

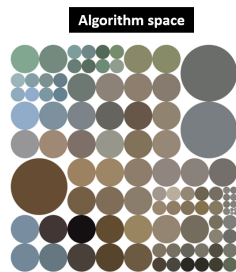
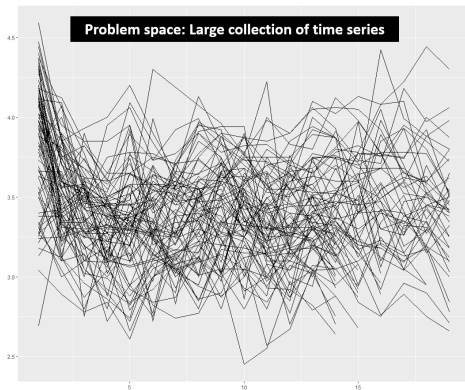


# Feature-based Time Series Forecasting

Thiyanga Talagala,  
Rob J Hyndman, George Athanasopoulos  
Feng Li, Yanfei Kang

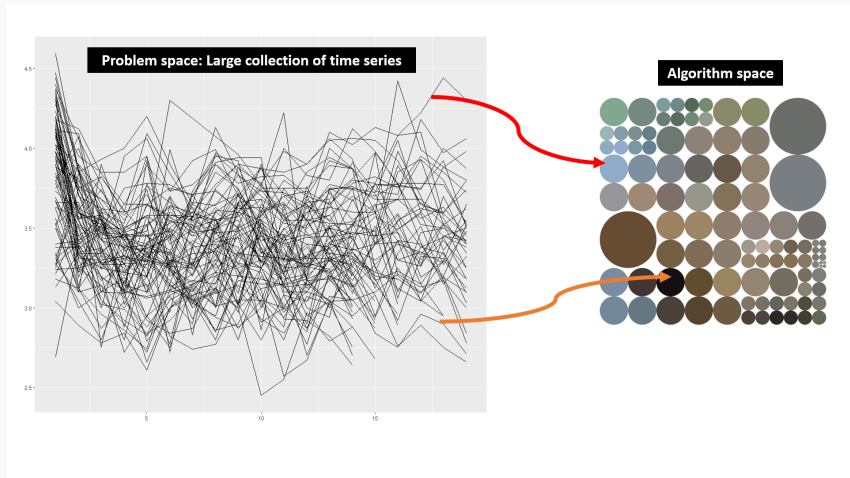
11 July 2019

# Big picture



- What algorithm is likely to perform best?

# Big picture



- What algorithm is likely to perform best?
- Algorithm selection problem, John Rice (1976)

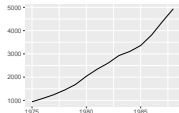
## Time series features

- Transform a given time series  $y = \{y_1, y_2, \dots, y_n\}$  to a feature vector  $F = (f_1(y), f_2(y), \dots, f_p(y))'$ .

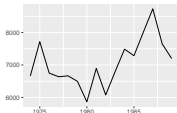
# Time series features

- Transform a given time series  $y = \{y_1, y_2, \dots, y_n\}$  to a feature vector  $F = (f_1(y), f_2(y), \dots, f_p(y))'$ .

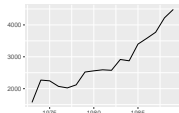
**N0001**



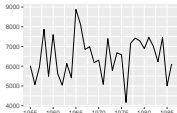
**N0633**



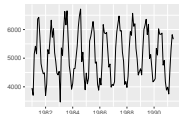
**N0625**



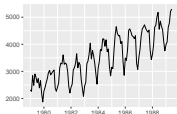
**N0645**



**N1912**



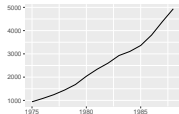
**N2012**



# Time series features

- Transform a given time series  $y = \{y_1, y_2, \dots, y_n\}$  to a feature vector  $F = (f_1(y), f_2(y), \dots, f_p(y))'$ .

