01-06: Boxplots 1

1 - Purpose

- Order boxplots by factor
- add text to a plot area
- find, and use, quantile values
- put error bars on boxplot

2 - Concepts

3 - Get data

For this lesson, we will work from the data file we create in the last lesson: LansingNOAA2016-2.csv.

```
source( file="scripts/reference.R" );
weatherData = read.csv( file="data/LansingNOAA2016-2.csv",
                        stringsAsFactors = FALSE );
```

4 - Creating three new columns

We are going to create, and then use, three new columns in ***weatherData***:

- Wind Speed (values: high, medium, low)
- Wind Direction (values: north, south, west, east)
- Change in Temperature (a numeric value giving the difference between high temperatures for consecutive days)

4.1 - Wind speed

The column ***windSpeed*** gives the wind speed in **miles/hour** for each day. We will categorize these wind speed values as **high, medium**, and **low** winds, where:

- **low**: bottom 30% (0-0.3 quantile) of wind speed values
- **medium**: 30%-70% (0.3-0.7 quantile) of wind speed values
- **high**: top 30% (0.7-1.0 quantile) of wind speed values

To do this we need to know what the **30th** (0.3) and **70th** (0.7) percentile (quantile) values for ***windSpeed*** are.

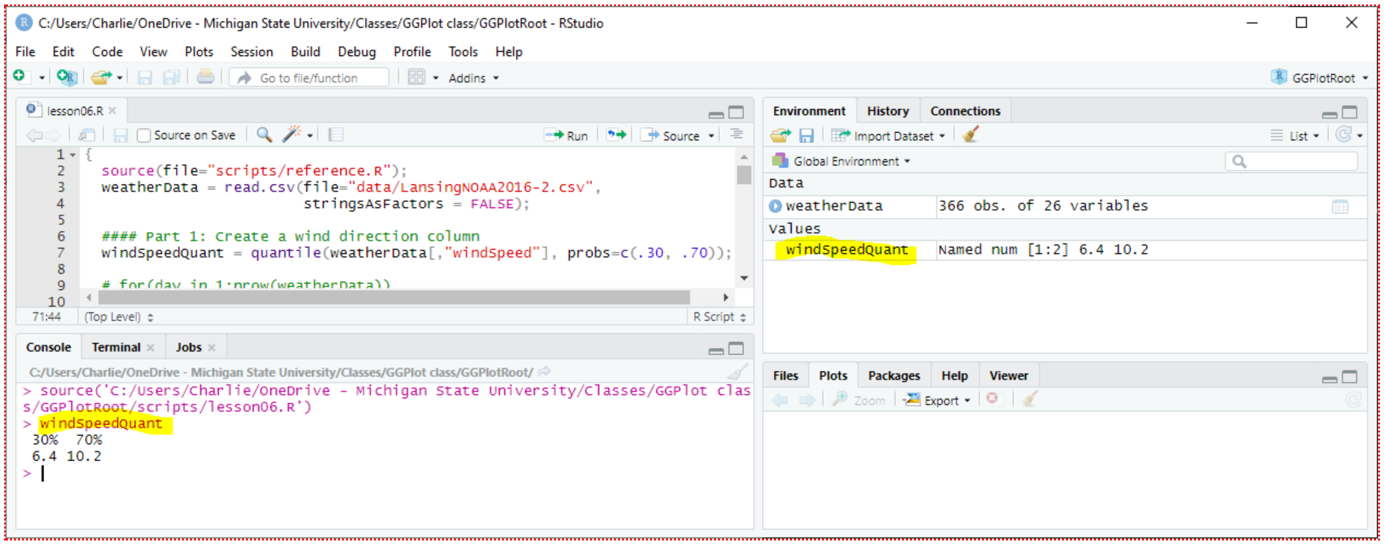
4.1.1 - Finding percentiles (quantiles)

We can find percentile values using ***quantile():***

In ***quantiles()***, we pass in the vector we want to check (***windSpeed***) and a vector, called ***probs***, that gives the quantile values we want to find.

```
windSpeedQuant = quantile(weatherData[,"windSpeed"], probs=c(.30, .70));
```

The above code says that the **0.30** and **0.70** quantile values will be found in ***windSpeed*** and these two values will be saved to the variable ***windSpeedQuant.*** The **0.30** and **0.70** quantile values are **6.2** and **10.4** as shown in *fig 1*.


C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
GGPlotRoot
lesson06.R
Source on Save
Run
Source
71:44
(Top Level)
R Script
Console Terminal Jobs
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
Environment History Connections
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
Zoom
Export

```r
{
  source(file="scripts/reference.R");
  weatherData = read.csv(file="data/LansingNOAA2016-2.csv",
                           stringsAsFactors = FALSE);

  #### Part 1: Create a wind direction column
  windSpeedQuant = quantile(weatherData[,"windSpeed"], probs=c(.30, .70));

  # for(day in 1:nrow(weatherData))
```

```
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot clas
s/GGPlotRoot/scripts/lesson06.R')
> windSpeedQuant
 30%  70%
 6.4 10.2
> 
```

```
Data
weatherData        366 obs. of 26 variables
Values
windSpeedQuant     Named num [1:2] 6.4 10.2
```

Fig 1: The 0.30 and 0.70 quantile values for wind speed

4.1.2 - Categorizing using quantiles

So, ***windSpeedQuant[1]*** = **6.4** and ***windSpeedQuant[2]*** = **10.2**. Now, we can compare all **366** values in ***windSpeed*** to the quantile values and place the wind speed values in a category: **low**, **medium**, or **high**.

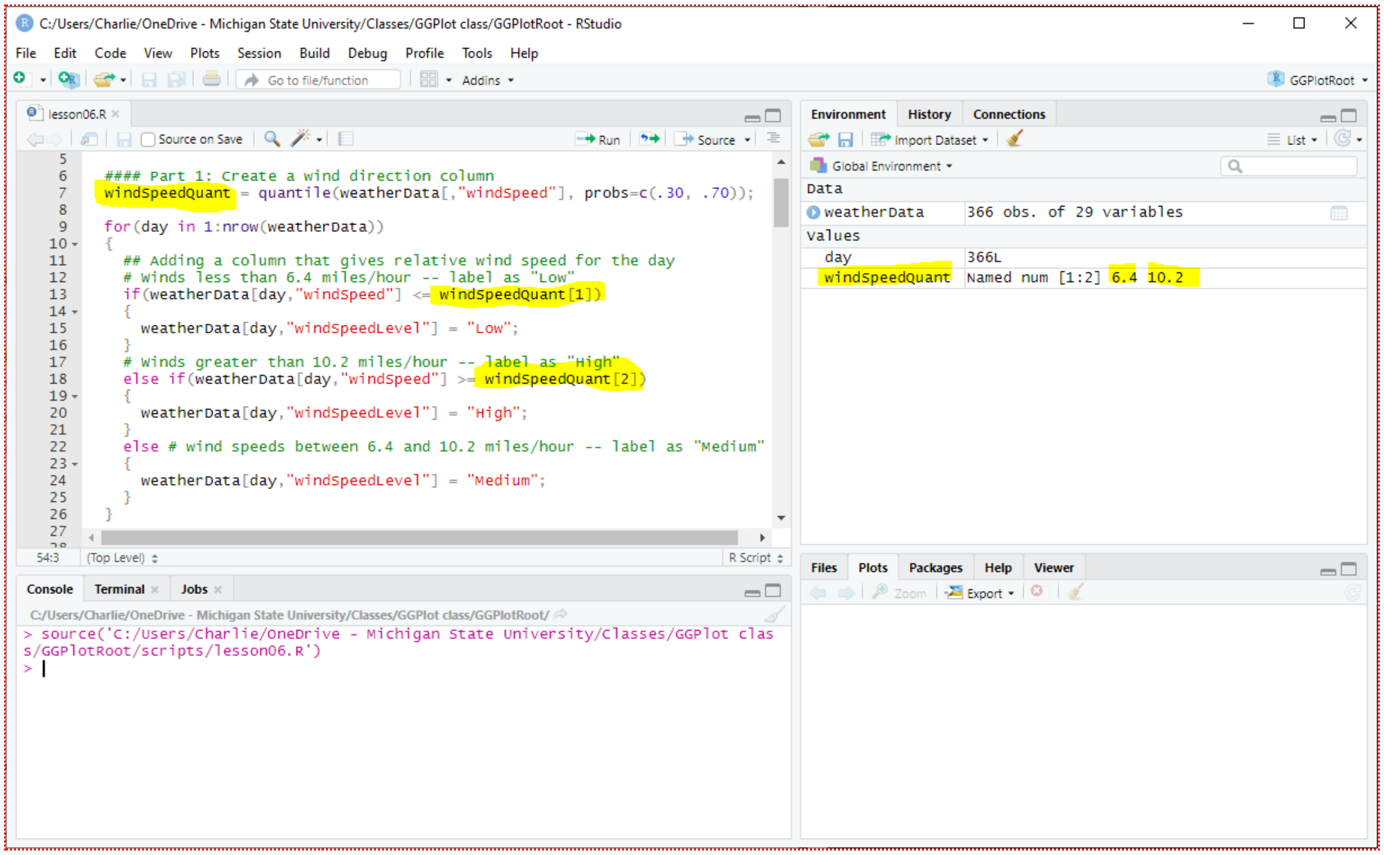
First, we use a ***for()*** to go through all rows of ***weatherData***. The number of rows is given by ***nrow()***

```r
for(day in 1:nrow(weatherData))  # nrow(weatherData) = 366
```

*Note: we could have used **for(i in 1:366)** but using **nrow(weatherData)** makes the code more scalable (i.e., it will still work if the number of rows, or days, changes)*

