Data Science and Statistical Modelling in space and time

Data Science and Statistical Modelling in space and time

Libraries

```
if(!require("geoR")) install.packages("geoR");
library(geoR)

if(!require("GGally")) install.packages("GGally");
library(GGally)
library(ggplot2)
library(dplyr)

if(!require("bestNormalize")) install.packages("bestNormalize");
library(bestNormalize)

if(!require("lubridate")) install.packages("lubridate");
library(lubridate)

if(!require("tidyr")) install.packages("tidyr");
library(tidyr)
library(stringr)

set.seed(42)
```

Section A: Spatial Modelling

Interpolating a set of sea surface temperature data for one month in the Kuroshio off Japan onto a grid with a resolution of .5° in both E and N directions > **Assumption:** Earth is flat

Data Loading

```
# Read data
data <- read.csv("kuroshio.csv")
gdata <- as.geodata(data, coords.col = 2:3, data.col = 6)
```

```
## as.geodata: 96 points removed due to NA in the data
## as.geodata: 130 replicated data locations found.
##  Consider using jitterDupCoords() for jittering replicated locations.
## WARNING: there are data at coincident or very closed locations, some of the geoR's functions may not
##  Use function dup.coords() to locate duplicated coordinates.
##  Consider using jitterDupCoords() for jittering replicated locations
```

```
# geoR can't handle different data values in the same position (What would such data tell us about)

# Find the duplicate data
dup <- dup.coords(gdata)
```

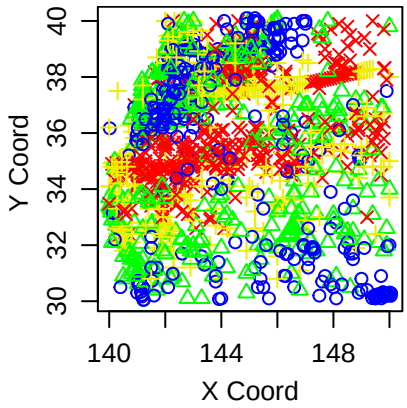
```
# Jitter the duplicate coordinates i.e. add a small random number to each x and y co-ordinate
gdata2 <- jitterDupCoords(gdata,max=0.1,min=0.05)
```

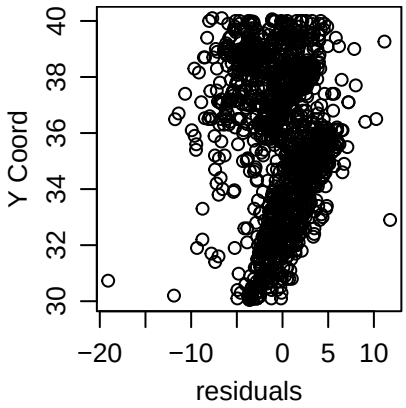
1. numerical and graphical summaries of the data.

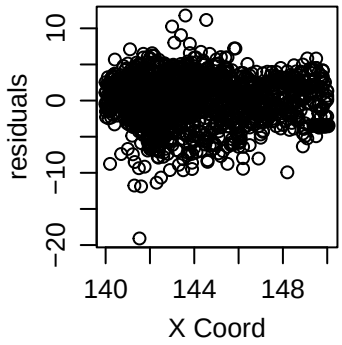
```
summary(gdata2)
```

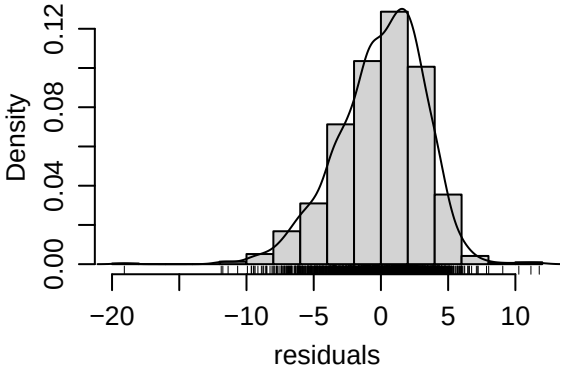
```
## Number of data points: 1550
##
## Coordinates summary
##          lon   lat
## min 139.9974 30.05
## max 150.0578 40.10
##
## Distance summary
##          min          max
##  0.004946679 13.366001646
##
## Data summary
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  0.00000 10.50000 14.00000 13.96465 18.30000 29.90000
##
## Other elements in the geodata object
## [1] "jitter.Random.seed"
```

```
plot(gdata2, trend="1st")
```









```
summary(data)
```

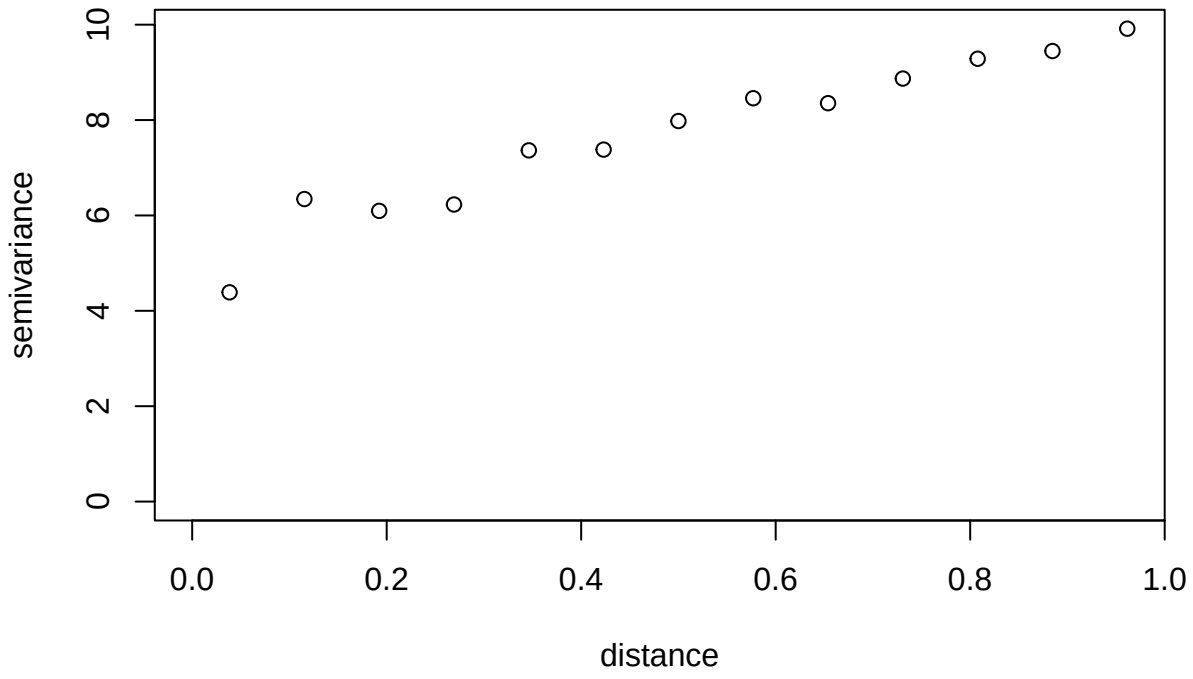
```
##      date                lon             lat             id
##  Length:1646        Min.   :140.0   Min.   :30.05   Length:1646
##  Class :character   1st Qu.:142.1   1st Qu.:33.90   Class :character
##  Mode  :character   Median :143.7   Median :36.10   Mode  :character
##                     Mean   :144.1   Mean   :35.77
##                     3rd Qu.:146.0   3rd Qu.:37.99
##                     Max.   :150.0   Max.   :40.10
##
##        pt              sst              sf              at
##  Min.   : 5.000   Min.   : 0.00   Min.   : 1.000   Min.   :-8.000
##  1st Qu.: 5.000   1st Qu.:10.50   1st Qu.: 1.000   1st Qu.: 5.400
##  Median : 5.000   Median :14.00   Median : 1.000   Median : 9.000
##  Mean   : 6.973   Mean   :13.96   Mean   : 1.942   Mean   : 8.854
##  3rd Qu.:12.000   3rd Qu.:18.30   3rd Qu.: 1.000   3rd Qu.:13.000
##  Max.   :12.000   Max.   :29.90   Max.   :15.000   Max.   :21.000
##                   NA's   :96                       NA's   :573
##        af
##  Min.   : 1.000
##  1st Qu.: 1.000
##  Median : 1.000
##  Mean   : 5.956
##  3rd Qu.:15.000
##  Max.   :15.000
##
```

2. Check Isotropy

```
isotropy <- variog(gdata2, max.dist=1)
```

```
## variog: computing omnidirectional variogram
```

```
plot(isotropy)
```



##

3. Fit Spatial Model

4. Fit by Bayesian methods

5. Differences between the two methods of estimation

B: Time Series Modelling

1. Which equation corresponds to which plot

2. Appropriate ARMA model for the five series

3.

