This document provides a demo that shows how our scripts perform. The examples are obtained from section **4.3 Regression and correlation**, section **5.2 Some useful predictors**, and section **8.9 Seasonal ARIMA models**. You can find the full scripts on GitHub by following **sas-viya-forecasting/Forecasting-Cookbook/Scripts/** or by clicking on the link https://github.com/sassoftware/sas-viya-forecasting/tree/master/Forecasting-Cookbook/Scripts.

Before running the scripts, you need to define a library (**time**) that contains your SAS datasets. Moreover, you need to set up a **CAS** session and link a library (**mycas**) to it.

Example 1: Regression and correlation

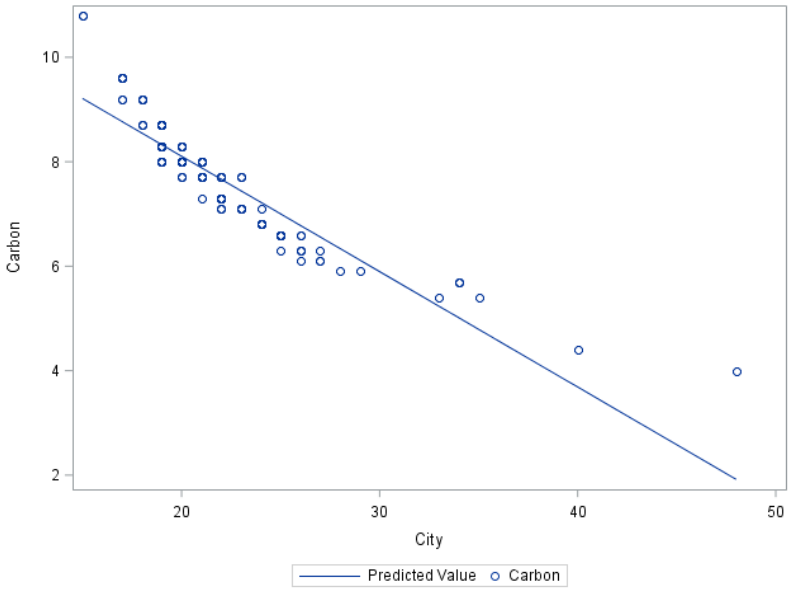
```
data mycas.fpp_fuel;
    set time.fpp_fuel;
run;

*run Viya enabled regression PROC REGSELECT;
proc regselect data=mycas.fpp_fuel;

    *model option clb requests confidence limits;
    model carbon = city /clb;

    *output the residules and copy the predictor variable City to the output
table;
    output out=mycas.fpp_fuel_residual predicted residual
           copyvars = (city carbon);
run;

proc sgplot data = mycas.fpp_fuel_residual;
    series x = City y=Pred;
    scatter x = City y=carbon;
run;
```

Parameter Estimates, and its relevant statistics are displayed in the output:

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|	95% Confidence Limits	
Intercept	1	12.525647	0.199232	62.87	<.0001	12.13155	12.91975
City	1	-0.220970	0.008878	-24.89	<.0001	-0.23853	-0.20341

The following is the plot of the fitted values and actual values of Carbon with respect to the city levels.



Example 2: Some useful predictors

```
data mycas.fpp_ausbeer;
    set time.fpp_ausbeer;
    keep date aus_beer;
    where 1992 <= year(date) < 2006;
run;
```

The code coming below shows how to create quarterly dummy variables for the Australian quarterly beer production regression model. The regression model is:

$$y_t = \beta_0 + \beta_1 t + \beta_2 d_{2,t} + \beta_3 d_{3,t} + \beta_4 d_{4,t} + e_t$$

where $d_{i,t} = 1$ if time t is in quarter i ($i = 2,3,4$) and takes zero otherwise.

```
proc tsmodel data=mycas.fpp_ausbeer
             outarray = mycas.fpp_ausbeer;
    id date interval=quarter;
    var aus_beer;
    outarrays q1 q2 q3 q4;
    submit;
        do i = 1 to dim(aus_beer);
            *initialize outarrays to 0's;
            q1[i] = 0; q2[i] = 0; q3[i] = 0; q4[i] = 0;

            *set outarray q's based on the pre-defined array _season_;
            if _season_[i] = 1 then q1[i] = 1;
            else if _season_[i] = 2 then q2[i] = 1;
            else if _season_[i] = 3 then q3[i] = 1;
            else q4[i] = 1;
        end;
    endsubmit;
quit;
```