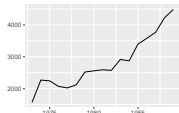
**N0001**



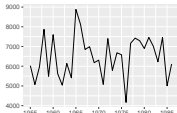
**N0633**



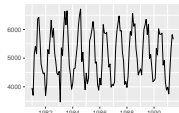
**N0625**



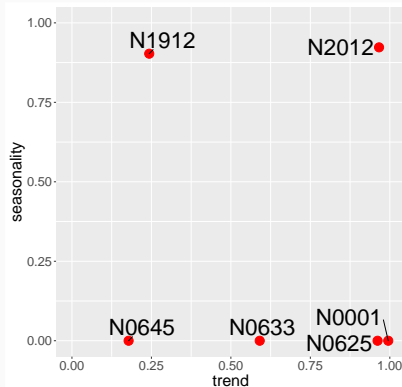
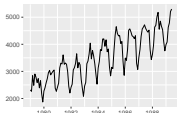
**N0645**



**N1912**



**N2012**



## More features

- length
- strength of seasonality
- strength of trend
- linearity
- curvature
- spikiness
- stability
- lumpiness
- spectral entropy
- Hurst exponent
- nonlinearity
- unit root test statistics
- parameter estimates of Holt's linear trend method
- parameter estimates of Holt-Winters' additive method
- ACF and PACF based features - calculated on raw, differenced, seasonally-differenced series and remainder series.

