Experimental Design Matrix for Frontier Methods with Preprocessing

Chocó Map

0. Preparing environment

1. Clear command window, workspace variables, and close all figures. Turn off warnings.

```
clear; clc; close all;
warning('off', 'all');
```

2. Set the path to the folder containing the layer data

```
layerfolder = 'data/layers/';
```

3. Read in the layer data from the specified folder using ReadLayers:

- To execute ReadLayers, there is one required input and three optional inputs.

Layers = ReadLayers(layer_folder, parallel, nanvalue)

```
Layers = ReadLayers(layerfolder);
```

```
----Reading layers----
Elapsed time is 2.400508 seconds.
```

Defining Experimental Matrix

```
mapsAmount = 20;
nicheOccupations = [0.3, 0.6, 0.9];
samples = [20, 50, 100, 500];
correlationPercentages = [0.7, 0.8, 0.9];
```

```
warning("off", "all")
[matrixPreprocessing, matrixNoPreprocessing]
= ExperimentalMatrixCode(Layers,
mapsAmount,nicheOccupations,samples,correlationPercentages)
```

```
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 10 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 8 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 8 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 10 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 10 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 13 components to compute TSQUARED.
```

```
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 9 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 14 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 10 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 12 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 12 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 12 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 9 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 8 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 8 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 12 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 7 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 9 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 9 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 12 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 9 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 12 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 9 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 10 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 11 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 10 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 9 components to compute TSQUARED.
Warning: Columns of X are linearly dependent to within machine precision.
Using only the first 7 components to compute TSQUARED.
matrixPreprocessing =
matrixPreprocessing(:,:,1,1,1) =

    0.8283    0.8650    0.8343
    0.8617    0.8269    0.7015
    0.6856    0.7733    0.8820
    0.7527    0.7897    0.7404
    0.8679    0.8018    0.7177
```

```
    0.7794    0.7720    0.8697
    0.4880    0.7982    0.7744
    0.7988    0.7359    0.7634
    0.8890    0.8583    0.8987
    0.8639    0.9120    0.9029
    0.8586    0.8529    0.7423
    0.8056    0.9372    0.8994
    0.6715    0.7556    0.7553
    0.6111    0.7803    0.8537
    0.8881    0.8796    0.8612
    0.5746    0.6394    0.8869
    0.8693    0.8682    0.7612
    0.8536    0.7689    0.7521
    0.7273    0.7479    0.8890
    0.8243    0.9007    0.9093


matrixPreprocessing(:,:,2,1,1) =

    0.6017    0.8536    0.9068
    0.8819    0.9167    0.8322
    0.6006    0.8498    0.8969
    0.6469    0.9304    0.7933
    0.9093    0.8687    0.8934
    0.8717    0.8594    0.7507
    0.6152    0.8890    0.8191
    0.6111    0.7939    0.9236
    0.8778    0.8967    0.9433
    0.6144    0.8602    0.9167
    0.7265    0.8599    0.7786
    0.9484    0.8924    0.7756
    0.8069    0.9067    0.8278
    0.7817    0.8826    0.7630
    0.6974    0.8828    0.8166
    0.8089    0.7284    0.8794
    0.8724    0.9208    0.7733
    0.6341    0.8643    0.8169
    0.5400    0.8581    0.9313
    0.6357    0.6702    0.8600


matrixPreprocessing(:,:,3,1,1) =

    0.5940    0.8150    0.9015
    0.9328    0.9105    0.9255
    0.6894    0.7327    0.8667
    0.5891    0.9065    0.8590
    0.7494    0.9143    0.8325
    0.8807    0.8901    0.9310
    0.6325    0.8722    0.7980
    0.6287    0.7491    0.9391
    0.8894    0.7991    0.9110
    0.7293    0.8183    0.9575
    0.7735    0.9208    0.7832
    0.9090    0.9032    0.9627
    0.7888    0.9031    0.8878
    0.5544    0.8802    0.8873
    0.6922    0.8007    0.9169
    0.6751    0.7231    0.8011
    0.8928    0.9243    0.8042
    0.7329    0.8767    0.8365
    0.6081    0.7095    0.9205
    0.6949    0.5968    0.9233
```

matrixPreprocessing(:,:,4,1,1) =

0.5641	0.6656	0.8872
0.8678	0.8599	0.9181
0.6867	0.7097	0.8034
0.5807	0.8050	0.9209
0.7705	0.9462	0.9344
0.8875	0.8890	0.9063
0.6871	0.7963	0.8621
0.6653	0.6820	0.8350
0.7648	0.7568	0.9333
0.7985	0.8181	0.8445
0.7336	0.8906	0.9387
0.7771	0.7388	0.8887
0.7555	0.8032	0.9525
0.6662	0.6676	0.9147
0.6447	0.7576	0.9167
0.7525	0.6990	0.9023
0.7741	0.9365	0.9568
0.6377	0.8295	0.9084
0.5992	0.6747	0.8114
0.7152	0.6585	0.8528

matrixPreprocessing(:,:,1,2,1) =

0.7766	0.7580	0.7149
0.8919	0.8045	0.7350
0.6365	0.8728	0.8649
0.8709	0.7533	0.7600
0.8676	0.7453	0.7288
0.7733	0.7808	0.8697
0.7715	0.8271	0.7769
0.9123	0.6248	0.7730
0.8566	0.8242	0.8818
0.8570	0.8767	0.8271
0.9288	0.7627	0.8174
0.8358	0.9250	0.8994
0.8300	0.7415	0.7388
0.8363	0.7871	0.8690
0.8760	0.8665	0.8672
0.5536	0.7470	0.8463
0.8397	0.8306	0.7615
0.8105	0.8141	0.7389
0.8512	0.7953	0.8899
0.7383	0.8784	0.8835

