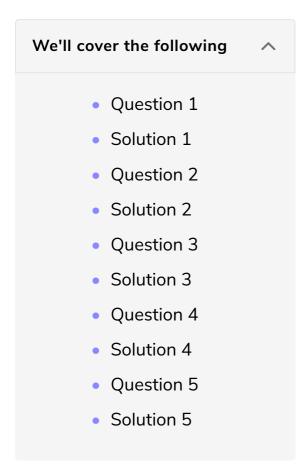
Exercises

Here are some challenges based on everything you've studied so far. Good luck!

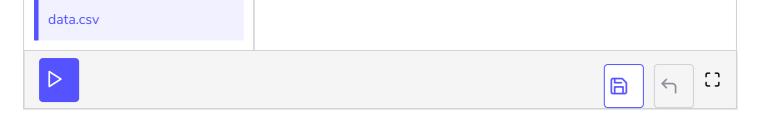


Here is a data file containing two columns of comma-separated data.

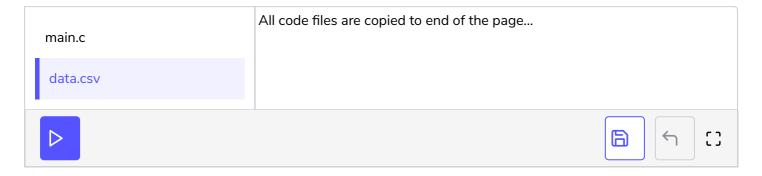
```
100,111
93,103
115,119
97,117
106,116
111,116
111,119
100,103
126,118
93,119
```

Question 1#

Write a program to read in the data file into one or more data structures, and print the values out to the screen. You can assume in your program that you know the number of rows of data (10).

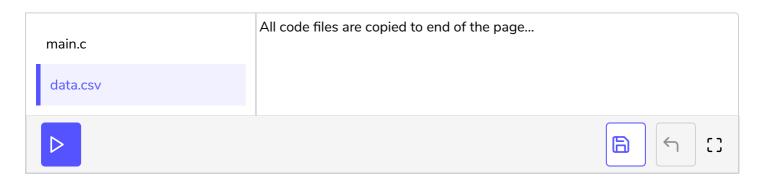


Solution 1#

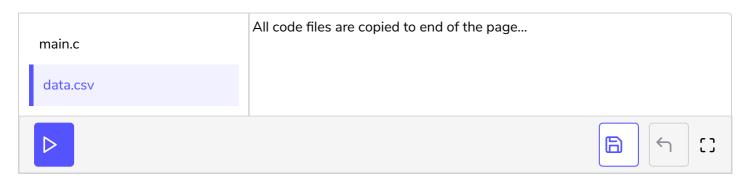


Question 2

Rewrite your program above assuming you don't know in advance how many rows of data you have.



Solution 2

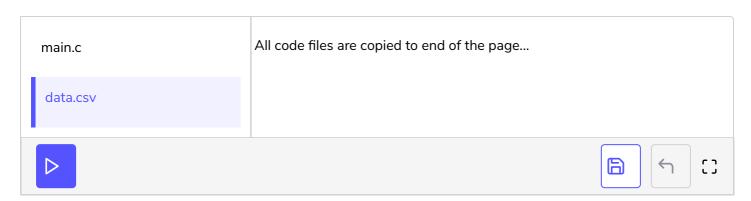


Question 3

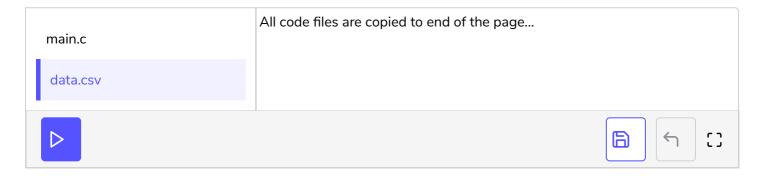
Add to your program a function that computes the value of a t statistic for the difference between means of the two columns of data. Assume it's an unpaired t-test and you can compute t using the following equation:

$$t = \frac{\bar{X}_2 - \bar{X}_1}{\sqrt{\frac{s_1^2}{n1} + \frac{s_2^2}{n2}}}$$

You are given the solution from Question 2.