Inside the ***for()***, we will use an ***if-else*** structure to compare each ***windSpeed*** values to the quantile values in ***windSpeedQuant*** and, accordingly, set the values in our new column, ***windSpeedLevel,*** to low, medium, or high.

```r
for(day in 1:nrow(weatherData))  # nrow(weatherData) = 366
{
  # if the value in windSpeed is less than or equal to the low quant value
  if(weatherData[day,"windSpeed"] <= windSpeedQuant[1])      # <= 6.4
  {
    weatherData[day,"windSpeedLevel"] = "Low";
  }
  # if the value in windSpeed is greater than or equal to the high quant value
  else if(weatherData[day,"windSpeed"] >= windSpeedQuant[2]) # >= 10.2
  {
    weatherData[day,"windSpeedLevel"] = "High";
  }
  else # the value in windSpeed is between 6.2 and 10.4
  {
    weatherData[day,"windSpeedLevel"] = "Medium";
  }
}
```



C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot - RStudio

```
#### Part 1: Create a wind direction column
windSpeedQuant = quantile(weatherData[,"windSpeed"], probs=c(.30, .70));

for(day in 1:nrow(weatherData))
{
  ## Adding a column that gives relative wind speed for the day
  # winds less than 6.4 miles/hour -- label as "Low"
  if(weatherData[day,"windSpeed"] <= windSpeedQuant[1])
  {
    weatherData[day,"windSpeedLevel"] = "Low";
  }
  # winds greater than 10.2 miles/hour -- label as "High"
  else if(weatherData[day,"windSpeed"] >= windSpeedQuant[2])
  {
    weatherData[day,"windSpeedLevel"] = "High";
  }
  else # wind speeds between 6.4 and 10.2 miles/hour -- label as "Medium"
  {
    weatherData[day,"windSpeedLevel"] = "Medium";
  }
}
```

```
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot clas
s/GGPlotRoot/scripts/lesson06.R')
> 
```

```
Data
weatherData      366 obs. of 29 variables
Values
  day            366L
  windSpeedQuant Named num [1:2] 6.4 10.2
```

Fig 2: Creating a wind speed level column using quantiles

The new column, ***windSpeedLevel***, is shown in *fig 4*.

4.2 - Wind direction

The column in ***weatherData*** called ***windSusDir*** gives the direction that the most sustained wind came from for the day. The direction is in degrees from **0** to **360** with **0** being due north and **180** being due south.

We are going to convert the degrees to the closest cardinal direction (Fig *3*) and save the value to the column ***windDir***:

- **North** is from **315** degrees to **45** degrees (true north is **0** degrees)
- **East** is from **45** degrees to **135** degrees (true East is **90** degrees)
- **South** is from **135** degrees to **225** degrees (true South is **180** degrees)
- **West** is from **225** degrees to **315** degrees (true West is **270** degrees)

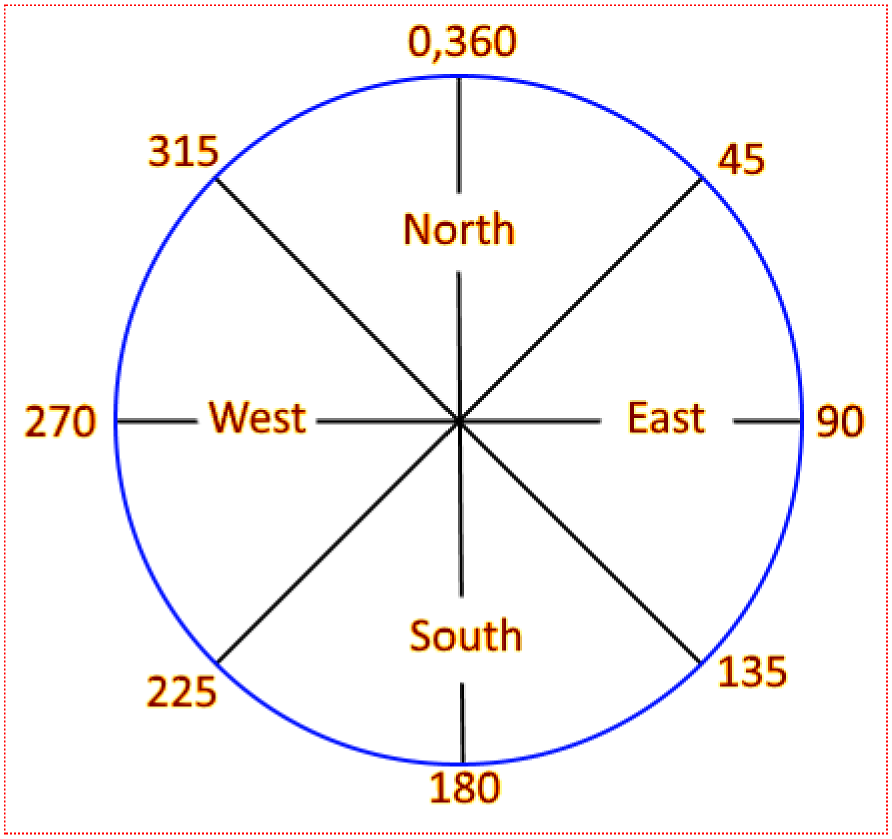


Fig 3: Compass angles and cardinal directions

Similar to ***windSpeedLevel***, we will use a ***for()*** to go through every value in the ***windSusDir*** column and use an ***if-else*** structure to set a direction for each value in the ***windDir*** column:

```
for(day in 1:nrow(weatherData))
{
  # if the direction is greater than 315 OR less than 45 degrees
  if(weatherData[day,"windSusDir"] >= 315 ||
     weatherData[day,"windSusDir"] < 45)
  {
    weatherData[day,"windDir"] = "North";
  }
  # if the direction is greater than 45 AND less than 135 degrees
  else if(weatherData[day,"windSusDir"] >= 45 &&
          weatherData[day,"windSusDir"] < 135)
  {
    weatherData[day,"windDir"] = "East";
  }
  # if the direction is greater than 135 AND less than 225 degrees
  else if(weatherData[day,"windSusDir"] >= 135 &&
          weatherData[day,"windSusDir"] < 225)
  {
    weatherData[day,"windDir"] = "South";
  }
  else # the direction is between 225 and 315 degrees
  {
    weatherData[day,"windDir"] = "West";
  }
}
```

The new ***windDir*** column is shown in *fig 4*.

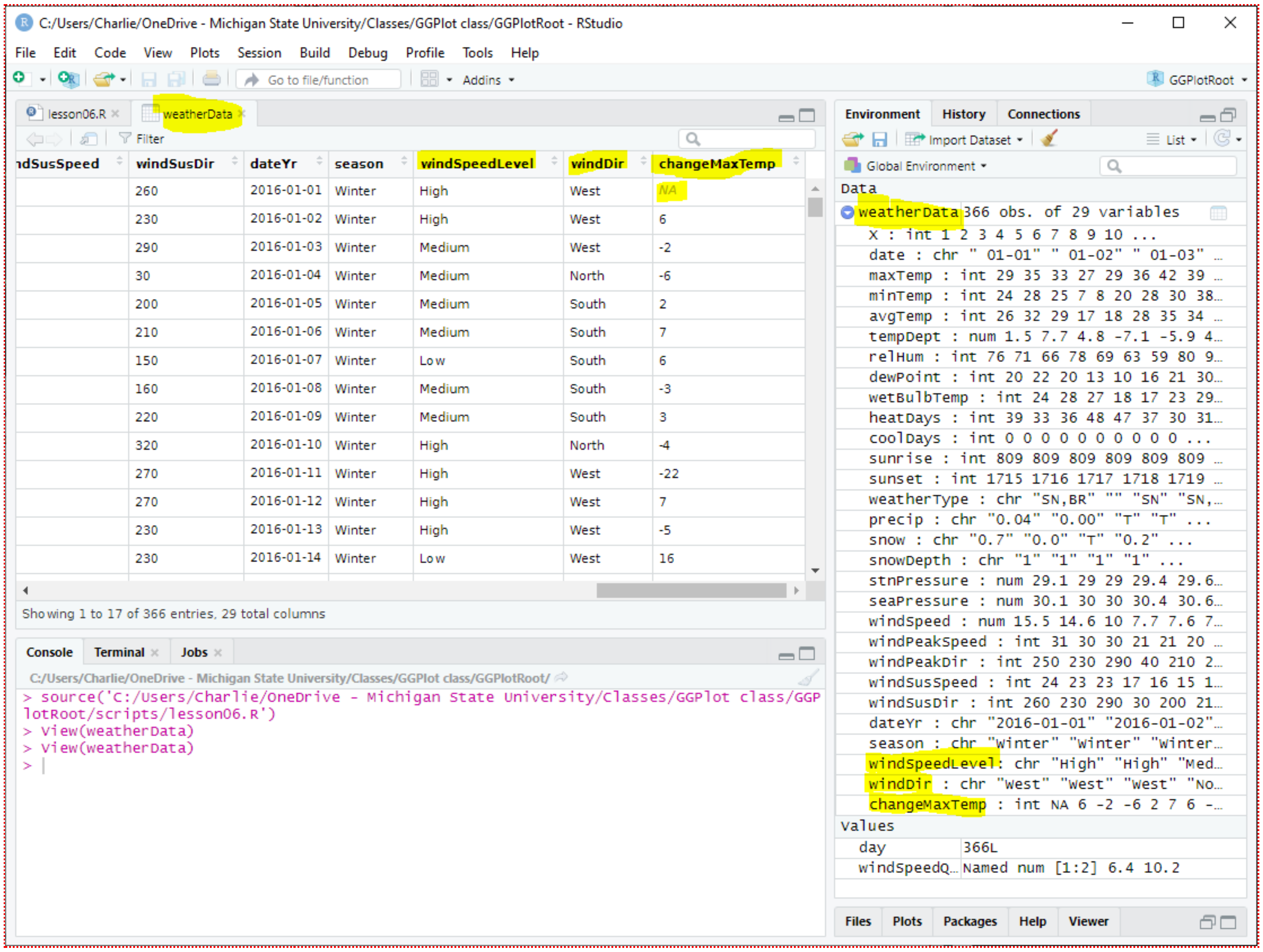
4.3 - Change in temperature

Finally, we are going to create a column that gives the difference between the current day's high temperature and the previous day's high temperature.

