Case 2
Effect of climate on Campylobacter infection in broilers

Frederik Boe Hüttel, s151689
Helle Rus Povlsen, s134885
Christian Glissov, s146996

January 16, 2018

DTU Compute - Institute for Mathematics and Computer science
Applied statistics and statistical software 02441 Jan 2018
Lasse Engbo Christiansen

Technical University of Denmark

Summary

This report will investigate how climatic variables can explain the seasonality in the ratio of *Campylobacter* infected broilers. Infections in broilers on danish farms have been recorded from 1998 to 2008 in parallel with multiple climatic attributes. The given data consist of climate data as well as a compilation of the observations on infection detected before slaughter in broilers from different regions. After an initial description of the data set, multiple linear models using various features of the attributes are created and reduced to only include significant, explanatory variables. Finally, all models will be evaluated by either AIC-score or fulfilment of assumptions. Independence of infection ratios within each region is tested revealing regional difference. Hence, unaccounted factors differentiating regions have an impact on the ratio of *Campylobacter* infected broilers.

Finally, the discussion will accentuate that temperature is the explanatory variable with the largest impact on the infection ratio, and that the climatic variables in general explain the variance of the response. In the final model most assumptions are satisfied, however, auto-correlation between residuals remain an unsolved problem, despite attempts.

Contents

Introduction

In Denmark, and in other developed countries, *Campylobacter* infections are becoming an increasing problem. The source of the infective bacteria is from undercooked meat or cross-contamination of mainly broilers. These farm animals show no symptoms of infection, and thus serve as infection reservoirs for the zoonosis. Carrier infection can only be detected from fecal samples, and therefore cloacal swabs on ten birds from a flock are carried out before slaughter. If any strain of *Campylobacter* is present in just a single swab, the entire flock is designated as positive of infection. These tests are necessary to retain infected meat from the market, however, a farmer can loose profit from investing in broilers that are being discarded. The aim is therefore to reduce the amount of infections in broilers. Since seasonal change highly affects the occurrence of *Campylobacter*, climatic variables relating to the seasons may explain the variation. By studying a model of proportion of infections in boiler flocks, certain explanatory variables may emphasise which measures can be taken to reduce *Campylobacter* infections.

Description of the data set

For this statistical analysis, the data set provided by Lasse Engbo Christiansen will be used.

The data set contains data on the flocks of slaughtered broilers in Denmark from 1998 to 2008 along with the climate data for the period. The broiler data is based on the number of total broilers slaughtered and the number of broilers that are infected with *Campylobacter*. These attributes are tracked from 8 different regions of Denmark. The climate data consist of average temperature, max temperature, relative humidity, sun hours and the precipitation for each given week. The climate attributes are considered as continuous attributes. The attributes sun hours could be considered a discrete attribute with many different discrete levels or considered a continuous attribute. The same argument can be made for the total and positive counts which exists as discrete integer observations, but these will also be regarded as continuous.

	Year	Week	Average Temperature	Max Temperature	Relative humidity	Sun hours	Precipitation	Total	Positive
ID	Factor	Factor	Cont.	Cont.	Cont.	Cont.	Cont.	Cont.	Cont.
1	1998	1	4.30	7.20	93	3.00	28.0	16	6
2	1998	2	4.50	9.20	92	13.0	19.0	68	27
3	1998	3	4.70	8.10	91	9.00	17.0	72	24
4	1998	4	-1.10	4.10	84	18.0	8.00	99	41

Table 1: Table showing head of the data. Individual regions have been omitted.

In total the data set contains 537 weeks and 25 attributes when adding the different region totals. For the purpose of the analysis the two factors `Year` and `Week` will be omitted as attributes, because the problem of interest is the relationship between the climate variables and the proportion of positive broiler flocks per week. The proportion will be calculated as a positive ratio based on the number of positives divided by the total count.

	Average Temperature	Max Temperature	Relative humidity	Sun hours	Precipitation	Total	Positive	Positive ratio
Min	-5.4	0.60	**0.00**	**0.00**	0.00	9	1.00	0.01
1st Qu.	3.9	8.80	70.00	13.65	5	68	14.00	0.19
Median	8.90	14.50	77.00	28.00	12.00	101.00	30.00	0.30
Mean	8.72	14.79	77.41	32.36	15.26	93.98	34.79	0.35
3rd Qu.	13.80	20.60	86.00	48.15	23.00	118.00	50.00	0.49
Max.	21.00	29.70	98.00	103.00	75.00	222.00	106.00	0.90
Nan			32	74				

Table 2: Table showing summary of the attributes including the positive ratio. Individual regions and dates have been omitted

From Table (2) a summary of the attributes can be seen. There appears to be some errors in the climate data. For example there is a record with a relative humidity at 0. This is almost a physical impossibility in Denmark, as some of the lowest recorded values of relative humidity is around 1%. Therefore, the zero values from the relative humidity will be set to NA. The same can be said for the sun hours. Some of the weeks have recorded 0 sun hours. These weeks with zero sun hours are week 20 through 23 in 2007. So for almost a whole month in the spring there was 0 sun hours. Assuming this is an error, the values will be set to NA, as well. As for the NA count there is missing relative humidity data for the the year of 2008. After the removal of the errors and the updated descriptive statistics can be seen in table (3)

	Average Temperature	Max Temperature	Relative humidity	Sun hours	Precipitation	Total	Positive	Positive ratio
Min	-5.4	0.60	48	0.4	0.00	9	1.00	0.01
1st Qu.	3.9	8.80	71.00	14.00	5	68	14.00	0.19
Median	8.90	14.50	77.00	28.00	12.00	101.00	30.00	0.30
Mean	8.72	14.79	77.72	32.64	15.26	93.98	34.79	0.35
3rd Qu.	13.80	20.60	86.00	48.5	23.00	118.00	50.00	0.49
Max.	21.00	29.70	98.00	103.00	75.00	222.00	106.00	0.90
Nan			34	78				

Table 3: Table showing summary of the attributes excluding zero values in Sun hours and relative humidity and including the positive ratio. Individual regions and dates have been omitted

Figure 2 shows the pairs plot of the continuous attributes with dates and regions omitted. As expected, there is a strong correlation between average temperature and max temperature. There also exists a negative correlation between relative humidity and sun hours, and a weaker correlation between any of the temperatures and relative humidity as well as sun hours. Precipitation is the variable that confers least correlation with any of the other variables. It also appears that at some break point there is a high correlation between the average temperature and the ratio of positive observations.
The date attributes (week and year) have been omitted from the pairs plot and will be omitted from the first part of the statistical analysis. If it appears that time is a significant it will be added to the statistical model.

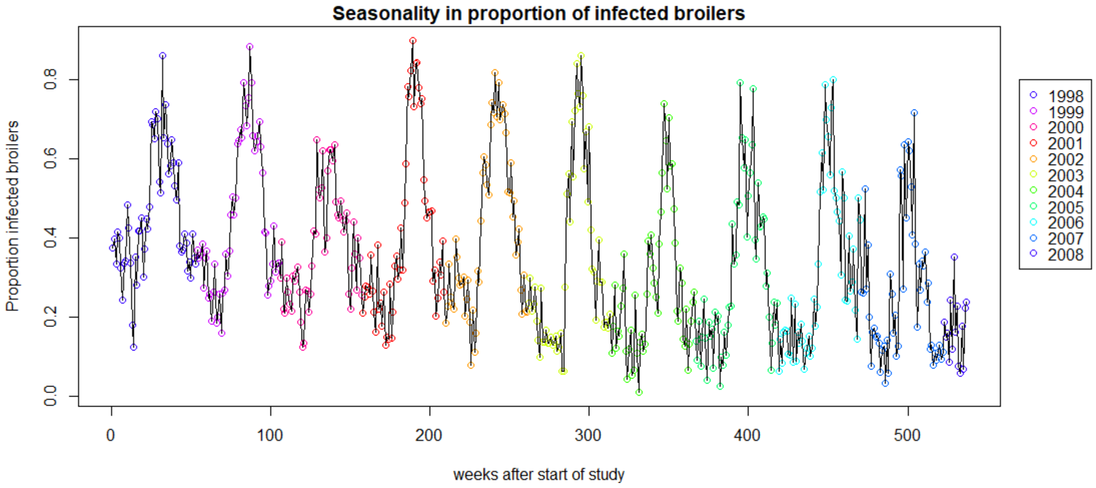


Figure 1: Showing the positive ratio based on the time

It has been stated that there is a seasonal dependency on the ratio of infections,and looking at figure (1), the positive ratio is clearly related to the different seasons. It appears there is a spike each summer with the ratio. However, this analysis is dedicated to investigate the climatic influence on the ratio of infections, so for the first part of the statistical analysis the time dependency will be discarded.

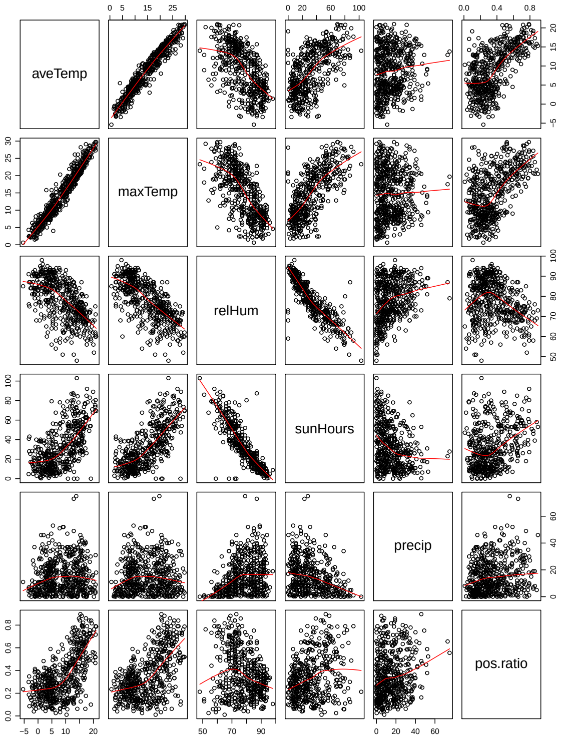


Figure 2: Pairs plot of the continuous attributes. The top 5 rows contain correlations between explanatory variables. The bottom row contains correlations between the response variable, positive ratio, and each of the explanatory variables.

To investigate what parameters are more important and which may interact a regression tree is helpful. A regression tree is build on infection ratio as a function of average temperature, maximum temperature, relative humidity, sun hours and precipitation. The resulting tree, see figure 3, accentuates that average temperature is the most important explanatory variable. From the large vertical distance to the next split points we can tell that average temperature explains most of the variance in the proportion of infections. Each limb of the tree branches out more times, which indicates highly complex interaction structure. It is interesting to notice that sun hours appear to be important in the range of 8.85 to 11.25 degrees Celsius of average temperatures. Similarly, between 11.25 and 14.45 degrees Celsius the sun hours are again important. With a high average temperature (above 14.45) we will expect high proportions of infections.

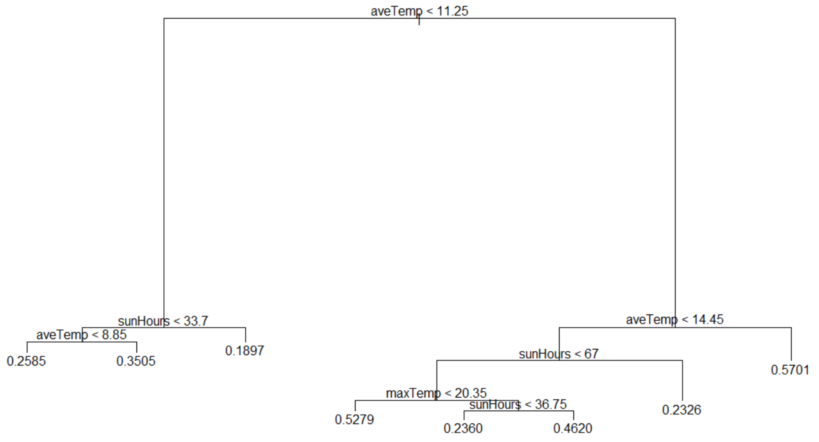


Figure 3: Regression tree of the three most imporant explanatory variables: average temperature, sun hours, and maximum temperature.

Statistical analysis

Before formulating the initial multiple regression, a general additive model (GAM) is fitted to detect linearity and/or non-linearity of the explanatory variables individually. For non-linearity, the optimal knot point(s) may be optimised by minimising the residual sum of squares, resulting in a piece-wise linear function. If the parameters from a linear regression of the piece-wise function and a linear term are significant, the variables may form part of the initial linear model.

Assumptions of linear models are evaluated based on the diagnostics plots. Another purpose of the plots is to detect outliers based on the leverage of the residuals. Any extreme observations must be validated from comparison with data points of similar conditions. If the observation is deemed to origin from an error it can be omitted from the data set. The initial model must once again be formulated and the assumptions checked once more.

