

# ggplot2: A Philistine's Guide

Mick Cooney  
michael.cooney@applied.ai

23 March 2016

# Structure of Talk

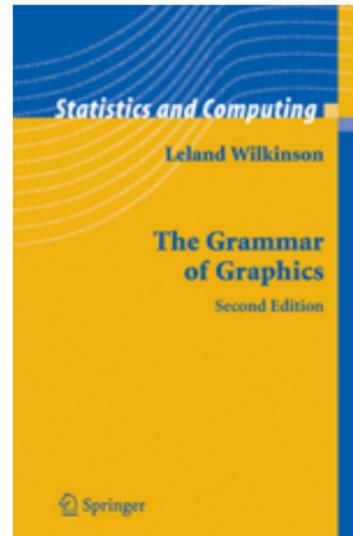
- The Grammar of Graphics
- Getting Started with qplot()
- Creating Plots
- Real-World Examples
- Summary

# The Grammar of Graphics

Leland Wilkinson, "The Grammar of Graphics" 2005

Plots/Graphics consist of:

- Data
- Layers
- Scales
- Coords
- Facets
- Themes



Descriptive not prescriptive

Static only, non-interactive

For interactive plots, try:

- ggviz
- shiny
- ggobi

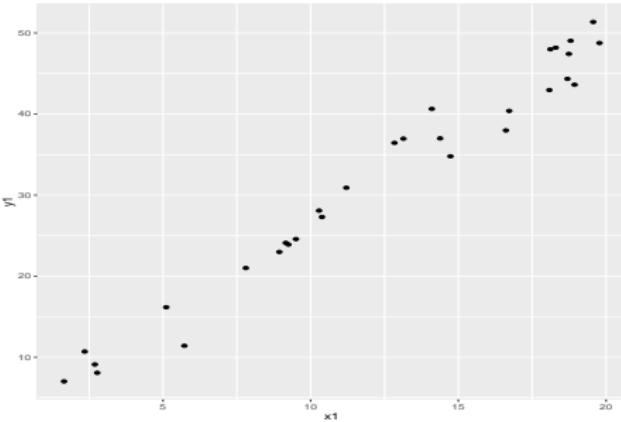
# Starting with qplot()

qplot() convenient (though also crutch)

Scatterplot of linear relationship

```
x1 <- runif(30, 0, 20)
y1 <- 2.4 * x1 + 3 + rnorm(30, 0, 2)

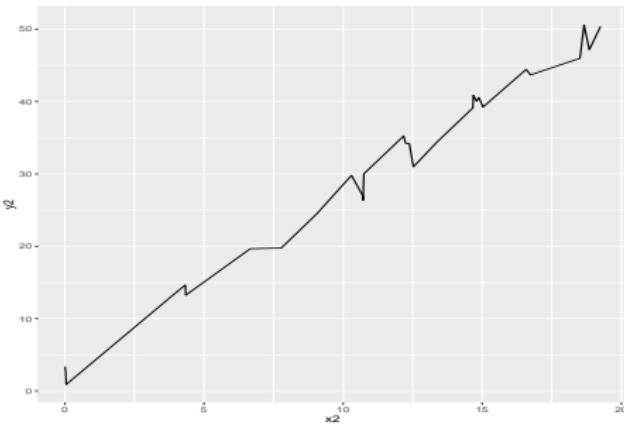
pv <- qplot(x1, y1, geom = 'point')
```



## Lineplot of linear relationship

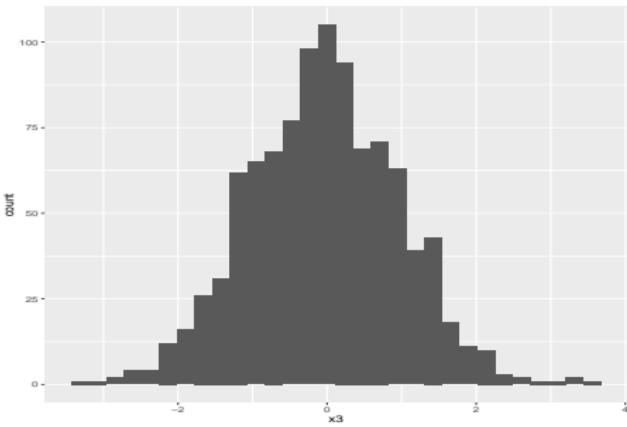
```
x2 <- runif(30, 0, 20)
y2 <- 2.4 * x2 + 3 + rnorm(30, 0, 2)

pv <- qplot(x2, y2, geom = 'line')
```



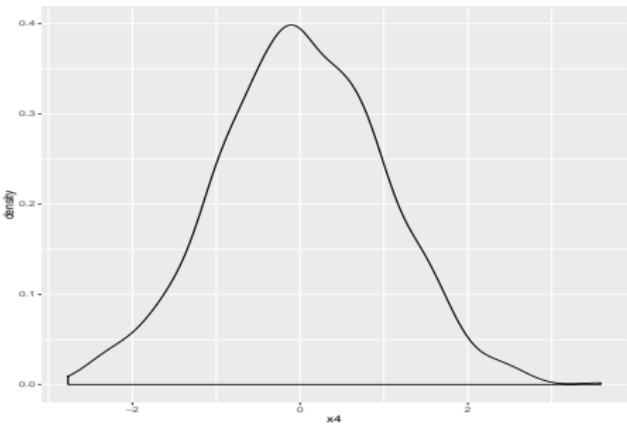
## Univariate data supported

```
x3 <- rnorm(1000, 0, 1)  
pv <- qplot(x3, geom = 'histogram')
```



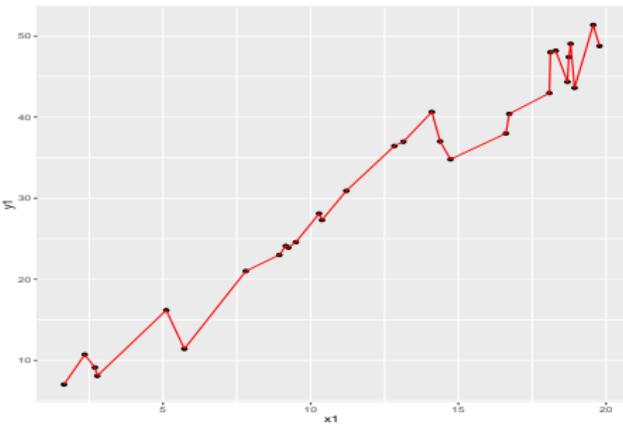
## Kernel density estimation

```
x4 <- rnorm(1000, 0, 1)  
p4 <- qplot(x4, geom = 'density')
```