matrixPreprocessing(:,:,2,2,1) =

0.5796	0.8346	0.8672
0.8849	0.8973	0.8311
0.6818	0.8567	0.8969
0.6517	0.9098	0.7748
0.9373	0.8499	0.8026
0.8618	0.8694	0.7519
0.6250	0.8933	0.7858
0.5867	0.7371	0.9381
0.8723	0.8980	0.9216
0.6155	0.8594	0.9089
0.7588	0.8599	0.7883
0.6726	0.8655	0.7703
0.7625	0.9015	0.8986

```
    0.7753    0.8835    0.7859
    0.6974    0.8173    0.8509
    0.6362    0.8011    0.8531
    0.7410    0.8911    0.7928
    0.8841    0.8468    0.7989
    0.6967    0.8027    0.9282
    0.6829    0.7568    0.8514


matrixPreprocessing(:,:,3,2,1) =

    0.6035    0.8686    0.9241
    0.9255    0.8216    0.9255
    0.6264    0.7082    0.8623
    0.5851    0.9128    0.8359
    0.7528    0.9129    0.8325
    0.8911    0.8938    0.9310
    0.6074    0.8752    0.7980
    0.6339    0.7925    0.9204
    0.8155    0.8502    0.9377
    0.7688    0.8871    0.9355
    0.6923    0.9152    0.8005
    0.9082    0.8029    0.9587
    0.7827    0.8265    0.8968
    0.5984    0.8760    0.8821
    0.6364    0.7462    0.9169
    0.6920    0.7637    0.8757
    0.8903    0.9289    0.7787
    0.7704    0.9055    0.8452
    0.6733    0.6935    0.9239
    0.6525    0.6149    0.8621


matrixPreprocessing(:,:,4,2,1) =

    0.5568    0.7820    0.9141
    0.8990    0.8150    0.9405
    0.6901    0.7121    0.8017
    0.6227    0.7733    0.8944
    0.7837    0.9462    0.9247
    0.8614    0.8890    0.9063
    0.6270    0.7963    0.8406
    0.6784    0.7176    0.8491
    0.7415    0.8095    0.8991
    0.8717    0.7774    0.9350
    0.7351    0.9115    0.9223
    0.7571    0.7418    0.8781
    0.7271    0.7902    0.9279
    0.5720    0.8196    0.9145
    0.6497    0.7989    0.9304
    0.6176    0.7221    0.9023
    0.7622    0.8397    0.9333
    0.7359    0.8331    0.8838
    0.6126    0.6652    0.8114
    0.5959    0.6811    0.8927


matrixPreprocessing(:,:,1,3,1) =

    0.7791    0.7580    0.7149
    0.8538    0.7985    0.7326
    0.7861    0.7181    0.8662
    0.8010    0.7585    0.7591
    0.8310    0.7374    0.7284
```

```
    0.8136    0.7515    0.8786
    0.8436    0.8310    0.7040
    0.8982    0.7646    0.7552
    0.8564    0.8066    0.8897
    0.8565    0.8358    0.8420
    0.8983    0.8039    0.7232
    0.7914    0.8922    0.8857
    0.9021    0.7818    0.7515
    0.8209    0.7658    0.8107
    0.8242    0.8665    0.8672
    0.5512    0.7095    0.8098
    0.8418    0.8527    0.7349
    0.8585    0.8141    0.7192
    0.7817    0.8945    0.7948
    0.8262    0.8796    0.9021


matrixPreprocessing(:,:,2,3,1) =

    0.6457    0.8769    0.8707
    0.8727    0.8806    0.8649
    0.5760    0.7253    0.8603
    0.7281    0.9162    0.7592
    0.9063    0.9157    0.8042
    0.8770    0.8465    0.7858
    0.6057    0.8899    0.7802
    0.5867    0.7325    0.9375
    0.9090    0.9119    0.9355
    0.7078    0.9039    0.8855
    0.7069    0.8597    0.7883
    0.6699    0.9277    0.7699
    0.6589    0.9032    0.8605
    0.7668    0.8695    0.7856
    0.7155    0.8975    0.8503
    0.6890    0.8484    0.8787
    0.8848    0.8901    0.7981
    0.8821    0.8947    0.7994
    0.7265    0.6796    0.9339
    0.8393    0.7568    0.8635


matrixPreprocessing(:,:,3,3,1) =

    0.8138    0.8410    0.9242
    0.9200    0.8469    0.8810
    0.6877    0.7039    0.8212
    0.5494    0.8985    0.8690
    0.7378    0.9140    0.8615
    0.8381    0.8950    0.8986
    0.6034    0.8306    0.7964
    0.6339    0.7925    0.9204
    0.8561    0.8620    0.9363
    0.9028    0.8960    0.9157
    0.6923    0.9152    0.8005
    0.6799    0.8145    0.9581
    0.7495    0.8706    0.8684
    0.7881    0.8760    0.8698
    0.6283    0.7436    0.9345
    0.6996    0.7222    0.8566
    0.8921    0.8952    0.7896
    0.7645    0.9038    0.8493
    0.6635    0.6347    0.9256
    0.6621    0.6250    0.9221
```

```
matrixPreprocessing(:,:,4,3,1) =

    0.6217    0.7550    0.9163
    0.9235    0.7872    0.9406
    0.6612    0.7145    0.8002
    0.6064    0.8400    0.9030
    0.7602    0.8962    0.9212
    0.8785    0.8984    0.9290
    0.5868    0.7642    0.8584
    0.6951    0.7176    0.9354
    0.7558    0.8457    0.9157
    0.7294    0.7774    0.9240
    0.7488    0.9115    0.9223
    0.7390    0.7146    0.9356
    0.6956    0.7446    0.9692
    0.6077    0.7763    0.9203
    0.6575    0.7989    0.9304
    0.5869    0.7064    0.8503
    0.7577    0.8618    0.9320
    0.7648    0.8250    0.8876
    0.6007    0.6517    0.7745
    0.6716    0.6838    0.8854


matrixPreprocessing(:,:,1,1,2) =

    0.6442    0.6560    0.6510