We need **366** values to complete the whole column, but there is no day previous to the first day on the column. So, there is no first value and we only have **365** difference values. To fix this issue, we set the first value to ***NA***.

Again, we are going to embed an ***if-else*** structure inside a ***for()*** that goes through all **366** rows in ***weatherData***. The first value is set to ***NA***, the other values are set to the different between the two days.

```
for(day in 1:nrow(weatherData))
{
  if( day == 1)  # no day before the first day
  {
    weatherData[day,"changeMaxTemp"] = NA;
  }
  else  # subtract previous day's maxTemp from current day's maxTemp
  {
    weatherData[day,"changeMaxTemp"] = weatherData[day,"maxTemp"] -
                                       weatherData[day-1,"maxTemp"];
  }
}
```


C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
GGPlotRoot
lesson06.R
weatherData
Filter
Environment History Connections
Import Dataset
List
Global Environment
Console Terminal Jobs
Files Plots Packages Help Viewer

ndSusSpeed	windSusDir	dateYr	season	windSpeedLevel	windDir	changeMaxTemp
	260	2016-01-01	Winter	High	West	NA
	230	2016-01-02	Winter	High	West	6
	290	2016-01-03	Winter	Medium	West	-2
	30	2016-01-04	Winter	Medium	North	-6
	200	2016-01-05	Winter	Medium	South	2
	210	2016-01-06	Winter	Medium	South	7
	150	2016-01-07	Winter	Low	South	6
	160	2016-01-08	Winter	Medium	South	-3
	220	2016-01-09	Winter	Medium	South	3
	320	2016-01-10	Winter	High	North	-4
	270	2016-01-11	Winter	High	West	-22
	270	2016-01-12	Winter	High	West	7
	230	2016-01-13	Winter	High	West	-5
	230	2016-01-14	Winter	Low	West	16

Showing 1 to 17 of 366 entries, 29 total columns

```
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGP
lotRoot/scripts/lesson06.R')
> View(weatherData)
> View(weatherData)
> |
```

```
Data
weatherData 366 obs. of 29 variables
  X : int 1 2 3 4 5 6 7 8 9 10 ...
  date : chr " 01-01" " 01-02" " 01-03" ...
  maxTemp : int 29 35 33 27 29 36 42 39 ...
  minTemp : int 24 28 25 7 8 20 28 30 38...
  avgTemp : int 26 32 29 17 18 28 35 34 ...
  tempDept : num 1.5 7.7 4.8 -7.1 -5.9 4...
  relHum : int 76 71 66 78 69 63 59 80 9...
  dewPoint : int 20 22 20 13 10 16 21 30...
  wetBulbTemp : int 24 28 27 18 17 23 29...
  heatDays : int 39 33 36 48 47 37 30 31...
  coolDays : int 0 0 0 0 0 0 0 0 0 0 ...
  sunrise : int 809 809 809 809 809 809 ...
  sunset : int 1715 1716 1717 1718 1719 ...
  weatherType : chr "SN,BR" "" "SN" "SN,...
  precip : chr "0.04" "0.00" "T" "T" ...
  snow : chr "0.7" "0.0" "T" "0.2" ...
  snowDepth : chr "1" "1" "1" "1" ...
  stnPressure : num 29.1 29 29 29.4 29.6...
  seaPressure : num 30.1 30 30 30.4 30.6...
  windSpeed : num 15.5 14.6 10 7.7 7.6 7...
  windPeakSpeed : int 31 30 30 21 21 20 ...
  windPeakDir : int 250 230 290 40 210 2...
  windSusSpeed : int 24 23 23 17 16 15 1...
  windSusDir : int 260 230 290 30 200 21...
  dateYr : chr "2016-01-01" "2016-01-02"...
  season : chr "Winter" "Winter" "Winter...
  windSpeedLevel: chr "High" "High" "Med...
  windDir : chr "West" "West" "West" "No...
  changeMaxTemp : int NA 6 -2 -6 2 7 6 -...
Values
  day          366L
  windSpeedQ…  Named num [1:2] 6.4 10.2
```

*Fig 4: The three new columns in **weatherData***

*Extension: one **for()** for all **if-else** structures*

4.4 - Saving the new dataframe

We are going to use the new dataframe in future lessons, so let's save it to the ***data*** folder as ***LansingNOAA2016-3.csv***:

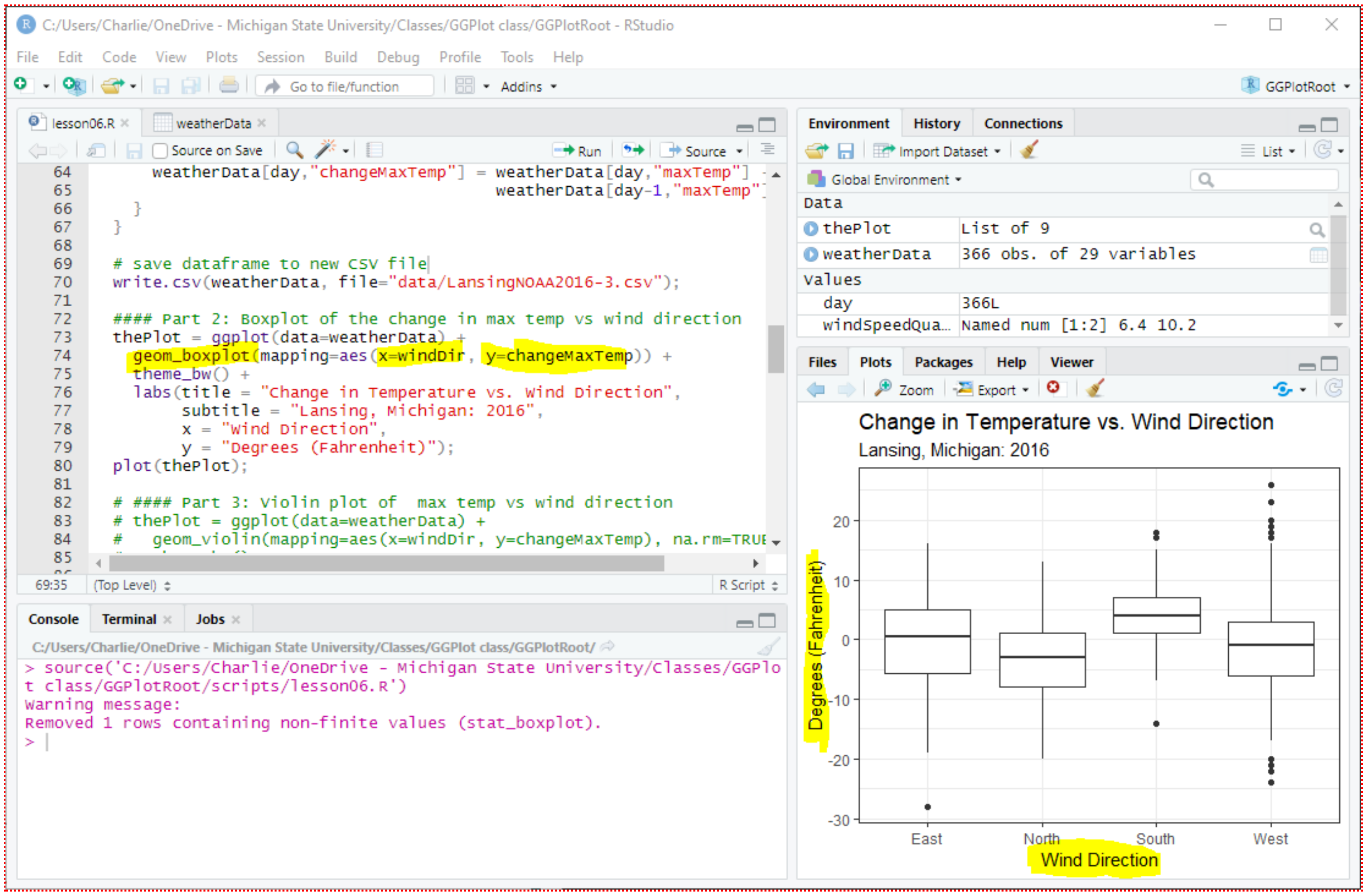
```
write.csv(weatherData, file="data/LansingNOAA2016-3.csv");
```

5 - Our first boxplot

The GGPlot component we use to create a boxplot is ***geom_boxplot()***. Our first boxplot will use two of the columns we just created: ***changeMaxTemp*** vs. ***windDir***.

```
#### Part 2: Plot Wind Speed vs. wind direction
thePlot = ggplot(data=weatherData) +
          geom_boxplot(mapping=aes(x=windDir, y=changeMaxTemp)) +
          theme_bw() +
          labs(title = "Change in Temperature vs. Wind Direction",
               subtitle = "Lansing, Michigan: 2016",
               x = "Wind Direction",
               y = "Degrees (Fahrenheit)");
plot(thePlot);
```

In *fig 5* we see that North winds seem to be associated with decreasing temperatures, and South winds seem to be associated with increasing temperatures. This makes sense as northerly winds often come from the Arctic and southerly winds often come from the Gulf of Mexico.



