RapidMiner チュートリアルとその解説

東京電機大学工学部3年   
情報通信工学科  
17EC084 平田智剛

# 第1章 目的

本稿はデータマイニングソフトRapidMinerのtutorialの英文を和訳し、同時に専門用語に関して補足説明をしていくことで、データマイニングソフトが何をどうするためのものであるか、解説する目的で執筆されたものである。

# 第2章 RapidMinerとは

## 2-1. RapidMinerの用途

RapidMinerは、データマイニングを行うソフトウェアである。複雑なテキスト入力をほとんど必要とせず、ブロックを組み立てることで、直観的、視覚的に操作できることが特徴といえるだろう。

データマイニングとは、統計学、パターン認識、人工知能、機械学習などデータ解析の技法を大量のデータセットに対して適用し、異常値、パターン、相関を発見するプロセス\*のことである。[1]

以下、RapidMinerの公式ホームページ[2]からの引用である。

RapidMiner（ラピッドマイナー）はプログラムの知識がなくても、データサイエンティストが行うような高度な分析業務をドラッグ&ドロップの簡単な操作で行うことができる、世界中で使われている機械学習プラットフォームです。

データ加工（前処理）からデータ可視化、モデル作成、評価、運用までを一つのプラットフォームで行うことができるのが特徴です。

## 2-2. RapidMinerの導入方法

以下の手順で、無料でRapidMiner Free版を導入することができた。  
1. ブラウザでhttps://rapidminer.com/get-started/ へアクセスした。

2. 右上にあるGET STARTED をクリックした。

3. 「Start your free 30-day trial」と書かれていることを確認の上、フォームに大学のメールアドレスを入力[3]し、job FunctionではAcademic / Researchを選択し、IndustryではOtherを選択した。その後DOWNLOADをクリックした。

# 第3章 Tutorialsの解説

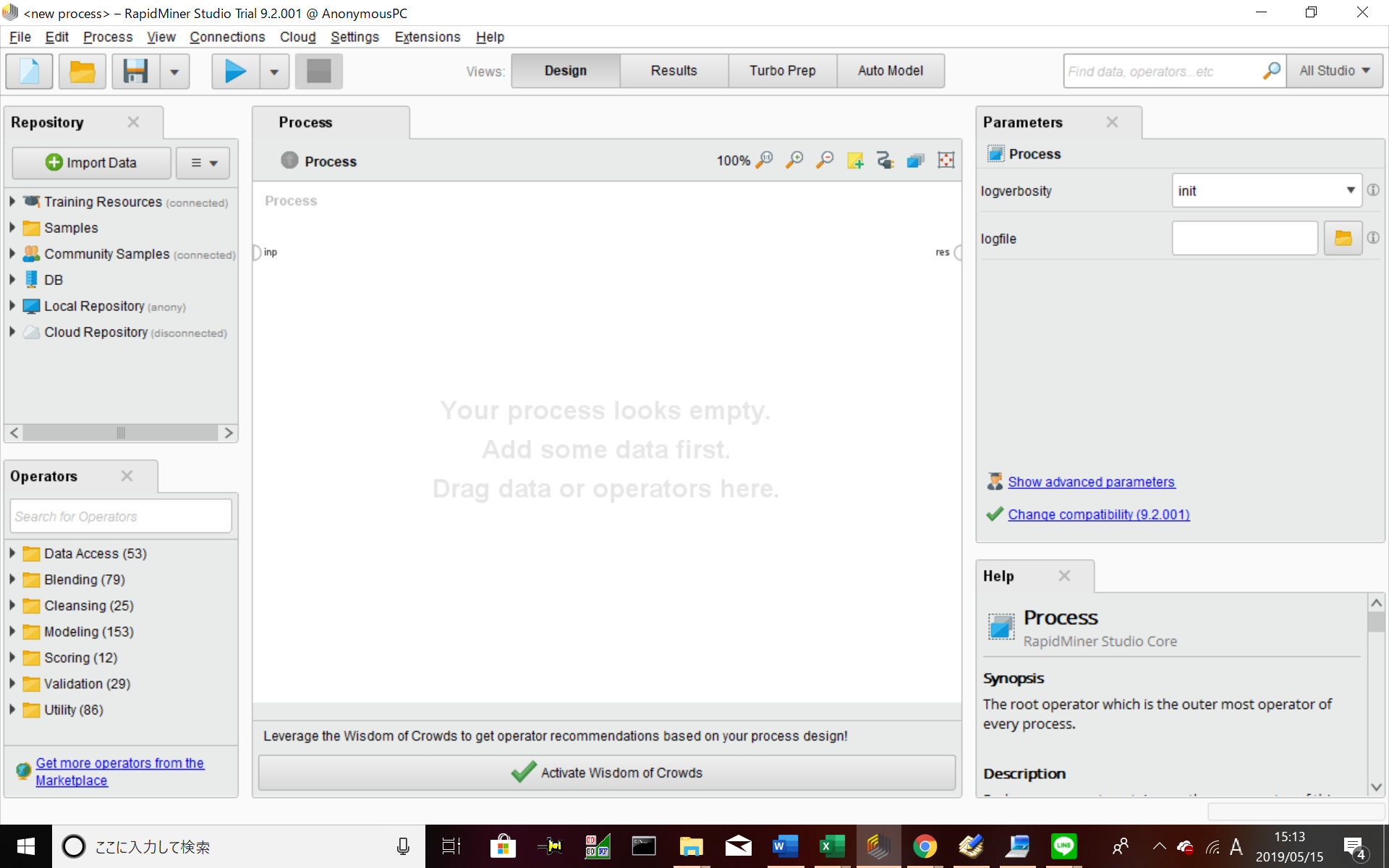
## 3-1.Tutorialsとは

RapidMinerをダウンロード、インストールしたのち起動すると、図3.1.1のように、Startタブ、Recentタブ、Learnタブを持つ画面が表示される。この画面が表示されず、図3.1.2の画面が表示された場合、HelpタブからTutorialsを選ぶことで、図3.1.1の画面を得る。

図3.1.1のLearnタブをクリックすると、Get started(0/8)、Prepare data(0/6)などのメニューが表示される。これらに番号Aを振る(Get startedを1、Prepare dataを2・・・とする)。各メニューをクリックすると、細分化されたメニューがすぐ右隣りに表示される。(ここに振られている番号をB)とする。細分化されたメニューをクリックすると、図3.1.2の画面が変化し、図3.1.3のようなTutorialsパネルが左側に表示される。図3.1.3のような、ウィンドウのようなものを、RapidMinerではパネルという[4]。



図3.1.1 Welcome to RapidMiner Studioウィンドウ

  
図3.1.2 RapidMinerのメインウィンドウ

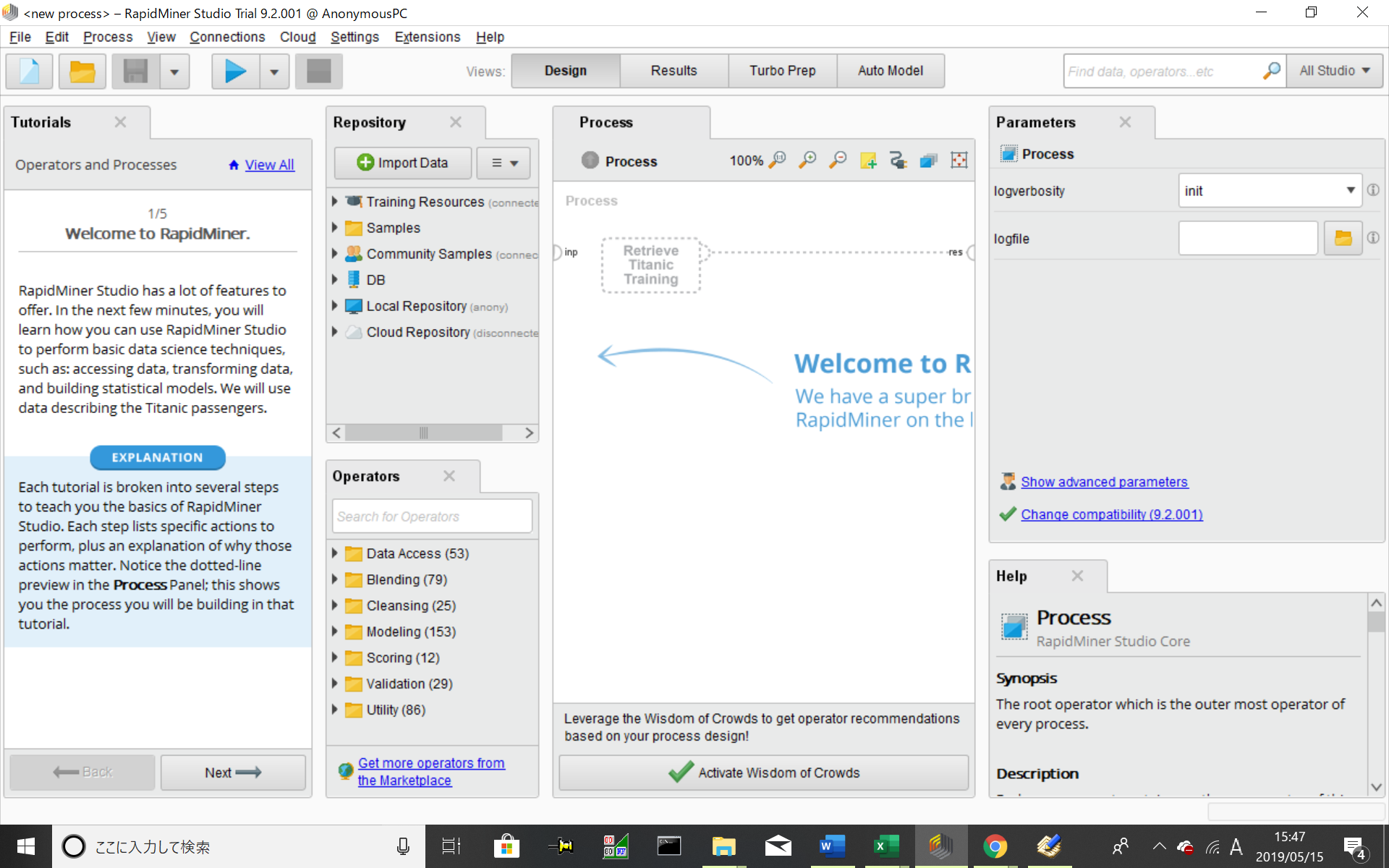


図3.1.3 Tutorials パネルの例

Tutorialsパネルでは、選択された図3.1.1のLearnタブで選んだものに応じて、RapidMinerを実際に動かしながら、このソフトに何ができるのかを紹介する文章が表示される。本稿ではこの内容をTutorials A-Bと呼ぶ。(AとBは前述した番号である)

## 3-2.Tutorialsの和訳と解説

Tutorialsは英語で書かれていて、しかも専門用語の解説がないため、  
データサイエンス\*(以下、「\*」は「用語等解説」にて解説されている単語であることを示す)の入門者には少々不便である。そこで、本稿ではTutorialsの和訳を行うと同時に、専門用語をその都度解説していく。

### Tutorials1-1 Operators and Processes

#### 原文と和訳

Welcome to RapidMiner.

ようこそRapidMinerへ

RapidMiner Studio has a lot of features to offer.

RapidMiner Studio(デスクトップ版RapidMinerの正式名称)はたくさんの機能を提供します。

In the next few minutes, you will learn how you can use RapidMiner Studio to perform basic data science techniques, such as: (～～～)

これから、RapidMiner Studioを使って(～～～)などといった基本的なデータサイエンスを体験する方法を学んでいきましょう。

accessing data, transforming data, and building statistical models.

データのアクセス、変換、統計モデルの構築

explanation

説明

Each tutorial is broken into several steps to teach you the basics of RapidMiner Studio.

各チュートリアルはRapidMinerの基本を(わかりやすく)説明するため、数ステップに分かれています。

Each step lists specific actions to perform, plus an explanation of why those actions matter.

各ステップでは、実行していただきたいことが、目的と合わせてリストされています。

Notice the dotted-line preview in the Process Panel; this shows you the process you will be building in that tutorial.

Process\* パネルの中の点線で描かれているものに注目してください。この点線は、このチュートリアルで構成してほしいブロックを示しています。

Retrieve data.

データの取り込み

Let’s start by retrieving some data about the Titanic passengers.

(今回は例として、)タイタニック号の乗客のデータを取得してみましょう。

Activity1

やること1(番号は本稿で勝手に振りました)

1.Find the Repository panel on the left of the screen.

1.画面左側にあるRepository\*パネルをみてください。

2.Open the folders Samples, and then data.

2.Samplesフォルダを開き、さらにその下のdataフォルダも開きます

3.Drag the Titanic Training dataset and drop it in the Process panel in the middle of the screen.

3.(dataフォルダの中の)Titanic Training dataset\*を画面中央のプロセス\*パネル(の、点線のところ)までドラッグします。

explanation

説明

Good job! You have just added your first operator to RapidMiner Studio, namely the Retrieve Operator.

　　Activity1の操作により、RapidMiner Studioに「演算子」を追加することができました(※1)。今回の演算子は「データ取得演算子」  
(Retrieve\* 演算子\*)です。

Operators perform actions: in this case, it retrieves data from the Repository.  
演算子は、処理を実行してくれます。今回の場合は、Repositoryからデータを取得します。

Now let’s see what you can do with it!

以降では、演算子がどう働くのか、実際に確認していきます。

Building your first process.

プロセスの構築

As a RapidMiner analyst, you create processes by adding and connecting operators. Operators are connected together via their Ports:

RapidMinerアナリストは、演算子を追加してそれぞれを接続することでプロセスを組み立てます。演算子はポートを介して互いに接続されます。

Activity2

やること2

1.Connect the output port ("out") of Retrieve Titanic Training with the result port ("res") on the right side of the Process panel.

1.Titanic Trainingを取得したデータ取得演算子

(以下、これを「データ取得演算子(Titanic Training)」または  
「Retrieve(Titanic Training)」と表現します)

の “out” ポート(出力ポート)を、Processパネルの右端にある結果ポート( “res” ポート)に接続してください。

2.Make the connection either by dragging a line between the ports, or by first clicking on one port and then on the other port.

2.両ポートの間をドラッグして接続するか、片方のポートをクリックしてそれに続けてもう片方のポートをクリックします。

explanation

説明

Excellent! You have built your first process in RapidMiner Studio.

以上で、RapidMiner Studioにおけるプロセスの構築ができました。

We can now look at the output of the Retrieve operator.

これで、データ取得演算子が出力する内容を見ることができるようになりま  
した。

Whenever you want to see the output of an operator, make sure it is connected to the 'res' port.

演算子の出力を確認したいときは、必ずその演算子を “res” ポートに接続するようにしてください。

Running processes.

プロセスの実行

After we have connected the elements of our process, we can hit the 'Run' button to view the output.

