Filters

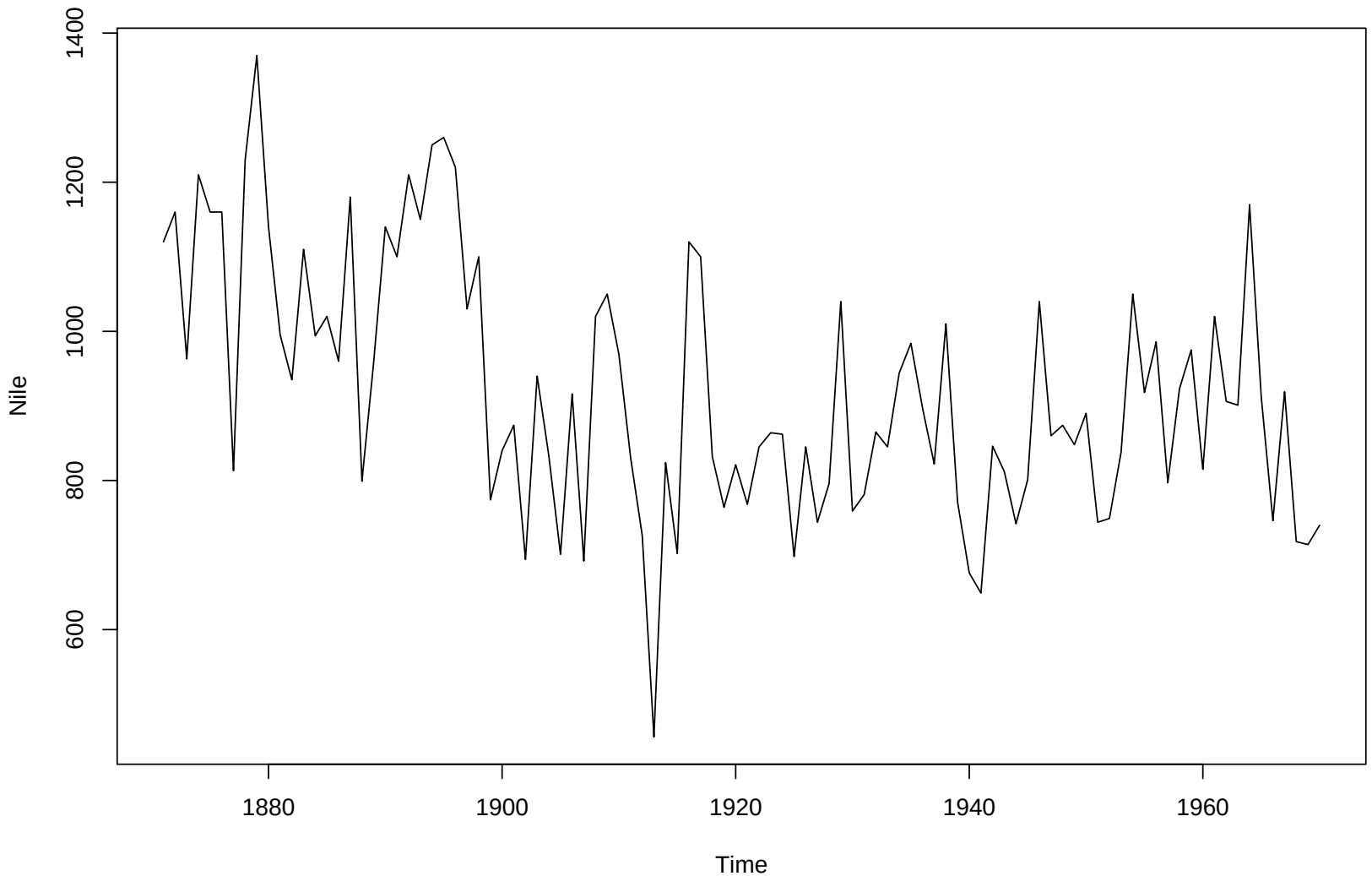
2024-10-07

Outline

Data

Flow of the river Nule

