2次元時間のデータ表(おまけ)

17ec084 平田智剛

# 2次元時間とは

時系列の本質は、列の名の通り列データである。正方形の対角線上に終端がくるように複製していけば、行が進むのと列が進むのが全く同じこととなり、すなわち2次元時間のデータ表となる。このデータ表を解析すること、つまり多量の列を使って、特定の行と、任意の行の関係性を見出し、それを関係性が強い順に並べることとは、多量の時間についてのデータがあれば、指定した時間と、任意の時間との関係性を見出せるということを意味する。

すなわち、「3日前の株価が1000円ほど、2日前が980円ほど、1日前が1010円ほどだったら、今日は1030円になりやすい」等の関係性を見出せる。

# からくり

列データを、下が過去、上が現在であるとみよう。(表1)

但し、今日の日付を0日とする。

表1. 列データ

|  |
| --- |
| 0日前 |
| 0日の株価 |
| -1日の株価 |
| -2日の株価 |
| -3日の株価 |
| -4日の株価 |

列数は問答無用で1である。メタデータを除く列数は5である(奇数とすべき)。

表1は表2に等しい。(適宜読み替えのこと)

表2. 表1と同じ列データ

|  |
| --- |
| 0日前 |
| 0日の株価 |
| 1日前の株価 |
| 2日前の株価 |
| 3日前の株価 |
| 4日前の株価 |

表1の、メタデータを除く部分をまずは正方形にする。(表3)

表3. 行を複製しただけの正方形表データ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0日前 | 0日前 | 0日前 | 0日前 | 0日前 |
| 0日の株価 | 0日の株価 | 0日の株価 | 0日の株価 | 0日の株価 |
| -1日の株価 | -1日の株価 | -1日の株価 | -1日の株価 | -1日の株価 |
| -2日の株価 | -2日の株価 | -2日の株価 | -2日の株価 | -2日の株価 |
| -3日の株価 | -3日の株価 | -3日の株価 | -3日の株価 | -3日の株価 |
| -4日の株価 | -4日の株価 | -4日の株価 | -4日の株価 | -4日の株価 |

これを、表4のように、1日ずつ前にずらして左右に複製したらどうだろうか。

表4. 行をずらして複製した直角二等辺三角形表データ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0日前 | 1日前 | 2日前 | 3日前 | 4日前 |
| 0日の株価 | -1日の株価 | -2日の株価 | -3日の株価 | -4日の株価 |
| -1日の株価 | -2日の株価 | -3日の株価 | -4日の株価 |  |
| -2日の株価 | -3日の株価 | -4日の株価 |  |  |
| -3日の株価 | -4日の株価 |  |  |  |
| -4日の株価 |  |  |  |  |

