Lachlan Bassi 22975276

Abstract

This study investigated the temperature dynamics inside a parked vehicle through various statistical models. Using data collected over ten days from a white Ford Focus Sedan, four statistical models were implemented to predict future temperatures: a Linear model, an ARIMA Linear model, a Generalised Least Squares (GLS) Linear model, and a Box-Jenkins ARIMA model. Each model underwent extensive diagnostics to assess its performance, comparing predicted and observed values, and considering metrics such as AIC, BIC, and RMSE values. Among these, the Box-Jenkins ARIMA model emerged as the most efficient and accurate, demonstrating superior prediction capabilities. This modelling approach holds potential in various applications, such as enhancing climate control systems in cars, aiding forensic investigations, and improving public health initiatives. Moreover, this study underlines the potential of integrating these models with others, such as the time-delayed model by Dadour et al. [1], for a comprehensive understanding of temperature dynamics in parked vehicles. Further improvements to this analysis could include incorporating additional variables and utilizing cross-validation techniques for model generation.

Introduction

Understanding the temperature variations inside a parked car holds significant implications in a variety of fields. This study explores this topic through modeling temperature dynamics in the interior of a parked vehicle. This research is based on data described by Dadour et al. [1], which measures temperature changes inside a white Ford Focus Sedan over a ten-day period.

From midnight on March 9 to midnight on March 18, 2008, temperature readings were taken every ten minutes. Accompanying each temperature measurement is the corresponding date and time, allowing for a detailed investigation of thermal behavior within the parked car.

Creating an accurate model of these temperature changes has practical applications across several domains. A key use case is in assessing the potential risks linked with high interior car temperatures. Such conditions can pose significant health threats, especially to children, pets, and vulnerable individuals. Through developing precise temperature models, we can better comprehend the factors that influence temperature fluctuations and devise strategies to reduce associated risks.

Moreover, these temperature models can support the design of energy-efficient climate control systems for vehicles. Understanding how temperature behaves inside a car can aid in the creation of more effective cooling systems, leading to reduced energy consumption and improved comfort for passengers. Accurate modeling of temperature variations can thus enable auto manufacturers and engineers to enhance climate control systems for better performance and sustainability.

In their study, Dadour et al. utilised an ARMA model with a temporal lag of 2 hours, capturing the delayed nature of temperature changes in the vehicle. Their approach demonstrated a strong positive relationship between predicted and actual values, providing valuable insights into the thermal dynamics of the vehicle cabin.

Various researchers have sought to predict parked car temperatures using physical models. One approach considers the cabin air temperature based on energy fluxes from different elements of the car cabin [2]. This model provides a comprehensive framework for understanding temperature changes within a parked car but requires knowledge or estimation of various physical properties and environmental factors.

Modeling parked car temperatures can also aid in forensic investigations and public health initiatives, among others. For instance, one study devised a straightforward 'greenhouse' model to predict daily internal vehicle temperatures, using local meteorological data [3]. This model revealed that black vehicles are generally hotter than white ones on hot summer days. Partially lowering the driver's window can reduce cabin temperatures, but not enough to significantly mitigate safety concerns for children or pets.

In a different approach, a dynamic thermal model focusing on energy management in electric and hybrid vehicles was presented [4]. This model is based on theoretical heat transfer, thermal inertia, and radiation treatment equations. The model proved highly accurate, indicating its potential utility in the design and performance assessment of HVAC systems in electric and hybrid vehicles.

In contrast, this study uses statistical models that lean on observed patterns in the dataset, including linear, ARIMA, and generalised least squares models. These models offer a practical means of predicting future temperatures based on past data. Comparing these approaches can enlighten us on the strengths and weaknesses of each and how they might complement one another for a comprehensive understanding of temperature dynamics in a parked car.

In the subsequent sections, we will dive into the methodology and models utilised to analyse the collected temperature data in the parked car.

Methodology

The methodology for this study can be described in the following steps, from data transformation to model comparison and analysis.

Data Transformation: The original data collected every ten minutes was converted to an hourly measurement scale for a more manageable analysis.

Model 1 - Linear Model: A basic linear model was fitted to the data. I addressed seasonality within the data by applying an Autoregressive (AR) term to the data creating new variables: templag, timelag, hourlag, and templast where the first three where the variables lagged by 2 and the final is the temperature with a lag of 240 periods since there is 240 hours within the data set.

Model Diagnostics for Model 1: Model diagnostics were performed to verify the assumptions of the linear model and to assess its adequacy. These included generating the AIC and BIC values, RMSE value, generating a histogram of the residuals, generating an ACF and PACF plot, and finally plotting the fitted vs observed values.

Model 2 - ARIMA Linear Model: The second model was developed using the ARIMA (Autoregressive Integrated Moving Average) approach, which allowed me to include a time trend and seasonality in the model. Two models were created one using the auto.arima function which automatically generates an ARIMA model and then I created my own using custom values through a process of trial and error.

Model Diagnostics for Model 2: Similar to Model 1, diagnostics were carried out for the ARIMA models to compare performance and determine the ARIMA model of best fit. These diagnostics were the same as mentioned in “Model Diagnostics for Model 1”.

Model 3 - GLS Linear Model: The third model was developed using Generalised Least Squares (GLS). This approach allowed me to account for any heteroscedasticity in the data, and to include an appropriate covariance structure. An AR(2) term was included in this model and the model was tested with multiple covariance structures and correlation structures and from here models were compared using anova tests were the best model was determined by having the lowest AIC and BIC values as well as having the least number of variables.

Model Diagnostics for Model 3: The GLS model was also subject to the same diagnostic checks mentioned for the previous two models.

Model 4 - Box-Jenkins ARIMA Model: Lastly, a Box-Jenkins ARIMA model was fitted to the data, accounting for seasonality within the dataset. Two were created similar to Model 2 were one is generated using the auto.arima function and another is generated using custom values determined through a process of trial and error.

Model Comparison: Finally, the four models were compared in terms of their performance, similarities, differences, and shortcomings using fitted vs observed plots, residuals vs fitted value plots, and comparing AIC and BIC values. The best model was chosen based on this comparison.

Results

Model 1

The initial linear model fitted to the data was of the form temperature = timestamp + hour_label. However, this model left a lot to be desired. Analysing the diagnostic plots generated the first is a plot of the residual's vs fitted values and the range of residuals stretches from -10 to 5 showing strong variance in the residuals, an undesirable trait. It should also be noted that the bottom tail of the qqplot consists of the points falling off below the line and at the top tail the points also trail off below the line implying the residuals do not follow an ideal normal distribution with mean 0 and standard deviation 1.

Also, analysing the fitted vs observed values plot for time and temperature it appears as though there is a lot left to be desired, while the linear model somewhat follows the observed values there is a clear gap between the two lines at all of the points (the peaks in the graph). Finally, it had an AIC value of 1284 and a BIC value of 1374 which are very high and undesirable values.

When analysing the ACF graph it shows every Lag value having a spike greater than the upper threshold and the PACF graph shows a positive spike above the upper threshold at lag 1, indicating a direct correlation, and a negative spike below the lower threshold at lag 2, suggesting an indirect or mediated correlation.

Given these patterns in the ACF and PACF graphs, an ARMA (2) model is selected to improve the original linear model, where both autoregressive (AR) and moving average (MA) terms are included up to lag 2, as it captures the significant autocorrelation observed in the data. This involves creating the following variables templag, timelag, and hourlag which are the respective variables but with lag 2 applied. Also, a new variable is added templast which is lagged by 240 hours or ten days.

By including templast as a lagged variable, there is now the notion that the current temperature can be influenced by its own past values. This can be particularly relevant in temperature data, as there may be a natural persistence or memory effect, where the temperature at a given time is influenced by the temperature shortly before it.

Also, the intercept was removed, as by omitting the intercept, I can interpret the coefficients of the explanatory variables as the incremental effects on the temperature, given the specified hour lag, hour of the day, and previous temperature. This can be useful in understanding the impact of these variables on temperature fluctuations rather than estimating the absolute temperature level.

This resulted in a model that outperformed the previous and became my final model 1. It had the formula:

templag = -0.0008582 * timelag - 0.0889182 * hourlag0 + 0.3289029 * hourlag1 + 0.5215647 * hourlag2 + 0.8357989 * hourlag3 + 1.0041095 * hourlag4 + 0.9064965 * hourlag5 + 0.7364695 * hourlag6 + 0.9612705 * hourlag7 + 6.4318295 * hourlag8 + 14.0855349 * hourlag9 + 18.0539080 * hourlag10 + 19.1654652 * hourlag11

+ 18.0836465 * hourlag12 + 15.0550126 * hourlag13 + 10.4893089 * hourlag14 + 5.9823956 * hourlag15 + 2.6401963 * hourlag16 + 0.3269147 * hourlag17 - 3.4119718 * hourlag18 - 8.8551893 * hourlag19 - 9.2307629 * hourlag20 - 5.0041067 * hourlag21 - 2.1052855 * hourlag22 - 0.7735985 * hourlag23 + 0.8853505 * templast

It's diagnostic plots showed improved performance, it's ACF still showed violations, surpassing the blue line limits, at 8 of the 24 lags but this is still an improvement on the previous model. Also, the fitted vs observed values plot consists of the models fitted values doing a better job at fitting the observed values even if it still has small deviations at the points/peaks on the graph, they aren't as extreme as before.

The PACF consists of spikes that violate the boundaries at (1, 2, 3, 4, 5, 6, and 8) which is not ideal and actually worse than before. However, it does have much improved AIC (928) and BIC (1022) values compared to the previous linear model. Overall while the model has been improved the PACF and ACF graphs indicate that a linear model may not be the best way to fit the data as even after improving the model there is more to be desired.

Model 2

Two models were created and compared for the second model, the first used auto.arima in R to automatically choose the best parameters and then over a long process of trial and error I created another model with custom parameters and compared the performance of the two.

The automatically generated ARIMA model had a lower AIC (682 < 703) but a higher BIC value (769 > 734) when compared to the custom ARIMA model. This indicates the auto generated ARIMA model is a better fit for the data but when considering complexity the BIC value penalises more complex models and implies that the custom ARIMA model is better.

This makes sense when comparing the two models as the custom model has 9 parameters compared to the auto ARIMA model having 25. Other metrics that can be analysed include the ACF chart where it is apparent that the custom ARIMA model does a better job compared to the auto ARIMA model as it contains only one very small violation at lag 6 compared to the auto ARIMA model containing 8 clear violations of the blue boundaries. Violations indicate the presence of significant autocorrelation in the residual errors of the ARIMA model meaning that the custom ARIMA model does a better job at capturing and reducing autocorrelation in the residual errors. This autocorrelation of the residuals is supported by the outcomes of the Ljung-Box test.

The outcomes of the Ljung-Box test are as follows: The custom model yields a p-value of 0.786, exceeding the significance level of 0.05. This indicates that there's no substantial evidence of autocorrelation within the residuals. In contrast, the auto ARIMA model presents a p-value of 2.2e-16, substantially lower than the significance threshold. This implies a pronounced autocorrelation among the residuals of the auto ARIMA model, suggesting that there are undetected trends or patterns in the data.

Some other things to note: the residual plot for the custom ARIMA model shows a more symmetrical normal distribution of the residuals compared to the auto ARIMA model that has a strong tail to the left. The custom ARIMA model has a lower RMSE value (0.828) compared to the auto ARIMA model having an RMSE of (0.900) implying it does a better job at fitting the observed values.

Based on the comparisons I believe the custom ARIMA model provides the best fit and has the equation:

The final model equation is:

temperature[t] = 30.7798 + 1.6677*temperature[t-1] - 0.7244*temperature[t-2] - 0.0582*e[t-1] - 0.1774*e[t-2] + 0.9999*temperature[t-24] - 0.9708*e[t-24] - 0.0084*t + e[t]

Where:

temperature[t] is the predicted temperature at time t.

temperature[t-1] and temperature[t-2] are the temperatures at times t-1 and t-2, respectively.

e[t-1] and e[t-2] are the errors at times t-1 and t-2, respectively.

temperature[t-24] is the temperature 24 time periods prior (given the seasonal component).

e[t-24] is the error term 24 time periods prior.

t is the time index (corresponding to xreg in the model).

e[t] is the error term at time t.

This model equation incorporates both autoregressive (AR) and moving average (MA) aspects, along with seasonality and a time trend. It provides a comprehensive representation of the factors influencing the prediction of cabin temperature at any given time point.

<u>Model 3</u>

Model 3: The process involved fitting a linear model to the data using the 'gls' command in R. Initially, the model was created without any additional terms represented by:

model3 <- gls(temperature ~ timestamp + hour_label, data = cabin_new)

However, after analysing the ACF chart which consisted of every lag spike violating the upper limit along with the extremely high AIC value of (1,249) and extremely high BIC value of (1,337) it was determined that the model needed to be altered.

Implementing AR(2) again by using the lagged variables from Model 1 the result was:

model3.lag <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new)

This model had much lower AIC (935) and BIC (1,025) values and the ACF plot now only had 9 violations compared to the previous 24.

To further improve this model three covariance structures were fitted to the model, varIdent(form = ~1 | hourlag), varExp(form = ~ timelag), and varPower(form = ~ timelag). Using anova tests to compare the covariance structures and then selecting the model with the lowest AIC and BIC values while also taking into account the models complexity in terms of the number of parameters, the best performing was varPower(form = ~ timelag).

From here the same process was used to compare four correlation structures once again using anova tests. The best performing was cor = corARMA(p = 1, q = 1). However, a problem with this model was it's incredibly high RMSE of 3.01 and because of this the second best fitting model was selected which had the correlation structure of, corARMA(p = 2, form = ~ 1 | hourlag).

templag = -0.2398002 * timelag - 0.2307011 * hourlag0 - 0.444 * hourlag1 - 0.447 * hourlag2 - 0.451 * hourlag3 - 0.453 * hourlag4 - 0.455 * hourlag5 - 0.457 * hourlag6 - 0.459 * hourlag7 - 0.461 * hourlag8 - 0.461 * hourlag9 - 0.455 * hourlag10 - 0.447 * hourlag11 - 0.441 * hourlag12 - 0.438 * hourlag13 - 0.436 * hourlag14 - 0.436 * hourlag15 - 0.438 * hourlag16 - 0.440 * hourlag17 - 0.443 * hourlag18 - 0.449 * hourlag19 - 0.459 * hourlag20 - 0.467 * hourlag21 - 0.473 * hourlag22 + 0.392 * templast - 0.891 * horlg0 - 0.869 * horlg1 - 0.902 * horlg2 - 0.887 * horlg3 - 0.875 * horlg4 - 0.866 * horlg5 - 0.859 * horlg6 - 0.849 * horlg7 - 0.838 * horlg8 - 0.831 * horlg9 - 0.892 * horlg10 - 0.936 * horlg11 - 0.959 * horlg12 - 0.971 * horlg13 - 0.977 * horlg14 - 0.980 * horlg15 - 0.980 * horlg16 - 0.979 * horlg17 - 0.977 * horlg18 - 0.973 * horlg19 - 0.966 * horlg20 - 0.947 * horlg21 - 0.926 * horlg22 - 0.907 * horlg23 + e

Where:

templag represents the predicted temperature at time t,

timelag is the lagged variable of timelag,

hourlag0 to hourlag23 represent the lagged variables of hourlag at different time lags,

templast is the lagged variable of templast,

horlg0 to horlg23 are the lagged variables of horlg at different time lags,

e represents the error term.

This model has AIC and BIC values of 899 and 1003. The range of residuals is from -6.6 to 2.6 with the majority falling between -0.43 and 0.55 which implies mostly a normal structure but with a skew left. It has an RMSE value of 1.52, which is an undesirable value indicating the model does a poor job at fitting against the observed values.

Analysing the diagnostic plots the PACF plot shows 7 which violate the blue boundaries. Also, the ACF plot has 9 violations of the blue lines which is not ideal indicating there is autocorrelation between residuals. The fitted vs observed plots show that it struggles fitting the observed values especially in the middle at point 100 which is in the middle of the data analysis, day 5 indicating maybe something changed in the data collection.

Based on this analysis the best performing gls model according to the metrics of AIC and BIC failed to provide a good fit with an RMSE of 3.01, because of this the second best model from the analysis was used which had an RMSE of 1.52. This model still struggled fitting the data accurately and this implies that a gls model is probably not the best model to apply to the data. Also, it indicates that selecting an ideal model for time series data simply based on AIC and BIC will not always produce an ideal model.

Model 4

Finally a fourth model, was implemented via auto.arima. This model treated temperature as a time series variable with a frequency of 24 to match the 24 daily observations. Furthermore, the model was fitted with the 'seasons' parameter set to TRUE, indicating that the function would find the optimal ARIMA model while considering a seasonal component with a period of 24.

We also created another model using custom parameters. However, this time, the auto-generated model exhibited lower AIC (638 versus 701) and BIC (658 versus 725) values compared to the custom ARIMA model, unlike in model 2 where the opposite was the case.

The auto ARIMA model displays a noticeable spike on its ACF graph at lag 48, and its residual histogram shows a slight leftward skew. On the other hand, the custom model's residual graph is more centrally located, indicating less evidence of autocorrelated residuals. This is supported by the Ljung-Box test, where the auto ARIMA model failed with a p-value of 0.0112 (less than 0.05), compared to the custom model, which had a substantially higher p-value of 0.731. This larger p-value, along with the fact that most points are contained within the ACF plot, and the few points that breach the boundary are not extreme, all indicate that the residuals of the custom model are not significantly autocorrelated and that the residuals in the auto ARIMA model are.

Also, the custom ARIMA model had a lower RMSE value of 0.812 compared to the auto ARIMA model which had a RMSE value of 0.952 indicating the custom ARIMA did a better job at fitting the data. Because of this the custom ARIMA model was selected and the model equation is:

temperature[t] = 0.8855 * temperature[t-1] + 0.6867 * error[t-1] + 0.2871 * error[t-2] - 0.4645 * seasonal_error[t-24] + 0.0100 * drift[t] + error[t]

temperature[t] represents the predicted temperature at time t.

temperature[t-1] is the temperature at the previous time point.

error[t-1] and error[t-2] are the residual errors at time t-1 and t-2, respectively.

seasonal_error[t-24] is the seasonal residual error at time t-24.

drift[t] represents the drift component at time t.

error[t] is the residual error at time t.

Comparing 4 Models

When comparing the four proposed models, it becomes apparent that Model 4, utilising the Box-Jenkins ARIMA method, outperforms the others in terms of the evaluation criteria:

Model 4 boasts the lowest AIC value of 701, along with the smallest parameter count of 7, effectively making it the most efficient and simplest model. It also maintains the smallest BIC value, which stands at 725.3.

Both model 4 and model 2 have a similar range of residuals with most points in both models falling between (-4, 4) and this outperforms the other two models significantly which both have ranges of (2, -10).

Model 4 also boasts the smallest RMSE value of the four of 0.812 although it is important to note that Model 2's RMSE of 0.828 is closely behind but overall model 4 has the best fit against the observed values.

When evaluating the plots of fitted versus observed values, Models 2 and 4 outshine the others significantly. However, upon closer inspection, Model 4 appears to perform marginally better than Model 2. This is primarily observable at the very first dip in the lower left corner of the plot, where Model 4 fits the dip with superior accuracy.

In conclusion, considering all the above evaluation metrics, Model 4 emerges as the most competent predictor of temperature in our analysis.

Discussion

When comparing all four models, it is clear that each one has its strengths and weaknesses.

Model 1 incorporates lagged variables and an ARMA structure, which enables it to capture more temporal patterns and autocorrelation in the data than the original linear model. However, its ACF and PACF plots still indicate significant autocorrelation and indirect correlation, and its AIC and BIC values are relatively high compared to the other models.

Model 2, especially the custom ARIMA model, exhibits improved performance over Model 1 in terms of reducing autocorrelation in the residuals and fitting the observed values. This is supported by lower AIC and BIC values, as well as fewer violations of the ACF chart boundaries. Its main drawback is its complexity, with more parameters than the other models, which could make it more prone to overfitting.

Model 3, despite its advanced structure with covariance and correlation components, still fails to capture all the variance in the residuals, and its diagnostic plots indicate that there are still patterns in the data that the model isn't capturing. However, it has improved AIC and BIC values compared to Model 1, indicating a better fit.

Model 4, the Box-Jenkins ARIMA model, emerges as the best performer among the four in terms of AIC, BIC, and RMSE values, indicating it provides the best fit for the data. Its residual histogram and ACF graph suggest that it effectively captures the autocorrelation structure in the data. Consequently, Model 4 establishes itself as the superior model in this analysis.

Comparison to Dadour et al.

While the best model in this report, Model 4, uses a Box-Jenkins ARIMA structure, Dadour et al. use a structure that incorporates time delay in their modelling approach. These structures both aim to capture the temporal dynamics of the data, but they do so in different ways.

Model 4's ARIMA structure selects ideal AR, I, and MA parameters based on the characteristics of the data. This includes autocorrelation, seasonality, and trends, with the goal of providing the best fit while taking into account model complexity. It appears to have done well in this regard, as indicated by its low AIC and BIC values, and its diagnostic plots showing a good capture of autocorrelation structure.

Dadour et al.'s time delay structure, on the other hand, directly incorporates the delay between the time a temperature change occurs and the time it's detected. This structure captures the temporal lag in temperature changes, a characteristic that seems particularly relevant for vehicle cabin temperatures. Their results show a strong positive relationship between predicted and actual values, suggesting effective capturing of the patterns in their data.

