Advanced Visualization

Brian A. Fannin

August 21, 2017

Overview

ggplot2

ggplot2 developed by Hadley Wickham, based on the "grammar of graphics". Particularly well suited (IMHO) for multi-dimensional, multivariate analysis.

Requires 3 things:

1. Data
2. Mapping
3. Geometric layers

Data

```
library(raw)
data(RegionExperience)
library(ggplot2)

basePlot <- ggplot(RegionExperience)
class(basePlot)
## [1] "gg"     "ggplot"
```

Notice that we assigned the result of the function call to an object called `basePlot`. This means we don't automatically get output. Take a bit of time to have a look at what's contained in the `basePlot` object.

Aesthetics

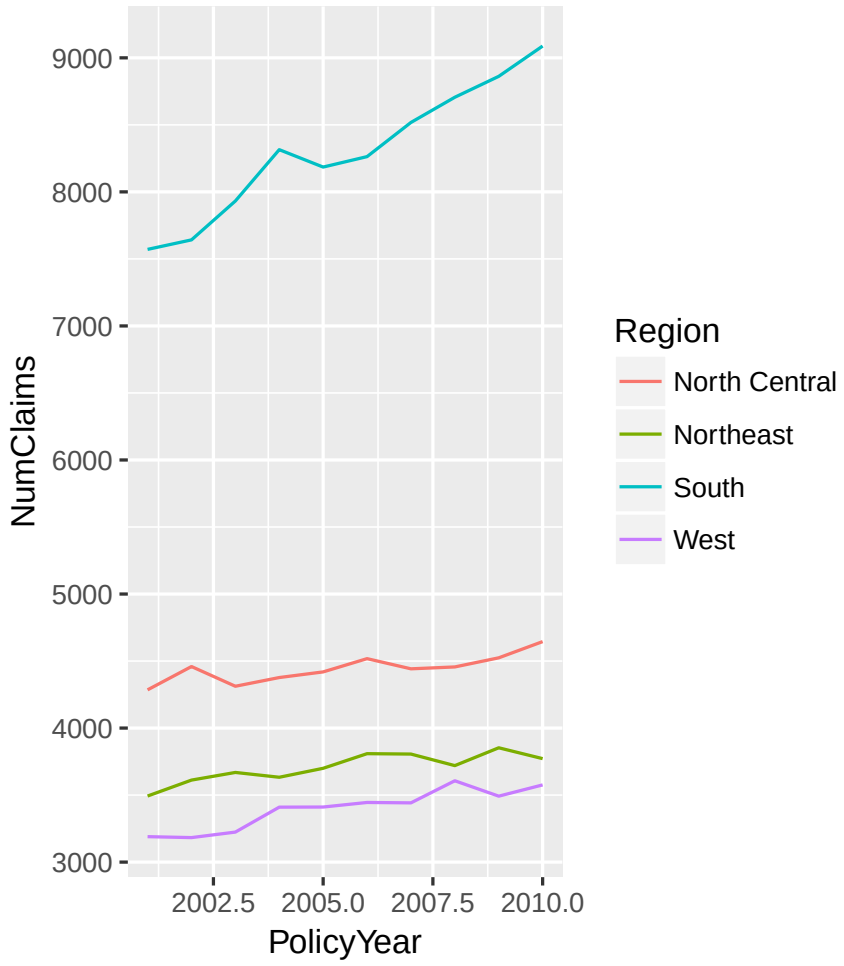
Aesthetics are anything visible on the plot. When an aesthetic is "mapped", it means that the data will define how the aesthetic appears. We'll map our aesthetics with the `aes` function.

```
basePlot <- basePlot + aes(x = PolicyYear, y = NumClaims,
    color = Region)
```

Adding layers

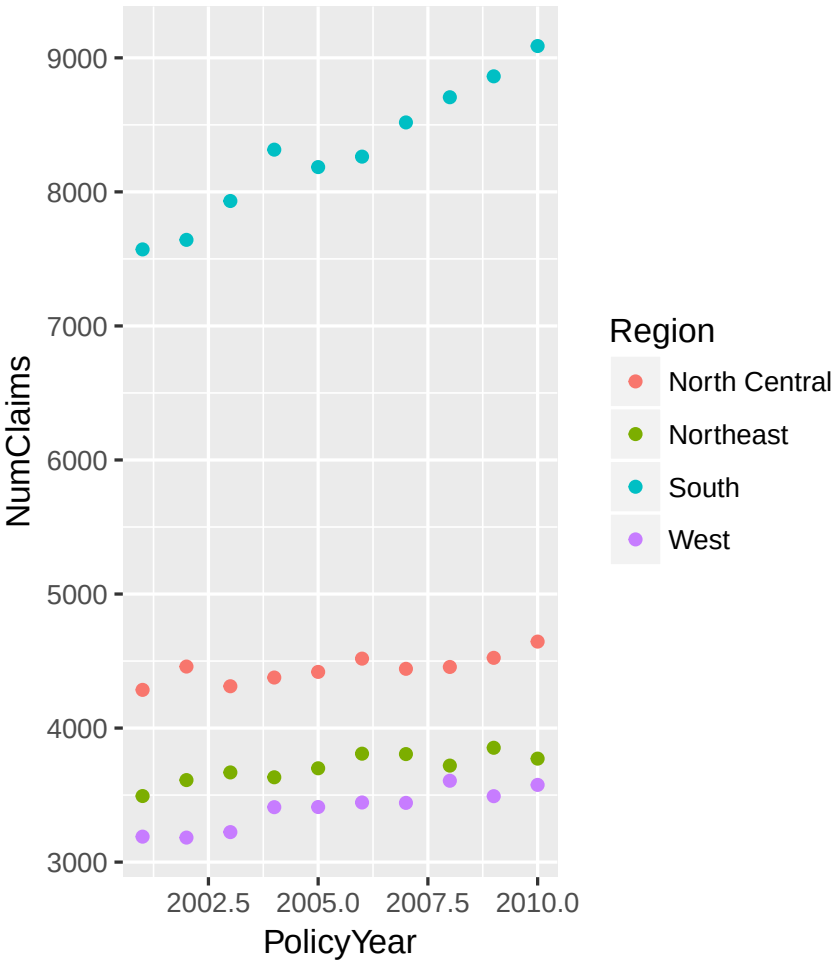
The `geom_*` functions add geometric shapes.

```
p <- basePlot + geom_line()
p
```



