

# Introduction to **ggplot2** package

**Felipe Dalla Lana, Ph.D.**



**Penn State**  
College of  
Agricultural Sciences

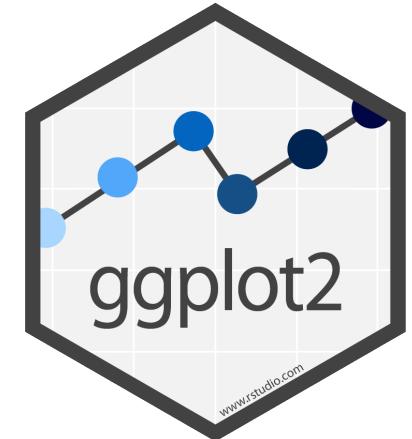
# What is ggplot2



## Hadley Wickham

- Chief Scientist at Rstudio
- University of Auckland
- Stanford University
- Rice University

**First released - July of 2007**  
**40 version since**

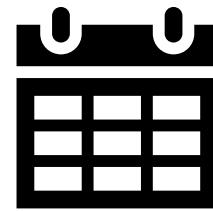
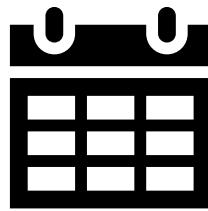
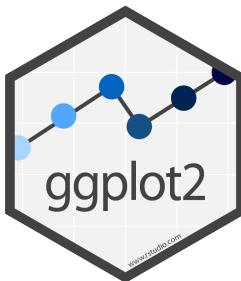


## Authors of current version (3.3.3)

Hadley Wickham  
Winston Chang  
Lionel Henry  
Thomas Lin Pedersen  
Kohske Takahashi  
Claus Wilke  
Kara Woo  
Hiroaki Yutani  
Dewey Dunnington

A screenshot of a website showing the contributors to ggplot2. At the top, it says "Contributors 250". Below this, there are two rows of circular profile pictures of people. At the bottom, there is a blue icon of three interconnected circles and the text "+ 239 contributors".

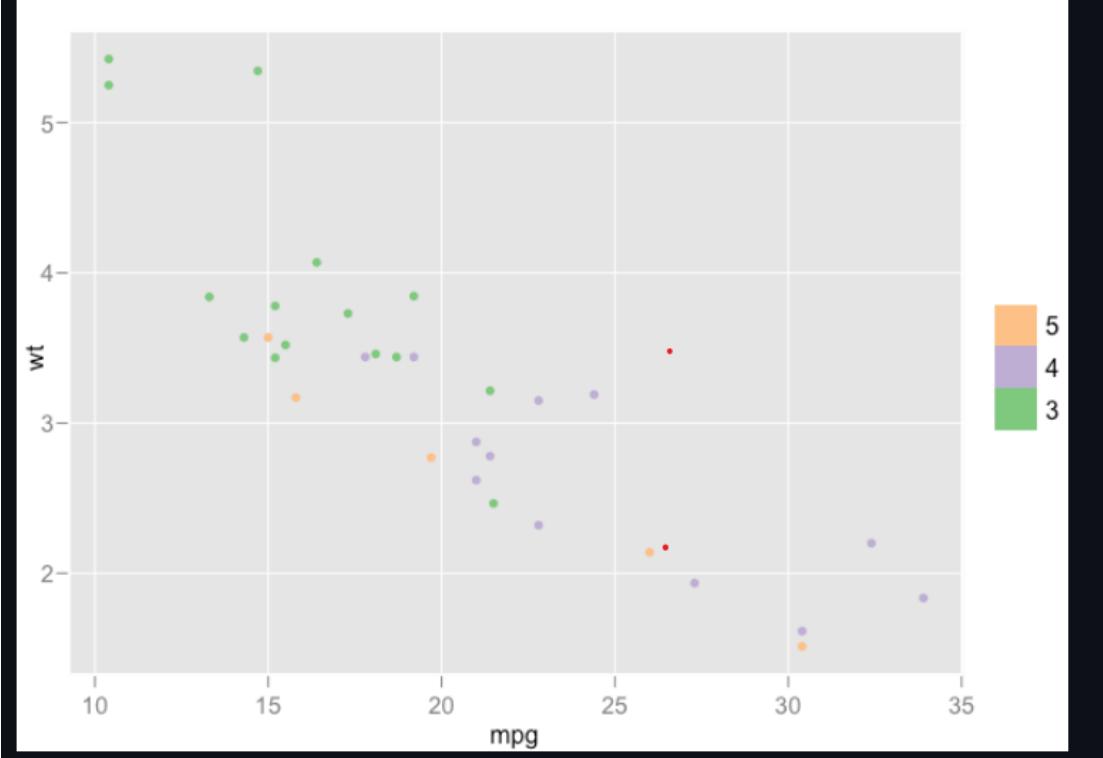
# How about ggplot1?

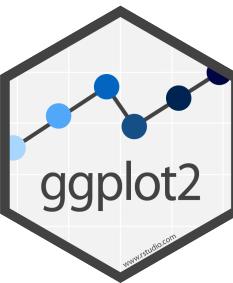


April 2006  
v. 0.2.2

October 2008  
v. 0.4.2

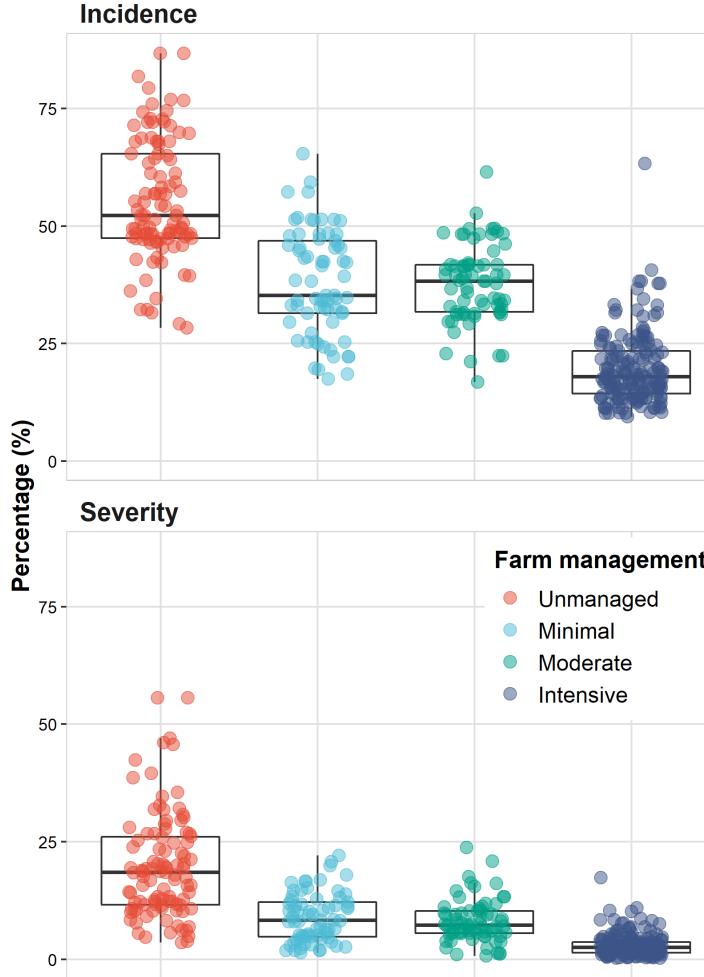
```
mtcars %>%
  ggplot(list(x = mpg, y = wt)) %>%
  ggpoint(list(colour = gear)) %>%
  scbrewer()
```





# The package grammar

Disease distribution across farm management



```
ggplot(survey_data_long, aes(x= farm_management, y = rate))+  
  geom_boxplot(outlier.alpha = 0)+  
  geom_jitter(aes(color = farm_management),  
             size = 3, width = 0.2, alpha = .5)+  
  
  facet_wrap(~metric, ncol = 1)+  
  scale_color_npg()  
  
  labs(title = "Disease distribution across farm management",  
        x = NULL,  
        y = "Percent (%)",  
        color = "Farm management")  
  
  theme(  
    panel.background = element_blank(),  
    panel.grid.major = element_line(colour = "grey88"),  
    panel.border = element_rect(colour = "grey80", fill = NA),  
  
    axis.text.y = element_text(colour = "black"),  
    axis.text.x = element_blank(),  
    title = element_text(face = "bold", size = 13),  
  
    legend.position = c(0.81, .37),  
    legend.key = element_rect(fill = "white"),  
    legend.text = element_text(colour = "black", size=12),  
  
    strip.text.x = element_text(hjust = 0.01, face = "bold", size = 14),  
    strip.background = element_blank())  
  
  ggsave("graph/target.tiff",  
         width = 140, height = 200, units = "mm", dpi = 300)
```

ggplot function and data

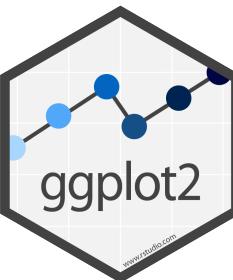
Geometric elements

Aesthetic mappings

Facets, scale, labs, coordinates

Theme

Save



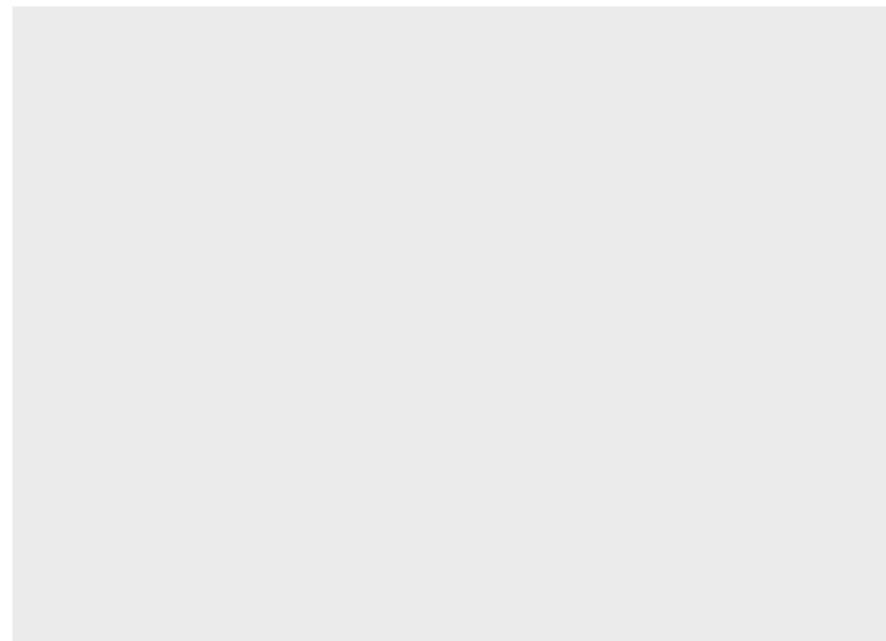
# Data and ggplot function

```
survey_data <- read_csv(here::here("data", "survey_clean.csv")) %>%  
  select(-c(farm, region, zone, district, lon, lat))
```

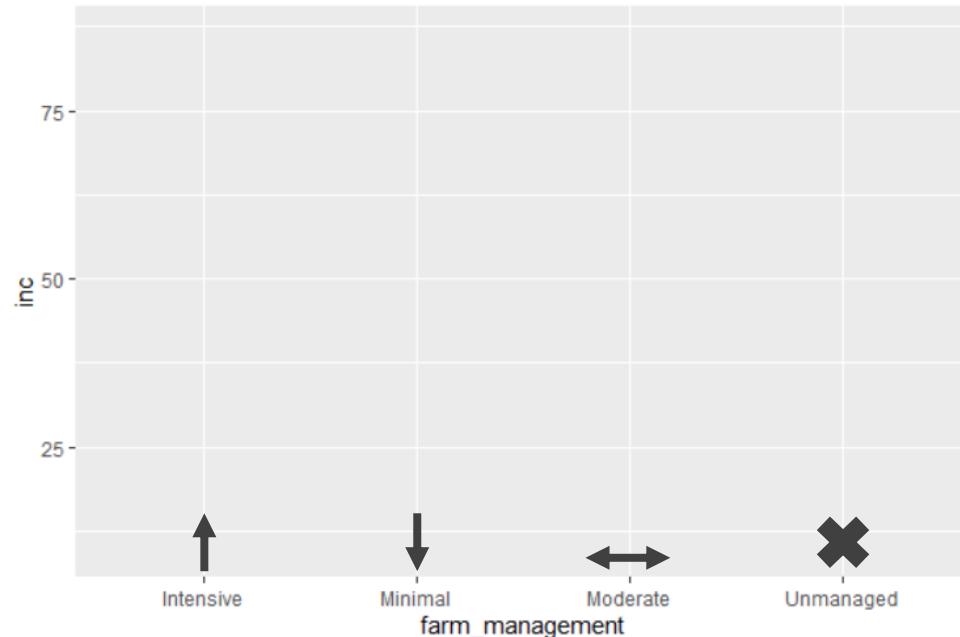
```
> head(survey_data)
```

```
# A tibble: 6 x 7  
  altitude cultivar shade    cropping_system farm_management   inc   sev2  
        <dbl> <chr>  <chr>    <chr>          <chr>      <dbl>  <dbl>  
1     1100 Local   Sun     Plantation     Unmanaged     86.7  55.6  
2     1342 Mixture Mid shade Plantation    Minimal      51.3  17.9  
3     1434 Mixture Mid shade Plantation    Minimal      43.2  8.25  
4     1100 Local   Sun     Plantation     Unmanaged     76.7  46.1  
5     1400 Local   Sun     Plantation     Unmanaged     47.2  12.3  
6     1342 Mixture Mid shade Plantation    Minimal      51.3  19.9
```

```
ggplot()
```

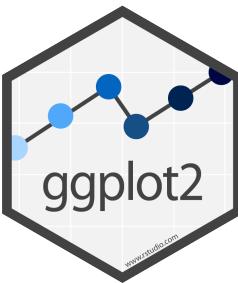


```
ggplot(survey_data, aes(x= farm_management, y = inc))
```



```
> class(survey_data$farm_management)  
[1] "character"  
> levels(survey_data$farm_management)  
NULL
```

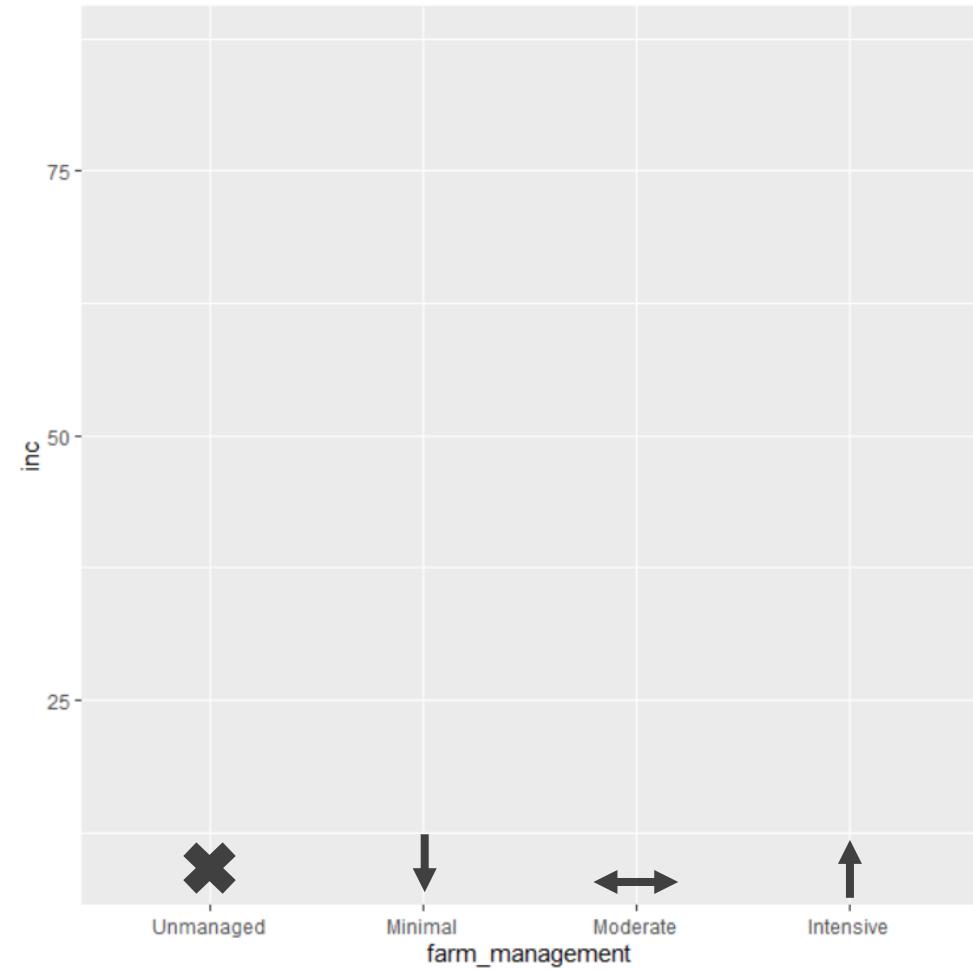
# Data and ggplot function

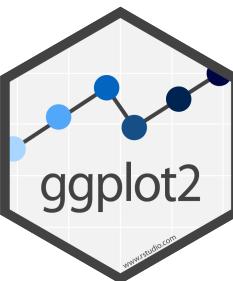


```
survey_data = survey_data %>%  
  mutate(farm_management = factor(farm_management,  
    levels = c("Unmanaged", "Minimal", "Moderate", "Intensive")))  
  
head(survey_data)
```

```
> head(survey_data)  
# A tibble: 6 x 7  
  altitude cultivar shade  cropping_system farm_management   inc  sev2  
    <dbl> <chr>   <chr> <chr>           <chr>      <dbl> <dbl>  
1     1100 Local    Sun   Plantation Unmanaged       86.7  55.6  
2     1342 Mixture  Mid shade Plantation Minimal        51.3  17.9  
3     1434 Mixture  Mid shade Plantation Minimal        43.2  8.25  
4     1100 Local    Sun   Plantation Unmanaged       76.7  46.1  
5     1400 Local    Sun   Plantation Unmanaged       47.2  12.3  
6     1342 Mixture  Mid shade Plantation Minimal        51.3  19.9
```

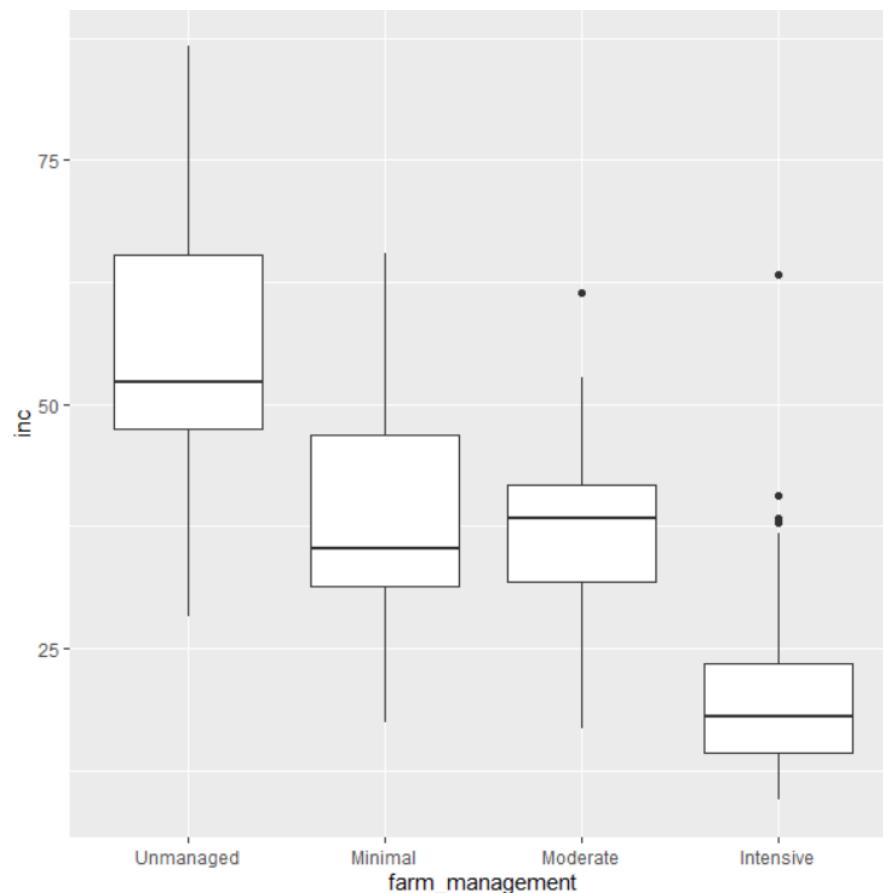
```
> class(survey_data$farm_management)  
[1] "factor"  
> levels(survey_data$farm_management)  
[1] "Unmanaged" "Minimal"   "Moderate"  "Intensive"
```



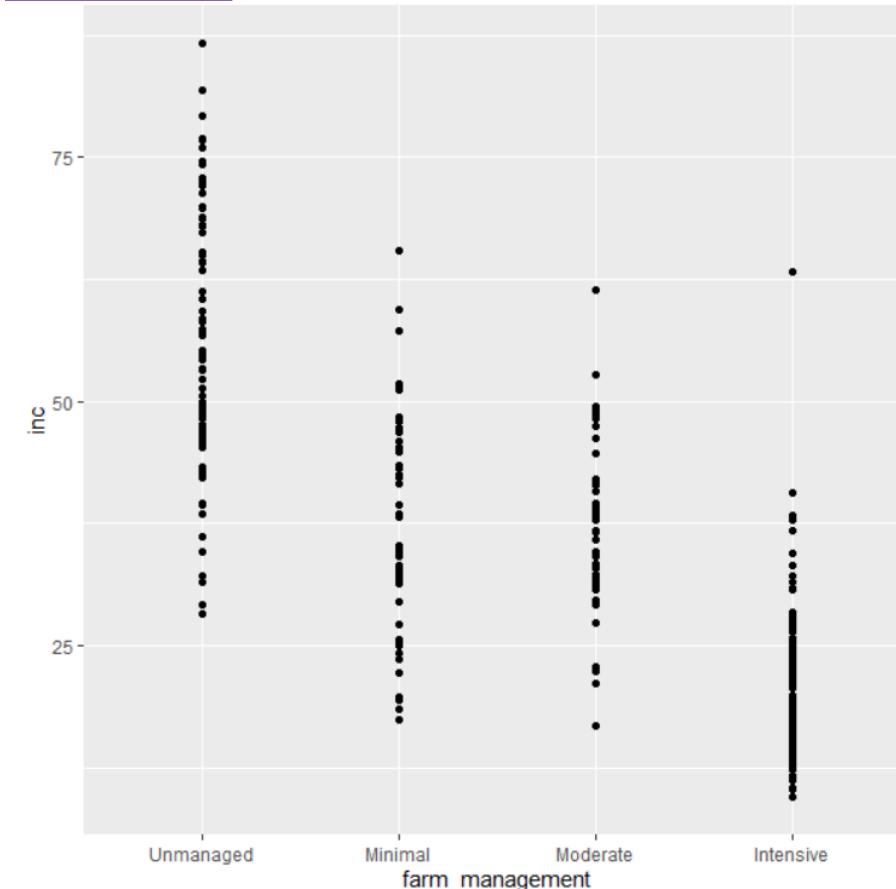


# Geometric elements

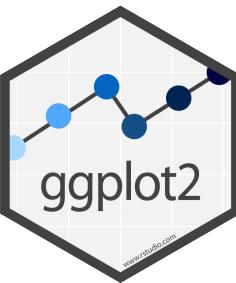
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_boxplot()
```



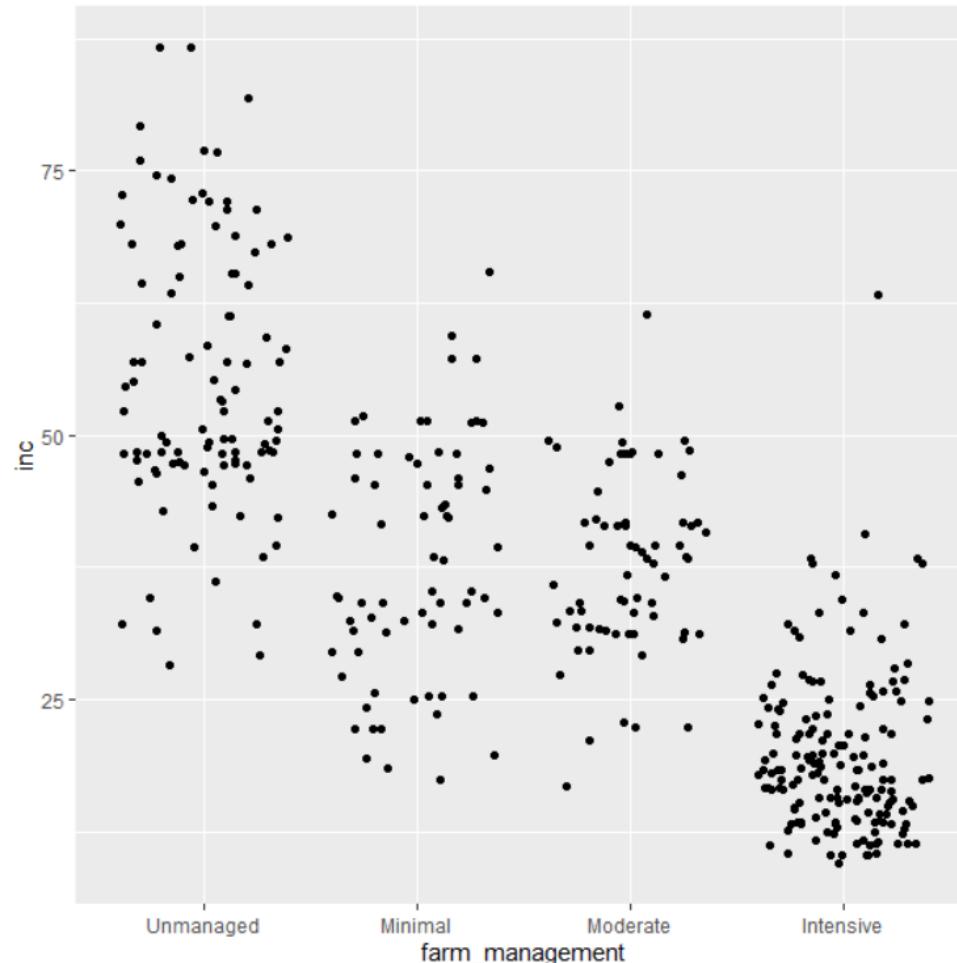
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_point()
```