In comparison to the models of this report, Dadour et al.'s time delay structure seems more similar to the structure used in Model 3, which also incorporates correlation components. However, Model 3 fails to capture all the variance in the residuals, suggesting that it may not be as effective in capturing the time-delayed nature of the data as Dadour et al.'s model.

Despite the differences in the structures used, both Model 4 in this report and the time delay model used by Dadour et al. provide valuable insights into the temporal dynamics of vehicle cabin temperatures. These models could potentially be combined or compared in future research to provide even more robust predictions. It would also be beneficial to see a direct comparison if Dadour et al. were to calculate the AIC and BIC values for their model.

Model 4 suggests that the temperature in the cabin of the car is influenced by recent temperatures (both the last time point and the same time on the previous day), recent errors in the model, any slow linear trend over time, and random fluctuations. It captures both the short-term and long-term (seasonal) dependencies in the data, as well as correcting for its own previous mistakes.

Lachlan Bassi 22975276

This analysis can be improved in a few ways. Firstly, the inclusion of more variables such as solar radiation, vehicle colour and ventilation status which are all considered in [1]. Also, the use of cross-validation can improve the analysis as by parting the data into a test and training set a model can be generated to prevent overfitting, to do this a larger data set may be required.

References:

[1] Dadour, I. R. and Almanjahie, I. and Fowkes, N. D. and Keady, G. and Vijayan, K. 2011. Temperature variations in a parked vehicle. Forensic Science International. 207 (1-3): pp. 205-211.

[2] Horak, J., Schmerold, I., Wimmer, K. et al. Cabin air temperature of parked vehicles in summer conditions: life-threatening environment for children and pets calculated by a dynamic model. Theor Appl Climatol 130, 107–118 (2017). https://doi.org/10.1007/s00704-016-1861-3

[3] Dadour IR, Almanjahie I, Fowkes ND, Keady G, Vijayan K. Temperature variations in a parked vehicle. Forensic science international. 2011 Apr 15;207(1-3):205-11.

[4] Marcos D, Pino FJ, Bordons C, Guerra JJ. The development and validation of a thermal model for the cabin of a vehicle. Applied Thermal Engineering. 2014 May 1;66(1-2):646-56.

examADA

2023-05-24

```
library(dplyr)
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
library(lubridate)
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
library(stringr)
library(forecast)
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
library(Metrics)
```

```
##
## Attaching package: 'Metrics'
## The following object is masked from 'package:forecast':
##
##     accuracy
library(nlme)
##
## Attaching package: 'nlme'
## The following object is masked from 'package:forecast':
##
##     getResponse
## The following object is masked from 'package:dplyr':
##
##     collapse
# Read the data from the text file
cabin <- read.table("cabin.txt", sep = "\t")

# Check the structure of the data frame
str(cabin)
## 'data.frame':    1441 obs. of  2 variables:
##  $ V1: chr  "09/03/2008 00:01" "09/03/2008 00:11" "09/03/2008 00:21" "09/03/2008
00:31" ...
##  $ V2: chr  "16.631 °C" "16.384 °C" "16.140 °C" "15.893 °C" ...
# Rename the column headers
names(cabin)[names(cabin) == "V1"] <- "timestamp"
names(cabin)[names(cabin) == "V2"] <- "temperature"
```

First convert the data to temperature measured hourly.

```
# Creating an index with a step of 6
index_seq <- seq(1, nrow(cabin), by = 6)
cabin_new <- cabin[index_seq, ]

# Converting temperature to numeric and trimming to first 4 characters
cabin_new$temperature <- as.numeric(substr(cabin_new$temperature, 1, 4))

# Creating a numeric sequence for time
cabin_new$timestamp <- 1:nrow(cabin_new)

# Generating factor variable for hours
cabin_new$hour_label <- as.factor(rep(0:23, each = 1, length.out = nrow(cabin_new))
)
```

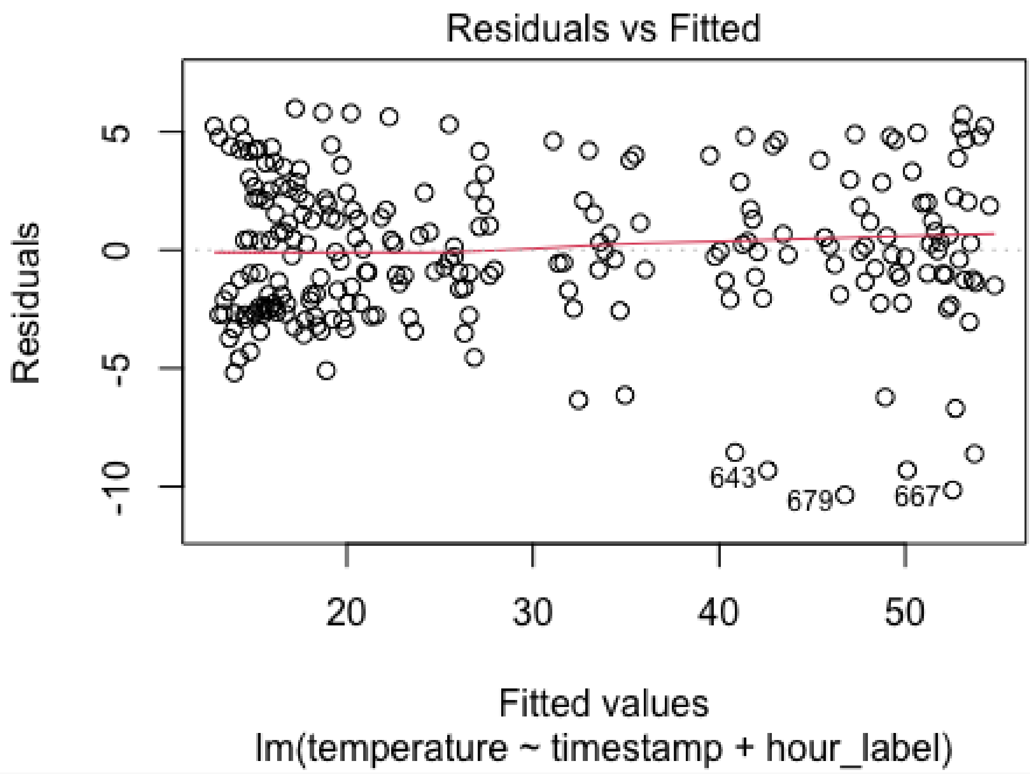
(b) Model 1: Fit a linear model to the data. You will also need to account for any seasonality in the data.

```
# Fit a linear model with hour as a factor to account for seasonality
model_1 <- lm(temperature ~ timestamp + hour_label, data = cabin_new)

# Print the summary of the model
summary(model_1)
```

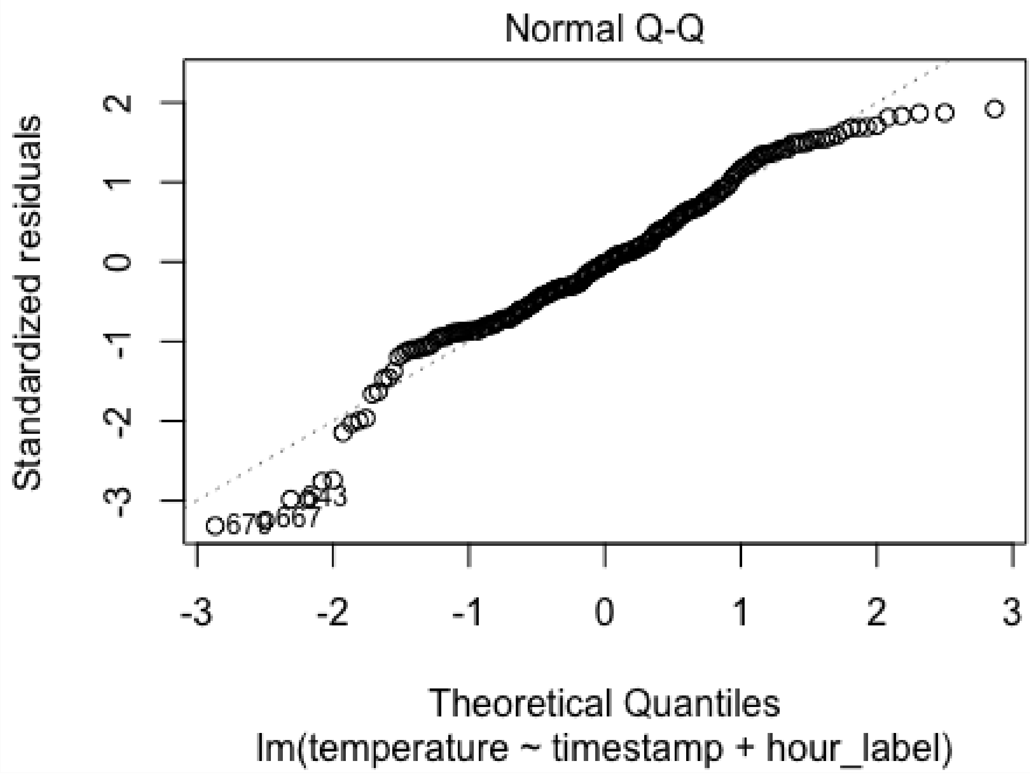
```
##
## Call:
## lm(formula = temperature ~ timestamp + hour_label, data = cabin_new)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.3556  -2.1020  -0.0696   2.0920   5.9833
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.94006    1.05902  18.829  < 2e-16 ***
## timestamp    -0.01130    0.00306  -3.693 0.000281 ***
## hour_label1  -1.51703    1.43820  -1.055 0.292690
## hour_label2  -2.22573    1.43813  -1.548 0.123169
## hour_label3  -2.72443    1.43807  -1.895 0.059494 .
## hour_label4  -3.18313    1.43801  -2.214 0.027905 *
## hour_label5  -3.72183    1.43796  -2.588 0.010301 *
## hour_label6  -4.29753    1.43792  -2.989 0.003126 **
## hour_label7  -4.54923    1.43789  -3.164 0.001781 **
## hour_label8   0.41207    1.43786   0.287 0.774703
## hour_label9   7.84337    1.43783   5.455 1.34e-07 ***
## hour_label10 16.20467    1.43782  11.270  < 2e-16 ***
## hour_label11 23.89597    1.43781  16.620  < 2e-16 ***
## hour_label12 30.21727    1.43781  21.016  < 2e-16 ***
## hour_label13 33.99857    1.43781  23.646  < 2e-16 ***
## hour_label14 35.02987    1.43782  24.363  < 2e-16 ***
## hour_label15 33.87117    1.43783  23.557  < 2e-16 ***
## hour_label16 31.44247    1.43786  21.868  < 2e-16 ***
## hour_label17 28.10377    1.43789  19.545  < 2e-16 ***
## hour_label18 22.21507    1.43792  15.449  < 2e-16 ***
## hour_label19 13.81637    1.43796   9.608  < 2e-16 ***
## hour_label20  8.22767    1.43801   5.722 3.50e-08 ***
## hour_label21  5.01898    1.43807   3.490 0.000585 ***
## hour_label22  2.97027    1.43813   2.065 0.040081 *
## hour_label23  1.46158    1.43820   1.016 0.310646
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.291 on 216 degrees of freedom
## Multiple R-squared:  0.9544, Adjusted R-squared:  0.9493
## F-statistic: 188.4 on 24 and 216 DF,  p-value: < 2.2e-16
AIC(model_1)
## [1] 1283.643
BIC(model_1)
## [1] 1374.248
```

Lachlan Bassi 22975276

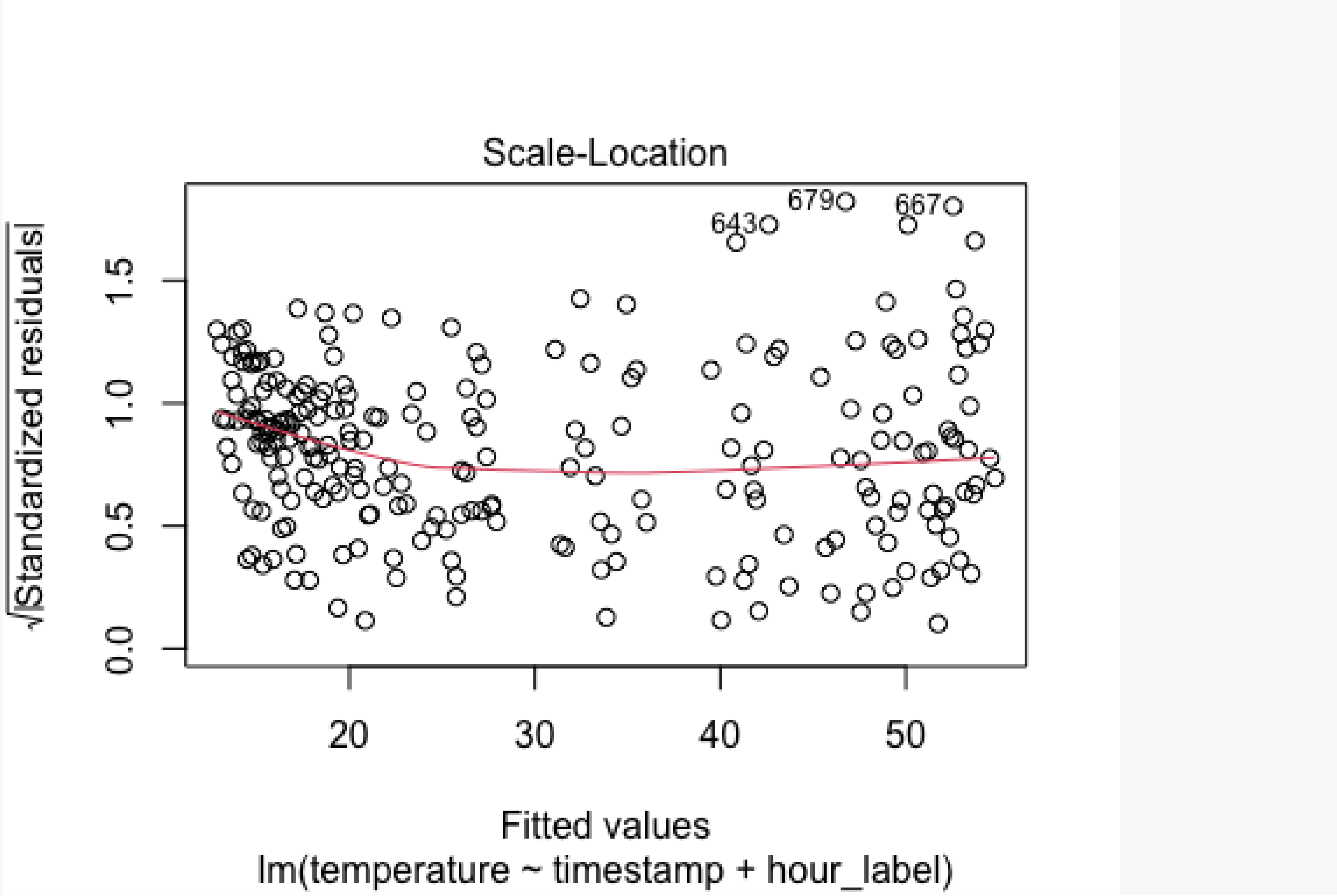
```r
plot(model_1)
```

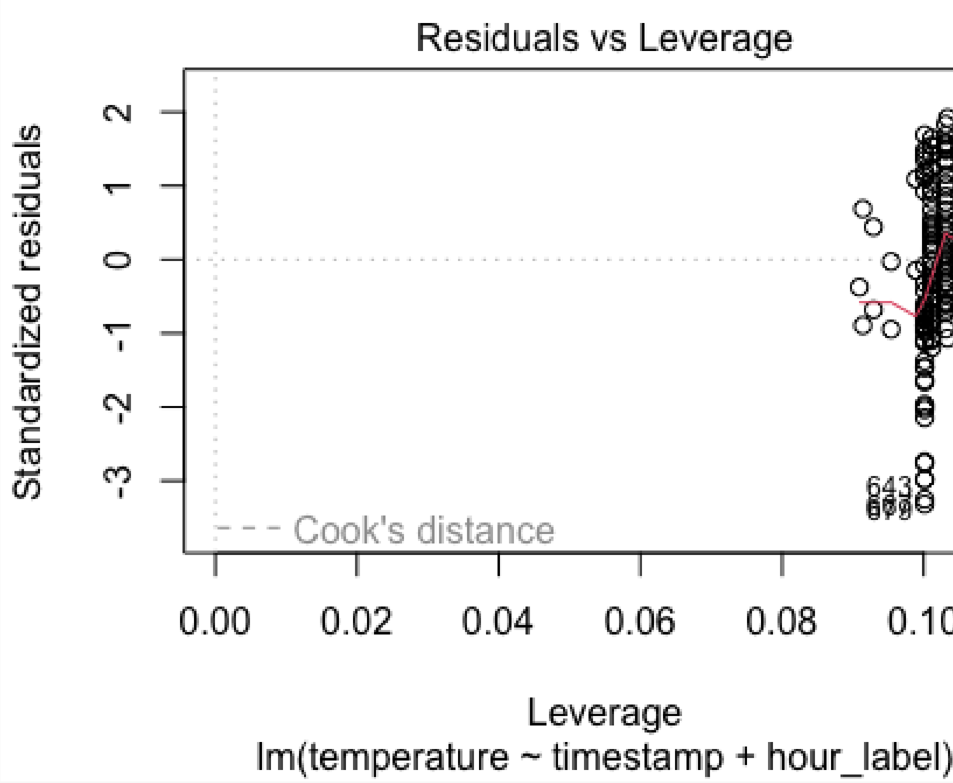


Lachlan Bassi 22975276

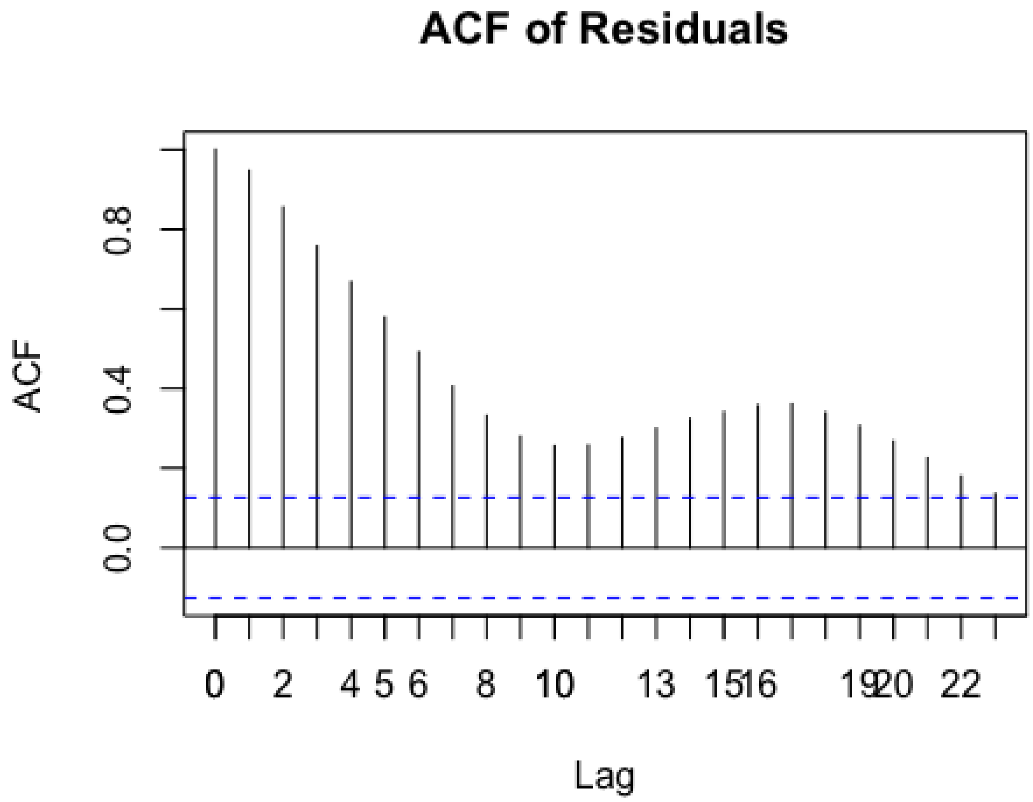


Lachlan Bassi 22975276



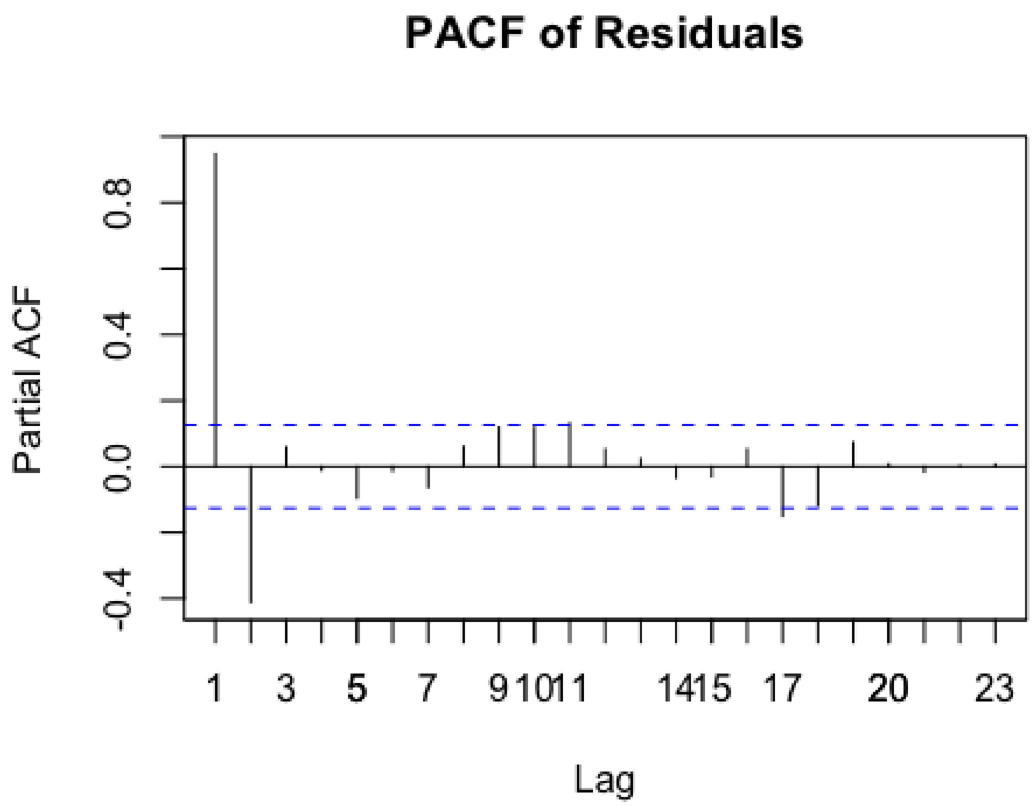



```r
acf(resid(model_1), main="ACF of Residuals")
axis(side = 1, at = seq(0, length(cabin_new$temperature), by = 1))
```

