

YUIMA チュートリアル オープニング (平成30年度)

吉田朋広

1 YUIMA

YUIMA (ユイマ) とは

- 確率過程に対する統計解析およびシミュレーションのためのパッケージ
- R 上に実装されている
- 確率微分方程式や点過程をオブジェクト化する
- 最先端の確率統計学の理論に基づく結果を容易に利用できる

2 一つの例

YUIMA を使った統計解析の例を一つあげる。コード等の詳細はチュートリアルで説明するので、今ほどのようなことができるか眺めるだけで良い。

ここでのタスクは

1. YUIMA をインストール
2. 株価データをダウンロード
3. 確率モデルをデータに当てはめる
4. 当てはめたモデルを用いて将来の値動きをシミュレーションで予測する

である。

3 YUIMA の準備

YUIMA パッケージをインストールする。

```
> # install.packages("yuima") "#はコメントアウト。外すとそこが実行される。
```

YUIMA パッケージを読み込む。

```
> require(yuima)
```

4 データオブジェクトの作成

株価データを取得するために `quantmod`(Quantitative Financial Modelling Framework) パッケージをインストールし、読み込む。

```
> # install.packages("quantmod")
> require(quantmod)
```

IBM のデータをダウンロード。

```
> getSymbols("IBM", from = "2016-01-01", to = "2017-12-31")
```

```
[1] "IBM"
```

```
> # str(IBM) # Yahoo Financeからダウンロードされたデータが
> #           xts (time series) オブジェクトIBMに変換されている
```

データの始めと終わりを見してみる。

```
> head(IBM)
```

	IBM.Open	IBM.High	IBM.Low	IBM.Close	IBM.Volume	IBM.Adjusted
2016-01-04	135.60	135.97	134.24	135.95	5229400	120.9894
2016-01-05	136.76	136.89	134.85	135.85	3924800	120.9004
2016-01-06	134.38	135.58	133.62	135.17	4310900	120.2952
2016-01-07	133.70	135.02	132.43	132.86	7025800	118.2394
2016-01-08	133.18	133.82	131.32	131.63	4762700	117.1448
2016-01-11	131.81	133.82	131.76	133.23	4967000	118.5687

```
> tail(IBM)
```

	IBM.Open	IBM.High	IBM.Low	IBM.Close	IBM.Volume	IBM.Adjusted
2017-12-21	153.17	153.46	151.49	151.50	4153900	146.7899
2017-12-22	151.82	153.00	151.50	152.50	2990600	147.7588
2017-12-26	152.51	153.86	152.50	152.83	2479000	148.0786
2017-12-27	152.95	153.18	152.61	153.13	2149300	148.3693
2017-12-28	153.20	154.12	153.20	154.04	2687600	149.2509
2017-12-29	154.17	154.72	153.42	153.42	3327100	148.6502

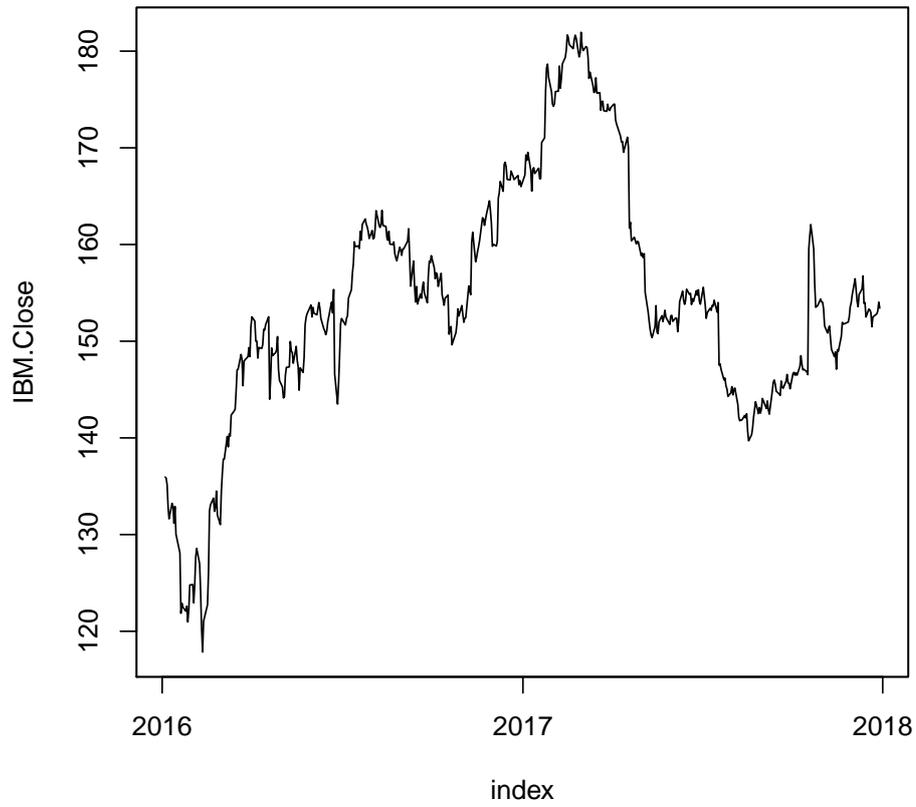
`setDate`, `setYuima` でデータオブジェクト、YUIMA オブジェクトをつくる。

```
> x <- setYuima(data=setData(IBM$IBM.Close))
> # str(x@data) # xのデータスロットにIBM.Closeデータが確認できる
```

