# Code Documentation

## Stats Module

### stats.h

Header file defining functions for statistical calculations.

```c

# ifndef STATS\_H

# define STATS\_H

double mean(double \*data, int size);  
double median(double \*data, int size);  
double std\_dev(double \*data, int size, double mean);

# endif

```

### stats.c

Implementations of statistical functions.

```c

# include "stats.h"

# include "helpers.h"

# include

double mean(double \*data, int size) {  
 double sum = 0;  
 for (int i = 0; i &lt; size; i++) {  
 sum += data[i];  
 }  
 return sum / size;  
}

double median(double \*data, int size) {  
 sort(data, size); // Assuming sort is implemented in helpers.c  
 if (size % 2 == 0) {  
 return (data[size / 2 - 1] + data[size / 2]) / 2;  
 } else {  
 return data[size / 2];  
 }  
}

double stddev(double \*data, int size, double mean) {  
 double sumsqdiff = 0;  
 for (int i = 0; i &lt; size; i++) {  
 sumsqdiff += pow(data[i] - mean, 2);  
 }  
 return sqrt(sumsq\_diff / size);  
}  
```

## Helpers Module

### helpers.h

Header file with helper functions.

```c

# ifndef HELPERS\_H

# define HELPERS\_H

void sort(double \*data, int size);

# endif

```

### helpers.c

Helper functions implementation.

```c

# include "helpers.h"

// Simple insertion sort for simplicity  
void sort(double \*data, int size) {  
 for (int i = 1; i &lt; size; i++) {  
 double key = data[i];  
 int j = i - 1;  
 while (j &gt;= 0 &amp;&amp; data[j] &gt; key) {  
 data[j + 1] = data[j];  
 j = j - 1;  
 }  
 data[j + 1] = key;  
 }  
}  
```

## Main Module

### main.c

Main application file to demonstrate the usage of statistical functions.

```c

# include "stats.h"

# include

int main() {  
 double data[] = {12.0, 15.5, 18.7, 11.3, 14.2};  
 int size = sizeof(data) / sizeof(data[0]);  
 double m = mean(data, size);  
 double md = median(data, size);  
 double sd = std\_dev(data, size, m);

}  
```