

SAS Club

Session: Tipps und Tricks



Kashif Din
Sr. Technical Architect

5. November 2020

Tipps und Tricks

Umgang mit beweglichen Feiertagen



Automatisches Erstellen von Referenzlinien



Weiterführende Links und Beispiele





Beispiel 1

Georg und der Besuch seiner Schwiegereltern



Georg und der Besuch seiner Schwiegereltern

- Zu Ostern: Karfreitag bis Ostersonntag 
- Zu Weihnachten: Samstag vor dem 24.Dez bis 26.Dezember 

Kürzeste Dauer: 4 Tage

wenn der 24.Dez -> Sonntag, Sa, 23.Dez – Di, 26.Dez



Längste Dauer: 10 Tage

wenn der 24.Dez -> Samstag, Sa, 17.Dez – Mo, 26.Dez



Anzahl der Tage zwischen Ostern und Weihnachten relevant

Georg und der Besuch seiner Schwiegereltern

SAS Funktionen

- Intck() „**INT**erval **C**heck“

years=intck('year','01jan2009'd,'01jan2010'd);	->	years=1
weeks=intck('week','01jan2009'd,'01jan2010'd);	->	weeks=52
days=intck('day','01jan2009'd,'01jan2010'd);	->	days=365

- Holiday()

holiday('EASTER',2019)	->	21APR2019
------------------------	----	-----------

- Intnx() „**INT**erval **N**e**X**t“











day=intnx('day2', '01FEB2010'd, 2);	->	05FEB2010
week=intnx('week1.3', '01FEB2010'd, 1);	->	02FEB2010
year=intnx('year1.3', '01FEB2010'd, 1);	->	01MAR2011

Georg und der Besuch seiner Schwiegereltern

```
data family_in_law_free_interval;
  format Year          8.
         EasterSunday  date9.
         SatBefore2412 weekdatx.;
  do Year = 2020 to 2035;
    SatBefore2412 = intnx('week.7',mdy(12,23,year),0);
    EasterSunday  = holiday('EASTER',year);
    RelaxingDays  = SatBefore2412-EasterSunday;
    RelaxingWeeks = intck('WEEK',EasterSunday,SatBefore2412)+1;
    DaysXmas      = mdy(12,26,year) - SatBefore2412 + 1;
  output;
end;
run;
```

„week.7“ -> Zählung der Woche beginnt mit Samstag

Georg und der Besuch seiner Schwiegereltern

 Year	 EasterSunday	 SatBefore2412	 RelaxingDays	 RelaxingWeeks	 DaysXmas
2020	12APR2020	Saturday, 19 December 2020	251	36	8
2021	04APR2021	Saturday, 18 December 2021	258	37	9
2022	17APR2022	Saturday, 17 December 2022	 244	35	 10
2023	09APR2023	Saturday, 23 December 2023	258	37	 4
2024	31MAR2024	Saturday, 21 December 2024	 265	38	6
2025	20APR2025	Saturday, 20 December 2025	244	35	7
2026	05APR2026	Saturday, 19 December 2026	258	37	8
2027	28MAR2027	Saturday, 18 December 2027	265	38	9
2028	16APR2028	Saturday, 23 December 2028	251	36	4
2029	01APR2029	Saturday, 22 December 2029	265	38	5
2030	21APR2030	Saturday, 21 December 2030	244	35	6
2031	13APR2031	Saturday, 20 December 2031	251	36	7
2032	28MAR2032	Saturday, 18 December 2032	265	38	9
2033	17APR2033	Saturday, 17 December 2033	244	35	10
2034	09APR2034	Saturday, 23 December 2034	258	37	4
2035	25MAR2035	Saturday, 22 December 2035	272	39	5

Beispiel 2

Johanna und die Ski Wochenenden



Johanna und die Ski Wochenenden

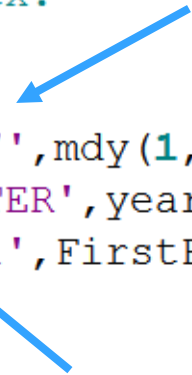
- unterrichtet bis zum letzten Samstag im Januar
- erst ab dem ersten Wochenende im Februar
- Skisaison bis Ostermontag

Wieviele Wochenenden findet Sie vor?









