I wrote a program that inputs a comma-separated values (CSV) text file downloaded from Thinkorswim–a program that is used by day traders to view the stock market–and outputs the total dollar volume traded within the parameters. I use this program to track dollar volume traded between specific price ranges of stocks i.e. 0-2, 3-6, 6-10, 11-50 et cetera. This data can help you notice trends in the market and help identify when the market goes into a lull. It can also directly show at what range the most money is traded.

I have ideas and notes on how to automate this process. Instead of taking the data three times a day manually I could take the data every hour, thirty minutes, or even less to greatly increase accuracy. After an automated CSV download and run through my program I could have the program extract the data and display it on a chart. This chart would be a visual representation of the amount of dollar volume traded on the market in any given price range. After enough time has passed you could notice trends like the summer lull that normally happens but with specific data to support your claims.

```
/*
Dillon Poe
2022
Dollar Volume Calculator
*/

#include <iostream>
#include <fstream>
#include <string>
#include <bits/stdc++.h>
#include <sstream>
#include <cstdlib>
#include <algorithm>
using namespace std;
int firstrun();
int secrun();
int thirdrun();

int main ()
{
  firstrun(); //0900
  secrun(); //1100
  thirdrun(); //0105

  remove( "output1.txt" );
  remove( "output3.txt" );
  remove( "output4.txt" );

  cout << "All runs completed" << '\n';
```

```
  return 0;
}

int firstrun()
{
  string line;
  ifstream input ("OTC-001-011.csv");
  ofstream output ("output1.txt");
  if (input.is_open())
  {
    while ( getline (input,line) )
    {
      string pps;
      pps.clear();
      string volume;
      volume.clear();
      string totald;
      totald.clear();
      if (output.is_open())
      {
        int e = 0; //setup for not outputting the 2nd time
        int f = 0; //setup for not outputting the 2nd time
        for( size_t i=0; i<line.length(); i++)
        {
          char c = line[i];

          //get the ticker
          if( isupper(line[i]))
          {
            i++;
            if( isupper(line[i]))
            {
            output << c;
            c = line[i];
            output << c;
                while( c != ',')
                {
                i++;
                c = line[i];
                  if(c != ',')
                  {
                  output << c;
                  }
                }
```

```
    output << " ";
    }
  }

  //gets the PPS, price per share
  if( c == '.')
  {
  i++;
      if (f <= 0) //only takes the first numbers behind the first '.'
      {
      f++;
      c = line[i];
        output << '.';
        pps.append(1, '.');
        while( c >= '0')
        {
        output << c;
        pps.append(1, c);
        i++;
        c = line[i];
        }
      output << " ";
      }
  }

  //gets the volume
  if( c == '\"' )
  {
  i++;
      if (e < 1)
      {
      e++;
      c = line[i];
        while( c != '\"')
        {
        output << c;
        volume.append(1, c);
        i++;
        c = line[i];
        }
      output << " " << "=" << " " << "$" << '\n';
      }
  }
}
```

```
    }
    else cout << "Unable to open file output";
    //inside the input loop
    //every new line goes through here
   }
  }
  else cout << "Unable to open file input";
//need to reopen the output file as an input then output again? prob not good idea, but did it
anyways, can combine these parts in the future
  input.close();
  output.close();

//start of a new opening of inputs to create a final output with math done
//need to remove all strings and use ints
  unsigned long long int totaldollar = 0;
  unsigned long long int volume = 0;
  long double pps = 0.0;
  int x;
  x = 0;
  string line2;
  ifstream input2 ("output1.txt");
  ofstream output2 ("2021-10-  -watchlist0900.csv");
  if (input2.is_open())
  {
    while ( getline (input2,line2) )
    {
      if (output2.is_open())
      {
        for( size_t i=0; i<line2.length(); i++)
        {
          char c = line2[i];
          //get the ticker
          if( isupper(line2[i]))
          {
            i++;
            if( isupper(line2[i]))
            {
            output2 << c;
            c = line2[i];
            output2 << c;
                while( c != ' ')
                {
                i++;
                c = line2[i];
```

```
        if(c != ' ')
        {
        output2 << c;
        }
      }
  output2 << " ";
  }
  x = 0;//resets the loop for the PPS and volume
}

if( c == ' ' ) //this if statement gets both numbers on the line
{
i++;