The data used, `overturning.csv`, are the measured strength of the overturning in the North Atlantic from moorings at 26N between April 2004 and march 2014.

```
overturning <- read.csv("overturning.csv")
```

a. averaging the data to quaterly means

b. Fitting ARMA and ARIMA model to the data

c. Fitting DLM to the data including both trend and a seasonal component

d. Results Comparison

C. Project

2 Data sets used are from National Oceanic and Atmospheric Administration (NOAA)'s National Centers for Environmental Information (NCEI): * `metadataCA.txt`: has a number of sites, their elevations above sea level in feet, their geographic coordinates in latitude and longitude, and in the two right hand most columns, a reference point's coordinates on the west coast of California linked to the site that can be used to learn the site's distance from the ocean. * `MaxTempCalifornia.csv`: has maximum daily temperatures in degrees Celsius for those sites from Jan 1, 2012 to December 30, 2012

Initial Data Analysis

```
metadataCA <- read.csv("metadataCA.csv")
maxtempcalifornia <- read.csv("MaxTempCalifornia.csv")

head(metadataCA)
```

```
##        Location  Elev     Lat      Long  Ref_Lat  Ref_Long
## 1 San Francisco  45.7 37.7705 -122.4269 37.76889 -122.5156
## 2          Napa   4.3 38.2102 -122.2847 38.39222 -123.0892
## 3     San Diego   4.6 32.7336 -117.1831 32.72222 -117.2683
## 4        Fresno 100.0 36.7525 -119.7017 36.25833 -121.8389
## 5    Santa Cruz  39.6 36.9905 -121.9911 36.95528 -122.0933
## 6  Death Valley -59.1 36.4622 -116.8669 35.41750 -120.8369
```

```
head(maxtempcalifornia)
```

```
##          X San.Francisco Napa San.Diego Fresno Santa.Cruz Death.Valley Ojai
## 1 20120101          14.4 16.7      19.4   18.3       22.8         20.6 27.2
## 2 20120102          12.8 16.7      20.6   18.3       15.0         21.1 27.2
## 3 20120103          11.7 15.6      21.7   13.3       17.2         20.6 26.7
## 4 20120104          13.9 19.4      26.1   16.7       18.9         21.1 27.2
## 5 20120105          16.1 17.8      28.3   17.8       18.3         21.7 26.7
## 6 20120106          13.3 14.4      20.0   17.8       15.0         21.1 23.9
##   Barstow   LA CedarPark Redding
## 1    20.6 27.2      19.4    17.2
## 2    17.2 23.9      21.7    15.0
## 3    18.3 24.4      10.6    18.3
## 4    18.9 29.4       3.3    19.4
## 5    19.4 28.3       8.9    19.4
## 6    20.0 22.8      16.1    17.2
```

```
maxtempcalifornia_long <- maxtempcalifornia %>%
  gather(Location, Max_Temp, -c(X))
maxtempcalifornia_long$Location <- maxtempcalifornia_long$Location %>%
  str_replace("\\.", " ")
maxtempcalifornia_long$Date <- ymd(maxtempcalifornia_long$X)
head(maxtempcalifornia_long)
```

```
##          X      Location Max_Temp       Date
## 1 20120101 San Francisco     14.4 2012-01-01
## 2 20120102 San Francisco     12.8 2012-01-02
## 3 20120103 San Francisco     11.7 2012-01-03
## 4 20120104 San Francisco     13.9 2012-01-04
## 5 20120105 San Francisco     16.1 2012-01-05
## 6 20120106 San Francisco     13.3 2012-01-06
```

1. Numerical and Graphical summaries of the data from each site

```
summary(metadataCA)
```

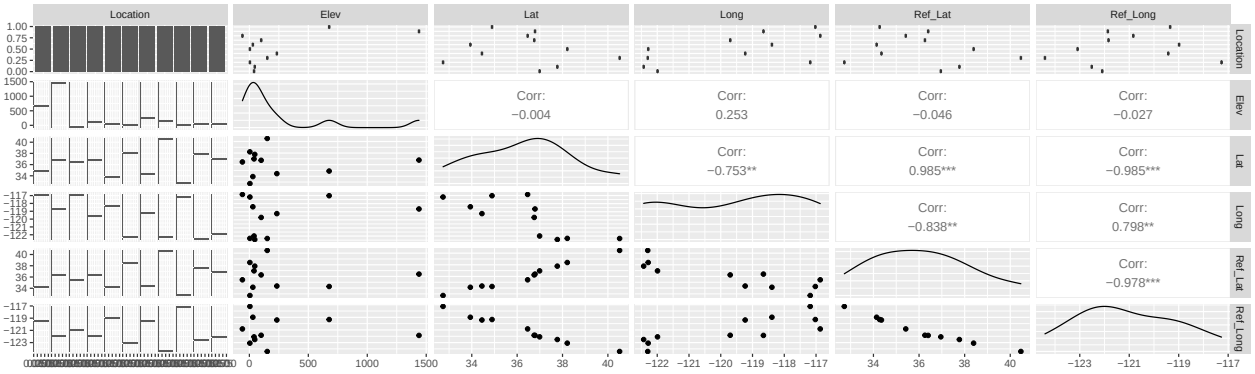
```
##    Location             Elev             Lat             Long
##  Length:11          Min.   : -59.1   Min.   :32.73   Min.   :-122.4
##  Class :character   1st Qu.:  17.1   1st Qu.:34.67   1st Qu.:-122.1
##  Mode  :character   Median :  45.7   Median :36.75   Median :-119.2
##                     Mean   : 242.3   Mean   :36.32   Mean   :-119.6
```

```
##                     3rd Qu.: 192.3   3rd Qu.:37.38   3rd Qu.:-117.8
##                     Max.   :1438.7   Max.   :40.52   Max.   :-116.9
##     Ref_Lat          Ref_Long
##  Min.   :32.72   Min.   :-124.4
##  1st Qu.:34.31   1st Qu.:-122.3
##  Median :36.26   Median :-121.8
##  Mean   :36.10   Mean   :-121.1
##  3rd Qu.:37.36   3rd Qu.:-119.4
##  Max.   :40.47   Max.   :-117.3
```

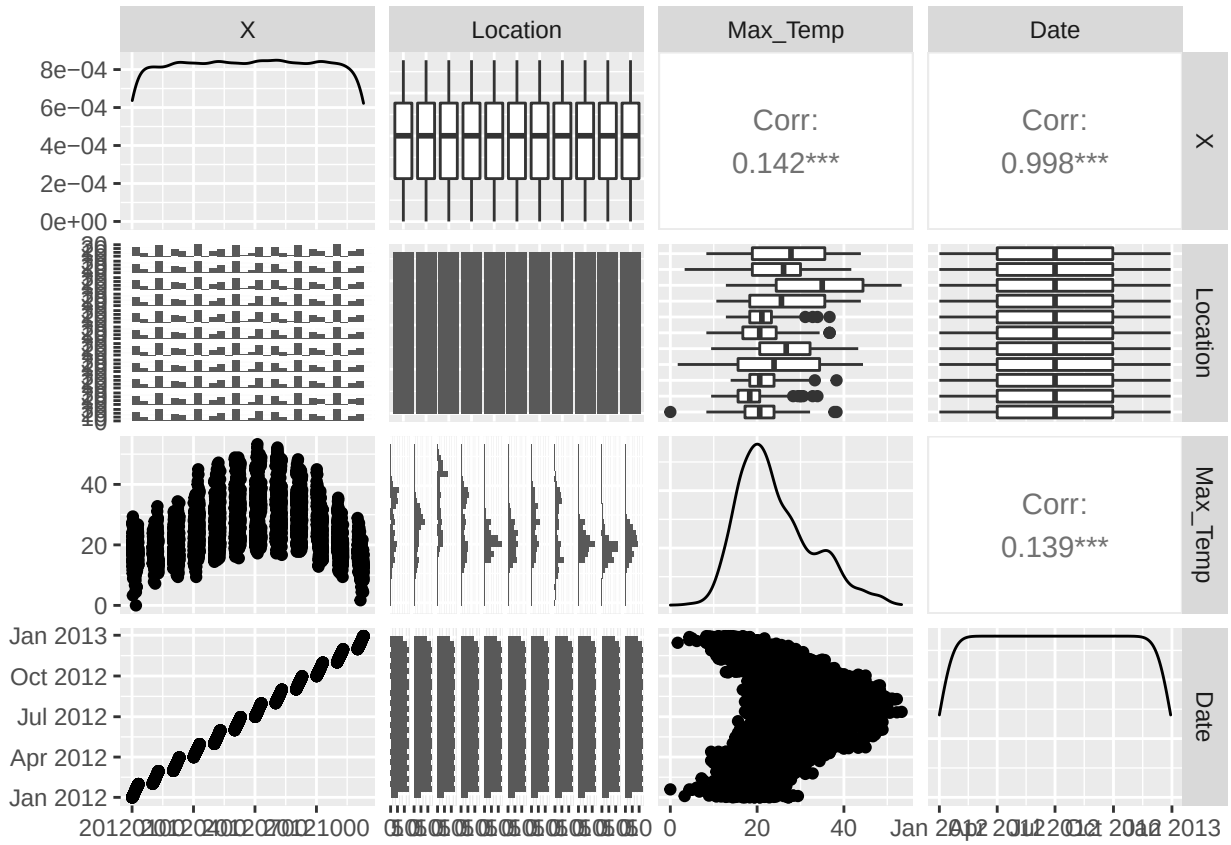
```
summary(maxtempcalifornia_long)
```

```
##        X             Location           Max_Temp          Date
##  Min.   :20120101   Length:4015        Min.   : 0.00   Min.   :2012-01-01
##  1st Qu.:20120401   Class :character   1st Qu.:17.80   1st Qu.:2012-04-01
##  Median :20120701   Mode  :character   Median :22.20   Median :2012-07-01
##  Mean   :20120666                      Mean   :24.15   Mean   :2012-07-01
##  3rd Qu.:20120930                      3rd Qu.:28.90   3rd Qu.:2012-09-30
##  Max.   :20121230                      Max.   :53.30   Max.   :2012-12-30
```

```
metadataCA %>%
  ggpairs()
```

