Problem 1

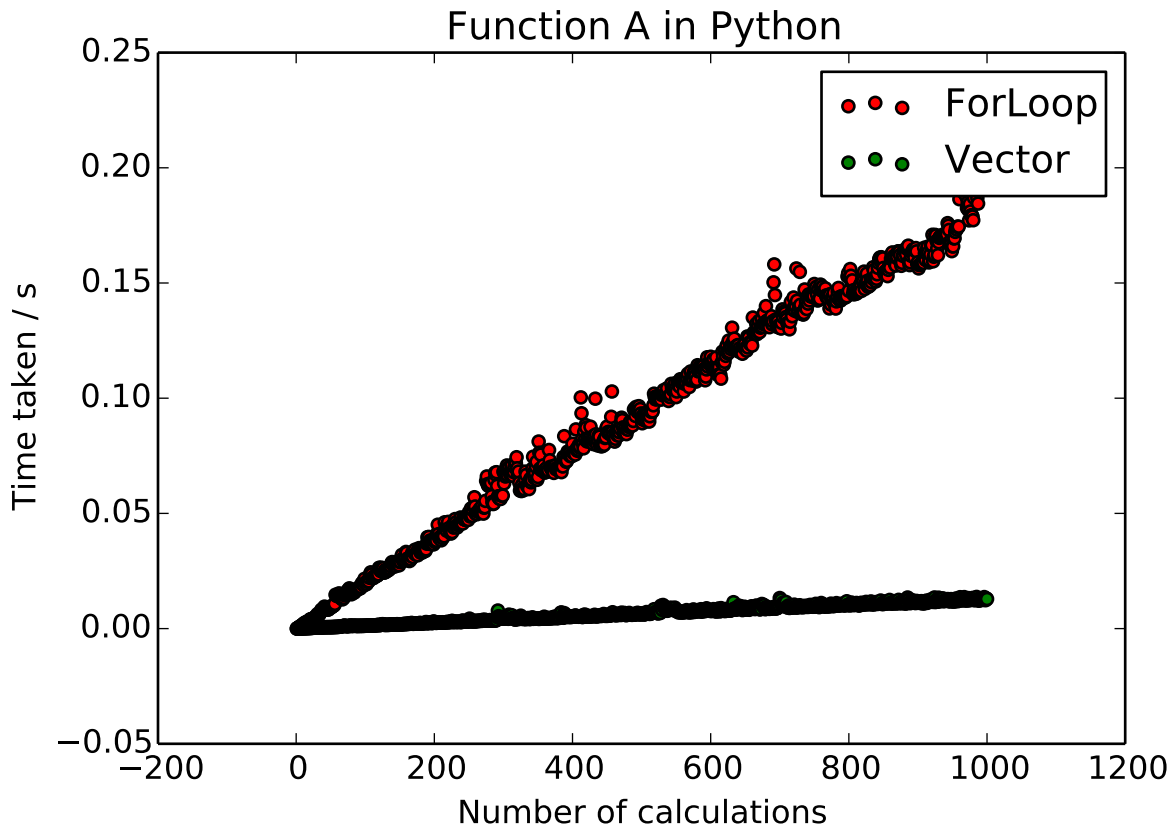
Percy Zhai & Zach Zheng

11/5/2019

Function A in Python: for and vec

```
import numpy as np
import timeit
import matplotlib.pyplot as plt

def Stu1FuncAFor(x):
  mx = np.zeros(10)
  nx = np.zeros(10)
  res = np.zeros(10)
  M = np.random.normal(size = [10,10])
  N = np.random.normal(size = [10,10])
  G = np.random.normal(size = 10)
  for i in range(10):
    sum = 0
    for j in range(10):
      sum += M[i,j]*x[j]
    mx[i] = sum
  for i in range(10):
    sum = 0
    for j in range(10):
      sum += N[i,j]*x[j]
    nx[i] = sum
  for i in range(10):
    res[i] = mx[i] + nx[i] + G[i]
  return res

def Stu1FuncAVec(x):
  M = np.random.normal(size = [10,10])
  N = np.random.normal(size = [10,10])
  G = np.random.normal(size = 10)
  res = M.dot(x)+N.dot(x)+G
  return res


x = np.zeros(10)
looprange = 1000

timefor = np.zeros(looprange)
for j in range(looprange):
  tic = timeit.default_timer()
  for i in range(j):
    res1for = Stu1FuncAFor(x)
  toc = timeit.default_timer()
  timefor[j] = toc-tic
plt.scatter(range(1,looprange+1),timefor,c="r", label="ForLoop")
```

