STATS 207 Project RMD

```
library(astsa)
library(Metrics)
library(xts)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: ’zoo’
```

```
## The following objects are masked from ’package:base’:
##
##     as.Date, as.Date.numeric
```

```
library(dlm)
library(ggplot2)
```

```
##
## Attaching package: ’ggplot2’
```

```
## The following object is masked from ’package:dlm’:
##
##     %+%
```

```
library(TSstudio)

## Import data and preprocess data
## We use the closing data here
## For Kalman filtering specifically, use monthly close intervals rather than daily ones
## Test sample over the most recent six months
full_ark <- read.csv(’ARKGMONTH.csv’)
fulla_ts <- as.ts(full_ark$Close, full_ark$Date)

full_qqq <- read.csv(’QQQMONTH.csv’)
fullq_ts <- as.ts(full_qqq$Close, full_qqq$Date)

full_schf <- read.csv(’SCHFMONTH.csv’)
fulls_ts <- as.ts(full_schf$Close, full_schf$Date)

full_vt <- read.csv(’VTMONTH.csv’)
fullv_ts <- as.ts(full_vt$Close, full_vt$Date)

full_xlf <- read.csv(’XLFMONTH.csv’)
fullx_ts <- as.ts(full_xlf$Close, full_xlf$Date)
```

```
## Kalman Filtering for ARKG Data
split_df <- ts_split(ts.obj = fulla_ts, sample.out = 6)

training = split_df$train
```

```r
model <- function(p) {
  return(dlmModPoly(2, dV = p[1], dW = p[2:3]) + dlmModSeas(12, dV = p[4]))
}

## Parameter Estimation
mle <- dlmMLE(training, parm = c(0.1,0.001,1,1), build = model)
if (mle$convergence == 0) print (’converge’) else print(’nonconverge’)
```

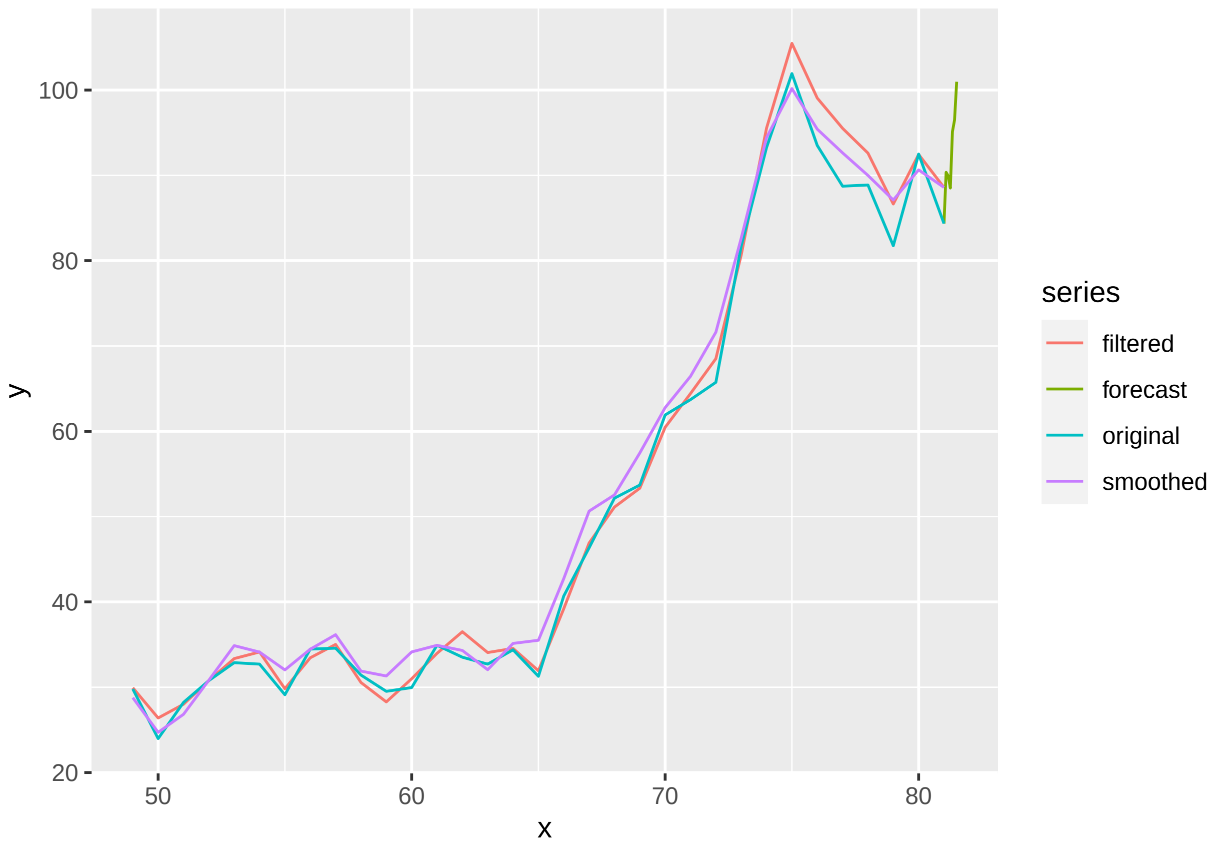
```
## [1] "converge"
```

```r
mle$par
```

```
## [1] -0.45000007 12.50348787 -0.08802203  0.44999993
```

```r
modelfit = model(mle$par)

## Filtering and Smoothing
modelfilter <- dlmFilter(training, modelfit)
modelsmoothed <- dlmSmooth(training, modelfit)
```

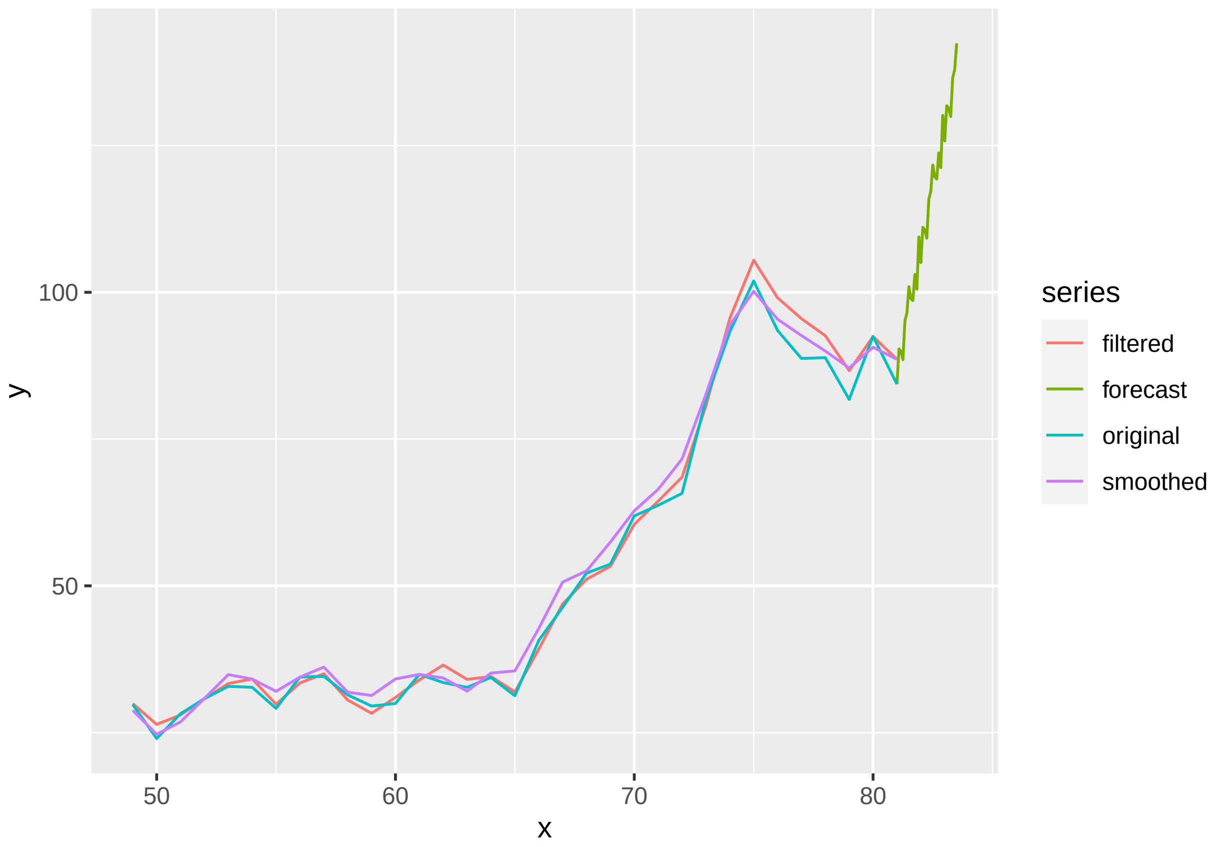
```r
n <- 1*6
a6_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a6_forecast$a[,1]* (-1)
aa <- cbind(a6_forecast$a[,1], aa)
a <- drop(a6_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
data_zoo <- as.zoo(fulla_ts)
actual <- as.vector(data_zoo[(length(data_zoo)-5):length(data_zoo)])
predicted <- as.vector(a6_forecast$f)
sqrt(mse(actual, predicted))
```

```
## [1] 28.00608
```

```
n <- 1*30
a30_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a30_forecast$a[,1]* (-1)
aa <- cbind(a30_forecast$a[,1], aa)
a <- drop(a30_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
## Kalman Filtering for ARKG Data
split_df <- ts_split(ts.obj = fullq_ts, sample.out = 6)

training = split_df$train

model <- function(p) {
  return(dlmModPoly(2, dV = p[1], dW = p[2:3]) + dlmModSeas(12, dV = p[4]))
}

## Parameter Estimation
mle <- dlmMLE(training, parm = c(0.1,0.001,1,1), build = model)
if (mle$convergence == 0) print ('converge') else print('nonconverge')
```

