# Visual Presentation of Data using R

March 18, 2019

## 1 The Perfect Graphic

### 1.1 Best Practices

There is no such thing as the perfect graphic, but there are conventions that can be used to guide us to create accurtate, accessible, and visually pleasing grahics. But like many things, it takes some practices.

Here are some general rules:

- Be sure that you introduce the graphic/table with text – i.e. text first, then graphic.
- Cite the graphic/table with a figure or table number.
- Describe the graphic/table with a caption.
- Manage data range and transformations to effectively analyze and display the data.
- Make sure the axes are labeled with appropriate units
- Manage axis labels, value font size, and orientation to make them easy to reads.
- Avoid graphic titles unless you have more than one panel, i.e. graphics that are side by side or on top of each other.
- Do not connect data points with lines unless you can 'reasonable' interpolate between the points, e.g. a continous data set with some level of autocorrelation.
- Are the graphics accessible? For example, black and white can be better than color in terms of accessibility (universal design) and sustainability.
- Use the caption to describe what the reader is supposed to see in the figure.

## 1.2 How to Cite Software

In the text, students often make a bigger deal out the software than it deserses. Probably, because we feel like we climbed a big mountain to have some success and want demonstrate that. However, in general, environmental scientists downplay the software, unless they wrote a specific function or library.

Thus, for our purposes, the following is usually sufficient...
"Statistical analysis was conducted using using R (CRAN 2019)."

You don't need to mention how you imported it, used Rstudio, or talk about the functions. In the text, you might mention that you used a linear model, regression, analysis of variance (AOV), but the details of the R code is usually not mentioned.

However, you should also cite the program in your endnote:

To cite R in publications use:

R Core Team. 2017. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL `https://www.R-project.org/`.

You can get an updated version of the citation using the following function in R: citation(). Although the format isn't quite the same.

## 1.3 Reporting Results

In statistics, we reject the null hypothesis – and don't prove anything. So, we need to be careful how we report our results.

First, when we report statistics, do not report the p-value if the result in non-significant. Just report that null hypothesis cannot be rejected. For example, "there was no relationship between y and x". If the p-value is less then 0.001, just report it as '$p < 0.001$'. If the p-value is between 0.05 and 0.001, then I suggest you report the actual value (rounded to the nearest one-hundredth or thousandth as appropriate. For example, "evidence suggest that y depends on x ($p = 0.03$)."

Second, when we report the $r^2$ value, round to the nearest one-hundredth and describe it as the 'amount of variation explained by the model'. In fact, if you multiply by 100, it's can thought of as a percent variation explained.

Finally, avoid making the statistics the subject of the sentence! Describe the results as the subject/verb and then add a parenthetical about the stats, e.g. Water quality declined through the study period ($p = 0.03$; $r^2 = 0.09$), although little of the variation was explained by the model.

## 1.4 On Rounding and Decimaled Numbers

When numbers are less than 1 and reported as decimals, always report the leading zero, e.g. 0.02 instead of .02. It reduces ambiguities and helps ensure you haven't misreported the values with a typo.

We always want to round to the highest signficant figure. Thus, if temperatures are reported to the nearest degree, then can't report and average of

Table 1: Special Symbols for Rmarkdown and LaTeX

| Symbol | Markdown Code | LaTeXcode |
|---|---|---|
| $r^2$ | r^2^ | r$^2$ |
| $CO_2$ | CO<sub>2</\sub> | CO$_2$ |
| $\alpha$ | $\alpha$ | $\alpha$ |
| $\mu$ | $\mu$ | $\mu$ |
| ° | ° | $\degree$ |

4.00324432° C. For an average, we can report one decimal to the right of the reported instrument precision, thus 4.0° C would be appropriate. By reporting the following zero, we are explicit about the significant figures – thus is a reporting requirement.

### 1.5 Special Symbols

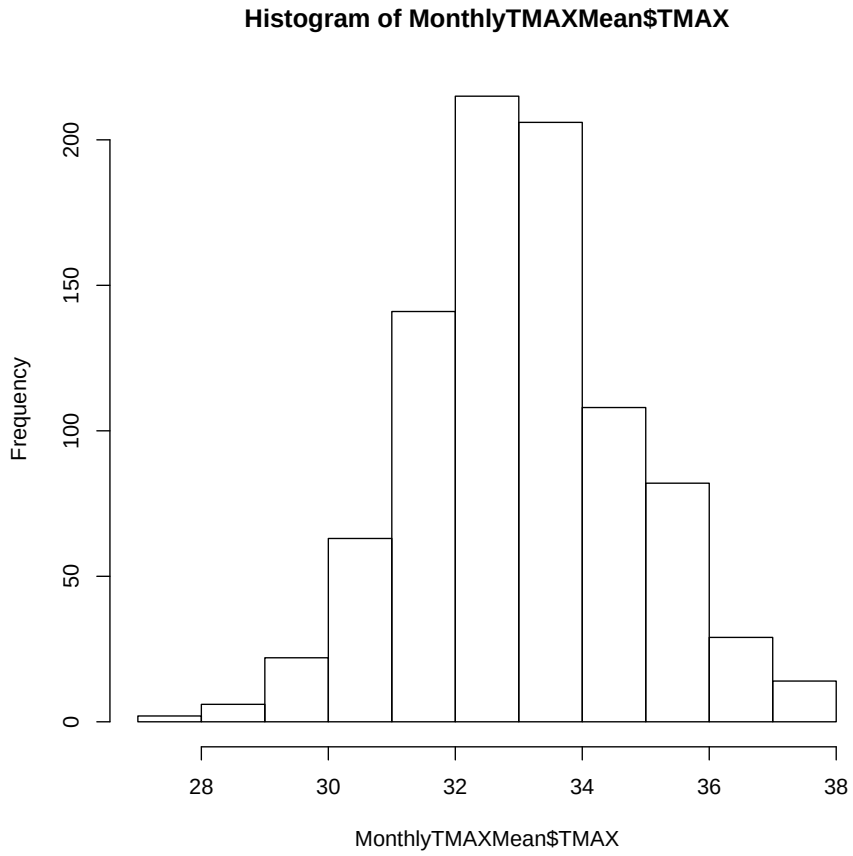
Science often relies on specialized characters, e.g. ° for degrees, $\alpha$ for our statistical test criteria. Table 1 shows the codes for symbols for Rmarkdown and LaTeX.

