CVEN 6833
HW 3
December 2020
Nathan Bonham

P1 KNN-Conditional Ensemble Forecast/Simulation

1. Use K-NN bootstrapping approach to simulate/forecast Spring streamflow on the Colorado River at Lees Ferry based on suite of predictors, also called, 'feature vectors', at three lead times – Jan 1, Feb 1 and Apr 1.

I used five covariates to predict spring time flow: AMO, PDO, ENSO, tmax_C, and win.pcp_in. Further, I chose to use 300 bootstrap samples when sampling from the KNN, and compared the results of taking the mean and median values. **RPSS is based on terciles of the historical flow**. Results are below:

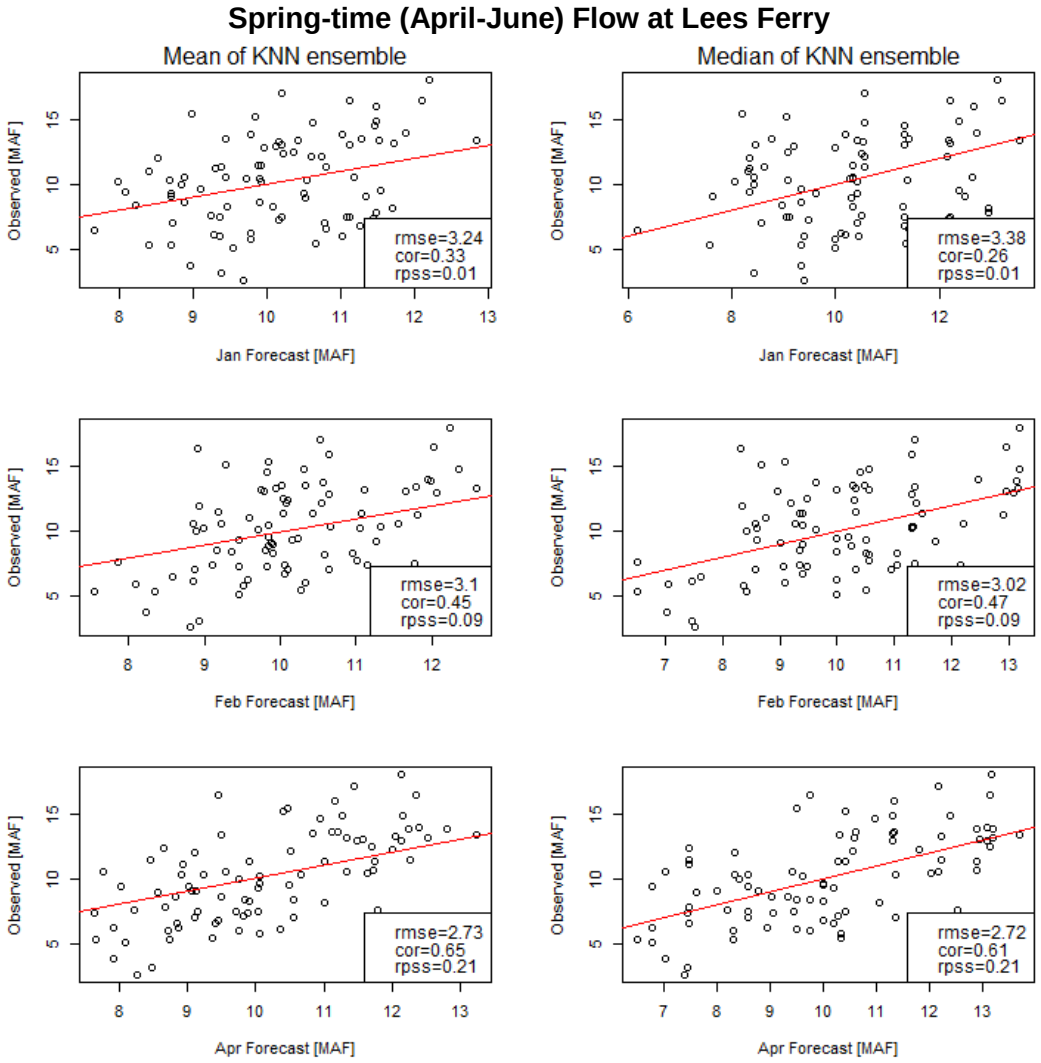

Figure 1: KNN algorithm to predict spring time flow using AMO, PDO, ENSO, tmax_C, and win.pcp_in as covariates. Each row shows the forecast at decreasing lead times (Jan, Feb, April

from top to bottom). Each column shows the forecast using the mean and median of the bootstrap ensemble, respectively. At the January forecast, the mean outperforms the median according to RMSE and correlation. At the Feb forecast, mean and median have similar skill. For the April forecast, the the mean has greater correlation but greater RMSE compared to the median. RPSS does not change from mean to median because this metric considers the ensemble of predictions, not mean or median.

In addition to the above results, I attempted the KNN method using all covariates in the feature vectors plus scaling distances when sorting the neighbors by distance. However, the best results, by far, come from the five covariates unscaled. I made several changes to the KNN and bootstrapping functions provided from Dr. Balaji to allow for these changes, see Appendix.

Conclusions:

- When using all covariates, scaled or unscaled distances, the forecasts are very poor (results not shown, correlation near 0). This suggests the entirety of the feature vector has too much noise.
- Using only five covariates (3 climate indices and 2 weather parameters) unscaled gives the best forecasts according to RMSE and correlation.
- The mean of the bootstrapped KNN ensemble has better performance than the median at Jan lead times. The performace is similar at Feb and Apr lead times.
- According to RPSS, the Jan forecast is about the same as climatology (33% chance probability to belong to 1st, 2nd, or third tercile). This increases to a 21% improvement by April.

