ASHR Data Analysis

Contents

Price **2**

Plots 2

Time Series Analysis 5

Realised Volatility **9**

Computations 9

Plots 10

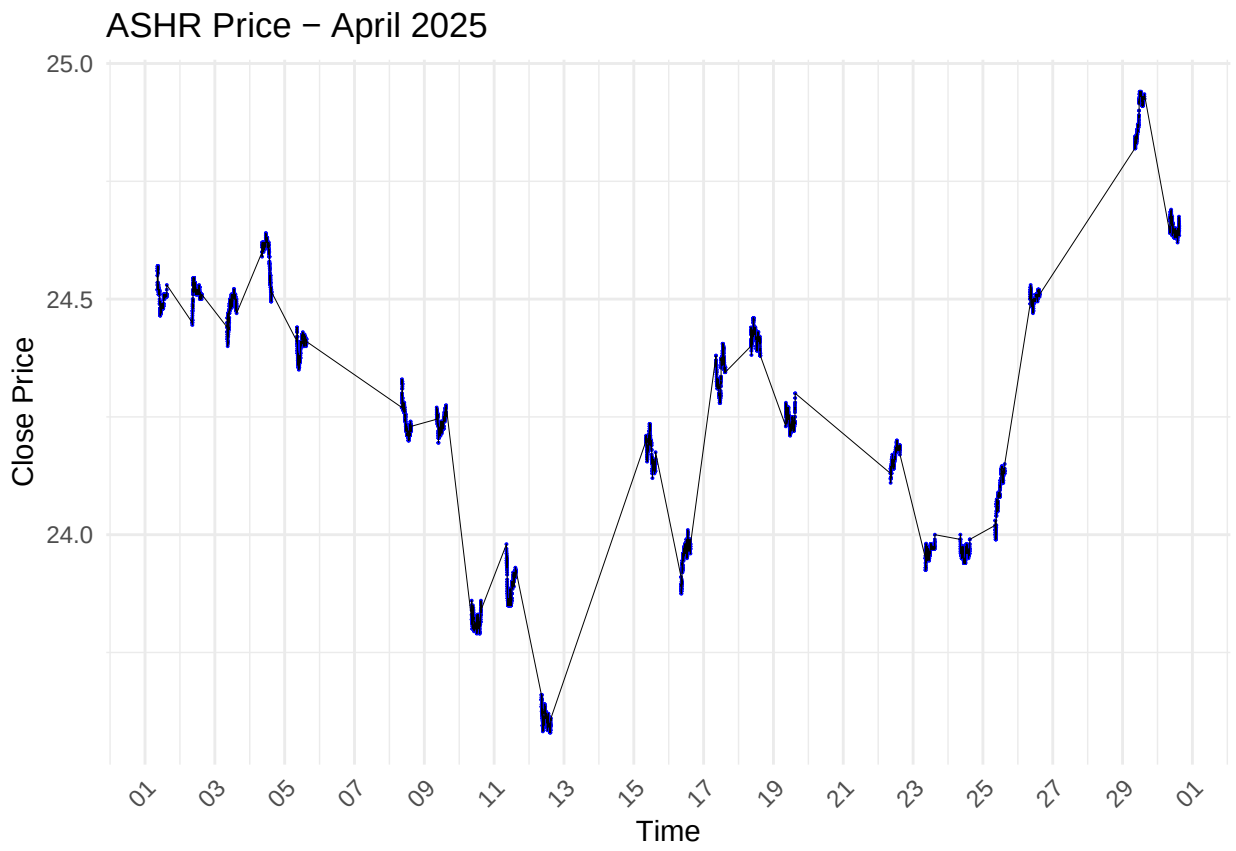
Price

Plots

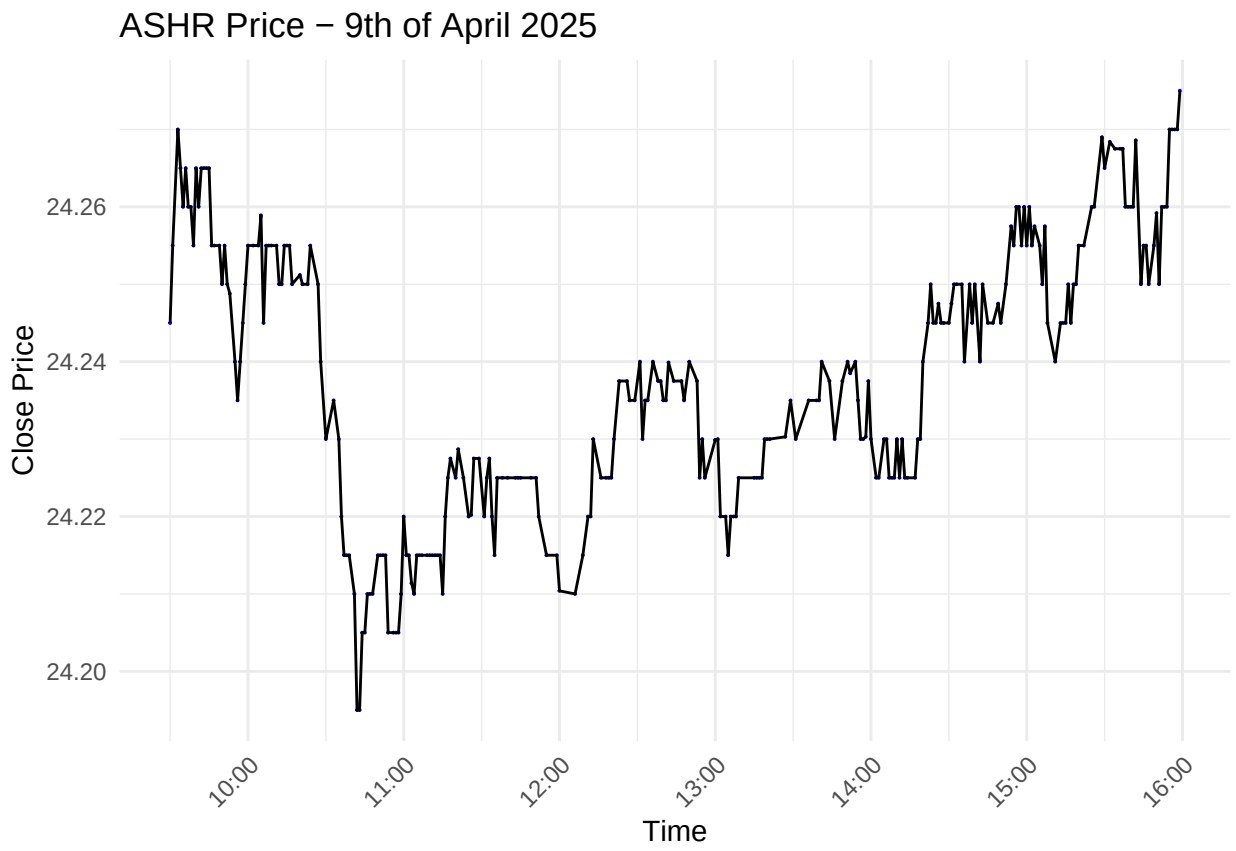
```
#All Time
price_plotter(raw_ASHR,"ASHR Price Over Time")
```


```
#2024
ASHR_2024 = year_selector(raw_ASHR,2024)
