Using High Performance Forecasting to measure regressors of a time-series

aka. Measuring ROI

Tim Manns
https://www.linkedin.com/in/tim-manns

Business Problems:

"why are our sales down 10% this month?"

"we just invested in staff training, what has been the impact upon sales or net promoter score (NPS)?"

"Is online advertising better than radio?"

Why use forecasting?

- business performance often has seasonality, especially in retail industry.

- activities can have lag/delay until influence is observed.
    - although, recent activities are usually powerful

- next business questions are often ‘what-if’

- IOT places greater focus upon time series data.

Which forecasting approach to use?

- structural time series models are similar to multiple regression whereby the coefficients vary over time.

- the use of 'multiple regressions' (ie. components/layers) allows us to individually measure trend, seasonal and cycle.

- By investing energy in understanding each component/layer it is more likely we can achieve an accurate forecast (we can better understand errors in each component).

- sometimes also called 'dynamic linear models'

Unobserved Components Model (UCM)

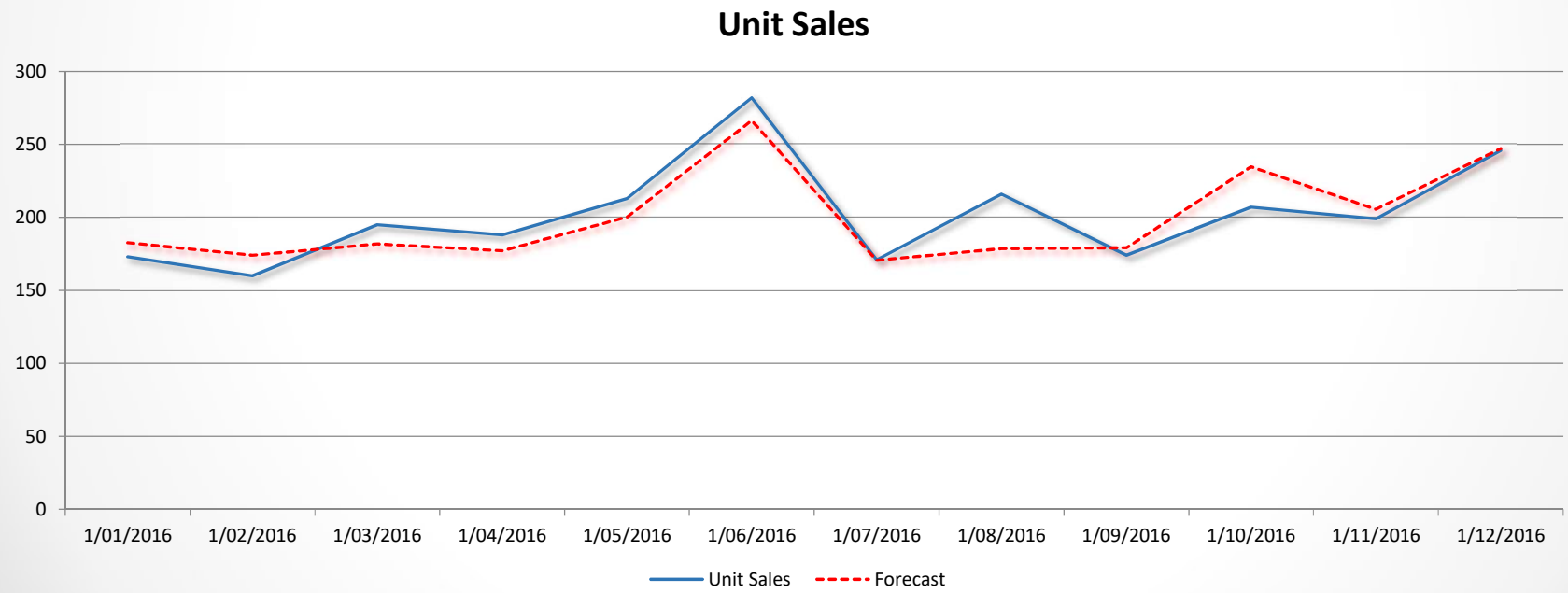
- SAS's Unobserved Components Model (UCM) is an implementation of this structural time series forecasting approach.