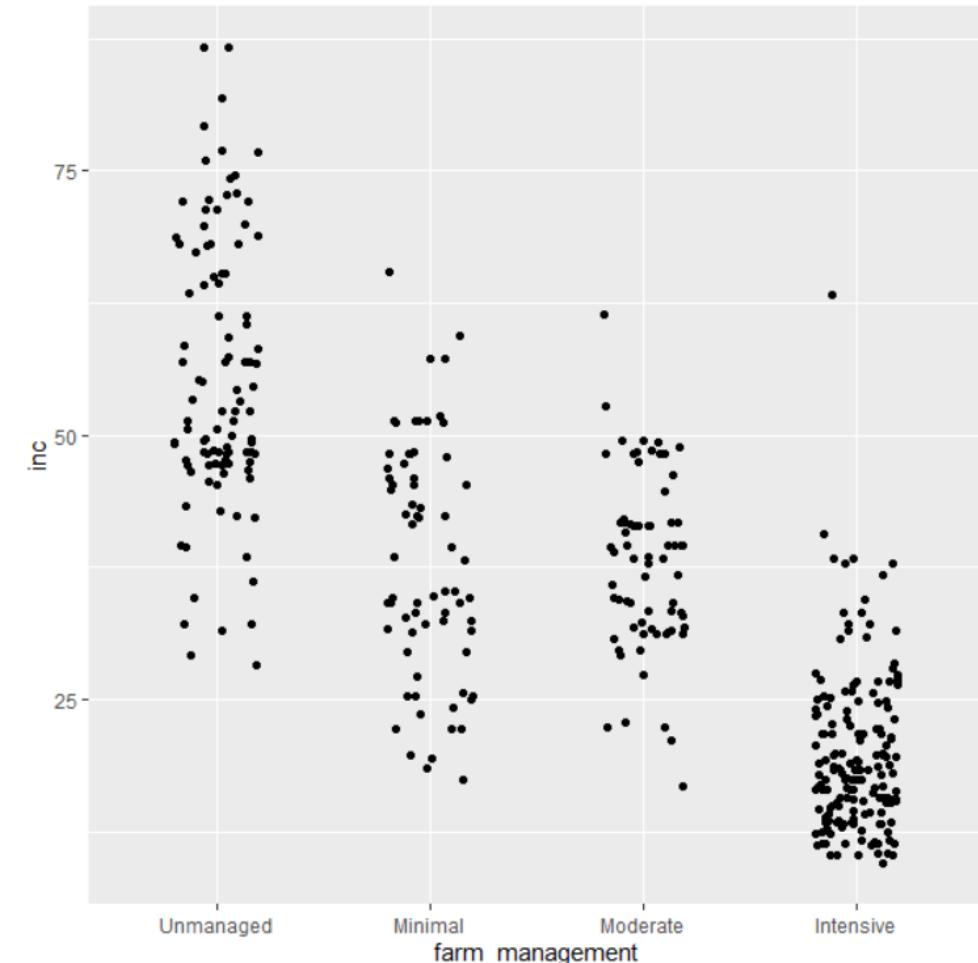
# Geometric elements



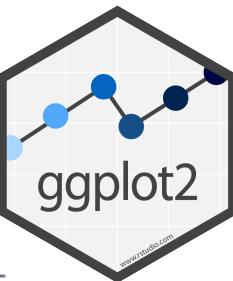
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter()
```



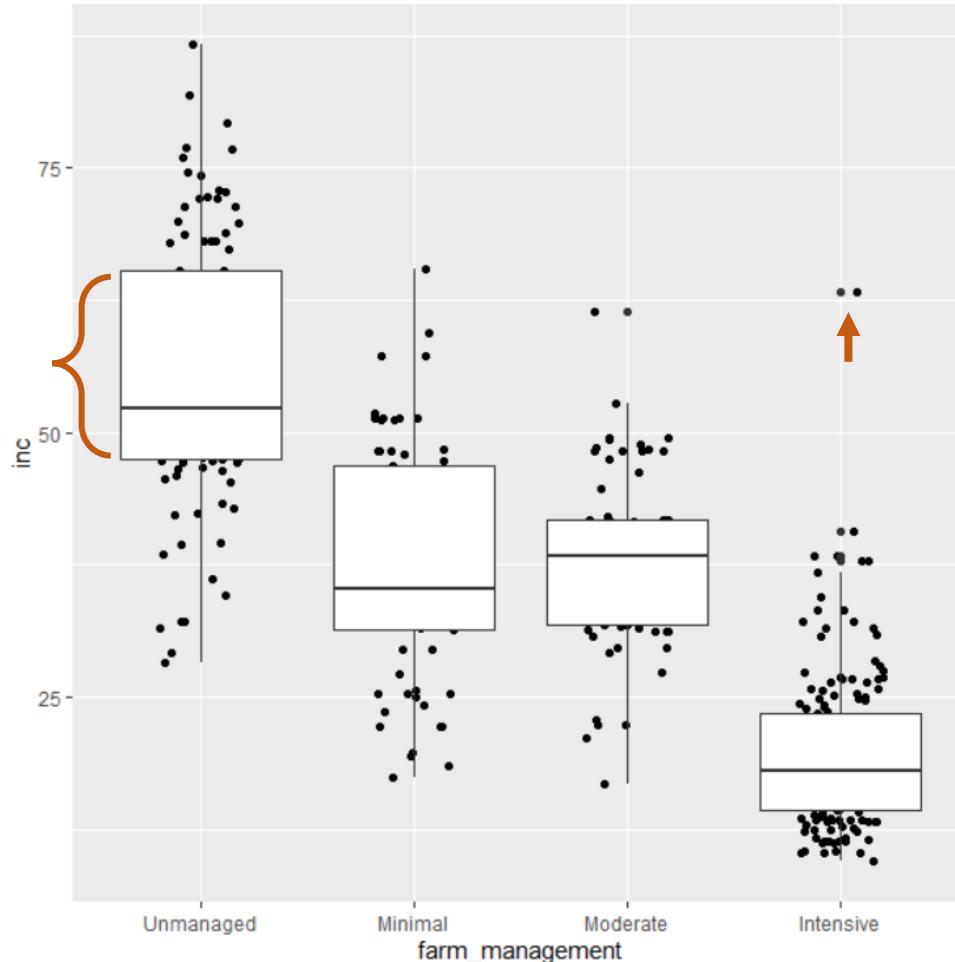
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter(width = 0.2)
```



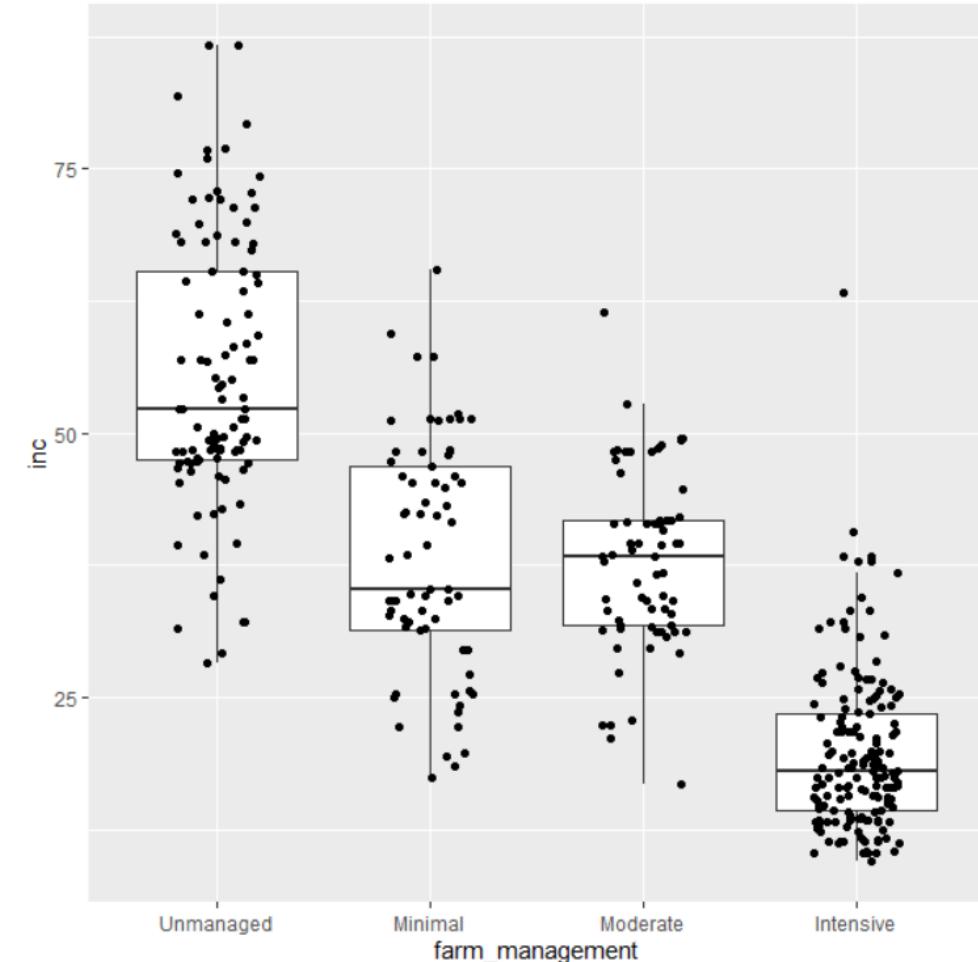
# Geometric elements



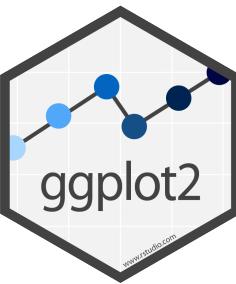
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter(width = 0.2)+ ←  
  geom_boxplot()
```



```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_boxplot(outlier.alpha = 0) + ←  
  geom_jitter(width = 0.2)
```

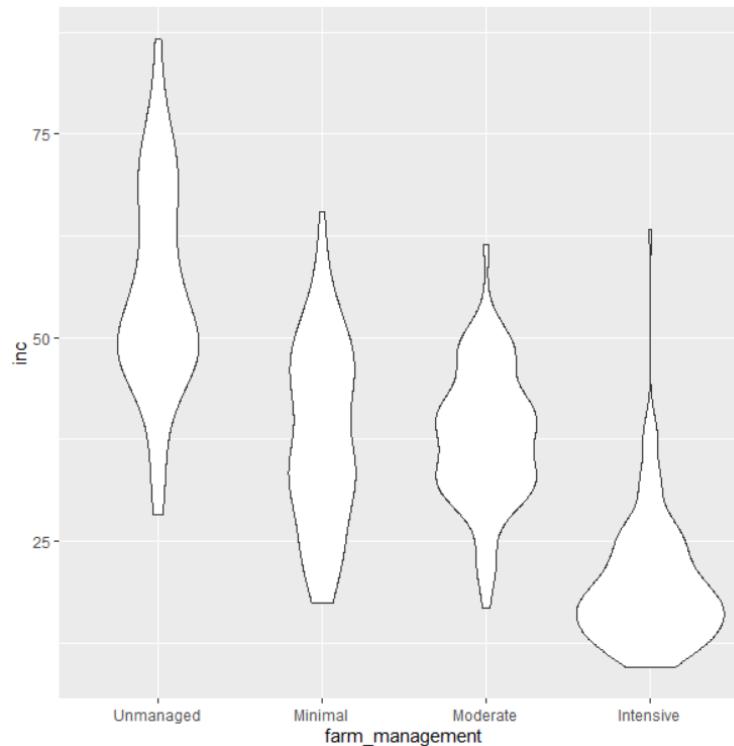


# Geometric elements



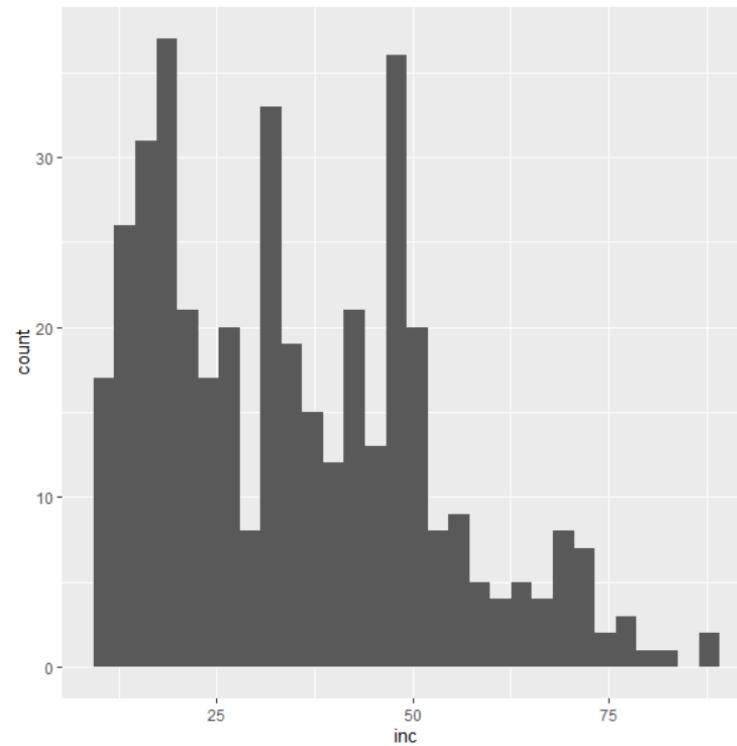
## geom\_violin()

```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_violin()
```



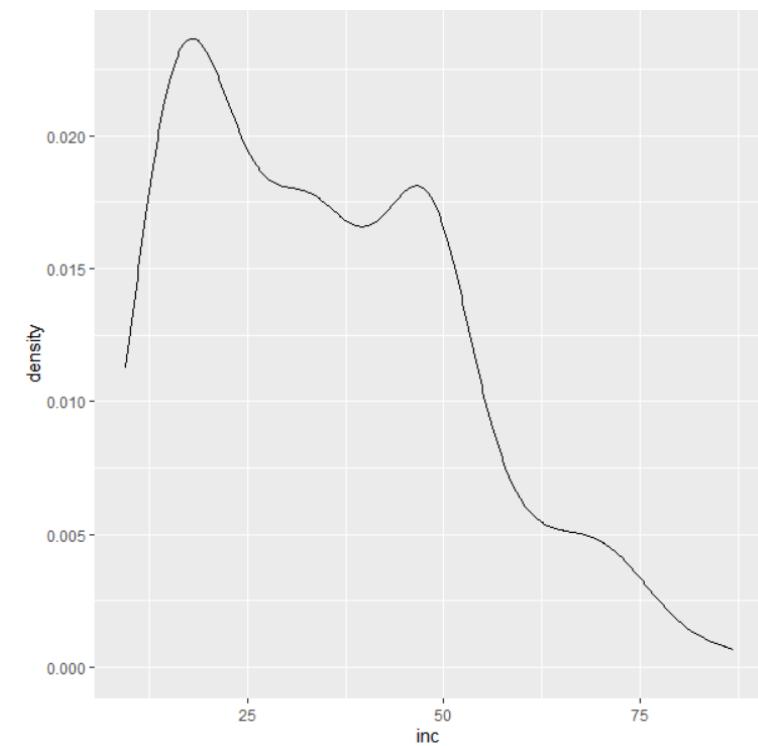
## geom\_histogram()

```
ggplot(survey_data, aes(x= inc))+  
  geom_histogram()
```

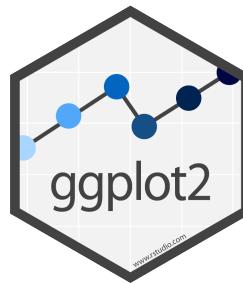


## geom\_density()

```
ggplot(survey_data, aes(x= inc))+  
  geom_density()
```

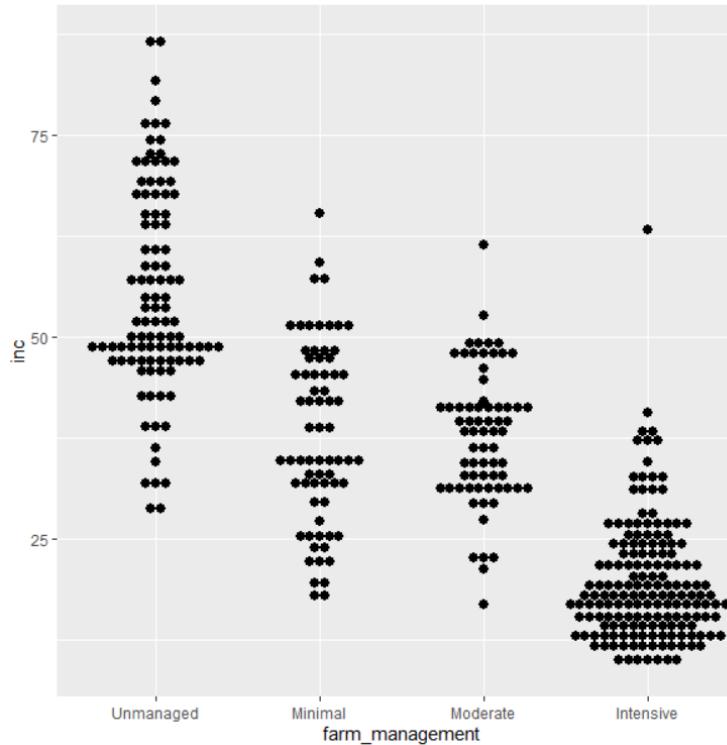


# Geometric elements



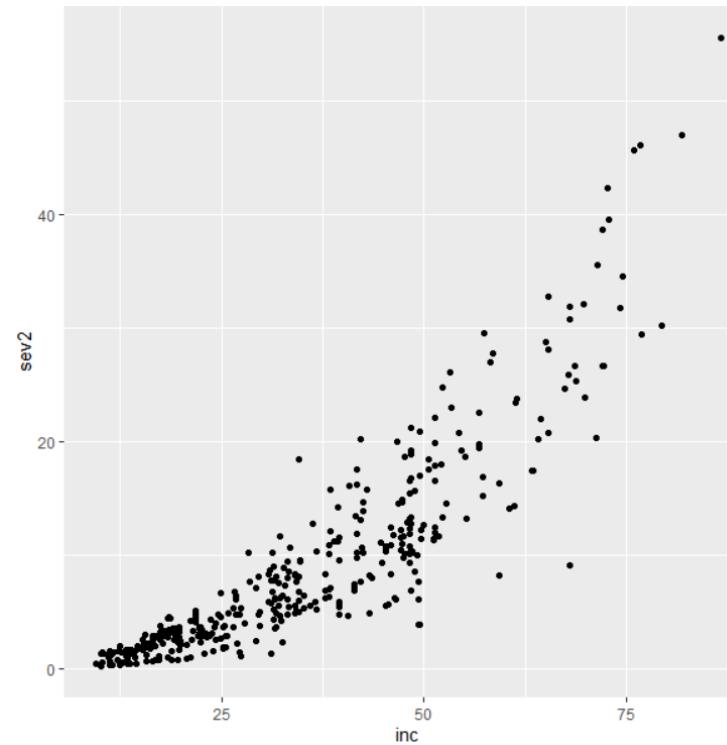
## geom\_dotplot()

```
ggplot(survey_data,aes(x= farm_management, y = inc))+  
  geom_dotplot(binaxis = "y",  
               stackdir = "center", binwidth = 1.2)
```



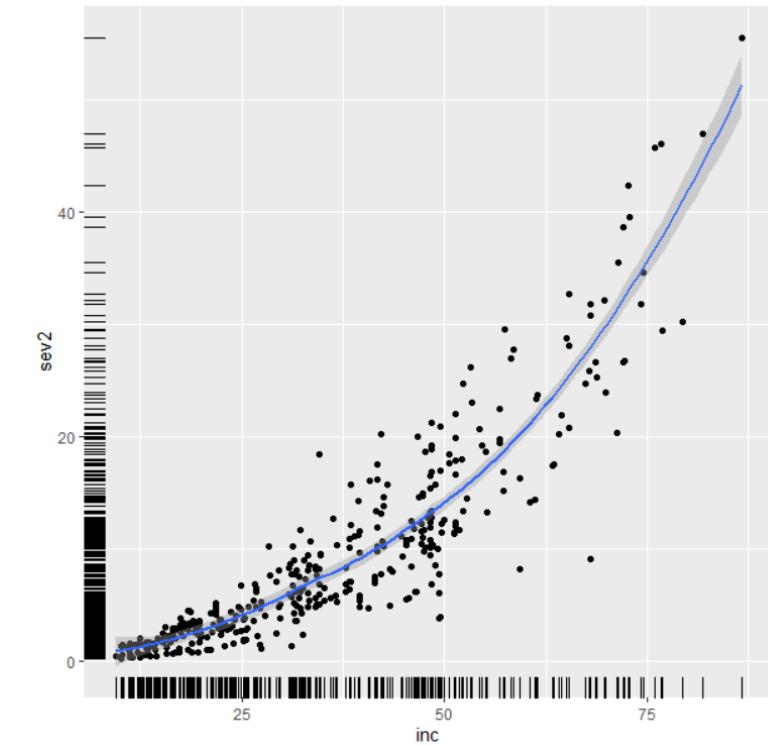
## geom\_point()

```
ggplot(survey_data, aes(x = inc, y = sev2))+  
  geom_point()
```



**geom\_point() +  
geom\_smooth() +  
geom\_rug()**