Johanna und die Ski Wochenenden

```
data skiing_weekends;
  format Year 8.
         FirstFullSatInFeb weekdatx.
         EasterMonday date9.;
  do Year = 2020 to 2035;
    FirstFullSatInFeb = intnx('week.7',mdy(1,31,year),1);
    EasterMonday      = holiday('EASTER',year)+1;
    NumSkiWeekends    = intck('week.1',FirstFullSatInFeb,EasterMonday);
    output;
  end;
run;
```



„week.1“ -> Woche beginnend mit Sonntag

Johanna und die Ski Wochenenden

 Year	 FirstFullSatInFeb	 EasterMonday	 NumSkiWeekends
2020	Saturday, 1 February 2020	13APR2020	11
2021	Saturday, 6 February 2021	05APR2021	9
2022	Saturday, 5 February 2022	18APR2022	11
2023	Saturday, 4 February 2023	10APR2023	10
2024	Saturday, 3 February 2024	01APR2024	9
2025	Saturday, 1 February 2025	21APR2025	 12
2026	Saturday, 7 February 2026	06APR2026	9
2027	Saturday, 6 February 2027	29MAR2027	 8
2028	Saturday, 5 February 2028	17APR2028	11
2029	Saturday, 3 February 2029	02APR2029	9
2030	Saturday, 2 February 2030	22APR2030	12
2031	Saturday, 1 February 2031	14APR2031	11
2032	Saturday, 7 February 2032	29MAR2032	8
2033	Saturday, 5 February 2033	18APR2033	11
2034	Saturday, 4 February 2034	10APR2034	10
2035	Saturday, 3 February 2035	26MAR2035	8



Beispiel 3

Foodblogger „Jack and Mary“



Food Blogger „Jack and Mary“





Planen einen Trip ins Burgenland / Steckerlfisch essen

- zwischen Ostern und Pfingsten (50 Tage)
- Wetter soll passen
 - Jack: Kein Regen (<1mm) 
 - Mary: warm (>25°C) 





Wann ist denn die beste Zeit für diese Reise?

Food Blogger „Jack and Mary“

Historische Daten vom Wasserportal Burgenland / Hydrology Burgenland

	 OrtID	 MessID	 DatumZeit	 Wert
38418	I	N	01SEP00:02:15:00	0.60
38419	I	N	01SEP00:02:30:00	0.50
38420	I	N	01SEP00:02:45:00	0.40
38421	I	N	01SEP00:03:00:00	0.40
38422	I	N	01SEP00:03:15:00	0.40
38423	I	N	01SEP00:03:30:00	0.10
38424	I	N	01SEP00:03:45:00	0.10
38425	I	N	01SEP00:04:00:00	0.10
38426	I	N	01SEP00:04:15:00	0.10
38427	I	N	01SEP00:04:30:00	0.40
38428	I	N	01SEP00:04:45:00	0.40
38429	I	N	01SEP00:05:00:00	0.80
38430	I	N	01SEP00:05:15:00	0.70
38431	I	N	01SEP00:05:30:00	2.20
38432	I	N	01SEP00:05:45:00	1.10
38433	I	N	01SEP00:06:00:00	0.60
38434	I	N	01SEP00:06:15:00	0.50
38435	I	N	01SEP00:06:30:00	0.40
38436	I	N	01SEP00:06:45:00	0.50
38437	I	N	01SEP00:07:00:00	0.40
38438	I	N	01SEP00:07:15:00	0.20
38439	I	N	01SEP00:07:30:00	0.20
38440	I	N	01SEP00:07:45:00	0.40
38441	I	N	01SEP00:08:00:00	0.00





Niederschlag (mm)

