Homework 2
Haixiao Dong

Question 1#

In this problem, 5%, 10%, 25%, 50% and 75% were set as the missing rate. The stochastic imputation method was used to impute the missing values. The accuracy was calculated as correlation coefficient and match proportion.

The number of replicates is 30, the accuracy values are shown in Fig 1, and the average and standard deviations (SD) are shown in table 1. With the increase of the missing rate, the imputation accuracy has no significant increase or decrease, but becomes more stable.

Table 1. Accuracy for stochastic imputation method with different missing rates

Missing Rate (%)	**correlation coefficient**		**match proportion**	
	average	**SD**	**average**	**SD**
5	0.3860769	0.004335285	0.6669100	0.002264766
10	0.386469	0.003562729	0.667233	0.001764067
25	0.3865514	0.002240052	0.6672618	0.001071501
50	0.3873513	0.0013526924	0.6676126	0.0006970124
75	0.3870073	0.0011092593	0.6674341	0.0005564907

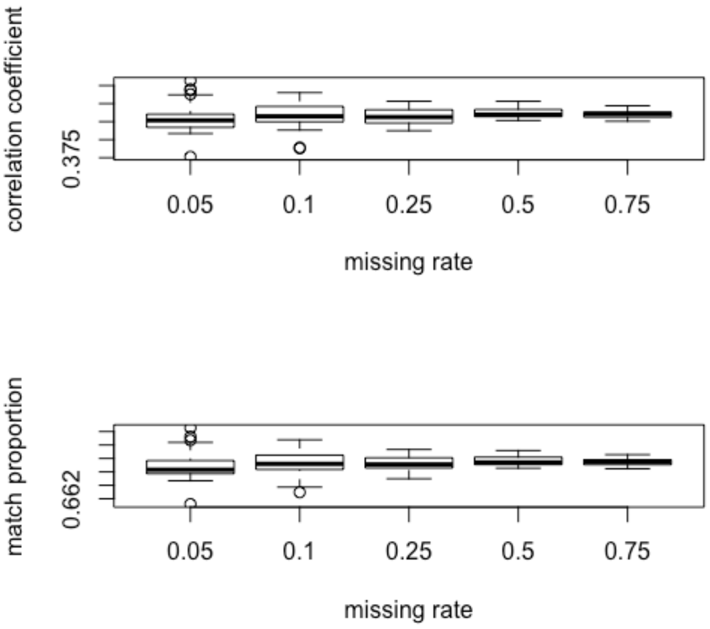

Fig 1.

Question 2#

In this problem, missing rate was fixed at 25%, and 75%, 50%, and 25% of individuals were randomly sampled to perform imputation with the stochastic imputation method. The imputation accuracy was calculated as correlation coefficient and match proportion.

The number of replicates is 30, the accuracy values are shown in Fig 2, and the average and SD are shown in table 2. With the increase of the sample size, the imputation accuracy has no significant increase or decrease, but becomes more stable.

Table 2. Accuracy for stochastic imputation method with different sample size

sample size (%)	**correlation coefficient**		**match proportion**	
	average	**SD**	**average**	**SD**
75	0.3873948	0.002589735	0.6678006	0.001852151
50	0.3865797	0.003511369	0.6672066	0.002414688
25	0.3863943	0.005610924	0.6669023	0.004010182

