CSAA Reserving Project

by

Cody Pulliam

UC Santa Barbara

Abstract

Prior research has shown that ARIMA modeling is the better modeling technique over general linear regression for predicting Pure Premium costs. Using Auto and Home insurance data from 2005Q1 to 2013Q1, ARIMA models were fitted to the individual coverages for each state. Using R programming functions, such as auto.arima, ARIMA models were identified to determine if they had a seasonal component or not. The underlying question was to identify if a general type of ARIMA can be classified by coverage and/or state. Multiple methods were used to approach this question of classification. The study concludes with Neural Networks representing the best model for classifying the seasonal versus nonseasonal ARIMAs by State and Coverage.

June 22, 2017

1 Introduction

Before this project, I participated in an Actuarial research class, PSTAT 296, that focused on the applications and effectiveness of Time Series analysis. A group and I concluded that ARIMA modeling best forecasted the loss trends of CSAAs Pure Premiums in their Auto and Home insurance. Following our research, the reserving group of CSAA asked that we consider finding any possible correlations to the type of ARIMAs that were fitted to the individual series by state and/or coverage. Example, Bodily Injury, an auto insurance coverage that is required in almost all states may experience a seasonal component if its earned exposure is above a certain limit. The same could be asked for each state. If such a classification exists, the application of this would not only be useful for forecasting future series that come in, but also for understanding what type of reserves they should expect for new states they wish to open coverages in.

2 Raw Data

The data set provided by CSAA consists of two types of Insurance, Home and Auto Insurance. Both sets of data summarize quarterly losses by state and coverage up to a maximum of eight years (2005 Q1 to 2013 Q1). There was 18 states worth of data provided by either ACA Insurance or Western Union Insurance Company (WU) for both Home and Auto Insurance. In the data sets each row corresponds to a quarter, auto and home had a total of 5941 and 2674 quarters respectively. Each quarter contained data for four quantifiable attributes: Frequency, Severity, Pure Premium, and Earned Exposure.

- Earned Exposure = Number of bookings for the quarter. For example, given quarterly data, a client with an insurance policy for the entire quarter is considered one full unit of earned exposure. A policyholder that either starts or cancels a certain coverage during that quarter is perceived to be a partial unit. In other words, a client that starts a policy half way through the quarter is accredited 1/2 a unit of earned exposure.
- Frequency = $\frac{\text{number of claims}}{\text{number of exposures}}$, the rate at which claims occur.
- Severity = $\frac{\text{total losses}}{\text{number of claims}}$, average cost of claims.

- Pure Premium $= \frac{\text{total losses}}{\text{number of exposures}} = \text{Severity} \times \text{Frequency}$, the average loss per exposure, also known as the insurers expected risk per policy holder.

Both insurance types summarize each of the above variables by quarter for each of the various coverages/forms. Figure 1 below shows the first 5 rows of the data set and what attributes we will be dealing with going forward. As a personal insert, I have added the Region and Number of Claims attributes (Regions classified as those represented in map found in the Appendix, Figure 8)

	Insurance	State	Coverage	Model	Frequency	Severity
1	Auto	AZ	BI	ARIMA(1,0,0)(1,0,0)[4] with non-zero mean	0.921873349	14862.7945900
2	Auto	AZ	COLL	ARIMA(0,1,0) with drift	4.519925602	2948.9757060
3	Auto	AZ	COMP	ARIMA(0,1,0)(2,0,0)[4] with drift	13.695262050	499.7178473
4	Auto	AZ	MP	ARIMA(1,0,0)(2,0,0)[4] with non-zero mean	0.947761721	3206.8059540
5	Auto	AZ	PD	ARIMA(0,1,1)(0,0,1)[4] with drift	2.956986595	3031.1889900
6	Auto	AZ	RENT	ARIMA(0,1,0)(0,0,2)[4] with drift	4.033777206	304.4539191

	Pure.Premium	Earned.Exposure	Ratio.F.S	Number.of.Claims	SorN	Region
1	13702	87837	0.00009	80975	Seasonal	Western
2	13329	73552	0.00178	332450	NonSeasnl	Western
3	6844	76563	0.03268	1048550	Seasonal	Western
4	3039	51957	0.00036	49243	Seasonal	Western
5	8963	87829	0.00111	259709	Seasonal	Western
6	1228	50541	0.01210	203871	Seasonal	Western

Figure 1: First 5 rows of the data

3 Modifications

3.1 Challenges

The original data sets for home and auto insurance imposed many challenges that would impede the ability to formulate accurate predictive models. These challenges include missing data, outliers, and negative values. For simplicity, we will define the notation of each state and coverage as State-Coverage using their abbreviations and refer to these as series. (i.e., series AZ-BI is Arizona-Bodily Injury). Examples of each of the following challenges are found below:

1. Negative Values: Data was discovered that had negative values which can be explained by subrogation (recovering damages from a third-party at fault) and salvage (reselling damaged property for a profit).

This data was considered extremely rare in occurrence and thus thrown out.

2. Insufficient Data: There were zeroes found in numerous series that would adversely affect our time series analysis. Although it was accu rate data, it may be best to discard this data and choose series with values for all quarters to achieve better trend analysis.

3. Outliers: Large fluctuations in the frequency and severity data presented problems for predicting models. A possible cause could be a natural disaster causing a spike in severity.

However, although the above challenges existed, the underlying question is to understand the overall classification of their identified ARIMA models. I came to this conclusion because I am only concerned with if the ARIMA returns a seasonal component or not. The only concern should be if the series were missing its last two observations, meaning the data was not accessible at the time it was extracted or they dont carry that coverage/form anymore. Only the series which did not have any data for quarters 2012-Q4 and 2013-Q1 were excluded.

4 Methods

4.1 Decision Tree Based Methods

This project is centered around classification, in which we are trying to identify when a series returns a seasonal ARIMA model, can it be classified into groups by State and/or Coverage. From what we have learned in class and within the assigned textbook, *Introduction to Data Mining* by Tan, Steinbach and Kumar, there are a variety of methods to use for classifying data. Below is the decision tree based methods I will explore for this data set.

- **CART** (Classification and Regression Trees)- A method in which rules are created and form a decision tree. These rules are created through computing the Gini impurity for a set of items/values and classifying them into sets based on how much information is gained through the set of predictor variables given a response variable.