Assumptions are met if the variance shows homogeneity and residuals are being independent and identically distributed with a normal distribution. Power transformation of the response variable may be applied to induce variance homogeneity.

The minimal adequate model is derived using the given function `stepP`, which deselects insignificant parameters. Since `stepP` does not remove insignificant parameters which form part of an interaction the final model refinement is done manually.

Lastly, the model assumptions must be evaluated again and the above steps repeated if assumptions are not met. Moreover, inspection of the residuals according to each of the explanatory variables reveal whether any explanatory variable needs transformation. Finally, nested models will be evaluated based on AIC.

Regional differences

To investigate regional effects a model may be built including regions as factors. Additionally, a χ^2-test reveals independence between the response variable and the regions.

Time dependency

To investigate if there is a time dependency in the model. The autocorrelation will be looked at, to determine if there is any autocorrelation within the data. This would suggest that time should be added as a indepeding variable.

Results

The aim of the analysis is to find a simple statistical model to predict the ratio of positive flocks by using several different climatic variables. The first thing to look at is the complexity of each independent variable. To do this a generalised additive model (GAM) is used.

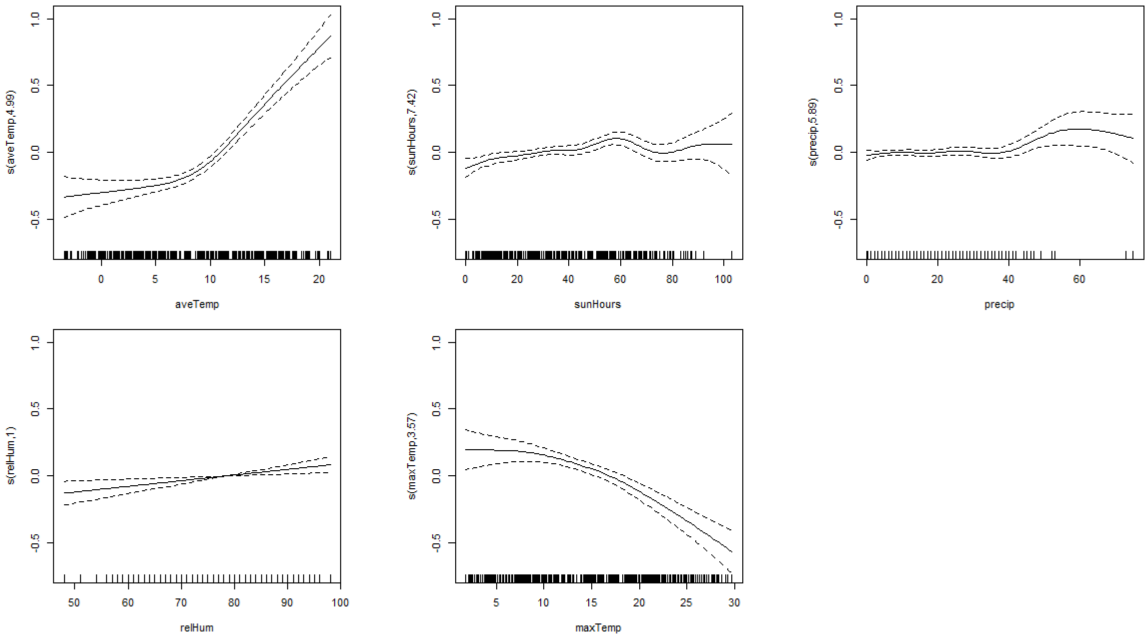


Figure 4: GAM plot of each explanatory variable.

For the plots it's clear that it's not viable to fit a model based on the assumption that each independent variable is linear. The first plot to the top left shows a clear non-linear relationship. A cutoff point is needed to do a piece-wise of the variable, it appears that the slope is changing around 8. By optimising over the residual sum of squares for a model with average temperature as the independent variable and the response as the ratio of positive tested flocks, the cutoff point is found to be approximately 8.28 for the average temperature. Another thing to look at is the maximum temperature (`maxTemp`). The Maximum temperature variable seems to follow a second order polynomial. `Precip` also appears to need a piece-wise cutoff, however there is only a few points at the end which makes the confidence interval large due to these outliers. A cutoff for `Precip` therefore doesn't seem reasonable. The same goes for some of the interval span of sun hours. One model with piece-wise sun hours and one model without piece-wise sun hours will be made and evaluated, to check if the piece-wise cutoff for sun hours is significant. It is assumed that the rest of the variables in figure 4 is linear. Another thing to notice is that `aveTemp` is highly correlated with `maxTemp`, this can also be seen by looking at the GAM plots. When `aveTemp` increases it seems like `maxTemp` decreases. This will have to be taken into evaluation when testing the different models.

The initial model will be the most complex model (see figure **??**), meaning it will contain all the relevant transformations, all of the explanatory variables and their two-way interactions. Greater interactions than two have been omitted for simplicity reasons. The first model will be the full model with piece-wise sun hours, the cutoff is 59. This model will be referred to as y_1
Looking at the residuals vs fitted plot of the initial model it is seen that the residuals, normality and scale-location is acceptable, but might be improved (see figure 19) by using a transformation of the response variable. A lot of the residuals seem to be bunched up together at certain parts of the scale-location and residuals

plot. An evenly distribution of the residuals would be desirable. Using the inbuilt box-cox function in R, the most likely power-transformation is approximately $\lambda \approx 0.6$ (figure 21). The change is not huge (figure (20)), however, it appears that the distribution of residuals is a bit more evenly distributed. Conversely, the residuals seem to translate further away from certain outliers. One might argue it makes the model worse. No significant beneficial change can be observed and therefore the transformed model is omitted.

Now reducing the fully complex model with the function `stepP` and manually removing non-significant terms that are left. When the model has been reduced the outlier seen in figure (19) observation 379 is no longer an outlier. Taking a look at the residual plots (figure 22) it can be seen that the piece-wise sun hours variable might need to be transformed, the low residuals need to be translated towards the outlier, a simple square root transformation should be enough. This requires a new model, y_2, repeating all the reducing steps for the new model and the following residual plot is given:

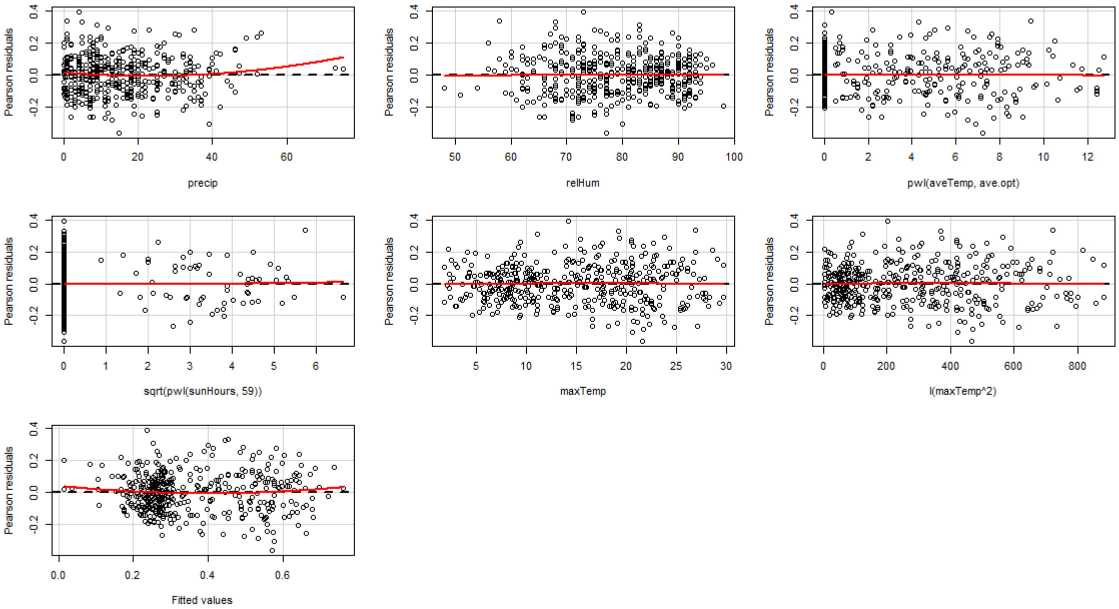


Figure 5: Residuals for each explanatory variable of model y_2.

It seems like the square root transformation worked, everything seems evenly distributed and independent. Slight upward trend for `precip`, but this is only due to two outliers, not enough to justify a transformation of the variable. The curvature test is also non-significant. This means that there are no need for a quadratic term:

Coefficient	**Test Stat**	**Pr(> |t|)**
precip	1.829	0.068
relHum	0.447	0.655
pwl(aveTemp, ave.opt)	-0.244	0.807
$\sqrt{\text{pwl(sunHours, 59)}}$	0.138	0.890
maxTemp	-0.718	0.473
I(maxTemp^2)	-0.272	0.786
Tukey test	1.443	0.149

Table 4: Curvature test

The last thing to look at is the diagnostics plot

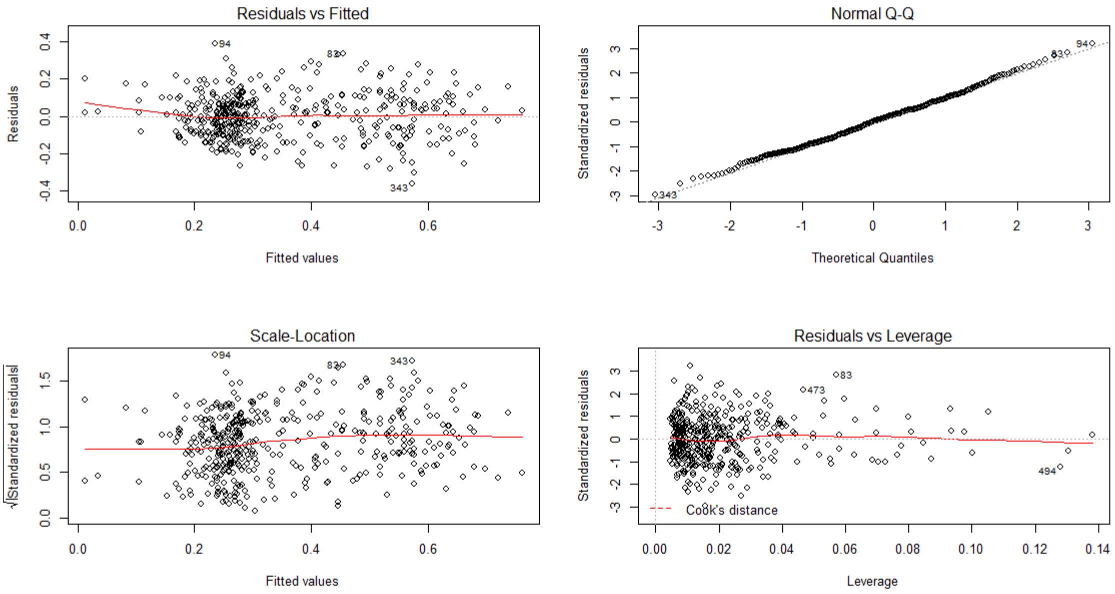


Figure 6: Diagnostic plot of the final model of y_2

Everything seems to be fine, no extreme leverages or outliers. The normality is fine and the scale-location and residual plot looks good. A summary of the model for further details can be seen in appendix, figure (23). Later a comparison using AIC will be made of all the models to see if a better score has been achieved, it is expected the transformation of the explanatory variable `sunHours` should give a slightly better AIC score, however intuitions can be prone to disappointments. The next interesting thing would be to look at a model without the piece-wise sun hours. A new model, y_3, is created and reduced following the same pattern as before. Finally another model without `maxTemp`, y_4, is fitted and reduced. The following assumptions of doing AIC is fulfilled, using the same observations and no transformation of the response variable has been made. Comparing the four models gives the following AIC scores:

Model	AIC
y_1, Piece-wise sunHours	-564.35
y_2, **Transformed piece-wise sunHours**	-567.42
y_3, No piece-wise sunHours	-555.16
y_4, No `maxTemp`	-510.33

Table 5: AIC scores of each model

The lowest value is chosen and the model with the transformed piece-wise `sunHours` seems to be best (y_2). It can therefore also be concluded that `maxTemp` should be in the model, however one should be vary of the interpretation and the prediction values used (a great example can be seen in appendix figure 24). The final model will therefore look like the following:

$$y_2 = \beta_p x_p + \beta_{rH} x_{rH} + \beta_{aTp} x_{aTp} + \beta_{sHp} x_{sHp} + \beta_{mT} x_{mT} + \beta_{mT^2}(x_{mT}^2) \tag{1}$$

$$+ \beta_{p:rH}(x_p x_{rH}) + \beta_{p:s}(x_p x_s) + \beta_{rH:aTp}(x_{aTp} x_{rH}) \tag{2}$$