    0.7575    0.7448    0.6593
    0.7546    0.7589    0.7577
    0.7822    0.7046    0.6241
    0.8070    0.7142    0.6596
    0.6470    0.6356    0.6096
    0.6098    0.7041    0.6108
    0.8285    0.7898    0.7420
    0.7464    0.6989    0.7172
    0.8405    0.8081    0.7453
    0.7706    0.6763    0.6384
    0.7766    0.7599    0.7026
    0.6928    0.7187    0.6876
    0.7384    0.6551    0.6411
    0.7785    0.7296    0.6744
    0.7274    0.6857    0.7100
    0.7156    0.7199    0.6221
    0.7122    0.6963    0.5889
    0.7960    0.7299    0.7027
    0.7465    0.7763    0.6868


matrixPreprocessing(:,:,2,1,2) =

    0.7430    0.6736    0.6990
    0.7958    0.7847    0.7864
    0.8153    0.8411    0.7829
    0.7881    0.7741    0.7008
    0.8497    0.7721    0.7703
    0.7190    0.6882    0.6102
    0.7841    0.7542    0.6493
    0.7878    0.8288    0.7848
    0.8157    0.7709    0.7737
    0.8320    0.7770    0.7477
    0.8206    0.7510    0.6926
    0.7958    0.7691    0.7019
    0.7631    0.7831    0.7312
```

```
    0.7956    0.7262    0.6134
    0.8234    0.7713    0.6830
    0.7983    0.7704    0.7106
    0.7236    0.7650    0.6668
    0.7426    0.7510    0.6838
    0.7112    0.8260    0.7776
    0.7415    0.7177    0.7068


matrixPreprocessing(:,:,3,1,2) =

    0.7666    0.7036    0.6959
    0.8445    0.7749    0.7904
    0.8456    0.7917    0.8167
    0.7351    0.7988    0.7432
    0.8278    0.8055    0.7312
    0.7321    0.7040    0.7377
    0.8088    0.7681    0.6402
    0.8316    0.8085    0.8110
    0.8267    0.7979    0.7708
    0.9111    0.7995    0.7861
    0.8601    0.8046    0.7145
    0.8840    0.8048    0.7716
    0.7765    0.7951    0.7377
    0.7616    0.7456    0.6907
    0.8183    0.7724    0.7668
    0.8310    0.7573    0.7233
    0.7644    0.7702    0.7136
    0.7974    0.7676    0.6673
    0.8168    0.7939    0.8021
    0.8180    0.6936    0.7306


matrixPreprocessing(:,:,4,1,2) =

    0.7005    0.6684    0.7152
    0.8897    0.7973    0.7952
    0.8511    0.8074    0.7967
    0.7186    0.8010    0.7576
    0.8427    0.8443    0.7846
    0.8205    0.7392    0.6964
    0.8657    0.7965    0.7190
    0.8399    0.7830    0.7776
    0.8404    0.7675    0.7860
    0.9320    0.8350    0.7524
    0.8706    0.8229    0.7374
    0.8204    0.7513    0.7752
    0.8367    0.7855    0.7989
    0.8179    0.6916    0.7201
    0.7853    0.7937    0.7751
    0.8570    0.7762    0.7678
    0.7977    0.8150    0.7963
    0.7816    0.7875    0.7280
    0.7700    0.7869    0.7643
    0.8561    0.7415    0.7176


matrixPreprocessing(:,:,1,2,2) =

    0.5787    0.6148    0.6125
    0.7681    0.7130    0.6598
    0.7398    0.8015    0.7723
    0.8110    0.6890    0.6531
    0.8065    0.6830    0.6733
```

```
    0.6036    0.6431    0.6096
    0.7235    0.7014    0.6124
    0.8000    0.7456    0.7401
    0.7404    0.7263    0.7421
    0.8608    0.8057    0.7154
    0.7670    0.6874    0.6482
    0.7822    0.7283    0.7026
    0.7157    0.7158    0.6766
    0.7668    0.6490    0.6398
    0.7359    0.7049    0.6902
    0.7171    0.7195    0.6947
    0.6874    0.7050    0.6435
    0.7350    0.7167    0.5820
    0.7917    0.7562    0.7078
    0.7067    0.7043    0.6761


matrixPreprocessing(:,:,2,2,2) =

    0.7267    0.6736    0.6813
    0.8051    0.8064    0.7859
    0.8262    0.8303    0.7829
    0.7741    0.7746    0.6582
    0.8345    0.7599    0.7200
    0.7015    0.6851    0.6040
    0.7928    0.7565    0.6418
    0.7588    0.8159    0.7977
    0.7999    0.7626    0.7534
    0.8387    0.7992    0.7550
    0.8313    0.7510    0.6924
    0.7402    0.7675    0.7130
    0.7968    0.7564    0.7400
    0.8073    0.7219    0.6194
    0.8234    0.7680    0.7170
    0.7935    0.7579    0.7225
    0.7586    0.7807    0.6841
    0.7609    0.7510    0.6574
    0.8371    0.8146    0.7726
    0.7570    0.7717    0.6830


matrixPreprocessing(:,:,3,2,2) =

    0.7718    0.7266    0.7336
    0.8455    0.7515    0.7904
    0.8053    0.7803    0.8160
    0.7484    0.7989    0.7298
    0.8314    0.7986    0.7312
    0.7528    0.7175    0.7377
    0.7863    0.7723    0.6402
    0.8233    0.8611    0.7978
    0.8254    0.8043    0.7772
    0.9126    0.8045    0.7803
    0.8285    0.7996    0.6993
    0.8561    0.7612    0.7647
    0.8123    0.7869    0.7318
    0.7972    0.7296    0.6686
    0.7804    0.7767    0.7668
    0.8388    0.7824    0.7601
    0.7798    0.7957    0.7069
    0.8125    0.7691    0.6771
    0.8329    0.7873    0.8051
    0.7842    0.6915    0.7014
```