 OrtID	 MessID	 DatumZeit	 Wert
I	T	01SEP99:12:15:00	17.95
I	T	01SEP99:12:30:00	17.05
I	T	01SEP99:12:45:00	17.55
I	T	01SEP99:13:00:00	16.90
I	T	01SEP99:13:15:00	15.70
I	T	01SEP99:13:30:00	15.61
I	T	01SEP99:13:45:00	16.04
I	T	01SEP99:14:00:00	16.46
I	T	01SEP99:14:15:00	16.89
I	T	01SEP99:14:30:00	17.25
I	T	01SEP99:14:45:00	17.30
I	T	01SEP99:15:00:00	16.90
I	T	01SEP99:15:15:00	16.48
I	T	01SEP99:15:30:00	16.23
I	T	01SEP99:15:45:00	16.15
I	T	01SEP99:16:00:00	16.25
I	T	01SEP99:16:15:00	16.35
I	T	01SEP99:16:30:00	16.45
I	T	01SEP99:16:45:00	16.70
I	T	01SEP99:17:00:00	16.87
I	T	01SEP99:17:15:00	16.80

Temperatur (°C)

Food Blogger „Jack and Mary“

```
proc sql;
  *** Daily Aggregation;
  create table Weather_Illmitz
  as
  select datepart(t.datumzeit) as Date format=date9.,
         mean(t.wert) as TempMean format=8.1,
         max(t.wert) as TempMax format=8.1,
         sum(n.wert) as RainSum format=8.1
  from h.n_i_oldhist as n,
       h.t_i_oldhist as t
  where n.datumzeit = t.datumzeit|
  group by Date;
quit;
```

		Date	 TempMean	 TempMax	 RainSum
3961		31MAY2010	14.3	16.1	0.5
3962		01JUN2010	13.0	14.2	2.3
3963		02JUN2010	12.1	13.3	14.6
3964		03JUN2010	14.5	17.5	8.2
3965		04JUN2010	16.3	19.9	1.2
3966		05JUN2010	18.9	23.8	0.0
3967		06JUN2010	20.2	25.9	0.0

Food Blogger „Jack and Mary“

```
proc sql;
  *** Yearly Statistics;
  create table Weather_Yearly
  as select year(Date) as Year,
           holiday('easter',year(Date)) as EasterSunday format=date9.,
           mdy(month(calculated EasterSunday),day(calculated EasterSunday),2018)
           as Easter format=date9.,
           sum(TempMax > 25) as DaysTemp25,
           sum(RainSum > 1) as RainDays
  from Weather_Illmitz
  where holiday('easter',year(Date)) <= date <= holiday('easter',year(Date))+49
  group by Year, EasterSunday;
quit;
```

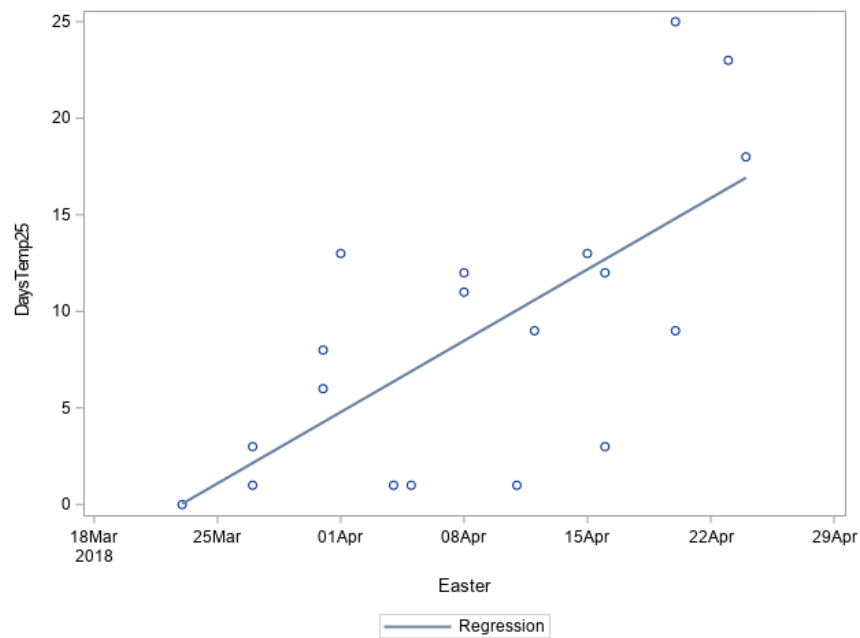
Kombination von
Holiday()
Calculated option
Sum
Where clause
Group by

