Rebecca Aviles Barahona
Professor Goodney
CSCI 114
Student Lifestyle and the Predictor Analyses Between GPA and Stress

The dataset I found interesting was the "Student Lifestyle Dataset" on Kaggle. It has survey data that allows the participants to allocate time spent her day in hours as well as stress and their GPA averages. Being in college, it can be difficult to perform in a high stress environment when involved in many extracurriculars. Being a student, I have always wondered if succeeding in college is just about having excellent time management or if students suffer at the expense of their grades. Sharing my own personal experience, it is quite difficult to be active on campus while still leaving room in my schedule to sleep, eat healthy, go the gym, and study all in a span of 12 hours. Because I have always wondered how disciplined you must be to overwork yourself and do well in college, the goal of this project is to provide insight into student management as well as potential deficiencies students are compromising if any, to do well. Based on factors including physical activity, extracurriculars and study hours per day, if a student is maximizing the number of outside hours per day and they are stressed does this negatively impact their GPA? Additionally, predictor models are being done for both stress and GPA as individual, factors. If extracurriculars outside of classroom learning are significantly high, which of the variables within the dataset are affecting GPA and stress levels independently? If students compromise sleep, then GPA will be impacted. If we study more throughout the day, then GPA's will increase.

Using a linear regression technique, by parsing out the data and quantifying the different factors, the model can help determine if there is any correlational value and if any of the factor's effect on one another. Likewise, the data could support my hypothesis that if there are more extracurriculars that are being done in a day then that alone could affect GPA or impact stress negatively.

Figure 1: Factors with Respect to GPA

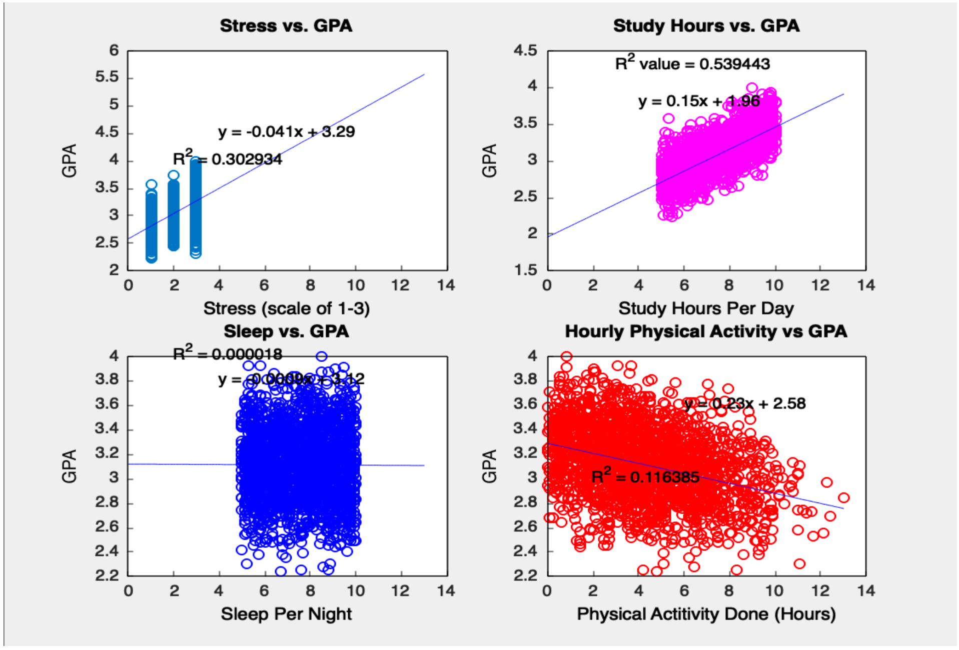


Figure 1.1 Using GPA as a beta predictor and revolving the main key components: Study Houts, Stress Levels (which we quantify to a scale of 1-3), Physical Activity per day, and Sleep Hours.

In the first figure above, a linear regression model was done using GPA as a dependent source to assess the plots. In the graphs above, it shows a positive correlation between Study Hours and GPA. This makes practical sense as the more you study throughout the day and week, grades will increase, and GPA would be high. This has a correlational strength of about 0.539. In this case this shows a moderate, positive association.

Stress with respect to GPA was also higher. With a positive linear regression line, the more stress you are the higher your GPA becomes. This has its limitations of course because the survey does not quantify the stress level, hard coding a baseline could provide a non-comprehensive or accurate representation of stress. By assessing a positive correlational relationship, the strength of this graph is about 0.30. Physical activity can also be seen than those who do more physical activity have lower GPA's. Originally the expectation within the hypothesis showed that those that incorporate daily activity regularly would have a decreased mental fog; however, this shows a negative association with a weaker strength of about 0.12.

Figure 2 Factors with Respect to Stress

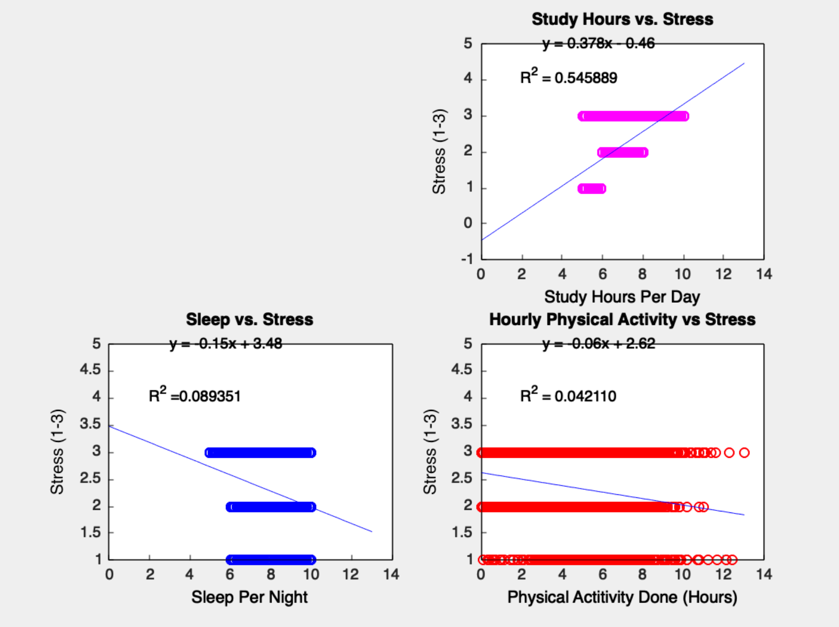
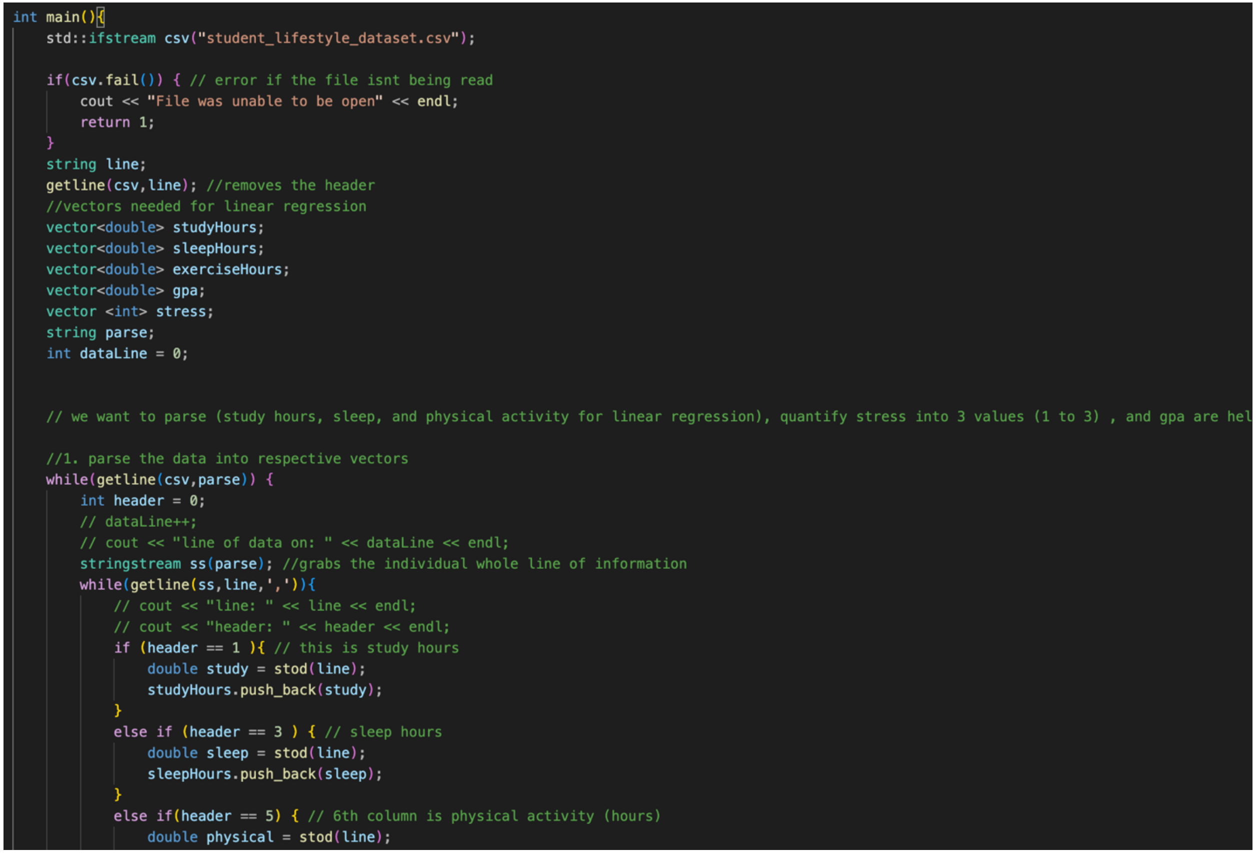