matrixPreprocessing(:,:,4,2,2) =

0.6684	0.7412	0.7245
0.8903	0.7895	0.8062
0.8702	0.8115	0.7961
0.7602	0.7877	0.7607
0.8592	0.8443	0.8057
0.8195	0.7417	0.6964
0.7901	0.7965	0.7011
0.8526	0.8284	0.7871
0.8322	0.8010	0.7796
0.9289	0.8256	0.7855
0.8693	0.8352	0.7557
0.8828	0.7668	0.7672
0.8188	0.7943	0.7871
0.7307	0.7567	0.7235
0.8065	0.8215	0.7775
0.8106	0.7771	0.7678
0.8195	0.7945	0.7939
0.8294	0.7873	0.7143
0.8001	0.7764	0.7643
0.7429	0.7708	0.7241

matrixPreprocessing(:,:,1,3,2) =

0.6713	0.6148	0.6125
0.7308	0.7023	0.6547
0.8098	0.7932	0.7727
0.7868	0.6870	0.6513
0.7875	0.6645	0.6688
0.5996	0.6067	0.6272
0.7365	0.7036	0.5657
0.8501	0.7821	0.7482
0.7559	0.7328	0.7268
0.8348	0.8195	0.7274
0.7909	0.6787	0.6152
0.7539	0.7651	0.7498
0.7722	0.7173	0.6887
0.7634	0.6442	0.5519
0.7863	0.7049	0.6902
0.7298	0.7040	0.7131
0.7237	0.6965	0.6092
0.7284	0.7167	0.5649
0.7680	0.7560	0.7107
0.7344	0.7342	0.6662

matrixPreprocessing(:,:,2,3,2) =

0.7594	0.7022	0.6795
0.8193	0.7974	0.7994
0.7887	0.8000	0.7646
0.7980	0.7582	0.6608
0.8377	0.7715	0.7065
0.7219	0.6853	0.6346
0.7850	0.7581	0.6346
0.7588	0.8182	0.7960
0.8248	0.7668	0.7644
0.8806	0.8044	0.7459
0.7922	0.7420	0.6924
0.7583	0.8028	0.7072
0.7032	0.7780	0.7502

```
    0.8223    0.7188    0.6193
    0.8114    0.7812    0.7170
    0.8038    0.7855    0.7227
    0.7997    0.7549    0.6811
    0.7686    0.7720    0.6575
    0.8403    0.7975    0.7816
    0.8034    0.7717    0.6881


matrixPreprocessing(:,:,3,3,2) =

    0.8003    0.7010    0.7277
    0.8362    0.7746    0.7766
    0.8530    0.7876    0.7943
    0.6956    0.7887    0.7572
    0.8373    0.7979    0.7453
    0.7171    0.7275    0.7437
    0.7921    0.7733    0.6393
    0.8233    0.8611    0.7978
    0.8230    0.8084    0.7871
    0.9022    0.8131    0.7744
    0.8285    0.7996    0.6993
    0.7861    0.7697    0.7640
    0.8032    0.7787    0.7556
    0.8407    0.7296    0.6638
    0.7525    0.7772    0.7683
    0.8465    0.7652    0.7553
    0.7939    0.7404    0.7079
    0.8093    0.7669    0.6710
    0.8314    0.7489    0.8006
    0.8225    0.7209    0.7259


matrixPreprocessing(:,:,4,3,2) =

    0.7932    0.7328    0.7130
    0.8989    0.7921    0.7906
    0.8656    0.8010    0.7976
    0.8027    0.8045    0.7637
    0.8585    0.8463    0.7897
    0.8221    0.7411    0.7200
    0.7328    0.7789    0.7166
    0.8715    0.8284    0.8158
    0.8366    0.8165    0.7798
    0.9108    0.8256    0.7758
    0.8631    0.8352    0.7557
    0.8811    0.7411    0.7769
    0.7936    0.7504    0.7883
    0.7657    0.7548    0.7216
    0.8316    0.8215    0.7775
    0.7860    0.7706    0.7504
    0.8037    0.8020    0.7929
    0.8606    0.7840    0.7190
    0.7939    0.7678    0.7542
    0.8374    0.7732    0.7293
matrixNoPreprocessing =
matrixNoPreprocessing(:,:,1,1) =

    0.7701    0.7606    0.8562
    0.8040    0.8164    0.7838
    0.7603    0.8742    0.9643
    0.9034    0.7243    0.6643
    0.8269    0.7801    0.7000
    0.7748    0.7166    0.7859
```

```
    0.5946    0.8315    0.6897
    0.7100    0.7778    0.7373
    0.8495    0.7546    0.9015
    0.8345    0.7908    0.8837
    0.8080    0.7838    0.8955
    0.6078    0.8852    0.7994
    0.8675    0.7806    0.7344
    0.8143    0.7433    0.8211
    0.7034    0.8602    0.8366
    0.8258    0.8130    0.7959
    0.8668    0.8504    0.7500
    0.8484    0.8717    0.7286
    0.8617    0.9156    0.8500
    0.8500    0.8107    0.9223


matrixNoPreprocessing(:,:,2,1) =

    0.6785    0.8773    0.8535
    0.8958    0.8562    0.8016
    0.8554    0.8786    0.9485
    0.8438    0.8779    0.8487
    0.8114    0.8686    0.7797
    0.8806    0.8471    0.7722
    0.6007    0.8382    0.7596
    0.6574    0.8259    0.9672
    0.8924    0.9062    0.9453
    0.9467    0.9414    0.9304
    0.6565    0.8587    0.8749
    0.6880    0.6941    0.9012
    0.8516    0.9372    0.8717
    0.9046    0.8470    0.7669
    0.6697    0.8878    0.8991
    0.6555    0.8348    0.8692
    0.7133    0.9304    0.8183
    0.8585    0.9356    0.8233
    0.7577    0.7988    0.9067
    0.8681    0.8548    0.8671


matrixNoPreprocessing(:,:,3,1) =

    0.7138    0.8954    0.9214
    0.8545    0.9398    0.8994
    0.8212    0.8111    0.8210
    0.7031    0.8794    0.8316
    0.7687    0.8681    0.8488
    0.7848    0.8522    0.9503
    0.6029    0.9121    0.8441
    0.6951    0.7688    0.9562
    0.8231    0.9124    0.9369
    0.8440    0.9411    0.9217
    0.6011    0.8365    0.8066
    0.6152    0.7913    0.9555
    0.7654    0.9155    0.9006
    0.6766    0.8993    0.8538
    0.7806    0.8982    0.9259
    0.6521    0.7369    0.8083
    0.8799    0.9325    0.8230
    0.7115    0.8976    0.8427
    0.7899    0.7352    0.8782
    0.8697    0.6496    0.9341
```

```
matrixNoPreprocessing(:,:,4,1) =

    0.6959    0.8491    0.9181
    0.8969    0.9463    0.9743
    0.8063    0.8162    0.8146
    0.6947    0.9077    0.9167
    0.7754    0.9022    0.8743
    0.9082    0.9097    0.8990
    0.5891    0.8033    0.8734
    0.6781    0.7400    0.9129
    0.7217    0.8468    0.9530
    0.9370    0.8051    0.8908
    0.6551    0.7997    0.9397
    0.8421    0.7643    0.9196
    0.7885    0.9066    0.9691
    0.6943    0.9079    0.9141
    0.7132    0.7966    0.8324
    0.6492    0.7091    0.8649
    0.7907    0.8530    0.9405
    0.7584    0.8528    0.8822
    0.6088    0.7283    0.8445
    0.7367    0.6928    0.9104


matrixNoPreprocessing(:,:,1,2) =

    0.6490    0.6186    0.6720
    0.8019    0.7166    0.6725
    0.7824    0.7436    0.7602
    0.7838    0.6825    0.5651
    0.7645    0.7093    0.6573
    0.6148    0.5813    0.5867
    0.7421    0.7090    0.5478
    0.8434    0.7896    0.7690
    0.7637    0.6988    0.7509
    0.8385    0.7465    0.7631
    0.7196    0.7038    0.7004
    0.7142    0.7350    0.7036
    0.7458    0.7321    0.6622
    0.7611    0.6578    0.6444
    0.7771    0.7336    0.6850
    0.7312    0.7458    0.7126
    0.7001    0.6765    0.6380
    0.7027    0.7163    0.5083
    0.7903    0.7992    0.7492
    0.7272    0.7453    0.7318


matrixNoPreprocessing(:,:,2,2) =

    0.7586    0.6971    0.7050
    0.7986    0.7802    0.7988
    0.8230    0.7829    0.8649
    0.7908    0.7405    0.6762
    0.8415    0.7745    0.7465
    0.7418    0.6758    0.6740
    0.7660    0.7389    0.6132
    0.7995    0.8559    0.8037
    0.8149    0.7868    0.7791
    0.8792    0.8106    0.7702
    0.7533    0.7392    0.7116
    0.7312    0.7074    0.7764
    0.8158    0.7887    0.7547
    0.7942    0.7067    0.6484
```

```
    0.7730    0.7705    0.7621
    0.7588    0.7766    0.7137
    0.7284    0.8014    0.6962
    0.7534    0.7666    0.6255
    0.7946    0.8047    0.7704
    0.7849    0.8085    0.7045


matrixNoPreprocessing(:,:,3,2) =

    0.8094    0.7179    0.7480
    0.8486    0.7927    0.7644
    0.8678    0.7858    0.8040
    0.8081    0.7850    0.7322
    0.8371    0.7941    0.7726
    0.6903    0.7228    0.7659
    0.7723    0.7918    0.6573
    0.8794    0.8411    0.8136
    0.8240    0.8137    0.7806
    0.9003    0.7970    0.7924
    0.7780    0.7924    0.7234
    0.7011    0.7806    0.7672
    0.8213    0.8206    0.7589
    0.8105    0.7750    0.6985
    0.7904    0.7843    0.7578
    0.7857    0.7756    0.7526
    0.8031    0.7769    0.7389
    0.8054    0.7685    0.6717
    0.8298    0.8265    0.7818
    0.8123    0.7249    0.7526


matrixNoPreprocessing(:,:,4,2) =

    0.8187    0.7724    0.7609
    0.8811    0.8554    0.8140
    0.8836    0.8424    0.7875
    0.8299    0.8119    0.7907
    0.8410    0.8543    0.8188
    0.7958    0.7394    0.7392
    0.7613    0.7916    0.7228
    0.8417    0.8242    0.8153
    0.8171    0.7973    0.7930
    0.9136    0.8122    0.7790
    0.7941    0.7974    0.7774
    0.9078    0.7948    0.7901
    0.8428    0.8325    0.7928
    0.8326    0.7905    0.7664
    0.8316    0.8038    0.7820
    0.7973    0.7639    0.7697
    0.8148    0.7858    0.7917
    0.8316    0.7980    0.7067
    0.7704    0.8070    0.7941
    0.8693    0.7618    0.7498
```

```
save('results.mat','matrixPreprocessing','matrixNoPreprocessing')
```

```
matrixNoPreprocessingL = matrixNoPreprocessing;
matrixPreprocessingL = matrixPreprocessing;
```

```
matrixPreprocessingFinal = NaN(48,
length(nicheOccupations),length(samples),length(correlationPercentages),2);
matrixNoPreprocessingFinal = NaN(48,
length(nicheOccupations),length(samples),2);
matrixPreprocessingFinal(1:20,:,:,:,:) = matrixPreprocessingL;
matrixPreprocessingFinal(21:48,:,:,:,:) = matrixPreprocessing(1:28,:,:,:,:);
matrixNoPreprocessingFinal(1:20,:,:,:) = matrixNoPreprocessingL;
matrixNoPreprocessingFinal(21:48,:,:,:) = matrixNoPreprocessing(1:28,:,:,:);
```