```
ggplot(survey_data, aes(x = inc, y = sev2))+  
  geom_point() + geom_smooth() + geom_rug()
```



# Geometric elements

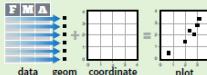
## Data Visualization with ggplot2

Cheat Sheet

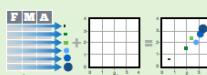


### Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same few components: a **data set**, a set of **geoms**—visual marks that represent data points, and a **coordinate system**.



To display data values, map variables in the data set to aesthetic properties of the geom like **size**, **color**, and **x** and **y** locations.



Build a graph with **qplot()** or **ggplot()**

**aesthetic mappings**    **data**    **geom**  
`ggplot(qplot(x = city, y = hwy, color = cyl, data = mpg, geom = "point"))`  
Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

**ggplot(data = mpg, aes(x = cyl, y = hwy))**

Begins a plot that you finish by adding layers to. No defaults, but provides more control than qplot().

**data**    **add layers, elements with +**    **additional elements**  
`ggplot(mpg, aes(hwy, city)) + geom_point(aes(color = cyl)) + geom_smooth(method = "lm") + coord_cartesian() + scale_color_gradient() + theme_bw()`

Add a new layer to a plot with a **geom\_\***() or **stat\_\***() function. Each provides a geom, a set of aesthetic mappings, and a default stat and position adjustment.

**last\_plot()**

Returns the last plot

**ggsave("plot.png", width = 5, height = 5)**  
Saves last plot as 5"x5" file named "plot.png" in working directory. Matches file type to file extension.

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.		
<b>One Variable</b>		
<b>Continuous</b>		
<pre>a + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size b + geom_area(aes(y = ..density..), stat = "bin")</pre>	<pre>f + geom_blank()</pre>	<pre>i + geom_bin2d(binwidth = c(5, 0.5)) xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size, weight</pre>
<pre>a + geom_density(kernel = "gaussian") x, y, alpha, color, fill, linetype, size, weight</pre>	<pre>f + geom_jitter()</pre>	<pre>i + geom_density2d() x, y, alpha, colour, linetype, size</pre>
<pre>a + geom_dotplot()</pre>	<pre>f + geom_point()</pre>	<pre>i + geom_hex() x, y, alpha, colour, fill, size</pre>
<pre>a + geom_freqpoly()</pre>	<pre>f + geom_quantile()</pre>	<pre>j &lt;- ggplot(economics, aes(date, unemployed))</pre>
<pre>x, y, alpha, color, linetype, size b + geom_freqpoly(aes(y = ..density..))</pre>	<pre>f + geom_rug(sides = "bl") alpha, color, linetype, size</pre>	<pre>j + geom_area() x, y, alpha, color, fill, linetype, size</pre>
<pre>a + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight</pre>	<pre>f + geom_smooth(model = lm) x, y, alpha, color, fill, linetype, size, weight</pre>	<pre>j + geom_line() x, y, alpha, color, fill, linetype, size</pre>
<pre>b + geom_histrogram(aes(y = ..density..))</pre>	<pre>f + geom_text(aes(label = city)) AB</pre>	<pre>j + geom_step(direction = "hv") x, y, alpha, color, linetype, size, vjust</pre>
<b>Discrete</b>		
<pre>b &lt;- ggplot(mpg, aes(fl))</pre>	<pre>b + geom_bar()</pre>	<pre>df &lt;- data.frame(grp = c("A", "B"), fit = 4.5, se = 1.2)</pre>
<pre>x, alpha, color, fill, linetype, size, weight</pre>	<pre>k + geom_bar(stat = "identity") x, y, alpha, color, fill, linetype, size, weight</pre>	<pre>k &lt;- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se))</pre>
<b>Graphical Primitives</b>		
<pre>c &lt;- ggplot(map, aes(long, lat))</pre>	<pre>g + geom_bar(stat = "identity") x, y, alpha, color, fill, linetype, size, weight</pre>	<pre>k + geom_crossbar(fatten = 2) x, y, ymax, ymin, alpha, color, fill, linetype, size</pre>
<pre>c + geom_polygon(aes(group = group)) x, y, alpha, color, fill, linetype, size</pre>	<pre>g + geom_boxplot() lower, middle, upper, x, ymax, ymin, alpha, color, fill, linetype, size, width (also <b>geom_errorbar()</b>)</pre>	<pre>k + geom_errorbar() x, ymax, ymin, alpha, color, linetype, size, weight</pre>
<pre>d &lt;- ggplot(economics, aes(date, unemployed))</pre>	<pre>g + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill</pre>	<pre>k + geom_linerange() x, ymin, ymax, alpha, color, linetype, size</pre>
<pre>d + geom_path(lineend = "butt", linejoin = "round", linemetre = 1) x, y, alpha, color, linetype, size</pre>	<pre>g + geom_ribbon(aes(ymin = unemployed - 900, ymax = unemployed + 900)) x, ymax, ymin, alpha, color, fill, linetype, size</pre>	<pre>k + geom_pointrange() x, y, ymin, ymax, alpha, color, fill, linetype, size</pre>
<pre>d + geom_rect(aes(xmin = long, ymin = lat, xmax = long + delta_long, ymax = lat + delta_lat)) x, xmin, ymax, ymin, alpha, color, fill, linetype, size</pre>	<pre>h &lt;- ggplot(seals, aes(x = long, y = lat))</pre>	<pre>maps &lt;- data(maps)</pre>
<pre>e + geom_segment(aes(xend = long + delta_long, yend = lat + delta_lat)) x, xend, yend, alpha, color, linetype, size</pre>	<pre>e + geom_jitter()</pre>	<pre>maps &lt;- data(maps)</pre>
<pre>e + geom_rect(aes(xmin = long, ymin = lat, xmax = long + delta_long, ymax = lat + delta_lat)) x, xmin, ymax, ymin, alpha, color, fill, linetype, size</pre>	<pre>h + geom_jitter()</pre>	<pre>m + geom_map(aes(map_id = state), map = map) + expand_limits(x = map\$long, y = map\$lat) map_id, alpha, color, fill, linetype, size</pre>
<b>Three Variables</b>		
<pre>e + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)</pre>	<pre>m + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE)</pre>	<pre>m + geom_contour(aes(z = z)) x, y, z, alpha, colour, linetype, size, weight</pre>
<pre>sealsSz &lt;- with(seals, sqrt(delta_long^2 + delta_lat^2))</pre>	<pre>m + geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size</pre>	<pre>m + geom_hex(stat = bin_hex()) hexagonal heatmap of 2d bin counts</pre>
<pre>m &lt;- ggplot(seals, aes(long, lat))</pre>		<pre>geom_hex(stat = bin_hex())</pre>
<pre>e + geom_contour(aes(z = z)) x, y, z, alpha, colour, linetype, size, weight</pre>		<pre>geom_hex(stat = bin_hex())</pre>

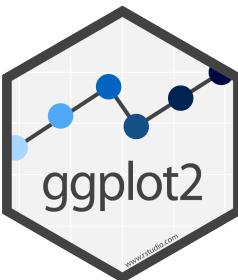
Learn more at [docs.ggplot2.org](http://docs.ggplot2.org) • ggplot2: 0.9.3.1 • Updated: 3/15



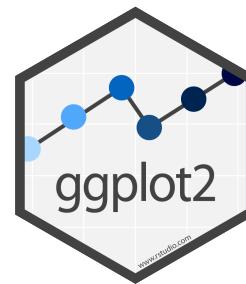
part of the **tidyverse**  
3.3.2

Search...

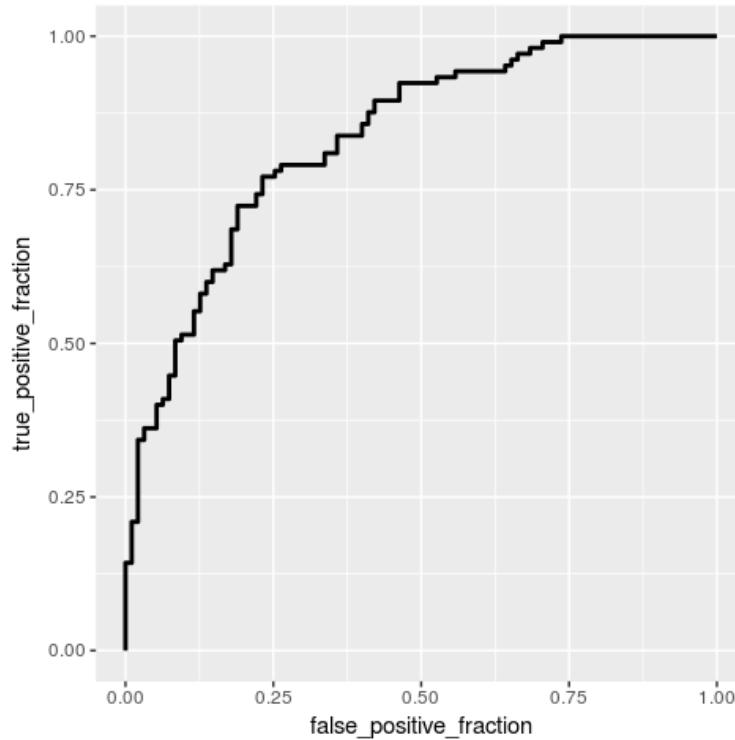
	<code>geom_bar()</code>	Bar charts
	<code>geom_col()</code>	
	<code>stat_count()</code>	
	<code>geom_bin2d()</code>	Heatmap of 2d bin counts
	<code>geom_blank()</code>	Draw nothing
	<code>geom_boxplot()</code>	A box and whiskers plot (in the style of Tukey)
	<code>stat_boxplot()</code>	
	<code>geom_contour()</code>	2D contours of a 3D surface
	<code>geom_contour_filled()</code>	
	<code>stat_contour()</code>	
	<code>stat_contour_filled()</code>	
	<code>geom_count()</code>	Count overlapping points
	<code>stat_sum()</code>	
	<code>geom_density()</code>	Smoothed density estimates
	<code>stat_density()</code>	
	<code>geom_density_2d()</code>	Contours of a 2D density estimate
	<code>geom_density_2d_filled()</code>	
	<code>stat_density_2d()</code>	
	<code>stat_density_2d_filled()</code>	
	<code>geom_dotplot()</code>	Dot plot
	<code>geom_errorbar()</code>	Horizontal error bars
	<code>geom_function()</code>	Draw a function as a continuous curve
	<code>stat_function()</code>	
	<code>geom_hex()</code>	Hexagonal heatmap of 2d bin counts
	<code>stat_hex()</code>	
	<code>geom_frequpoly()</code>	Histograms and frequency polygons
	<code>geom_histrogram()</code>	
	<code>stat_hex()</code>	
	<code>geom_jitter()</code>	Jittered points
	<code>geom_crossbar()</code>	Vertical intervals: lines, crossbars & errorbars
	<code>geom_errorbar()</code>	
	<code>geom_linerange()</code>	
	<code>geom_pointrange()</code>	



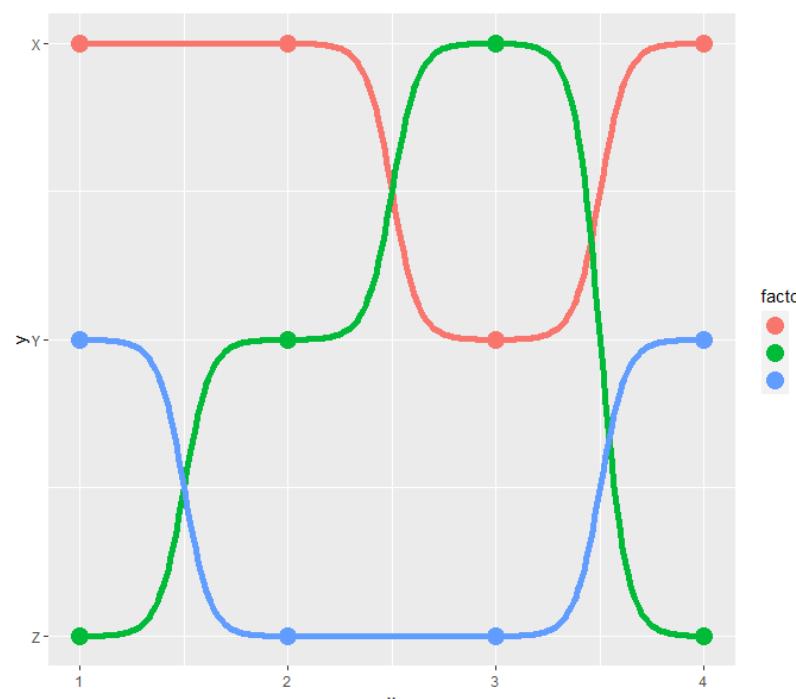
# Geometric elements – extensions packages



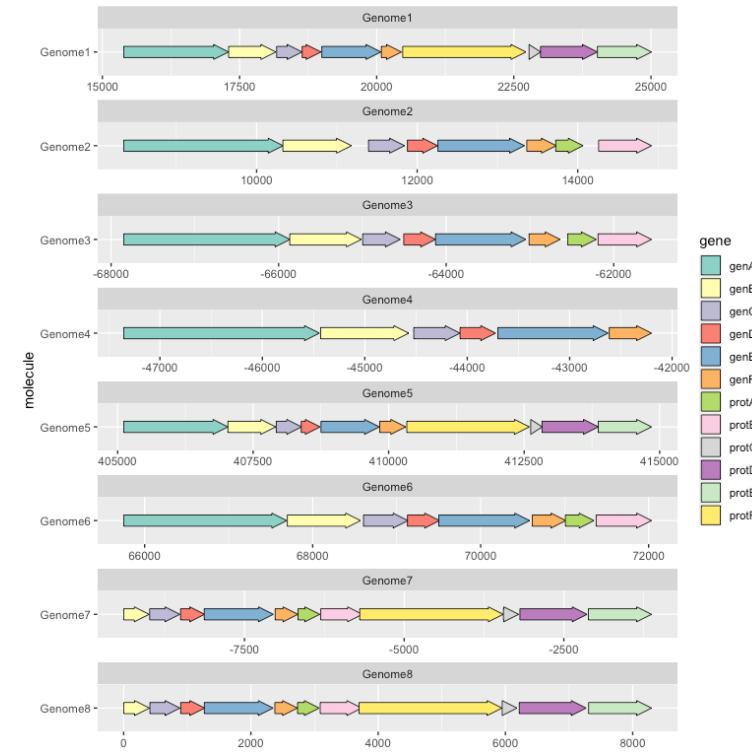
**plotROC package**  
`geom_roc()`

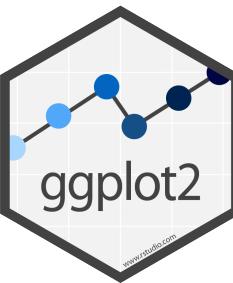


**ggbump package**  
`geom_bump()`



**gggenes package**  
`geom_gene_arrow()`





# Aesthetics

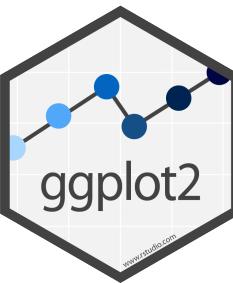
- How the variables will be mapped
- Inside of aes() command
- If aes() is in ggplot() function, the aesthetics will be the same for all geom\_xxx()
- If aes() is in a geom\_xxx(), the aesthetics will be especificaly for that geom\_xxx()

```
ggplot(survey_data, aes(x= farm_management, y = inc,  
                        shape = farm_management,  
                        color = farm_management))+  
  geom_boxplot(outlier.alpha = 0) +  
  geom_jitter(width = 0.2)
```

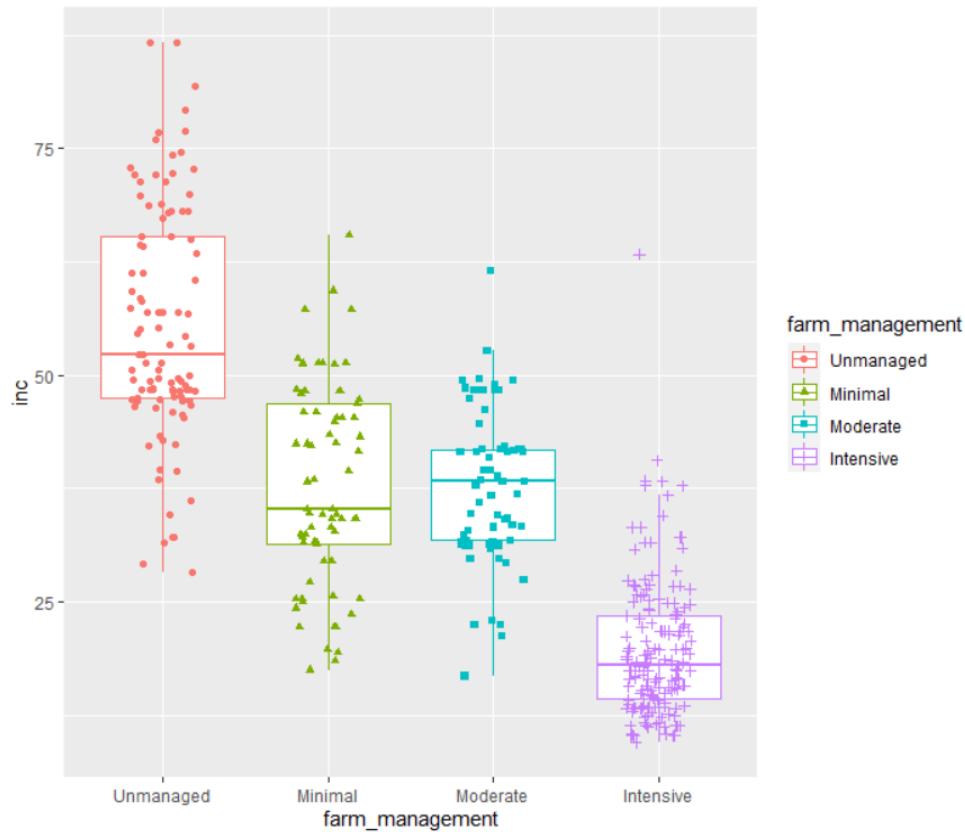
## Most common

- X and y-axis
- Color; fill; and alpha
- Shape
- Size
- Linetype

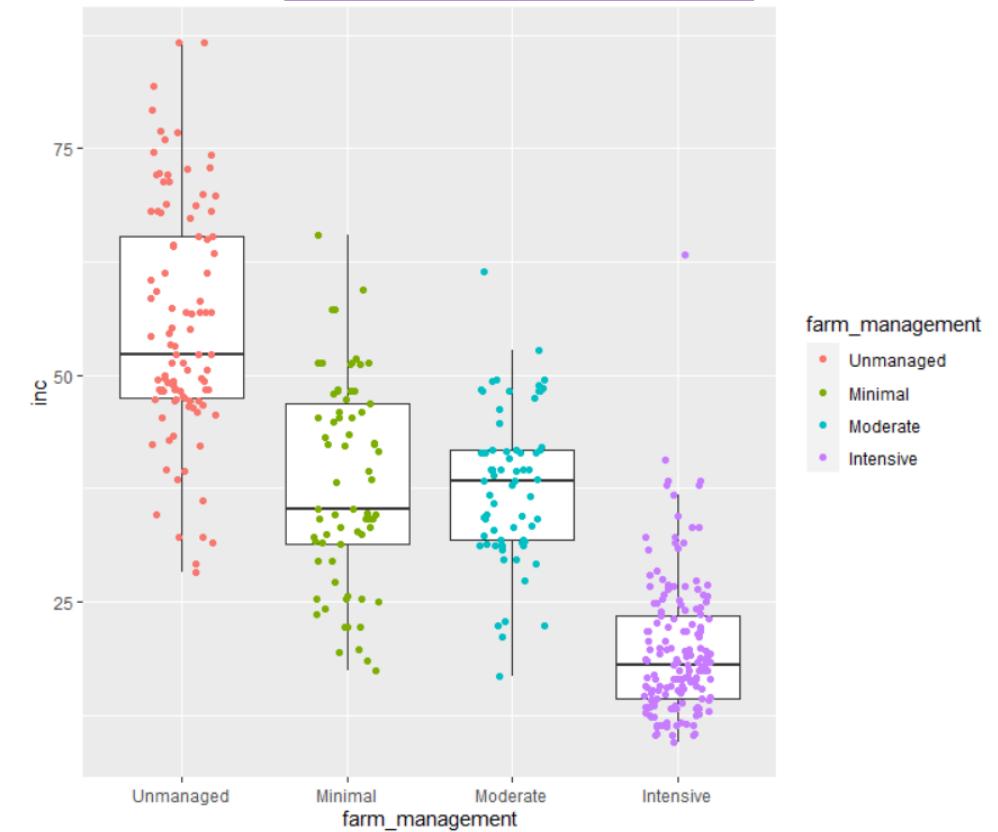
# Aesthetics

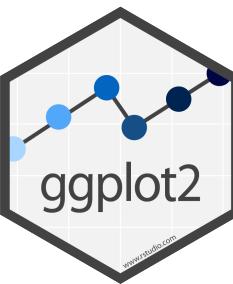


```
ggplot(survey_data, aes(x= farm_management, y = inc,  
                        shape = farm_management,  
                        color = farm_management))+  
  geom_boxplot(outlier.alpha = 0) +  
  geom_jitter(width = 0.2)
```



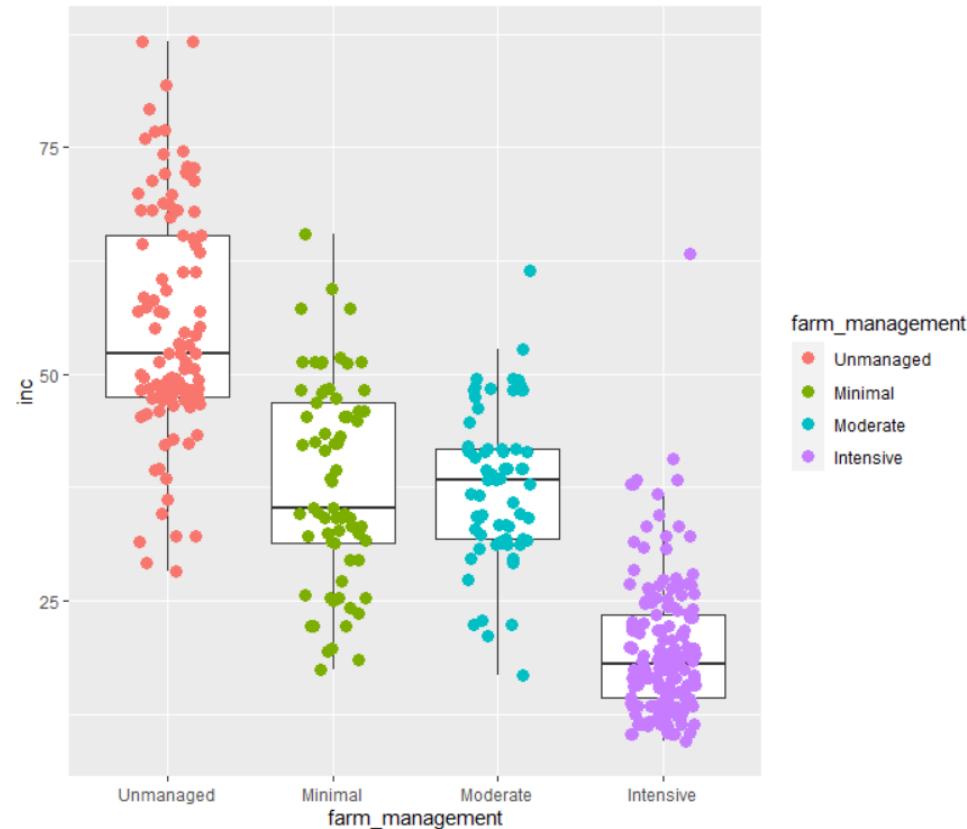
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_boxplot(outlier.alpha = 0) +  
  geom_jitter(aes(color = farm_management),width = 0.2)
```



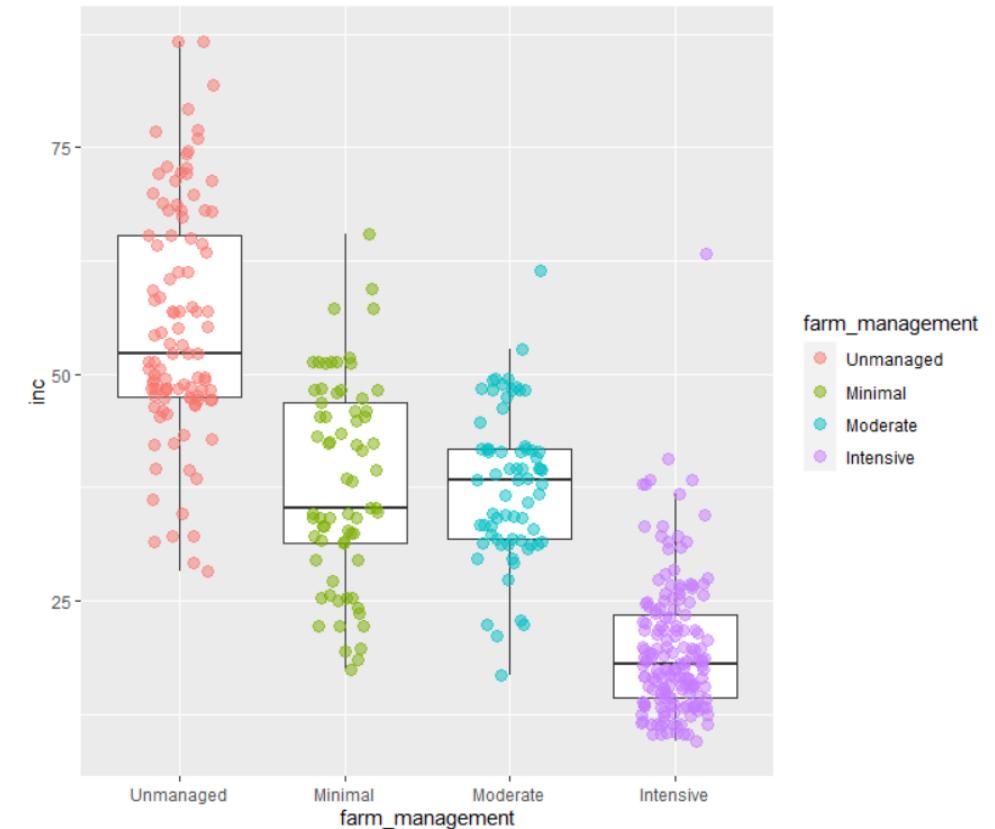


# Aesthetics - color

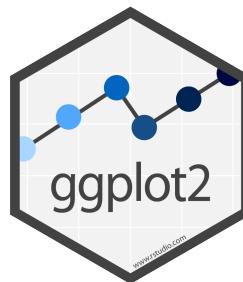
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_boxplot(outlier.alpha = 0) +  
  geom_jitter(aes(color = farm_management), width = 0.2,  
              size = 3)
```



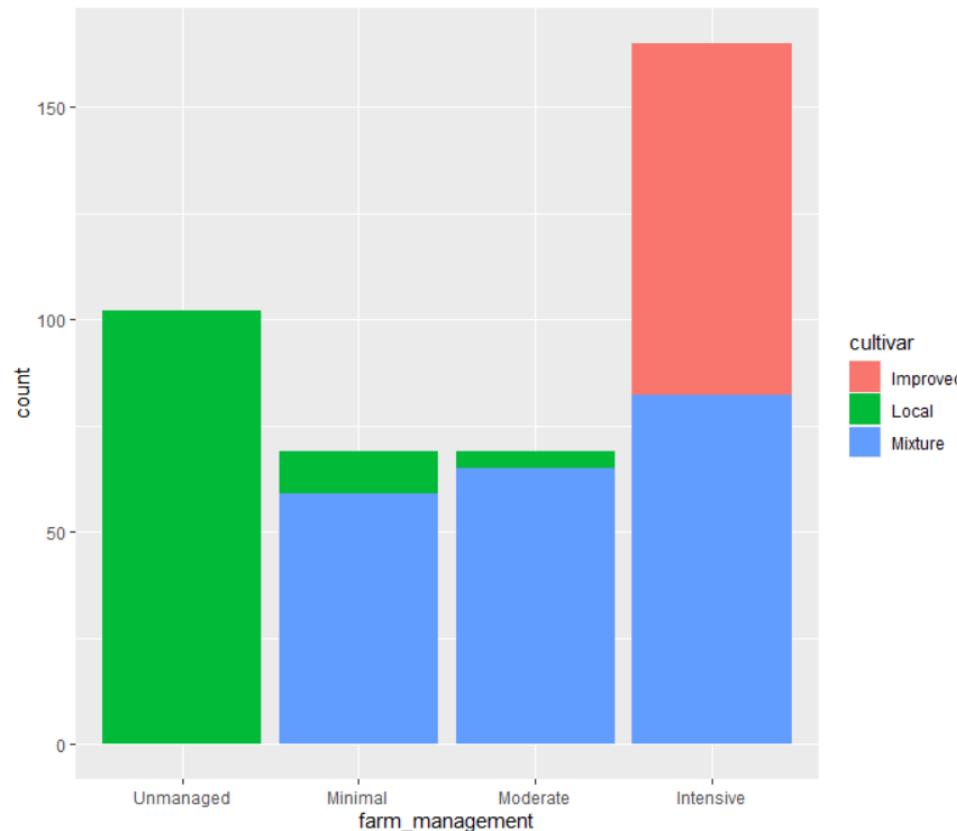
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_boxplot(outlier.alpha = 0) +  
  geom_jitter(aes(colour = farm_management), width = 0.2,  
              size = 3, alpha = 0.5)
```



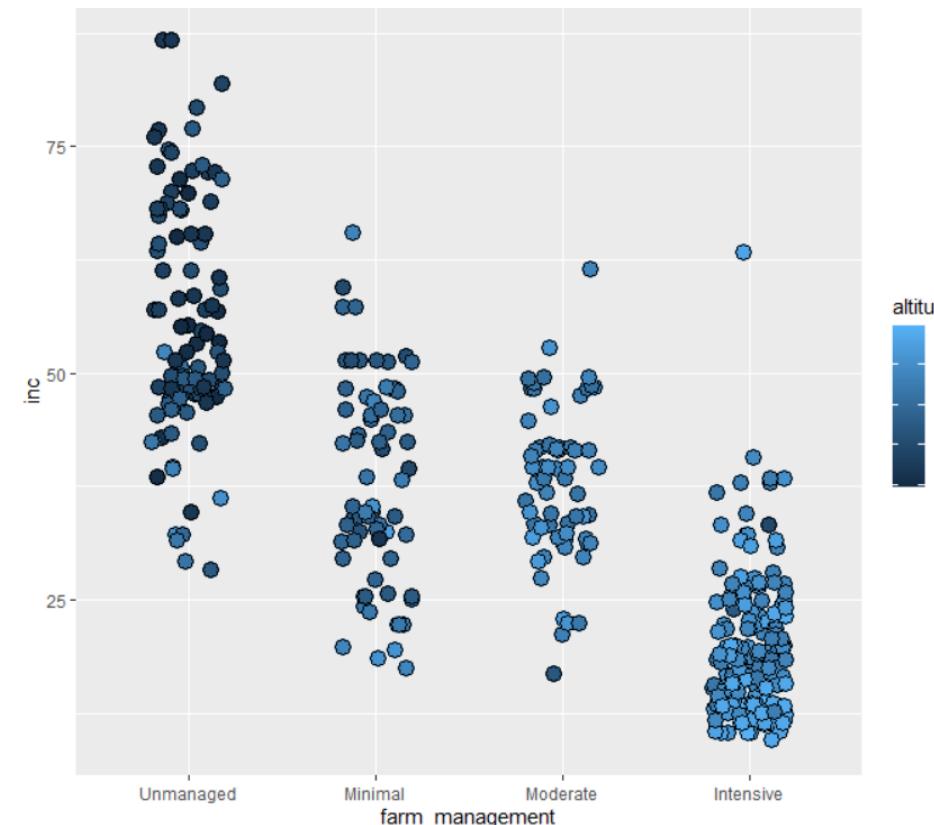
# Aesthetics - Fill

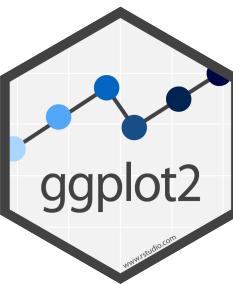


```
ggplot(survey_data, aes(x= farm_management))  
  geom_bar(aes(fill = cultivar))
```



