High-Frequency Data Analysis and Market Microstructure

GROUP 9 PROJECT REPORT

Members:
Shruti Agarwal, Dev Priya Goel, Trikay Nalamada,
Divyansh Mangal, Mahfoozur Rahman Khan

[November 22, 2020]

Disclaimer

The content of this report and the product made is only meant for the learning process as part of the course. This is not for use in making publication or making commercialization without the mentor's consent. My contribution won't demand any claim in the future for further progress of Mentor's development and innovation along the direction unless there is a special continuous involvement

High Frequency Data

High-frequency trading extends negotiations to programs that are usually executed by computational algorithms molding a set of trading orders at high speeds and increasing market liquidity.
The concept of market liquidity is defined as the immediate ability to buy or sell a huge amount of stocks without significant effect to its price. Liquidity gives us information about the ability of the market to convert an asset into cash, or vice-versa. The term high frequency refers to the increase of market liquidity when compared to manual negotiations. High frequency trading intensifies market liquidity. High frequency trading operations generally use sophisticated algorithms to conduct analysis of numerous markets and execute multiple orders arbitrage strategies.

Auto-Regressive Conditional Duration Model

It is a duration model. Duration models look at time duration between trades. A long-duration would indicate that there were no trading activities in that duration. So, we get no new information in that period. Consider **adjusted time duration** $(\Delta t_i^{\,*})$ given by

$$\Delta t_i^{\,*} = \Delta t_i / f(t_i)$$

Here, $f(t_i)$ is the cyclic component of Δt_i^*. $f(t_i)$ depends on the underlying asset and on the market behavior. A commonly used approach for estimating $f(t_i)$ is a smoothing spline.

Auto-Regressive Conditional Duration (ACD) models (Engle and Russell, 1998) are used to describe the evolution of time duration in heavy-tailed stocks using the ideation behind GARCH.

The ACD model is most conveniently specified in terms of the conditional density of the durations. For ease define notation $x_i = \Delta t_i^*$ (the interval between two arrival times), which will be called the duration. Also, let the conditional expectation of the adjusted duration between i-th and $i-1$-th trades given the information up to trade (represented by F_{i-1}) given by $\psi_i = E(x_i \mid F_{i-1})$. Consequently, the **ACD model** is defined by

$$x_i = \psi_i \varepsilon_i$$

Here, $\{\varepsilon_i\}$ is a sequence of non-negative independent and identically distributed random variables with $E[\varepsilon_i] = 1$. Generally, ε_i's are assumed to have either standard exponential or standardized Weibull distribution. Also, ψ_i's have the form

$$\psi_i = \omega + \sum_{j=1}^{r} \gamma_j x_{i-j} + \sum_{j=1}^{r} \omega_j \psi_{i-j}$$

This defines the ACD (r, s) model. When ε_i follow a Weibull distribution the model is referred to as WACD. Similarly, it is called EACD (r, s) when ε_i are distributed exponentially.

Parallel to GARCH models the sequence, we define $\eta_i = x_i \psi_i$. Now, η_i is a martingale difference sequence with $E[\eta_i | F_{i-1}] = 0$. As a result, the ACD (r, s) model is given by

$$\psi_i = \omega + \sum_{j=1}^{max(r,s)} (\gamma_j + \omega_j) x_{i-j} - \sum_{j=1}^{s} \omega_j \eta_{i-j} + \eta_j$$

Under the assumption that $\gamma_j = 0$ for $j > r$ and $\omega_j = 0$ for $j > s$. For calculating $E(x_i)$ we assume weak stationarity and take expectation on both sides to get

$$E(x_i) = \frac{\omega}{1 - \sum_{j=1}^{max(r,s)} (\gamma_j + \omega_j)}$$

Note that $\omega > 0$ and $1 > (\gamma_j + \omega_j)$ since expectation is positive

Hazard Function

A useful concept in modeling duration is the hazard function implied by a distribution function. For a random variable X, the survival function is defined as

$$S(x) \equiv P(X > x) = 1 - P(X \leq x) = 1 - CDF(x), \ x > 0,$$

which gives the probability that a subject, which follows the distribution of X, survives at the time x.
The hazard function (or intensity function) of X is then defined by

$$h(x) = \frac{f(x)}{S(x)}$$

where $f(.)$ and $S(.)$ are the pdf and survival function of X, respectively.

Testing the model

LM test of no Remaining ACD (testRmACD)

Tests if there is any remaining ACD structure in the residuals. If the model is correctly specified, the error terms are independent and accordingly the residuals should not show any further ACD structured dependency. We can run this test on our fitted model by calling

```
testRmACD(fitModel, pStar = 2, robust = TRUE)
```

Here pStar is the number of α -parameters in the possibly remaining ACD structure and letting robust = TRUE makes the test robust to possible misspecifications of the error term
Ref: Meitz, M. and Terasvirta, T. (2006). Evaluating models of autoregressive conditional duration. Journal of Business and Economic Statistics 24: 104-124.

LM test against Smooth Transition ACD (testSTACD)

Tests if the alpha parameters and the constant should be varying with the value of the lagged durations, according to a logistic transition function. The linearity imposed by the standard ACD(p, q) model may not be enough to describe the dynamics of durations.

ACD Model - Implementation in R

We have primarily used the R package “ACDm” for fitting the ACD model and several extensions of it. This can be installed using the command

```
install.packages("ACDm")
```

The dataset we have used in our models can be found here.
IBM transactions data (11/1/90-1/31/91)
The columns are date/time, volume, bid quote, ask quote, and transaction price.
Download link → ibm.txt
Following some data cleaning, we then went on to fit the models - EACD(1,1), WACD(1,1) and GACD(1,1) on the data set.

First we import the data from the file the txt file

```
transIBM <- read.csv("ibm.txt", header = FALSE, sep= "")
names(transIBM) <- c("time", "Volume", "Bid Price", "Ask Price","Price")
head(transIBM, 11)
```

We split the timestamps into the date and time components

```r
# Splitting timestamps into date and time parts
time <- transIBM[ , 1]
time <- substr(time, 1, 6)
```

Split it into year, month and date as a string which we then converted to POSIXlt format

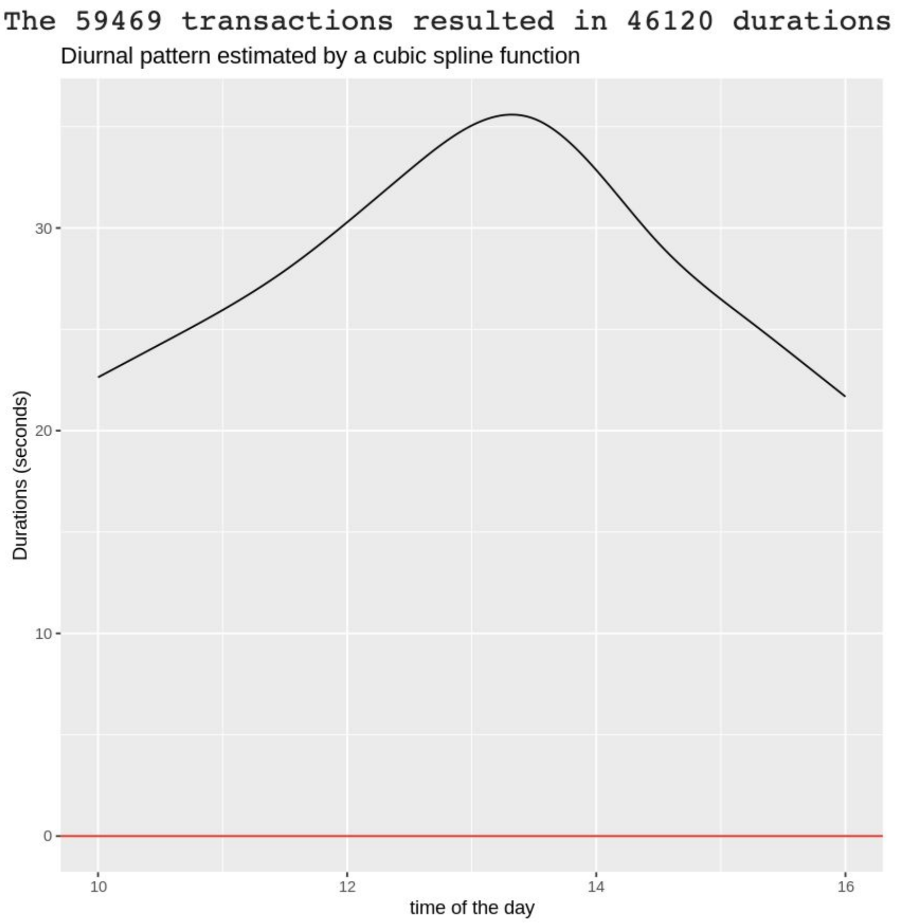
```r
time <- strptime(time, format = "%y%m%d")
time <- time + as.numeric(substr(transIBM[ ,1], 7, 11))
transIBM$time <- as.POSIXlt(time)
```