Food Blogger „Jack and Mary“

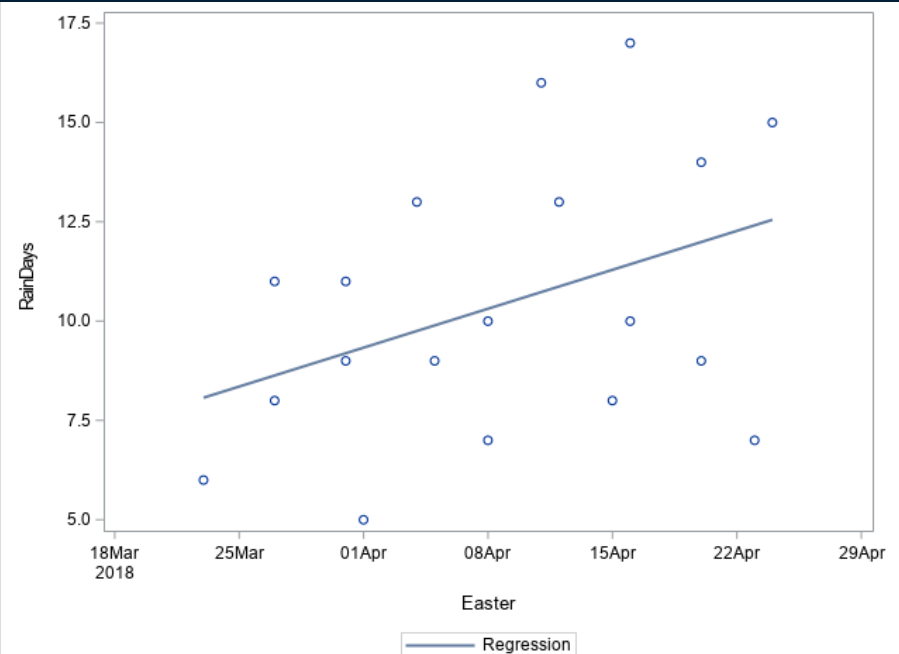
Year	EasterSunday	Easter	DaysTemp25	RainDays
2000	23APR2000	23APR2018	23	7
2001	15APR2001	15APR2018	13	8
2002	31MAR2002	31MAR2018	8	9
2003	20APR2003	20APR2018	25	9
2004	11APR2004	11APR2018	1	16
2005	27MAR2005	27MAR2018	3	8
2006	16APR2006	16APR2018	3	17
2007	08APR2007	08APR2018	12	10
2008	23MAR2008	23MAR2018	0	6
2009	12APR2009	12APR2018	9	13
2010	04APR2010	04APR2018	1	13
2011	24APR2011	24APR2018	18	15
2012	08APR2012	08APR2018	11	7
2013	31MAR2013	31MAR2018	6	11
2014	20APR2014	20APR2018	9	14
2015	05APR2015	05APR2018	1	9
2016	27MAR2016	27MAR2018	1	11
2017	16APR2017	16APR2018	12	10
2018	01APR2018	01APR2018	13	5

Food Blogger „Jack and Mary“

```
proc sgplot data=Weather_Yearly;  
  reg x=Easter y=DaysTemp25;  
run;
```



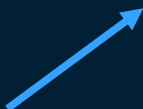
```
proc sgplot data=Weather_Yearly;  
  reg x=Easter y=RainDays;  
run;
```



Food Blogger „Jack and Mary“

```
proc reg data=Weather_Yearly;  
  model DaysTemp25 = DaysPast21MAR;  
run;
```