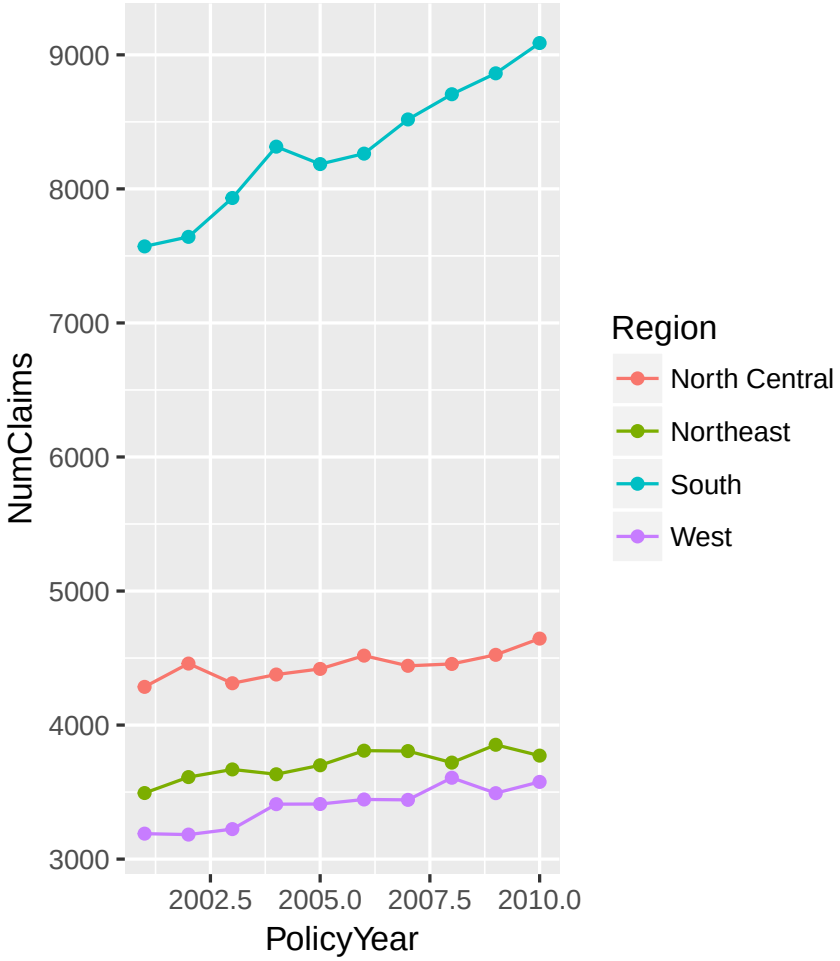
Adding layers

```
p <- basePlot + geom_point()
p
```



Nothing wrong with adding two layers

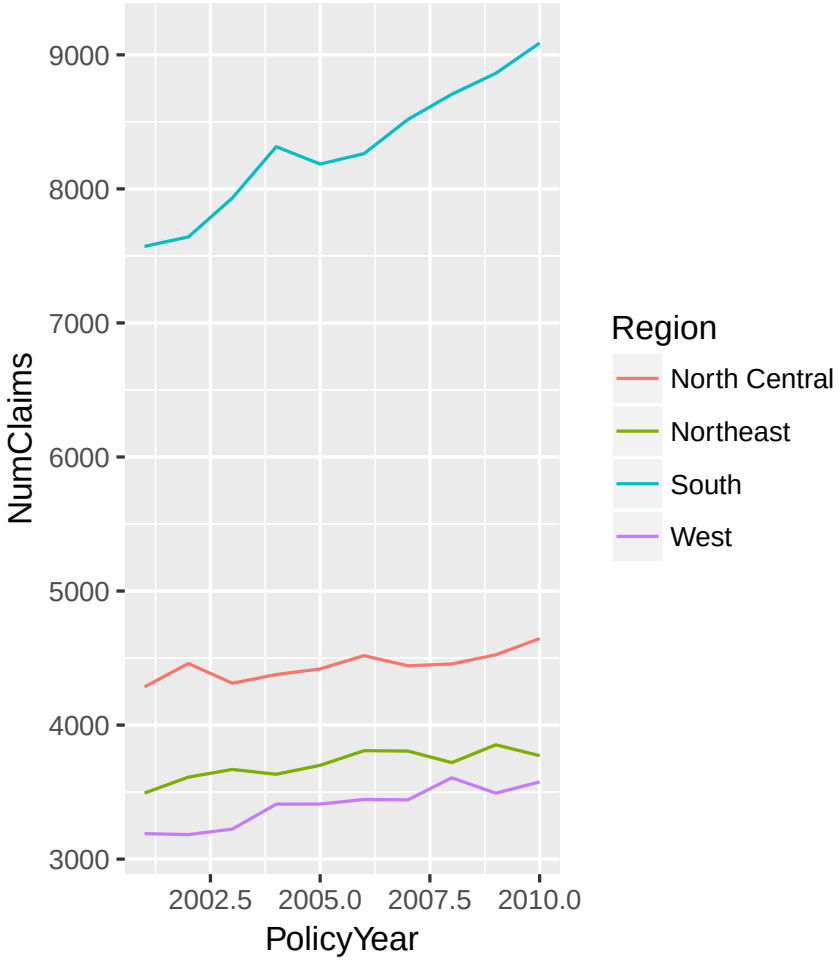
```
p <- basePlot + geom_point() + geom_line()
p
```