プロセスを正しく接続したら、(データ取得演算子の)出力を確認するために “Run” ボタンを押してください。

Activity3

やること3

1.Click Run (top left) to execute the process.

1.画面左上にある実行ボタン(▶)をクリックし、プロセスを実行します。

explanation

説明

Nice! You have run your first process in RapidMiner Studio.

これで、RapidMiner Studioで「プロセスを実行する」ことができました。

When you click "Run", the operators in your process will perform their actions.

▶をクリックすると、プロセス内の演算子が動作します。

RapidMiner executes the process and shows the data connected to the result port.

そして、“res”ポートに接続されているデータを表示します。

In the center of the Results view, you can see our raw data about the Titanic passengers such as their family size or their age.

Resultsビュー(の中央付近)に、タイタニック号の乗客に関する生データ(家族の人数や年齢など)を確認できます。

On the Statistics tab we get a statistics summary, which provides useful information; for example, 349 passengers survived the Titanic accident.

Statistics(統計)タブで、有用な情報を提供する統計の概要が表示されます。例えば、349人の乗客がタイタニック号の事故で生還したことがわかります。

Congratulations - Halfway there!

半人前になりました。

Well done - you just mastered your first tutorial! To quickly review:

これでTutorial1-1をマスターしました。要点をまとめると次の通りです。

･Operators perform individual actions.

演算子にはそれぞれ固有の働きがあります。

･You can connect operators together to build a Process.

演算子を接続することでプロセスを構成することができます。

･To see the output of an operator, connect it to the 'res' port.

演算子の出力をみたいときは、その演算子を “res” ポートに接続します。

･Running the process performs all operators and automatically shows the results

プロセスを実行すると、すべての演算子を動作させ、その結果をみることができます。

explanation

説明

To become a true master move on to the next tutorial below.

真のマスターになるには、以降のチュートリアルにも取り組んでください。

#### Activity

Activity1を実行した結果、図3.2.1.1.2.1のようになった。

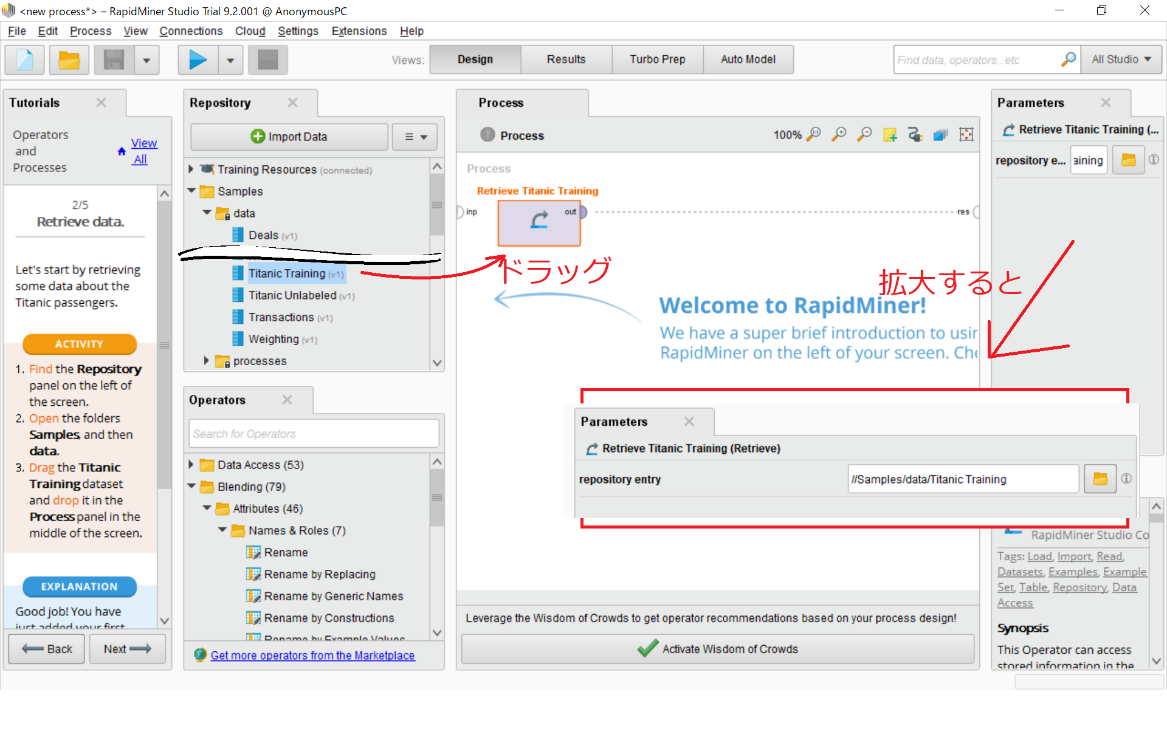
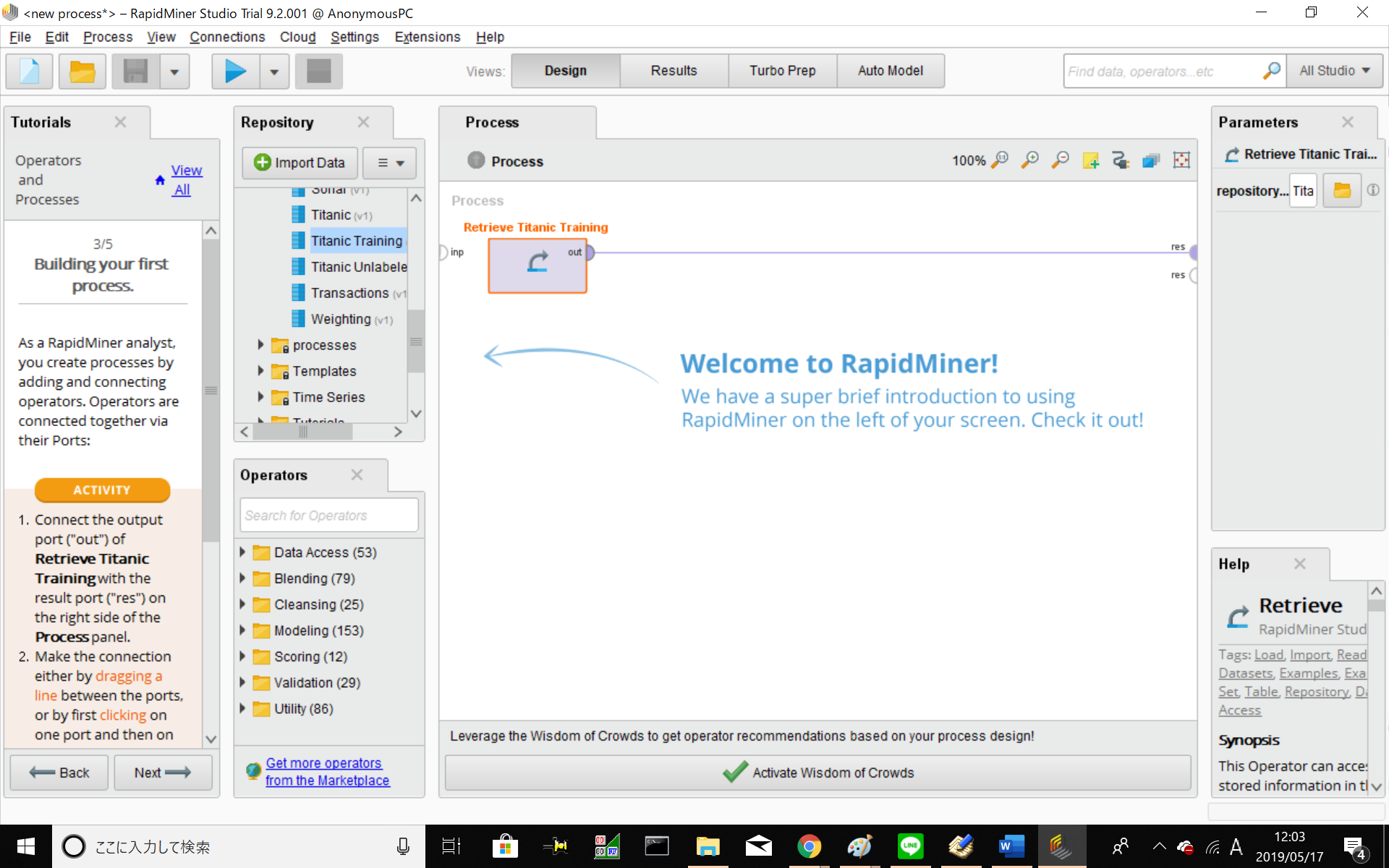


図3.2.1.1.2.1 Tutorial1-1 Activity1 の実行画面

RepositoryパネルからTitanic Training というデータをProcessパネルまでドラッグすると、図3.2.1.1.2.1の通り、データ取得演算子が生成され、その引数はTitanic Training となる。演算子の引数は、その演算子を選択の上、Parametersパネルから確認できる。

　Activity2を実行した結果、プロセスパネルは図3.2.1.1.2.2のようになった。

  
図3.2.1.1.2.2 Tutorial1-1 Activity2 を実行したときのProcessパネル

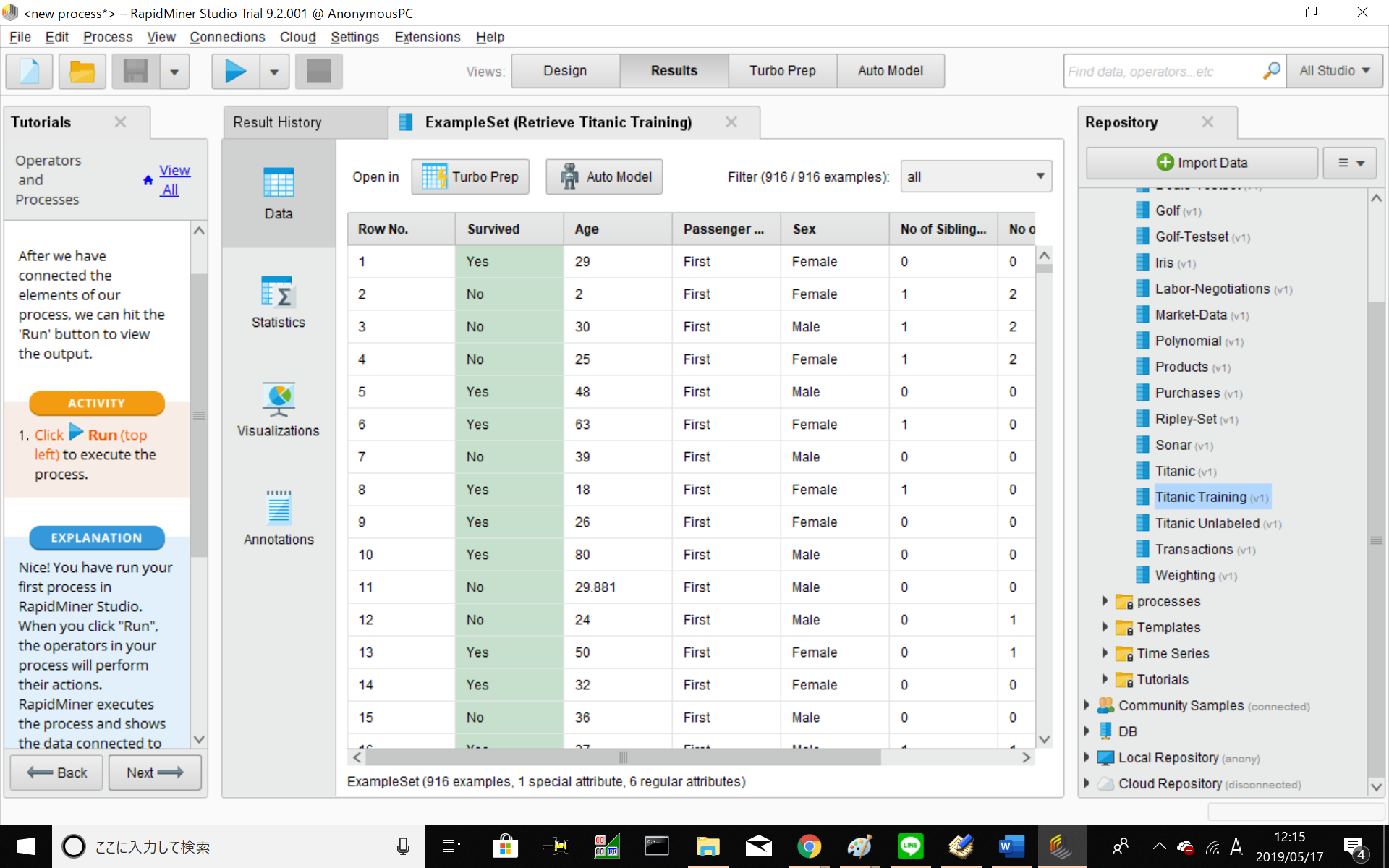


図3.2.1.1.2.3 Tutorial1-1 Activity3 を実行したときの実行画面

図3.2.1.1.2.3ではDataタブが選択されていて、そこでは生のデータ(分析などを施さず、アンケートなどで集計した直後のデータ)が表示されている。  
Row No.は行の番号  
Survivedは生存した(Yes)か否(No)か  
Ageは年齢  
Passenger Classはグレード(JRの電車でいう普通車やグリーン車のようなもの)、  
Sexは性別(Female=女、male=男)  
である。