プロットしてみる。

```
> plot(x, main="data with the original time stamps")
```

data with the original time stamps



5 モデルのフィット

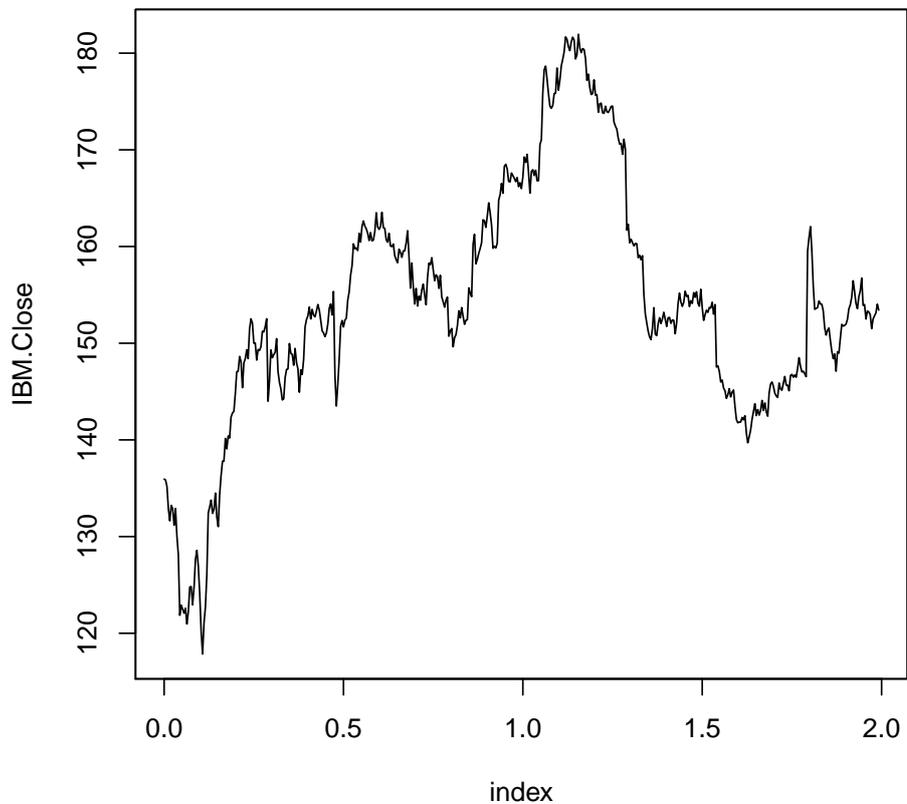
時間の単位として1年を1単位時間とする。 $\Delta = 1/252 = 0.00397$ 、252は標準的な1年のワーキングデーの数。

```
> y <- setYuima(data=setData(IBM$IBM.Close, delta=1/252))  
> # str(y@data) # 1ステップの時間が0.00397となった。
```

プロットする。

```
> plot(y, main="time stamps of data rescaled")
```

time stamps of data rescaled



Black-Scholes モデルのモデルオブジェクトを作る。

$$dX_t = \mu X_t dt + \sigma X_t dW_t$$

```
> IBM.mod <- setModel(  
+   drift=c("mu*x"),diffusion=matrix(c("sigma*x"),1,1))  
> # str(IBM.mod)
```

