CVEN 6833
HW 2: Due November 17, 2020
Nathan Bonham

P1. Repeat problem 8 from HW1 with a Bayesian Hierarchical Spatial Model (see Verdin et al. 2015, for tips)

[In the Bayesian models, plot the posterior histogram/PDF of the parameters; spatial maps of posterior mean and standard error]

a. Fit a GLM (from P3 of HW 1, use Gamma with log link function)

```
> summary(glmfit)

Call:
glm(formula = Pm ~ ., family = Gamma(link = "log"), data = X)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.46974  -0.16585   0.01506   0.13873   0.38964

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.144e+01  7.731e-01  14.800  < 2e-16 ***
Lon          5.831e-02  7.015e-03   8.313 5.36e-12 ***
Lat         -4.394e-02  8.976e-03  -4.895 6.20e-06 ***
Elev         5.843e-06  5.004e-05   0.117    0.907
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 0.04053641)

    Null deviance: 7.6003  on 72  degrees of freedom
Residual deviance: 2.9313  on 69  degrees of freedom
AIC: 501.47

Number of Fisher Scoring iterations: 5
```

From the model summary, longitude and latitude are statistically significant whereas elevation adds minimal skill. The kriging is fitted to the GLM residuals.

b. Calculate a semivariogram to obtain plausible ranges and initial values for sigma squared, tau squared, and phi.

Performed using the variog() function from the geoR package. A plausible range for phi (effective range) and initial estimates for the sill and nugget are shown below.

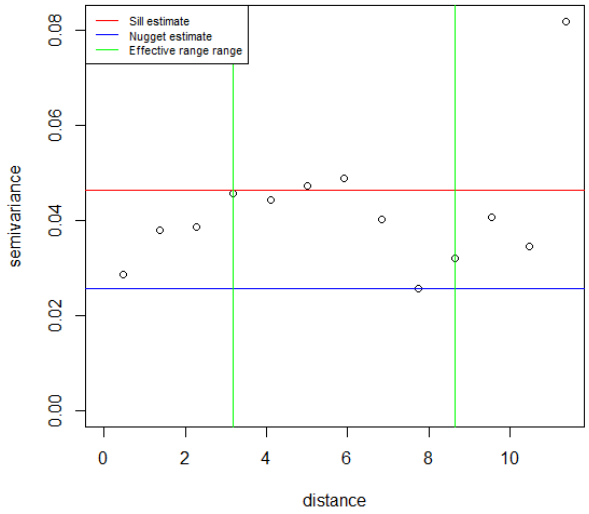

Figure 1: Phi (effective range) range chosen as the 25th and 75th percentile of distance. Initial value for nugget and sill chosen as the minimum and 90th percentile semivariance, respectively.

Beta (regression coefficients) are assigned a flat prior, phi is uniform across it's plausible range, and both sigma squared and tau squared follows inverse gamma with shape = 2 and scale = 1. This shape and scale were chosen because to represent greater uncertainty in the prior.

c. Markov Chain Monte Carlo to simulate posteriors with spLM() and spRecover() functions:

```
# Perform Monte Carlo Markov Chain Analysis
bat.fm = spLM(Pm~., data=X, coords=as.matrix(X[,1:2]), priors=priors, tuning=tuning, starting=starting,
              cov.model = "exponential", n.samples = n.samples, verbose = FALSE, n.report = 50)
```

Note that the semivariogram function was chosen as exponential based on visual inspection of figure 1. Starting values for sill and effective range are shown in figure 1, and the starting value for phi taken as the median distance.

```
# burn-in samples
bat.fm = spRecover(bat.fm, start=((1/3)*n.samples)+1, thin=1, verbose=T)
  #spRecover: function for recovering regression coefficients and spatial
  # random effects from spLM using composition sampling
```

d. Plot posterior PDFs of parameters

Beta parameters (intercept and coefficients on longitude, latitude, and density)

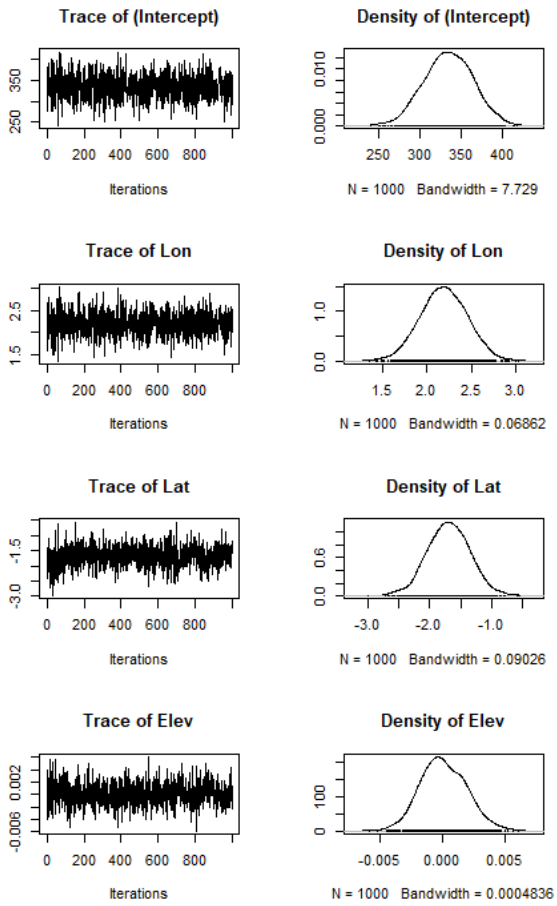

Figure 2: Posterior distributions of intercept and the regression coefficients for longitude, latitude, and elevation in the Bayesian hierarchical model. The left column shows the coefficient values as a function of iteration from Markov Chain Monte Carlo analysis. The right column shows the pdf of each parameter. Note that the prior distribution is flat, and the posteriors appear normal. Density is centered near zero with small variance, suggesting elevation is insignificant.

Theta parameters (semivariogram parameters)

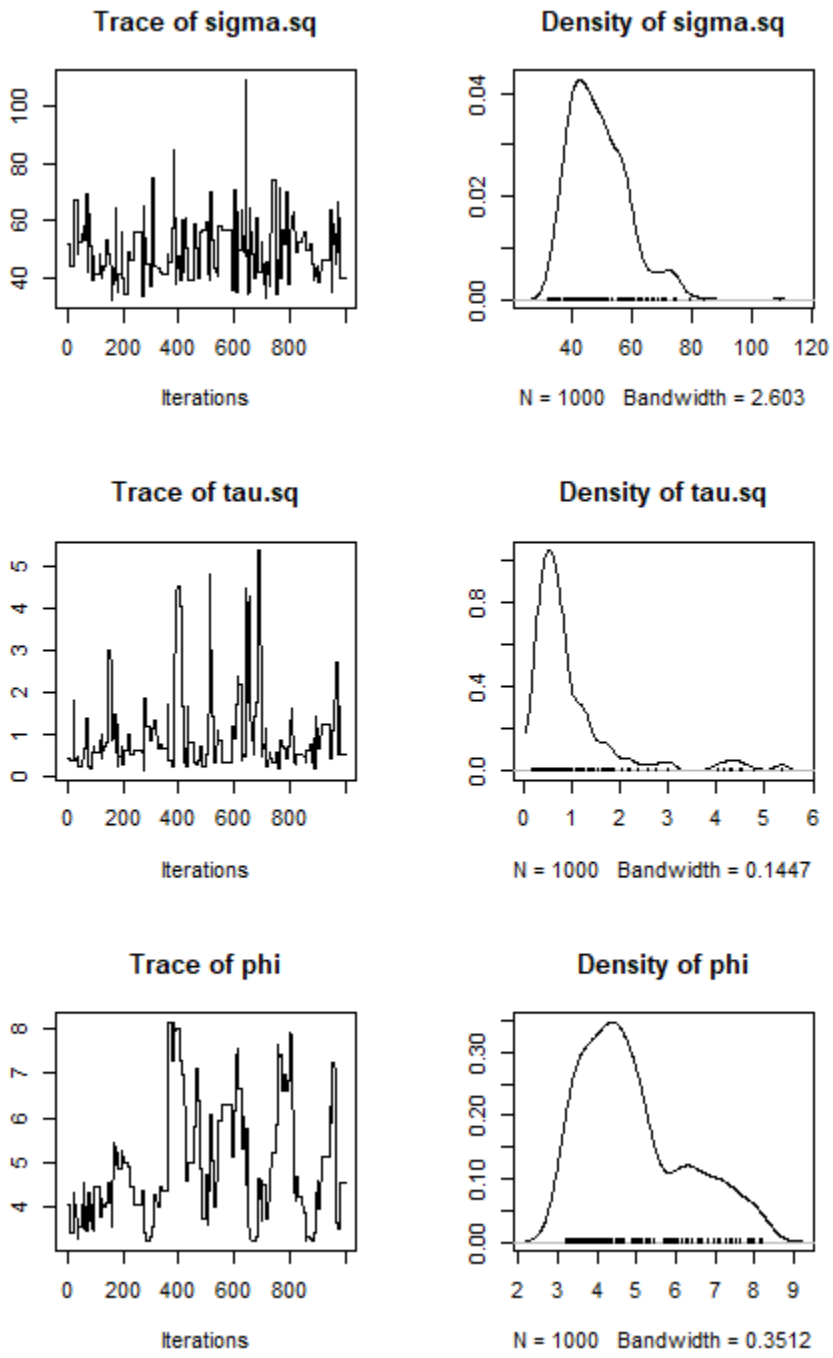

Figure 3: Posterior distributions for theta parameters in the Bayesian hierarchical model. Note that sigma squared around 45. Tau squared is about 0.5, and phi is centered near 5 with a long right tail. Tau and phi values are similar to estimates from figure 1.

e. Spatial plot of predictions and standard error

See problem three or the appendix for the code that predicts form the posterior distributions. The results are shown below.

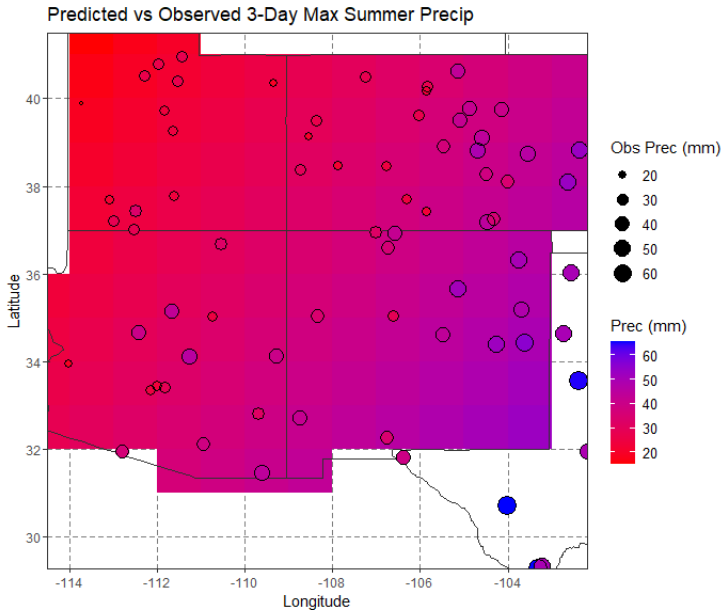

*Figure 4:*Predicted vs observed max summer precipitation for Bayesian spatial hierarchy model. The grid shows predicted precipitation at each new location, whereas the points show observed precipitation. The grid and points use the same color scale. The largest residuals (as indicated by color differences) are observed in central AZ.

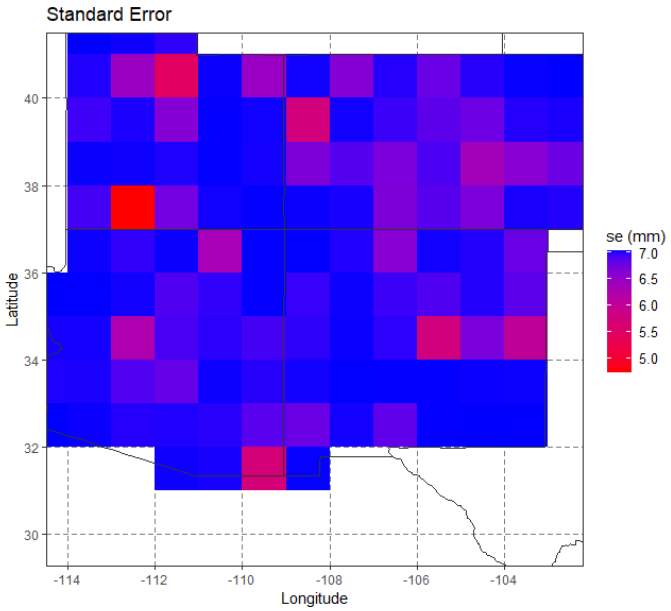

Figure 5: Standard error for Bayesian spatial hierarchy model. Standard error ranges from 4.5 to 7.0 mm.

Conclusions (Bayesian spatial hierarchy)

- The posterior distributions converge nicely.
- Similar to regression models in HW1, the model predicts increasing precipitation in the southeast direction.
- Standard error appears to be larger in mountainous areas (Wasatch front and Rockies, specifically)

P2. Logistic Regression

Compute the 75th percentile of the average summer 3-day maximum precipitation across all the stations and using this as the threshold categorize the annual precipitation at each station to a binary variable – 0 if annual precipitation less than the threshold, 1, otherwise.

Using quantile(Pm, 0.75) command, the 75th percentile precipitation is 44.5 mm (assigned to variable Q75). Observations were assigned 1 if above 44.5 mm and 0 otherwise.

i. Fit a 'best' GLM (i.e. logistic regression) with the appropriate link function using one of the objective functions. Test the model goodness using ANOVA

Binomial GLMs with logit, probit, and cauchit link funcitons were tested using elevation, lat, and lon as predictors. The best model according to AIC used the cauchit link function. Compared to probit and logit, the cuachit pdf has less variance and larger range than probit and logit (see figure 1 in this reference: https://arxiv.org/pdf/1502.04742.pdf)

From ANOVA, longitude reduces deviance the most followed by latitude. Elevation adds only slight improvements in model fit.

```
Analysis of Deviance Table

Model: binomial, link: cauchit

Response: Pm

Terms added sequentially (first to last)


     Df Deviance Resid. Df Resid. Dev
NULL                    72     81.203
Lon   1   48.283        71     32.920
Lat   1    5.970        70     26.950
Elev  1    3.698        69     23.252
```

ii. Estimate the function on the grid and plot the surface. Also plot the standard error.