Statisticsタブに切り替えると、パネルは図3.2.1.1.2.4のようになる。

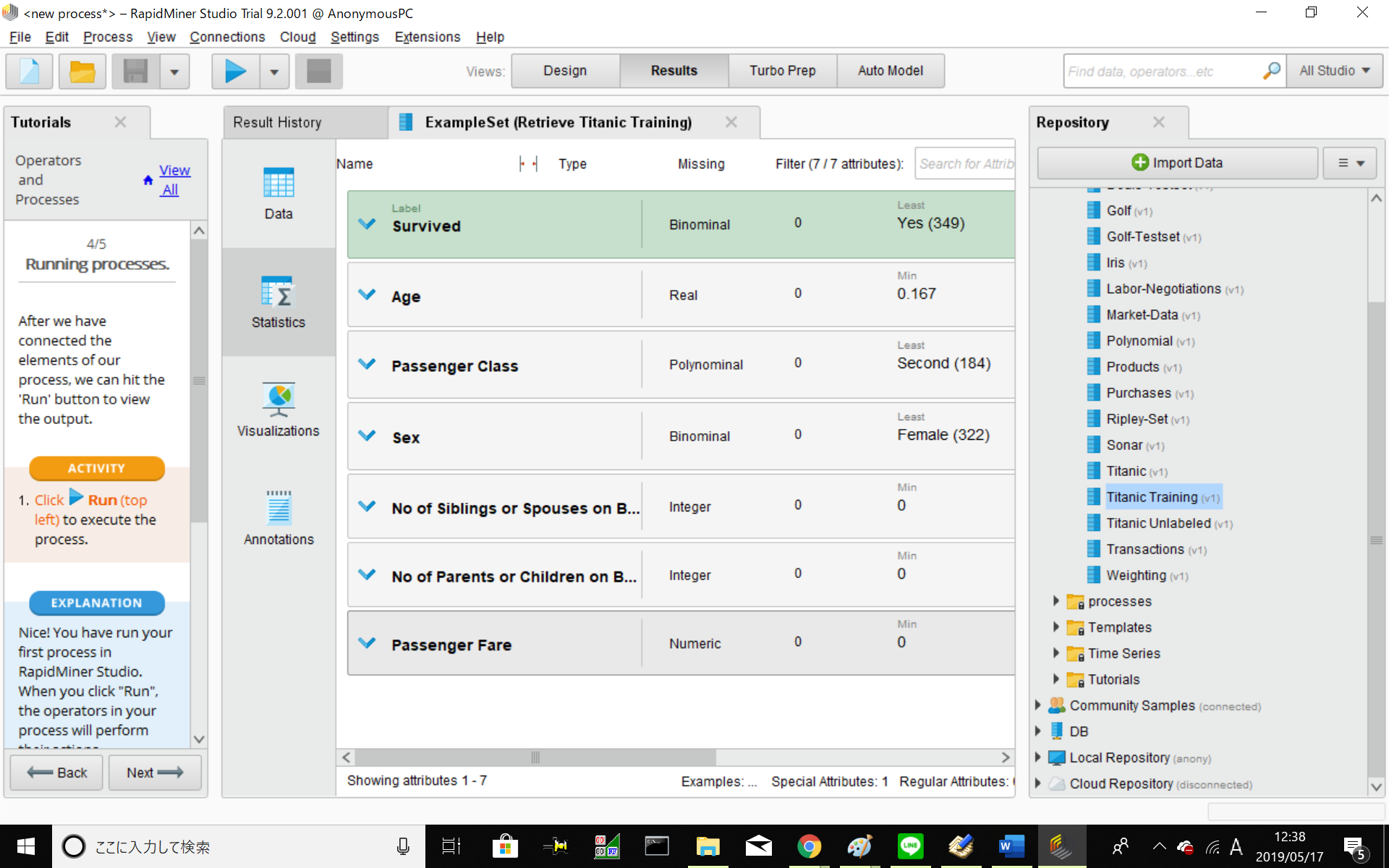
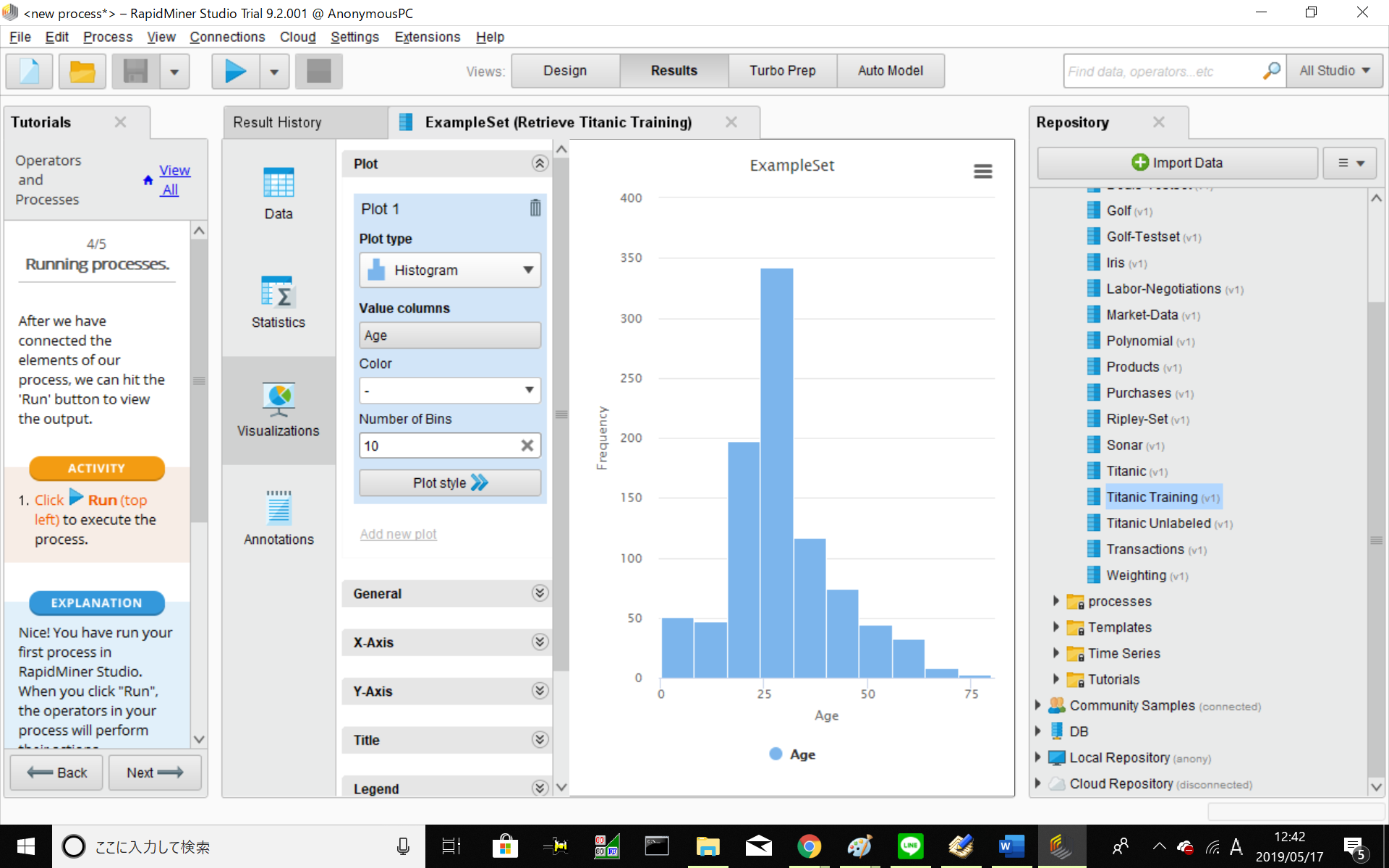
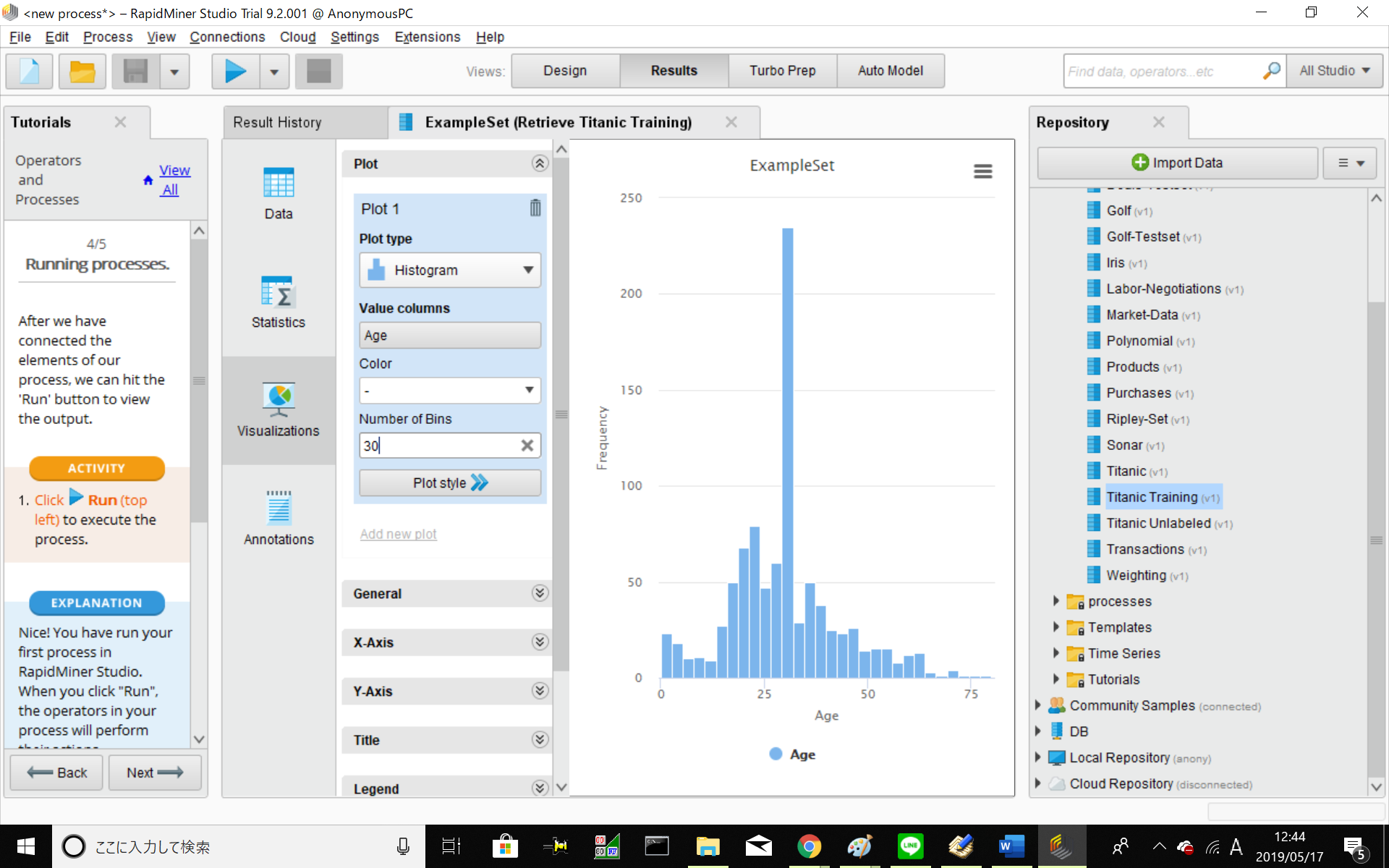


図3.2.1.1.2.4 Statisticsタブの内容

　図3.2.1.1.2.4の画面でAgeをクリックし、open visualizationをクリックすると、図3.2.1.1.2.5のような棒グラフを得る。  
 

(a) (b)

図3.2.1.1.2.5 年齢(Age)の出現数(Frequency)を示す棒グラフ

(a)では、binの値を10に、(b)では30に設定したものである。図と見比べることで、binは棒グラフの棒の本数を示すものと分かる。binを大きくすると、(b)のようにより細かくデータをみることができる。

#### 用語等解説

##### データサイエンス[5][6]

様々なデータの共通点を見つけ、そこから何らかの結論を導くために用いられる学問のこと。デジタル大辞泉[6]による説明を引用すると、次の通り。

データの分析についての学問分野。統計学、数学、計算機科学などと関連し、主に大量のデータから、何らかの意味のある情報、法則、関連性などを導き出すこと、またはその処理の手法に関する研究を行う。これらの研究者および技術者はデータサイエンティストとよばれる。

RapidMinerでは、自動モデリングの機能があり、またデータ可視化の方法も豊富である。さらに本来複雑な処理を、演算子を組み立ててプロセスを構築するだけで実行可能である。つまり数学、統計学、計算機科学の難しさを、ソフトウェアによる自動化が解消しているということもできるだろう。

##### retrieve(リトリブ、レトリブ、データ取得)[7][8]

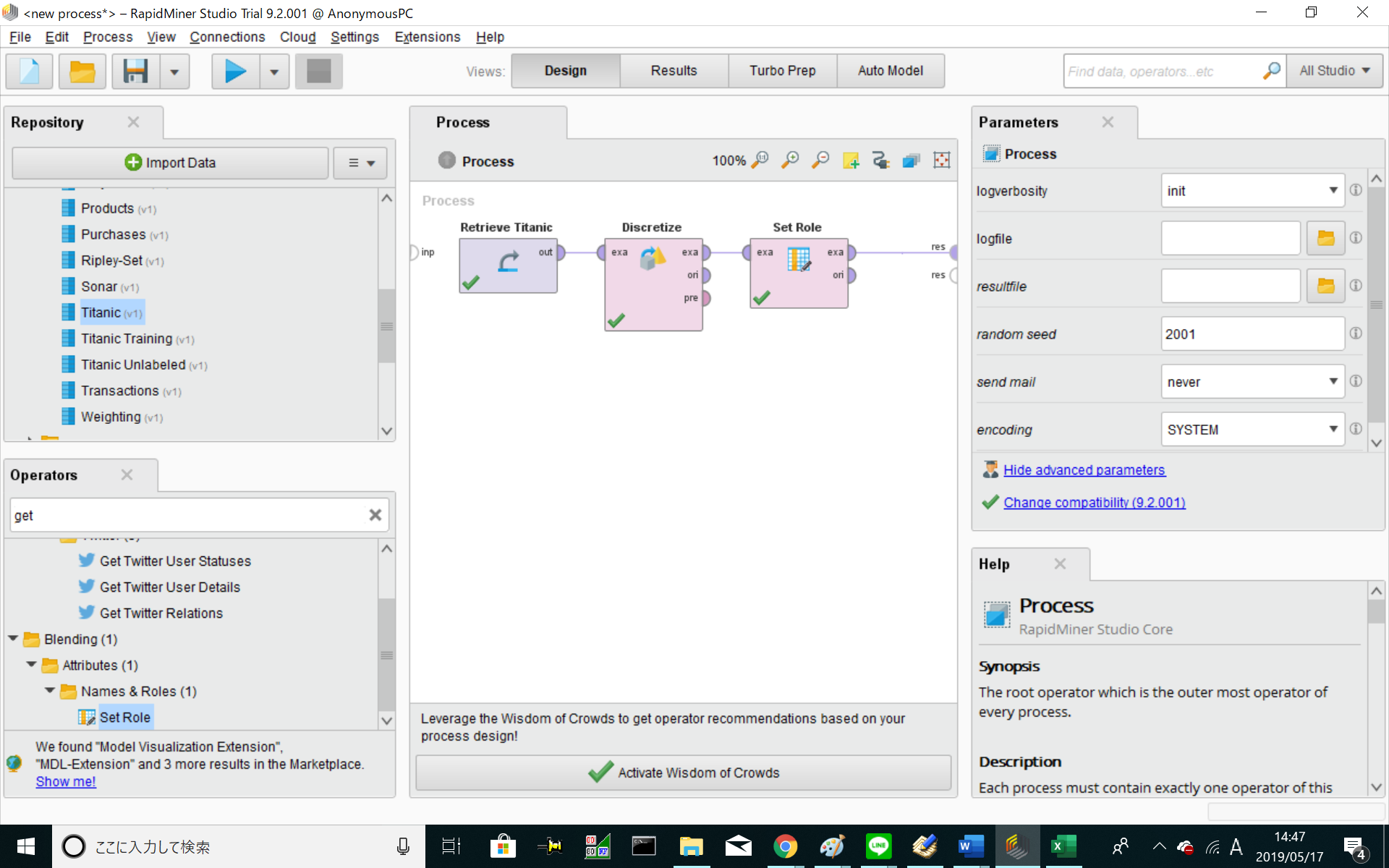
　retrieveは、RapidMinerにおいて、「データ取得」を意味する単語である。

retrieveは「取り戻す」、「回収する」、「救い出す」などを意味する他動詞であるが、目的語がdataなどの場合、「検索する」や「取得する」という意味になる。「マイクロソフト用語集」ではretrieveの意味の一つを「保留解除」と説明している。今、自分の手元にない状態を仮の状態とみなし、その状態を解除することと考えれば「取り戻す」と「検索する」や「取得する」の言葉の共通点が見いだせる。