モデルとデータを束ねて YUIMA オブジェクトを作る。

```
> IBM.mod.data <- setYuima(model=IBM.mod, data=y@data)  
> # str(IBM.mod.data@data)
```

モデルのデータへのフィット

```
> IBM.fit <- qmle(IBM.mod.data,  
+               lower=list(mu=0.01, sigma=0.01),  
+               start=list(mu=1, sigma=1),  
+               upper=list(mu=10, sigma=10),  
+               method="L-BFGS-B")  
> # str(IBM.fit)  
> IBM.fit@coef # パラメータの推定値
```

```
sigma      mu
0.18820717 0.07668582
```

6 シミュレーションによるIBMのパスの予測

先ほどダウンロードしたデータは2016-01-04から2017-12-29であった。2018年はじめから約3ヶ月先までの株価のパスをシミュレーションで予測する。

後の比較に用いるデータをダウンロードする。

```
> getSymbols("IBM", from = "2018-01-02", to = "2018-04-05")
```

```
[1] "IBM"
```

```
> IBM.first <- IBM$IBM.Close["2018-01-02"]
> # 年の初めは2018-01-02である。
```

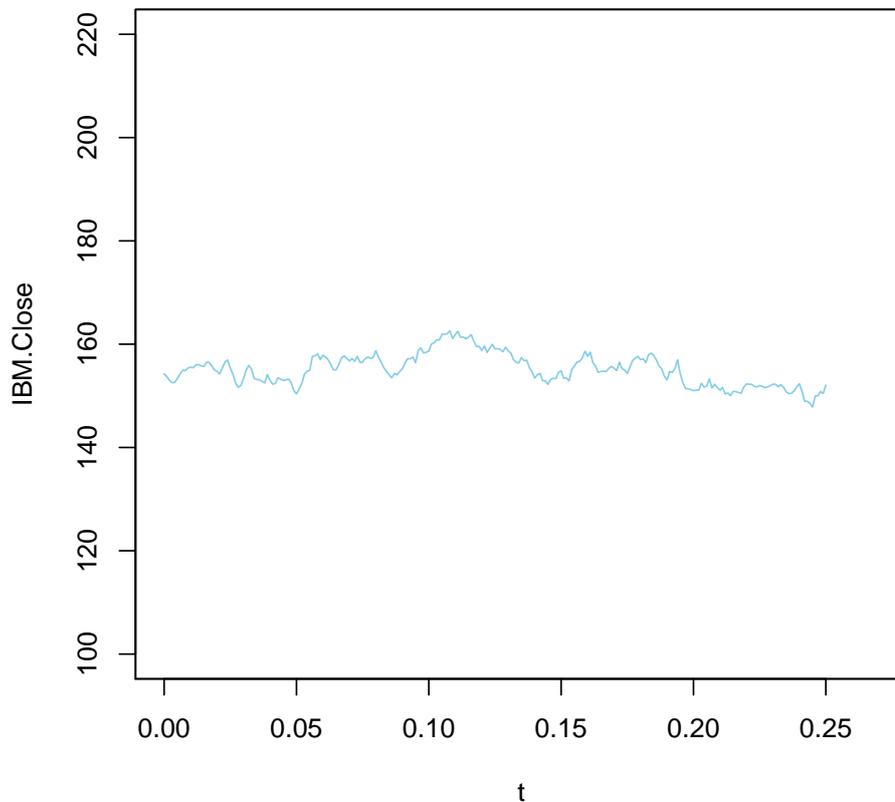
0.25年先までのモデルを設定する。

```
> IBM.mod.pred <- setYuima(model=IBM.mod,
+                           sampling = setSampling(Terminal=0.25, n=250))
> # str(IBM.mod.pred)
```

推定したモデルを使ってパスを生成してプロットする。

```
> set.seed(127)
> IBM.pred <- simulate(IBM.mod.pred,
+                      true.parameter = list(sigma=IBM.fit@coef[1],
+                                             mu=IBM.fit@coef[2]),
+                      xinit = IBM.first)

> plot(IBM.pred, ylim=c(100,220), xlim=c(0,0.27), col="skyblue",
+       ylab=c("IBM.Close"), xlab=c("t"))
```

