

Farbige Achsen

Jörg Sellmann
 Julius Kühn-Institut
 Stahnsdorfer Damm 81
 14532 Kleinmachnow
 joerg.sellmann@julius-kuehn.de

Zusammenfassung

Kategorien führen bei sgplot-Grafiken zur Achsenbeschriftung. Gruppen bewirken eine farbliche Unterscheidung innerhalb der Grafik und der Legende. Möchte man aber die Achsenbeschriftung anhand einer dritten Variablen, nennen wir es Klasse, farblich unterschiedlich kennzeichnen, so hilft an dieser Stelle SGANNO.

Schlüsselwörter: PROC SGPLOT, PROC SGANNO, discreteoffset, reflate

1 Daten und Formate

1.1 Daten

Im Rahmen des Projektes „Demonstrationsbetriebe integrierter Pflanzenschutz“ (<http://demo-ips.julius-kuehn.de>) werden jährlich bundesweit Daten zu den durchgeführten Pflanzenschutzmaßnahmen erhoben. Ein wichtiges Auswertungsmerkmal ist der Behandlungsindex BI, der sich aus den vier Kategorien H (Herbizide), I (Insektizide), F (Fungizide) und W (Wachstumsregler) zusammensetzt. Über die Projektlaufzeit hinweg soll u. a. analysiert werden, wie sich der BI verändert.

Die Datenbankabfrage liefert (hier mit fiktiven Werten) folgende Daten:

Tabelle 1: (fiktive) Daten zum Behandlungsindex

Region	Jahr	H	I	F	W	Schläge
N	2012	1	1	2	2	12
N	2013	2	1	2	3	17
N	2014	1	2	1	2	15
N	2015	2	2	1	1	22
O	2012	1	1	2	2	18
O	2013	2	3	2	1	19
O	2014	3	2	1	2	15
O	2015	2	1	1	3	14
S	2012	3	1	2	2	12
S	2013	2	3	2	3	10

S	2014	3	.	3	2	11
S	2015	2	2	1	3	11
W	2012	3	4	4	2	22
W	2013	2	4	2	3	24
W	2014	3	4	3	2	23
W	2015	2	2	4	3	21

Eine Wunschgrafik wäre, die Zusammensetzung des BI über die Jahre 2012-2015 und die Regionen N, S, O und W anzuzeigen. Da die Daten in der vorliegenden Struktur für die gewünschte Grafik ungeeignet sind, werden sie mittels `transpose` umgewandelt:

```
proc transpose data=bi out=bit2(rename=( _name_=KAT)) prefix=J;
  by Region;
  copy n;
  id Jahr;
  var h i f w;
run;
```

Der resultierende Datensatz BIT2 sieht wie folgt aus:

Tabelle 2: transponierte Daten zum Behandlungsindex

Region	n	KAT	J2012	J2013	J2014	J2015
N	12	H	1	2	1	2
N	17	I	1	1	2	2
N	15	F	2	2	1	1
N	22	W	2	3	2	1
O	18	H	1	2	3	2
O	19	I	1	3	2	1
O	15	F	2	2	1	1
O	14	W	2	1	2	3
S	12	H	3	2	3	2
S	10	I	1	3	.	2
S	11	F	2	2	3	1
S	11	W	2	3	2	3
W	22	H	3	2	3	2
W	24	I	4	4	4	2
W	23	F	4	2	3	4
W	21	W	2	3	2	3

1.2 Formate

Um unsere „eigenen“ Farben und Bezeichnungen zu verwenden, definieren wir sie in zwei Datenschritten bzw. mit zwei Formaten:

```

/* Farben definieren */
data kat1;
input id $ value $ fillcolor $;
cards;
KAT F blue
KAT H green
KAT I red
KAT W yellow
;
run;

data kat2;
input id $ value $10. fillcolor $;
cards;
KAT Fungizid blue
KAT Herbizid green
KAT Insektizid red
KAT W.regler yellow
;
run;

/* ordentliche Beschriftung */
proc format;
value $region N=Nord O=Ost W=West S=Süd;
value $kat I=Insektizid H=Herbizid F=Fungizid W=W.regler;
run;

```

Zu beachten ist die Verwendung der Farben KAT1 bzw. KAT2 mittels `datrmap`. Wird die Region unter Einsatz des Formats `$region` verwendet, so kommt KAT2 zur Anwendung, ansonsten KAT1.

