Time Series Modeling for Temperature Variations in Vehicles

Yiyong Duan[a]

[a] *Department of Mathematics and Statistics, University of Western Australia, Crawley, WA 6009, Australia*

Abstract

Heatstroke happens when cabin's temperature increases. The paper investigate how cabin's temperature changes over time. To forecast the future temperature, several time series models are discussed in the paper, including simple linear model, Auto Regressive Moving Average (ARMA) model, General Least Squares (GLS) model, and Auto Regressive Integrated Moving Average (ARIMA) model. Besides, the seasonality and stationarity are considered. The findings conclude that the ARMA and ARIMA models fit the data well, and perform better than other time series models.

1 Introduction

The high temperature in a parked enclosed vehicle may cause children or pets suffer heat stress, and even fatal death[1]. On average, 38 children die each year due to heatstroke after being left in a vehicle since 1998[2]. The temperature on children's body arises three to five times faster than adults', especially when ambient temperature of vehicles reach between 72 and 96℉ [2]. Besides, the estimate of the time of death (post mortem interval or PMI) can be used for legal reference when the tribunal investigate the heatstroke deaths in vehicles [3].

Thus, for preventing the heatstroke cases during the torridity hours, this paper investigate the temperature variations over time in vehicles. The modeling approaches for time series can be classified into simple linear model, Auto Regressive Moving Average (ARMA), General Least Squares (GLS) and Auto Regressive Integrated Moving Average (ARIMA). Among the approaches, the ARIMA model is analyzed within Box-Jenkins framework, which contains model identification, parameter estimation and statistical model checking. In model identification, autocorrelation function (ACF) and partial autocorrelation function (PACF) are introduced to reveal the correlation between two consecutive values. The time series data are influenced by seasonality and stationarity. Seasonality represents the regular changes that happen periodic, while stationarity describes the changes are constant, which does not depend on the time observed.

The data is collected from the temperature record in the cabin of a white Ford Focus Sedan, measured hourly from midnight 9 March 2008 to midnight 18 March 2008. The variables are Date, Time, Temperature. The Time series contains 1440 samples. For convenience, we convert the data as hourly measured, and the data remains 240 samples.

2 Methodology

The methodology of this paper follows Box-Jenkins framework[4], including data inspection, model identification, model assumption, model diagnostics, and model re-evaluation.

Firstly, the data frame contains:

(a) **Date**: chr "09/03/2008" "09/03/2008" "09/03/2008"

(b) **Time_Index**: int 1 2 3 4 5 6 7 8 9 10 ...

(c) **Temp**: num 16.4 15 14 14 14 ...

(d) **Hour** : Factor w/ 24 levels "0","1","2","3",..: 1 2 3...

The time series data may contain different patterns, which includes trends, seasonality, cyclicality and random variability. We follow the steps to identify model:

(a) Check Unusual Trends or Tyclicality. Plot the data with respect to time. From the Figure 1, It has a downtrend around the 100 hours, which can be seen as unpredictable cycles.

(b) Check Seasonality. If the data is seasonal, the changes of data may be regular and predictable that recur every season. From the Figure 1, the hourly temperature change appear stationary, but it has a downtrend around the 100 hours.

(c) Check Stationarity. If the data appears to change with mean, or the time plot looks horizontally, it indicates stationarity. Otherwise, use ACF and PACF to examine further. If both of them decay to or near zero quickly (within one or two lags), it indicates stationarity, otherwise, differencing the data until stationarity is achieved. For non-seasonality data, differencing once and re-evaluate with ACF and PACF. For seasonality data, apply seasonal differences.

After model identification, fit simple linear model as baseline. Improve the model with ARMA, GLS and ARIMA, and then evaluate the performance of these models.

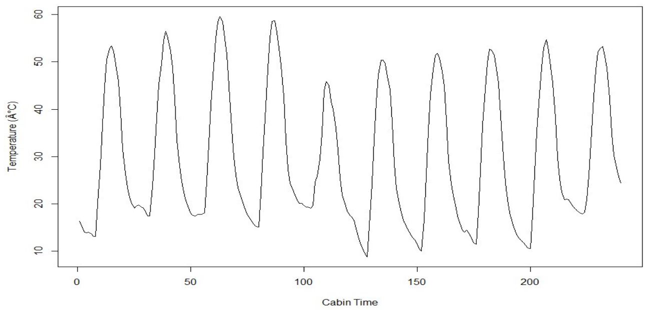


Figure 1. Observed Temperature against Time

3 Results

3.1 Linear model

The linear model choose Temperature as response, Time as trend intercept, Hour as seasonality components. The fitted plot is shown as figure 2, the linear model seems to fit the linear trend well, except for some overfitting and underfitting at the peaks and troughs. Besides, the seasonality is not well captured by the model.

Figure 3 shows the ACF and PACF for model diagnostics. The ACF shows positive alternative decaying pattern at lags 0 to 10, indicating an autoregressive (AR) function. The periodic pattern indicates hourly seasonal components should be included. The PACF are significant at lags 1, 2 and 10, but lag 10 is relatively small which may be overlooked. So we choose seasonal AR(2) model for now.

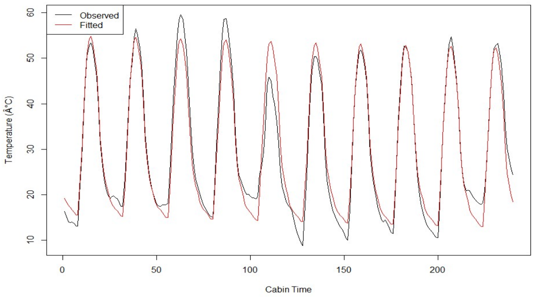


Figure 2. The fitted and observed temperature of linear model

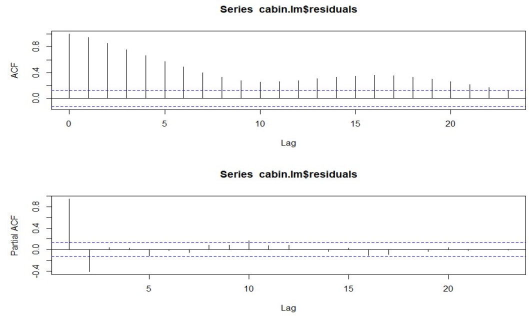


Figure 3. The ACF and PACF of linear model

3.2 ARMA model

The ARMA model error is assumed to be uncorrelated and normally distributed with homogeneous variance.

The ARMA model includes lagged-two temperature as lagged variables. The plot of observed and fitted temperature over time is shown as figure 4. The plot of AR(2) model seems to fit better than previous linear model, especially in the peaks and troughs, and it fits well in non-stationary part, due to seasonal components. Use ACF and PACF plots, diagnostic test and test for normality to check the model.

