Machine Learning in Python with No Strings Attached

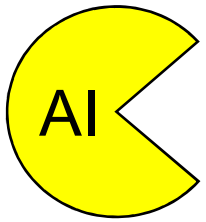
Guillaume Baudart, Martin Hirzel, Kiran Kate, Louis Mandel, and Avi Shinnar

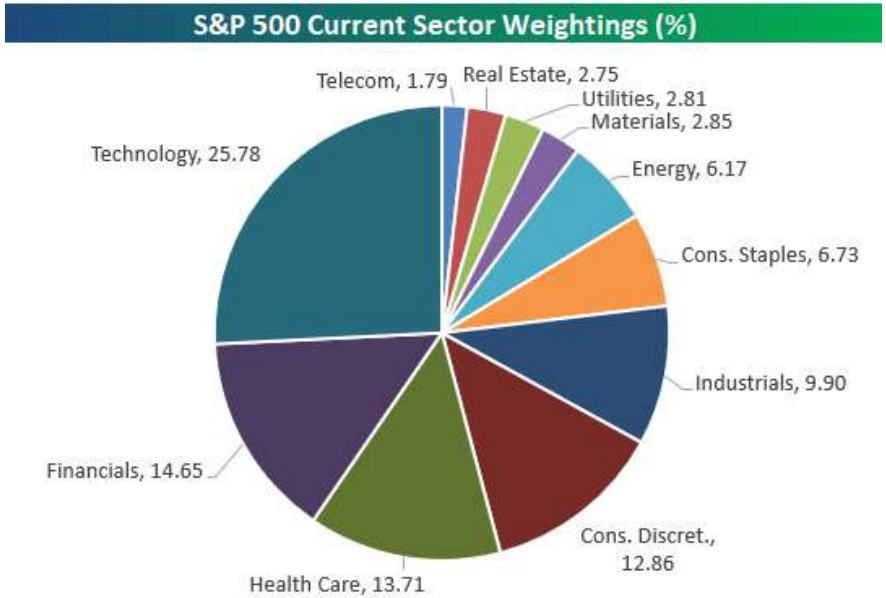
IBM Research, USA

6/22/2019 talk at MAPL Workshop

Why PL for AI?

- IBM customers want AI
  - Large non-tech companies
  - Not enough skilled data scientists
- PL can lower the required skill level
  - Consistency, conciseness, error checking

