Final Presentation - Math 531T

Elijah Amirianfar, Nathaniel Rice, Henry Surjono, Cory Suzuki

June 26, 2024

Introduction

- What is our data set?

Introduction

- What is our data set?
- Exploring energy use in a home in Belgium

Introduction

- What is our data set?
- Exploring energy use in a home in Belgium
- What is the goal here?

Introduction

- What is our data set?
- Exploring energy use in a home in Belgium
- What is the goal here?
- To predict the volatility of how much energy will be used in the future.

Exploratory Data Analysis

Exploratory Data Analysis

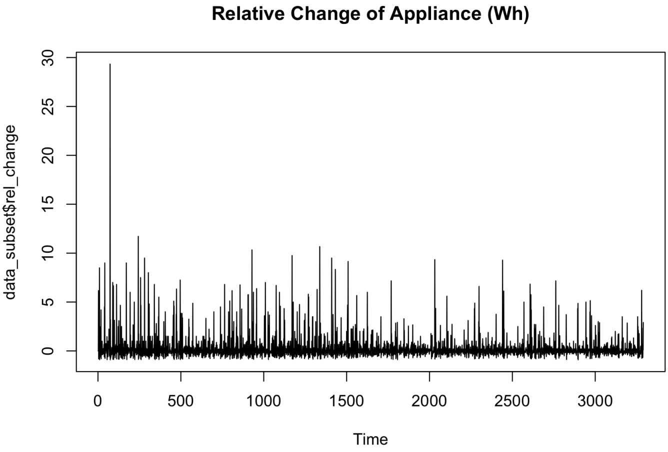
- What do we calculate?

Exploratory Data Analysis

- What do we calculate?
- Relative change of energy use over every hour

```
(Appliances - lag(Appliances))
/lag(Appliances)
```

Exploratory Data Analysis

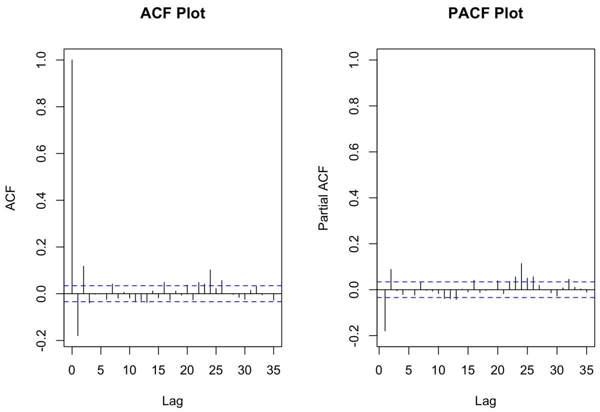


Exploratory Data Analysis

- 1st Check for Stationarity

Exploratory Data Analysis

- 1st Check for Stationarity



Exploratory Data Analysis

- 2nd Check for Stationarity

Exploratory Data Analysis

- 2nd Check for Stationarity

```
kpss.test(data_subset$rel_change)
```

```
> kpss.test(data_subset$rel_change)

        KPSS Test for Level Stationarity

data:  data_subset$rel_change
KPSS Level = 2.4255, Truncation lag parameter = 9, p-value = 0.01
```

- Stationarity is not achieved

Exploratory Data Analysis

- 3rd Check for Stationarity

Exploratory Data Analysis

- 3rd Check for Stationarity

```
kpss.test(diff(data_subset$rel_change,1)))
```

```
> kpss.test(diff(data_subset$rel_change,1))

        KPSS Test for Level Stationarity

data:  diff(data_subset$rel_change, 1)
KPSS Level = 0.0035198, Truncation lag parameter = 9, p-value = 0.1
```

- Stationarity is now achieved after taking 1 difference

Exploratory Data Analysis

- Checking for Non-Constant Variance

Exploratory Data Analysis

- Checking for Non-Constant Variance
- Data was broken up into 5 even groups (658 time stamps each)
- Which gave us the following output:

Exploratory Data Analysis

- Checking for Non-Constant Variance
- Data was broken up into 5 even groups (658 time stamps each)
- Which gave us the following output:

```
> bartlett.test(Appliances ~ groups,data = data_subset)

        Bartlett test of homogeneity of variances

data:  Appliances by groups
Bartlett's K-squared = 95.372, df = 4, p-value < 2.2e-16
```

Brief History of ARCH/GARCH

Brief History of ARCH/GARCH

- Early Time Series modeling focuses on modeling the mean of a time series

Brief History of ARCH/GARCH

- Early Time Series modeling focuses on modeling the mean of a time series
- Since 1980s, there's been an increase in focus on volatility and what factors play into volatility

Brief History of ARCH/GARCH

- Early Time Series modeling focuses on modeling the mean of a time series
- Since 1980s, there's been an increase in focus on volatility and what factors play into volatility
- The most popular of these models is ARCH/GARCH

Many Flavors of ARCH

Many Flavors of ARCH

GARCH, AARCH, ADCC-GARCH, AGARCH, ANN_GARCH, APARCH, ARCH-M, ARCH_SM, AT-GARCH, AugGARCH, LMGARCH, RGARCH, GQARCH, GARJI, Flex-GGARCH, DVEC-GARCH, reGARCH, weak GARCH, t-GARCH, Spline GARCH, GARCH-X, MGARCH, FIGARCH,Matrix GARCH, VCCGARCH, ZARCH,SPARCH, SWARCH, QARCH, NGARCH, NAGARCH, IGARCH, EGARCH, GARCH-M, QGARCH, GJR-GARCH, fGARCH, COGARCH, Spatial GARCH, ZD-GARCH, Gaussian Process GARCH.

Many Flavors of ARCH

GARCH, AARCH, ADCC-GARCH, AGARCH, ANN_GARCH, APARCH, ARCH-M, ARCH_SM, AT-GARCH, AugGARCH, LMGARCH, RGARCH, GQARCH, GARJI, Flex-GGARCH, DVEC-GARCH, reGARCH, weak GARCH, t-GARCH, Spline GARCH, GARCH-X, MGARCH, FIGARCH,Matrix GARCH, VCCGARCH, ZARCH,SPARCH, SWARCH, QARCH, NGARCH, NAGARCH, IGARCH, EGARCH, GARCH-M, QGARCH, GJR-GARCH, fGARCH, COGARCH, Spatial GARCH, ZD-GARCH, Gaussian Process GARCH.

- We will be exploring Plain Vanilla ARCH!

Introducing ARCH

Introducing ARCH

- ARCH - AutoRegressive Conditional Heteroskedasticity

Introducing ARCH

- ARCH - AutoRegressive Conditional Heteroskedasticity
- An autoregressive time series approach to modeling changing volatility (in this case, the rate of change in electricity per hour)

Introducing ARCH

- ARCH - AutoRegressive Conditional Heteroskedasticity
- An autoregressive time series approach to modeling changing volatility (in this case, the rate of change in electricity per hour)
- The ARCH model specifies that the variance of the current error term depends on error of previous periods

ARCH Introduction

- After fitting a mean process to our time series such as one of the Box-Jenkens methods we have been learning in this class, the residuals can be split into a white noise piece and a time dependent standard deviation.

ARCH Introduction

- After fitting a mean process to our time series such as one of the Box-Jenkens methods we have been learning in this class, the residuals can be split into a white noise piece and a time dependent standard deviation.

$$\epsilon_t = w_t \sigma_t$$

where

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2$$

ARCH Introduction

- After fitting a mean process to our time series such as one of the Box-Jenkens methods we have been learning in this class, the residuals can be split into a white noise piece and a time dependent standard deviation.

$$\epsilon_t = w_t \sigma_t$$

where

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2$$

- Why do we use AutoRegressive with ϵ_{t-1}^2 instead of σ_{t-1}^2?