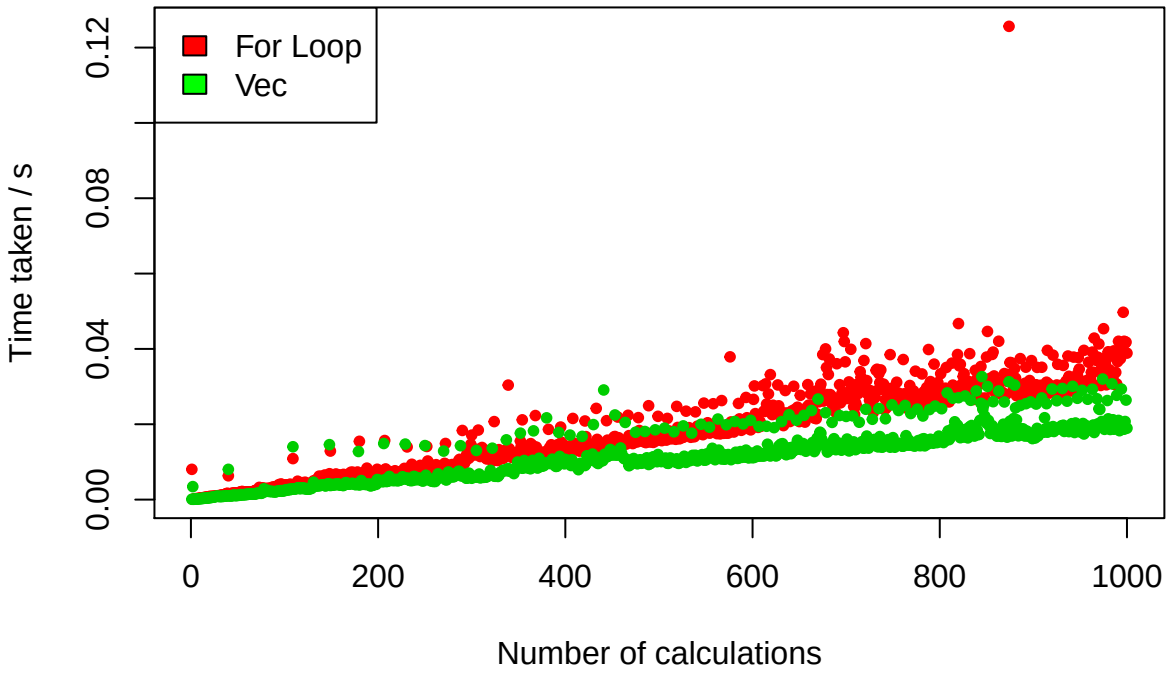
```
timevec = np.zeros(looprange)
for j in range(looprange):
  tic = timeit.default_timer()
  for i in range(j):
    res1vec = Stu1FuncAVec(x)
  toc = timeit.default_timer()
  timevec[j] = toc-tic
plt.scatter(range(1,looprange+1),timevec,c="g", label="Vector")
plt.legend()
plt.xlabel("Number of calculations")
plt.ylabel("Time taken / s")
plt.title("Function A in Python")
```



Function B in R: for and vec

```
Stu1FunBFor <- function(x){
  m3x <- matrix(NA, ncol = 1, nrow = 10)
  res <- matrix(NA, ncol = 1, nrow = 10)
  M <- matrix(rnorm(100), ncol = 10, nrow = 10)
  G <- matrix(rnorm(100), ncol = 1, nrow = 10)
  for(i in 1:10){
    sum <- 0
    for(j in 1:10){
      sum <- sum + M[i,j]^3*x[j]
    }
    m3x[i] <- sum
```