- Neural Networks- Where a network is an interconnected group of nodes. Often used for pattern classification and pattern recognition.

- Naive Bayes - A naive Bayes classifier is a simple probabilistic classifier based on applying Bayes theorem with strong independence assumptions. Computes conditional probabilities of different given information (predictor variables) to form its decision rules.

There are far more types of methods out there in the Data Mining world, but I have narrowed it down to the ones above due to the nature of the data. The variables described earlier are numerical values observed on a quarterly basis. It would be meaningless to apply all the quarters of a particular series and its respective ARIMA model because all of the quarters were used in the model result. For example, since we are trying to classify a series as having a seasonal component or not, that binary column would say its ARIMA did return a seasonal component for all its quarterly rows from 2005 to 2013. This would be meaningless due to the scope of this project and try to classify this by state and/or coverage. As a result, only the last quarter, 2013Q1, will be kept for each series. This process then disqualifies many Data Mining tools that rely mostly on numerical attributes. Since ours is a combination of categorical and numerical, with a bias towards categorical, the above methods were chosen because they have been known to work well with mixed attribute types of data.

4.2 Training and Test Sets

The data will be divided into two randomly selected sets labeled as the Training and Test Sets. The Training set, which will consist of 80 percent of the data will be used for determining a model for each of the methods in Section 4.1. The models for each of the methods will be chosen through their respective cross-validation performance. The model where its cross- validation error is lowest (or its Accuracy is highest) will be the selected model to represent that particular method. Next, predictions will be generated using only the Test set (the other 20 percent of the data) to test the validity of the model. Confusion matrices will be generated that show the misclassification error. The model with the lowest misclassification error will be the method that best works for the data. Further analysis such as its Cost Matrix and Model Accuracy will determine if the model should hold as a representable model.

5 Results

5.1 Correlation Matrix

Before jumping into the model selection, Id first like to see how the variables interact and if there are any noticeable correlations between the variables. To start, with only the numerical attributes, a useful computation method known as the correlation matrix will help identify any attributes that may have a strong correlation to one another.

	Frequency	Severity	Earned.Exposure	Ratio.F.S	Number.of.Claims
Frequency	1.000	-0.276	-0.087	0.465	0.396
Severity	-0.276	1.000	0.058	-0.251	-0.143
Earned.Exposure	-0.087	0.058	1.000	-0.002	0.613
Ratio.F.S	0.465	-0.251	-0.002	1.000	0.359
Number.of.Claims	0.396	-0.143	0.613	0.359	1.000

Figure 2: Correlation Table of Numerical Attributes

Right away you will notice that the diagonal green cells all have a correlation of 1.0, which is what it should be since they are correlations of themselves. However, apart from that diagonal, highlighted in dark orange, shows a very strong correlation between the Earned Exposure and Number of Claims attribute. This makes sense because the Number of Claims attribute is made up of the Frequency multiplied by the Earned Exposure. Its always good to get rid of repetitive predictive variables, so I chose to exclude the Number of Claims attribute going forward.

5.2 Data Visualization

Another useful technique before getting into modeling is to look for any possible clustering or obvious separation in the data. Due to the mixture of my data being both categorical (State, Coverage, Seasonal/NonSeasnl) and numerical, Ive created multiple plots with different variations to their axis, which can be found in the Appendix. Of all the plots I found no visual separation or clustering. I did find however, that the observations that departed greatly from the rest were generally classified as Non-Seasonal.

5.3 Model Selection

Using the R package caret, I am able to use thetrain function to return the best model for each of the three methods (rpart, Naive Bayes, and Neural Networks).

Beginning with rpart, which uses the methods found in **CART**, the model can be controlled through what is known as a cost complexity factor.

This is an optional parameter, but essentially what does is allows splitting only when the overall lack of fit is decreased by that cost complexity factor. Its a very important input and can dramatically change the result of the tree model. That is why the cost complexity factor will be chosen by whichever returns the highest accuracy (Cross-Validation).

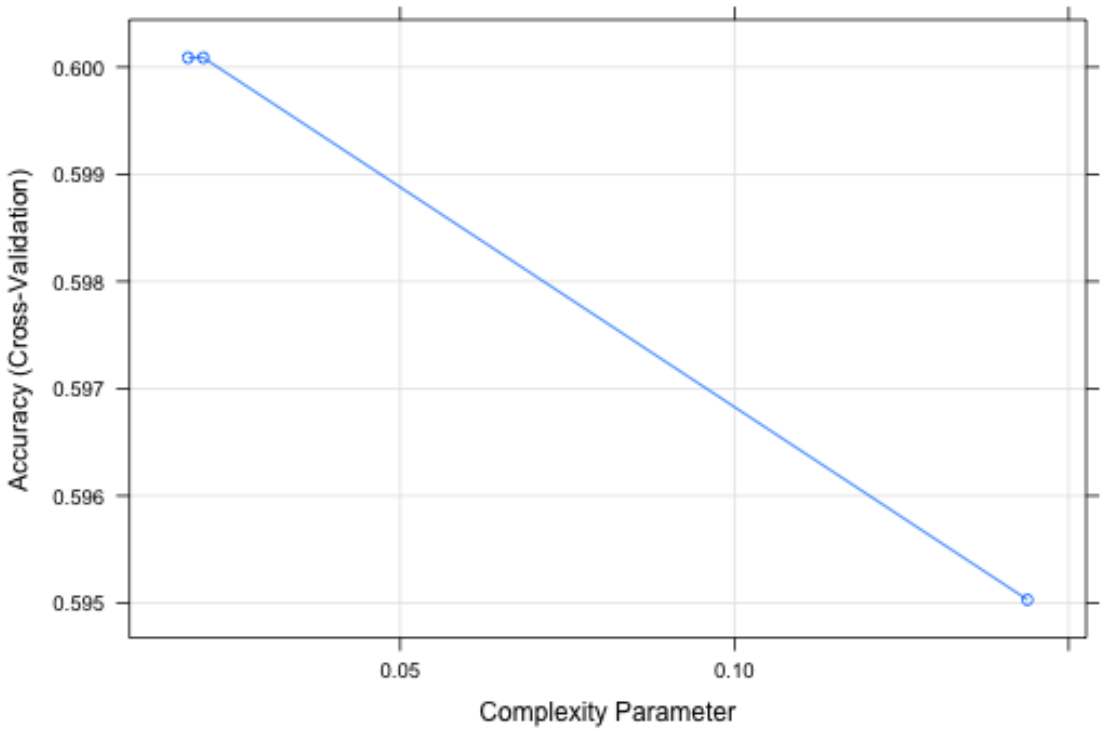
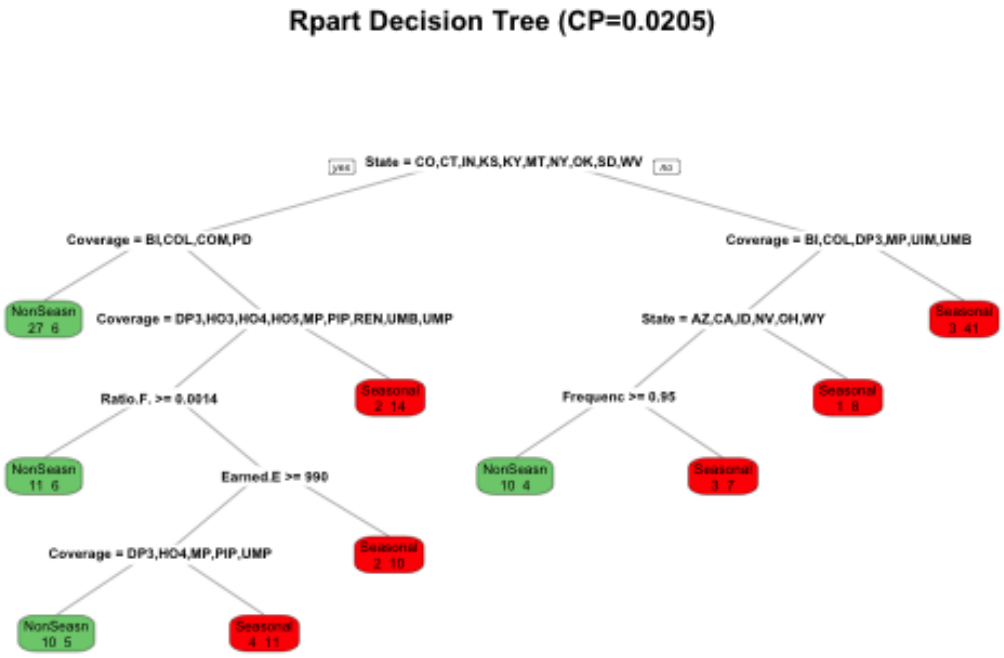


