Examples for the plotseries package

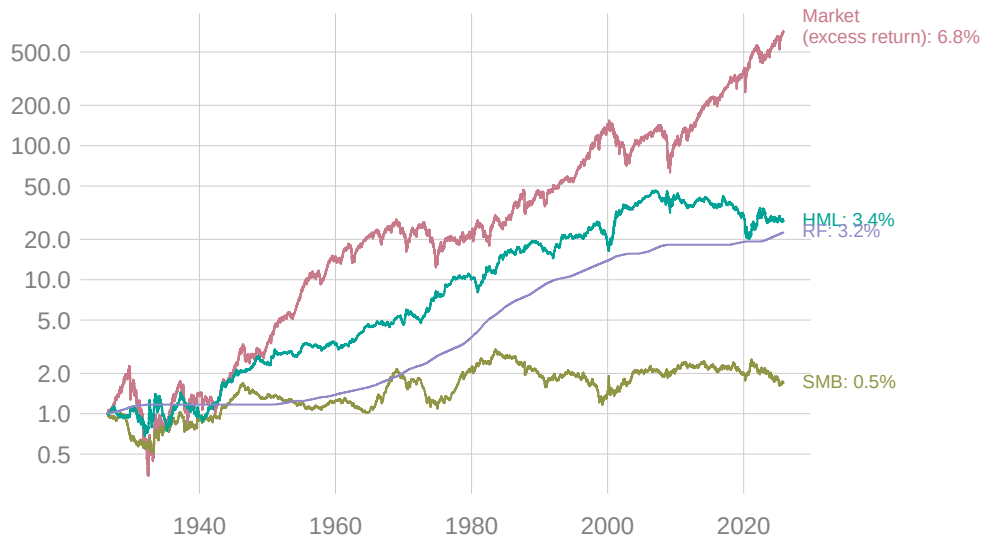
Package version 0.4-0

Enrico Schumann
es@enricoschumann.net

Time-series plots

Start with several classic time-series: the Fama–French factors [Fama and French, 1992]. Package NMOF provides a function `French` to fetch the most current data from Kenneth French's website.

```
> library("NMOF")
> series <- French("~/Downloads/French/",
                   "F-F_Research_Data_Factors_daily_CSV.zip",
                   frequency = "daily",
                   price.series = TRUE)
> plotseries(series,
             t = as.Date(row.names(series)),
             col = hcl.colors(n = ncol(series), palette = "Dark 2"),
             log.scale = TRUE,
             labels = c("Market\n(excess return)", "SMB", "HML", "RF"),
             labels.col = TRUE)
```

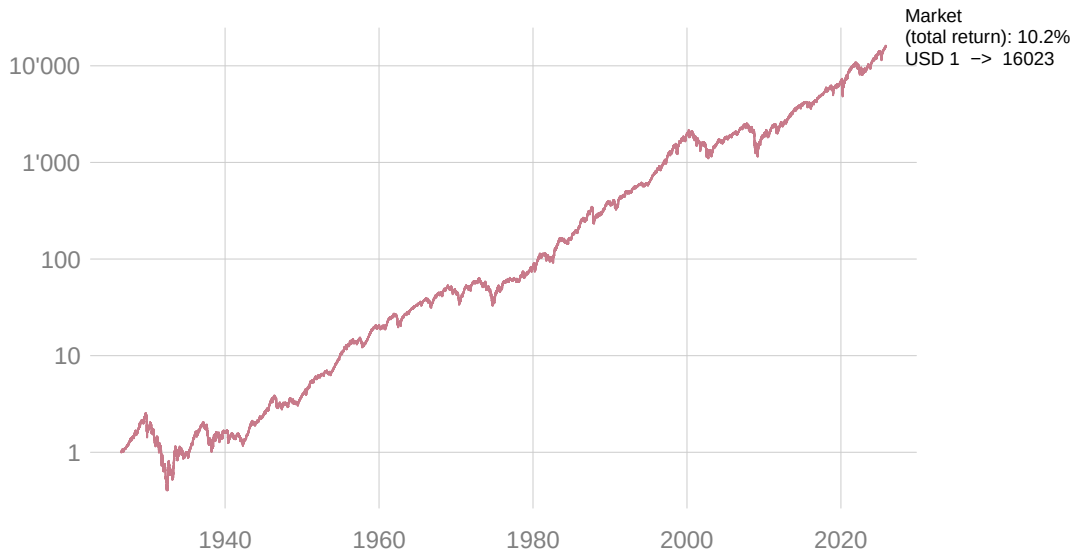


The numbers next to the labels show the annualized returns.