We then moved on to remove the two days with no trade

```r
# Removing dates November 23 and December 27 with no trades
transIBM <- transIBM[transIBM[ ,1]$yday != strptime("901227", format
="%y%m%d")$yday, ]
transIBM <- transIBM[transIBM[ ,1]$yday != strptime("901123", format
="%y%m%d")$yday, ]
```

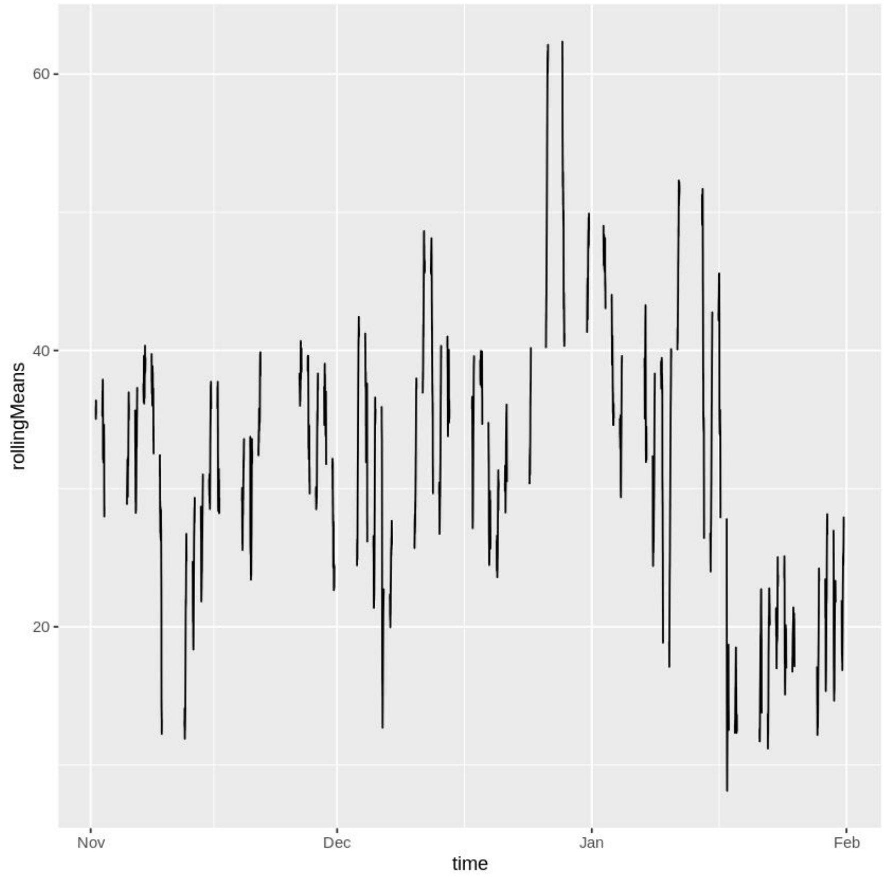
This was followed by removing the trades at the end of trade cycle and computing adjusted time duration (defined above) using a cubic spline

```r
# Removing trades at the end of day cycle and performing diurnal adjustments
DurationsIBM <- computeDurations(transIBM, open = "10:00:00", close
="16:00:00")
# Obtaining diurnally adjusted duration by removing daily seasonal variation to
get a stationary duration process
AdjDurationsIBM <- diurnalAdj(DurationsIBM, aggregation = "all", method=
"cubicSpline", nodes = c(seq(600, 900, 60), 930, 960))
```

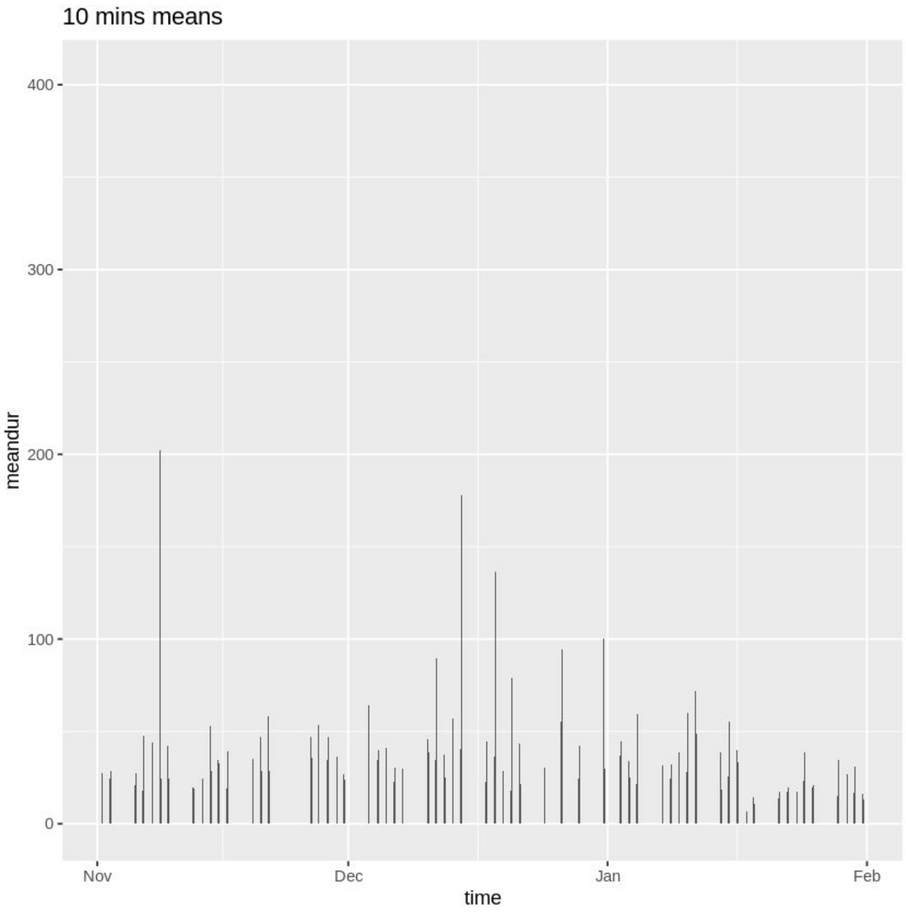


We plot the rolling mean for the obtained durations by

```
# Plotting the Rolling Mean for the Duration
plotRollMeanAcd(DurationsIBM, window = 500)
```

