Adith Arun & Chloe Warren

Geostatistics: Understanding Spatial Data

Professor Scot Miller

May 13th, 2021

Final Project: Daily Mean Temperature in California

For the final project, we wanted to model temperature in California. We chose California not only because of our attachments to the state but also because of its wild variability in temperature between different regions of the state. We knew that this would be much more challenging than anything we had modeled in class thus far, and this pushed us to explore it with careful attention to issues and questions that came up. We were surprised to see that we were able to create a visually pleasing variogram, however kriging, as expected, was a disaster. While our kriging results were disappointing, we learned a great deal about why California is such a complicated area to model and new tools like the greatest circle distance to run on our data. In total, the final project was a wonderful opportunity for hands-on learning.

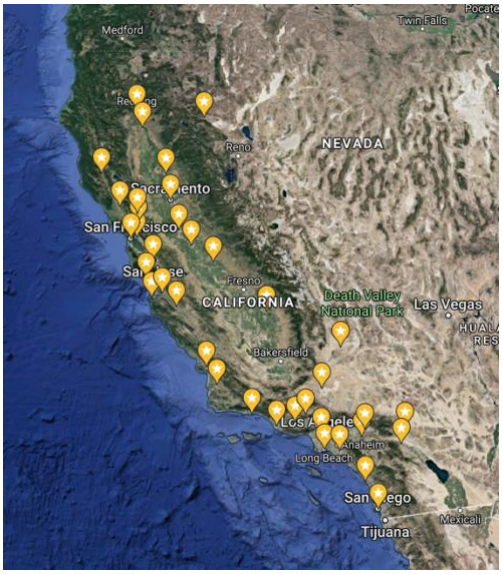


Figure 1: Map of Locations (based on city not station)

Our data is taken from NOAA. We are looking at the daily mean temperature from 37 cities (all marked in Figure 1) in California from January to March 2010. March was found to be a much smaller data set, due to missing values, and so we ended up only using January and February to fit variograms and run kriging on. We have available to us from NOAA the following records (as shown in Figure 2): date, datatype, station, value, "fl_c", units, city, and the

city's respective id number. The units are marked as unknown, but reading the technical documentation from NOAA we surmised that all reported values were on the same scale.

```
# A tibble: 6 x 8
  date                datatype        station           value fl_c  units            city                cityid
  <chr>               <chr>           <chr>             <int> <chr> <chr>            <chr>               <chr>
1 2010-01-01T00:00:00 DLY-TAVG-NORMAL GHCND:USC00044405   474 Q     unknown; see docs Yucca Valley, CA US CITY:US060048
2 2010-01-02T00:00:00 DLY-TAVG-NORMAL GHCND:USC00044405   475 Q     unknown; see docs Yucca Valley, CA US CITY:US060048
3 2010-01-03T00:00:00 DLY-TAVG-NORMAL GHCND:USC00044405   476 Q     unknown; see docs Yucca Valley, CA US CITY:US060048
4 2010-01-04T00:00:00 DLY-TAVG-NORMAL GHCND:USC00044405   477 Q     unknown; see docs Yucca Valley, CA US CITY:US060048
5 2010-01-05T00:00:00 DLY-TAVG-NORMAL GHCND:USC00044405   478 Q     unknown; see docs Yucca Valley, CA US CITY:US060048
6 2010-01-06T00:00:00 DLY-TAVG-NORMAL GHCND:USC00044405   479 Q     unknown; see docs Yucca Valley, CA US CITY:US060048
```

Figure 2: First Few Rows of Dataset

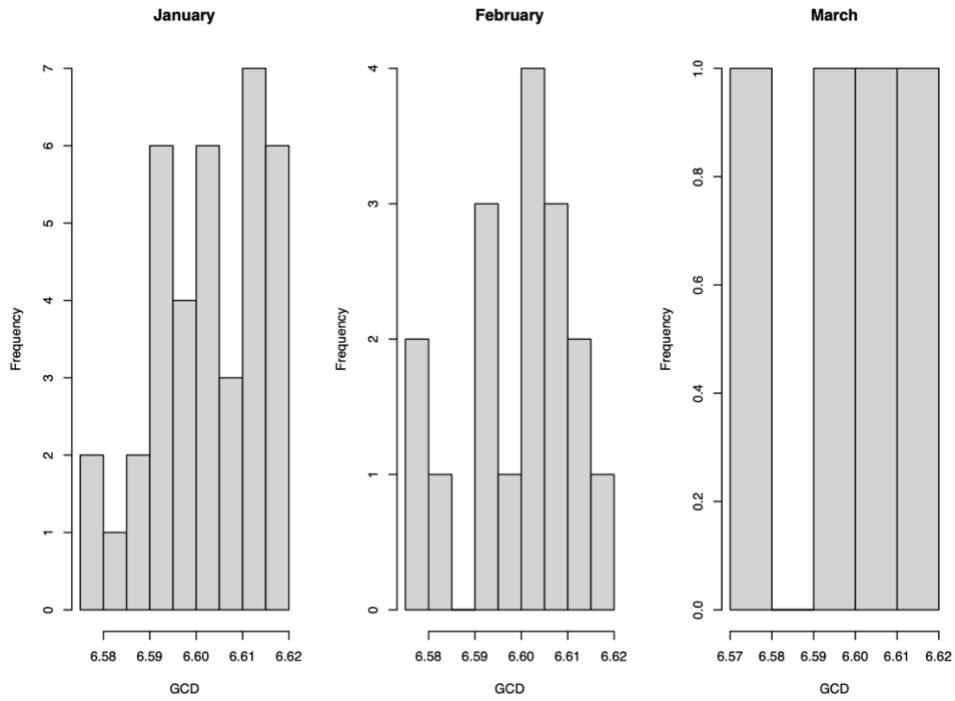


Figure 3: GCD (Greatest Circle Distance) Results

The first issue we ran into with our data was with our location format. Our data location was in the form of longitude and latitude. The issue with this was that it is not measured equally, so we had to convert the locations with some help during office hours. With help, we transformed these two-dimensional values into a distance metric using the greatest circle distance (GCD). Once we had the right format (utilizing GCD) we started to explore the data and ran into

another issue that hurt our ability to model, such as missing March values. We knew that there was an issue with March from our GCD result (Figure 3).

We ran a tree diagram of values to cities to check that they all had the same amount of observations. As you can see in the tree diagram (Figure 4) there is a disproportionate distribution of values per location which indicates something fundamentally wrong with the data collected. This is a record of daily mean temperature so the only reason why there should be fewer observed values in one location compared to another would be if that location has missing values or if the one with more has duplicate values.

