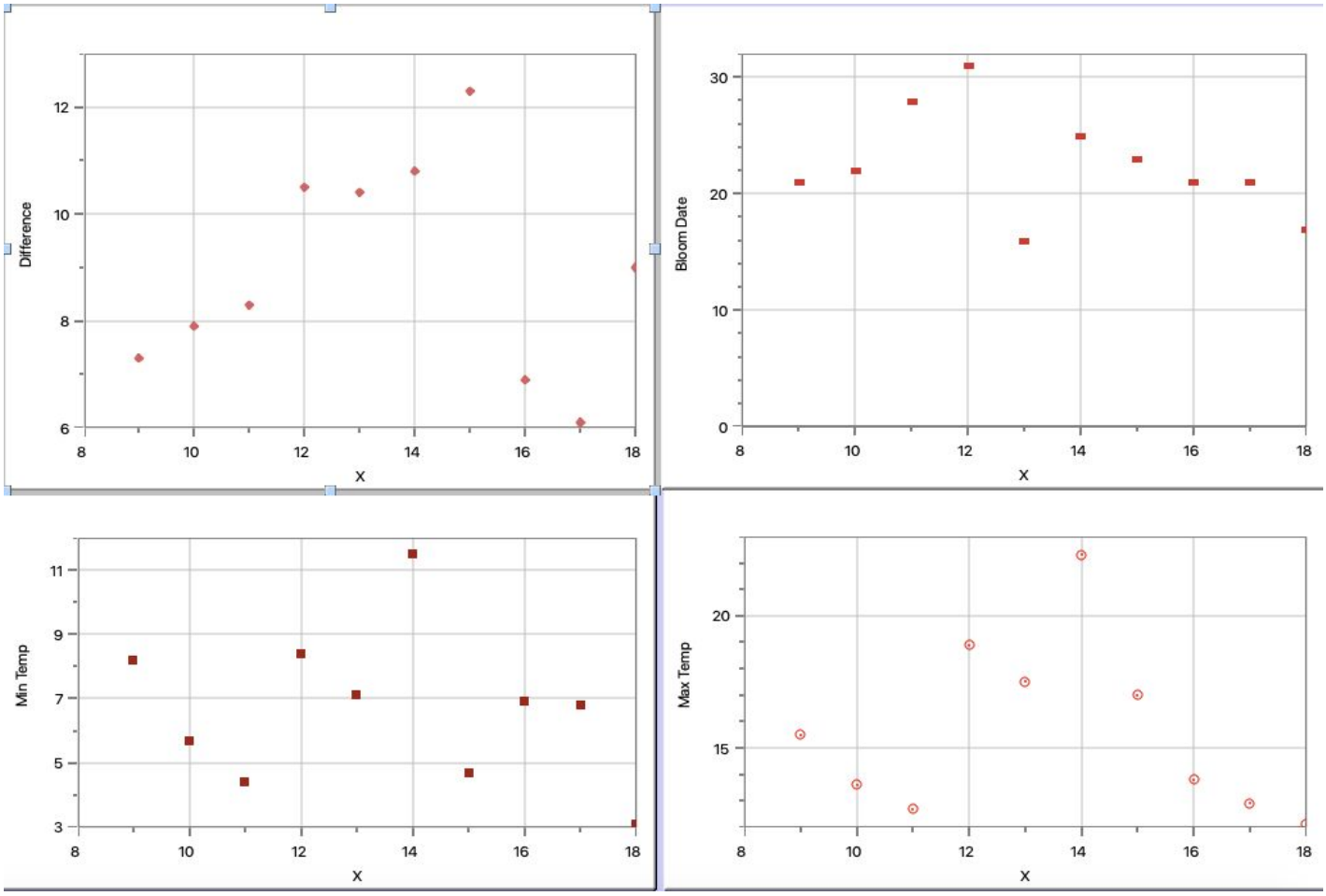

This is my initial analysis of the data to try to find a trend. I graphed the min and max temps for the first bloom dates (lower left and right). I also graphed the day in March that the first bloom occurred for each year (upper right). Finally I calculated the difference between the min and max temps for the bloom dates by using code shown below (upper right). I was surprised by this data because for the most part there was no visible trend. However, using this data I decided to look into the relationship between the min temps and the bloom date as lower min temps meant a later bloom date.

```
ArrayList<YearData> yearsList = readData(file);

// Creates an ArrayList to store the first bloom dates.
ArrayList<Integer> bloomDates = new ArrayList<Integer>();
Integer[] dates = {21, 22, 28, 31, 16, 25, 23, 21, 21, 17, 21};
for (int a = 0; a < dates.length; a++) {
   bloomDates.add(dates[a]);
}

// Loop to print out the existing data to see how it is structured.
for(int i = 0; i < yearsList.size(); i++) {
   ArrayList<Record> temp = yearsList.get(i).getData();
   for(int j = 0; j < temp.size(); j++) {
      if (temp.get(j).getMonth() == 2) {
         if (temp.get(j).getDay() == bloomDates.get(0)){
            System.out.println(temp.get(j));
            bloomDates.remove(0);
         }
      }
```

akura.java | Record.java | YearData.java

ile Messages | jGRASP Messages | Run I/O | Interactions

```
Month: 3 Day: 21 max temp: 15.5 min temp: 8.2
Month: 3 Day: 22 max temp: 17.0 min temp: 13.0
Month: 3 Day: 28 max temp: 10.8 min temp: 4.9
Month: 3 Day: 31 max temp: 13.6 min temp: 7.3
Month: 3 Day: 16 max temp: 21.6 min temp: 10.7
Month: 3 Day: 25 max temp: 7.2 min temp: 3.2
Month: 3 Day: 23 max temp: 10.0 min temp: 2.7
Month: 3 Day: 21 max temp: 11.6 min temp: 4.9
Month: 3 Day: 21 max temp: 16.2 min temp: 7.7
Month: 3 Day: 17 max temp: 16.8 min temp: 6.6
Month: 3 Day: 21 max temp: 14.5 min temp: 7.0
```

I used code to isolate the data from the past bloom dates.