## 2 Exploring the Histogram

Data exploration can include many steps, but starting with a histogram gives the researcher the ability to evaluate the distribution of the data.

Below is a default histogram for TMAX values, where we might be able to visually how normally distributed the data might be.

```
hist(MonthlyTMAXMean$TMAX)
```




The default graphic is hideous – so, let's start fixing it.

## 2.1 Title and Axis Labels

For stand alone figures, we usually add titles, but in papers and lab reports it's a good practice to remove the title and use the caption to describe the graphic. Changes to the title can be made with arguments within the plot command, i.e. 'main=NULL'.

In addition, we can change the x-axis label, with the 'xlab' argument. Specifying the units is also required. And in this case, we want to add the °symbol and create a text string with the axis label in quotes that can be referenced in the hist() funtion. Notice that I rotated the y-axis values (las=1).

```
TMAXlabel <- "Maximum Temperature (ÂřC)"
hist(MonthlyTMAXMean$TMAX, main=NULL, xlab=TMAXlabel, las=1)
```

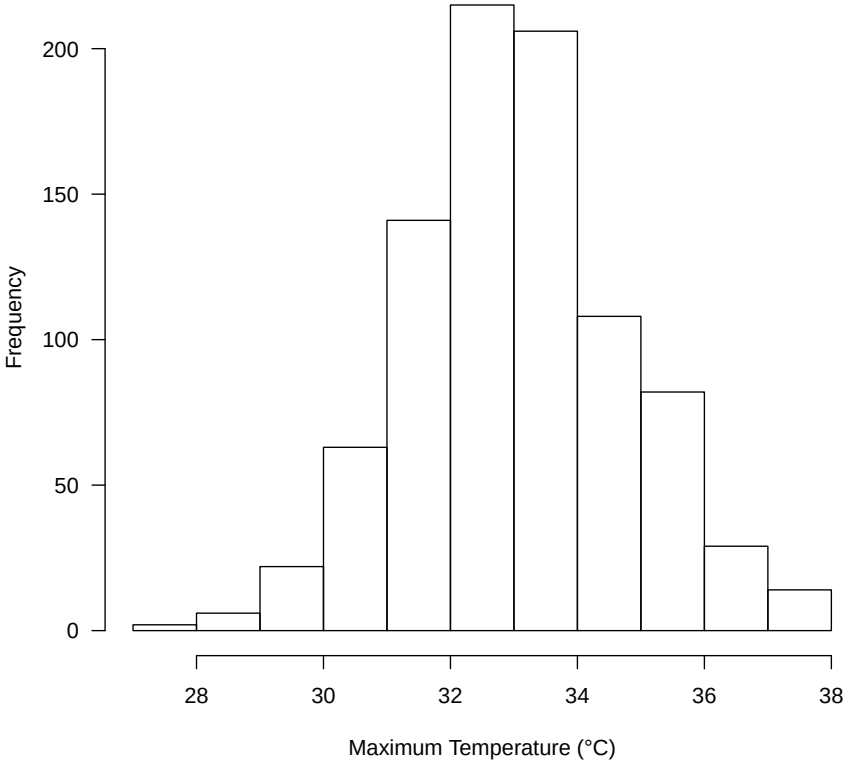



Figure 1: Histogram of Maximum Temperatures (ÂřC) (XXX, Thailand, 1940-2018)

## 2.2 Putting Multiple Figures in a Row

To create two graphics in one row, we can change the graphic parameters with the par() function. In this case, we'll create two column panels in one row using the 'mfrow' option and a vector that defines the number of rows and the number of columns, e.g. 'par(mfrow=c(1, 2))'. It's often a good idea to set the graphic parameter back to the default afterwards. In this case, I added a title because we have a panel with two graphics. Often people will put letters, e.g. A and B to refer to each one separately, but I prefer to put the actually description in the title, so the reader doesn't have to go back and forth between the caption and the figures.

Note: the title is too long! So, I will need to figure out how to adjust the size of the font at some point using 'cex' arguments. See the boxplots below to see how that can be done.

```
par(mfrow=c(1,2))
hist(MonthlyTMAXMean$TMAX, main='Maximum Temperature', xlab=TMAXlabel)
TMINlabel <- "Minimum Temperature (ÂřC)"
hist(MonthlyTMINMean$TMIN, main='Minimum Temperature', xlab=TMINlabel)
```

```
par(mfrow=c(1,1))
```