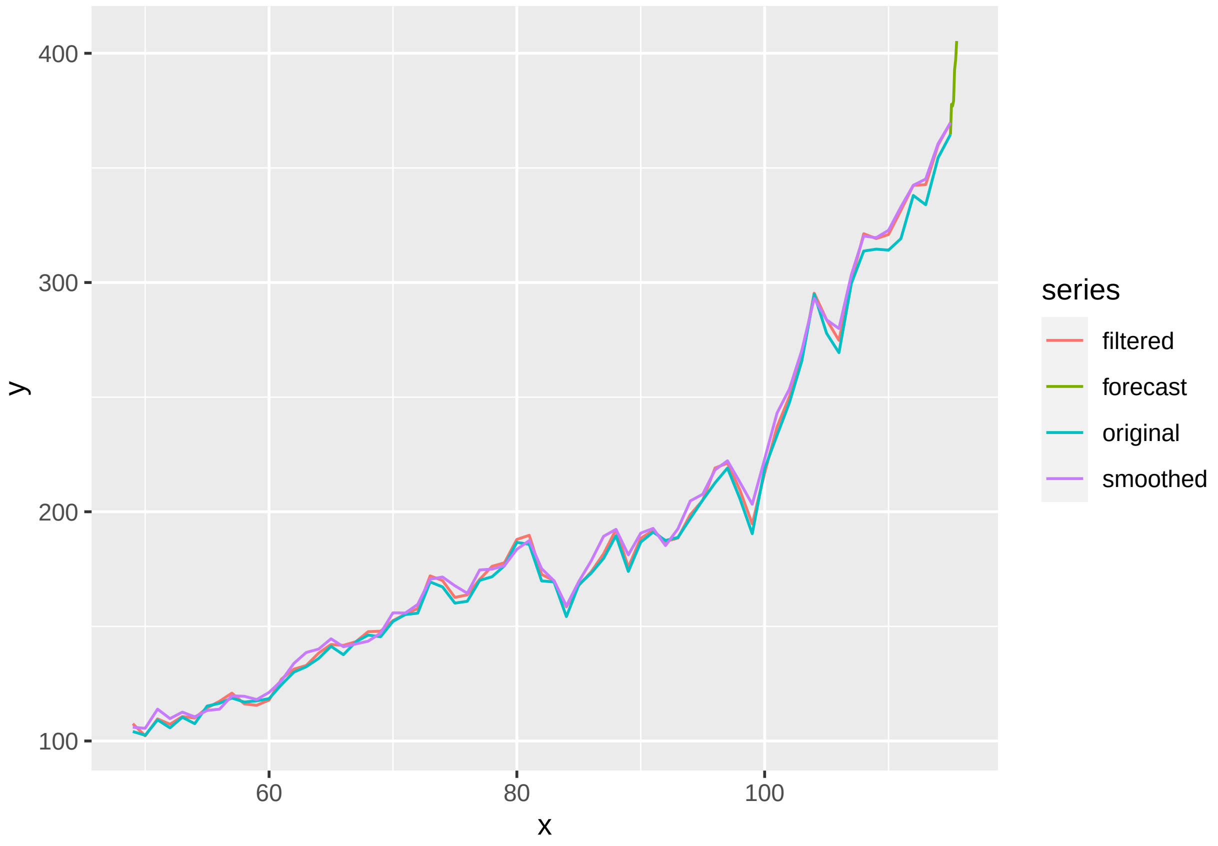
```
## [1] "converge"
```

```
mle$par
```

```
## [1] -0.4499949 49.8078331 -0.2252524  0.4500051
```

```
modelfit = model(mle$par)

## Filtering and Smoothing
modelfilter <- dlmFilter(training, modelfit)
modelsmoothed <- dlmSmooth(training, modelfit)
```

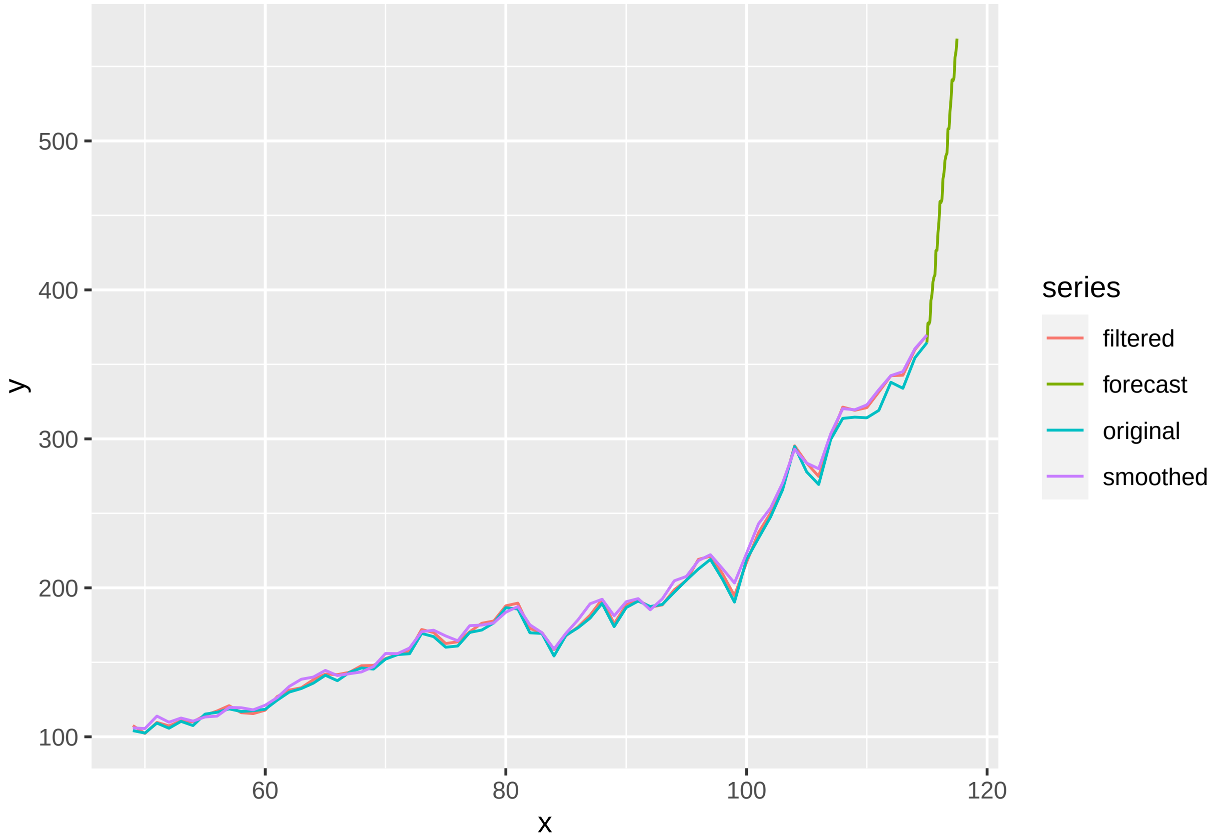
```
n <- 1*6
a6_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a6_forecast$a[,1]* (-1)
aa <- cbind(a6_forecast$a[,1], aa)
a <- drop(a6_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
data_zoo <- as.zoo(fullq_ts)
actual <- as.vector(data_zoo[(length(data_zoo)-5):length(data_zoo)])
predicted <- as.vector(a6_forecast$f)
sqrt(mse(actual, predicted))
```

```
## [1] 13.53402
```

```
n <- 1*30
a30_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a30_forecast$a[,1]* (-1)
aa <- cbind(a30_forecast$a[,1], aa)
a <- drop(a30_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
## Kalman Filtering for ARKG Data
split_df <- ts_split(ts.obj = fulla_ts, sample.out = 6)

training = split_df$train

model <- function(p) {
```

```r
  return(dlmModPoly(2, dV = p[1], dW = p[2:3]) + dlmModSeas(12, dV = p[4]))
}

## Parameter Estimation
mle <- dlmMLE(training, parm = c(0.1,0.001,1,1), build = model)
if (mle$convergence == 0) print (’converge’) else print(’nonconverge’)
```

