Danmarks Tekniske Universitet

02441
Applied Statistics and statistical software (R)

Case 2: Campylobacter

Agnieszka Golinska (s151222)
Manuel Montoya Catala (s162706)
Paulina Jaworska (s151330)

January 2017

Contents

1 Summary

The following report is a statistical analysis of the risk of a broiler flock to develop the Campylobacter regarding seasonality and different regions of Denmark. Throughout the report we work with preprocessed data. First, we started with data description, interpreting time series, interaction scatterplots between all of the variables, and boxplots for 8 different regions. Then, a Box-Cox transformation of the response variable was performed, showing that a square root transformation is the one to use in this case. The next step was a temporal analysis of the data, done in order to shoe the periodicity if Campylobacter development.
The first part of our main analysis was an investigation of the relationship between bacteria development and weather variables. Generalized Additive Model studied which transformation should be used on the predictor variables. Then, we performed two transformations of the initial multivariate linear model - quadratic expansion and split transformation. Based on that the final model was build up and tested. Second part of the analysis trained regional differences considering the proportion of infected flocks. At the end the results were discussed and the report was summed up.

2 Introduction

Campylobacteriosis is an infection of intestines caused by Campylobacter that in last few years was found to be a main cause of such infections in Denmark. Undercooked meat or cross-contaminated food can be a source of this bacteria. It appears that seasonality has a big influence on the amount of bacteria in meat and their spread.

Bacteria development on a chicken meat has been studied from 1998 to 2007. Ten chickens from each batch were examined for Campylobacter and every batch with at least one positive result was taken into account. Weather variables for the same time range were based on data from Danish Meteorological Institute and are the country wide averages for every week of the year. There are 8 different regions of Denmark considered.

In this report we investigate if any of the climate variables can explain the seasonality and what is the relationship between the weather and probability of risk of a broiler flock to develop the Campylobacter. Investigation of the presence of regional differences is also performed.

In order to do statistical tests of the data we need to build and train multiple regression linear models. Those models consist of variable y which is the response (in our case development of Campylobacter in a broiler) and the explanatory continuous variables $X = x_1, x_2, ..., x_p$ which are the variables (e.g. climate variables) that may influence the response. The general structure of the model is as follows:

$$y = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p + \epsilon$$

where β_i are the coefficients, x_p are the explanatory variables, and ϵ is the error term to capture other sources of variation.

3 Description of data

At the beginning of the project the data we werw given was distributed in three different files (data obtained before year 2002, between 2002 and 2005, and after year 2005) and consisted on different variables. We preprocessed this files using the instructions given, the process is described in a later section (see section5 and Appendix A). After that we were given the data already preprocessed, this data consisted of the variables:

- year: Year;
- week: Week within year;
- aveTemp: Average weekly temperature (C);
- maxTemp: Maximum weekly temperature (C);
- sunhours: Hours of sunshine per week (h);
- relHum: Average weekly relative humidity (%);
- daysPrecip: Days with precipitation per week;
- precip: Precipitation per week (mm);
- total: Number of broiler flocks slaughtered per week;
- pos: How many of those flocks were positive;
- total1 - total8: Number of broiler flocks slaughtered per week and per region;
- pos1-pos8: How many of those were positive – again per region.

The resulting data consisted of 537 observations of 25 variables. Five of them, average temperature, maximum temperature, hours of sunshine, relative humidity and precipitation per week were numeric observations, while the rest were integer. The most important climate variables are displayed in Table 1. Looking at this table we can observe that some information are missing, 32 for humidity and 74 for sunshine hours, this is important because if our linear model used these variables, then we will not be able to use those samples, if our final model does not use them then we have more samples. Analyzing the data we can observe that most of it is reasonable. However, questionable are minimum values for humidity and sunshine hours since in Denmark this situation is unlikely to happen.

	aveTemp	maxTemp	relHum	sunHours	precip
1	Min. :-5.400	Min. : 0.60	Min. : 0.00	Min. : 0.00	Min. : 0.00
2	1st Qu.: 3.900	1st Qu.: 8.80	1st Qu.:70.00	1st Qu.: 13.65	1st Qu.: 5.00
3	Median : 8.900	Median :14.50	Median :77.00	Median : 28.00	Median :12.00
4	Mean : 8.717	Mean :14.79	Mean :77.41	Mean : 32.36	Mean :15.26
5	3rd Qu.:13.800	3rd Qu.:20.60	3rd Qu.:86.00	3rd Qu.: 48.15	3rd Qu.:23.00
6	Max. :21.000	Max. :29.70	Max. :98.00	Max. :103.00	Max. :75.00
7			NA's :32	NA's :74	

Table 1: Summary of climate variables.

In order to estimate the relationship between the weather variables and the Campylobacter, we will create objective variable (response) "ratio of positive flocks" which is calculated as:

$$rp = \frac{\text{POSITIVE cases of detection of Campylobacter}}{\text{Total number of flocks tested}}$$

Initially, we will not differentiate between regions and this variable will be the ratio for the whole country in a given week, later on we will consider if there are differences between the regions.

In Figure 1 we can visualize how climate variables have been changed from year 1998 to 2008, as well as during the years themselves. We can observe that climate variables were following a periodic pattern with time. A important plot here is the proportion of positive flocks during these 10 years. It is easy to observe that throughout the summer about 80% of the flocks were infected, while in the winter season, when bacteria don't spread that easily, only 20% of them were positive. In those plots we can also see, like in the Table 1, that we miss some data for sunshine hours and humidity.

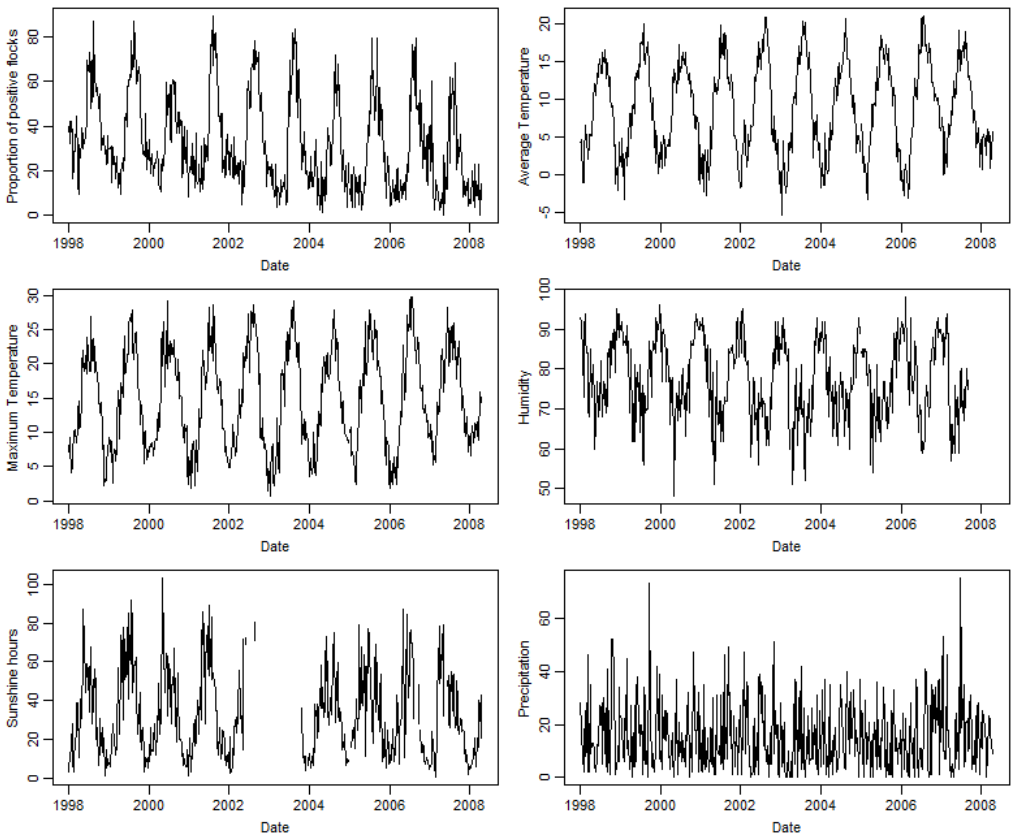

Figure 1: Time series plots of climate variables and proportion of positive flocks.

The next figure, Figure 2, shows interactions between climate variables and proportion of positive flocks. We can observe strong relationships within the climate variables, for example between humidity and sun hours or average and max temperatures (almost linear relation). However, it is more important to investigate interactions between the climate variables and proportion of positive flocks. Thanks to this figure we can observe that the average and maximum temperature have a strong influence on Campylobacter development, while humidity, sunshine hours and precipitation are not so meaningful. However, we can observe clusters in their maximal (for humidity) or minimal (sun hours and precipitation) values which can mean that these points can be valuable (we are more sure about them since they are not accidental).

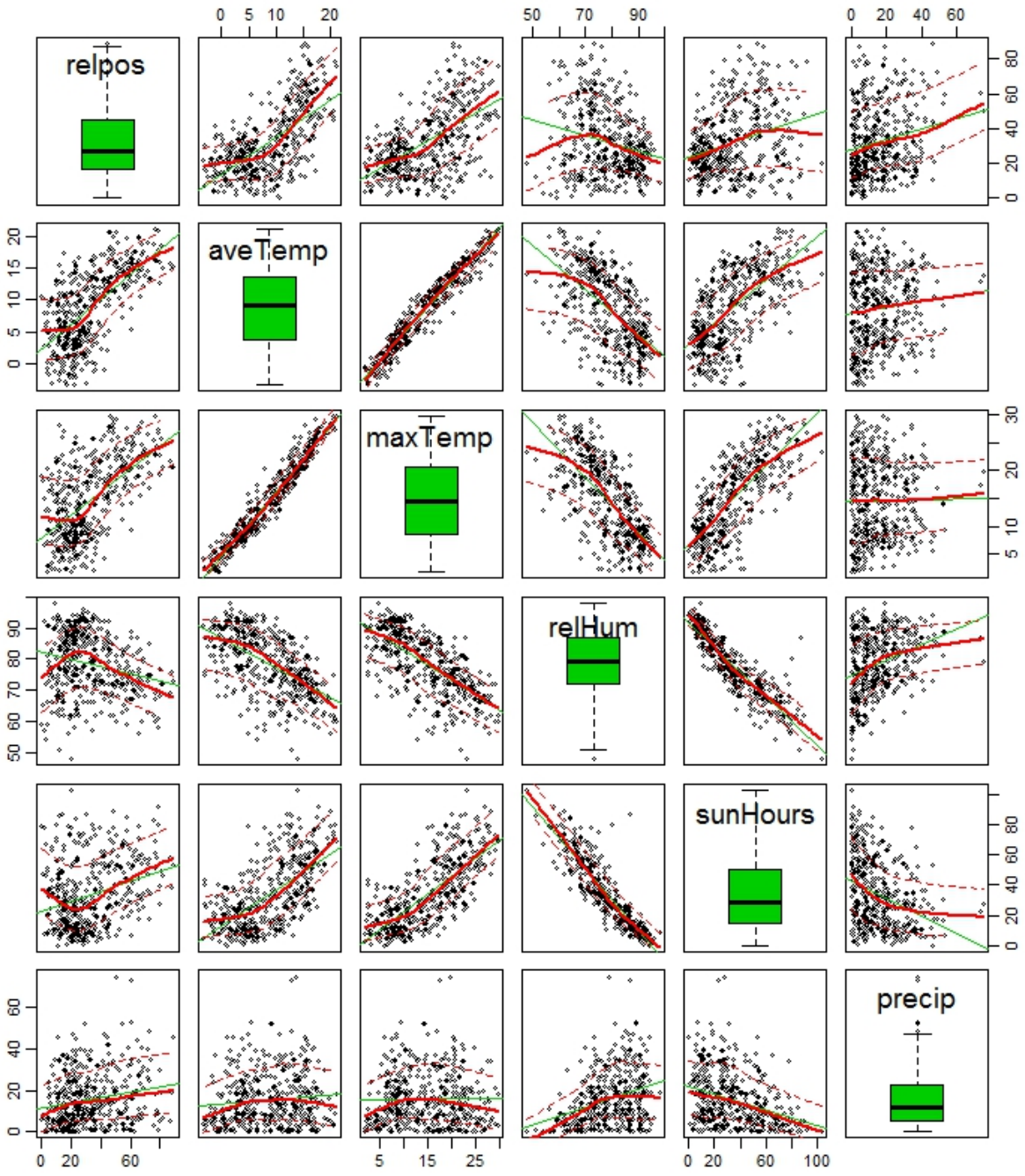

Figure 2: Interactions scatterplot between variables.

3.1 Box-Cox transformation of the Response variable

We will initially consider a transformation of the response variable to reduce possible non-normality of the errors of the linear models that we will further train with the data. We will use the Box-Cox transformation in this regard. In order to do so, we had to add a bias of 0.0001 to those cases where the ratio was 0 since the BoxCox transformation is only applicable to positive data.

The Box-Cox plot (Figure 3 top) shows a λ value around 0.5 which indicates that a square root transformation of the variable should be performed. The next figure (Figure 3 bottom) shows the Box-Cox plot for the transformed response variable. We can appreciate how after the transformation, the λ value lies around 1, we consider the transformation successful.

