MTHM505J Data Science and Statistical Modelling in space and time - REF-DEF Assignment

MTHM505J Data Science and Statistical Modelling in space and time - REF-DEF Assignment

Libraries

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

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

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)

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

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

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

if(!require("devtools")) install.packages("devtools");
if (!require("rspatial")) devtools::install_github('rspatial/rspatial');
library(rspatial)

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

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

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

set.seed(42)
if(!require("spatialreg")) install.packages("spatialreg");
library(spatialreg)
```

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

```r
# 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
```

```r
# 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)
```

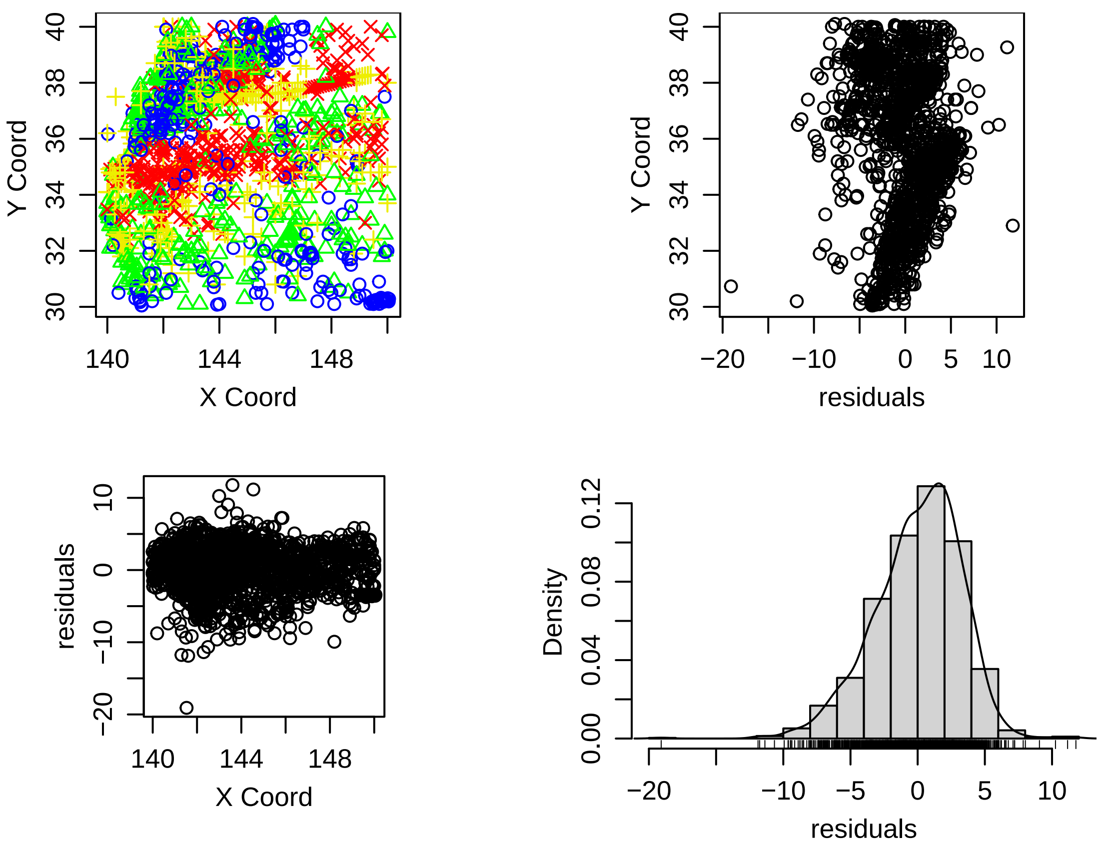
1. numerical and graphical summaries of the data.

```r
# Get the summary of jittered coordinates
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"
```

```
# visualize
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  
##
```

```
data %>%
  dplyr::select(-date,-id)%>%
  ggpairs()
```

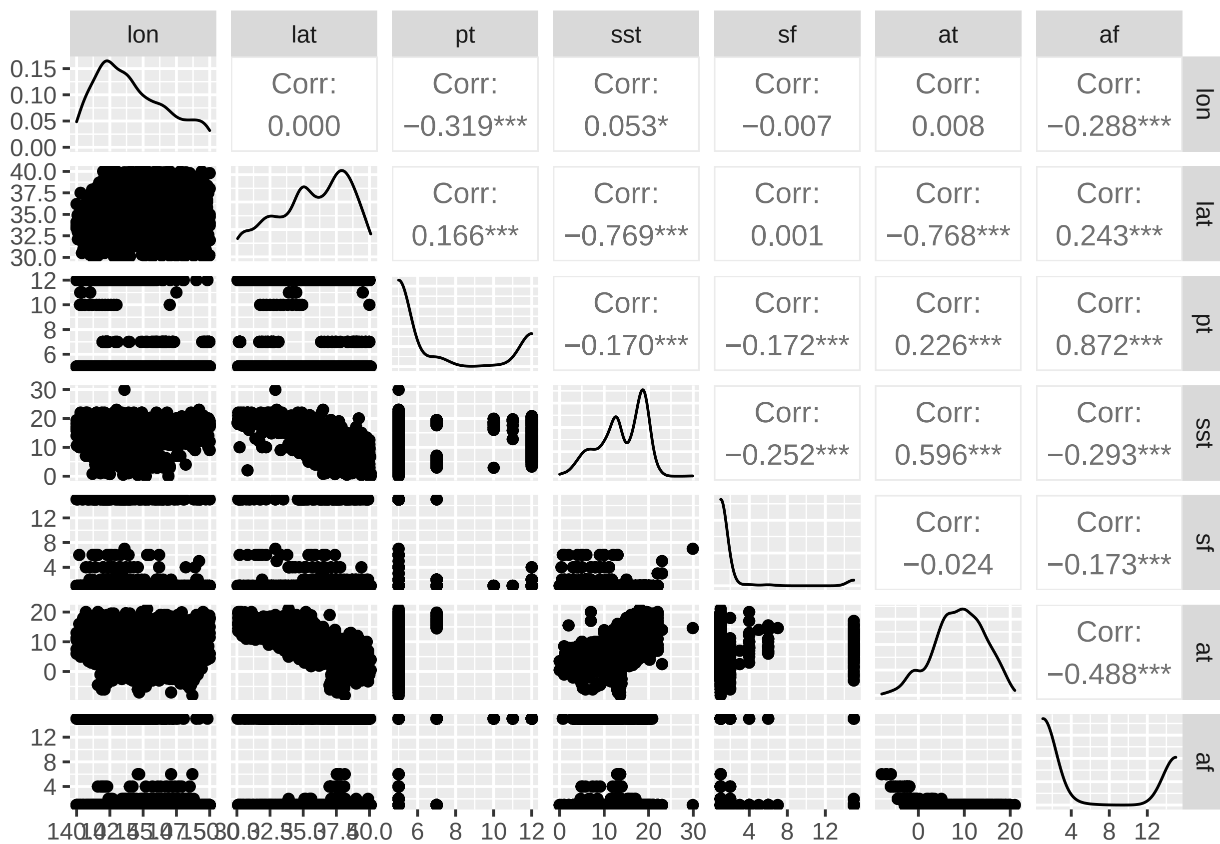


Figure 1: Distributions of the Columns

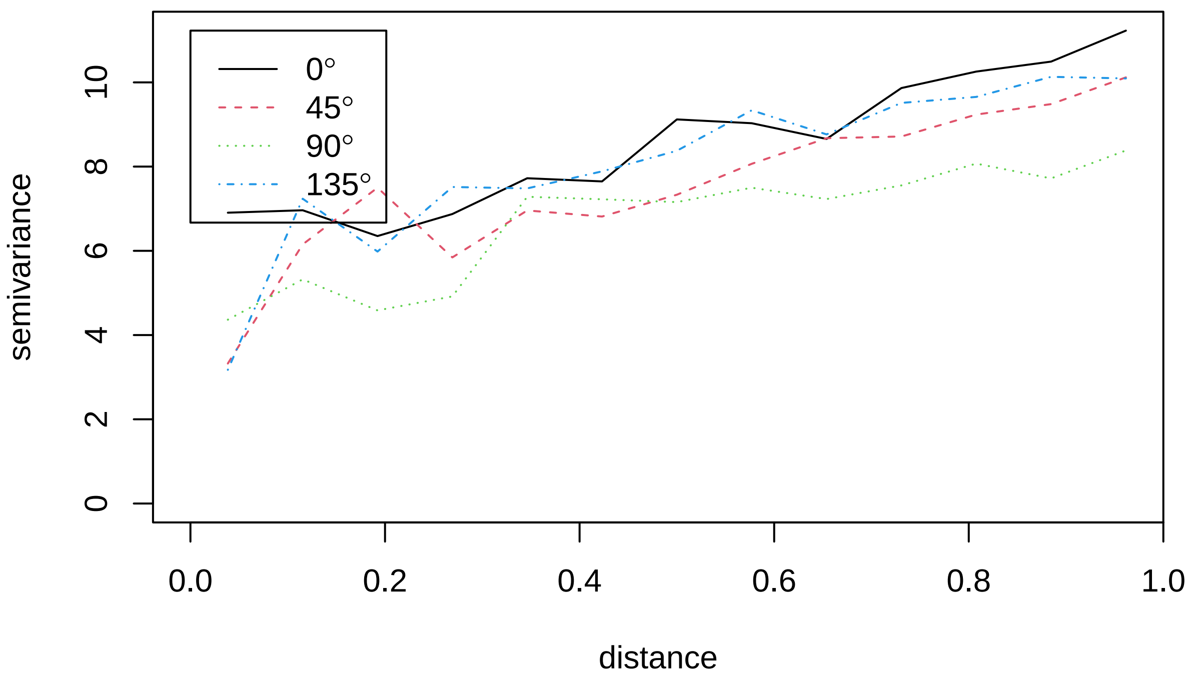
From the numerical summary we can see the data has missing values only in the `sst` and `at` columns. Outliers exist in `sf` columns, leading to skewness in the distribution of this column. Majority of the variables have negative correlation such as (`sst` and `pt`, `at` and `af`). The plotted geospatial graph distribution shows that the residuals can be normally distributed in absence of the outliers.

2. Check Isotropy

```r
# use variog to check for isotropy
isotropy <- variog4(gdata2, max.dist=1)
```

```
## variog: computing variogram for direction = 0 degrees (0 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 45 degrees (0.785 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 90 degrees (1.571 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing variogram for direction = 135 degrees (2.356 radians)
##         tolerance angle = 22.5 degrees (0.393 radians)
## variog: computing omnidirectional variogram
```

```r
plot(isotropy)
```



From the directional variograms, their is a need for a trend in spatial model.

3. Fit Spatial Model

We will try various models and pick the one that fits best

```r
isotropy <- variog(gdata2, uvec=seq(0,1,l=11))
```

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

```r
# Fitting models with nugget fixed to zero
ml <- likfit(gdata2, ini = c(1,0.5), fix.nugget = T)
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optimize.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optimize.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```r
reml <- likfit(gdata2, ini = c(1,0.5), fix.nugget = T, method = "RML")
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optimize.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optimize.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```r
ols <- variofit(isotropy, ini = c(1,0.5), fix.nugget = T, weights="equal")
```

```
## variofit: covariance model used is matern
## variofit: weights used: equal
## variofit: minimisation function used: optim
```

```r
wls <- variofit(isotropy, ini = c(1,0.5), fix.nugget = T)
```

```
## variofit: covariance model used is matern
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```r
# Fitting models with a fixed value for the nugget
ml.fn <- likfit(gdata2, ini = c(1,0.5), fix.nugget = T, nugget = 0.15)
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```
reml.fn <- likfit(gdata2, ini = c(1,0.5), fix.nugget = T, nugget = 0.15, method = "RML")
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```
ols.fn <- variofit(isotropy,ini = c(1,0.5), fix.nugget = T, nugget = 0.15, weights="equal")
```

```
## variofit: covariance model used is matern
## variofit: weights used: equal
## variofit: minimisation function used: optim
```

```
wls.fn <- variofit(isotropy, ini = c(1,0.5), fix.nugget = T, nugget = 0.15)
```

```
## variofit: covariance model used is matern
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```
# Fitting models estimated nugget
ml.n <- likfit(gdata2, ini = c(1,0.5), nug = 0.5)
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```r
reml.n <- likfit(gdata2, ini = c(1,0.5), nug = 0.5, method = "RML")
```

```
## ---------------------------------------------------------------
## likfit: likelihood maximisation using the function optim.
## likfit: Use control() to pass additional
##          arguments for the maximisation function.
##         For further details see documentation for optim.
## likfit: It is highly advisable to run this function several
##         times with different initial values for the parameters.
## likfit: WARNING: This step can be time demanding!
## ---------------------------------------------------------------
## likfit: end of numerical maximisation.
```

```r
ols.n <- variofit(isotropy, ini = c(1,0.5), nugget=0.5, weights="equal")
```

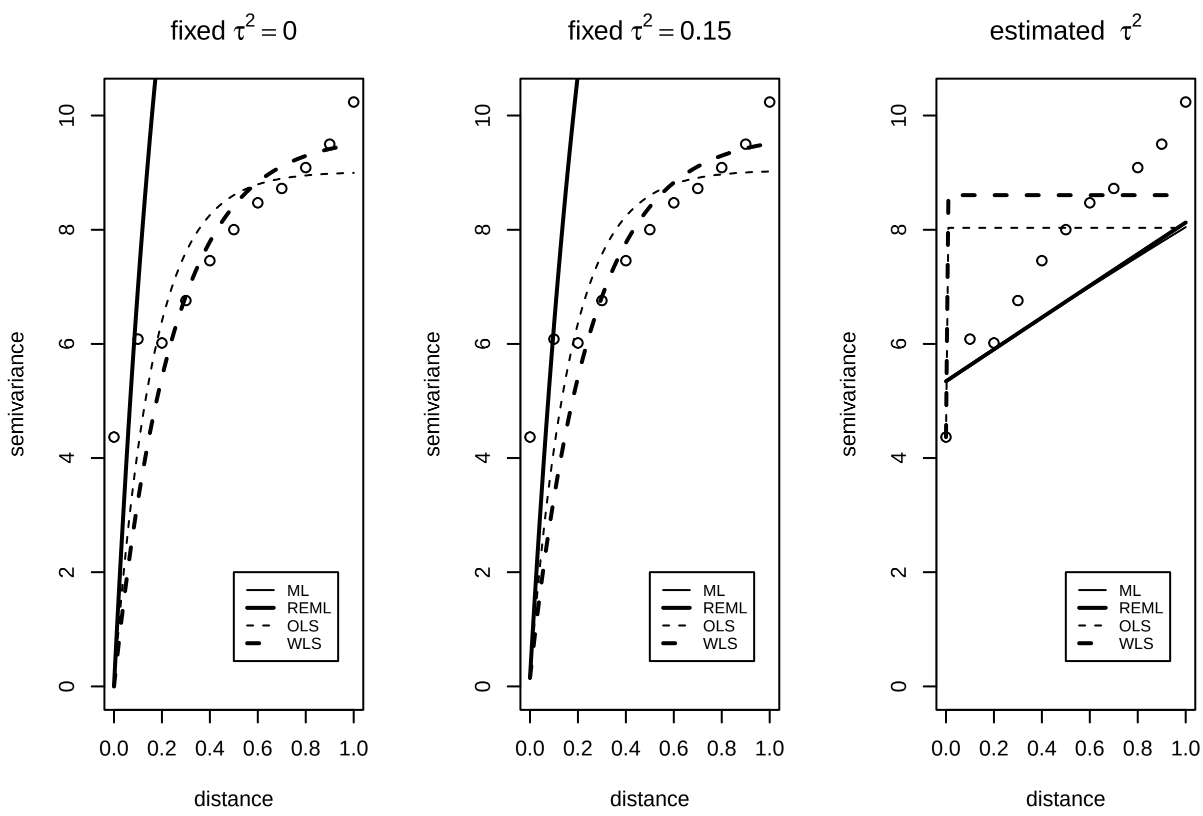
```
## variofit: covariance model used is matern
## variofit: weights used: equal
## variofit: minimisation function used: optim
```

```r
wls.n <- variofit(isotropy, ini = c(1,0.5), nugget=0.5)
```

```
## variofit: covariance model used is matern
## variofit: weights used: npairs
## variofit: minimisation function used: optim
```

```r
# Now, plotting fitted models against empirical variogram
par(mfrow = c(1,3))
plot(isotropy, main = expression(paste("fixed ", tau^2 == 0)))
lines(ml, max.dist = 1)
lines(reml, lwd = 2, max.dist = 1)
lines(ols, lty = 2, max.dist = 1)
lines(wls, lty = 2, lwd = 2, max.dist = 1)
legend(
  0.5,
  2,
  legend=c("ML","REML","OLS","WLS"),
  lty=c(1,1,2,2),
  lwd=c(1,2,1,2),
  cex=0.7
)

plot(isotropy, main = expression(paste("fixed ", tau^2 == 0.15)))
lines(ml.fn, max.dist = 1)
lines(reml.fn, lwd = 2, max.dist = 1)
lines(ols.fn, lty = 2, max.dist = 1)
lines(wls.fn, lty = 2, lwd = 2, max.dist = 1)
legend(
  0.5,
  2,
  legend=c("ML","REML","OLS","WLS"),
  lty=c(1,1,2,2),
```

```
  lwd=c(1,2,1,2),
  cex=0.7
)

plot(isotropy, main = expression(paste("estimated  ", tau^2)))
lines(ml.n, max.dist = 1)
lines(reml.n, lwd = 2, max.dist = 1)
lines(ols.n, lty = 2, max.dist = 1)
lines(wls.n, lty =2, lwd = 2, max.dist = 1)
legend(
  0.5,
  2,
  legend=c("ML","REML","OLS","WLS"),
  lty=c(1,1,2,2),
  lwd=c(1,2,1,2),
  cex=0.7
)
```




```
par(par(no.readonly = TRUE))
```

The directional variogram revealed a trend in spatial model therefore the best spatial model that will be fitted with this data will be based on likelihood based parameter estimation for Gaussian Random Fields.

```r
print(ml.n)
```

```
## likfit: estimated model parameters:
##      beta    tausq  sigmasq      phi
## "14.415" " 5.337" "27.869" " 9.787"
## Practical Range with cor=0.05 for asymptotic range: 29.3194
##
## likfit: maximised log-likelihood = -3680
```

```r
summary(ml.n)
```

```
## Summary of the parameter estimation
## -----------------------------------
## Estimation method: maximum likelihood
##
## Parameters of the mean component (trend):
##    beta
## 14.4155
##
## Parameters of the spatial component:
##    correlation function: exponential
##       (estimated) variance parameter sigmasq (partial sill) =  27.87
##       (estimated) cor. fct. parameter phi (range parameter)  =  9.787
##    anisotropy parameters:
##       (fixed) anisotropy angle = 0  ( 0 degrees )
##       (fixed) anisotropy ratio = 1
##
## Parameter of the error component:
##       (estimated) nugget =  5.337
##
## Transformation parameter:
##       (fixed) Box-Cox parameter = 1 (no transformation)
##
## Practical Range with cor=0.05 for asymptotic range: 29.3194
##
## Maximised Likelihood:
##    log.L n.params      AIC      BIC
##  "-3680"      "4"   "7368"   "7389"
##
## non spatial model:
##    log.L n.params      AIC      BIC
##  "-4713"      "2"   "9430"   "9441"
##
## Call:
## likfit(geodata = gdata2, ini.cov.pars = c(1, 0.5), nugget = 0.5)
```

The maximum likelihood of the model is -3680.

4. Fit by Bayesian methods

Bayesian methods is implemented by the function `krige.bayes`. It can be performed for different "degrees of uncertainty", hence the model parameters can be treated as fixed or random. We will consider a model without nugget and including uncertainty in the mean, sill and range parameters.

```r
baye.model <- krige.bayes(
  gdata2,
  loc = matrix(
    c(0.2, 0.6, 0.2, 1.1, 0.2, 0.3, 1.0, 1.1),
    ncol=2
  ),
  prior = prior.control(
    phi.discrete = seq(0,5,l=101), phi.prior="rec"
  ),
  output=output.control(n.post=5000)
)
```

```
## krige.bayes: model with constant mean
## krige.bayes: computing the discrete posterior of phi/tausq.rel
## krige.bayes: computing the posterior probabilities.
##              Number of parameter sets:  101
## 1, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101,
##
## krige.bayes: sampling from posterior distribution
## krige.bayes: sample from the (joint) posterior of phi and tausq.rel
##              [,1]   [,2] [,3]
## phi          0.25    0.3 0.35
## tausq.rel    0.00    0.0 0.00
## frequency 3540.00 1451.0 9.00
##
## krige.bayes: starting prediction at the provided locations
## krige.bayes: phi/tausq.rel samples for the predictive are same as for the posterior
## krige.bayes: computing moments of the predictive distribution
## krige.bayes: sampling from the predictive
##              Number of parameter sets:  3
## 1, 2, 3,
## krige.bayes: preparing summaries of the predictive distribution
```

5. Differences between the two methods of estimation

```r
print("Bayesian Methods")
```

```
## [1] "Bayesian Methods"
```

```r
print(summary(baye.model))
```

```
##              Length Class                 Mode
## posterior      6    posterior.krige.bayes list
## predictive     7    -none-                list
## prior          4    prior.geoR            list
## model          6    model.geoR            list
## .Random.seed 626    -none-                numeric
## max.dist       1    -none-                numeric
## call           5    -none-                call
```

```r
print("Spatial Models")
```

```
## [1] "Spatial Models"
```

```r
print(summary(ml.n))
```

```
## Summary of the parameter estimation
## -----------------------------------
## Estimation method: maximum likelihood
##
## Parameters of the mean component (trend):
##    beta
## 14.4155
##
## Parameters of the spatial component:
##    correlation function: exponential
##       (estimated) variance parameter sigmasq (partial sill) =  27.87
##       (estimated) cor. fct. parameter phi (range parameter)  =  9.787
##    anisotropy parameters:
##       (fixed) anisotropy angle = 0  ( 0 degrees )
##       (fixed) anisotropy ratio = 1
##
## Parameter of the error component:
##       (estimated) nugget =  5.337
##
## Transformation parameter:
##       (fixed) Box-Cox parameter = 1 (no transformation)
##
## Practical Range with cor=0.05 for asymptotic range: 29.3194
##
## Maximised Likelihood:
##    log.L n.params      AIC      BIC
##  "-3680"      "4"   "7368"   "7389"
##
## non spatial model:
##    log.L n.params      AIC      BIC
##  "-4713"      "2"   "9430"   "9441"
##
## Call:
## likfit(geodata = gdata2, ini.cov.pars = c(1, 0.5), nugget = 0.5)
```

From the comparison, the spatial models seems to be the better fit than the bayesian method, it has a Practical Range with cor=0.05 for asymptotic range: 29.3194. The bayesian method has a mean of 15.23 and a variance of 22.37 while the spatial model has mean component of 14.4155.

B: Time Series Modelling

1. Which equation corresponds to which plot

- Fig A: equation (ii); the introduced coefficient $\rho = 0.02$ the figure matches the equation since the cycles are narrow and almost close to each other. Where error (white noise) at time point remains constant, the equation differs with equation (iii) since it has a smaller ρ coefficient.

- Figure B: equation (iii), as explained in A above, the ρ coefficient in B is larger, resulting to broader cycles. Where the error $\in_t$ is constant.
- Figure C: equation (1) this figure shows an aggregated time series that can be based on quaterly or seasonal data hence resulting to an increase point with increase in aggregated time point.
- Figure D: equation (v) this because the figure depends on 2 previous time points ($0.1X_{t-1}$, $0.9X_{t-2}$) to determine the current position. Based on this trend, the size of the cycles increases almost doubly with the previous trends
- Figure E: equation (iv) this is because of a negative in the ρ coefficient, as a result of this, it will form a decrease in the trend of the graph.

2. Appropriate ARMA model for the five series

- The PCAF, Series A has cut off on PACF curve after 2nd lag which means this is mostly an Autoregressive AR(2) Model.
- In Series, the graph has a cut off on ACF, Series B curve after 2nd lag which means this is mostly a Moving Average MA(2) process.
- Series C is the same as the Series B in that the graph has a cut off on ACF, Series C curve after 2nd lag which means this is mostly a Moving Average MA(2) Process.
- In Series D, both ACF and PACF are demonstrating a gradual decreasing pattern (slow decay) hence the ARMA (1,1) model would be appropriate for the series.
- In Series E, PACF Series E cuts off on PACF curve after the 1st lag which means this is mostly an Autoregressive AR(1) Model

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")
head(overturning)
```

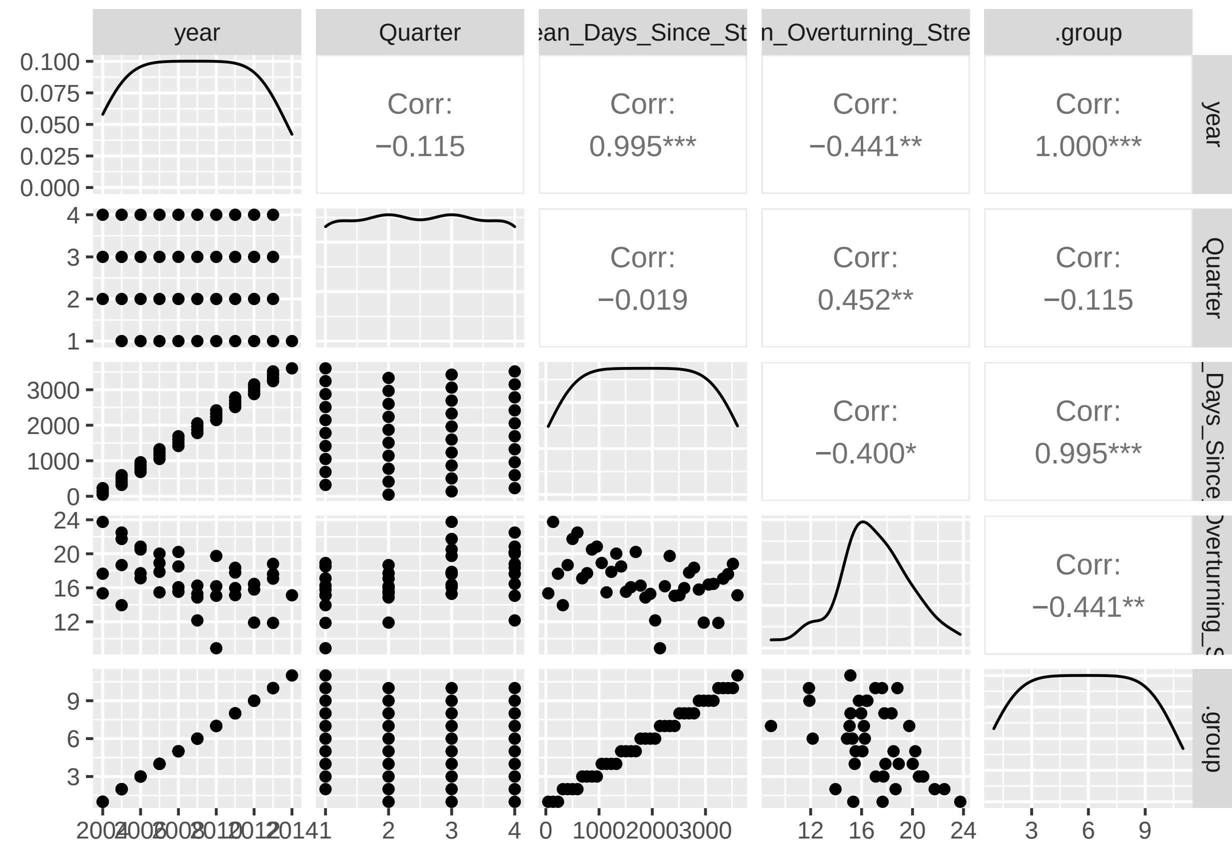
```
##   year month day hour Quarter Days_since_start Overturning_Strength
## 1 2004     4   2    0       2              1.0             9.689933
## 2 2004     4   2   12       2              1.5            10.193495
## 3 2004     4   3    0       2              2.0            10.660849
## 4 2004     4   3   12       2              2.5            11.077229
## 5 2004     4   4    0       2              3.0            11.432414
## 6 2004     4   4   12       2              3.5            11.721769
```

a. averaging the data to quaterly means

```
quaterly_means <- overturning %>%
  group_by(year,Quarter) %>%
  summarise(
    Mean_Days_Since_Start=mean(Days_since_start),
    Mean_Overturning_Strength=mean(Overturning_Strength)
  )

summary(quaterly_means)
```

```
##       year         Quarter     Mean_Days_Since_Start Mean_Overturning_Strength
##  Min.   :2004   Min.   :1.00   Min.   :  45.75       Min.   : 8.89
##  1st Qu.:2006   1st Qu.:1.75   1st Qu.: 935.75       1st Qu.:15.33
##  Median :2009   Median :2.50   Median :1826.00       Median :16.77
##  Mean   :2009   Mean   :2.50   Mean   :1825.84       Mean   :16.98
##  3rd Qu.:2011   3rd Qu.:3.25   3rd Qu.:2715.75       3rd Qu.:18.70
##  Max.   :2014   Max.   :4.00   Max.   :3602.00       Max.   :23.75
```

```
quaterly_means %>%
  ggpairs()
```



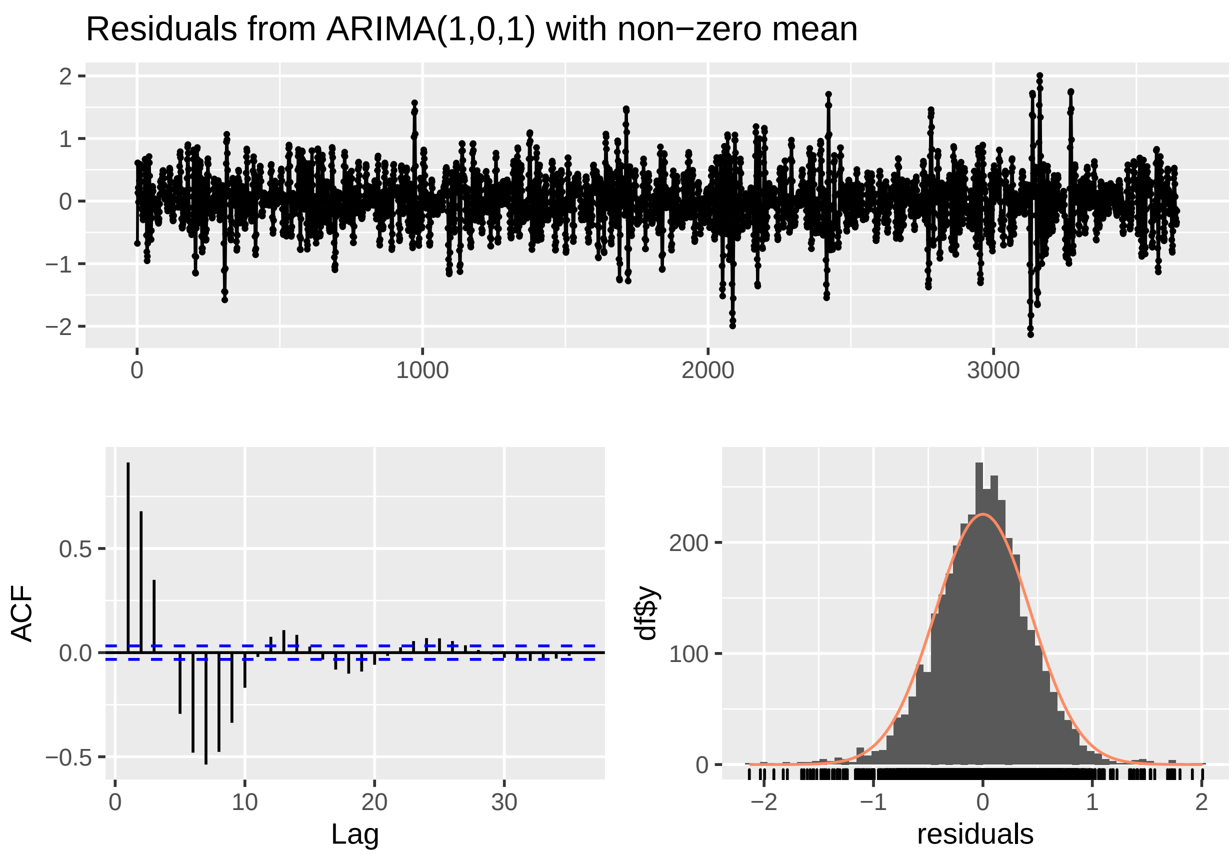
There are potential outliers in the average quaterly overturning strengths, from the distribution ploted, the graph is left skewed.

```
quaterly_means_ts <- ts(quaterly_means)
summary(quaterly_means_ts)
```


```
##       year         Quarter     Mean_Days_Since_Start Mean_Overturning_Strength
##  Min.   :2004   Min.   :1.00   Min.   :  45.75       Min.   : 8.89
##  1st Qu.:2006   1st Qu.:1.75   1st Qu.: 935.75       1st Qu.:15.33
##  Median :2009   Median :2.50   Median :1826.00       Median :16.77
##  Mean   :2009   Mean   :2.50   Mean   :1825.84       Mean   :16.98
##  3rd Qu.:2011   3rd Qu.:3.25   3rd Qu.:2715.75       3rd Qu.:18.70
##  Max.   :2014   Max.   :4.00   Max.   :3602.00       Max.   :23.75
```