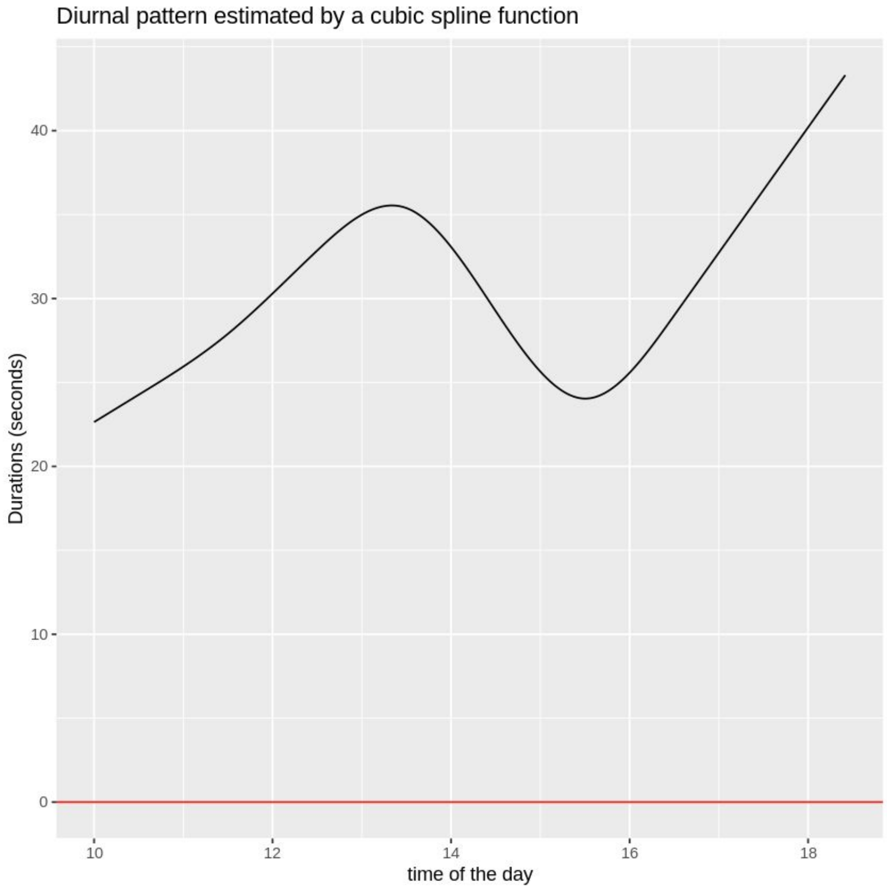


```
# Plotting mean duration over a specified interval length in a bar plot
plotHistAcd(DurationsIBM, windowunit = "mins", window = 10)
```

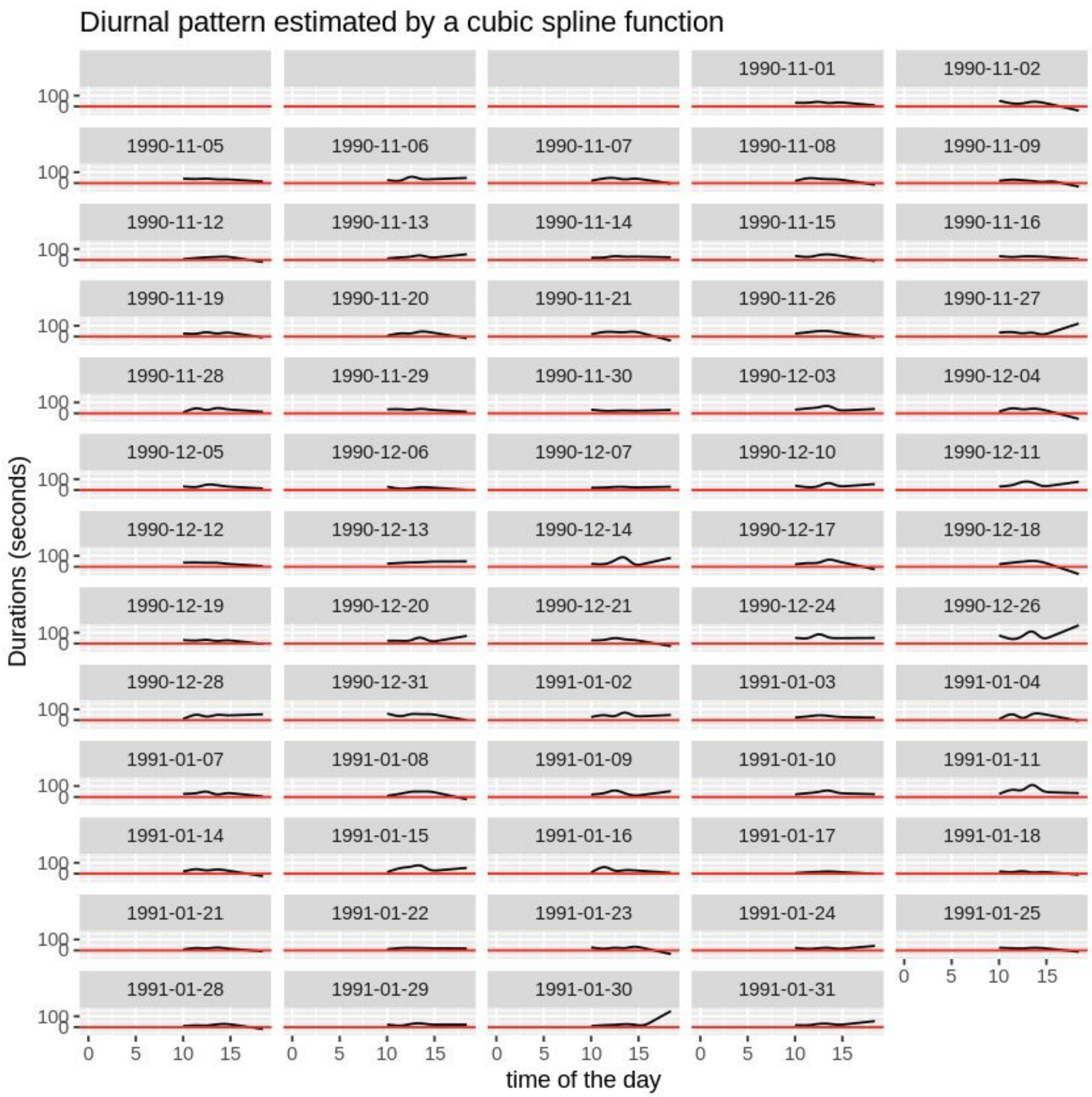


We then compared and contrasted various ways of obtaining Adjusted Durations for various durations

