Breakpoints for farm field runoff

Wesley Brooks

First read the data:

```
[1] 831
```

The following is the R function that produces a piecewise regression model:

And the next function (`peicewise_plot`) plots the piecewise regression model:

We identify the best breakpoint by using R's optimize function to minimize the residual standard error:

```
> #soil_moisture_limits = c(33, 43) #set the limits
> #optimize(piecewise, soil_moisture_limits, x=data$sm, y=data$rc)$minimum
```

Now, let us do the soil moisture breakpoint analysis. The breakpoints at each farm individually, and for all farms in aggregate are found by the following code (breakpoints are stored in the variable `sm_breakpoints` for use in plotting [next code chunk]):

```
Breakpoints by farm:

df1: 35
df3: 35
df5: 40
df7: 40
df8: 40
koepke: 35
pagel: 35
saxon: 40
```

Then the plots are produced by calling the function `piecewise_plot` (code is listed above):

Now get the I30 breakpoints:

```
df1: 2.3
df3: 0.7
df5: 2.2
df7: 2.1
df8: 3.4
koepke: 1.5
pagel: 2.3
saxon: 2.1
```

When we put the events in bins based on their antecedent soil moisture (SM: high, medium, and low), the following are the I30 breakpoints (units are centimeters of rain per hour):

```
Intensity breakpoints at koepke when binned by soil moisture:
0 <= SM < 30: 3.3
30 <= SM < 35: 1.8
35 <= SM < Inf: 1.6

Intensity breakpoints at pagel when binned by soil moisture:
0 <= SM < 30: 1.3
30 <= SM < 35: 0.6
35 <= SM < Inf: 0.6

Intensity breakpoints at saxon when binned by soil moisture:
0 <= SM < 35: 1.8
35 <= SM < 40: 1.2
40 <= SM < Inf: 2
```

When we put the events in bins based on their antecedent soil moisture (SM: high, medium, and low), the following are the precipitation breakpoints (units are centimeters of rain):

```
Precipitation breakpoints at koepke when binned by soil moisture:
0 <= SM < 30: 2.8
30 <= SM < 35: 1.72
35 <= SM < Inf: 1.58

Precipitation breakpoints at pagel when binned by soil moisture:
0 <= SM < 30: 2.63
```