price_plotter_year(ASHR_2024,"ASHR Price - 2024")
```


```
#April 2025
ASHR_2025_04 = month_selector(raw_ASHR,2024,04)
price_plotter_month(ASHR_2025_04,"ASHR Price - April 2025")
```

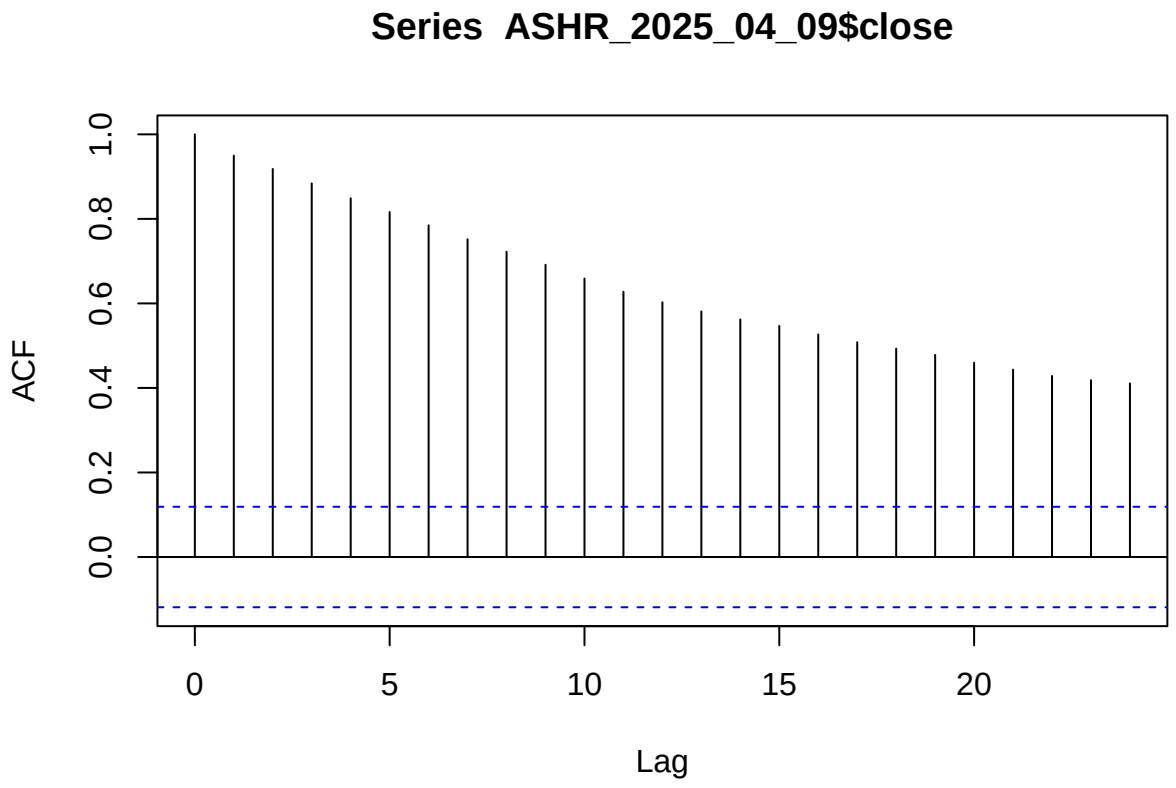



```
#9th of April 2025
ASHR_2025_04_09 = day_selector(raw_ASHR,2024,04,09)
price_plotter_day(ASHR_2025_04_09,"ASHR Price - 9th of April 2025")
```