P2) Copula Regression

(i) Use copula regression to simulate spring streamflow at different lead times (Jan 1, Feb 1, Apr 1). Use the gcmr package.

The monthly streamflow and feature vectors were truncated such that the year range matched. The resulting years are 1921 to 2006. Then, I plotted histograms of streamflow for spring months April, May, and June to determine the form of their marginal pdfs.

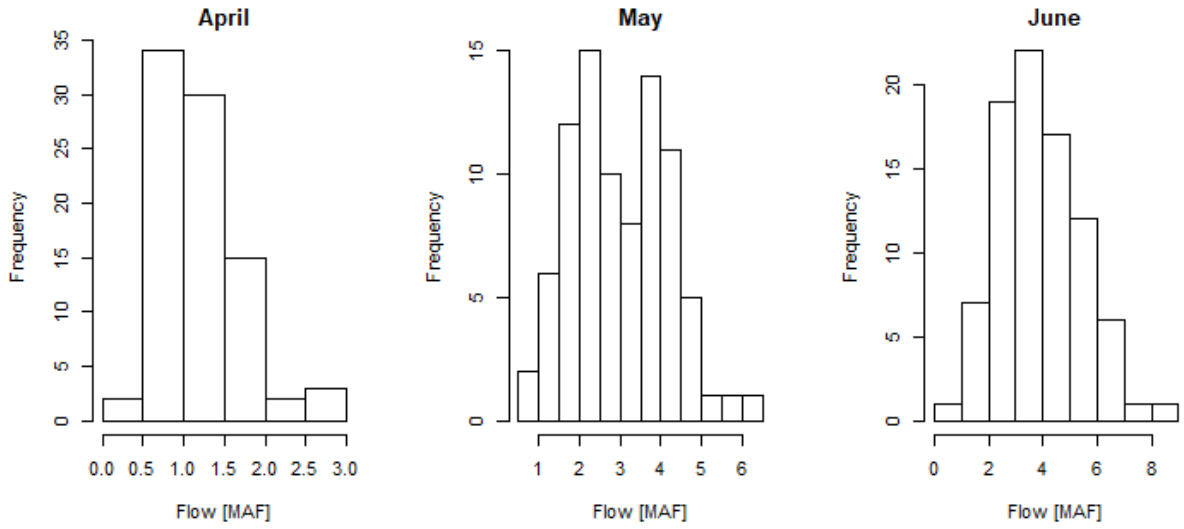

Figure 2: Histograms of spring time flow at Lees Ferry, AZ on the Colorado River. Weibull and Gamma distributions are able to capture the right skewness in April and June, but Gamma distributions were chosen for each month because it resulted in smaller RMSE compared to Weibull when forecasting with the gcmr package. May flows are bimodal and no parametric distribution captures this adequately. Therefore, Gamma distribution was also used for consistency with April and June flows.

Then, I used the gcmr() function to fit a model to each month using the Jan 1 features as predictors. The fitted values were plotted against observed values, then this was repeated for Feb 1 and Apr 1 lead times. The features are AMO, PDO, ENSO, tmax_C, and win.pcp_in.The results are shown in the figure below.