あるいは、表5のようにしてもよい(かもしれない)。

表5. 列をずらして複製した小さな正方形表データ

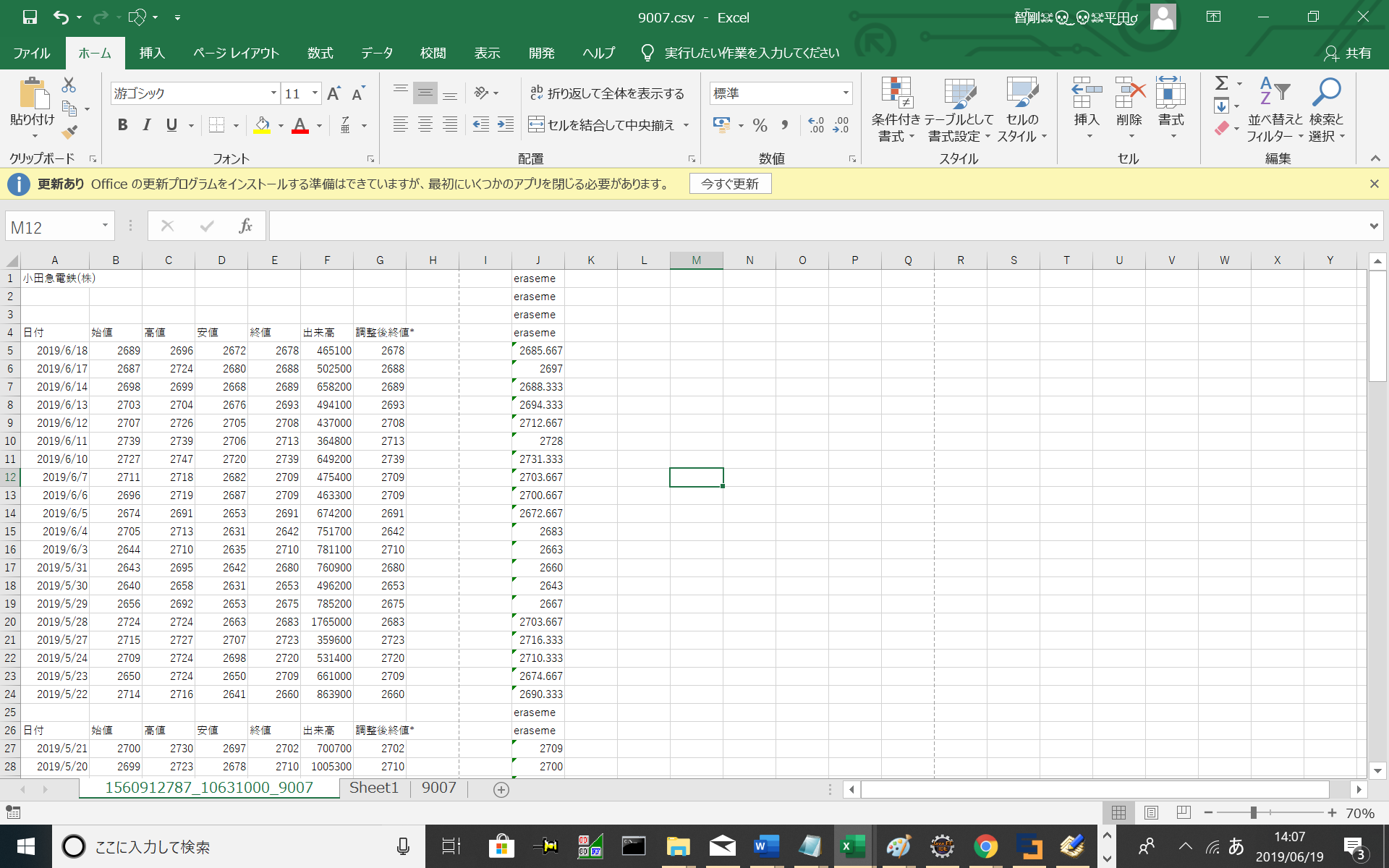
|  |  |  |
| --- | --- | --- |
| 0日前 | 1日前 | 2日前 |
| 0日の株価 | -1日の株価 | -2日の株価 |
| -1日の株価 | -2日の株価 | -3日の株価 |
| -2日の株価 | -3日の株価 | -4日の株価 |

「0日前」属性を予測する決定木を作ることで、何日前がどんな株価の時、0日前がどのような株価になるのかを予測することができるのではなかろうか。

# 実際にやってみた

9007 小田急電鉄(株) 東証一部の、令和元年6月19日から、同年元旦までの株価を日足で得た。このことは、<http://wakash.com/cgi-bin/get_kabuka/select_meigara.php> からできた。

そのままのCSVデータでは、終値、始値などがあって株価が必要以上に複雑で、しかも途中で改行があったり、属性名を改めて示したりしていて、そのままでは扱いにくい。(図1)



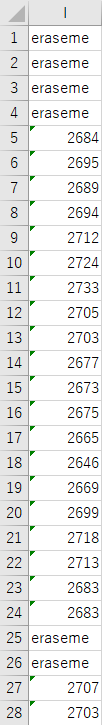
(行続く)

図1. 株価の時系列データ

そこで、I1セル(I列1行)に次のように入力し、十分に下へオートフィルする。

|  |
| --- |
| =  IFERROR  (  ROUND  (  AVERAGE(B1:E1)  ,  0  )  ,  "eraseme"  ) |

これは「同じ行の終値、始値などを平均し、整数に丸めよ。でももし終値などが見つからなかったら、erasemeと表示せよ。」というものである。これに従うと、I列は図2のようになる。



(続く)

図2. 時系列データのI列

これをコピーし、別のシートのA1セルで右クリックして、「行列入れ替え」をして、CSVで保存する。すると、次のようなテキストファイルを得る。

|  |
| --- |
| eraseme,eraseme,eraseme,eraseme,2684,2695,2689,2694,2712,2724,2733,2705,2703,2677,2673,2675,2665,2646,2669,2699,2718,2713,2683,2683,eraseme,eraseme,2707,2703,2734,2715,2667,2618,2607,2590,2596,2627,2625,2621,2590,2563,2563,2504,2479,2513,2538,2556,eraseme,eraseme,2575,2539,2529,2514,2536,2565,2583,2601,2648,2704,2708,2677,2668,2689,2678,2618,2653,2638,2638,2669,eraseme,eraseme,2657,2643,2636,2661,2617,2616,2591,2595,2607,2626,2612,2610,2606,2576,2560,2551,2543,2525,2486,2462,eraseme,eraseme,2413,2409,2429,2403,2399,2421,2458,2462,2463,2447,2459,2434,2431,2420,2450,2465,2503,2511,2512,2511,eraseme,eraseme,2498,2513,2516,2530,2502,2490,2506,2511,2405 |