C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
GGPlotRoot
lesson06.R
weatherData
Source on Save
Run
Source
Environment History Connections
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
Zoom
Export
69:35
(Top Level)
R Script
Console Terminal Jobs
Change in Temperature vs. Wind Direction
Lansing, Michigan: 2016
Degrees (Fahrenheit)
20
10
0
-10
-20
-30
East
North
South
West
Wind Direction

```
    weatherData[day,"changeMaxTemp"] = weatherData[day,"maxTemp"] -
                                       weatherData[day-1,"maxTemp"]
  }
}

# save dataframe to new CSV file
write.csv(weatherData, file="data/LansingNOAA2016-3.csv");

#### Part 2: Boxplot of the change in max temp vs wind direction
thePlot = ggplot(data=weatherData) +
  geom_boxplot(mapping=aes(x=windDir, y=changeMaxTemp)) +
  theme_bw() +
  labs(title = "Change in Temperature vs. Wind Direction",
       subtitle = "Lansing, Michigan: 2016",
       x = "Wind Direction",
       y = "Degrees (Fahrenheit)");
plot(thePlot);

# #### Part 3: Violin plot of  max temp vs wind direction
# thePlot = ggplot(data=weatherData) +
#   geom_violin(mapping=aes(x=windDir, y=changeMaxTemp), na.rm=TRUE
```

Data	
thePlot	List of 9
weatherData	366 obs. of 29 variables
Values	
day	366L
windSpeedQua…	Named num [1:2] 6.4 10.2

```
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlo
t class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> 
```

Fig 5: Our first boxplot: change in temperature vs. wind direction

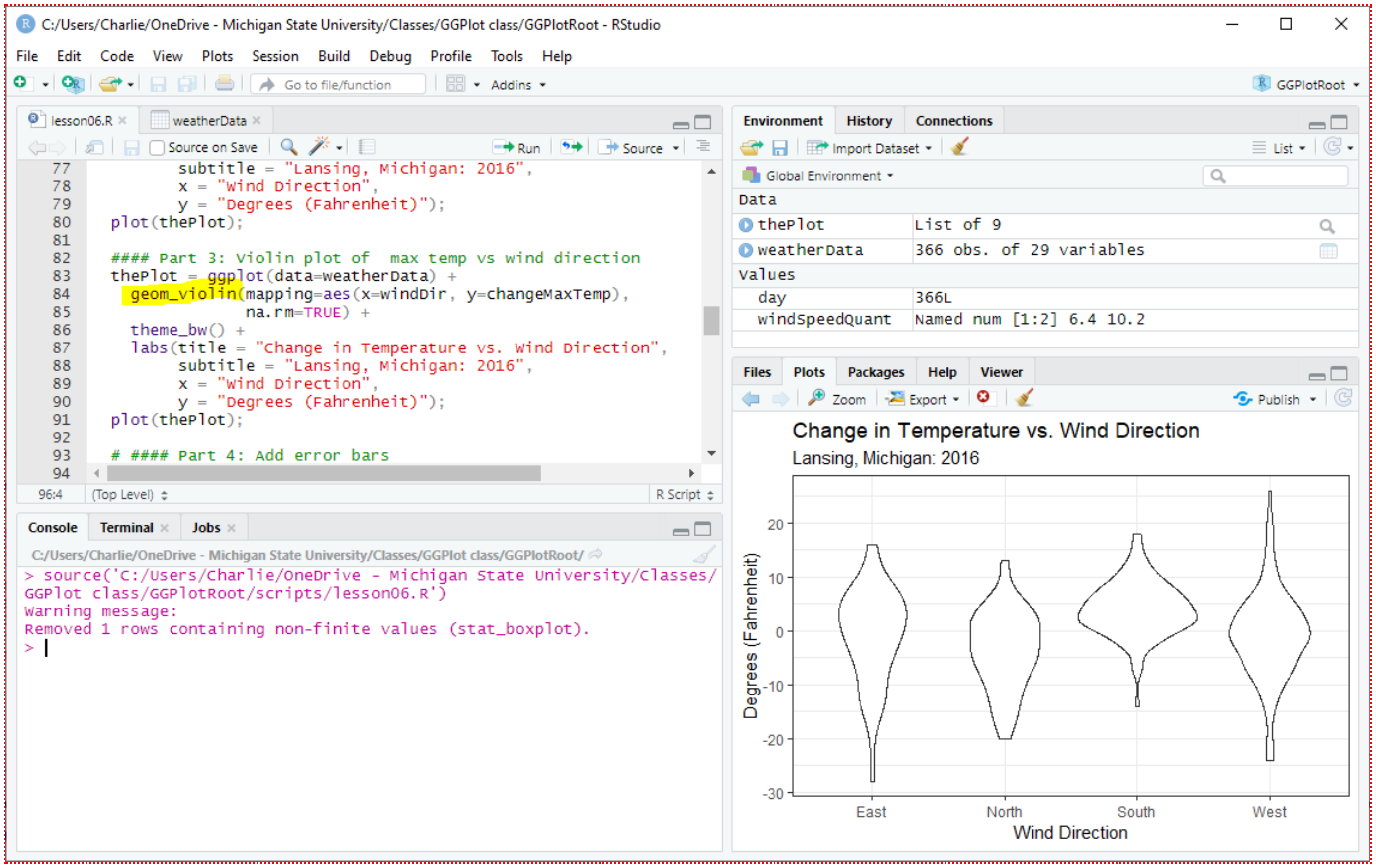
Note: The warning in fig 5 is appearing because there is an ***NA*** *value in the change in temperature data.*

5.1 - Variation: violin plot

One variation of a boxplot is the violin plot, which shows distribution instead of quantiles. The component ***geom_violin()*** plots a violin plot but, otherwise, works like a ***geom_boxplot().***

The parameter ***na.rm=TRUE*** was added to stop GGPlot from warning us every time that there is an **NA** value in the data (because the first value in ***changeMaxTemp*** is ***NA***). This will be added to all subsequent plots.

```
#### Part 3: Violin plot of Wind Speed vs. wind direction
thePlot = ggplot(data=weatherData) +
          geom_violin(mapping=aes(x=windDir, y=changeMaxTemp),
                      na.rm=TRUE) +
          theme_bw() +
          labs(title = "Change in Temperature vs. Wind Direction",
               subtitle = "Lansing, Michigan: 2016",
               x = "Wind Direction",
               y = "Degrees (Fahrenheit)");
plot(thePlot);
```



```
        subtitle = "Lansing, Michigan: 2016",
        x = "Wind Direction",
        y = "Degrees (Fahrenheit)");
plot(thePlot);

#### Part 3: Violin plot of  max temp vs wind direction
thePlot = ggplot(data=weatherData) +
  geom_violin(mapping=aes(x=windDir, y=changeMaxTemp),
              na.rm=TRUE) +
  theme_bw() +
  labs(title = "Change in Temperature vs. Wind Direction",
        subtitle = "Lansing, Michigan: 2016",
        x = "Wind Direction",
        y = "Degrees (Fahrenheit)");
plot(thePlot);

# #### Part 4: Add error bars
```

```
Data
thePlot          List of 9
weatherData      366 obs. of 29 variables
Values
day              366L
windSpeedQuant   Named num [1:2] 6.4 10.2
```

```
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/
GGPlot class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> 
```

Fig 6: Violin plot of change in temperature vs wind direction

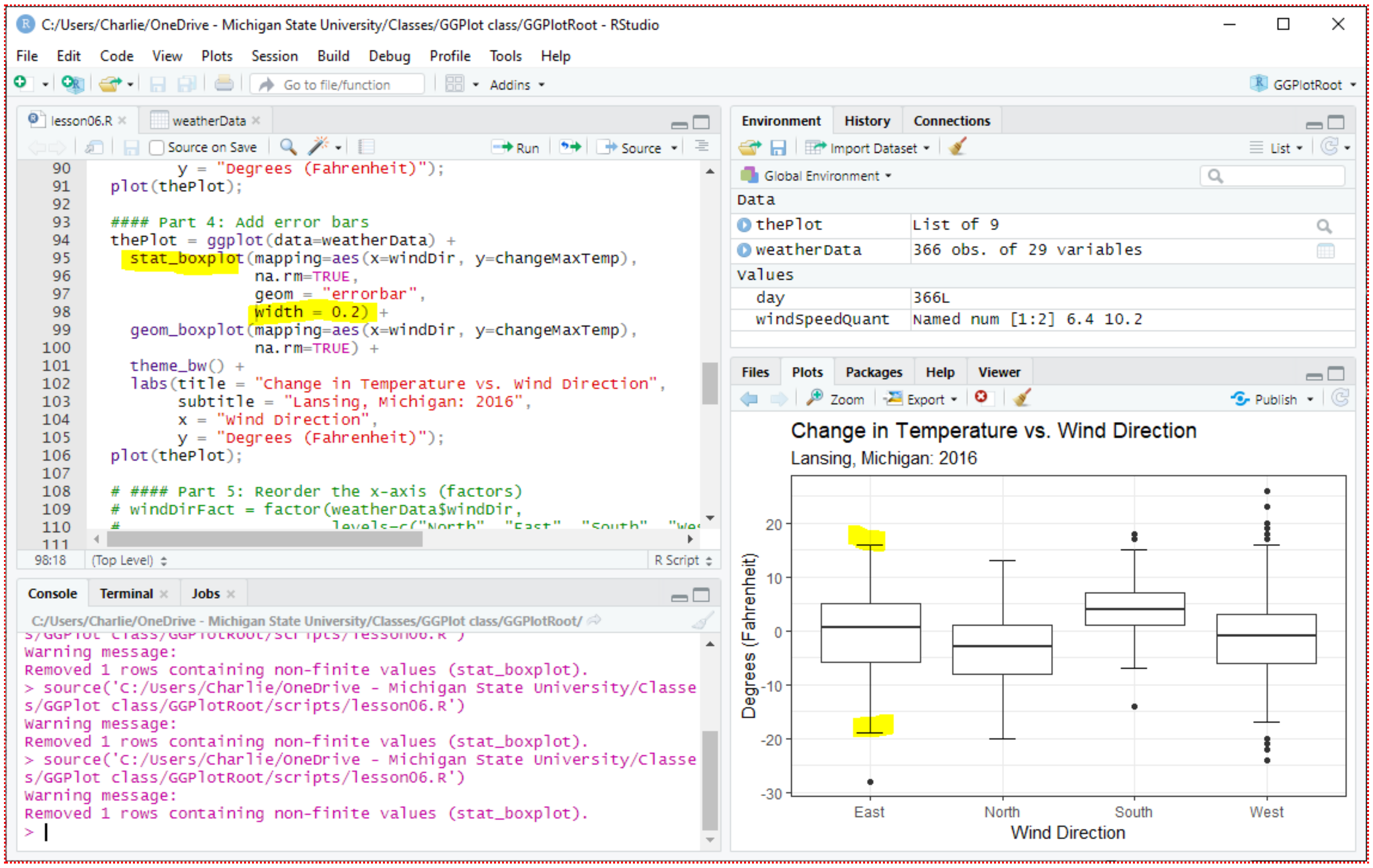
5.2 - Variation: Bars at the end of the whiskers

geom_boxplot() has no support for bars on the end of the boxplot whiskers. One somewhat hacky way to add bars is to create another plotting component on top of the boxplot. The component is ***stat_boxplot()***, which contains error bars.