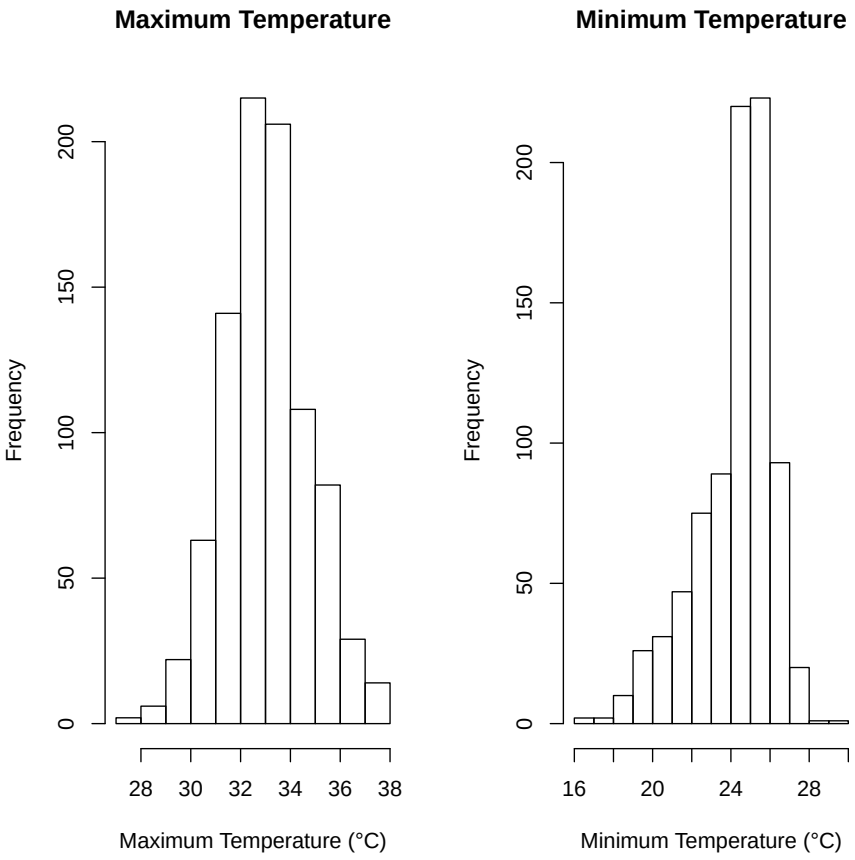



Figure 2: Mean monthly maximum and minimum tempetures (C) ((Bangkok, Thailand, 1940-2018))

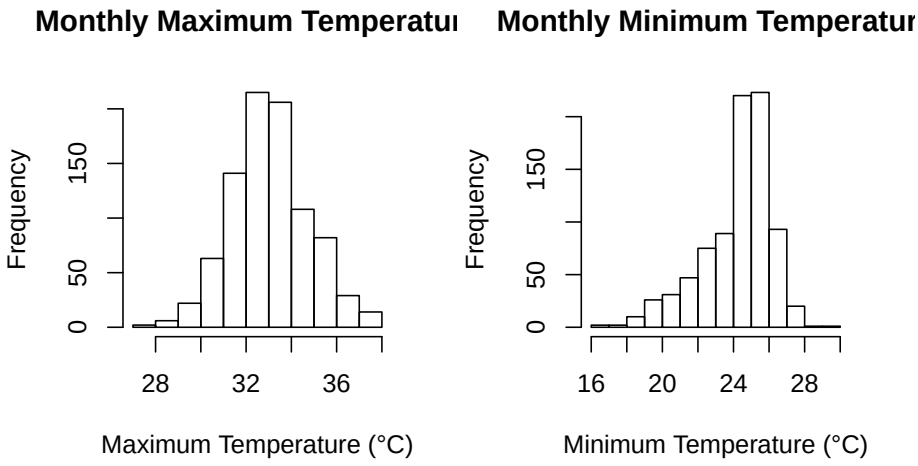



Figure 3: Mean monthly maximum and minimum tempetures (C) ((XXX, Thailand, 1940-2018)).

Because the figure is rather distorted, I have constrained the size using a fig.height and fig.width option.

```
par(mfrow=c(1,2))
hist(MonthlyTMAXMean$TMAX, main='Monthly Maximum Temperature', xlab=TMAXlabel)
TMINlabel <- "Minimum Temperature (ÂřC)"
hist(MonthlyTMINMean$TMIN, main='Monthly Minimum Temperature', xlab=TMINlabel)
```

```
par(mfrow=c(1,1))
```

Note that the title is too big and is cut off. In general, most publications prefer you put the description in the caption and avoid long titles. So, for the next example, I'll create a caption to refer to both figures as panel letters and get rid of the figure titles (main=NULL).

Altnernatively, I could change the relative size of the titles, using 'cex.main = 0.8', which multiplies the size by 0.8.

To create the panel letters, we'll use mtext for margin text.

```
par(mfrow=c(1,2))
hist(MonthlyTMAXMean$TMAX, main=NULL, xlab=TMAXlabel)
TMINlabel <- "Minimum Temperature (ÂřC)"
mtext("A", side=3, line=0, adj=0, cex=1.4)
hist(MonthlyTMINMean$TMIN, main=NULL, xlab=TMINlabel)
mtext("B", side=3, line=0, adj=0, cex=1.4)
```

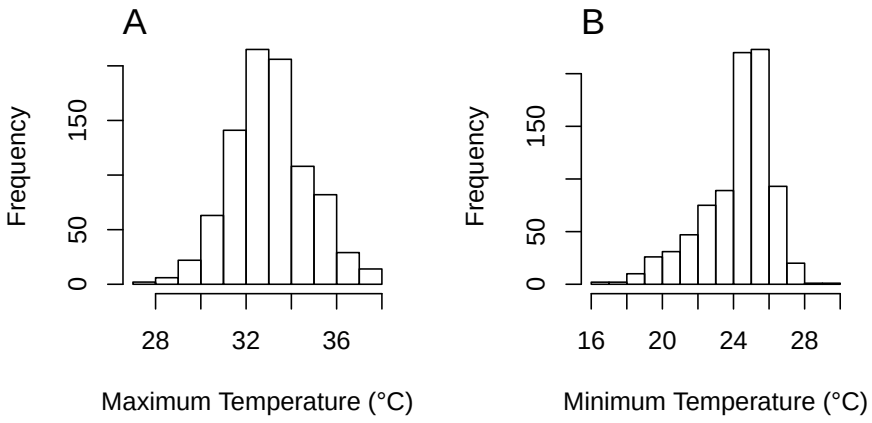



Figure 4: Mean monthly maximum (Panel A) and minimum (Panel B) tempetures (C) ((Bangkok, Thailand, 1940-2018)).

```
par(mfrow=c(1,1))
```

# 3 Boxplot

Box plots are great ways to display quantitative data from controlled experiments. For example, if you had high and low treatment categories and measured some response. Let's use the following example, ants colonies collected from three locations farm, grassland, and forest.