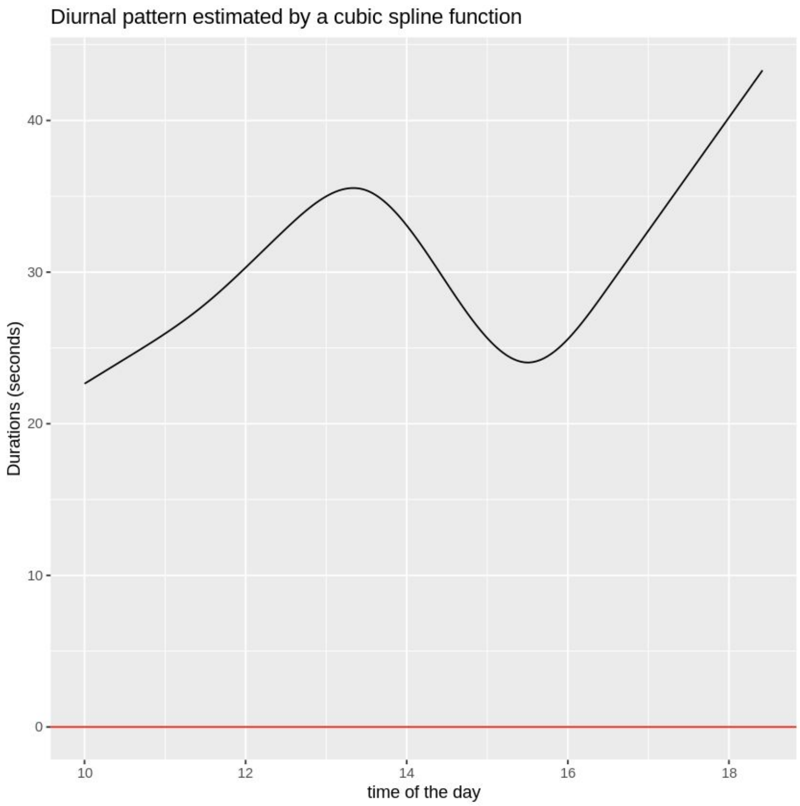
```
diurnalAdj(DurationsIBM)
```



```
diurnalAdj(DurationsIBM, aggregation = "none")
```



```
adjDurData <- diurnalAdj(DurationsIBM, aggregation = "all")
```



Model EACD(1,1)

EACD is a specific type of ACD model of order (1,1) where in the ε_i are iid random variables following the standard exponential distribution.
The model is as follows:

$$x_i = \psi_i \varepsilon_i \qquad \psi_i = \omega + \gamma_1 x_{i-1} + \omega_1 \psi_{i-1}$$

Due to the standard exponential, we have $E(\varepsilon_i) = Var(\varepsilon_i) = 1$
Further,

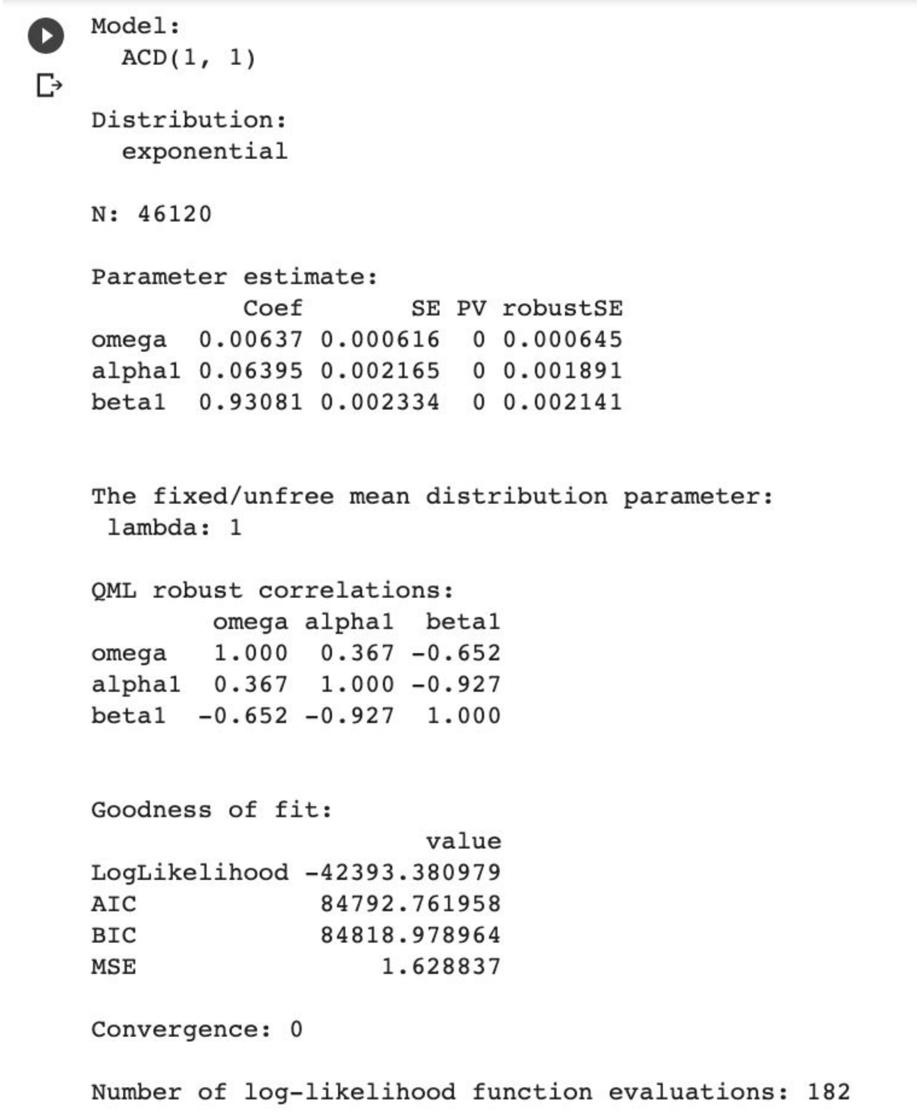
$$\mu_x = E(x_i) = \frac{\omega}{1-\gamma_1-\omega_1}$$

$$Var(x_i) = 2E(\psi^2{}_i) - \mu^2{}_x = \mu^2{}_x \times \frac{1-\omega^2{}_1-2\gamma_1\omega_1}{1-\omega^2{}_1-2\gamma_1\omega_1-2\gamma_1{}^2}$$

We proceed to fit a EACD(1,1) to our data using the acdFit command as shown below:

```
fitModel <- acdFit(durations = adjDurData, model = "ACD", dist = "exponential",
order =c(1,1), dailyRestart = 1)
```

Specifying the parameters dist = ‘exponential’ and order = c(1,1) ensures that it is a EACD(1,1) model. Results of the run are as below:

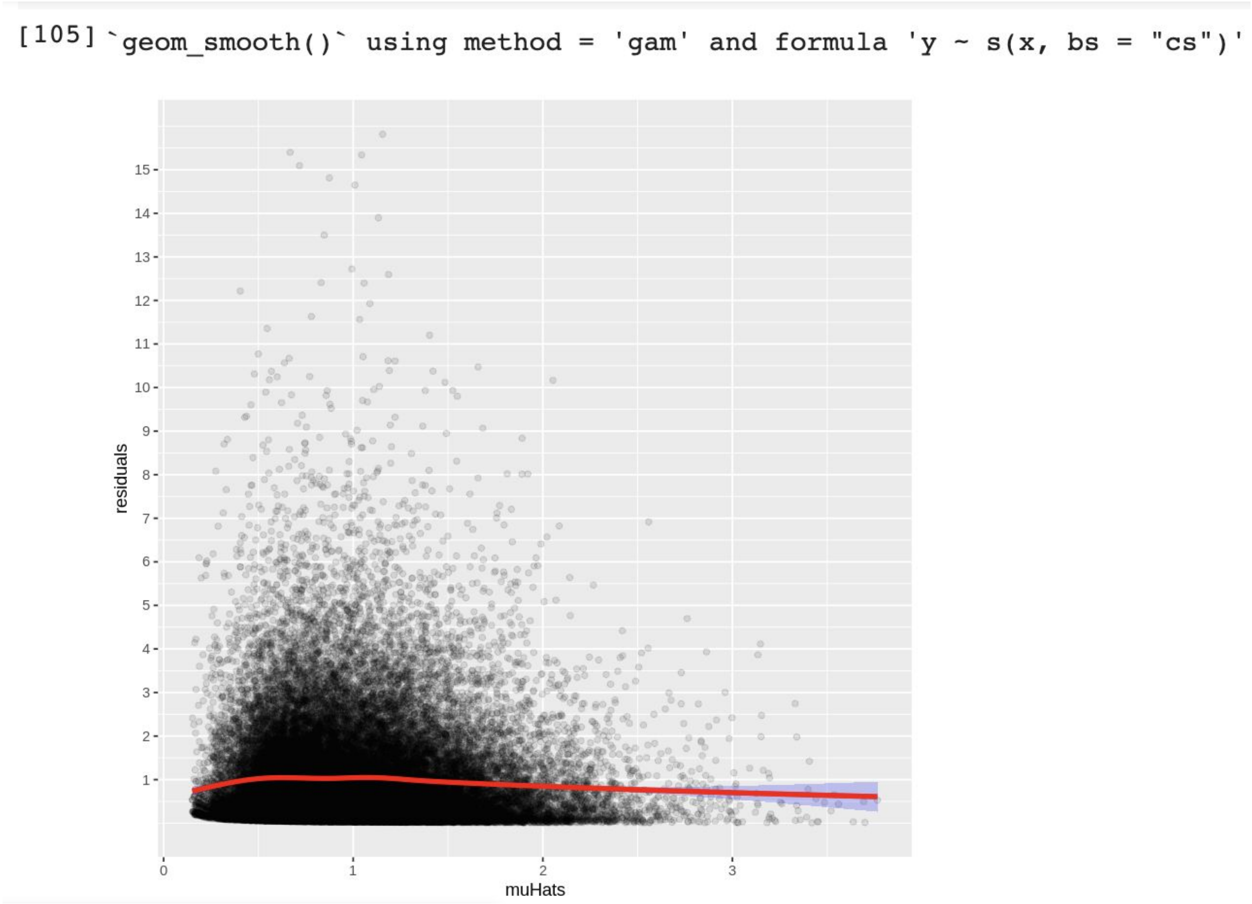


```
Model:
  ACD(1, 1)

Distribution:
  exponential

N: 46120

Parameter estimate:
          Coef         SE PV robustSE
omega  0.00637 0.000616  0 0.000645
alpha1 0.06395 0.002165  0 0.001891
beta1  0.93081 0.002334  0 0.002141


The fixed/unfree mean distribution parameter:
 lambda: 1

QML robust correlations:
        omega alpha1  beta1
omega   1.000  0.367 -0.652
alpha1  0.367  1.000 -0.927
beta1  -0.652 -0.927  1.000


Goodness of fit:
                     value
LogLikelihood -42393.380979
AIC            84792.761958
BIC            84818.978964
MSE                1.628837

Convergence: 0

Number of log-likelihood function evaluations: 182
```

Quasi maximum likelihood was used to obtain fitted parameters. Alpha1 and beta1 play the role of γ_1 and ω_1 . A convergence value of 0 indicates that the process converged to a local maxima.

Next we plot the estimated conditional means against the residuals. In theory, residuals are independent of conditional means, so we should see the mean of residual to be constant across various means.

```
plotScatterAcd(fitModel, x = "muHats", y = "residuals", colour = NULL,
ylag = 0, xlim = NULL, ylim = NULL, alpha = 1/10, smoothMethod =
"auto")
```