One step

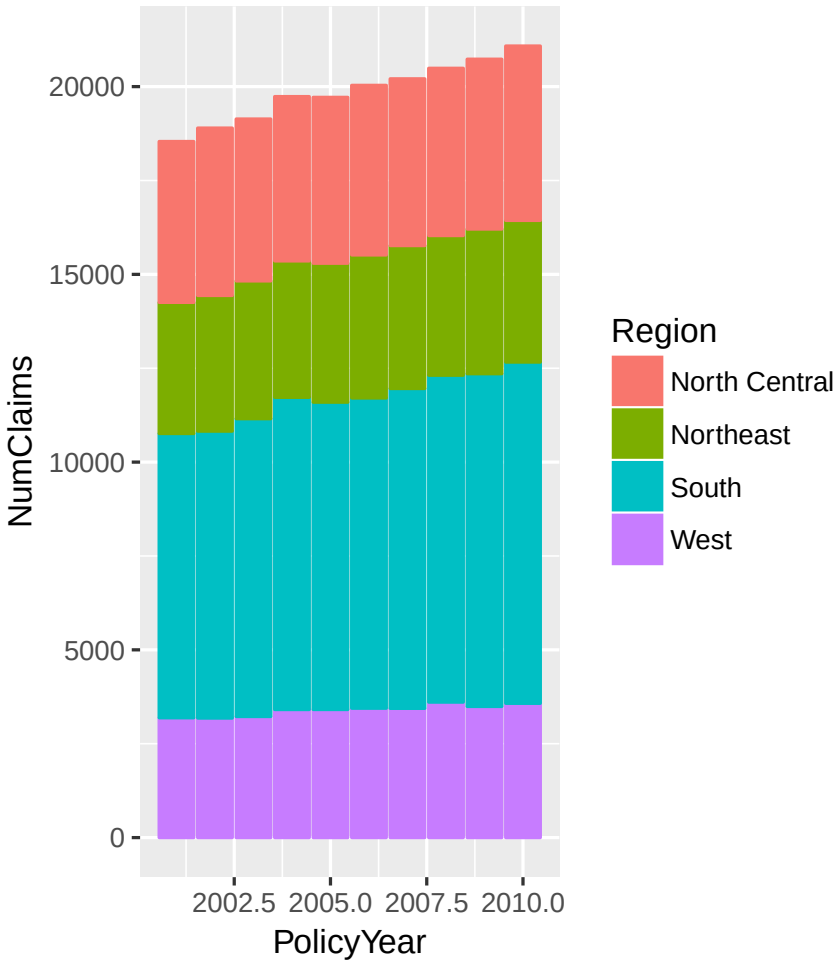
Typically we don't do this in steps.

```
p <- ggplot(RegionExperience, aes(x = PolicyYear,
    y = NumClaims, group = Region, color = Region)) +
    geom_line()
p
```



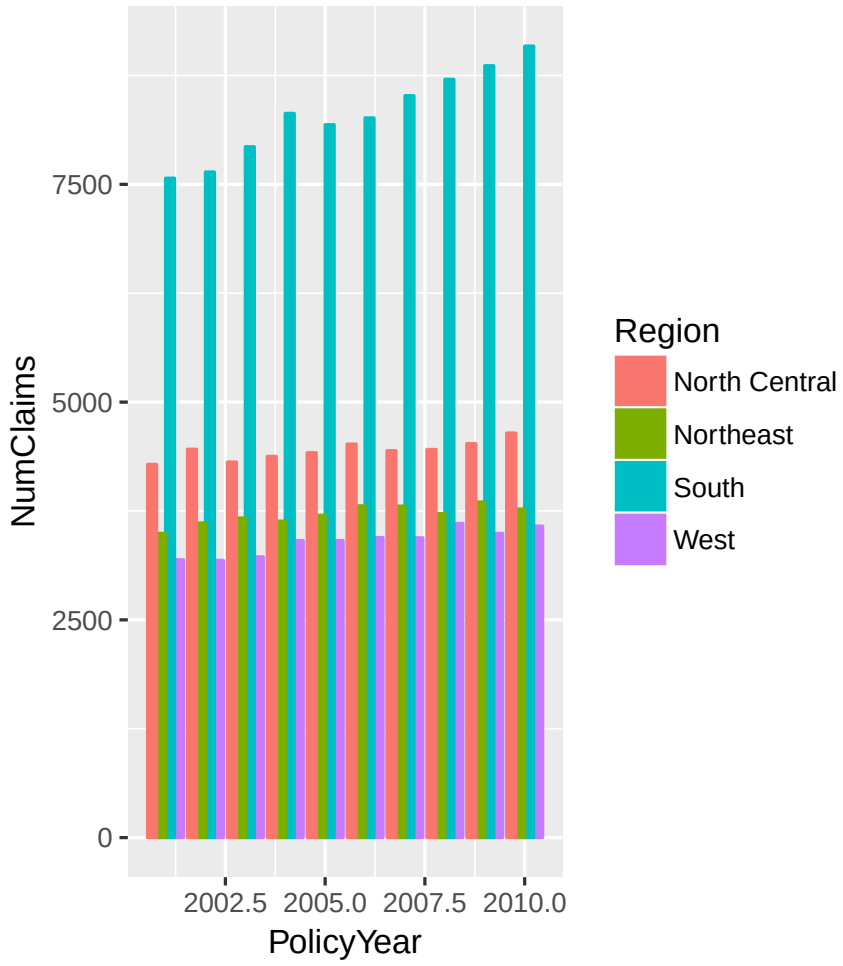
Layers have loads of parameters

```
p <- basePlot + geom_bar(stat = "identity", aes(fill = Region))
p
```



