MTHM033 - Statistical Modelling in Space and Time
Assessed Coursework 2

670033412

Strength of Overturning In The North Atlantic

The global overturning of ocean waters involves the equatorward transport of cold, deep waters and the poleward transport of warm, near-surface waters [Loz12]. Here we use data collected every 12 hours between 2004-04-02 00:00:00 and 2014-03-22 00:00:00 from a mooring positioned at 26°N measured in Sverdrup (Sv) or cubic hectometers per second (hm^3/s).

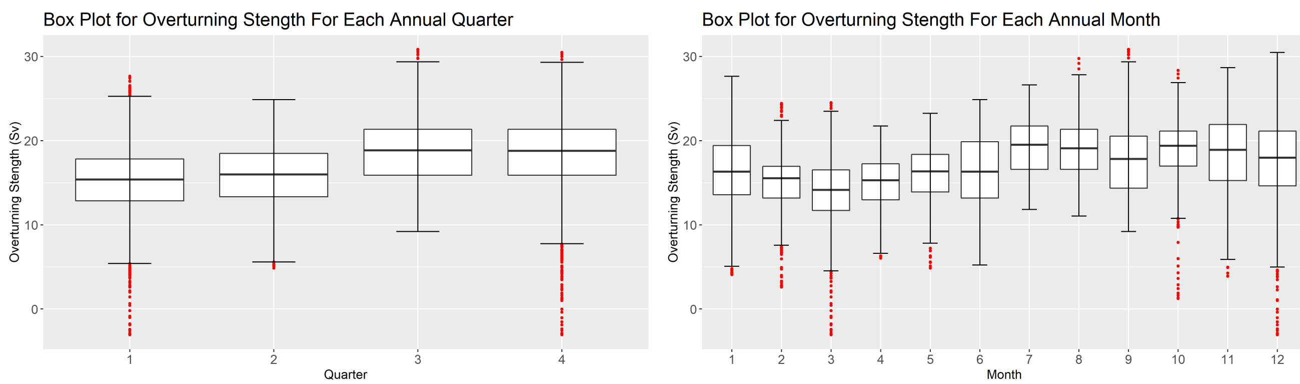
1 Data Integrity

We begin by first running a summary on the data to ensure that all the values are within the ranges we would expect.

Listing 1: Summary of Data

```
      year          month            day              hour            Quarter       Days_since_start Overturning_Strength
 Min.   :2004   Min.   : 1.000   Min.   : 1.0    Min.   : 0.000   Min.   :1.000   Min.   :   1.0   Min.   :-3.073
 Max.   :2014   Max.   :12.000   Max.   :31.0    Max.   :12.000   Max.   :4.000   Max.   :3642.0   Max.   :30.822
```

We can see that our data ranges are appropriate for our date and time columns however, without prior knowledge of the range of overturning strength it is difficult to detect outliers. We may attempt to do so using a box plot



(a) Data grouped into annual quarters. (b) Data grouped into annual months.

Figure 1: Box plot showing the distribution of overturning strength (Sv) for both annual quarters and for each month. Outliers defined as $\frac{3}{2}$IQR from the upper or lower quartiles are highlighted in red.

We can see in Figure 1 which uses the interquartile range as a method to detect outliers we find an extremely high number of outliers especially in the first and fourth quartile. To investigate further into these outliers we can extract them from our data and plot them against time.

Looking at Figure 2 we can see that the outliers tend to be grouped together but are spread well throughout the data. We may then conclude that these outliers are instead byproducts of the complex nature and long term trends of the ocean.

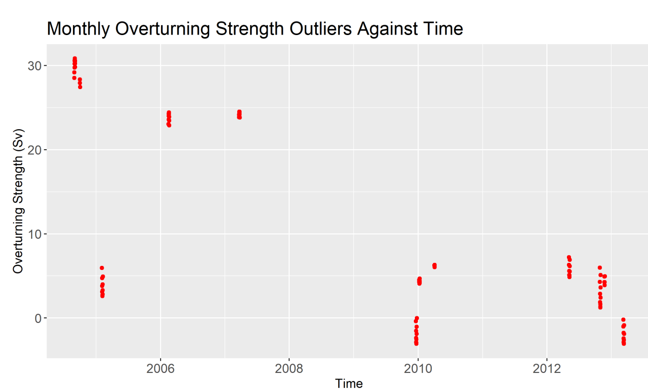


Figure 2: Monthly overturning strength outliers plotted against time.

To get an overview of our data we begin by grouping it into years and annual quarters and then take the mean of each group. This is to reduce noise in our data and also reduces the number of points which helps speeds up the model building computation. We can see this data as a time series in Figure 3.

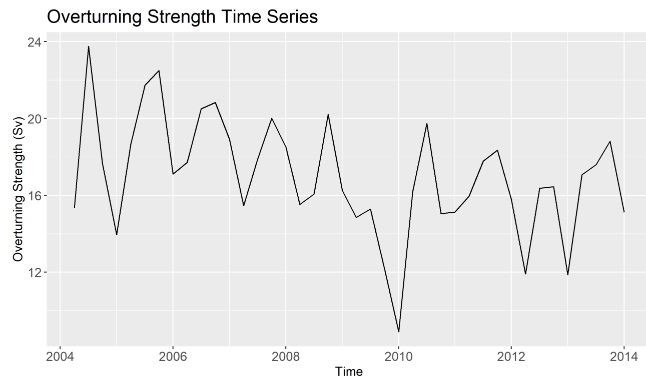


Figure 3: A time series of our original data where each point is the mean taken over the annual quarter.

