Solutions to the Exam Part 2 (June 2021)

Advanced Empirical Finance

General remarks

The exam is a written assignment consisting of two parts:

- Part 1: The first part is based on one of the mandatory assignments worked on during the course. The student can use the peer feedback received during the course to improve this assignment.
- Part 2: The second part is a new assignment given in English.

The part consists of 4 problems. All problems must be solved. Hand-in a pdf with your solutions together with the associated `Rmarkdown` file or the `R` script that replicates your results. The source code has to create every figure, table and number used for your report. The approximate weight of each problem is stated. A problem can consist of different sub-questions that do not necessarily have equal weight. Within the report describe every major step and make sure that tables and figures are self-explanatory by providing meaningful captions and variable names.

To solve the exam, you get access to two datasets: `spy_data.rds` and `equity_premium_data.rds` which can be imported in R with the function `tidyverse::read_rds()`.

Problem 1 (Realized volatility, 20%)

Import the file `spy_data.rds` which contains a tibble with 2 columns: `ts` are minute-level time stamps and `price` contains the corresponding minute level close price for *SPY*, an ETF that tracks the S&P 500 index. The sample starts in August 1998 and ends in December 2019. For each day with regular trading hours the observations start at 09:35:00 and end at 15:55:00.

Problem: Compute the daily realized volatility based on minute-level observations. Clearly state the estimator in your report. Compute the daily close-to-close return and visualize the time-series of daily prices, returns and realized volatilities (you can assume the last observation per day corresponds to the close price).

Solution: The code below implements the required computations. The realized variance on day T is simply computed as

$$RV_T := \sum_{i=2}^{T_t} r_{i,T}^2$$

where T denotes the trading day, T_t is the number of observed minute level prices on that day and $r_{i,T} := \log(P_{T,i}) - \log(P_{T,i-1})$ are the minute-level log returns where $P_{T,i}$ is the i-th price observation on the trading day. The figure below illustrates the daily close-to-close (log) returns, prices and the realized volatility estimate (square root of the realized variance).

```
library(tidyverse)
library(lubridate)
library(kableExtra)
library(hms)
library(scales)
Sys.setenv(TZ='GMT')

data <- read_rds("spy_data.rds")

# Compute daily close-to-close returns and realized volatilities
daily_data <- data %>%
  group_by(ts = as.Date(floor_date(ts, "day"))) %>%
  mutate(return = (log(price) - log(lag(price)))) %>% # Intra-day returns
  summarise(price = last(price), # close price
            RV = 100 * sqrt(sum(return ^2, na.rm = TRUE))) %>% # Realized volatility in percent
  mutate(return = 100 * (log(price) - log(lag(price)))) %>% # Daily close-to-close returns in percent
  drop_na(return)

# Visualize time series
daily_data %>%
  pivot_longer(-ts, names_to = "Variable") %>%
  mutate(Variable = str_to_sentence(Variable)) %>%
  ggplot(aes(x = ts, y = value)) +
  geom_line() +
  facet_wrap(~Variable, ncol = 1, scales = "free_y") +
  theme(legend.position = "bottom") +
  scale_x_date(breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               minor_breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               expand = c(0,0),
               labels = date_format("%y")) +
  labs(x = "", y = "")
```