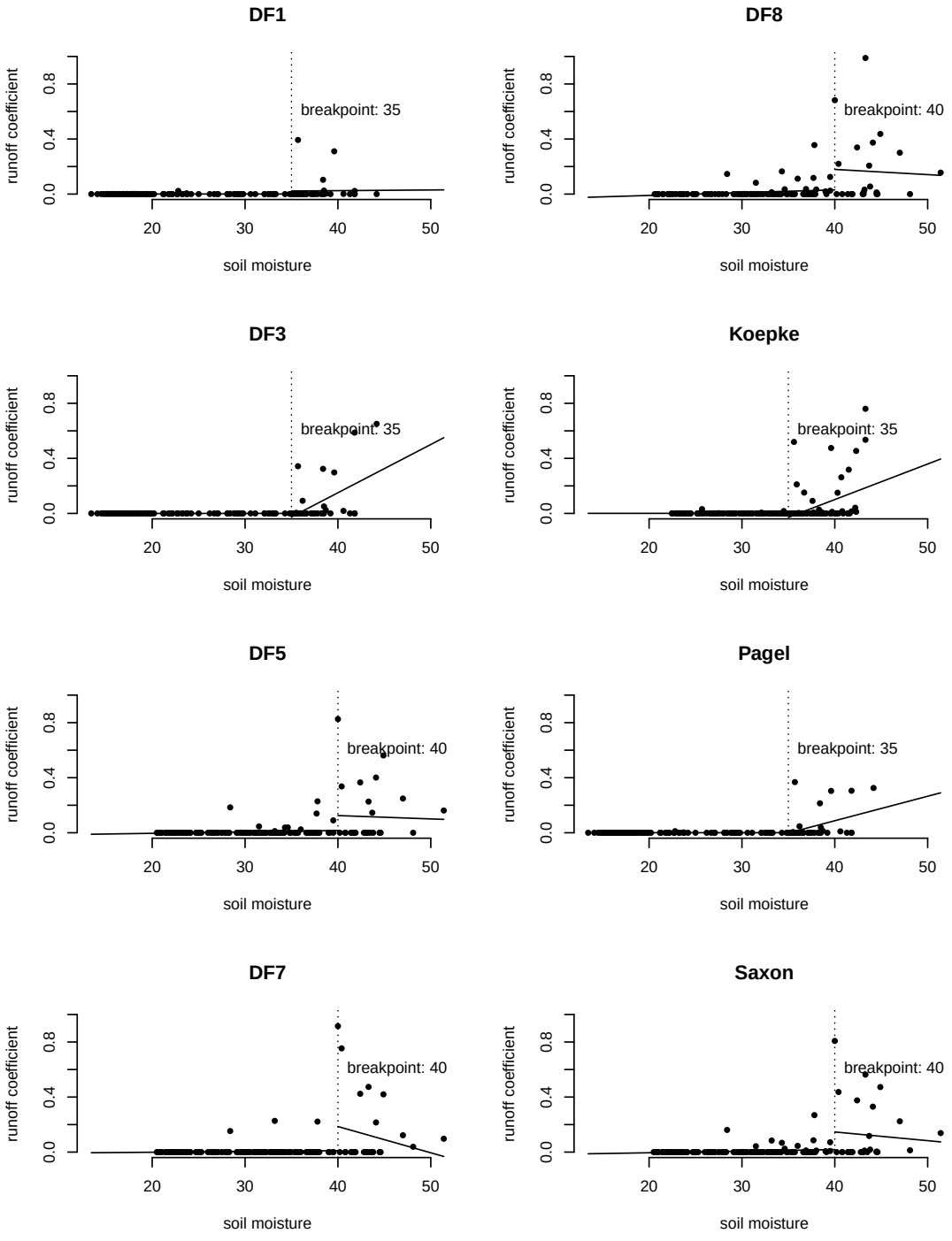

Figure 1: caption.