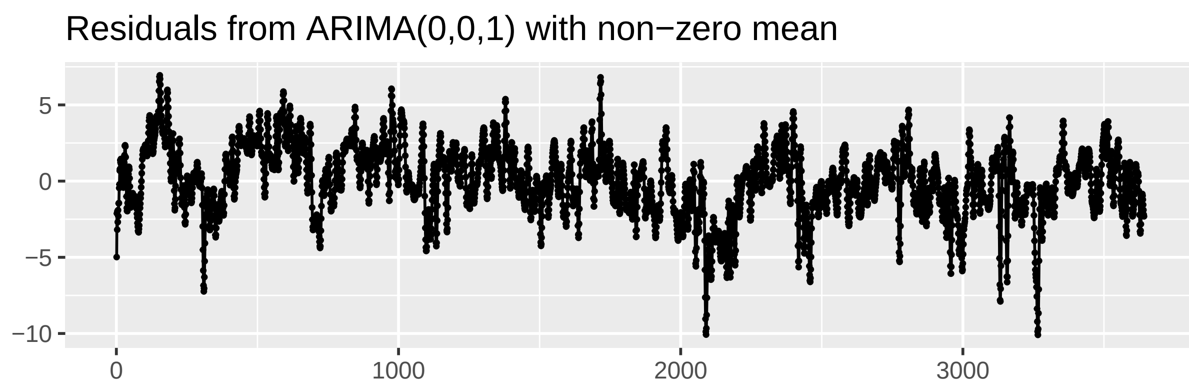
b. Fitting ARMA and ARIMA model to the data

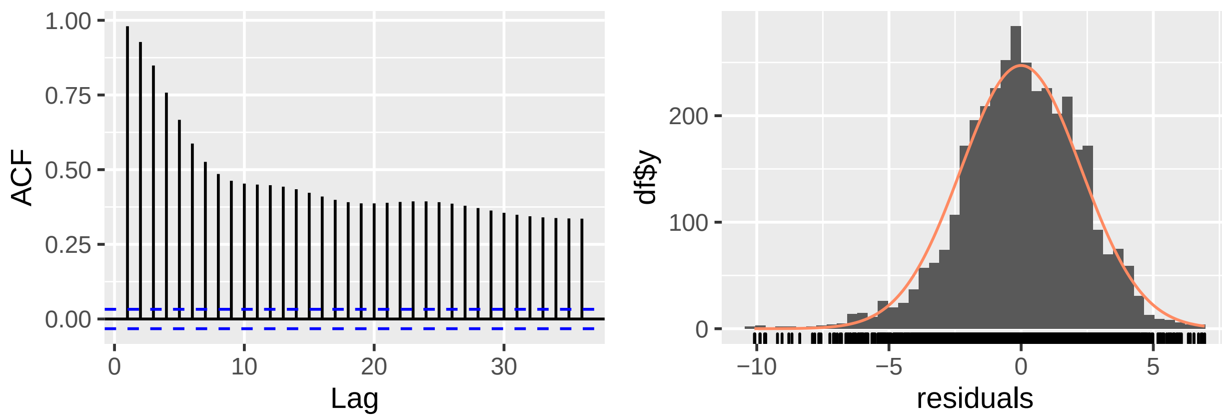
```
overturning2 <- overturning %>%
  mutate(date=paste(year,month,day, sep="-"))%>%
  dplyr::select(date,Overturning_Strength)%>%
  mutate(date=ymd(date)) %>%
  group_by(date)%>%
  summarise(Overturning_Strength=mean(Overturning_Strength))

overturning2_xts <- as.xts(overturning2[,-1], order.by = overturning2$date)
AR1 <- arima(overturning2_xts, c(1,0,1))
checkresiduals(AR1)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1) with non-zero mean
## Q* = 8722, df = 7, p-value < 2.2e-16
##
## Model df: 3.   Total lags used: 10
```

```
MA <- arima(overturning2_xts, order = c(0,0,1))
checkresiduals(MA)
```






```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(0,0,1) with non-zero mean
## Q* = 17646, df = 8, p-value < 2.2e-16
##
## Model df: 2.   Total lags used: 10
```

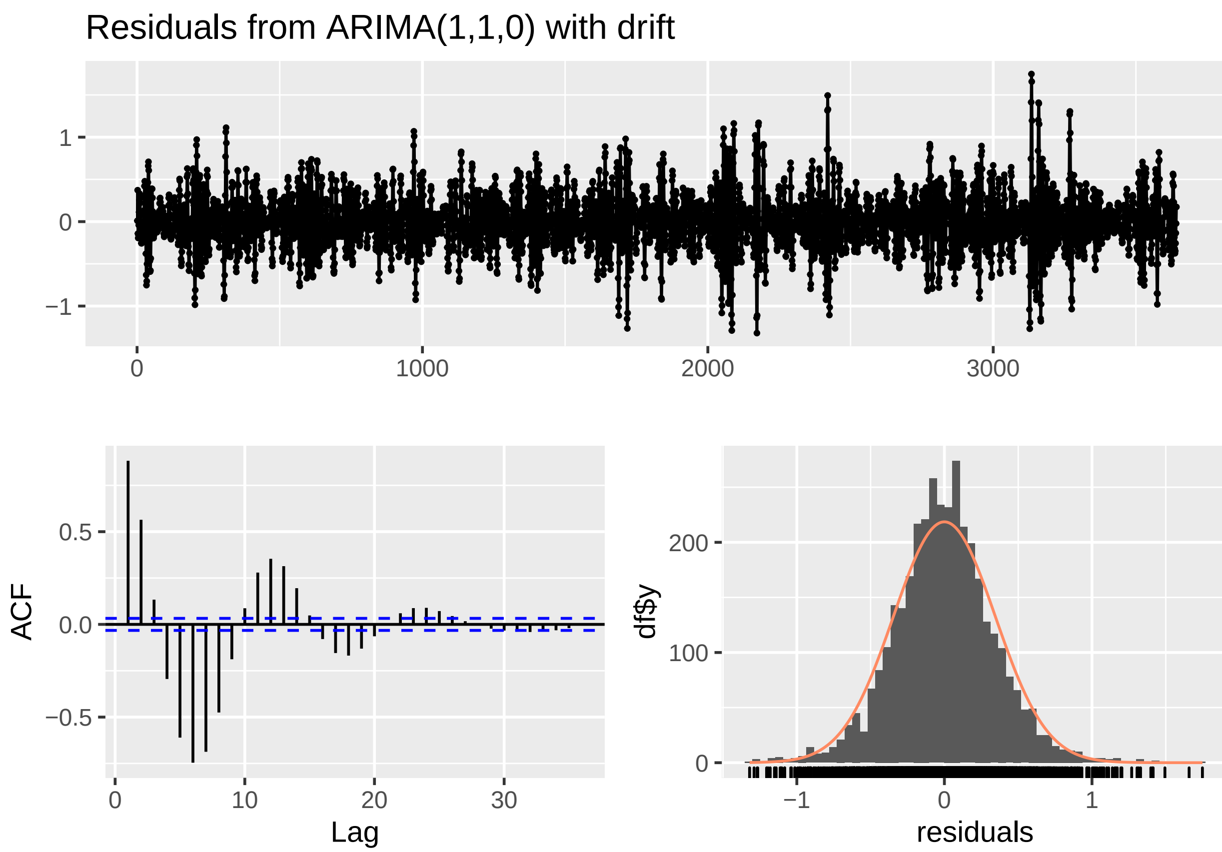
```
# Find the best Model
best.model <- auto.arima(overturning2_xts)
summary(best.model)
```

```
## Series: overturning2_xts
## ARIMA(1,1,0) with drift
##
## Coefficients:
##          ar1   drift
##       0.9156  0.0007
## s.e.  0.0066  0.0677
##
## sigma^2 estimated as 0.1196:  log likelihood=-1299.89
## AIC=2605.79   AICc=2605.79   BIC=2624.39
##
## Training set error measures:
##                           ME      RMSE       MAE       MPE      MAPE       MASE
```

```
## Training set -0.0002536988 0.3456563 0.2629011 0.3296394 2.038565 0.4028823
##                     ACF1
## Training set 0.8824904
```

The best and appropriate model is ARIMA(1, 1, 0) with Drift that has 2.04 MAPE. The plot below is it's residuals.

```
checkresiduals(best.model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0) with drift
## Q* = 10477, df = 8, p-value < 2.2e-16
##
## Model df: 2.   Total lags used: 10
```

```
# predict 6 3-month periods from April 2014 to september 2015
arima.forecast <- forecast(best.model, h=6)
arima.forecast
```

```
##      Point Forecast     Lo 80    Hi 80     Lo 95    Hi 95
## 3643       12.09989 11.656736 12.54305 11.422141 12.77765
## 3644       11.74972 10.792076 12.70735 10.285132 13.21430
## 3645       11.42914  9.877809 12.98046  9.056585 13.80169
```

```
## 3646       11.13566  8.935142 13.33618  7.770257 14.50107
## 3647       10.86700  7.977842 13.75617  6.448412 15.28560
## 3648       10.62107  7.015310 14.22683  5.106536 16.13560
```

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

```
fn <- function(parm) {
    dlmModPoly(order = 1, dV = exp(parm[1]), dW = exp(parm[2]))
}
dlm.fit <- dlmMLE(overturning2_xts, rep(0, 2), build = fn, hessian = TRUE)
dlm.forcast <- dlmForecast(
  dlmFilter(
    overturning2_xts,
    mod = fn(dlm.fit$par)
  ), nAhead=6
)
dlm.forcast$a
```

```
##      2014-03-22
## [1,]    12.4824
## [2,]    12.4824
## [3,]    12.4824
## [4,]    12.4824
## [5,]    12.4824
## [6,]    12.4824
```

d. Results Comparison

```
preds_comparison <- data.frame(ARIMA=arima.forecast$mean, dlm.forcast$a)
names(preds_comparison) <- c("ARIMA_Preds", "DLM_Preds")
preds_comparison
```

```
##   ARIMA_Preds DLM_Preds
## 1    12.09989   12.4824
## 2    11.74972   12.4824
## 3    11.42914   12.4824
## 4    11.13566   12.4824
## 5    10.86700   12.4824
## 6    10.62107   12.4824
```

The DLM hasn't yielded good results (it has predicted same values, 12.4824) as compared to ARIMA(1,1,0), ARIMA has the best fit has has smaller error as compared to DLM. Hence for future predictions, ARIMA model would be the preferred model.

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

```
# Read the csv datasets
metadataCA <- read.csv("metadataCA.csv")
maxtempcalifornia <- read.csv("MaxTempCalifornia.csv")

# preview the heads
head(metadataCA)
```

```
##      ï..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
```

```
# Tranform the data from wide to long
maxtempcalifornia_long <- maxtempcalifornia %>%
  gather(Location, Max_Temp, -c(X))
maxtempcalifornia_long$Location <- maxtempcalifornia_long$Location %>%
  str_replace("\\.", " ")
maxtempcalifornia_long$Date <- ymd(maxtempcalifornia_long$X)
maxtempcalifornia_long <- maxtempcalifornia_long %>%
  subset(select=-X)
head(maxtempcalifornia_long)
```

```
##        Location Max_Temp       Date
## 1 San Francisco     14.4 2012-01-01
## 2 San Francisco     12.8 2012-01-02
## 3 San Francisco     11.7 2012-01-03
## 4 San Francisco     13.9 2012-01-04
## 5 San Francisco     16.1 2012-01-05
## 6 San Francisco     13.3 2012-01-06
```

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

```
summary(metadataCA)
```

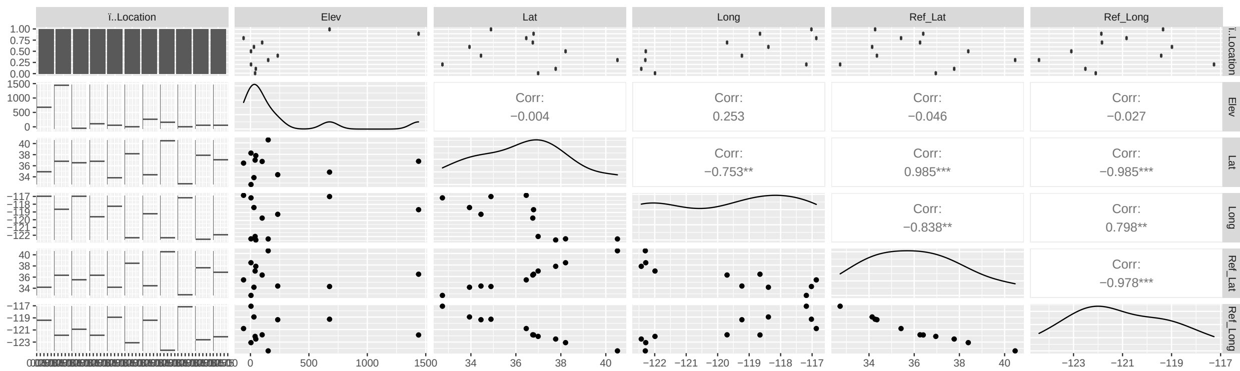
```
##  ï..Location             Elev             Lat              Long
##  Length:11          Min.   : -59.1   Min.   :32.73   Min.   :-122.4
##  Class :character   1st Qu.:  17.1   1st Qu.:34.67   1st Qu.:-122.1
##  Mode  :character   Median :  45.7   Median :36.75   Median :-119.2
##              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

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

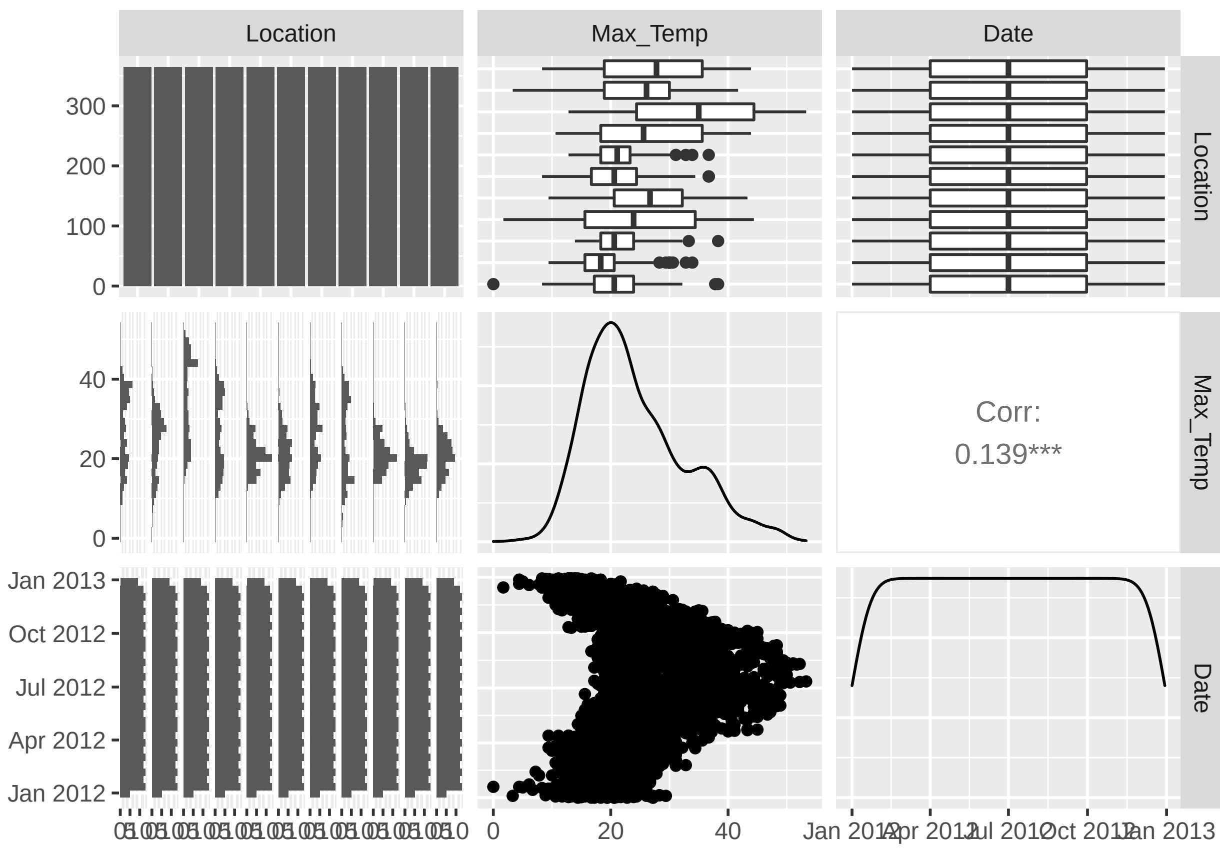
From this summary, we can see that the maximum value in `Elev`, and `Max_Temp` columns are very extreme, this shows that there are outliers in the datasets. The scatter matrix below reveals more of the datasets.

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



From the above diagram, checking `Elev` distribution is right skewed, this is as a result of outliers.

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

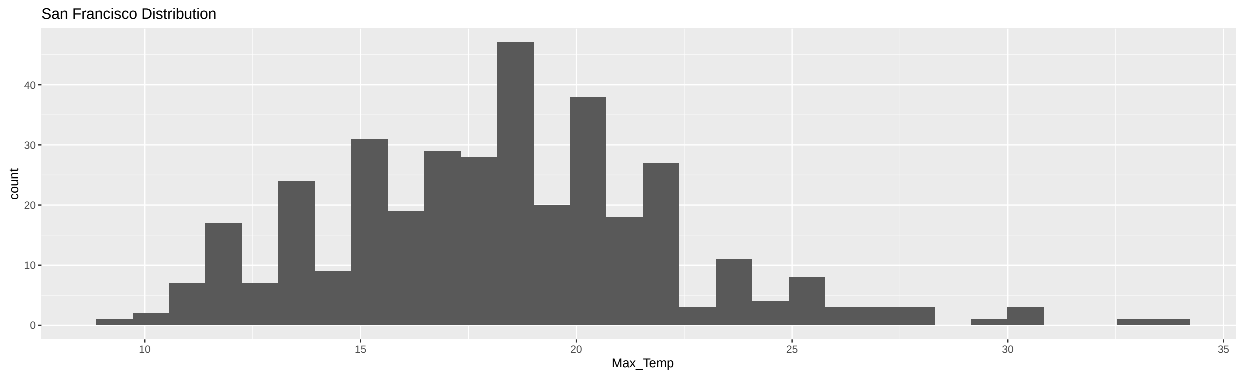


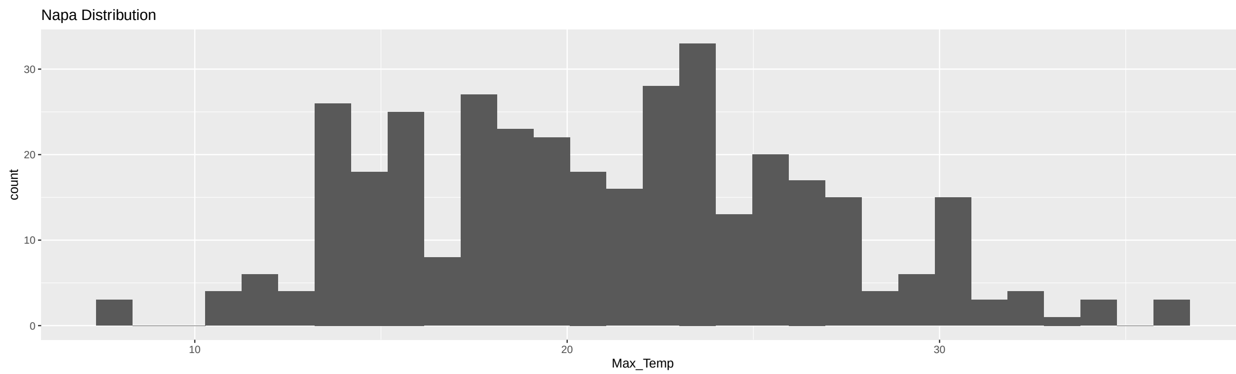
The `Max_Temp` Column was almost a normal distribution were it not for the outliers present forcing it to be a little bit right skewed.

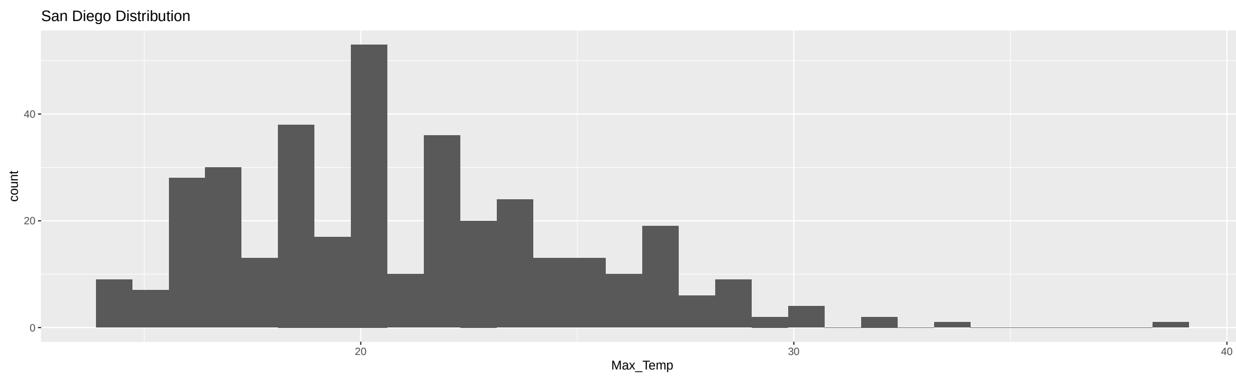
2. Distributions of the data at each location

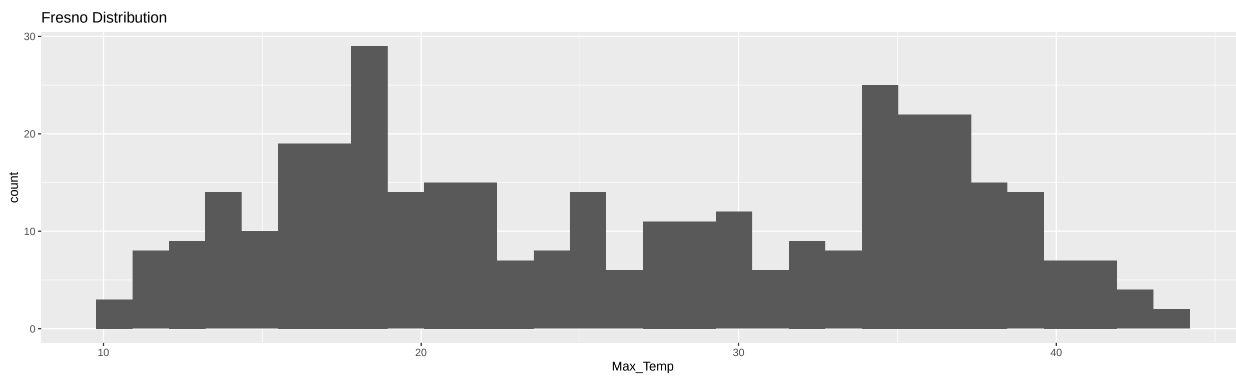
```
for(location in unique(maxtempcalifornia_long$Location)){
  p <- maxtempcalifornia_long %>%
    filter(Location==location) %>%
```