Where the coefficients are given by:

Coefficient	**Estimate**	**Std. Error**	**t-value**	**Pr(> |t|)**
Precip (β_p)	-2.908e-02	8.573e-03	-3.39	0.000761
relHum (β_{rH})	1.376e-03	3.148e-04	4.37	1.57e-05
aveTempPW, Piece-wise (8.28) (β_{atp})	1.548e-01	2.062e-02	7.50	3.65e-13
$\sqrt{\text{sunHours}}$, Piece-wise (59) (β_{sHp})	-3.049e-02	6.596e-03	-4.62	5.07e-06
maxTemp (β_{mT})	3.103e-02	4.414e-03	7.03	8.49e-12
I(maxTemp^2) (β_{mT^2})	-1.669e-03	2.078e-04	-8.03	9.93e-15
Precip:relHum ($\beta_{p:rH}$)	3.128e-04	9.354e-05	3.34	0.000899
Precip:sunHours ($\beta_{p:s}$)	1.681e-04	3.929e-05	4.27	2.34e-05
relHum:aveTempPW ($\beta_{rH:aTp}$)	-9.883e-04	2.594e-04	-3.81	0.000160

Table 6: Table of all relevant information of the final model. Further details can be found in figure (23)

The model can now be used to predict the number of positive flocks based on the explanatory climate variables. First looking at a 2d example. Keeping every variable fixed to their respective mean and let `aveTemp` vary based on its range the following prediction can be seen:

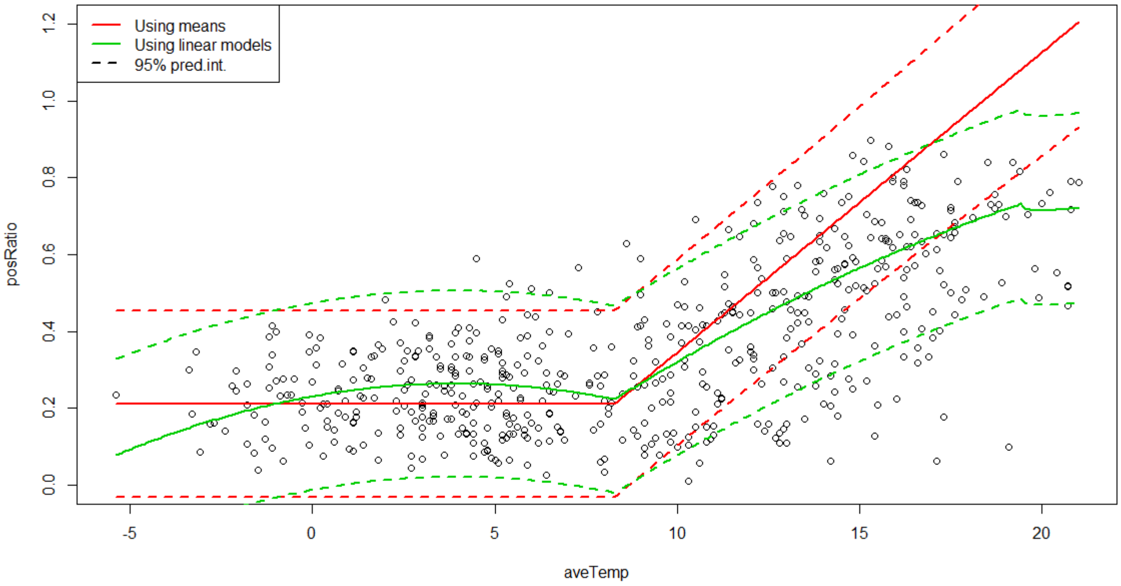


Figure 7: All explanatory variables are fixed to their respective mean values for the red fit. `aveTemp` is fitted to all the explanatory variables to optimise the fit, this is the green fit.

Clearly one should be vary of using means for predicting, this is due to interactions, correlations etc. The red fit even predict values greater than 1, which makes no sense, when the positive ratio has a range of 0 to 1. Using linear models to estimate each explanatory variable based on the non-fixed variable will give a much more optimal fit. This can be seen in the green fit and therefore using linear models is the preferred way to predict. The small sudden changes of the slope is the piece-wise function being used. One can see that the positive ratio is slightly increasing the warmer it gets. However looking at the 2d case of predictions doesn't tell much about the interactions going on, therefore it is preferred to look at the contour plots of the predictions, by letting two variables move freely and therefore also show the effect of their interactions. This would make room for a much better interpretation of the model.
The same procedure will be used, linear models instead of strict mean values will be used in order to avoid bad predictions. Looking at number of sun hours and precipitation, the following contour can be seen:

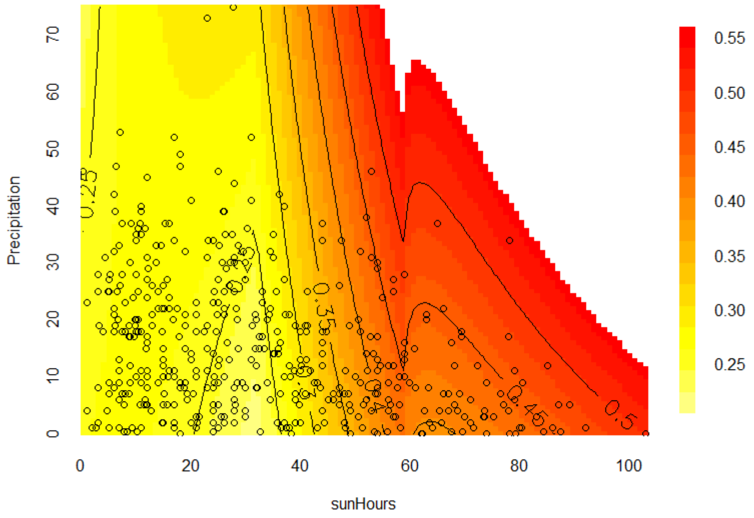


Figure 8: Contour plot of the predictions

The white area is a truncated area due to no observations was located in that area and therefore the prediction is invalid. By looking at the model and the contour plot, it can be seen that the amount of precipitation increases the proportion of positive tested flocks is almost constant or decreasing a tiny amount. However when sun hours increases a large ratio of positive results also increases. The interaction of sun hours and precipitation to have a negative impact, which means the proportion of flocks tested positive increases. Looking at the standard deviation of each prediction it can be seen that the prediction uncertainty increases the further away from the original data one predicts. This should be taken into account when doing predictions. The following standard deviation of the predictions can be seen in the following contour plot:

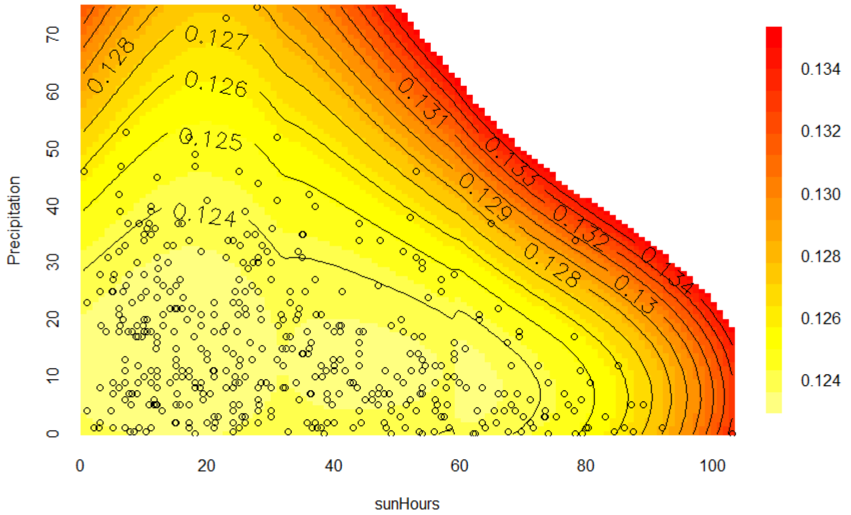


Figure 9: Contour plot of standard deviation of the predictions.

Yet again looking at another interaction. The interaction between relative humidity and average temperature. From the model coefficients, average temperature above 8.28 degrees should increase the amount of positive results and so should relative humidity. However the interaction of the piece-wise temperature and relative humidity should slightly decrease the amount of positive results.

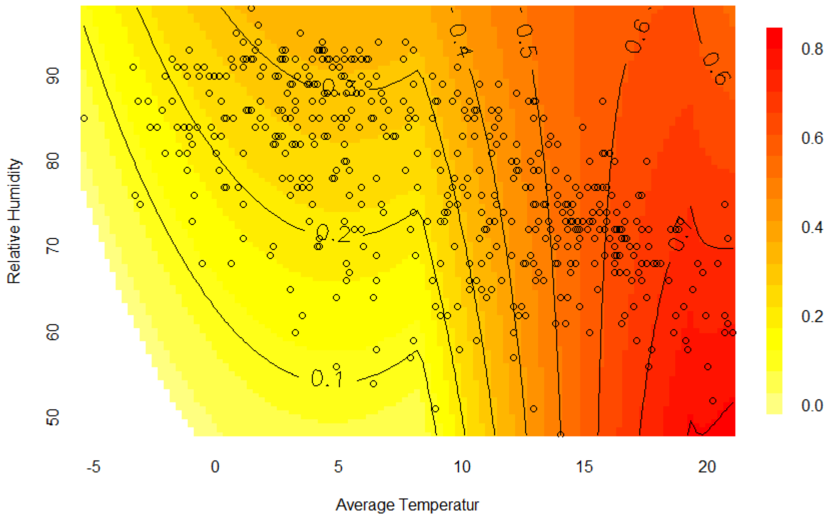


Figure 10: Contour plot of the predictions

Looking at the contour plot one can see that before the temperature hits 8.28 degrees, then the positive ratio still slightly increases. This is because even though average temperature doesn't have an effect the relative humidity interact with the precipitation, which have a positive coefficient. When average temperatures reach above 8.28 the positive rates increases by a lot, due to the very high value of the coefficient compared to the other coefficients of the piece-wise average temperature. The ratio decreases again by a bit when having high humidity levels and high temperatures. Maybe due to the *Campylobacter* not finding this environment ideal.

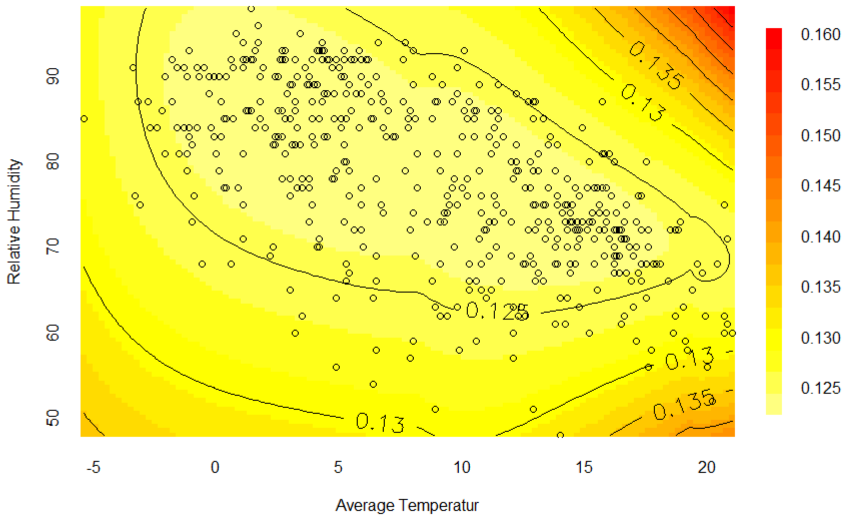


Figure 11: Contour plot of standard deviation of the predictions.

Once again the uncertainty is plotted and it can be seen the uncertainty increases the further away from the general data one is.

Finally the last interaction is shown by a contour plot. This is the interaction between precipitation and relative humidity. Precipitation has a negative coefficient and relative humidity has a positive coefficient. The ratio also seems to decrease when only having precipitation and increase slightly when only having humidity. However the interaction is positive and the tendency when increasing both the precipitation and relative humidity is also that the ratios increases and slightly falls again when precipitation reaches a certain level compared to the relative humidity, due to the negative coefficient of precipitation.