## Can combine geoms

```
pv <- qplot(x1, y1, geom = 'point') +  
  geom_line(colour = 'red')
```



# Creating Plots

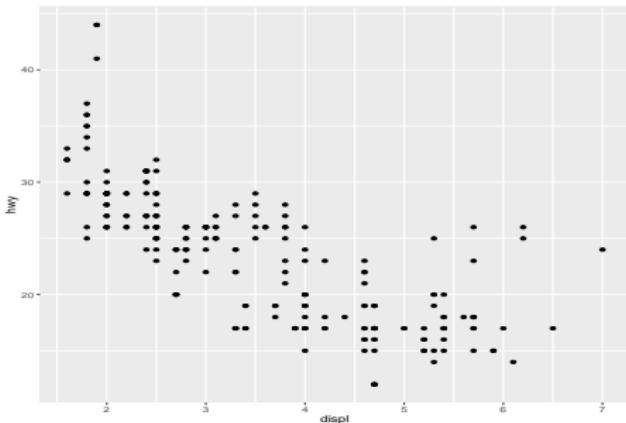
Use well-known `mtcars` dataset

```
head(mtcars)

##          mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4     21.0   6 160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag 21.0   6 160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710    22.8   4 108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360 175 3.15 3.440 17.02  0  0    3    2
## Valiant       18.1   6 225 105 2.76 3.460 20.22  1  0    3    1
```

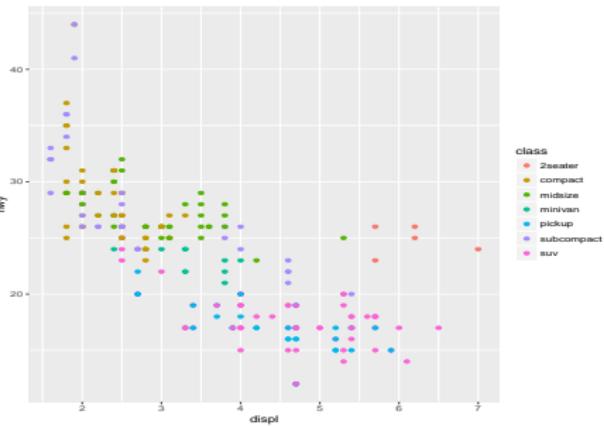
## First scatterplot of engine displacement vs highway mileage

```
pv <- ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point()
```



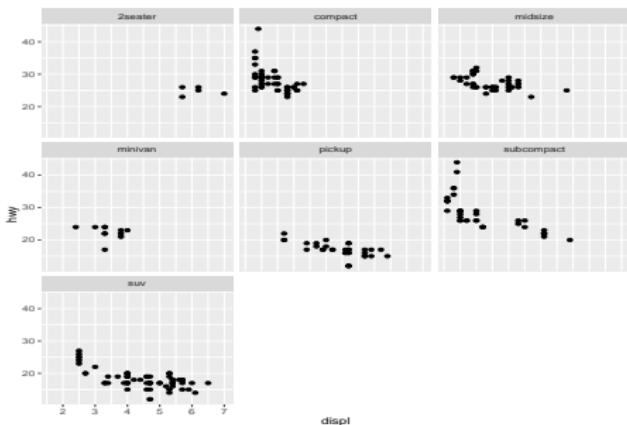
## Add colour for car class

```
pv <- ggplot(mpg, aes(x = displ, y = hwy, colour = class))  
geom_point()
```



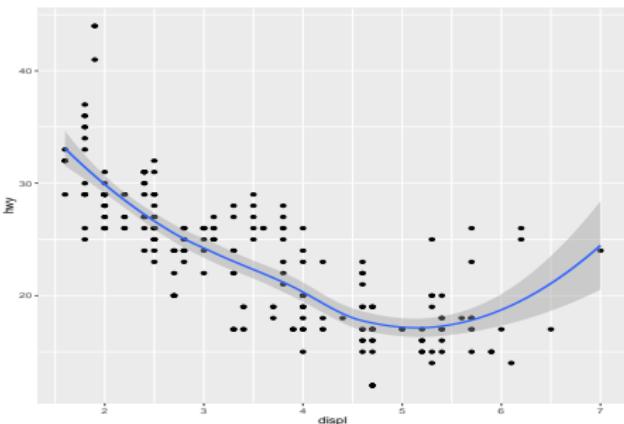
Instead of colour for class, use facetting

```
pv <- ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  facet_wrap(~class)
```



## Add stats (loess smoother)

```
pv <- ggplot(mpg, aes(x = displ, y = hwy)) +  
  geom_point() +  
  geom_smooth()
```



# Real-World Examples

- R Workshop: Comparing Multiple Probability Distributions
- Quant Finance: Unconditional (Long-term) Volatility
- Insurance: Modeling Loss Curves
- Quant Finance: Bootstrapped Vega-Neutral Portfolio
- Insurance/Quant Finance: Modelling Retirement Portfolios
- R Workshop: Gaussian Processes with Data Uncertainty
- Quant Finance: Assessing Equity and Equity Option Strategies

# Comparing Multiple Probability Distributions

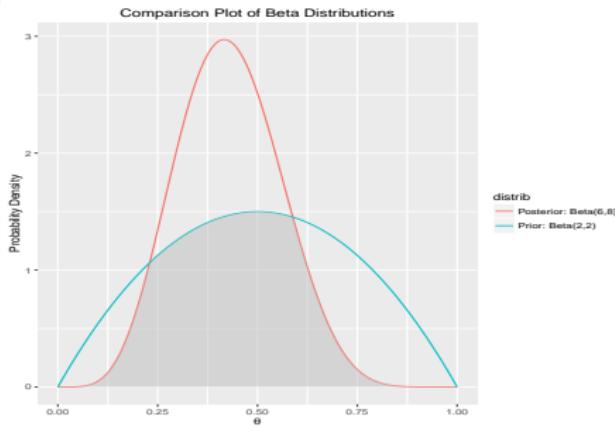
Prior: Beta(2, 2), Posterior: Beta(6, 8)