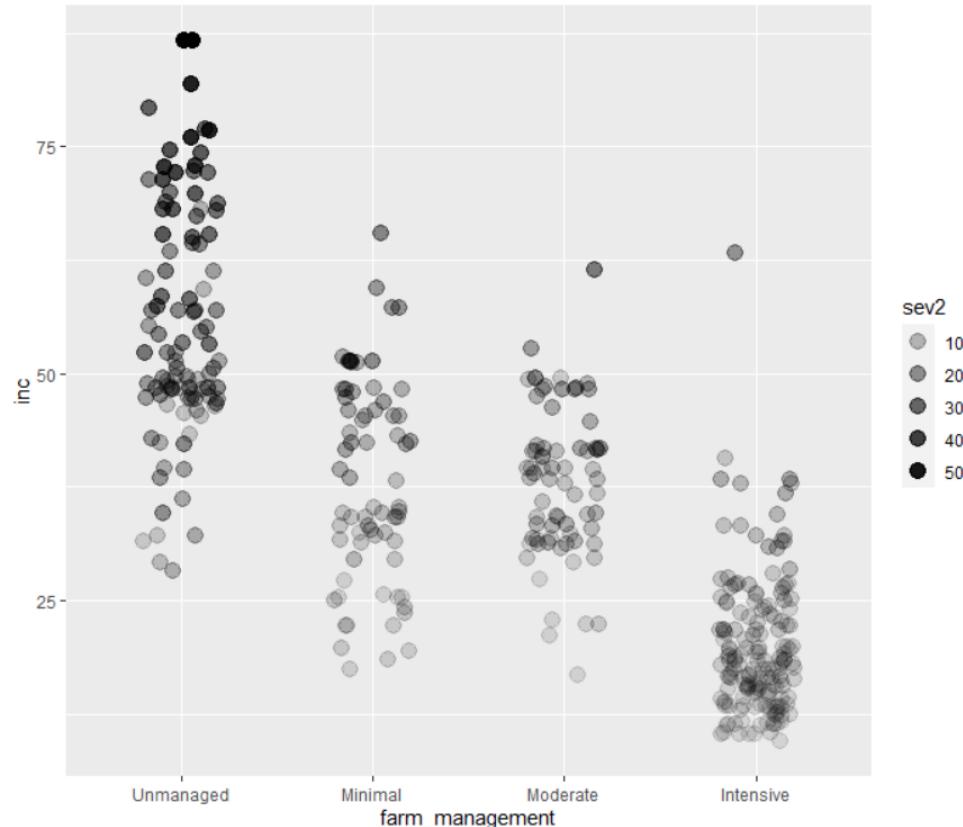
```
ggplot(survey_data, aes(x = farm_management, y = inc)) +  
  geom_jitter(width = 0.2, size = 4, shape = 21,  
              aes(fill = altitude ))
```



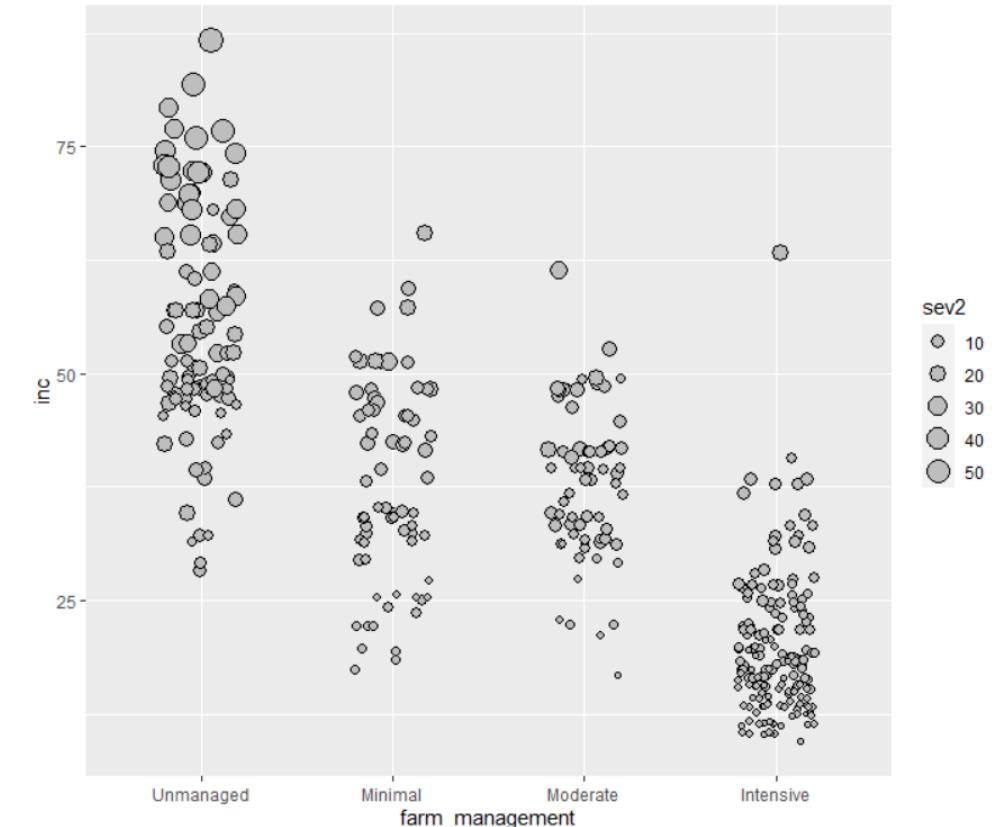


# Aesthetics – Alpha and size

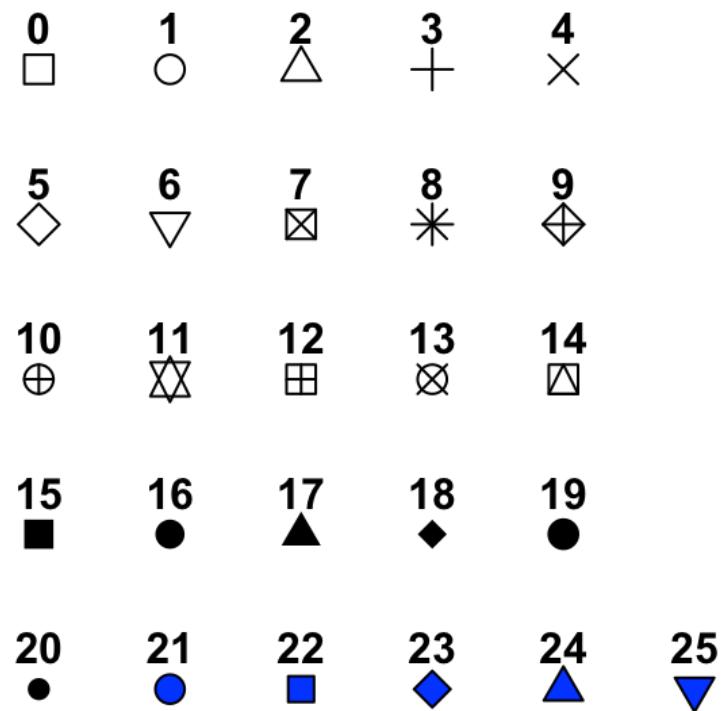
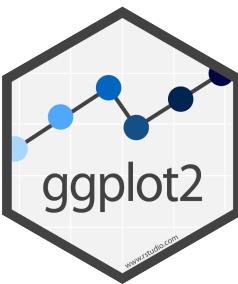
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter(width = 0.2, size = 4,  
              aes(alpha = sev2 ))
```



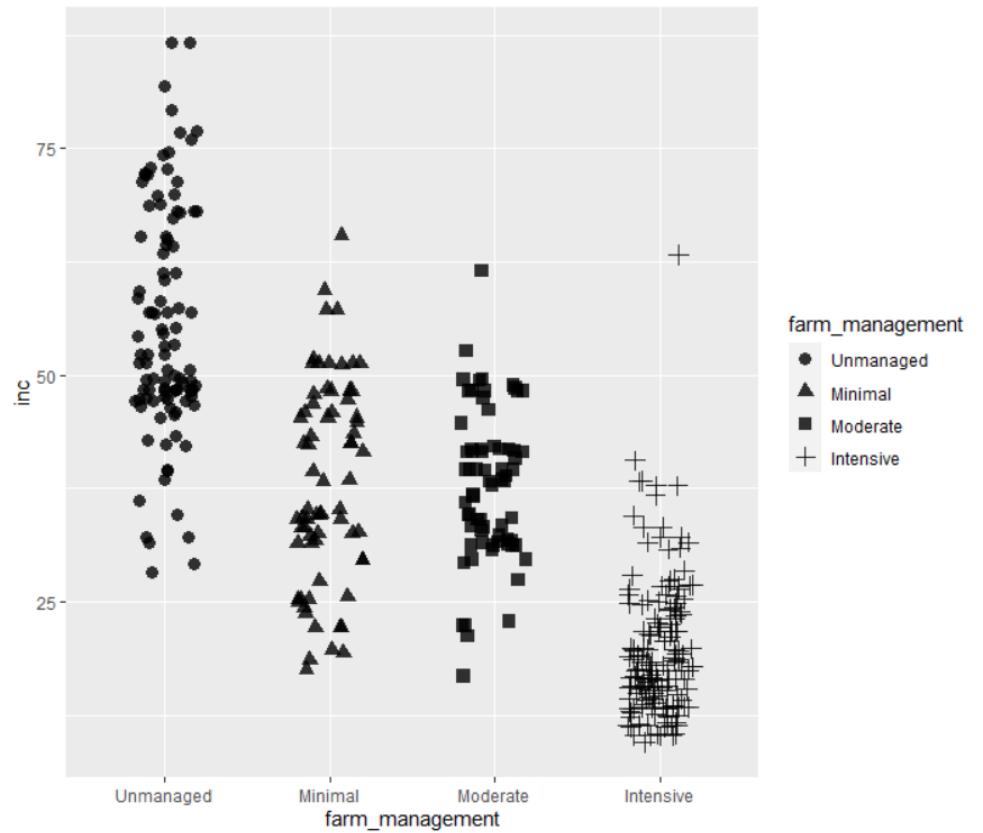
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter(width = 0.2, shape = 21, fill = "grey",  
              aes(size = sev2 ))
```



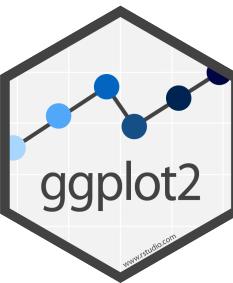
# Aesthetics - shape



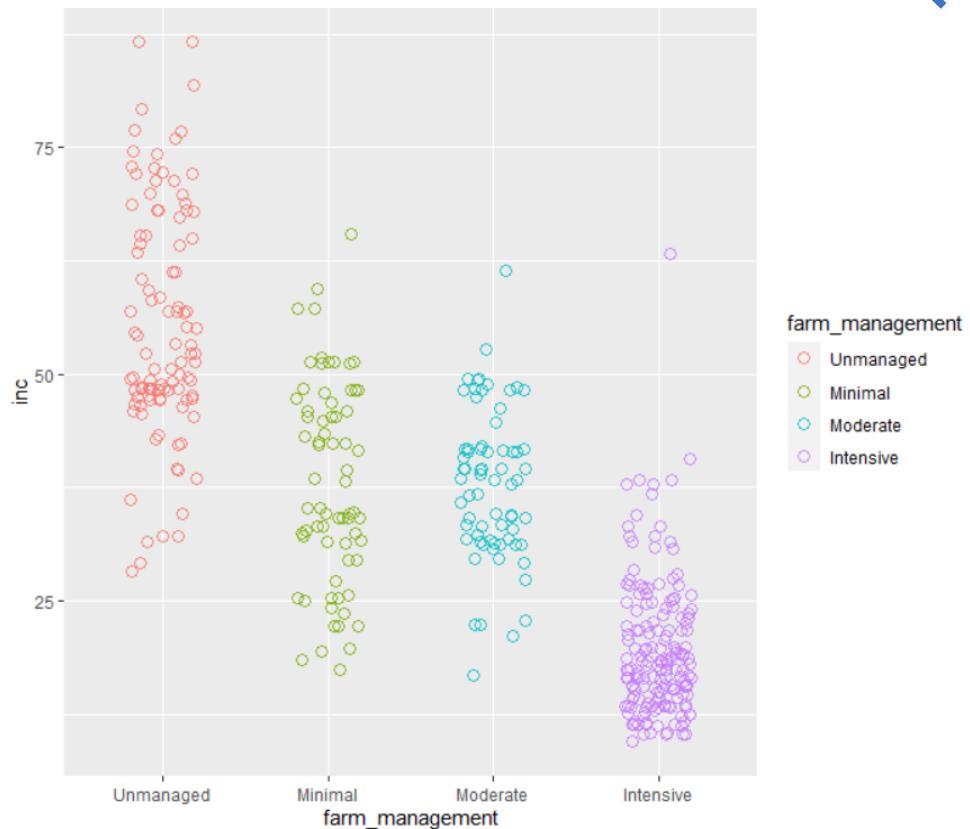
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter(width = 0.2, size = 3, alpha = 0.8,  
  aes(shape = farm_management))
```



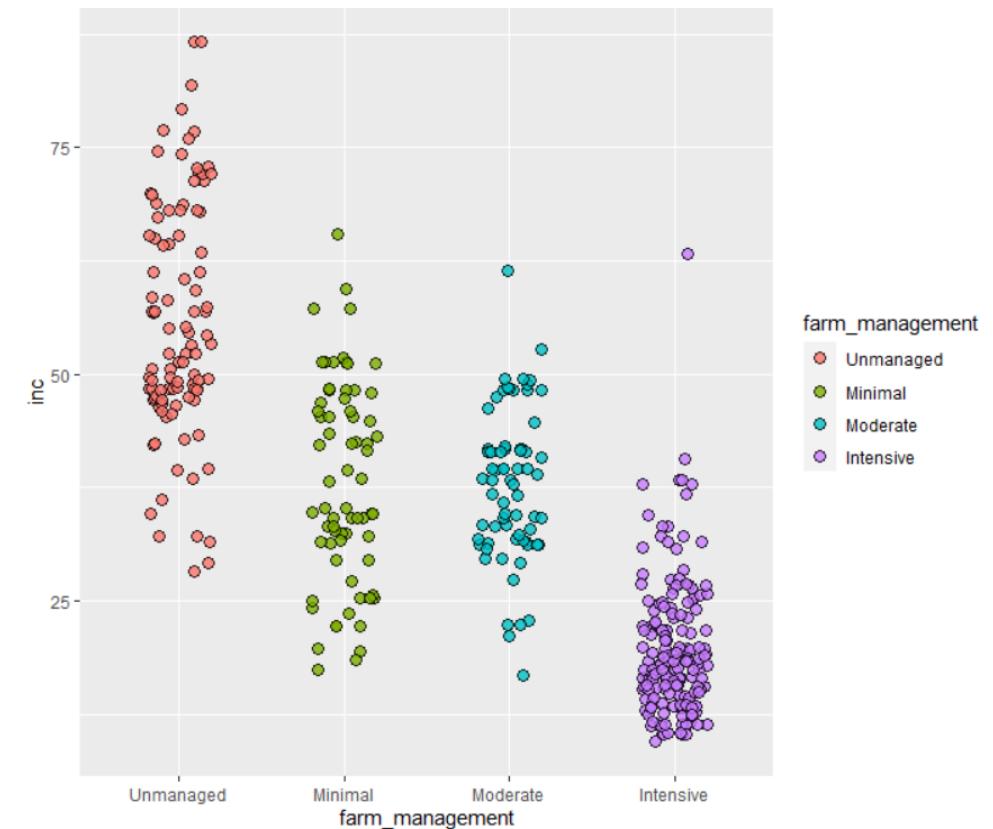
# Aesthetics



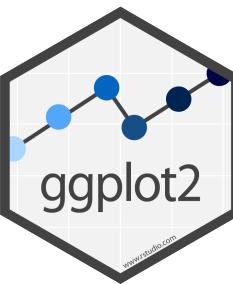
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter(width = 0.2, size = 3, alpha = 0.8, shape = 21,  
              aes(color = farm_management))
```



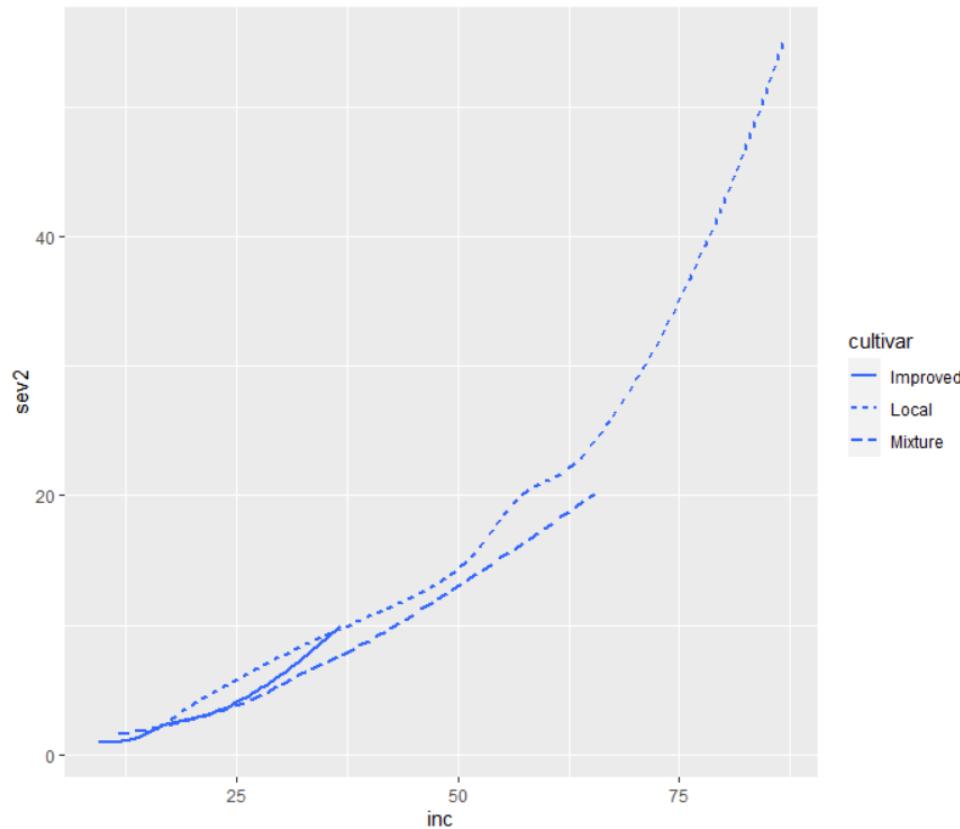
```
ggplot(survey_data, aes(x= farm_management, y = inc))+  
  geom_jitter(width = 0.2, size = 3, alpha = 0.8, shape = 21,  
              aes(fill = farm_management))
```



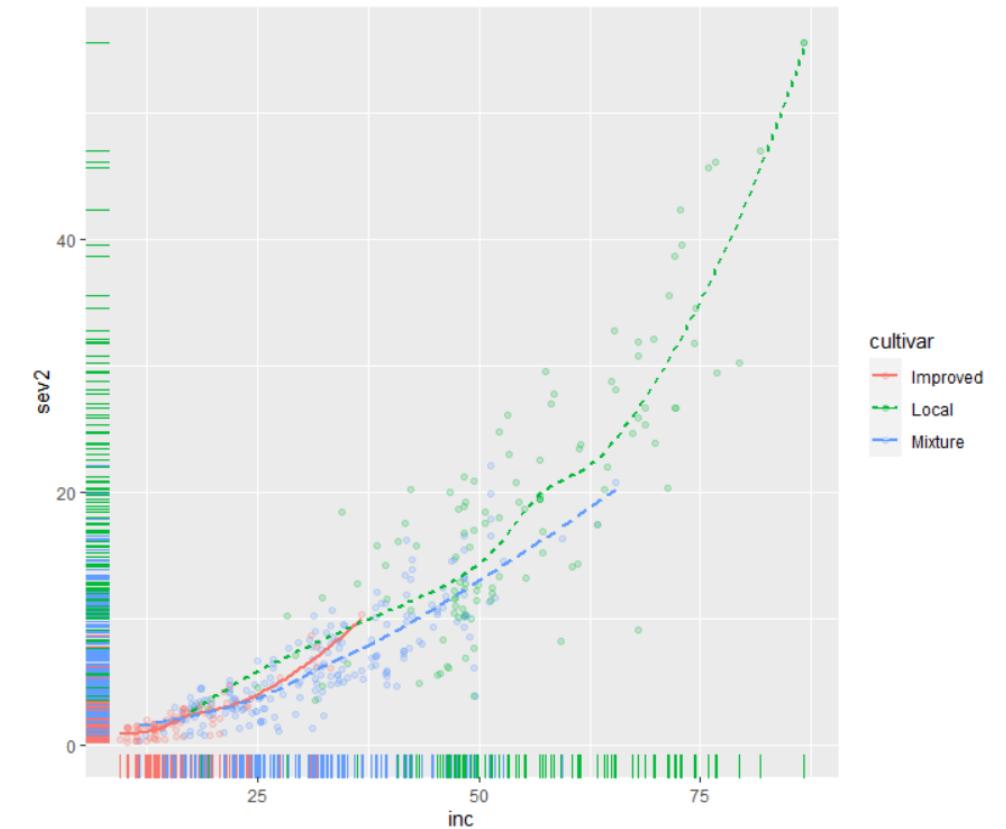
# Aesthetics

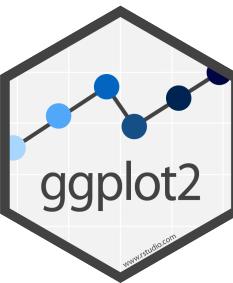


```
ggplot(survey_data, aes(x=inc , y = sev2))+  
  geom_smooth(aes(linetype = cultivar), se = FALSE)
```



```
ggplot(survey_data, aes(x=inc , y = sev2, color = cultivar))+  
  geom_point(alpha = 0.2)+  
  geom_smooth(aes(linetype = cultivar), se = FALSE)+  
  geom_rug()
```





# Scales

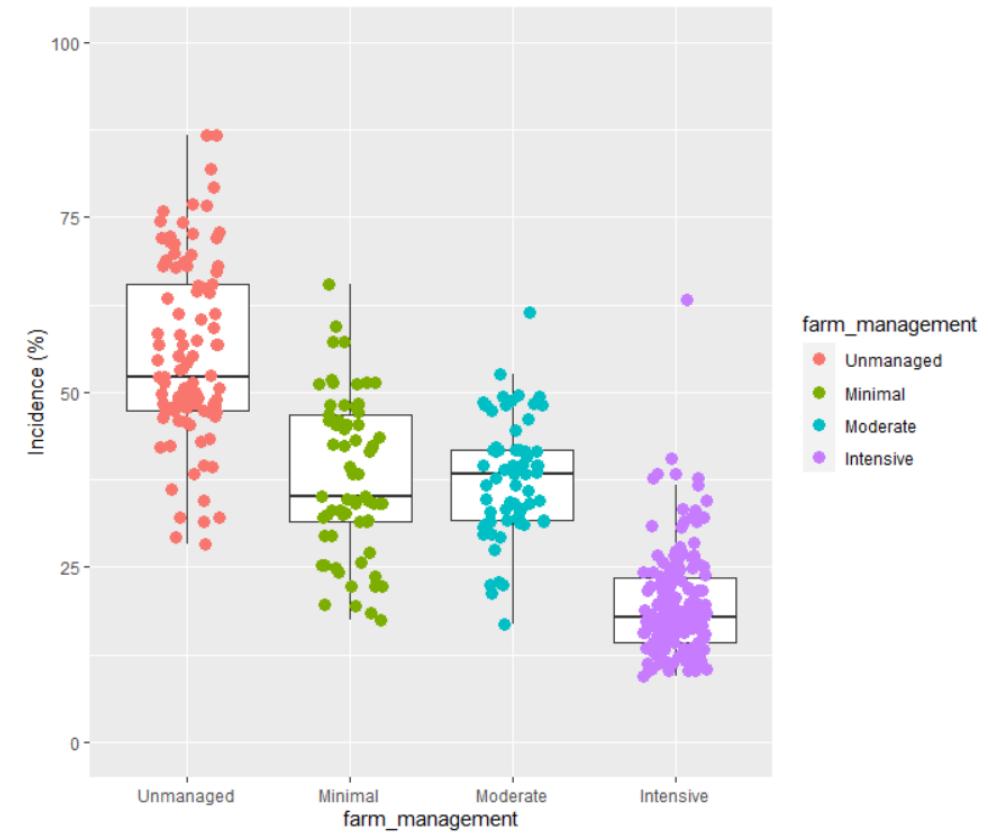
## → Control aesthetics mapping

### Most common

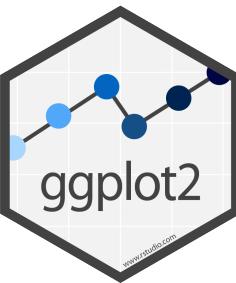
- `scale_*_continuous()`
- `scale_*_discrete()`
- `scale_*_manual()`
- `scale_*_date()`
- `scale_*_reverse()`
- `scale_*_log10()`
- `scale_*_gradient2()`

```
p = ggplot(survey_data, aes(x= farm_management, y = inc))+
  geom_boxplot(outlier.alpha = 0) +
  geom_jitter(aes(colour = farm_management),
              width = 0.2, size = 3)

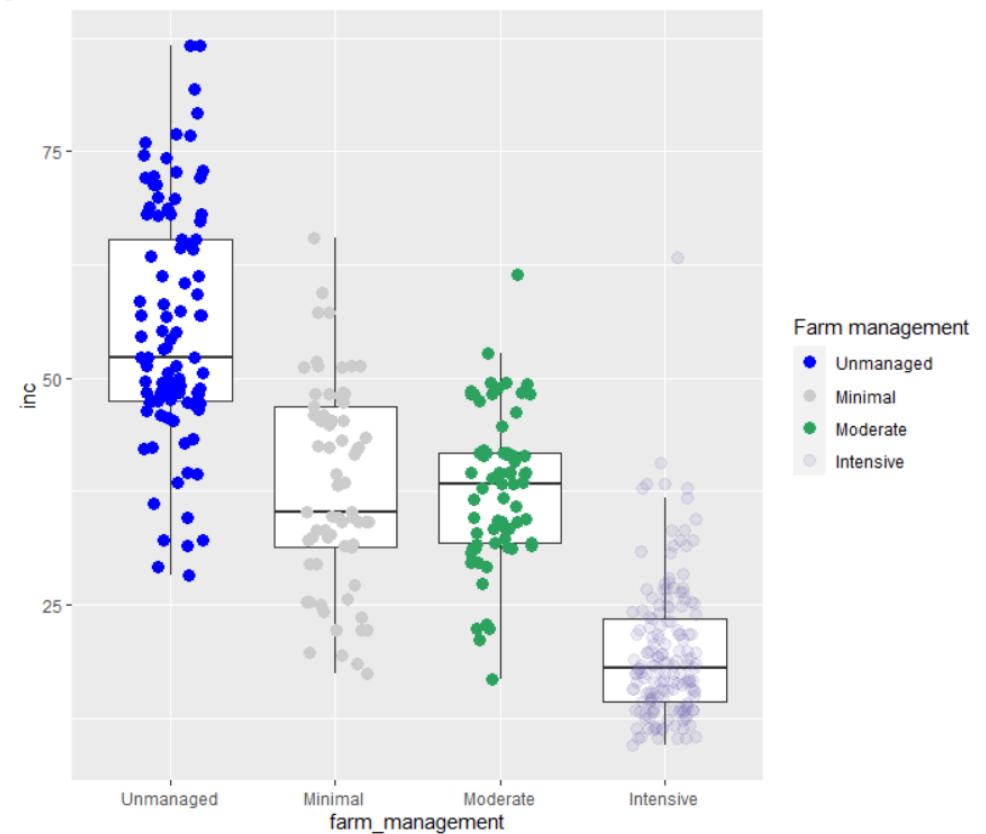
p + scale_y_continuous(name = "Incidence (%)",
                        breaks = c(0,25,50, 75, 100),
                        limits = c(0,100))
```



