Machine Learning for Finance, Homework 1

Maximilian Hein
Mahdiazhari Austian

May 2022

1 Exponential Weighted Moving Average Variances (EWMA-Var)

Start with the original equation of Exponential Weighted Moving Average (EWMA):

$$\sigma^2_{EWMA}(t) = (1-\lambda)\sum_{k=1}^{m}\lambda^{k-1}r^2_{t-k} \tag{1}$$

Now, move the equation back one period such that

$$\sigma^2_{EWMA}(t-1) = (1-\lambda)\sum_{k=2}^{m+1}\lambda^{k-1}r^2_{t-k} \tag{2}$$

multiply by λ

$$\lambda\sigma^2_{EWMA}(t-1) = \lambda * (1-\lambda)\sum_{k=2}^{m+1}\lambda^{k-1}r^2_{t-k} \tag{3}$$

subtract the third equation from the first

$$\sigma^2_{EWMA}(t) - \lambda\sigma^2_{EWMA}(t-1) = (1-\lambda)\sum_{k=1}^{m}\lambda^{k-1}r^2_{t-k} - \lambda * (1-\lambda)\sum_{k=2}^{m+1}\lambda^{k-1}r^2_{t-k} \tag{4}$$

$$\sigma^2_{EWMA}(t) = \lambda\sigma^2_{EWMA}(t-1) + (1-\lambda)\sum_{k=1}^{m}\lambda^{k-1}r^2_{t-k} - \lambda * (1-\lambda)\sum_{k=2}^{m+1}\lambda^{k-1}r^2_{t-k} \tag{5}$$

now verify that: (start by dividing by $(1-\lambda)$

$$(1-\lambda)\sum_{k=1}^{m}\lambda^{k-1}r^2_{t-k} - \lambda * (1-\lambda)\sum_{k=2}^{m+1}\lambda^{k-1}r^2_{t-k} = (1-\lambda)r^2_{t-1} \tag{6}$$

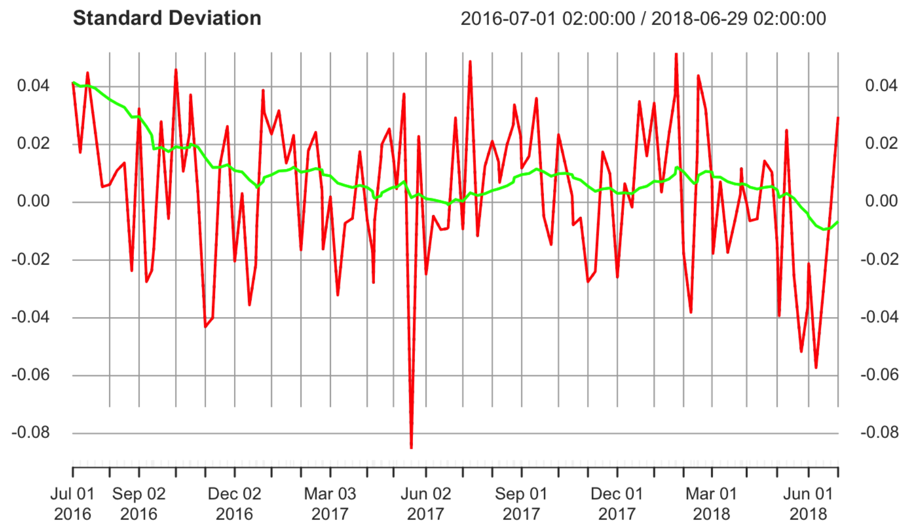


Figure 1: SD vs EMA Brazil

$$\sum_{k=1}^{m} \lambda^{k-1} r_{t-k}^2 - \lambda * \sum_{k=2}^{m+1} \lambda^{k-1} r_{t-k}^2 = r_{t-1}^2 \tag{7}$$

Given that these sums are almost exactly the same, except for their "start" and "end", it is trivial to see that the last sums cancel to:

$$\sigma_{EWMA}^2(t) = \lambda \sigma_{EWMA}^2(t-1) + (1-\lambda) r_{t-1}^2 \tag{8}$$

For the EMA estimation vs standard deviation, we took Brazil's index from number 3 and computed the two metrics and plotted them in Figure 1. The green line represents the EWMA Variances whilst the red line is the rolling standard deviation (we used rollapply function to calculate this). We can observe that these two measures are close but the rolling standard deviation is more volatile compared to the EWMA variances. We can observe that the EWMA variance is smoother due to the smoothing parameter lambda.

2 Forecasting X_{t+h} with information set Z

Start by noting that the forecast for X_{t+h} is a function of the information set:

$$F(Z) = \hat{X_{t+h}} \tag{9}$$

Next, the joint distribution of X and Z is normally distributed:

$$\begin{bmatrix} X \\ Z \end{bmatrix} = N(\begin{bmatrix} \mu_X \\ \mu_Z \end{bmatrix}, \begin{bmatrix} \sigma_X X & \sigma_X Z \\ \sigma_Z X & \sigma_Z Z \end{bmatrix}) \tag{10}$$

Another way of putting this problem is to note that we try to minimize the (squared) error of our prediction:

$$\underset{F(:)}{min} \, E[X - F(Z)]^2 \tag{11}$$

note that

$$E[X - F(Z)]^2 \tag{12}$$

$$E[(X - E(X|Z)) + (E(X|Z) - F(Z))]^2 \tag{13}$$

$$E[X - E(X|Z))^2 + 2E(X - E(X|Z)) * (E(X|Z) - F(Z)) + E(E(X|Z) - F(Z)]^2 \tag{14}$$

the middle terms is equal to

$$E_{Y,Z}[X - E(X|Z)][E(X|Z) - F(X)] \tag{15}$$
$$E_Z E_{X|Z}[X - E(X|Z))(E(X|Z) - F(Z)] \tag{16}$$
$$E_X[E_{X|Z}(XE(X|Z) - E(X|Z)^2 - F(Z)E_{X|Z}X + F(Z)E(X|Z)] \tag{17}$$
$$E_X[E(X|Z)^2 - E(X|Z)^2 - F(Z)E(X|Z) + F(Z)E(X|Z)] = 0 \tag{18}$$

so equation 13 is clearly minimised when

$$F(Z) = E(X|Z)$$

3 Causality Analysis and Volatility Estimation

Our team got the period 2016/07-2018/06. Our code can be found in hw1_3.rmd file.

We firstly had to handle the NAs present in the data, we used the imputeTS library and used interpolation technique to do this.

Figure 2 shows the NAs present in the weekly jKSE data. We then utilized the grangertest function in lmtest package to do granger tests for both weekly and monthly data from 1-4 lags. For the volatility calculation, we used the EMA package from TTR.

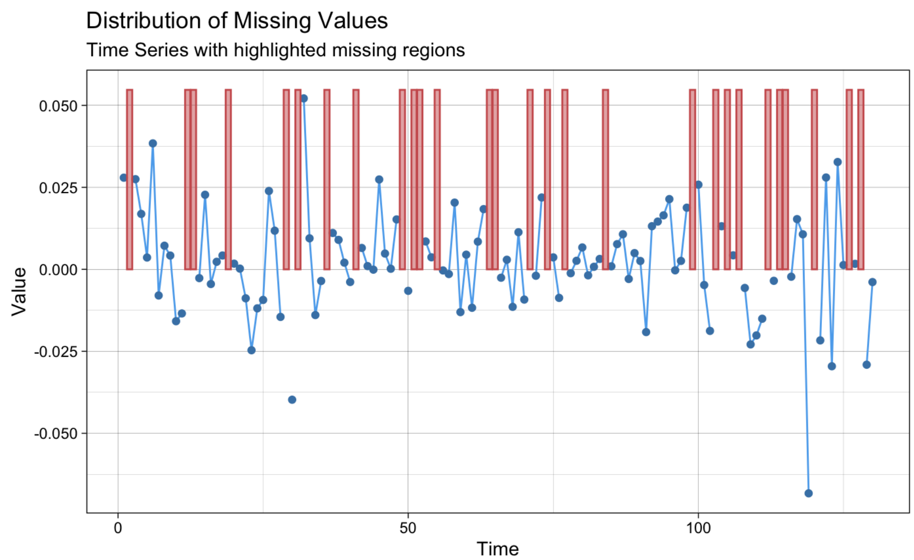


Figure 2: NAs in JKSE weekly data

Weekly tables are attached in: Table 1 which shows the weekly granger test for returns and Table 3 shows the weekly granger test for volatility.

