

```
In [41]: ► with pm.Model() as model:
    mus = [
        MvNormal("mu_%d" % i, mu=pm.floatX(np.zeros(2)), tau=pm.floatX(0.1 * np.eye(2)), shape=(2,))
        for i in range(2)
    ]
    pi = Dirichlet("pi", a=pm.floatX(0.1 * np.ones(2)), shape=(2,))
    xs = DensityDist("x", logp_gmix(mus, pi, np.eye(2)), observed=data)

    start = find_MAP()
    step = Metropolis()
    trace = sample(1000, step, start=start)
```

100.00% [35/35 00:01<00:00 logp = -3.6535e+05, ||grad|| = 0.0029943]

C:\Users\Mengj\AppData\Local\Temp\ipykernel\_3628\1911979648.py:11: DeprecationWarning: Call to deprecated Parameter start. (renamed to `initvals` in PyMC v4.0.0) -- Deprecated since v3.11.5.  
 trace = sample(1000, step, start=start)

C:\Users\Mengj\anaconda3\envs\pm3env\lib\site-packages\deprecat\classic.py:215: FutureWarning: In v4.0, pm.sample will return an `arviz.InferenceData` object instead of a `MultiTrace` by default. You can pass return\_inferencedata=True or return\_inferencedata=False to be safe and silence this warning.  
 return wrapped\_(\*args\_, \*\*kwargs\_)

Multiprocess sampling (4 chains in 4 jobs)  
 CompoundStep  
 >Metropolis: [pi]  
 >Metropolis: [mu\_1]  
 >Metropolis: [mu\_0]

```

-----
TypeError                                 Traceback (most recent call last)
Cell In[41], line 11
      9 start = find_MAP()
     10 step = Metropolis()
--> 11 trace = sample(1000, step, start=start)

File ~\anaconda3\envs\pm3env\lib\site-packages\deprecat\classic.py:215, in deprecat.<locals>.wrap
per_function(wrapped_, instance_, args_, kwargs_)
     213     else:
     214         warnings.warn(message, category=category, stacklevel=_routine_stacklevel)
--> 215 return wrapped_(*args_, **kwargs_)

File ~\anaconda3\envs\pm3env\lib\site-packages\pymc3\sampling.py:575, in sample(draws, step, in
it, n_init, initvals, trace, chain_idx, chains, cores, tune, progressbar, model, random_seed,
discard_tuned_samples, compute_convergence_checks, callback, jitter_max_retries, start, retur
n_inferencedata, idata_kwargs, mp_ctx, pickle_backend, **kwargs)
     573 _print_step_hierarchy(step)
     574 try:
--> 575     trace = _mp_sample(**sample_args, **parallel_args)
     576 except pickle.PickleError:
     577     _log.warning("Could not pickle model, sampling singlethreaded.")

File ~\anaconda3\envs\pm3env\lib\site-packages\pymc3\sampling.py:1480, in _mp_sample(draws, tun
e, step, chains, cores, chain, random_seed, start, progressbar, trace, model, callback, disca
rd_tuned_samples, mp_ctx, pickle_backend, **kwargs)
     1477     strace.setup(draws + tune, idx + chain)
     1478     traces.append(strace)
-> 1480 sampler = ps.ParallelSampler(
     1481     draws,
     1482     tune,
     1483     chains,
     1484     cores,
     1485     random_seed,
     1486     start,
     1487     step,
     1488     chain,
     1489     progressbar,
     1490     mp_ctx=mp_ctx,
     1491     pickle_backend=pickle_backend,
     1492 )
     1493 try:
     1494     try:

File ~\anaconda3\envs\pm3env\lib\site-packages\pymc3\parallel_sampling.py:423, in ParallelSample
r._init__(self, draws, tune, chains, cores, seeds, start_points, step_method, start_chain_nu
m, progressbar, mp_ctx, pickle_backend)
     421 if mp_ctx.get_start_method() != "fork":
     422     if pickle_backend == "pickle":
--> 423         step_method_pickled = pickle.dumps(step_method, protocol=-1)
     424     elif pickle_backend == "dill":
     425         try:

File ~\anaconda3\envs\pm3env\lib\site-packages\pymc3\distributions\distribution.py:608, in Dens
ityDist.__getstate__(self)
     603 def __getstate__(self):
     604     # We use dill to serialize the logp function, as this is almost
     605     # always defined in the notebook and won't be pickled correctly.
     606     # Fix https://github.com/pymc-devs/pymc3/issues/3844 (https://github.com/pymc-devs/pymc3/issue
s/3844)
     607     try:
--> 608         logp = dill.dumps(self.logp)
     609 except RecursionError as err:
     610     if type(self.logp) == types.MethodType:

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:263, in dumps(obj, protocol, byref
f, fmode, recurse, **kwds)
     239 """
     240 Pickle an object to a string.
     241
     242 (...)