Analyzing Results

Frontier Depth with Preprocessing

```
matrixPFinal=
NaN(length(nicheOccupations),length(samples),length(correlationPercentages),
2);
valFinalMin = 1;
valFinalMax = 0;

for i=1:3
    for j=1:4
        for k =1:3
            matrixPFinal(i,j,k,1)=mean(matrixPreprocessingFinal(:,i,j,k,1));
            matrixPFinal(i,j,k,2)=mean(matrixPreprocessingFinal(:,i,j,k,2));
        end
        [valMin,idxMin]=min(matrixPFinal(i,j,:,2));
        if valFinalMin > valMin
            iFinalMin = i; jFinalMin = j; kFinalMin = idxMin; valFinalMin =
valMin;
        end
        [valMax,idxMax]=max(matrixPFinal(i,j,:,2));
        if valFinalMax < valMax
            iFinalMax = i; jFinalMax = j; kFinalMax = idxMax; valFinalMax =
valMax;
        end
    end
end
```

i: occupation, j: number of samples, k: correlation%

The experimental conditions that lead to the **worst** results are:

```
valFinalMin, iFinalMin, jFinalMin, kFinalMin
```

```
valFinalMin = 0.6612
iFinalMin = 3
jFinalMin = 1
kFinalMin = 3
```

