PS8-Lanza

Ariel Lanza

6/7/2021

As we did last time, we import the data and we define the function for the Realized Volatility

```
db <- read_csv("SPY_HF.csv", col_names = c("Date", "Time", "Price"))
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Date = col_double(),
##   Time = col_double(),
##   Price = col_double()
## )
```

```
#data %>% ggplot(aes(x=Date, y=Price)) + geom_line()

db <- db %>%
  mutate(
    Time = Time %>%
      as.character() %>%
      str_pad(4, pad="0"),
    Date = Date %>%
      as.character(),
    Date_full = paste(Date, Time) %>% ymd_hm
  )

RV <- function(vec, step){
  mask <- rep(c(T, rep(F, step-1)), length.out=length(vec))
  mask[length(mask)] <- T
  subvec <- vec[mask]
  #lnret <- subvec[2:length(subvec)]/subvec[1:length(subvec)-1]-1
  #lnret <- log(subvec[2:length(subvec)]/subvec[1:length(subvec)-1])
  #lnret <- log(subvec[2:length(subvec)])-log(subvec[1:length(subvec)-1])
  lnret <- subvec[2:length(subvec)] - subvec[1:(length(subvec)-1)]
  return(sum(lnret^2))
}
```

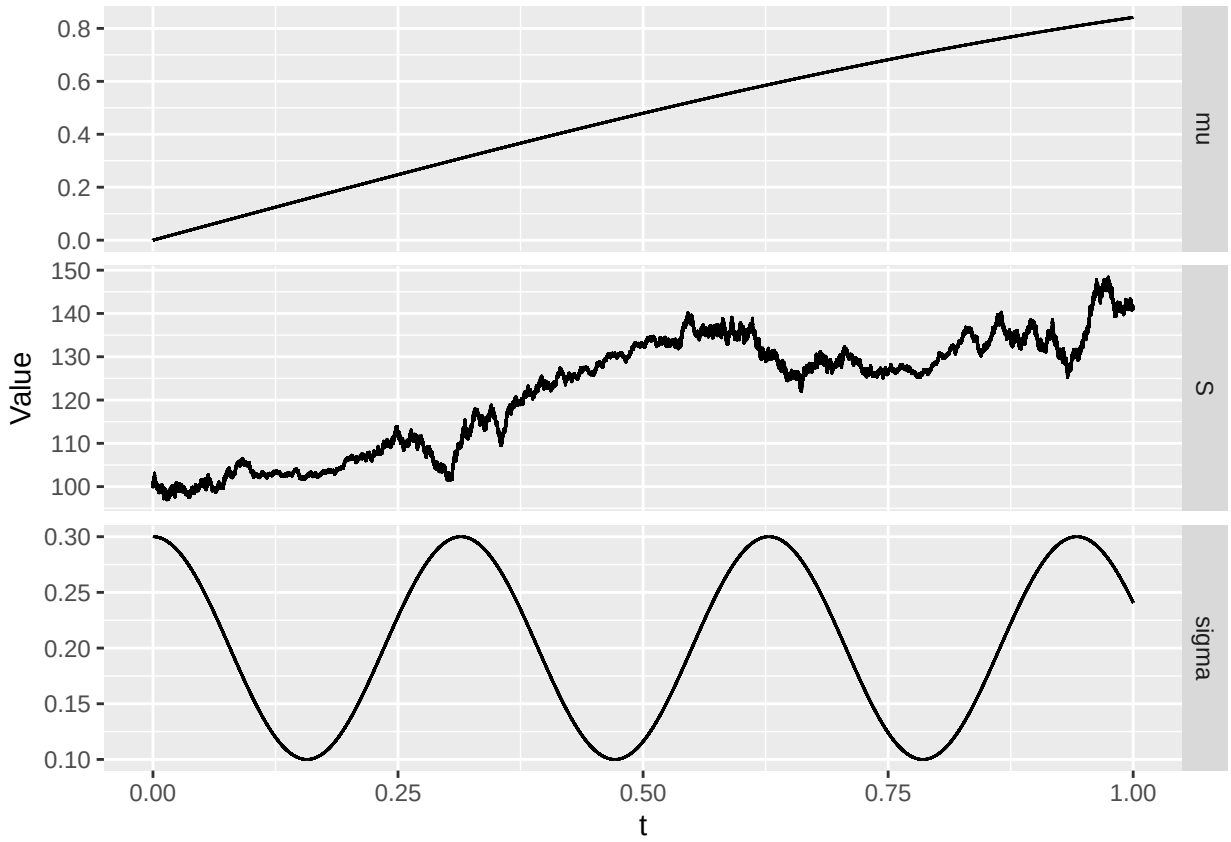
0) An experiment with simulated data

Let's start by generating one full year of data, meaning 365 days with $24 \times 60 = 1440$ minutes for each day. We will set

$$\mathrm{d}S_t = S_t \sin(t)\mathrm{d}t + S_t \frac{2 + \cos(20t)}{10}\mathrm{d}W_t$$