2 Grafiken

2.1 Ausgangsgrafik

Das SAS-Statement für die erste Grafik ist bereits komplexer als gewöhnlich. Da die beiden Untergliederungsmöglichkeiten `(category=)region` und `group=KAT` bereits ausgeschöpft sind, werden als weitere Unterteilung, nennen wir es *klasse*, die 4 Balkengrafiken übereinander gepackt. Dabei werden die Säulen mittels `discreteoffset=` nach links bzw. rechts verschoben.

```
proc sgplot data=bit2 noautolegend datrmap=kat2;  
styleattrs datacontrastcolors=(grey);  
vbar region / response=J2012 group=KAT attrid=kat  
discreteoffset=-0.3 barwidth=0.15 name='A';  
vbar region / response=J2013 group=KAT attrid=kat  
discreteoffset=-.1 barwidth=0.15;  
vbar region / response=J2014 group=KAT attrid=kat  
discreteoffset=.1 barwidth=0.15;  
vbar region / response=J2015 group=KAT attrid=kat  
discreteoffset=0.3 barwidth=0.15;  
axis display=(nolabel noticks);  
yaxis label="Behandlungsindex" min=0 values=(0 to 14 by 2);  
format region $region. kat $kat.;  
keylegend 'A' / title='';  
run;
```