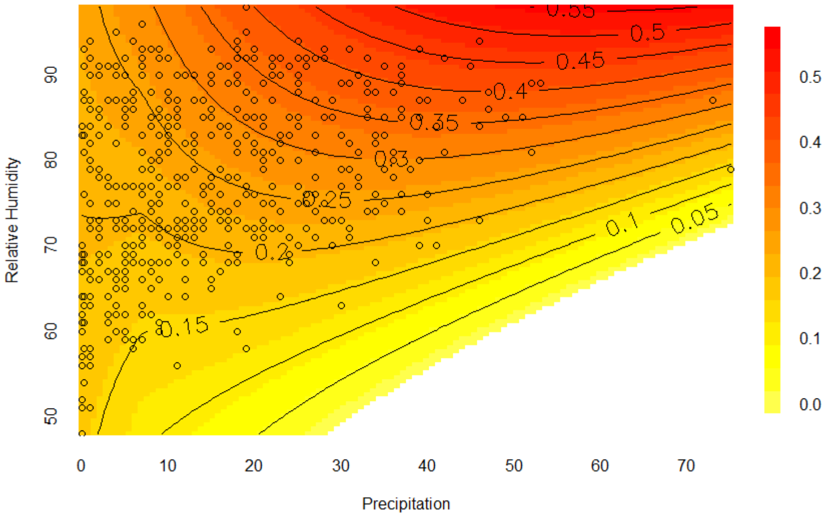


Figure 12: Contour plot of the predictions

Once again the uncertainty is plotted and we see the same thing as before.

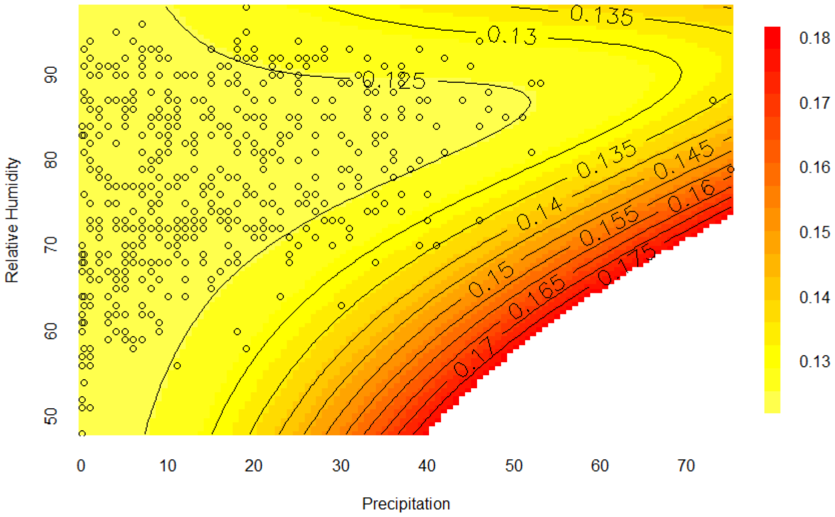


Figure 13: Contour plot of standard deviation of the predictions.

One may notice that the standard deviation is fairly large compared to the positive ratio this means the 95% prediction interval is also fairly large. However the observations are quite spread as well, so this will also increase the uncertainty of the model and therefore give a larger prediction interval.

Regional Differences

In the previous models we assumed that there was no regional dependency. It would therefore be interesting to test this assumption and test if there is a significant difference between the regions. A new model is created by adding the regional data as a factor. The regional splits can be seen in figure (15). In order to add the regional data as a factor, a new column with the regional ratios is created along with the regional factor. When the data is set up with the two new attributes (The original ratios and total is omitted) a linear regression model can be made using the new data.

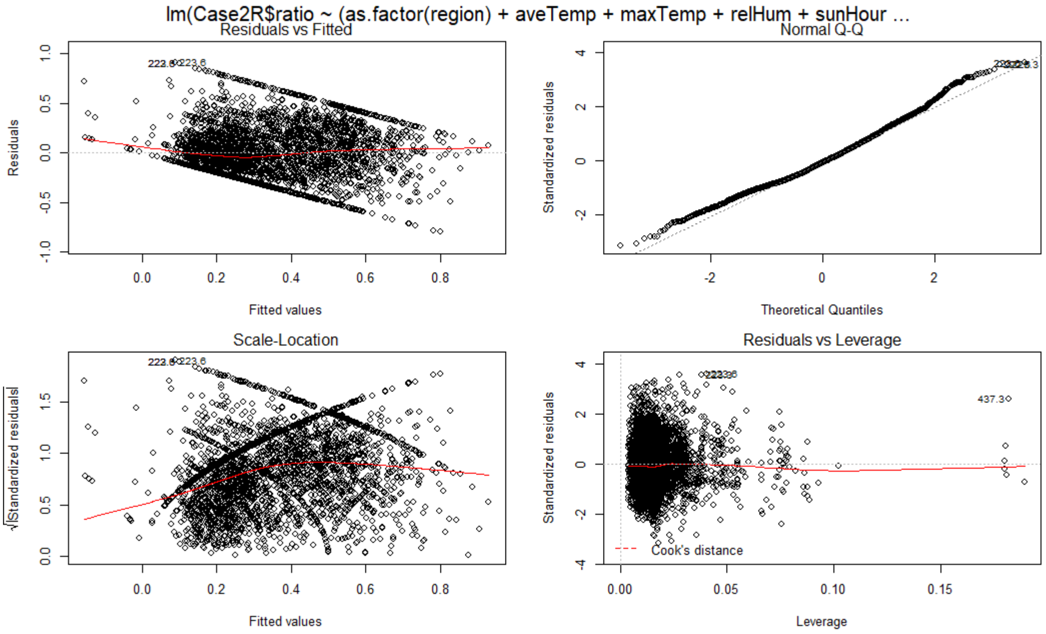


Figure 14: Dianostics plot for the new model using regions as a factorResidual vs fitted plot for the full model.

From the diagnostics plots (figure (14).) it can be seen that the residuals of the model appears to have some underlying dependency. There appears to be two lines of observations that are depended on each other, as the residuals are decreasing as the fitted values increase. Further investigation shows that these two lines are the observations where the regional ratios is either 0 or 1, meaning either all the broilers contained *Campylobacter* or none of them did. Since the models residuals appear to be dependent. Different methods are used to make a model for this kind of data, which is out of scope for the course.

So to investigate the regional differences a χ^2-test can be used to check for dependency of the number of positive broilers from each region. The table for the test is generated by the count of positive and negative counts for each region. The table can be seen in Table (7).

	R1	R2	R3	R4	R5	R6	R7	R8
Positive	1744	1894	3615	1997	1667	1389	966	1272
Negative	3671	3579	6085	1801	4563	2517	1645	2536

Table 7: Count of Positive and negative broilers for each region.

For this test the null-hypothesis is:

$$\begin{aligned} h_0 = &\text{ There is not a dependency between the region} \\ &\text{and the number of positive broilers slaughtered} \end{aligned}$$

The alternate hypothesis is then that there is a dependency between the region and the number of positive broilers slaughtered. If the test gives a p-value below 0.05 the result is significant on a 5% significance level and the null-hypothesis will be disregarded, otherwise one fails to reject the null-hypothesis.

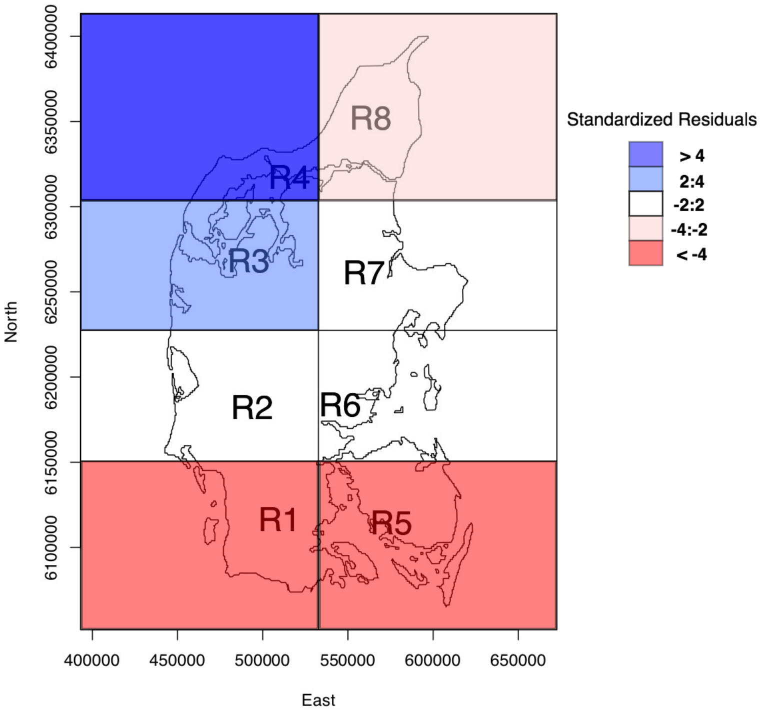


Figure 15: Map showing the different standardised Residuals based on the χ^2 test on the different regions. The residuals are the residuals for the positive attribute

Calculating the p-value with 7 means the result is very significant ($P_{\chi^2} \approx 0$), resulting in a rejection of h_0. This means the result of positive tested flocks depends of the region. Looking at the residuals based on the positive count for the test in figure (15) it can be seen that there is high residuals in region R3 and R4 meaning that these regions have an over representation of positive positive broilers, resulting in these regions performing worse than others. On the other hand region R1 and

R5 has high positive residuals meaning that these are underrepresented and doing better significantly better than the R3 and R4. So there is a regional difference, this information was not taken into account by the linear models.

Model using time

For the previous model y_2 a possible significant factor not included is time. Positive results change based on seasonality. A logical choice would therefore be to add time as a new independent variable. Making a new model y_{time}, which includes all the previous explanatory variables and the time, which doesn't interact with any of the other variables. A problem with the independence of the residuals in the old model y_2 is that the residuals seems to be decreasing based on their index (time), meaning that the residuals are not independent.

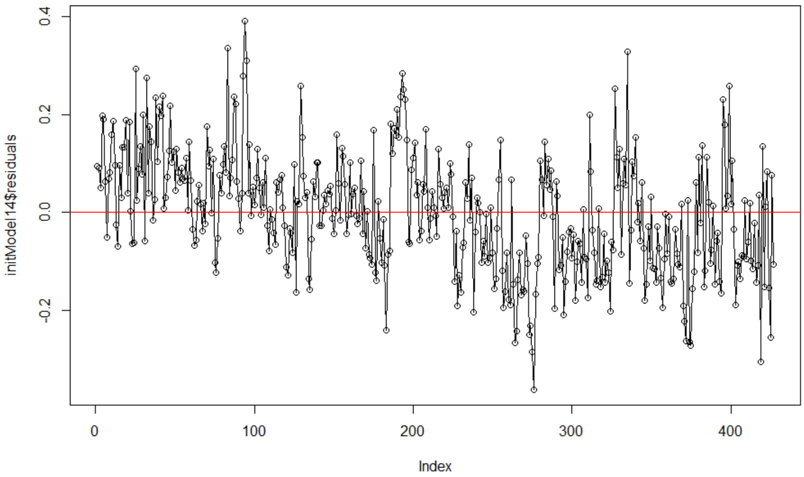


Figure 16: Residuals plotted against their respective index. A decrease can be noted. Red line is constant 0

A trend seems to be observed in figure 16. Due to the assumption of independence of the residuals for the linear regression model, this is not desirable. Looking at the ACF (figure 17) plot of the residuals, there seems to be some auto-correlation.

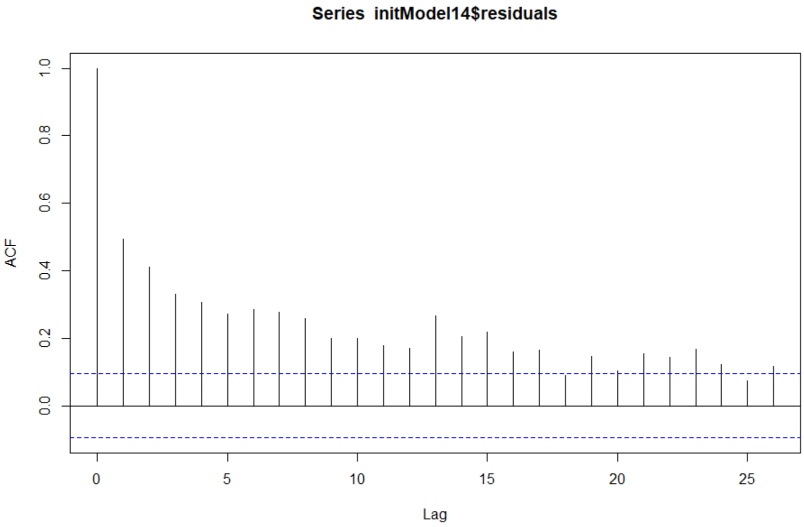


Figure 17: Auto-correlation between the residuals.

This means that some predictive information is not captured by the linear model. To be able to describe some of the auto-correlation, time is included into the full model. The model is reduced and the assumptions are checked, the final summary of the new model y_{time}, the residual plots and the assumption plots can be seen in appendix in the following figures; 25, 26 and 27.
If we now take a look at the residuals over time Figure (18) for the reduced model with time as an independent variable. It appears that the residual doesn't decrease based on the time anymore. The residuals therefore seems to be less dependent on the time.