Mmm, parameters

```
p <- basePlot + geom_bar(stat = "identity", position = "dodge",
    aes(fill = Region))
p
```



Layer parameters

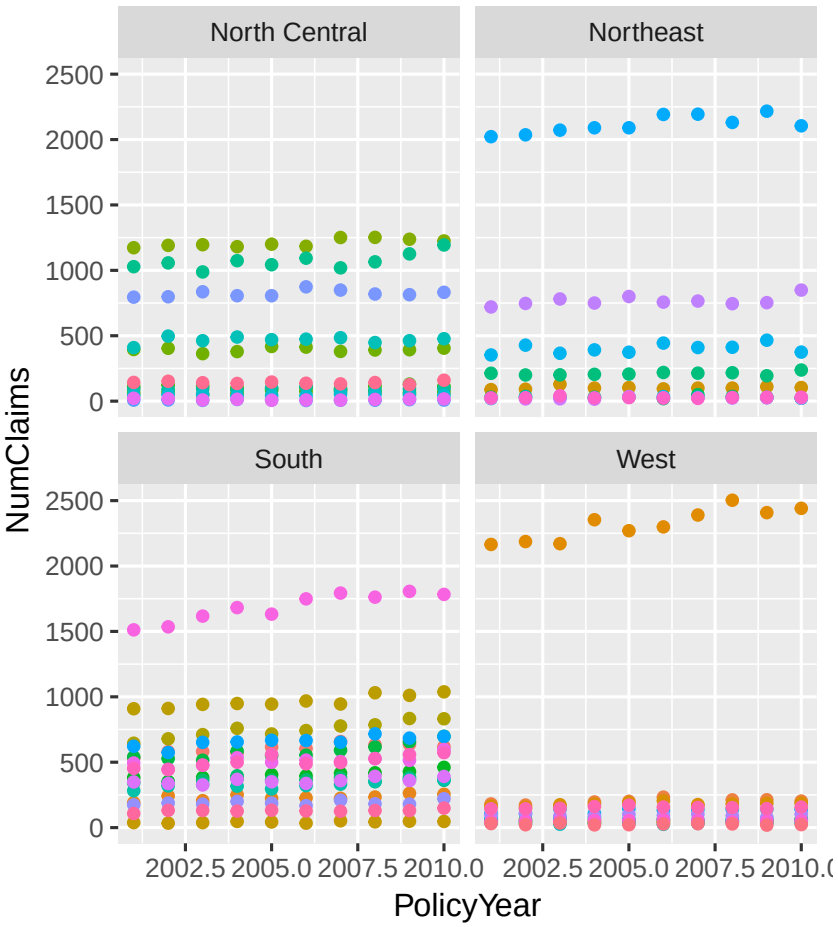
How do I know which parameters do what? RTFM

Help, google and stack overflow are your friends. Also, loads of books on the subject.

More features

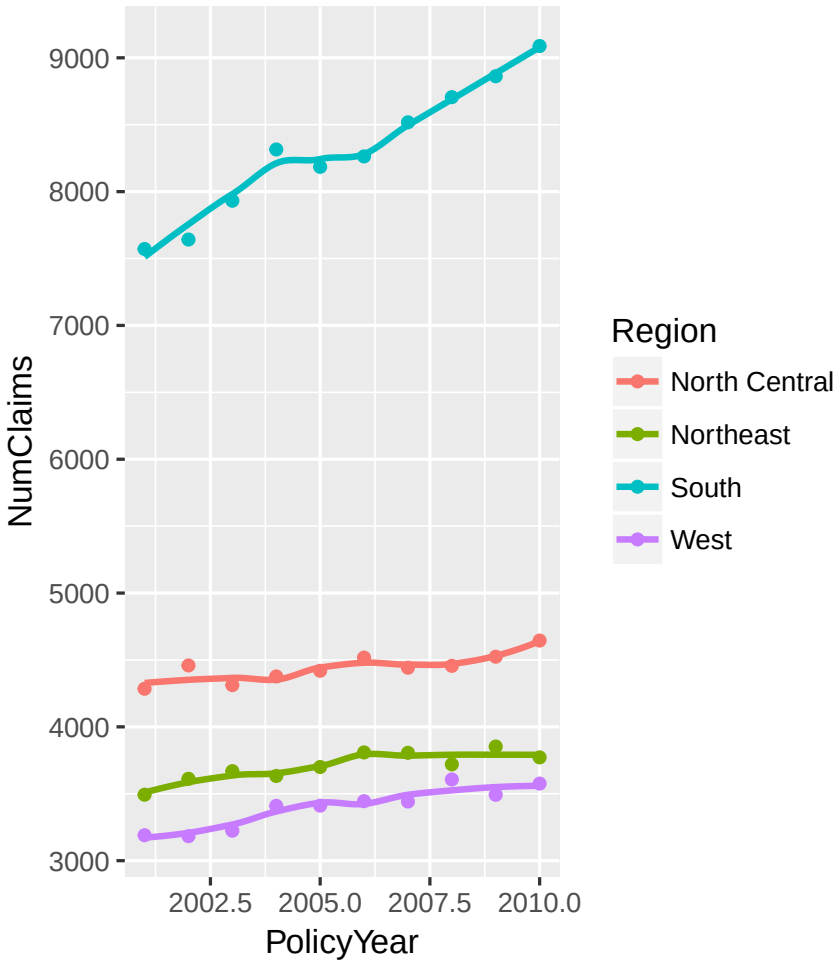
Facets

```
data(StateExperience)
p <- ggplot(StateExperience, aes(x = PolicyYear,
    y = NumClaims, color = State)) + geom_point() +
    facet_wrap(~Region)
p <- p + theme(legend.position = "none")
p
```