ARCH Introduction

$$Var(\epsilon_t) = \sigma_t^2 = \frac{1}{t}\sum_{i=1}^{t}(\epsilon_t - \bar{\epsilon})^2$$

ARCH Introduction

$$Var(\epsilon_t) = \sigma_t^2 = \frac{1}{t}\sum_{i=1}^{t}(\epsilon_t - \bar{\epsilon})^2$$

- A core assumption of this model is that the average value of ϵ is 0. Thus for a single time point:

ARCH Introduction

$$Var(\epsilon_t) = \sigma_t^2 = \frac{1}{t}\sum_{i=1}^{t}(\epsilon_t - \bar{\epsilon})^2$$

- A core assumption of this model is that the average value of ϵ is 0. Thus for a single time point:

$$\sigma_t^2 = \frac{1}{1}\sum_{i=1}^{1}(\epsilon_t - 0)^2 = \epsilon_t^2$$

ARCH Introduction

- Thus, now have the following:

ARCH Introduction

- Thus, now have the following:

$$
\begin{aligned}
\sigma_t^2 &= \alpha_0 + \alpha_1 \epsilon_{t-1}^2 \\
\epsilon_t &= w_t \sigma_t \\
&= w_t \sqrt{\alpha_0 + \alpha_1 \epsilon_{t-1}^2}
\end{aligned}
$$

ARCH Introduction

- Thus, now have the following:

$$\begin{aligned} \sigma_t^2 &= \alpha_0 + \alpha_1 \epsilon_{t-1}^2 \\ \epsilon_t &= w_t \sigma_t \\ &= w_t \sqrt{\alpha_0 + \alpha_1 \epsilon_{t-1}^2} \end{aligned}$$

- where w_t is white noise.

Additional ARCH Assumptions

Addtl. ARCH Assumptions

1. Time Series of interest has weak stationarity
2. Conditional Variance
3. Residuals in mean equation should not have autocorrelation
4. Mean of the time series can be models with ARMA Process
5. Moments of each distribution are finite
6. Parameters of the Cond. Variance equation are non negative, $\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2$
7. Normality of white noise

GARCH Introduction

GARCH Introduction

- Generalized AutoRegressive Conditional Heteroskedasticity

GARCH Introduction

- Generalized AutoRegressive Conditional Heteroskedasticity
- The same as ARCH but we are adding the estimated volatility from p previous time points

GARCH Introduction

- Generalized AutoRegressive Conditional Heteroskedasticity
- The same as ARCH but we are adding the estimated volatility from p previous time points

$$\sigma_t^2 = \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \hat{\sigma}_{t-1}^2$$

$$\epsilon_t = w_t \sigma_t = w_t \sqrt{\alpha_0 + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \hat{\sigma}_{t-1}^2}$$

Modeling Our Data

Modeling Our Data

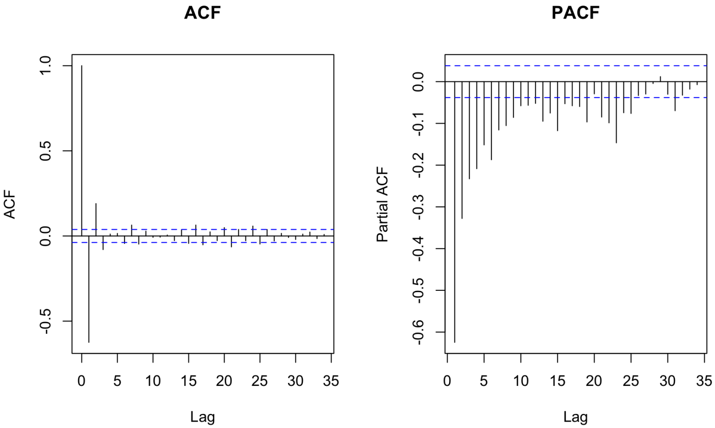
- Split data into training and testing

Modeling Our Data

- Split data into training and testing
- Train: About 3 months
- Test: Last 27 days

Modeling Our Data

- ACF and PACF of Differenced Data



Parameter Selection

MODEL 1

```
*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
------------------------------------
GARCH Model              : sGARCH(1,1)
Variance Targeting       : FALSE

Conditional Mean Dynamics
------------------------------------
Mean Model               : ARFIMA(1,0,0)
Include Mean             : TRUE
GARCH-in-Mean            : FALSE

Conditional Distribution
------------------------------------
Distribution    :  norm
Includes Skew   :  FALSE
Includes Shape  :  FALSE
Includes Lambda :  FALSE
```