また、コンピュータに関わる用語は日外アソシエーツが提供する  
「コンピュータ用語辞典」[8]でも解説されている場合がある。

##### process(プロセス)

process(プロセス)は、実行したい処理を記述した意味の塊であり、視覚的にはブロックの塊である。プログラミングにおけるコードのようなものである。それぞれのブロックはポートを介して接続されていて、最終的な出力が “res” ポートに接続されて初めて成立する。プロセスはProcessパネル内で構築される。図3.2.1.1.3.1はプロセスの例である。

  
図3.2.1.1.3.1 Processパネル内に構築されたプロセスの例

##### repository(リポジトリ、レポジトリ)[8]

repositoryとは設計情報を保管するデータベースやディレクトリのことである。

##### training dataset[8]

training datasetは訓練データを意味する単語である。trainingが訓練の意味で、datasetはある規則に則って配列されたデータの集合のことである。

訓練データというのは主に機械学習の教師あり学習で使われる、学習用のデータを意味する。(ここでも同じ意味かは不明)  
　訓練データは欠損値があったり矛盾があったりしてはいけないため(誤ったことを訓練したらどうしようもない)、単純化されている。そのため、我々がRapidMinerやデータサイエンスの練習をするために最適化されたデータのことを意味する単語ととらえて差し支えない場合がある。即ち、Titanic Training dataは、Titanic dataを初心者向けに単純化したデータであるということもできる。

##### operator(オペレータ、演算子)

RapidMinerにおける演算子は、プロセスを構成する要素であり、入力されたデータなどに適切な処理を施して出力を行うものである。

演算子というものの、入力を受け取り処理を行って出力を行う本質は、プログラミングでいうところの関数(メソッド)に似ているものである。

そこで、本稿では演算子やその引数を  
演算子名(引数名=値, 引数名=値, ...)  
のように表現することがある。

##### row

表において、列のこと

##### bin[9]

RapidMinerにおけるbinは棒グラフでいう、棒の本数である。厳密には、数値属性を(等範囲で)離散化するとき、いくつに離散化するかを示す数字のことである。

#### 演算子の説明[9][13]

##### データ取得演算子(Retrieve演算子)

###### 概要

　データ取得演算子はレポジトリに保存された情報にアクセスし、それをプロセスへ読み込むという働きをする。

　アクセスされる情報(RapidMiner Object)はExampleSet、Collection、Modelである。  
(ExampleSetとはデータセット、サンプルセット(表データ)のことであり、Example(サンプル)はデータセットの任意の行を意味する)[13]

この演算子では、メタデータを一緒に読み込むことができるのが特徴である。メタデータは、RapidMiner Objectに関する追加情報で、  
表データの場合、属性の名前、型、範囲、欠落地の数などである。メタデータがあることで、他の演算子のパラメータをプルダウンメニューから選べるようになり、簡単に設定できるようになる。

※1  
Repositoryパネルから演算子(＝operators)を追加したことになるのならOperatorsパネルに並んでいるもの(図3.2.1.1.2.1にみられるRename、Rename by Replacingなど)は演算子ではないのかという疑問が生じるであろう。実はOperatorsパネルにはRetrieveというものも並んでおり、RepositoryパネルからProcessパネルへ、データ(Titanic Training datasetやTitanic datasetなど)がドラッグされた場合、実際にはデータ取得演算子(Retrieve演算子)が追加され、データ取得演算子の引数(Parameters)として、データが指定される。(図3.2.1.1.2.1のParametersパネルでも、Titanic Training datasetが引数repository entryの値に指定されていることが確認できる。)

###### 出力

・出力(out)

引数「repository\_entry」で指定されたパスを持つRapidMiner Object

###### 引数

repository\_entry

読み込みたいRapidMiner Objectのパス

(パスの文法は、参考資料[9]に詳しい)

### Tutorials1-2 Modeling

#### 原文と和訳

Let's do some data science.

データ科学

In the last tutorial, we imported a dataset and ran a process.

Tutorials1-1では、データセットを読み込み、プロセスの実行をしてみました。

We found out that 349 passengers survived the Titanic accident.

その結果、タイタニック号の事故で349名の乗客が生還したことがわかりました。(図3.2.1.1.2.4でsurvivedの行にYes(349)とある)

Let's build a Process to see if those survivors have something in common or not.

これらの生存者に何か共通点があるのか否かを探るべく、プロセスを構築しましょう。

Retrieve the Titanic Training data.

Titanic Training データの取得

Load the Titanic Training data.

Titanic Trainingデータを読み込みます。

Activity1

やること1

1.Drag the Titanic Training data from the Samples repository into the Process panel.

SamplesレポジトリからProcessパネルへTitanic Trainingデータをドラッグします。

explanation

説明  
You have defined the first operation of your process.

これにより最初のオペレータをプロセスに入れることができました。

Now we will learn how easy it is to build a machine learning model on this data.

ここからは、データの機械学習モデルがいとも簡単に構築できてしまうことを見ていきましょう。

Build a decision tree.

決定木の構築

Decision Trees are a popular statistical modeling technique that finds hidden patterns in the data. Let’s build one!

決定木を構築しましょう。決定木はデータに潜む隠れた法則を見つけるためによく使われる統計モデリング手法です。

Activity2

やること2

1. Find the Operators panel on the left.

1. 画面左のOperatorsパネルを見つけてください。

2. Open the folder Modeling, Predictive, and finally Trees.

2. Modeling(モデリング)フォルダ→ Predictive(予測)フォルダ→Trees(木構造)フォルダを開きます。

3. Drag in the Decision Tree operator and drop it after Retrieve Titanic Training.

3. Decision Tree演算子(決定木演算子)を 「データ取得(Titanic Training)」の後ろへドラッグしてください。

explanation

説明

Good, we now have the operator in the process but we still need to connect the two operations and define what we want to see as output before we can run it.

これで、プロセスに必要な演算子が揃いました。ただ、2つの演算子を接続し、また何を出力させたいのか決定する必要があります。

Connect & execute.

接続と実行

The Decision Tree operator will build a Decision Tree for us based on our Titanic Training data.

決定木演算子はTitanic Trainingデータに基づいて、決定木を作ります。

We still need to connect these operators together to build a Process, though:

プロセスを完成させるには、データ取得演算子と決定木演算子を接続する必要があります。

Activity3

やること3

Connect the output port ("out") of Retrieve Titanic Training with the input port ("tra" for "training") of Decision Tree.

データ取得(Titanic Training)の「out」ポートを決定木演算子の「tra」(「訓練」の意味)に接続します。

You can click on the ports, or you can drag on them!

双方のポートをクリックしてもドラッグしても接続できます。

Connect the first output ("mod") of the Decision Tree with the result port ("res") on the right side of the Process panel.

決定木演算子の第1出力である「mod」ポートをプロセスパネルの右端にある「res」ポートに接続します。

Press the ▶ Run button in the toolbar to execute the process.

ツールバーにある▶ プロセスの実行(Run)ボタンを押し、プロセスを実行してください。

explanation

説明

Great job! Your process should now be complete and deliver a decision tree model, which explains to you what most of the survivors and most of the victims had in common.

これで、プロセスは完成しました。そして完成したプロセスは多くの生存者／犠牲者はそれぞれに共通するどんな特徴をもっているのかを説明する決定木モデルを生成しました。

Congratulations!

完了

Great - you just have built your first machine learning model.

機械学習モデルの構築ができました。

This was not hard, was it?

簡単にできたのではないでしょうか。

You now know how to load data, add a machine learning model to the process, and execute both steps.

ここまでで、データの読み込み方や機械学習モデルをプロセスに追加する方法を体得してきました。

explanation

説明

From our resulting decision-tree, we can clearly see that survival was not coincidental at all.

得られた決定木から、乗客の生死は偶然によって決まるものでは全くないということが、はっきりとわかります。

In fact, it tells us that as a female passenger, with a small family, you were really lucky - at least if you were holding an expensive ticket!

女、家族の少ない人、(一応、高価なチケットを持ってる人)は命の選別で選ばれることができます。

These tutorials have only shown you the tip of the iceberg of what you can do with RapidMiner.

この例でお見せしたのは、RapidMinerにできることの氷山の一角にすぎません。

To keep learning and become a RapidMiner Master, click the link below to start the next tutorial.

さらに経験を重ね、RapidMinerのマスターになりましょう。下のリンクをクリックして、次のチュートリアルへ進んでください。

Welcome again to RapidMiner!

改めまして、RapidMinerへようこそ。

#### Activity

Activity1およびActivity2の解説は省略する。

Activity3で、手順2.までを実行した結果、図3.2.1.2.2.1のようになった。



図3.2.1.2.2.1 Tutorial1-2 Activity2 手順2までの実行画面

Operatorパネルでは、演算子を「検索」することができるので、実際には手順にあるようにModelingフォルダ→Predictiveフォルダ→Treesフォルダ→Decision Tree演算子のようにたどっていく必要はない。図3.2.1.2.2.1では演算子名の先頭4文字「deci」をキーワードに検索した。

Activity3で、手順3を実行した結果、図3.2.1.2.2.2のようになった。

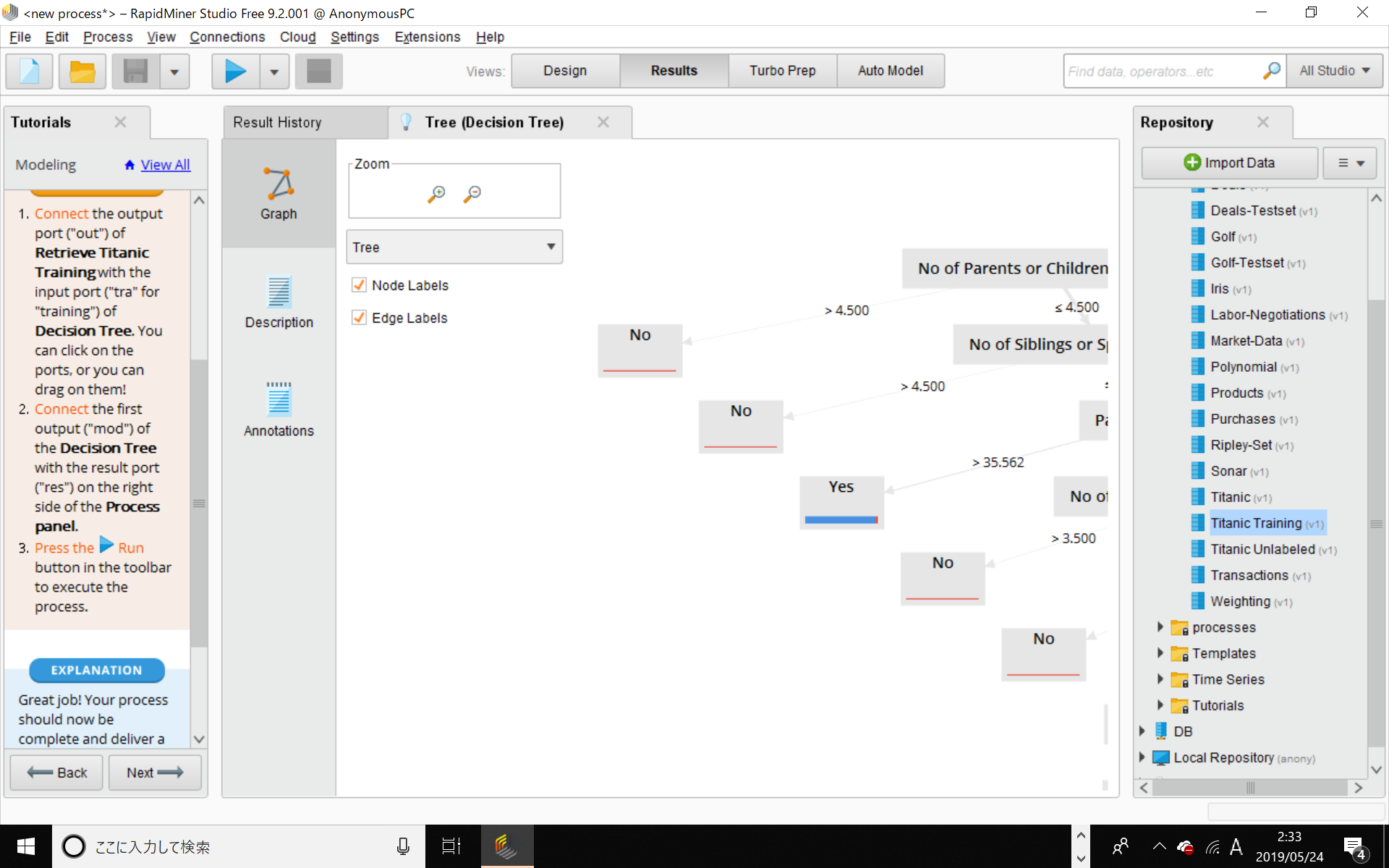


図3.2.1.2.2.2 Tutorial1-2 Activity2の実行画面

これでは何の結果が得られたかわかりにくいため、さらに次のようにした。  
1. プルダウンをTreeからTree (Tight)に変更した。  
2. 不要なパネルを閉じた。  
　 Tutorialsパネルを閉じる際、Leave tutorialというダイアログが出たが、   
 Continue with current processを選択した。  
3. Zoomで⊖をクリックすることで、グラフを縮小し、余白部分をドラッグして移   
 動させた。  
その結果、図3.2.1.2.2.3のようになった。