After further investigation, it became clear to us that they were missing values specific to March. This made fitting a variogram of March impossible without data interpolation. Ultimately, we only had data for four cities and decided that the data was too sparse to interpolate with any degree of accuracy. Thus, we decided to leave March out of variogram fitting and did variogram fitting for January and February only given the circumstances of the data we had. In the month of January, we had data for 36 cities and for February, we had data for 17 cities.

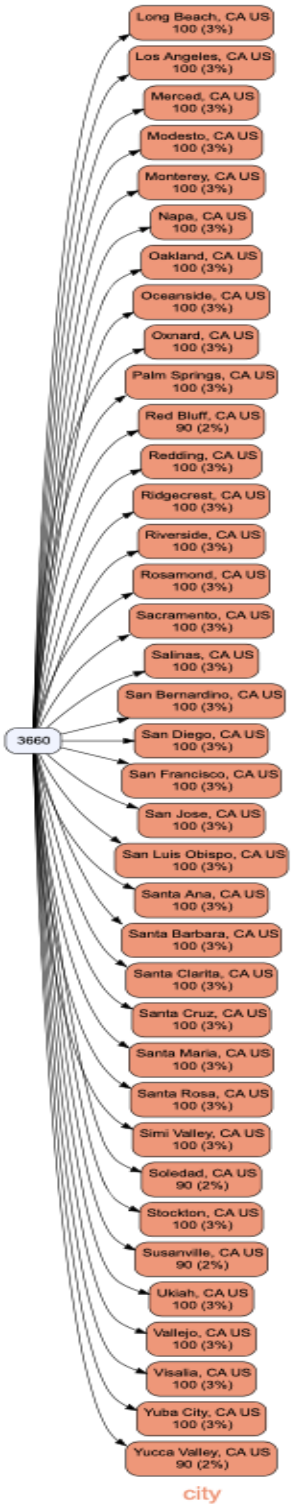


Figure 4: Tree diagram

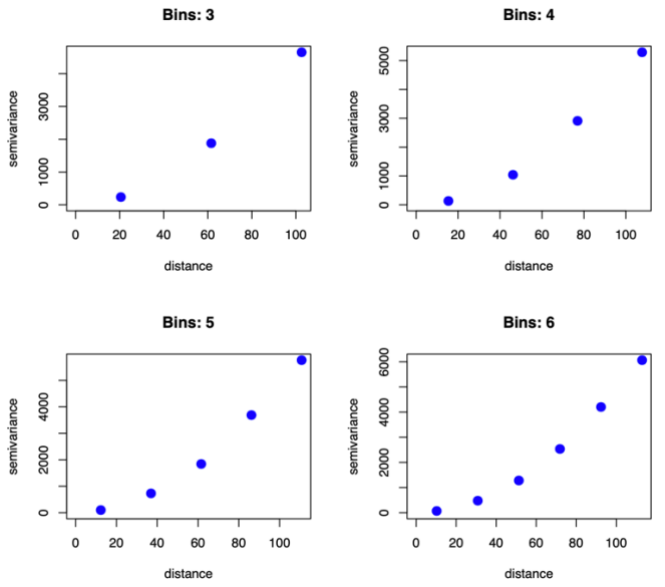
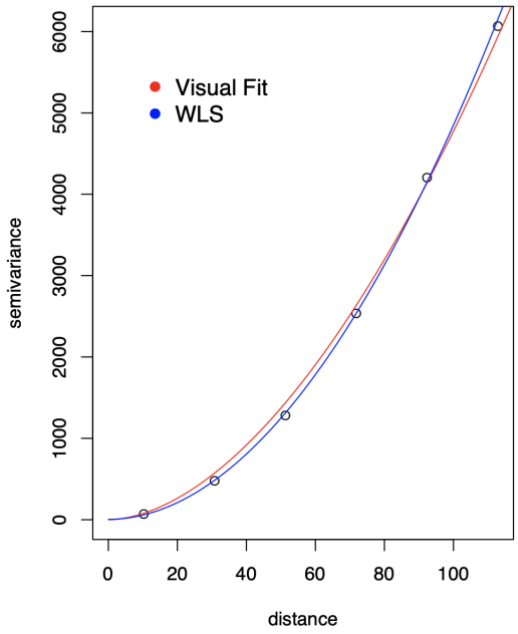


Figure 5: January Experimental Variograms



6: January Variogram

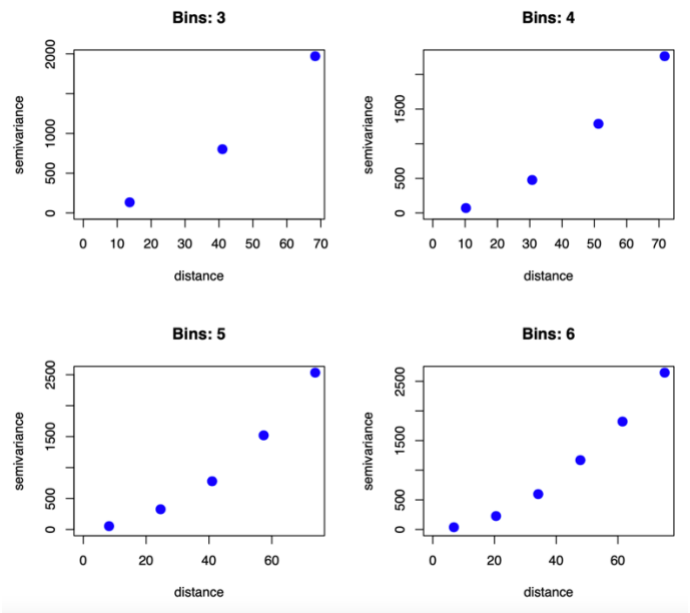


Figure 7: February Experimental Variograms

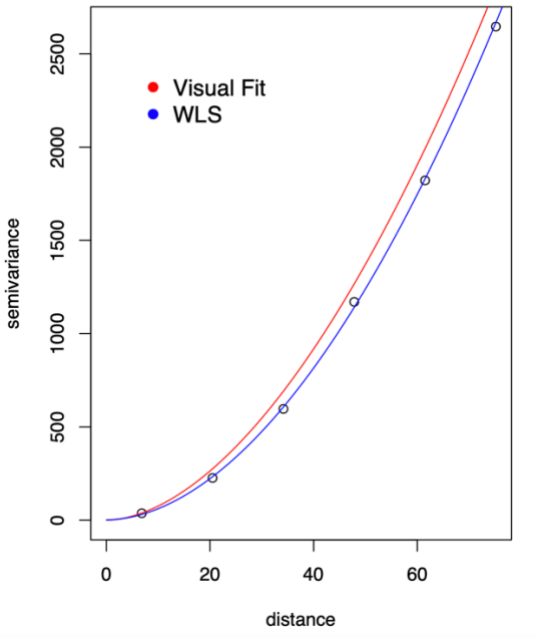


Figure 8: February Variogram

For variogram fitting, we chose to use 6 bins for both January (Figures 5 and 6) and February (Figures 7 and 8) because we felt it reproduced the structure of the curve the best. We