Monthly tables are shown in: Table 2 which shows the monthly granger test for returns and Table 4 shows the monthly granger test for volatility.

The 1 indicates if there is a significant causality at the 5 % level.

3.1 Discussion of Results

Granger's test for causality assumes a linear relation among the causes and effects. Given two random variables X, Y : X is said to (Granger) cause Y , if Y can be better predicted using the histories of both X and Y than by using the history of Y alone.

We can utilize causality analysis to select meaningful (in terms of forecasting potential) variables to predict others.

In this case, it might be that some stock markets have close association between one another, perhaps money from one country moves to another country in response to shocks between markets or because of close International trade and cooperation between them.

For instance, when we observe Monthly granger test for returns in Table 2 Indonesia (JKSE), for our chosen time period, the UK stock index, USA, and China have significant causality in affecting monthly market returns across varying lags. For the weekly returns, it is now China, Mexico, and Japan returns having causality across varying lags.

Indonesia seems to affect USA on 4 lags for monthly returns, while for the other way around USA -¿ Indonesia it seems to affect Indonesia only for 2 lags. However, it cannot be exactly said that the relationship is bidirectional.

Speaking of weekly returns in Table 1, Japan has a significant causality with many countries across different lags, despite having zero causality results for the monthly returns.

In the case of monthly volatility in Table 4, we can immediately see that the US stock market affects a lot of exchanges in other countries. We also observe that Indonesia and Taiwan are affected by 5 other countries across different lags. When we switch to weekly volatility (table 3, now it is the Chinese stock exchange that is affected by the most countries. And the Taiwan stock exchange weekly is affecting 5 countries. This seems to align with the fact that Taiwan is one of the world's largest exporters.

These results might vary across different time horizons chosen. But it does give us a good idea of what kind of effects foreign market returns or volatility might have on the stocks we want to predict, and whether it will affect the forecasting on a weekly or monthly scale.

	country	India	Brazil	UK	Germany	USA	China-Shanghai	Spain	Indonesia	Mexico	Japan	Taiwan	VLIC	VIX
1	India	0	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
2	Brazil	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
3	UK	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(0, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)
4	Germany	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
5	USA	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)
6	China-Shanghai	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
7	Spain	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
8	Indonesia	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
9	Mexico	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
10	Japan	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 1, 0, 0)	c(0, 0, 1, 1)	c(0, 1, 1, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(1, 1, 1, 1)
11	Taiwan	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	0	c(1, 1, 1, 0)	c(1, 1, 0, 0)
12	VLIC	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)
13	VIX	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0

Table 1: Weekly Granger Test for Returns

	country	India	Brazil	UK	Germany	USA	China-Shanghai	Spain	Indonesia	Mexico	Japan	Taiwan	VLIC	VIX
1	India	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)
2	Brazil	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
3	UK	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(1, 1, 1, 0)
4	Germany	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)
5	USA	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 1, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(0, 0, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
6	China-Shanghai	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 1, 0, 1)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
7	Spain	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)
8	Indonesia	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 1, 1, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(1, 1, 1, 1)
9	Mexico	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
10	Japan	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
11	Taiwan	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)
12	VLIC	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 1, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)
13	VIX	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0

Table 2: Monthly Granger Test for Returns

	country	India	Brazil	UK	Germany	USA	China-Shanghai	Spain	Indonesia	Mexico	Japan	Taiwan	VLIC	VIX
1	India	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
2	Brazil	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 1, 0)	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
3	UK	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)
4	Germany	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
5	USA	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	0	c(1, 1, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
6	China-Shanghai	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
7	Spain	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
8	Indonesia	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
9	Mexico	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 1, 1, 1)	c(0, 1, 1, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)
10	Japan	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(1, 1, 1, 1)	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 1, 1, 1)
11	Taiwan	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(1, 1, 0, 0)	c(0, 0, 1, 1)	c(0, 1, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(1, 1, 1, 1)	c(0, 1, 1, 0)
12	VLIC	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(1, 1, 1, 1)
13	VIX	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 1, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 1, 1)	c(1, 1, 1, 0)	c(0, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	0

Table 3: Weekly Granger Test for Volatility

	country	India	Brazil	UK	Germany	USA	China-Shanghai	Spain	Indonesia	Mexico	Japan	Taiwan	VLIC	VIX
1	India	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 1)	c(0, 0, 0, 1)	c(0, 0, 0, 1)	c(0, 1, 1, 1)	c(0, 1, 0, 1)
2	Brazil	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)
3	UK	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 1, 1, 1)
4	Germany	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(1, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(0, 0, 0, 0)
5	USA	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(1, 1, 0, 0)	0	c(0, 1, 0, 0)	c(1, 1, 0, 0)	c(1, 0, 1, 1)	c(0, 0, 1, 1)	c(1, 1, 1, 0)	c(0, 1, 1, 1)	c(1, 1, 0, 1)	c(0, 0, 0, 0)
6	China-Shanghai	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 0)	0	c(0, 0, 0, 0)	c(1, 1, 1, 1)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(1, 1, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)
7	Spain	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 1, 1, 0)
8	Indonesia	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(1, 0, 0, 0)	c(1, 1, 1, 1)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(1, 0, 1, 0)	c(0, 1, 1, 1)	c(1, 1, 1, 0)	c(1, 1, 1, 1)
9	Mexico	c(0, 0, 0, 0)	c(0, 0, 1, 1)	c(1, 1, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 1, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 0, 0)	c(0, 1, 1, 1)	c(1, 1, 0, 0)	c(0, 0, 0, 0)
10	Japan	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(0, 0, 1, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)
11	Taiwan	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(1, 0, 0, 0)	c(0, 0, 0, 0)
12	VLIC	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 0, 1)	c(1, 1, 1, 1)	c(0, 0, 0, 0)	c(1, 0, 0, 1)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	0	c(1, 0, 0, 0)
13	VIX	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(0, 0, 0, 0)	c(1, 1, 1, 0)	c(0, 0, 0, 0)	c(1, 0, 0, 0)	0

Table 4: Monthly Granger Test for Volatility

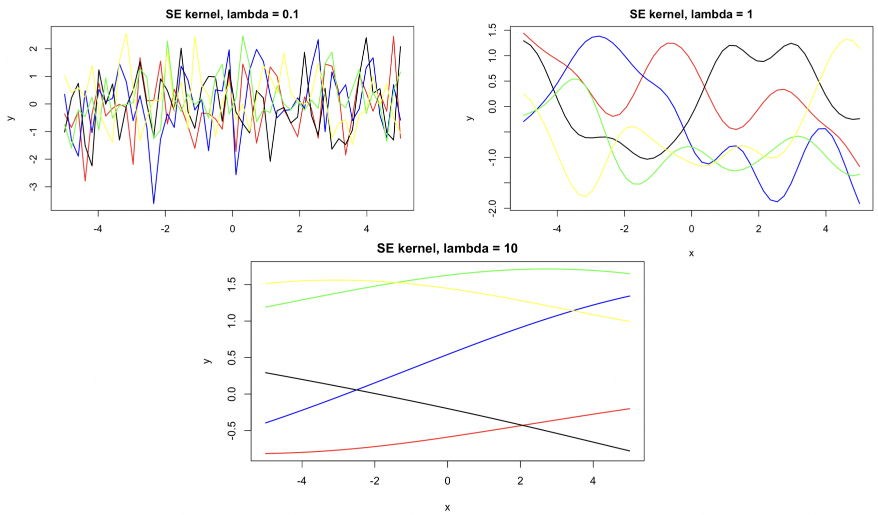


Figure 3: SE Kernel for increasing lambda

4 Kernels

We plotted 5 functions for the different kernels. Each of them sampled 50 times (50 points per functions). We first did that for various values of lambda for the SE kernel.