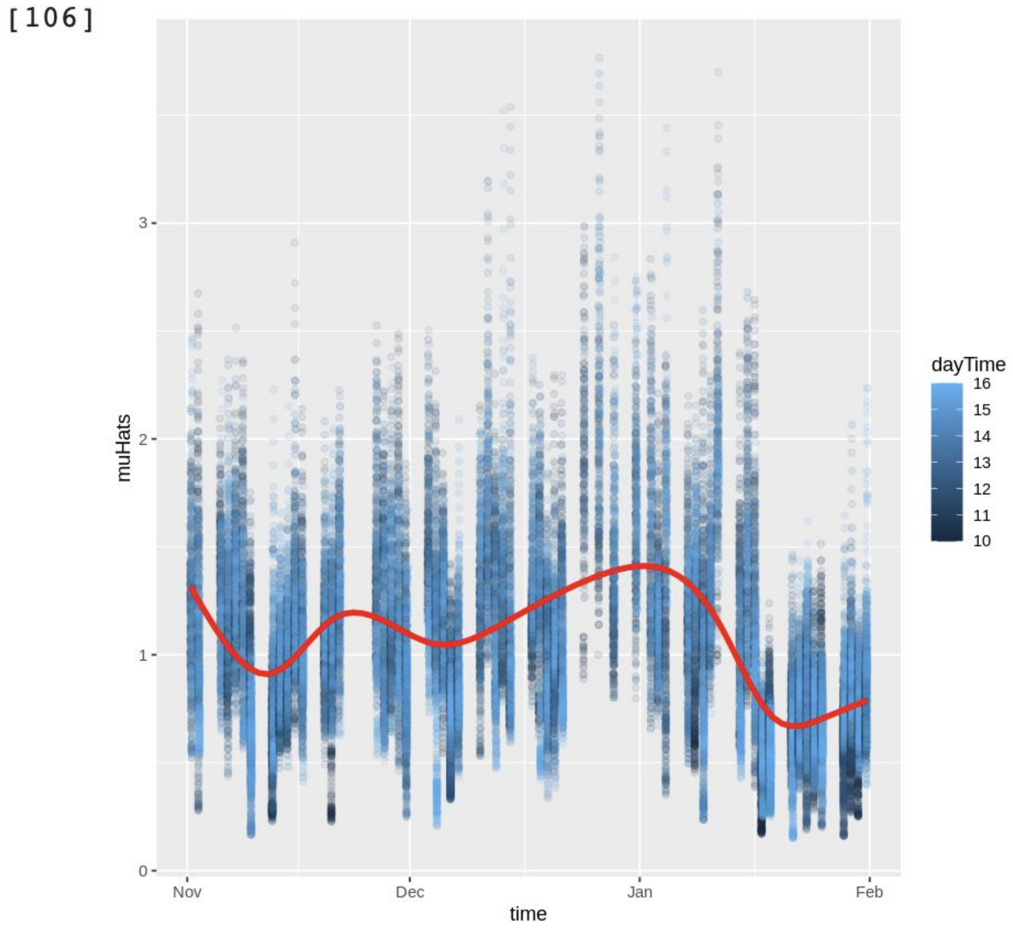


```
[105] `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

The mean is nearly constant except for tails where it dips a bit, indicating that the estimated means were overpredicted at the end points.

Next we see how the estimated conditional means vary with different times in a day.

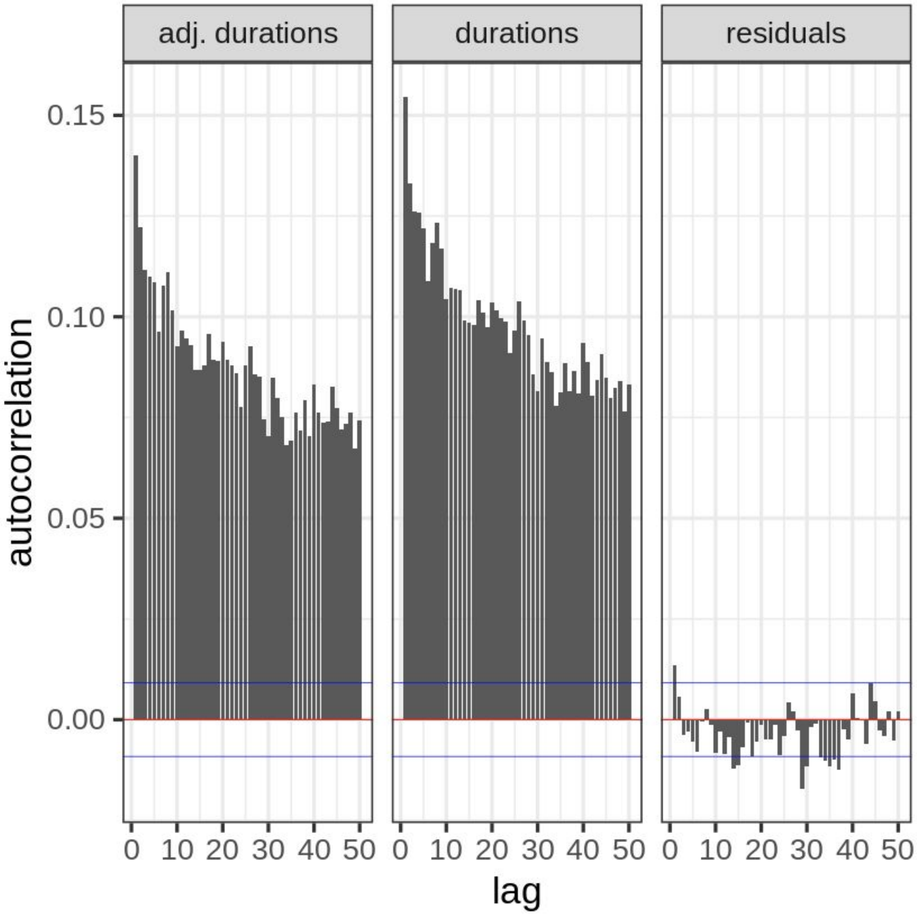
```
plotScatterAcd(fitModel, x = "time", y = "muHats", colour = "dayTime",
ylag = 0, xlim = NULL, ylim = NULL, alpha = 1/10, smoothMethod =
"auto")
```



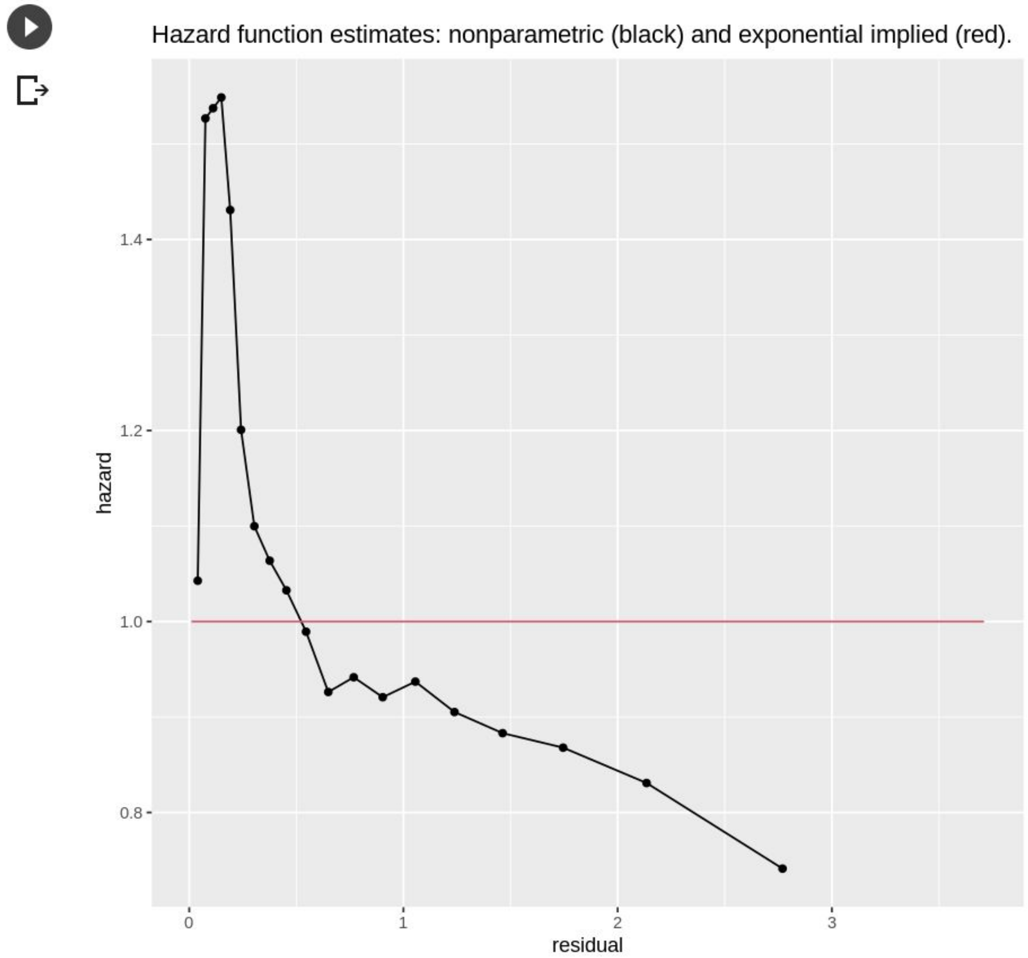
The red line represents the mean across various times of the day. We notice that earlier times usually result in a smaller estimated mean.

We next plot the acf of the duration, adjusted duration and residuals to notice heavy serial correlation of durations. The ACF is slowly decreasing with lag. Blue line represent 95% confidence interval

```
acf_acd(fitModel, conf_level = 0.95, max = 50)
```



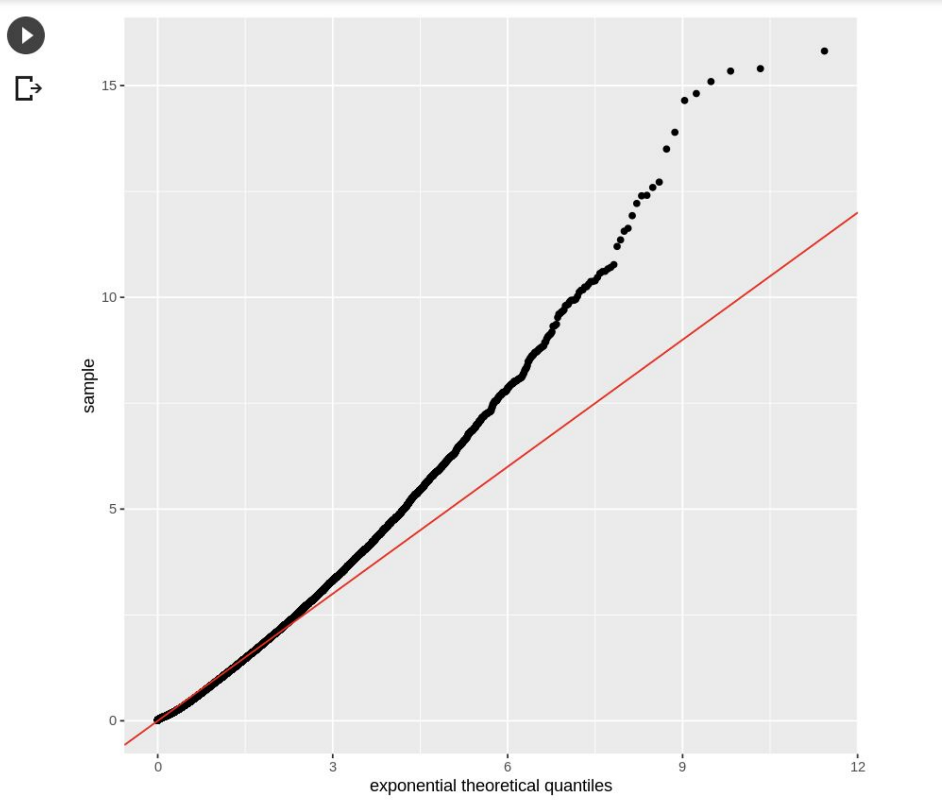
Another way of evaluating our model is by looking at how well the hazard functions match:



A poor match between the estimated and implied functions means that our initial choice of selecting the errors to be standard exponentially distributed was a bad one. We try other distributions in the models listed below.

Finally we display the qqplots for completion