`French` also offers a series for the total return of the market, computed as the sum of T-bill and excess return, and then compounded.

```
> market <- French("~/Downloads/French/",
                   "market",
                   frequency = "daily",
                   price.series = TRUE)
> plotseries(market,
             t = as.Date(row.names(series)),
             col = hcl.colors(n = 12, palette = "Dark 2"),
             log.scale = TRUE,
             dollars.show = TRUE,
```

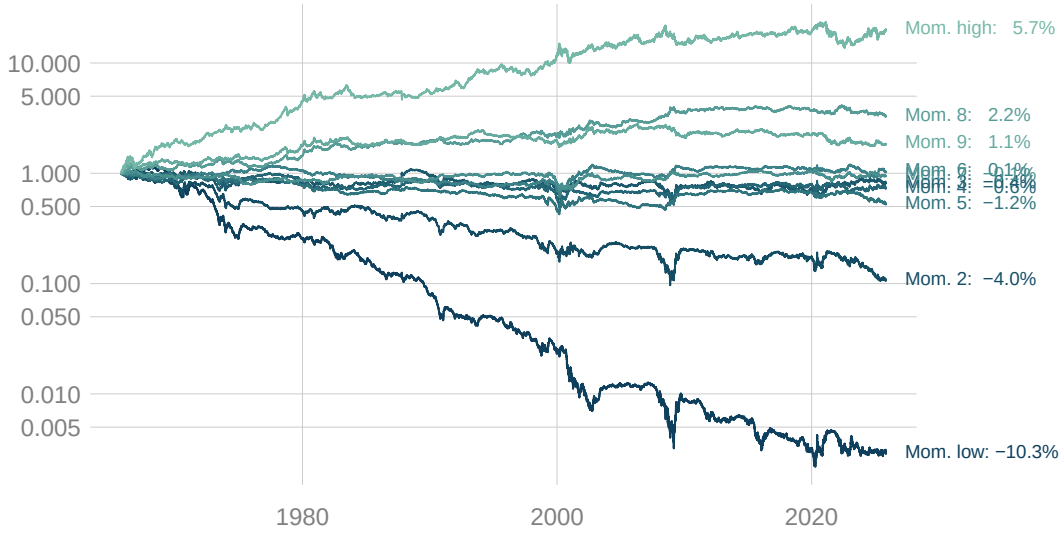
```
            dollars.arrow = " -> ",
            labels = c("Market\n(total return)"))
```



Note that the plot uses a log scale, and we have added a note showing what would have come become of one dollar invested in the market since 1965.

For relative plots [Schumann, 2013], use the `bm` ('benchmark') argument. This will lead to plot of the ratio of the series and its benchmark. An upward-sloping line indicates a higher return of the series when compared with the benchmark and vice versa; a sideways line indicates equal returns of series and benchmark. We demonstrate such a plot with time-series of momentum portfolios (decile portfolios), relative to the overall market, with all portfolio value-weighted.