In the SE kernel, we see that for increasing values of lambda, the function becomes smoother as seen in Figure 3.

For the RQ function, we observe that higher alpha values do not change the appearance of the graphs very much as seen in Figure 4.

But the same trend regarding the lambdas is once more visible, wherein higher lambda values create smoother functions (as seen in the Appendix attached to the very last page).

We can observe that when we have a large alpha of 50 (closer to infinity) in Figure 5, and with lambda=1, we observe the same results as the SE kernel with lambda=1 (as seen in in Figure 3). Here is the Figure enlarged: 6.

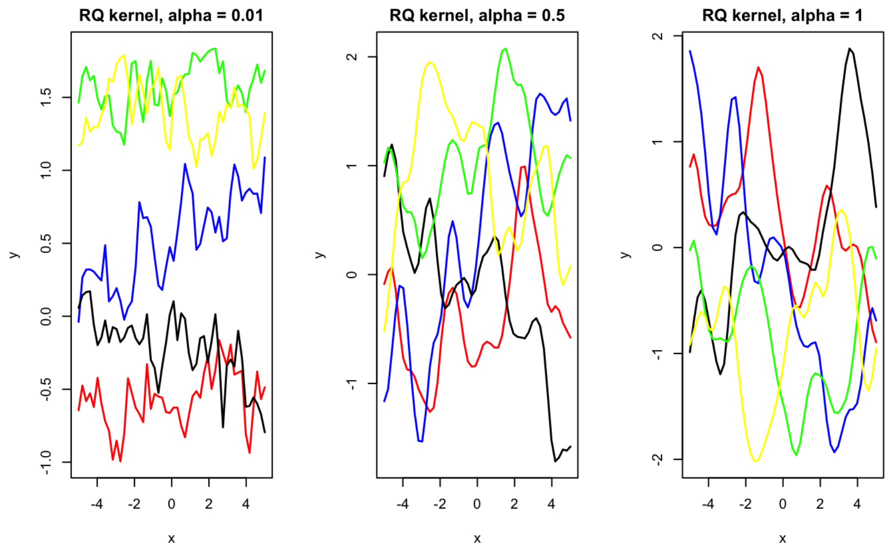


Figure 4: RQ Kernel for increasing alpha

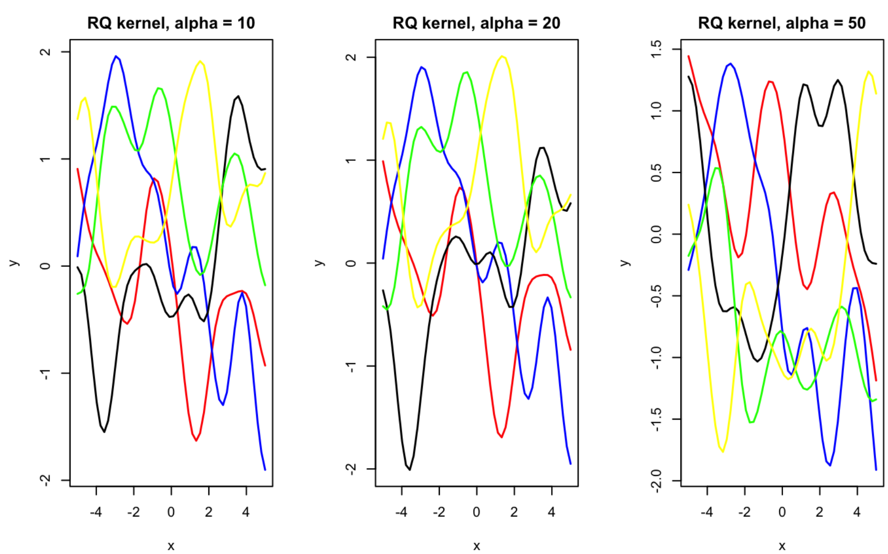


Figure 5: RQ Kernel for increasing alpha until alpha=50

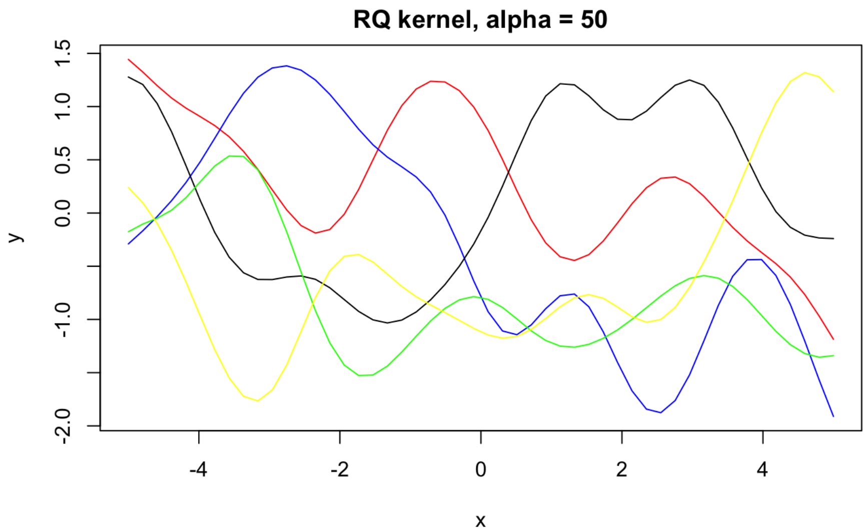


Figure 6: RQ Kernel for alpha=50 enlarged, it is the same as SE kernel with lambda=1

5 GP Forecasting

For this exercise, we evaluated the models based on the percentage of outperforming direct sample mean (sample expected value) and gap between train and test mse.

5.1 Variables

Target: For our target, we used the log returns.

Our team was assigned the following variables:

- ep (earnings to price)
- svar (stock variance)
- bm (book to market).

We took 3 lags for the log returns, and 2 lags for the variables. We followed the steps outlined in the RLab3 files to obtain our target and feature variables. Thus, our lag periods are monthly (tau=1 month).

5.2 Variable Selection Procedure

We used LASSO BIC to do Feature Selection. This will be implemented using the glmnet package in the HDEconometrics repository. Before running LASSO, we separated the data into test and training.

Sources:

- https://www.r-bloggers.com/2017/04/lasso-adalasso-and-the-glmnet-package/
- https://github.com/gabrielrvsc/HDeconometrics

5.2.1 LASSO Results

The following plot below 7 shows the variables going to zero as the penalty increases in the objective function of the LASSO.