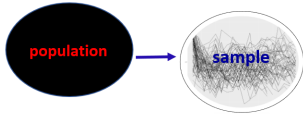
# Algorithm selection framework



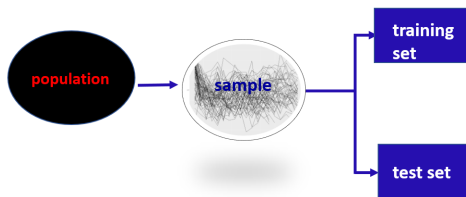
population



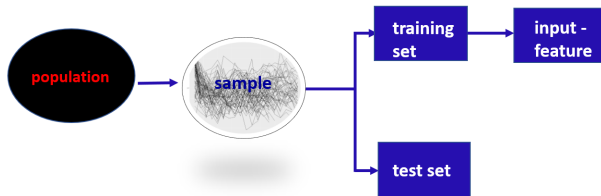
# Algorithm selection framework



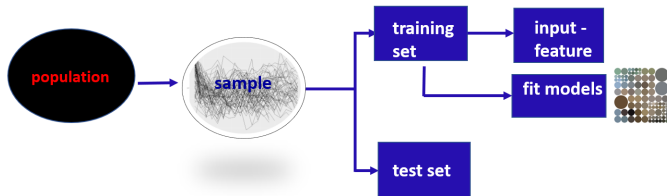
# Algorithm selection framework



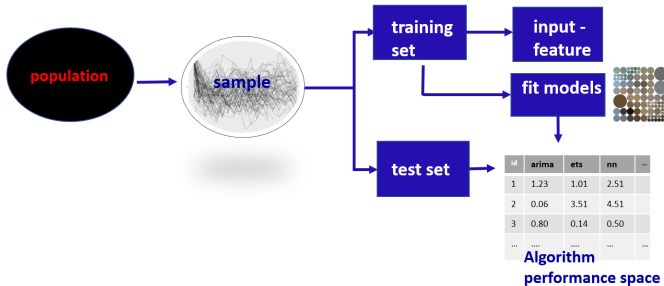
# Algorithm selection framework



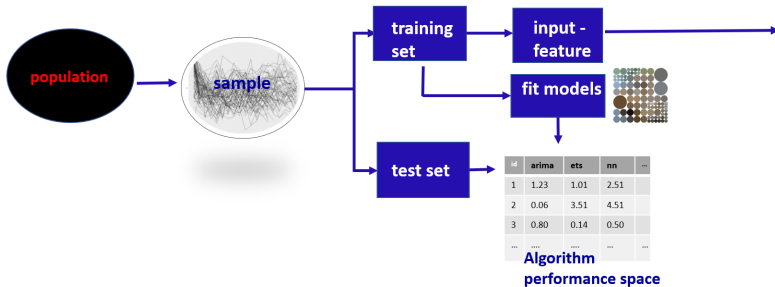
# Algorithm selection framework



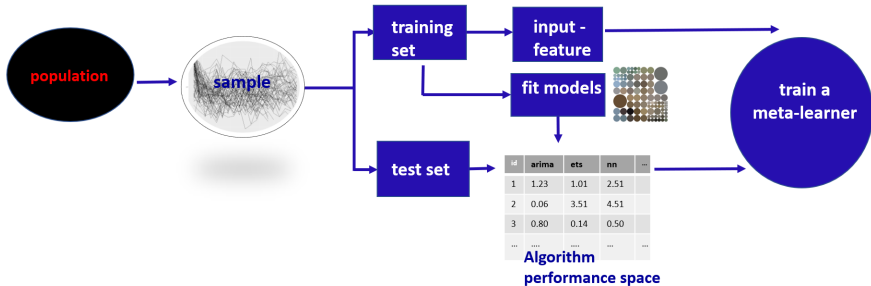
# Algorithm selection framework



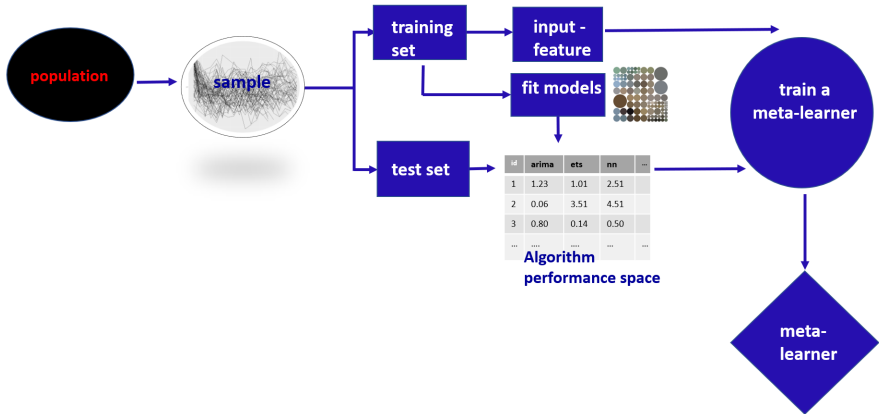
# Algorithm selection framework



# Algorithm selection framework

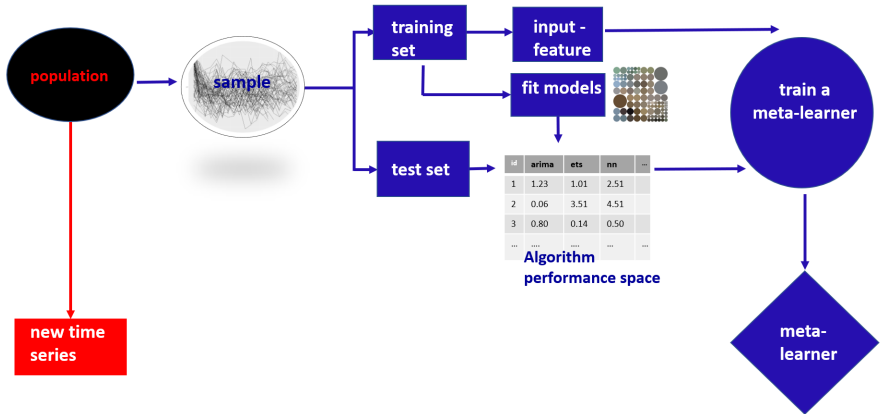


# Algorithm selection framework

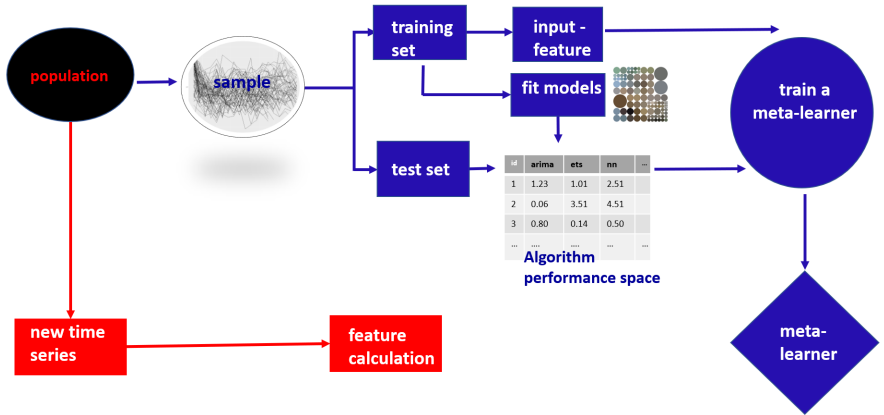




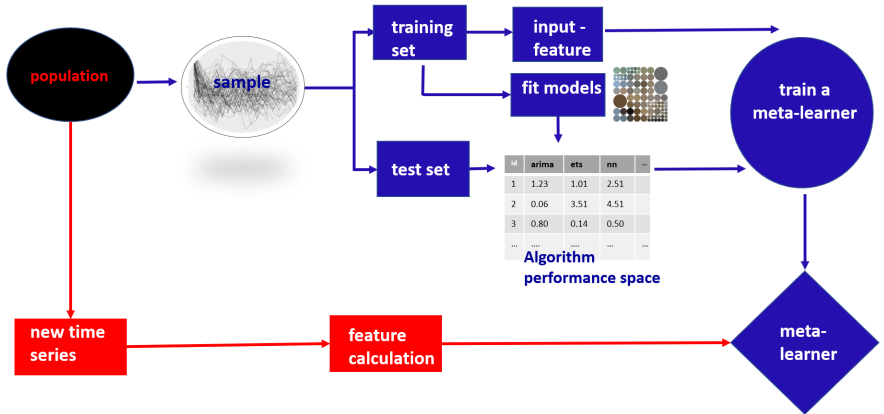
# Algorithm selection framework



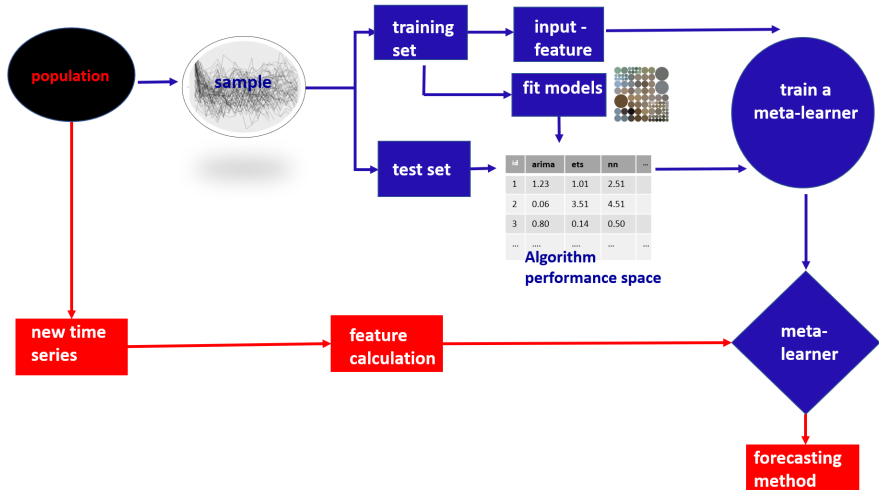
# Algorithm selection framework



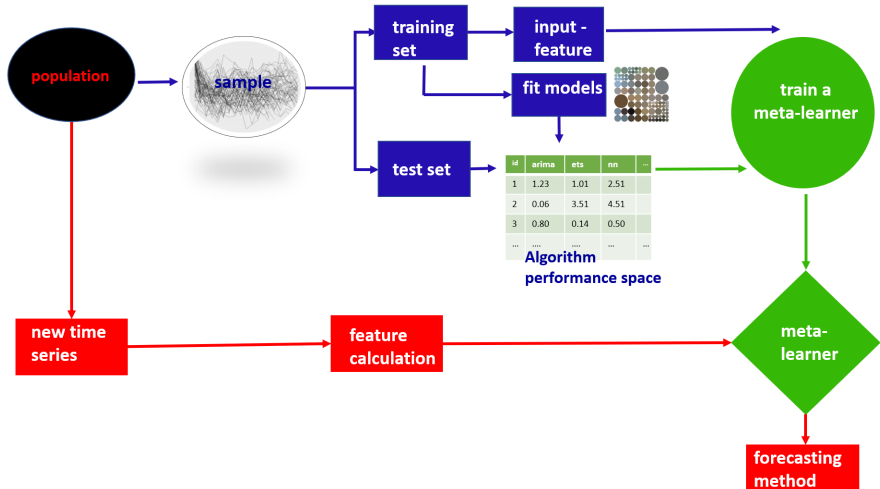
# Algorithm selection framework



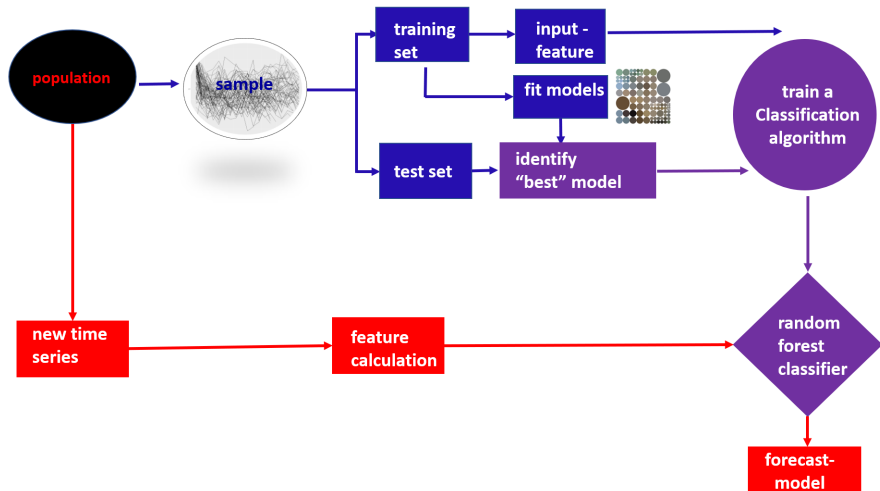
# Algorithm selection framework



# Algorithm selection framework

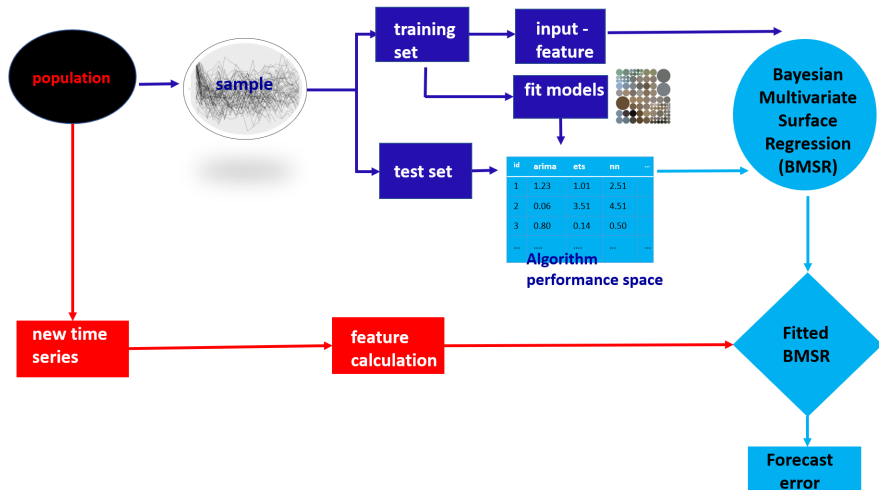


# FFORMS: Feature-based FOREcast Model Selection



■ three algorithms: **FFORMS**, FFORMPP

# FFORMPP: Feature-based FOREcast Model Performance Prediction



■ three algorithms: **FFORMS**, **FFORMPP**