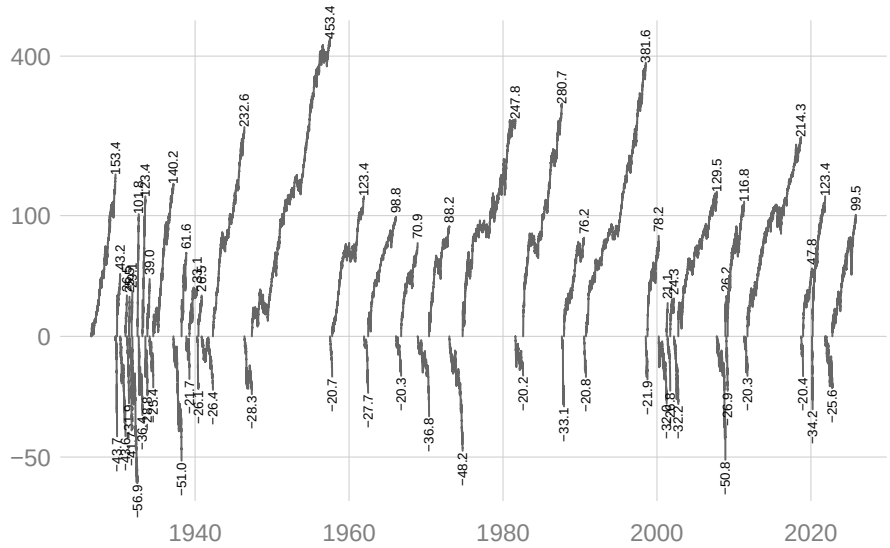
```
> series <- French("~/Downloads/French/",
                   "10_Portfolios_Prior_12_2_Daily_CSV.zip",
                   weighting = "value",
                   frequency = "monthly",
                   price.series = TRUE)
> series <- series[rownames(series) >= "1965-1-1", ]
> plotseries(series,
             bm = market[rownames(market) >= "1965-1-1", ],
             t = as.Date(row.names(series)),
             col = hcl.colors(n = ncol(series)+5,
                              palette = "Dark Mint"),
             log.scale = TRUE,
             dollars.show = FALSE,
             labels = paste("Mom.", c("low", 2:9, "high")),
             labels.col = TRUE)
```



Streaks

These charts rely on function `streaks` in package `PMwR`. It breaks a series into subperiods of uninterrupted up and down movements – streaks. Uninterrupted is meant in the sense that no countermovement of `streaks.down` (`streaks.up`) percent or more occurs in up (down) movements.

```
> plotseries(market,
             t = as.Date(row.names(market)),
             series.type = "streaks")
```

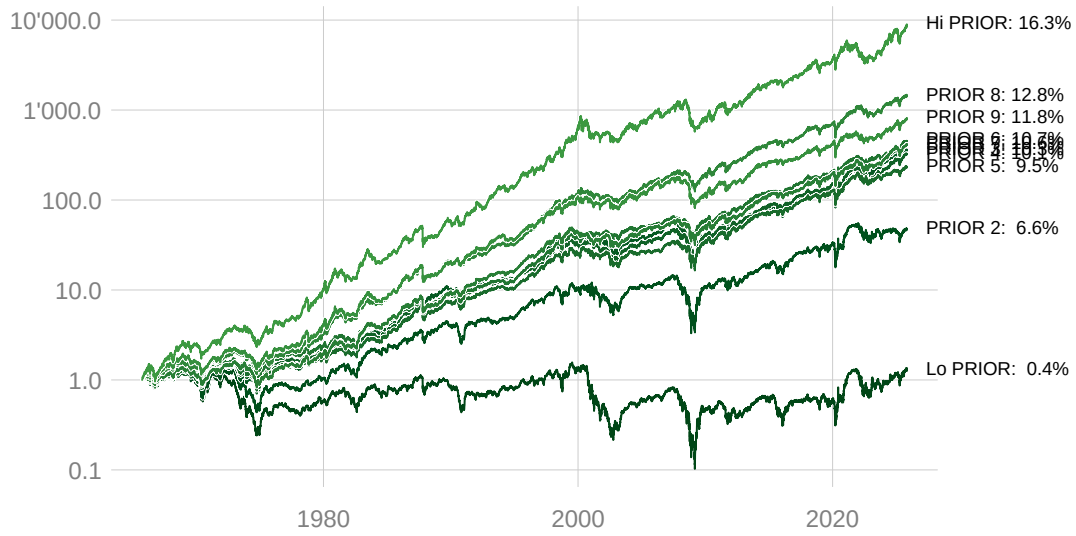


The vertical axis always uses a log scale, so that a drop of 50% takes the same vertical distance as a rise of 100%.

Sometimes it is also useful to compute relative streaks. Let us look at momentum.

