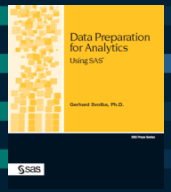
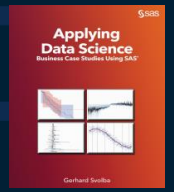
Bringen Sie Bewegung in Ihre SAS Ergebnisse! – Illustration der Erstellung von SAS Animated GIFs anhand eines Fallbeispiels





Gerhard Svolba, Data Scientist
SAS Austria

Data Scientist @SAS - Medium LinkedIn Github SAS-Books SAS Articles
Youtube DataPreparation4DataScience Data Science Use Cases

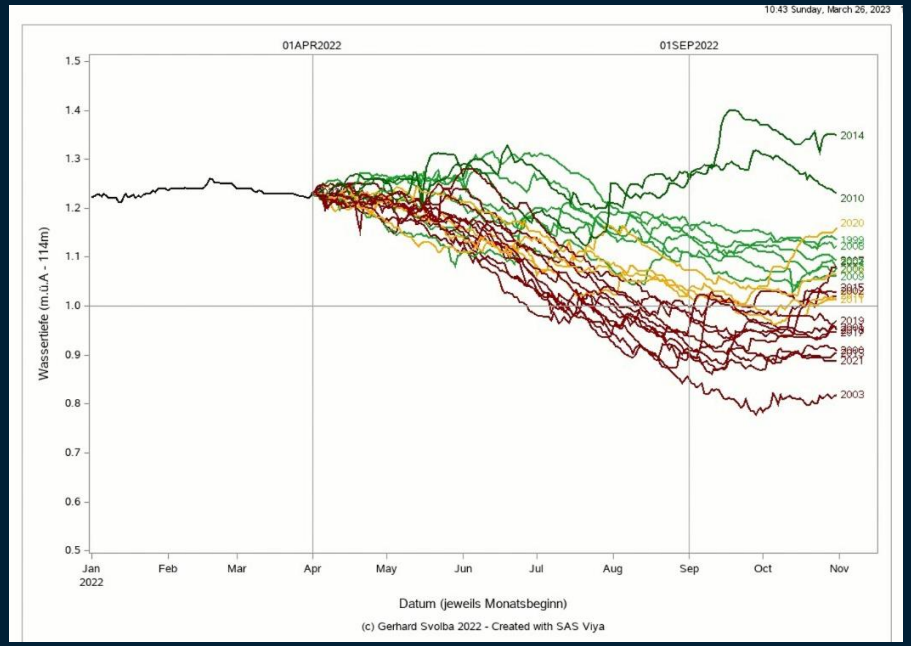
Hinweis

- Ein Beitrag mit Beispiel Code zum Thema dieses Vortrags ist in Vorbereitung. Der Link findet sich dann in dieser Sammlung
- Data Science and Data Preparation Article Overview by Gerhard
  - https://communities.sas.com/t5/SAS-Communities-Library/Data-Science-and-Data-Preparation-Article-Overview-by-Gerhard/ta-p/727875

Das erwartet Sie in diesem Vortrag

-1-
Welche Möglichkeiten bietet SAS mit dem Animated GIF Treiber?

-2-
Deep Dive in ein komplexeres Beispiel



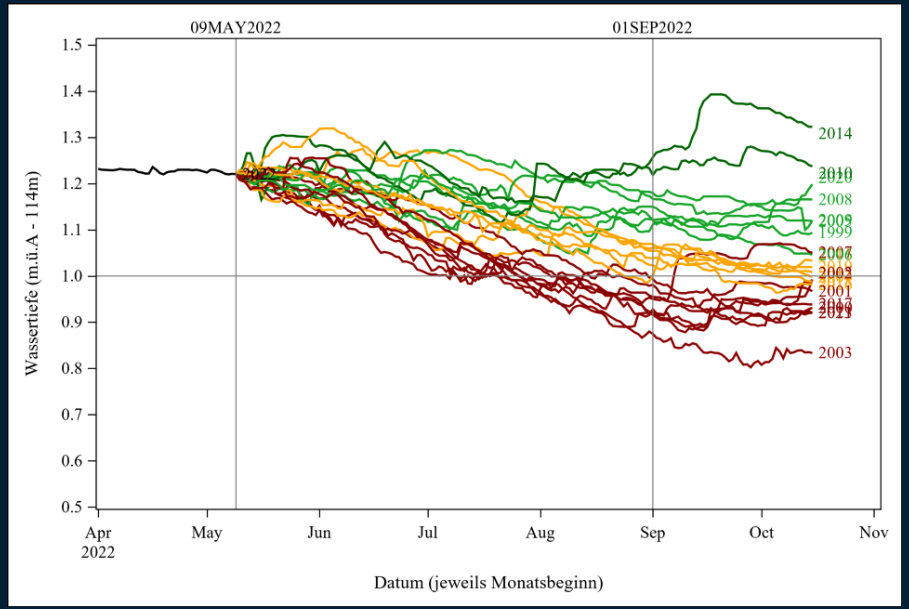
-3-
Dynamische Formatierung von Graphikelementen (z.B. Linienfarbe)

-4-
Do's and Don'ts bei der Erstellung von animierten Graphiken

Das erwartet Sie in diesem Vortrag

-1-

Welche Möglichkeiten bietet SAS mit dem Animated GIF Treiber?



The (old) GIFANIM Driver works for SAS/GRAPH Procs only!

(just for completeness, not the recommended solution)

- GPLOT
- GCHART
- …

```
filename anim 'c:\tmp\Wurf1.gif';
goptions reset      = all
         device     = gifanim
         gsfname    = anim
         gsfmode    = replace|append
         delay      = 100 ;

 proc gchart data = ;
 quit;

filename anim clear;
```

Creating Animated GIF Images and SVG Documents

```
options printerpath=gif
        animation=start
        animduration=5
        animloop=yes
        noanimoverlay;

ods printer file='myfile.gif';

*** <your SAS statements>;

options printerpath=gif
        animation=stop;

ods printer close;
```

Tells SAS what type of file to put all of the graphs into.

Tells SAS to start the animation

Sets frame speed in seconds

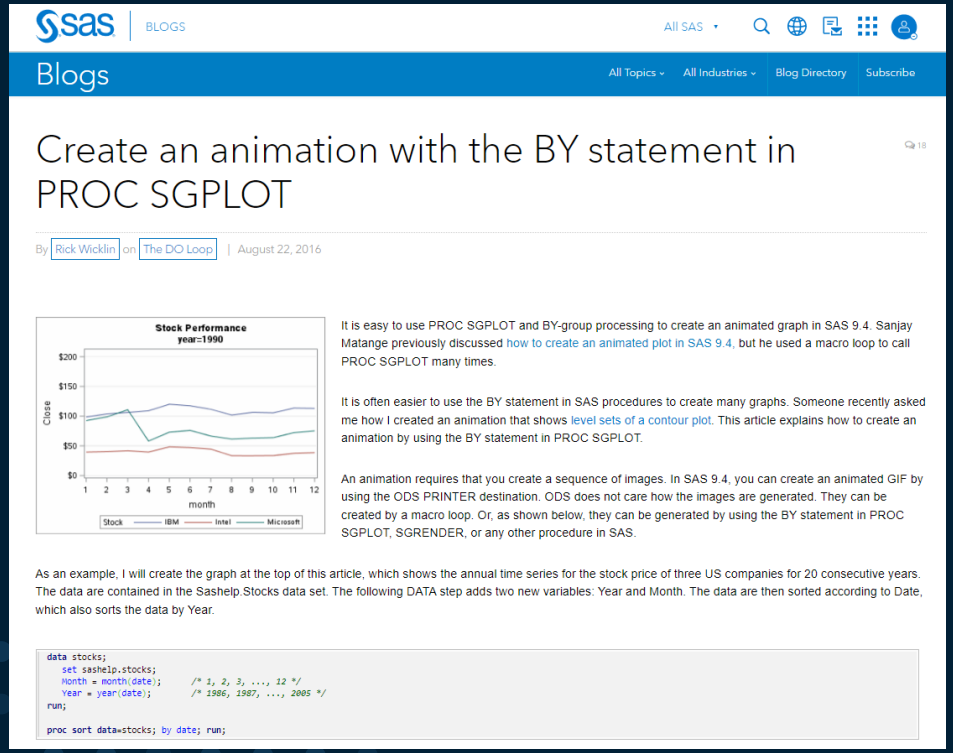
Loop the animation

Replace each graph with the next instead of overlaying

Specify the location of your GIF file

Tells SAS to stop the animation

Create an animation with the BY statement in PROC SGPLOT – Rick Wicklin



SAS BLOGS

All SAS

Blogs

All Topics | All Industries | Blog Directory | Subscribe

Create an animation with the BY statement in PROC SGPLOT

18

By Rick Wicklin on The DO Loop | August 22, 2016

It is easy to use PROC SGPLOT and BY-group processing to create an animated graph in SAS 9.4. Sanjay Matange previously discussed how to create an animated plot in SAS 9.4, but he used a macro loop to call PROC SGPLOT many times.

It is often easier to use the BY statement in SAS procedures to create many graphs. Someone recently asked me how I created an animation that shows level sets of a contour plot. This article explains how to create an animation by using the BY statement in PROC SGPLOT.

An animation requires that you create a sequence of images. In SAS 9.4, you can create an animated GIF by using the ODS PRINTER destination. ODS does not care how the images are generated. They can be created by a macro loop. Or, as shown below, they can be generated by using the BY statement in PROC SGPLOT, SGRENDER, or any other procedure in SAS.

As an example, I will create the graph at the top of this article, which shows the annual time series for the stock price of three US companies for 20 consecutive years. The data are contained in the Sashelp.Stocks data set. The following DATA step adds two new variables: Year and Month. The data are then sorted according to Date, which also sorts the data by Year.