```

```

260 Default values for keyword arguments can be set in :mod:`dill.settings`.
261 """
262 file = StringIO()
--> 263 dump(obj, file, protocol, byref, fmode, recurse, **kwds)#, strictio)
264 return file.getvalue()

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:235, in dump(obj, file, protocol,
byref, fmode, recurse, **kwds)
233 _kwds = kwds.copy()
234 _kwds.update(dict(byref=byref, fmode=fmode, recurse=recurse))
--> 235 Pickler(file, protocol, **_kwds).dump(obj)
236 return

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:394, in Pickler.dump(self, obj)
392 def dump(self, obj): #NOTE: if settings change, need to update attributes
393     logger.trace_setup(self)
--> 394     StockPickler.dump(self, obj)

File ~\anaconda3\envs\pm3env\lib\pickle.py:487, in _Pickler.dump(self, obj)
485 if self.proto >= 4:
486     self.framer.start_framing()
--> 487 self.save(obj)
488 self.write(STOP)
489 self.framer.end_framing()

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, sav
e_persistent_id)
386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
558 f = self.dispatch.get(t)
559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
561     return
563 # Check private dispatch table if any, or else
564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:1824, in save_function(pickler, ob
j)
1821 if state_dict:
1822     state = state, state_dict
-> 1824 _save_with_postproc(pickler, (_create_function, (
1825     obj.__code__, globs, obj.__name__, obj.__defaults__,
1826     closure
1827 ), state), obj=obj, postproc_list=postproc_list)
1829 # Lift closure cell update to earliest function (#458)
1830 if _postproc:

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:1089, in _save_with_postproc(pickle
r, reduction, is_pickler_dill, obj, postproc_list)
1087     continue
1088 else:
-> 1089     pickler.save_reduce(*reduction)
1090 # pop None created by calling preprocessing step off stack
1091 pickler.write(bytes('0', 'UTF-8'))

File ~\anaconda3\envs\pm3env\lib\pickle.py:692, in _Pickler.save_reduce(self, func, args, state,
listitems, dictitems, state_setter, obj)
690 else:
691     save(func)
--> 692     save(args)
693     write(REDUCE)
695 if obj is not None:
696     # If the object is already in the memo, this means it is
697     # recursive. In this case, throw away everything we put on the
698     # stack, and fetch the object back from the memo.

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, sav
e_persistent_id)
386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType

```

```

    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
    558 f = self.dispatch.get(t)
    559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
    561     return
    563 # Check private dispatch table if any, or else
    564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\pickle.py:886, in _Pickler.save_tuple(self, obj)
    884 if n <= 3 and self.proto >= 2:
    885     for element in obj:
--> 886         save(element)
    887     # Subtle. Same as in the big comment below.
    888     if id(obj) in memo:

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:603, in _Pickler.save(self, obj, save_persistent_id)
    599     raise PicklingError("Tuple returned by %s must have "
    600                          "two to six elements" % reduce)
    602 # Save the reduce() output and finally memoize the object
--> 603 self.save_reduce(obj=obj, *rv)

File ~\anaconda3\envs\pm3env\lib\pickle.py:717, in _Pickler.save_reduce(self, func, args, state, listitems, dictitems, state_setter, obj)
    715 if state is not None:
    716     if state_setter is None:
--> 717         save(state)
    718         write(BUILD)
    719     else:
    720         # If a state_setter is specified, call it instead of load_build
    721         # to update obj's with its previous state.
    722         # First, push state_setter and its tuple of expected arguments
    723         # (obj, state) onto the stack.

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
    558 f = self.dispatch.get(t)
    559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
    561     return
    563 # Check private dispatch table if any, or else
    564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:1186, in save_module_dict(pickler, obj)
    1183     if is_dill(pickler, child=False) and pickler._session:
    1184         # we only care about session the first pass thru
    1185         pickler._first_pass = False
-> 1186     StockPickler.save_dict(pickler, obj)
    1187     logger.trace(pickler, "# D2")
    1188     return

File ~\anaconda3\envs\pm3env\lib\pickle.py:971, in _Pickler.save_dict(self, obj)
    968     self.write(MARK + DICT)
    970     self.memoize(obj)
--> 971 self._batch_setitems(obj.items())

File ~\anaconda3\envs\pm3env\lib\pickle.py:997, in _Pickler._batch_setitems(self, items)
    995     for k, v in tmp:

```

```

    996         save(k)
--> 997         save(v)
    998     write(SETITEMS)
    999 elif n:

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:603, in _Pickler.save(self, obj, save_persistent_id)
    599     raise PicklingError("Tuple returned by %s must have "
    600                          "two to six elements" % reduce)
    602 # Save the reduce() output and finally memoize the object
--> 603 self.save_reduce(obj=obj, *rv)

File ~\anaconda3\envs\pm3env\lib\pickle.py:717, in _Pickler.save_reduce(self, func, args, state, listitems, dictitems, state_setter, obj)
    715 if state is not None:
    716     if state_setter is None:
--> 717         save(state)
    718         write(BUILD)
    719     else:
    720         # If a state_setter is specified, call it instead of load_build
    721         # to update obj's with its previous state.
    722         # First, push state_setter and its tuple of expected arguments
    723         # (obj, state) onto the stack.

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
    558 f = self.dispatch.get(t)
    559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
    561     return
    563 # Check private dispatch table if any, or else
    564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:1186, in save_module_dict(pickler, obj)
    1183     if is_dill(pickler, child=False) and pickler._session:
    1184         # we only care about session the first pass thru
    1185         pickler._first_pass = False
-> 1186     StockPickler.save_dict(pickler, obj)
    1187     logger.trace(pickler, "# D2")
    1188 return

File ~\anaconda3\envs\pm3env\lib\pickle.py:971, in _Pickler.save_dict(self, obj)
    968     self.write(MARK + DICT)
    970 self.memoize(obj)
--> 971 self._batch_setitems(obj.items())

File ~\anaconda3\envs\pm3env\lib\pickle.py:997, in _Pickler._batch_setitems(self, items)
    995     for k, v in tmp:
    996         save(k)
--> 997         save(v)
    998     write(SETITEMS)
    999 elif n:

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
    558 f = self.dispatch.get(t)

```

```

    559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
    561     return
    563 # Check private dispatch table if any, or else
    564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\pickle.py:931, in _Pickler.save_list(self, obj)
    928     self.write(MARK + LIST)
    930 self.memoize(obj)
--> 931 self._batch_appends(obj)

File ~\anaconda3\envs\pm3env\lib\pickle.py:958, in _Pickler._batch_appends(self, items)
    956     write(APPENDS)
    957 elif n:
--> 958     save(tmp[0])
    959     write(APPEND)
    960 # else tmp is empty, and we're done

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:603, in _Pickler.save(self, obj, save_persistent_id)
    599     raise PicklingError("Tuple returned by %s must have "
    600                          "two to six elements" % reduce)
    602 # Save the reduce() output and finally memoize the object
--> 603 self.save_reduce(obj=obj, *rv)

File ~\anaconda3\envs\pm3env\lib\pickle.py:687, in _Pickler.save_reduce(self, func, args, state, listitems, dictitems, state_setter, obj)
    684     raise PicklingError(
    685         "args[0] from __newobj__ args has the wrong class")
    686 args = args[1:]
--> 687 save(cls)
    688 save(args)
    689 write(NEWOBJ)

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
    558 f = self.dispatch.get(t)
    559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
    561     return
    563 # Check private dispatch table if any, or else
    564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\dill.py:1698, in save_type(pickler, obj, postproc_list)
    1696         postproc_list = []
    1697         postproc_list.append((setattr, (obj, '__qualname__', obj_name)))
-> 1698     _save_with_postproc(pickler, (_create_type, (
    1699         type(obj), obj.__name__, obj.__bases__, _dict
    1700     )), obj=obj, postproc_list=postproc_list)
    1701     logger.trace(pickler, "# T2")
    1702 else:

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\dill.py:1070, in _save_with_postproc(pickler, reduction, is_pickler_dill, obj, postproc_list)
    1067     pickler._postproc[id(obj)] = postproc_list
    1069 # TODO: Use state_setter in Python 3.8 to allow for faster cPickle implementations
-> 1070 pickler.save_reduce(*reduction, obj=obj)
    1072 if is_pickler_dill:
    1073     # pickler.x -= 1
    1074     # print(pickler.x* ' ', 'pop', obj, id(obj))
    1075     postproc = pickler._postproc.pop(id(obj))