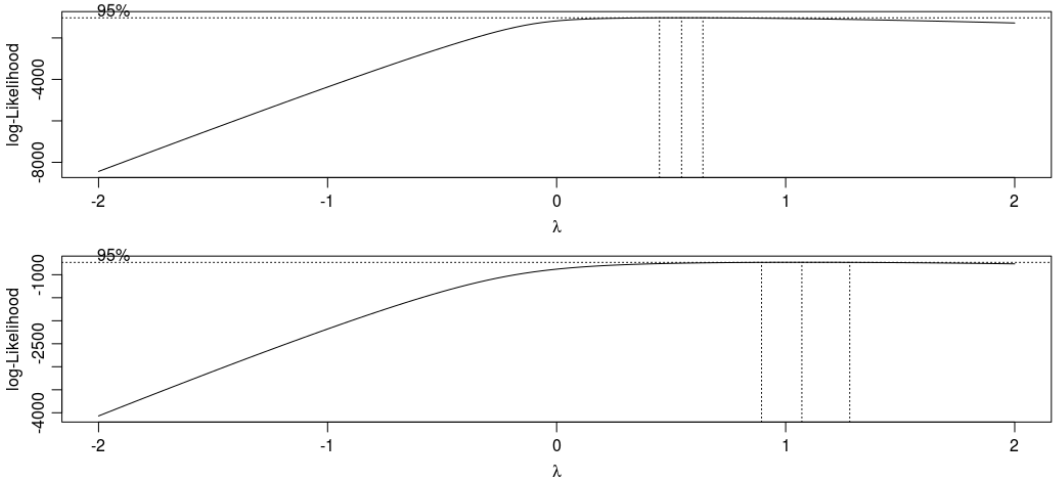

Figure 3: BoxCox plot of the original and transformed data

We can also look at the histogram of the response variable before and after the transformation. Next figure (Figure 5) shows those distributions. We can appreciate how the transformation makes the distribution of the Response variable more Gaussian-like (Figure 5 bottom).

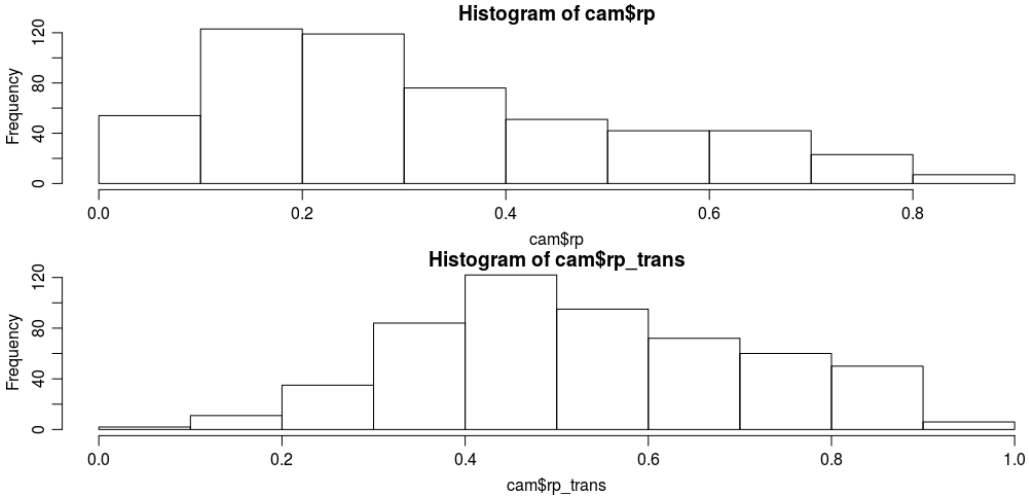

Figure 4: Histogram of the original (top) and transformed (bottom) data.

During the following steps we will study the statistical relationships between the response variable (ratio of flocks with positive Campylobacter) and the explanatory variables. We will mainly fit linear models in which we will assume that the different samples are independent and the residual is Gaussian noise. We will need to check those assumptions once the model is fitted to the data. From now on, we will use this transformation of the response variable.

We can see now how the scatterplot look like related to the new transformed response variable. We would expect the relations to be more linear since that is the main purpose of the transformation. The next figure (Figure 5) shows such relationships. As we can observe:

- The relation between the response variable and aveTemp, maxTemp is now much more linear. Before the transformation we saw that a quadratic transformation of these variables could be a good option, it is the same as to take the square root of the response variable.
- We have not gained much linearity for the rest of the samples, although their variance seems to be more uniform. This is alright since the variables that had the most information where the temperature ones and so, the BoxCox transformation is bias towards a good transformation of these.

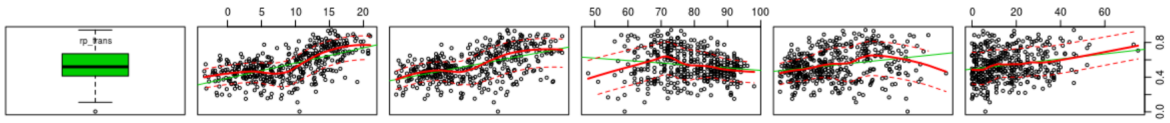

Figure 5: Interactions scatterplot for the transformed response. From the left: response, aveTemp, maxTemp, relHum, sunHours, precip.

3.2 Temporal analysis of the data

In this report we mainly try to investigate the relationships between the Campylobacter and the weather conditions. We also have the time information, but we are not using it. The week variable itself is very predictive, due to its periodic nature with the Campylobacter, but correlation does not imply causality. Our model should be able to extract this information from the weather variables.

Of course we are missing some information if we do not use the time data at all. The weather conditions are correlated in time series and we are not taking this into account in our model. Moreover, we will not be able to detect patterns such as: if a high average temperature and relative humidity continues during 2 weeks, then the Campylobacter will increase.

The figure below shows the scatter plot between the week information and some of the important variables. We can see how the week information has a high predictive power due to the periodicity of weather within a year, but once again, time itself is not the cause.

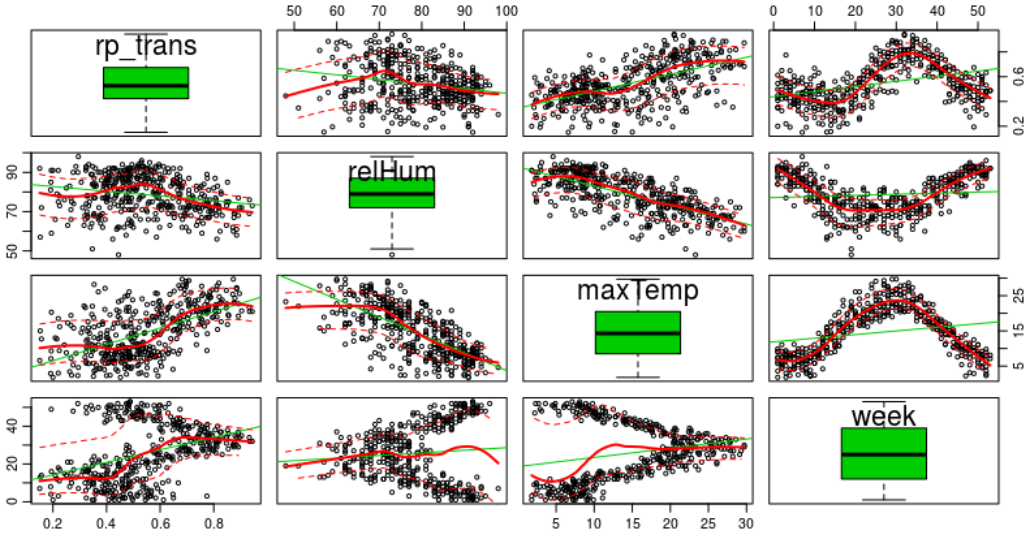

Figure 6: Time Scatter

We could however, use the time information in other way, not as an input variable directly. For example, maybe the weather at a given week actually affects the Campylobacter measures in the week after because it takes time for the Campylobacter to grow. Or maybe we should use a simple mean average of the weather variables to both reduce measurement noise and maybe the bacteria depends on the accumulation of the variables during a certain ammount of time. We did not have time to develop these ideas but it is something to take into account.

4 Statistical analysis

In this section we present our statistical analysis of the case about Campylobacter development in different weather conditions and in different regions. R code used to do the analysis is attached in Appendix B.

4.1 Relationship between weather and bacteria development

4.1.1 Generalized Additive Model

The first thing we can do after observing the data is fitting a Generalized Additive Model (GAM). In this linear model, the explanatory variables x_i (see introduction) are transformed by a smooth function $s_i(x_i)$, which aim is to maximize the linear relation between the response and explanatory variables, yielding the equation:

$$y = \beta_0 + s_1(x_1) + ... + s_p(x_p) + \epsilon$$

We can use this model to get an idea about the kind of transformation to be used on our explanatory variables in later models since this GAMs have high expressivity and their transformation functions $s_i(x_i)$ are able to capture the shape of the relationship between the response y and the explanatory variables x_i in a smooth way.

So we train the GAM using the ratio of positive flocks as the response variable and we use the transformation of the five weather variables: average temperature, maximum temperature, relative humidity, hours of sunshine, and precipitation; as the explanatory variables.

Figure 7 shows the transformation of the explanatory variables. The 5 graphs show in the x-axis the original independent explanatory variables x_i, and in the y-axis their transformation $s_i(x_i)$. The solid lines are the transformation values, while the dashed lines are confidence intervals of such transformations.

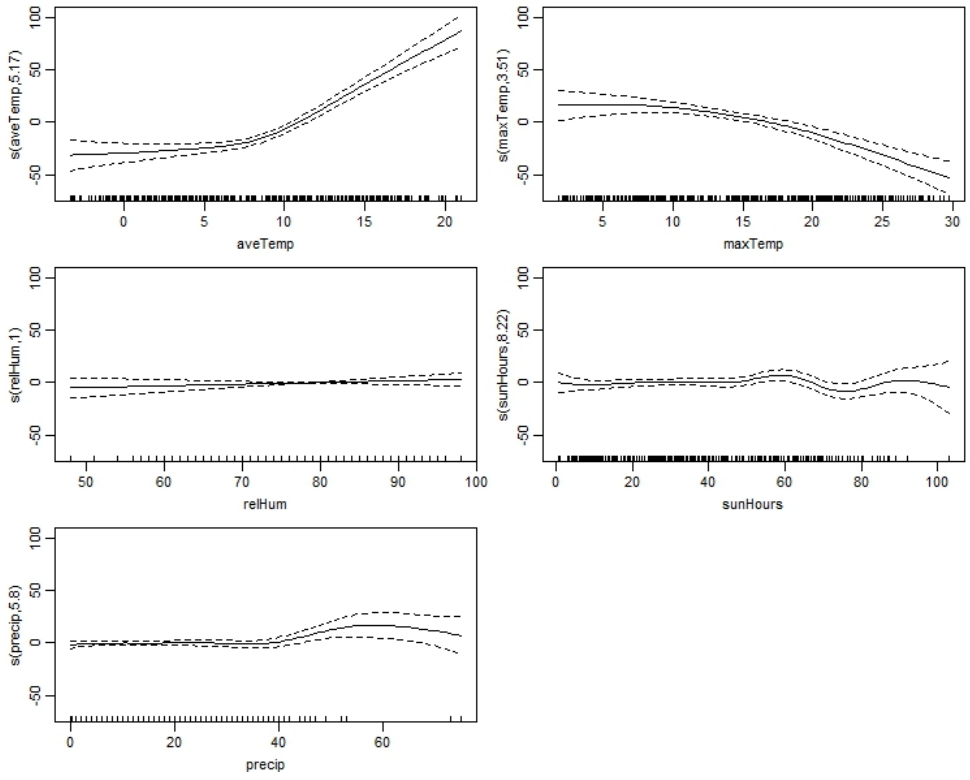

Figure 7: Transformations of the GAM model using the ratio as response variable.

From the graphs we can make the following observations:

- The more samples we have in a given range, the narrower the confidence interval (e.g. precipitation plot);
- Both shapes of temperature plots (average and maximum temperature) make us consider a square root transformation to be the best option for this data. Moreover, they appear to be most important variables considering Campylobacter development.

- Humidity has a clear linear behavior at 0, which means that it doesn't have to be transformed. This is probably due to a more less the same amount of samples along x axis;
- Sunshine hours and precipitation at first follow the same linear behavior, but their slope is close to 0. Due to that, these variables are probably just not important. Also, the area at the end where the transformation is not 0 is an area with little samples and high uncertainty, so probably the transformation is also 0 in this area.

