



Auto Loss Cost Trends: Q4 2017

August 2018



Q4 2017 Trend Analysis

Contents

ARIMA Models	1
Bodily Injury	2
Property Damage	9
Comprehensive	16
Collision	23
Personal Injury Protection	30
Code	33

ARIMA Models

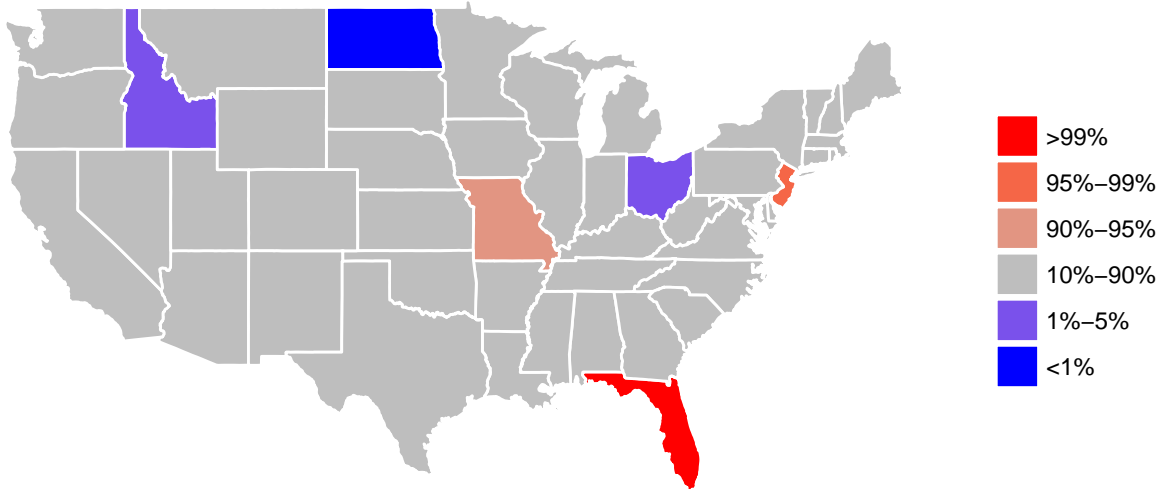
Using FAST TRACK PLUS data, we analyzed how auto insurance claims change over time. In this report, we identify which state-level claims are unexpectedly extreme. Using an ARIMA model, we forecast the 4th quarter 2017 value for each state, metric (frequency, severity and loss cost), and coverage (bodily injury, property damage, comprehensive, collision, and personal injury protection). ARIMA models allow us to account for both the quarter-to-quarter trends and any trends across years (for example if all the 3Q values are higher than any other quarters in the year). The state maps show the accuracy of the 4th quarter prediction. Blue states indicate that the true 4Q metric was significantly less than expected, and red states indicate that the true 4Q metric was significantly greater than expected.

Below each state map, full time series plots are displayed for the states that had a 4th quarter metric fall outside of the 10%-90% prediction interval. The green dot represents the true 4th quarter value. The blue dot represents the point forecast with a surrounding grey area representing the 10%-90% prediction interval.

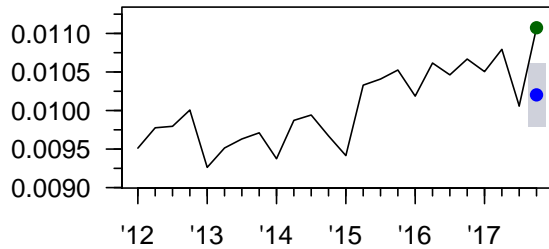
Some states and metrics experienced an extreme or abnormal observation due to weather or other events; these outliers decreased the accuracy of the ARIMA model. The time series with outliers were smoothed as to improve the 4th quarter prediction accuracy. These smoothed outliers are marked with an orange square on the time series plots.

Bodily Injury

Bodily Injury Frequency



Florida Bodily Injury Frequency



Idaho Bodily Injury Frequency



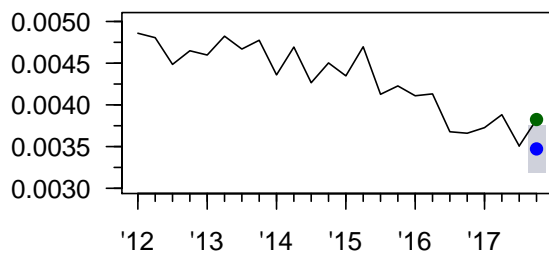
Missouri Bodily Injury Frequency



North Dakota Bodily Injury Frequency



New Jersey Bodily Injury Frequency



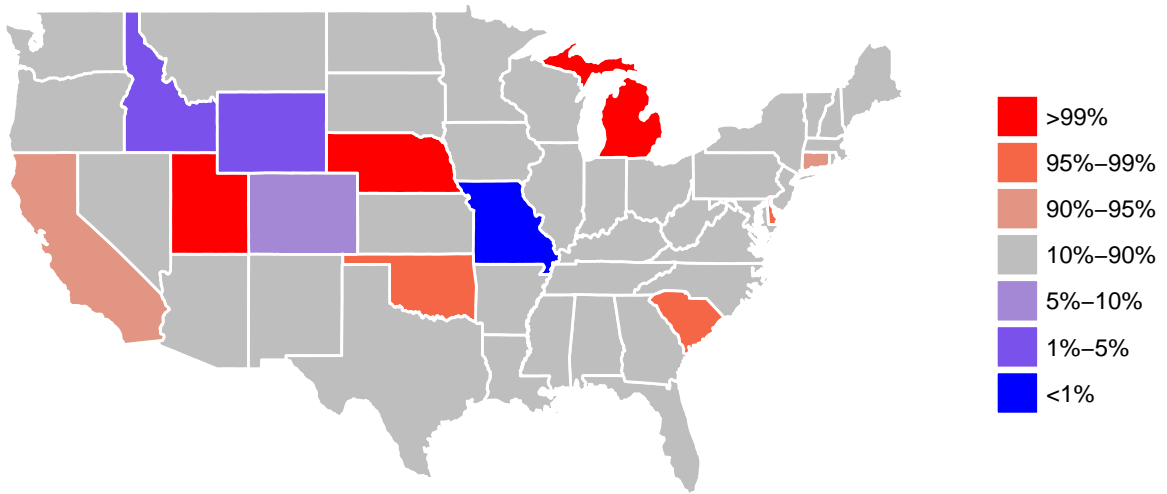
Ohio Bodily Injury Frequency



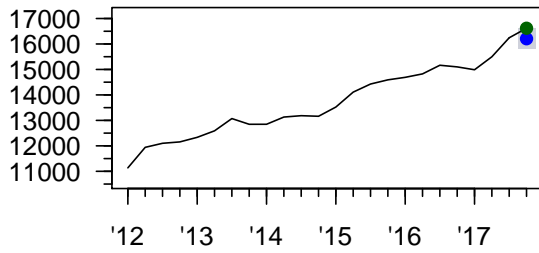
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