Predictions were made using the predict.glm() command with new elevation, lat, and lon locations and type=response. The surface is plotted below:

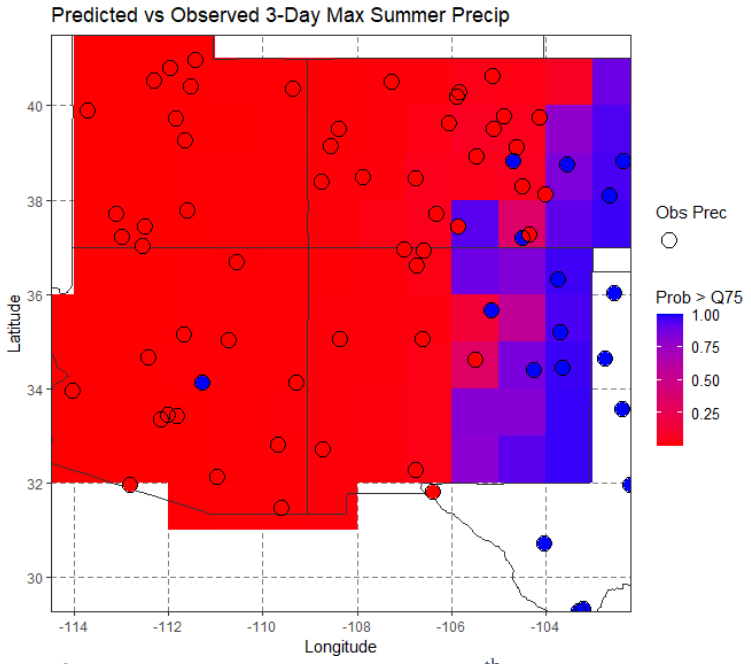

Figure 6: Probability of precipitation greater than the 75th percentile with GLM. The predictions are shown as the raster, whereas observed values are shown as points. If above 0.5 are classified as true, then no points are misclassified in UT, one point is misclassified in AZ, and multiple points are misclassified in CO and NM towards the east.

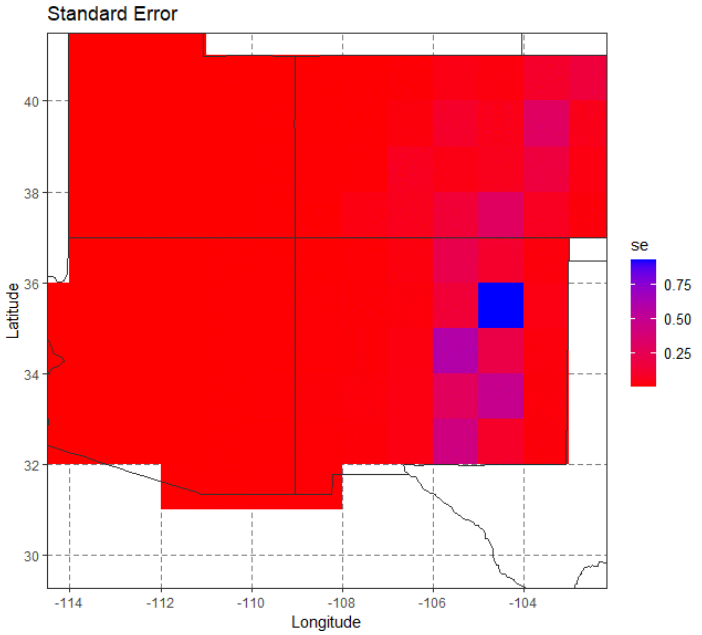

Figure 7: Standard error for the GLM model. The standard error is small where predictions are very near 0 and 1 and larger when predictions are closer to 0.5.

iii. Repeat (i) and (ii) with Local GLM

Local GLM models were tested with varying neighborhood sized and polynomial order 1 and 2, where the GLM is the same as in part 1 above. The best model according to GCV is order 2 and alpha = 0.6.

Using the F-test, I tested the null hypothesis that a linear regression model best fits the data, and the alternate hypothesis is that local polynomial is better. The results are below:

```
> loc_Ftest(Pm_bin, Pmhat, X, local_mod)
[1] "F-test:"
[1] "Reject the Null because F(local poly) = 5.46 > 2.23 = F(linear model)."
```

The conclusion is that the local polynomial significantly outperforms a linear model.

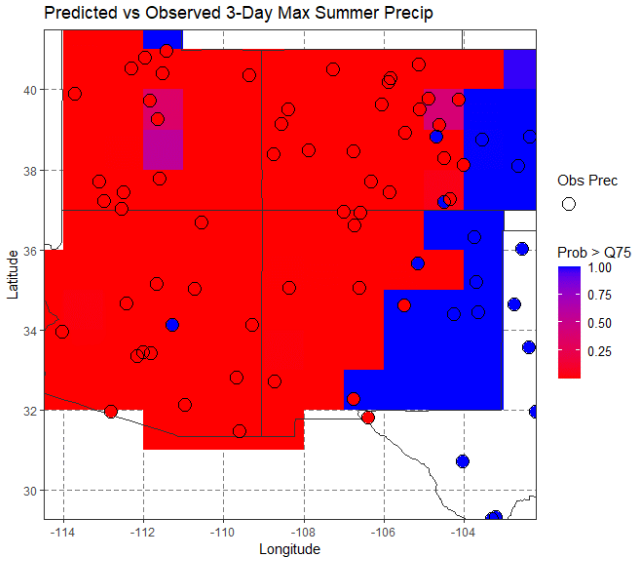

Figure 8: Local logistic regression model predictions are shown as raster while observations are shown as points, the same as Figure 6 above. The local GLM predicts much higher probabilities (closer to 1) across eastern CO and NM. Further, this model predicts slightly higher probabilities in central UT where the elevation is higher in the Manti-La Sal National Forest.

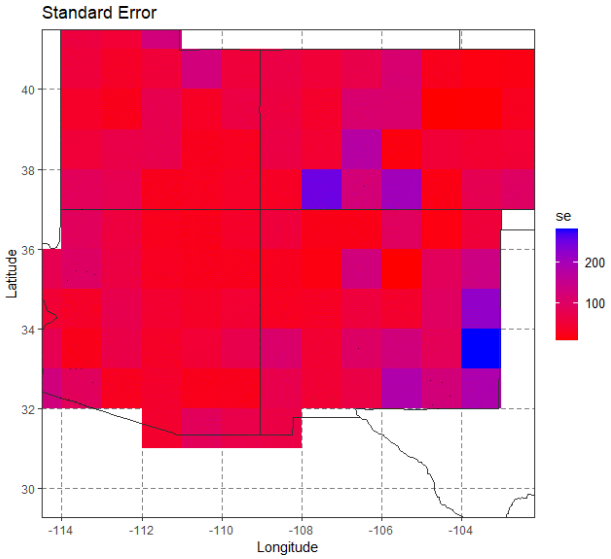

Figure 9: The local logistic model demonstrates larger standard error across the entire domain than does the normal logistic regression. Greatest standard error is in the CO Rockies and eastern NM.

Conclusions (GLM logistic regression and local GLM logistic regression)

- The non-local model predicts near 1 only at the eastern edge of domain. This aligns with the spatial patterns predicted with the regression models in HW1 and problem 1.
- The local model assigns higher probabilities in the Wasatch front, which demonstrates how the 'local' nature of this model.
- Similar to what I observed in HW1, local polynomials result in very high standard error.

P3: Bayesian Logistic Regression

I performed spatial logistic regression using binomial for the y distribution and logit as the link function. I utilized the spBayes package like in Problem 1.
The resulting posterior distributions are shown below:

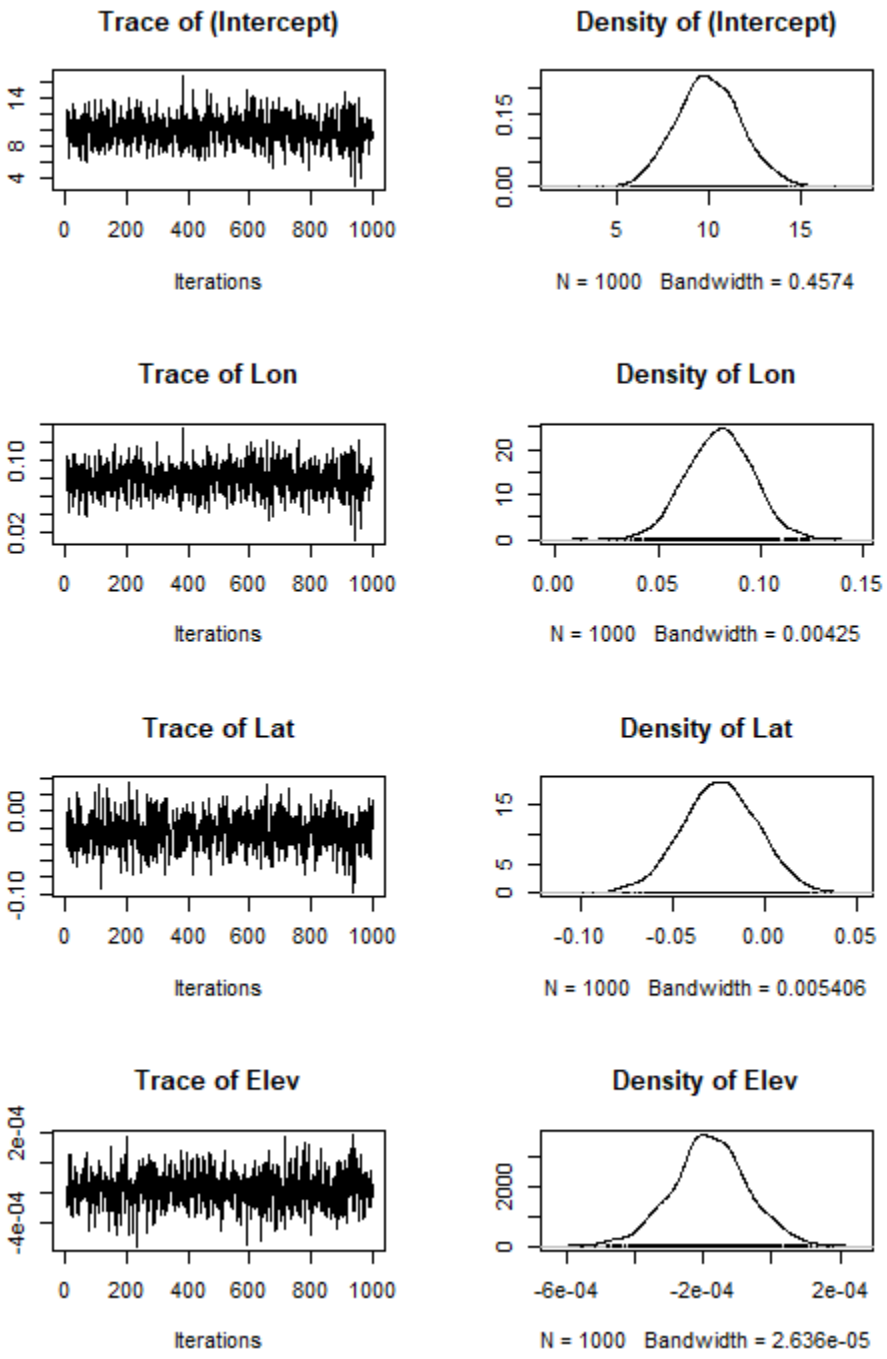

Figure 10: Posterior distributions of logistic regression. The intercept dominates the model (near 10) while latitude, longitude, and elevation are near 0.

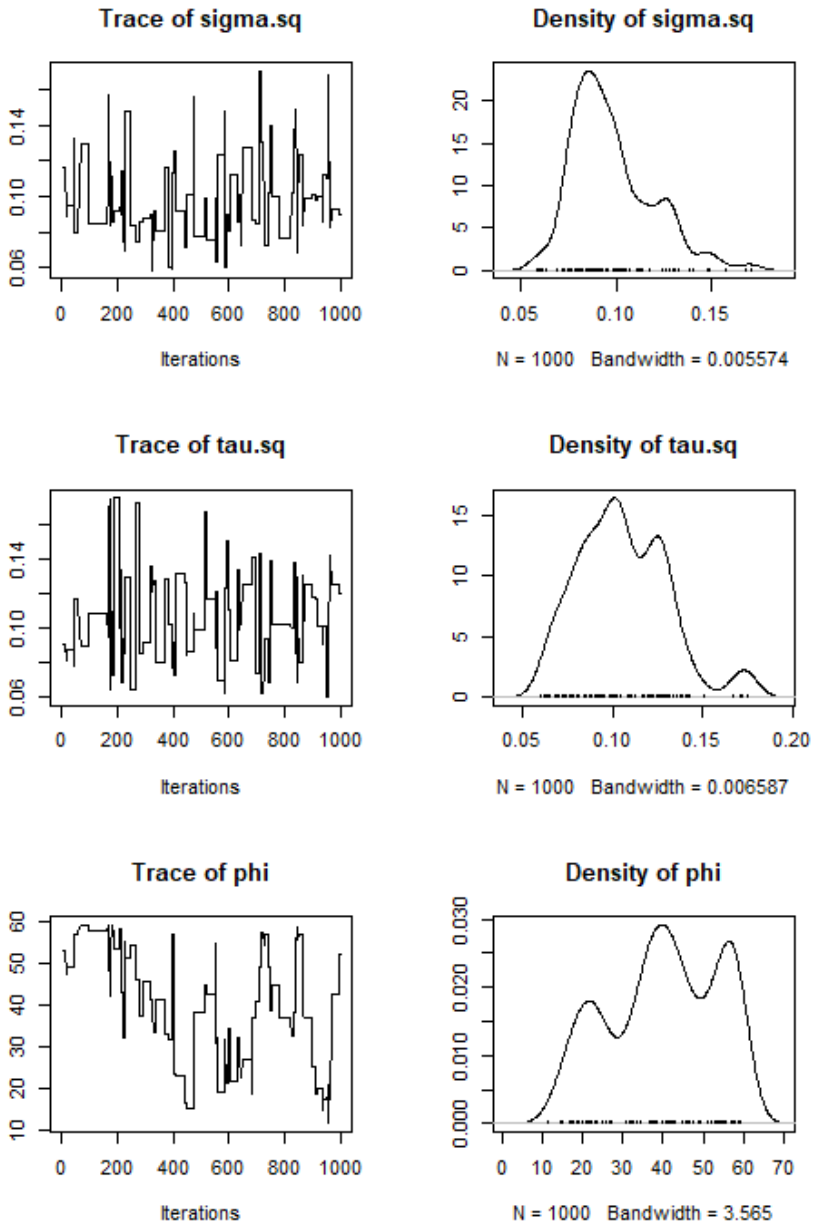

Figure 11: The posterior distributions of the spatial parameters for the logistic regression show multiple peaks. This is unlike problem 1.

The probability of precipitation greater than the 75th percentile is modeled for each point in the posterior distributions, and the mean is taken as the final prediction. The result is shown in the plot below, and the code for predicting follows.

