Andrew Garwood

Question 1:

(a)

```
% 1a
x1 = 0;
for i = 1:25000000
   x1 = x1 + .1;
end
x1;

x2 = 0;
for i = 1:12500000
   x2 = x2 + .2;
end
x2;

x3 = 0;
for i = 1:10000000
   x3 = x3 + .25;
end
x3;

x4 = 0;
for i = 1:5000000
   x4 = x4 + .5;
end
x4;
```

(b)

```
the_val = 2500000;

y1 = abs(the_val - x1);
y2 = abs(the_val - x2);
y3 = abs(the_val - x3);
y4 = abs(the_val - x4);
```

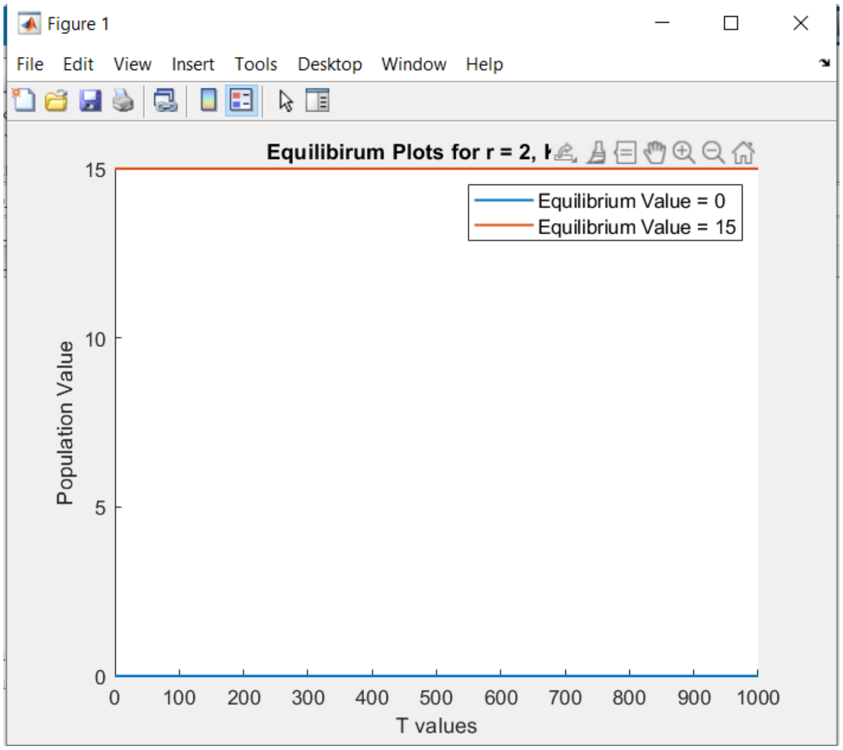
(c)

```
% y1 is larger. y1 - y2 = 0.001148897223175. This is because the summation
% of .1 25000000 times resulted in a larger error than the sum of .2
% 12500000 times. x1 iterated through the summation loop twice as much as
% x2 so it would have created a larger discrepancy because of the rounding
% error involved when python stores the specified numbers.
```

(d)
y3 and y4 are exactly zero, while y1 and y2 are not.

(e)
% Perhaps part of the reason why these values are exactly zero is because
% they had less iterations of their for loops. A computer stores numbers as
% binary in bytes, I believe. From a google search to convert decimals to
% binary, .1 and .2 respective binary counterparts are much larger than .5
% and .25. This confirms the idea that it is more strenuous to store/add a
% lot of these values, while it would be easier to store/add .25 and .5;
% this also explains why .1 and .2 have more rounding errors.

Problem 2
(a)
R = 2, k = 30
% These graphs makes sense, it follows the logic of a steady state value. The values do not change because they are already at their steady state value



(code was commented when writing later parts of assignment, apologies for mess)
P_eq_1 = 0;
P_eq_2 = 15;