```
    ggplot(aes(x=Max_Temp))+
    geom_histogram()+
    ggtitle(paste(location,"Distribution"))
  print(p)
}
```

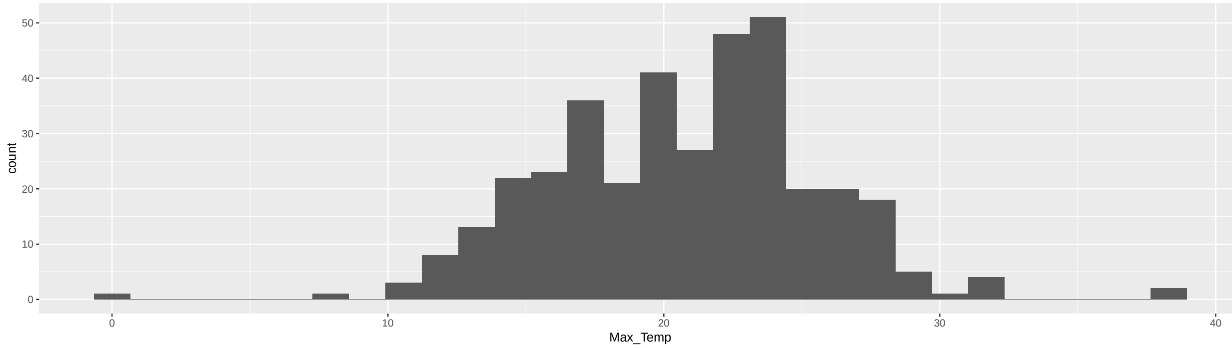




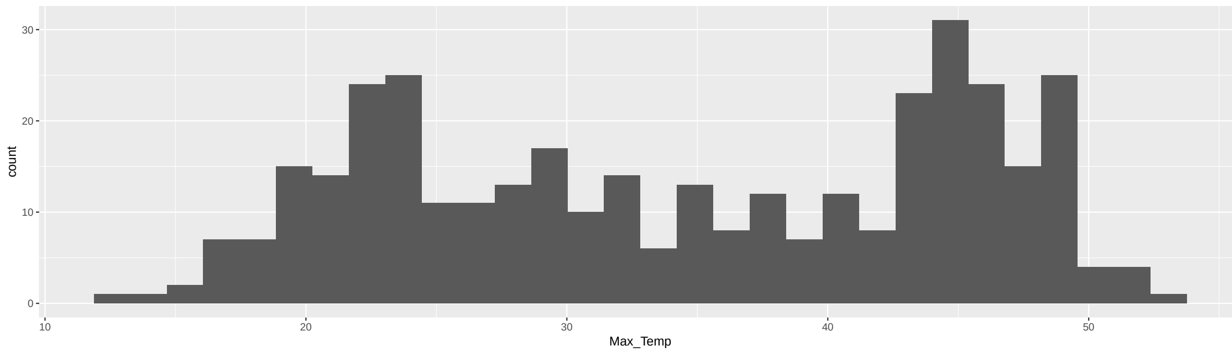




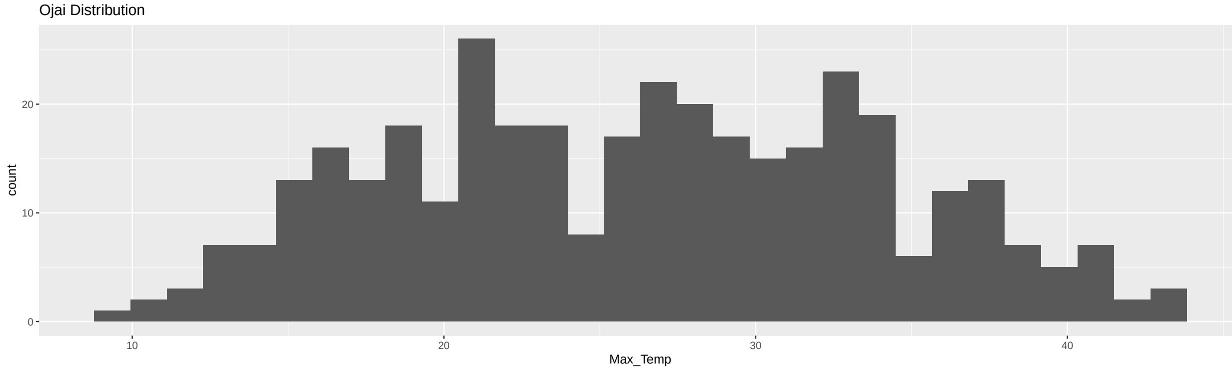
Santa Cruz Distribution



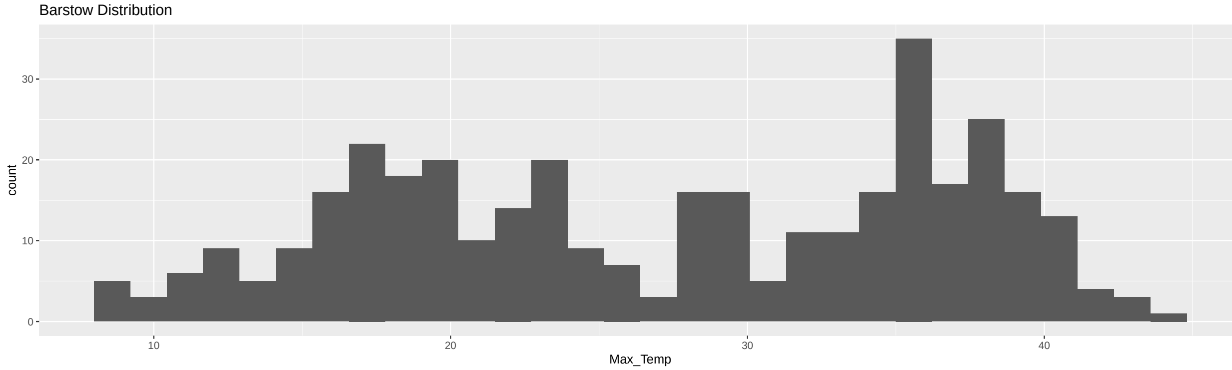
Death Valley Distribution

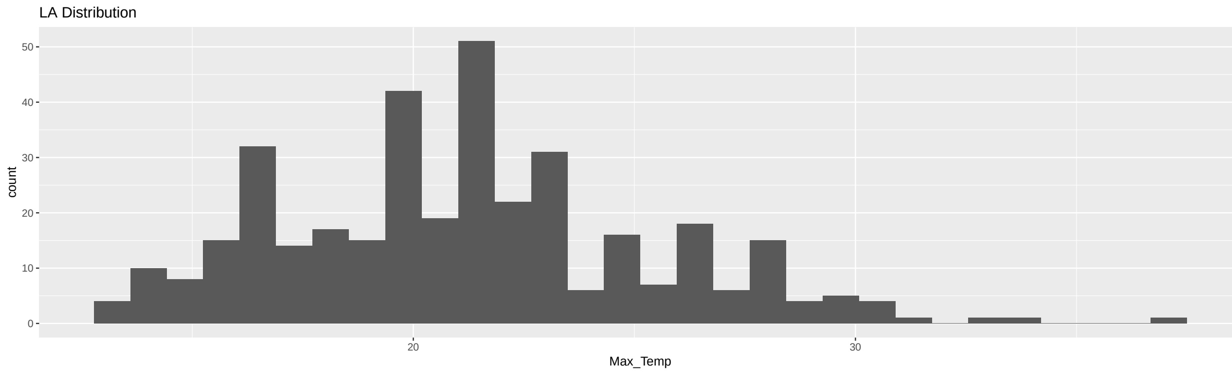


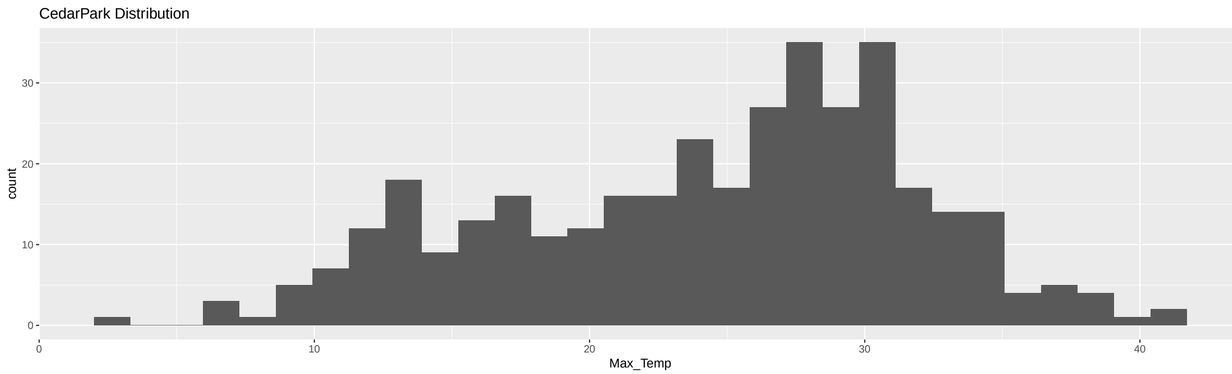
Ojai Distribution

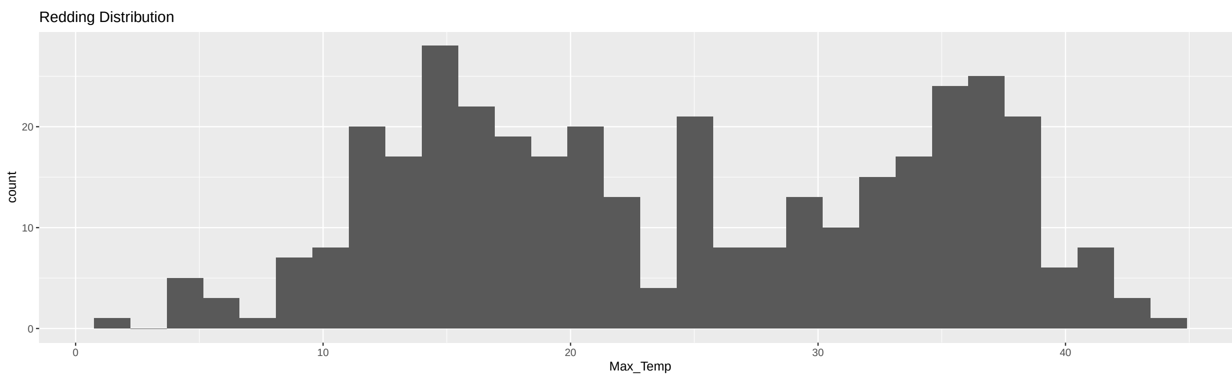


Barstow Distribution



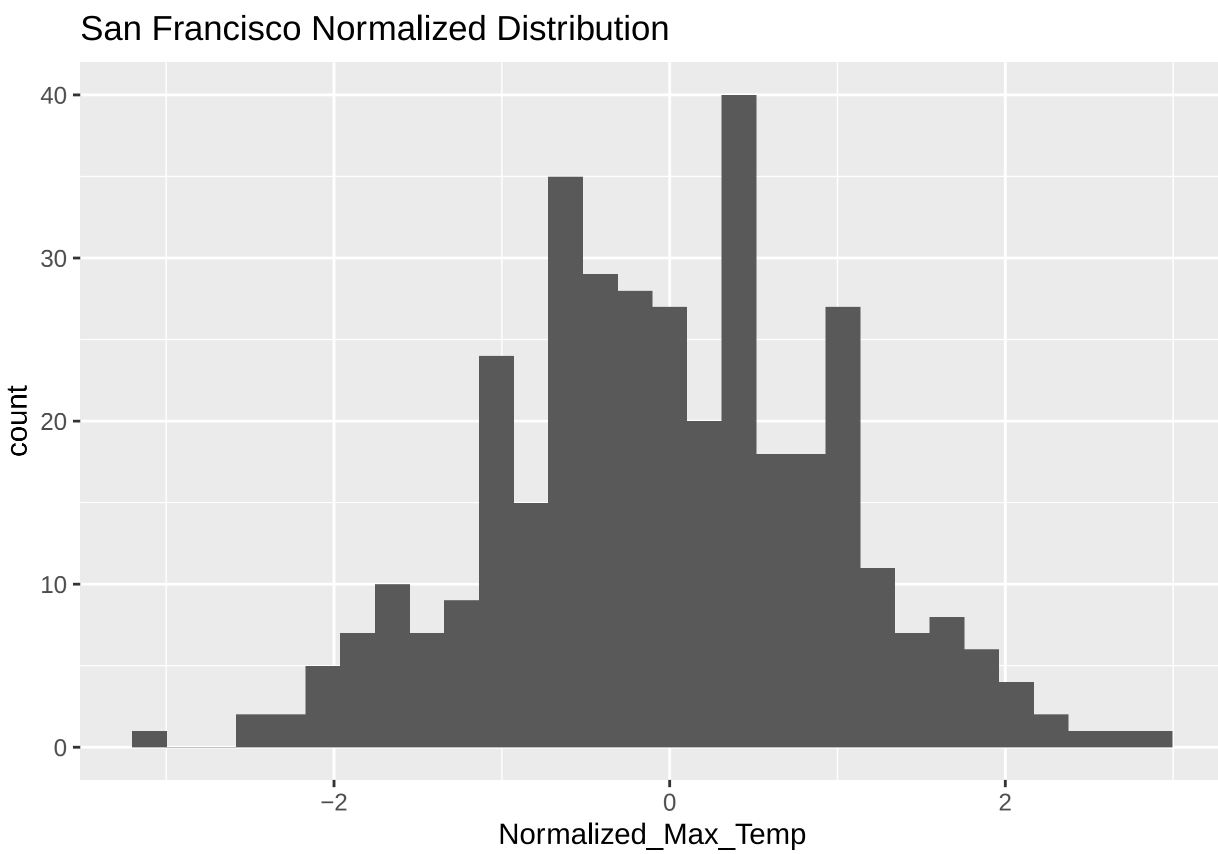


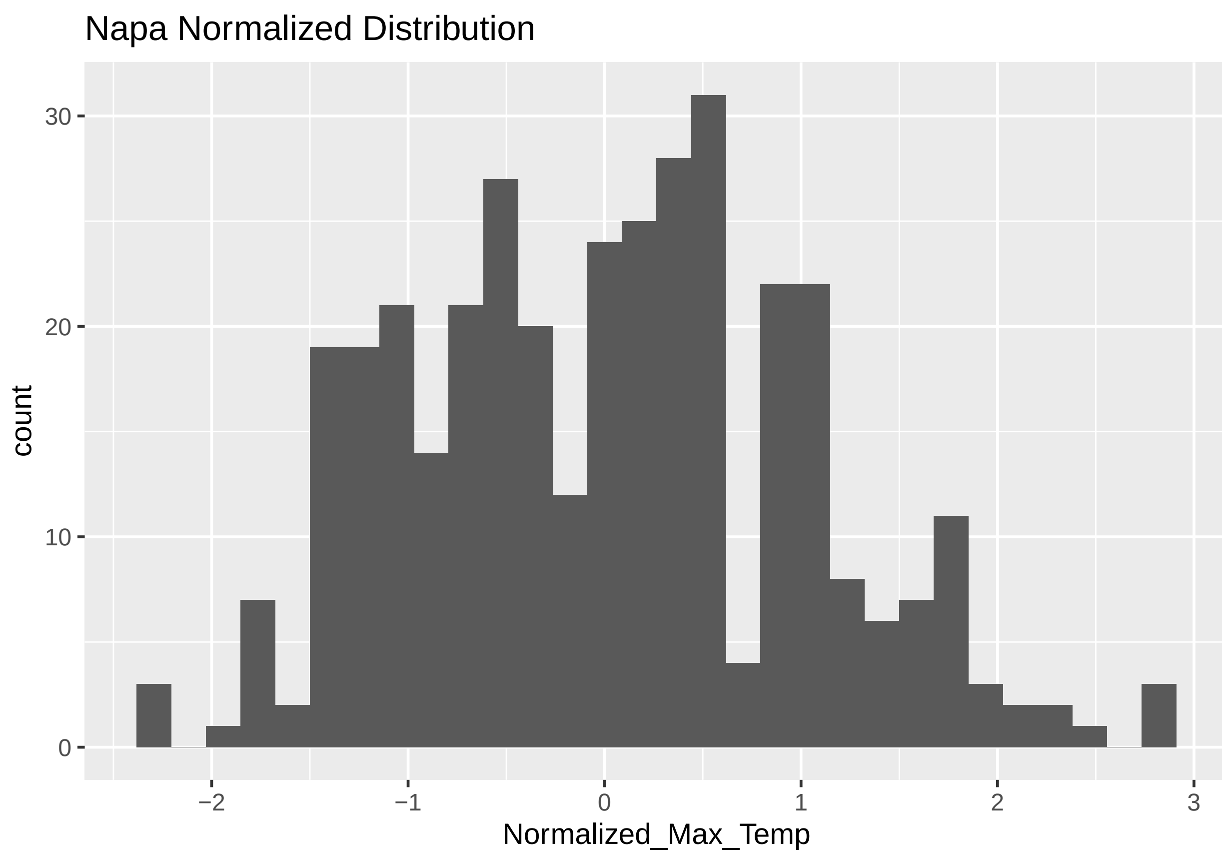


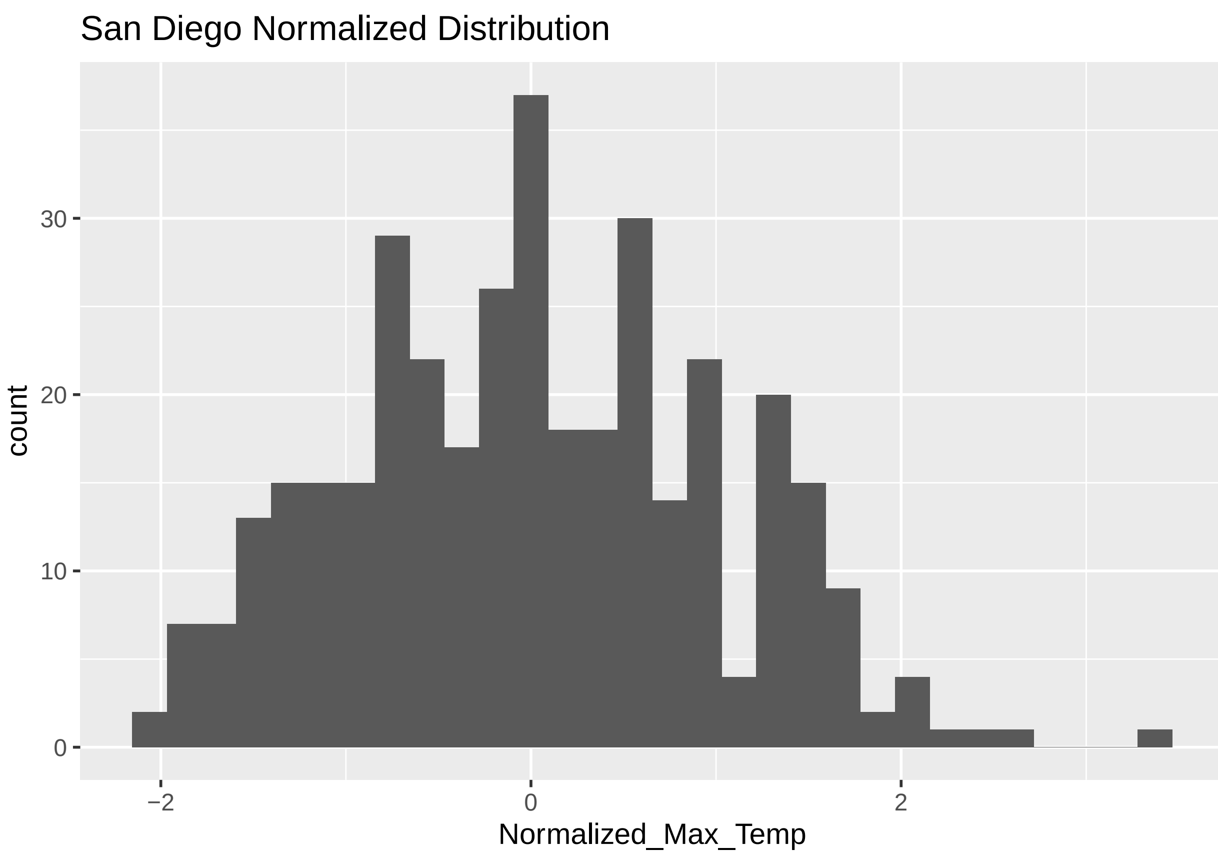


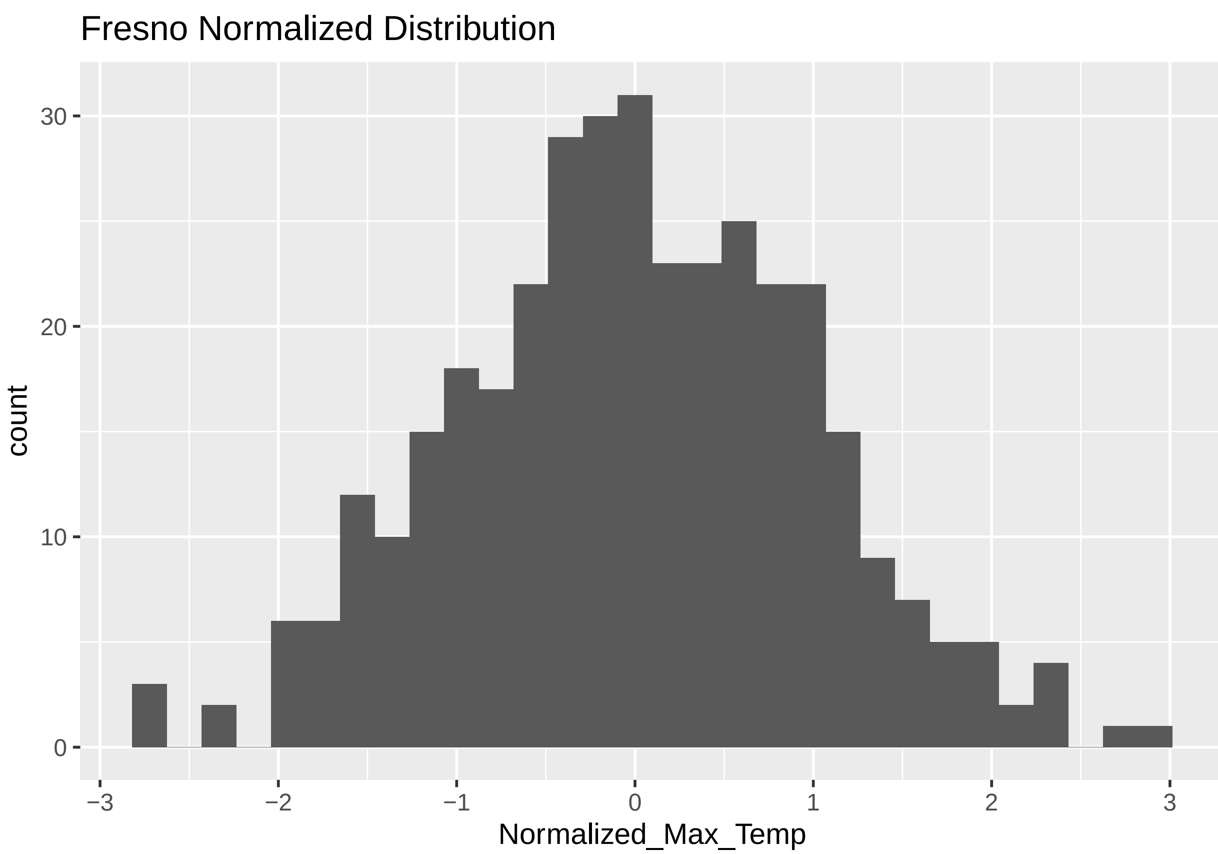
For each location, the data doesn't look Normally distributed, therefore transformation for each site needs to be done. The **Ojai** location is almost normally distributed. For this transformation we will use `bestNormalize` package to transform the data at each site to be normally distributed

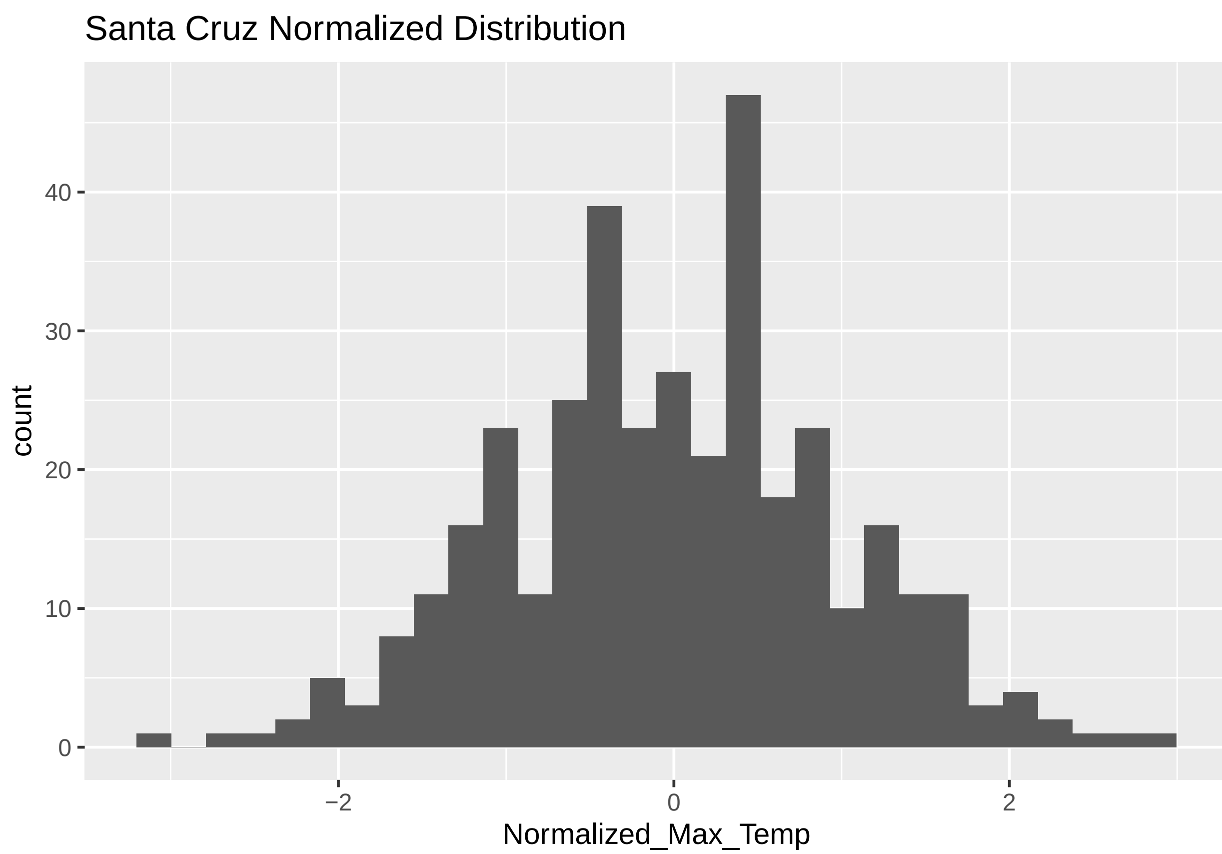
```
maxtempcalifornia_long$Normalized_Max_Temp <- 0
for(location in unique(maxtempcalifornia_long$Location)){
  maxtempcalifornia_long[maxtempcalifornia_long$Location==location, c("Normalized_Max_Temp")] <- bestNo
}
for(location in unique(maxtempcalifornia_long$Location)){
  p <- maxtempcalifornia_long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

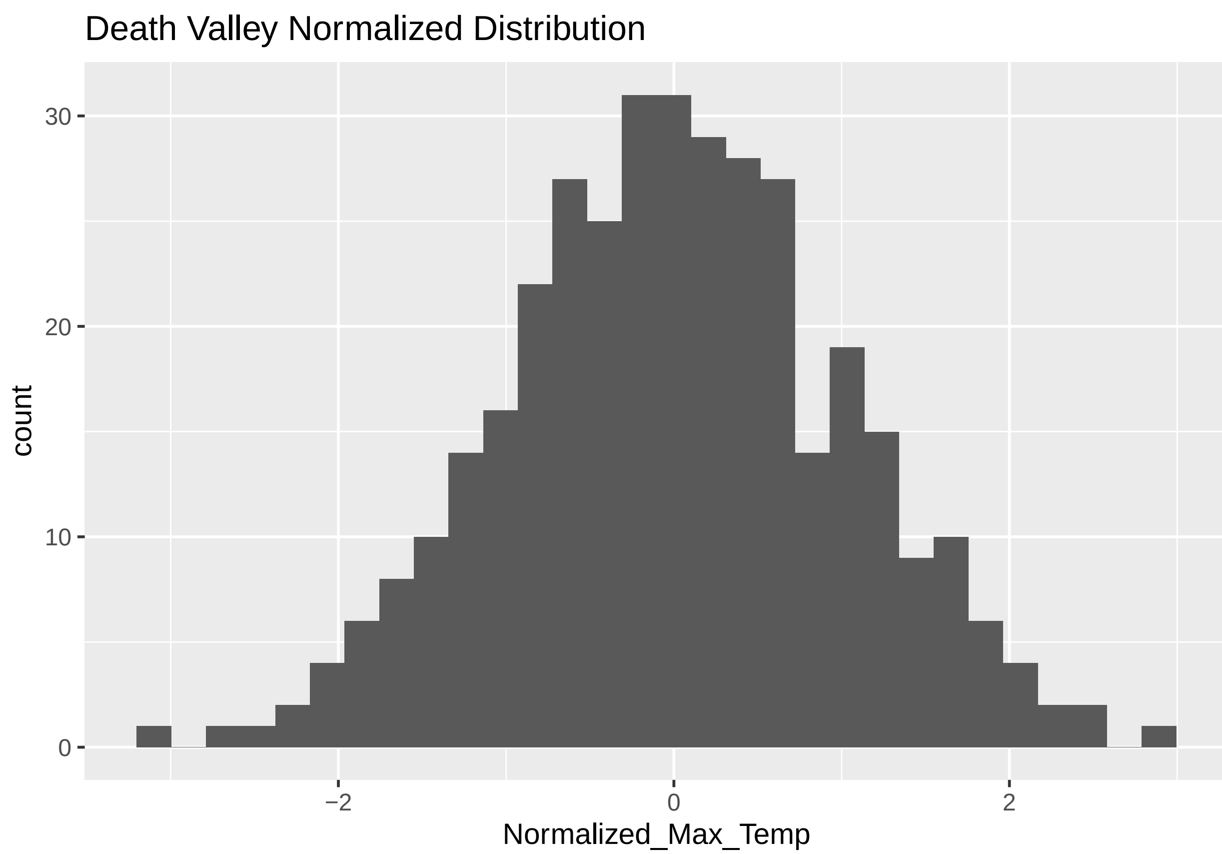

San Francisco Normalized Distribution
count
40
30
20
10
0
−2
0
2
Normalized_Max_Temp

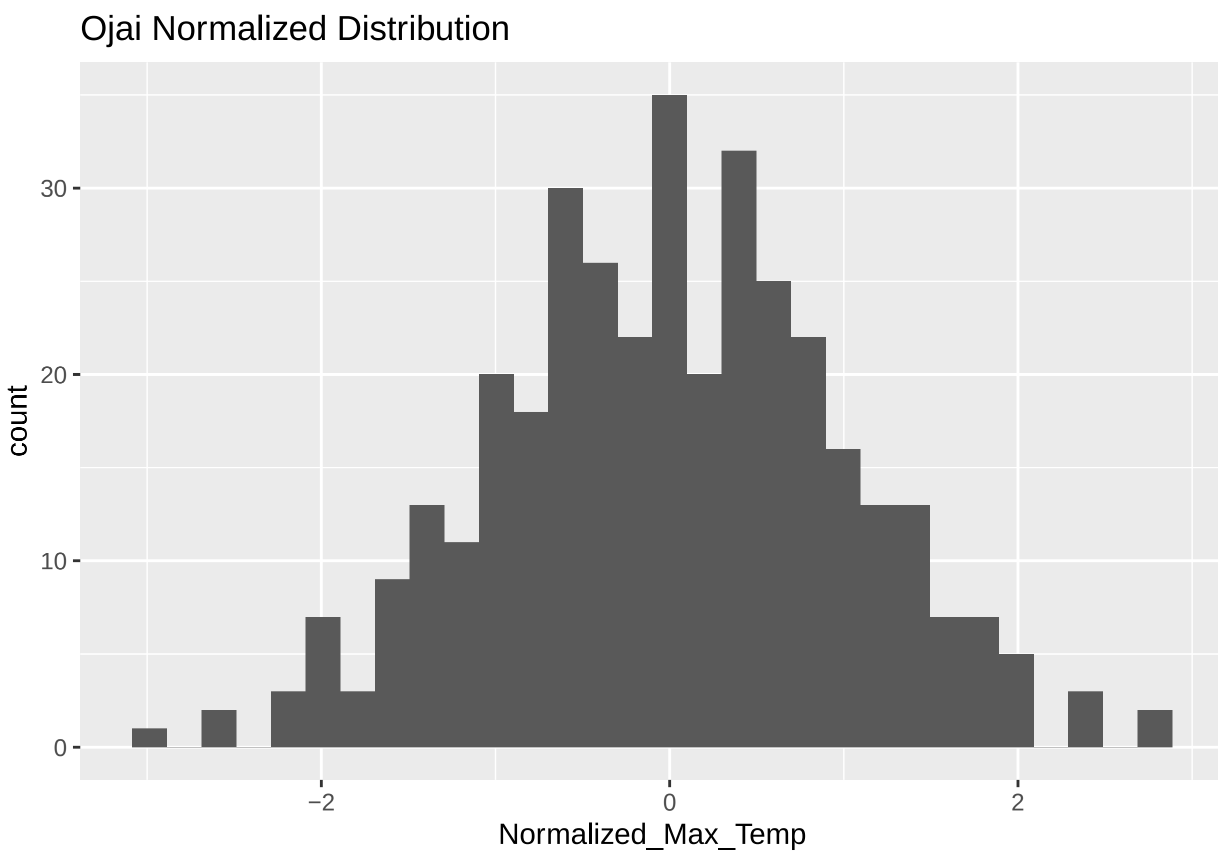

Napa Normalized Distribution
count
30
20
10
0
−2
−1
0
1
2
3
Normalized_Max_Temp


San Diego Normalized Distribution
count
30
20
10
0
−2
0
2
Normalized_Max_Temp

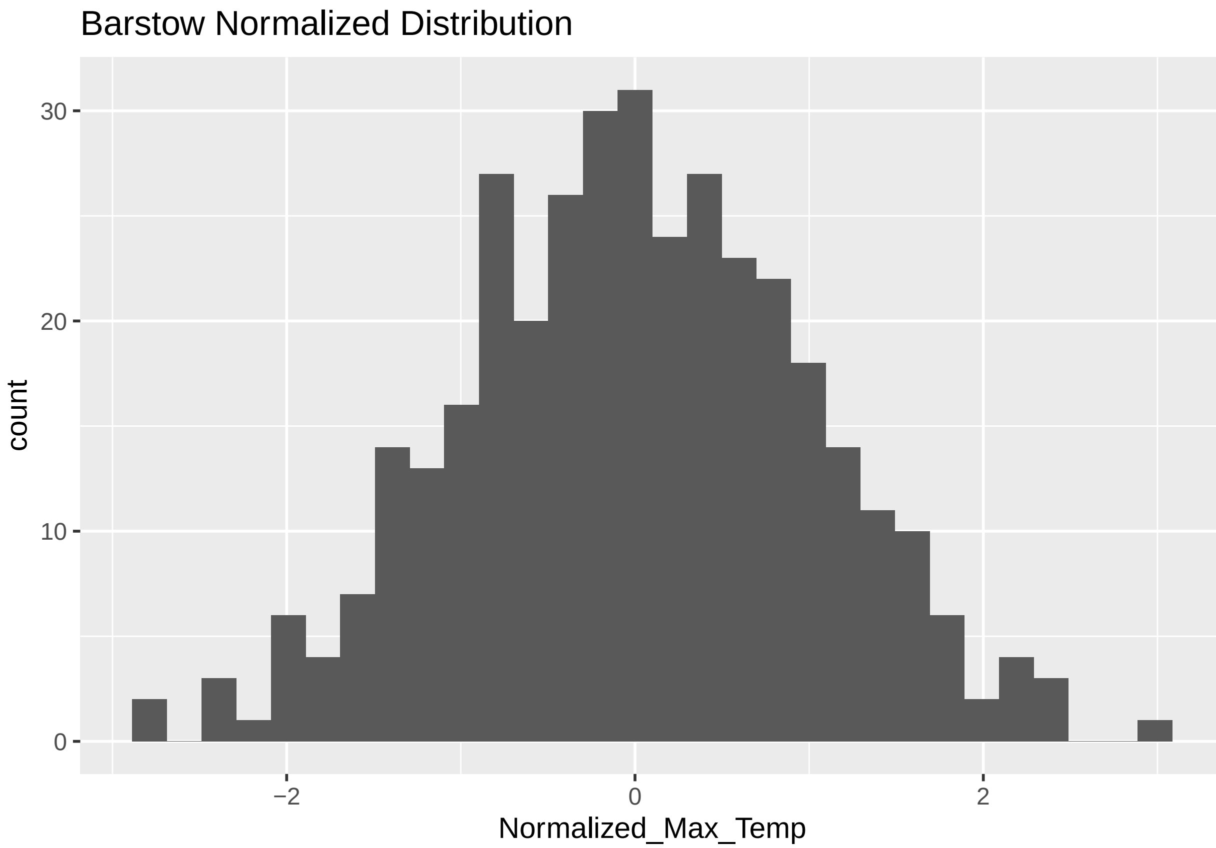

Fresno Normalized Distribution
count
30
20
10
0
−3
−2
−1
0
1
2
3
Normalized_Max_Temp

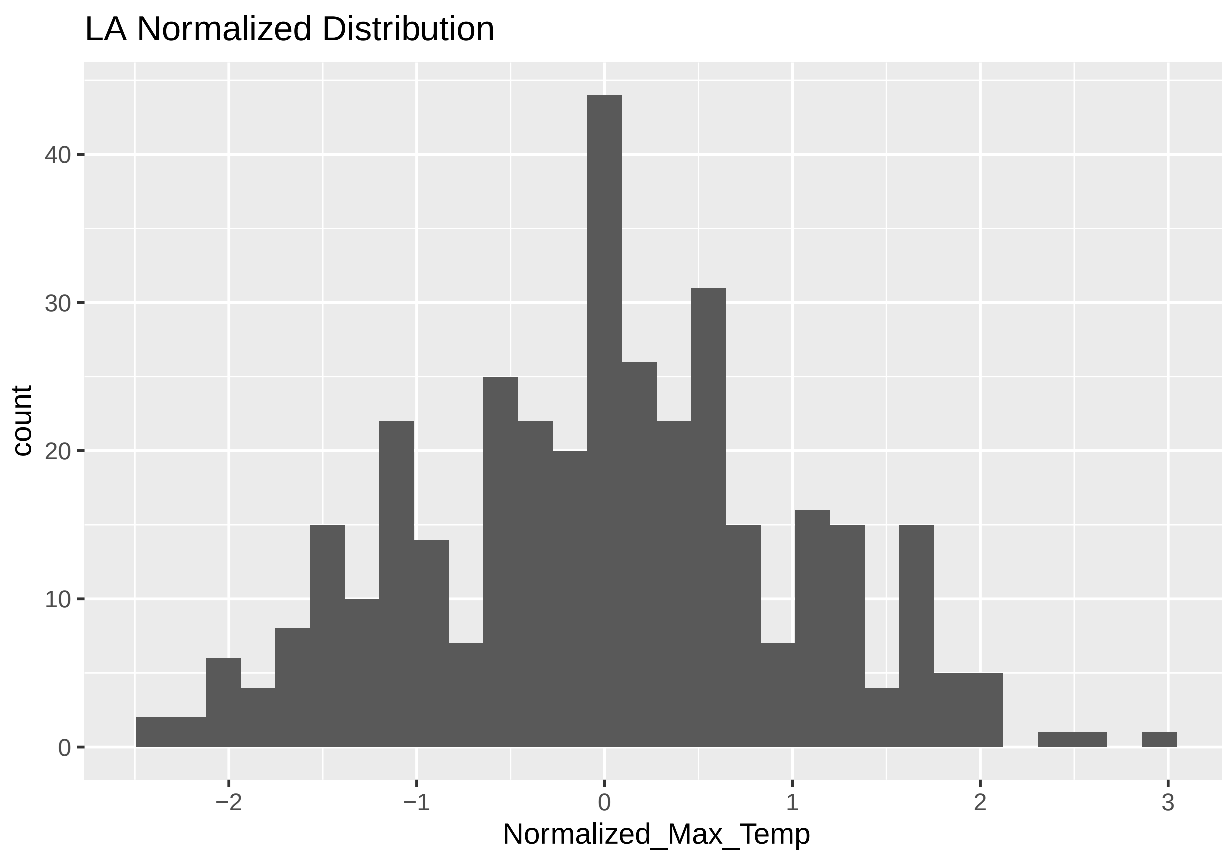

Santa Cruz Normalized Distribution
count
40
30
20
10
0
−2
0
2
Normalized_Max_Temp