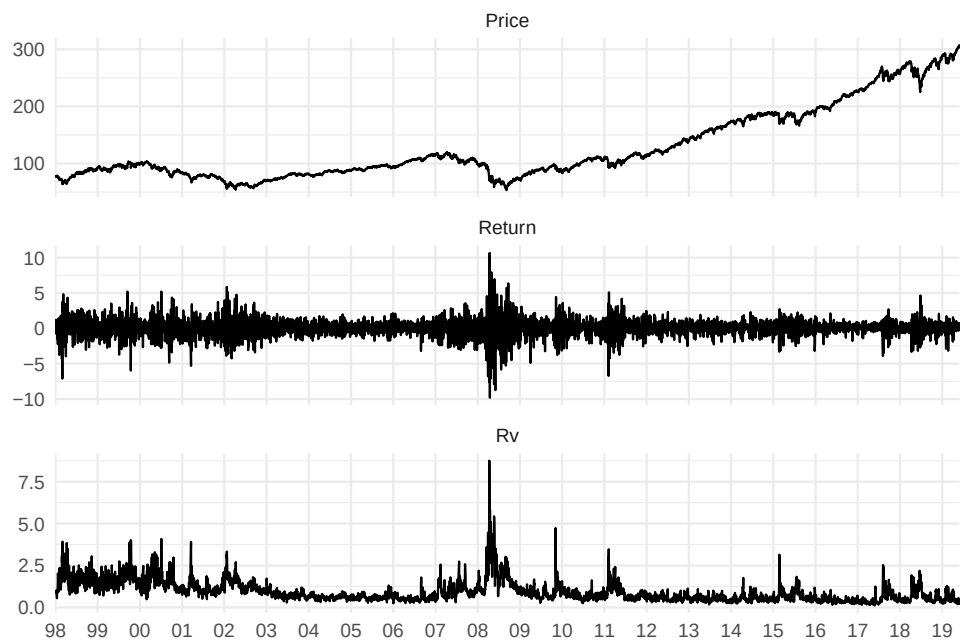

Figure 1: Time-series of daily SP 500 prices, returns and realized volatilies

- *Problem:* Provide meaningful summary statistics for the returns and the realized volatility and discuss the dynamics of the realized volatility time-series.

Solution: The choice and presentation of some summary statistics can obviously differ.

Table 1: Summary statistics: Values denote averages (standard deviations in brackets). Price is in USD, Return in percent and Rv is the annualized realized volatility

	Price	Return	Rv
1998 - 2001	89.29 (9.52)	0.03 (1.34)	27.02 (8.71)
2001 - 2001	73.28 (8.98)	-0.02 (1.39)	18.99 (7.29)
2004 - 2001	89.28 (7.38)	0.04 (0.66)	9.6 (2.45)
2007 - 2001	94.05 (18.13)	-0.02 (1.82)	19.65 (13.9)
2010 - 2001	104.72 (10.87)	0.04 (1.15)	12.5 (6.57)
2013 - 2001	165.52 (19.2)	0.06 (0.77)	8.82 (3.92)
2016 - 2001	226.5 (30.15)	0.03 (0.8)	8.61 (5.64)
2019 - 2001	279.07 (14.05)	0.11 (0.81)	8.6 (3.91)

Below I do provide a table based on grouped annual information regarding the time series. In combination with the Figure from exercise 1.2 one can conclude that realized volatility exhibits the "common" stylized facts such as persistence (optimally illustrated with the sample autocorrelation coefficients) and time-variation. Intuitively it should make sense that the "usual" periods of extreme distress such as the global financial crisis exhibit particular high volatility. The table includes the standard variation and mean for prices, but one can of course argue that for such obvious non-stationary time series the interpretation of the sample moments is difficult. Instead it is more meaningful to consider returns which have been positive on average.

```
# Summary statistics (in 3 year buckets)
daily_data %>%
    mutate(RV = RV * sqrt(250)) %>% # annualized values
    pivot_longer(-ts) %>%
    mutate(name = str_to_title(name)) %>%
    group_by(year = year(ceiling_date(ts, "3 years")), name) %>%
    summarise(mean = mean(value, na.rm = TRUE),
              sd = sd(value, na.rm = TRUE))  %>%
    mutate(Mean = paste0(round(mean, 2), " (", round(sd, 2), ")")) %>%
    select(-mean, -sd) %>%
    ungroup() %>%
  mutate(year = paste0(year - 3, " - ", min(2019, year))) %>%
  pivot_wider(names_from = name, values_from = Mean) %>%
  rename(" " = year) %>%
  kable(caption = "Summary statistics: Values denote averages (standard deviations in brackets).
        Price is in USD, Return in percent and Rv is the annualized realized volatility")
```

- *Problem:* Provide a brief discussion why the realized volatility may be subject to market microstructure noise and what the potential implications for realized volatility estimation are. Which sampling frequency should one choose in the absence of microstructure noise?