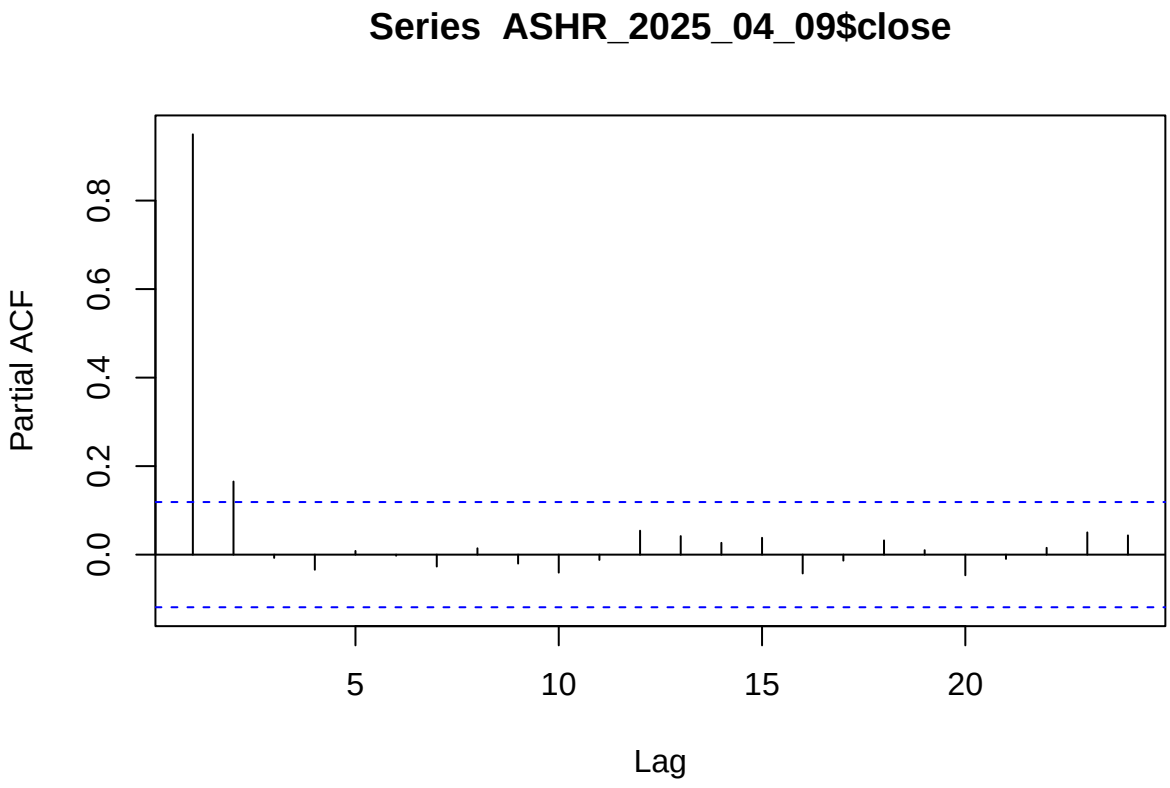


Time Series Analysis

```
acf(ASHR_2025_04_09$close)
```




```
pacf(ASHR_2025_04_09$close)
```




```
AR1 = arima(ASHR_2025_04_09$close,c(1,0,0),method="ML")
```

```
## Warning in arima(ASHR_2025_04_09$close, c(1, 0, 0), method = "ML"): possible
## convergence problem: optim gave code = 1
```

```
AR2 = arima(ASHR_2025_04_09$close,c(2,0,0),method="ML")
AR3 = arima(ASHR_2025_04_09$close,c(3,0,0),method="CSS")
table1 = export_summs(AR1,AR2,AR3, model.names = c("AR1","AR2","AR3"), digits = 4)
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
## Warning in sqrt(diag(vcov(object))): NaNs produced
```

```
## Warning in FUN(X[[i]], ...): tidy() does not return p values for models of
## class data.frame; significance stars not printed.
## Warning in FUN(X[[i]], ...): tidy() does not return p values for models of
## class data.frame; significance stars not printed.
## Warning in FUN(X[[i]], ...): tidy() does not return p values for models of
## class data.frame; significance stars not printed.
```

```
huxtable::caption(table1) <- "AR Estimations"
huxtable::set_width(table1, 0.8)
```

Table 1: AR Estimations

	AR1	AR2	AR3
ar1	0.9996	0.7743	0.7493
	(NaN)	(0.0598)	(0.0593)
intercept	24.2383	24.2427	24.2388
	(0.1550)	(0.0091)	(0.0106)
ar2		0.1962	0.2006
		(0.0602)	(0.0740)
ar3			0.0227
			(0.0593)
nobs	272	272	272
sigma	0.0051	0.0049	0.0048
logLik	1048.5063	1057.8756	
AIC	-2091.0125	-2107.7512	
BIC	-2080.1951	-2093.3280	
nobs.1	272.0000	272.0000	272.0000

*** p < 0.001; ** p < 0.01; * p < 0.05.

```
AR1res = as.numeric(AR1$residuals)
AR1res_lagged <- lag(AR1res, 1)
iidcheck1 = lm(AR1res ~ AR1res_lagged)
AR2res = as.numeric(AR2$residuals)
AR2res_lagged <- lag(AR2res, 1)
iidcheck2 = lm(AR2res ~ AR2res_lagged)
AR3res = as.numeric(AR3$residuals)
AR3res_lagged <- lag(AR3res, 1)
iidcheck3 = lm(AR3res ~ AR3res_lagged)
table2 = export_summs(iidcheck1,iidcheck2,iidcheck3,
             model.names = c("AR1 Residuals","AR2 Residuals","AR3 Residuals"),
             digits = 4)
huxtable::caption(table2) <- "Checking Residuals"
huxtable::set_width(table2, 0.8)
```