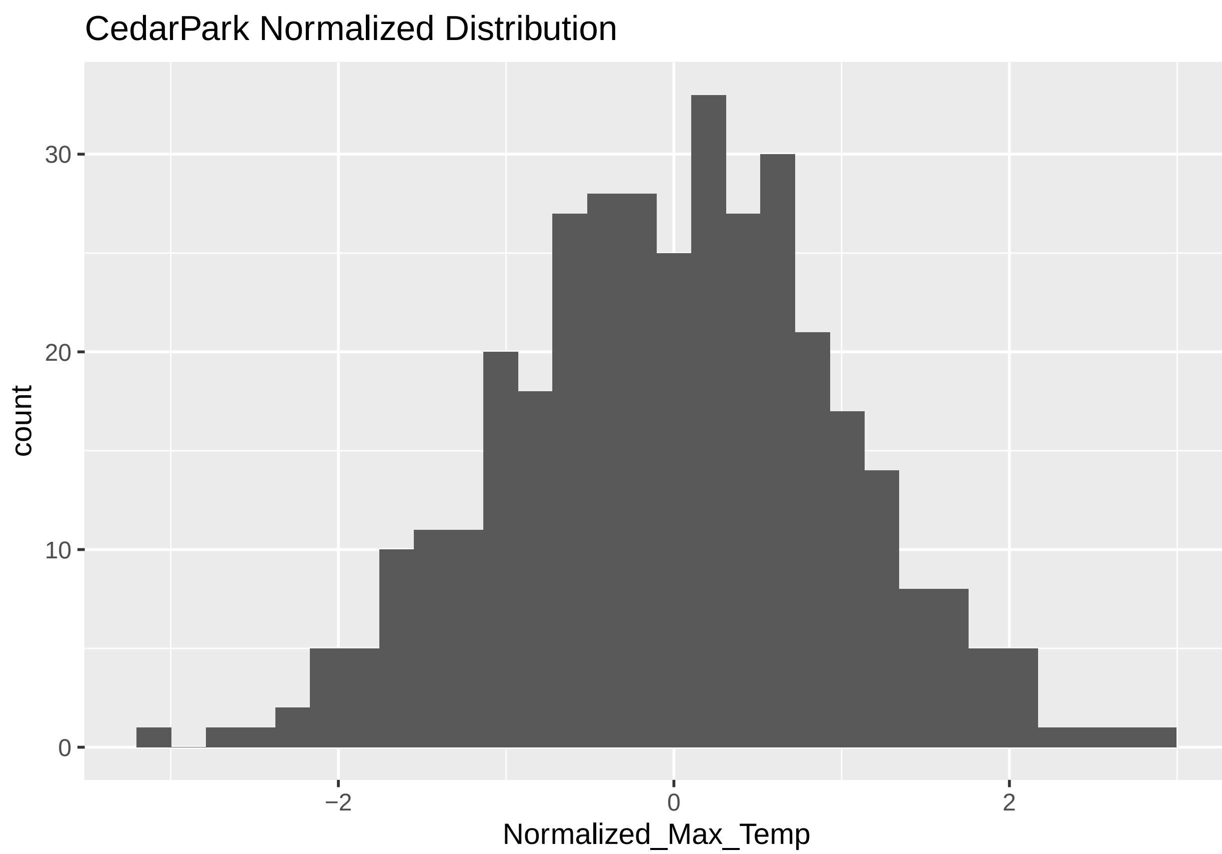

Death Valley Normalized Distribution
count
30
20
10
0
−2
0
2
Normalized_Max_Temp

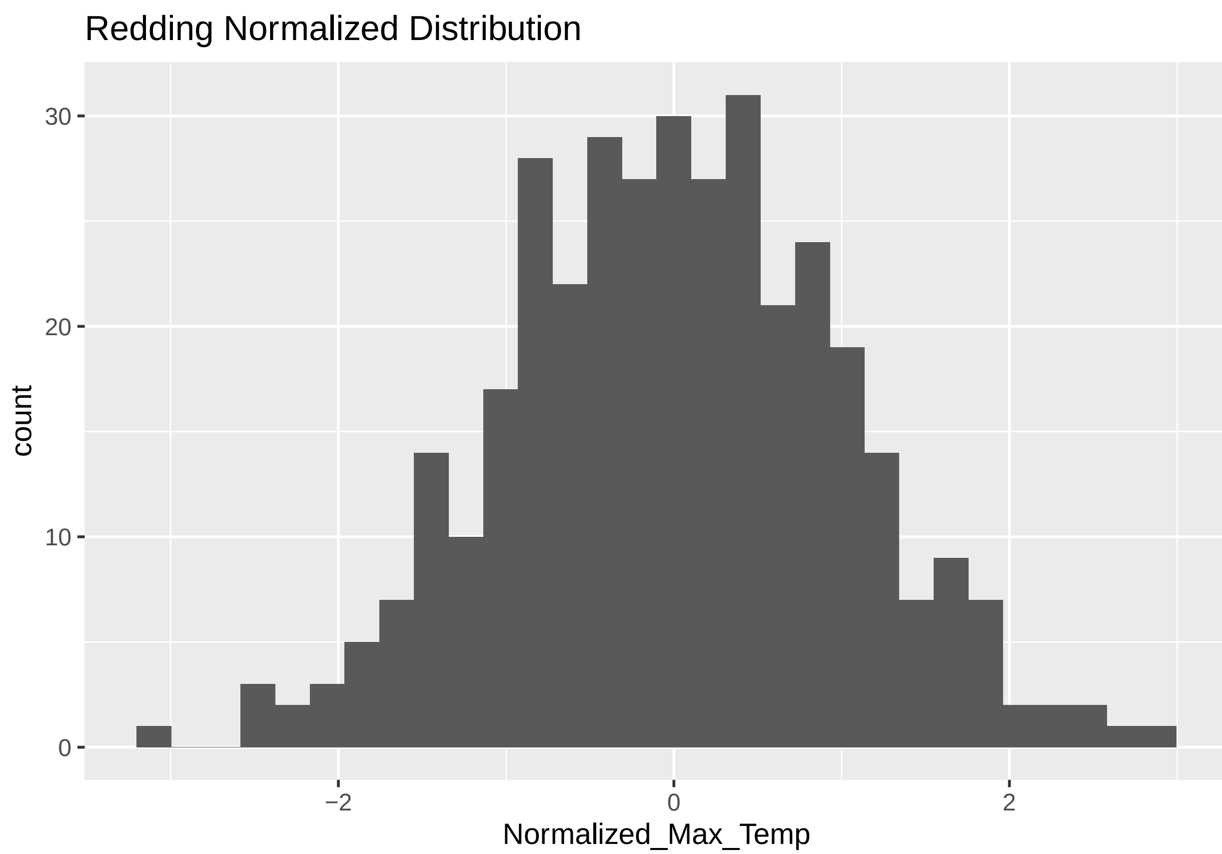


Ojai Normalized Distribution
count
30
20
10
0
−2
0
2
Normalized_Max_Temp


Barstow Normalized Distribution
30
20
10
0
count
−2
0
2
Normalized_Max_Temp


LA Normalized Distribution
count
40
30
20
10
0
−2
−1
0
1
2
3
Normalized_Max_Temp


CedarPark Normalized Distribution
30
20
10
0
count
−2
0
2
Normalized_Max_Temp

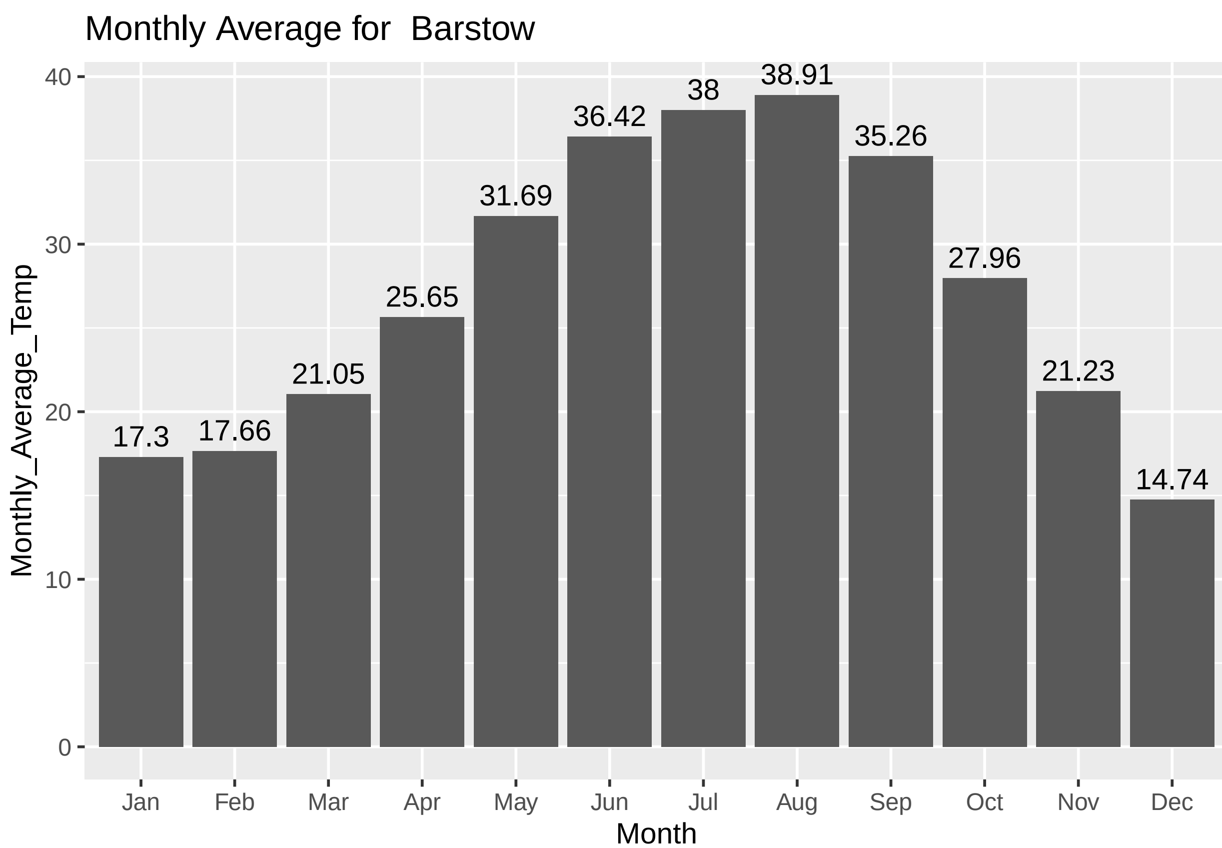


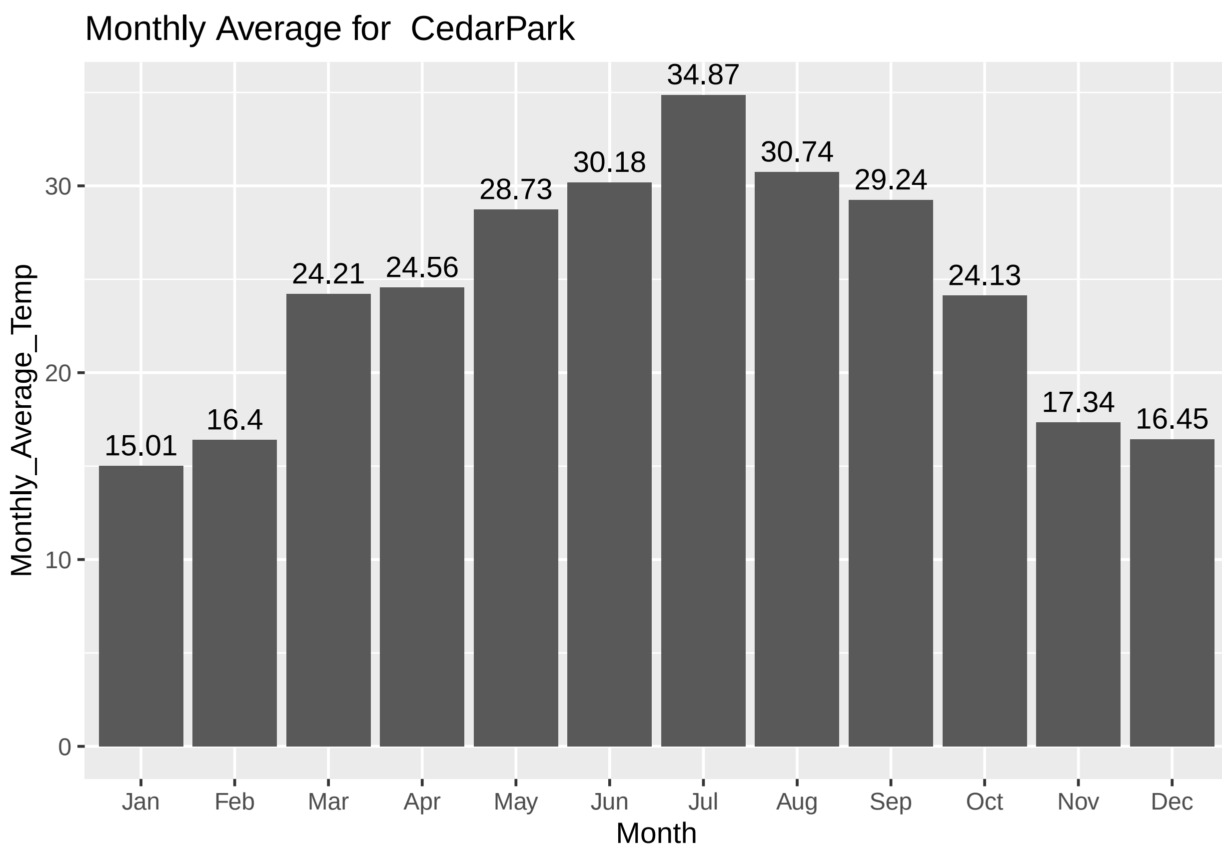
After Normalizing based on each location, it now seems reasonable and the distributions are now normally distributed.

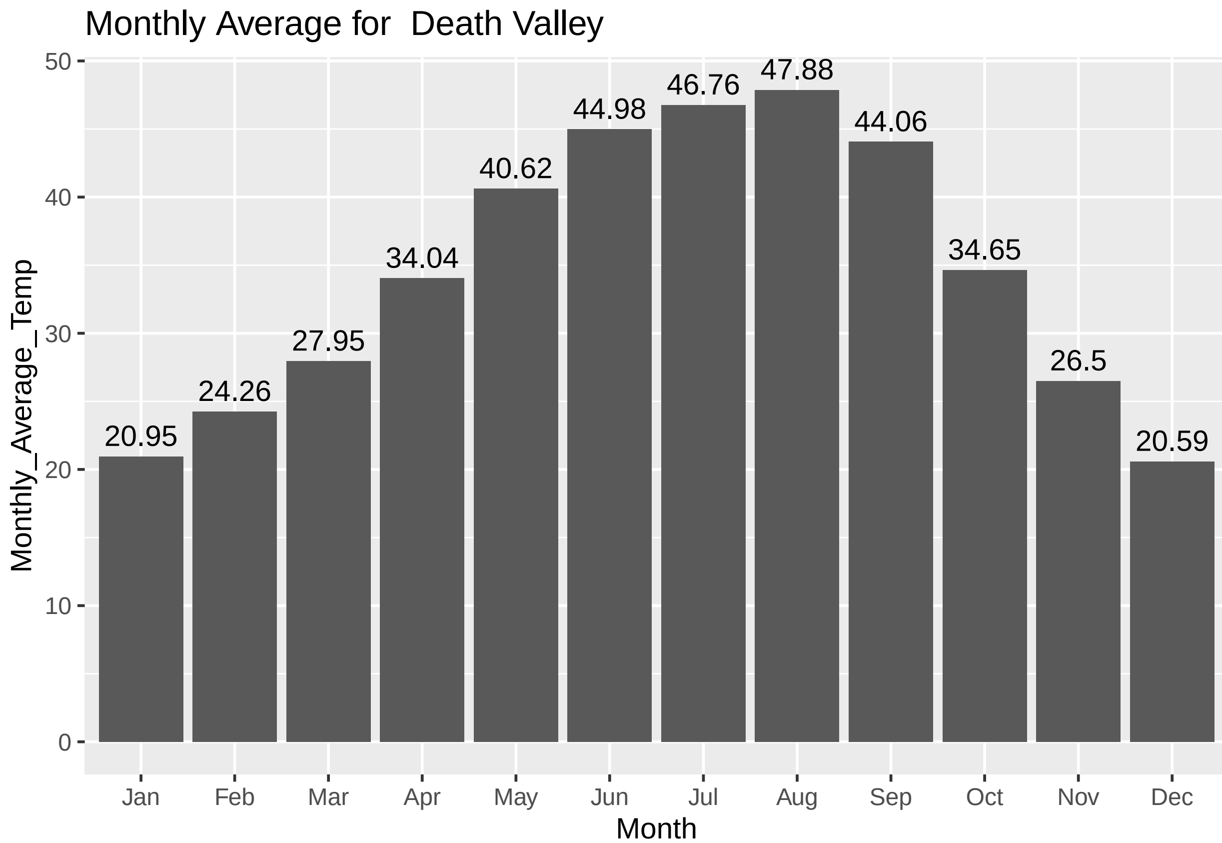
3. Monthly average (max) temperatures for each site

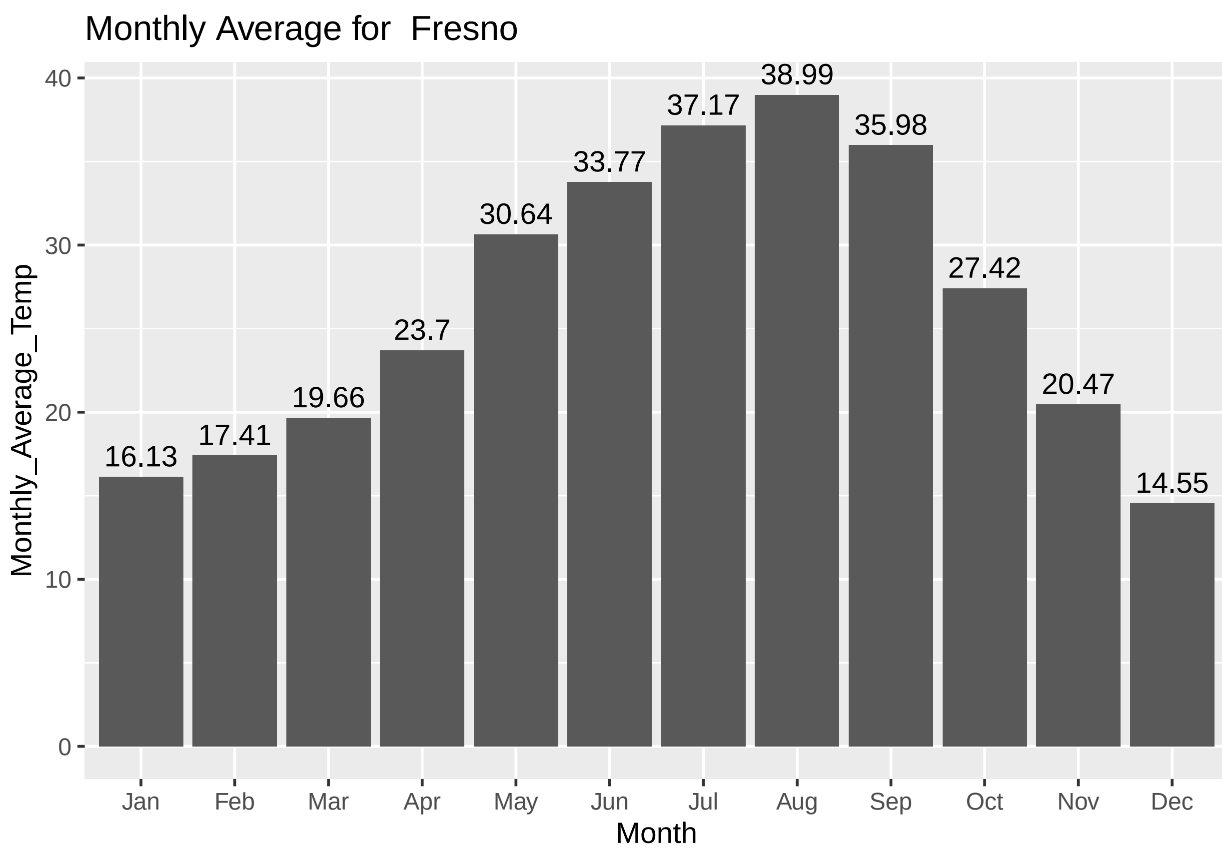
```
monthly_average_temp<-maxtempcalifornia_long %>%
  group_by(Location, Month=month(Date, label = T)) %>%
  summarise(Monthly_Average_Temp=mean(Max_Temp))
```

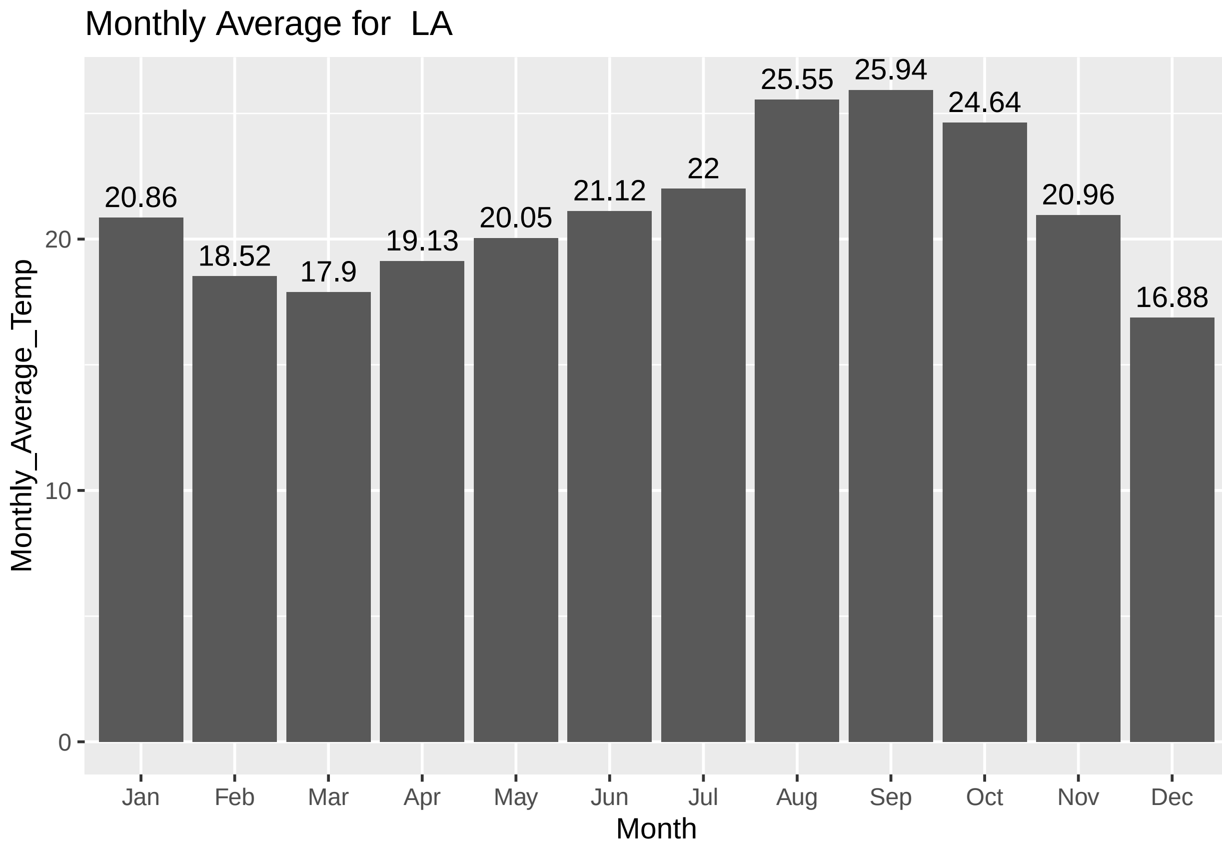
```
for(location in unique(monthly_average_temp$Location)){
  p<-monthly_average_temp %>%
    filter(Location==location) %>%
    dplyr::select(Month, Monthly_Average_Temp) %>%
    ggplot(aes(Month,Monthly_Average_Temp)) +
    geom_col() +
    ggtitle(paste("Monthly Average for ", location))+
    geom_text(aes(label = round(Monthly_Average_Temp, 2)), vjust = -0.5)
  print(p)
}
```

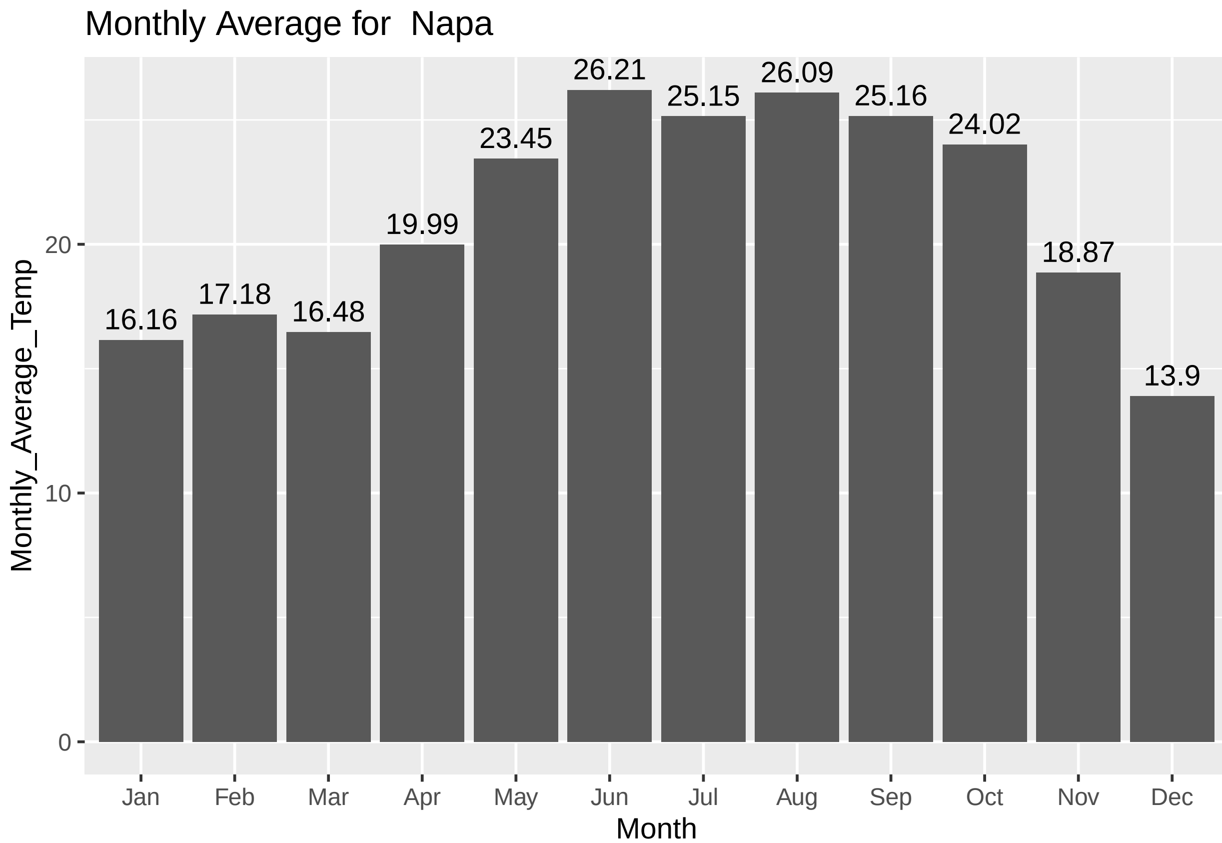

Monthly Average for Barstow
Monthly_Average_Temp
Month
40
30
20
10
0
17.3
17.66
21.05
25.65
31.69
36.42
38
38.91
35.26
27.96
21.23
14.74
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec


Monthly Average for CedarPark
Monthly_Average_Temp
0
10
20
30
15.01
16.4
24.21
24.56
28.73
30.18
34.87
30.74
29.24
24.13
17.34
16.45
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

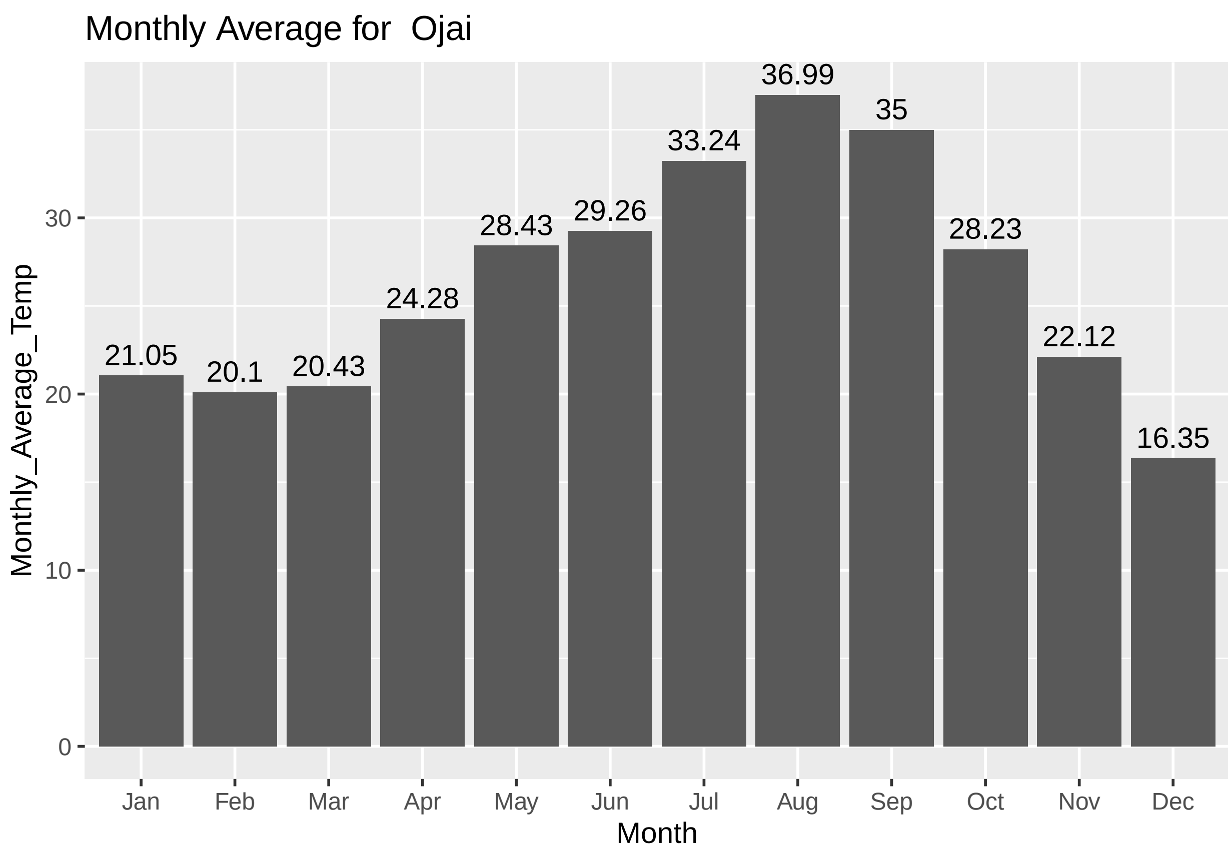

Monthly Average for Death Valley
Monthly_Average_Temp
Month
0
10
20
30
40
50
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
20.95
24.26
27.95
34.04
40.62
44.98
46.76
47.88
44.06
34.65
26.5
20.59