Solution: Microstructure frictions such as bid-ask spreads, asynchronous trading time, strategic order placement and endogenous liquidity provisioning do render the efficient price process unobservable and instead one usually assumes that the observed prices contain measurement error. Therefore, while it is theoretically optimal to sample at the highest frequency possible, market microstructure noise requires careful treatment in order to reduce the bias in the realized volatility estimator. Usually the literature proposes adjustments, e.g. based on sampling on different time scales, to overcome these issues. In the example above, sampling at high frequencies without adjusting for the presence of microstructure noise could lead to a biased estimate of the realized volatility.

Problem 2 (Volatility estimation, 25%)

Compute two measures of the (conditional) daily S&P 500 return volatility $\sigma_t^2 = \text{Var}\left(r_t | r_{t-1}, \ldots, r_1\right)$ based on the daily close-to-close returns: i) the rolling return standard deviation (based on the last 250 trading days) and ii) the conditional volatility based on a GARCH(1, 1) model

- *Problem:* Briefly state the estimation framework and explain the intuition behind the GARCH(1,1) specification. Which stylized facts of return time series justify (generalized) autoregressive conditional heteroscedasticity models and which potential shortcomings do you see in the selected GARCH(1,1) specification?

Solution: The GARCH(p,q) model is given by

$$r_t = \mu_t + \varepsilon_t \text{ where } \varepsilon_t = \sigma_t z_t \qquad \sigma_t^2 = \alpha_0 + \sum_{j=1}^{p} \alpha_j \varepsilon_{t-j}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2.$$

Positive σ_t^2 is ensured by $\alpha_0 \geq 0, \alpha_i \geq 0, \beta_j \geq 0$. μ_t can be computed sequentially. By constructing the conditional volatility as an autoregressive process, GARCH models can (up to a certain extent) capture persistence in the volatility. Estimation can be conducted via simple MLE, e.g. using the log likelihood function under the normality assumption for z_t

$$\begin{aligned} \log \mathcal{L}\left(\theta | r_1\right) &= \log f\left(r_2 | r_1 \theta\right) + \ldots + \log f\left(r_T | r_{t-1}, \ldots, r_1, \theta\right) \\ &= -\frac{T}{2} \log(2\pi) - \sum_{t=2}^{T} \left(\log(\sigma_t^2) + \frac{(r_t - \mu_t)^2}{2\sigma_t^2} \right) \end{aligned}$$

where σ_t^2 and μ_t^2 are computed recursively. Obvious restrictions of a GARCH(1,1) model include the pre-specified lag selection which may be sub-optimal as well as the presence of leverage effects which may call for asymmetric responses to past shocks.

- *Problem:* Illustrate the time-series of estimated conditional volatilities.

Solution: The figure below also contains the RV estimator for the sake of completeness.

```r
# Volatility estimation
library(slider)
library(rugarch)
model.spec <- ugarchspec(variance.model = list(model = 'sGARCH' , garchOrder = c(1 , 1)))
model.fit <- ugarchfit(spec = model.spec , data = daily_data %>% pull(return))

daily_data <- daily_data %>%
  mutate(Rolling = slide_dbl(return, sd,
                             .before = 250)) %>%
  mutate(Garch = (sigma(model.fit))) %>%
  unnest(Garch)

daily_data %>%
  select(ts, return, everything(), -price) %>%
  na.omit() %>%
  pivot_longer(-c(ts, return), names_to = "Model", values_to = "sigma") %>%
  ggplot(aes(x = ts,
             y = sqrt(sigma),
             color = Model)) +
  geom_line(alpha = 0.5) +
  labs(x = "", y = "Estimated Volatility") +
  theme(legend.position = "bottom") +
  scale_x_date(breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               minor_breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               expand = c(0,0),
               labels = date_format("%y"))
```

