

Homework for Lecture 10 of Dr. Z.'s Dynamical Models in Biology class

Yan Dong.

Email the answers (as .pdf file) to

ShaloshBEkhad@gmail.com

by 8:00pm Monday, Oct. 13, 2025.

Subject: hw8

with an attachment hw8FirstLast.pdf and/or hw8FirstLast.txt

Using

<http://sites.math.rutgers.edu/~zeilberg/Bio25/DMB10.txt>

1. Using procedure **ExactPD** find the probability of winning, and expected duration of the game in a Gambler's Ruin with max. capital 40 dollars, starting capital 20 dollars and prob. of winning a dollar at any one round $\frac{19}{40}$.

[0.110000, 292.0600]

Now using **EstPD**, estimate it with (i) $K=100$, (ii) $K=1000$ (iii) $K=5000$ (Warning: this may take some time) Which of them gives the best estimate? Can you explain why?

2.

Plot **ForPD**(n,100,p)[1] from $p=0.45$ to $p=0.5$ for $n=10,20,30,40,50,60,70,80,90$.

1. i) [0.110000, 292.0600] for $K=100$
ii) [0.131000, 308.3000] for $K=1000$
iii) [0.123400, 307.17280] for $K=5000$ } $Z_{\text{est PD}}(20, 40, \frac{19}{40}, K);$
 $K=100, 1000, 5000$

But $Z_{\text{exact PD}}(20, 40, \frac{19}{40}); \rightarrow [0.1190277811, 304.777751]$

so. $K=5000$ is best estimate. b/c larger sample size reduce sample errors.

2. see below plot.

> read `DMB10.txt`

For a list of the Main procedures type: Help10(); for help with a specific procedure type: Help10(ProcedureName); for example Help10(RandomMC);

For a list of the other procedures type: Help10a(); for help with a specific procedure type: Help10(ProcedureName); for example Help10(RandomMC);

> Help10();

The available procedures are: EstPD, ExactPD, ForPD, PrLA, Simu, Story

> Help10(EstPD);

EstPD(n,L,p,K): uses simulation to estimate the prob. of exiting a winner in a Gambler's Ruin with max. capital L dolalrs, initial capital n dolalrs

prob. of winning a dollar at any round p, by running Simu(n,L,p) K times and averaging. The output is the pair of numbers

[Estimated Prob. of winning, EstinatedDuration], Try:

EstPD(5,10,1/2,100);

> EstPD(20, 40, $\frac{19}{40}$, 100);

[0.1100000000, 292.0600000]

> EstPD(20, 40, $\frac{19}{40}$, 1000);

[0.1310000000, 308.3000000]

> EstPD(20, 40, $\frac{19}{40}$, 5000);

[0.1234000000, 307.1728000]

> ExactPD(20, 40, $\frac{19}{40}$);

[0.1190277811, 304.7777751]

> Help10(PrLA);

PrLA(L,p): Inputs the maximal capital L and prob. of winning a dollar p outputs the list of length L-1 such that its n-th component is the prob. of exiting a winner. Try:

PrLA(10,1/2);

> Help10(ForPD);

ForPD(n,L,p): The formula for the prob. and expected duration in a gambler's ruin with max. capital L and starting capital n and p less than 1. p can be symbolic

Try:

ForPD(50,100,p);

```
> evalf(ExactPD(20, 40,  $\frac{19}{40}$ ));
```

(10)

```
> PrLA(40,  $\frac{19}{40}$ );
```

(11)

```
[ 74368742344158402044370289529129338200416023056379
37996244189694673481419199314741370297084422504640800 ',
3914144333903073791808962606796280957916632792441
949906104742366837035479982868534257427110562616020 ',
247415123421978506524345478461175443708309262301139
37996244189694673481419199314741370297084422504640800 ',
10842505080063916320800450434338728415281531281
1184421576985494809271172048464506555395399704010 ',
570658162108627174778971075491512021856922699
47258951406397098363583129019418346864101440800 ',
14536855917634835709186649339532541742885060899163
949906104742366837035479982868534257427110562616020 ',
717050793439593243924201102434778552096387136493059
37996244189694673481419199314741370297084422504640800 ',
83198449060887472631428936505541918917761
3646595701336490567395434906387604003040005 ',
1032519789718617745496659574005195674534135470128219
37996244189694673481419199314741370297084422504640800 ',
230466617897195215045509519405933293401
7203888815340553392211080305389827576004 ',
1417898812873880363761075601602298031861135512048179
37996244189694673481419199314741370297084422504640800 ',
51169577987417143453579247825668913398890520723
1184421576985494809271172048464506555395399704010 ',
1888680389636957412665805430328952158124866865196939
37996244189694673481419199314741370297084422504640800 ',
54046439851204099085295986234159806245302774483167
949906104742366837035479982868534257427110562616020 ',
3064411976677330191229250496333350637483642499
47258951406397098363583129019418346864101440800 ',
268483670441499734415376776000296310386242
3646595701336490567395434906387604003040005 ',
3166348375861116349527267936630319433431295088156859
37996244189694673481419199314741370297084422504640800 ']
```

89350423681082174972257450224066007273506712222849
949906104742366837035479982868534257427110562616020 ’
4024597999823580874502068088908889659765975931856019
37996244189694673481419199314741370297084422504640800 ’
37589973457545958193355601
315808402904497506830552002 ’
5073041446270802080911727000695176279748231477759979
37996244189694673481419199314741370297084422504640800 ’
142035363784507701760512240152173893816474691409251
949906104742366837035479982868534257427110562616020 ’
6353826930711867321982806169941027801665612906412739
37996244189694673481419199314741370297084422504640800 ’
681118925660478364844589436086532489579843
3646595701336490567395434906387604003040005 ’
9848797443563404405118275290061988689478282299
47258951406397098363583129019418346864101440800 ’
220658301271591900587305924976633112843054870813573
949906104742366837035479982868534257427110562616020 ’
9829788377509274006956842762180277484945999785340659
37996244189694673481419199314741370297084422504640800 ’
340987272004583547356832808614712043263016635927
1184421576985494809271172048464506555395399704010 ’
12164700605715097569589166421505929827959839564703819
37996244189694673481419199314741370297084422504640800 ’
2563238526223389291691028971823790921403
7203888815340553392211080305389827576004 ’
15017044906653513611475190614754164407652646608191779
37996244189694673481419199314741370297084422504640800 ’
1600068923835821199066092268284981838360964
3646595701336490567395434906387604003040005 ’
18501487667633683900870583050273475902512834713948539
37996244189694673481419199314741370297084422504640800 ’
513084535690587330996217473101047962340233465040777
949906104742366837035479982868534257427110562616020 ’
28306066919963399343485086847949962997008842099
47258951406397098363583129019418346864101440800 ’
786411002770387129849113162863670494186455131289
1184421576985494809271172048464506555395399704010 ’
27958019497415442698715452161002683997333631436044459
37996244189694673481419199314741370297084422504640800 ’

```
774383441513504350410352014318565554433966006046059
949906104742366837035479982868534257427110562616020 '
34310268261888561262291511975192027534228386816671619 ]
37996244189694673481419199314741370297084422504640800 ]
```

```
> ForPD(20, 40, 0.45);
[0.01775080964, 192.8996761] (12)
```

```
> ForPD(20, 40, 0.5);
[Float(undefined), Float(undefined)] (13)
```

```
> PrLA(100, 0.47);
[7.731341487 × 10-7, 1.644966274 × 10-6, 2.628096117 × 10-6, 3.736731898 × 10-6,
4.986895650 × 10-6, 6.396654776 × 10-6, 7.986383151 × 10-6, 9.779055574 × 10-6,
0.00001180057980, 0.00001408017094, 0.00001665077372, 0.00001954953856,
0.00002281835848, 0.00002650447456, 0.00003066115865, 0.00003534848327,
0.00004063418975, 0.00004659466727, 0.00005331605681, 0.00006089549609,
0.00006944252335, 0.00007908066048, 0.00008994919810, 0.0001022052086,
0.0001160258162, 0.0001316107567, 0.0001491852640, 0.0001690033255,
0.0001913513522, 0.0002165523186, 0.0002449704296, 0.0002770163845,
0.0003131533124, 0.0003539034652, 0.0003998557651, 0.0004516743161,
0.0005101080012, 0.0005760013057, 0.0006503065214, 0.0007340975094,
0.0008285852192, 0.0009351351898, 0.001055287284, 0.001190777944, 0.001343565284,
0.001515857391, 0.001710144234, 0.001929233653, 0.002176291935, 0.002454889571,
0.002769052864, 0.003123322108, 0.003522817213, 0.003973311694, 0.004481316108,
0.005054172150, 0.005700158750, 0.006428611724, 0.007250058696, 0.008176371238,
0.009220936445, 0.01039885040, 0.01172713635, 0.01322499072, 0.01491406055,
0.01681875630, 0.01896660471, 0.02138864653, 0.02411988518, 0.02719979259,
0.03067287967, 0.03458933957, 0.03900577308, 0.04398600661, 0.04960201463,
0.05593495984, 0.06307636615, 0.07112944134, 0.08021056869, 0.09045098889,
0.1019986968, 0.1150205801, 0.1297048316, 0.1462636683, 0.1649363991, 0.1859928828,
0.2097374282, 0.2365131922, 0.2667071388, 0.3007556317, 0.3391507408, 0.3824473532,
0.4312711927, 0.4863278628, 0.5484130439, 0.6184239929, 0.6973725098, 0.7863995608,
0.8867917672]
```

```
> DuLA(100, 0.48);
[24.93038654, 49.85497196, 74.77327282, 99.68476543, 124.5888824, 149.4850092,
174.3724798, 199.2505730, 224.1185073, 248.9754361, 273.8204423, 298.6525324,
323.4706299, 348.2735690, 373.0600862, 397.8288133, 422.5782676, 447.3068431,
472.0127999, 496.6942531, 521.3491608, 545.9753107, 570.5703064, 595.1315519,
619.6562344, 644.1413071, 668.5834693, 692.9791449, 717.3244602, 741.6152184,
765.8468731, 790.0144991, 814.1127606, 838.1358772, 862.0775868, 885.9311056,
```