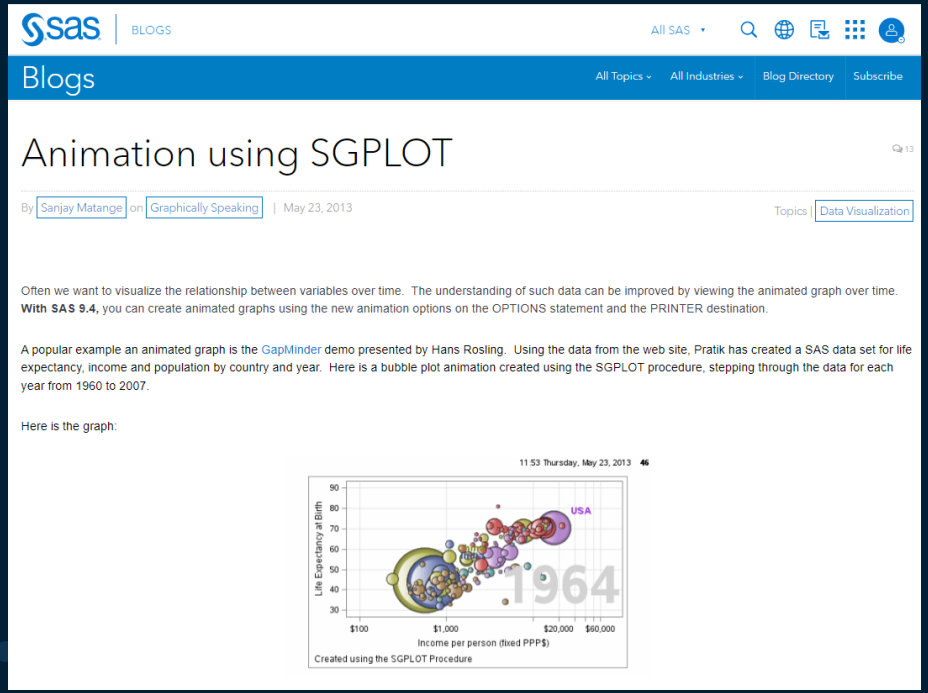
```
data stocks;
   set sashelp.stocks;
   Month = month(date);      /* 1, 2, 3, ..., 12 */
   Year = year(date);        /* 1986, 1987, ..., 2005 */
run;

proc sort data=stocks; by date; run;
```

- Link

Create an animation with the BY statement in PROC SGPLOT

```
data stocks;
   set sashelp.stocks;
   Month = month(date);      /* 1, 2, 3, ..., 12 */
   Year = year(date);        /* 1986, 1987, ..., 2005 */
run;

proc sort data=stocks; by date; run;

%let path = /home/autges/sasuser.viya;

ods graphics / imagefmt=GIF width=4in height=3in;     /* each image is 4in x 3in GIF */

options papersize=('4 in', '3 in')                    /* set size for images */
        nodate nonumber                               /* do not show date, time, or frame number */
        animduration=0.5 animloop=yes noanimoverlay   /* animation details */
        printerpath=gif animation=start;              /* start recording images to GIF */

ods printer file="&path./StockBYAnim.gif";            /* images saved into animated GIF */


proc sgplot data=stocks;
title "Stock Performance";
   by year;                                           /* create 20 images, one for each year */
   series x=month y=close / group=stock;              /* each image is a time series */
   xaxis integer values=(1 to 12);
   yaxis min=10 max=210 grid;                         /* set common vertical scale for all graphs */
run;


options printerpath=gif animation=stop;               /* stop recording images */

ods printer close;
```

Animation Content: BY Statements, Macro Loops, …

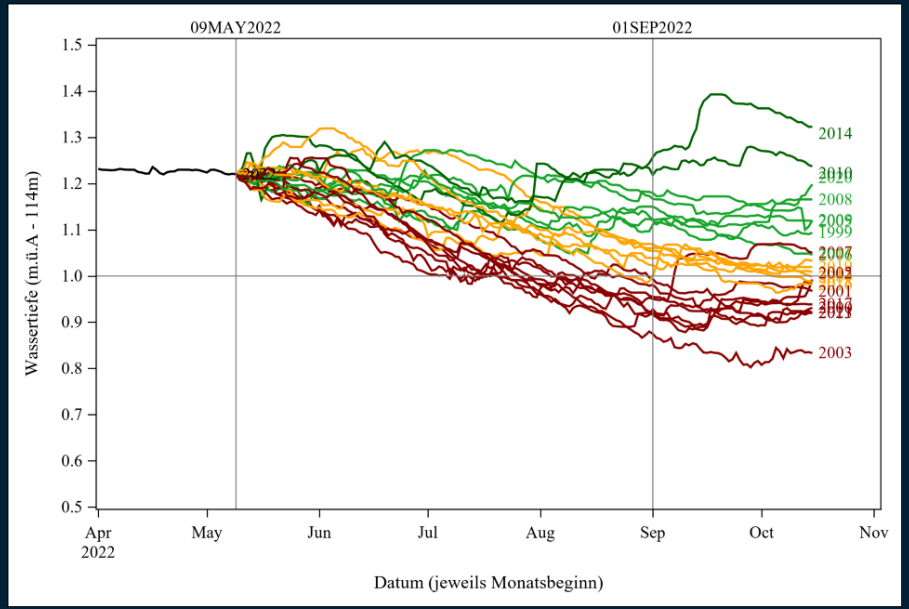



```
options papersize=('5 in', '3 in') printerpath=gif animation=start
        animduration=0.1 animloop=yes noanimoverlay;
ods printer file='GapMinder.gif';

ods graphics / width=5in height=3in imagefmt=GIF;
  *Create multiple graphs using SGPLOT procedure;

options printerpath=gif animation=stop;
ods printer close;

/*--This program requires SAS 9.4--*/
%macro GapMinder(start=, end=, incr=);
  %do year=&start %to &end %by &incr;

    proc sgplot data=GapMinder noautolegend;
      format gdp dollar12.0;
      footnote j=l 'Created using the SGPLOT Procedure';
      where yearnum=&year;
      format name $name.;
```

