ggplot2 Introduction

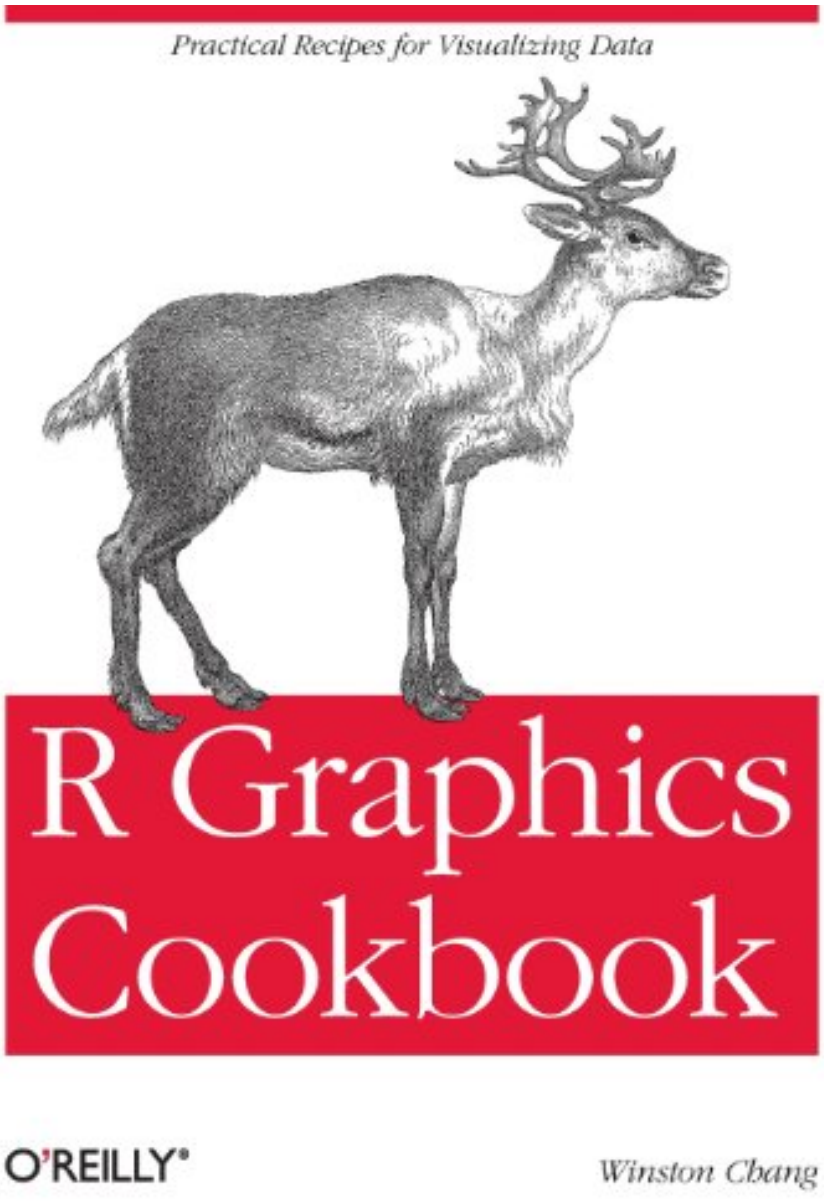
Jean-Baptiste Lecomte

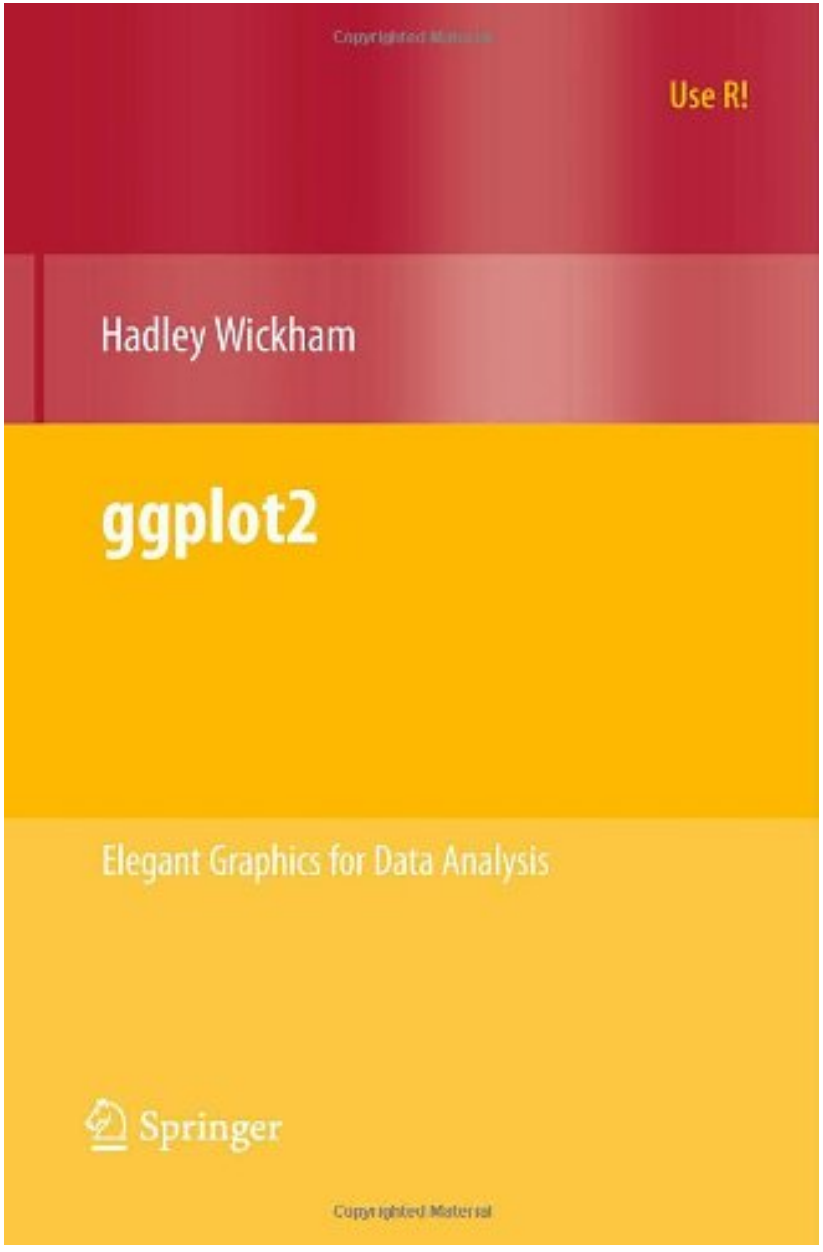
February 15, 2016

Introduction

- developped by Hadley Wickham (Rice University, Houston, USA)
- highly recommanded R packages to work with ggplot2: reshape and plyr (also developped by H. Wickham)
- first version called in 2007

Useful books





Online ressources

- ggplot2 official documentation:
  http://docs.ggplot2.org/current/
- R code related to ggplot2 cookbook:
  http://www.cookbook-r.com/Graphs/
- R code related to useR! ggplot2 book:
  http://ggplot2.org/book/
- Google groups to ask questions:
  ggplot2@googlegroups.com
- Statistical tools for high-throughput data analysis:
  http://www.sthda.com/english/wiki/ggplot2-essentials
- Github repository:
  https://github.com/yhat/ggplot/

Introduction

- based on new aesthetic principles
- based on *The grammar of graphics* developed by Wilkinson in 2005
- efficient way to produce simple graphics with a length reduction of R code

Forget about R base graphics:

`plot()`, `hist()`, `par()`, `layout()`, `points()`, `lines()`,`legend()`

Principle

ggplot2 is based on a **layer** system which can be used as objects.

Main layers

- data $\rightarrow$ raw data
- mapping $\rightarrow$ graphic projection
- geom $\rightarrow$ geometric objects (points, lines, polygons, ...)
- stat $\rightarrow$ statistics transformation (histogram, model)
- scale $\rightarrow$ aesthetics customization (color, shape, size, axes, legend)
- coord $\rightarrow$ coordinate system (axes, grid)
- facet $\rightarrow$ subdivision (lattice, trellis)

Base functions

ggplot2 is based on two functions:

1. `qplot()` for **q**uick **plot**
   - easy and fast, but too simple in most cases
   - `qplot(x, y, data=data)`

2. `ggplot()`
   - more complex but more powerful and flexible by adding `layers`
   - `ggplot(data=data, aes(x, y)) + layers`

Getting Started

Data format

Always work with a `data.frame`

Our data frame is based on the surveys XXXX and simulated data. Github repository:
https://github.com/JBLecomte/ggplot2-Introduction.git

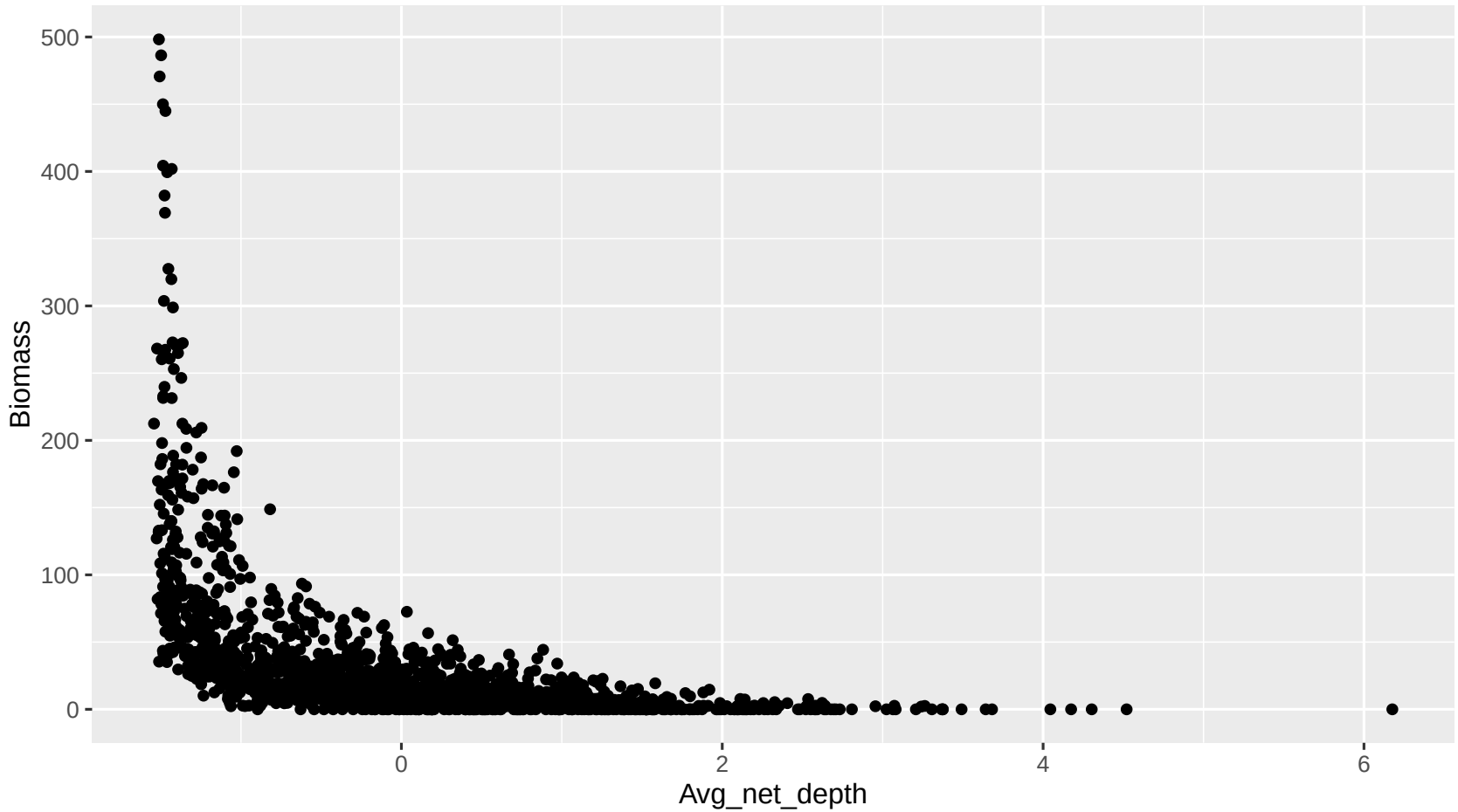
Getting Started

```
str(df_data)
```

```
## 'data.frame': 1909 obs. of  18 variables:
##  $ Year            : int  2005 2005 2005 2005 2005 2005 2005 2005 2005 2005 ...
##  $ Month           : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ DURATION_MINUTES: int  21 20 21 21 20 20 20 21 21 20 ...
##  $ AREA            : Factor w/ 2 levels "5AB","5CD": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Avg_net_depth   : num  -0.316 -0.435 -0.442 -0.234 -0.171 ...
##  $ Avg_net_temp    : num  0.3939 0.4339 0.3004 0.1335 -0.0267 ...
##  $ Date            : Date, format: "2005-07-06" "2005-07-06" ...
##  $ Lon             : num  -128 -128 -128 -128 -128 ...
##  $ Lat             : num  51.2 51.1 51.6 51.6 51.7 ...
##  $ X               : num  572025 570307 553665 551917 546338 ...
##  $ Y               : num  5668122 5665874 5717947 5719597 5723992 ...
##  $ X_km            : num  572 570 554 552 546 ...
##  $ Y_km            : num  5668 5666 5718 5720 5724 ...
##  $ Pres            : num  1 1 1 1 1 1 1 0 0 1 ...
##  $ Year_fac        : Factor w/ 5 levels "2005","2007",..: 1 1 1 1 1 1 1 1 1 1 ..
##  $ AREA_num        : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ nFish           : int  1 1 3 1 3 2 2 1 1 2 ...
##  $ Biomass         : num  2.52 2.46 7.46 1.93 6.77 ...
```

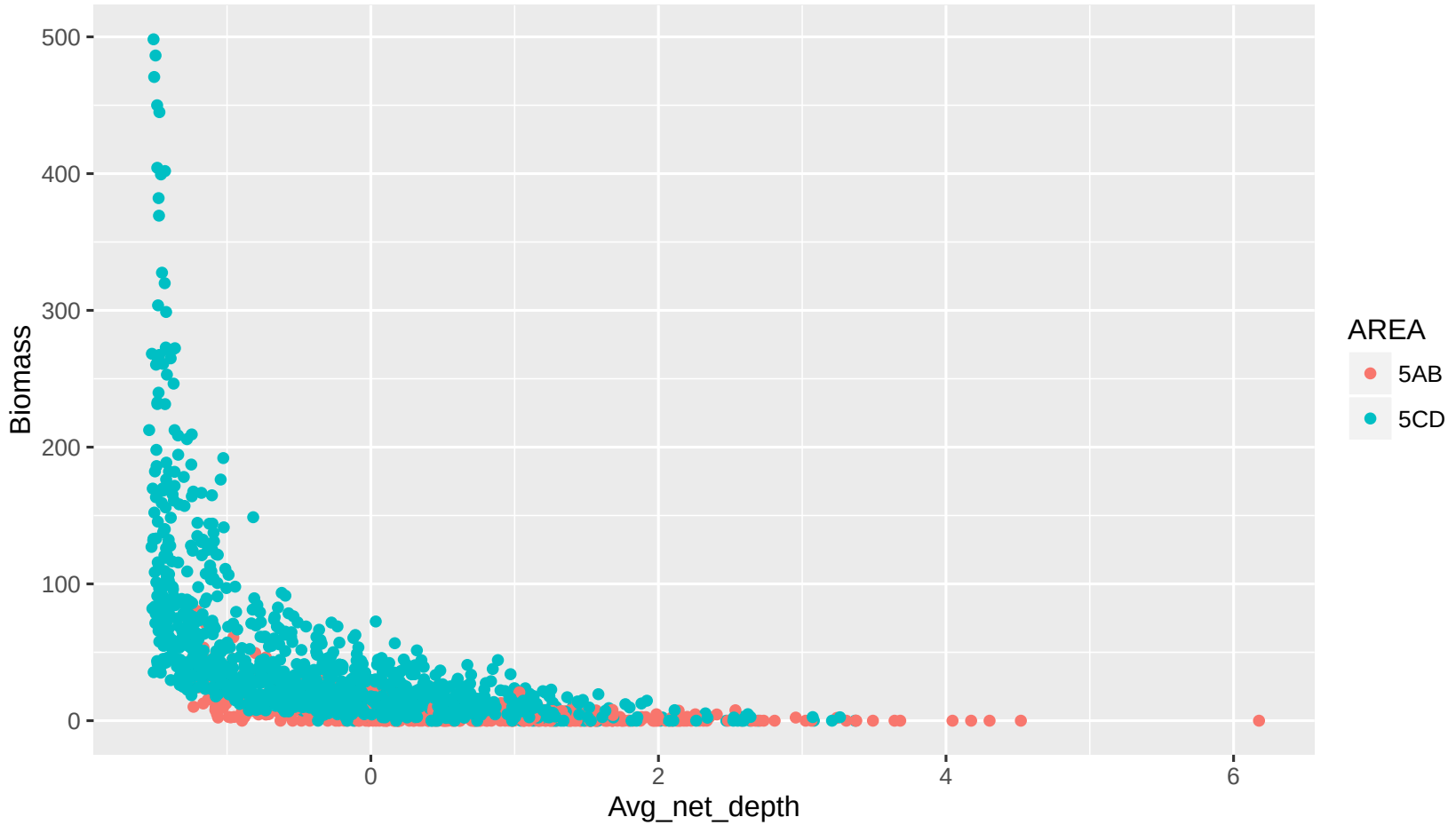
Scatter plot: Depth and Biomass

```
sp <- ggplot(data=df_data, aes(x=Avg_net_depth, y=Biomass)) +
  geom_point()
print(sp)
```



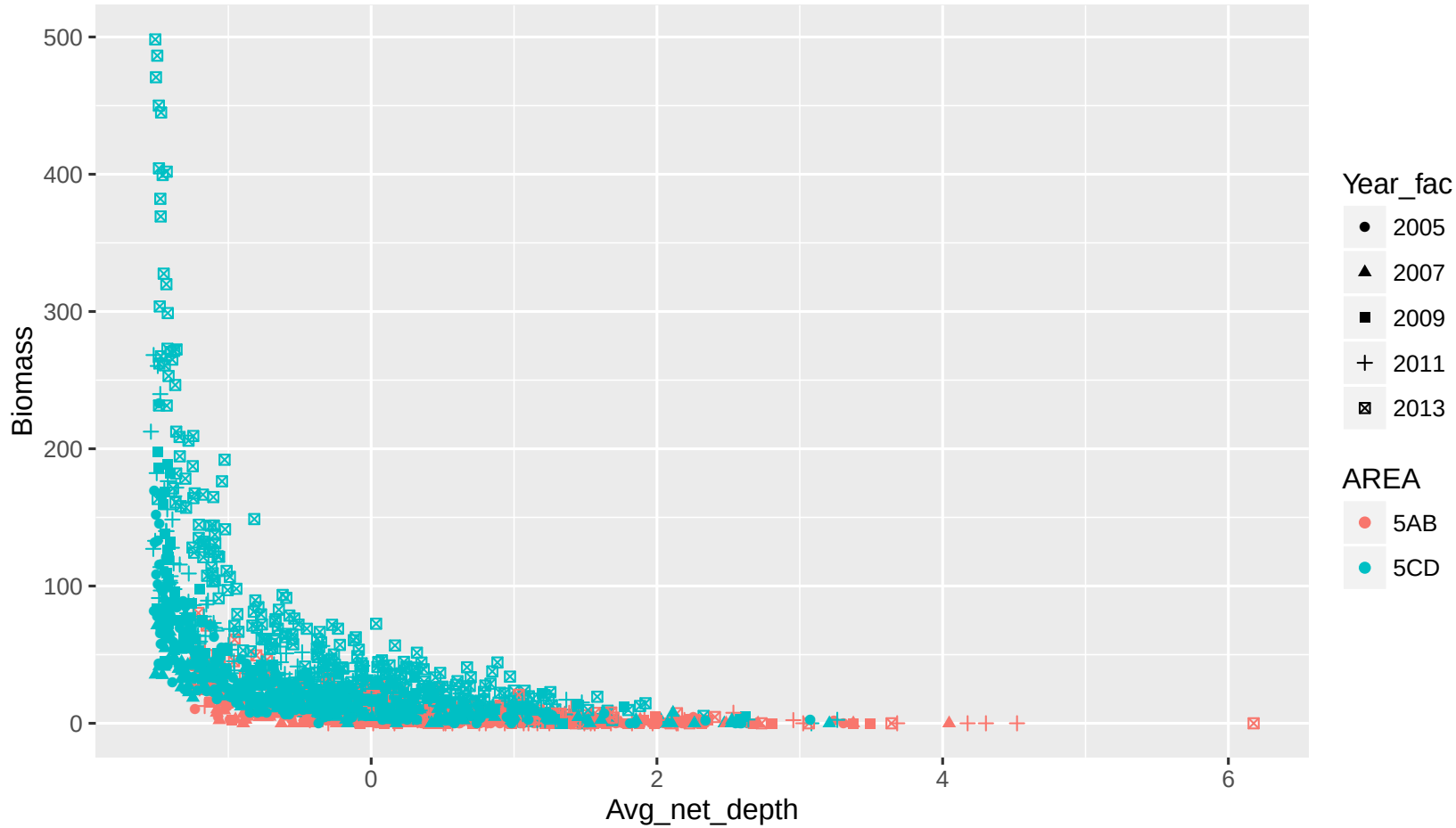
Scatter plot with color: Depth and Biomass

```
sp_color <- ggplot(df_data, aes(x=Avg_net_depth, y=Biomass,
                                color=AREA)) +
  geom_point()
print(sp_color)
```



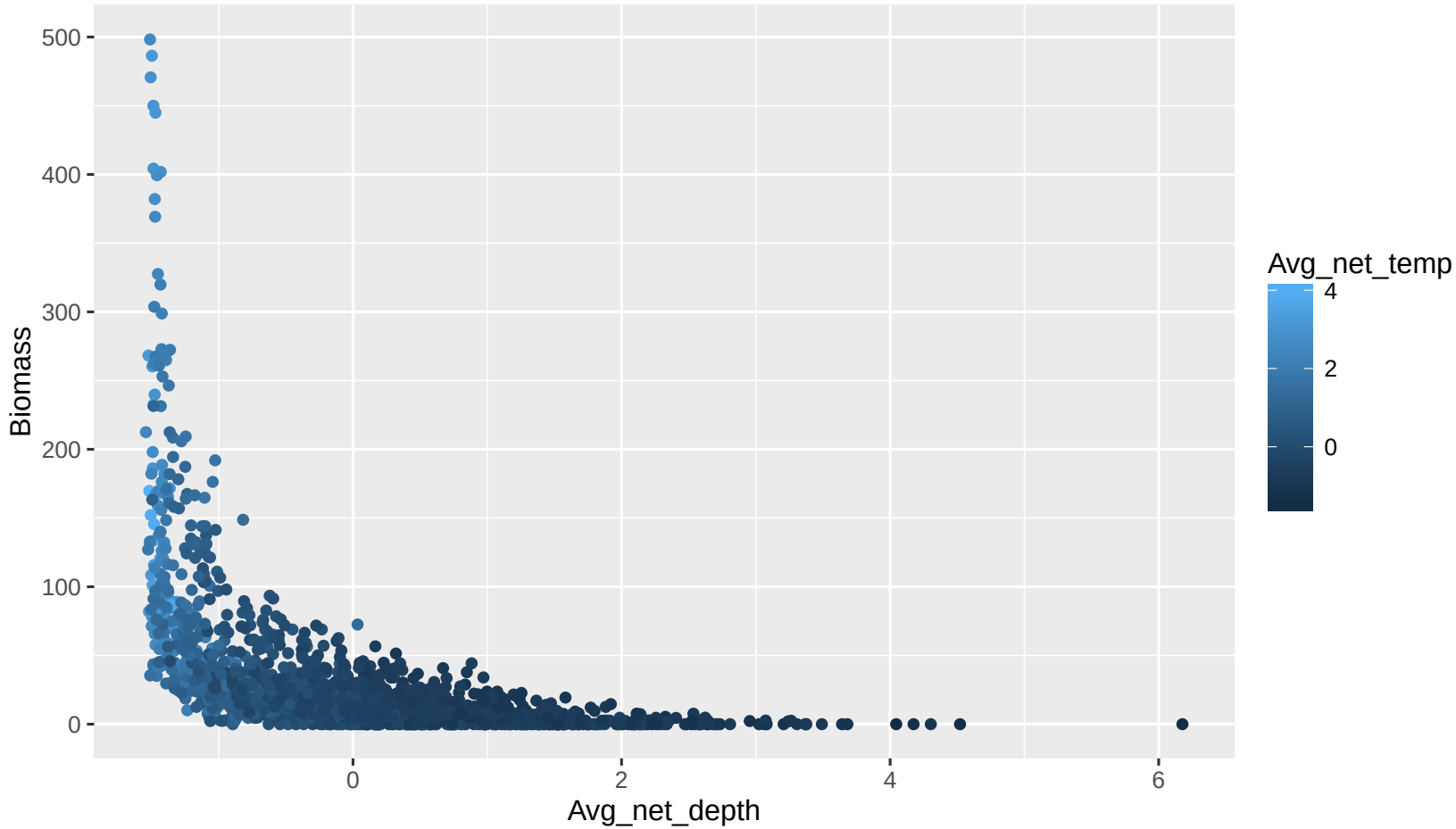
Scatter plot with shape: Depth and Biomass

```
sp_shape <- ggplot(df_data, aes(x=Avg_net_depth, y=Biomass,
                                color=AREA, shape=Year_fac)) +
  geom_point()
print(sp_shape)
```