先頭に一つだけ「,」を追加し、メモ帳の「置換」機能などで「,eraseme」を空文字に置き換える処理を一括で行う。最後に先頭の「,」を取り除くと、次のようになる。

|  |
| --- |
| 2684,2695,2689,2694,2712,2724,2733,2705,2703,2677,2673,2675,2665,2646,2669,2699,2718,2713,2683,2683,2707,2703,2734,2715,2667,2618,2607,2590,2596,2627,2625,2621,2590,2563,2563,2504,2479,2513,2538,2556,2575,2539,2529,2514,2536,2565,2583,2601,2648,2704,2708,2677,2668,2689,2678,2618,2653,2638,2638,2669,2657,2643,2636,2661,2617,2616,2591,2595,2607,2626,2612,2610,2606,2576,2560,2551,2543,2525,2486,2462,2413,2409,2429,2403,2399,2421,2458,2462,2463,2447,2459,2434,2431,2420,2450,2465,2503,2511,2512,2511,2498,2513,2516,2530,2502,2490,2506,2511,2405 |

これを改行し、次の行に複製したら、先頭1つを削除する。

|  |
| --- |
| 2684,**2695,2689,2694,2712,2724,2733,2705,2703,2677,2673,2675,2665,2646,2669,2699,2718,2713,2683,2683,2707,2703,2734,2715,2667,2618,2607,2590,2596,2627,2625,2621,2590,2563,2563,2504,2479,2513,2538,2556,2575,2539,2529,2514,2536,2565,2583,2601,2648,2704,2708,2677,2668,2689,2678,2618,2653,2638,2638,2669,2657,2643,2636,2661,2617,2616,2591,2595,2607,2626,2612,2610,2606,2576,2560,2551,2543,2525,2486,2462,2413,2409,2429,2403,2399,2421,2458,2462,2463,2447,2459,2434,2431,2420,2450,2465,2503,2511,2512,2511,2498,2513,2516,2530,2502,2490,2506,2511,2405***(改行)*  **2695,2689,2694,2712,2724,2733,2705,2703,2677,2673,2675,2665,2646,2669,2699,2718,2713,2683,2683,2707,2703,2734,2715,2667,2618,2607,2590,2596,2627,2625,2621,2590,2563,2563,2504,2479,2513,2538,2556,2575,2539,2529,2514,2536,2565,2583,2601,2648,2704,2708,2677,2668,2689,2678,2618,2653,2638,2638,2669,2657,2643,2636,2661,2617,2616,2591,2595,2607,2626,2612,2610,2606,2576,2560,2551,2543,2525,2486,2462,2413,2409,2429,2403,2399,2421,2458,2462,2463,2447,2459,2434,2431,2420,2450,2465,2503,2511,2512,2511,2498,2513,2516,2530,2502,2490,2506,2511,2405** |

(あるいはエクセルで開いて同じようになるようコピペする)

それを、最後の行が1列だけになるまで繰り返す

すると、次のような直角二等辺三角形表データを得る(図3)