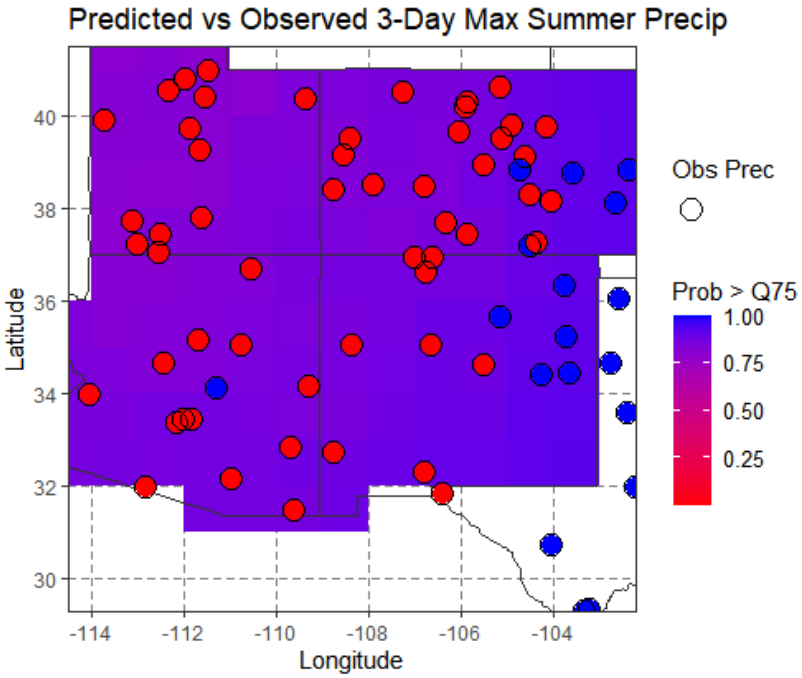

Figure 12: The Bayesian logistic model is dominated by the large intercept, which causes all probabilities to be high (near one). Nevertheless, the probability still increases in the southeast direction like with normal (non Bayesian) logistic regression.

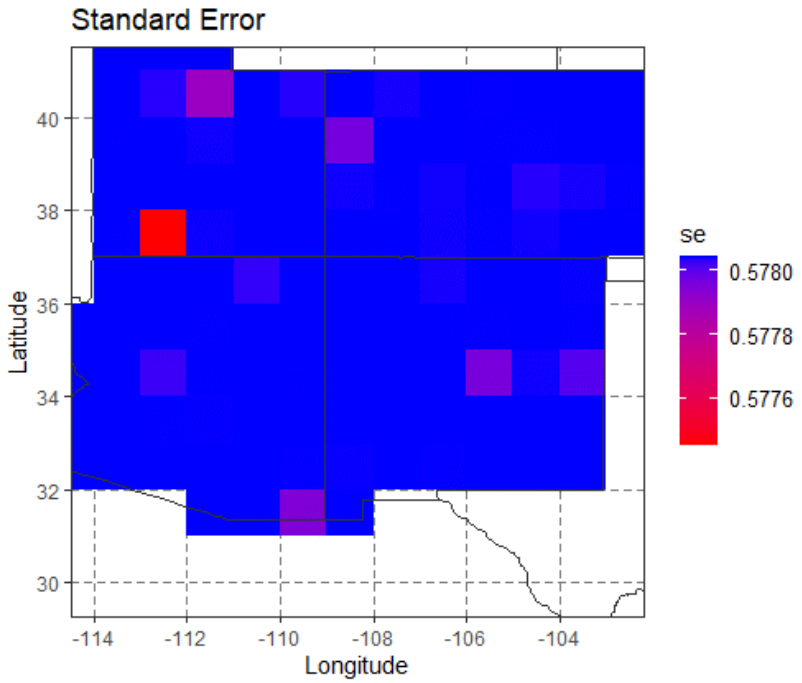

Figure 13: The standard error varies minimally across space. I imagine this is because the model predictions change minimally across space.

The below code shows how I made predictions from the posterior.

```
# loop through every vector of Beta and Theta in the posterior distribution
for (i in 1:n.samples2){
  zz = Krig(X[,1:2],log_mod$residuals,rho=theta[i,1],theta=1/theta[i,3],m=1,sigma2=theta[i,2])
  y2 = predict.Krig(zz,x=x1[,1:2],drop.Z=TRUE)
  yselog[,i] = predictSE(zz, x=x1[,1:2], drop.Z=TRUE)
  y1 = beta[i,1] + beta[i,2]*x1[,1]+beta[i,3]*x1[,2]+beta[i,4]*x1[,3]
  ylog[,i] = y1+y2
}
y=logit2prob(ylog) #to obtain the real value
yse=logit2prob(yselog) #to obtain the real value
mean.y = apply(y, 1, FUN = mean) # take the mean and convert from percent to fraction
mean.yse = apply(yse, 1, FUN = mean) # take the mean and convert from percent to fraction
ypred=data.frame(fit=mean.y, se=mean.yse)
```

Conclusions (Bayesian spatial logistic model)

- This model is dominated by the intercept, resulting in less spatial variation in probability than in problem 1.
- Like problem 1, the spatial pattern is increasing probability in the southeast direction. But, the pattern is less obvious because of how small the latitude, longitude, and elevation coefficients are.
- The regression coefficients converge nicely in the posterior, but the spatial parameters show multiple peaks in the posteriors.

P4) PCA in space and time

Principal component analysis was performed with the var() and svd() functions in base R. I wrote these commands, plus proportions of variance calculations and scree plotting, into an R function called pca(). See HW2 library for code.

i. PCA on summer global SST anomalies

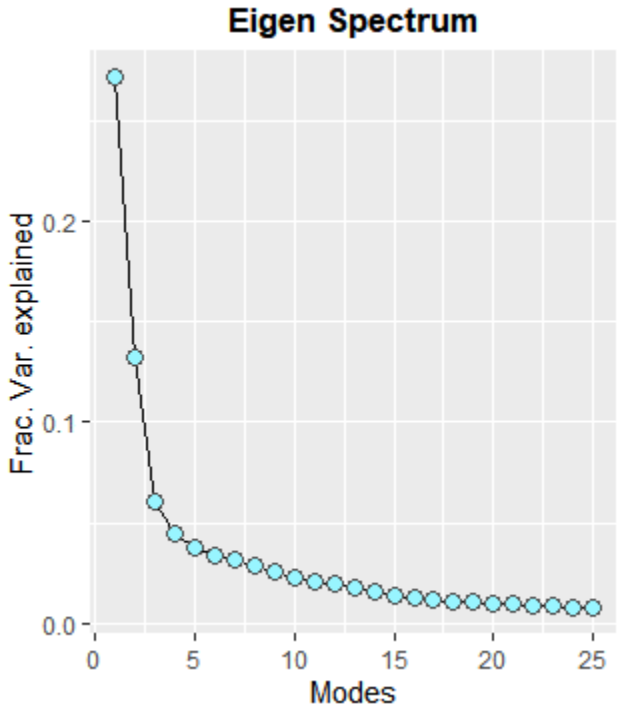

Figure 14: Scree plot for PCA on SST anomalies. The first 3 PCAs explain 46% of variance, and the first four explain 51% of variance. The remaining PCAs will be considered noise.

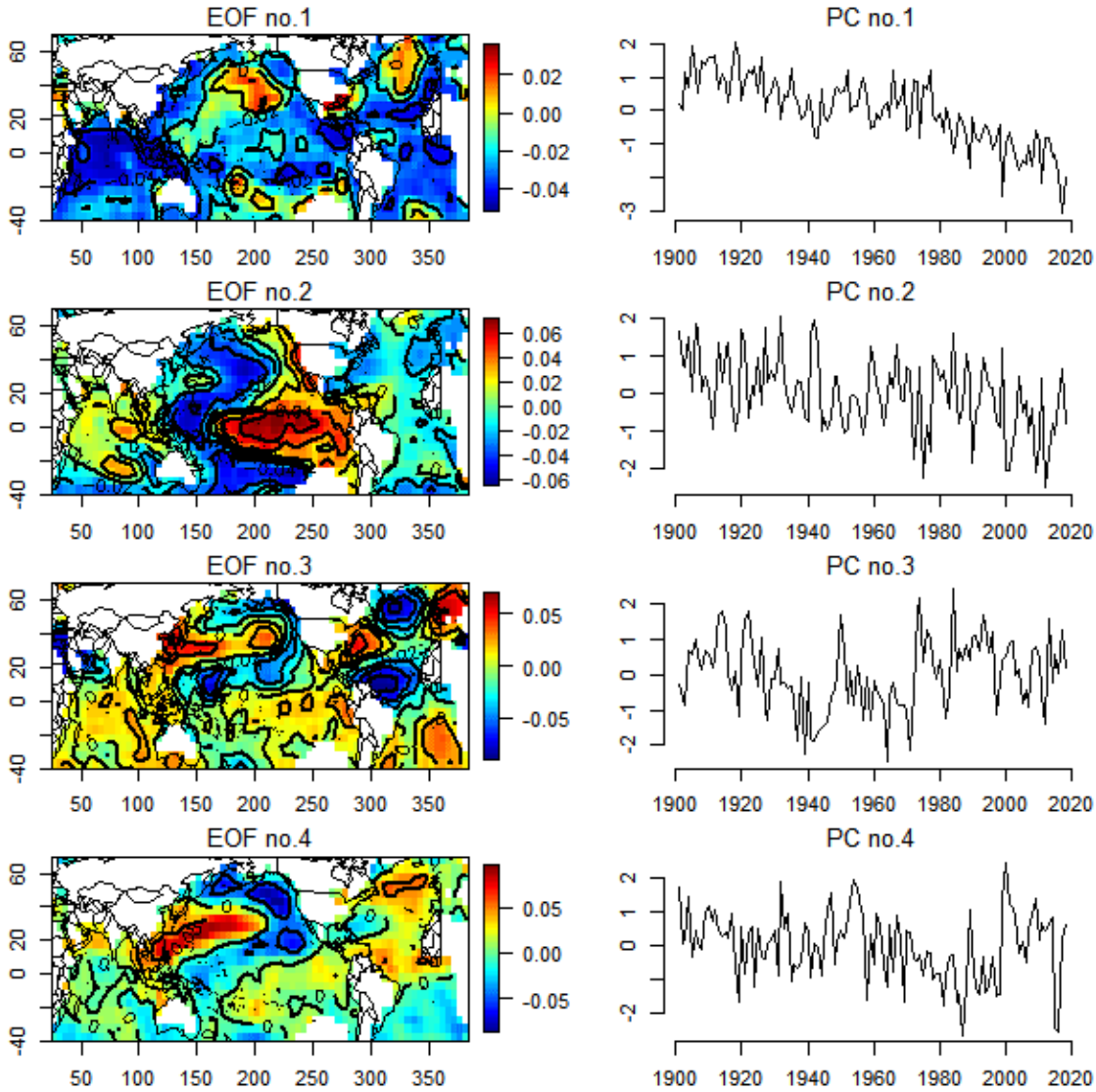

Figure 15: The left column shows the eigen values (AKA loading scores or phi values) for each latitude (y axis) and longitude (x axis) coordinate pair. EOF stands for empirical orthogonal function. The right column shows the principal component projection (Z value) for every year. Rows show results for PC1 through PC4. Note that the bottom left of the spatial plots is near South Africa and the bottom right is near South America.

ii. Perform a rotated PCA (rotate the first 6 PCs) and plot the leading 4 spatial and temporal modes of variability. Compare the results with (i) above.

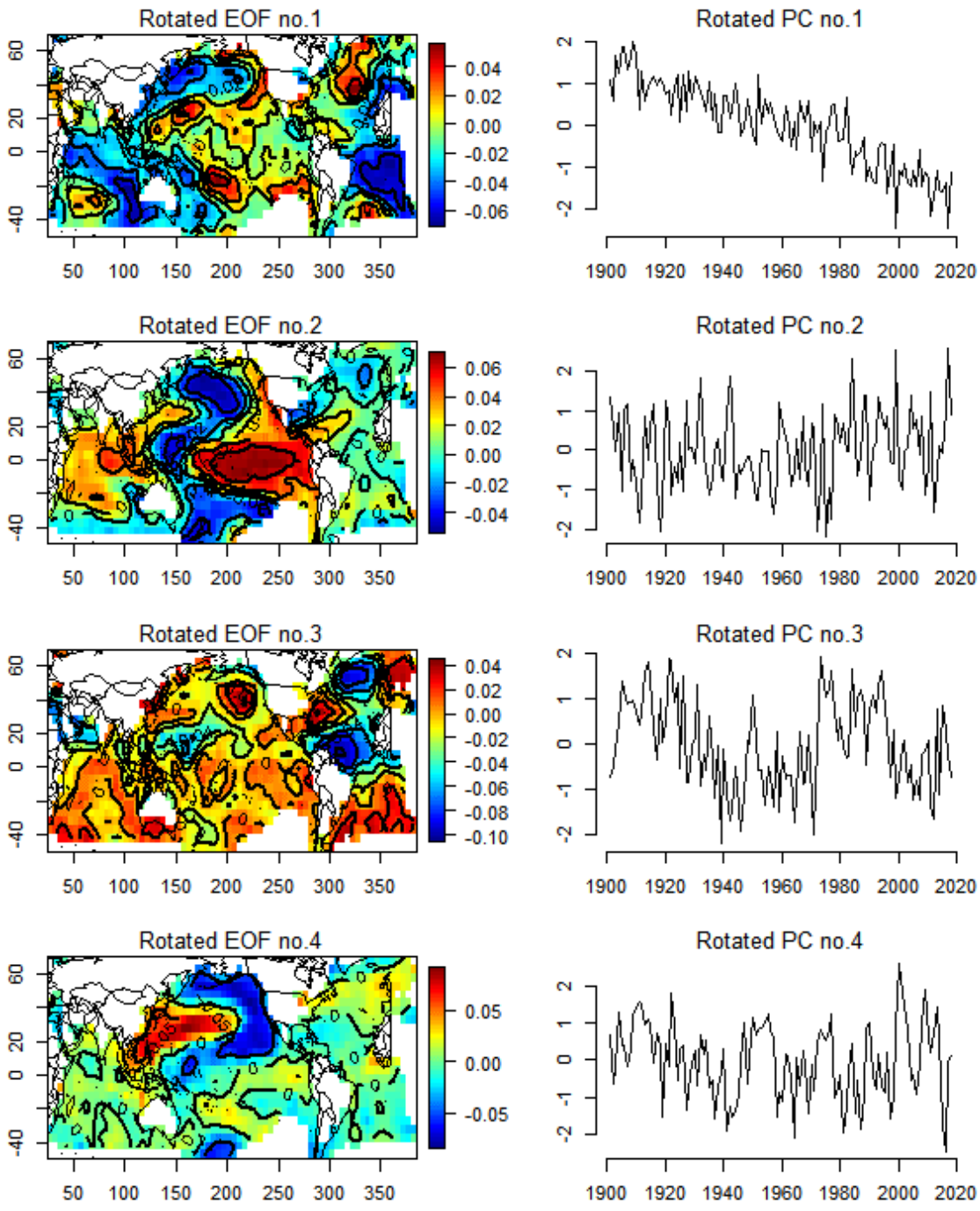

Figure 16: The rotated empirical orthogonal function does not appear to create any simpler revealing structures compared to unrotated results. This suggests that the unrotated results are robust.

iv) Correlate the first four PCs of rainfall with SSTs and vice-versa.