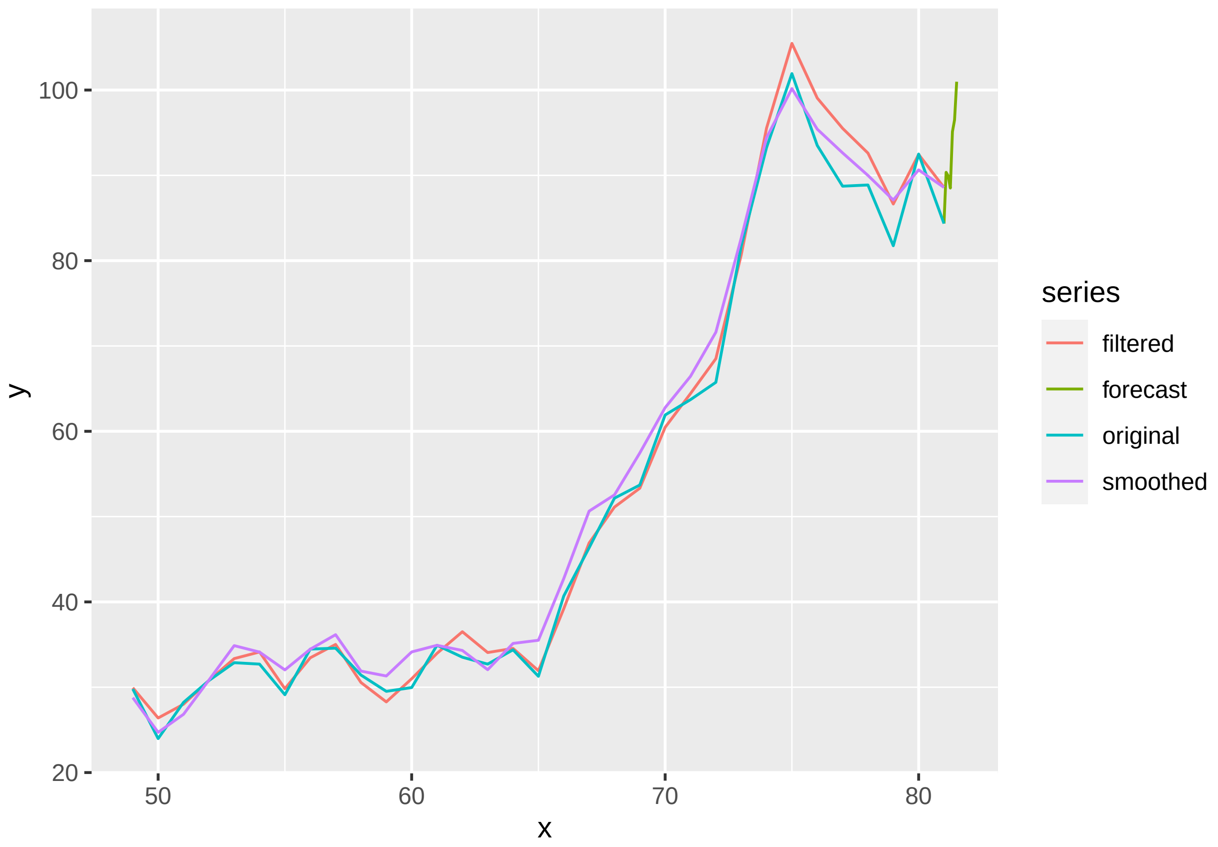
```
## [1] "converge"
```

```r
mle$par
```

```
## [1] -0.45000007 12.50348787 -0.08802203  0.44999993
```

```r
modelfit = model(mle$par)

## Filtering and Smoothing
modelfilter <- dlmFilter(training, modelfit)
modelsmoothed <- dlmSmooth(training, modelfit)
```