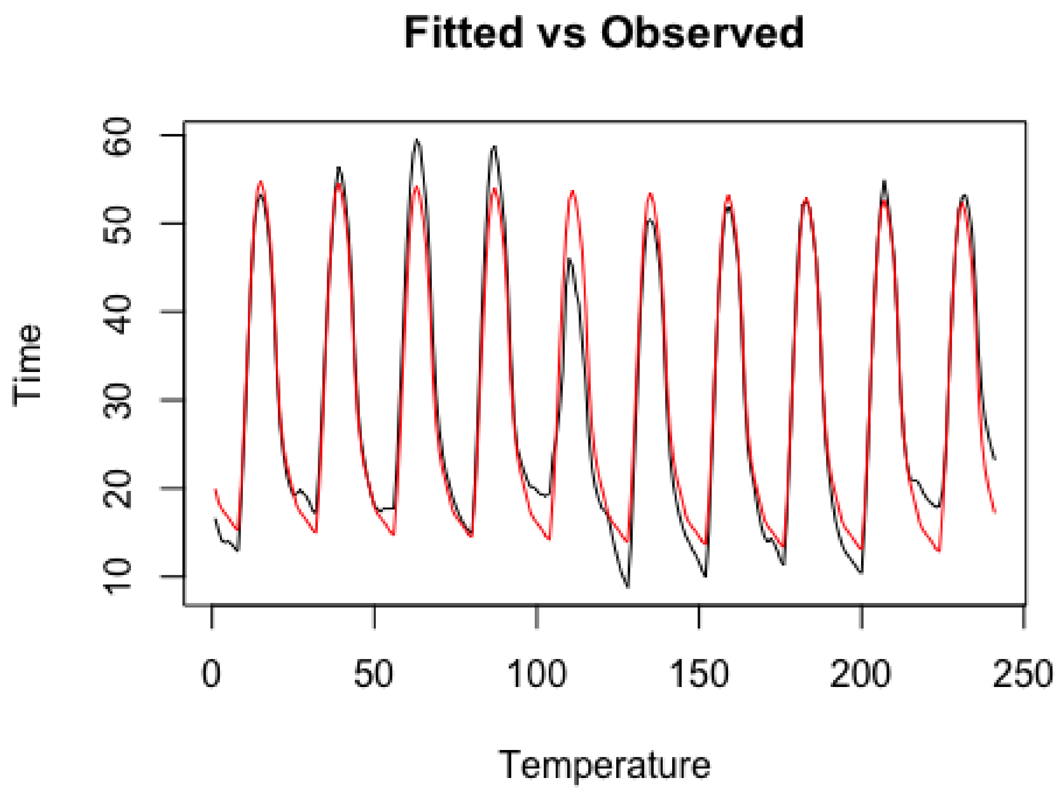



```
pacf(resid(model_1), main="PACF of Residuals")
axis(side = 1, at = seq(0, length(cabin_new$temperature), by = 1))
```

Lachlan Bassi 22975276




```r
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l", main = "Fitted vs Observed", xlab = "Temperature", ylab = "Time")
lines(model_1$fitted ~ cabin_new$timestamp, col = "red")
```



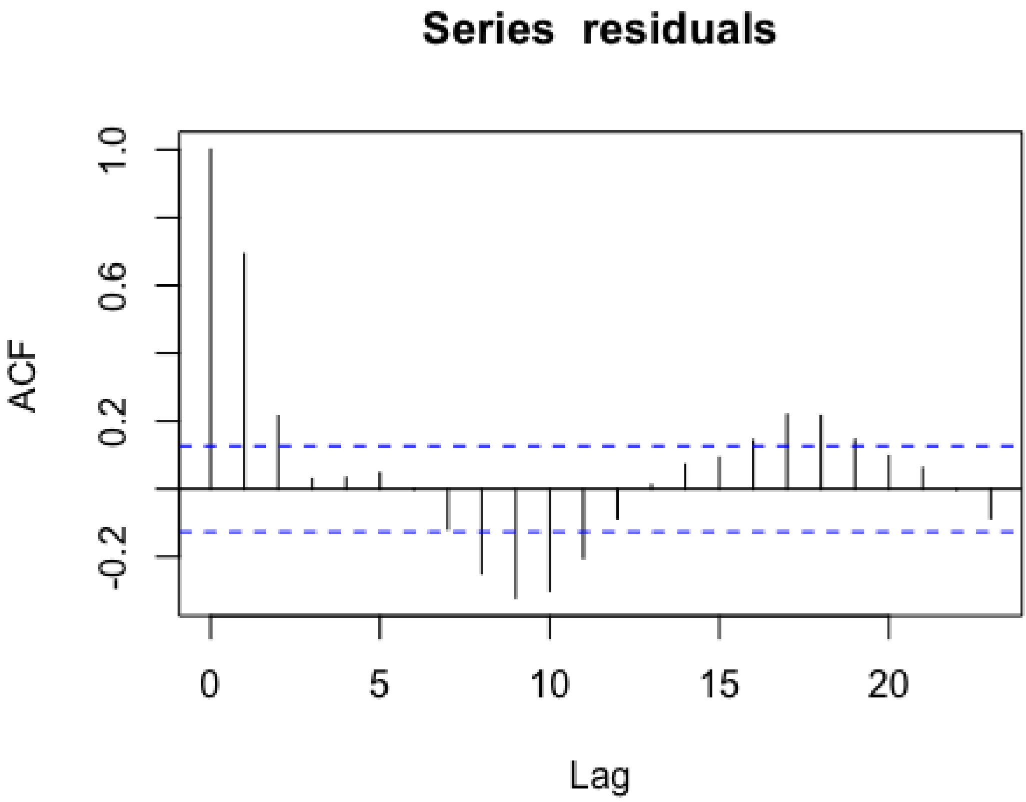

```
AIC(model_1)
## [1] 1283.643
BIC(model_1)
## [1] 1374.248
rmse <- sqrt(mean((cabin_new$temperature - model_1$fitted)^2))
print(paste("RMSE:", rmse))
## [1] "RMSE: 3.11533814987333"
```

(c) By analysing the residuals, select appropriate ARMA terms and include them in the linear model.

THIS MY IDEAL MODEL

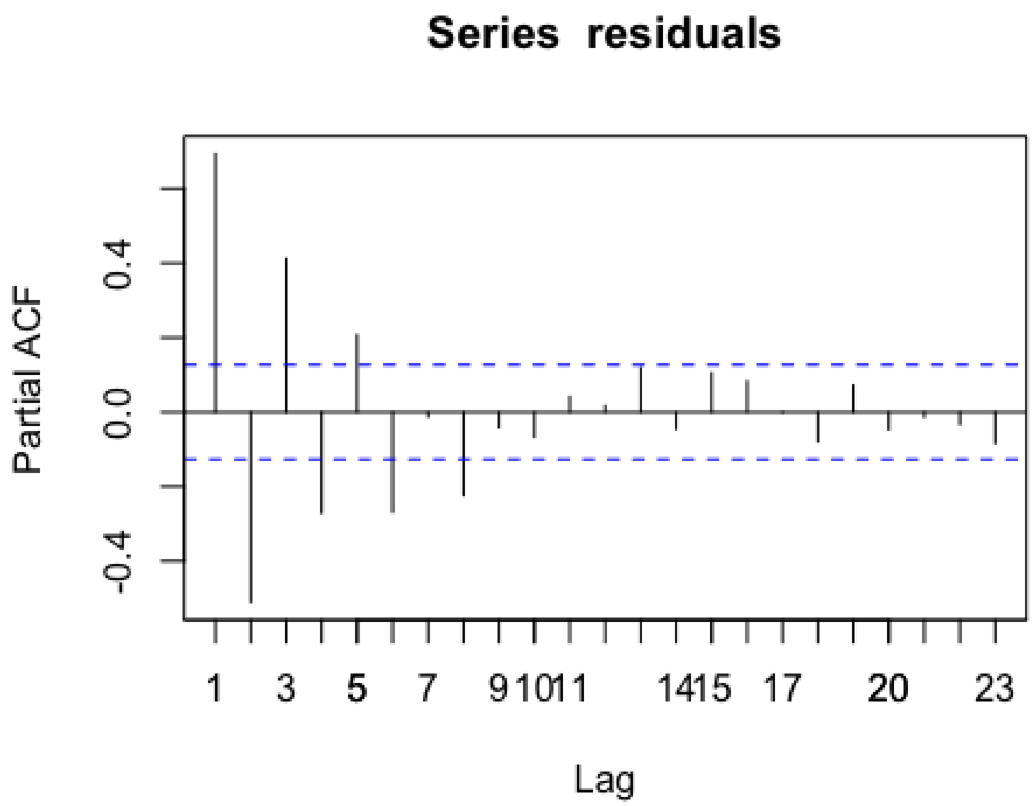
```
# Applying the AR(2) term to the data (creating new variables for each)
templag <- cabin_new$temperature[c(-1, -2)]
timelag <- cabin_new$timestamp[c(-1, -2)]
hourlag <- cabin_new$hour_label[c(-1, -2)]
templast <- cabin_new$temperature[c(-241,-240)]

# Fit the updated linear model
model_1_lagged <- lm(templag ~ timelag + hourlag + templast -1, data = cabin_new)
summary(model_1_lagged)
##
## Call:
## lm(formula = templag ~ timelag + hourlag + templast - 1, data = cabin_new)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.0564  -0.6673   0.0349   0.8536   3.0344
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## timelag   -0.0008582  0.0015626  -0.549 0.583413
## hourlag0  -0.0889182  0.9474190  -0.094 0.925314
## hourlag1   0.3289029  0.9009982   0.365 0.715442
## hourlag2   0.5215647  0.8440657   0.618 0.537289
## hourlag3   0.8357989  0.8206748   1.018 0.309630
## hourlag4   1.0041095  0.8030554   1.250 0.212539
## hourlag5   0.9064965  0.7909575   1.146 0.253050
## hourlag6   0.7364695  0.7800122   0.944 0.346149
## hourlag7   0.9612705  0.7672900   1.253 0.211648
## hourlag8   6.4318295  0.7539115   8.531 2.71e-15 ***
## hourlag9  14.0855349  0.7483622  18.822  < 2e-16 ***
## hourlag10 18.0539080  0.8733494  20.672  < 2e-16 ***
## hourlag11 19.1654652  1.0816715  17.718  < 2e-16 ***
## hourlag12 18.0836465  1.3331292  13.565  < 2e-16 ***
## hourlag13 15.0550126  1.5732824   9.569  < 2e-16 ***
## hourlag14 10.4893089  1.7745390   5.911 1.34e-08 ***
## hourlag15  5.9823956  1.8961655   3.155 0.001837 **
## hourlag16  2.6401963  1.9295965   1.368 0.172671
## hourlag17  0.3269147  1.8924462   0.173 0.863014
## hourlag18 -3.4119718  1.8145952  -1.880 0.061433 .
## hourlag19 -8.8551893  1.7080593  -5.184 5.03e-07 ***
## hourlag20 -9.2307629  1.5219318  -6.065 5.94e-09 ***
## hourlag21 -5.0041067  1.2624937  -3.964 0.000101 ***
## hourlag22 -2.1052855  1.0961782  -1.921 0.056121 .
## hourlag23 -0.7735985  1.0042440  -0.770 0.441958
## templast   0.8853505  0.0335916  26.356  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.597 on 213 degrees of freedom
## Multiple R-squared:  0.998,  Adjusted R-squared:  0.9977
## F-statistic:  4003 on 26 and 213 DF,  p-value: < 2.2e-16
# Obtain residuals
residuals <- residuals(model_1_lagged)

# Analyze ACF and PACF of residuals
acf(residuals)
```

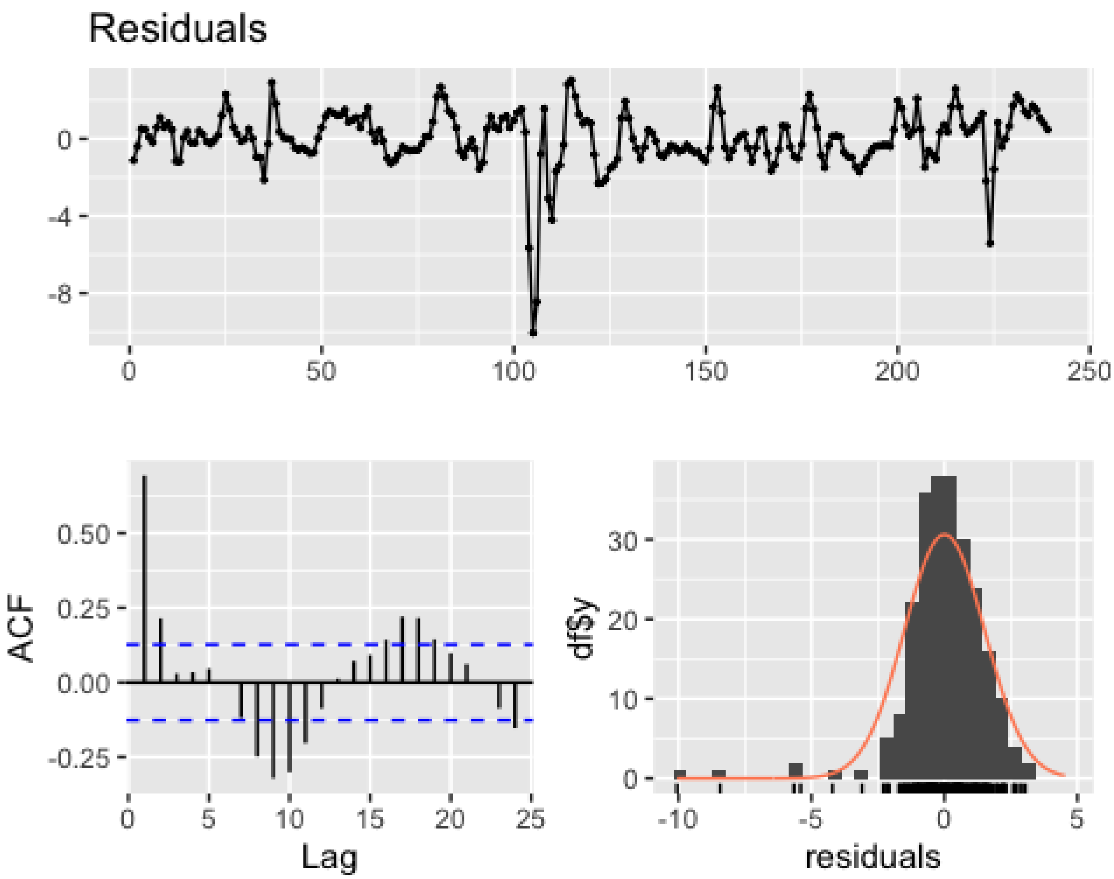



```
pacf(residuals)
axis(side = 1, at = seq(0, length(cabin_new$temperature), by = 1))
```

Lachlan Bassi 22975276




```r
# Based on the analysis, select appropriate ARMA terms

checkresiduals(model_1_lagged)
```

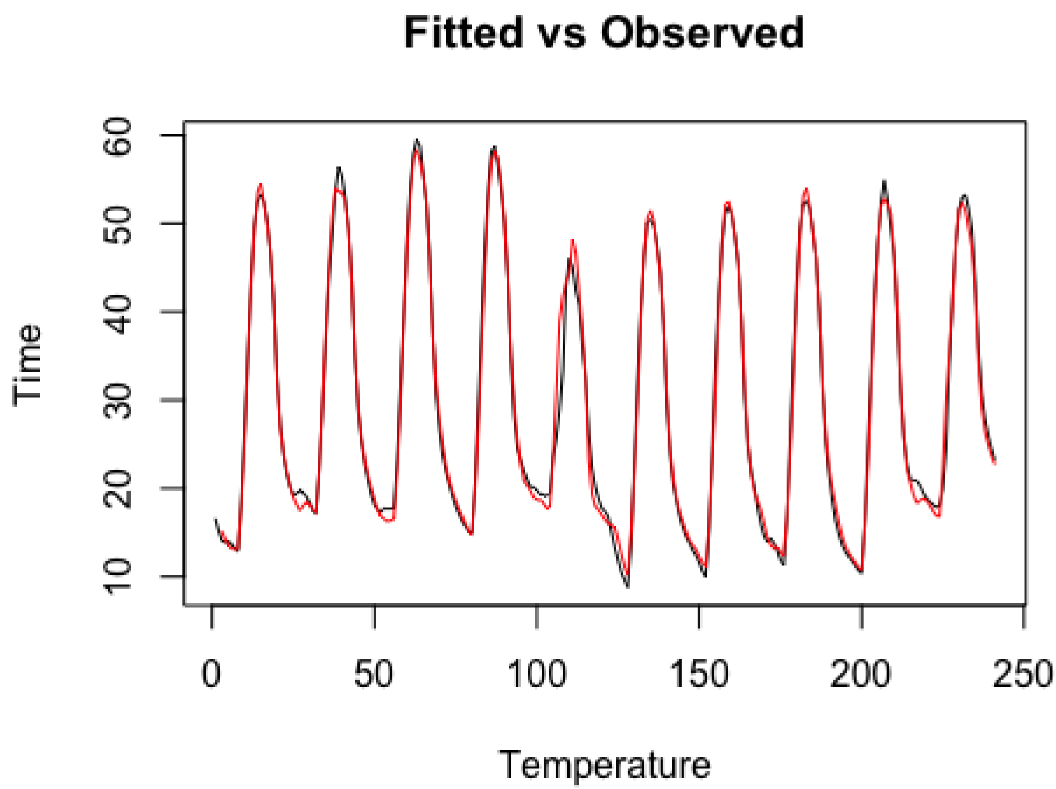



```
##
##  Breusch-Godfrey test for serial correlation of order up to 29
##
## data:  Residuals
## LM test = 184.65, df = 29, p-value < 2.2e-16
AIC(model_1_lagged)
## [1] 928.4838
BIC(model_1_lagged)
## [1] 1022.348
rmse(templag, model_1_lagged$fitted)
## [1] 1.50761
```

(d) Perform appropriate model diagnostics and verify model assumptions.

```
# Original data
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l",
     main = "Fitted vs Observed", xlab = "Temperature", ylab = "Time")

# Fitted values from the lagged model
lines(fitted(model_1_lagged) ~ timelag, col = "red")
```

Lachlan Bassi 22975276

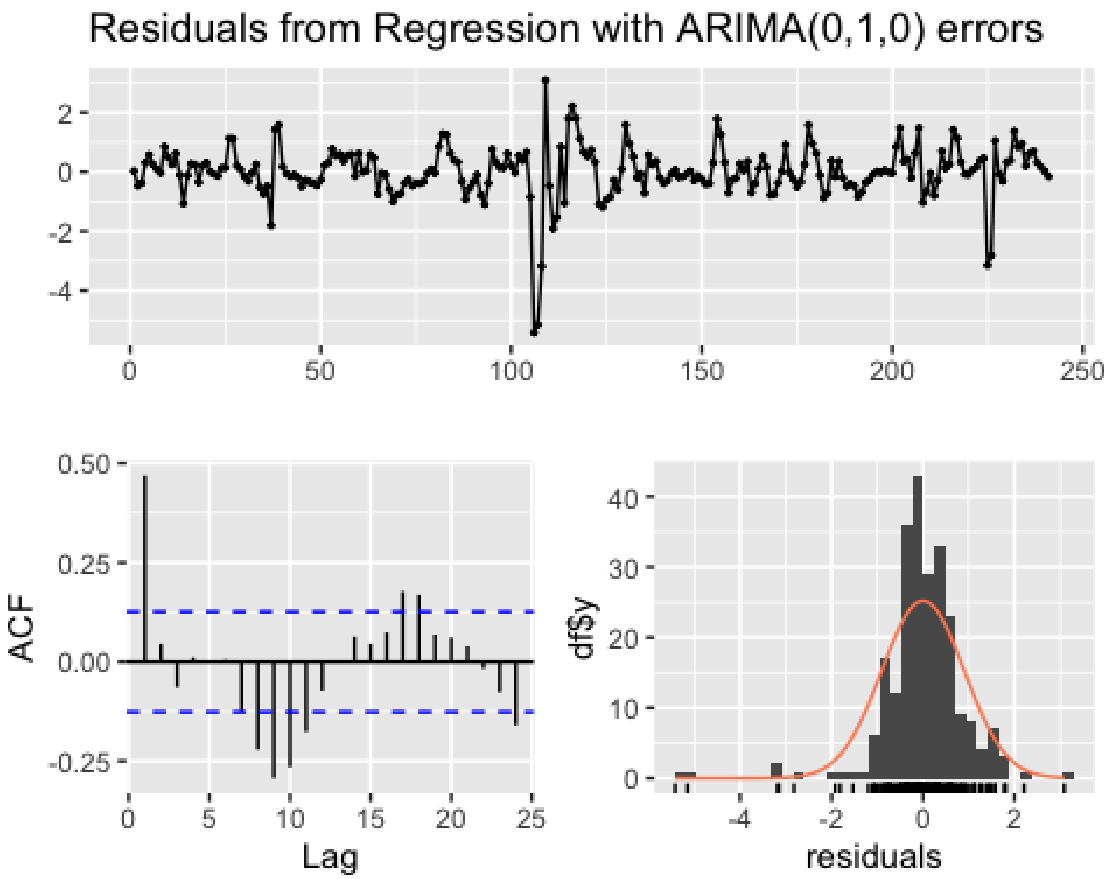



```r
predicted_values <- fitted(model_1_lagged)
actual_values <- templag  # Actual observed values

rmse_val <- rmse(actual_values, predicted_values)
print(paste("RMSE: ", rmse_val))
## [1] "RMSE:  1.50760965161663"
# AIC
aic_value <- AIC(model_1_lagged)
print(paste("AIC: ", aic_value))
## [1] "AIC:  928.483752598191"
# BIC
bic_value <- BIC(model_1_lagged)
print(paste("BIC: ", bic_value))
## [1] "BIC:  1022.34826850034"
# Ljung-Box test
ljung_box_result <- Box.test(residuals(model_1_lagged), type="Ljung", lag=10)
print(ljung_box_result)
##
##  Box-Ljung test
##
## data:  residuals(model_1_lagged)
## X-squared = 196.03, df = 10, p-value < 2.2e-16
model1.final = model_1_lagged
```

(e) Model 2: Now fit a linear model to the data using the Arima command, and perform model diagnostics. You will be able to include a time trend and sasonality in this model. See the lecture notes for details.