Data: 4H, 6T

```
head(plot_dt, 3)

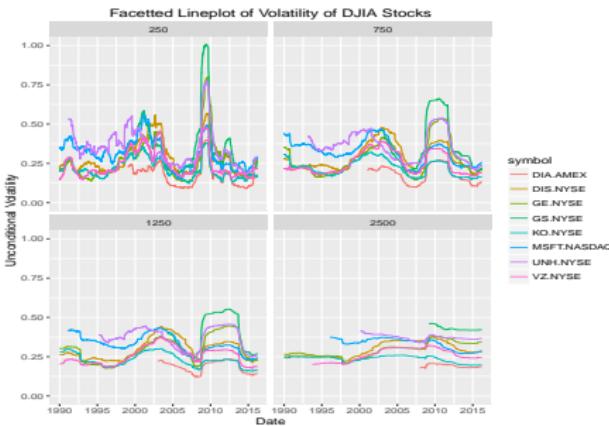
##      x     dens      distrib
## 1: 0.000 0.000000 Prior: Beta(2,2)
## 2: 0.001 0.005994 Prior: Beta(2,2)
## 3: 0.002 0.011976 Prior: Beta(2,2)

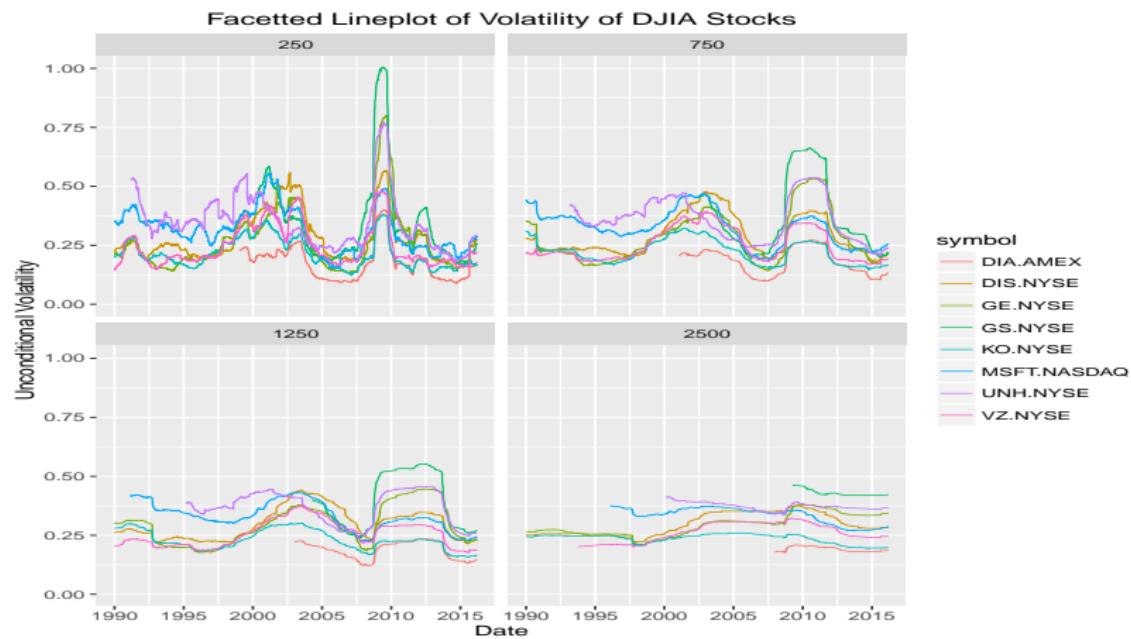
pv <- ggplot() +
  geom_line(aes(x = x, y = dens, colour = distrib),
            data = plot_dt) +
  geom_line() +
  geom_area(aes(x = theta_seq, y = minval),
            data = plot_dt[, .(minval = min(dens)),
                           by = x],
            fill = 'grey', alpha = 0.5) +
  xlab(expression(theta)) +
  ylab("Probability Density") +
  ggtitle("Comparison Plot of Beta Distributions")
```



# Unconditional (Long-term) Volatility

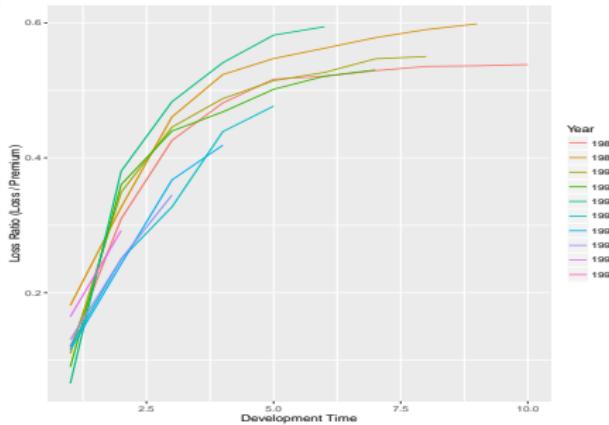
```
pv <- ggplot(aes(x = date, y = histvol, colour = symbol  
,data = longterm_dt) +  
facet_wrap(~horizon) +  
geom_line(size = 0.3) +  
expand_limits(y = 0) +  
xlab("Date") +  
ylab("Unconditional Volatility") +  
ggtitle("Facetted Lineplot of Volatility of DJIA St
```

