

```
In[ ]:= dataGoogl =
  FinancialData["AMZN", "Close", {{2013, 12, 31}, {2020, 7, 20}, "Daily"}];
  [информация о финансовых ин... [заккрыть]
```

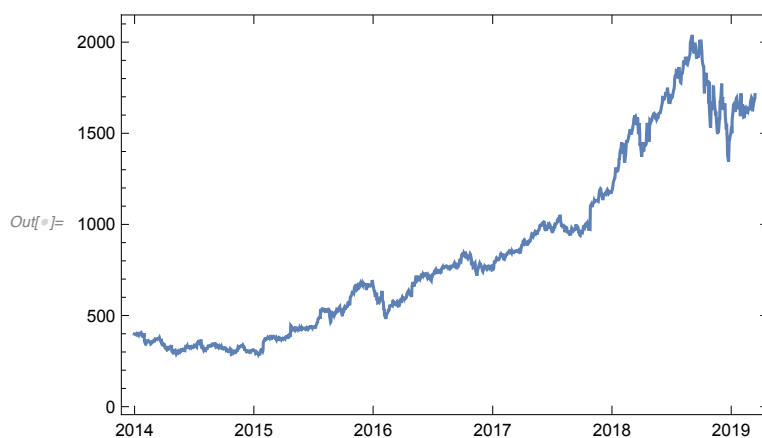
```
In[ ]:= stocksGoogl = QuantityMagnitude[dataGoogl]
  [модуль размерной величины]
```

```
Out[ ]:= TimeSeries[
  [ +  Time: 31 Dec 2013 to 20 Jul 2020
    Data points: 1648 ] ]
```

```
In[ ]:= trainSet = TimeSeriesWindow[stocksGoogl, {{2013, 12, 31}, {2019, 3, 15}}];
  [часть временного ряда в окне]

testSet = TimeSeriesWindow[stocksGoogl, {{2019, 3, 18}, {2020, 7, 2}}];
  [часть временного ряда в окне]
```

```
In[ ]:= DateListPlot[trainSet]
  [график от календарного времени]
```



```
In[ ]:= Clear[a, b, c];
  [очистить]

proc =
  EstimatedProcess[trainSet["Values"], GeometricBrownianMotionProcess[a, b, c]]
  [оценить процесс по данным] [значения] [процесс геометрического броуновского движения]
```

```
Out[ ]:= GeometricBrownianMotionProcess[0.00130354, 0.0194668, 398.79]
```

```
In[ ]:= paths = RandomFunction[
  [случайная функция]
  GeometricBrownianMotionProcess[0.001303539922648453`, 0.019466799007122632`,
  [процесс геометрического броуновского движения]
  testSet["FirstValue"]], {1, testSet["PathLength"], 1}, 200];
```

```
In[ ]:= td = TemporalData[paths["ValueList"],
  [темпоральные данные]
  {testSet["FirstDate"], testSet["LastDate"]}], ValueDimensions -> 1]
  [размерность значений време]
```

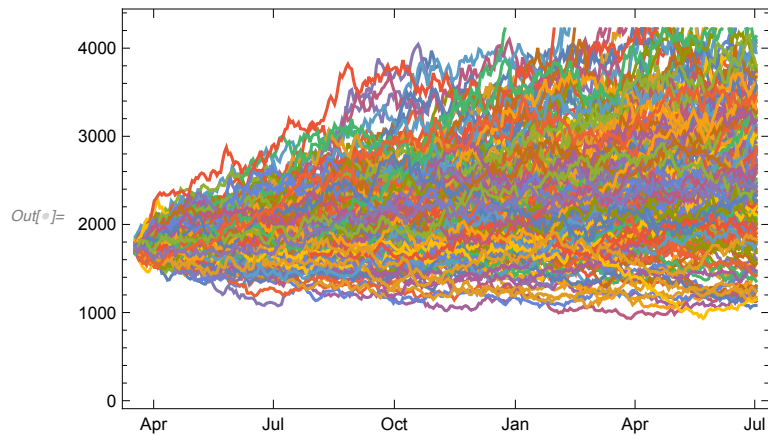
```
In[ ]:= TemporalData[
  [ +  Time: 18 Mar 2019 to 02 Jul 2020
    Data points: 65600 Paths: 200 ] ]
  [темпоральные дан]
```

```
Out[ ]:= TemporalData[  Time: 18 Mar 2019 to 02 Jul 2020  
Data points: 65 600 Paths: 200 ]
```