The experimental conditions that lead to the **best** results are:

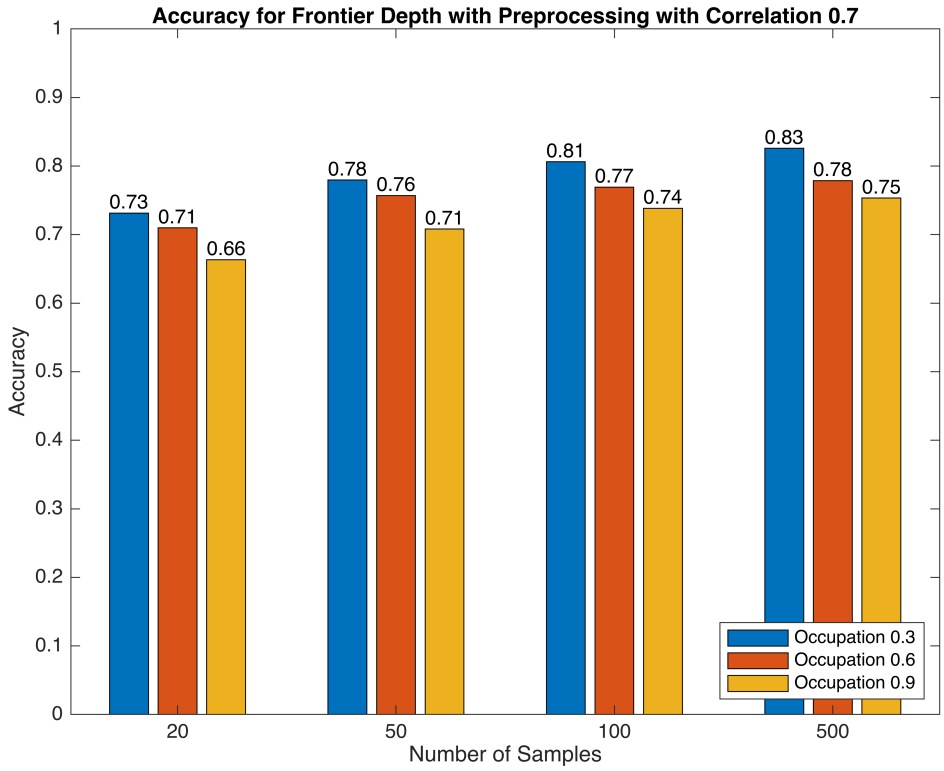
```
valFinalMax, iFinalMax, jFinalMax, kFinalMax
```

```
valFinalMax = 0.8259
iFinalMax = 1
jFinalMax = 4
kFinalMax = 1
```

Visualizing Results

Accuracy for Frontier Depth with Preprocessing with Correlation 0.7

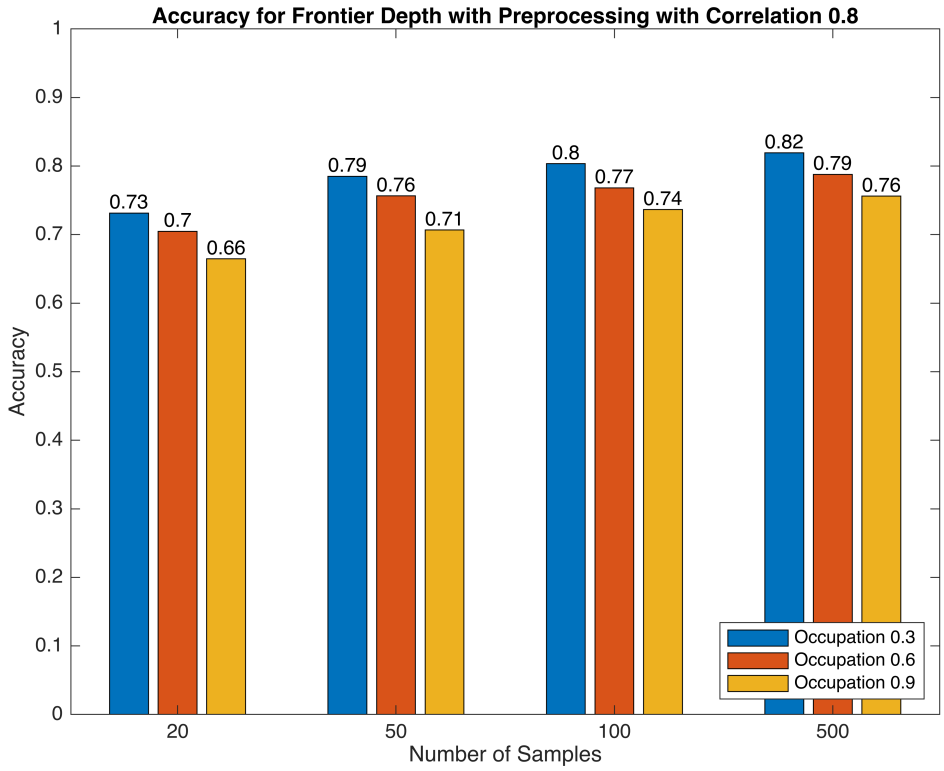
```
a = NaN(3,4);
a(:,:)= matrixPFinal(:,:,1,2);
b = bar(a');
xlabel("Number of Samples")
xticklabels({"20","50","100","500"})
ylim([0 1])
ylabel("Accuracy")

xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(round(b(1).YData,2));
text(xtips1,ytips1,labels1,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips2 = b(2).XEndPoints;
ytips2 = b(2).YEndPoints;
labels2 = string(round(b(2).YData, 2));
text(xtips2,ytips2,labels2,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips3 = b(3).XEndPoints;
ytips3 = b(3).YEndPoints;
labels3 = string(round(b(3).YData,2));
text(xtips3,ytips3,labels3,'HorizontalAlignment','center','VerticalAlignment
','bottom')

legend({"Occupation 0.3","Occupation 0.6","Occupation
0.9"},'Location','southeast')
title("Accuracy for Frontier Depth with Preprocessing with Correlation 0.7")
```