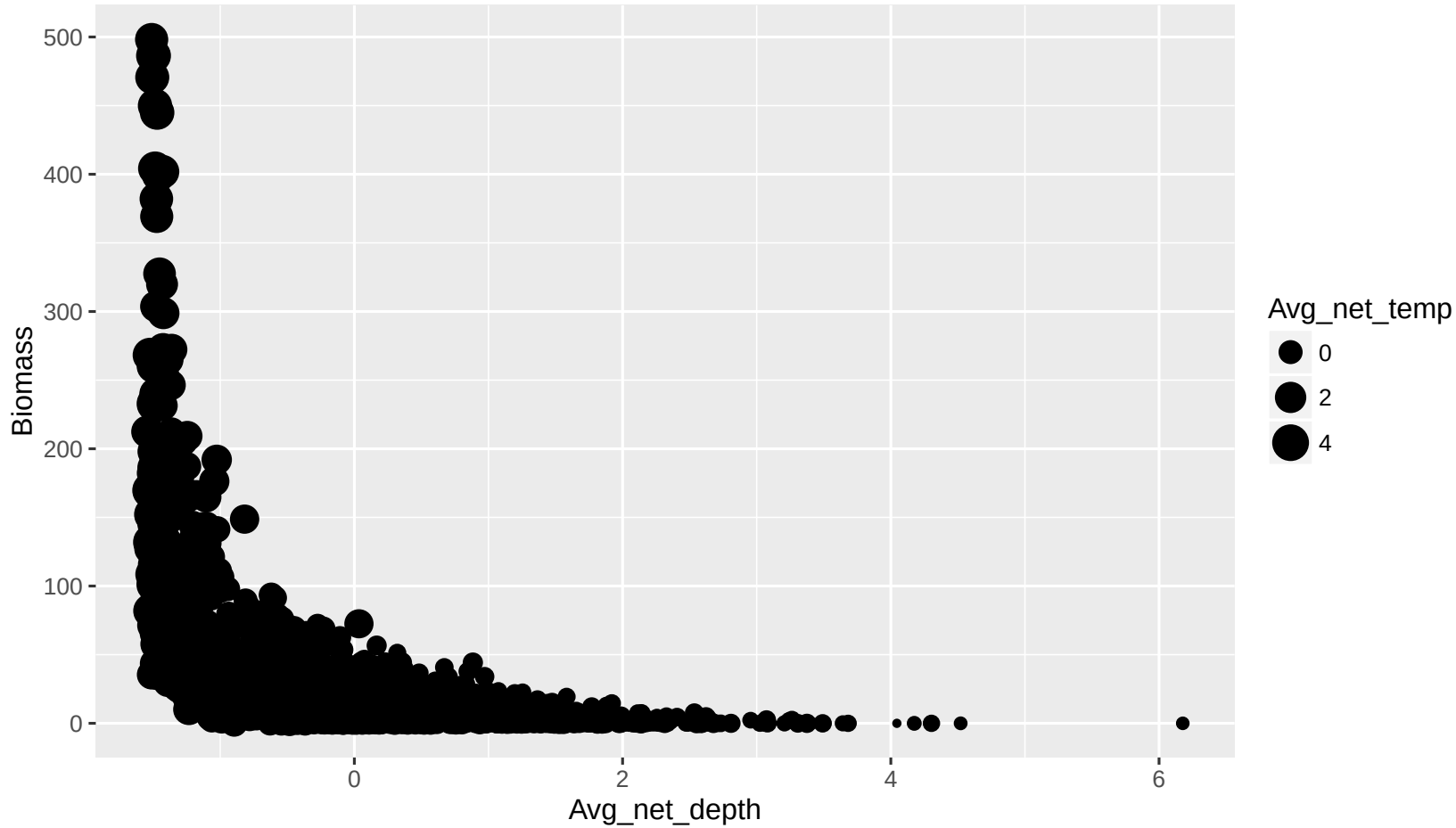
Scatter plot with continuous color: Depth and Biomass

```
sp_color_cont <- ggplot(df_data, aes(x=Avg_net_depth, y=Biomass,
                                     color=Avg_net_temp)) +
  geom_point()
print(sp_color_cont)
```



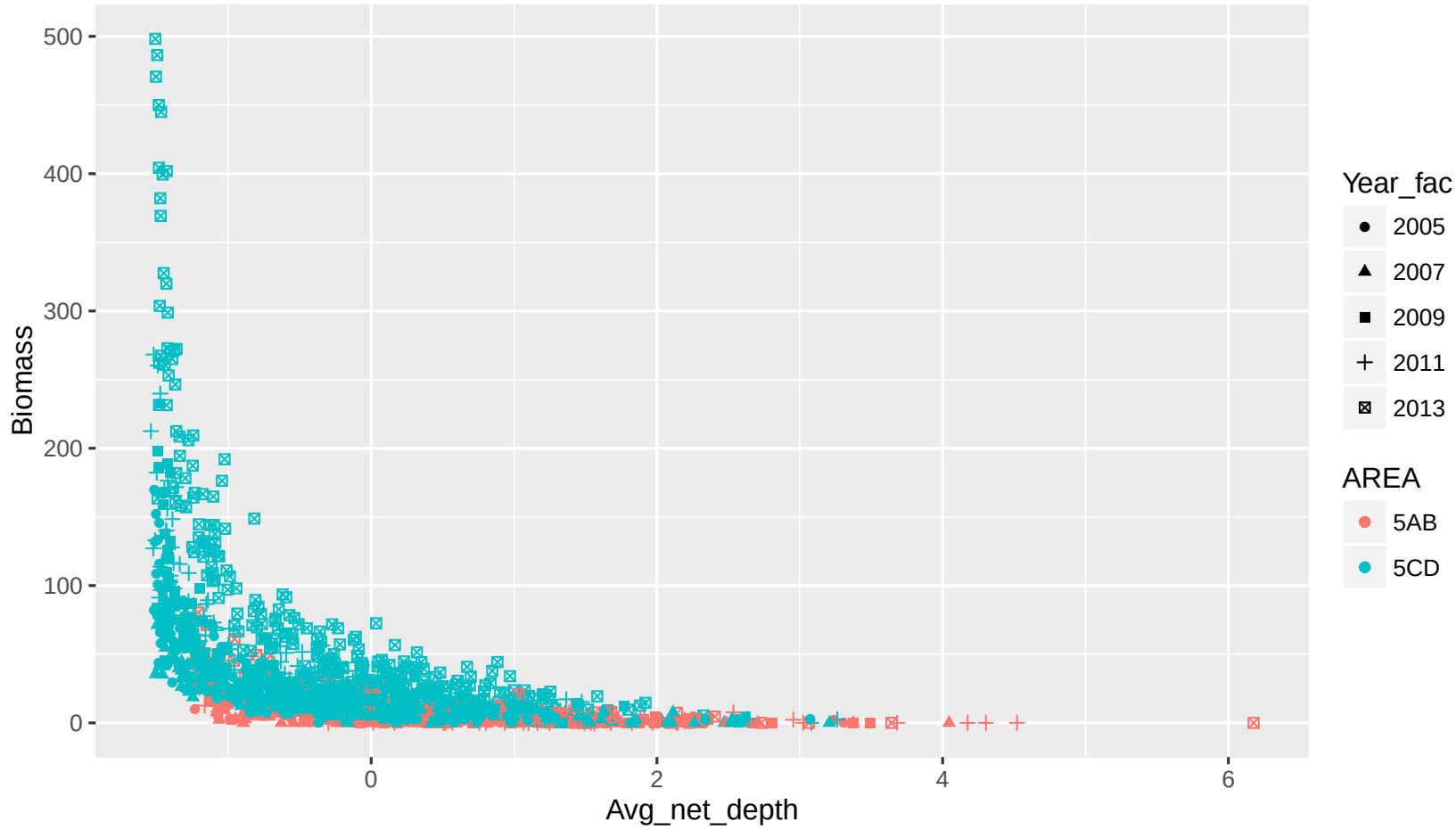
Scatter plot with size: Depth and Biomass

```
sp_area <- ggplot(df_data, aes(x=Avg_net_depth, y=Biomass,
                               size=Avg_net_temp)) +
  geom_point()
print(sp_area)
```



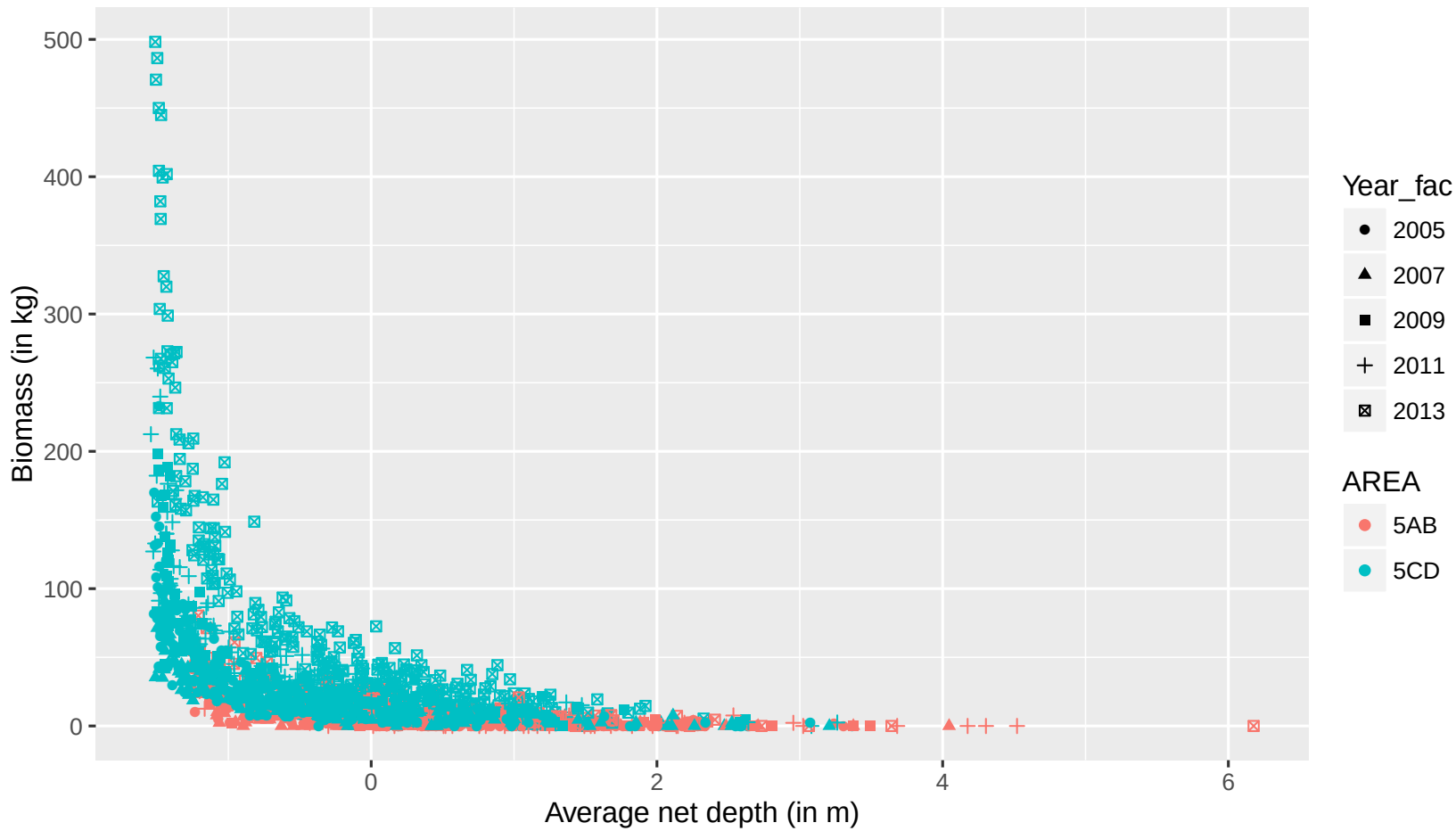
Improvement of a plot

```
print(sp_shape)
```



Improvement of a plot: axes names

```
sp_shape_imp1 <- sp_shape +
  xlab('Average net depth (in m)') +
  ylab('Biomass (in kg)')

print(sp_shape_imp1)
```



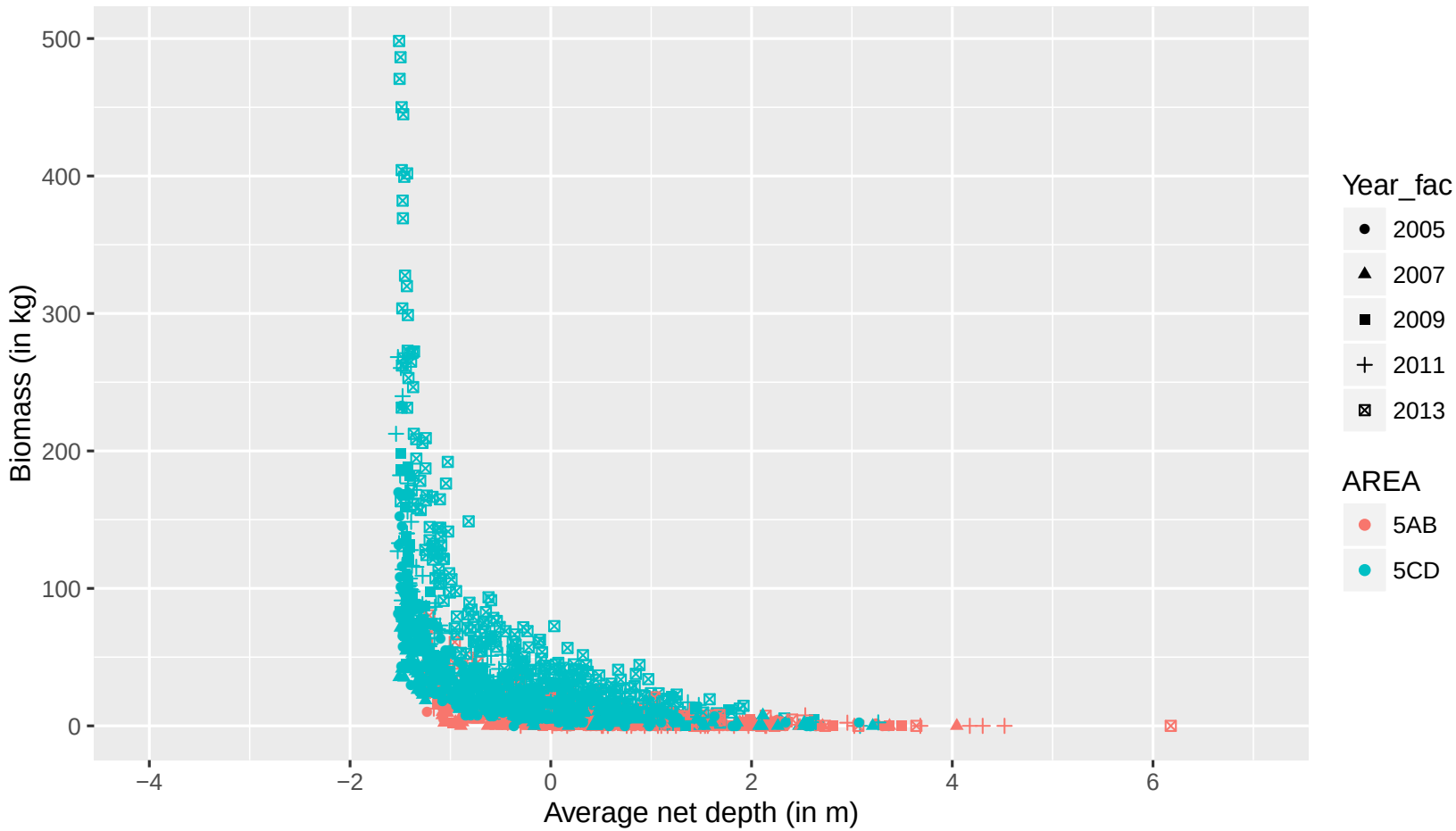
Improvement of a plot: axes options

scale_x_continuous (name, breaks, labels , limits , trans)
scale_y_continuous (name, breaks, labels , limits , trans)

- name : x or y axis labels
- breaks : to control the breaks in the guide (axis ticks, grid lines, âĂę). Among the possible values, there are : NULL : hide all breaks waiver() : the default break computation a character or numeric vector specifying the breaks to display
- labels : labels of axis tick marks. Allowed values are : NULL for no labels waiver() for the default labels character vector to be used for break labels
- limits : a numeric vector specifying x or y axis limits (min, max) trans for axis transformations. Possible values are âĂIJlog2âĂİ, âĂIJlog10âĂİ, âĂę

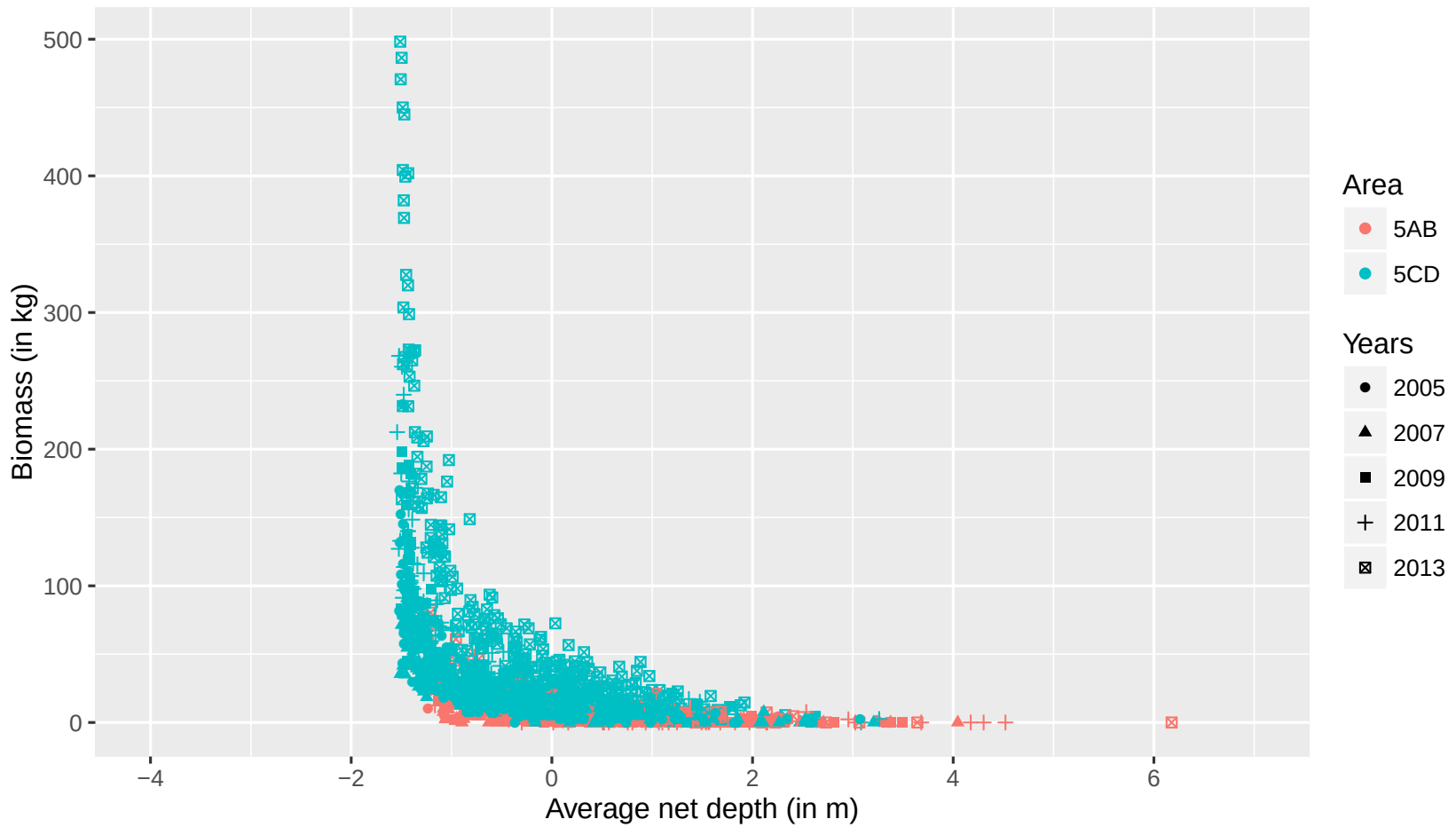
Improvement of a plot: axes options

```
sp_shape_imp1 <- sp_shape +
  scale_x_continuous(name='Average net depth (in m)',
                     breaks=seq(-4,7,2), limits=c(-4,7)) +
  ylab('Biomass (in kg)')
print(sp_shape_imp1)
```



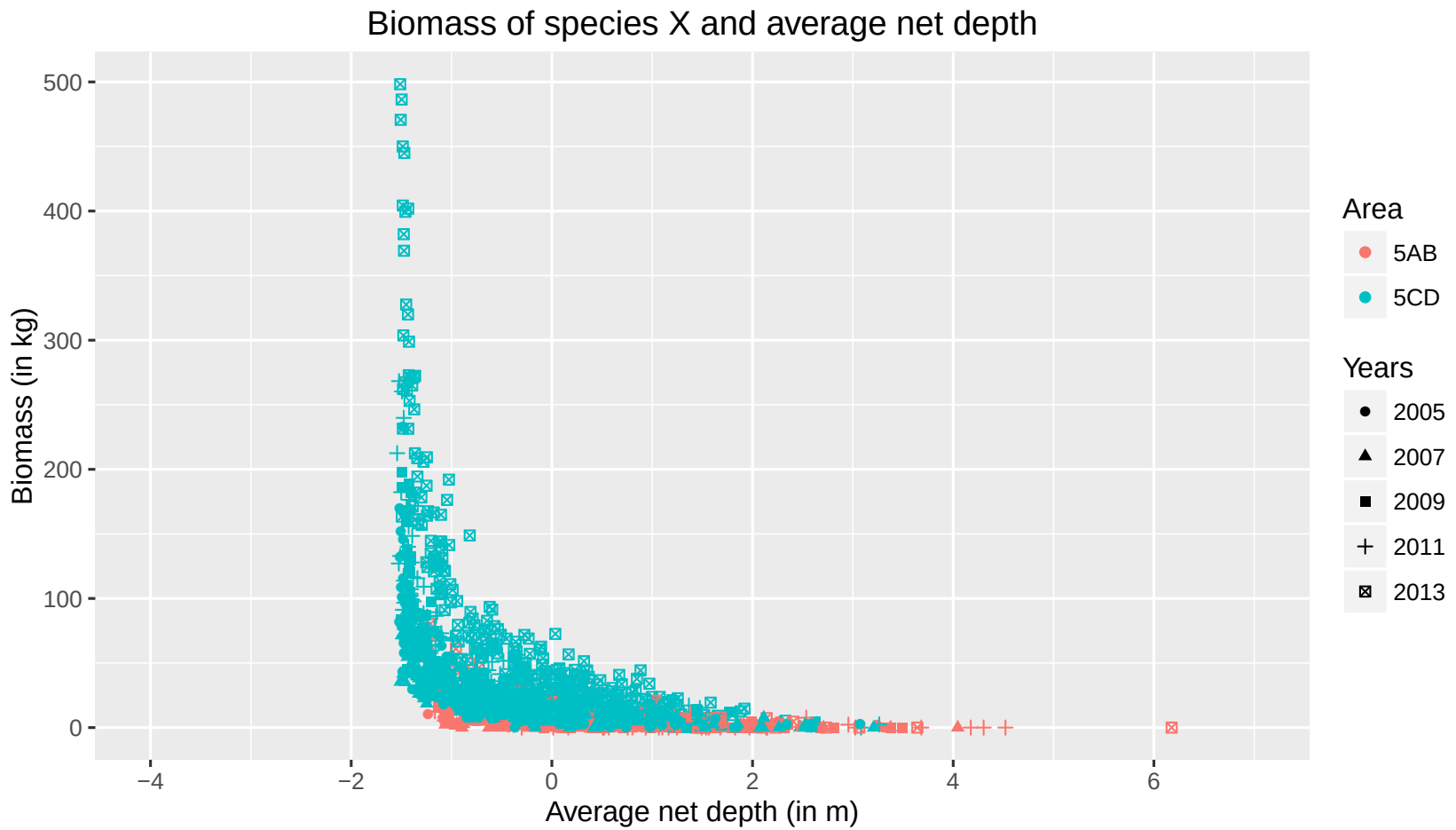
Improvement of a plot: legend titles

```
sp_shape_imp2 <- sp_shape_imp1 +
  scale_shape_discrete(name="Years") +
  scale_color_discrete(name="Area")

print(sp_shape_imp2)
```