Figure 2.1 In this figure, Stress is the main predictor in which we compare Study Hours, Sleep per night, and the daily physical activity is being compared

Stress being compared in the second figure showing a stronger association with study hours and stress. With a strength of about 0.549 and a positive association; however, the trend is harder to ascertain. Encompassing a larger range of hours, those who study 4-11 hours a day have the highest stress level. In contrast, those who have lower study hours tend to have lower stress. In the next graph with regards to sleep hours, it seems that regardless of sleep, stress does not have an association with it. This is a similar trend when looking at physical activity. In fact those who work out more can equally have the same amount of stress.

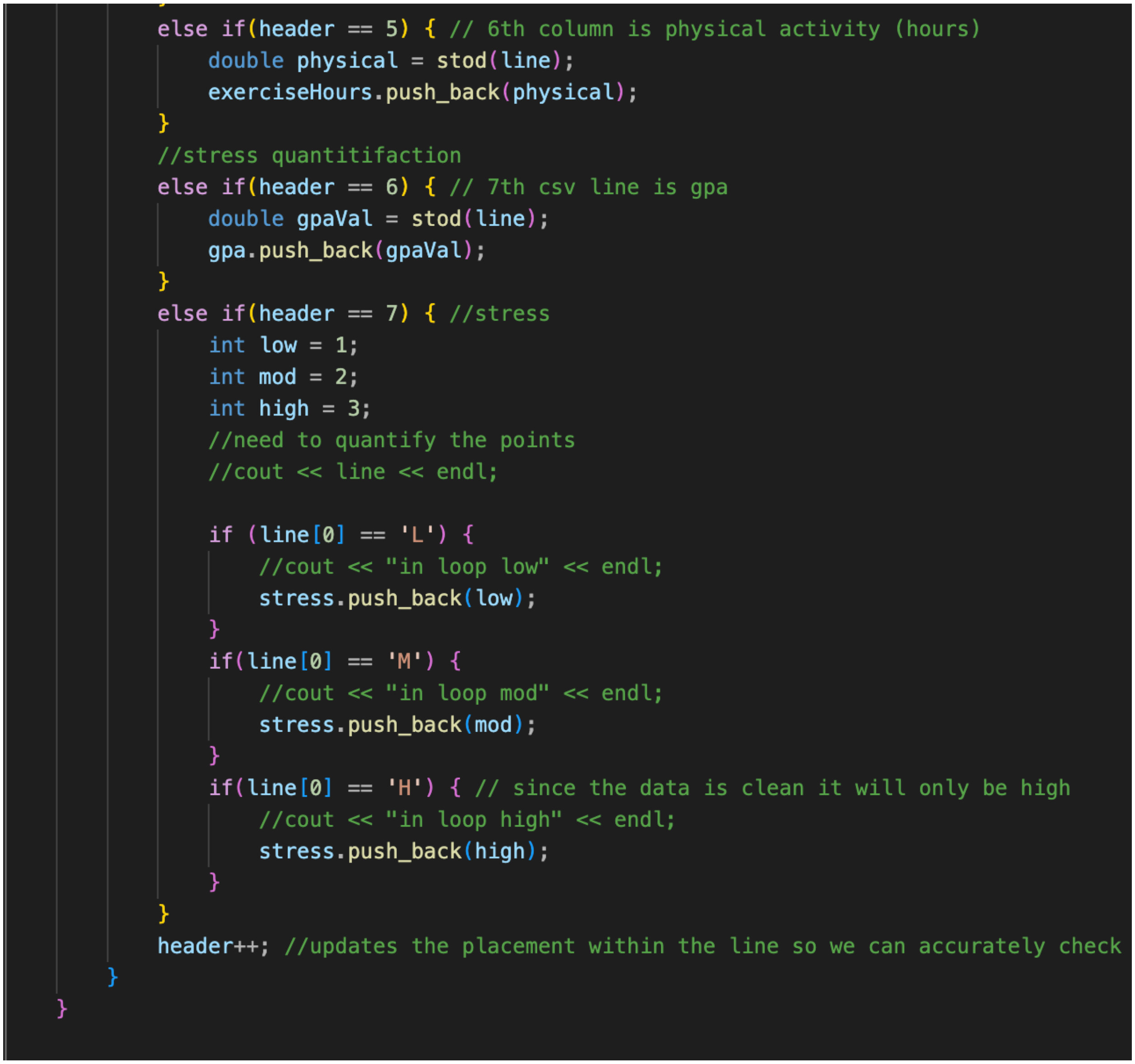
Based on the predictors with respect to GPA and Stress, it seems to show that GPA plays a bigger role in providing holistic information regarding lifestyle as a college student. Assuming the limitations of the stress, it is possible it was not an accurate representation of stress through the basis of the survey not being quantified. In the GPA and other factors, these factors were rated on a scale; however, stress was not, so it needed to be quantified based on the premise to use it as a predictor. The strongest associations between the two variables were study hours. Those who studied for longer period of times were typically those with higher GPA's but also those with high stress. In this case, looking at students with high stress and those who spend the most time in the library studying they have the better GPA's. Another association that is worth noting is stress and GPA, this association is also seen in when we just look at GPA only. In this case, the most we can attribute is only to these factors because the rest do not have the best associations where we do not see the strongest associations.

Line of Code Reasoning:



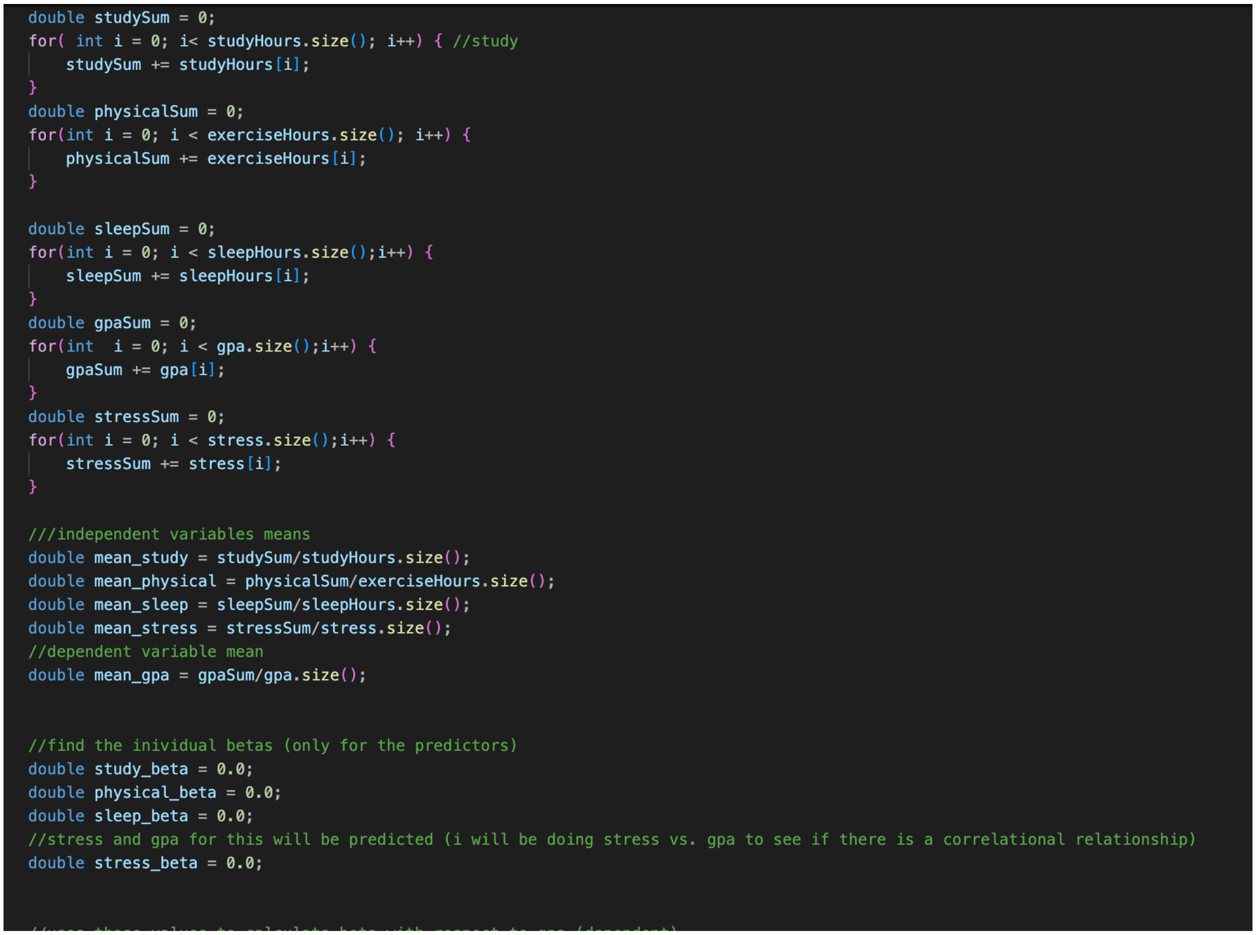
```
int main(){
    std::ifstream csv("student_lifestyle_dataset.csv");

    if(csv.fail()) { // error if the file isnt being read
        cout << "File was unable to be open" << endl;
        return 1;
    }
    string line;
    getline(csv,line); //removes the header
    //vectors needed for linear regression
    vector<double> studyHours;
    vector<double> sleepHours;
    vector<double> exerciseHours;
    vector<double> gpa;
    vector <int> stress;
    string parse;
    int dataLine = 0;


    // we want to parse (study hours, sleep, and physical activity for linear regression), quantify stress into 3 values (1 to 3) , and gpa are hel

    //1. parse the data into respective vectors
    while(getline(csv,parse)) {
        int header = 0;
        // dataLine++;
        // cout << "line of data on: " << dataLine << endl;
        stringstream ss(parse); //grabs the individual whole line of information
        while(getline(ss,line,',')){
            // cout << "line: " << line << endl;
            // cout << "header: " << header << endl;
            if (header == 1 ){ // this is study hours
                double study = stod(line);
                studyHours.push_back(study);
            }
            else if (header == 3 ) { // sleep hours
                double sleep = stod(line);
                sleepHours.push_back(sleep);
            }
            else if(header == 5) { // 6th column is physical activity (hours)
                double physical = stod(line);
```

In this line of code this is just the initialization step. We load in the csv file, and we also remove the header fille in order to parse out the data. The objective of this first part of the code is to parse out the data. We get the line and parse it into a stringstream so we can use it like an array. By doing this, we can make a more comprehensive code and push back into their respective vector.



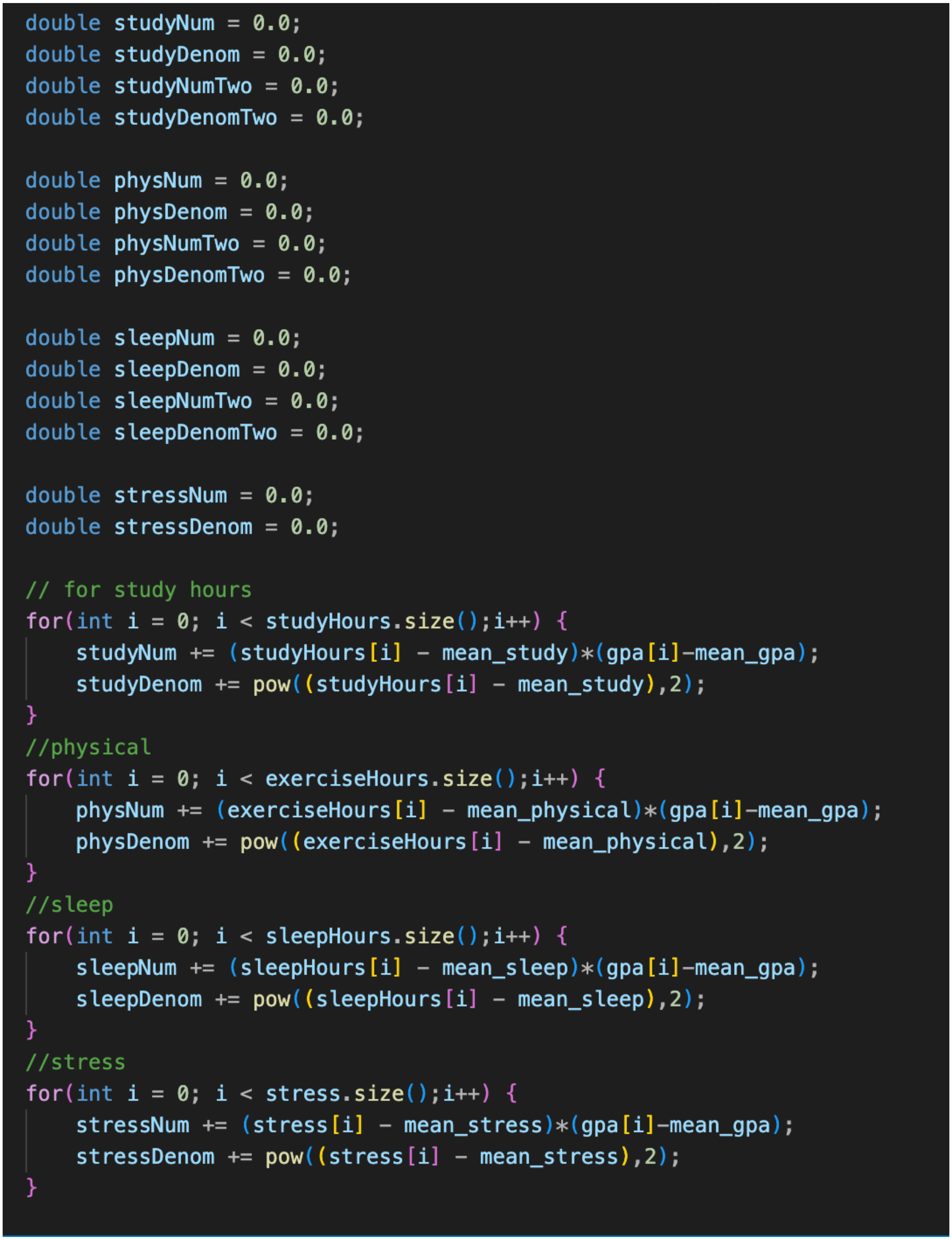
```
            else if(header == 5) { // 6th column is physical activity (hours)
                double physical = stod(line);
                exerciseHours.push_back(physical);
            }
            //stress quantitifaction
            else if(header == 6) { // 7th csv line is gpa
                double gpaVal = stod(line);
                gpa.push_back(gpaVal);
            }
            else if(header == 7) { //stress
                int low = 1;
                int mod = 2;
                int high = 3;
                //need to quantify the points
                //cout << line << endl;

                if (line[0] == 'L') {
                    //cout << "in loop low" << endl;
                    stress.push_back(low);
                }
                if(line[0] == 'M') {
                    //cout << "in loop mod" << endl;
                    stress.push_back(mod);
                }
                if(line[0] == 'H') { // since the data is clean it will only be high
                    //cout << "in loop high" << endl;
                    stress.push_back(high);
                }
            }
            header++; //updates the placement within the line so we can accurately check
        }
    }
```

In this part, we are still parsing and if we get the data for stress. Since the data is strings we can use the first part of the string to then based if the stress is Low, Moderate, or High then we will push it back otherwise.