Table 2: Checking Residuals

	AR1 Residuals	AR2 Residuals	AR3 Residuals
(Intercept)	0.0001	-0.0000	-0.0000
	(0.0003)	(0.0003)	(0.0003)
AR1res_lagged	-0.2090 ***		
	(0.0597)		
AR2res_lagged		0.0025	
		(0.0611)	
AR3res_lagged			0.0021
			(0.0611)
N	271	271	271
R2	0.0435	0.0000	0.0000

*** p < 0.001; ** p < 0.01; * p < 0.05.

Realised Volatility

Computations

```
#avg per day for each month of any dataset
vol_ASHR_daily = r.vol_daily(raw_ASHR,merge=F)
head(vol_ASHR_daily)
```

timestamp	r__vol__d
2023-01-03	1.25e-07
2023-01-04	9.35e-08
2023-01-05	6.14e-08
2023-01-06	9.84e-08
2023-01-09	7.4e-08
2023-01-10	7.14e-08

```
#can then filter out years, months, or days
vol_24d = year_selector(vol_ASHR_daily,2024)
vol_24_08d = month_selector(vol_ASHR_daily,2024,08)
vol_24_11_04d = day_selector(vol_ASHR_daily,2024,11,04) #scalar
```

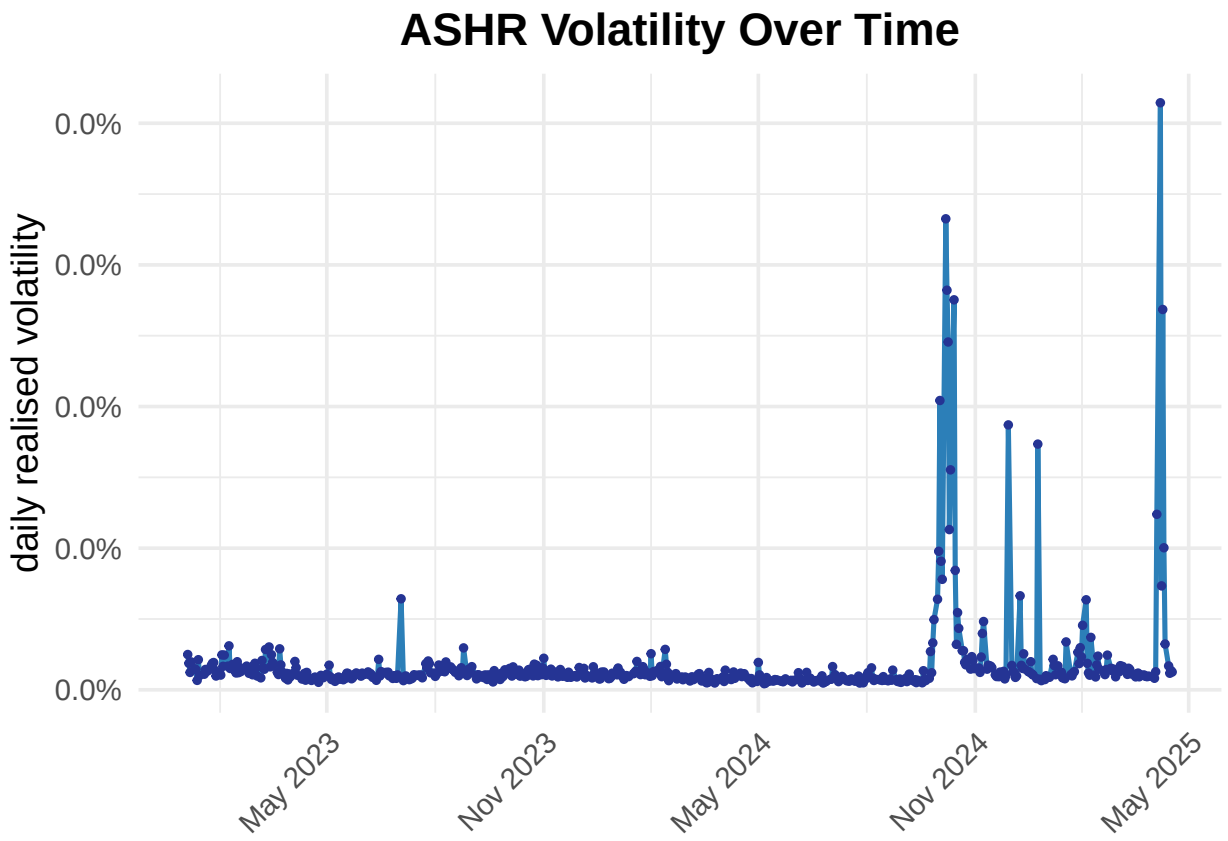
```
#avg per hour for each day of each month of any dataset
vol_ASHR_hourly = r.vol_hourly(raw_ASHR,merge=F)
head(vol_ASHR_hourly)
```

timestamp	r__vol__h
2023-01-03 09:00:00	4.86e-07
2023-01-03 10:00:00	1.26e-07
2023-01-03 11:00:00	1.01e-07
2023-01-03 12:00:00	9.23e-08
2023-01-03 13:00:00	5.3e-08
2023-01-03 14:00:00	1.02e-07