# Scales



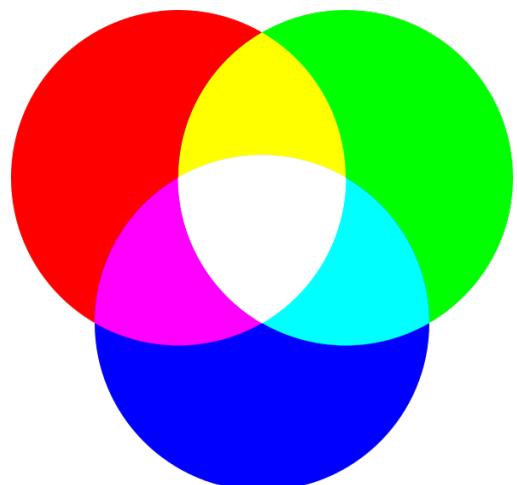
```
p + scale_color_manual(  
  name = "Farm management",  
  values = c("blue", "grey80", "#2ca25f", "#756bb120"))
```



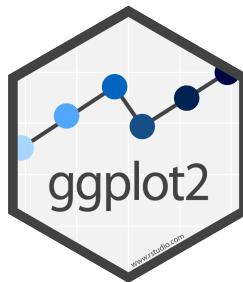
white
antiquewhite3
aquamarine3
azure3
bisque2
blue
blueviolet
brown4
burlywood4
cadetblue4
chartreuse4
chocolate4
coral4
cornsilk3
cyan3

grey1
grey7
grey13
grey19
grey25
grey31
grey37
grey43
grey49
grey55
grey61
grey67
grey73
grey79
grey85
grey91
grey97

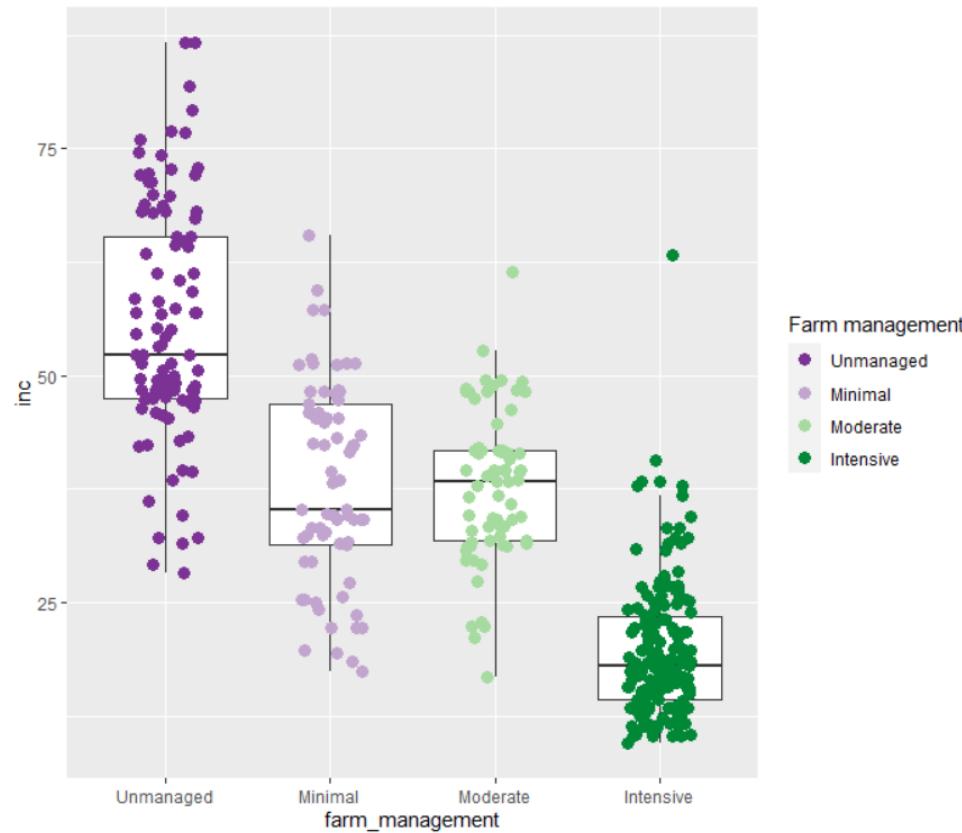
#RRGGBBAA



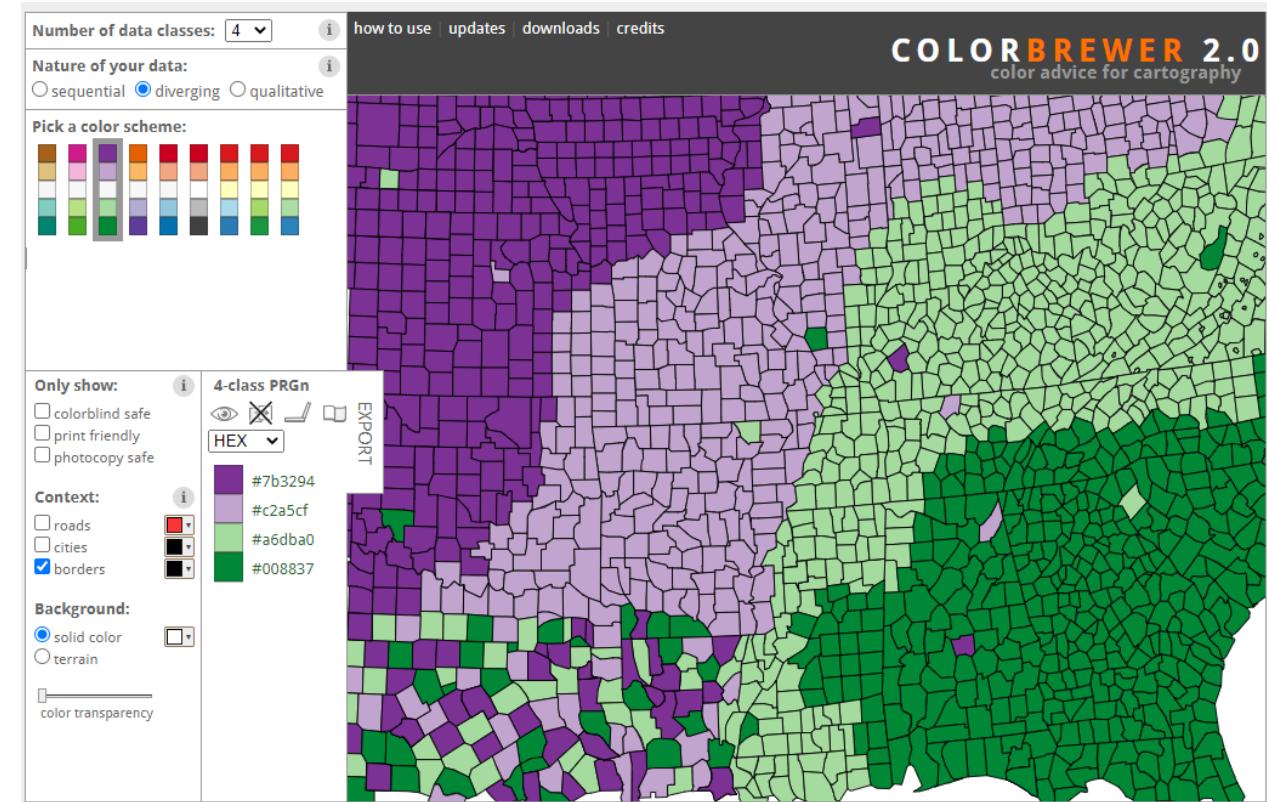
# Scales



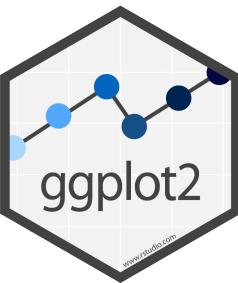
```
p + scale_color_brewer(name = "Farm management",  
                         palette = "PRGn")
```



[colorbrewer2.org](http://colorbrewer2.org)

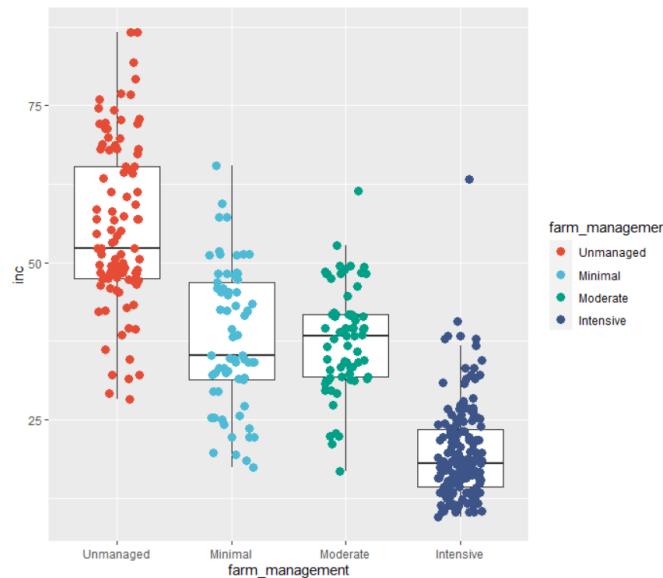


# Scales

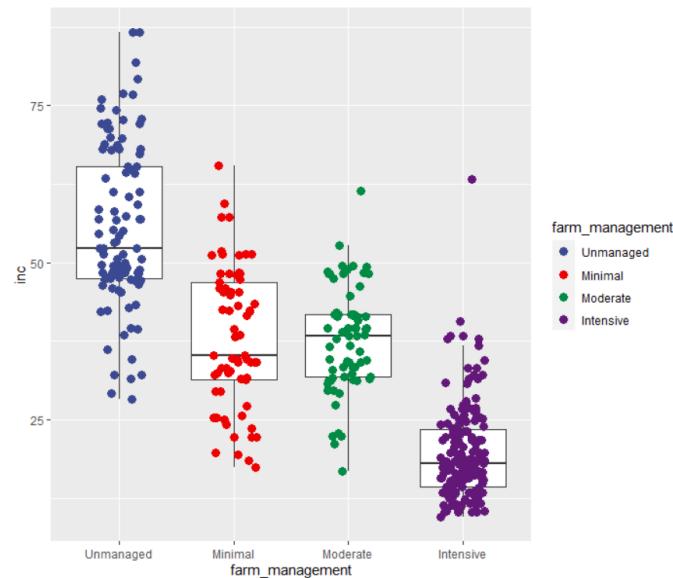


## ggsci package

`p + ggsci::scale_color_npg()`



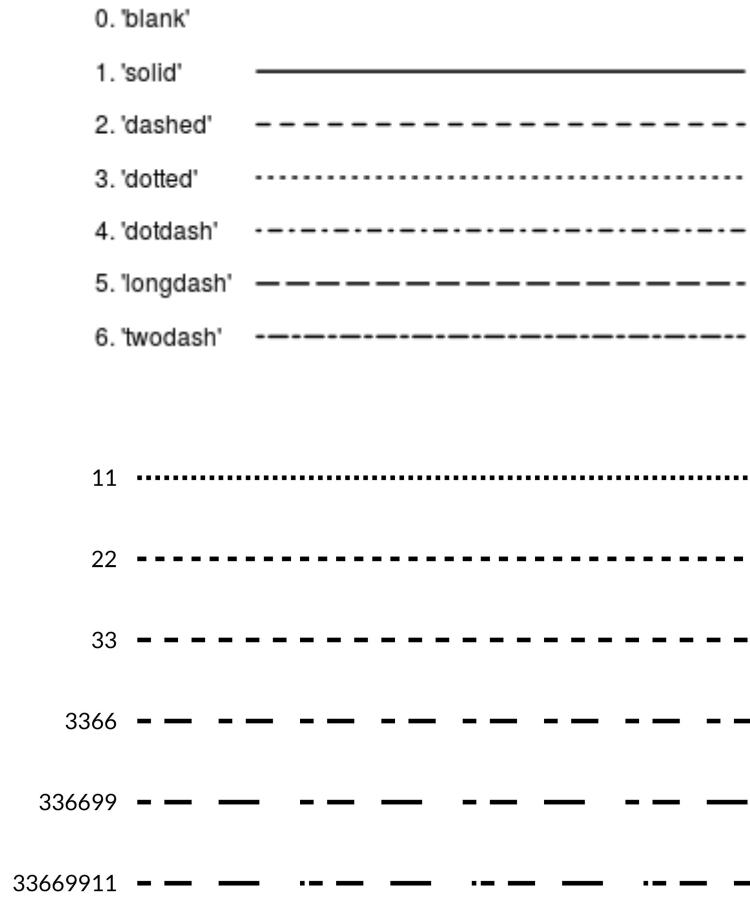
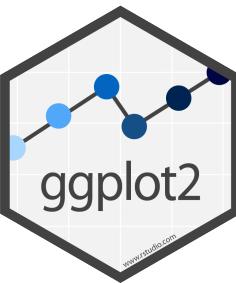
`p + ggsci::scale_color_aaas()`



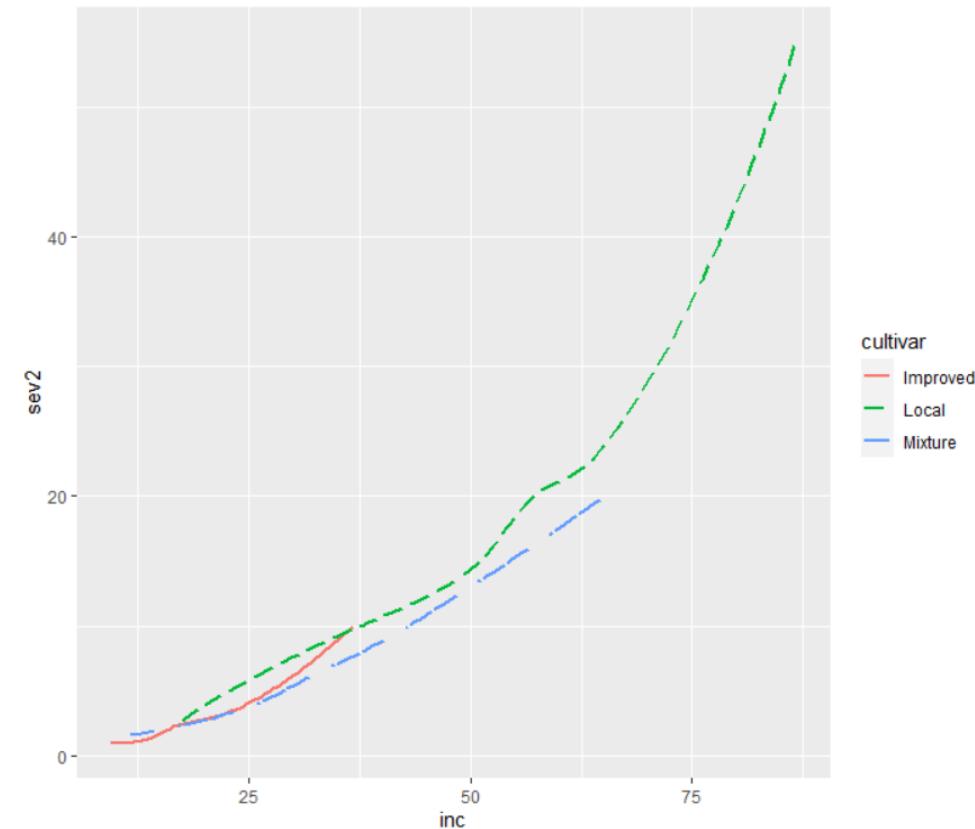
## inauguration package



# Scales

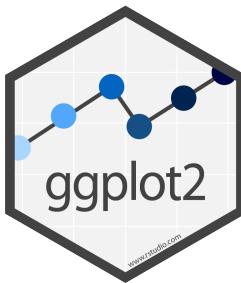


```
ggplot(survey_data, aes(x=inc , y = sev2))+  
  geom_smooth(se = FALSE, aes(linetype = cultivar, color = cultivar))+  
  # scale_linetype_manual(values = c(1,2,4)) +  
  scale_linetype_manual(values = c("solid","longdash", "991191"))
```



# Facets

## What if we want the same plot with severity side-by-side?



### Prepare data

```
> head(survey_data)
# A tibble: 6 x 7
  altitude cultivar shade    cropping_system farm_management   inc    sev2
  <dbl> <chr>    <chr>    <chr>        <fct>      <dbl>    <dbl>
1 1100 Local     Sun      Plantation Unmanaged       86.7   55.6
2 1342 Mixture   Mid shade Plantation Minimal        51.3   17.9
3 1434 Mixture   Mid shade Plantation Minimal        43.2   8.25
4 1100 Local     Sun      Plantation Unmanaged       76.7   46.1
5 1400 Local     Sun      Plantation Unmanaged       47.2   12.3
6 1342 Mixture   Mid shade Plantation Minimal        51.3   19.9
```

```
survey_data_long = survey_data %>%
  pivot_longer(cols = c(inc, sev2),
               names_to = "metric",
               values_to = "rate") %>%

  mutate(metric = factor(metric,
                         levels = c("inc", "sev2"),
                         labels = c("Incidence", "Severity")))

> head(survey_data_long)
# A tibble: 6 x 7
  altitude cultivar shade    cropping_system farm_management metric   rate
  <dbl> <chr>    <chr>    <chr>        <fct>      <fct>    <dbl>
1 1100 Local     Sun      Plantation Unmanaged   Incidence  86.7
2 1100 Local     Sun      Plantation Unmanaged   Severity   55.6
3 1342 Mixture   Mid shade Plantation Minimal    Incidence  51.3
4 1342 Mixture   Mid shade Plantation Minimal    Severity   17.9
5 1434 Mixture   Mid shade Plantation Minimal    Incidence  43.2
6 1434 Mixture   Mid shade Plantation Minimal    Severity   8.25
```

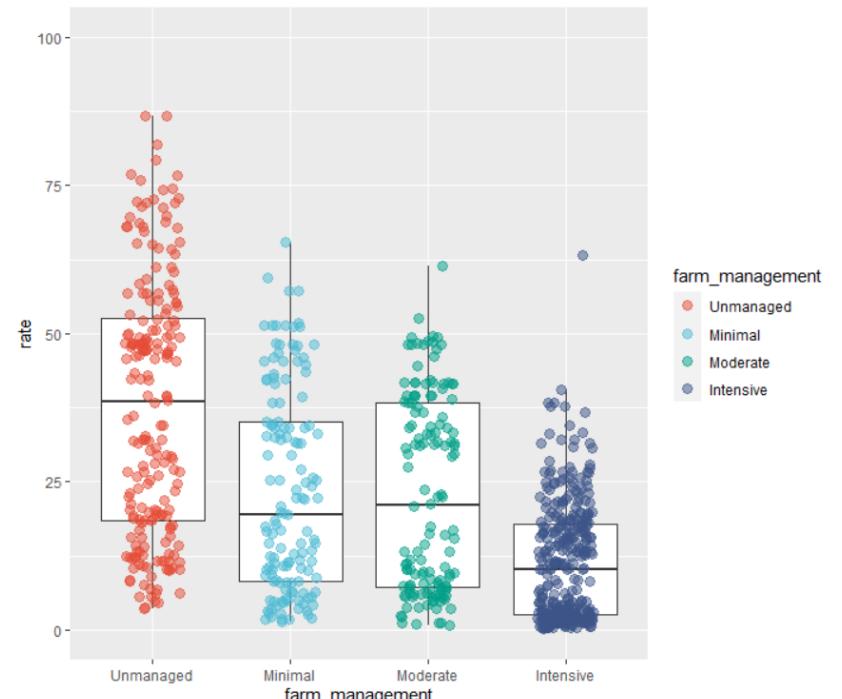
### Prepare a base plot

```
p2 = ggplot(survey_data_long, aes(x = farm_management, y = rate))+
  geom_boxplot(outlier.alpha = 0)+
  geom_jitter(aes(color = farm_management),
              size = 3, width = 0.2, alpha = .5)+
```

```
scale_y_continuous(breaks = seq(0,100,25),
                   limits = c(0,100))+
```

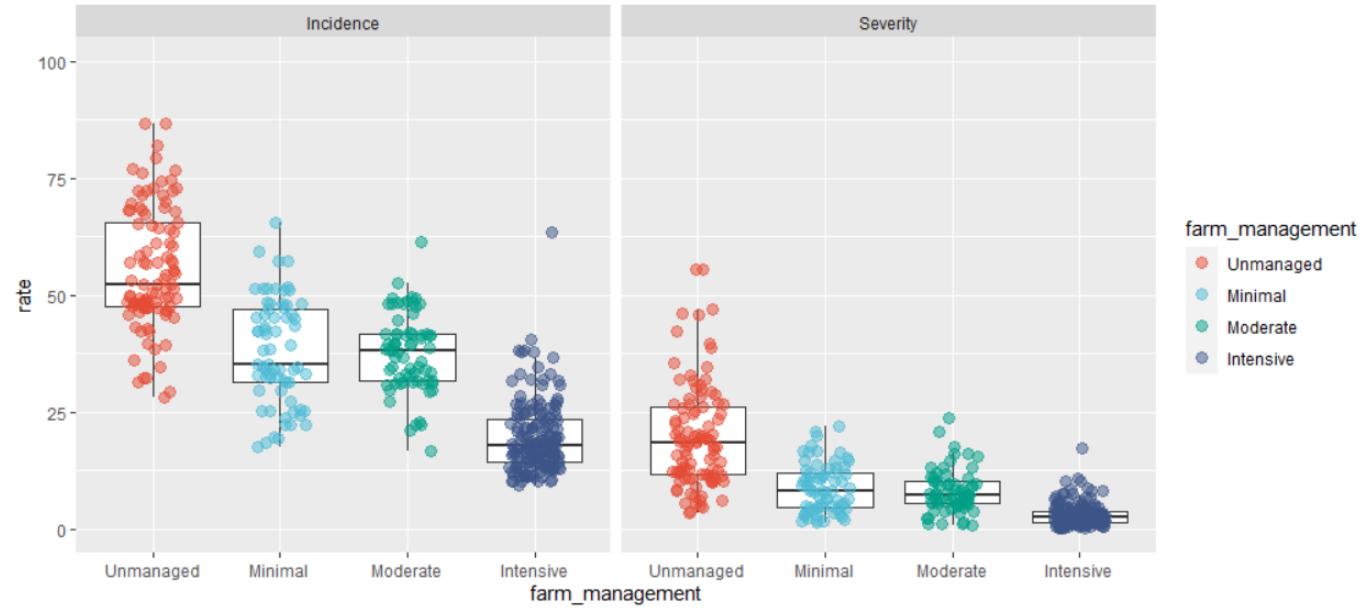
```
scale_color_npg()
```

```
p2
```

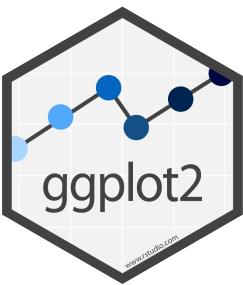
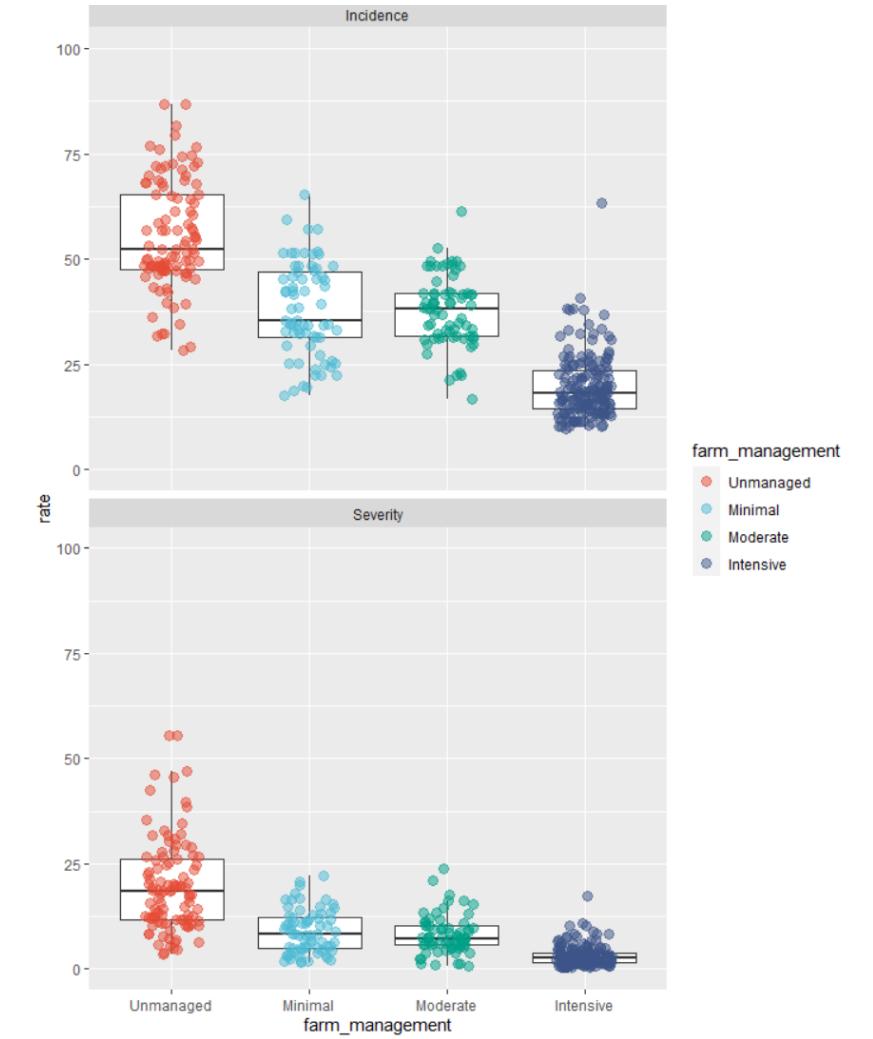