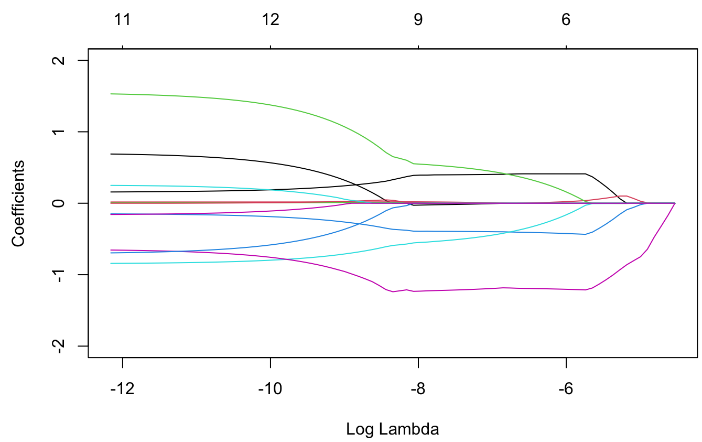


Figure 7: LASSO Process log Lambda

The plot in 8 below shows the BIC curve and the selected model (the lines).

Lasso output:

```
(Intercept)      lag.1      lag.2      lag.3         bm         ep       svar       bm.1       bm.2       ep.1
      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE      FALSE       TRUE      FALSE
       ep.2     svar.1     svar.2
      FALSE      FALSE      FALSE
```

We can see that bm (book-to-market lagged 2 months) is the variable to drop.

5.3 Kernels

We used three kernels:

- Custom Kernel
- RBF Kernel
- Vanilladot kernel

5.3.1 Custom Kernel

```
    MyKer <- function(x,y) {
  2*exp(sum(abs(x-y))/(-2*1.5^2)) + 1.5*sum(x*y)
}
```

The above kernel function was given to us in RLabGPLab3.R. It seems to be a variation of a dot kernel that is similar to what can be found in number 4 (k4). We obtained a train error of 1.133e-06/0.000001133.

We plotted line plots in and scatter plots in Figure 9 of actual vs predicted and observed that the model was able to follow the trends in the stock market during the period.

percentage of outperforming : 61.86034. gap: 0.0002899509.

We obtained a 61 percent of outperforming the mean. With a positive gap, we believe that the testing error is higher than the training error. Even though it is very small, we believe that it might show signs that this model is slightly overfitting, this can be seen in the very small training error.

5.3.2 RBF Kernel

The RBF/SE kernel was the second kernel used. We obtained a train error of 0.002258327.

The scatter plot and line graphs are in Figure: 10.

percentage of outperforming: 12.86225 gap: -0.0007389061

This kernel gives the worst results compared to the custom kernel and vanilladot kernel. We obtained a 13 percent of outperforming the mean, which is very poor compared to our custom kernel.

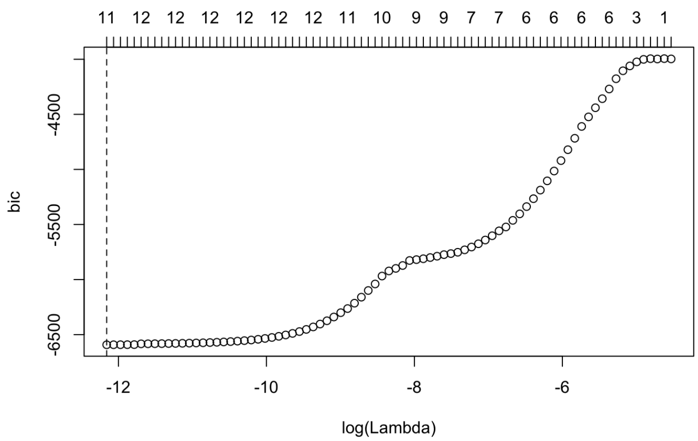
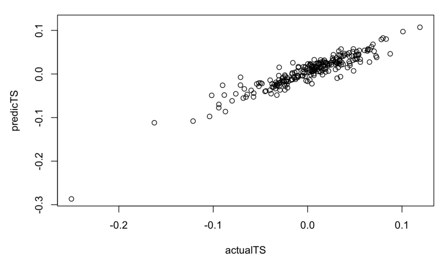
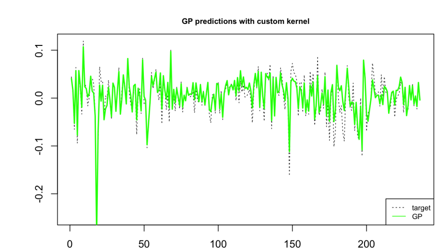


Figure 8: LASSO BIC Curve

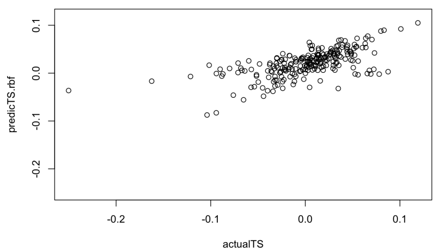


(a) custom kernel scatter

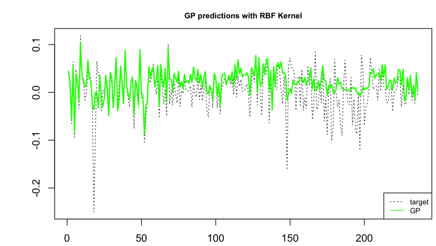


(b) custom kernel lines

Figure 9: Custom Kernel Results



(a) rbf scatter



(b) rbf lines

Figure 10: RBF Kernel Results

5.3.3 Vanilladot

The Vanilladot kernel was the last kernel tested. We obtained a train error of 0.022946675. We can immediately see that the model's training error is the highest compared to the two before this.

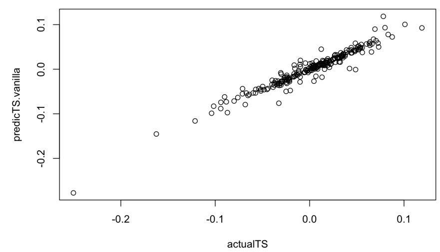
The scatter plot and line graphs are in Figure: 11.

percentage: 74.61839 gap: -0.02281776

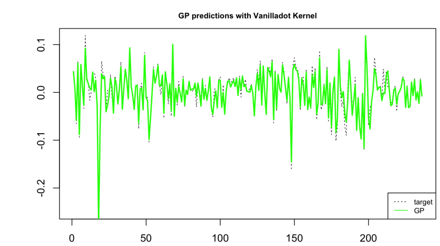
Despite the high training error, the model obtains the highest percentage of outperforming the mean, and with a negative gap indicating that performance during tests is the best.

5.4 Conclusion

Based on our target of Log Returns, the best kernel is the vanilladot kernel. It obtains the best percentage of outperforming the mean whilst maintaining good indicators of not overfitting. Based on the performance, It seems that the custom kernel is a variation of the dot product kernel but perhaps with hypertuned parameters. Regarding variable selection, Lasso identified one variable that could be thrown away, which is the 2 period lag of book to market. Famous economists have proposed different financial indicators as information to forecast the SP500 index. Most notably the Price-to-Earning ratio (P/E). In our exercise, the ep was not dropped by our variable selection, perhaps we were



(a) vanilladot scatter



(b) vanilladot lines

Figure 11: Vanilladot Kernel Results

seeing this outcome in practice. For further improvements, we could utilize more hyperparameter tuning on the kernels to improve their predictive performance.