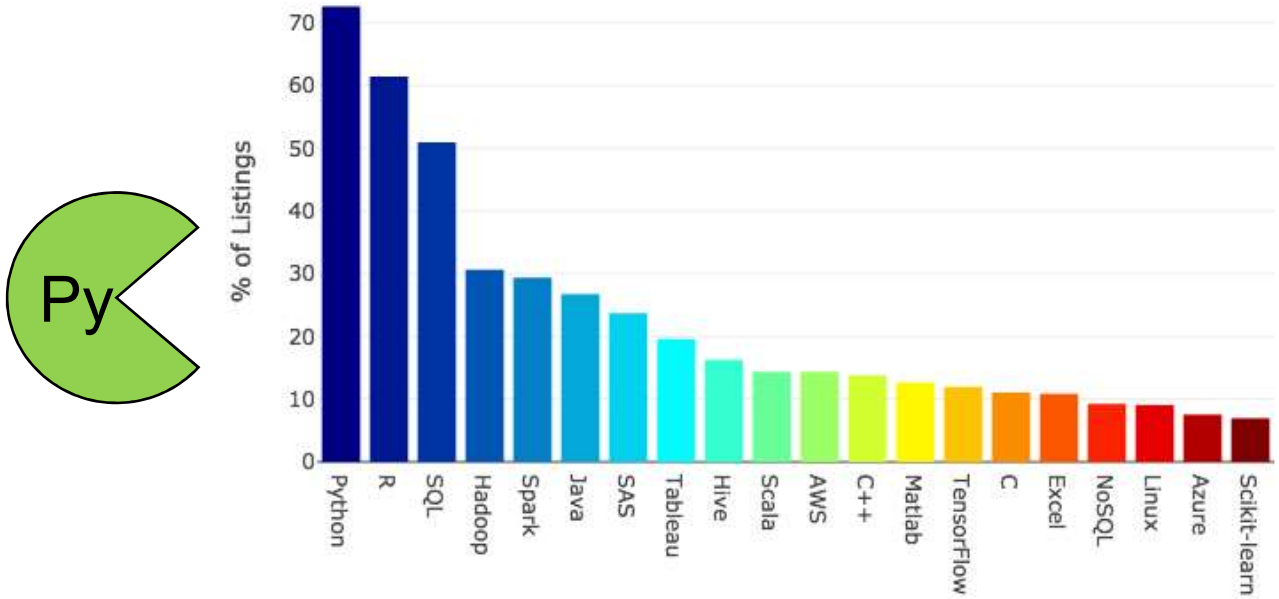




Source: https://seekingalpha.com/article/4172093-s-and-p-500-sector-weightings-tech-nears-26-percent

Why Python?

- Good PL design: consistency, conciseness, …
- Many libraries: Keras, Sklearn, Spark, …
- Jupyter notebooks
- Popularity
  - Top 20 technology skills in data science job listings



Source: https://towardsdatascience.com/the-most-in-demand-skills-for-data-scientists-4a4a8db896db

Why Embedded Languages?

Build computational graph

- Host language: Python
- Guest language: Keras, Sklearn, Spark SQL, …

Evaluate

- Train vs. predict
- Run on GPU or cluster
- Save for later reuse
- Automatic differentiation
- Back-propagation
- Optimization
- Monte-Carlo sampling

Just a library –*but*– also a language.

Keras Example

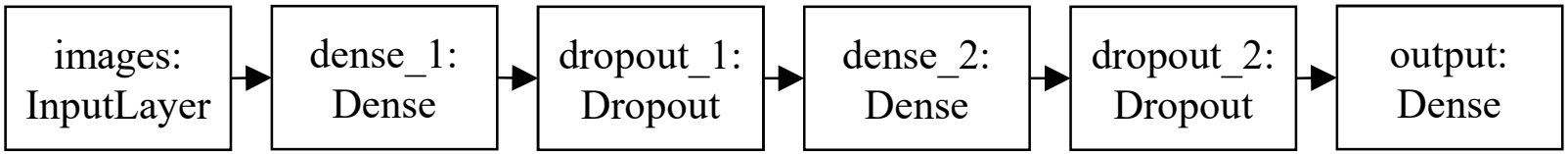
In [2]:
```
import keras
from keras.models import Model
from keras.layers import Dense, Dropout, Input

images = Input(shape=(N_PIXELS,), name='images')
hidden1 = Dense(512, activation='relu')(images)
dropped1 = Dropout(0.2)(hidden1)
hidden2 = Dense(512, activation='relu')(dropped1)
dropped2 = Dropout(0.2)(hidden2)
output = Dense(N_LABELS, activation='softmax',
               name='output')(dropped2)

model = Model(inputs=images, outputs=output)
```

In [3]:
```
keras_utils.plot_model(model)
```

Out[3]:



In [5]:
```
model.compile(loss='categorical_crossentropy',
              optimizer='RMSprop',
              metrics=['accuracy'])
model.fit({'images': x_train}, {'output': y_train},
          batch_size=128, epochs=3,
          validation_data=(x_test, y_test))
```