If we use as the response variable the square root transformation of the ratio, given by the Box-Cox transformation, we obtain the following GAM functions (Figure 8):

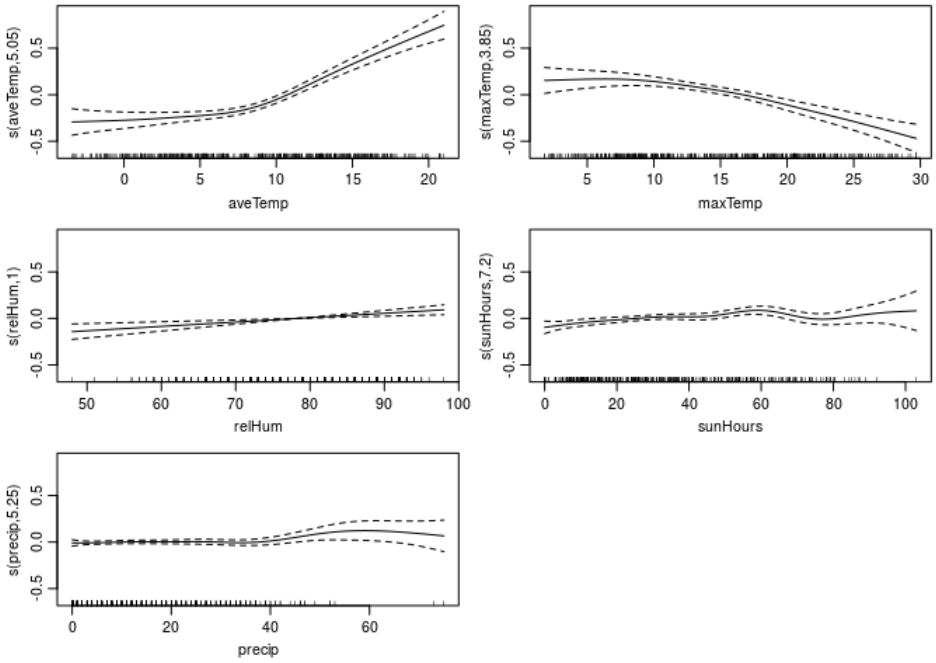

Figure 8: Transformations of the GAM model using the sqrt(ratio) as response variable.

As we can see in Figure 8, the transformation functions are very similar to the previous ones, so we can apply the same reasoning.

The significance of the parameters of the model can be seen in the tables below. As we can see, the bias (intercept) significance is very big (Table 2), and also the significance of the transformed temperatures (Table 3). Relative Humidity is somewhat significant at a 0.06 level and sunHours and precip (Table 3) offer little information.

	Estimate	Std. Error	t value	Pr($>\|t\|$)
(Intercept)	0.55	0.01	99.14	0.00

Table 2: Summary of the intercept of the GAM model.

	edf	Ref.df	F	p-value
s(aveTemp)	5.30	6.46	21.90	0.00
s(maxTemp)	3.55	4.56	8.23	0.00
s(relHum)	1.00	1.00	3.68	0.06
s(sunHours)	1.59	1.99	0.60	0.53
s(precip)	5.43	6.49	1.13	0.28

Table 3: Summary of the transformations of the GAM model.

This model has a mean squared error of 0.11 and a Multiple R-squared of 0.565, so this will be our starting point. We should also check the assumptions of Gaussianity and Independence of the residual. The following two Figures (Figure 9 and 10) show the analysis of residuals of the GAM model. We will elaborate more in the meaning of these graphs later in the report but in the first figure we can appreciate

in the Q-Q plot that the distribution of the residual could be considered Gaussian, the mean and variance of the residual does not seem to change with the fitted values, although we can 2 differentiated clusters of fitted values. In the second figure we check the dependency of the residual with the input variables and time. We can observe that the residual is mostly independent (its mean and variance) of the variables. In the time dependency, we can still appreciate a somewhat sinusoidal form, so this model could not learn all the time information from the weather variables.

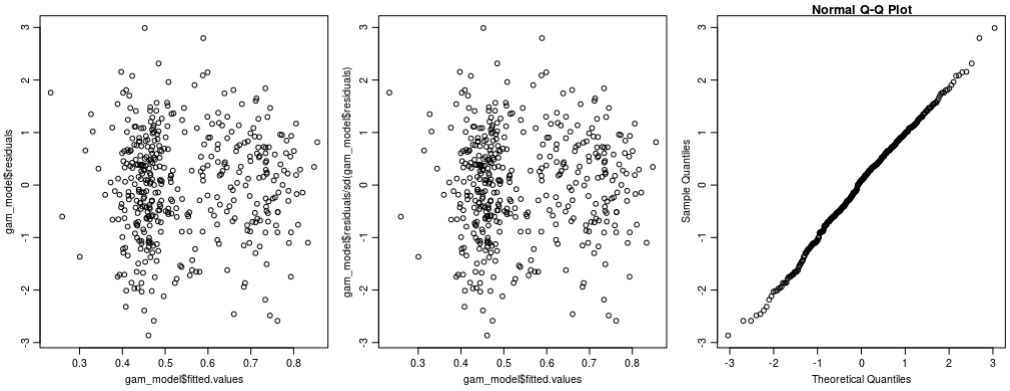

Figure 9: Analysis of Guassianity of the residual of the GAM model.

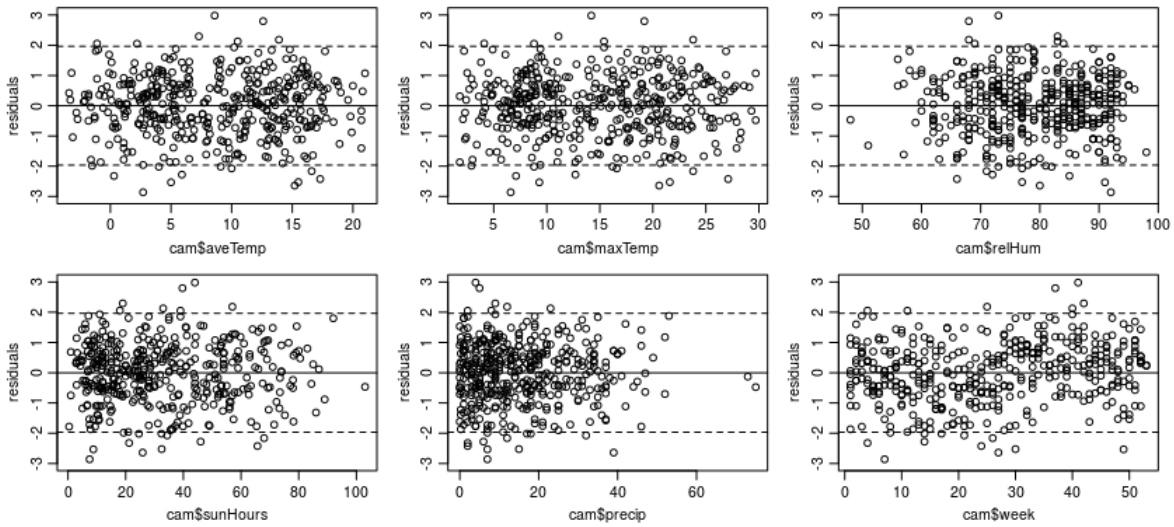

Figure 10: Dependence of the residual of the GAM model with the explanatory variables.

Due to the high expressivity of this model, it will be hard to find a basic linear model with better characteristics if we only look at the residual error of the samples used for fitting the model. But we can use this model as an initial upper bound and it gives information about which variables are significant (temperature and humidity) and which ones are not, and which transformations we should perform on them.

We also did not use any combination of the explanatory variables in this model, because due to the high expressivity of it, we could end up overfitting the training data, and the system would not generalize good for new samples. A simple combination that we could do is the multiplication of maximum temperature and relative humidity. Doing this we get a more expressive model (less residual) and as we can see in Figure 11, the transformation of some of the variables change.

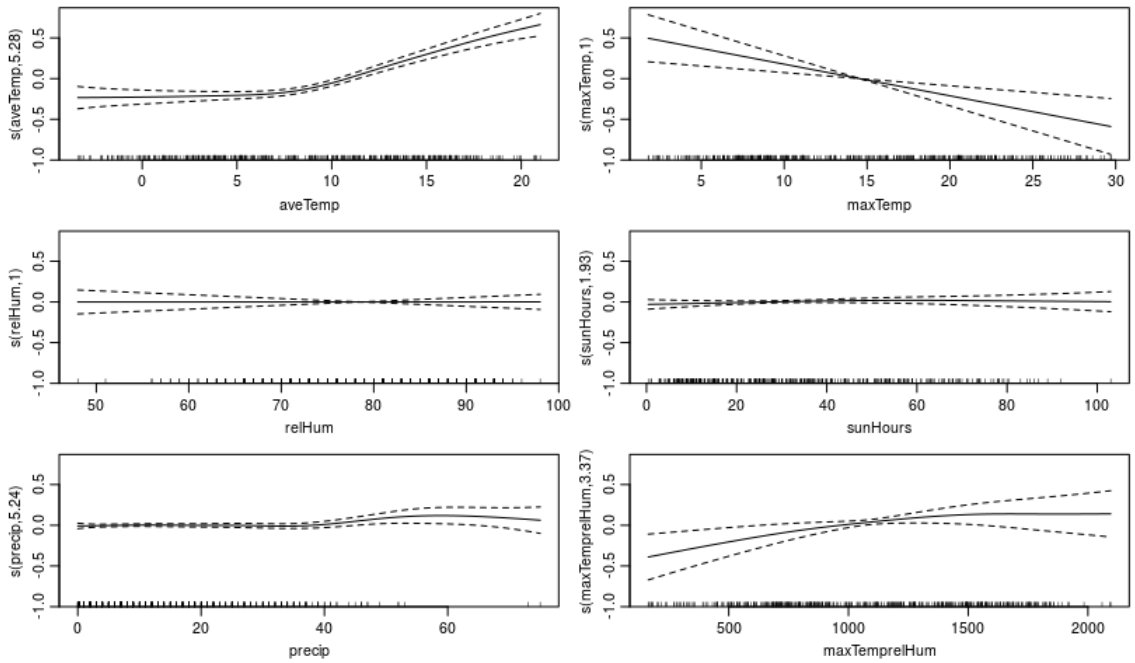

Figure 11: Transformations of the GAM model using the sqrt(ratio) as response variable and the combination variable.

Now, the relative humidity alone is not significant, the new combined variable is significant and the maxTemp only requires a linear transformation. We can see in this example how the importance of a variable depends highly on the others and the model.

In the following we will use normal linear models to fit the data, we will perform transformations of the explanatory variables to be able to model non-linear relationships, we will detect outliers and check the Gaussianity assumptions and independence assumptions of the residual of the models.

4.1.2 Initial Multivariate linear model for the ratio of positive flocks

Given the information about the variables gathered in the previous sections, we will formulate and test linear models in order to find a good estimation function. As a baseline, given the relation between the samples we will use the model:

$$\begin{aligned}\sqrt{ratio} =& \beta_0 + \beta_1 aveTemp + \beta_2 maxTemp + \\ & \beta_3 relHum + \beta_4 sunHours + \beta_5 precip + \epsilon.\end{aligned} \tag{1}$$

When we fit this linear model, we see that the significance of the coefficients for the variables sunHours and precip are very low, their p-values are 0.11 and 0.44, respectively. Regarding that, we proceed to reduce the model with the *step()* function in both directions. The parameters of the final model are shown in Table 4. As we can see, all the parameters are significant except for the variable sunHours and the bias.

	Estimate	Std. Error	t value	$Pr(>\|t\|)$
(Intercept)	0.1410	0.1740	0.81	0.4186
aveTemp	0.0410	0.0056	7.28	0.0000
maxTemp	-0.0196	0.0055	-3.59	0.0004
relHum	0.0037	0.0018	2.07	0.0397
sunHours	0.0013	0.0008	1.49	0.1375

Table 4: Parameters of the initial model.

Now, lets check the Gaussianity and independence assumption of the residual for this model. Figure 12 shows an analysis of the residuals for this model. We can observe that:

- In the Residuals vs fitted graph (top left) we can see some curvilinear pattern, we tend to have positive residuals in the extremes and negative residuals in the middle. In the standardized error

graph (bottom left) this pattern disappears because the sign information disappears and the variance of the residual is pretty much constant independently of the fitted value, which is good.

- The QQ plot (top right) shows a distribution that is very close to Gaussian in most of the domain.
- We can see how the samples 436, 382 and 331 have a very high Cook's Distance and seems like an outlier in the remaining graphs (bottom right). We will remove them in the further analysis.