## Installation

```
devtools::install_github("thiyanagt/seer")  
library(seer)
```







## Installation

```
devtools::install_github("thiyanagt/seer")  
library(seer)
```

## Example datasets

**observed time series - M1 yearly series (181)**

```
library(Mcomp)  
yearlym1 <- subset(M1, "yearly")
```



## Installation

```
devtools::install_github("thiyanagt/seer")  
library(seer)
```

## Example datasets

**observed time series - M1 yearly series (181)**

```
library(Mcomp)  
yearlym1 <- subset(M1, "yearly")
```

# Input: features

```
cal_features(yearlym1[1:3], database="M1",  
h=6, highfreq=FALSE)
```

```
# A tibble: 2 x 25  
  entropy lumpiness stability hurst trend spikiness linearity curvature  
  <dbl>    <dbl>    <dbl> <dbl> <dbl>    <dbl>    <dbl>    <dbl>  
1  0.683    0.0400    0.977 0.985 0.985    1.32e-6    4.46    0.705  
2  0.711    0.0790    0.894 0.988 0.989    1.54e-6    4.47    0.613  
# ... with 17 more variables: e_acf1 <dbl>, y_acf1 <dbl>,  
# diff1y_acf1 <dbl>, diff2y_acf1 <dbl>, y_pacf5 <dbl>,  
# diff1y_pacf5 <dbl>, diff2y_pacf5 <dbl>, nonlinearity <dbl>,  
# lmres_acf1 <dbl>, ur_pp <dbl>, ur_kpss <dbl>, N <int>, y_acf5 <dbl>,  
# diff1y_acf5 <dbl>, diff2y_acf5 <dbl>, alpha <dbl>, beta <dbl>
```

# Input: features

```