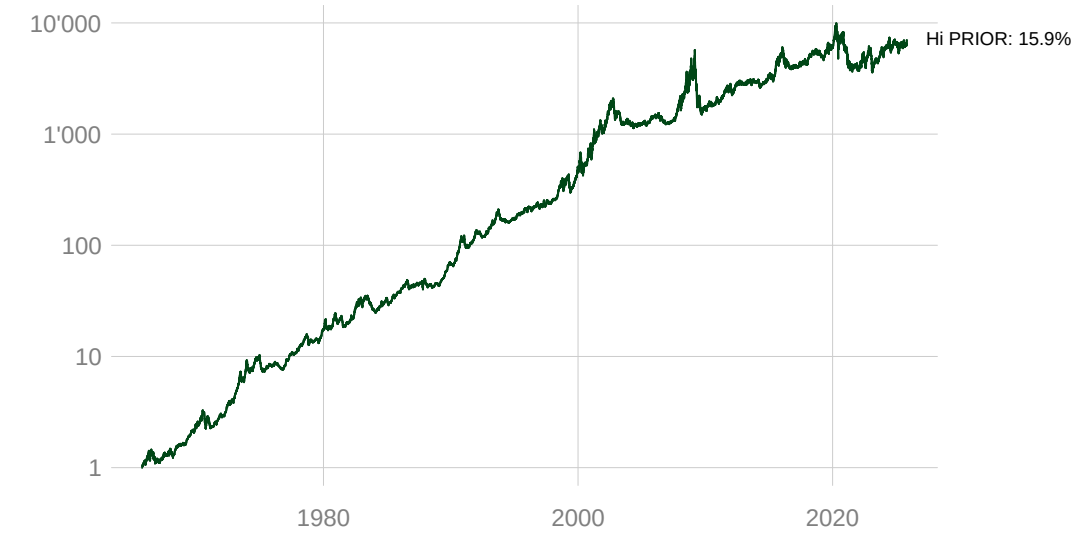
```
> library("PMwR")   ## for scale1
> series <- French("~/Downloads/French/",
                   "10_Portfolios_Prior_12_2_Daily_CSV.zip",
                   frequency = "daily",
                   price.series = TRUE)
> series <- series[rownames(series) > "1965-1-1", ]
```

```
> series <- scale1(series)
> plotseries(series, as.Date(row.names(series)),
             col = hcl.colors(n = 30, palette = "Greens"),
             log.scale = TRUE,
             labels = colnames(series),
             white.underlay = TRUE)
```



Let us plot the the decile portfolio with the highest returns (the winners) against the decile portfolio with the lowest returns (the losers).

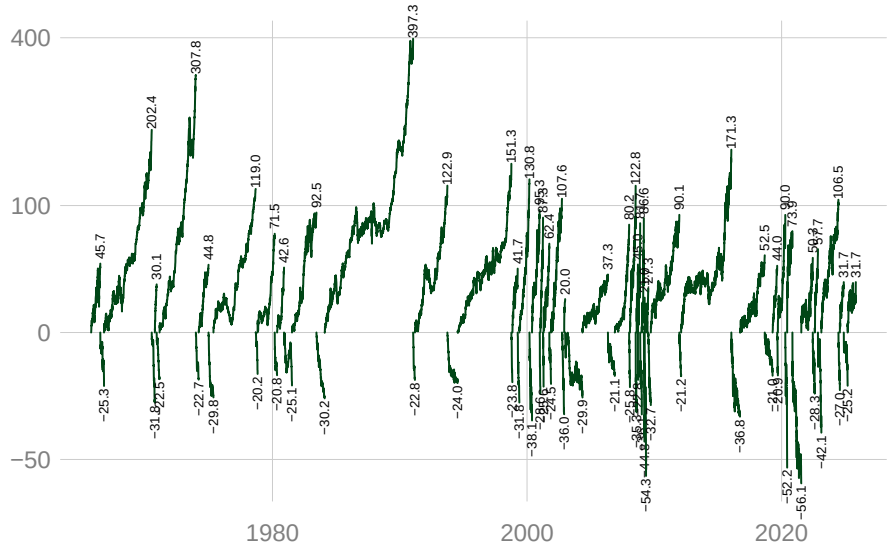
```
> plotseries(series[, 10], as.Date(row.names(series)),
             col = hcl.colors(n = 30, palette = "Greens")[1],
             log.scale = TRUE,
             labels = colnames(series)[10],
             bm = series[, 1])
```



Keep in mind that this is on a log scale: so the drawdown in the first quarter of 2009, for instance, is substantial.