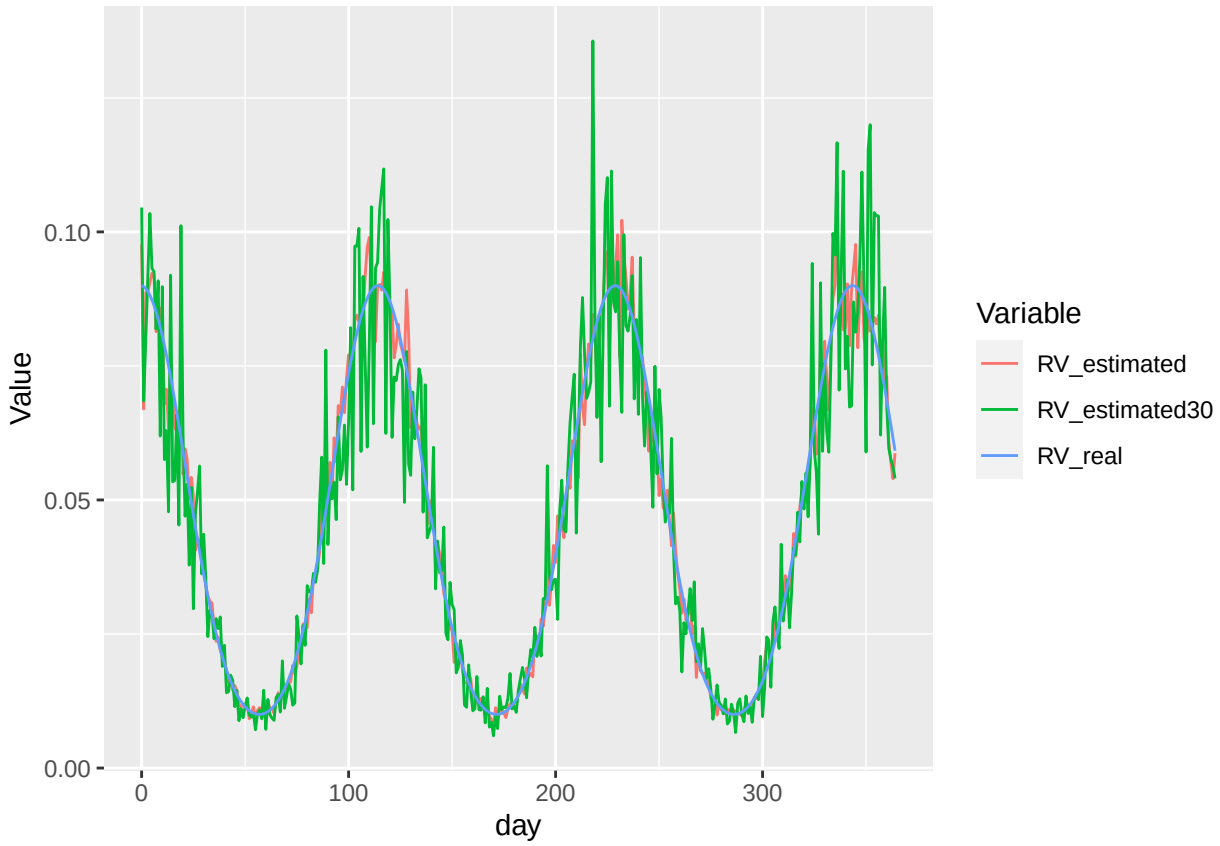
```
# Test Data
```

```r
T_years = 1
T_granularity = 365*24*60
#T_granularity = 10
T_len = T_years*T_granularity
delta_t = 1/(T_granularity)

S_vec = rep(100, T_len+1)
t_vec = (0:T_len)*delta_t
mu_t = sin(t_vec)
sigma_t = (2+cos(20*t_vec))/10

set.seed(12345)
delta_W = rnorm(T_len+1, 0, sqrt(delta_t))

for(s_it in 2:(T_len+1)){
  S_vec[s_it] = S_vec[s_it-1]*(
      1 +
      mu_t[s_it] * delta_t +
      sigma_t[s_it] * delta_W[s_it]
    )
}

df_sym <- data.frame(t = t_vec, mu = mu_t, sigma = sigma_t, S = S_vec) %>%
  mutate(
    day = as.integer(t_vec %/% (delta_t * 24*60)),
    time = as.integer((t_vec %% (delta_t * 24*60))/delta_t),
    full_time = ymd(20200101) + period(days=day) + period(minutes = time),
    full_time_indb = full_time %in% db$Date_full
  )

df_sym %>% pivot_longer(
    cols = c(mu, sigma, S),
    names_to = "Variable",
    values_to = "Value"
  ) %>%
  ggplot(aes(x=t, y=Value)) +
  geom_line() +
  facet_grid(Variable ~.,scales = "free_y")
```



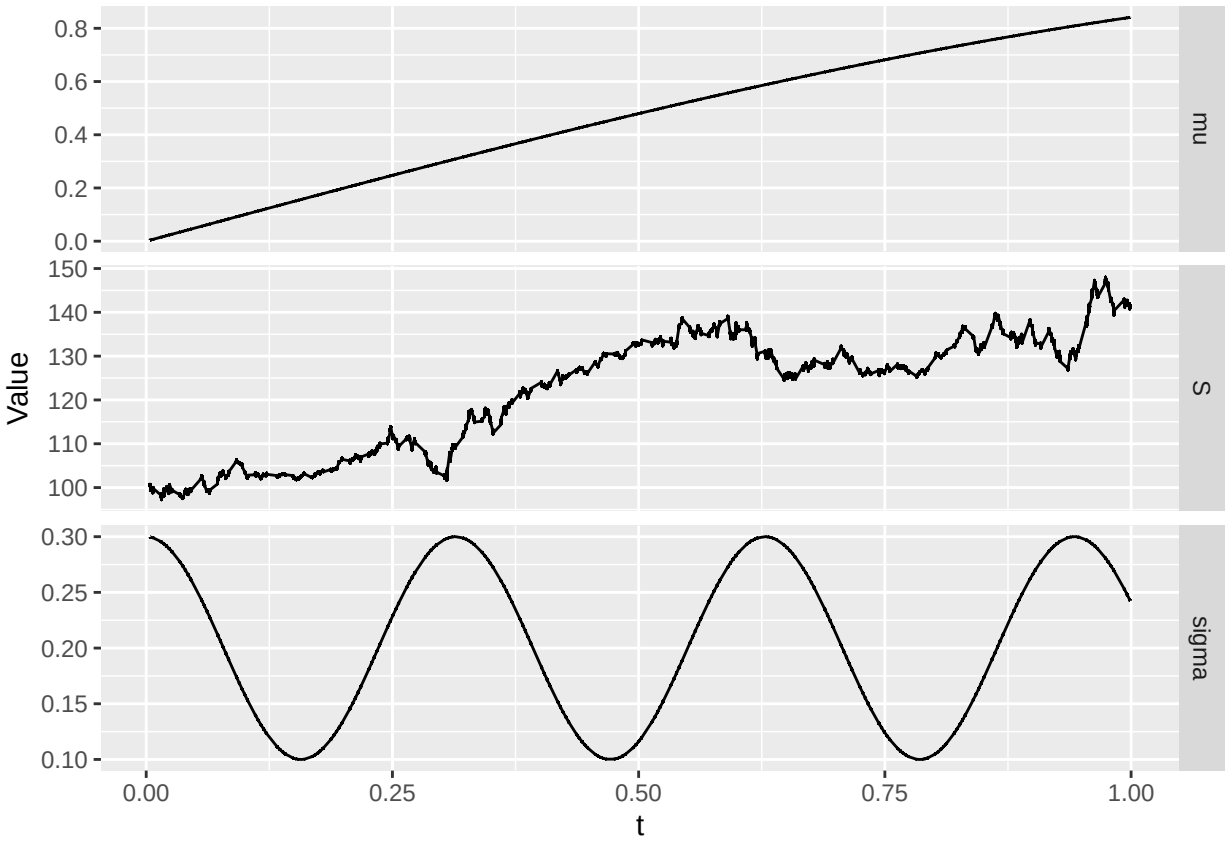
We show that the real volatility (of the log price) computed at 1, 5 and 30 minutes are good approximations of the quadratic variation, i.e. $\frac{\sigma(t)^2}{\Delta t}$.

```
estimate_sym <- df_sym %>%
  group_by(day) %>%
  summarise(
    RV_estimated = RV(log(S), 5)*365,
    RV_estimated30 = RV(log(S), 30)*365,
    RV_real = mean(sigma)^2,
    S = mean(S),
    S_real = mean(S)*mean(sigma)/100
  )

estimate_sym %>% pivot_longer(
    cols = c(RV_estimated, RV_estimated30, RV_real),
    names_to = "Variable",
    values_to = "Value"
  ) %>%
  ggplot(aes(x=day, y=Value, color=Variable)) +
  geom_line()
```



