ASK THE EXPERT

Tips and Tricks For Better Forecasting With SAS®

Gerhard Svolba
Advisory Presales Solutions Architect

Spiros Potamitis
Senior Product Marketing Manager

Gerhard Svolba

Advisory Presales Solutions Architect, SAS

Gerhard is involved in numerous analytic and data science projects in different business and research domains, including demand forecasting, analytical CRM, risk modeling, fraud prediction and production quality. His project experience ranges from business and technical conceptual considerations to data preparation and analytic modeling across industries. He is the author of the SAS Press books Data Preparation for Analytics Using SAS, Data Quality for Analytics Using SAS and Applying Data Science: Business Case Studies Using SAS. As a part-time lecturer, Svolba teaches data science methods at the University of Vienna, the Medical University of Vienna and for business schools.

Spiros Potamitis

Senior Product Marketing Manager, SAS

Spiros is a data scientist and a Global Product Marketing Manager of forecasting and optimization at SAS. He has extensive experience in the development and implementation of advanced analytics solutions across different industries and provides subject matter expertise in the areas of forecasting, machine learning and AI. Prior to joining SAS, Potamitis has worked on and led advanced analytics teams in various sectors such as credit risk, customer insights and CRM.

Motivation

"A 15% forecast accuracy improvement will deliver a 3% or higher pre-tax improvement to the bottom line"
IBF Study

"Focusing on the quality of data fueling AI systems will help unlock its full power."
Andrew Ng

As a rule, a data-centric approach will drive greater performance gains compared to a model-centric one

Data Preparation – a topic with many dimensions

Data Preparation for Data Science

- Data Assembly
- Data Quality for Analytics
- Feature Generation

Data Preparation for Data Science

Data Assembly

Data Quality for Analytics

Feature Generation

Main Types of Analytic Data Sturctures

One-Row-per-Subject Data Mart

	POLICYNO	CLM_FLAG	CAR_USE	CAR_TYPE	AGE	GENDER	DENSITY
1	160	No	Private	Sedan	60	M	Highly Urban
2	24836	No	Commercial	Sedan	43	M	Highly Urban
3	28046	No	Private	Van	48	M	Urban
4	28960	No	Private	SUV	35	F	Highly Urban
5	40933	No	Private	Sedan	51	M	Highly Urban
6	55277	No	Private	SUV	50	F	Urban
7	63212	Yes	Commercial	Sports Car	34	F	Highly Urban
8	69651	No	Private	SUV	54	F	Highly Urban
9	88070	Yes	Private	Sedan	40	M	Urban
10	93553	No	Commercial	SUV	44	F	Rural

BY <analysis subject>
BY CustID;
BY AccountID;
BY PatientID;

Multiple-Row-per-Subject Data Mart

CUSTOMER	TIME	PRODUCT
0	0	hering
0	1	corned_b
0	2	olives
0	3	ham
0	4	turkey
0	5	bourbon
0	6	ice_crea
1	0	baguette
1	1	soda
1	2	hering
1	3	cracker
1	4	heineken
1	5	olives

Longitudinal Data Mart

	Product_ID	YearMonth	Quantity
1	10002	201401	1173
2	10002	201402	601
3	10002	201403	584
4	10002	201404	987
5	10002	201405	461
6	10002	201406	457
7	10002	201407	497
8	10002	201408	402
9	10003	201401	4513
10	10003	201402	2395
11	10003	201403	2421
12	10003	201404	2903
13	10003	201405	1203

BY <timevar>
<cross-sectional variables>;
BY Date, Region;
BY Date, Region, ProductGroup;
BY Date, Region, SKU_ID;

Transposing Data between One-Row-Per-Subject and Multiple-Row-Per-Subject

	id	weight	time
1	1	77	1
2	1	79	2
3	1	83	3
4	2	62	1
5	2	58	2
6	2	59	3
7	3	99	1
8	3	97	2
9	3	92	3

Makewide

Makelong

	id	weight1	weight2	weight3
1	1	77	79	83
2	2	62	58	59
3	3	99	97	92

Transposing from LONG to WIDE

Using the TRANSPOSE procedure

The following code shows how you can use the TRANSPOSE procedure to

```
PROC TRANSPOSE DATA = dogs_long
               PREFIX =histamine
               OUT = dogs_wide_hist1;
   BY dogid drug depleted;
   VAR Histamine;
   ID Measurement;
RUN;
```

Proc Transpose and/or %MAKEWIDE and %MAKELONG Macro

Transpose your analysis data with the %MAKELONG and %MAKEWIDE macro

Started: 01-23-2022 | Modified: 04-01-2022 | Views: 3,551

MAKEWIDE and MAKELONG Examples.sas Create_dogs_wide_data.sas Create_dogs_long_data.sas
Macro - MAKEWIDE and MAKELONG.sas

This article introduces the macros %MAKEWIDE and %MAKELONG to transpose your data between different formats. The macros have been introduced with the SAS Press Book Data Preparation for Ana
Both macros are based on the TRANSPOSE proce

- you can transpose more than one variable in
- you can write shorter code, especially when r

Link

Transposing from WIDE to LONG

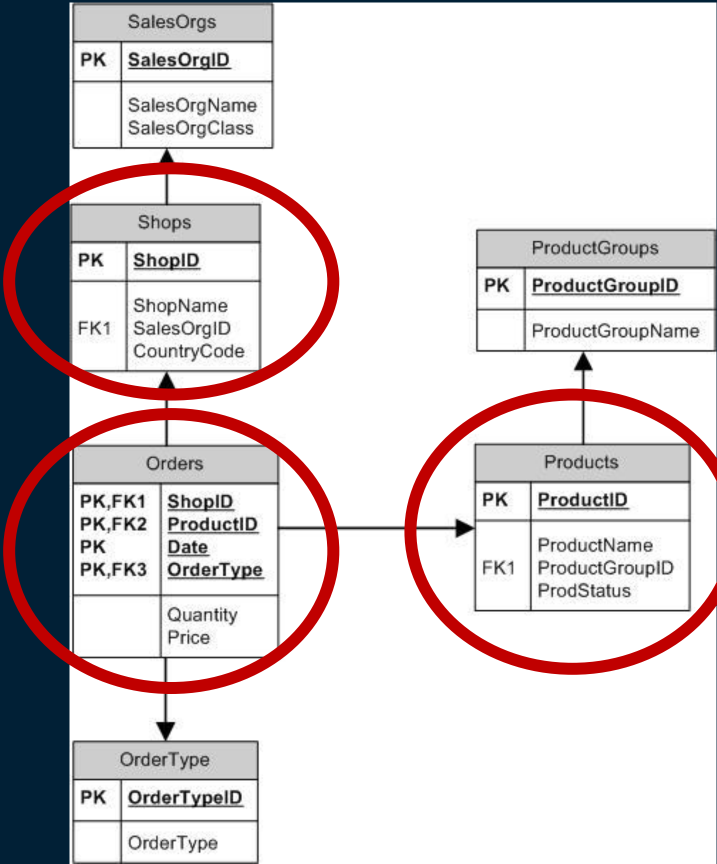
Using the TRANSPOSE procedure

You can also use the TRANSPOSE procedure to transpose the data from a WIDE to LONG structure.

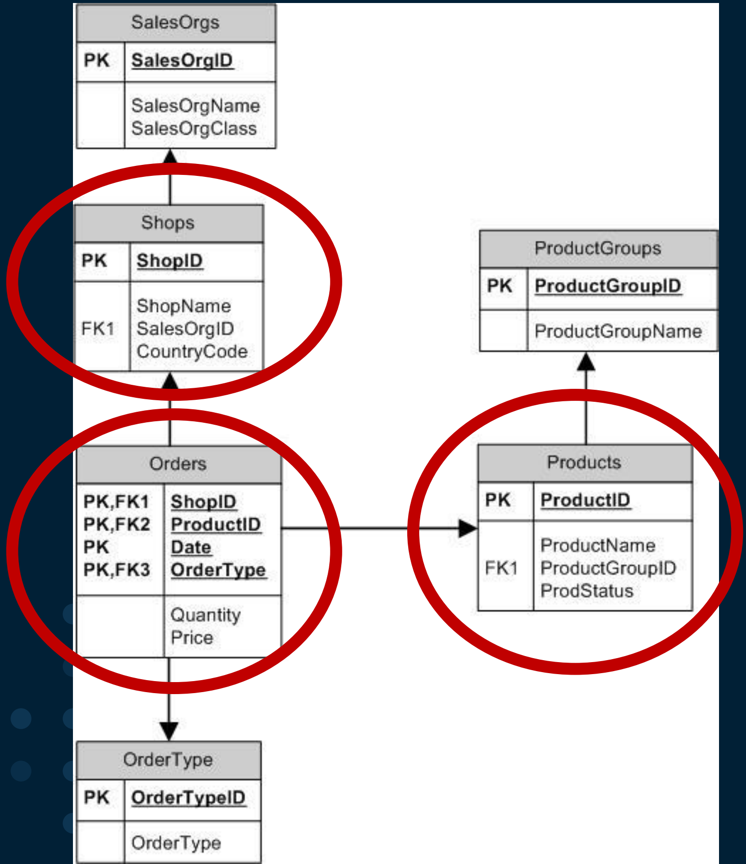
```
proc transpose data=dogs_wide Name=_measure
               out=dogs_long_1var(rename=(col1=Histamine) where=(_measure contains "Hist
 by dogid drug depleted;
run;
```