tried to fit the variogram model visually by playing with parameters and then ran a weighted least-squares fit which modeled the curved better for both months. Further details into how we ran WLS can be found in the attached code. The actual fitting of the variogram was a straightforward process given our experience from class assignments, but it was particularly exciting to see that our data followed an intrinsic non-stationary model. The raw variogram cloud showed us that our data followed a power model. And since we did not have much exposure to intrinstic non-stationary models in class, we were happy to get the experience working with one. The power model covariance is of the form: $C(h) = -bh^c$, and variogram model is of the form: $\gamma(h) = bh^c$. Note that b > 0 and 0 < c < 2 and must be fitted from the data. We learned that the shape can vary immensely depending on the parameters, which we knew from class, but going through visually fitting it made us see this effect. The power model's self-similar structure was evident – an interesting phenomenon itself, reminiscent of Mandelbrot's fractals.

The next portion of our project was to estimate the mean monthly temperature at different locations in California. We used standard kriging, ordinary kriging, and linear regression. The specifics of all three methodologies can be found in our code. To assess the accuracy of our estimation procedures, we essentially performed a leave one out cross-validation (LOOCV). This meant that for method i, we left out a city, city j, and used the data from all other cities to estimate city j's mean monthly temperature for month k. Then, iterating through all cities we could measure the root mean square error (RMSE) between our estimate from method i for month k. Since we already had plenty of experience deriving our kriging equations in the various problem sets, we opted to use the *geoR* package built-in kriging models. We chose to go with linear regression as a standard of comparison against the kriging methods. Note that for the linear model, the only covariates used were the greatest circle distance and an intercept – a very simple

model. Once we performed ordinary and simple kriging, it became pretty clear that kriging was not effective with this data so we felt that these two options were enough. Of course, universal kriging, block kriging or other adaptations of kriging may be better at estimation and this is a future avenue of development for this project. We could also look to incorporate more data types such as distance from the coast, altitude, and mean monthly rainfall to improve our estimates.

After running kriging, we were ready to look at our estimation errors, quantified by the RMSE. The RMSE CV errors can be found in the figure below (Figure 9).

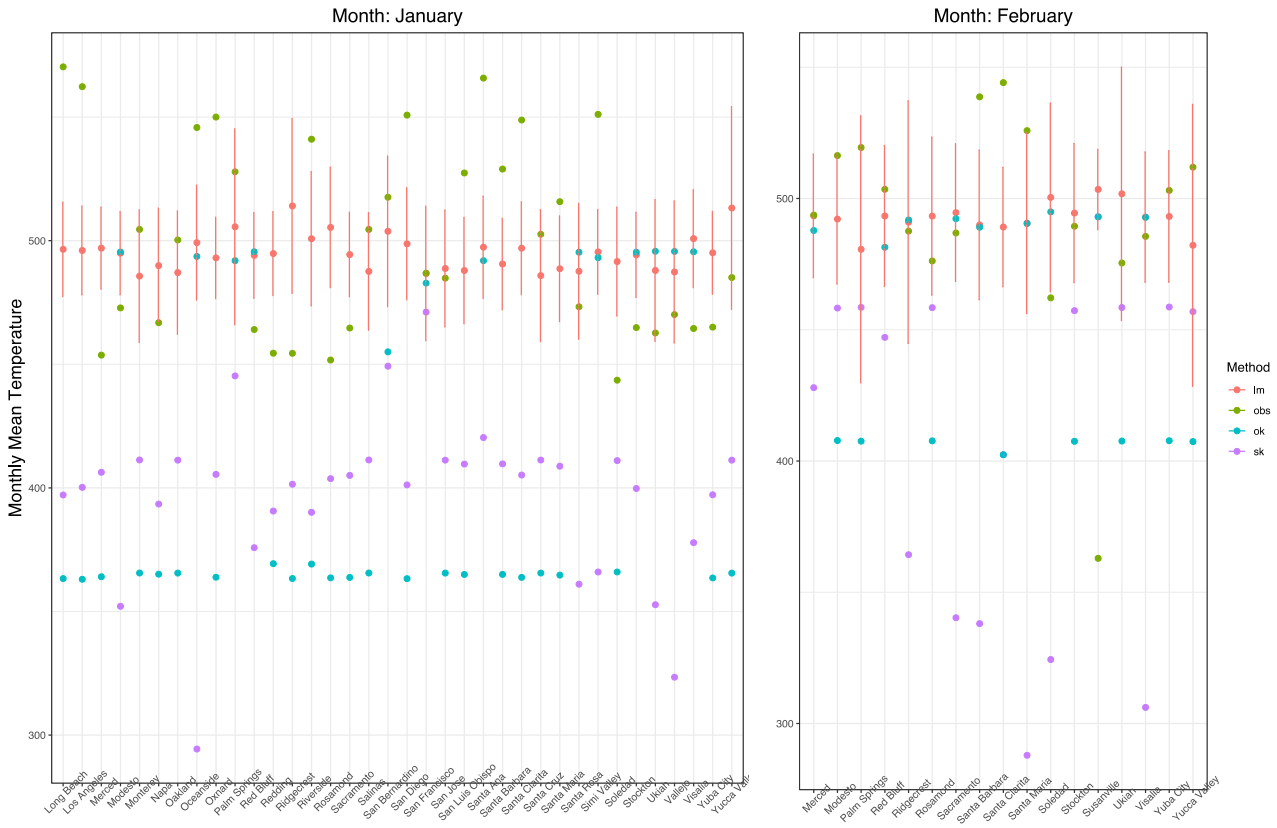


Figure 9: RMSD

The green is the actual observed value, purple represents the simple kriging estimate, blue represents the ordinary kriging estimate, and red is the linear regression estimate. Ordinary kriging did slightly better than Standard kriging, but both were extremely inaccurate. The error bars around the linear model represent the 95% confidence interval. The reason why there are

no error bars on kriging results is that we are using a power model. Taking the power covariance model at a point of which the distance is zero will always be zero. So, it was not possible to include confidence intervals for the kriging models since the tools we were using

LM	OK	SK	MONTH
1546.837	13527.123	12883.07	JAN
1892.616	6044.117	14292.68	FEB

could not accurately represent the error in a meaningful way. One *ad hoc* method to estimate confidence intervals for the kriging estimates could be to use a non-zero distance in the error estimation process. The exact value to use could be the estimated radius of each city and could be developed in a further extension of this project. Linear regression performed best. Overall, our results were contrary to our experiences in class where the kriging estimates usually performed better than the linear models. Figure 10 is a summary representation of the LOOCV RMSE performance for each month and model type.