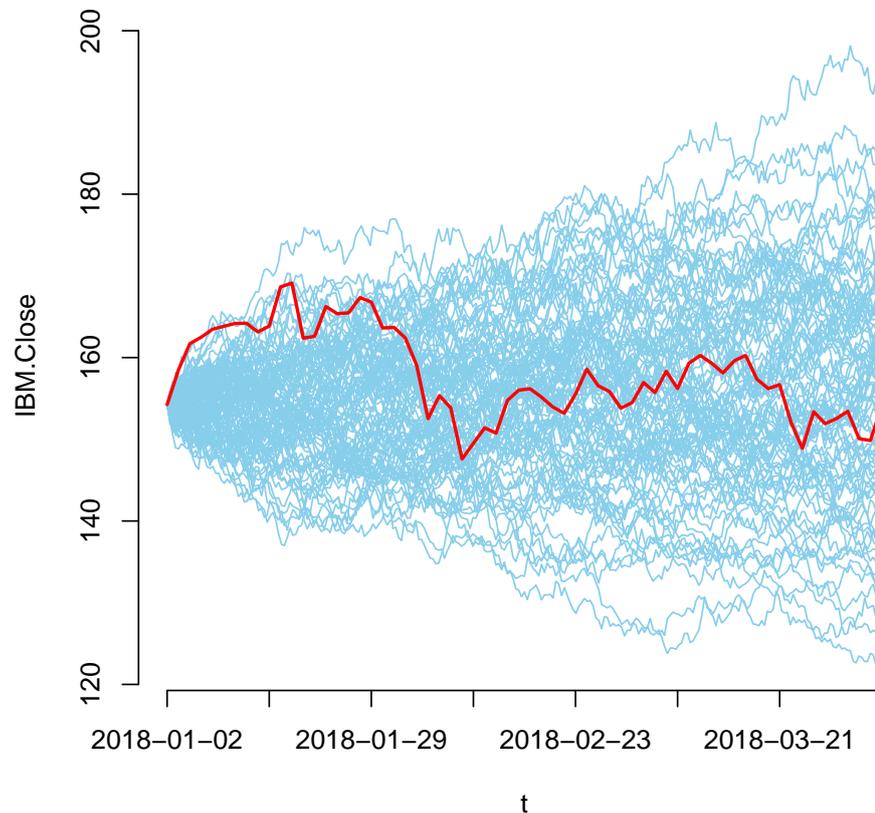


1本のパスでは傾向がわからないので、100本のパスをモデルから生成して重ねてプロットする。最後に実際のパスを重ねてみる。

```
> ## シミュレーターの作成
> mypred <- function(param){
+   result <- simulate(IBM.mod.pred,
+                     true.parameter = list(sigma=param[1], mu=param[2]),
+                     xinit = IBM.first)
+   return(get.zoo.data(result)[[1]])
+ }
> ## 100本のパスをシミュレートして結果を保存
> path <- replicate(100, mypred(IBM.fit@coef))
> ## シミュレートしたパスをプロット
> time.idx <- IBM.mod.pred@sampling@grid[[1]] # タイムインデックス
> matplot(time.idx, path, type = "l", lty = 1, col="skyblue",
+         xlab = "t", ylab = "IBM.Close", axes = FALSE,
+         main="IBM 2018.01.02-2018.04.04")
> ## 実際のパスを上書き
> IBM2018.y <- setYuima(data=setData(IBM$IBM.Close, delta=1/252))
> IBM2018.y <- get.zoo.data(IBM2018.y)[[1]]
> lines(time(IBM2018.y), IBM2018.y, col="red", lwd=2)
```

```
> ## 軸の設定
> idx <- seq(1, length(time(IBM)), by = 9)
> axis(1, at = time(IBM2018.y)[idx], labels = time(IBM)[idx])
> axis(2)
```