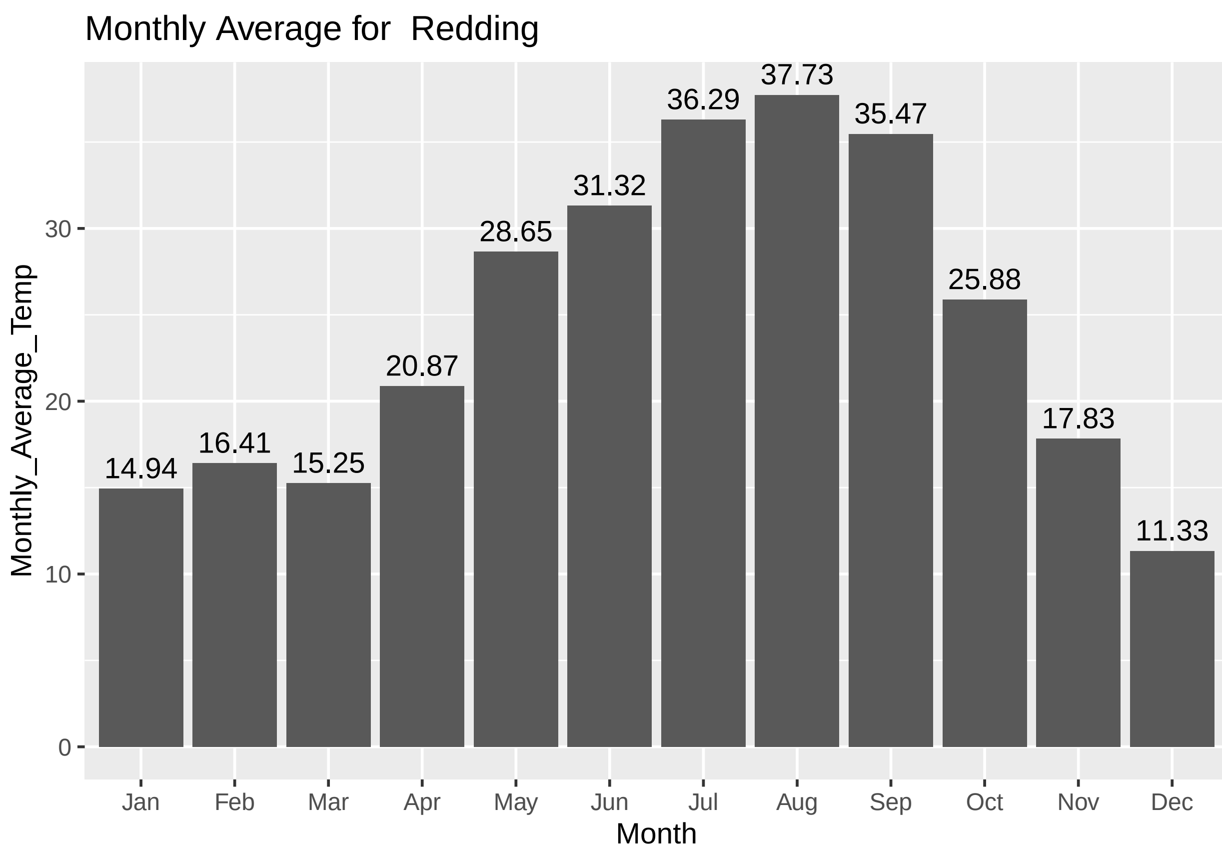

Monthly Average for Fresno
Monthly_Average_Temp
0
10
20
30
40
16.13
17.41
19.66
23.7
30.64
33.77
37.17
38.99
35.98
27.42
20.47
14.55
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month


Monthly Average for LA
Monthly_Average_Temp
0
10
20
20.86
18.52
17.9
19.13
20.05
21.12
22
25.55
25.94
24.64
20.96
16.88
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

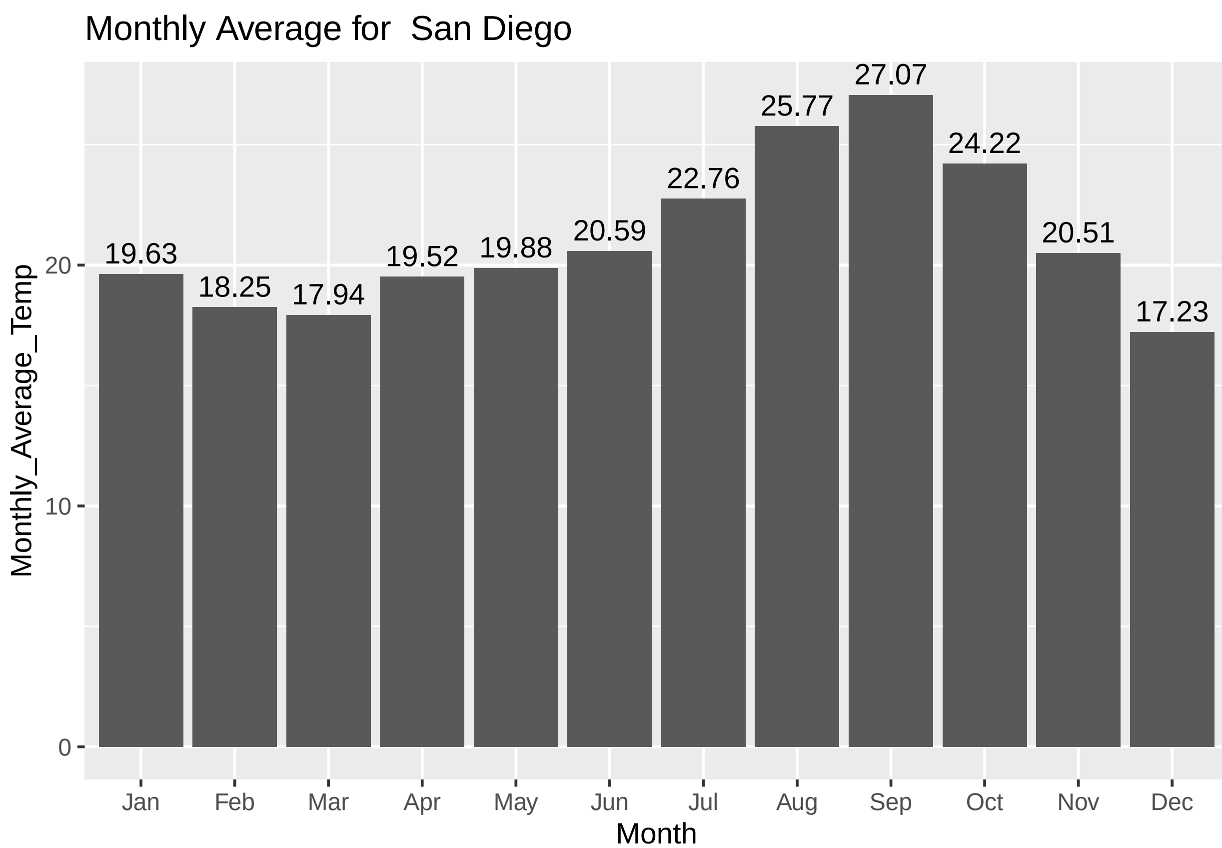


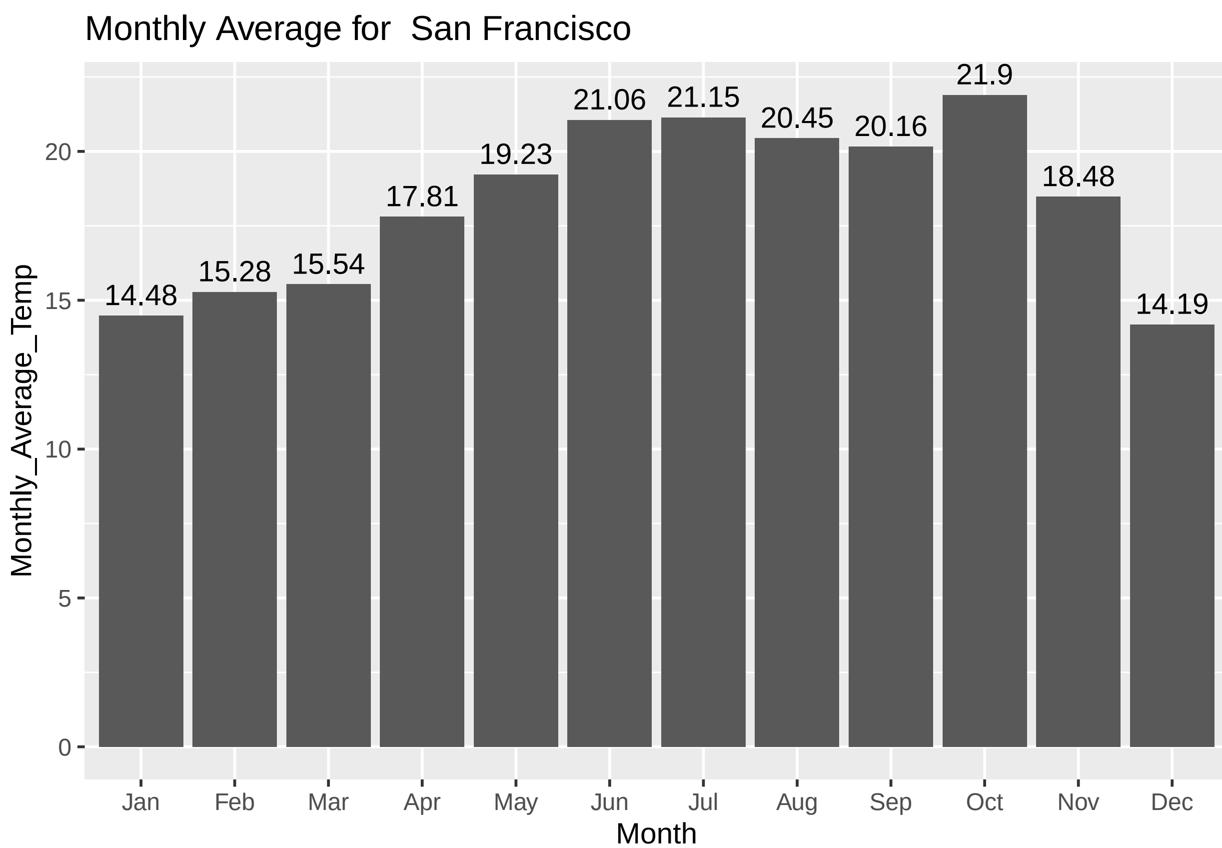
Monthly Average for Napa
Monthly_Average_Temp
Month
0
10
20
16.16
17.18
16.48
19.99
23.45
26.21
25.15
26.09
25.16
24.02
18.87
13.9
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec


Monthly Average for Ojai
Monthly_Average_Temp
Month
0
10
20
30
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
21.05
20.1
20.43
24.28
28.43
29.26
33.24
36.99
35
28.23
22.12
16.35

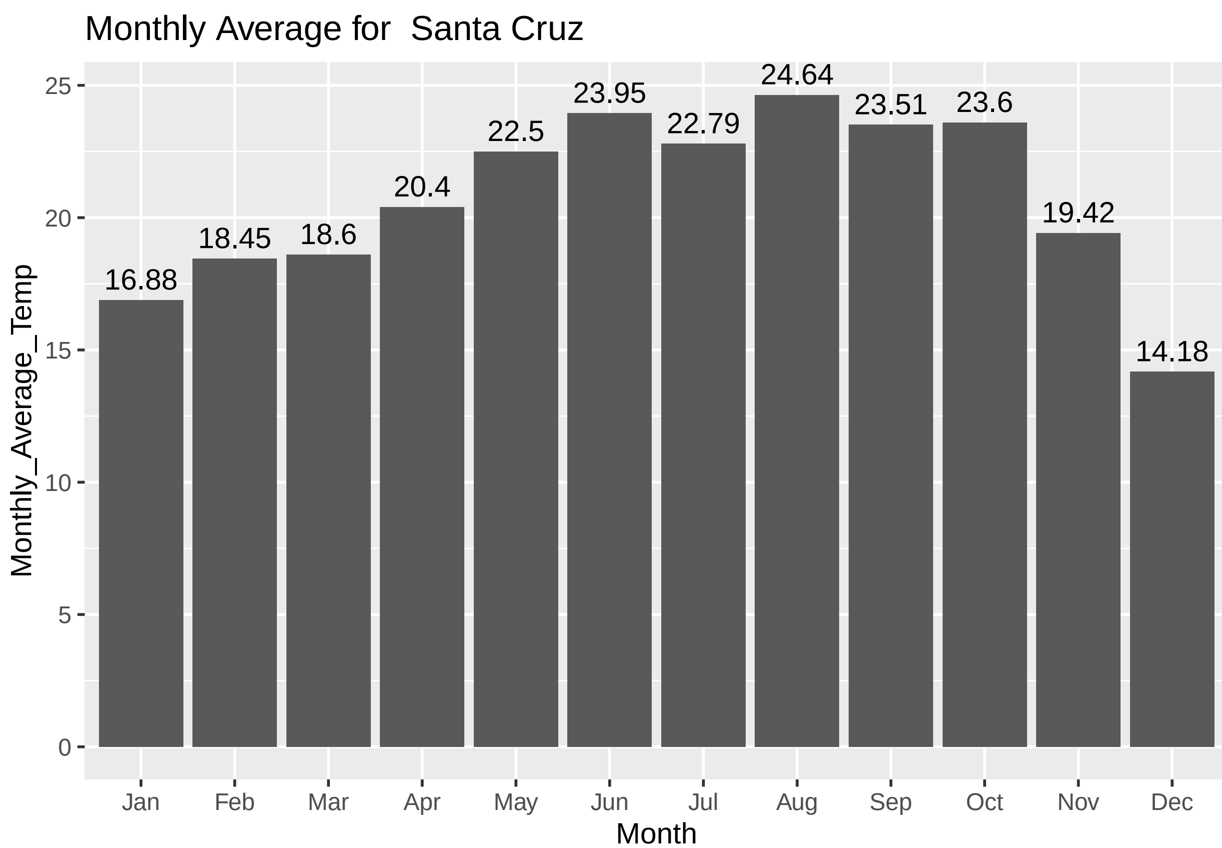


Monthly Average for Redding
Monthly_Average_Temp
Month
0
10
20
30
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
14.94
16.41
15.25
20.87
28.65
31.32
36.29
37.73
35.47
25.88
17.83
11.33


Monthly Average for San Diego
Monthly_Average_Temp
0
10
20
19.63
18.25
17.94
19.52
19.88
20.59
22.76
25.77
27.07
24.22
20.51
17.23
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month



Monthly Average for San Francisco
Monthly_Average_Temp
Month
0
5
10
15
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
14.48
15.28
15.54
17.81
19.23
21.06
21.15
20.45
20.16
21.9
18.48
14.19



From the above plots, it seems that the month of Jun-July records the highest average temperatures in various locations. Low monthly average (max) temperatures are recorded in the first quarter.

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

To determine whether there are differences in (max) temperatures at different locations, and whether there are differences between months, we will Anova test.

```
summary(mod.aov <- aov(
  Monthly_Average_Temp ~ Location + Month,
  data = monthly_average_temp
))
```

```
##                Df Sum Sq Mean Sq F value Pr(>F)
## Location       10   2374   237.4   20.48 <2e-16 ***
## Month          11   4149   377.2   32.54 <2e-16 ***
## Residuals     110   1275    11.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The `p-value` <0.05 indicating that the ANOVA has detected a significant effect of the factors which in this case is different locations and different months. Below are the Multiple comparisons (post-hoc comparisons) of different locations and different Months to help quantify the differences between groups and determine the groups that significantly differ from each other..

```
TukeyHSD(mod.aov)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Monthly_Average_Temp ~ Location + Month, data = monthly_average_temp)
##
## $Location
##                                    diff         lwr         upr     p adj
## CedarPark-Barstow           -2.833191509  -7.4026425   1.7362595 0.6228875
## Death Valley-Barstow         7.282732357   2.7132814  11.8521833 0.0000410
## Fresno-Barstow              -0.830985354  -5.4004363   3.7384656 0.9999483
## LA-Barstow                  -6.024809047 -10.5942600  -1.4553581 0.0015649
## Napa-Barstow                -6.099216104 -10.6686671  -1.5297651 0.0012799
## Ojai-Barstow                -0.866147880  -5.4355989   3.7033031 0.9999242
## Redding-Barstow             -2.823306143  -7.3927571   1.7461448 0.6278133
## San Diego-Barstow           -6.041667903 -10.6111189  -1.4722169 0.0014955
## San Francisco-Barstow       -8.845569151 -13.4150201  -4.2761182 0.0000003
## Santa Cruz-Barstow          -6.410888333 -10.9803393  -1.8414374 0.0005398
## Death Valley-CedarPark      10.115923866   5.5464729  14.6853748 0.0000000
## Fresno-CedarPark             2.002206155  -2.5672448   6.5716571 0.9351900
## LA-CedarPark                -3.191617538  -7.7610685   1.3778334 0.4439480
## Napa-CedarPark              -3.266024595  -7.8354756   1.3034264 0.4085264
## Ojai-CedarPark               1.967043629  -2.6024073   6.5364946 0.9420672
## Redding-CedarPark            0.009885366  -4.5595656   4.5793363 1.0000000
## San Diego-CedarPark         -3.208476394  -7.7779274   1.3609746 0.4358289
## San Francisco-CedarPark     -6.012377642 -10.5818286  -1.4429267 0.0016180
## Santa Cruz-CedarPark        -3.577696824  -8.1471478   0.9917541 0.2752788
## Fresno-Death Valley         -8.113717711 -12.6831687  -3.5442667 0.0000029
## LA-Death Valley            -13.307541404 -17.8769924  -8.7380904 0.0000000
## Napa-Death Valley          -13.381948461 -17.9513994  -8.8124975 0.0000000
## Ojai-Death Valley           -8.148880237 -12.7183312  -3.5794293 0.0000026
## Redding-Death Valley       -10.106038500 -14.6754895  -5.5365875 0.0000000
## San Diego-Death Valley     -13.324400260 -17.8938512  -8.7549493 0.0000000
## San Francisco-Death Valley -16.128301508 -20.6977525 -11.5588505 0.0000000
## Santa Cruz-Death Valley    -13.693620690 -18.2630717  -9.1241697 0.0000000
## LA-Fresno                   -5.193823693  -9.7632747  -0.6243727 0.0127371
## Napa-Fresno                 -5.268230750  -9.8376817  -0.6987798 0.0106845
## Ojai-Fresno                 -0.035162526  -4.6046135   4.5342884 1.0000000
## Redding-Fresno              -1.992320789  -6.5617718   2.5771302 0.9371774
## San Diego-Fresno            -5.210682549  -9.7801335  -0.6412316 0.0122427
## San Francisco-Fresno        -8.014583797 -12.5840348  -3.4451328 0.0000040
## Santa Cruz-Fresno           -5.579902979 -10.1493540  -1.0104520 0.0049826
## Napa-LA                     -0.074407057  -4.6438580   4.4950439 1.0000000
## Ojai-LA                      5.158661167   0.5892102   9.7281121 0.0138277
## Redding-LA                   3.201502904  -1.3679481   7.7709539 0.4391811
## San Diego-LA                -0.016858856  -4.5863098   4.5525921 1.0000000
## San Francisco-LA            -2.820760104  -7.3902111   1.7486909 0.6290803
## Santa Cruz-LA               -0.386079286  -4.9555303   4.1833717 1.0000000
## Ojai-Napa                    5.233068224   0.6636173   9.8025192 0.0116133
## Redding-Napa                 3.275909962  -1.2935410   7.8453609 0.4039060
## San Diego-Napa               0.057548202  -4.5119028   4.6269992 1.0000000
## San Francisco-Napa          -2.746353047  -7.3158040   1.8230979 0.6657299
```

```
## Santa Cruz-Napa             -0.311672228  -4.8811232   4.2577787 1.0000000
## Redding-Ojai                -1.957158262  -6.5266092   2.6122927 0.9439054
## San Diego-Ojai              -5.175520022  -9.7449710  -0.6060690 0.0132946
## San Francisco-Ojai          -7.979421271 -12.5488722  -3.4099703 0.0000045
## Santa Cruz-Ojai             -5.544740452 -10.1141914  -0.9752895 0.0054418
## San Diego-Redding           -3.218361760  -7.7878127   1.3510892 0.4310924
## San Francisco-Redding       -6.022263008 -10.5917140  -1.4528120 0.0015756
## Santa Cruz-Redding          -3.587582190  -8.1570332   0.9818688 0.2715282
## San Francisco-San Diego     -2.803901248  -7.3733522   1.7655497 0.6374508
## Santa Cruz-San Diego        -0.369220430  -4.9386714   4.2002305 1.0000000
## Santa Cruz-San Francisco     2.434680818  -2.1347702   7.0041318 0.8050435
##
## $Month
##                diff         lwr           upr     p adj
## Feb-Jan   0.5940237  -4.2541973  5.442245e+00 0.9999996
## Mar-Jan   1.9645161  -2.8837048  6.812737e+00 0.9696590
## Apr-Jan   5.1412023   0.2929814  9.989423e+00 0.0276007
## May-Jan   9.1340176   4.2857967  1.398224e+01 0.0000004
## Jun-Jan  11.4057478   6.5575269  1.625397e+01 0.0000000
## Jul-Jan  13.3451613   8.4969404  1.819338e+01 0.0000000
## Aug-Jan  14.5777126   9.7294917  1.942593e+01 0.0000000
## Sep-Jan  13.0418084   8.1935875  1.789003e+01 0.0000000
## Oct-Jan   8.4762463   3.6280254  1.332447e+01 0.0000035
## Nov-Jan   2.7566569  -2.0915640  7.604878e+00 0.7573376
## Dec-Jan  -2.0915249  -6.9397459  2.756696e+00 0.9525527
## Mar-Feb   1.3704925  -3.4777285  6.218713e+00 0.9984602
## Apr-Feb   4.5471787  -0.3010422  9.395400e+00 0.0879057
## May-Feb   8.5399939   3.6917730  1.338821e+01 0.0000028
## Jun-Feb  10.8117241   5.9635032  1.565995e+01 0.0000000
## Jul-Feb  12.7511376   7.9029167  1.759936e+01 0.0000000
## Aug-Feb  13.9836889   9.1354680  1.883191e+01 0.0000000
## Sep-Feb  12.4477847   7.5995638  1.729601e+01 0.0000000
## Oct-Feb   7.8822227   3.0340017  1.273044e+01 0.0000216
## Nov-Feb   2.1626332  -2.6855877  7.010854e+00 0.9404015
## Dec-Feb  -2.6855486  -7.5337695  2.162672e+00 0.7865511
## Apr-Mar   3.1766862  -1.6715347  8.024907e+00 0.5616267
## May-Mar   7.1695015   2.3212805  1.201772e+01 0.0001732
## Jun-Mar   9.4412317   4.5930107  1.428945e+01 0.0000002
## Jul-Mar  11.3806452   6.5324242  1.622887e+01 0.0000000
## Aug-Mar  12.6131965   7.7649756  1.746142e+01 0.0000000
## Sep-Mar  11.0772923   6.2290714  1.592551e+01 0.0000000
## Oct-Mar   6.5117302   1.6635093  1.135995e+01 0.0010491
## Nov-Mar   0.7921408  -4.0560802  5.640362e+00 0.9999928
## Dec-Mar  -4.0560411  -8.9042620  7.921799e-01 0.1973235
## May-Apr   3.9928152  -0.8554057  8.841036e+00 0.2165908
## Jun-Apr   6.2645455   1.4163245  1.111277e+01 0.0019934
## Jul-Apr   8.2039589   3.3557380  1.305218e+01 0.0000081
## Aug-Apr   9.4365103   4.5882893  1.428473e+01 0.0000002
## Sep-Apr   7.9006061   3.0523851  1.274883e+01 0.0000204
## Oct-Apr   3.3350440  -1.5131769  8.183265e+00 0.4848499
## Nov-Apr  -2.3845455  -7.2327664  2.463675e+00 0.8891505
## Dec-Apr  -7.2327273 -12.0809482 -2.384506e+00 0.0001447
## Jun-May   2.2717302  -2.5764907  7.119951e+00 0.9178093
## Jul-May   4.2111437  -0.6370772  9.059365e+00 0.1553090
```

```
## Aug-May   5.4436950   0.5954741  1.029192e+01 0.0143200
## Sep-May   3.9077908  -0.9404301  8.756012e+00 0.2444813
## Oct-May  -0.6577713  -5.5059922  4.190450e+00 0.9999990
## Nov-May  -6.3773607 -11.2255816 -1.529140e+00 0.0014909
## Dec-May -11.2255425 -16.0737634 -6.377322e+00 0.0000000
## Jul-Jun   1.9394135  -2.9088074  6.787634e+00 0.9724045
## Aug-Jun   3.1719648  -1.6762561  8.020186e+00 0.5639305
## Sep-Jun   1.6360606  -3.2121603  6.484282e+00 0.9927910
## Oct-Jun  -2.9295015  -7.7777224  1.918719e+00 0.6803679
## Nov-Jun  -8.6490909 -13.4973118 -3.800870e+00 0.0000020
## Dec-Jun -13.4972727 -18.3454937 -8.649052e+00 0.0000000
## Aug-Jul   1.2325513  -3.6156696  6.080772e+00 0.9994224
## Sep-Jul  -0.3033529  -5.1515738  4.544868e+00 1.0000000
## Oct-Jul  -4.8689150  -9.7171359 -2.069403e-02 0.0480128
## Nov-Jul -10.5885044 -15.4367253 -5.740283e+00 0.0000000
## Dec-Jul -15.4366862 -20.2849071 -1.058847e+01 0.0000000
## Sep-Aug  -1.5359042  -6.3841251  3.312317e+00 0.9957800
## Oct-Aug  -6.1014663 -10.9496872 -1.253245e+00 0.0030099
## Nov-Aug -11.8210557 -16.6692766 -6.972835e+00 0.0000000
## Dec-Aug -16.6692375 -21.5174585 -1.182102e+01 0.0000000
## Oct-Sep  -4.5655621  -9.4137830  2.826589e-01 0.0850493
## Nov-Sep -10.2851515 -15.1333724 -5.436931e+00 0.0000000
## Dec-Sep -15.1333333 -19.9815543 -1.028511e+01 0.0000000
## Nov-Oct  -5.7195894 -10.5678104 -8.713685e-01 0.0076019
## Dec-Oct -10.5677713 -15.4159922 -5.719550e+00 0.0000000
## Dec-Nov  -4.8481818  -9.6964027  3.910913e-05 0.0500038
```

Prediction

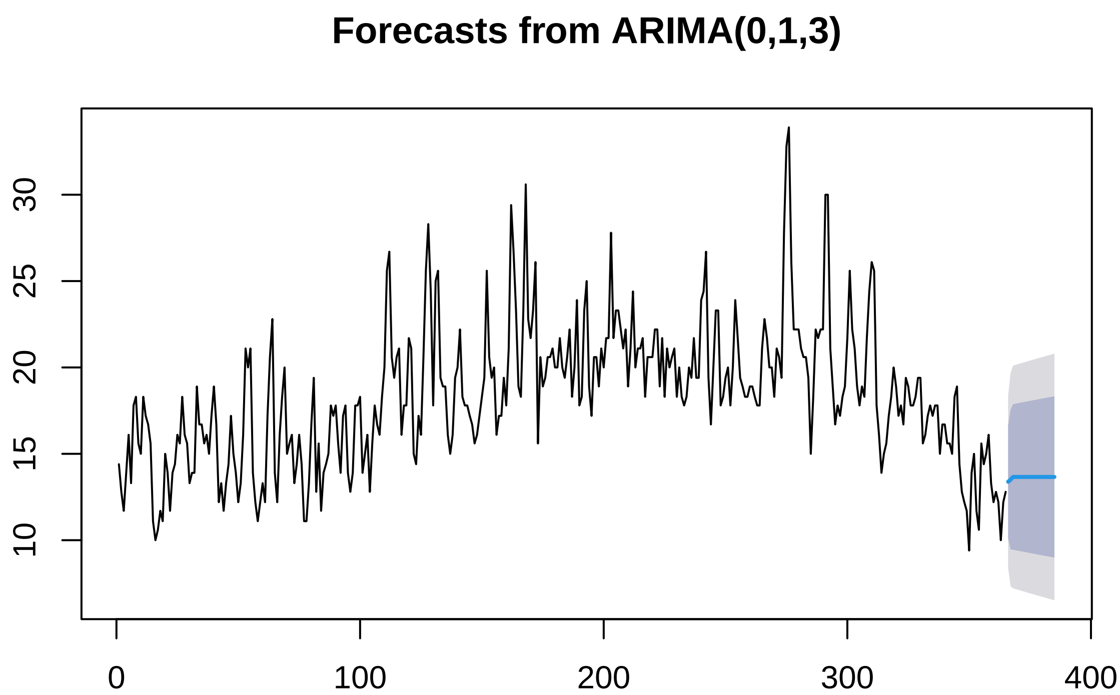
5. Developing a time series model of San Francisco and applying it to data from other locations to predict maximum temperatures for all locations, for the 1st to 8th August 2012

```
# Select only the San Francisco data
sanfrancisco <- maxtempcalifornia_long %>%
  filter(Location=="San Francisco") %>%
  dplyr::select(Date, Max_Temp)

# Create a time Series
sanfrancisco_xts = xts(sanfrancisco[, -1], order.by = sanfrancisco$Date)
head(sanfrancisco_xts)
```

```
##            [,1]
## 2012-01-01 14.4
## 2012-01-02 12.8
## 2012-01-03 11.7
## 2012-01-04 13.9
## 2012-01-05 16.1
## 2012-01-06 13.3
```

```
# create and Find the best ARIMA model
fit <- auto.arima(
```

```r
  sanfrancisco_xts
)
plot(forecast(fit, h=20))
```




```r
summary(fit)
```

```
## Series: sanfrancisco_xts
## ARIMA(0,1,3)
##
## Coefficients:
##           ma1      ma2      ma3
##       -0.2609  -0.3753  -0.2141
## s.e.   0.0520   0.0473   0.0512
##
## sigma^2 estimated as 6.438:  log likelihood=-854.48
## AIC=1716.96   AICc=1717.08   BIC=1732.55
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.01336338 2.523348 1.916014 -1.78497 10.69129 0.9164642
##                      ACF1
## Training set -0.004181537
```

```r
# Select the predicted for the 1st-8th August 2012
pred_period = yday(
  seq(ymd('2012-08-01'),ymd('2012-08-08'), by='1 day')
)

# Get predictions of the entire year
preds <- forecast(fit, h=365)$fitted

# Predicted Maximum temperature for all locations for 1st-8th August
req_preds <- preds[pred_period]
print(req_preds)
```

```
## [1] 19.61836 21.55233 21.47097 21.49281 19.14388 21.36925 20.93775 20.66508
```

Let's compare the predicted maximum temperatures with the observed measurements in all locations.