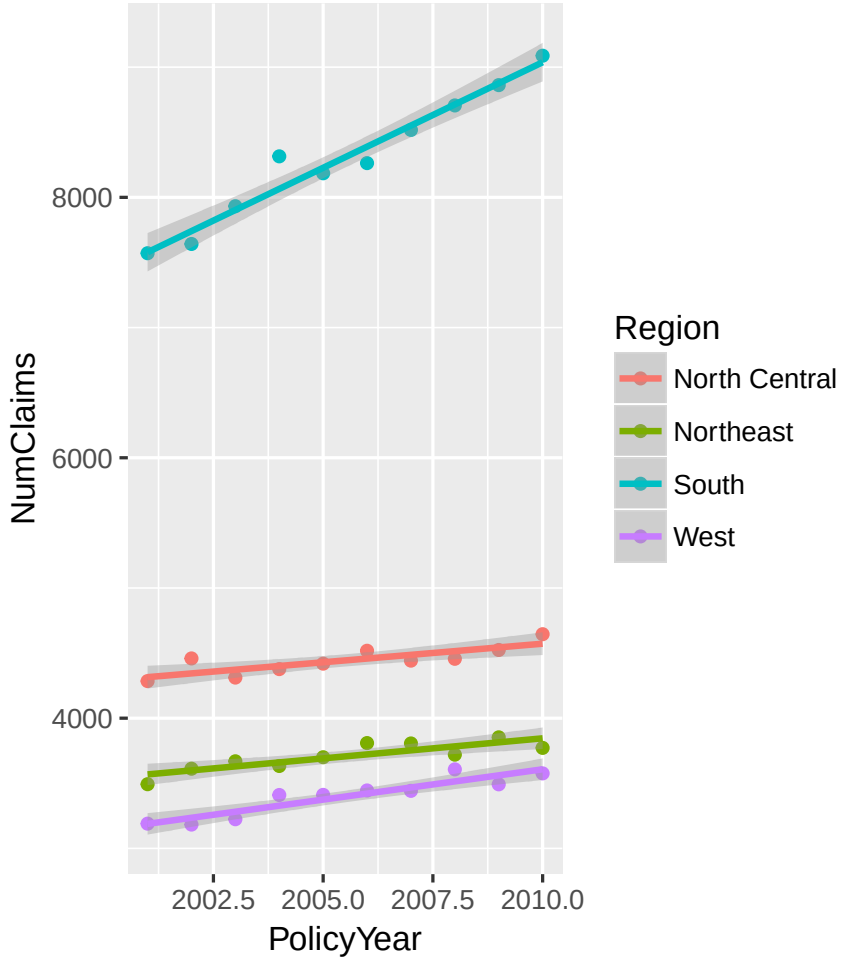
Statistical smoothers

```
p <- ggplot(RegionExperience, aes(x = PolicyYear,
    y = NumClaims, group = Region, color = Region)) +
    geom_point()
p + geom_smooth(se = FALSE)
```



Linear smoother

```
p + geom_smooth(method = lm)
```



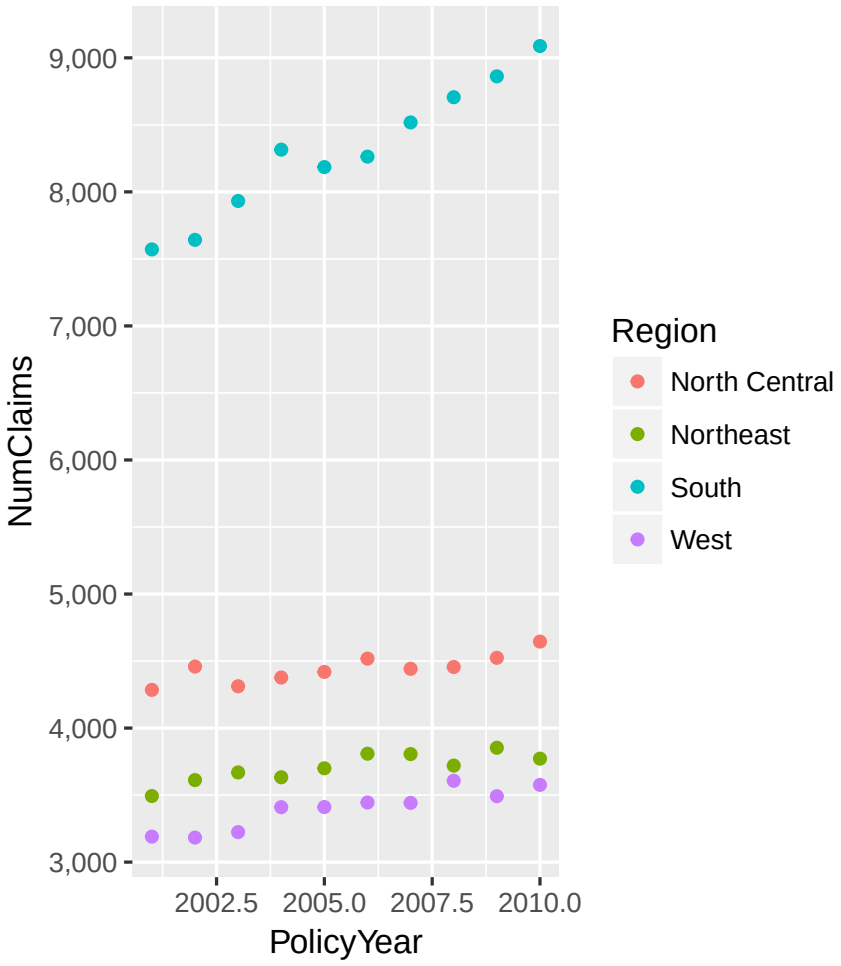
Scales

Scales

Scales control how things render on the plot. We must scale our numbers to the plotting device (typically a section of a computer screen.) We can also map things like color and axes to data values and control formatting.

