```
========================
PREDICTION DATA
========================

('random-walk.mat')
------------------

y(1) = 1;

N = 10000;
num_observations = 500;


-----------------------------------------------

for k = 1 : num_observations

for i = 2 : N

if(rand() >= .52)

y(i) = y(i - 1) + sin(pi/4);

else

y(i) = y(i - 1) - sin(pi/4);

endif

endfor

data_array1{k} = y;

endfor


for k = 1 : num_observations

for i = 2 : N

if(rand() >= .48)

y(i) = y(i - 1) + sin(pi/4);

else

y(i) = y(i - 1) - sin(pi/4);
```

```
endif

endfor

data_array2{k} = y;

endfor

-------------------------------------------------------
CATEGORIZE
-------------------------------------------------------

data_array = {data_array1{:} data_array2{:}};

tic;

[category_tree delta_tree anchor_tree] = generate_data_tree_N(data_array,N);

toc

-------------------------------------------------------------
DISPLAY
-------------------------------------------------------------

temp_cat = data_array;

num_charts = size(temp_cat,2);

figure

hold on

for i = 1 : num_charts

y = temp_cat{i};

plot(1:N, y)

endfor

=============================================================
PREDICTION
=============================================================

=============================================================
GENERATES NEW DATA
=============================================================
```

```
clear y

predicted_vector_array = {};

y(1) = 1;

for i = 2 : N

if(rand() >= .48)

y(i) = y(i - 1) + sin(pi/4);

else

y(i) = y(i - 1) - sin(pi/4);

endif

endfor

new_data_item{1} = y;


==================================
Generates Predictions
==================================

M = 2000;

missing_data_vector = M+1:N;

tic;

[category_index predicted_vector final_delta min_difference predicted_vector_array]
= predict_best_fit_tree_N(anchor_tree, delta_tree, new_data_item, 1, 1,
missing_data_vector, N);

toc

=======================================
Displays Average Path of All Predictions
=======================================


num_predictions = size(predicted_vector_array,2)

average_path = zeros(1,N);
```

```
for i = 1 : num_predictions

temp = predicted_vector_array{i};

average_path = average_path + temp;

endfor

average_path = average_path/num_predictions;

figure

hold

plot(1:N,average_path)
plot(1:N, predicted_vector)
plot(1:N,y)
legend('average path','best-fit path', 'actual path', 2)


=====================================================
GENERATES BAD DATA
=====================================================

clear y

predicted_vector_array = {};

y(1) = 1;

for i = 2 : N

if(rand() >= .55)

y(i) = y(i - 1) + sin(pi/4);

else

y(i) = y(i - 1) - sin(pi/4);

endif

endfor

new_data_item{1} = y;


================================
```

```
Displays Predicted Vector Array
================================


num_predictions = size(predicted_vector_array,2);

figure, plot(1:N, y)

figure

hold on

for i = 1 : num_predictions

temp = predicted_vector_array{i};

plot(1:N,temp)

endfor




=====================================================
CALCULATES ACCURACY OF PREDICTION
=====================================================

cnt = 1;

clear avg_vector

for M = 100: 100: 9900

missing_data_vector = M+1:N;

[category_index predicted_vector final_delta min_difference predicted_vector_array]
= predict_best_fit_tree_N(anchor_tree, delta_tree, new_data_item, 1, 1,
missing_data_vector, N);

avg_percent_error = 0;

for i = M + 1 : N

avg_percent_error = avg_percent_error + abs(y(i) - predicted_vector(i))/abs(y(i));

endfor

avg_vector(cnt) = avg_percent_error/(N - M);
```

```
cnt = cnt + 1;

endfor

figure, plot(100: 100: 9900, 100*avg_vector)


=========================================================
TESTS NEW INPUTS FOR ACCURACY
=========================================================

num_new_items = 25;

for k = 1 : num_new_items

y(1) = 1;

for i = 2 : N

if(rand() >= .48)

y(i) = y(i - 1) + sin(pi/4);

else

y(i) = y(i - 1) - sin(pi/4);

endif

endfor

new_data_array{k} = y;

endfor


figure

hold

for k = 1 : num_new_items

new_data_item{1} = new_data_array{k};

cnt = 1;

clear avg_vector
```

```
for M = 100: 100: 9900

missing_data_vector = M+1:N;

[category_index predicted_vector final_delta min_difference predicted_vector_array]
= predict_best_fit_tree_N(anchor_tree, delta_tree, new_data_item, 1, 1,
missing_data_vector, N);

avg_percent_error = 0;

for i = M + 1 : N

avg_percent_error = avg_percent_error + abs(y(i) - predicted_vector(i))/abs(y(i));

endfor

avg_vector(cnt) = avg_percent_error/(N - M);

cnt = cnt + 1;

endfor

plot(100: 100: 9900, 100*avg_vector)

endfor
```