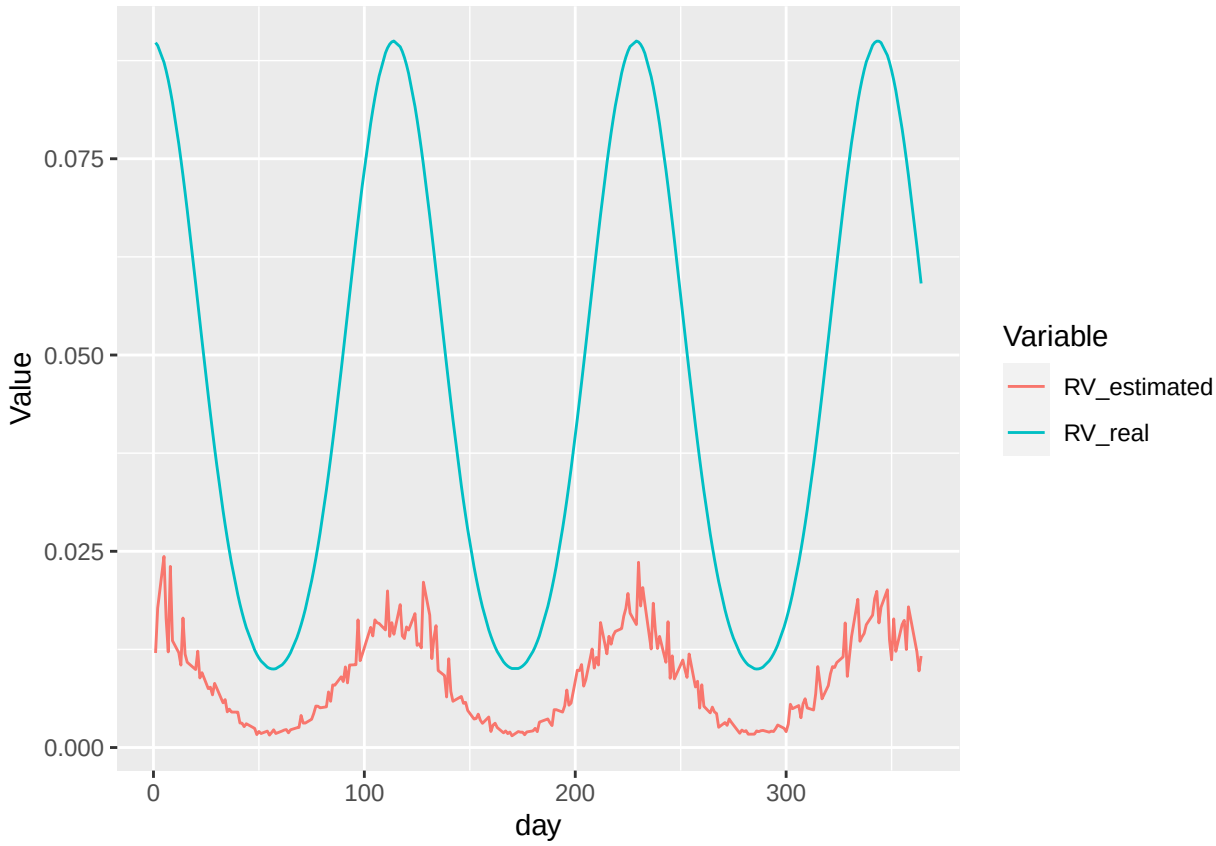
Now lets's filter only the data observed during the trading day and see what happens

```
df_sym_obs <- df_sym %>%
  filter(
    full_time_indb
  )
df_sym_obs %>% pivot_longer(
    cols = c(mu, sigma, S),
    names_to = "Variable",
    values_to = "Value"
  ) %>%
  ggplot(aes(x=t, y=Value)) +
  geom_line() +
  facet_grid(Variable ~.,scales = "free_y")
```



Now we scale with respect to the convention that there are 252 trading days per year. We notice that not including the overnight data we underestimate the realized volatility (and this happens also if we consider 365 trading days).

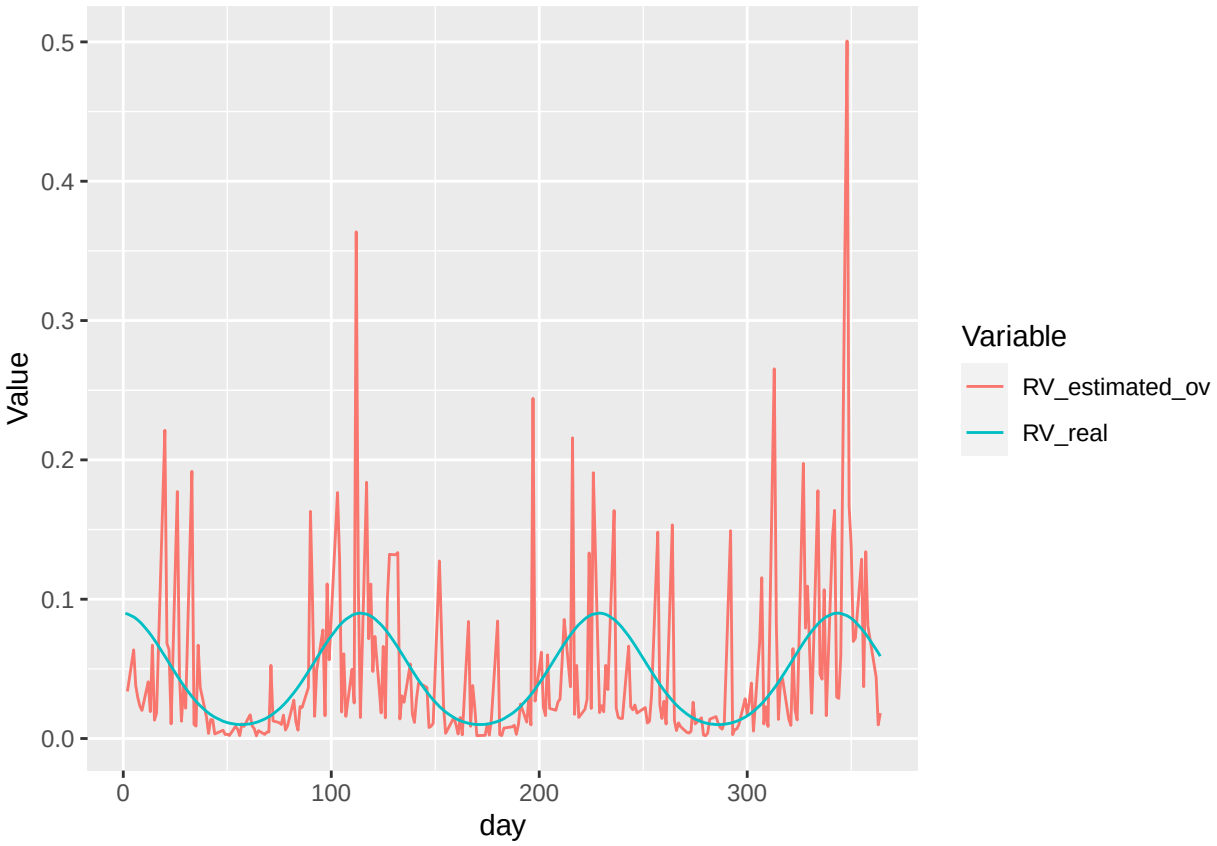
```
estimate_sym_obs <- df_sym_obs %>%
  group_by(day) %>%
  summarise(
    RV_estimated = RV(log(S), 5)*252,
    RV_real = mean(sigma)^2,
    S = mean(S),
    S_real = mean(S)*mean(sigma)/100,
    S_open = first(S),
    S_close = last(S),
  ) %>%
  mutate(
    S_close_previous = lag(S_close),
    RV_overnight = (log(S_open) - log(S_close_previous))^2*252,
    RV_estimated_ov = RV_estimated + RV_overnight
  )

estimate_sym_obs %>% pivot_longer(
    cols = c(RV_estimated,  RV_real),
    names_to = "Variable",
    values_to = "Value"
  ) %>%
  ggplot(aes(x=day, y=Value, color=Variable)) +
  geom_line()
```