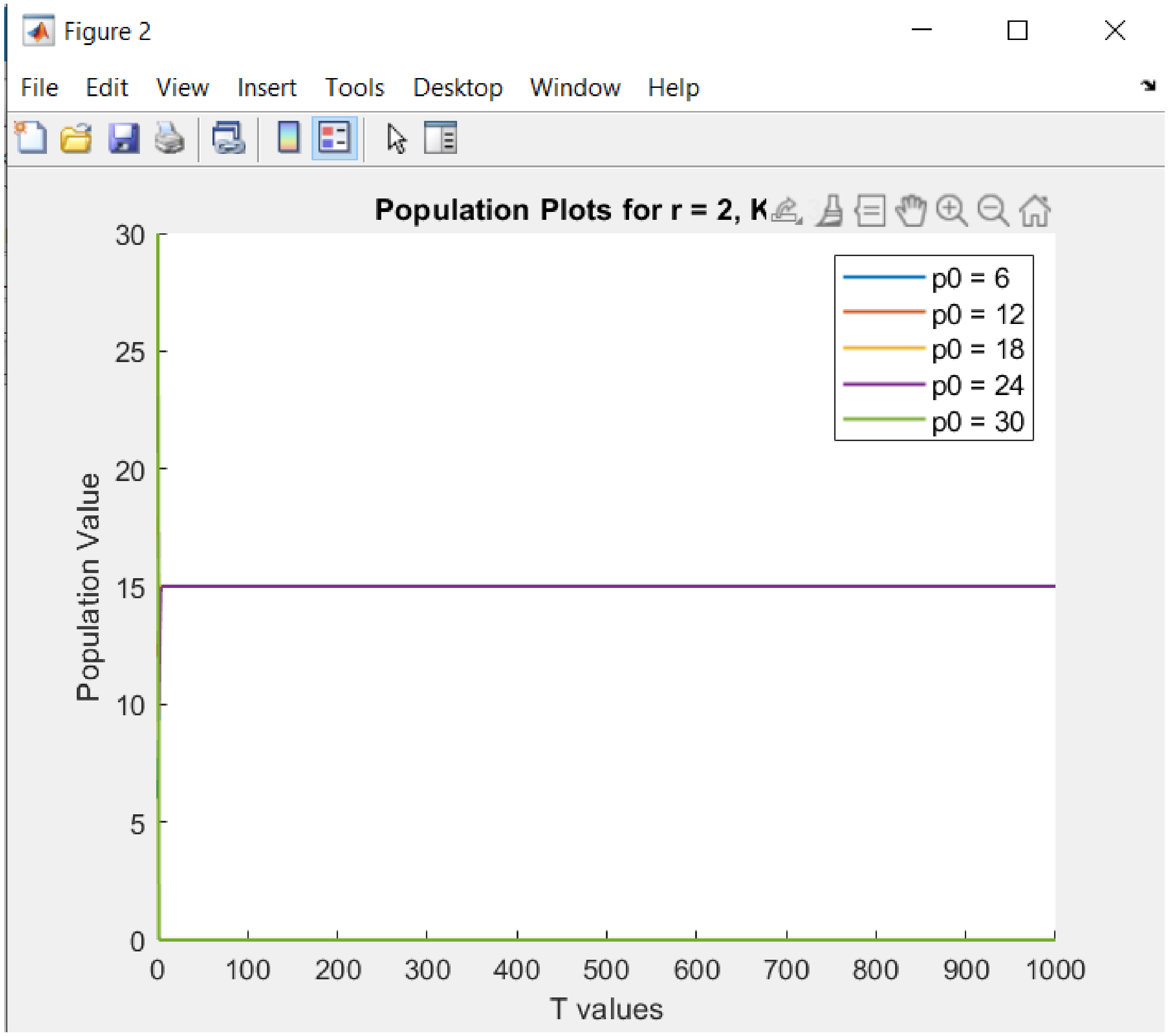
% case 1: Let p0 = 0.
% expect all 0s . . .
eq1_vector = population_model_vector(2, P_eq_1, 30, 1000);

% case 1: Let p0 = 0
% we get 1x1001 of 15
eq2_vector = population_model_vector(2, P_eq_2, 30, 1000);

x = 0:1:1000;
% figure()
% hold on
% plot(x, eq1_vector, 'DisplayName', 'Equilibrium Value = 0', 'LineWidth', 1.25)
% plot(x, eq2_vector, 'DisplayName', 'Equilibrium Value = 15', 'LineWidth', 1.25)
% hold off
% xlabel('T values')

% ylabel('Population Value')
% title('Equilibirum Plots for r = 2, K = 30')
% legend('FontSize', 10)

(b)