Outline

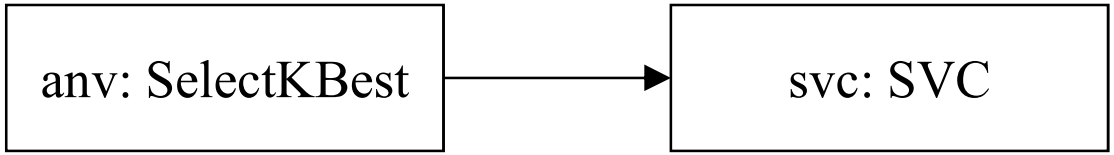
Strings Considered Harmful

Name Reflection: Keras.na

Reinterpreted Python: YAPS

Sklearn Example

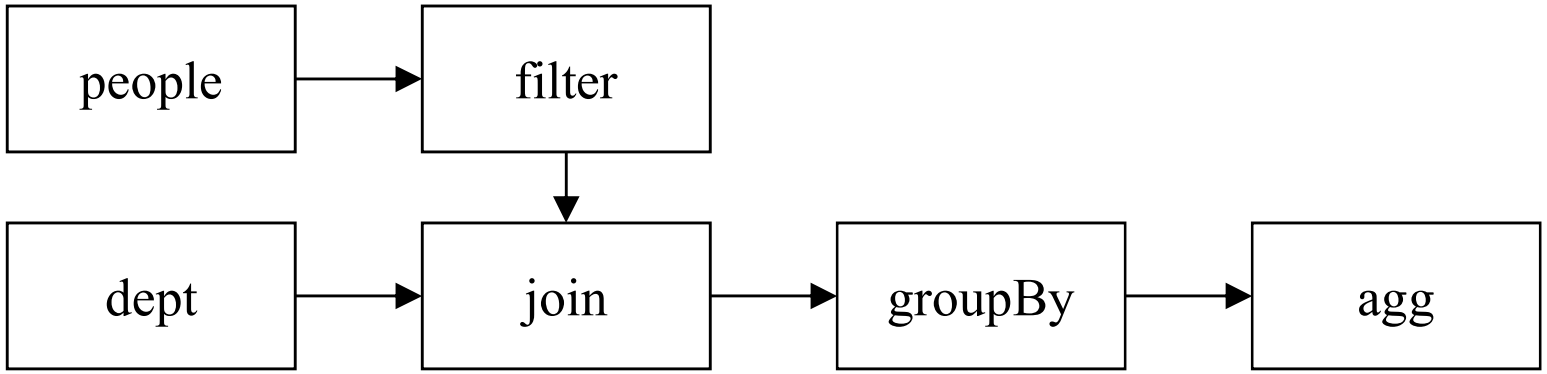
```
filter = SelectKBest(f_regression, k=5)
clf = svm.SVC(kernel='linear')
anv_svm = Pipeline([('anv', filter), ('svc', clf)])
anv_svm.set_params(anv__k=10, svc__C=.1).fit(X, y)
```



Source: http://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

Spark SQL Example

```
people = sqlContext.read.parquet("people.parquet")
dept = sqlContext.read.parquet("dept.parquet")
people.filter(people.age > 30) \
  .join(dept, people.deptId == dept.id) \
  .groupBy(dept.name, "gender") \
  .agg({"salary": "avg", "age": "max"})
```



Source: https://spark.apache.org/docs/2.2.0/api/python/pyspark.sql.html#pyspark.sql.DataFrame