Data Model Deriving the time series data from a star-schema (relational model)

- Consider the tables: ORDERS, SHOPS, PRODUCTS
- Tables shall be joined to an ORDERMART table with monthly aggregats
- What steps are needed?
  - Merge tables based on the key columns
  - Aggregate data per BY-groupAggregate (accumulate) data per desired time interval
  - Improve and enhance data from an analytical point of view
    - Contiguity
    - Missing Values



Assemble your time series data



	ProductID	ProductGro...	MonthYear	Quantity	AvgPrice
281	11350	26	2004.07	86	940.18
282	11350	26	2004.06	32	940.18
283	11350	26	2004.07	279	940.18
284	11350	26	2004.10	86	940.18
285	11350	26	2004.07	85	940.18
286	11350	26	2004.05	157	940.18
287	11350	26	2004.07	104	940.18
288	11350	26	2004.05	152	940.18
289	11350	26	2004.04	138	940.18
290	11350	26	2004.12	60	940.18
291	11350	26	2004.09	77	940.18
292	11350	26	2004.11	61	940.18
293	11350	26	2004.06	64	940.18

Frequent ways to Merge, Aggregate and Enhance your Timeseries Data in SAS

	Merge	Aggregate	Enhance	
PROC SQL/ PROC FEDSQL	YES	YES		Merge and aggregate in one step
SAS/Datastep	YES			Fast, requires sorted data.
PROC MEANS / PROC SUMMARY/ PROC MDSUMMARY		YES		Fast, requires data in one table
PROC TIMESERIES		YES	YES	Powerful in time series data preparation, Requires SAS/ETS or SAS Econometrics licencse

4 Methods How to Join a (Lookup) Table to a Master Table

	month	product	actual
18	01JUN1994	SOFA	$431.00
19	01JUL1994	SOFA	$511.00
20	01AUG1994	SOFA	$157.00
21	01SEP1994	SOFA	$520.00
22	01OCT1994	SOFA	$114.00
23	01NOV1994	SOFA	$277.00
24	01DEC1994	SOFA	$561.00
25	01JAN1993	BED	$220.00
26	01FEB1993	BED	$444.00
27	01MAR1993	BED	$178.00
28	01APR1993	BED	$756.00
29	01MAY1993	BED	$329.00

\+

	PRODUCT	PRODTYPE
1	BED	FURNITURE
2	SOFA	FURNITURE
3	CHAIR	OFFICE
4	DESK	OFFICE
5	TABLE	OFFICE

=

	month	product	actual	PRODTYPE
284	01DEC1994	BED	$630.00	FURNITURE
285	01DEC1994	BED	$444.00	FURNITURE
286	01DEC1994	BED	$638.00	FURNITURE
287	01DEC1994	BED	$390.00	FURNITURE
288	01DEC1994	BED	$804.00	FURNITURE
289	01JAN1993	CHAIR	$468.00	OFFICE
290	01JAN1993	CHAIR	$251.00	OFFICE
291	01JAN1993	CHAIR	$35.00	OFFICE
292	01JAN1993	CHAIR	$774.00	OFFICE
293	01JAN1993	CHAIR	$401.00	OFFICE
294	01JAN1993	CHAIR	$697.00	OFFICE
295	01JAN1993	CHAIR	$292.00	OFFICE
296	01JAN1993	CHAIR	$251.00	OFFICE

Joining the lookup table explicitly

- Proc SQL
- Datastep

„Applying“ the lookup table to the source table

- SAS Format
- Hash Table

Method 1+2: Joining the Lookup Table Explicitly

```
PROC SQL;
 CREATE TABLE prdsale_sql_lj
 AS SELECT *
    FROM prdsale AS a
    LEFT JOIN lookup  AS b
    ON a.product = b.product
    ORDER BY product, month;
QUIT;
```

```
proc sort data = lookup;
by product;run;
proc sort data = prdsale;
by product;run;

data prdsale_ds;
 merge prdsale(in=in1)
      lookup;
 by product;
 if in1;
run;

proc sort data = prdsale_ds;
by product month;run;
```

Method 3: Using a SAS Format

```
DATA FMT_PG(RENAME =(Product=start
                     ProdType=label));
 SET lookup end=last;
 RETAIN fmtname 'PG' type 'c';
RUN;

PROC FORMAT LIBRARY=work CNTLIN=FMT_PG;
RUN;

DATA prdsale_fmt;
 SET prdsale;
 FORMAT Prodtype $12.;
 Prodtype = PUT(product,$PG.);
RUN;
```

Convert the LOOKUP Table into a control table (with specific variable names)

Use PROC FORMAT to create a SAS Format based on that table

Use the SAS Format to retrieve the value from the lookup table

Method 4: Using a Hash-Table

Define the HASH Table in the SAS Datastep

Call the HASH to retrieve the Values based on the Key-Column

```
DATA prdsale_hash;
length Product ProdType $10.;

if _n_ = 1 then do;
   declare hash h(dataset: "lookup");
   h.definekey('Product');
   h.definedata('ProdType');
   h.definedone();
   call missing(Product, ProdType);
end;

 SET prdsale;
 rc = h.find();
 drop rc;

RUN;
```

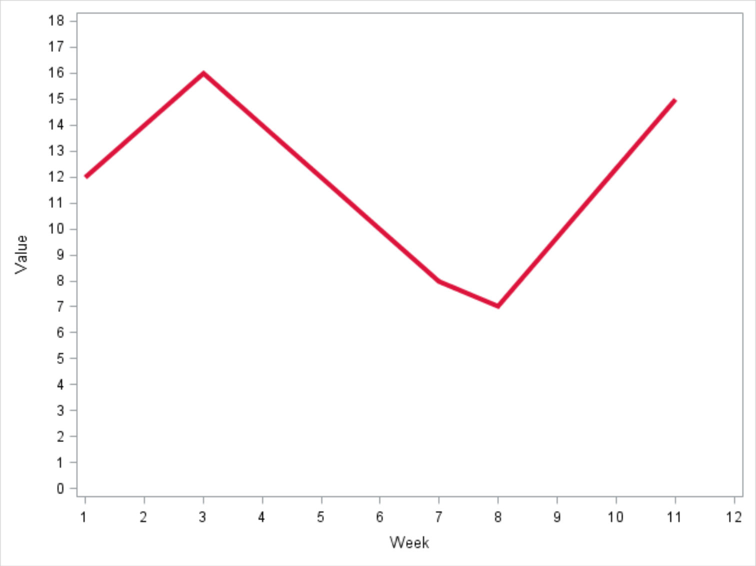
Data Preparation for Data Science

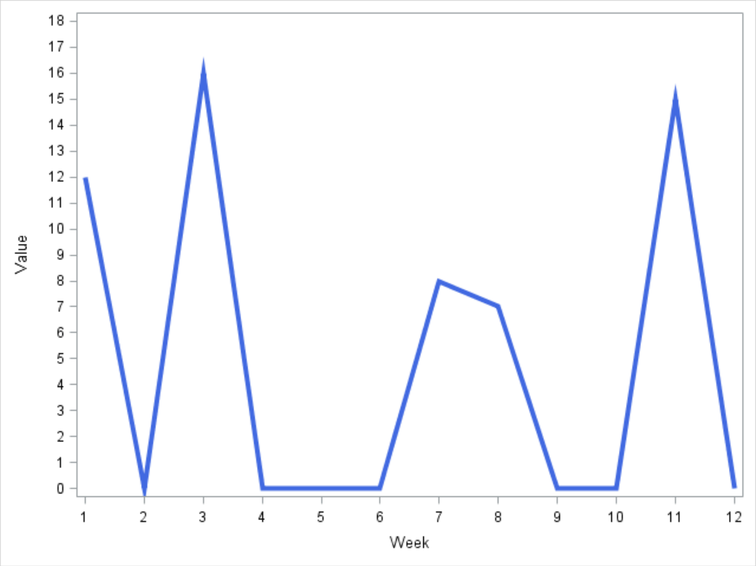
Data Assembly

Data Quality for Analytics

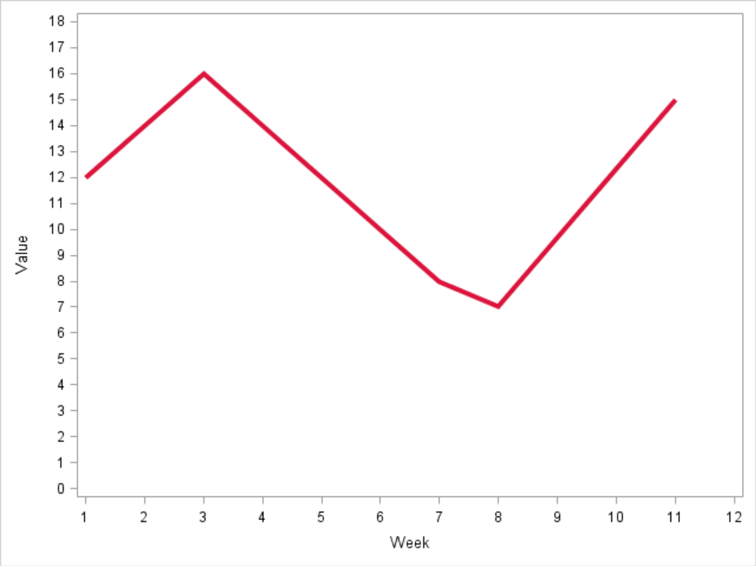
Feature Generation

Are these two graphs based on the same data?