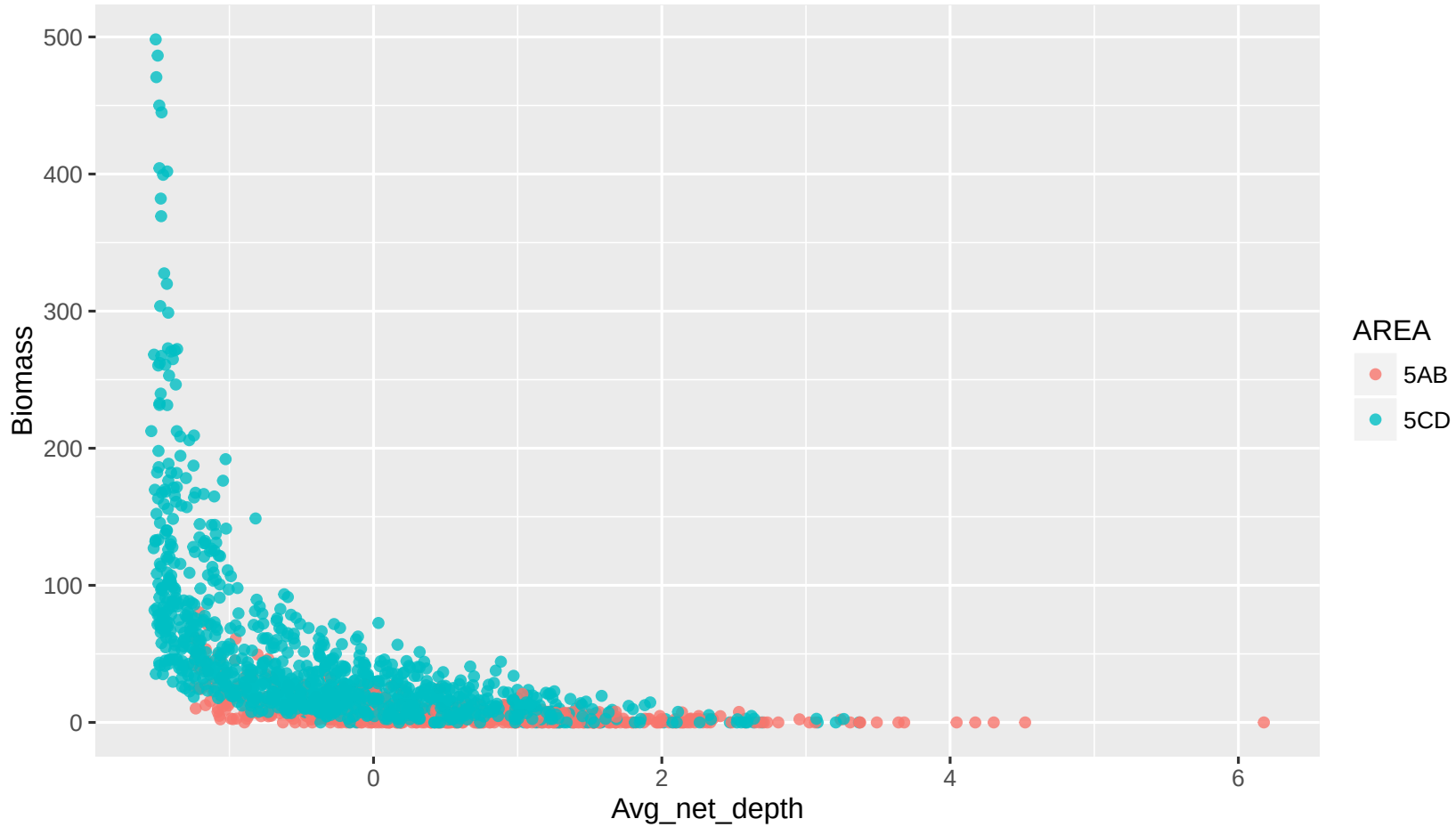
Improvement of a plot: plot title

```
sp_shape_imp3 <- sp_shape_imp2 +
  ggtitle("Biomass of species X and average net depth")

print(sp_shape_imp3)
```



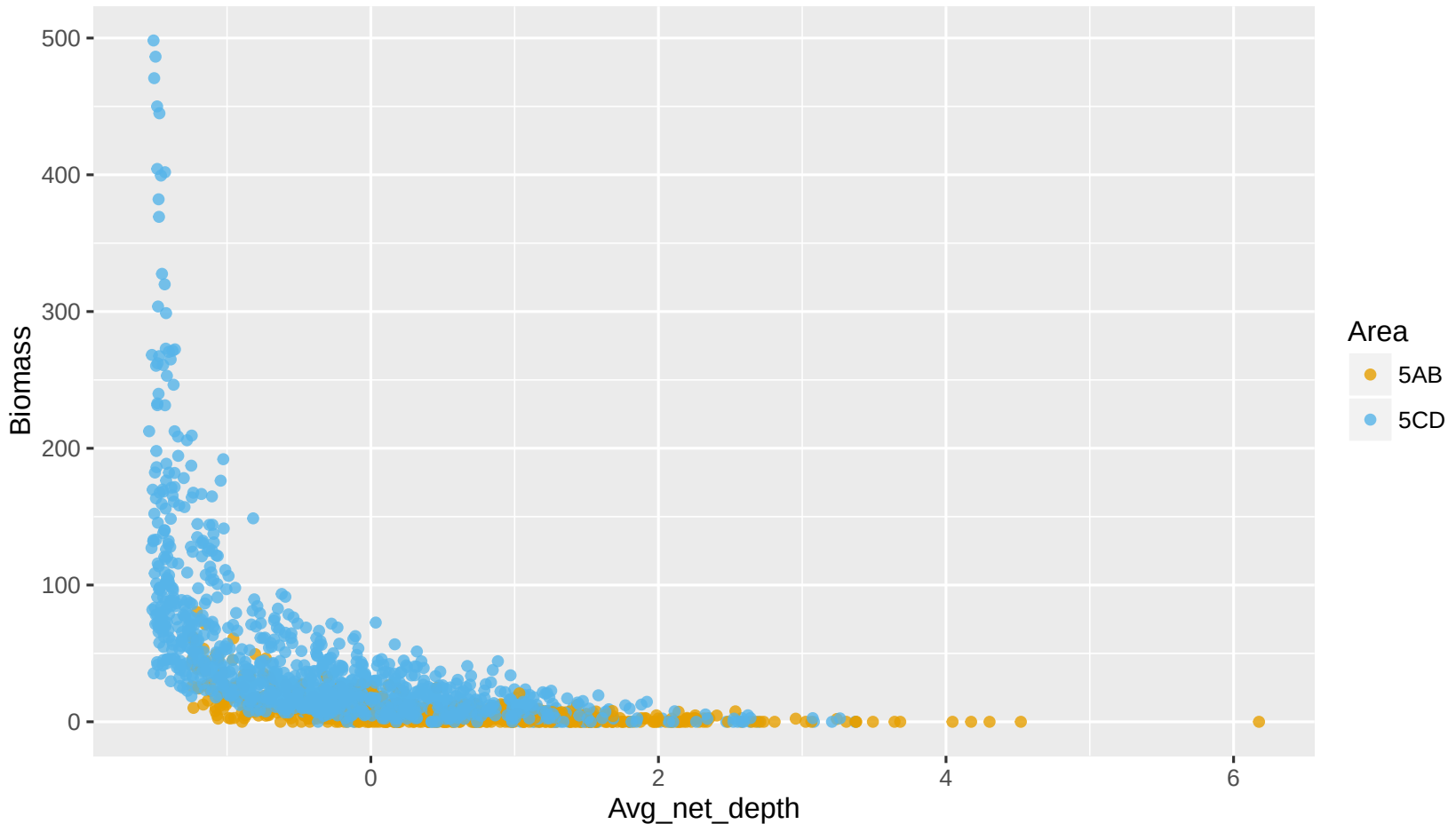
Discrete color scale

```
sp_color <- ggplot(df_data, aes(x=Avg_net_depth, y=Biomass,
                                color=AREA)) +
  geom_point(alpha=0.8)
print(sp_color)
```



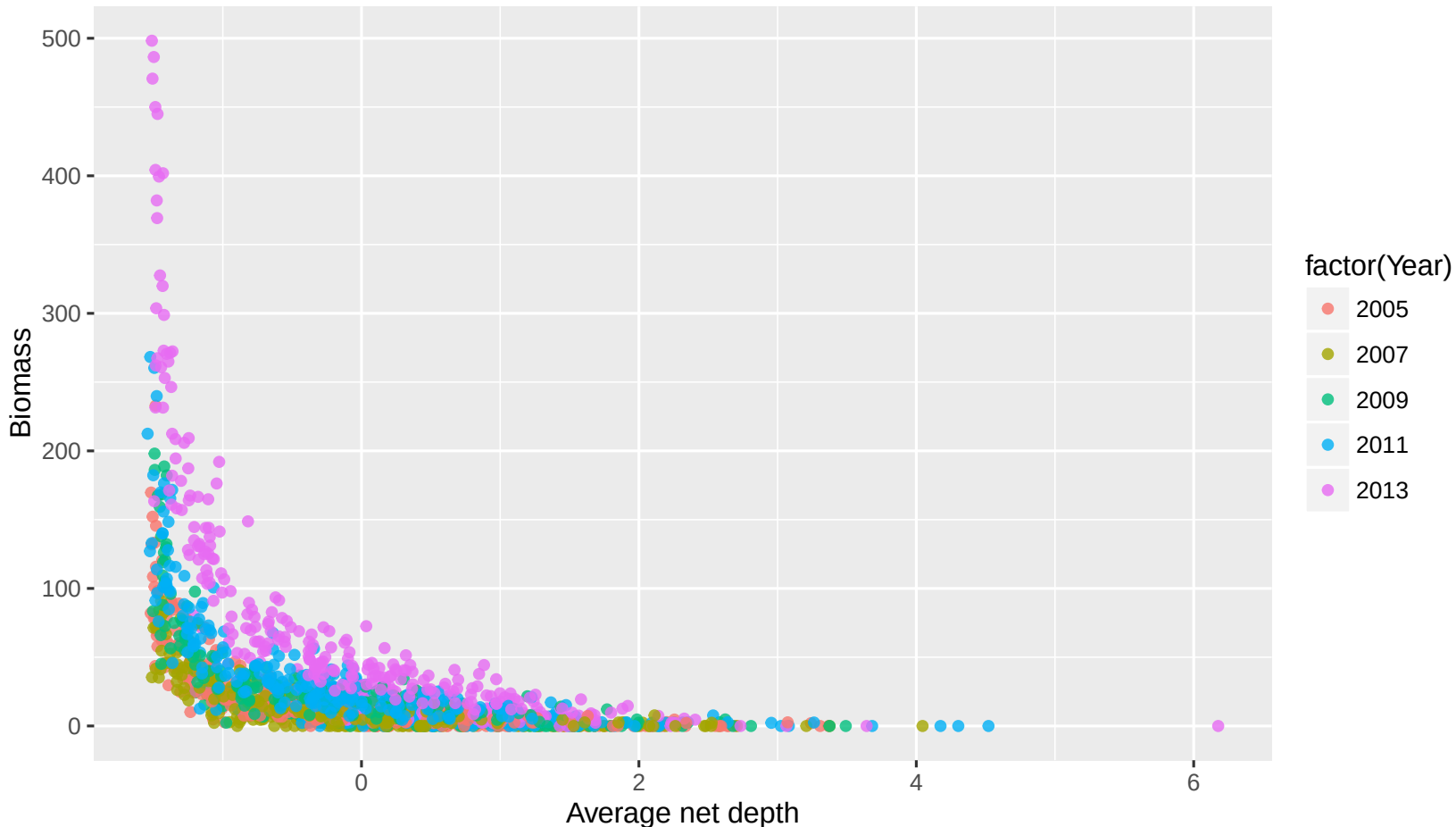
Discrete color scale: manual

```
sp_c_manual <- ggplot(df_data, aes(x=Avg_net_depth, y=Biomass,
                                   color=AREA)) +
  geom_point(alpha=0.8) +
  scale_color_manual(name='Area', values=c("#E69F00", "#56B4E9"))
print(sp_c_manual)
```



Discrete color scale: manual

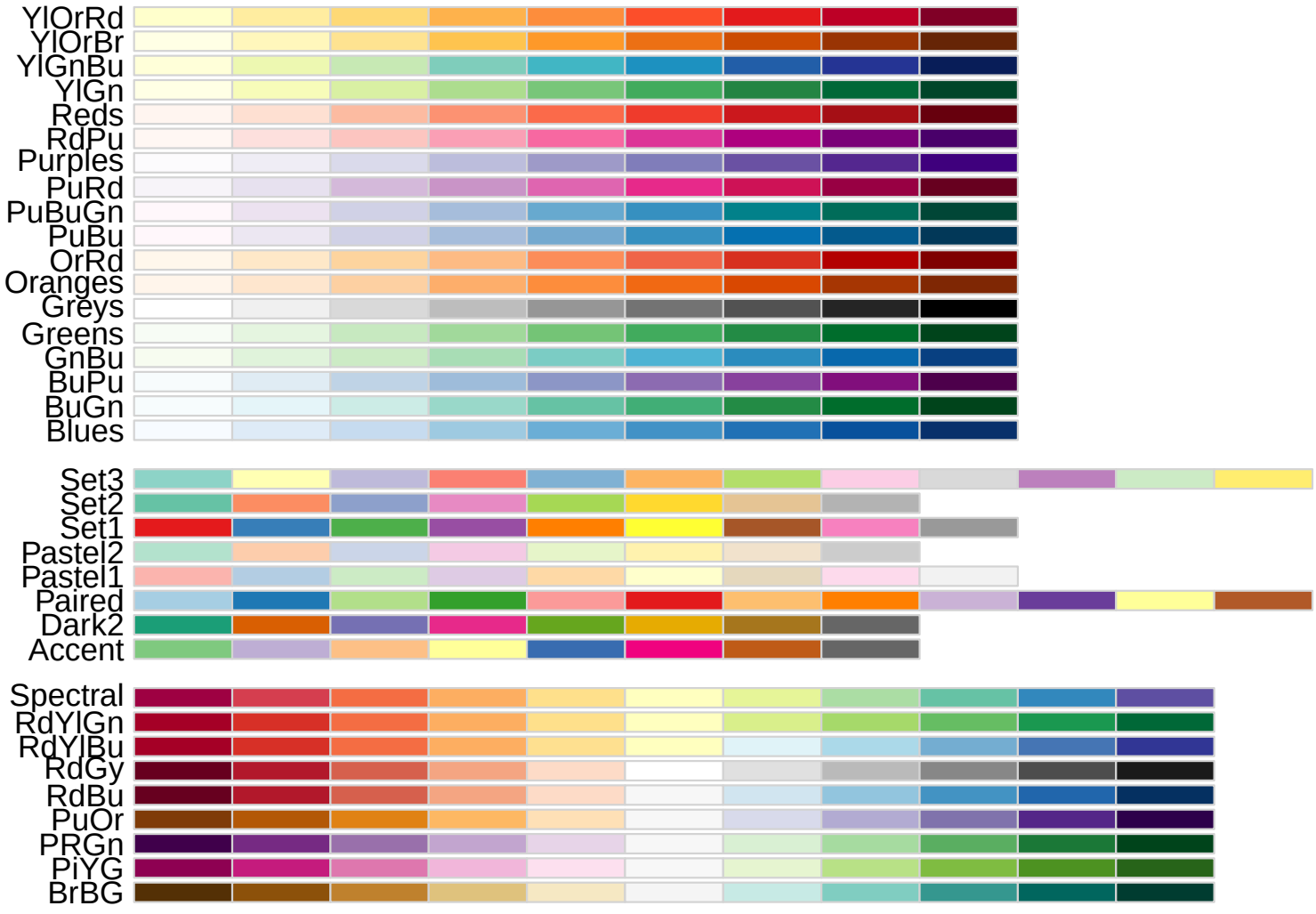
```
sp_c <- ggplot(df_data, aes(x=Avg_net_depth, y=Biomass,
                            color=factor(Year))) +
  geom_point(alpha=0.8) +
  xlab('Average net depth')
print(sp_c)
```



Discrete color scale: brewer palette

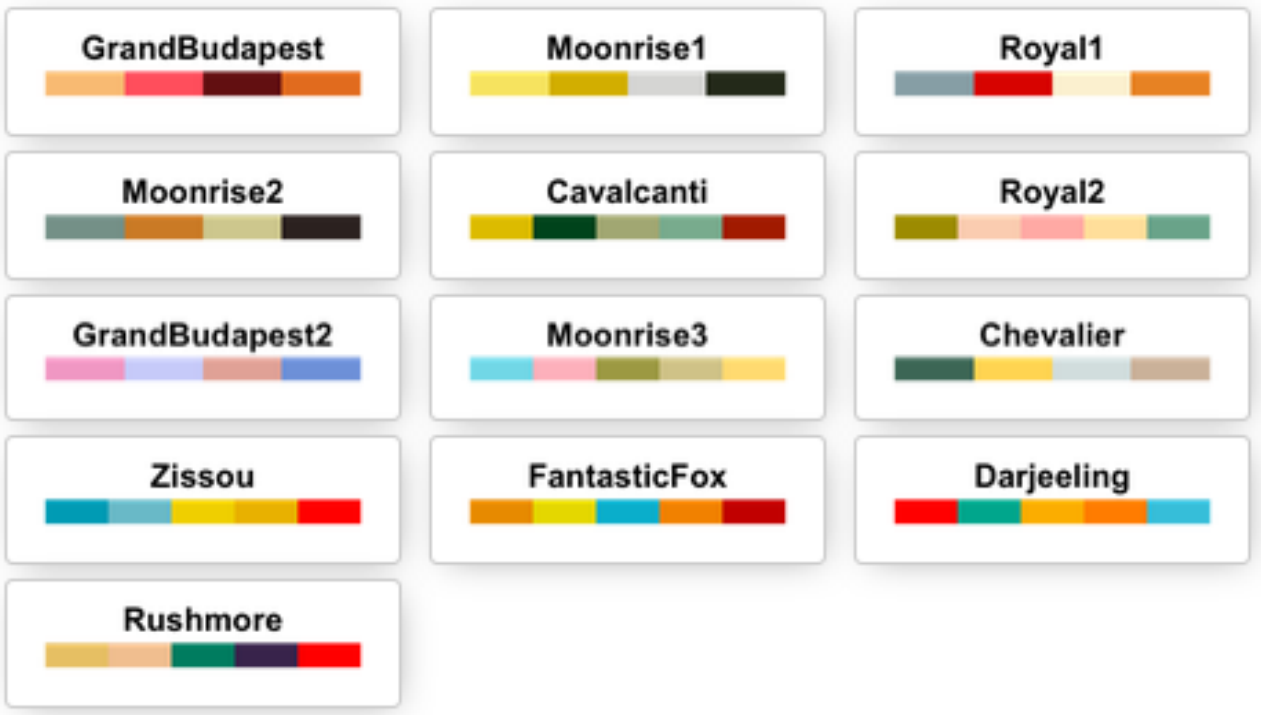
```
sp_c_brewer <- sp_c +
  scale_color_brewer(name='Year', palette="Dark2")
print(sp_c_brewer)
```

Discrete color scale: brewer palettes



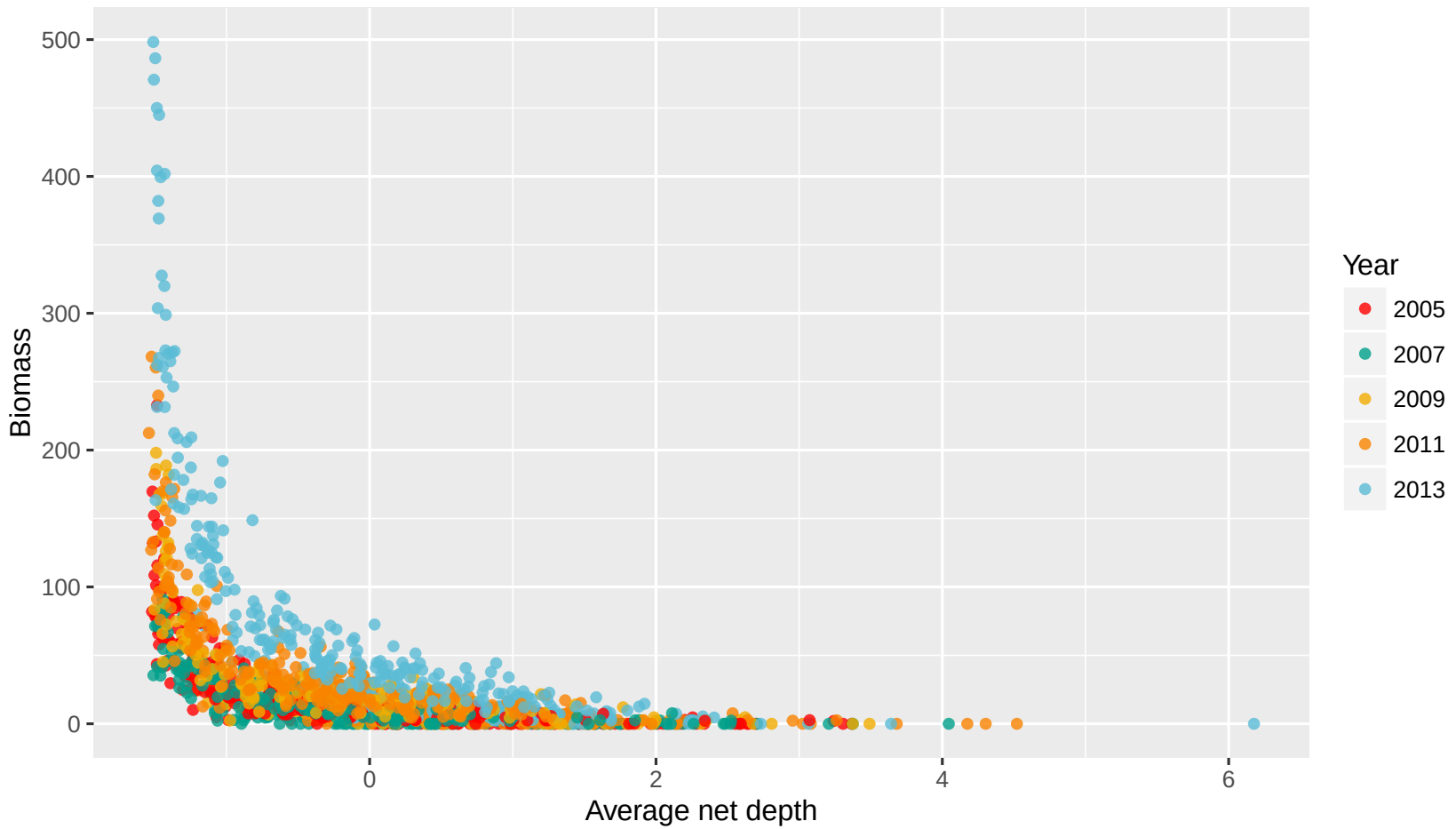
Discrete color scale: wesanderson palettes

```
library(wesanderson)
```



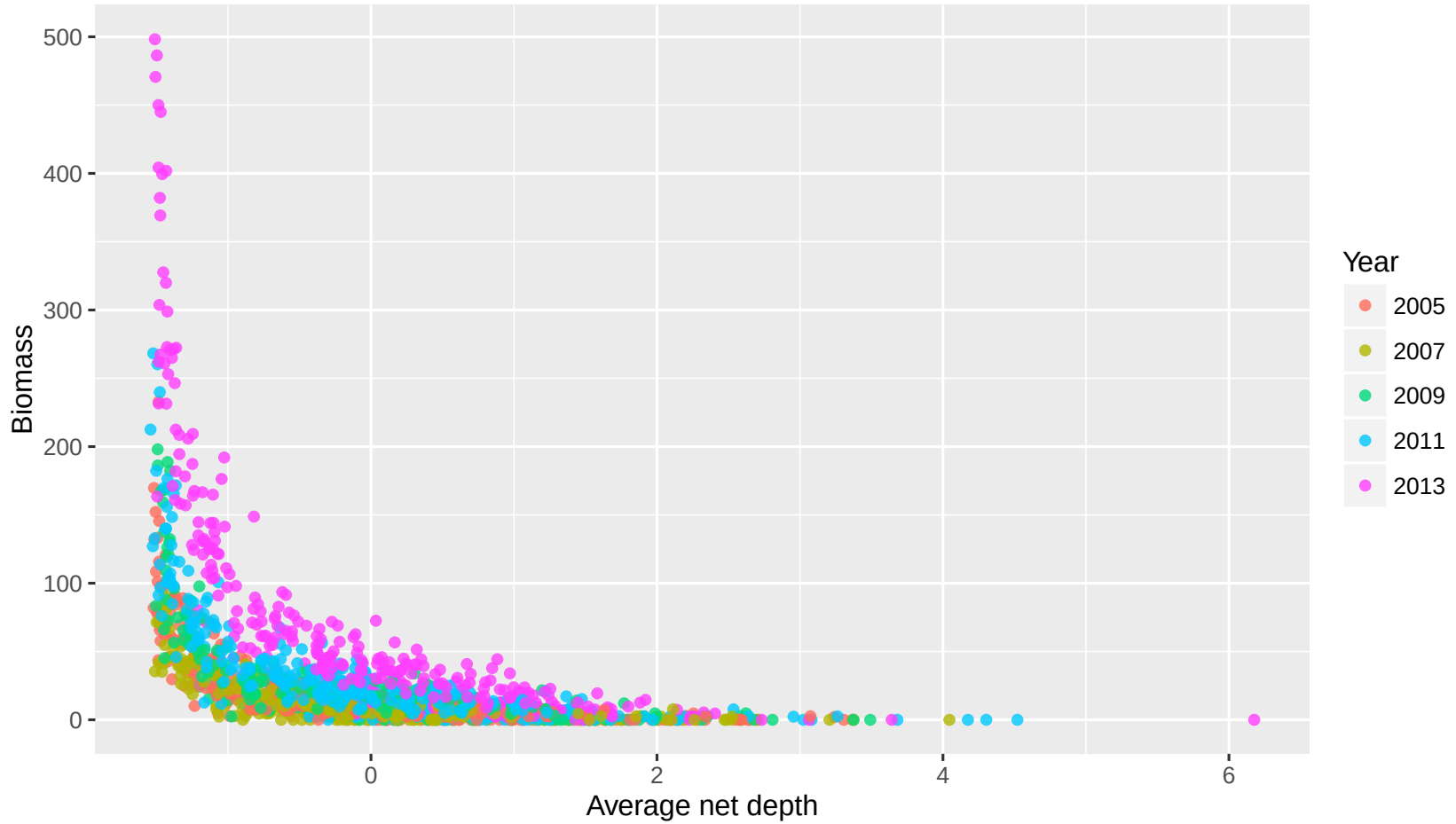
Discrete color scale: wesanderson palettes

```
sp_c_wanderson <- sp_c +
  scale_color_manual(name='Year', values=wes_palette(name="Darjeeling"))
print(sp_c_wanderson)
```