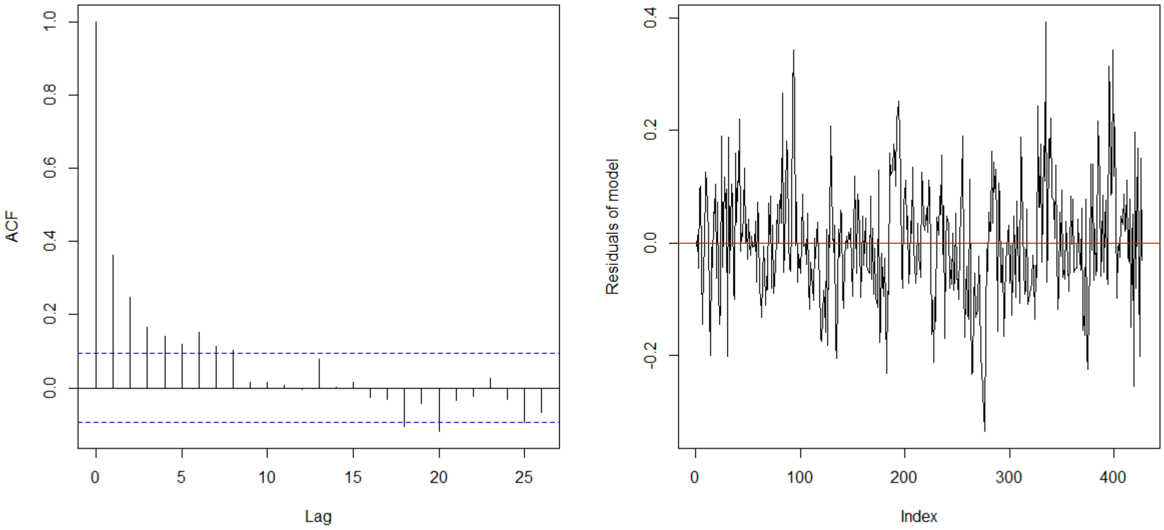


Figure 18: Left: auto-correlation of the residuals in the model. Right: Residuals plotted against time.

Looking at the auto-correlation for y_{time} in figure 18, it can be seen that it's better

than the auto-correlation for the model without time figure 17.
It appears that the model with time included as a variable, is the best for holding the assumption that the residuals of the model should be independent. So it can be assumed that one with the time as a variable is a more descriptive model capturing the overall predictive information better. For comparison reasons an AIC has been taken of the new model, $AIC(y_{time}) = -684.43$. This is the best model so far, when comparing it to the other AIC values in table 5.

Discussion

During this study we investigated what climatic variables explain the variation in proportions of *Campylobacter* infections. By evaluating AIC of several models based on the same five explanatory variables we arrived at a single model. This model states that the infection rate depends on all five variables (see table 6) of which the average temperature (when above 8.28 degrees Celsius) has the highest impact. Not surprisingly, the highly correlated maximum temperature also has a large impact on the response. The quadratic term of the maximum temperature has a negative impact as to compensate for the correlation to average temperature.

Negative impacts on the infection rate are provided by precipitation, sun hours (when above 59 hours) and the interaction between relative humidity and average temperature above 8.28 degrees Celsius. The optimal condition for infections with *Campylobacter* is seen with high temperature and low relative humidity (see figure 10). However, interaction between precipitation and sun hours show a positive impact (see figure 8), and thus increases the infection ratio. With sun hours in the interval of 40 to 60 hours the infection ratio is mostly impacted by increase in sun hours. Above 60 hours the highest ratio of infections can be achieved gradually even with decreasing precipitation. The third interaction is between precipitation and relative humidity (see figure 12) and reveals that high infection ratios are a product of high precipitation and high relative humidity. These two variables are also correlated, which means that if we have high humidity, we will probably also have high precipitation and therefore high infection ratios.

Since we find that the ratio of infected broilers depend on the above mentioned climatic variables, the seasonality of the infections are reflected by the explanatory variables. In late summer, with generally the highest temperatures of the year we would see the highest ratio of infections.

To avoid high infection ratios the broilers should be kept in a constrained environment. Means to reduce the infection ratio could be through artificial simulation of the optimal climate. Keeping broilers inside a closed air-conditioning system with low average temperatures should be an effective measure. The relative humidity should be kept at a minimum. In such a setting the amount of precipitation or sun hours ought not to influence the infection ratio.

Based on the χ^2-test of the dependency of regions, there is a significant difference between some of the regions. Since it is assumed that the climate data is the same for all the regions, there must be some additional regional differences that has an impact. Regional climate data could have helped make an estimate of what parameters have a influence on the ratio of infected broilers. Moreover, other variables, such as the relative area per chicken, the fodder and hygiene of the broilers could have been included in the analysis. Factors such as whether the broiler is free range or in farm houses might also be interesting to take into consideration. These factors could also be assumed to have an impact on the ratio of broilers infected with *Campylobacter*.

For further analysis it would be interesting to make a better model taking into the account of time. The model containing the time seemed to be the best model, due to the residuals being auto-correlated.

Conclusion

Seasonality in the ratio of *Campylocater* infections have in this analysis been explained by climatic variables. From initial GAM analysis two variables are estimated to be non-linear and are incorporated into one of several initial models as piece-wise linear function. Correlations were observed between several of the climatic variables, however, every attribute was included in the initial models. Based on AIC the reduced models are compared and the lowest scoring model contained both piece-wise functions. The piece-wise function of average temperature is the climatic variable with the largest impact on infection ratios. Thus, with increasing temperature we expect an increase in the ratio of infections. Eight other parameters were also significant in the model. This result corresponds to the high ratio observed in the late summer season. Recommendations based on this model consist of constricting the average temperature as well as humidity in the farm houses. A problem with the presented model is auto-correlation between residuals. By including time as a continuous variable the slight correlation to infection ratio is accounted for. The auto-correlation is reduced, however, not eliminated. For further studies one might look more in depth with the impact of time on the data set and conduct a time series analysis of the data.

Appendix

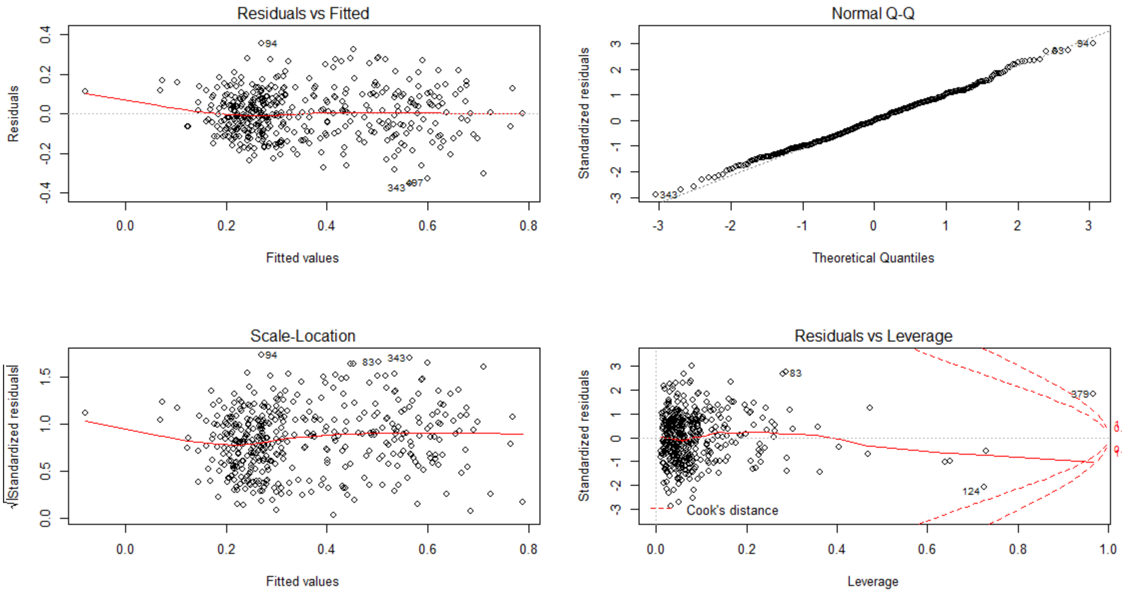


Figure 19: Residual vs fitted plot of the full initial model. y_1

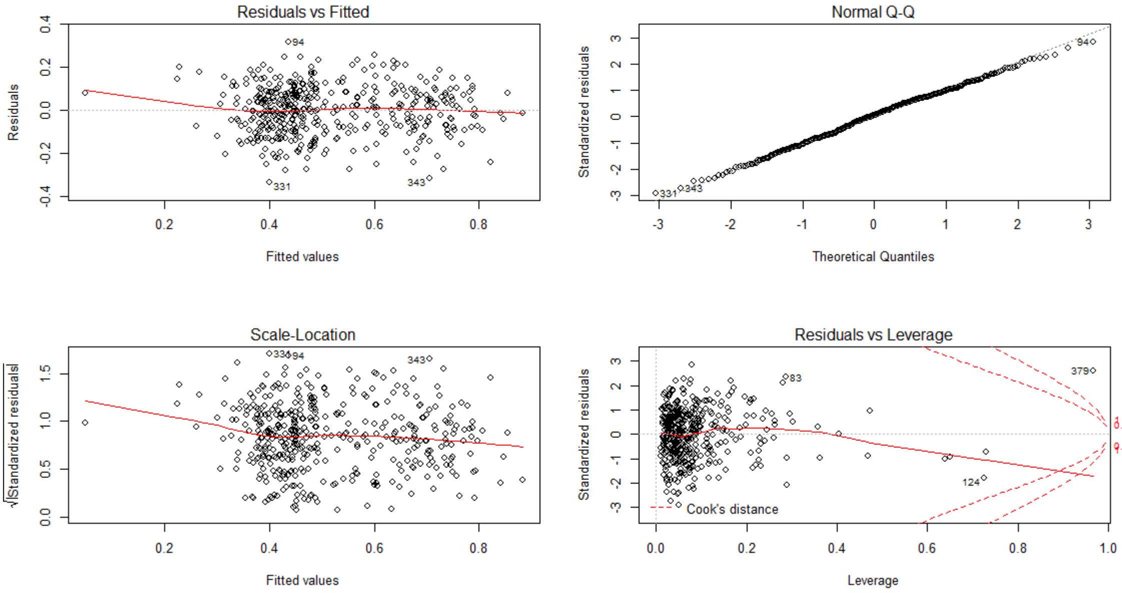


Figure 20: Residual vs fitted plot of the initial transformed model.

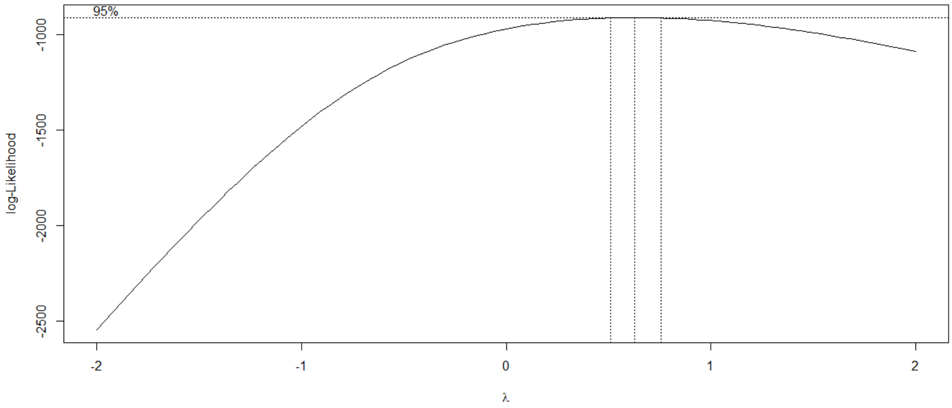


Figure 21: Plot of the most likely power transformation.