Figure 3: Rpart Cross-Validation

The Complexity Parameter that gives the highest accuracy is at 0.0205 just before the line darts down. From this, we receive the following decision tree.

The confusion matrix below gives us a general idea of how well the model did on classifying the training set.

Confusion Matrix	NonSeasnl	Seasonal
NonSeasonl	58	15
Seasonal	21	91



The next model, also decision based, but does not output a decision tree, is Naive Bayes. In Figure 4, you will notice that its model is controlled by either being Gaussian or Nonparametric. From the graph, we are able to say that Gaussian returns a more accurate model and thus should be used as the primary method when the model is created.

The last method, Neural Networks, was a bit harder to run but returned some very interesting results. Instead of using all possible predictor variables to generate the model, for Neural Networks I tried just using the State and Coverage attributes because that was what we initially are trying to use to classify which series are Seasonal versus NonSeasnl.

Although hard to interpret, Figure 5, the inputs are on the left and consist of a state and/or coverage that are assigned to a number. The links we see connected to the dots are the networks found within the training set and lead us to the output of our Seasonal or NonSeasnl attribute.

Neural Networks have what are called hidden units, the hidden layers

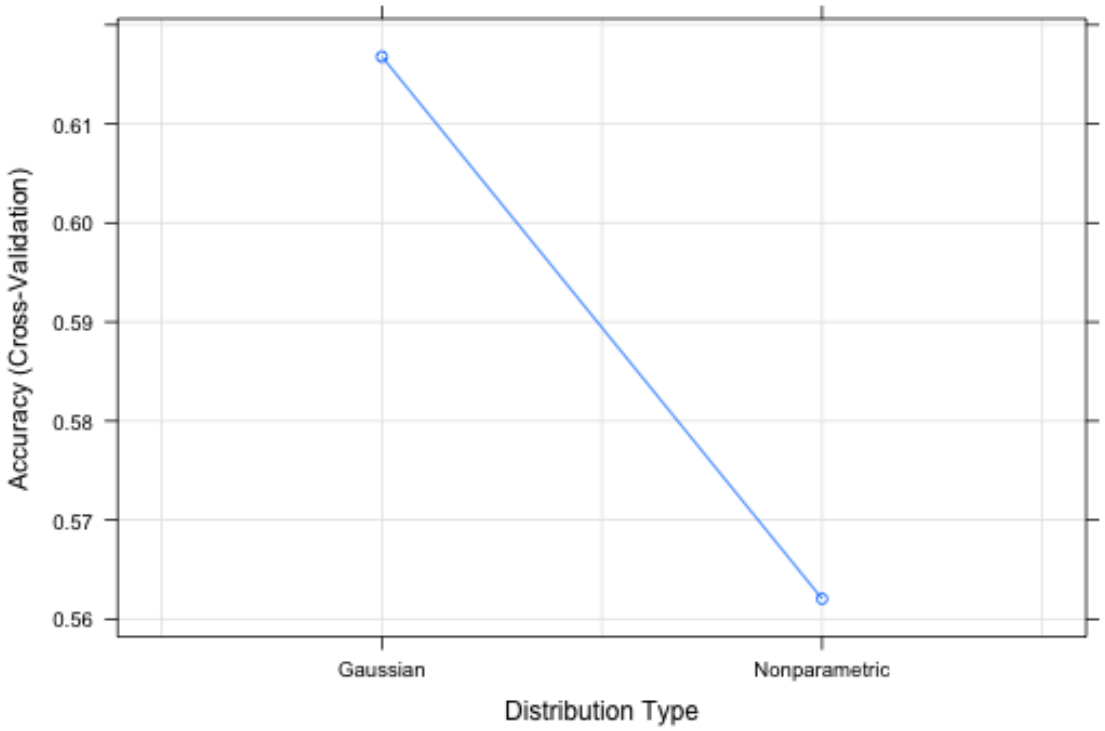


Figure 4: Naive Bayes Cross-Validation Plot

job is to transform the inputs into something that the output layer can use. The number of hidden units can be chosen through cross-validation like we have done for the other two methods.