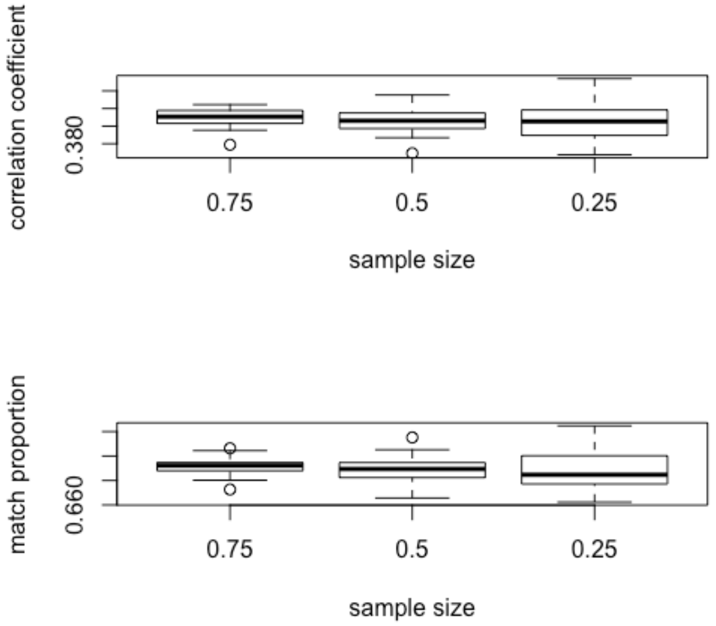

Fig 2.

Question 3#

In this problem, the missing rate was fixed at 25% and all the individuals were used to perform imputation with KNN method with K=2, 5, 10 and 20. The imputation accuracy was calculated as correlation coefficient and match proportion.

The number of replicates is 20, the accuracy values are shown in Fig 3, and the average and SD are shown in table 3. With the increase of K (the number of nearest neighbors), the correlation coefficient increases while the match proportion decreases.

Table 3. Accuracy for KNN method with different K

K	**correlation coefficient**		**match proportion**	
	average	**SD**	**average**	**SD**
2	0.6023541	0.002622131	0.5806388	0.003358265
5	0.6399470	0.001653259	0.4951294	0.005597869
10	0.6485342	0.001609084	0.4247312	0.006908058
20	0.651405	0.001766041	0.361426	0.005771558

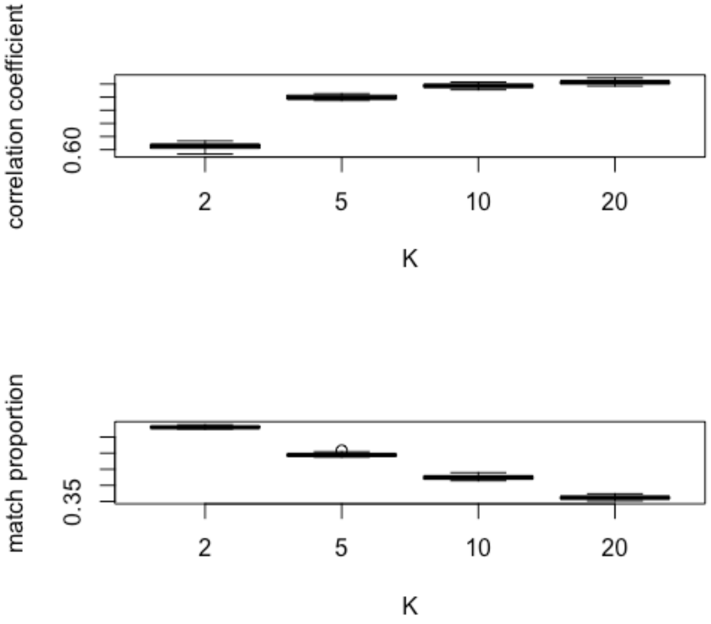

Fig 3.

Question 4#

In this problem, the missing rate was set at 25%, and all individuals were used to perform imputation with the stochastic method, KNN and BEAGLE.

The number of replicates is 10, the accuracy values are shown in Fig 4, and the average and SD are shown in table 4. According to the correlation coefficient, KNN works best, then BEAGLE and the stochastic method. According to the match proportion, Beagle works best, then the stochastic method and KNN.