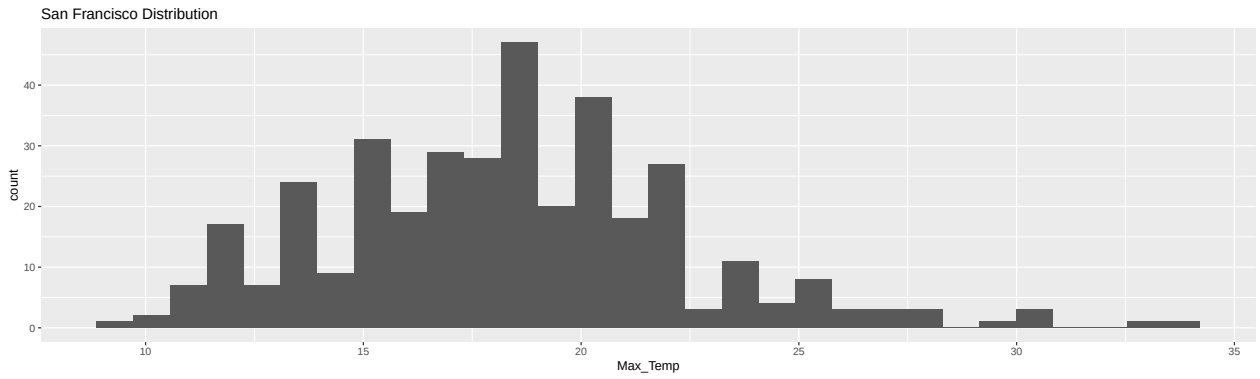


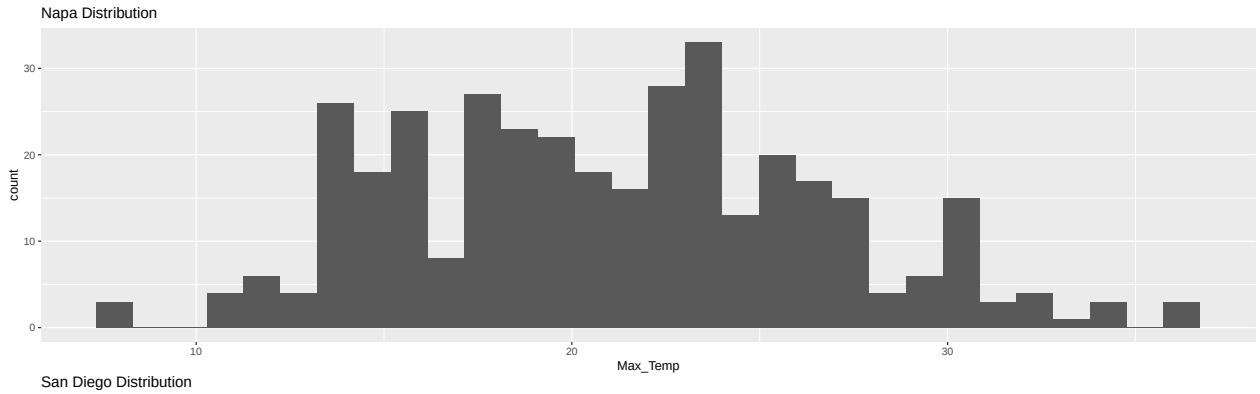

```
maxtempcalifornia_long %>%
  ggpairs()
```

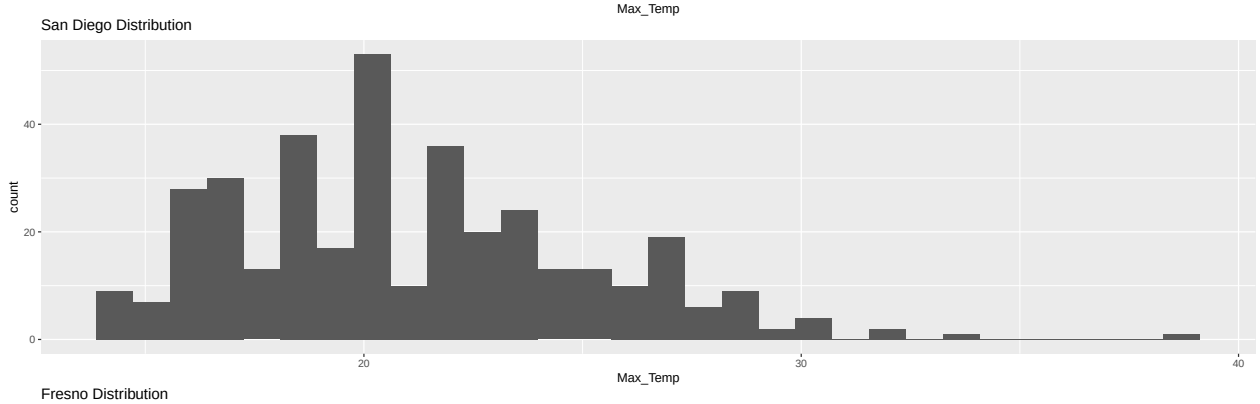


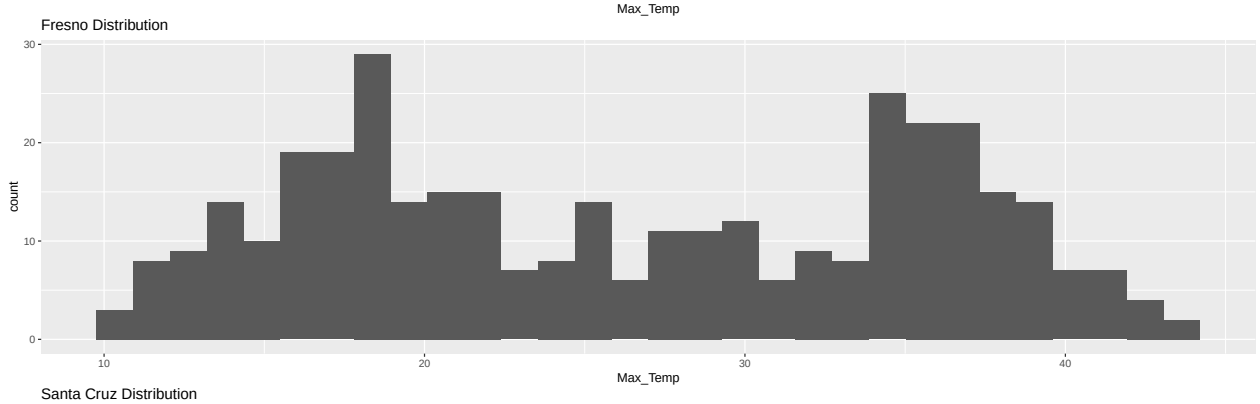
2. Distributions of the data at each location

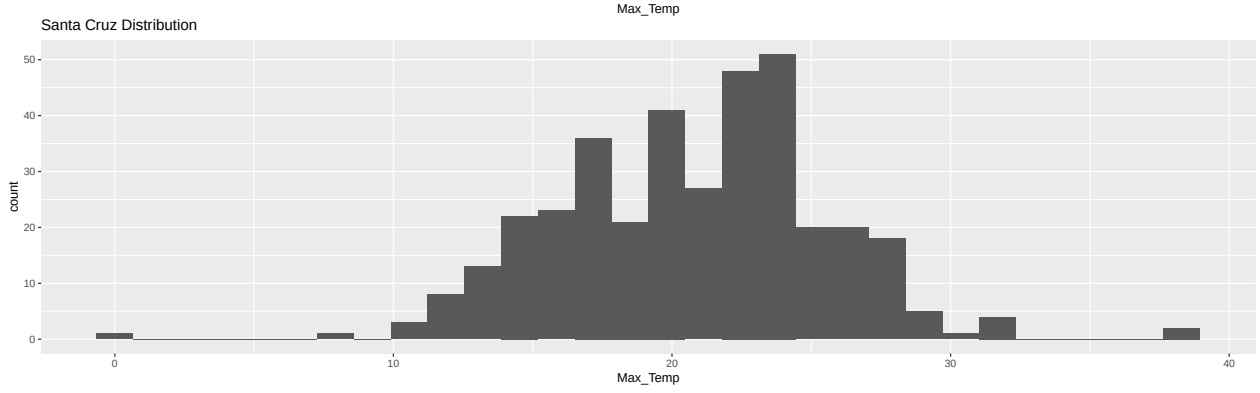
```
for(location in unique(maxtempcalifornia_long$Location)){
  p <- maxtempcalifornia_long %>%
    filter(Location==location) %>%
    ggplot(aes(x=Max_Temp))+
    geom_histogram()+
    ggtitle(paste(location,"Distribution"))
  print(p)
}
```