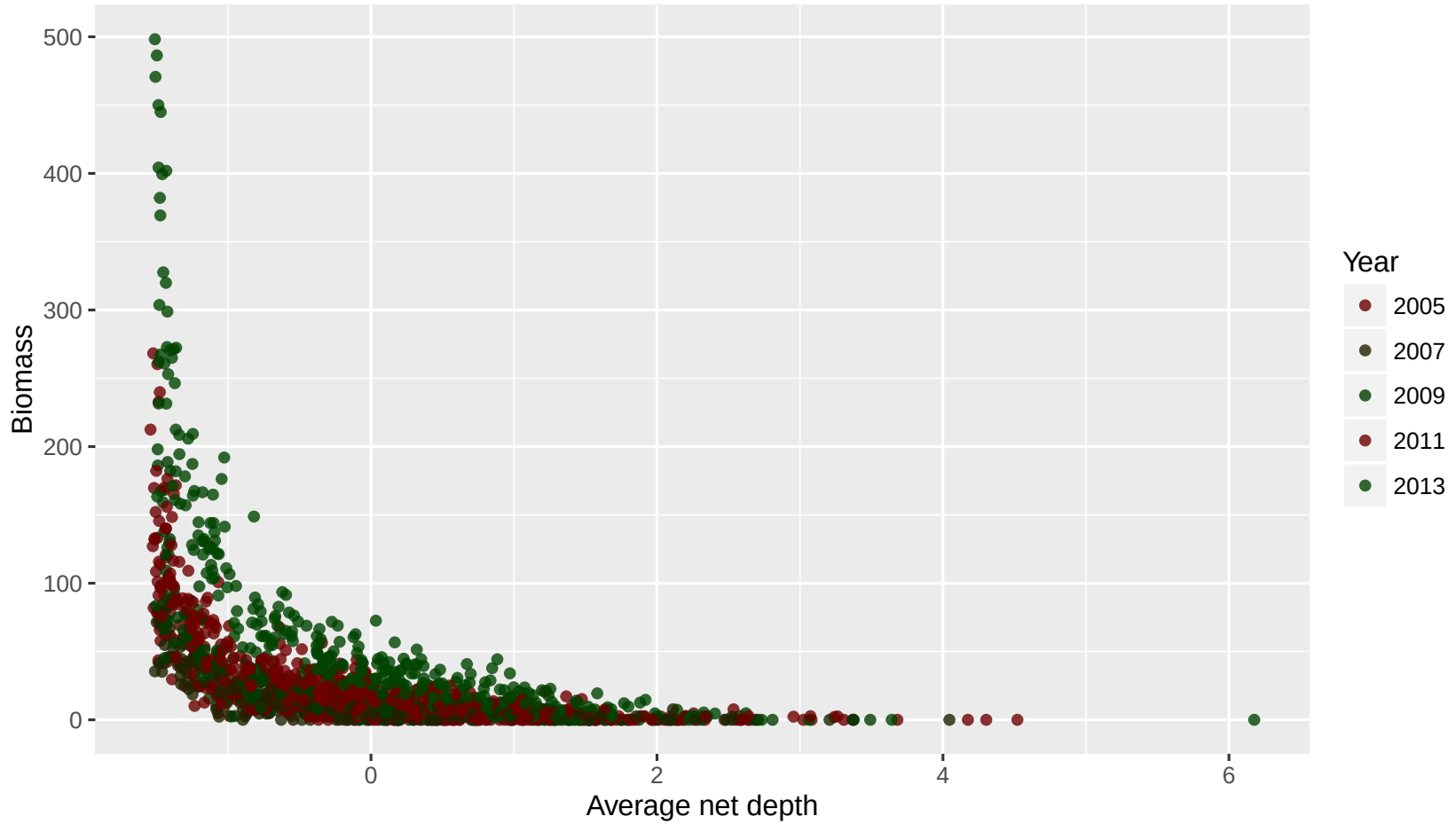
Discrete color scale: scale_colour_hue

```
# Adjust luminosity and chroma
sp_c_hue <- sp_c +
 scale_colour_hue(name='Year', l=70, c=150)
print(sp_c_hue)
```



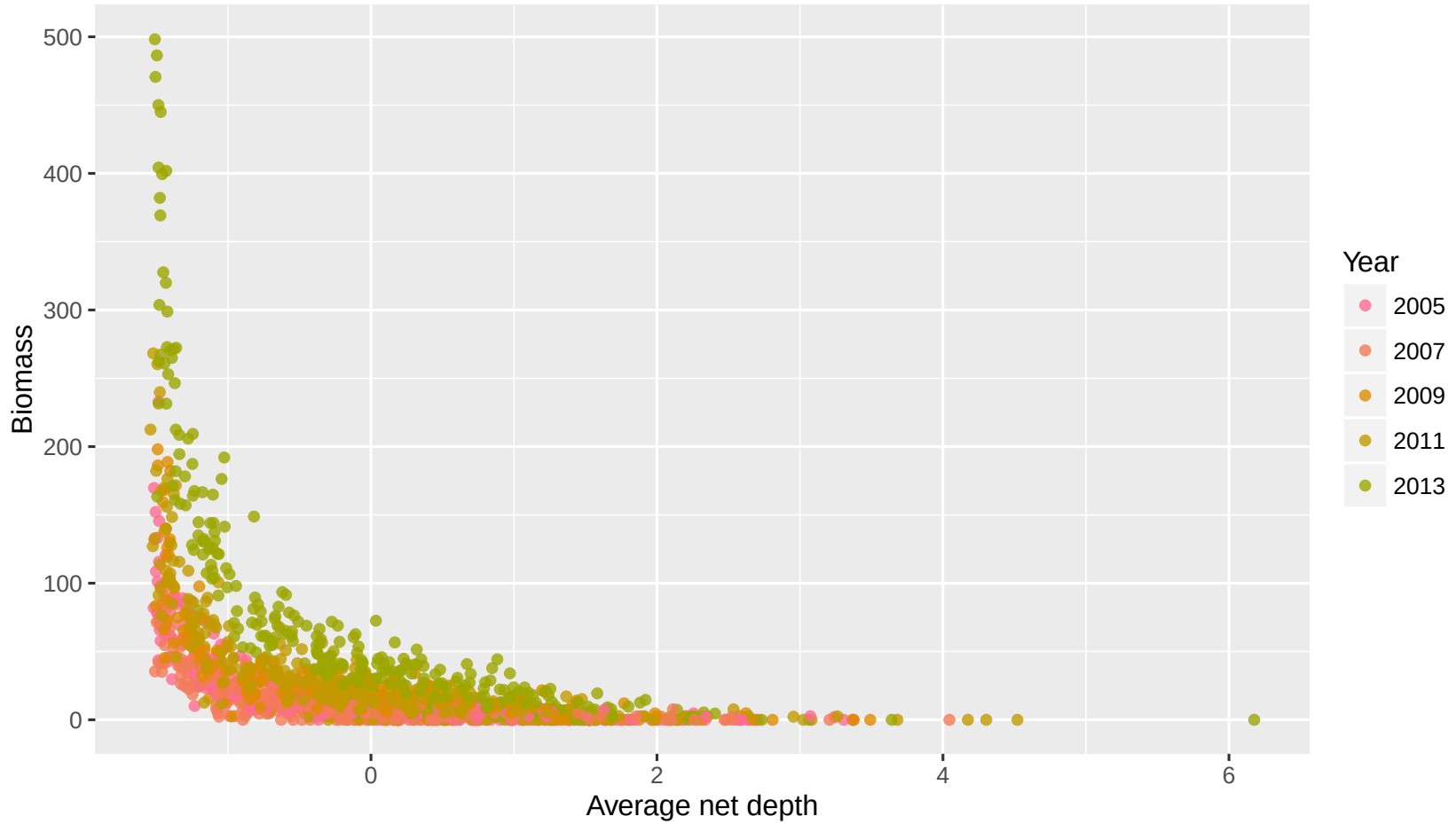
Discrete color scale: scale_colour_hue

```
# Adjust luminosity and chroma
sp_c_hue <- sp_c +
 scale_colour_hue(name='Year', l=10, c=150)
print(sp_c_hue)
```



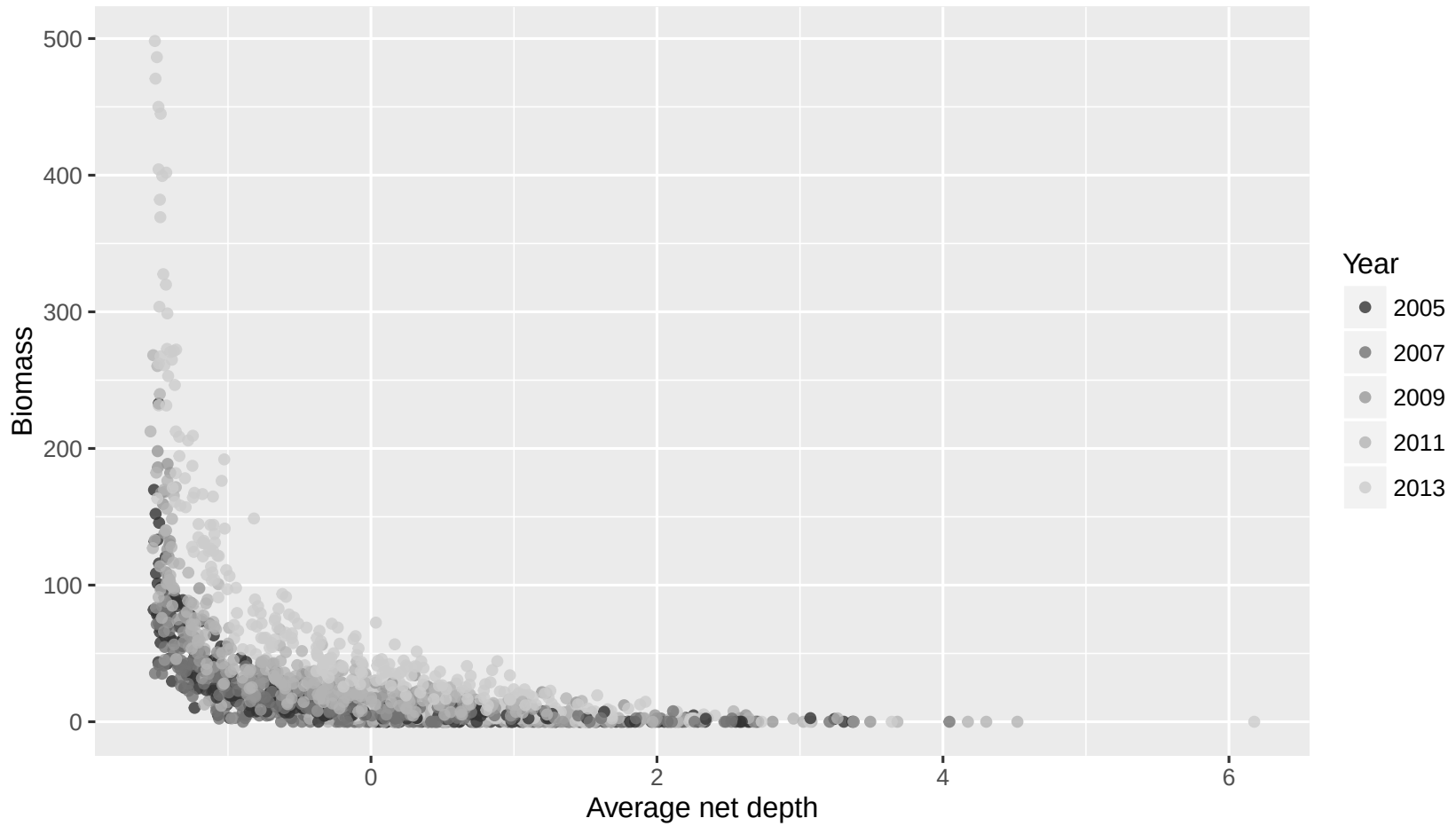
Discrete color scale: scale_colour_hue

```
# Change range of hues used
sp_c_hue <- sp_c +
 scale_colour_hue(name='Year', h=c(0, 90))
print(sp_c_hue)
```



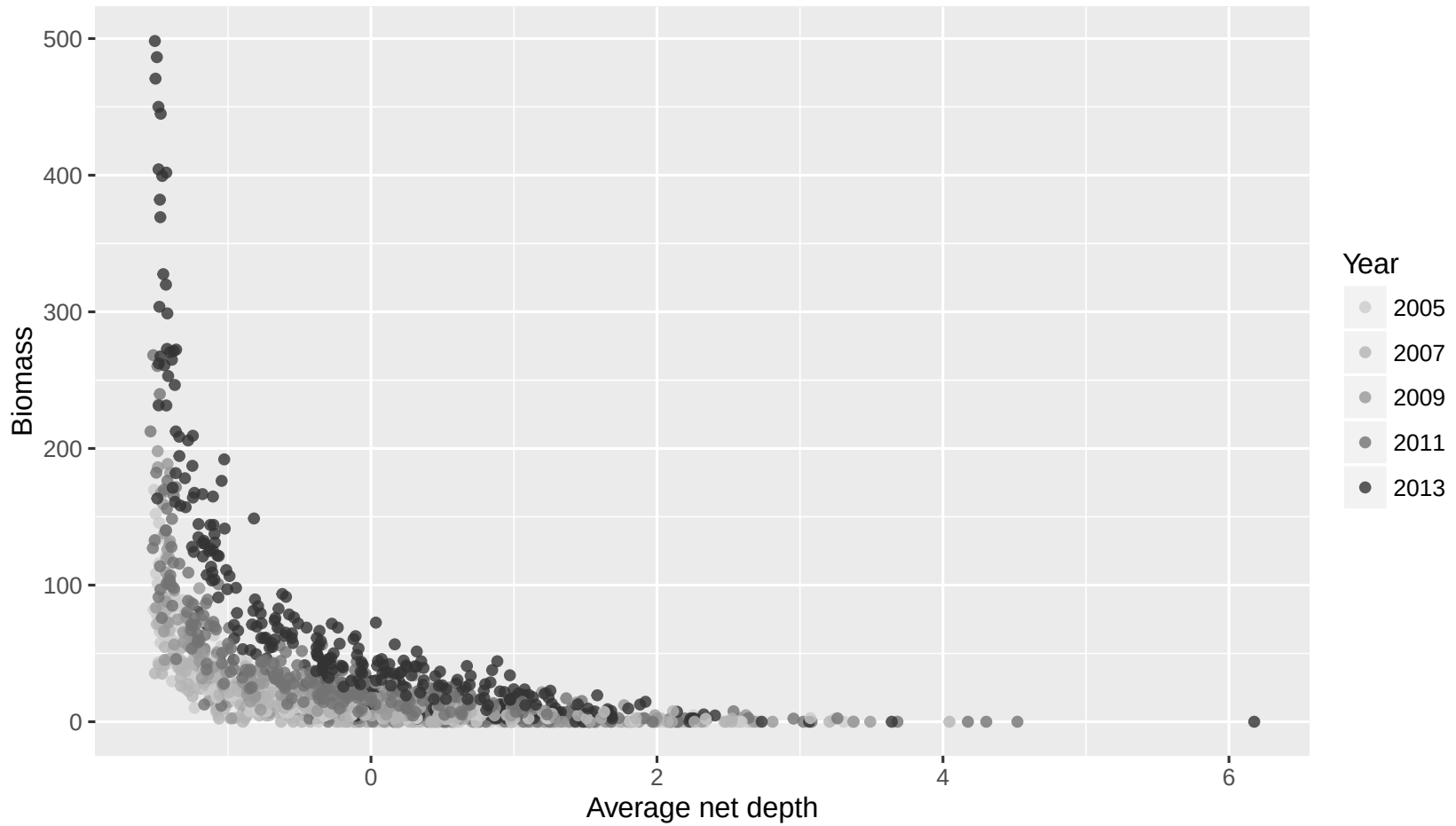
Discrete color scale: scale_color_grey

```
sp_c_grey <- sp_c +
  scale_color_grey(name='Year')
print(sp_c_grey)
```



Discrete color scale: scale_color_grey

```
sp_c_grey <- sp_c +
  scale_color_grey(name='Year', start=0.8, end=0.2)
print(sp_c_grey)
```

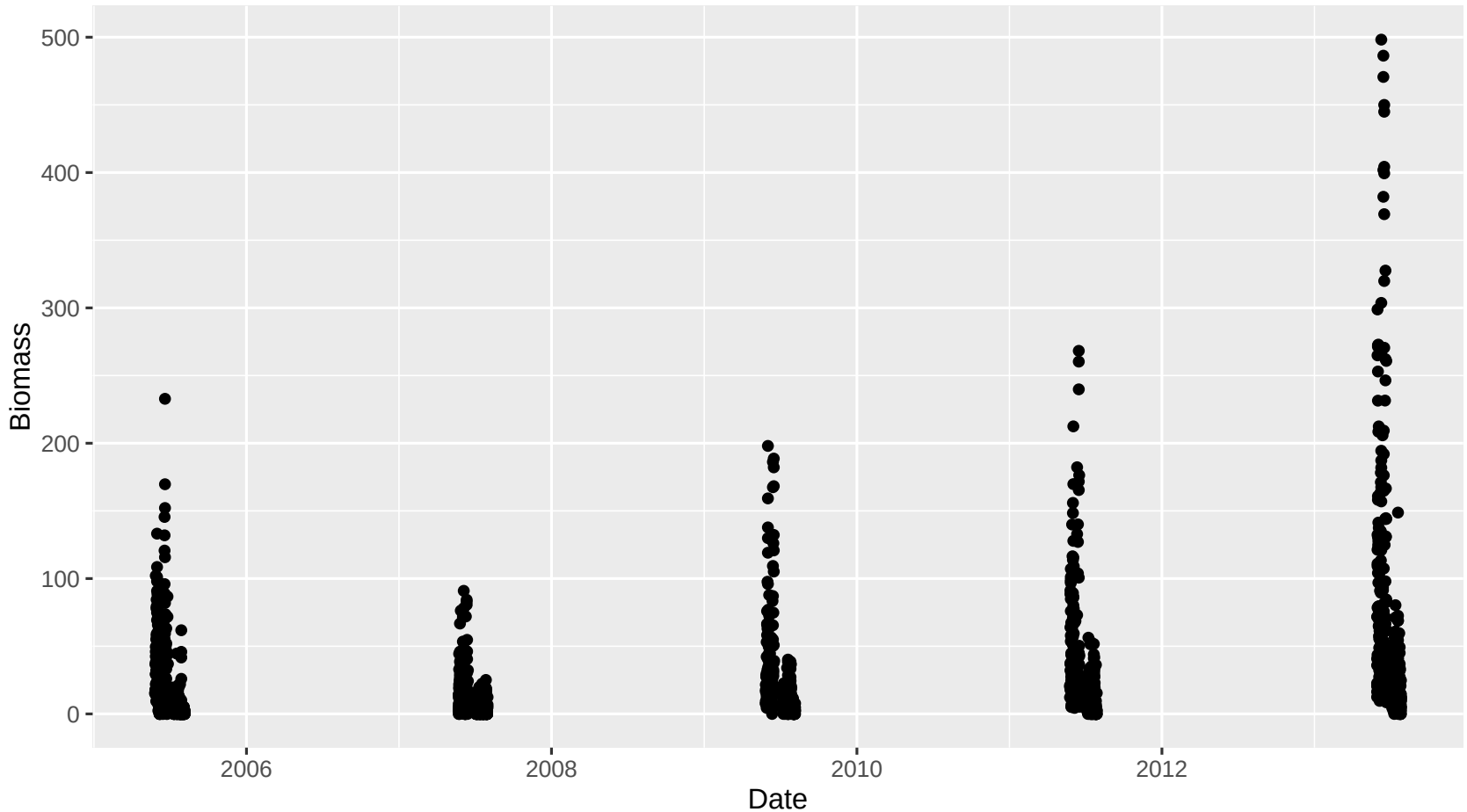


A quick overview of the ggplot2 types

- Points, as for a scatterplot geom_point()
- Lines geom_line()
- Histogram geom_freqpoly(geom_histogram, stat_bin)
- Boxplot geom_boxplot()
- Polygon geom_polygon
- Draw rectangles geom_raster
- Smooth density estimate geom_density
- Ribbons and area plots geom_ribbon(geom_area)
- Map ggmap()

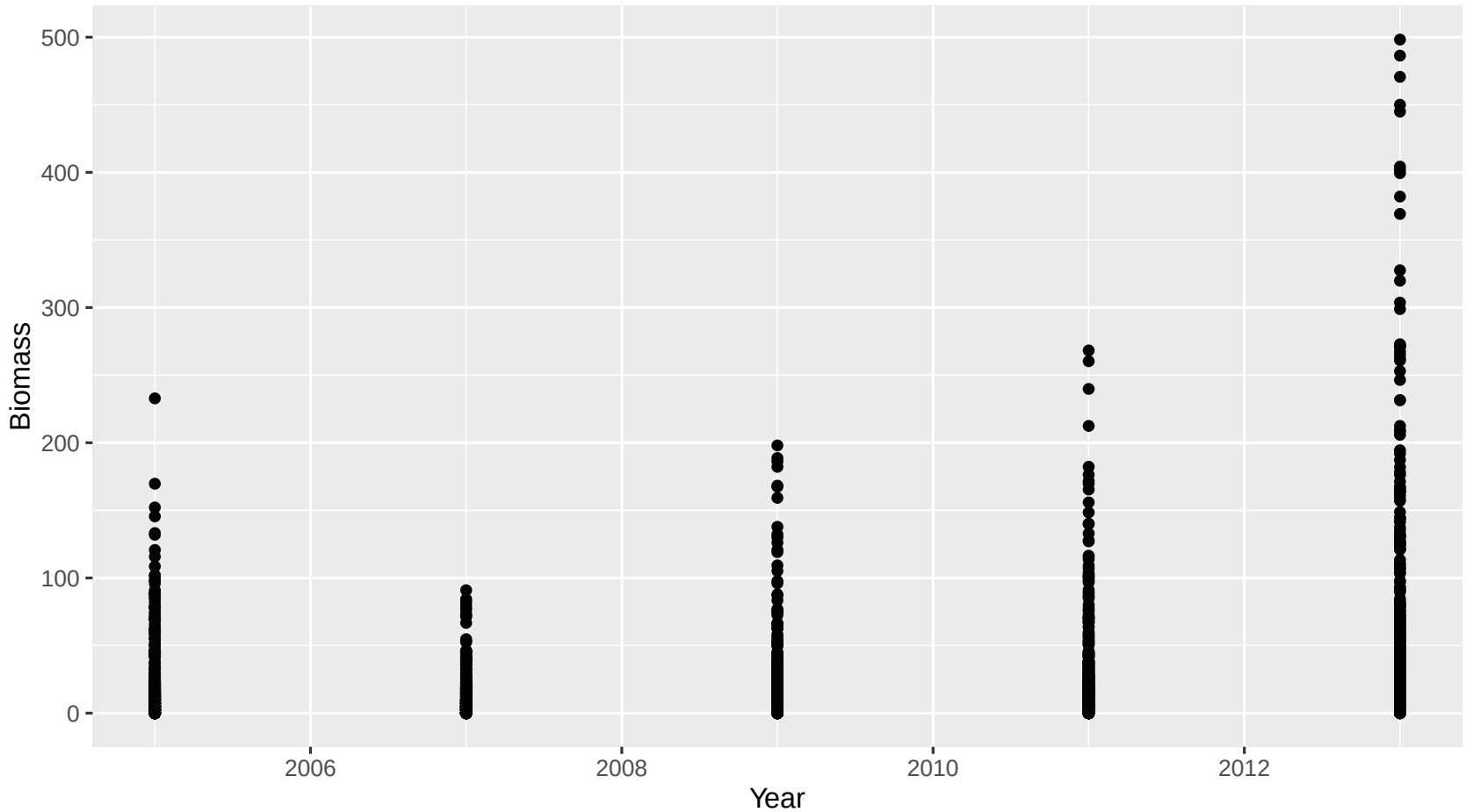
Time series

```
TSplot <- ggplot(data=df_data, aes(x=Date, y=Biomass)) +
  geom_point()
print(TSplot)
```