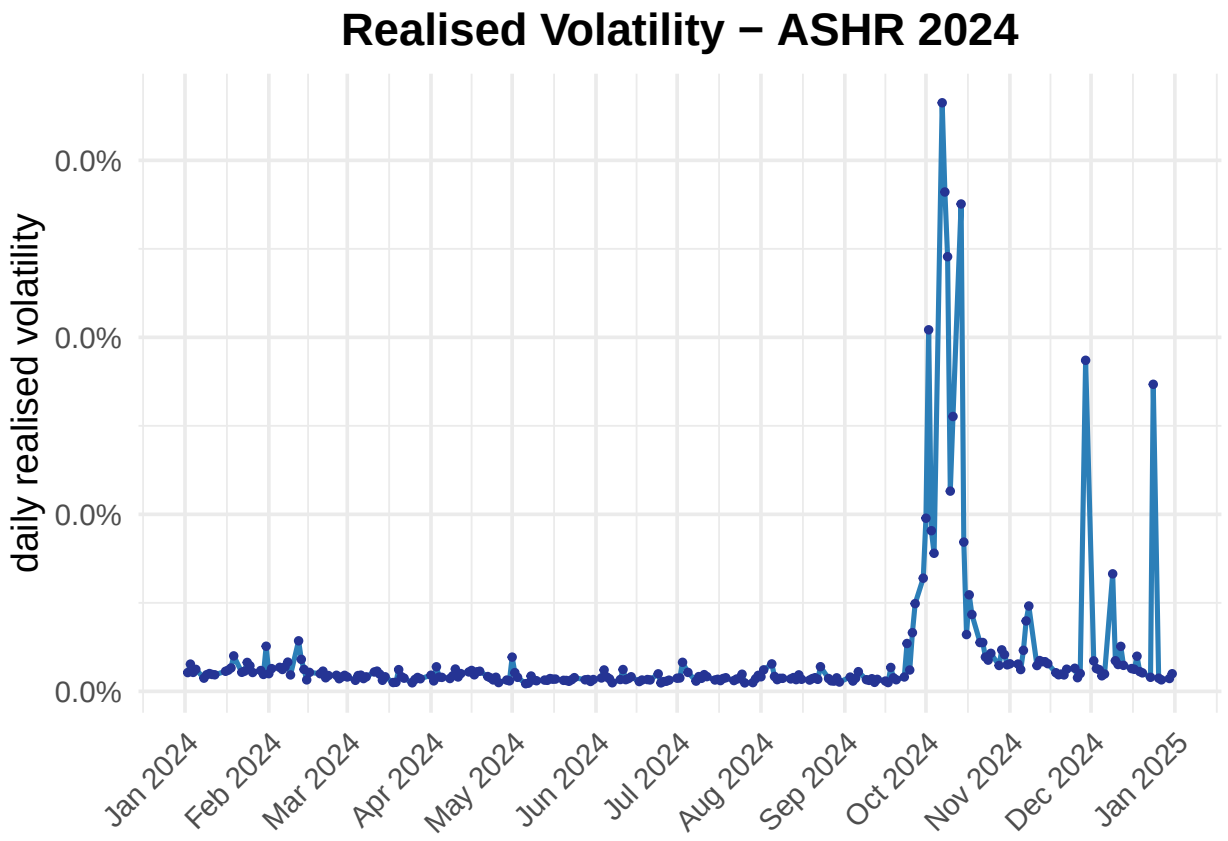
```
#can then filter out years, months, or days
vol_24h = year_selector(vol_ASHR_hourly,2024)
vol_24_08h = month_selector(vol_ASHR_hourly,2024,08)
vol_24_11_04h = day_selector(vol_ASHR_hourly,2024,11,04) #vector
```