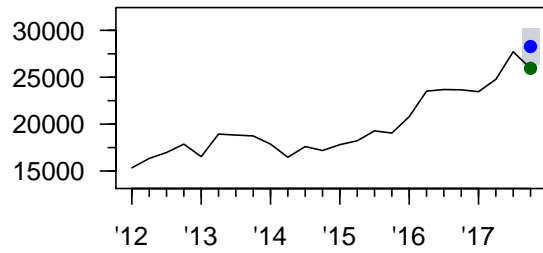
Bodily Injury Severity



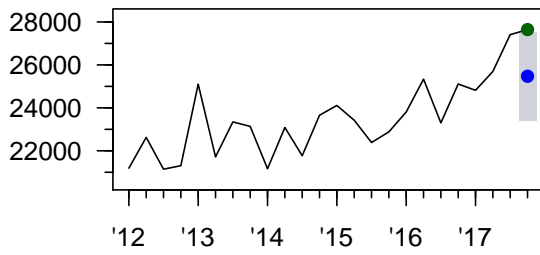
California Bodily Injury Severity



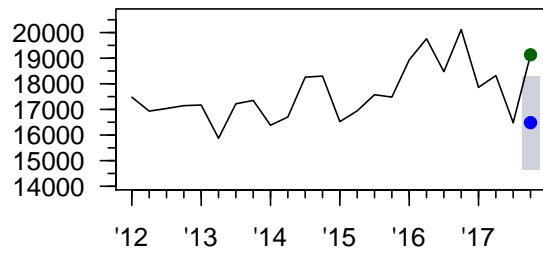
Colorado Bodily Injury Severity



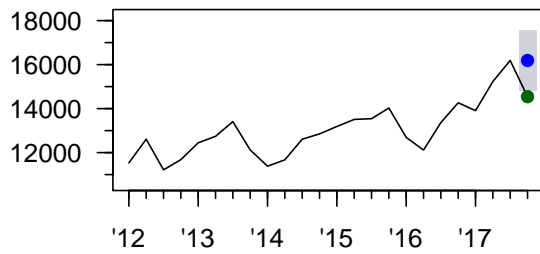
Connecticut Bodily Injury Severity



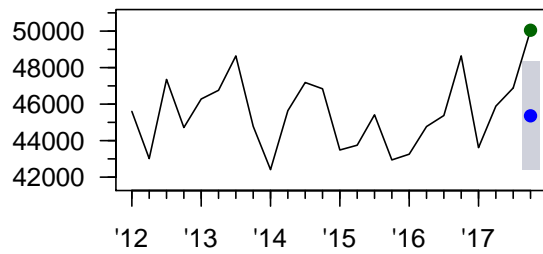
Delaware Bodily Injury Severity



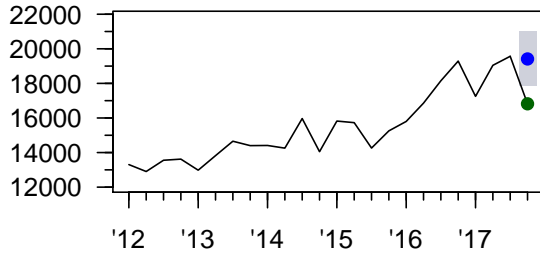
Idaho Bodily Injury Severity



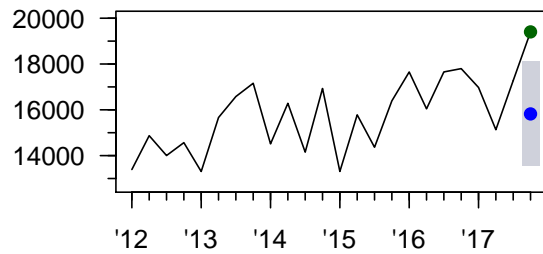
Michigan Bodily Injury Severity



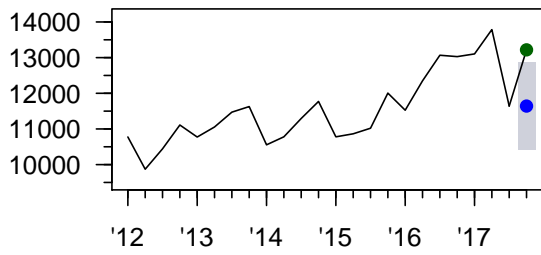
Missouri Bodily Injury Severity



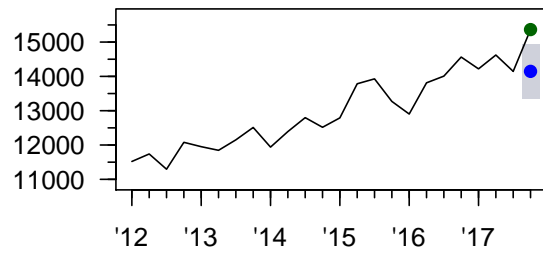
Nebraska Bodily Injury Severity



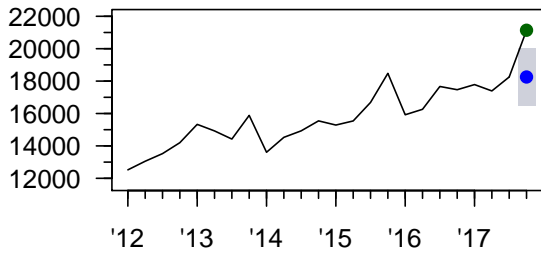
Oklahoma Bodily Injury Severity



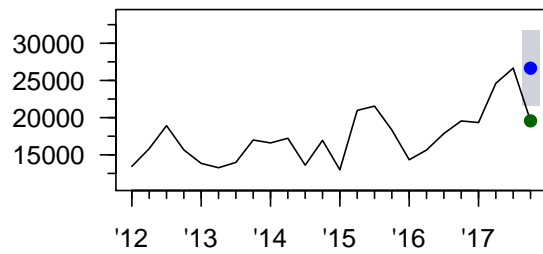
South Carolina Bodily Injury Severity



Utah Bodily Injury Severity



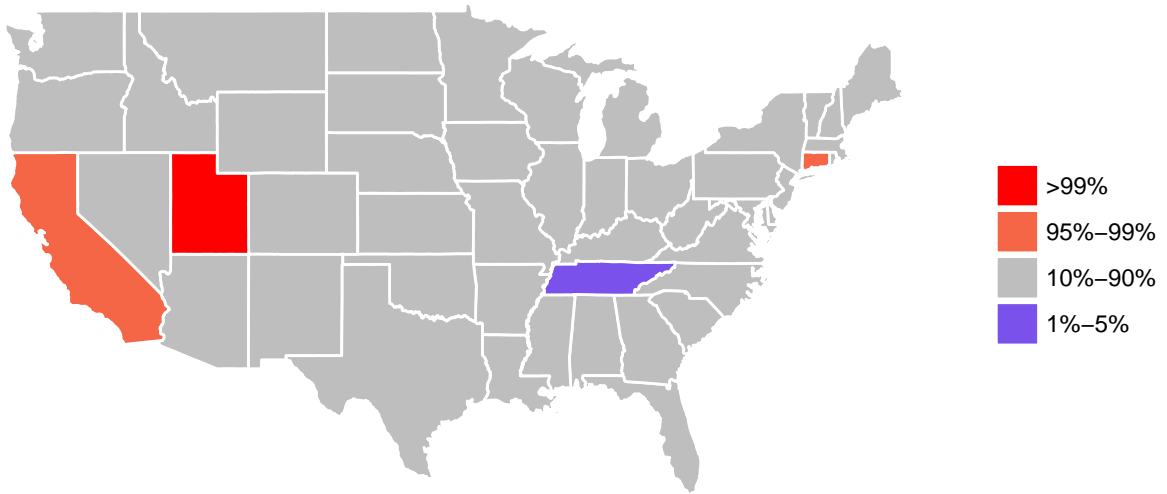
Wyoming Bodily Injury Severity



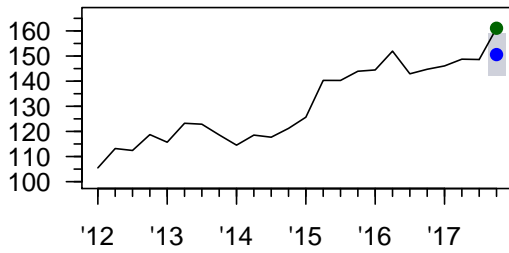
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

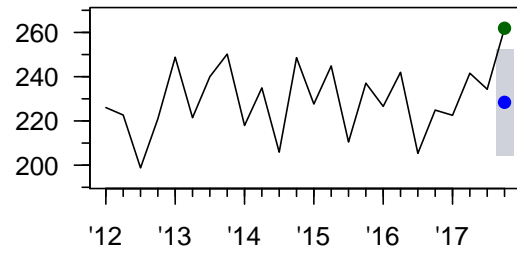
Bodily Injury Loss Cost



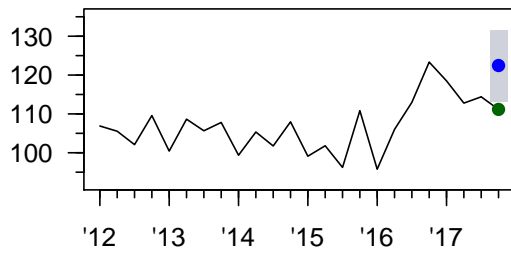
California Bodily Injury Loss Cost



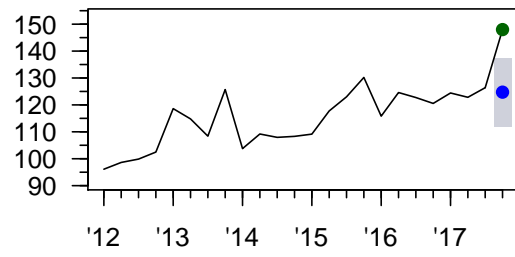
Connecticut Bodily Injury Loss Cost



**Tennessee Bodily Injury
Loss Cost**



**Utah Bodily Injury
Loss Cost**

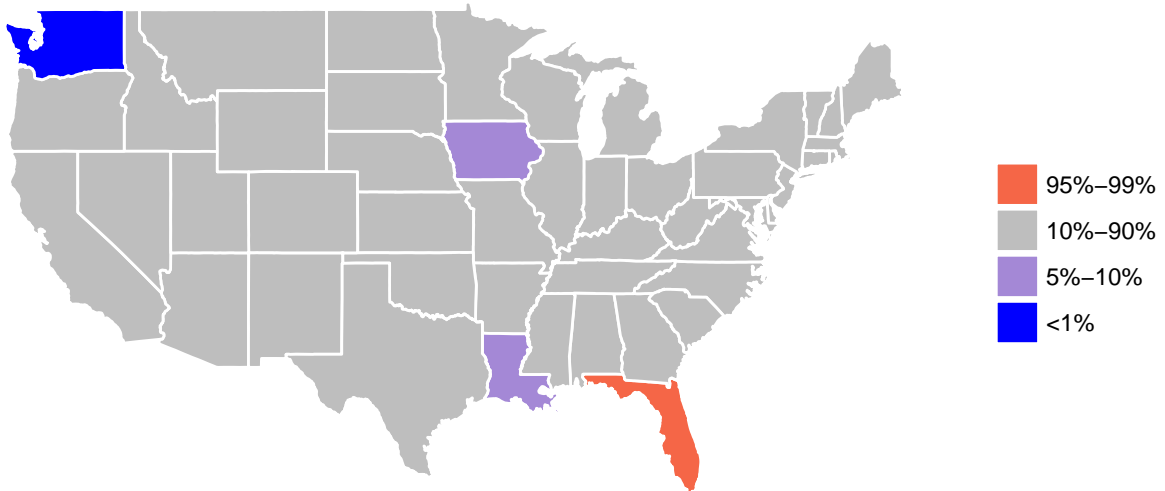


Time Series Legend

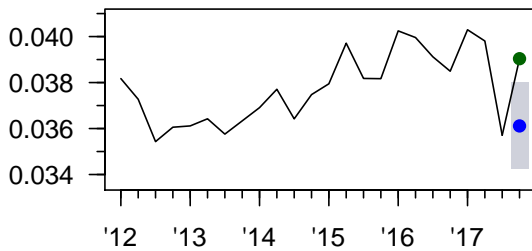
- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

Property Damage

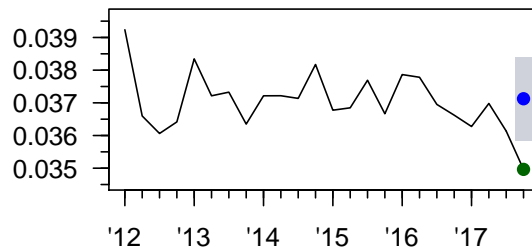
Property Damage Frequency



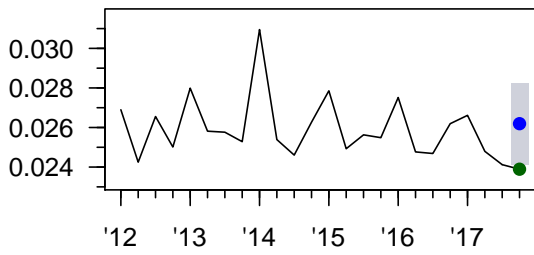
Florida Property Damage Frequency



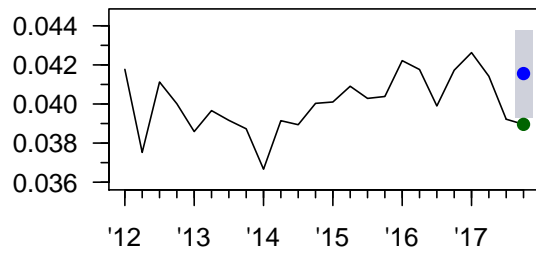
Hawaii Property Damage Frequency



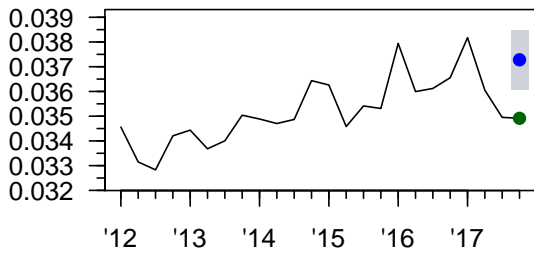
Iowa Property Damage Frequency



Louisiana Property Damage Frequency



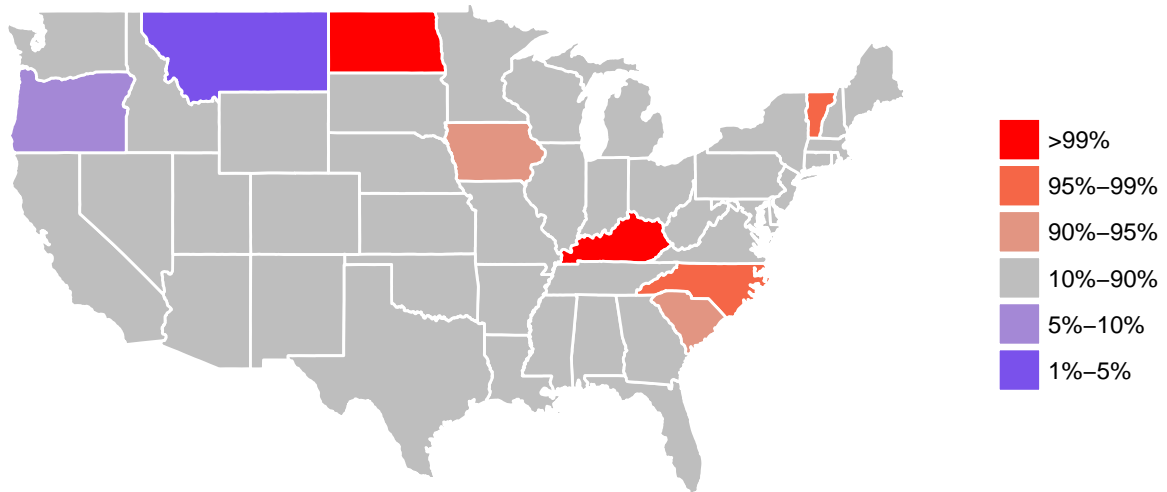
Washington Property Damage Frequency



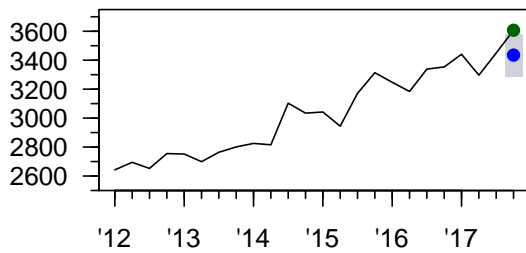
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