Taking a look at Figure 3 we can see that there appears to be a pattern of changing between up and down within fairly consistent values roughly every half of year. However, this pattern is not followed between 2009 and 2011 when we see a drop shortly followed by another drop before a very sharp upwards tick till halfway through 2010. Whilst this significant drop may seem as an outlier it is generally accepted in the literature, and its origin is uncertain [BKMM14].

With this quick assessment of the data concluding that there are no outliers, we will then continue to build our models without removing any data from the provided data set.

2 ARMA and ARIMA Model

The first step in building an autoregressive (integrated) moving average (AR(I)MA) model is to determine which process best describes our data. Our process is considered to consist of:

Autoregressive:

$$\mathrm{AR}\,(p) : x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \epsilon_t$$

Integrated: d is the degree of differencing (the number of times the data have had past values subtracted).

Moving Average:

$$\mathrm{MA}\,(q) : x_t = \sum_{i=0}^{q} \beta_i \epsilon_{t-i}$$

Where x_t represents the value measured at the t^{th} time step (Δt) and ϵ_t are identical and independent realizations from a normal (Gaussian) distribution with zero mean and variance σ^2.

2.1 Autoregressive Moving Average Model (ARMA)

An ARMA model only consist off the $\mathrm{AR}\,(p)$ and $\mathrm{MA}\,(q)$ often referred to as $\mathrm{ARMA}\,(p,q)$ but is identical to $\mathrm{ARIMA}\,(p,0,q)$, expressed mathematically as:

$$\mathrm{ARMA}\,(p,q) : x_t = \sum_{i=1}^{p} \alpha_i x_{t-i} + \sum_{i=0}^{q} \beta_i \epsilon_{t-i} \quad \text{where } \epsilon_t \sim \mathrm{N}\left(0, \sigma^2\right)$$

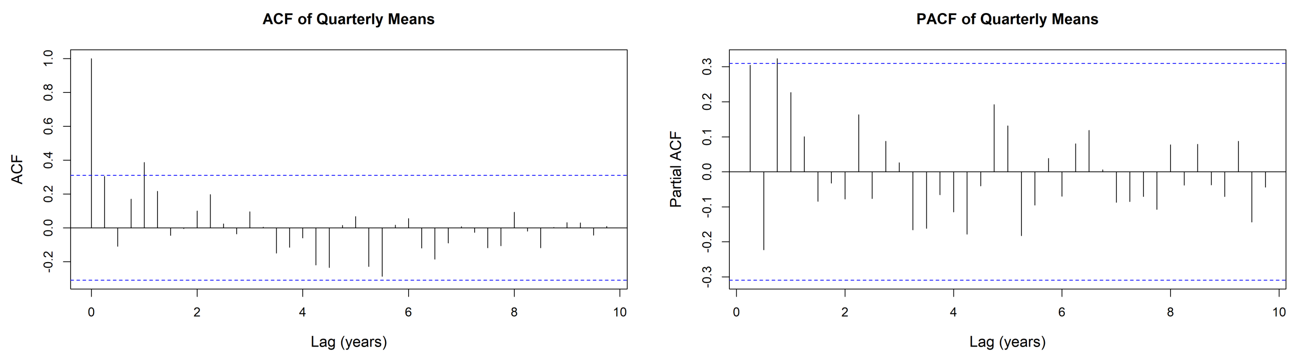
Our problem then is to fit parameters (p, q) that best describe the underlying process of data.

2.1.1 Finding optimal parameters

It is possible that our data may have one of the parameters equal 0 meaning that we just an AR or MA process. One way to check this is to calculate both the auto correlation (ACF) and partial auto correlation functions (PACF). Then using the table below we can see if our data satisfies the criteria.

	AR(p)	MA(q)	ARMA(p,q)
ACF	Tails off	Cuts off after lag q	Tails off
PACF	Cuts off after lag p	Tails off	Tails off

The ACF and PACF functions of our data is shown in Figure 4. Taking a look at the ACF Figure 4a we see that we have two peaks at 0.25 and 1.00 (excluding peak at 0). Then looking at the PACF Figure 4b we have one peak at 0.75 but also one peak just below the 95% confidence interval at 0.25. Since any of these peaks could be down to noise, it is hard to exactly determine our parameters from this analysis alone.



(a) Auto correlation function

(b) Partial auto correlation function

Figure 4: Auto correlation and partial auto correlation functions of our quarterly means data plotted for different lag values in years, with 95% significance level shown in blue.