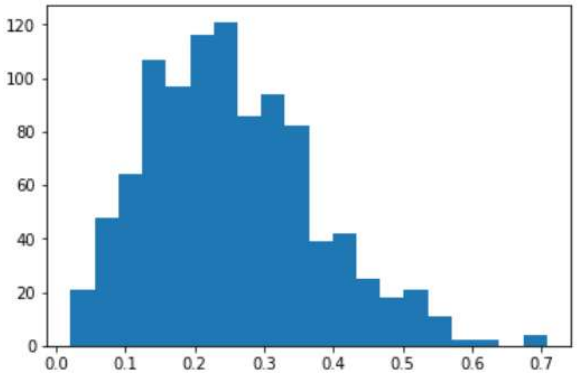
PyStan Example

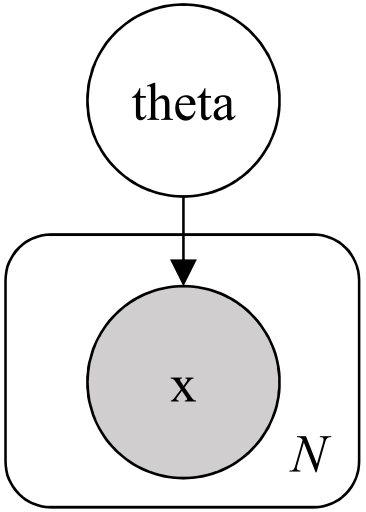
In [2]:
```
model = Model('''
data {
  int<lower=0,upper=1> x[10];
}
parameters {
  real<lower=0,upper=1> theta;
}
model {
  theta ~ uniform(0,1);
  for (i in 1:10)
    x[i] ~ bernoulli(theta);
}''')
```

In [3]:
```
flips = np.array([0, 1, 0, 0, 0, 0, 0, 0, 0, 1])
run = model.sample(data={'x': flips}, random_='seed=42')
```

In [4]:
```
theta = run['theta']
plt.hist(theta, bins=20)
print('mean of theta: {:.3f}'.format(theta.mean()))
```

```
mean of theta: 0.254
```





Outline

Strings Considered Harmful

Name Reflection: Keras.na

Reinterpreted Python: YAPS

Keras with Fewer Strings

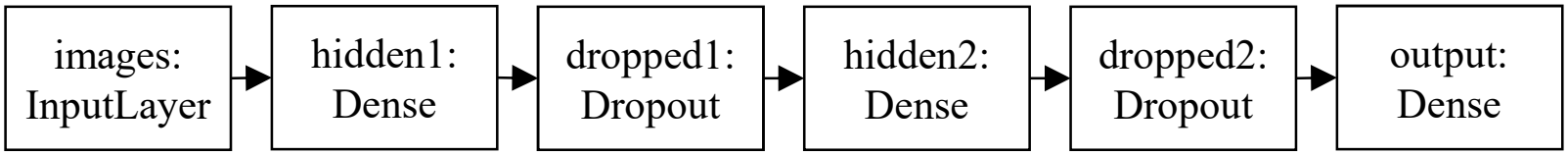
In [2]:
```
import keras
from keras.models import Model
from keras.layers import Dense, Dropout, Input
keras_utils.wrap_layers(globals())

images = Input(shape=(N_PIXELS,))
hidden1 = Dense(512, activation='relu')(images)
dropped1 = Dropout(0.2)(hidden1)
hidden2 = Dense(512, activation='relu')(dropped1)
dropped2 = Dropout(0.2)(hidden2)
output = Dense(N_LABELS, activation='softmax')(dropped2)

model = Model(inputs=images, outputs=output)
```

In [3]:
```
keras_utils.plot_model(model)
```

Out[3]:



In [5]:
```
model.compile(loss='categorical_crossentropy',
              optimizer='RMSprop',
              metrics=['accuracy'])
model.fit({images: x_train}, {output: y_train},
          batch_size=128, epochs=3,
          validation_data=(x_test, y_test))
```

Name Reflection

```
1  import traceback
2  tb = traceback.extract_stack()
3  file_name, line_number, function_name, text = tb[-2]
4  import ast
5  tree = ast.parse(text, file_name)
6  assert isinstance(tree, ast.Module)
7  if isinstance(tree.body[0], ast.Assign):
8      targets = tree.body[0].targets
9      if len(targets) == 1 and isinstance(targets[0], ast.Name):
10         name = targets[0].id
```

Naming Heuristics

Case	Example	Name mapping
If explicit `name` argument	`aux_out = Dense(…,` `  name='output2')(input)`	`output2`
Else if assigned to variable with unique name	`aux_out = Dense(…)(input)`	`aux_out`
Else	`h = Dense(…)(input)` `h = Dense(…)(h)`	`h` `dense_2`