The boxplot can be improved dramatically by rotating the y-axis values ('las=1'), increasing the size of the axis labels ('cex.lab') and axis values ('cex.axis') (Figure 6.

```
boxplot(Colonies ~ Location, data=ants, las=1,
        ylab="No. of Colonies", xlab="Habitat", cex.lab=1.6,
        cex.axis=1.4, col="gray")
```

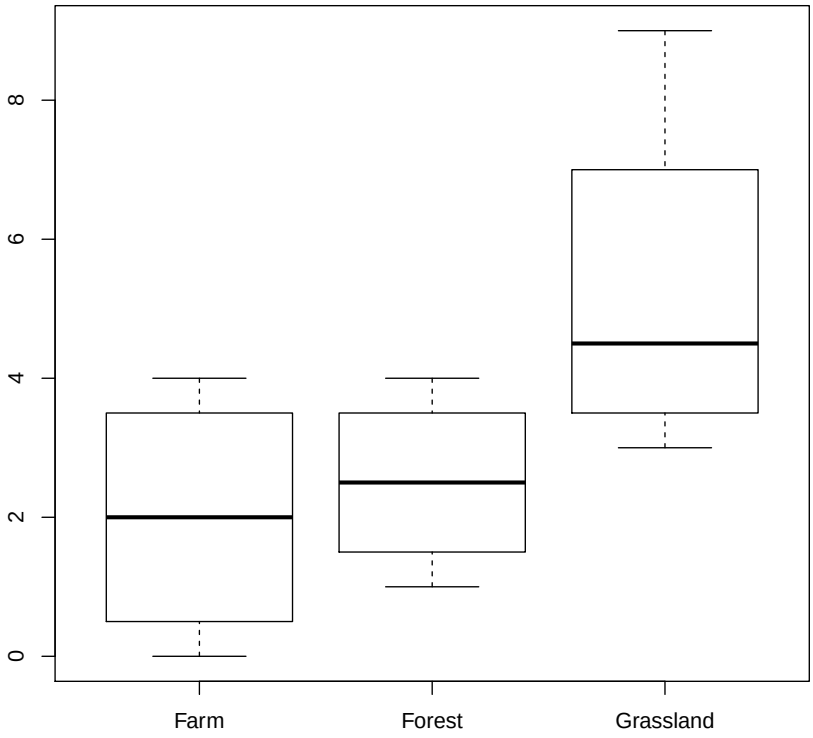



Figure 5: Number of ant colonies by habitat type.

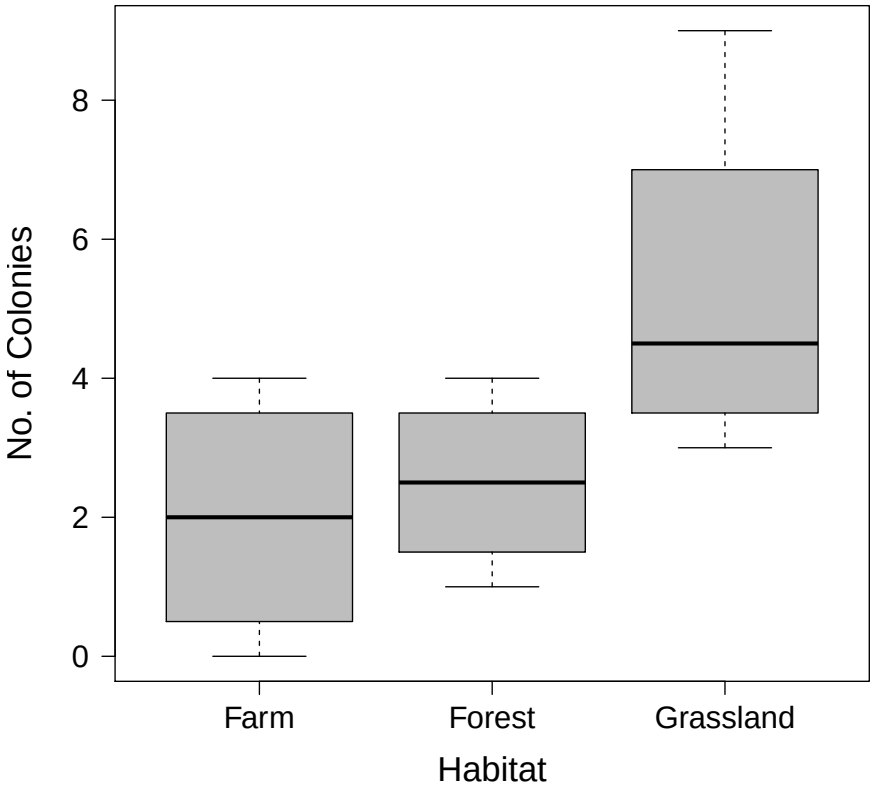



Figure 6: Number of ant colonies by habitat type.