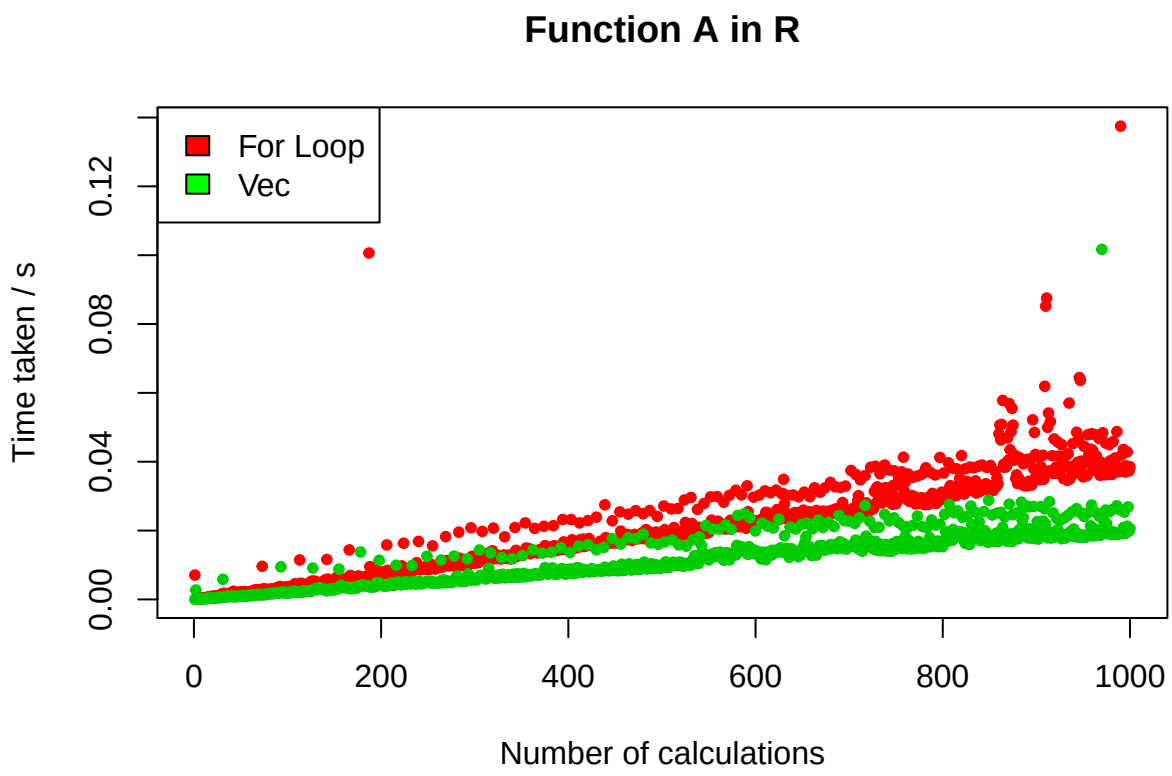
```
  }
  for(i in 1:10){
    res[i] = m3x[i] + 3*G[i]
  }
  return(res)
}

Stu1FunBVec <- function(x){
  M <- matrix(rnorm(100), ncol = 10, nrow = 10)
  G <- matrix(rnorm(100), ncol = 1, nrow = 10)
  res <- (M**3)%*%x + 3*G
  return(res)
}

x <- matrix(1, ncol = 1, nrow = 10)

timefor <- vector("numeric", 1000)
for(j in 1:1000){
  tic <- Sys.time()
  for(i in 1:j){
    resBfor <- Stu1FunBFor(x)
  }
  toc <- Sys.time()
  timefor[i] <- toc - tic
}


timevec <- vector("numeric", 1000)
for(j in 1:1000){
  tic <- Sys.time()
  for(i in 1:j){
  resBvec <- Stu1FunBVec(x)
  }
  toc <- Sys.time()
  timevec[i] <- toc - tic
}

plot(timefor,col = 2, main = "Function B in R",xlab = "Number of calculations",ylab = "Time taken / s",
points(timevec, col = 3, pch=20)
legend("topleft",c("For Loop","Vec"), fill = c("red","green"))
```



Function A in R: for and vec

```
X = matrix(rnorm(10),nrow=10)
Std2FuncA_for<-function(X){
  M<-matrix(rnorm(100),nrow=10)
  N<-matrix(rnorm(100),nrow=10)
  G<-matrix(rnorm(10),nrow=10)
  R<-matrix(rep(0,10),nrow=10)
  for (i in 1:10){
    for (j in 1:10){
      R[i] = R[i]+M[i,j]*X[j]+N[i,j]*X[j]
    }
    R[i] = R[i] + G[i]
  }
  return(R)
}

Std2FuncA_vec<-function(X){
  M<-matrix(rnorm(100),nrow=10)
  N<-matrix(rnorm(100),nrow=10)
  G<-matrix(rnorm(10),nrow=10)
  R<-M%*%X+N%*%X+G
  return(R)
}

timefor=rep(0,1000)
timevec=rep(0,1000)
for(i in 1:1000){
  tic1=Sys.time()
  for(j in 1:i){
```