If we include the overnight estimate we get a much dirtier estimate

```
estimate_sym_obs %>% pivot_longer(
    cols = c(RV_estimated_ov,  RV_real),
    names_to = "Variable",
    values_to = "Value"
  ) %>%
  ggplot(aes(x=day, y=Value, color=Variable)) +
  geom_line()
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```



However, it seems that the averages are much more in line with what we expected

```
estimate_sym_obs %>% pivot_longer(
    cols = c(RV_real,RV_estimated, RV_estimated_ov),
    names_to = "Estimator_Type",
    values_to = "Value"
  ) %>%
  group_by(
    Estimator_Type
  ) %>%
  summarise(
    mean = mean(Value,na.rm=T),
    variance = var(Value, na.rm=T)
  )
```

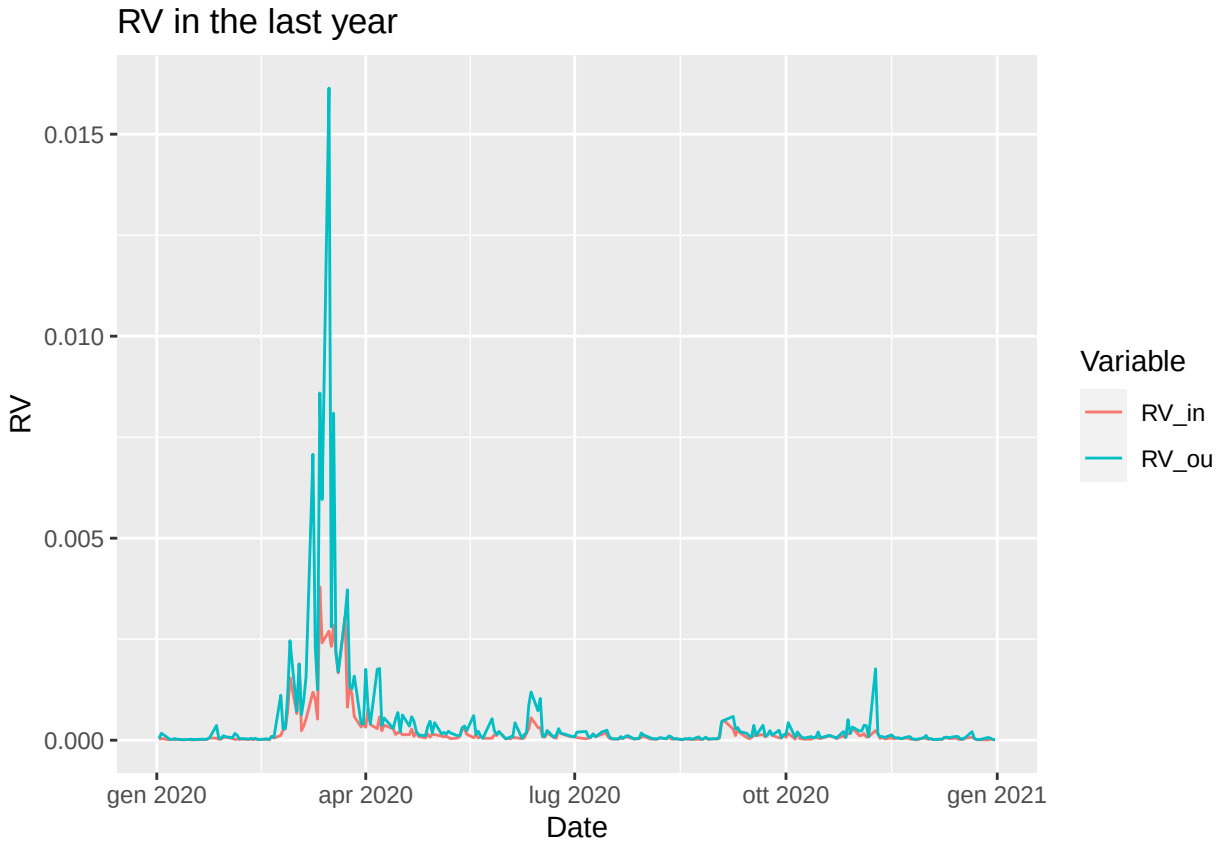
```
## # A tibble: 3 x 3
##   Estimator_Type     mean  variance
##   <chr>             <dbl>     <dbl>
## 1 RV_estimated    0.00871 0.0000324
## 2 RV_estimated_ov 0.0462  0.00392
## 3 RV_real         0.0467  0.000839
```

So we see that using the overnight data provides a better mean of the variance, however, the variance of the variance gets overestimated.

1 and 2

We plot the estimated volatilizes for the last year using both the intraday estimates and the estimate using the overnight movement, that gives a plot similar to the one obtained previously.

```r
RV_series <- db %>%
  group_by(Date) %>%
  summarise(
    RV_in = RV(log(Price), 5),
    Open = first(Price),
    Close = last(Price),
  ) %>%
  mutate(
    Date =ymd(Date),
    ClosePrev = lag(Close),
    Overnight = (log(Open)-log(ClosePrev))^2,
    RV_ou = RV_in + Overnight,
    RV_in_ann = RV_in*252,
    RV_ou_ann = RV_ou*252,
    RV_in_perc = sqrt(RV_in_ann)*100,
    RV_ou_perc = sqrt(RV_ou_ann)*100
  ) %>%
  drop_na()

RV_series %>%
  pivot_longer(
    cols = c(RV_in, RV_ou),
    names_to = "Variable",
    values_to = "Value",
  )%>% filter(Date>ym(202001)) %>% ggplot(aes(x=Date, y=Value, color=Variable)) +
  geom_line() +
  labs(
    x="Date",
    y="RV",
    title="RV in the last year"
  )
```



We now proceed to print the table

```
RV_series %>% pivot_longer(
    cols = c(RV_in,RV_ou),
    names_to = "Estimator_Type",
    values_to = "Value"
  ) %>%
  group_by(
    Estimator_Type
  ) %>%
  summarise(
    mean = mean(Value,na.rm=T),
    variance = var(Value, na.rm=T)
  )
```

```
## # A tibble: 2 x 3
##   Estimator_Type     mean     variance
##   <chr>             <dbl>        <dbl>
## 1 RV_in          0.000102 0.0000000479
## 2 RV_ou          0.000149 0.000000170
```

3)

As before, to obtain the annualized version we need to multiply the estimate by 252. While to obtain the percentage we divide by the value of the index (wlog by the previous day close) and we multiply by 100.