Table 4. Comparison of the stochastic method, KNN and BEAGLE

Methods	**correlation coefficient**		**match proportion**	
	average	**SD**	**average**	**SD**
Stochastic	0.3874751	0.0014738399	0.6676402	0.0007237142
KNN	0.6483769	0.0015631163	0.4258270	0.0063753279
BEAGLE	0.6119752	0.0020275542	0.7554531	0.0009562062

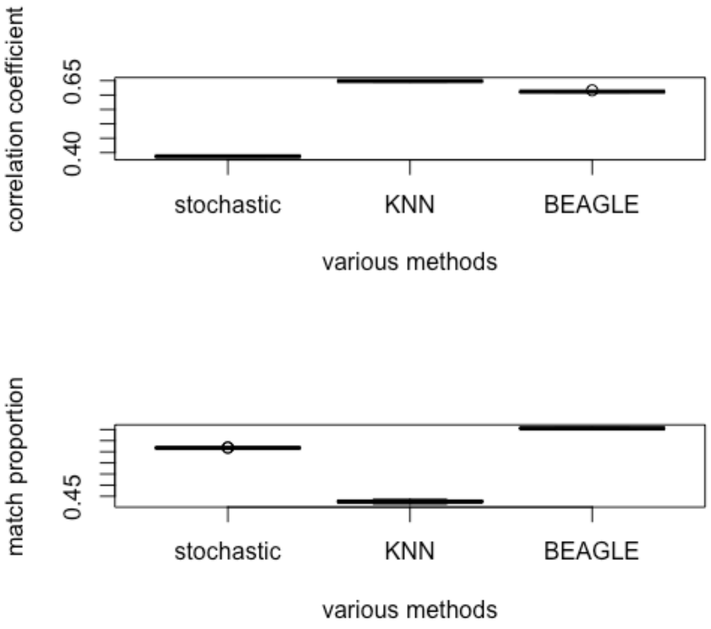

Fig 4.

Question 5#

In this problem, the missing rate was fixed at 25% and all the individuals were used to perform imputation with KNN method with K=2, 5, 10 and 20. The imputation accuracy was calculated as correlation coefficient and match proportion. Here I switched neighbors to genetic markers and attribute to individuals (In the code, stop transpose X before imputation).

The number of replicates is 20, the accuracy values are shown in Fig 5, and the average and SD are shown in table 5. With the increase of K (the number of nearest neighbors), the correlation coefficient increases while the match proportion decreases. Compared to question 3#, the coefficients are better, but the match proportions show more significant decrease with the increase of K.

Table 5. Accuracy for KNN method with different K (switching neighbors and attributes)

K	**correlation coefficient**		**match proportion**	
	average	**SD**	**average**	**SD**
2	0.6506265	0.001668839	0.6192237	0.001496099
5	0.6813710	0.001547220	0.4704089	0.002420381
10	0.6923917	0.001551711	0.3024489	0.002214593
20	0.6903060	0.001581919	0.1721516	0.002439105

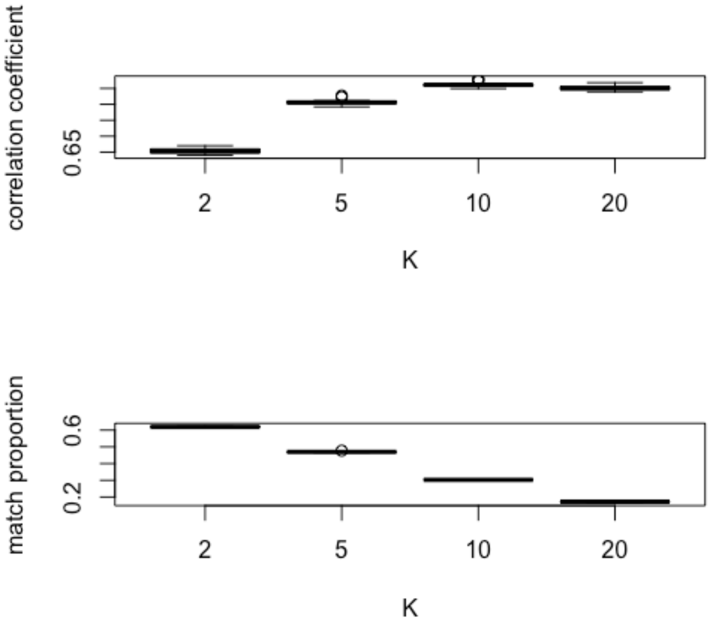

Fig 5.