We averaged out the dates for the first bloom and we got 22.3. We then looked at the date and found that the average max temp is 15.63 and the average min temp is 6.68.

```
// Loop to print out March dates of each year close to the average min temp of bloom dates
 ArrayList<Record> minDates = new Arraylist<Record>();
 for(int i = 0; i < yearsList.size(); i++) {
   ArrayList<Record> temp = yearsList.get(i).getData();
   for(int j = 0; j < temp.size(); j++) {
      if (temp.get(j).getMonth() == 2){
         if (temp.get(j).getMinTemp() > 6 && temp.get(j).getMinTemp() < 7) {
            minDates.add(temp.get(j));
          // System.out.println("Year: " + temp.get(j).getYear() + ", day: " + temp.get(j).getDay())
```

```
// Loop to print out March dates of each year close to the average max temp of bloom dates
ArrayList<Record> maxDates = new Arraylist<Record>();
for(int i = 0; i < yearsList.size(); i++) {
  ArrayList<Record> temp = yearsList.get(i).getData();
  for(int j = 0; j < temp.size(); j++) {
     if (temp.get(j).getMonth() == 2){
        if (temp.get(j).getMaxTemp() > 15 && temp.get(j).getMaxTemp() < 16) {
           maxDates.add(temp.get(j));
          // System.out.println("Year: " + temp.get(j).getYear() + ", day: " +temp.get(j).getDay());
```

```
Year: 2009, day: 10
Year: 2009, day: 14
Year: 2009, day: 16
Year: 2009, day: 24
Year: 2009, day: 25
Year: 2010, day: 1
Year: 2010, day: 21
Year: 2011, day: 14
Year: 2011, day: 19
Year: 2011, day: 30
Year: 2012, day: 18
Year: 2012, day: 19
Year: 2013, day: 25
Year: 2013, day: 27
Year: 2013, day: 31
Year: 2014, day: 14
Year: 2014, day: 17
Year: 2014, day: 20
Year: 2015, day: 22
Year: 2016, day: 17
Year: 2016, day: 21
Year: 2017, day: 14
Year: 2017, day: 21
Year: 2017, day: 30
Year: 2018, day: 1
Year: 2018, day: 9
Year: 2018, day: 20
Year: 2018, day: 25
Year: 2019, day: 1
Year: 2019, day: 2
Year: 2019, day: 10
Year: 2019, day: 16
Year: 2019, day: 19
```

```
Year: 2009, day: 21
Year: 2010, day: 12
Year: 2011, day: 13
Year: 2011, day: 29
Year: 2011, day: 31
Year: 2012, day: 6
Year: 2012, day: 7
Year: 2013, day: 12
Year: 2014, day: 27
Year: 2015, day: 19
Year: 2015, day: 24
Year: 2015, day: 26
Year: 2016, day: 4
Year: 2016, day: 7
Year: 2017, day: 5
Year: 2017, day: 16
Year: 2017, day: 17
Year: 2017, day: 23
Year: 2018, day: 12
Year: 2018, day: 18
Year: 2019, day: 5
Year: 2019, day: 17
Year: 2019, day: 24
Year: 2019, day: 26
```

Left is min temps, right is max temps

Code by Jessie

Jessie then used code to find the dates with the average min or max temp close to the average that I found. This led us to believe that the number of mins/maxs affected the bloom date as when there was only one the bloom date was within 2-3 days, however when there were more than 1 max/min the actual bloom date occurred within 2-3 days of the second, or third (depending on the number of times the max/min occured).

```
// Loops to find dates matching in minDates and maxDates
ArrayList<Record> sameDates = new ArrayList<Record>();
for (int y = 0; y < maxDates.size(); y++) {
  for (int z = 0; z < minDates.size(); z++) {
     if (maxDates.get(y).equals(minDates.get(z)))
     sameDates.add(maxDates.get(y));
  }
}
for (Record day : sameDates) {
  System.out.println(day);
}
```

```
int year;
double max;
double min;
double deltaMinMax;
 for(int q=0;q<yearsList.size();q++){
 year=yearsList.get(q).getYear();
    ArrayList<Record> year2009=yearsList.get(q).getData();
    for(int w=0;w<year2009.size();w++){
    if(year==2009){
    max=year2009.get(w).getMaxTemp();
    min=year2009.get(w).getMinTemp();
    deltaMinMax=max-min;

    System.out.println(year2009.get(w).getDay() +": "+ deltaMinMax);
    }

 }}
```

This code prints out the difference between the min and max temps for the bloom dates to see if there is any correlation between the difference and the date. However, as shown in the above graphs, we were not able to see any correlation.

So using all of this data, we predict that the first bloom of 2020 will occur on March 14 as this is when there is a predicted high of 15 for the third time and predicted low of 6 degrees.