```
RV_series %>% pivot_longer(
    cols = c(RV_in,RV_ou, RV_in_ann, RV_ou_ann, RV_in_perc, RV_ou_perc),
    names_to = "Estimator_Type",
```

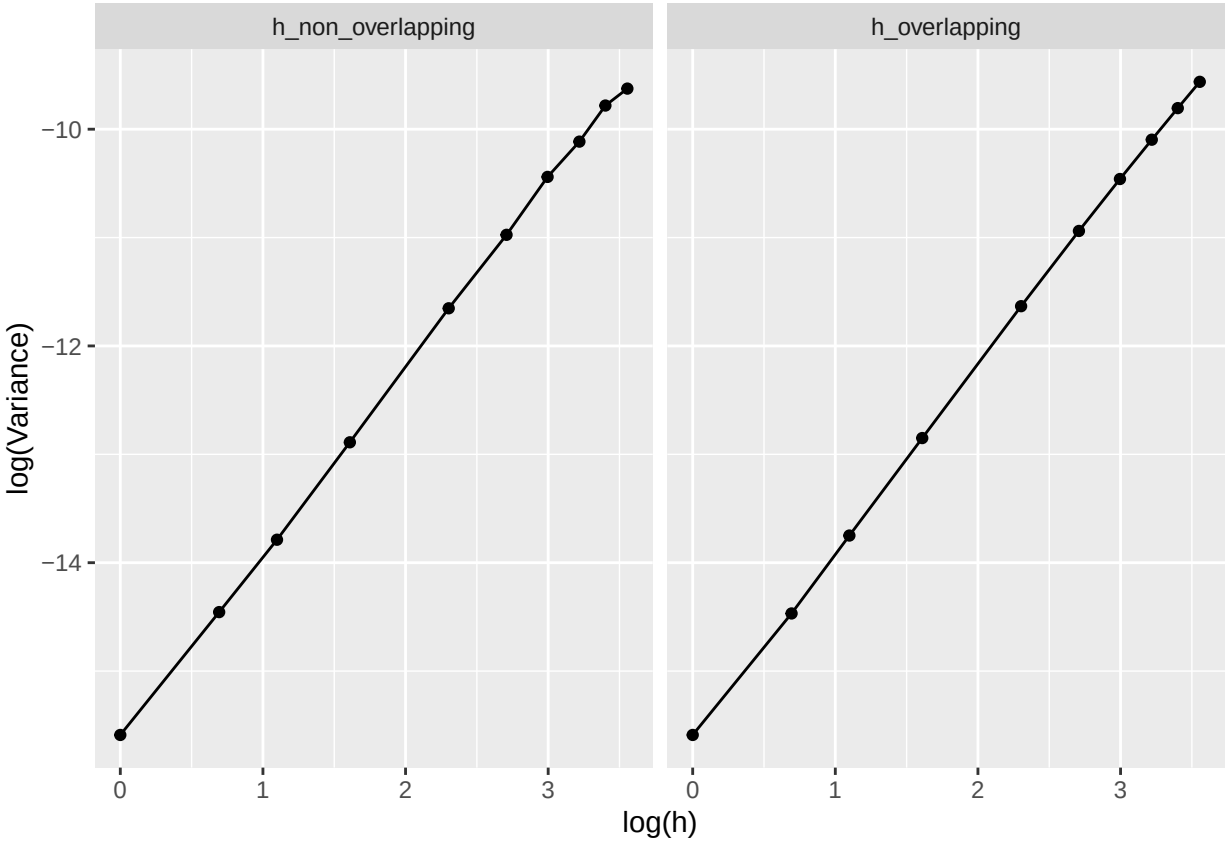
```
    values_to = "Value"
  ) %>%
  group_by(
    Estimator_Type
  ) %>%
  summarise(
    mean = mean(Value,na.rm=T),
    variance = var(Value, na.rm=T)
  )
```

```
## # A tibble: 6 x 3
##   Estimator_Type      mean variance
##   <chr>              <dbl>    <dbl>
## 1 RV_in           0.000102  4.79e-8
## 2 RV_in_ann       0.0258    3.04e-3
## 3 RV_in_perc     13.3       8.08e+1
## 4 RV_ou           0.000149  1.70e-7
## 5 RV_ou_ann       0.0374    1.08e-2
## 6 RV_ou_perc     15.6       1.30e+2
```

We notice that the value in percentage doesn't change much since on average the value of the index is 143.616, therefore multiplying by 100 and dividing by it we won't change order of magnitude.

4

```
h_vec <- c(1,2,3,5,10,15,20,25,30,35)
h_ov <- rep(0, length(h_vec))
h_nov <- rep(0, length(h_vec))
rsum.cumsum <- function(x, n = 3L){
  if(n==1){
    return(x)
  }
  return(tail(cumsum(x) - cumsum(c(rep(0, n), head(x, -n))), -n + 1))
}
for(it in 1:length(h_vec)){
  h_it <- h_vec[it]
  vec_it <- RV_series$RV_ou_ann/252
  vec_sum <-rsum.cumsum(vec_it, h_it)
  mask <- c(T, rep(F,h_it-1))
  vec_sum_nov <- vec_sum[mask]
  h_ov[it] <- var(vec_sum,na.rm = T)
  h_nov[it] <- var(vec_sum_nov, na.rm = T)
}
var_series <- data.frame(h = h_vec, h_overlapping = h_ov, h_non_overlapping = h_nov)

var_series %>% pivot_longer(
  cols = c(h_overlapping, h_non_overlapping),
  names_to = "Type",
  values_to = "Variance"
  ) %>%
  ggplot(aes(x=log(h), y=log(Variance)))+
  geom_line() +
  geom_point()+
  facet_grid(. ~ Type)
```



They look like straight lines. The slopes are given by

```
lm(log(h_overlapping) ~ log(h), data=var_series)
```

```
##
## Call:
## lm(formula = log(h_overlapping) ~ log(h), data = var_series)
##
## Coefficients:
## (Intercept)       log(h)
##     -15.612        1.713
```

and