Plots

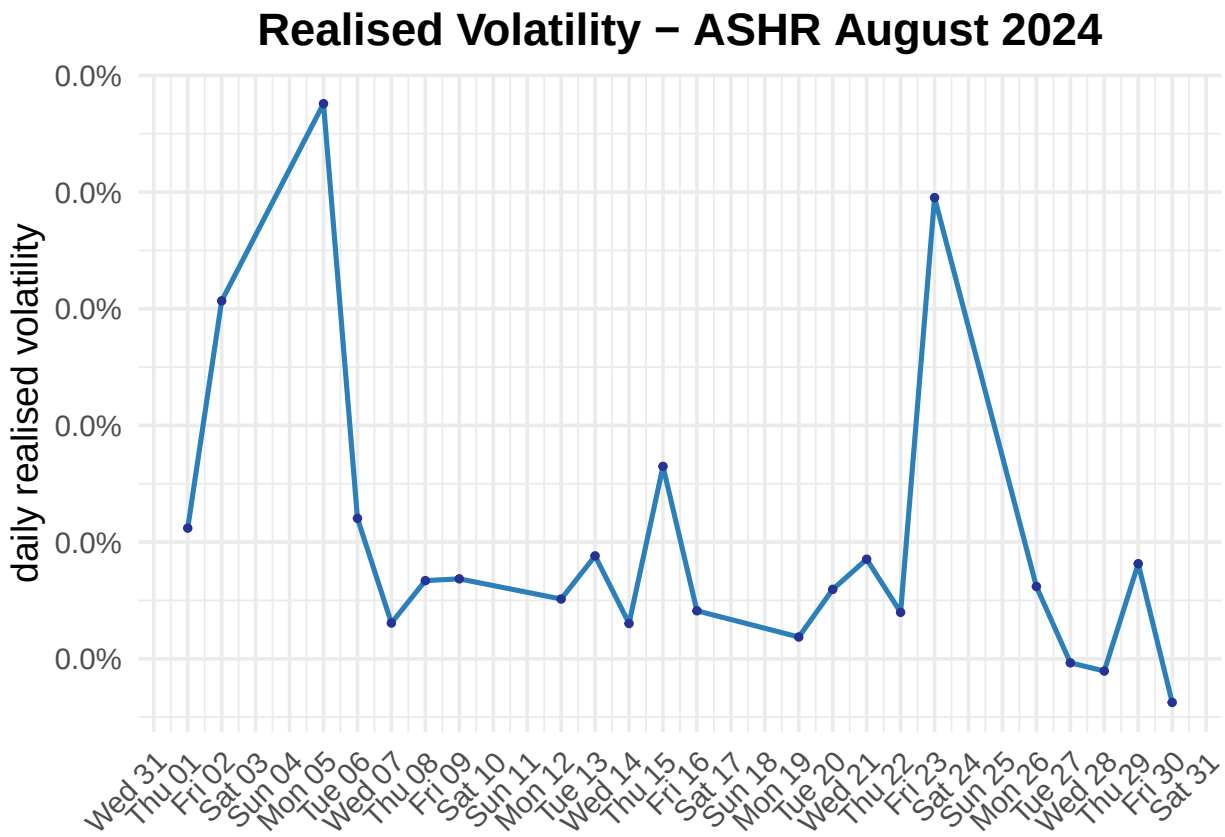
```
#avg per day volatility all time
dvol_plotter(vol_ASHR_daily,breaks="yearly",
             title="ASHR Volatility Over Time")
```




```
#avg per day volatility in a year
dvol_plotter(vol_24d,breaks="monthly",
             title="Realised Volatility - ASHR 2024")
```

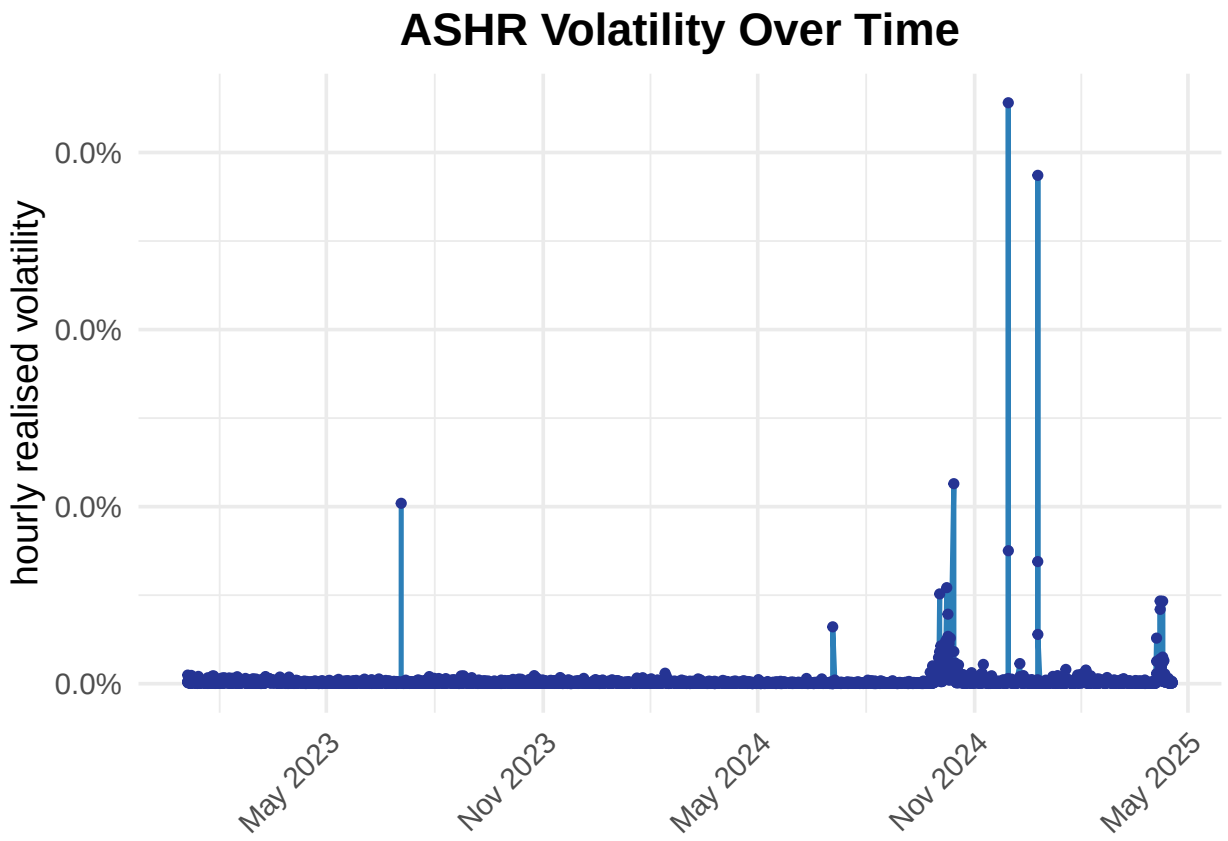



```
#avg per day volatility in a month
dvol_plotter(vol_24_08d,breaks="daily",
             title="Realised Volatility - ASHR August 2024")
```