From the plots of ACF and PACF in figure 5, we can see the ACF has significant spike on lag 3, but it can almost be overlooked. In the PACF plot, although the autocorrelation values at lags 1 and 2 has been eliminated, they exceed the confidence band slightly at lags 3, 4 and 17, which indicates there are some correlation amongst the residuals.

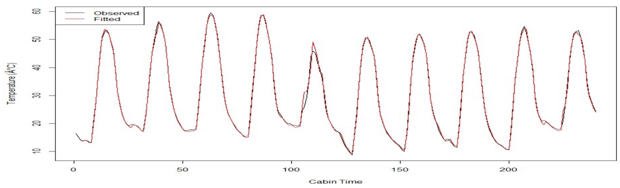


Figure 4. The fitted and observed temperature of AR(2) model

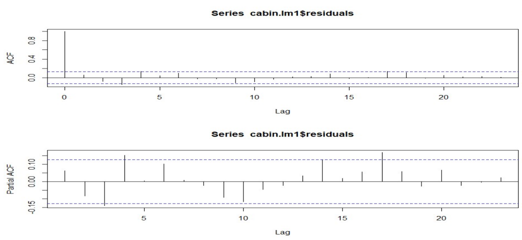


Figure 5. The ACF and PACF of AR(2) model

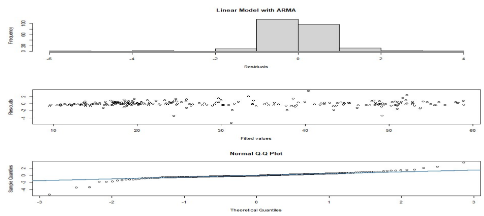


Figure 6. The diagnostic of AR(2) model

In figure 6, the frequency of residuals histogram shows some large absolute residual values. The residuals against fitted values plot shows some heterogeneous variance, and normal probability plot shows dispersion at larger absolute residual values. Taken together, the model may be lack of fit at some extremes.

3.3 Linear model with ARIMA

Based on the AR(2) model, we choose $Arima(2,0,0)$ for optimal model. First we assume the model error is uncorrelated and normally distributed with homogeneous variance, then use ARIMA command to build the model, and do the model diagnostic.

The figure 7 shows the fitted against observed values of ARIMA(2,0,0) model, the performance is quite similar to AR(2) model. The figure 8 shows the model diagnostics.

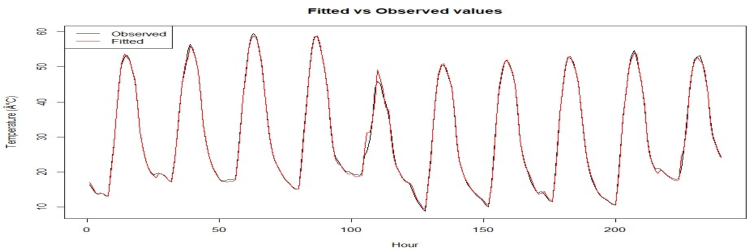


Figure 7 Fitted vs Observed values of ARIMA(2,0,0)

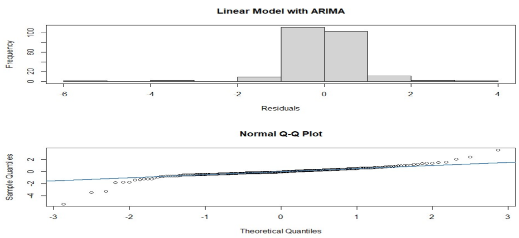


Figure 8 Frequency and Normal Q-Q plot of ARIMA(2,0,0)

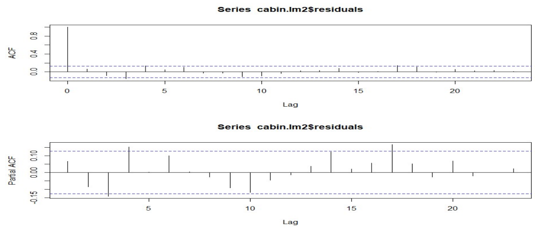


Figure 9 ACF and PACF of ARIMA(2,0,0)

From the plots of ACF and PACF (Figure 9), we can see ACF quickly decays under the confidence boundary, though PACF has significant values on lags 2, 3 and 17, it is adequate for the model. Note the AIC of ARIMA model is 620.52.

3.4 Linear model with Generalized Least Squared

For correlation structure, the ARMA(2,0) is included, and vector 0.5 * c(0.9, -0.8) obtained from autocorrelation of data is chosen as correlation value. For variance function, after comparing with varPower and varIdent, we choose varIdent as its degree freedom is simpler. From the autocorrelation of temperature,

The fitted plot (figure 10) and diagnostic plots (figure 11, 12) are shown below. From the plots, the ACF shows positive alternative decaying pattern over the confidence band, the PACF drops under the significant level after lag 1. It doesn't seem to be better than the AR(2) and $Arima(2,0,0)$ model.

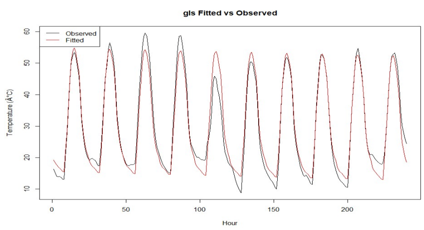


Figure 10 Frequency and Normal Q-Q plot of GLS

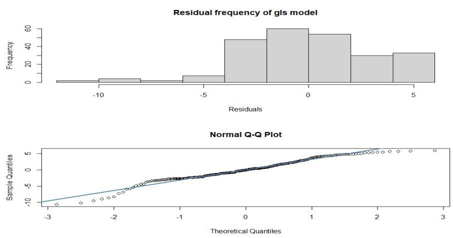


Figure 11 Frequency and Q-Q plot of GLS

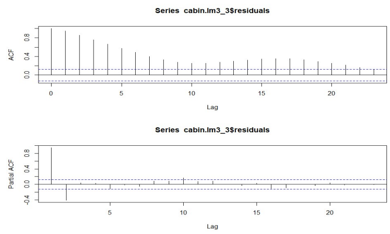
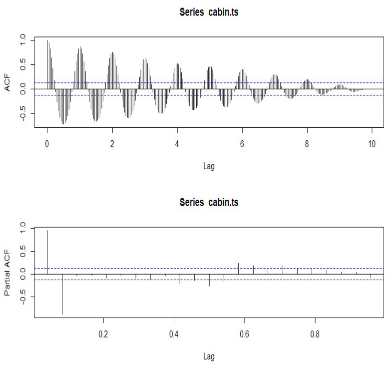