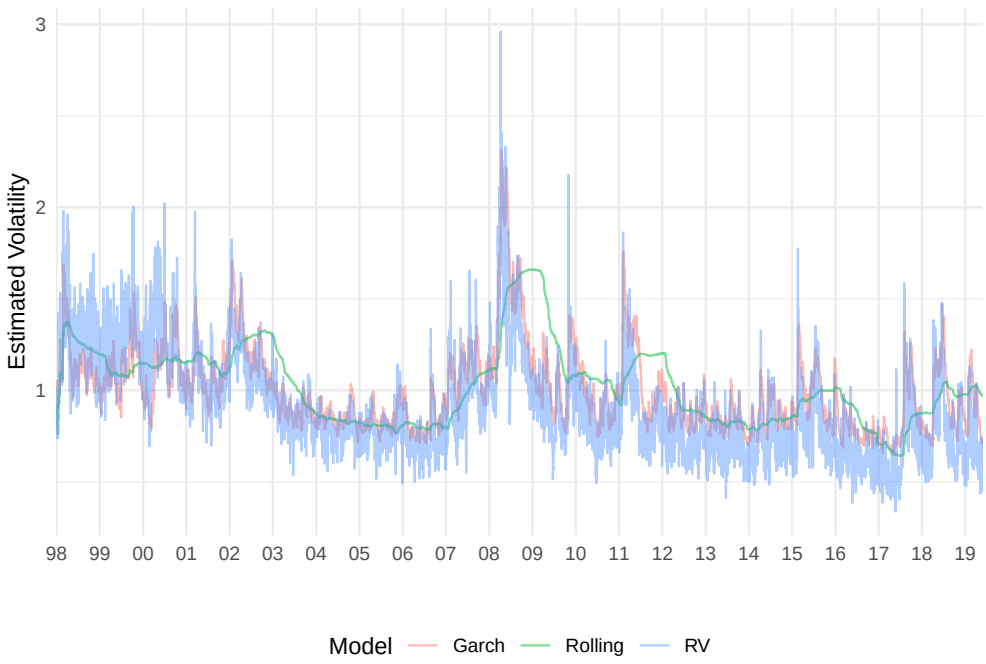

Figure 2: Comparison of daily conditional volatilities

- *Problem:* Discuss the differences between the two estimated conditional volatility dynamics and the potential implications for portfolio allocation decisions and risk management.

Solution: The most straightforward difference is of course the slow moving change for the rolling standard deviation in the conditional volatility based on long estimation windows. As a result, the rolling window estimates do not capture fast changes in the economy (market volatility). A GARCH model, instead, which incorporates the term ε_{t-1}^2 can react faster to more recent events. As a result, when comparing to the realized volatility, the GARCH model is very likely to provide better forecasts of future volatility. This is especially desirable when it comes to capital budgeting and risk management decisions. More accurate forecasts reduce capital requirements and improve allocational efficiency. On the other hand, in the presence of transaction costs it may be desirable to "smooth" allocation decisions over longer time horizons.

Problem 3 (Equity premium estimation, 25%)

Read in the dataset `equity_premium_data.rds`.

```
welch_data <- read_rds("equity_premium_data.rds")
```

It contains 6 columns with monthly data. `date` denotes the month, **tms** (*Term Spread*), **dfy** (*Default Yield Spread*), **bm** (*Book-to-Market Ratio*) and **svar** (*Stock Variance*) are predictors from the dataset in the paper *"A Comprehensive Look at the Empirical Performance of Equity Premium Prediction"* by Goyal and Welch (2008). **rp_div_lead** is the equity premium

one-month ahead defined as the difference between the (log) US market return and the risk-free rate in the next month.

- *Problem:* Discuss why macroeconomic variables could qualify as predictors for the equity risk premium. Would such return predictability contradict the notion of efficient markets?

Solution: While market efficiency naturally imposes an upper bound on asset return predictability, it is not at odds with efficient markets if the equity risk premium is predictable. The equity risk premium resembles the required compensation of investors for bearing the market risk. This compensation depends on the risk aversion and, more specifically, on the marginal utility from consumption. Business cycles during which consumption is low (high marginal utility) represent periods during which a higher compensation is required. Macroeconomic variables that predict consumption therefore qualify as predictors for the equity risk premium.