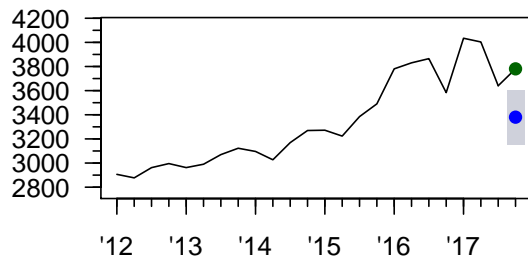
Property Damage Severity



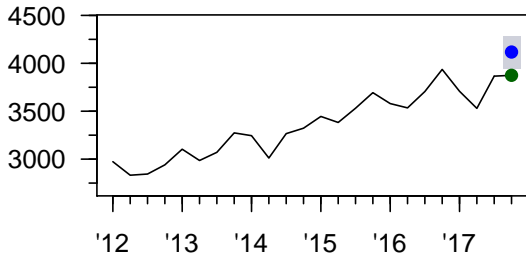
Iowa Property Damage Severity



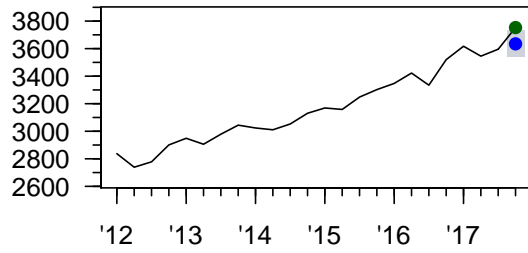
Kentucky Property Damage Severity



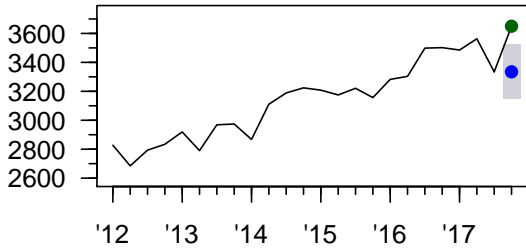
Montana Property Damage Severity



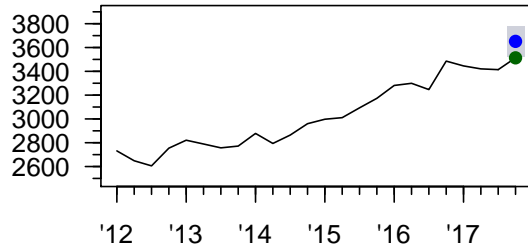
North Carolina Property Damage Severity



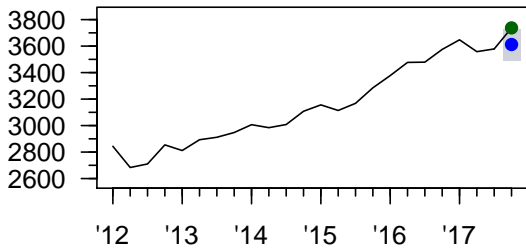
North Dakota Property Damage Severity



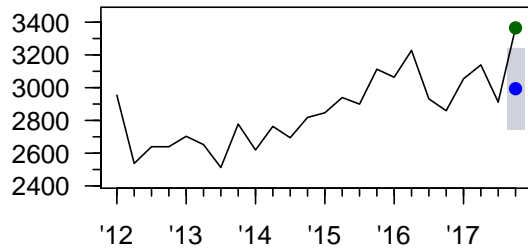
Oregon Property Damage Severity



South Carolina Property Damage Severity



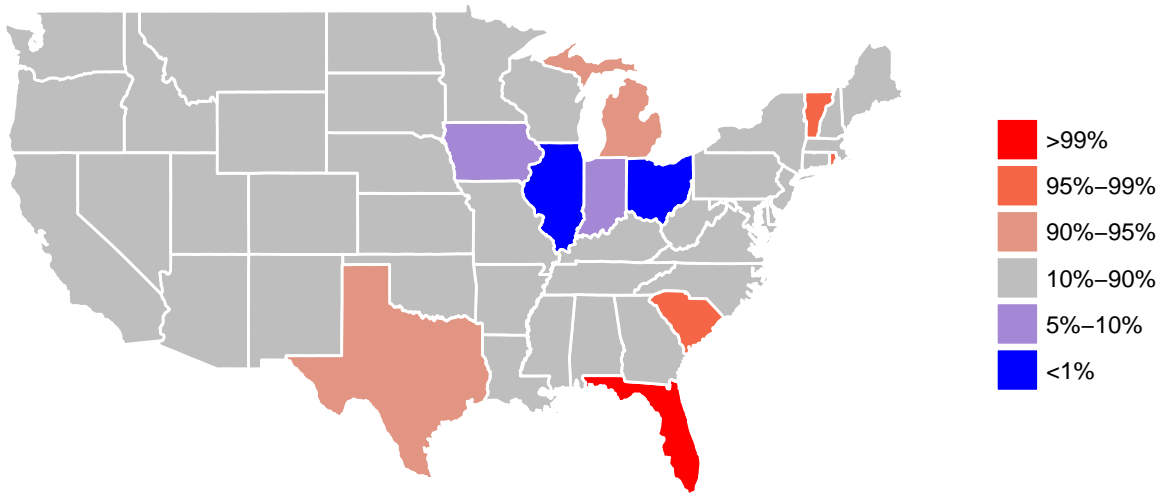
Vermont Property Damage Severity



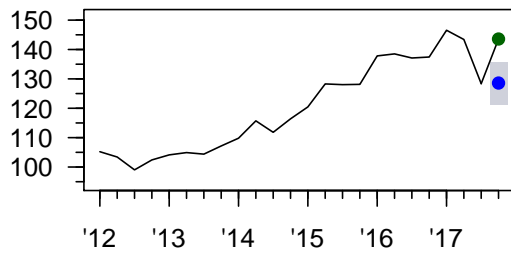
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

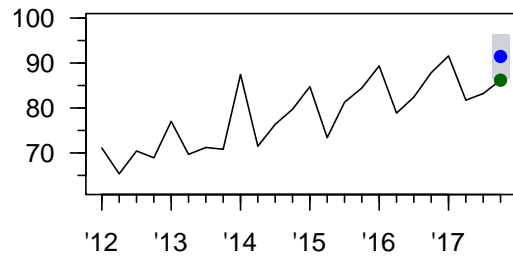
Property Damage Loss Cost



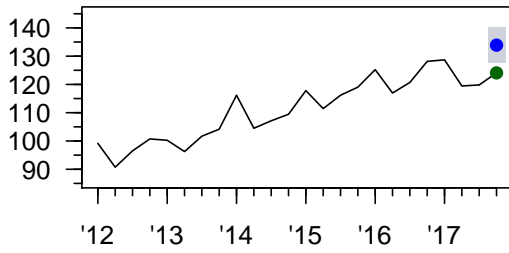
Florida Property Damage Loss Cost



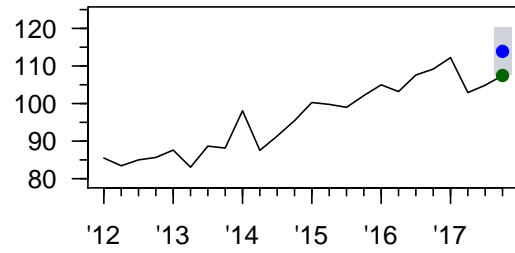
Iowa Property Damage Loss Cost



Illinois Property Damage Loss Cost



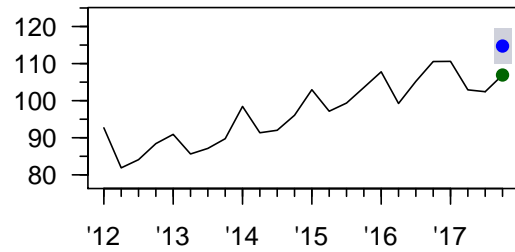
Indiana Property Damage Loss Cost



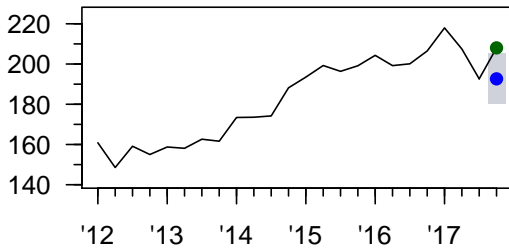
Michigan Property Damage Loss Cost



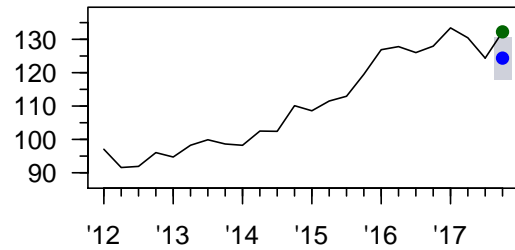
Ohio Property Damage Loss Cost



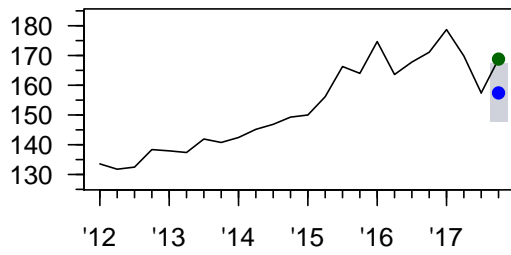
Rhode Island Property Damage Loss Cost



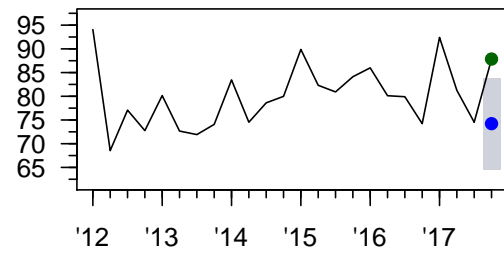
South Carolina Property Damage Loss Cost



**Texas Property Damage
Loss Cost**



**Vermont Property Damage
Loss Cost**



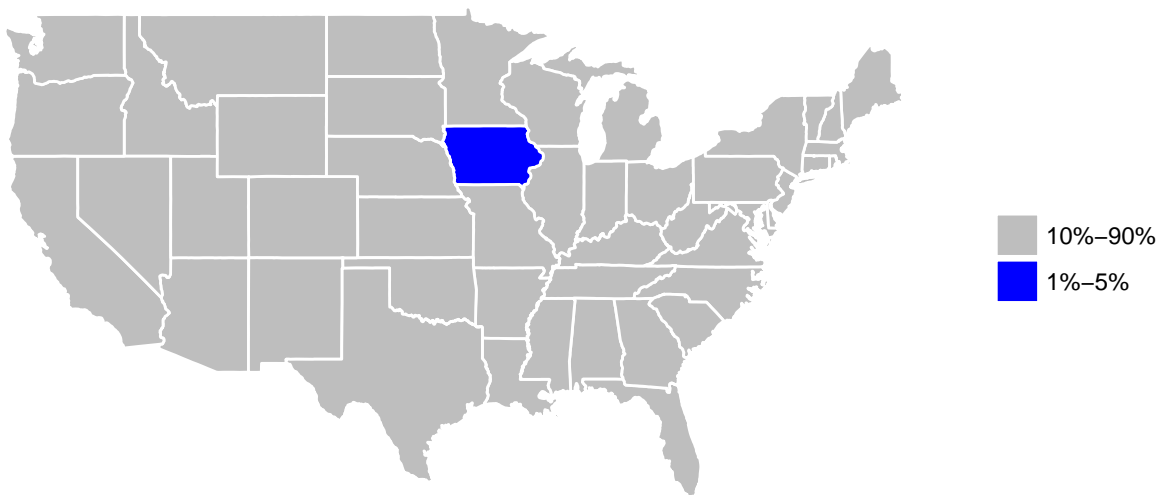
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