Accuracy for Frontier Depth with Preprocessing with Correlation 0.8

```
a = NaN(3,4);
a(:,:)= matrixPFinal(:,:,2,2);
b = bar(a');
xlabel("Number of Samples")
xticklabels({"20","50","100","500"})
ylim([0 1])
ylabel("Accuracy")

xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(round(b(1).YData,2));
text(xtips1,ytips1,labels1,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips2 = b(2).XEndPoints;
ytips2 = b(2).YEndPoints;
labels2 = string(round(b(2).YData, 2));
text(xtips2,ytips2,labels2,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips3 = b(3).XEndPoints;
ytips3 = b(3).YEndPoints;
labels3 = string(round(b(3).YData,2));
```

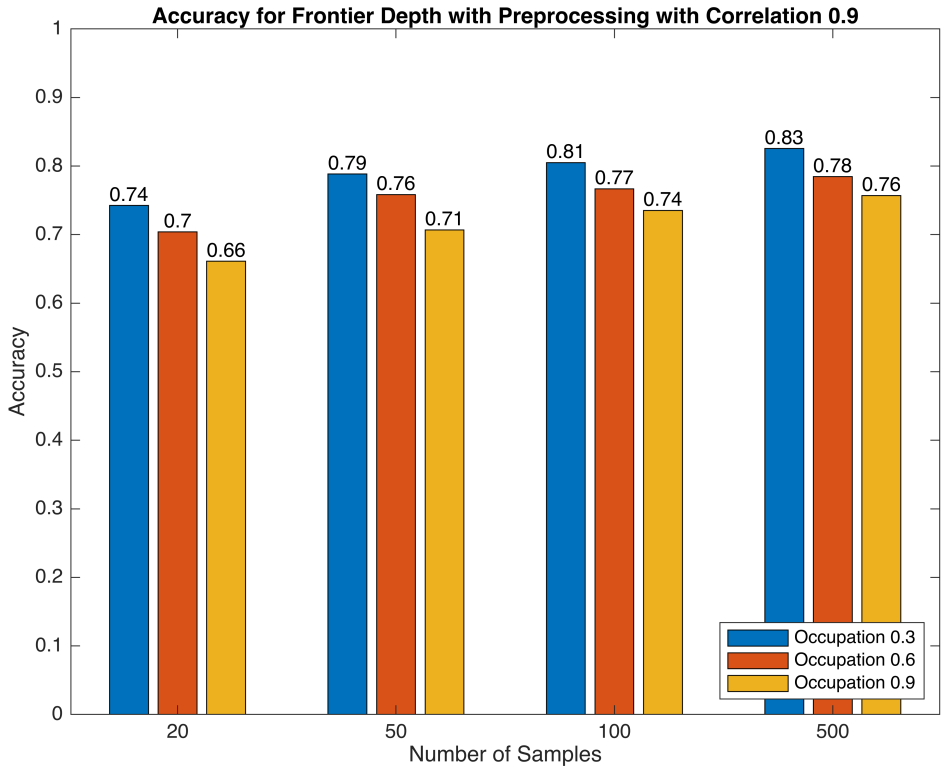
```
text(xtips3,ytips3,labels3,'HorizontalAlignment','center','VerticalAlignment
','bottom')

legend({"Occupation 0.3","Occupation 0.6","Occupation
0.9"},'Location','southeast')
title("Accuracy for Frontier Depth with Preprocessing with Correlation 0.8")
```



Accuracy for Frontier Depth with Preprocessing with Correlation 0.7

```
a = NaN(3,4);
a(:,:)= matrixPFinal(:,:,3,2);
b = bar(a');
xlabel("Number of Samples")
xticklabels({"20","50","100","500"})
ylim([0 1])
ylabel("Accuracy")

xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(round(b(1).YData,2));
text(xtips1,ytips1,labels1,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips2 = b(2).XEndPoints;
ytips2 = b(2).YEndPoints;
labels2 = string(round(b(2).YData, 2));
```

```
text(xtips2,ytips2,labels2,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips3 = b(3).XEndPoints;
ytips3 = b(3).YEndPoints;
labels3 = string(round(b(3).YData,2));
text(xtips3,ytips3,labels3,'HorizontalAlignment','center','VerticalAlignment
','bottom')

legend({"Occupation 0.3","Occupation 0.6","Occupation
0.9"},'Location','southeast')
title("Accuracy for Frontier Depth with Preprocessing with Correlation 0.9")
```



Analyzing Results

Frontier Depth with Preprocessing

```
matrixNPFinal= NaN(length(nicheOccupations),length(samples),2);
valFinalMin = 1;
valFinalMax = 0;
for i=1:3
    for j=1:4
            matrixNPFinal(i,j,1)=mean(matrixNoPreprocessingFinal(:,i,j,1));
            matrixNPFinal(i,j,2)=mean(matrixNoPreprocessingFinal(:,i,j,2));
    end
```

```
    [valMin,idxMin]=min(matrixNPFinal(i,:,2));
    if valFinalMin > valMin
        iFinalMin = i; jFinalMin = idxMin; valFinalMin = valMin;
    end
    [valMax,idxMax]=max(matrixNPFinal(i,:,2));
    if valFinalMax < valMax
        iFinalMax = i; jFinalMax = idxMax; valFinalMax = valMax;
    end

end
```

i: occupation, j: number of samples

The experimental conditions that lead to the **worst** results are:

```
valFinalMin, iFinalMin, jFinalMin
```

```
valFinalMin = 0.6690
iFinalMin = 3
jFinalMin = 1
```

The experimental conditions that lead to the **best** results are:

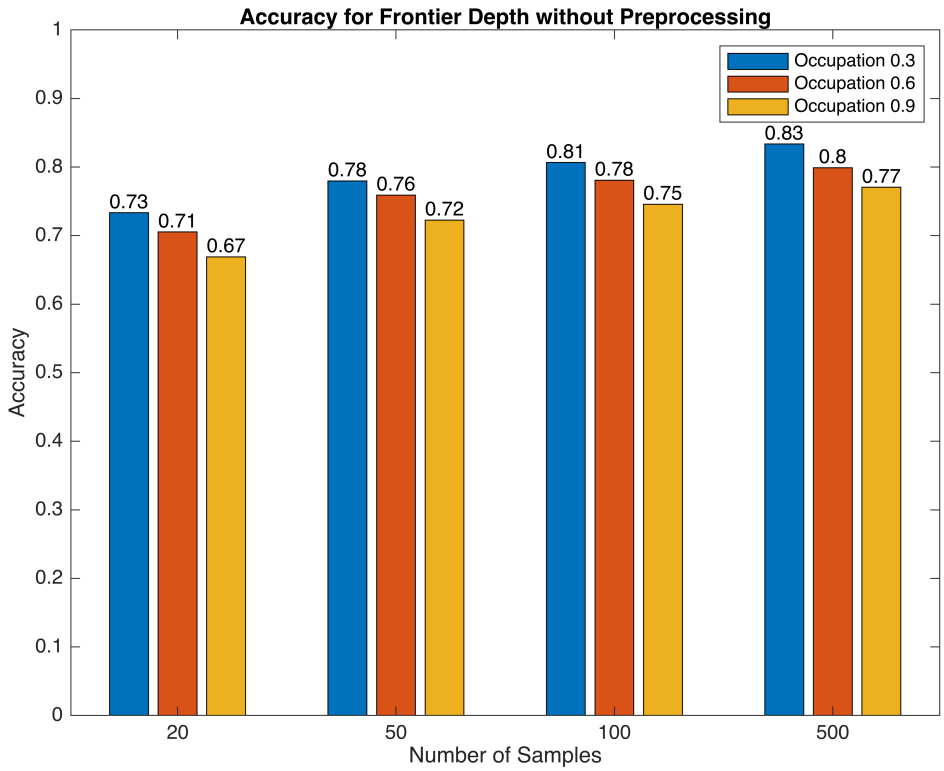
```
valFinalMax, iFinalMax, jFinalMax
```

```
valFinalMax = 0.8336
iFinalMax = 1
jFinalMax = 4
```

Visualizing Results

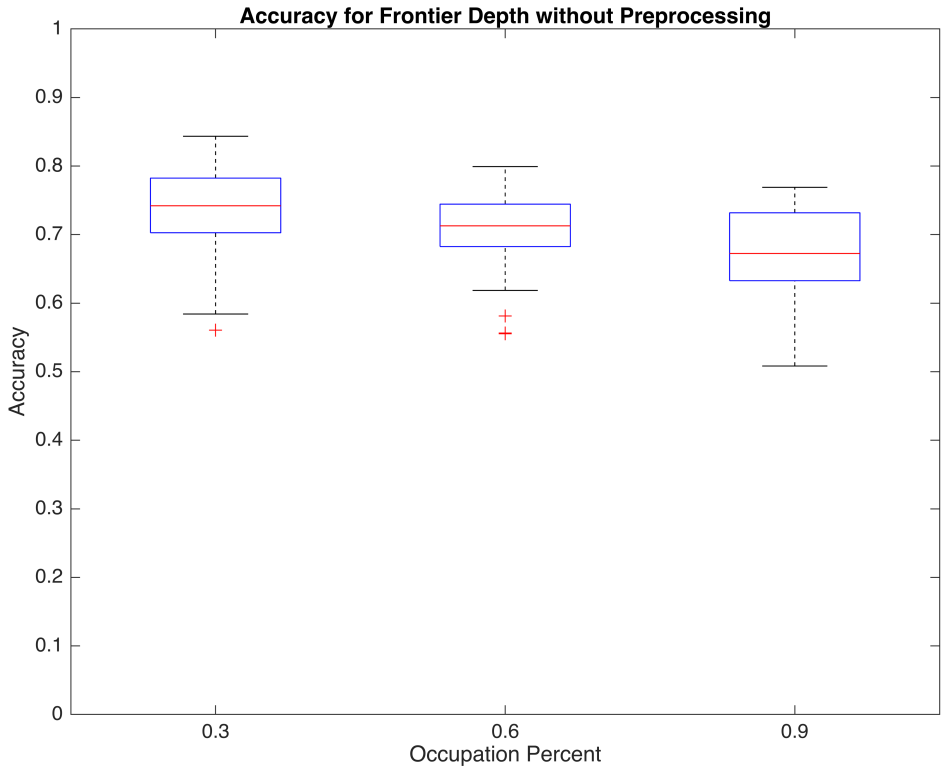
```
a = NaN(3,4);
a(:,:)= matrixNPFinal(:,:,2);
b = bar(a');
xlabel("Number of Samples")
xticklabels({"20","50","100","500"})
ylim([0 1])
ylabel("Accuracy")

xtips1 = b(1).XEndPoints;
ytips1 = b(1).YEndPoints;
labels1 = string(round(b(1).YData,2));
text(xtips1,ytips1,labels1,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips2 = b(2).XEndPoints;
ytips2 = b(2).YEndPoints;
labels2 = string(round(b(2).YData, 2));
text(xtips2,ytips2,labels2,'HorizontalAlignment','center','VerticalAlignment
','bottom')

xtips3 = b(3).XEndPoints;
```

```
ytips3 = b(3).YEndPoints;
labels3 = string(round(b(3).YData,2));
text(xtips3,ytips3,labels3,'HorizontalAlignment','center','VerticalAlignment
','bottom')

legend({"Occupation 0.3","Occupation 0.6","Occupation 0.9"})
title("Accuracy for Frontier Depth without Preprocessing")
```



Visualizing Results Using a Boxplot

```
boxplot(matrixNoPreprocessingFinal(:,:,1,2))
xlabel("Number of Samples")
xticklabels({"0.3","0.6","0.9"})
ylim([0 1])
ylabel("Accuracy")
xlabel({"Occupation Percent"})
title("Accuracy for Frontier Depth without Preprocessing")
```


Accuracy for Frontier Depth without Preprocessing
Accuracy
Occupation Percent
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0.3
0.6
0.9