```

```

File ~\anaconda3\envs\pm3env\lib\pickle.py:692, in _Pickler.save_reduce(self, func, args, state,
listitems, dictitems, state_setter, obj)
    690 else:
    691     save(func)
--> 692     save(args)
    693     write(REDUCE)
    695 if obj is not None:
    696     # If the object is already in the memo, this means it is
    697     # recursive. In this case, throw away everything we put on the
    698     # stack, and fetch the object back from the memo.

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, sav
e_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
    558 f = self.dispatch.get(t)
    559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
    561     return
    563 # Check private dispatch table if any, or else
    564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\pickle.py:901, in _Pickler.save_tuple(self, obj)
    899 write(MARK)
    900 for element in obj:
--> 901     save(element)
    903 if id(obj) in memo:
    904     # Subtle. d was not in memo when we entered save_tuple(), so
    905     # the process of saving the tuple's elements must have saved
    (... )
    909     # could have been done in the "for element" loop instead, but
    910     # recursive tuples are a rare thing.
    911     get = self.get(memo[id(obj)][0])

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, sav
e_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:560, in _Pickler.save(self, obj, save_persistent_id)
    558 f = self.dispatch.get(t)
    559 if f is not None:
--> 560     f(self, obj) # Call unbound method with explicit self
    561     return
    563 # Check private dispatch table if any, or else
    564 # copyreg.dispatch_table

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:1186, in save_module_dict(pickler,
obj)
    1183     if is_dill(pickler, child=False) and pickler._session:
    1184         # we only care about session the first pass thru
    1185         pickler._first_pass = False
-> 1186     StockPickler.save_dict(pickler, obj)
    1187     logger.trace(pickler, "# D2")
    1188     return

File ~\anaconda3\envs\pm3env\lib\pickle.py:971, in _Pickler.save_dict(self, obj)
    968     self.write(MARK + DICT)
    970     self.memoize(obj)
--> 971     self._batch_setitems(obj.items())

File ~\anaconda3\envs\pm3env\lib\pickle.py:997, in _Pickler._batch_setitems(self, items)
    995     for k, v in tmp:
    996         save(k)
--> 997         save(v)
    998     write(SETITEMS)
    999     elif n:

```

```

File ~\anaconda3\envs\pm3env\lib\site-packages\dill\_dill.py:388, in Pickler.save(self, obj, save_persistent_id)
    386     msg = "Can't pickle %s: attribute lookup builtins.generator failed" % GeneratorType
    387     raise PicklingError(msg)
--> 388 StockPickler.save(self, obj, save_persistent_id)

File ~\anaconda3\envs\pm3env\lib\pickle.py:578, in _Pickler.save(self, obj, save_persistent_id)
    576 reduce = getattr(obj, "__reduce_ex__", None)
    577 if reduce is not None:
--> 578     rv = reduce(self.proto)
    579 else:
    580     reduce = getattr(obj, "__reduce__", None)

TypeError: cannot pickle '_abc_data' object

```

```

In [52]: ▶ minibatch_size = int(200)
s# In memory Minibatches for better speed
data_t = pm.Minibatch(data, minibatch_size)

with pm.Model() as model:
    mus = [
        MvNormal("mu_%d" % i, mu=pm.floatX(np.zeros(2)), tau=pm.floatX(0.1 * np.eye(2)), shape=(2,))
        for i in range(2)
    ]
    pi = Dirichlet("pi", a=pm.floatX(0.1 * np.ones(2)), shape=(2,))
    xs = DensityDist("x", logp_gmix(mus, pi, np.eye(2)), observed=data_t, total_size=len(data))

```

```

-----
IndexError                                Traceback (most recent call last)
Cell In[52], line 4
      1 minibatch_size = int(200)
      3 # In memory Minibatches for better speed
----> 4 data_t = pm.Minibatch(data, minibatch_size)
      6 with pm.Model() as model:
      7     mus = [
      8         MvNormal("mu_%d" % i, mu=pm.floatX(np.zeros(2)), tau=pm.floatX(0.1 * np.eye(2)), shape=
(2,))
      9         for i in range(2)
     10     ]

File ~\anaconda3\envs\pm3env\lib\site-packages\theano\configparser.py:48, in _ChangeFlagsDecorator.__call__.<locals>.res(*args, **kwargs)
     45 @wraps(f)
     46 def res(*args, **kwargs):
     47     with self:
--> 48         return f(*args, **kwargs)

File ~\anaconda3\envs\pm3env\lib\site-packages\pymc3\data.py:316, in Minibatch.__init__(self, data, batch_size, dtype, broadcastable, name, random_seed, update_shared_f, in_memory_size)
     314     data = np.asarray(data, dtype)
     315     in_memory_slc = self.make_static_slices(in_memory_size)
--> 316     self.shared = theano.shared(data[in_memory_slc])
     317     self.update_shared_f = update_shared_f
     318     self.random_slc = self.make_random_slices(self.shared.shape, batch_size, random_seed)

IndexError: only integers, slices (:`), ellipsis (...), numpy.newaxis (None) and integer or boolean arrays are valid indices

```

```

In [51]: ▶ int(200)

```

Out[51]: 200