5.4 Model Selection

Now that we have identified the models for each of the methods, we will now compare their classification error to determine which is best.

Confusion Matrix	NonSeasnl	Seasonal
NonSeasonl	7	6
Seasonal	15	19

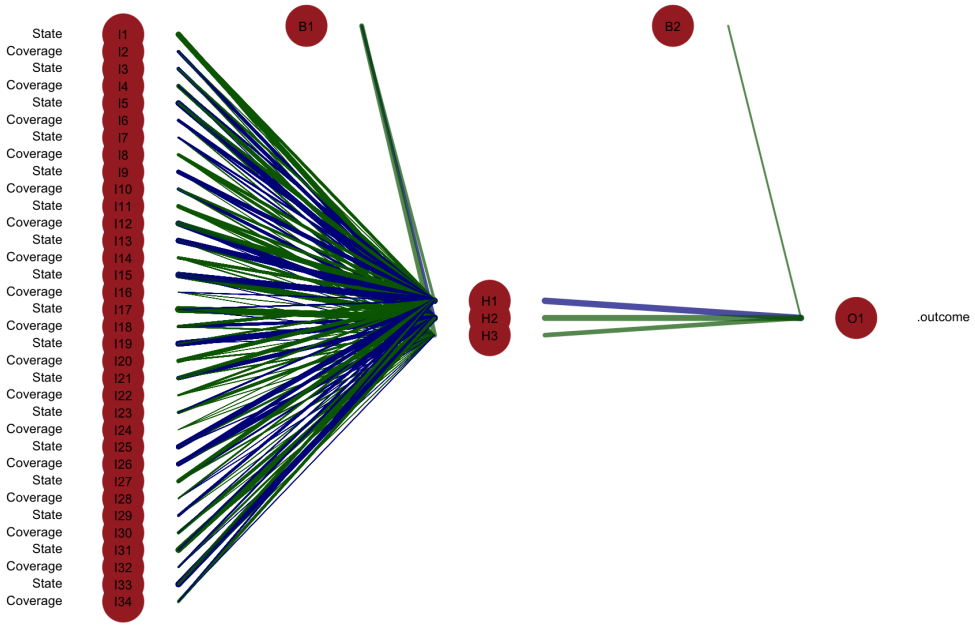


Figure 5: Neural Networks Modelled Diagram

Confusion Matrix	NonSeasnl	Seasonal
NonSeasonl	4	6
Seasonal	18	19

Confusion Matrix	NonSeasnl	Seasonal
NonSeasonl	15	10
Seasonal	7	15

Another means of comparing these classification errors is by the below formula. In my case, the Neural Networks model undoubtedly won with 64%, while the others didn't even make it over 20%.

$$\text{Classification Error} = \frac{\text{Number of Correct Classifications}}{\text{Total Number in Test Set}}$$

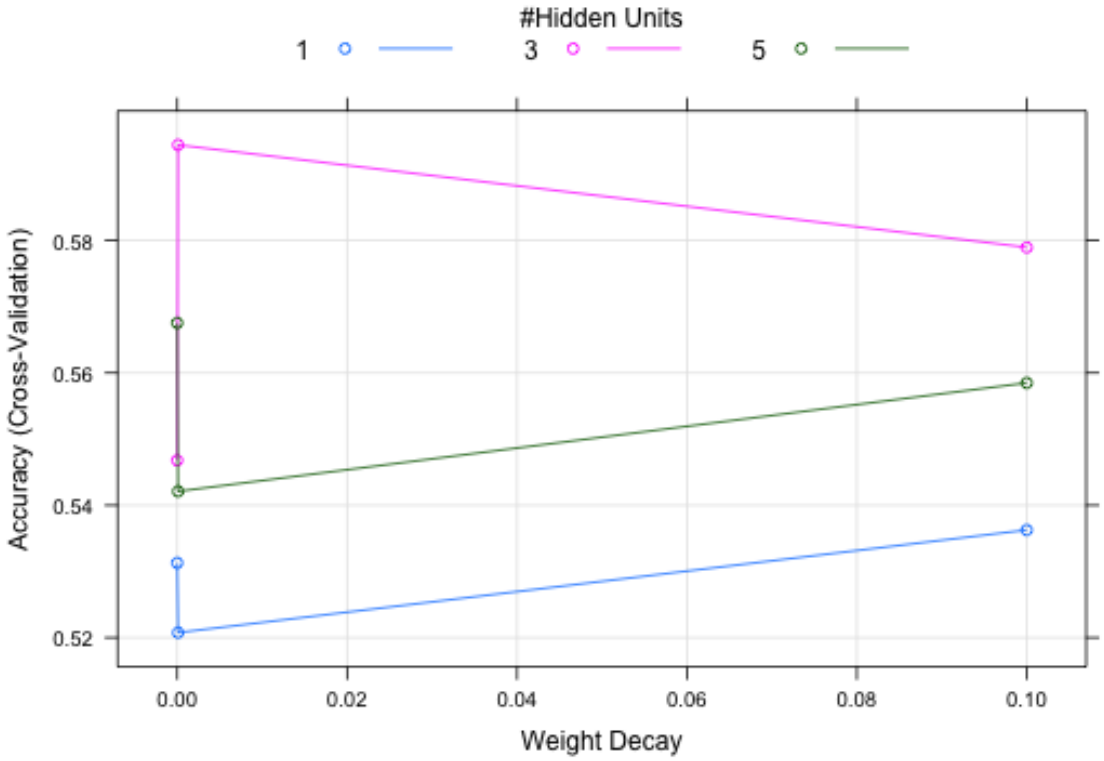


Figure 6: Neural Networks Cross-Validation

6 Conclusion

We found that Neural Networks sufficed as the better method through classification error and chose the best model by maximizing accuracy by using Cross-Validation. Although hard to interpret from the Neural Network graph, it is safe to say that there is a way of classifying seasonal versus non-seasonal series by State and Coverage. Although not shown in the above results, multiple variations of which predictor variables to include had been experimented with the Rpart and Naive Bayes methods. Only to find that their classification error dramatically lessened. Which makes it even more impressive that the Neural Networks only requires two categorical attributes to outperform the other two methods.