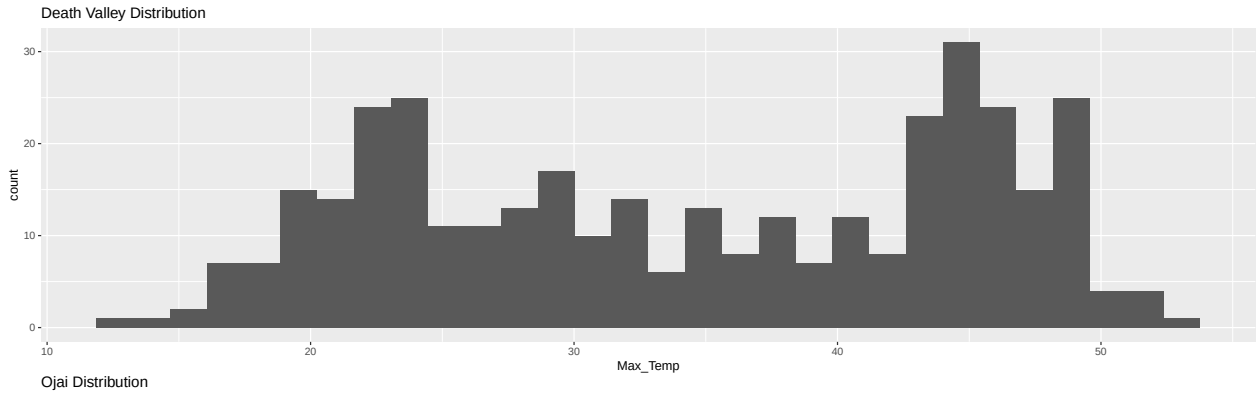


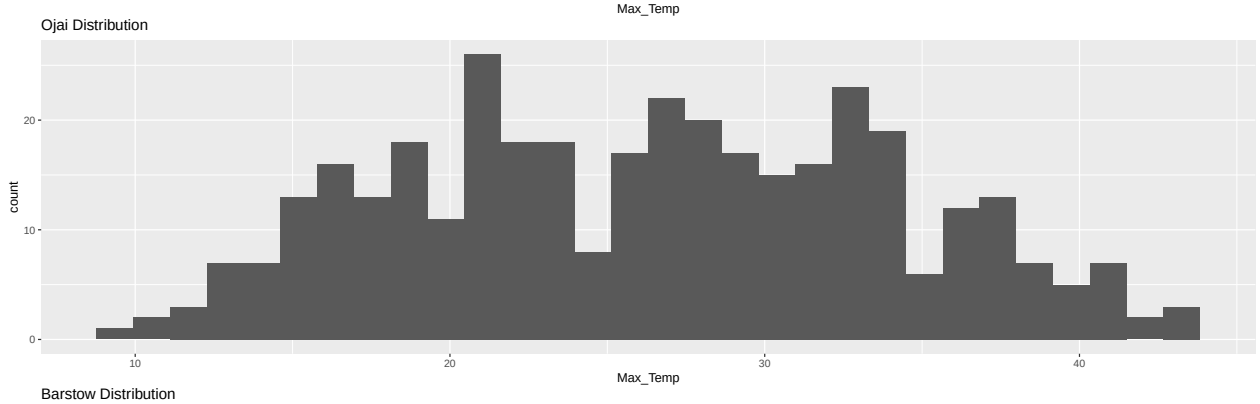

Napa Distribution
count
Max_Temp

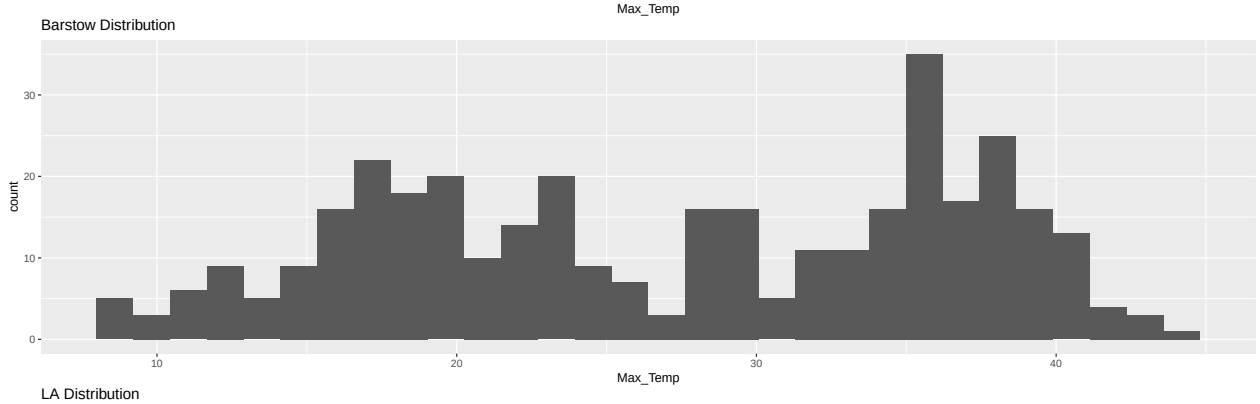

San Diego Distribution
count
Max_Temp

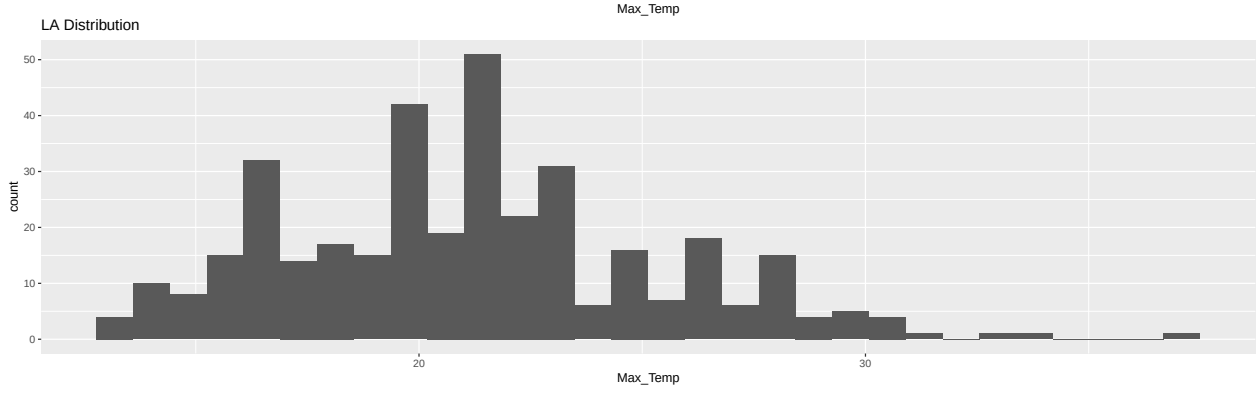

Fresno Distribution
count
Max_Temp

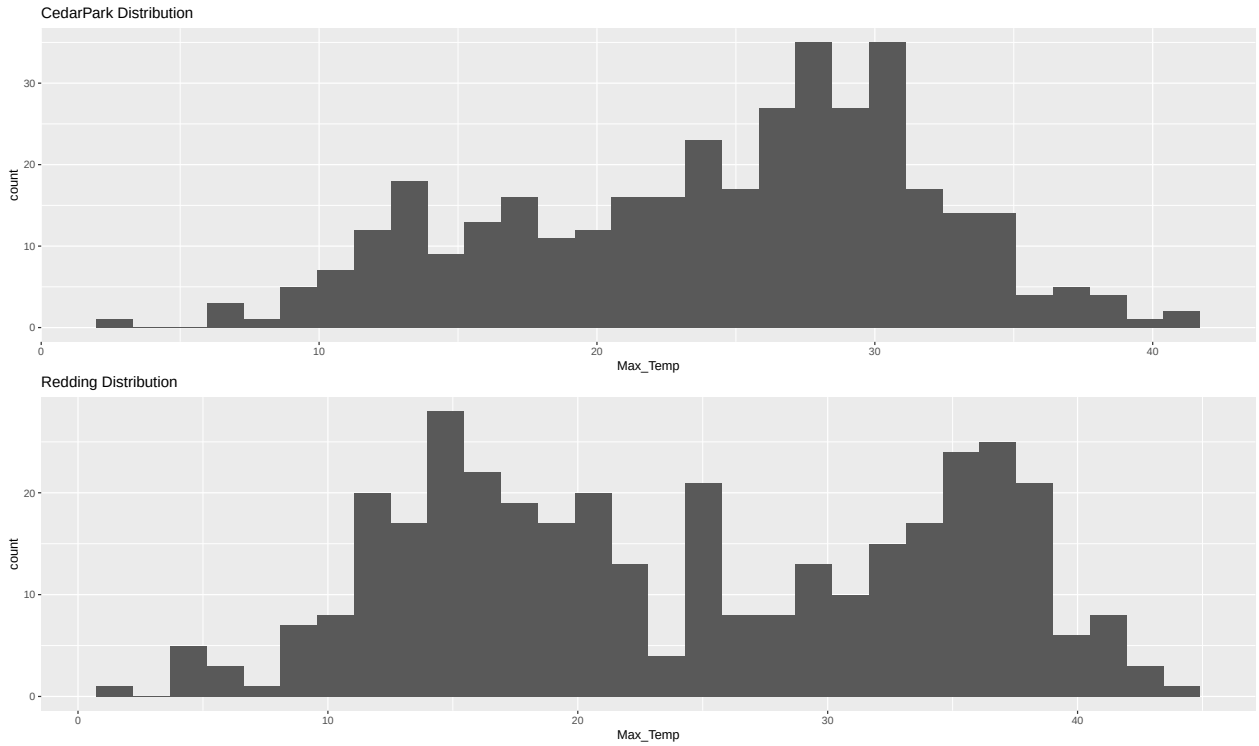

Santa Cruz Distribution
count
Max_Temp


Death Valley Distribution
count
Max_Temp


Ojai Distribution
count
Max_Temp


Barstow Distribution
count
Max_Temp

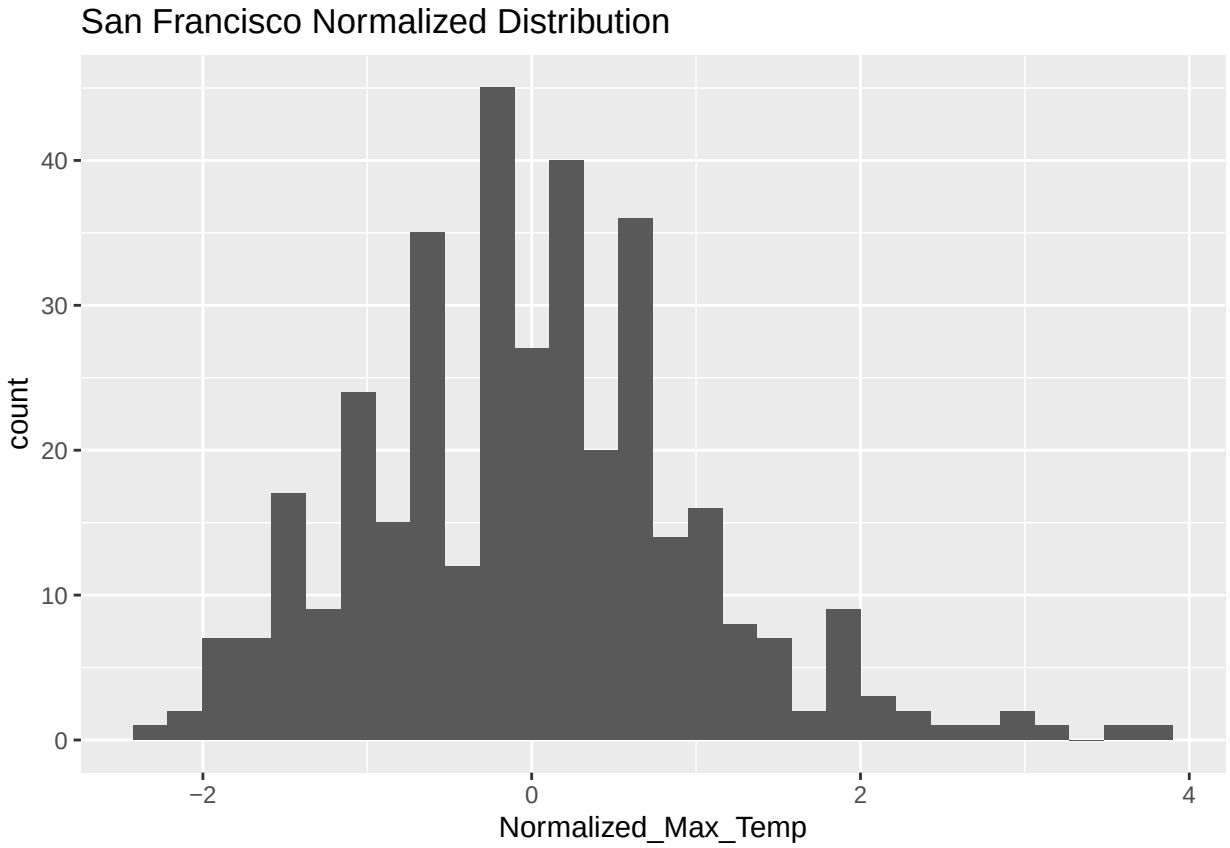

LA Distribution
count
Max_Temp



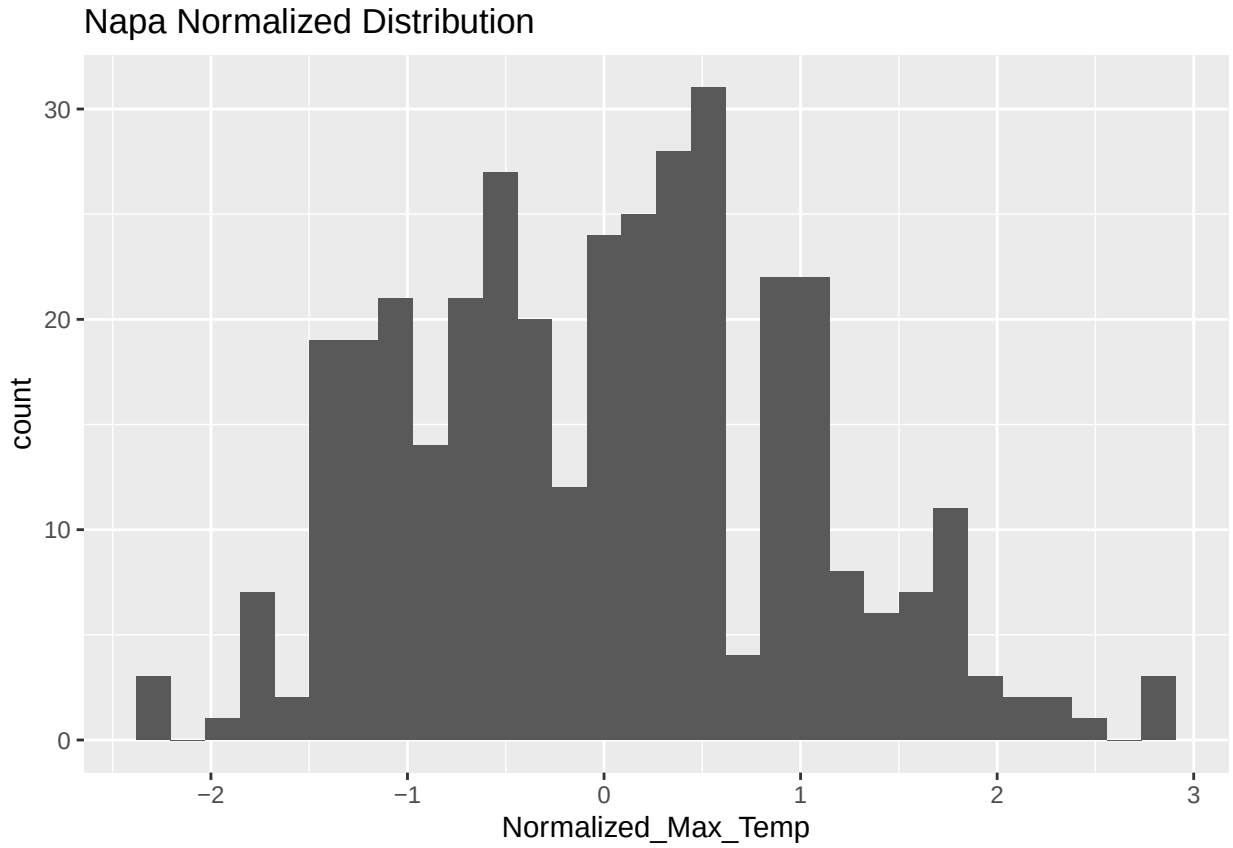
For each location, the data doesn't look Normally distributed, therefore transformation for each site needs to be done. The **Ojai** location