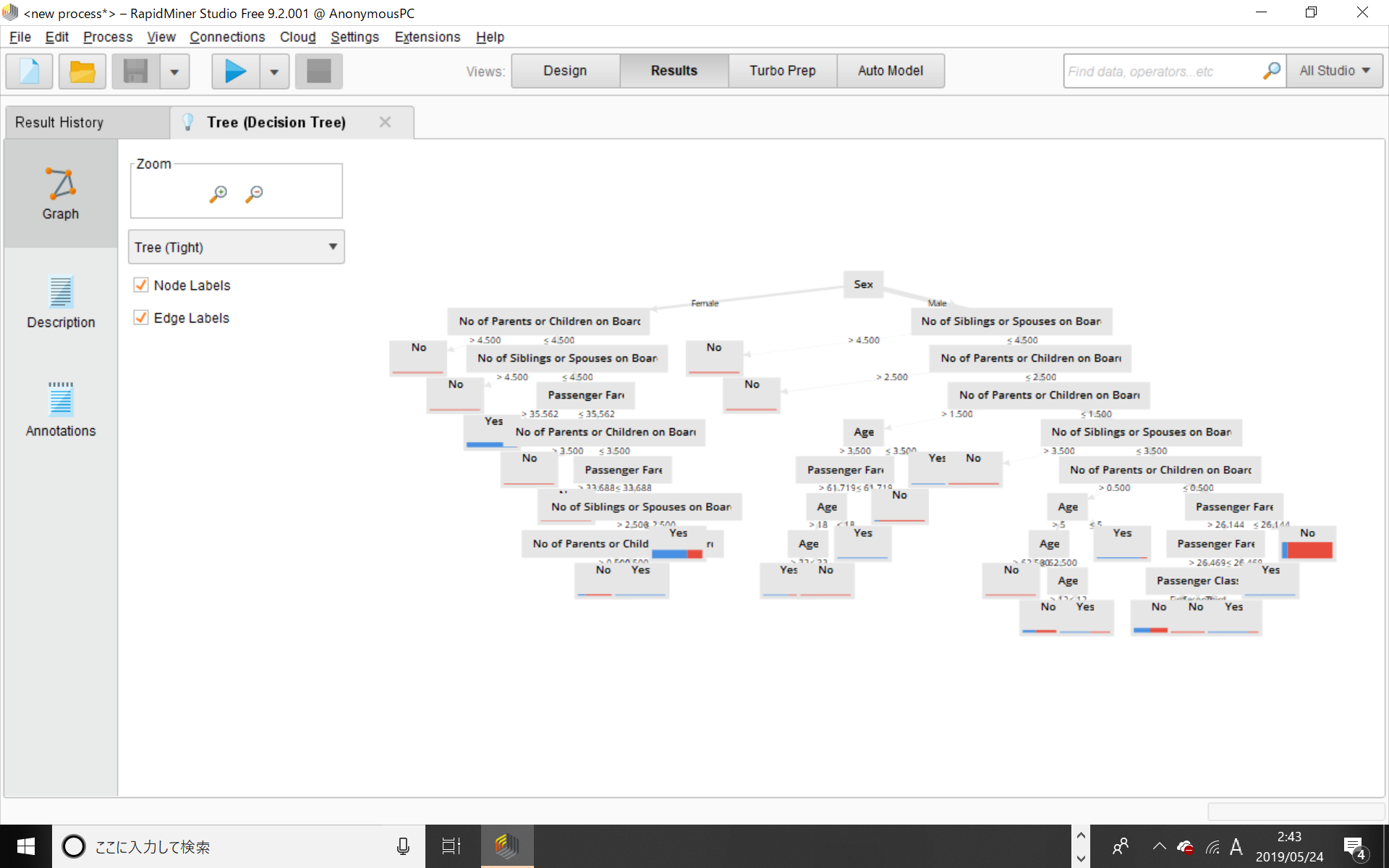


図3.2.1.2.2.3 Titanic Trainingデータの統計モデルを木構造で表した図

結論から言うと、図3.2.1.2.2.3で見られる木構造の図は、タイタニック号の事故から生還した者の持つ特徴を分析したグラフであり、生存者は女、犠牲者は男性という傾向が最も顕著であったということを示すものである。青と赤で構成された小さな帯グラフが無数にあるが、その中で2個、太く強調された帯グラフが存在することがわかる。それらは左から順に女／男性の生存者の割合を示す。このことは、木構造の一番初めの部分にSexと記され、左のリンクがFemale(女)、右のリンクがMale(男性)となっていることから確認できる。

我々は生存者を目的語としてデータ分析を施したわけであるが、目的語が生存者となることの根拠はどこにあるのだろうか。このことを確認するには、次のようにする。  
1. Designビューをクリック  
2. データ取得演算子を選択し、outポートにマウスポインタを置き、マウスオーバーの画面が表示されることを待つ  
3. F3キーを押す  
すると、名前のないウィンドウが表示され、その中に表3.2.1.2.2.1を得る。

表3.2.1.2.2.1 Titanic Trainingデータの属性及びその情報

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Role | Name | Type | Range | Missings | Comment |
|  | Age | real | =[0.167 – 80] | =0 |  |
|  | Passenger Class | polynominal | =[First, Second, Third] | =0 |  |
|  | Sex | binominal | =[Female, Male] | =0 |  |
|  | No of Siblings or Spouses  on Board | integer | =[0 – 8] | =0 |  |
|  | No of Parents or Children  on Board | integer | =[0 – 9] | =0 |  |
|  | Passenger Fare | numeric | =[0 – 512.329] | =0 |  |
| label | Survived | binominal | =[No, Yes] | =0 |  |

表3.2.1.2.2.1を和訳したものが、表3.2.1.2.2.2である。

表3.2.1.2.2.2 Titanic Trainingデータの属性及びその情報(和訳)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 役割 | 属性名 | 型 | 値の範囲 | 欠損数 | コメント |
|  | 年齢 | 浮動小数点 | =[0.167 – 80] | =0 |  |
|  | 乗客等級 (グリーン車・ 普通車のようなもの) | 3つ以上の 文字列 | =[特上,上,並] | =0 |  |
|  | 性別 | 2つの 文字列 | =[女,男] | =0 |  |
|  | 同乗していた きょうだいや 配偶者の人数 | 整数 | =[0 – 8] | =0 |  |
|  | 同乗していた 親か子の人数 | 整数 | =[0 – 9] | =0 |  |
|  | 運賃 | 数値 | =[0 – 512.329] | =0 |  |
| ﾗﾍﾞﾙ | 安否 | 2つの 文字列 | =[死亡,生存] | =0 |  |

※数値(numeric)は浮動小数点(real)と整数(integer)のどちらにもなりうる[12]

表3.2.1.2.2.1および表3.2.1.2.2.2から、安否情報(Survived)にラベル(label)という役割(Role)が与えられていることがわかる。決定木演算子は、このようにラベル付けされた属性を目的語として統計モデルを作り、決定木を表示する。

#### 用語等解説

##### decision tree[8]

決定木。意思決定に利用される2分木のリスト構造(で書かれたグラフ)で、葉以外の各ノードに条件判断がつく。

##### statistical modeling[10]

統計モデル。データを定量的に説明するための手段。以下、参考文献[10]から説明を引用する。

なぜ統計モデル（statistical model）という考えかたがデータ解析に必要とされるのかを考えてみましょう。統計モデルは、

観察によってデータ化された現象を説明するために作られる

確率分布が基本的な部品であり、これはデータに見られるばらつきを表現する手段である

データとモデルを対応づける手つづきが準備されていて、モデルがデータにどれぐらい良くあてはまっているかを定量的に評価できる

といった特徴をもつ数理モデルです。

ここで、数理モデルとは「数学の言葉で表したモデル」のことである。

##### modeling[11]

では、モデリングやモデルとはそもそもどのようなことを言うのか。

モデリングはモデル化することを言う。

モデルとは、「枝葉を切り落として抽象的にすることで、物事を単純化して捉えた模型」のことである。厳密さを犠牲にする代わりに迅速に概算を行うことができるため、フェルミ推定等で大いに役立つ。例えば「日本の国土面積」をフェルミ推定する場合を想定しよう。新幹線での移動時間などを用いて、「縦」と「横」を求めた積として求めることができるが、これは「日本列島の形を長方形にモデル化した」からなせる業である。

##### drag in[7]

「～をドラッグする」。  
「～へドラッグする」ではないので要注意。  
drag in A and drop it in Bで「ＡをＢまでドラッグする」

##### though[7]

ただし

##### coincidental[7]

調和する、同時に起こる、偶然一致する、偶然に

##### iceberg of[7]

の氷山の一角

#### 演算子の説明[9]

##### 決定木演算子(Decision Tree演算子)

###### 概要

　決定木演算子は分類・回帰に有用な、決定木モデルを生成する。決定木は木のような構造をした、ノードの集合である。値がどう階級づけられるかだったり、あるいは数値の予測値だったりを決定することができる。  
(統計学における「階級」は中学数学にて学習済みのはず)

それぞれのノードは1つの特定の属性に対する分割規則を表す。  
(分割規則(splitting role):どんな理由で分割するか(性別、年齢(とその閾値)など))  
分割規則は異なる階級に所属している値を分割することで分類分けを行う。分割することで、パラメータで指定された条件下における最適な方法でエラーを減らして回帰を行うためである。  
(回帰(regression): 統計学において、データをモデルに当てはめること)  
停止基準を満たすまで、ノートは延々と作られていく。数値の予測は葉の平均値で得られる一方、ラベル属性の階級値の予測値は生成中に葉に到達したものの大部分によって決まる。

この演算子は文字列と数値の両方を含むサンプルセット  
(表データ,ExampleSets)を処理することができる。

ラベル属性は、分類を行う場合は文字列でなくてはならなく、  
回帰を行う場合は数値でなくてはならない。

生成後、決定木モデルはデータ適用演算子(Apply Model演算子)を用いてサンプル(表データの行Example)に当てはめることができる。

各サンプルは、葉に達するまで分割規則に従って木の枝をたどる。

###### 入力

・訓練データ(tra)

決定木モデルの生成に使われる入力データ

###### 出力

・モデル(mod)

生成された決定木モデル

・サンプルセット(exa)

入力されたデータセット(表データ)をそのまま出力

・重み(wei)

属性と重み値を持つデータセット。各重みは(おそらく引数で)指定された属性の、特徴の重要性を表す。  
重みは、指定された属性の選択に関するノードに対して行う改善の合計によって決まる。(？)改善の回数は、引数criterionによって決まる。

###### 引数

・基準(criterion)

分割される属性を選ぶための基準を定める。  
分割値は各基準に対して最適化される。   
以下のどれかが選択される。

・情報利得(information\_gain)

全ての属性に対してエントロピーが計算され、最小であったものが分割に使用される。  
この方法では、多数の値を持つ属性ほど選択されやすくなる。

(訳とは別に補足:[14]

情報利得、つまり(分割によって)情報を得ることとは、エントロピーすなわち混沌さを失うことである。  
言い換えれば分割によって、何らかの「傾向」が強くなれば、情報利得が大きくなる。  
例を挙げるのであれば、『タイタニック号事故の生存者』の決定木を作ろうとしたとき、『年齢』で分割した時、分割されたどのノードにおける『生存者』の割合はさほど偏りがなく、これは「傾向」が強いとは言えない。  
一方『性別』で分割した時、『性別＝女』のノードでは生存者の割合が高く、『性別＝男性』のノードでは生存者の割合が低くなる。これが「傾向」の強い例である。これは「『性別』という属性が、『生存者』(または犠牲者)に対してエントロピーが小さい」ということに当たる。)

・利得比(gain\_ratio)

情報利得を変形したものである。情報利得の各属性に値域と均一性を持たせたもの。

・ジニ係数(gini\_index)

ジニ係数はラベル属性の不均衡さを測る指標である。  
枝分かれしていく毎に、平均ジニ係数は減少する。

・正確度(accuracy)  
決定木全体の正確度が最大になるよう分割される属性が選ばれる。  
(訳とは別に補足[15]:正確度と精度は別物である。  
正確度(accuracy)は「真値」にどれだけ近い値であるかを示す尺度であり、精度(precision)は測定値間のばらつきの尺度である。)

・最小二乗法(least\_square)

分割用の属性は、「真値(モデルとする関数の値)とノードの値」の差の2乗が最小になるように選ばれる。  
(訳とは別に補足[16]:正確度と精度は別物である。  
正確度(accuracy)は「真値」にどれだけ近い値であるかを示す尺度であり、精度(precision)は測定値間のばらつきの尺度である。)  
以下、参考サイト[16]から引用する。

最小二乗法の意味

最小二乗法とは、モデル関数をとするとき、

が最小となるように を求めることである。

・深さの上限(maximal depth)

決定木の深さは表データの大きさや特徴によって様々である。この引数は決定木の深さを制限するのに使われる。「-1」に設定すると、他の条件によって停止するまでどこまでも深い決定木が生成される。「1」にすると、「ノード1つ」が生成される。

・枝切りを行う(プルーニング)(apply pruning)

決定木を生成してから枝切りを行う。チェックを入れると信頼度(confidence)パラメータに従って、一部の枝が葉に置き換わる。

・信頼度(confidence)

枝切りの悲観的誤差計算に使用される信頼度の閾値を指定する。

・枝切りを予め行う(プリプルーニング)(apply prepruning)

決定木を深くしていくのを止める条件を「深さの上限」(maximal depth)以外にも用意するか。

チェックを入れると「最小利得」(minimal gain)、「葉の保有するサンプル数の最小値」(minimal leaf size)、「分割されるノードが保有するサンプル数の最小値」(minimal size for split)、「プリプルーニングの選択肢の数」(number of prepruning alternatives)が停止の条件として用意される。

・「最小利得」(minimal gain)

### Tutorials1-3 Accessing Data

#### 原文と和訳

Importing Data into RapidMiner.

RapidMinerへデータを読み込む

Getting your data into RapidMiner is often the first task you will need to complete your analysis.

分析を行うとき、最初の壁となるのは、データをRapidMinerに読み込むことでしょう。

In this tutorial, you will learn how to import files into the central storage of RapidMiner, called the Repository.

今回のチュートリアルでは、RapidMinerの中央ストレージ(レポジトリと言います)へファイルを読み込む方法についてみていきましょう。

We will continue to use data describing the Titanic accident, but this time it will be imported from an Excel file.

ここでもタイタニック号の事故に関するデータを使いますが、  
今回はエクセルファイルからの読み込みをやってみましょう。

explanation

説明

We will now go through the most important steps of building analytical processes starting with data import.

これから、プロセス構築を行う上で最も重要となるステップを、データの読み込みから体験していきます。

In later tutorials, we will cover data preparation and modeling.

(今後のチュートリアルで、データプレパレーションとモデリングについても扱っていきます。

We also will discuss some of the steps you saw so far in more detail.

また、これまでに触れた手順についても、より詳しく説明します)

Download your data.

データのダウンロード

Activity1

やること1

1.Download this Excel file to your computer.

1.こちらのエクセルファイルをダウンロードしてください。

2.To import the downloaded data into RapidMiner, click Import Data in the  
 Repository panel and follow the steps in the wizard.

ダウンロードしたら、そのデータをRapidMinerへ読み込みます。RepositoryパネルのImport Data(データの読み込み)をクリックし、以降はウィザードに従ってください。

3.When you complete the import, store the data as Titanic in your Local Repository.

