

MRMR

Brian A. Fannin

14 July 2014

- Flexible framework for reserving based on linear models
- Linear models = Chain ladder and Additive methods
- Emphasis on:
 - Ease of use
 - Multi-dimensional treatment of data
 - Visualization supports exploratory analysis and assessment of model fit

- Leigh Halliwell
- Gelman & Hill, “Data Analysis Using Regression and Multilevel/Hierarchical Models”
- Jim Guszcza
- ggplot2, lubridate, sp

- OriginPeriod
- St*Measures
- Triangle
- TriangleModel
- TriangleProjection

- May be thought of as data frames with constraints and metadata
- [, \$ accessors
- rbind, c, length
- New generics:
 - write.excel
 - Grow

```
op = OriginPeriod(StartDate = as.Date("2001-01-01")
                  , NumPeriods = 10
                  , Period=as.period(1, "years")
                  , Type="Accident Year")
op$Moniker = paste0("AY ", year(op$StartDate))
```

```
head(as.data.frame(op)[, 1:3])
```

StartDate	EndDate	Moniker
2001-01-01	2001-12-31	AY 2001
2002-01-01	2002-12-31	AY 2002
2003-01-01	2003-12-31	AY 2003
2004-01-01	2004-12-31	AY 2004
2005-01-01	2005-12-31	AY 2005
2006-01-01	2006-12-31	AY 2006

Non-standard OriginPeriods

```
startDates = seq(as.Date("2001/01/01")
                 , as.Date("2005/12/31"), by="6 months")
endDates = startDates + as.period(6, "months") - days(1)
op = OriginPeriod(startDates, endDates)
```



```
op = OriginPeriod(seq(2001:2010))
op$Moniker = paste0("AY ", year(op$StartDate))
x = op[1]
y = op[2:3]
z = c(x, y)
y = op[c("AY 2004", "AY 2008")]
y = op$StartDate
y = op$Type
y = op$Moniker[3]
```

- St*ticMeasures contain a set of observations associated with an OriginPeriod
- StaticMeasure objects do not vary over time
 - Earned Premium, Payroll, Number of autos
- StochasticMeasure objects vary over time
 - Paid loss, number of open claims

- Contain a list of Levels of arbitrary complexity
 - Each Level has a name and a set of attributes
 - Ex: Line level may have attributes “GL”, “Motor”, etc.
- Contain a set of Measures
 - StochasticMeasure will automatically compute incremental and prior measures

All examples use NAIC data compiled by Glen Meyers and Peng Shi of ISO. Available at http://www.casact.org/research/index.cfm?fa=loss_reserves_data

StaticMeasure construction

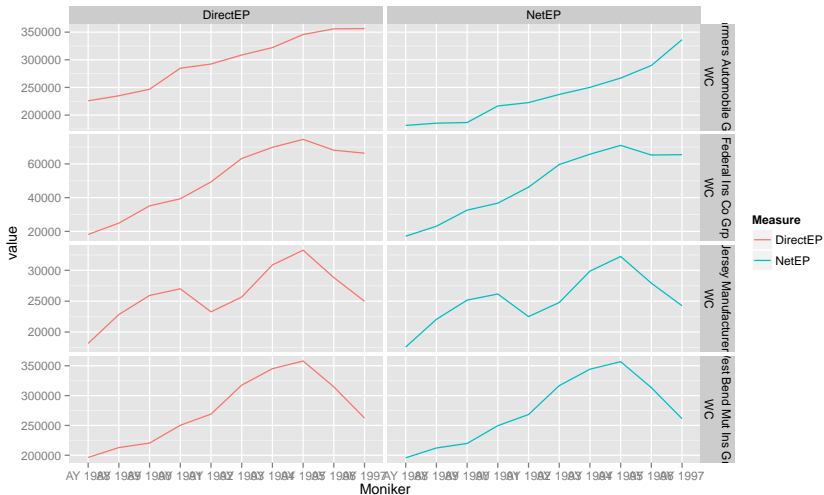
```
smWC =  
  StaticMeasure(OriginPeriod = op  
    , Level=list(Company=companies  
      , Line="WC")  
    , Measure=c("DirectEP", "NetEP")  
    , Data=dfSubWC[dfSubWC$Lag == 1, ])
```

plot(smWC)