- it can be used to additively decompose a time series (say, historical unit sales) into components for trend, season, cycle, and irregular movements.

- we can also include additional input variables (regressors) that can also be represented in the time series forecast components. For example, marketing spend.

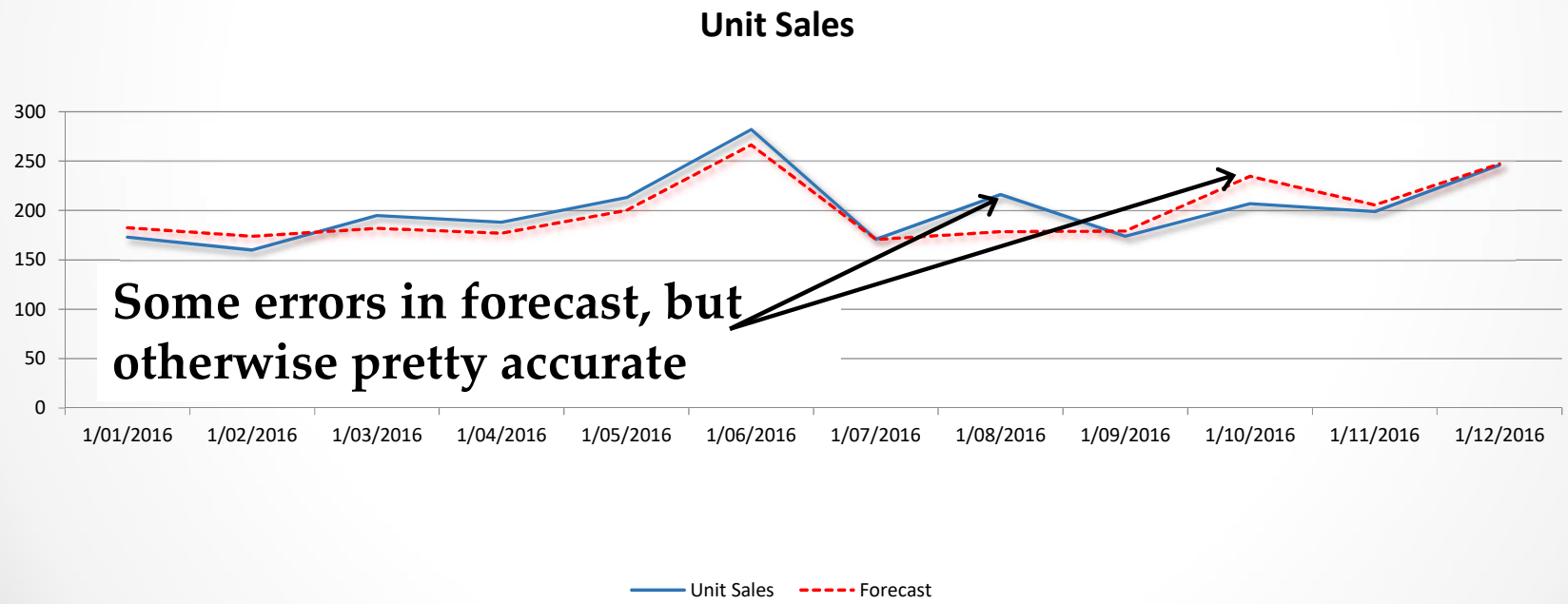
First Steps: Can we forecast it anyway?

- we can claim we understand influences upon unit sales if we can accurately forecast it.