6 Ornstein-Uhlenbeck Process

We consider an Ornstein-Uhlenbeck process with a kernel of absolute differences $k(x_i, x_j) = k(|x_i - x_j|) = \frac{\sigma^2}{2*\gamma} * e^{(-\gamma|i-j|)}$ for t ϵ $(0, \infty)$. If λ and σ ¿ 0, the process takes the following form:

$$U_{\lambda,\sigma}(t) = \sigma \int_{-\infty}^{t} e^{-\lambda(t-s)}, dB(s)$$

or, given as its correspondent differential equation:

$$dU_{\lambda,\sigma}(t) = -\lambda U_{\lambda,\sigma}(t)dt + \sigma dB(t)$$

According to this equation, the trajectory of the Ornstein-Uhlenbeck process x is a sequence dependent on Brownian motion, which exhibits mean-reversion (we assume the mean to be equal to zero) with a reversal speed given by λ; the σ determines the magnitude of the white noise term. For samples of the Ornstein-Uhlenbeck process with regular time intervals iτ: i = 0,1,2, ... , n for τ ¿ 0, the series $X_i = x(i\tau)$ can be written as an AR(1) process. This follows from the fact that:

$$X_{i+1} = \sigma \int_{-\infty}^{(i+1)\tau} e^{-\lambda((i+1)\tau - s)} dB(s)$$

$$= \sigma e^{-\lambda(i\tau - s)} \int_{-\infty}^{i\tau} e^{-\lambda(i\tau - s)} dB(s) + \sigma \infty_{i\tau}^{(i+1)\tau} e^{-\lambda((i+1)\tau - s)} dB(s)$$

$$= e^{-\lambda\tau} X_i + Z_{i+1}$$

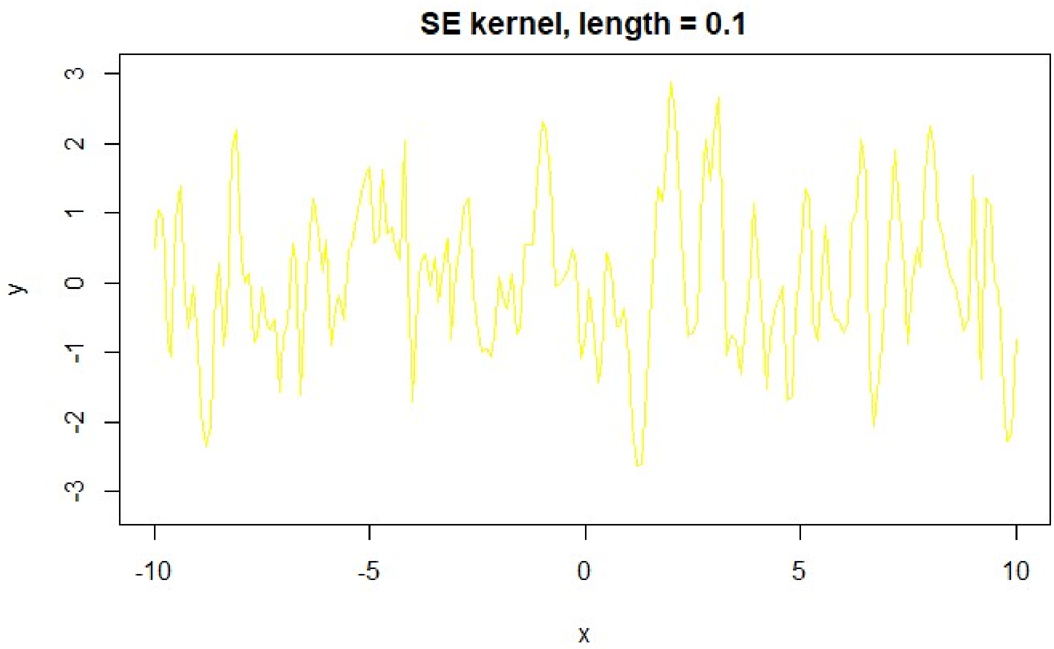
where $Z_{i+1} = \sigma \int_{i\tau}^{(i+1)\tau} e^{-\lambda((i+^)\tau - s)} dB(s)$ is a Gaussian innovation independent of x(t): t$\leq i\tau$ and B(t): t$\leq i\tau$ with variance:

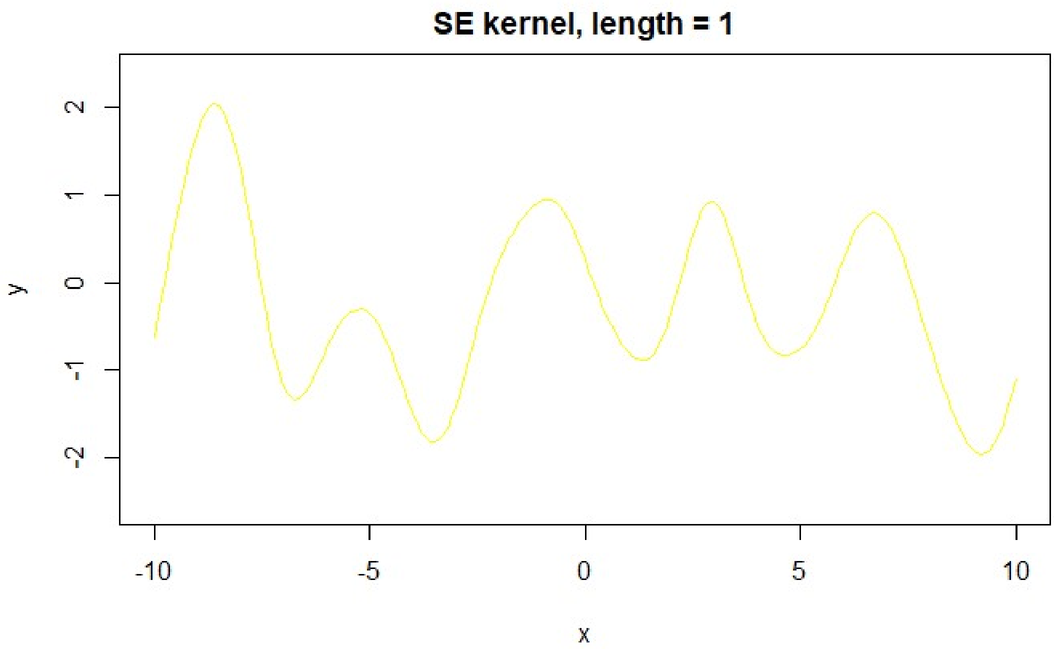
$$\sigma^2 \int_{i\tau}^{(i+1)\sigma} e^{-2\lambda((i+1)\tau - s} ds$$

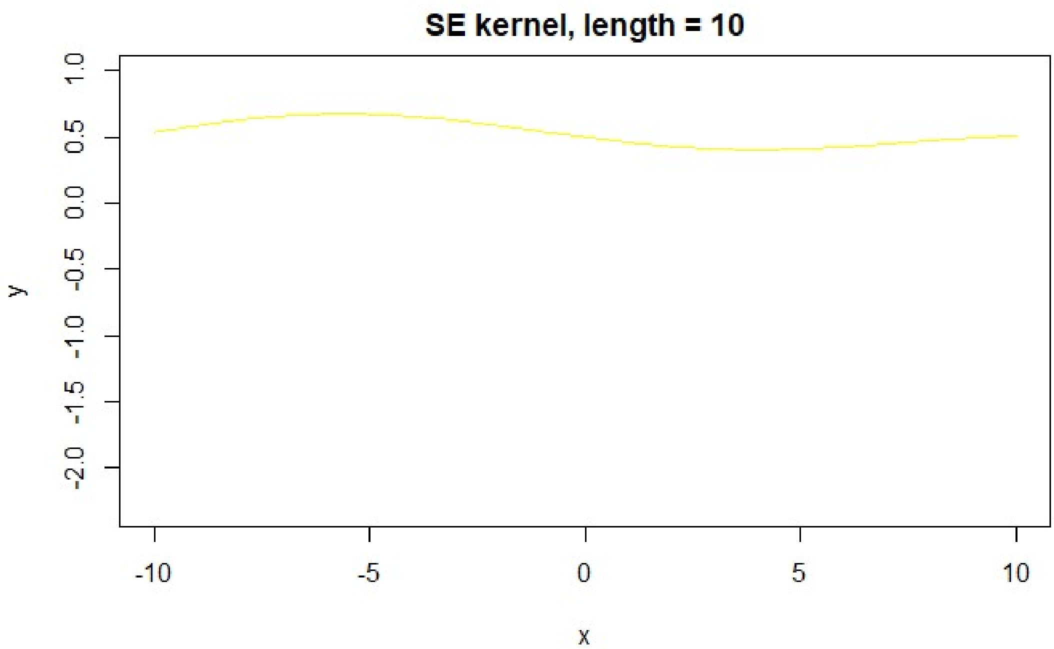
$$= \sigma^2 \int_{-\tau}^{0} e^{2\lambda s} ds$$