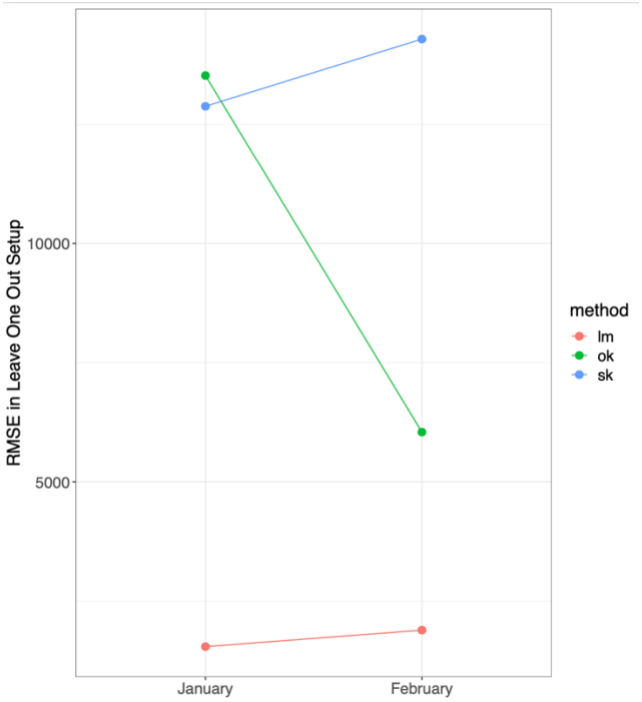


Figure 10

We can see that linear regression had the lowest cross-validated error out of the three. The actual numerical values on the y-axis are not important for analysis but we can evaluate the relative differences between each of the three methods. Simple kriging stayed relatively consistent in error compared to ordinary kriging which changed between the two months. This may be due to the difference in data values for each month and the difference in weather variability between the two months.

Often, assumptions that simplify a statistical problem can be made at the expense of capturing the exact process. In this vein, we considered the possibility that the power model used to model our variogram could be replaced by an exponential model. Namely, we re-performed the LOOCV with an exponential variogram model (Figure 11). We were hoping for somewhat more accurate results but this fell short as well, ensuring that it was not an issue with the model. We think that this lack of accuracy is because California is a tough state to model, and lead to our appreciation of the complexities in modeling seemingly simple quantities like temperature.

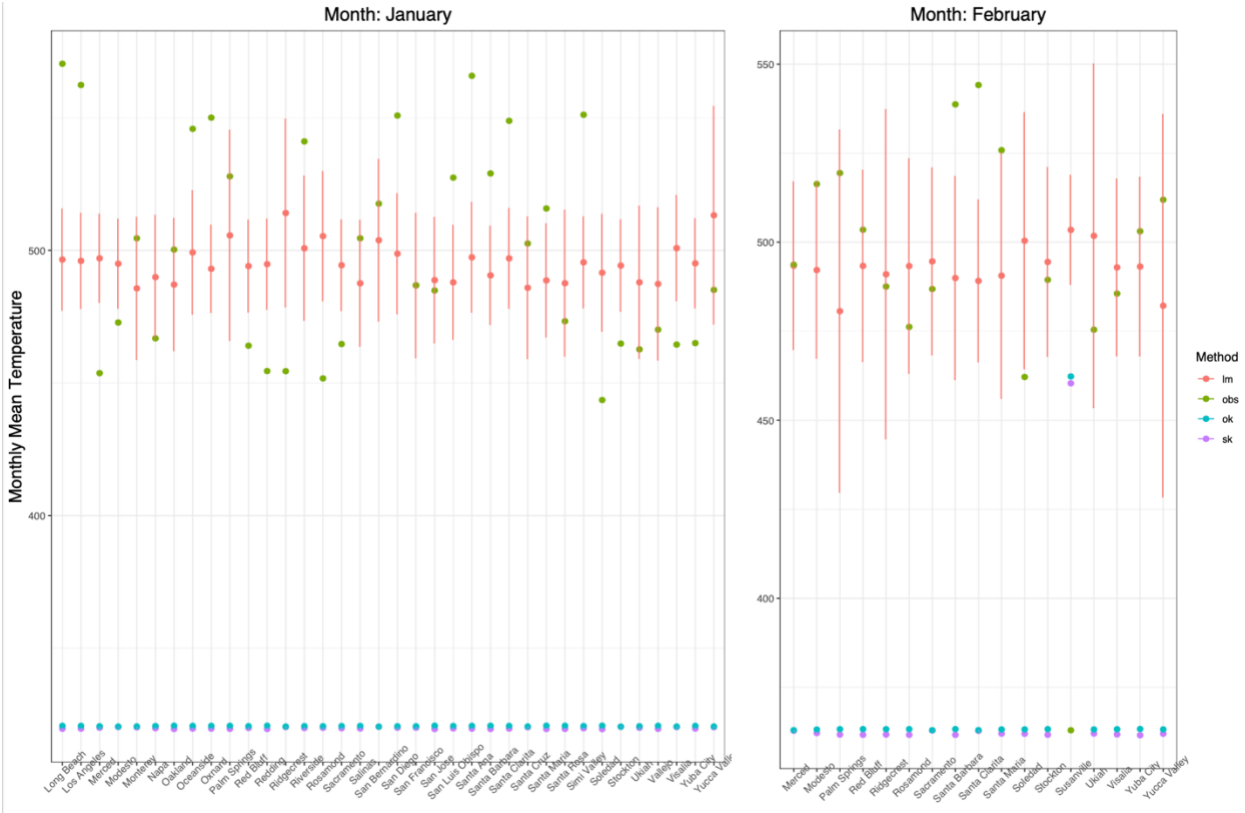


Figure 11: Attempt with Exponential Model

After evaluating our estimates, we wanted to get a better idea of what the data looked like and how it was distributed so that we could get insights into the results we found. We explored the set in detail, the contents of which can be found in the additional attached code

("dataexplor.R"). Although not every question we had could be answered, we were grateful to get some explanation as to why kriging had estimated the observed values so poorly. The first thing we looked at was a plot of the frequency of each temperature value (Figure 12). This was to see what general temperature tends to stay around and how much spread we are seeing.