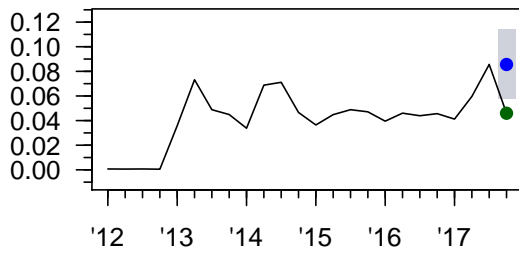
Comprehensive

Several states' comprehensive metrics seemed to be correlated with severe weather events. In Iowa, there was much less storm damage than usual, possibly leading to a small observed frequency metric. In Connecticut, storm damage was much higher than expected, thus the severity and loss cost metrics were high as well. New Hampshire had flash floods that caused significant damage and the severity and loss cost metrics were both extremely high. Oregon had a much lower damage amount than usual, thus our severity metric was also smaller than usual.

Comprehensive Frequency



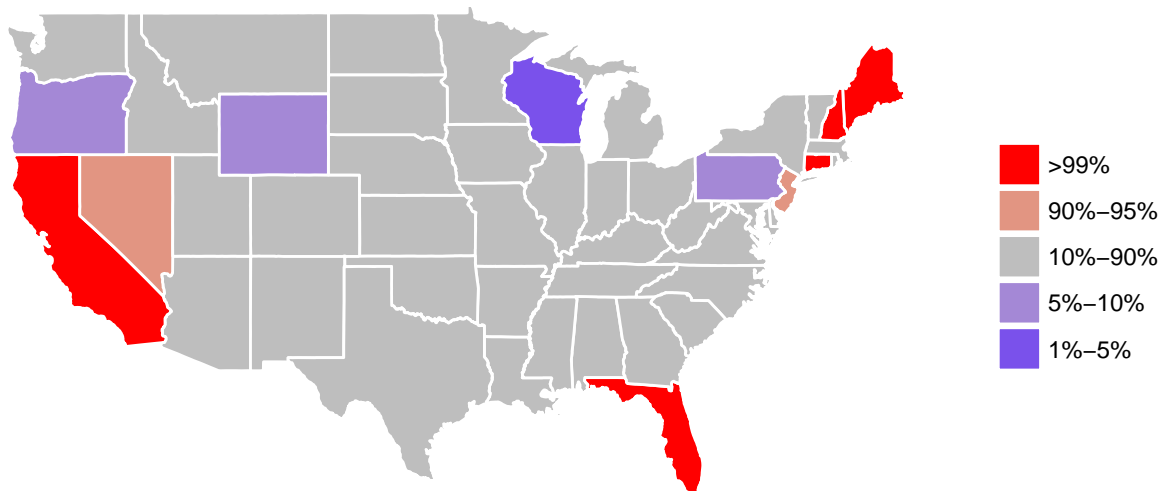
Iowa Comprehensive Frequency



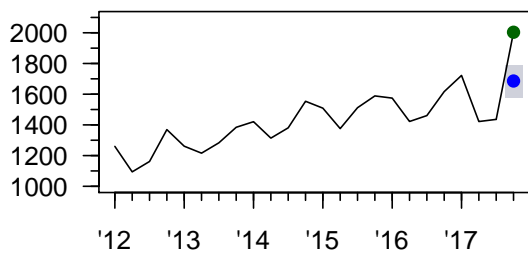
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

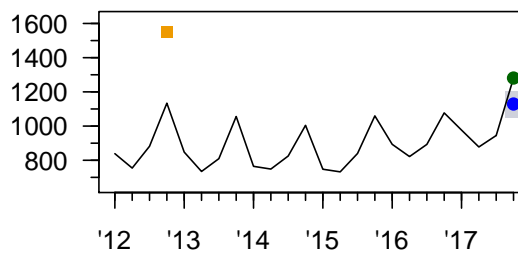
Comprehensive Severity



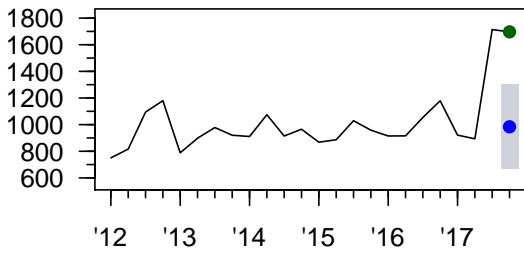
California Comprehensive Severity



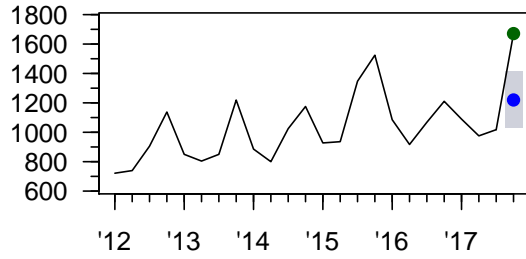
Connecticut Comprehensive Severity



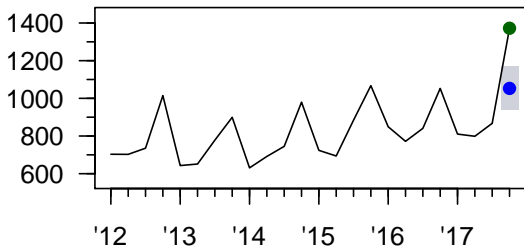
Florida Comprehensive Severity



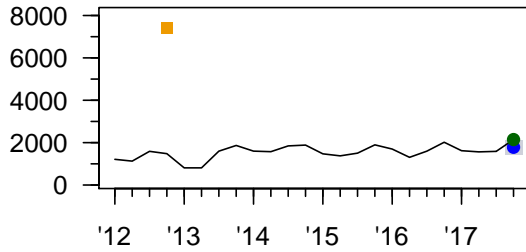
Maine Comprehensive Severity



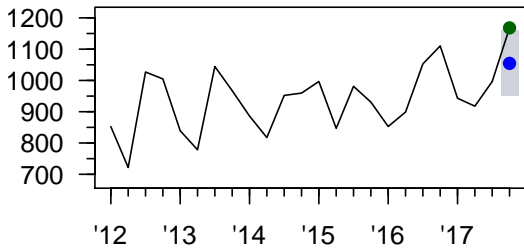
New Hampshire Comprehensive Severity



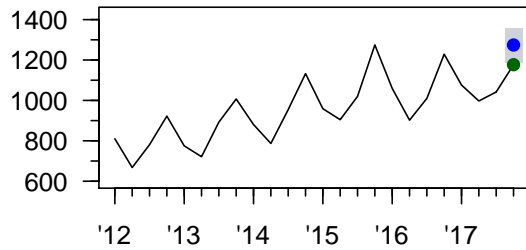
New Jersey Comprehensive Severity



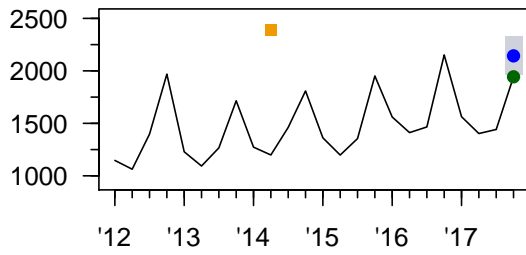
Nevada Comprehensive Severity



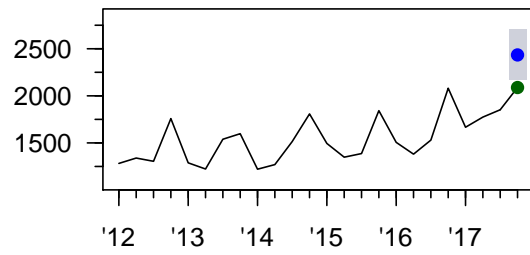
Oregon Comprehensive Severity



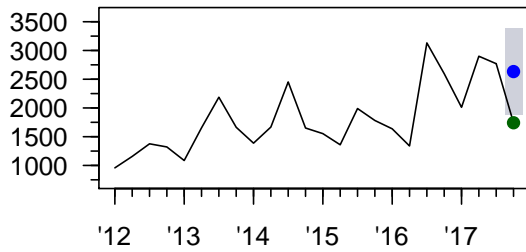
Pennsylvania Comprehensive Severity



Wisconsin Comprehensive Severity



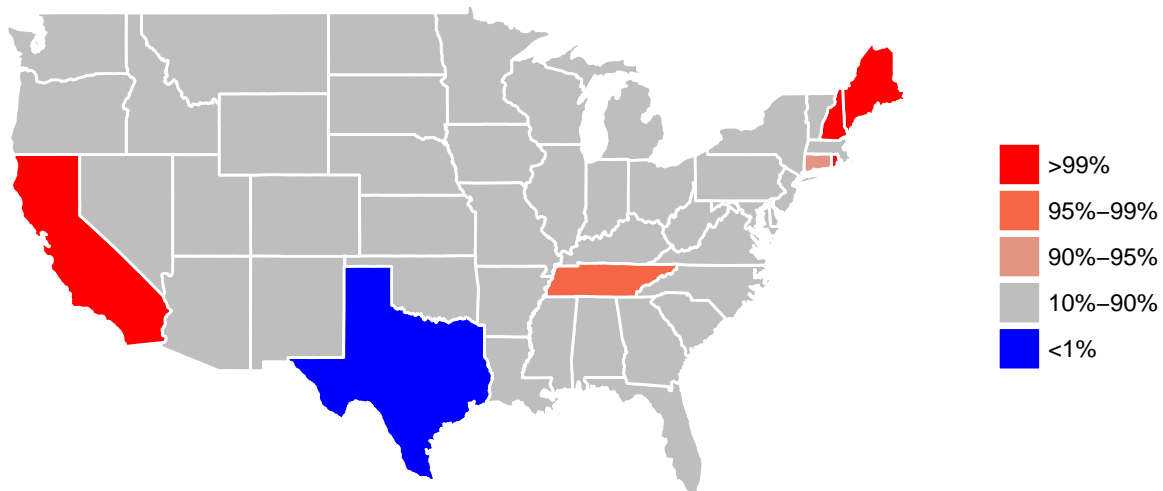
Wyoming Comprehensive Severity



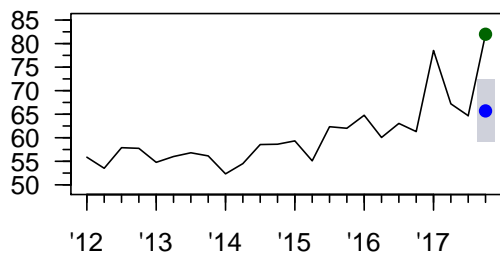
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