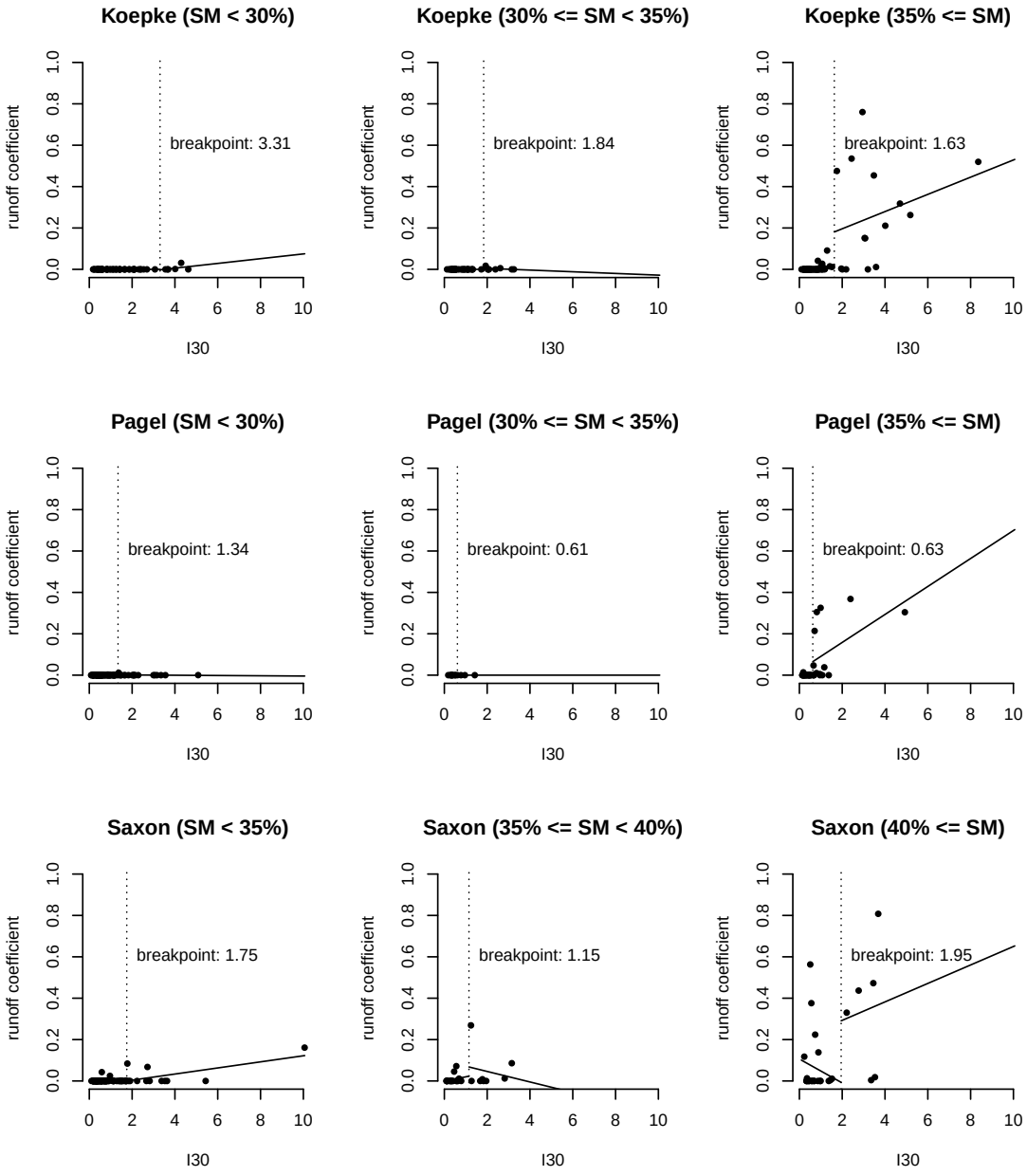

Figure 2: Top row: Koepke, middle row: Pagel, bottom row: Saxon.

```
30 <= SM < 35: 1.17
35 <= SM < Inf: 2.03

Precipitation breakpoints at saxon when binned by soil moisture:
0 <= SM < 35: 2.51
35 <= SM < 40: 1.18
40 <= SM < Inf: 2.26
```

Rather than three bins, let's try two (above vs. below the SM breakpoint):

```
Intensity breakpoints at koepke when binned by soil moisture:
-Inf <= SM < 35: 1.9
35 <= SM < Inf: 1.6

Intensity breakpoints at pagel when binned by soil moisture:
-Inf <= SM < 35: 1.3
35 <= SM < Inf: 0.6

Intensity breakpoints at saxon when binned by soil moisture:
-Inf <= SM < 40: 0.9
40 <= SM < Inf: 2
```

When we put the events in bins based on their antecedent soil moisture (SM: high or low), the following are the precipitation breakpoints (units are centimeters of rain):

```
Precipitation breakpoints at koepke when binned by soil moisture:
-Inf <= SM < 35: 1.97
35 <= SM < Inf: 1.58

Precipitation breakpoints at pagel when binned by soil moisture:
-Inf <= SM < 35: 2.62
35 <= SM < Inf: 2.03

Precipitation breakpoints at saxon when binned by soil moisture:
-Inf <= SM < 40: 2.01
40 <= SM < Inf: 2.26
```

```
>     anova.bp <- function(data, split.on, site=NA, conts=FALSE, discrete.xrange=FALSE
+     {
+         if(!conts)
+         {
+             if(discrete.xrange)
+                 th <- which.min( sapply(X=xrange, FUN=piecewise_disjoint, x=data[dat
```