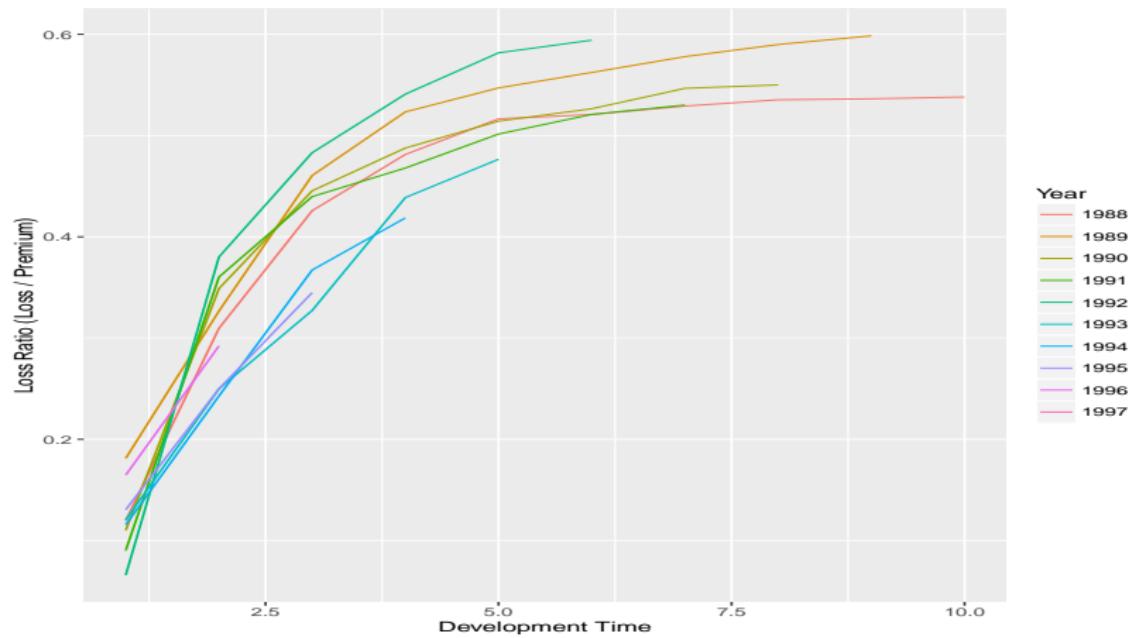




# Modeling Loss Curves

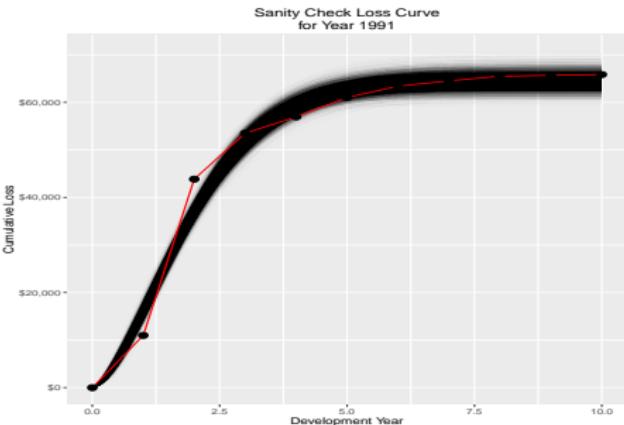
```
pv <- ggplot(aes(x = DevelopmentLag  
                  ,y = CumPaidLoss / EarnedPremDIR  
                  ,colour = as.character(AccidentYear))  
                  ,data = losscurves_dt) +  
  geom_line() +  
  guides(colour = guide_legend(title = "Year")) +  
  xlab("Development Time") +  
  ylab("Loss Ratio (Loss / Premium)")  
  
head(sanity_1991_data_dt, 3)  
  
##      DevLag      CPL  
## 1:      0       0  
## 2:      1 10985  
## 3:      2 43846  
  
head(sanity_1991_samp_df, 3)  
  
##    Var1 iterations     value t.data  
## 1    1          1 0.00000 0.0  
## 2    2          1 427.566  0.1  
## 3    3          1 1291.356  0.2
```





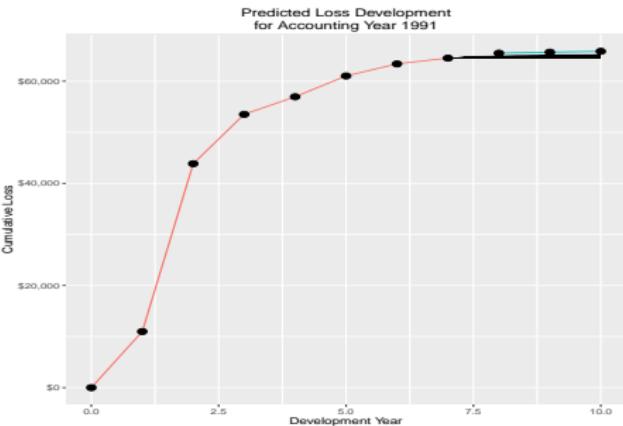
## Output of Model for Accounting Year 1991

```
pv <- ggplot() +  
  geom_line (aes(x = t.data, y = value  
                 ,group = iterations)  
             ,data = sanity_1991_samp_df  
             ,alpha = 0.01, colour = 'black') +  
  geom_line (aes(x = DevLag, y = CPL)  
             ,data = sanity_1991_data_dt  
             ,colour = 'red') +  
  geom_point(aes(x = DevLag, y = CPL)  
             ,data = sanity_1991_data_dt, size = 3) +  
  xlab("Development Year") +  
  ylab("Cumulative Loss") +  
  ggtitle("Sanity Check Loss Curve\\nfor Year 1991") +  
  scale_y_continuous(label = dollar) +  
  expand_limits(y = 0);
```



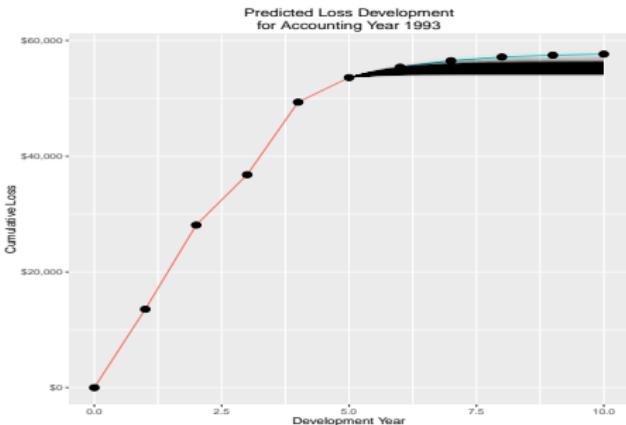
## Predict Development for Accounting Year 1991