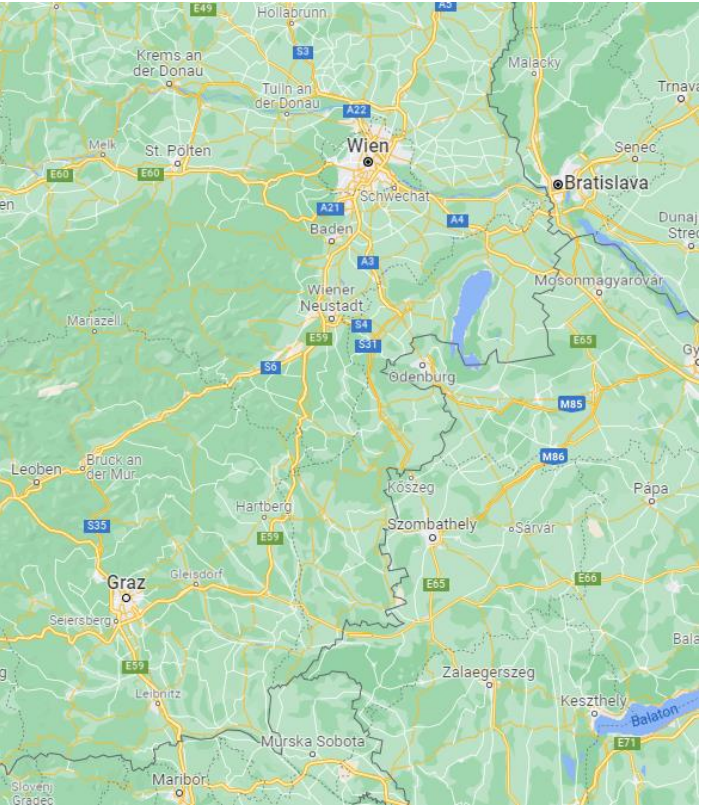
- Link

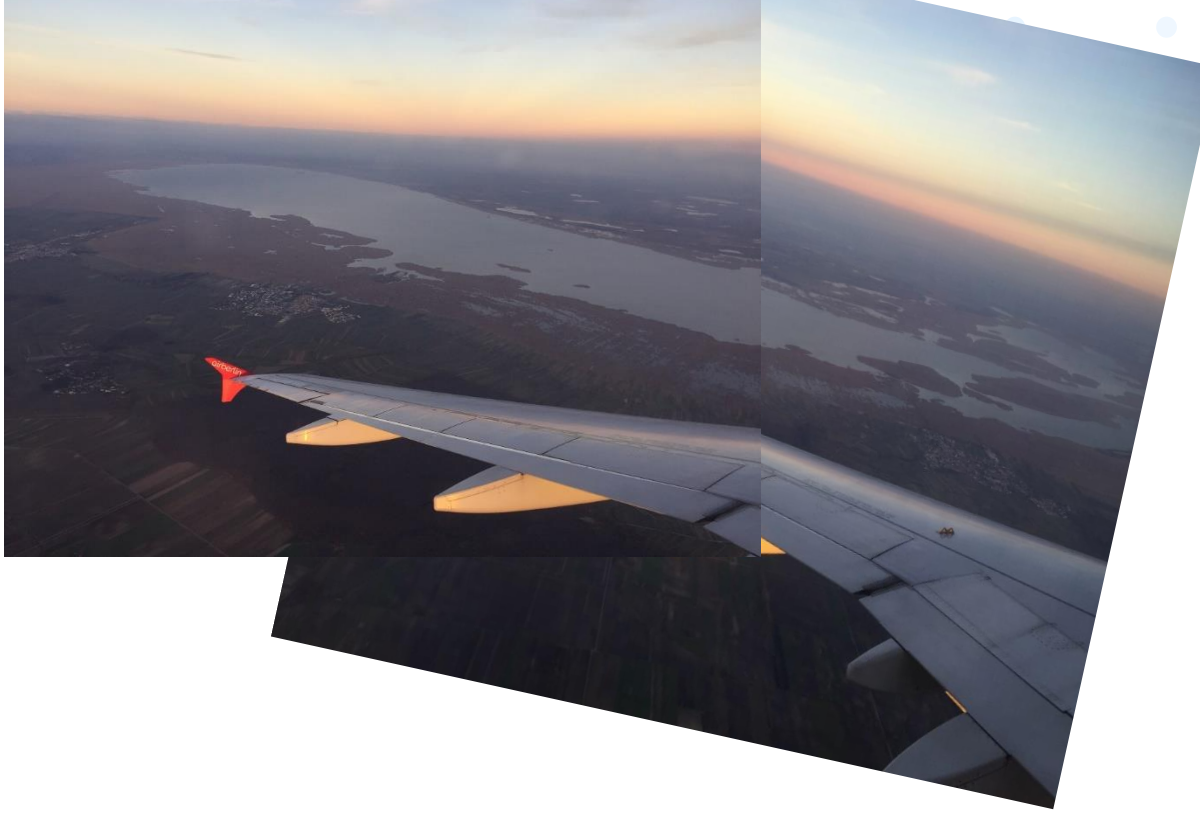
Das erwartet Sie in diesem Vortrag

-2-
Deep Dive in ein komplexeres Beispiel



Lake Neusiedl
Burgenland, Austria
47°50′N 16°45′E

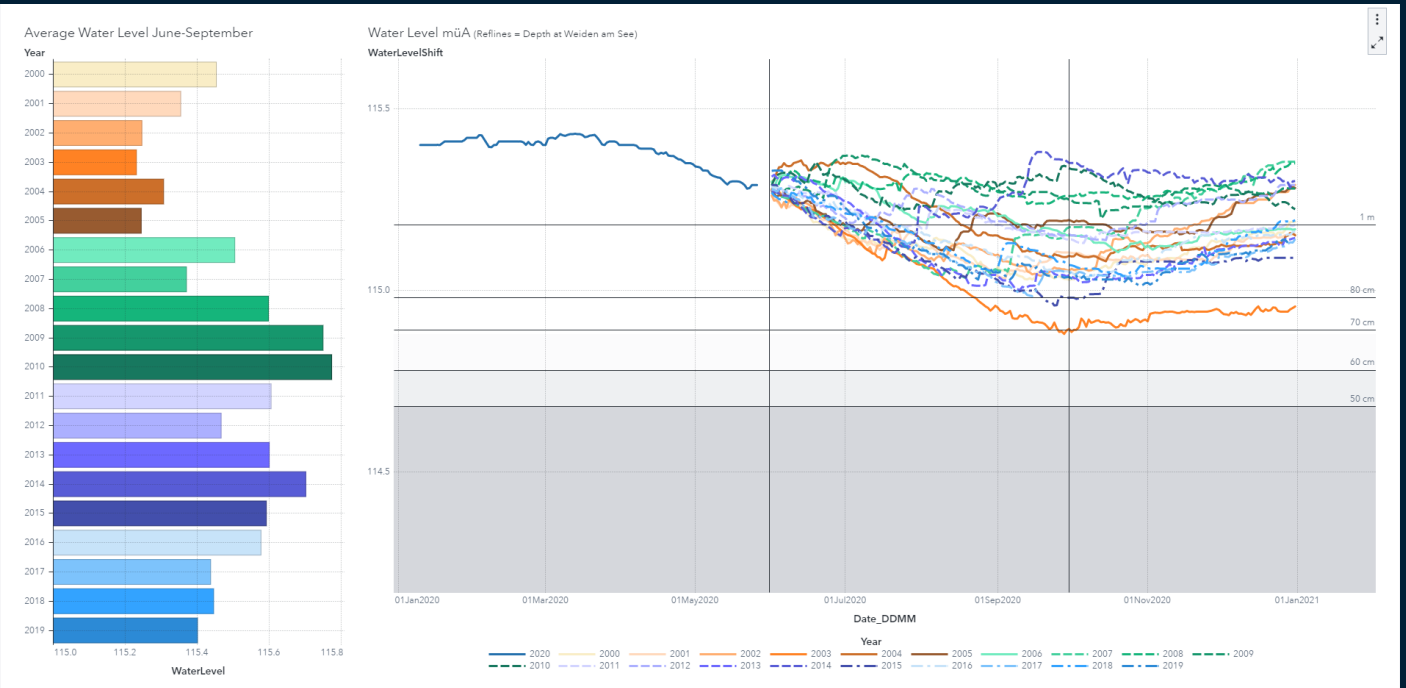


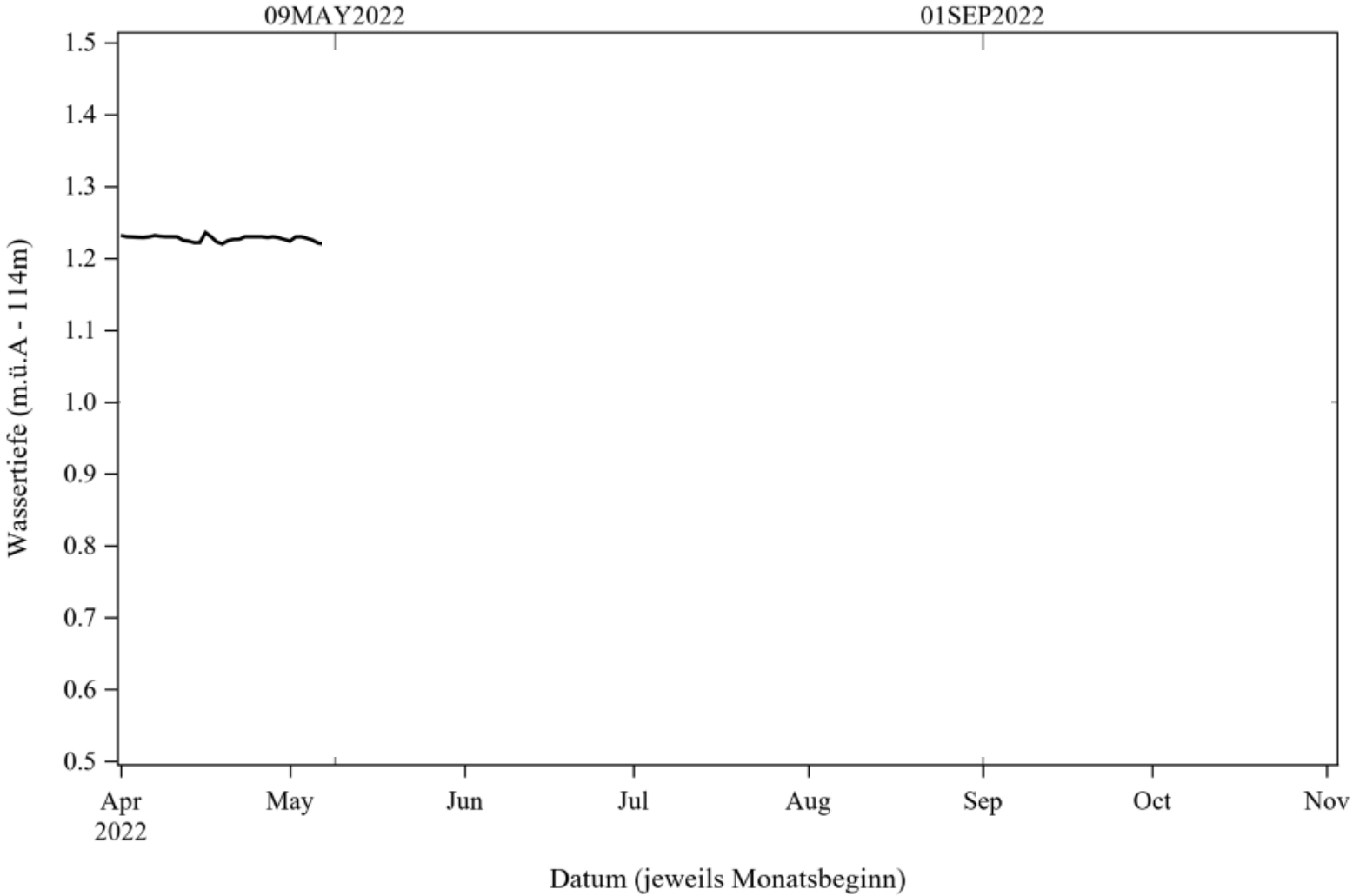


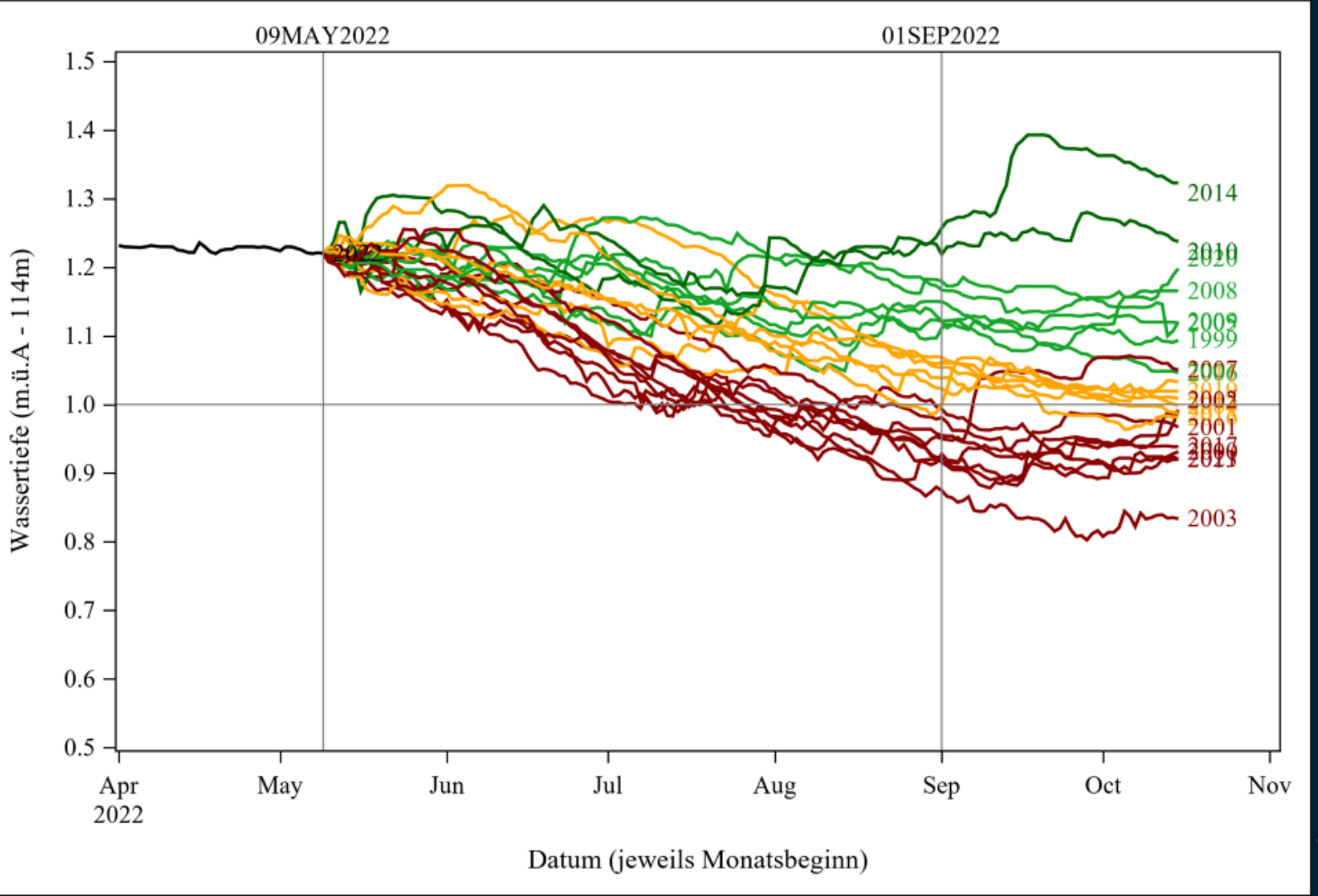
© Gerhard Svolba 2015

Historically Lowest Level at Lake Neusiedl

What if, the weather in the summer half-year is like ...
- in the extremely dry year 2003,
- in the rainy year 2014,
- ...