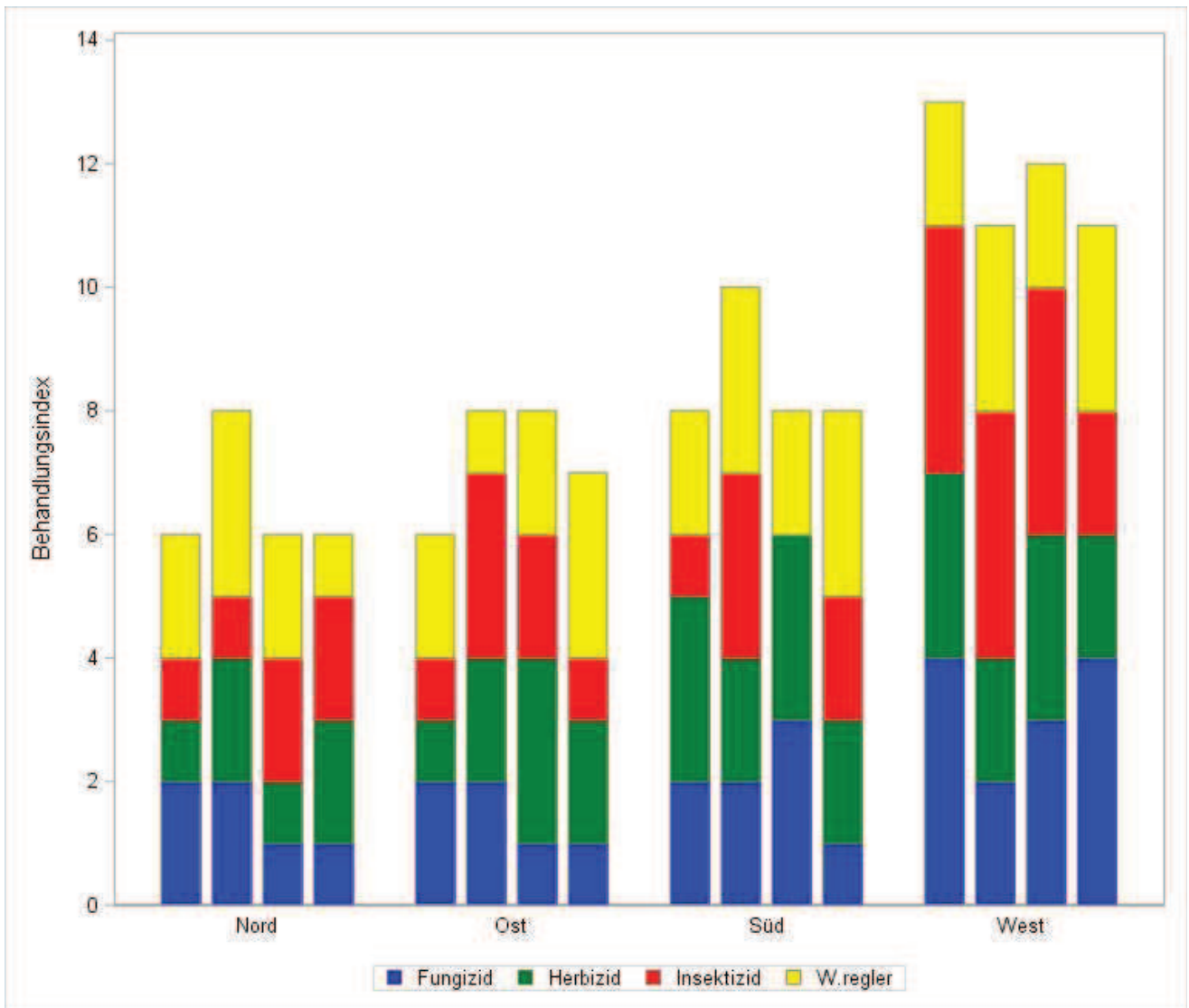


Abbildung 1: Ausgangsgrafik

Es ist inklusive der Farben und den Bezeichnungen alles wunschgemäß dargestellt. Lediglich die Angabe der Jahre fehlt.

2.2 Jahre hinzufügen

Um die Jahresangaben hinzuzufügen, ist eine zweite X-Achsenbeschriftung erforderlich. Das lässt sich nur mittels eines `sgraph`-Statements erreichen. Dazu wird ein entsprechender Datensatz erstellt, der in der Variablen `discreteoffset` die relative X-Position bzgl. der Region und in der Variablen `y1` die Y-Position der Bezeichnungen enthält. Alle in diesem Datensatz verwendeten Variablennamen sind zwingend vorge-schrieben.

```
data anno_jahrel;
retain
  function 'text' x1space 'datavalue' y1space graphpercent' y1 8
  textsize 8 textcolor 'black' anchor 'top' justify 'center';
input xcl $ label $ discreteoffset;
cards;
Nord 2012 -0.3
Nord 2013 -0.1
Nord 2014 0.1
Nord 2015 0.3
Ost 2012 -0.3
Ost 2013 -0.1
Ost 2014 0.1
Ost 2015 0.3
Süd 2012 -0.3
Süd 2013 -0.1
Süd 2014 0.1
Süd 2015 0.3
West 2012 -0.3
West 2013 -0.1
West 2014 0.1
West 2015 0.3
;
run;
```

Wird jetzt die obige Anweisung für die Abb. 1 um die Option `sgraph=anno_jahrel` erweitert, sind wir dem Ziel einen bedeutenden Schritt näher:

```
proc sgraph data=bit2 noautolegend dattrmap=kat2
  sgraph=anno_jahrel;
styleattrs datacontrastcolors=(grey);
vbar region / response=J2012 group=KAT attrid=kat
  discreteoffset=-0.3 barwidth=0.15 name='A';
vbar region / response=J2013 group=KAT attrid=kat
  discreteoffset=-.1 barwidth=0.15;
vbar region / response=J2014 group=KAT attrid=kat
  discreteoffset=.1 barwidth=0.15;
vbar region / response=J2015 group=KAT attrid=kat
  discreteoffset=0.3 barwidth=0.15;
xaxis display=(nolabel noticks);
yaxis label="Behandlungsindex" min=0 values=(0 to 14 by 2);
format region $region. kat $kat.;
keylegend 'A' / title='';
run;
```