Correlation of SST anomalies to winter precipitation PCs:

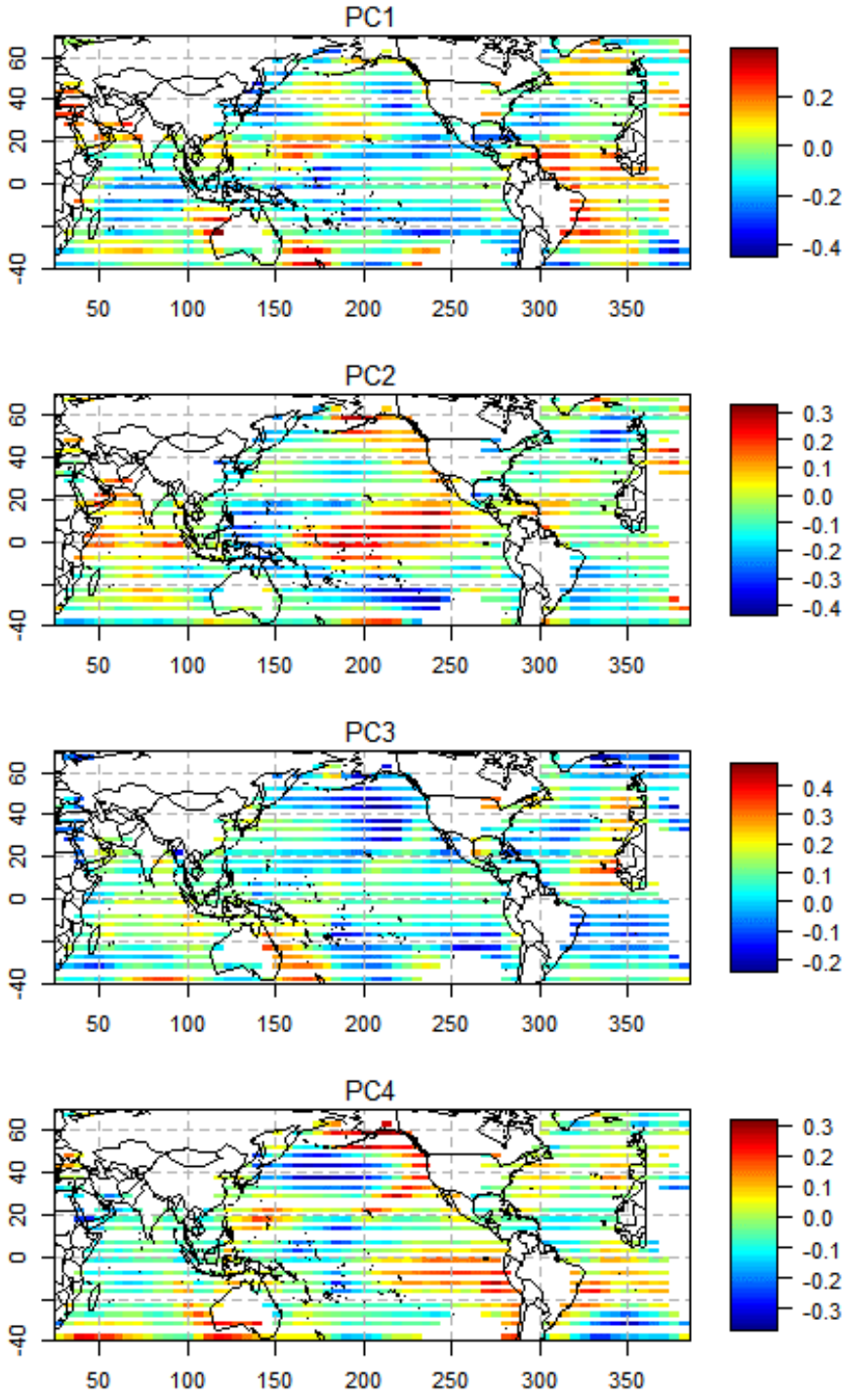

Figure 17: Correlation of SST to winter max precipitation. For PC1, note that the largest magnitudes of correlation occur in the Pacific near western US coast, the Indian ocean, and Atlantic ocean off the US coast (all indicated by blue). The largest magnitude of correlations is 0.4. PC2 has high correlation in the Pacific near the equator.

Correlation of winter precipitation to SST PCs:

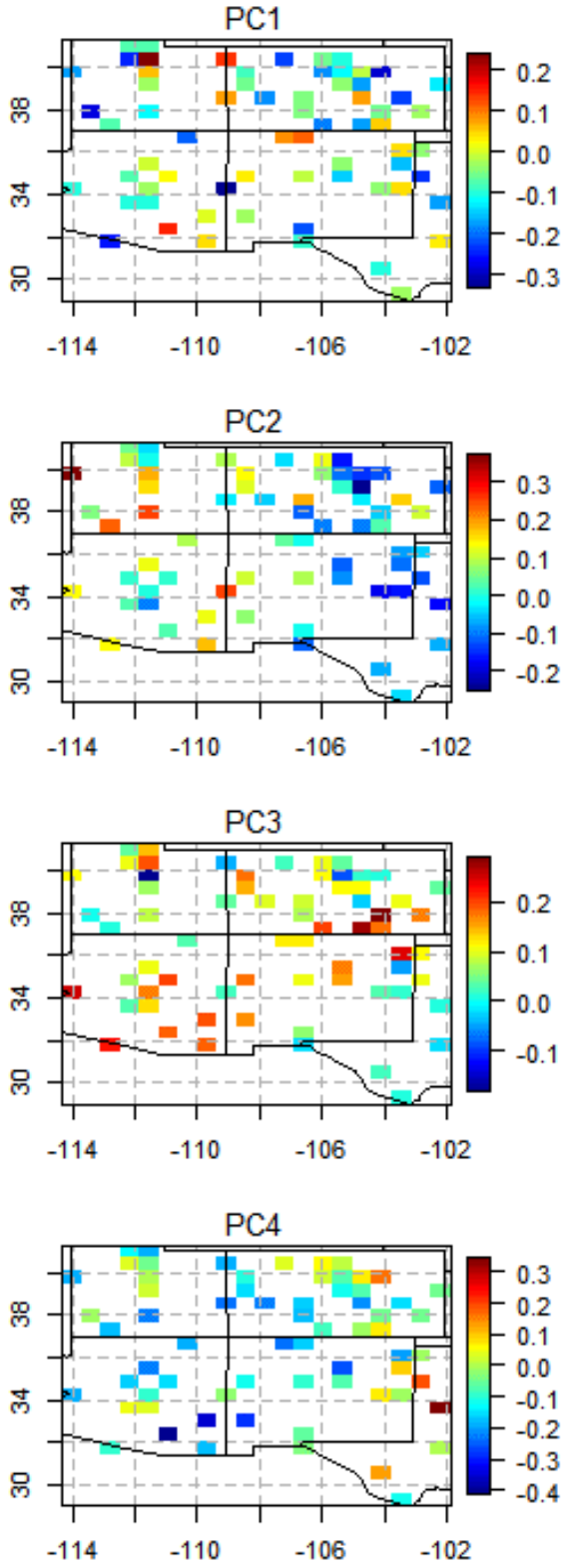

Figure 18: Correlation of winter max precipitation to SST anomaly PCs. A notable pattern exists in PC2. High correlation occurs across the eastern part of the domain.

Conclusions (PCA and correlation analysis between SST anomalies and winter max precipitation)

- Varimax rotation did not reveal any obvious simpler structures in the EOF.
- PC1 shows a clear negative trend.
- Max correlation between SST and precip PCs are around 0.4
- The correlation between precipitation and SST PC2 are strongest along the Colorado front range and in eastern NM.

P5) PCA + Multinomial regression

(a) Perform a PCA on the attribute data – show the Eigen spectrum and the leading four Eigen vectors

Uses pca() as in P4, documented in Library 2.R in appendix.

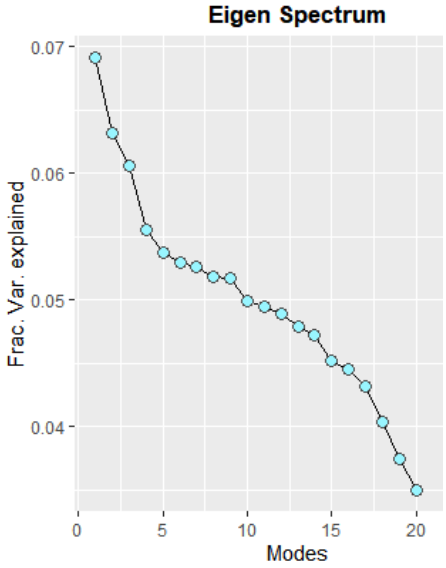

Figure 19: The eigen spectrum shows a slight knee at PC (mode) 5. After PC5, the PVE decreases from Modes 5 to 14, then increases again from Modes 15 to 20.

Plot loading scores of each variable for first 4 PCs:

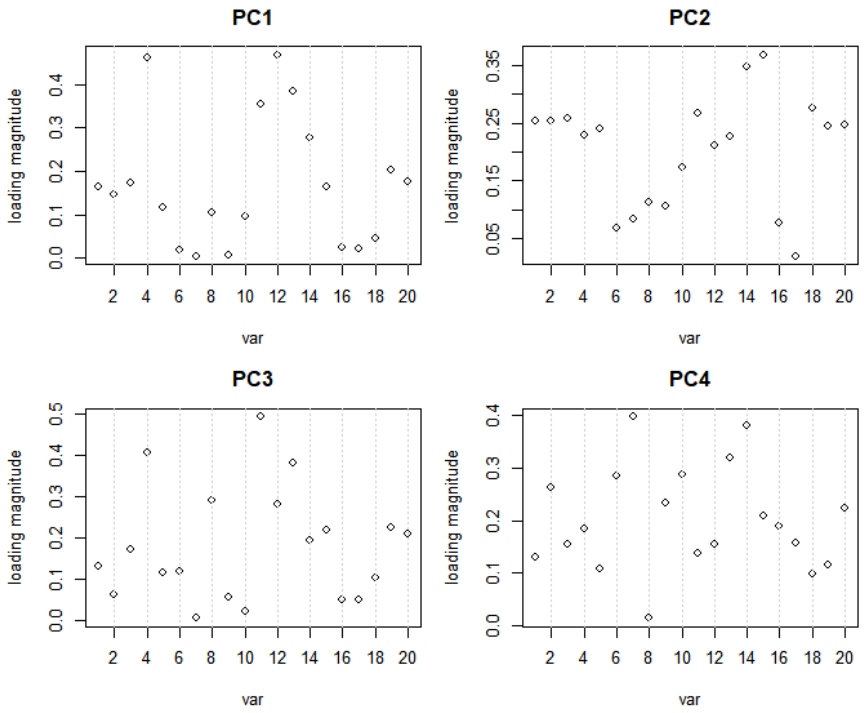

Figure 20: The magnitude of loading scores are plotted on the y axis for the first 20 variables (x axis) for PC1-PC4. Variables with similar magnitudes are often correlated and thus contribute similarly to the PC. For example, variables 4 and 12 are the two greatest for PC1 (welding and small particle), and scenarios with welding often produce small particles. See table 1 for names of the variables.

index	name
1	heavy.material.tool
2	spool
3	stairs
4	welding
5	wire
6	drill
7	hammer
8	hazardous.substance
9	insect
10	powered.tool
11	slag
12	small.particle
13	wind
14	lifting.pulling.manual.handling
15	exiting.transitioning
16	sharp.edge
17	splinter.sliver
18	improper.body.position
19	object.on.the.floor
20	uneven.surface

Table 1: The description of each variable in the multinomial construction injury problem.

(b) Fit a best multinomial regression with the principal components as predictors to model the categorical ***probability of injuries to the five category of body parts*** and compute the RPSS

Using stepwise backward BIC selection, the best model uses all of the first 5 PCs (V2-V6). The model summary is shown below:

```
multinom(formula = bpartbin ~ V2 + V3 + V4 + V5 + V6, data = pcs)

Coefficients:
  (Intercept)         V2          V3        V4          V5          V6
2  -2.2438649 -0.7766471 0.246149026 1.0620629 -0.55517676 -0.08112037
3  -0.5846061 -1.1999575 0.414659520 0.4808961  0.03422784  0.31944365
4   0.6476484 -1.2181742 0.609665762 0.8093743 -0.66989856 -0.05310383
5  -0.1351540 -1.3072489 0.001516386 0.8625743 -0.23205286  0.09080287

Std. Errors:
  (Intercept)         V2         V3         V4         V5         V6
2  0.12749813 0.14803604 0.13897740 0.12714744 0.13099048 0.12385872
3  0.06625238 0.07892963 0.07146568 0.07695133 0.08252865 0.06899861
4  0.05005971 0.06071060 0.06097837 0.06031323 0.06281243 0.06169986
5  0.05868497 0.07307324 0.06805990 0.07266988 0.07348332 0.07067359

Residual Deviance: 9709.119
AIC: 9757.119
```

The **RPSS is 0.26**, using the rps() function in the verification package:

```
rps(obs=acc2, pred=ypred, baseline=NULL)$rpss
```

(c) Repeat (b) to ***predict Injury Severity***

Note that the PCA has not changed from part b, since only the Y variable has changed.

The best model, according to backward stepwise selection with BIC uses only PC1 (V2) as a predictor. The model summary is shown below:

```
multinom(formula = sevbin ~ V2, data = pcs)

Coefficients:
  (Intercept)          V2
2   1.4381595  0.02726605
3   1.0679115 -0.26862427
4  -0.4684275 -0.09234009
5  -1.9504183 -0.42255660

Std. Errors:
  (Intercept)         V2
2  0.07705400 0.05862423
3  0.08055404 0.06774241
4  0.11150628 0.09158521
5  0.20333914 0.20148034

Residual Deviance: 4551.719
AIC: 4567.719
```

The **RPSS is 0.0151**, again using rps() function. Note that a perfect forecast is RPSS =1. RPSS = 0 indicates performance equivalent to observations, and negative RPSS means performance worse than observations.

(d) Apply CART and compare the results

I used rpart() instead of tree() to create a classification tree such that I could gain more control of which tree was returned from the functions. Using the tree function, the best tree used only PC1 and predicted severity of '2' always. I believe this is because the vast majority of observations are of class '2' (almost half). Therefore, I used rpart() and reduced cp parameter, which controls how much improvement in purity is needed for a new split to be added. The code is below:

```
sev_tree=rpart(sev ~ V2+V3+V4+V5+V6, data=pcs, model=T, control = rpart.control(minbucket = 1,cp=0.003))
```

The resulting tree, which is shown below, achieved an **RPSS of 0.0189.** This is a **slight improvement over the multinomial model**.