For some measurements (inventory data) this might be the appropriate view

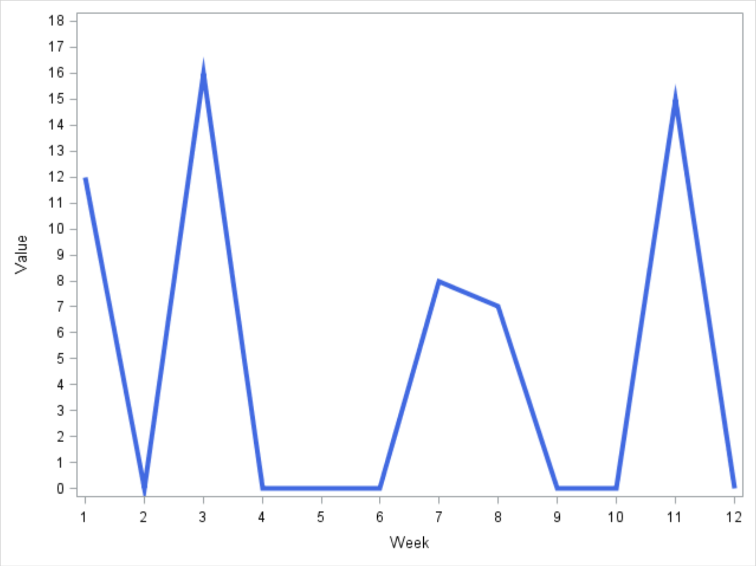


	week	value
1	1	12
2	3	16
3	7	8
4	8	7
5	11	15

For other measurements (movement data) this might be the appropriate view

Be careful with line-charts and missing values!

	week	value
1	1	12
2	2	.
3	3	16
4	4	.
5	5	.
6	6	.
7	7	8
8	8	7
9	9	.
10	10	.
11	11	15
12	12	.



Transactional Data or Timeseries Data?

Transactional
one record per event/case/…

	Session Identifier	requested_file
1	43d0a4da826149b5 2002-02-17 08:38:12	/Home.jsp
2	43d0a4da826149b5 2002-02-17 08:38:12	/Cookie_Check.jsp
3	43d0a4da826149b5 2002-02-17 08:38:12	/Home.jsp
4	43d0a4da826149b5 2002-02-17 08:38:12	/Corporate_Relations.jsp
5	43d0a4da826149b5 2002-02-17 08:38:12	/Retail_Store.jsp
6	43d0a4da826149b5 2002-02-17 08:38:12	/Store/Store_Locations.jsp
7	43d639ebce6c73d8 2002-02-17 23:43:16	/Home.jsp
8	43d639ebce6c73d8 2002-02-17 23:43:16	/Cookie_Check.jsp
9	43d639ebce6c73d8 2002-02-17 23:43:16	/Home.jsp
10	43d639ebce6c73d8 2002-02-17 23:43:16	/Department.jsp
11	43d639ebce6c73d8 2002-02-17 23:43:16	/Department.jsp
12	43bb8704bb370e09 2002-02-17 13:44:04	/Home.jsp
13	43bb8704bb370e09 2002-02-17 13:44:04	/Home.jsp
14	43bb8704bb370e09 2002-02-17 13:44:04	/Subcategory.jsp
15	43bb8704bb370e09 2002-02-17 13:44:04	/Product.jsp
16	43bb8704bb370e09 2002-02-17 13:44:04	/Department.jsp
17	43bb8704bb370e09 2002-02-17 13:44:04	/Product.jsp
18	43bb8704bb370e09 2002-02-17 13:44:04	/Department.jsp

Timeseries Data
data accumulated to time intervals

	Time	NumberOfReqestedFiles
1	1:00:00	116
2	2:00:00	93
3	3:00:00	17
4	4:00:00	158
5	6:00:00	30
6	7:00:00	66
7	8:00:00	210
8	9:00:00	130
9	10:00:00	143
10	11:00:00	298
11	12:00:00	239
12	13:00:00	145

Explicit or implicit missing values in longitudinal data

PNR	date	amount
56	2004-02-01	48
56	2004-03-01	51
56	2004-04-01	42
56	2004-05-01	36
56	2004-06-01	6
56	2004-07-01	.
56	2004-08-01	48
56	2004-09-01	36
56	2004-10-01	66
56	2004-11-01	15
56	2004-12-01	33
58	2005-06-01	39
58	2005-07-01	63
58	2005-08-01	84
58	2005-09-01	18
58	2005-12-01	69
58	2006-03-01	0
58	2006-07-01	90
58	2006-10-01	57
58	2007-01-01	48

Existing Record
Value Missing

Missing Record
No Continuity

Replacing and interpolating missing values in longitudinal data with SAS

Insert missing records | Replace with 0 | Replace with last known value | Replace with mean | Interpolate based on splines

	DATE	air_mv	air_mv_zero	air_mv_previous	air_mv_mean	air_expand
57	SEP53	.	0	264	284.54385965	246.26342876
58	OCT53	211	211	211	211	211
59	NOV53	180	180	180	180	180
60	DEC53	201	201	201	201	201
61	JAN54	204	204	204	204	204
62	FEB54	188	188	188	188	188
63	MAR54	235	235	235	235	235
64	APR54	227	227	227	227	227
65	MAY54	.	0	227	284.54385965	233.15157085
66	JUN54	264	264	264	264	264
67	JUL54	.	0	264	284.54385965	291.59030488
68	AUG54	293	293	293	293	293

Use PROC TIMESERIES and PROC EXPAND for these tasks

Aggregation and Processing of Data in One Step with the TIMESERIES Procedure

```
proc timeseries data = air_missing
 out = air_setmissing_zero;
 id date interval =month setmiss=0;
 var air_MV;
run;
```

```
proc timeseries data = air_missing
 out = air_setmissing_mean;
 id date interval =month setmiss=MEAN;
 var air_MV;
run;
```

```
proc timeseries data = air_missing
 out = air_setmissing_previous;
 id date interval =month setmiss=PREVIOUS;
 var air_MV;
run;
```

Option value	Missing values are set to
<number>	Any number. (for example, 0 to replace missing values with zero)
MISSING	Missing
MINIMUM	Minimum value of the time series
FIRST	First non-missing value
NEXT	Next non-missing value

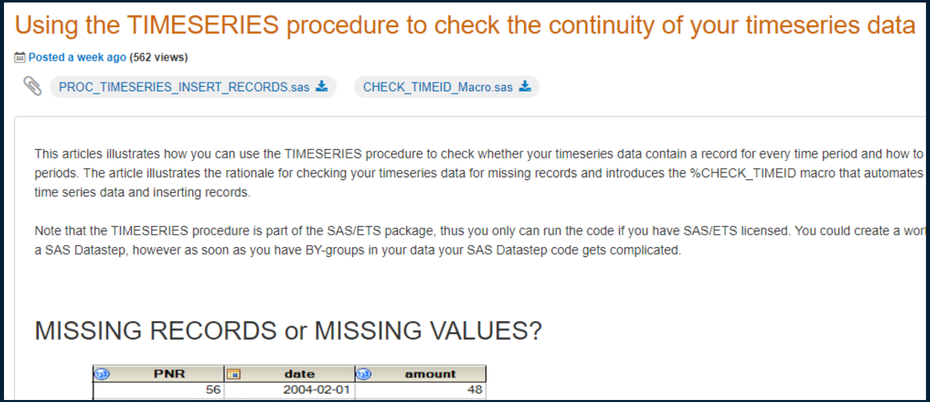
Convert Leading and Trailing Zeros to Missing Values