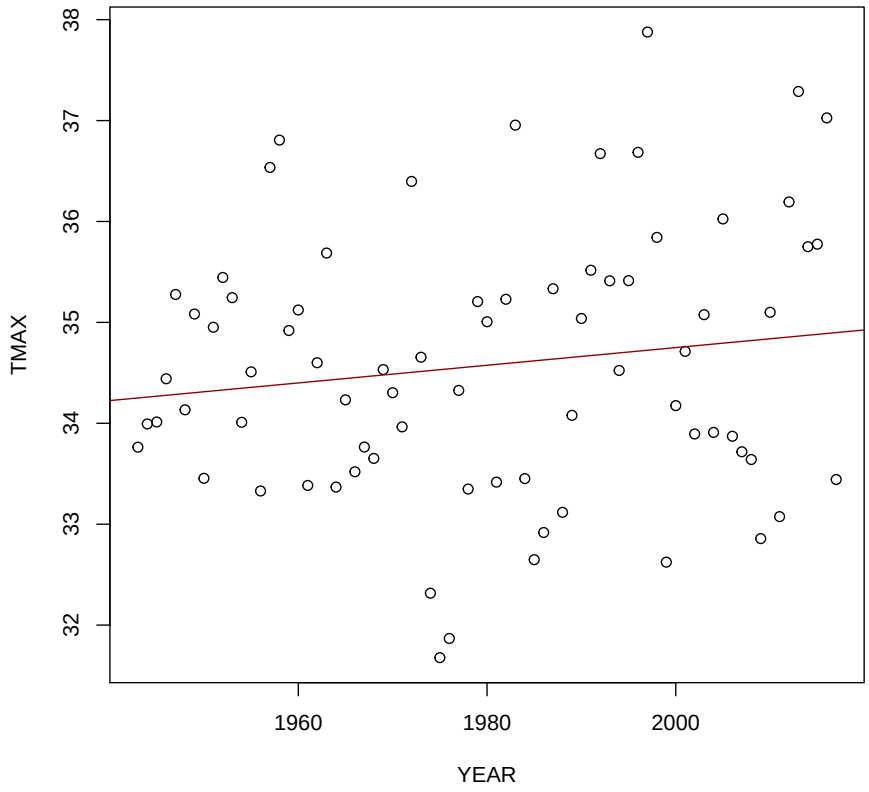
# 4 Scatter Plot – Non-time series

## 4.1 Scatter Plot – Time Series

Time series data sets often require special treatment. For example, data are often autocorrelated, thus data can be represented by lines instead of points. However, this requires some careful thought.

When we are graphing temperature data from one year to the next, we are averaging many days and connecting one year to the next might be appropriate if the data series is long enough, e.g. more than 50 years. However, with shorter time-series, i.e. 15 or less, connect the years with a line might be problematic.

```
plot(TMAX ~ YEAR, data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,])
abline(coef(lm(TMAX ~ YEAR, data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,])),
    col='darkred')
```




Let's fix the y-axis label as we did above (TMAX is not a very helpful label!). Furthermore, the x-axis needs to be calmed down some, so let's change the case

for these. We will also change the symbols to make it less busy with the 'pch' argument. You can look online to see the choices one has in R.

I am also not impress with the vertical orientation of the y-axis, so it's important to change these as well.

Finally, it's important that the image works in black and white. So, let's see if we can modify the graphic to make it less resource intensive. Finally, let's add a caption and reference to the figure (Figure 7).

```
ylabel <- "Maximum Temperature (ÂřC)"
plot(TMAX ~ YEAR, data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,],
     ylab=ylabel, xlab='Year', pch=20, las=1, col='gray')

abline(coef(lm(TMAX ~ YEAR,
    data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,])), col='black')
```

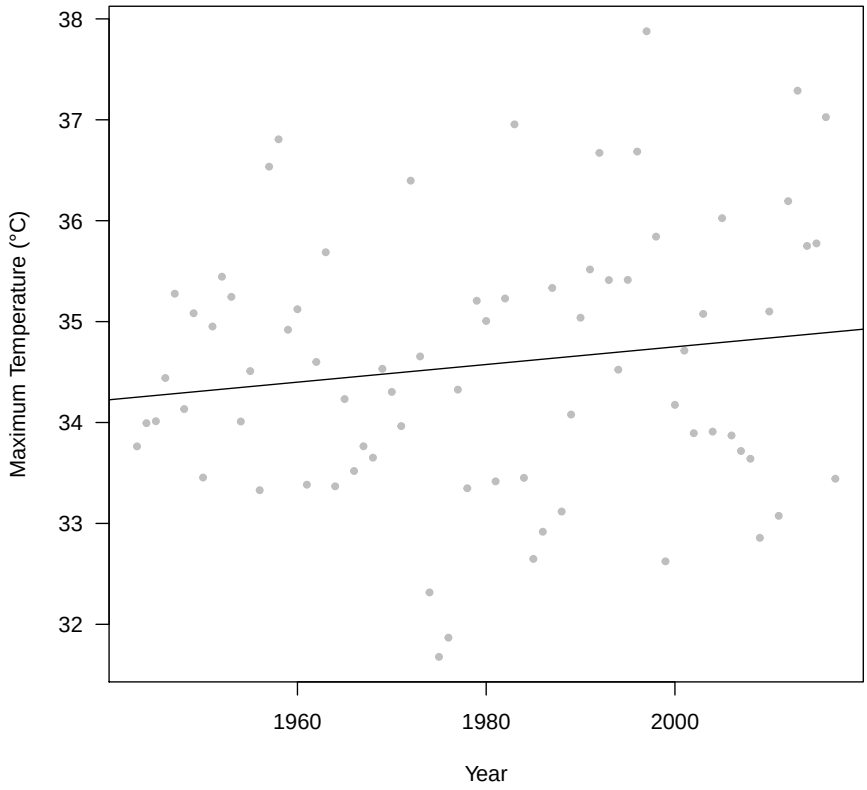



Figure 7: Monthly Average of Daily Maximum Temperatures (°C). Notice the slightly darker line in the x-axis for the middle section. I am not sure how to get rid of this, but it bugs me!