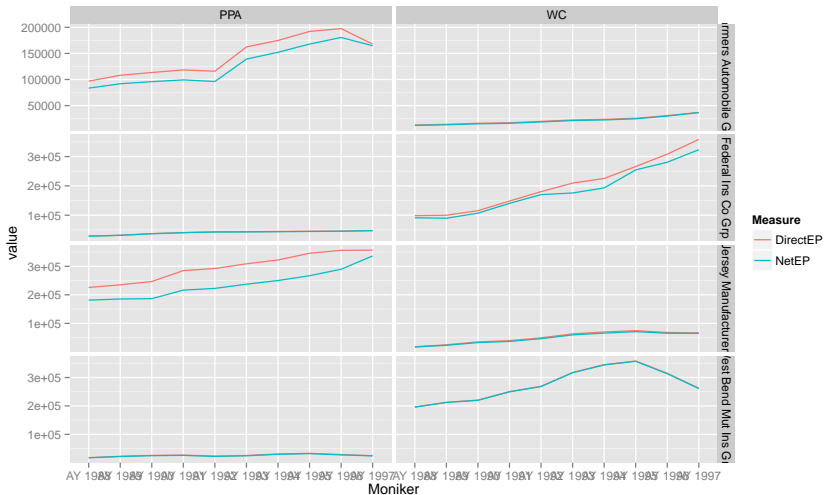
StaticMeasure facet plot

```
plot(smWC  
  , FacetFormula=as.formula(Line+Company~Measure))
```

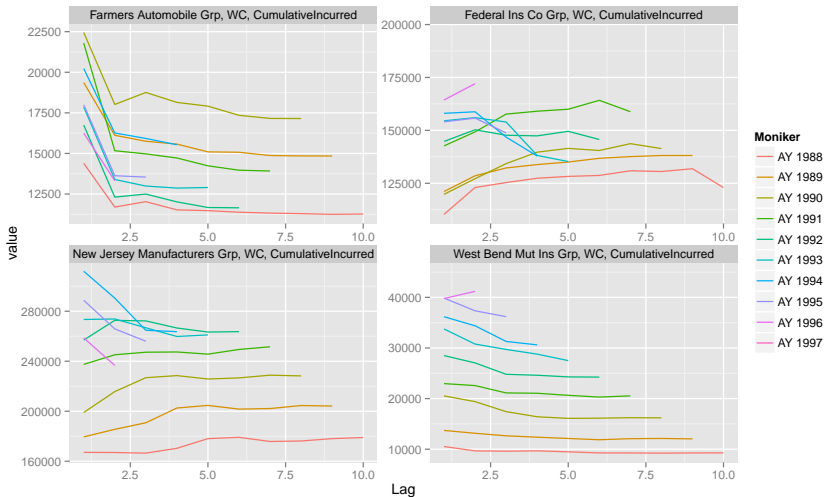


StaticMeasure multidimensional plot

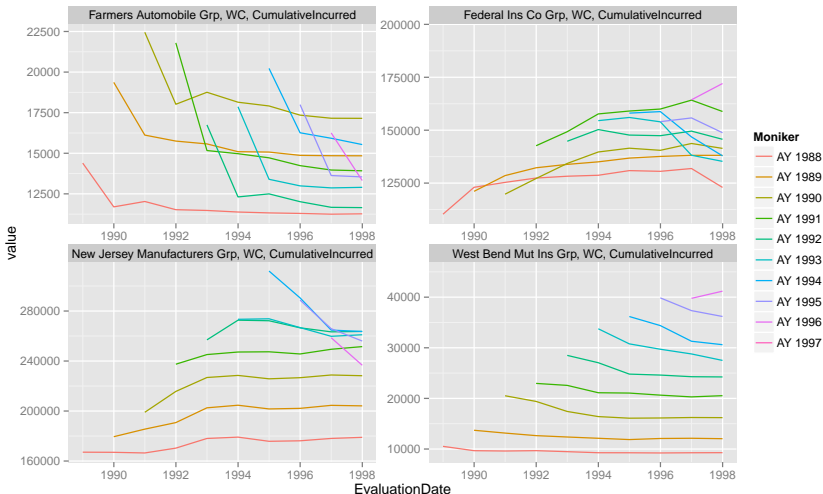
```
plot(smMulti  
  , FacetFormula=as.formula(Company~Line))
```




```
plot(UpperTriangle(scmWC)
     , Measure="CumulativeIncurred")
```



```
plot(UpperTriangle(scmWC)
     , Measure="CumulativeIncurred"
     , TimeAxis="EvaluationDate")
```