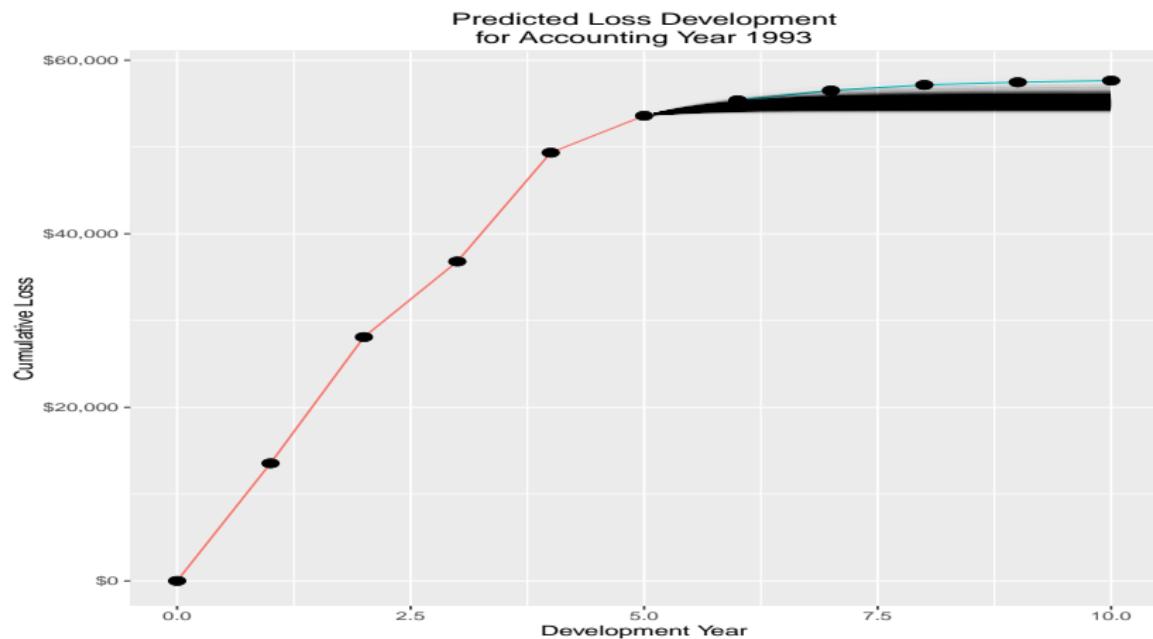
```
pv <- ggplot() +  
  geom_line (aes(x = DevLag, y = CPL  
                ,colour = datatype)  
             ,data = predict_1991_data_dt) +  
  geom_point(aes(x = DevLag, y = CPL)  
             ,data = predict_1991_data_dt, size = 3) +  
  geom_line (aes(x = t.data, y = value  
                ,group = iterations)  
             ,data = predict_1991_samp_df, alpha = 0.05)  
  xlab("Development Year") +  
  ylab("Cumulative Loss") +  
  ggtitle("Predicted Loss Development\\nfor Accounting Year 1991") +  
  scale_y_continuous(label = dollar) +  
  theme(legend.position = 'none') +  
  expand_limits(y = 0);
```



## Predict Development for Accounting Year 1993

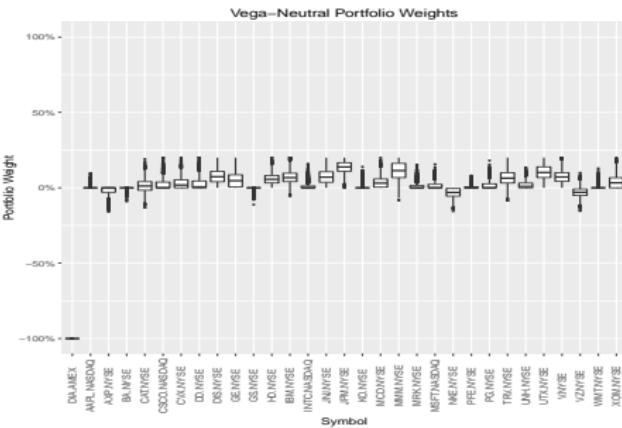
```
pv <- ggplot() +  
  geom_line (aes(x = DevLag, y = CPL  
                ,colour = datatype)  
             ,data = predict_1993_data_dt) +  
  geom_point(aes(x = DevLag, y = CPL)  
             ,data = predict_1993_data_dt  
             ,size = 3) +  
  geom_line (aes(x = t.data, y = value  
                ,group = iterations)  
             ,data = predict_1993_samp_df  
             ,alpha = 0.01) +  
  xlab("Development Year") +  
  ylab("Cumulative Loss") +  
  ggtitle("Predicted Loss Development\\nfor Accounting Year 1993") +  
  scale_y_continuous(label = dollar) +  
  theme(legend.position = 'none') +  
  expand_limits(y = 0);
```