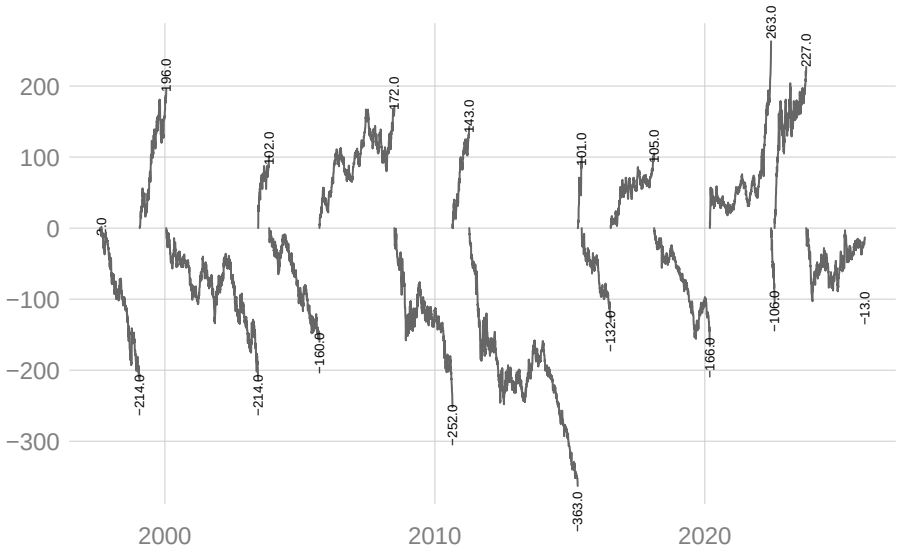
```
> plotseries(series[, 10], as.Date(row.names(series)),
             col = hcl.colors(n = 30, palette = "Greens")[1],
             log.scale = TRUE,
             labels = colnames(series)[10],
```

```
        bm = series[, 1],
        series.type = "streaks")
```



As a final example: For some time-series types, such as interest rates, it is more sensible and more common to look at differences, not relative changes. As an example, we chart the 10-year yield of German government bonds, as published by the Bundesbank. We define a streak as a yield move of 100 basis points. The key setting is `FALSE` for `streaks.relative`.

```
> library("bundesbank")
> y <- getSeries("BBSIS.D.I.ZST.ZI.EUR.S1311.B.A604.R10XX.R.A.A._Z._Z.A",
                 start = "1997-01", return.class = "zoo",
                 dest.dir = "~/Downloads/bundesbank")
> plotseries(y,
             series.type = "streaks",
             streaks.relative = FALSE,
             streaks.up = 1)
```

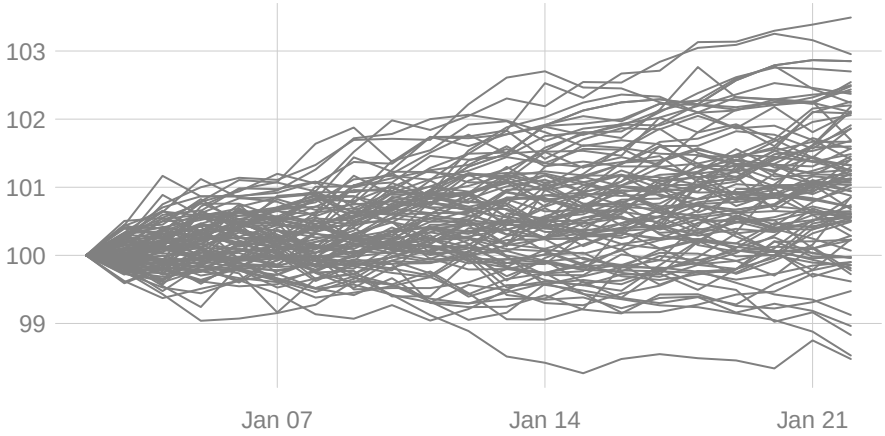


Fan-plots and quantile-plots

Fanplots build on code of Gilli et al. [2019, Chapter 15].