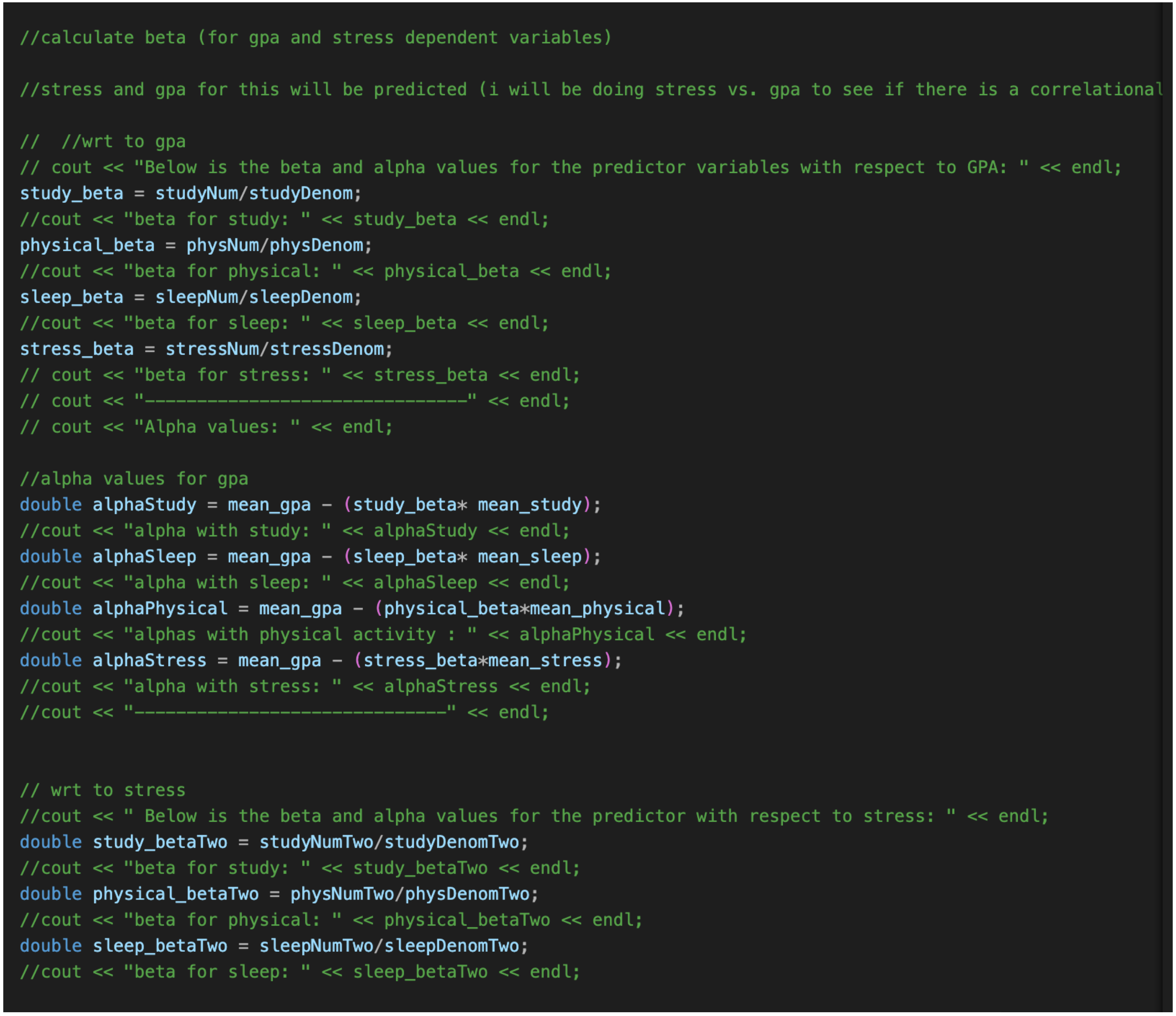
```
double studySum = 0;
for( int i = 0; i< studyHours.size(); i++) { //study
    studySum += studyHours[i];
}
double physicalSum = 0;
for(int i = 0; i < exerciseHours.size(); i++) {
    physicalSum += exerciseHours[i];
}

double sleepSum = 0;
for(int i = 0; i < sleepHours.size();i++) {
    sleepSum += sleepHours[i];
}
double gpaSum = 0;
for(int  i = 0; i < gpa.size();i++) {
    gpaSum += gpa[i];
}
double stressSum = 0;
for(int i = 0; i < stress.size();i++) {
    stressSum += stress[i];
}

///independent variables means
double mean_study = studySum/studyHours.size();
double mean_physical = physicalSum/exerciseHours.size();
double mean_sleep = sleepSum/sleepHours.size();
double mean_stress = stressSum/stress.size();
//dependent variable mean
double mean_gpa = gpaSum/gpa.size();


//find the inividual betas (only for the predictors)
double study_beta = 0.0;
double physical_beta = 0.0;
double sleep_beta = 0.0;
//stress and gpa for this will be predicted (i will be doing stress vs. gpa to see if there is a correlational relationship)
double stress_beta = 0.0;


//uses these values to calculate beta with respect to gpa (dependent)
```

In the next part of the code, I have the parsed-out data in their respective vectors. In this case we only did study hours, stress, gpa, exercise, and sleep. In this next part of the code, we are now going to make out linear regression plots by first finding the mean for each vector.



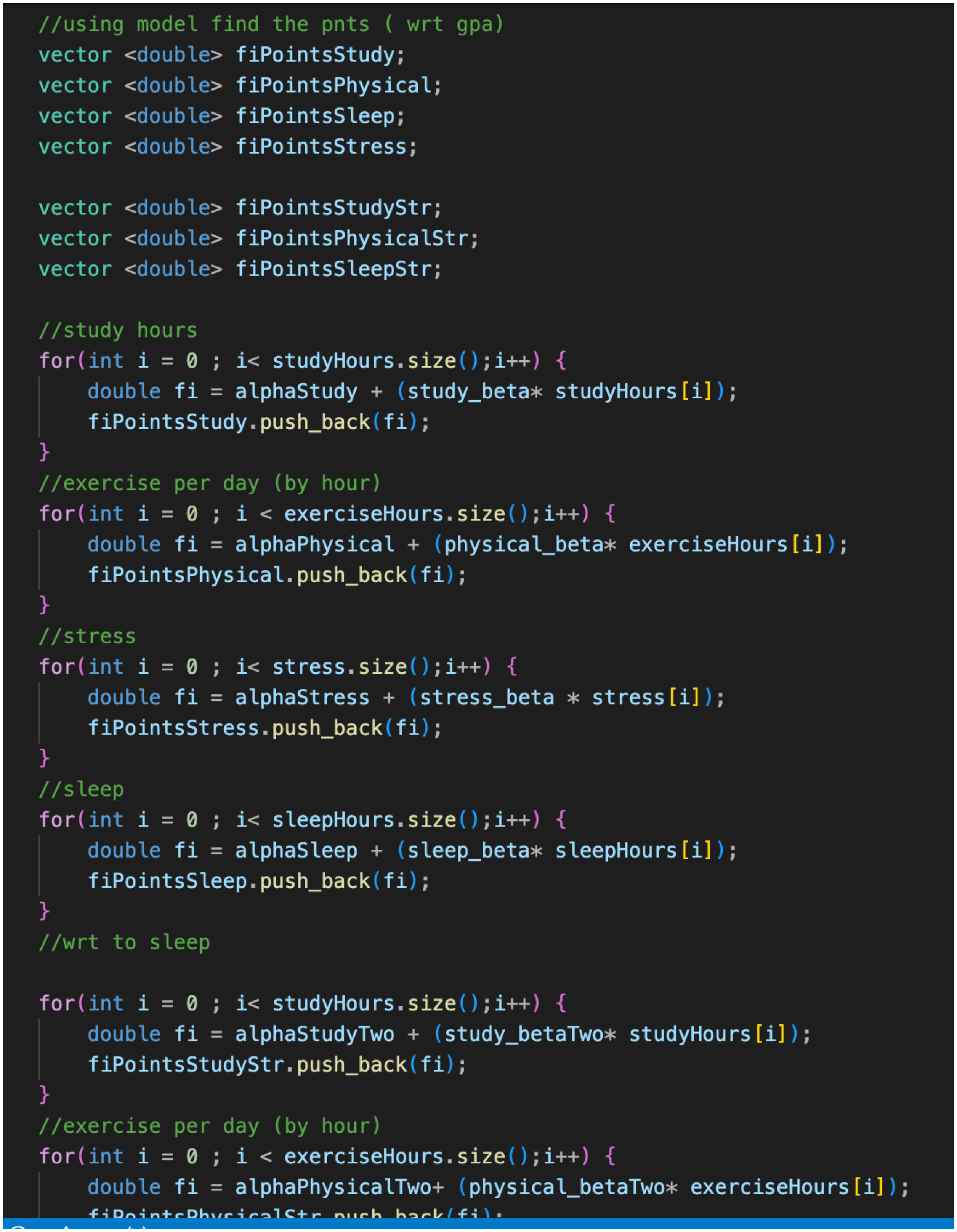
```
double studyNum = 0.0;
double studyDenom = 0.0;
double studyNumTwo = 0.0;
double studyDenomTwo = 0.0;

double physNum = 0.0;
double physDenom = 0.0;
double physNumTwo = 0.0;
double physDenomTwo = 0.0;

double sleepNum = 0.0;
double sleepDenom = 0.0;
double sleepNumTwo = 0.0;
double sleepDenomTwo = 0.0;

double stressNum = 0.0;
double stressDenom = 0.0;

// for study hours
for(int i = 0; i < studyHours.size();i++) {
    studyNum += (studyHours[i] - mean_study)*(gpa[i]-mean_gpa);
    studyDenom += pow((studyHours[i] - mean_study),2);
}
//physical
for(int i = 0; i < exerciseHours.size();i++) {
    physNum += (exerciseHours[i] - mean_physical)*(gpa[i]-mean_gpa);
    physDenom += pow((exerciseHours[i] - mean_physical),2);
}
//sleep
for(int i = 0; i < sleepHours.size();i++) {
    sleepNum += (sleepHours[i] - mean_sleep)*(gpa[i]-mean_gpa);
    sleepDenom += pow((sleepHours[i] - mean_sleep),2);
}
//stress
for(int i = 0; i < stress.size();i++) {
    stressNum += (stress[i] - mean_stress)*(gpa[i]-mean_gpa);
    stressDenom += pow((stress[i] - mean_stress),2);
}
```