IBM 2018.01.02–2018.04.04



一般的な注意

- ここでの予測は使った確率論的モデルが現実の株価変動を完全に表現できているという仮定のもとに行われたものである。
- モデルが正しくても、推定と予測における統計的誤差（ランダムネスによる不確定性）は避けられない。計算過程が正しかったとしても、得られた結果は統計的誤差を含み、真の値ではないという認識は重要である。
- モデルが正しくても、使われる推定法が有効となる条件が満たされなければ、結果の利用は慎重にすべきである。たとえば、確率過程のエルゴード性（大数の法則）が成り立つことを前提にしている統計量は、観測時間が短いとその条件が崩れるため、理論はその妥当性を保証できない。本講座における適用例はソフトウェアの使用例であって、方法の適用が許される前提条件が各々の状況で満たされていることを保証するものではない。
- 計算に用いられている方法の未発達や瑕疵、パッケージ自身およびそれが依存するパッケージのバグ、確率論的方法の限界など、リスクは残っている。利用は自己責任でお願いしたい。

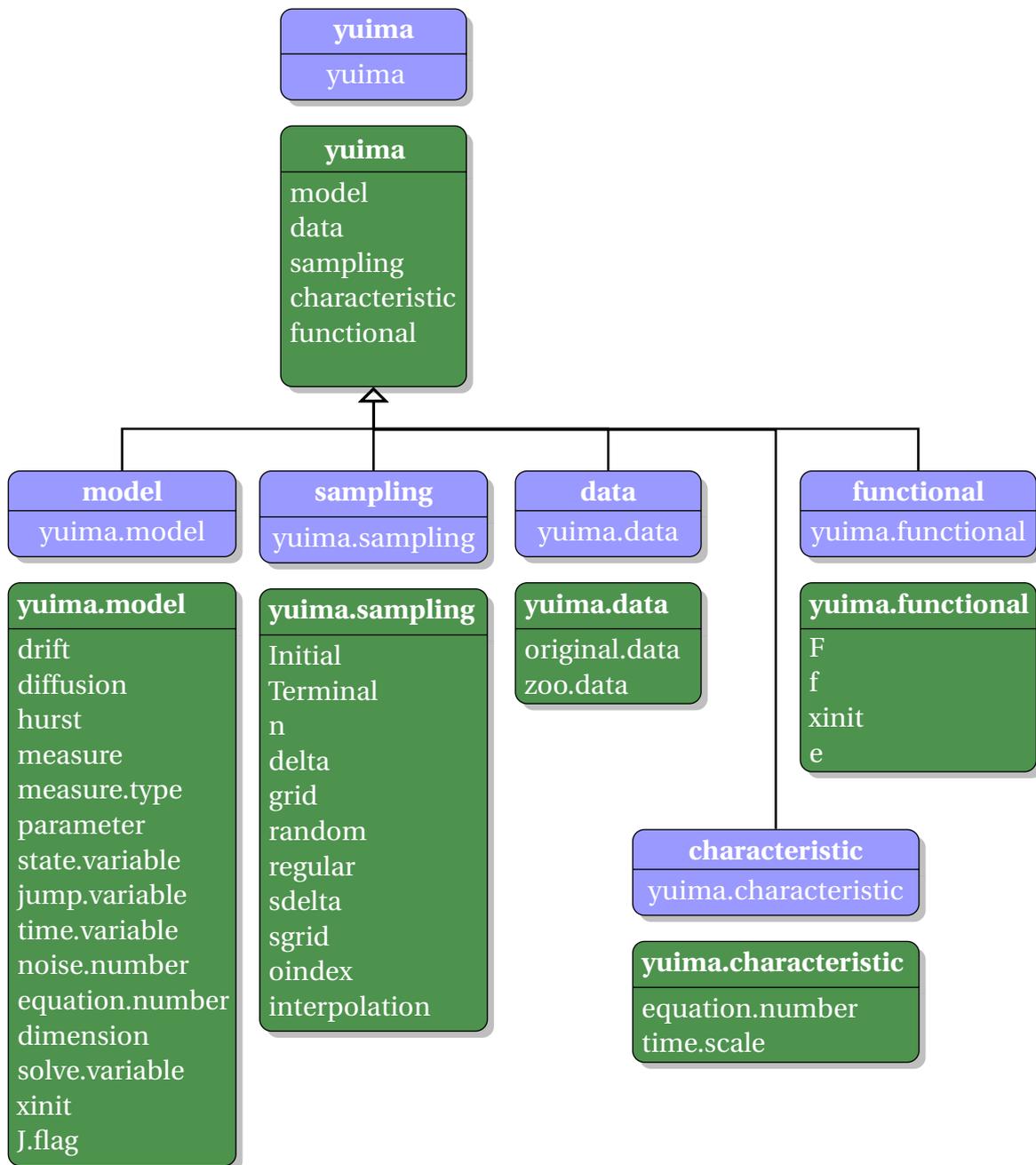


図 1: YUIMA オブジェクトの構造

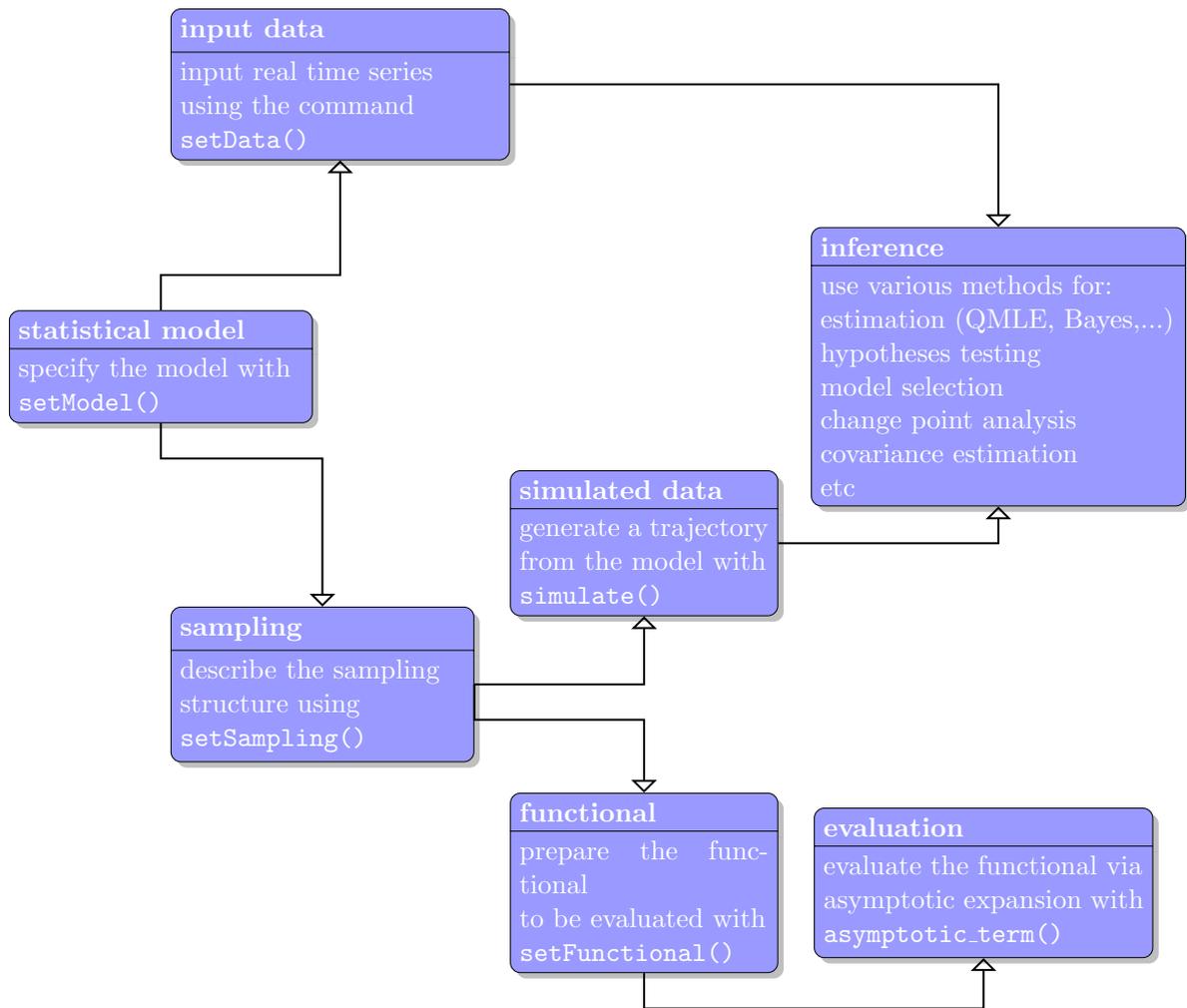


図 2: YUIMA のフロー