09MAY2022
01SEP2022
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
Wassertiefe (m.ü.A - 114m)
Apr
2022
May
Jun
Jul
Aug
Sep
Oct
Nov
Datum (jeweils Monatsbeginn)


09MAY2022
01SEP2022
Wassertiefe (m.ü.A - 114m)
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
2014
2010
2020
2008
2009
1999
2007
2006
2002
2001
2003
Apr
2022
May
Jun
Jul
Aug
Sep
Oct
Nov
Datum (jeweils Monatsbeginn)

Calculation Procedure for one graph

1. Prepare a repository of daily water levels for each (historic) year
2. Select a cutoff date (most likely the actual date)
3. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date

Calculation Procedure (Code Examples)

1. Prepare a repository of daily water levels for each (historic) year
2. Select a cutoff date (most likely the actual date)
3. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date

```
data hydro3.Daily_WaterLevel;
Format Datum date9. Date_DDMM date5.;
 set work.WaterLevel_Till2018Oct15(where=(Datum <= "15OCT2018"d))
     work.WaterLevel_from2018Oct15(where=(Datum >  "15OCT2018"d));
Format Date_DDMM date5.;
Date_DDMM = mdy(month(Datum),day(Datum),&current_Year.);
Year = year(Datum);
run;

proc sort data=hydro3.Daily_WaterLevel;
 by Datum;
run;
```

Calculation Procedure (Code Examples)

1. Prepare a repository of daily water levels for each (historic) year
2. Select a cutoff date (most likely the actual date)
3. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date

```
%let CutDate = %sysfunc(date(),date9.);

data _null_;
 call symput ("cut_year",year("&CutDate"d));
 call symput ("cut_month",month("&CutDate"d));
 call symput ("cut_day",day("&CutDate"d));
run;
```

Calculation Procedure (Code Examples)

1. Prepare a repository of daily water levels for each (historic) year
2. Select a cutoff date (most likely the actual date)
3. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date

```
proc sql noprint;
 *** Level at CutDate;
 select WaterLevel
 into :WaterLevel_AtCutDate
 from hydro3.Daily_WaterLevel
 where datum="&CutDate"d;
 *** Level at CutDate DDMM per year;
 quit;
```

```
data work.Reference_Days;
 set work.Reference_Days;
 format WaterLevel_Ref WaterLevel_Diff WaterLevel_CutYear Wassertiefe Wasse
 retain WaterLevel_Ref;
 if (year < &cut_year and day(datum) =&cut_day. and month(datum) = &cut_mon
        WaterLevel_Ref = WaterLevel;
    end;
 WaterLevel_Diff = WaterLevel-WaterLevel_Ref;
 WaterLevel_CutYear =  &WaterLevel_AtCutDate. + WaterLevel_Diff;
 if year = &cut_year then WaterLevel_CutYear = WaterLevel;
 Month = month(datum);
 Wassertiefe = WaterLevel_CutYear - 114;
 Wassertiefe_Hafen = WaterLevel_CutYear - 114.18;
run;
```

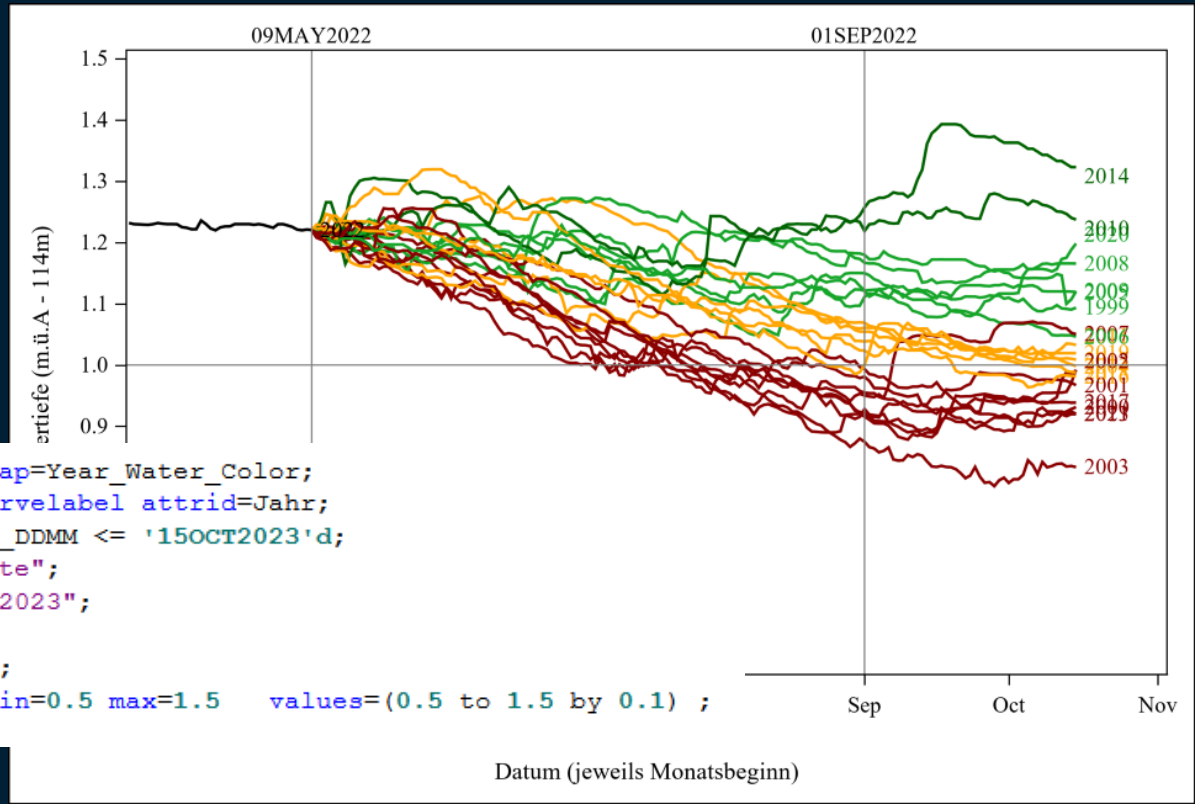
Calculation Procedure (Code Examples)

1. Prepare a repository of daily water levels for each (historic) year
2. Select a cutoff date (most likely the actual date)
3. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date

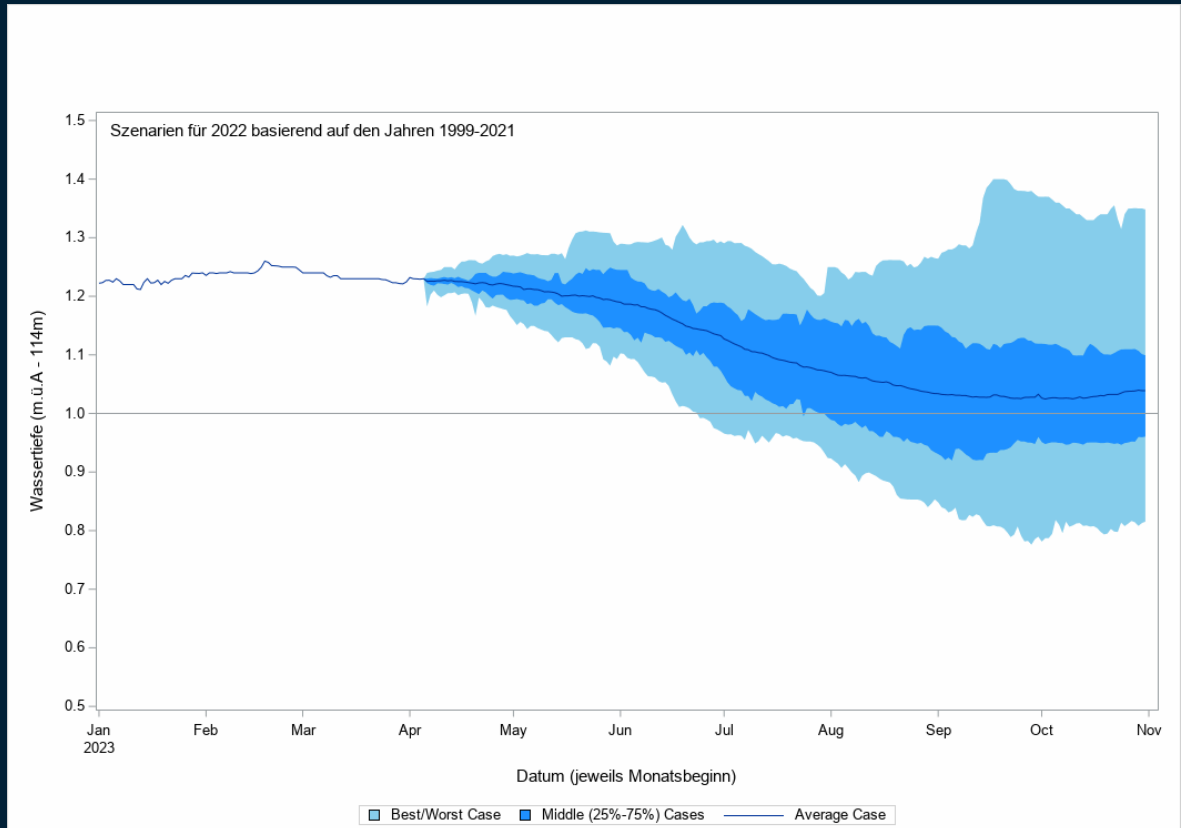
```
data work.Reference_Days;
 set hydro3.Daily WaterLevel;
 if (year = &cut_year and datum <= "&CutDate."d) then output;
 if (year < &cut_year and day(datum) >=&cut_day. and month(datum) = &cut_month.) then output;
 if (year < &cut_year  and &cut_month. < month(datum) <= 10) then output;
run;
```

Use the SPLOT Procedure to produce the line chart

```
proc sgplot data=work.reference_days dattrmap=Year_Water_Color;
 series x=Date_DDMM y=&var. / group=Year curvelabel attrid=Jahr;
 where year >=1999 and '01APR2023'd <= Date_DDMM <= '15OCT2023'd;
 refline "&CutDate"d / axis=x label="&cutdate";
 refline "01SEP2023"d / axis=x label="01SEP2023";
 refline 1 / axis=y ;
 xaxis label="Datum (jeweils Monatsbeginn)";
 yaxis label="Wassertiefe (m.ü.A - 114m)" min=0.5 max=1.5   values=(0.5 to 1.5 by 0.1) ;
run;
```