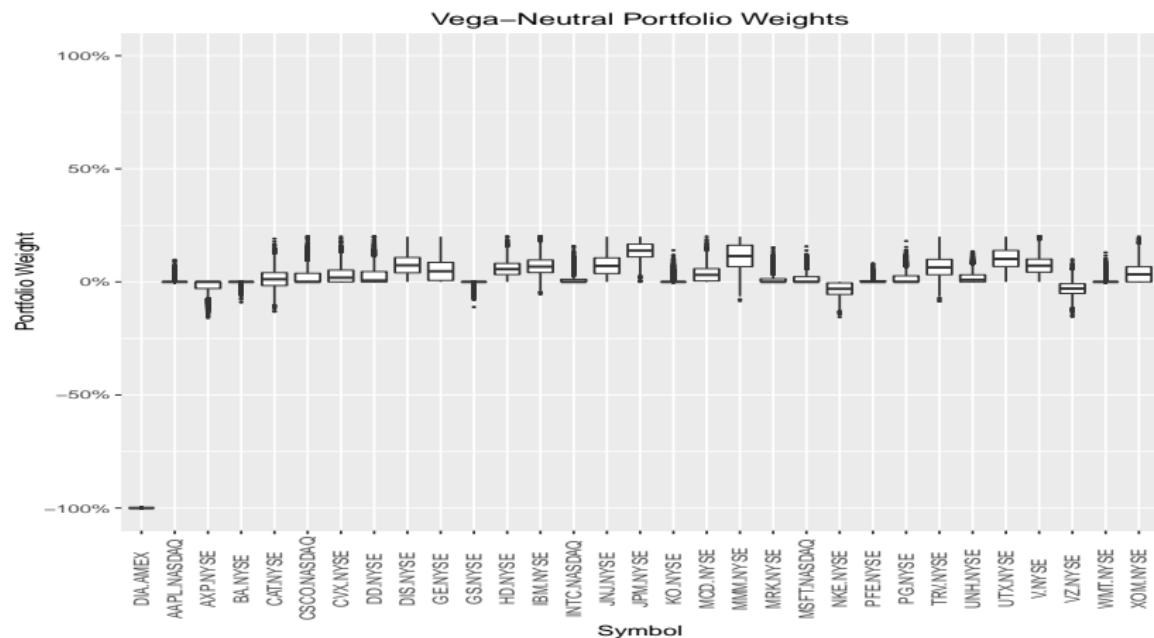




# Bootstrapped Vega-Neutral Portfolio

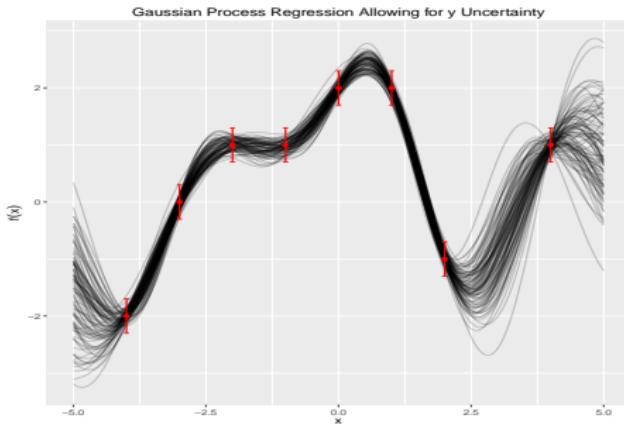
```
pv <- ggplot(aes(Var2, value)
              ,data = melt(weight_matrix)) +
  geom_boxplot(outlier.size = 0.25) +
  scale_y_continuous(labels = percent
                      ,limits = c(-1, 1)) +
  theme(axis.text.x = element_text(angle = 90
                                    ,vjust = 0.5)) +
  xlab("Symbol") +
  ylab("Portfolio Weight") +
  ggtitle("Vega-Neutral Portfolio Weights")
```

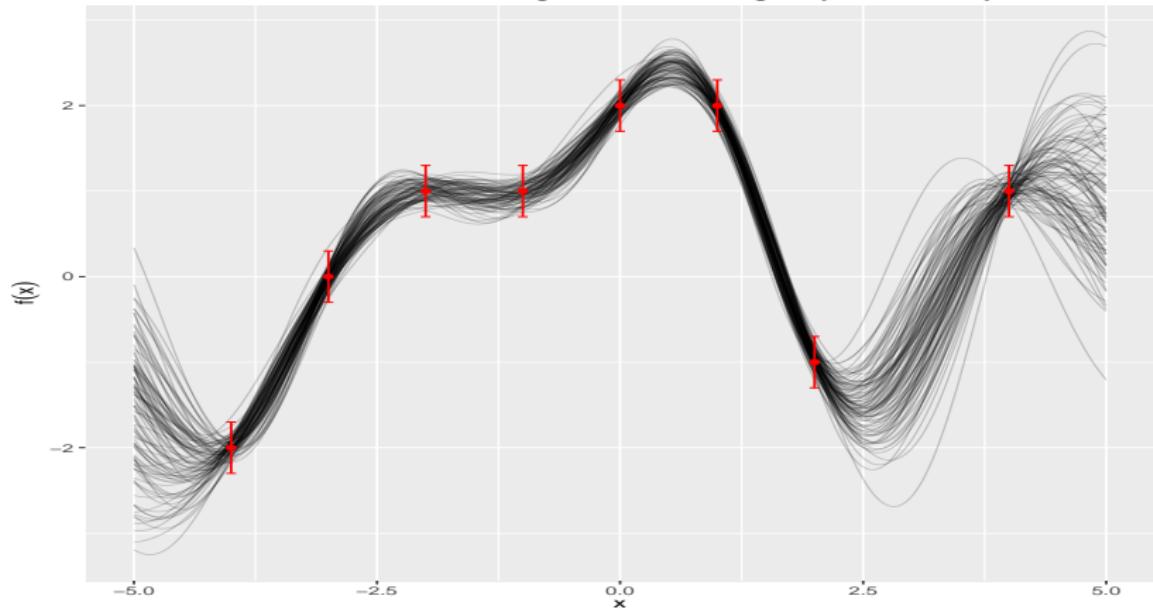




# Gaussian Processes with Data Uncertainty

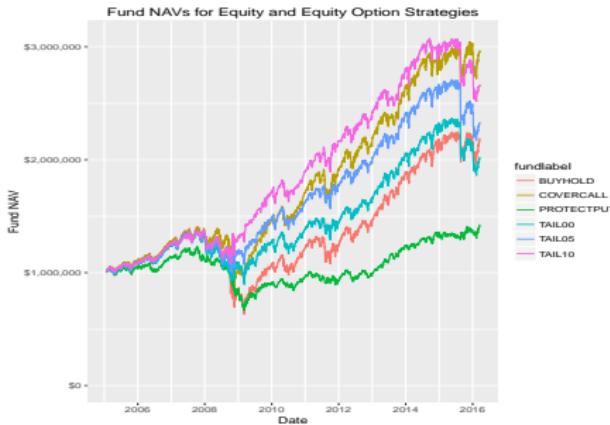
```
pv <- ggplot() +  
  geom_line(aes(x = x, y = value, group = Var1),  
            data = gpplot_dt  
            ,size = 0.3, alpha = 0.2) +  
  geom_point(aes(x = x, y = y),  
             data = gpdata_dt, colour = 'red') +  
  geom_errorbar(aes(x, ymin = ymin, ymax = ymax),  
                data = gpdata_dt  
                ,colour = 'red', width = 0.1) +  
  xlab(expression(x)) +  
  ylab(expression(f(x))) +  
  ggtitle("Gaussian Process Regression Allowing for y  
          Uncertainty")
```