In the following code, ***stat_boxplot()*** plots just the whiskers and error bars. The parameter ***width*** takes multipliers *values* with **1** meaning the error bar takes up the whole column (i.e., same width as the boxplot). So, ***width=0.2*** means the error bar takes up 20% of the column.

```
#### Part 4: Add error bars
thePlot = ggplot(data=weatherData) +
            stat_boxplot(mapping=aes(x=windDir, y=changeMaxTemp),
                         na.rm=TRUE,
                         geom = "errorbar",
                         width = 0.2) +
            geom_boxplot(mapping=aes(x=windDir, y=changeMaxTemp),
            geom_boxplot(mapping=aes(x=windDir, y=changeMaxTemp),
                         na.rm=TRUE) +
            theme_bw() +
            labs(title = "Change in Temperature vs. Wind Direction",
                 subtitle = "Lansing, Michigan: 2016",
                 x = "Wind Direction",
                 y = "Degrees (Fahrenheit)");
plot(thePlot);
```

*Note: **stat_boxplot()** is coded before and, hence, plotted behind **geom_boxplot()***



```
     y = "Degrees (Fahrenheit)");
plot(thePlot);

#### Part 4: Add error bars
thePlot = ggplot(data=weatherData) +
  stat_boxplot(mapping=aes(x=windDir, y=changeMaxTemp),
               na.rm=TRUE,
               geom = "errorbar",
               width = 0.2) +
  geom_boxplot(mapping=aes(x=windDir, y=changeMaxTemp),
               na.rm=TRUE) +
  theme_bw() +
  labs(title = "Change in Temperature vs. Wind Direction",
       subtitle = "Lansing, Michigan: 2016",
       x = "Wind Direction",
       y = "Degrees (Fahrenheit)");
plot(thePlot);

# #### Part 5: Reorder the x-axis (factors)
# windDirFact = factor(weatherData$windDir,
#                      levels=c("North", "East", "South", "We
```

```
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classe
s/GGPlot class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classe
s/GGPlot class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> 
```

```
Data
thePlot          List of 9
weatherData      366 obs. of 29 variables
Values
  day            366L
  windSpeedQuant Named num [1:2] 6.4 10.2
```

Fig 7: Adding error bars to a boxplot (there is currently no elegant solution for this)

6 - Reordering x-axis values

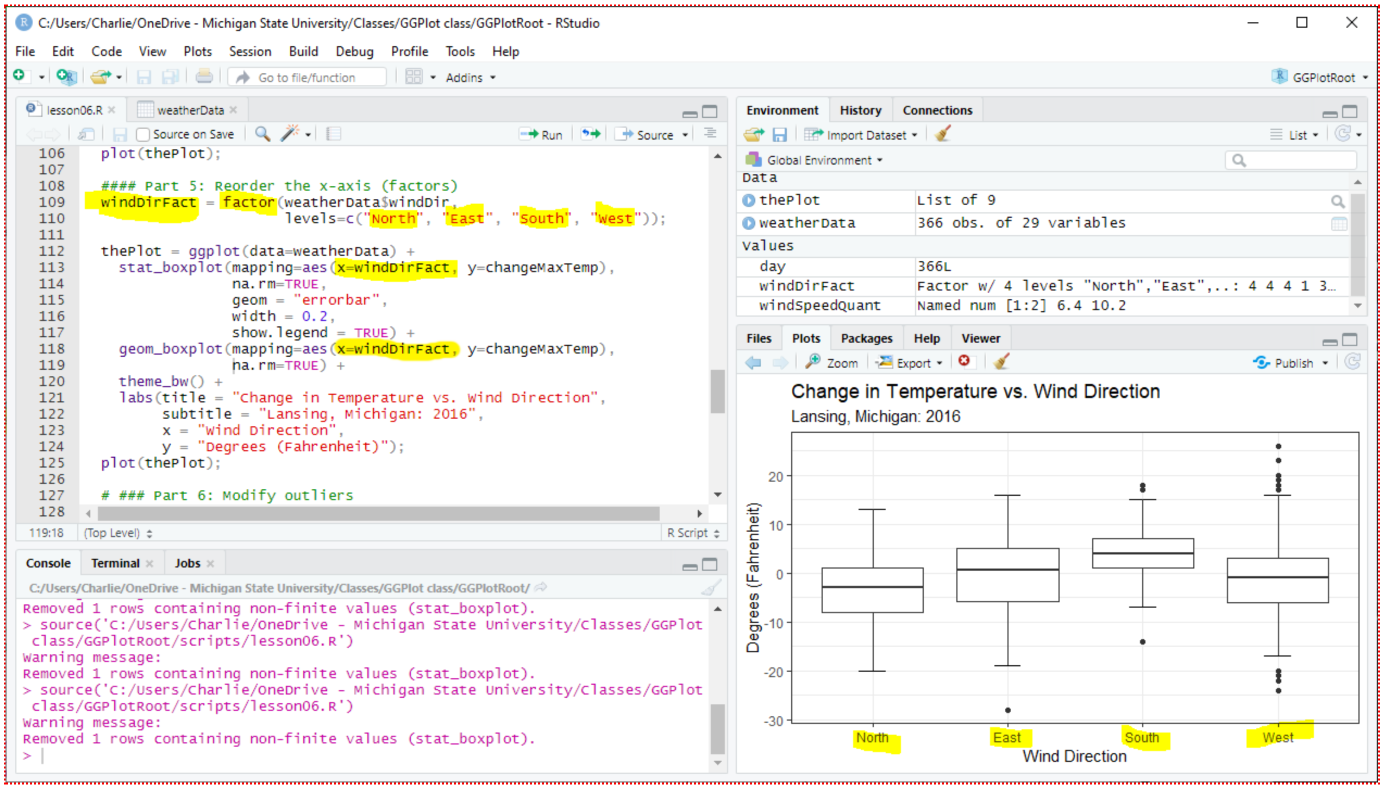
The x-axis (Wind Direction) labels go in alphabetical order: **East**, **North**, **South**, **West**. GGPlot will default to an alphabetical ordering if factor levels are not provided. We can force the issue by creating a factor, called ***windDirFact***, with the levels provided in the preferred order:

```
### Part 5: Re-order the directions on the x-axis using factor(s)
windDirFact = factor(weatherData$windDir,
                     levels=c("North", "East", "South", "West"));
```

Now we use the factor ***windDirFact*** in place of ***windDir*** -- just remember to replace ***windDir*** with ***winDirFact*** in both the ***stat_boxplot()*** and ***geom_boxplot()*** components.

```
thePlot = ggplot(data=weatherData) +
          stat_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                       na.rm=TRUE,
                       geom = "errorbar",
                       width = 0.2) +
          geom_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                       na.rm=TRUE) +
          theme_bw() +
          labs(title = "Change in Temperature vs. Wind Direction",
               subtitle = "Lansing, Michigan: 2016",
               x = "Wind Direction",
               y = "Degrees (Fahrenheit)");
plot(thePlot);
```

The boxplots are rearranged in order of the factor levels.



```
plot(thePlot);

#### Part 5: Reorder the x-axis (factors)
windDirFact = factor(weatherData$windDir,
                     levels=c("North", "East", "South", "West"));

thePlot = ggplot(data=weatherData) +
  stat_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
               na.rm=TRUE,
               geom = "errorbar",
               width = 0.2,
               show.legend = TRUE) +
  geom_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
               na.rm=TRUE) +
  theme_bw() +
  labs(title = "Change in Temperature vs. Wind Direction",
       subtitle = "Lansing, Michigan: 2016",
       x = "Wind Direction",
       y = "Degrees (Fahrenheit)");
plot(thePlot);

# ### Part 6: Modify outliers
```

```
Data
thePlot          List of 9
weatherData      366 obs. of 29 variables
Values
day              366L
windDirFact      Factor w/ 4 levels "North","East",..: 4 4 4 1 3…
windSpeedQuant   Named num [1:2] 6.4 10.2
```

```
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
Removed 1 rows containing non-finite values (stat_boxplot).
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> |
```

Fig 8: Reordering the categorical values on the x-axis

7 - Modifying Outliers

There are many parameters in the component ***geom_boxplot()*** that relate to outliers:

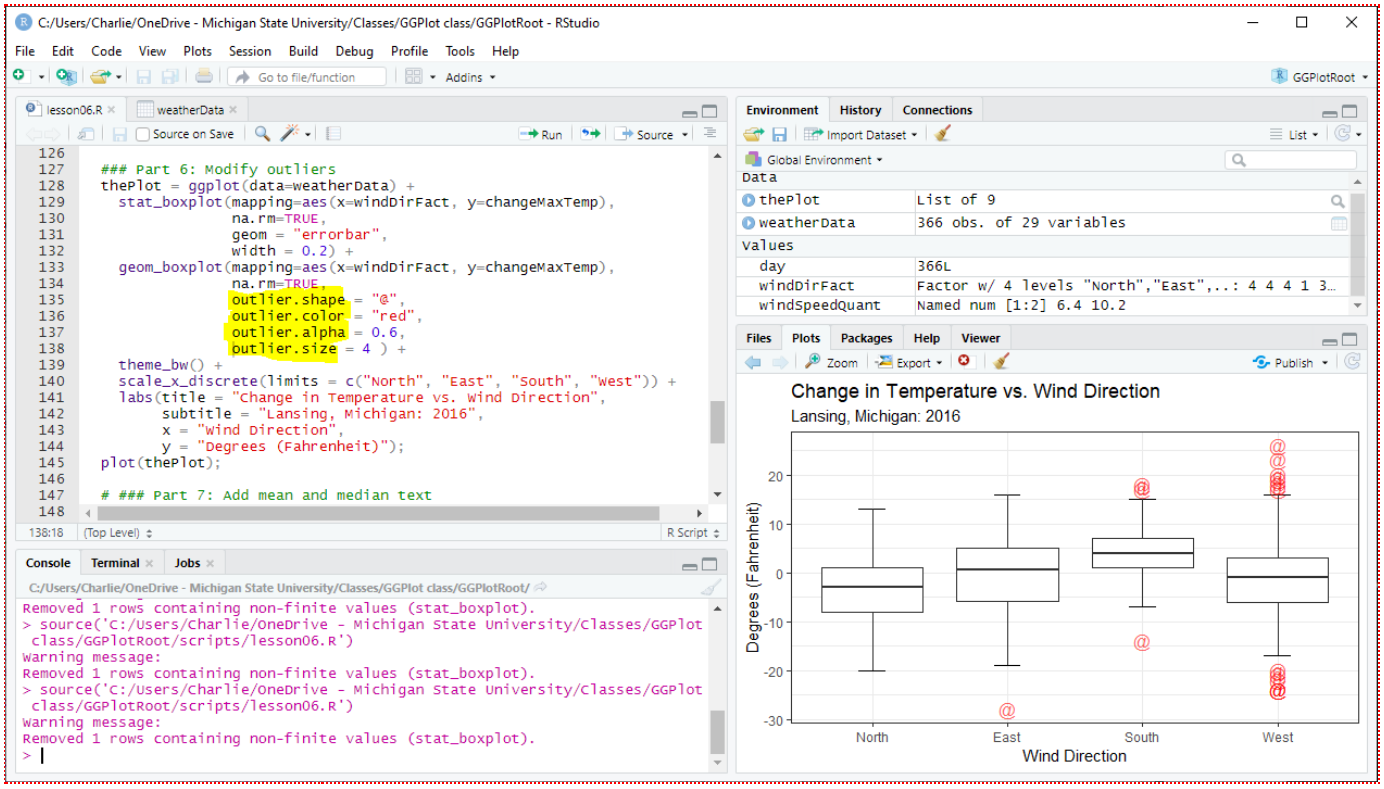
- ***outlier.shape***: the shape number (1-25) or character (in quotes) used to represent the outlier
  *note: **outlier.shape** = **NA** will remove the outliers from the plot*
- ***outlier.color***: outlier color

- ***outlier.size***: outlier size, as a numeric multiplier
- ***outlier.alpha***: outlier transparency (from **0**-invisible to **1**-opaque) -- good for situations where outliers overlap
- ***outlier.fill***: outlier fill color (only ***outlier.shape*** between **21**-**25** use fill color)
- ***outlier.stroke***: size of the border as a multiplier (only useful for shapes that have a fill -- otherwise, use ***outlier.size***)

In the following example, I change four of the outlier parameters

```
thePlot = ggplot(data=weatherData) +
          stat_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                       na.rm=TRUE,
                       geom = "errorbar",
                       width = 0.2) +
          geom_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                       na.rm=TRUE,
                       outlier.shape = "@",
                       outlier.color = "red",
                       outlier.alpha = 0.6,
                       outlier.size = 4 ) +
          theme_bw() +
          labs(title = "Change in Temperature vs. Wind Direction",
               subtitle = "Lansing, Michigan: 2016",
               x = "Wind Direction",
               y = "Degrees (Fahrenheit)");
plot(thePlot);
```

The outliers are now red, 60% opaque (40% transparent), 4mm tall, and shaped like an "@".



```
### Part 6: Modify outliers
thePlot = ggplot(data=weatherData) +
  stat_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
               na.rm=TRUE,
               geom = "errorbar",
               width = 0.2) +
  geom_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
               na.rm=TRUE,
               outlier.shape = "@",
               outlier.color = "red",
               outlier.alpha = 0.6,
               outlier.size = 4 ) +
  theme_bw() +
  scale_x_discrete(limits = c("North", "East", "South", "West")) +
  labs(title = "Change in Temperature vs. Wind Direction",
       subtitle = "Lansing, Michigan: 2016",
       x = "Wind Direction",
       y = "Degrees (Fahrenheit)");
plot(thePlot);

# ### Part 7: Add mean and median text
```

```
Removed 1 rows containing non-finite values (stat_boxplot).
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot
 class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot
 class/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
>
```

```
Data
thePlot          List of 9
weatherData      366 obs. of 29 variables
Values
day              366L
windDirFact      Factor w/ 4 levels "North","East",..: 4 4 4 1 3…
windSpeedQuant   Named num [1:2] 6.4 10.2
```

Fig 9: Modifying the shape, color, alpha, and size of the outliers

Note: the author recognizes the tacky nature of the outliers' styling -- this is being used for demo purposes. For the next plot, I will set ***outlier.shape*** to ***NA*** to remove the outliers.

8 - Adding text to the plot

We are going to add text to (i.e., annotate) the plot. The text will display the median **Change In Temperature** values for days where the winds come from the **North** and days where the winds come from the **South**.

To this we need to:

- Find the rows in ***weatherData*** that have **North** winds and the rows in ***weatherData*** that have **South** winds
- Find the median ***changeMaxTemp*** for rows that have **North** winds and rows that have **South** winds
- Add the median value to the plot using the ***annotate()*** component

8.1 - Find rows which() meet a condition

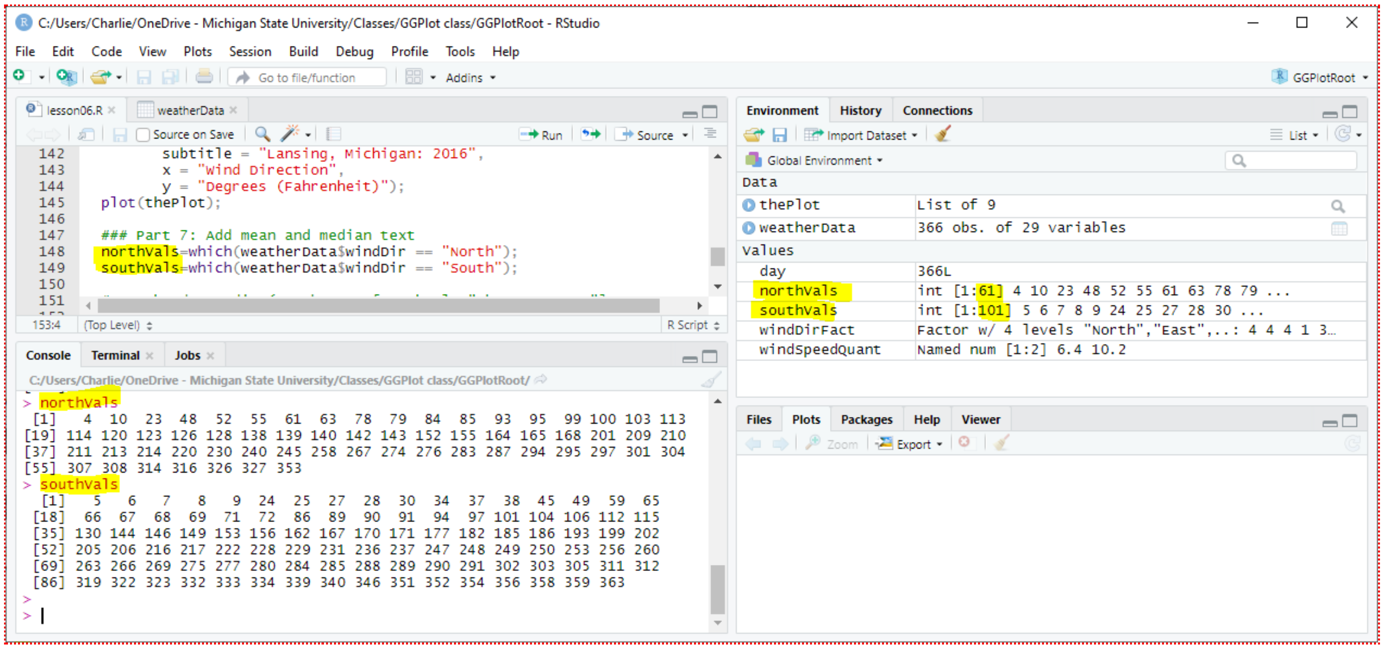
We use ***which()*** to find the indices of a vector that meet a condition. In this case the condition is: which values in ***windDir*** *are* equal to **North**. The index of the values are stored in ***northVals***.

```
northVals=which(weatherData$windDir == "North");
```

And which values in ***windDir*** *are* equal to **South**

```
southVals=which(weatherData$windDir == "South");
```

northVals and ***southVals*** are a vector of numbers. There are **61 *North*** values (starting with days 4,10, and 23) representing the 61 days that had northerly winds. There are **101 *South*** values (starting with days 5, 6, and 7) representing the 101 days that had southerly winds (*fig 10*).



```r
        subtitle = "Lansing, Michigan: 2016",
        x = "Wind Direction",
        y = "Degrees (Fahrenheit)");
plot(thePlot);

### Part 7: Add mean and median text
northVals=which(weatherData$windDir == "North");
southVals=which(weatherData$windDir == "South");
```