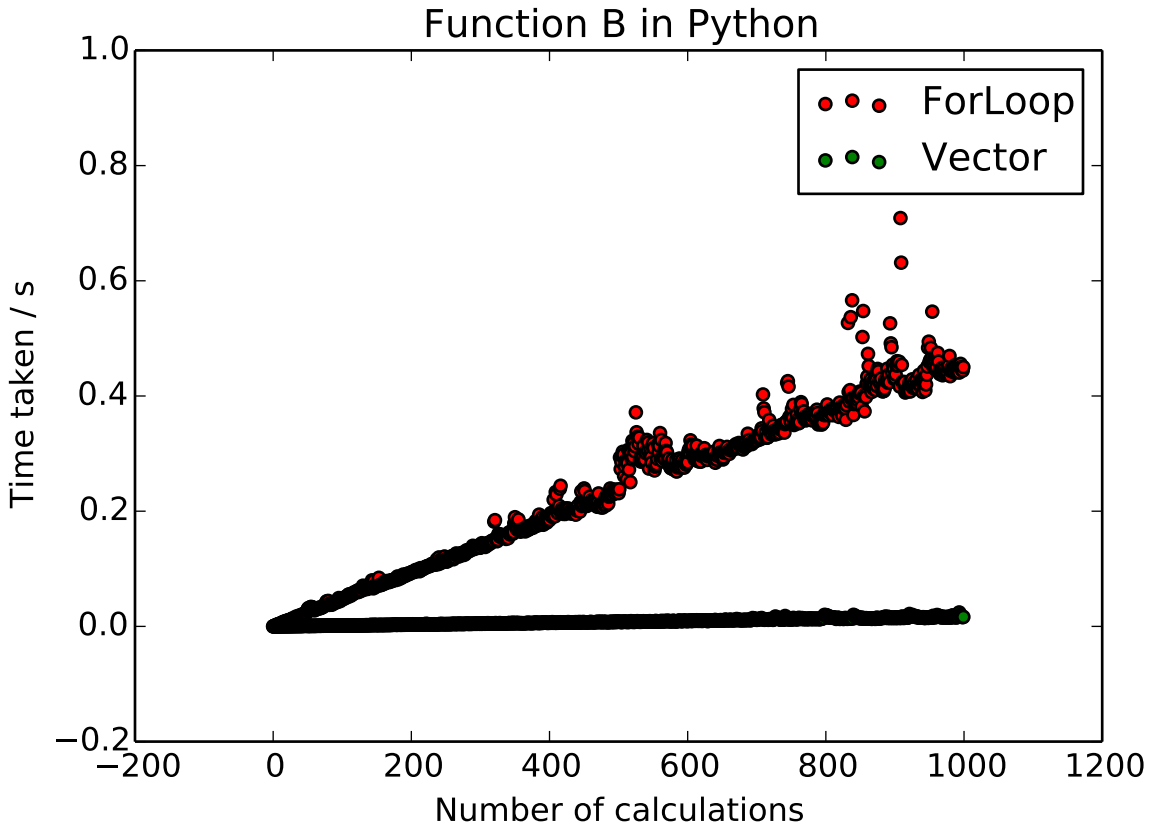
```
    a = Std2FuncA_for(X)
  }
  timefor[i] = Sys.time()-tic1
}

for(i in 1:1000){
  tic2=Sys.time()
  for (j in 1:i){
    b = Std2FuncA_vec(X)
  }
  timevec[i] = Sys.time()-tic2
}
plot(timefor,col = 2, main = "Function A in R",xlab = "Number of calculations",ylab = "Time taken / s",
points(timevec, col = 3, pch = 20)
legend("topleft",c("For Loop","Vec"), fill = c("red","green"))
```



Function B in python: for and vec

```
import timeit
import numpy as np
import matplotlib.pyplot as plt
X = np.random.normal(size = [10,1]);
def Std2FuncB_for(X):
  R = np.zeros(10);
  M_3 = np.zeros([10,10]);
  M = np.random.normal(size = [10,10]);
  G = np.random.normal(size = 10);
  for i in range(10):
    for j in range(10):
      M_3[i,j] = M[i,j]*M[i,j]*M[i,j];
```

```
  for i in range(10):
    for j in range(10):
      R[i] = R[i] + M_3[i,j] * X[j];
    R[i]=R[i]+3*G[i];
  return(R);

def Std2FuncB_vec(X):
  M = np.random.normal(size = [10,10]);
  G = np.random.normal(size = 10);
  M_3 = M**3;
  R = M_3.dot(X)+3*G;
  return(R);

timefor=np.zeros(1000);
timevec=np.zeros(1000);
X = np.random.normal(size = [10,1]);

for i in range(1000):
  tic=timeit.default_timer();
  for j in range(i):
    a = Std2FuncB_for(X);
  timefor[i] = timeit.default_timer()-tic;
plt.scatter(range(1000),timefor,c="r",label="ForLoop")

for i in range(1000):
  tic=timeit.default_timer();
  for j in range(i):
    b = Std2FuncB_vec(X);
  timevec[i] = timeit.default_timer()-tic;
plt.scatter(range(1000),timevec,c="g",label="Vector")
plt.legend()
plt.xlabel("Number of calculations")
plt.ylabel("Time taken / s")
plt.title("Function B in Python")
```


Function B in Python
ForLoop
Vector
Time taken / s
Number of calculations
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−200
0
200
400
600
800
1000
1200