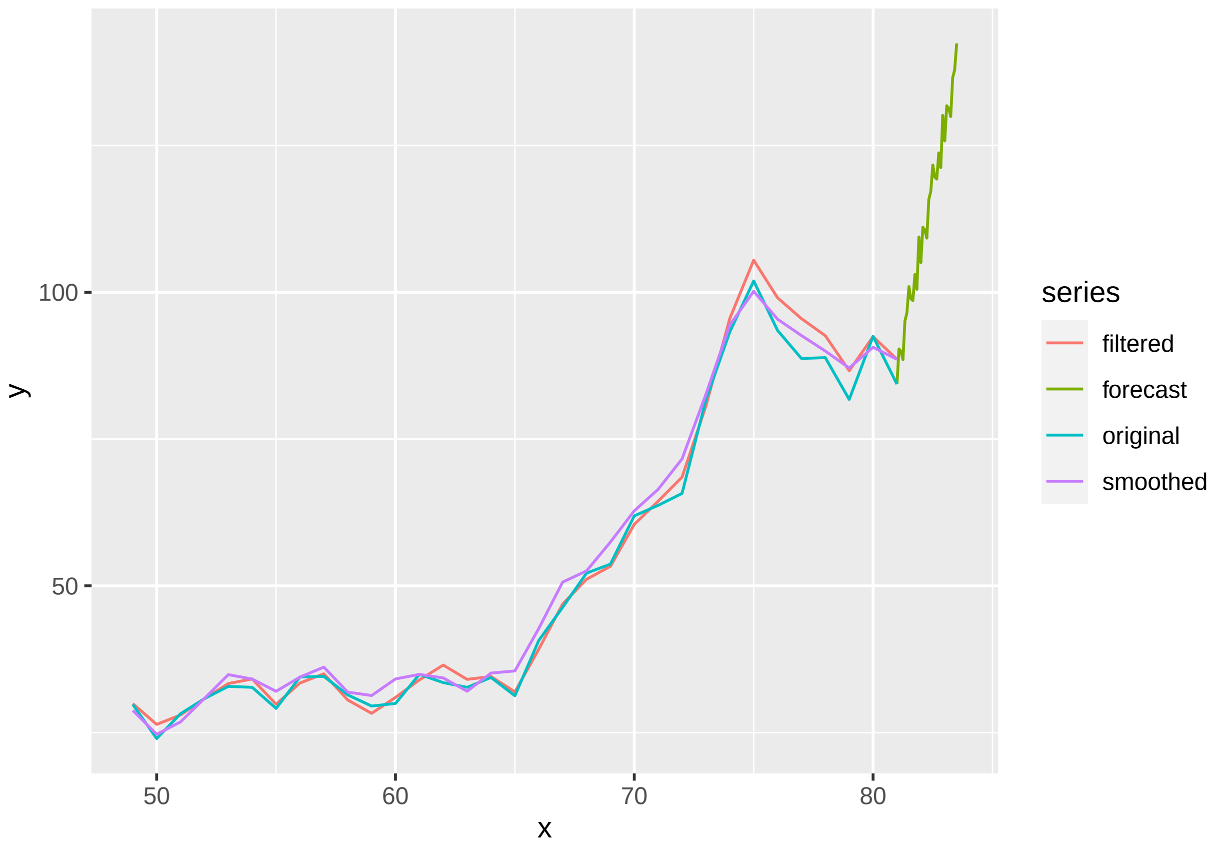
```r
n <- 1*6
a6_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a6_forecast$a[,1]* (-1)
aa <- cbind(a6_forecast$a[,1], aa)
a <- drop(a6_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
data_zoo <- as.zoo(fulla_ts)
actual <- as.vector(data_zoo[(length(data_zoo)-5):length(data_zoo)])
predicted <- as.vector(a6_forecast$f)
sqrt(mse(actual, predicted))
```

```
## [1] 28.00608
```

```
n <- 1*30
a30_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a30_forecast$a[,1]* (-1)
aa <- cbind(a30_forecast$a[,1], aa)
a <- drop(a30_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
a30_forecast$f
```

```
## Time Series:
## Start = 82
## End = 111
## Frequency = 1
##         Series 1
##  [1,]  90.35795
##  [2,]  89.92575
##  [3,]  88.51091
##  [4,]  95.12237
##  [5,]  96.48860
##  [6,] 100.97376
##  [7,]  98.92405
##  [8,]  98.57546
##  [9,] 103.03829
## [10,] 100.50961
## [11,] 109.42942
## [12,] 105.06333
## [13,] 111.07127
## [14,] 110.63907
## [15,] 109.22423
## [16,] 115.83570
## [17,] 117.20192
## [18,] 121.68708
## [19,] 119.63738
## [20,] 119.28878
## [21,] 123.75161
## [22,] 121.22294
## [23,] 130.14275
## [24,] 125.77665
## [25,] 131.78460
## [26,] 131.35240
## [27,] 129.93756
## [28,] 136.54903
## [29,] 137.91525
## [30,] 142.40041
```

```
## Kalman Filtering for QQQ Data
split_df <- ts_split(ts.obj = fullq_ts, sample.out = 6)

training = split_df$train

model <- function(p) {
  return(dlmModPoly(2, dV = p[1], dW = p[2:3]) + dlmModSeas(12, dV = p[4]))
}

## Parameter Estimation
mle <- dlmMLE(training, parm = c(0.1,0.001,1,1), build = model)
if (mle$convergence == 0) print (’converge’) else print(’nonconverge’)
```

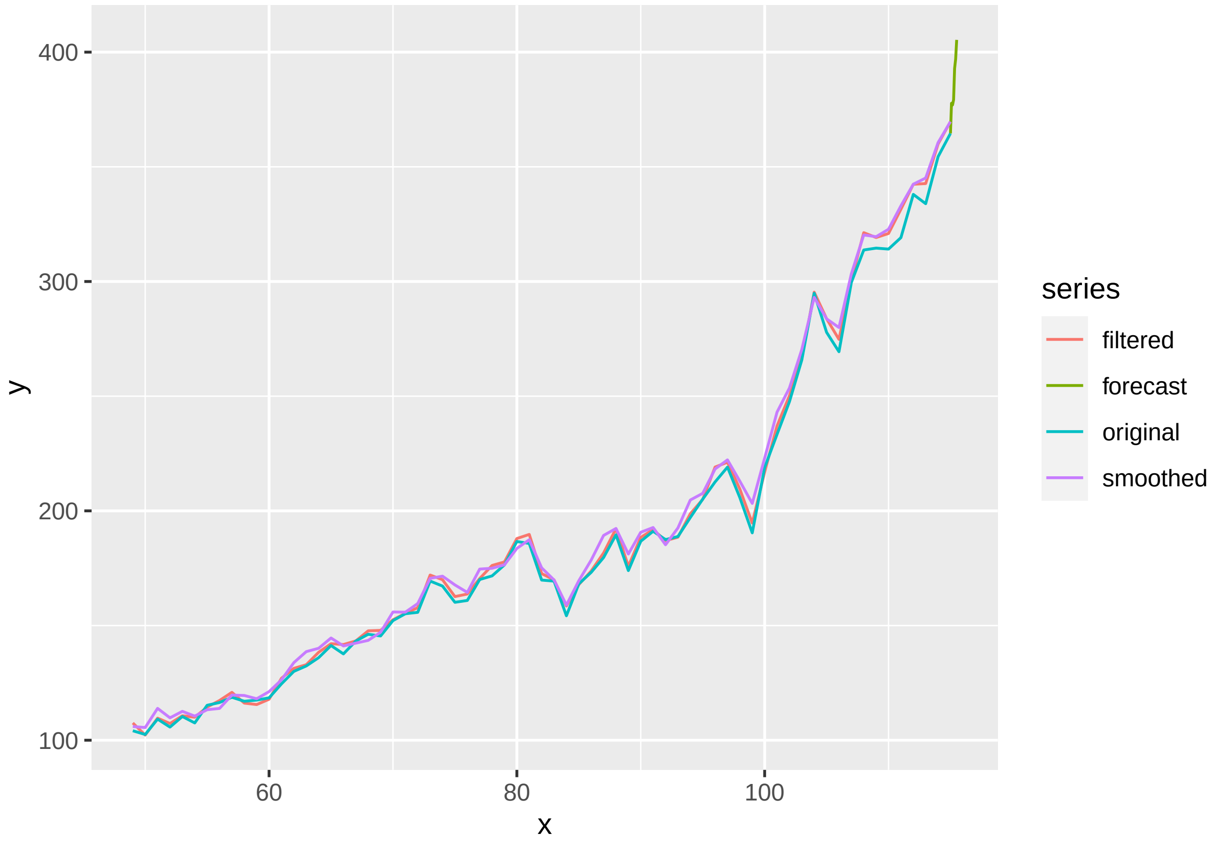
```
## [1] "converge"
```

```
mle$par
```

```
## [1] -0.4499949 49.8078331 -0.2252524  0.4500051
```

```
modelfit = model(mle$par)

## Filtering and Smoothing
modelfilter <- dlmFilter(training, modelfit)
modelsmoothed <- dlmSmooth(training, modelfit)
```