```r
cal_01_08<-maxtempcalifornia_long%>%
  filter((Date>="2012-08-01")&(Date<="2012-08-08"))
cal_01_08$pred <- req_preds
cal_01_08
```

```
##         Location Max_Temp       Date Normalized_Max_Temp     pred
## 1  San Francisco     21.1 2012-08-01          0.80309849 19.61836
## 2  San Francisco     21.1 2012-08-02          0.80309849 21.55233
## 3  San Francisco     21.7 2012-08-03          0.96559561 21.47097
## 4  San Francisco     18.3 2012-08-04          0.03434412 21.49281
## 5  San Francisco     20.6 2012-08-05          0.64250099 19.14388
## 6  San Francisco     20.6 2012-08-06          0.64250099 21.36925
## 7  San Francisco     20.6 2012-08-07          0.64250099 20.93775
## 8  San Francisco     22.2 2012-08-08          1.12574381 20.66508
## 9           Napa     26.1 2012-08-01          0.90416214 19.61836
## 10          Napa     26.7 2012-08-02          1.01223522 21.55233
## 11          Napa     22.2 2012-08-03          0.20168708 21.47097
## 12          Napa     23.3 2012-08-04          0.39982107 21.49281
## 13          Napa     25.6 2012-08-05          0.81410123 19.14388
## 14          Napa     28.3 2012-08-06          1.30043012 21.36925
## 15          Napa     28.9 2012-08-07          1.40850320 20.93775
## 16          Napa     30.6 2012-08-08          1.71471028 20.66508
## 17     San Diego     23.9 2012-08-01          0.75348060 19.61836
## 18     San Diego     23.3 2012-08-02          0.61717602 21.55233
## 19     San Diego     21.7 2012-08-03          0.23582178 21.47097
## 20     San Diego     22.8 2012-08-04          0.50088529 21.49281
## 21     San Diego     23.3 2012-08-05          0.61717602 19.14388
## 22     San Diego     25.0 2012-08-06          0.99472723 21.36925
## 23     San Diego     26.1 2012-08-07          1.22560051 20.93775
## 24     San Diego     26.7 2012-08-08          1.34746916 20.66508
## 25        Fresno     41.1 2012-08-01          1.89922996 19.61836
## 26        Fresno     40.6 2012-08-02          1.72276237 21.55233
## 27        Fresno     38.9 2012-08-03          1.39956428 21.47097
## 28        Fresno     37.2 2012-08-04          1.03937549 21.49281
## 29        Fresno     37.2 2012-08-05          1.03937549 19.14388
## 30        Fresno     38.3 2012-08-06          1.25093305 21.36925
```

```
## 31        Fresno     38.9 2012-08-07          1.39956428 20.93775
## 32        Fresno     39.4 2012-08-08          1.54111964 20.66508
## 33    Santa Cruz     23.3 2012-08-01          0.51261712 19.61836
## 34    Santa Cruz     22.2 2012-08-02          0.16557280 21.55233
## 35    Santa Cruz     20.6 2012-08-03         -0.04465342 21.47097
## 36    Santa Cruz     17.2 2012-08-04         -0.68530626 21.49281
## 37    Santa Cruz     23.9 2012-08-05          0.66803760 19.14388
## 38    Santa Cruz     26.7 2012-08-06          1.35523573 21.36925
## 39    Santa Cruz     28.3 2012-08-07          1.72276237 20.93775
## 40    Santa Cruz     29.4 2012-08-08          2.01565928 20.66508
## 41  Death Valley     46.1 2012-08-01          0.94392988 19.61836
## 42  Death Valley     46.1 2012-08-02          0.94392988 21.55233
## 43  Death Valley     47.8 2012-08-03          1.24345930 21.47097
## 44  Death Valley     48.9 2012-08-04          1.60004302 21.49281
## 45  Death Valley     47.8 2012-08-05          1.24345930 19.14388
## 46  Death Valley     48.9 2012-08-06          1.60004302 21.36925
## 47  Death Valley     49.4 2012-08-07          1.84010090 20.93775
## 48  Death Valley     50.6 2012-08-08          2.13358567 20.66508
## 49          Ojai     35.0 2012-08-01          1.06324418 19.61836
## 50          Ojai     33.9 2012-08-02          0.87135263 21.55233
## 51          Ojai     29.4 2012-08-03          0.36138960 21.47097
## 52          Ojai     31.7 2012-08-04          0.58447005 21.49281
## 53          Ojai     36.7 2012-08-05          1.23605437 19.14388
## 54          Ojai     43.3 2012-08-06          2.77740699 21.36925
## 55          Ojai     40.6 2012-08-07          1.87878415 20.93775
## 56          Ojai     41.1 2012-08-08          2.04256125 20.66508
## 57       Barstow     33.3 2012-08-01          0.33948211 19.61836
## 58       Barstow     34.4 2012-08-02          0.45090947 21.55233
## 59       Barstow     38.9 2012-08-03          1.36388825 21.47097
## 60       Barstow     38.9 2012-08-04          1.36388825 21.49281
## 61       Barstow     36.7 2012-08-05          0.83187509 19.14388
## 62       Barstow     40.0 2012-08-06          1.63824851 21.36925
## 63       Barstow     40.6 2012-08-07          1.80398999 20.93775
## 64       Barstow     41.7 2012-08-08          2.10127950 20.66508
## 65            LA     22.2 2012-08-01          0.34413988 19.61836
## 66            LA     22.2 2012-08-02          0.34413988 21.55233
## 67            LA     22.8 2012-08-03          0.47968390 21.47097
## 68            LA     23.3 2012-08-04          0.58994499 21.49281
## 69            LA     21.7 2012-08-05          0.22836389 19.14388
## 70            LA     23.9 2012-08-06          0.71918219 21.36925
## 71            LA     28.3 2012-08-07          1.57826715 20.93775
## 72            LA     26.7 2012-08-08          1.28237255 20.66508
## 73     CedarPark     27.8 2012-08-01          0.25334710 19.61836
## 74     CedarPark     33.3 2012-08-02          1.30532906 21.55233
## 75     CedarPark     37.8 2012-08-03          2.16828530 21.47097
## 76     CedarPark     35.0 2012-08-04          1.62524865 21.49281
## 77     CedarPark     33.9 2012-08-05          1.44682763 19.14388
## 78     CedarPark     33.3 2012-08-06          1.30532906 21.36925
## 79     CedarPark     32.8 2012-08-07          1.20000606 20.93775
## 80     CedarPark     28.9 2012-08-08          0.46234265 20.66508
## 81       Redding     38.3 2012-08-01          1.43711637 19.61836
## 82       Redding     38.9 2012-08-02          1.57581479 21.55233
## 83       Redding     40.6 2012-08-03          1.92050200 21.47097
## 84       Redding     37.2 2012-08-04          1.20000606 21.49281
```

```
## 85       Redding     36.7 2012-08-05          1.08773451 19.14388
## 86       Redding     35.6 2012-08-06          0.84652439 21.36925
## 87       Redding     37.8 2012-08-07          1.30532906 20.93775
## 88       Redding     39.4 2012-08-08          1.69323388 20.66508
```

To determine how the model performed, we will check the Root Mean Squared Error, which is an estimator of the Root average of squares of the errors.

```
rmse(cal_01_08$Max_Temp, cal_01_08$pred)
```

```
## [1] 14.18539
```

The summary below is how the model fits and predicts the data.

```
summary(fit)
```

```
## Series: sanfrancisco_xts
## ARIMA(0,1,3)
##
## Coefficients:
##           ma1      ma2      ma3
##       -0.2609  -0.3753  -0.2141
## s.e.   0.0520   0.0473   0.0512
##
## sigma^2 estimated as 6.438:  log likelihood=-854.48
## AIC=1716.96   AICc=1717.08   BIC=1732.55
##
## Training set error measures:
##                       ME     RMSE      MAE      MPE     MAPE      MASE
## Training set -0.01336338 2.523348 1.916014 -1.78497 10.69129 0.9164642
##                      ACF1
## Training set -0.004181537
```

The `auto.arima()` function in R uses a combination of unit root tests, minimization of the AIC and MLE to obtain an ARIMA model. `auto.arima` determines the best ARIMA model to be used as the time series model. It chose ARIMA(0,1,3) that means the ARIMA model has 0 autoregressive term, 1 seasonal autoregressive term and 1 seasonal difference term. The training set errors measures are as indicated in the summary. We can see that the Root Mean Squared Error is 2.52

6. Developing a spatial model to predict maximum temperatures for San Fransisco and Death Valley for 1st Jan 2012 using only data from Napa, San Diego, Fresno, Santa Cruz, Ojai, Barstow, LA and CedarPark

```
# Merge Metadata with Maximum Temperature
merged <- merge(
  maxtempcalifornia_long,
  metadataCA,
  by.x="Location",
  by.y="ï..Location"
)
```

```r
# specify columns containing coordinates of locations
coordinates(merged) <- c("Long", "Lat")

# set coordinate reference system
crs.geo1 <- CRS("+proj=longlat")
proj4string(merged) <- crs.geo1

# Select from locations that are not Redding, San Francisco and Death Valley
train_data <- merged[
  !merged$Location %in% c("Redding", "San Francisco", "Death Valley"),
]
# Fit Spatial Lag Model
spl.model <- lagsarlm(
  Max_Temp~Elev,
  data=train_data,
  nb2listw(
    knn2nb(
      knearneigh(coordinates(train_data), longlat = TRUE)
    )
  )
)
```

```r
test_data <- merged[
  (merged$Location %in% c("San Francisco", "Death Valley")) & (merged$Date=="2012-01-01"),
]
test_data_lw <- nb2listw(
  knn2nb(
    knearneigh(coordinates(test_data), longlat = T)
  )
)
row.names(test_data) = attributes(test_data_lw)$region.id
spl.preds <- predict(
  spl.model,
  test_data,
  test_data_lw
)
print(spl.preds)
```

```
##        fit    trend   signal
## 1 22.70389 21.23370 1.470197
## 2 22.95248 21.49821 1.454274
```

How the model performs and measures of uncertainty for the predictions

```r
summary(spl.model)
```

```
##
## Call:
## lagsarlm(formula = Max_Temp ~ Elev, data = train_data, listw = nb2listw(knn2nb(knearneigh(coordinates
##     longlat = TRUE))))
##
## Residuals:
```

```
##       Min        1Q    Median        3Q       Max
## -22.95670  -5.14661  -0.91371   4.25676  21.09256
##
## Type: lag
## Coefficients: (asymptotic standard errors)
##               Estimate Std. Error z value  Pr(>|z|)
## (Intercept) 2.1383e+01 4.4885e-01 47.6389 < 2.2e-16
## Elev        2.5239e-03 2.8672e-04  8.8029 < 2.2e-16
##
## Rho: 0.064054, LR test value: 3.1192, p-value: 0.077377
## Asymptotic standard error: 0.018342
##     z-value: 3.4922, p-value: 0.00047907
## Wald statistic: 12.195, p-value: 0.00047907
##
## Log likelihood: -9920.651 for lag model
## ML residual variance (sigma squared): 52.304, (sigma: 7.2321)
## Number of observations: 2920
## Number of parameters estimated: 4
## AIC: 19849, (AIC for lm: 19850)
## LM test for residual autocorrelation
## test value: 0.61402, p-value: 0.43328
```

Report

7. Analysis of maximum temperatures over both space and time

Introduction

This report outlines the analysis of maximum temperatures over both space and time for California. The aim of this analysis is to summarize the spatial and temporal variations in maximum temperatures in California in 2012 using various spatial methods and time series analysis.

Initial Data Analysis

The 2 data files that have been used in coming up with this analysis are from National Oceanic and Atmospheric Administration (NOAA)'s National Centers for Environmental Information (NCEI). The data sets are:

- `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.

```
##     ï..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
```

- `MaxTempCalifornia.csv`: has maximum daily temperatures in degrees Celsius for those sites from Jan 1, 2012 to December 30, 2012.

```
##          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
```

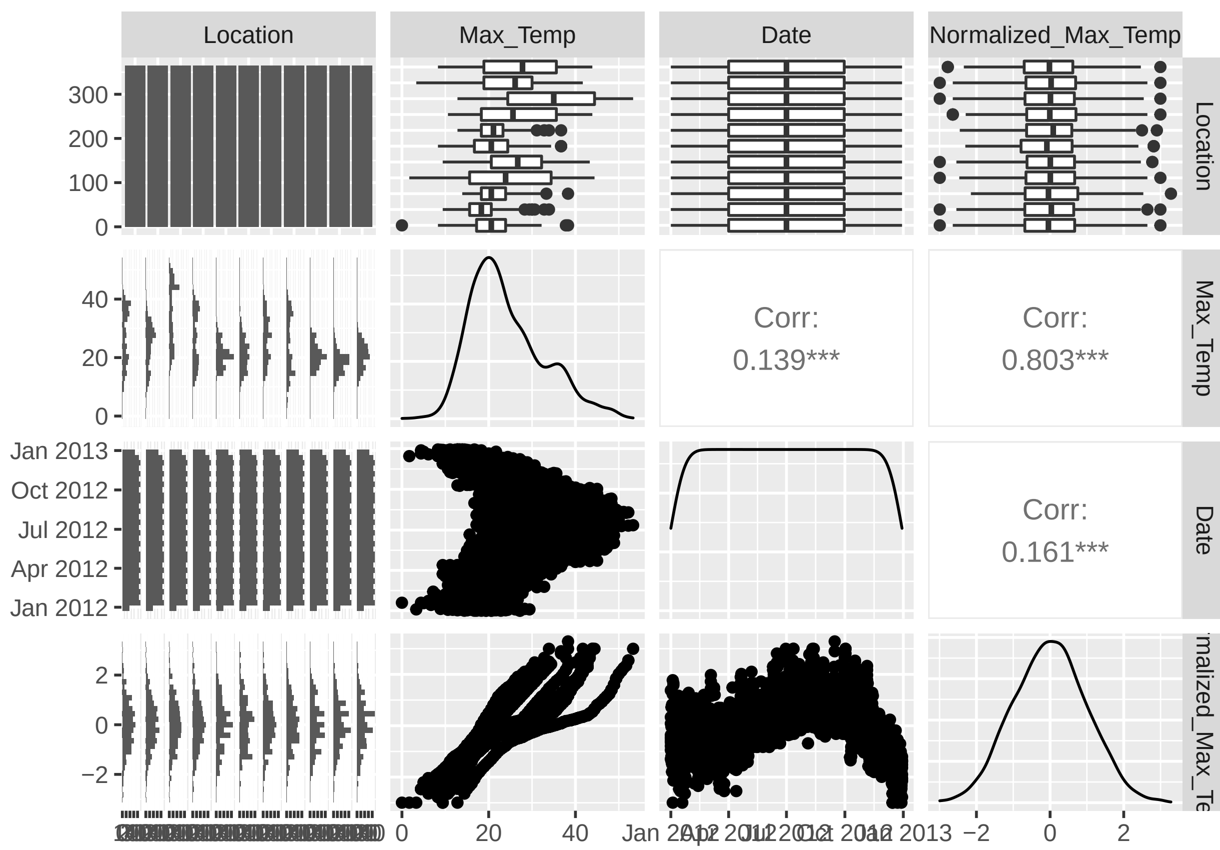
The following are the summaries for the 2 datasets

