############################## COPYRIGHT ##############################

All rights reserved. The authorized use of this data/code is limited to informational and educational purposes only, and NOT for operational or commercial purposes. No part of this zip file including all the data generation files, example data output and code files may be reproduced, distributed, or transmitted in any form or by any means, without the prior written permission of Prof. Siqian Shen (siqian@umich.edu). For permission requests, write to Siqian Shen (siqian@umich.edu), with a subject “Attention: Permissions for Using DR-noshow Data.”

########################################################################

======================== files =========================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LIST OF FILES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

test\_CP.py -- solve by using cutting-plane method with/without valid inequalities

test\_ED.py -- solve by using ED(2) and ED(n+1) models

test\_SLP.py -- solve by using SLP

obj\_text.py -- to generate the objective comparison table

insample\_test.py -- to generate in-sample statistics

simulate.py -- to generate out-of-sample statistics

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

test\_CP.py -- solve by using cutting-plane method with/without valid inequalities

INPUT:

-- R-list of R values

-- p0-list of expected number of showing-up

-- n0\_list-list of number of appointments

-- K-list of numbers of consecutive appts ruled out

-- train\_name-file name of training data

OUTPUT:

-- CPUtime\_CP\_n%s.xlsx-a spread sheet for each n0

-- CPn%sK%sp%sR%s\_%s.mat -- a .mat file containing solutions for each n0, K, R

test\_ED.py -- solve by using ED(2) and ED(n+1) models

INPUT:

-- R-list of R values

-- p0-list of expected number of showing-up

-- n0\_list-list of number of appointments

-- train\_name-file name of training data

OUTPUT:

-- CPUtime\_CPED\_n%s.xlsx-a spread sheet for each n0

-- CPEDn%sp%sR%s\_%s.mat -- a .mat file containing solutions for each n0, R, p

test\_SLP.py -- solve by using SLP

INPUT:

-- R-list of R values

-- p0-list of expected number of showing-up

-- n0\_list-list of number of appointments

-- train\_name-file name of training data

OUTPUT:

-- sol&obj\_SLP.xlsx-a spread sheet containing CPU time and solution details

-- SLPn%sp%sR%s\_%s.mat -- a .mat file containing solutions for each n0, R, p

obj\_text.py -- to generate the objective comparison table

INPUT:

-- R-list of R values

-- p0-list of expected number of showing-up

-- n0-number of appointments

-- train\_name-file name of training data

OUTPUT:

-- obj\_test.xlsx-a spread sheet containing objective values from solving ED(2), ED(k), and ED(n+1) models and also their solutions

insample\_test.py -- to generate in-sample statistics

INPUT:

-- R-list of R values

-- p0-list of expected number of showing-up

-- n0-number of appointments

-- mat-file name of in-sample data

-- sol\_file-file name of solutions

OUTPUT:

-- inSample\_sim.xlsx-a spread sheet containing statistics (mean, std, and quantiles) from solving ED(2),ED(n+1), and SLP models

simulate.py -- to generate out-of-sample statistics

INPUT:

-- R-list of R values

-- p0-list of expected number of showing-up

-- n0-number of appointments

-- mat-file name of in-sample data

-- sol\_file-file name of solutions

OUTPUT:

-- inSample\_sim.xlsx-a spread sheet containing statistics (mean, std, and quantiles) from solving ED(2), ED(n+1), and SLP models

======================== Folders =======================================

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*LIST OF FILES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

test\_data: Out-of-sample test samples

train\_data: sample data for training models

solution: files containing solutions generated from .py files

sample\_output\_xlsx\_files: sample output xlsx files as you can generate by running .py files

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*DETAILS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

test\_data: Out-of-sample test samples

perfect distributional info

10seed\_1465265434\_2R05.mat: n0 = 10, expect 2 no-shows out of 10 appointments

10seed\_1465265434\_4R05.mat: n0 = 10, expect 4 no-shows out of 10 appointments

misspecified distribution

cutrevisedn10seed\_2R\_test.mat: n0 = 10, expect 2 no-shows out of 10 appointments

cutrevisedn10seed\_4R\_test.mat: n0 = 10, expect 4 no-shows out of 10 appointments

train\_data: sample data for training models

training data for solving SLP (10,000 scenarios):

x1%SLPseed\_1465265434\_x2R.mat -- n0=x1, expect x2 no-shows out of 10 appointments

x1 can be 10,15,20,25,30,35,40,45,50; x2 can be 2,4

training data for solving DR models (1000 scenarios):

x1%seed\_1465265434\_4R.mat -- n0=x1, expect 4 no-shows out of 10 appointments

x1 can be 10,15,20,25,30,35,40,45,50

training data for in-sample test (1000 scenarios from the training data for solving SLP):

10seed\_1465265434\_x1R05.mat -- n0=10, expect x1 no-shows out of 10 appointments

x2 can be 2,4

sample\_output\_xlsx\_files: sample output xlsx files as you can generate by running .py files

solution: files containing solutions generated from .py files