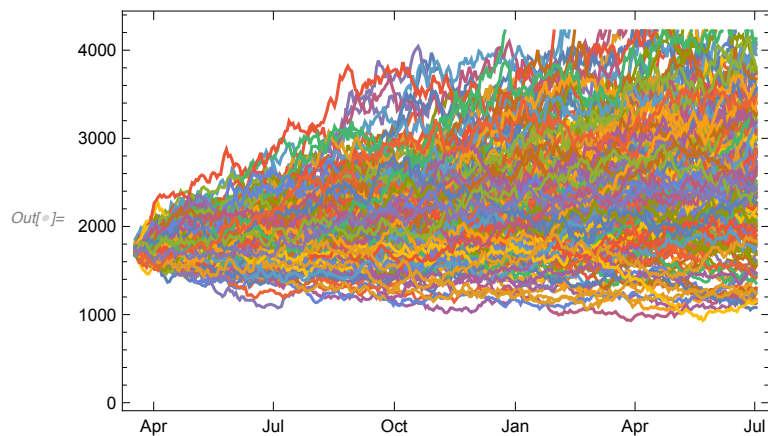
```
In[ ]:= TimeSeriesForecast[proc, 1]  
[прогнозировать значение временного ряда]
```

```
Out[ ]:= TimeSeriesForecast[  
GeometricBrownianMotionProcess[0.00130354, 0.0194668, 398.79], 1]
```

```
In[ ]:= DateListPlot[td]  
[график от календарного времени]
```



```
In[ ]:= pp = DateListPlot[td, Joined → True]  
[график от календарн... [соединё... [истина]
```

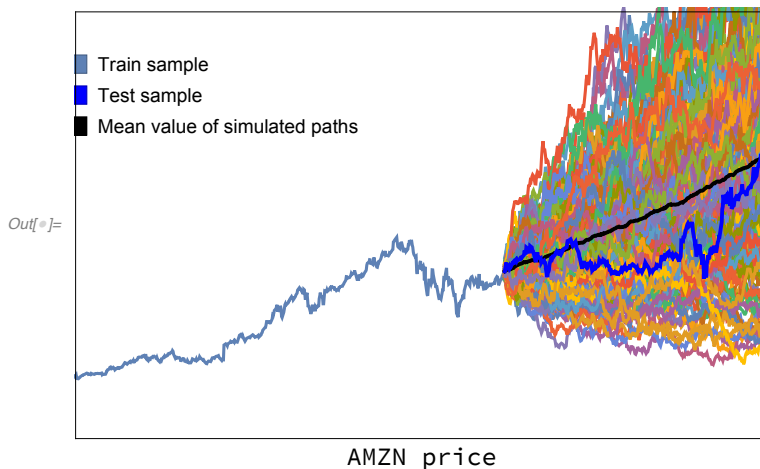


```
In[ ]:= mF = TimeSeriesThread[Mean, td];  
[нанизать значения вр... [среднее значе]
```

```

In[ ]:= Labeled[Legended[Show[DateListPlot[trainSet,
  |с пометкой... |с легендой |показывает график от календарного времени
    PlotRange -> {{DateList[{"01/02/2017", {"Day", "Month", "Year"}}],
      |отображаемый... |дата как список
        td["LastTime"]}], {Min[stocksGoogl], 4000}], Joined -> True], pp,
          |минимум |соединённый... |истина
    DateListPlot[mF, Joined -> True, PlotStyle -> Directive[Thick, Black]],
      |график от календарного... |соединённый... |истина |стиль графика |директива |жирный |чёрный
    DateListPlot[testSet, Joined -> True, PlotStyle -> Directive[Thick, Blue]],
      |график от календарного вре... |соединённый... |истина |стиль графика |директива |жирный |синий
    Placed[SwatchLegend[{{Blue, Black}}, {"Train sample", "Test sample",
      |расположение... |легенда с образцами... |синий |чёрный
        "Mean value of simulated paths"}], {.2, .8}]], "AMZN price"]
      |среднее значение

```



Доходность предсказанная:

```

In[ ]:= mF["Values"][[ -1]] / stocksGoogl["LastValue"] - 1
      |значения