Reverse Name Mapping

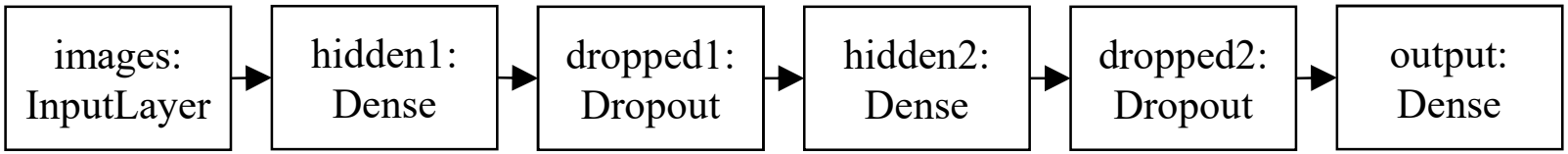
In [2]:
```
import keras
from keras.models import Model
from keras.layers import Dense, Dropout, Input
keras_utils.wrap_layers(globals())

images = Input(shape=(N_PIXELS,))
hidden1 = Dense(512, activation='relu')(images)
dropped1 = Dropout(0.2)(hidden1)
hidden2 = Dense(512, activation='relu')(dropped1)
dropped2 = Dropout(0.2)(hidden2)
output = Dense(N_LABELS, activation='softmax')(dropped2)

model = Model(inputs=images, outputs=output)
```

In [3]:
```
keras_utils.plot_model(model)
```

Out[3]:



In [5]:
```
model.compile(loss='categorical_crossentropy',
              optimizer='RMSprop',
              metrics=['accuracy'])
model.fit({images: x_train}, {output: y_train},
          batch_size=128, epochs=3,
          validation_data=(x_test, y_test))
```

- Names images, output are in scope (dict key does not start scope)
- Model wrapper re-keys the dictionary with strings from reflection

Keras.na Error Messages

In [1]:
```
import keras
from keras.models import Model
from keras.layers import Input, Conv2D
import keras_utils
keras_utils.wrap_layers(globals())

img = Input(shape=(28,28))
conv = Conv2D(filters=10, kernel_size=3)(img)
model = keras.models.Model(inputs=img, outputs=conv)
model.compile(optimizer='sgd', loss='mean_squared_error')
```

```
Using TensorFlow backend.

ValueError: Input 0 is incompatible with layer conv: expected ndi
m=4, found ndim=3
```

Outline

Strings Considered Harmful

Name Reflection: Keras.na

Reinterpreted Python: YAPS

Why Stan Probabilistic PL

- Widely adopted in the sciences
- Hundreds of examples
- Text books
- Dedicated conference
- High-level and self-contained language