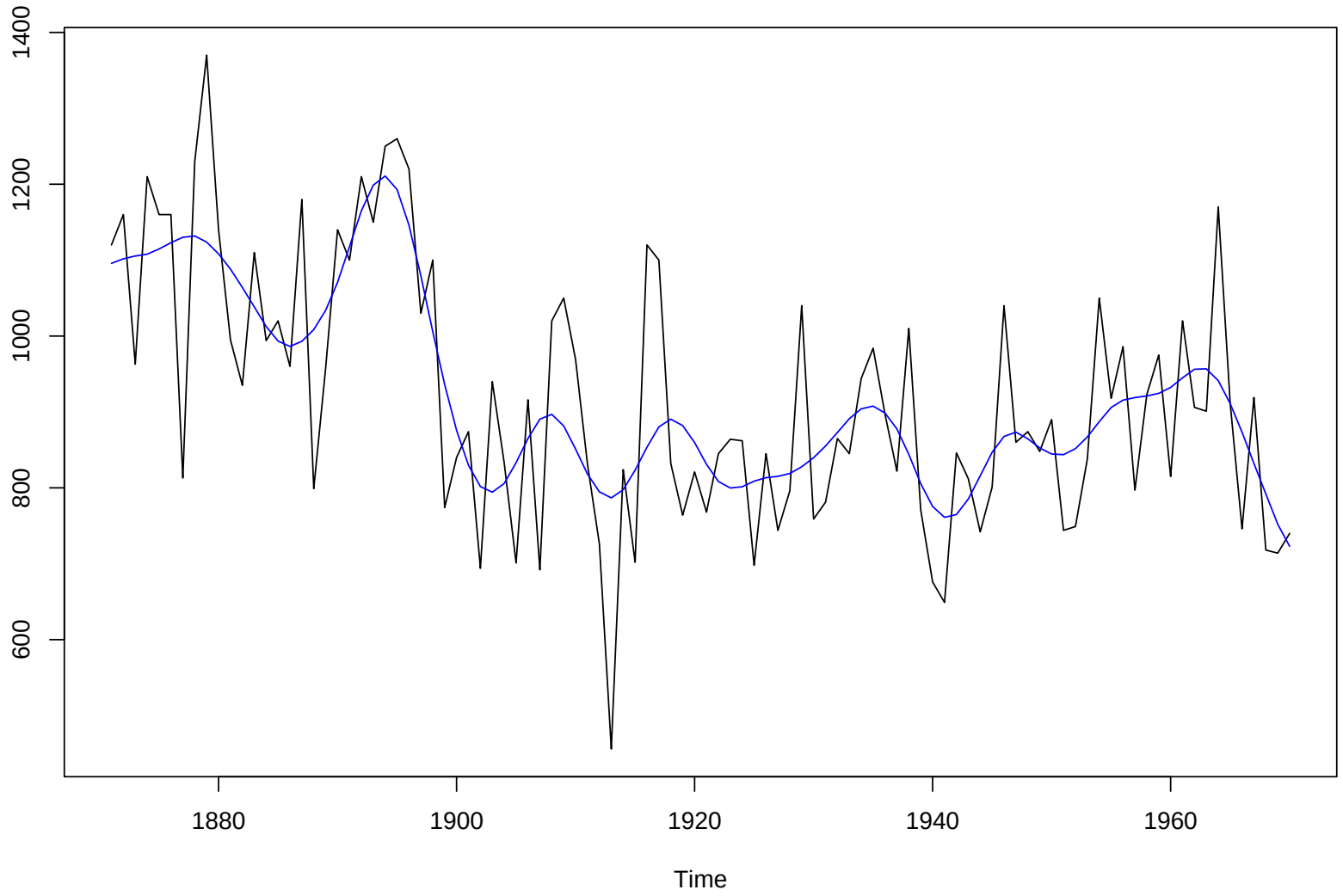
```
plot(Nile)
```



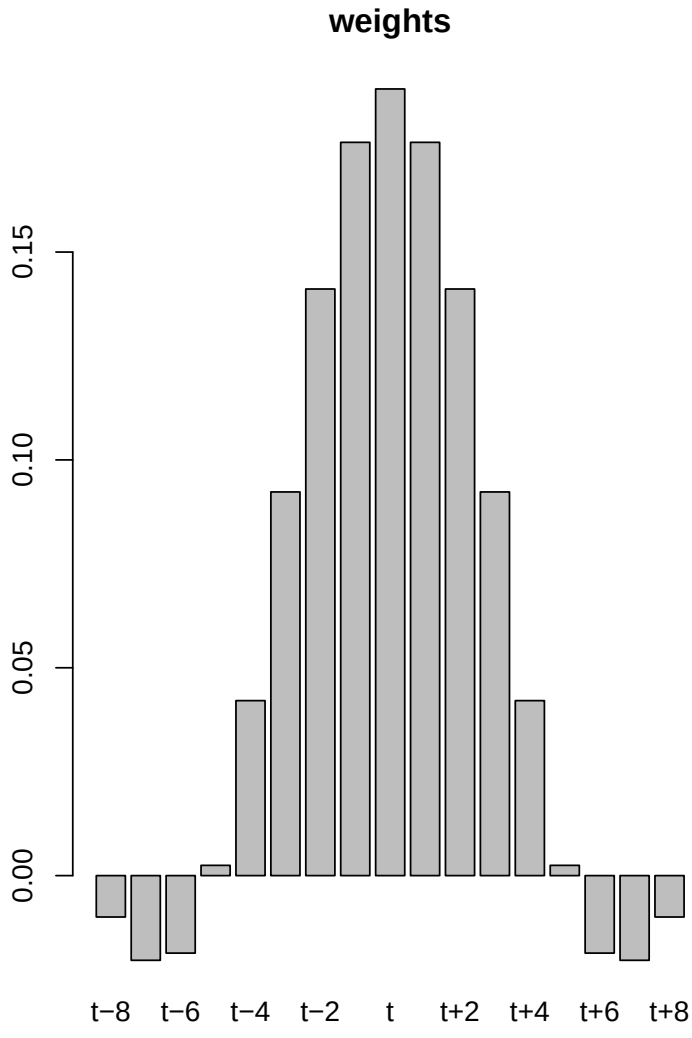