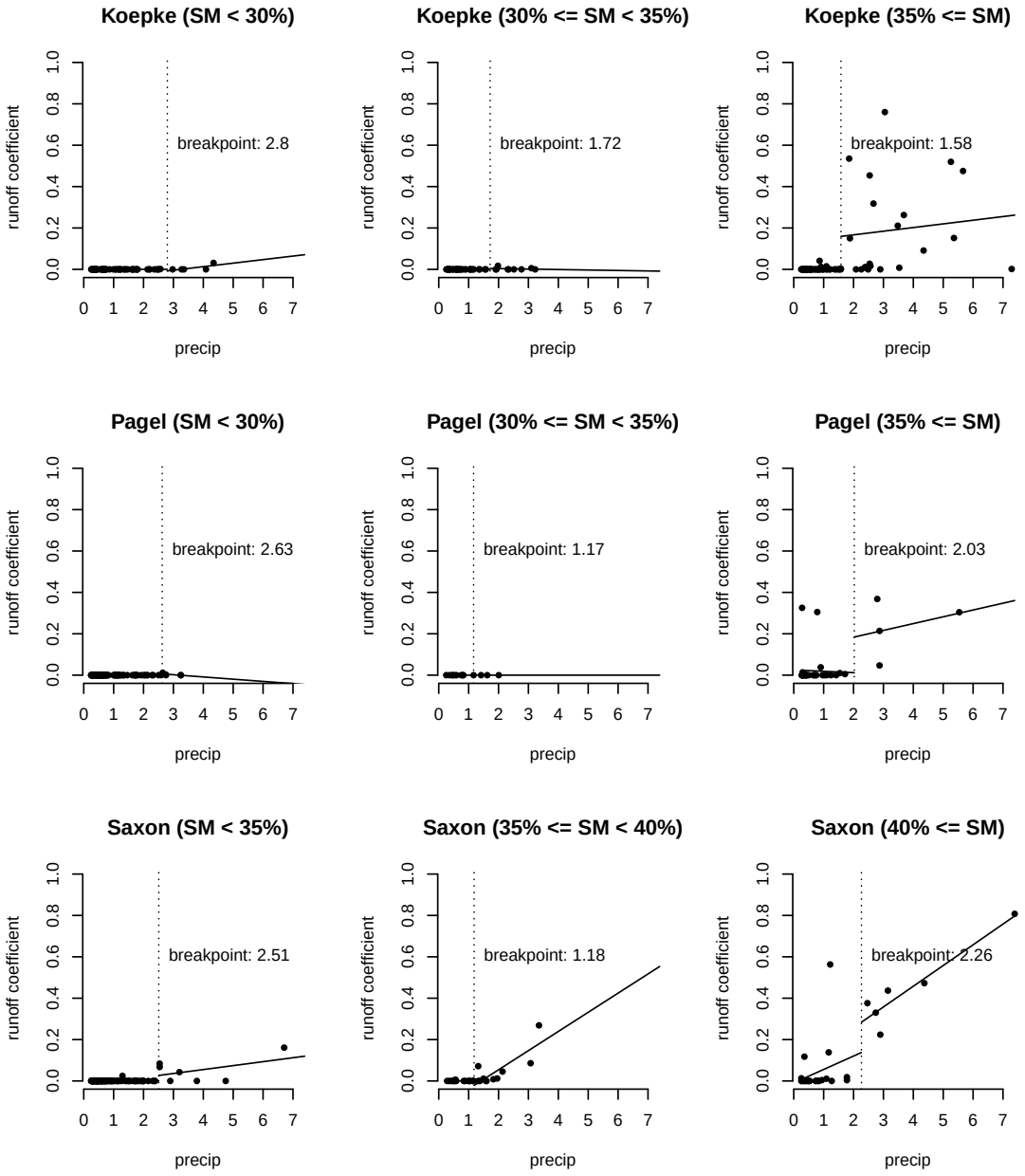

Figure 3: Top row: Koepke, middle row: Pagel, bottom row: Saxon.