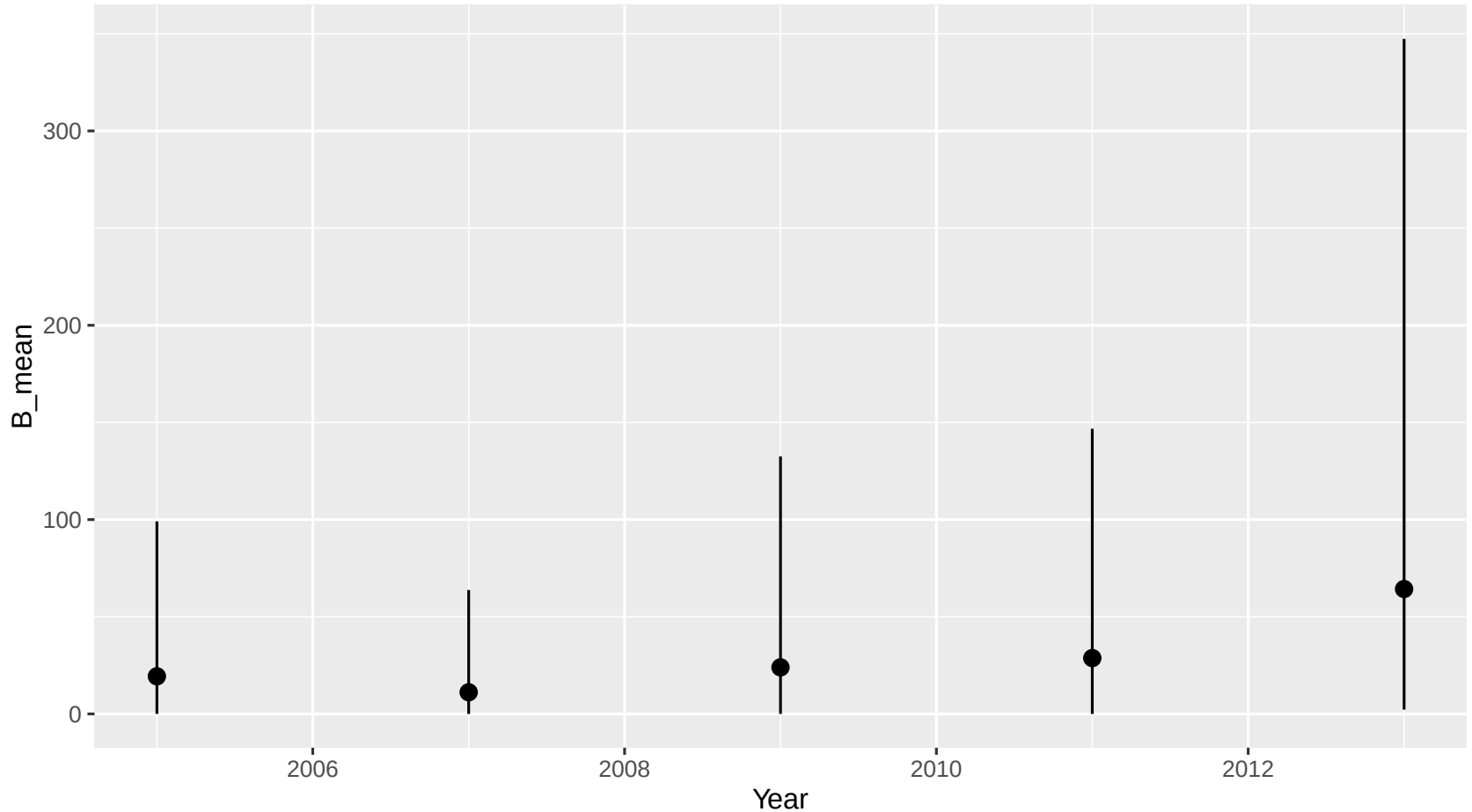
Time series

```
TSplot_year <- ggplot(data=df_data, aes(x=Year, y=Biomass)) +
  geom_point()
print(TSplot_year)
```



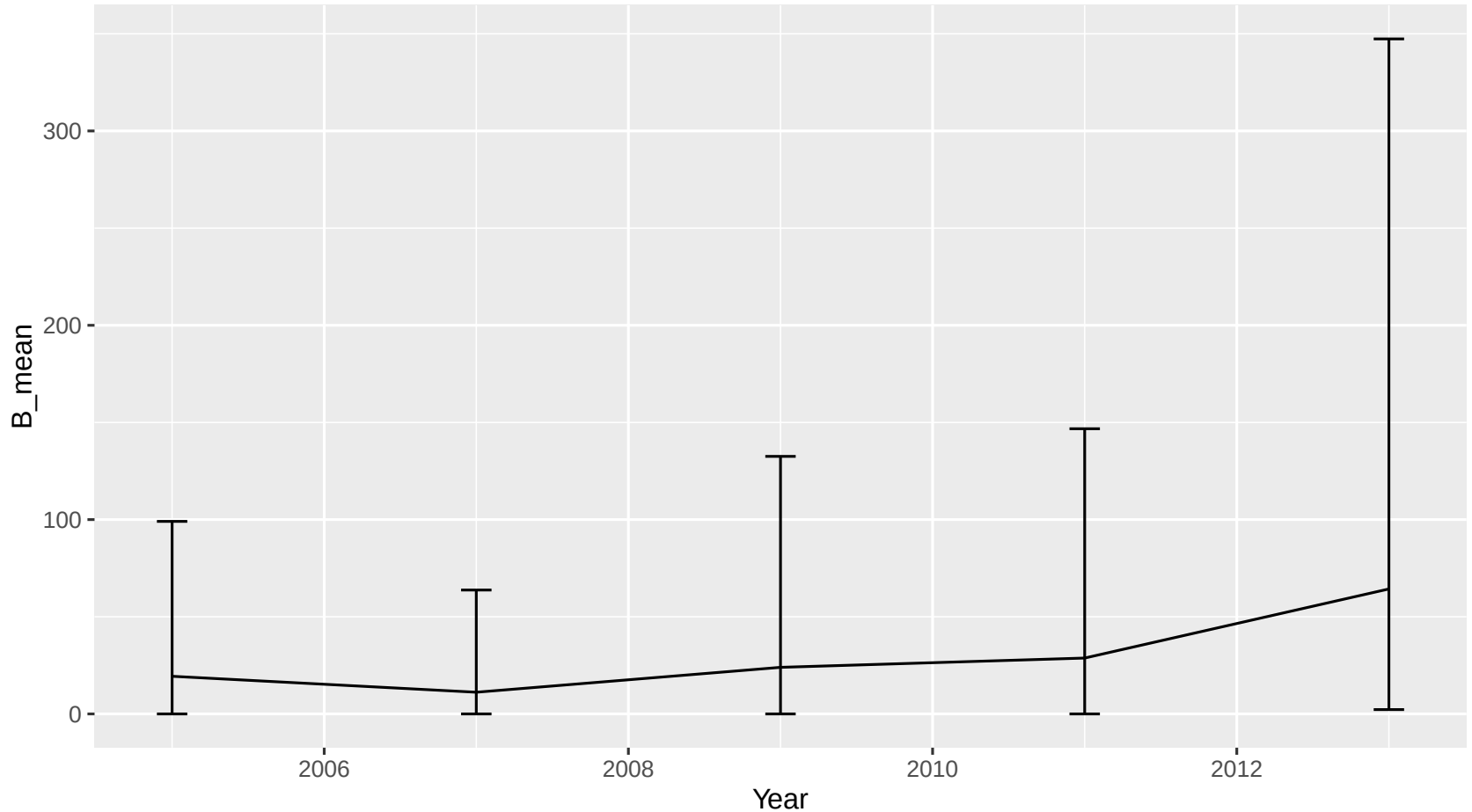
Time series with error bars

```
TSplot_i95 <- ggplot(data=df_data_summary, aes(x=Year, y=B_mean)) +
  geom_point() +
  geom_pointrange(aes(ymin = B_q025, ymax = B_q975))
print(TSplot_i95)
```



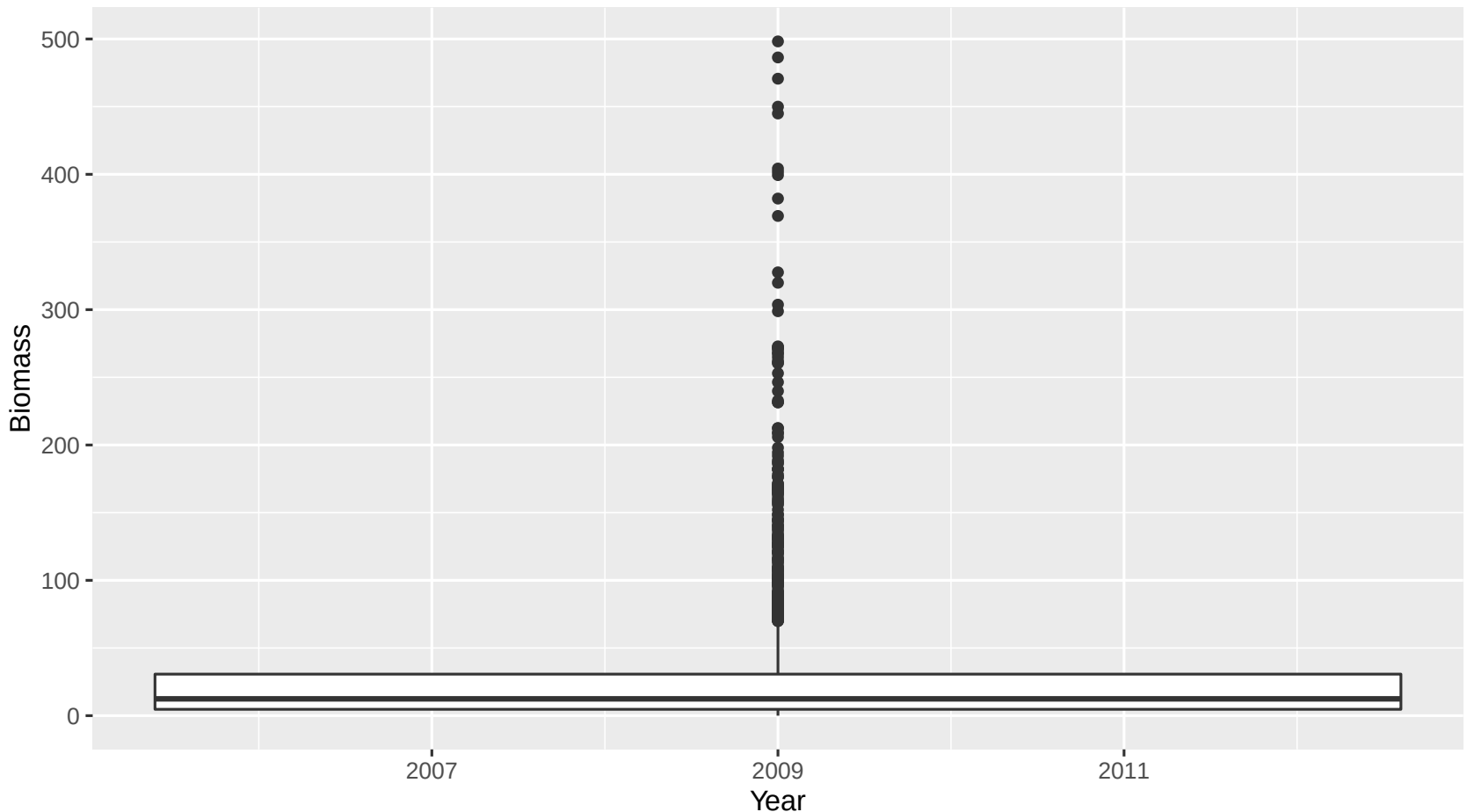
Time series with error bars

```
TSplot_errori95 <- ggplot(data=df_data_summary, aes(x=Year, y=B_mean)) +
  geom_line() +
  geom_errorbar(aes(ymin = B_q025, ymax = B_q975), width = 0.2)
print(TSplot_errori95)
```



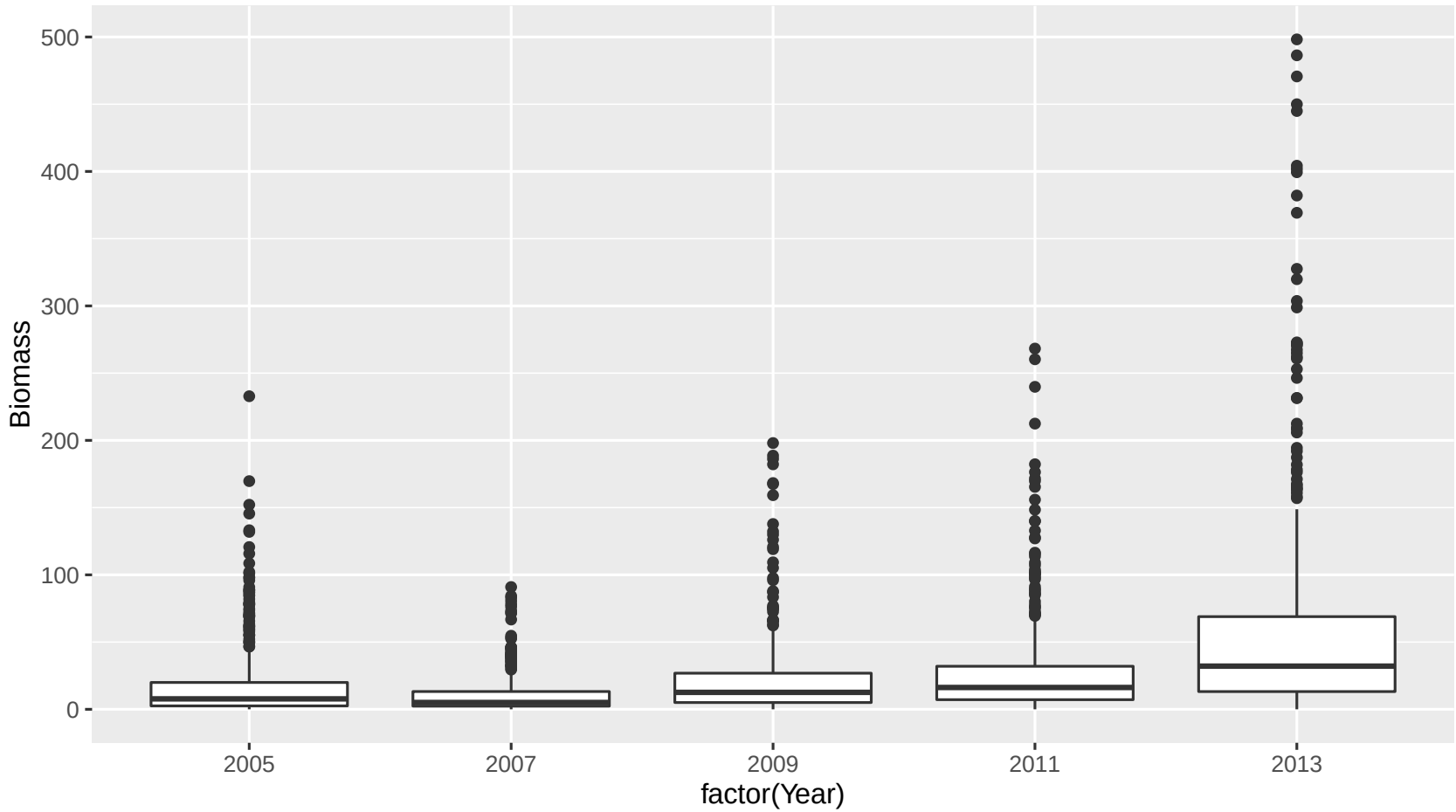
Boxplot

```
boxplot_TS <- ggplot(data=df_data, aes(x=Year, y=Biomass)) +
  geom_boxplot()
print(boxplot_TS)

## Warning:  Continuous x aesthetic -- did you forget aes(group=...)?
```



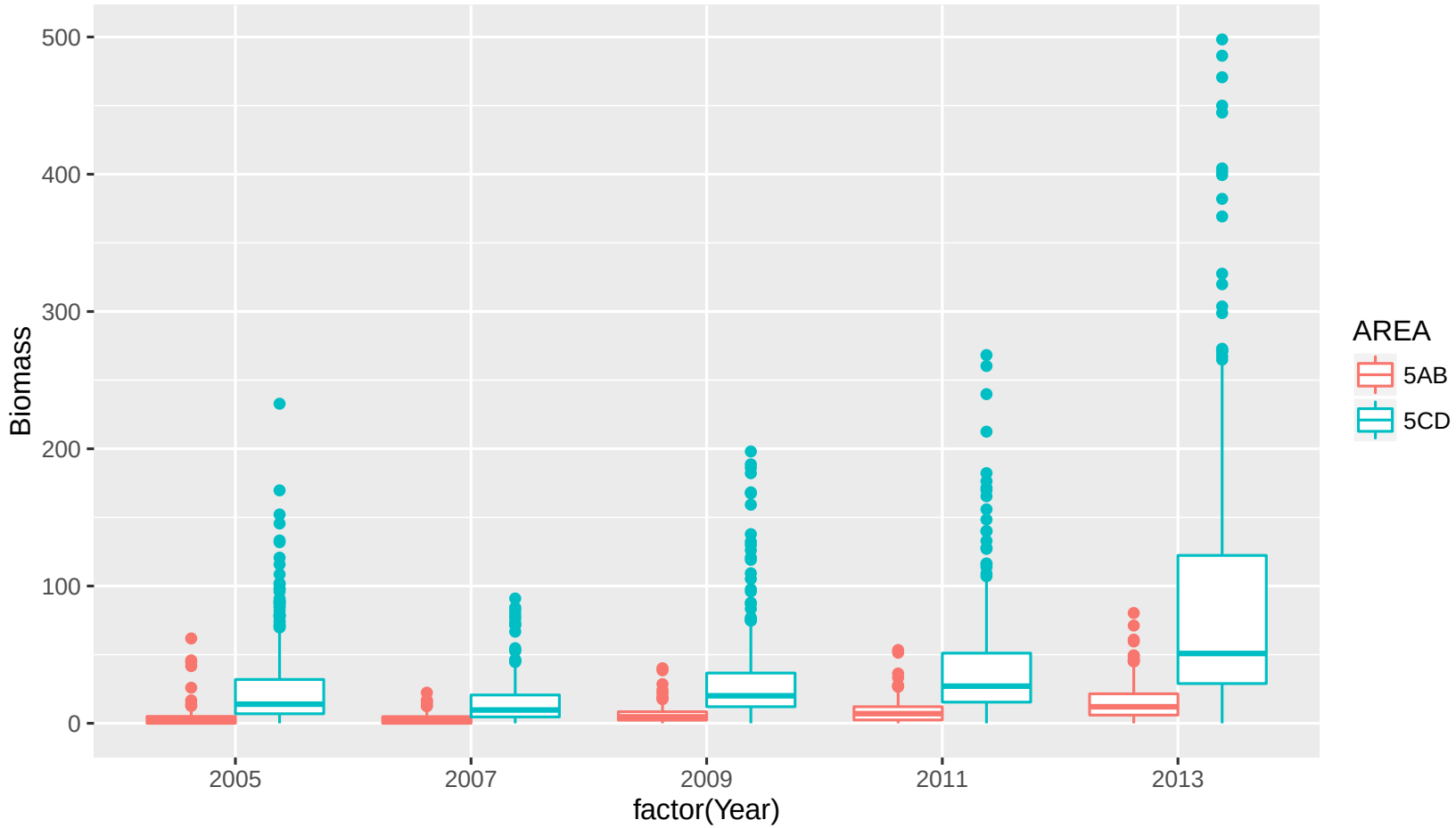
Boxplot with Year as a factor

```
boxplot_TSf <- ggplot(data=df_data, aes(x=factor(Year), y=Biomass)) +
  geom_boxplot()
print(boxplot_TSf)
```



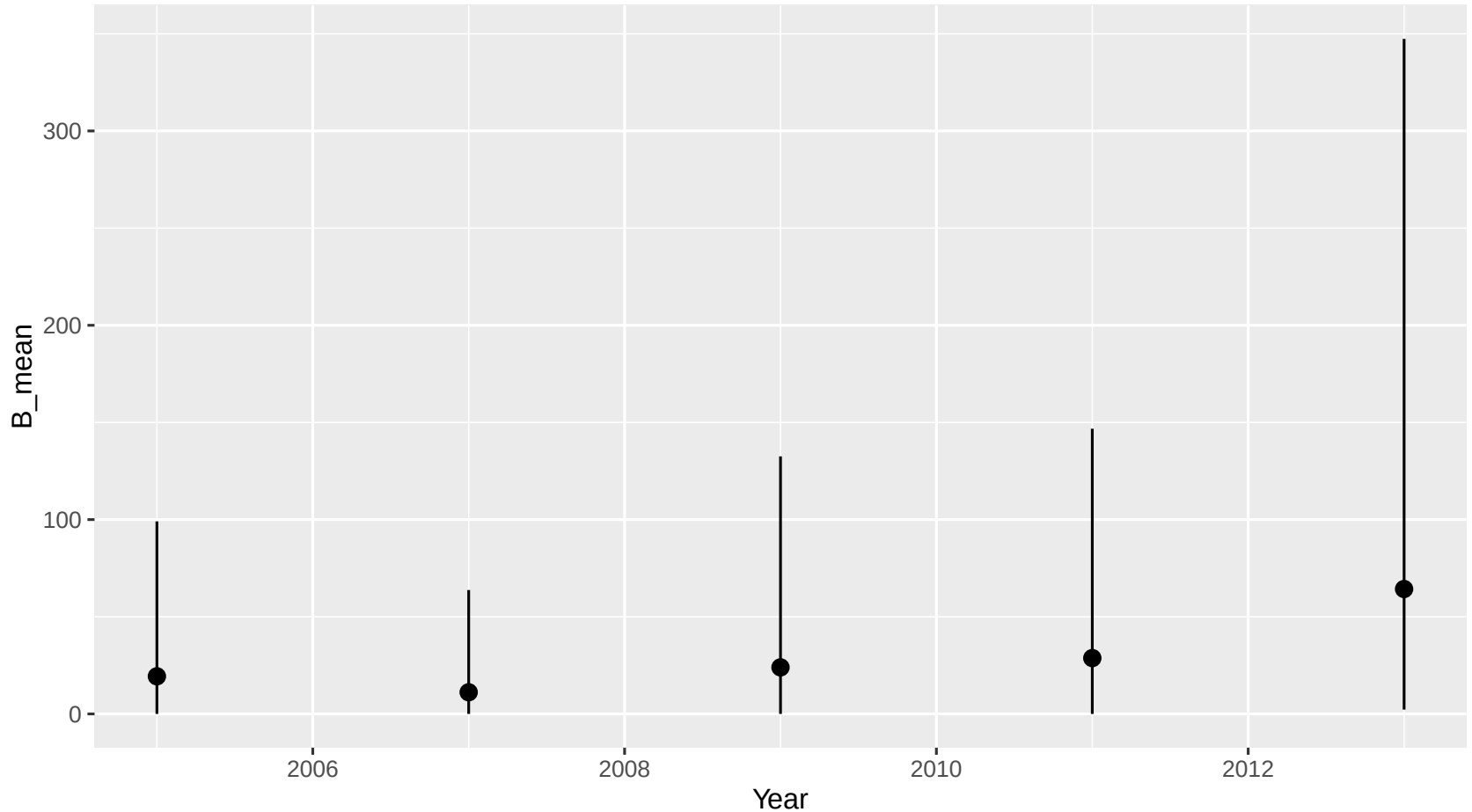
Boxplot with Year and Area as a factor

```
boxplot_TS_AREA <- ggplot(data=df_data, aes(x=factor(Year), y=Biomass,
                                            colour=AREA)) +
  geom_boxplot()
print(boxplot_TS_AREA)
```



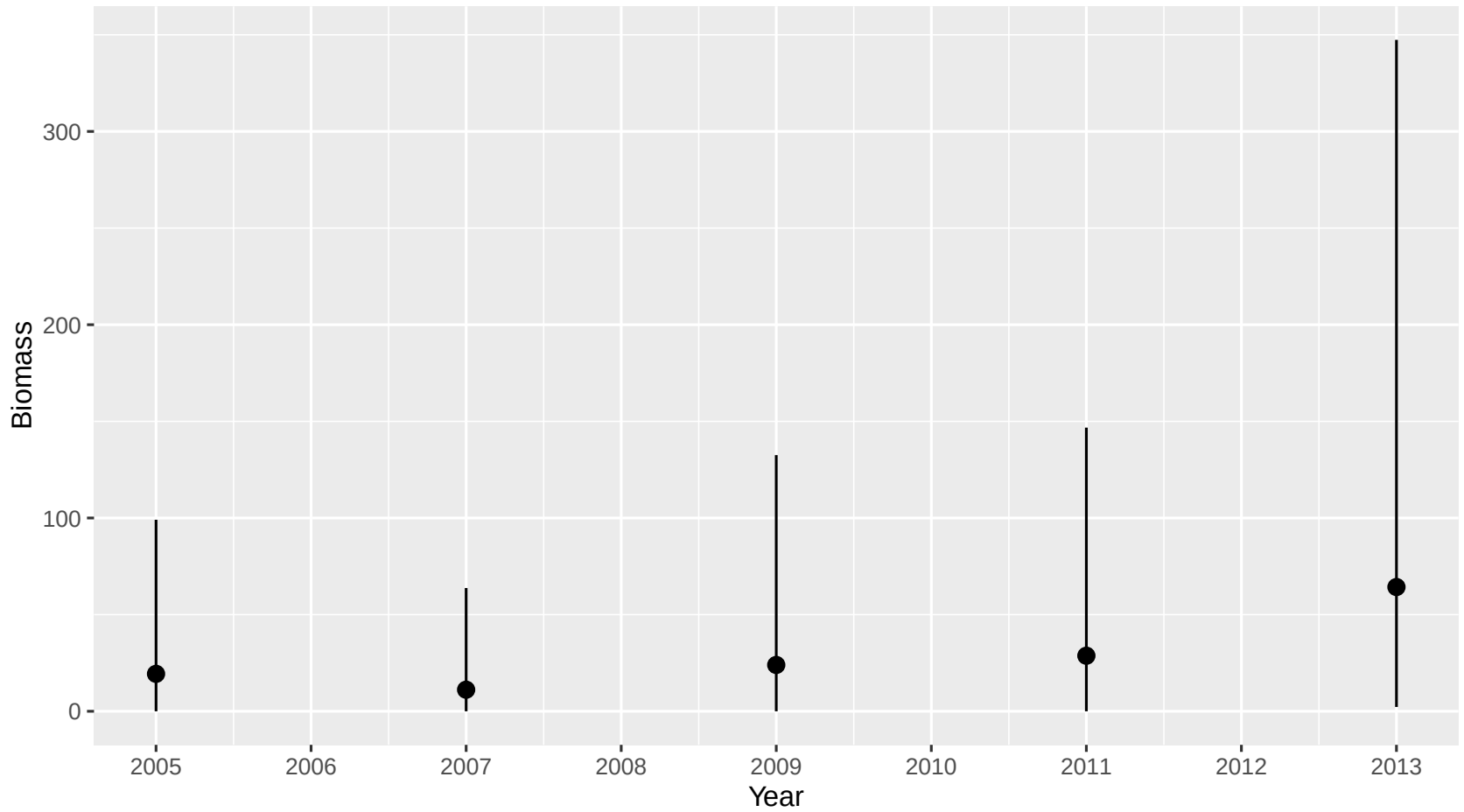
Time series with error bars: Improvement

```
TSplot_i95 <- ggplot(data=df_data_summary, aes(x=Year, y=B_mean)) +
  geom_point() +
  geom_pointrange(aes(ymin = B_q025, ymax = B_q975))
print(TSplot_i95)
```