# Facets

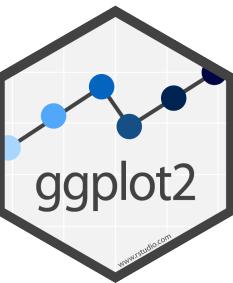
`p2 + facet_wrap(~metric)`



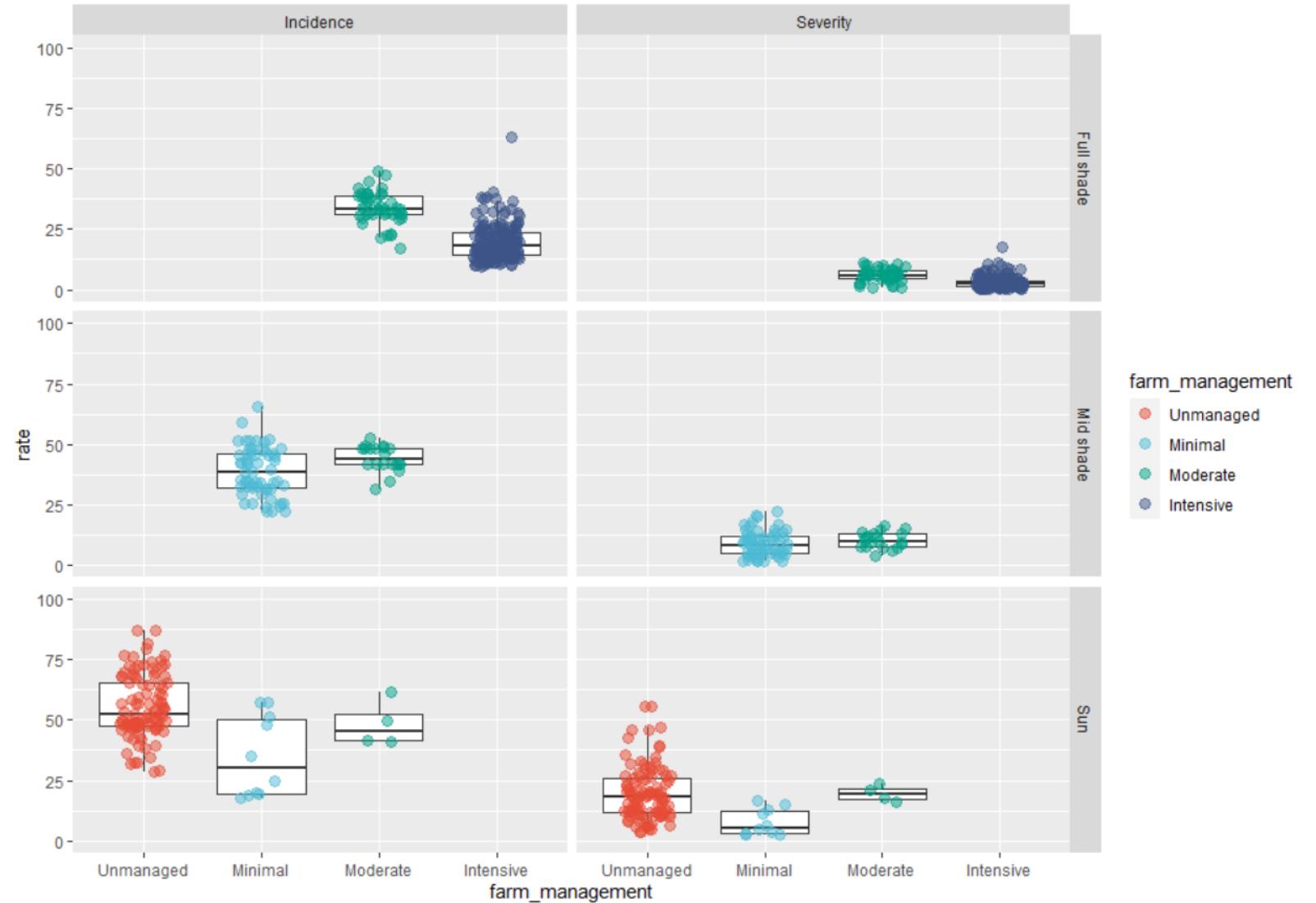
`p2 + facet_wrap(~metric, ncol = 1)`



# Facets

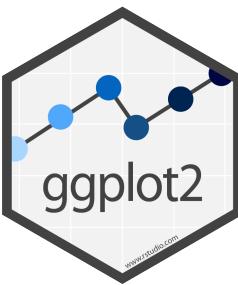
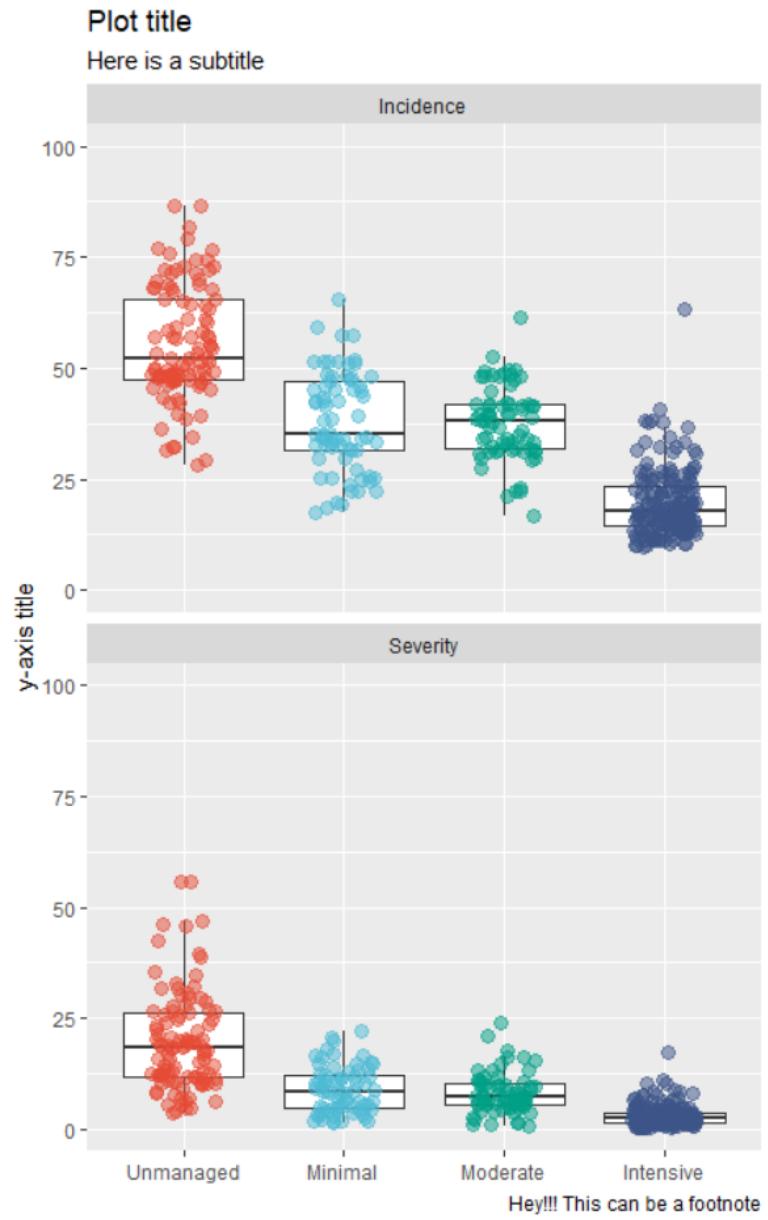


p2 + facet\_grid(shade~metric)

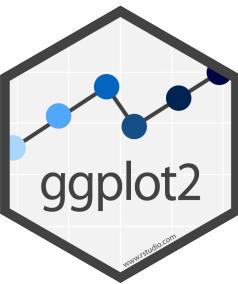


# Labels

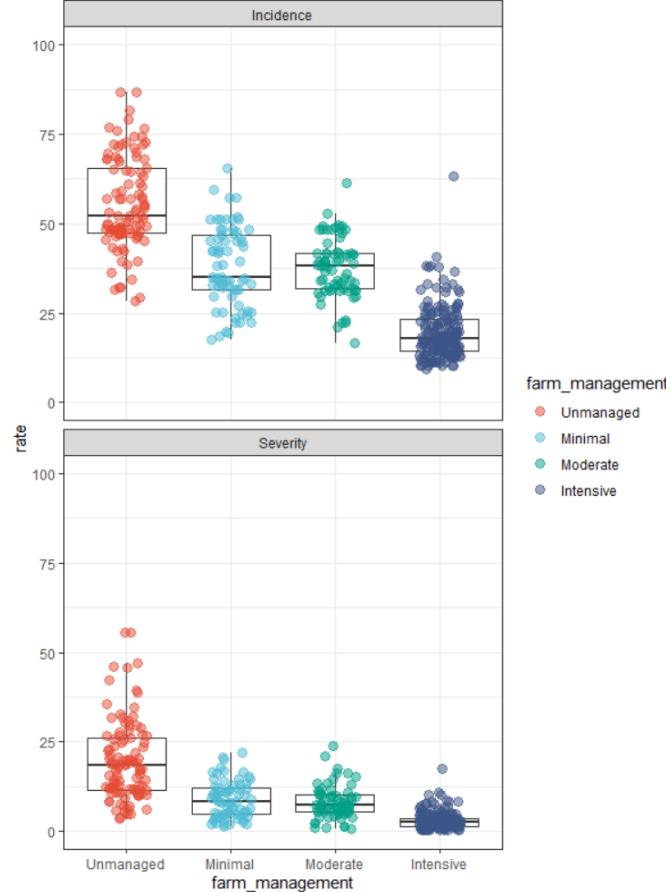
```
p2 + facet_wrap(~metric, ncol = 1)+  
  
  labs(title = "Plot title",  
        subtitle = "Here is a subtitle",  
        x = NULL, # suppress x-axis title  
        y = "y-axis title",  
        color = "Legend title",  
        caption = "Hey!!! This can be a footnote")
```



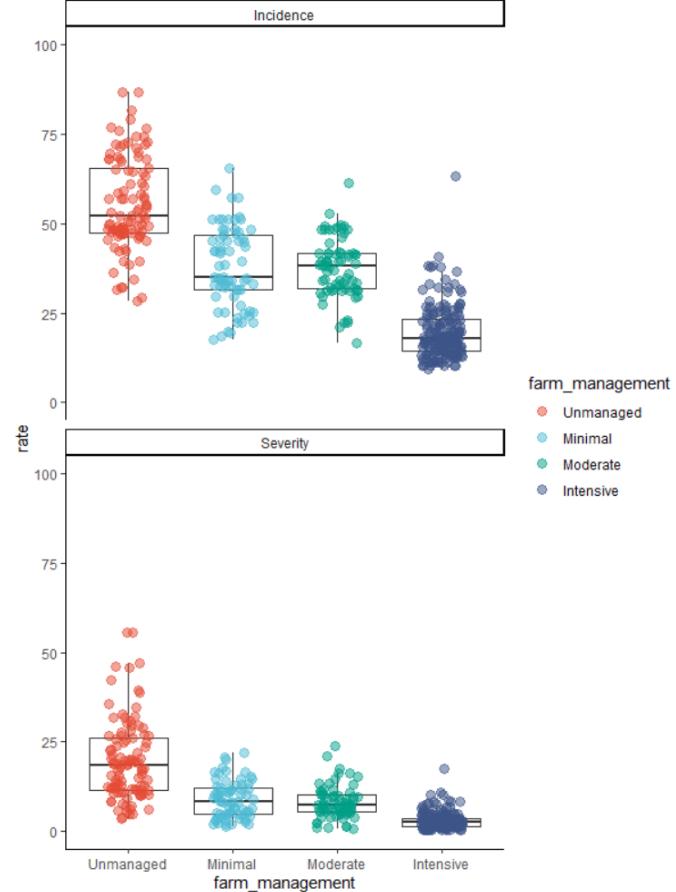
# Themes



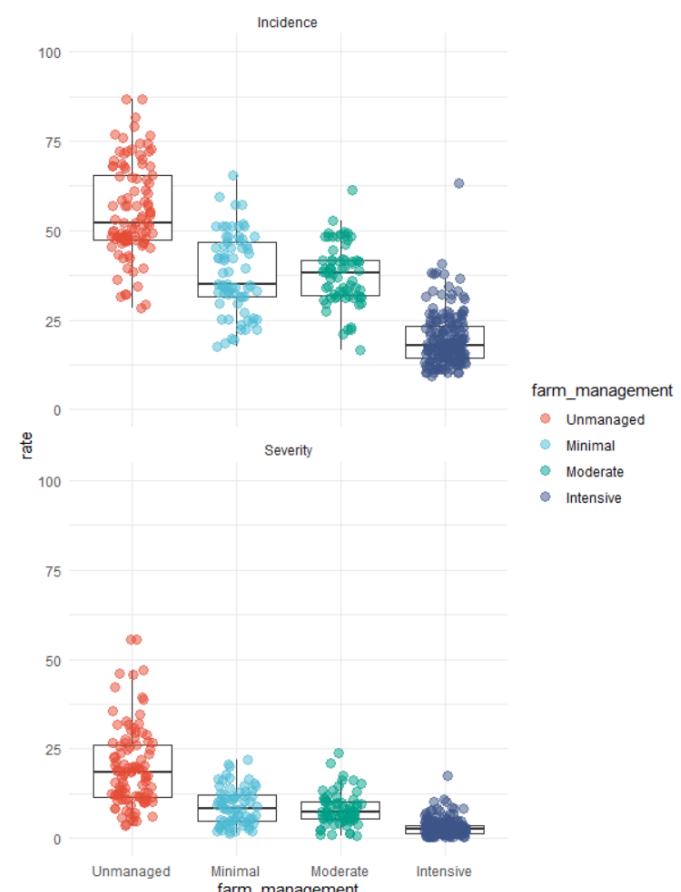
```
p2 + facet_wrap(~metric, ncol = 1)+  
  theme_bw()
```

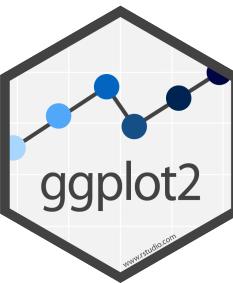


```
p2 + facet_wrap(~metric, ncol = 1)+  
  theme_classic()
```



```
p2 + facet_wrap(~metric, ncol = 1)+  
  theme_minimal()
```

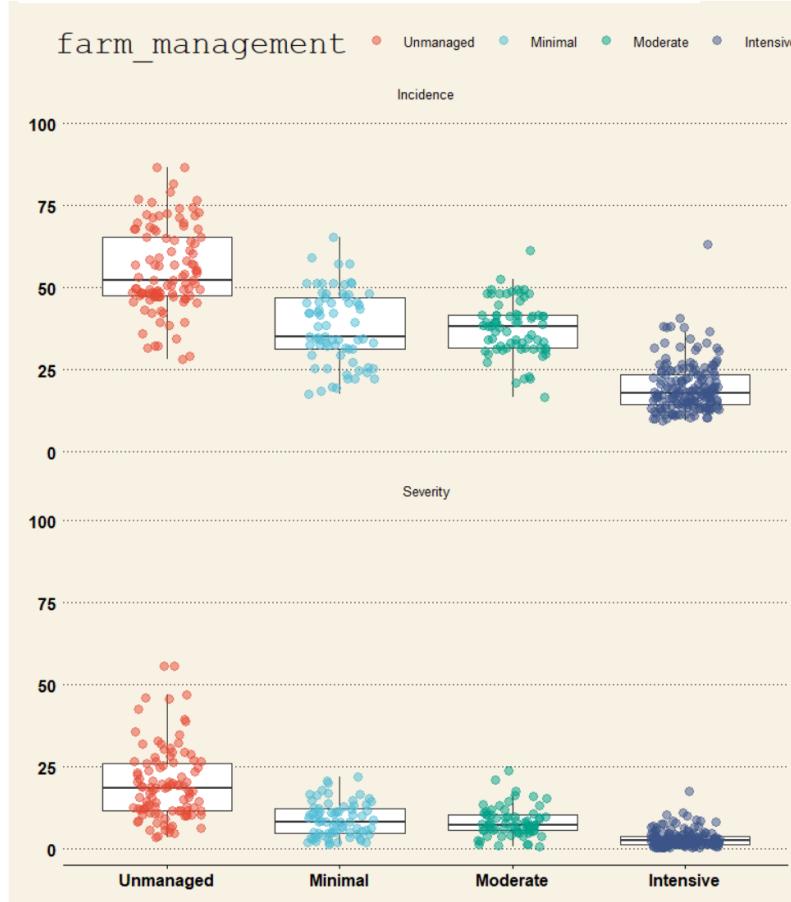




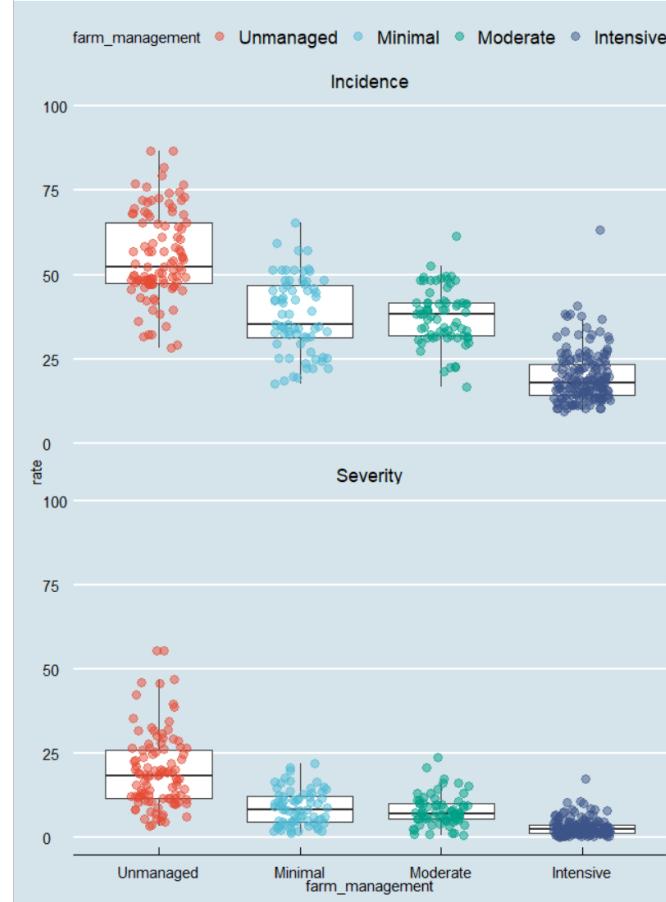
# Themes

## ggthemes package

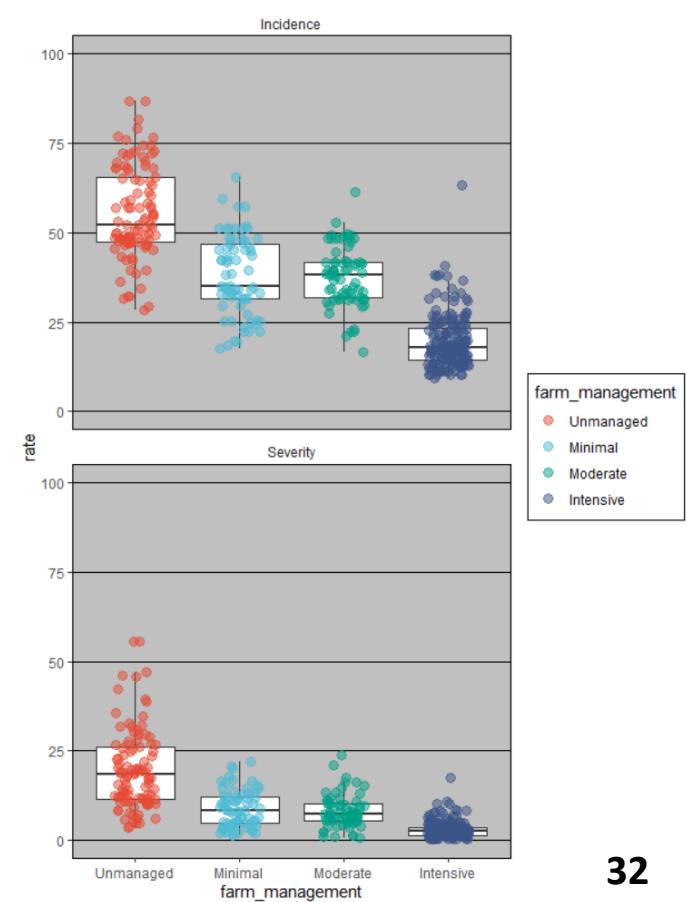
```
p2 + facet_wrap(~metric, ncol = 1) +
  ggthemes::theme_wsj()
```

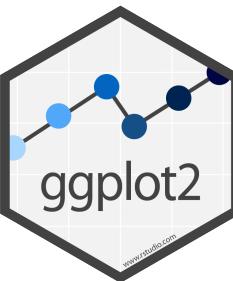


```
p2 + facet_wrap(~metric, ncol = 1) +
  ggthemes::theme_economist()
```



```
p2 + facet_wrap(~metric, ncol = 1) +
  ggthemes::theme_excel()
```

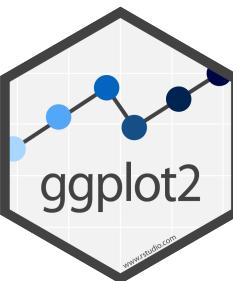




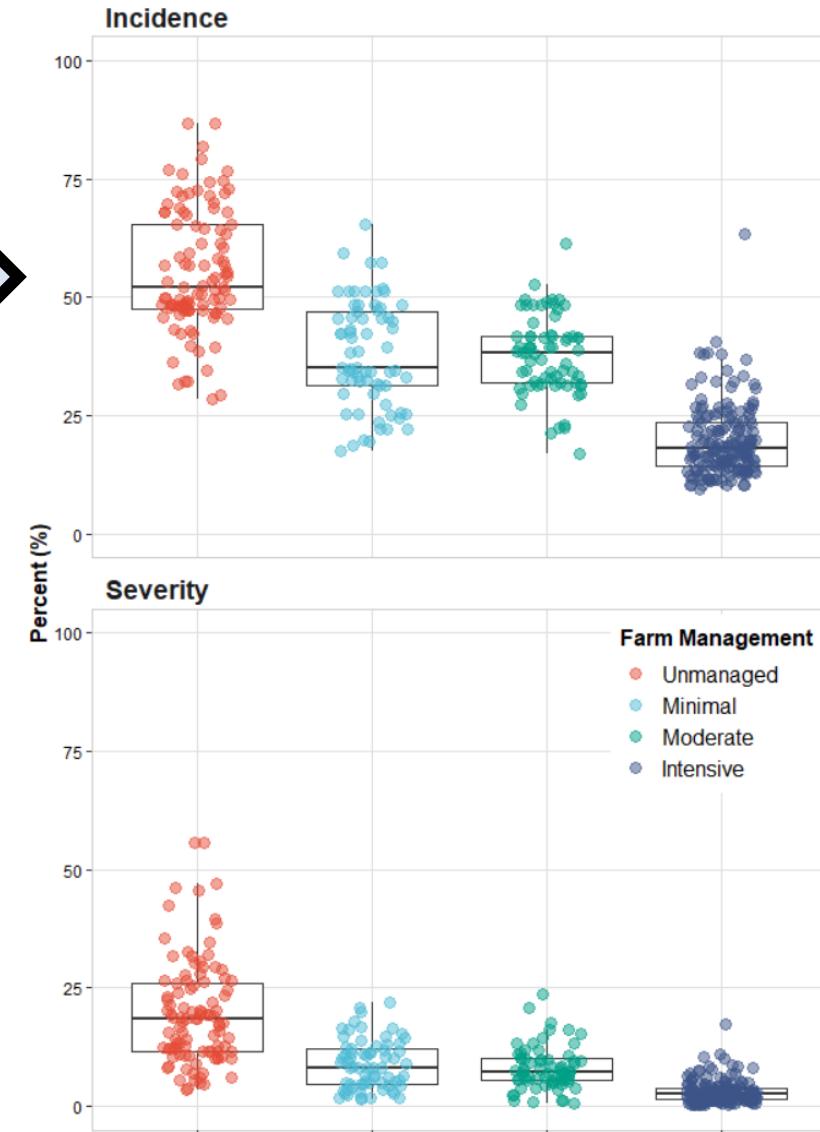
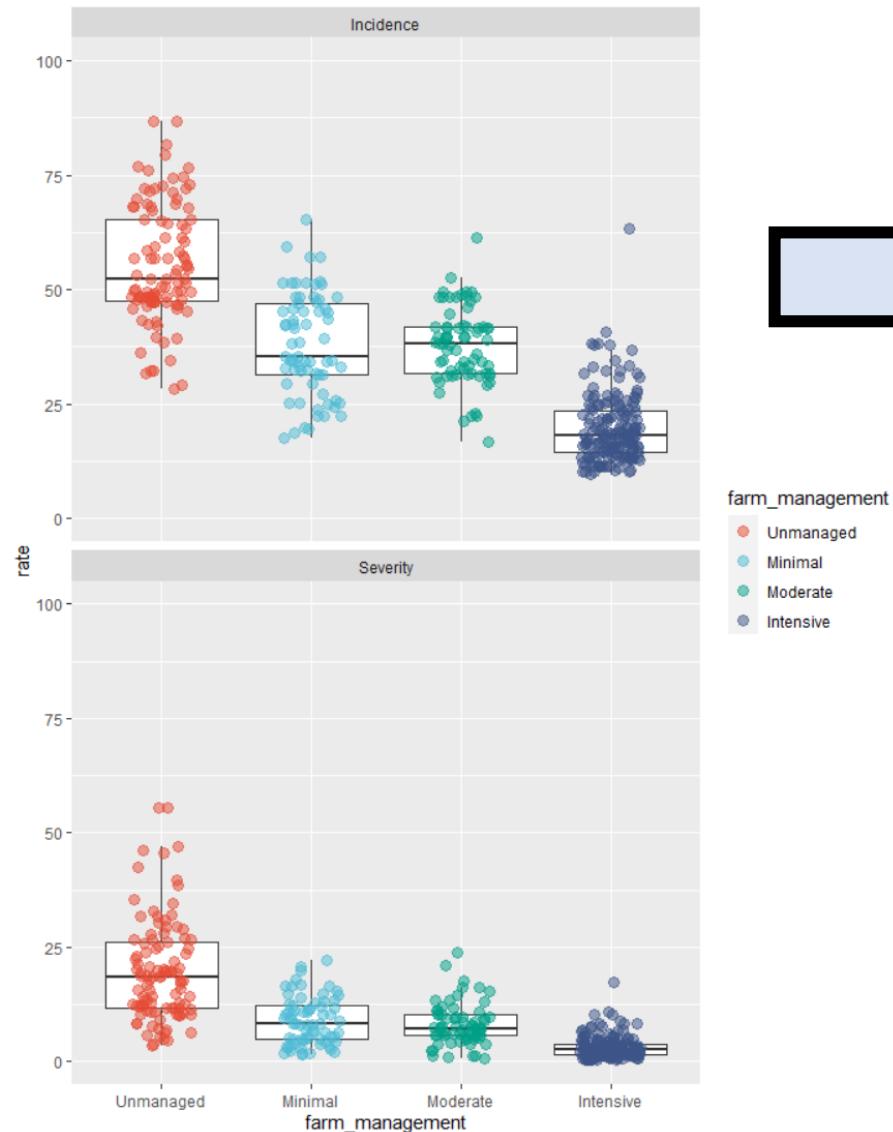
# Themes - customization