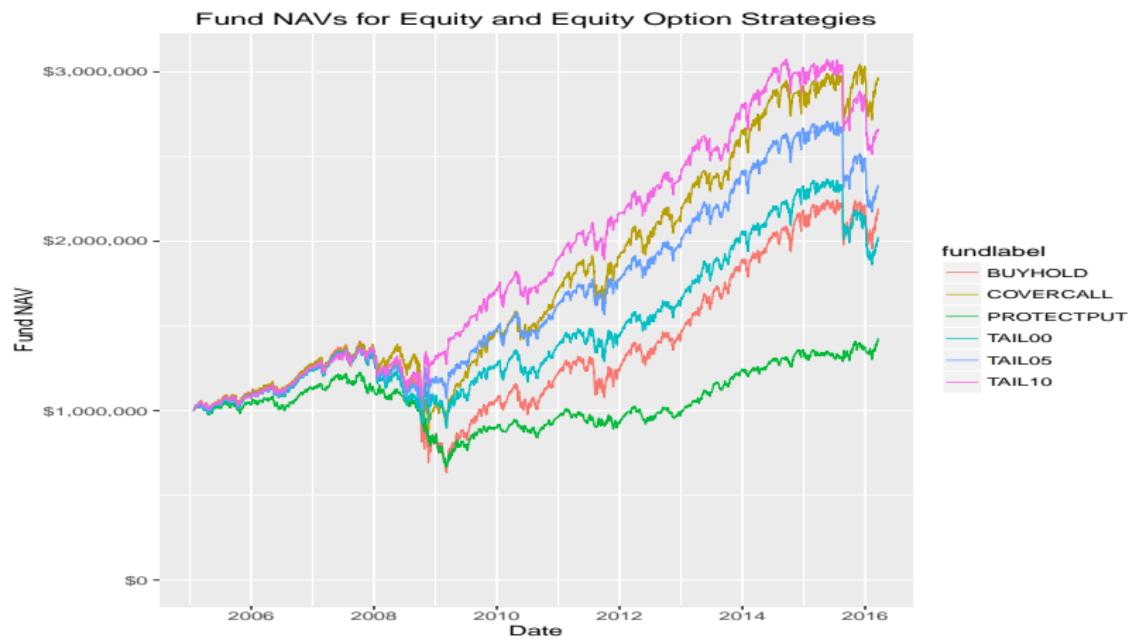


**Gaussian Process Regression Allowing for y Uncertainty**

# Assessing Equity and Equity Option Strategies

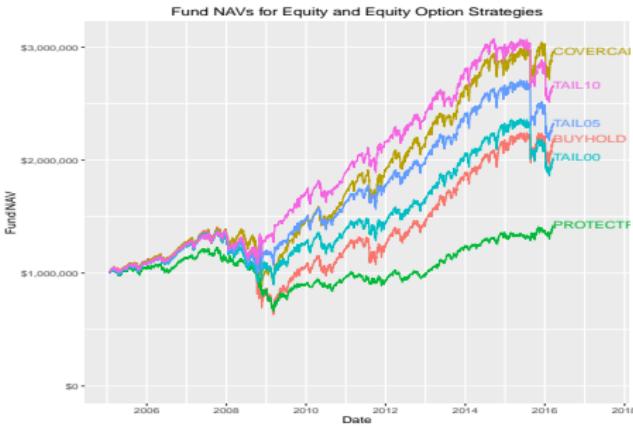
```
pv <- ggplot() +  
  geom_line(aes(x = date, y = nav  
               , colour = fundlabel)  
            , data = fundnav_dt  
            , size = 0.3) +  
  scale_y_continuous(labels = dollar) +  
  expand_limits(y = 0) +  
  xlab("Date") +  
  ylab("Fund NAV") +  
  ggtitle("Fund NAVs for Equity and Equity Option Strate  
gy Comparison")
```