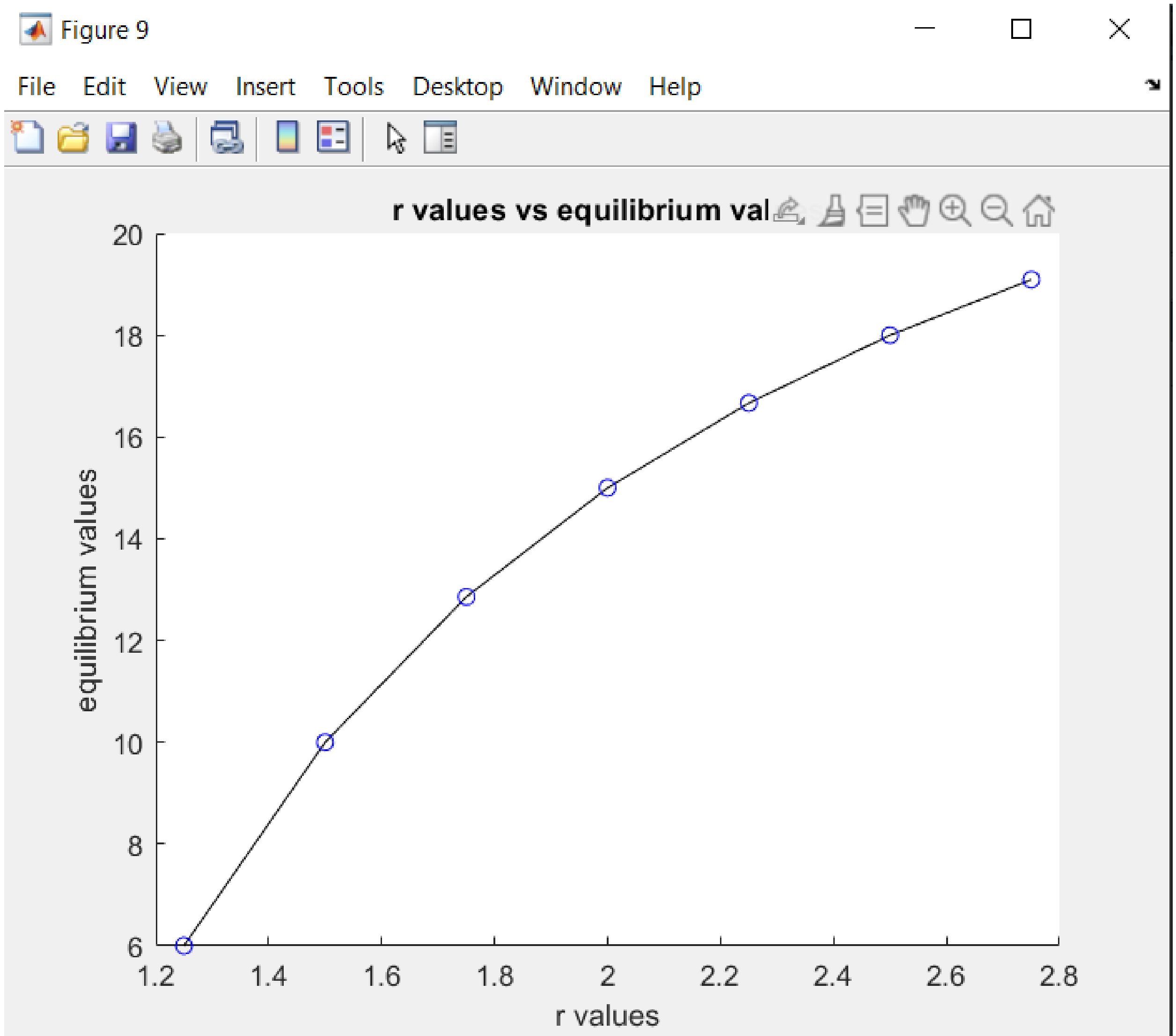


x = 0:1000;

% 0 is unstable, 15 is stable. Solutions rapidly approach 15. I believe we
% did something like this in Math 307.

```
% figure()
% hold on
% plot(x, population_model_vector(2, 6, 30, 1000), 'DisplayName', 'p0 = 6', 'LineWidth', 1.25)
% plot(x, population_model_vector(2, 12, 30, 1000), 'DisplayName', 'p0 = 12', 'LineWidth', 1.25)
% plot(x, population_model_vector(2, 18, 30, 1000), 'DisplayName', 'p0 = 18', 'LineWidth', 1.25)
% plot(x, population_model_vector(2, 24, 30, 1000), 'DisplayName', 'p0 = 24', 'LineWidth', 1.25)
% plot(x, population_model_vector(2, 30, 30, 1000), 'DisplayName', 'p0 = 30', 'LineWidth', 1.25)
% hold off
% xlabel('T values')
% ylabel('Population Value')
% title('Population Plots for r = 2, K = 30')
% legend('FontSize', 10)
```