	DATE	sales
1	JAN49	0
2	FEB49	0
3	MAR49	0
4	APR49	0
5	MAY49	0
6	JUN49	0
7	JUL49	148
8	AUG49	148
9	SEP49	136
10	OCT49	119
11	NOV49	104
12	DEC49	118

	DATE	sales
1	JAN1949	.
2	FEB1949	.
3	MAR1949	.
4	APR1949	.
5	MAY1949	.
6	JUN1949	.
7	JUL1949	148
8	AUG1949	148
9	SEP1949	136
10	OCT1949	119
11	NOV1949	104
12	DEC1949	118

```
proc timeseries
    data=sales_original
    out=sales_corrected;
id date interval=month
        zeromiss=both;
var sales;
run;
```

Two related Articles at Communities.sas.com



Using the TIMESERIES procedure to check the continuity of your timeseries data

Posted a week ago (562 views)

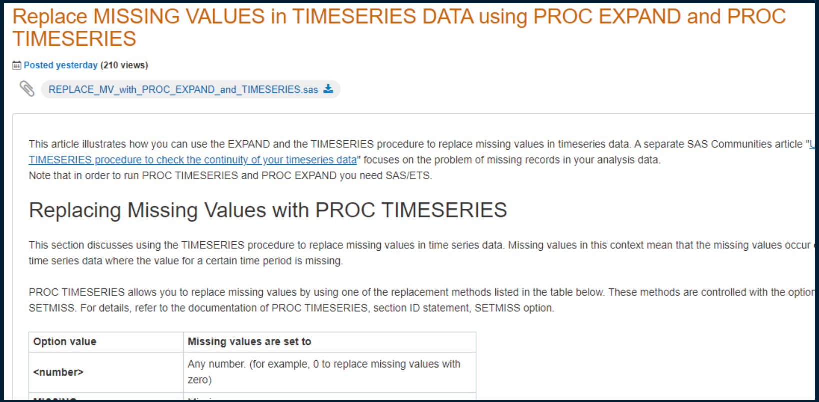
PROC_TIMESERIES_INSERT_RECORDS.sas CHECK_TIMEID_Macro.sas

This articles illustrates how you can use the TIMESERIES procedure to check whether your timeseries data contain a record for every time period and how to periods. The article illustrates the rationale for checking your timeseries data for missing records and introduces the %CHECK_TIMEID macro that automates time series data and inserting records.

Note that the TIMESERIES procedure is part of the SAS/ETS package, thus you only can run the code if you have SAS/ETS licensed. You could create a wor a SAS Datastep, however as soon as you have BY-groups in your data your SAS Datastep code gets complicated.

MISSING RECORDS or MISSING VALUES?

PNR	date	amount
56	2004-02-01	48



Replace MISSING VALUES in TIMESERIES DATA using PROC EXPAND and PROC TIMESERIES

Posted yesterday (210 views)

REPLACE_MV_with_PROC_EXPAND_and_TIMESERIES.sas

This article illustrates how you can use the EXPAND and the TIMESERIES procedure to replace missing values in timeseries data. A separate SAS Communities article "U TIMESERIES procedure to check the continuity of your timeseries data" focuses on the problem of missing records in your analysis data.
Note that in order to run PROC TIMESERIES and PROC EXPAND you need SAS/ETS.

Replacing Missing Values with PROC TIMESERIES

This section discusses using the TIMESERIES procedure to replace missing values in time series data. Missing values in this context mean that the missing values occur time series data where the value for a certain time period is missing.

PROC TIMESERIES allows you to replace missing values by using one of the replacement methods listed in the table below. These methods are controlled with the optior SETMISS. For details, refer to the documentation of PROC TIMESERIES, section ID statement, SETMISS option.

Option value	Missing values are set to
<number>	Any number. (for example, 0 to replace missing values with zero)

https://communities.sas.com/t5/SAS-Communities-Library/Using-the-TIMESERIES-procedure-to-check-the-continuity-of-your/ta-p/714678

https://communities.sas.com/t5/SAS-Communities-Library/Replace-MISSING-VALUES-in-TIMESERIES-DATA-using-PROC-EXPAND-and/ta-p/714806

SGF-Paper: Want an Early Picture of the Data Quality Status of Your Analysis Data? SAS® Visual Analytics Shows You How

Data Preparation for Data Science

Data Assembly

Data Quality for Analytics

Feature Generation

Feature Engineering 1 - Indicating a Promotional Period

	ProductID	ProductGro...	MonthYear	Quantity	AvgPrice
281	11350	26	2004.07	86	940.18
282	11350	26	2004.06	32	940.18
283	11350	26	2004.07	279	940.18
284	11350	26	2004.10	86	940.18
285	11350	26	2004.07	85	940.18
286	11350	26	2004.05	157	940.18
287	11350	26	2004.07	104	940.18
288	11350	26	2004.05	152	940.18
289	11350	26	2004.04	138	940.18
290	11350	26	2004.12	60	940.18
291	11350	26	2004.09	77	940.18
292	11350	26	2004.11	61	940.18
293	11350	26	2004.06	64	940.18

```
DATA SALES.ORDERMART;
 SET SALES.ORDERMART;
 IF '01SEP2004'd <=
                monthyear
                          <= '30NOV2004'd
                                   THEN
        Promotion =1;
 ELSE Promotion = 0;
RUN;
```

Feature Engineering 2 - Aggregating a derived variable from the data

	ProductID	ProductGro...	MonthYear	Quantity	AvgPrice
281	11350	26	2004.07	86	940.18
282	11350	26	2004.06	32	940.18
283	11350	26	2004.07	279	940.18
284	11350	26	2004.10	86	940.18
285	11350	26	2004.07	85	940.18
286	11350	26	2004.05	157	940.18
287	11350	26	2004.07	104	940.18
288	11350	26	2004.05	152	940.18
289	11350	26	2004.04	138	940.18
290	11350	26	2004.12	60	940.18
291	11350	26	2004.09	77	940.18
292	11350	26	2004.11	61	940.18
293	11350	26	2004.06	64	940.18

```
PROC sql;
 create table sales.nr_shops as
 select productid,
        mdy(1,1,year(monthyear))
                   as Year format = year4.,
        count(distinct shopid) as Nr_Shops
 from sales.ordermart
 group by productid,
           calculated Year;
quit;
```

	ProductID	Year	Nr_Shops
1	11350	2003	14
2	11350	2004	13
3	11350	2005	9
4	13101	2003	15
5	13101	2004	15
6	13101	2005	15
7	13105	2003	13
8	13105	2004	12
9	13105	2005	9

Feature Engineering 2 – Joining variable "#Shops" to the data

	ProductID	Year	Nr_Shops
1	11350	2003	14
2	11350	2004	13
3	11350	2005	9
4	13101	2003	15
5	13101	2004	15
6	13101	2005	15
7	13105	2003	13
8	13105	2004	12
9	13105	2005	9

```
PROC sql;
 create table sales.ordermart_enh  as
 select o.*,   n.Nr_Shops
 from sales.ordermart as o,
 left join sales.nr_shops as n
   on o.productid = n.productid
       and year(o.monthyear) = year(n.year);
quit;
```

	ProductID	ProductGro...	MonthYear	Quantity	AvgPrice	Promotion	Nr_Shops
281	11350	26	2004.07	86	940.18	0	13
282	11350	26	2004.06	32	940.18	0	13
283	11350	26	2004.07	279	940.18	0	13
284	11350	26	2004.10	86	940.18	1	13
285	11350	26	2004.07	85	940.18	0	13
286	11350	26	2004.05	157	940.18	0	13
287	11350	26	2004.07	104	940.18	0	13
288	11350	26	2004.05	152	940.18	0	13
289	11350	26	2004.04	138	940.18	0	13
290	11350	26	2004.12	60	940.18	0	13
291	11350	26	2004.09	77	940.18	1	13
292	11350	26	2004.11	61	940.18	1	13
293	11350	26	2004.06	64	940.18	0	13

Feature Engineering 3 – Complex aggregations along the time axis

How many skiing weekends fall between the end of the winter term at the University and Easter?

```
data skiing_weekends;
 format  Year                8.
         FirstFullSatInFeb   weekdatx.
         EasterMonday        date9.;
 do Year = 2015 to 2025;
  FirstFullSatInFeb = intnx('week.7',mdy(1,31,year),1);
  EasterMonday      = holiday('EASTER',year)+1;
  NumSkiWeekends    = intck('week.1',FirstFullSatInFeb,EasterMonday);
  output;
 end;
run;
```

Year	FirstFullSatInFeb	EasterMonday	NumSkiWeekends
2015	Saturday, 7 February 2015	06APR2015	9
2016	Saturday, 6 February 2016	28MAR2016	8
2017	Saturday, 4 February 2017	17APR2017	11
2018	Saturday, 3 February 2018	02APR2018	9
2019	Saturday, 2 February 2019	22APR2019	12
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