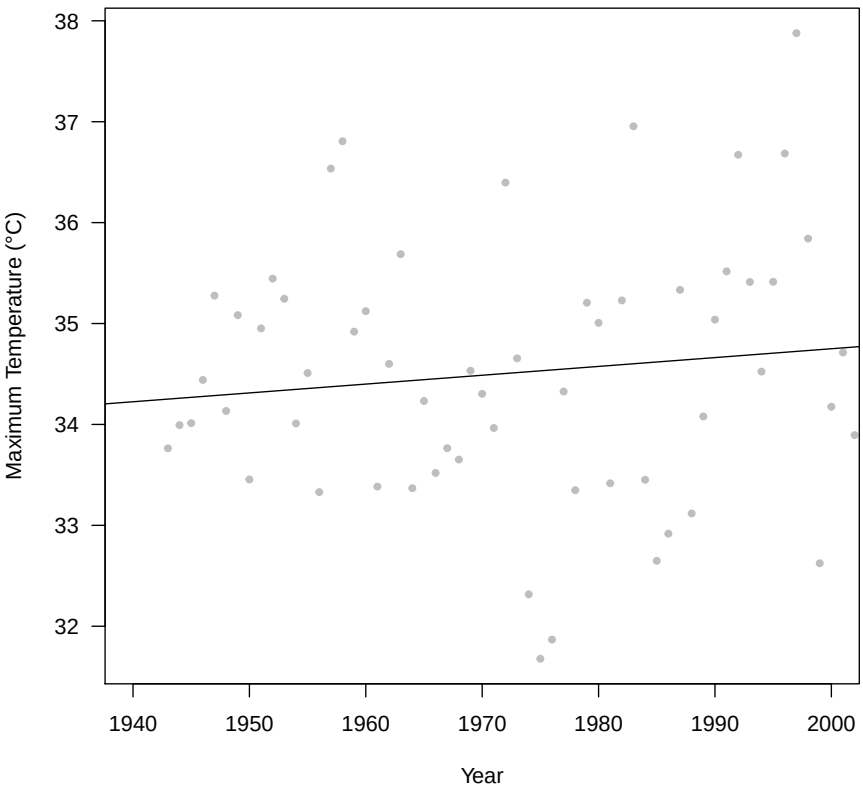



Figure 8: Monthly Average of Daily Maximum Temperatures (C).

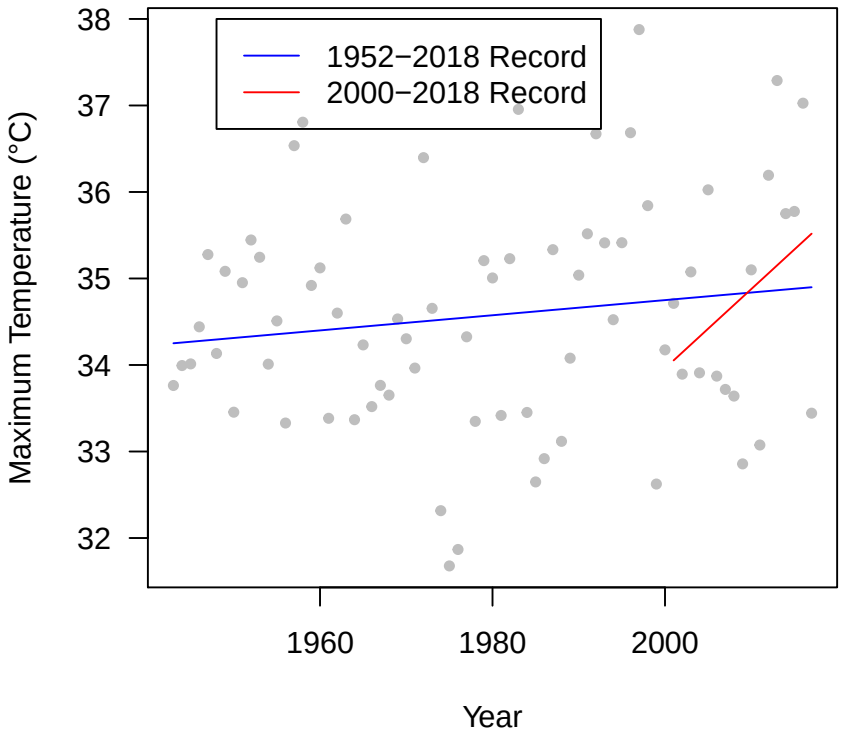
Now, what if we only want to display part of the data. We can limit the x-axis range using the 'xlim' argument, where we create a vector of for the start and end of the range. Finally, we can improve the plot by adding 'las=1' to rotate the y-axis labels.

```
ylabel <- "Maximum Temperature (ÂřC)"
plot(TMAX ~ YEAR, data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,],
     xlim=c(1940, 2000),
     ylab=ylabel, xlab='Year', pch=20, las=1, col='gray')

abline(coef(lm(TMAX ~ YEAR,
    data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,])), col='black')
```

Alternatively, you may want to creat a best fit line that only covers the range for the existing data without extrapolating, which is usually a very good idea for most scientific endeavors!