is almost normally distributed. For this transformation we will use `bestNormalize` package to transform the data at each site to be normally distributed

```
maxtempcalifornia_long$Normalized_Max_Temp <- 0
for(location in unique(maxtempcalifornia_long$Location)){
  maxtempcalifornia_long[maxtempcalifornia_long$Location==location, c("Normalized_Max_Temp")] <- bestNo
}
for(location in unique(maxtempcalifornia_long$Location)){
  p <- maxtempcalifornia_long %>%
    filter(Location==location) %>%
    ggplot(aes(x=Normalized_Max_Temp))+
    geom_histogram()+
    ggtitle(paste(location,"Normalized Distribution"))
  print(p)
}
```

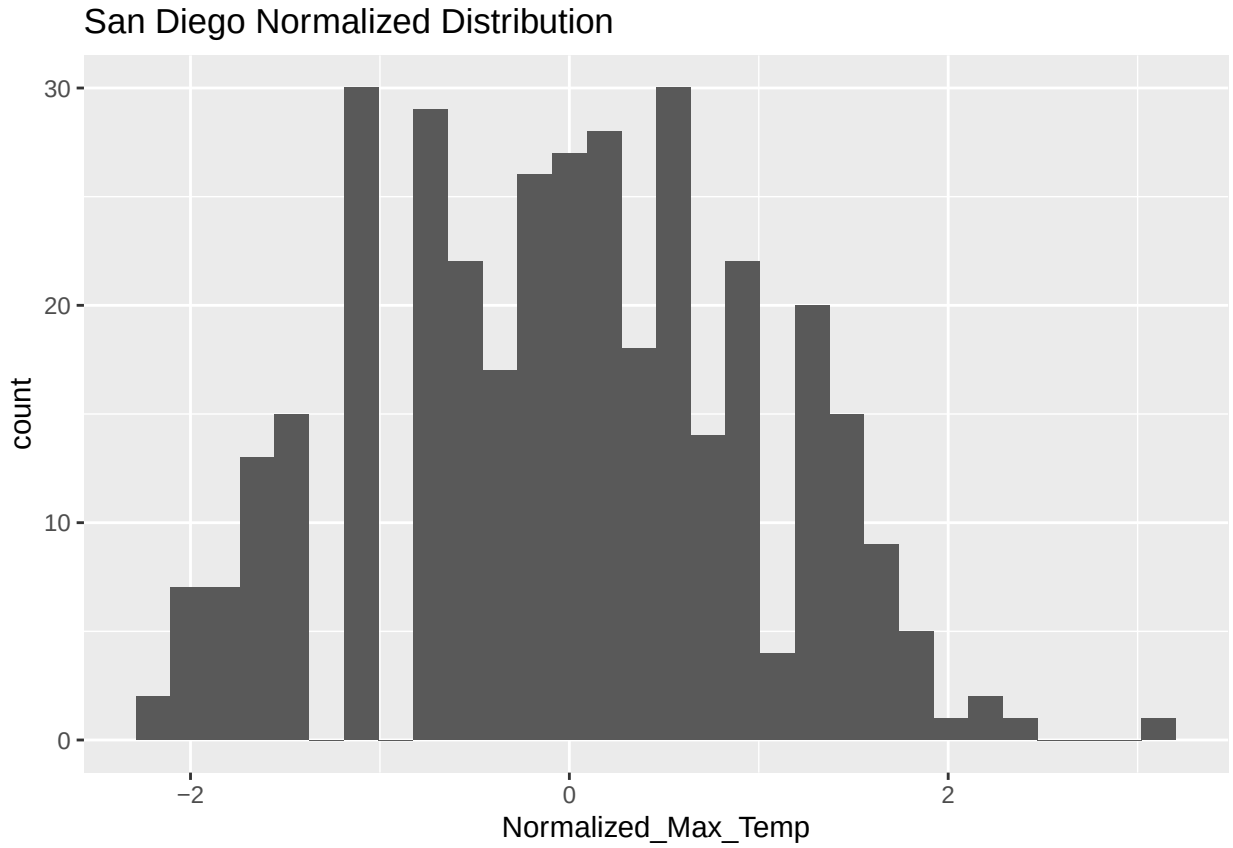
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