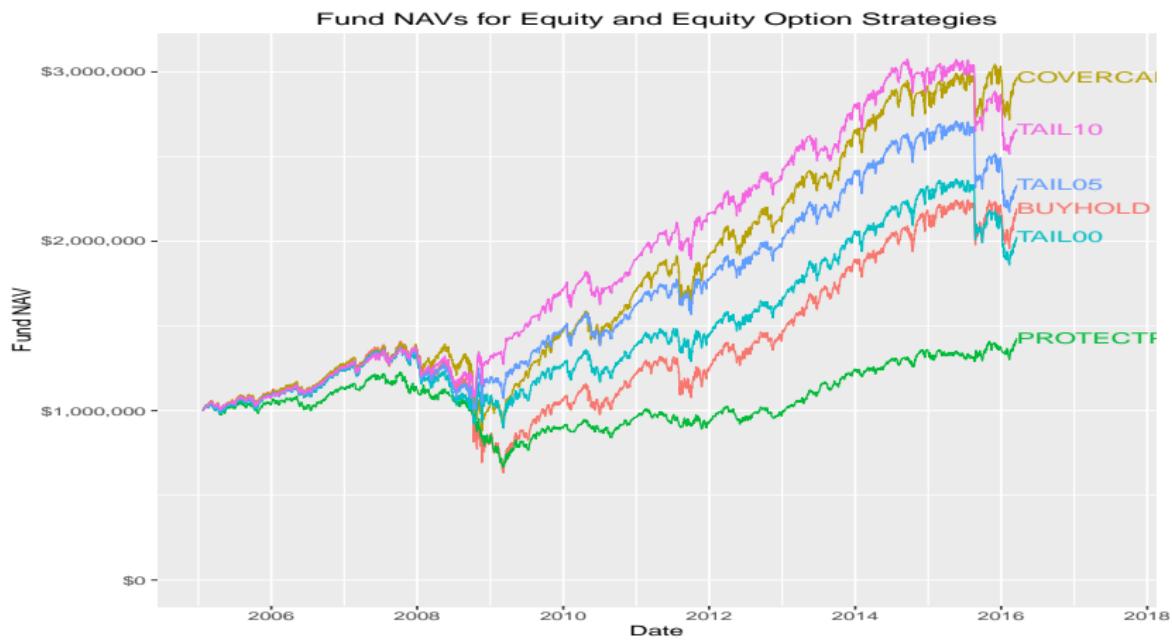




## directlabels rather than legends

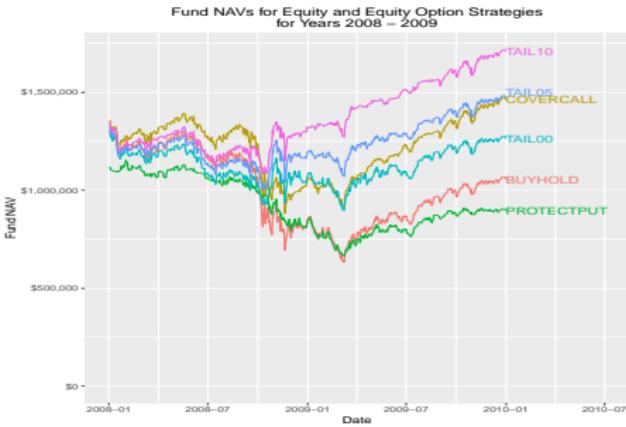
```
pv <- ggplot(data = fundnav_dt) +  
  geom_line(aes(x = date, y = nav  
                ,colour = fundlabel)  
            ,size = 0.3) +  
  scale_y_continuous(labels = dollar) +  
  theme(legend.position = 'none') +  
  geom_dl(aes(x = date, y = nav  
                ,colour = fundlabel  
                ,label = fundlabel)  
          ,method = list('last.bumpup')) +  
  expand_limits(x = as.Date('2017-07-01')  
                ,y = 0) +  
  xlab("Date") +  
  ylab("Fund NAV") +  
  ggtitle("Fund NAVs for Equity and Equity Option Stri
```

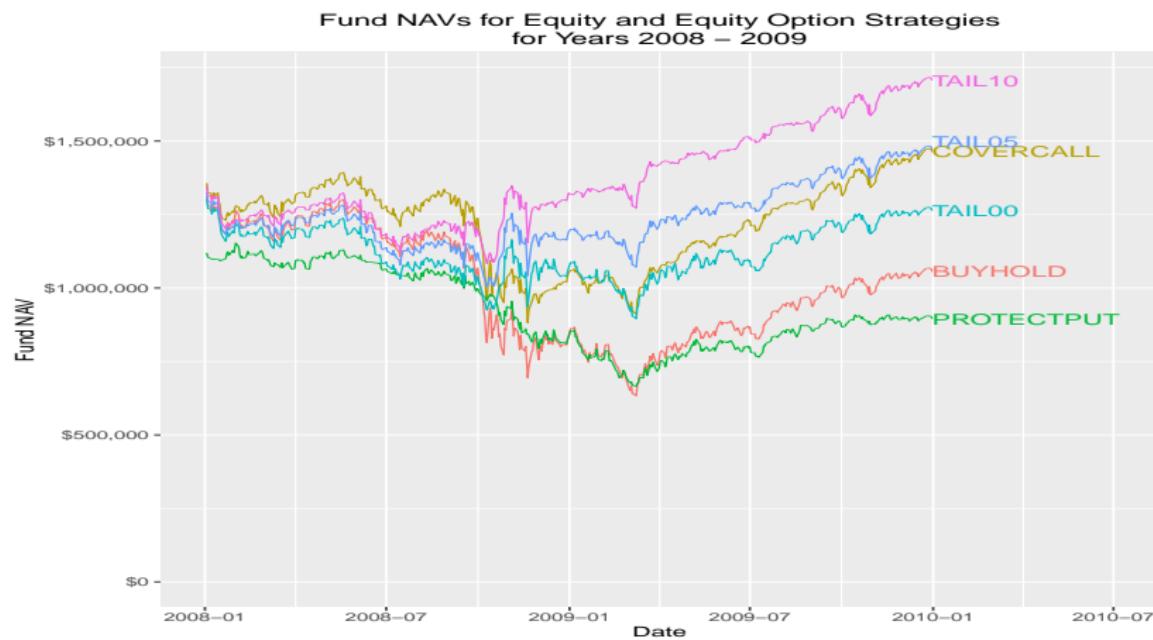




## Check 2008—2009

```
pv <- ggplot(data = fundnav_dt[date >= as.Date('2008-01-01') &
  date <= as.Date('2010-01-01'),
  geom_line(aes(x = date, y = nav,
    colour = fundlabel),
    size = 0.3) +
  scale_y_continuous(labels = dollar) +
  theme(legend.position = 'none') +
  geom_dl(aes(x = date, y = nav,
    colour = fundlabel,
    label = fundlabel),
    method = list('last.bumpup')) +
  expand_limits(x = as.Date('2010-07-01'),
    ,y = 0) +
  xlab("Date") +
  ylab("Fund NAV") +
  ggtitle("Fund NAVs for Equity and Equity Option Strategies for Years 2008 – 2009")
```





# Where Next?

Lots more add-ons

- `directlabels` (provides `geom_dl`)
- `GGally`
- `autoplot`
- `gridExtra`
- `ggfortify`
- `ggExtra`

Hadley Wickham's book at GitHub:

<https://github.com/hadley/ggplot2-book>

# Conclusions

## Benefits

- Clean, intuitive interface
- Easy to learn
- Enables complex plots

## Downsides

- Only properly available in R
- Not interactive

*If I hadn't seen such riches, I could live with being poor*

# Get In Touch

Mick Cooney  
michael.cooney@applied.ai

Slides and code available on BitBucket:  
[https://www.bitbucket.org/kaybenleroll/dublin\\_r\\_workshops](https://www.bitbucket.org/kaybenleroll/dublin_r_workshops)