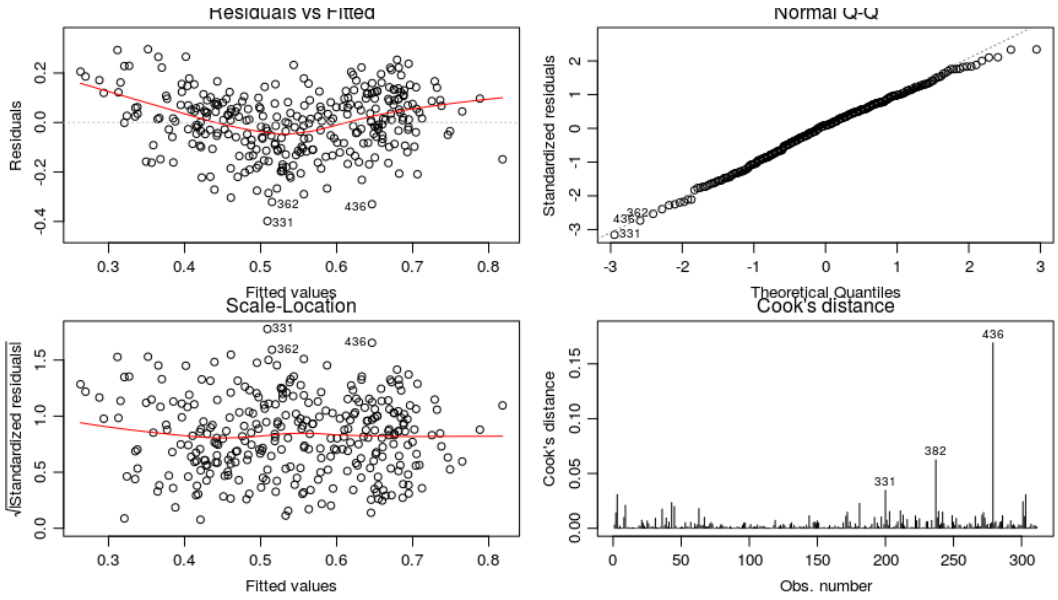

Figure 12: Analysis of the Residual for the initial model.

Once the outliers are removed, all the parameters of the model become statistically significant and the overall previous analysis of the residual remains the same. For this model we have a residual standard error: 0.1281 on 417 degrees of freedom with Multiple R-squared: 0.4535.

Now lets look at the residuals as a function of the different explanatory variables. If no pattern is identified we can accept the independence and guassianity assumption. Figure 13 shows such analysis, also viewing the residual in terms of time, viewing the residual against the week number of the samples.

- We can appreciate a curvilinear "V" pattern for aveTemp and maxTemp, and we should make a transformation of this variables that removes this pattern. It was suggested by the shape of the GAM transformation as well. We might use a quadratic transformation or model it with a spline.
- No pattern is seen for relHum, sunHours, or precip.
- Regarding the time dependency with the week number, we see a clear error pattern which is similar to a cosine function, maybe due to periodicity of the problem. So we kind of failed so far to try to explain the initial time dependency pattern using the weather variables. Maybe if we get rid of the "V" shape in the temperature variables, this pattern will be reduced.

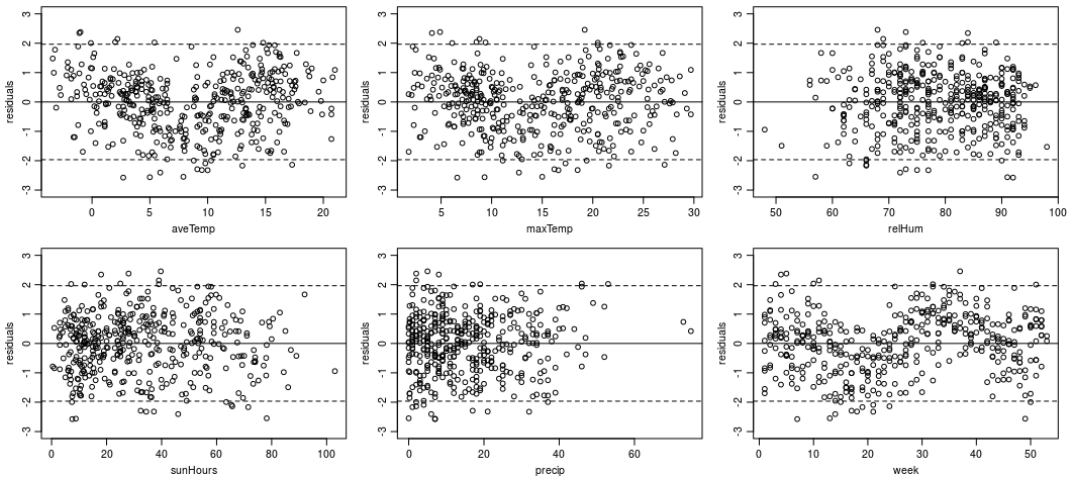

Figure 13: Analysis of the residuals as a function of the different explanatory variables.

Now, we will try to find transformations and combinations of the explanatory variables that will be easy to learn for the learn model, that is, we find new variables that are a transformation of the original explanatory variables. Doing that we hope that this variables have a linear relationship with the response variable, that is, they are correlated with the response variable.

4.1.3 Quadratic expansion of the individual variables

All the previous studies indicate that we should perform a transformation of the temperature variables, the first transformation we try is the quadratic expansion of the variables, in order to give more expressivity to the model. The equation of the model looks as follows:

$$\begin{aligned}\sqrt{ratio} =& \beta_0 + \beta_1 aveTemp + \beta_2 maxTemp + \\ & \beta_3 relHum + \beta_4 sunHours + \beta_5 precip + \\ & \beta_5 aveTemp^2 + \beta_6 maxTemp^2 + \epsilon.\end{aligned} \tag{2}$$

After reducing the model with the *step()* function, we get a model with the following parameters (Table 5). As we can see, the model now only depends on 4 variables; we see that sunHours, precip and aveTemp (linear part) have been erased. The variables with "_2" mean that they are the squared value of their root variable.

	Estimate	Std. Error	t value	Pr($>\|t\|$)
(Intercept)	0.1481	0.0845	1.75	0.0803
maxTemp	0.0125	0.0038	3.30	0.0011
relHum	0.0027	0.0009	3.12	0.0020
aveTemp_2	0.0024	0.0002	11.49	0.0000
maxTemp_2	-0.0010	0.0002	-6.08	0.0000

Table 5: Parameters of the reduced quadratic model.

The residual standard error is 0.1171 on 417 degrees of freedom with Multiple R-squared 0.5354 so our model has improved significantly (of course it would improve since we are adding more variables, worst case scenario is that it is as bad as before). The residuals for the different variables and time is shown in the figure below (Figure 15), and as we can see, the patterns on the residual have been reduced but they still exist.

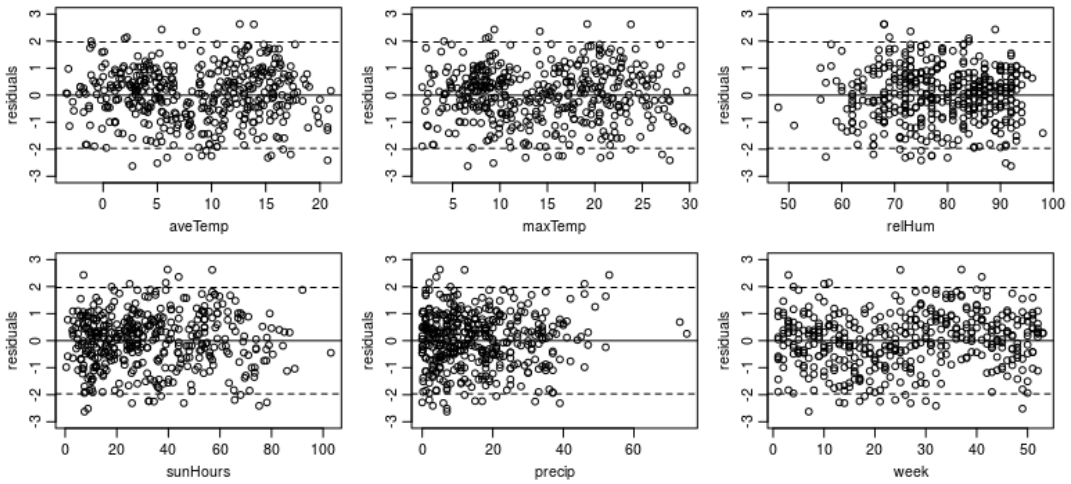

Figure 14: Analysis of the Residual for the reduced quadratic model.

4.1.4 Split transformation of the variables

As a next step we can try a linear spline of the variables aveTemp and maxTemp to make a more expressive model. What we have done is:

- Using the optimization function of R, we first find the best 1-knot spline of the variable aveTemp that minimizes:
  - the residual of the reduced model (using the *step()* function) of the linear model fitted to the initial variables plus the spline;
  - the optimal value found is $aveTemp_{knot} = 7.785$ which is right in the centre of the "V" shape of the residuals seen before.
- We add the spline variable and we do the same process to find the optimal spline point for the maxTemp variable
  - the optimal value found is $maxTemp_{knot} = 8.00$.

Now we have these 2 new spline variables, which are actually just a subset of the original aveTemp and maxTemp variables. Thanks to them, the model will be able to fit different slopes to the different regions of the split. The results of this model are similar to the ones previously obtained with the quadratic model. The analysis of the parameters (Table 6) and the residuals (Figure 15) are seen below.

	Estimate	Std. Error	t value	Pr(> |t|)
(Intercept)	0.4173	0.0130	32.06	0.0000
aveTemp	0.0174	0.0033	5.35	0.0000
aveTemp_pwl	0.0482	0.0046	10.53	0.0000
maxTemp_pwl	-0.0259	0.0035	-7.45	0.0000
precip	0.0011	0.0005	2.27	0.0240

Table 6: Parameters for the model after split transformation.

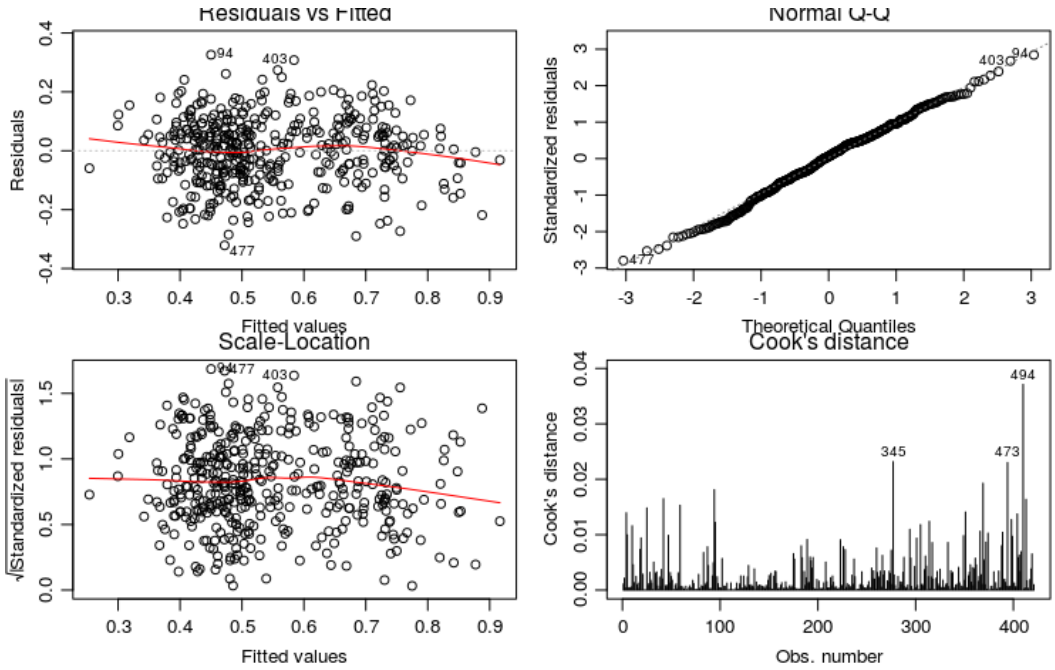

Figure 15: Analysis of the Residual for model after split transformation.

As we can see, the spline partitions are significant to our model. The partitions are expressed as:

$$aveTemp\_pwl = (aveTemp > 7.785) * (aveTemp - 7.785)$$

$$maxTemp\_pwl = (maxTemp > 8.0) * (maxTemp - 8.0)$$

Also now, the variable precip is significant, which highlights how the importance of a variable in a given model depends heavily on its relation with the rest of the variables.

4.1.5 Final Model: Combining Quadratic transformation and the Split transformation of the variables

In order to build the final model we will combine both approaches (quadratic and split transformation) and we will start with a model that takes all the initial variables, plus the splines, plus the quadratic forms of the temperatures, in the normal and split forms. After reducing with *step()*, the final model is:

$$\begin{aligned}\sqrt{ratio} =& \beta_0 + \beta_1 aveTemp\_pwl + \beta_2 maxTemp + \\ & \beta_3 maxTemp^2 + \beta_4 maxTemp\_pwl^2 + \\ & \beta_5 relHum + \epsilon.\end{aligned} \tag{3}$$

So the model mainly depends on the maximum temperature (3 transformations of it) but also on the average temperature and the relative humidity.

The model has a Residual standard error of 0.1128 on 415 degrees of freedom with a Multiple R-squared of 0.5704. Table 7 shows the significance of the parameters. As we can see, now relHum has become significant. The spline variables are the most important for the model. So the model is focusing on what is happening when the temperatures are higher than 8 degrees. These are the areas with the highest ratio, so they could yield the biggest error if the bias (intercept) is low.

Checking the Gaussianity and independence of the residuals of this final model (Figure 16), we can observe a pattern in the Residuals vs fitted - there is a differentiated cluster of points right at 0.5 and then there are less points. This is most likely caused by the spline, because there is a discontinuity on the models applied. But the variance and mean of the residual does not change with location and the Q-Q of the plot indicates high Gaussianity.