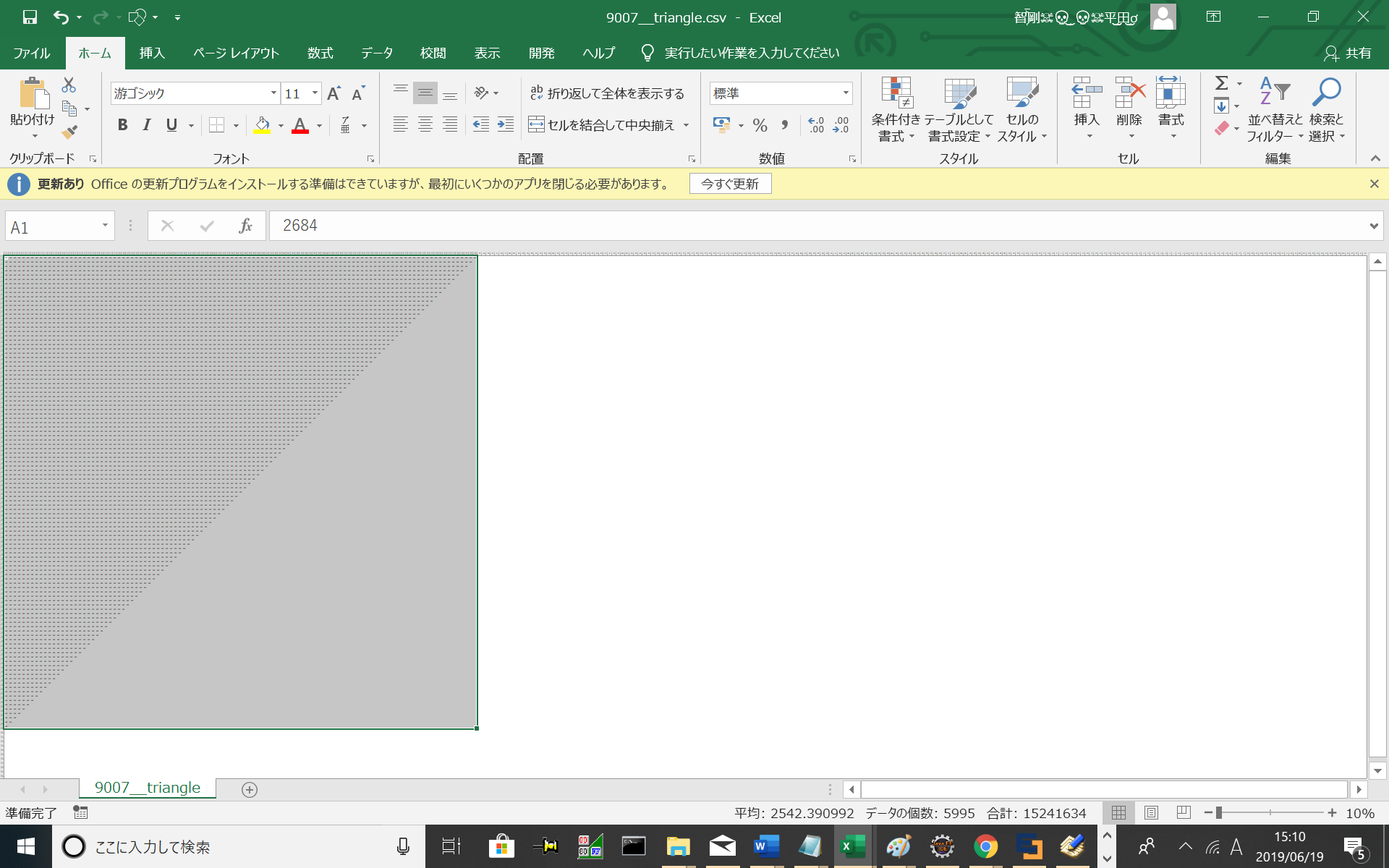


図3 直角二等辺三角形表データ

さらに、一番上に行を挿入し、A1セルに「0日前」と入力し、1辺の長さだけ右にオートフィルする。

これをラピッドマイナーに読み込む。(図4)