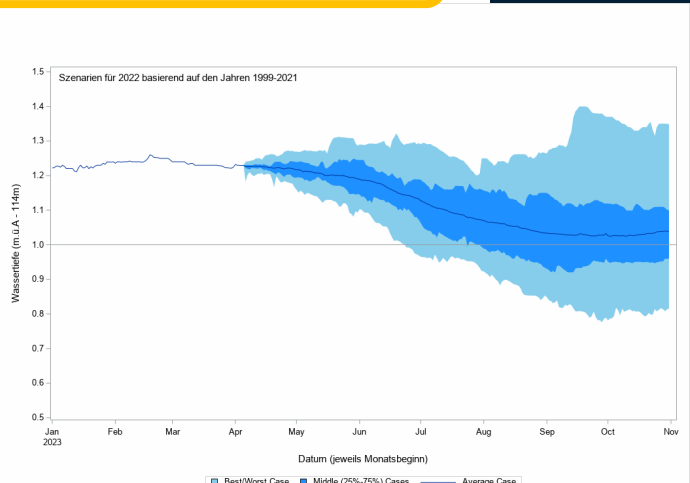


Use the SPLOT Procedure to produce the quantile plot



Use the SPLOT Procedure to produce the quantile plot

```
proc means data=work.reference_days noprint nway;
 var &var.;
 class Date_DDMM ;
 output out=work.ref_days_quantiles max= p99= p95= p75= median= mean= p25= p5= p1= min= / autoname;
run;

proc sgplot data=work.ref_days_quantiles;
 band lower=&var._max upper=&var._min x=Date_DDMM  / fillattrs=(color=skyblue) legendlabel="Best/Worst Case";
 band lower=&var._p25 upper=&var._p75 x=Date_DDMM / fillattrs=(color=dodgerblue) legendlabel="Middle (25%-75%) Cases";
 series y=&var._Mean x=Date_DDMM / legendlabel="Average Case";
 where '01APR2023'd <= Date_DDMM <= '15OCT2023'd;
 refline "&CutDate"d / axis=x label="&cutdate";
 refline "01SEP2023"d / axis=x label="01SEP2023";
 refline 1 / axis=y ;
 xaxis label="Datum (jeweils Monatsbeginn)";
 yaxis label="Wassertiefe (m.ü.A - 114m)" min=0.5 max=1.5    values=(0.5 to 1.5 by 0.1)
run;
```



Creating an animation of the development from April1st – Sep1st

We have full flexibility with our SAS Code!

```
options printerpath=gif
        animation=start
        animduration=5
        animloop=yes
        noanimoverlay;

ods printer file='myfile.gif';

*** <your SAS statements>;

options printerpath=gif
        animation=stop;

ods printer close;
```

Tells SAS what type of file to put all of the graphs into.