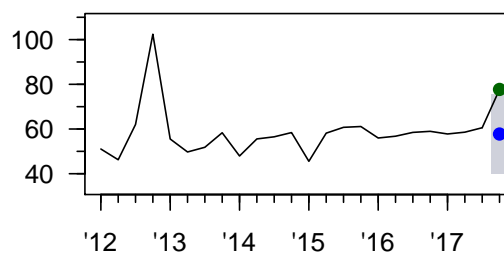
Comprehensive Loss Cost



California Comprehensive Loss Cost



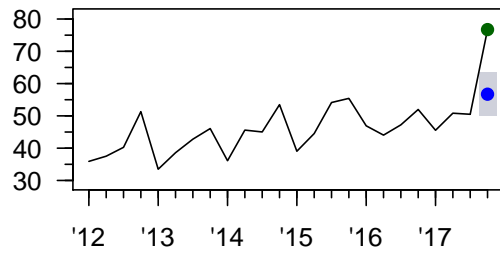
Connecticut Comprehensive Loss Cost



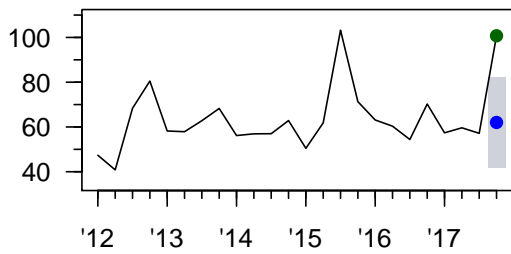
Maine Comprehensive Loss Cost



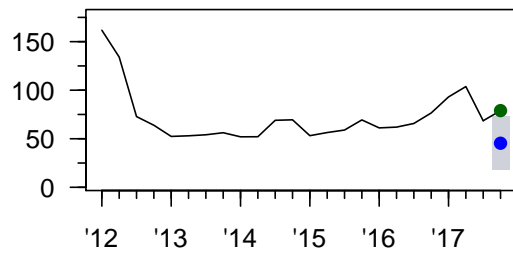
New Hampshire Comprehensive Loss Cost



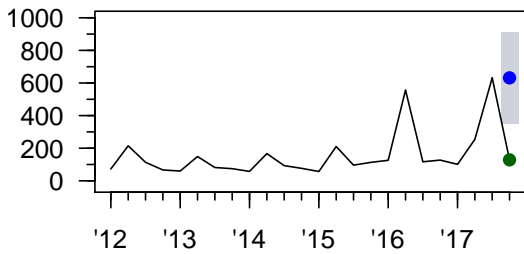
Rhode Island Comprehensive Loss Cost



Tennessee Comprehensive Loss Cost



Texas Comprehensive Loss Cost



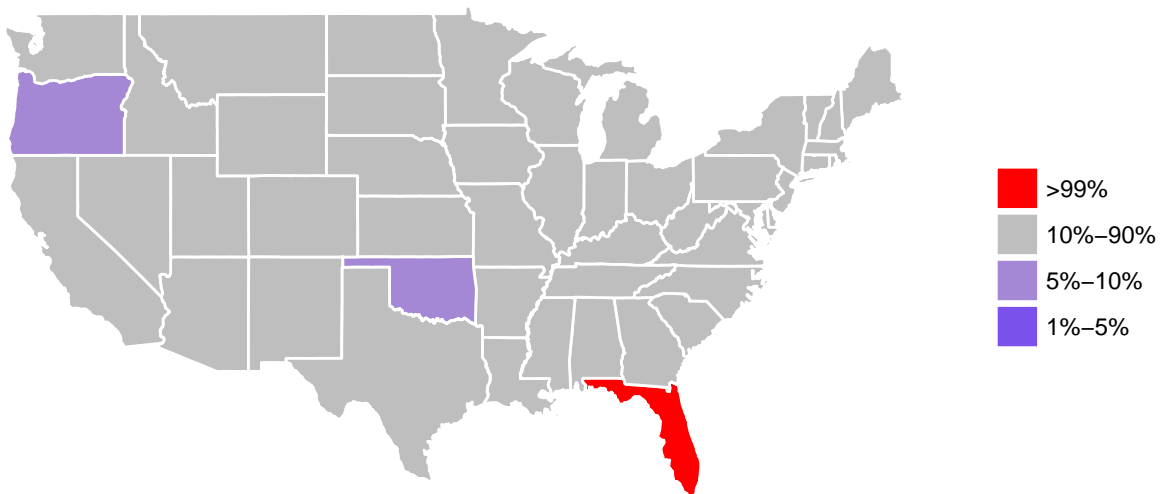
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

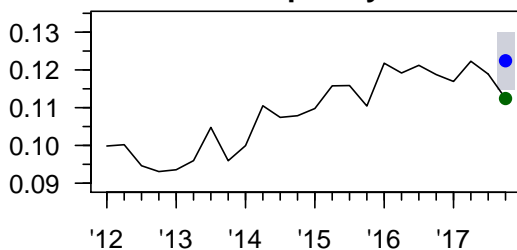
Collision

Several states' collision metrics often have an inverse relationship with severe weather damage amounts. Our guess is that due to severe weather, less vehicles are on the road and less collision claims are made. In Florida, there was very little weather damage this quarter, and our frequency, severity, and loss cost metrics were extremely high in this quarter. Georgia often experiences a variety of severe weather events in the fourth quarter of each year, but in 2017 those metrics were very low; this could have possibly lead to the high loss cost and severity metrics.

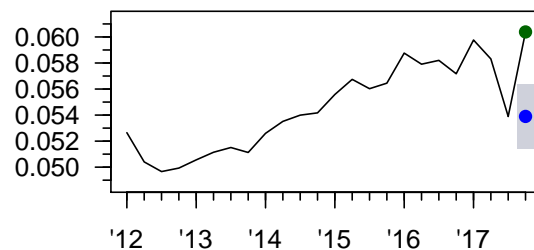
Collision Frequency



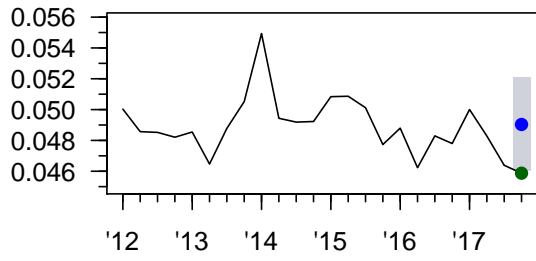
**District of Columbia
Collision
Frequency**



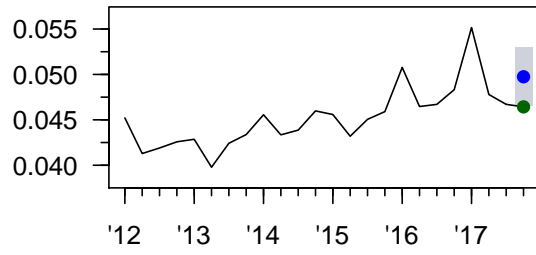
**Florida Collision
Frequency**



Oklahoma Collision Frequency



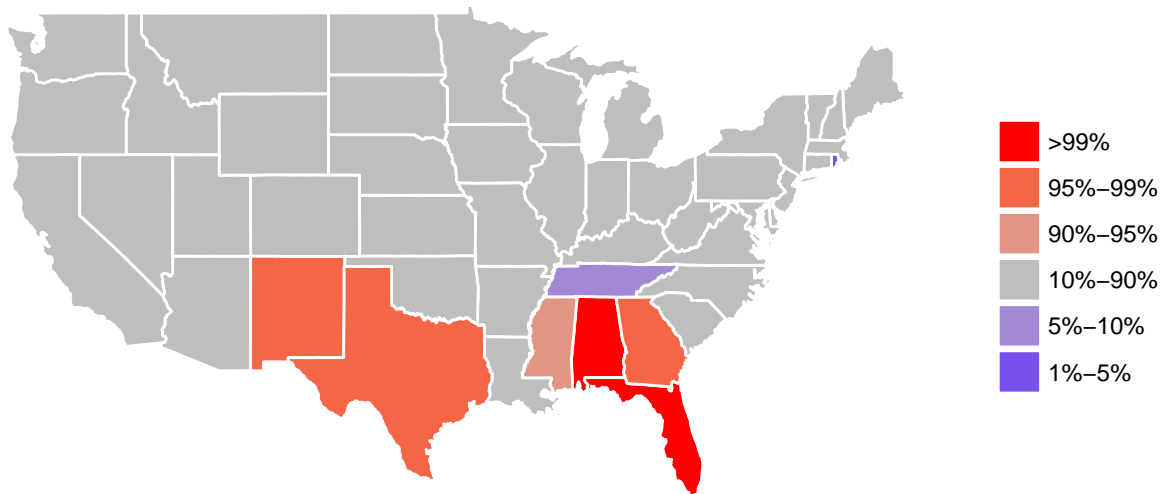
Oregon Collision Frequency



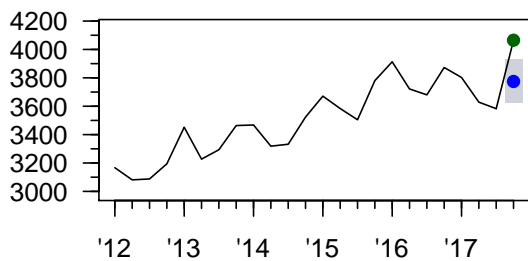
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