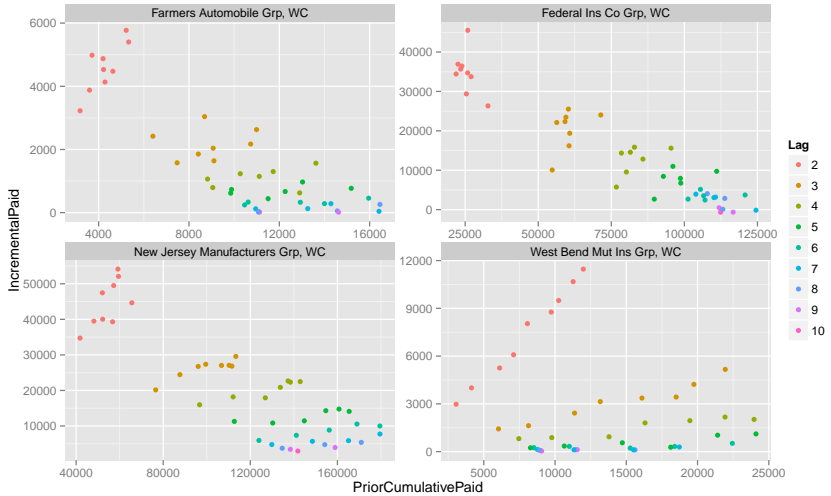


Very simple object. Composed of a StaticMeasure and a StochasticMeasure

```
triWC = Triangle(smWC, scmWC, "Workers Comp Triangle")

plot(UpperTriangle(triWC)
     , Response="IncrementalPaid"
     , Predictor="PriorCumulativePaid")
```

Triangle plot output

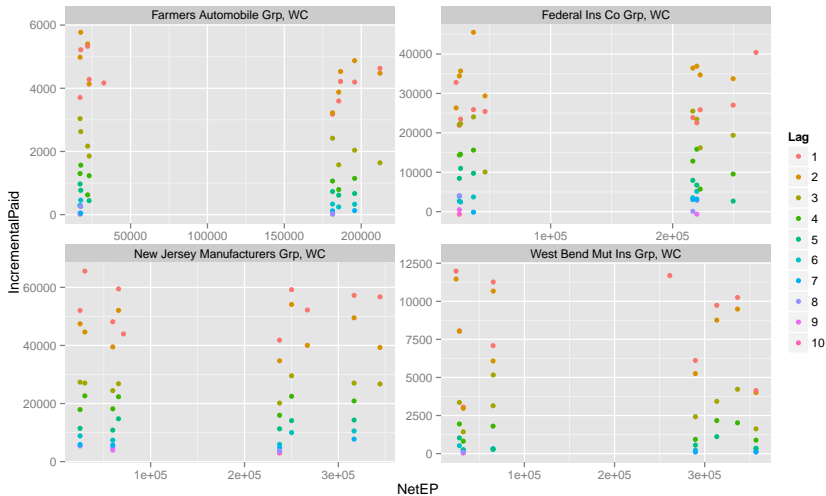


Triangle plot switch variables

```
plot(UpperTriangle(triWC)  
     , Response="IncrementalPaid"  
     , Predictor="PriorCumulativePaid")
```

```
plot(UpperTriangle(triWC)  
     , Response="IncrementalPaid"  
     , Predictor="NetEP")
```

Triangle plot - 2

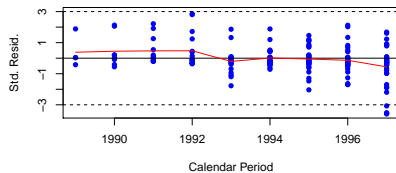
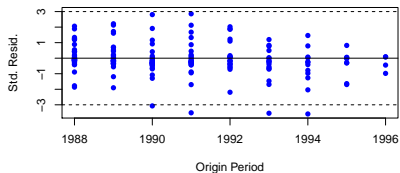
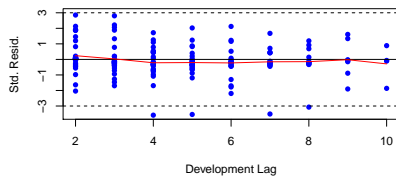
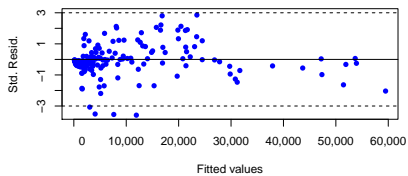


3 Model Types

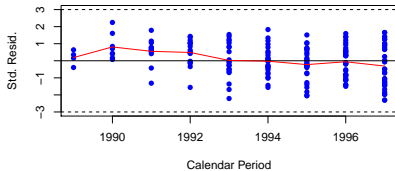
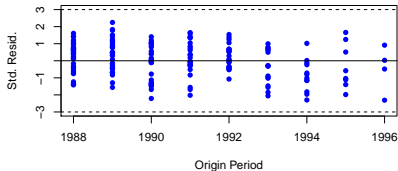
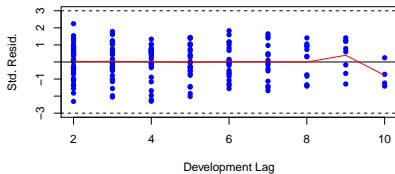
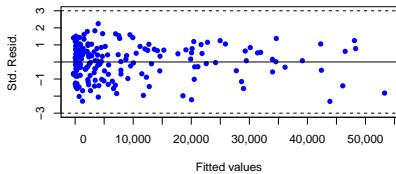
- individual - Ignore the other data
- pooled - All data is equal
- blended - Weighted average of first two

```
PaidCLPooled =  
  TriangleModel(UpperTriangle(triWC)  
    , Response = "IncrementalPaid"  
    , Predictor = "PriorCumulativePaid"  
    , ModelType = "pooled")
```