We will fit several ARMA models for varying values, $(p, q) \in [0, 3] \times [0, 7]$ and calculate both the AIC and BIC of the fit. This produces the following results:

p/q	0	1	2	3	4	5	6	7
0	412.05	399.19	399.91	405.23	403.42	404.96	408.73	413.08
1	410	400.07	405.44	405.16	407.67	409.72	414.09	418.62
2	410.03	405.24	410.92	403.7	409.39	413.58	412.62	424.28
3	401.39	404.6	408.75	409.39	415.08	412.44	417.67	421.29

Table 1: Sum of AIC and BIC for different (p, q) pairs. For $q > 7$ or $p > 3$ the parameter pair approach the end of the stationarity region.

- The minimum pair hello of (p, q) when minimizing the AIC is $p = 2$ and $q = 3$
- The minimum pair hello of (p, q) when minimizing the BIC is $p = 0$ and $q = 1$
- The minimum pair hello of (p, q) when minimizing the Sum of AIC and BIC is $p = 0$ and $q = 1$

It is worth mentioning that the models allowed for there to be a mean and the best fit was calculated by first using conditional-sum-of-squares to find starting values, then maximum likelihood.

Continuing with $p = 0$ and $q = 1$, we shall take the residuals of our ARMA $(0, 1)$ model and plot them against time along with there ACF and the distribution of the residuals. This can be seen in Figure 5. It seems we have removed the correlation between points from looking at the ACF, performing a Ljung-Box test returned a p-value of 0.22 . This indicates the residuals are independent [LB78]. However, we can also see that the residuals do not fit well to a normal distribution. It is for this reason we will try other parameters.

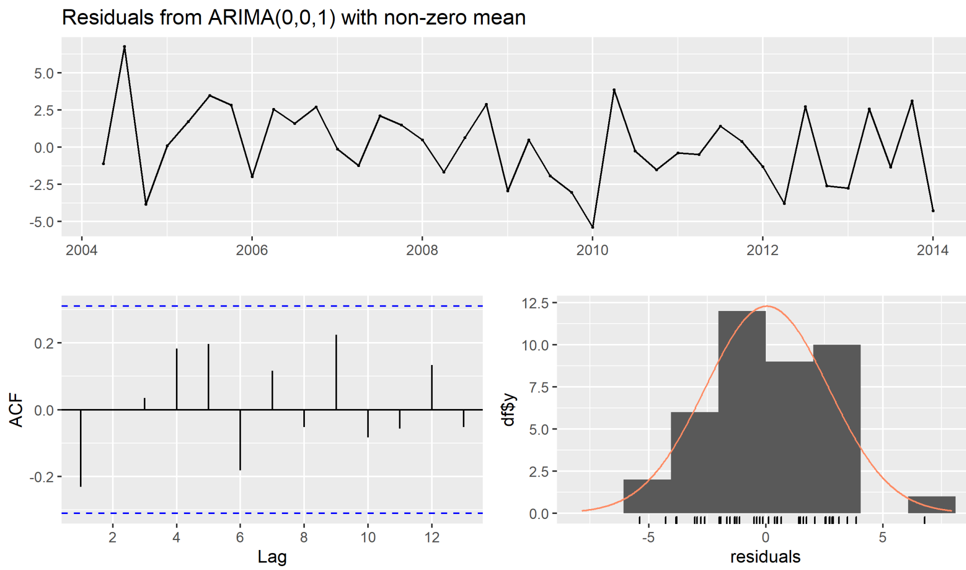


Figure 5: Residuals from our ARMA $(0, 1)$ model plotted against time. The ACF between residuals and their distribution.

In an attempt to find parameters which fit better we refer to the "auto.arima()" function which is from the R package "forecast" [HAB+22]. This is a function which uses a variation of the Hyndman-Khandakar algorithm [HK08], which combines unit root tests, minimisation of the AICc (Second-order Akaike Information Criterion) and MLE to obtain an ARIMA model.

When using AICc as the measure of so-called "goodness" we are told that $p = 0$ and $q = 2$ are optimal. The residuals of our ARMA $(0, 2)$ model can be seen in Figure 6. We can immediately see that the residuals follow the normal distribution much better. Also, we still have low correlation between points from looking at our ACF. Performing a Ljung-Box test we get a p-value of 0.22 , this again indicates independence between our residuals.