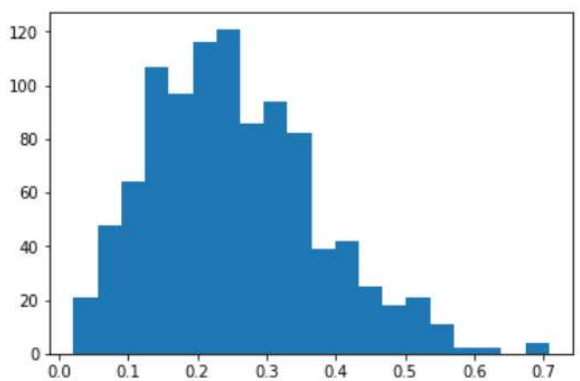
YAPS Example

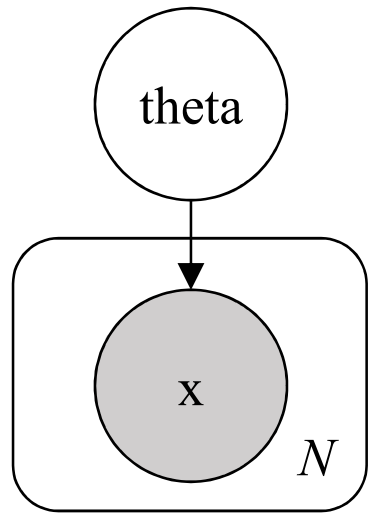
```
In [2]: @yaps.model
        def coin(x: int(lower=0, upper=1)[10]):
            theta: real(lower=0, upper=1) <- uniform(0, 1)
            for i in range(1,11):
                x[i] <- bernoulli(theta)
```

```
In [3]: flips = np.array([0, 1, 0, 0, 0, 0, 0, 0, 0, 1])
        coin_obs = coin(x=flips)
        coin_obs.sample(data=coin_obs.data, random_='seed=42')
```

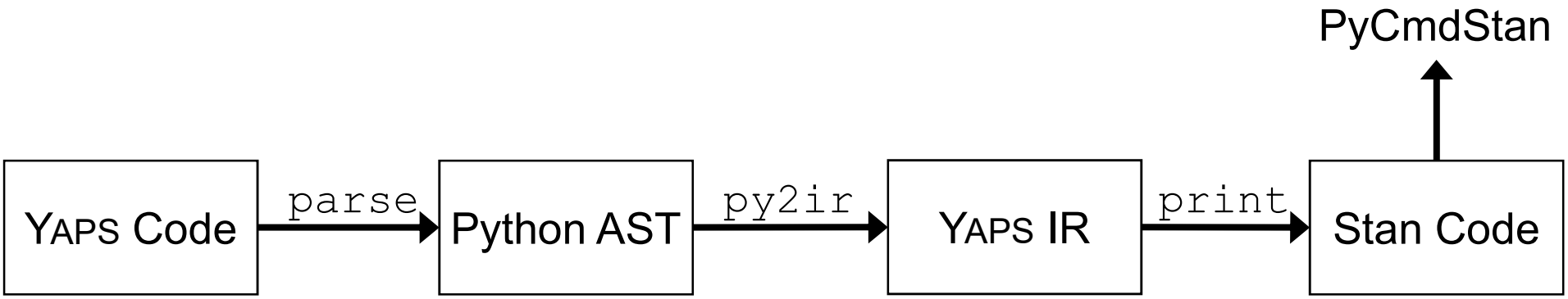
```
In [4]: theta = coin_obs.posterior.theta
        plt.hist(theta, bins=20)
        print('mean of theta: {:.3f}'.format(theta.mean()))
```

```
mean of theta: 0.254
```





Reinterpreted Python



- User interacts with Yaps just like with any other Python library.
- Decorator `@yaps.model` replaces function, internally triggers Stan code generation.
- PyCmdStan takes care of the rest, invoking external Stan and C++ compilers.

Standard Python Tooling

```
@yaps.model
def coin(x: int(lower=0,
    theta: real(lower=0,
    for i in range(1, 11
        x[i] is bernoulli(thetap)
```

Undefined variable 'thetap'