(c)
R vs. eq values



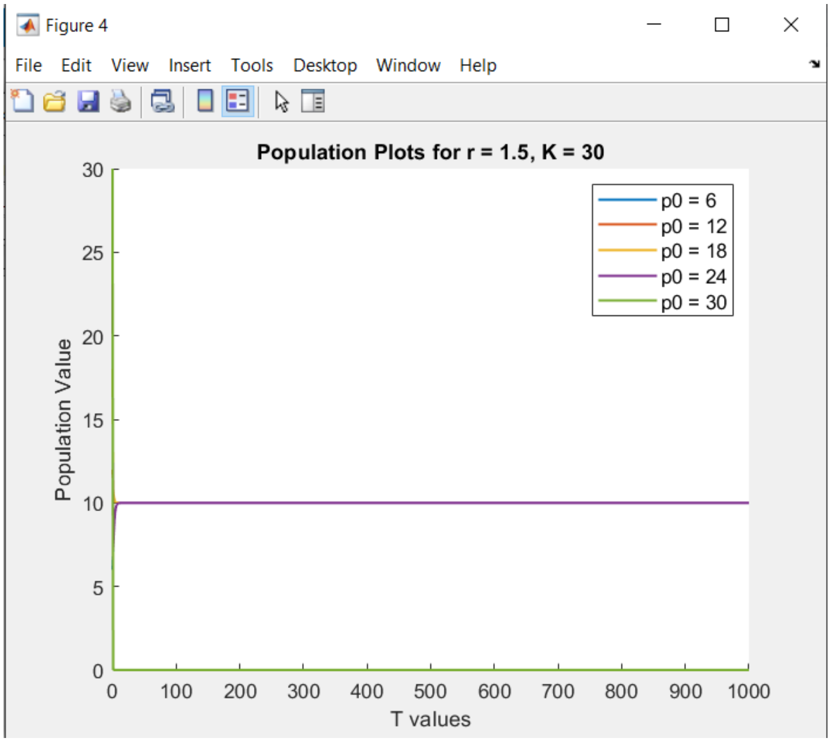


Figure 4
File Edit View Insert Tools Desktop Window Help
Population Plots for r = 1.5, K = 30
p0 = 6
p0 = 12
p0 = 18
p0 = 24
p0 = 30
Population Value
T values
0 100 200 300 400 500 600 700 800 900 1000
0 5 10 15 20 25 30

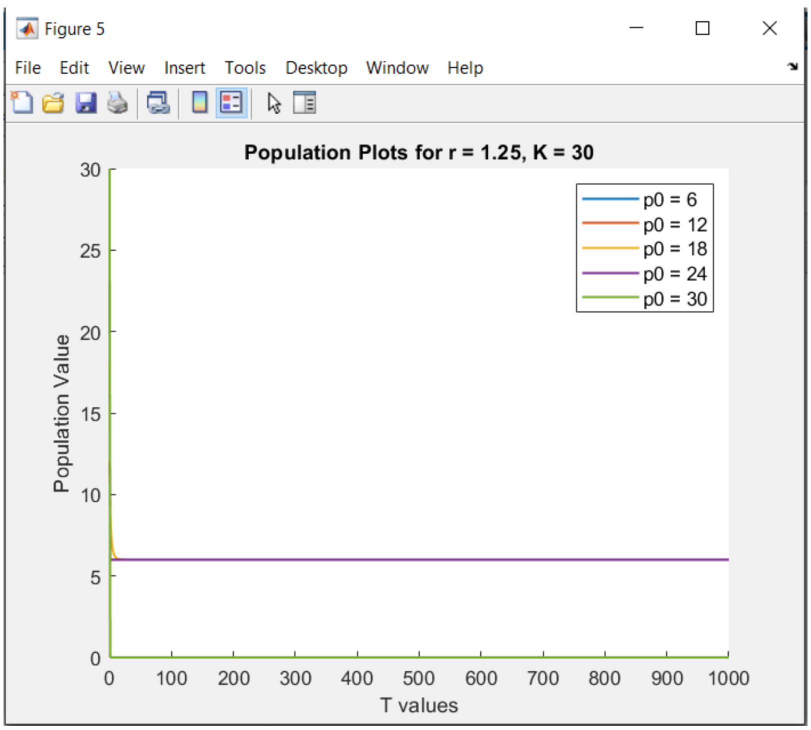


Figure 5
File Edit View Insert Tools Desktop Window Help
Population Plots for r = 1.25, K = 30
p0 = 6
p0 = 12
p0 = 18
p0 = 24
p0 = 30
Population Value
T values
0 100 200 300 400 500 600 700 800 900 1000
0 5 10 15 20 25 30