読み込みができたら、そのデータをTitanicと名付け、Local Repositoryへ保存します。

explanation

説明

The Repository panel, in the upper left corner by default, is the place to store all your data, processes, and results.

(デフォルトでは左上端にある) Repositoryパネルにはすべてのデータ、処理、結果をしまっておくことができます。

You should always import data into the repository, especially when it comes from files like XLS or CSV.

特にxlsファイル、csvファイルと言ったファイルを読み込む場合は、必ずレポジトリに読み込むようにしてください。

This will simplify the design of analytical processes a lot since RapidMiner's repository stores describing meta data together with the data.

RapidMinerはデータと一緒にメタデータもレポジトリに保存しています。  
このことにより、分析プロセスの構築がとてもシンプルに行えるようになっています。

Add data to the process.

データをプロセスへ追加する

Activity2

やること2

1.Click the Design tab to return to the Process panel.

1.Designタブをクリックして、Processパネルへ戻ります。

2.Drag the imported Titanic data from the Repository panel into the Process panel.

2.レポジトリに読みこんだTitanicデータをProcessパネルへドラッグして  
ください。

explanation

説明

When you drag data from the repository into the process, it transforms into a data-loading operator (in this case, Retrieve Titanic).

データをレポジトリからプロセスへドラッグすることで、そのデータを読み込む演算子(今回の場合は「データ取得(Titanic)」)が出てきます。

Data is not actually loaded (or delivered at the round output ports of each operator) until you run the process.

プロセスを実行するまで、データは実際には読み込まれず、各演算子の出力ポートまでデータが流れることもありません。

Create a connection to view results.

接続して結果をみる

Activity3

やること3

1.Connect the output port of Retrieve Titanic with the result port ("res") on the right side of the Process panel.

1.データ取得(Titanic)の出力ポートを、Processパネル右側の結果ポート「res」に繋ぎます。

2.Make the connection either by dragging a line between the ports, or by first  
 clicking on one port and then on the other port.

2.ポート間をドラッグするか、両端を順にクリックすることで接続できます。

explanation

説明

Only data which is delivered to one of the result ports on the right can be seen after the execution of the process.

Processパネルの右端にある結果ポートまでたどり着いたデータだけが、  
プロセスの実行結果として表示されます。

If your process does not have at least one connection to a result port, you won't see any results when you execute it!

結果ポートに何も繋がれていなければ、何もみることができません。

Execute the process.

プロセスの実行

Activity4

やること4

1.Press ▶ Run (top left) to execute the process.

1.(左上の)▶ プロセスの実行(Run)ボタンをおしてプロセスを実行して  
 ください。

explanation

説明

Once run, you automatically switch to the Results view where your results are displayed.

すると自動的に、結果が表示される結果(Results)ビューへ移動します。

Remember? Those are operator outputs you connected with one of the result ports on the right side of the Process panel.

繰り返しますが、Processパネルの右端の結果ポートには、演算子の出力が繋がれる必要があるのです。

At any time, click the Design tab to return to the Process panel.

Processパネルに戻る時は、いつでも設計(Design)タブをクリックしてください。

Congratulations!

完了

You just have imported your first data set!

データの読み込みができました。

From now on, each tutorial includes some additional questions - to further improve your skills - take a look at the challenges below!

今後の各チュートリアルには、さらにスキルを磨くための追加課題があります。下記の「挑戦」に取り組んでください。

Challenge

挑戦

You can see the data in the results.

(1)結果のデータを見てください。

Can you find out how missing values are shown?

欠損値はどのように表されているか答えなさい。

The Statistics tab shows summaries of the data in the columns.

(2)Statistics(統計)タブにデータの概要が表示されます。

How many people did travel in First class?

ファーストクラスの乗客の人数を求めなさい。

And how many people did not survive the Titanic accident?

(3)タイタニック号の事故における犠牲者数を求めなさい。

Play around with some of the Charts if you like. Can you see some interesting patterns?

(4)グラフを好きなようにいじくりまわし、面白い関係性を見つけなさい。

#### Activity

#### 用語等解説

#### 演算子の説明[9]

###### 概要

###### 出力

###### 引数

### Tutorials1-4 Filtering and Sorting

#### 原文と和訳

Find out the highest fare paid by women.

女性客の運賃における最大値を調べる

In the last tutorial, we learned how to import data into the RapidMiner repository, which is the central storage for all your data, models, and processes.

前回のチュートリアルで、レポジトリにデータを読み込む演習をしました。  
レポジトリはデータ、モデル、プロセスなどがすべて格納される、中央ストレージです。

In this tutorial, we will apply a filter to the Titanic data to only look at female passengers.

今回は、女性客のデータを取り出すため、タイタニック号のデータに対して抽出を行います(フィルタをかけてみます)。

Then, we can easily sort the data to find the highest ticket fares paid by women.

次に女性客の運賃最大のものを見つけるべく、データの並び替えを行います。これは容易にできます。

You can then apply your new skills to answer the same question for men.

もちろん、男性客のデータに対しても同じことができます。

Do you expect a difference for the passenger fare?

運賃の男女差は果たして存在するのでしょうか。

Get data into the workflow.

データをプロセスに読み込む

Activity1

やること1

1.Drag the Titanic data from the Samples repository into the Process.

1.SamplesレポジトリからTitanicデータをドラッグし、プロセスに追加しましょう。

explanation

説明

In RapidMiner, rows are called examples and data tables are called example sets.

RapidMinerでは、データの表をサンプルセット(ExampleSet)といい、その行を特にサンプル(Example)と言います。

You'll find these terms throughout RapidMiner, so it is worth learning them right away.

この表現はこれからもずっと出てくるので、ここで覚えておくのをお勧めします。

explanation

説明

There are many ways to find the highest fare paid for women.

In this tutorial, you will remove men from the table or "filter examples out of the example set."

女性客の最高運賃を調べる方法はいろいろありかすが、今回は男性のデータを表から取り除く、つまり「サンプルセットから(男性の)サンプルをフィルタでブロック」していきます。

Set up filtering.

フィルタの構築

Only examples (rows) that meet the defined filter criteria - in this case women - stay in the example set. All other examples are removed.

決められたフィルタリング(抽出)基準に合致するサンプル(今回は女性)のみが表データに残り、ほかのサンプルは除外されます。

Activity2

やること2

1.Search for the Filter Examples operator using the search box at the top of the  
 Operator panel.

Operatorパネル上部にある検索(search)ボックスを使って、サンプル抽出演算子(Filter Examples 演算子)を検索してください。

Drag Filter Examples into the Process panel.

Processパネルへサンプル抽出演算子をドラッグします。

2.Connect the output port of Retrieve Titanic with the input port of Filter Examples.

2.データ取得(Titanic) の出力ポートを、サンプル抽出演算子の入力ポートへ接続してください。

3.Click Filter Examples to select it, then click Add Filters in the Parameters panel to  
 define a filter.

3.サンプル抽出演算子をクリックして選択し、抽出方法(フィルタ)を定義するため、引数(Parameters)パネルのフィルタの追加(Add Filters)をクリックします。

4.Select Sex on the left, equals in the middle, and type Female in the box on the right.

左枠は性別(Sex)、真ん中枠は等号(equals)、を選び、右枠には女(Female)を入力します。

Instead of typing, you can click on the magic wand and select Female from the list.

入力の代わりに、魔法の杖をクリックして出てくるリストから女(Female)を選択することもできます。

explanation

説明

Whenever you add an operator to your process, you should immediately connect it.

演算子を追加するときは、必ず接続するようにしてください。

Remember that data flows between operators, so an operator's connection can influence its parameters.

また、データは演算子の間を流れているので、演算子の引数は接続に依存しているということにも注意が必要です。

For example: How could the Filter operator "know" about the column Sex, if it is not connected to the data source?

例えば、データ源と繋がっていない時、サンプル抽出演算子は性別に対応する列を「知る」ことはできません。

Sort to show the highest fares first.

並び替えを行い、運賃の最大値を一番上に表示させる

1.Search for and then drag the Sort operator into the Process.

1.並び替え演算子(Sort 演算子)を探し、プロセスに追加してください。

2.Connect the output of Filter with the input of Sort.

2.サンプル抽出演算子の出力を、並び替え演算子の入力に接続します。

3.Click on Sort to select it. Make the following changes in the Parameters panel:

3.並び替え演算子を選択し、Parametersパネルで以下のように設定してください。

4.Set attribute name to Passenger Fare.

3-1.属性値を運賃(Passenger Fare)とします。

5.Change sorting direction to decreasing.

3-2.並び替えの方向(sorting direction)を降順(decreasing)に設定します。

6.Connect the Sort output port to the result port on the right of the Process panel.

3-3.並び替え演算子の出力を結果ポートに接続してください。

7.Run the process and inspect the result.

3-4.プロセスを実行して、結果を見ていきましょう。

explanation

説明

Most operators have settings which define how the operators are working.

大半の演算子では、動き方を選ぶオプションが付いています。

You can find those on the right side of the Parameters panel, after you have selected an operator with a click.

演算子を選択すると、画面右側のParametersパネルにそれらのオプションが表示されます。

Congratulations!

完了

Well done! You can now see the highest fare paid for a woman on the Titanic: it's the first value in the column Passenger Fare.

このようにして、運賃(Passenger fare)の列の一番上に、タイタニック号における女性客の運賃の最高金額を確認できます。

Challenge

挑戦

Can you change the process to see the highest fare paid by a man?

(1)男性客の運賃の最高金額を求めなさい。

Is it different than the highest fare paid by a woman?

また、男性客と女性客で、運賃の最高金額に差があるか述べなさい。

Welcome to RapidMiner.

ようこそRapidMinerへ

explanation

説明

Activity1

やること1(番号は本稿で勝手に振りました)

1.Find the Repository panel on the left of the screen.

1.画面左側にあるRepository\*パネルをみてください。

#### Activity

#### 用語等解説

#### 演算子の説明[9]

###### 概要

###### 出力

###### 引数

### Tutorials1-5 Merging and Grouping

#### 原文と和訳

Say hello to two new data sets...

2つのデータを同時に扱う

Let's take a break from the Titanic and learn about some other frequently used tasks in data preparation, specifically merging and grouping data together.

タイタニック号のことはとえあえず置いておいて、連結やグループ分けまど、データプレパレーションでよく使われるスキルについて学んでいきましょう。

We will deal with two data sets:

今回は2つのデータセットを扱っていきます。

one containing the products sold by an organization and one with the transactions (information about which customer purchased which product).

1つ目はとある組織が販売した商品についてのデータで、2つ目は取引(どの顧客がどの商品を購入したか)のデータです。

After combining these sets, we can answer questions about the most frequently purchased product or who is your most loyal customer.

2つのデータセットを比較することで、どの製品が最もよく売れるか、あるいは一番のお得意様は誰かということを調べることができます。

Let's get started...

ではやってみましょう。

Get data into the workflow.

プロセスにデータを読み込む

Activity1

やること1

1.Expand the Samples repository in the Repository panel.

Next, expand the data folder within the samples repository to retrieve the Products and Transactions data.

1.RepositoryパネルでSample→dataを開きます。

2.Drag the Products data and the Transactions data from the Samples - Data folder into the Process panel.

2.dataフォルダにある製品(Products)データと取引(Transactions)データをProcessパネルまでドラッグしてください。

explanation

説明

Remember, RapidMiner transforms the data into Retrieve Products and Retrieve Transactions operators when it enters the process, but doesn't load the data until you execute the process.

RapidMinerでは、データをプロセスに追加しようとすると、自動的にデータ取得演算子に変換されます。そして、プロセスが実行されるまで、データは読まれません。

Join the data.

データの連結

Activity2

やること2

1.Search for the Join operator in the search box at the top of the Operator panel.  
 Drag Join into the Process panel.

1.データ連結演算子(Join演算子)をOperatorパネルから探し、Processパネル  
 までドラッグしましょう。

2.Connect the output port of Retrieve Products to an input port of Join  
 (it doesn't matter which one).

データ取得(Products)の出力ポートを、データ連結演算子の入力ポートのどれか(どれでも構いません)に接続します。

3.Connect Retrieve Transactions to the other Join input port.

3.データ取得(Tramsactions) も、データ連結演算子の空いている入力ポートへ接続します。

4.Click on Join to select it.

4.データ連結演算子を選択してください。

In the Parameters panel, de-select use id attribute as key.  
The key attributes field appears.

Parametersパネルを開き、「id属性をキーとして使用する」  
(use id attribute as key)の選択を解除します。

するとキー属性(key attributes)フィールドが現れます。

5.Click Edit List. Select Product ID for the left and right key attributes. Then, click Apply.

5.リストの編集(Edit List)をクリックし、左右のキー属性に製品ID(Product ID)を選択してください。

explanation

説明

Remember to ALWAYS connect the operators before you start changing their parameters.

演算子は必ず引数を編集する前に接続するようにしてください。

You can only select the Product ID from the list after the connection was made since otherwise the operator would not know what data is available.

リストから「製品id」を選べるのは、演算子がきちんと接続された時だけです。そうでないと、演算子はどんなデータが利用可能であるかを知りえません。

explanation

説明

The result of the Join will be a table showing each transaction and its product details.

データ結合演算子の出力は、各取引(transaction)とその商品(product)の詳細を示す表になります。

The two ID columns you defined as key attributes for the join define the mapping between the rows of the two original tables.

(結合の)キー属性として定義した2つのIDの列は、2つの元のテーブルの行同士のマッピングを定義します。

Group the data to count product purchases.

データをグループ化して製品購入数を数える

Activity3

やること3

1.Drag the Aggregate operator into the process.

1.プロセスにデータ集計演算子(Aggregate演算子)をドラッグしてください。

Connect it to the output of Join.

それをデータ連結演算子の出力に接続します。

2.Click Aggregate to select.

2.データ集計演算子を選択します。

Make the following changes in the Parameters panel:

Parametersパネルを開き、次のように設定してください。

3.Click on group by attributes.

3.「属性でグループ化する」(group by attributes)をクリックします。

Then, select the Product ID by moving it to the right. Click Apply.

次の画面で、製品ID(Product ID)を右に移すことで選択してください。そして適用(Apply)をクリックします。

4.Click on aggregation attributes.

4.集計属性(aggregation attributes)をクリックします。

5.Select Customer ID in the left box and set function to count in the right box.

5.左のボックスで顧客ID(Customer ID)を選び、右のボックスで総数(count)を調べる集計関数(function)を選びます。

6.Stay in this dialog and add another entry Product Name with function set to mode. Click Apply.

追加(Add entry)をクリックして、商品名(Product Name)を選び、最頻値(mode)を調べる集計関数とします。適用(Apply)を押します。

explanation

説明

Aggregate performs a "group-by-function" that you may know from databases.

データ集計演算子は、データベースをやったことのある人なら知っているような、GROUP BY FUNCTIONを実行します。

Next to joins and filters, the aggregate function is one of the most important operators for data blending.