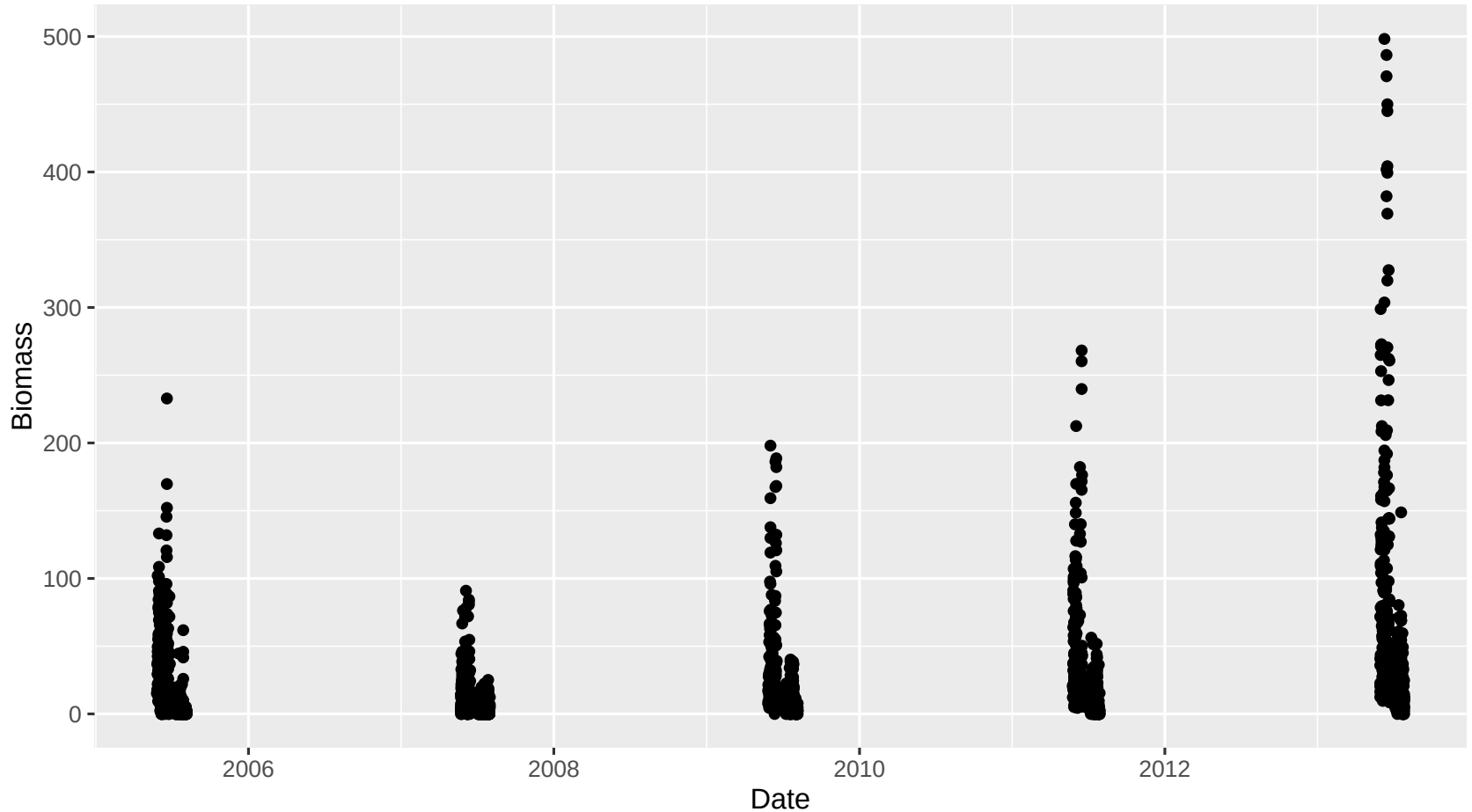
Time series with error bars: Improvement

```
TSplot_i95 <- ggplot(data=df_data_summary, aes(x=Year, y=B_mean)) +
  geom_point() + geom_pointrange(aes(ymin = B_q025, ymax = B_q975)) +
  ylab('Biomass') +
  scale_x_continuous(name = 'Year', breaks = seq(2005, 2013, by = 1))
print(TSplot_i95)
```



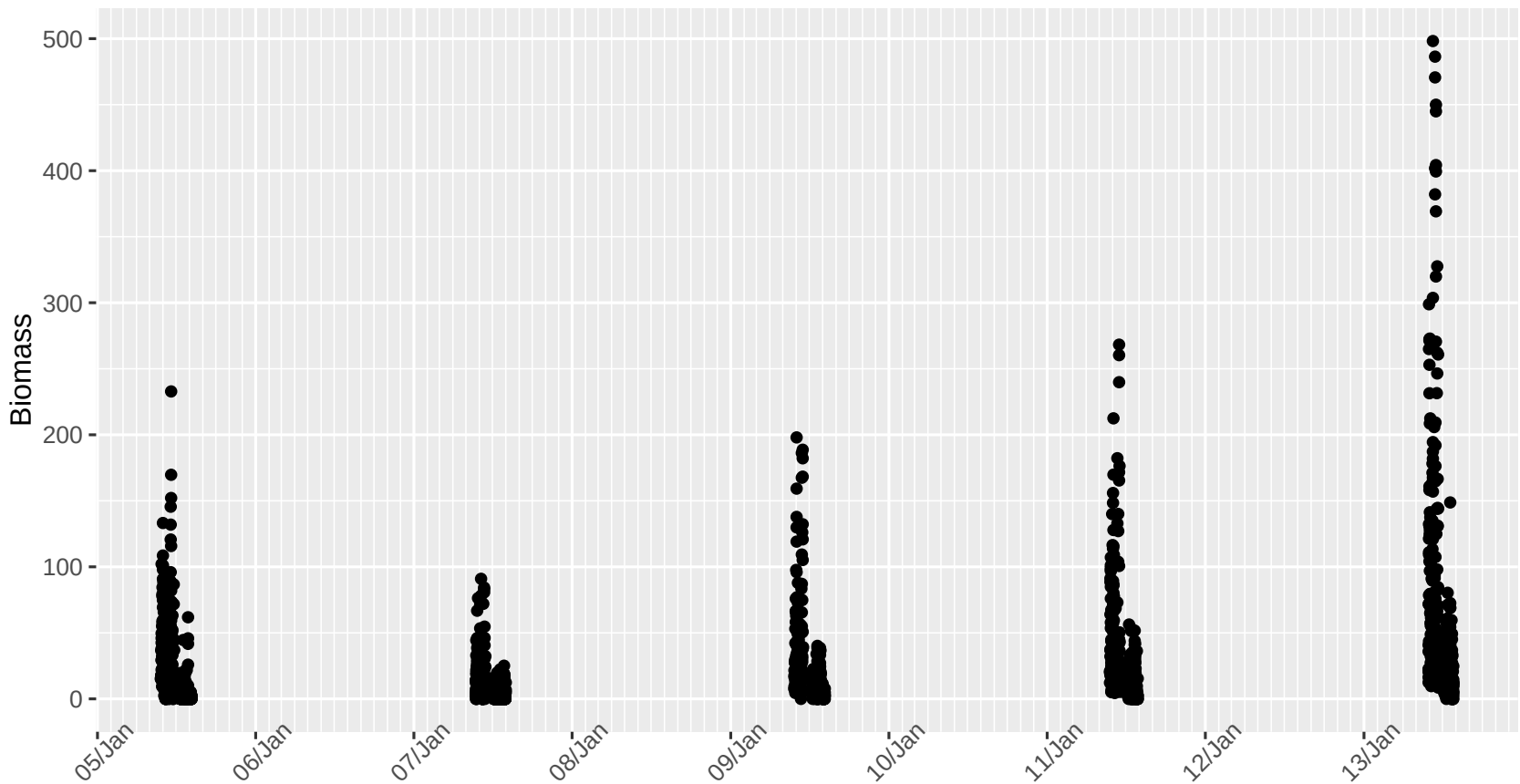
Time series: x-axis as date

```
TSplot <- ggplot(data=df_data, aes(x=Date, y=Biomass)) +
  geom_point() +
  ylab('Biomass')
print(TSplot)
```



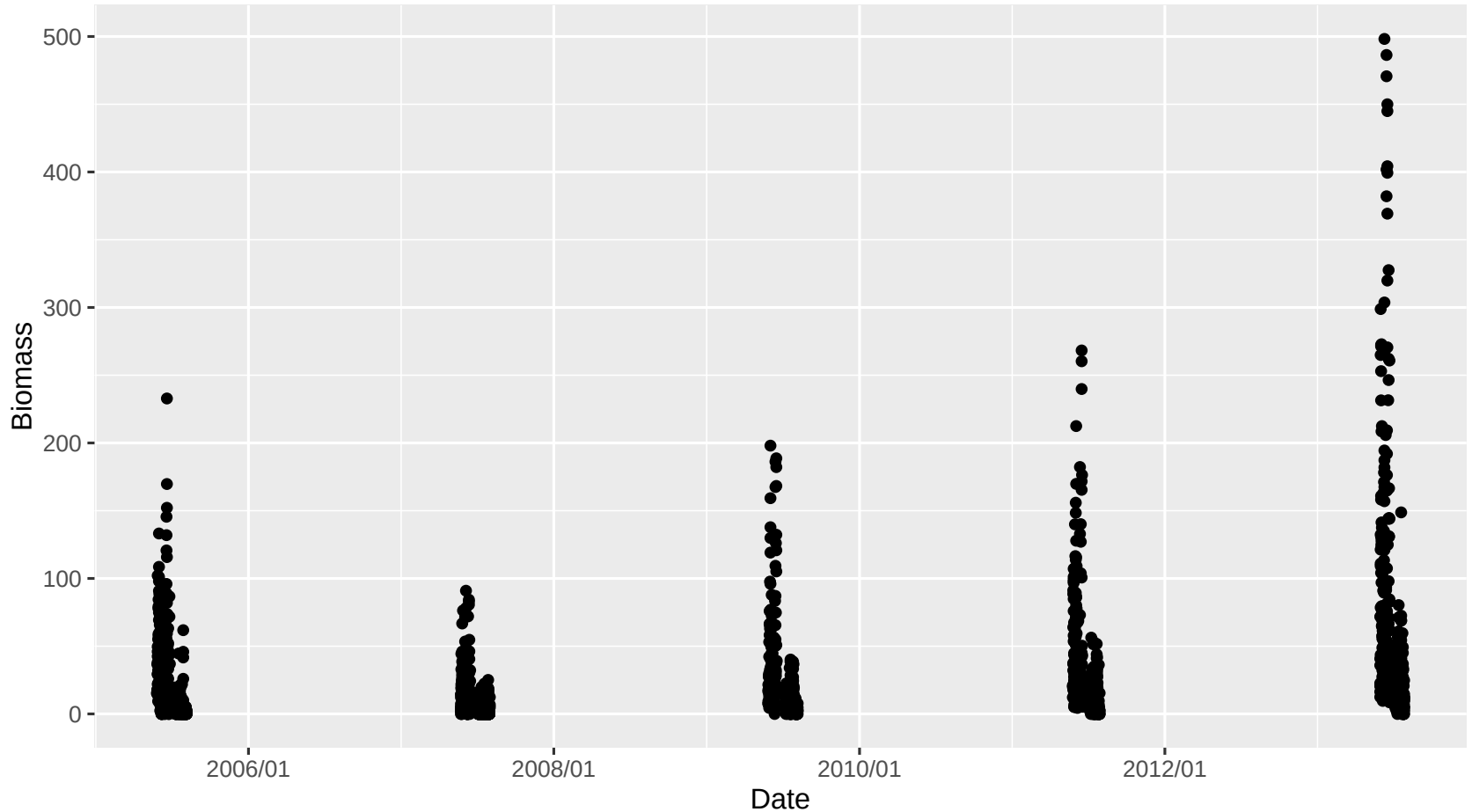
Time series: x-axis as date

```
# Months only
TSplotm <- TSplot +
  scale_x_date(labels = date_format("%y/%b"), date_breaks="1 year",
               date_minor_breaks = "1 month") +
  theme(axis.text.x = element_text(angle=45))
  print(TSplotm)
```



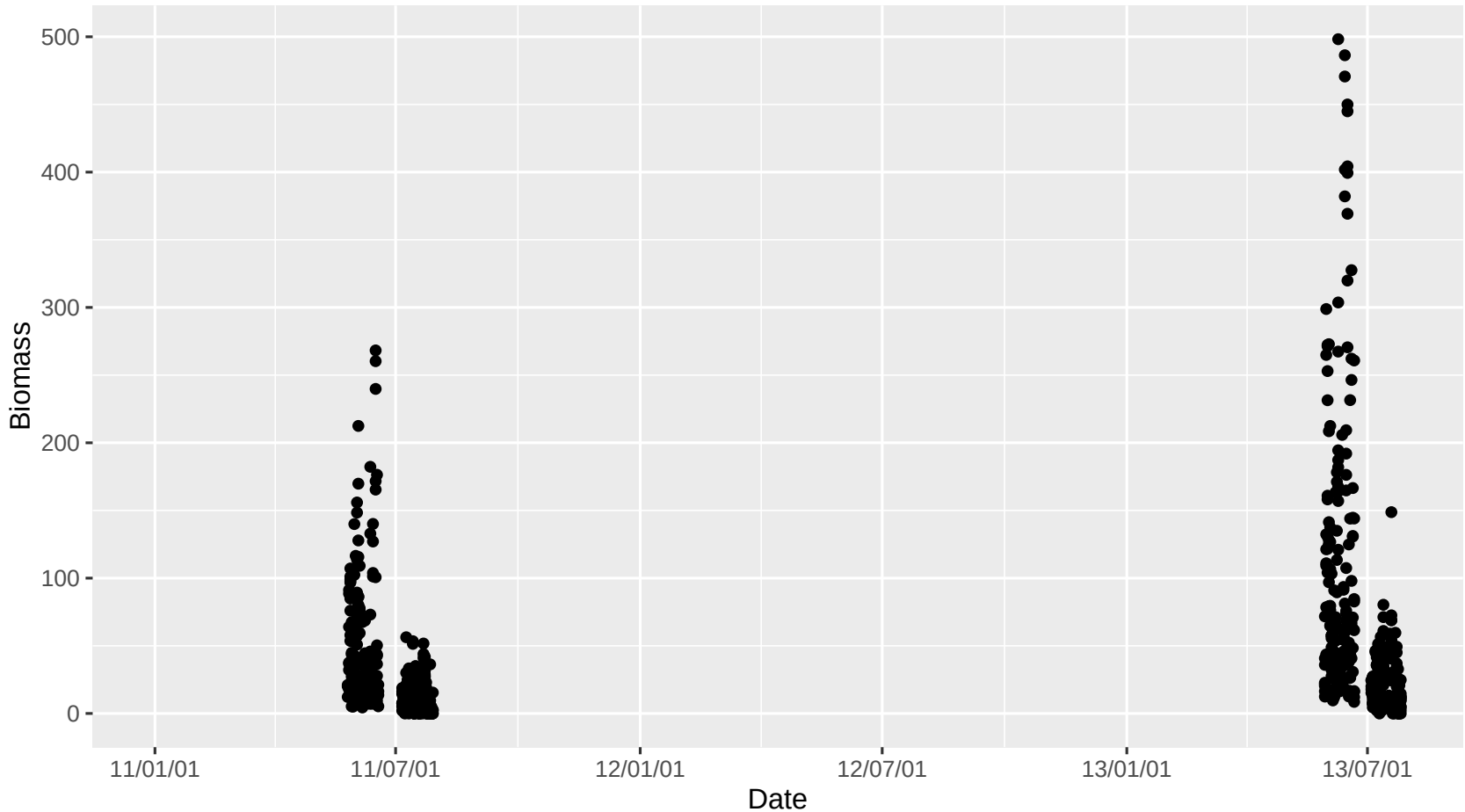
Time series: x-axis as date

```
# Format : Week
TSplotym <- TSplot +
  scale_x_date(labels = date_format("%Y/%m"))
  print(TSplotw)
```



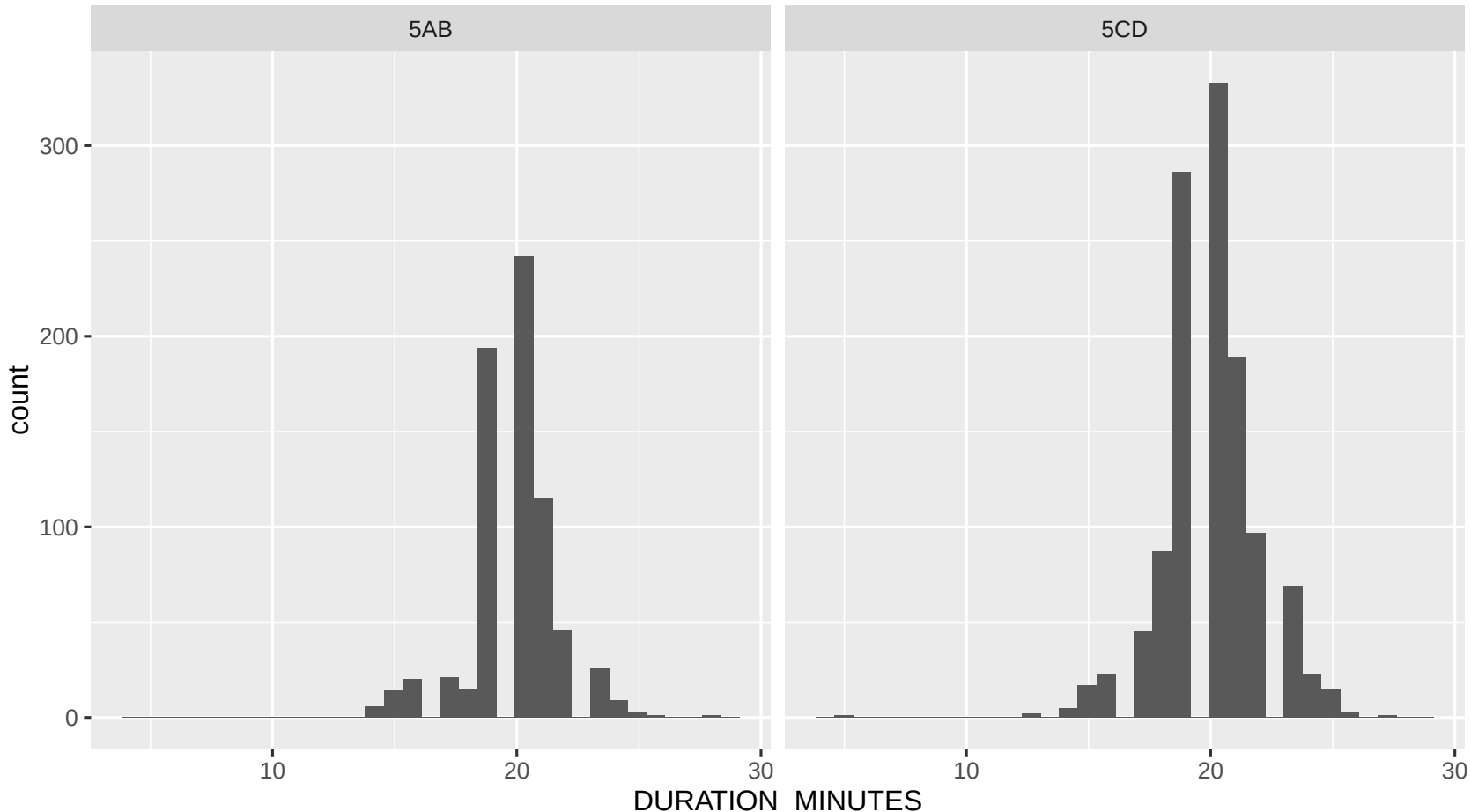
Time series: x-axis as date

```
# Format : Year/month/day
lmin <- as.Date("2011-1-1"); lmax <- max(df_data$Date)
TSplotymdl <- TSplot +
  scale_x_date(labels = date_format("%y/%m/%d"), limits = c(lmin, lmax))
  print(TSplotymdl)
```



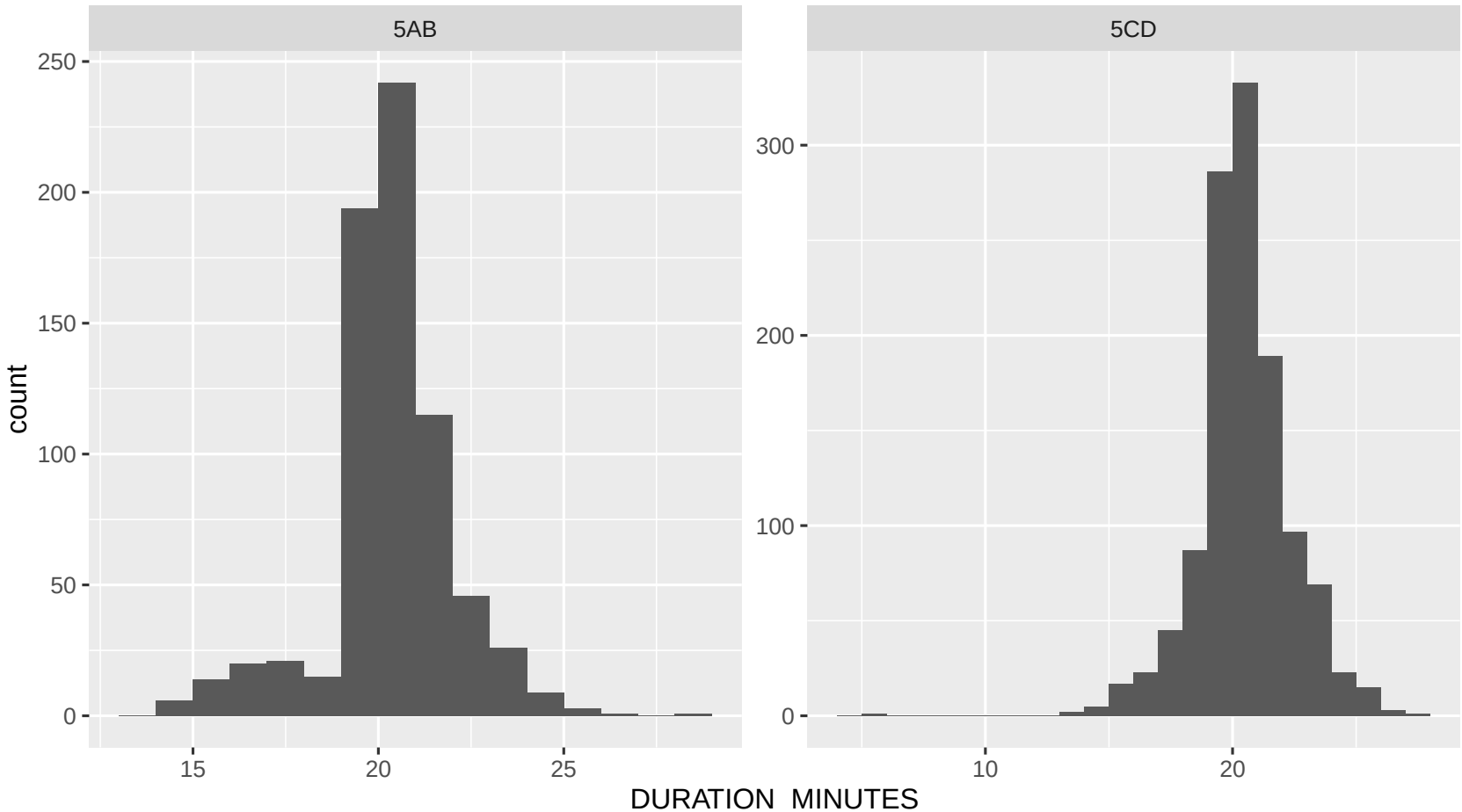
Faceting: facet_wrap

```
fw1 <- ggplot(data=df_data, aes(x=DURATION_MINUTES)) +
  geom_histogram() + facet_wrap(~ AREA)
print(fw1)

## `stat_bin()` using `bins = 30`.  Pick better value with `binwidth`.
```