It can be debatable that the method of classifying a series as seasonal

ornot is completely biased to what the ARIMA model believes it to be. Some may argue that a series classified as seasonal is not by a simple personal interpretation of the graph. However, this is where I remind you that the original scope of the project was to find the effectiveness and applicability of time series. In which we concluded that the ARIMA models worked best for predicting the companys future Pure Premiums. The underlying question that motivates this data mining project, was a request by the company to further find any similarities of the ARIMA models among the series. In which the response to this question is that there is a classification of seasonal versus nonseasonal ARIMA models and can be found through the methodology of Neural Networks.

References

References

[1] P.-N. Tan, M. Steinbach, V. Kumar. Introduction to Data Mining, Addison-Wesley, 2005.

[2] Brockwell, P. J. and R. A. Davis. Introduction to Time Series and Forecasting. 2nd ed. New York, NY: Springer, 2002.

[3] Joshua Ulrich (2013). TTR: Technical Trading Rules. R package version 0.22-0. http://CRAN.Rproject. org/package=TTR

[4] R Core Team (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.

7 Appendix

7.1 Rcode

```
setwd("~/Documents")
#######Prediction Clusters###############
mydata <- read.delim("ErrorTable1.txt", header=TRUE)
names(mydata)
mydata <- mydata[,1:(dim(mydata)[2]-4)]

#########
```

```
responseY <- as.matrix(mydata[,11])
predictorX <- as.matrix(mydata[,)])


##########Correlation Matrix###############
d <- data.frame(mydata$Frequency,mydata$Severity, mydata$Earned.Exposure, mydata$Rati
dv <- cor(d) # get correlations ##Strong Correlations between Number of Claims and Ea

write.table(dv, "CorrelationMatrix.txt", sep="\t")  ###export a text file which will


# Create training and test sets
set.seed(1)
N <- nrow(mydata)
index.train <- sample(1:N, size=floor(N*.8),replace=FALSE)  ###Create a random Traini
index.test <- setdiff(1:N,index.train)  ###Create random Test Set

mydata.train <- mydata[index.train,]
mydata.test <- mydata[index.test,]

# Model 1 - Decision Tree using Gini
# Create a decision tree using gini
tree.rpart <- rpart(SorN ~ Frequency + Severity + Earned.Exposure + Ratio.F.S + State
summary(tree.rpart) ###show summary and variables that were used

cv.gini <- cv.tree(tree.gini, FUN = prune.misclass, K=5)

prune.gini <- prune.misclass(tree.gini, # Original tree
                             best=9)
tree.gini <- prune.gini

pdf(file="q2g.pdf",width=12,height=6)
plot(cv.gini$size ,cv.gini$dev ,type="b",
     main = "Cross-Validated Error",
     xlab = "No. of nodes",
     ylab = "Misclassification",
     col = " blue")
dev.off()

pdf(file="tree2.pdf",width=15,height=9)   ####Print the decision tree
```

```
plot(tree.gini)
text(tree.gini ,pretty = 0, cex = 1, col = "red")
title("Classification Tree")
dev.off()

# Predict on test set
tree.gini.pred <- predict(tree.gini,mydata.test,type="class")

#Confusion Matrix
gini.confusion <- table(mydata.test$SorN,tree.gini.pred)
gini.confusion

# Predict on training set

tree.gini.pred.train <- predict(tree.gini,mydata.train,type="class")

#Confusion Matrix
gini.confusion.train <- table(mydata.train$SorN,tree.gini.pred.train)
gini.confusion.train


printcp(tree.gini) # print cost-complexity parameter

####Pretty decision Tree####
prp(tree.gini, extra=1, box.col=c("palegreen3", "red")[tree.gini$frame$yval],
    main = "Rpart Decision Tree (CP=0.0205)")
par(op)


#########Create Plots that show the summary visuals of the Data########
g1 <- qplot(Frequency, Coverage, data = mydata, colour = SorN,
            geom = c("jitter"))
g2 <- qplot(Severity, Coverage, data = mydata, colour = SorN,
            geom = c("jitter"))
g3 <- qplot(Frequency, State, data = mydata, colour = SorN,
            geom = c("jitter"))
g4 <- qplot(Severity, State, data = mydata, colour = SorN,
            geom = c("jitter"))
g5 <- qplot(Pure.Premium, State, data = mydata, colour = SorN,
            geom = c("jitter"))
```

```
g6 <- qplot(Pure.Premium, Coverage, data = mydata, colour = SorN,
            geom = c("jitter"))
g7 <- qplot(Earned.Exposure, State, data = mydata, colour = SorN,
            geom = c("jitter"))
g8 <- qplot(Earned.Exposure, Coverage, data = mydata, colour = SorN,
            geom = c("jitter"))
g9 <- qplot(Frequency, Severity, data = mydata, colour = Region,
            geom = c("jitter"))
g10 <- qplot(State, Coverage, data = mydata, colour = SorN,
            geom = c("jitter"))

qplot(Frequency, Severity,data=mydata)
pdf(file="gplots.pdf", height=6, width=5)
# Plot graphs in the same plot with grid.arrange()
grid.arrange(g1,g2,ncol = 1, nrow = 2)
             dev.off()
pdf(file="gplots2.pdf", height=6, width=5)
# Plot graphs in the same plot with grid.arrange()
grid.arrange(g3,g4,ncol = 1, nrow = 2)
dev.off()
pdf(file="gplots3.pdf", height=6, width=5)
# Plot graphs in the same plot with grid.arrange()
grid.arrange(g5,g6,ncol = 1, nrow = 2)
dev.off()
pdf(file="gplots4.pdf", height=11, width=5)
# Plot graphs in the same plot with grid.arrange()
grid.arrange(g7,g8,ncol = 1, nrow = 4)
dev.off()



#####Naive Bayes###############
names(mydata)
x.train = mydata.train[-c(1,4,5,6,7,8,9,10,11,12)]  ####Only STATE and Coverage Attri
x.train = mydata.train[-c(1,4,10,11)] ####all attributes but the Arima Model, type of

y.train = mydata.train$SorN
x.test = mydata.test[-c(1,4,5,6,7,8,9,10,11,12)]
x.test = mydata.test[-c(1,4,10,11)]
```

```
y.test = mydata.test$SorN
modelnb = train(x.train,y.train,'nb',trControl=trainControl(method='cv',number=10))
table(predict(modelnb$finalModel,x.train)$class,y.train)
table(predict(modelnb$finalModel,x.test)$class,y.test)

plot(modelnb)


##########Logistic Model#################
modelLMT = train(x.train,y.train,'LMT',trControl=trainControl(method='cv',number=10))
table(predict(modelLMT,x.train),y.train)
plot(modelLMT, plotType = "scatter")
plot(modelLMT$finalModel)


############Neural Networks Model################
modelNN2 = train(x.train,y.train,'nnet',trControl=trainControl(method='cv',number=10)
table(predict(modelNN2,x.test),y.test)  #####confusion matrix on test
table(predict(modelNN2,x.train),y.train) #####confusion matrix on training

###Create neural nework plot##
library(devtools)
source_url('https://gist.github.com/fawda123/7471137/raw/c720af2cea5f312717f020a09946
plot.nnet(modelNN2,nid=T)
plot.nnet(modelNN2,pos.col='darkgreen',neg.col='darkblue',alpha.val=0.7,rel.rsc=15,
          circle.cex=10,cex=1.4,
          circle.col='brown')
table(predict(modelNN2,x.test)$class,y.test)  ###Confustion Matrix
table(predict(modelNN2,x.train),y.train) ###Confustion Matrix

#############Rpart (CRAN) Models#################
modelRPart = train(x.train,y.train,'rpart',trControl=trainControl(method='cv',number=
table(predict(modelRPart,x.test),y.test)
table(predict(modelRPart,x.train),y.train)



summary(modelRPart)
```

```
plot(modelRPart)
```