Tells SAS to start the animation

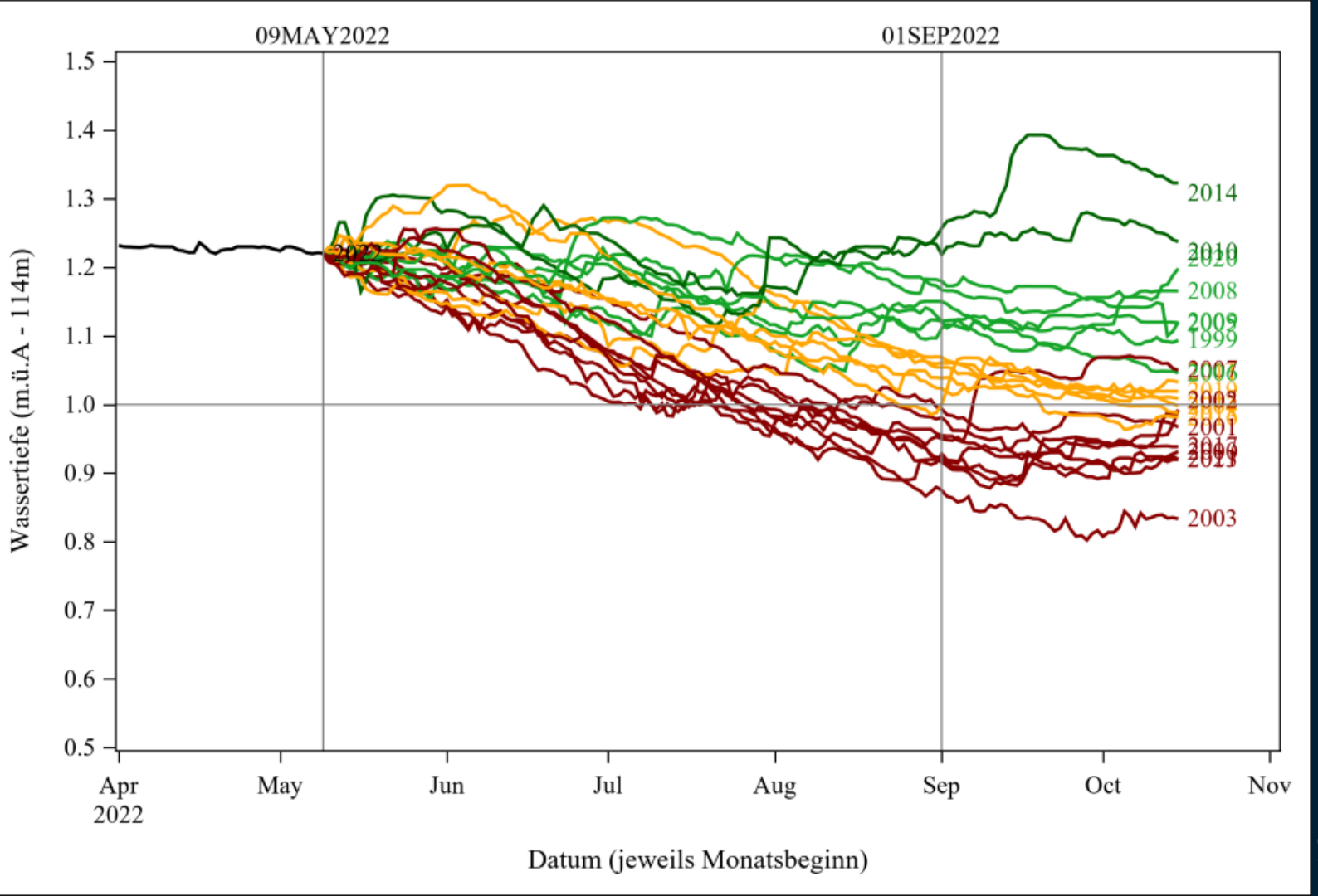
Sets frame speed in seconds

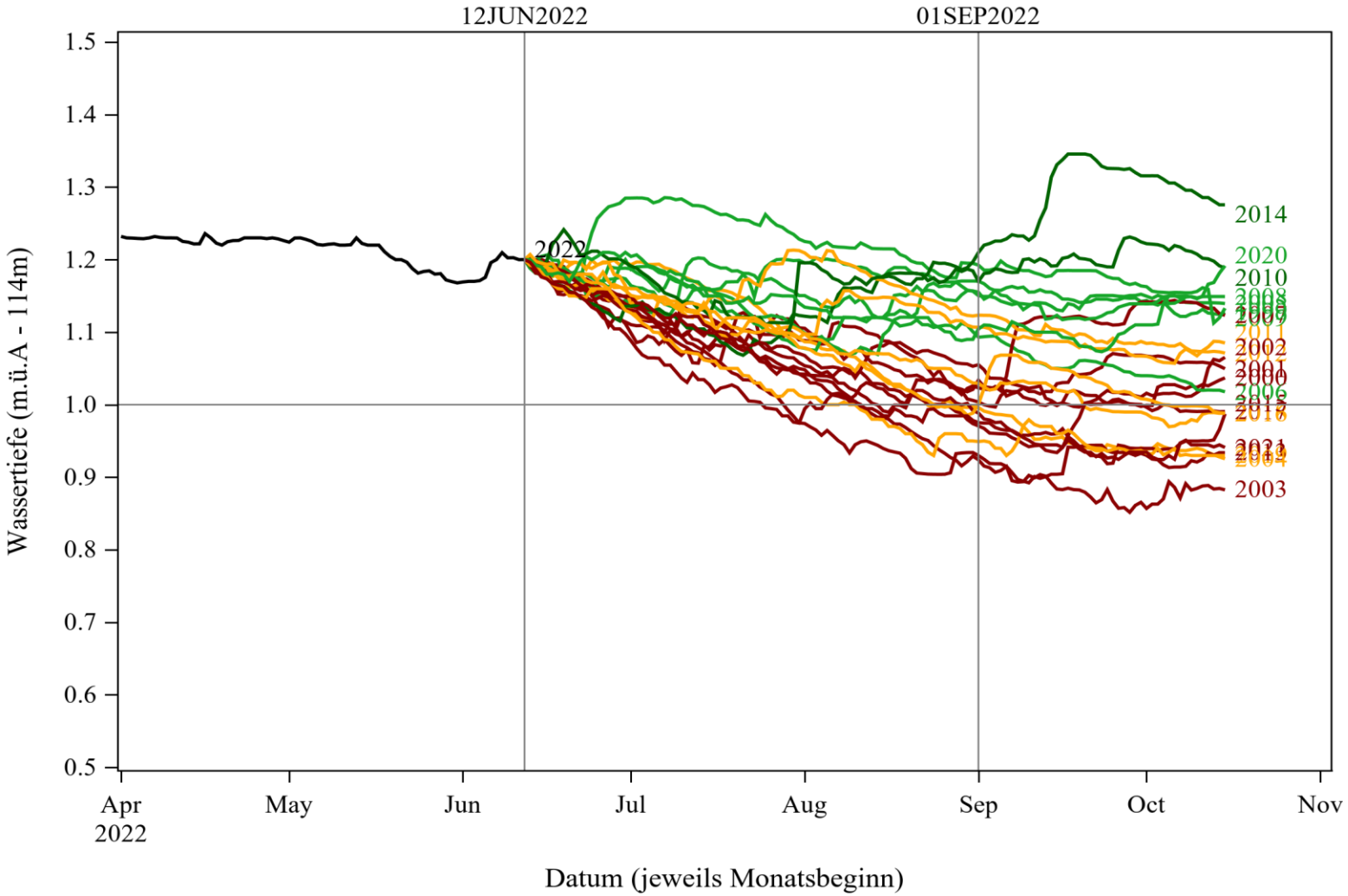
Loop the animation

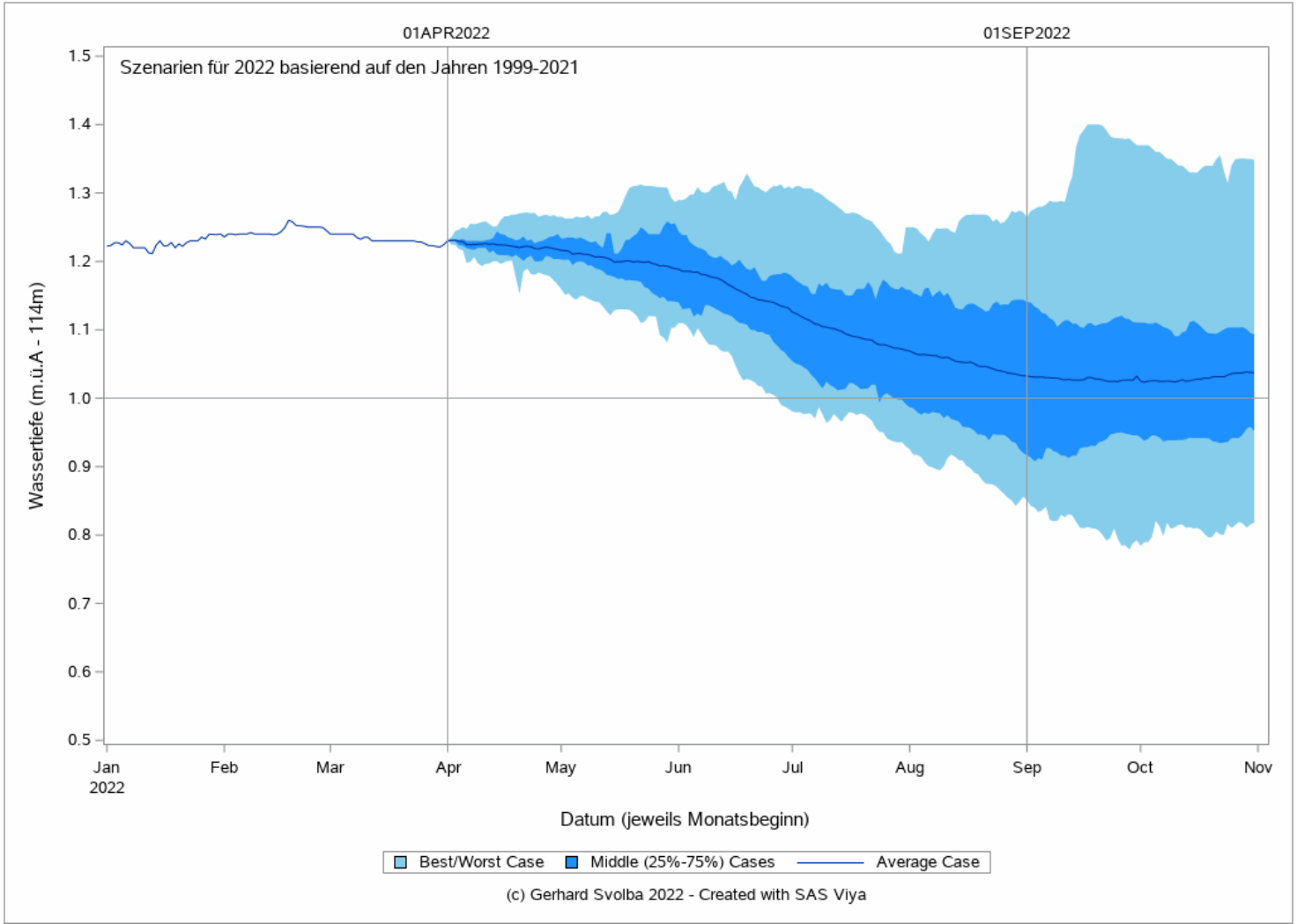
Replace each graph with the next instead of overlaying

Specify the location of your GIF file

Tells SAS to stop the animation


09MAY2022
01SEP2022
Wassertiefe (m.ü.A - 114m)
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
2014
2010
2020
2008
2009
1999
2007
2006
2002
2001
2003
Apr
2022
May
Jun
Jul
Aug
Sep
Oct
Nov
Datum (jeweils Monatsbeginn)


12JUN2022
01SEP2022
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
Wassertiefe (m.ü.A - 114m)
2022
2014
2020
2010
2002
2003
Apr
2022
May
Jun
Jul
Aug
Sep
Oct
Nov
Datum (jeweils Monatsbeginn)


01APR2022
01SEP2022
Szenarien für 2022 basierend auf den Jahren 1999-2021
Wassertiefe (m.ü.A - 114m)
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
Jan 2022
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Datum (jeweils Monatsbeginn)
Best/Worst Case
Middle (25%-75%) Cases
Average Case
(c) Gerhard Svolba 2022 - Created with SAS Viya

Calculation Procedure for one graph

1. Prepare a repository of daily water levels for each (historic) year
2. Select a cutoff date (most likely the actual date)
3. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date

Calculation Procedure for a series of graphs

1. Prepare a repository of daily water levels for each (historic) year
2. Define a SAS Macro

```
%macro Water_Level_Anim(from,till,anim_duration=0.5);
```

3. Initialize the animation
4. Use each day of the analysis period (e.g. April 1st to September 1st) as cutoff
5. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date
6. Stop the animation
7. Call the SAS Macro

Calculation Procedure for a series of graphs

1. Prepare a repository of daily water levels for each (historic) year
2. Define a SAS Macro
3. Initialize the animation

```
options papersize=('28 cm', '20 cm') printerpath=gif animation=start
        animduration=&anim_duration. animloop=yes noanimoverlay;
ods printer file='c:\tmp\gif\WaterLevel_Quantile.gif';
ods graphics /height=20cm width=28cm imagefmt=GIF;;
```

4. Use each day of the analysis period (e.g. April 1st to September 1st) as cutoff
5. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date
6. Stop the animation
7. Call the SAS Macro

Calculation Procedure for a series of graphs

1. Prepare a repository of daily water levels for each (historic) year
2. Define a SAS Macro
3. Initialize the animation

```
data _null_;
 call symput ("AnimDays","&till."d - "&from."d);
run;


%do i = 0 %to &AnimDays;
```

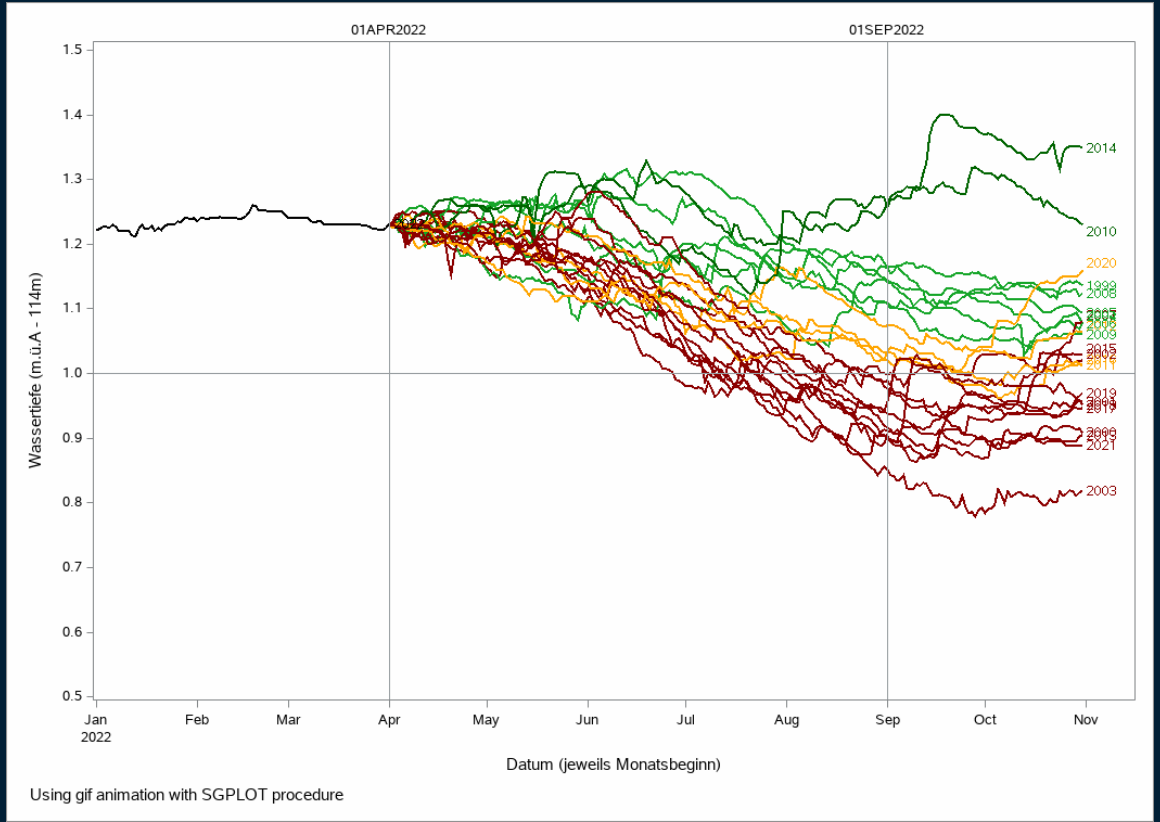
4. Use each day of the analysis period (e.g. April 1st to September 1st) as cutoff
5. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date
6. Stop the animation
7. Call the SAS Macro