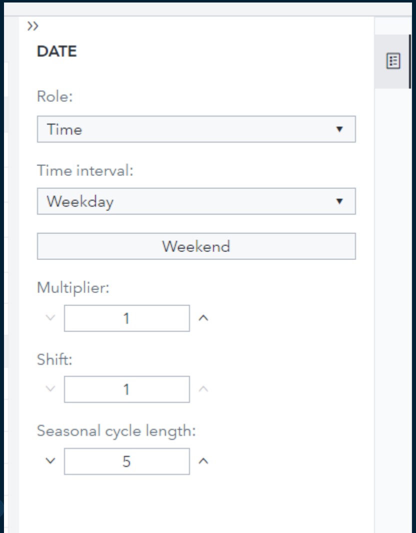
Link

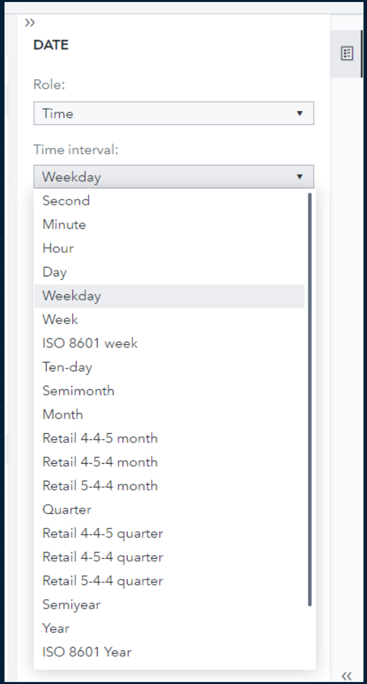
Data Preparation in SAS Visual Forecasting

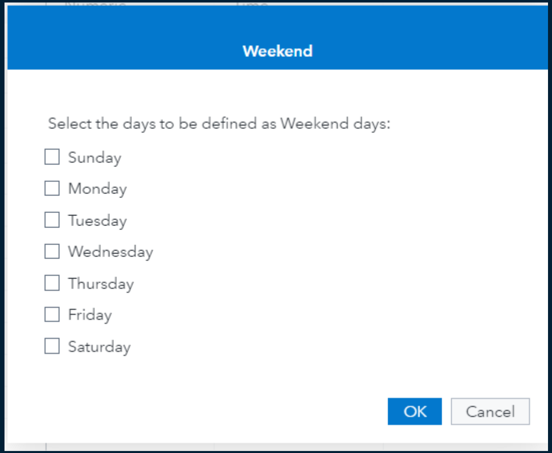
Time Interval

Your data can already be in the time series form you require

but you can also handle it in SAS Visual Forecasting UI







Multiplier: affects the length of the interval
Shift: affects the starting point of the interval

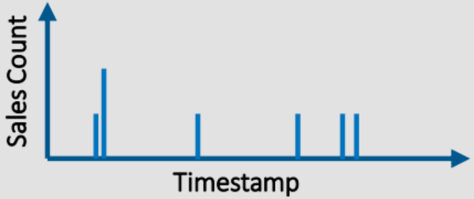
[Click here to find out more about time intervals]

Time Series Accumulation

Accumulation combines data within the same time interval into a summary value for that time interval.

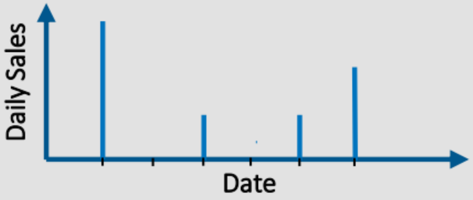
Transactional Data

Timestamp	Sales Count
Feb. 10, 2:00pm	1
Feb. 10, 3:00pm	2
Feb. 12, 9:00am	1
Feb. 14, 10:00am	1
Feb. 15, 1:00pm	1
Feb. 15, 4:00pm	1



Time Series Data

Date	Daily Sales
Feb. 10	3
Feb. 11	0
Feb. 12	1
Feb. 13	0
Feb. 14	1
Feb. 15	2



Time Series Hierarchical Aggregation

Select the aggregation method that you want to use for all the time series in each level of the hierarchy.

Sales per region, store and item

Region	Store	Item	Date	Price	Sales
USA	1	1	Mar-23	30	100
USA	2	1	Mar-23	35	200
Europe	3	1	Mar-23	20	500
Europe	4	1	Mar-23	40	250
Asia	5	1	Mar-23	30	100
Asia	6	1	Mar-23	50	80

Aggregate to Region

- Sum of Sales
- Average Price

Region	Date	Sum Sales	Average Price
USA	Mar-23	300	32.5
Europe	Mar-23	750	30
Asia	Mar-23	180	40

Missing Value Interpretation

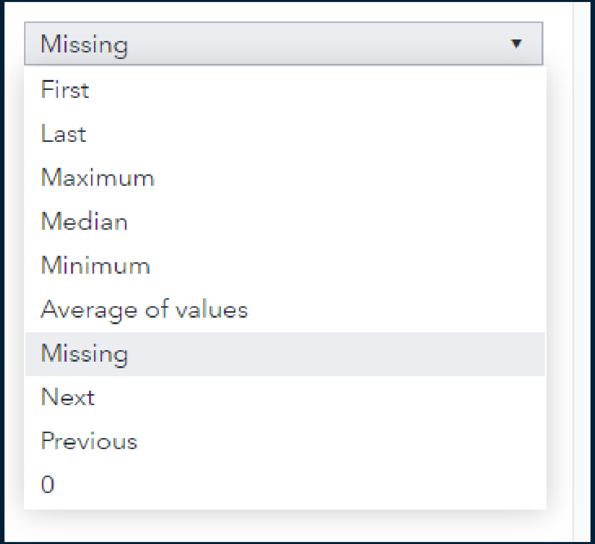
Select how you want to treat missing values

Sales per region, store and item

Region	Store	Item	Date	Price	Sales
USA	1	1	Mar-23	30	.
USA	2	1	Mar-23	35	200
Europe	3	1	Mar-23	.	500
Europe	4	1	Mar-23	40	.
Asia	5	1	Mar-23	.	.
Asia	6	1	Mar-23	50	80

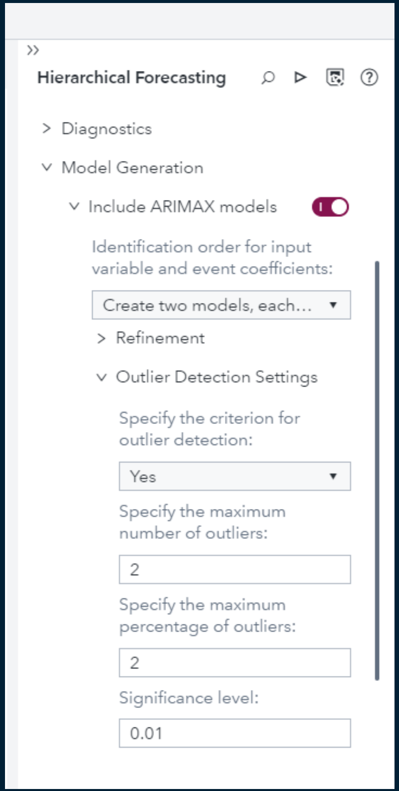
- The correct imputation is based on the business problem.
- When value is set to missing, the data will be automatically treated in the modeling phase based on the algorithm used.