- *Problem:* Compute monthly one-step ahead equity premium forecasts based on a simple rolling window predictive regression specification of the form

$$r_{t+1} = \alpha + b_1 tms_t + b_2 dfy_t + b_3 bm_t + b_4 svar_t + \varepsilon_t$$

  where r_{t+1} is the one-month ahead equity premium. State and justify your chosen estimation window length in the report. To generate predictions, first estimate the parameters $\hat{\alpha}, \hat{\beta}_1, \ldots, \hat{\beta}_4$ and then compute the prediction $\hat{r}_{t+1} = \hat{\alpha} + \hat{b}_1 tms_t + \hat{b}_2 dfy_t + \hat{b}_3 bm_t + \hat{b}_4 svar_t$. Illustrate your predictions relative to the realized equity premia.

Solution: The sample code is below. I have choosen 10 years of monthly data in order to capture long-run components in the business cycle. The obvious problem of shorter estimation windows is the resulting higher variance in the estimator which may induce additional estimation uncertainty. From a statistical perspective the choice of the window length depends on the structure of the data: If the data series are stationary, then more observations clearly reduces the estimation uncertainty.

The figure below illustrates the limited amount of information in the predictive regression.

```
predictive_regression <- function(x) {
  if (nrow(x) < 30) { # minimum requirement for number of observed variables
    result <- tibble(date = x %>% filter(row_number() == n()) %>% pull(date),
                     pred_return = NA)
  } else {
    x_old <- x[-nrow(x), ]
    reg <- lm(rp_div_lead ~  tms + dfy + bm + svar, data = x_old)
    result <- tibble(date = x %>% filter(row_number() == n()) %>% pull(date),
           pred_return = predict(reg, x[nrow(x), ]))
  }
  output <- result %>%
    mutate(rp_div_lead = x %>% filter(row_number() == n()) %>% pull(rp_div_lead))
  return(output)
}

predicted_equity_premium <- slide_index_dfr(welch_data,
                                            welch_data$date,
                            ~predictive_regression(.x),
                            .before = months(120),
                            .complete = FALSE) %>%
  na.omit()

predicted_equity_premium %>%
  rename("Prediction" = pred_return,
         "Realized" = rp_div_lead) %>%
```

```
  pivot_longer(-date, names_to = " ") %>%
  ggplot(aes(x = date,
             y = value,
             color = ` `)) +
  geom_line() +
  labs(x = "Year", y = "Equity premium") +
  theme(legend.position = "bottom") +
  scale_x_date(breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               minor_breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               expand = c(0,0),
               labels = date_format("%y")) +
  geom_hline(aes(yintercept = 0), linetype = "dotted")
```