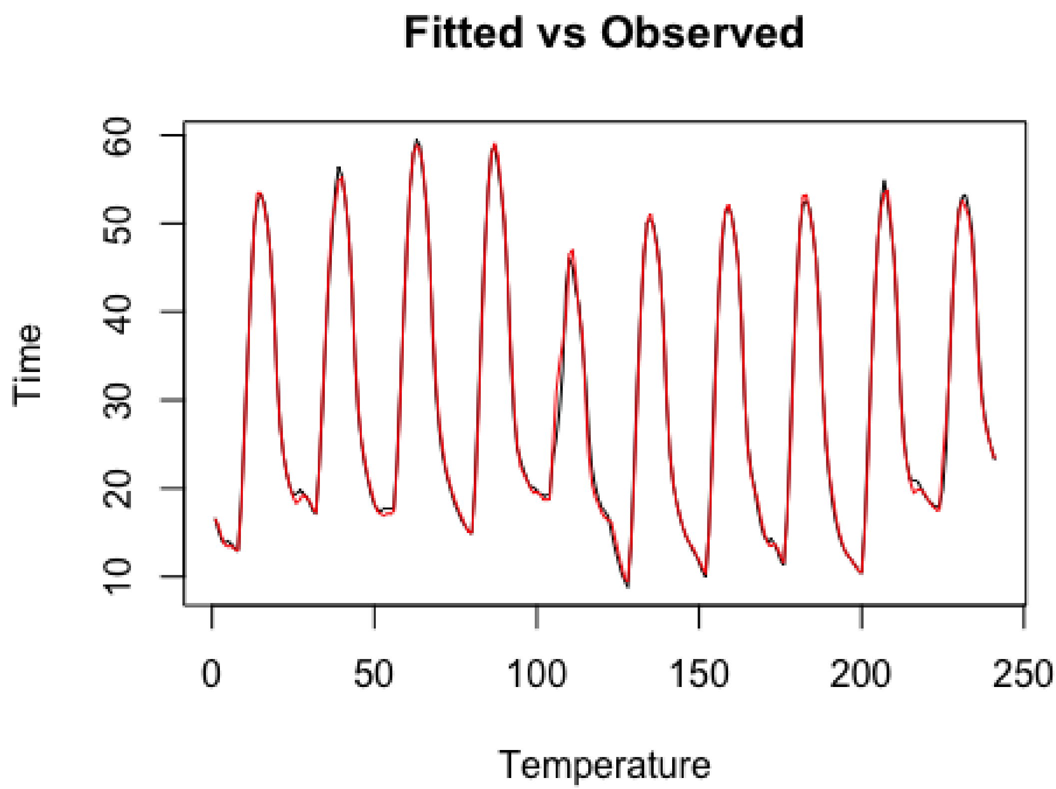
```
# Create a dummy variable for each unique hour_label, excluding the first one as a
reference category
dummy_hour_label <- model.matrix(~hour_label - 1, data = cabin_new)

model_2_auto <- auto.arima(cabin_new$temperature, xreg = cabin_new$time + dummy_hou
r_label, seasonal = TRUE)

summary(model_2_auto)
## Series: cabin_new$temperature
## Regression with ARIMA(0,1,0) errors
##
## Coefficients:
##        hour_label0  hour_label1  hour_label2  hour_label3  hour_label4
##           -11.4806     -12.4381     -13.1856     -13.7231     -14.2206
## s.e.        0.4030       0.4030       0.4030       0.4030       0.4030
##        hour_label5  hour_label6  hour_label7  hour_label8  hour_label9
##           -14.7981     -15.4126     -15.7031     -10.7806      -3.3881
## s.e.        0.4030       0.4030       0.4030       0.4030       0.4030
##        hour_label10  hour_label11  hour_label12  hour_label13  hour_label14
##              4.9344       12.5869       18.8694       22.6119       23.6044
## s.e.         0.4030        0.4030        0.4030        0.4030        0.4030
##        hour_label15  hour_label16  hour_label17  hour_label18  hour_label19
##             22.4069       19.9394       16.5619       10.6344        2.1969
## s.e.         0.4030        0.4030        0.4030        0.4030        0.4030
##        hour_label20  hour_label21  hour_label22  hour_label23
##             -3.4306       -6.6781       -8.7656      -10.3131
## s.e.         0.4030        0.4030        0.4030        0.4030
##
## sigma^2 = 0.9037:  log likelihood = -315.75
## AIC=681.5   AICc=687.58   BIC=768.52
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.000116403 0.8999793 0.5789878 -0.1071167 2.257679 0.1759396
##                   ACF1
## Training set 0.4692152
checkresiduals(model_2_auto)
```

Lachlan Bassi 22975276




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(0,1,0) errors
## Q* = 111.06, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
# Plotting fitted values against observed for arima model
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l", main = "Fitted vs Obs
erved", xlab = "Temperature", ylab = "Time")
lines(model_2_auto$fitted ~ cabin_new$timestamp, col = "red")
```

Lachlan Bassi 22975276

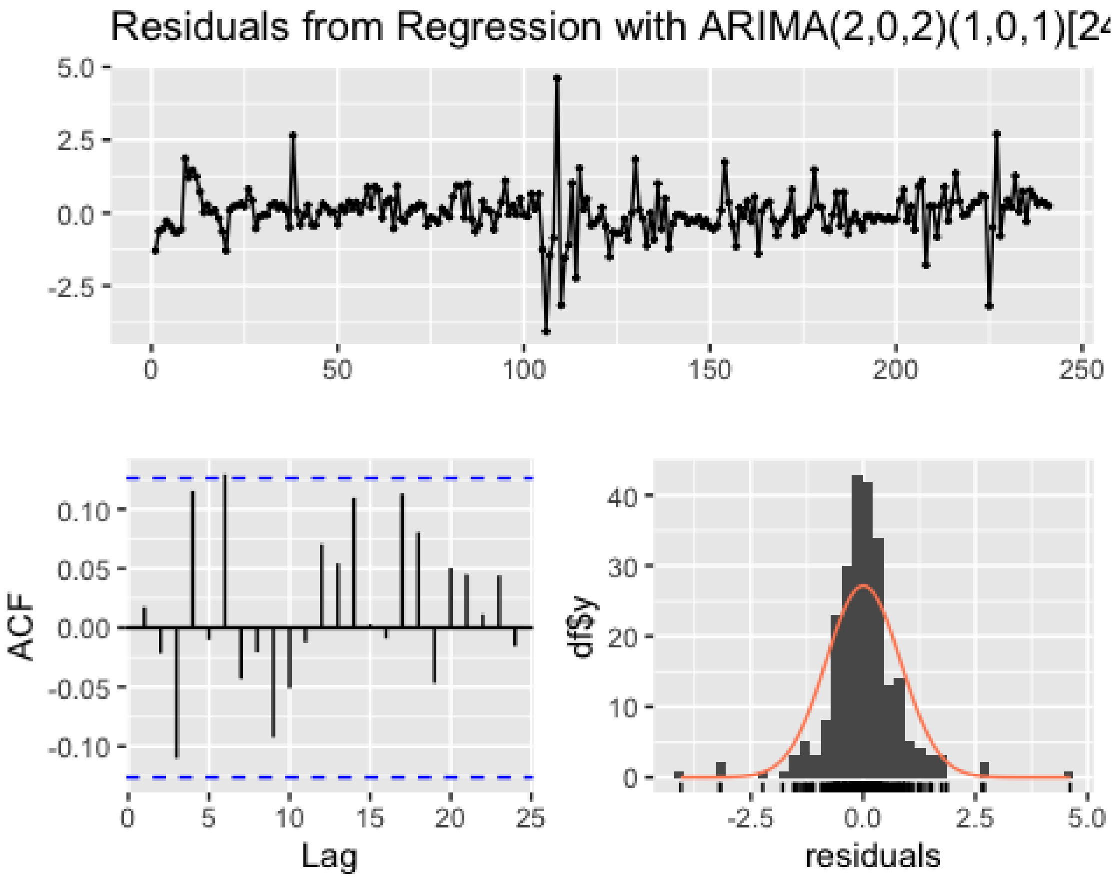



```
rmse(cabin_new$temperature, model_2_auto$fitted)
## [1] 0.8999793
```

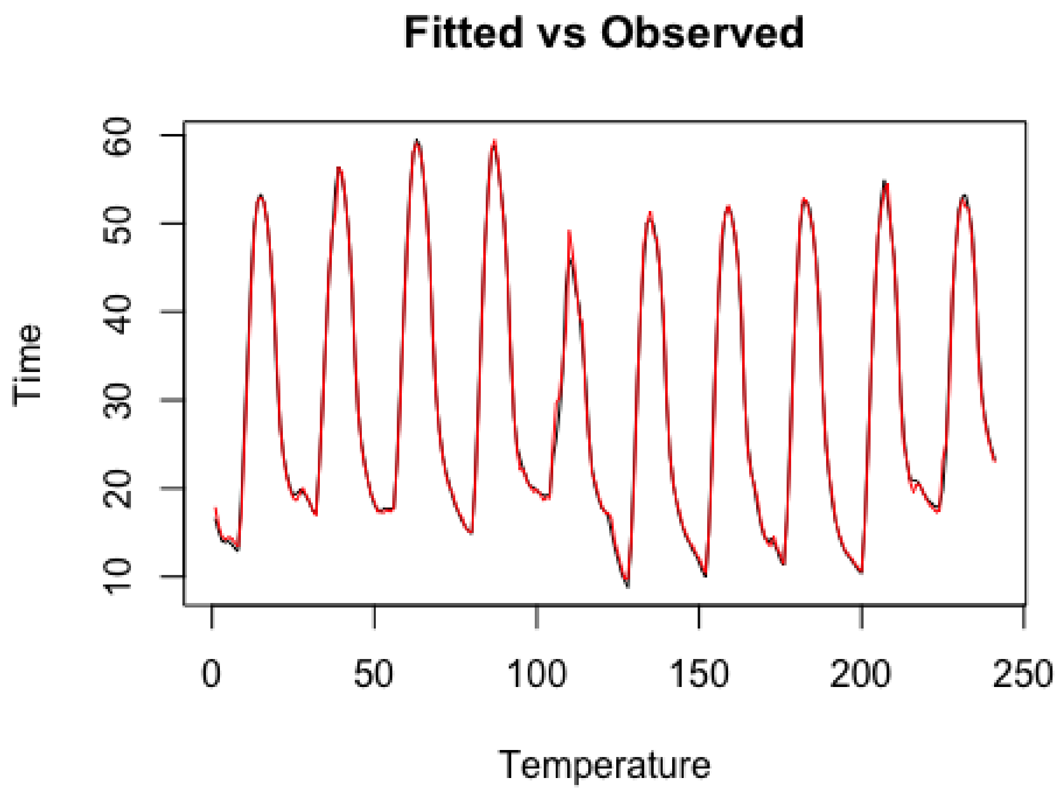
##FINAL MODEL

```
model2_improved <- Arima(cabin_new$temperature, order = c(2, 0, 2), seasonal = list
(order = c(1, 0, 1), period = 24),
               xreg = 1:nrow(cabin_new))
summary(model2_improved)
## Series: cabin_new$temperature
## Regression with ARIMA(2,0,2)(1,0,1)[24] errors
##
## Coefficients:
##          ar1      ar2      ma1      ma2    sar1     sma1  intercept     xreg
##       1.6677  -0.7244  -0.0582  -0.1774  0.9999  -0.9708    30.7798  -0.0084
## s.e.  0.1313   0.1227   0.1571   0.1075  0.0003   0.0508     5.2867   0.0100
##
## sigma^2 = 0.709:  log likelihood = -342.41
## AIC=702.82   AICc=703.6   BIC=734.18
##
## Training set error measures:
##                      ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.00460222 0.8279104 0.5426746 -0.2493218 2.099361 0.1649049
```

```
##                        ACF1
## Training set 0.01737354
# Residual diagnostics
checkresiduals(model2_improved)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from Regression with ARIMA(2,0,2)(1,0,1)[24] errors
## Q* = 14.088, df = 4, p-value = 0.007019
##
## Model df: 6.   Total lags used: 10
# Ljung-Box test
Box.test(residuals(model2_improved), type = "Ljung-Box")
##
##  Box-Ljung test
##
## data:  residuals(model2_improved)
## X-squared = 0.073653, df = 1, p-value = 0.7861
# Plotting fitted values against observed for arima model
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l", main = "Fitted vs Obs
erved", xlab = "Temperature", ylab = "Time")
lines(model2_improved$fitted ~ cabin_new$timestamp, col = "red")
```




```r
rmse(cabin_new$temperature, model2_improved$fitted)
## [1] 0.8279104
model2.final = model2_improved
AIC(model2_improved, model_2_auto)
## Warning in AIC.default(model2_improved, model_2_auto): models are not all
## fitted to the same number of observations
##                 df      AIC
## model2_improved  9 702.8199
## model_2_auto    25 681.5033
BIC(model2_improved, model_2_auto)
## Warning in BIC.default(model2_improved, model_2_auto): models are not all
## fitted to the same number of observations
##                 df      BIC
## model2_improved  9 734.1830
## model_2_auto    25 768.5193
```

(f) Model 3: Fit a linear model to the data using the gls command. Following this, model any heterogeneous variance terms, and include appropriate covariance structure. Note that you can include an AR(2) term in this model if it is appropriate. Again perform appropriate model diagnostics.

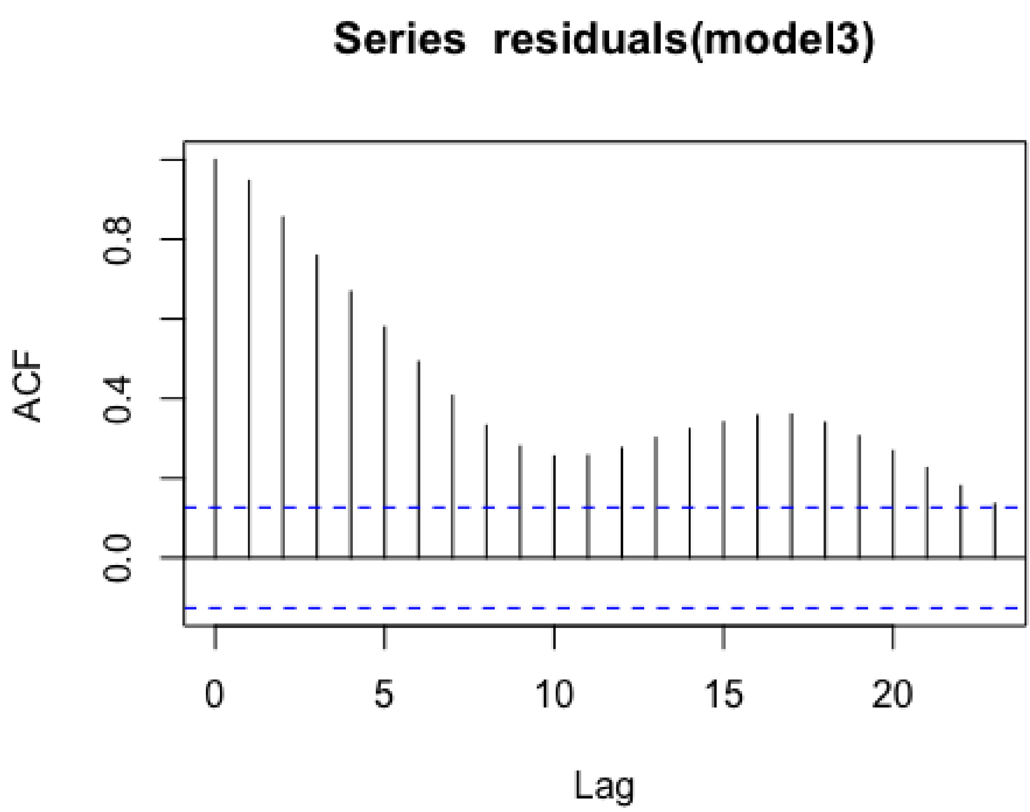
```r
model3 <- gls(temperature ~ timestamp + hour_label, data = cabin_new)
# Print the summary
summary(model3)
```

Lachlan Bassi 22975276

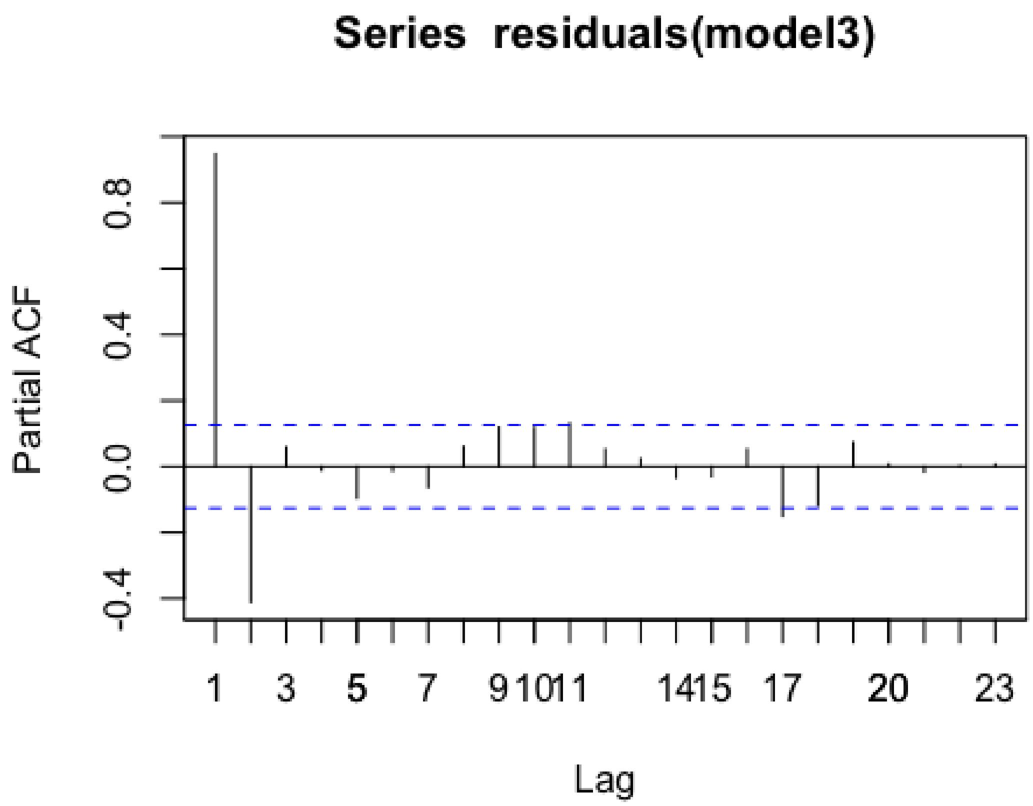
```
## Generalized least squares fit by REML
##   Model: temperature ~ timestamp + hour_label
##   Data: cabin_new
##        AIC      BIC    logLik
##   1248.853 1336.611 -598.4267
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept)  19.94005 1.0590233 18.828721  0.0000
## timestamp    -0.01130 0.0030602 -3.692661  0.0003
## hour_label1  -1.51703 1.4381995 -1.054812  0.2927
## hour_label2  -2.22573 1.4381311 -1.547654  0.1232
## hour_label3  -2.72443 1.4380693 -1.894505  0.0595
## hour_label4  -3.18313 1.4380139 -2.213559  0.0279
## hour_label5  -3.72183 1.4379651 -2.588261  0.0103
## hour_label6  -4.29753 1.4379227 -2.988706  0.0031
## hour_label7  -4.54923 1.4378869 -3.163829  0.0018
## hour_label8   0.41207 1.4378576  0.286587  0.7747
## hour_label9   7.84337 1.4378348  5.454988  0.0000
## hour_label10 16.20467 1.4378185 11.270318  0.0000
## hour_label11 23.89597 1.4378088 16.619715  0.0000
## hour_label12 30.21727 1.4378055 21.016245  0.0000
## hour_label13 33.99857 1.4378088 23.646102  0.0000
## hour_label14 35.02987 1.4378185 24.363209  0.0000
## hour_label15 33.87117 1.4378348 23.557069  0.0000
## hour_label16 31.44247 1.4378576 21.867585  0.0000
## hour_label17 28.10377 1.4378869 19.545191  0.0000
## hour_label18 22.21507 1.4379227 15.449421  0.0000
## hour_label19 13.81637 1.4379651  9.608282  0.0000
## hour_label20  8.22767 1.4380139  5.721554  0.0000
## hour_label21  5.01897 1.4380693  3.490079  0.0006
## hour_label22  2.97028 1.4381311  2.065371  0.0401
## hour_label23  1.46158 1.4381995  1.016253  0.3106
##
##  Correlation:
##              (Intr) tmstmp hr_lb1 hr_lb2 hr_lb3 hr_lb4 hr_lb5 hr_lb6 hr_lb7
## timestamp    -0.350
## hour_label1  -0.655  0.023
## hour_label2  -0.654  0.021  0.476
## hour_label3  -0.653  0.019  0.476  0.476
## hour_label4  -0.652  0.017  0.476  0.476  0.476
## hour_label5  -0.652  0.015  0.476  0.476  0.476  0.476
## hour_label6  -0.651  0.013  0.476  0.476  0.476  0.476  0.476
## hour_label7  -0.650  0.011  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label8  -0.649  0.009  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label9  -0.649  0.006  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label10 -0.648  0.004  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label11 -0.647  0.002  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label12 -0.647  0.000  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label13 -0.646 -0.002  0.476  0.476  0.476  0.476  0.476  0.476  0.476
```