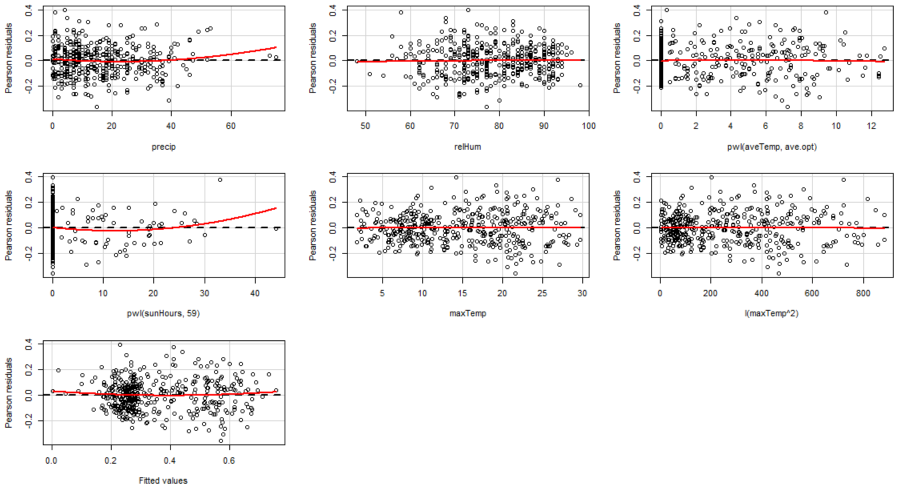
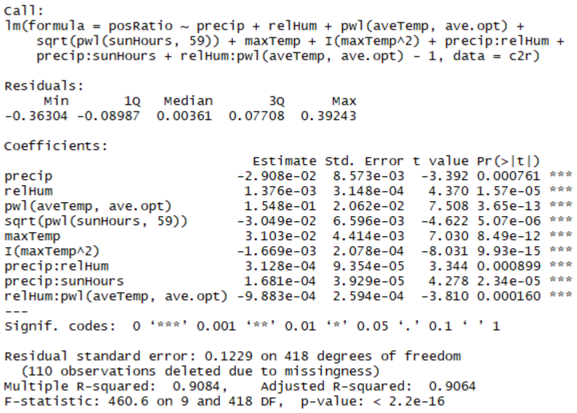


Figure 22: Plot of the residuals for the first reduced model y_1



```
Call:
lm(formula = posRatio ~ precip + relHum + pwl(aveTemp, ave.opt) +
    sqrt(pwl(sunHours, 59)) + maxTemp + I(maxTemp^2) + precip:relHum +
    precip:sunHours + relHum:pwl(aveTemp, ave.opt) - 1, data = c2r)

Residuals:
     Min       1Q   Median       3Q      Max
-0.36304 -0.08987  0.00361  0.07708  0.39243

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
precip                      -2.908e-02  8.573e-03  -3.392 0.000761 ***
relHum                       1.376e-03  3.148e-04   4.370 1.57e-05 ***
pwl(aveTemp, ave.opt)        1.548e-01  2.062e-02   7.508 3.65e-13 ***
sqrt(pwl(sunHours, 59))     -3.049e-02  6.596e-03  -4.622 5.07e-06 ***
maxTemp                      3.103e-02  4.414e-03   7.030 8.49e-12 ***
I(maxTemp^2)                -1.669e-03  2.078e-04  -8.031 9.93e-15 ***
precip:relHum                3.128e-04  9.354e-05   3.344 0.000899 ***
precip:sunHours              1.681e-04  3.929e-05   4.278 2.34e-05 ***
relHum:pwl(aveTemp, ave.opt) -9.883e-04  2.594e-04  -3.810 0.000160 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1229 on 418 degrees of freedom
  (110 observations deleted due to missingness)
Multiple R-squared:  0.9084,    Adjusted R-squared:  0.9064
F-statistic: 460.6 on 9 and 418 DF,  p-value: < 2.2e-16
```

Figure 23: The summary of the final model, The degrees of freedom should be 416, due to the two piece-wise estimations. However the result of the p-values won't change enough to make a difference.

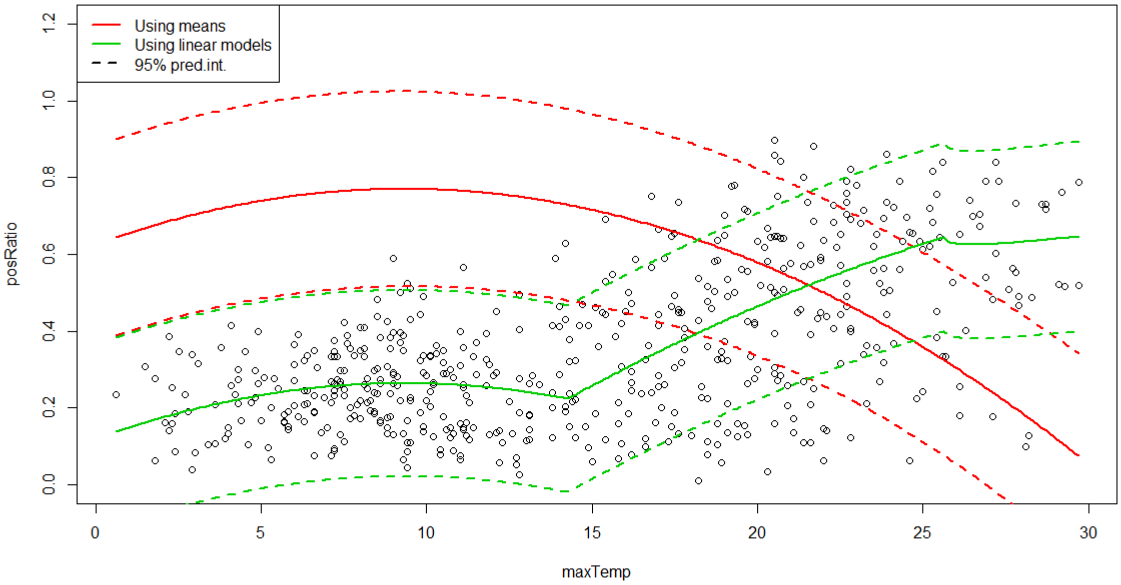
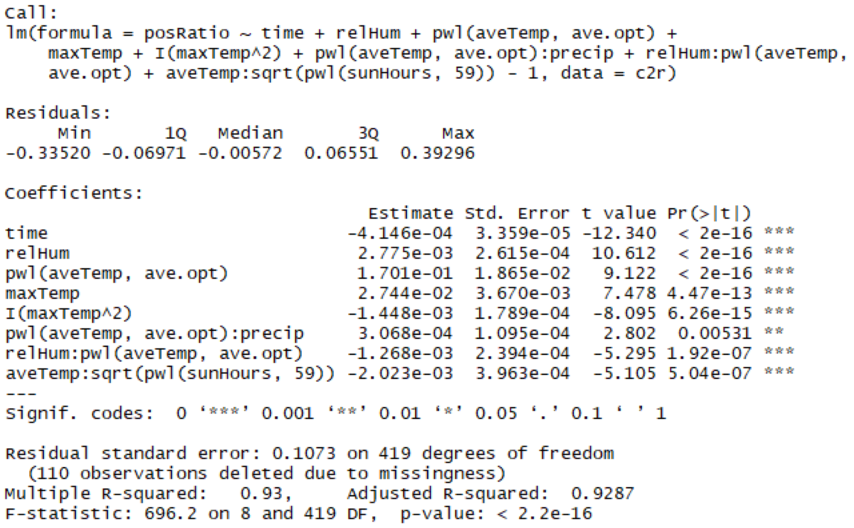


Figure 24: Prediction gone wrong. The correlation between `aveTemp` and `maxTemp` messes up the prediction, when using means as fixed values.



```
Call:
lm(formula = posRatio ~ time + relHum + pwl(aveTemp, ave.opt) +
    maxTemp + I(maxTemp^2) + pwl(aveTemp, ave.opt):precip + relHum:pwl(aveTemp,
    ave.opt) + aveTemp:sqrt(pwl(sunHours, 59)) - 1, data = c2r)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.33520 -0.06971 -0.00572  0.06551  0.39296 

Coefficients:
                                  Estimate Std. Error t value Pr(>|t|)    
time                            -4.146e-04  3.359e-05 -12.340  < 2e-16 ***
relHum                           2.775e-03  2.615e-04  10.612  < 2e-16 ***
pwl(aveTemp, ave.opt)            1.701e-01  1.865e-02   9.122  < 2e-16 ***
maxTemp                          2.744e-02  3.670e-03   7.478 4.47e-13 ***
I(maxTemp^2)                    -1.448e-03  1.789e-04  -8.095 6.26e-15 ***
pwl(aveTemp, ave.opt):precip     3.068e-04  1.095e-04   2.802  0.00531 ** 
relHum:pwl(aveTemp, ave.opt)    -1.268e-03  2.394e-04  -5.295 1.92e-07 ***
aveTemp:sqrt(pwl(sunHours, 59)) -2.023e-03  3.963e-04  -5.105 5.04e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1073 on 419 degrees of freedom
  (110 observations deleted due to missingness)
Multiple R-squared:   0.93,	Adjusted R-squared:  0.9287 
F-statistic: 696.2 on 8 and 419 DF,  p-value: < 2.2e-16
```