	Estimate	Std. Error	t value	Pr(> |t|)
(Intercept)	0.1650	0.0772	2.14	0.0331
aveTemp_pwl	0.0707	0.0054	13.09	0.0000
maxTemp_pwl	-0.0250	0.0099	-2.53	0.0119
relHum	0.0026	0.0009	3.04	0.0025
maxTemp_2	0.0014	0.0005	3.02	0.0027
maxTemp_pwl_2	-0.0025	0.0004	-5.61	0.0000

Table 7: Parameters of the final model.

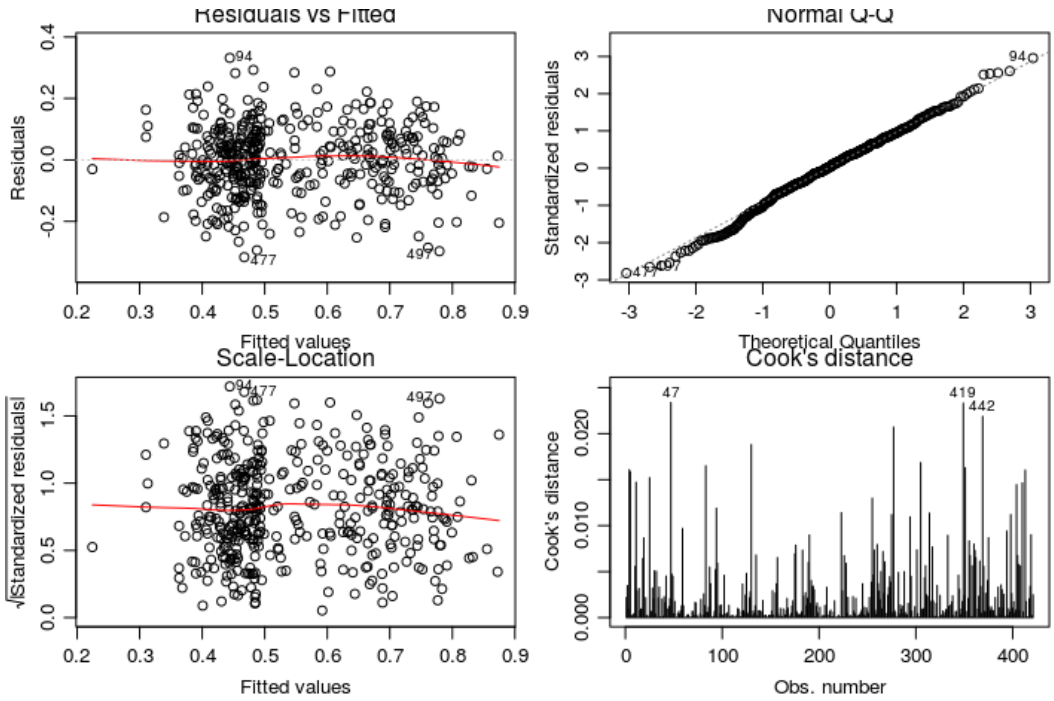

Figure 16: Analysis of the Residual for the final model.

Finally, we check the properties of the residual in terms of the individual variables and the time. As we can see in Figure 17, the patterns of the residual with aveTemp and maxTemp have almost completely disappeared and the sine shape regarding time has been reduced.

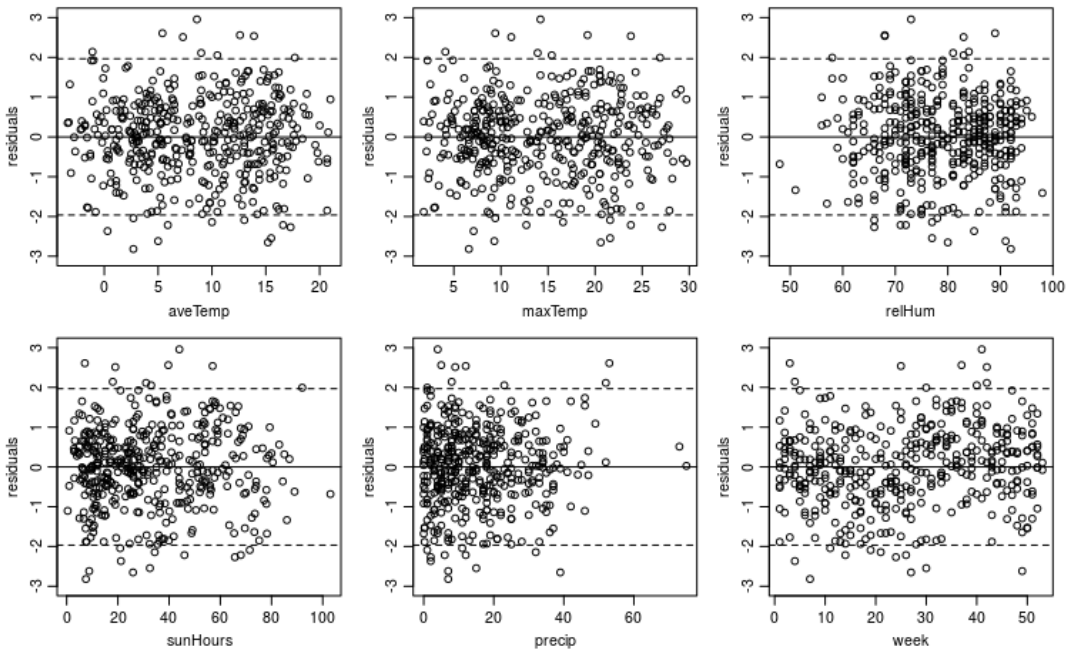

Figure 17: Analysis of the Residual for the final model in terms of variables.

Now that we have our final model we can do some predictions on the data to see how our prediction system works. What we can do is:

- We set all the explanatory variables except one of them to a fixed value, usually a common value, so usually the mean, and that is what we are doing here.
- We make a grid of the variable we did not set.
- We predict with our model the output for the grid search and we can also plot points that lie within the area of the plot to check that our prediction in that area fits the data

In our case, we have only 3 explanatory variables and we made grid search on maxTemp since it is the most significant. We set relHum to its mean and we set the aveTemp to different values, yielding the graphs we see below. In the graphs we also plot set of points that are within the 2 degrees range of the aveTemp. Figure 18 shows the results of the prediction.

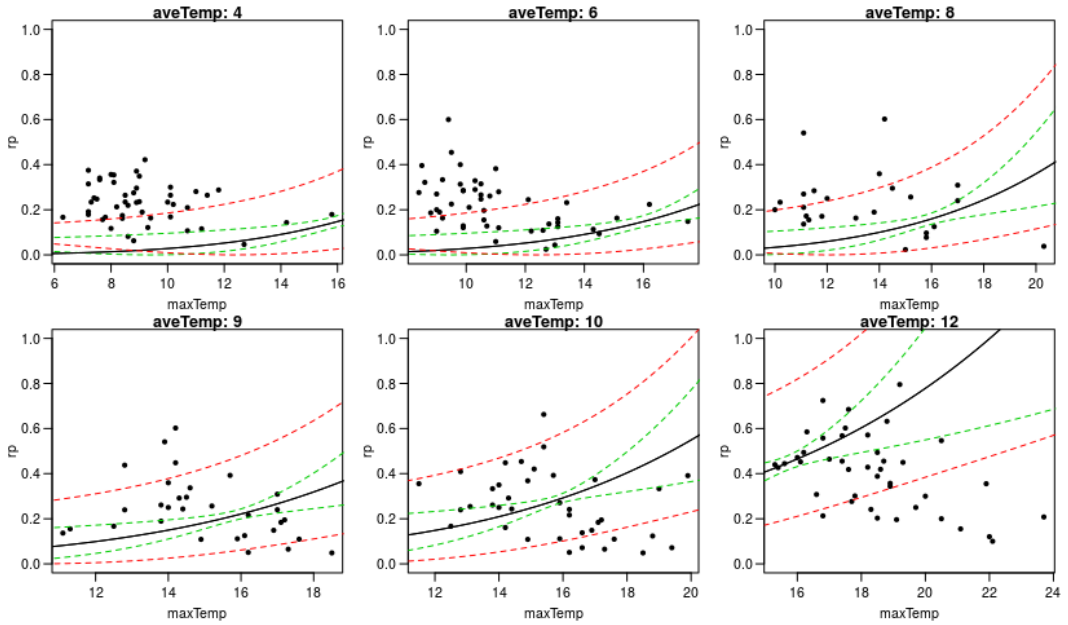

Figure 18: Prediction plots.

Some observations about the graph are:

- The predictive model is the same for aveTemp equal to 4 and 6. This is because the model only depends on the spline of aveTemp, so aveTemp will only have an effect when it is greater than 7.85. This fact is the one that creates the cluster of points the Fitted-Residual graph seen before.
- The model fits best the points when the aveTemp is around 9. This so happens to be the average of aveTemp and where most of the points are, and it is middle way between the extremes, so it makes sense that the model focus in this area to minimize the error.
- There is still some variance of the points that comes from the relHum and the range of points around aveTemp that we are plotting.
- The statistical model fails for low and high maxTemp values, many points observed do not fall into the 95% observation interval.

This is the final model we have chosen. Notice that we also tried some combination of the variables, such as maxTemp*relHum, and even though they improved the residual of the model, it was not a significant change and we did not want to complicate the model.

4.2 Analysis of regional differences

4.2.1 Comparison of the regions

It is of interest to investigate how the infected flocks differ from one region to another. In Table 8 information about every region can be found. We can see that different amounts of flocks have been tested in different regions, and the ratio of positive flocks (that developed the bacteria) to the total number of flocks also differs, this could mean that the relation between the ratio of positive flocks depends on the region, therefore we should fit different models to every region. It also tells us as the contribution of each region to the total number of flocks is different so our previous models could be biased to a particular region.

	Positive	Negative
Region 1	4937.00	3343.00
Region 2	4707.00	3166.00
Region 3	7314.00	4783.00
Region 4	2564.00	1367.00
Region 5	5770.00	4213.00
Region 6	3403.00	2244.00
Region 7	1929.00	1262.00
Region 8	3125.00	2159.00

Table 8: Number of infected flocks in particular regions.

In order to make the comparison between regions more visible, a mosaic plot was created (Figure 19). It was divided into two blocks: negative (healthy) and positive (infected) flocks. Each of these blocks was then split into eight different regions. We can notice that this figure confirms data included in Table 8. It can be immediately observed that there were more positive than negative samples, and also which region had the most (region 3), and which the least observation (region 7). However, the most important information that the mosaic plot provides us, is the difference in a proportion of negative and positive flocks. Looking at this figure we can observe that for region 4 and 5 this proportion varies from the rest.

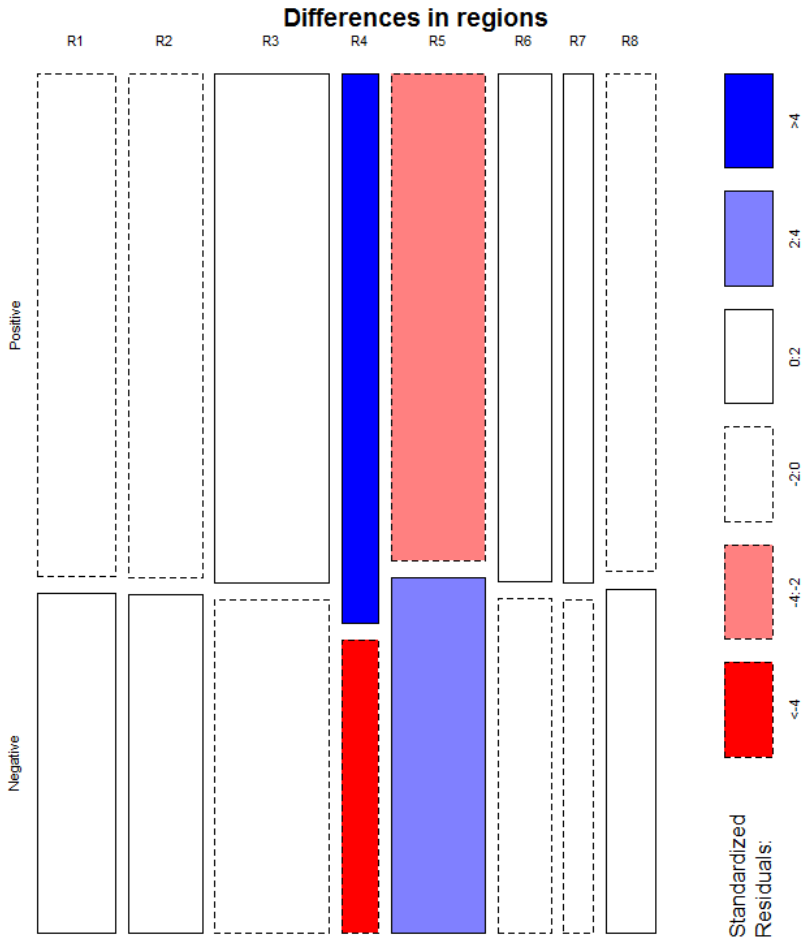

Figure 19: Mosaic plots of different regions.