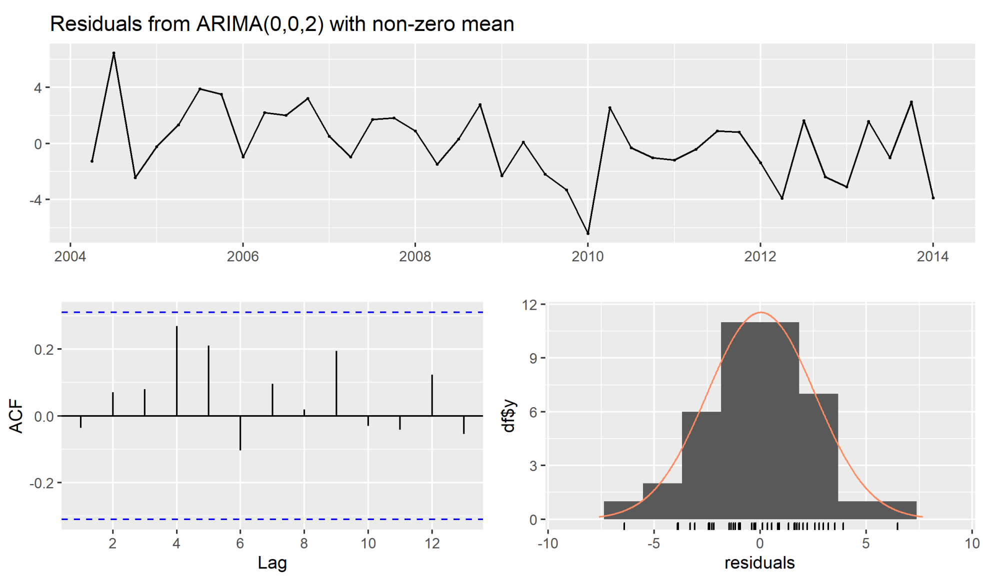


Figure 6: Residuals from our ARMA $(0, 2)$ model plotted against time. The ACF between residuals and their distribution.

2.1.2 Building our model

Accepting $p = 0$ and $q = 2$ as our optimal parameters, from our maximum likelihood (MLE) method our full model is:

$$x_t = 16.87 + 0.63\epsilon_{t-1} - 0.24\epsilon_{t-2} \quad \epsilon_t \sim \mathrm{N}(0, 6.835)$$

Now, using this model to forecast 6 quarters ahead of our data.

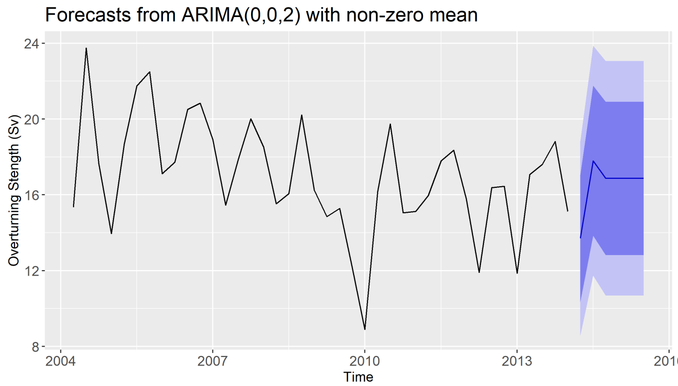


Figure 7: Forecast of overturning strength (Sv) 6 quarters into the future with 80% and 95% confidence intervals shown in dark blue and light blue respectively.

?Discussion?

2.2 Autoregressive Integrated Moving Average Model (ARIMA)

An ARIMA model consists of three parameters, p, q and d. p and q are identical to an ARIMA model and d is the degree of differencing (the number of times the data have had past values subtracted).

2.2.1 Finding optimal parameters

To find the optimal parameters we will again use the "auto.arima" function mentioned in subsubsection 2.1.1.

The optimal parameters determined are $p = 0$, $d = 1$ and $q = 0$. Checking the residuals of our ARIMA $(0, 1, 0)$ model as we did previously:

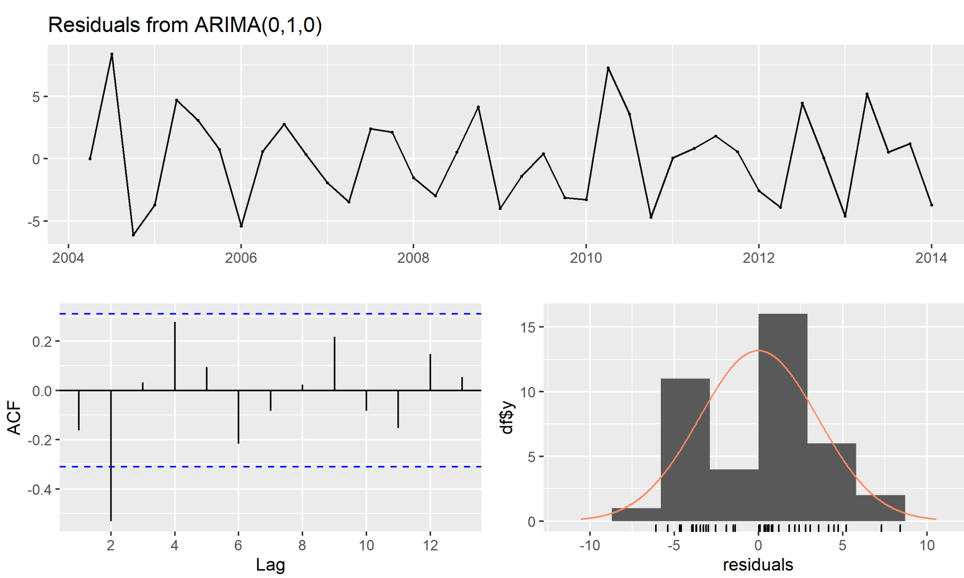


Figure 8: Residuals from our ARIMA $(0, 1, 0)$ model plotted against time. The ACF between residuals and their distribution.

We can see from Figure 8 that the distribution does not appear to be normal and from the ACF plot we can see that there appears to be some correlation between the residuals; this is confirmed by a p-value of 0.01 from our Ljung-Box test. This p-value indicates that the data are not independently distributed; they exhibit serial correlation [LB78].