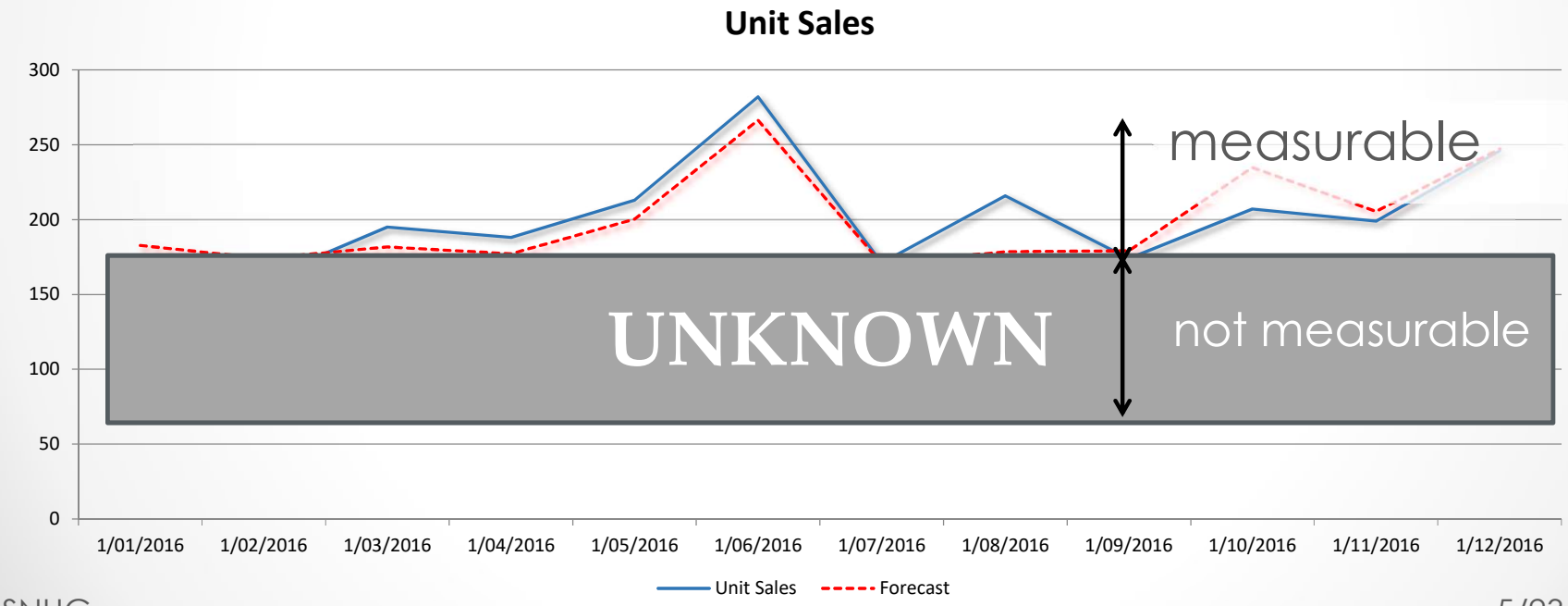
First Steps: Can we forecast it anyway?

- we can claim we understand influences upon unit sales if we can accurately forecast it.