MODEL 2

```
*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
------------------------------------
GARCH Model              : sGARCH(1,1)
Variance Targeting       : FALSE

Conditional Mean Dynamics
------------------------------------
Mean Model               : ARFIMA(1,0,2)
Include Mean             : TRUE
GARCH-in-Mean            : FALSE

Conditional Distribution
------------------------------------
Distribution    :  norm
Includes Skew   :  FALSE
Includes Shape  :  FALSE
Includes Lambda :  FALSE
```

MODEL 3

```
*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
------------------------------------
GARCH Model              : sGARCH(1,1)
Variance Targeting       : FALSE

Conditional Mean Dynamics
------------------------------------
Mean Model               : ARFIMA(1,0,3)
Include Mean             : TRUE
GARCH-in-Mean            : FALSE

Conditional Distribution
------------------------------------
Distribution    :  norm
Includes Skew   :  FALSE
Includes Shape  :  FALSE
Includes Lambda :  FALSE
```

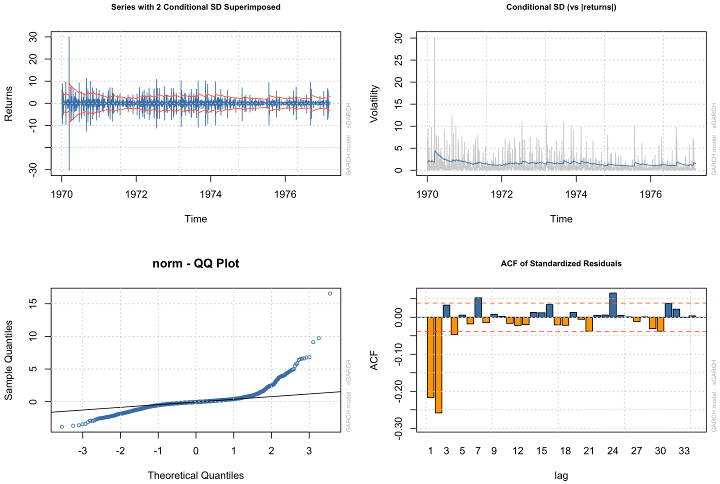
MODEL 4

```
*---------------------------------*
*       GARCH Model Spec          *
*---------------------------------*

Conditional Variance Dynamics
------------------------------------
GARCH Model              : sGARCH(1,1)
Variance Targeting       : FALSE

Conditional Mean Dynamics
------------------------------------
Mean Model               : ARFIMA(10,0,2)
Include Mean             : TRUE
GARCH-in-Mean            : FALSE

Conditional Distribution
------------------------------------
Distribution    :  norm
Includes Skew   :  FALSE
Includes Shape  :  FALSE
Includes Lambda :  FALSE
```

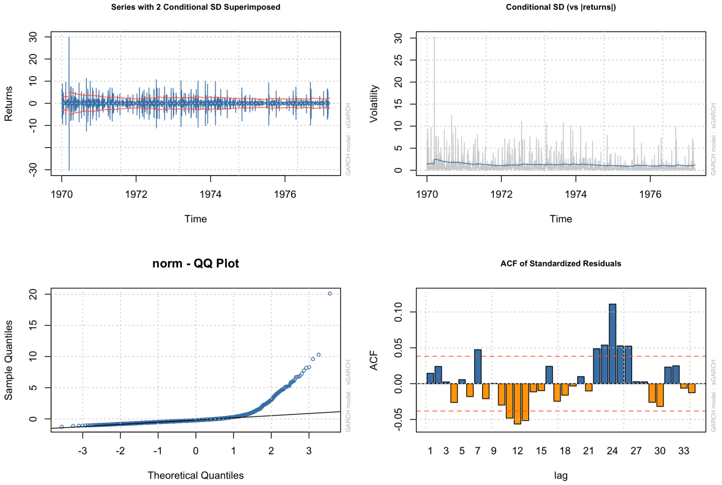
Model Residual Plots - The Code

```
test_garch1 <- ugarchspec(mean.model = list(armaOrder = c(1, 0)),
                          variance.model = list(model = "sGARCH"))
garch_model1 <- ugarchfit(spec = test_garch1,
                          data = train_rel_change_diff)
```