```

Out[ ]:= 0.738675

```

In[ ]:= stocksGoogl["LastValue"]

```

Out[ ]:= 1823.28

```

In[ ]:= mF["Values"][[ -1]]
      |значения

```

Out[ ]:= 3170.09

```

In[ ]:= 2816.552430035657`

```

Out[ ]:= 2816.55

Предскажем остальные фин. активы:




```

In[ ]:= data = FinancialData[{"MPC", "FANG", "UNH", "EQIX", "GOOGL"},
  |информация о финансовых инструментах
    "Close", {{2014, 1, 1}, {2019, 5, 25}, "Daily"}];
  |закрывать

```

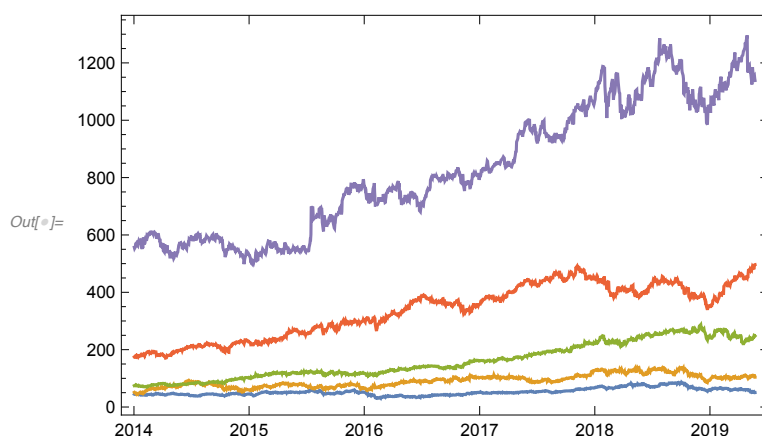
```
In[ ]:= stocks = QuantityMagnitude[data]
```

[\[модуль размерной величины\]](#)

```
Out[ ]:= {TimeSeries[  Time: 02 Jan 2014 to 24 May 2019  
Data points: 1356 ],  
  
TimeSeries[  Time: 02 Jan 2014 to 24 May 2019  
Data points: 1345 ],  
  
TimeSeries[  Time: 02 Jan 2014 to 24 May 2019  
Data points: 1357 ],  
  
TimeSeries[  Time: 02 Jan 2014 to 24 May 2019  
Data points: 1356 ],  
  
TimeSeries[  Time: 02 Jan 2014 to 24 May 2019  
Data points: 1356 ] }
```

```
In[ ]:= DateListPlot[stocks]
```

[\[график от календарного времени\]](#)



```
In[ ]:= Clear[a, b, c];
```

[\[очистить\]](#)

```
proc2 = EstimatedProcess[stocks[ [#] ] ["Values"],  
  \[оценить процесс по данным\] \[значения\]  
  GeometricBrownianMotionProcess[a, b, c] & /@ Range[1, 5]  
  \[процесс геометрического броуновского движения\] \[диапазон\]
```

```
Out[ ]:= {GeometricBrownianMotionProcess[0.000294287, 0.0203954, 44.74],  
  GeometricBrownianMotionProcess[0.000825253, 0.0243563, 51.16],  
  GeometricBrownianMotionProcess[0.000972771, 0.0132416, 74.57],  
  GeometricBrownianMotionProcess[0.000869431, 0.0140568, 174.76],  
  GeometricBrownianMotionProcess[0.000635756, 0.0147105, 557.089] }
```

```
In[ ]:= paths2 = RandomFunction[proc2[ [#] ], {stocks[ [#] ] ["PathLength"],  
  \[случайная функция\]  
  stocks[ [#] ] ["PathLength"] + 120, 1}, 1000] & /@ Range[1, 5];  
  \[диапазон\]
```

```
In[ ]:= td2 = TemporalData[paths2[[#]]["ValueList"],
  {data[[#]]["LastDate"], Automatic}, ValueDimensions -> 1] & /@ Range[1, 5]
```