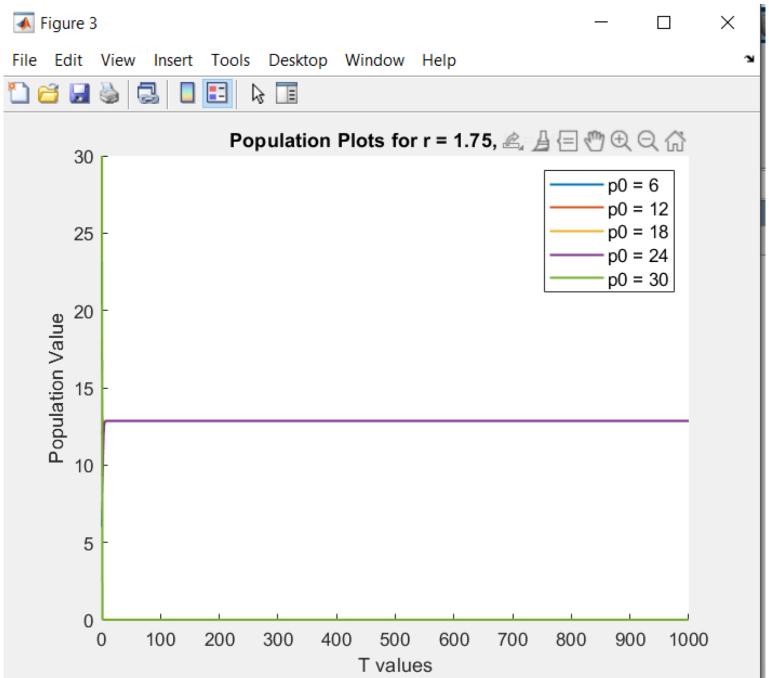


Figure 3
File Edit View Insert Tools Desktop Window Help
Population Plots for r = 1.75,
p0 = 6
p0 = 12
p0 = 18
p0 = 24
p0 = 30
Population Value
T values
0
5
10
15
20
25
30
0
100
200
300
400
500
600
700
800
900
1000

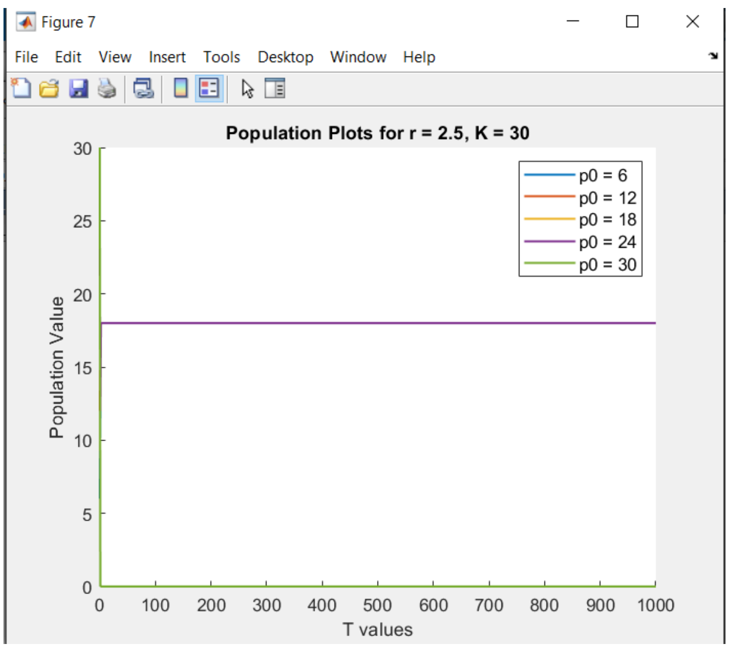


Figure 7
File Edit View Insert Tools Desktop Window Help
Population Plots for r = 2.5, K = 30
p0 = 6
p0 = 12
p0 = 18
p0 = 24
p0 = 30
Population Value
T values
0
5
10
15
20
25
30
0
100
200
300
400
500
600
700
800
900
1000