```
lm(log(h_non_overlapping) ~ log(h), data=var_series)
```

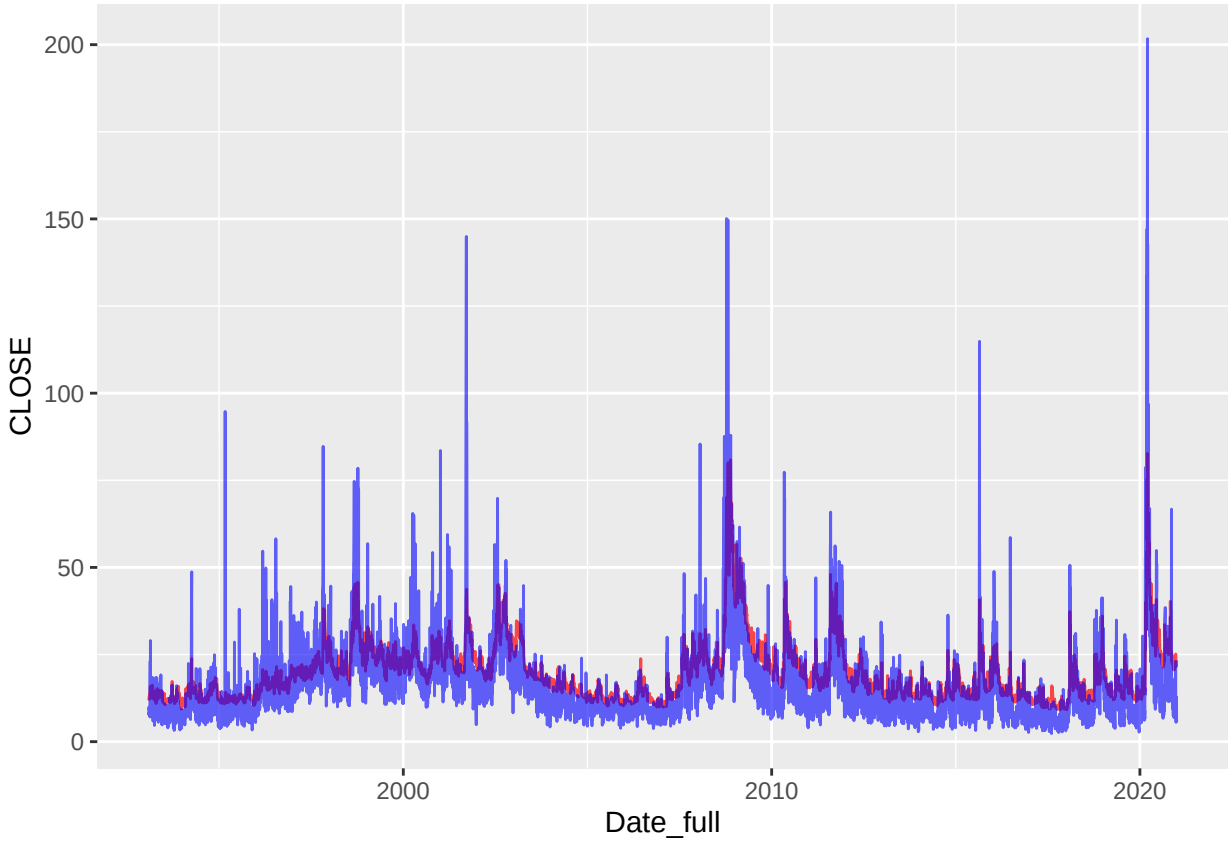
```
##
## Call:
## lm(formula = log(h_non_overlapping) ~ log(h), data = var_series)
##
## Coefficients:
## (Intercept)       log(h)
##      -15.62         1.71
```

d is given by the slope s as $d = \frac{s-1}{2}$. Therefore d should be approximately 0.355.

5

We now display both series. Vix in red and realized volatility in blue.

```
Vix <- read.csv("VIX_History.csv") %>%
  mutate(
    Date_full = mdy(DATE)
  )
date_from <- max(min(Vix$Date_full), min(RV_series$Date))
date_to <- min(max(Vix$Date_full), max(RV_series$Date))
Vix_overlap <-Vix %>% filter(
    Date_full <= date_to & Date_full >= date_from
  )
RV_series_overlap <- RV_series %>% filter(
  Date <= date_to & Date >= date_from
)

Vix_overlap %>% ggplot(aes(x=Date_full, y=CLOSE))+
  geom_line(color="red", alpha=0.7) +
  geom_line(data=RV_series_overlap, aes(x=Date, y=sqrt(RV_ou_ann)*100), color="blue", alpha=0.6)
```



We now compute mean and variance for the Realized Volatility.

```
print(c(mean (sqrt(RV_series_overlap$RV_ou_ann)*100), var(sqrt(RV_series_overlap$RV_ou_ann)*100) ))
```

```
## [1]  15.64629 129.63694
```

And for the Vix index.

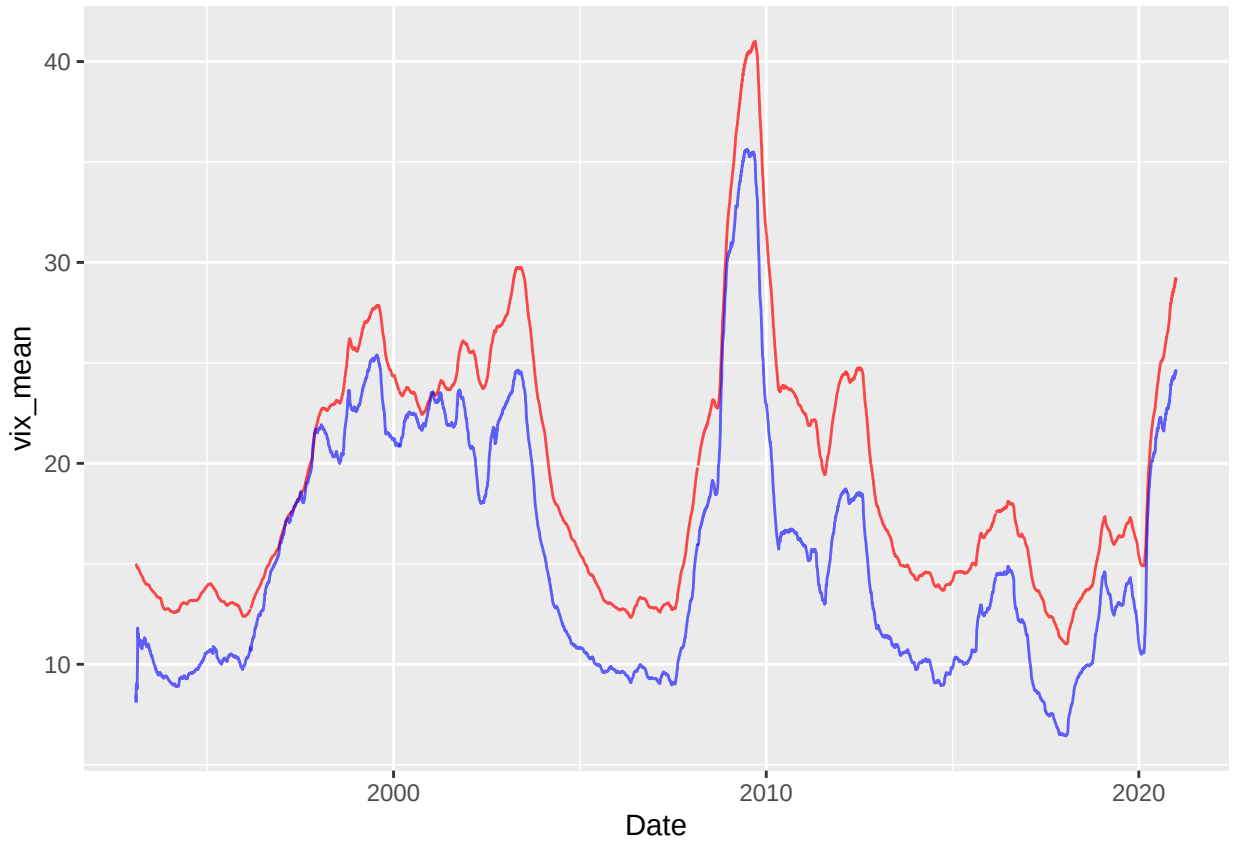
```
print(c(mean (Vix_overlap$CLOSE), var(Vix_overlap$CLOSE) ))
```