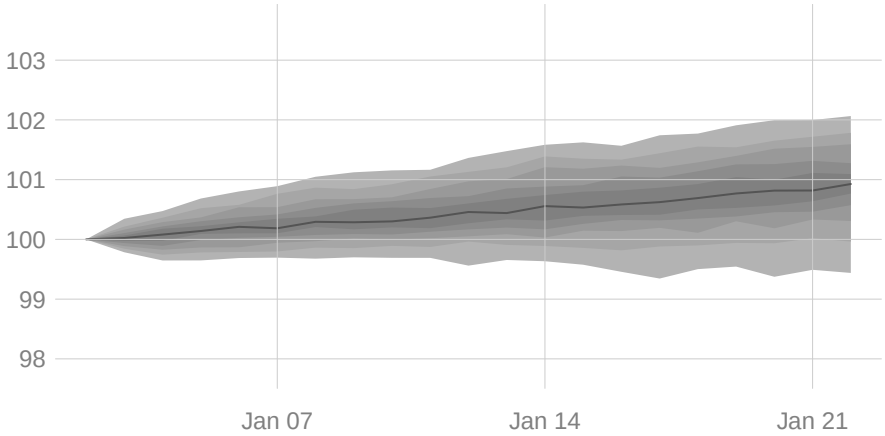
Suppose you wanted to plot many paths.

```
> x <- gbm(100, 20, r=0.01, 0.01^2,1, S0 = 100)
> t <- as.Date("2019-1-1") + seq_len(nrow(x))
> plotseries(x, t, col = rep(grey(.5), ncol(x)))
```



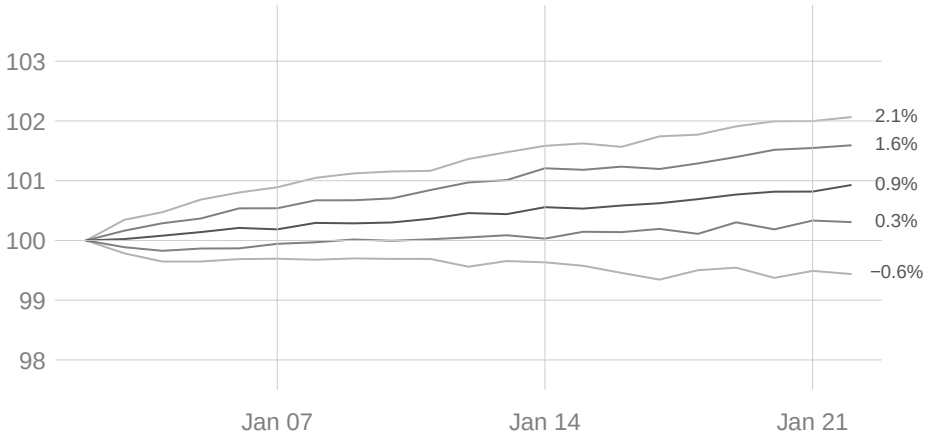
A fan-plot of the same data.

```
> plotseries(x, t, series.type = "fan")
```



A quantile-plot plots only particular quantiles.

```
> plotseries(x, t, series.type = "quantile",
             probs = c(0.1, 0.25, 0.5, 0.75, 0.9))
```



References

Eugene F. Fama and Kenneth R. French. The cross-section of expected stock returns. *Journal of Finance*, 47(2):427–465, 1992.

Manfred Gilli, Dietmar Maringer, and Enrico Schumann. *Numerical Methods and Optimization in Finance*. Elsevier/Academic Press, 2nd edition, 2019. URL `https://enricoschumann.net/NMOF`.

Enrico Schumann. Two pitfalls in comparing financial time-series. available from `https://enricoschumann.net/files/pitfalls.pdf`, 2013.