For example, we have seen several papers that select parts of the record to make dubious claims.

```
ylabel <- "Maximum Temperature (ÂřC)"
plot(TMAX ~ YEAR, data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,],
     #xlim=c(1940, 2000),
     ylab=ylabel, xlab='Year', pch=20, las=1, col='gray')

MonthlyTMAX.lm = (lm(TMAX ~ YEAR,
    data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5, ]))
interpolated = predict(MonthlyTMAX.lm,
      MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5,])

lines(MonthlyTMAXMean$YEAR[MonthlyTMAXMean$MONTH==5],
      interpolated, col='blue')

MonthlyTMAX.lm2 = (lm(TMAX ~ YEAR,
    data=MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5 &
            MonthlyTMAXMean$YEAR>2000, ]))

interpolated2 = predict(MonthlyTMAX.lm2,
            MonthlyTMAXMean[MonthlyTMAXMean$MONTH==5 &
            MonthlyTMAXMean$YEAR>2000,])

lines(MonthlyTMAXMean$YEAR[MonthlyTMAXMean$MONTH==5 &
            MonthlyTMAXMean$YEAR>2000], interpolated2, col='red')

legend(1948, 38, legend=c("1952-2018 Record", "2000-2018 Record"),
       lty=1, col=c("blue", "red"))
```


1952−2018 Record
2000−2018 Record
Maximum Temperature (°C)
38
37
36
35
34
33
32
1960
1980
2000
Year

# 5 Bar Graphs

## 5.1 What are Barcharts

Bar charts are used when you are graphing categorical data with continuous data that are a sum of something, for example dollars spent, no of events, and such. Rainfall is a good variable to put into barcharts with monthly or annual sums.

## 5.2 Introducing a Figure and Table

We introduce our figures and tables in the text before the appear in the text. Just like a good friend, we introduce them to new people so they don't have to try to initiate a conversation without any context. Thus, we provide a bit of context to help the reader with the figure.

One important note here – when you report the results of a figure or a table, do not use the table or figure as the subject of the sentence. Instead use the results as the subject and reference the table or figure as a parenthetical.

For example, 'The number of wildfires has dramatically increased after 2014 (Figure 9).

In my example, the figure should land after the text if possible.

This is preferable to 'Figure 9' shows that the number of wildfires has increased after 2014.' The emphasis is placed on the figure, which detracts to what the focus of the text should be on. Now there may be times when we want to put the focus on a table or complex figure, but usually we reserve that type of text for the figure caption.

```
fires = data.frame(Year = 2010:2018,
                   Fires = c(1, 0, 2, 0, 1, 13, 0, 16, 13))
# Note: barplot uses names.arg to assign the category names
barplot(fires$Fires, names.arg = fires$Year, las=1, ylab="No. of Fires")
```

## 5.3 When to use Barcharts

Rainfall data and other types of 'count' data are often best displayed as barcharts.

Notice how Figure 9 cuts out every-other year. You'll need to decide if the reader will be bothered by that. If so, you can reduce the x-axis labels fonts or increase the width of the chart.

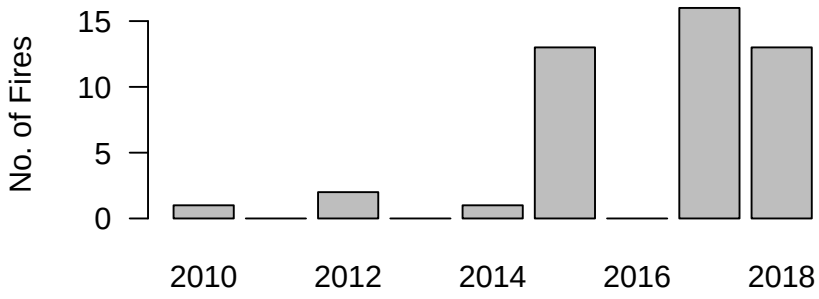



Figure 9: Number of Unsafe Air Quality Days in Washington State due to Fires.

# 6 Tables

Sometimes (often times?), tables are better than graphs. In the case below, I summarized the grades for each set of drafts using a table (Table 1) and through a series of histograms (Figure 10).

```
print(xtable(grades.df,
      caption='Table of Blog Draft Scores'), type='latex',
      caption.placement = "top",
      label="tab:scores")
```

| | Range | 1st Draft | 2nd Draft | 3nd Draft | 4th Draft | 5th Draft | Final |
|---|---|---|---|---|---|---|---|
| 1 | [0,10) | 4 | 0 | 0 | 0 | 0 | 0 |
| 2 | [10,20) | 0 | 3 | 0 | 0 | 0 | 1 |
| 3 | [20,30) | 7 | 2 | 1 | 0 | 0 | 3 |
| 4 | [30,40) | 4 | 7 | 2 | 0 | 0 | 3 |
| 5 | [40,50) | 0 | 1 | 7 | 4 | 1 | 8 |

The histogram is simple (Figure 10), but for these data, I am not sure it's all that useful. Which do you think is more effective?

The use of color in for this figure is really unnecessary. But it's worth illustrating. It's important that the use of color doesn't distract or confuse readers, especially those that are visually impaired, e.g. color blind.