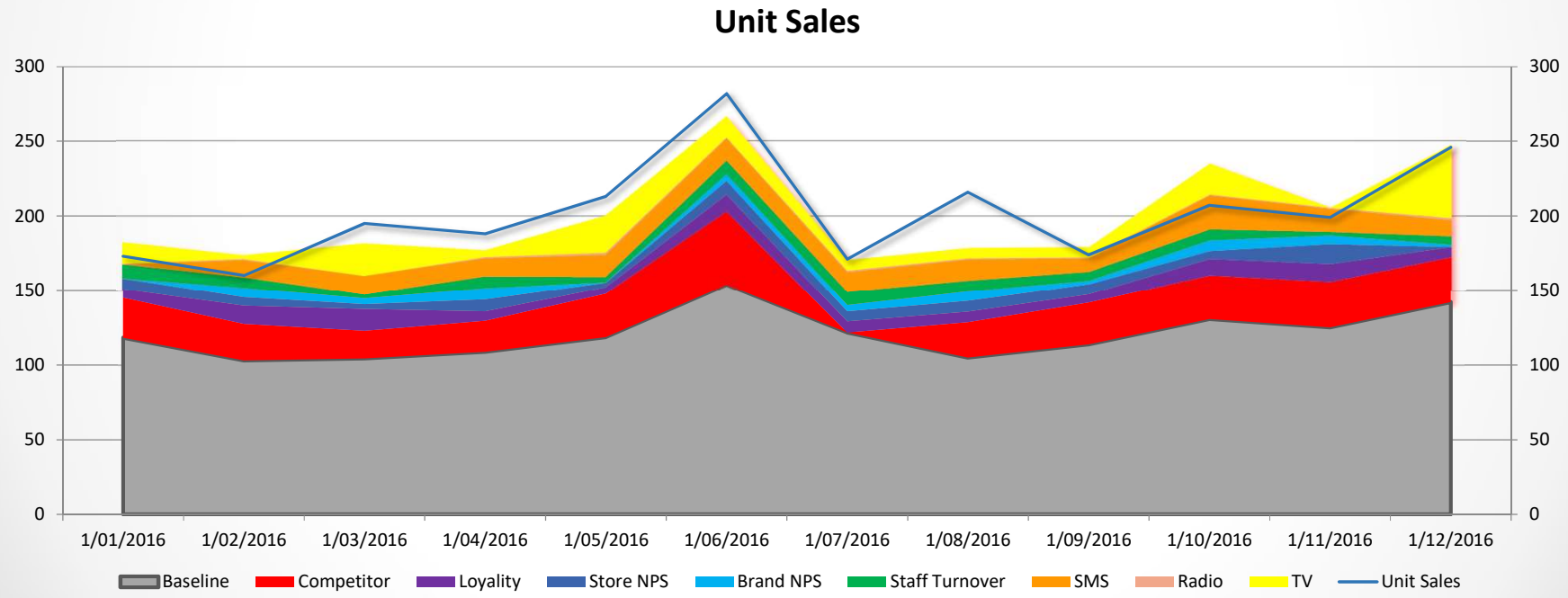
Second Steps: Establish what influence can be measured (and what cannot).

- what influences could we possibly determine?



Third Steps: Simple presentation

- because we are forecasting a core metric (unit sales) then our influences give us direct ROI



Data

- it is 'real' data but names and meaning of the columns have been anonymised.

DT	SALES_CNT	COMPETITOR_CNT	LOYALITY_PERCENT	STORE_NPS_SCORE_AVG	BRAND_NPS_SCORE_AVG	STAFF_TURNOVER_PERCENT	SMS_CAMPAIGN_CONTACT_COUNT	RADIO_ADVERTISING	REPUR_SALES_TV_ADVERTISING
1/01/2016	173	195	78.78%	0.94	0.91	2.1%	0.00	0.25	5.22
1/02/2016	160	200	77.37%	0.94	0.89	0.0%	0.00	0.26	8.13
1/03/2016	195	168	75.51%	0.96	0.90	4.1%	0.01	0.19	5.41
1/04/2016	188	150	75.64%	0.94	0.87	0.0%	0.01	0.21	7.69
1/05/2016	213	166	76.44%	0.97	0.91	4.0%	0.02	0.25	4.62
1/06/2016	282	233	76.13%	0.96	0.92	0.0%	0.01	0.23	5.19
1/07/2016	171	96	75.95%	0.95	0.92	0.0%	0.03	0.26	6.67
1/08/2016	216	94	75.90%	0.95	0.90	0.0%	0.04	0.26	8.29
1/09/2016	174	117	76.30%	0.94	0.92	0.0%	0.03	0.26	9.93
1/10/2016	207	115	75.81%	0.95	0.90	0.0%	0.06	0.24	8.67
1/11/2016	199	135	74.88%	0.91	0.90	5.4%	0.07	0.23	12.28
1/12/2016	246	132	75.36%	0.96	0.93	7.4%	0.05	0.25	4.67

Data

Dependent variable (target)

Independent variables (regressors)