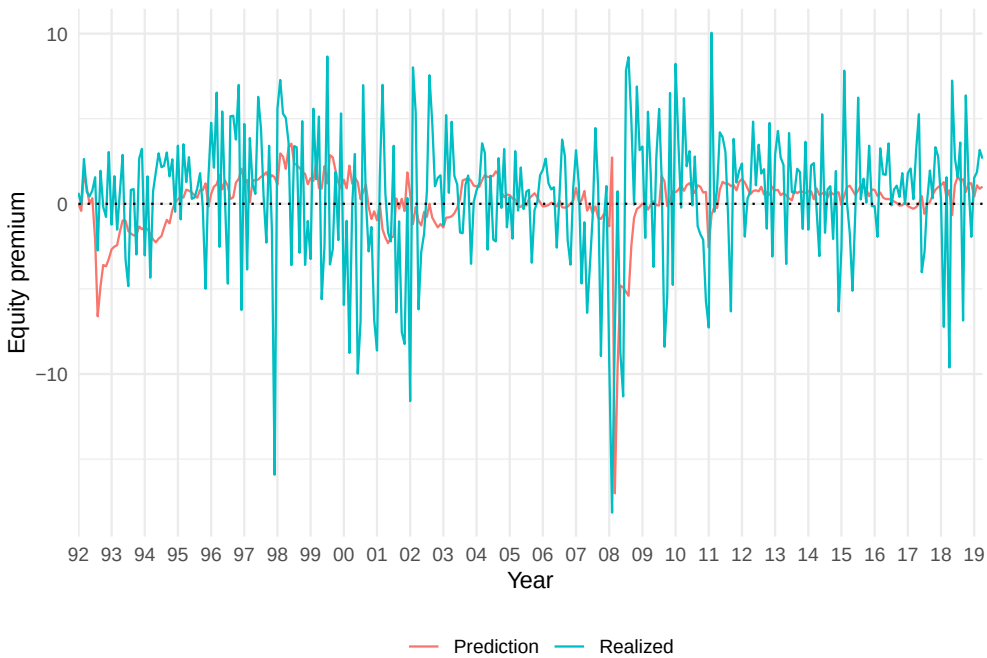

Figure 3: Equity premium and predictive regressions

```
daily_data <- daily_data %>%
  mutate(month = floor_date(ts, "month")) %>%
  left_join(predicted_equity_premium, by = c("month" ="date")) %>%
  na.omit() %>%
  select(-month)
```

- *Problem:* Discuss the forecasting approach in light of the results in the paper *"Empirical Asset Pricing via Machine Learning"* by Gu, Kelly and Xiu (2020). What are potential shortcomings of the predictive linear regression framework and what measures could one take to improve the predictive performance?

Solution: Gu, Kelly and Xiu (2020) focus on predicting the entire cross-section of stock returns instead of the market risk premium and do find benefits of applying machine learning methods for financial forecasting. More specifically, their results indicate that linear models are outperformed out-of-sample relative to nonlinear methods such as neural networks and regression trees and that shrinkage is key. The predictive regression from above neither features nonlinearities (e.g. reflecting that especially during periods during which firms exhibit financial distress required risk premia increase substantially, whereas economic upturns do not reflect such extreme shifts). Given the limited predictive ability, the predictive regression may benefit from imposing regularization methods such as Lasso or Ridge-regressions in order

to decrease the dimensionality of the problem even further.

Problem 4 (Static portfolio allocation with a single risky asset, 30%)

Consider the following portfolio choice problem: Two assets are available to an investor at time t. One is riskless with risk-free rate r_f. The risky asset is the market portfolio with conditional mean $\mu_t = E_t(r_{t+1})$ and conditional variance $\sigma_t^2 = \text{Var}_t\left(r_{t+1}\right)$. The investor puts a share α_t of her wealth into the market and $(1-\alpha_t)$ in the risk-free asset. The investor follows myopic mean-variance optimization such that

$$\alpha_t^* = \arg\max_a E_t(r_{pf,t+1}) - \frac{\gamma}{2}\text{Var}_t\left(r_{pf,t+1}\right)$$

where $r_{pf,t+1}$ are the portfolio returns from t until $t+1$ and γ is the risk-aversion.

- *Problem:* Derive the optimal investment α_t^* as a function of $\gamma, \sigma_t^2, \mu_t$ and r_f.

Solution: The investor allocates a fraction α of wealth to the risky asset and the remainder to the risk-free asset. The resulting portfolio return is thus

$$r_{\text{pf, t}} = \alpha r_t + (1-\alpha)r_f = r_f + \alpha(r_t - r_f).$$

The expected portfolio return is $E(r_{pf,t}) = r_f + \alpha E(r_t - r_f)$ and the variance is $\text{Var}\left(r_{pf,t}\right) = \alpha^2\sigma_t^2$. As a result, the optimization problem is simply

$$\alpha_t^* = \arg\max_a \alpha E(r_t - r_f) - \frac{\gamma}{2}\alpha^2\sigma_t^2 = \frac{\mu_t - r_f}{\gamma\sigma_t^2}.$$

- *Problem:* Suppose $\gamma = 4$ and $r_f = 0$ and set μ_t to the sample average of the annualized close-to-close returns of the S&P 500 (from Exercise 1). Use your estimated annualized conditional covariances $\hat{\sigma}_t^2$ (GARCH, the rolling window from Exercise 2 and RV_t from Exercise 1) and compute the resulting allocation α_t^* into the market for each day t for each of the three different volatility models. Discuss if the time series of $\hat{\alpha}_t^*$ seems intuitive: During which periods does the investor decide to implement a high exposure to the market risk?