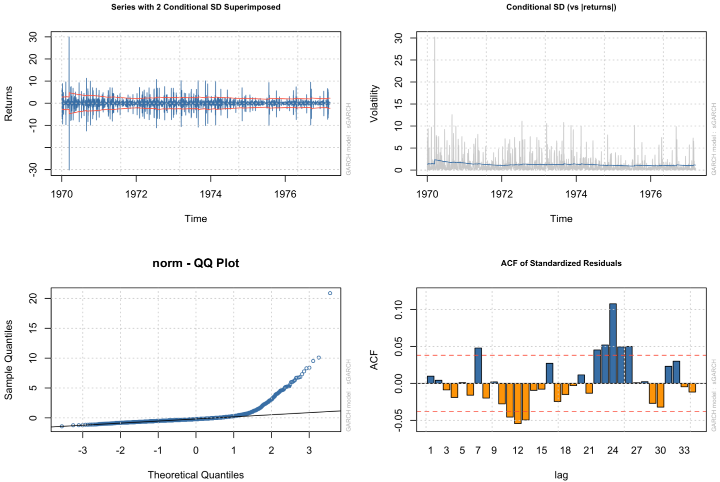
Model 1 Residual Plots - ARFIMA(1,0,0)



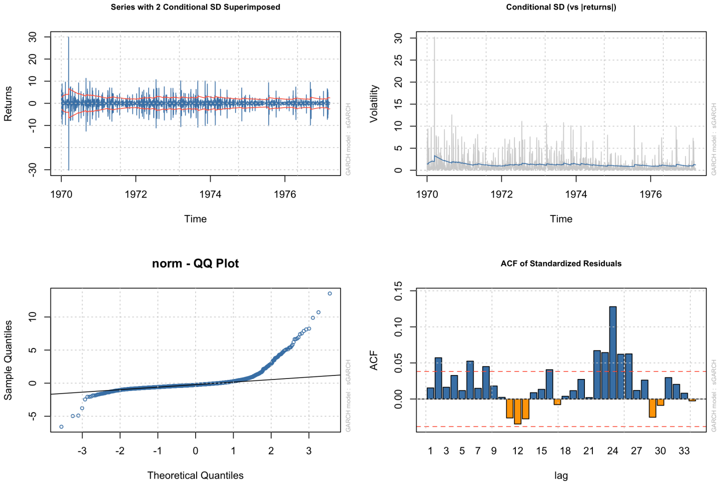
Model 2 Residual Plots - ARFIMA(1,0,2)



Model 3 Residual Plots - ARFIMA(1,0,3)



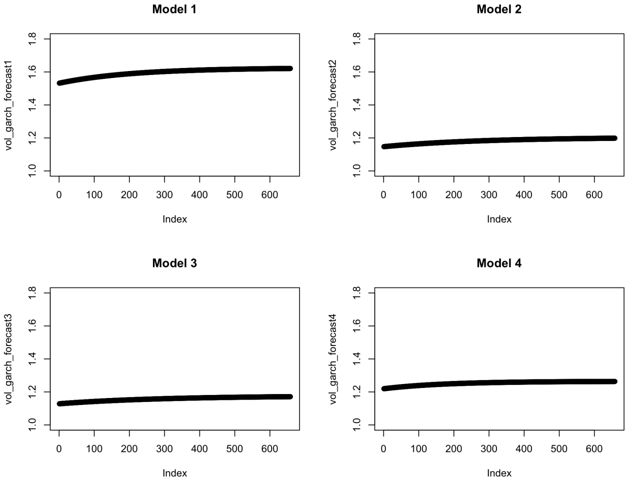
Model 4 Residual Plots - ARFIMA(10,0,2)