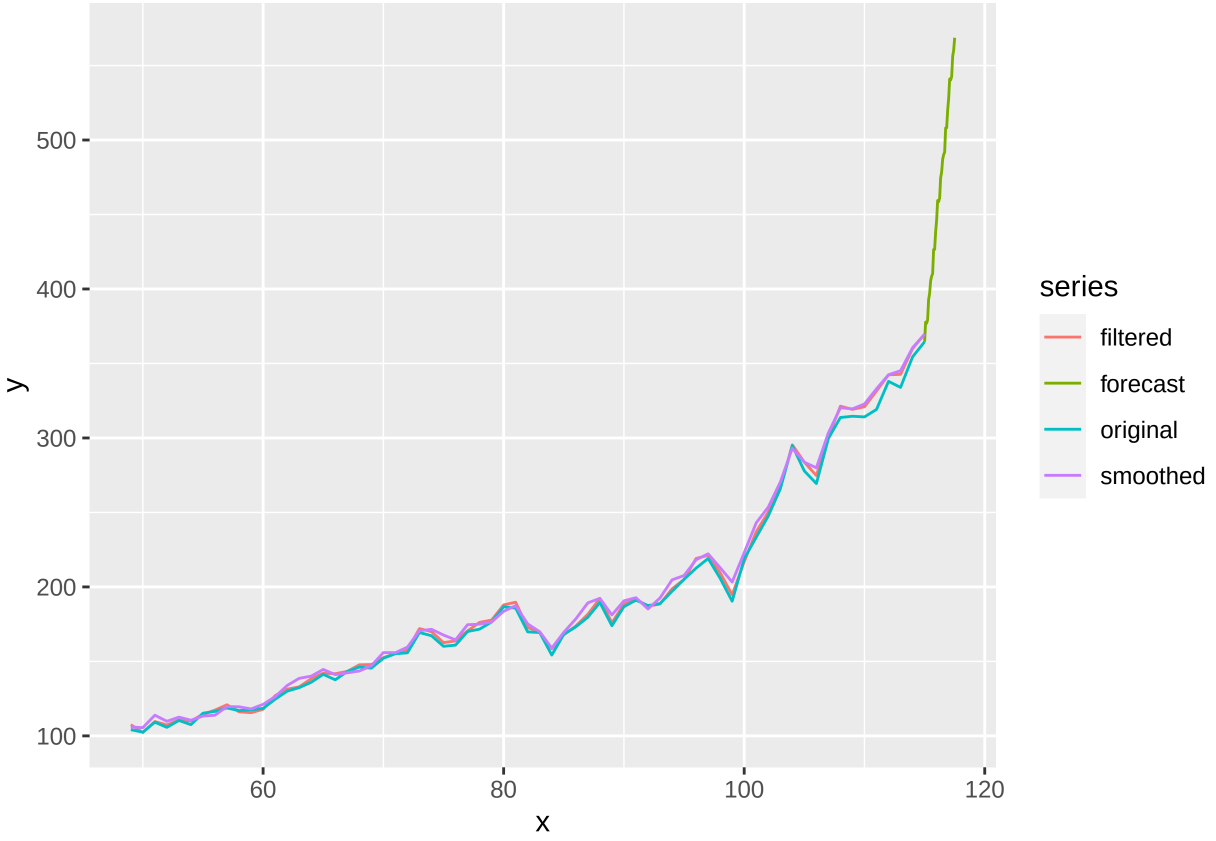
```
n <- 1*6
a6_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a6_forecast$a[,1]* (-1)
aa <- cbind(a6_forecast$a[,1], aa)
a <- drop(a6_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
data_zoo <- as.zoo(fullq_ts)
actual <- as.vector(data_zoo[(length(data_zoo)-5):length(data_zoo)])
predicted <- as.vector(a6_forecast$f)
sqrt(mse(actual, predicted))
```

```
## [1] 13.53402
```

```
n <- 1*30
a30_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a30_forecast$a[,1]* (-1)
aa <- cbind(a30_forecast$a[,1], aa)
a <- drop(a30_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
a30_forecast$f
```

```
## Time Series:
## Start = 116
## End = 145
## Frequency = 1
##        Series 1
##  [1,] 377.7297
##  [2,] 376.9825
##  [3,] 379.2573
##  [4,] 392.9051
##  [5,] 396.9421
##  [6,] 405.3486
##  [7,] 408.5625
##  [8,] 410.1801
##  [9,] 426.4433
## [10,] 426.5356
## [11,] 438.4147
## [12,] 446.2247
## [13,] 459.3844
## [14,] 458.6372
## [15,] 460.9120
## [16,] 474.5598
## [17,] 478.5968
## [18,] 487.0033
## [19,] 490.2172
## [20,] 491.8348
## [21,] 508.0979
## [22,] 508.1903
## [23,] 520.0694
## [24,] 527.8794
## [25,] 541.0391
## [26,] 540.2919
## [27,] 542.5667
## [28,] 556.2145
## [29,] 560.2515
## [30,] 568.6580
```

```
## Kalman Filtering for SCHF Data
split_df <- ts_split(ts.obj = fulls_ts, sample.out = 6)

training = split_df$train

model <- function(p) {
  return(dlmModPoly(2, dV = p[1], dW = p[2:3]) + dlmModSeas(12, dV = p[4]))
}

## Parameter Estimation
mle <- dlmMLE(training, parm = c(0.1,0.001,1,1), build = model)
if (mle$convergence == 0) print ('converge') else print('nonconverge')
```

```
## [1] "nonconverge"
```

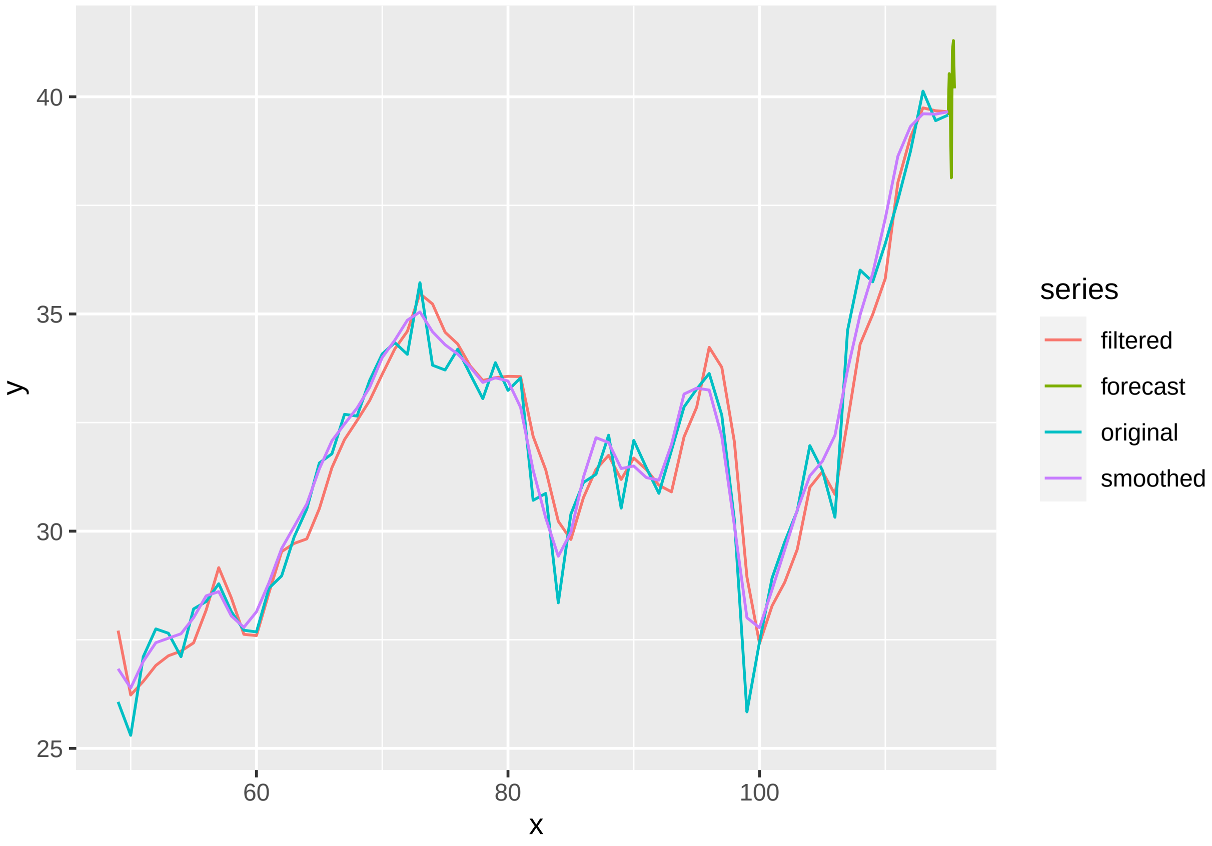
```
mle$par
```

```
## [1] -4.500005e-01 -1.111263e+00  5.309822e-09  4.499995e-01
```

```
modelfit = model(mle$par)

## Filtering and Smoothing
modelfilter <- dlmFilter(training, modelfit)
modelsmoothed <- dlmSmooth(training, modelfit)
```

```
n <- 1*6
a6_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a6_forecast$a[,1]* (-1)
aa <- cbind(a6_forecast$a[,1], aa)
```

```
a <- drop(a6_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```

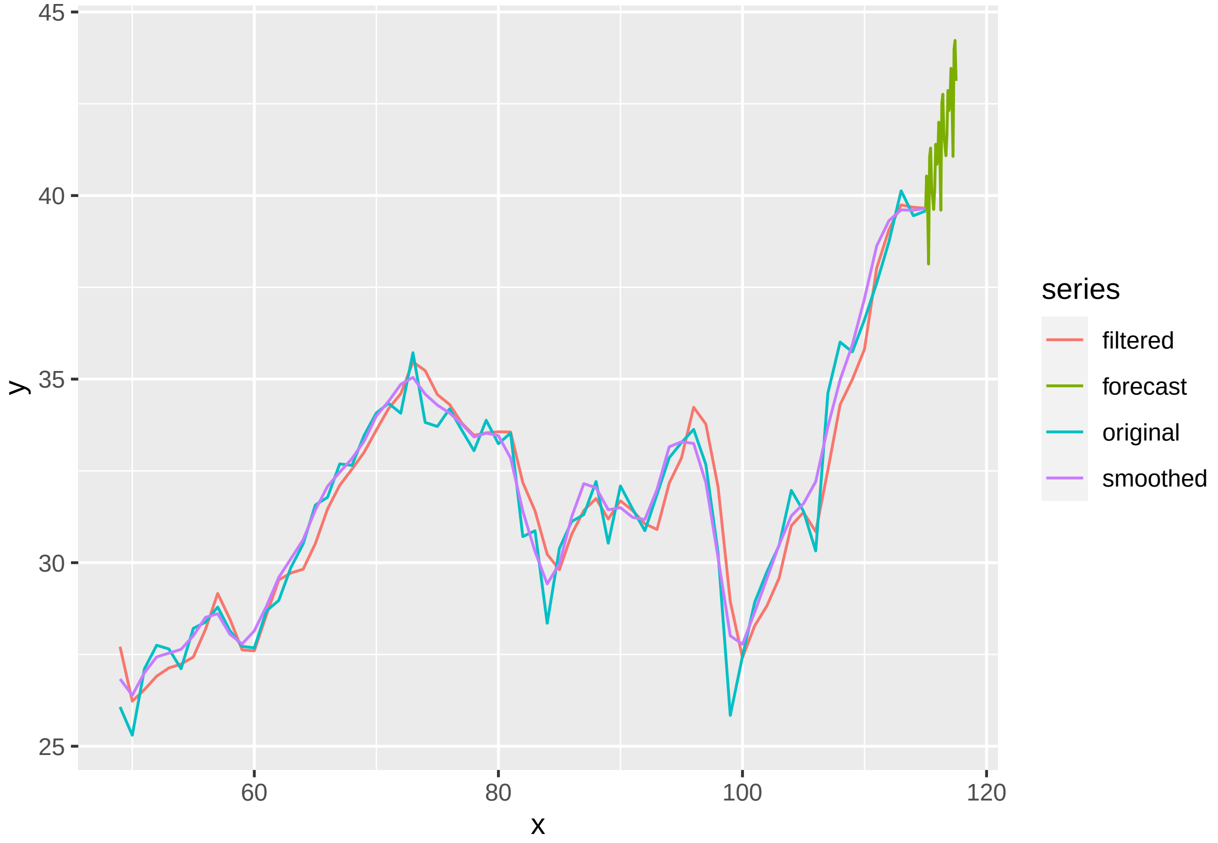