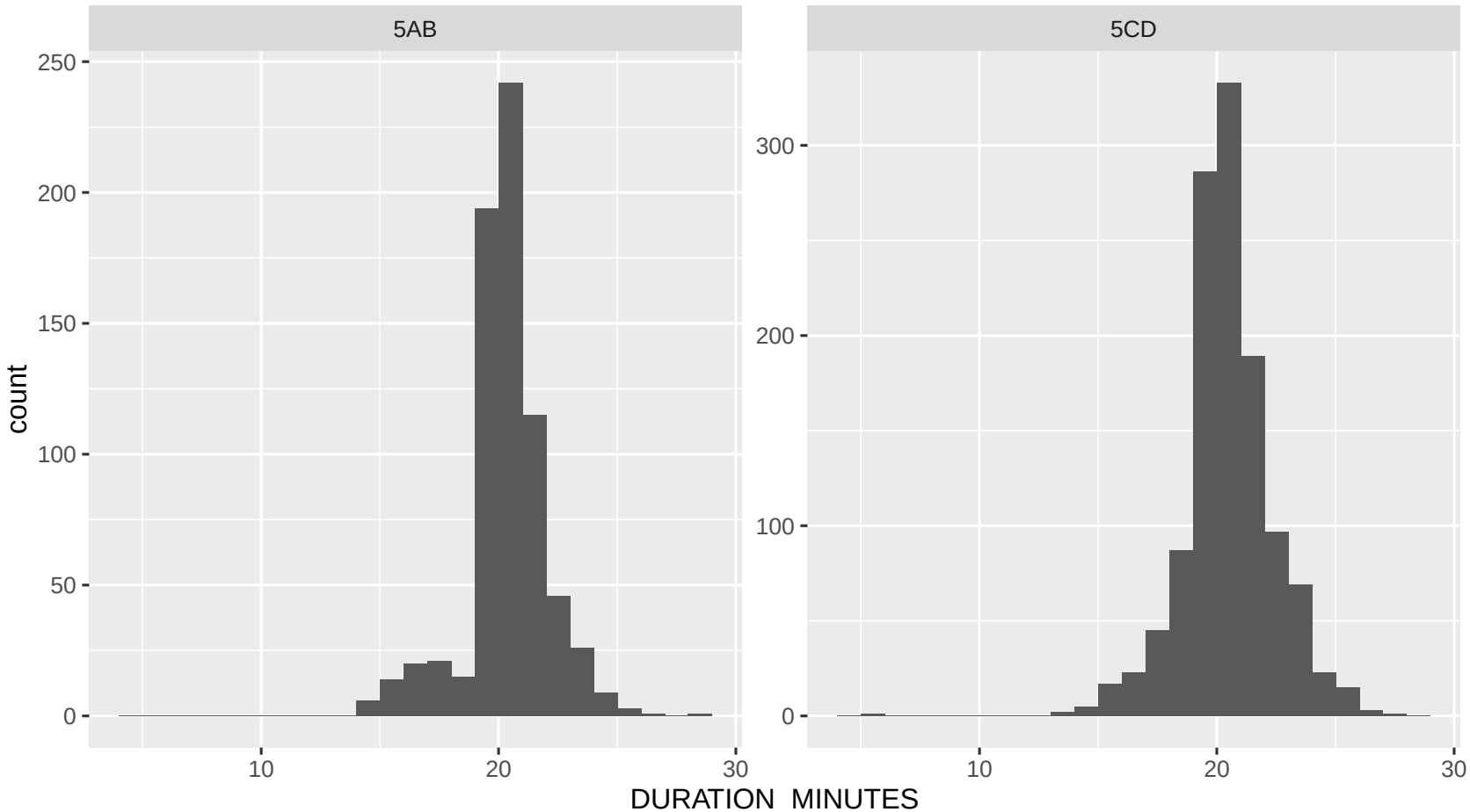
facet_wrap: free scales

```
fw1_free <- ggplot(data=df_data, aes(x=DURATION_MINUTES)) +
  geom_histogram(binwidth=1) + facet_wrap(~ AREA , scales = 'free')
print(fw1_free)
```



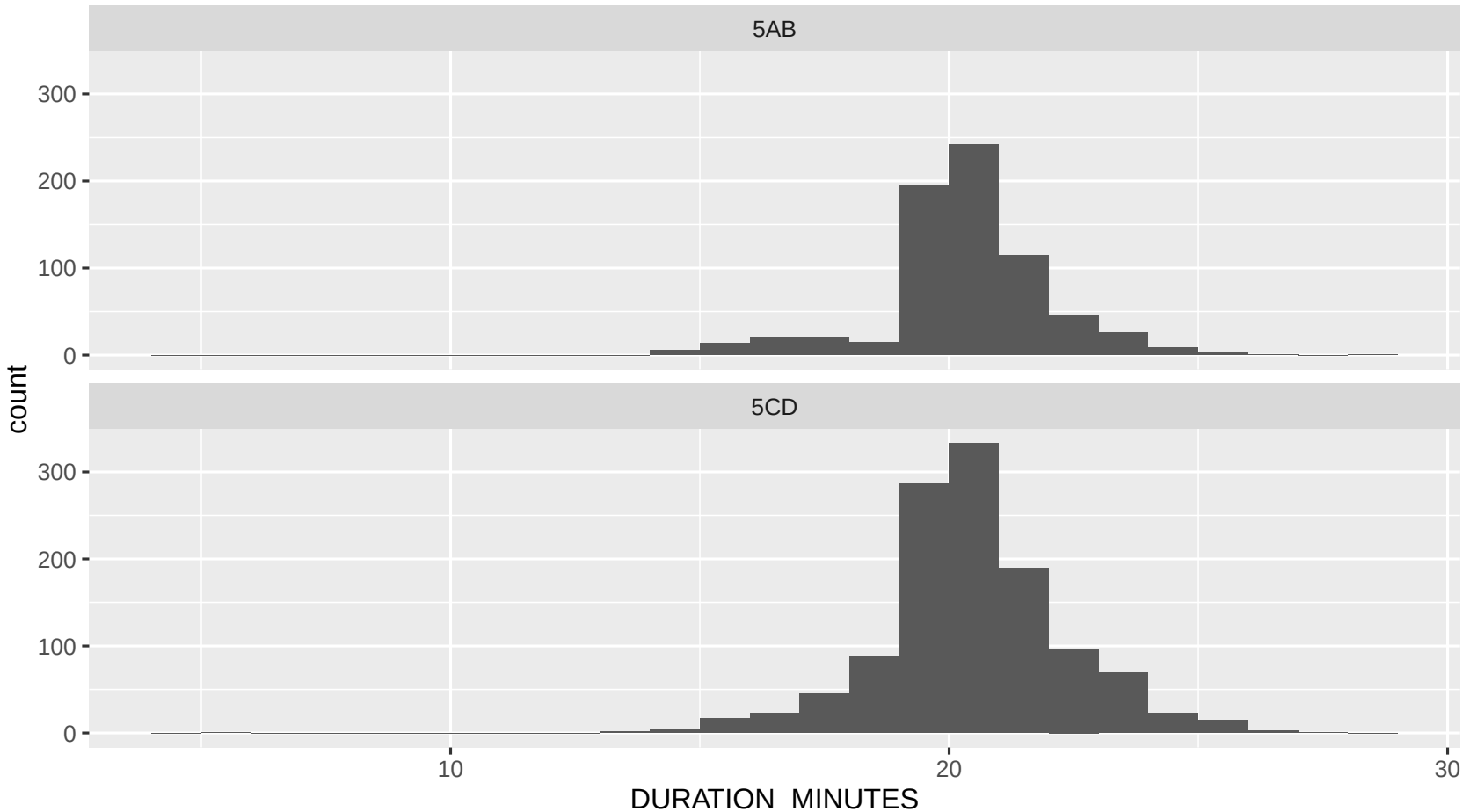
facet_wrap: free y scale

```
fw1_free_y <- ggplot(data=df_data, aes(x=DURATION_MINUTES)) +
  geom_histogram(binwidth=1) + facet_wrap(~ AREA , scales = 'free_y')
print(fw1_free_y)
```



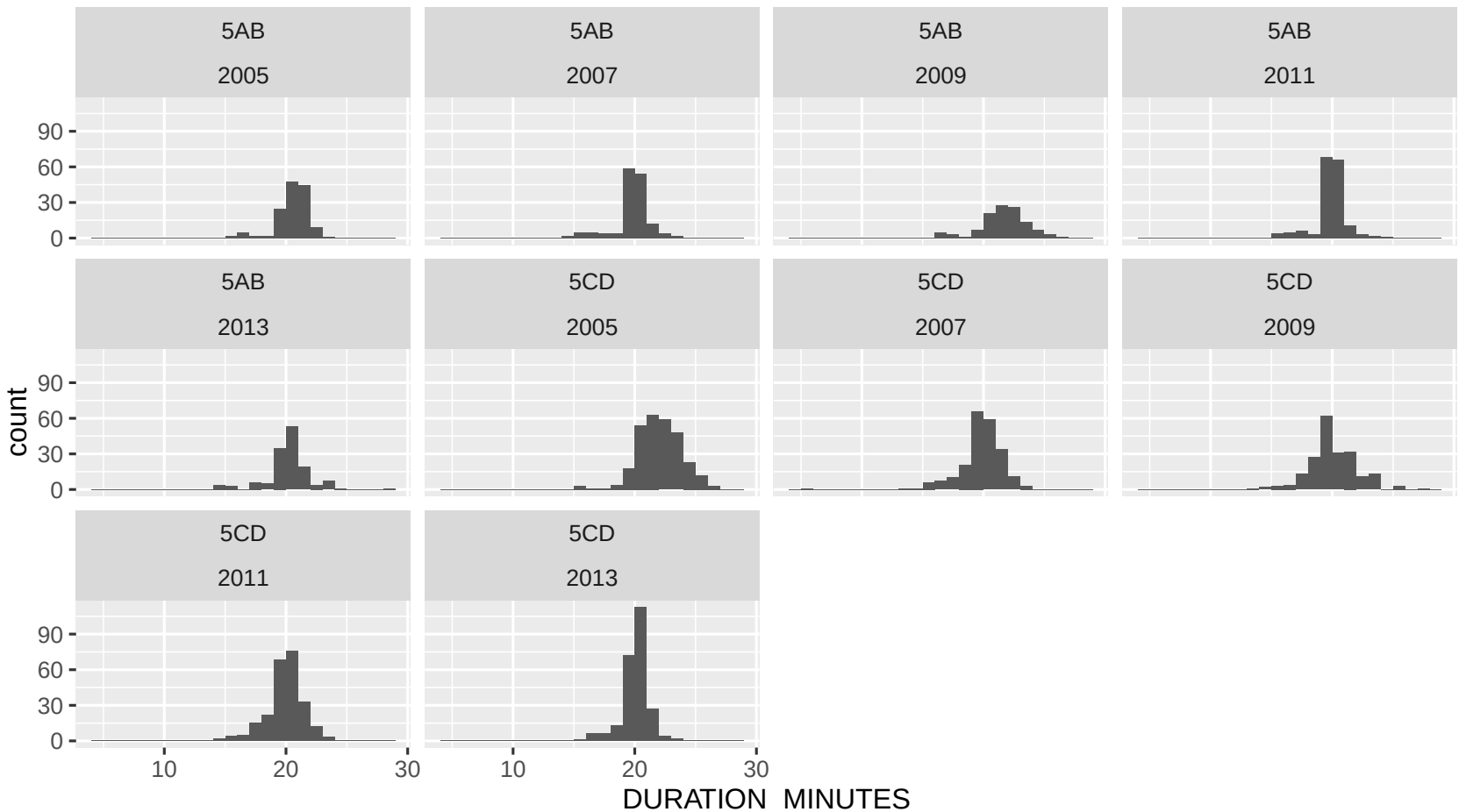
facet_wrap: free y scale

```
fw1_col <- ggplot(data=df_data, aes(x=DURATION_MINUTES)) +
  geom_histogram(binwidth=1) + facet_wrap(~ AREA, ncol = 1, nrow = 2, scale
print(fw1_col)
```



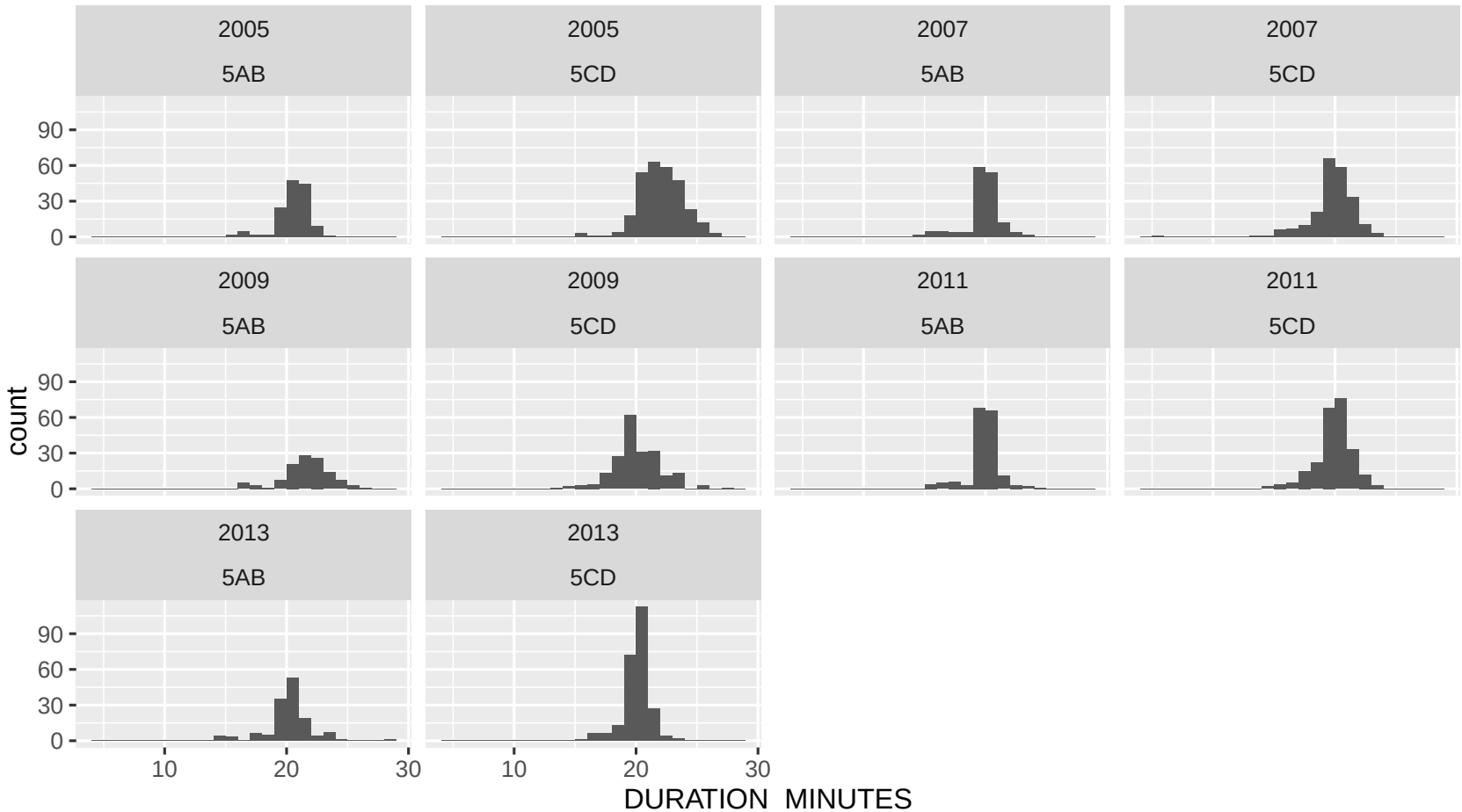
facet_wrap: free y scale

```
fw2 <- ggplot(data=df_data, aes(x=DURATION_MINUTES)) +
  geom_histogram(binwidth=1) + facet_wrap(~ AREA + Year_fac)
print(fw2)
```



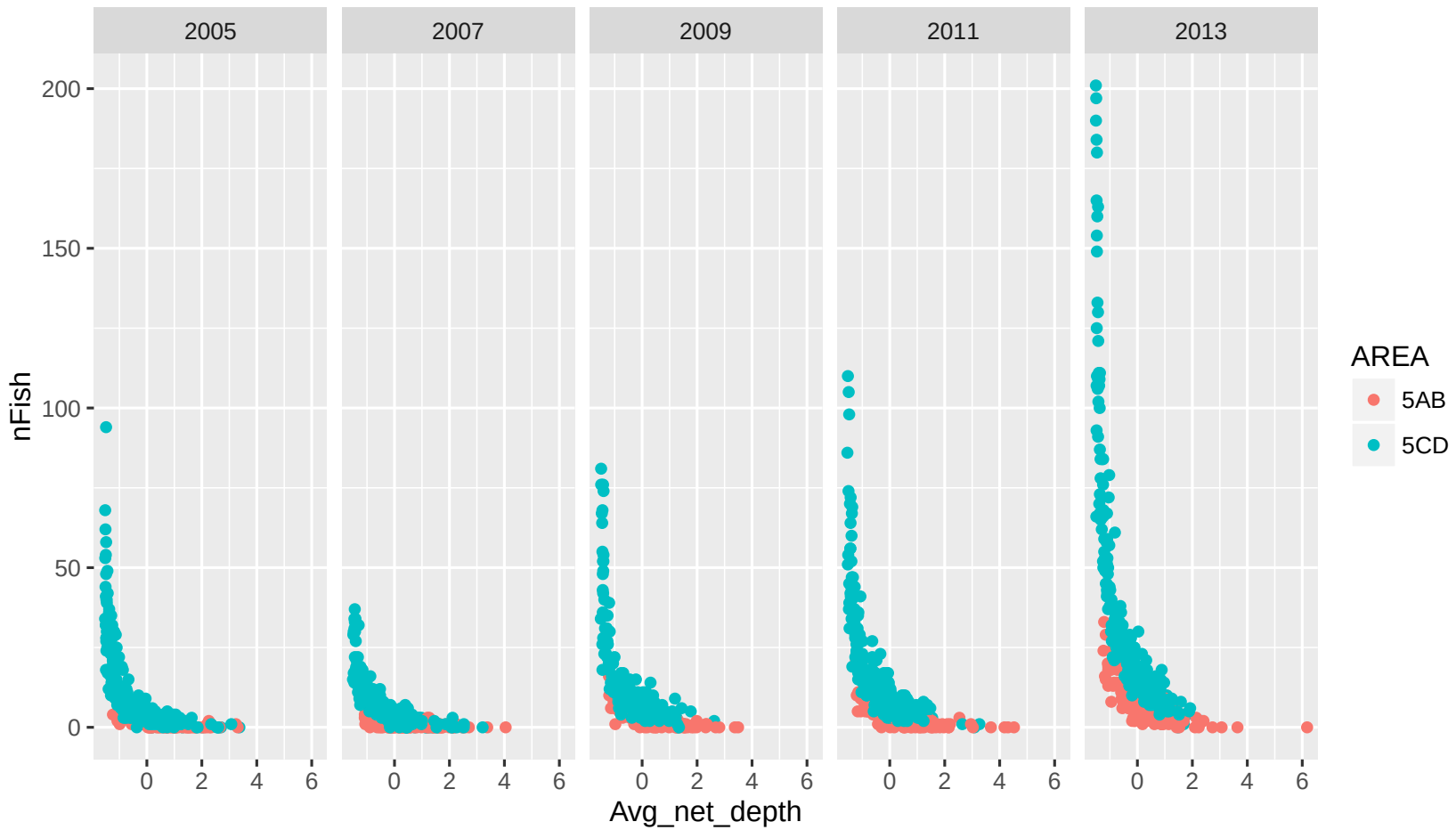
facet_wrap: free y scale

```
fw2_2 <- ggplot(data=df_data, aes(x=DURATION_MINUTES)) +
  geom_histogram(binwidth=1) + facet_wrap( ~ Year_fac + AREA)
print(fw2_2)
```



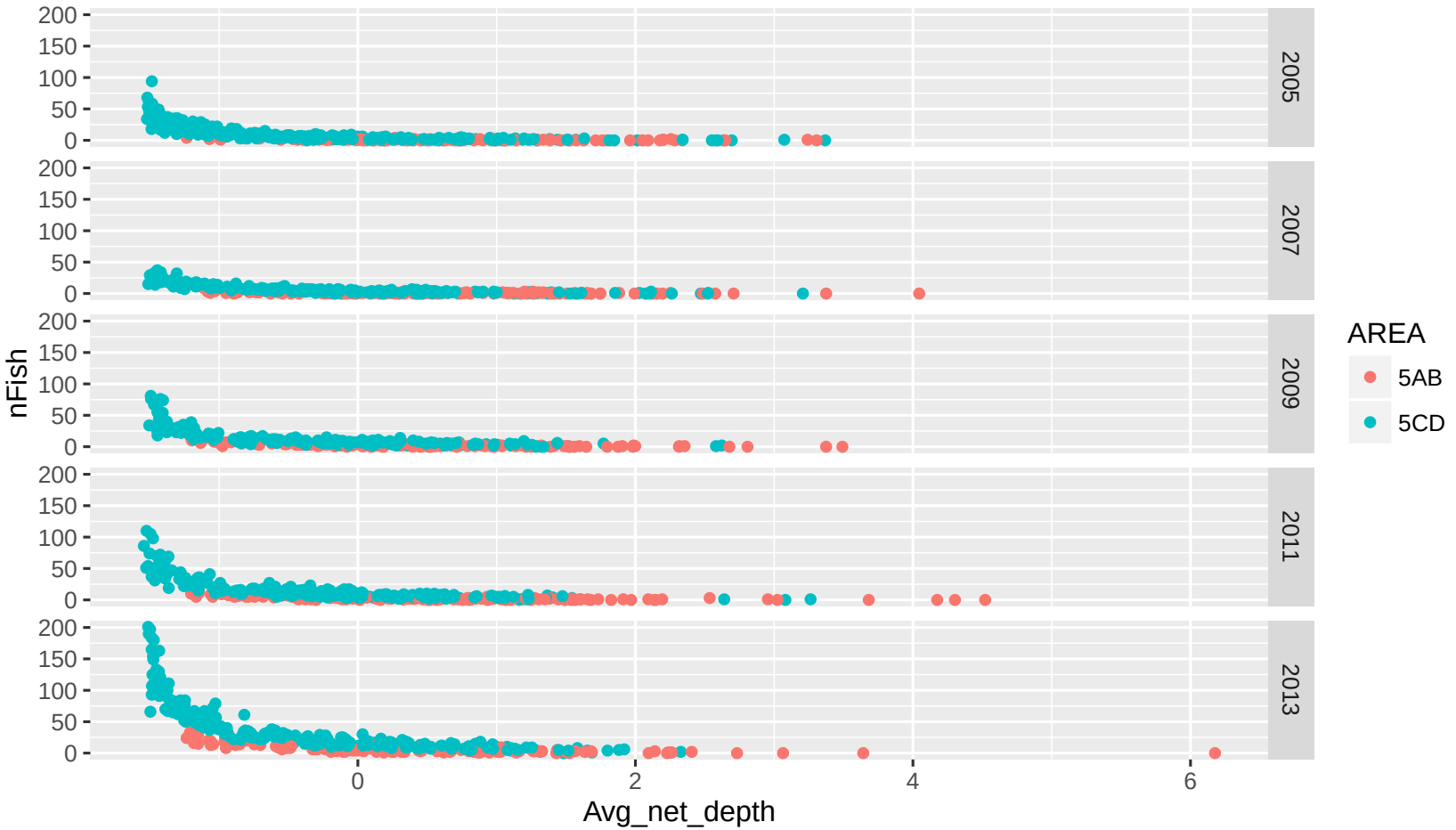
facet_grid : more flexible

```
fg1_1 <- ggplot(data=df_data, aes(x=Avg_net_depth, y=nFish, color=AREA)) +
  geom_point() + facet_grid(. ~ Year)
print(fg1_1)
```



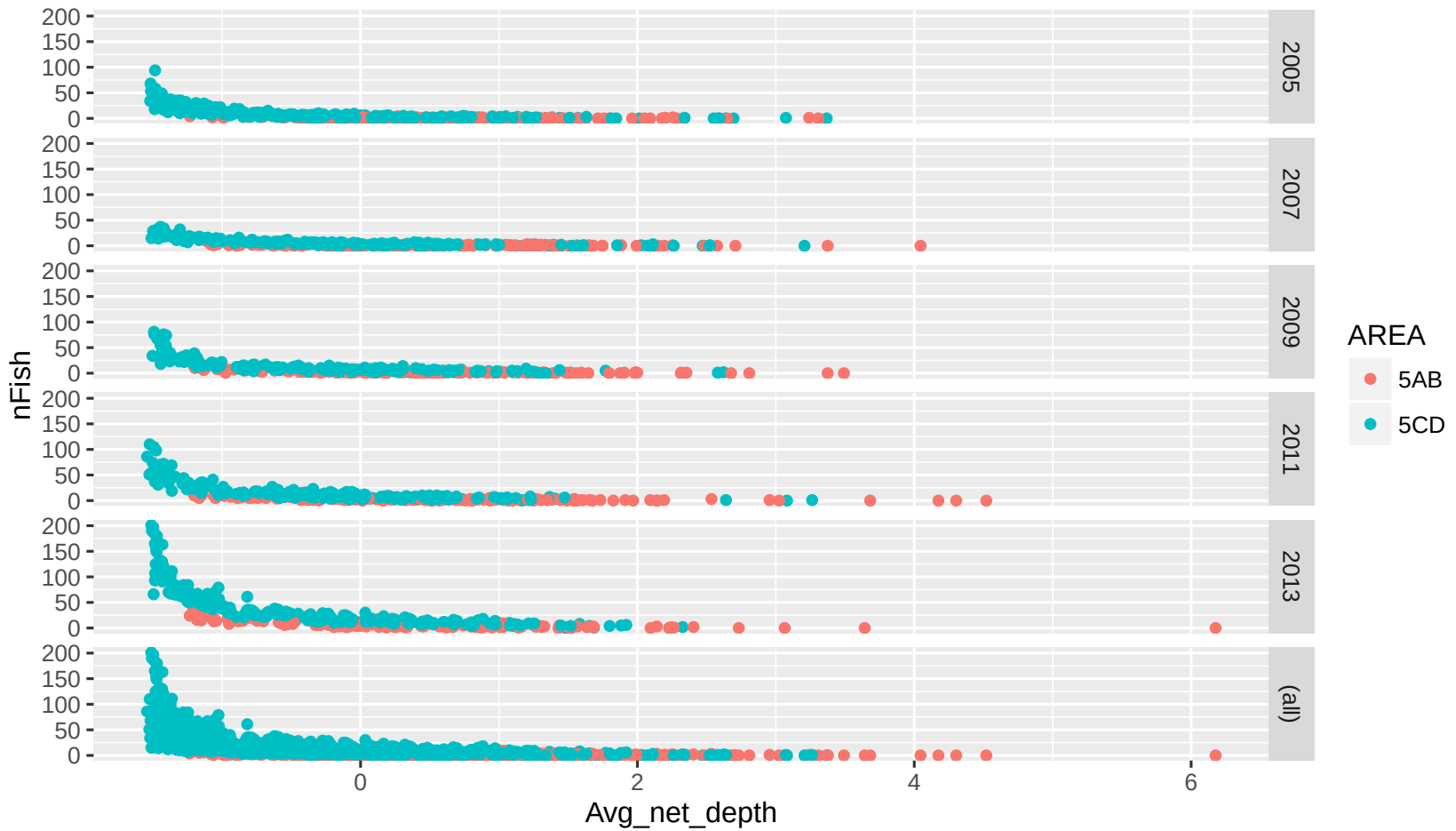
facet_grid : change faceting display

```
fg1_2 <- ggplot(data=df_data, aes(x=Avg_net_depth, y=nFish, color=AREA)) +
  geom_point() + facet_grid(Year ~ .)
print(fg1_2)
```