Next, we are trying to find with respect to gpa being the independent variable. In order to do this, we go through a for in the whole vector’s length. In this case we have two variables one for the numerator and denominator and this is done so we can help make the beta without accidentally messing it up. In this we individually subtract the by the mean we found and multiplied by the dependent variable by the person's gpa and subtract by the mean gpa. The denominator squares the x value's variable by its own mean.



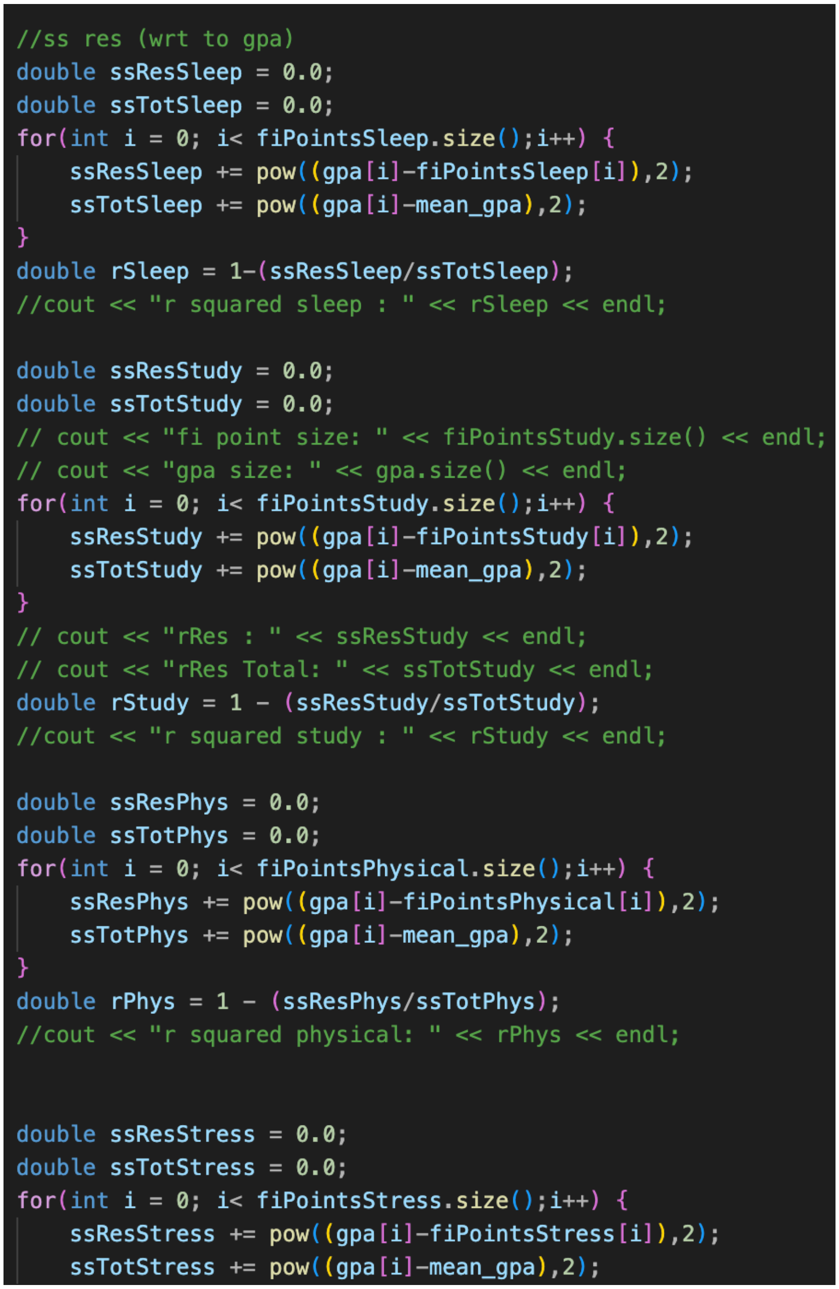
```
//calculate beta (for gpa and stress dependent variables)

//stress and gpa for this will be predicted (i will be doing stress vs. gpa to see if there is a correlational

//  //wrt to gpa
// cout << "Below is the beta and alpha values for the predictor variables with respect to GPA: " << endl;
study_beta = studyNum/studyDenom;
//cout << "beta for study: " << study_beta << endl;
physical_beta = physNum/physDenom;
//cout << "beta for physical: " << physical_beta << endl;
sleep_beta = sleepNum/sleepDenom;
//cout << "beta for sleep: " << sleep_beta << endl;
stress_beta = stressNum/stressDenom;
// cout << "beta for stress: " << stress_beta << endl;
// cout << "------------------------------" << endl;
// cout << "Alpha values: " << endl;

//alpha values for gpa
double alphaStudy = mean_gpa - (study_beta* mean_study);
//cout << "alpha with study: " << alphaStudy << endl;
double alphaSleep = mean_gpa - (sleep_beta* mean_sleep);
//cout << "alpha with sleep: " << alphaSleep << endl;
double alphaPhysical = mean_gpa - (physical_beta*mean_physical);
//cout << "alphas with physical activity : " << alphaPhysical << endl;
double alphaStress = mean_gpa - (stress_beta*mean_stress);
//cout << "alpha with stress: " << alphaStress << endl;
//cout << "-----------------------------" << endl;


// wrt to stress
//cout << " Below is the beta and alpha values for the predictor with respect to stress: " << endl;
double study_betaTwo = studyNumTwo/studyDenomTwo;
//cout << "beta for study: " << study_betaTwo << endl;
double physical_betaTwo = physNumTwo/physDenomTwo;
//cout << "beta for physical: " << physical_betaTwo << endl;
double sleep_betaTwo = sleepNumTwo/sleepDenomTwo;
//cout << "beta for sleep: " << sleep_betaTwo << endl;
```

This is me checking to see if the values are good and make sense as well as formulating the beta with the numerator and denominator. Each beta value is done with either stress or gpa as those the values we are trying to help predict for. We also find the alpha so this is done with the respect of the gpa and similarly to a line equation we multiple the mean by the beta for both predictors