```
data_zoo <- as.zoo(fulls_ts)
actual <- as.vector(data_zoo[(length(data_zoo)-5):length(data_zoo)])
predicted <- as.vector(a6_forecast$f)
sqrt(mse(actual, predicted))
```

```
## [1] 2.040619
```

```
n <- 1*30
a30_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
```

```
aa <- a30_forecast$a[,1]* (-1)
aa <- cbind(a30_forecast$a[,1], aa)
a <- drop(a30_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
a30_forecast$f
```

```
## Time Series:
## Start = 116
## End = 145
## Frequency = 1
##        Series 1
##  [1,] 40.53340
##  [2,] 39.69097
##  [3,] 38.13425
##  [4,] 41.05571
##  [5,] 41.29494
##  [6,] 40.19318
##  [7,] 39.93281
##  [8,] 39.62062
##  [9,] 40.17932
## [10,] 41.39753
## [11,] 40.84912
## [12,] 41.04548
## [13,] 41.99888
## [14,] 41.15645
## [15,] 39.59973
## [16,] 42.52119
## [17,] 42.76042
## [18,] 41.65866
## [19,] 41.39829
## [20,] 41.08610
## [21,] 41.64480
## [22,] 42.86301
## [23,] 42.31460
## [24,] 42.51096
## [25,] 43.46436
## [26,] 42.62193
## [27,] 41.06521
## [28,] 43.98667
## [29,] 44.22590
## [30,] 43.12414
```

```
## Kalman Filtering for VT Data
split_df <- ts_split(ts.obj = fullv_ts, sample.out = 6)

training = split_df$train

model <- function(p) {
  return(dlmModPoly(2, dV = p[1], dW = p[2:3]) + dlmModSeas(12, dV = p[4]))
}

## Parameter Estimation
mle <- dlmMLE(training, parm = c(0.1,0.001,1,1), build = model)
if (mle$convergence == 0) print ('converge') else print('nonconverge')
```

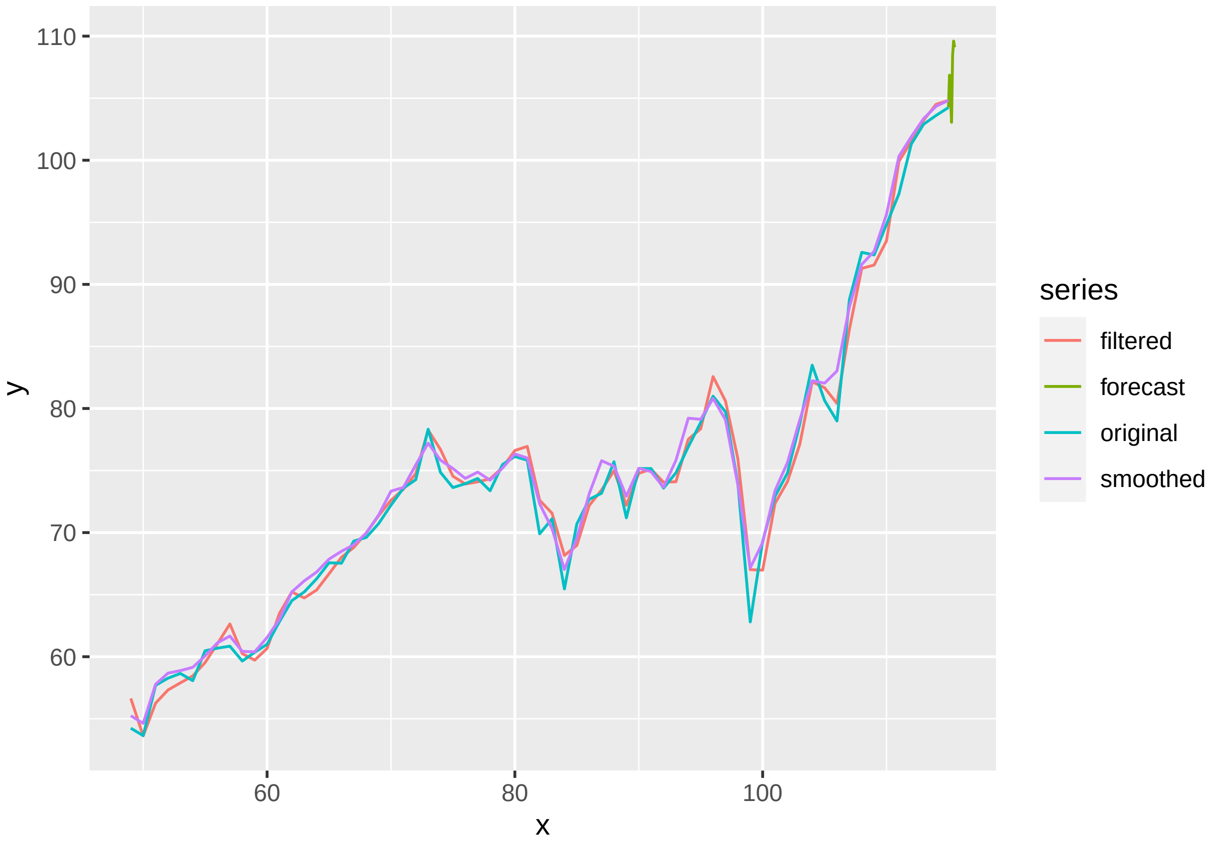
```
## [1] "converge"
```