```
## hour_label14 -0.645 -0.004  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label15 -0.644 -0.006  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label16 -0.644 -0.009  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label17 -0.643 -0.011  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label18 -0.642 -0.013  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label19 -0.641 -0.015  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label20 -0.640 -0.017  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label21 -0.640 -0.019  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label22 -0.639 -0.021  0.475  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label23 -0.638 -0.023  0.475  0.475  0.476  0.476  0.476  0.476  0.476
##              hr_lb8 hr_lb9 hr_l10 hr_l11 hr_l12 hr_l13 hr_l14 hr_l15 hr_l16
## timestamp
## hour_label1
## hour_label2
## hour_label3
## hour_label4
## hour_label5
## hour_label6
## hour_label7
## hour_label8
## hour_label9   0.476
## hour_label10  0.476  0.476
## hour_label11  0.476  0.476  0.476
## hour_label12  0.476  0.476  0.476  0.476
## hour_label13  0.476  0.476  0.476  0.476  0.476
## hour_label14  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label15  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label16  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label17  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label18  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label19  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label20  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label21  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label22  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
## hour_label23  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476  0.476
##              hr_l17 hr_l18 hr_l19 hr_l20 hr_l21 hr_l22
## timestamp
## hour_label1
## hour_label2
## hour_label3
## hour_label4
## hour_label5
## hour_label6
## hour_label7
## hour_label8
## hour_label9
## hour_label10
## hour_label11
## hour_label12
## hour_label13
```

```
## hour_label14
## hour_label15
## hour_label16
## hour_label17
## hour_label18  0.476
## hour_label19  0.476  0.476
## hour_label20  0.476  0.476  0.476
## hour_label21  0.476  0.476  0.476  0.476
## hour_label22  0.476  0.476  0.476  0.476  0.476
## hour_label23  0.476  0.476  0.476  0.476  0.476  0.476
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -3.14694058 -0.63876784 -0.02114309  0.63572896  1.81825148
##
## Residual standard error: 3.290689
## Degrees of freedom: 241 total; 216 residual
# Autocorrelation of residuals
acf(residuals(model3))
```




```
pacf(residuals(model3))
axis(side = 1, at = seq(0, length(cabin_new$temperature), by = 1))
```



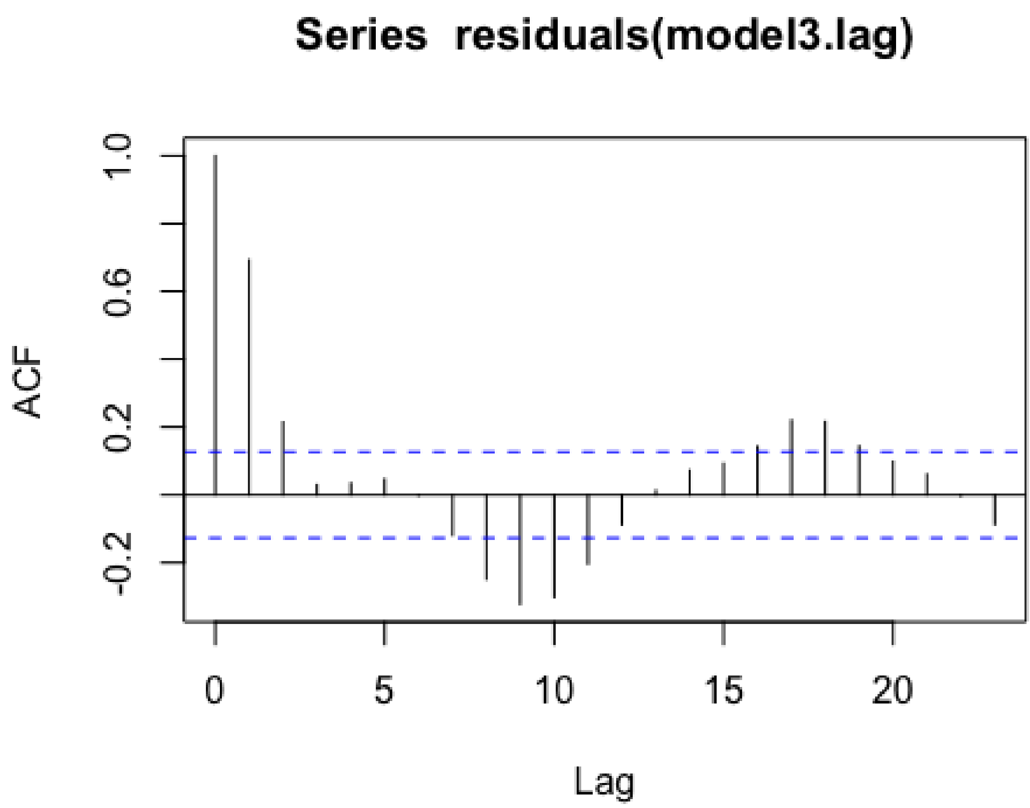

```
model3.lag <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new)
summary(model3.lag)
## Generalized least squares fit by REML
##   Model: templag ~ timelag + hourlag + templast - 1
##   Data: cabin_new
##        AIC      BIC    logLik
##   934.6985 1025.453 -440.3493
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## timelag   -0.000858 0.0015626 -0.549244  0.5834
## hourlag0  -0.088918 0.9474190 -0.093853  0.9253
## hourlag1   0.328903 0.9009982  0.365043  0.7154
## hourlag2   0.521565 0.8440657  0.617920  0.5373
## hourlag3   0.835799 0.8206748  1.018429  0.3096
## hourlag4   1.004109 0.8030554  1.250361  0.2125
## hourlag5   0.906496 0.7909575  1.146075  0.2531
## hourlag6   0.736469 0.7800122  0.944177  0.3461
## hourlag7   0.961270 0.7672900  1.252812  0.2116
## hourlag8   6.431829 0.7539115  8.531279  0.0000
## hourlag9  14.085535 0.7483622 18.821815  0.0000
## hourlag10 18.053908 0.8733494 20.672032  0.0000
## hourlag11 19.165465 1.0816715 17.718378  0.0000
```

```
## hourlag12 18.083646 1.3331292 13.564811  0.0000
## hourlag13 15.055013 1.5732824  9.569174  0.0000
## hourlag14 10.489309 1.7745390  5.911005  0.0000
## hourlag15  5.982396 1.8961655  3.154997  0.0018
## hourlag16  2.640196 1.9295965  1.368263  0.1727
## hourlag17  0.326915 1.8924462  0.172747  0.8630
## hourlag18 -3.411972 1.8145952 -1.880294  0.0614
## hourlag19 -8.855189 1.7080593 -5.184357  0.0000
## hourlag20 -9.230763 1.5219318 -6.065162  0.0000
## hourlag21 -5.004107 1.2624937 -3.963669  0.0001
## hourlag22 -2.105285 1.0961782 -1.920569  0.0561
## hourlag23 -0.773599 1.0042440 -0.770329  0.4420
## templast   0.885351 0.0335916 26.356277  0.0000
##
##  Correlation:
##           timelg horlg0 horlg1 horlg2 horlg3 horlg4 horlg5 horlg6 horlg7 horlg8
## hourlag0  -0.425
## hourlag1  -0.408  0.683
## hourlag2  -0.401  0.678  0.646
## hourlag3  -0.404  0.667  0.636  0.632
## hourlag4  -0.407  0.658  0.627  0.623  0.613
## hourlag5  -0.409  0.651  0.621  0.616  0.607  0.598
## hourlag6  -0.411  0.644  0.614  0.610  0.600  0.592  0.587
## hourlag7  -0.414  0.636  0.607  0.602  0.593  0.585  0.579  0.574
## hourlag8  -0.416  0.627  0.598  0.593  0.584  0.577  0.571  0.566  0.559
## hourlag9  -0.418  0.623  0.594  0.590  0.581  0.573  0.567  0.562  0.555  0.548
## hourlag10 -0.411  0.690  0.658  0.654  0.643  0.634  0.628  0.621  0.613  0.604
## hourlag11 -0.396  0.747  0.712  0.707  0.695  0.685  0.677  0.670  0.661  0.651
## hourlag12 -0.380  0.779  0.742  0.738  0.725  0.714  0.706  0.698  0.688  0.677
## hourlag13 -0.367  0.795  0.757  0.753  0.739  0.728  0.719  0.711  0.701  0.690
## hourlag14 -0.359  0.803  0.765  0.761  0.747  0.735  0.726  0.718  0.708  0.696
## hourlag15 -0.355  0.806  0.769  0.764  0.750  0.738  0.729  0.721  0.711  0.699
## hourlag16 -0.354  0.807  0.769  0.765  0.751  0.739  0.730  0.722  0.711  0.700
## hourlag17 -0.356  0.806  0.769  0.764  0.750  0.738  0.730  0.721  0.711  0.699
## hourlag18 -0.360  0.804  0.767  0.762  0.748  0.737  0.728  0.719  0.709  0.698
## hourlag19 -0.366  0.801  0.764  0.759  0.745  0.734  0.725  0.717  0.706  0.695
## hourlag20 -0.376  0.793  0.756  0.751  0.738  0.726  0.718  0.710  0.700  0.689
## hourlag21 -0.394  0.773  0.737  0.732  0.719  0.709  0.701  0.693  0.684  0.673
## hourlag22 -0.409  0.750  0.715  0.710  0.698  0.688  0.681  0.674  0.665  0.655
## hourlag23 -0.419  0.731  0.697  0.692  0.681  0.671  0.664  0.657  0.649  0.639
## templast   0.271 -0.819 -0.781 -0.776 -0.761 -0.748 -0.738 -0.729 -0.718 -0.705
##           horlg9 hrlg10 hrlg11 hrlg12 hrlg13 hrlg14 hrlg15 hrlg16 hrlg17 hrlg18
## hourlag0
## hourlag1
## hourlag2
## hourlag3
## hourlag4
## hourlag5
## hourlag6
## hourlag7
```

```
## hourlag8
## hourlag9
## hourlag10  0.600
## hourlag11  0.646  0.720
## hourlag12  0.672  0.751  0.818
## hourlag13  0.684  0.766  0.836  0.876
## hourlag14  0.691  0.774  0.845  0.887  0.908
## hourlag15  0.693  0.777  0.849  0.891  0.913  0.924
## hourlag16  0.694  0.778  0.850  0.892  0.914  0.925  0.930
## hourlag17  0.694  0.777  0.849  0.891  0.913  0.924  0.929  0.930
## hourlag18  0.692  0.776  0.847  0.888  0.910  0.921  0.926  0.927  0.926
## hourlag19  0.690  0.772  0.843  0.884  0.905  0.916  0.921  0.922  0.921  0.918
## hourlag20  0.683  0.764  0.833  0.873  0.893  0.904  0.909  0.910  0.909  0.906
## hourlag21  0.668  0.745  0.810  0.848  0.867  0.877  0.881  0.882  0.881  0.879
## hourlag22  0.650  0.723  0.785  0.820  0.838  0.847  0.851  0.852  0.851  0.849
## hourlag23  0.635  0.705  0.764  0.797  0.814  0.822  0.826  0.827  0.826  0.824
## templast  -0.699 -0.790 -0.868 -0.915 -0.940 -0.953 -0.959 -0.960 -0.959 -0.955
##           hrlg19 hrlg20 hrlg21 hrlg22 hrlg23
## hourlag0
## hourlag1
## hourlag2
## hourlag3
## hourlag4
## hourlag5
## hourlag6
## hourlag7
## hourlag8
## hourlag9
## hourlag10
## hourlag11
## hourlag12
## hourlag13
## hourlag14
## hourlag15
## hourlag16
## hourlag17
## hourlag18
## hourlag19
## hourlag20  0.901
## hourlag21  0.874  0.864
## hourlag22  0.845  0.835  0.813
## hourlag23  0.820  0.812  0.791  0.767
## templast  -0.949 -0.935 -0.903 -0.869 -0.841
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -6.29712551 -0.41784024  0.02187851  0.53450484  1.90006308
##
## Residual standard error: 1.596975
## Degrees of freedom: 239 total; 213 residual
```

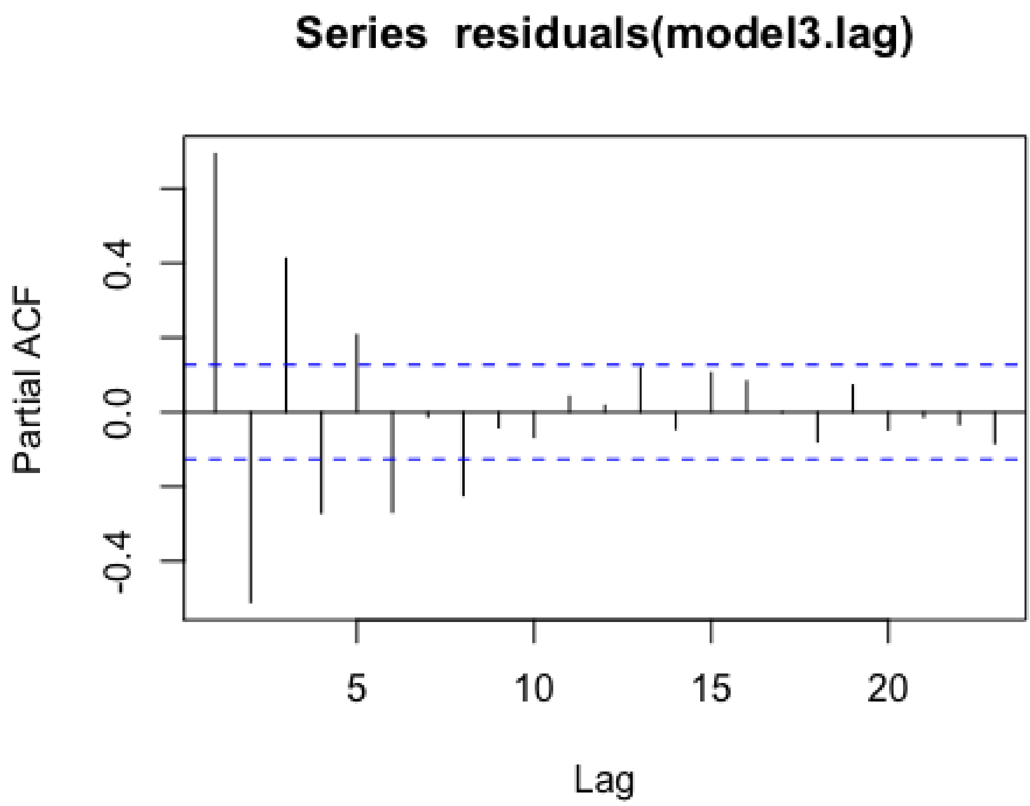
Lachlan Bassi 22975276

```
# Autocorrelation of residuals
acf(residuals(model3.lag))
```




```
pacf(residuals(model3.lag))
axis(side = 1, at = seq(0, length(cabin_new$templag), by = 1))
```

Lachlan Bassi 22975276



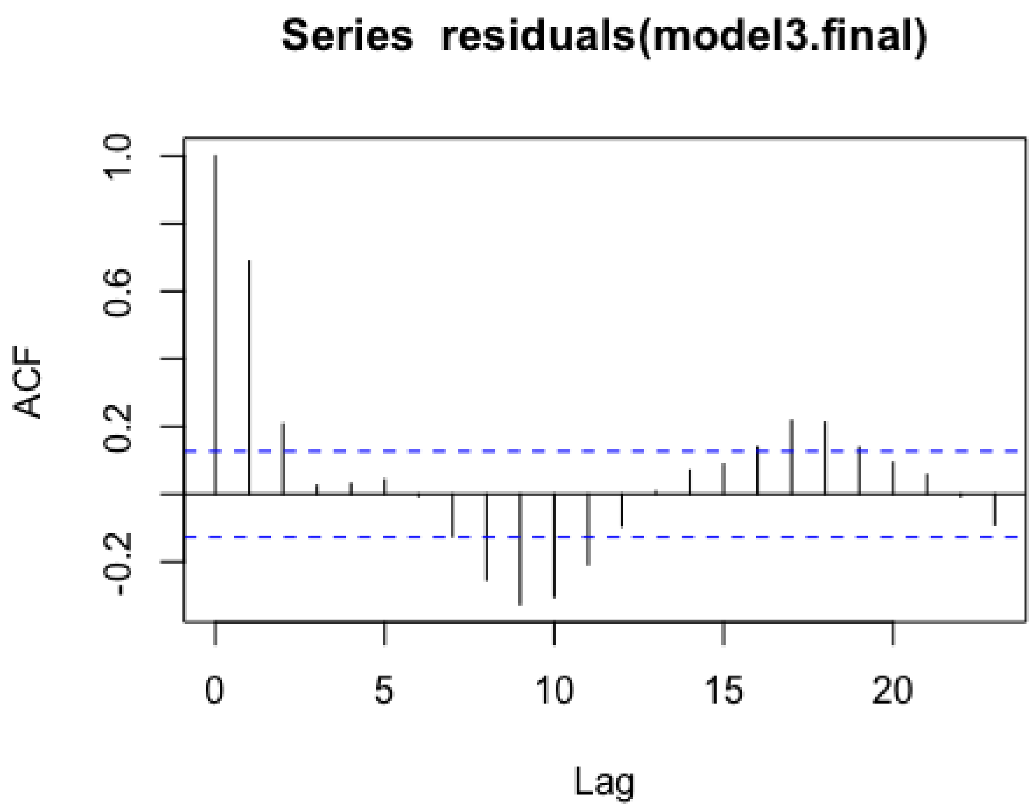

```
model3.lag <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, met
hod = 'ML')
model3_cov_1 <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, w
eights = varIdent(form = ~1 | hourlag), method = "ML")
model3_cov_2 <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, w
eights = varExp(form = ~ timelag), method = "ML")
model3_cov_3 <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, w
eights = varPower(form = ~ timelag), method = "ML")
anova(model3_cov_1, model3_cov_2)
##              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## model3_cov_1     1 50 846.6589 1020.482 -373.3295
## model3_cov_2     2 28 929.3446 1026.686 -436.6723 1 vs 2 126.6857  <.0001
anova(model3_cov_2, model3_cov_3)
##              Model df      AIC      BIC    logLik
## model3_cov_2     1 28 929.3446 1026.686 -436.6723
## model3_cov_3     2 28 915.7145 1013.056 -429.8572
anova(model3.lag, model3_cov_3)
##              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## model3.lag       1 27 928.4838 1022.348 -437.2419
## model3_cov_3     2 28 915.7145 1013.056 -429.8572 1 vs 2 14.76927   1e-04
model3_cor1 <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, we
ights = varPower(form = ~ timelag), cor = corAR1(), method = "ML")
model3_cor2 <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, we
```