Forecasting Our Models - The Code

```
#forecast on test data, with the same length as our test data
garch_forecast1 <- ugarchforecast(garch_model1, n.ahead = length(test_rel_change_diff))
#extract the fitted sigma from the forecast
vol_garch_forecast1 <- garch_forecast1@forecast[["sigmaFor"]]
```

Forecasting Our Models



SSE for Models

- SSE for Model 1: 1148.399
- SSE for Model 2: 1148.197
- SSE for Model 3: 1148.202
- SSE for Model 4: 1148.145

Advanced Forecasting Techniques

Advanced Forecasting Techniques

- In this class, we have used traditional forecasting techniques.

Advanced Forecasting Techniques

- In this class, we have used traditional forecasting techniques.
- In this project, we will use our best GARCH model and introduce an advanced forecasting method:

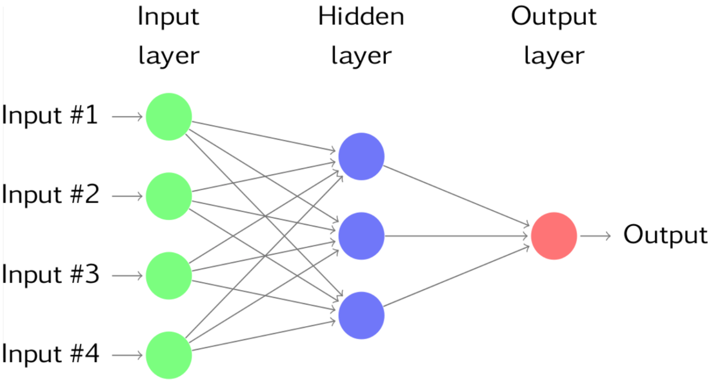
Advanced Forecasting Techniques

- In this class, we have used traditional forecasting techniques.
- In this project, we will use our best GARCH model and introduce an advanced forecasting method:
- Neural Network Autoregression (NNAR)

Advanced Forecasting Techniques

- In this class, we have used traditional forecasting techniques.
- In this project, we will use our best GARCH model and introduce an advanced forecasting method:
- Neural Network Autoregression (NNAR)
- The big question: Are the NNAR forecasting methods consistent with the traditional GARCH forecasting?

Neural Network Introduction and Terminology



Objective Function

Inputs to hidden neuron j linearly combined:

$$z_j = b_j + \sum_{i=1}^{4} w_{ij} x_j$$

Modified using nonlinear function such as a sigmoid:

$$s(z) = \frac{1}{1 + e^{-z}}$$

- Inputs form a linear combination in which the b's and weights w's are found from "learning" the data.
- Nonlinear function in the hidden layer transforms inputs to reduce the number of outliers from data.

The NNAR(p,k) Forecasting Model

The NNAR(p,k) Forecasting Model

- The lagged values of a time series model will be used as inputs.

The NNAR(p,k) Forecasting Model

- The lagged values of a time series model will be used as inputs.
- P is the number of lagged outputs.

The NNAR(p,k) Forecasting Model

- The lagged values of a time series model will be used as inputs.
- P is the number of lagged outputs.
- K is the number of neuronal nodes in the hidden layer.

The NNAR(p,k) Forecasting Model

- The lagged values of a time series model will be used as inputs.
- P is the number of lagged outputs.
- K is the number of neuronal nodes in the hidden layer.
- Can be extended to NNAR(p,P,k)[m] to handle stationarity.

The NNAR(p,k) Forecasting Model

- The lagged values of a time series model will be used as inputs.
- P is the number of lagged outputs.
- K is the number of neuronal nodes in the hidden layer.
- Can be extended to NNAR(p,P,k)[m] to handle stationarity.

Fun fact: An NNAR(p,0) model is equivalent to an ARIMA(p,0,0) model, which you have seen many times in this course before.

R Implementation of NNAR Forecasting

```
neuralnet_energy = nnetar(diff_garch_series4, lambda = 0)
autoplot(forecast(neuralnet_energy, h=30))
```