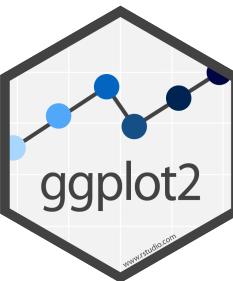
```
theme(  
  line,  
  rect,  
  text,  
  title,  
  aspect.ratio,  
  axis.title,  
  axis.title.x,  
  axis.title.x.top,  
  axis.title.x.bottom  
  axis.title.y,  
  axis.title.y.left,  
  axis.title.y.right,  
  axis.text,  
  axis.text.x,  
  axis.text.x.top,  
  axis.text.x.bottom,  
  axis.text.y,  
  axis.text.y.left,  
  axis.text.y.right,  
  axis.ticks,  
  axis.ticks.x,  
  axis.ticks.x.top,  
  axis.ticks.x.bottom,  
  axis.ticks.y,  
  axis.ticks.y.left,  
  axis.ticks.y.right,  
  axis.ticks.length,  
  axis.ticks.length.x,  
  axis.ticks.length.x.top,  
  axis.ticks.length.x.bottom,  
  axis.ticks.length.y,  
  axis.ticks.length.y.left,  
  axis.ticks.length.y.right,  
  axis.line,  
  axis.line.x,  
  axis.line.x.top,  
  axis.line.x.bottom,  
  axis.line.y,  
  axis.line.y.left,  
  axis.line.y.right,  
  legend.background,  
  legend.margin,  
  legend.spacing,  
  legend.spacing.x,  
  legend.spacing.y,  
  legend.key,  
  legend.key.size,  
  legend.key.height,  
  legend.key.width,  
  legend.text,  
  legend.text.align,  
  legend.title,  
  legend.title.align,  
  legend.position,  
  legend.direction,  
  legend.justification,  
  legend.box,  
  legend.box.just,  
  legend.box.margin,  
  legend.box.background,  
  legend.box.spacing,  
  panel.background,  
  panel.border,  
  panel.spacing,  
  panel.spacing.x,  
  panel.spacing.y,  
  panel.grid,  
  panel.grid.major,  
  panel.grid.minor,  
  panel.grid.major.x,  
  panel.grid.major.y,  
  panel.grid.minor.x,  
  panel.on top,  
  plot.background,  
  plot.title,  
  plot.title.position,  
  plot.subtitle,  
  plot.caption,  
  plot.caption.position,  
  plot.tag,  
  plot.tag.position,  
  plot.margin,  
  strip.background,  
  strip.background.x,  
  strip.background.y,  
  strip.placement,  
  strip.text,  
  strip.text.x,  
  strip.text.y,  
  strip.switch.pad.grid,  
  strip.switch.pad.wrap,  
  ...,  
  complete = FALSE,  
  validate = TRUE  
)
```

The options are countless



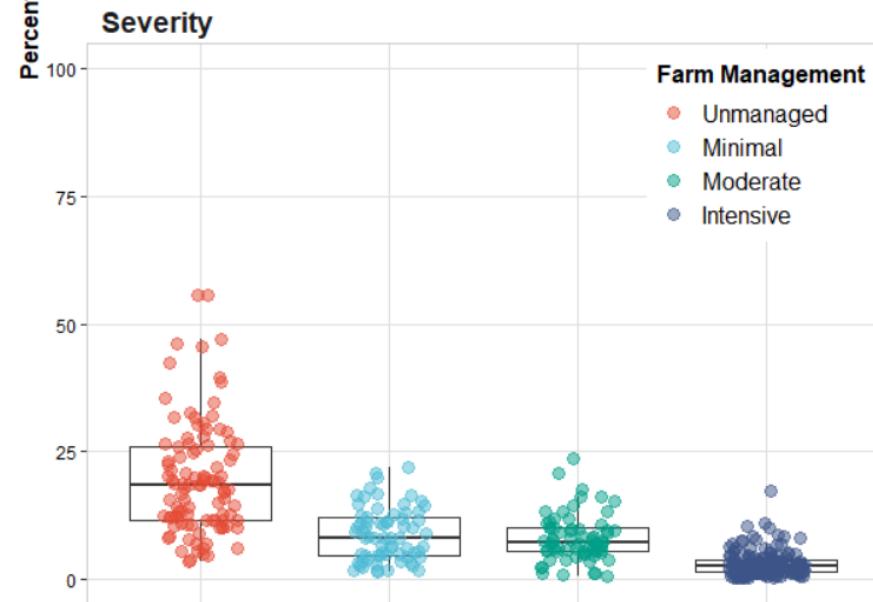
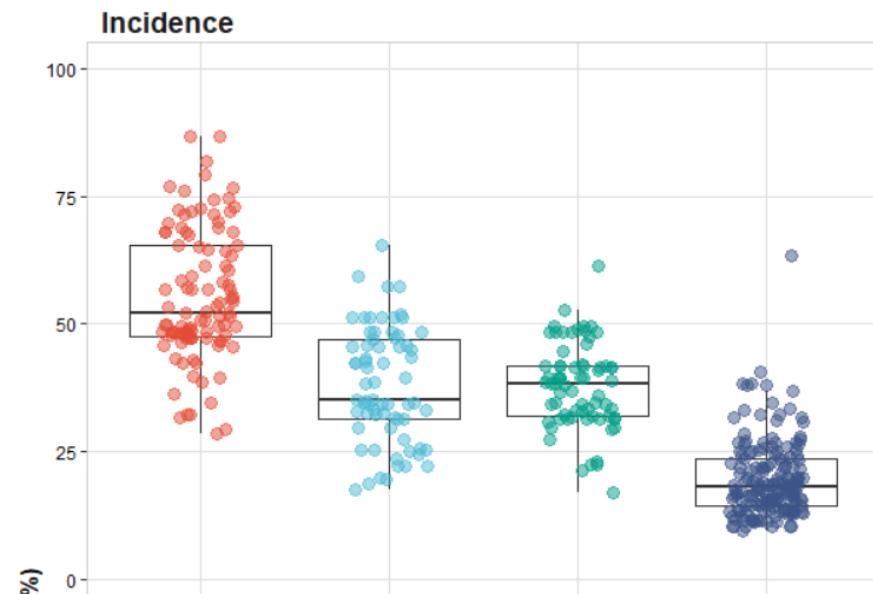
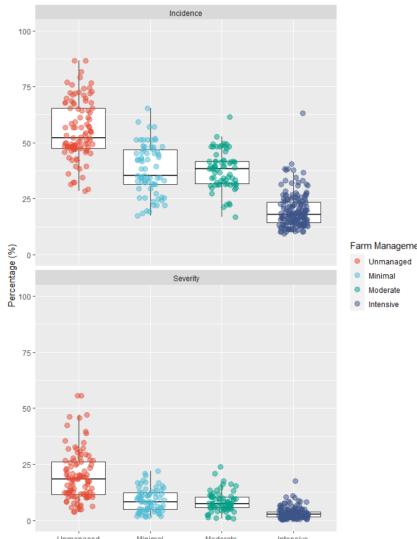
# Themes - customization





# Themes - customization

```
p2 + facet_wrap(~metric, ncol = 1) +
  labs(x = NULL, y = "Percent (%)", color = "Farm Management") +
  theme(
    panel.background = element_blank(),
    panel.grid.major = element_line(colour = "grey88"),
    panel.border = element_rect(colour = "grey80", fill = NA),
    axis.text.y = element_text(colour = "black"),
    axis.text.x = element_blank(),
    title = element_text(colour = "black", size = 12, face = "bold"),
    legend.position = c(0.85, .39),
    legend.key = element_rect(fill = "white"),
    legend.text = element_text(colour = "black", size = 12),
    strip.text.x = element_text(hjust = 0.01, face = "bold", size = 14),
    strip.background = element_blank())
```



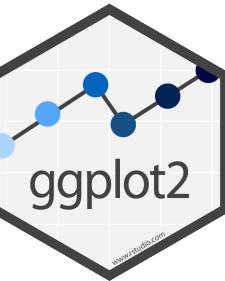
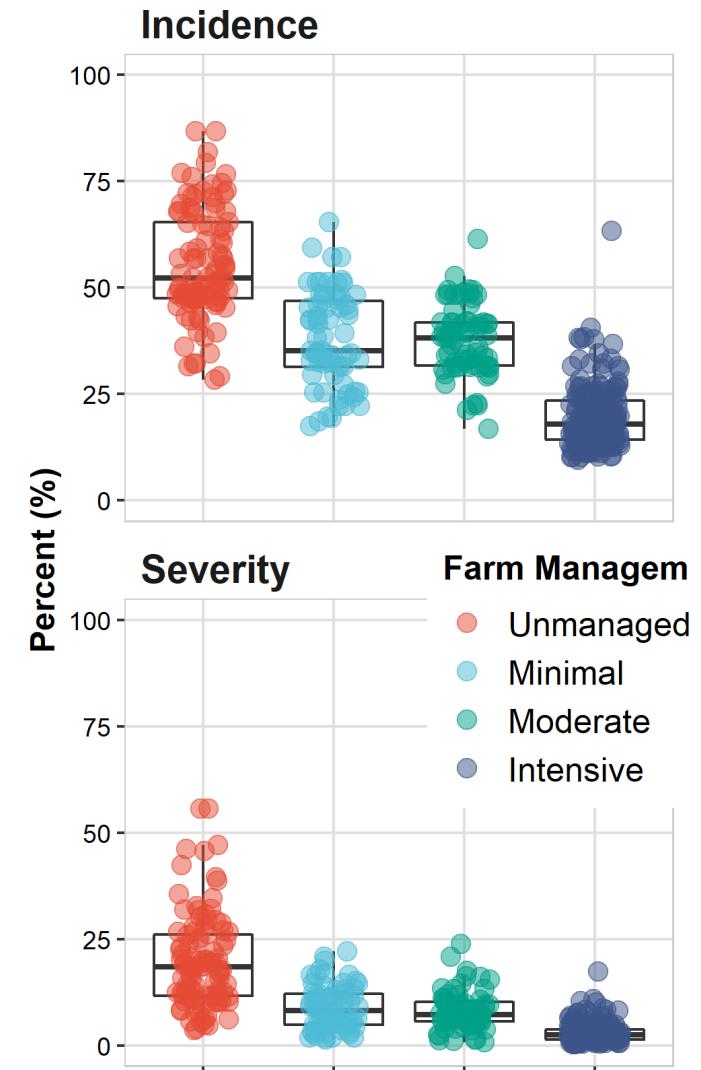
# Save



## File Specifications

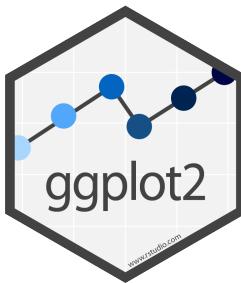
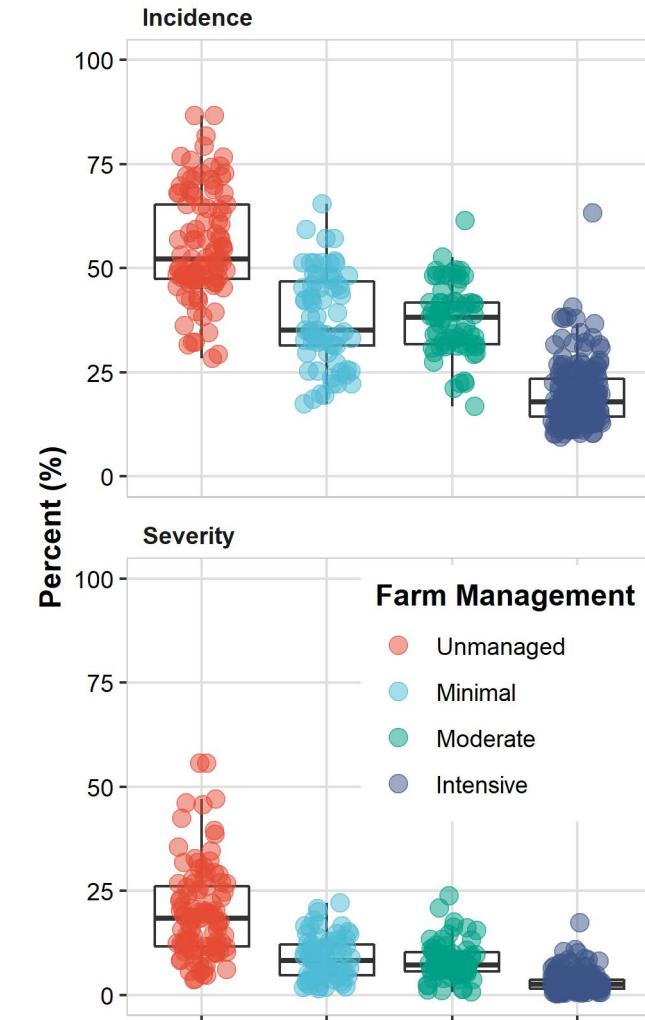
- **Size:** Figures should be sized at the final production size/resolution when possible.
  - 1 column: 3.25 inches (85 mm) Minimum 1,050 pixels wide
  - 2 column: 7 inches (178 mm) Minimum 2,100 pixels wide
  - The maximum depth of a figure is 9.7 inches (246 mm)
- **Format:** Figure file formats accepted: .tif, .jpg, or .eps are preferred. When saving TIFF files, use only LZW compression; do not use JPEG compression. All images must be flattened, layered images are not supported. If images are submitted in color, they should be submitted in RGB format.
- If high-resolution image files cannot be provided in the formats listed above, original image files generated with MS Office programs (such as Word, PowerPoint, or Excel) can be submitted.

```
ggsave(filename = "graph/APS_plot.tiff",
        width = 85, height = 140, units = "mm",
        dpi = 300)
```



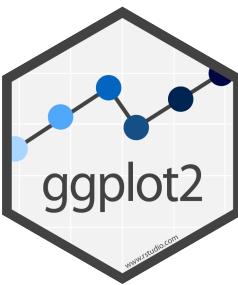
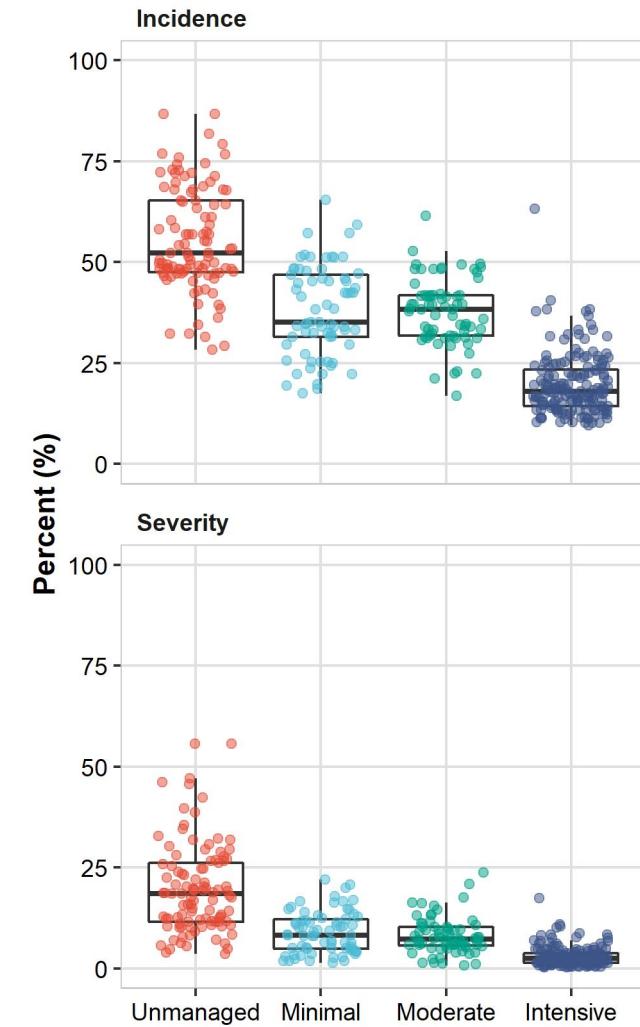
# Save

```
p3 =  
p2 + facet_wrap(~metric, ncol = 1)+  
  labs(x = NULL, y = "Percent (%)", color = "Farm Management") +  
  
  theme(  
    panel.background = element_blank() ,  
    panel.grid.major = element_line(colour = "grey88") ,  
    panel.border      = element_rect(colour = "grey80", fill = NA) ,  
  
    axis.text.y   = element_text(colour = "black") ,  
    axis.text.x   = element_blank() ,  
    title = element_text(colour = "black", face = "bold") ,  
  
    legend.position = c(0.72, .33) ,  
    legend.key = element_rect(fill = "white") ,  
    legend.text = element_text(colour = "black") ,  
  
    strip.text.x = element_text(hjust =0.01, face = "bold") ,  
    strip.background = element_blank())  
  
ggsave(plot = p3, filename = "graph/APS_plot_2.tiff" ,  
       width =85, height = 140, units = "mm" ,  
       dpi = 300)
```

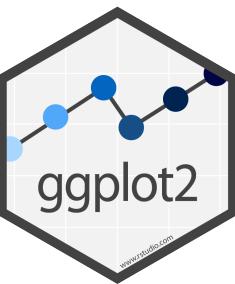


# Save

```
p4 =  
  ggplot(survey_data_long, aes(x= farm_management, y = rate))+  
  
  geom_boxplot(outlier.alpha = 0)+  
  geom_jitter(aes(color = farm_management),  
              size = 1.5, width = 0.3, alpha = .5)+  
  
  scale_y_continuous(breaks = seq(0,100,25),  
                     limits = c(0,100))+  
  scale_color_npg()  
  
  facet_wrap(~metric, ncol = 1)+  
  labs(x = NULL, y = "Percent (%)")  
  
  theme(  
    panel.background = element_blank() ,  
    panel.grid.major = element_line(colour = "grey88") ,  
    panel.border      = element_rect(colour = "grey80", fill = NA) ,  
  
    axis.text   = element_text(colour = "black") ,  
    title       = element_text(colour = "black", face = "bold") ,  
  
    legend.position = "none" ,  
  
    strip.text.x = element_text(hjust =0.01, face = "bold") ,  
    strip.background = element_blank())  
  
  ggsave(plot = p4, filename = "graph/APS_plot_3.tiff",  
         width =85, height = 140, units = "mm",  
         dpi = 300)
```



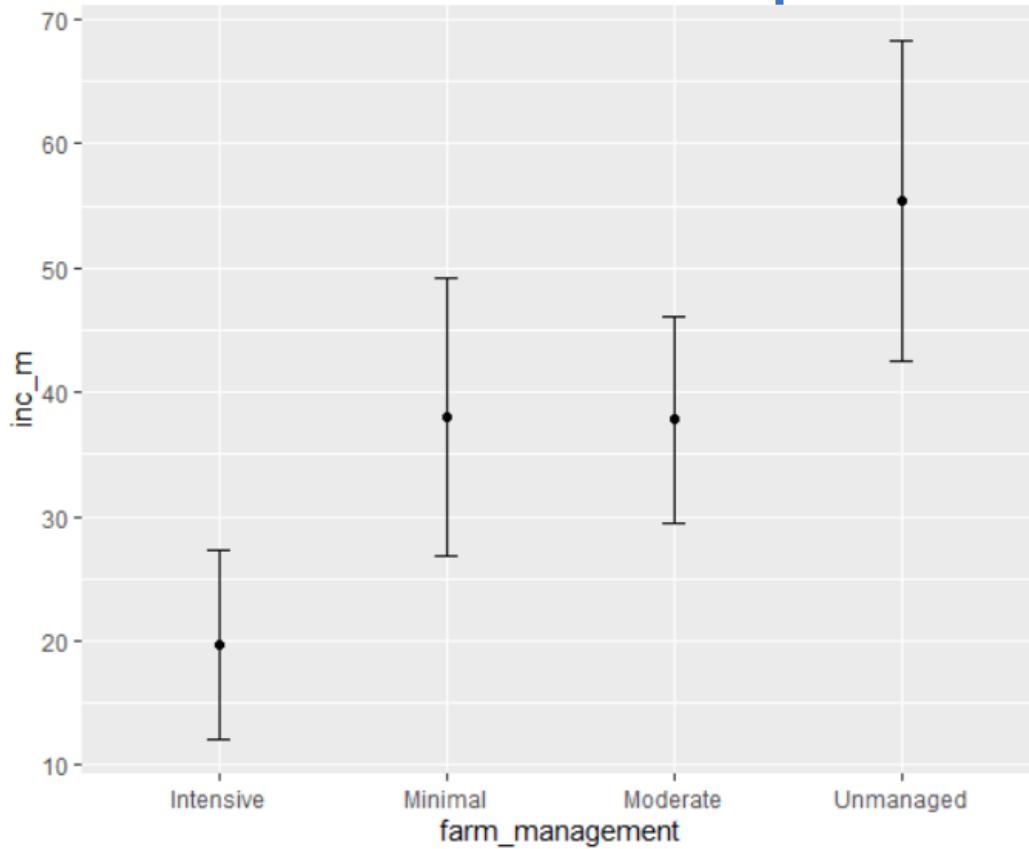
# Extra



```
demo_data =  
  survey_data %>%  
  group_by(farm_management) %>%  
  summarise(inc_m = mean(inc),  
            inc_sd = sd(inc),  
            lower = inc_m-inc_sd,  
            upper = inc_m+inc_sd)
```

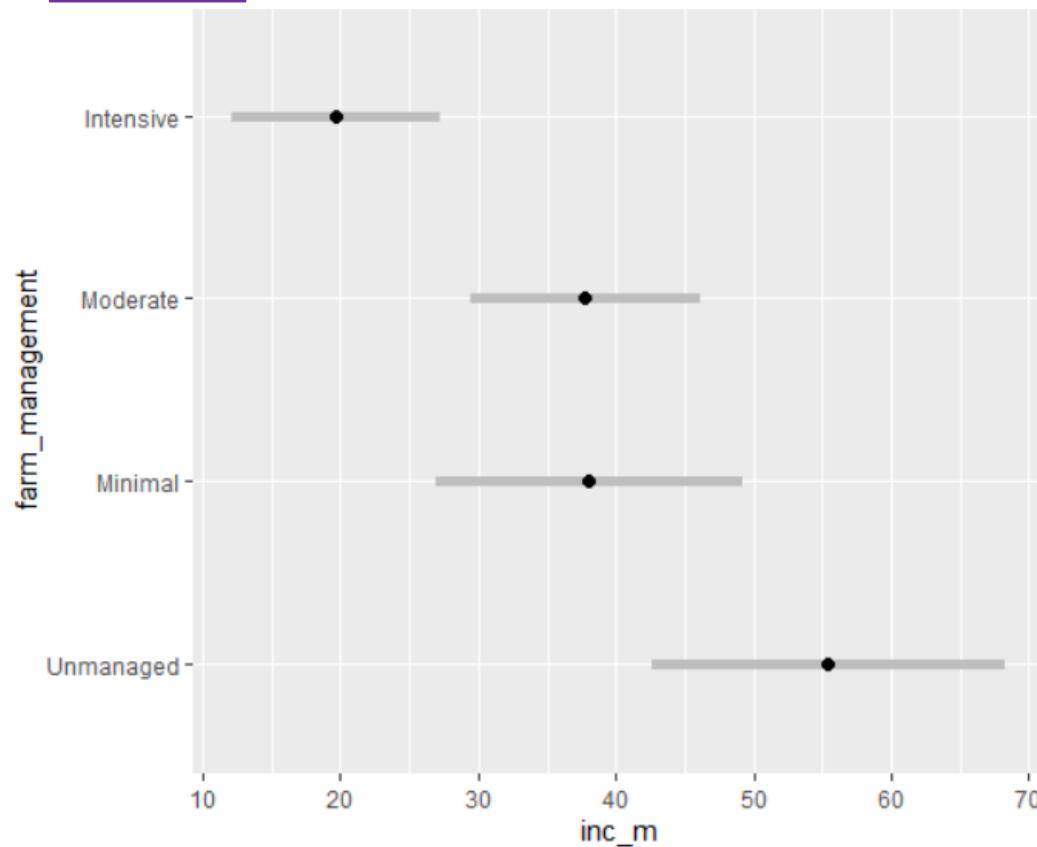
```
> demo_data  
# A tibble: 4 x 5  
  farm_management inc_m inc_sd lower upper  
  <fct>          <dbl>   <dbl>  <dbl>  <dbl>  
1 Unmanaged       55.4    12.9   42.6   68.3  
2 Minimal         38.0    11.1   26.9   49.2  
3 Moderate        37.8    8.31   29.5   46.1  
4 Intensive       19.7    7.56   12.1   27.2
```

```
ggplot(demo_data, aes(x = farm_management, y = inc_m))+  
  geom_errorbar(aes(ymin = lower, ymax = inc_m+inc_sd), width=.1)+  
  geom_point()
```

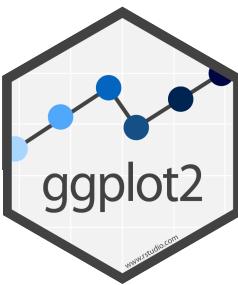
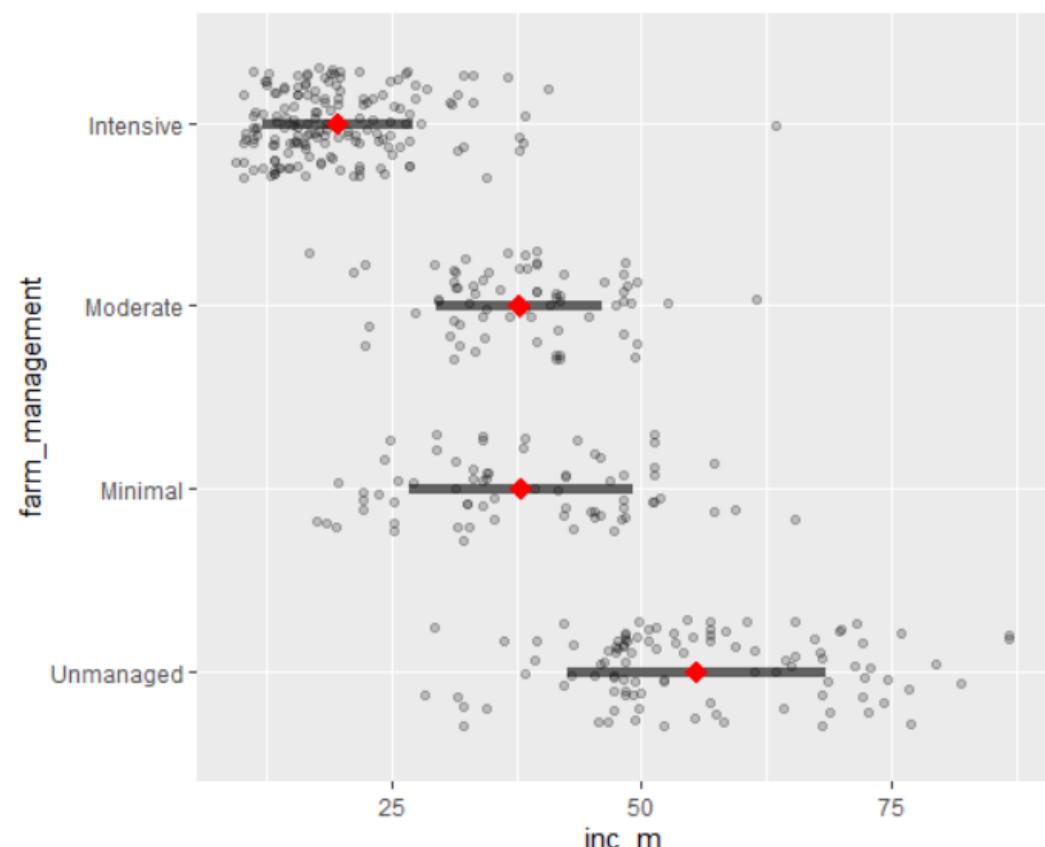