Figure 20, where the distribution of observations for every of the regions is presented, confirms our observations from mosaic plot (shown above). There is a different number of observations in the regions, which can then influence the response variable. We can also observe that there are many point that lie either in 100 or 0, meaning that all the flocks where either infected or not. In many cases this is due to the fact that we have very little samples for a Region in a given week.

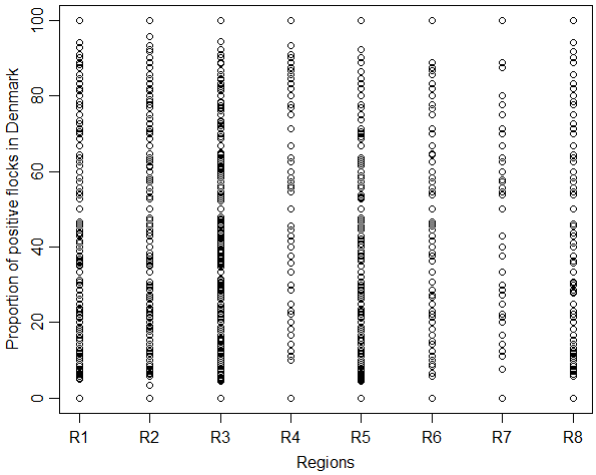

Figure 20: Distribution of observations in different regions.

In order to observe correlations among the regions Table 9 was made. We can observe that they are indeed correlated with each other - the values are from 0.25 to 0.56. The strongest interaction is between

proportion of positive flocks in region 1 and region 3. While the weakest relation is among region 4 and 6.

	Region 1	Region 2	Region 3	Region 4	Region 5	Region 6	Region 7	Region 8
Region 1	1.00	0.49	0.56	0.40	0.52	0.45	0.44	0.44
Region 2	0.49	1.00	0.51	0.39	0.41	0.43	0.43	0.44
Region 3	0.56	0.51	1.00	0.39	0.51	0.44	0.43	0.43
Region 4	0.40	0.39	0.39	1.00	0.39	0.25	0.30	0.35
Region 5	0.52	0.41	0.51	0.39	1.00	0.39	0.44	0.41
Region 6	0.45	0.43	0.44	0.25	0.39	1.00	0.34	0.35
Region 7	0.44	0.43	0.43	0.30	0.44	0.34	1.00	0.33
Region 8	0.44	0.44	0.43	0.35	0.41	0.35	0.33	1.00

Table 9: Correlation between the regions

Looking at Figure 23 and taking into account all the information that we obtained about the regional differences, no geographical correlations between regions were found.

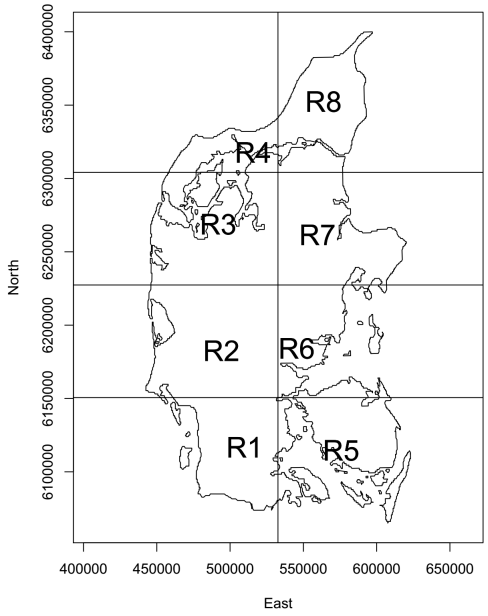

Figure 21: Map with the regions.

We decided to eliminate those samples in which the total number of flocks tested in the region is lower than 5. This way we will eliminate the extreme values of the ratios obtained, reducing the ”sampling noise” of our observations. Figure 22 shows the boxplots of the square root transformation of the ratio of positive flocks (as in the previous section) for the different regions. It is easily seen that region 4 differs from the others - its mean is much higher. This indicates that in this region bacteria development is increased comparing to the other regions (around 40% of flocks were found to be infected). The rest of the means fluctuate between 20% and 30%.

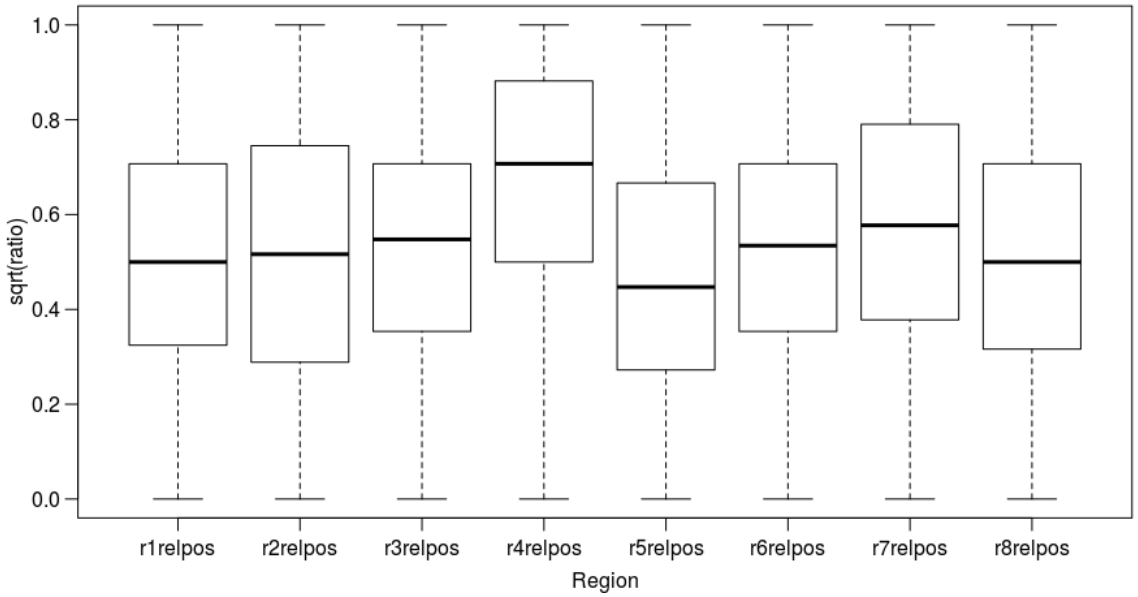

Figure 22: Boxplots of ratios in different regions.

We can check if the mean of the different Regions differ significantly by building a factor model of the response variable with the Region factor.

$$\sqrt{ratio} = regionFactor + \epsilon$$

The analysis of the factor significance is shown in Table 10. The row regionrXrelpos is the contribution of the value X of the factor region (the factor is named "region" and the 8 different values are "regionrXrelpos"). From the table we can see that region 1 (intercept,global mean), region 4, and region 5 are those whose ratio means differ from the means of ratios in other regions. This observation is based on the p-values that are lower than 0.05. We can see that Region 4 has a lower number of samples than the other region so the difference could be caused by sampling error. We see that Region 5 is situated in a different island so maybe it can be different because is less communicated with the other regions.

	Estimate	Std. Error	t value	Pr($>\|t\|$)
(Intercept)	0.4637	0.0147	31.52	0.0000
regionr2relpos	-0.0196	0.0209	-0.94	0.3485
regionr3relpos	0.0177	0.0208	0.85	0.3941
regionr4relpos	0.0777	0.0212	3.66	0.0003
regionr5relpos	-0.0496	0.0208	-2.38	0.0173
regionr6relpos	0.0074	0.0210	0.35	0.7229
regionr7relpos	-0.0366	0.0215	-1.70	0.0890
regionr8relpos	-0.0345	0.0213	-1.62	0.1051

Table 10: T-test results for difference in means in the new model.

Besides testing the means of ratios of positive flocks in the regions it was also a good idea to investigate if there is any difference in the variances in bacteria development throughout all of the 8 regions. Table 4.2.1 shows the results of ANOVA test on the new model.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
region	7	5.80	0.83	7.19	0.0000
Residuals	4079	470.59	0.12		

From the ANOVA test we can see that the p-value for the regions is lower than 0.05, which means that there is a difference in the variances among those regions.

4.2.2 Importance of the "region" factor given the rest of the variables

In this subsection we expand the analysis of regional differences by building up a new linear model with "regions" as a factor with 8 levels and all of the weather variables. Reducing it with *step ()* function, none of the variables was removed, so the structure of the model is:

$$\begin{aligned}\sqrt{ratio} =& \beta_0 + \beta_1 regionFactor + \\ & \beta_9 maxTemp + \beta_{10} relHum + \beta_{11} sunHours + \\ & \beta_{12} precip + \epsilon.\end{aligned} \tag{4}$$

Table 4.2.2 shows the significance of the parameters in the model. As we can see, adding the weather variables, regions 4 and 5 are still those that differ the most, they are significant, so adding the region information can help our model. All of the weather variables are found to be significant.

	Estimate	Std. Error	t value	Pr($>\|t\|$)
(Intercept)	0.0653	0.0901	0.72	0.4687
regionr2relpos	-0.0162	0.0196	-0.83	0.4080
regionr3relpos	0.0277	0.0185	1.50	0.1340
regionr4relpos	0.1078	0.0244	4.42	0.0000
regionr5relpos	-0.0517	0.0190	-2.73	0.0065
regionr6relpos	0.0054	0.0210	0.26	0.7979
regionr7relpos	0.0316	0.0282	1.12	0.2631
regionr8relpos	-0.0267	0.0224	-1.19	0.2336
aveTemp	0.0457	0.0043	10.74	0.0000
maxTemp	-0.0184	0.0040	-4.56	0.0000
relHum	0.0037	0.0009	4.29	0.0000

Table 11: Parameters for the new model with all of the variables.

We can observe the residual of this model in terms of the region. As we can observe that there is still some difference for the region 4 but the residuals are very similar, the model taking into account the region factor is able to adapt to each specific region to some extent.

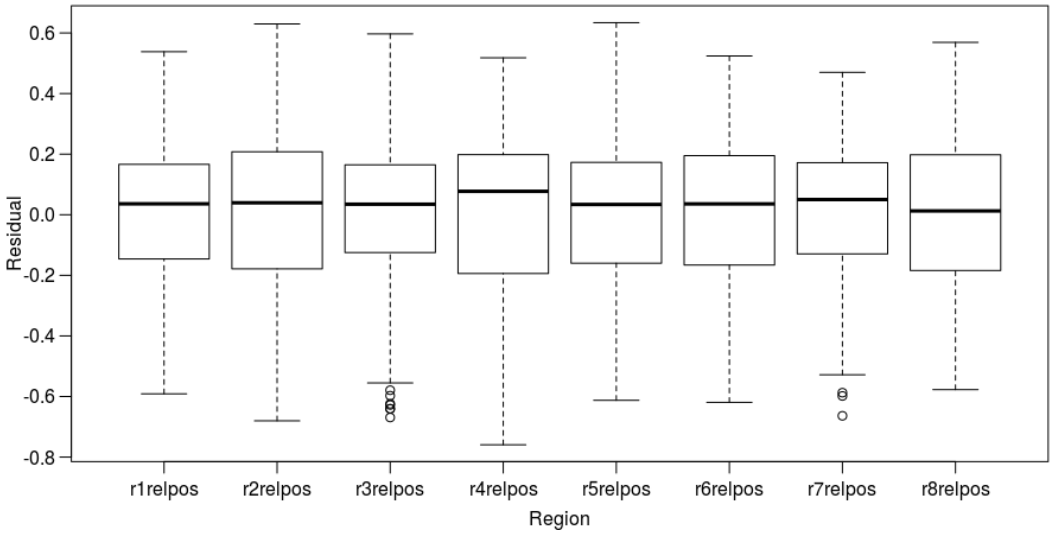

Figure 23: Residual as a function of the Regions.

5 Preprocessing of the data

Before being given the preprocessed data, we were given a more RAW version divided in different files that we needed to preprocess. In Appendix A we attached the code used to perform such preprocessing, and we will give some explanations about the hard parts of it.

In 4) we were asked to convert the dates to a common format, the 2nd and 3rd file were no problem since they had a standard format, but for the first file, where the month were given as abbreviations, we

defined the format to convert, if worked for some months abbreviations, but not for all. At some point we did not want to spend more time into fixing this that seems like an internal bug. Then we found out that the error was because internally, the function was using the Spanish abbreviations instead of the English one, some of them matched, some did not.

In 5) we renamed all the relevant columns from the different files to a common format and then define the common columns we would use in the merging.

In 9) we first changed all the "POSITIVE" to "POS" and "NEGATIVE" to "NEG", then we changed all the empty entries to "NEG" and we finally changed everything that was neither "POS" or "NEG" to "POS" to account for the file where we the type is specified.

6 Evaluative discussion and conclusions

In this report we mainly investigated the risk of developing the Campylobacter for a broiler flock in eight different regions of Denmark. We analysed the relationships between the ratio of positive findings of the Campylobacter and the weather variables (average temperature, max. temperature, relative humidity, sunshine hours and precipitation). From the analysis we created linear models that could learn these relationships.