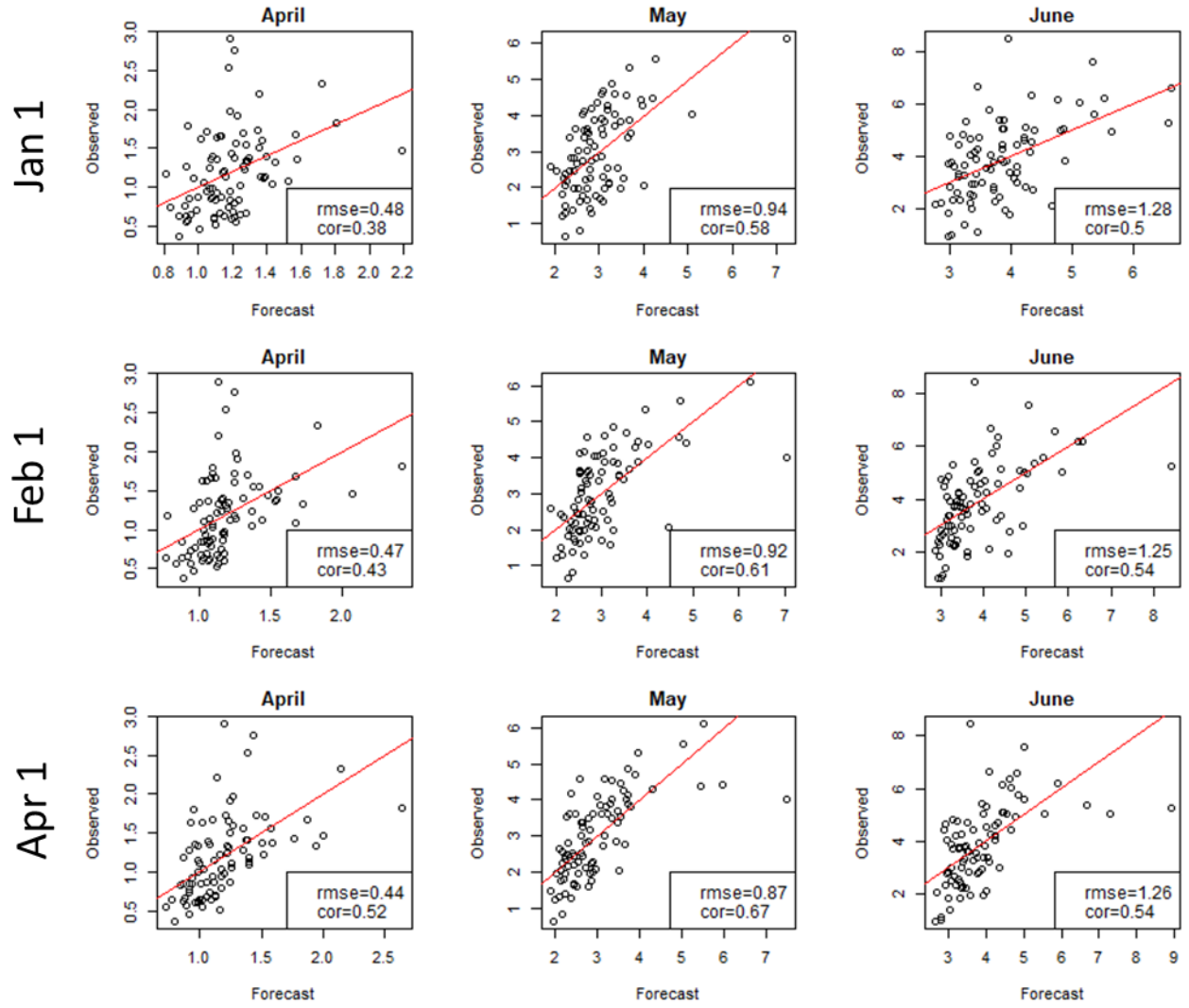

Figure 3: The results of copula regression at different lead times independently for April, May, and June. For each subplot, the y-axis is the observed flow values while the x-axis shows the forecasted flow value (units of MAF). The red line is the 1-1 line (perfect forecast). Each row is the result of copula regression at different lead times (Jan to Apr), and each column is the forecast for the three spring months. To compare the forecasts at different lead times, RMSE and correlation are shown in the bottom right of each figure. The improvements in RMSE as lead time decreases is most notable for May then April flows. The improvement for June flows is negligible. It should be noted that RMSE should not be compared across spring months (across columns) because the magnitude of flows (and thus RMSE) increases from April to June on average. Correlation increases noticeably as lead time decreases for every month except June.

Alternatively, I fit a model to the sum of spring flow instead of to individual months. Below is the histogram of spring time flow. I chose gamma as the marginal distribution.

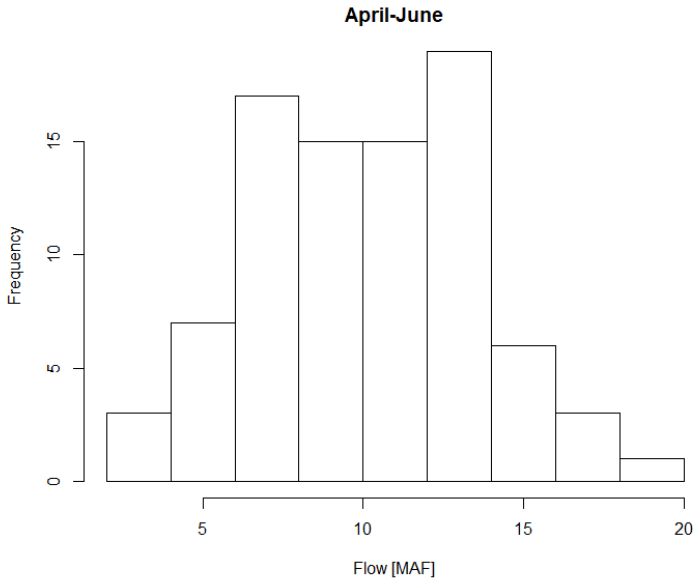

Figure 4: Histogram of the total spring-time flows at Lees Ferry, AZ. Note that summing the flows mitigates the bimodality conundrum present in May flows (although it is still present), and that it is more justified to use Gamma as the marginal distribution.

The same method was used to predict total spring time flow at the three different lead times. Results are below.