```
mle$par
```

```
## [1] -0.449998378  5.488027323 -0.003092807  0.450001623
```

```
modelfit = model(mle$par)

## Filtering and Smoothing
modelfilter <- dlmFilter(training, modelfit)
modelsmoothed <- dlmSmooth(training, modelfit)
```

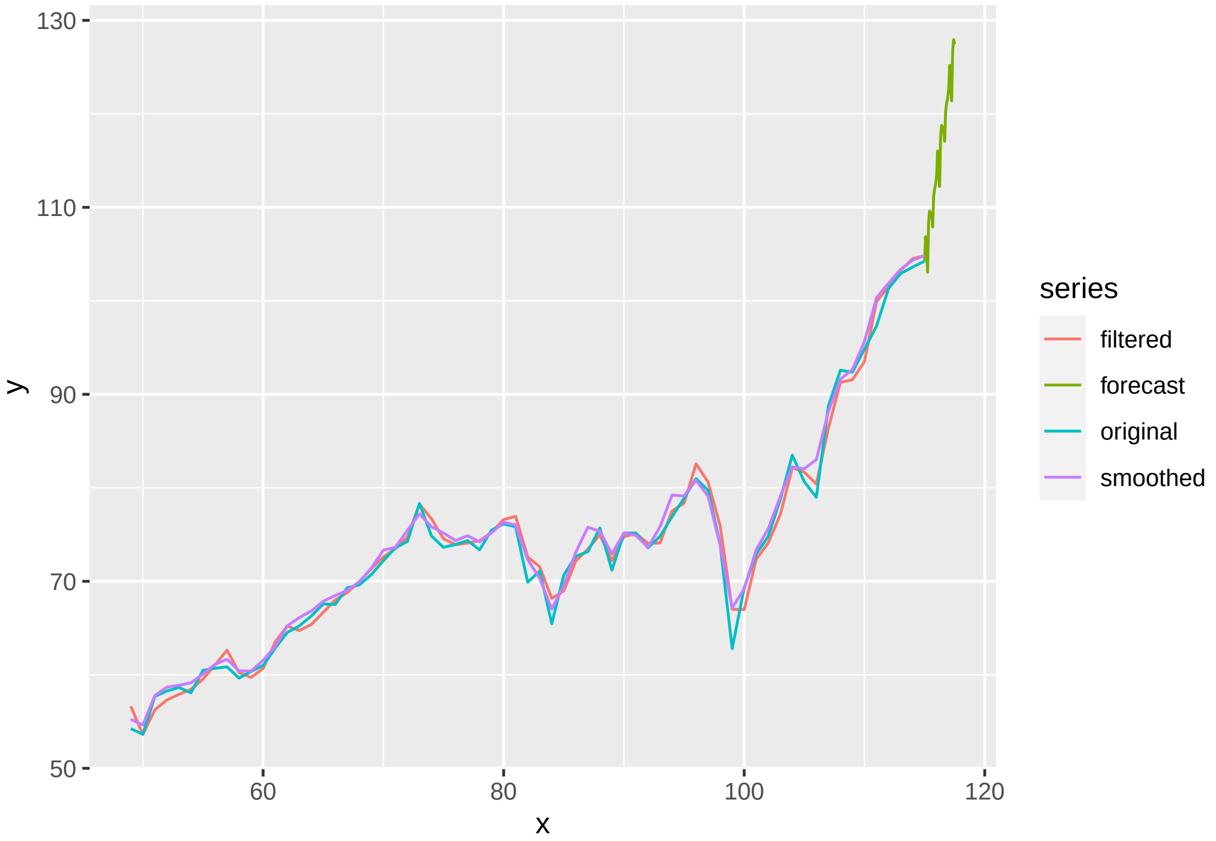
```
n <- 1*6
a6_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a6_forecast$a[,1]* (-1)
aa <- cbind(a6_forecast$a[,1], aa)
a <- drop(a6_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
data_zoo <- as.zoo(fullv_ts)
actual <- as.vector(data_zoo[(length(data_zoo)-5):length(data_zoo)])
predicted <- as.vector(a6_forecast$f)
sqrt(mse(actual, predicted))
```

```
## [1] 4.189202
```

```
n <- 1*30
a30_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a30_forecast$a[,1]* (-1)
aa <- cbind(a30_forecast$a[,1], aa)
a <- drop(a30_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
a30_forecast$f
```

```
## Time Series:
## Start = 116
## End = 145
## Frequency = 1
##       Series 1
##  [1,] 106.8574
##  [2,] 104.9247
##  [3,] 103.0605
##  [4,] 108.4527
##  [5,] 109.6058
##  [6,] 109.1064
##  [7,] 109.3614
##  [8,] 107.9049
##  [9,] 111.1133
## [10,] 112.0042
## [11,] 112.4869
## [12,] 113.4170
## [13,] 116.0244
## [14,] 114.0917
## [15,] 112.2275
## [16,] 117.6197
## [17,] 118.7728
## [18,] 118.2734
## [19,] 118.5285
## [20,] 117.0720
## [21,] 120.2804
## [22,] 121.1712
## [23,] 121.6540
## [24,] 122.5841
## [25,] 125.1915
## [26,] 123.2588
## [27,] 121.3946
## [28,] 126.7868
## [29,] 127.9398
## [30,] 127.4405
```

```
## Kalman Filtering for XLF Data
split_df <- ts_split(ts.obj = fullx_ts, sample.out = 6)

training = split_df$train

model <- function(p) {
  return(dlmModPoly(2, dV = p[1], dW = p[2:3]) + dlmModSeas(12, dV = p[4]))
}