```
qqplotAcd(fitModel)
```



The sample and theoretical quantiles match only for a low value, which again denotes a bad choice of distribution.

Model WACD(1,1)

WACD(1,1) is an ACD model wherein the ε_i are independent and identically distributed random variables following the Weibull distribution.
The model is as follows:

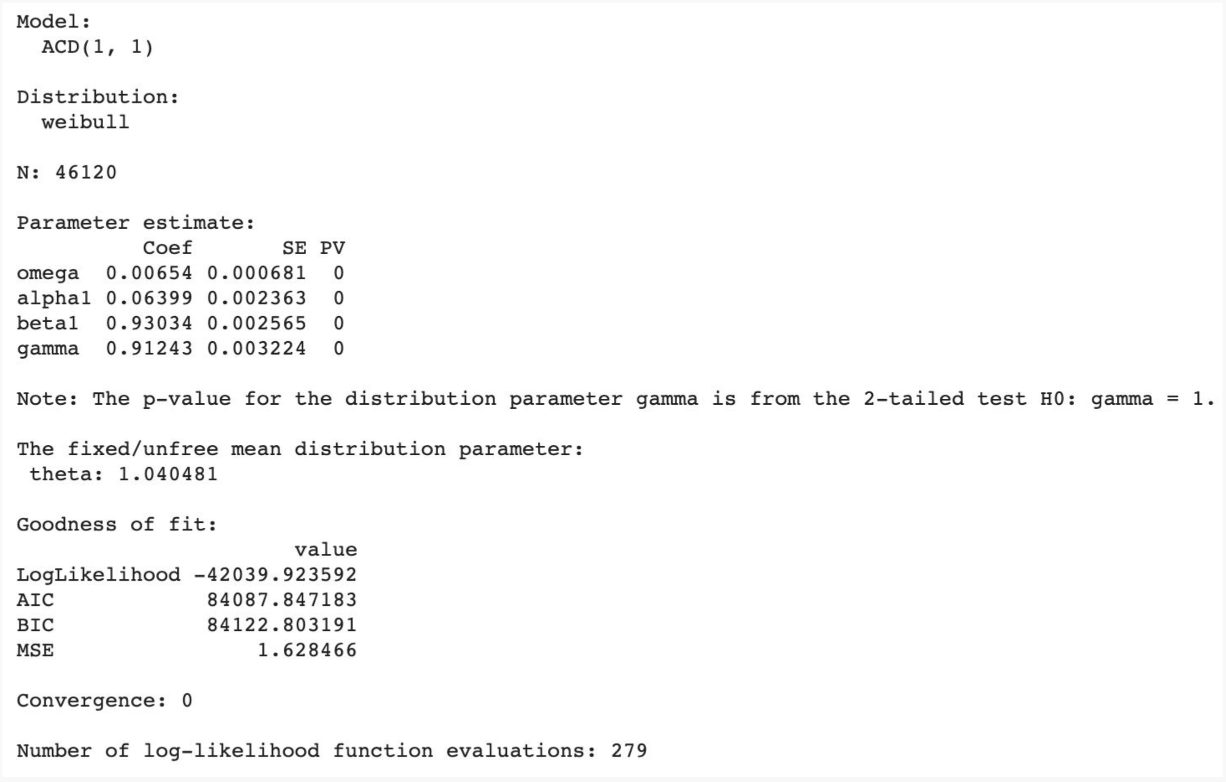
$$x_i = \psi_i \varepsilon_i \qquad \psi_i = \omega + \gamma_1 x_{i-1} + \omega_1 \psi_{i-1}$$

Where ε_i is Weibull distributed with mean $E(\varepsilon_i) = 1$

We proceed to fit a WACD(1,1) to our data using the acdFit command as shown below:

```
fitModel <- acdFit(durations =adjDurData, model = "ACD", dist = "weibull",
order =c(1,1), dailyRestart = 1)
```

Specifying the parameters dist = ‘’weibull’’ and order = c(1,1) ensures that it is a WACD(1,1) model. Results of the run are as below:



```
Model:
  ACD(1, 1)

Distribution:
  weibull

N: 46120

Parameter estimate:
          Coef       SE PV
omega  0.00654 0.000681  0
alpha1 0.06399 0.002363  0
beta1  0.93034 0.002565  0
gamma  0.91243 0.003224  0

Note: The p-value for the distribution parameter gamma is from the 2-tailed test H0: gamma = 1.

The fixed/unfree mean distribution parameter:
 theta: 1.040481

Goodness of fit:
                      value
LogLikelihood -42039.923592
AIC            84087.847183
BIC            84122.803191
MSE                1.628466

Convergence: 0

Number of log-likelihood function evaluations: 279
```

Following this, we repeat the same process as above. The code for this can be found in the corresponding Notebook.

Model GACD(1,1)

GACD(1,1) is an ACD model wherein the ε_i are independent and identically distributed random variables following the generalized gamma distribution also called as gengamma distribution.
The model is as follows

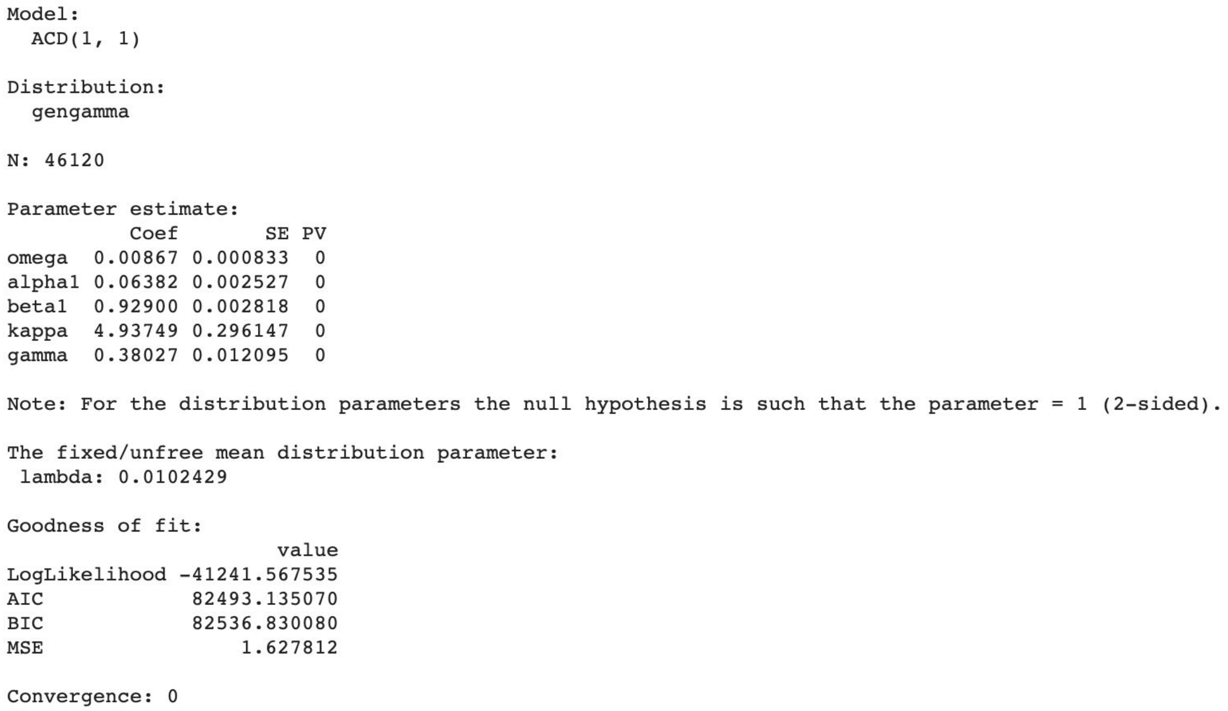
$$x_i = \psi_i \varepsilon_i \qquad \psi_i = \omega + \gamma_1 x_{i-1} + \omega_1 \psi_{i-1}$$

where ε_i is gengamma distributed.

We proceed to fit a GACD(1,1) to our data using the acdFit command as shown below:

```
fitModel <- acdFit(durations =adjDurData, model = "ACD", dist = "gengamma",
order =c(1,1), dailyRestart = 1)
```

Specifying the parameters dist = ‘gengamma’ and order = c(1,1) ensures that it is a GACD(1,1) model. Results of the run are as below:



```
Model:
  ACD(1, 1)

Distribution:
  gengamma

N: 46120

Parameter estimate:
          Coef       SE PV
omega  0.00867 0.000833  0
alpha1 0.06382 0.002527  0
beta1  0.92900 0.002818  0
kappa  4.93749 0.296147  0
gamma  0.38027 0.012095  0

Note: For the distribution parameters the null hypothesis is such that the parameter = 1 (2-sided).

The fixed/unfree mean distribution parameter:
 lambda: 0.0102429

Goodness of fit:
                     value
LogLikelihood -41241.567535
AIC            82493.135070
BIC            82536.830080
MSE                1.627812

Convergence: 0
```

Following this, we repeat the same process as above. The code for this can be found in the corresponding Notebook.

Model LACD

In the ACD model sufficient conditions are required for the parameters to ensure the positivity of durations. If one wants to add linearly in the autoregressive equation some variables taken from the microstructure literature and having expected negative coefficients, the durations might become negative. To avoid this situation log-ACD (logarithmic ACD) was introduced.

As in the ACD model, let x_i be the duration between two quotes. The logarithmic version of the ACD model is defined by:

$x_i = e^{\psi_i}\varepsilon_i$

Here, $\{\varepsilon_i\}$ is a sequence of non-negative independent and identically distributed random variables with $E[\varepsilon_i] = 1$. Generally, ε_i's are assumed to have standardized Weibull distribution. Also, the autoregressive equation is specified on the logarithm of the conditional duration ψ_i. There are 2 variants of log-ACD:

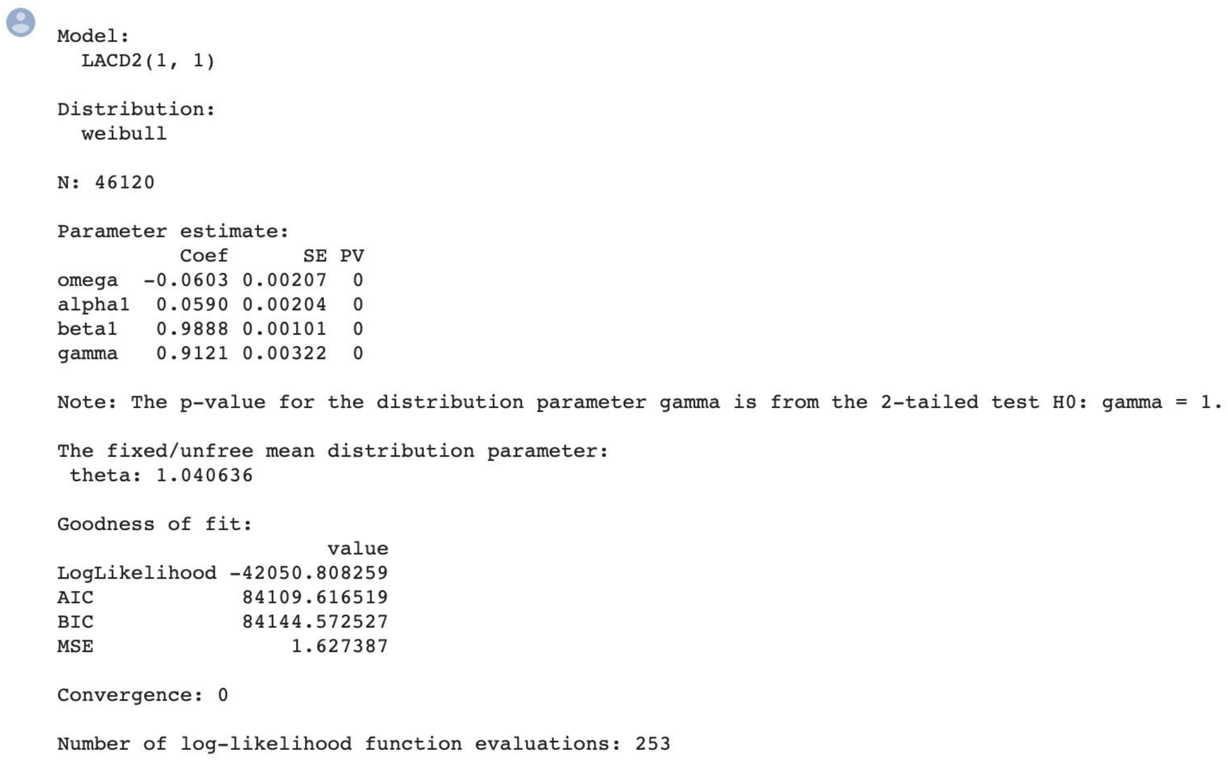
Log-ACD1(r, s)

$$ln(\psi_i) = \omega + \sum_{j=1}^{r} \gamma_j ln(x_{i-j}) + \sum_{j=1}^{s} \omega_j ln(\psi_{i-j})$$

Log-ACD2(r, s)

$$ln(\psi_i) = \omega + \sum_{j=1}^{r} \gamma_j (x_{i-j}/\psi_{i-j}) + \sum_{j=1}^{s} \omega_j ln(\psi_{i-j})$$

In practice, the Log-ACD2 model is often preferred as it seems to fit the data better.



```
Model:
  LACD2(1, 1)

Distribution:
  weibull

N: 46120

Parameter estimate:
          Coef      SE PV
omega  -0.0603 0.00207  0
alpha1  0.0590 0.00204  0
beta1   0.9888 0.00101  0
gamma   0.9121 0.00322  0

Note: The p-value for the distribution parameter gamma is from the 2-tailed test H0: gamma = 1.

The fixed/unfree mean distribution parameter:
 theta: 1.040636

Goodness of fit:
                     value
LogLikelihood -42050.808259
AIC            84109.616519
BIC            84144.572527
MSE                1.627387

Convergence: 0

Number of log-likelihood function evaluations: 253
```

Following this, we repeat the same process as above. The code for this can be found in the corresponding Notebook.

References

[1] Analysis of Financial Time Series, Ruey S. Tsay
[2] Documentation for R packages
[3] Meitz, M. and Terasvirta, T. (2006). Evaluating models of autoregressive conditional duration. Journal of Business and Economic Statistics 24: 104-124.