Scales

```
p + scale_y_continuous(labels = scales::comma)
```



Scales

The use of scales is a *very* detailed topic, particularly when we start talking about color. Your specific problem will likely require a bit of research and experimentation. StackOverflow.com is your friend.

Non-data visual elements

Non-data elements are things like labels. Here's a sample of a few:

- `xlab()`, `ylab()` -> `plt + ylab("This is my y label") + xlab("Here is an x label")`
- `ggtitle()` -> `plt + ggtitle("Title of my plot")`
- `labs()` -> `plt + labs(x = "x-axis title", title = "My title")`
- `theme_bw()`, `theme_minimal()` -> `plt + theme_bw()`
- The `theme()` function gives complete control over all non-data related visual elements
- Check out the `ggthemes` package

Summary

Summary

- `ggplot2` is difficult at first, but will repay your investment.
- Works very well with grouped data to color/facet points.
- Fine-tuning things like axis labels can be a headache, but will get easier. Yes, Excel makes it easier to add data labels and change colors. `ggplot2` makes it easier to work with data.

Questions to ask when debugging

1. How am I mapping my data to an aesthetic?
2. How are these aesthetics handled by the geom that I'm using?
3.

Reference

- `http://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf`
- R Graphics Cookbook by Winston Chang
- `http://vita.had.co.nz/papers/layered-grammar.html`

Your turn

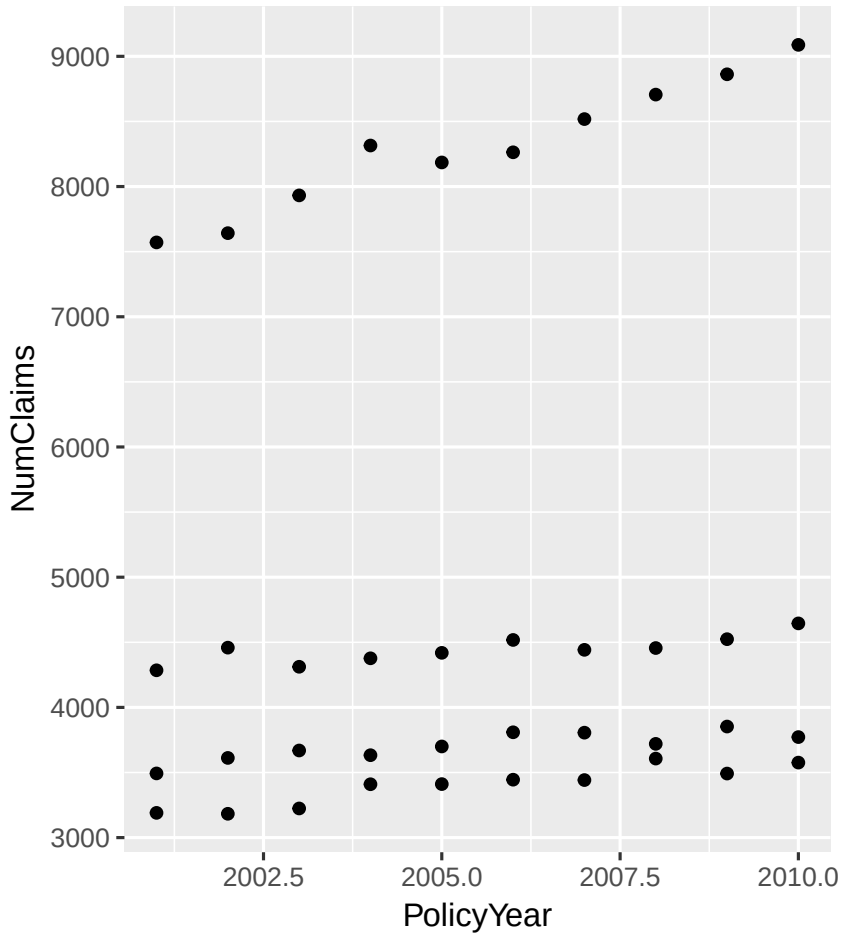
1. Create a scatter plot for policy year and number of claims
2. Color each point based on region
3. Add a linear smoother. Which region is showing the greatest increase in claims?
4. Form the policy frequency by taking the ratio of claims to policies. Plot this.