Data	
thePlot	List of 9
weatherData	366 obs. of 29 variables
Values	
day	366L
northVals	int [1:61] 4 10 23 48 52 55 61 63 78 79 ...
southVals	int [1:101] 5 6 7 8 9 24 25 27 28 30 ...
windDirFact	Factor w/ 4 levels "North","East",..: 4 4 4 1 3…
windSpeedQuant	Named num [1:2] 6.4 10.2

```
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
> northVals
 [1]   4  10  23  48  52  55  61  63  78  79  84  85  93  95  99 100 103 113
[19] 114 120 123 126 128 138 139 140 142 143 152 155 164 165 168 201 209 210
[37] 211 213 214 220 230 240 245 258 267 274 276 283 287 294 295 297 301 304
[55] 307 308 314 316 326 327 353
> southVals
  [1]   5   6   7   8   9  24  25  27  28  30  34  37  38  45  49  59  65
 [18]  66  67  68  69  71  72  86  89  90  91  94  97 101 104 106 112 115
 [35] 130 144 146 149 153 156 162 167 170 171 177 182 185 186 193 199 202
 [52] 205 206 216 217 222 228 229 231 236 237 247 248 249 250 253 256 260
 [69] 263 266 269 275 277 280 284 285 288 289 290 291 302 303 305 311 312
 [86] 319 322 323 332 333 334 339 340 346 351 352 354 356 358 359 363
> 
> |
```

*Fig 10: Finding the values in a vector that meet a condition using **which()***

8.2 - Find median value of indexed values

northVals is a vector with all the days (i.e., rows) that had northerly winds. ***southVals*** is a vector with all the days (i.e., rows) that had southerly winds.

We want a vector that contains the change in temperature, ***changeMaxTemp***, for the 61 days in ***northVals:***

```r
weatherData[northVals, "changeMaxTemp"]
```

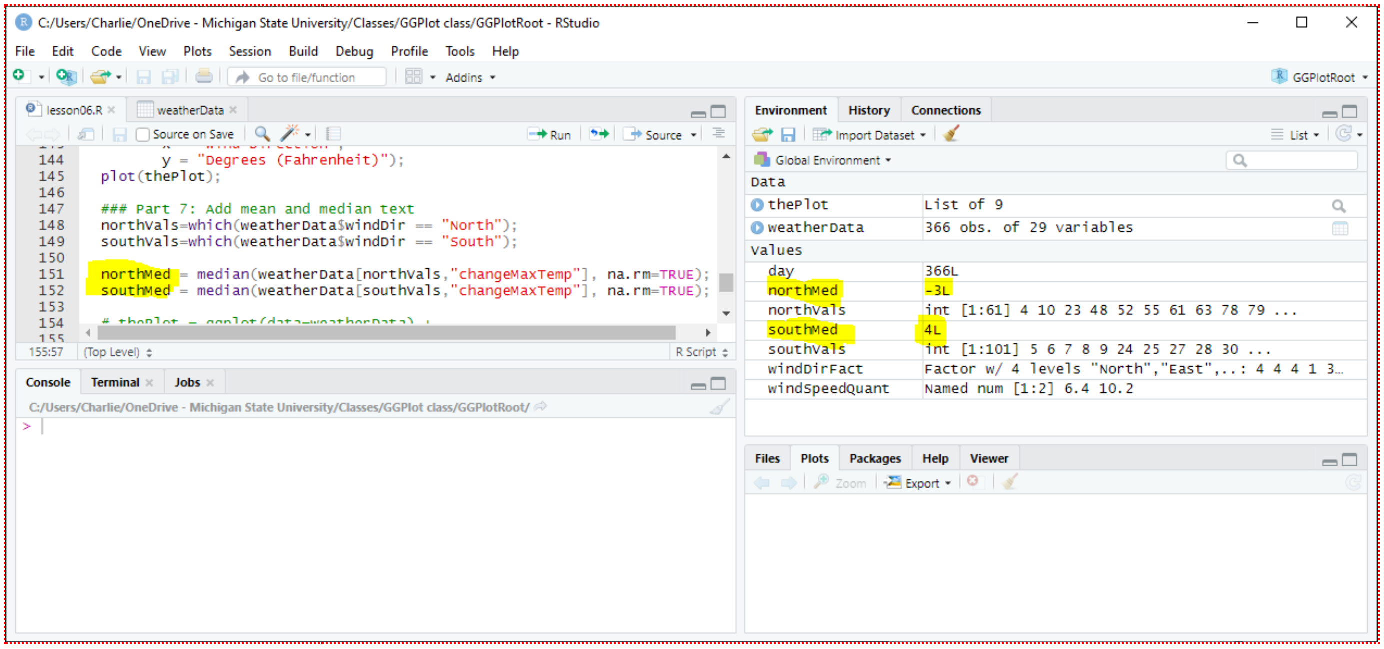
And a vector that contains ***changeMaxTemp*** values for the 101 days in ***southVals***:

```r
weatherData[southVals, "changeMaxTemp"]
```

To get the median of the indexed values above, we use ***median()*** and set ***na.rm = TRUE*** because we want to ignore the **NA** values in ***changeMaxTemp.***

In the following code, the median values for noth and south winds are saved to the variables ***northMed*** and ***southMed***.

```r
northMed = median(weatherData[northVals,"changeMaxTemp"], na.rm=TRUE);
southMed = median(weatherData[southVals,"changeMaxTemp"], na.rm=TRUE);
```



```
         y = "Degrees (Fahrenheit)");
plot(thePlot);

### Part 7: Add mean and median text
northVals=which(weatherData$windDir == "North");
southVals=which(weatherData$windDir == "South");

northMed = median(weatherData[northVals,"changeMaxTemp"], na.rm=TRUE);
southMed = median(weatherData[southVals,"changeMaxTemp"], na.rm=TRUE);

# thePlot = ggplot(data=weatherData) +
```

```
Data
thePlot          List of 9
weatherData      366 obs. of 29 variables
Values
  day            366L
  northMed       -3L
  northVals      int [1:61] 4 10 23 48 52 55 61 63 78 79 ...
  southMed       4L
  southVals      int [1:101] 5 6 7 8 9 24 25 27 28 30 ...
  windDirFact    Factor w/ 4 levels "North","East",..: 4 4 4 1 3…
  windSpeedQuant Named num [1:2] 6.4 10.2
```

*Fig 11: Median **changeMaxTemp** values for days with **North** and **South** winds*

8.3 - Annotate plot with median values

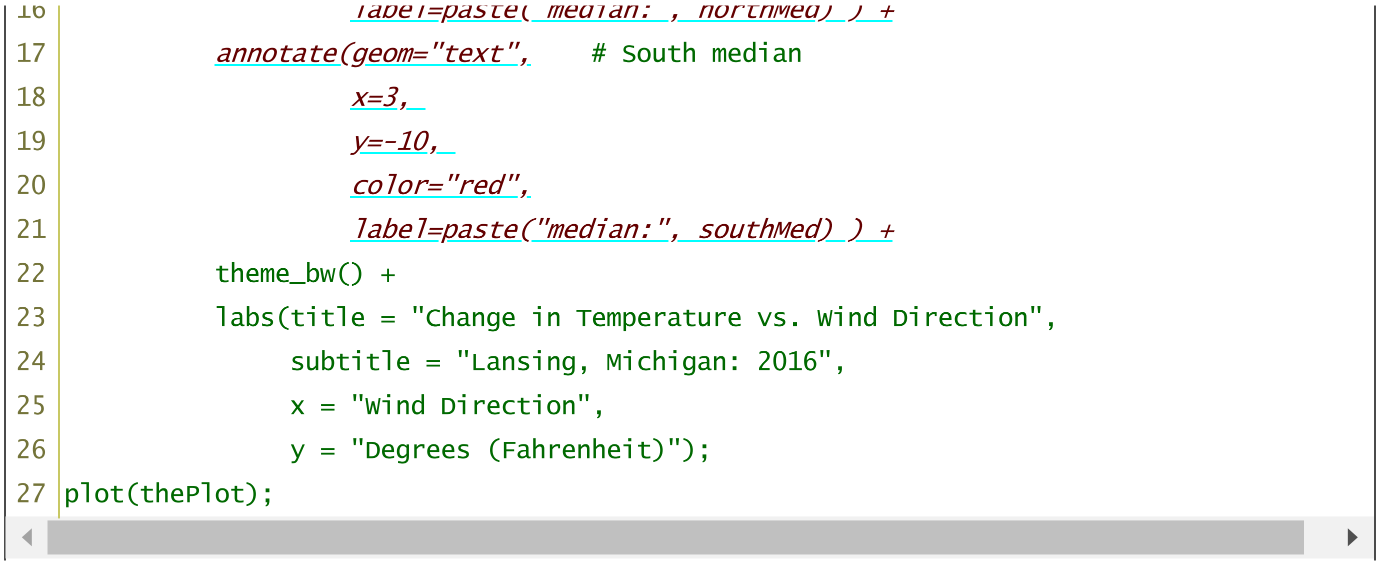
Now we just need to put the median values (***northMed*** and ***southMed***) on the plot using the ***annotate()*** component.