```
ights = varPower(form = ~ timelag), cor = corCompSymm(), method = "ML")
model3_cor3 <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, we
ights = varPower(form = ~ timelag), cor = corARMA(p = 1, q = 1), method = "ML")
model3_cor4 <- gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, we
ights = varPower(form = ~ timelag), correlation = corARMA(p = 2, form = ~ 1 | hourl
ag), method = "ML")
anova(model3_cor1, model3_cor2)
##             Model df      AIC      BIC    logLik
## model3_cor1     1 29 671.2092 772.0267 -306.6046
## model3_cor2     2 29 887.9243 988.7418 -414.9622
anova(model3_cor1, model3_cor3)
##             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## model3_cor1     1 29 671.2092 772.0267 -306.6046
## model3_cor3     2 30 608.4440 712.7379 -274.2220 1 vs 2 64.76523  <.0001
anova(model3_cor3, model3_cor4)
##             Model df      AIC       BIC    logLik
## model3_cor3     1 30 608.4440  712.7379 -274.2220
## model3_cor4     2 30 899.3273 1003.6212 -419.6636
anova(model3_cor3, model3_cov_3)
##              Model df      AIC       BIC    logLik   Test  L.Ratio p-value
## model3_cor3      1 30 608.4440  712.7379 -274.2220
## model3_cov_3     2 28 915.7145 1013.0555 -429.8572 1 vs 2 311.2705  <.0001
anova(model3_cor4, model3_cov_3)
##              Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## model3_cor4      1 30 899.3273 1003.621 -419.6636
## model3_cov_3     2 28 915.7145 1013.056 -429.8572 1 vs 2 20.38723  <.0001
model3.final = gls(templag ~ timelag + hourlag + templast - 1, data = cabin_new, we
ights = varPower(form = ~ timelag), correlation = corARMA(p = 2, form = ~ 1 | hourl
ag), method = "ML")
summary(model3.final)
## Generalized least squares fit by maximum likelihood
##   Model: templag ~ timelag + hourlag + templast - 1
##   Data: cabin_new
##        AIC      BIC    logLik
##   899.3273 1003.621 -419.6636
##
## Correlation Structure: ARMA(2,0)
##  Formula: ~1 | hourlag
##  Parameter estimate(s):
##       Phi1       Phi2
## -0.2398002 -0.2307011
## Variance function:
##  Structure: Power of variance covariate
##  Formula: ~timelag
##  Parameter estimates:
##     power
## 0.3236996
##
## Coefficients:
##                Value Std.Error   t-value p-value
```

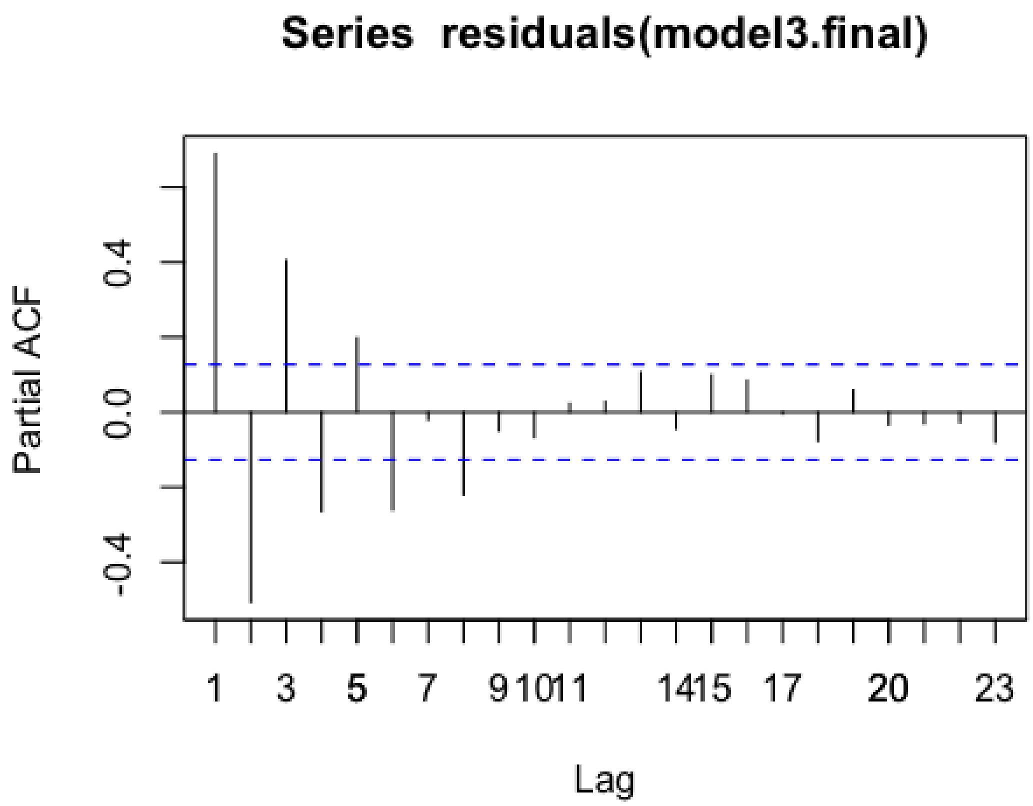
```
## timelag   -0.002055 0.0011058 -1.858006  0.0645
## hourlag0  -0.298947 0.7838516 -0.381383  0.7033
## hourlag1   0.238266 0.7396407  0.322137  0.7477
## hourlag2   0.235831 0.6309076  0.373797  0.7089
## hourlag3   0.765707 0.6123696  1.250399  0.2125
## hourlag4   1.121013 0.6003769  1.867182  0.0632
## hourlag5   1.020008 0.5950295  1.714214  0.0879
## hourlag6   0.740339 0.5915732  1.251475  0.2121
## hourlag7   0.919578 0.5831105  1.577021  0.1163
## hourlag8   6.679777 0.5723190 11.671423  0.0000
## hourlag9  14.404897 0.5690885 25.312226  0.0000
## hourlag10 17.924881 0.7091128 25.277898  0.0000
## hourlag11 18.932655 0.9193130 20.594351  0.0000
## hourlag12 17.791716 1.1572163 15.374581  0.0000
## hourlag13 14.590287 1.3844159 10.538948  0.0000
## hourlag14 10.072381 1.5697047  6.416736  0.0000
## hourlag15  5.551561 1.6813541  3.301840  0.0011
## hourlag16  2.112680 1.7141527  1.232492  0.2191
## hourlag17 -0.294983 1.6800813 -0.175577  0.8608
## hourlag18 -3.915871 1.6075167 -2.435975  0.0157
## hourlag19 -9.232227 1.5061031 -6.129877  0.0000
## hourlag20 -9.723926 1.3352735 -7.282347  0.0000
## hourlag21 -5.459601 1.0919449 -4.999887  0.0000
## hourlag22 -2.415322 0.9301577 -2.596680  0.0101
## hourlag23 -1.018614 0.8398694 -1.212825  0.2265
## templast   0.896943 0.0302844 29.617321  0.0000
##
##  Correlation:
##           timelg horlg0 horlg1 horlg2 horlg3 horlg4 horlg5 horlg6 horlg7 horlg8
## hourlag0  -0.477
## hourlag1  -0.467  0.793
## hourlag2  -0.444  0.817  0.797
## hourlag3  -0.447  0.805  0.785  0.811
## hourlag4  -0.451  0.796  0.776  0.801  0.789
## hourlag5  -0.453  0.789  0.769  0.793  0.782  0.772
## hourlag6  -0.455  0.783  0.764  0.787  0.776  0.767  0.760
## hourlag7  -0.457  0.775  0.756  0.779  0.768  0.759  0.752  0.747
## hourlag8  -0.459  0.766  0.747  0.770  0.759  0.750  0.743  0.738  0.731
## hourlag9  -0.461  0.761  0.742  0.764  0.753  0.745  0.738  0.733  0.726  0.718
## hourlag10 -0.461  0.812  0.792  0.817  0.805  0.795  0.788  0.782  0.774  0.765
## hourlag11 -0.455  0.847  0.826  0.854  0.841  0.830  0.823  0.816  0.808  0.798
## hourlag12 -0.447  0.865  0.844  0.873  0.859  0.848  0.841  0.834  0.825  0.815
## hourlag13 -0.441  0.874  0.853  0.883  0.869  0.857  0.849  0.843  0.834  0.823
## hourlag14 -0.438  0.879  0.857  0.887  0.873  0.862  0.854  0.847  0.838  0.827
## hourlag15 -0.436  0.881  0.859  0.889  0.875  0.864  0.856  0.849  0.840  0.829
## hourlag16 -0.436  0.881  0.859  0.890  0.876  0.864  0.856  0.849  0.840  0.830
## hourlag17 -0.438  0.881  0.858  0.889  0.875  0.864  0.855  0.849  0.840  0.829
## hourlag18 -0.440  0.879  0.857  0.887  0.873  0.862  0.854  0.847  0.838  0.828
## hourlag19 -0.443  0.876  0.855  0.885  0.871  0.859  0.851  0.845  0.836  0.825
## hourlag20 -0.449  0.871  0.849  0.879  0.865  0.854  0.846  0.839  0.830  0.820
```

```
## hourlag21 -0.459  0.857  0.836  0.864  0.851  0.840  0.832  0.826  0.817  0.807
## hourlag22 -0.467  0.840  0.820  0.846  0.833  0.823  0.816  0.810  0.802  0.792
## hourlag23 -0.473  0.826  0.805  0.831  0.818  0.808  0.801  0.795  0.787  0.778
## templast   0.392 -0.891 -0.869 -0.902 -0.887 -0.875 -0.866 -0.859 -0.849 -0.838
##           horlg9 hrlg10 hrlg11 hrlg12 hrlg13 hrlg14 hrlg15 hrlg16 hrlg17 hrlg18
## hourlag0
## hourlag1
## hourlag2
## hourlag3
## hourlag4
## hourlag5
## hourlag6
## hourlag7
## hourlag8
## hourlag9
## hourlag10  0.760
## hourlag11  0.792  0.847
## hourlag12  0.809  0.866  0.905
## hourlag13  0.817  0.875  0.915  0.937
## hourlag14  0.821  0.879  0.921  0.942  0.953
## hourlag15  0.823  0.881  0.923  0.944  0.955  0.961
## hourlag16  0.823  0.882  0.923  0.945  0.956  0.961  0.964
## hourlag17  0.823  0.881  0.922  0.944  0.955  0.961  0.963  0.963
## hourlag18  0.821  0.879  0.920  0.942  0.953  0.959  0.961  0.961  0.960
## hourlag19  0.819  0.877  0.918  0.939  0.950  0.955  0.958  0.958  0.957  0.955
## hourlag20  0.814  0.871  0.911  0.932  0.943  0.948  0.951  0.951  0.950  0.948
## hourlag21  0.802  0.857  0.896  0.916  0.926  0.932  0.934  0.934  0.933  0.931
## hourlag22  0.786  0.840  0.877  0.897  0.907  0.911  0.914  0.914  0.913  0.911
## hourlag23  0.773  0.825  0.861  0.880  0.889  0.894  0.896  0.896  0.895  0.894
## templast  -0.831 -0.892 -0.936 -0.959 -0.971 -0.977 -0.980 -0.980 -0.979 -0.977
##           hrlg19 hrlg20 hrlg21 hrlg22 hrlg23
## hourlag0
## hourlag1
## hourlag2
## hourlag3
## hourlag4
## hourlag5
## hourlag6
## hourlag7
## hourlag8
## hourlag9
## hourlag10
## hourlag11
## hourlag12
## hourlag13
## hourlag14
## hourlag15
## hourlag16
## hourlag17
## hourlag18
```

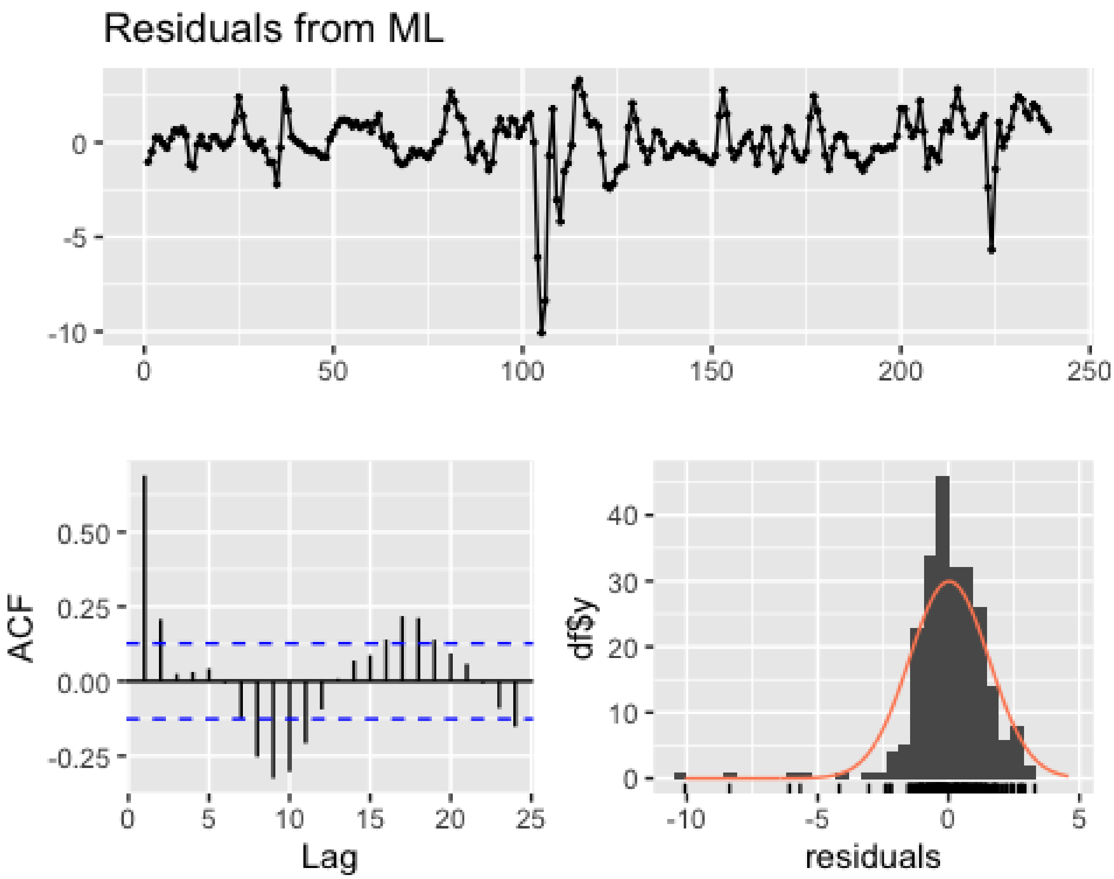
```
## hourlag19
## hourlag20  0.945
## hourlag21  0.929  0.922
## hourlag22  0.909  0.903  0.888
## hourlag23  0.891  0.885  0.871  0.854
## templast  -0.973 -0.966 -0.947 -0.926 -0.907
##
## Standardized residuals:
##          Min           Q1          Med           Q3          Max
## -6.612169171 -0.431450520  0.001305558  0.554027929  2.563041278
##
## Residual standard error: 0.33566
## Degrees of freedom: 239 total; 213 residual
# Autocorrelation of residuals
acf(residuals(model3.final))
```



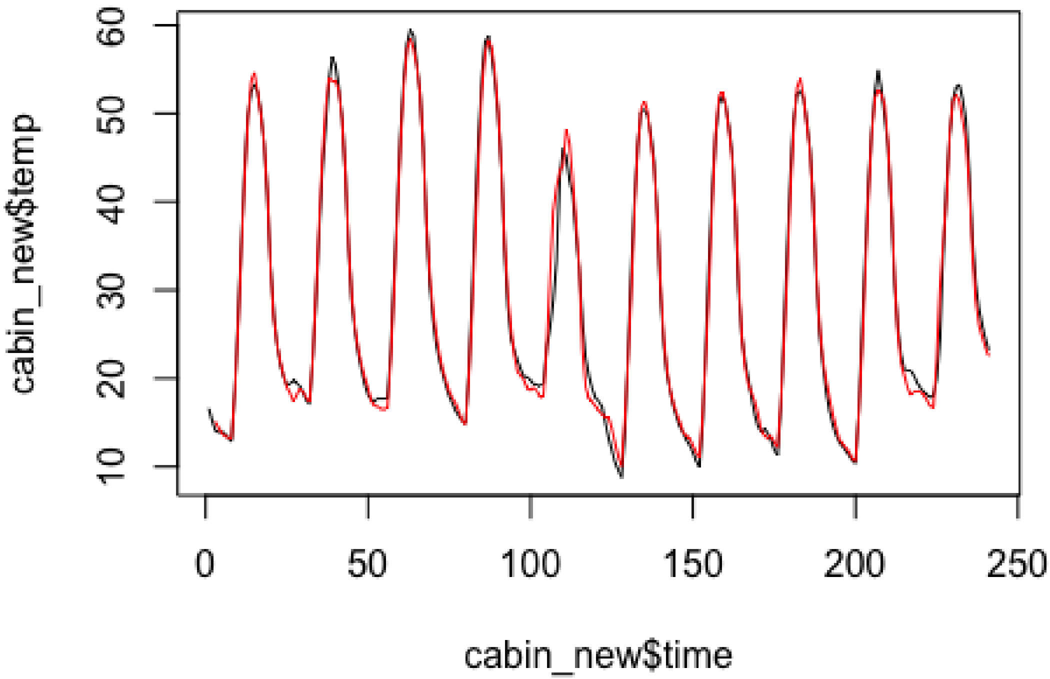

```
pacf(residuals(model3.final))
axis(side = 1, at = seq(0, length(cabin_new$temperature), by = 1))
```




```
checkresiduals(model3.final)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ML
## Q* = 195.73, df = 10, p-value < 2.2e-16
##
## Model df: 0.   Total lags used: 10
plot(cabin_new$temp ~ cabin_new$time, type = "l")
lines(model3.final$fitted ~ timelag, col = "red")
```



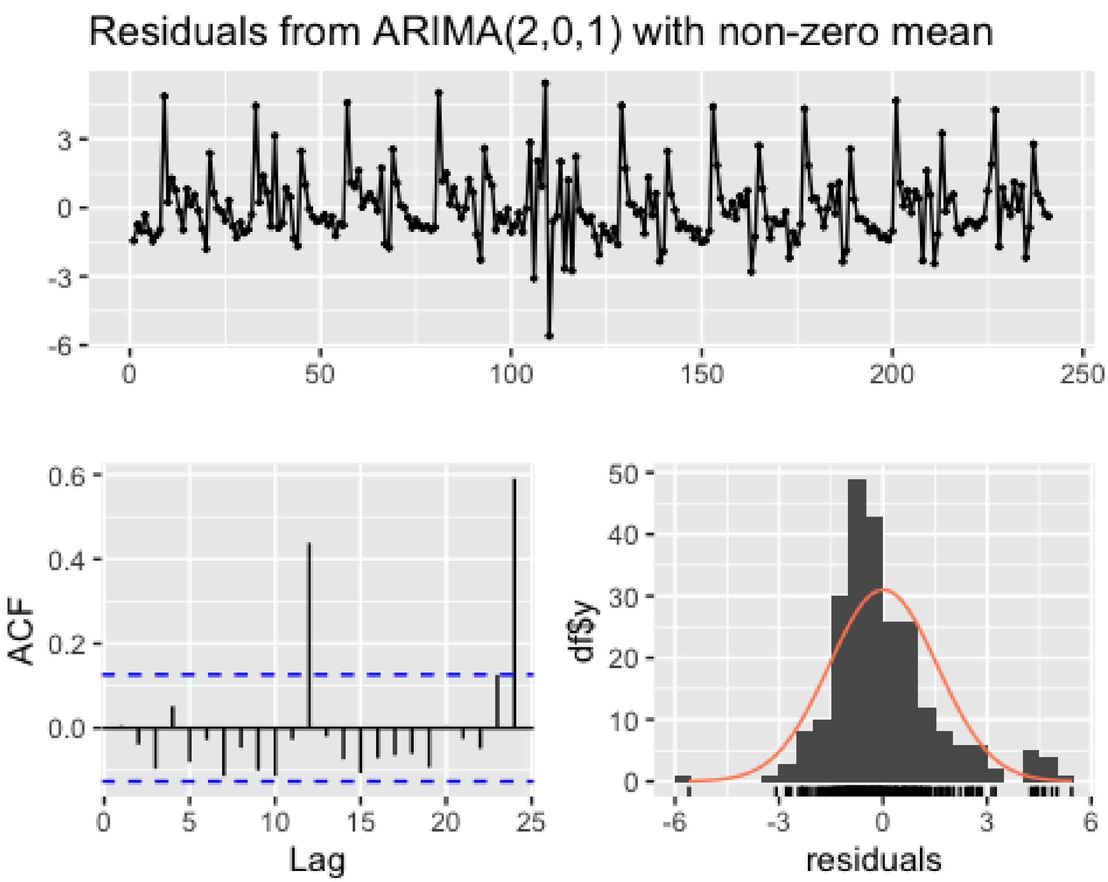