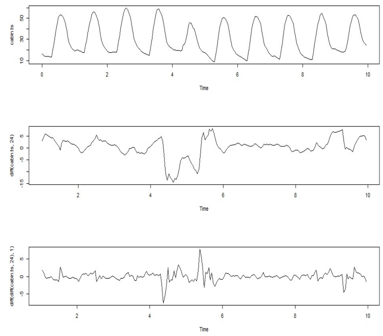
Figure 12 ACF and PACF of GLS

3.5 Box-Jenkins ARIMA model

The model identification is shown in ACF and PACF plot of original data (Figure 13). From the ACF of time series, we can see the autocorrelations slowly decay to near 0 after lags 8, which indicates the mean will change over time, that the process may be non-stationary. Then we include seasonal differences, and use time plot, ACF and PACF plots to re-evaluate the stationarity. The differenced time plot of once, seasonal differences, twice seasonal differences (Figure 14) appear horizontal after differencing. For convenience, we choose seasonal differences for our model.

After differencing (Figure 15), we can see the ACF quickly decays to 0 at lag 1, the PACF also decays to significance bound after lag 1. Then we include the seasonal differences in ARIMA model.





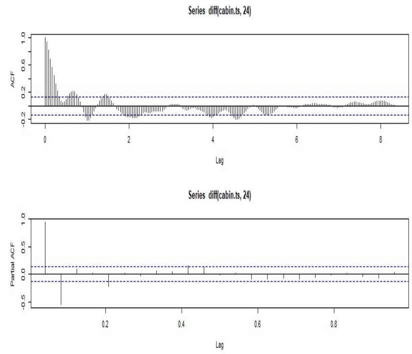
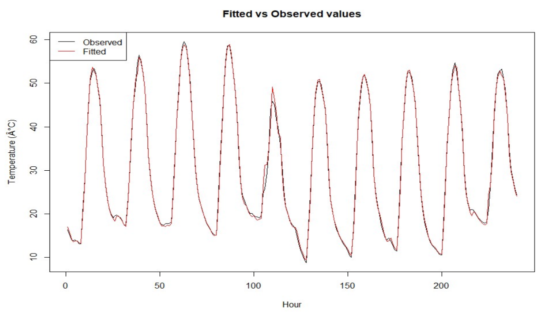


Figure 13. ACF and PACF plot of Cabin data Figure 14. Once, 24 and twice-24 differencing plot Figure 15 ACF and PACF of seasonal differences

The fitted plot and model diagnostics are shown in Figure 16 and 17.



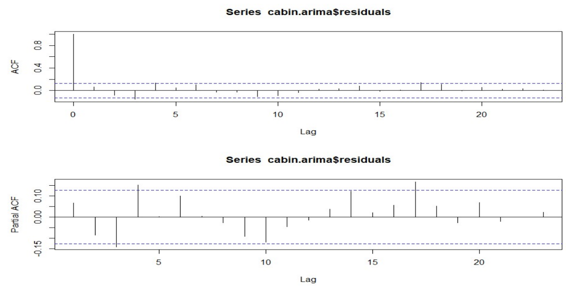


Figure 16. Fitted vs Observed values of ARIMA(2,0,0)(2,24,0)$_{24}$ Figure 17. ACF and PACF of ARIMA(2,0,0)(2,24,0)$_{24}$

No much difference from AR(2) and ARIMA model from fitted plots and ACF, PACF plots, same as the frequency distribution and normal diagnostic plot (see in Appendix Figure A.1). Noted that the AIC of ARMIA(2,0,0)(2,24,0) is also 620.52.

4. Discussion

Among all the time series approaches, linear model with ARMA and ARIMA have the same good performance as integrated ARIMA model with Box-Jenkins structure. The following components may account for its performance.

(a) Seasonality. The ARMA model, GLS linear model and ARIMA model have included seasonality and choose lag 2 as p. However, only GLS model hasn't accounted the seasonal components, probably because of the lack of covariance value influence.

(b) Stationarity. From the time plot and ACF, PACF plots of each differenced data, we can roughly estimate if the data stationarity. In ARMA and ARIMA linear models, we didn't conclude differences for non-stationarity issue, however, their performances are approximately the same as ARIMA(2,0,0)(2,24,0)$_{24}$, which indicates that the data may be stationarity. Therefore, the ADF (Authenticated Dickey-Fuller) test may be applied to test if the data is stationarity or not.

Overall, the Box-Jenkins ARIMA framework is appropriate for modeling and predicting the temperature or else over the time.

References

[1]. A. Guard, S.S. Gallagher. Heat related deaths to young children in parked vehicles: an analysis of 171 fatalities in the United States, 1995–2002, Inj. Prev. 11 (2005) 33–37.

[2]. Jan Null, CCM. Heatstroke Deaths of Children in Vehicles. Noheatstroke: https://www.noheatstroke.org/

[3]. I.R. Dadour, I. Almanjahie, N.D. Fowkes, G. Keady, K. Vijayan. Temperature variations in a parked vehicle, Forensic Science International, Volume 207, Issues 1–3, 2011, Pages 205-211.

[4]. Din, Marilena. (2015). ARIMA by Box Jenkins Methodology for Estimation and Forecasting Models in Higher Education. 10.13140/RG.2.1.1259.6888.

Appendix B

* Noted that some plots in output have been removed if they have occurred in the paper

```
library(formatR)

## Warning: package 'formatR' was built under R version 4.1.3

setwd("E:/Class/STAT4065 MMEM/Final Exam")
cabin <- read.table("cabin.txt", header=TRUE)
colnames(cabin) <- c("Date","Time","Temp","Scale")
```

1. First convert the data to temperature measured hourly, and visualize the data. We choose the time sampling interval as one hour, from 00:11 09/03/2008 to 23:11, 18/03/2008.

```
library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages --------------------------------------- tidyve
rse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse_co
nflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Extract every 6 rows as hour measure
ind <- seq(1, nrow(cabin), 6)
cabin1 <- cabin[ind,]
cabin1$Time_Index <- 1:nrow(cabin1)
cabin1$Hour <- as.factor(rep(c(0:23), each = 1, length = nrow(cabin1)))
# Convert data as time series for differencing
cabin.ts <- ts(cabin1$Temp, start = 0, freq = 24)
# Plot
plot(cabin1$Temp ~ cabin1$Time_Index, type='l', xlab = 'Cabin Time', yla
b = "Temperature (Â°C)")
```