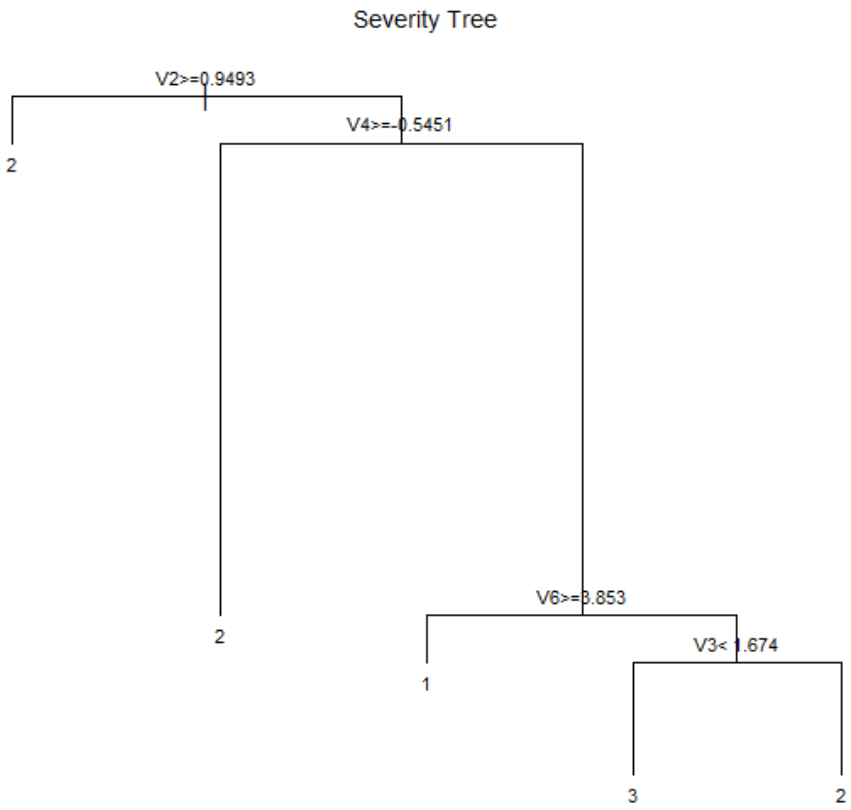

Figure 21: The classification tree for injury severity using PCs 1-5 (V2-V6). Although there are five categories of injuries, the tree only predicts for classes 1-3 since 4 and 5 are so rare.

Conclusions (Multinomial and CART on construction injury type/severity)

- PCA does not reveal an obvious sharp knee at which to interpret some PCs as signal and others as noise.
- RPSS for multinomial model of injury type is 0.26, meaning a 26% improvement over using empirical probabilities.
- RPSS for multinomial model on injury severity 0.0151, only a <2% improvement over using empirical probabilities.
- CART model for injury severity slightly improved RPSS to 0.0189, but still only a ~2% improvement over empirical probabilities.
- Tixier et al. (2016) modeled these using Random Forests and Stochastic Boosting. They could not get good skill for Injury Severity. **Compare your results with theirs.**
    - From figure 8 of Tixier et al. 2016, the median RPSS of cross validation was -.5 for boosted trees and -.8 for random forests. The training set values were between .2 and .3. This indicates their model struggles significantly with prediction. My multinomial and CART model have smaller RPSS of 0.015 and 0.0189 respectively. However, I expect the multinomial model to outperform there model in terms of predictive power because of it's relative simplicity (uses only PC1 as predictor).

P6) Multivariate Forecasting with CART and Random Forests

NOTE: I used 10 PCs instead of 4 to see if the model improved. It did, but not very much.

The SST anomaly data was truncated to match the same years as the winter precip data (1964-2018). Then, PCA was performed for both using the pca(scale=T) function shown in the appendix.

Then, a regression tree was fit to each of the first 10 winter precipitation PCs using the first 10 PCs of SST as predictors, and pruned to minimize deviance. Where the pruning resulted in a single node, then the next size larger tree was taken for prediction. See code below:

```
for (i in 1:npc){ # model first npc PCs
  df=data.frame(precip_pc=pca_precip$pcs[,i], pca_SST$pcs[,1:npc])
  tree.unpruned=tree(precip_pc ~ ., data = df, model =T) # need model = T for cv.tree to have dataframe

  plot(tree.unpruned)
  text(tree.unpruned, cex=.7)
  mtext(paste('PC', i, 'untrimmed'), line=1.1 ,cex=.9)

  #Perform CV on tree object
  cvTree <- cv.tree(tree.unpruned)
  optTree <- which.min(cvTree$dev)
  bestTree <- cvTree$size[optTree]

  if (bestTree ==1){
    bestTree = 2
    print(paste ( 'Min deviance for PC', i, ' is 1 node.'))
  }

  #prune Tree based on CV results
  pruneTree <- prune.tree(tree.unpruned, best = bestTree)

  tree.list[[paste('PC',i, sep='')]]=pruneTree

  plot(tree.list[[paste('PC',i, sep='')]])
  text(tree.list[[paste('PC',i, sep='')]], cex=.7)
  mtext(paste('PC', i, 'trimmed'), line=1.1, cex=.9)
}
```

The resulting trees are shown below:

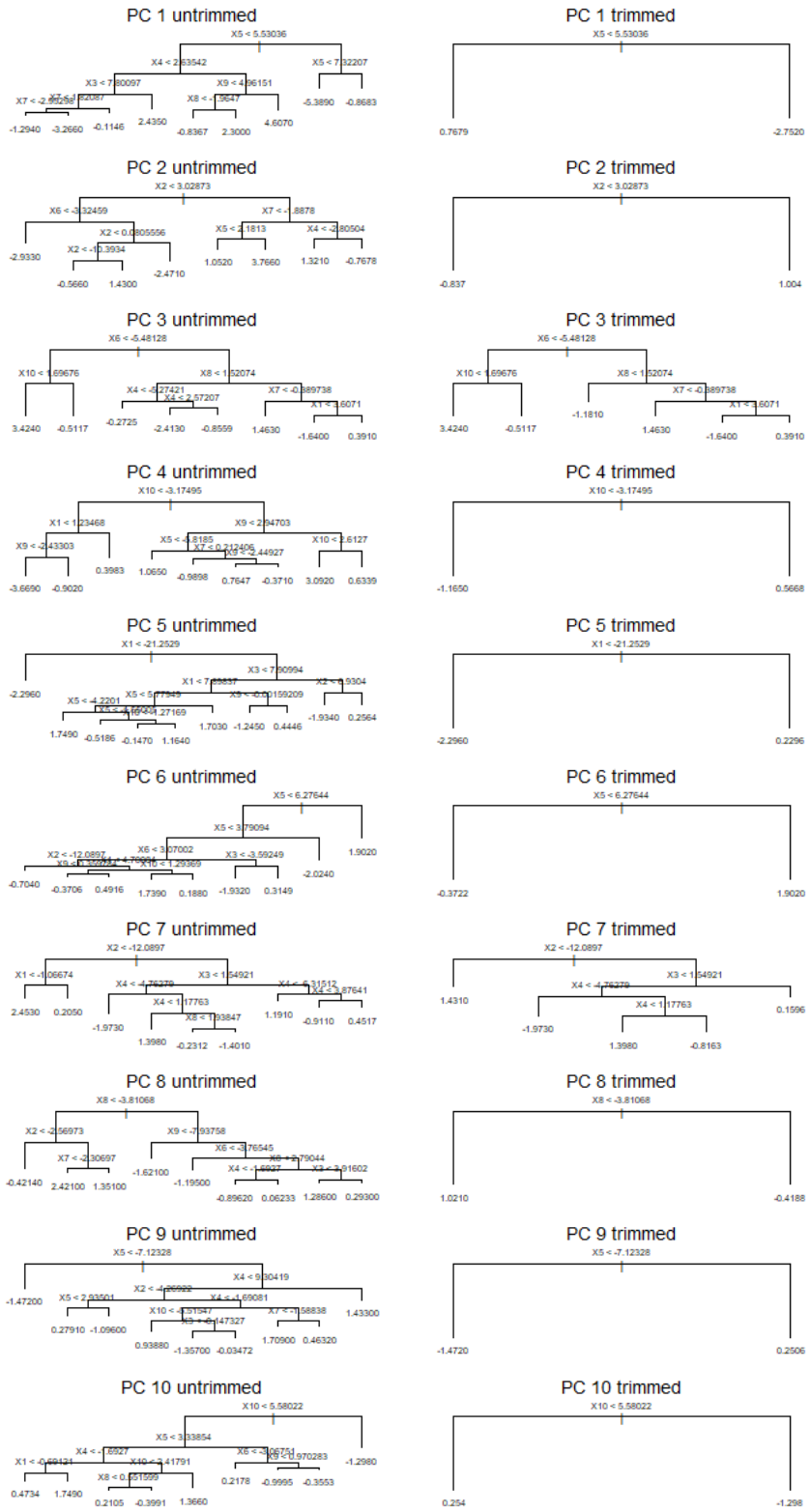

Figure 22: Fitted CART models for winter PCs as function of SST PCs. The left column shows unpruned trees and the right shows pruned trees to minimize deviance. Each row is for PC 1-10. The purpose of this figure is to demonstrate how simple each tree is after pruning, minus PC3 and PC7.

a) Compare CART model precipitation with historic, and perform drop-10% cross validation.

Correlation of predictions vs observed precipitation is shown in the figure below.

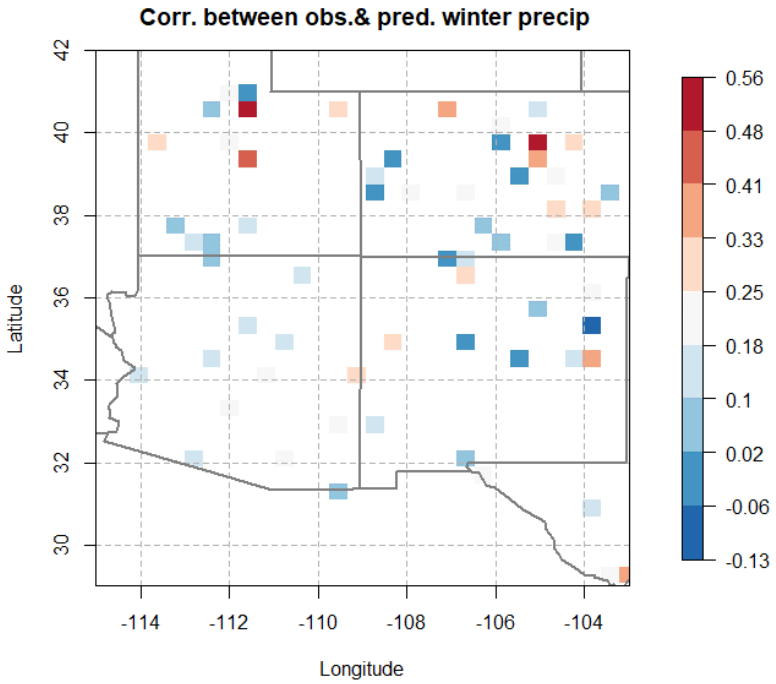

Figure 23: Correlation of historical max winter precipitation with predicted values with CART model. Overall, the correlations are low minus one site in CO and 2 in Utah.

Drop 10 analysis

I wrote two functions to perform drop 10 analysis. The first function is called CART_PCs. This function fits a tree to each of the first npc PCs of the y variable using the first npc PCs of the x variable. This function uses the same code shown at the beginning of problem 6. Then, function CART_drop10 calls function CART_PCs to iterate drop 10 analysis. Note that 10 percent of rows are dropped, then trees are fit to the remaining rows for each of npc, then the dropped rows are predicted. Correlation and RMSE of each row to the observed y values are computed, and the boxplots lumps all columns into a single plot. In this problem, rows correspond to years and columns to locations. The results are below:

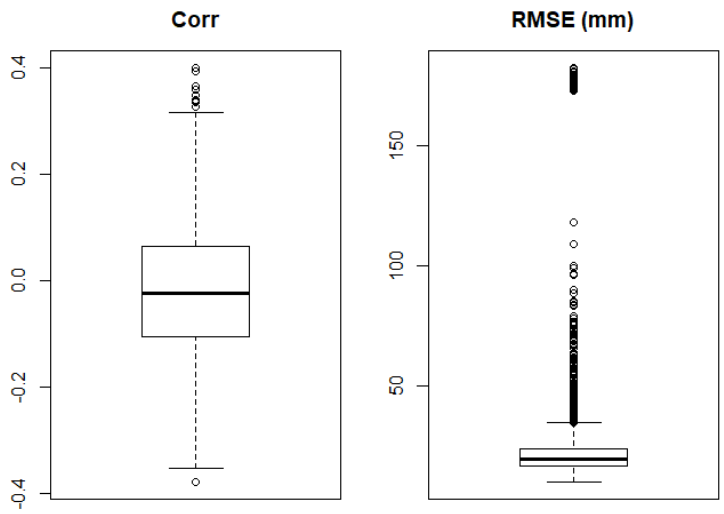

Figure 24: Drop 10 analysis on the multivariate CART forecasting model shows that correlation is usually near zero under prediction, which is similar to the results of the fitted model (from visual inspection of figure 23). RMSE is usually between 20 and 30mm.

b) Repeat (a) with random forest

Random forests were created using the randomForest package. Similar to part a, I created two functions: the first, RF_PCs, fits a RandomForest model to the first npc PCs of the Y variable using the first npc PCs of the X variable as predictors. The second function iterates the RF_PCs function for drop 10 analysis. Results are below.

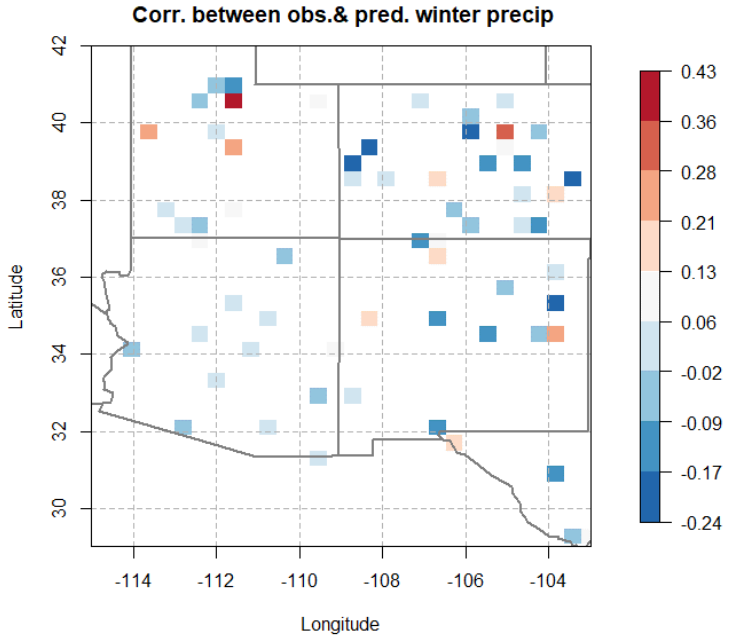

Figure 25: Correlation of historical winter precipitation and predictions with Random Forest model. Comparing to Figure 23 (correlation with CART model), the performance of fitted model is worse than CART. However, the spatial pattern of correlation is very similar. I expect random forest model to have worse fitting performance and better prediction performance than CART.