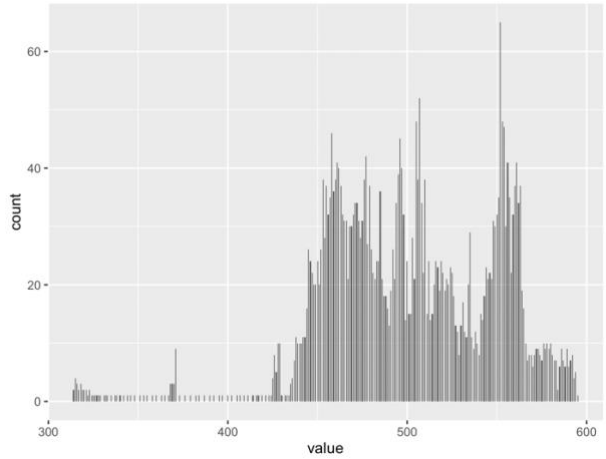


Figure 12: Temperature Value Distribution (1)

Next, we color-coded the plot by city (Figure 13) to get an idea of how they compare to one another. This turned out to be quite messy and hard to decipher anything from so we went ahead and plotted an ANOVA box plot (Figure 14) to get a better look at variability on a city-wide basis and how they compare to one another. We can see that Stockton falls significantly below all other cities with significant variability, we assumed this would happen given that Adith is from California and knows from personal experience what the weather is usually like in a handful of cities. Palm Springs also had a large amount of variability in daily average temperature, whereas certain cities like Santa Cruz seemed to follow a more generally applicable overall daily average temperature. We knew that this was going to cause issues in estimation but were not nearly prepared for how poorly ordinary and standard kriging did at estimating the daily average temperature at the "missing city."

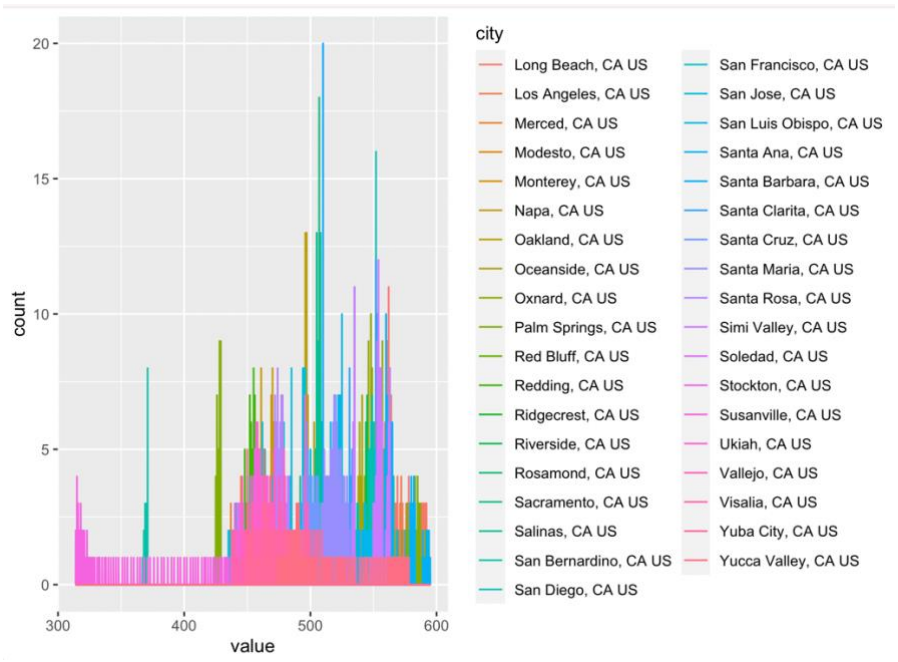


Figure 13: Temperature Value Distribution (2)

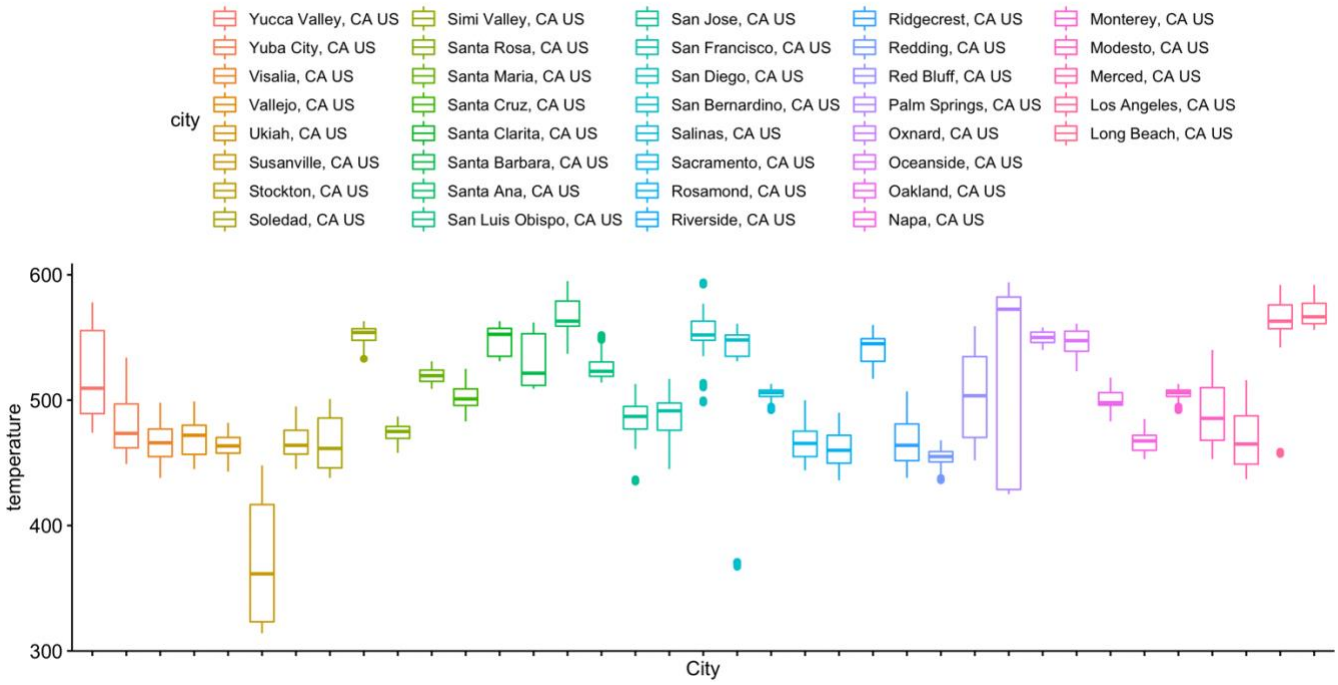


Figure 14: ANOVA Box Plot

This project was a great opportunity for us to learn more about concepts that we had covered in class but had not done in homework, and learn new approaches like GCD. California