You have 5 opportunities to re-submit to improve your grade, but only two weeks to accomplish this (Due: March 17, 2019). Since it takes two days to read them, you'll need to submit every 2 days if you want to take full advange of this opportunity. However, as I mentioned in an announcement, I am willing to provide the final feedback for submissions by the 17th and receive all submissions until the 22nd of March for the final grade.

```
par(mfrow=c(2,3), las=1)

colors <- c("darkgreen", "blue", "darkgoldenrod3", "violet", "darkred", "black")
labels <- c("Draft 1", "Draft 2", "Draft 3", "Draft 4", "Draft 5", "Final Blog")

hist(grades1, main=labels[1], xlab="Score",
     xlim=c(0,50), col=colors[1])
hist(grades2, main=labels[2], xlab="Score",
     xlim=c(0,50), col=colors[2])
hist(grades3, main=labels[3], xlab="Score",
     xlim=c(0,50), col=colors[3])
hist(grades4, main=labels[4], xlab="Score",
     xlim=c(0,50), col=colors[4])
hist(grades5, main=labels[5], xlab="Score",
     xlim=c(0,50), col=colors[5])
hist(grades6, main=labels[6], xlab="Score",
     xlim=c(0,50), col=colors[6])
```

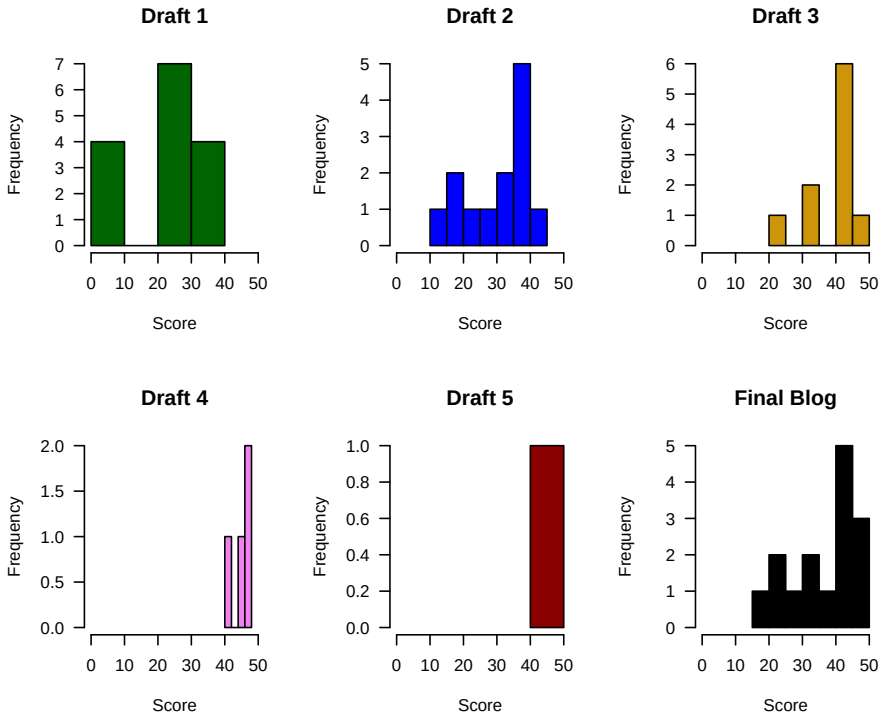



Figure 10: Histogram of Blog Scores

```
par(mfrow=c(1,1))
```

# 7 Probability Density

Another useful way to display data is by using probability densities (Figure 11). Based on the normal distribution, but modified for small N, the t-distribution can be used to test the probability of differences between each draft.

```
ttest = t.test(grades1, grades2, paired=F)
ttestp = t.test(grades1, grades2, paired=T)
```

The mean was for the first draft was 21.1 and the second draft was 31.4. We can test the null hypothesis that were was no difference between the two scores. Since the difference between the scores of the first and second submission was 10.3 points, we can reject the null hypothesis that the difference was zero using the t.test() function in R and obtain a p-value of 0.0237. We can also run the

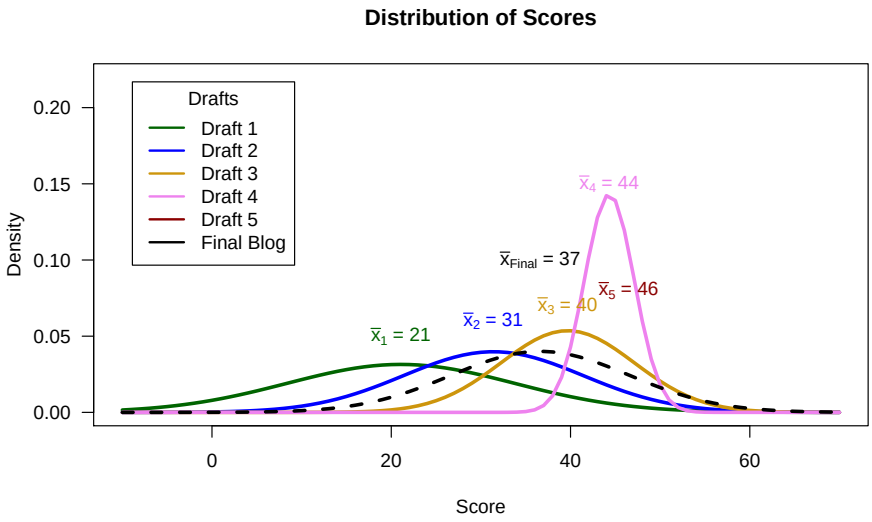



Figure 11: Probability Density for Scores

test as a pair test, where each student's first score is matched with their second score. This is useful when you have data from the same individuals; in this case the p-value is 0. However, once the probability is below 0.001, we report it as p $< 0.001$.

Notice how the probabilities go well beyond the scale possible, less than 0 and greater than 50. There are a number of ways of dealing with these issues, but that's beyond our scope. Suffice to say that a normal distribution is not appropriate for all data types.

Finally, please note how the scores changed when students took full advantage of the drafting process. With each draft the mean increased, but because not everyone took advantage of the drafting process, the final mean $\bar{x}_{Final}$ was quite abit lower than that mean for scores after Draft 3, 4, and 5.