集計関数は統合や抽出の次に、データブレンディングで最も重要なものの一つです。

In this case, it groups the data by product, counts the number of purchases for each product, and uses the product name for describing this product.

この場合、製品(product)ごとにデータをグループ化し、それぞれ売り上げ(purchases)数を数え(count)、その結果を表現するときに商品名(product name)を用いています。

The result is a table of all products with the following attributes:

product ID, product name, and the number of customers that have purchased the product.

結果は全商品(product)の一覧表として得られ、属性は商品ID(product ID)、商品名(product name)、その商品を購入した顧客(customer)数となります。

Execute the process.

プロセスの実行

Activity4

やること4

1.Connect Aggregate to the result port on the right.

1.データ集計演算子を結果ポートに繋ぎます。

2.Press ▶ Run to execute the process.

実行▶(Run)ボタンを押してプロセスを実行しましょう。

explanation

説明

In the Results view, click on column headers to sort the column data to ascending or descending order.

結果ビューで、列見出しをクリックし、列見出しを昇順または降順に並び替えます。

Congratulations!

完了

Great, you just made your first steps for data blending with RapidMiner!

以上で、RapidMinerでのデータブレンドの第一歩を踏み出せました。

Before you move on to the next tutorial, try to answer the questions below:

次のチュートリアルへ進む前に、次の追加課題に取り組んでください。

Challenge

挑戦

Which product has been most often sold? And which product was sold only 5 times?

(1)ベストセラーを求めなさい。また、5回しか売れていない商品を求めなさい。

Can you find out in the Statistics tab what the average number of transactions was?

(2)統計(Statistics)タブから、取引(transaction)の平均回数を求めなさい。

Can you also see the visual distribution of values in this tab?

また、統計(Statistics)タブからデータの分布を視覚的に確認する方法を述べなさい。

The count function counts the number of transactions for each product,

but each product can also be purchased multiple times in each transaction.

(3)count関数は各取引における取引の回数を数えますが、1回の取引で複数個同じ商品が売れる場合もあります。

Can you change the parameters of Aggregate so that the total sum for each product is calculated?

データ集計演算子の引数をいじり、各製品の総数を計算させなさい。

Which products have been sold more than 65 times?

また、65回以上売れた商品を求めなさい。

#### Activity

#### 用語等解説

#### 演算子の説明[9]

###### 概要

###### 出力

###### 引数

### Tutorials1-6 Creating and Removing Columns

#### 原文と和訳

Working with Attributes.

属性について

You are almost ready to build your first predictive model in RapidMiner!

ここまで取り組んできた皆さんは、予測モデルを構築できるレベルまであともう少しのところにいます。

But first, we need to deal with two very important operations for transforming your data sets into a format more suitable for learning.

予測モデルを構築する前に、データを機械学習に最適な形にするためになくてはならない演算子を2つ紹介します。

The beginning of this process is the same as the previous one;

プロセスの最初の部分は前回と同じです。

this is a great opportunity for you to implement what you have learned so far.

ちょうどいい機会なので、今までやってきたことをここで復習してみてください。

We will then create a new data column and remove some unused/unnecessary columns from the data.

それからデータ列(属性)を追加・削除していきます。

Add product details to transactions.

取引データに製品詳細を加える

Activity1

やること1

1.Drag the Transactions and Products datasets into the Process.

1.取引(Transaction)表データと製品(Products)表データをプロセスに読み込みます。

2.Add a Join operator.

2.データ連結(Join)演算子を追加します。

3.Connect all operators.

3.演算子をそれぞれ接続します。

4.Specify the columns to use in the Parameters for Join.

4.データ連結演算子の引数に使う属性を決めましょう

That is, de-select use id attributes as key, click Edit List, and use Product ID for the left and right key attributes.

「id属性をキーとして利用する(use id attributes as key)」の選択を解除し、Edit Listをクリックし、製品ID(Product ID)を左と右のキー属性として使います。

explanation

説明

The resulting example set contains all transactions with product details for each transaction.

出力される表データは、取引ごとの属性、製品詳細の情報が含まれます。

Define a new attribute.

新しい属性の定義

explanation

説明

The term attribute is RapidMiner lingo for "column."

属性(attribute)というのは(表データの)列を意味するRapidMiner 用語です。

In machine learning, each row of a data set is an example for a specific situation and the attributes (columns) are the properties that describe the situation.

機械学習でいうと、表データの「行」はある場合の「例(サンプル)」(example)であり、「列」は各場合の「性質」(property)であり、場合を説明するためのものです。

Activity2

やること2

1.Add the Generate Attributes operator.

1.属性追加演算子(Generate Attributes 演算子)を追加してください。

2.Connect the operator with Join.

2.データ連結演算子と接続します。

3.Click Edit List in the Parameters for Generate Attributes to define the new  
 attribute (column). A dialog will pop up.

3.属性追加演算子のParametersパネルでEdit Listをクリックします。すると、新しく属性(列)を追加するためのダイアログが開きます。

4.In the left column of the dialog, enter Total for the attribute name.

ダイアログの欄(左)にTotalと入力します。これが新しい属性の名前になります。

(訳注:Totalは合計の意味)

5.In the right column, type Amount\*Price for the function expression.

5.右の欄には属性の数式として「Amount\*Price」(数量×値段)を入力します。

explanation

説明

You can also click on the calculator to create the expression using the Expression Editor.

電卓マークの数式エディタ(Expression Editor)を使うこともできます。

You should try this now - this is much easier than typing complex expressions into the text field!

複雑な数式の場合は、入力欄に直接タイピングするよりも数式エディタを用いた方がはるかに簡単なので、ぜひ使ってください。

#### Activity

#### 用語等解説

#### 演算子の説明[9]

###### 概要

###### 出力

###### 引数

# 第4章 英文和訳のコツ

英文和訳のフローを、プログラム言語のように述べると次のようになる。

|  |
| --- |
| main()  {  sentence=getSentence();//文を読む  /\*  例:sentence= “Let’s build a decision tree, which is a popular statistical modeling technique that finds hidden patterns in the data.”  \*/  sentenceIdiom=convertIdiom(sentence);//イディオムを見つけ、そこだけ和訳  /\*  “a decision treeをbuildしましょう, which is a popular statistical modeling technique  that finds hidden patterns in the data.”  \*/  while(sentenseOld!=sentense)  {  sentenseOld=sentense;  sentense=parse(sentence);//文法を解析して  print(buildJapanese(sentence) + “\n---\n”);//日本語の語順にして、メモ書き。  }  /\*  ☆  \*/  print(chokuyaku=translate(buildJapanese(sentense)) + “\n---\n”);//各単語を和訳  print(iyaku=naturalize(chokuyaku));//自然にする  /\*  ☆2  \*/  }  //以下、関数定義  String function parse(sentence)  {  //引数sentenceは、文字列の場合と、配列の場合の2通りが考えられる。  if(isString(sentence))//文字列だったら  arr=parseString(sentence);  else//配列だったら  {  arr=sentence;  foreach arr as elem  elem=parseString(elem);  }  return arr;//配列を返却  }  String function parseString(sentence)  {  //こちらでは、sentenceは必ず文字列  arr=array(V=0,S=0,O=0,O2=0,C=0); //まずVを見つける  V=getV(sentence);  tmp=rtrim(sentence,V);//Vより右を無視  S=getV(tmp);  tmp=ltrim(sentence,V); //Vより左を無視  O=getO(tmp,V);  C=getC(tmp,V); //Vの隣がOかCかは、Vによって決まる  if(O!=0)  O2=getO2(tmp);  while((M=getM(tmp))!=-1)  array\_push(arr,getM(tmp));  }  String function getV(sentence)  {  ...(略)//動詞となる単語のデータベースと照合し、探す  }  ...(略) |

☆の部分について解説する。

parse関数は、入力された文字列あるいは配列を、一つ次元が上な配列に変換することで、

一段階ずつ構文解析を行っていく。その為、返り値は配列となるが、その詳細は次の通り。

array(V,S,O,O2,C,M1,M2,M3,...):

ループ1回目

whileループ1回目におけるparse関数の返り値は次の通り。

array(“build”, 0, “a decision tree”, 0, 0, “which is a popular statistical modeling technique that finds hidden patterns in the data.”, -1)  
最後の「-1」は、これ以上Mが続かないことを意味する

この配列(array)を人間用に書くと

V= “build”

S=無

O=“a decision tree”

O2=無

C=無

M1= “which is a popular statistical modeling technique that finds hidden patterns in the data.”

となる。

ループ2回目

whileループ2回目におけるparse関数の返り値は次の通り。

array(“build”, 0, “a decision tree”, 0, 0, array(“is”, “which”, 0, 0, “a technique”, “popular statistical modeling”, “that finds hidden patterns”, “in the data.”, -1), -1);

この配列を人間用に書くと

V= “build”

S=無

O=“a decision tree”

O2=無

C=無

M1=

V’ = “is”

S’ = “which”

O’ =無

O2’ =無  
C’ = “a technique”

M1’= “popular statistical modeling”

M2’= “that finds hidden patterns”

M3’= “in the data”

ループ3回目

whileループ3回目におけるparse関数の返り値は次の通り。

array(“build”, 0, “a decision tree”, 0, 0, array(“is”, “which”, 0, 0, “a technique”, “popular statistical modeling”, array(“finds”, “that” , “hidden patterns” , -1), “in the data.”, -1), -1);

V= “build”

S=無

O=“a decision tree”

O2=無

C=無

M1=

V’ = “is”

S’ = “which”

O’ =無

O2’ =無  
C’ = “a technique”

M1’= “popular statistical modeling”

M2’=

V’’ = “finds”

S’’ = “that”

O’’ = “hidden patterns”  
O2’’ =無

C’’ =無

M3’= “in the data”

したがって、whileループを抜けた直後、printによる「メモ書き」は次のようになる。

a decision treeをbuildしましょう, which is a popular statistical modeling technique that finds hidden patterns in the data.

---

a decision treeをbuildしましょう, which は popular statistical modeling なa technique ですthat finds hidden patterns、in the data.

---

a decision treeをbuildしましょう, which は popular statistical modeling なa technique をisする、that はhidden patterns をfinds する、in the data.

---

但し、「a decision treeをbuildしましょう」はsentenceIdiomから持ってくる。

☆2

printによるメモ書きは、結局全体で次のようなものになる。

a decision treeをbuildしましょう, which is a popular statistical modeling technique that finds hidden patterns in the data.

---

a decision treeをbuildしましょう, which は popular statistical modeling なa technique ですthat finds hidden patterns、in the data.

---

a decision treeをbuildしましょう, which は popular statistical modeling なa technique をisする、that はhidden patterns をfinds する、in the data.

---

決定木を構築しましょう、それは人気のある統計モデリングな方法である、それは隠れた法則を発見する、データの中で

---

決定木を構築しましょう。決定木はデータに潜む隠れた法則を見つけるためによく使われる統計モデリング手法です。

再勉強のフローをtry-catchで追記すると、次の通り。

# 参考、注釈、翻訳元原文

[1]  
<https://www.sas.com/ja_jp/insights/analytics/data-mining.html>

[2]  
https://www.rapidminer.jp/  
[3]Enter your work email とあるためフリーメールなどのアドレスではダウンロードできない可能性があると判断した。

[4] <https://docs.rapidminer.com/latest/cloud/getting-started/running-process.html>  
パネルの呼称:Tutorials1-1

[5] <https://data.wingarc.com/what-is-datascience-6119>

[6] <https://dictionary.goo.ne.jp/jn/277810/meaning/m0u/>

[7] <https://ejje.weblio.jp/content/retrieve>

[8] <https://ejje.weblio.jp/cat/computer/cmpyg>

[9]<https://docs.rapidminer.com/latest/studio/operators/rapidminer-studio-operator-reference.pdf> (翻訳元)

[10] やり直し『データ解析のための統計モデリング入門』

[11](フェルミ推定の本など)

[12]<https://community.rapidminer.com/discussion/21242/solved-data-type-real-vs-numeric>

[13]<https://community.rapidminer.com/discussion/55608/please-tell-me-what-exampleset-is/p1?new=1>  
ちなみに質問者(ユーザ@\_\_\_)は本稿執筆者です。

[14] <https://dev.classmethod.jp/machine-learning/2017ad_20171211_dt-2/#sec4>

[15]<https://www.hitachi-hightech.com/jp/products/device/semiconductor/accuracy-precision.html>

[16](引用) https://sci-pursuit.com/math/statistics/least-square-method.html

# パーツ

### Tutorials1-1 Operators and Processes

#### 原文と和訳

Welcome to RapidMiner.

ようこそRapidMinerへ

explanation

説明

Activity1

やること1(番号は本稿で勝手に振りました)

1.Find the Repository panel on the left of the screen.