Question 6#

In this problem, 5%, 10%, 25%, 50% and 75% were set as the missing rate. KNN method was used to perform imputation.

With the increase of missing rate from 5% to 25%, the correlation coefficient for all genotypes decreases, while the match proportion for all genotypes, major and minor allele homozygous increase. The match proportion for major allele homozygous is higher than that of all genotypes and minor allele homozygous, which indicate the KNN method performs better imputation for the major allele homozygous.

There's a steep drop-off when the missing rate reaches 50%. This may be due to "with more than 50% entries missing; mean imputation used for these rows" according to the warning messages,.

Note: when missing rate is 75%, all the imputation values are 0s. This explains why the correlation coefficient's average and SD are NAs and the average for major or minor allele homozygous are almost the same (~50%).

Table 6. Accuracy for KNN method with different missing rates

Missing Rate (%)	**correlation coefficient**		**match proportion**					
	average	**SD**	**average**	**SD**	**average (major homozygous)**	**SD (major homozygous)**	**average (minor homozygous)**	**SD (minor homozygous)**
5	0.659455923	0.0037165425	0.404233330	0.0035781746	0.536550898	0.0048037633	0.001389565	0.0005664475
10	0.65705850	0.0026849627	0.41054314	0.0045593008	0.54463531	0.0059716974	0.00143364	0.0003930109
25	0.648452752	0.0014389223	0.427045393	0.0081174081	0.565685970	0.0105372769	0.003780317	0.0009491814
50	0.420438893	0.005123985	0.248313634	0.006692337	0.328607599	0.008938082	0.005568222	0.001062401
75	NA	NA	0.4975298	0.0003102780	0.5203161	0.0003431685	0.5219858	0.0007377933

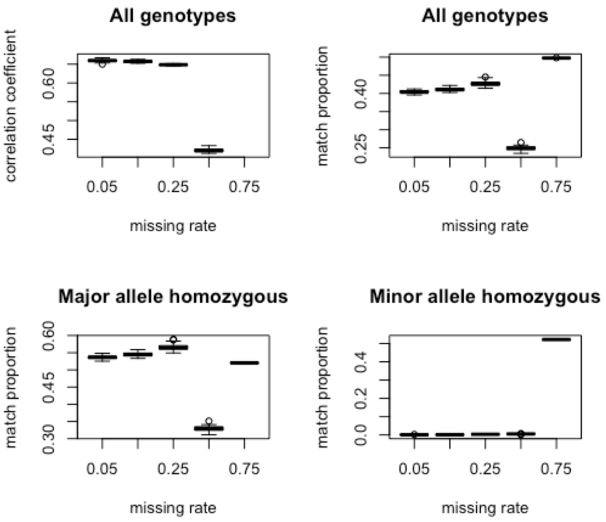

Fig 6.

Supplementary: Note for the KNN method

The impute.knn function contains set.seed() inside. As the set.seed is global, this will cause after the first replicate or loop, all other replicates or loops will generate the same X (genotype with simulated missing values) which will lead to always one identical result.

To solve this problem, in the question 3-6, I generate all the Xs' index before the replicates.

Wrong code:

```
for (i in 1:length(K)){
  myimp.knn <- replicate(nrep, {
    #missing value simulation
    X=X.raw
    index.m=FIndex.m(X=X, mr)
    X[index.m]=NA
    #imputation using KNN
    X.knn= impute.knn(as.matrix(t(X)), k=K[i])
… …
}
```

Correct code:

```
for (i in 1:length(K)){
  #missing value index simulation
  set.seed(99164)
  index.nrep <- replicate(nrep, {
    X=X.raw
    index.m=FIndex.m(X=X, mr)
  })
  myimp.knn <- lapply(1:nrep, function(r){
    #missing value simulation
    X=X.raw
    index.m=index.nrep[,,r]
    X[index.m]=NA
    #imputation using KNN
    X.knn= impute.knn(as.matrix(t(X)), k=K[i])
… …
}
```