After experimenting with forcing different d values into the "auto.arima" function, a better parameter triplet was found: ARIMA $(4, 2, 0)$

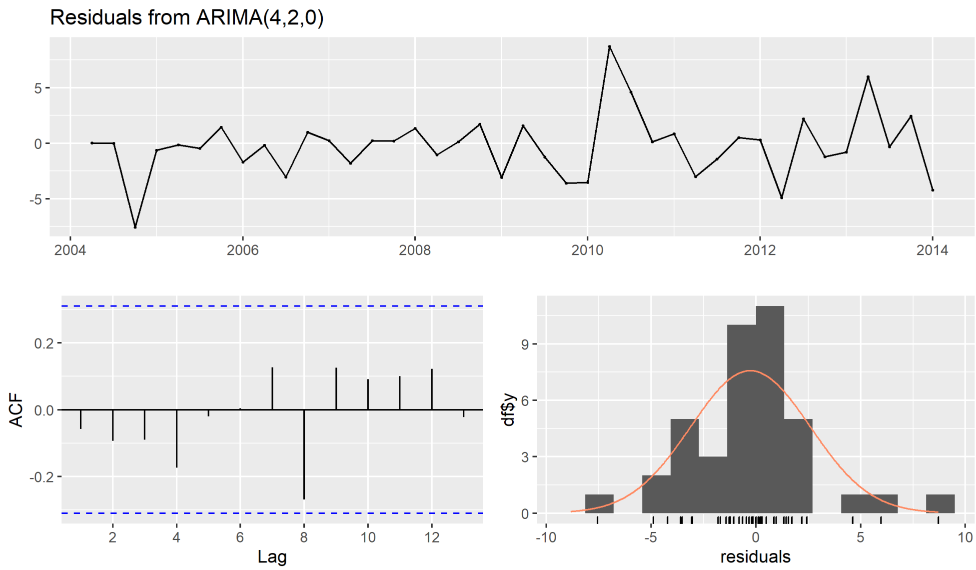


Figure 9: Residuals from our ARIMA $(4, 2, 0)$ model plotted against time. The ACF between residuals and their distribution.

We can see in Figure 9 that we have managed to remove the correlation between residuals in the ACF and our p-value from our Ljung-Box test is 0.14 which indicates independence. However, we can see that our distribution of residuals are not normal.

2.2.2 Building our model

Our ARIMA $(4, 2, 0)$ model has the following coefficients:

Listing 2: ARIMA(4,2,0) Coefficients

```
Coefficients:
           ar1      ar2      ar3      ar4
       -1.0195  -1.3529  -0.9128  -0.5612
s.e.    0.1473   0.1666   0.1627   0.1527
```

Now, forecasting 6 quarters ahead with our model:

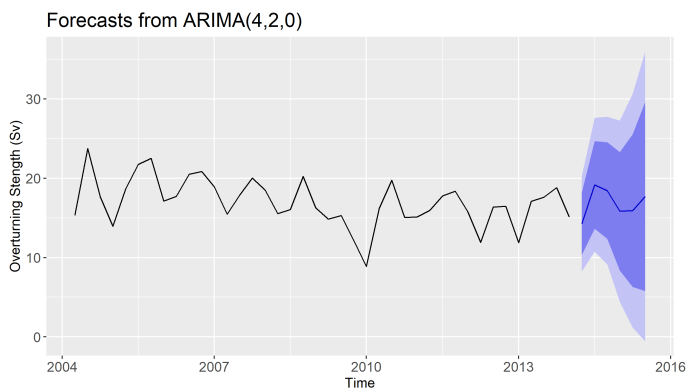


Figure 10: Forecast of overturning strength (Sv) 6 quarters into the future with 80% and 95% confidence intervals shown in dark blue and light blue respectively.

2.3 Comparing our ARMA and ARIMA Models

Here is quick overview of each model:

Listing 3: ARMA(0,2)

```
ARIMA(0,0,2) with non-zero mean

Coefficients:
         ma1      ma2     mean
      0.6323  -0.2403  16.8684
s.e.  0.1461   0.1367   0.5528

sigma^2 = 6.835:  log likelihood = -94.29
AIC=196.57   AICc=197.72   BIC=203.33
```

Listing 4: ARIMA(4,2,0)

```
ARIMA(4,2,0)

Coefficients:
          ar1      ar2      ar3      ar4
      -1.0195  -1.3529  -0.9128  -0.5612
s.e.   0.1473   0.1666   0.1627   0.1527

sigma^2 = 9.443:  log likelihood = -96.26
AIC=202.51   AICc=204.39   BIC=210.7
```

We can see that the ARMA model has the best fit for all 4 measurements (log likelihood, AIC, BIC and AICc). This is reflected in the forecast made by the models where the ARMA model has much smaller confidence intervals. However, the forecast for the ARIMA model seems to take on a shape that closer resembles the previous behavior in the time series, opposed to the ARMA models near straight line.