Pre-specified filters

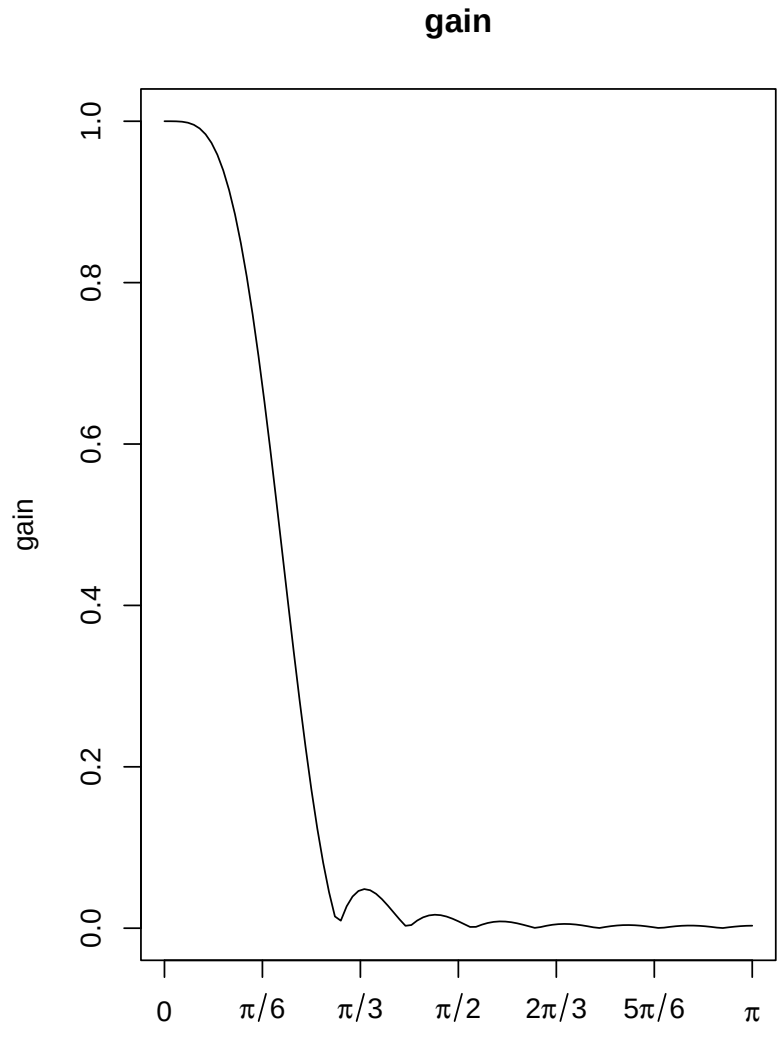
Filters based on well-defined mathematical properties