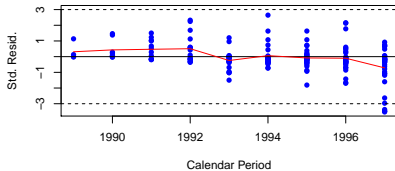
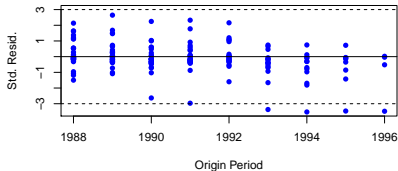
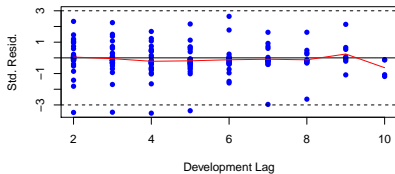
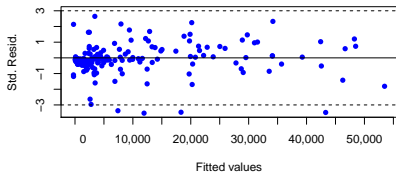

PlotResiduals(PaidCLPooled)



PlotResiduals(PaidCLIndividual)



PlotResiduals(PaidCLBlended)



```
RMSE(PaidCLPooled)
```

```
## [1] 3832
```

```
RMSE(PaidCLIndividual)
```

```
## [1] 2454
```

```
RMSE(PaidCLBlended)
```

```
## [1] 2482
```

```
projPaidCLPooled =  
  TriangleProjection(PaidCLPooled  
    , AsOfDate=as.Date("2006-12-31")  
    , MaxLag=10)
```

```
sqrt(sum((dfCompare$Actual - dfCompare$Pooled)^2))
```

```
## [1] 262455
```

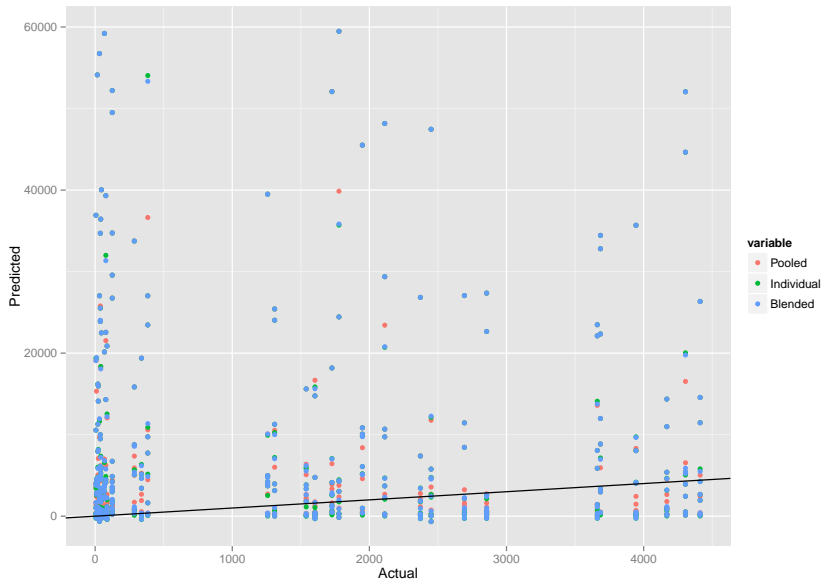
```
sqrt(sum((dfCompare$Actual - dfCompare$Individual)^2))
```

```
## [1] 266518
```

```
sqrt(sum((dfCompare$Actual - dfCompare$Blended)^2))
```

```
## [1] 266283
```

Model comparison



- Lots of fun with S4 classes
- melt + ggplot = fun
- Multi-level models -> spoiled for choice
- XLConnect makes everyone happy

- Project on GitHub:
<https://github.com/PirateGrunt/MRMR>
- Finalize for CRAN
 - Complete accessor and assignment functions
 - New generics methods for all objects
 - Vectorization of TriangleModel
 - Various of convenience functions
 - More visualizations!
 - Vignette
- Shiny application “shout”
- Full Bayesian model

Who was that guy?

I work for The Redwoods Group.

We're a niche provider of commercial insurance and we're dedicated to making the world a better place.

We're small, but want to make a big impact. R is an ideal tool for us.