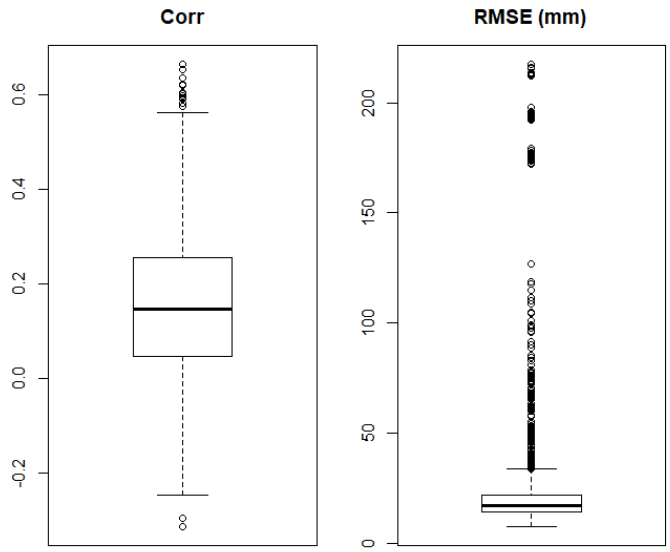

Figure 26: The correlation and RMSE of drop-10 analysis using random forest model. As expected, the random forest performs better in prediction than does the CART model. For example, the median correlation increases from 0 to 0.15 when using random forest compared to CART.

Conclusions (CART and Random Forests for multivariate forecasting of winter precip as function of SST PCs)

- The correlation of fitted values to observed values is pretty low, centered near zero, for both random forest and CART. I am not surprised, as the correlation analysis in problem 4 indicates that correlation between winter max precip and SSTs is not that strong (max of 0.4, usually closer to 0).
- The CART model achieves superior correlation and RMSE compared to random forest. However, random forest achieves better prediction skill in drop 10 analysis. This demonstrates the conceptual underpinning of random forests: use naïve trees and aggregate votes to avoid overfitting.

P7) K means clustering

Note: I used latitude, longitude, and precipitation to assign clusters. Otherwise, winter and summer clusters would be the same since observations are at the same locations.

a. Cluster the winter max precipitation

I calculated within cluster sum of squares for k= 1:15 using K means clustering. The results are shown below:

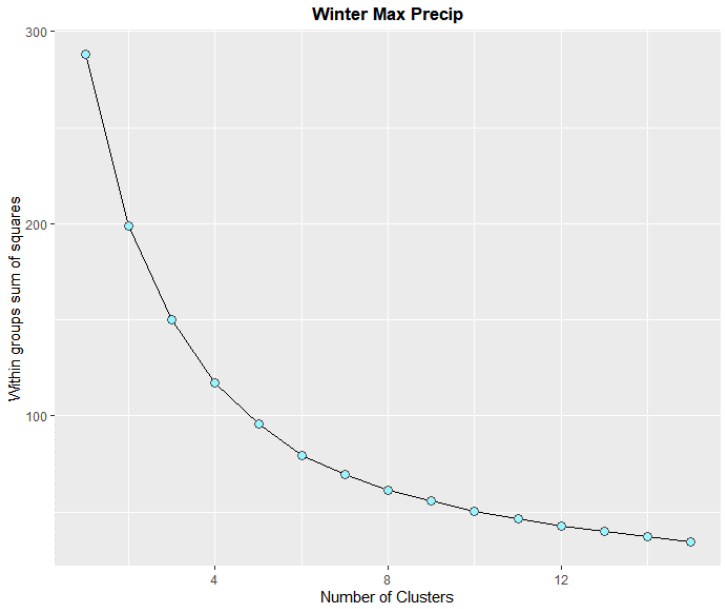

Figure 27: Within cluster sum of squares (WSS) plotted against number of clusters. The knee occur around K=5 clusters. Therefore, clustering was performed with 5 clusters.

K=5 was chosen as the knee where within cluster sum of squares (WSS) begins to drop off.

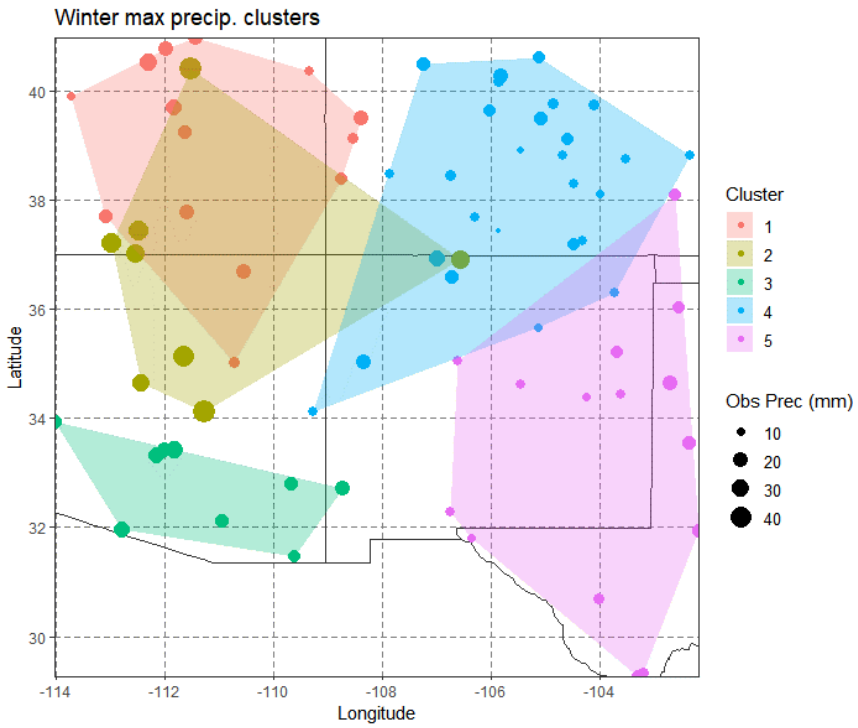

Figure 28: The resulting five clusters of winter max precipitation. Cluster three is the only cluster spatially distinct form all other clusters.

b. Clustering summer max precipitation average.

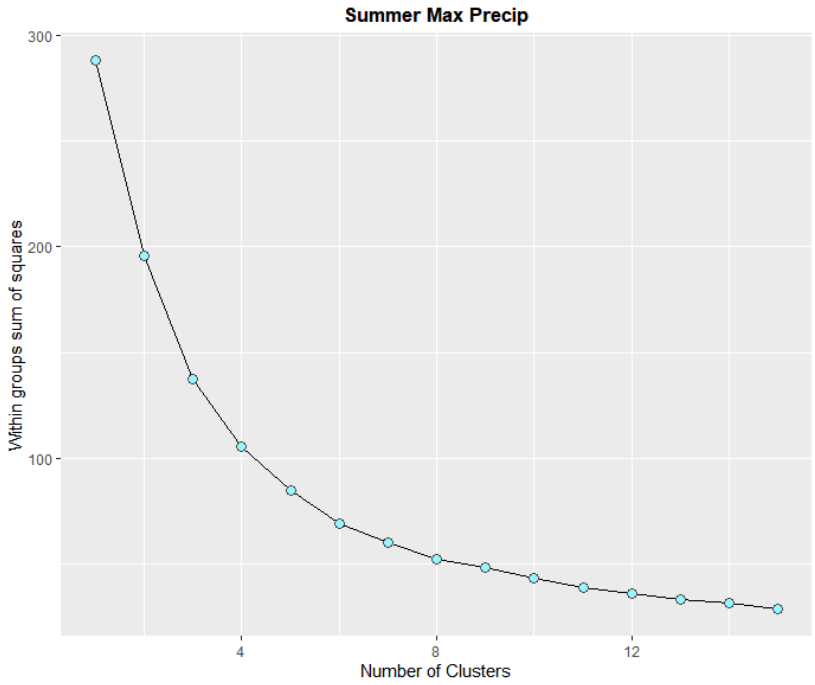

Figure 29: WSS plotted against number of clusters, K, for summer max precipitation. Like winter precipitation, K=5 was chosen as the knee.

I chose K =5 as the ideal number of clusters.

The resulting cluster assignments are shown in the spatial map below.

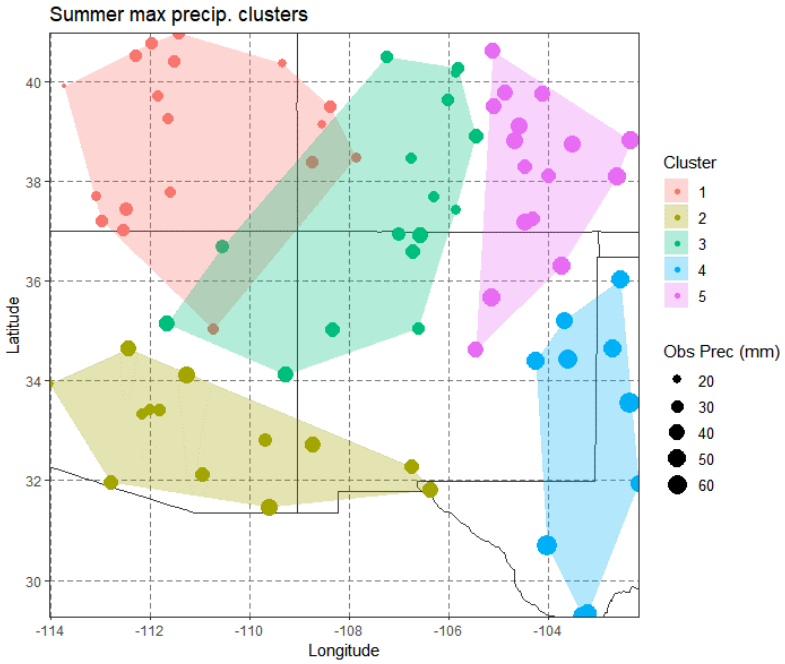

Figure 30: The summer clusters are similar in location to the winter clusters (based on shared parameters of elevation, lat, and lon). However, summer clusters demonstrate greater spatial separation than the winter clusters.

c. Repeat with another clustering method. I chose **hierarchical clustering**

Note: I chose K=5 clusters to match k-means clustering from parts a and b.

Example code for hierarchical clustering:

```
######## summer

summer_hier=hclust(d=dist(summer_scale), method='complete')
plot(summer_hier)
rect.hclust(summer_hier, k=5, border=2:6)

# append cluster assignment

summer_cut=as.character(cutree(summer_hier, k=5))
data_3 = data.frame(Summer_Precip, Cluster=summer_cut)
```

Winter Results:

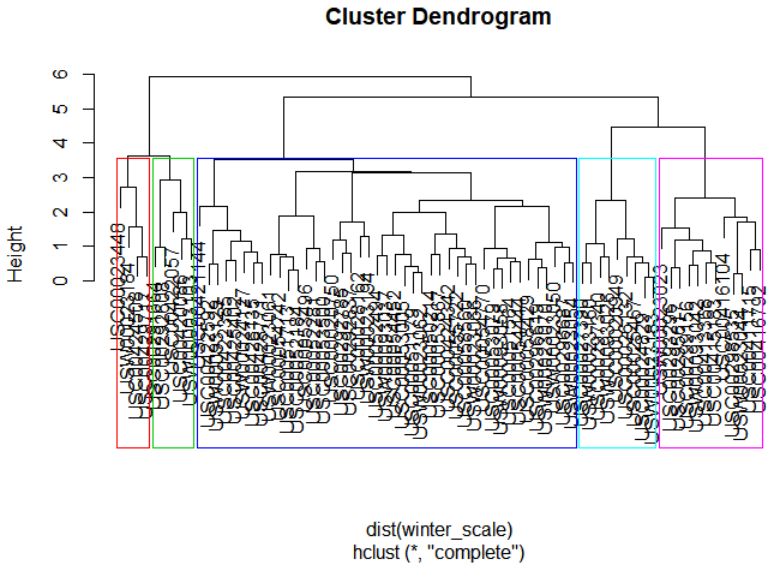

Figure 31: Cutting the winter hierarchical tree such that 5 clusters are chosen.

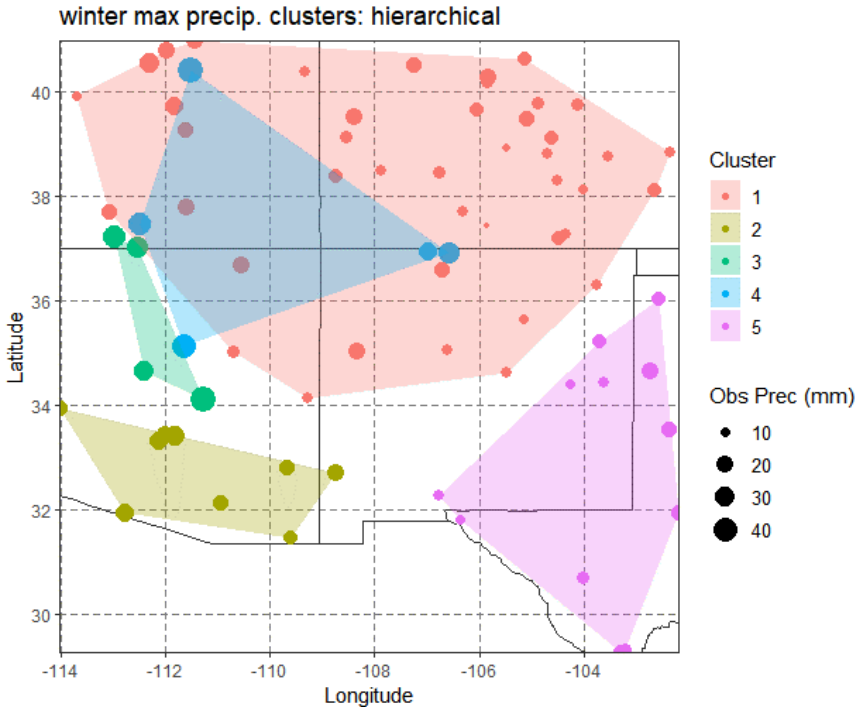

Figure 32: The resulting five winter hierarchical clusters. Clusters in southern AZ (2) and Texas/NM (5) are similar to clusters from k-means. However, clusters 1, 3, and 4 are very different from k-means.