From the time plot, the hourly temperature change is stationary, but it has a downtrend around the 100 hours, which appears non-stationarity. For further stationarity check, we need to examine the ACF and PACF test.

```
str(cabin1)

## 'data.frame':    240 obs. of  6 variables:
##  $ Date      : chr  "09/03/2008" "09/03/2008" "09/03/2008" "09/03/200
8" ...
##  $ Time      : chr  "00:11" "01:11" "02:11" "03:11" ...
##  $ Temp      : num  16.4 15 14 14 14 ...
##  $ Scale     : chr  "Â°C" "Â°C" "Â°C" "Â°C" ...
##  $ Time_Index: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Hour      : Factor w/ 24 levels "0","1","2","3",..: 1 2 3 4 5 6 7 8
 9 10 ...
```

2. Fit a linear model to the data.

```
cabin.lm <- lm(Temp ~ Time_Index + Hour - 1, data = cabin1)
summary(cabin.lm)

##
## Call:
## lm(formula = Temp ~ Time_Index + Hour - 1, data = cabin1)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5914  -2.1054  -0.0453   2.1538   5.9092
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## Time_Index -0.011870   0.003072  -3.864 0.000147 ***
## Hour0      19.276468   1.090041  17.684  < 2e-16 ***
## Hour1      18.407438   1.090988  16.872  < 2e-16 ***
## Hour2      17.730109   1.091943  16.237  < 2e-16 ***
## Hour3      17.271279   1.092906  15.803  < 2e-16 ***
## Hour4      16.786849   1.093877  15.346  < 2e-16 ***
## Hour5      16.240120   1.094856  14.833  < 2e-16 ***
## Hour6      15.673590   1.095842  14.303  < 2e-16 ***
## Hour7      15.633561   1.096836  14.253  < 2e-16 ***
## Hour8      21.806631   1.097838  19.863  < 2e-16 ***
## Hour9      29.201801   1.098847  26.575  < 2e-16 ***
## Hour10     37.640572   1.099864  34.223  < 2e-16 ***
## Hour11     45.171542   1.100889  41.032  < 2e-16 ***
## Hour12     51.093412   1.101921  46.368  < 2e-16 ***
## Hour13     54.409683   1.102961  49.331  < 2e-16 ***
## Hour14     55.023253   1.104008  49.840  < 2e-16 ***
## Hour15     53.593923   1.105063  48.499  < 2e-16 ***
## Hour16     50.776894   1.106126  45.905  < 2e-16 ***
## Hour17     47.500464   1.107196  42.902  < 2e-16 ***
## Hour18     40.890334   1.108273  36.896  < 2e-16 ***
## Hour19     32.705905   1.109358  29.482  < 2e-16 ***
## Hour20     27.627075   1.110451  24.879  < 2e-16 ***
## Hour21     24.704645   1.111550  22.225  < 2e-16 ***
## Hour22     22.715016   1.112658  20.415  < 2e-16 ***
## Hour23     21.329686   1.113772  19.151  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.28 on 215 degrees of freedom
## Multiple R-squared:  0.9913, Adjusted R-squared:  0.9903
## F-statistic: 982.1 on 25 and 215 DF,  p-value: < 2.2e-16

# Plot the trend of temperature with time
plot(cabin1$Temp ~ cabin1$Time_Index, type = "l", xlab = 'Cabin Time', y
lab = 'Temperature (Â°C)')
lines(cabin.lm$fitted.values ~ cabin1$Time_Index, col = "red")
legend("topleft", legend = c("Observed", "Fitted"), lty = c(1,1), col =
c("black", "red"))
```

From the summary, all the coefficients are significant. From the plot, the model seems to fit the linear trend well, except for overfitting and underfitting at the peaks and troughs. Besides, the seasonality is not well captured by the model.

3. Auto Regressive Moving Averages Time Series Modeling

Use ACF and PACF function to determine what type of series we have.

```
oldpar <- par(mfrow=c(2,1))
acf(cabin.lm$residuals)
pacf(cabin.lm$residuals)
```

```
par(oldpar)
```

- The ACF shows positive alternative decaying pattern at lags 0 to 10, indicating an autoregressive (AR) function. The periodic pattern indicates hourly seasonal components should be included. Use PACF to determine the order of AR term.
- The PACF are significant at lags 1, 2 and 10, but lag 10 is relatively small which may be overlooked. So we choose seasonal AR(2) model for now.

AR(2) model

```
library(Hmisc)

## Warning: package 'Hmisc' was built under R version 4.1.3

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units

# Create lagged variables
cabin1$Temp1 <- Lag(cabin1$Temp)
cabin1$Temp2 <- Lag(cabin1$Temp, 2)
# Fit a linear model with AR(2) term
cabin.lm1 <- lm(Temp ~ Time_Index + Hour + Temp1 + Temp2 -1, data = cabi
n1)
summary(cabin.lm1)
```

```
##
## Call:
## lm(formula = Temp ~ Time_Index + Hour + Temp1 + Temp2 - 1, data = cab
in1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.3693 -0.3162 -0.0547  0.3462  3.5814
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## Time_Index -0.0006784  0.0008181  -0.829 0.407917
## Hour0       0.8795161  0.4885632   1.800 0.073256 .
## Hour1       0.9192643  0.4685492   1.962 0.051085 .
## Hour2       0.8804148  0.4388027   2.006 0.046090 *
## Hour3       0.9617638  0.4270990   2.252 0.025362 *
## Hour4       0.7991972  0.4183963   1.910 0.057471 .
## Hour5       0.7196309  0.4125360   1.744 0.082544 .
## Hour6       0.6969500  0.4064222   1.715 0.087842 .
## Hour7       1.1982294  0.3996563   2.998 0.003042 **
## Hour8       7.1472074  0.3938861  18.145  < 2e-16 ***
## Hour9       5.6636533  0.5612333  10.091  < 2e-16 ***
## Hour10      6.5569063  0.6811273   9.627  < 2e-16 ***
## Hour11      5.6519271  0.8147793   6.937 4.83e-11 ***
## Hour12      4.9591202  0.8939537   5.547 8.63e-08 ***
## Hour13      3.5188383  0.9558231   3.681 0.000295 ***
## Hour14      2.3155836  0.9905102   2.338 0.020336 *
## Hour15      1.6533860  1.0088312   1.639 0.102722
## Hour16      1.1923426  1.0079432   1.183 0.238162
## Hour17      1.2483209  0.9848549   1.268 0.206367
## Hour18     -2.0596149  0.9402980  -2.190 0.029592 *
## Hour19     -2.3860166  0.9292292  -2.568 0.010928 *
## Hour20      0.9960477  0.8711857   1.143 0.254200
## Hour21      1.2955648  0.6829853   1.897 0.059205 .
## Hour22      0.9763300  0.5749817   1.698 0.090978 .
## Hour23      0.9941794  0.5238201   1.898 0.059069 .
## Temp1       1.4350688  0.0598108  23.993  < 2e-16 ***
## Temp2      -0.4968542  0.0602084  -8.252 1.67e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8287 on 211 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9994
## F-statistic: 1.434e+04 on 27 and 211 DF,  p-value: < 2.2e-16
```

```
# Plot the observed values
plot(cabin1$Temp ~ cabin1$Time_Index, type = "l", xlab = 'Cabin Time', y
lab = 'Temperature (Â°C)')
# Plot the fitted lines without the first two hours.
```

```
lines(cabin.lm1$fitted.values ~ cabin1$Time_Index[-c(1,2)], col = "red
")
legend("topleft", legend = c("Observed", "Fitted"), lty = c(1,1), col =
c("black", "red"))
```