was extremely difficult to model due to its range in elevation, anisotropic nature, weather dips, and much more. California is a very large state and most of its size comes from vertical distance meaning very different weather patterns throughout. The distance from the coast can have an impact on the temperature and wind speed. The lack of consistent precipitation activity in the state makes a large impact on its unpredictability as well. More recently, extreme natural disasters have made headway in frequency in the state but the issues underlying why these droughts, high winds, and wildfires are happening have always been there; only to be strengthened with growing climate change. Some factors of this also impact the weather in the state: placement over the Pacific and North American tectonic plates, fluctuating shorelines, mudslides, and more. Many factors were working against us in our data not only from a standpoint of the area we chose but also an extensive amount of missing values that forced us to drop a month entirely in our model. Overall, modeling California's montly average temperature by city was a great educational experience for both of us and we are glad that we chose to tackle such a difficult topic to learn more about what causes it and what it looks like when attempting to model it.

```r
options(noaakey = "FsbwNbQyoIOHLnPdaNSZJQZquVXxakKu")
library(rnoaa)
library(tidyverse)



get_data <- function(x){
  data <- ncdc(datasetid='NORMAL_DLY', limit=100, datatypeid='dly-tavg-normal', startdate = '2010-01-01',
 enddate = '2010-03-31',  add_units=TRUE, locationid=x$id)
  data$data %>% as_tibble() %>% mutate(city=x$name, cityid=x$id)
}

#collect data on the mean daily temperature for all cities in California with a NOAA temperature monitor.
datafile <- "~/Desktop/school/geostats/data_final.rds"

if (file.exists(datafile)){
  data <- readRDS(datafile)
} else{
  cities <- ncdc_locs(locationcategoryid = "CITY", sortfield = "name", sortorder='desc')
  ca.cities <- cities$data %>% as_tibble() %>% filter(grepl("CA US", name))
  data <- lapply(1:dim(ca.cities)[1], function(x) get_data(ca.cities[x,]))
  data <- do.call(rbind, data) %>% as_tibble()
}

#collect latitude, longitude data on the stations within the cities
latlong <- readRDS("latlong.rds")

#merge data types
df <- latlong %>% left_join(data, by=c("city"="city")) %>% as_tibble()

df <- df %>% select(city, lat, long, date,value) %>% mutate(lat = as.numeric(lat),long = as.numeric(long))

#let us use a location that is somewhat far from all California places. Let us choose New York City, New York
ref.loc <- c(40.7128, 74.0060)

#let us compute the great circle distance because we know that a 1 latitude change is not always the same
 amount of distance
source("greatcircledist.r")

gcd <- unlist(apply(df, 1, function(x) greatcircledist(ref.loc[1],as.numeric(x[2]),
 ref.loc[2],as.numeric(x[3]))))
df$gcd <- gcd
#these values are really big, let us log base 10 transform them
df <- df %>% select(-c(lat, long)) %>% mutate(gcd = log(gcd, base=10)) %>% mutate(value = as.numeric(value))

library(geoR)

#let us now fit some variograms
#we are interested in one month's worth of data at a time, so let us split by month
#and summarize by taking the mean at each month
df$month <- str_sub(df$date, 6, 7)
df <- df %>% group_by(month, city) %>% mutate(mean=mean(value)) %>% ungroup()

gdat <- df %>% select(-c(date, value)) %>% distinct()
split.data <- split(gdat, gdat$month)

#let us look at each month separately (change the index on split.data) for now (Think about how to best
 display data for all 3 months)
summ.df <- split.data[[1]] %>% select(-city) %>% distinct() %>% select(-month) %>% mutate(dummy=0)
gdata <- as.geodata(summ.df, coord.col=c(1,3), data.col = 2)

#raw variogram
vcloud = variog(gdata, option = "cloud")
plot(vcloud, main = "Variogram Cloud")
```

```r
#let us now look at an experimental variogram
max.dist = vcloud$max.dist/2
#Now, Kitanidis' book recommends 3-6 bins, and Dr. Miller recommends 5-6 bins.
#But, for sake of being thorough, let us look at the variograms when we iterate from 3 to 6 bins inclusive.
nbins <- 3:6
par(mfrow=c(2,2))
for (i in nbins){
  uvec = i
  vexp = variog(gdata, option = "bin",max.dist=max.dist, uvec = uvec)

  plot(vexp, main=paste0("Bins: ", i), pch=19, cex=1.5, col="blue")
}
#I like 6 bins

#what variogram model to choose? Clearly, this is nonstationary! (what is intrinsic nonstationary?)
#the two variogram models that model intrinsic nonstationarity are the linear and power model
#I think the power will do better, the variogram is clearly not linear


#let us visually play with the parameters in the power model to find good inital guesses for our parameters of
 choice
power <- function(x, b, c){
  x^c * b
}
max.dist = vcloud$max.dist/2
vexp = variog(gdata, option = "bin",max.dist=max.dist, uvec = 6)
xvec <- seq(0, max.dist)

plot(vexp)
lines(xvec, power(xvec, b=1.2, c=1.8))

#we can fit the power model parameters b and c
#(0 < c < 2 by definition)
#Restricted Maximum Likelihood and Weighted Least Squares)

b.guess <- 1.2
c.guess <- 1.8

weightsoption = "npairs"
initialguess = c(1.2, 1.8)
covmodel = "power"
vario.WLS = variofit(vexp, ini.cov.pars = initialguess, weights = weightsoption, cov.model = covmodel)

#the WLS estimate for b is 0.5755  and c is 1.9627

plot(vexp)
lines(xvec, power(xvec, 1.2, 1.8), col="red")
lines(vario.WLS, col="blue")
legend("topleft",
       legend = c("Visual Fit",  "WLS"),
       col = c("red", "blue"),
       pch = c(19,19),
       bty = "n",
       pt.cex = 1.1,
       cex = 1.2,
       text.col = "black",
       horiz = F ,
       inset = c(0.1, 0.1))