We started the project from clearing and describing the data. We have noticed that from all of the weather variables, the temperature is the one that has the biggest influence on Campylobacter development in all the analysis and models carried out. Another aspect that we observed was seasonality - during winter period the proportion of positive flocks was much lower than during summer, we did not use the time variable as a explanatory variable and we observed most of the time information can be learn from the weather variables. The next step in our analysis was building up linear models that could capture the relationship between the ratio of positive flocks in the country and the weather variables.

We started by using the Box-Cox transformation which showed that a square root transformation is the one that should be used for the response variable. Thanks to this transformation, the distribution of the data was more Gaussian, but still not satisfying. Therefore, in a further analysis we fitted a Generalized Additive Model that transformed the explanatory variables by a smooth function causing maximization of the linear relation between the response and the transformed explanatory response. This was our upper bound model and it showed that the most important variables are temperature and relative humidity. The residual of this model met the Gaussian and independence assumptions.

Then we started to build linear models from the raw input explanatory weather variables. The p-values of the parameters for this initial new were checked to show the constribution (significance) of each variable. They showed that, as opposed to sunshine hours and precipitation, the temperatures and humidity are significant, therefore our further analysis focused especially on them.

In order to give more expressivity to our model. The quadratic expansion and split transformation of the temperatures was carried out. By combining those two transformations and removing the insignificant variables from the model we received our final model. Once again, the temperatures and the relative humidity were the most significant variables. Plotting the residuals in terms of variables we could observe that they finally indicate high Gaussianity, which means that the model fulfill the assumptions (independence's and Gaussianity of the residuals). We also made prediction plots which showed that the model fits the data best for the average temperature around 9.

In the final section we carried out an analysis of regional differences. By using the mosaic plot we have noticed that region 4 and 5 vary from the rest. Thanks to boxplots of the ratio for every region we observed that mean for region 4 is higher (40% of flocks were infected) in comparison to the others (20-30%) - development of bacteria there is increased. Subsequently, by using t-test and ANOVA test we also checked that there are some significant differences in means and variances between the regions (especially for region 4, and 5). At the end of our report we built up a new model with regions as a factor with 8 levels. Testing of this model confirmed previous observations that region 4 and 5 are the ones that differ the most. No geographical correlation between the regions was found.

A R code - preprocessing

```
#Files
# setwd("")

# We load the files
Camlob1 <-read.delim("campy_pre2002.txt")
Camlob2 <-read.csv("campy_2002-2005.csv", sep=",")
Camlob3 <-read.csv("campy_2005-.csv", sep=",")

# 1) pre2002: Remove those with SEKTION=="res"
Camlob1<-Camlob1[!(Camlob1$SEKTION == "res"),]
# 2) pre2002: Only keep those with AKTVNR==5133
Camlob1<-Camlob1[(Camlob1$AKTVNR == 5133),]
# 3) All files: Valid CHR numbers are 10000 and above
Camlob1<-Camlob1[(Camlob1$CHR_NR >= 10000),]
Camlob2<-Camlob2[(Camlob2$Chrnr >= 10000),]
Camlob3<-Camlob3[(Camlob3$Chrnr >= 10000),]

#4) Convert dates to common format.
# Camlob1$PRV_DATO <- as.Date(Camlob1$PRV_DATO)

# The first format transformation only works for some months !?!?!?
Camlob1$PRV_DATO <- as.Date(strptime(Camlob1$PRV_DATO, format = "%d%b%Y:%H:%M:%S"))
Camlob2$Prvdato <- as.Date(Camlob2$Prvdato, format = "%m/%d/%Y")
Camlob3$Provedato <- as.Date(Camlob3$Provedato, format = "%m/%d/%Y")

# as.Date(Camlob1$PRV_DATO[4])
# as.Date(Camlob3$Provedato[4])
# as.Date(Camlob2$Prvdato[3])
# typeof(Camlob1$PRV_DATO[3])
# typeof(Camlob2$Prvdato[3])
# typeof(Camlob3$Provedato[4])

# 5)get same order of columns to keep and then rename.
## We will change numbers to lower case in 1 and put the names of 3.
colnames(Camlob1)[which(names(Camlob1) == "JNR")] <- "Jnr"
colnames(Camlob1)[which(names(Camlob1) == "DYRNR")] <- "Dyrnr"
colnames(Camlob1)[which(names(Camlob1) == "MATR")] <- "Materialeart"
colnames(Camlob1)[which(names(Camlob1) == "EPINR")] <- "Epinr"
colnames(Camlob1)[which(names(Camlob1) == "PRV_DATO")] <- "Provedato"
colnames(Camlob1)[which(names(Camlob1) == "MATR")] <- "Materialeart"
colnames(Camlob1)[which(names(Camlob1) == "CHR_NR")] <- "Chrnr"
## "BAKTFUND" Blank if no campy and subspecies if positive
colnames(Camlob1)[which(names(Camlob1) == "BAKTFUND")] <- "Resultat"

colnames(Camlob2)[which(names(Camlob2) == "Epi.nr")] <- "Epinr"
colnames(Camlob2)[which(names(Camlob2) == "Prvdato")] <- "Provedato"

colnames(Camlob3)[which(names(Camlob3) == "Tolkning")] <- "Resultat"
## So the common variables are:
common_columns = c("Jnr","Chrnr", "Epinr", "Provedato", "Materialeart", "Resultat")

# 6) Some tests are recorded in two files with different JNR!?!
# (Due to transitions between databases ...)
# (This step can be skipped at first and handled if time permits)
# 7) Merge the data using "rbind"
# rbind will only join if both dataframes have the same columns (not ordered necesarily)
# merge would do it better though
Camlob1 <- Camlob1[common_columns]
Camlob2 <- Camlob2[common_columns]
```

```r
Camlob3 <- Camlob3[common_columns]

## Join the data sets
Camlob <- rbind(Camlob1,Camlob2)
Camlob <- rbind(Camlob,Camlob3)
head(Camlob)

# 8) Remove records with chrnr<=10000 and those with NA as epinr.
# Hint: Use "!is.na(epinr)"
Camlob <- Camlob[!(Camlob$Chrnr <= 10000),]
Camlob <- Camlob[!is.na(Camlob$Epinr),]

# 9) Reduce the levels of resultat to only "POS" or "NEG"
# We have to transform the POSITIV to POS, and NEGATIV to NEG
Camlob[(Camlob$Resultat == "NEGATIV"),]$Resultat <- "NEG"
Camlob[(Camlob$Resultat == "POSITIV"),]$Resultat <- "POS"
## We fill the empty with NEG and the rest with POC
Camlob[(Camlob$Resultat == ""),]$Resultat <- "NEG"
Camlob[(Camlob$Resultat != "NEG" & Camlob$Resultat != "POS"),]$Resultat <- "POS"
Camlob$Resultat <- factor(Camlob$Resultat)  # This way we remove the others

# 10) Remove records with duplicated jnr (Keep first record)
# Hint: Use "duplicated"
Camlob <-Camlob[!(duplicated(Camlob$Jnr)),]

# 11) Only keep records with "matr" in c("Kloaksvaber","Svaberprøve","766","772")
Camlob = Camlob[(Camlob$Materialeart == "766" | (Camlob$Materialeart == "772") |
                   (Camlob$Materialeart == "Kloaksvaber")|(Camlob$Materialeart == "Svaberprøve")),

# 12) Add week number since week one 1998 for each record
# Camlob$Provedato
weeks = as.integer(strftime(Camlob$Provedato,format="%W"))
years = as.integer(strftime(Camlob$Provedato,format="%Y"))
Camlob$weeks <- weeks
Camlob$years <- years

unique(years)
unique(weeks)
# 13) Only keep those with positive week number
Camlob <-Camlob[Camlob$weeks >= 0,]
# 14) Some chrnr should be removed due to various reasons.
# Skip!

# 15) It may be decided only to include data from farms that
# have delivered more than 10 flocks, as those with less may have
# a bias. This could also be included in the analysis ...

# 16) Use "split" to split the data by week
splitted_Camlob = split(Camlob, list(Camlob$years, Camlob$weeks ))
splitted_Camlob = split(Camlob, list(years))
splitted_Camlob = split(splitted_Camlob, list(weeks))
sapply(Camlob, function(x) apply(x, Resultat, mean))

get_ratio <- function(x){
  Total = length(x)
  Npos = sum(x == "POS")
  ratio = as.double(Npos)/Total
}

jk = by(Camlob$Resultat, list(years,weeks), get_ratio)
plot(jk)
```

```
splitted_Camlob = split(Camlob, weeks)
# 17) Summarize number of flocks slaughtered and number of positive flocks per week.
summary(splitted_Camlob$'32'$Resultat == "POS")

## Apply summary to all weeks.
tapply(Camlob$Resultat, Camlob$weeks, summary)
# 18) Save your data file!
write.csv(Camlob, file = "Camlob.csv")

# Get the ratio and do some plotting !!
split_names = names(splitted_Camlob)

ratio = c()
for (sn in split_names){
  Total = length(splitted_Camlob[[sn]]$Resultat)
  print(Total)
  Npos = sum(splitted_Camlob[[sn]]$Resultat == "POS")
  print(Npos)
  print(as.double(Npos)/Total)
  ratio = c(ratio, as.double(Npos)/Total)
}

plot(Camlob$Resultat)
plot(ratio)
```