3 Dynamic Linear Models (DLM)

A DLM is specified by the following equations:

$$\begin{cases} y_t = F_t\theta_t + \nu_t, & \nu_t \sim \mathrm{N}(0, V_t) \\ \theta_t = G_t\theta_{t-1} + \omega_t, & \omega_t \sim \mathrm{N}(0, W_t) \end{cases}$$

for $t = 1, \ldots, n$ together with a prior distribution for θ_0:

$$\theta_0 \sim \mathrm{N}(m_0, C_0)$$

Here y_t is an m-dimensional vector, representing the observation at time t, while θ_t is a generally unobservable p-dimensional vector representing the state of the system at time t. The ν_t's are observation errors and the ω_t's evolution errors. The matrices F_t and G_t have dimension m by p and p by p, respectively, while V_t and W_t are variance matrices of the appropriate dimension.

3.1 Building the model

Here we let our prior consist of a second order polynomial and seasonal[4] component.

After fitting a model to our data and smoothing we are able to extract the follow plots shown in Figure 11. We can see our model has estimated a strong seasonal component and a downward trend.

Using our model we are able to apply a filter and then produce a forecast 6 quarters past our data, this is shown in Figure 12. Comparing the forecast produced using DLM to our ARMA model, Figure 7 we can see that the 95% confidence interval is smaller for DLM implying that it is more confident in its prediction. It is also worth noting that the ARMA forecast has uniform error margins, whereas in the DLM forecast the error bars get larger as the forecast moves further into the future. The DLM forecast also takes on a very different shape proceeding with a slight continuous increase opposed to the big rises and falls predicted by the ARMA model.

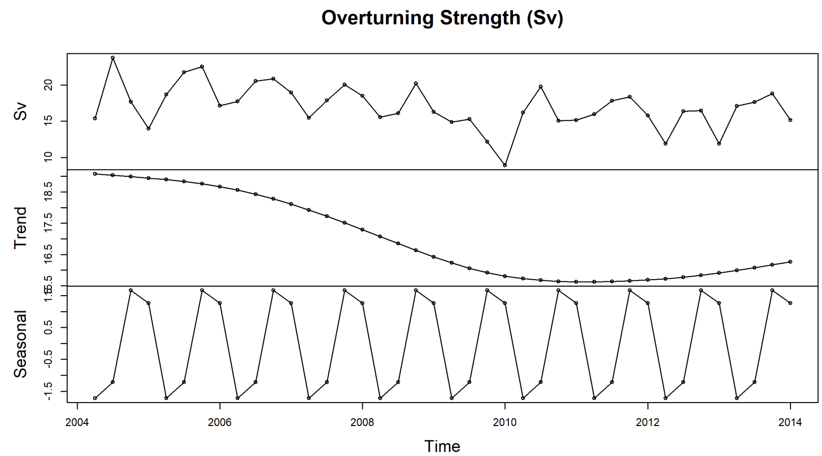


Figure 11: From top to bottom: The observed overturning strength, the estimated trend of our data from our model, the estimated seasonal component of data from our model.

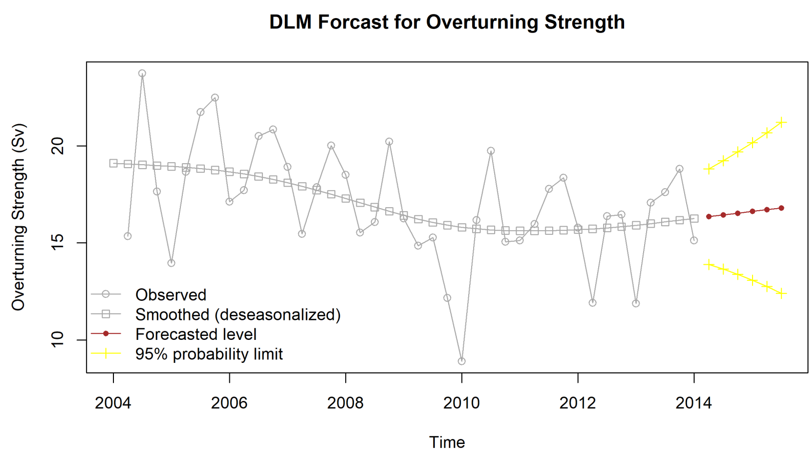


Figure 12: Forecast of overturning strength (Sv) 6 quarters into the future with 95% confidence intervals shown in yellow.

References

[BKMM14] H.L. Bryden, B.A. King, G. McCarthy, and Elaine Mcdonagh. Impact of a 30% reduction in atlantic meridional overturning during 2009-2010. *Ocean Science*, 11, 02 2014.

[HAB+22] Rob Hyndman, George Athanasopoulos, Christoph Bergmeir, Gabriel Caceres, Leanne Chhay, Mitchell O'Hara-Wild, Fotios Petropoulos, Slava Razbash, Earo Wang, and Farah Yasmeen. *forecast: Forecasting functions for time series and linear models*, 2022. R package version 8.16.

[HK08] Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of statistical software*, 27:1–22, 2008.

[LB78] G. M. LJUNG and G. E. P. BOX. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 08 1978.

[Loz12] M. Susan Lozier. Overturning in the north atlantic. *Annual Review of Marine Science*, 4(1):291–315, 2012. PMID: 22457977.