## Parameter Estimation
mle <- dlmMLE(training, parm = c(0.1,0.001,1,1), build = model)
if (mle$convergence == 0) print (’converge’) else print(’nonconverge’)
```

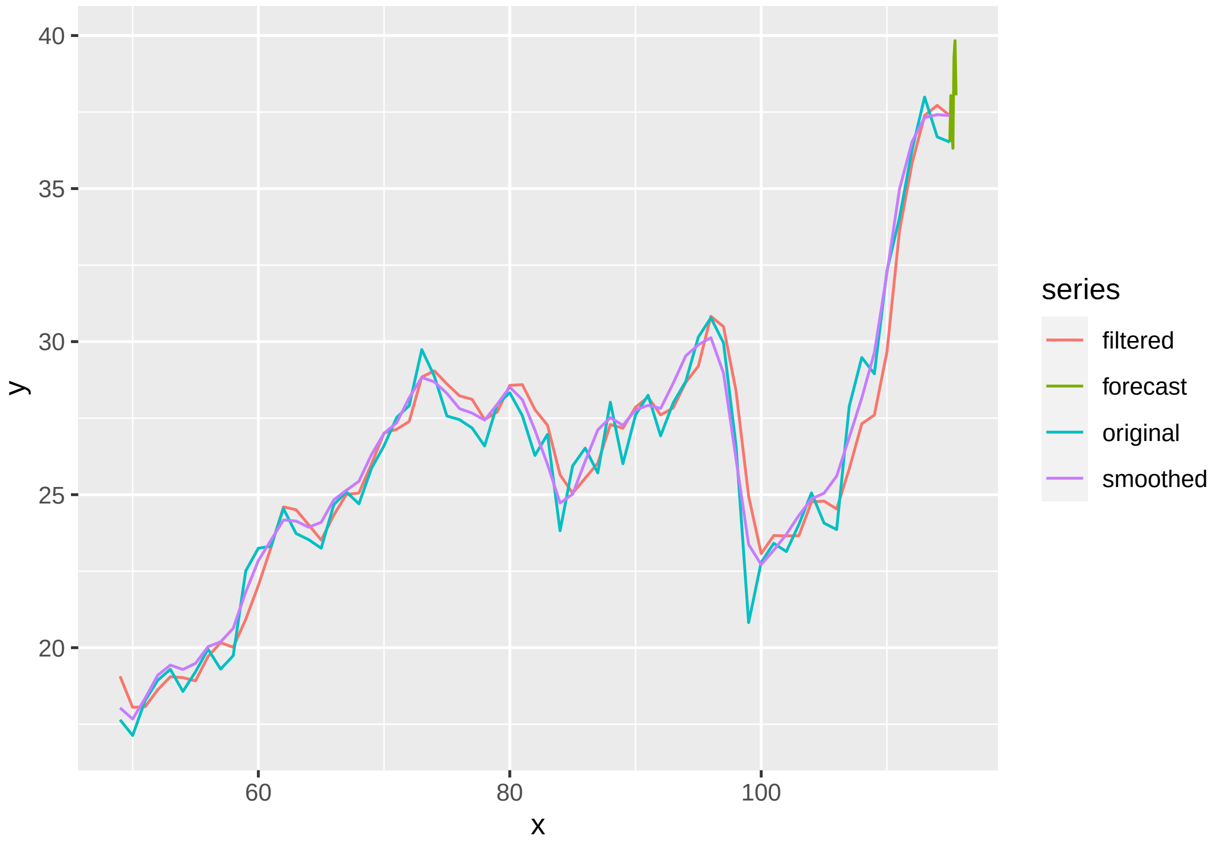
```
## [1] "converge"
```

```
mle$par
```

```
## [1] -4.498194e-01 -1.122207e+00 -4.487833e-06  4.501806e-01
```

```
modelfit = model(mle$par)

## Filtering and Smoothing
modelfilter <- dlmFilter(training, modelfit)
modelsmoothed <- dlmSmooth(training, modelfit)
```

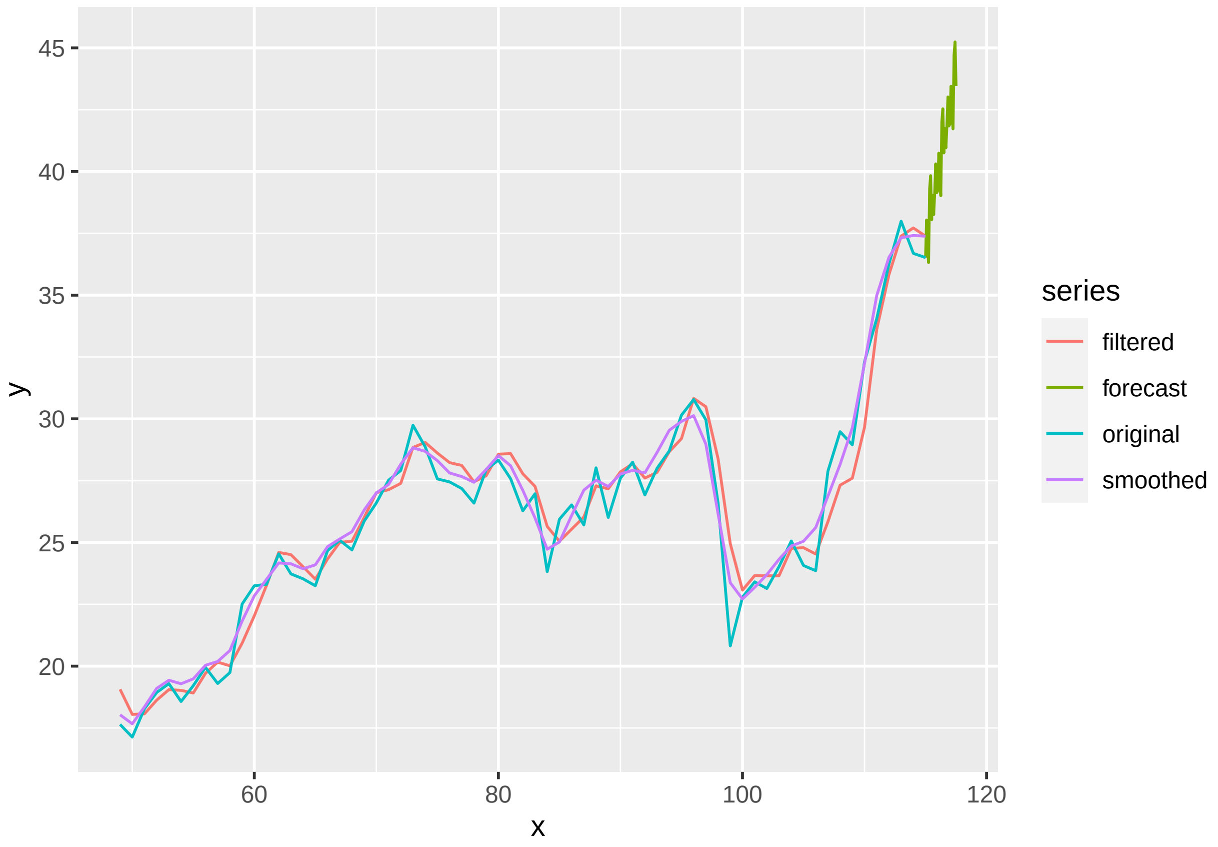
```
n <- 1*6
a6_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a6_forecast$a[,1]* (-1)
aa <- cbind(a6_forecast$a[,1], aa)
a <- drop(a6_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
data_zoo <- as.zoo(fullx_ts)
actual <- as.vector(data_zoo[(length(data_zoo)-5):length(data_zoo)])
predicted <- as.vector(a6_forecast$f)
sqrt(mse(actual, predicted))
```

```
## [1] 1.878895
```

```
n <- 1*30
a30_forecast <- dlmForecast(modelfilter, nAhead = n)

x <- index(training)
xf <- seq(max(x), max(x) + n/12, 1/12)
aa <- a30_forecast$a[,1]* (-1)
aa <- cbind(a30_forecast$a[,1], aa)
a <- drop(a30_forecast$a%*%t(FF(modelfit)))
a <- c(tail(training, 1), a)
df <- rbind(
  data.frame(x = x, y = as.numeric(training), series = "original"),
  data.frame(x = x, y = apply(modelfilter$m[-1,1:2],1,sum), series = "filtered"),
  data.frame(x = x, y = apply(modelsmoothed$s[-1,1:2],1,sum), series = "smoothed"),
  data.frame(x = xf, y = a, series = "forecast")
)

dlm <- ggplot(subset(df, x > 48), aes(x=x, y=y, colour = series)) + geom_line()
dlm
```



```
a30_forecast$f
```

```
## Time Series:
## Start = 116
## End = 145
## Frequency = 1
##       Series 1
##  [1,] 38.03820
##  [2,] 36.85741
##  [3,] 36.31544
##  [4,] 39.29646
##  [5,] 39.83338
##  [6,] 38.04798
##  [7,] 39.03736
##  [8,] 38.25141
##  [9,] 39.14232
## [10,] 40.30857
## [11,] 39.13782
## [12,] 39.22401
## [13,] 40.74214
## [14,] 39.56135
## [15,] 39.01937
## [16,] 42.00039
## [17,] 42.53732
## [18,] 40.75192
## [19,] 41.74129
## [20,] 40.95534
## [21,] 41.84625
## [22,] 43.01250
## [23,] 41.84176
## [24,] 41.92795
## [25,] 43.44607
## [26,] 42.26528
## [27,] 41.72330
## [28,] 44.70432
## [29,] 45.24125
## [30,] 43.45585
```