

```
pip install pomegranate
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pomegranate
  Downloading pomegranate-0.14.8.tar.gz (4.3 MB)
    |████████████████████| 4.3 MB 5.1 MB/s
Installing build dependencies ... done
Getting requirements to build wheel ... done
Preparing wheel metadata ... done
Requirement already satisfied: scipy>=0.17.0 in /usr/local/lib/python3.7/dist-packages (from pomegranate) (1.7.3)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.7/dist-packages (from pomegranate) (6.0)
Requirement already satisfied: numpy>=1.20.0 in /usr/local/lib/python3.7/dist-packages (from pomegranate) (1.21.6)
Requirement already satisfied: joblib>=0.9.0b4 in /usr/local/lib/python3.7/dist-packages (from pomegranate) (1.2.0)
Requirement already satisfied: networkx>=2.4 in /usr/local/lib/python3.7/dist-packages (from pomegranate) (2.6.3)
Building wheels for collected packages: pomegranate
  Building wheel for pomegranate (PEP 517) ... done
    Created wheel for pomegranate: filename=pomegranate-0.14.8-cp37-cp37m-linux_x86_64.whl size=15066337 sha256=47c83136ce5d665dc04e9f96f7
    Stored in directory: /root/.cache/pip/wheels/24/68/69/0eaab474ef1d65abedcd47de8a38ab21d221d329954d7edd24
Successfully built pomegranate
Installing collected packages: pomegranate
Successfully installed pomegranate-0.14.8
```

```
import math
from pomegranate import *

burglary = DiscreteDistribution( { 'T': 0.002, 'F': 0.998 } )

earthquake= DiscreteDistribution( { 'T': 0.001, 'F': 0.999 } )

alarm = ConditionalProbabilityTable(
[[ 'T', 'T', 'T', 0.94 ],
 [ 'T', 'T', 'F', 0.06 ],
 [ 'T', 'F', 'T', 0.95 ],
 [ 'T', 'F', 'F', 0.05 ],
 [ 'F', 'T', 'T', 0.69 ],
 [ 'F', 'T', 'F', 0.31 ],
 [ 'F', 'F', 'T', 0.99 ],
 [ 'F', 'F', 'F', 0.01 ]], [burglary, earthquake] )

david = ConditionalProbabilityTable(
[[ 'T', 'T', 0.91 ],
 [ 'T', 'F', 0.09 ],
 [ 'F', 'T', 0.05 ],
 [ 'F', 'F', 0.95 ]], [alarm] )

sophia = ConditionalProbabilityTable(
[[ 'T', 'T', 0.75 ],
 [ 'T', 'F', 0.25 ],
 [ 'F', 'T', 0.02 ],
 [ 'F', 'F', 0.98 ]], [alarm] )

d1 = State( burglary, name="burglary" )
d2 = State( earthquake, name="earthquake" )
d3 = State( alarm, name="alarm" )
d4 = State( david, name="david" )
d5 = State( sophia, name="sophia" )

#Building the Bayesian Network
network = BayesianNetwork( "Solving the Monty Hall Problem With Bayesian Networks" )
network.add_states(d1, d2, d3, d4, d5)
network.add_edge(d1, d3)
network.add_edge(d2, d3)
network.add_edge(d3, d4)
network.add_edge(d3, d5)
network.bake()

beliefs = network.predict_proba({ 'burglary' : 'F', 'earthquake' : 'F', 'david' : 'T', 'sophia' : 'T' })
beliefs = map(str, beliefs)
print("\n".join( "{}t{}".format( state.name, belief ) for state, belief in zip( network.states, beliefs ) ))

burglarytFearthquaketFalarmt{
  "class" : "Distribution",
  "dtype" : "str",
  "name" : "DiscreteDistribution",
  "parameters" : [
```

```
{
  "F" : 1.4799795762825234e-05,
  "T" : 0.9999852002042372
},
"frozen" : false
}ndavidtTnsophiatT
```

---

[Colab paid products](#) - [Cancel contracts here](#)