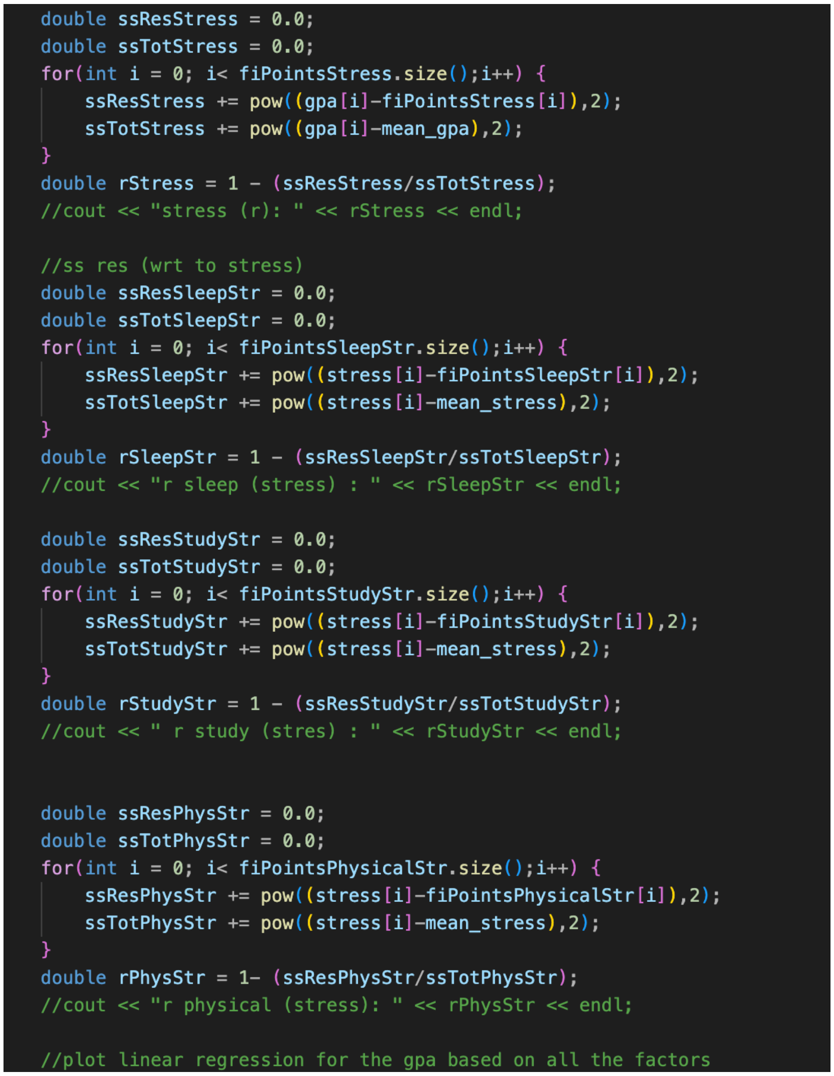


```
//using model find the pnts ( wrt gpa)
vector <double> fiPointsStudy;
vector <double> fiPointsPhysical;
vector <double> fiPointsSleep;
vector <double> fiPointsStress;

vector <double> fiPointsStudyStr;
vector <double> fiPointsPhysicalStr;
vector <double> fiPointsSleepStr;

//study hours
for(int i = 0 ; i< studyHours.size();i++) {
    double fi = alphaStudy + (study_beta* studyHours[i]);
    fiPointsStudy.push_back(fi);
}
//exercise per day (by hour)
for(int i = 0 ; i < exerciseHours.size();i++) {
    double fi = alphaPhysical + (physical_beta* exerciseHours[i]);
    fiPointsPhysical.push_back(fi);
}
//stress
for(int i = 0 ; i< stress.size();i++) {
    double fi = alphaStress + (stress_beta * stress[i]);
    fiPointsStress.push_back(fi);
}
//sleep
for(int i = 0 ; i< sleepHours.size();i++) {
    double fi = alphaSleep + (sleep_beta* sleepHours[i]);
    fiPointsSleep.push_back(fi);
}
//wrt to sleep

for(int i = 0 ; i< studyHours.size();i++) {
    double fi = alphaStudyTwo + (study_betaTwo* studyHours[i]);
    fiPointsStudyStr.push_back(fi);
}
//exercise per day (by hour)
for(int i = 0 ; i < exerciseHours.size();i++) {
    double fi = alphaPhysicalTwo+ (physical_betaTwo* exerciseHours[i]);
    fiPointsPhysicalStr.push_back(fi);
```

In the next part, we are making the fi points for each vector in order to make the line we want to see to determine the correlation. This is done by incorporating the alpha and beta and incorporating with the vector points of the variable we want.



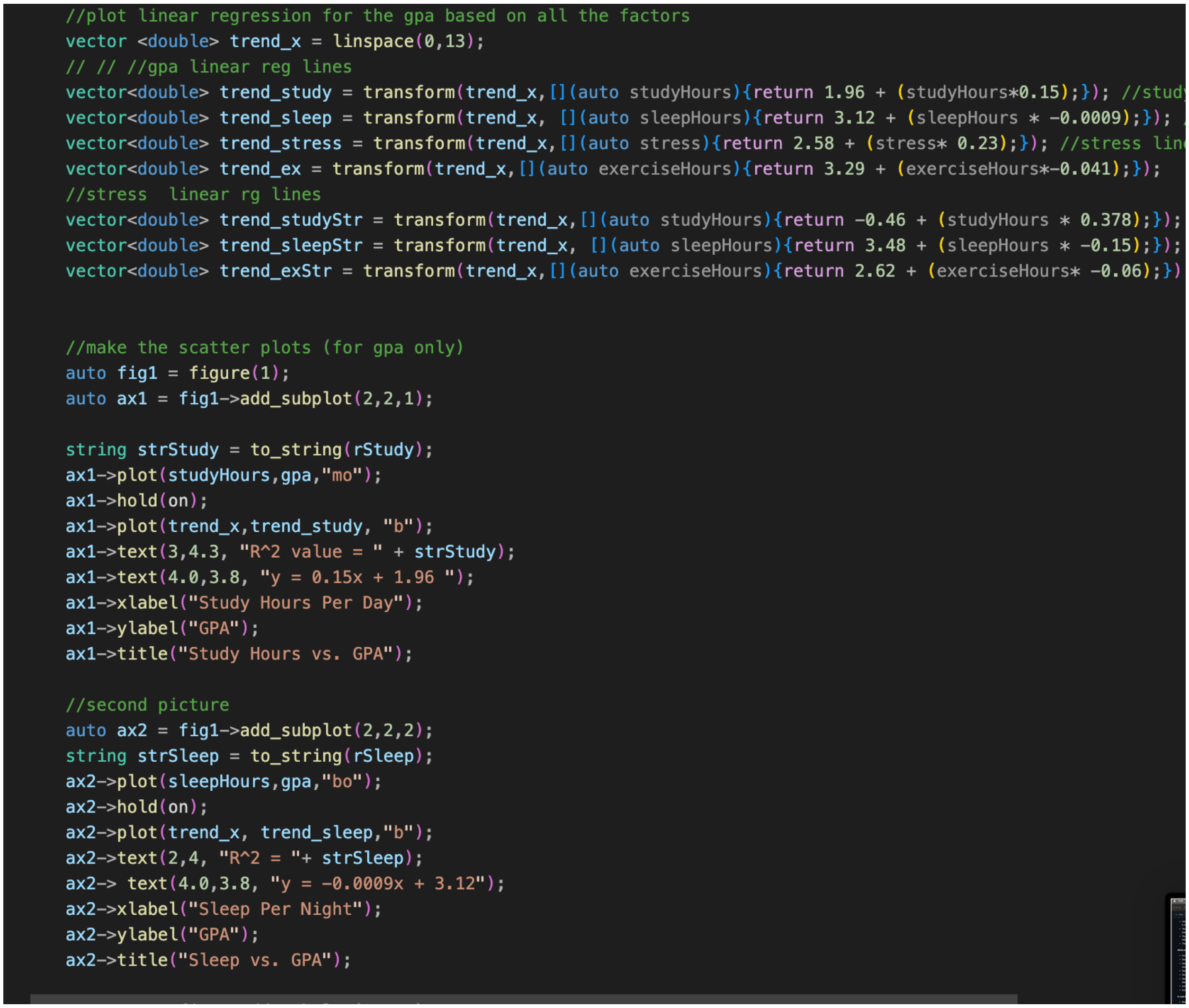
```
//ss res (wrt to gpa)
double ssResSleep = 0.0;
double ssTotSleep = 0.0;
for(int i = 0; i< fiPointsSleep.size();i++) {
    ssResSleep += pow((gpa[i]-fiPointsSleep[i]),2);
    ssTotSleep += pow((gpa[i]-mean_gpa),2);
}
double rSleep = 1-(ssResSleep/ssTotSleep);
//cout << "r squared sleep : " << rSleep << endl;

double ssResStudy = 0.0;
double ssTotStudy = 0.0;
// cout << "fi point size: " << fiPointsStudy.size() << endl;
// cout << "gpa size: " << gpa.size() << endl;
for(int i = 0; i< fiPointsStudy.size();i++) {
    ssResStudy += pow((gpa[i]-fiPointsStudy[i]),2);
    ssTotStudy += pow((gpa[i]-mean_gpa),2);
}
// cout << "rRes : " << ssResStudy << endl;
// cout << "rRes Total: " << ssTotStudy << endl;
double rStudy = 1 - (ssResStudy/ssTotStudy);
//cout << "r squared study : " << rStudy << endl;

double ssResPhys = 0.0;
double ssTotPhys = 0.0;
for(int i = 0; i< fiPointsPhysical.size();i++) {
    ssResPhys += pow((gpa[i]-fiPointsPhysical[i]),2);
    ssTotPhys += pow((gpa[i]-mean_gpa),2);
}
double rPhys = 1 - (ssResPhys/ssTotPhys);
//cout << "r squared physical: " << rPhys << endl;


double ssResStress = 0.0;
double ssTotStress = 0.0;
for(int i = 0; i< fiPointsStress.size();i++) {
    ssResStress += pow((gpa[i]-fiPointsStress[i]),2);
    ssTotStress += pow((gpa[i]-mean_gpa),2);
```