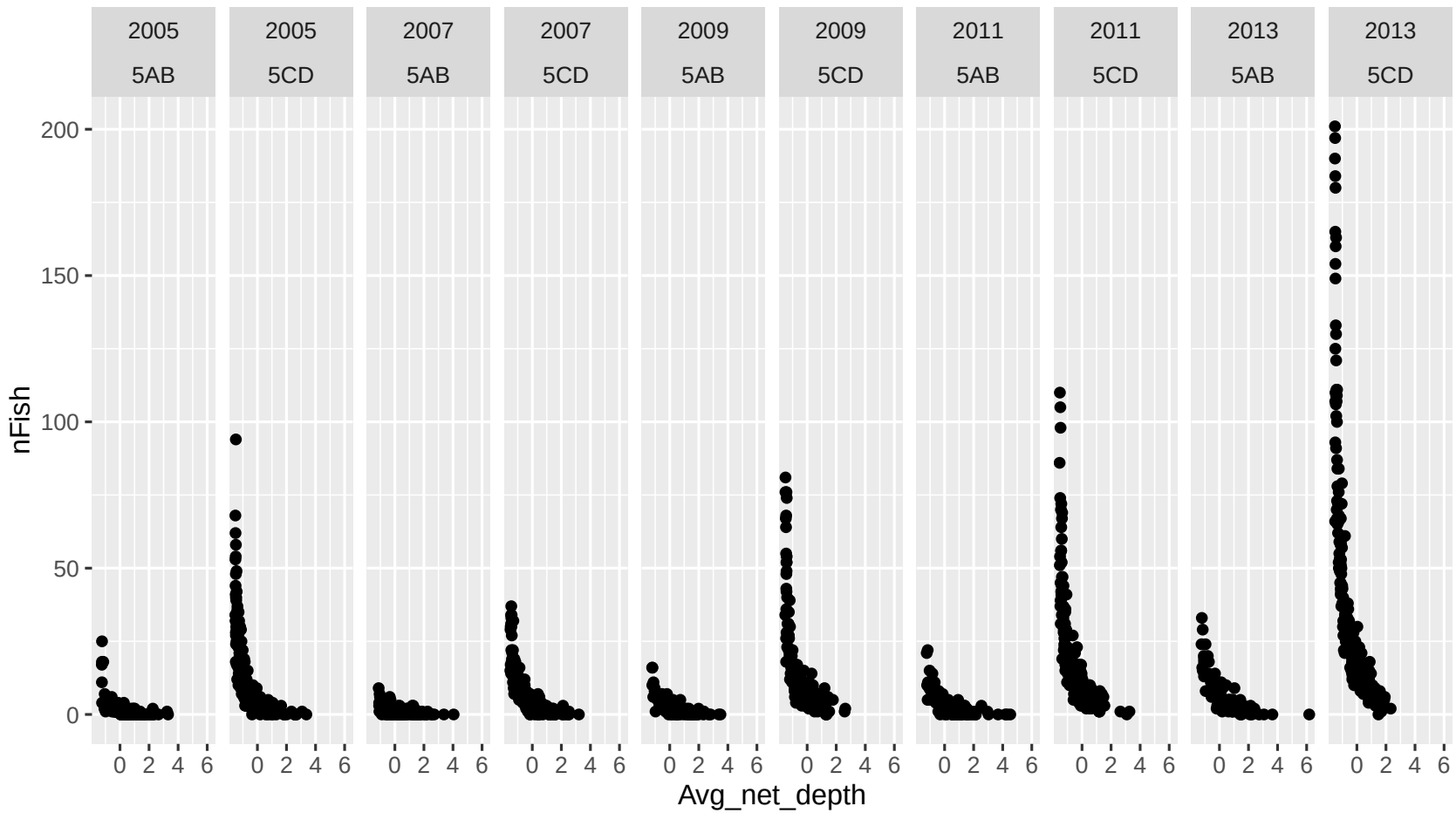
facet_grid : facet with all the data

```
fg1_3 <- ggplot(data=df_data, aes(x=Avg_net_depth, y=nFish, color=AREA)) +
  geom_point() + facet_grid(Year ~ ., margins = TRUE)
print(fg1_3)
```



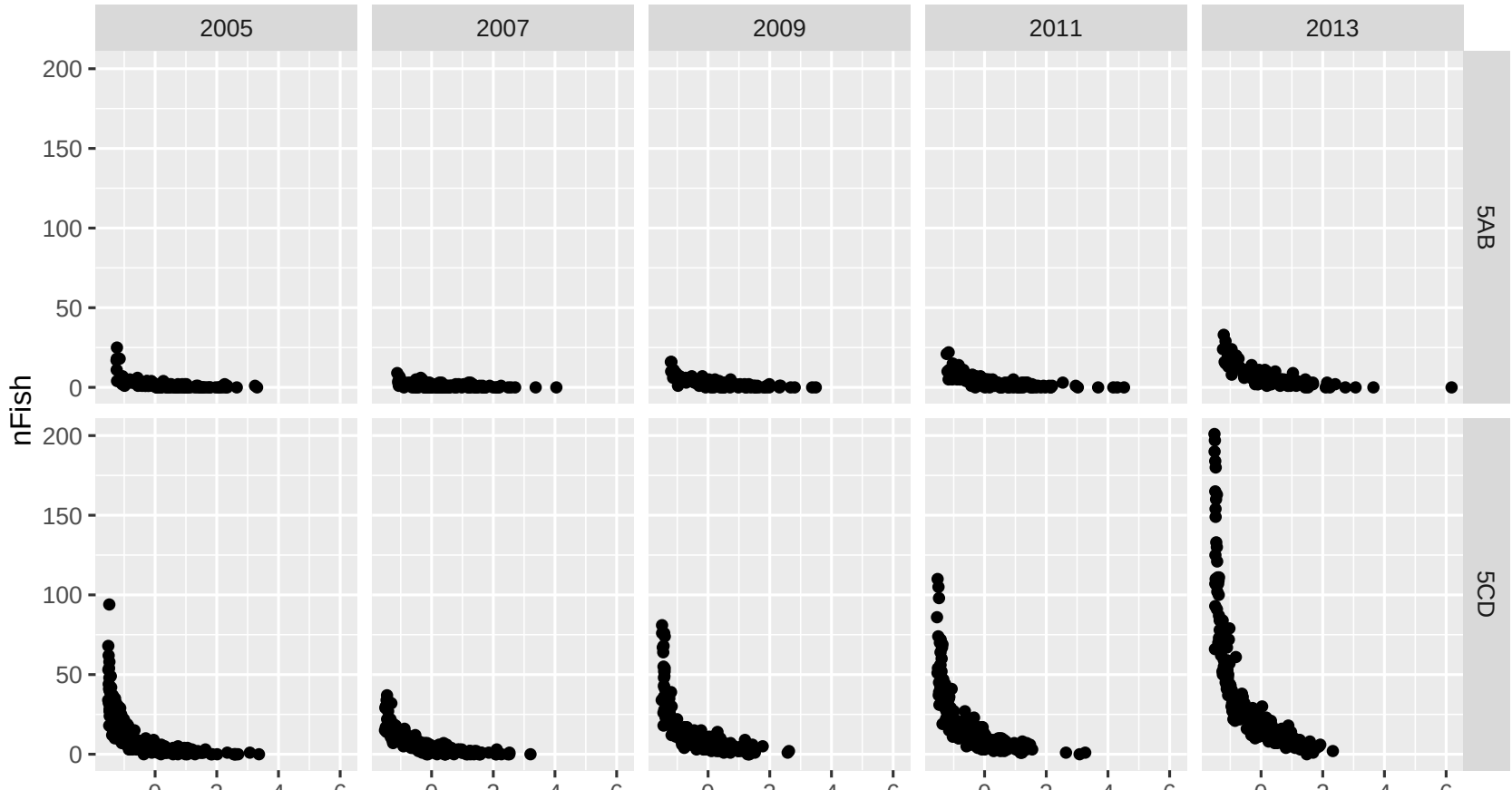
facet_grid : two faceting factors

```
fg2_1 <- ggplot(data=df_data, aes(x=Avg_net_depth, y=nFish)) +
  geom_point() + facet_grid(~ Year + AREA)
print(fg2_1)
```



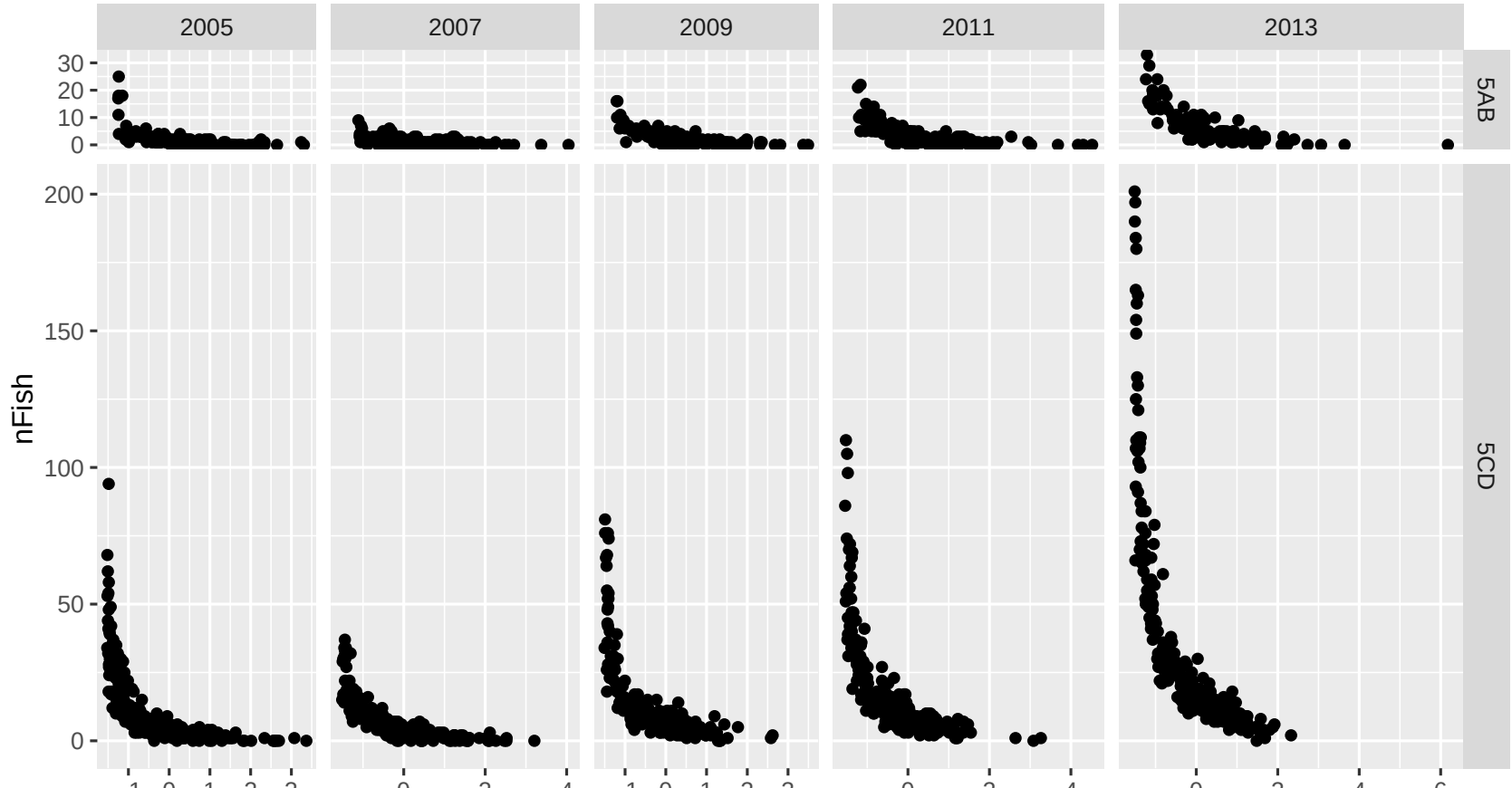
facet_grid : two faceting factors

```
fg2_2 <- ggplot(data=df_data, aes(x=Avg_net_depth, y=nFish)) +
  geom_point(binwidth=1) + facet_grid(AREA ~ Year)

## Error:  Unknown parameters:  binwidth

print(fg2_2)
```



facet_grid : scales and space free

```
fg2_3 <- ggplot(data=df_data, aes(x=Avg_net_depth, y=nFish)) +
  geom_point(binwidth=1) + facet_grid(AREA ~ Year, scales='free', space = '

## Error:  Unknown parameters:  binwidth

print(fg2_3)
```



facet_grid : renaming labels

Replace manually names factor:

```
levels(df_data$Year) <- letters[1:nlevels(df_data$Year)]
```

Or write a function:

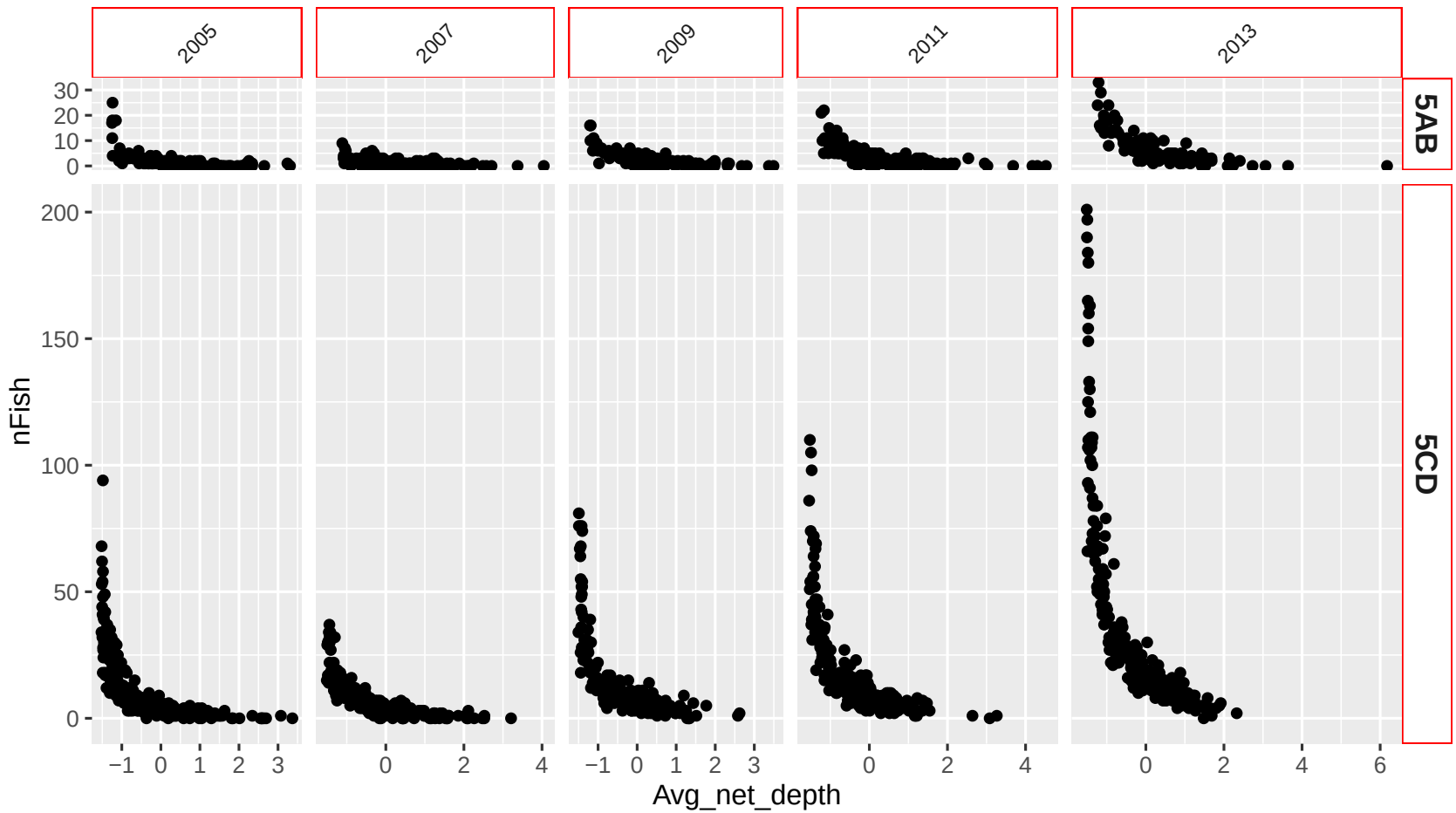
```
## string is the levels of a factor
fn_alphabetic_label <- function(string){
  for ( i in 1:length(string)){
    string[i] <- letters[i]
  }
  return(string)
}
```

facet_grid : renaming labels

```
fg1_3a <- ggplot(data=df_data, aes(x=Avg_net_depth, y=nFish, color=AREA)) +
  geom_point(binwidth=1) +
  facet_grid(Year ~ ., labeller = labeller(Year = fn_alphabetic_label))

## Error:  Unknown parameters:  binwidth

print(fg1_3a)
```

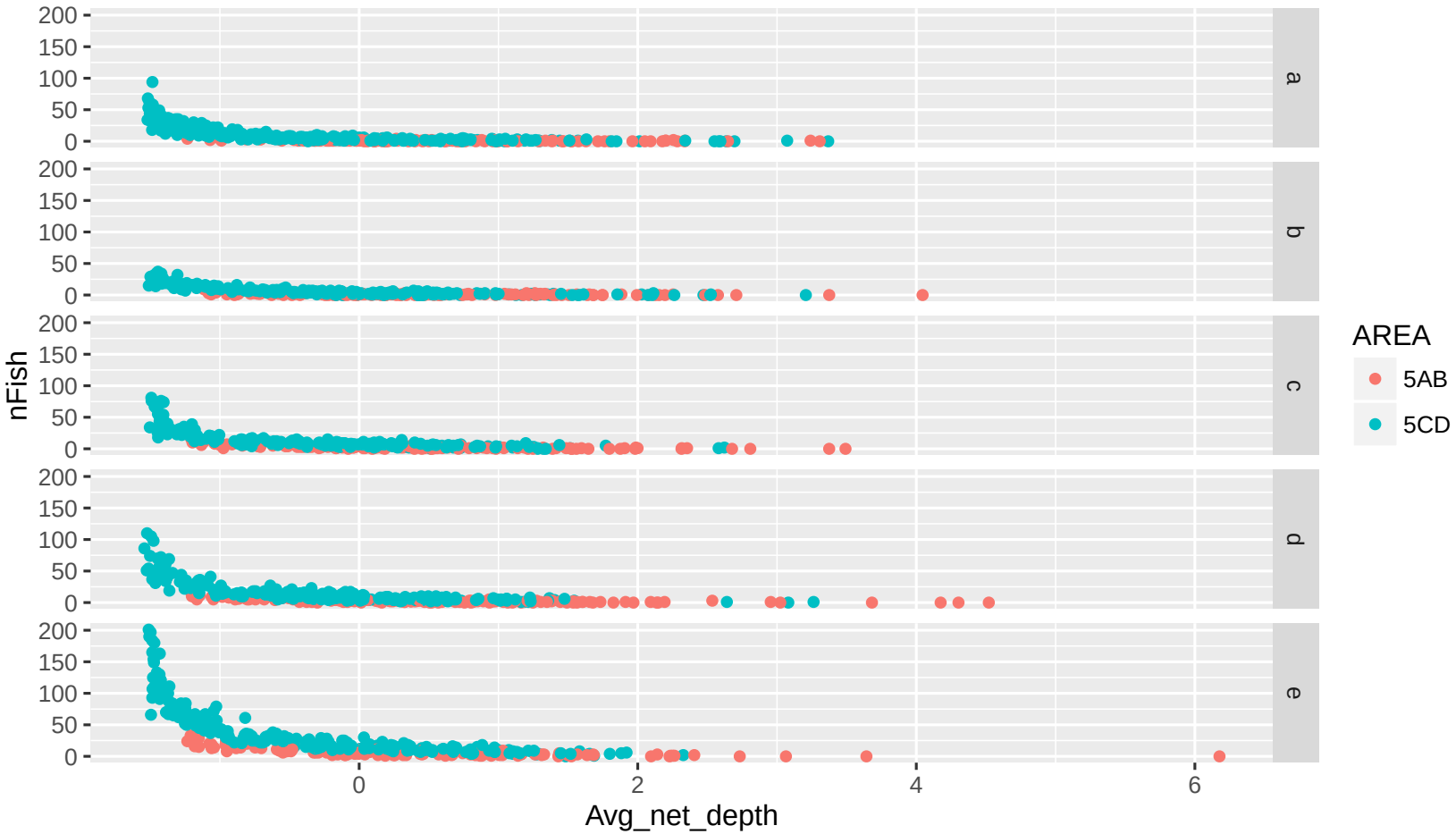
facet_grid : changing facets

```
fg2_3b <- fg2_3 +
    theme(strip.text.x = element_text(size=8, angle=45),
          strip.text.y = element_text(size=12, face="bold"),
          strip.background = element_rect(colour="red", fill=NA))
print(fg2_3b)
```



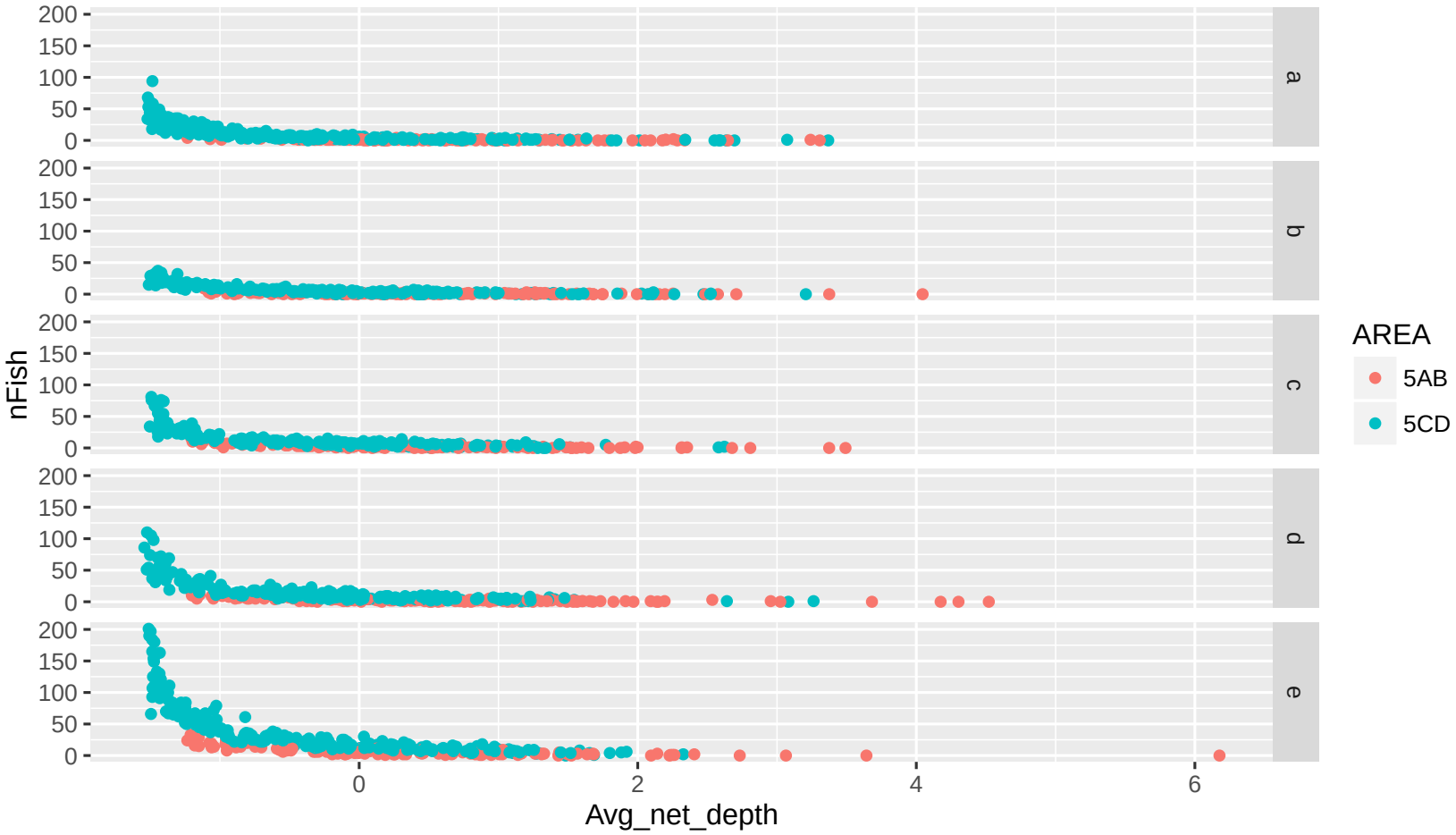
Theme

```
print(fg1_3a)
```



facet_grid : changing facets

```
print(fg1_3a)
```



Useful R packages which use ggplot2

- ggfortify and its autoplot() function allows plotting some popular R packages using a standardized approach.
  Diagnostic plots with Generalized Linear Models (GLM), Plotting Principal Component Analysis ...
- MCMC plots: ggmcmc
  install.packages("ggmcmc", dependencies=TRUE)
- Correlation plots: GGally
  install.packages("GGally", dependencies=TRUE))
- Latex expression in plot: latex2exp
  install.packages("latex2exp", dependencies=TRUE))

Useful R packages which use ggplot2

Thank you for your attention.

Questions ?

Code available at:
https://github.com/JBLecomte/ggplot2-Introduction.git