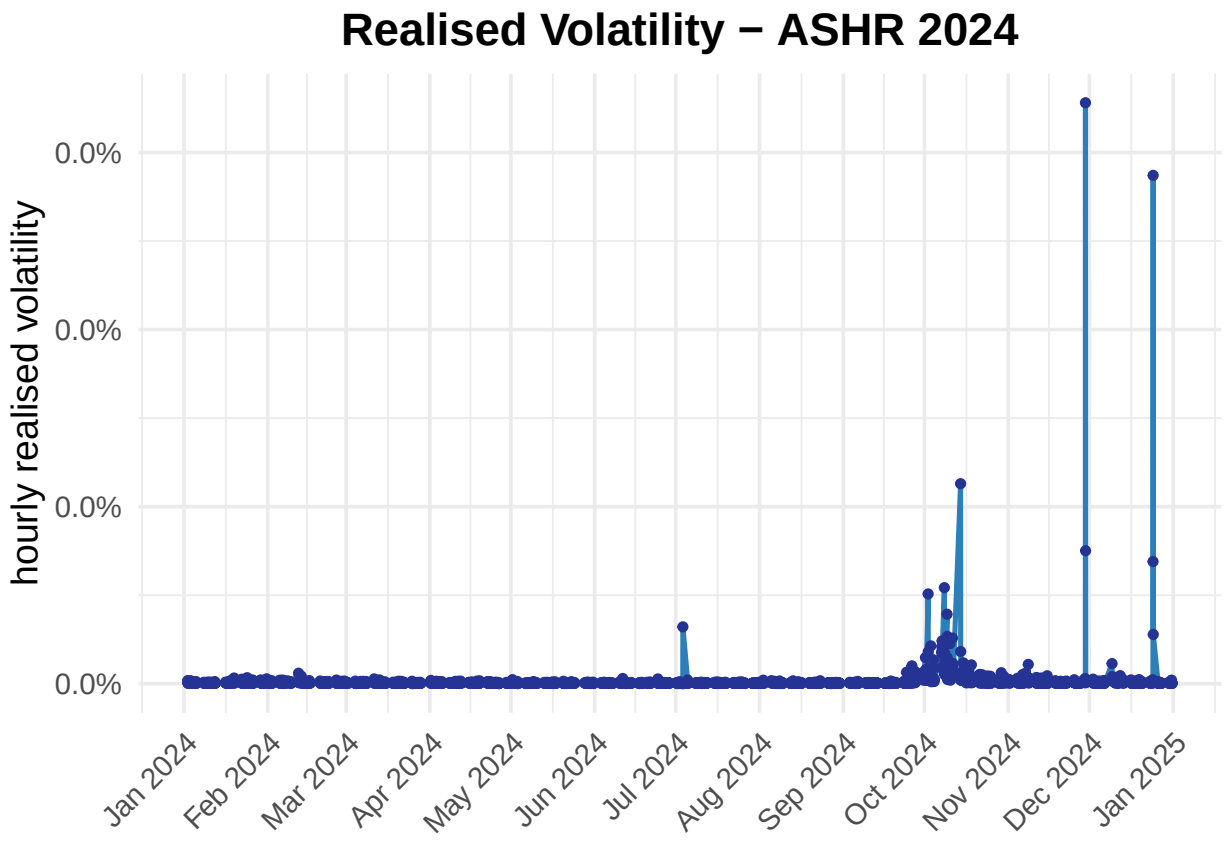



```
#hourly volatility all time
hvol_plotter(vol_ASHR_hourly,breaks="yearly",
             title="ASHR Volatility Over Time")
```

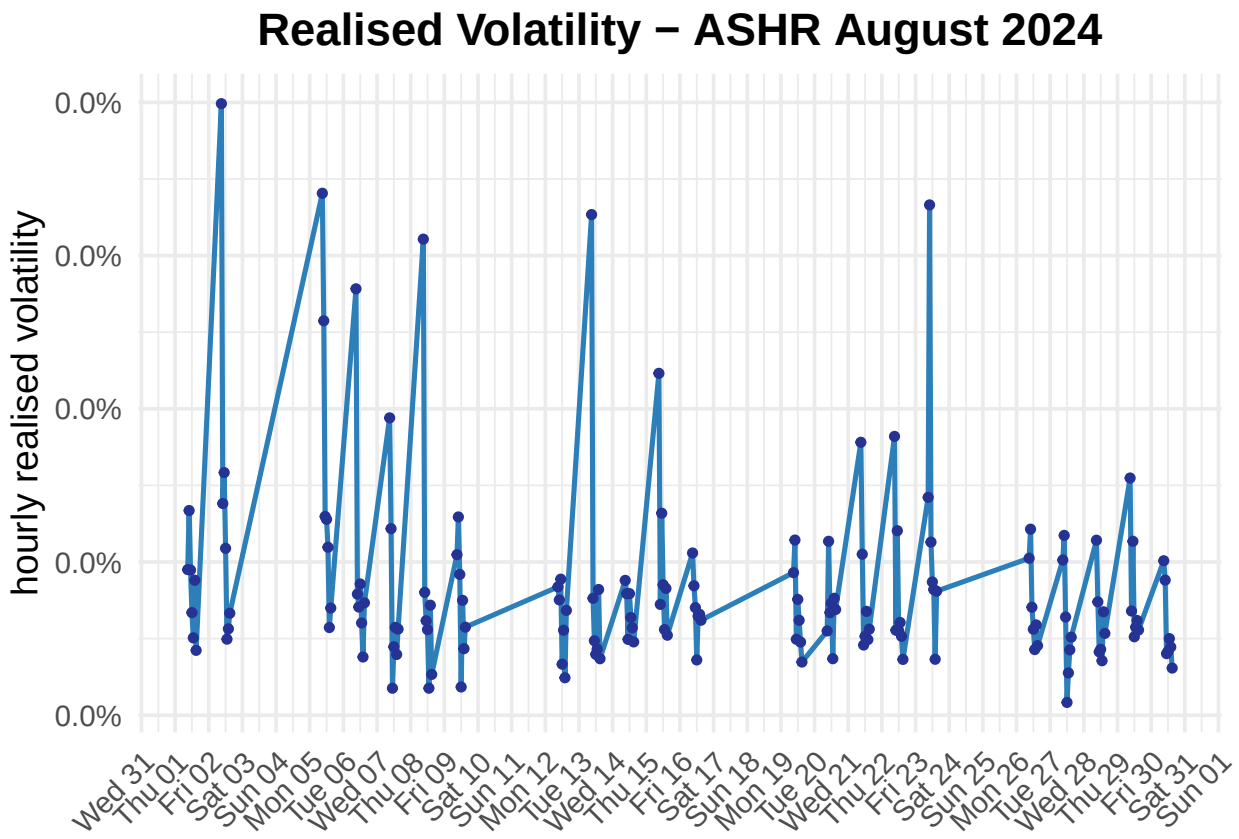
```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point()`).
```