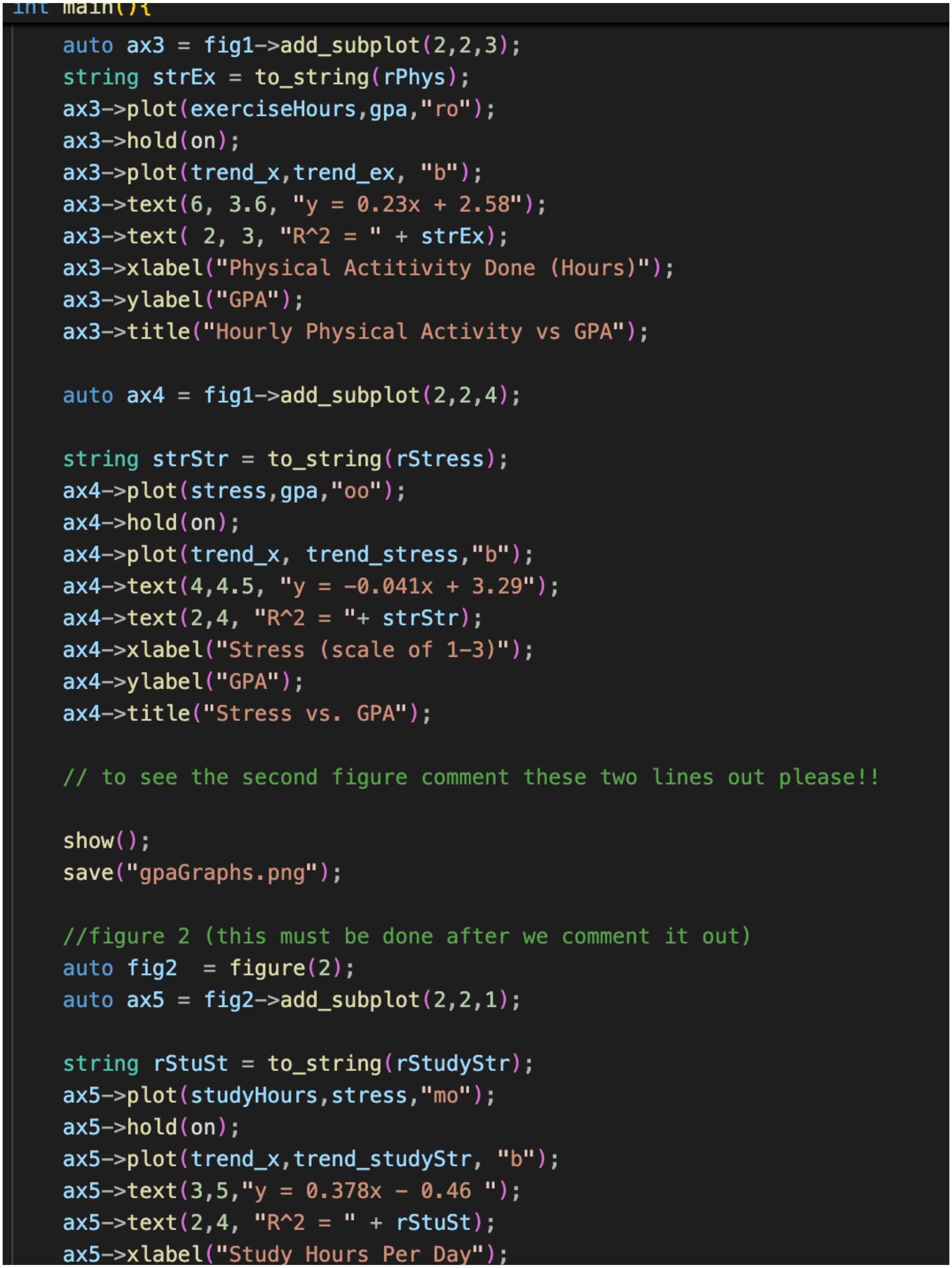
```
    double ssResStress = 0.0;
    double ssTotStress = 0.0;
    for(int i = 0; i< fiPointsStress.size();i++) {
        ssResStress += pow((gpa[i]-fiPointsStress[i]),2);
        ssTotStress += pow((gpa[i]-mean_gpa),2);
    }
    double rStress = 1 - (ssResStress/ssTotStress);
    //cout << "stress (r): " << rStress << endl;

    //ss res (wrt to stress)
    double ssResSleepStr = 0.0;
    double ssTotSleepStr = 0.0;
    for(int i = 0; i< fiPointsSleepStr.size();i++) {
        ssResSleepStr += pow((stress[i]-fiPointsSleepStr[i]),2);
        ssTotSleepStr += pow((stress[i]-mean_stress),2);
    }
    double rSleepStr = 1 - (ssResSleepStr/ssTotSleepStr);
    //cout << "r sleep (stress) : " << rSleepStr << endl;

    double ssResStudyStr = 0.0;
    double ssTotStudyStr = 0.0;
    for(int i = 0; i< fiPointsStudyStr.size();i++) {
        ssResStudyStr += pow((stress[i]-fiPointsStudyStr[i]),2);
        ssTotStudyStr += pow((stress[i]-mean_stress),2);
    }
    double rStudyStr = 1 - (ssResStudyStr/ssTotStudyStr);
    //cout << " r study (stres) : " << rStudyStr << endl;


    double ssResPhysStr = 0.0;
    double ssTotPhysStr = 0.0;
    for(int i = 0; i< fiPointsPhysicalStr.size();i++) {
        ssResPhysStr += pow((stress[i]-fiPointsPhysicalStr[i]),2);
        ssTotPhysStr += pow((stress[i]-mean_stress),2);
    }
    double rPhysStr = 1- (ssResPhysStr/ssTotPhysStr);
    //cout << "r physical (stress): " << rPhysStr << endl;

    //plot linear regression for the gpa based on all the factors
```

By making the fi points, we also want to see how meaningful the values are so we do this making a r sqared values using the fi points the actual data to see how well they compare to their residuals. In this case, we also find sqare and find use the differences between the gpa and expected mean. We then for each variable for each graph in this case there are 7. We make r squared values dividing the residuals over the total and subtracting it by one. By using vectors, for loops and math, we can successfully make a data structure for a linear regression model.

Onto the data part where we use matplot to design the plots.