    while (x < 1) //getting the PPS, first set of number
    {
    x++;
    c = line2[i];
    string storing;
      while( c != ' ')
      {
      //instead of outputting store as a string then extract
      storing.append(1, c);
      line2.append(1, c);
      i++;
      c = line2[i];
      }
      //turn storing into a double
      stringstream ss;
      ss << storing;
      ss >> pps;
      output2 << pps;
    }

if( c == ' ' )
{
i++;
    if (x < 2) //getting the volume, second set of number
    {
    x++;
    c = line2[i];
    string storing;
      while( c != ' ')
      {
      storing.append(1, c);
```

```
            line2.append(1, c);
            i++;
            c = line2[i];
            }
    storing.erase(remove(storing.begin(), storing.end(), ','), storing.end()); //removes all commas
          stringstream ss;  //start of string conversion to long int
          ss << storing;
          ss >> volume; //volume (an int) holds the value that was in storing (a string)
          output2 << " x " << volume;
          unsigned long long int dollarv = volume * pps; //dollar volume
          output2 << " = $" << dollarv << '\n';
          totaldollar = dollarv + totaldollar; //keeps adding all totals
        }
       }

       }
      }
    }
    else cout << "Unable to open file 0900";
  }
 }
 else cout << "Unable to open file output1";

 time_t now = time(0);
 char* dt = ctime(&now);

 output2 << '\n' << "Total: $" << totaldollar << '\n' << '\n' << "Date of compile:" << '\n' << dt;

 input2.close();
 output2.close();

 cout << "Run 1 completed" << '\n';
 return 0;
}

int secrun()
{
 string line;
 ifstream input ("OTC-01-40.csv");
 ofstream output ("output3.txt");
 if (input.is_open())
 {
  while ( getline (input,line) )
  {
```

```
string pps;
pps.clear();
string volume;
volume.clear();
string totald;
totald.clear();
if (output.is_open())
{
  int e = 0;
  int f = 0;
  for( size_t i=0; i<line.length(); i++)
  {
    char c = line[i];
    if( isupper(line[i]))
    {
      i++;
      if( isupper(line[i]))
      {
      output << c;
      c = line[i];
      output << c;
          while( c != ',')
          {
          i++;
          c = line[i];
            if(c != ',')
            {
            output << c;
            }
          }
      output << " ";
      }
    }
    if( c == '.')
    {
    i++;
        if (f <= 0)
        {
        f++;
        c = line[i];
          output << '.';
          pps.append(1, '.');
          while( c >= '0')
          {
```

```
            output << c;
            pps.append(1, c);
            i++;
            c = line[i];
            }
          output << " ";
          }
        }
        if( c == '\"' )
        {
        i++;
          if (e < 1)
          {
          e++;
          c = line[i];
            while( c != '\"')
            {
            output << c;
            volume.append(1, c);
            i++;
            c = line[i];
            }
          output << " " << "=" << " " << "$" << '\n';
          }
        }
      }
     }
     else cout << "Unable to open file output";
   }
  }
  else cout << "Unable to open file input";
//need to reopen the output file as an input then output again? prob not good idea, but did it
anyways, can combine these parts in the future
  input.close();
  output.close();

  unsigned long long int totaldollar = 0;
  unsigned long long int volume = 0;
  long double pps = 0.0;
  int x;
  x = 0;
  string line2;
  ifstream input2 ("output3.txt");
  ofstream output2 ("2021-10-  -watchlist1100.csv");
```

```
if (input2.is_open())
{
  while ( getline (input2,line2) )
  {
    if (output2.is_open())
    {
      for( size_t i=0; i<line2.length(); i++)
      {
        char c = line2[i];
        if( isupper(line2[i]))
        {
          i++;
          if( isupper(line2[i]))
          {
          output2 << c;
          c = line2[i];
          output2 << c;
              while( c != ' ')
              {
              i++;
              c = line2[i];
                if(c != ' ')
                {
                output2 << c;
                }
              }
          output2 << " ";
          }
          x = 0;
        }

        if( c == ' ' )
        {
        i++;
            while (x < 1)
            {
            x++;
            c = line2[i];
            string storing;
              while( c != ' ')
              {
              storing.append(1, c);
              line2.append(1, c);
              i++;
```

```
            c = line2[i];
            }
            stringstream ss;
            ss << storing;
            ss >> pps;
            output2 << pps;
          }

      if( c == ' ' )
      {
      i++;
         if (x < 2)
         {
         x++;
         c = line2[i];