```
## [1] 19.54778 70.47127
```

Therefore, on the whole period we observe a higher mean but a lower variance for the Vix.

We now display the rolling mean and variance for both, we start with the mean, as before, VIX in red and realized in blue.

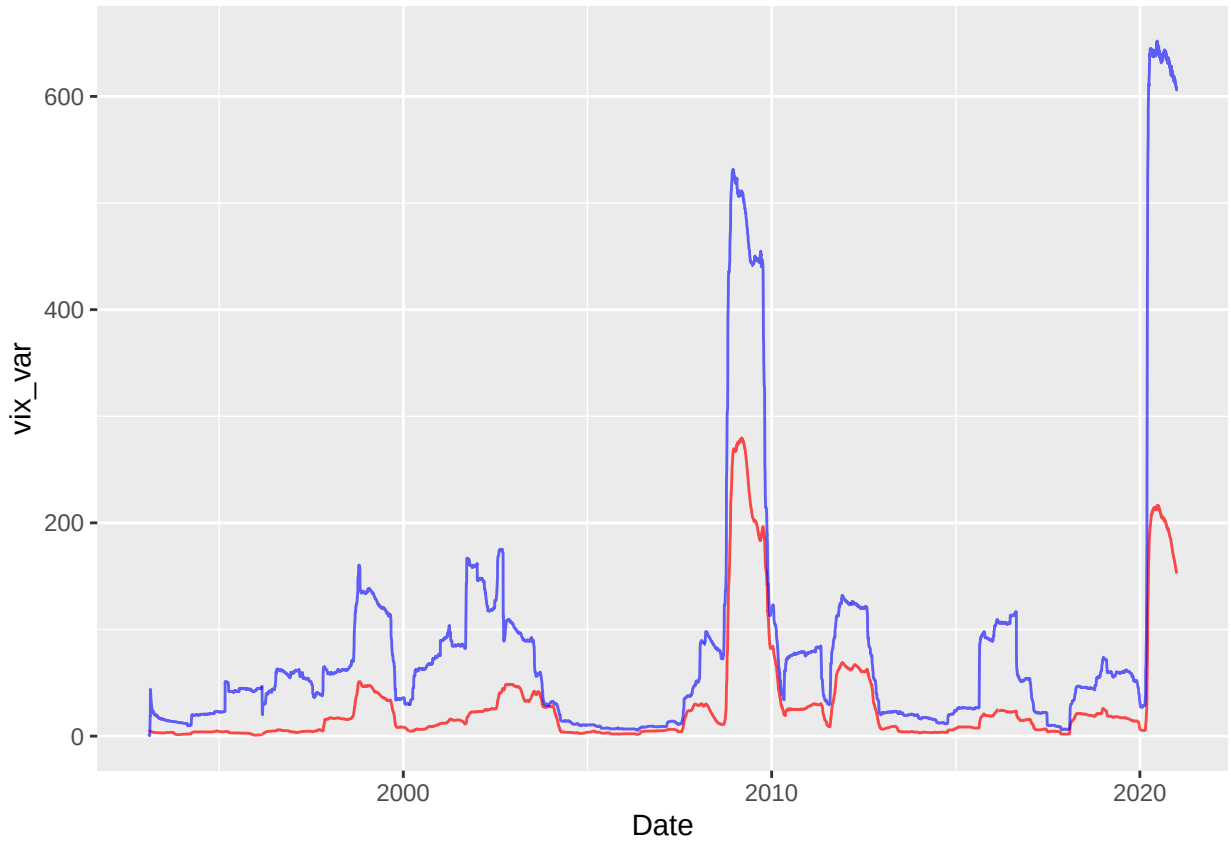
```
# There must be something better to have a rolling mean which is not dependent
# on the number of observation but only on the date. However, this
# implementation works (even if it isn't superefficient). I will think about
# that in the future.

vix_mean = rep(0, length(Vix_overlap$Date_full))
vix_var = rep(0, length(Vix_overlap$Date_full))

for(it in 1:length(vix_mean)){
  date_it = Vix_overlap$Date_full[it]
  date_start = date_it - period(year = 1)
  temp_vec <- Vix %>% filter(
    Date_full >= date_start & Date_full <= date_it
    ) %>%
    pull(CLOSE)
  vix_mean[it] <- mean(temp_vec, na.rm=T)
  vix_var[it] <- var(temp_vec, na.rm=T)
}

RV_mean = rep(0, length(RV_series_overlap$Date))
RV_var = rep(0, length(RV_series_overlap$Date))
for(it in 1:length(RV_mean)){
  date_it = RV_series_overlap$Date[it]
  date_start = date_it - period(year = 1)
  temp_vec <- RV_series %>% filter(
    Date >= date_start & Date <= date_it
    ) %>%
    pull(RV_ou_ann) %>%
    sqrt()*100
  RV_mean[it] <- mean(temp_vec, na.rm=T)
  RV_var[it] <- var(temp_vec, na.rm=T)
}


vix_summary <- data.frame(Date = Vix_overlap$Date_full, vix_mean, vix_var)
RV_summary <- data.frame(Date = RV_series_overlap$Date, RV_mean, RV_var)

vix_summary %>% ggplot(aes(x=Date,y=vix_mean)) +
  geom_line(color="red", alpha=0.7) +
  geom_line(data=RV_summary, aes(x=Date, y=RV_mean), color="blue", alpha=0.6)
```



And for the variance we get

```
vix_summary %>% ggplot(aes(x=Date,y=vix_var)) +
  geom_line(color="red", alpha=0.7) +
  geom_line(data=RV_summary, aes(x=Date, y=RV_var), color="blue", alpha=0.6)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