   Extra credit:

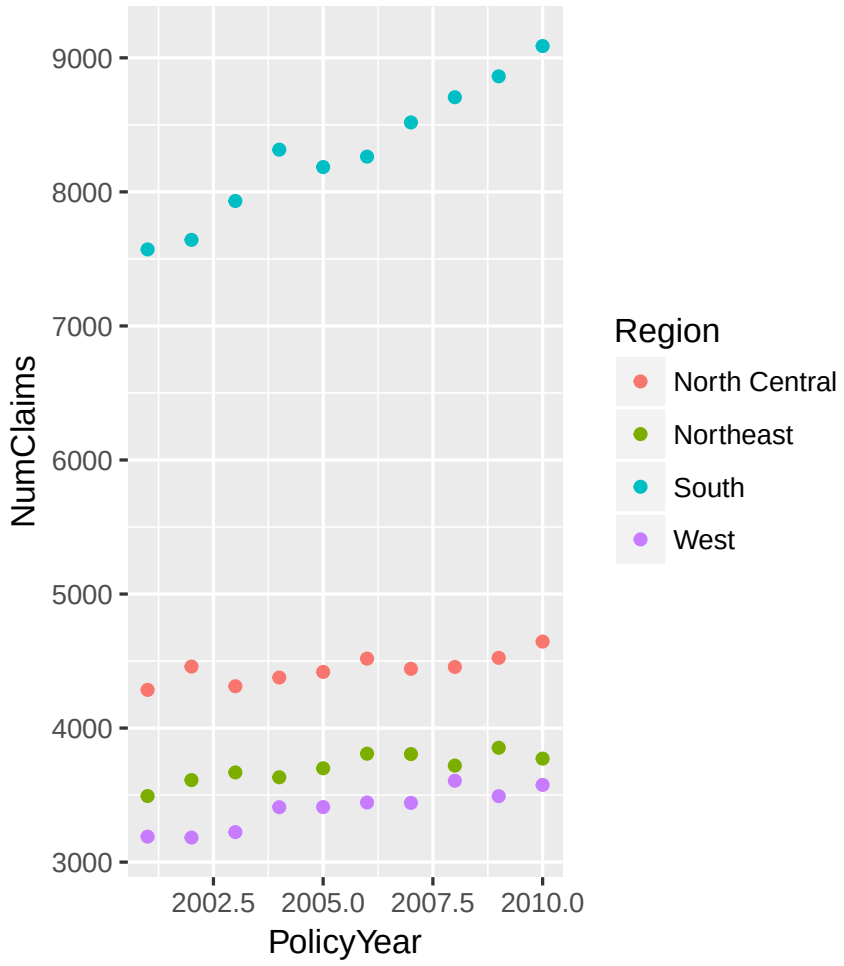
1. Use the state data to create a time series number of claims. Facet by region.

Answers

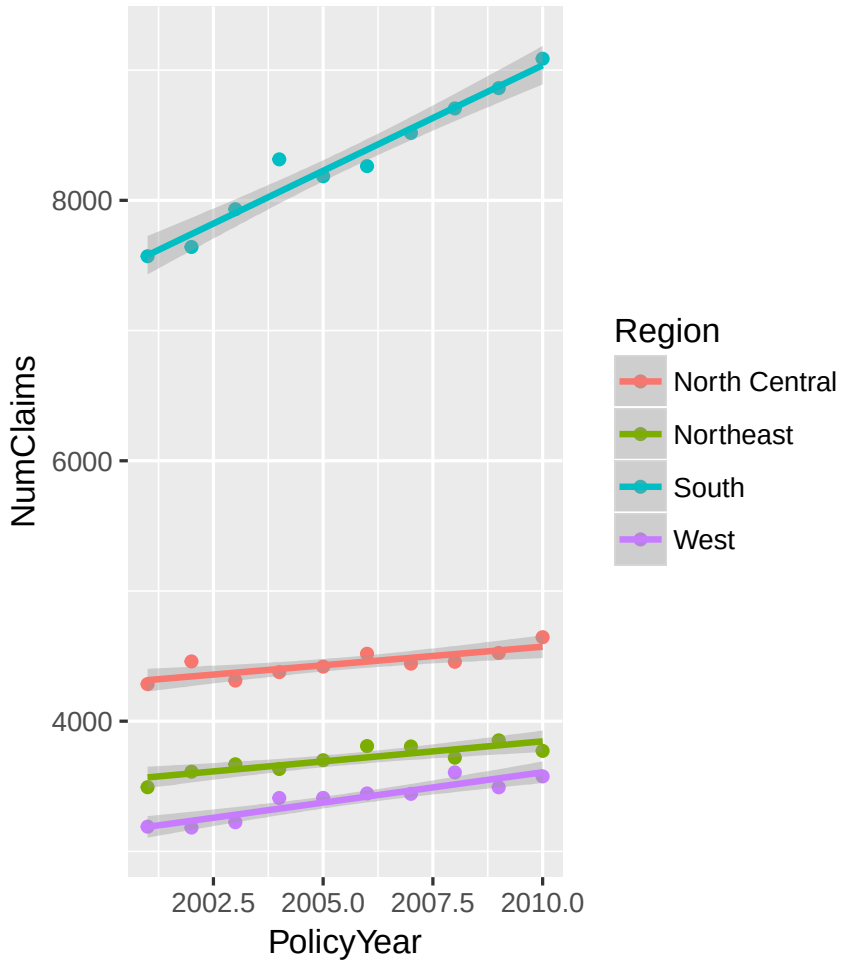
```
library(raw)
data("RegionExperience")
plt1 <- ggplot(RegionExperience, aes(x = PolicyYear,
    y = NumClaims)) + geom_point()
plt1
```




```
plt2 <- plt1 + aes(color = Region)
plt2
```