From the summary, the time trend coefficients are out of significance bound, as well as some Hour coefficients. However, the plot of AR(2) model seems to fit better than previous linear model, especially in the peaks and troughs, and it fits well in non-stationary part, due to seasonal components.

Model Diagnostics and Assumptions

The model error is assumed to be uncorrelated and normally distributed with homogeneous variance. Use ACF and PACF plots, diagnostic test and test for normality to check the model.

```
# ACF and PACF plots
oldpar <- par(mfrow=c(2,1))
acf(cabin.lm1$residuals)
pacf(cabin.lm1$residuals)
```

```
par(oldpar)
```

The autocorrelation of ACF has significant spike on lag 3, but it can almost be overlooked. In the PACF plot, although the autocorrelation values at lags 1 and 2 has been eliminated, they exceed the confidence band slightly at lags 3, 4 and 17, which indicates there are some correlation amongst the residuals.

```
# Histogram of Residuals frequency
oldpar <- par(mfrow=c(3,1))
hist(cabin.lm1$residuals, xlab = "Residuals", main = "Linear Model with
ARMA")
# Residuals vs fitted values plot
plot(cabin.lm1$residuals~ cabin.lm1$fitted.values, xlab="Fitted values
", ylab = "Residuals")
# Normality distribution plot
qqnorm(cabin.lm1$residuals)
qqline(cabin.lm1$residuals, col = "steelblue", lwd = 2)
```

```
par(oldpar)
```

The frequency of residuals histogram shows some large absolute residual values. The residuals against fitted values plot shows some heterogeneous variance, and normal probability plot shows dispersion at larger absolute residual values. Taken together, the model may be lack of fit at some extremes, and stationarity is implied.

Arima model

For $Arima(p, d, q)(P, D, Q)_{24}$ model, based on the AR(2) model, seasonality and stationarity, we choose $Arima(2,0,0)(2,0,0)_{24}$ for optimal model. First we assume the model error is uncorrelated and normally distributed with homogeneous variance, then use Arima command to build the model, and do the model diagnostic.

```
library(fastDummies)

## Warning: package 'fastDummies' was built under R version 4.1.3

library(forecast)

## Warning: package 'forecast' was built under R version 4.1.3

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

# First set up the Hour indicator variables as seasonality
cabin2 <- dummy_cols(cabin1, select_columns= "Hour")

# Build Exogenous Regressors
xregs <- as.matrix(data.frame(x1 = cabin2$Time_Index, x2 = cabin2$Hour_
0,
  x3 = cabin2$Hour_1, x4 = cabin2$Hour_2,
  x5 = cabin2$Hour_3, x6 = cabin2$Hour_4, x7 = cabin2$Hour_5,
  x8 = cabin2$Hour_6, x9 = cabin2$Hour_7, x10 = cabin2$Hour_8,
  x11 = cabin2$Hour_9, x12 = cabin2$Hour_10, x13 = cabin2$Hour_11,
  x14 = cabin2$Hour_12, x15 = cabin2$Hour_13, x16 = cabin2$Hour_14,
  x17 = cabin2$Hour_15, x18 = cabin2$Hour_16, x19 = cabin2$Hour_17,
  x20 = cabin2$Hour_18, x21 = cabin2$Hour_19, x22 = cabin2$Hour_20,
  x23 = cabin2$Hour_21, x24 = cabin2$Hour_22, x25 = cabin2$Hour_23))

# Build linear model with ARIMA
cabin.lm2 <- Arima(cabin2$Temp, order = c(2,0,0),  xreg = xregs, includ
e.constant = FALSE)
(x <- as.matrix(coef(cabin.lm2), ncol = 4))

##              [,1]
## ar1  1.433447604
## ar2 -0.494175493
## x1  -0.005297365
## x2  19.081570170
## x3  18.157203064
## x4  17.419251792
```

```
## x5  16.900266234
## x6  16.358928584
## x7  15.759672983
## x8  15.145245402
## x9  15.061722784
## x10 21.195249046
## x11 28.554294963
## x12 36.959800544
## x13 44.459837892
## x14 50.352584205
## x15 53.641037262
## x16 54.227588378
## x17 52.771539187
## x18 49.927604478
## x19 46.623586310
## x20 39.984710116
## x21 31.769855804
## x22 26.658376245
## x23 23.700455665
## x24 21.671816467
## x25 20.243235518
```

```
summary(cabin.lm2)
```

```
## Series: cabin2$Temp
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2       x1       x2       x3       x4       x5
x6
##       1.4334  -0.4942  -0.0053  19.0816  18.1572  17.4193  16.9003  1
6.3589
## s.e.  0.0559   0.0566   0.0111   1.6613   1.6628   1.6638   1.6648
1.6660
##           x7       x8       x9      x10      x11      x12      x13
x14
##       15.7597  15.1452  15.0617  21.1952  28.5543  36.9598  44.4598  5
0.3526
## s.e.   1.6676   1.6696   1.6720   1.6748   1.6778   1.6811   1.6844
1.6877
##          x15      x16      x17      x18      x19      x20      x21
x22
##       53.6410  54.2276  52.7715  49.9276  46.6236  39.9847  31.7699  2
6.6584
## s.e.   1.6907   1.6932   1.6951   1.6961   1.6959   1.6943   1.6911
1.6862
##          x23      x24      x25
##       23.7005  21.6718  20.2432
## s.e.   1.6800   1.6733   1.6681
##
```

```
## sigma^2 = 0.6844:  log likelihood = -282.26
## AIC=620.52   AICc=628.22   BIC=717.98
##
## Training set error measures:
##                       ME      RMSE       MAE        MPE     MAPE      MAS
E
## Training set 0.007526434 0.7793551 0.4908785 -0.1203926 1.899034 0.14
91637
##                    ACF1
## Training set 0.06723037
```