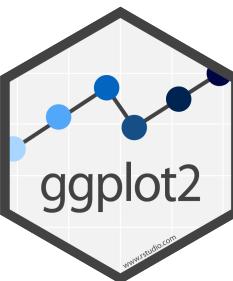
# Extra

```
ggplot(demo_data, aes(x = farm_management, y = inc_m))+  
  geom_linerange(aes(ymin = lower, ymax = inc_m+inc_sd),  
                 color = "grey", size = 2)+  
  geom_point(size=2)+  
  coord_flip()
```

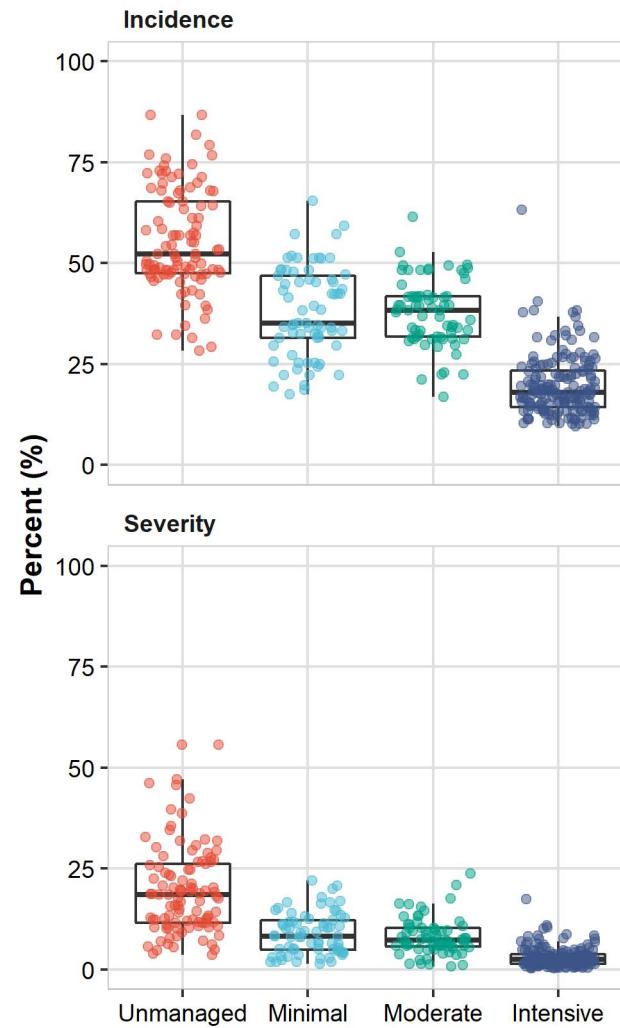


```
ggplot(data = demo_data, aes(x = farm_management, y = inc_m))+  
  geom_jitter(data = survey_data, aes(x = farm_management, y = inc),  
              width = 0.3, alpha = 0.2)+  
  geom_linerange(aes(ymin = inc_m-inc_sd, ymax = inc_m+inc_sd),  
                 color = "black", size = 2, alpha = 0.6)+  
  geom_point(size=4, shape = 18, color = "red") + coord_flip()
```





# The package grammar



DATA (survey\_data\_long) %>%

**ggplot function –**

**Geometric elements** (geom\_boxplot and geom\_jitter)

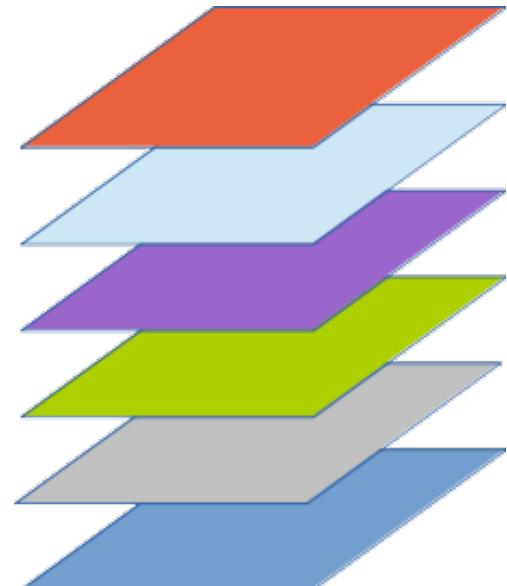
**Aesthetic** (x, y, and color)

**Scale**(y-continues and color [ggsci])

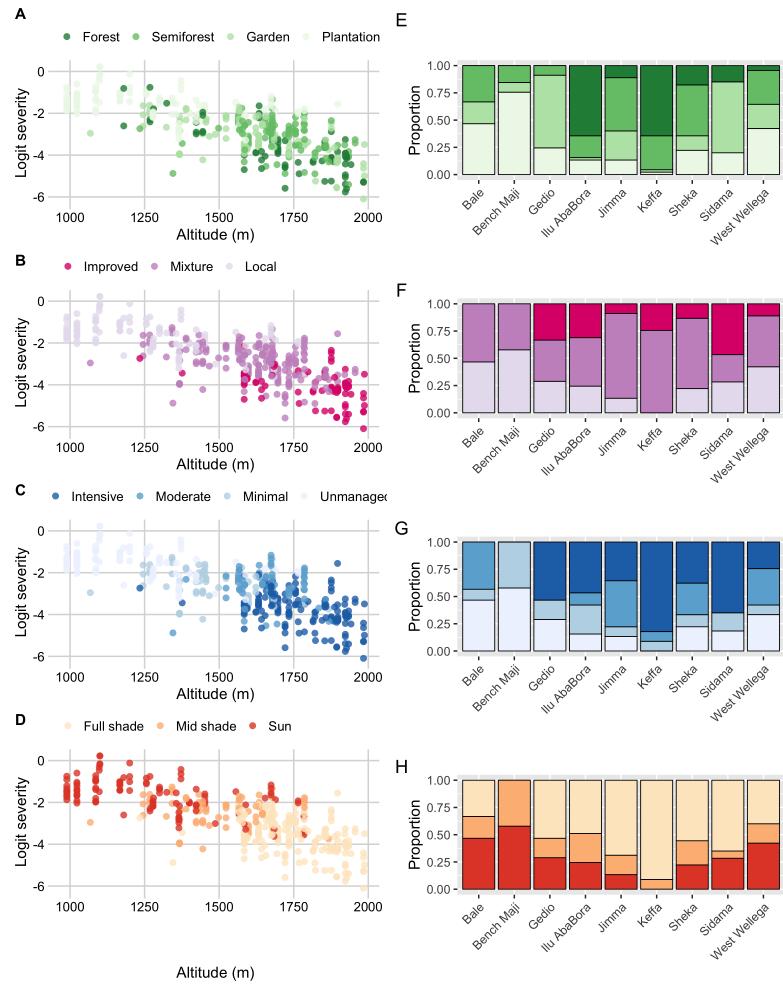
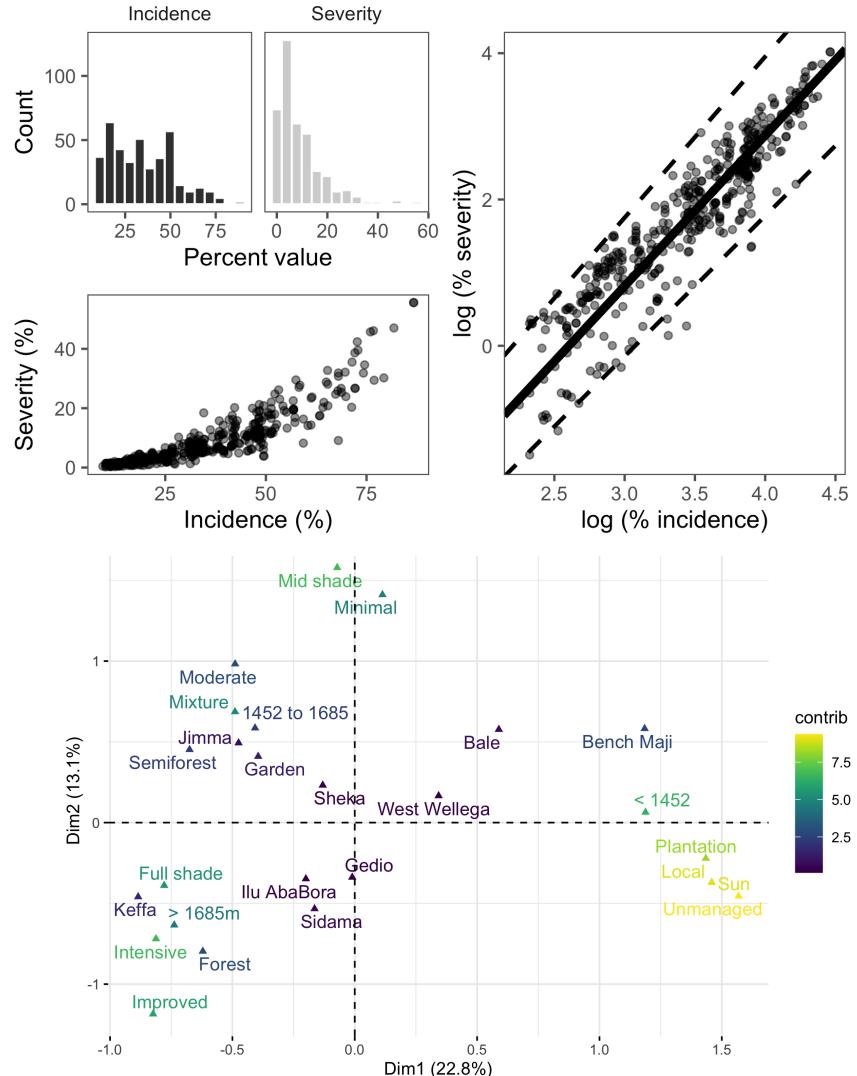
**Facets** (facet\_wrap)

**Themes**(customized)

**Save**(ggsave – tiff, 85x140 mm)

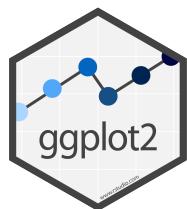


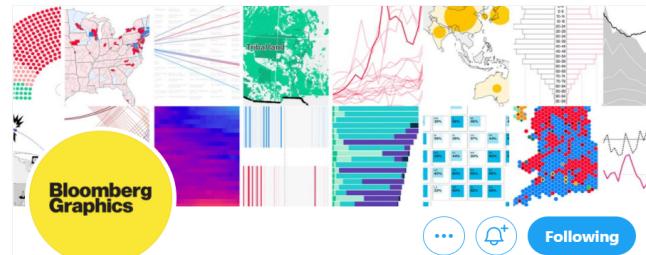
# Get inspired



Del Ponte, E. M., & Belachew, K.  
<https://doi.org/10.17605/OSF.IO/XEJAZ>

Belachew, K., Senbeta, G.A., Garedew, W. et al. Altitude is the main driver of coffee leaf rust epidemics: a large-scale survey in Ethiopia. *Trop. plant pathol.* **45**, 511–521 (2020).  
<https://doi.org/10.1007/s40858-020-00383-4>





### Bloomberg Graphics

@BBGVisualData

Graphs, maps and data journalism from the Bloomberg Graphics team.

🔗 [bloomberg.com/graphics](http://bloomberg.com/graphics) 📅 Joined May 2013



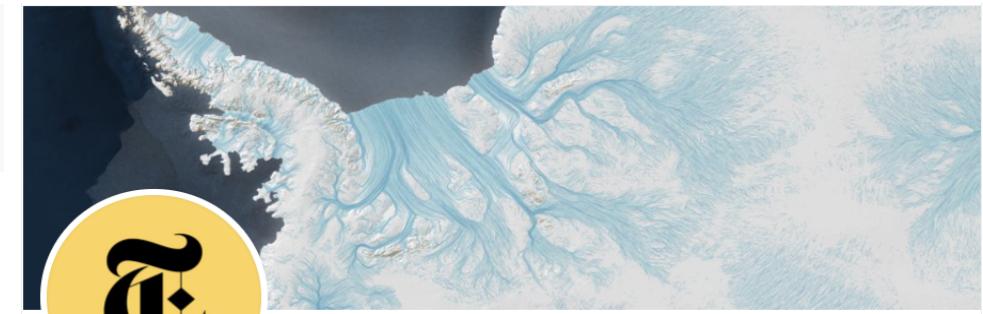
### Post Graphics

@PostGraphics

Graphics, visualizations and maps from The Washington Post.

We're hiring: [wapo.st/gfx-jobs](http://wapo.st/gfx-jobs)

📍 Washington, D.C. ↗ [washingtonpost.com](http://washingtonpost.com) 📅 Born March 8  
📅 Joined November 2009



### NYT Graphics

@nytgraphics

The New York Times Graphics Department

📍 New York, NY ↗ [nytimes.com](http://nytimes.com) 📅 Joined October 2009



# The Guardian



### Guardian Data

@GuardianData

Data-driven journalism by the @Guardian.

📍 London, UK ↗ [theguardian.com/news/datablog](http://theguardian.com/news/datablog) 📅 Joined March 2009

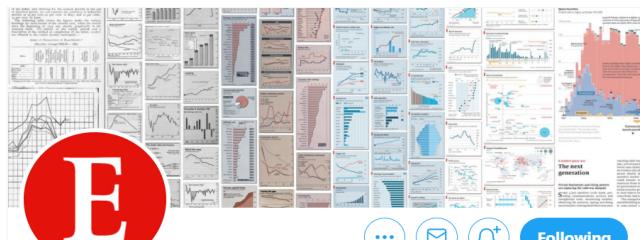


### FT Data

@ftdata

Welcome to FT Data, where we share a range of graphics and data-driven stories from the @financialtimes. Run by the FT's Visual and Data Journalism team.

📍 London ↗ [ft.com/graphics](http://ft.com/graphics) 📅 Joined August 2010



### The Economist Data Team

@ECONdailycharts

Charts, maps and data-driven journalism from The Economist data team

📍 London ↗ [economist.com/graphic-detail/](http://economist.com/graphic-detail/) 📅 Joined September 2009

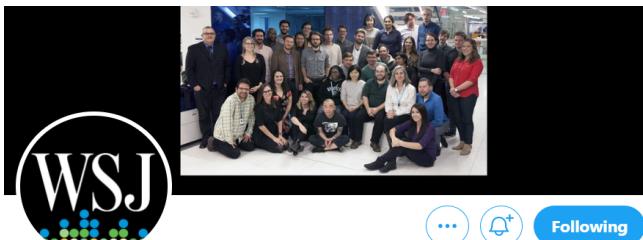


### Reuters Graphics

@ReutersGraphics

Information graphics from Thomson Reuters

📍 The world ↗ [graphics.reuters.com](http://graphics.reuters.com) 📅 Joined March 2013



### WSJ Graphics

@WSJGraphics

Information graphics and data from The Wall Street Journal. Tweets by @kathryntam, @sestamm, @anarivasWSJ

📍 Manhattan, NY ↗ [graphics.wsj.com](http://graphics.wsj.com) 📅 Joined January 2009

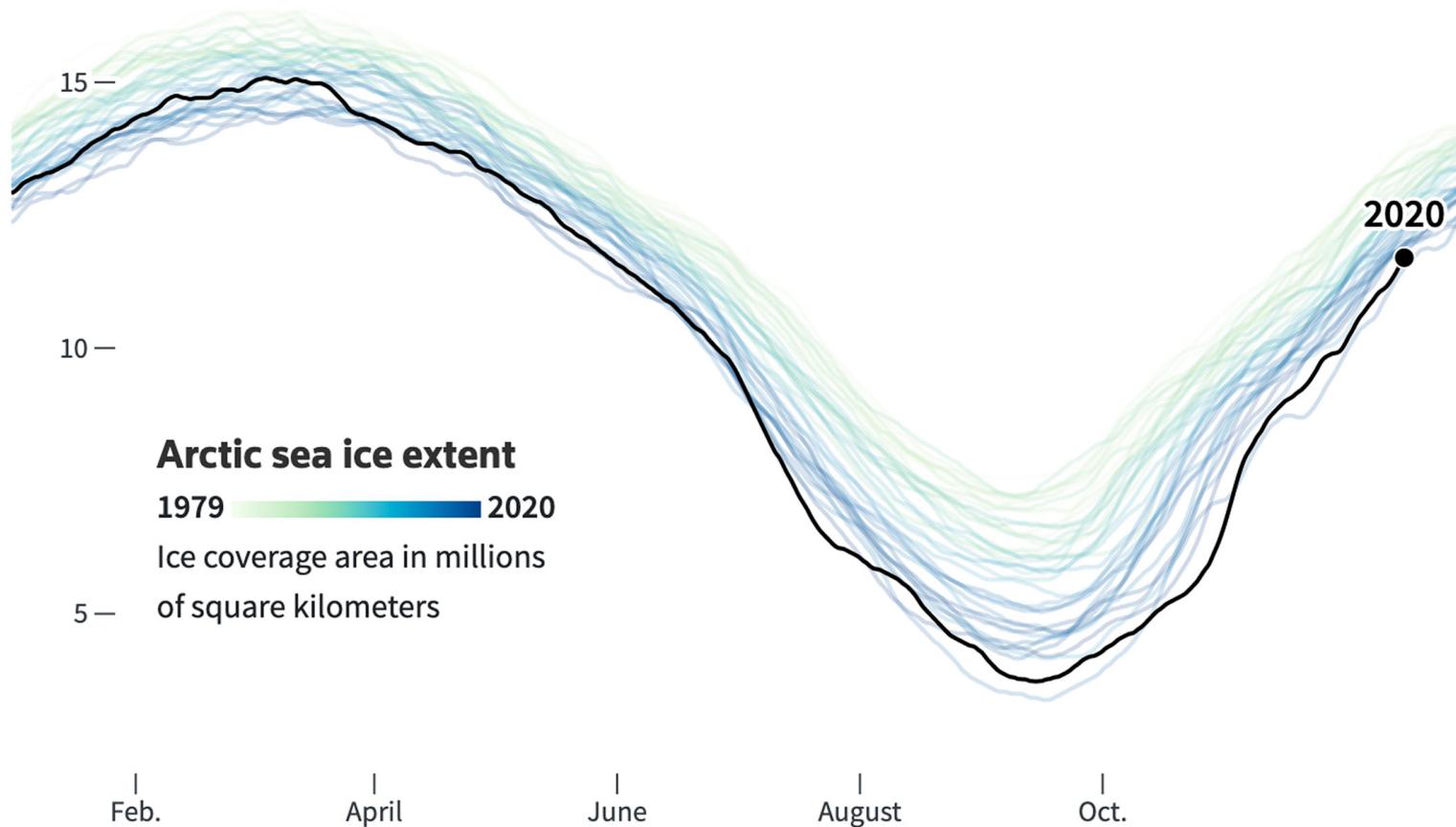


Reuters Graphics   
@ReutersGraphics

Replying to @ReutersGraphics

And in the Arctic, sea ice shrank to its second-lowest extent ever recorded.

9:10 AM · Dec 23, 2020 · Twitter Web App



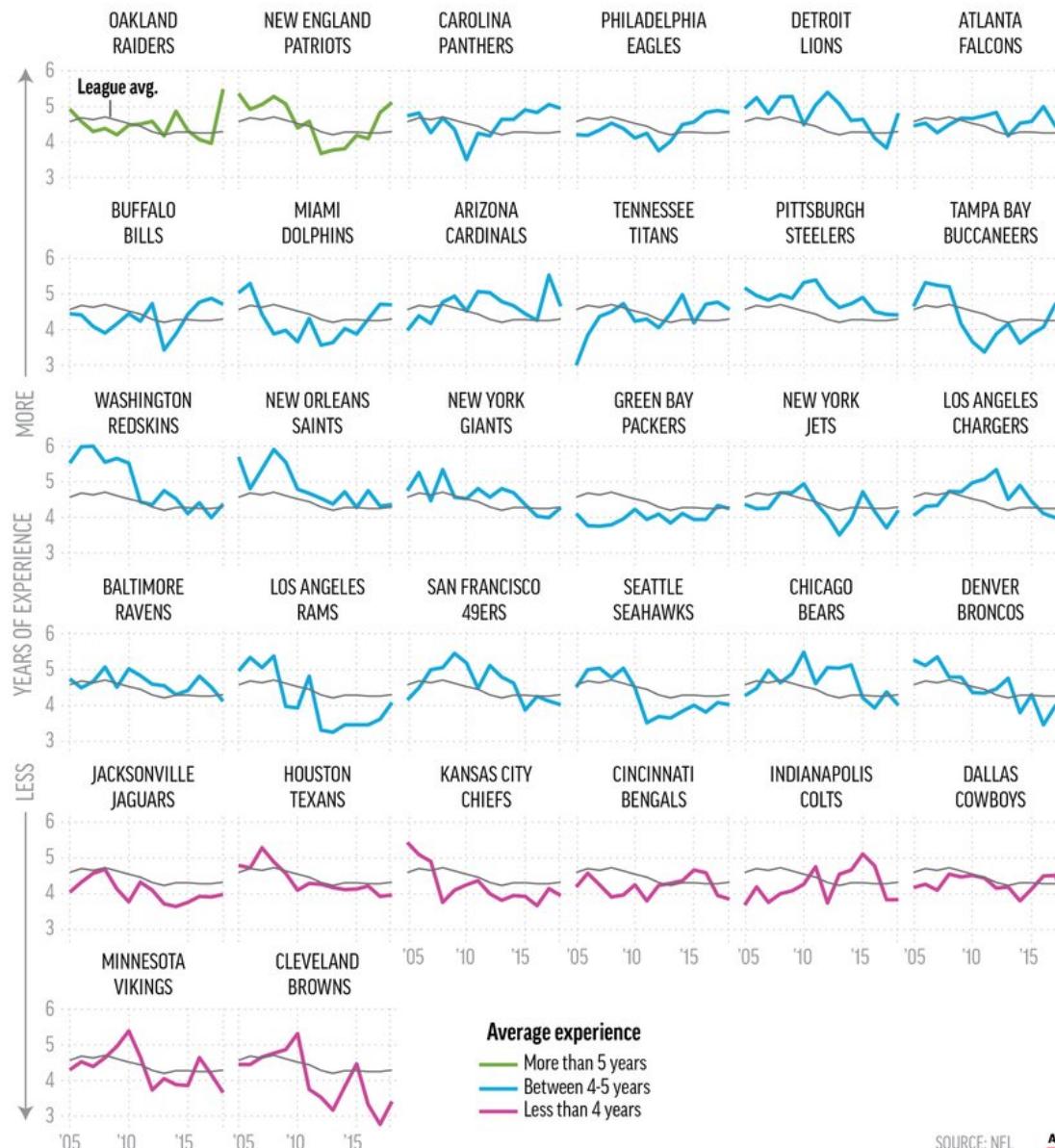
Source: [National Snow and Ice Data Center](#)

An [@APGraphics](#) and [@AP\\_Sports](#) look at how the [#NFL](#) has been getting younger and cheaper.  
[apne.ws/ToWGjwZ](http://apne.ws/ToWGjwZ)

8:38 AM · Jan 28, 2019 · SocialFlow

## Fewer NFL teams keeping experienced players on roster

AVERAGE NFL PLAYER EXPERIENCE BY TEAM, 2005-2018

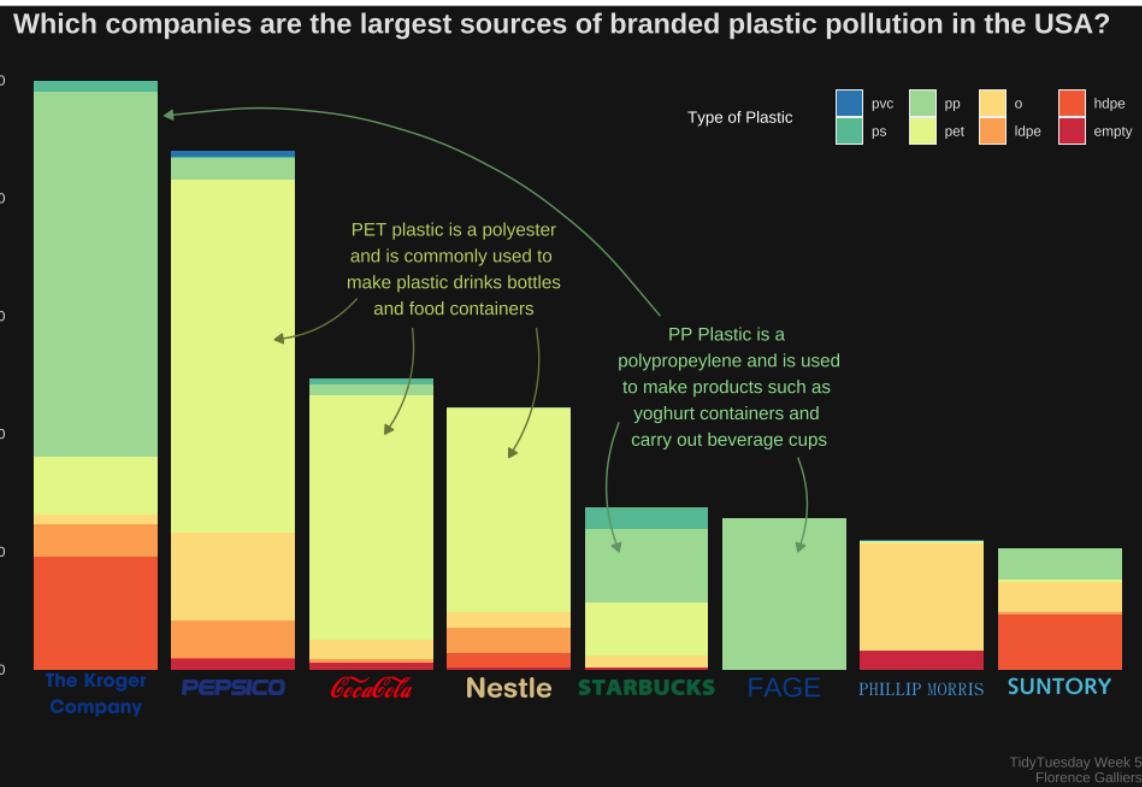


SOURCE: NFL

**AP**

# TidyTuesday

A weekly data project in R from the  
R4DS online learning community



@florencelydia11