7.2 Additional Plots and Figures

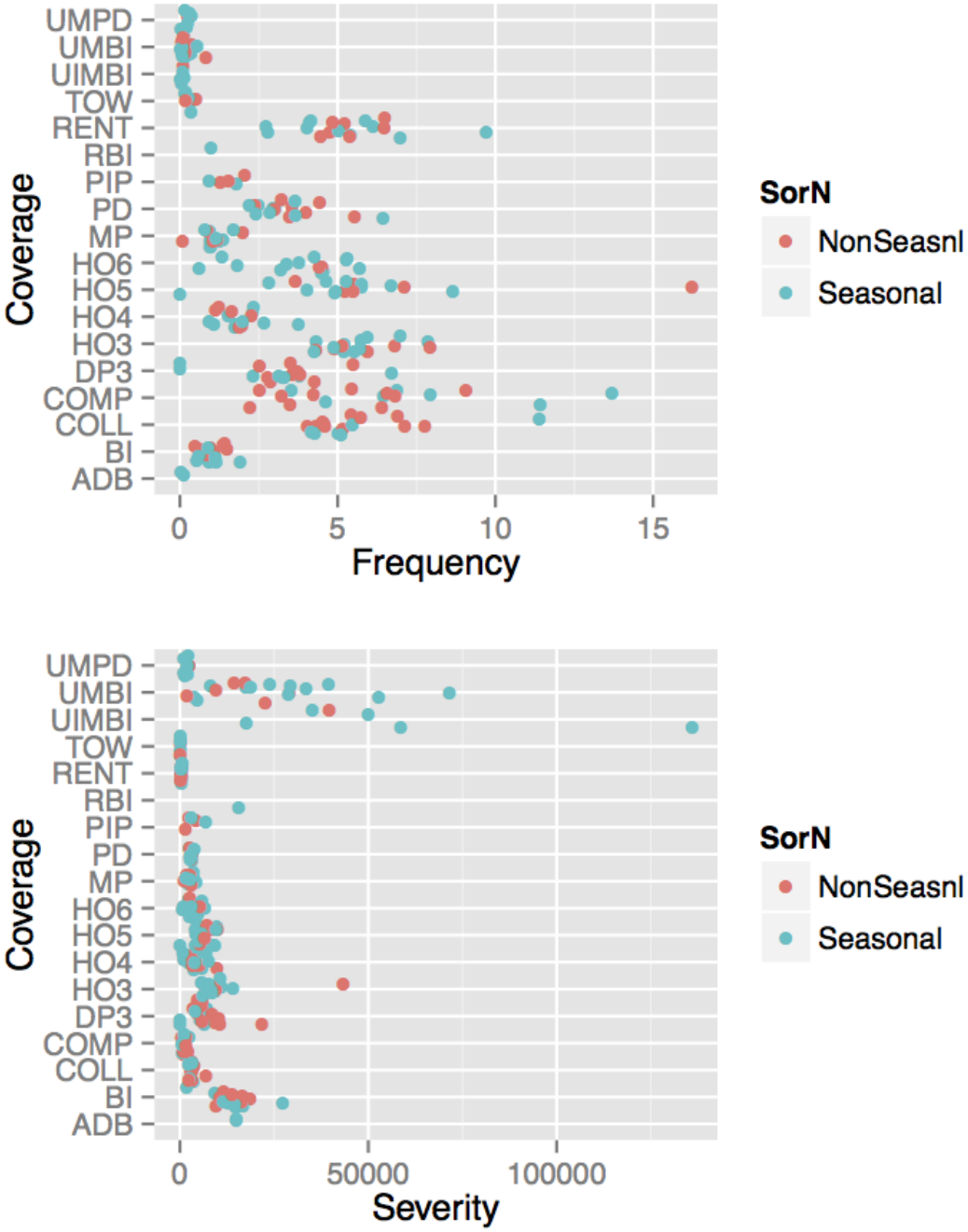
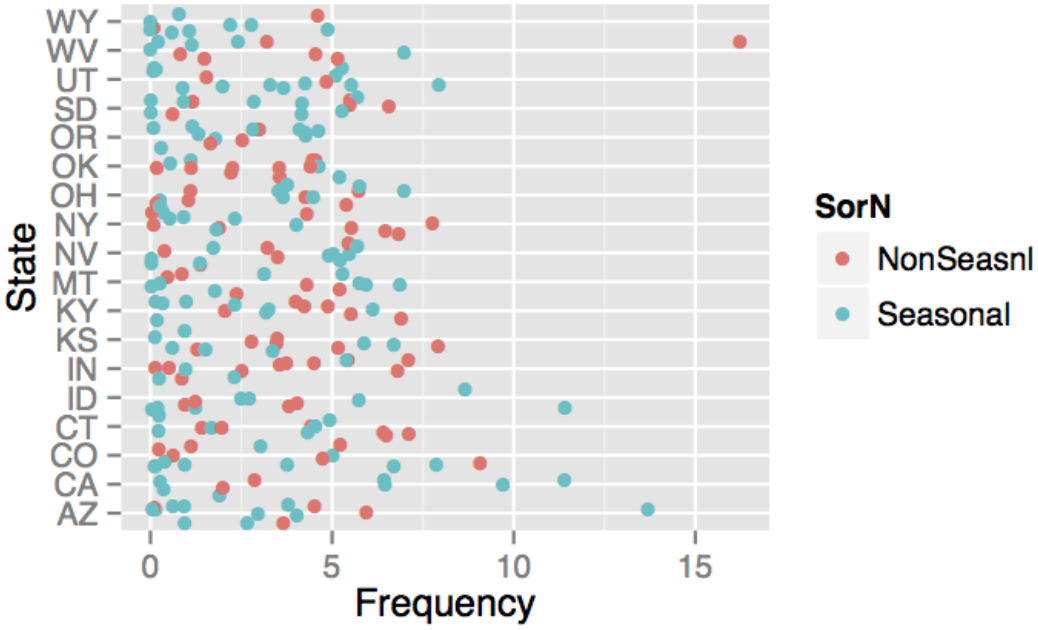