Figure 25: Summary of time model

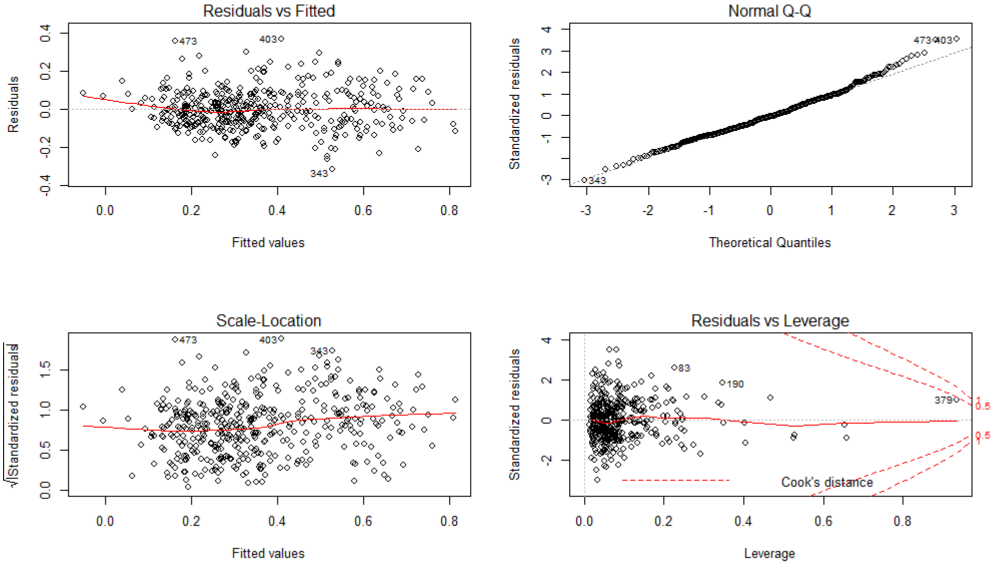


Figure 26: Assumption plots of time model

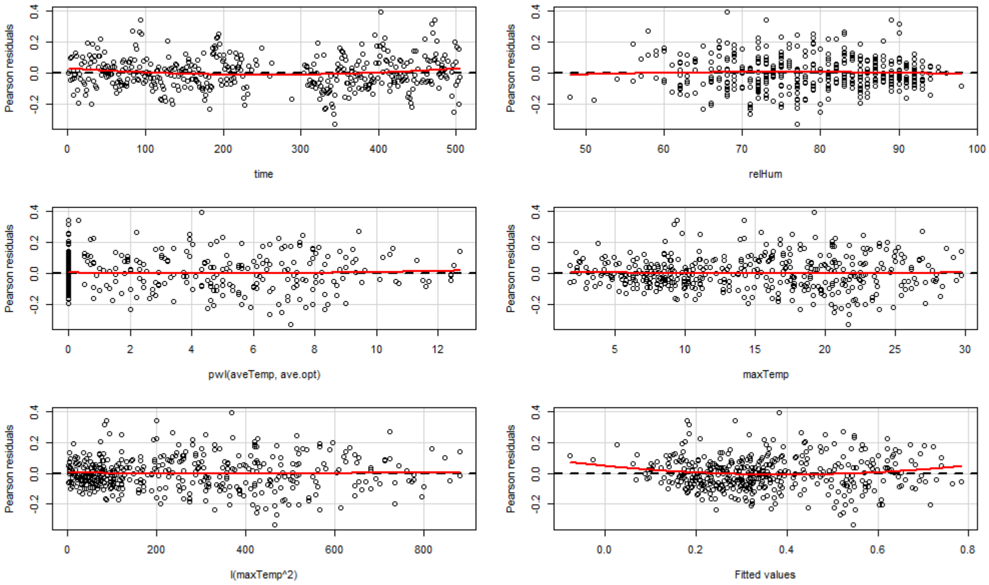


Figure 27: Residual plot of time model

```
######################
## R file for cleaning of the raw data files

climate <- read.delim(
    "~/Applied␣statistics/case2/climate.txt")
pre2002 <- read.delim(
    "~/Applied␣statistics/case2/campy_pre2002.txt")
cam0205 <- read.csv(
    "~/Applied␣statistics/case2/campy_2002-2005.csv")
cam05 <- read.csv(
    "~/Applied␣statistics/case2/campy_2005-.csv")

# Step 1
pre2002 <- pre2002[pre2002$SEKTION != "res",]

# Step 2
pre2002 <- pre2002[pre2002$AKTVNR==5133,]

# Step 3
pre2002 <- pre2002[pre2002$CHR_NR >= 1000,]
cam0205 <- cam0205[cam0205$Chrnr >= 1000,]
cam05 <- cam05[cam05$Chrnr >= 1000,]

# Step 4
pre2002$PRV_DATO <- gsub("OCT", "OKT", pre2002$PRV_DATO)
pre2002$PRV_DATO <- gsub("MAY", "MAJ", pre2002$PRV_DATO)

pre2002$new_Date<-as.Date(pre2002$PRV_DATO, format="%d%b%Y")
cam0205$new_Date<-as.Date(cam0205$Prvdato, format="%m/%d/%y")
cam05$new_Date<-as.Date(cam05$Provedato, format="%m/%d/%y")

# Step 5
colnames(pre2002)<-c("jnr","dyrnr","matr", "resultat",
    "dyreart", "olddato", "chrnr","aktvnr", "sektion",
    "epinr", "sysbem", "analyse","region", "prvdato")

colnames(cam0205)<-c("matr", "jnr", "dyrnr", "chrnr", "epinr",
    "olddato","resultat","region", "prvdato")

colnames(cam05)<-c("matr", "chrnr", "epinr", "olddato",
    "modtdato", "jnr","provenr","resultat","region", "prvdato")

pre2002<-pre2002[c("chrnr", "epinr",
    "jnr","matr","resultat", "prvdato","region")]
cam0205<-cam0205[c("chrnr", "epinr", "jnr",
    "matr","resultat", "prvdato","region")]
cam05<-cam05[c("chrnr", "epinr", "jnr",
    "matr","resultat", "prvdato","region")]

# Step 6
```

```
campy <- rbind(pre2002[,1:7], cam0205[,1:7], cam05[,1:7])

# Step 7
campy <- campy[!is.na(campy$epinr),]

# Step 8
campy$resultat[campy$resultat=="POSITIV"]="POS"
campy$resultat[campy$resultat=="NEGATIV"]="NEG"
campy$resultat[campy$resultat==""]="NEG"
campy$resultat[campy$resultat!="NEG" &
    campy$resultat!="POS"]="POS"

# Step 9
campy <- campy[campy$matr %in% c("Kloaksvaber","Svaberproeve",
        "766","772"),]
summary(as.factor(campy$matr))


# Step 13
campy<-campy[!(duplicated(campy$jnr)
    | duplicated(campy$jnr, fromLast = T)) ,]

# Step 10
campy <- campy[!duplicated(campy[,-3]),]

# Step 11
campy$week<-strftime(campy$prvdato, format = "%V")

# Step 12
campy$year <- substr(campy$prvdato,1,4)
campy <- campy[campy$year != "1997",]

# Step 14
# Step 15
campy <- campy[campy$chrnr %in%
    names(table(campy$chrnr))[table(campy$chrnr) >= 10],]

# Step 16
#campy <- campy[na.omit(campy$region),]
count_flok <- function(df){
  tempt <- as.data.frame(with(df, table(week,year)))
  tempt$week <- as.numeric(tempt$week)
  (tempt <- tempt[ !(tempt$year == 2008 & tempt$week > 15),])
  return(tempt)
}
temp.t <- count_flok(campy)
# Clean temp.t
temp.t[temp.t$week==53,]
(temp.t <- temp.t[(temp.t$Freq > 0),])
temp.t <- temp.t[!(temp.t$year == 1999 & temp.t$week == 53),]
temp.t$Freq[temp.t$week == 53 & temp.t$year == 1998]
    <- temp.t$Freq[temp.t$week == 53 & temp.t$year == 1998]+2

temp.p <- count_flok(subset(campy, campy$resultat == "POS"))

# Left join - could have used cbind
dftempt <- merge(x = temp.t, y = temp.p,
    by = c("week","year"), all.x = TRUE)

# left join with the climate dataset
final <- merge(x = climate, y = dftempt,
    by = c("week","year"), all.x = TRUE)

# Order based on year and week
final <- final[order(final$year,final$week),]
rownames(final) <- 1:nrow(final)
colnames(final)[c(8,9)] <- c("total_slaughter",
    "infected_slaughter")
final$pos.ratio <-
    final$infected_slaughter/final$total_slaughter
head(final)
```

```
## 'stepP' is like 'step' except that it uses a p-value as
##criterium
## Arguments:
##   object:  An lm object
##   level:   The model is reduced when p-values are above
##   (Default 0.05)
##   verbose: Add a print of each model as they are reduced
##
## Output:
##   object:  The reduced model
##   history: The history of the model reduction
rm(list=ls())
stepP <- function(object, level=0.05, verbose=FALSE){
  if (!("lm" %in% class(object))){
    error("First␣argument␣should␣be␣an␣lm␣object")
  }
  d1 <- drop1(object, test="F")[-1,]
  maxP <- max(d1[["Pr(>F)"]])
  lmTmp <- object
  maxVar <- row.names(d1)[d1[["Pr(>F)"]]==maxP]
  history <-NULL # For storing the history of models
  tmpFormula <- paste(as.character(formula(lmTmp))[c(2,1,3)],
  collapse="␣")

  while(maxP > level & nrow(d1)>=1){
    maxVar <- row.names(d1)[d1[["Pr(>F)"]]==maxP]
    history <- rbind(history,data.frame(formula= tmpFormula,
    maxP=maxP, maxVar = maxVar) )
    lmTmp <- update(lmTmp, paste(".~.-",maxVar))
    d1 <- drop1(lmTmp, test="F")[-1,]
    maxP <- max(d1[["Pr(>F)"]])
    tmpFormula <- paste(as.character(formula(lmTmp))[c(2,1,3)],
    collapse="␣")
    if (verbose)
      print(tmpFormula) # Print the formula after
      #each reduction
  }
  # Also adding the final model to document the p-value
  maxVar <- row.names(d1)[d1[["Pr(>F)"]]==maxP]
  tmpFormula <- paste(as.character(formula(lmTmp))[c(2,1,3)],
  collapse="␣")
  history <- rbind(history,data.frame(formula= tmpFormula,
  maxP=maxP, maxVar = maxVar) )
  return(list(object=lmTmp, history=history))
}

library(readr)
library(car)
library(ggplot2)
library(reshape2)
library(MASS)
library(mgcv)
library(tree)
library(lattice)
library(fields)

setwd("C:/Users/Christian_2/Desktop/Campy")
c2r <- read.table("case2regions4.txt", header=TRUE,
                  "\t")
c2r$week <- as.factor(c2r$week)
c2r$year<- as.factor(c2r$year)

###############DESCRIPTIVE STATISTICS################
#Pairs looking at correlation
scatterplotMatrix(c2r[2:7], diagonal="boxplot")

boxplot(c2r$maxTemp)
#remove humidity 0 ?
boxplot(c2r$relHum)

#set 0 hours of sun to NA, seems very unlikely
c2r[which(c2r$sunHours == 0),6]<-NA

#fixing data
c2r$posRatio<-c2r$pos/c2r$tot
c2r[which(c2r$relHum==0), ]$relHum <- NA

#looking for further outliers
plot(c2r$relHum, c2r$precip)

##Checking complexity
par(mfrow=c(2,3))
model<-gam(posRatio ~ s(aveTemp)+s(sunHours)+
             s(precip)+s(relHum)+ s(maxTemp), data=c2r)
plot(model)

#checking when to piecewise
par(mfrow=c(1,1))
tree.model<-tree(posRatio ~ aveTemp + relHum +
                   sunHours + precip + maxTemp, data=c2r)
plot(tree.model)
text(tree.model)

##########################MODEL###########################
pwl<-function(x,x0){
  ## x is data
  ## x0 is cut off
  ## The associated estimated parameter is for x > x0
  return( (x > x0) * (x-x0) )
}
```

```
optim <- optimize(function(zz){
  sum(residuals(lm(posRatio ~ aveTemp +
                     pwl(aveTemp, zz), data=c2r))^2 )
},c(5,15))

(ave.opt<-optim$minimum)
#piecewise at 8.28:

#splines, binary PW, significant effect nar over 8.22
summary(lm(posRatio~aveTemp + pwl(aveTemp, 8.28), data=c2r ))

#not taking sunhours and precip piecewise, not enough data at
#ends.
summary(lm(posRatio ~ I(maxTemp^2), data=c2r))

initModel <- lm(posRatio ~ (precip+ relHum+
                             aveTemp+pwl(aveTemp,8.28)
                             +sunHours+maxTemp+
                             I(maxTemp^2))^2, data=c2r)
summary(initModel)
par(mfrow=c(2,2))
plot(initModel)

#Might need to transform:
par(mfrow=c(1,1))
bc<-boxcox(initModel)
lam.opt<- bc$x[which(bc$y==max(bc$y))]
lam.opt

initModelT <- lm(posRatio^(0.6) ~ (precip+ relHum+
                                     aveTemp+pwl(aveTemp,8.22)
                                     +sunHours+maxTemp+
                                     I(maxTemp^2))^2, data=c2r)
summary(initModelT)
par(mfrow=c(2,2))
plot(initModelT)
#transformation didn’t seem to be better

#before using old model we look at
hv <- as.data.frame(hatvalues(initModel))
mn <-mean(hatvalues(initModel))
hv$warn <- ifelse(hv[, ’hatvalues(initModel)’]>3*mn, ’Warn’,
                  ifelse(hv[, ’hatvalues(initModel)’]>2*mn,
                  ’Warn’, ’-’ ))
subset(hv, warn=="Warn")
subset(hv, warn%in%c("Warn", "Warn"))
hv[order(hv[’hatvalues(initModel)’]), ]
par(mfrow=c(1,1))
plot(hatvalues(initModel), type = "h")
```

```
#leverage doesn't seem too bad, so we let it be.

#Non transformed model without outlier
initModel3 <- lm(posRatio ~ (precip+ relHum+ aveTemp+
                               pwl(aveTemp,ave.opt)+
                               sunHours+maxTemp+
                               I(maxTemp^2))^2, data=c2r)
par(mfrow=c(2,2))
plot(initModel3)

initModel4<-stepP(initModel3)$object
summary(initModel4)

initModel5<-update(initModel4, .~.-sunHours)
summary(initModel5)
par(mfrow)
plot(initModel5)
residualPlots(initModel5)

initModel6<-update(initModel5, .~.-relHum)
par(mfrow=c(2,2))
summary(initModel6)
plot(initModel6)
residualPlots(initModel6)
#looks good! However we might consider taking a piecewise of
#sunhours into account, let's try it.

#optimizing residuals
optim <- optimize(function(zz){
  sum(residuals(lm(posRatio ~ sunHours +
                     pwl(sunHours, zz), data=c2r))^2 )
},c(20,90))

(x0.opt2<-optim$minimum)

#59 is the cutoff
summary(lm(posRatio~sunHours + pwl(sunHours, 59), data=c2r ))
#It's significant.

#new model including piecewise sunhours
initModel7<-lm(posRatio ~ (precip+ relHum+ aveTemp+
                             pwl(aveTemp, ave.opt)+sunHours+
                             pwl(sunHours, 59)+maxTemp+
                             I(maxTemp^2))^2, data=c2r)
par(mfrow=c(2,2))
plot(initModel7)

#379 is an outlier the data itself looks fine though.
#Reducing model first.
initModel8<-stepP(initModel7)$object
```

```
summary(initModel8)
#remove sunHours
initModel9<-update(initModel8, .~.-1)
summary(initModel9)
#remove intercept
initModel10<-update(initModel9, .~.-sunHours)
summary(initModel10)
#everything is significant
residualPlots(initModel10)
#We can maybe improve the piecewise sunhours be
#transforming with a sqrt
plot(initModel10)

initModel11<-lm(posRatio ~ (precip+ relHum+ aveTemp+
                               pwl(aveTemp, ave.opt)+sunHours+
                               sqrt(pwl(sunHours, 59))+maxTemp+
                               I(maxTemp^2))^2, data=c2r)
par(mfrow=c(2,2))
plot(initModel11)
#379 is an outlier the data itself looks fine though.
#Reducing model first.
initModel12<-stepP(initModel11)$object
summary(initModel12)
#remove sunHours
initModel13<-update(initModel12, .~.-sunHours)
summary(initModel13)
#remove intercept
initModel14<-update(initModel13, .~.-1)
summary(initModel14)
#everything is significant
residualPlots(initModel14)
par(mfrow=c(2,2))
plot(initModel14) #<3 <3

#Due to high correlation between average temp and maxtemp,
#we try to make a model without maxTemp to compare
modNoMT<-update(initModel14, .~.-I(maxTemp^2)-maxTemp)
summary(modNoMT) #reducing
modNoMT2<-update(modNoMT, .~.-relHum:pwl(aveTemp, ave.opt))
summary(modNoMT2)

#The dependent hasn’t been transformed and same observations,