Imputation methods



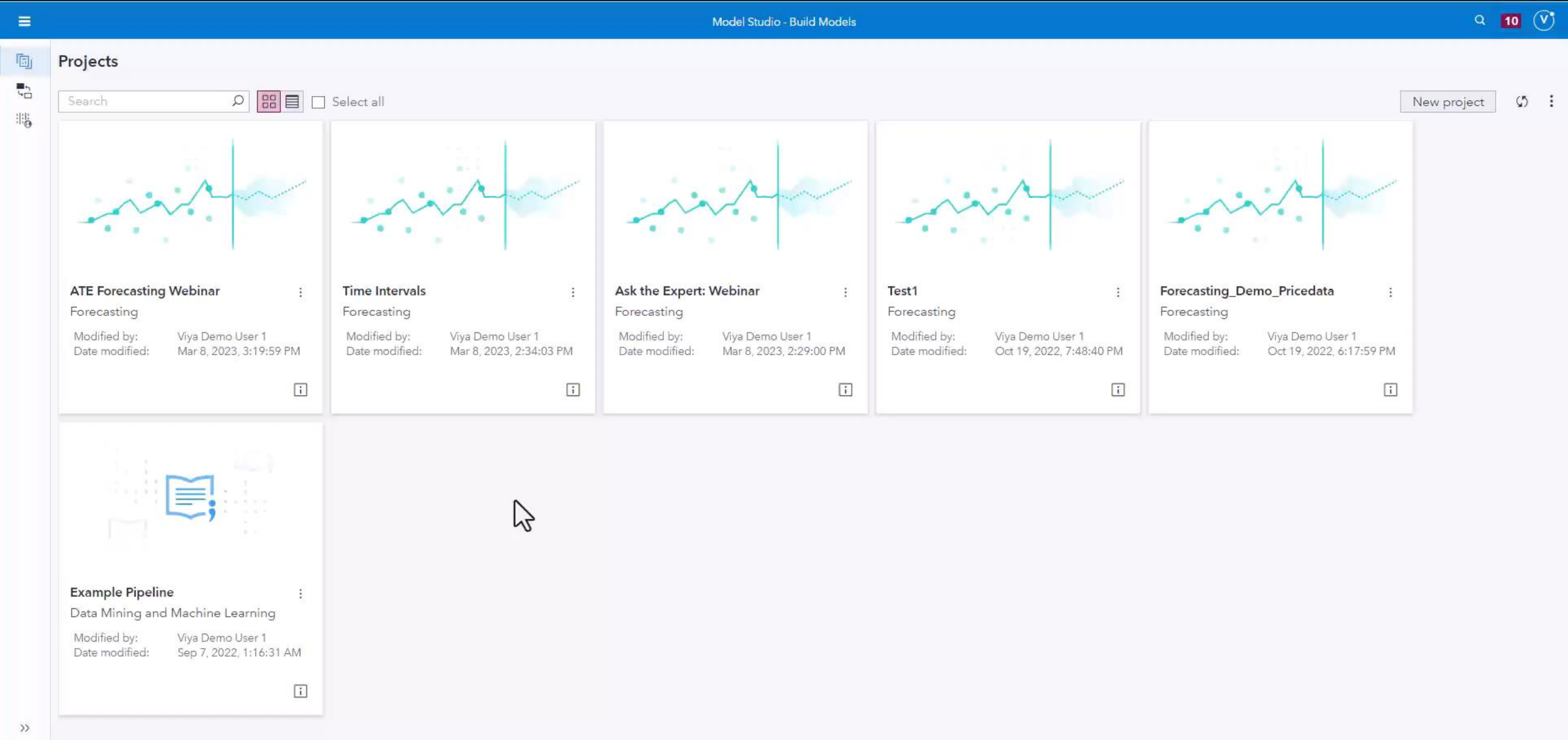
Outlier Detection to Use as Inputs in Forecasting

Automatic in Hierarchical Forecasting Node



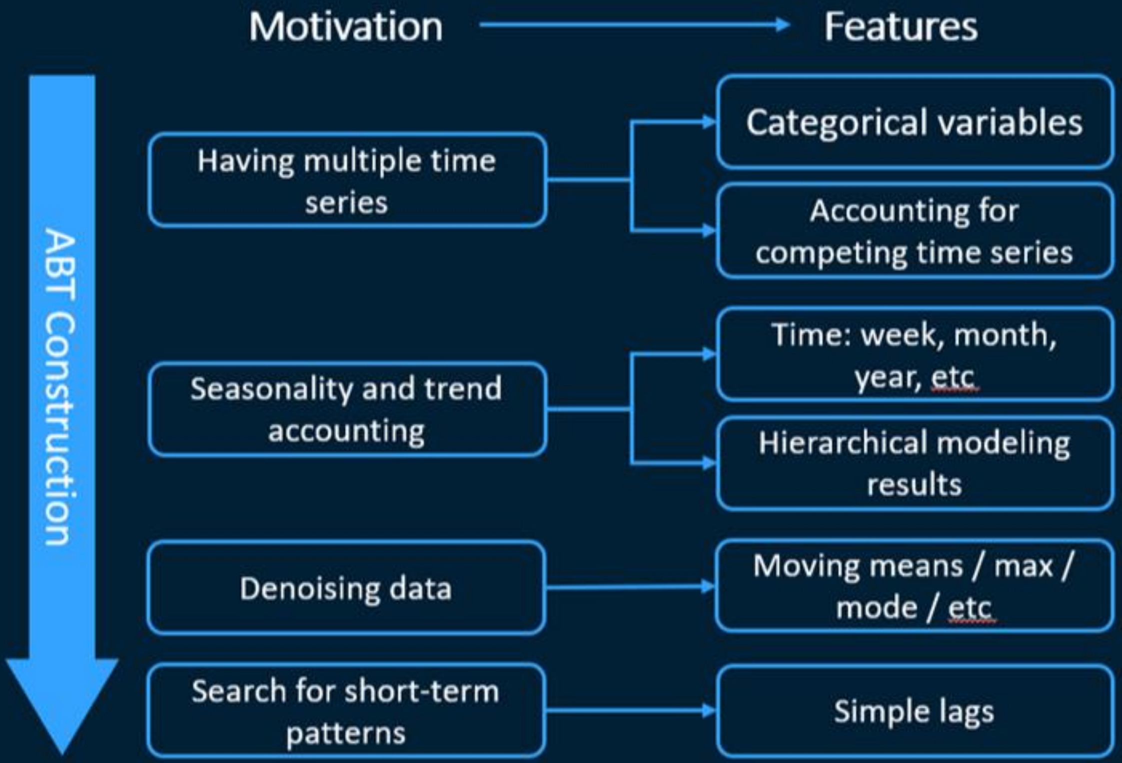
Add one line of code to include in other nodes such as Auto-Forecasting

```
diagSpec.setARIMAXOutlier('detect', 'YES');
```


Model Studio - Build Models
10
Projects
Search
Select all
New project
ATE Forecasting Webinar
Forecasting
Modified by: Viya Demo User 1
Date modified: Mar 8, 2023, 3:19:59 PM
Time Intervals
Forecasting
Modified by: Viya Demo User 1
Date modified: Mar 8, 2023, 2:34:03 PM
Ask the Expert: Webinar
Forecasting
Modified by: Viya Demo User 1
Date modified: Mar 8, 2023, 2:29:00 PM
Test1
Forecasting
Modified by: Viya Demo User 1
Date modified: Oct 19, 2022, 7:48:40 PM
Forecasting_Demo_Pricedata
Forecasting
Modified by: Viya Demo User 1
Date modified: Oct 19, 2022, 6:17:59 PM
Example Pipeline
Data Mining and Machine Learning
Modified by: Viya Demo User 1
Date modified: Sep 7, 2022, 1:16:31 AM

Getting the data ML ready

Data Preparation for Machine Learning



Data Example for Machine Learning

Dependent Variable lags

Independent Variable lags

Independent Variable lags

Moving averages

productName	date	sale	sale_lag3	sale_lag2	sale_lag1	price	price_lag3	price_lag2	price_lag1	discount	discount_lag3	discount_lag2	discount_lag1	price_movave_3m	sale_movave_3m
Product1	JAN98	355	.	.	.	52.3	.	.	.	0	.	.	.	52.3	355
Product1	FEB98	398	.	.	355	52.3	.	.	52.3	0	.	.	0	52.3	376.5
Product1	MAR98	387	.	355	398	52.3	.	52.3	52.3	0	.	0	0	52.3	380
Product1	APR98	380	355	398	387	52.3	52.3	52.3	52.3	0	0	0	0	52.3	388.33333333
Product1	MAY98	555	398	387	380	44.455	52.3	52.3	52.3	0.15	0	0	0	49.685	440.66666667
Product1	JUN98	385	387	380	555	52.3	52.3	52.3	44.455	0	0	0	0.15	49.685	440
Product1	JUL98	390	380	555	385	52.3	52.3	44.455	52.3	0	0	0.15	0	49.685	443.33333333
Product1	AUG98	377	555	385	390	52.3	44.455	52.3	52.3	0	0.15	0	0	52.3	384