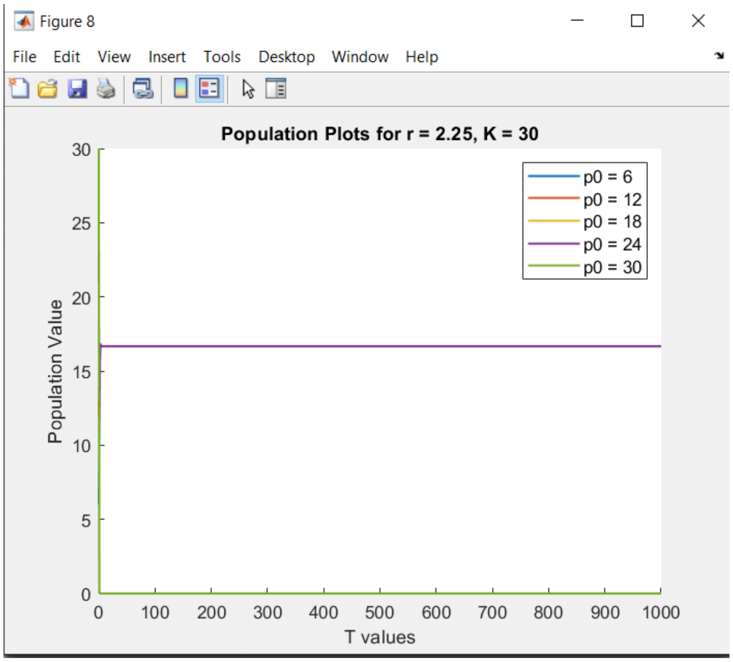


Figure 8
File Edit View Insert Tools Desktop Window Help
Population Plots for r = 2.25, K = 30
p0 = 6
p0 = 12
p0 = 18
p0 = 24
p0 = 30
Population Value
T values
0 100 200 300 400 500 600 700 800 900 1000
0 5 10 15 20 25 30

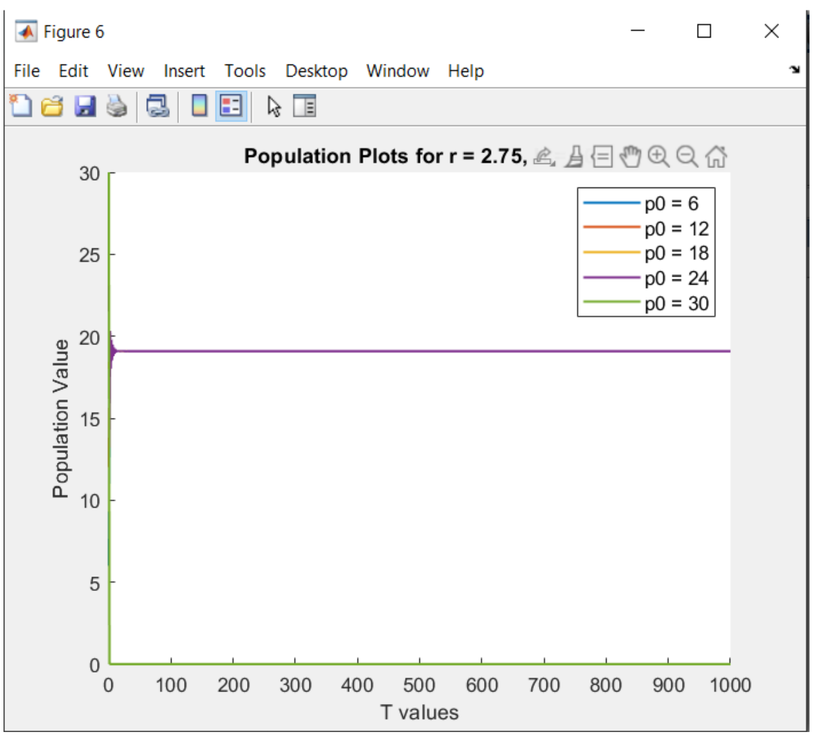


Figure 6
File Edit View Insert Tools Desktop Window Help
Population Plots for r = 2.75,
p0 = 6
p0 = 12
p0 = 18
p0 = 24
p0 = 30
Population Value
T values
0 100 200 300 400 500 600 700 800 900 1000
0 5 10 15 20 25 30