Henderson filters



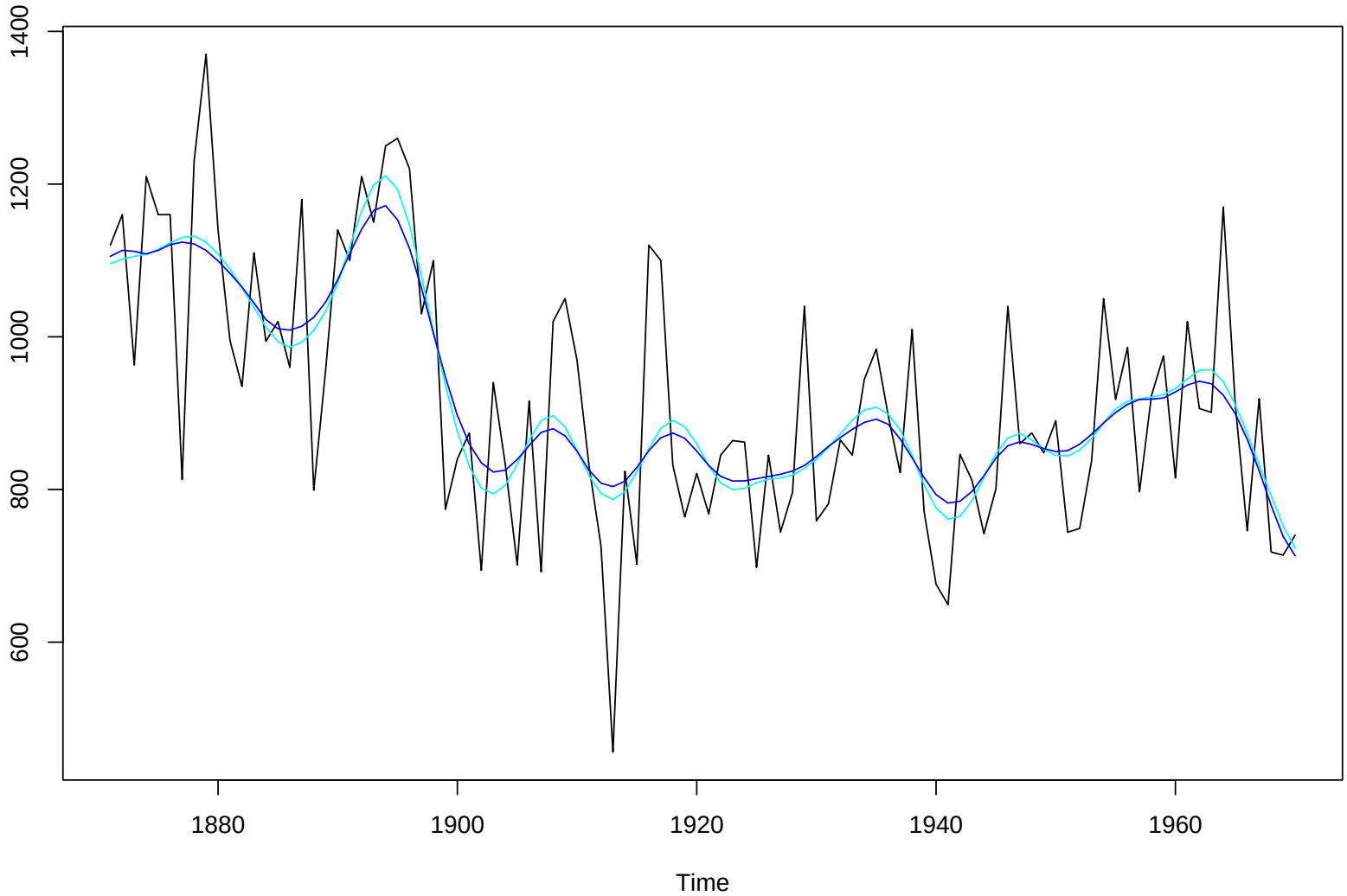
Henderson filter properties



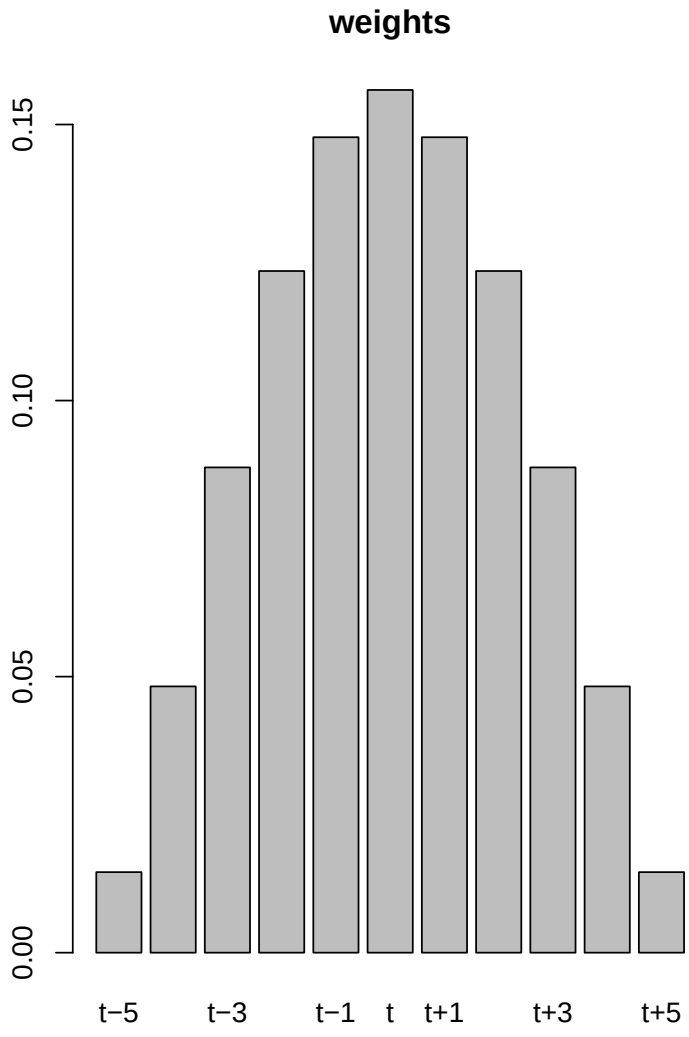


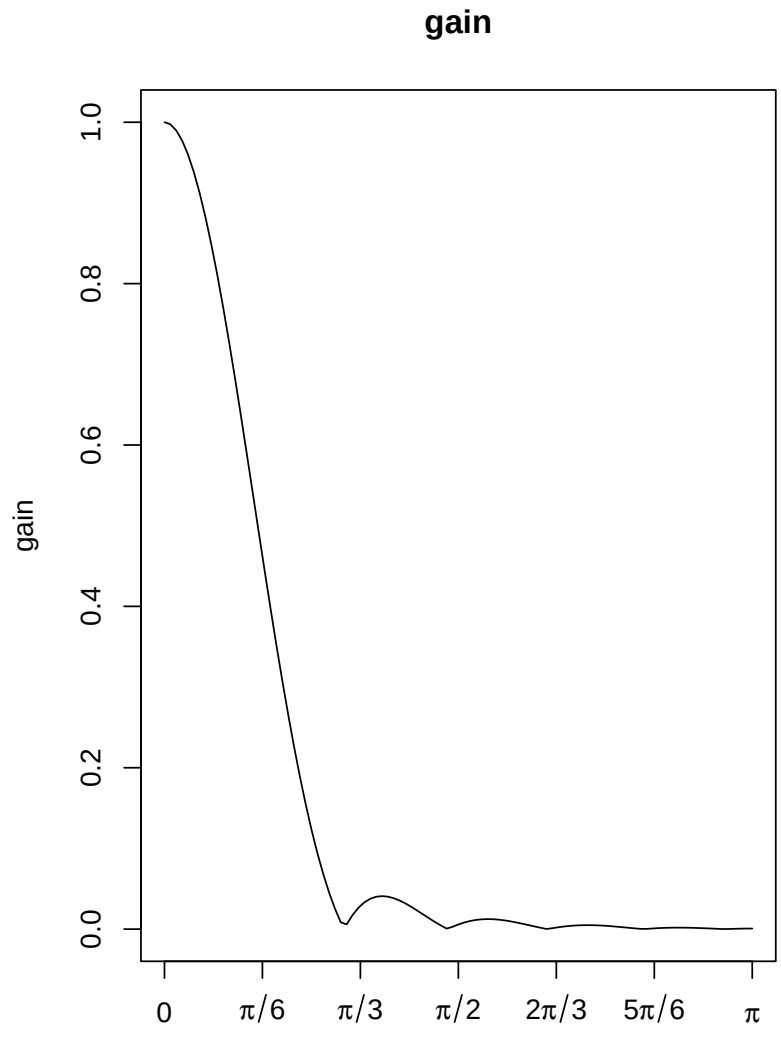
Local polynomials

Regression with a linear trend on 11 periods, using a bi-weight kernel $\alpha(1 - x^2)^2)$



Loess filter properties





Model-based filters (I)

Structural model (fixed parameters)

$$
\begin{aligned}
y(t) &= T(t) + N(t) \\
\Delta^2 T(t) &= \epsilon_T(t) && \epsilon_T \sim N(0,1) \\
N(t) &= \epsilon_N(t) && \epsilon_N \sim N(0,\lambda)
\end{aligned}
$$