909.6890842, 933.3435611, 956.8859111, 980.3067902, 1003.596076, 1026.742802,
1049.735089, 1072.560066, 1095.203792, 1117.651161, 1139.885811, 1161.890015,
1183.644569, 1205.128670, 1226.319779, 1247.193480, 1267.723323, 1287.880654,
1307.634428, 1326.951017, 1345.793988, 1364.123874, 1381.897916, 1399.069796,
1415.589332, 1431.402163, 1446.449396, 1460.667233, 1473.986555, 1486.332488,
1497.623915, 1507.772961, 1516.684428, 1524.255183, 1530.373502, 1534.918347,
1537.758596, 1538.752199, 1537.745268, 1534.571094, 1529.049072, 1520.983547,
1510.162563, 1496.356496, 1479.316591, 1458.773360, 1434.434860, 1405.984818,
1373.080606, 1335.351043, 1292.394017, 1243.773905, 1189.018783, 1127.617402,
1059.015905, 982.6142838, 897.7625273, 803.7564578, 699.8332158, 585.1663704,
458.8606211, 319.9460594, 167.3719509]

> *plot*

plot

(16)

> *display all plots*

display all plots

(17)

> *plot*(*[seq(ForPD*(*n*, 100, *p*)[1], *n* = [10, 20, 30, 40, 50, 60, 70, 80, 90])], *p* = 0.45 ..0.5);