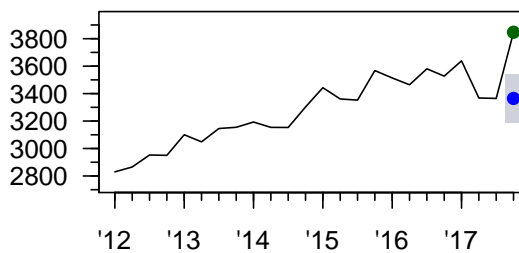
Collision Severity



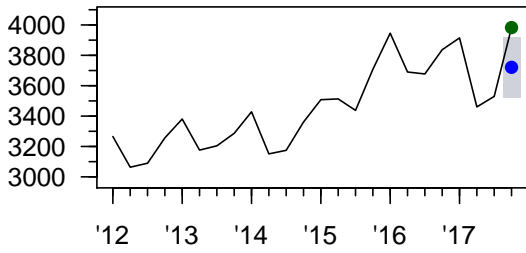
Alabama Collision Severity



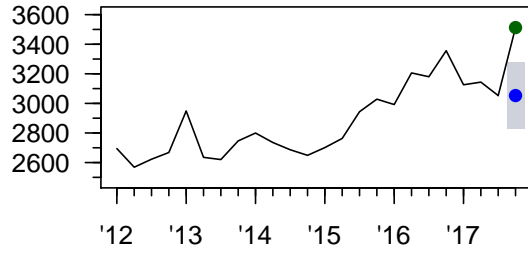
Florida Collision Severity



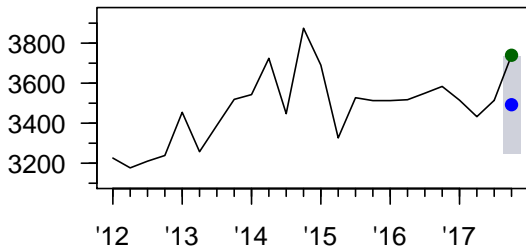
Georgia Collision Severity



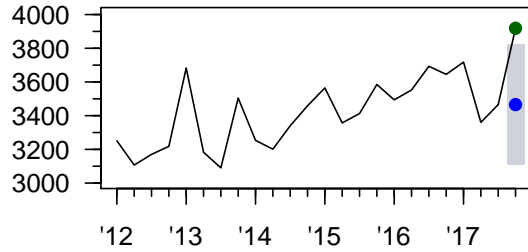
Hawaii Collision Severity



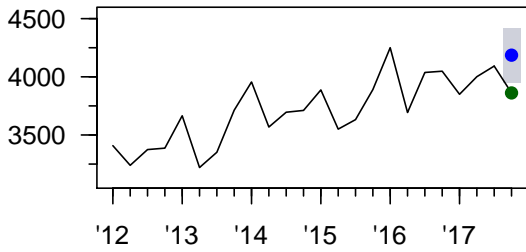
Mississippi Collision Severity



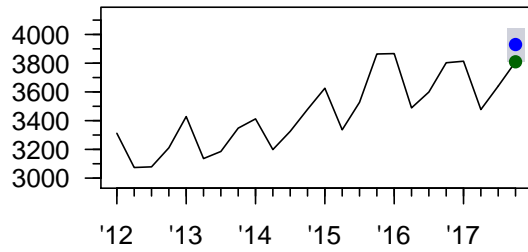
New Mexico Collision Severity



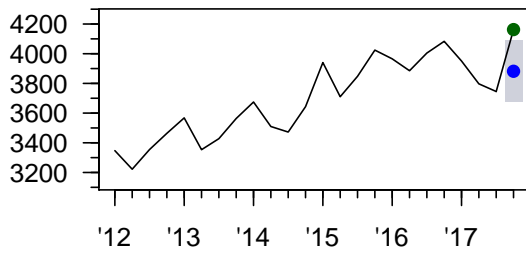
Rhode Island Collision Severity



Tennessee Collision Severity



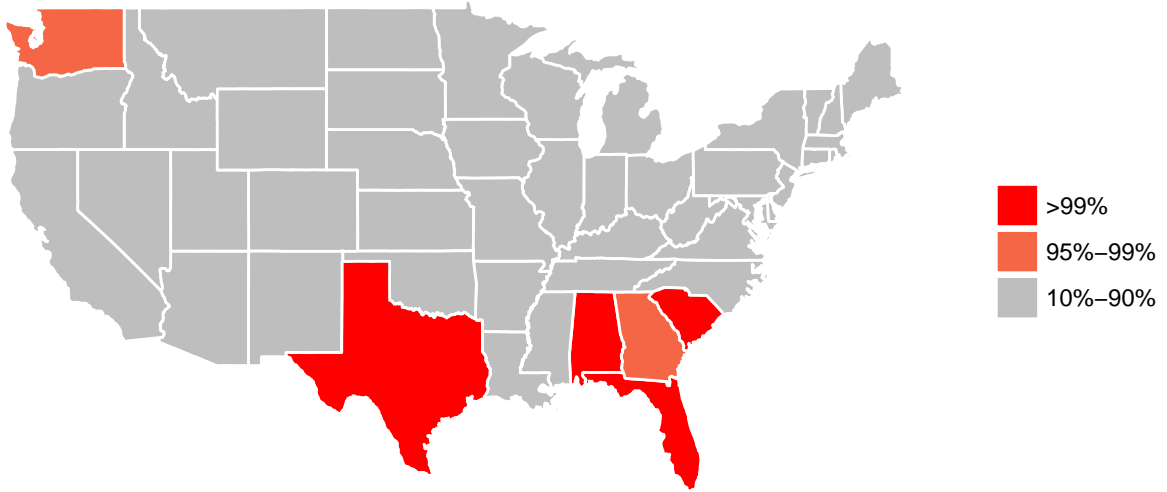
Texas Collision Severity



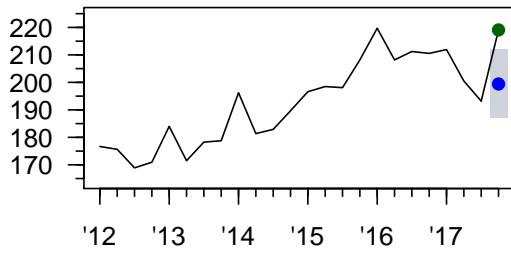
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

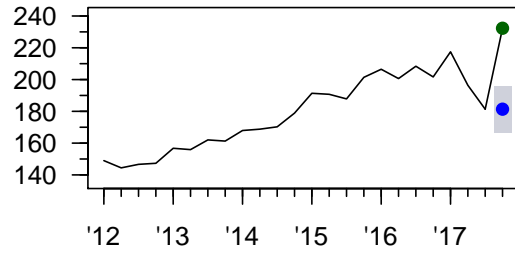
Collision Loss Cost



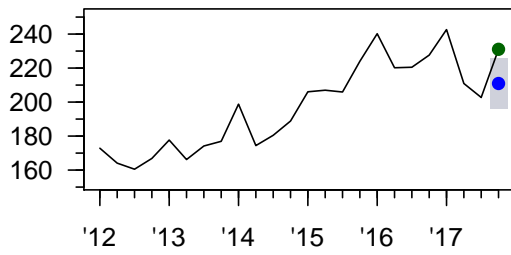
Alabama Collision Loss Cost



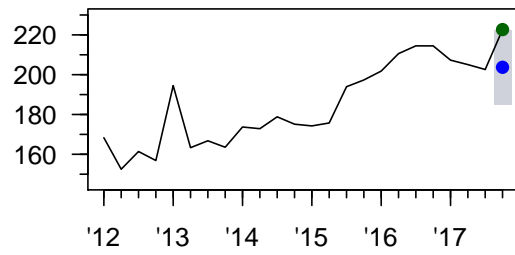
Florida Collision Loss Cost



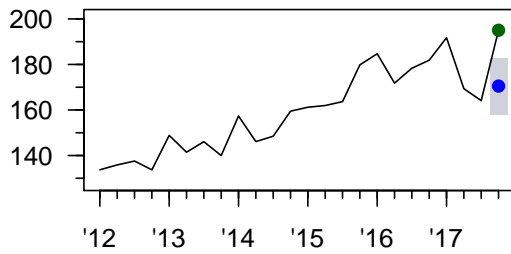
Georgia Collision Loss Cost



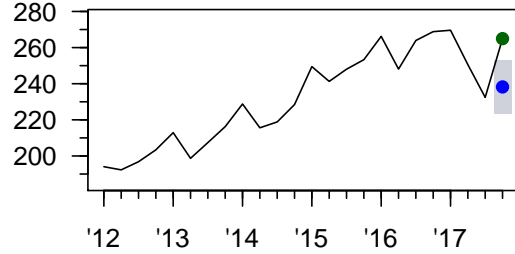
Hawaii Collision Loss Cost



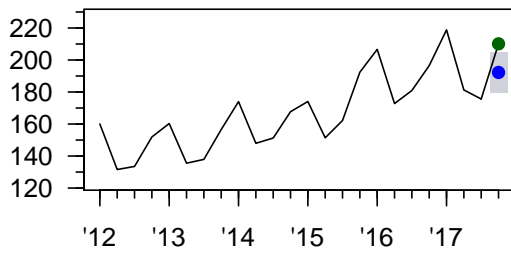
South Carolina Collision Loss Cost



Texas Collision Loss Cost



Washington Collision Loss Cost



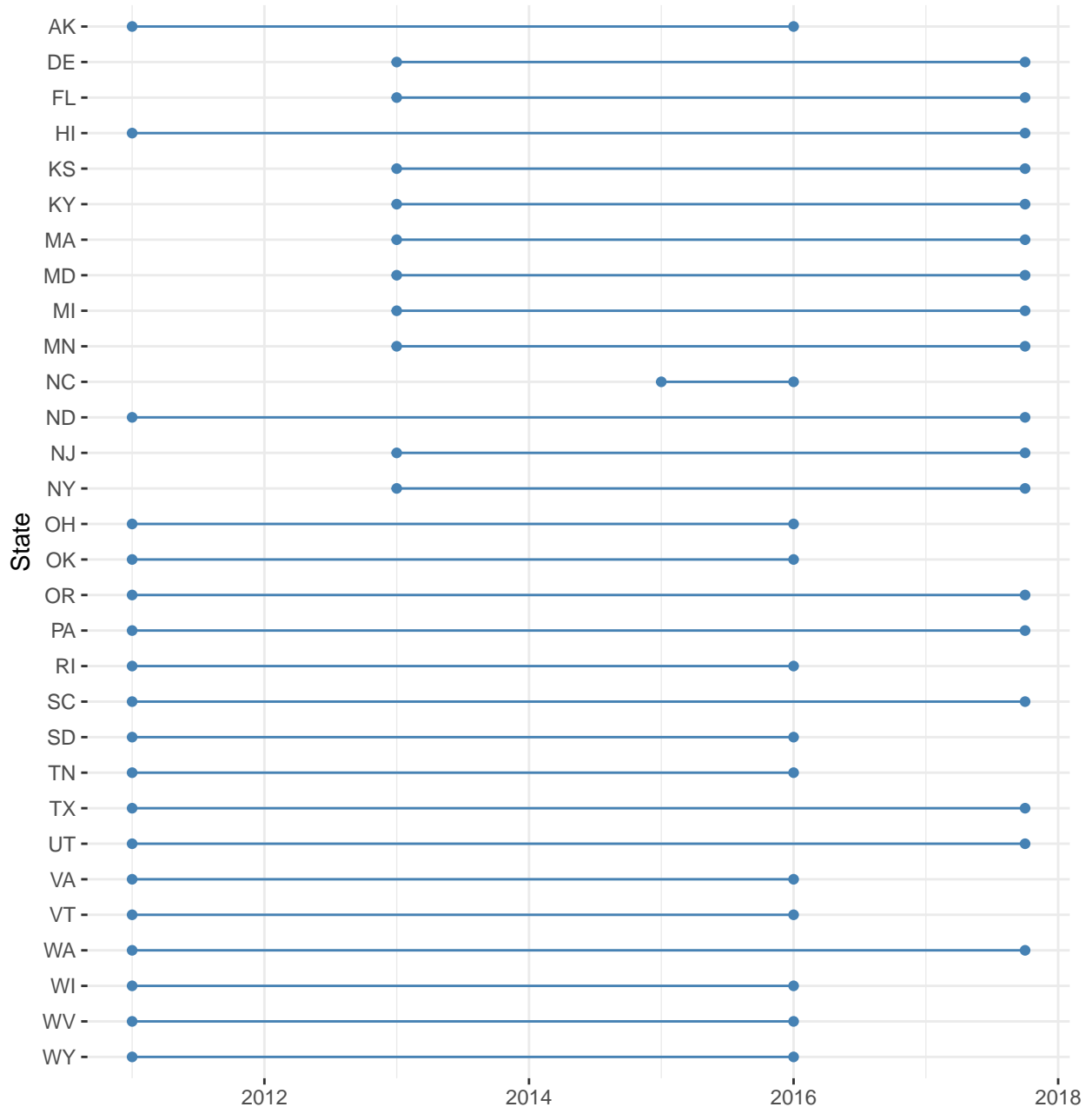
Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

Personal Injury Protection

The PIP data are different than the other insurance coverages. The amount of PIP data we have varies for each state; specifically, when the data begin and end is not consistent. The plot below gives us information about the start and end dates for each state we have data on.