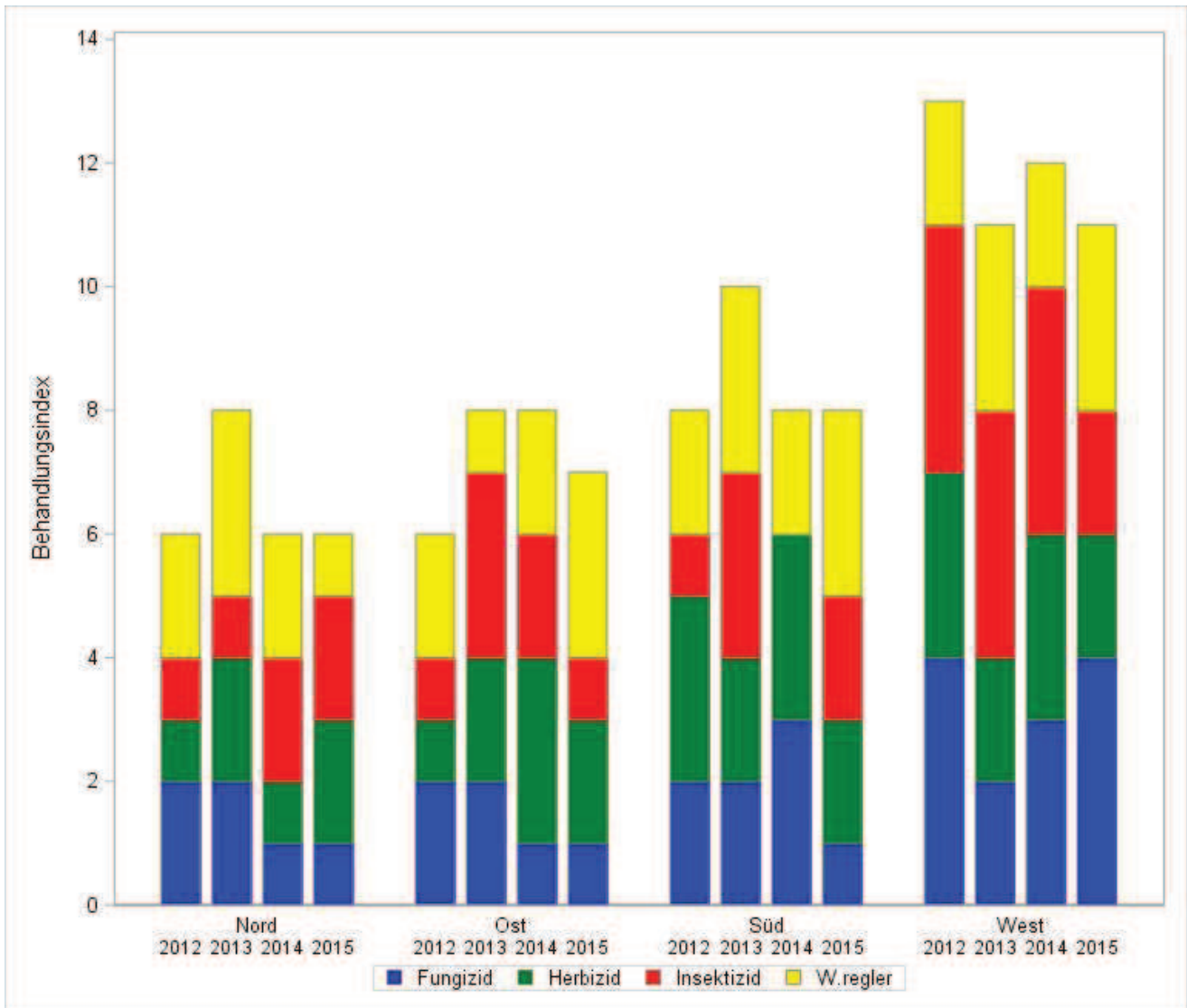


Abbildung 2: Beschriftung der Jahre hinzufügen

2.3 Farben hinzufügen

Die Anordnung der Regionen und Jahre ist noch nicht optimal. Zum zweiten möchten wir auch einzelne Jahre in der Achse farblich markieren, hier im Beispiel konsequent das Jahr 2013. Hierfür erzeugen wir einen zweiten Datensatz, der die Variable `textcolor` zeilenweise unterschiedlich belegt. Des Weiteren definieren wir auch die Positionen, Größen und Farben der Regionen.

```

data anno_jahre2;
retain function 'text' x1space 'datavalue' y1space graphpercent'
anchor 'top' justify 'center';
input xcl $ label $ discreteoffset y1 textsize textcolor $ @@;
cards;
Nord 2012 -0.3 9 8 black Nord 2013 -0.1 9 8 red
Nord 2014 0.1 9 8 black Nord 2015 0.3 9 8 black
Ost 2012 -0.3 9 8 black Ost 2013 -0.1 9 8 red
Ost 2014 0.1 9 8 black Ost 2015 0.3 9 8 black
Süd 2012 -0.3 9 8 black Süd 2013 -0.1 9 8 red

```

```
Süd 2014 0.1 9 8 black   Süd 2015 0.3 9 8 black
West 2012 -0.3 9 8 black West 2013 -0.1 9 8 red
West 2014 0.1 9 8 black   West 2015 0.3 9 8 black
Nord Nord 0 6 10 blue     Ost Ost 0 6 10 blue
Süd Süd 0 6 10 blue       West West 0 6 10 blue
;
run;
```

Zum besseren Verständnis ist der Datensatz in Tab. 3 vollständig angegeben, wobei sofort ersichtlich ist, wie die Positionen, Größen und Farben verändert werden können.

Der beispielhaft gewählte statische Ansatz in Form eines Datasteps lässt sich natürlich auch anderweitig per Makro, SQL-Anweisungen oder Einlesen aus Textdateien realisieren. Mit etwas Probieren ist zudem eine gute Abstimmung von `barwidth` und `discreteoffset` zu finden.