The following is the first 10 observations of **mycas.fpp_ausbeer** table:

Obs	_STATUS_	_SEASON_	_CYCLE_	Date	Aus_Beer	q1	q2	q3	q4
1	0	1	1	1992:1	443	1	0	0	0
2	0	2	2	1992:2	410	0	1	0	0
3	0	3	3	1992:3	420	0	0	1	0
4	0	4	4	1992:4	532	0	0	0	1
5	0	1	5	1993:1	433	1	0	0	0
6	0	2	6	1993:2	421	0	1	0	0
7	0	3	7	1993:3	410	0	0	1	0
8	0	4	8	1993:4	512	0	0	0	1
9	0	1	9	1994:1	449	1	0	0	0
10	0	2	10	1994:2	381	0	1	0	0

We can build the regression model based on the information in the **mycas.fpp_ausbeer** table.

```
* no need to define trend variable. proc tsmodel will generate _cycle_
variable which addresses time;
proc regselect data=mycas.fpp_ausbeer;
      model aus_beer = _cycle_ q2 q3 q4;
      output out=mycas.fpp_ausbeer_fitted
      copyvars=(aus_beer date) residual;
run;
```

The **Parameter Estimates** is produced in the output:

Parameter Estimates					
Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	441.814148	4.533798	97.45	<.0001
CYCLE	1	-0.382005	0.107795	-3.54	0.0009
q2	1	-34.046566	4.917386	-6.92	<.0001
q3	1	-18.093132	4.920929	-3.68	0.0006
q4	1	76.074588	4.926829	15.44	<.0001

Example 3: Seasonal ARIMA models

```
data mycas.fpp_h02_train;
      set time.fpp_h02;
      if (Date<='01JUN2006'd) then output;
run;

*The ARIMA(3,0,0)(2,1,0)[12] configuration was taken from the book example;
proc carima data=MYCAS.FPP_H02_TRAIN outstat=MYCAS.outStatTemp
            outfor=MYCAS.predict1;
      id Date interval=month;
      identify h02;
      estimate p=(1 2 3) (12 24) diff=(12) noint transform=log method=ML;
      forecast lead=24 alpha=0.05;
run;
```

The **outfor** table created in this step provides some statistics for the current data (fitted values and confidence intervals) and prediction values for 24 months ahead. The last 30 observations of this table are as follows:

Obs	_NAME_	_MODEL_	Date	ACTUAL	PREDICT	ERROR	STD	UPPER	LOWER
175	H02	arimaModel0	JAN2006	1.230691	1.1000558315	0.1306351685	0.0675070376	1.2381835992	0.9736704077
176	H02	arimaModel0	FEB2006	0.587135	0.5953183759	-0.008183376	0.0365328548	0.6700691257	0.5269222426
177	H02	arimaModel0	MAR2006	0.706959	0.6650755265	0.0418834735	0.0408136363	0.7485852859	0.5886649937
178	H02	arimaModel0	APR2006	0.639641	0.6969119224	-0.057270922	0.0427673378	0.7844191973	0.6168437057
179	H02	arimaModel0	MAY2006	0.807405	0.7320578281	0.0753471719	0.0449241338	0.8239781749	0.647951698
180	H02	arimaModel0	JUN2006	0.79797	0.8560769972	-0.058106997	0.0525348081	0.9635697272	0.7577223037
181	H02	arimaModel0	JUL2006	.	0.9174032556	.	0.0562982117	1.0325963759	0.8120027878
182	H02	arimaModel0	AUG2006	.	1.0102508762	.	0.0629238849	1.1390808999	0.8925289843
183	H02	arimaModel0	SEP2006	.	1.098455281	.	0.0747372576	1.2520656451	0.9592500917
184	H02	arimaModel0	OCT2006	.	1.0977774641	.	0.0821228235	1.2673324967	0.9456150914
185	H02	arimaModel0	NOV2006	.	1.1414512739	.	0.0885647274	1.3246444816	0.9777070417
186	H02	arimaModel0	DEC2006	.	1.2033849378	.	0.0985790397	1.4078764158	1.0217390126
187	H02	arimaModel0	JAN2007	.	1.221353323	.	0.104331642	1.4382771407	1.029633066
188	H02	arimaModel0	FEB2007	.	0.6231762873	.	0.0550452363	0.7378430004	0.5222549571
189	H02	arimaModel0	MAR2007	.	0.7314243121	.	0.066825217	0.8709058766	0.6091965801
190	H02	arimaModel0	APR2007	.	0.6837343545	.	0.0642595398	0.8180891328	0.5664413748
191	H02	arimaModel0	MAY2007	.	0.8140742727	.	0.0784946888	0.9784513984	0.671072945
192	H02	arimaModel0	JUN2007	.	0.832887063	.	0.0822323025	1.0053480025	0.6833494343
193	H02	arimaModel0	JUL2007	.	0.9593467248	.	0.1111269643	1.194958356	0.7599933832
194	H02	arimaModel0	AUG2007	.	1.0241669641	.	0.1218303926	1.2829782049	0.80615744
195	H02	arimaModel0	SEP2007	.	1.1306465058	.	0.1410225378	1.4313088374	0.879459959
196	H02	arimaModel0	OCT2007	.	1.1421428539	.	0.1494020969	1.4618701997	0.8773315299
197	H02	arimaModel0	NOV2007	.	1.1870731128	.	0.1598911543	1.5300641966	0.9045587401
198	H02	arimaModel0	DEC2007	.	1.2420572619	.	0.1729699872	1.6141391811	0.9375627789
199	H02	arimaModel0	JAN2008	.	1.2288372288	.	0.1760340782	1.6084271964	0.9199521545
200	H02	arimaModel0	FEB2008	.	0.6256424994	.	0.0918357475	0.8240938597	0.4649623484
201	H02	arimaModel0	MAR2008	.	0.7184979744	.	0.1079669619	0.9522949516	0.5301297432
202	H02	arimaModel0	APR2008	.	0.6911669527	.	0.1060316611	0.9212034784	0.5066497927
203	H02	arimaModel0	MAY2008	.	0.7895210132	.	0.1234398319	1.057791799	0.5752262622
204	H02	arimaModel0	JUN2008	.	0.8508933903	.	0.1353928769	1.1456234087	0.616381428

The next step is to put the prediction and actual values for after **Jul2006** in order to compute the root mean square error (RMSE):

```
* This section shows how to compute the RMSE based on the above model
forecast and observations in testing set;
data work.fpp_h02_testing;
    set time.fpp_h02;
    if (Date>'01JUN2006'd) then output;
    keep Date h02;
run;

data mycas.fpp_h02_testing;
    set work.fpp_h02_testing;
run;

data MYCAS.forecast;
    set MYCAS.predict1;
    if (Date>'01JUN2006'd) then output;
    keep Date PREDICT;
run;

data MYCAS.forecast;
    merge MYCAS.forecast MYCAS.fpp_h02_testing;
    by Date;
```

```
run;

data work.forecast;
    set MYCAS.forecast;
    label H02='Actual Values';
run;

proc sort data = work.forecast;
    by Date;
run;
```

The following is only the last year values in the **forecast** table, containing both actual and prediction values of **H02** variable for after **Jul2006**:

Obs	Time ID Values	Predicted Values	Actual Values
13	JUL2007	0.9593467248	0.954144
14	AUG2007	1.0241669641	1.07821949
15	SEP2007	1.1306465058	1.11098161
16	OCT2007	1.1421428539	1.10997914
17	NOV2007	1.1870731128	1.1635343
18	DEC2007	1.2420572619	1.176589
19	JAN2008	1.2288372288	1.219941
20	FEB2008	0.6256424994	0.761822
21	MAR2008	0.7184979744	0.649435
22	APR2008	0.6911669527	0.827887
23	MAY2008	0.7895210132	0.816255
24	JUN2008	0.8508933903	0.762137

Finally, RMSE is evaluated by a **data** step:

```
data work.RMSE;
    retain so_far;
    set work.forecast end=last;
    error=PREDICT-h02;
    square_error=error*error;
    if _n_ eq 1 then so_far = square_error;
        else so_far=so_far+ square_error;
    if last then so_far = sqrt(so_far/_n_);
    keep so_far;
    label so_far='RMSE';
    if last then output;
run;
```

Here is the value of **RMSE**:

Obs	RMSE
1	0.066814