DT	SALES_CNT	COMPETITOR_CNT	LOYALITY_PERCENT	STORE_NPS_SCORE_AVG	BRAND_NPS_SCORE_AVG	STAFF_TURNOVER_PERCENT	SMS_CAMPAIGN_CONTACT_COUNT	RADIO_ADVERTISING	REPUR_SALES_TV_ADVERTISING
1/01/2016	173	195	78.78%	0.94	0.91	2.1%	0.00	0.25	5.22
1/02/2016	160	200	77.37%	0.94	0.89	0.0%	0.00	0.26	8.13
1/03/2016	195	168	75.51%	0.96	0.90	4.1%	0.01	0.19	5.41
1/04/2016	188	150	75.64%	0.94	0.87	0.0%	0.01	0.21	7.69
1/05/2016	213	166	76.44%	0.97	0.91	4.0%	0.02	0.25	4.62
1/06/2016	282	233	76.13%	0.96	0.92	0.0%	0.01	0.23	5.19
1/07/2016	171	96	75.95%	0.95	0.92	0.0%	0.03	0.26	6.67
1/08/2016	216	94	75.90%	0.95	0.90	0.0%	0.04	0.26	8.29
1/09/2016	174	117	76.30%	0.94	0.92	0.0%	0.03	0.26	9.93
1/10/2016	207	115	75.81%	0.95	0.90	0.0%	0.06	0.24	8.67
1/11/2016	199	135	74.88%	0.91	0.90	5.4%	0.07	0.23	12.28
1/12/2016	246	132	75.36%	0.96	0.93	7.4%	0.05	0.25	4.67

SAS High Performance Forecasting

There is a three step process to implementing UCM in SAS high performance forecasting;

1) Specify the forecast model properties

2) Build the forecast model using historical data

3) Score the forecast model using data (old or new)

3B) Manipulate the outputted data into something we can use

Let's Code: forecast model properties

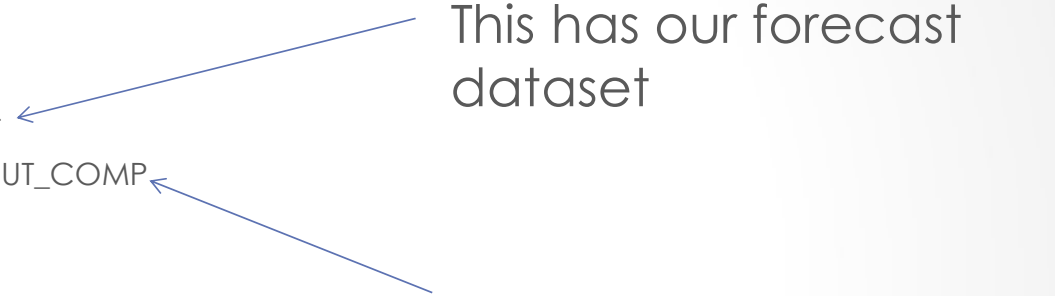
```
PROC HPFUCMSPEC
REPOSITORY = WORK.UCM
NAME=UCM_MODEL;
FORECAST SYMBOL = NEW_UNITS;
INPUT VAR = COMPETITOR_CNT;
INPUT VAR = LOYALITY_PERCENT;
INPUT VAR = STORE_NPS_SCORE_AVG;
INPUT VAR = BRAND_NPS_SCORE_AVG;
INPUT VAR = STAFF TURNOVER PERCENT;
INPUT VAR = SMS_CAMPAIGN_CONTACT_COUNT;
INPUT VAR = RADIO_ADVERTISING;
INPUT VAR = REPUR_SALES_TV_ADVERTISING;
CYCLE ;
IRREGULAR;
LEVEL;
SLOPE VARIANCE=1 NOEST;
SEASON LENGTH = 12 TYPE=TRIG;
RUN;
```