Appendix A: R Code

All files can be found: https://github.com/leele2/Statistcial-Modelling-in-Space-and-Time-CW2/tree/master/RFiles

Listing 5: ../../RFiles/Master.R

```
#https://otexts.com/fpp2/seasonal-arima.html
rm(list = ls())
############# Set as Source Directory #############
source <- paste0("C:/Users/dj-lu/OneDrive - University of Exeter/University of E",
    "xeter/05 - Fifth Year/Statistical Modelling in Space and Time/Coursework",
    "s/Coursework 2")
#############                         #############
## Preamble ##
setwd(paste0(source, "/RFiles"))
#Install required Packages
if (!require(pacman)){
    install.packages("pacman")
    library(pacman)
}
p_load("httpgd", "ggplot2", "car", "ggfortify", "zoo", "tibble", "stringr",
    "forecast", dlm)
#Graphical variables
# Height, Width, DPI, ggplot font size, base font size (%)
i_sz <- 5 * (1)
i_sz <- c(i_sz[1], (60 / 35) * i_sz[1], 300, 12, 1.2)
theme0 <- theme(title = element_text(size = 1.2 * floor(i_sz[4]), hjust = 0.5),
                axis.title = element_text(size = i_sz[4]),
                axis.text = element_text(size = i_sz[4]))
#Initializing plot enviroment for httpgd
ggplot()
##          ##

## Reading Data ##
source("Read_data.R")
##              ##

## Validating Data ##
source("Validate.R")
##                 ##

## Building ARMA Model ##
source("ARMA.R")
##                     ##

## Building DLM Model ##
source("DLM.R")
##                     ##
```

Listing 6: ../../RFiles/Read_data.R

```
## Reading Data ##
#Importing data
data <- read.csv("Data/Overturning data.csv")
#Creating DateTime Column
data$DateTime <- as.POSIXct(with(data, paste(paste(year, month, day, sep = "-"),
                                             paste(hour, 0, 0, sep = ":")),
                                 "%Y-%m-%d %H:%M:%S"))
#Creating {year}-{Quarter} column
data$qy <- paste(data$year, data$Quarter, sep = "-")
#Creating time series from quarterly means
data_mean.ts <- ts(as.vector(tapply(data$Overturning_Strength, data$qy, mean)),
                   start = c(2004, 2), frequency = 4)
##              ##
```

Listing 7: ../../RFiles/ARMA.R

```
## Preamble ##
sav_dir <- paste0(source, "/Latex_Files/Main/Sections/ARIMA")
##          ##

## Plotting ACF and PACF ##
png(paste0(sav_dir, "/Plots/ACF.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
plot(acf(data_mean.ts, lag.max = 40),
     main = NA, xlab = "Lag (years)", cex.lab = i_sz[5])
title(main = "ACF of Quarterly Means", cex.main = i_sz[5])
dev.off()
png(paste0(sav_dir, "/Plots/PACF.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
plot(pacf(data_mean.ts, lag.max = 40),
     main = NA, xlab = "Lag (years)", cex.lab = i_sz[5])
title(main = "PACF of Quarterly Means", cex.main = i_sz[5])
dev.off()
##                       ##
```

```r

## Brute Force testing (p,q) Combinations ##
p_vals <- 0:3
q_vals <- 0:7
aIc <- matrix(0, length(p_vals), length(q_vals))
bIc <- matrix(0, length(p_vals), length(q_vals))
sIc <- matrix(0, length(p_vals), length(q_vals))
#Calculating AIC and BIC for each different (p,q) combinations
for (i in c(1:length(p_vals))){
  for (j in c(1:length(q_vals))){
    tmp <- Arima(data_mean.ts, order = c(as.numeric(p_vals[i]), 0,
                                         as.numeric(q_vals[j])))
    aIc[i, j] <- AIC(tmp)
    bIc[i, j] <- BIC(tmp)
    sIc[i, j] <- aIc[i, j] + bIc[i, j]
  }
}
rm(j)
#Finding minimum value across combinations
mins = matrix(0, 3, 2)
mins[1, ] <- which(aIc == min(aIc), arr.ind = TRUE)
mins[2, ] <- which(bIc == min(bIc), arr.ind = TRUE)
mins[3, ] <- which(sIc == min(sIc), arr.ind = TRUE)
#Outputting minimum combinations
tmp <- c("AIC", "BIC", "Sum of AIC and BIC")
for (i in c(1:3)){
  output <- paste0("The minimum pair hello of $(p,q)$ when minimizing the ", tmp[i],
                   " is $p$ = ", mins[i, 1] - 1, " and $q$ = ", mins[i, 2] - 1)
  writeLines(output, paste0(sav_dir, "/Outputs/", "min", i, ".txt"))
}
output = sIc
colnames(output) <- 0:(length(q_vals) - 1)
rownames(output) <- 0:(length(p_vals) - 1)
rm(aIc, bIc, sIc, p_vals, q_vals, i)
write.table(as.data.frame(round(output, 2)) %>% rownames_to_column('p/q'),
            paste0(substr(sav_dir, 1, str_locate(sav_dir, "Main/")[2]),
            "S2tab1.csv"), quote = F, sep = ",", row.names = F)
rm(output)
##                                          ##

## Building and testing ARMA model ##
#Checking model
fit.sIc <- Arima(data_mean.ts, order = c(mins[3, 1] - 1, 0 , mins[3, 2] - 1))
png(paste0(sav_dir, "/Plots/manual_res.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
tmp <- checkresiduals(fit.sIc, plot = T)
dev.off()
capture.output(cat(round(tmp$p.value, 2)), file = paste0(sav_dir,
  "/Outputs/manual_res.txt"))
#Comparing to auto model
fit.auto <- auto.arima(data_mean.ts, max.d = 0, seasonal = F)
png(paste0(sav_dir, "/Plots/auto_res.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
tmp <- checkresiduals(fit.auto, plot = T)
dev.off()
capture.output(cat(round(tmp$p.value, 2)), file = paste0(sav_dir,
  "/Outputs/auto_res.txt"))
#Producing forecast
png(paste0(sav_dir, "/Plots/forecast.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
tmp <- autoplot(forecast(fit.auto, h = 6), ylab = "Overturning Stength (Sv)") +
 theme0 + theme(plot.margin = unit(c(0.5,0.5,0.5,0.7), "cm"))
print(tmp)
dev.off()
rm(mins, )
##                                 ##

## Building and testing ARIMA model ##
fit.auto.arima <- auto.arima(data_mean.ts, seasonal = F)
png(paste0(sav_dir, "/Plots/auto__arima_res.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
tmp <- checkresiduals(fit.auto.arima, plot = T)
dev.off()
capture.output(cat(round(tmp$p.value, 2)), file = paste0(sav_dir,
  "/Outputs/auto_arima_res.txt"))
fit.auto.arima2 <- auto.arima(data_mean.ts, seasonal = F, d = 2)
png(paste0(sav_dir, "/Plots/auto__arima_res2.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
tmp <- checkresiduals(fit.auto.arima2, plot = T)
dev.off()
capture.output(cat(round(tmp$p.value, 2)), file = paste0(sav_dir,
  "/Outputs/auto_arima_res2.txt"))
#Producing forecast
png(paste0(sav_dir, "/Plots/forecast_arima.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
tmp <- autoplot(forecast(fit.auto.arima2, h = 6),
  ylab = "Overturning Stength (Sv)") +
  theme0 + theme(plot.margin = unit(c(0.5,0.5,0.5,0.7), "cm"))
```

```
print(tmp)
dev.off()
rm(sav_dir, tmp, fit.sIc, fit.auto.arima, fit.auto.arima2)
##                                   ##
```