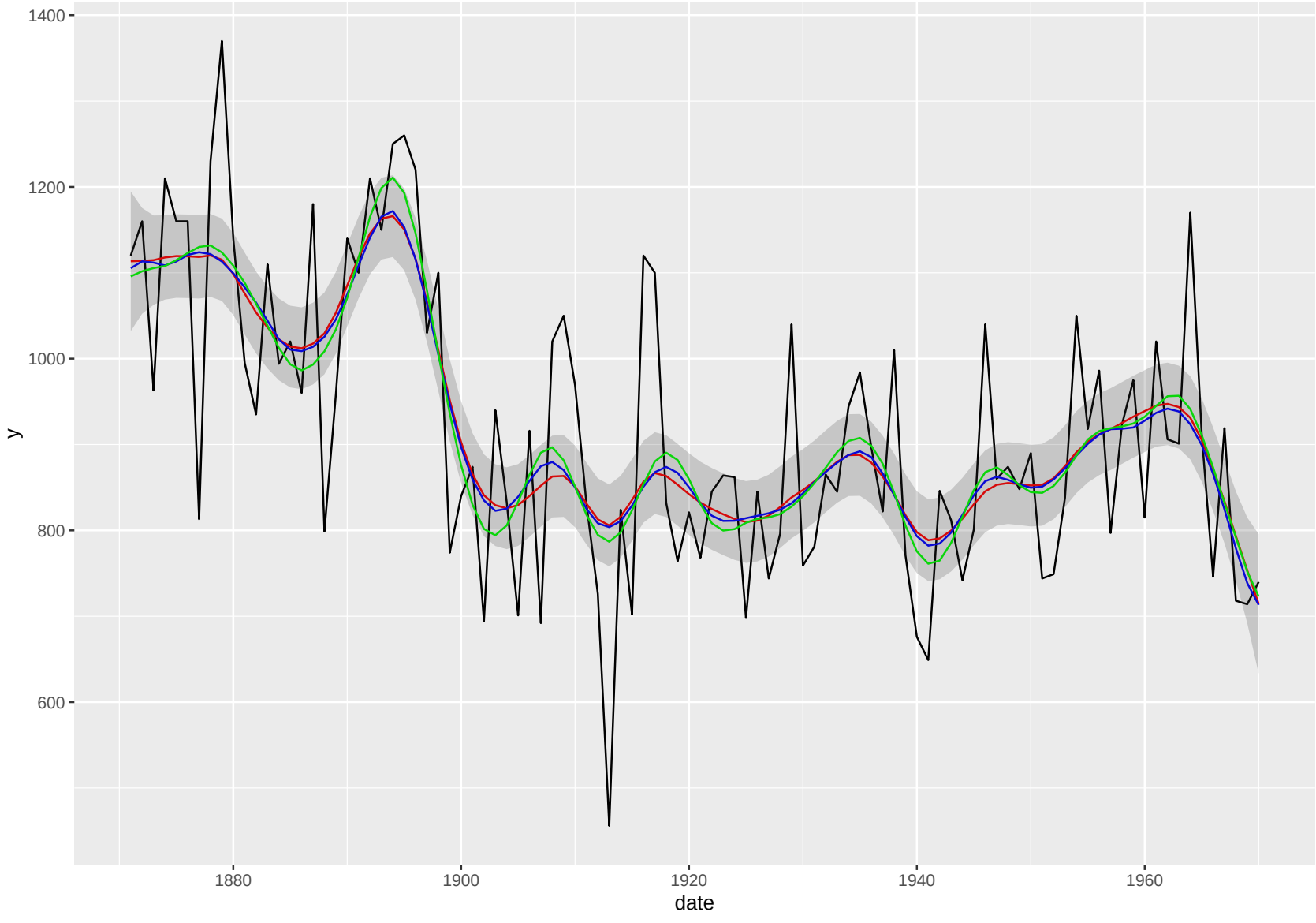
```
hp_ucm<-function(lambda=25){
  i2<-rjd3toolkit::arima_model("trend", delta = c(1,-2, 1))
  noise<-rjd3toolkit::arima_model(variance=lambda)
  ucm<-rjd3toolkit::ucarima_model(components=list(i2, noise))
}

ucm_estimate<-function(x, ucm, stdev=TRUE){
  jucm<-rjd3toolkit:::.r2jd_ucarima(ucm)
  jcmps<-rJava::.jcall("jdplus/toolkit/base/r/arima/UcarimaModels",
            "Ljdplus/toolkit/base/api/math/matrices/Matrix;", "estimate",
            as.numeric(x), jucm, as.logical(stdev))
  return(rjd3toolkit:::.jd2r_matrix(jcmps))
}

hp_estimate<-function(s, lambda=25){
  ucm<-hp_ucm(lambda = lambda)
  rslt<-ucm_estimate(s, ucm)
}
```

