Code

```
from matplotlib import pyplot as plt
import subprocess


def form_command(a,b,c,n, optimised = True):
  if optimised:
    return "../d4-7/dineroIV -l1-usize {}K -l1-ubsize {} -l1-uassoc {} -l1-uwalloc a -l1-uwback a -l1-uccc  -informat d < ../trace_files/optimised/nopt{}.din".format(c,b,a,n)
  else:
    return "../d4-7/dineroIV -l1-usize {}K -l1-ubsize {} -l1-uassoc {} -l1-uwalloc a -l1-uwback a -l1-uccc  -informat d < ../trace_files/unoptimised/n{}.din".format(c,b,a,n)



# variation in associativity
print('==============================================================================')
print('Variation in Associativity')
print('==============================================================================')
sampler = [1, 2, 4, 8, 16, 32]
fig, ax = plt.subplots(2,2, figsize = (15,15))
fig.suptitle('Variation in a')
for i, (b,c,n) in enumerate([(64,4,100),(64,8,100),(128,4,100),(128,8,100)]):
  ans_opt = []
  ans_unopt = []
  f = open('../csvs/variationa/{}_{}_{}.csv'.format(b,c,n), 'w')
  f.write('Opt,a,b,c,n,Total Fetches,Total Misses,Miss Rate,Compulsory miss, Conflict miss, Capacity miss\n')
  f.flush()
  for a in sampler:
    print("a = ",a,", b = ",b,", c = ",c,", n = ",n)
    bashCommand = form_command(a,b,c,n, optimised = True)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_opt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('Yes',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))

```

```python
    bashCommand = form_command(a,b,c,n, optimised = False)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_unopt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('No',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))
  f.flush()
  f.close()
  ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='optimised')
  ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='unoptimised')
  ax[i//2,i%2].set_xlabel('Associativity')
  ax[i//2,i%2].set_ylabel('Demand Miss Rate')
  ax[i//2,i%2].set_title('b = {}, c = {}, n = {}'.format(b,c,n))
  ax[i//2,i%2].legend(loc = 'upper right')
plt.savefig('../plots/varya.png')
print('\n\n\n')

# variation in block size
print('==============================================================================')
print('Variation in Block Size')
print('==============================================================================')
sampler = [1, 2, 4, 8, 16, 32, 64, 128, 256]
fig, ax = plt.subplots(2,2,figsize = (15,15))
fig.suptitle('Variation in b')
for i, (a,c,n) in enumerate([(2,8,100),(4,8,100),(2,16,100),(4,16,100)]):
  ans_opt = []
  ans_unopt = []
  f = open('../csvs/variationb/{}_{}_{}.csv'.format(a,c,n), 'w')
  f.write('Opt,a,b,c,n,Total Fetches,Total Misses,Miss Rate,Compulsory miss, Conflict miss, Capacity miss\n')
  f.flush()
  for b in sampler:
    print("a = ",a,", b = ",b,", c = ",c,", n = ",n)
    bashCommand = form_command(a,b,c,n, optimised = True)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_opt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('Yes',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))

    bashCommand = form_command(a,b,c,n, optimised = False)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_unopt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('No',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))
  f.flush()
  f.close()
  ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='optimised')
  ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='unoptimised')
  ax[i//2,i%2].set_xlabel('Block Size')
  ax[i//2,i%2].set_ylabel('Demand Miss Rate')
  ax[i//2,i%2].set_title('a = {}, c = {}, n = {}'.format(a,c,n))
  ax[i//2,i%2].legend(loc = 'upper right')
plt.savefig('../plots/varyb.png')
print('\n\n\n')

# variation in cache size
print('=============================================================================
')
print('Variation in Cache Size')
print('=============================================================================
')
sampler = [1, 2, 4, 8, 16, 32, 64, 128, 256]
fig, ax = plt.subplots(2,2,figsize = (15,15))
fig.suptitle('Variation in c')
for i, (a,b,n) in enumerate([(2,64,100),(4,64,100),(2,32,100),(4,32,100)]):
  ans_opt = []
  ans_unopt = []
  f = open('../csvs/variationc/{}_{}_{}.csv'.format(a,b,n), 'w')
  f.write('Opt,a,b,c,n,Total Fetches,Total Misses,Miss Rate,Compulsory miss, Conflict miss, Capacity miss\n')
  f.flush()
  for c in sampler:
    print("a = ",a,", b = ",b,", c = ",c,", n = ",n)
    bashCommand = form_command(a,b,c,n, optimised = True)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_opt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('Yes',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))

    bashCommand = form_command(a,b,c,n, optimised = False)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_unopt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('No',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))
  f.flush()
  f.close()
  ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='optimised')
  ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='unoptimised')
  ax[i//2,i%2].set_xlabel('Cache Size')
  ax[i//2,i%2].set_ylabel('Demand Miss Rate')
  ax[i//2,i%2].set_title('a = {}, b = {}, n = {}'.format(a,b,n))
  ax[i//2,i%2].legend(loc = 'upper right')
plt.savefig('../plots/varyc.png')
print('\n\n\n')

# variation in matrix size
print('=============================================================================')
print('Variation in Matrix Size')
print('=============================================================================')
sampler = [4, 8, 16, 32, 64, 128]
fig, ax = plt.subplots(2,2,figsize = (15,15))
fig.suptitle('Variation in n')
for i, (a,b,c) in enumerate([(2,64,8),(4,64,16),(2,32,8),(4,32,64)]):
  ans_opt = []
  ans_unopt = []
  f = open('../csvs/variationn/{}_{}_{}.csv'.format(a,b,c), 'w')
  f.write('Opt,a,b,c,n,Total Fetches,Total Misses,Miss Rate,Compulsory miss, Conflict miss, Capacity miss\n')
  f.flush()
  for n in sampler:
    print("a = ",a,", b = ",b,", c = ",c,", n = ",n)
    bashCommand = form_command(a,b,c,n, optimised = True)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_opt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('Yes',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))

    bashCommand = form_command(a,b,c,n, optimised = False)
    output = subprocess.check_output(['bash','-c', bashCommand]).decode('utf8')
    ans_unopt.append(float(output.split("\n")[35].split()[3]))
    total_fetches = output.split("\n")[31].split()[2]
    total_misses = output.split("\n")[34].split()[2]
    miss_rate = str(float(total_misses)/float(total_fetches))
    miss_compolsory = output.split("\n")[36].split()[2]
    miss_capacity = output.split("\n")[37].split()[2]
    miss_conflict = output.split("\n")[38].split()[2]
    f.write('{},{},{},{},{},{},{},{},{},{},{}\n'.format('No',a,b,c,n,total_fetches,total_misses,miss_rate,miss_compolsory,miss_conflict,miss_capacity))
  f.flush()
  f.close()
  ax[i//2,i%2].plot(sampler, ans_opt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='optimised')
  ax[i//2,i%2].plot(sampler, ans_unopt, linestyle='dashed', linewidth = 2, marker='o', markersize=4,label='unoptimised')
  ax[i//2,i%2].set_xlabel('Matrix Size')
  ax[i//2,i%2].set_ylabel('Demand Miss Rate')
  ax[i//2,i%2].set_title('a = {}, b = {}, c = {}'.format(a,b,c))
  ax[i//2,i%2].legend(loc = 'upper right')
plt.savefig('../plots/varyn.png')
print('\n\n\n')
```