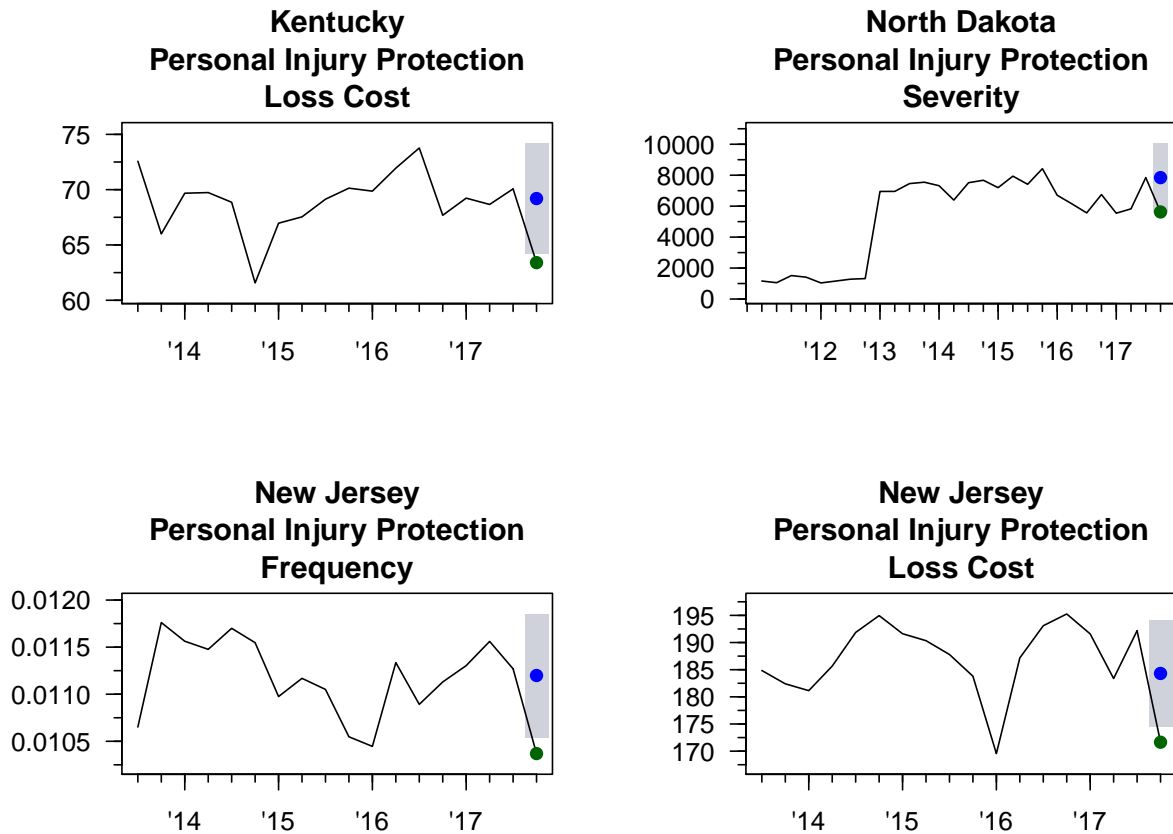
When the PIP Data Begin and End



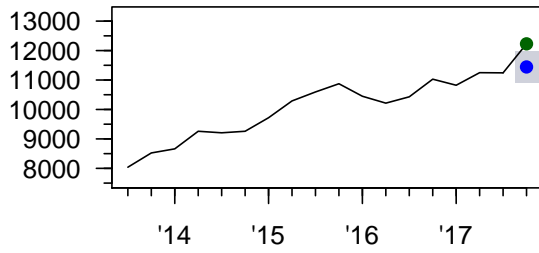
ARIMA model on PIP data

The same ARIMA model was used to forecast the 2017 4Q values on the PIP data. However, only 18 states had PIP data through quarter 4 of 2017; these were the states included in the model.

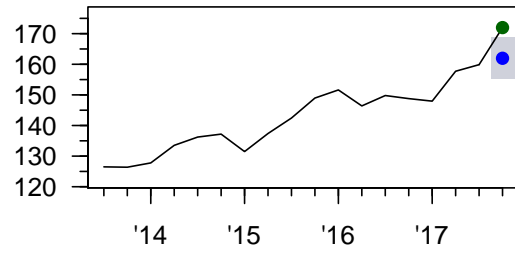
Of the 54 metrics forecasted, only 6 4Q metrics fell outside of the 10%-90% prediction interval. These plots are shown below.



**New York
Personal Injury Protection
Severity**



**New York
Personal Injury Protection
Loss Cost**



Time Series Legend

- Point Forecast
- True Value
- Prediction Interval
- Removed Outlier

Code

All the code used to create this document:

```
# libraries needed for the analysis
library(readxl)
library(data.table)
library(dplyr)
library(ggplot2)
library(maps)
library(tseries)
library(tsoutliers)
library(forecast)
library(zoo)
library(gridExtra)
library(devtools)
# install_github('sinhrks/ggfortify')
library(ggfortify)

# Reading in the raw Q3 and Q4 data:
setwd("~/Google Drive/AutoLossCosts")
ts4 <- read_excel("AutoLossCostv20_2017Q4.xlsm",
  sheet = "TSData") %>% data.table()
ts3 <- read_excel("AutoLossCostv20_2017Q3_TJ-EH.xlsm",
  sheet = "TSData") %>% data.table()

# Joining the 2 data files
key1 <- paste0(ts3$State, ts3$Coverage, ts3$Metric)
key2 <- paste0(ts4$State, ts4$Coverage, ts4$Metric)
ts_data <- ts3 %>% select(State:`2012 Q4`) %>% cbind(ts4 %>%
  select(`2013 Q1`:`2017 Q4`))

# outlier analysis and removal Smoothing just
# these series with outliers
outlier.metrics <- c("PA, Comp, Severity
MI, Comp, Severity
AR, Comp, Severity
ID, Coll, Frequency
LA, Comp, Severity
MT, BI, Severity
NC, Comp, Severity
SD, BI, Severity
TX, Comp, Severity
VA, Comp, Severity
CT, Comp, Severity
CO, Comp, Severity
NY, Comp, Severity
NJ, Comp, Severity")
remove <- x <- strsplit(outlier.metrics, "\n") %>%
  unlist()
remove <- gsub(pattern = " ", replacement = "",
  gsub(pattern = "\\,", replacement = "", remove))
boolVec <- paste0(ts_data$State, ts_data$Coverage,
```

```

    ts_data$Metric) %in% remove
# making sure I detected each row
sum(boolVec) == length(x)
# the outliers b4 they're smoothed
outliers <- ts_data[boolVec, ]

# cleaning these time series by replacing
# outliers
cleanTS <- function(x, critical = NULL) {
  outliers[x, ] %>% select(`2012 Q1`:`2017 Q4`) %>%
    t() %>% ts(start = c(2012, 1), end = c(2017,
4), frequency = 4) %>% tso(type = "A0",
    cval = critical)
}

smoothedValues <- rbind(c(cleanTS(1)$yadj), c(cleanTS(2)$yadj),
  c(cleanTS(3, 5)$yadj), c(cleanTS(4)$yadj), c(cleanTS(5,
4.1)$yadj), c(cleanTS(6, 5)$yadj), c(cleanTS(7)$yadj),
  c(cleanTS(8)$yadj), c(cleanTS(9)$yadj), c(cleanTS(10)$yadj),
  c(cleanTS(11)$yadj), c(cleanTS(12)$yadj), c(cleanTS(13,
5)$yadj), c(cleanTS(14, 5)$yadj))
smoothedValuesDF <- cbind(outliers[, 1:3], smoothedValues)
names(smoothedValuesDF) <- names(outliers)

# data frame w/ outliers smoothed
tsAdj <- ts_data[!boolVec, ]
tsAdj <- tsAdj %>% rbind(smoothedValuesDF) %>% arrange(Coverage,
  State, Metric)

# loop through each 612 values and predict the
# 4Q data value
df.1 <- data.frame(Point.Forecast = c(1), Lo.90 = c(1),
  Hi.90 = c(1), Lo.95 = c(1), Hi.95 = c(1), Lo.99 = c(1),
  Hi.99 = c(1))
for (i in 1:nrow(tsAdj)) {
  ar.fit <- tsAdj[i, ] %>% select(`2012 Q1`:`2017 Q3`) %>%
    t() %>% ts(start = c(2012, 1), end = c(2017,
3), frequency = 4) %>% auto.arima()
  pred <- forecast(ar.fit, h = 1, level = c(90,
95, 99)) %>% data.frame()
  df.1[i, ] <- pred
}

pred.df <- tsAdj %>% select(State, Coverage, Metric,
  `2017 Q4`) %>% cbind(df.1) %>% data.table()
pred.df$true <- pred.df$`2017 Q4`
pred.df[, `:=`(pi, ifelse(true < Lo.99, "<1%", ifelse(true <
Lo.95, "1%-5%", ifelse(true < Lo.90, "5%-10%",
ifelse(true < Hi.90, "10%-90%", ifelse(true <
Hi.95, "90%-95%", ifelse(true < Hi.99, "95%-99%",
">99%"))))))))]
pred.df <- pred.df %>% select(State:Metric, true,

```

```

Point.Forecast, pi)

# Saved the results, and re-read in the data
pred.df1 <- read_excel("AutoLossCostv20_2017Q4-TJ.xlsx",
  sheet = "Q4Forecast") %>% data.table()
state.mappings <- read_excel("AutoLossCostv20_2017Q3-TJ-EH.xlsx",
  sheet = "Sheet1")[, 6:8] %>% data.table()
ts_adj <- read_excel("AutoLossCostv20_2017Q4-TJ.xlsx",
  sheet = "TS_adjusted") %>% data.table()
map.data <- pred.df1 %>% left_join(state.mappings,
  by = c(State = "Abbreviation")) %>% right_join(map_data("state"),
  by = c(LCState = "region")) %>% data.table()

# data cleaning
pi.levels <- c("<1%", "1%-5%", "5%-10%", "10%-90%",
  "90%-95%", "95%-99%", ">99%")
map.data[, `:=`(my.col, ifelse(pi == pi.levels[1],
  1, ifelse(pi == pi.levels[2], 2, ifelse(pi ==
  pi.levels[3], 3, ifelse(pi == pi.levels[4],
  4, ifelse(pi == pi.levels[5], 5, ifelse(pi ==
  pi.levels[6], 6, 7))))))]
map.data$pi <- factor(map.data$pi, levels = pi.levels)
coverage.df <- data.frame(abbrev = c("BI", "Coll",
  "Comp", "PD", "PIP"), full = c("Bodily Injury",
  "Collision", "Comprehensive", "Property Damage",
  "Personal Injury Protection"))

# making functions to make maps and ARIMA plots
make.map <- function(coverage = "PD", metric = "Severity") {
  map.dat <- filter(map.data, Coverage == coverage,
    Metric == metric)
  labs <- sort(unique(map.dat$pi))
  brks <- sort(unique(map.dat$my.col))

  if (metric == "LossCost") {
    plt.title <- paste0(coverage.df$full[coverage.df$abbrev ==
      coverage], " Loss Cost")
  } else {
    plt.title <- paste0(coverage.df$full[coverage.df$abbrev ==
      coverage], " ", metric)
  }

  ggplot(data = map.dat) + geom_polygon(aes(x = long,
    y = lat, fill = my.col, group = group),
    color = "white") + coord_fixed(1.3) + labs(x = NULL,
    y = NULL, title = plt.title) + theme_bw() +
    theme(panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.border = element_blank(), axis.ticks = element_blank(),
      axis.text = element_blank()) + theme(legend.title = element_blank()) +
    guides(fill = guide_legend(reverse = TRUE)) +
    scale_fill_gradient2(midpoint = 4, low = "blue",