1.画面左側にあるRepository\*パネルをみてください。

Challenge

挑戦

#### Activity

Activity1を実行した結果、図3.2.1.1.2.1のようになった。

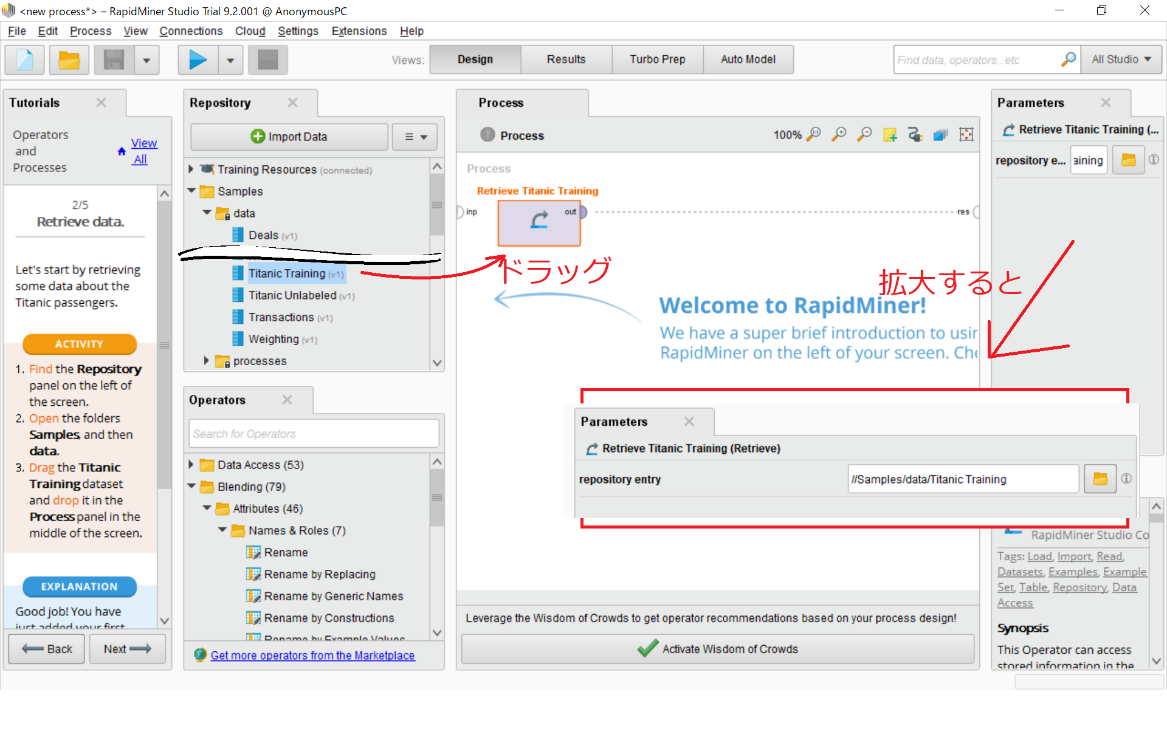


図3.2.1.1.2.1 Tutorial1-1 Activity1 の実行画面

RepositoryパネルからTitanic Training というデータをProcessパネルまでドラッグすると、図3.2.1.1.2.1の通り、データ取得演算子が生成され、その引数はTitanic Training となる。演算子の引数は、その演算子を選択の上、Parametersパネルから確認できる。

Statisticsタブに切り替えると、パネルは図3.2.1.1.2.4のようになる。

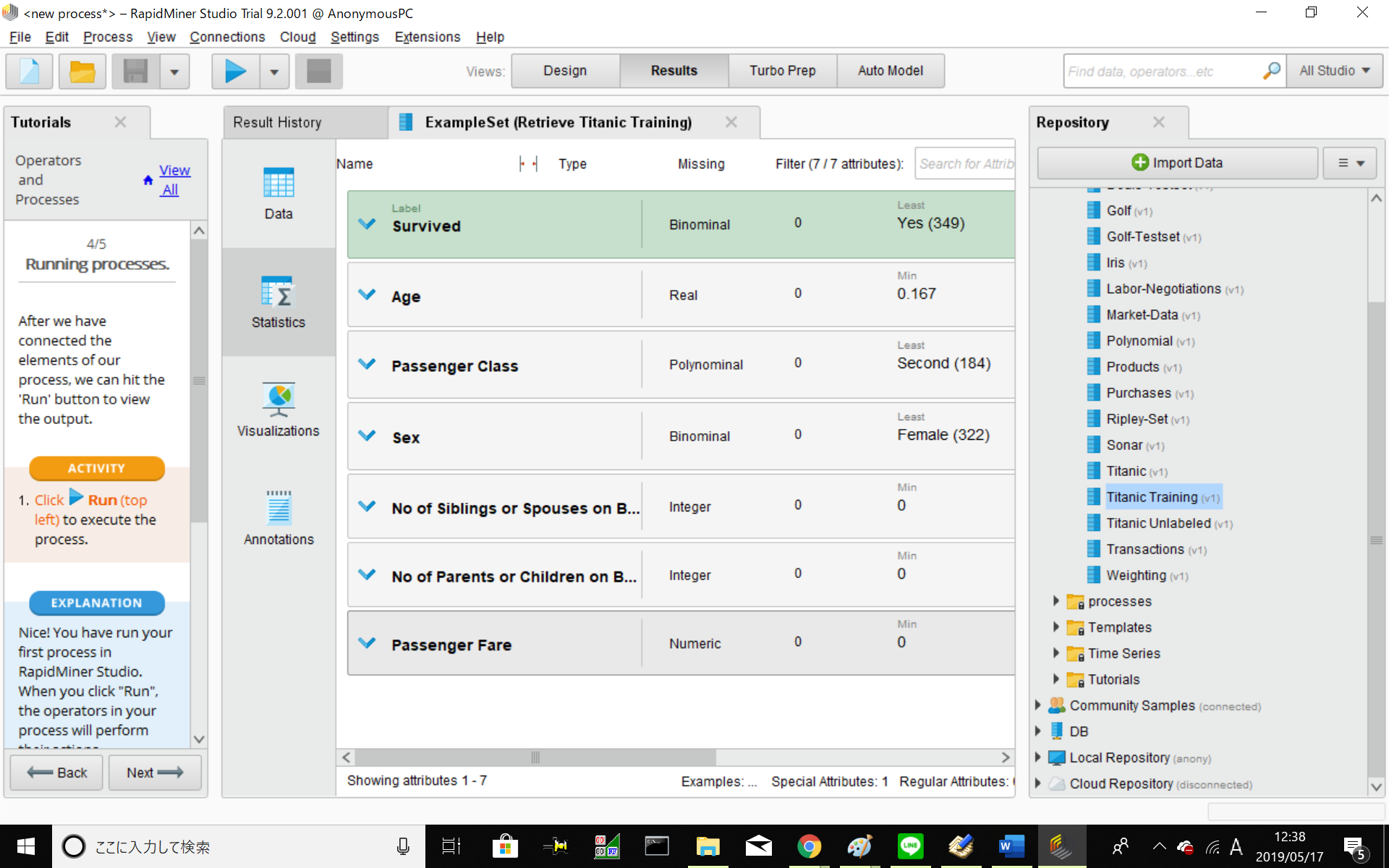
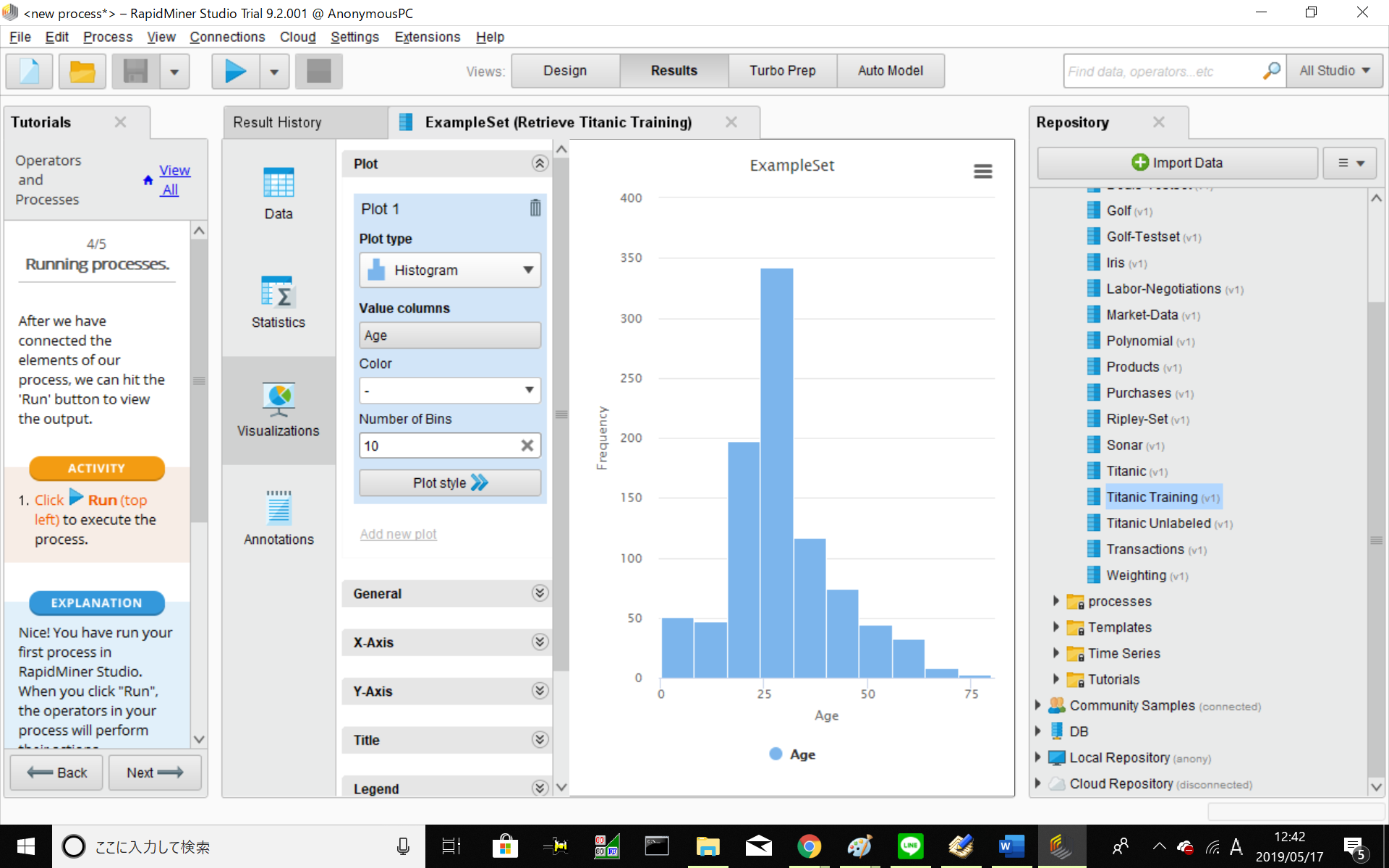
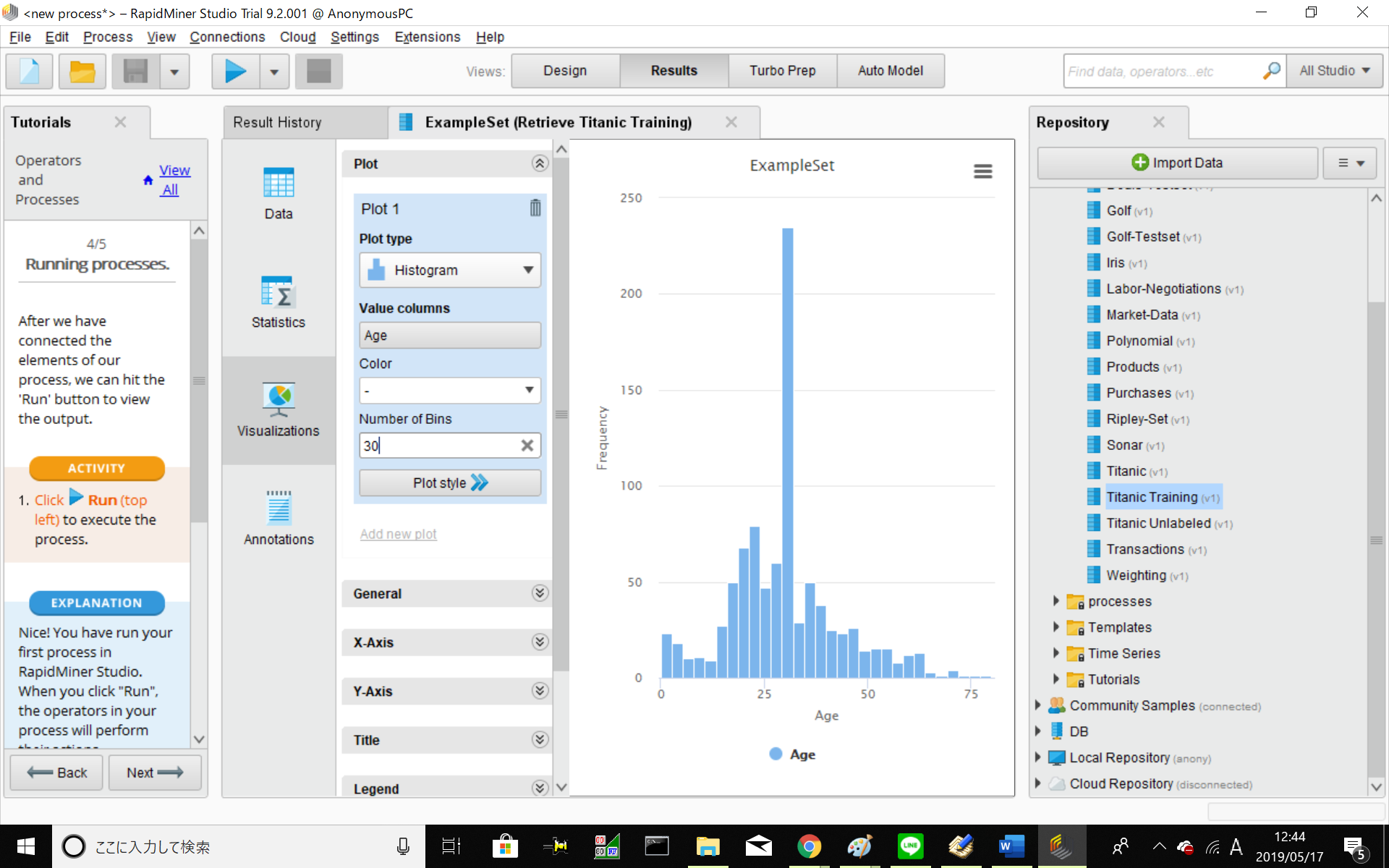


図3.2.1.1.2.4 Statisticsタブの内容

　図3.2.1.1.2.4の画面でAgeをクリックし、open visualizationをクリックすると、図3.2.1.1.2.5のような棒グラフを得る。  
 

(a) (b)

図3.2.1.1.2.5 年齢(Age)の出現数(Frequency)を示す棒グラフ

(a)では、binの値を10に、(b)では30に設定したものである。図と見比べることで、binは棒グラフの棒の本数を示すものと分かる。binを大きくすると、(b)のようにより細かくデータをみることができる。

#### 用語等解説

##### データサイエンス[5][6]

様々なデータの共通点を見つけ、そこから何らかの結論を導くために用いられる学問のこと。デジタル大辞泉[6]による説明を引用すると、次の通り。

データの分析についての学問分野。統計学、数学、計算機科学などと関連し、主に大量のデータから、何らかの意味のある情報、法則、関連性などを導き出すこと、またはその処理の手法に関する研究を行う。これらの研究者および技術者はデータサイエンティストとよばれる。

RapidMinerでは、自動モデリングの機能があり、またデータ可視化の方法も豊富である。さらに本来複雑な処理を、演算子を組み立ててプロセスを構築するだけで実行可能である。つまり数学、統計学、計算機科学の難しさを、ソフトウェアによる自動化が解消しているということもできるだろう。

#### 演算子の説明[9]

##### データ取得演算子(Retrieve演算子)

###### 概要

　データ取得演算子はレポジトリに保存された情報にアクセスし、それをプロセスへ読み込むという働きをする。

　アクセスされる情報(RapidMiner Object)はExampleSet、Collection、Modelである。

この演算子では、メタデータを一緒に読み込むことができるのが特徴である。メタデータは、RapidMiner Objectに関する追加情報で、ExampleSetの場合、属性の名前、型、範囲、欠落地の数などである。メタデータがあることで、他の演算子のパラメータをプルダウンメニューから選べるようになり、簡単に設定できるようになる。

###### 出力

引数「repository\_entry」で指定されたパスを持つRapidMiner Object

###### 引数

repository\_entry

読み込みたいRapidMiner Objectのパス

(パスの文法は、参考資料[9]に詳しい)