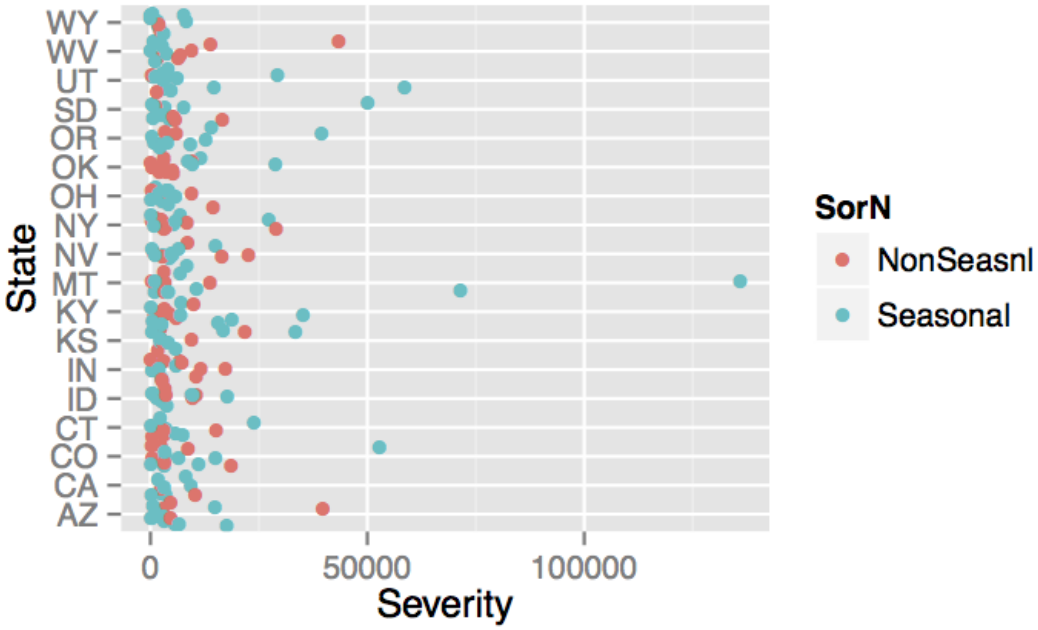
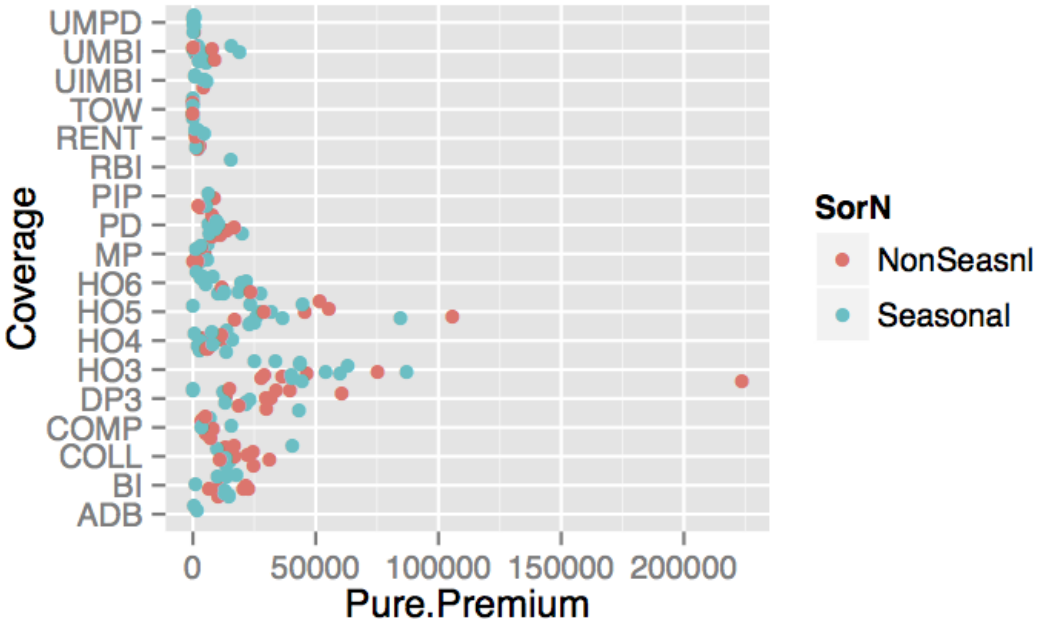