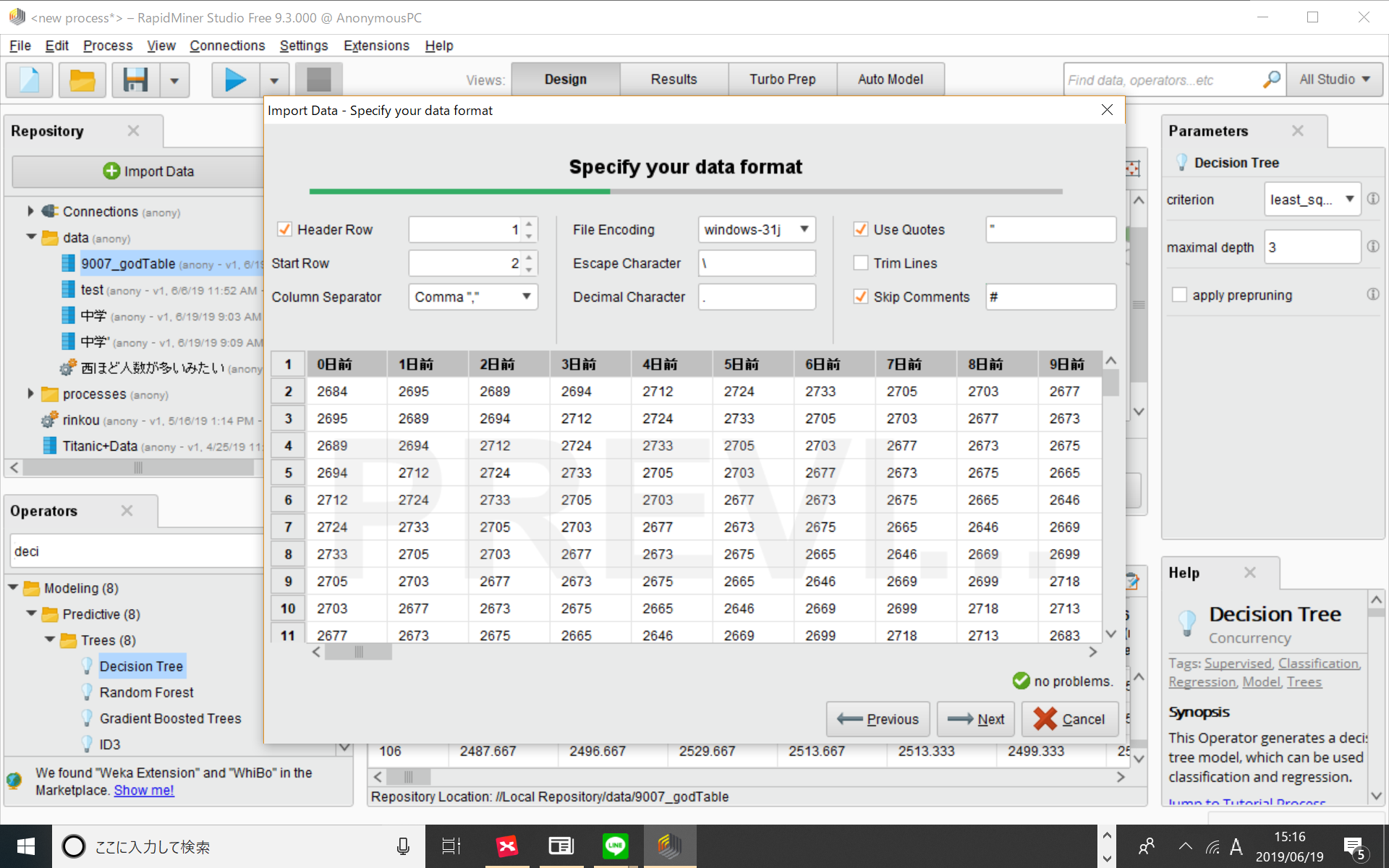


図4. 直角二等辺三角形表データ

このように読み込んだデータを用いて図5のようにプロセスを組む。但し9007\_triangleは読み込んだデータの名前でSet Roleで「0日前」属性にlabelを付け、Decision Treeでは(見やすさのため)最大の深さを3とした。すると、図6を得た。