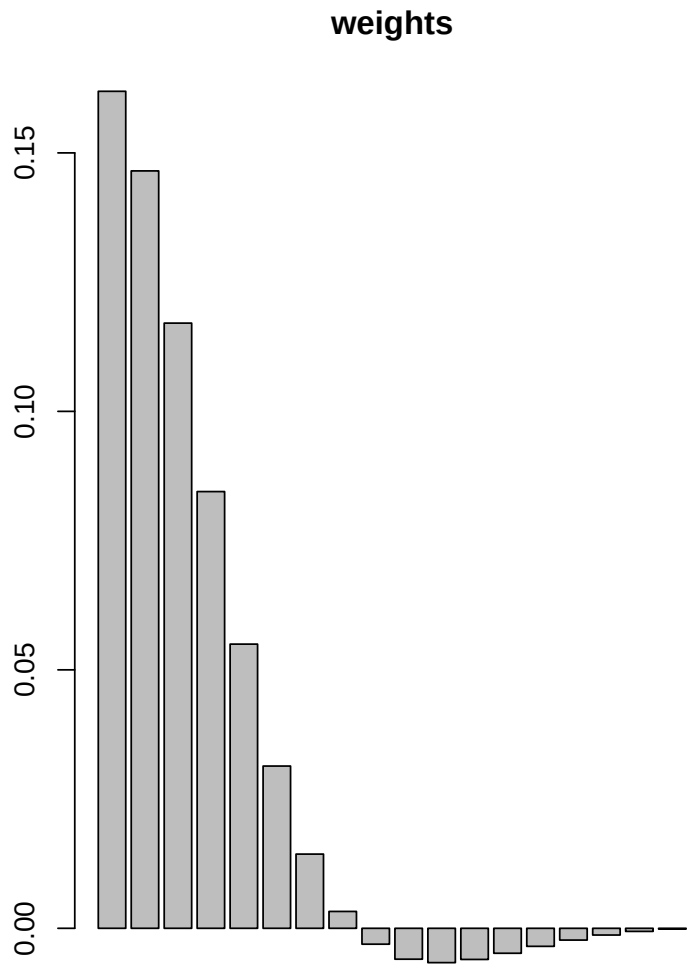
Model-based filters (I)

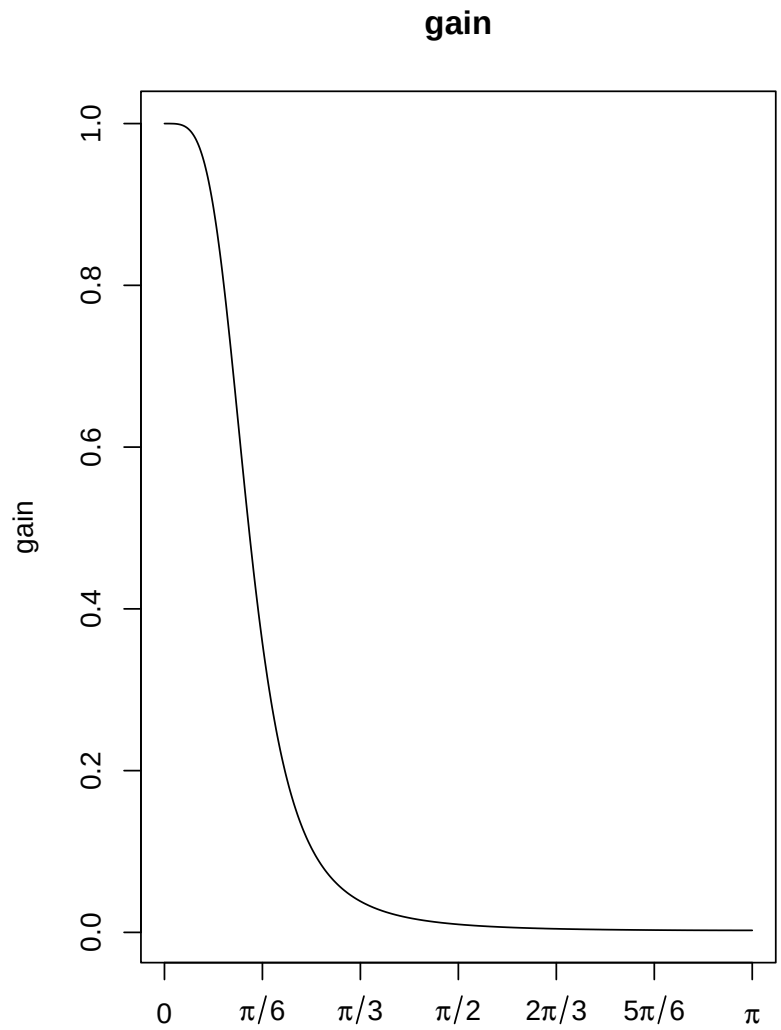
Structural model (fixed parameters)