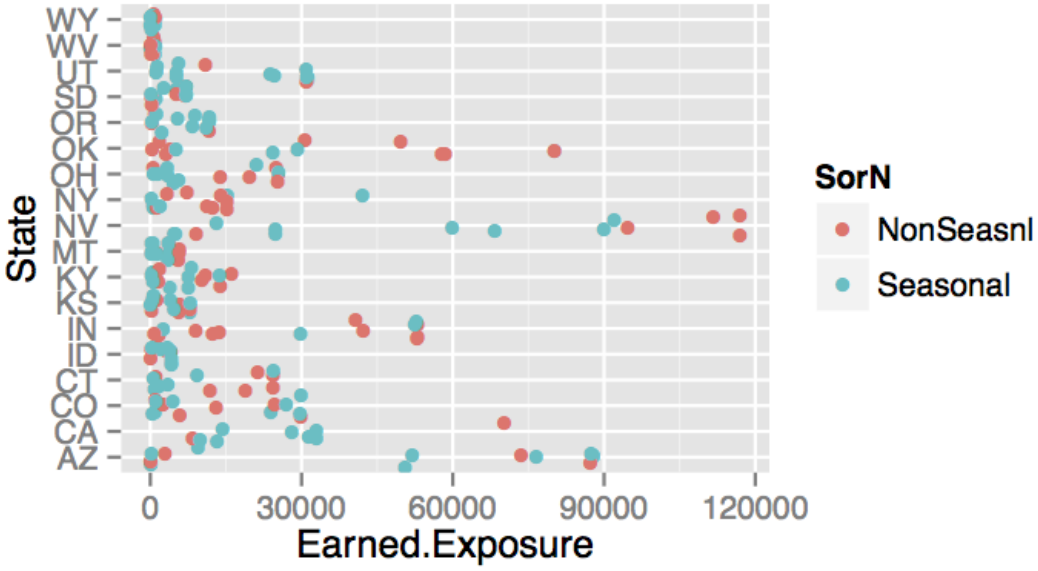
Figure 7: Visualizing the Data, More in Appendix


WY
WV
UT
SD
OR
OK
OH
NY
NV
MT
KY
KS
IN
ID
CT
CO
CA
AZ
State
0
5
10
15
Frequency
SorN
NonSeasnl
Seasonal


WY
WV
UT
SD
OR
OK
OH
NY
NV
MT
KY
KS
IN
ID
CT
CO
CA
AZ
State
0
50000
100000
Severity
SorN
NonSeasnl
Seasonal


State
WY
WV
UT
SD
OR
OK
OH
NY
NV
MT
KY
KS
IN
ID
CT
CO
CA
AZ
0
50000
100000
150000
200000
Pure.Premium
SorN
NonSeasnl
Seasonal


Coverage
UMPD
UMBI
UIMBI
TOW
RENT
RBI
PIP
PD
MP
HO6
HO5
HO4
HO3
DP3
COMP
COLL
BI
ADB
0
50000
100000
150000
200000
Pure.Premium
SorN
NonSeasnl
Seasonal


WY
WV
UT
SD
OR
OK
OH
NY
NV
MT
KY
KS
IN
ID
CT
CO
CA
AZ
State
0
30000
60000
90000
120000
Earned.Exposure
SorN
NonSeasnl
Seasonal


UMPD
UMBI
UIMBI
TOW
RENT
RBI
PIP
PD
MP
HO6
HO5
HO4
HO3
DP3
COMP
COLL
BI
ADB
Coverage
0
30000
60000
90000
120000
Earned.Exposure
SorN
NonSeasnl
Seasonal

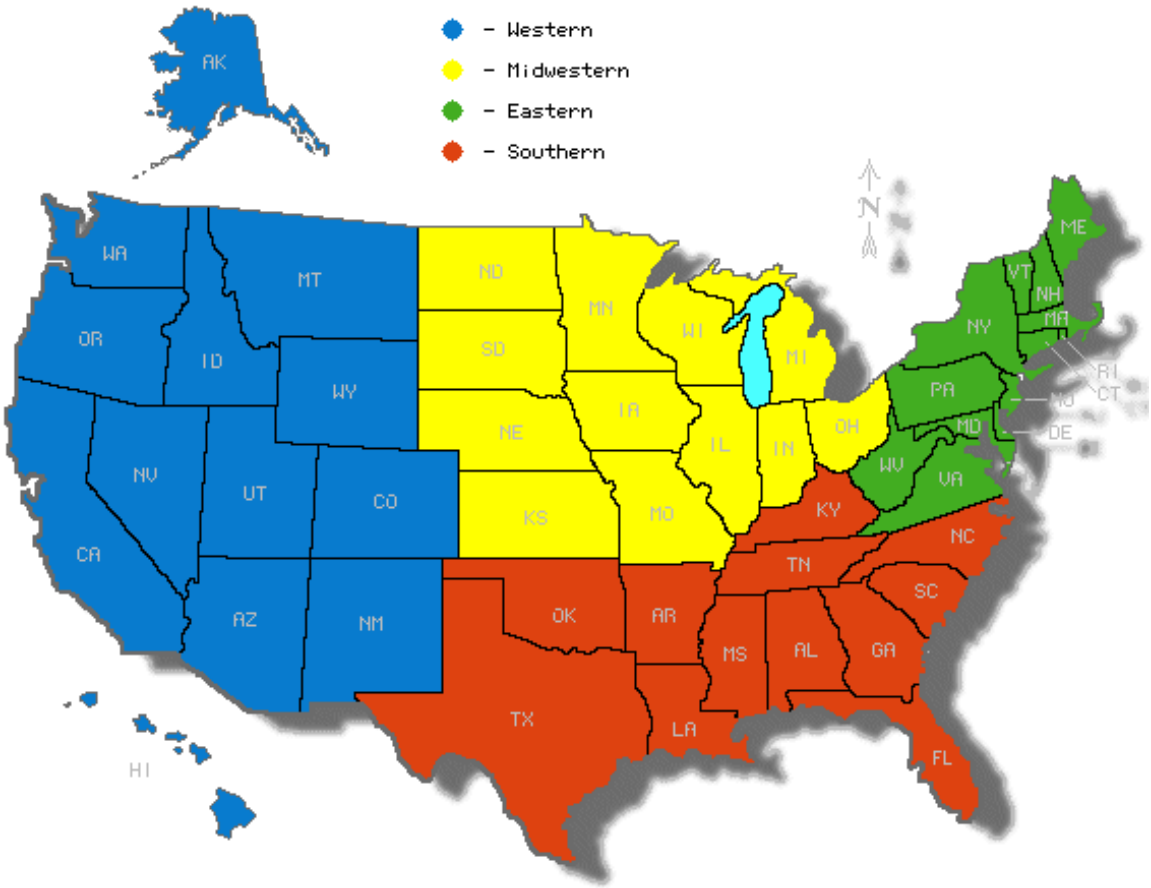


Figure 8: United States Regions Map

Region	# of States	# of Series
Western	9	114
Midwestern	4	54
Eastern	3	37
Southern	2	27

Figure 9: Regions Chart