Listing 8: ../../RFiles/DLM.R

```
## Preamble ##
sav_dir <- paste0(source, "/Latex_Files/Main/Sections/DLM")
##          ##

## Building DLM Model ##
#Defining prior
dlmSv <- dlmModPoly(order=2) + dlmModSeas(4)
buildFun <- function(x) {
    diag(W(dlmSv))[2:4] <- exp(x[1:3])
    V(dlmSv) <- exp(x[4])
    return(dlmSv)
}
#Fit data using MLE
(fit <- dlmMLE(data_mean.ts, parm = rep(0, 4), build = buildFun))$conv
dlmSv <- buildFun(fit$par)
drop(V(dlmSv))
diag(W(dlmSv))[2:3]
#Smoothing estimates and plot
SvSmooth <- dlmSmooth(data_mean.ts, mod = dlmSv)
xs <- cbind(data_mean.ts, dropFirst(SvSmooth$s[,c(1,4)]))
colnames(xs) <- c("Sv", "Trend", "Seasonal")
png(paste0(sav_dir, "/Plots/trends.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
plot(xs, type = 'o', main = "Overturning Strength (Sv)")
dev.off()
#Filter and forcast
Svfilt <- dlmFilter(data_mean.ts,mod=dlmSv)
Svfore <- dlmForecast(Svfilt,nAhead=6)
#Plotting forcast
sqrtR <- sapply(Svfore$R, function(x) sqrt(x[1,1]))
pl <- Svfore$a[,1] + qnorm(0.025, sd = sqrtR)
pu <- Svfore$a[,1] + qnorm(0.975, sd = sqrtR)
x <- ts.union(window(data_mean.ts), window(SvSmooth$s[,1]),Svfore$a[,1], pl, pu)
png(paste0(sav_dir, "/Plots/forecast.png"), i_sz[2], i_sz[1],
    units = "in", res = i_sz[3])
plot(x, plot.type = "single", type = 'o', pch = c(1, 0, 20, 3, 3),col = c("darkgrey", "darkgrey", "brown", "yellow", "
     yellow"), ylab = "Overturning Strength (Sv)")
title(main = "DLM Forcast for Overturning Strength")
legend("bottomleft", legend = c("Observed", "Smoothed (deseasonalized)", "Forecasted level", "95% probability limit"),
      bty = 'n', pch = c(1, 0, 20, 3, 3), lty = 1,col = c("darkgrey", "darkgrey", "brown", "yellow", "yellow"))
dev.off()
##                    ##
```