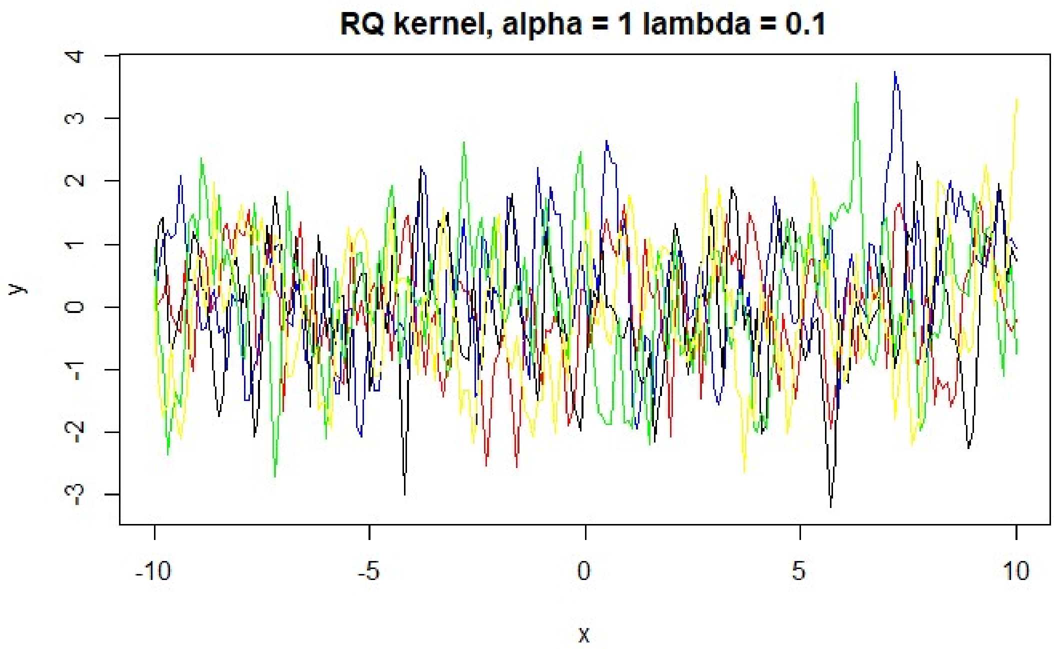
$$= \frac{\sigma^2}{2\lambda}(1 - e^{-2\tau\lambda})$$

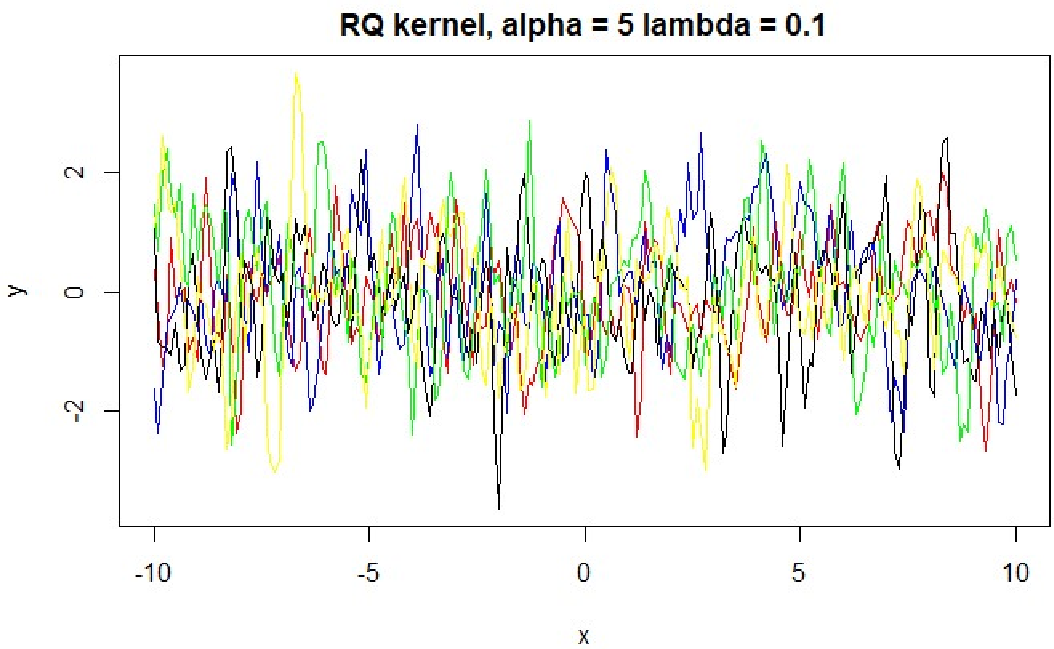
That means that the Ornstein-Uhlenbeck process is a continuous time interpolation of an AR(1) process.