#Now, for estimation we will do a leave one out approach.
#Namely, we will leave out city j, fit the variogram and do the kriging/OLS to estimate monthly temp at city j
```

```
#loc is the value for which we want to estimate the response for
#data is the data frame
idw <- function(loc, data){
  vals <- matrix(data$mean, ncol=1)
  p <- 2
  distance <- loc - data$gcd
  pwr <- distance ^ (-p)
  total.sum <- sum(pwr)
  lambdas <- pwr/total.sum
  lambdas <- matrix(lambdas, nrow=1)
  lambdas %*% vals
}

cov.model <- "power"
get_est <- function(tmp.df, j) {
  temp <- tmp.df[-j, ]

  x <- tmp.df[j, 2]
  city.guess <- tmp.df[j, 1]
  z <- tmp.df[j, 4]

  summ.df <- temp %>% select(-city) %>% distinct() %>% select(-month) %>% mutate(dummy=0)
  gdata <- as.geodata(summ.df, coord.col=c(1,3), data.col = 2)

  vcloud = variog(gdata, option = "cloud")
  max.dist = vcloud$max.dist/2
  vexp = variog(gdata, option = "bin", max.dist=max.dist, uvec = 6)
  vario.WLS = variofit(vexp, ini.cov.pars = initialguess, weights = weightsoption, cov.model = cov.model)

  #linear regression
  m <- lm(mean ~ gcd, temp)

  #fit, lower, upper 95% CI
  lm.guess <- predict.lm(m, x, interval="confidence") %>% unname()

  #idw(x %>% unname() %>% unlist(), temp)

  #kriging vars are 0 which makes sense because power model at distance 0 gives covariance of 0; nugget is
 estimated to be about 0 by variogram fitting (as we can see w our eyes too)
  sk.guess <- krige.conv(gdata, locations = data.frame(gcd = x %>% unname()  %>% unlist(), dummy=0),krige =
 krige.control(type.krige = "SK", cov.model=cov.model, beta = mean(temp$gcd), obj.model = vario.WLS) )
 $predict[[1]]
  ok.guess <- krige.conv(gdata, locations = data.frame(gcd = x %>% unname()  %>% unlist(), dummy=0),krige =
 krige.control(type.krige = "OK", cov.model=cov.model, obj.model = vario.WLS) )$predict[[1]]

  c(lm.guess[1], lm.guess[2], lm.guess[3], sk.guess, ok.guess, z %>% unname() %>% unlist())

}


rmse <- list()
res <- list()

for (k in 1:2){

  tmp.df <- split.data[[k]]
  n.cities <- dim(tmp.df)[1]

  if (k==1){
    #for january
    #at position six,Error in solve.default(Vcov, trend.d) :
    #Lapack routine dgesv: system is exactly singular: U[36,36] = 0
    indexes <- c(1:n.cities)[-6]
```

```
  } else{
    indexes <- c(1:n.cities)
  }

  results <- do.call(rbind, lapply(indexes, function(j) get_est(tmp.df, j) )) %>% as_tibble()

  colnames(results) <- c("lm", "lm.low", "lm.upper", "sk", "ok", "obs")

  if (k==1){
    results$city <- tmp.df$city[-6]
    results$month <- tmp.df$month[-6]
  } else{
    results$city <- tmp.df$city
    results$month <- tmp.df$month
  }

  r.long <- results %>% pivot_longer(-c(lm.low, lm.upper, month, city), names_to="type", values_to="val")
  r.long <- r.long %>% mutate(city = gsub(", CA US", "", city))

  r.long <- r.long %>% mutate(lm.low = ifelse(type=="lm", lm.low, val), lm.upper = ifelse(type=="lm",
lm.upper, val))

  plt <- ggplot(r.long, aes(x=city, y=val, color=type, group=type)) + geom_point(size=2) + theme_bw()  +
theme(axis.text.x = element_text(angle = 45))
  plt <- plt + theme(axis.title.y = element_text(size=15), axis.title.x = element_blank()) + ylab("Monthly
Mean Temperature")
  plt <- plt + geom_errorbar(aes(ymin=lm.low, ymax=lm.upper), width=.2, position=position_dodge(0.05)) +
labs(color="Method")


  #RMSE for each method
  n <- (dim(results)[1])
  lm.rmse <- sum ( (results$lm - results$obs)^2 )  / n
  ok.rmse <- sum ( (results$ok - results$obs)^2 )  / n
  sk.rmse <- sum ( (results$sk - results$obs)^2 )  / n

  if (k==1){
    plt <- plt + ggtitle("Month: January") + theme(plot.title=element_text(hjust=0.5, size=16)) +
theme(legend.position="none")
    month.rmse <- data.frame(lm=lm.rmse, ok=ok.rmse, sk=sk.rmse, month="January")
  }

  if (k==2){
    plt <- plt + ggtitle("Month: February") + theme(plot.title=element_text(hjust=0.5, size=16))  + ylab("")
    month.rmse <- data.frame(lm=lm.rmse, ok=ok.rmse, sk=sk.rmse, month="February")
  }

  rmse[[k]] <- month.rmse
  res[[k]] <- plt
}


#rmse comparisons
t <- do.call(rbind, rmse)
t <- t %>% pivot_longer(-c(month), names_to="method", values_to="val")
p <- ggplot(t, aes(x=month, y=val, color=method, group=method)) + geom_point(size=3) + theme_bw() +
 geom_line()
loocv.rmse <- p + xlab("") + ylab("RMSE in Leave One Out Setup")  + theme(axis.title=element_text(size=16),
 axis.text=element_text(size=14), legend.text=element_text(size=14), legend.title=element_text(size=16))
ggsave(loocv.rmse, filename="plots/loocv.rmse.pdf", width=8, height=9,units="in", device=cairo_pdf )

#dimensions 8 x 6
#much less data in March! So we won't krig with it!
par(mfrow=c(1,3))
```

```
 hist(split.data[[1]]$gcd, xlab="GCD", main="January")
 hist(split.data[[2]]$gcd, xlab="GCD", main="February")
 hist(split.data[[3]]$gcd, xlab = "GCD", main="March")


 #dimensions 21 x 10
library(cowplot)
fin.res <- plot_grid(res[[1]], res[[2]], nrow=1, rel_widths=c(1.75, 1.25))
ggsave(fin.res, filename="plots/fin.res.pdf", width=15, height=10,units="in", device=cairo_pdf )


#what if we make an approximation and say that the variogram model is stationary (say exponential in nature)
rmse <- list()
res <- list()

cov.model <- "exponential"
for (k in 1:2){

  tmp.df <- split.data[[k]]
  n.cities <- dim(tmp.df)[1]

  if (k==1){
    #for january
    #at position six,Error in solve.default(Vcov, trend.d) :
    #Lapack routine dgesv: system is exactly singular: U[36,36] = 0
    indexes <- c(1:n.cities)[-6]
  } else{
    indexes <- c(1:n.cities)
  }

  results <- do.call(rbind, lapply(indexes, function(j) get_est(tmp.df, j) )) %>% as_tibble()

  colnames(results) <- c("lm", "lm.low", "lm.upper", "sk", "ok", "obs")

  if (k==1){