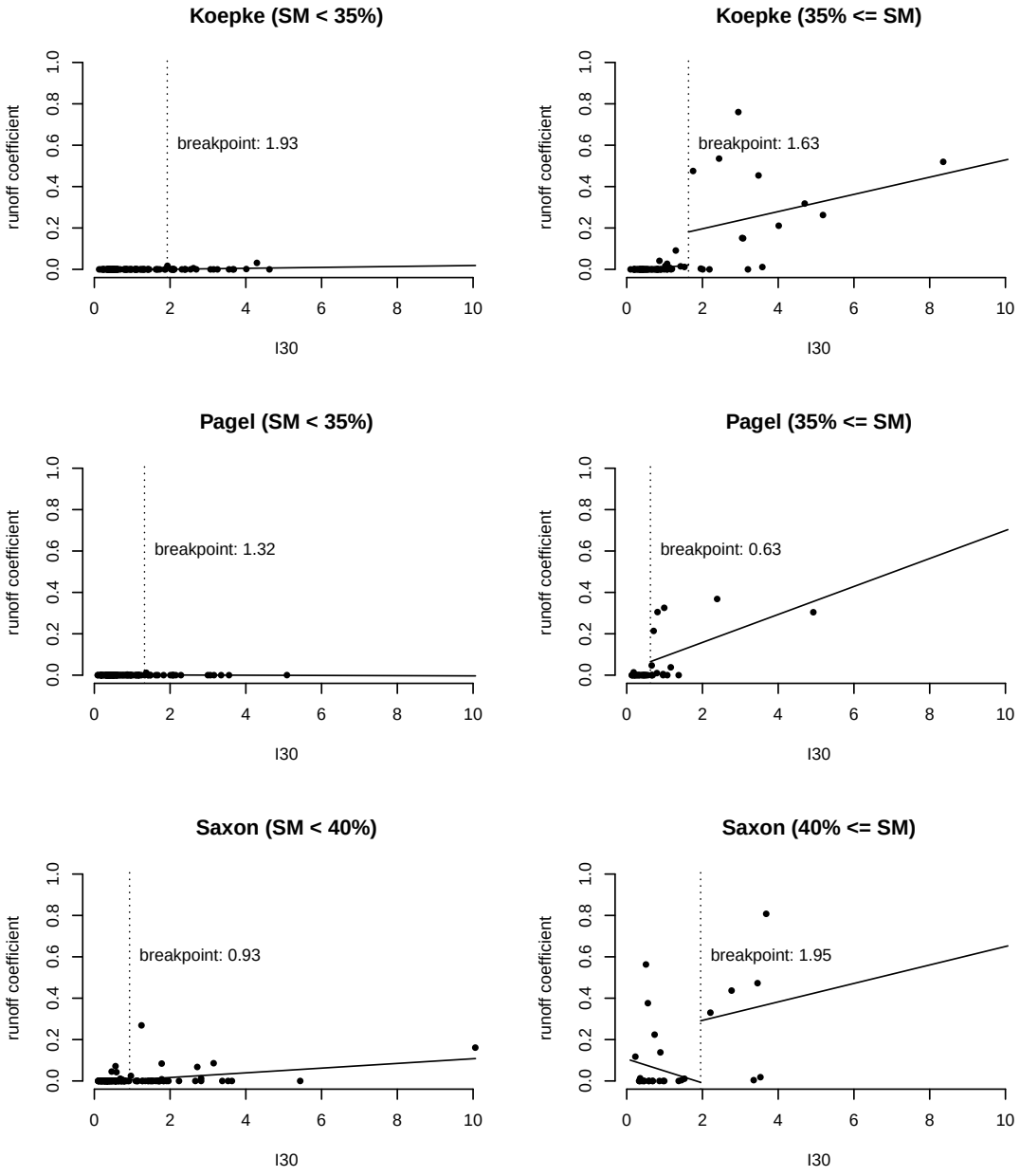

Figure 4: Top row: Koepke, middle row: Pagel, bottom row: Saxon.

```
+             else
+             {
+                 lower_lim = sort( data[data$site==site,split.on] )[min.in.node]
+                 upper_lim = sort(data[data$site==site,split.on], decreasing=T )[min.
+                 th <- optimize(piecewise_disjoint, upper=upper_lim, lower=lower_lim,
+             }
+
+             X.bp = cbind( ifelse(data[data$site==site,split.on]>=th, 1, 0), data[dat
+         }
+         else
+         {
+             if(discrete.xrange)
+                 th <- which.min( sapply(X=xrange, FUN=piecewise_conts, x=data[data$s
+             else
+             {
+                 lower_lim = sort( data[data$site==site,split.on] )[min.in.node]
+                 upper_lim = sort(data[data$site==site,split.on], decreasing=T )[min.
+                 th <- optimize(piecewise_conts, upper=upper_lim, lower=lower_lim, x=
+             }
+
+             X.bp = cbind( data[data$site==site,split.on], pmax(0, data[data$site==si
+         }
+
+         X.line = data[data$site==site,split.on]
+         y = data[data$site==site,]$rc
+
+         if(return.X)
+             return(list(line=X.line, bp=X.bp, theta=th))
+
+         fit.bp = lm(y~X.bp)
+         fit.line = lm(y~X.line)
+
+         if(return.models)
+             return(list(bp=fit.bp, line=fit.line, theta=th))
+
+         cat(paste("theta= ", th, "\n", sep=""))
+
+         if(!conts)
+             1-pf(q=(sum(fit.line$resid**2) - sum(fit.bp$resid**2)) / 3 / (sum(fit.bp
+         else
+             1-pf(q=(sum(fit.line$resid**2) - sum(fit.bp$resid**2)) / 2 / (sum(fit.bp
+     }
```

