Assignment 4: Applied data science

**ANSWERS:**

1. Solution are described in the following sections:
2. Using the rat example from class, together with R or Python, iterate the associated Markov matrix to obtain the transition probabilities two steps ahead, five steps ahead, 10 steps ahead, and 25 steps ahead:

Markov transition matrix when step = 2

A B C

A 0.3125 0.5 0.1875

B 0.2500 0.5 0.2500

C 0.1875 0.5 0.3125

Markov transition matrix when step = 5

A B C

A 0.2578125 0.5 0.2421875

B 0.2500000 0.5 0.2500000

C 0.2421875 0.5 0.2578125

Markov transition matrix when step = 10

A B C

A 0.2502441 0.5 0.2497559

B 0.2500000 0.5 0.2500000

C 0.2497559 0.5 0.2502441

Markov transition matrix when step = 25

A B C

A 0.25 0.5 0.25

B 0.25 0.5 0.25

C 0.25 0.5 0.25

1. Consider a three-state system with the following transition probabilities.

|  |  |  |
| --- | --- | --- |
| 1 | 0 | 0 |
| 0.25 | 0.50 | 0.25 |
| 0 | 0 | 1 |

This system is consistent with two absorbing states: Room A and Room C. Using R or Python, iterate this Markov matrix to obtain the transition probabilities two steps ahead, five steps ahead, 10 steps ahead, and 25 steps ahead..

Markov transition matrix when step = 2

A B C

A 1.0000 0.000 0.0000

B 0.4375 0.125 0.4375

C 0.0000 0.000 1.0000

Markov transition matrix when step = 5

A B C

A 1.0000000 0.000000 0.0000000

B 0.4921875 0.015625 0.4921875

C 0.0000000 0.000000 1.0000000

Markov transition matrix when step = 10

A B C

A 1.0000000 0.0000000000 0.0000000

B 0.4997559 0.0004882812 0.4997559

C 0.0000000 0.0000000000 1.0000000

Markov transition matrix when step = 25

A B C

A 1.0 0.000000e+00 0.0

B 0.5 1.490116e-08 0.5

C 0.0 0.000000e+00 1.0

Markov transition matrix when step = 1000

A B C

A 1.0 0.000000e+00 0.0

B 0.5 4.666318e-302 0.5

C 0.0 0.000000e+00 1.0

Markov transition matrix when step = 1e+05

A B C

A 1.0 0 0.0

B 0.5 0 0.5

C 0.0 0 1.0

1. Challenging question: Consider the maze presented in class, and add two rooms to the right of Room C, labeling them Room D and Room E. In this situation, treat Room A as an absorbing state. If the rat is in any room other than Rooms A or E, it has probability 0.5 of remaining in that room, probability 0.25 of moving left and probability 0.25 of moving right. For Room E, assume probability 0.5 of remaining in that room, and probability 0.5 of moving left. Write out the matrix of Markov transition probabilities. Iterate this matrix forward as many times as is necessary for you to determine empirically its limit. Based on this limit, what can you say about the evolution of the system if the rat begins in Room C? Is there a general conclusion you can draw:
2. A B C D E
3. A 1.00 0.00 0.00 0.00 0.00
4. B 0.25 0.50 0.25 0.00 0.00
5. C 0.00 0.25 0.50 0.25 0.00
6. D 1.00 0.00 0.25 0.50 0.25
7. E 0.00 0.00 0.00 0.50 0.50
8. Markov transition matrix when step = 2
9. A B C D E
10. A 1.000000 0.00000 0.00000 0.00000 0.000000
11. B 0.453125 0.21875 0.21875 0.09375 0.015625
12. C 0.125000 0.21875 0.31250 0.25000 0.093750
13. D 0.015625 0.09375 0.25000 0.40625 0.234375
14. E 0.000000 0.03125 0.18750 0.46875 0.312500
15. Markov transition matrix when step = 5
16. A B C D E
17. A 1.00000000 0.0000000 0.0000000 0.0000000 0.00000000
18. B 0.58105469 0.1049805 0.1425781 0.1206055 0.05078125
19. C 0.26708984 0.1425781 0.2255859 0.2441406 0.12060547
20. D 0.09570312 0.1206055 0.2441406 0.3461914 0.19335938
21. E 0.04492188 0.1015625 0.2412109 0.3867188 0.22558594