    results$city <- tmp.df$city[-6]
    results$month <- tmp.df$month[-6]
  } else{
    results$city <- tmp.df$city
    results$month <- tmp.df$month
  }

  r.long <- results %>% pivot_longer(-c(lm.low, lm.upper, month, city), names_to="type", values_to="val")
  r.long <- r.long %>% mutate(city = gsub(", CA US", "", city))

  r.long <- r.long %>% mutate(lm.low = ifelse(type=="lm", lm.low, val), lm.upper = ifelse(type=="lm",
lm.upper, val))

  plt <- ggplot(r.long, aes(x=city, y=val, color=type, group=type)) + geom_point(size=2) + theme_bw()  +
theme(axis.text.x = element_text(angle = 45))
  plt <- plt + theme(axis.title.y = element_text(size=15), axis.title.x = element_blank()) + ylab("Monthly
Mean Temperature")
  plt <- plt + geom_errorbar(aes(ymin=lm.low, ymax=lm.upper), width=.2, position=position_dodge(0.05)) +
labs(color="Method")


  #RMSE for each method
  n <- (dim(results)[1])
  lm.rmse <- sum ( (results$lm - results$obs)^2 )  / n
  ok.rmse <- sum ( (results$ok - results$obs)^2 )  / n
  sk.rmse <- sum ( (results$sk - results$obs)^2 )  / n

  if (k==1){
    plt <- plt + ggtitle("Month: January") + theme(plot.title=element_text(hjust=0.5, size=16)) +
```

```
 theme(legend.position="none")
    month.rmse <- data.frame(lm=lm.rmse, ok=ok.rmse, sk=sk.rmse, month="January")
  }

  if (k==2){
    plt <- plt + ggtitle("Month: February") + theme(plot.title=element_text(hjust=0.5, size=16))  + ylab("")
    month.rmse <- data.frame(lm=lm.rmse, ok=ok.rmse, sk=sk.rmse, month="February")
  }

  rmse[[k]] <- month.rmse
  res[[k]] <- plt
}

library(cowplot)
fin.res <- plot_grid(res[[1]], res[[2]], nrow=1, rel_widths=c(1.75, 1.25))
ggsave(fin.res, filename="plots/exponential.fin.res.pdf", width=15, height=10,units="in", device=cairo_pdf )
```

```r
#DATA EXPLOR
library(tidyverse)
data <- readRDS('data_final.rds')

#see how many cities we are looking at
length(unique(data$cityid)

#we have 37 cities we are looking at here are all of our cities
unique(data$city)

#here is the number of weather stations we are looking at
length(unique(data$station))

#let's see what in general the weather tends to stay around and how frequently each block occurs
ggplot(data) + geom_histogram(mapping = aes(x = value), binwidth = 0.5)

#let's see how each city maps compared to eachother
ggplot(data, mapping = aes(x = value, colour = city)) + geom_freqpoly(binwidth = 0.1)

#ANOVA box plot to look at variability of temperature per city
ggboxplot(data, x = "city", y = "value", color = "city", ylab = "temperature", xlab = "City") +
 theme(axis.text.x = element_blank())
#looking at this it is clear then

#let's check to see that we have all consisteint coverage for each city (we want to see that every single one
 has the same amount of values)
vtree(data, "city")
#clearly, while we like what we see, this is not perfect since there are a few cities in which there were days
 that tempeture data was either not collected or not uploaded, these cities are: Yucca Valley, susanville,
 soledad, and red bluff

#now let's see for each city within that just how many stations they have, and how it is distributed.
vtree(data, c("city", "station"))

#this tree is so large that it is basically unreadable so let's look at the ones that peaked interest on an
 individual level...
#let's seperate eachh city and learn a bit about them
Yucca_Valley=data %>% filter(data$city =='Yucca Valley, CA US')
Soledad=data %>% filter(data$city =='Soledad, CA US')
Santa_Ana=data %>% filter(data$city =='Santa Ana, CA US')
Sacramento=data %>% filter(data$city =='Sacramento, CA US')
Oxnard=data %>% filter(data$city =='Oxnard, CA US')
Los_Angeles=data %>% filter(data$city =='Los Angeles, CA US')
Yuba_City=data %>% filter(data$city =='Yuba City, CA US')
Simi_Valley=data %>% filter(data$city =='Simi Valley, CA US')
San_Luis=data %>% filter(data$city =='San Luis Obispo, CA US')
Rosamond=data %>% filter(data$city =='Rosamond, CA US')
Oceanside=data %>% filter(data$city =='Oceanside, CA US')
Long_Beach=data %>% filter(data$city =='Long Beach, CA US')
Visalia=data %>% filter(data$city =='Visalia, CA US')
Santa_Rosa=data %>% filter(data$city =='Santa Rosa, CA US')
San_Jose=data %>% filter(data$city =='San Jose, CA US')
Riverside=data %>% filter(data$city =='Riverside, CA US')
Oakland=data %>% filter(data$city =='Oakland, CA US')
Vallejo=data %>% filter(data$city =='Vallejo, CA US')
Santa_Maria=data %>% filter(data$city =='Santa Maria, CA US')
San_Francisco=data %>% filter(data$city =='San Francisco, CA US')
Ridgecrest=data %>% filter(data$city =='Ridgecrest, CA US')
Napa=data %>% filter(data$city =='Napa, CA US')
Ukiah=data %>% filter(data$city =='Ukiah, CA US')
Santa_Cruz=data %>% filter(data$city =='Santa Cruz, CA US')
San_Diego=data %>% filter(data$city =='San Diego, CA US')
Redding=data %>% filter(data$city =='Redding, CA US')
Monterey=data %>% filter(data$city =='Monterey, CA US')
Susanville=data %>% filter(data$city =='Susanville, CA US')
```

```
Santa_Clarita=data %>% filter(data$city =='Santa Clarita, CA US')
San_Bernardino=data %>% filter(data$city =='San Bernardino, CA US')
Red_Bluff=data %>% filter(data$city =='Red Bluff, CA US')
Modesto=data %>% filter(data$city =='Modesto, CA US')
Stockton=data %>% filter(data$city =='Stockton, CA US')
Santa_Barbara=data %>% filter(data$city =='Santa Barbara, CA US')
Salinas=data %>% filter(data$city =='Salinas, CA US')
Palm_Springs=data %>% filter(data$city =='Palm Springs, CA US')
Merced=data %>% filter(data$city =='Merced, CA US')
#looked up count of unique stations each

#let's see what in general the weather tends to stay around and how frequently each block occurs
ggplot(data) + geom_histogram(mapping = aes(x = data$value), binwidth = 0.5)
```