B R code - statistical analysis

```
#Files
library(xtable)
library(MASS)  #atach package
library(smooth)
cam <- read.delim("case2regionsOnePerBatch.txt")

# my.dataframe[ , "new.col"] <- a.vector
#cam<-cam[,-(8:25)]
cam$rp<-(Camlob$pos/Camlob$total);
cam$rp1<-(Camlob$R1pos/Camlob$R1total);
cam$rp2<-(Camlob$R2pos/Camlob$R2total)
cam$rp3<-(Camlob$R3pos/Camlob$R3total);
cam$rp4<-(Camlob$R4pos/Camlob$R4total);
cam$rp5<-(Camlob$R5pos/Camlob$R5total)
cam$rp6<-(Camlob$R6pos/Camlob$R6total);
cam$rp7<-(Camlob$R7pos/Camlob$R7total);
cam$rp8<-(Camlob$R8pos/Camlob$R8total)

## Just exploratory analysis of how to use time here.
cam$sma_aveTemp2 = sma(cam$aveTemp, h = 2)$fitted
cam$sma_aveTemp3 = sma(cam$aveTemp, h = 3)$fitted
cam$sma_aveTemp4 = sma(cam$aveTemp, h = 4)$fitted

## Removing the nan variables
cam <- cam[!is.na(cam$rp),]
cam[cam$rp == 0]
## Adding 0.00001 just in case, so that we do not have 0 values and we can run
# the bowplot thingy
cam$rp <- cam$rp + 0.00001
cam$date <- as.Date(paste("1", cam$week, cam$year, sep = "-"), format = "%w-%W-%Y")

 #Strength
```

```
str(cam)
#Summary of Weather data
summary(Camlob[,3:7])

#Number of slaughtered floks
colSums(Camlob[8])
##############################################
## Box Cox transform
par(mfrow=c(2,1),mgp=c(2,0.7,0),mar=c(3,3,1,1))

boxcox(rp ~ aveTemp + maxTemp+ relHum+sunHours+precip,data=cam, plotit=TRUE)
#Explore a transformation on the response
cam$rp_trans <-sqrt(cam$rp)
boxcox(rp_trans ~ aveTemp + maxTemp+ relHum+sunHours+precip,data=cam, plotit=TRUE)
#Explore a transformation on the response
# Could we transform the variables using the time information ?
## Histogram
hist(cam$rp)
hist(cam$rp_trans)

## Some little trial of time analysis
#Graphical description
library(car)#relHum+sunHours+precip,
scatterplotMatrix( ~ rp_trans + aveTemp + sma_aveTemp2 + sma_aveTemp3 + sma_aveTemp4,
                  diagonal= "boxplot", data = cam)
#ratio seems to be more depending on temperature and not so much on humidity and sunshine

#time series
par(mfrow=c(3,2), mgp = c(2,0.7,0), mar = c(3,3,1,1))
plot(cam$date,cam$relpos,type="l");plot(cam$date,cam$aveTemp,type="l");
plot(cam$date,cam$maxTemp,type="l");
plot(cam$date,cam$relHum,type="l")
plot(cam$date,cam$sunHours,type="l");plot(cam$date,cam$precip,type="l")

#We remove the 0 values
cam$relHum[which(cam$relHum==0)]<-NA
cam$sunHours[which(cam$sunHours==0)]<-NA

common_columns = c("year","week","rp","rp_trans", "aveTemp", "maxTemp",
                   "relHum", "sunHours","precip")
## Outliers removing
cam <- cam[-c(488),]
cam <- cam[-c(437),]
cam <- cam[-c(436),]
cam <- cam[-c(382),]
cam <- cam[-c(331),]

cam <- cam[,common_columns]
cam<-na.omit(cam)  # Omit nas !! Some func do not work with them

#Making a Generalized Additive Model
library(nlme)
library(mgcv)
par(mfrow = c(3,2), mgp = c(2,0.7,0), mar = c(3,3,1,1))

cam$maxTemprelHum =   cam$maxTemp* cam$relHum
gam_model = gam(rp_trans ~ s(aveTemp) + s(maxTemp)+ s(relHum)+
                  s(sunHours)+s(precip), data = cam)
res = mean(residuals(gam_model)^2)
plot(gam_model)
caca = summary(gam_model)
```

```
 xtable(summary(gam_model)$p.table)
 xtable(summary(gam_model)$s.table)
 par(mfrow = c(3,2), mgp = c(2,0.7,0), mar = c(3,3,1,1))
 plot(gam(rp_trans ~ s(aveTemp) + s(maxTemp)+ s(relHum)+
            s(sunHours)+s(precip), data = cam))

 par(mfrow=c(1,3))
 plot(gam_model$fitted.values, gam_model$residuals)
 plot(gam_model$fitted.values, gam_model$residuals/sd(gam_model$residuals))
 qqnorm(gam_model$residuals)

 par(mfrow=c(2, 3))

 gam_model$residuals <- gam_model$residuals/sd(gam_model$residuals)
 plot(gam_model$residuals~cam$aveTemp, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(gam_model$residuals~cam$maxTemp, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(gam_model$residuals~cam$relHum, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(gam_model$residuals~cam$sunHours, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(gam_model$residuals~cam$precip, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(gam_model$residuals~cam$week, data = cam, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)


 #Average temperature shows Piecewise linear
 # max temp shows some polynomial trend and .
 # realative humidity is almost linear
 #We try to describe the data with a piecewise linear

 #Making a model Linear Model with all interactions with removed NA weeks
 #cam<-cam[,1:8]
 #cam<-na.omit(cam)


 ######################### MODELS  ##############################

 ### Adding extra variables !

 ### Squaring variables
 cam$aveTemp_2 <-  cam$aveTemp^2
 cam$maxTemp_2 <-  cam$maxTemp^2
 Model_init<-lm(rp_trans ~aveTemp + maxTemp+ relHum+sunHours+precip +
                  aveTemp_2 + maxTemp_2,data=cam)
 #Test QQ and uniform Variance, CHECK!

 plot(Model_init,which=1:4)

 summary(Model_init)
 xtable(summary(Model_init))

 null <- lm(rp_trans~1, data = cam)
 Model_initRed<-step(Model_init, scope=list(lower=null, upper=Model_init),
                     direction="both")

 par(mfrow=c(2,2))
 plot(Model_initRed,which=1:4)
 summary(Model_initRed)
```

```
 xtable(summary(Model_initRed))

 #Testing for linearity, Not so nessecary when we did the MAR

 par(mfrow=c(2, 3))
 plot(rstandard(Model_initRed)~aveTemp, data = cam, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(rstandard(Model_initRed)~maxTemp, data = cam, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(rstandard(Model_initRed)~relHum, data = cam, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(rstandard(Model_initRed)~sunHours, data = cam, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)
 plot(rstandard(Model_initRed)~precip, data = cam, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)

 #Test for Independent Residuals, check!
 # par(mfrow=c(1,1))
 plot(rstandard(Model_initRed)~week, data = cam, ylim = c(-3,3), ylab="Standardized ...
residuals"); abline(h = 0); abline(h=c(-1, 1) * qt(.975, df = 419), lty = 2)

##### PIECEWISE LINEAR MODEL !!!!

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

 ## This was not the optimal split point So finding it:
 #optimize optimizes the function (finds the minimum)
 #for values of zz (the split point) between 3 and 8
 optim<-optimize(function(aveTemp_sp){
   model = lm(rp_trans ~ aveTemp + pwl(aveTemp,aveTemp_sp) +
                maxTemp +  # pwl(maxTemp, maxTemp_sp) + #+   maxTemp +
                # aveTemp_2 + maxTemp_2 +
                relHum+sunHours+precip, data = cam)
   null <- lm(rp_trans~1, data = cam)
   model_red<-step(model, scope=list(lower=null, upper=model), direction="both")
   sum( residuals(model_red)^2 )
 },c(3,8))
 (x0.opt<-optim$minimum)


 aveTemp_pwl_value = 7.785
 maxTemp_pwl_value = 8.00
 cam$aveTemp_pwl<-(cam$aveTemp > aveTemp_pwl_value)*(cam$aveTemp-aveTemp_pwl_value)
 cam$maxTemp_pwl<-(cam$maxTemp > maxTemp_pwl_value)*(cam$maxTemp-maxTemp_pwl_value)

 cam$aveTemp_pwl_2<- cam$aveTemp_pwl^2
 cam$maxTemp_pwl_2<- cam$maxTemp_pwl ^2

 Model_init = lm(rp_trans ~  aveTemp + aveTemp_pwl  +
                      maxTemp + maxTemp_pwl + #    maxTemp +
               relHum + sunHours + precip +
                   aveTemp_2 + maxTemp_2 + aveTemp_pwl_2 + maxTemp_pwl_2
               # aveTemp*aveTemp_pwl + relHum*aveTemp_pwl + precip*aveTemp_pwl
               ,data = cam)
 null <- lm(rp_trans~1, data = cam)
```

```
 Model_initRed<-step(Model_init, scope=list(lower=null, upper=Model_init),
                     direction="both")

# drop1(Model_init)
#  Model_initRed <- Model_init
#  drop1(Model_init)
#  drop1(Model_init)
#  drop1(Model_init)
 summary(Model_init)

 par(mfrow=c(2,2))
 plot(Model_initRed,which=1:4)
 summary(Model_initRed)
 xtable(summary(Model_initRed))

 #Plotting for means model
 # Prediciton in terms od the aveTemp
 par(mfrow=c(2,3))

 ###################################  1  ###################################
 aveTemp_seg = seq(from=min(cam$aveTemp), to=max(cam$aveTemp), length.out=500)
 maxTemp_seg =seq(from=min(cam$maxTemp), to=max(cam$maxTemp), length.out=500)

 ave_Temps = c(4,6,8,9,10,12)
 for (ave_i in ave_Temps){
   newData <- data.frame(
         #aveTemp=aveTemp_seg
         maxTemp=maxTemp_seg,
       # maxTemp=mean(cam$maxTemp),
        aveTemp=ave_i,precip=mean(cam$precip),
       relHum=mean(cam$relHum), sunHours=mean(cam$sunHours))

   newData$maxTemp_pwl <-  (mean(newData$maxTemp) > maxTemp_pwl_value)*
     (mean(newData$maxTemp)-maxTemp_pwl_value)
   newData$aveTemp_pwl <-  (mean(newData$aveTemp) > aveTemp_pwl_value)*
     (mean(newData$aveTemp)-aveTemp_pwl_value)
   newData$maxTemp_pwl_2 <-  newData$maxTemp_pwl^2
   newData$aveTemp_pwl_2 <-  newData$aveTemp_pwl^2
   newData$maxTemp_2= newData$maxTemp^2
   newData$aveTemp_2 = newData$aveTemp^2

   Pred.ci <- predict(Model_initRed, newdata=newData, interval="confidence",level=.95)
   Pred.pi <- predict(Model_initRed, newdata=newData,interval="prediction",level=.95)

   #Plot Data and CI and PI
   subset = cam
   subset = cam[cam$aveTemp < ave_i +1,]
   subset = subset[subset$aveTemp > ave_i - 1,]

   ## We plot the originial rp and the square of the predictions
   plot(rp ~ maxTemp, data = subset, pch = 20, las = 1,ylim=c(0,1),
        main = paste("aveTemp: ", as.character(ave_i), sep = "") )
   matlines(newData$maxTemp, Pred.ci^2, lty=c(1,2,2), col=c(1,3,3))
   matlines(newData$maxTemp, Pred.pi^2, lty=c(1,2,2), col=c(1,2,2))
 }

 # As we can see, around the average we have the best predictions
 ########################################################################
 #Looking at regions ratio distribution

#Open file
```

```
Campylobacter <- read.delim("case2regionsOnePerBatch.txt")
#Ratios

cam<-Campylobacter;cam<-cam[,-(8:25)]
cam$relpos<-(Campylobacter$pos/Campylobacter$total);
cam$r1relpos<-(Campylobacter$R1pos/Campylobacter$R1total);
cam$r2relpos<-(Campylobacter$R2pos/Campylobacter$R2total)
cam$r3relpos<-(Campylobacter$R3pos/Campylobacter$R3total);
cam$r4relpos<-(Campylobacter$R4pos/Campylobacter$R4total);
cam$r5relpos<-(Campylobacter$R5pos/Campylobacter$R5total)
cam$r6relpos<-(Campylobacter$R6pos/Campylobacter$R6total);
cam$r7relpos<-(Campylobacter$R7pos/Campylobacter$R7total);
cam$r8relpos<-(Campylobacter$R8pos/Campylobacter$R8total)
cam$date <- as.Date(paste("1", cam$week, cam$year, sep = "-"), format = "%w-%W-%Y")

#Data transformation
cam$relpos = sqrt(cam$relpos)
camall$relpos = sqrt(camall$relpos)

#Tree model
library(tree)
TREE<-tree(relpos~+aveTemp+maxTemp+relHum+sunHours+precip,data=cam1)
plot(TREE)
text(TREE)
summary(TREE)
#####################################################################
##Regions differences
#New dataset
camf<-cam[,9:17]
camf$year<-cam$year
camf$week<-cam$week

#Make regions a factor with 8 levels
library(reshape)
camf2<-melt(camf, id=c("year","week","date"))
camf2$aveTemp<-cam$aveTemp
camf2$maxTemp<-cam$maxTemp
camf2$relHum<-cam$relHum
camf2$sunHours<-cam$sunHours
camf2$precip<-cam$precip
camf2$relpos<-cam$relpos

names(camf2)<-c("year", "week", "date", "region", "ratio", "aveTemp",
                "maxTemp", "relHum", "sunHours", "precip", "relpos")
camf2$region = as.factor(camf2$region)

camf4<-Campylobacter[,1:2]
camf4$R1total<-Campylobacter$R1total
camf4$R2total<-Campylobacter$R2total
camf4$R3total<-Campylobacter$R3total
camf4$R4total<-Campylobacter$R4total
camf4$R5total<-Campylobacter$R5total
camf4$R6total<-Campylobacter$R6total
camf4$R7total<-Campylobacter$R7total
camf4$R8total<-Campylobacter$R8total
camf4$date<-cam$date

camf3<-melt(camf4, id=c("year", "week", "date"))
names(camf3)<-c("year", "week", "date", "Region", "totvalue")
camf3$aveTemp <-camf2$aveTemp
camf3$maxTemp <-camf2$maxTemp
```

```
camf3$relHum <-camf2$relHum
camf3$sunHours <-camf2$sunHours
camf3$precip <-camf2$precip
camf3$region<-camf2$region
camf3$ratio <-camf2$ratio


## Omit the Nans and also the measurements with less than 5 flocks
# camf22<-na.omit(camf3)
camf22 <- camf22[camf22$totvalue > 5,]

#Linear model with only 1 variable
model<-lm(sqrt(ratio)~region,data=camf2)
library(xtable)
xtable(summary(model))

## Boxplot
par(mfrow = c(1,1), mgp = c(2,0.7,0), mar = c(3,3,1,1))
camf22$rp_trans <- sqrt(camf22$ratio)
boxplot(data=camf22, rp_trans~region, xlab = 'Region',ylab = 'sqrt(ratio)', las = 1)

##Testing variances
library(xtable)
xtable(anova(model))

#Linear model with all of the variables
model2<-lm(sqrt(ratio)~region+aveTemp+maxTemp+precip, data=camf22)
library(xtable)
xtable(summary(model2))
model2b <- step(lm(sqrt(ratio)~region+aveTemp+maxTemp+precip, data=camf22))
xtable(summary(model2b))

boxplot(residuals(model2b)~camf22$region, xlab = 'Region',ylab = 'Residual', las = 1)
summary(model2b)

## The final model
camf22$aveTemp_2<- camf22$aveTemp^2
camf22$maxTemp_2<- camf22$maxTemp^2
aveTemp_pwl_value = 7.785
maxTemp_pwl_value = 8.00
camf22$aveTemp_pwl<-(camf22$aveTemp > aveTemp_pwl_value)*
  (camf22$aveTemp-aveTemp_pwl_value)
camf22$maxTemp_pwl<-(camf22$maxTemp > maxTemp_pwl_value)*
  (camf22$maxTemp-maxTemp_pwl_value)

camf22$aveTemp_pwl_2<- camf22$aveTemp_pwl^2
camf22$maxTemp_pwl_2<- camf22$maxTemp_pwl ^2

Model_init = lm(sqrt(ratio) ~  aveTemp + aveTemp_pwl  +
                  maxTemp + maxTemp_pwl + #    maxTemp +
                  relHum  +
                  aveTemp_2 + maxTemp_2 + aveTemp_pwl_2 + maxTemp_pwl_2
                  + region
                # aveTemp*aveTemp_pwl + relHum*aveTemp_pwl + precip*aveTemp_pwl
                ,data = camf22)
null <- lm(sqrt(ratio)~1, data = camf22)
Model_initRed<-step(Model_init, scope=list(lower=null, upper=Model_init), direction="both")
summary(Model_initRed)
```