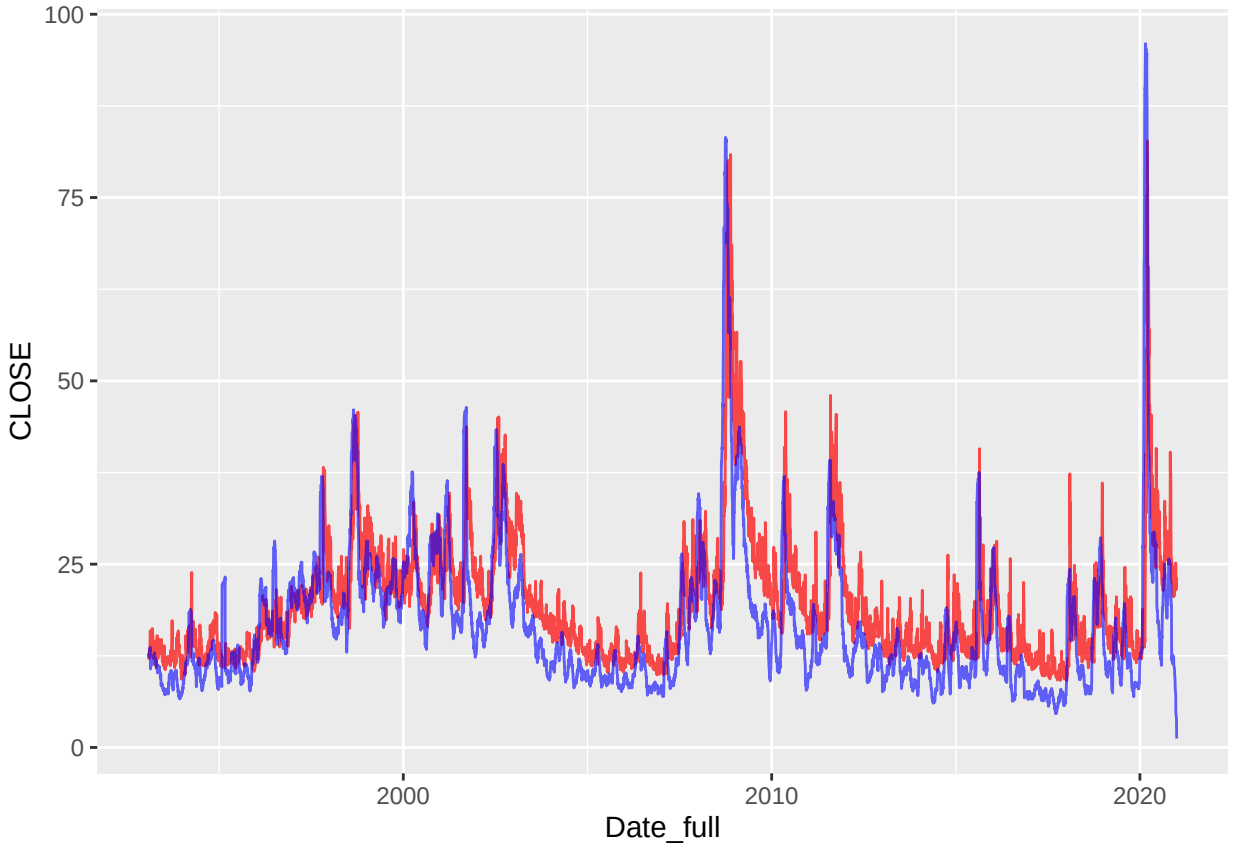


This is consistent with what we found overall, the VIX displays a higher mean but a lower variance.

6

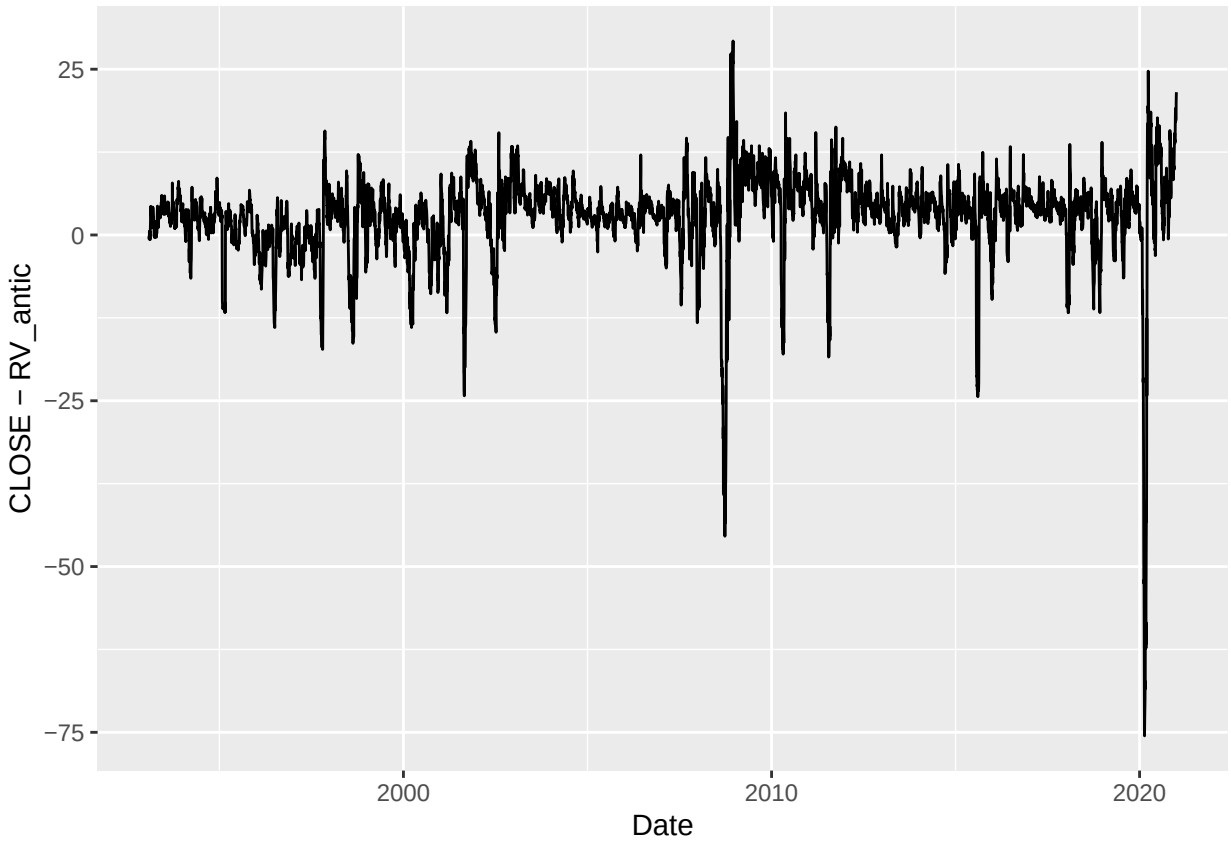
Now we will find the realized RV for the subsequent month.

```
RV_antic = rep(0, length(RV_series_overlap$Date))
for(it in 1:length(RV_mean)){
  date_it = RV_series_overlap$Date[it]
  date_end = date_it + period(days = 30)
  temp_vec <- RV_series %>% filter(
    Date <= date_end & Date >= date_it
    ) %>%
    pull(RV_ou_ann)
  RV_antic[it] <- 100*sqrt(sum(temp_vec, na.rm=T)/252*12)
}

RV_a_summary <- data.frame(Date = RV_series_overlap$Date, RV_antic)

Vix_overlap %>% ggplot(aes(x=Date_full,y=CLOSE)) +
  geom_line(color="red", alpha=0.7) +
  geom_line(data=RV_a_summary, aes(x=Date, y=RV_antic), color="blue", alpha=0.6)
```



We now plot the difference between the VIX and the RV

```
colnames(Vix_overlap) <- c("DATE", "OPEN", "HIGH", "LOW", "CLOSE", "Date")
merged <- merge(Vix_overlap, RV_a_summary, by="Date")
merged %>% ggplot(aes(x=Date, y=CLOSE-RV_antic)) +
  geom_line()
```



The mean and the variance of the VRP are

```
print(c(mean(merged$CLOSE-merged$RV_antic, na.rm=T), var(merged$CLOSE-merged$RV_antic, na.rm=T)))
```

```
## [1]  2.678533 45.665483
```