

```

# -*- coding: utf-8 -*-
"""
Created on Tue Mar  1 15:18:49 2016

@author: harora1
"""

import math

print ("-"*11)

a = 1
b = 2
c = 10
d = 20

manhattan = math.fabs(a - c) + math.fabs(b - d)
euclidean = (pow(pow(math.fabs(a - c), 2) +
                    pow(math.fabs(b - d), 2), 1/2))
minkowski = (pow(pow(math.fabs(a - c), 3) +
                    pow(math.fabs(b - d), 3), 1/3))

print ("a =", a, "b = ", b)
print ("c =", c, "d = ", d)
print ("Manhattan Distance =", round(manhattan,2))
print ("Euclidean Distance =", round(euclidean,2))
print ("Minkowski Distance =", round(minkowski,2))

print ("-"*11)

P = [1, 2]
Q = [10, 20]

manhattan = (math.fabs(P[0] - Q[0]) +
              math.fabs(P[1] - Q[1]))
euclidean = (pow(pow(math.fabs(P[0] - Q[0]), 2) +
                  pow(math.fabs(P[1] - Q[1]), 2), 1/2))

```

```

minkowski = (pow(pow(math.fabs(P[0] - Q[0]), 3) +
                  pow(math.fabs(P[1] - Q[1]), 3), 1/3))

print ("P[0] =", P[0], "P[1] = ", P[1])
print ("Q[0] =", Q[0], "Q[1] = ", Q[1])
print ("Manhattan Distance =", round(manhattan,2))
print ("Euclidean Distance =", round(euclidean,2))
print ("Minkowski Distance =", round(minkowski,2))

print ("-"*11)

P = [1, 2, 3]
Q = [10, 20, 30]

manhattan = (math.fabs(P[0] - Q[0]) +
              math.fabs(P[1] - Q[1]) +
              math.fabs(P[2] - Q[2]))
euclidean = pow(pow(math.fabs(P[0] - Q[0]), 2) +
                 pow(math.fabs(P[1] - Q[1]), 2) +
                 pow(math.fabs(P[2] - Q[2]), 2), 1/2)
minkowski = pow(pow(pow(math.fabs(P[0] - Q[0]), 3) +
                    pow(math.fabs(P[1] - Q[1]), 3) +
                    pow(math.fabs(P[2] - Q[2]), 3), 1/3))

print ("P =", P)
print ("Q =", Q)
print ("Manhattan Distance =", round(manhattan,2))
print ("Euclidean Distance =", round(euclidean,2))
print ("Minkowski Distance =", round(minkowski,2))

print ("-"*11)

P = [1, 2, 3]
Q = [10, 20, 30]
n = 3

sumpq = P[0]*Q[0] + P[1]*Q[1] + P[2]*Q[2]

```

```

sump = P[0] + P[1] + P[2]
sumq = Q[0] + Q[1] + Q[2]
sump2 = P[0]**2 + P[1]**2 + P[2]**2
sumq2 = Q[0]**2 + Q[1]**2 + Q[2]**2

nr = sumpq - (sump*sumq)/n
dr = (pow(sump2 - (sump**2)/n, 0.5) *
      pow(sumq2 - (sumq**2)/n, 0.5))

r = nr/dr

print ("P =", P)
print ("Q =", Q)
print ("Pearson Correlation =", round(r,2))

print ("-"*11)

```