Solution: The first panel in the Figure below illustrates the resulting fractions of wealth, α, invested in the market portfolio, proxied for by `SPY`, the S&P 500 tracking ETF. In general, the investment can exceed 100% - in that case, the investor simply borrows at the risk-free rate and invests the proceeds into the market. It is also not surprising that the investment is strictly positive at all times because the average market risk premium was positive during the last 30 years and $\sigma_t^2 > 0$. As expected, the investment decreases in σ_t and the time-series of α's therefore indicates that during volatile periods the investor tends to store her wealth in the risk-free asset to mitigate the increase in market risk. The setup therefore implies something which could be termed "flight-to-safety" as long as the volatility estimate/forecast reflects changes in the economy.

```r
# Allocation of wealth in market and risk-free rate (assumed to be 0)

sample_mu <- daily_data %>% pull(return) %>% mean() * 250

optimal_alpha <- function(sigma, mu = sample_mu, gamma = 4){mu / (gamma * sigma ^2)}

estimated_alphas <- daily_data %>%
  select(-c(price, rp_div_lead)) %>%
  pivot_longer(-c(ts, return, pred_return), names_to = "Model", values_to = "sigma") %>%
  mutate(alpha = map_dbl(sigma,
                         function(.) optimal_alpha(sigma = sqrt(250) * .)),
         alpha_return = map2_dbl(sigma, pred_return,
                                 function(., pred_return) optimal_alpha(sigma = sqrt(250) * .,
                                                                        mu = 12 * pred_return)))

estimated_alphas %>%
  select(ts, Model, "1. Volatility only" = alpha, "2. Volatility + Equity premium" = alpha_return) %>%
  pivot_longer(-c(ts, Model)) %>%
  ggplot(aes(x = ts,
             y = value,
             color = Model)) +
  geom_line(alpha = 0.5) +
  facet_wrap(~name, scales = "free_y", ncol = 1) +
  theme_minimal() +
  labs(x = "Year", y = "Fraction invested in market") +
  theme(legend.position = "bottom") +
  scale_x_date(breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               minor_breaks = function(x) seq.Date(from = min(x), to = max(x), by = "1 years"),
               expand = c(0,0),
               labels = date_format("%y"))
```

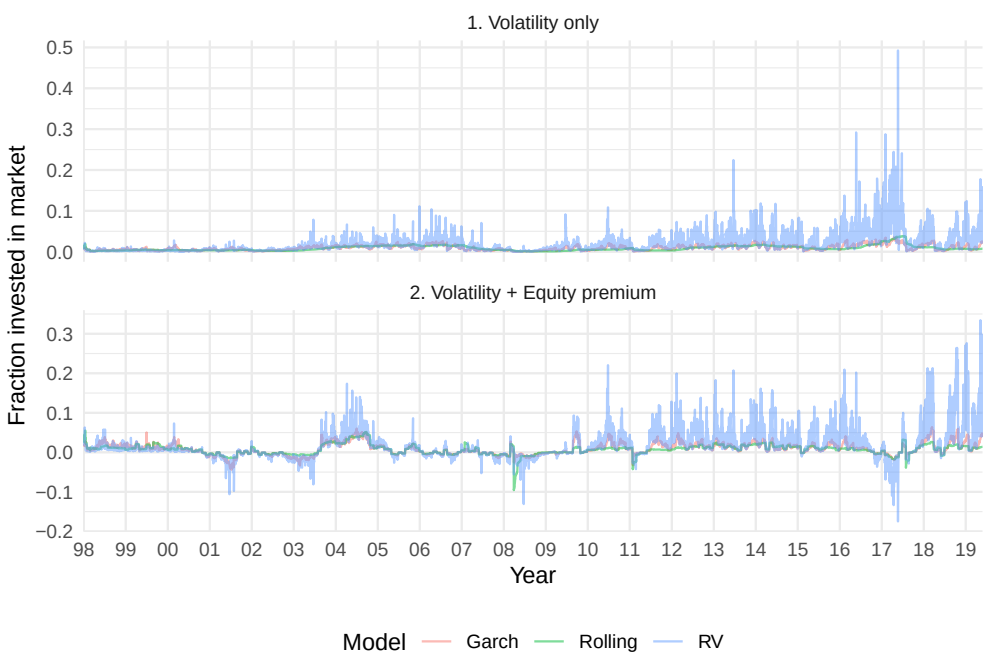

Figure 4: Investments in the market portfolio

- *Problem:* Briefly discuss how the investor could adjust for estimation uncertainty in her investment decision. How, do you expect, would the resulting optimal investment in the market portfolio change?