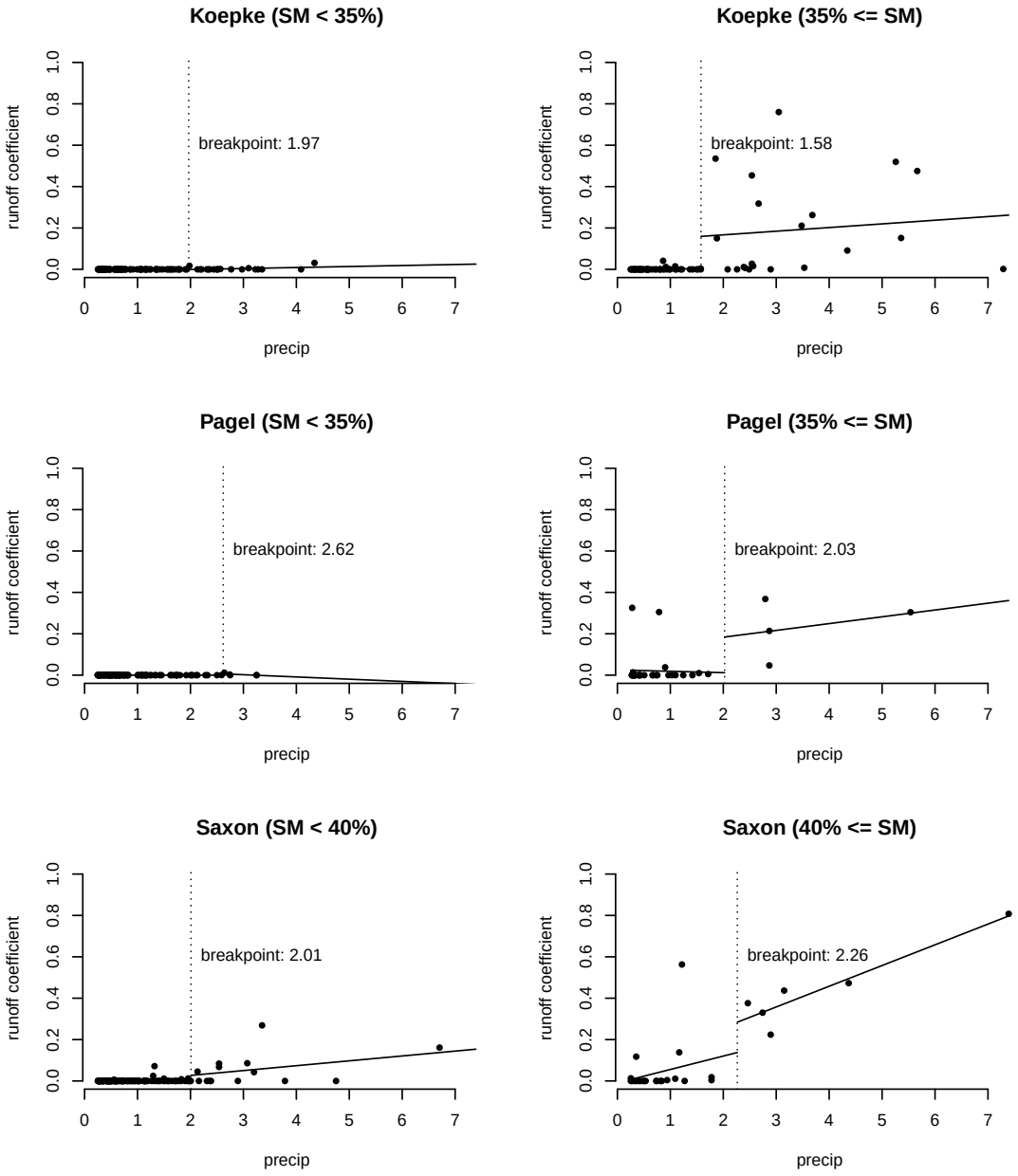

Figure 5: Top row: Koepke, middle row: Pagel, bottom row: Saxon.