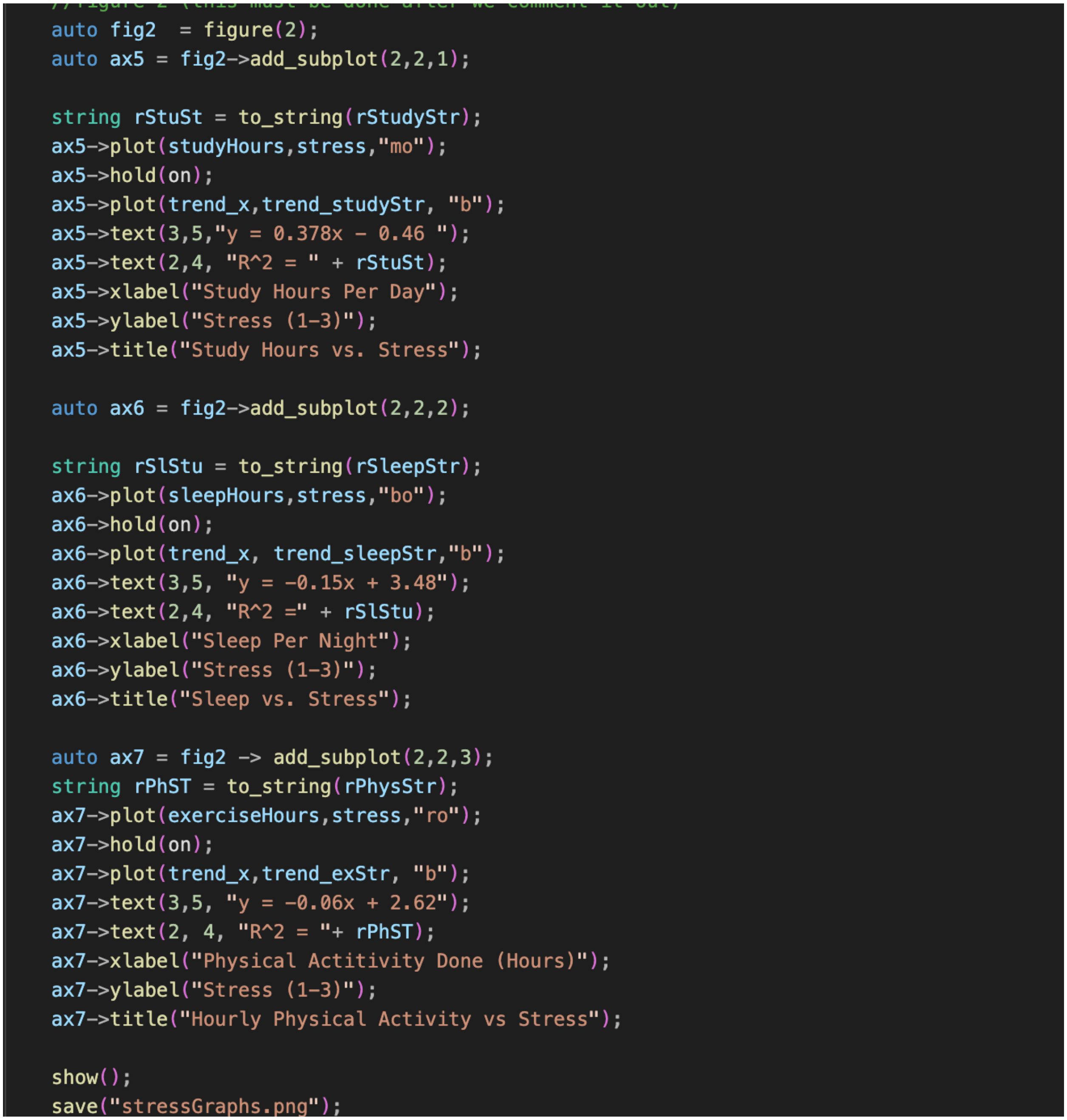


```
    //plot linear regression for the gpa based on all the factors
    vector <double> trend_x = linspace(0,13);
    // // //gpa linear reg lines
    vector<double> trend_study = transform(trend_x,[](auto studyHours){return 1.96 + (studyHours*0.15);}); //stud
    vector<double> trend_sleep = transform(trend_x, [](auto sleepHours){return 3.12 + (sleepHours * -0.0009);}); /
    vector<double> trend_stress = transform(trend_x,[](auto stress){return 2.58 + (stress* 0.23);}); //stress lin
    vector<double> trend_ex = transform(trend_x,[](auto exerciseHours){return 3.29 + (exerciseHours*-0.041);});
    //stress  linear rg lines
    vector<double> trend_studyStr = transform(trend_x,[](auto studyHours){return -0.46 + (studyHours * 0.378);});
    vector<double> trend_sleepStr = transform(trend_x, [](auto sleepHours){return 3.48 + (sleepHours * -0.15);});
    vector<double> trend_exStr = transform(trend_x,[](auto exerciseHours){return 2.62 + (exerciseHours* -0.06);})


    //make the scatter plots (for gpa only)
    auto fig1 = figure(1);
    auto ax1 = fig1->add_subplot(2,2,1);

    string strStudy = to_string(rStudy);
    ax1->plot(studyHours,gpa,"mo");
    ax1->hold(on);
    ax1->plot(trend_x,trend_study, "b");
    ax1->text(3,4.3, "R^2 value = " + strStudy);
    ax1->text(4.0,3.8, "y = 0.15x + 1.96 ");
    ax1->xlabel("Study Hours Per Day");
    ax1->ylabel("GPA");
    ax1->title("Study Hours vs. GPA");

    //second picture
    auto ax2 = fig1->add_subplot(2,2,2);
    string strSleep = to_string(rSleep);
    ax2->plot(sleepHours,gpa,"bo");
    ax2->hold(on);
    ax2->plot(trend_x, trend_sleep,"b");
    ax2->text(2,4, "R^2 = "+ strSleep);
    ax2-> text(4.0,3.8, "y = -0.0009x + 3.12");
    ax2->xlabel("Sleep Per Night");
    ax2->ylabel("GPA");
    ax2->title("Sleep vs. GPA");
```



```
int main(){
    auto ax3 = fig1->add_subplot(2,2,3);
    string strEx = to_string(rPhys);
    ax3->plot(exerciseHours,gpa,"ro");
    ax3->hold(on);
    ax3->plot(trend_x,trend_ex, "b");
    ax3->text(6, 3.6, "y = 0.23x + 2.58");
    ax3->text( 2, 3, "R^2 = " + strEx);
    ax3->xlabel("Physical Actitivity Done (Hours)");
    ax3->ylabel("GPA");
    ax3->title("Hourly Physical Activity vs GPA");

    auto ax4 = fig1->add_subplot(2,2,4);

    string strStr = to_string(rStress);
    ax4->plot(stress,gpa,"oo");
    ax4->hold(on);
    ax4->plot(trend_x, trend_stress,"b");
    ax4->text(4,4.5, "y = -0.041x + 3.29");
    ax4->text(2,4, "R^2 = "+ strStr);
    ax4->xlabel("Stress (scale of 1-3)");
    ax4->ylabel("GPA");
    ax4->title("Stress vs. GPA");

    // to see the second figure comment these two lines out please!!

    show();
    save("gpaGraphs.png");

    //figure 2 (this must be done after we comment it out)
    auto fig2  = figure(2);
    auto ax5 = fig2->add_subplot(2,2,1);

    string rStuSt = to_string(rStudyStr);
    ax5->plot(studyHours,stress,"mo");
    ax5->hold(on);
    ax5->plot(trend_x,trend_studyStr, "b");
    ax5->text(3,5,"y = 0.378x - 0.46 ");
    ax5->text(2,4, "R^2 = " + rStuSt);
    ax5->xlabel("Study Hours Per Day");
```

To make the line and the plots we first need to make double vectors where we can transform the data so it fits between the data, we can available. I chose 0 – 13 because it made the most practical sense as most of the data was looking at daily hours. Next, we assembled the linear regression where we use the x values to help formulate the line using the values that were originally couted. We manually add the beta and the alpha values for each line. We also start to see the graphs being made, so this is done by using figure to help make subplots of the data that we are seeing. We have two figures for the two equations we made for GPA and stress. In this first figure, I decided to modify it by adding the legends and the line and r squared values on the graph. This is done using the functions found in plot header file. I also had doubles that I wanted to use as a string so I converted them so I could concatenate it, so it could appear on the graph. In the second image we have a code that needs to be commented out to see the second figure so that is mentioned in the code.



```
    //figure 2 (this must be done after we comment it out)
    auto fig2  = figure(2);
    auto ax5 = fig2->add_subplot(2,2,1);

    string rStuSt = to_string(rStudyStr);
    ax5->plot(studyHours,stress,"mo");
    ax5->hold(on);
    ax5->plot(trend_x,trend_studyStr, "b");
    ax5->text(3,5,"y = 0.378x - 0.46 ");
    ax5->text(2,4, "R^2 = " + rStuSt);
    ax5->xlabel("Study Hours Per Day");
    ax5->ylabel("Stress (1-3)");
    ax5->title("Study Hours vs. Stress");

    auto ax6 = fig2->add_subplot(2,2,2);

    string rSlStu = to_string(rSleepStr);
    ax6->plot(sleepHours,stress,"bo");
    ax6->hold(on);
    ax6->plot(trend_x, trend_sleepStr,"b");
    ax6->text(3,5, "y = -0.15x + 3.48");
    ax6->text(2,4, "R^2 =" + rSlStu);
    ax6->xlabel("Sleep Per Night");
    ax6->ylabel("Stress (1-3)");
    ax6->title("Sleep vs. Stress");

    auto ax7 = fig2 -> add_subplot(2,2,3);
    string rPhST = to_string(rPhysStr);
    ax7->plot(exerciseHours,stress,"ro");
    ax7->hold(on);
    ax7->plot(trend_x,trend_exStr, "b");
    ax7->text(3,5, "y = -0.06x + 2.62");
    ax7->text(2, 4, "R^2 = "+ rPhST);
    ax7->xlabel("Physical Actitivity Done (Hours)");
    ax7->ylabel("Stress (1-3)");
    ax7->title("Hourly Physical Activity vs Stress");

    show();
    save("stressGraphs.png");
```

This is for producing the graph that is composed of the stress levels and their given factors. This is done in the same fashion as using the given header file and functionality that was talked about in the first figure.

In order to compose the code I have been using:

```
clang++ -g project.cpp -o project -std=c++17 -I/opt/homebrew/include -ldlib -lmatplot -lblas -L/opt/homebrew/lib
```

This is so I could use the matplot feature and have the most recent update in order to produce the most accurate representation of the code.