Let's Code: build forecast model

```
PROC HPFDIAGNOSE
DATA = SNUG.SALES_CNT_INPUT
OUTEST = SNUG.SALES_CNT_OUTPUT_EST
MODELREPOSITORY = work.ucm
CRITERION = MAPE
SEASONALITY=12;
BY SHOP_ID;
ID DT INTERVAL = MONTH;
FORECAST SALES_CNT;
INPUT  COMPETITOR_CNT / REQUIRED=YES ;
INPUT  LOYALITY_PERCENT / REQUIRED=YES ;
INPUT  STORE_NPS_SCORE_AVG / REQUIRED=YES ;
INPUT  BRAND_NPS_SCORE_AVG / REQUIRED=YES ;
INPUT  STAFF_TURNOVER_PERCENT / REQUIRED=YES;
INPUT  SMS_CAMPAIGN_CONTACT_COUNT / REQUIRED=YES;
INPUT  RADIO_ADVERTISING / REQUIRED=YES;
INPUT  REPUR_SALES_TV_ADVERTISING / REQUIRED=YES;
RUN;
```

Let's Code: score forecast model

```
PROC HPFENGINE
OUT= _NULL_
DATA = SNUG.SALES_CNT_INPUT
INEST = SNUG.SALES_CNT_OUTPUT_EST
OUTEST = SNUG.SALES_CNT_OUTPUT_EST2
OUTFOR = SNUG.SALES_CNT_OUTPUT_FCAST
OUTCOMPONENT = SNUG.SALES_CNT_OUTPUT_COMP
MODELREPOSITORY = work.ucm
LEAD=7 BACK=1;
BY SHOP_ID;
ID DT INTERVAL = MONTH;
FORECAST SALES_CNT;
INPUT COMPETITOR_CNT;
INPUT LOYALITY_PERCENT;
INPUT STORE_NPS_SCORE_AVG;
INPUT BRAND_NPS_SCORE_AVG;
INPUT STAFF TURNOVER PERCENT;
INPUT SMS_CAMPAIGN_CONTACT_COUNT;
INPUT RADIO_ADVERTISING;
INPUT REPUR_SALES_TV_ADVERTISING;
RUN;
```



Let's Code: prepare output

Transpose output from forecast model – COMP dataset

DT	_COMP_	_PREDICT_
1/01/2016	LEVEL	73.39051
1/01/2016	TREND	0
1/01/2016	SEASON	0
1/01/2016	MU	0
1/01/2016	STATIONARY	0
1/01/2016	Y	44.96716
1/01/2016	COMPETITOR_CNT	26.96649
1/01/2016	LOYALITY_PERCENT	5.801227
1/01/2016	STORE_NPS	7.085543
1/01/2016	BRAND_NPS	0
1/01/2016	STAFFTURNOVER	9.550266
1/01/2016	SMS	0
1/01/2016	RADIO	0.384287
1/01/2016	ADVERTISING	14.50047
1/02/2016	LEVEL	70.56649
1/02/2016	TREND	0
1/02/2016	SEASON	0
1/02/2016	MU	0
1/02/2016	STATIONARY	0
1/02/2016	Y	32.66365
1/02/2016	COMPETITOR_CNT	24.38341
1/02/2016	LOYALITY_PERCENT	12.34883
1/02/2016	STORE_NPS	5.896472
1/02/2016	BRAND_NPS	6.01708
1/02/2016	STAFFTURNOVER	7.191962
1/02/2016	SMS	11.647
1/02/2016	RADIO	0.787196
1/02/2016	ADVERTISING	2.479954

```
PROC TRANSPOSE DATA=SNUG.SALES_CNT_OUTPUT_FCAST
OUT=SNUG.SALES_CNT_OUTPUT_FCAST_TRANS
NAME=Source
LABEL=Label;
BY DEALER_ID DT;
ID _COMP_;
VAR _PREDICT_;
RUN;
QUIT;
```

DT	LEVEL	TREND	SEASON	MU	STATIONARY
1/01/2016	73.39051	0	0	0	0
1/02/2016	70.56649	0	0	0	0

Let's Code: add components to get forecast

If you are lucky… just add up the columns to get your forecast.