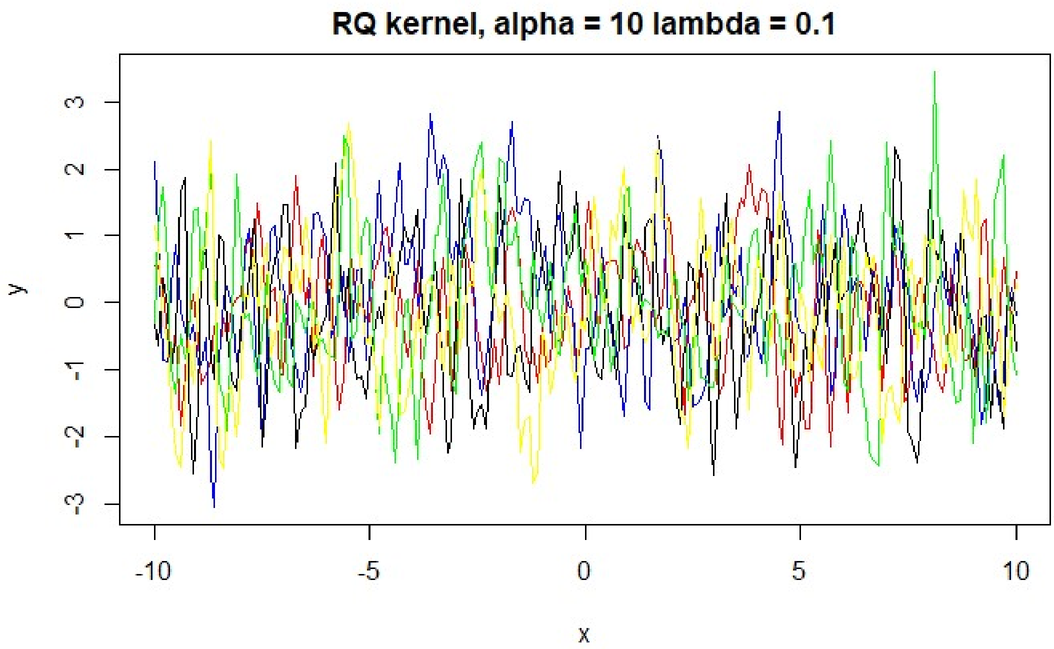

SE kernel, length = 0.1
y
3
2
1
0
-1
-2
-3
-10
-5
0
5
10
x

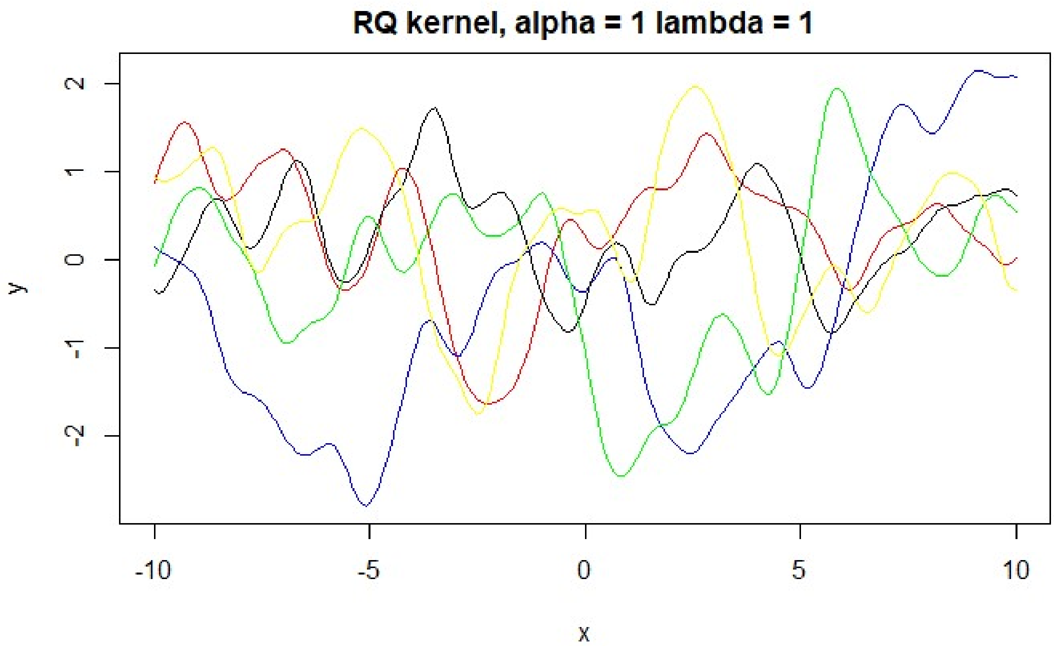

SE kernel, length = 1
y
2
1
0
-1
-2
-10
-5
0
5
10
x

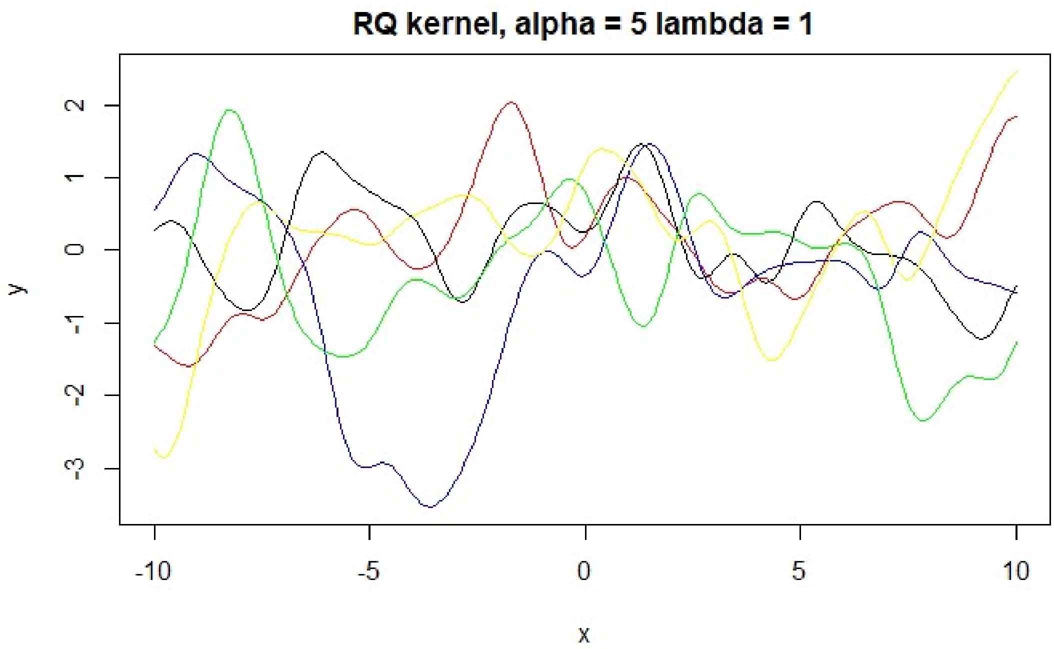

SE kernel, length = 10
y
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
-10
-5
0
5
10
x