```

```

        mid = "grey", high = "red", breaks = brks,
        labels = labs)
}

arima.plot <- function(state = "AL", coverage = "Comp",
  metric = "Severity", pi = 90) {
  dat <- ts_adj %>% filter(State == state, Coverage ==
    coverage, Metric == metric)
  orig <- ts_data %>% filter(State == state, Coverage ==
    coverage, Metric == metric) %>% select(`2012 Q1`:`2017 Q4`)
  true.val <- dat$`2017 Q4`
  ar.fit <- dat %>% select(`2012 Q1`:`2017 Q3`) %>%
    t() %>% ts(start = c(2012, 1), end = c(2017,
    3), frequency = 4) %>% auto.arima()
  params <- arimaorder(ar.fit)
  pred <- forecast(ar.fit, h = 1, level = pi)
  upr <- max(select(dat, `2012 Q1`:`2017 Q4`),
    pred$upper, orig)
  lwr <- min(select(dat, `2012 Q1`:`2017 Q4`),
    pred$lower, orig)
  range <- upr - lwr
  cov.full <- coverage.df$full[coverage.df$abbrev ==
    coverage]
  state.full <- state.mappings$State[state.mappings$Abbreviation ==
    state]

  main.l <- nchar(paste0(state.full, " ", cov.full))

  if (metric == "LossCost") {
    if (main.l >= 28) {
      main.title <- paste0(state.full, "\n",
        cov.full, "\n", " Loss Cost")
    } else {
      main.title <- paste0(state.full, " ",
        cov.full, "\n", " Loss Cost")
    }
  } else {
    if (main.l >= 28) {
      main.title <- paste0(state.full, "\n",
        cov.full, "\n", metric)
    } else {
      main.title <- paste0(state.full, " ",
        cov.full, "\n", metric)
    }
  }
}

plot(pred, ylim = c(lwr - 0.1 * range, upr +
  0.1 * range), las = 1, main = main.title,
  xaxt = "n")
lines(c(2017.5, 2017.75), c(dat$`2017 Q3`, dat$`2017 Q4`),
  col = "black")
points(2017.75, true.val, col = "darkgreen",
  pch = 19)

```

```

axis(1, at = c(2012:2017), labels = c("'12",
  "'13", "'14", "'15", "'16", "'17"))
Hmisc::minor.tick(nx = 4)

out <- tryCatch(removedBoolean %>% filter(State ==
  state, Coverage == coverage, Metric == metric) ==
  FALSE, error = function(x) x)
if (!(any(class(out) == "error"))) {
  ind <- removedBoolean %>% filter(State ==
  state, Coverage == coverage, Metric ==
  metric) == FALSE
  y. <- filter(outliers, State == state, Coverage ==
  coverage, Metric == metric)[ind] %>%
  as.numeric()
  x. <- names(ts_adj)[ind] %>% as.yearqtr() %>%
  as.numeric()
  points(x., y., col = "orange2", pch = 15)
}
}

arima.plots <- function(cov, met, loops = "all",
  pi = 90) {
  x <- pred.df1 %>% filter(pi != "10%-90%", Coverage ==
  cov, Metric == met) %>% select(State, Coverage,
  Metric)
  if (loops == "all") {
    for (i in 1:nrow(x)) {
      arima.plot(x[i, 1], x[i, 2], x[i, 3],
        pi)
    }
  } else {
    for (i in 1:loops) {
      arima.plot(x[i, 1], x[i, 2], x[i, 3],
        pi)
    }
  }
}

# Printing all the maps and plots

## BI
make.map("BI", "Frequency")
par(mfrow = c(1, 2))
arima.plots("BI", "Frequency")
l.names <- c("Point Forecast", "True Value", "Prediction Interval",
  "Removed Outlier")
l.cols <- c("blue", "darkgreen", "grey", "orange2")
legend.plt <- function() {
  par(mfrow = c(1, 1))
  par(mar = c(0, 0, 0, 0))
  plot.new()
  legend(x = "top", inset = 0, title = "Time Series Legend",
    legend = l.names, fill = l.cols, cex = 0.9,

```

```

        horiz = TRUE, bty = "n")
}

legend.plt()

make.map("BI", "Severity")
par(mfrow = c(1, 2))
arima.plots("BI", "Severity")
legend.plt()

make.map("BI", "LossCost")
par(mfrow = c(1, 2))
arima.plots("BI", "LossCost")
legend.plt()

## Property Damage
make.map("PD", "Frequency")
par(mfrow = c(1, 2))
arima.plots("PD", "Frequency")
legend.plt()

make.map("PD", "Severity")
par(mfrow = c(1, 2))
arima.plots("PD", "Severity")
legend.plt()

make.map("PD", "LossCost")
par(mfrow = c(1, 2))
arima.plots("PD", "LossCost")
legend.plt()

## Comprehensive
make.map("Comp", "Frequency")
par(mfrow = c(1, 2))
arima.plots("Comp", "Frequency")
legend.plt()

make.map("Comp", "Severity")
par(mfrow = c(1, 2))
arima.plots("Comp", "Severity")
legend.plt()

make.map("Comp", "LossCost")
par(mfrow = c(1, 2))
arima.plots("Comp", "LossCost")
legend.plt()

## Collision
make.map("Coll", "Frequency")
par(mfrow = c(1, 2))

```

```

arima.plots("Coll", "Frequency")
legend.plt()

make.map("Coll", "Severity")
par(mfrow = c(1, 2))
arima.plots("Coll", "Severity")
legend.plt()

make.map("Coll", "LossCost")
par(mfrow = c(1, 2))
arima.plots("Coll", "LossCost")
legend.plt()

# the PIP analysis
pip <- read_excel("AutoLossCostv20_2017Q4-TJ.xlsx",
  sheet = "PIPdata") %>% select(StateName, Year,
  Quarter, Frequency__1, Severity__1, LossCost__1) %>%
  data.table()
names(pip) <- c("State", "year", "quarter", "Frequency",
  "Severity", "LossCost")
pip$time <- paste0(pip$year, " Q", pip$quarter)
pip <- pip[!is.na(State), ] %>% select(-year, -quarter)
pip <- melt.data.table(pip, id.vars = c("State",
  "time"), variable.name = "Metric", value.name = "value")
pip <- dcast.data.table(pip, State + Metric ~ time,
  fun.aggregate = sum)
# removing CW and DC
pip <- pip[!(pip$State %in% c("CW", "DC")), ]

missing.df <- data.frame(State = "0", Start = 0,
  End = 0, Data.Missing = 0)
statez <- unique(pip$State)

missing.fun <- function(state) {
  temp <- pip %>% filter(State == state, Metric ==
    "Frequency") %>% data.table()
  temp <- melt.data.table(temp, id.vars = c("State",
    "Metric"), variable.name = "time")
  temp$time <- as.yearqtr(temp$time, format = "%Y Q%q")
  times <- temp[temp$value != 0, ]$time
  first <- times[1]
  last <- times[length(times)]
  numMis.inbetween <- sum(temp$value[inds <- temp$time >
    first & temp$time < last] == 0)
  data.frame(State = state, Start = first, End = last,
    Data.Missing = numMis.inbetween)
}

for (i in 1:length(statez)) {
  missing.df <- rbind(missing.df, missing.fun(statez[i]))
}
state.df <- missing.df[-1, ]

```



```

state.df$Data.Missing <- NULL
state.df$End <- as.yearqtr(state.df$End, format = "%Y Q%q")
state.df$Start <- as.yearqtr(state.df$Start, format = "%Y Q%q")
row.names(state.df) <- NULL

# Put on ggplot
state.df3 <- state.df %>% data.table()
state.df3$Start <- state.df3$Start %>% as.numeric()
state.df3$End <- state.df3$End %>% as.numeric()
state.df3 <- melt.data.table(state.df3, id.vars = "State",
  variable.name = "When", value.name = "Times")
state.df3 <- state.df3 %>% arrange(desc(State))
ggplot(state.df3, aes(x = State, y = Times, group = State)) +
  geom_point(color = "steelblue") + geom_line(color = "steelblue") +
  coord_flip() + labs(title = "When the PIP Data Begin and End",
  y = NULL) + theme(legend.position = "none") +
  theme_bw() + theme(panel.border = element_blank()) +
  scale_x_discrete(limits = rev(levels(state.df3$State)[-1]))

missing.df <- data.frame(State = "0", Start = 0,
  End = 0, Data.Missing = 0)
missing.fun <- function(state) {
  temp <- pip %>% filter(State == state, Metric ==
    "Frequency") %>% data.table()
  temp <- melt.data.table(temp, id.vars = c("State",
    "Metric"), variable.name = "time")
  temp$time <- as.yearqtr(temp$time, format = "%Y Q%q")
  times <- temp[temp$value != 0, ]$time
  first <- times[1]
  last <- times[length(times)]
  numMis.inbetween <- sum(temp$value[inds <- temp$time >
    first & temp$time < last] == 0)
  data.frame(State = state, Start = first, End = last,
    Data.Missing = numMis.inbetween)
}

for (i in 1:length(statez)) {
  missing.df <- rbind(missing.df, missing.fun(statez[i]))
}

state.df <- missing.df[-1, ]
state.df$Data.Missing <- NULL
state.df$End <- as.yearqtr(state.df$End, format = "%Y Q%q")
state.df$Start <- as.yearqtr(state.df$Start, format = "%Y Q%q")
row.names(state.df) <- NULL

tsdat <- pip[pip$`2017 Q4` != 0, ]
pip.pred.plot <- function(st, met, PI = 90) {
  start.date <- state.df %>% filter(State == st) %>%
    select(Start)
  loop.dat <- tsdat %>% filter(State == st, Metric ==
    met) %>% select(-State, -Metric) %>% t()
  inds <- row.names(loop.dat) %>% as.yearqtr() >=
    start.date
}

```

```

loop.dat <- loop.dat[inds, ]
true <- loop.dat[length(loop.dat)]
loop.dat <- loop.dat[-length(loop.dat)]
loop.ts <- ts(loop.dat, start = c(as.numeric(substr(start.date,
  0, 4)), as.numeric(substr(start.date, 4,
  5))), end = c(2017, 3), frequency = 4)
fit <- auto.arima(loop.ts)
fc <- forecast(fit, h = 1, level = PI)
min. <- min(loop.dat, true, fc$lower)
max. <- max(loop.dat, true, fc$upper)
params <- arimaorder(fit)
cov.full <- coverage.df$full[coverage.df$abbrev ==
  "PIP"]
state.full <- state.mappings$State[state.mappings$Abbreviation ==
  st]

if (met == "LossCost") {
  main.title <- paste0(state.full, "\n", cov.full,
    "\n", "Loss Cost")
} else {
  main.title <- paste0(state.full, "\n", cov.full,
    "\n", met)
}

plot(fc, ylim = c(min. - 0.1 * (max. - min.),
  max. + 0.1 * (max. - min.)), main = main.title,
  las = 1, axt = "n")
lines(c(2017.5, 2017.75), c(loop.ts[length(loop.ts)],
  true), col = "black")
points(2017.75, true, col = "darkgreen", pch = 19)
axis(1, at = c(2012:2017), labels = c("'12",
  "'13", "'14", "'15", "'16", "'17"))
Hmisc::minor.tick(nx = 4)
}

par(mfrow = c(1, 2))
pip.pred.plot("KY", "LossCost", 90)
pip.pred.plot("ND", "Severity", 90)
pip.pred.plot("NJ", "Frequency", 90)
pip.pred.plot("NJ", "LossCost", 90)
pip.pred.plot("NY", "Severity", 90)
pip.pred.plot("NY", "LossCost", 90)
legend.plt()

```