Quick Fix... Peek Problem

Reverse Source Mapping

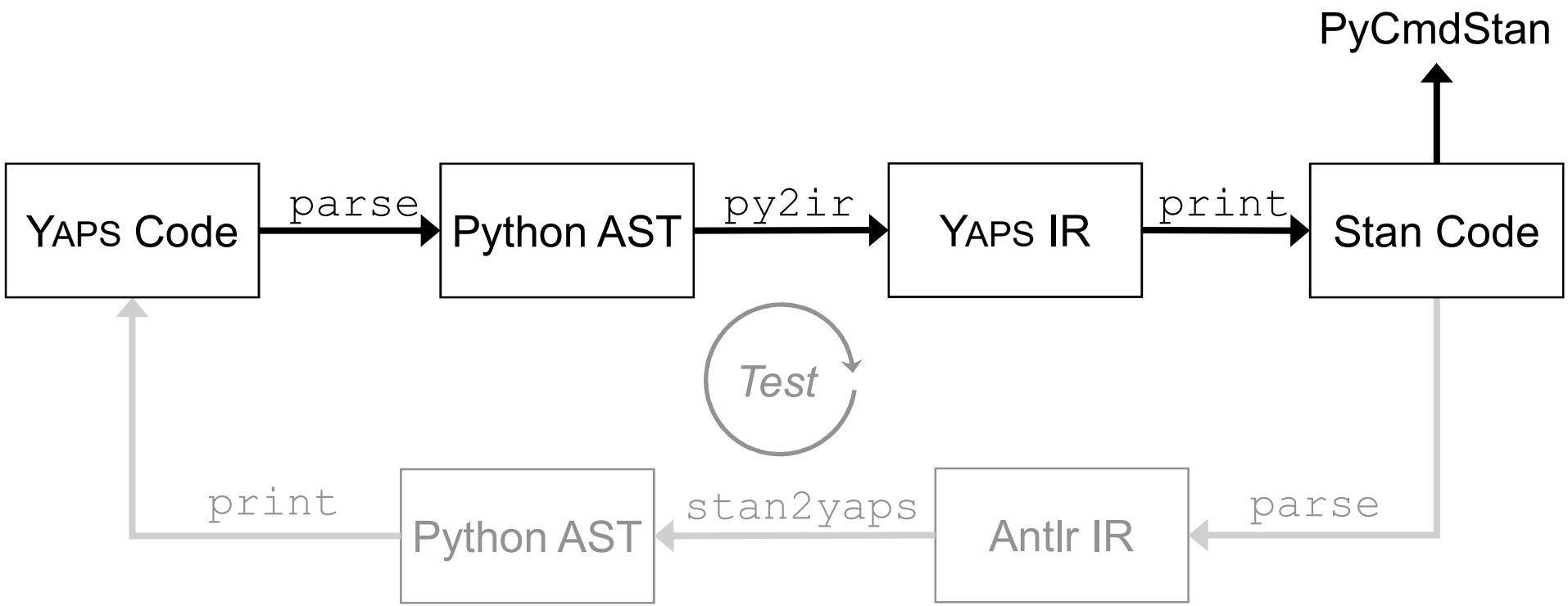
In [2]:
```
@yaps.model
def coin(x: int(lower=0, upper=1)[10]):
    for i in range(1,11):
        x[i] <~ bernoulli(theta)
```

In [3]:
```
import numpy as np
flips = np.array([0, 1, 0, 0, 0, 0, 0, 0, 0, 1])
coin_obs = coin(x=flips)
coin_obs.sample(data=coin_obs.data)
```

```
SYNTAX ERROR, MESSAGE(S) FROM PARSER:

variable "theta" does not exist.
  error in '/Users/[redacted]/.cache/pycmdstan/model-3aed19a9.stan
' at line 4, column 27
-------------------------------------------------
   1: @yaps.model
   2: def coin(x: int(lower=0, upper=1)[10]):
   3:     for i in range(1,11):
   4:         x[i] <~ bernoulli(theta)
                                ^
-------------------------------------------------
```

Roundtrip Validation



	Total	Success
User manual	61	100%
Dev repo	721	97%
Examples repo	500	82%

The failed cases used deprecated Stan syntax.

YAPS Discussion

- Runtime validation
  - 13 example Stan models with datasets
  - Translated to YAPS, then back to Stan
  - Ran inference, found identical results
- Limitations
  - Typevar declaration for dependent types
  - Name clash with Python and Stan keywords
  - Debugger integration
- Open-source
  - https://github.com/ibm/yaps/
  - https://pypi.org/project/yaps/

Conclusions

Strings attached	No strings attached	Where to find
Keras	Keras.na	This paper
PyStan	YAPS	This paper; github
Sklearn	LALE	arXiv:1906.03957
Spark SQL	?	
?		

Source: *Type-Driven Automated Learning with Lale*. Hirzel/Kate/Shinnar/Roy/Ram. arXiv:1906.03957, 2019.