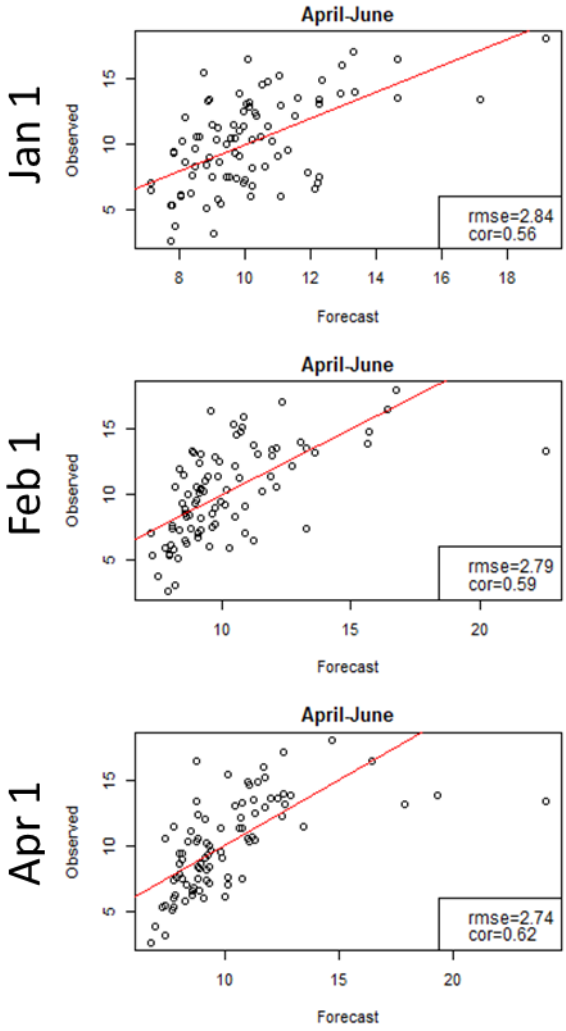

Figure 5: The results of copula regression on total spring-time flow at three different lead times. The y-axis is observed flow, and x-axis is the forecasted flow (units MAF). The red line is a perfect forecast. RMSE decreases for every decrease in lead-time.

Conclusions:

- It is difficult to justify using copula regression on May flows unless a non-parametric distribution capable of capturing the bimodality is used.
- Forecasting total spring-time flows helps alleviate the bimodality concern because the marginal pdf of total spring time flows is better fit by Gamma distribution. Bimodality is still noticeable in the histogram, but it is less profound than May flows.
- When predicting flows for each month separately, RMSE improves the most for May flows and the least for June flows as lead-time decreases. Correlation improves more notably than RMSE as lead time decreases.
- When using copula regression to predict the total spring time flow, the RMSE is greater than 2 MAF. From a reservoir-operations perspective, this might be too much error.
- Copula regression is superior in terms of correlation and RMSE at every lead time compared to KNN method. However, the results are close for April forecasts.

P3) Hidden Markov Model

I fit the HMM model to the spring flow contained in the Apr1features dataframe using gamma distributions. I computed AIC for 1 to 6 states. I was unable to get the built-in AIC function to work on the HMM object, so I wrote my own function, shown below.

```
303  # AIC (or BIC) for HMM
304
305  AIC_hmm=function(hmm, k=2){
306    LL=hmm$LL
307    n=length(hmm$x)
308
309    n_pdf=length(hmm$pm) # number of parameters to fit one marginal distribution
310    n_states=nrow(hmm$Pi) # number of states
311    n_transition=nrow(hmm$Pi)*ncol(hmm$Pi) # number of transition probabilities
312    npar=n_states*n_pdf+n_transition
313
314    return(-2*LL+k*npar)
315  }
```

In my AIC function, LL is the log-likelihood and the number of parameters is the number of states multiplied by the number of parameters for each distribution plus the number of transition probabilities. Using my AIC function, I found that AIC for 1 and 2 state models are very similar (464 and 466, respectively), but models with more states have much higher AIC. I chose a two state model such that I can continue this problem with the logistic regression. The histogram with gamma pdfs for the resulting 2 state model are shown below.

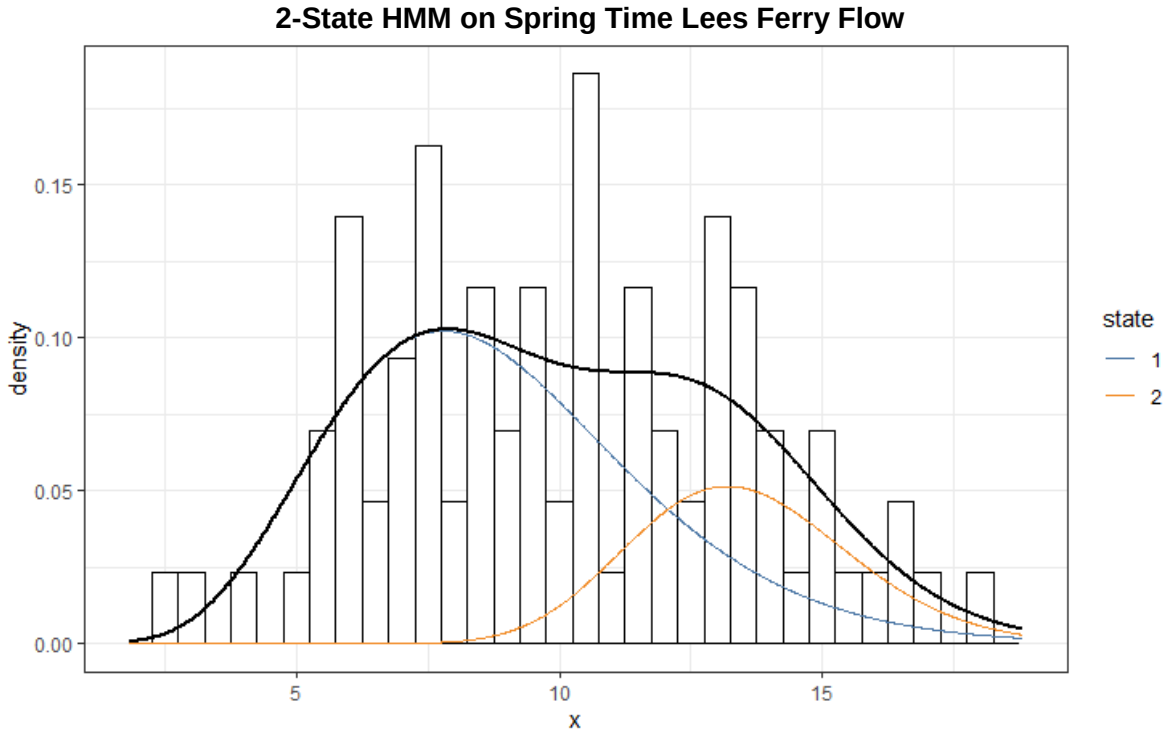

Figure 6: Histogram with pdfs of the 2-state Hidden Markov Model. The x value is spring time flow at Lees Ferry in MAF. The first state is more prevalent (indicated by area under the curve). The second state captures high flows and is less prevalent.

Next, I used the Viterbi() function to decode the state sequence and fit a logistic regression model to the sequence. I tested many combinations of covariates in the April features vector using the leaps() function and chose the best model according to PRESS. A model summary is shown below:

```
> glm_mod=GLM_fit(Y, X, family='binomial')
[1] "Results of PRESS for bestfit GLM"
         glm_press
logit     60.31578
probit    61.42229
cauchit   65.03472
[1] 7
> summary(glm_mod)

Call:
glm(formula = Pm ~ ., family = binomial(link = links[which.min(glm_press)]),
    data = xx)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.5899  -0.4958  -0.2295   0.1190   3.2076

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.0251     2.4866  -4.434 9.26e-06 ***
S_1           2.9200     0.7901   3.696 0.000219 ***
win.pcp_in    1.3487     0.3380   3.990 6.60e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 97.210  on 84  degrees of freedom
Residual deviance: 54.598  on 82  degrees of freedom
AIC: 60.598

Number of Fisher Scoring iterations: 6
```

The best model uses the logit link function and only two covariates. S_1 is the previous state and win.pcp_in is winter precipitation in inches. I used the logistic regression model to obtain the probability of a flow coming from state 1 or state 2. The probability changes in time because S_1 and winter precipitation change every year. Finally, I simulate from the gamma pdfs that correspond to state 1 and state 2 and take the mean and median values. The code is shown below:

```
for (i in 1: nrow(state_probs)){

  sim_vec=rep(NA, nsim)

  for (s in 1:nsim){
    state=sample(states, size=1, prob=state_probs[i,])

    if (state==1){
      sim_vec[s]= rgamma(1, shape=hmm$pm$shape[1], rate = hmm$pm$rate[1])  #simulate from distribution for state 1 with params in hmm object
    } else { # state 2
      sim_vec[s]= rgamma(1, shape=hmm$pm$shape[2], rate = hmm$pm$rate[2])   #simulate from distribution for state 2 with params in hmm object
    }

  }

  p1=sum(sim_vec < Qterc[1])/length(sim_vec)
  p3=sum(sim_vec > Qterc[2])/length(sim_vec)
  p2=1-(p3+p1)
  p=c(p1,p2,p3)

  terc_mat[i,]=p
  Q_mean[i]=mean(sim_vec)
  Q_median[i]=median(sim_vec)


}
```

Finally, I plotted the observed versus forecasted flows. The results are shown below.

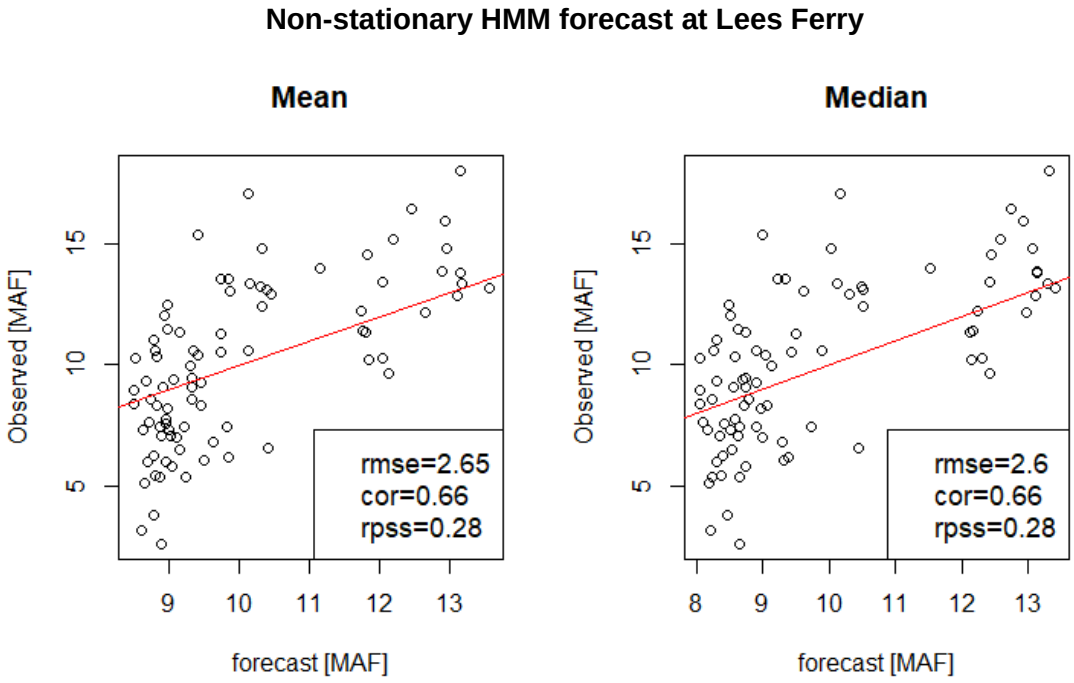

Figure 7: The April forecast when simulating from state 1 and state 2 gamma pdfs according to the state probabilities as predicted with the logistic regression model. This method results in a distribution of flow values, so I have taken the mean and median as predictions. The mean and median perform very similarly, with the median having a slightly smaller RMSE.

Conclusions

- The non-stationary HMM uses two states that represent normal flows and high flows.
- The most important covariate in predicting the occurrence of the high flow state is winter precipitation (besides the previous state).
- The mean and the median of the resulting ensemble predictions have similar RMSE and correlation.
- The non-stationary HMM outperforms KNN and copula regression according to RMSE, correlation, and RPSS.

P4) Perform a space-time nonstationary EVA on the winter 3-day maximum precipitation. Below are the steps:

a) At each location fit a nonstationary GEV model to the 3-day winter maximum precipitation as a function of the 3 covariates – the leading ~3 winter SST PCs. Make only the location parameter of the GEV non-stationary. This will result in 4 coefficients (intercept plus the three covariates)

The only change from Alvaro's code is using the winter max precipitation as the Y variable and the SST principal components as the covariates for the location parameter in the GEV. The code for fitting the GEV to each location is shown below.

```
for (i in 2:dim(data)[2]) { # looping through locations
  # print(i-1)

  fit.gev= fevd(data[,i],datafit,location.fun=~Z1+Z2+Z3,use.phi = TRUE)
  gev_ob[[paste("Est_",i-1,sep = "")]]=fit.gev
  params[i-1,2:dim(params)[2]]=fit.gev$results$par
}
```

Where Z1, Z2, and Z3 are the top three principal components of the SST anomaly data. The Z vectors correspond to years 1964 to 2018, so the location parameter changes as a function of time.

b) Fit a spatial model to each of the coefficients and to the scale and shape parameter. For couple of wet and dry years (you can select the years based on the average spatial 3-day precipitation). Estimate the 2-year, 50-year and 100-year return levels at each grid point and map them. Compare with 2-year return levels with the observed values in the selected years

When taking the average across all locations, the wettest year is 1973. In this year, the Park City, Utah area received a storm of roughly 5 feet. The driest year is 1972. The plots of the median 2, 50, and 100 year return level winter precipitations are shown below.

Wet year (1973)

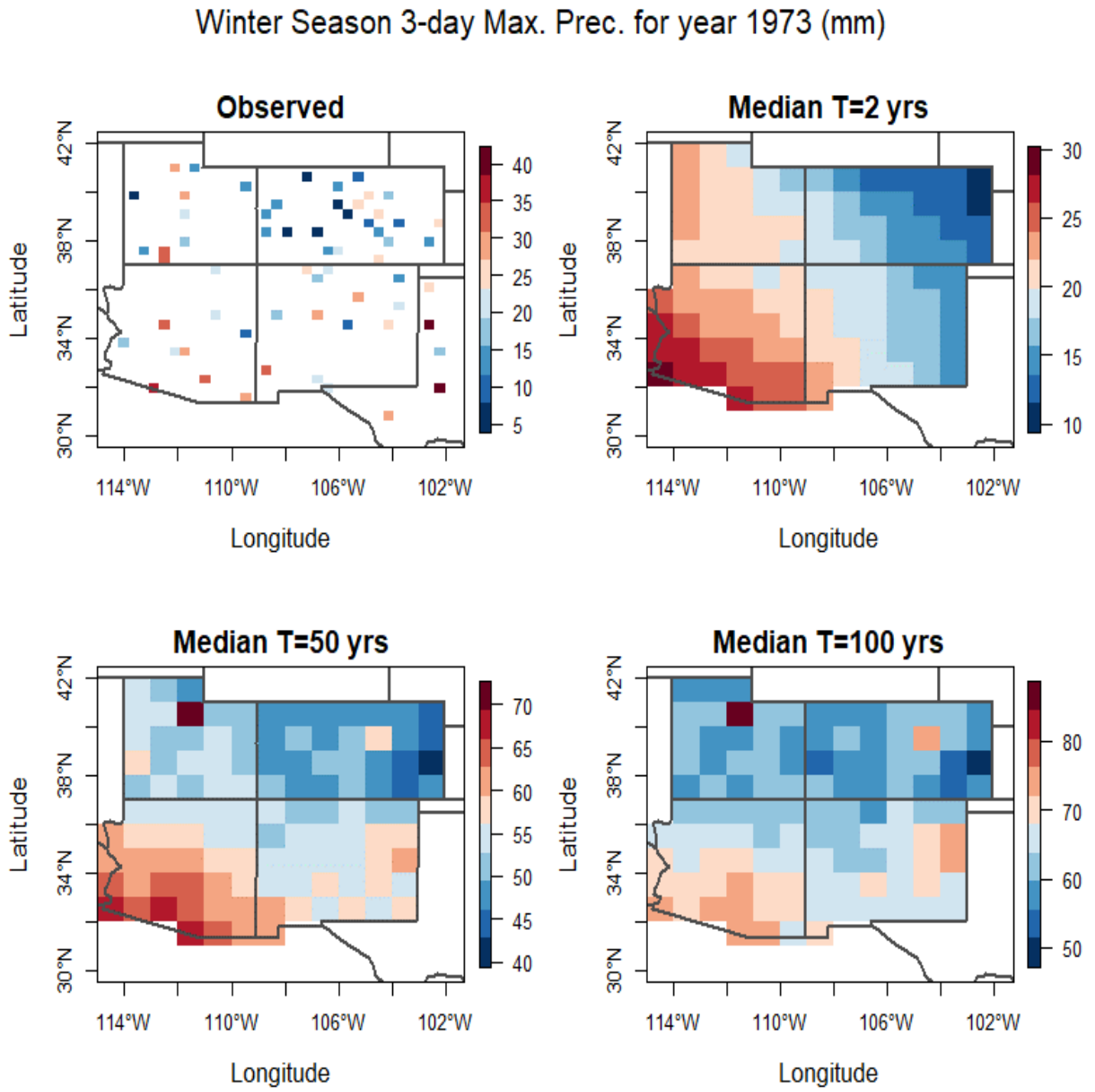

Figure 8: Return levels for the wettest winter in the record. Note that snowfalls above the 90th percentile were removed from the observed plot to achieve variation in the color scale for other locations. Much of the observed precipitation in Colorado was close to a 2 year return level. The largest snowfall (roughly 1500 mm in Utah) greatly exceeded the 100 year return level. The location of the largest snowfall corresponds to the dark red/brown grid in the T=50 and T=100 plots.

Dry year (1972)

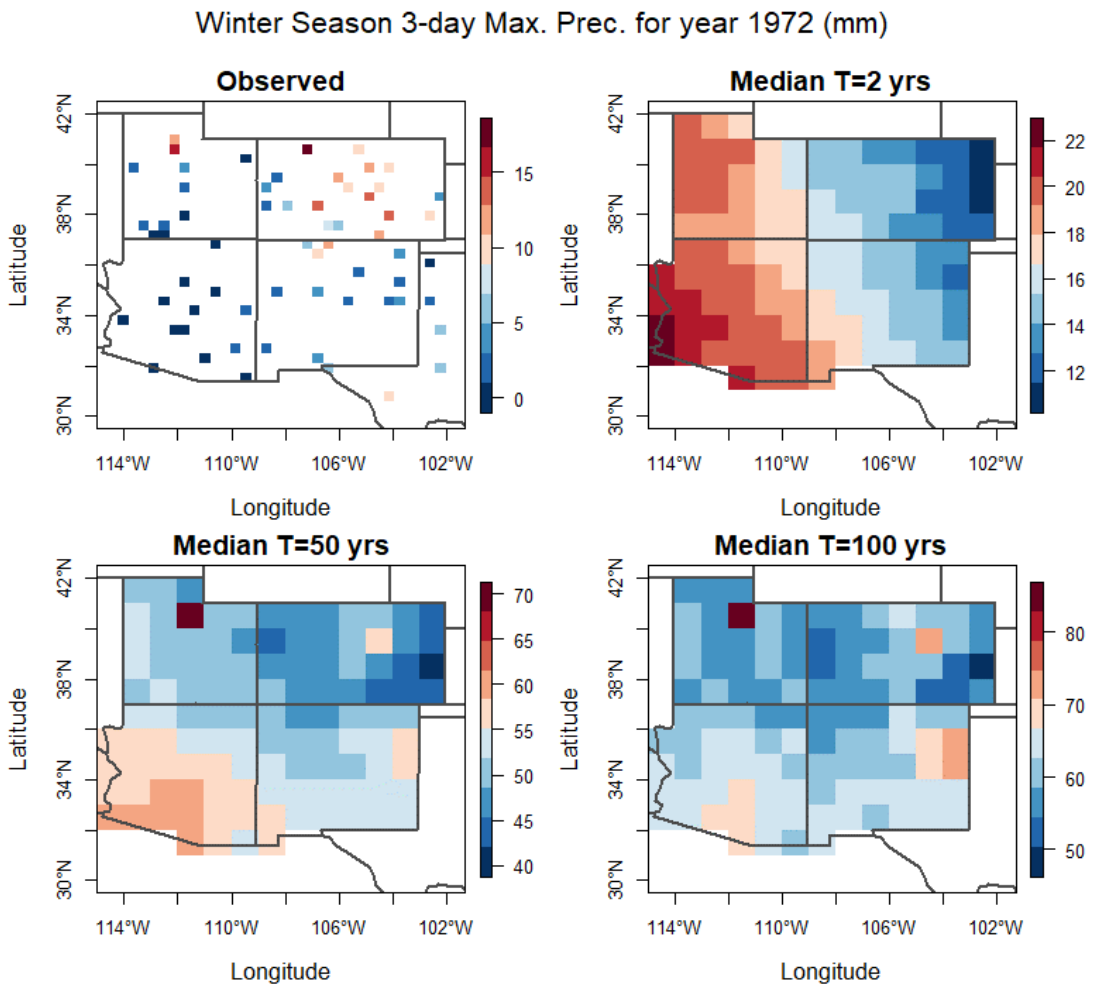

Figure 9: Return levels for the driest winter in the observed record. The mountain region of Colorado and the Park City, Utah area experienced snowfall near the 2 year return level, but most everywhere else experienced snowfall below the 2-year level.

c) For a couple of representative locations plot the time series of 3-day precipitation maximum along with the time varying return levels. Compare them with the stationary return levels

I chose three locations: mountainous Utah near the Wasatch front, central Arizona, and mountainous Colorado. The locations are shown below:

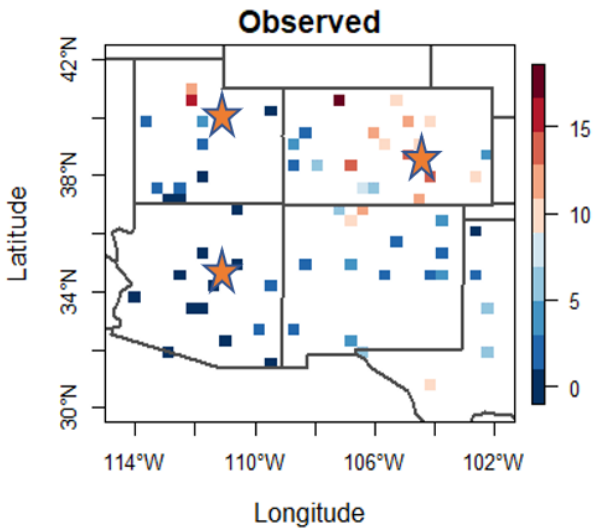

Figure 10: The three representative locations are: mountainous Utah, central Arizona, and mountainous Colorado.

I used the function qevd() in the extRemes package with the scale, shape, and time variant location parameter as fitted earlier in the code with fevd() function. See the code below.

```
locs_T=list() # preallocate a list to store data frame of return levels

c=1
for (i in locs){ # loop through locations
  work.df=T.df
  p=params[i,-1]

  for(t in 1:nrow(datafit)){ # loop through years

    Xloc=datafit[t,-1]
    loc=p[1]+sum(Xloc*p[2:4])
    scale=exp(p$sigma)
    shape=p$xi

    for(z in 1:length(rp)){ # loop through return levels
      prob=1/rp[z]
      work.df[t,(z+1)]=qevd(prob, loc=loc, scale=scale, shape=shape, type='GEV', lower.tail = F) # place values in dataframe
    }
  }
  locs_T[[c]]=work.df # store data frame in a list for each location
  c=c+1
}
```

The resulting non-stationary return levels are shown below for each location.

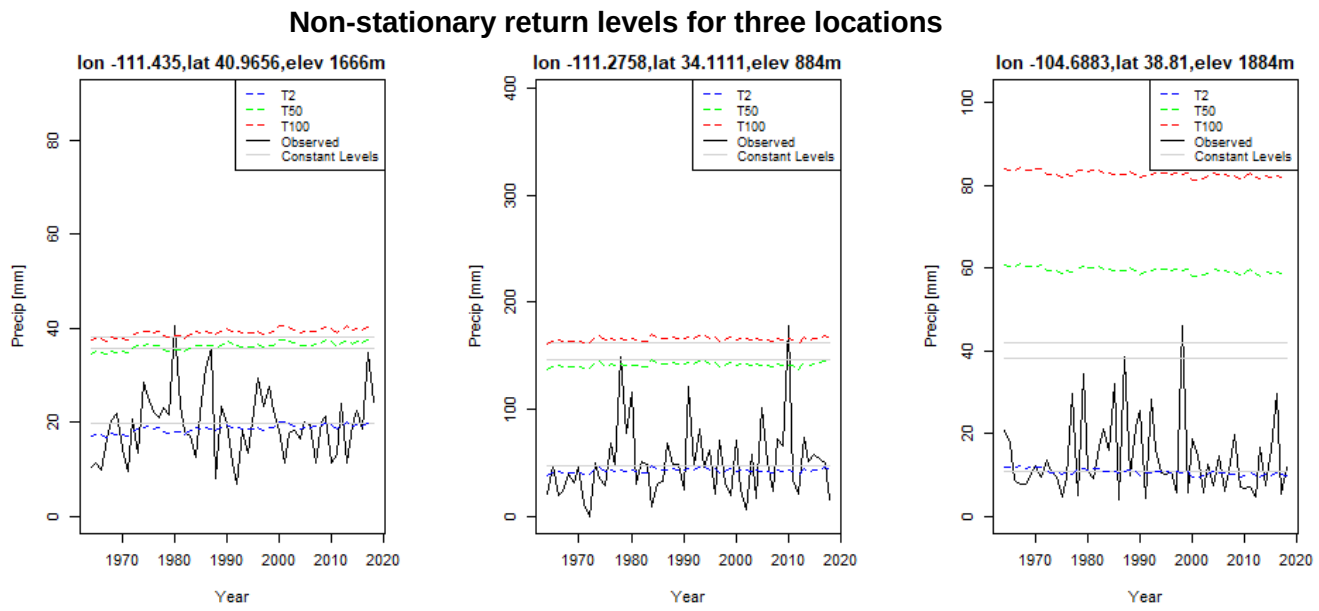

Figure 11: Non-stationary return levels for mountainous Utah, central Arizona, and mountainous Colorado (from left to right). Non-stationary levels are shown in blue (2-year), green (50-year), and red (100-year). The empirical return levels are shown as light grey lines, and observed values are shown in black. For the first two locations, the empirical return levels are close to the non-stationary levels. For the third location, the 50 and 100 year model results in much larger return levels. The first location (Utah) shows a positive trend for each return level.

Conclusions:

- Comparing the return levels for 1972 (dry year) to 1973 (wet year), the 2 year levels increased by ~5 to 10 mm across most of the spatial domain.
- The non-stationary return levels are relatively stable (no large changes year to year), but they are able to show trends in changing risk levels, as is the case for the Wasatch Front area in Utah.
- The GEV model results in very large 50 and 100 year return levels for the mountainous region of Colorado compared to the empirical 50 and 100 year levels.