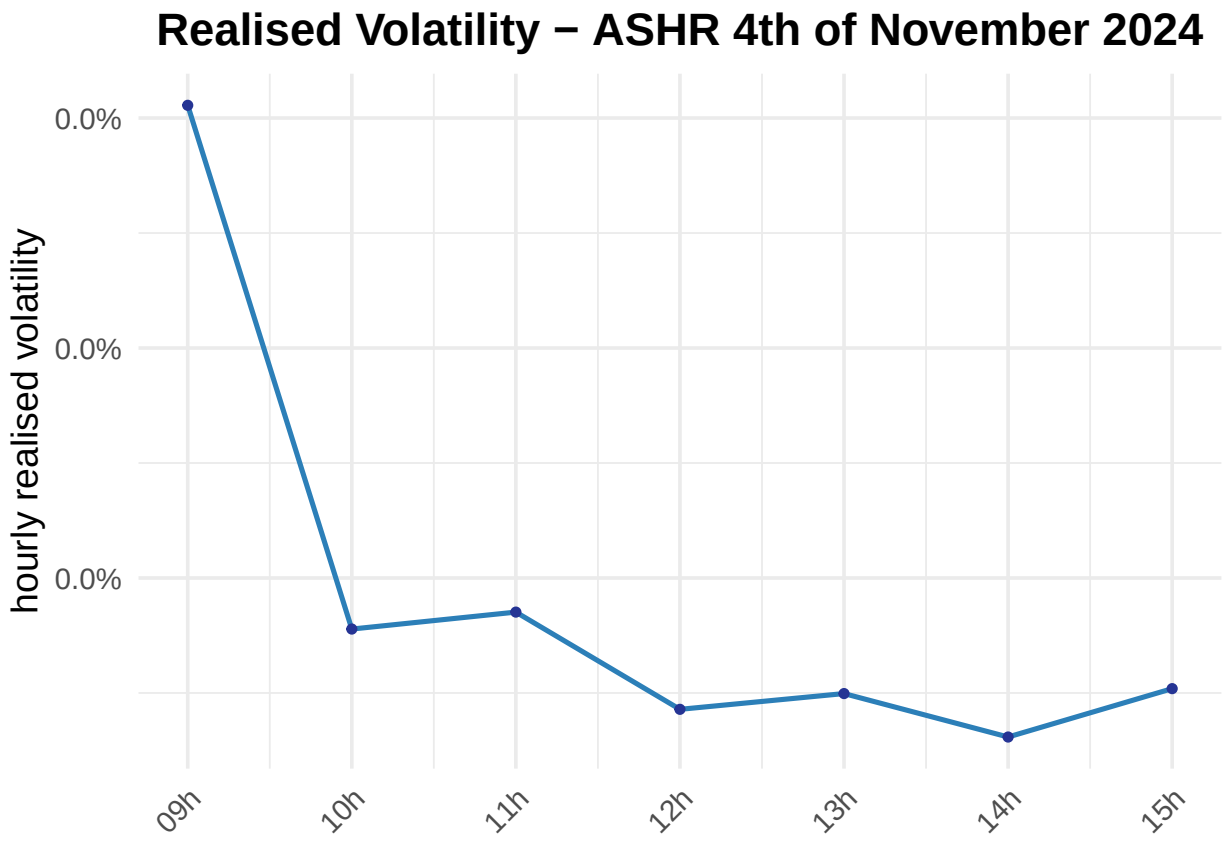

```
#hourly volatility in a year
hvol_plotter(vol_24h,breaks="monthly",
             title="Realised Volatility - ASHR 2024")
```




```
#hourly volatility in a month
hvol_plotter(vol_24_08h,breaks="daily",
             title="Realised Volatility - ASHR August 2024")
```




```
#hourly volatility in a day
hvol_plotter(vol_24_11_04h,breaks="hourly",
             title="Realised Volatility - ASHR 4th of November 2024")
```


Realised Volatility − ASHR 4th of November 2024
hourly realised volatility
0.0%
0.0%
0.0%
09h
10h
11h
12h
13h
14h
15h