         string storing;
           while( c != ' ')
           {
           storing.append(1, c);
           line2.append(1, c);
           i++;
           c = line2[i];
           }
   storing.erase(remove(storing.begin(), storing.end(), ','), storing.end());
         stringstream ss;
         ss << storing;
         ss >> volume;
         output2 << " x " << volume;
         unsigned long long int dollarv = volume * pps;
         output2 << " = $" << dollarv << '\n';
         totaldollar = dollarv + totaldollar;
       }
      }

      }
     }
   }
   else cout << "Unable to open file output2";
  }
}
else cout << "Unable to open file input2";

time_t now = time(0);
char* dt = ctime(&now);
```

```
  output2 << '\n' << "Total: $" << totaldollar << '\n' << '\n' << "Date of compile:" << '\n' << dt;

  input2.close();
  output2.close();
cout << "Run 2 completed" << '\n';
return 0;
}

int thirdrun()
{
  string line;
  ifstream input ("OTC-30-99.csv");
  ofstream output ("output4.txt");
  if (input.is_open())
  {
    while ( getline (input,line) )
    {
      string pps;
      pps.clear();
      string volume;
      volume.clear();
      string totald;
      totald.clear();
      if (output.is_open())
      {
        int e = 0;
        int f = 0;
        for( size_t i=0; i<line.length(); i++)
        {
          char c = line[i];
          if( isupper(line[i]))
          {
            i++;
            if( isupper(line[i]))
            {
            output << c;
            c = line[i];
            output << c;
                while( c != ',')
                {
                i++;
                c = line[i];
                  if(c != ','
```

```
          {
          output << c;
          }
        }
    output << " ";
    }
  }
  if( c == '.')
  {
  i++;
      if (f <= 0)
      {
      f++;
      c = line[i];
        output << '.';
        pps.append(1, '.');
        while( c >= '0')
        {
        output << c;
        pps.append(1, c);
        i++;
        c = line[i];
        }
      output << " ";
      }
  }
  if( c == '\"' )
  {
  i++;
      if (e < 1)
      {
      e++;
      c = line[i];
        while( c != '\"')
        {
        output << c;
        volume.append(1, c);
        i++;
        c = line[i];
        }
      output << " " << "=" << " " << "$" << '\n';
      }
  }
}
```

```
    }
    else cout << "Unable to open file output";
   }
  }
  else cout << "Unable to open file input";
//need to reopen the output file as an input then output again? prob not good idea, but did it
anyways, can combine these parts in the future
  input.close();
  output.close();

  unsigned long long int totaldollar = 0;
  unsigned long long int volume = 0;
  long double pps = 0.0;
  int x;
  x = 0;
  string line2;
  ifstream input2 ("output4.txt");
  ofstream output2 ("2021-10-  -watchlist0105.csv");
  if (input2.is_open())
  {
    while ( getline (input2,line2) )
    {
      if (output2.is_open())
      {
        for( size_t i=0; i<line2.length(); i++)
        {
          char c = line2[i];
          if( isupper(line2[i]))
          {
            i++;
            if( isupper(line2[i]))
            {
            output2 << c;
            c = line2[i];
            output2 << c;
                while( c != ' ')
                {
                i++;
                c = line2[i];
                  if(c != ' ')
                  {
                  output2 << c;
                  }
                }
```

```
  output2 << " ";
  }
  x = 0;
}

if( c == ' ' )
{
i++;
   while (x < 1)
   {
   x++;
   c = line2[i];
   string storing;
     while( c != ' ')
     {
     storing.append(1, c);
     line2.append(1, c);
     i++;
     c = line2[i];
     }
     stringstream ss;
     ss << storing;
     ss >> pps;
     output2 << pps;
   }

if( c == ' ' )
{
i++;
   if (x < 2)
   {
   x++;
   c = line2[i];
   string storing;
     while( c != ' ')
     {
     storing.append(1, c);
     line2.append(1, c);
     i++;
     c = line2[i];
     }
storing.erase(remove(storing.begin(), storing.end(), ','), storing.end());
   stringstream ss;
   ss << storing;
```

```
            ss >> volume;
            output2 << " x " << volume;
            unsigned long long int dollarv = volume * pps;
            output2 << " = $" << dollarv << '\n';
            totaldollar = dollarv + totaldollar;
          }
         }

         }
       }
     }
     else cout << "Unable to open file output2";
   }
  }
  else cout << "Unable to open file input2";

  time_t now = time(0);
  char* dt = ctime(&now);

  output2 << '\n' << "Total: $" << totaldollar << '\n' << '\n' << "Date of compile:" << '\n' << dt;

  input2.close();
  output2.close();
cout << "Run 3 completed" << '\n';
return 0;
}
```