#we check for AIC
AIC(modNoMT2)
AIC(initModel14)
AIC(initModel10)
AIC(initModel6)
#model 14 is best compared to the other models

hv <- as.data.frame(hatvalues(initModel14))
```

```
mn <-mean(hatvalues(initModel14))
hv$warn <- ifelse(hv[, ’hatvalues(initModel14)’]>3*mn, ’Warn’,
                  ifelse(hv[, ’hatvalues(initModel14)’]>2*mn,
                  ’Warn’, ’-’ ))
subset(hv, warn=="Warn")
subset(hv, warn%in%c("Warn", "Warn"))
hv[order(hv[’hatvalues(initModel14)’]), ]
par(mfrow=c(1,1))
plot(hatvalues(initModel14), type = "h")
c2r[345,]

################2D Predictions#############
## A function to predict the remaining predictors
## from one predictor:
## data: Data.frame with data
## reference: Which variable is to be the reference, e.g.
## "wind"
## others: Vector with names of the other variables to be
##         predicted the default is all remaining
##         columns of data.
## ref.values: vector of values where predictions should
##         be made the default is 30 equidistant
##         space values covering the
##         range of the reference.
#############################################
lec.fun<-function(data, reference,
                others=names(data)
                [names(data)!=reference], ref.values=
                seq(min(data[[reference]]),
max(data[[reference]]),length=30)){
  pdata<-data.frame(reference=ref.values)
  names(pdata)<-reference
  for(i in others){
    lmtmp<-lm(as.formula(paste(i,"~",reference)),data)
    pdata[[i]]<-predict(lmtmp,newdata=pdata[reference])
  }
  return(pdata)
}

pred.data<-data.frame(precip=mean(c2r$precip, na.rm=T),
sunHours = mean(c2r$sunHours, na.rm=T),
maxTemp=seq(0.6, 29.7, length=300),
                      relHum=mean(c2r$relHum, na.rm=T),
                      aveTemp=mean(c2r$maxTemp, na.rm=T))

pred.int <-predict(initModel14, int="p", newdata = pred.data)
par(mfrow=c(1,1))
plot(posRatio~maxTemp,data=c2r, ylim=c(0,1.2))
matlines(pred.data$maxTemp, pred.int, lty=c(1,2,2),col="red"
        ,lwd=2)
```

```
plec <- lec.fun(c2r,reference="maxTemp",
                others=c("precip","sunHours",
                "relHum", "aveTemp"),
                ref.values=seq(0.6, 29.7, length=300))
pred.plec<-predict(initModel14, int="p",newdata=plec)
matlines(pred.data$maxTemp,pred.plec,lty=c(1,2,2),col=3,
    lwd=2)
legend("topleft",legend=c("Using␣means",
"Using␣linear␣models",
"95%pred.int."),lty=c(1,1,2),col=c(2:3,1),lwd=2)

pred.data11<-data.frame(precip=mean(c2r$precip, na.rm=T),
sunHours = mean(c2r$sunHours, na.rm=T),
maxTemp=mean(c2r$maxTemp, na.rm=T),
             relHum=mean(c2r$relHum, na.rm=T),
             aveTemp=seq(-5.4, 21, length=300))
pred.int1 <-predict(initModel14, int="p",
                    newdata = pred.data11)
par(mfrow=c(1,1))
plot(posRatio~aveTemp,data=c2r, ylim=c(0,1.2))
matlines(pred.data11$aveTemp, pred.int1,
            lty=c(1,2,2),col="red",lwd=2)

plec2 <- lec.fun(c2r,reference="aveTemp",
                others=c("precip",
                "sunHours", "relHum", "maxTemp"),
                ref.values=seq(-5.4, 21, length=300))
pred.plec2<-predict(initModel14, int="p",newdata=plec2)
matlines(pred.data11$aveTemp,pred.plec2,
            lty=c(1,2,2),col=3,lwd=2)
legend("topleft",legend=c("Using␣means","Using␣linear␣models",
        "95%␣pred.int."),lty=c(1,1,2),col=c(2:3,1),lwd=2)


###########Contour Prediction###########
summary(initModel14)
range(c2r$precip, rm.na=T)
range(c2r$sunHours[!is.na(c2r$sunHours)])
range(c2r$relHum[!is.na(c2r$relHum)])
range(c2r$aveTemp[!is.na(c2r$aveTemp)])

###USING PREDFRAME###
pred.frame <- function(reference, data,
others=names(data)[ !(names(data)%in%names(reference)) ]){
  if (class(reference) == "list"){
  ## Need to run expand.grid( <list>)
    pdata <- expand.grid(reference)
  } else {
    pdata <- reference
  }
  ref.model <- names(reference)
  if(length(names(reference))>1)
    ref.model <- paste(ref.model, sep="+")
  for(i in others){
    lmtmp<-lm(as.formula(paste(i,"~",ref.model)),data)
    pdata[[i]]<-predict(lmtmp,newdata=pdata)
  }
  return(pdata)
}

sH<-seq(0.4,103, length=100)
pC<-seq(0.01, 75, length=91)

p.sunHours <- seq(0.4,103,length=110)
p.precip <- seq(0.01, 75, length=100)

pred.data <- pred.frame(reference =
list(sunHours=p.sunHours, precip=p.precip),
    data = c2r,
    others = c("maxTemp", "relHum", "aveTemp"))
pred <- predict(initModel14,
                newdata = pred.data,
                interval = "predict")

## Wrapping the predictions as a matrix
z <- matrix(pred[,"fit"], nrow=length(p.sunHours))
z2 <- z
z2[z2 > 0.55] <- NA
## First an image:
image.plot(p.sunHours, p.precip, z2, xlab = "sunHours",
            ylab = "Precipitation",
            col=heat.colors(20)[20:1])
## Adding a contour:
contour(p.sunHours, p.precip, z2,
        add=TRUE, labcex = 1.5, col="black")
points(precip ~ sunHours, data= c2r, cex=1, col="black")

#standard deviation of predictions
z <- matrix((pred[,"fit"]-pred[,"lwr"])/
            qt(0.975,df=initModel14$df.residual-2),
            nrow=length(p.sunHours))
z2 <- z
z2[z2 > 0.135] <- NA
image.plot(p.sunHours, p.precip, z2,
            xlab = "sunHours", ylab = "Precipitation",
            col=heat.colors(20)[20:1])
## Adding a contour:
contour(p.sunHours, p.precip, z2,
        add=TRUE, labcex = 1.5, col="black")
```

```
points(precip ~ sunHours,
        data= c2r, cex=1, col="black")

summary(initModel14)
#can we see that most of the error
#comes from noise of the data and
#little from the model itself?

#######Average temperature and relative humidity###########
p.aveTemp <- seq(-5.4,21,length=110)
p.relHum <- seq(48, 98, length=100)

pred.data <- pred.frame(reference =
                        list(aveTemp=p.aveTemp,
                            relHum=p.relHum),
                        data = c2r, others = c("maxTemp",
                        "sunHours", "precip"))
pred <- predict(initModel14,
                newdata = pred.data, interval = "predict")

## Wrapping the predictions as a matrix
z <- matrix(pred[,"fit"], nrow=length(p.aveTemp))

z2 <- z
z2[z2 < 0] <- NA
## First an image:
image.plot(p.aveTemp, p.relHum, z2,
            xlab = "Average␣Temperatur",
            ylab = "Relative␣Humidity",
            col=heat.colors(20)[20:1])
## Adding a contour:
contour(p.aveTemp, p.relHum, z2, add=TRUE,
        labcex = 1.5, col="black")
points(relHum ~ aveTemp, data= c2r, cex=1, col="black")

#std
z <- matrix((pred[,"fit"]-pred[,"lwr"])/
            qt(0.975,df=initModel14$df.residual-2),
            nrow=length(p.aveTemp))
z2 <- z
image.plot(p.aveTemp, p.relHum, z2,
            xlab = "Average␣Temperatur",
            ylab = "Relative␣Humidity",
            col=heat.colors(20)[20:1])
## Adding a contour:
contour(p.aveTemp, p.relHum, z2, add=TRUE,
        labcex = 1.5, col="black")
points(relHum ~ aveTemp, data= c2r, cex=1,
        col="black")
```

```
###########Precipitation and relative humidity###############
p.precip <- seq(0.01,75,length=110)
p.relHum <- seq(48, 98, length=100)

pred.data <- pred.frame(reference = list(precip=p.precip,
                        relHum=p.relHum), data = c2r,
                        others = c("maxTemp", "sunHours",
                        "aveTemp"))
pred <- predict(initModel14,
                newdata = pred.data, interval = "predict")

## Wrapping the predictions as a matrix
z <- matrix(pred[,"fit"], nrow=length(p.precip))

z2 <- z
z2[z2 <0] <- NA
## First an image:
image.plot(p.precip, p.relHum, z2,
            xlab = "Precipitation",
            ylab = "Relative␣Humidity",
            col=heat.colors(20)[20:1])
## Adding a contour:
contour(p.precip, p.relHum, z2,
        add=TRUE, labcex = 1.5, col="black")
points(relHum ~ precip, data= c2r,
        cex=1, col="black")

z <- matrix((pred[,"fit"]-pred[,"lwr"])/
            qt(0.975,df=initModel14$df.residual-2),
            nrow=length(p.aveTemp))
z2 <- z
z2[z2 > 0.18] <- NA
image.plot(p.precip, p.relHum, z2,
            xlab = "Precipitation",
            ylab = "Relative␣Humidity",
            col=heat.colors(20)[20:1])
## Adding a contour:
contour(p.precip, p.relHum, z2, add=TRUE,
        labcex = 1.5, col="black")
points(relHum ~ precip, data= c2r, cex=1, col="black")
##################################################


#############Region analysis#############
index <- expand.grid(climateRow = 1:nrow(c2r),
                    region = factor(paste0("R"
                                        ,1:8)))
Case2R <- c2r[index$climateRow, 1:7]
Case2R$region <- index$region
Case2R$totalR <-unlist(c2r[,10 + (0:7)*2])
```

```
Case2R$posR <- unlist(c2r[,11 + (0:7)*2])

head(Case2R)
Case2R$ratio<-Case2R$posR/Case2R$tot

regMod1<-lm(Case2R$ratio ~ (as.factor(region)
                            +aveTemp+maxTemp
                            +relHum+sunHours+precip )^2,
                            data=Case2R)
summary(regMod1)
par(mfrow=c(2,2))
plot(regMod1, main="")
#assumptions not fulfilled

#CHI SQUARE#
Total <- c(sum(c2r$R1total),sum(c2r$R2total),
            sum(c2r$R3total),sum(c2r$R4total),
           sum(c2r$R5total),sum(c2r$R6total),
           sum(c2r$R7total),sum(c2r$R8total))

Positiv <- c(sum(c2r$R1pos),sum(c2r$R2pos),
            sum(c2r$R3pos),sum(c2r$R4pos),
             sum(c2r$R5pos),sum(c2r$R6pos),
             sum(c2r$R7pos),sum(c2r$R8pos))

Negativ <- Total-Positiv
mat <- rbind(Positiv,Negativ)
colnames(mat) <- c("R1","R2","R3","R4","R5","R6","R7","R8")
mat
(test <- chisq.test(mat))
test$expected
par(mfrow=c(1,1))
mosaicplot(t(test$observed),shade=T)
#there is a difference check chi square!

#looking at auto-correlation
plot(initModel14$residuals)
lines(initModel14$residuals)
abline(h=0, col=2)
acf(initModel14$residuals)

# Add time as a variable to the linear model
# Add time index
c2r$time <- 1:nrow(c2r)

timeModel1<-lm(posRatio ~
    time+(precip+ relHum+ aveTemp+ pwl(aveTemp, ave.opt)+
    sunHours+sqrt(pwl(sunHours, 59))+maxTemp+I(maxTemp^2))^2,
    data=c2r)
par(mfrow=c(2,2))
```

```
plot(timeModel1)
reducedTimeMoel1 <- stepP(timeModel1)$object
summary(reducedTimeMoel1)

reducedTimeModel2 <- update(reducedTimeMoel1,.~.
    -1)
summary(reducedTimeModel2)

reducedTimeModel3 <- update(reducedTimeModel2,.~.
    -precip)
summary(reducedTimeModel3)

reducedTimeModel4 <- update(reducedTimeModel3,.~.
    -sqrt(pwl(sunHours, 59)))
summary(reducedTimeModel4)

reducedTimeModel5 <- update(reducedTimeModel4,.~.
    -aveTemp)
summary(reducedTimeModel5)

reducedTimeModel6 <- update(reducedTimeModel5,.~.
    -sunHours)
summary(reducedTimeModel6)

reducedTimeModel7 <- update(reducedTimeModel6,.~.
    -I(maxTemp^2):sunHours )
summary(reducedTimeModel7)

reducedTimeModel8 <- update(reducedTimeModel7,.~.
    -maxTemp:sunHours)
summary(reducedTimeModel8)

plot(reducedTimeModel8)
residualPlots(reducedTimeModel8)

acf(reducedTimeModel8$residuals)
par(mfrow=c(1,1))
plot(reducedTimeModel8$residuals,type=’l’)
abline(h=0,col=2)

AIC(reducedTimeModel8)
#################END####################
```