Dependent Variable

Independent Variable

Independent Variable

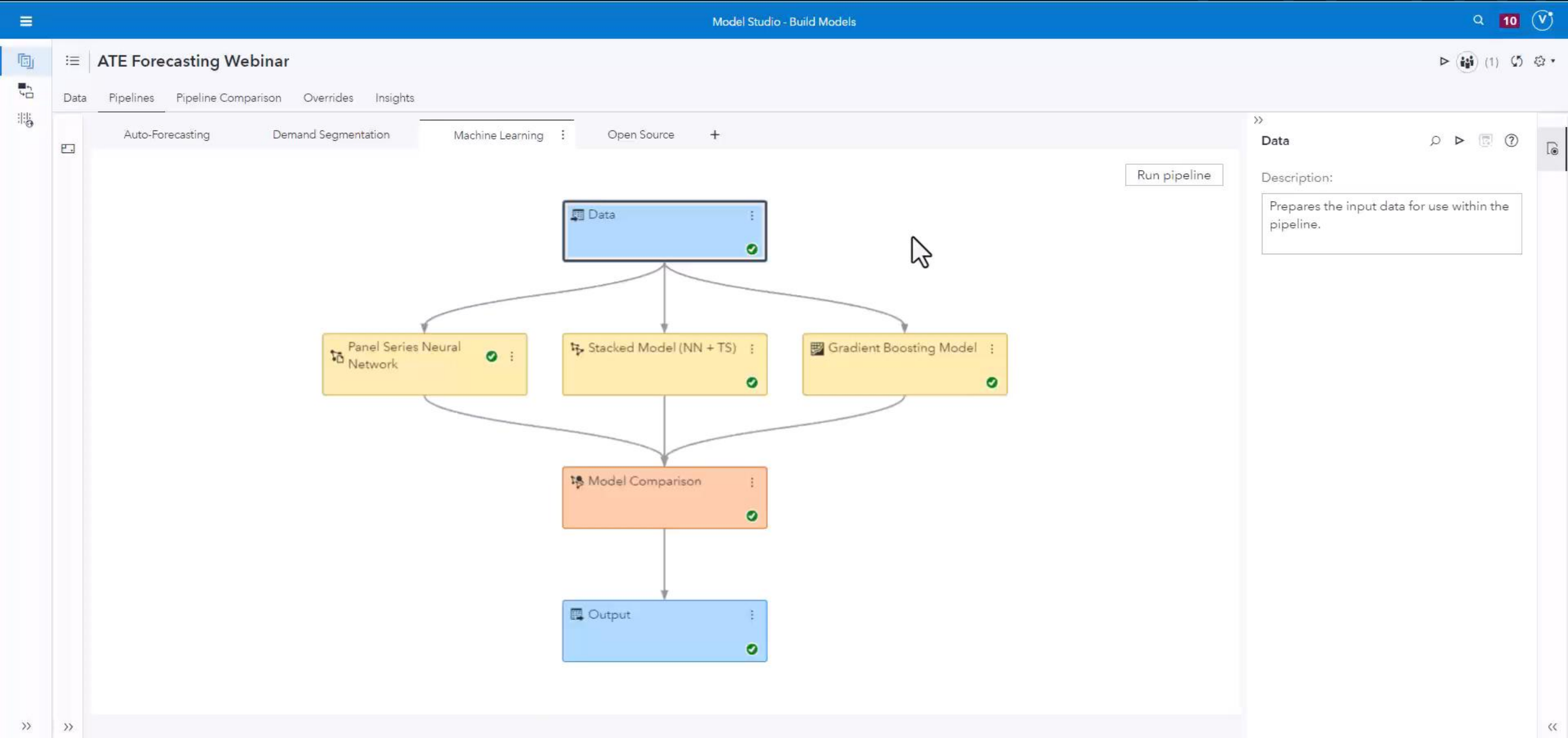
Tips & Tricks

1. This is automatically handled in VF hybrid nodes that use Neural Networks

2. This is automatically handled in VF Gradient Boosting Node that can be downloaded from this SAS repo: https://github.com/sassoftware/sas-viya-forecasting-pipelines

3. If you want to create your own ML nodes use the Gradient Boosting Node as a basis and amend the algorithm with the algorithm of choice

4. If you want to easily develop lags, moving averages etc. for all your variables and try your own experiments then Proc Expand is your ally!


Model Studio - Build Models
10
ATE Forecasting Webinar
(1)
Data
Pipelines
Pipeline Comparison
Overrides
Insights
Auto-Forecasting
Demand Segmentation
Machine Learning
Open Source
Run pipeline
Data
Panel Series Neural Network
Stacked Model (NN + TS)
Gradient Boosting Model
Model Comparison
Output
Data
Description:
Prepares the input data for use within the pipeline.

Proc Expand: Creating 3 month lags and moving averages

```
proc expand data=pricedata_transform out=out method=none;
   id date;
   by region productline productname;

   convert sale = sale_lag3   / transformout=(lag 3);
   convert sale = sale_lag2   / transformout=(lag 2);
   convert sale = sale_lag1   / transformout=(lag 1);
   convert sale;

   convert price = price_lag3   / transformout=(lag 3);
   convert price = price_lag2   / transformout=(lag 2);
   convert price = price_lag1   / transformout=(lag 1);
   convert price;

   convert discount = discount_lag3   / transformout=(lag 3);
   convert discount = discount_lag2   / transformout=(lag 2);
   convert discount = discount_lag1   / transformout=(lag 1);
   convert discount;

   convert price = price_movave_3m / transformout=(movave 3);
   convert sale = sale_movave_3m / transformout=(movave 3);

run;
```

RNNs in SAS Visual Forecasting

Automatic Data Prep for Forecasting with RNNs

Table 9: Training, Validation, and Forecast Regions

Training								Validation		Forecast		
y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	$\hat{y}_{11}$	$\hat{y}_{12}$	$\hat{y}_{13}$

Parameter settings

- number of holdout samples is 2
- input window size is 3
- forecast lead is 3

Available in TNF and TSM packages using SAS Visual Forecasting

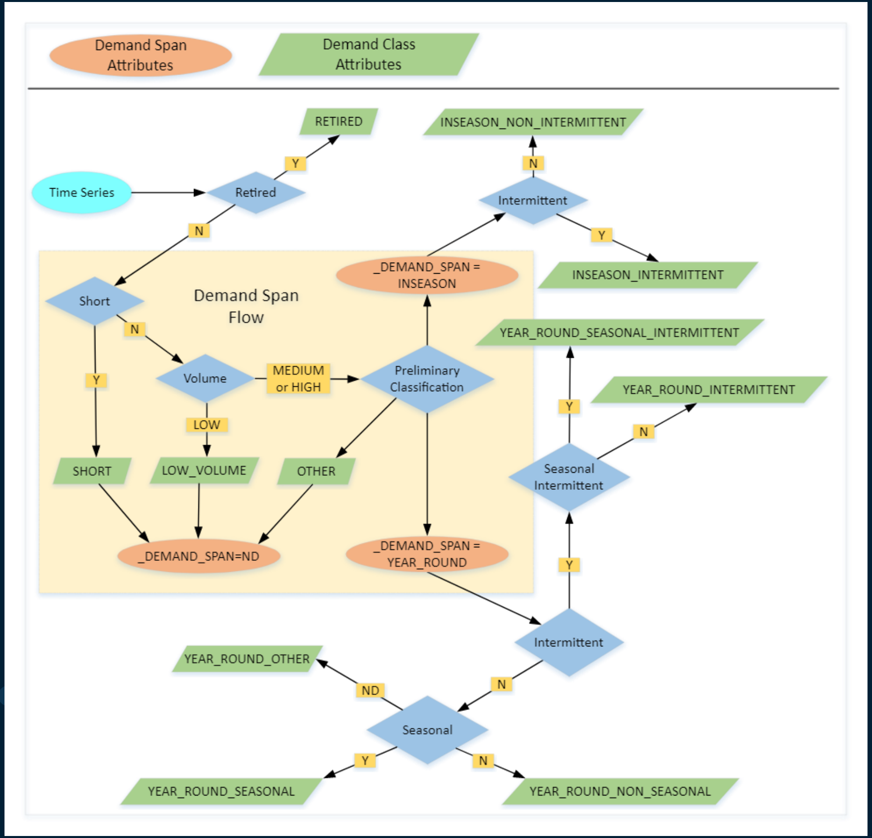
[Read the documentation to find out more]

Table 10: Training Data Structure for RNN Forecasting

Input			Target	Forecast
Training				
y_1	y_2	y_3	y_4	$\hat{y}_4$
y_2	y_3	y_4	y_5	$\hat{y}_5$
y_3	y_4	y_5	y_6	$\hat{y}_6$
y_4	y_5	y_6	y_7	$\hat{y}_7$
y_5	y_6	y_7	y_8	$\hat{y}_8$
Validation				
y_6	y_7	y_8	y_9	$\hat{y}_9$
y_7	y_8	y_9	y_{10}	$\hat{y}_{10}$
Forecast in future time period				
y_8	y_9	y_{10}	$y_{11} = \cdot$	$\hat{y}_{11}$
y_9	y_{10}	$\hat{y}_{11}$	$y_{12} = \cdot$	$\hat{y}_{12}$
y_{10}	$\hat{y}_{11}$	$\hat{y}_{12}$	$y_{13} = \cdot$	$\hat{y}_{13}$

Demand Segmentation

Automatic Time Series Segmentation



- Groups data in 11 segments that can be modelled independently
- Based on demand classification attributes
- Can enhance accuracy and optimize computational cost
- ML models perform better in interrelated series
- Experiment with advanced modeling techniques in the most demanding segments
- Read this blog for more info