Markov transition matrix when step = 10

A B C D E

A 1.0000000 0.00000000 0.00000000 0.0000000 0.00000000

B 0.6801729 0.05514336 0.09392357 0.1123104 0.05844975

C 0.4154892 0.09392357 0.16745377 0.2108231 0.11231041

D 0.2447290 0.11231041 0.21082306 0.2797642 0.15237331

E 0.1862793 0.11689949 0.22462082 0.3047466 0.16745377

Markov transition matrix when step = 25

A B C D E

A 1.0000000 0.00000000 0.00000000 0.00000000 0.00000000

B 0.8246035 0.02672794 0.04935541 0.06444496 0.03486817

C 0.6759350 0.04935541 0.09117290 0.11909175 0.06444496

D 0.5766219 0.06444496 0.11909175 0.15561785 0.08422358

E 0.5417537 0.06973634 0.12888992 0.16844716 0.09117290

Markov transition matrix when step = 1000

A B C D E

A 1 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

B 1 9.900928e-19 1.829453e-18 2.390295e-18 1.293619e-18

C 1 1.829453e-18 3.380388e-18 4.416690e-18 2.390295e-18

D 1 2.390295e-18 4.416690e-18 5.770684e-18 3.123071e-18

E 1 2.587237e-18 4.780591e-18 6.246143e-18 3.380388e-18

Markov transition matrix when step = 1e+07

A B C D E

A 1 0.000000e+00 0.000000e+00 0.00000e+00 0.000000e+00

B 1 7.410985e-323 1.333977e-322 1.72923e-322 9.387247e-323

C 1 7.410985e-323 1.333977e-322 1.72923e-322 9.387247e-323

D 1 7.410985e-323 1.333977e-322 1.72923e-322 9.387247e-323

E 1 7.410985e-323 1.333977e-322 1.72923e-322 9.387247e-323

Due to memory limitation, manual prediction for step = n is as follows:

Markov Transition matrix when step = n

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **A** | **B** | **C** | **D** | **E** |
| **A** | 1 | 0 | 0 | 0 | 0 |
| **B** | 1 | 0 | 0 | 0 | 0 |
| **C** | 1 | 0 | 0 | 0 | 0 |
| **D** | 1 | 0 | 0 | 0 | 0 |
| **E** | 1 | 0 | 0 | 0 | 0 |

**ANSWERS:**

1. Solution are described in the following sections:
2. Read this dataset into R or Python. You will see that it is smaller than the earlier version of the dataset but has an additional variable called “prior\_union”. For each individual in the sample, this variable is prior union status, for which a 0 indicates “not in union” and a 1 indicates “in union”. Using R or Python, create a two-by-two table that relates prior union status to current union status. Your results should match what was presented in class. Again using R or Python, create a two-by-two matrix of Markov transition probabilities based on these results (either in percentage or decimal format):
3. Summation table:
4. 0 1
5. 0 14758 2086
6. 1 2110 2812
7. As percentage:
8. 0 1
9. 0 0.8761577 0.1238423
10. 1 0.4286875 0.5713125

Calculate Markov matrix using markovchain library:

1. 0 1
2. 0 0.8761577 0.1238423
3. 1 0.4286875 0.5713125
4. Using R or Python, estimate the logit model presented in class, using prior union status as a characteristic. Present the results in a “nice” table:
5. Call:
6. glm(formula = union ~ age + grade + smsa + south + black + year +
7. prior\_union, family = "binomial", data = union\_ori)
8. Deviance Residuals:
9. Min 1Q Median 3Q Max
10. -1.7413 -0.5600 -0.4959 -0.3628 2.4232
11. Coefficients:
12. Estimate Std. Error z value Pr(>|z|)
13. (Intercept) -2.296040 0.385301 -5.959 2.54e-09 \*\*\*
14. age 0.023661 0.006152 3.846 0.00012 \*\*\*
15. grade 0.046393 0.007849 5.911 3.40e-09 \*\*\*
16. smsa 0.047586 0.043301 1.099 0.27179
17. south -0.662996 0.042135 -15.735 < 2e-16 \*\*\*
18. black 0.579656 0.043245 13.404 < 2e-16 \*\*\*
19. year -0.011370 0.006686 -1.701 0.08903 .
20. prior\_union 2.120996 0.037795 56.119 < 2e-16 \*\*\*
21. ---
22. Signif. codes: 0 ・\*\*・0.001 ・\*・0.01 ・・0.05 ・・0.1 ・・1
23. (Dispersion parameter for binomial family taken to be 1)
24. Null deviance: 23211 on 21765 degrees of freedom
25. Residual deviance: 18925 on 21758 degrees of freedom
26. AIC: 18941
27. Number of Fisher Scoring iterations: 4
28. Regression Results for logit
29. =============================================
30. Dependent variable:
31. ---------------------------
32. union
33. ---------------------------------------------
34. age 0.024\*\*\*
35. (0.006)
37. grade 0.046\*\*\*
38. (0.008)
40. smsa 0.048
41. (0.043)
43. south -0.663\*\*\*
44. (0.042)
46. black 0.580\*\*\*
47. (0.043)
49. year -0.011\*
50. (0.007)
52. prior\_union 2.121\*\*\*
53. (0.038)
55. Constant -2.296\*\*\*
56. (0.385)
58. ---------------------------------------------
59. Observations 21,766
60. Log Likelihood -9,462.543
61. Akaike Inf. Crit. 18,941.090
62. =============================================
63. Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

c. Using R or Python, reset your “prior\_union” variable so that it takes on value 0 for all observations:

> head(union\_2c$pred\_01)

[1] 0.1984455 0.2043759 0.2104370 0.2145504 0.2187219 0.2249275

> head(union\_2c$pred\_00)

[1] 0.8015545 0.7956241 0.7895630 0.7854496 0.7812781 0.7750725

d. Using R or Python, reset your “prior\_union” variable so that it takes on value 1 for all observations

> head(union\_pred$pred\_11)

[1] 0.6736977 0.6817509 0.6896969 0.6949331 0.7001193 0.7076127

> head(union\_pred$pred\_10)

[1] 0.3263023 0.3182491 0.3103031 0.3050669 0.2998807 0.2923873

e. In a two-by-two table, find the average values for the four predictions created above in a

manner consistent with that presented in class:

> print(p\_transition)

0 1

0 0.8714181 0.1285819

1 0.4629799 0.5370201

f. Use the Markov matrix calculated in 2e. to iterate the system forward until the Markov

matrix has converged to its limit:

Markov transition matrix when step = 2

0 1

0 0.8189003 0.1810997

1 0.6520786 0.3479214

Markov transition matrix when step = 5

0 1

0 0.7851106 0.2148894

1 0.7737439 0.2262561

Markov transition matrix when step = 10

0 1

0 0.7826680 0.2173320

1 0.7825388 0.2174612

Markov transition matrix when step = 25

0 1

0 0.7826399 0.2173601

1 0.7826399 0.2173601

Markov transition matrix when step = 1e+05

0 1

0 0.7826399 0.2173601

1 0.7826399 0.2173601

It could be observed that after reaching step 25, the values converged at 0.7826399 at first column and 0.2173601 at second column. The statistic comparison:

Naive:

1. 0 1
2. 0 0.8761577 0.1238423
3. 1 0.4286875 0.5713125
4. Model based:

0 1

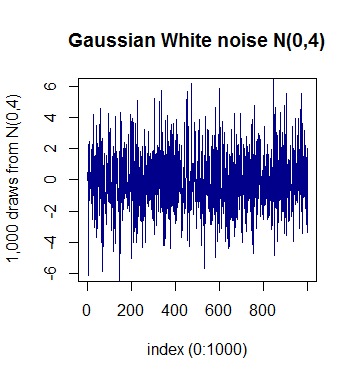
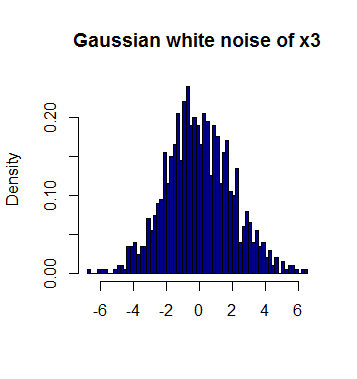
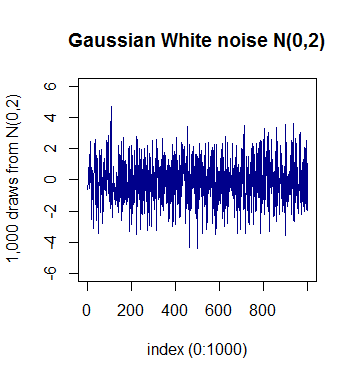
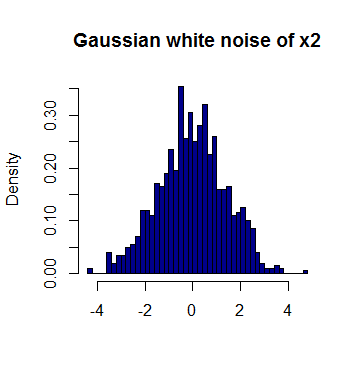
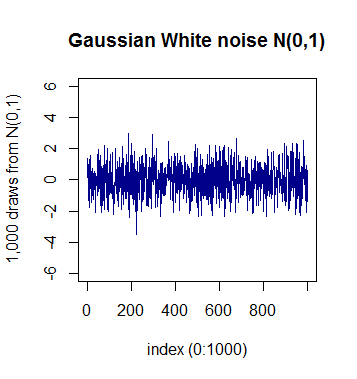
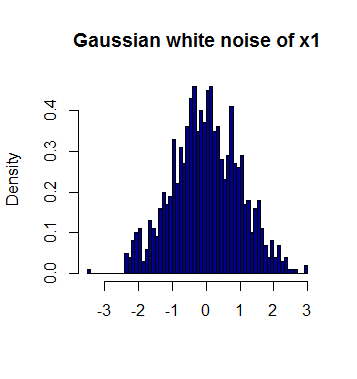
0 0.8714181 0.1285819

1. 1 0.4629799 0.5370201

We can also say that using iterative process for model based we can achieve convergent and more predictable long-term prediction

**ANSWERS:**

1. Solution are described in the following sections:
2. Generate and plot three Gaussian white noise random variables with 1,000 draws, the first with variance 1, the second with variance 2, the third with variance 4:



1. Linear model Summary of X2 ~ X1:
2. > summary(linear.model)
3. Call:
4. lm(formula = X2 ~ X1)
5. Residuals:
6. Min 1Q Median 3Q Max
7. -3.6607 -0.6669 0.0042 0.6520 3.1544
8. Coefficients:
9. Estimate Std. Error t value Pr(>|t|)
10. (Intercept) 0.052297 0.031885 1.640 0.101
11. X1 0.006434 0.031455 0.205 0.838
12. Residual standard error: 1.008 on 998 degrees of freedom
13. Multiple R-squared: 4.193e-05, Adjusted R-squared: -0.00096
14. F-statistic: 0.04184 on 1 and 998 DF, p-value: 0.838
15. repeat b. above 1,000 times, each time recording the estimated value of the slope coefficient of the bivariate regression. Generate a histogram of your 1,000 replications:

