Predicting Droughts in the Amazon Basin based on Global Sea Surface Temperatures

Dario Lepke

Contents

Introduction

With future climate change droughts in the Amazon forest may become more frequent and/or severe. Droughts can turn Amazon regions from rain forest into savanna, leading to high amounts of carbon released into the atmosphere. Therefore, predicting future droughts and understanding the underlying mechanisms is of great interest. Ciemer et al. (2020), established an early warning indicator for droughts in the central Amazon basin (CAB), based on tropical Atlantic sea surface temperatures (SSTs). In my thesis I would like to build on this work and improve the predictive power by using different statistical methods. Meaning, we seek to build a model that is able to predict droughts (resp. rainfall) based on preceding sea temperatures, desirably with as much lead time as possible. Also we want to identify those sea regions that are most important for doing so, making interpretability a point of interest, too. A first model could be a cross-validated (generalised) LASSO approach trying to identify the most important oceanic regions and the respective time-scales.

The thesis will be done in cooperation with Dr. Niklas Boers from the Postdam Institute for Climate Impact Research (Climate Impact Research (PIK) e. V. (2021)).

Chapter 1

Related work

As already mentioned the paper by Ciemer et al. (2020), created an early warning indicator for Amazon droughts. They did so using a complex network approach. They used two datasets, one for the Sea Surface Temperatures (Smith et al. (2008)) and one for the precipitation (Funk et al. (2015)) with monthly data for the time period of 1981 until 2016. The data can be downloaded for example in netcdf format and manipulated conveniently with Climate Data Operators (CDO, Schulzweida (2019)). CDO in turn can be used with wrappers for R and Python. The data is organized on a longitude/latitude grid.
They identify 4 oceanic regions that correlate the most with rain in the amazon basin, using a coupled network approach. Figure 1 shows the cross degree towards rainfall in the central Amazon basin (CAB, blue box), for positive and negative correlations. Darker shades indicate a larger cross degree, hence a larger number of links and correlations with rainfall at more grid points in the CBA.
The correlations are measured using a spearman rank-order correlation coefficient

$$\rho = 1 - \frac{6 \sum \Delta_{R_i}^2}{n(n^2 - 1)}. \tag{1.1}$$

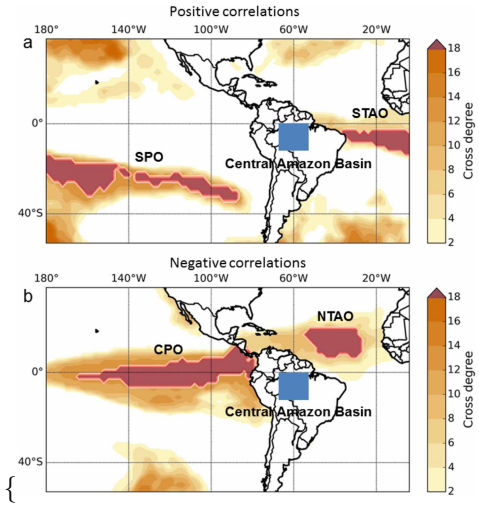
Where Δ_{R_i} denotes the difference between the ranks of observations of both variables at the same time i and n is the number of observations. An Adjacency Matrix describes the resulting network, where the threshold p_{th} was chosen so that only 10% of the strongest correlations are represented as links in the network

$$A_{ij} = \begin{cases} 0 \; p_{ij} < p_{th} \\ 1 \; p_{ij} \geq p_{th} \end{cases} . \tag{1.2}$$

The cross degree then gives the strength of correlation between a specific grid point i of network V_l (oceanic grid point) and another (sub)network V_m (all grid points j in central amazon basin)

$$k_i^{lm} = \sum_{j \in V_m} A_{ij}, i \in V_l \ . \tag{1.3}$$

\begin{figure}



\caption{"Cross degree between sea surface temperature and continental rainfall anomalies. For each sea surface temperature grid cell of the Atlantic and Pacific Ocean, the cross degree towards rainfall in the Central Amazon Basin (blue box) is shown, for a positive correlations and b negative correlations. Darker shading indicates a larger cross degree, implying a larger number of links, and thus significant correlations with rainfall at more grid points in the Central Amazon Basin. Red areas outline coherent oceanic regions with a the 20% highest cross degrees for positive correlations, found in the Southern Pacific Ocean (SPO) and Southern Tropical Atlantic Ocean (STAO), and b the 20% highest cross degrees for negative correlations, found in the Central Pacific Ocean (CPO) and Northern Tropical Atlantic Ocean (NTAO)" (Ciemer et al. 2020)} \end{figure}

They further explore the relationship by constructing (weighted) networks for sliding windows of 24 months between the Central Amazon Basin and each of the ocean regions. For each month except for the first two years, an individual network is computed based on the data of the previous 24 months. Then they take the average of the cross correlations for each of the networks which gives a new time series of average cross correlation (ACC) values. Each ACC summarizes the connectivity of one region with the CAB for the last 24 months.

They find that NTAO and STAO give the strongest signal, hence they apply the same sliding window coupled network approach between the ocean regions NTAO and STAO. Before they computed networks between ocean and continental regions, now it is computed between these two Atlantic regions, NTAO and STAO.
The resulting time series and its comparison to the drought index time series is shown in figure 1.1 below. They find that using a ACC threshold for a drought (SPI below -1.5), lets them forecast 6 of the 7 droughts in the observation period, missing the 2005 drought, while also giving one false alarm in 2002.

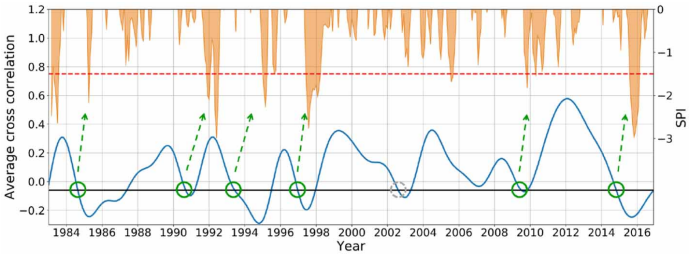


Figure 1.1: "Early-warning signal for droughts in the central Amazon basin. We compare the time evolution of the average cross correlation of the Northern Tropical Atlantic Ocean (NTAO) and Southern Tropical Atlantic Ocean (STAO), given by the blue curve, with the standardized precipitation index (SPI, orange) of the central Amazon basin. Orange dips indicate a negative SPI with a threshold for severely dry periods (SPI -1, dotted red line). We expect a drought event within the following one and a half years whenever the average cross correlation between NTAO and STAO SST anomalies falls below an empirically found threshold of -0.06. Green circles indicate a matching forecast based on the Atlantic SST correlation structure, with one false alarm in 2002 indicated by a grey circle, where the threshold is crossed but no drought took place in the direct aftermath (see Discussion). The temporal evolution of the average cross correlation shown here is smoothed using a Chebyshev type-I low-pass filter with a cutoff at 24 months" [@ciemer2020early].

The work by Ciemer et al. (2020) shows potential forecasting capabilities and limitations. While they are able to predict 5 out of 6 drought events, they also give one false negative and one false positive result. Their work uses a complex network approach that is applied stepwise (first two unweighted networks, then two weighted networks and in the end a dichotomous threshold decision rule). For the thesis we would like to create a more general predictive model that can learn the relationship between the SSTs and rainfall in the CAB. As already mentioned a first step can be a LASSO model. First findings show that, the classic LASSO only chooses single points in the ocean as predictors, though. But our motivation is to discover predictive regions and not only single separated points. Therefore in a next step we want to make use of a generalized form of the LASSO that also takes into account that chosen

predictors should be close to each other. This model is the so called Fused LASSO.

For the models we also need a form of evaluation. Classic Cross Validation assumes independence of the observations. In our setting this is clearly violated due to the time dependency of the data. We will explore different possibilities to use an adjusted form of Cross Validation that takes this characteristic into account.

Depending on how well the relationship between SST and rain can be established, we can take this a step further and use it as a so called "Emergent Constraint" (EC). Since different climate models give different answers about future climate there is a need to narrow this "spread", which can be done by ECs. To do so, we need a plausible relationship between a Variable X and Y (here: SST and drought).

According to how well the relationship is represented in a climate model we assign "credibility" to a climate model's future projections (here: projections of future droughts in the Amazon rain forest). In summary this can be used to reduce uncertainty in the ensemble of climate models' future projections, f.e by using ML techniques as done by Schlund et al. (2020).

Chapter 2

EDA

In this chapter we will inspect the values of the precipitation and SST data for the common observation period from 1981 until 2016.

2.1 EDA precipitation

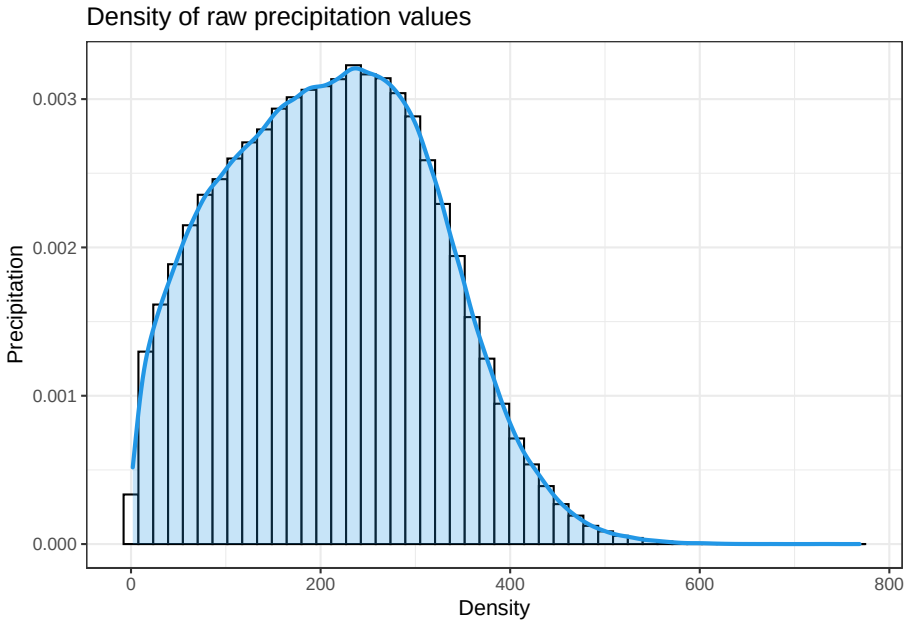
In this section we want to study the time series of precipitation in the Central Amazon Basin. The CHIRPS data set contains the **precipitation data**, created from in-situ and satellite measurements (Funk et al. (2015)). It can be downloaded for example from here [https://www.chc.ucsb.edu/data/chirps]. It contains observations **from 1981 to 2016** and comes on a **high resolution of 0.05 grid**, which we aggregate to a 0.5 grid.

Below the area of the Central Amazon Basin that is object of our study.

We will now inspect the precipitation values from three perspectives. Firstly the raw values without time or spatial dependency, second the mean and standard deviation for each spatial grid cell for the whole time series and then the mean and standard deviation for each grid cell but for each month of the year separately.

Firstly we plot precipitation values in general

Figure 2.1: Location of the area under study. The central amazon basin (CAB) spanning across 0,-10 latitude and -70,-55 longitude



Its form is a uni-modal, right-skewed density. The values range from 0 up to 769, but only few observations take these high values, forming a large tail. This might be a indication for large outliers in the data or due to some locations with very high precipitation values in general.

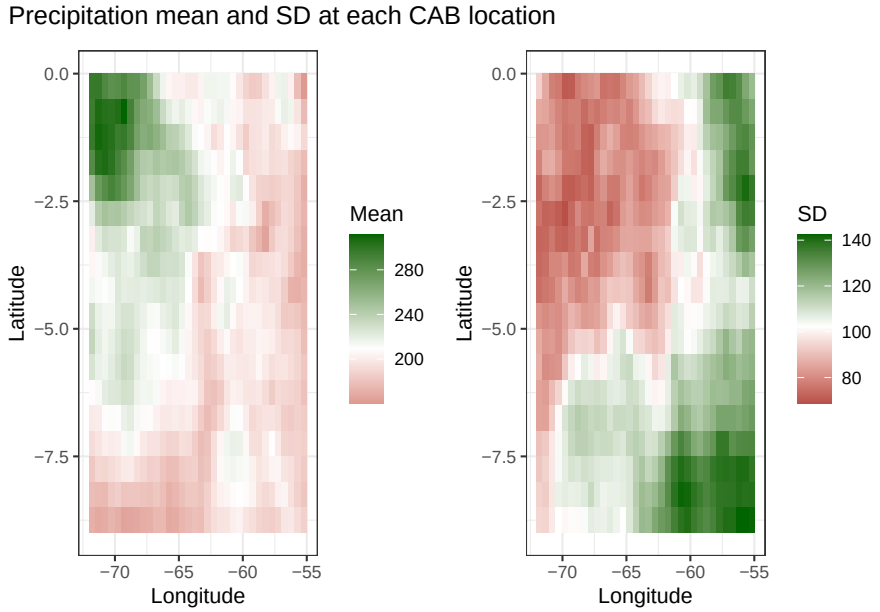
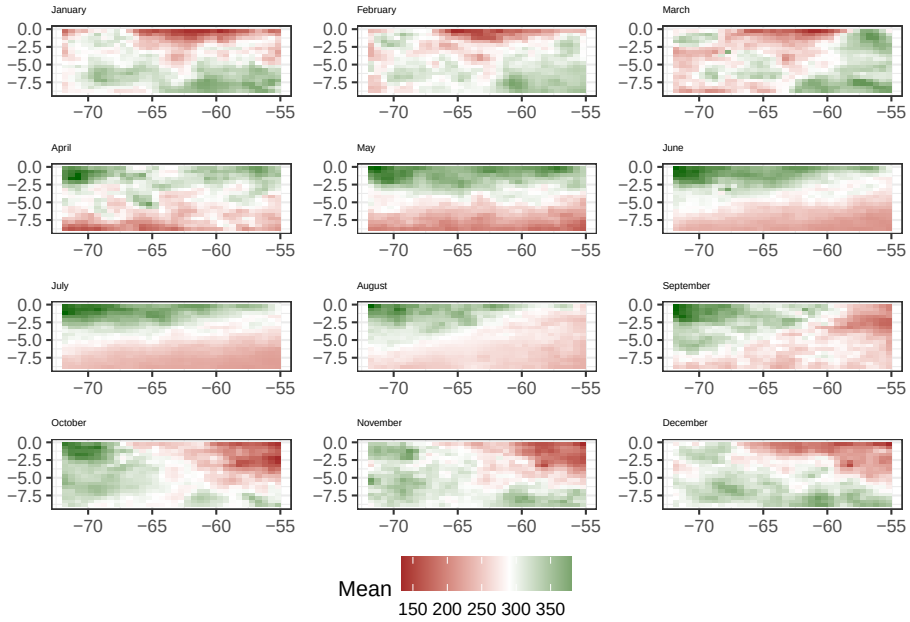


Figure 2.2: Mean and standard deviation at each location. The standard deviation was computed over the whole time period. The white line on the scale at the side of the plots indicates the mean of the respective quantity

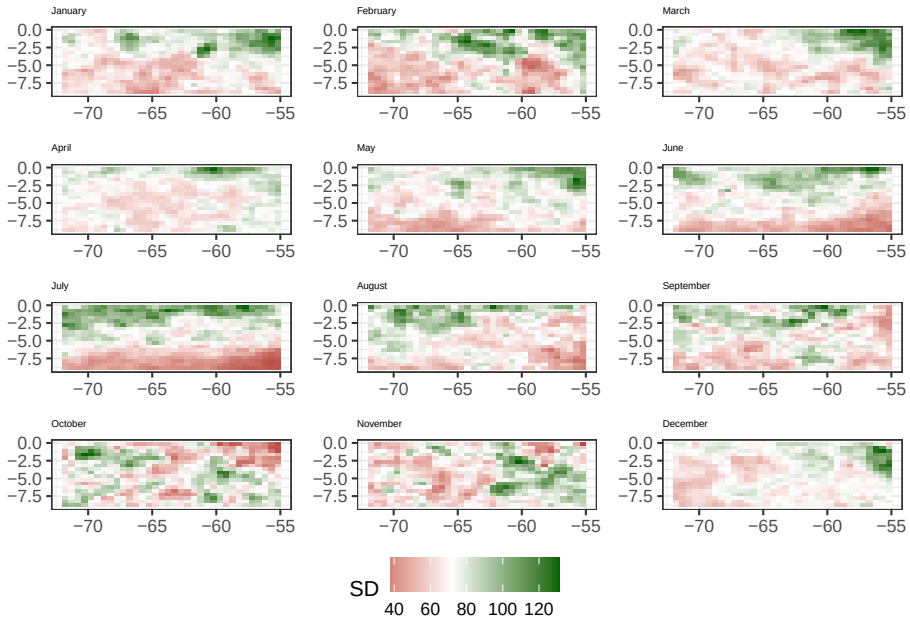
As we can see most locations have a mean precipitation of around 200 mm/month, over the whole time series. Regionally in the "upper left" corner of the Amazon Basin, mean precipitation is higher or equal to the mean. The reference point for "higher" is the mean of the location means. This region seems to be more or less spatially consistent. The rest of the region with lower mean precipitation has also some small areas where precipitation is again a little bit higher. For example in the upper right corner and on the bottom, right of the middle.

For the standard deviation we also see regional patterns. These patterns overlap with the regions of the mean but their magnitude is flipped. Meaning, in the upper left where we observe larger mean values we generally observe lower standard deviation and in the lower and upper right corners, higher standard deviations.

separately



We see spatial patterns of the mean evolving over time. For example: From May until August there is a spatial separation in two parts that dissolves in September. As expected there is a large seasonal component regarding the means.



For the standard deviation we see as well large differences in values during different months of the year.

2.2 Glyph plots

This section provides a graphical presentation of the precipitation data known as glyph plots. The idea of glyph maps, its application and general implementation that were used in this section are taken from @**??**wickham2012glyph. Glyph maps use a small icon or *glyph* to show multiple values at each location. In our case, we show a complete time series at each location instead of just single values. Different techniques can then be used compare the time series between all locations or their individual shape on a local scale. We will show seasonal, de-seasonalized, and de-seasonalized data on a local scale. Seasonal time series are computed by computing the averages of each month on each location. Each seasonal time series therefore has only 12 values and can be plotted without smoothing. The de-seasonalized time series are computed by omitting the seasonal effects on each time series for the *complete* observation period and therefore has to be smoothed to be visually inspectable. The de-seasonalized time series then can be used to compare the time series for each location on a common or local scale. On the common scale all values are displayed on the same axis range, while on the local scale the axis are changed so that their ranges refer to the range on the respective location. Rescaling is done as follows

$$x_{rescaled} = \frac{x - min(x)}{max(x) - min(x)}.$$

This will help us to see the changes in value at each location *relative* to the range of the values at the same location. But this also means that interpreting these plots has to be done carefully because, in this form of display, large difference might actually refer to only small changes in absolute values. It can be due to the small range of values at that location in general, that these changes seem to be large. To aid the interpretation of these plots we can use color shadings to draw attention to areas in which ranges are large, meaning larger differences in their relative values also point to larger differences in their absolute values (i.e unscaled values, values on the global scale). Therefore locations with large ranges are shaded in lighter colors and smaller ranges are shaded in dark color, to make the lighter shaded areas more easily visible.

To improve readability of glyph maps, one can also add boxed for each glyph as well as reference lines for global means. This way the trajectory of the glyphs can be viewed in comparison to the overall mean directly.

The above figure is a glyph-map of seasonal precipitation patterns (averages for each month) in the Central Amazon Basin. The gray reference lines show

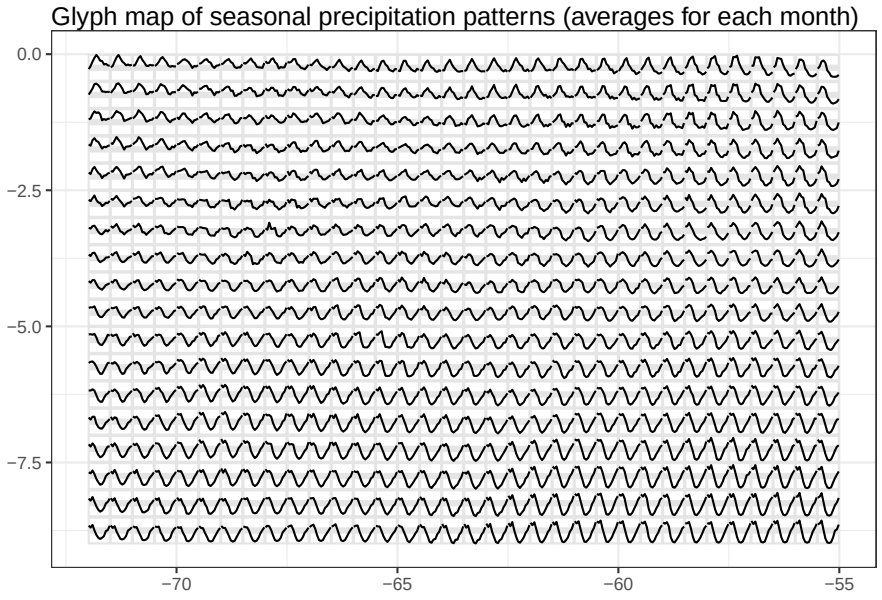


Figure 2.3: Glyph map of seasonal precipitaton pattern. Each location is presented by a time series. The time series are seperated by boxes. The gray reference lines inside the boxes show the mid-range for easier comparison.

the mid-range for easier comparison of the patterns. We see differences in the seasonal patterns across the map. In the upper left for example, the seasonal patterns stay above mid-range while on the bottom-left they have values clearly towards the low end of the range. Also some areas have multimodal patterns. The patterns differ in range and month of maximum and minimum precipitation.

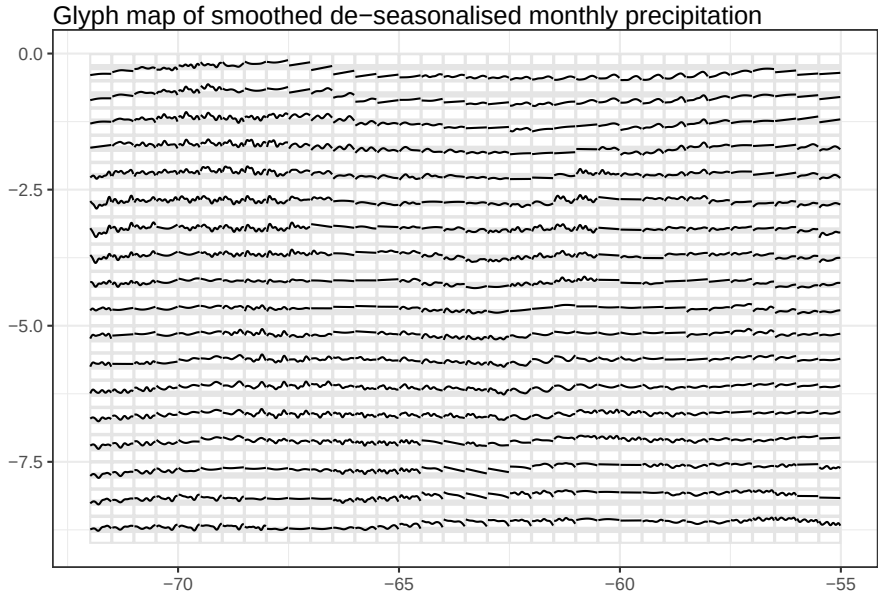


Figure 2.4: Glyph map of de-seasonalised and smoothed precipitation. Each location is presented by a time series. The time series are seperated by boxes. The gray reference lines inside the boxes show the mid-range for easier comparison. The time series are scaled globally, same positions inside the cells correspond to the same values in all locations.

This plot shows the smoothed de-seasonalized monthly precipitation, after global scaling. The same position within each cell corresponds to the same value in all locations. Some areas have almost a linear course, increasing, decreasing or constant. Others show a more "wiggly" courses. As overall pattern we can see that the forms of the patterns have a spatial connection, patterns are close to similar patterns, at the same latitude. Also regarding latitude the closer to the equator the less precipitation.

Now we inspect the glyph-map with de-seasonalized locally scaled values. This form of scaling emphasizes the individual shapes. Because of the applied scaling, big patterns may be just be tiny effects. Therefore colors are added according to range. Areas with lighter color have larger ranges than darker areas. The areas with steep linear increases and decreases have smaller ranges

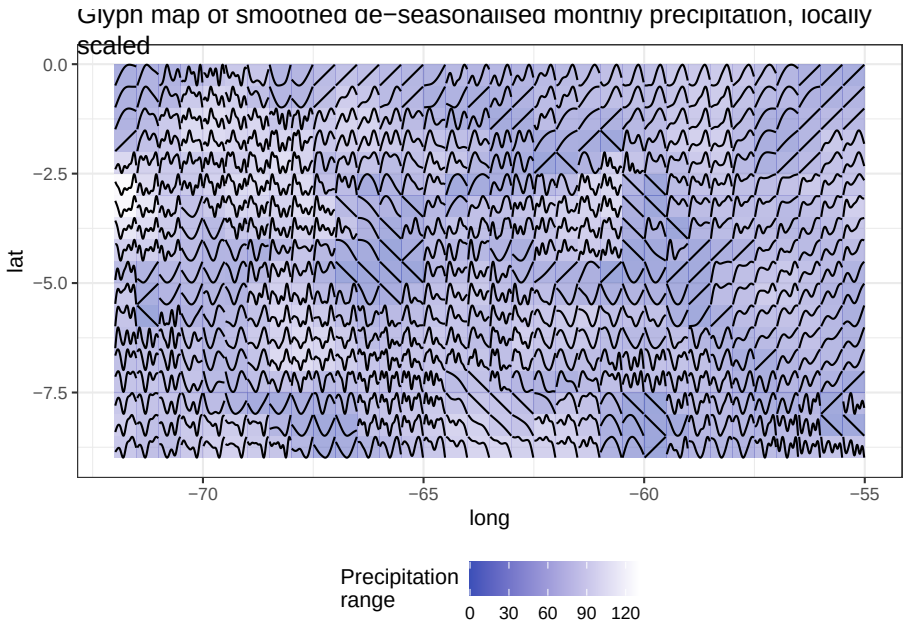


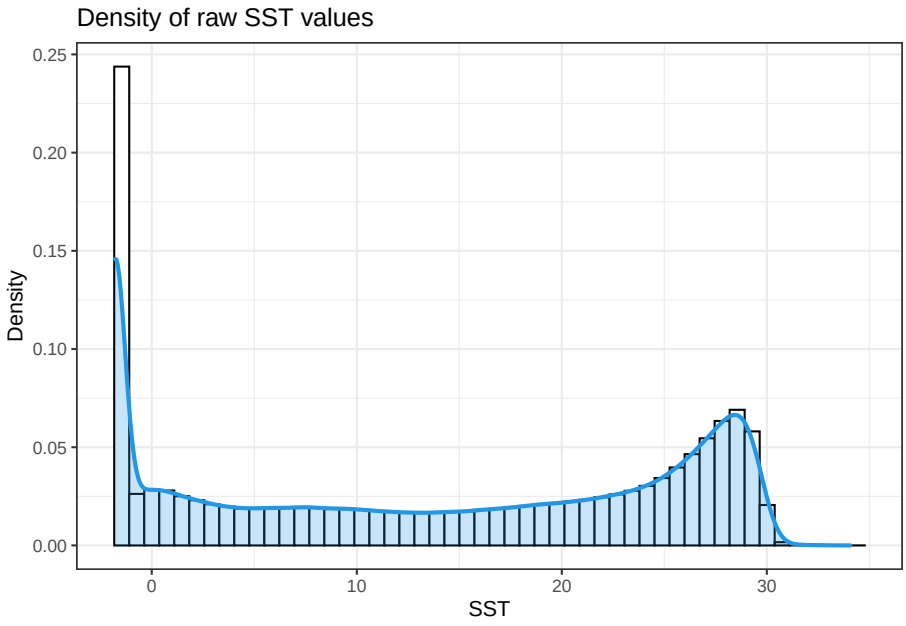
Figure 2.5: Glyph map of de-seasonalised and smoothed precipitation. The time series are scaled locally, ranges are not the same in all cells. The different ranges are given in color shades, where lighter shading indicates a larger range and darker shades smaller ranges.

than or example the areas below -2.5 latitude in the left.

The results of the precipitation glyphs indicate that the CAB might be separable in different regions. If we can find a way to quantify the differences in these regions and separate them into clusters, we could then apply our regression models to each of these clusters and eventually improve model accuracy on each region as compared to the complete are on average. Therefore in a later section we will discuss and apply clustering algorithms to the precipitation data. But for now we will have a look at the SST data.

2.3 EDA SST

We explore the sea surface temperature data set, used in the paper by Ciemer et al (Ciemer et al. (2020)). ERSST (Extended Reconstructed Sea Surface Temperature, Huang et al. (2017)) is a reanalysis from observed data given in the International Comprehensive Ocean-Athmosphere Data Set (ICOADS). Which contains observations *from 1800 until 2016*, made by ships and buoys for example. The data comes on a 2x2 degree grid, where data was missing interpolation techniques were used. See paper for reference. the file contains two variables that are measured across different dimensions. The two variables contain the sea surface temperatures and the respective SST anomalies (with respect to the 1971-2000 monthly climatology). Here we analyze the raw SST values, since the anomalies are computed to a climatology that spans a time frame we will use in the analysis. Using the anomalies therefore would eventually introduce information leakage because during the training process future values were used for fitting the model.



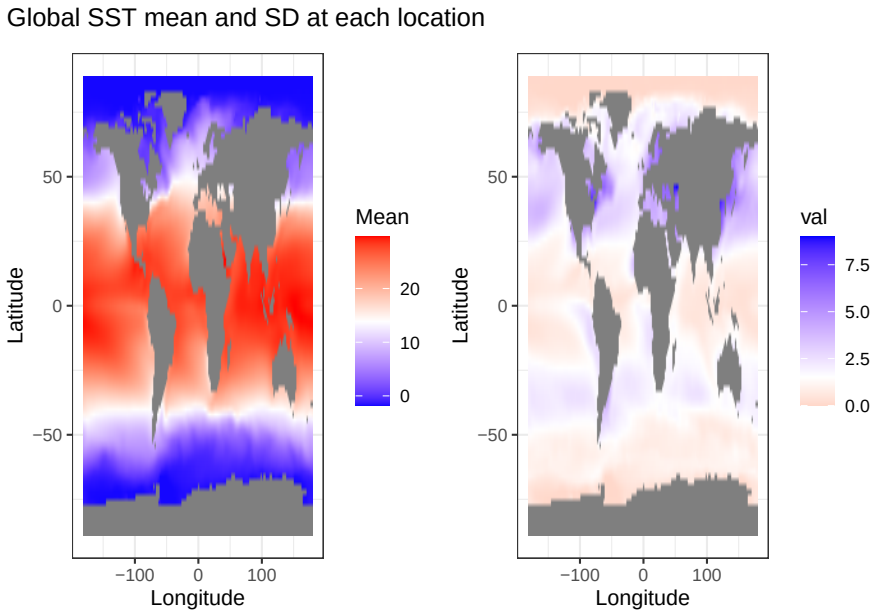
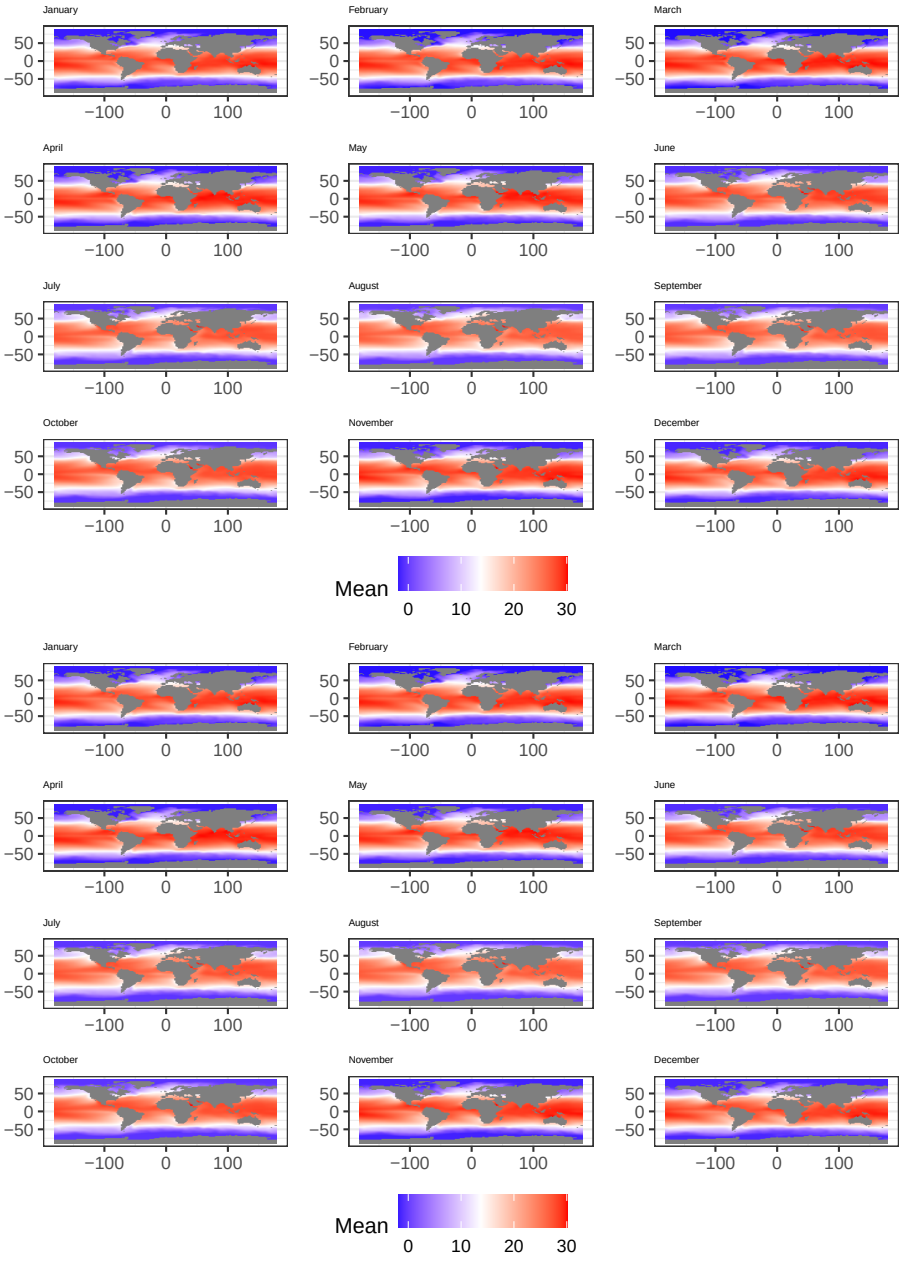
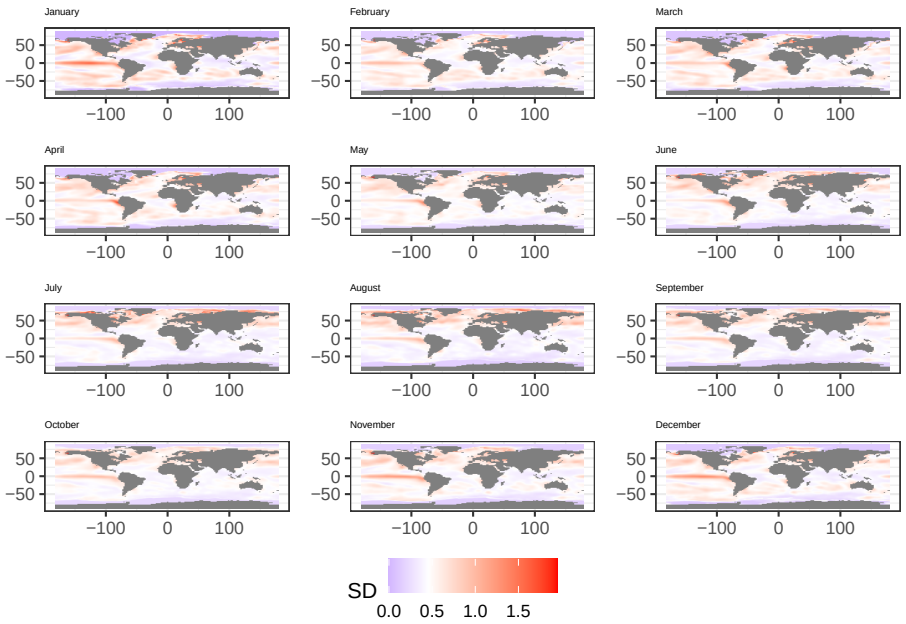


Figure 2.6: Mean and SD on the spatial map.



We see spatial patterns of the mean evolving over time. For example: From May until August there is a spatial separation in two parts that dissolves in september. As expected there is a large seasonal component regarding the means.



For the standard deviation we see as well large differences in values during different months of the year.

Chapter 3

Correlation analysis

3.1 Short Recap

We give a short overview over the correlation between monthly sea surface temperature and monthly mean precipitation in the Central Amazonas Basin (CAB). First we will analyse the original and then the deaseasonalised data. SST and precipitation data have been deseasonalised, meaning first each time series was decomposed by the stl algorithm according to

$$Monthly\ Data = Seasonal + Trend + Remainder$$

Afterwards only trends and remainders time series were kept to constitute a new time series that will be used as predictor (sst) and target (precipitation).

In a next step we compute the correlations between each sst grid point time series and the mean precipitation time series. Since our goal is to predict the precipitation on the sst information, we are also interested in predicting future precipitation some months ahead. To examine this we also compute the correlations for different time lags. For example we might use January sst data to predict precipitation in June, given a time lag of 6 months.

We consider time lags of 0,3,6 and 12 months. And show the density of the correlation values as well as their spatial distribution on a map. We also display the highest positive and negative correlation based on their respective 2.5% and 97.5% quantiles. All correlations that are between these values are set to 0 then.

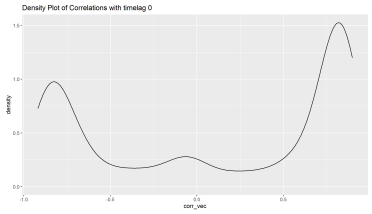
The correlation measure we use is the

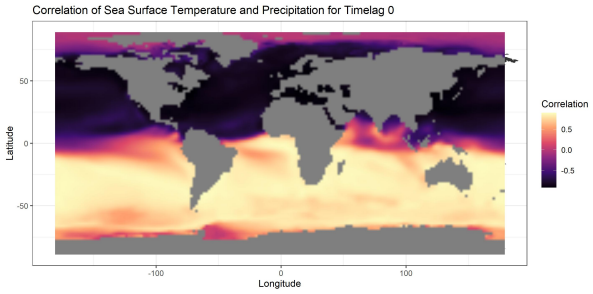
3.2 Correlation of Sea Surface Temperature and Precipitation

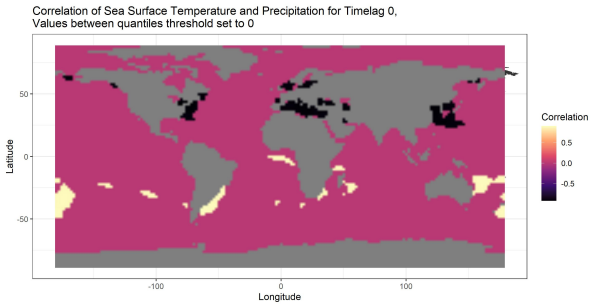
3.2.1 Original Data

Following, for each timelag we show the respective density of correlation values, their location on the map and also the 5% strongest positive and negative correlations.

3.2.1.1 Timelag 0



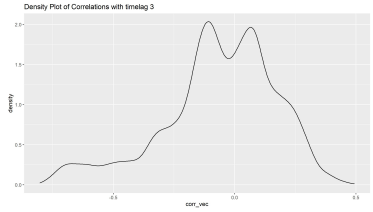


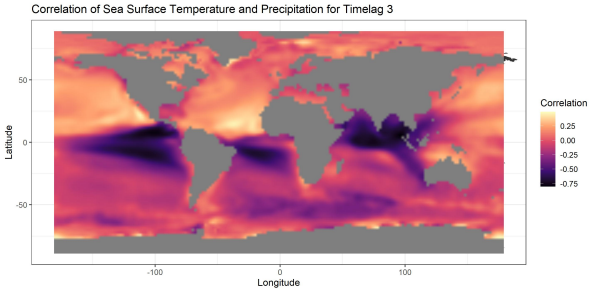


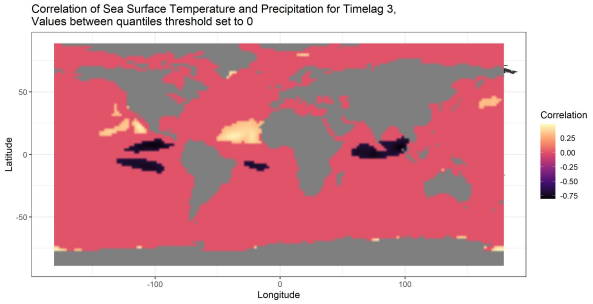
Inspecting the density plot for timelag 0, we see two modi for correlations, one for negative correlations around -0.8 and one for possitive correlations around

0.8. Also a small spike can be seen for low negative correlations. If we plot these correlations on the respective grid points we see a clear north-south negative-positive correlation distinction. The "boarder" is organised around the equator. The plot for the strongest 5% of correlations reveals areas with strong positive and negative correlations in the north and south respectively.

3.2.1.2 Timelag 3



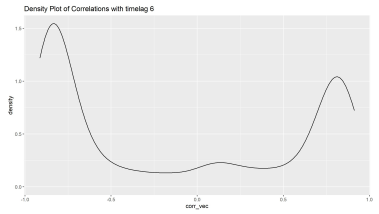


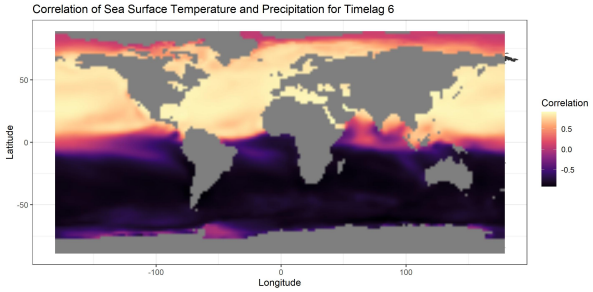


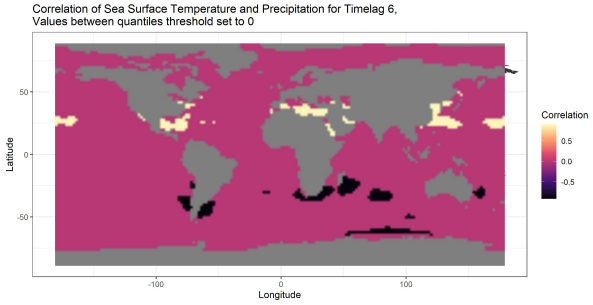
The density of correlations for timelag 3, is left-skewed and has two modi that are organised around 0 and -0.125 respectively. The correlation map shows that the high positive and negative correlations are more close to equator here. Note that the legend for the correlationmap is "shifted" here, because the maximal negative correlation has a higher absolute value than the maximal

positive correlation. The strongest correlations also seem to be shifted towards the equator.

3.2.1.3 Timelag 6

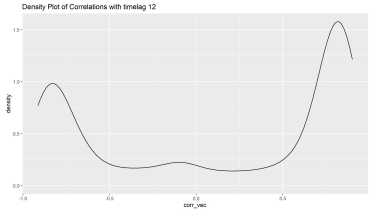


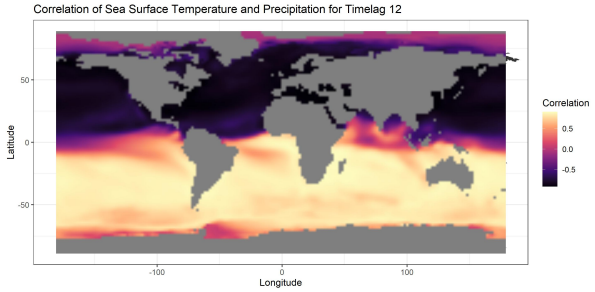


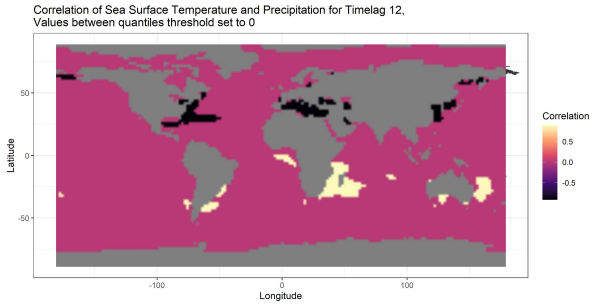


We can see the density plot for timelag 6 is pretty similar to the one of timelag 0 but seems to be "flipped" around 0. Similarly the correlation map shows (high) negative correlations in the south now and high postive correlations in the north.

3.2.1.4 Timelag 12





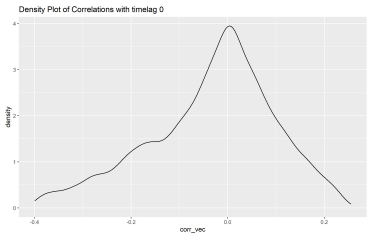


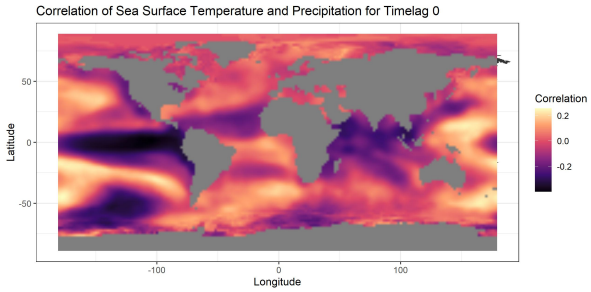
Giving a timelag of one year, we can see that the distribution of correlations is now again similar to the distribution for timelag 0. This also hold for the location of positive and negative correlations in general, as well as for the strongest 5% of correlations.

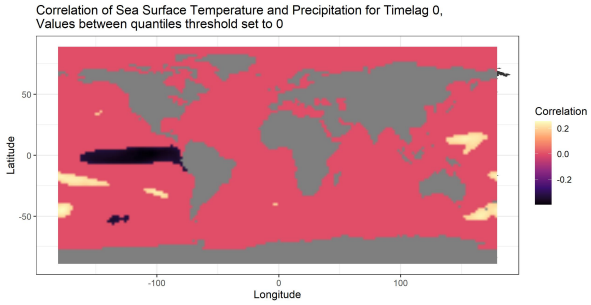
3.2.2 Deseasonalised Data

Following, for each timelag we show the respective density of correlation values, their location on the map and also the 5% strongest positive and negative correlations.

3.2.2.1 Timelag 0

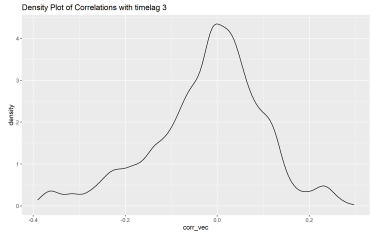


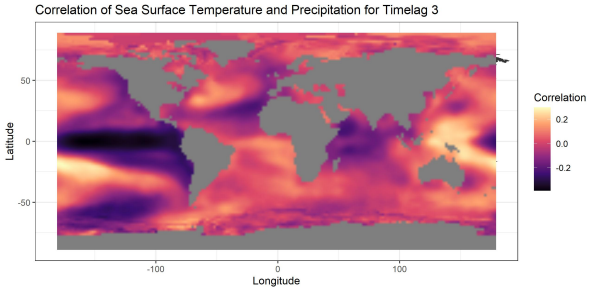


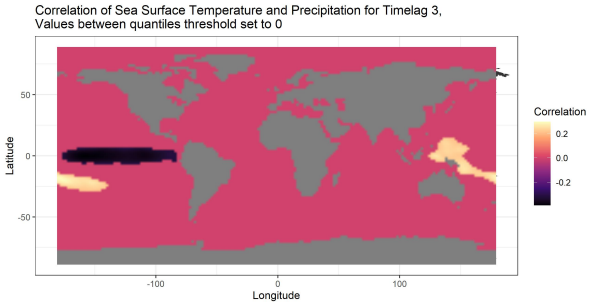


Inspecting the density plot for timelag 0, we see that after excluding seasonality from the time series we get a left-skewed distribution of correlations. With a mode around 0. In general the correlation values are a lot lower than in the original data. With a maximum at around -0.4 and +2.5 respectively. We plot these correlations on the respective grid and see that the clear north south distinction in the correlations before deseasonalising the data does not appear anymore. The plot for the strongest 5% of correlations reveals areas with strongest positive and negative correlations. But as stated before the values are in general much lower.

3.2.2.2 Timelag 3

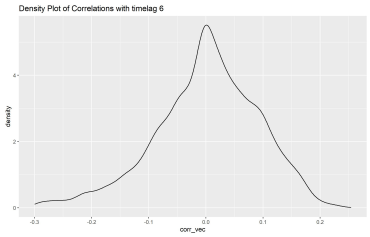


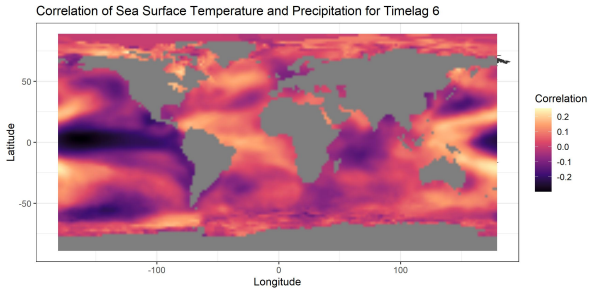


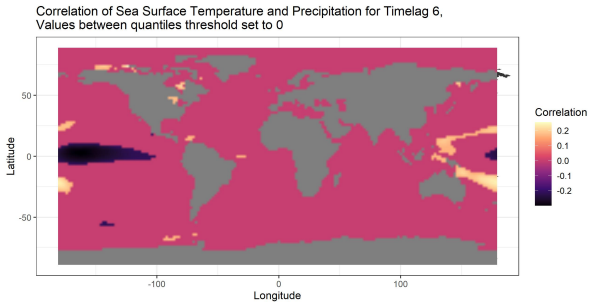


For the density of timelag 3 we get a similar picture as for timelag 0. The mode is a higher and the tails get a bit more mass. Also the correlation map does not seem to change a lot. The strongest correlations appear to be shifted to the left.

3.2.2.3 Timelag 6

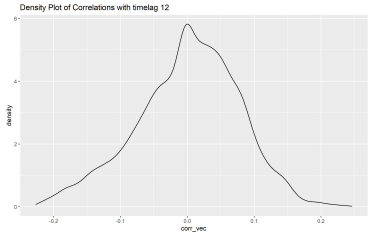


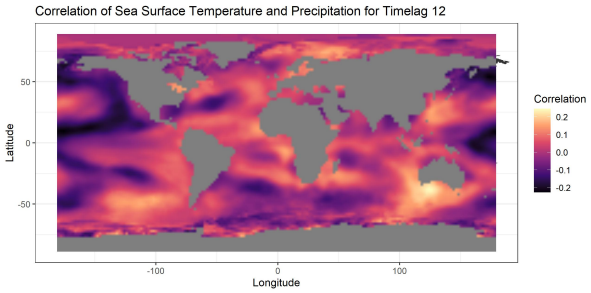


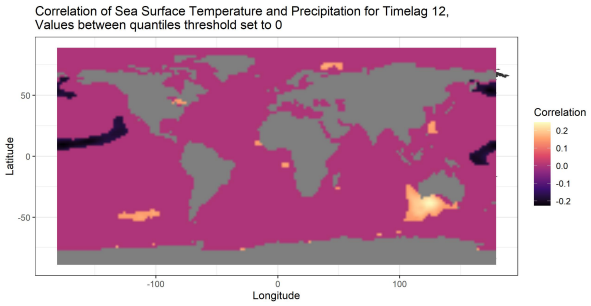


Since the distributions of correlation values are all unimodal we do not observe the "flip" we saw in the original data when comparing the densities of timelag 0 and 6. The mode again gets larger and the maximum postive and negative correlation values get smaller. The strongest negative correlations are shifted further to left.

3.2.2.4 Timelag 12







Given a timelag of one year, the distribution now has a mode around 6, and started at around 4 when timelag was 0. Also neither the positive nor negative correlations exceed values of 2.5. The consistent region of strong negative and positive correlations is now less organised or more scattered.

3.3 Summary

3.3.1 Original Data

We can observe that the positive and negative correlations of sst and precipitation follow a spatial and temporal pattern. The location and density of the positive and negative correlation "wanders" over the equator in opposite directions. The densities and correlationmaps for timelag 0 and 6 appear to be quite similar but "flipped". The densities and correlationmaps for timelag 0 and 12 appear again to be similar. The same pattern seems to hold for the strongest correlations.

3.3.2 Deseasonalised Data

Correlation values are in general a lot lower than in the original data and decrease with increasing timelag. We still observe temporal and regional patterns, although these dissolve a bit for a timelag of 12.

Chapter 4

Clustering

In this chapter we will first summarize the main ideas of clustering and then apply it to the precipitation data. If not indicated otherwise the information is taken from Elements of Statistical Learning.

4.1 Main Idea Clustering

We can describe an object by a set of measurements or its similarity to other objects. Using this similarity we can put a collection of objects into subgroups or clusters. The objects in the subgroups should then be more similar to one another than to objects of different subgroups. This means inside the clusters we aim for homogeneity and for observations of different clusters for heterogeneity. With the clustering analysis applied to the precipitation data we want to study if there are distinct groups (regions) apparent in the CAB. So that if we later apply the regression models we predict the precipitation for each group and not for the whole region.

To explore the grouping in the data we need a measure of (dis)similarity. This measure is central and depends on subject matter considerations. We construct the dissimilarities based on the measurements taken for each month. We interpret this as a multivariate analysis where, each month is one variable. So given the area in the CAB (resolution 5°x5°), we have 612 cells and 432 months, resulting in a 612×432 data matrix. we want to cluster cells into homogen groups.

4.2 Clustering Methods

4.2.1 K-means

In the following we briefly describe the K-means procedure. Beforehand we have to specify a number of clusters C we believe exist in our data. Then we randomly initialize C cluster centers in the feature space. Now two steps are repeated until convergence:

1. For each center we identify the points that are the closest to this center. These points "belong" now to a cluster C.
2. In each cluster we compute the mean of each variable and get a vector of means. This mean vector is now the new center of the cluster.

As a measure of dissimilarity we use the Euclidean distance:

$$d(x_i, x_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = ||x_i - x_{i'}||^2 \tag{4.1}$$

Meaning for the points i and i' we compute the squared difference for each variable and sum them up. As stated above we are searching for clusters that are themselves compact, meaning homogeneous. We do so by minimizing the mean scatter inside the clusters. We summarize this scatter as within-sum-of-squares

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} ||x_i - x_{i'}||^2 = \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \bar{x}_k||^2 \tag{4.2}$$

where $\bar{x} = (\bar{x}_{1k}, ..., \bar{x}_{pk})$ stands for the mean vectors of the kth cluster and $N_k = \sum_{i=1}^{N} I(C(i) = k)$.

4.2.2 Kmeans characteristics

variance of each distribution of each attribute (variable) is spherical, variance is symmetrical? all variables have same variance, not the case in our example, therefore scaling or pca equal number of observations in each clusters, we don‘t know

Since we use the Euclidean distance the similarity measures will be sensitive to outliers and scale. K-means assumes that the variance of a variable´s distribution is spherical, meaning it wight not work well in situations that violate this assumptions (f.e non-spherical data). Further assumptions are same variance of the variables, and equally sized clusters. Now how "large" these violations have to be so that k-means does not work well anymore has no clear-cut answer.

4.2.3 K-medoids

We can adjust k-means procedure so that we can use other distances than the Euclidean distance. The only part of the k-means algorithm that uses Euclidean distance is when we compute the cluster centers. We can replace this step by formalizing an optimization with respect to the cluster members. For example so that each center has to be one of the observations assigned to the cluster. K-medoids is far more computationally intensive than K-means.

4.2.3.1 K-medoids characteristics

K-medoids is less sensitive to outliers, because it minimizes sum of pairwise dissimilarities instead of sum of squared Euclidean distances. As stated above it is also more computationally intensive.

4.2.4 PCA

Goal is reduction of correlated and eventually large number p variables to a few. We accomplish this by creating new variables that are linear combinations of the original ones. We call these new variables principle components. The new variables are not correlated any more and ordered according to the variance they explain. The first $k < p$ principal components then contain the majority of variance (Fahrmeir et al. (1996)). As they are ordered they also provide a sequence of best linear approximations of our data. Let $x_1, x_2, ..., x_N$, be our observations and we represent them by a rank-q linear model

$$f(\lambda) = \mu + V_q \lambda \tag{4.3}$$

with μ a location vector in $\mathbb{R}^p$ and V_q is a $p \times q$ matrix with columns being orthogonal unit vectors as columns. λ is a q vector of parameters. In other words we are trying to fit a hyperplane of rank q to the data. If we fit this model to minimize reconstruction error using least squares we solve

$$\min_{\mu, \{\lambda_i\}, V_q} \sum_{i=1}^{N} ||x_i - \mu - V_q \lambda_i||^2 \tag{4.4}$$

If we partially optimize for μ and λ_i we obtain

$$\hat{\mu} = \bar{x},$$

$$\hat{\lambda_i} = V_q^T(x_i - \bar{x})$$

Therefore we need to search for the orthogonal matrix V_q

$$\min_{V_q} \sum_{i=1}^{N} ||(x_i - \bar{x}) - V_q V_q^t (x_i - \bar{x})||^2 \tag{4.5}$$

We can assume here that $\bar{x}$ is 0, if this not the case we can simply center the observations $\tilde{x}_i = x_i - \bar{x}$. $H_q = V_q V_q^T$ projects each observation x_i from the original feature space onto the subspace that is spanned by the columns of V_q. $H_q x_i$ is then the orthogonal projection of x_i Hence H_q is also called the *projection matrix*.

We find V_q then by constructing the *singular value decomposition* of our data matrix X. X contains the (centered) observations in rows, giving a $N \times p$ matrix. The SVD is then:

$$X = UDV^T \tag{4.6}$$

Where U is an orthogonal matrix containing the *left singular vectors* u_j as columns, and V contains the *right singular vectors* v_j. The columns of U span the columns space of X and the columns of V span the row space. D is diagonal matrix which contains *singular values*, $d_1 \leq d_2 \leq ... \leq d_p \leq 0$. *Singular values* are square roots of non-negative eigenvalues. The columns of UD are the principal components of X. So the SVD gives us the matrix V (the first q columns give the solution to the minimization problem above) as well as the principal components from UD (Hastie et al. (2009)).

4.2.5 Gap statistic

The idea of the gap statistic was introduced by R. Tibshirani, Walther, and Hastie (2001) As stated above we usually measure how compact our clusters are by assessing $W(C)$ or $log(W_c)$. Where low values indicate compact clusters. To compare the value then, we need a reference. We therefore want to estimate how large W_c were if there were no clusters present in our data. The larger the difference between the W_c from the data and the one from the reference the more likely we are to say that the found number of clusters is indeed correct. We construct reference data by sampling from a uniform distribution based on our data. Say we have p variables. We sample n times from each of the p uniformly distributed variables, where maximum and minimum are obtained from our data. We then cluster the reference data in the same we cluster our observed data and compute W_c. We repeat this

process several times and compute the average of W_c, $E\{log(W_c)\}$. The gap statistic is then the difference

$$E\{log(W_c)\} - log(W_c). \tag{4.7}$$

So in cases where our data is formed of clusters we would expect a high gap statistic. As R. Tibshirani, Walther, and Hastie (2001) note and as it is also done in the used R function clusgap, doing a PCA on the data and compute the gap statistic on the PCA scores can improves the results of gap statistic.

4.3 Analyse clustering results

We now analyze further the results we found. Namely the clusters we find after performing a PCA on the data, using the 3 first principal components and searching for 5 clusters using k-means. The first 3 principal components explain 67.8% of the variance We found 5 as optimal number of clusters based on the gap statistic and the criteria from Tibsherani.

We show the map plot for the k-means clustering after applying the PCA as well as the time series of the resulting clusters.

The plot above shows the grid cells in the Central Amazon Basin colored according to the clusters that are assigned by k-means. The clusters are almost completely spatially coherent. Meaning that the clusters are not scattered across different areas. One exception can be seen for Cluster 1 and 4. Parts of cluster 1 (orange) are inside cluster 4 (blue) and on the edge to cluster 3 (green).

We now inspect the original (centered) time series inside the clusters. The time series are shown in gray and the monthly mean in the cluster is shown in blue. Since the time series are centered before applying the PCA and clustering, the zero value is the mean of the respective month of the whole CAB.
The clusters differ in their monthly differences from the monthly CAB mean (here 0 because the time series were centered before PCA and K-means) and their fluctuation/ variance. Also the size of the clusters are not all the same. The mean in cluster 3 has lowest variability around the CAB mean, followed by clusters 2 and 5, and clusters 1 and 4 have the highest variability.
On average cluster 3 is on the level of the CAB overall mean. The clusters 2 and 5 are slightly below and cluster 4 is above the CAB mean, on average. Cluster 1 is on average also on the CAB mean but shows more variance than cluster 3.

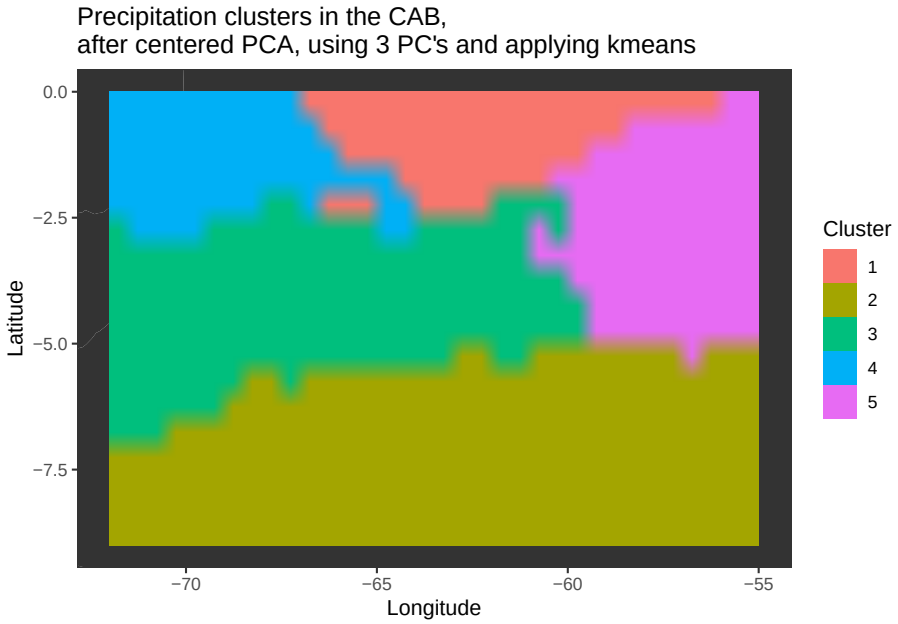


Figure 4.1: Spatial distribution of the found clusters in the CAB. We applied a centered PCA on the data and used 3 principal components before applying the K-means algorithm

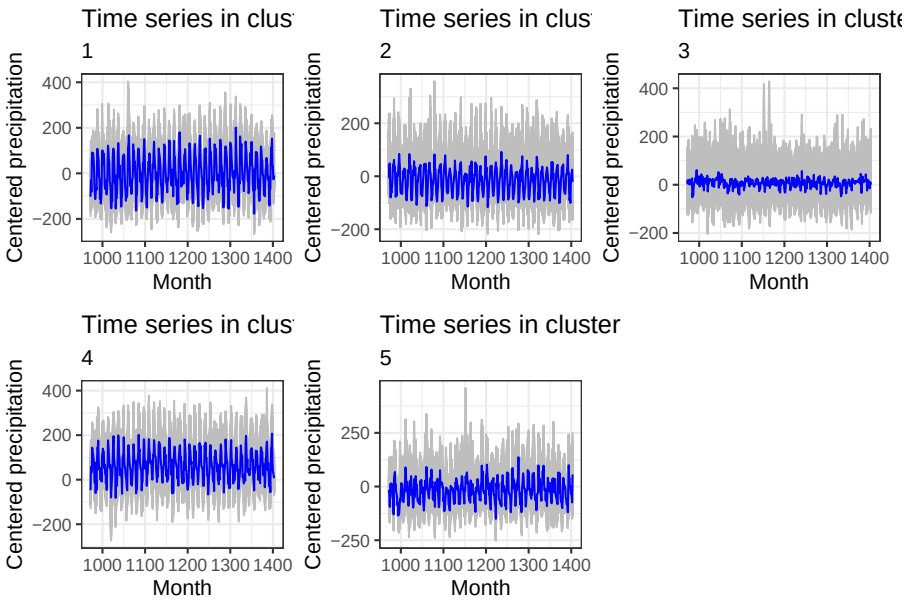


Figure 4.2: Plots of the time series in the clusters we found using the K-means algorithm. The x-axis shows the month of measurement, the y-axis the centered precipitation. Centering was done according to the overall CAB mean in each month. The mean inside each cluster for a month is displayed in blue.

Chapter 5

LASSO Regression

5.1 The LASSO

We want to create a model that has predictive and explanatory power. Predictive power meaning it can predict the precipitation in the Central Amazon Basin, "reasonably well". Explanatory power in the sense of being interpretable, so that we can identify those regions in sea that give us most information about future precipitation. Our problem setting is high dimensional with n « p. The number of predictors is a lot bigger than the number of actual observations. This creates issues with a classic linear model since the linear problem is underdetermined. One possible model for the problem at hand is a LASSO regression model.

In general for the linear model:

$$y_i = \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta} + \epsilon_i \tag{5.1}$$

see (5.1) Where the y_i refers to the mean precipitation for a given month i and x_i is the vector of sea surface temperature at different locations around the globe. ϵ_i is the residual and we wish to estimate the β´s from the data. As already stated this is not possible with a classic linear model since the number of predictors exceeds the number of observations. We therefore can not estimate a β for every grid point in the sea. From a physical point of view it also seems reasonable that some regions in the ocean have a higher predictive power than others. For example regions that are closer to the Amazon may have more influence on precipitation in the same month. But regions more far away may have more information on the precipitation half a year in the future. We therefore would like to use a model that can find the most important regions in the sea for predicting precipitation for some point in the future.

One possible solution for this is a LASSO regression model, as implemented in R by the *glmnet* package (Friedman, Hastie, and Tibshirani (2010)). This model "automatically" performs model selection, but be aware that because of the time dependencies in our data, normal Cross Validation methods may be unjustified or at least have to be applied with caution. The glmnet package implements the regression problem in the following manner, solving:

$$\min_{\beta_0,\beta} \frac{1}{N} \sum_{i=1}^{N} w_i l(y_i, \beta_0 + \beta^T x_i) + \lambda[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1] \tag{5.2}$$

This is a lasso regression for $\alpha = 1$ and ridge regression for $\alpha = 0$, α controls the overall strength of regularization or penalty. Intuitively this means we try to find those β´s that minimize the negative log likelihood of our data (this is equal to maximizing the log-likelihood). But at the same time we can not include too many β since this will make the second and third term in the formula grow. As result the algorithm chooses only those predictors that have the most predictive power. How many predictors are included depends on the strength of regularization given by α. *Remark*: Among strongly correlated predictors only one is chosen in the classical lasso model. Ridge regression shrinks the coefficients to zero. Elastic net with $\alpha = 0.5$ tends to either include or drop the entire group together. To specifically choose a group of predictors, variations of the lasso or other models have to be considered.

5.2 Optimization

The glmnet function finds a solution path for the lasso problem via coordinate descent. The implemented algorithm was suggested by Van der Kooij (2007). We can write down the optimization procedure as follows: Given N observation pairs (x_i, y_i) with $Y \in \mathbb{R}$ and $X \in \mathbb{R}^p$, we approximate the regression function with $E(Y|X = x) = \beta_0 + x^T\beta$, Here x_{ij} are considered standardized, so $\sum_{i=1}^{N} = 0, \frac{1}{N}\sum_{i=1}^{N} x_{ij}^2 = 1$ for $j = 1, ..., p$. We then solve the problem:

$$\min_{(\beta_0,\beta)\in\mathbb{R}^{p+1}} [\frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T\beta)^2 + \lambda[(1-\alpha)||\beta||_2^2/2 + \alpha||\beta||_1] \tag{5.3}$$

Note that this solves the elastic net problem that also uses a ridge penalty. We follow the elastic net description but in our case $\alpha = 1$, using only the lasso penalty. We consider now a coordinate descent step for solving (5.3). Given we have estimates $\tilde{\beta}_0$ and $\tilde{\beta}_l$ and we want to partially optimize with respect to β_j, and $i \neq j$. When $\beta_j > 0$,

$$\left.\frac{\partial R_\lambda}{\partial \beta_j}\right|_{\beta=\tilde{\beta}} = -\frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \tilde{\beta}_0 - x_i^\top \tilde{\beta}\right) + \lambda(1-\alpha)\beta_j + \lambda\alpha. \tag{5.4}$$

And similar expressions exist for $\tilde{\beta}_j < 0$. $\tilde{\beta}_j = 0$ is treated separately. The coordinate-wise update has then the form:

$$\tilde{\beta}_j \leftarrow \frac{S\left(\frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \tilde{y}_i^{(j)}\right), \lambda\alpha\right)}{1+\lambda(1-\alpha)}. \tag{5.5}$$

with

- $\tilde{y}_i^{(j)} = \tilde{\beta}_0 + \sum_{\ell\neq j} x_{i\ell}\tilde{\beta}_\ell$ standing for fitted value without the contribution from x_{ij}, and therefore $y_i - \tilde{y}_i^{(j)}$ is the partial residual when fitting β_j. Because we applied a standardization, $\frac{1}{N}\sum_{i=1}^{N} x_{ij}\left(y_i - \tilde{y}_i^{(j)}\right)$ denotes the simple least-squares coefficient for fitting this partial residual to x_{ij}.
- $S(z,\gamma)$ being the soft-thresholding operator. It's value is given by:

$$\text{sign}(z)(|z|-\gamma)_+ = \begin{cases} z-\gamma & \text{if } z>0 \text{ and } \gamma<|z| \\ z+\gamma & \text{if } z<0 \text{ and } \gamma<|z| \\ 0 & \text{if } \gamma \geq |z|. \end{cases} \tag{5.6}$$

So in summary the steps are as follows: Compute the simple least-squares coefficient on the partial residual, then apply soft-thresholding and proportional shrinkage for the lasso and ridge penalty, respectively. Again for our use case, since we use the lasso and $\alpha = 1$, we only apply soft-thresholding and no proportional shrinkage.

The solutions are computed starting from smallest λ_{max} for which all elements in $\hat{\beta} = 0$. For all larger λ the coefficients then stay 0. The smallest λ value λ_{min} is then selected by $\lambda_{min} = \epsilon\lambda_{max}$. The complete searched vector is constructed as sequence of K values, typical values are $\epsilon = 0.001$ and $K = 100$. This procedure is an example of so called *warm starts*. By default they always center the predictor variable. For additional information on other methods how speedup is obtainred refer to Section 2 in Friedman, Hastie, and Tibshirani (2010).

5.3 TODO here

maybe drop into with linear model just put lasso formula directly talk about the stuff that is written already note probelms with correlation of predictors and grouping

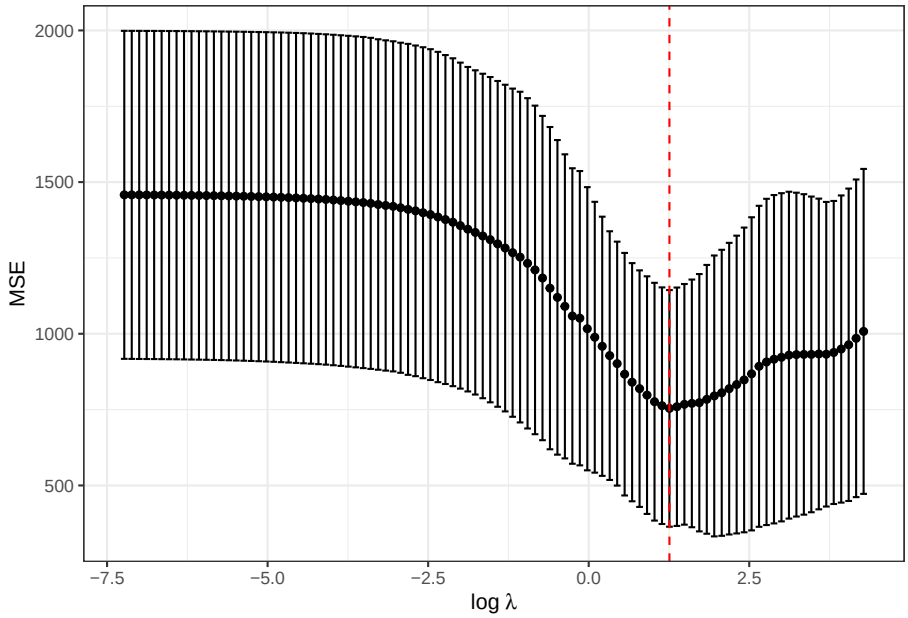
Chapter 6

lasso center

6.1 LASSO model

We fit a LASSO model on the precipitation and SST data. The precipitation target is the monthly mean precipitation in the Central Amazon Basin, the SST data are monthly temperatures over the globe. We use a 5-fold CV approach to find an optimal lambda. Each fold consists of 5 consecutive years of training data followed by 2 years of test data. In each fold we fit a LASSO model on a set of predetermined lambda values and choose the lambda that minimizes the MSE on the test set in that fold. After determining the best lambda in each fold we choose the lambda that minimizes the MSE over all folds and refit the model to the complete training data. Afterwards we evaluate the fitted model on a separate validation set with 5 years length which was not included in the training phase.

6.2 Error plots



?? shows the results from the 5 fold-CV plotted for each lambda. The lambdas are given on the log scale. The upper and lower bars indicate mean MSE +/- one standard deviation from the mean MSE. We note that the upper and lower bars are quite wide indicating big differences in MSE for the different folds. We therefore also inspect the MSE for each individual fold.

At first glance 6.1 shows that the MSE for fold 1 and 2 have similar trajectories, the same for 3 and 4, while fold 5 is the only one that only has a local maximum somewhat in the middle of the log lambda range. But fold 5 also has its minimum MSE at a larger regularization value than the other folds which is also reflected in the number of coefficients it includes in the model, as we will see in the coefficient plots. Also obviously the MSE differ greatly in their values as can be seen on the differences of their respective y-axis. While this plot works well for getting an overview of the trajectories we can replot them with a common y-axis to compare their values more easily.

We can see now that fold 2 settles for far larger errors than fold 5 for example. Fold 5 chooses the highest lambda but also has the lowest minimal MSE.

Below we can see the minimum MSE for each fold.

```
## [1]  497.4612 1379.5470  855.1009  501.0376  293.3958
```

And which lambda in the lambda vector resulted in the lowest prediction error on the folds´ test set.

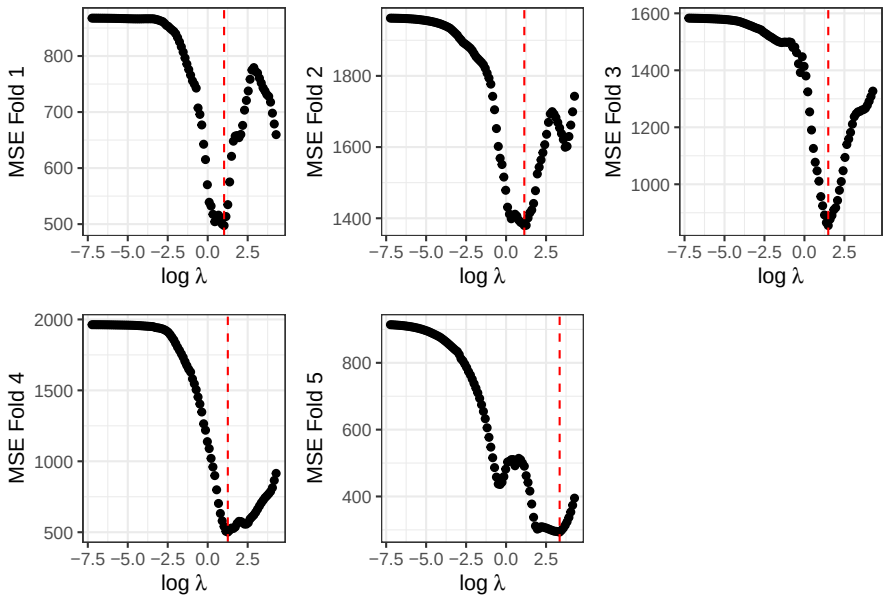


Figure 6.1: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained.

```
## [1]  2.786097  3.129690  4.436255  3.515656 28.516570
```

6.3 Coefficient plots

The plots displays the nonzero coefficients in each fold computed for the lambda that minimizes the MSE on the test set in the respective fold. The LASSO chooses among correlated variables only one and discards the others, which can be seen here since the variables chosen are scattered across the map and can but don't have to be close to each other. If we take a look again at **??** we can see that the model includes locations that have high correlations.

6.4 Inspect predictions from each fold

Following we inspect the folds precipitation time series and the predictions made by the model.

In general the model fits the data sufficiently to predict the general form of the time series but misses some modes and is off in the larger values in fold 2.

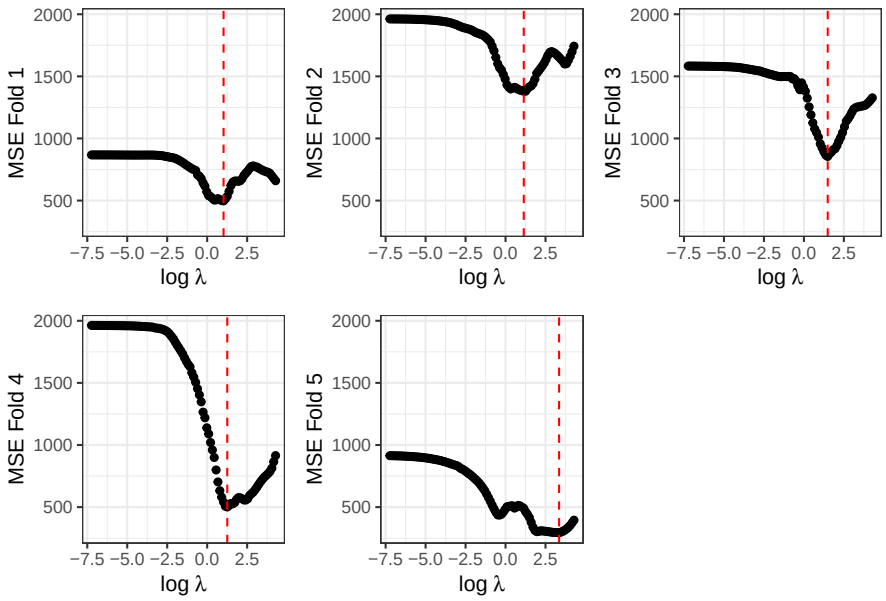


Figure 6.2: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained. See (**ef?**)(fig:err-folg-lasso-og), but this time the y-axis has the same range for all plots.

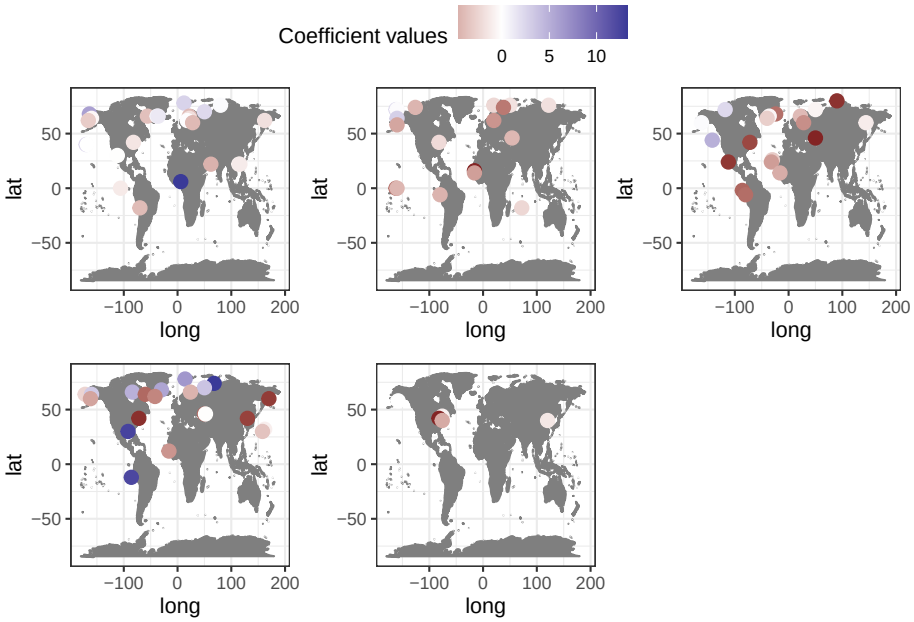


Figure 6.3: Coefficient map plot for the different folds. Longitude and Latitude on the x and y-axis respectively. Positive values are coloured in blue, negative values in red.

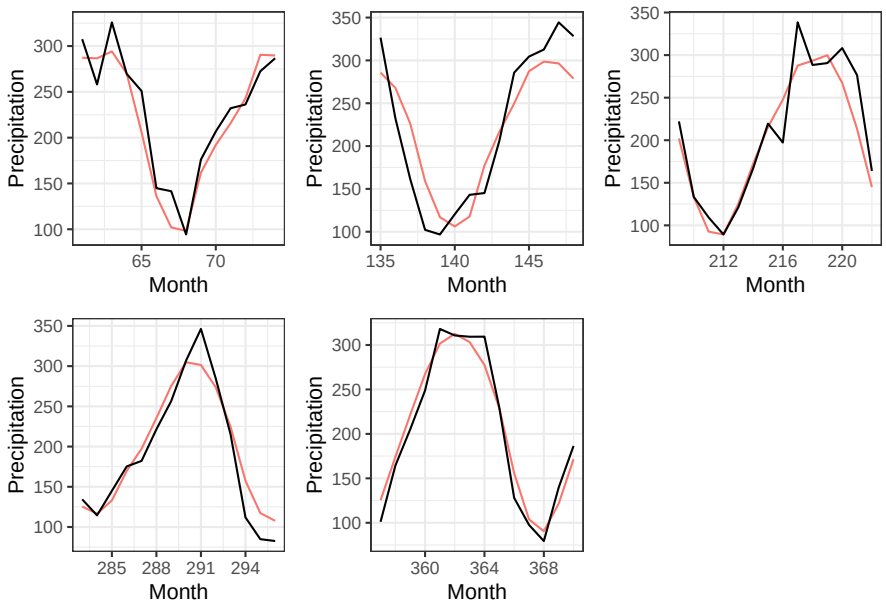


Figure 6.4: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.

Also it does not fit well rapid changes as in fold 3. Therefore it seems that the model generally underfits the data.

6.5 Inspect predictions from best CV-lambda

A common practice is too choose the largest lambda so that it‘s mean MSE is smaller than the MSE of the lambda that minimizes mean MSE plus one SE. But since in our case the largest lambda that satisfies these criteria is the maximum lambda we choose the lambda with minimum mean MSE instead.

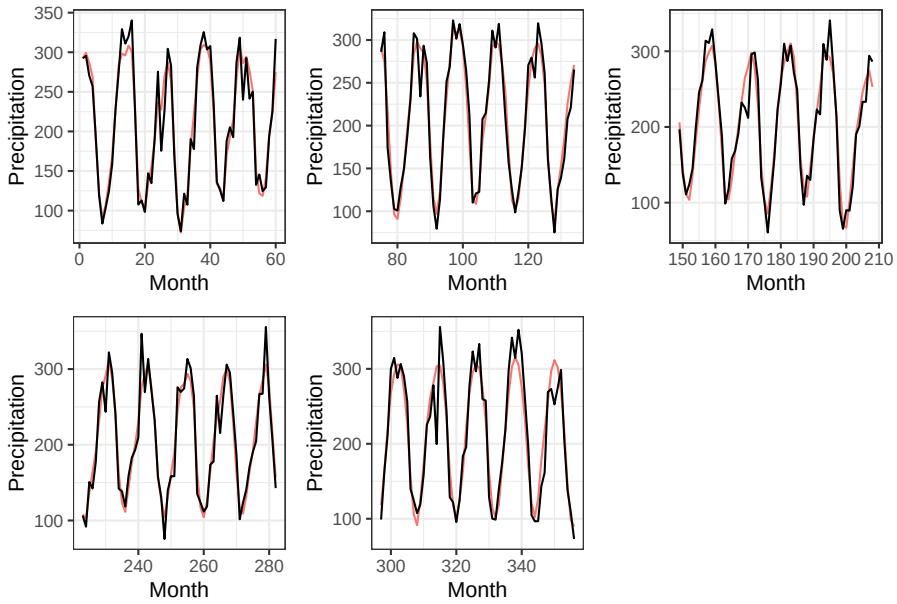
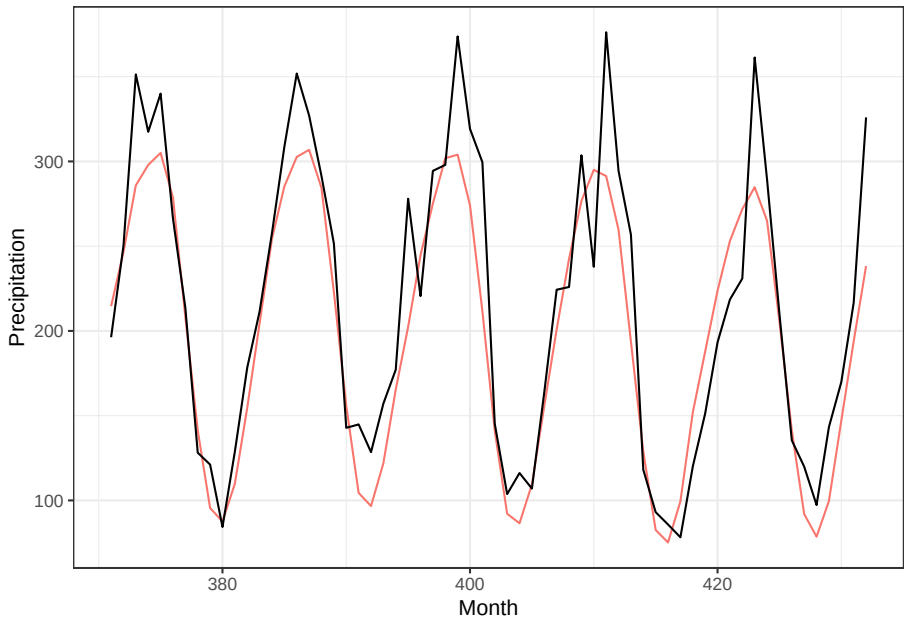
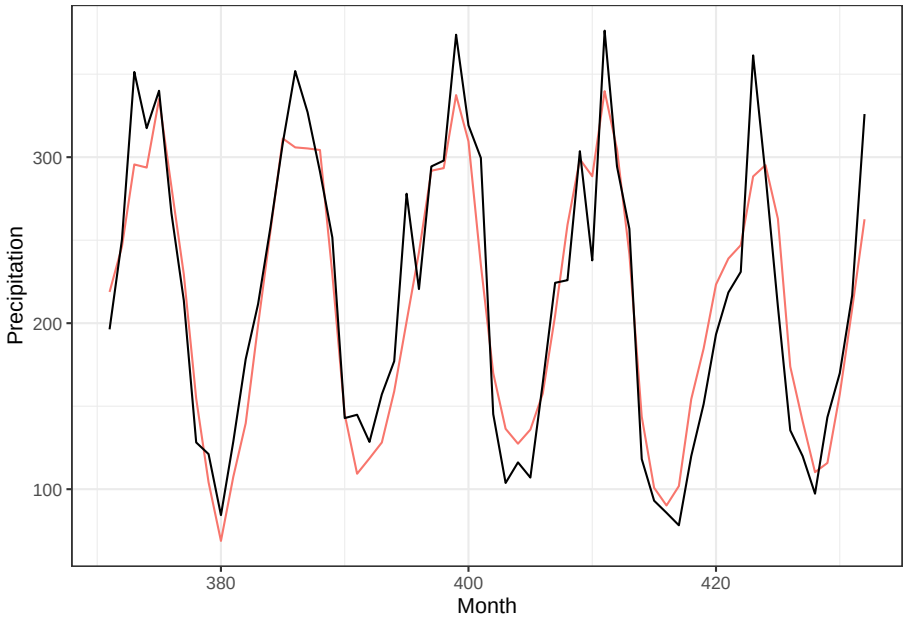
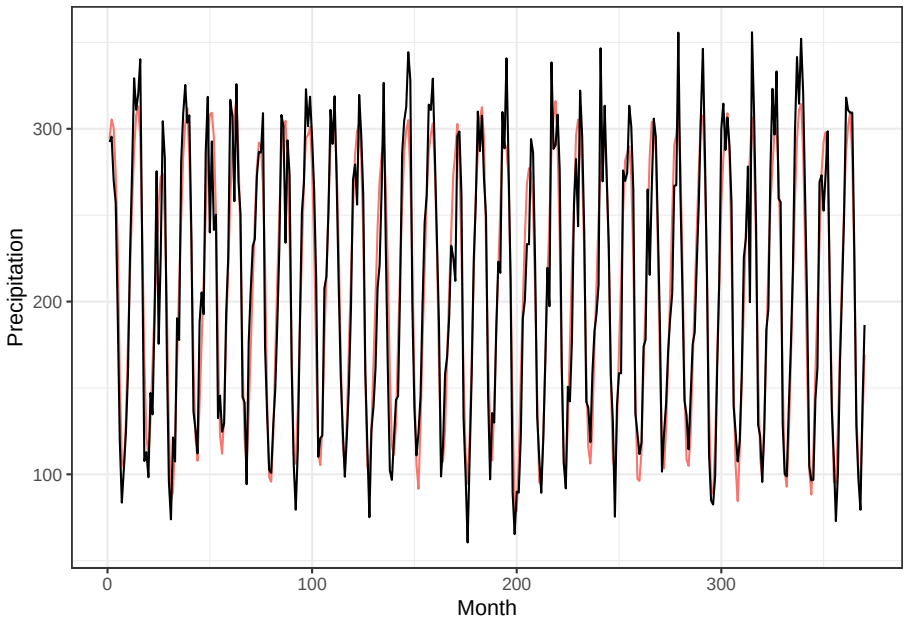


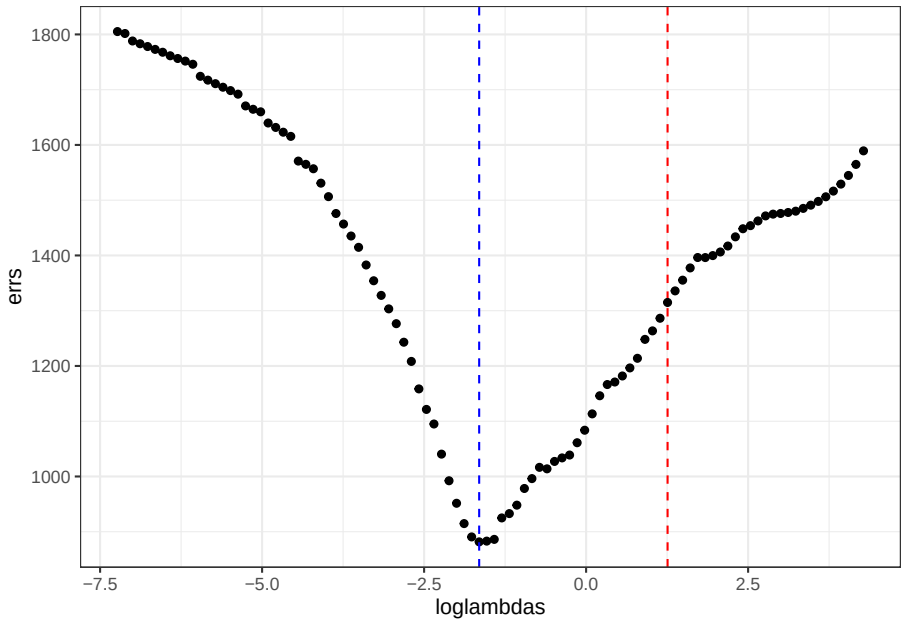
Figure 6.5: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.



```
## [1] 1314.929
```


Precipitation
300
200
100
380
400
420
Month

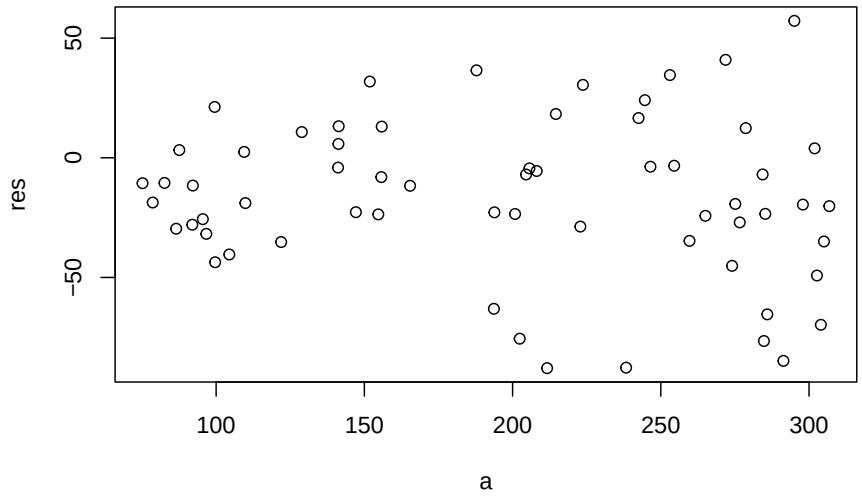

Precipitation
300
200
100
0
100
200
300
Month

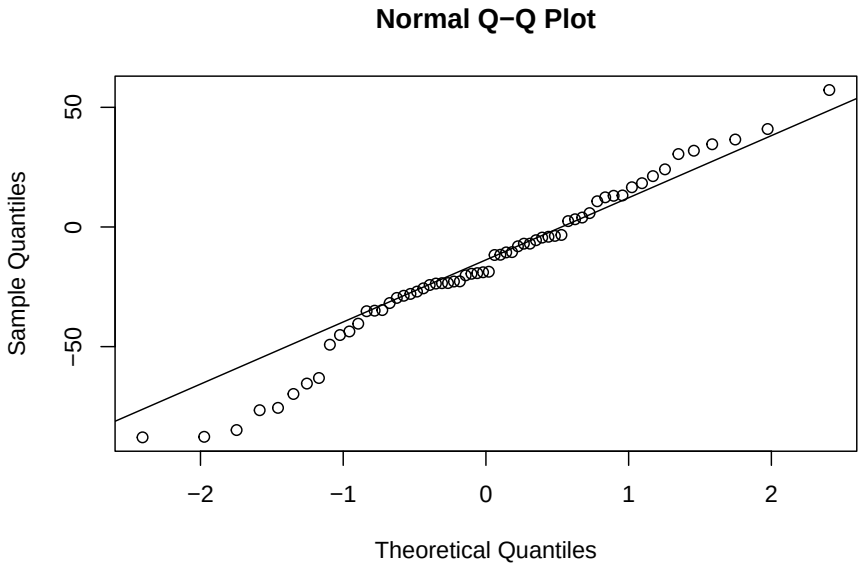


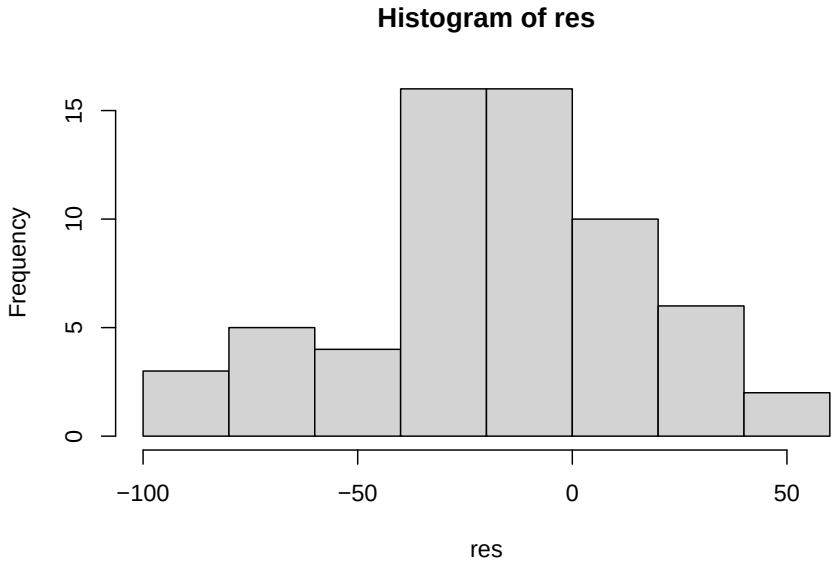
Over the more than 5 years of validation data the model predicts the seasonal pattern of the precipitation time series quite well, but constantly fails to predict the higher values of precipitation. The MSE is `mse_full` and the RSME `sqrt(mse_full)`.

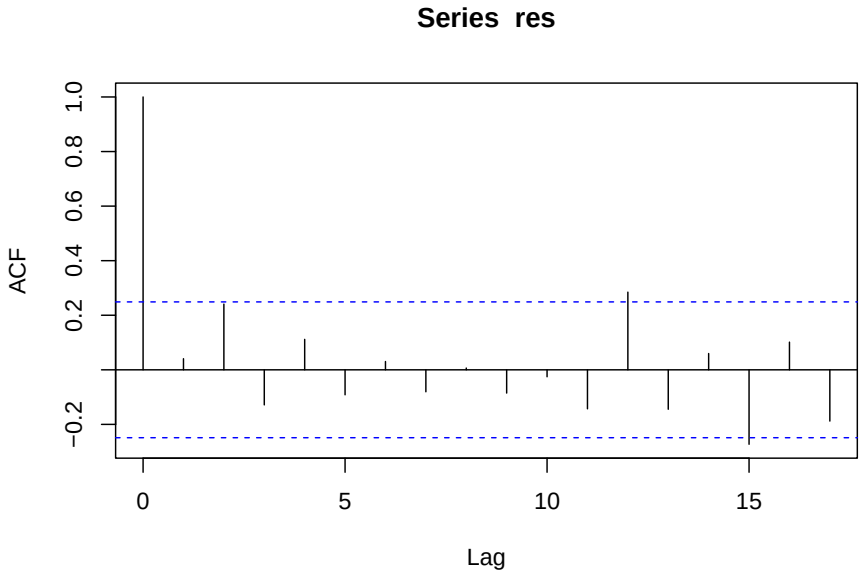
6.6 Summary

We fitted a LASSO model for predicting the mean precipitation in the Central Amazon Basin and used a 5-fold blocked Cross Validation approach to find the optimal level of regularization. After training the model we evaluated its performance on a separate validation set that was not used in the training process. The model shows predicting capabilities but misses out on higher values of the precipitation target. It also misses on rapid changes and in general underfits the data. This may be due to the choice of blocked cross validation. Locations with higher variability get included in the model more easily and are not necessarily geographically close.


res
50
0
−50
100
150
200
250
300
a

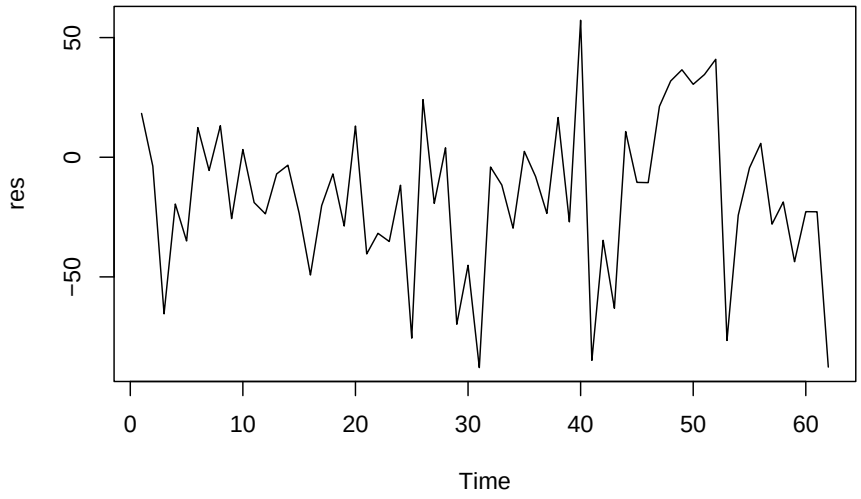

Normal Q−Q Plot
Sample Quantiles
50
0
−50
−2
−1
0
1
2
Theoretical Quantiles

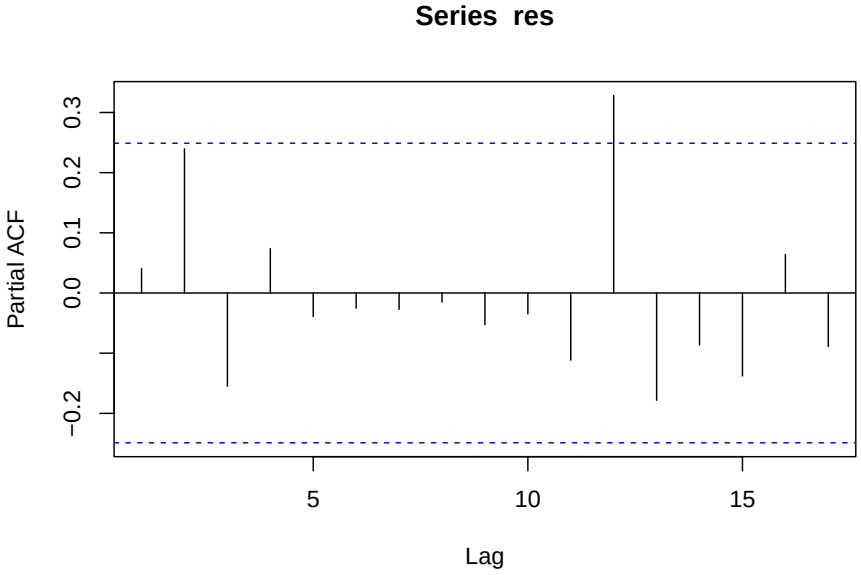


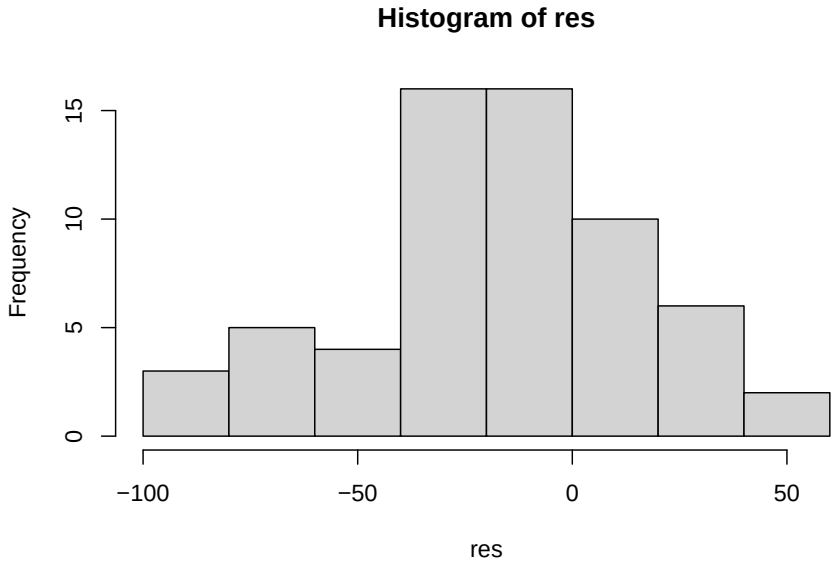



```
##
##  Box-Ljung test
##
## data:  res
```

```
## X-squared = 29.347, df = 20, p-value = 0.08115
```






Histogram of res
Frequency
0
5
10
15
−100
−50
0
50
res

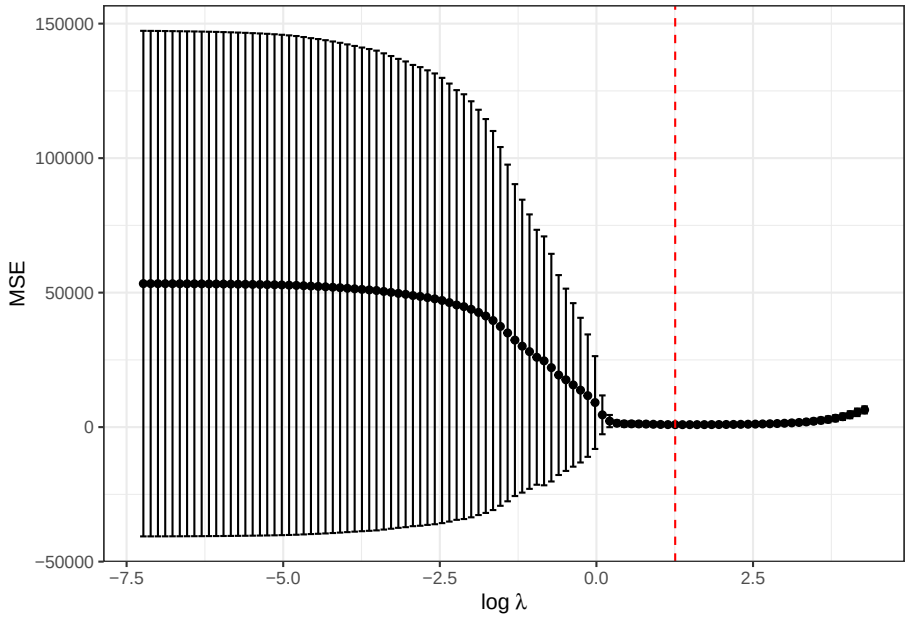
Chapter 7

lasso stand

7.1 LASSO model

We fit a LASSO model on the precipitation and SST data. The precipitation target is the monthly mean precipitation in the Central Amazon Basin, the SST data are monthly temperatures over the globe. We use a 5-fold CV approach to find an optimal lambda. Each fold consists of 5 consecutive years of training data followed by 2 years of test data. In each fold we fit a LASSO model on a set of predetermined lambda values and choose the lambda that minimizes the MSE on the test set in that fold. After determining the best lambda in each fold we choose the lambda that minimizes the MSE over all folds and refit the model to the complete training data. Afterwards we evaluate the fitted model on a separate validation set with 5 years length which was not included in the training phase.

7.2 Error plots



?? shows the results from the 5 fold-CV plotted for each lambda. The lambdas are given on the log scale. The upper and lower bars indicate mean MSE +/- one standard deviation from the mean MSE. We note that the upper and lower bars are quite wide indicating big differences in MSE for the different folds. We therefore also inspect the MSE for each individual fold.

At first glance 6.1 shows that the MSE for fold 1 and 2 have similar trajectories, the same for 3 and 4, while fold 5 is the only one that only has a local maximum somewhat in the middle of the log lambda range. But fold 5 also has its minimum MSE at a larger regularization value than the other folds which is also reflected in the number of coefficients it includes in the model, as we will see in the coefficient plots. Also obviously the MSE differ greatly in their values as can be seen on the differences of their respective y-axis. While this plot works well for getting an overview of the trajectories we can replot them with a common y-axis to compare their values more easily.

We can see now that fold 2 settles for far larger errors than fold 5 for example. Fold 5 chooses the highest lambda but also has the lowest minimal MSE.

Below we can see the minimum MSE for each fold.

```
## [1]  594.0372 1887.9140  785.0662  561.1272  347.1841
```

And which lambda in the lambda vector resulted in the lowest prediction error on the folds´ test set.

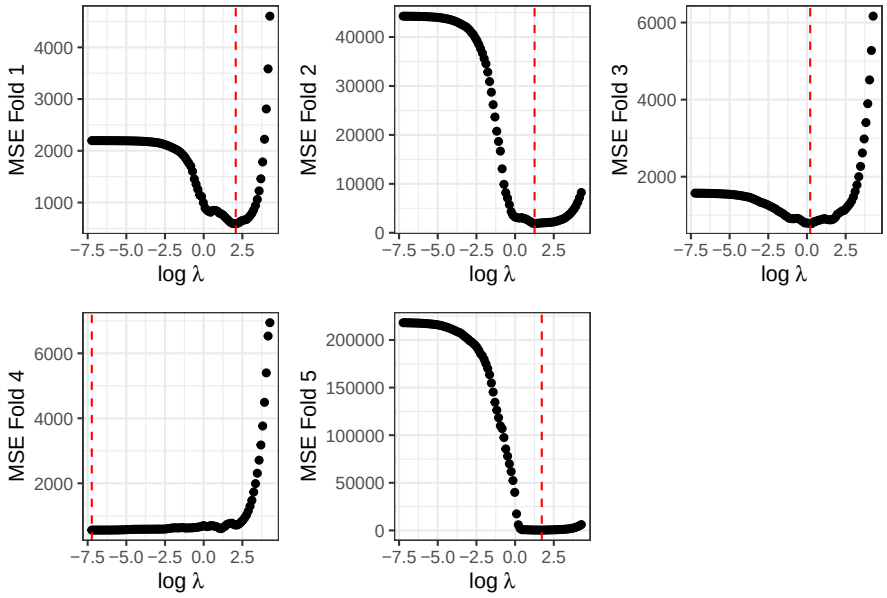


Figure 7.1: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained.

```
## [1] 7.934904903 3.515655993 1.234414226 0.000722999 5.597918545
```

7.3 Coefficient plots

The plots displays the nonzero coefficients in each fold computed for the lambda that minimizes the MSE on the test set in the respective fold. The LASSO chooses among correlated variables only one and discards the others, which can be seen here since the variables chosen are scattered across the map and can but don't have to be close to each other. If we take a look again at **??** we can see that the model includes locations that have high correlations.

7.4 Inspect predictions from each fold

Following we inspect the folds precipitation time series and the predictions made by the model.

In general the model fits the data sufficiently to predict the general form of the time series but misses some modes and is off in the larger values in fold 2.

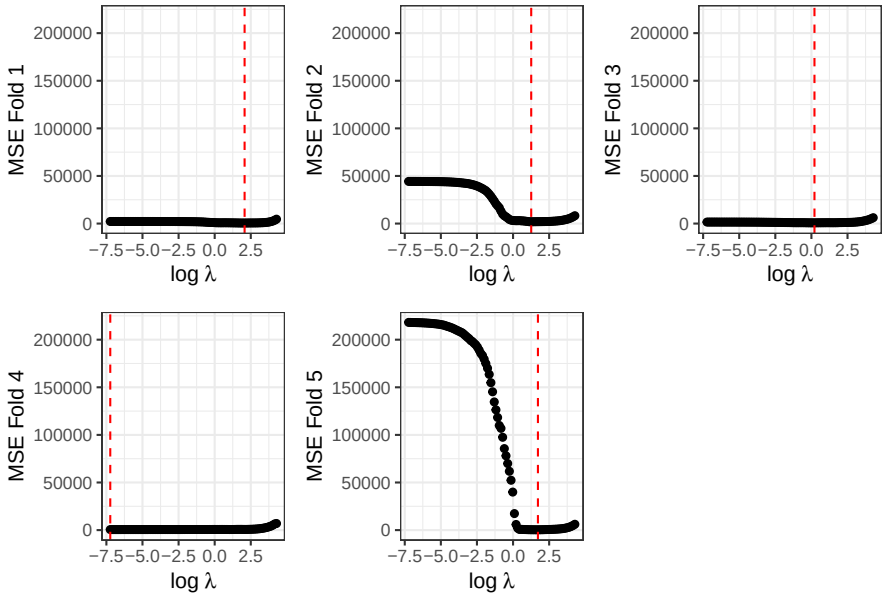


Figure 7.2: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained. See (**ef?**)(fig:err-folg-lasso-og), but this time the y-axis has the same range for all plots.

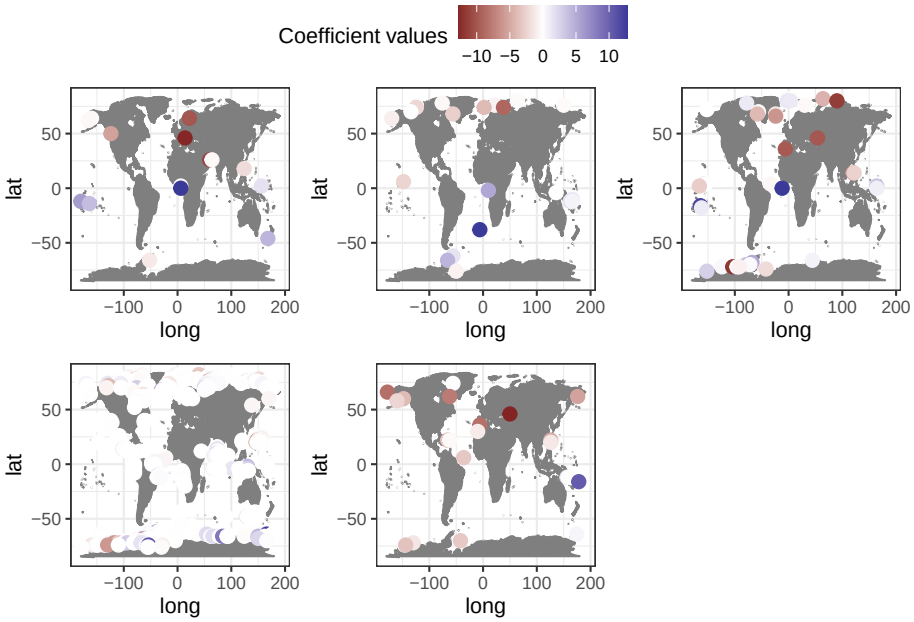


Figure 7.3: Coefficient map plot for the different folds. Longitude and Latitude on the x and y-axis respectively. Positive values are coloured in blue, negative values in red.

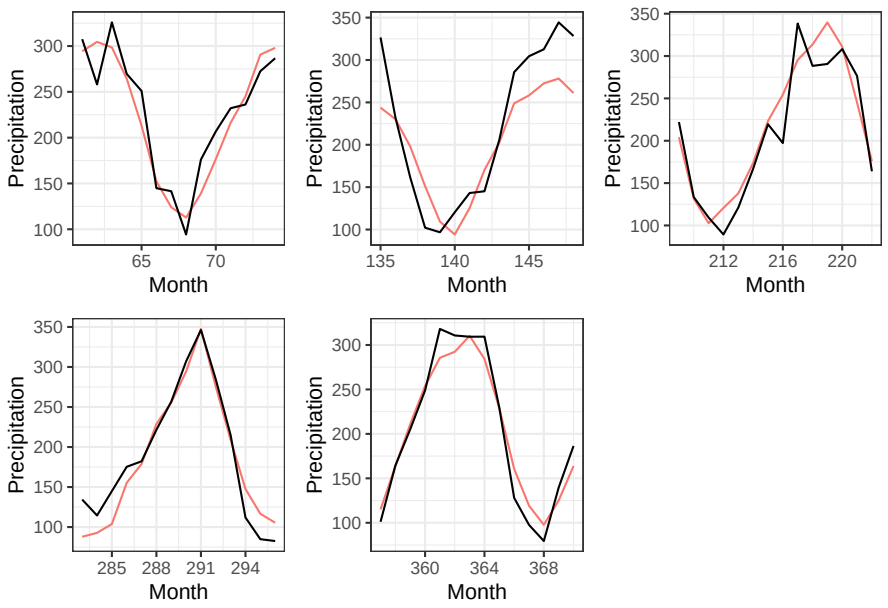


Figure 7.4: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.

Also it does not fit well rapid changes as in fold 3. Therefore it seems that the model generally underfits the data.

7.5 Inspect predictions from best CV-lambda

A common practice is too choose the largest lambda so that it‘s mean MSE is smaller than the MSE of the lambda that minimizes mean MSE plus one SE. But since in our case the largest lambda that satisfies these criteria is the maximum lambda we choose the lambda with minimum mean MSE instead.

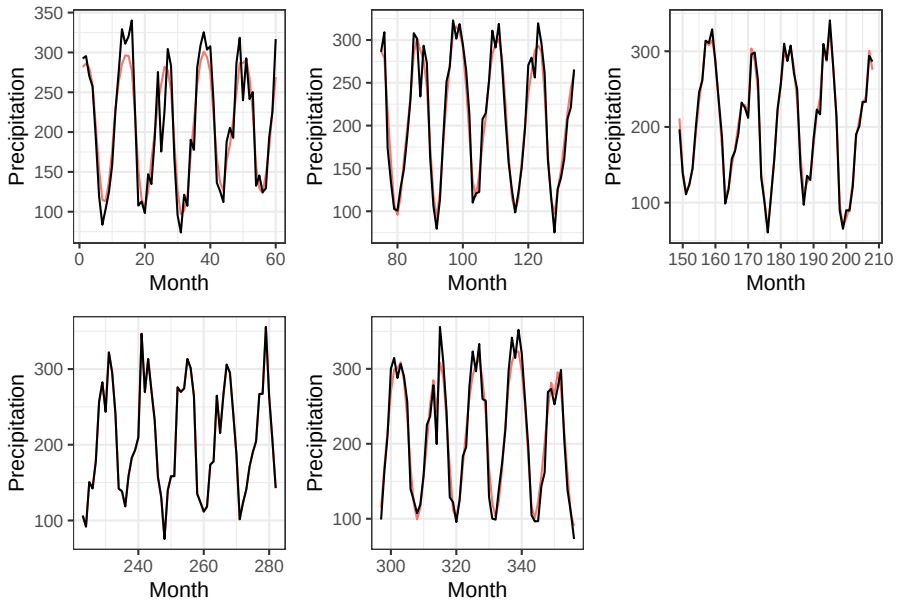
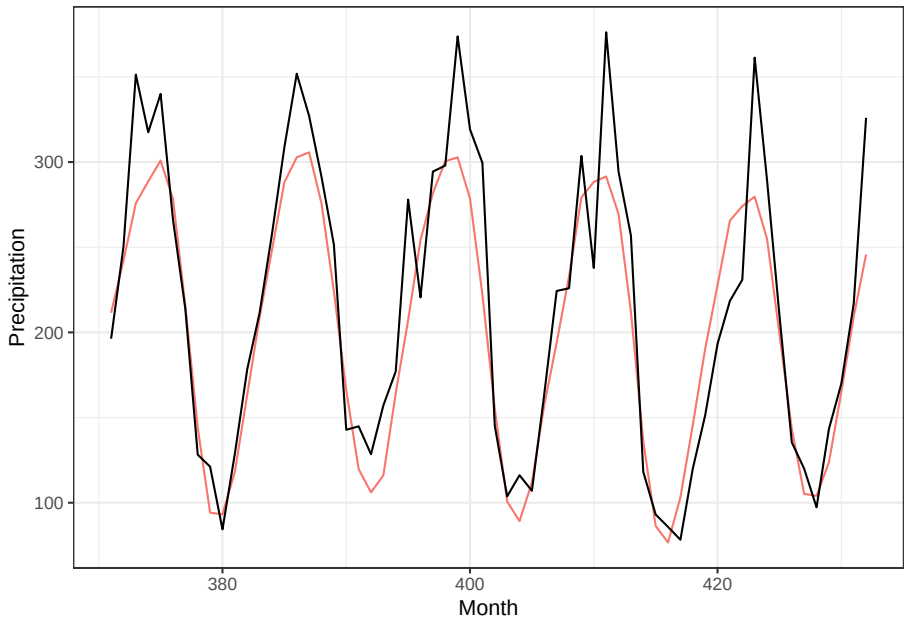
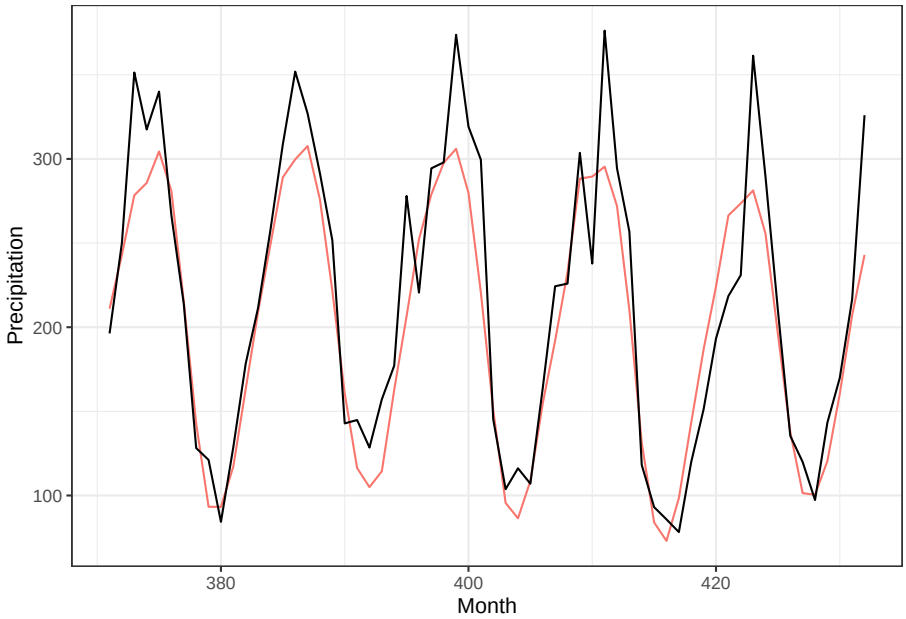
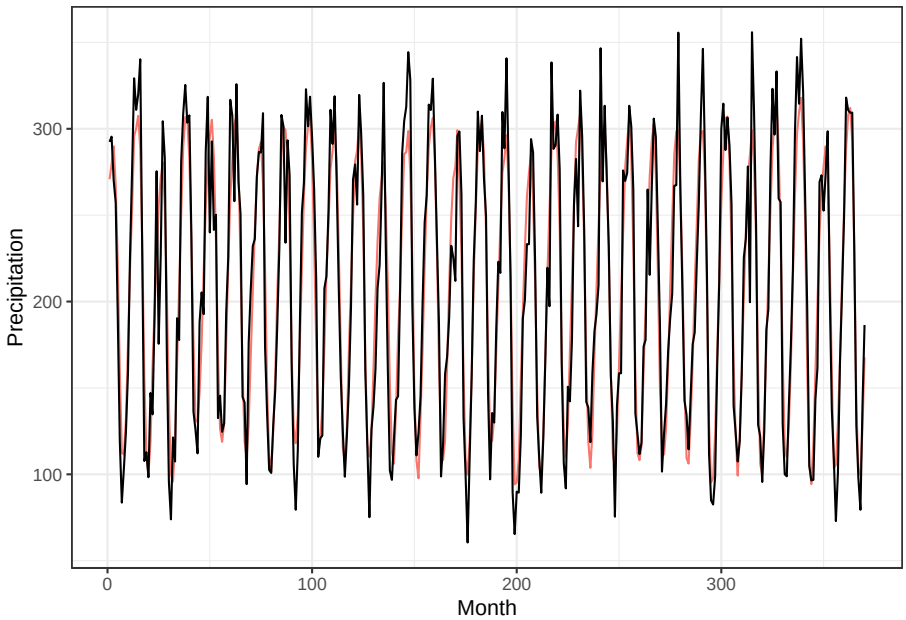


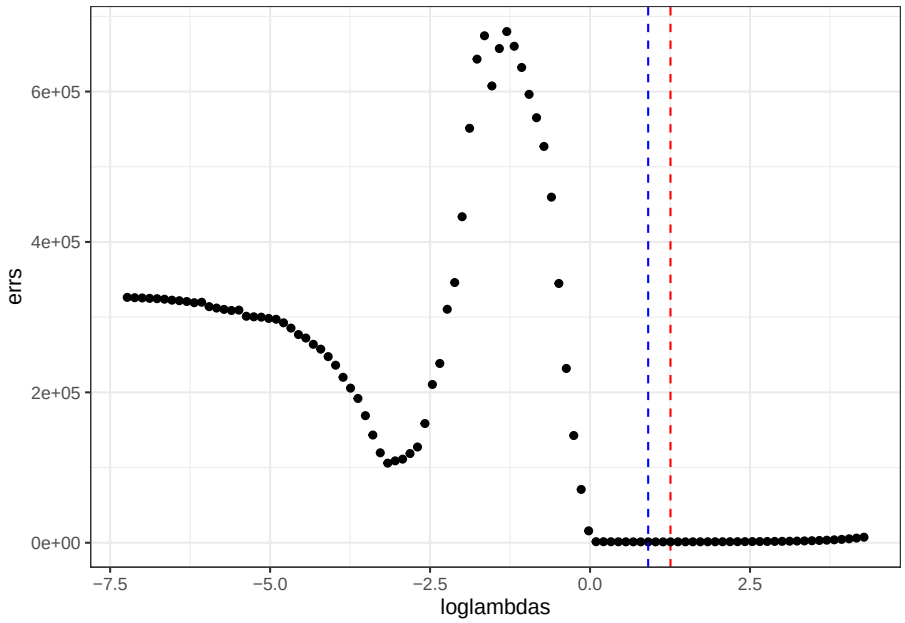
Figure 7.5: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.



```
## [1] 1214.489
```


Precipitation
300
200
100
380
400
420
Month

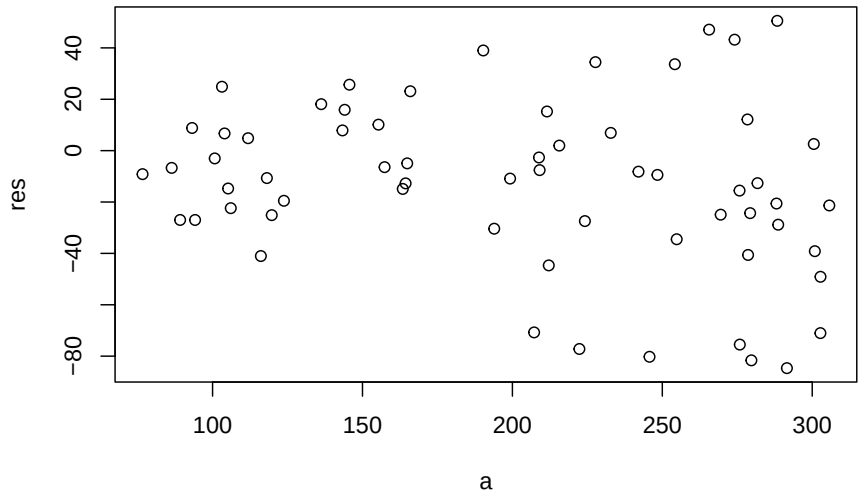

Precipitation
300
200
100
0
100
200
300
Month

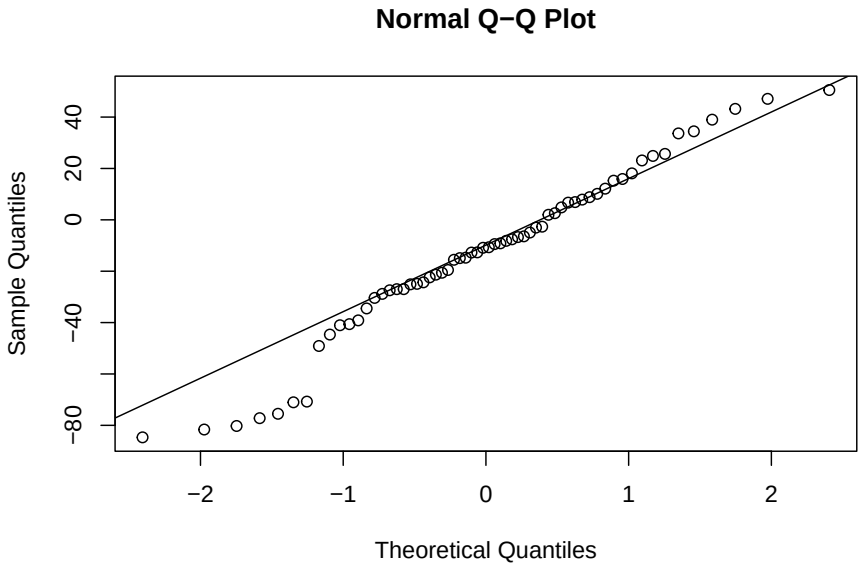


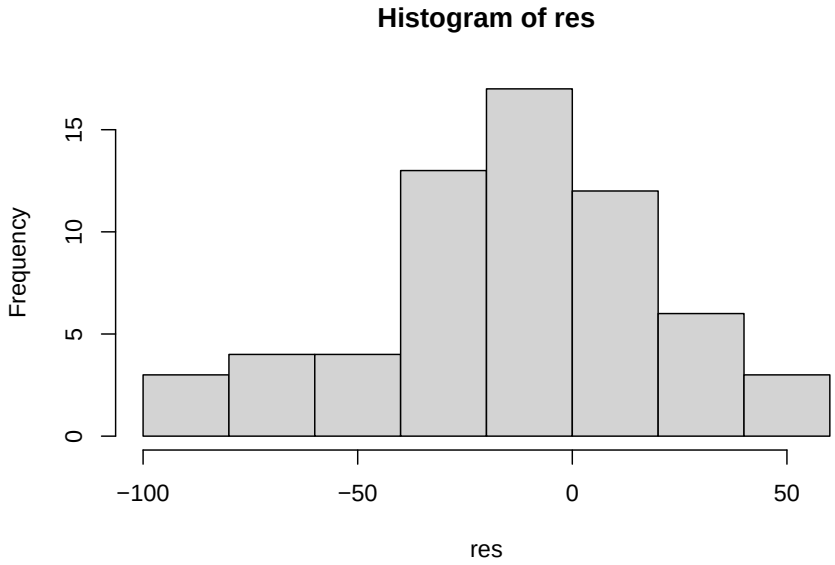
Over the more than 5 years of validation data the model predicts the seasonal pattern of the precipitation time series quite well, but constantly fails to predict the higher values of precipitation. The MSE is `mse_full` and the RSME `sqrt(mse_full)`.

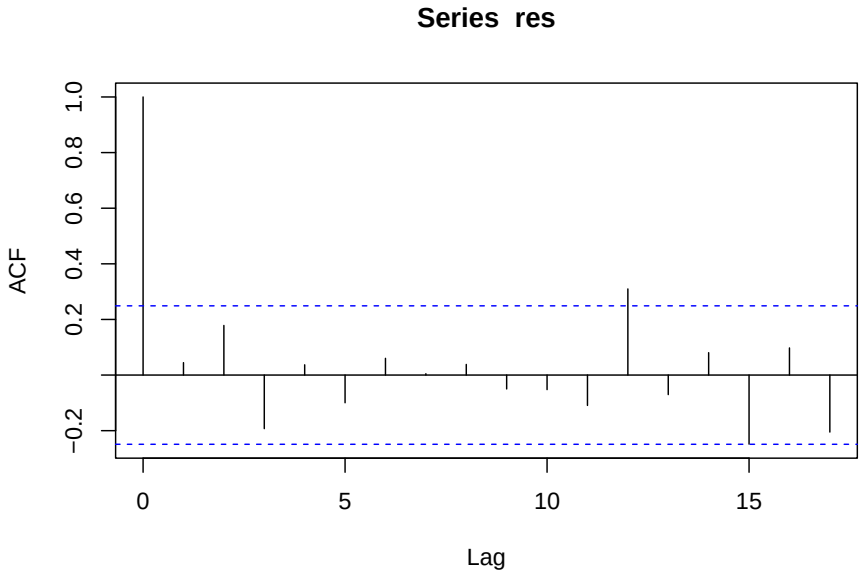
7.6 Summary

We fitted a LASSO model for predicting the mean precipitation in the Central Amazon Basin and used a 5-fold blocked Cross Validation approach to find the optimal level of regularization. After training the model we evaluated its performance on a separate validation set that was not used in the training process. The model shows predicting capabilities but misses out on higher values of the precipitation target. It also misses on rapid changes and in general underfits the data. This may be due to the choice of blocked cross validation. Locations with higher variability get included in the model more easily and are not necessarily geographically close.


res
a
100
150
200
250
300
−80
−40
0
20
40

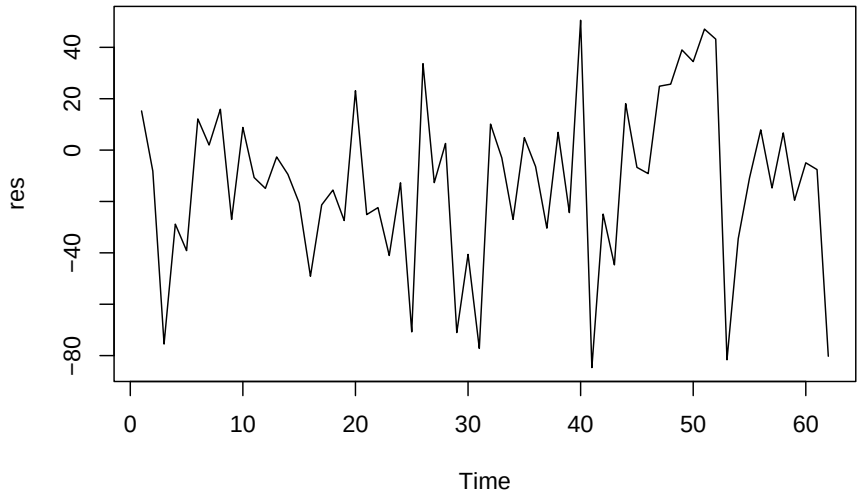

Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−80
−40
0
20
40

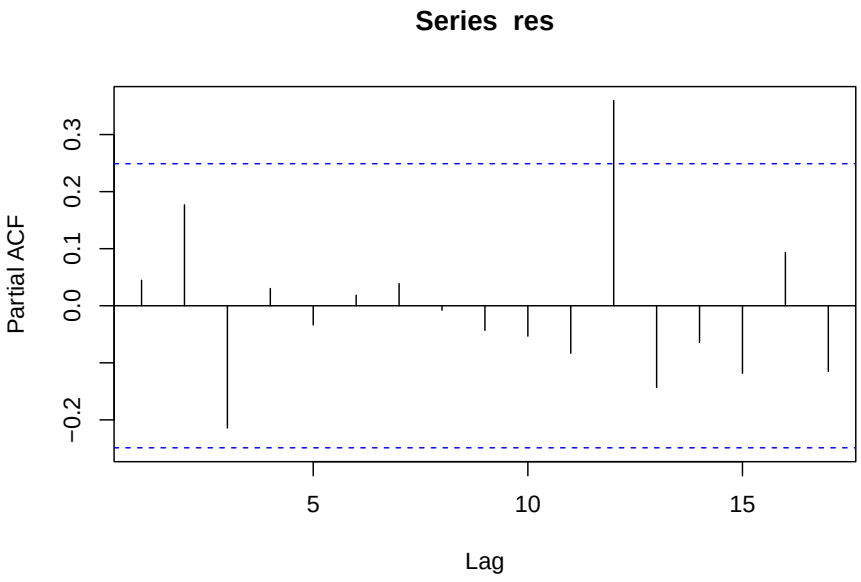


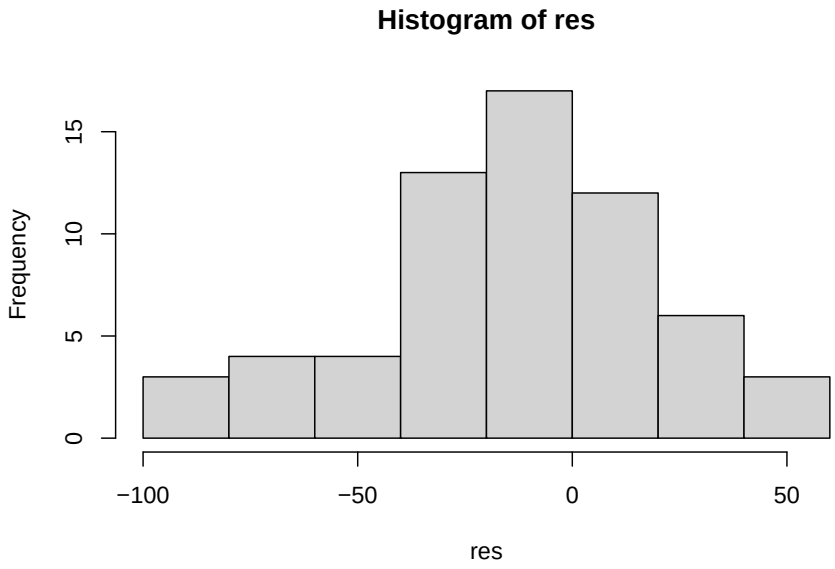



```
##
##  Box-Ljung test
##
## data:  res
```

```
## X-squared = 26.427, df = 20, p-value = 0.1522
```






Histogram of res
Frequency
0
5
10
15
−100
−50
0
50
res

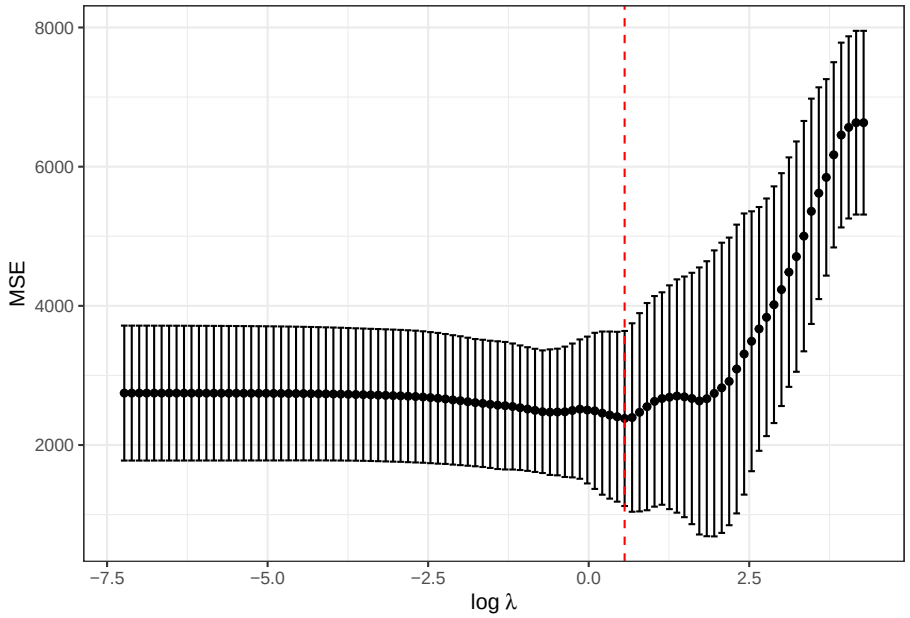
Chapter 8

lasso center

8.1 LASSO model

We fit a LASSO model on the precipitation and SST data. The precipitation target is the monthly mean precipitation in the Central Amazon Basin, the SST data are monthly temperatures over the globe. We use a 5-fold CV approach to find an optimal lambda. Each fold consists of 5 consecutive years of training data followed by 2 years of test data. In each fold we fit a LASSO model on a set of predetermined lambda values and choose the lambda that minimizes the MSE on the test set in that fold. After determining the best lambda in each fold we choose the lambda that minimizes the MSE over all folds and refit the model to the complete training data. Afterwards we evaluate the fitted model on a separate validation set with 5 years length which was not included in the training phase.

8.2 Error plots



?? shows the results from the 5 fold-CV plotted for each lambda. The lambdas are given on the log scale. The upper and lower bars indicate mean MSE +/- one standard deviation from the mean MSE. We note that the upper and lower bars are quite wide indicating big differences in MSE for the different folds. We therefore also inspect the MSE for each individual fold.

At first glance 6.1 shows that the MSE for fold 1 and 2 have similar trajectories, the same for 3 and 4, while fold 5 is the only one that only has a local maximum somewhat in the middle of the log lambda range. But fold 5 also has its minimum MSE at a larger regularization value than the other folds which is also reflected in the number of coefficients it includes in the model, as we will see in the coefficient plots. Also obviously the MSE differ greatly in their values as can be seen on the differences of their respective y-axis. While this plot works well for getting an overview of the trajectories we can replot them with a common y-axis to compare their values more easily.

We can see now that fold 2 settles for far larger errors than fold 5 for example. Fold 5 chooses the highest lambda but also has the lowest minimal MSE.

Below we can see the minimum MSE for each fold.

```
## [1] 1160.1141 3665.6846  967.6527 2499.7806 1432.5529
```

And which lambda in the lambda vector resulted in the lowest prediction error on the folds´ test set.

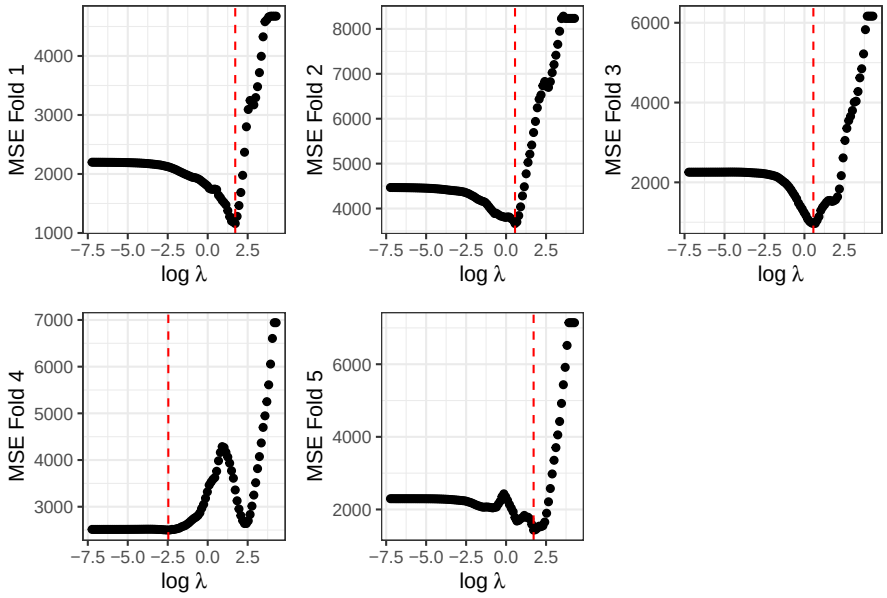


Figure 8.1: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained.

```
## [1] 5.59791854 1.74975027 1.74975027 0.08508338 5.59791854
```

8.3 Coefficient plots

The plots displays the nonzero coefficients in each fold computed for the lambda that minimizes the MSE on the test set in the respective fold. The LASSO chooses among correlated variables only one and discards the others, which can be seen here since the variables chosen are scattered across the map and can but don't have to be close to each other. If we take a look again at **??** we can see that the model includes locations that have high correlations.

8.4 Inspect predictions from each fold

Following we inspect the folds precipitation time series and the predictions made by the model.

In general the model fits the data sufficiently to predict the general form of the time series but misses some modes and is off in the larger values in fold 2.

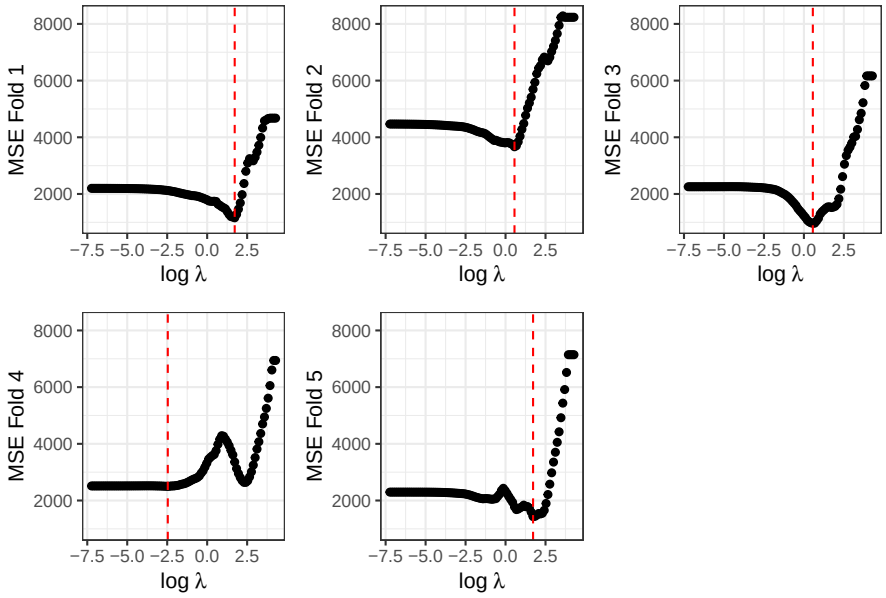


Figure 8.2: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained. See (**ef?**)(fig:err-folg-lasso-og), but this time the y-axis has the same range for all plots.

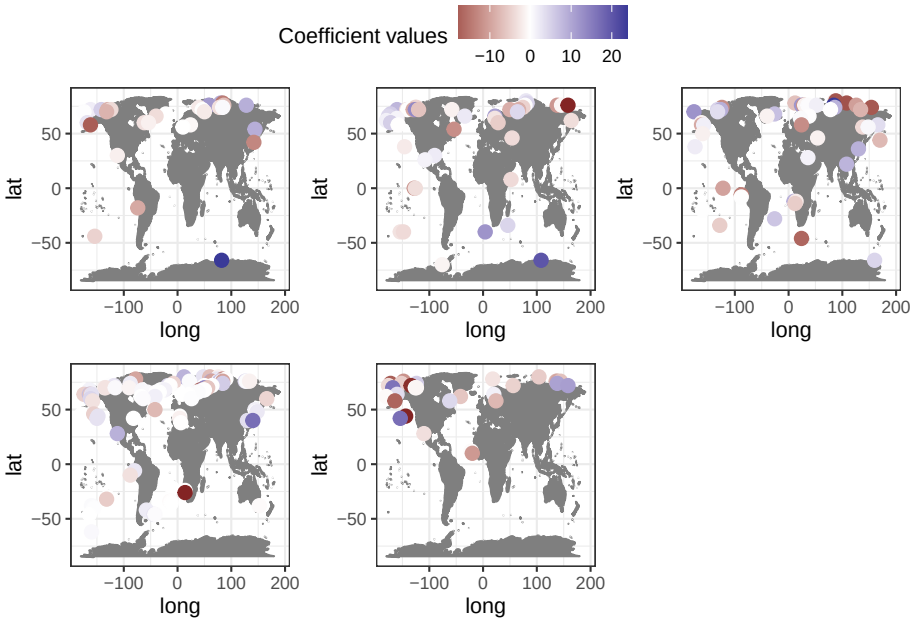


Figure 8.3: Coefficient map plot for the different folds. Longitude and Latitude on the x and y-axis respectively. Positive values are coloured in blue, negative values in red.

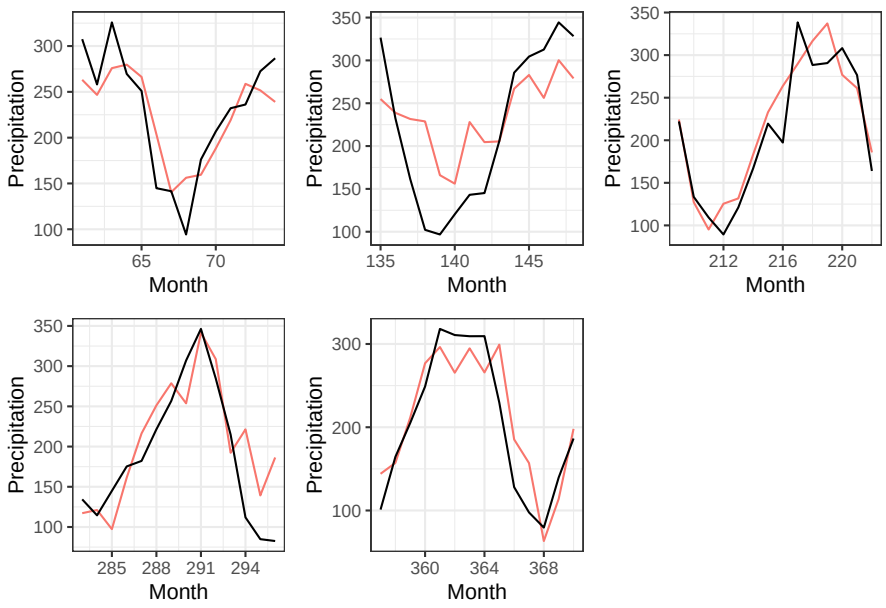


Figure 8.4: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.

Also it does not fit well rapid changes as in fold 3. Therefore it seems that the model generally underfits the data.

8.5 Inspect predictions from best CV-lambda

A common practice is too choose the largest lambda so that it's mean MSE is smaller than the MSE of the lambda that minimizes mean MSE plus one SE. But since in our case the largest lambda that satisfies these criteria is the maximum lambda we choose the lambda with minimum mean MSE instead.

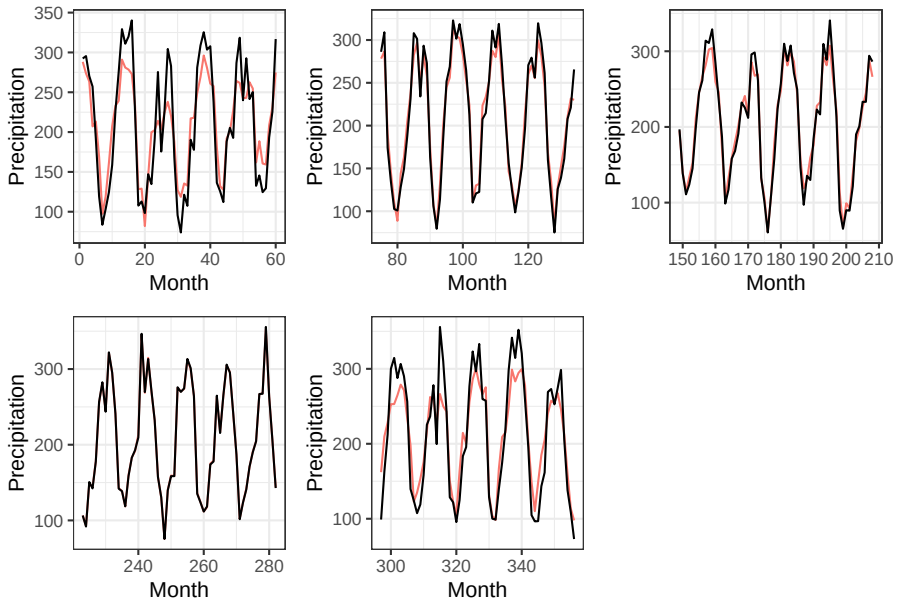
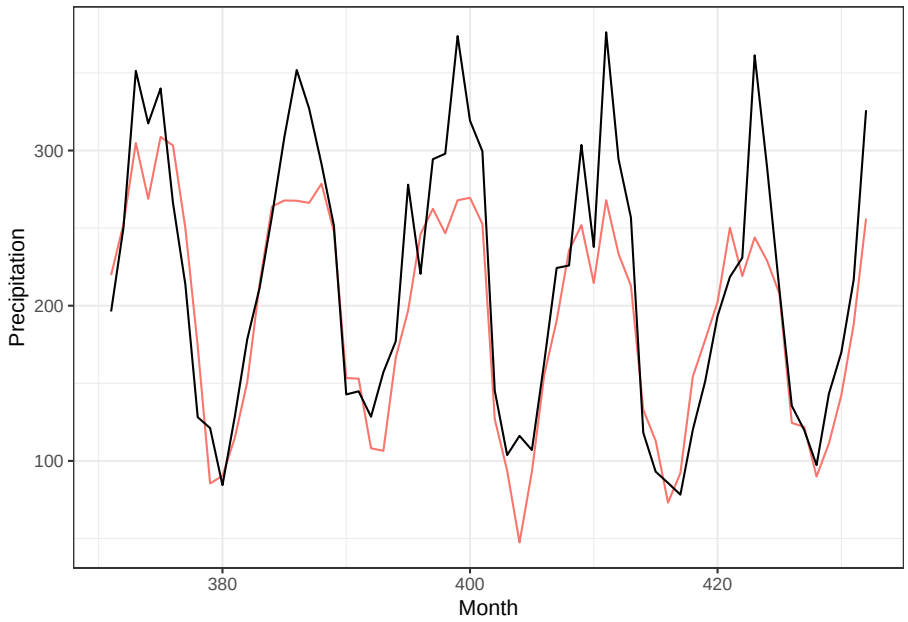
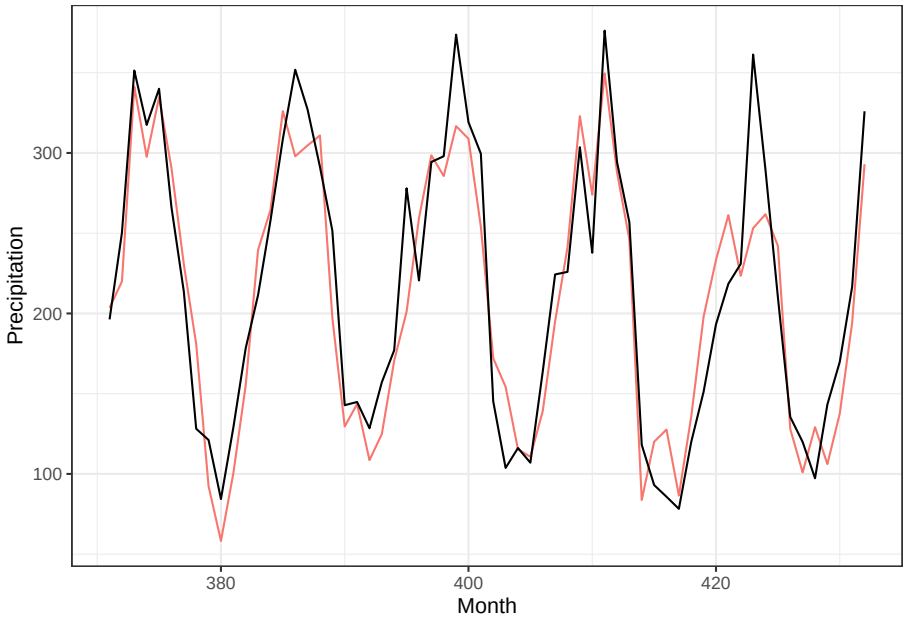
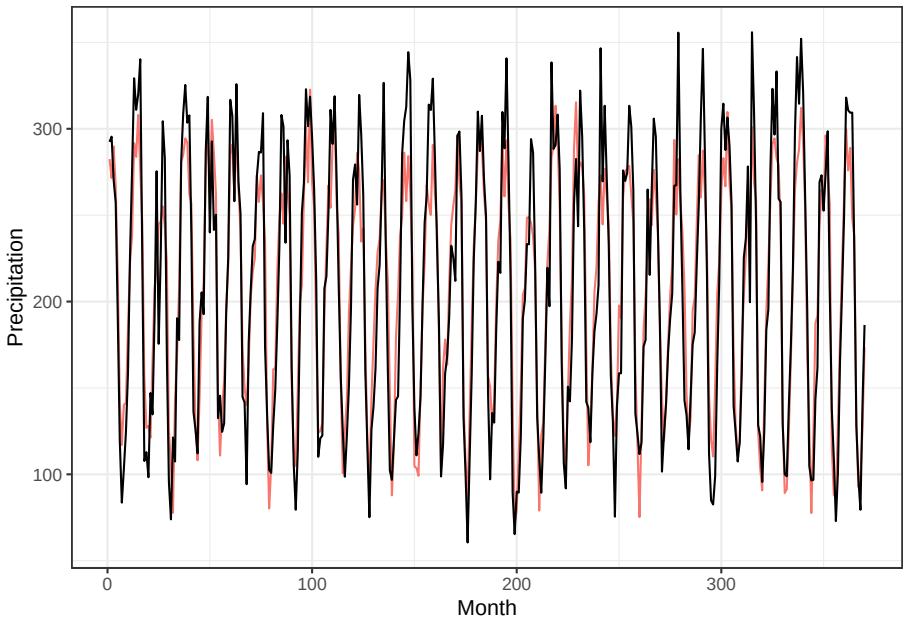


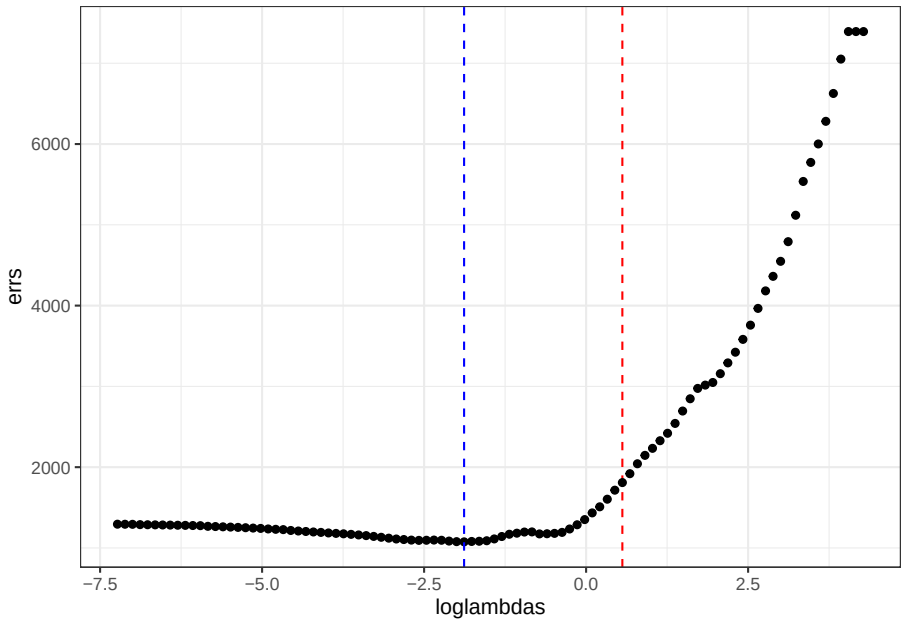
Figure 8.5: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.



```
## [1] 1809.455
```


Precipitation
300
200
100
380
400
420
Month

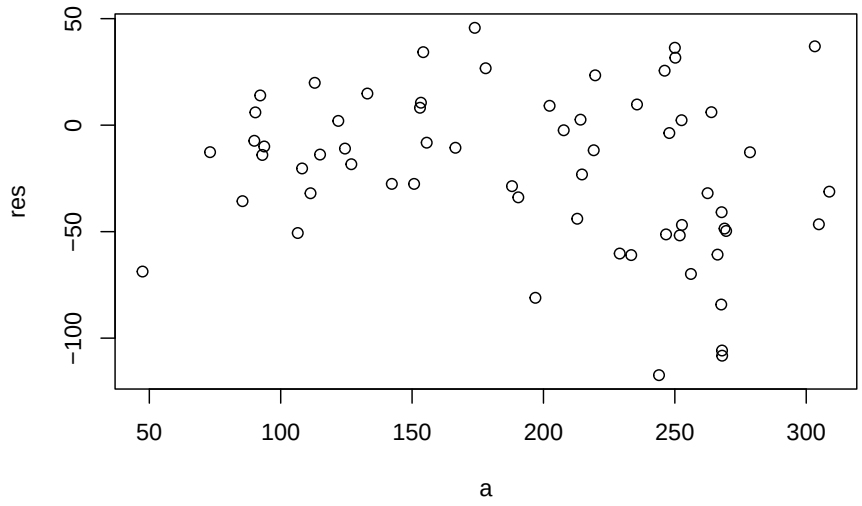

Precipitation
300
200
100
0
100
200
300
Month

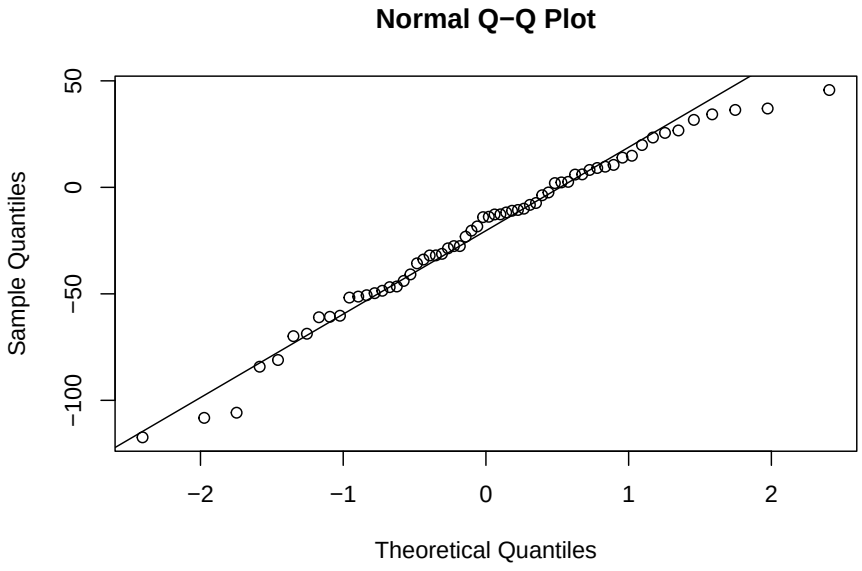


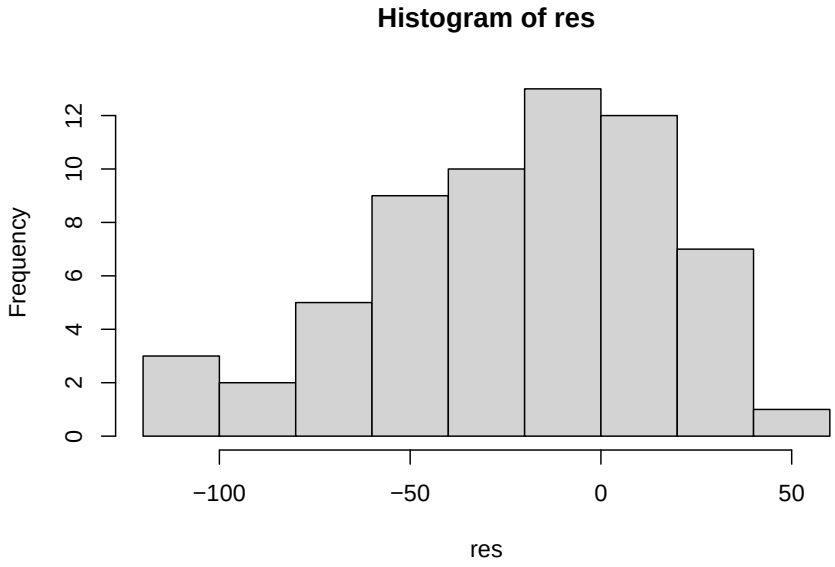
Over the more than 5 years of validation data the model predicts the seasonal pattern of the precipitation time series quite well, but constantly fails to predict the higher values of precipitation. The MSE is `mse_full` and the RSME `sqrt(mse_full)`.

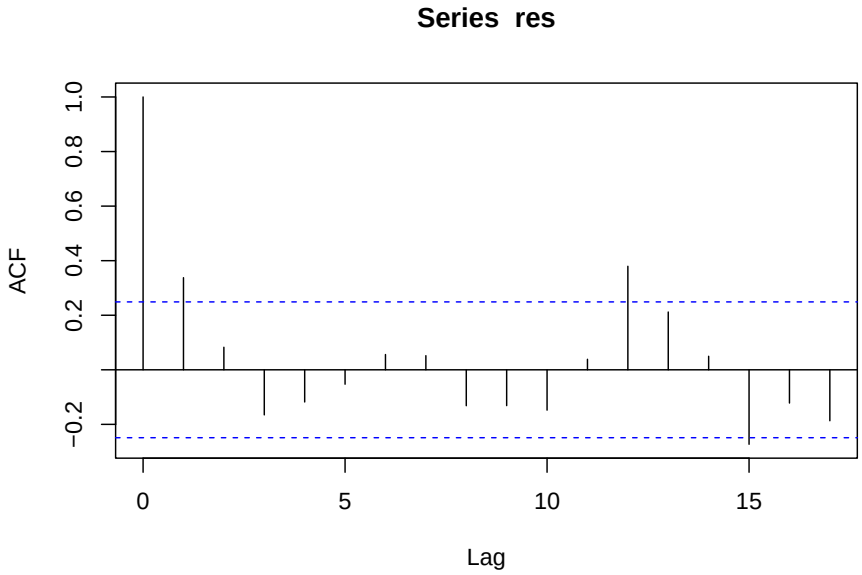
8.6 Summary

We fitted a LASSO model for predicting the mean precipitation in the Central Amazon Basin and used a 5-fold blocked Cross Validation approach to find the optimal level of regularization. After training the model we evaluated its performance on a separate validation set that was not used in the training process. The model shows predicting capabilities but misses out on higher values of the precipitation target. It also misses on rapid changes and in general underfits the data. This may be due to the choice of blocked cross validation. Locations with higher variability get included in the model more easily and are not necessarily geographically close.


res
50
0
−50
−100
50
100
150
200
250
300
a

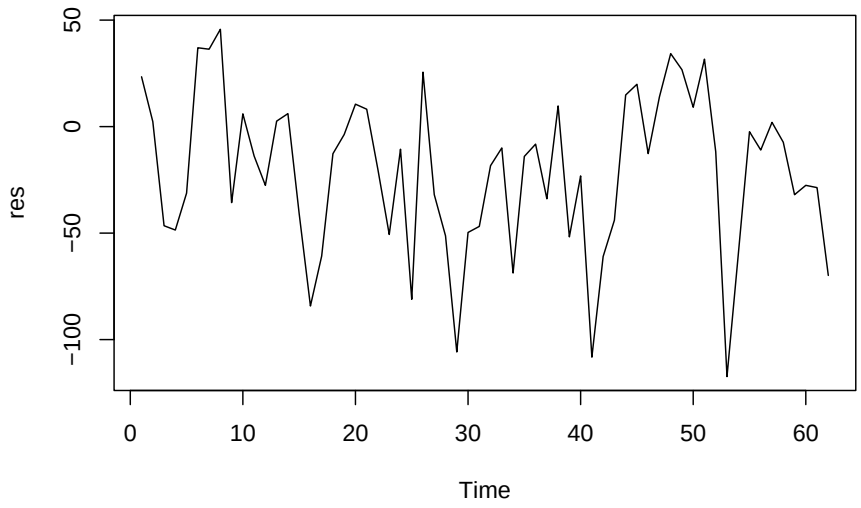

Normal Q−Q Plot
Sample Quantiles
50
0
−50
−100
−2
−1
0
1
2
Theoretical Quantiles

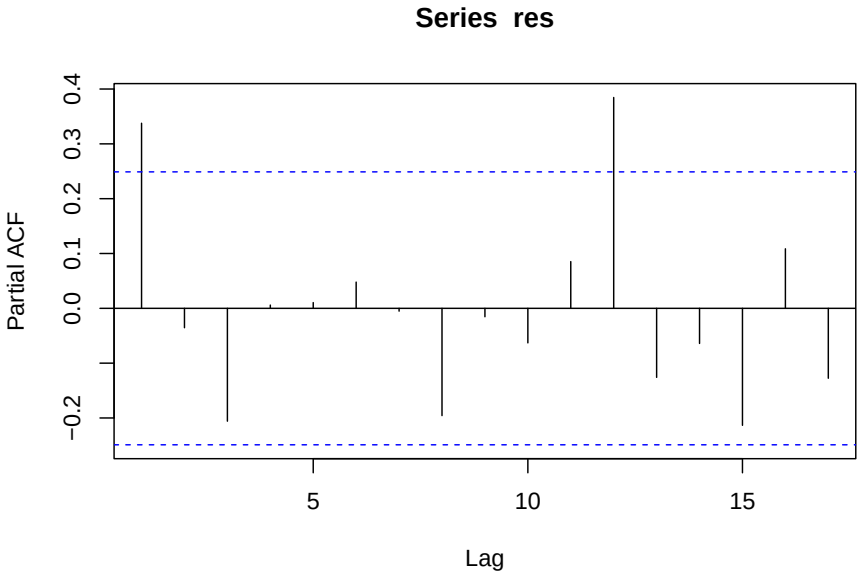


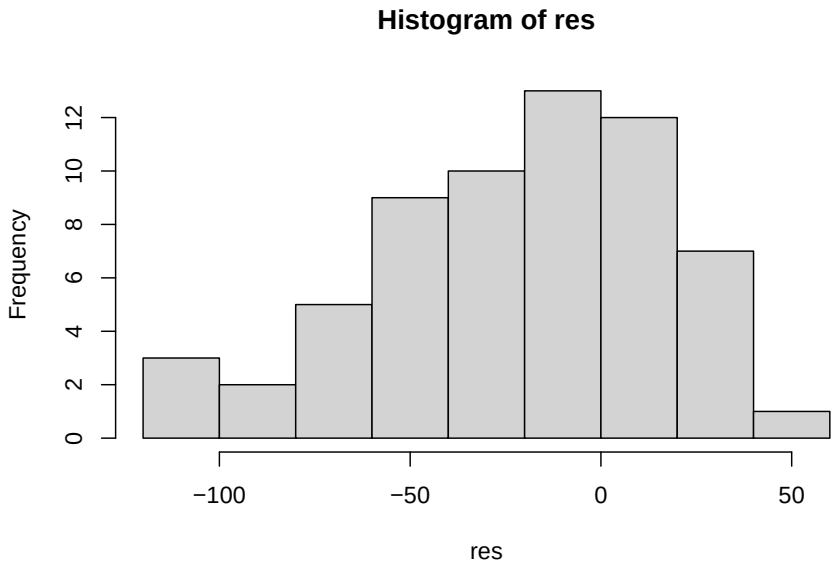



```
##
##  Box-Ljung test
##
## data:  res
```

```
## X-squared = 42.177, df = 20, p-value = 0.002622
```






Histogram of res
Frequency
0
2
4
6
8
10
12
−100
−50
0
50
res

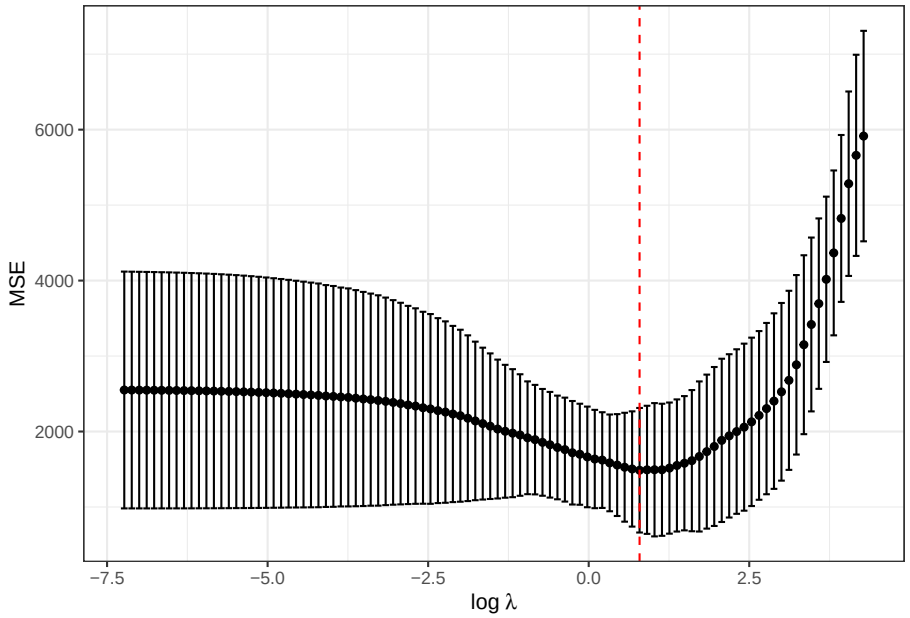
Chapter 9

lasso center

9.1 LASSO model

We fit a LASSO model on the precipitation and SST data. The precipitation target is the monthly mean precipitation in the Central Amazon Basin, the SST data are monthly temperatures over the globe. We use a 5-fold CV approach to find an optimal lambda. Each fold consists of 5 consecutive years of training data followed by 2 years of test data. In each fold we fit a LASSO model on a set of predetermined lambda values and choose the lambda that minimizes the MSE on the test set in that fold. After determining the best lambda in each fold we choose the lambda that minimizes the MSE over all folds and refit the model to the complete training data. Afterwards we evaluate the fitted model on a separate validation set with 5 years length which was not included in the training phase.

9.2 Error plots



?? shows the results from the 5 fold-CV plotted for each lambda. The lambdas are given on the log scale. The upper and lower bars indicate mean MSE +/- one standard deviation from the mean MSE. We note that the upper and lower bars are quite wide indicating big differences in MSE for the different folds. We therefore also inspect the MSE for each individual fold.

At first glance 6.1 shows that the MSE for fold 1 and 2 have similar trajectories, the same for 3 and 4, while fold 5 is the only one that only has a local maximum somewhat in the middle of the log lambda range. But fold 5 also has its minimum MSE at a larger regularization value than the other folds which is also reflected in the number of coefficients it includes in the model, as we will see in the coefficient plots. Also obviously the MSE differ greatly in their values as can be seen on the differences of their respective y-axis. While this plot works well for getting an overview of the trajectories we can replot them with a common y-axis to compare their values more easily.

We can see now that fold 2 settles for far larger errors than fold 5 for example. Fold 5 chooses the highest lambda but also has the lowest minimal MSE.

Below we can see the minimum MSE for each fold.

```
## [1]  652.7684 2407.0244 1256.4045 2007.5454  624.0219
```

And which lambda in the lambda vector resulted in the lowest prediction error on the folds´ test set.

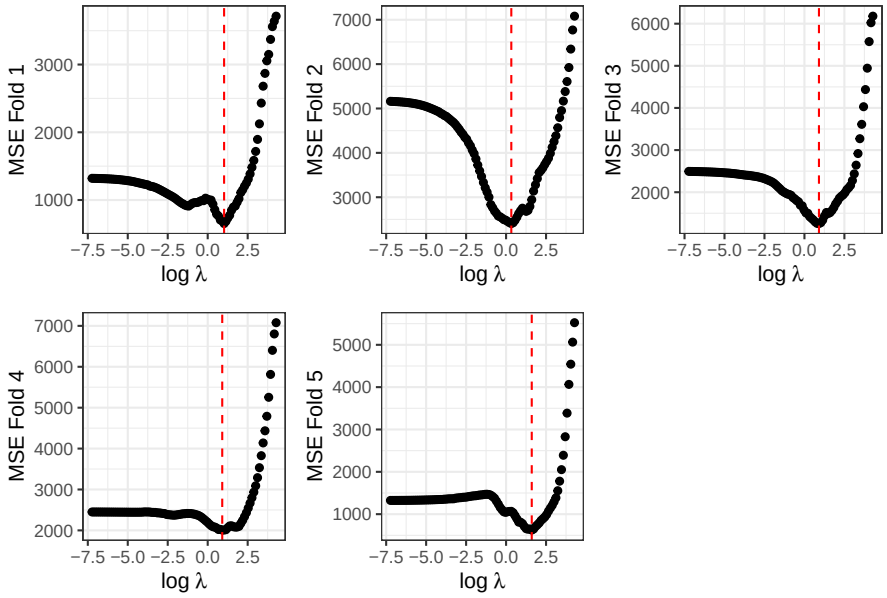


Figure 9.1: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained.

```
## [1] 2.786097 1.386647 2.480226 2.480226 4.983352
```

9.3 Coefficient plots

The plots displays the nonzero coefficients in each fold computed for the lambda that minimizes the MSE on the test set in the respective fold. The LASSO chooses among correlated variables only one and discards the others, which can be seen here since the variables chosen are scattered across the map and can but don't have to be close to each other. If we take a look again at **??** we can see that the model includes locations that have high correlations.

9.4 Inspect predictions from each fold

Following we inspect the folds precipitation time series and the predictions made by the model.

In general the model fits the data sufficiently to predict the general form of the time series but misses some modes and is off in the larger values in fold 2.

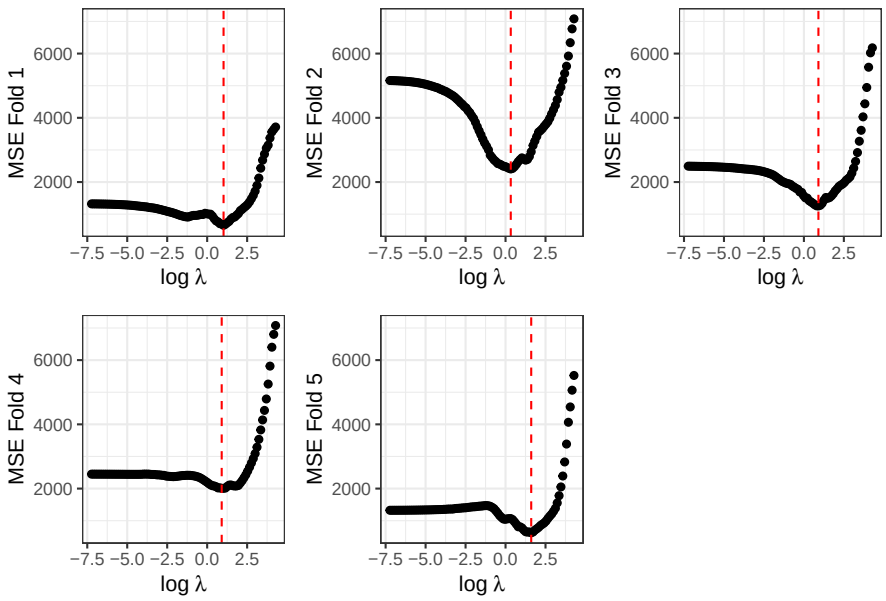


Figure 9.2: MSE of the CV for the different lambda values on the a log scale. The red dotted line shows the lambda for which minimum MSE was obtained. See (**ef?**)(fig:err-folg-lasso-og), but this time the y-axis has the same range for all plots.

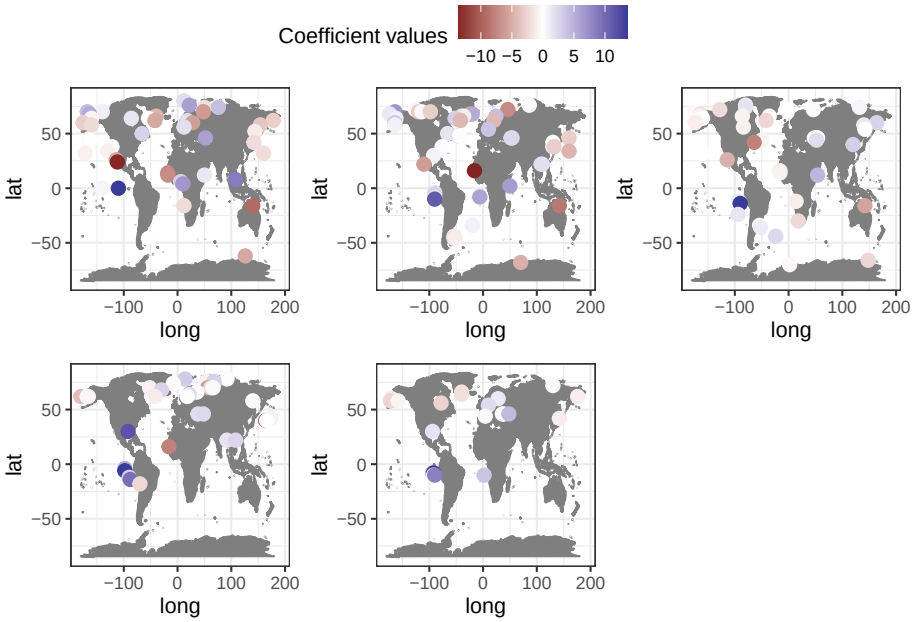


Figure 9.3: Coefficient map plot for the different folds. Longitude and Latitude on the x and y-axis respectively. Positive values are coloured in blue, negative values in red.

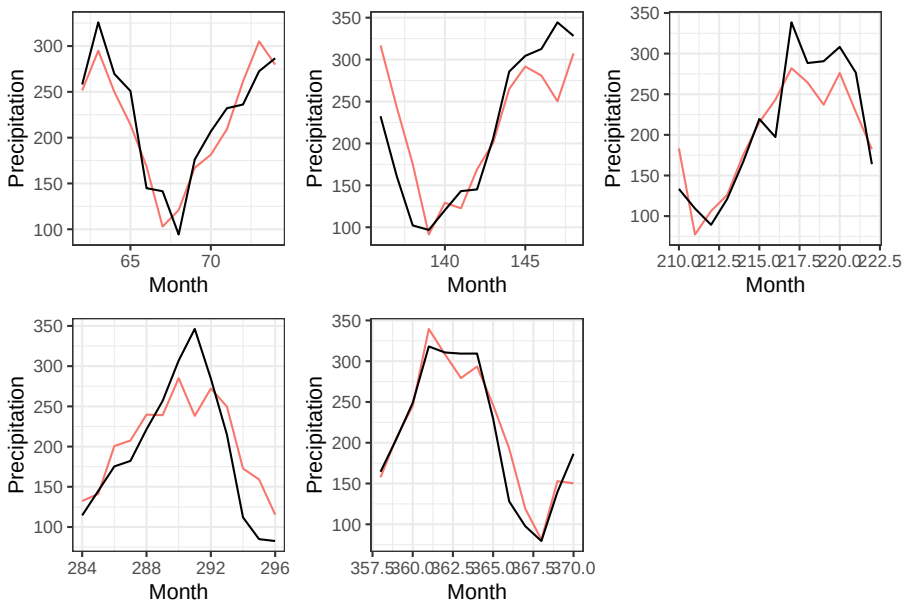


Figure 9.4: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.

Also it does not fit well rapid changes as in fold 3. Therefore it seems that the model generally underfits the data.

9.5 Inspect predictions from best CV-lambda

A common practice is too choose the largest lambda so that it‘s mean MSE is smaller than the MSE of the lambda that minimizes mean MSE plus one SE. But since in our case the largest lambda that satisfies these criteria is the maximum lambda we choose the lambda with minimum mean MSE instead.

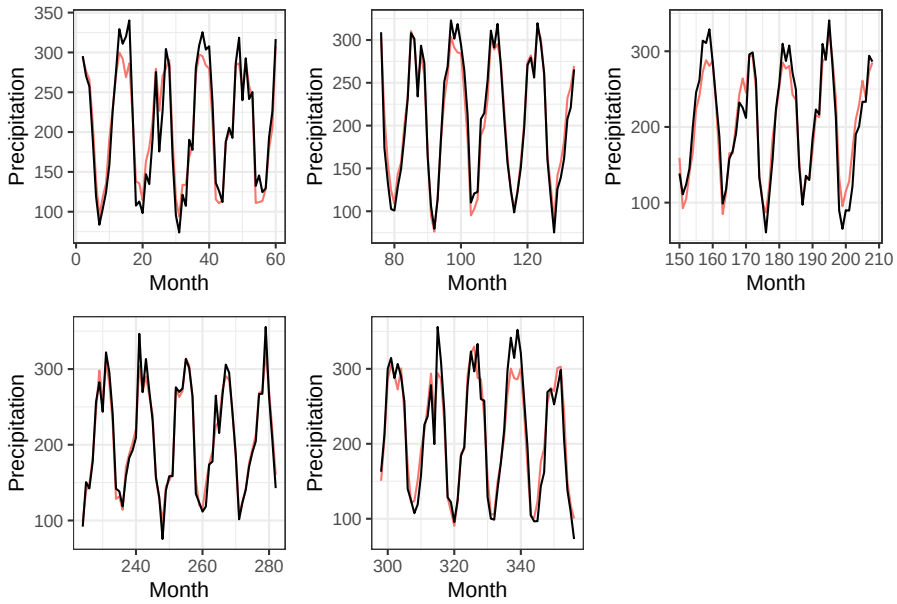
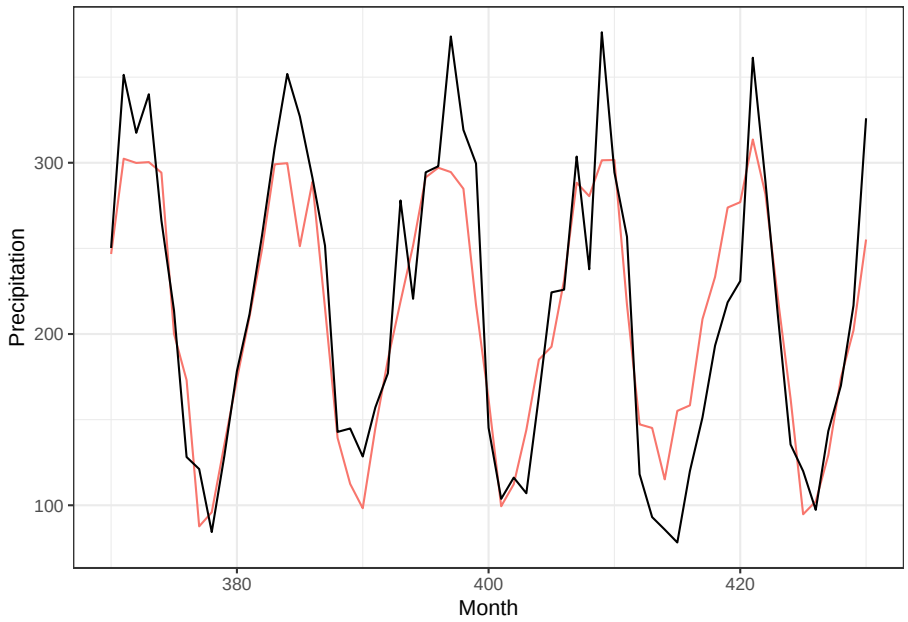
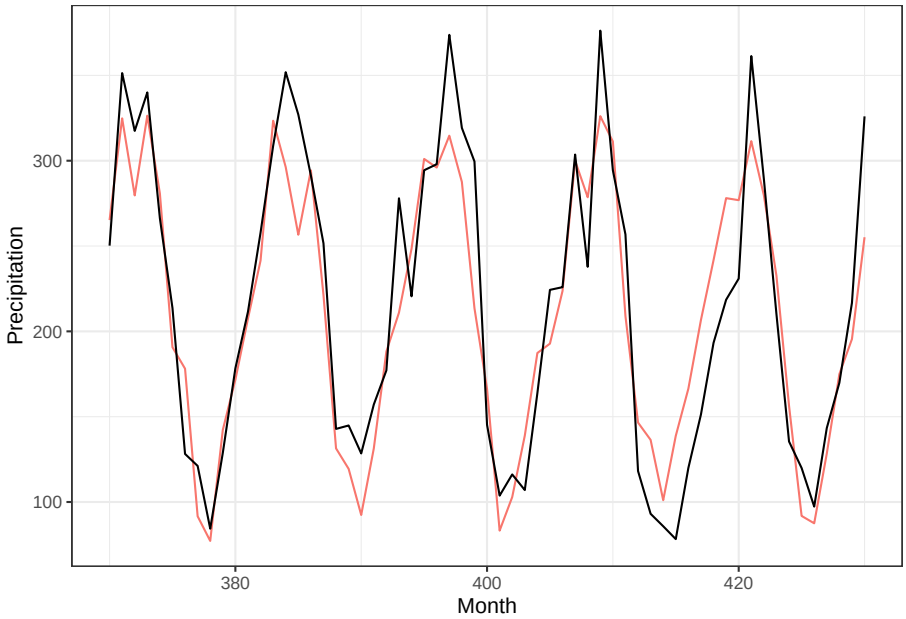
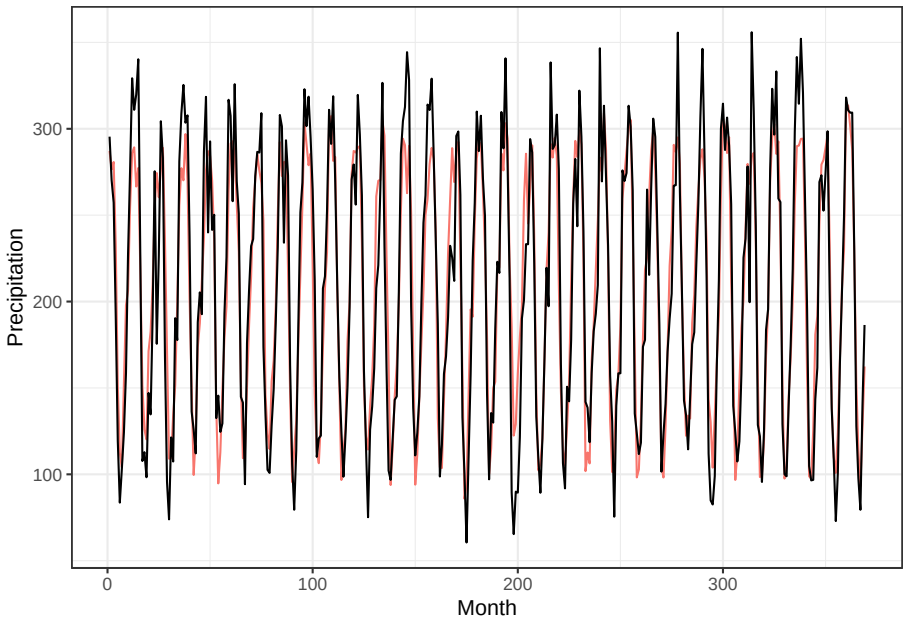


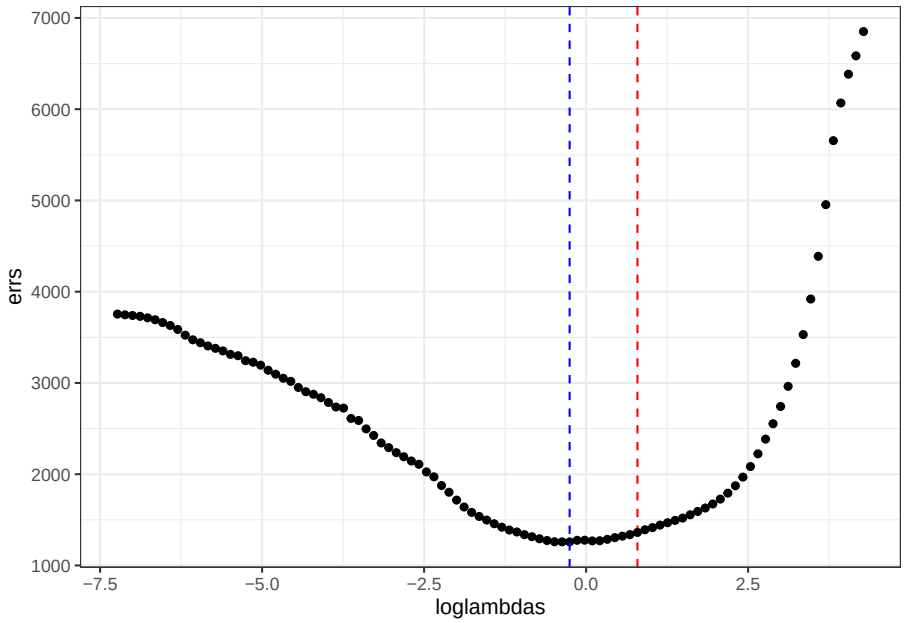
Figure 9.5: Precipitation prediction and target values in the test set in each fold. Predictions in red and target values in black.



```
## [1] 1361.82
```


Precipitation
300
200
100
380
400
420
Month

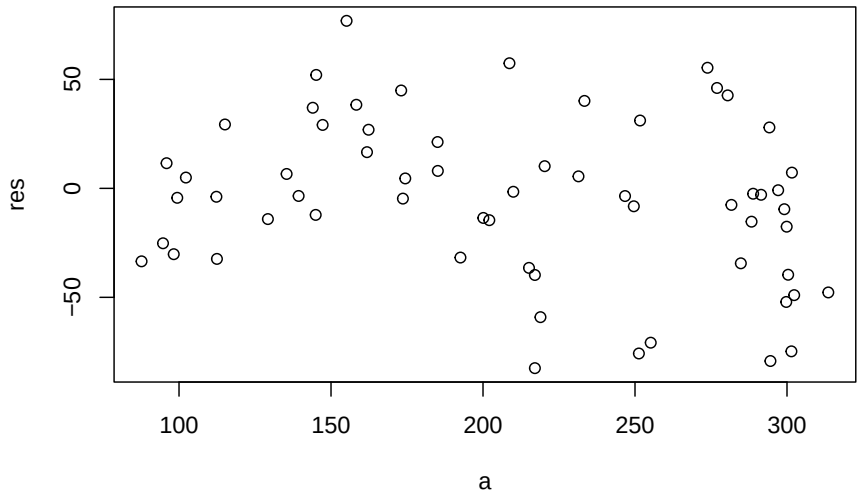

Precipitation
300
200
100
0
100
200
300
Month

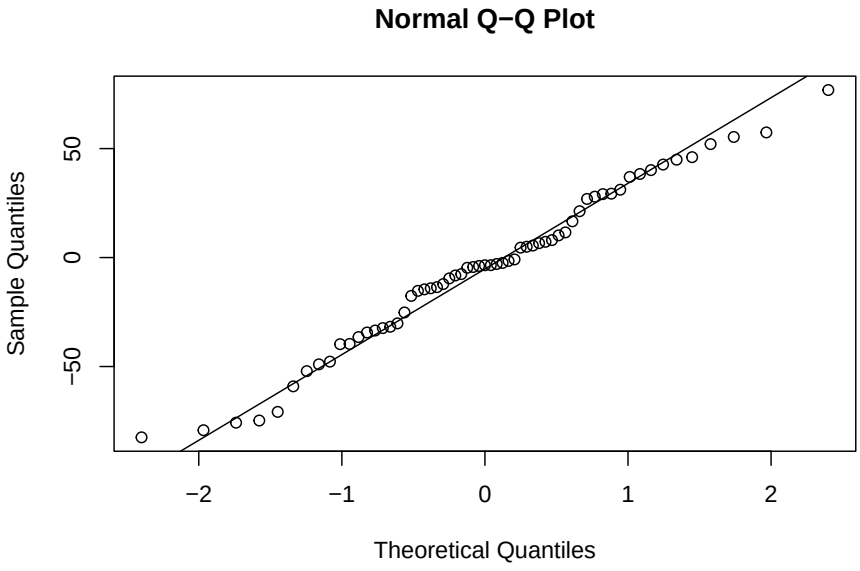


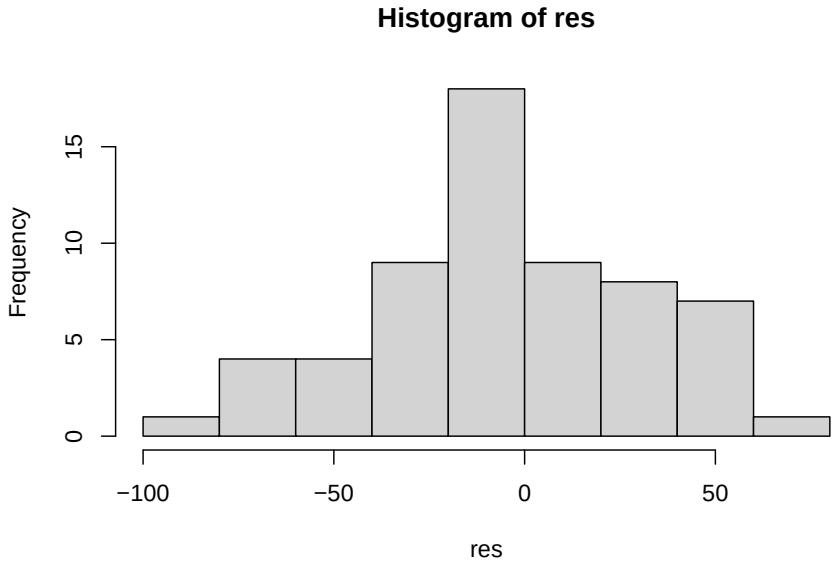
Over the more than 5 years of validation data the model predicts the seasonal pattern of the precipitation time series quite well, but constantly fails to predict the higher values of precipitation. The MSE is `mse_full` and the RSME `sqrt(mse_full)`.

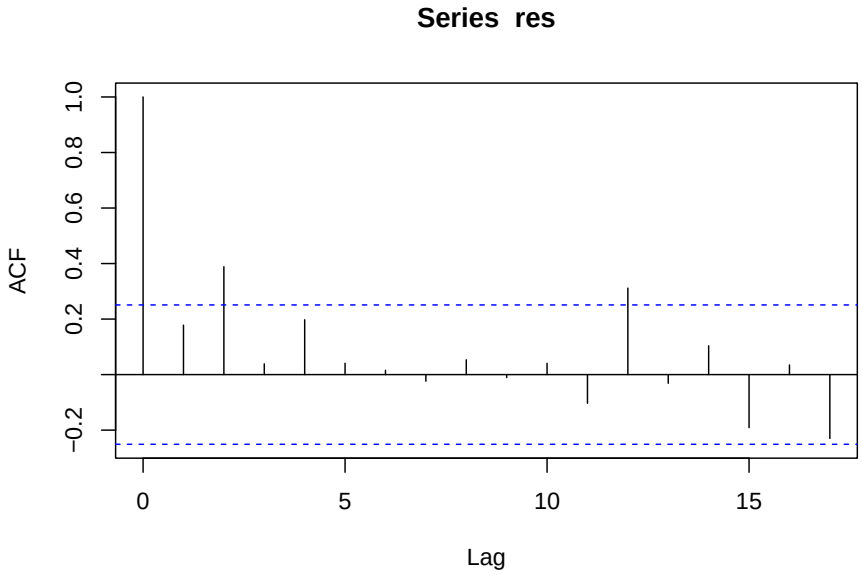
9.6 Summary

We fitted a LASSO model for predicting the mean precipitation in the Central Amazon Basin and used a 5-fold blocked Cross Validation approach to find the optimal level of regularization. After training the model we evaluated its performance on a separate validation set that was not used in the training process. The model shows predicting capabilities but misses out on higher values of the precipitation target. It also misses on rapid changes and in general underfits the data. This may be due to the choice of blocked cross validation. Locations with higher variability get included in the model more easily and are not necessarily geographically close.


res
50
0
−50
100
150
200
250
300
a

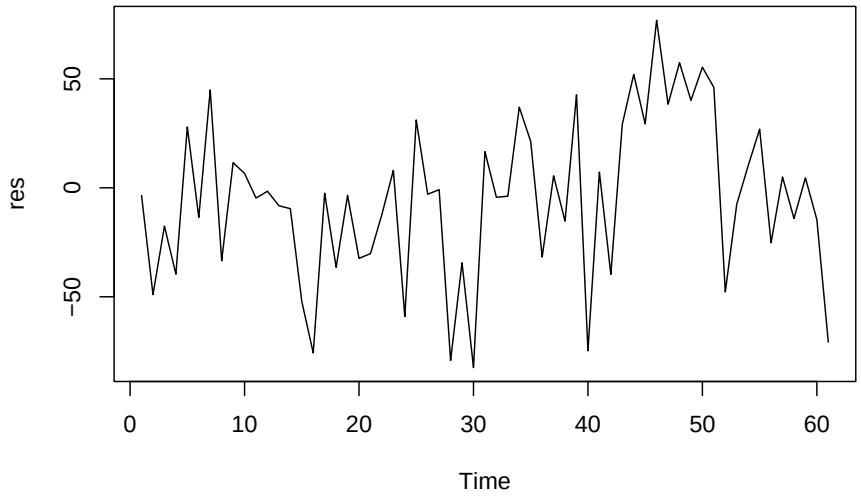

Normal Q−Q Plot
Sample Quantiles
50
0
−50
−2
−1
0
1
2
Theoretical Quantiles

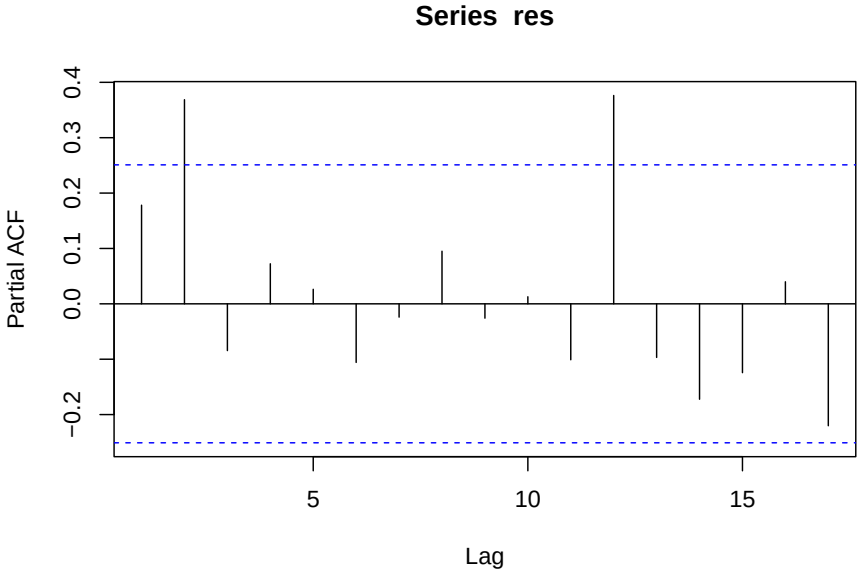


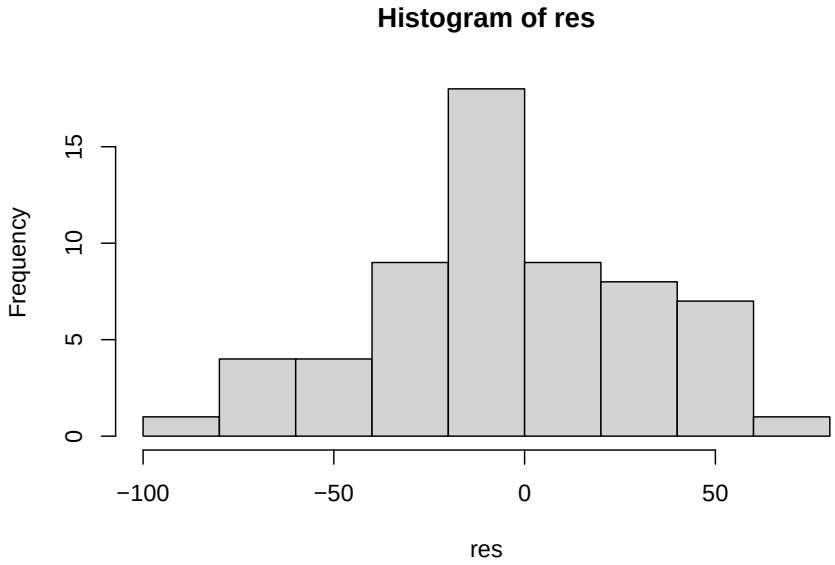



```
##
##  Box-Ljung test
##
## data:  res
```

```
## X-squared = 36.013, df = 20, p-value = 0.01533
```






Histogram of res
Frequency
0
5
10
15
−100
−50
0
50
res

Chapter 10

The fused lasso

10.1 General

As expected and seen in the results, the different LASSO models choose single SST regions as predictors as opposed to whole regions. Since the LASSO only regularizes the magnitude of coefficients but ignores their ordering. We therefore use the so-called *fused lasso* as implemented in the *genlasso* package and the respective *fusedlasso* function. (R. Tibshirani et al. (2005), Taylor B. Arnold and Tibshirani (2020)) The fused lasso is a generalization of the lasso for problems with features that can be ordered in a meaningful way. It penalizes not only the coefficients' L_1-norm but also their differences given their ordering, introducing sparsity in both of them R. Tibshirani et al. (2005). In our case the fused lasso thus penalizes the differences of SST coefficients that are close to each other.

The fused LASSO as implemented in *genlasso* package solves the problem:

$$\min_{\beta} 1/2 \sum_{i=1}^{n} (y_i - x_i^T \beta_i)^2 + \lambda \sum_{i,j \in E} |\beta_i - \beta_j| + \gamma \cdot \lambda \sum_{i=1}^{p} |\beta_i|, \tag{10.1}$$

with x_i being the ith row of the predictor matrix and E is the edge set of an underlying graph. Regularizing $|\beta_i - \beta_j|$, penalizes large differences in close coefficients. In our case "close" means small distances as defined on 2-dimensional longitude/latitude grid. This grid defines a graph that can be used to compute the distances for each location. The third term $\gamma \cdot \lambda \sum_{i=1}^{p} |\beta_i|$, controls the sparsity of the coefficients. $\gamma = 0$ leads to complete fusion of the coefficients (no sparsity) and $\gamma > 0$ introduces sparsity to the solution, with higher values placing more priority on sparsity. $\widehat{\beta}$ is computed as a function of λ, with fixed γ.

10.1.1 Implementation

The summary of the algorithm is taken from the paper proposing the implementation, Taylor B. Arnold and Tibshirani (2016) and the original paper introducing the algorithm R. J. Tibshirani and Taylor (2011). In the fused lasso setting the coefficients $\beta \in \mathbb{R}^p$ can be thought of as nodes of a given undirected Graph G, with edge set $E \subset 1, ..., p^2$. Now lets assume that E has m edges which are enumerated $e_1, ..., e_m$. The fused lasso penalty matrix D is then $m \times p$, where each row corresponds to an edge in E. So when $e_l = (i, j)$, we write l_{th} row of D as

$$D_l = (0, ..., -1, ..., 1, ...) \in \mathbb{R}^p, \tag{10.2}$$

meaning D_l has all zeros except for the the i_{th} and j_{th} location.

(10.1) is solved by a dual path algorithm that was proposed by Taylor B. Arnold and Tibshirani (2016) for different use cases of the (sparse) fused lasso. They describe the dual path algorithm based on the notation of the generalized lasso problem R. J. Tibshirani and Taylor (2011):

$$\hat{\beta} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|D\beta\|_1, \tag{10.3}$$

where $y \in \mathbb{R}^n$ is the vector of the outcome, $X \in \mathbb{R}^{n \times p}$ a predictor matrix, $D \in \mathbb{R}^{m \times p}$ denotes a penalty matrix, and $\lambda \geq 0$ is a regularization parameter. The dual path algorithm solves not the primal but the dual solution of the problem and computes the solution for a whole path instead of single values of λ. Hence the "dual" and "path" that make up the name. Taylor B. Arnold and Tibshirani (2016) argue that the strength of the original algorithm R. J. Tibshirani and Taylor (2011) lays in the fact that it applies to a unified framework in which D can be a general penalty matrix. Let's consider the case when $X = I$ and $rank(X) = p$ (this is called the "signal approximator" case), the dual problem of (10.3) is then:

$$\hat{u} \in \underset{u \in \mathbb{R}^\omega}{\arg\min} \frac{1}{2} \left\| y - D^T u \right\|_{\frac{2}{2}} \text{ subject to } \|u\|_\infty \leq \lambda. \tag{10.4}$$

The primal and dual solutions, $\hat{\beta}$ and $\hat{u}$ are related by:

$$\hat{\beta} = y - D^T \hat{u}. \tag{10.5}$$

While the primal solution is unique, this does not need to be the case for dual solution (note the element notation in (10.4). The dual path algorithm starts at $\lambda = \infty$ and computes the path until $\lambda = 0$. Conceptually the algorithm keeps track of the coordinates of the dual solutions it computed for each lambda $\hat{u}(\lambda)$. The solutions are equal to $\pm\lambda$, meaning they lie on the boundary of the region $[-\lambda, \lambda]$. Along the path it computes the critical values of λ, $\lambda_1 \geq \lambda_2, ...$, at which the coordinates of these solutions hit or leave the boundary.

There are two algorithms described in the paper and the various specialized implementations that can increase efficiency depending on the use cases. This depends on X, and/or the special structure of D. Algorithm 1 handles the $X = I$ case and Algorithm 2 the general X case. As we introduced the dual in (10.4), it assumed $X = I$, which is not satisfied in our case. For the general X case the problem formulation can be rewritten so that the formula only changes D and y to $\tilde{D}$ and $\tilde{y}$ and then the same Algorithm can be applied. $\tilde{D} = DX^+$ and $\tilde{y} = XX^+y$, where X^x is the Moore-Penrose pseudoinverse of $X \in \mathbb{R}^{n \times p}$. Algorithm 2 therefore transforms X and y in a certain way and then applies Algorithm 1 to the transformed problem. Its also easy to see that in our case $p > n$ and X is column rank deficient. They solve this by adding a small fixed $l_2 $penalty to the original problem,which leads to:

$$\text{minimize}_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|D\beta\|_1 + \varepsilon\|\beta\|_2^2, \tag{10.6}$$

and this is the same as

$$\underset{\beta}{\text{minimize}} \frac{1}{2}\left\|y^* - (X^*)\,\beta\right\|_2^2 + \lambda\|D\beta\|_1, \tag{10.7}$$

with $y^* = (y, 0)^T$ and $X^* = \begin{bmatrix} x \\ \varepsilon \cdot I \end{bmatrix}$. Because $\text{rank}\,(X^*) = p$, again it is possible to apply one of the algorithms. Instead of solving linear systems in each, we can apply a QR decomposition that can be updated in a neat way to avoid solving the complete linear system in each step. See the appendix of Taylor B. Arnold and Tibshirani (2016), for details. Special care has to be taken though for general X and certain D. "Blindly" applying the algorithms then would lead to a large drop in relative efficiency. Since $\tilde{D} = DX^+$ destroys the special structures that are present in the penalty matrix special implementations are constructed as well for our case with general X and D coming from the sparse fused lasso.

It can be shown that the computational costly steps in the algorithm reduce to solving linear systems of the form $DD^T = Dc$. When D is the oriented incidence matrix of a graph, D will be sparse but can also be rank deficient

when the graph has more edges than nodes, hence $m > p$. It can be shown that for sparse undetermined systems it is possible to find an arbitrary solution (here called the *basic* solution), but computing the solution with minimum l_2 norm is a lot more difficult in general. It is possible though to derive the minimum l_2 norm solution from the basic solution. In the case of penalty matrices that come from a graph the structure of D can be used to improve efficiency when When D is the incidence matrix of a graph, then $D^T D$ is the Laplacian matrix of G. The Laplacian linear systems are then solved using a sparse Cholesky decomposition. For further details of the steps used in our case refer to Section 4 and in Taylor B. Arnold and Tibshirani (2016).

Chapter 11

References

Arnold, Taylor B., and Ryan J. Tibshirani. 2020. *Genlasso: Path Algorithm for Generalized Lasso Problems*. https://CRAN.R-project.org/package=genlasso.

Arnold, Taylor B, and Ryan J Tibshirani. 2016. "Efficient Implementations of the Generalized Lasso Dual Path Algorithm." *Journal of Computational and Graphical Statistics* 25 (1): 1–27.

Ciemer, Catrin, Lars Rehm, Juergen Kurths, Reik V Donner, Ricarda Winkelmann, and Niklas Boers. 2020. "An Early-Warning Indicator for Amazon Droughts Exclusively Based on Tropical Atlantic Sea Surface Temperatures." *Environmental Research Letters* 15 (9): 094087.

Climate Impact Research (PIK) e. V., Potsdam Institute for. 2021. "Potsdam Insitute for Climate Impact Research." 2021. https://www.pik-potsdam.de/en.

Fahrmeir, Ludwig, Wolfgang Brachinger, Alfred Hamerle, and Gerhard Tutz. 1996. *Multivariate Statistische Verfahren*. Walter de Gruyter.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2010. "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software* 33 (1): 1–22. https://doi.org/10.18637/jss.v033.i01.

Funk, Chris, Pete Peterson, Martin Landsfeld, Diego Pedreros, James Verdin, Shraddhanand Shukla, Gregory Husak, et al. 2015. "The Climate Hazards Infrared Precipitation with Stations-a New Environmental Record for Monitoring Extremes." *Scientific Data* 2 (1): 1–21.

Hastie, Trevor, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Vol. 2. Springer.

Huang, Boyin, Peter W Thorne, Viva F Banzon, Tim Boyer, Gennady Chepurin, Jay H Lawrimore, Matthew J Menne, et al. 2017. "NOAA Extended Reconstructed Sea Surface Temperature (ERSST), Version 5." *NOAA National Centers for Environmental Information* 30: 8179–8205.

Schlund, Manuel, Veronika Eyring, Gustau Camps-Valls, Pierre Friedlingstein, Pierre Gentine, and Markus Reichstein. 2020. "Constraining Uncertainty in Projected Gross Primary Production with Machine Learning." *Journal of Geophysical Research: Biogeosciences* 125 (11): e2019JG005619.

Schulzweida, Uwe. 2019. "CDO User Guide (Version 1.9. 6)." *Max Planck Institute for Meteorology: Hamburg, Germany*.

Smith, Thomas M, Richard W Reynolds, Thomas C Peterson, and Jay Lawrimore. 2008. "Improvements to NOAA's Historical Merged Land–Ocean Surface Temperature Analysis (1880–2006)." *Journal of Climate* 21 (10): 2283–96.

Tibshirani, Robert, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. 2005. "Sparsity and Smoothness via the Fused Lasso." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67 (1): 91–108.

Tibshirani, Robert, Guenther Walther, and Trevor Hastie. 2001. "Estimating the Number of Clusters in a Data Set via the Gap Statistic." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 63 (2): 411–23.

Tibshirani, Ryan J, and Jonathan Taylor. 2011. "The Solution Path of the Generalized Lasso." *The Annals of Statistics* 39 (3): 1335–71.

Van der Kooij, Anita J. 2007. *Prediction Accuracy and Stability of Regression with Optimal Scaling Transformations*. Leiden University.