Demand Segments Details

1. SHORT: Time series with a short record of historical data. This could be a new series with only a few observations.

2. LOW_VOLUME: Time series with low volumes. The Naive Forecasting pipeline is selected for this segment.

3. INSEASON_INTERMITTENT: Short time span series with intermittent patterns.

4. INSEASON_NON_INTERMITTENT: Short time span series without intermittent patterns.

5. YEAR_ROUND_INTERMITTENT: Long time span series with intermittent patterns.

6. YEAR_ROUND_SEASONAL: Long time span series with seasonal patterns.

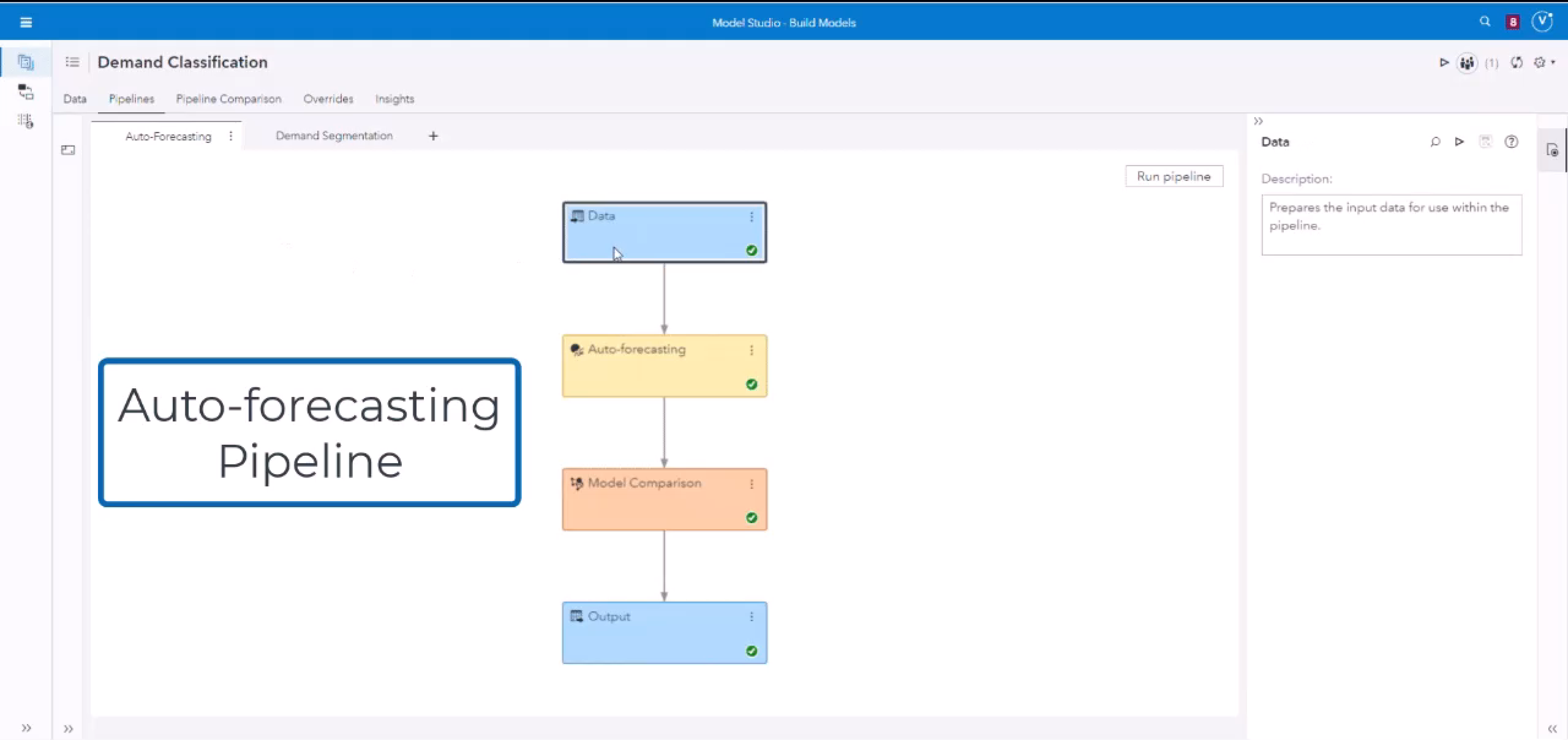
7. YEAR_ROUND_NON_SEASONAL: Long time span series without seasonal patterns.

8. YEAR_ROUND_SEASONAL_INTERMITTENT: Long time span series with seasonal and intermittent patterns.

9. YEAR_ROUND_OTHER: Long time span series with no patterns that can be classified.

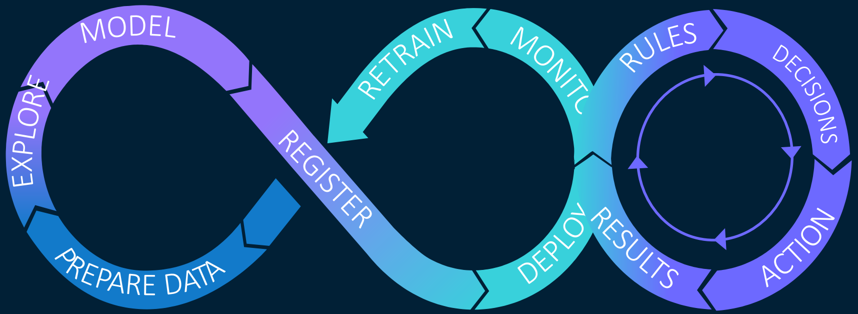
10. OTHER: Time series that do not span long time periods and cannot be classified.

11. RETIRED: Time series that are retired or are no longer active. The Retired Series model is selected for this segment.


Model Studio - Build Models
Demand Classification
Data
Pipelines
Pipeline Comparison
Overrides
Insights
Auto-Forecasting
Demand Segmentation
Run pipeline
Data
Auto-forecasting
Model Comparison
Output
Auto-forecasting Pipeline
Data
Description:
Prepares the input data for use within the pipeline.

Conclusion

- Data Preparation is all over the analytic lifecycle!



- Data Preparation is much more than just coding!

All you need to prepare your data for data science is available in the integrated SAS Viya platform

- Data Preparation / Data Quality / Feature Engineering / Variety of Analytical Methods / Visualizing Relationships / Comparing Models / What-If Scenarios / Access for different Persona Roles / Model Ops / …

Thank you

sas.com

Selected SAS Training Courses

- https://support.sas.com/edu/schedules.html?crs=STSM&ctry=US - Time Series Modeling Essentials
- This course discusses the fundamentals of modeling time series data. The course focuses on the applied use of the three main model types used to analyze univariate time series: exponential smoothin...

- https://support.sas.com/edu/schedules.html?crs=FVVF&ctry=US - Forecasting Using Model Studio in SAS® Viya®
- This course provides a hands-on tour of the forecasting functionality in Model Studio, a component of SAS Viya. The course begins by showing how to load the data into memory and visualize the time ...

- https://support.sas.com/edu/schedules.html?crs=VFSP&ctry=US - Large-Scale Forecasting Using SAS® Viya®: A Programming Approach
- This course teaches students to develop and maintain a large-scale forecasting project using SAS Visual Forecasting tools. For the course project, students build and then refine a large-scale forec...

- https://support.sas.com/edu/schedules.html?crs=MTSS&ctry=US Models for Time Series and Sequential Data
- This course teaches students to build, refine, extrapolate, and, in some cases, interpret models designed for a single, sequential series. There are three modeling approaches presented. The traditi...

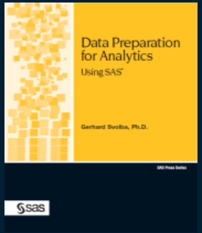
- https://support.sas.com/edu/schedules.html?crs=TSFM&ctry=US - Time Series Feature Mining and Creation
- In this course, you learn about data exploration, feature creation, and feature selection for time sequences. The topics discussed include binning, smoothing, transformations, and data set operatio...

Useful Resources 1

- Webinar: Data Preparation for Data Science" im SAS DACH Youtube Channel

- SAS Communities: Data Science and Data Preparation Article Overview by Gerhard
- Transpose your analysis data with the %MAKELONG and %MAKEWIDE macro
- 3 ways to consider movable holidays in SAS
- Replace MISSING VALUES in TIMESERIES DATA using PROC EXPAND and PROC TIMESERIES
- Using the TIMESERIES procedure to check the continuity of your timeseries data
- Have a look at your TIMESERIES data from a bird's-eye view - Profile their missing value structure

- Book: Data Preparation for Analytics Using SAS
- Book: Data Quality for Analytics Using SAS



Useful Resources 2

SAS Communities Library Articles

- How to incorporate RNNs in your SAS VF pipelines
- How to create a custom TensorFlow node in SAS VF with GUI parameters
- Modernizing Scenario Analysis with SAS Viya and SAS Visual Analytics
- Free SAS Sample Data Sets for Forecasting
- Step by step guide for using open-source models in SAS VF

SAS Papers

- Neural Network–Based Forecasting Strategies in SAS® Viya®
- Writing a Gradient Boosting Model Node for SAS® Visual Forecasting
- Scalable Cloud-Based Time Series Analysis and Forecasting Using Open-Source Software

SAS Forecasting E-Book

- Forecasting with SAS: Special Collection