```
##  ï..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

##    Location           Max_Temp          Date            Normalized_Max_Temp
##  Length:4015        Min.   : 0.00   Min.   :2012-01-01   Min.   :-2.995525
##  Class :character   1st Qu.:17.80   1st Qu.:2012-04-01   1st Qu.:-0.685306
##  Mode  :character   Median :22.20   Median :2012-07-01   Median : 0.003553
##                     Mean   :24.15   Mean   :2012-07-01   Mean   : 0.000033
##                     3rd Qu.:28.90   3rd Qu.:2012-09-30   3rd Qu.: 0.663752
##                     Max.   :53.30   Max.   :2012-12-30   Max.   : 3.282423
```

From this summary, we can see that the maximum value in `Elev` (from metadataCA), and `Max_Temp` (from maxtempcalifornia dataset) columns are very extreme, this shows that there are outliers in the datasets. The scatter matrix below reveals more of the datasets.

From the above diagram, checking `Elev` distribution is right skewed, this is as a result of outliers.Other distributions seems okay.



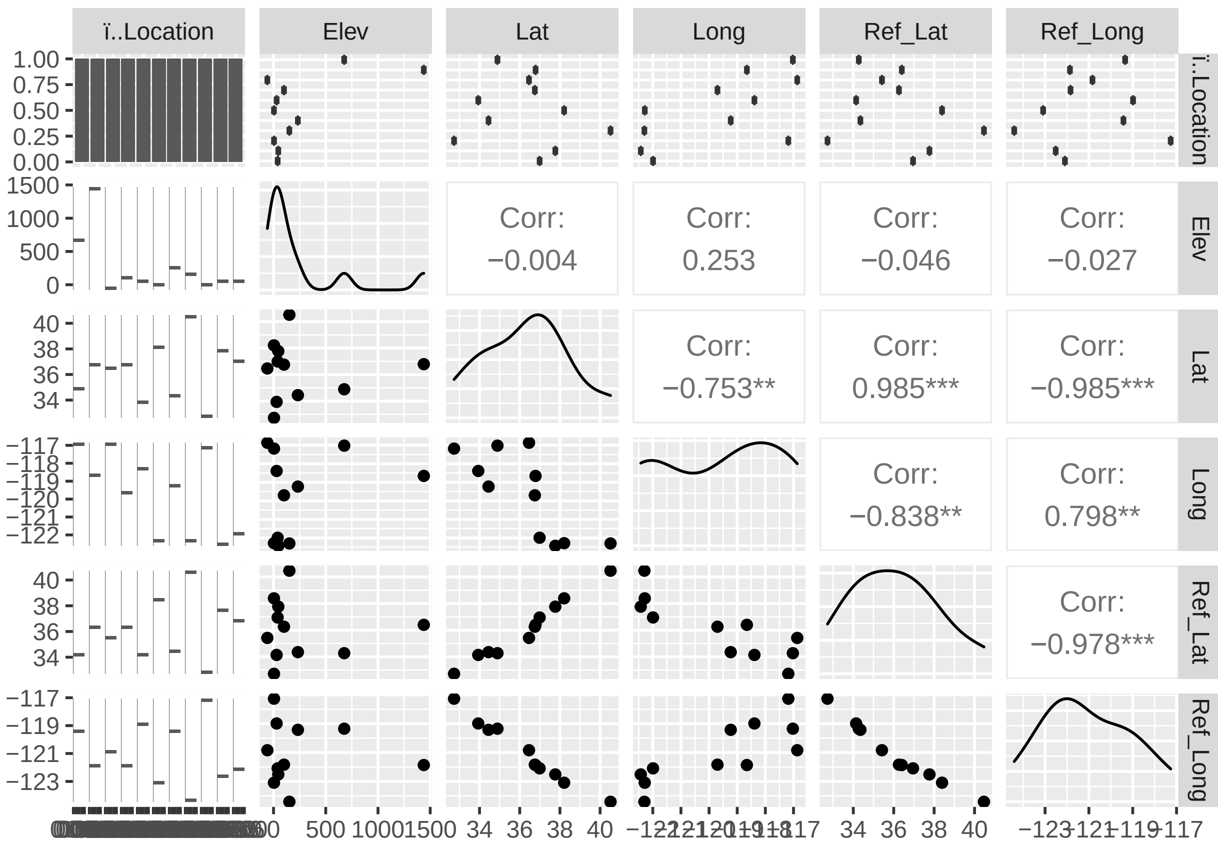


Figure 2: MetadataCA Scatter Matrix

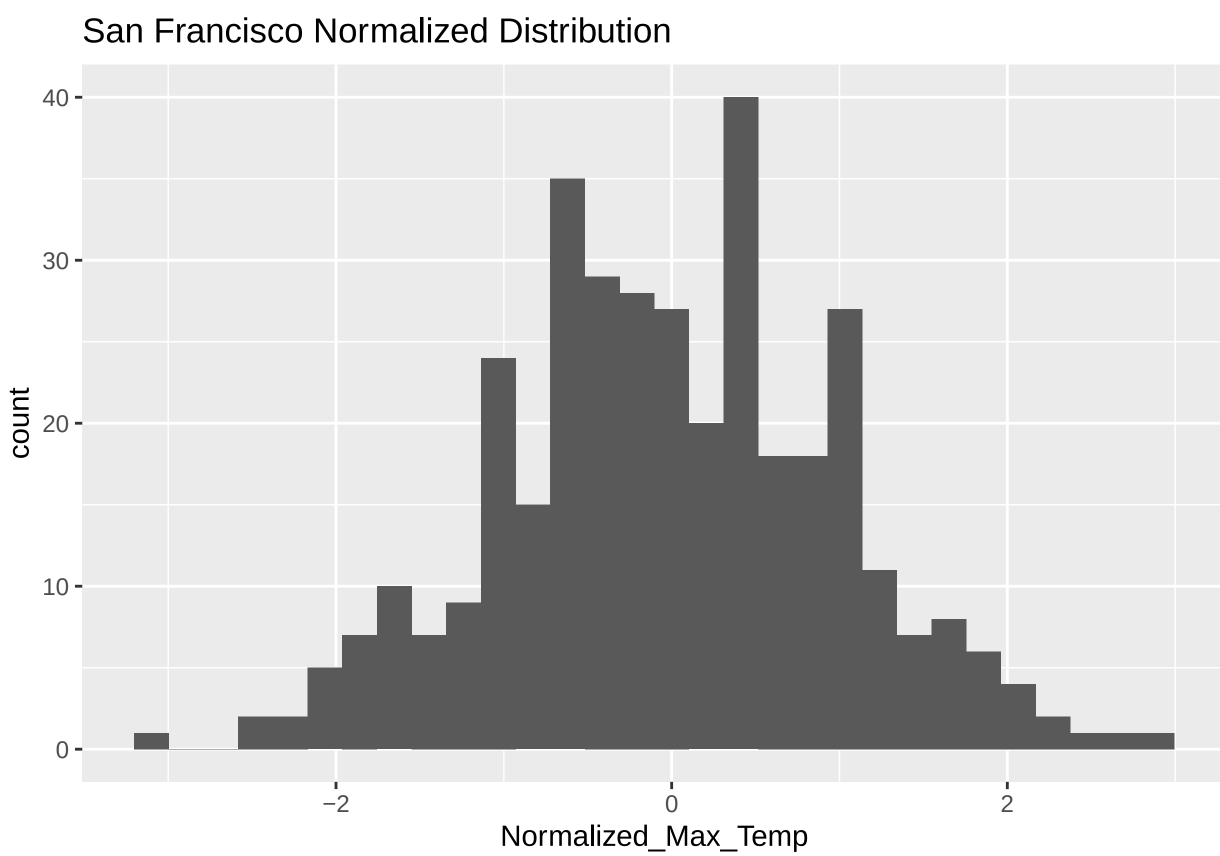
The `Max_Temp` Column was almost a normal distribution were it not for the outliers present forcing it to be a little bit right skewed.

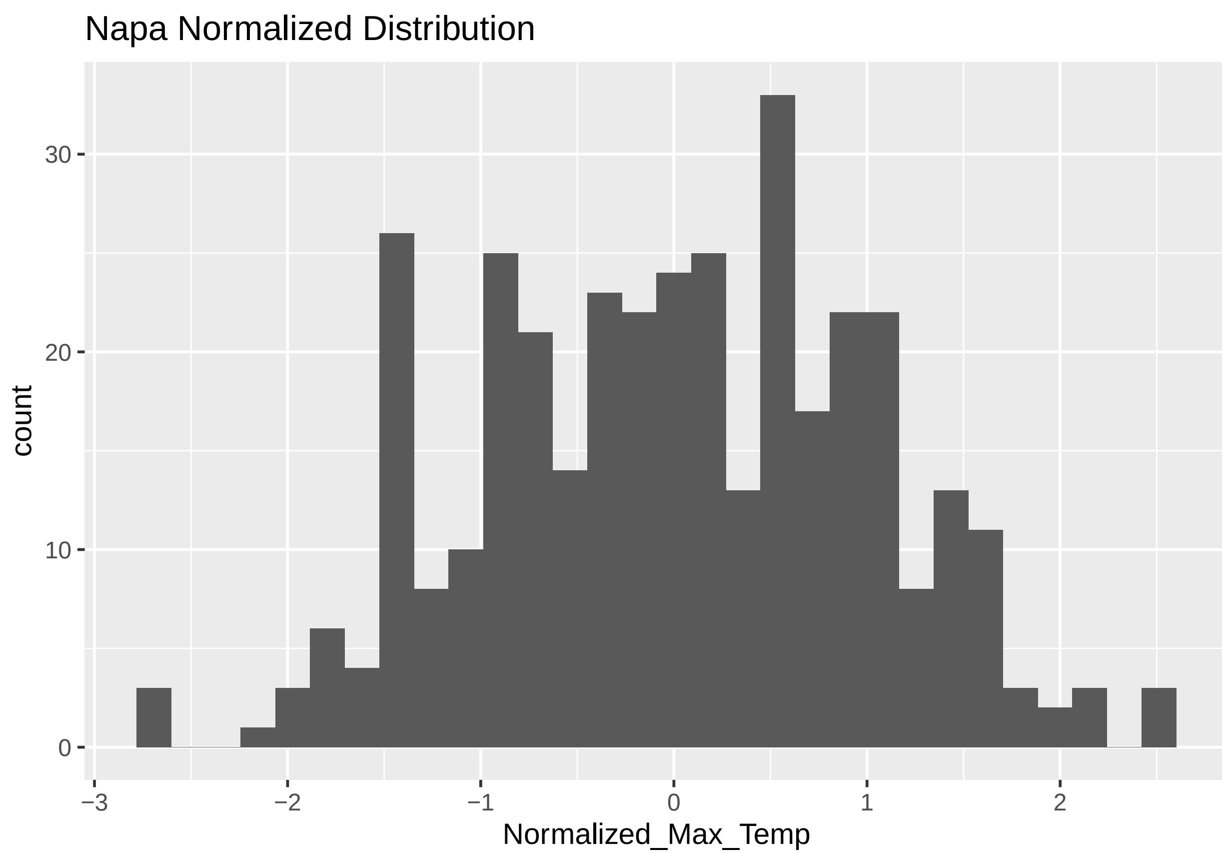
Methods

For the main analysis, the following methods were done:

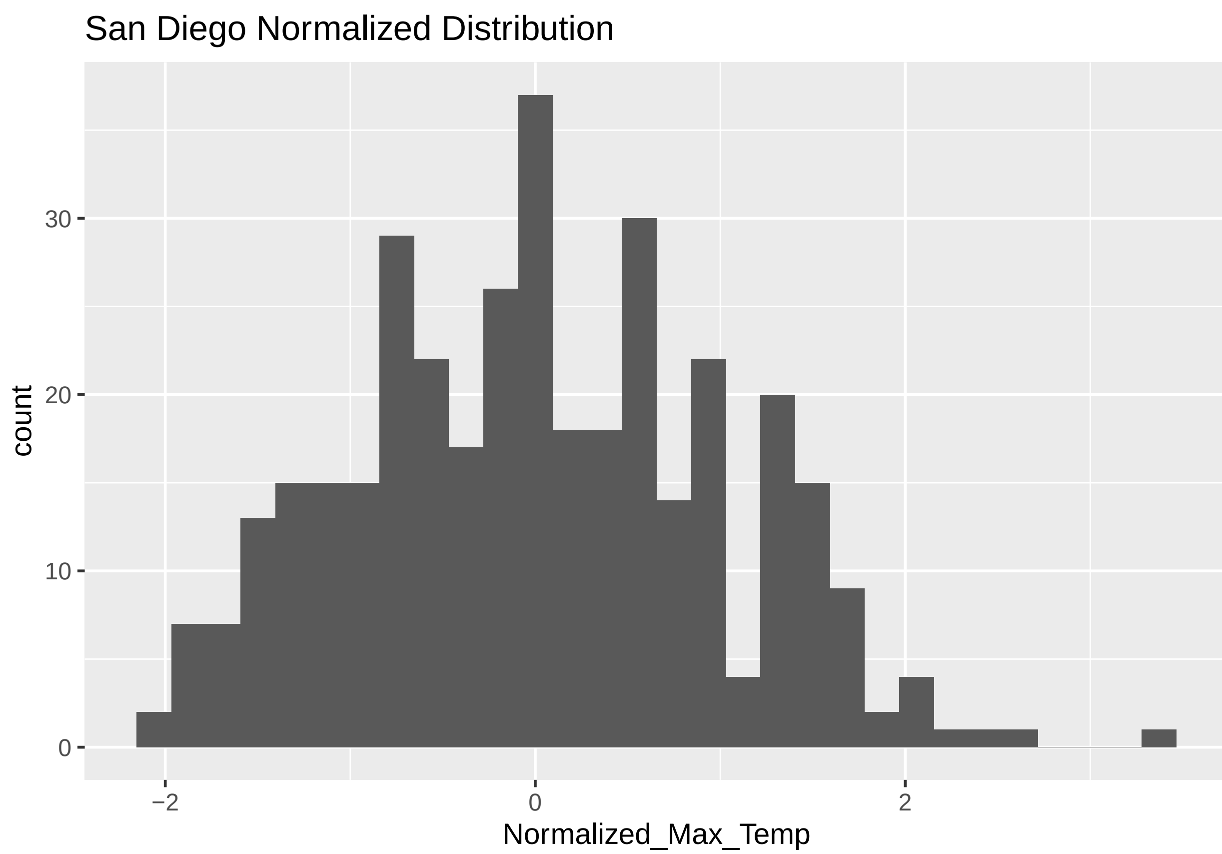
- **Normalization**: Normalization was done to the `Max_Temp` column in order to minimize redundancy and impute the outliers/anomalies with considerable values
- **Statistical Analysis**: Statistical Analysis to determine whether there are differences in (max) temperatures at different locations, and whether there are (statistically significant) differences between months.
- **Prediction**:
  - Develop a time series model to predict maximum temperatures in various locations
  - Develop a spatial model to predict maximum temperatures from San Francisco and Death Valley.

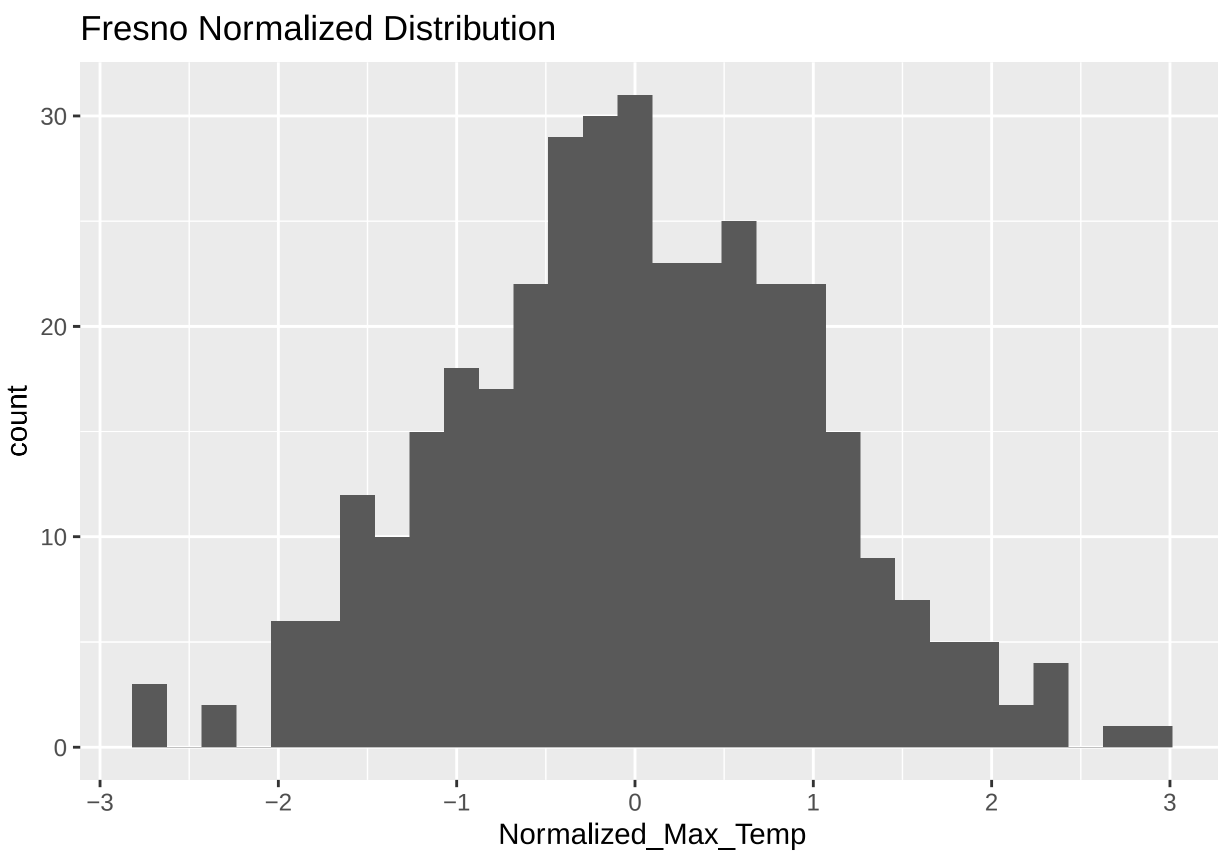
For each location, the data doesn't look Normally distributed, therefore transformation for each site was done. The **Ojai** location is almost normally distributed. For this transformation `bestNormalize` package was used to transform the data at each site to be normally distributed. The data for each location were normalized, in order to form a normal distribution at each location, after normalizing, this is how it looked like.

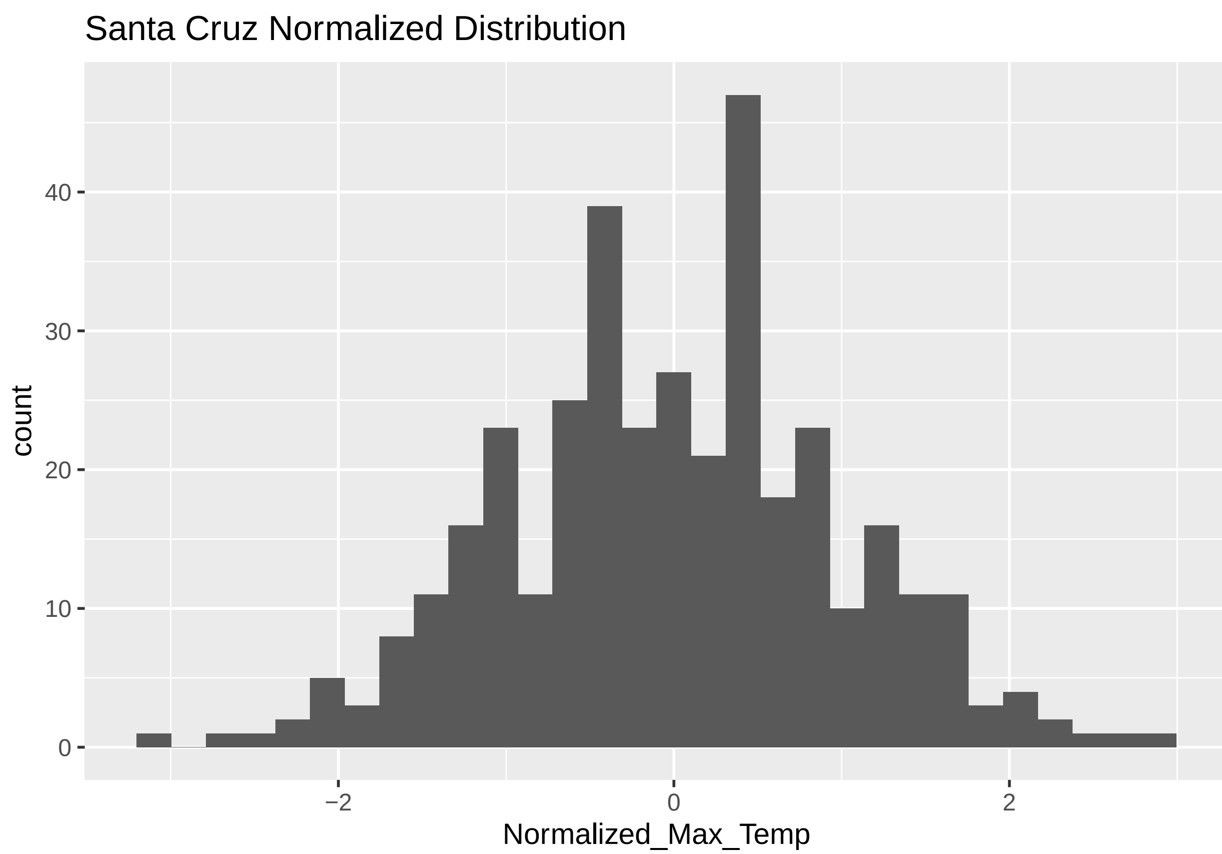




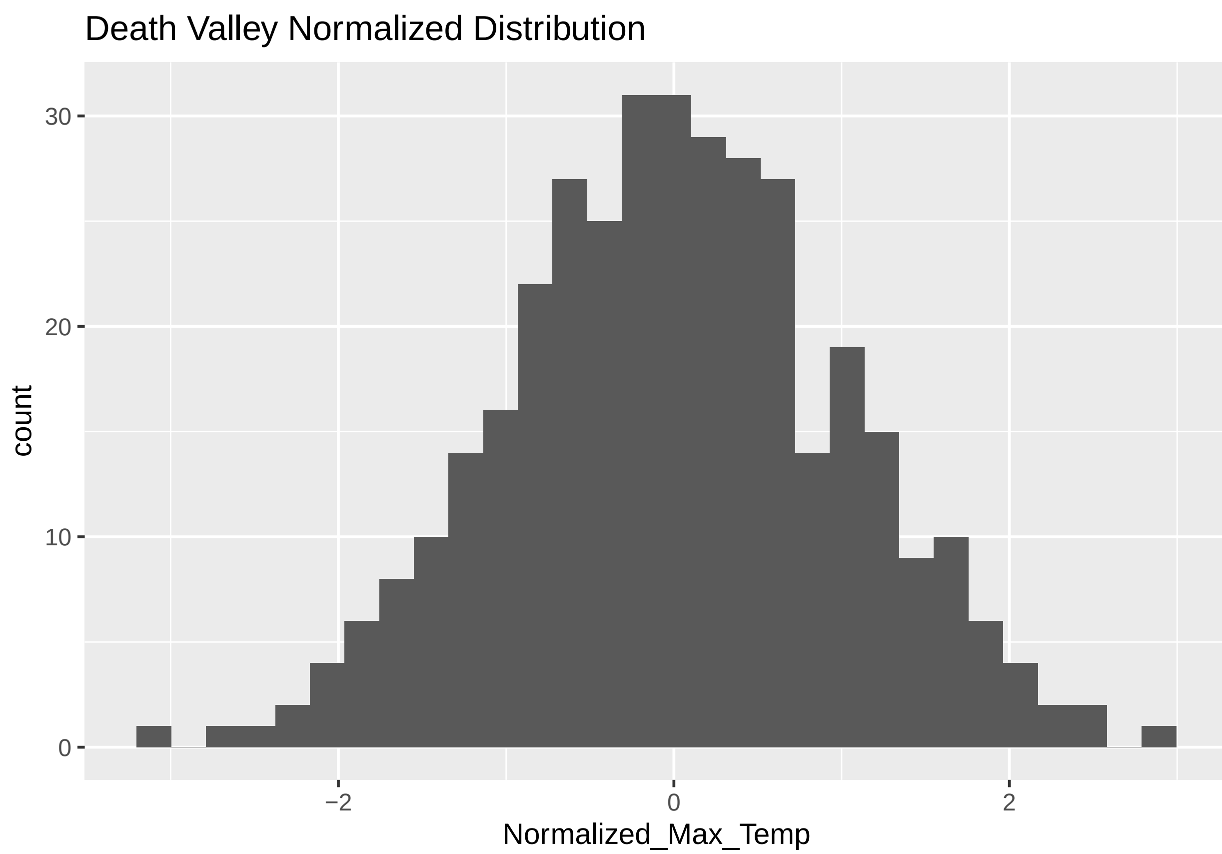
Napa Normalized Distribution
count
30
20
10
0
−3
−2
−1
0
1
2
Normalized_Max_Temp

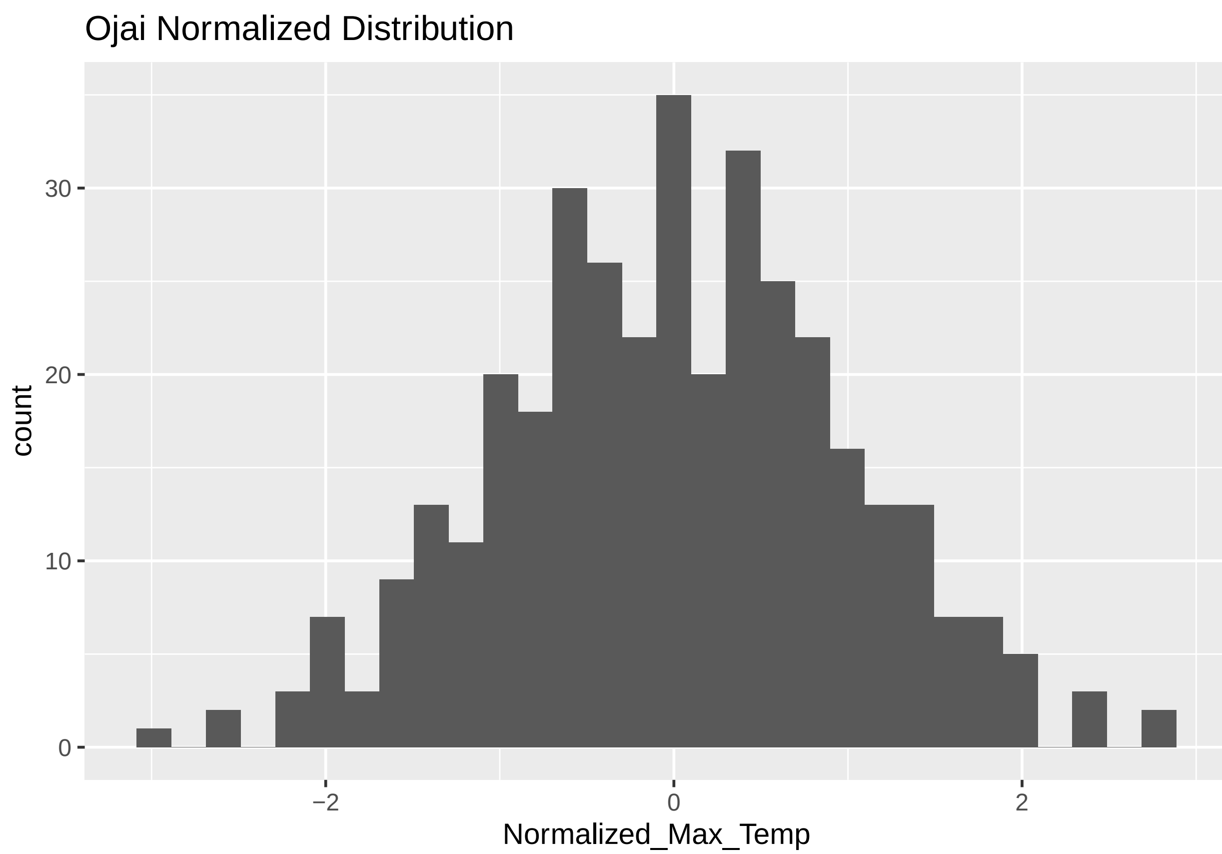

San Diego Normalized Distribution
count
30
20
10
0
−2
0
2
Normalized_Max_Temp

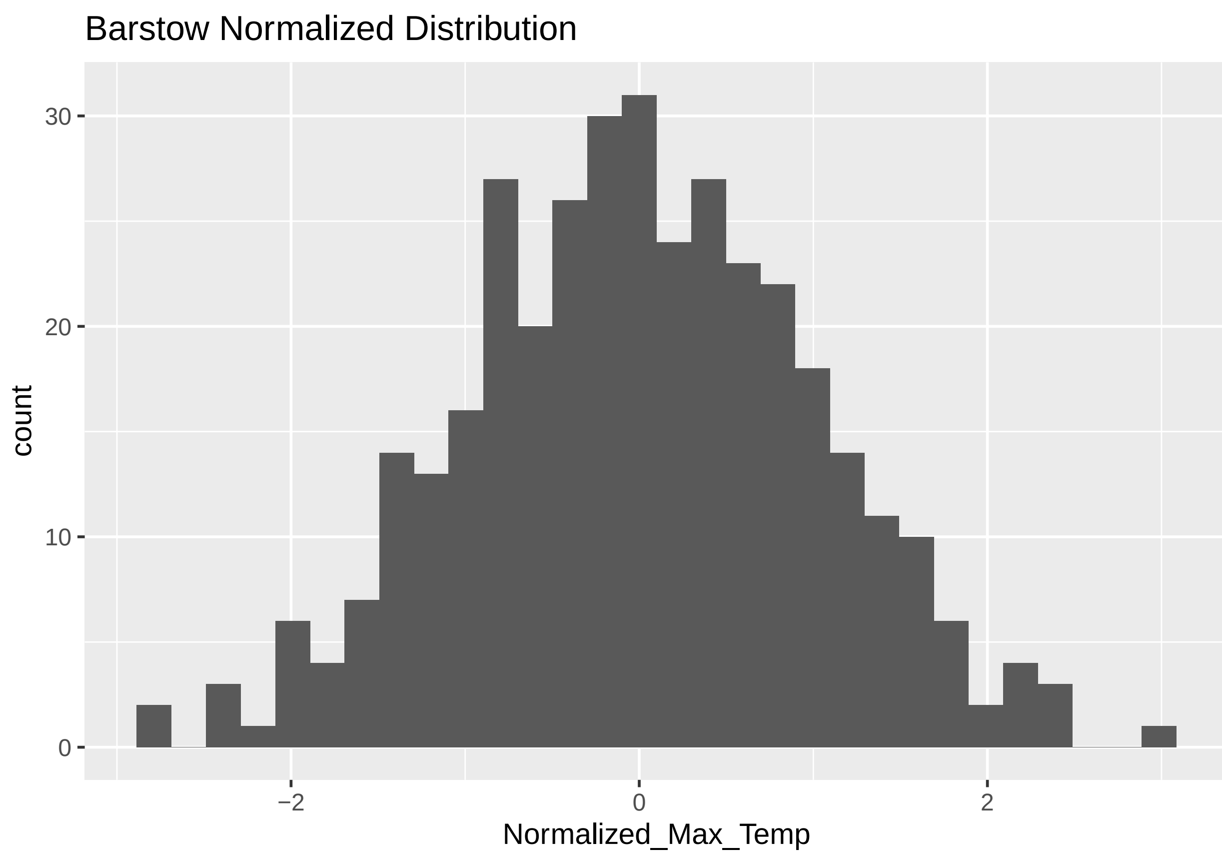

Fresno Normalized Distribution
count
30
20
10
0
−3
−2
−1
0
1
2
3
Normalized_Max_Temp



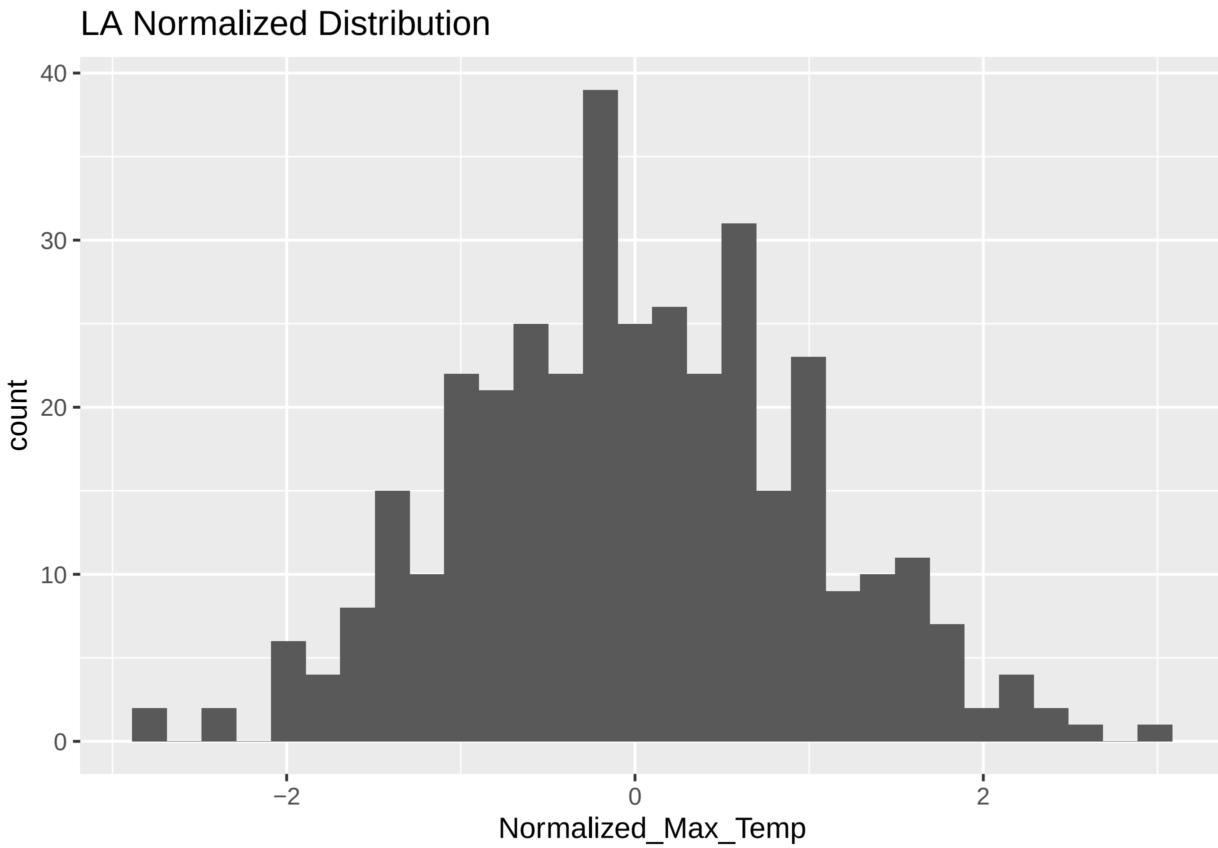
Santa Cruz Normalized Distribution
count
40
30
20
10
0
−2
0
2
Normalized_Max_Temp

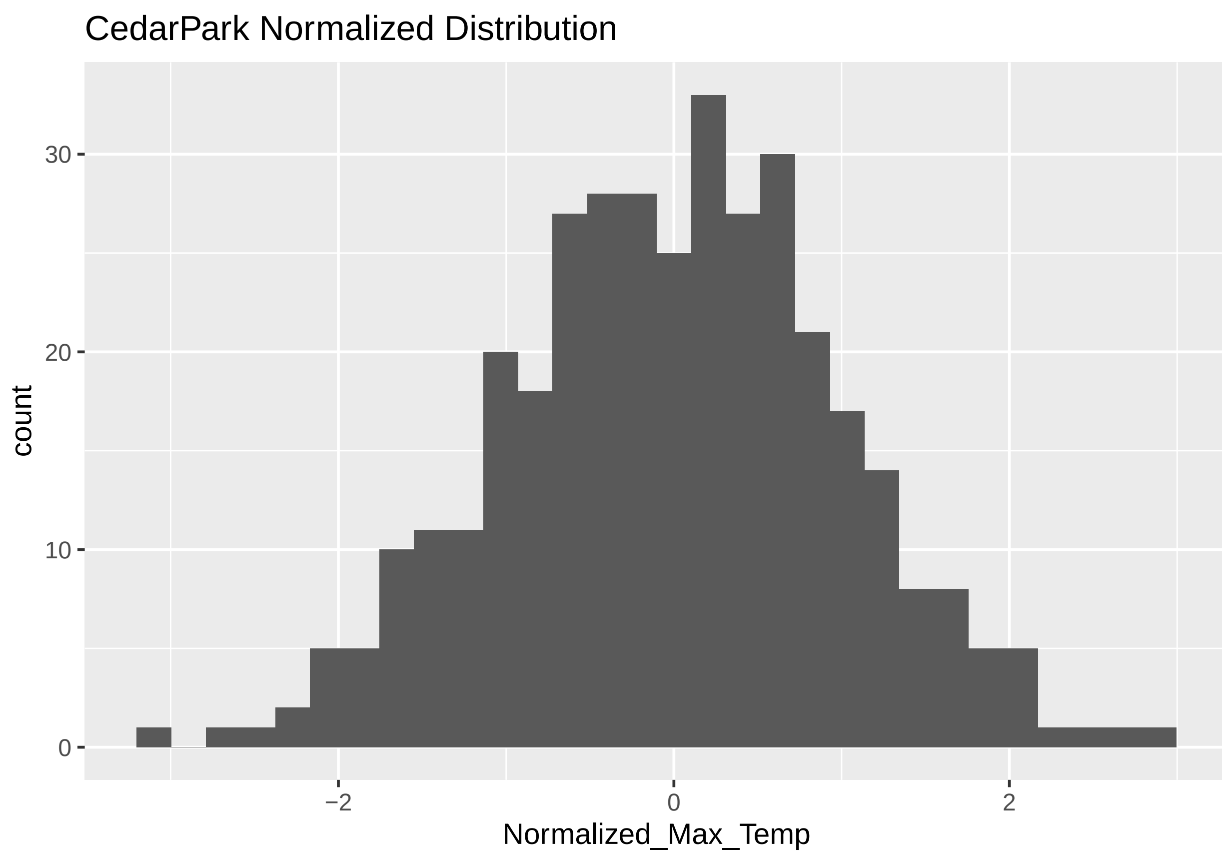

Death Valley Normalized Distribution
30
20
10
0
count
−2
0
2
Normalized_Max_Temp

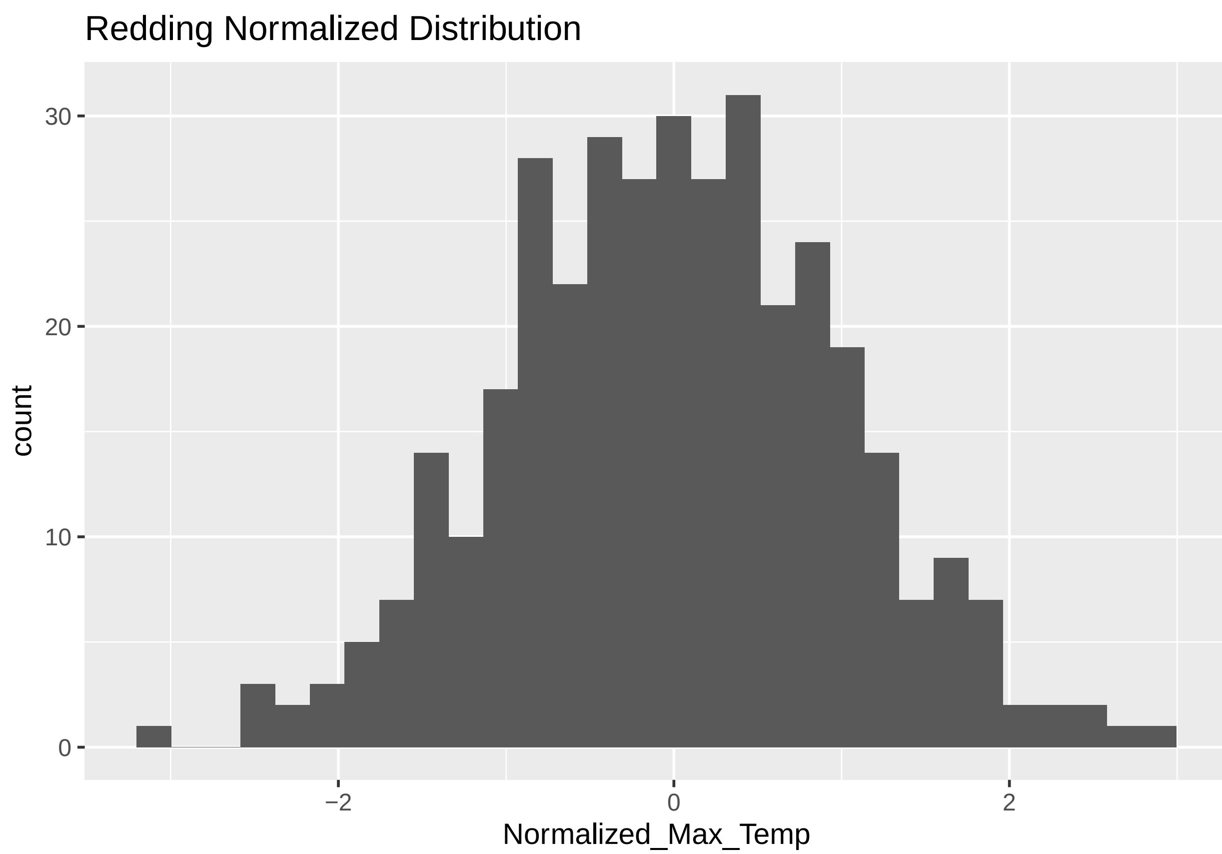

Ojai Normalized Distribution
count
30
20
10
0
−2
0
2
Normalized_Max_Temp



Barstow Normalized Distribution
30
20
10
0
count
−2
0
2
Normalized_Max_Temp

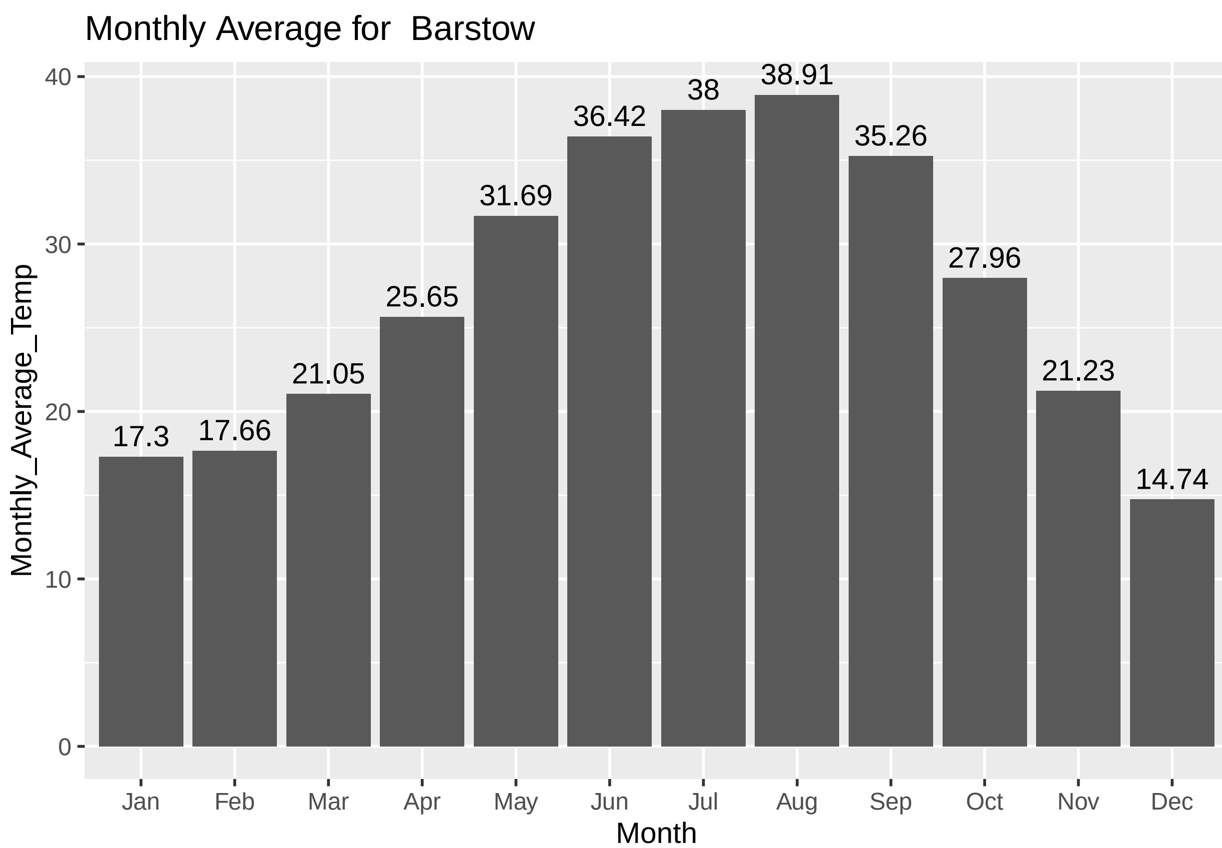

LA Normalized Distribution
count
40
30
20
10
0
−2
0
2
Normalized_Max_Temp

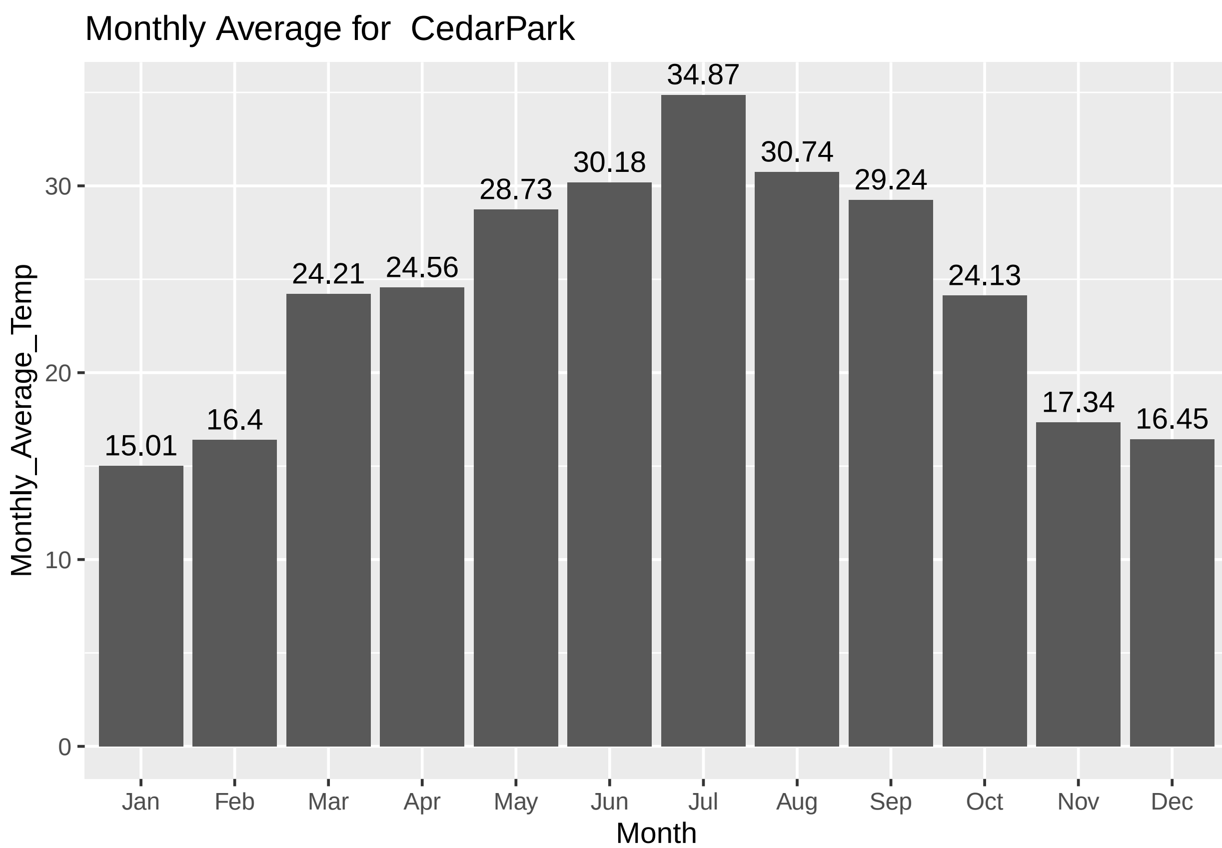

CedarPark Normalized Distribution
30
20
10
0
count
−2
0
2
Normalized_Max_Temp



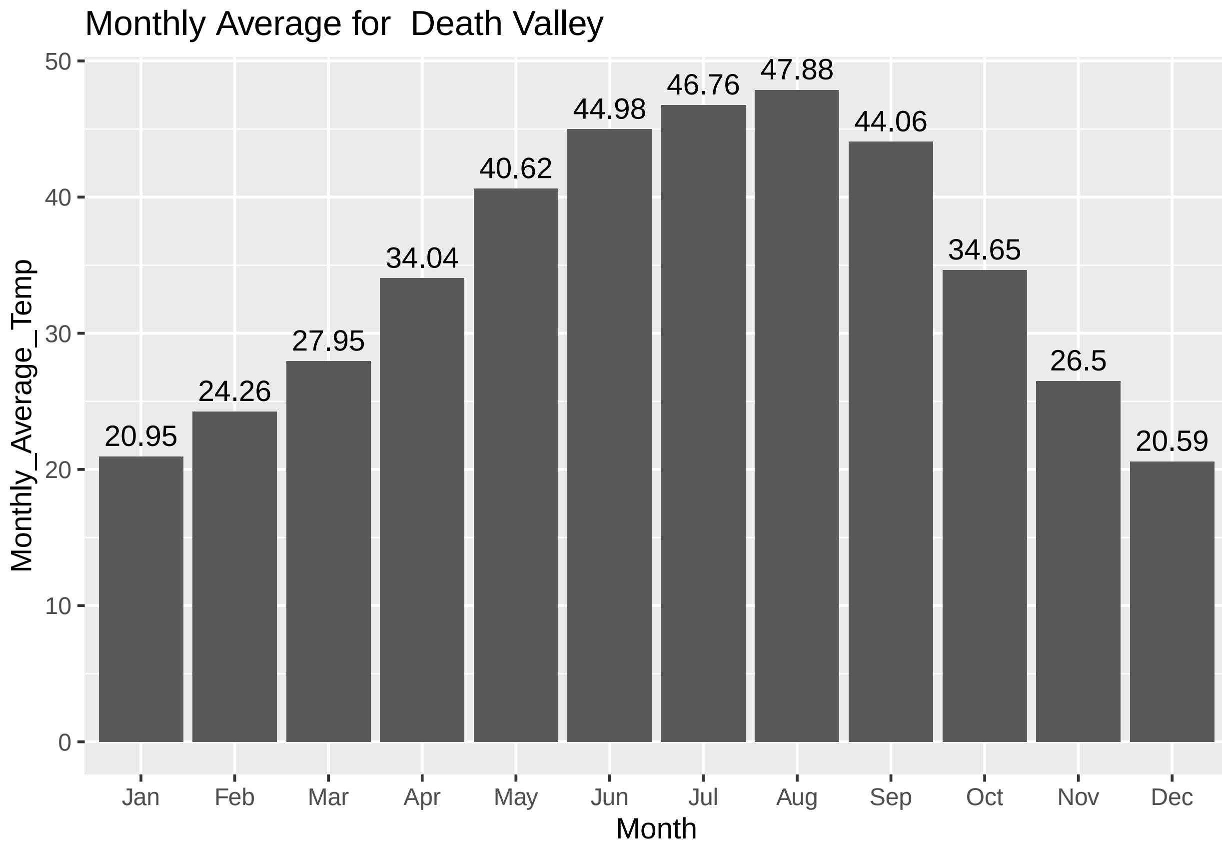
Results

From the analysis, it was found that the months of Jun-July records the highest average temperatures in many locations in the California State. Low monthly average (max) temperatures are recorded in the first quarter (Jan- April). The plots below shows the montly average temperature for the locations in California.

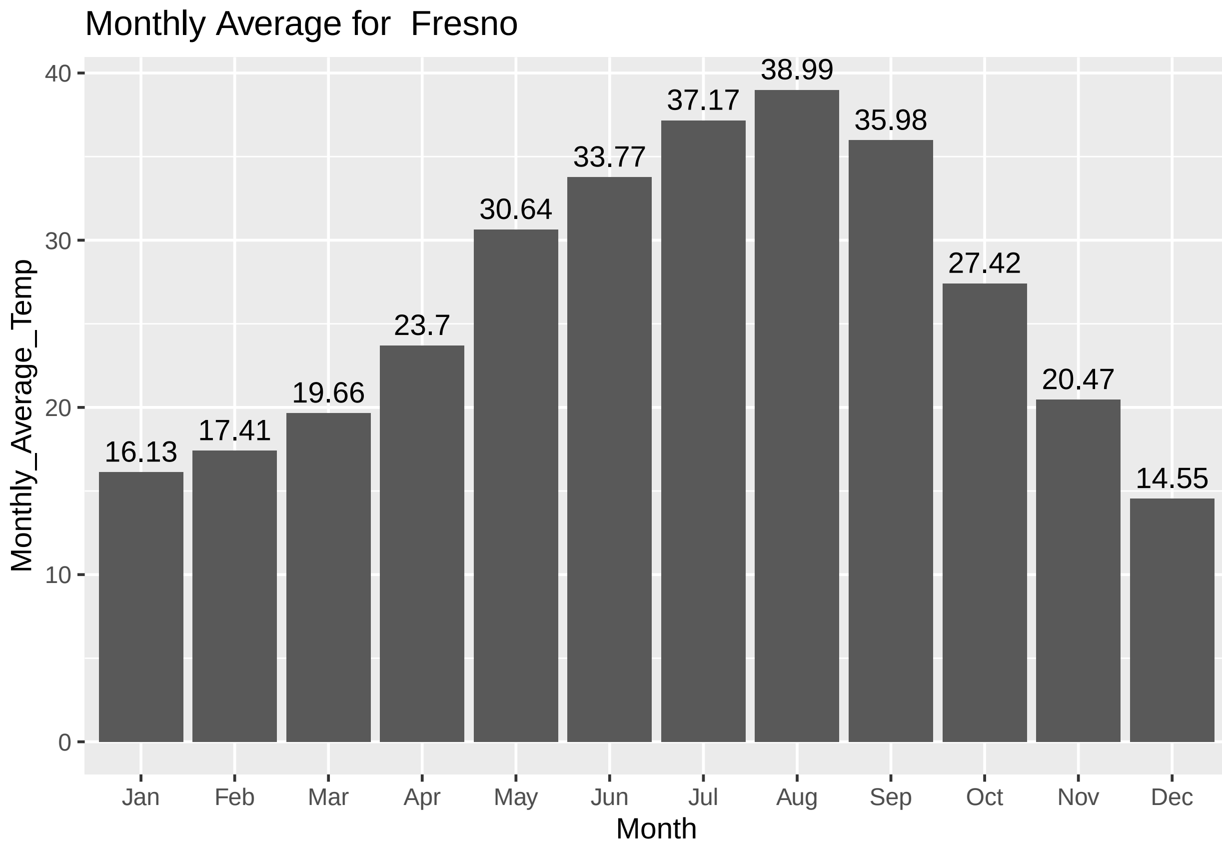

Monthly Average for Barstow
Monthly_Average_Temp
0
10
20
30
40
17.3
17.66
21.05
25.65
31.69
36.42
38
38.91
35.26
27.96
21.23
14.74
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

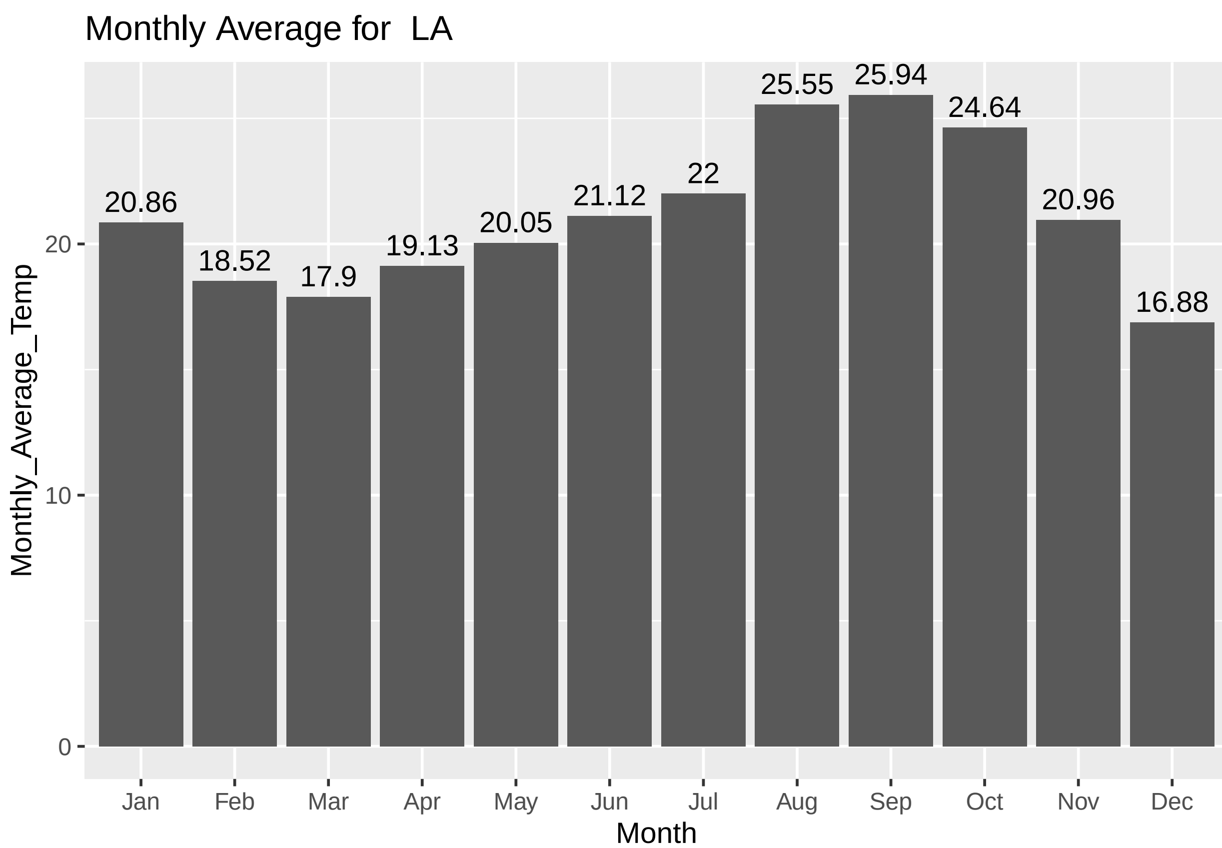


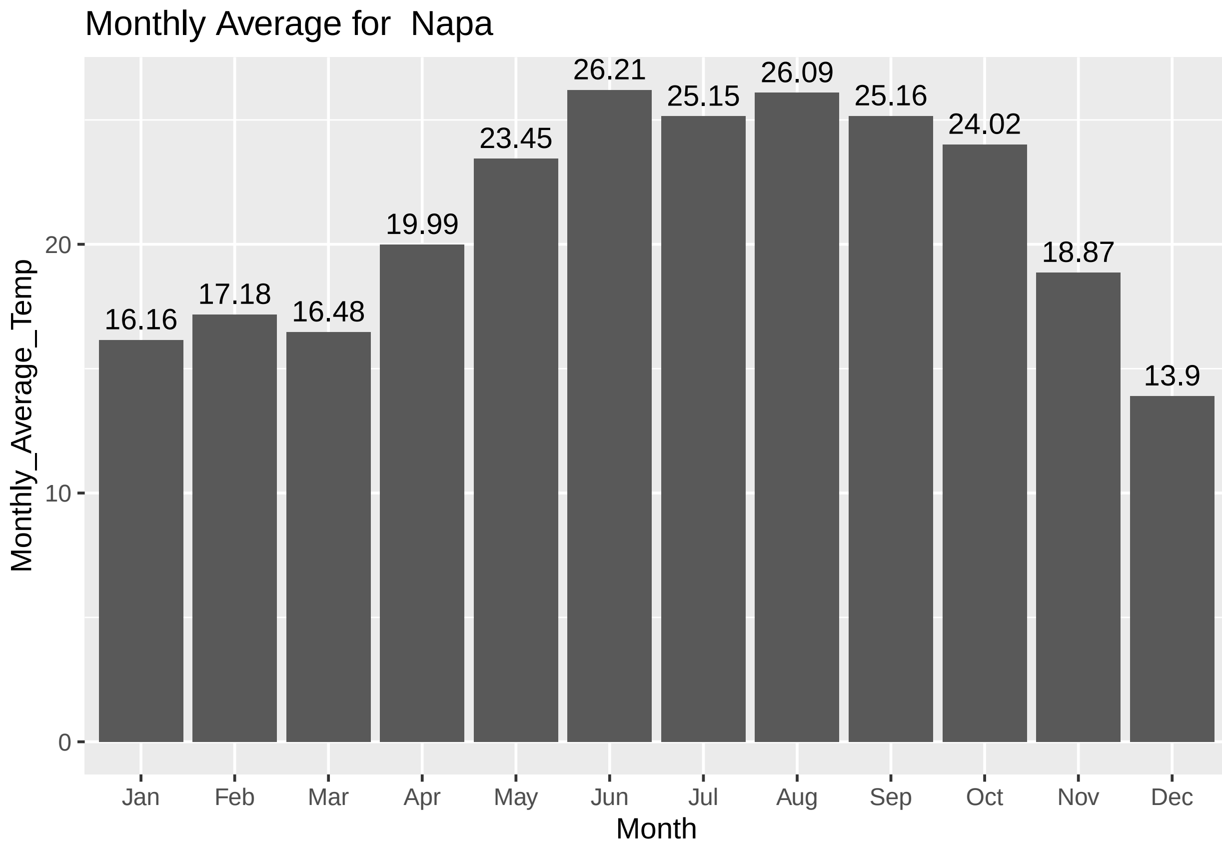
Monthly Average for CedarPark
Monthly_Average_Temp
0
10
20
30
15.01
16.4
24.21
24.56
28.73
30.18
34.87
30.74
29.24
24.13
17.34
16.45
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month



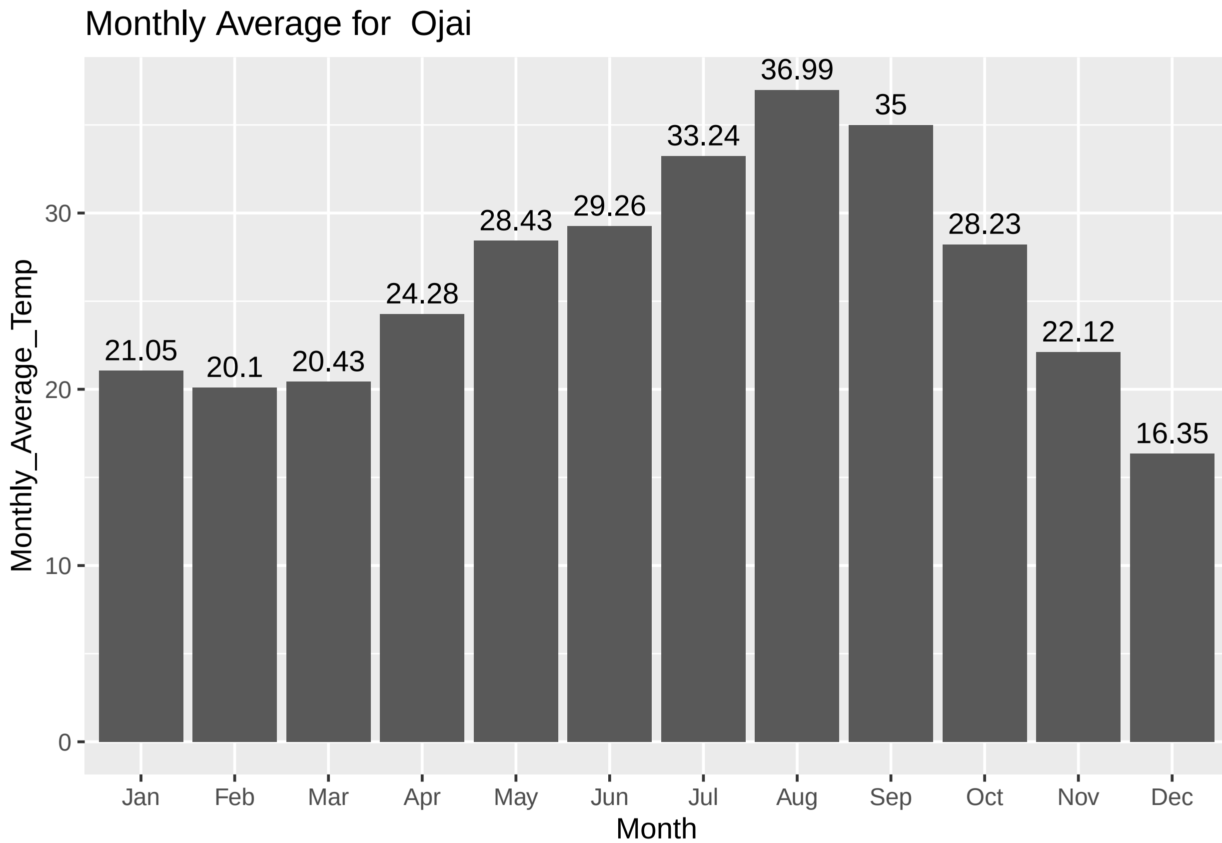
Monthly Average for Death Valley
Monthly_Average_Temp
Month
0
10
20
30
40
50
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
20.95
24.26
27.95
34.04
40.62
44.98
46.76
47.88
44.06
34.65
26.5
20.59

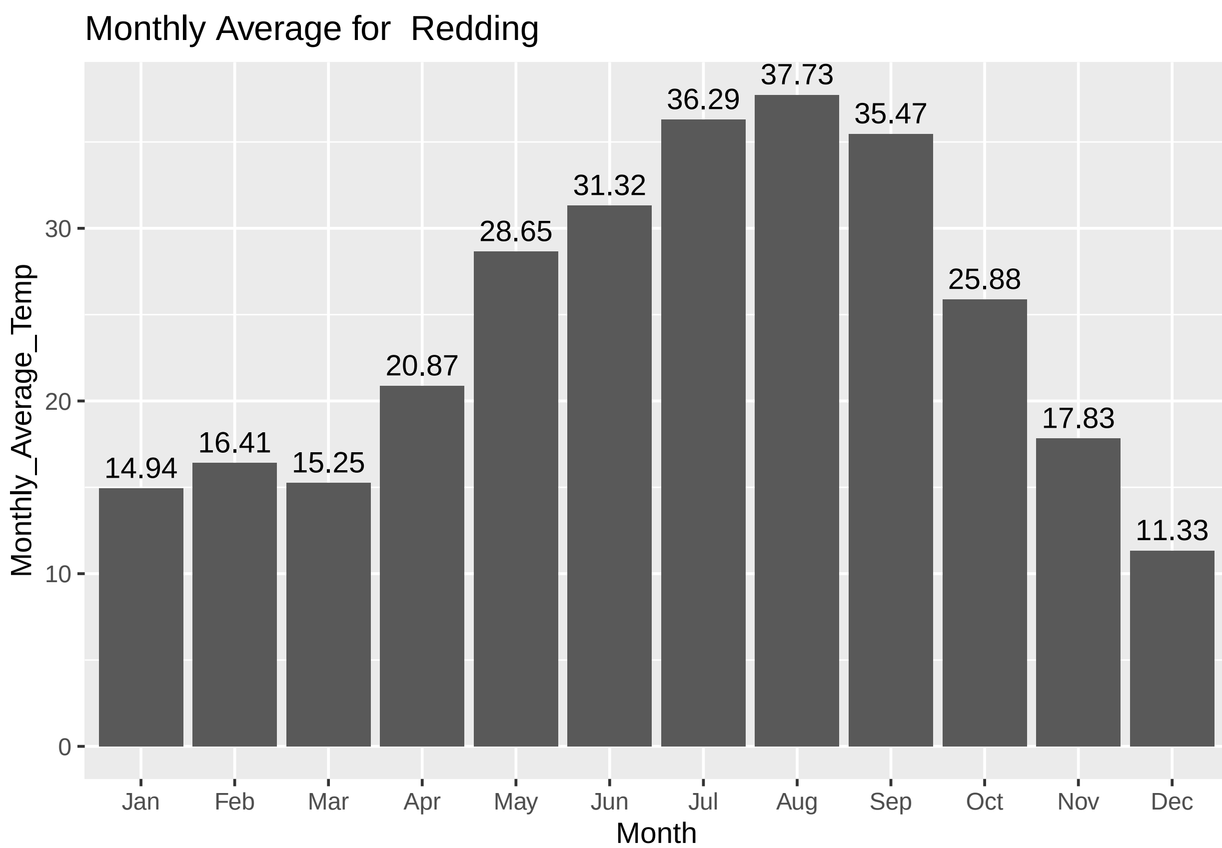

Monthly Average for Fresno
Monthly_Average_Temp
0
10
20
30
40
16.13
17.41
19.66
23.7
30.64
33.77
37.17
38.99
35.98
27.42
20.47
14.55
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

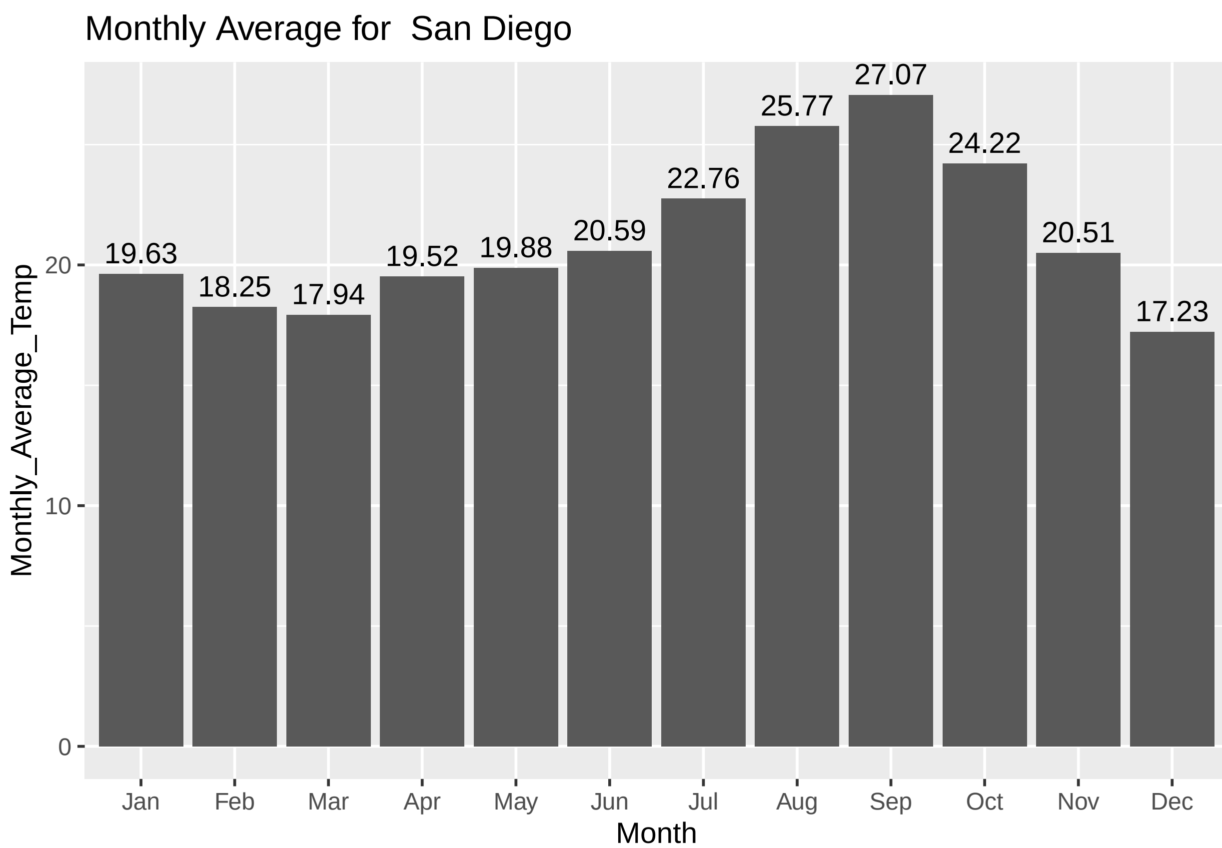

Monthly Average for LA
Monthly_Average_Temp
Month
20.86
18.52
17.9
19.13
20.05
21.12
22
25.55
25.94
24.64
20.96
16.88
0
10
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec



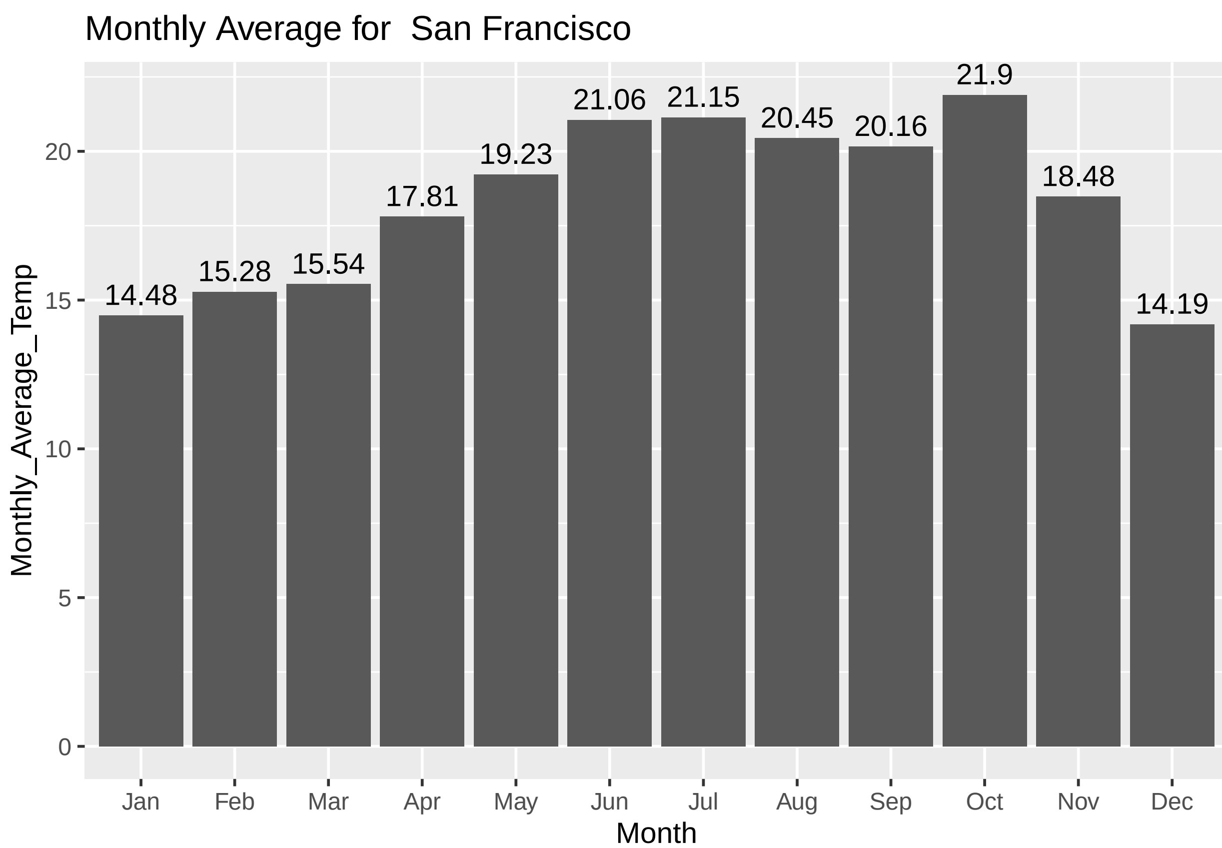
Monthly Average for Napa
Monthly_Average_Temp
0
10
20
16.16
17.18
16.48
19.99
23.45
26.21
25.15
26.09
25.16
24.02
18.87
13.9
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month

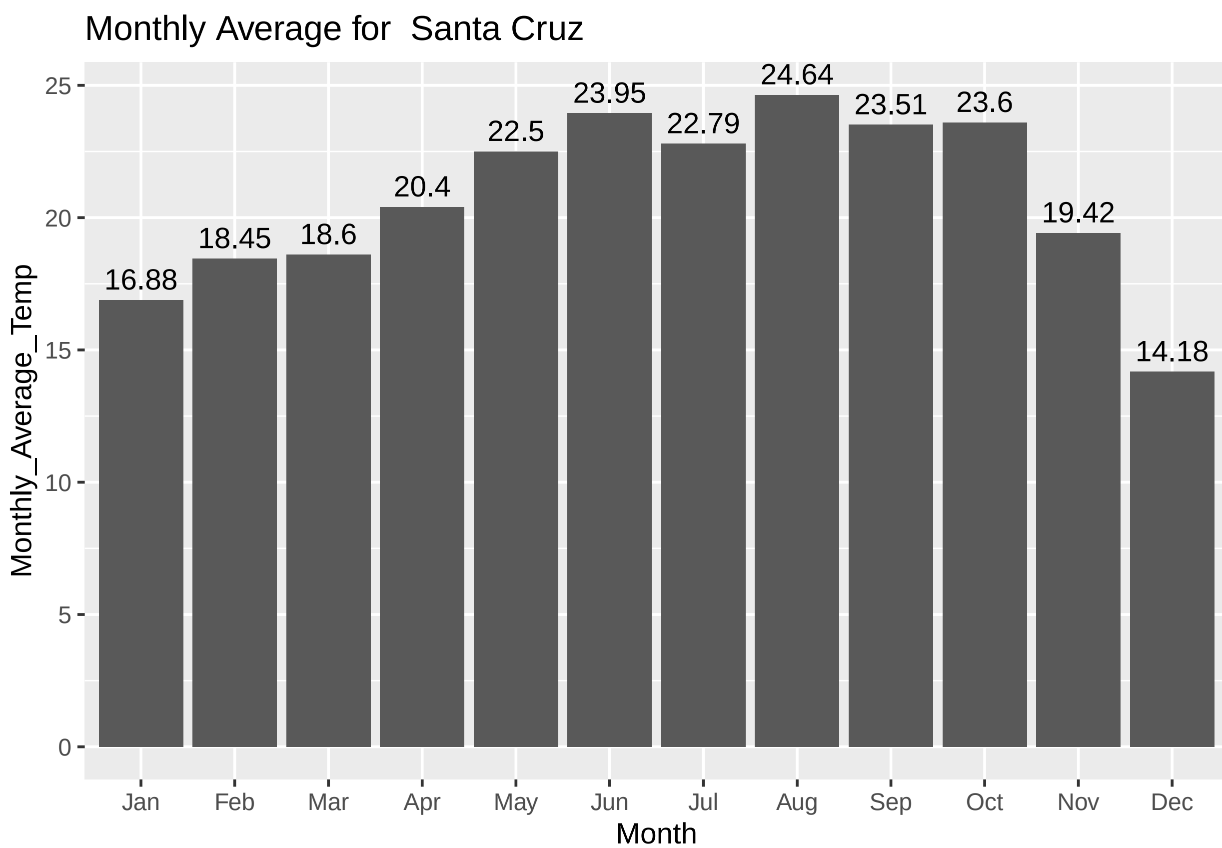

Monthly Average for Ojai
Monthly_Average_Temp
Month
0
10
20
30
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
21.05
20.1
20.43
24.28
28.43
29.26
33.24
36.99
35
28.23
22.12
16.35


Monthly Average for Redding
Monthly_Average_Temp
Month
0
10
20
30
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
14.94
16.41
15.25
20.87
28.65
31.32
36.29
37.73
35.47
25.88
17.83
11.33



Monthly Average for San Diego
27.07
25.77
24.22
22.76
20.59
20.51
19.63
18.25
17.94
19.52
19.88
17.23
20
10
0
Monthly_Average_Temp
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Month


Monthly Average for San Francisco
Monthly_Average_Temp
Month
0
5
10
15
20
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
14.48
15.28
15.54
17.81
19.23
21.06
21.15
20.45
20.16
21.9
18.48
14.19



From the statistical analysis to determine whether there are differences in (max) temperatures at different locations, and whether there are (statistically significant) differences between months., there was a p-value of $2e - 16$.

The `p-value` <0.05 indicating that the ANOVA has detected a significant effect of the factors which in this case is different locations and different months. Below are the Multiple comparisons (post-hoc comparisons) of different locations and different Months to help quantify the differences between groups and determine the groups that significantly differ from each other.

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## Location     10   2374   237.4   20.48 <2e-16 ***
## Month        11   4149   377.2   32.54 <2e-16 ***
## Residuals   110   1275    11.6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The best time series model to predict maximum temperatures in all location was determined by `auto.arima` and was found to be ARIMA(0,1,3). In comparing the actual and the predicted values, this model had a Root Mean Squared Error of 14.18539

Summary

- The months of Jun-July records the highest average temperatures in many locations in the California State. Low monthly average (max) temperatures are recorded in the first quarter (Jan- April).

Bibliography

Appendix