```r
rmse <- sqrt(mean((model3_cor4$residuals)^2))
print(paste("RMSE:", rmse))
## [1] "RMSE: 1.51770631574388"
```

(g) Model 4: Fit a Box-Jenkins ARIMA model to the data. Again you will need to account for any seasonality in the data.

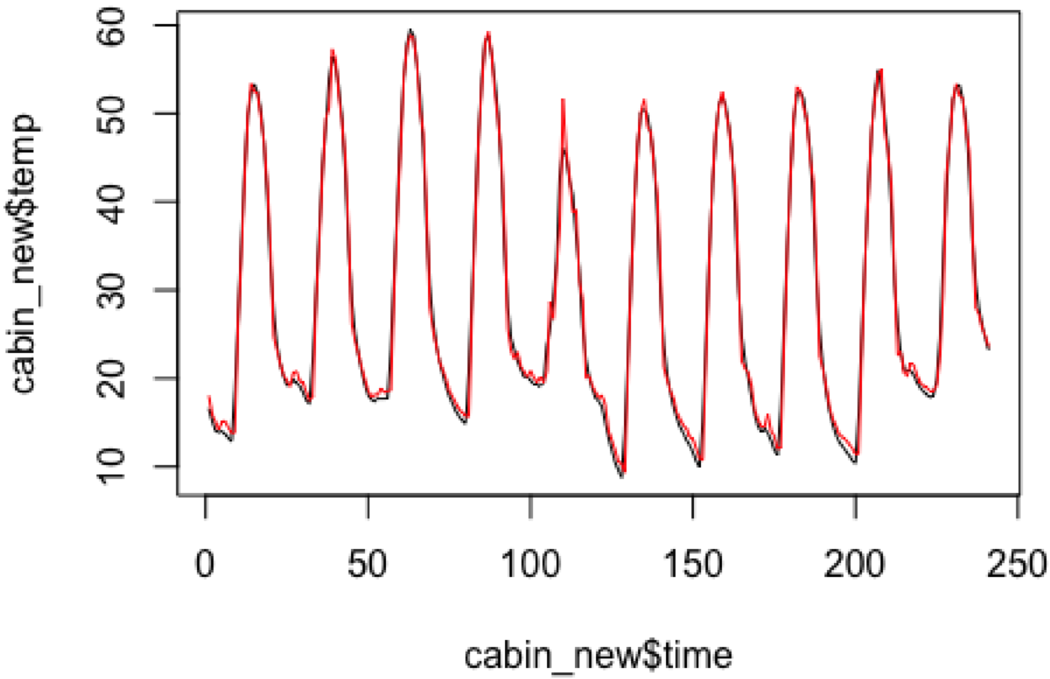
```r
# Fit Box-Jenkins ARIMA model
model4 <- auto.arima(cabin_new$temperature, seasonal = TRUE)
summary(model4)
## Series: cabin_new$temperature
## ARIMA(2,0,1) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ma1     mean
##       1.7943  -0.8828  0.3192  29.8405
## s.e.  0.0307   0.0306  0.0623   1.4863
##
## sigma^2 = 2.448:  log likelihood = -451.12
## AIC=912.24   AICc=912.49   BIC=929.66
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 0.002414351 1.551725 1.132739 -0.6668196 4.973083 0.3442103
```

Lachlan Bassi 22975276

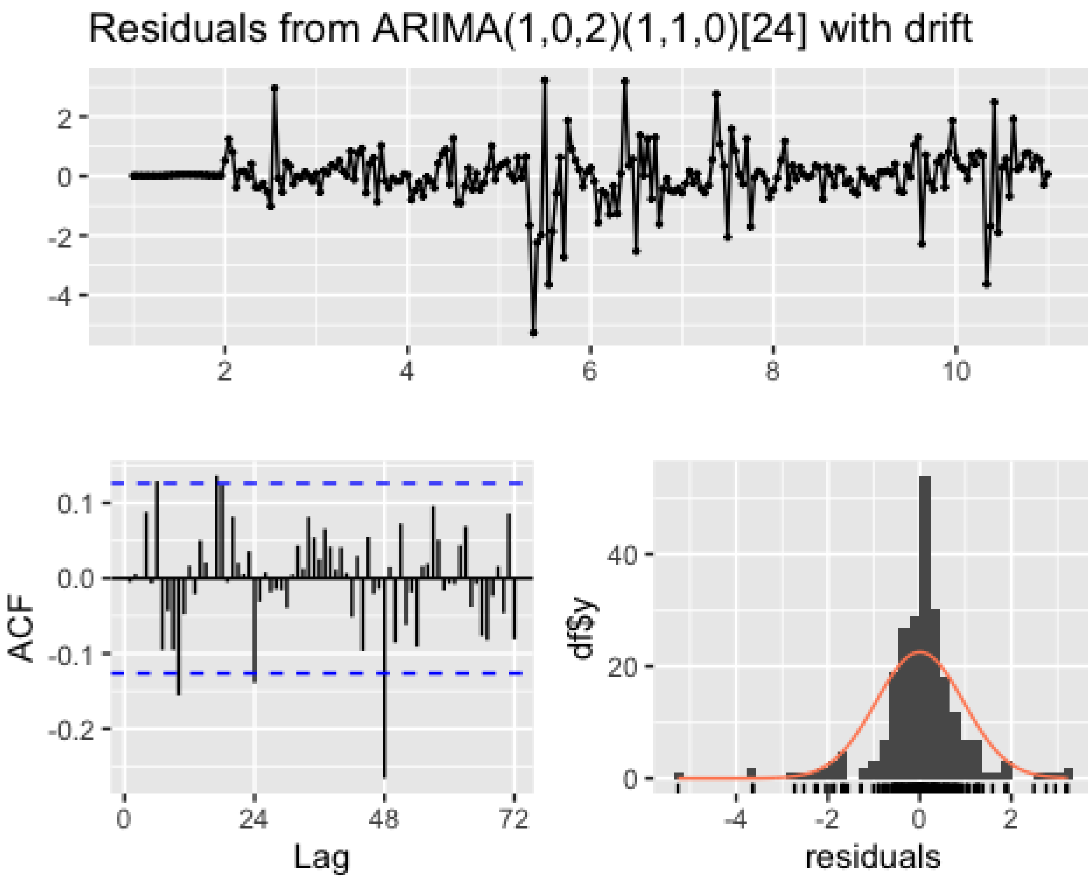
```
##                        ACF1
## Training set 0.00549752
# Check residuals
checkresiduals(model4)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,1) with non-zero mean
## Q* = 14.638, df = 7, p-value = 0.04093
##
## Model df: 3.   Total lags used: 10
plot(cabin_new$temp ~ cabin_new$time, type = "l")
lines(model4$fitted ~ cabin_new$time, col = "red")
```

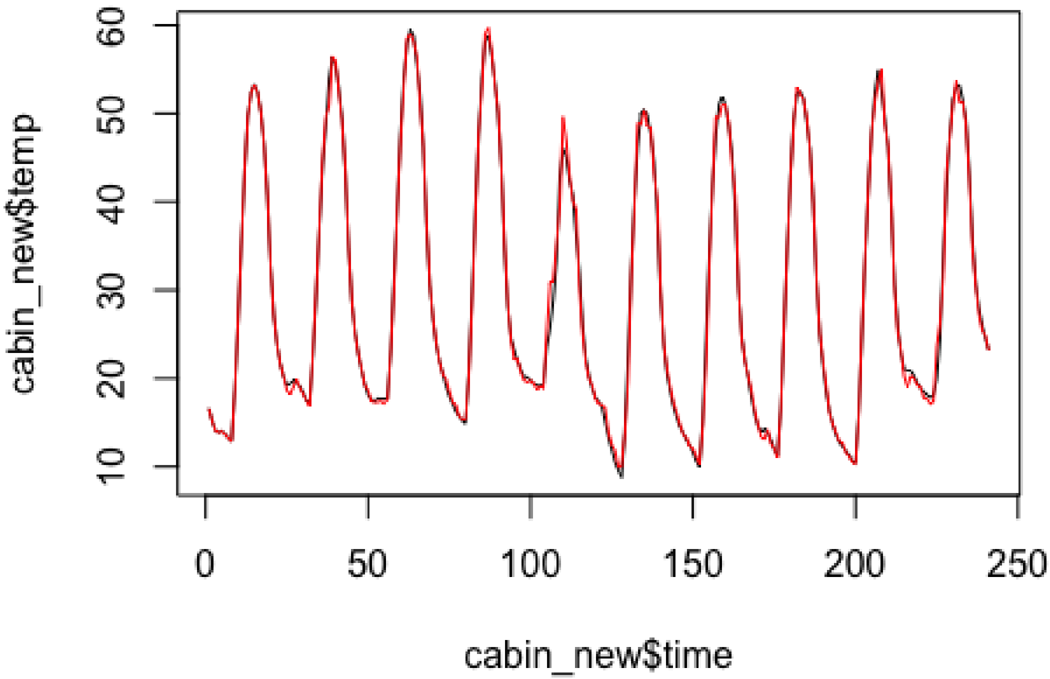



```
tempe <- ts(cabin_new$temperature, frequency = 24)
model4.auto <- auto.arima(tempe, seasonal = TRUE)
summary(model4.auto)
## Series: tempe
## ARIMA(1,0,2)(1,1,0)[24] with drift
##
## Coefficients:
##          ar1     ma1     ma2     sar1   drift
##       0.8855  0.6867  0.2871  -0.4645  0.0100
## s.e.  0.0343  0.0732  0.0646   0.0625  0.0333
##
## sigma^2 = 1.03:  log likelihood = -313.06
## AIC=638.12   AICc=638.52   BIC=658.4
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.002942734 0.9517384 0.5913368 -0.1241812 2.200309 0.2043574
##                      ACF1
## Training set -0.005276432
checkresiduals(model4.auto)
```



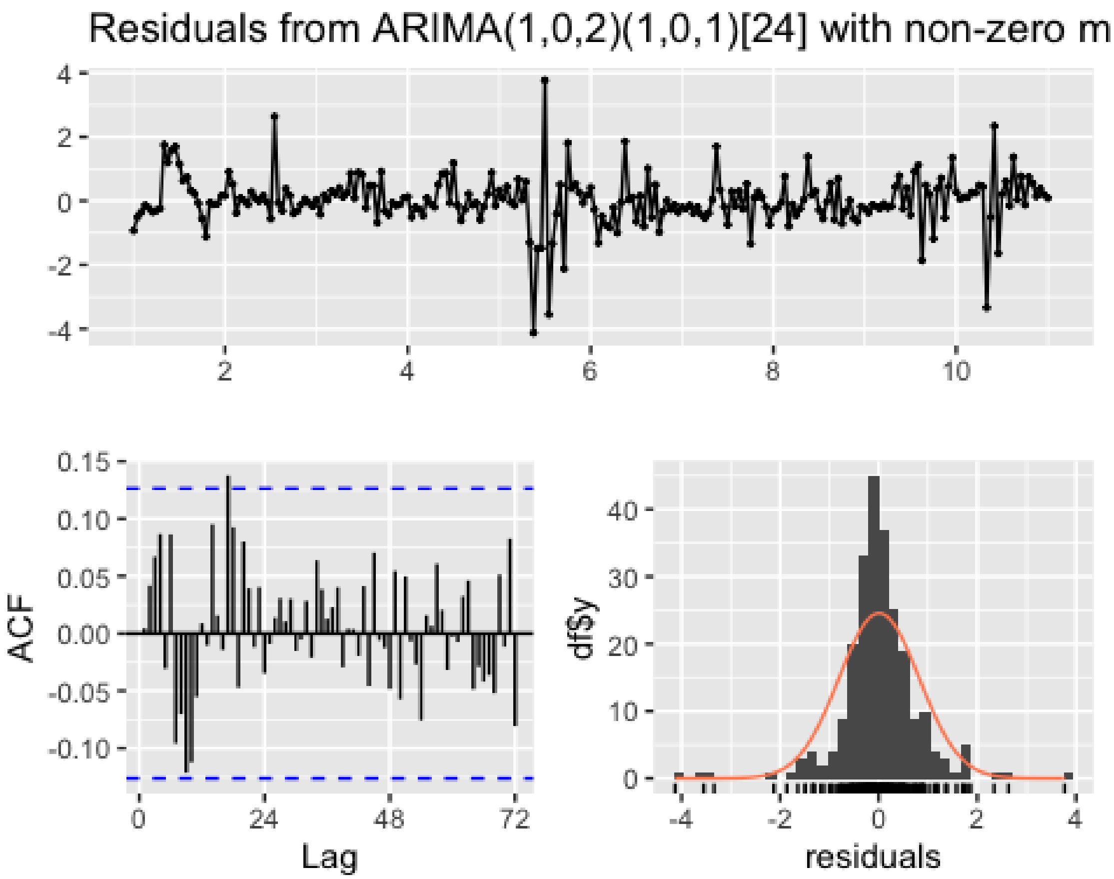

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,2)(1,1,0)[24] with drift
## Q* = 68.192, df = 44, p-value = 0.01115
##
## Model df: 4.   Total lags used: 48
plot(cabin_new$temp ~ cabin_new$time, type = "l")
lines(model4.auto$fitted ~ cabin_new$time, col = "red")
```

Lachlan Bassi 22975276



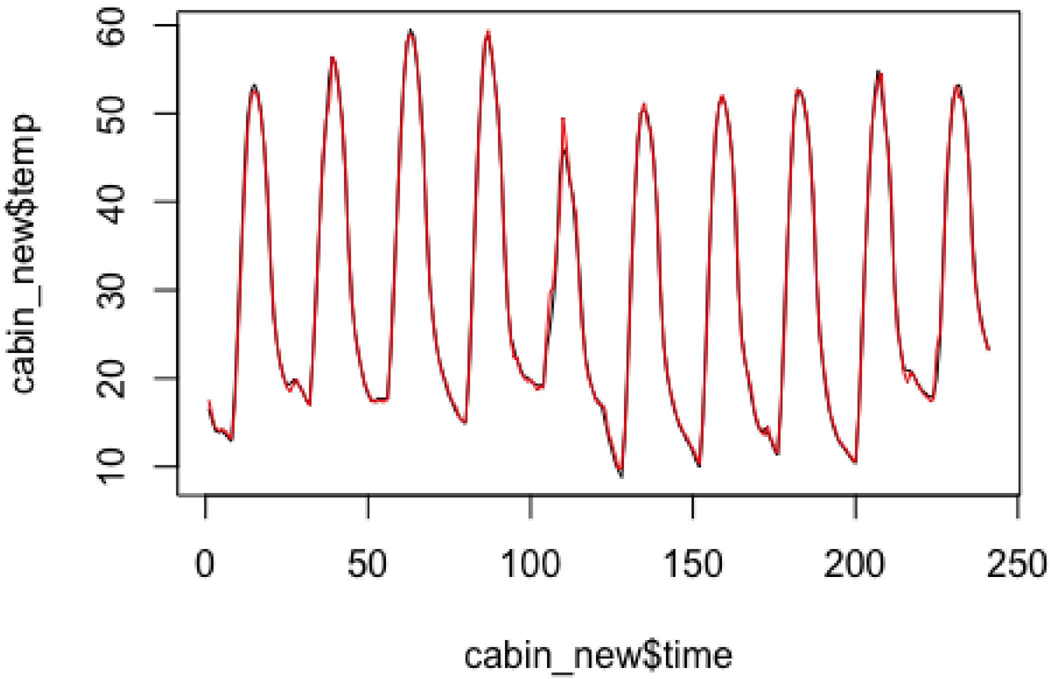

```
# Fit a seasonal ARIMA model
model4 <- Arima(tempe, order = c(1, 0, 2), seasonal = c(1, 0, 1), include.mean = TR
UE)

# Summary of the model
summary(model4)
## Series: tempe
## ARIMA(1,0,2)(1,0,1)[24] with non-zero mean
##
## Coefficients:
##          ar1     ma1     ma2   sar1     sma1     mean
##       0.8994  0.7330  0.2843  1e+00  -0.9721  29.8897
## s.e.  0.0297  0.0712  0.0617  2e-04   0.0465   9.4327
##
## sigma^2 = 0.6769:  log likelihood = -343.46
## AIC=700.92   AICc=701.4   BIC=725.31
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.006096616 0.8124055 0.525711 -0.2446142 1.992179 0.1816781
##                     ACF1
## Training set 0.004518531
checkresiduals(model4)
```

Lachlan Bassi 22975276




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,2)(1,0,1)[24] with non-zero mean
## Q* = 36.927, df = 43, p-value = 0.7309
##
## Model df: 5.   Total lags used: 48
plot(cabin_new$temp ~ cabin_new$time, type = "l")
lines(model4$fitted ~ cabin_new$time, col = "red")
```

Lachlan Bassi 22975276




```r
model4.final = model4
```

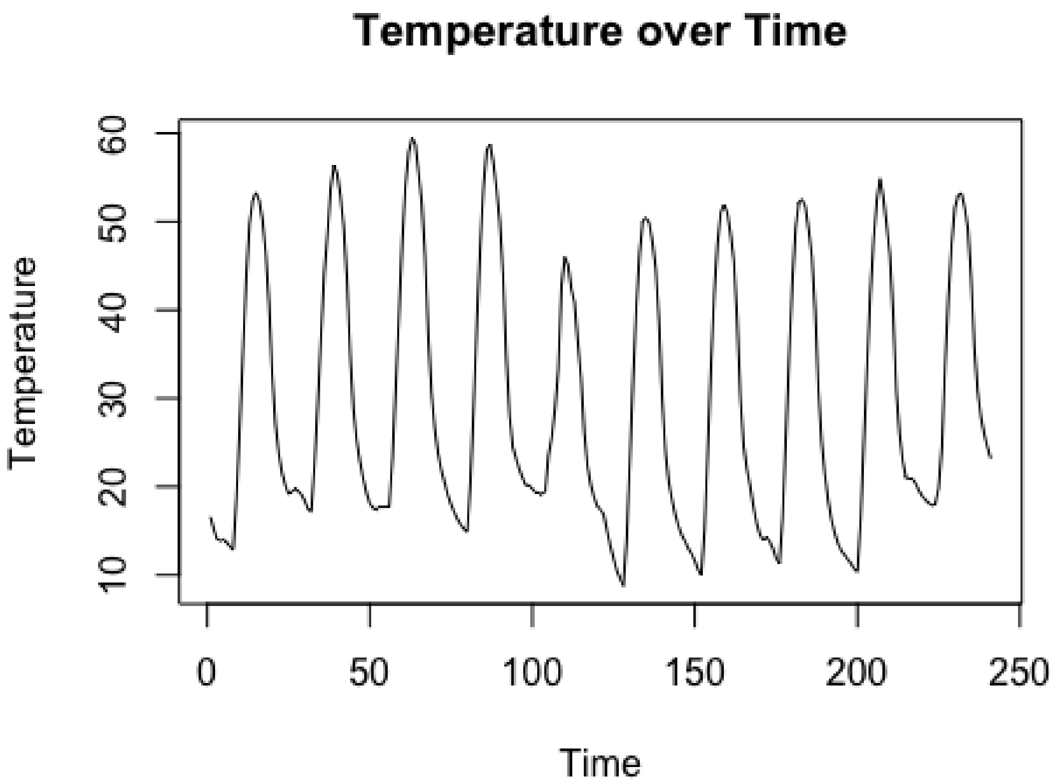
(h) Compare the four models. Discuss any similarities, differences and shortcomings of the models, and select the best model, justifying your selection.

```r
AIC(model4.final, model3.final, model1.final, model2.final)
## Warning in AIC.default(model4.final, model3.final, model1.final, model2.final):
## models are not all fitted to the same number of observations
##              df      AIC
## model4.final  7 700.9171
## model3.final 30 899.3273
## model1.final 27 928.4838
## model2.final  9 702.8199
BIC(model4.final, model3.final, model1.final, model2.final)
## Warning in BIC.default(model4.final, model3.final, model1.final, model2.final):
## models are not all fitted to the same number of observations
##              df       BIC
## model4.final  7  725.3107
## model3.final 30 1003.6212
## model1.final 27 1022.3483
## model2.final  9  734.1830
```

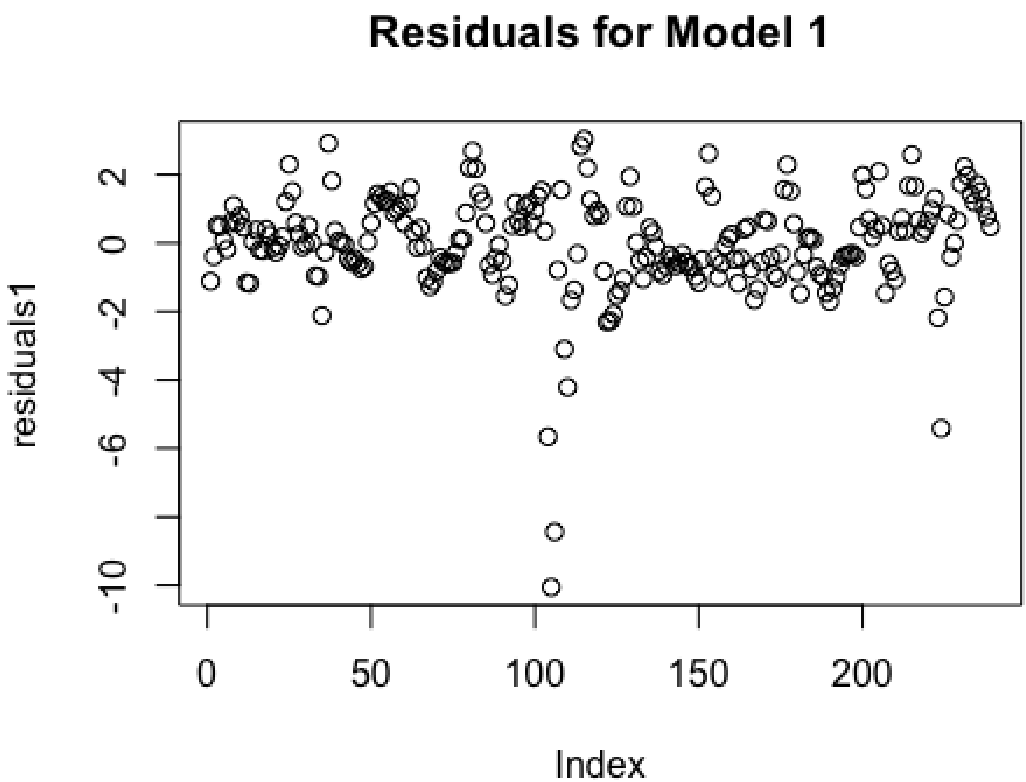
(i) Discuss the behaviour of the temperature in the cabin of the car with respect to time and day. Compare between the four models you have fitted, discussing

Lachlan Bassi 22975276

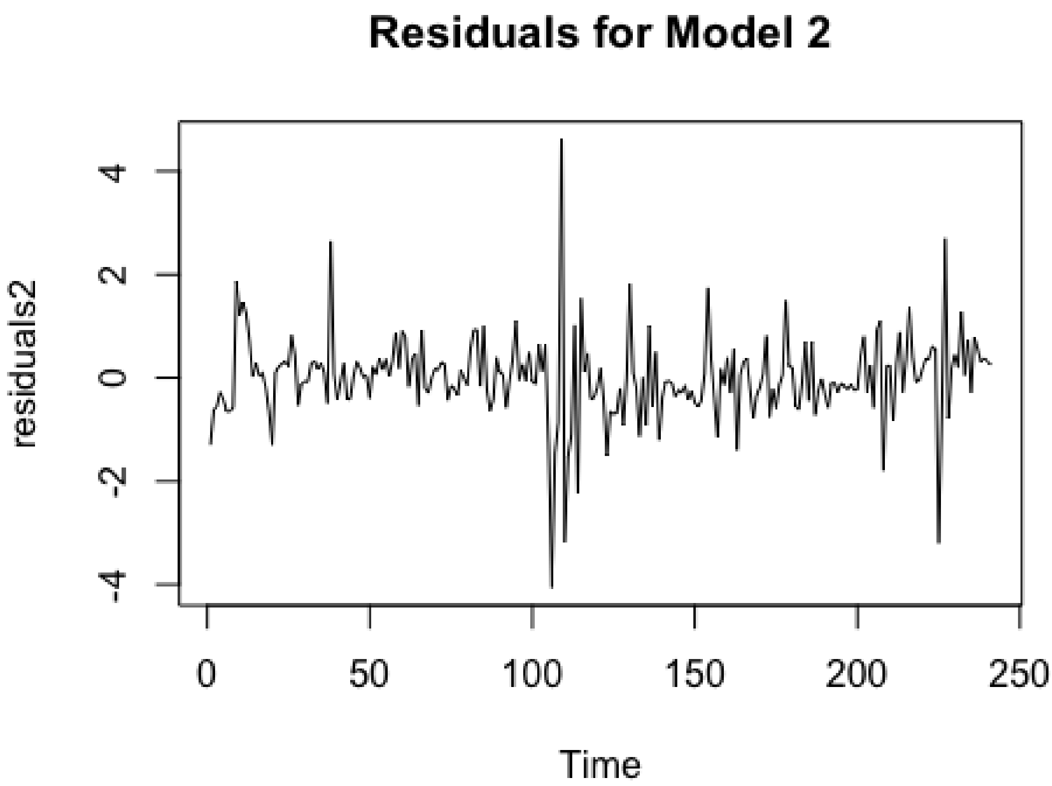
```r
# Plotting temperature against time
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l",
     main = "Temperature over Time", xlab = "Time", ylab = "Temperature")
```