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	-1.01790	2.84854	-0.36	0.7252
DaysPast21MAR	1	0.52756	0.13537	3.90	0.0012



Coeff: 0.528 für DaysTemp25

Steigerung von 5.3°C / 10 Tage

```
proc reg data=Weather_Yearly;  
  model RainDays = DaysPast21MAR;  
run;
```

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > t
Intercept	1	7.79094	1.65457	4.71	0.0002
DaysPast21MAR	1	0.13998	0.07863	1.78	0.0929



Coeff: 0.14 für RainDays

1.4 mm mehr / 10 Tage

Mary



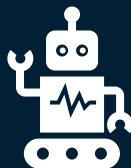
Jack

Beispiel 4:

Automatisches Berechnen von Referenzlinien

Automatisches Berechnen von Referenzlinien

Szenario: Fluglinie analysiert Passagierdaten






- Kennzeichnung aller Monate in denen der Ostersonntag nicht in den April fällt
- Entwicklung der Passagierzahlen zum Vormonat
- Darstellung in einem Diagramm mit Referenzlinien

Automatisches Berechnen von Referenzlinien

Historische Daten auf Monatsebene zu

Anzahl der Passagiere

Anzahl der Flüge

 Date	 Passengers	 Nflights
199001	3720819	18592
199002	3607839	18168
199003	4415454	18218
199004	4135558	17334
199005	4123492	17375
199006	4419061	18232
199007	4603631	17770
199008	4886615	17361
199009	3898926	19074
199010	4124753	17671
199011	3934611	16902
199012	3934201	19386
199101	3671667	19046
199102	3368862	16961
199103	3947748	17005
199104	3987814	17698
199105	4101884	16761
199106	4316618	17259

Automatisches Berechnen von Referenzlinien

```
data flights_911_flag;  
  set flights_911;  
  Pass_Change_PrevMonth = dif(passengers);  
  EasterMonth = (month(date) = month(holiday('Easter', year(date))));  
run;
```

Date	Passengers	Nflights	Pass_Change_PrevMonth	EasterMonth
199001	3720819	18592	.	0
199002	3607839	18168	-112979.6	0
199003	4415454	18218	807614.1	0
199004	4135558	17334	-279895.7	1
199005	4123492	17375	-12065.5	0
199006	4419061	18232	295568.2	0
199007	4603631	17770	184570.5	0
199008	4886615	17361	282983.8	0
199009	3898926	19074	-987688.4	0
199010	4124753	17671	225826.1	0
199011	3934611	16902	-190141.2	0