Solution: A common procedure would be to make use of Bayesian methods that rely on explicitly deriving the posterior predictive distribution of the returns which accounts for uncertainty in the estimated parameters. Such a setup is usually accompanied by imposing fatter tails on the return distribution and therefore implicitly reflects higher risk in the

investment decision. Depending on the utility function that could entirely change the investment strategy of the investor. In the case of a mean-variance framework the resulting investment into the market portfolio is going to be less ($|\alpha|$ would be lower) than in the case above where the investor ignores estimation uncertainty. The smaller $|\alpha|$ therefore reflects the risks due to estimation uncertainty.

- *Problem:* What is the resulting Sharpe ratio? What do the results imply with respect to the predictive performance of the individual volatility models?

Solution: The question is answered together with the following question.

- *Problem:* What is the resulting Sharpe ratio if you instead suppose that μ_t changes each month according to your annualized equity return predictions $\hat{\mu}_t$ from Problem 3? Can you conclude that macroeconomic variables improve the portfolio performance?

Solution: The table below illustrates the annualized Sharpe ratios of the different allocation strategies (with and without making use of the predictive regressions). There is a clear increase in the average returns when taking high-frequency data into account. At the same time, the resulting portfolio return standard deviation as a measure of the riskiness of the investment is highest when using the RV estimator. The table reveals that the GARCH specification does not necessarily deliver superior results relative to the rather naive rolling window estimation. In terms of Sharpe ratios, the rolling window framework even performs best when ignoring the additional information from the predictive regression. Incorporating additional information from Exercise 3 has two major effects: The variation in the portfolio holdings increases, negative excess return predictions are associated with shorting the market. Despite the noise, the macroeconomic predictors seem to convey some information and improve the out-of-sample performance substantially. As a result, high-frequency data paired with macroeconomic data delivers the highest possible Sharpe ratio.

```r
# Out-of-sample performance
estimated_alphas %>%
  select(ts, return, Model, "1. Volatility only" = alpha, "2. Volatility + Equity premium" = alpha_return) %>%
  pivot_longer(-c(ts, return, Model)) %>%
  group_by(Model, name) %>%
  mutate(out_of_sample_return =  value * lead(return, 1)) %>%
  na.omit() %>%
  summarise(mu = 250 * mean(out_of_sample_return),
            sd = sqrt(250) * sd(out_of_sample_return),
            `Sharpe Ratio` =  mu / sd) %>%
  rename(" " = name) %>%
  kable(booktabs = TRUE,
        digits = 3,
        caption = "Out-of-sample performance of the implemented allocation strategies. mu and sd are the annualized average return and standard deviation")
```

- *Problem:* Suppose rebalancing is costly. What would be your expectation on how the performance of the individual models changes if the investor would *not* incorporate transaction costs into her decision?

Solution: If rebalancing is costly but the investor would nevertheless follow her strategies, a higher variation in changes of the portfolio structure, $\mathrm{Var}\left(\alpha_t - \alpha_{t-1}\right)$ would harm the out-of-sample performance. The figure above illustrates that the realized volatility and GARCH models imply very high turnover and would thus be penalized more severely than the rolling window approach based on a long estimation window. If the structure of the transaction costs is known, the investor could obviously adjust her behavior in the spirit of Hautsch et al (2019).

Table 2: Out-of-sample performance of the implemented allocation strategies. mu and sd are the annualized average return and standard deviation

Model		mu	sd	Sharpe Ratio
Garch	1. Volatility only	0.050	0.125	0.400
Garch	2. Volatility + Equity premium	0.119	0.235	0.504
Rolling	1. Volatility only	0.064	0.127	0.501
Rolling	2. Volatility + Equity premium	0.217	0.431	0.503
RV	1. Volatility only	0.119	0.250	0.479
RV	2. Volatility + Equity premium	0.257	0.400	0.644