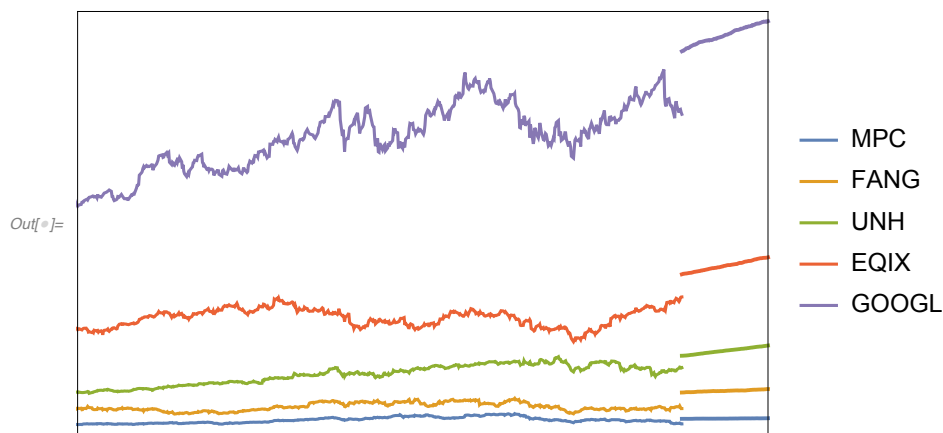
```
Out[ ]:= {TemporalData[
  TemporalData[
    TemporalData[
      TemporalData[
        TemporalData[
          Time: 24 May 2019 to 21 Sep 2019
          Data points: 121 000 Paths: 1000
        ],
        Time: 24 May 2019 to 21 Sep 2019
        Data points: 121 000 Paths: 1000
      ],
      Time: 24 May 2019 to 21 Sep 2019
      Data points: 121 000 Paths: 1000
    ],
    Time: 24 May 2019 to 21 Sep 2019
    Data points: 121 000 Paths: 1000
  ],
  Time: 24 May 2019 to 21 Sep 2019
  Data points: 121 000 Paths: 1000
],
  Time: 24 May 2019 to 21 Sep 2019
  Data points: 121 000 Paths: 1000
],
  Time: 24 May 2019 to 21 Sep 2019
  Data points: 121 000 Paths: 1000
},
  Time: 24 May 2019 to 21 Sep 2019
  Data points: 121 000 Paths: 1000
}]
```

```
In[ ]:=
```

Находим средние значения пересказанных доходностей:

```
In[ ]:=
```

```
In[ ]:= mF2 = TimeSeriesThread[Mean, td2[[#]]] & /@ Range[1, 5];
Show[DateListPlot[stocks, td["LastTime"], Joined -> True,
  PlotLegends -> {"MPC", "FANG", "UNH", "EQIX", "GOOGL"},
  PlotRange -> {{DateList[{"01/02/2017", {"Day", "Month", "Year"}]},
    td2[[1]]["LastTime"]}, {0, 1500}}],
  DateListPlot[mF2, Joined -> True, PlotStyle -> Directive[Thick]]]
```



```
In[ ]:=
```

```
In[*]:= Export["stocks.png", %82]
      ↳экспортировать
```

```
Out[*]:= stocks.png
```

```
In[*]:= SystemOpen["stocks.png"]
      ↳открыть в операционной системе
```

```
In[*]:= SystemOpen[DirectoryName[AbsoluteFileName["stocks.png"]]]
      ↳открыть в оп... ↳название директ... ↳полный путь к файлу
```

Средние предсказанные доходности по акциям:  
MPC:

```
In[*]:= mF2[[1]]["Values"][[-1]] / stocks[[1]]["LastValue"] - 1
      ↳значения
```

```
Out[*]:= 0.396613
```

FANG:

```
In[*]:= mF2[[2]]["Values"][[-1]] / stocks[[2]]["LastValue"] - 1
      ↳значения
```

```
Out[*]:= 0.589719
```

UNH:

```
In[*]:= mF2[[3]]["Values"][[-1]] / stocks[[3]]["LastValue"] - 1
      ↳значения
```

```
Out[*]:= 0.30571
```

EQIX:

```
In[*]:= mF2[[4]]["Values"][[-1]] / stocks[[4]]["LastValue"] - 1
      ↳значения
```

```
Out[*]:= 0.273853
```

GOOGL:

```
In[*]:= mF2[[5]]["Values"][[-1]] / stocks[[5]]["LastValue"] - 1
      ↳значения
```

```
Out[*]:= 0.230699
```