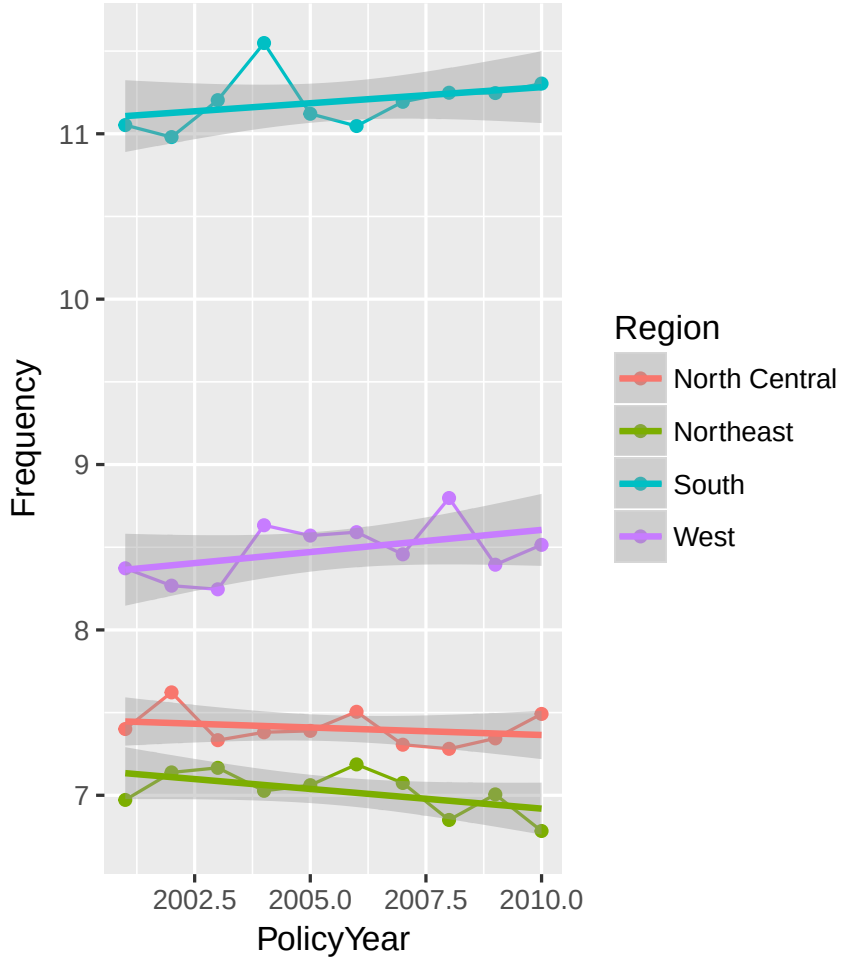



```
plt3 <- plt2 + stat_smooth(method = "lm")
plt3
```

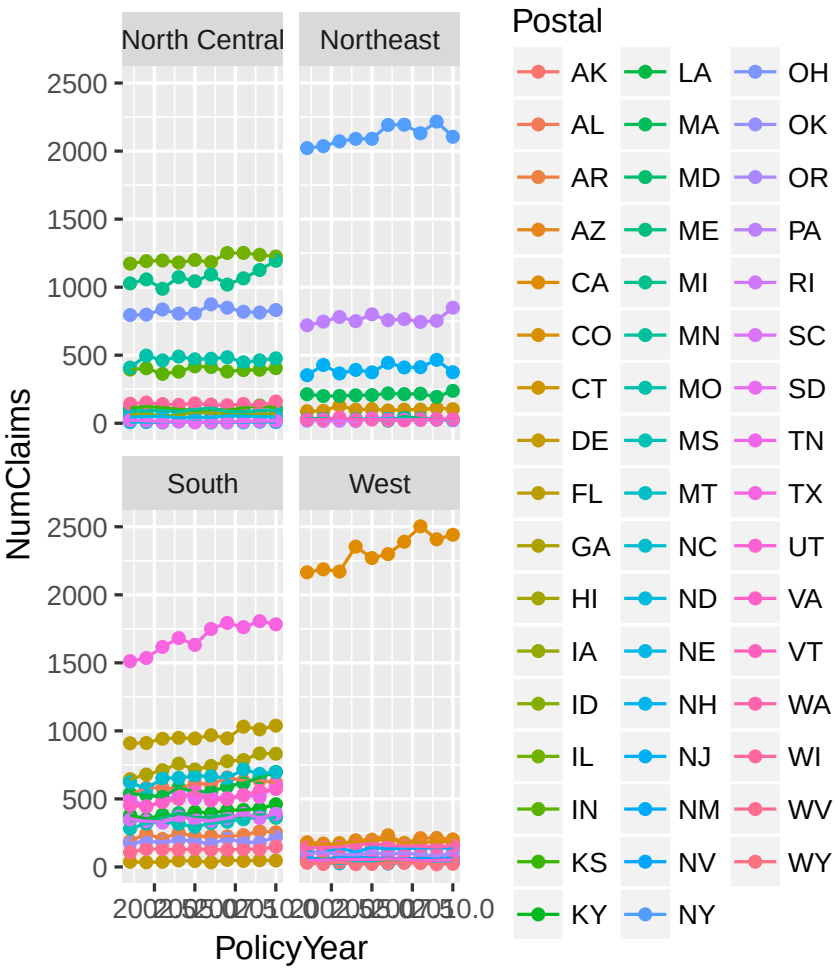



```
RegionExperience$Frequency <- with(RegionExperience,
    NumClaims/NumPolicies)

plt4 <- ggplot(RegionExperience, aes(x = PolicyYear,
    y = Frequency, color = Region)) + geom_point() +
    geom_line() + stat_smooth(method = lm)
plt4
```




```
data("StateExperience")
pltExtra <- ggplot(StateExperience, aes(x = PolicyYear,
    y = NumClaims, color = Postal)) + geom_point() +
    geom_line()
pltExtra + facet_wrap(~Region)
```


North Central
Northeast
South
West
NumClaims
2500
2000
1500
1000
500
0
PolicyYear
Postal
AK
AL
AR
AZ
CA
CO
CT
DE
FL
GA
HI
IA
ID
IL
IN
KS
KY
LA
MA
MD
ME
MI
MN
MO
MS
MT
NC
ND
NE
NH
NJ
NM
NV
NY
OH
OK
OR
PA
RI
SC
SD
TN
TX
UT
VA
VT
WA
WI
WV
WY