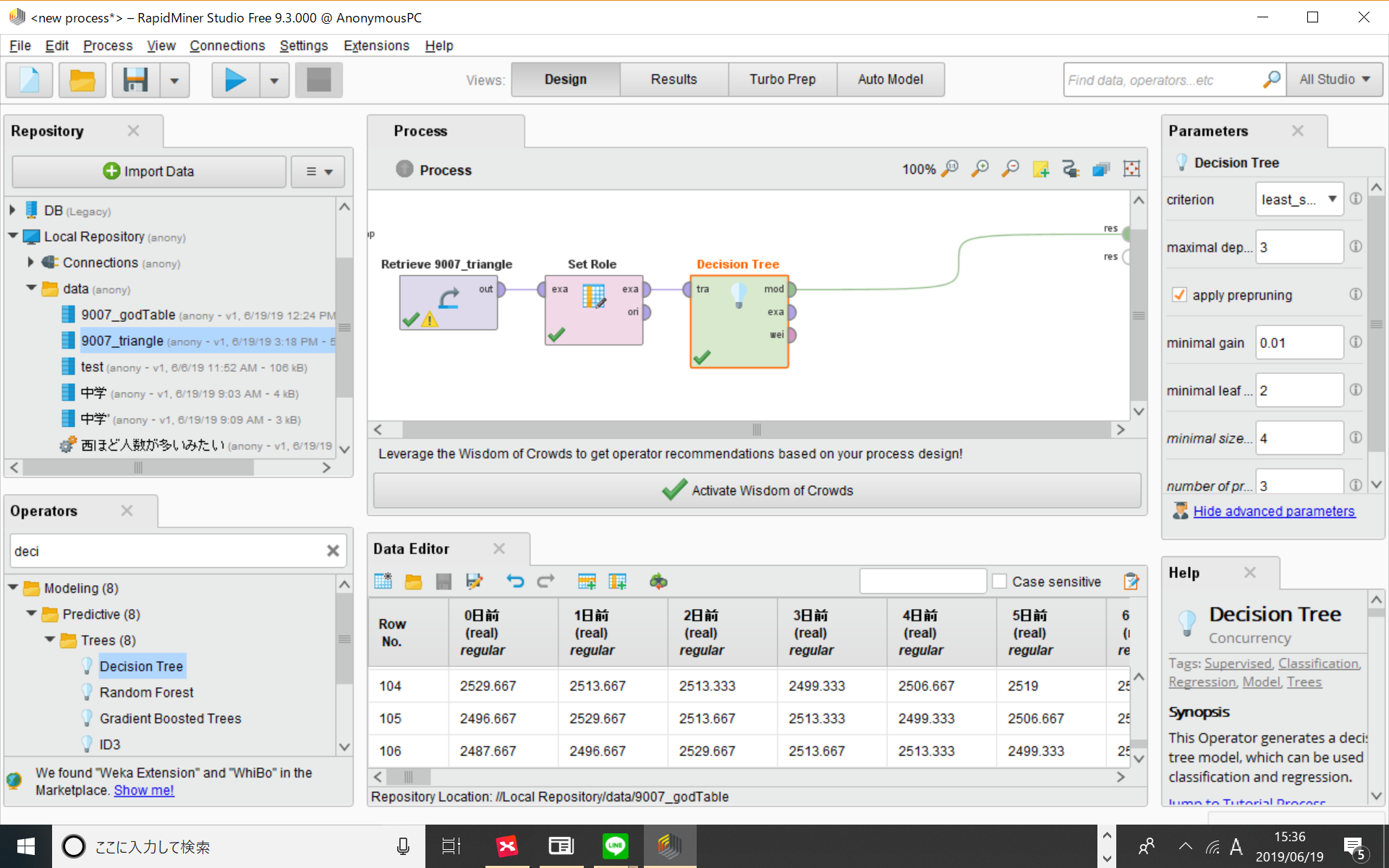


図5. 構成したプロセス

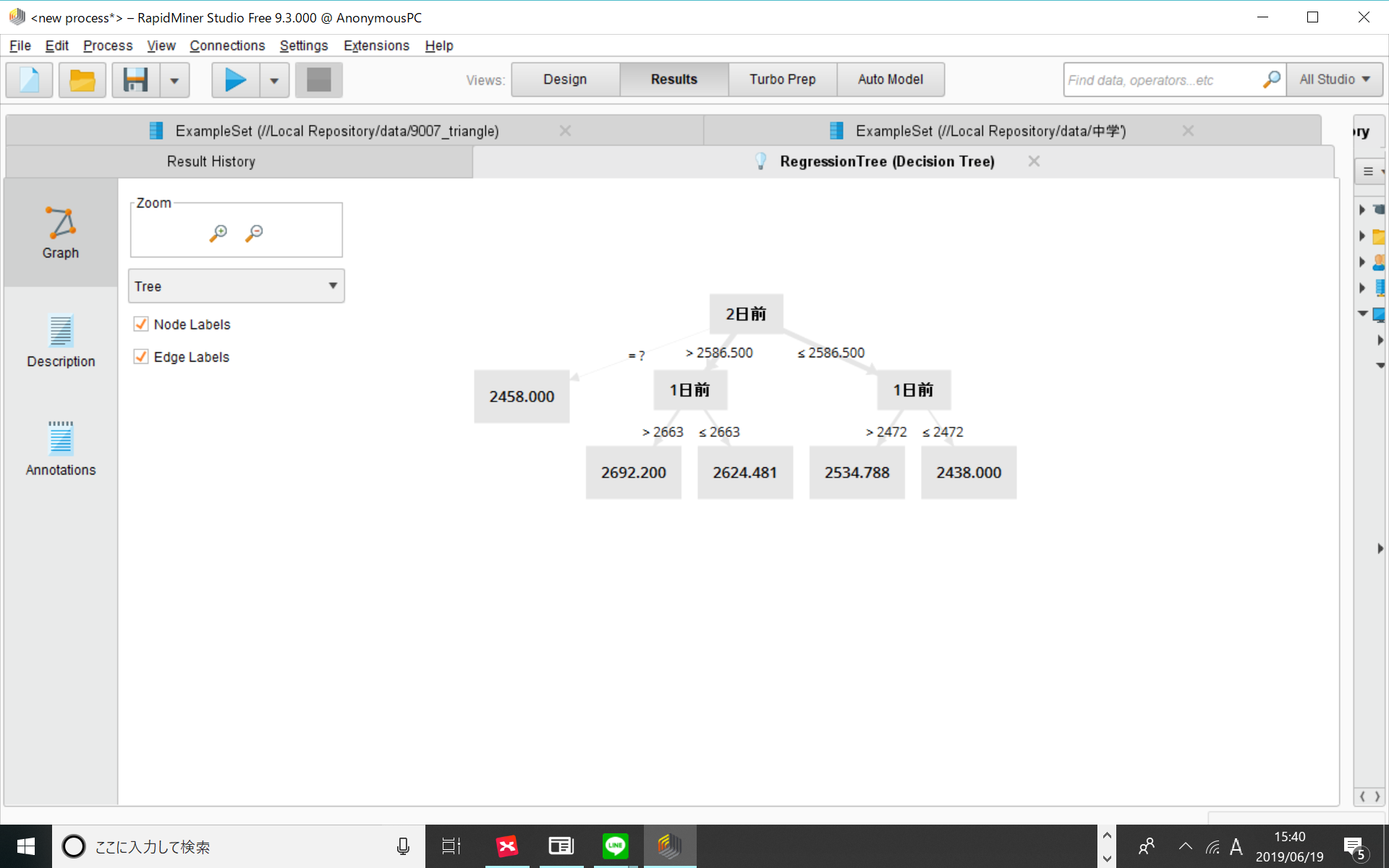


図6. 得られた決定木

図6の決定木から、例えば2586.5円以上→2663円以上と続いた翌日はだいたい2692.200円になりやすいということがわかった。

# デイトレード用のデータマイニング1

これだけでは、条件がそろいにくく、実用性が低い。そこで、増減に注目したマイニングを行おう。それぞれ前日との差をとればよい。新しくシートを作り、旧いシートを参照して差をとって表示すれば容易に作れる。

デイトレードが目的なので、属性選択演算子を用いて、1~3日前のもののみを用いるようにしてよいと思う。

面白い法則は次のようなものが見つかった(図7～8)