Summer Results:

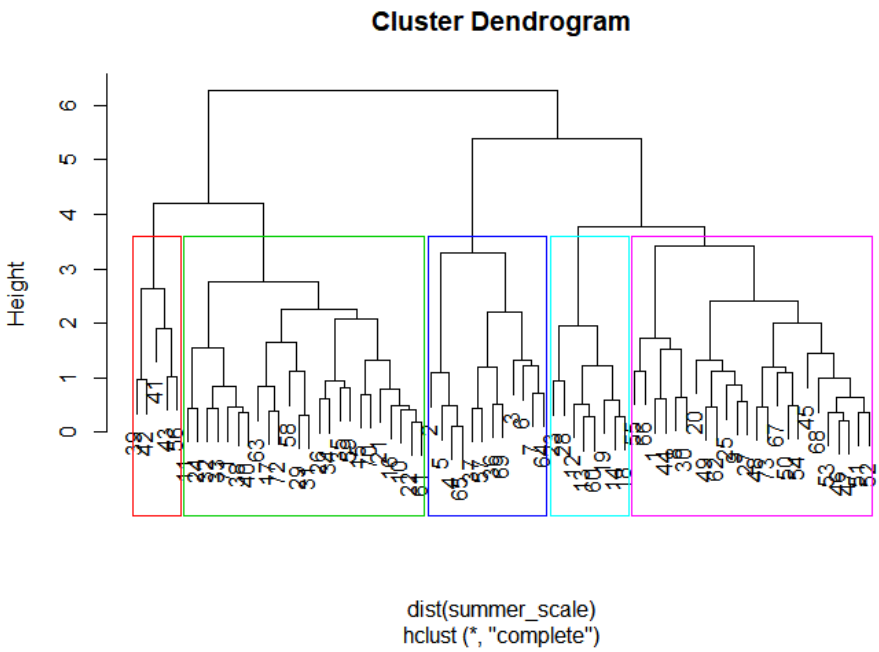

Figure 33: Cutting the summer hierarchical tree such that five clusters are chosen.

With k=5 clusters

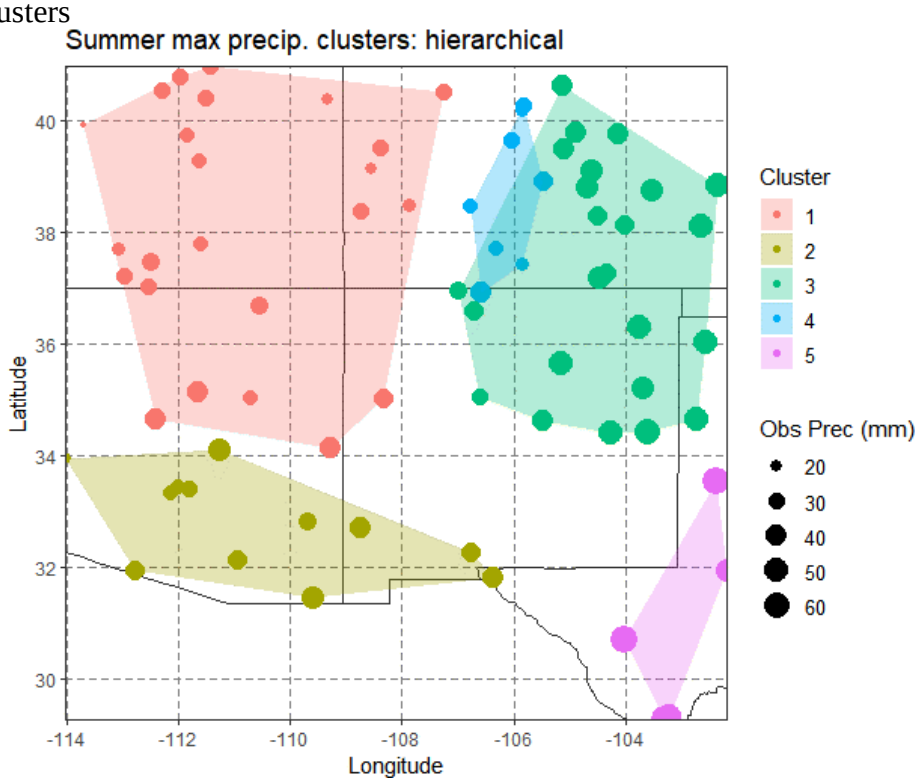

Figure 34: The resulting five clusters for summer precipitation using hierarchical clustering. Notice how this method results in clusters with greater variation in the number of clusters in each compared to k-means.

Conclusions (k-means and hierarchical clustering)

- K-means clustering with five clusters resulted in similar clusters for both summer and winter.
- Both clustering methods resulted in spatial overlap of clusters.
- Hierarchical clustering resulted in both larger and smaller sizes (enumerated by number of sites included in each) than did k-means clustering
    - For example, the winter hierarchical clustering resulted in a massive cluster (1) and two tiny clusters (3 and 5). I believe this is because of the bottom-up approach of agglomerating clusters that are most similar. The small clusters are apparently oddballs.

P8) Extremes clustering

For the winter 3-day maximum precipitation over Southwest US perform clustering of the Extremes using the method proposed in Bernard et al. (2013) and used in Bracken et al. (2015) – these papers are linked on the class page. Comment on the cluster patterns in comparison with those from problem 7 above.

a) Silhouette width

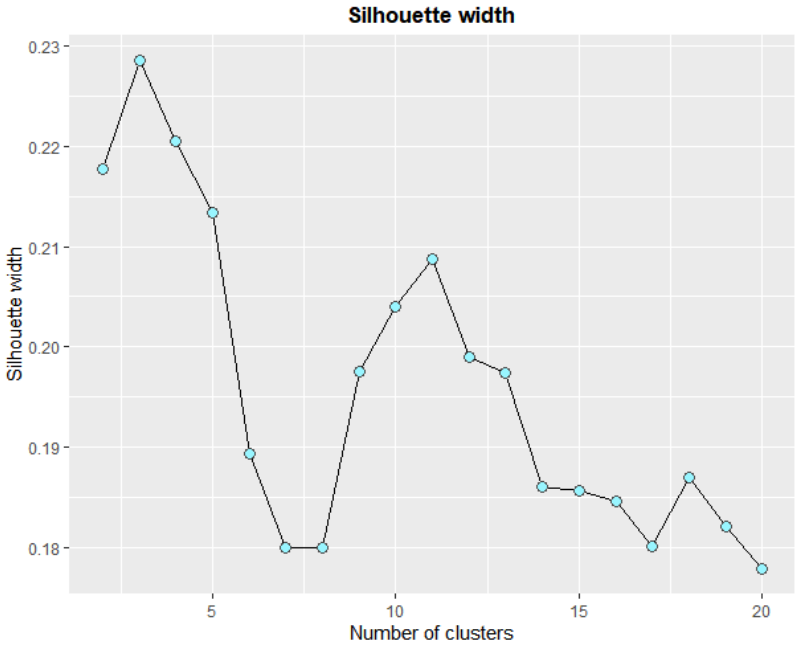

Figure 35: The silhouette width plotted against number of clusters. The optimal number of clusters is where silhouette width is maximized. This occurs at K=3.

I chose K=3 according to silhouette width.

b) Plot the clusters

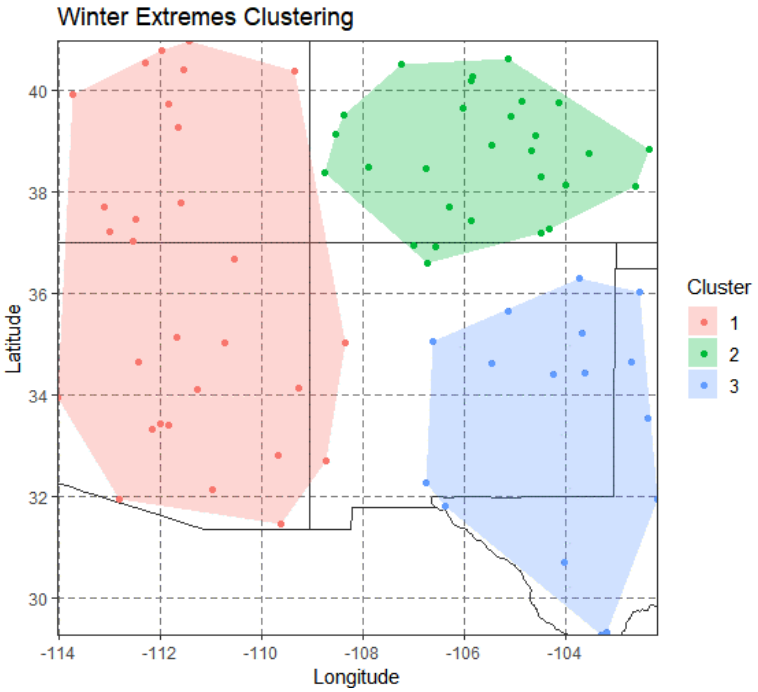

Figure 36: This method, based on the madogram using time series of precipitation, results in three spatially distinct clusters. Cluster one covers the western part of the domain, cluster 2 covers CO, and cluster 3 covers eastern NM and western Texas.

c) I also use k=5 to compare directly with K-means in P7

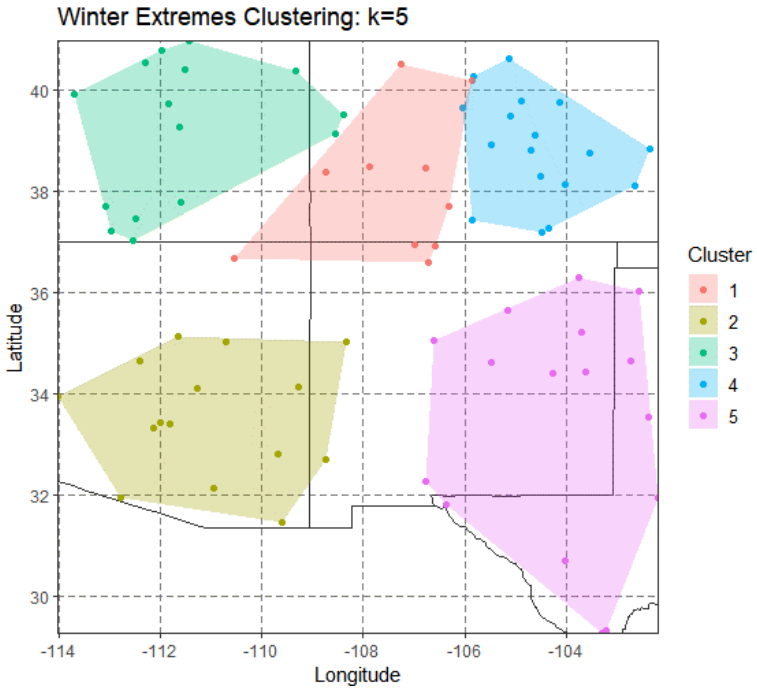

Figure 37: Clustering with the modogram function and K=5. Compared to problem 7, the locations are spatially distinct, but still follow a similar pattern.

Conclusions (extremes clustering):

- For both K=3 and K=5, the extremes clustering algorithm results in spatially distinct winter clusters.
- For K=3, the clusters largely coincide with states: cluster 1 is AZ and UT, 2 is CO, and 3 is NM.
- For K=5, the clusters are similar to k-means clustering in P7. There are clusters in AZ and NM that are more distinct from other clusters, and there three clusters along 38 degrees latitude.
- Extremes clustering was able to spatially disjoin these three clusters.

P9) Self Organizing Maps

I used the package kohonen to perform SOM analysis with the som() function. For the winter max precipitation data, I used a 3x3 rectangular grid. Results are shown below.

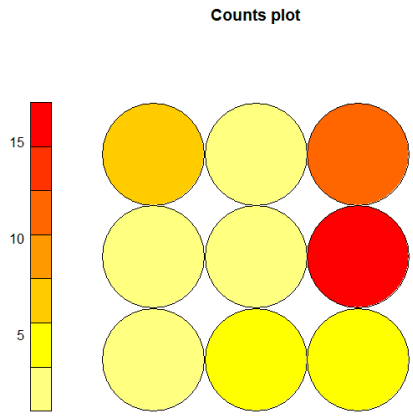

Figure 38: The above counts plot shows how many observations fall into each node. The top left is node 1 and bottom right is node 9.

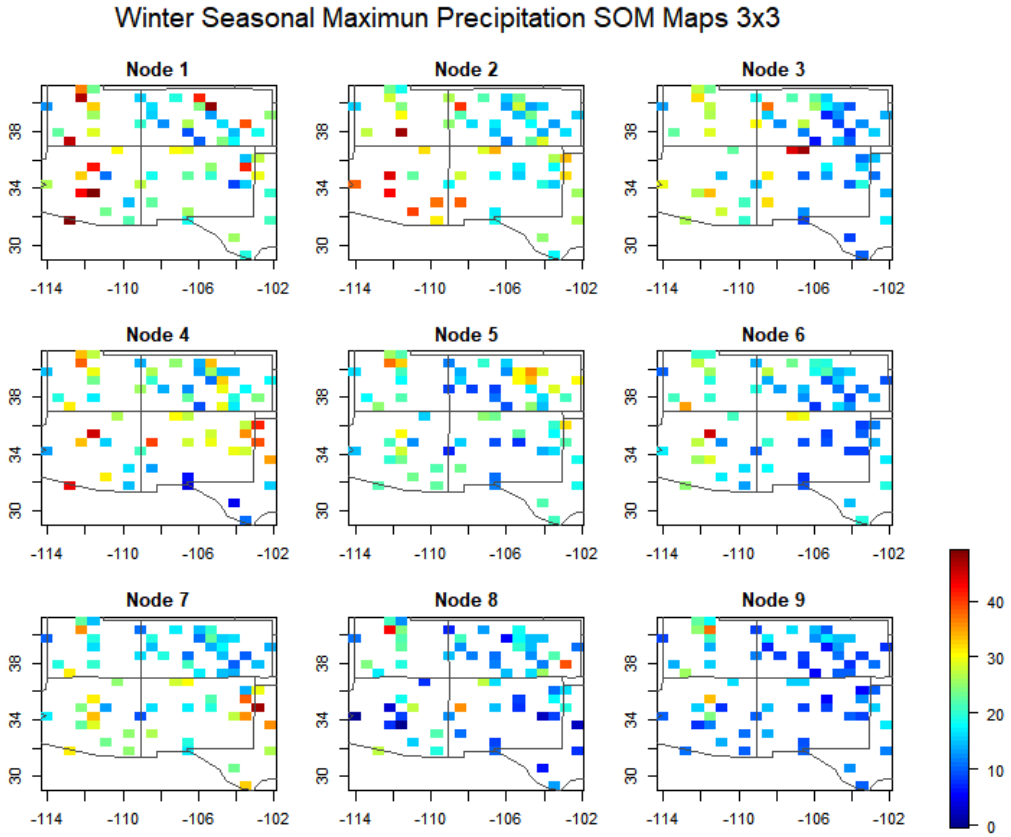

Figure 39: Mean of the max winter snowfall across all years of observations contained in each node. Neighboring nodes are more similar and nodes opposite each other are dissimilar. For example, Node 1 is wetter, and node 9 is drier.