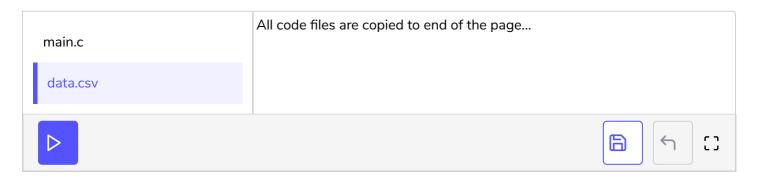
Solution 3



Question 4#

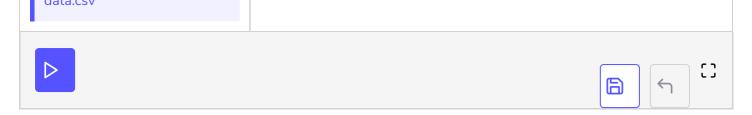
Implement a bootstrapping test of the t statistic you get above. Iterate <code>nboot</code> times, each time taking a random sample (with replacement) from the set of 20 observations, and assigning them to each group, then re-do the t-test. Count up how many times out of <code>nboot</code> you get a t value as large or larger as the one you computed above (so this is a one-tailed test). Set <code>nboot</code> to 1 million and report execution time. If you have a fast machine set <code>nboot</code> to 10 million so you have some dynamic range. If you have a slow machine set <code>nboot</code> to 1e5 (or 1e4 if it's really slow).

You will continue from Solution 3.



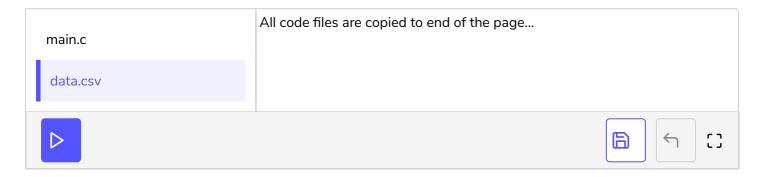
Solution 4



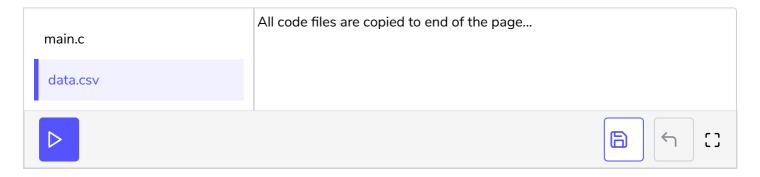


Question 5

Parallelize the bootstrap loop to make use of multiple CPU cores. Report execution time.



Solution 5



Code Files Content !!!


```
#include
int main() {
  // your code goes here
  return 0;
}
```

```
data.csv [1]
100,111
93,103
115,119
97,117
106,116
111,116
111,119
100,103
126,118
93,119
******************************
______
| main.c [2]
#include
int main(int argc, char *argv[])
{
 FILE *fid;
 int nr = 10;
 // open the file
 fid = fopen("data.csv", "r");
 printf("FILE OPENNEEED!!!");
 // load in the data
 double c1[nr], c2[nr];
 int i;
 for (i=0; i
int main() {
 // your code goes here
 return 0;
}
data.csv [3]
100,111
93,103
115,119
97,117
```

106,116 111,116

```
126,118
93,119
*********************************
| main.c [4]
#include
int main(int argc, char *argv[])
 FILE *fid;
 int nr = 0;
 char buf[255];
 // open the file
 fid = fopen("data.csv", "r");
 // count the number of lines
 while (!feof(fid)) {
   fgets(buf, 255, fid);
   nr++;
 }
 nr--;
 // go back to beginning of file
 fseek(fid, 0, SEEK_SET);
 // load in the data
 double c1[nr], c2[nr];
 int i;
 for (i=0; i
int main(int argc, char *argv[])
{
 FILE *fid;
 int nr = 0;
 char buf[255];
 // open the file
 fid = fopen("data.csv", "r");
 // count the number of lines
 while (!feof(fid)) {
   fgets(buf, 255, fid);
   nr++;
 }
 nr--;
 // go back to beginning of file
 fseek(fid, 0, SEEK_SET);
```