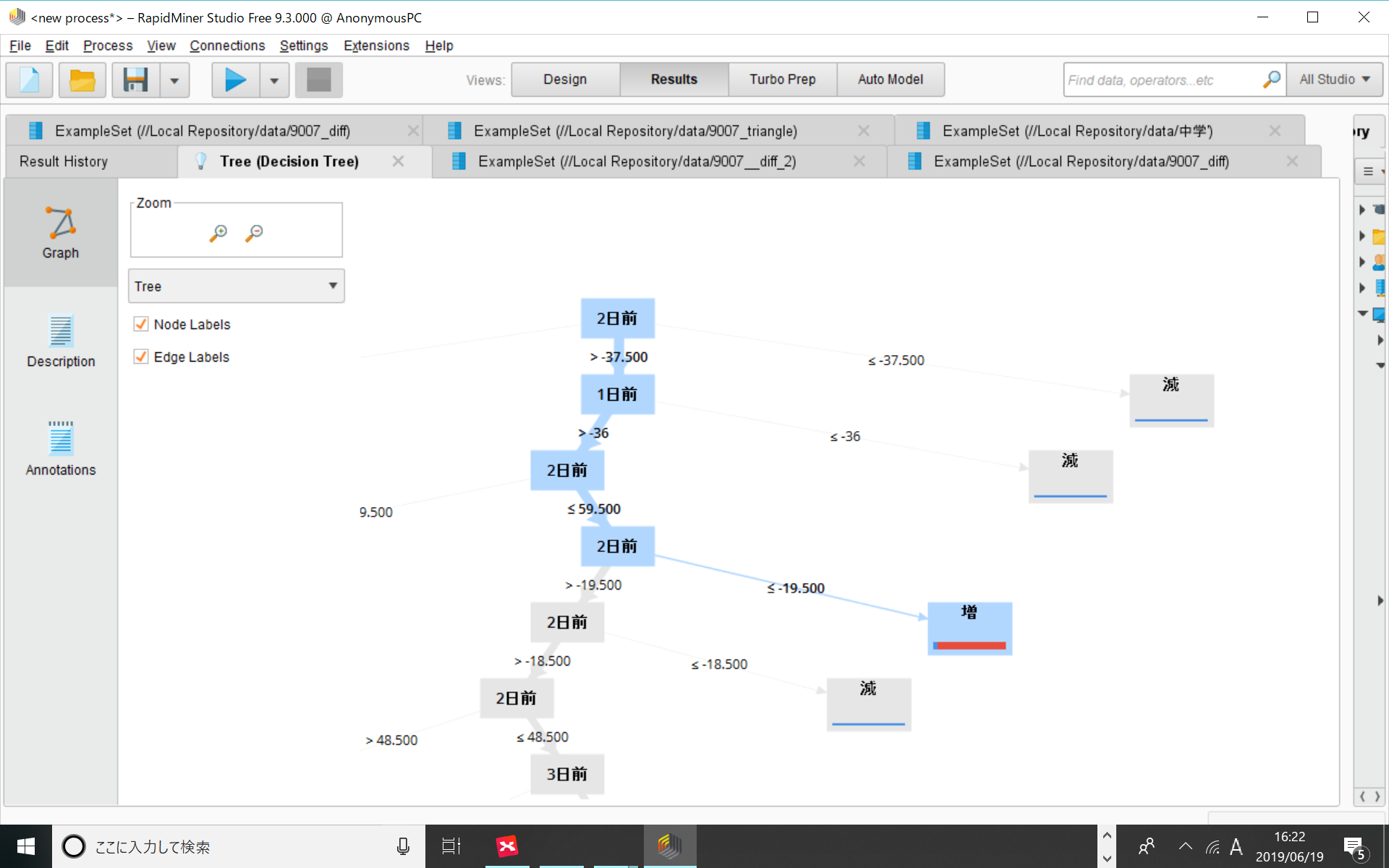


図7. 面白い決定木その1

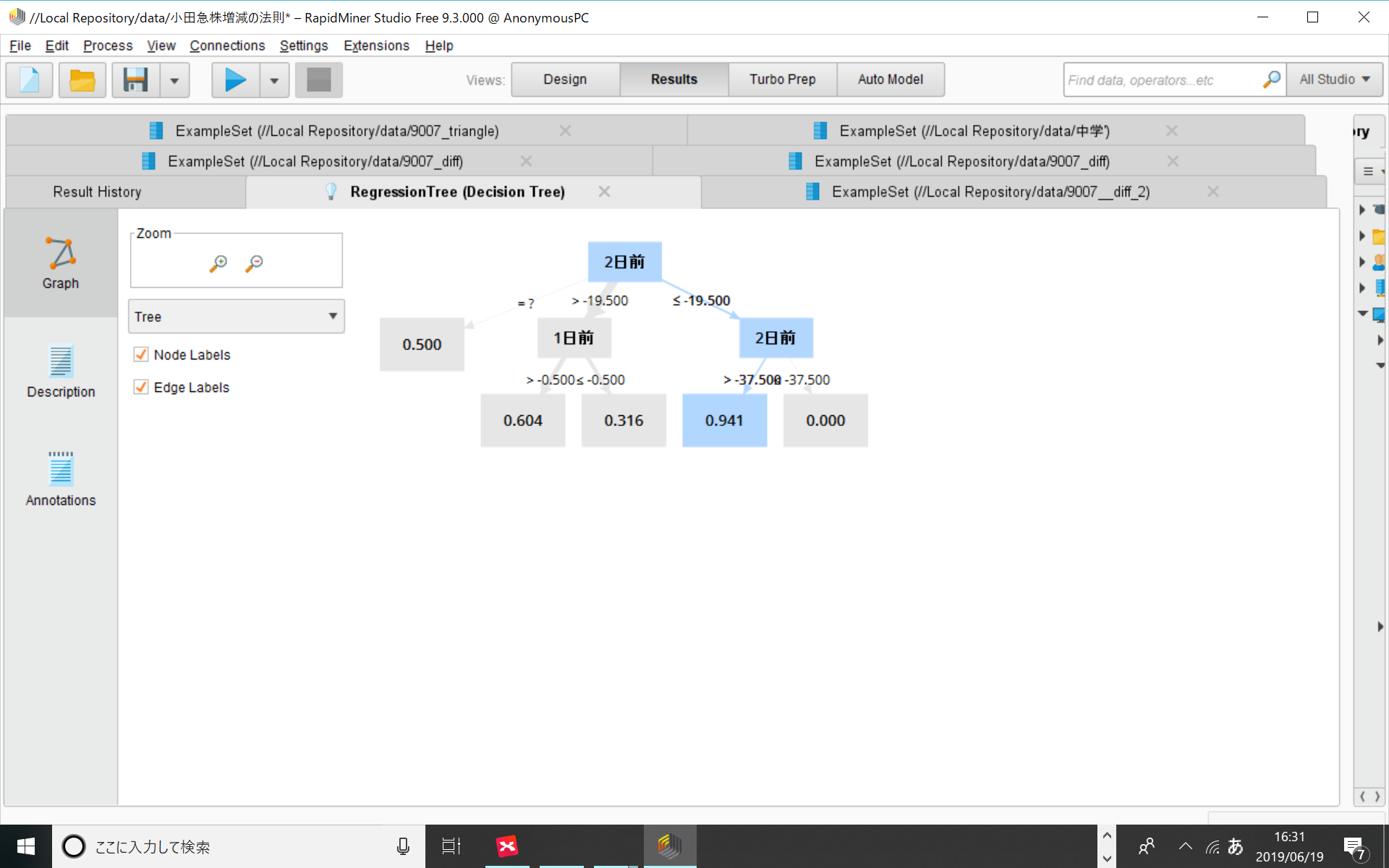


図8. 面白い決定木その2

図7「-37.5<2日目の前日比<=-19.5なら、 17回中16回増えた」

図8「図7の言う通り(94.1％の確率で増える)」

# デイトレード用のデータマイニング2

出来高の大きさが何日後にどう影響するか見てみよう。

前日比の方のエクセルデータ(107日前まである)に、出来高(107日前以前)の列を加えて読み込みなおせばよい。この「107日前」を「0日前」にずらして考えれば、最大で107日後までの未来予測ができたことになる。