Automatisches Berechnen von Referenzlinien

```
filename refl_dif 'C:\temp\sasclub\reflines_dif.sas';

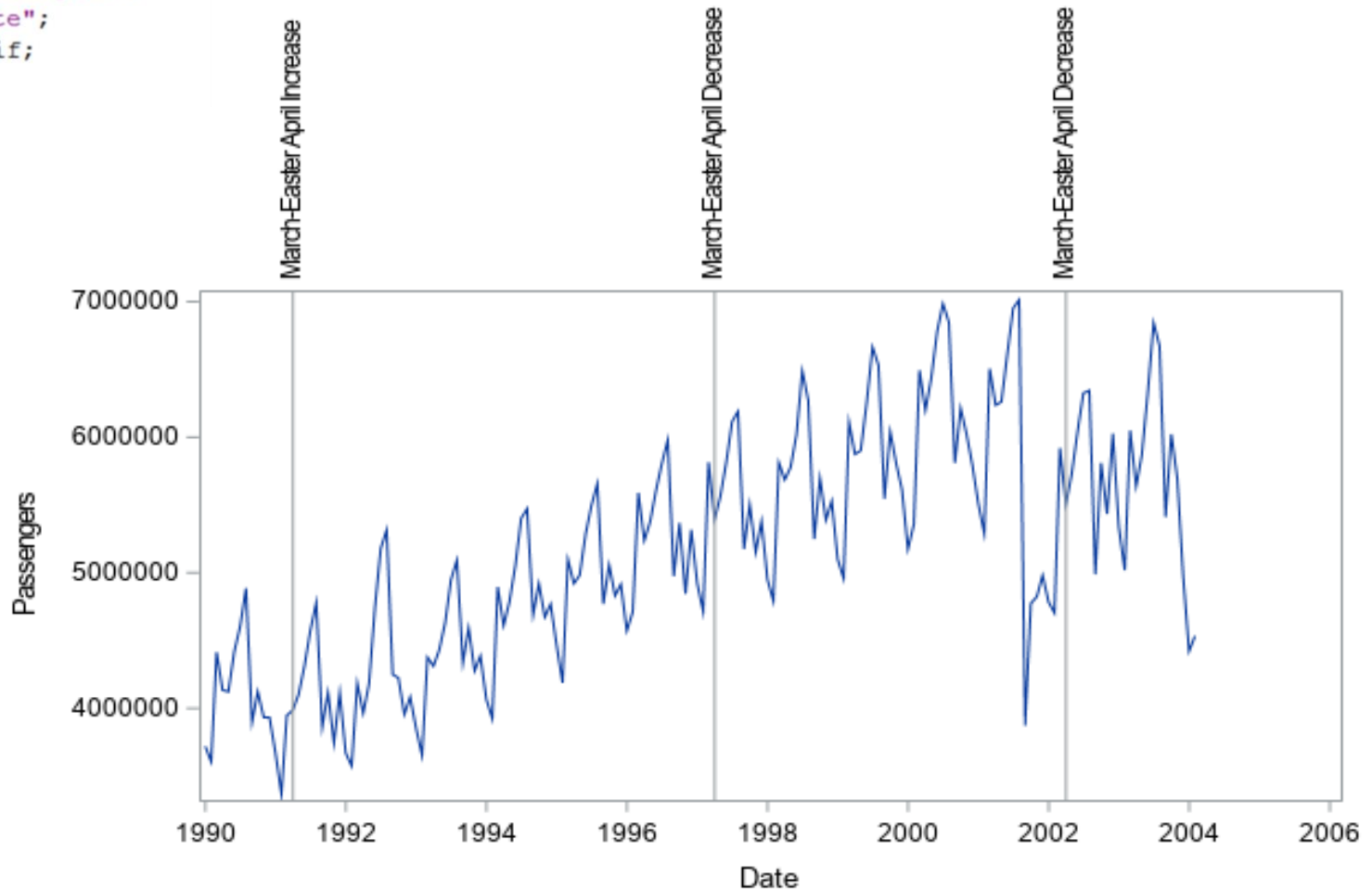
data _null_;
  set work.flights_911_flag;
  format Date 8.;
  file refl_dif;
  if month(date)=4 and EasterMonth=0 then do;
    if Pass_Change_PrevMonth<0 then
      put @04 "refline " Date " / axis = x label = 'March-Easter April Decrease';"
    else
      put @04 "refline " date " / axis = x label = 'March-Easter April Increase';"
  end;
run;
```



reflines_dif.sas	
1	refline 11413 / axis = x label = 'March-Easter April Increase';
2	refline 13605 / axis = x label = 'March-Easter April Decrease';
3	refline 15431 / axis = x label = 'March-Easter April Decrease';
4	

Automatisches Berechnen von Referenzlinien

```
proc sgplot data=flights_911;  
  series x=date y=passengers;  
  yaxis label='Passengers';  
  xaxis label="Date";  
  %include refl_dif;  
run;
```



weitere Beispiele mit analytischen Prozeduren

<https://communities.sas.com/t5/SAS-Communities-Library/Automatically-highlight-data-driven-events-with-reference-lines/ta-p/645944>

- Create reference line for breakpoints calculated by the ADAPTIVEREG procedure
- Create reference line for outliers detected with the X13 procedure
- [Detecting Structural Changes and Outliers in Longitudinal Data](#)



Vielen Dank!