(d)

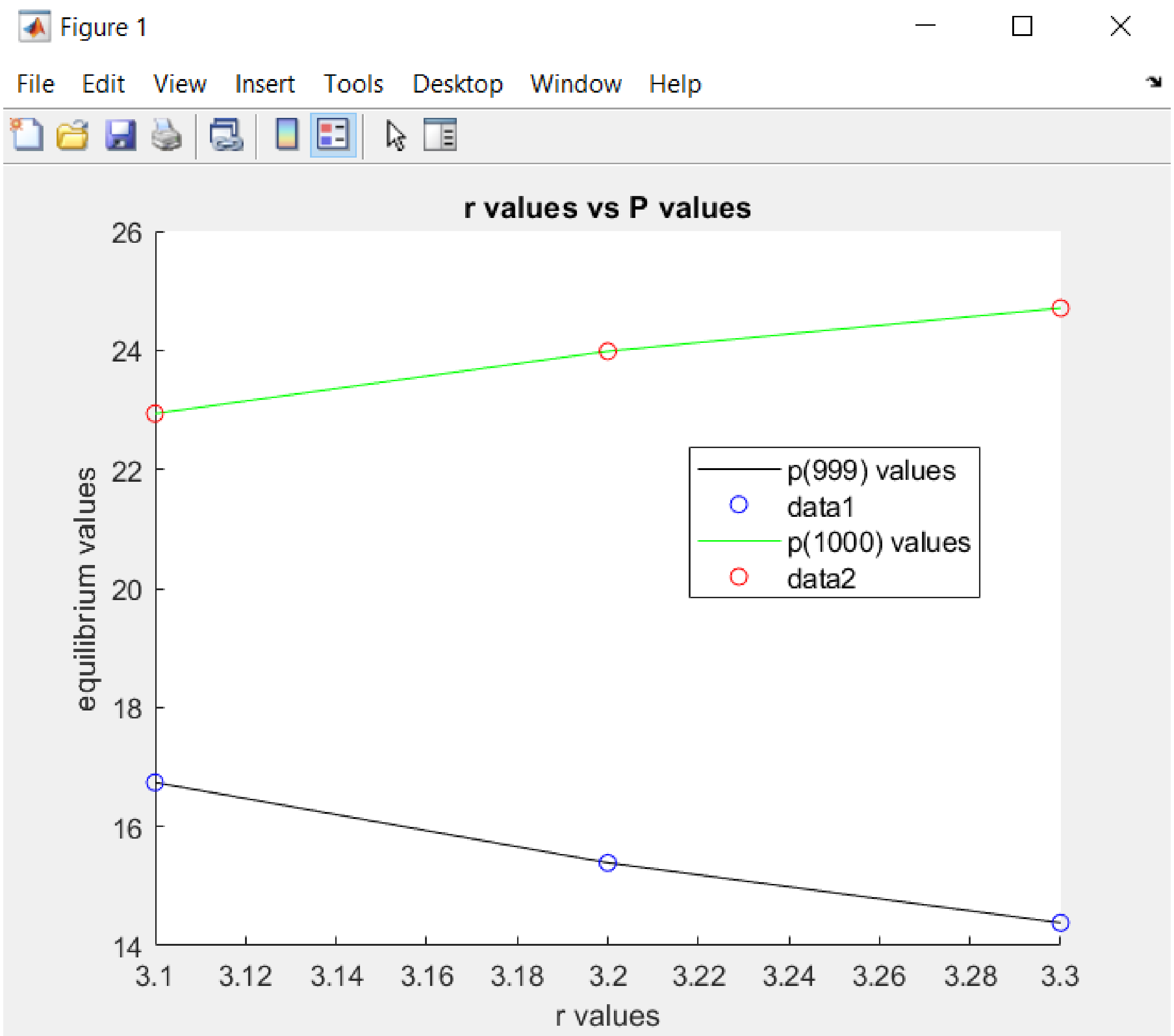


Figure 1
File Edit View Insert Tools Desktop Window Help
r values vs P values
equilibrium values
r values
26
24
22
20
18
16
14
3.1
3.12
3.14
3.16
3.18
3.2
3.22
3.24
3.26
3.28
3.3
p(999) values
data1
p(1000) values
data2