- Compare to the SNUG.SALES_CNT_OUTPUT_FCAST dataset

Trend, cycle etc components							Independent components (regressors)						
FCST_SALES_CNT_LEVEL	FCST_SALES_CNT_TREND	FCST_SALES_CNT_SEASON	FCST_SALES_CNT_MU	FCST_SALES_CNT_STATIONARY	FCST_SALES_CNT_Y	FCST_SALES_CNT_COMPETITOR_CNT	FCST_SALES_CNT_LOYALITY_PERCENT	FCST_SALES_CNT_STORE_NPS	FCST_SALES_CNT_BRAND_NPS	FCST_SALES_CNT_STAFFTURNOVER	FCST_SALES_CNT_SMS	FCST_SALES_CNT_RADIO	FCST_SALES_CNT_TV_ADVERTISING
73.39051	0	0	0	0	44.96716	26.96649	5.801227	7.085543	0	9.550266	0	0.384287	14.50047

= 182.65

You may not be lucky (this may be because there are significant events, level shift, some variables have a negative influence, or other challenges)

Recode: go back, data prep or rescale

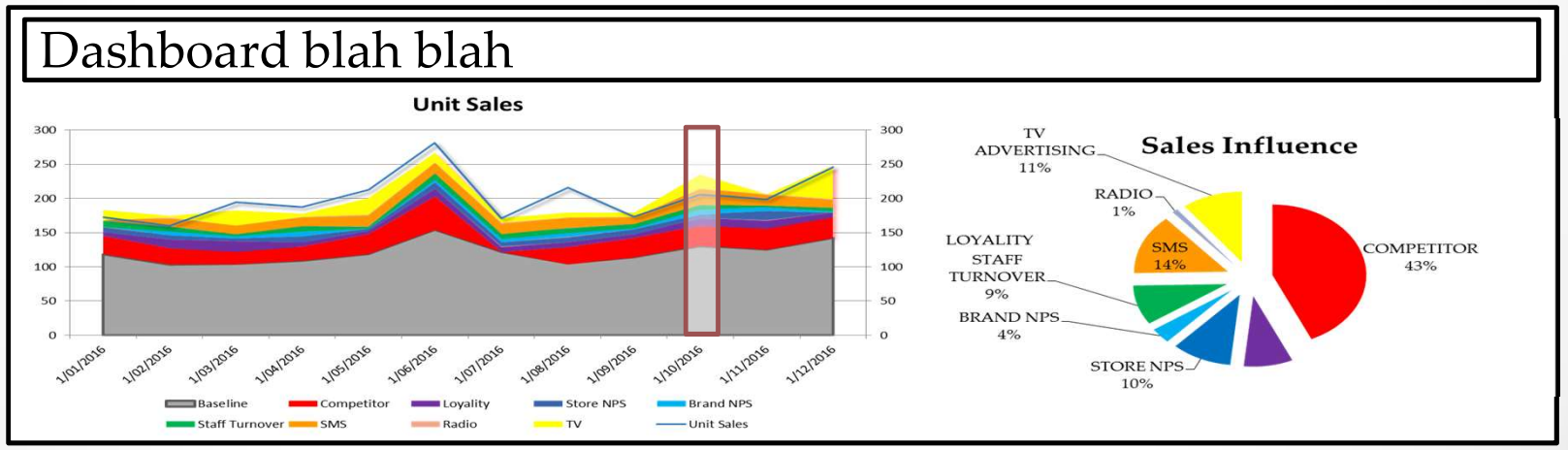
If you are unlucky.

- Consider transforming your input variables
- Remove outliers (and other obvious data preparation steps you might have missed)
- If all fails, a lazy fix:
  Sum the components, convert each to a percentage, then multiple by the forecast (from SNUG.SALES_CNT_OUTPUT_FCAST).
  Example:

73.4	0.0	0.0	0.0	0.0	45.0	27.0	5.8	7.1	0.0	9.6	0.0	0.4	14.5
40.2%	0.0%	0.0%	0.0%	0.0%	24.6%	14.8%	3.2%	3.9%	0.0%	5.2%	0.0%	0.2%	7.9%

Final Steps

1) Consider analysing the differences between the influence of input variables over time. When was a specific channel or input most influential?
2) Create a fancy dashboard - yes, even a piechart showing influence for selected time period.



Questions

Tim Manns
https://www.linkedin.com/in/tim-manns