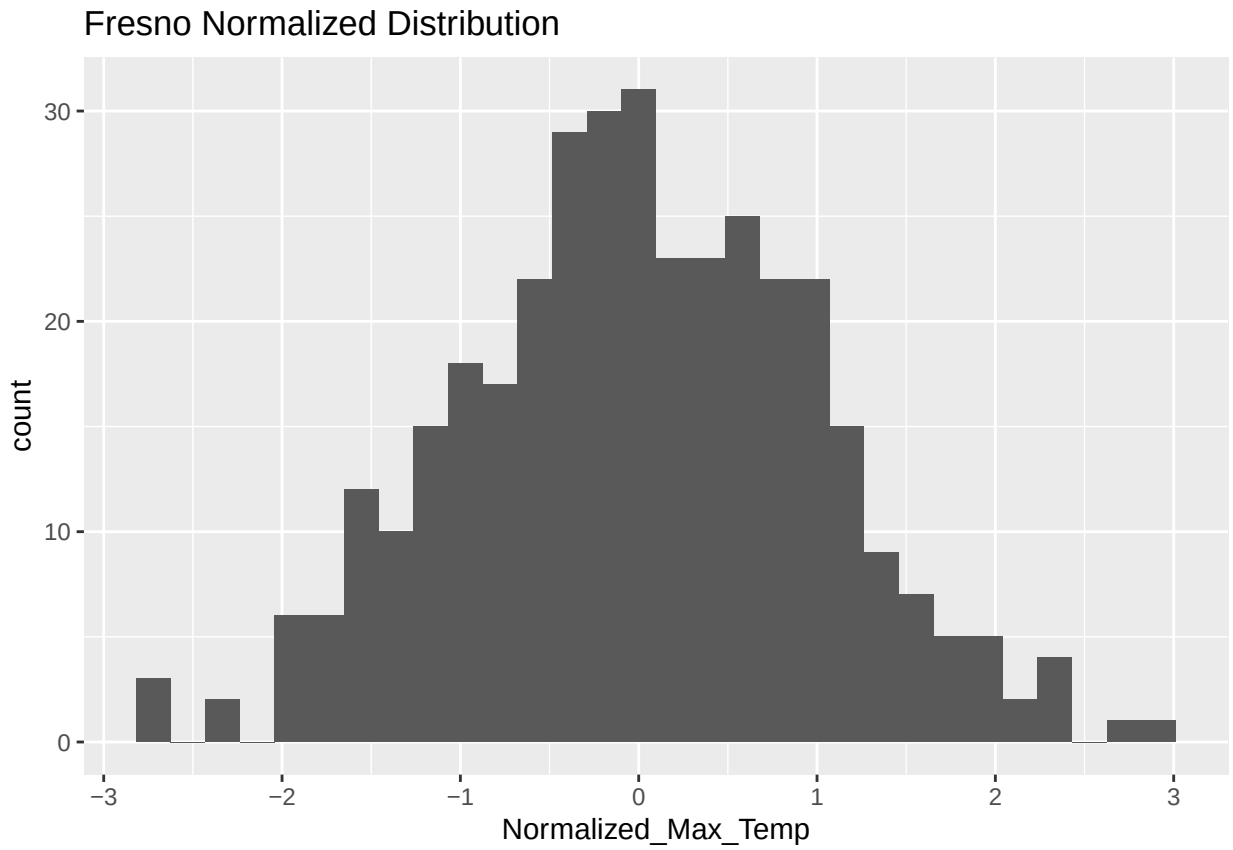

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



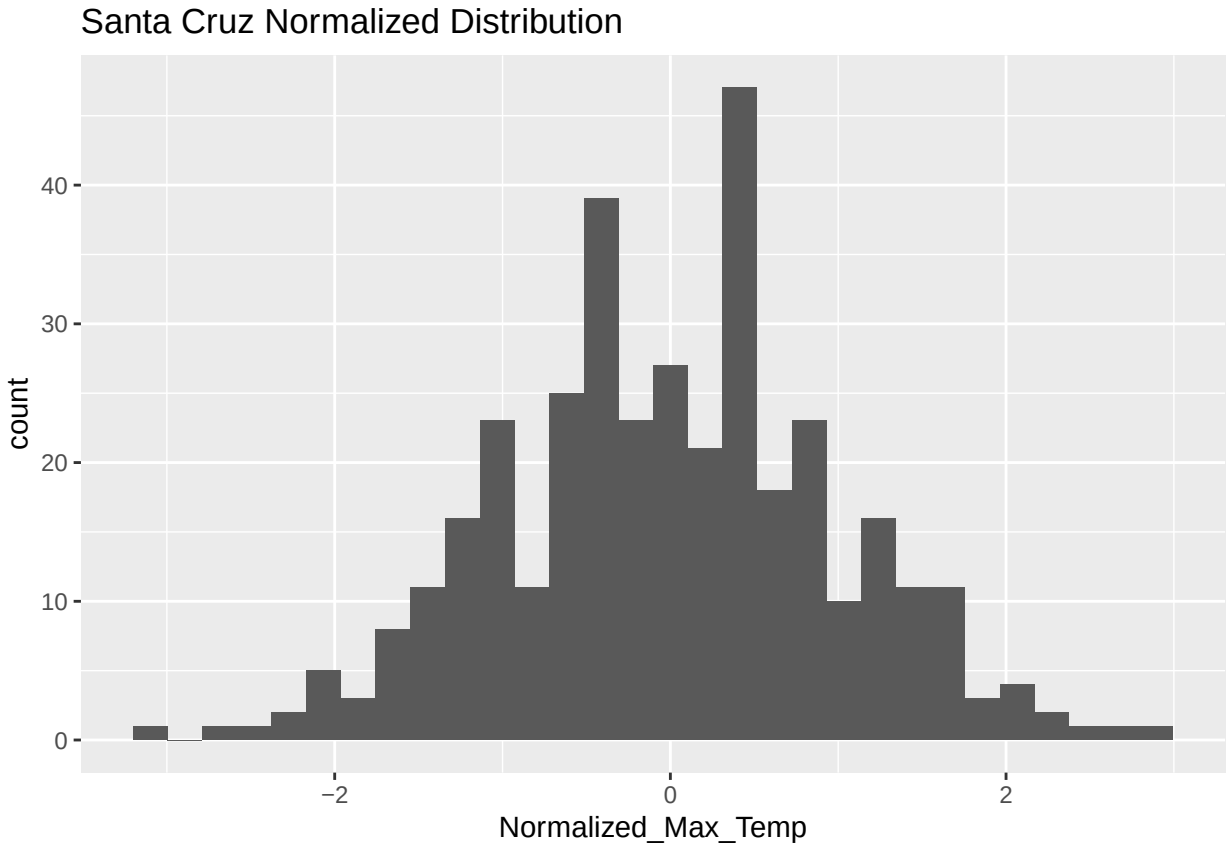

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



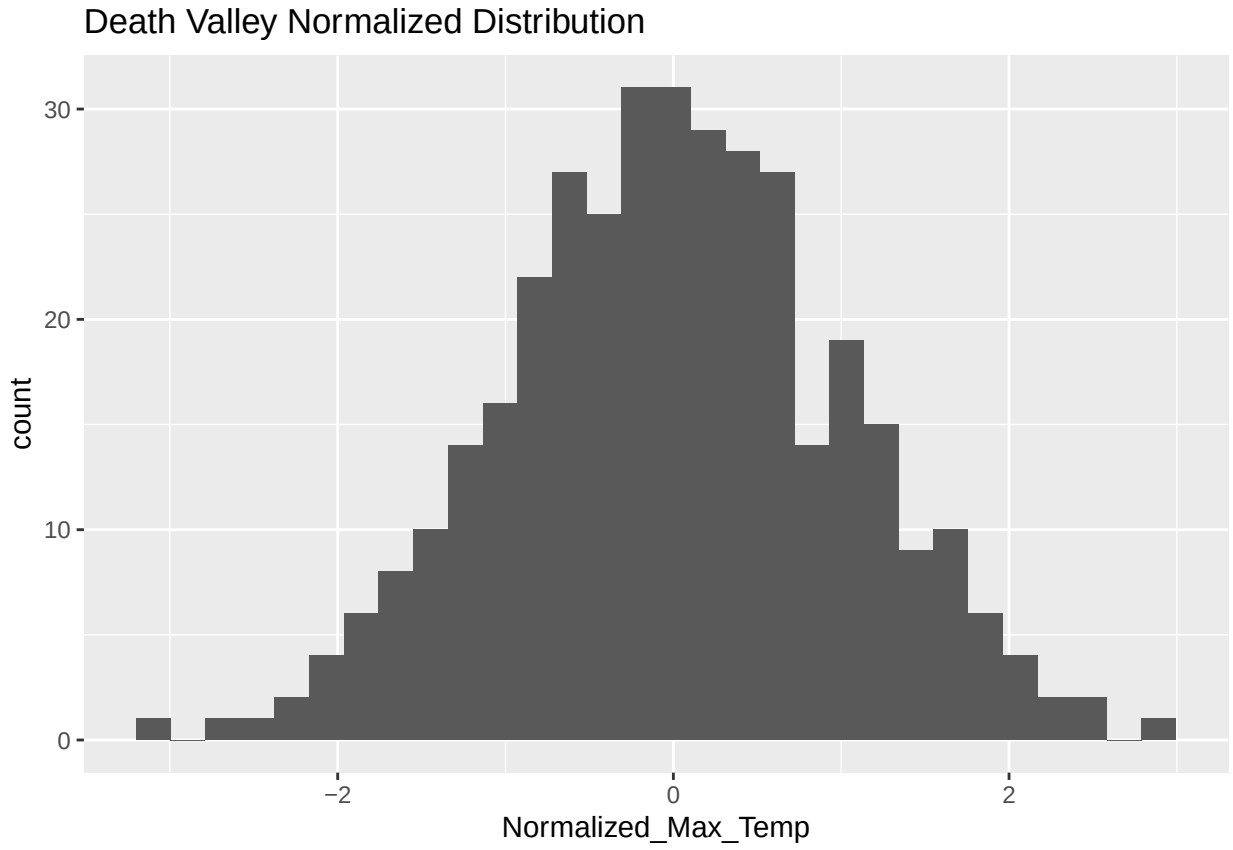

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



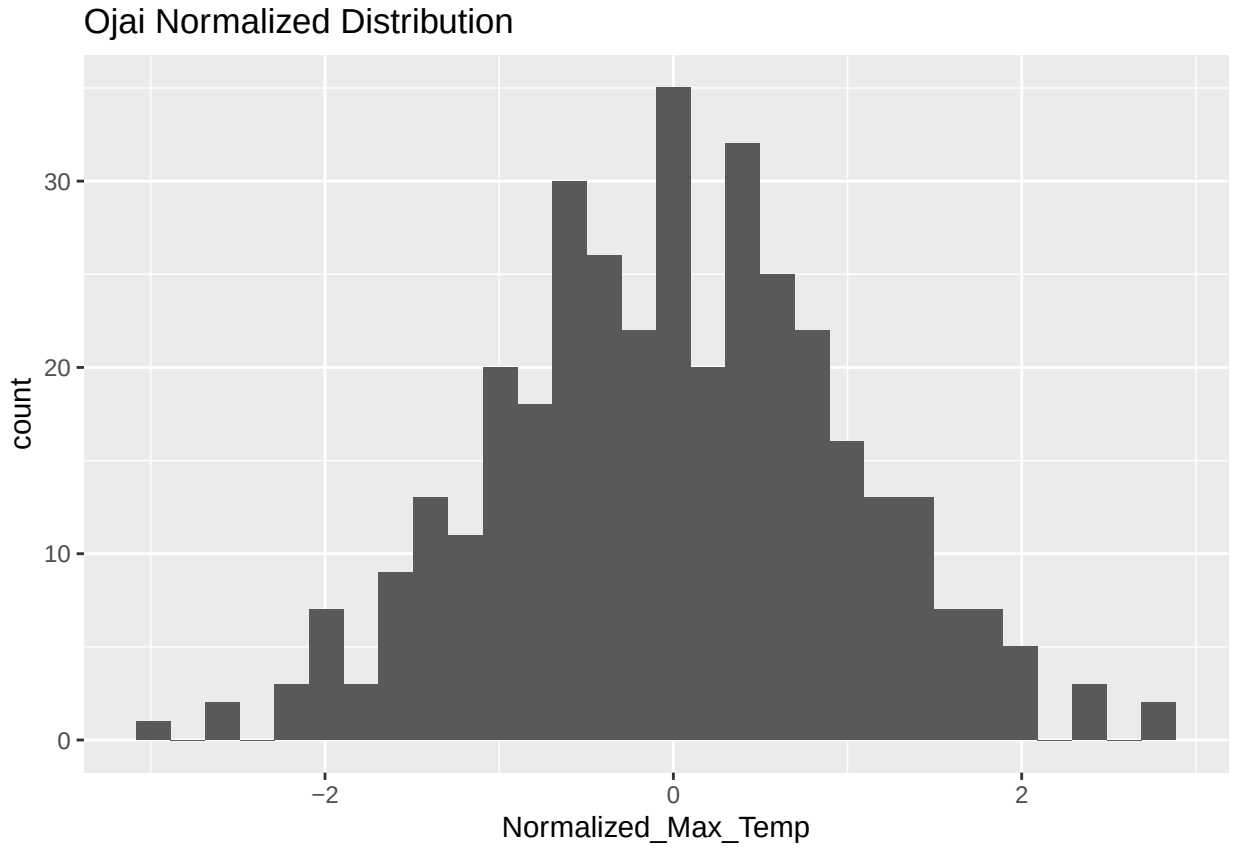

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



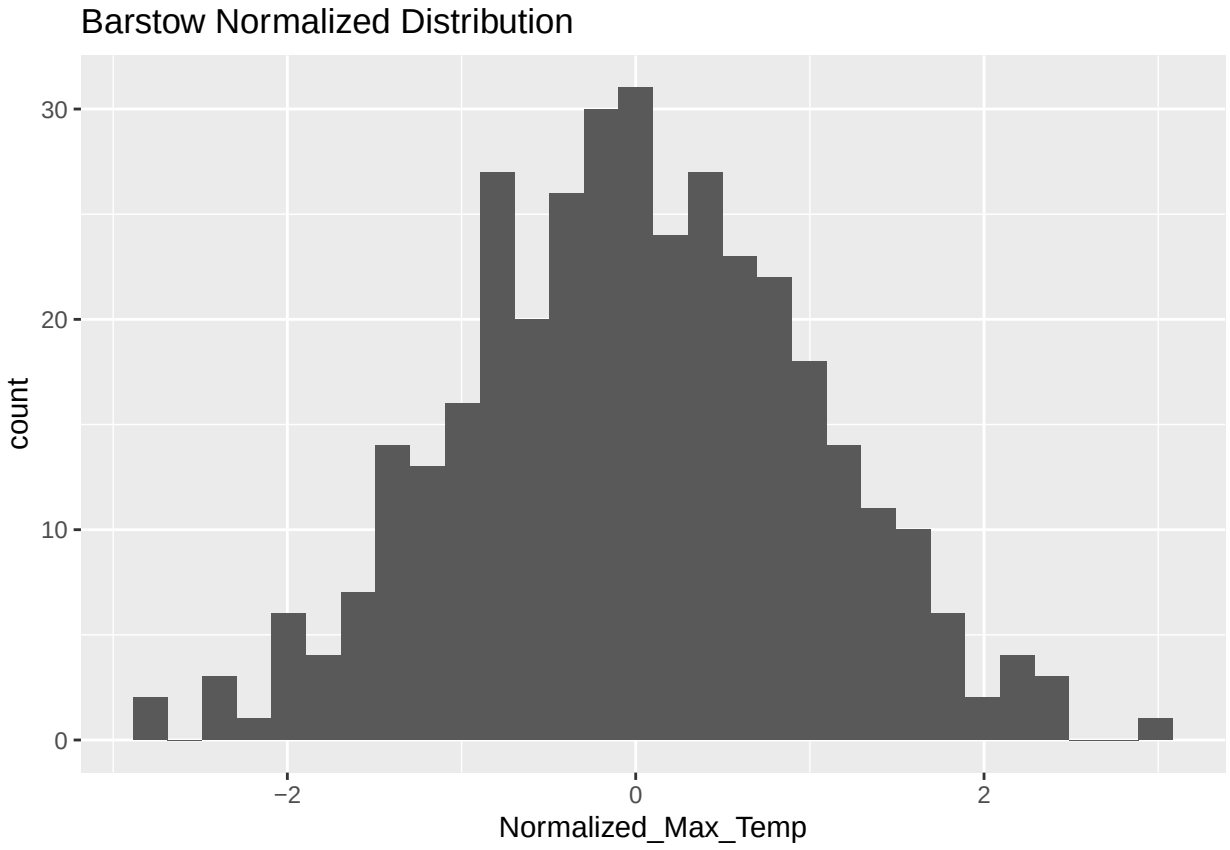

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