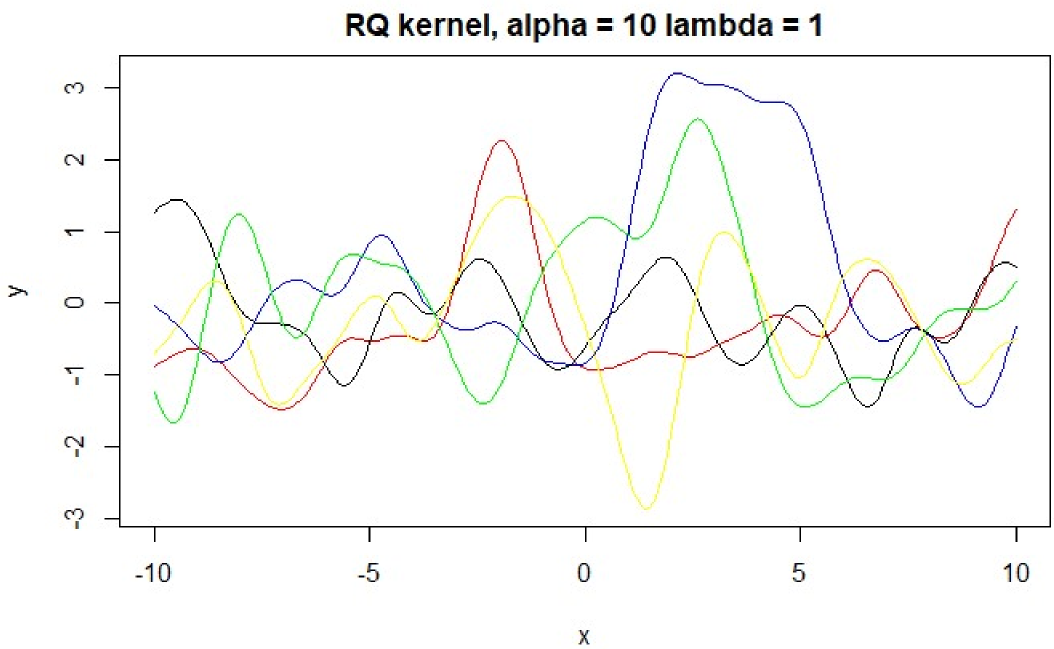

RQ kernel, alpha = 1 lambda = 0.1
y
x
-10
-5
0
5
10
-3
-2
-1
0
1
2
3
4

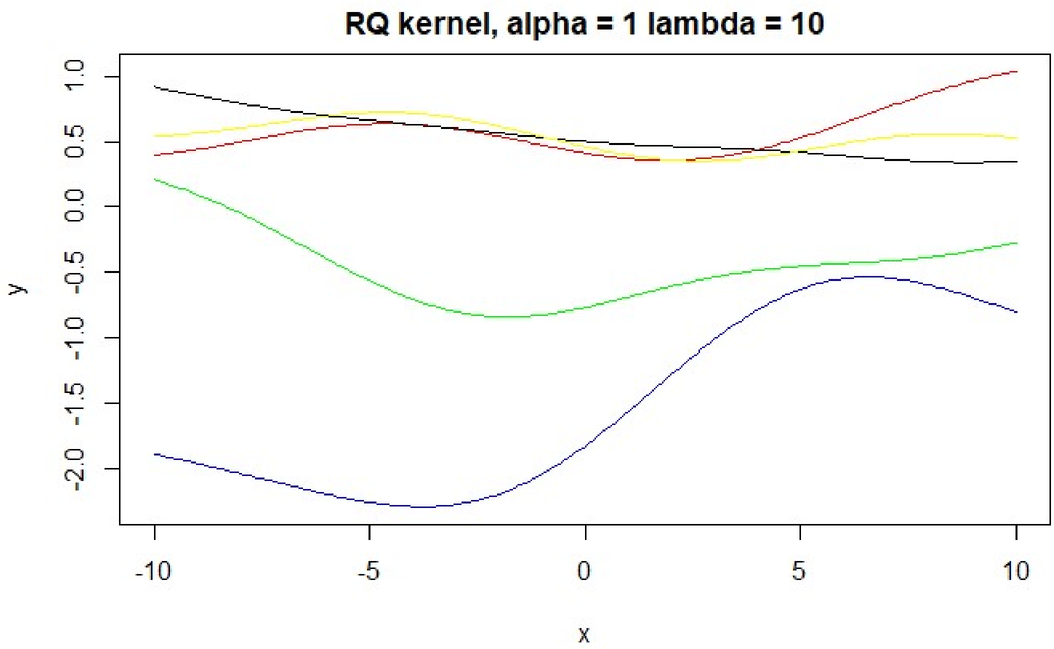

RQ kernel, alpha = 5 lambda = 0.1
y
x
-10
-5
0
5
10
-2
0
2

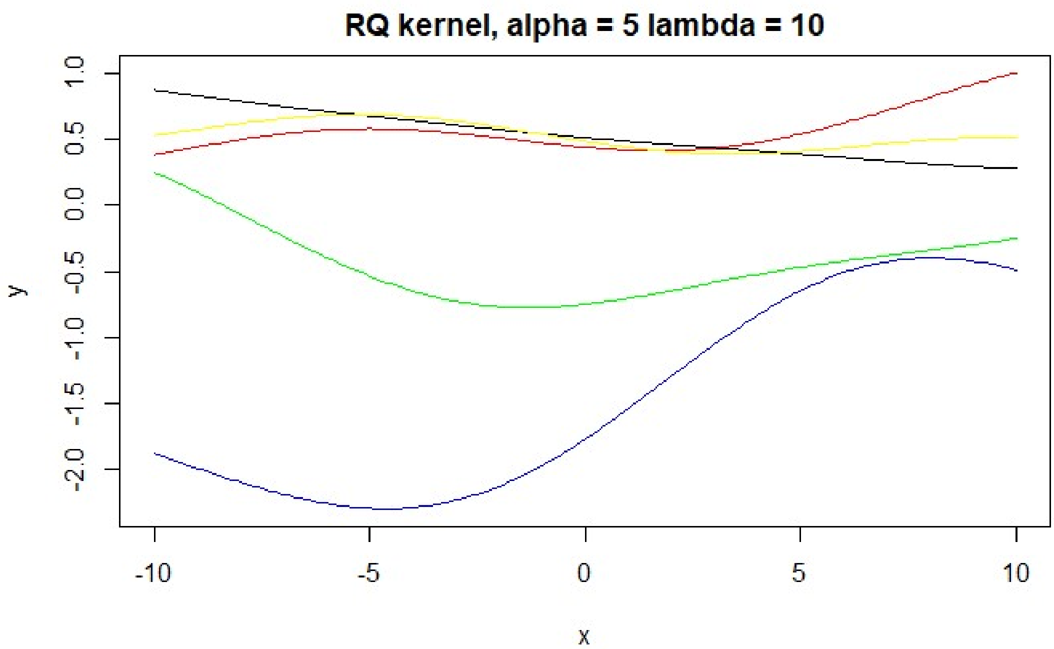

RQ kernel, alpha = 10 lambda = 0.1
y
3
2
1
0
-1
-2
-3
-10
-5
0
5
10
x

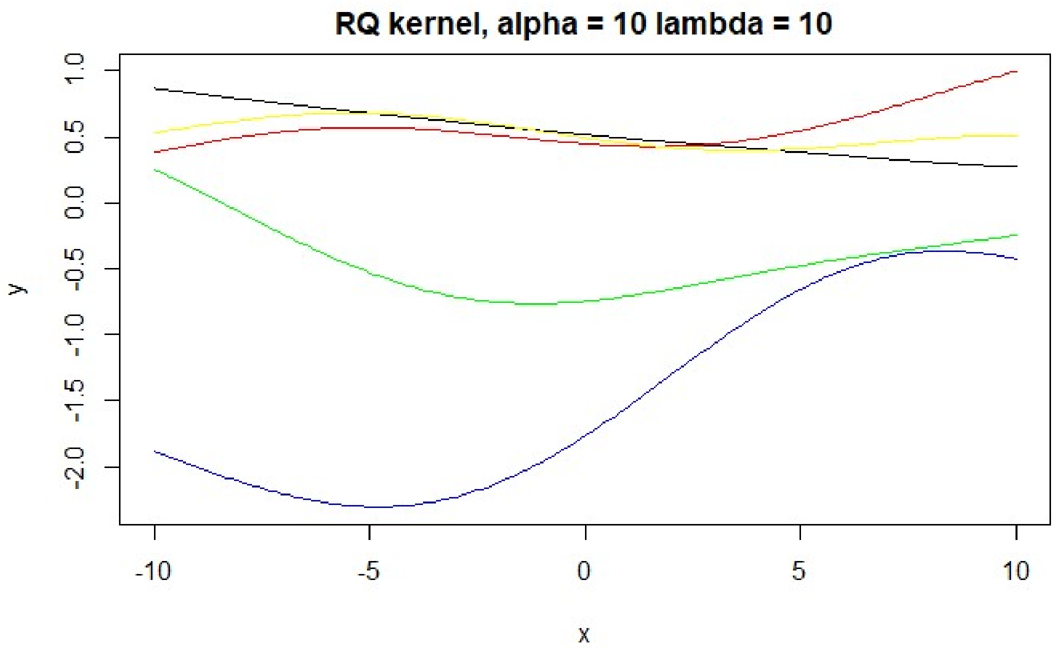

RQ kernel, alpha = 1 lambda = 1
y
2
1
0
-1
-2
-10
-5
0
5
10
x


RQ kernel, alpha = 5 lambda = 1
y
x
-10
-5
0
5
10
-3
-2
-1
0
1
2


RQ kernel, alpha = 10 lambda = 1
y
x
-10
-5
0
5
10
-3
-2
-1
0
1
2
3


RQ kernel, alpha = 1 lambda = 10
y
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
-10
-5
0
5
10
x


RQ kernel, alpha = 5 lambda = 10
y
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
-10
-5
0
5
10
x


RQ kernel, alpha = 10 lambda = 10
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
y
-10
-5
0
5
10
x