Tabelle 3: Datensatz `anno_jahre2`

function	x1space	y1space	anchor	justify	xc1	label	discreteoffset	y1	text size	text color
text	datavalue	graphpercent	top	center	Nord	2012	-0.3	9	8	black
text	datavalue	graphpercent	top	center	Nord	2013	-0.1	9	8	red
text	datavalue	graphpercent	top	center	Nord	2014	0.1	9	8	black
text	datavalue	graphpercent	top	center	Nord	2015	0.3	9	8	black
text	datavalue	graphpercent	top	center	Ost	2012	-0.3	9	8	black
text	datavalue	graphpercent	top	center	Ost	2013	-0.1	9	8	red
text	datavalue	graphpercent	top	center	Ost	2014	0.1	9	8	black
text	datavalue	graphpercent	top	center	Ost	2015	0.3	9	8	black
text	datavalue	graphpercent	top	center	Süd	2012	-0.3	9	8	black
text	datavalue	graphpercent	top	center	Süd	2013	-0.1	9	8	red
text	datavalue	graphpercent	top	center	Süd	2014	0.1	9	8	black
text	datavalue	graphpercent	top	center	Süd	2015	0.3	9	8	black
text	datavalue	graphpercent	top	center	West	2012	-0.3	9	8	black
text	datavalue	graphpercent	top	center	West	2013	-0.1	9	8	red
text	datavalue	graphpercent	top	center	West	2014	0.1	9	8	black
text	datavalue	graphpercent	top	center	West	2015	0.3	9	8	black
text	datavalue	graphpercent	top	center	Nord	Nord	0	6	10	blue
text	datavalue	graphpercent	top	center	Ost	Ost	0	6	10	blue
text	datavalue	graphpercent	top	center	Süd	Süd	0	6	10	blue
text	datavalue	graphpercent	top	center	West	West	0	6	10	blue