Diagnostic plots

```
# Residuals vs fitted values plot
plot(cabin1$Temp ~ cabin1$Time_Index, type = 'l', xlab = "Hour", ylab="T
emperature (Â°C)", main= "Fitted vs Observed values")
lines(cabin.lm2$fitted ~ cabin1$Time_Index, col = "red")
legend("topleft", legend = c("Observed", "Fitted"), lty = c(1,1), col =
c("black", "red"))
```

```
# Histogram of Residuals frequency
oldpar <- par(mfrow=c(2,1))
hist(cabin.lm2$residuals, xlab = "Residuals", main = "Linear Model with
ARIMA")
box()
# Normality distribution plot
qqnorm(cabin.lm2$residuals)
qqline(cabin.lm2$residuals, col = "steelblue", lwd = 2)
```

```
par(oldpar)
```

The fitted vs observed values plot and the diagnostic plots of $Arima(2,0,0)$ model are approximately the same as AR(2) model.

```
oldpar <- par(mfrow=c(2,1))
acf(cabin.lm2$residuals)
pacf(cabin.lm2$residuals)

par(oldpar)
```

From the plots of ACF and PACF, we can see ACF quickly decays under the confidence boundary, though PACF has significant values on lags 2, 3 and 17, it is adequate for the model. Note the AIC of ARIMA model is 620.52.

Then we fit a linear model to the data using gls command.

```r
# Estimate correlations
x<- acf(cabin1$Temp)
```

```r
x[[1]]
```

```
## , , 1
##
##              [,1]
##  [1,]  1.00000000
##  [2,]  0.95282434
##  [3,]  0.82660964
##  [4,]  0.64093871
##  [5,]  0.41807242
##  [6,]  0.17956001
##  [7,] -0.05380491
##  [8,] -0.26497996
##  [9,] -0.44179231
## [10,] -0.57537416
## [11,] -0.66375650
## [12,] -0.70856366
## [13,] -0.71615768
## [14,] -0.69239623
## [15,] -0.63602542
## [16,] -0.54383763
## [17,] -0.41464100
## [18,] -0.25047698
## [19,] -0.05770528
## [20,]  0.15244597
## [21,]  0.36467155
## [22,]  0.56128887
## [23,]  0.72332232
## [24,]  0.83199986
```

```r
library(nlme)
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:forecast':
##
##     getResponse

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```r
cabin.lm3 <- gls(Temp ~ Time_Index + Hour -1,
                weights = varIdent(form = ~1 | Hour),
                correlation = corARMA(value = 0.5 * c(0.9, -0.8), # vect
```

```
or with length of 'p+q'
                form = ~ Temp | Time_Index, p = 2, q = 0, fixed = F), dat
a = cabin1)
summary(cabin.lm3)

## Generalized least squares fit by REML
##   Model: Temp ~ Time_Index + Hour - 1
##   Data: cabin1
##        AIC      BIC    logLik
##   1277.764 1449.666 -587.8818
##
## Correlation Structure: ARMA(2,0)
##  Formula: ~Temp | Time_Index
##  Parameter estimate(s):
##  Phi1  Phi2
##  0.45 -0.40
## Variance function:
##  Structure: Different standard deviations per stratum
##  Formula: ~1 | Hour
##  Parameter estimates:
##         0         1         2         3         4         5         6
    7
## 1.0000000 1.0790990 1.1600597 1.1822635 1.2401124 1.3543426 1.4587697
 1.6080223
##         8         9        10        11        12        13        14
   15
## 1.2737962 0.9676534 1.3120504 1.5856582 1.3105175 1.4948133 1.7044788
 1.8635806
##        16        17        18        19        20        21        22
    23
## 1.6898749 1.7718664 1.5786059 1.3355340 1.1842599 1.0945454 1.1024328
 1.1823344
##
## Coefficients:
##               Value Std.Error  t-value p-value
## Time_Index -0.01130 0.0028752 -3.92903   1e-04
## Hour0      19.21392 0.8150479 23.57398   0e+00
## Hour1      18.34432 0.8713263 21.05333   0e+00
## Hour2      17.66642 0.9293326 19.00979   0e+00
## Hour3      17.20701 0.9460148 18.18895   0e+00
## Hour4      16.72201 0.9879934 16.92523   0e+00
## Hour5      16.17471 1.0704095 15.11077   0e+00
## Hour6      15.60760 1.1462839 13.61583   0e+00
## Hour7      15.56700 1.2549862 12.40412   0e+00
## Hour8      21.73950 1.0157123 21.40320   0e+00
## Hour9      29.13409 0.8032197 36.27164   0e+00
## Hour10     37.57229 1.0447825 35.96183   0e+00
## Hour11     45.10269 1.2419185 36.31695   0e+00
## Hour12     51.02398 1.0455920 48.79913   0e+00
## Hour13     54.33968 1.1781109 46.12442   0e+00
```

```
## Hour14     54.95268 1.3302979 41.30855   0e+00
## Hour15     53.52277 1.4467530 36.99510   0e+00
## Hour16     50.70517 1.3212611 38.37634   0e+00
## Hour17     47.42817 1.3814778 34.33147   0e+00
## Hour18     40.81746 1.2425868 32.84878   0e+00
## Hour19     32.63246 1.0701137 30.49438   0e+00
## Hour20     27.55306 0.9651354 28.54838   0e+00
## Hour21     24.63005 0.9043755 27.23432   0e+00
## Hour22     22.63985 0.9109677 24.85253   0e+00
## Hour23     21.25395 0.9671502 21.97585   0e+00
##
##  Correlation:
##        Tm_Ind Hour0  Hour1  Hour2  Hour3  Hour4  Hour5  Hour6  Hour7
Hour8
## Hour0  -0.385

## Hour1  -0.363  0.140

## Hour2  -0.343  0.132  0.125

## Hour3  -0.340  0.131  0.124  0.117

## Hour4  -0.329  0.126  0.119  0.113  0.112

## Hour5  -0.306  0.118  0.111  0.105  0.104  0.101

## Hour6  -0.288  0.111  0.105  0.099  0.098  0.095  0.088

## Hour7  -0.266  0.102  0.096  0.091  0.090  0.087  0.081  0.077

## Hour8  -0.331  0.127  0.120  0.114  0.113  0.109  0.101  0.096  0.088

## Hour9  -0.422  0.162  0.153  0.145  0.144  0.139  0.129  0.122  0.112
 0.140
## Hour10 -0.327  0.126  0.119  0.112  0.111  0.108  0.100  0.094  0.087
 0.108
## Hour11 -0.278  0.107  0.101  0.095  0.095  0.091  0.085  0.080  0.074
 0.092
## Hour12 -0.333  0.128  0.121  0.114  0.113  0.109  0.102  0.096  0.088
 0.110
## Hour13 -0.298  0.114  0.108  0.102  0.101  0.098  0.091  0.086  0.079
 0.099
## Hour14 -0.266  0.102  0.096  0.091  0.090  0.087  0.081  0.077  0.071
 0.088
## Hour15 -0.246  0.095  0.089  0.085  0.084  0.081  0.075  0.071  0.065
 0.082
## Hour16 -0.272  0.105  0.099  0.093  0.093  0.089  0.083  0.078  0.072
 0.090
## Hour17 -0.262  0.101  0.095  0.090  0.089  0.086  0.080  0.076  0.070
 0.087
```

```
## Hour18 -0.294  0.113  0.107  0.101  0.100  0.097  0.090  0.085  0.078
 0.097
## Hour19 -0.344  0.132  0.125  0.118  0.117  0.113  0.105  0.099  0.091
 0.114
## Hour20 -0.384  0.148  0.139  0.132  0.131  0.126  0.118  0.111  0.102
 0.127
## Hour21 -0.413  0.159  0.150  0.142  0.141  0.136  0.127  0.119  0.110
 0.137
## Hour22 -0.413  0.159  0.150  0.142  0.141  0.136  0.127  0.119  0.110
 0.137
## Hour23 -0.392  0.151  0.142  0.135  0.134  0.129  0.120  0.113  0.104
 0.130
##        Hour9  Hour10 Hour11 Hour12 Hour13 Hour14 Hour15 Hour16 Hour17
 Hour18
## Hour0

## Hour1

## Hour2

## Hour3

## Hour4

## Hour5

## Hour6

## Hour7

## Hour8

## Hour9

## Hour10  0.138

## Hour11  0.117  0.091

## Hour12  0.141  0.109  0.092

## Hour13  0.126  0.098  0.083  0.099

## Hour14  0.112  0.087  0.074  0.088  0.079

## Hour15  0.104  0.081  0.068  0.082  0.073  0.066

## Hour16  0.115  0.089  0.076  0.091  0.081  0.072  0.067

## Hour17  0.111  0.086  0.073  0.087  0.078  0.070  0.065  0.071
```

```
## Hour18  0.124  0.096  0.082  0.098  0.087  0.078  0.072  0.080  0.077

## Hour19  0.145  0.113  0.096  0.114  0.102  0.091  0.085  0.094  0.090
 0.101
## Hour20  0.162  0.126  0.107  0.128  0.114  0.102  0.095  0.105  0.101
 0.113
## Hour21  0.175  0.135  0.115  0.138  0.123  0.110  0.102  0.112  0.108
 0.121
## Hour22  0.175  0.135  0.115  0.138  0.123  0.110  0.102  0.112  0.108
 0.121
## Hour23  0.166  0.129  0.109  0.131  0.117  0.104  0.097  0.107  0.103
 0.115
##        Hour19 Hour20 Hour21 Hour22
## Hour0
## Hour1
## Hour2
## Hour3
## Hour4
## Hour5
## Hour6
## Hour7
## Hour8
## Hour9
## Hour10
## Hour11
## Hour12
## Hour13
## Hour14
## Hour15
## Hour16
## Hour17
## Hour18
## Hour19
## Hour20  0.132
## Hour21  0.142  0.159
## Hour22  0.142  0.159  0.171
## Hour23  0.135  0.151  0.162  0.162
##
## Standardized residuals:
##         Min          Q1         Med          Q3         Max
## -2.40227605 -0.63532946 -0.01054631  0.64485020  2.21276389
##
## Residual standard error: 2.379262
## Degrees of freedom: 240 total; 215 residual
```