Calculation Procedure for a series of graphs

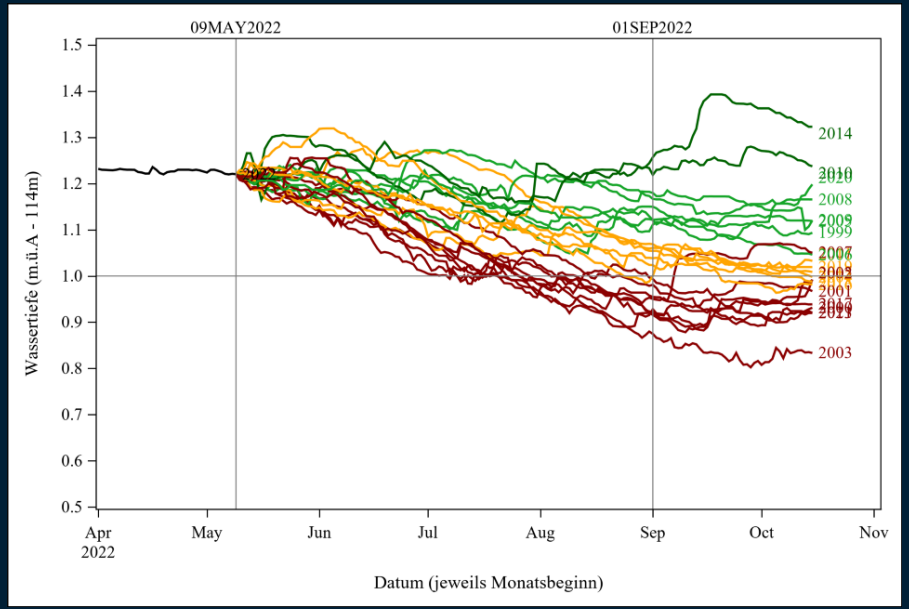
1. Prepare a repository of daily water levels for each (historic) year
2. Define a SAS Macro
3. Initialize the animation
4. Use each day of the analysis period (e.g. April 1st to September 1st) as cutoff
5. For each historic year in the repository
   a) Calculate the difference at the cutoff date between the actual value and the value of the historic year
   b) Shift the values of the historic year
   c) Concatenate the values of the actual year UNTIL the cutoff date with the shifted values of the historic year AFTER the cutoff date
6. Stop the animation
7. Call the SAS Macro

```
%end; *** Do Loop;
options printerpath=gif animation=stop;
ods printer close;

%mend Water_Level_Anim;

%Water_Level_Anim(01APR2022,14APR2022,anim_duration=0.25);
```

Observe how the range of possible water levels changes over the year



Das erwartet Sie in diesem Vortrag

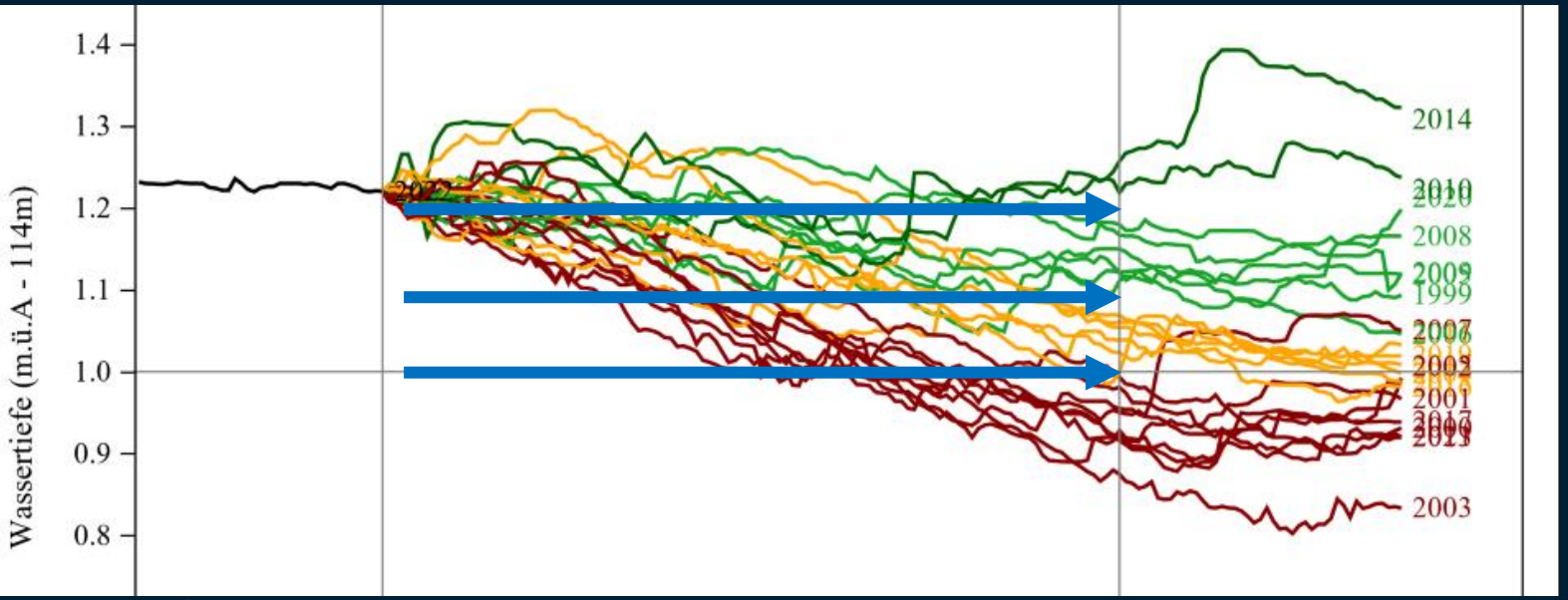


-3-

Dynamische Formatierung von Graphikelementen (z.B. Linienfarbe)

Define line colors based on Scenario Water Level on Sep1st

- Limits at 1.0 meters, 1.1 meters, 1.2 meters



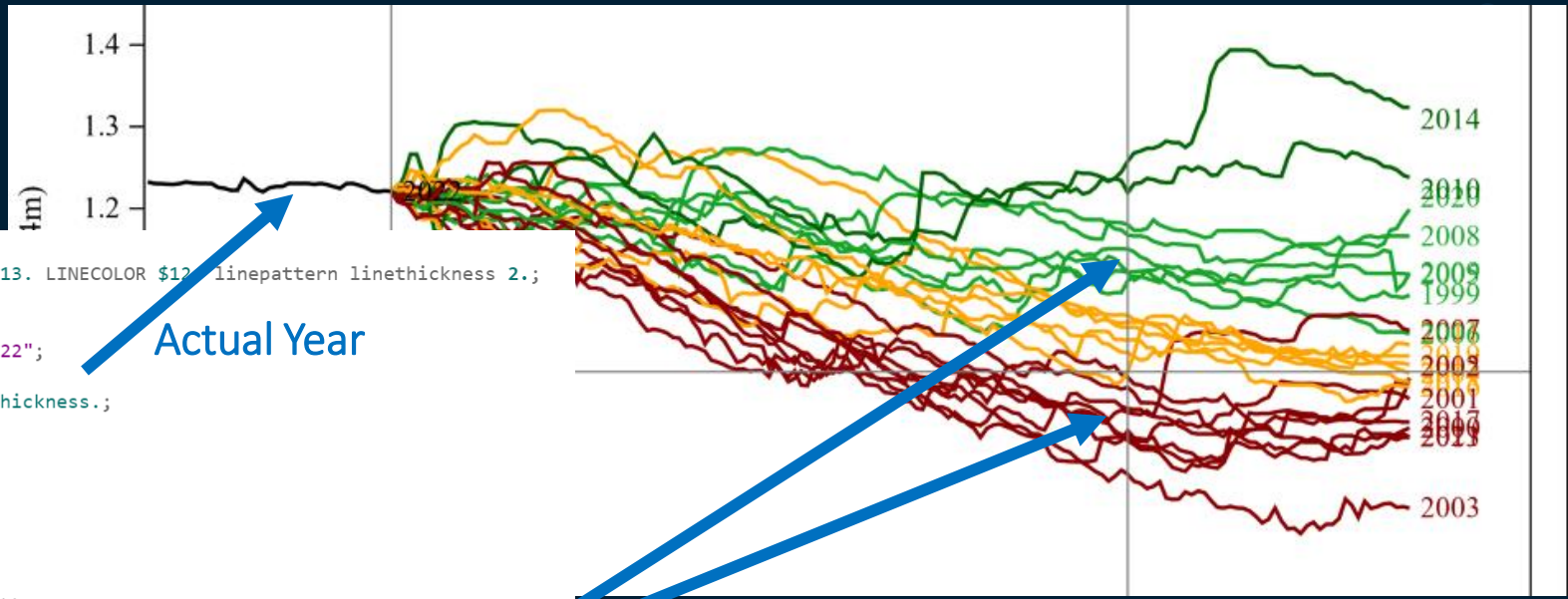
Defining an attribute map

```
data Year_Water_Color;
      format id $10.  value $13. LINECOLOR $12. linepattern linethickness 2.;

      if _N_ = 1 then do;
         id="Jahr"; value="2022";
         LINECOLOR="black";
         linethickness=&linethickness.;
         linepattern=1;
         output;
      end;

      set year_water_color;

      id="Jahr";
      value=strip(put(year,8.));

      linepattern=&linepattern;
      linethickness=&linethickness;
           if Wassertiefe = .   then linecolor="blue";
      else if Wassertiefe < 1.0 then linecolor="darkred";
      else if Wassertiefe < 1.1 then linecolor="orange";
      else if Wassertiefe < 1.2 then linecolor="viyg";
      else                           linecolor="CX" || put(0,hex2.) || put(100,h

 output;
run;
```

Actual Year

	id	value	LINECOLOR	linepattern	linethickness	Year	Wassertiefe
1	Jahr	2022	black	1	2	.	.
2	Jahr	1999	orange	1	2	1999	1.09
3	Jahr	2000	darkred	1	2	2000	0.97
4	Jahr	2001	darkred	1	2	2001	0.94
5	Jahr	2002	orange	1	2	2002	1.01
6	Jahr	2003	darkred	1	2	2003	0.86
7	Jahr	2004	orange	1	2	2004	1.04
8	Jahr	2005	viyg	1	2	2005	1.12
9	Jahr	2006	viyg	1	2	2006	1.11
10	Jahr	2007	darkred	1	2	2007	0.94
11	Jahr	2008	viyg	1	2	2008	1.18
12	Jahr	2009	viyg	1	2	2009	1.18

Defining an attribute map

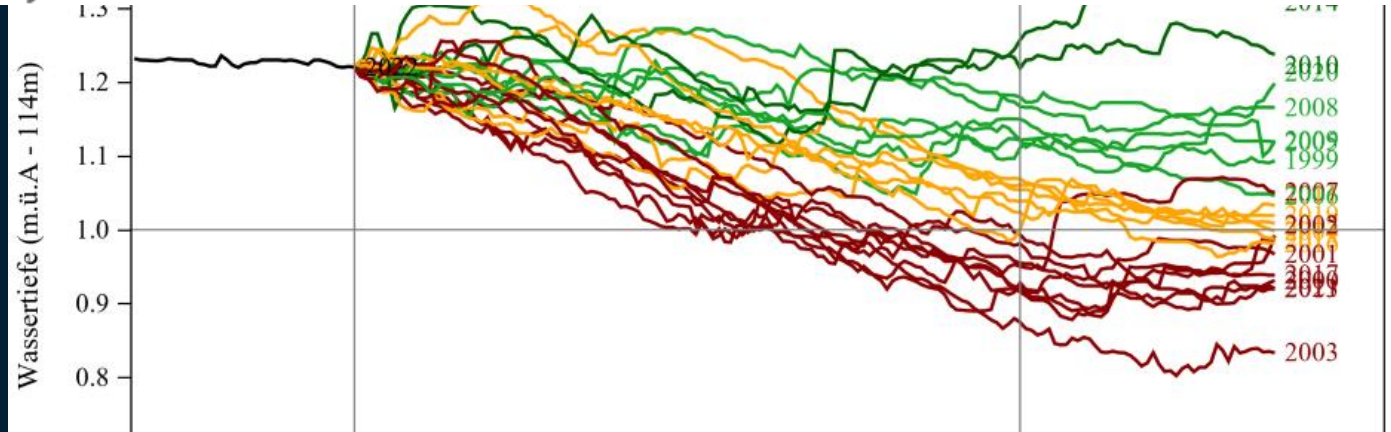


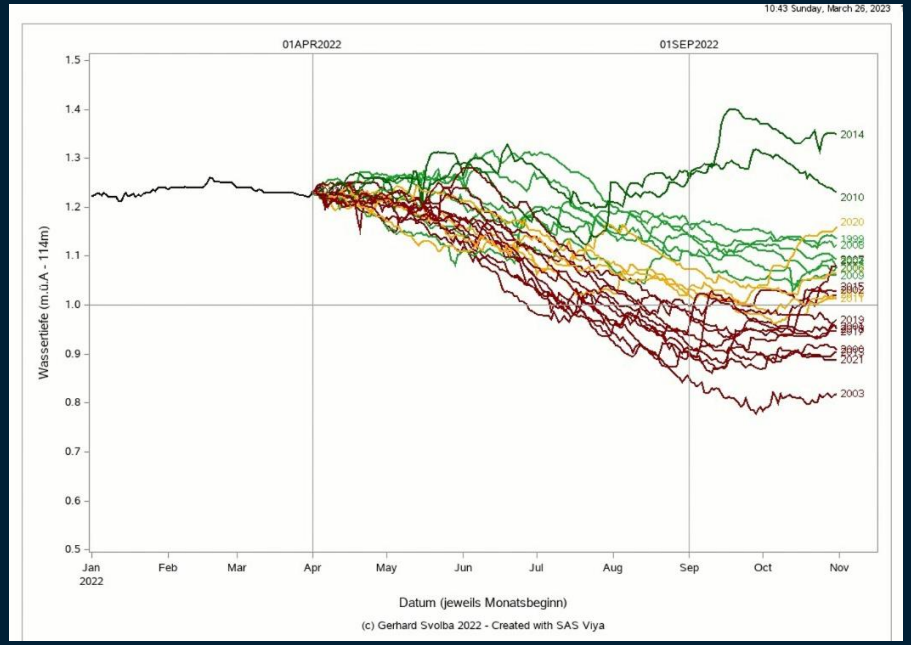

```
data Year_Water_Color;
      format id $10.  value $13. LINECOLOR $12.  linepattern linethickness 2.;

      if _N_ = 1 then do;
         id="Jahr"; value="2022";
         LINECOLOR="black";
         linethickness=&linethickness.;
         linepattern=1;
         output;
      end;

      set year_water_color;

      id="Jahr";
      value=strip(put(year,8.));

      linepattern=&linepattern;
      linethickness=&linethickness;
           if Wassertiefe = .   then linecolor="blue";
      else if Wassertiefe < 1.0 then linecolor="darkred";
      else if Wassertiefe < 1.1 then linecolor="orange";
      else if Wassertiefe < 1.2 then linecolor="viyg";
      else                           linecolor="CX" || put(0,hex2.) || put(100,h

 output;
run;
```

Actual Year

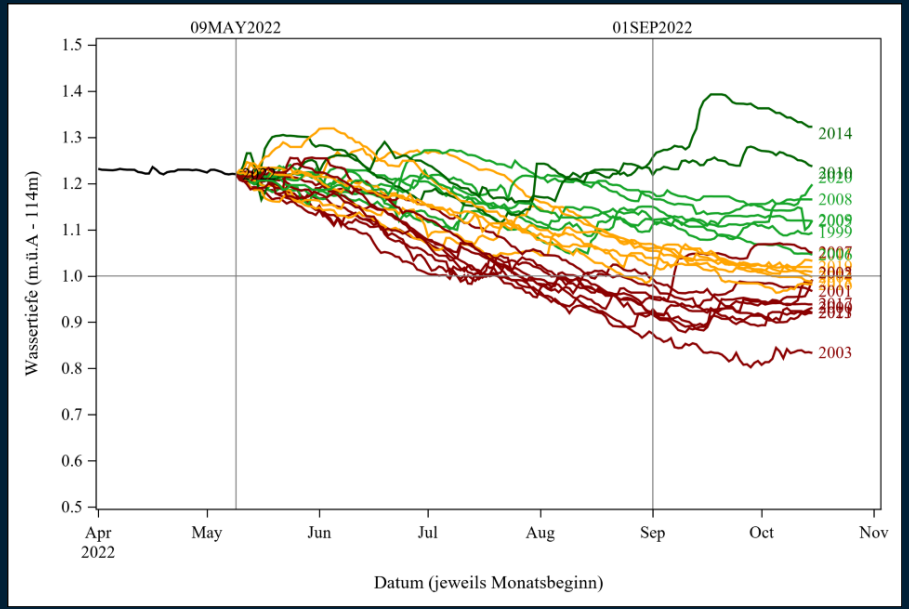
Using the attribute map

```
proc sgplot data=work.reference_days dattrmap=Year_Water_Color;
 series x=Date_DDMM y=&var. / group=Year curvelabel attrid=Jahr;
 where year >=1999 and Date_DDMM <= '31OCT2022'd;
 refline "&CutDate"d / axis=x label="&cutdate";
 refline "01SEP2022"d / axis=x label="01SEP2022";
 refline 1 / axis=y ;
 xaxis label="Datum (jeweils Monatsbeginn)";
 yaxis label="Wassertiefe (m.ü.A - 114m)" min=0.5 max=1.5   values=(0.5 to 1.5 by 0.1) ;
run;
```





Das erwartet Sie in diesem Vortrag



-4-

Do’s and Don’ts bei der Erstellung von animierten Graphiken

Tipp #1

Freeze the scaling of the x-axis and the y-axis!

```
proc sgplot data=stocks;
title "Stock Performance";
   by year;
   series x=month y=close / group=stock;
   xaxis integer values=(1 to 12);
   yaxis min=10 max=210 grid;
run;
```

```
proc sgplot data=work.reference_days dattrmap=Year_Water_Color;
 series x=Date_DDMM y=&var. / group=Year curvelabel attrid=Jahr;
 where year >=1999 and Date_DDMM <= '31OCT2022'd;
 refline "&CutDate"d / axis=x label="&cutdate";
 refline "01SEP2022"d / axis=x label="01SEP2022";
 refline 1 / axis=y ;
 xaxis label="Datum (jeweils Monatsbeginn)";
 yaxis label="Wassertiefe (m.ü.A - 114m)" min=0.5 max=1.5   values=(0.5 to 1.5 by 0.1) ;
run;
```

Tipp #2

Carefully choose the speed of your animation!

Avoid boring or overwhelming your audience

```
options printerpath=gif
        animation=start
        animduration=5
        animloop=yes
        noanimoverlay;
```

Tipp #3

(before you give up, because your GIFs don’t move):
Make sure that your graphics viewer supports animated GIF!

Summary

- SAS output can easily be converted into animated GIFs
- High flexibility for BY processing, macro loops, …
  “anything you want to put into your analysis”
- Attribute maps can used to make you graphs better visible and interpetabale

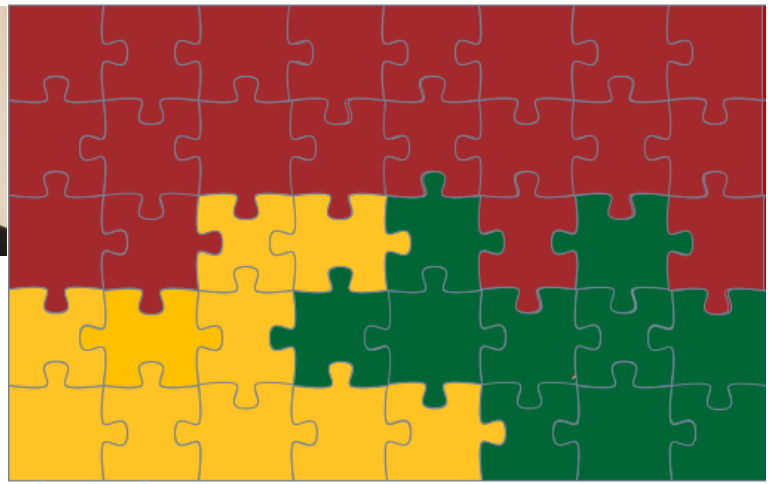
Links

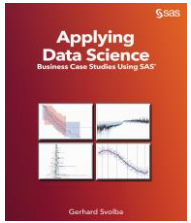
- Ein Beitrag mit Beispiel Code zum Thema dieses Vortrags ist in Vorbereitung. Der Link findet sich dann in dieser Sammlung - Data Science and Data Preparation Article Overview by Gerhard
  - https://communities.sas.com/t5/SAS-Communities-Library/Data-Science-and-Data-Preparation-Article-Overview-by-Gerhard/ta-p/727875
- Using Animation to Make Statistical Graphics Come to Life, Jesse Pratt, MWSUG 2016, Paper DV1
  - https://www.mwsug.org/proceedings/2016/DV/MWSUG-2016-DV01.pdf
- Create an animation with the BY statement in PROC SGPLOT
  - https://blogs.sas.com/content/iml/2016/08/22/animation-by-statement-proc-sgplot.html
- Animation Using SGPLOT
  - https://blogs.sas.com/content/graphicallyspeaking/2013/05/23/animation-using-sgplot/
- Der Wasserstand am Neusiedler See im Jahr 2022 — Sichtweisen eines Seglers und Statistikers
  - https://medium.com/@gerhard-svolba/der-wasserstand-am-neusiedler-see-im-jahr-2022-sichtweisen-eines-seglers-und-statistikers-9c33059f225e

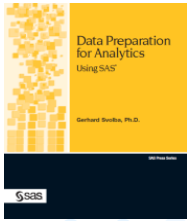
Data Preparation for Data Science

- Data Assembly
- Data Quality for Analytics
- Feature Generation

Gerhard Svolba,
Data Scientist @SAS
mailto:sastools.by.gerhard@gmx.net







Articles and Blogs — Medium, LinkedIn

Webinars — YouTube

Tipps & Tricks — sas COMMUNITIES

Macros & Downloads