seer::fcast_accuracy(tslist=yearlym1[1:2],  
                     models= c("arima","ets","rw", "theta", "nn"),  
                     database ="M1", cal_MASE, h=6,  
                     length_out = 1,  
                     fcast_save = TRUE)
```

\$accuracy

	arima	ets	rw	theta	nn
YAF2	10.527612	10.319029	13.52428	12.088375	11.688241
YAF3	5.713867	7.704409	7.78949	6.225463	6.700765

\$ARIMA

	YAF2	YAF3
"ARIMA(0,1,0) with drift"	"ARIMA(0,1,1) with drift"	

\$ETS

	YAF2	YAF3
"ETS(A,A,N)"	"ETS(M,A,N)"	

\$forecasts

\$forecasts\$arima

	YAF2	YAF3
[1,]	579581.0	390955.9
[2,]	605761.9	407325.1
[3,]	631942.9	423694.4
[4,]	658123.8	440063.6
[5,]	684304.8	456432.8
[6,]	710485.7	472802.0

# Training set

```
prepare_trainingset(accuracy_set = accuracy_m1,  
feature_set = features_m1)$trainingset
```

```
# A tibble: 2 x 26  
  entropy lumpiness stability hurst trend spikiness linearity curvature  
  <dbl>    <dbl>    <dbl> <dbl> <dbl>    <dbl>    <dbl>    <dbl>  
1  0.683    0.0400    0.977 0.985 0.985    1.32e-6    4.46    0.705  
2  0.711    0.0790    0.894 0.988 0.989    1.54e-6    4.47    0.613  
# ... with 18 more variables: e_acf1 <dbl>, y_acf1 <dbl>,  
# diff1y_acf1 <dbl>, diff2y_acf1 <dbl>, y_pacf5 <dbl>,  
# diff1y_pacf5 <dbl>, diff2y_pacf5 <dbl>, nonlinearity <dbl>,  
# lmres_acf1 <dbl>, ur_pp <dbl>, ur_kpss <dbl>, N <int>, y_acf5 <dbl>,  
# diff1y_acf5 <dbl>, diff2y_acf5 <dbl>, alpha <dbl>, beta <dbl>,  
# classlabels <chr>
```

# FFORMS classifier

```
rf <- build_rf(training_set = training_set,  
               testset= M3yearly_features,  
               rf_type="ru", ntree=100, seed=1,  
               import=FALSE, mtry = 8)
```

## Predictions

```
head(rf$predictions)
```

```
##           1           2           3           4           5           6  
## ETS-trend      rwd      rwd      rwd      rwd      rwd  
## 10 Levels: ARIMA ARMA/AR/MA ETS-dampedtrend ... wn
```

## FFORMS classifier

```
rf$randomforest
```

```
## randomForest(formula = classlabels ~ ., data = training_set,  
##             importance = import, ntree = ntree, mtry = mtry)
```

# Pre-trained classifiers

## Load hourly data FFORMS classifier

```
data("hourly_fforms")
```

# Pre-trained classifiers

## Load hourly data FFORMS classifier

```
data("hourly_fforms")
```

## Forecast hourly time series in the M4-competition

```
fcast.models <- predict(hourly_fforms, features_M4H)  
head(fcast.models)
```

```
##      1      2      3      4      5      6  
## tbats      nn stlar stlar      nn      nn  
## Levels: mstlarima mstlets nn rw rwd snaive stlar tba
```