The subcomponents we set in ***annotate()*** are:

- **geom**: the type of value we are putting on the plot (in this case, **text**)
- **x**: x-position of the text -- for discrete axis, x=1 is the location of the first plot, x=2 is the second plot...
- **y**: y-position of the text -- for continuous plot, this value matches the value on the axis
- **color**: color of text
- **label**: the text on the plot -- using ***paste()***, we will put together the text "median:" and the variable that holds the calculated median value

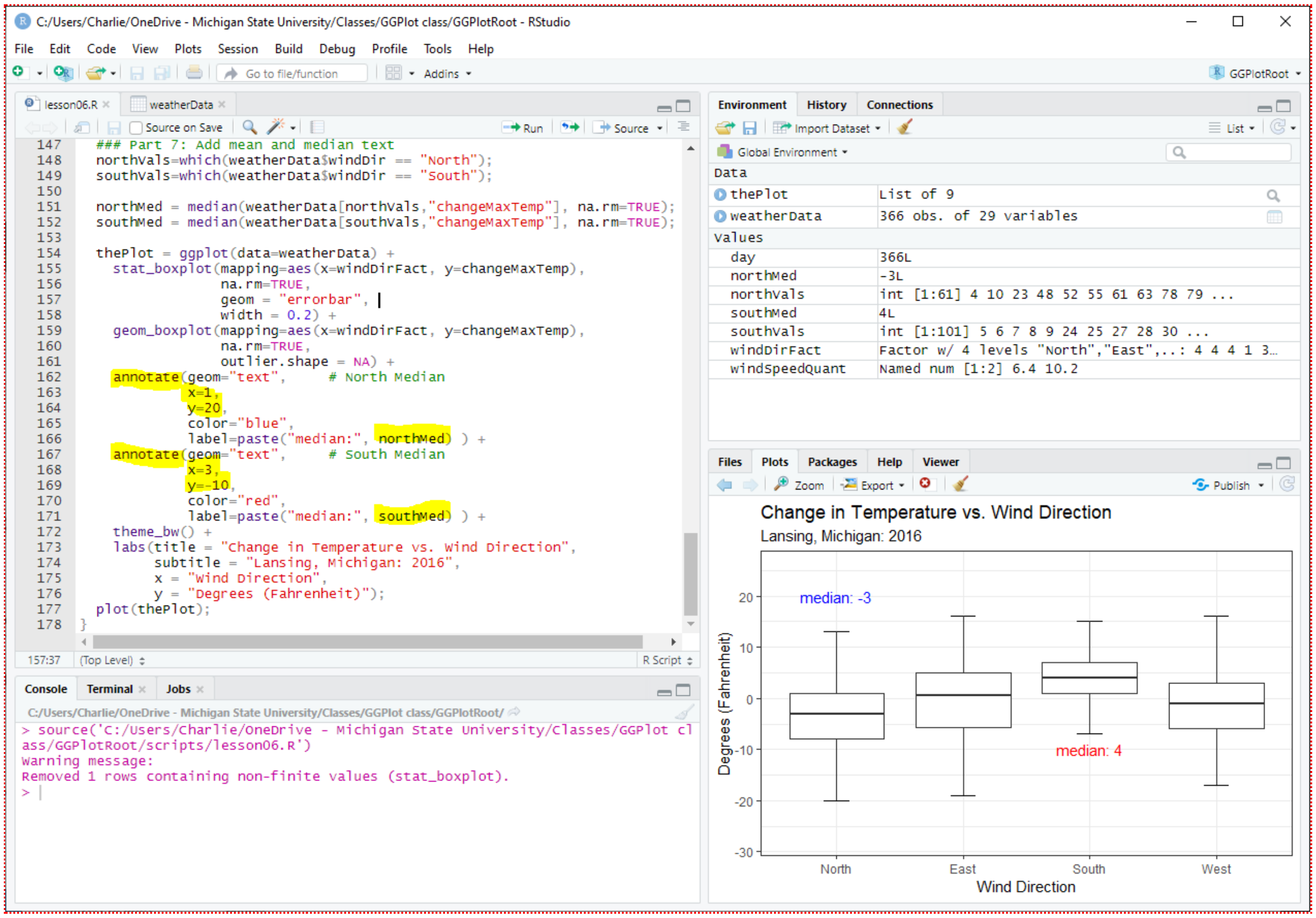
We are adding two ***annotate()*** components added to the plot -- one for **North**, the other for **South**

```
thePlot = ggplot(data=weatherData) +
          stat_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                       na.rm=TRUE,
                       geom = "errorbar",
                       width = 0.2) +
          geom_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                       na.rm=TRUE,
                       outlier.shape = "@",
                       outlier.color = "red",
                       outlier.alpha = 0.6,
                       outlier.size = 4 ) +
          annotate(geom="text",     # North median
                   x=1,
                   y=20,
                   color="blue",
                   label=paste("median:", northMed) ) +
```



```
                   label=paste("median:", northMed) ) +
          annotate(geom="text",     # South median
                   x=3,
                   y=-10,
                   color="red",
                   label=paste("median:", southMed) ) +
          theme_bw() +
          labs(title = "Change in Temperature vs. Wind Direction",
               subtitle = "Lansing, Michigan: 2016",
               x = "Wind Direction",
               y = "Degrees (Fahrenheit)");
plot(thePlot);
```

The text for the **North** median is on column **1** at the **20** degree line.
The text for the **South** median is on column **3** at the **-10** degree line.



C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
GGPlotRoot
lesson06.R
weatherData
Source on Save
Run
Source
157:37
(Top Level)
R Script
Environment History Connections
Import Dataset
List
Global Environment
Files Plots Packages Help Viewer
Zoom
Export
Publish
Console Terminal Jobs

```
  ### Part 7: Add mean and median text
  northVals=which(weatherData$windDir == "North");
  southVals=which(weatherData$windDir == "South");

  northMed = median(weatherData[northVals,"changeMaxTemp"], na.rm=TRUE);
  southMed = median(weatherData[southVals,"changeMaxTemp"], na.rm=TRUE);

  thePlot = ggplot(data=weatherData) +
    stat_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                 na.rm=TRUE,
                 geom = "errorbar",
                 width = 0.2) +
    geom_boxplot(mapping=aes(x=windDirFact, y=changeMaxTemp),
                 na.rm=TRUE,
                 outlier.shape = NA) +
    annotate(geom="text",     # North Median
             x=1,
             y=20,
             color="blue",
             label=paste("median:", northMed) ) +
    annotate(geom="text",     # South Median
             x=3,
             y=-10,
             color="red",
             label=paste("median:", southMed) ) +
    theme_bw() +
    labs(title = "Change in Temperature vs. Wind Direction",
         subtitle = "Lansing, Michigan: 2016",
         x = "Wind Direction",
         y = "Degrees (Fahrenheit)");
  plot(thePlot);
}
```

Data	
thePlot	List of 9
weatherData	366 obs. of 29 variables
Values	
day	366L
northMed	-3L
northVals	int [1:61] 4 10 23 48 52 55 61 63 78 79 ...
southMed	4L
southVals	int [1:101] 5 6 7 8 9 24 25 27 28 30 ...
windDirFact	Factor w/ 4 levels "North","East",..: 4 4 4 1 3…
windSpeedQuant	Named num [1:2] 6.4 10.2

```
C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot class/GGPlotRoot/
> source('C:/Users/Charlie/OneDrive - Michigan State University/Classes/GGPlot cl
ass/GGPlotRoot/scripts/lesson06.R')
Warning message:
Removed 1 rows containing non-finite values (stat_boxplot).
>
```

*Fig 12: Adding median values to the plot using **annotate()***

9 - Application

1. Create a script file in your GGPlot Class Project called ***app06.r***.
2. Create a vector that holds the quantile values for the 20, 40, 60, and 80th percentile of ***stnPressure***.

3. Create a new column called ***pressureLevel*** that creates five evenly spaced levels for ***stnPressure***
4. Make a boxplot of ***windSusSpeeds*** vs ***pressureLevel***
5. Find the dates for the three outliers at the lowest pressure level. The three outliers are also the three highest ***windSusSpeed*** for the year.
6. Label the outliers for the lowest pressure level with the dates on the plot.

10 - Extension: One for() for all if-else structures

The three ***for()*** used to create the three new column in ***weatherData*** (***windSpeedLevel***, ***windDir***, and ***changeMaxTemp***) could all be condensed into one ***for()*** because the three ***for()*** all execute over the same number of iterations, **366**.

```
for(day in 1:nrow(weatherData))
{
  ## Adding a column that gives relative wind speed for the day
  # Winds less than 6.4 miles/hour -- label as "Low"
  if(weatherData[day,"windSpeed"] <= windSpeedQuant[1])
  {
    weatherData[day,"windSpeedLevel"] = "Low";
  }
  # Winds greater than 10.2 miles/hour -- label as "High"
  else if(weatherData[day,"windSpeed"] >= windSpeedQuant[2])
  {
    weatherData[day,"windSpeedLevel"] = "High";
  }
  else # wind speeds between 6.4 and 10.2 miles/hour -- label as "Medium"
  {
    weatherData[day,"windSpeedLevel"] = "Medium";
  }

  ## Adding a column that gives the cardinal wind direction
  # if the direction is greater than 315 OR less than 45 degrees
  if(weatherData[day,"windSusDir"] >= 315 ||
  weatherData[day,"windSusDir"] < 45)
  {
    weatherData[day,"windDir"] = "North";
  }
  # if the direction is greater than 45 AND less than 135 degrees
  else if(weatherData[day,"windSusDir"] >= 45 &&
  weatherData[day,"windSusDir"] < 135)
  {
    weatherData[day,"windDir"] = "East";
  }
  # if the direction is greater than 135 AND less than 225 degrees
  else if(weatherData[day,"windSusDir"] >= 135 &&
          weatherData[day,"windSusDir"] < 225)
  {
    weatherData[day,"windDir"] = "South";
  }
  else # the directions is between 225 and 315 degrees
  {
    weatherData[day,"windDir"] = "West";
  }
  ### Adding a changeMaxTemp column
  if(day == 1)
  {
    weatherData[day,"changeMaxTemp"] = NA;
  }
  else
  {
    weatherData[day,"changeMaxTemp"] = weatherData[day,"maxTemp"] -
    weatherData[day-1,"maxTemp"];
  }
}
```