```r
# Residuals for Model 1
residuals1 <- residuals(model1.final)
plot(residuals1, main="Residuals for Model 1")
```

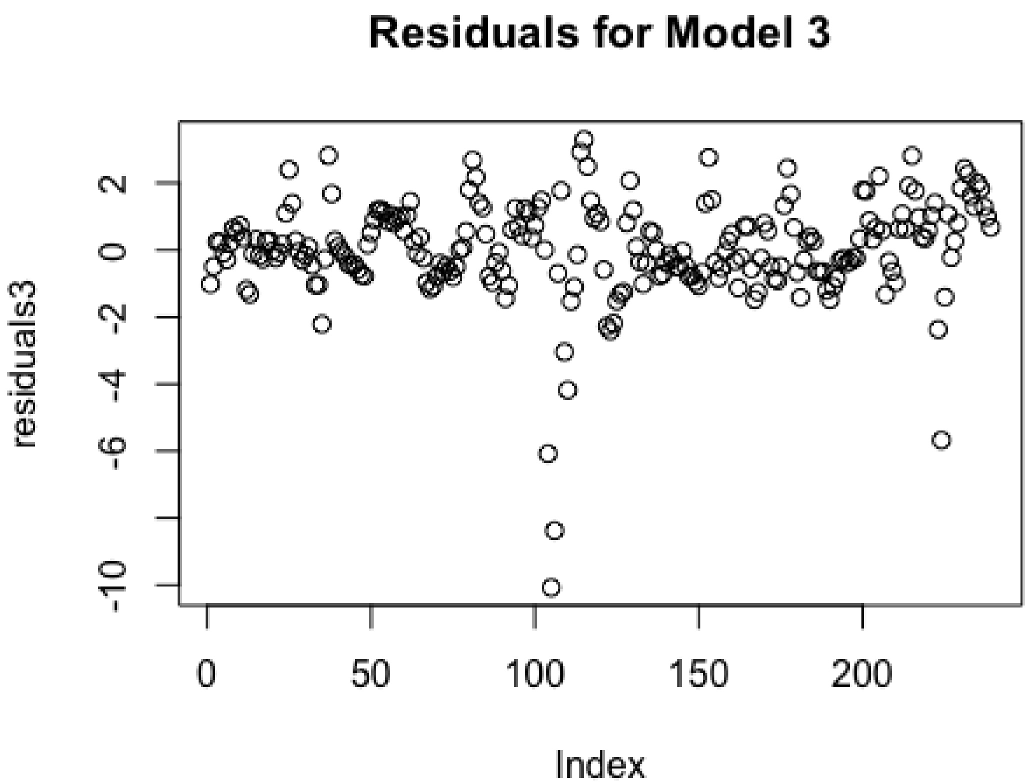



```r
# Residuals for Model 2
residuals2 <- residuals(model2.final)
plot(residuals2, main="Residuals for Model 2")
```

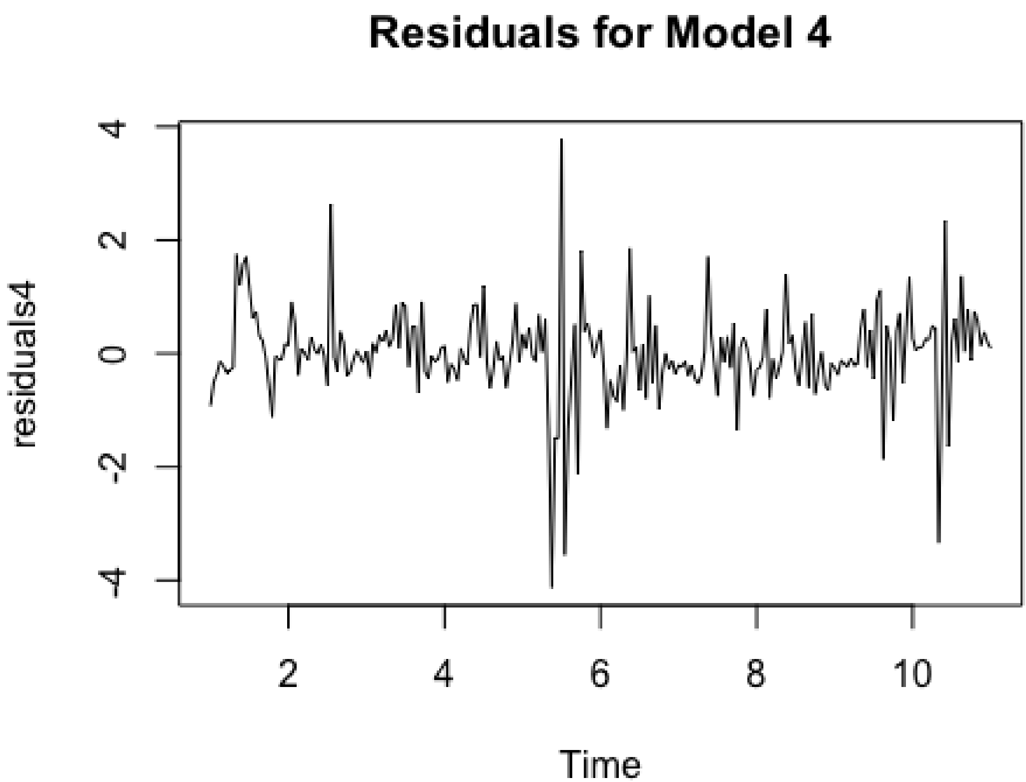



```r
# Residuals for Model 3
residuals3 <- residuals(model3.final)
plot(residuals3, main="Residuals for Model 3")
```

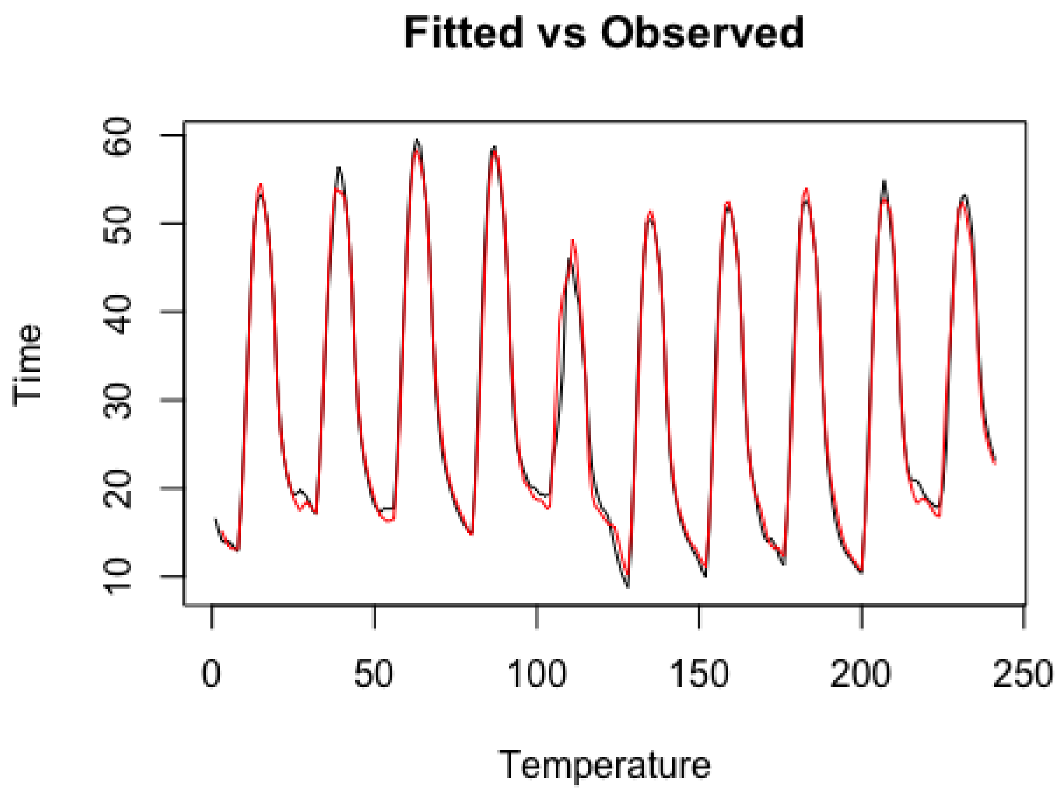
Lachlan Bassi 22975276




```r
# Residuals for Model 4
residuals4 <- residuals(model4.final)
plot(residuals4, main="Residuals for Model 4")
```

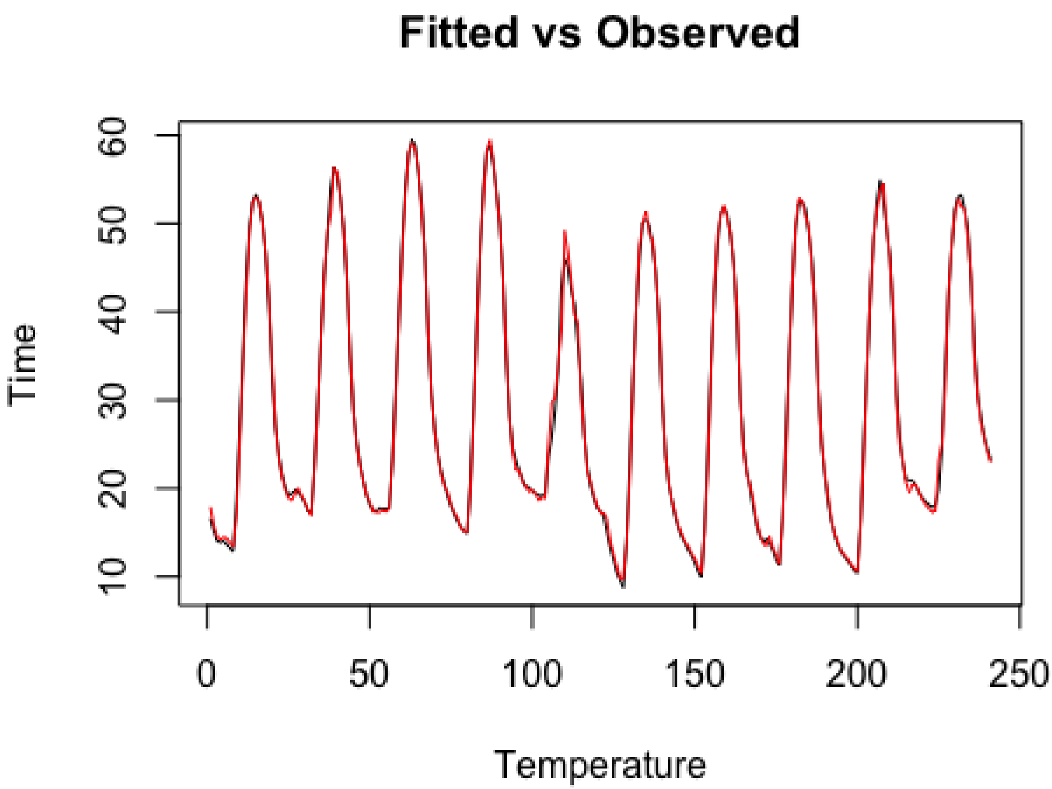



```
# Fitted vs Actual values for Model 1
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l", main = "Fitted vs Obs
erved", xlab = "Temperature", ylab = "Time")
lines(fitted(model1.final) ~ cabin_new$timestamp[-c(1,2)], col = "red")
```



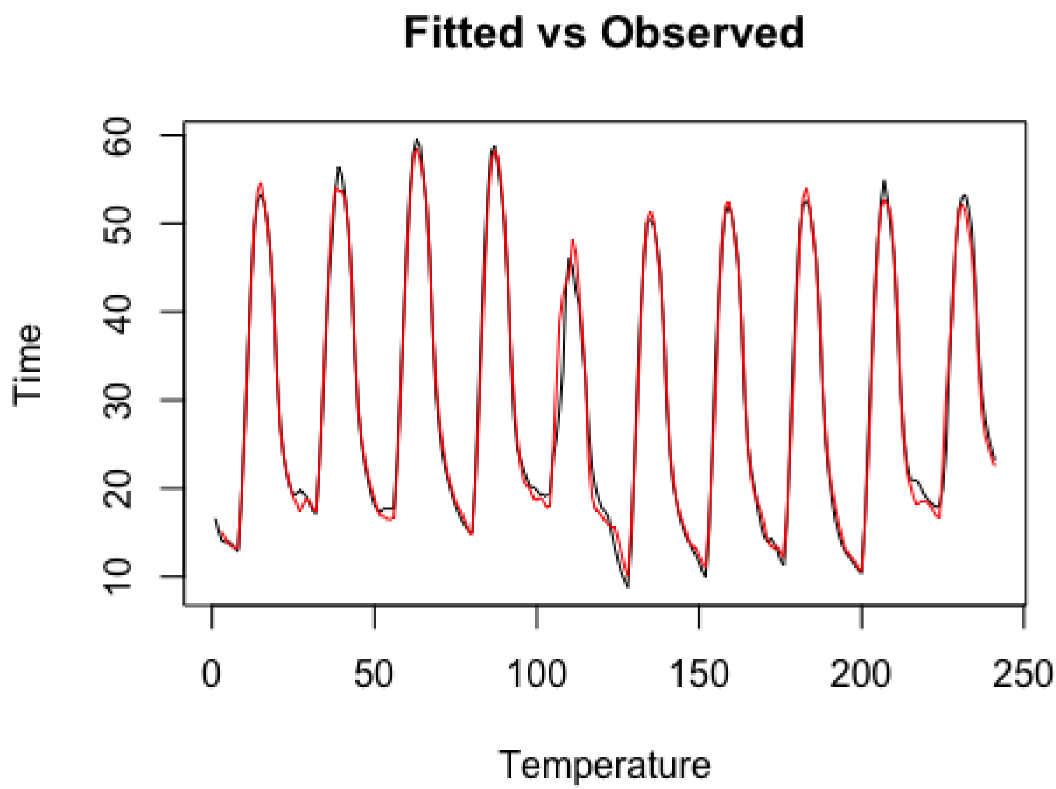

```r
# Fitted vs Actual values for Model 2
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l", main = "Fitted vs Obs
erved", xlab = "Temperature", ylab = "Time")
lines(fitted(model2.final) ~ cabin_new$timestamp, col = "red")
```

Lachlan Bassi 22975276

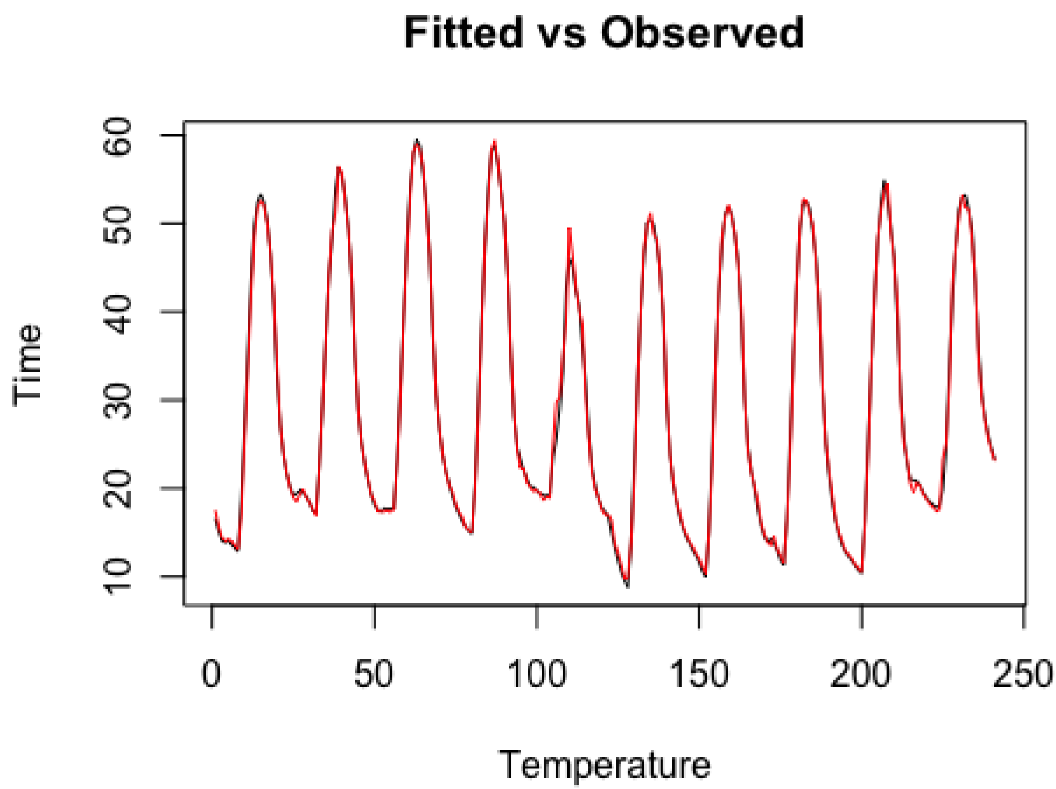



```r
# Fitted vs Actual values for Model 3
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l", main = "Fitted vs Obs
erved", xlab = "Temperature", ylab = "Time")
lines(fitted(model3.final) ~ cabin_new$timestamp[-c(1,2)], col = "red")
```

Lachlan Bassi 22975276




```r
# Fitted vs Actual values for Model 2
plot(cabin_new$temperature ~ cabin_new$timestamp, type = "l", main = "Fitted vs Obs
erved", xlab = "Temperature", ylab = "Time")
lines(fitted(model4.final) ~ cabin_new$timestamp, col = "red")
```




```
summary(model4.final)
## Series: tempe
## ARIMA(1,0,2)(1,0,1)[24] with non-zero mean
##
## Coefficients:
##          ar1     ma1     ma2   sar1     sma1     mean
##       0.8994  0.7330  0.2843  1e+00  -0.9721  29.8897
## s.e.  0.0297  0.0712  0.0617  2e-04   0.0465   9.4327
##
## sigma^2 = 0.6769:  log likelihood = -343.46
## AIC=700.92   AICc=701.4   BIC=725.31
##
## Training set error measures:
##                        ME      RMSE      MAE        MPE     MAPE      MASE
## Training set 0.006096616 0.8124055 0.525711 -0.2446142 1.992179 0.1816781
##                      ACF1
## Training set 0.004518531
# Subset the first 24 unique hours of data
first_24 <- cabin_new[1:24, ]

# Plot data
plot(first_24$temperature ~ first_24$hour_label, type = "l",
     main = "Temperature over Time", xlab = "Time (Hours)", ylab = "Temperature",
```

Lachlan Bassi 22975276

```
     xaxt='n') # This disables the original x-axis
axis(1, at=1:24, labels=first_24$hour_label) # This creates a new x-axis with label
s from 'hour_label'
```