Model diagnostic

ACF and PACF

From the plots, the ACF shows positive alternative decaying pattern over the confidence band, the PACF drops under the significant level after lag 1. It doesn't seem to be better than the AR(2) and $Arima(2,0,0)$ model.

```
oldpar <- par(mfrow=c(2,1))
acf(cabin.lm3$residuals)
pacf(cabin.lm3$residuals)
```

```
par(oldpar)
```

Fitted against Observed

```
plot(cabin1$Temp ~ cabin1$Time_Index, type = 'l', xlab = 'Hour', ylab =
'Temperature (Â°C)', main = "gls Fitted vs Observed")
lines(cabin.lm3$fitted ~ cabin1$Time_Index, col = 'red')
legend("topleft", legend = c("Observed", "Fitted"), lty = c(1,1), col =
c("black", "red"))
```

```
oldpar <- par(mfrow=c(2,1))
hist(cabin.lm3$residuals, xlab = "Residuals", main = "Residual frequency
 of gls model")
qqnorm(cabin.lm3$residuals)
qqline(cabin.lm3$residuals, col = 'steelblue', lwd = 2)
```

```
par(oldpar)
```

The fitted values against observed plot does not fit well in the trough and peaks. The residuals against fitted values plot shows some heterogeneous variance, and the start and end of points in normal probability plot are out of the straight line, indicates the normal distribution is not good.

More gls model with different variance-covariate relationships:

```
cabin.lm3_2 <- gls(Temp ~ Time_Index + Hour -1,
                weights = varPower(form = ~ Time_Index|Hour),
                correlation = corARMA(value = 0.5 * c(0.9, -0.8), # vect
or with length of 'p+q'
                form = ~ Temp | Time_Index, p = 2, q = 0, fixed = F), dat
a = cabin1)
cabin.lm3_3 <- gls(Temp ~ Time_Index + Hour -1,
                weights = varExp(form = ~ Time_Index),
                correlation = corARMA(value = 0.5 * c(0.9, -0.8), # vect
or with length of 'p+q'
                form = ~ Temp | Time_Index, p = 2, q = 0, fixed = F), dat
a = cabin1)
anova(cabin.lm3_2, cabin.lm3)

##             Model df      AIC      BIC    logLik   Test  L.Ratio p-valu
e
## cabin.lm3_2     1 52 1277.247 1452.520 -586.6234

## cabin.lm3       2 51 1277.764 1449.666 -587.8818 1 vs 2 2.516888  0.11
26

anova(cabin.lm3_3, cabin.lm3)

##             Model df      AIC      BIC    logLik   Test  L.Ratio p-valu
e
## cabin.lm3_3     1 29 1247.957 1345.705 -594.9784

## cabin.lm3       2 51 1277.764 1449.666 -587.8818 1 vs 2 14.19314  0.89
45

anova(cabin.lm3_3, cabin.lm3_2)

##             Model df      AIC      BIC    logLik   Test  L.Ratio p-valu
e
## cabin.lm3_3     1 29 1247.957 1345.705 -594.9784

## cabin.lm3_2     2 52 1277.247 1452.520 -586.6234 1 vs 2 16.71003  0.8
232

oldpar <- par(mfrow=c(2,1))
acf(cabin.lm3_3$residuals)
pacf(cabin.lm3_3$residuals)
```

```
par(oldpar)
hist(cabin.lm3_3$residuals, xlab = "Residuals", main = "Residual frequen
cy of gls model")
```

```
qqnorm(cabin.lm3_3$residuals)
qqline(cabin.lm3_3$residuals, col = 'steelblue', lwd = 2)
```

Above all, the gls model performance is similar to simple linear model, but worse than AR(2) and $Arima(2,0,0)$ model.