Ergänzen wir, um unten Platz für die Beschriftungen zu schaffen, die erste Zeile der vorherigen `sgplot`-Anweisung um die Option `pad=(bottom=40px)`, verwenden jetzt `sganno=anno_jahre2` und blenden die X-Achsen Beschriftung vollständig aus (`xaxis display=(nolabel noticks novalues)`), so erhalten wir die gewünschte Grafik:

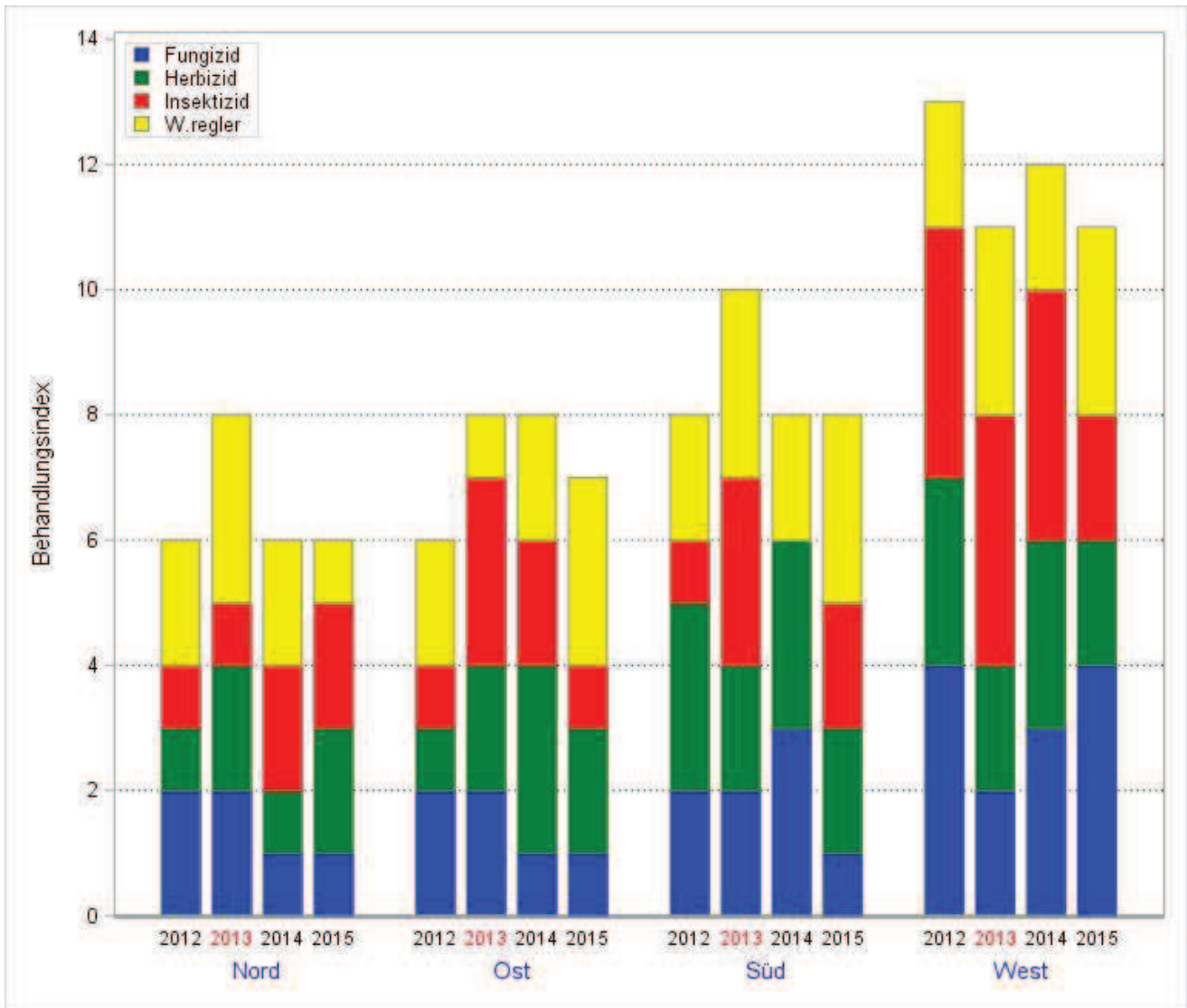


Abbildung 3: X-Beschriftungen vertauschen und 2013 farbig markieren

Ein kleiner Wermutstropfen bleibt: Die Legende kann nicht wie zuvor unterhalb platziert werden. Sie würde dann über den Balken liegen.

3 Bonus

Zusätzlich möchten wir noch eine Region vollständig herausheben, hier die Region S. Dazu wird die `refline`-Anweisung Zweck entfremdet. Wird die Linie entsprechend dick (`thickness=120px`), transparent und zeitlich vor den Balken gezeichnet, so erscheint sie als Block hinter/unter der entsprechenden Region.