Corresponding SSTs (mean taken)

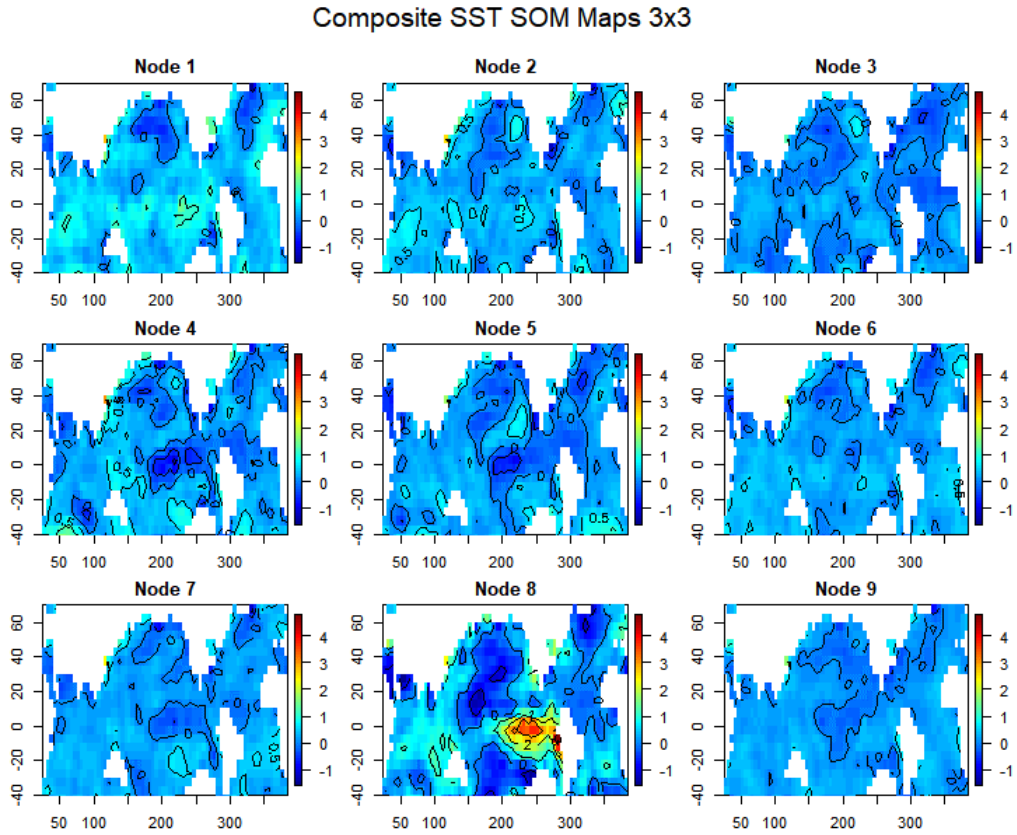

Figure 40: The mean of the SST anomalies for every observation (year) in each node. Although patterns and differences are difficult to distinguish, it is worth noting that most all values in node 1 are positive (light blue or green) whereas most values in node 9 are negative (dark blue). This corresponds to Node 1 being wetter and node 9 being drier. Also, note the distinct high values in node in the Pacific ocean near the equator (lon=250 and lat = 0). However, the connection to precipitation values in node 8 is unclear.

Conclusions (SOM):

- Nodes 3,6,and 9 are drier than nodes 1,4, and 7.
- Nodes 3 and 9 differ in that node 3 is dry in CO and NM and that node 9 is dry in CO, NM, UT, and AZ.
- Node 1 (more precipitation) is associated with positive SST anomalies. I assume this is because higher temperatures result in more energy and moisture for precipitation to occur.
- In contrast, node 9 (less precipitation) is associated with negative SST anomalies. I assume this is because lower temperatures result in less energy and moisture.

P10) Canonical Correlation Analysis

i) Take the leading ~ 4 PCs of the winter SST and perform CCA with the leading ~ 4 PCs of winter precipitation.

PCA was performed with pca() in HW2 Library.R (see appendix). CCA was performed with the following commands:

```
############ CCA on the PCs ###################

M=dim(sstPC)[2]
J=dim(precPC)[2]
J=min(M,J) # number of PCs
N = length(precPC[,1]) #number of years
Qx1 = qr.Q(qr(sstPC)) # qr is QR decomposition of matrix to solve Ax=b. qr.Q returns orthogonal transformation, Q
Qy1 = qr.Q(qr(precPC))
T11 = qr.R(qr(sstPC)) # returns component R
T22 = qr.R(qr(precPC))
VV=t(Qx1) %*% Qy1
BB = solve(T22) %*% svd(VV)$v * sqrt(N-1)
wm1 = precPC %*% BB
AA = solve(T11) %*% svd(VV)$u * sqrt(N-1)
vm1 = sstPC %*% AA
```

Where sstPC and precPC are the principal components 1 through 4 of the SST and precipitation data, respectively.

ii) Fit a regression for each canonical variate (precipitation variate as a function of SST variate)

Regression was performed with the below codes to find the regression coefficients, betahat. Then, predictions of precipitation PCs are made with the model in the below code.

```
### Predict the winter precipitation PCS
betahat = solve(t(AA) %*% t(sstPC)%*% sstPC %*% AA) %*% t(AA) %*% t(sstPC) %*% precPC
ypred=sstPC %*% AA %*% betahat
### first npc PCs from the PC forecast above and the remaining PCs are set to
## their means - i.e., 0

N1 = dim(winter_precip)[2]-(npc)

precPCpred = cbind(ypred,matrix(rep(0,N),ncol=N1,nrow=N)) # Z matrix, which is X*E

## back transform to get the winter precipitation field

###  Keep only the first npc Eigen Vectors and set rest to zero
E = matrix(0,nrow=dim(winter_precip)[2],ncol=dim(winter_precip)[2])
E[,1:npc]=pca_precip$u[,1:npc] # E matrix, ie loading or eigenvectors


## back transform to get the winter precipitation field
precpred =  precPCpred %*% t(E)
```

Then, after transforming the PC prediction by multiplying the PCs by the eigenvectors, the values are 'unscaled' by multiplying by the standard deviation and adding the mean (not shown, see Appendix).

iii) Evaluate performance by computing R^2 between observed and predicted winter precipitation at each grid point.

I chose to plot both Correlation and R^2 since I am a little unclear as to which is desired. They are shown in the figures below.

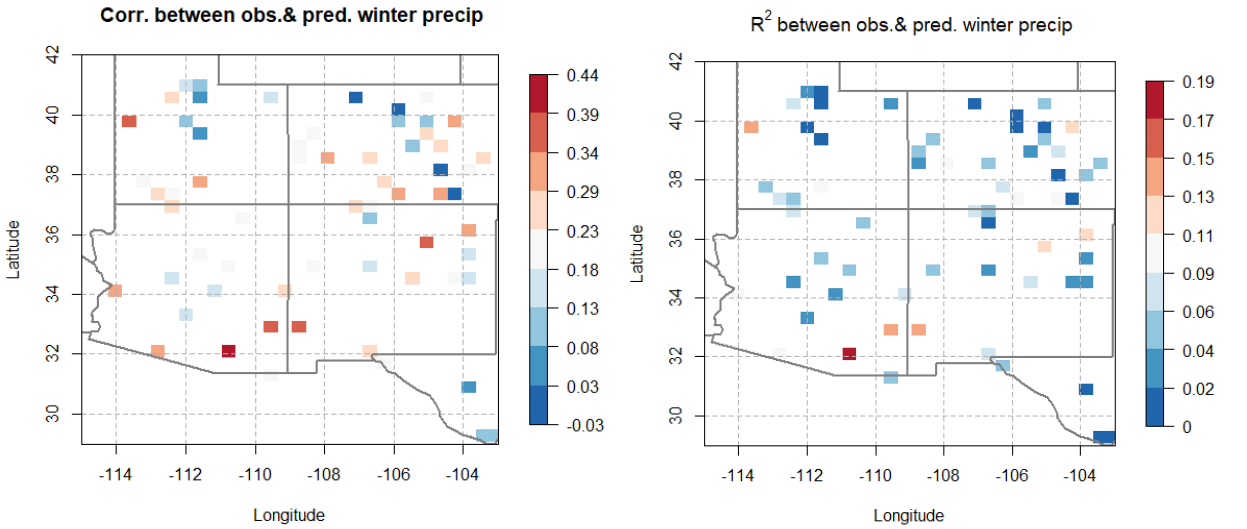

Figure 41: The resulting correlation and R^2 values for each location from Canonical Correlation Analysis compared to observations. Note that R2 was calculated as the square of the correlation coefficient. Peak correlation occurs in Southern AZ and at the westernmost site in UT with a maximum value of 0.44. Fit in CO and NM is poor with most locations having a correlation of less than 0.3.

iv) Perform LOOCV and compare results with problem 6

LOOCV was performed as follows: drop one row of SST PCs and one row of precipitation PCs, corresponding to one year of observations. Perform canonical correlation analysis with the remaining years, fit beta coefficients for precipitation PCs as function of SST pcs, then predict the dropped year. Repeat for every year 1964 to 2018. For every year predicted, calculate the correlation and RMSE of the year compared to the observed precipitation across the 73 sites. See Appendix for code. The results are shown in the below figures.

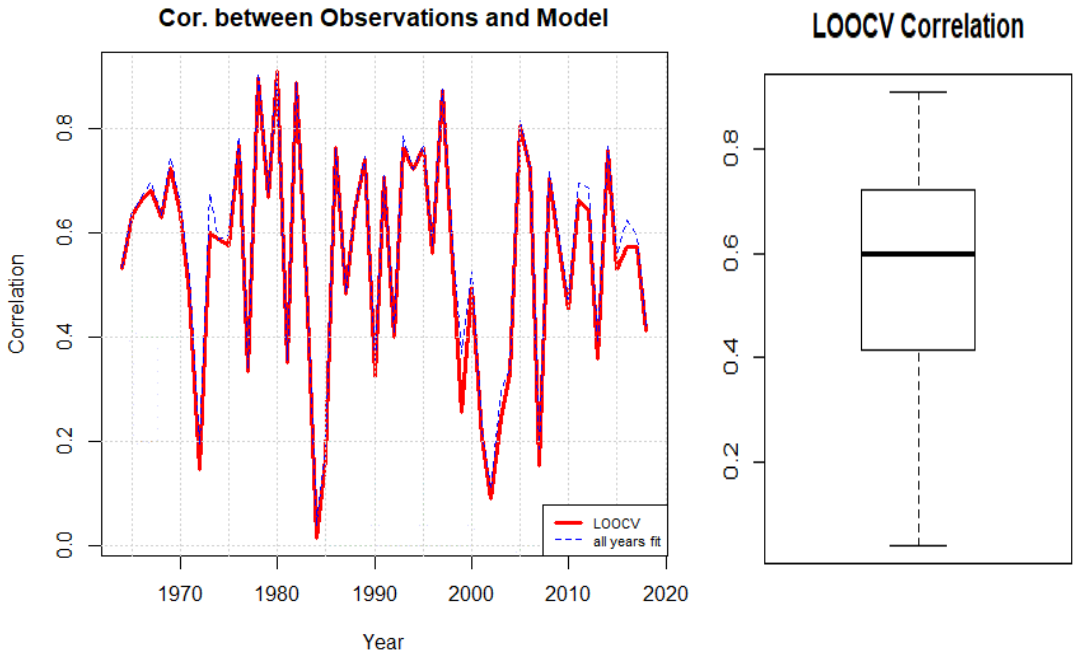

Figure 42: Correlation between observations and the CCA model, shown as a time series such that the prediction is explicitly displayed for each year. The red line shows the LOOCV prediction while the blue line shows model fitted to all values. Comparing the two lines, the predictive skill of CCA is very similar to that of model fit, which indicates the model is robust to overfitting. Correlation ranges from 0.86 (1979 and 1981) to 0.02 in 1984.

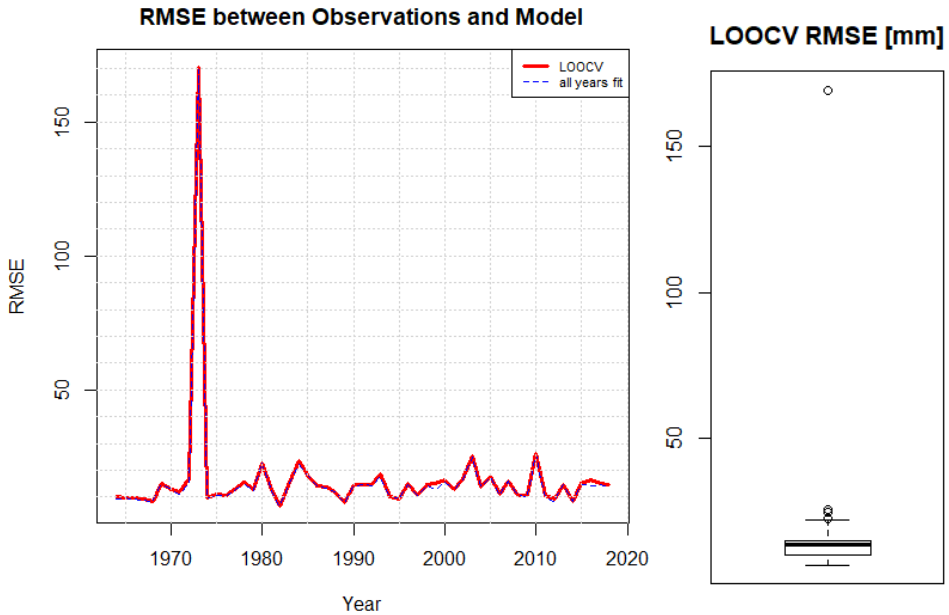

Figure 43: RMSE between observations and the CCA model, shown as a time series such that the prediction is explicitly displayed for each year. The red line shows the LOOCV prediction while the blue line shows model fitted to all values. Comparing the two lines, the predictive skill of CCA is very similar to that of model fit, which indicates the model is robust to overfitting. RMSE ranges from 170 mm in 1973 to 8 mm in 1982.

Conclusions (CCA):

- Compare the results to problem 6: Problem 6 (PCA and CART/random forests) attained correlation as high 0.56 (CART) and 0.43 (random forest), which is similar to CCA values.
- CCA performed much better in prediction according to correlation. Median correlation in LOOCV for CCA is 0.6 (compared to 0.15 in drop10 with random forest).
- Median RMSE for CCA and random forest is similar (near 15 mm). However, CCA has a tighter range of RMSE and less extreme outliers than random forest.