776e-05 7.65576759e-05
 7.58030936e-05 7.50981106e-05 7.45724805e-05 7.37126933e-05
 7.26998536e-05 7.23732672e-05 7.13899133e-05 7.07091910e-05
 7.01600931e-05 6.82024657e-05 6.73119491e-05 6.69535230e-05
 6.66839428e-05 6.58472168e-05 6.54152509e-05 6.51298885e-05
 6.37861779e-05 6.32658311e-05 6.26770140e-05 6.17638636e-05
 6.11019405e-05 5.99386287e-05 5.94646804e-05 5.82988907e-05
 5.78406156e-05 5.71973132e-05 5.54883035e-05 5.52375174e-05
 5.46075120e-05 5.35365577e-05 5.30118055e-05 5.23294732e-05
 5.17212997e-05 5.10679332e-05 5.04610091e-05 5.03765335e-05
 4.88577530e-05 4.84496082e-05 4.78408641e-05 4.70591770e-05
 4.59865381e-05 4.53896259e-05 4.44145408e-05 4.31185222e-05
 4.25026905e-05 4.19901001e-05 4.07482349e-05 4.05317183e-05
 3.96772063e-05 3.89453214e-05 3.75388229e-05 3.48953150e-05
 3.37625881e-05 3.15625035e-05 2.96045980e-05 2.88188348e-05
 2.60902740e-05 2.46645951e-05 2.35903444e-05 6.97012414e-18]
50% explanable: 5  principal components
```

```
60% explanable: 9  principal components
70% explanable: 16  principal components
80% explanable: 32  principal components
90% explanable: 76  principal components
```

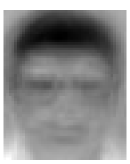
- 50% explanable: 5 principal components
- 60% explanable: 9 principal components
- 70% explanable: 16 principal components
- 80% explanable: 32 principal components
- 90% explanable: 76 principal components
- If we want to plot an image with 80% explaining of the total variance, we take n=32.

```
loading_matrix, factor_matrix, communality, uniqueness, prop_va = manual_fa(ORLface_data, 32)


first_PC = loading_matrix[:, 0]

min_value = np.min(first_PC)
max_value = np.max(first_PC)
scaled_pc = (first_PC - min_value) * (255 / (max_value - min_value))

scaled_pc_2D = scaled_pc.reshape(56,46)
img = Image.fromarray(scaled_pc_2D.astype('uint8'))
img.save('r.png', 'PNG')
img.show()
```



Question 3

a

find a oackage to perform PLSR on the AutoMPG dataset. Take 300 cars randomly to build the model and the rest 92 cars to test. In PLSR, we can see the relationship of multiple x and multiple y altogether, and see if the relashionship are strong and give us high R^2 score.

```
from sklearn.cross_decomposition import PLSRegression
from sklearn.model_selection import train_test_split

df = pd.read_csv('/content/drive/MyDrive/autompg.csv')
df.replace('?', np.nan, inplace=True)
df.dropna(inplace=True)
data = pd.DataFrame(df)

# Split data into training and testing sets
```

```python
train, test = train_test_split(data, test_size=92)
X_train = train.drop(['mpg', 'car name'], axis=1)
y_train = train['mpg']
X_test = test.drop(['mpg', 'car name'], axis=1)
y_test = test['mpg']

# Fit PLSR model
plsr = PLSRegression(n_components=2)
plsr.fit(X_train, y_train)

# Predict on test set
y_pred = plsr.predict(X_test)
# print(y_pred)
# Print R^2 score
score = plsr.score(X_test, y_test)
print(f'R^2 score: {score:.3f}')
```

```
R^2 score: 0.792
```

We can see that the R^2 is 0.792, which is really a nice number! However, if we take the model year as a part of y, things will be different.

```python
# Split data into training and testing sets
train, test = train_test_split(data, test_size=92, random_state=42)
X_train = train.drop(['mpg', 'car name', 'model year'], axis=1)
y_train = train[['mpg', 'model year']]
X_test = test.drop(['mpg', 'car name', 'model year'], axis=1)
y_test = test[['mpg', 'model year']]

# Fit PLSR model
plsr = PLSRegression(n_components=2)
plsr.fit(X_train, y_train)

# Predict on test set
y_pred = plsr.predict(X_test)
# print(y_pred)
# Print R^2 score
score = plsr.score(X_test, y_test)
print(f'R^2 score: {score:.3f}')
```

```
R^2 score: 0.355
```

We can see the R^2 score drop dramatically. In my opinion, the model year shouldn't be in the dependent part, since it makes the result worse.