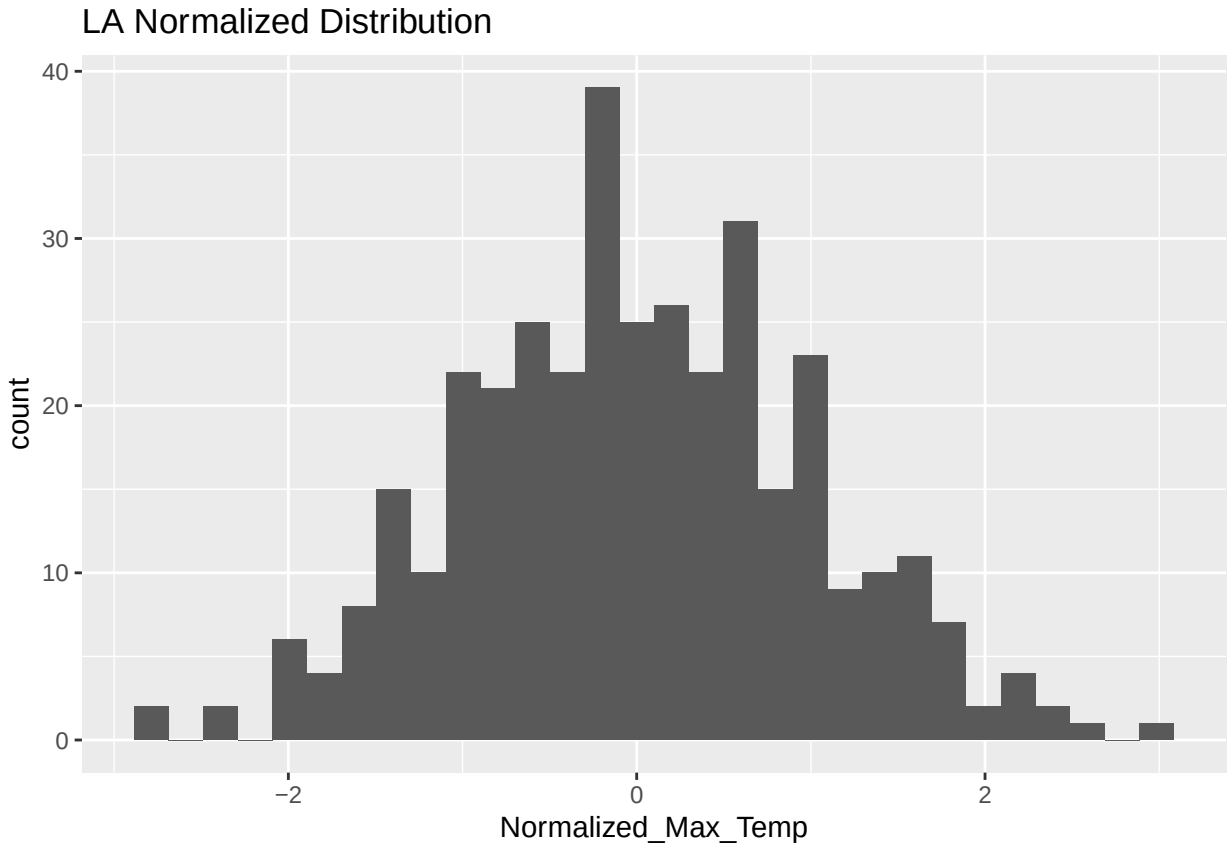

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



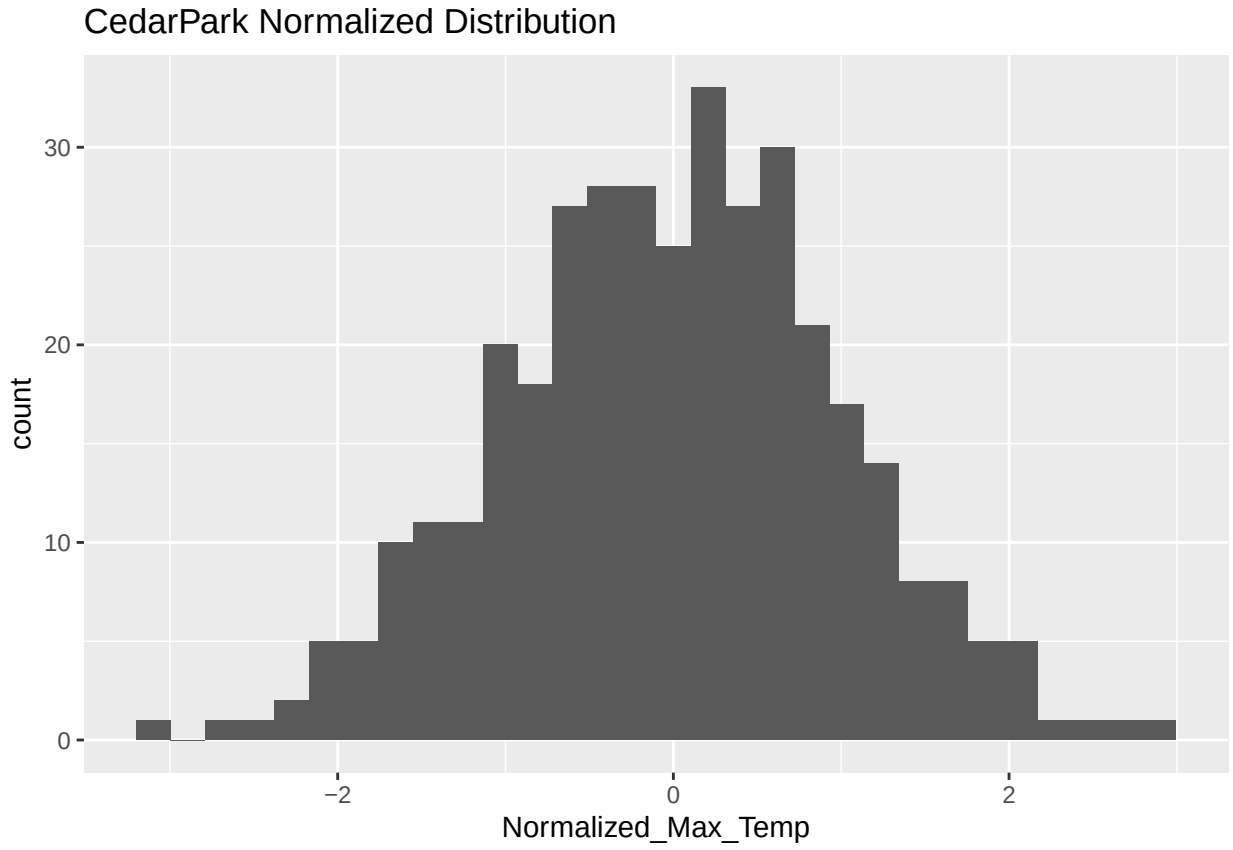

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



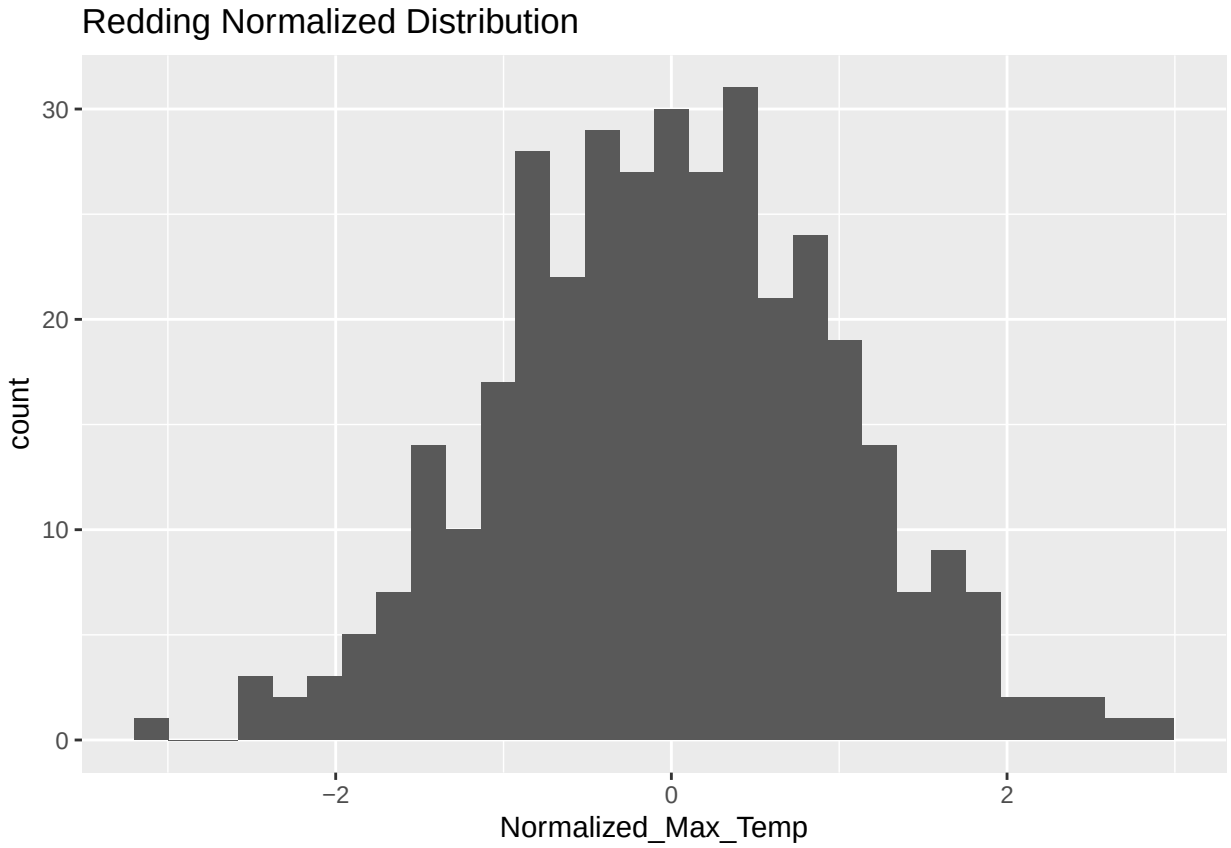

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



After Normalizing based on each location, it now seems reasonable and the distributions are now normally distributed.

3. Monthly average (max) temperatures for each site

4. Statistical Analysis of whether there are differences in (max) temperatures at different locations, and whether there are (statistically significant) differences between months.

Prediction

5. Developing a time series model and applying it to data from other locations to predict maximum temperatures for all locations, for the 1st to 8th August 2012

6. Developing a spatial model to predict maximum temperatures for San Fransisco and Death Valley for 1st Jan 2012

Report

7. Analysis of maximum temperatures over both space and time