Model-based filters properties

Structural model (fixed parameters)

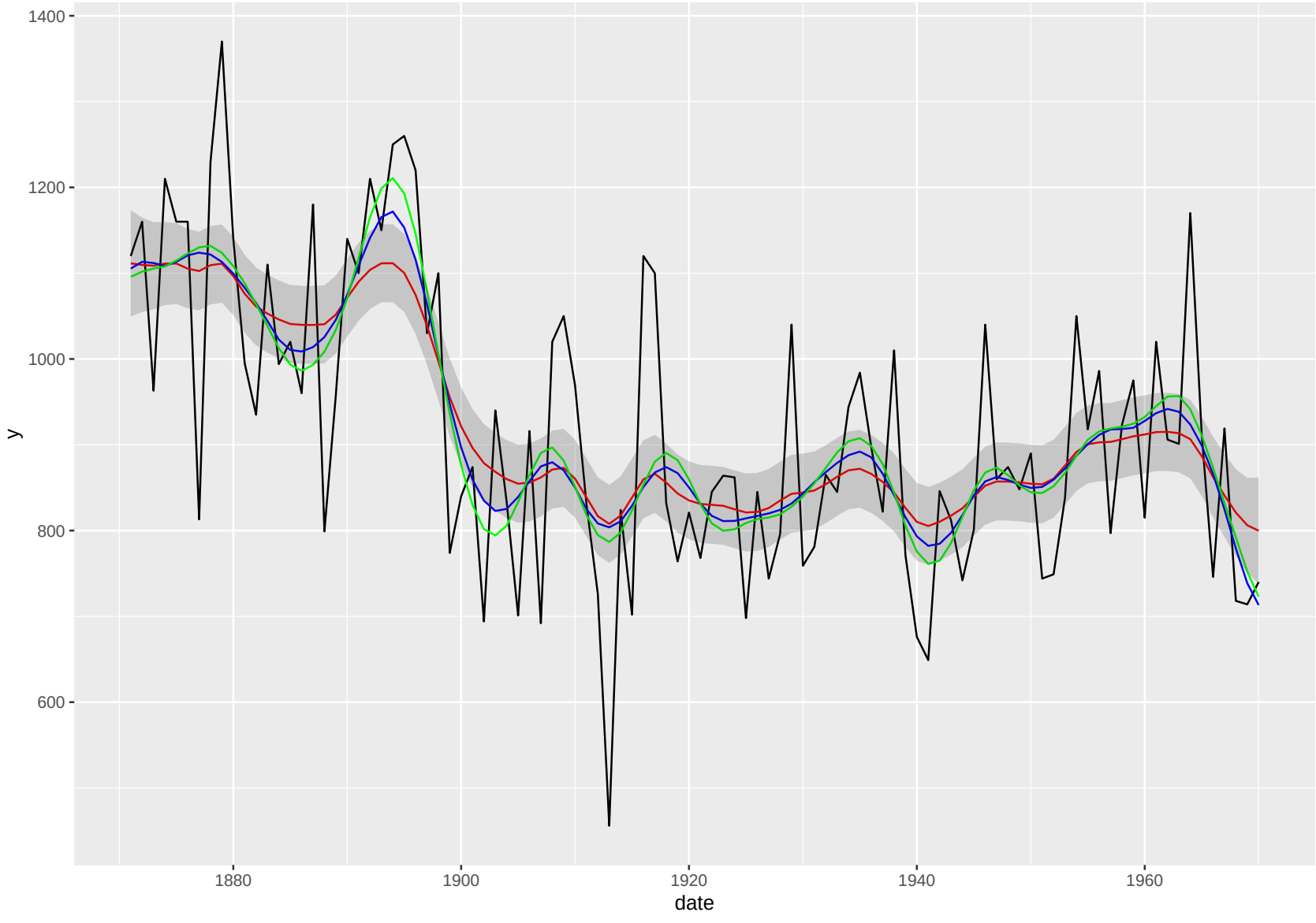




Model-based filters properties

Model-based filters

Canonical decomposition



Model-based filters properties

Canonical decomposition