Box-Jenkins ARIMA model

```
# acf and pacf stationary diagnostic
oldpar <- par(mfrow=c(2,1))
acf(cabin.ts ,lag.max = 400)
pacf(cabin.ts)
```

```
par(oldpar)
```

From the acf of time series, we can see the autocorrelations slowly decay to near 0 after lags 8, which indicates the mean will change over time, that the process may be non-stationary. Then we include seasonal differences, and use time plot, ACF and PACF plots to re-evaluate the stationarity.

```
# Plots after once and twice seasonal differences
oldpar <- par(mfrow=c(3,1))
plot(cabin.ts)
plot(diff(cabin.ts,24))
plot(diff(diff(cabin.ts,24),1))
```

```
par(oldpar)

# ACF and PACF plots of seasonal differences
oldpar <- par(mfrow=c(3,1))
acf(diff(cabin.ts, 24), lag.max = 200)
pacf(diff(cabin.ts, 24))
par(oldpar)
```

We can see the ACF quickly decays to 0 at lag 1, the PACF decays to significance bound after lag 1. Then we include the seasonal differences in ARIMA model.

```
# Box_Jenkins ARIMA model
cabin.arima <- Arima(cabin2$Temp, order = c(2,0,0), seasonal = c(2,24,
0),  xreg = xregs, include.constant = FALSE)
(x <- as.matrix(coef(cabin.arima), ncol = 4))
```

```
##             [,1]
## ar1  1.433447604
## ar2 -0.494175493
## x1  -0.005297365
## x2  19.081570170
## x3  18.157203064
## x4  17.419251792
## x5  16.900266234
## x6  16.358928584
## x7  15.759672983
## x8  15.145245402
## x9  15.061722784
## x10 21.195249046
## x11 28.554294963
## x12 36.959800544
## x13 44.459837892
## x14 50.352584205
## x15 53.641037262
## x16 54.227588378
## x17 52.771539187
## x18 49.927604478
## x19 46.623586310
## x20 39.984710116
## x21 31.769855804
## x22 26.658376245
## x23 23.700455665
## x24 21.671816467
## x25 20.243235518
```

```
summary(cabin.arima)
```

```
## Series: cabin2$Temp
## Regression with ARIMA(2,0,0) errors
##
## Coefficients:
##          ar1      ar2       x1       x2       x3       x4       x5
x6
##       1.4334  -0.4942  -0.0053  19.0816  18.1572  17.4193  16.9003  1
6.3589
## s.e.  0.0559   0.0566   0.0111   1.6613   1.6628   1.6638   1.6648
1.6660
##            x7       x8       x9      x10      x11      x12      x13
x14
##       15.7597  15.1452  15.0617  21.1952  28.5543  36.9598  44.4598  5
0.3526
## s.e.   1.6676   1.6696   1.6720   1.6748   1.6778   1.6811   1.6844
1.6877
##           x15      x16      x17      x18      x19      x20      x21
x22
##       53.6410  54.2276  52.7715  49.9276  46.6236  39.9847  31.7699  2
```

```
6.6584
## s.e.   1.6907   1.6932   1.6951   1.6961   1.6959   1.6943   1.6911
1.6862
##           x23      x24      x25
##       23.7005  21.6718  20.2432
## s.e.   1.6800   1.6733   1.6681
##
## sigma^2 = 0.6844:  log likelihood = -282.26
## AIC=620.52   AICc=628.22   BIC=717.98
##
## Training set error measures:
##                      ME      RMSE       MAE        MPE     MAPE      MAS
E
## Training set 0.007526434 0.7793551 0.4908785 -0.1203926 1.899034 0.14
91637
##                    ACF1
## Training set 0.06723037

# Model Diagnostic
oldpar <- par(mfrow=c(2,1))
hist(cabin.arima$residuals, xlab = "Residuals", main = "Residual frequen
cy of gls model")
qqnorm(cabin.arima$residuals)
qqline(cabin.arima$residuals, col = 'steelblue', lwd = 2)
```

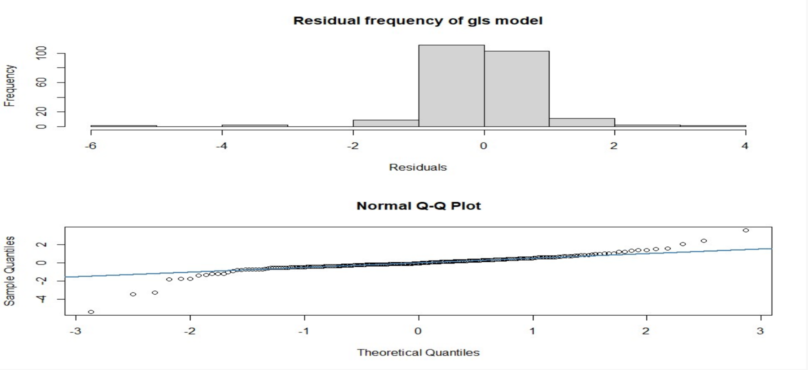


Figure A.1

```
par(oldpar)

# ACF and PACF plots of $ARIMA(2,0,0)(2,24,0)_{24}$
oldpar <- par(mfrow=c(2,1))
acf(cabin.arima$residuals)
pacf(cabin.arima$residuals)

par(oldpar)
# Residuals vs fitted values plot
plot(cabin1$Temp ~ cabin1$Time_Index,type = 'l', xlab = "Hour", ylab="Te
mperature (Â°C)", main= "Fitted vs Observed values")
lines(cabin.arima$fitted ~ cabin1$Time_Index, col = 'red')
```

```
legend("topleft", legend = c("Observed", "Fitted"), lty = c(1,1), col =
c("black", "red"))
```