```
# Read the datasets
metadataCA <- read.csv("metadataCA.csv")
maxtempcalifornia <- read.csv("MaxTempCalifornia.csv")

# Preview the head
head(metadataCA)
head(maxtempcalifornia)

# Tranform California from wide to long
maxtempcalifornia_long <- maxtempcalifornia %>%
  gather(Location, Max_Temp, -c(X))
maxtempcalifornia_long$Location <- maxtempcalifornia_long$Location %>%
  str_replace("\\.", " ")
maxtempcalifornia_long$Date <- ymd(maxtempcalifornia_long$X)
maxtempcalifornia_long <- maxtempcalifornia_long %>%
  subset(select=-X)
head(maxtempcalifornia_long)

# Check the numerical summaries
summary(metadataCA)
summary(maxtempcalifornia_long)

# plot scatter matrix
metadataCA %>%
  ggpairs()

maxtempcalifornia_long %>%
  ggpairs()

# Distribution of 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)
}

# Normalize the locations distribution
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()+
```

```r
    ggtitle(paste(location,"Normalized Distribution"))
  print(p)
}

# Statistical analysis
summary(mod.aov <- aov(
  Monthly_Average_Temp ~ Location + Month,
  data = monthly_average_temp
))

TukeyHSD(mod.aov)

# Select only the San Francisco data
sanfrancisco <- maxtempcalifornia_long %>%
  filter(Location=="San Francisco") %>%
  dplyr::select(Date, Max_Temp)

# Create a time Series
sanfrancisco_xts = xts(sanfrancisco[, -1], order.by = sanfrancisco$Date)
head(sanfrancisco_xts)

# create and Find the best ARIMA model
fit <- auto.arima(
  sanfrancisco_xts
)
plot(forecast(fit, h=20))

summary(fit)

# Select the predicted for the 1st-8th August 2012
pred_period = yday(
  seq(ymd('2012-08-01'),ymd('2012-08-08'), by='1 day')
)

# Get predictions of the entire year
preds <- forecast(fit, h=365)$fitted

# Predicted Maximum temperature for all locations for 1st-8th August
req_preds <- preds[pred_period]
print(req_preds)

# Comparison
cal_01_08<-maxtempcalifornia_long%>%
  filter((Date>="2012-08-01")&(Date<="2012-08-08"))
cal_01_08$pred <- req_preds
cal_01_08

# Calculate root mean squared error
rmse(cal_01_08$Max_Temp, cal_01_08$pred)

# Merge Metadata with Maximum Temperature
merged <- merge(
  maxtempcalifornia_long,
```

```r
  metadataCA,
  by.x="Location",
  by.y="ï..Location"
)
# specify columns containing coordinates of locations
coordinates(merged) <- c("Long", "Lat")

# set coordinate reference system
crs.geo1 <- CRS("+proj=longlat")
proj4string(merged) <- crs.geo1

# Select from locations that are not Redding, San Francisco and Death Valley
train_data <- merged[
  !merged$Location %in% c("Redding", "San Francisco", "Death Valley"),
]
# Fit Spatial Lag Model
spl.model <- lagsarlm(
  Max_Temp~Elev,
  data=train_data,
  nb2listw(
    knn2nb(
      knearneigh(coordinates(train_data), longlat = TRUE)
    )
  )
)

test_data <- merged[
  (merged$Location %in% c("San Francisco", "Death Valley")) & (merged$Date=="2012-01-01"),
]
test_data_lw <- nb2listw(
  knn2nb(
    knearneigh(coordinates(test_data), longlat = T)
  )
)
row.names(test_data) = attributes(test_data_lw)$region.id
spl.preds <- predict(
  spl.model,
  test_data,
  test_data_lw
)
print(spl.preds)

# spl model summary
summary(spl.model)
```