111,119 100,103

```
// load in the data
  double c1[nr], c2[nr];
  int i;
  for (i=0; i
#include
double mean(int n, double v[n]);
double sd(int n, double v[n]);
double ttest(int n1, int n2, double v1[n1], double v2[n2]);
int main(int argc, char *argv[])
{
  FILE *fid;
  int nr = 0;
  char buf[255];
  // open the file
  fid = fopen("data.csv", "r");
  // count the number of lines
  while (!feof(fid)) {
   fgets(buf, 255, fid);
   nr++;
  }
  nr--;
  // go back to beginning of file
  fseek(fid, 0, SEEK_SET);
  // load in the data
  double c1[nr], c2[nr];
  int i;
  for (i=0; i
#include
double mean(int n, double v[n]);
double sd(int n, double v[n]);
double ttest(int n1, int n2, double v1[n1], double v2[n2]);
int main(int argc, char *argv[])
  FILE *fid;
  int nr = 0;
  char buf[255];
  // open the file
  fid = fopen("data.csv", "r");
  // count the number of lines
  while (!feof(fid)) {
    fgets(buf, 255, fid);
   nr++;
  }
  nr--;
  // go back to beginning of file
  fseek(fid, 0, SEEK_SET);
  // load in the data
  double c1[nr], c2[nr];
  int i;
  for (i=0; i
```

```
#include
#include
#include
double mean(int n, double v[n]);
double sd(int n, double v[n]);
double ttest(int n1, int n2, double v1[n1], double v2[n2]);
int main(int argc, char *argv[])
  FILE *fid;
  int nr = 0;
  char buf[255];
  // open the file
  fid = fopen("data.csv", "r");
  // count the number of lines
  while (!feof(fid)) {
   fgets(buf, 255, fid);
   nr++;
  }
  nr--;
  // go back to beginning of file
  fseek(fid, 0, SEEK_SET);
  // load in the data
  double c1[nr], c2[nr];
  int i;
  for (i=0; i= tobs) tcount++;
  }
  printf("%d/%d, p=%.5f\n", tcount, nboot, (double)tcount/nboot);
  return 0;
}
// helper functions
double mean(int n, double v[n])
  double sum = 0.0;
  int i;
  for (i=0; i
#include
#include
#include
double mean(int n, double v[n]);
double sd(int n, double v[n]);
double ttest(int n1, int n2, double v1[n1], double v2[n2]);
int main(int argc, char *argv[])
  FILE *fid;
  int nr = 0;
  char buf[255];
  // open the file
  fid = fopen("data.csv", "r");
```

```
// count the number of lines
  while (!feof(fid)) {
   fgets(buf, 255, fid);
    nr++;
  }
  nr--;
  // go back to beginning of file
  fseek(fid, 0, SEEK_SET);
  // load in the data
  double c1[nr], c2[nr];
  int i;
  for (i=0; i= tobs) tcount++;
  printf("%d/%d, p=%.5f\n", tcount, nboot, (double)tcount/nboot);
  return 0;
}
// helper functions
double mean(int n, double v[n])
{
  double sum = 0.0;
  int i;
  for (i=0; i
#include
#include
#include
#include
double mean(int n, double v[n]);
double sd(int n, double v[n]);
double ttest(int n1, int n2, double v1[n1], double v2[n2]);
int main(int argc, char *argv[])
  FILE *fid;
  int nr = 0;
  char buf[255];
  // open the file
  fid = fopen("data.csv", "r");
  // count the number of lines
  while (!feof(fid)) {
    fgets(buf, 255, fid);
   nr++;
  }
  nr--;
  // go back to beginning of file
  fseek(fid, 0, SEEK_SET);
  // load in the data
  double c1[nr], c2[nr];
  int i;
  for (i=0: i= tobs) tcount++:
```

```
printf("%d/%d, p=%.5f\n", tcount, nboot, (double)tcount/nboot);

return 0;
}

// helper functions

double mean(int n, double v[n])
{
    double sum = 0.0;
    int i;
    for (i=0; i
```