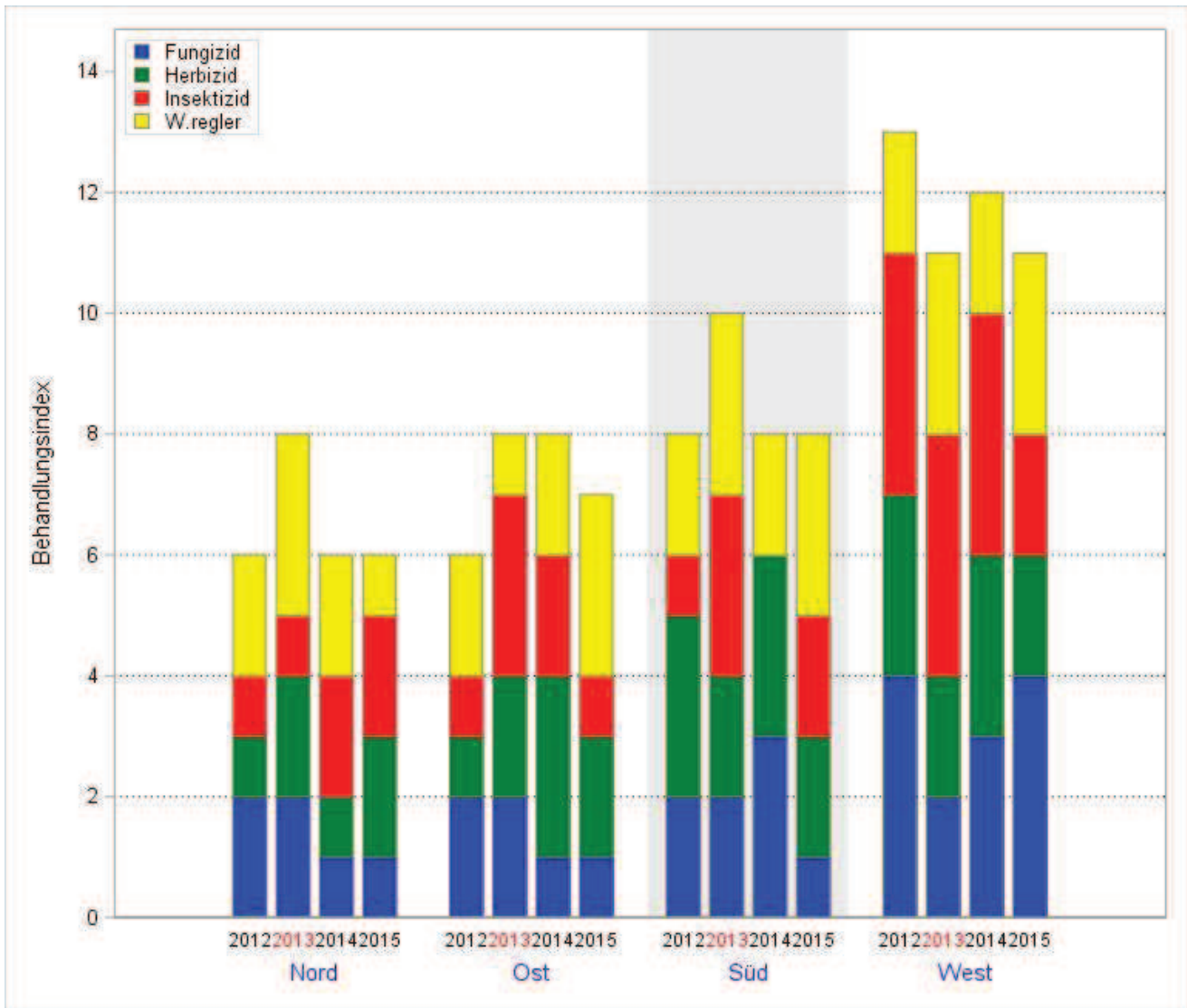


Abbildung 4: Eine Region hervorheben mittels REFLINE

Das vollständige Statement, welches ohne die erste `refline`-Anweisung die Abb. 3 ergeben würde, lautet:

```
proc sgplot data=bit2 noautolegend dattrmap=kat2
  sgranno=anno_jahre2 pad=(bottom=40px);
  styleattrs datacontrastcolors=(grey);
  reffline 'Süd' / axis=x lineattrs=(thickness=120px
    color=lightgray) transparency=0.5 labelloc=inside;
  reffline 2 to 12 by 2 / axis=y lineattrs=(pattern=dot
    thickness=1px color=black);
  vbar region / response=J2012 group=KAT attrid=kat
    discreteoffset=-0.3 barwidth=0.15 name='A';
  vbar region / response=J2013 group=KAT attrid=kat
    discreteoffset=-.1 barwidth=0.15;
  vbar region / response=J2014 group=KAT attrid=kat
    discreteoffset=.1 barwidth=0.15;
  vbar region / response=J2015 group=KAT attrid=kat
    discreteoffset=0.3 barwidth=0.15;
  xaxis display=(nolabel noticks novalues);
```

```
yaxis offsetmin=0.0 offsetmax=0.05 label="Behandlungsindex"  
  min=0 values=(0 to 14 by 2);  
format region $region. kat $kat.;  
keylegend 'A' / across=1 location=inside position=topleft  
  title='' opaque;  
run;
```

Extrem wichtig hierbei ist die Anweisung

```
yaxis offsetmin=0.0 offsetmax=0.05
```

Der Autor überlässt es dem geneigten Leser herauszufinden, was sie bewirkt.

4 Ausblick

Mit den hier gezeigten Anweisungen möchte der Autor dem interessierten Anwender Wege aufzeigen, die eigenen Grafiken anschaulicher und übersichtlicher zu gestalten. Die verwendeten Elemente `discreteoffset`, `sganno`, `format`, `refline`, `dattrmap` sind in nahezu jeder `sgplot`-Grafik einsetzbar und halten neben anderen noch eine Unzahl weiterer Gestaltungsideen bereit.

Die zudem noch vielfach beobachtete Vorgehensweise, Daten mit SAS aufzubereiten und anschließend mittels Copy&Paste nach Excel zu transportieren, um die Grafiken zu erzeugen, sollte langsam der Vergangenheit angehören.

5 Programme

Das hier verwendete Programm *colored_legend.sas*, wie auch die Programme des Autors aus den letzten Jahren, finden Sie unter:

<http://sf.julius-kuehn.de/sas>