## Yearly: Correlation between MASE values across different forecast-models



# FFORMPP: Feature-based FOREcast Model Performance Prediction

id	seasonality	...	entropy	trend	rw	rwd	...	arma
1	0.89		0.21	0.82	1.02	0.89		0.78
2	0.20		0.82	0.10	1.10	2.81		2.87
N	0.50		0.40	0.30	0.87	0.89		0.99



- Efficient Bayesian Multivariate Surface Regression (Feng Li & Mattias Villani, 2013)
  - ▶ handles interactions and nonlinear relationships
  - ▶ allows the knot locations to move freely in the feature space

## Installation

```
devtools::install_github("thiayangt/fformpp")  
library(fformpp)
```

## Train a model

```
fit_fformpp(feamat=features_mat, accmat=forecast.error,  
            sknots=2, aknots=2,  
            fix.s=0, fix.a=0, fix.shrinkage=1:5,  
            fix.covariance=0,  
            fix.coefficients=0, n.iter=100,  
            knot.moving.algorithm="Random-Walk",  
            ptype=c("identity", "identity", "identity"),  
            prior.knots=100)
```

# FFORMPP: online phase

```
predict.m1 <- predict(fformpp.model, features.m1.df,  
  c("ets", "arima", "rw", "rwd", "wn", "theta", "nn"),  
  log=FALSE, final.estimate=median)  
head(predict.m1)
```

##	ets	arima	rw	rwd	wn	theta	nn
## [1,]	5.015336	5.065616	5.149868	4.293450	16.681046	4.316341	4.554838
## [2,]	1.990880	1.831033	1.830689	2.010443	7.845106	1.434183	2.864783
## [3,]	3.825084	3.284397	3.893353	3.876207	12.867128	3.279123	2.885896
## [4,]	2.169089	3.162256	2.178721	2.481028	3.126736	2.216428	1.832553
## [5,]	5.199962	3.970234	4.630903	4.174412	15.631346	4.101041	5.765485
## [6,]	4.295996	4.494820	5.135292	3.523215	16.085372	4.021210	3.916389

# Results: M4 Competition data

	Yearly	Quarterly	Monthly	Weekly	Daily	Hourly
FFORMS_individual	3.17	1.20	0.98	2.31	3.57	0.84
FFORMPP_combination	3.07	1.13	0.89	2.46	3.62	0.96
auto.arima	3.40	1.17	0.93	2.55	-	-
ets	3.44	1.16	0.95	-	-	-
theta	3.37	1.24	0.97	2.64	3.33	1.59
rwd	3.07	1.33	1.18	2.68	3.25	11.45
rw	3.97	1.48	1.21	2.78	3.27	11.60
nn	4.06	1.55	1.14	4.04	3.90	1.09
stlar	-	2.02	1.33	3.15	4.49	1.49
snaive	-	1.66	1.26	2.78	24.46	2.86
tbats	-	1.19	1.05	2.49	3.27	1.30
wn	13.42	6.50	4.11	49.91	38.07	11.68
mstlarima	-	-	-	-	3.84	1.12
mstlets	-	-	-	-	3.73	1.23
combination (mean)	4.09	1.58	1.16	6.96	7.94	3.93
M4-1st	2.98	1.12	0.88	2.36	3.45	0.89
M4-2nd	3.06	1.11	0.89	2.11	3.34	0.81
M4-3rd	3.13	1.23	0.95	2.16	2.64	0.87

# Thank you

## R packages and papers

### R packages

- **seer**: FFORMS

[github.com/thiyanagt/seer](https://github.com/thiyanagt/seer)

- **fformpp**: FFORMPP

[github.com/thiyanagt/fformpp](https://github.com/thiyanagt/fformpp)

### Papers

Available from [robjhyndman.com](https://robjhyndman.com)

Slides: <https://thiyanaga.netlify.com/talk/user19-talk/>

email: [thiyanaga.talagala@monash.edu](mailto:thiyanaga.talagala@monash.edu)