all_graphs.R

mwilmes

Mon Nov 6 09:11:54 2017

```r
# This file contains all commands of the Chapter "Data Visualisation"

# load the library
library(tidyverse)
# read the data
popsize <- read_tsv("../data/FauchaldEtAl2017/pop_size.csv")
```

```
## Parsed with column specification:
## cols(
##   Herd = col_character(),
##   Year = col_integer(),
##   Pop_Size = col_integer()
## )
```

```r
ndvi <- read_tsv("../data/FauchaldEtAl2017/ndvi.csv")
```

```
## Parsed with column specification:
## cols(
##   Herd = col_character(),
##   Year = col_integer(),
##   NDVI_May = col_double(),
##   NDVI_June_August = col_double()
## )
```

```r
seaice <- read_tsv("../data/FauchaldEtAl2017/sea_ice.csv")
```

```
## Parsed with column specification:
## cols(
##   Herd = col_character(),
##   Year = col_integer(),
##   Jan = col_double(),
##   Feb = col_double(),
##   Mar = col_double(),
##   Apr = col_double(),
##   May = col_double(),
##   Jun = col_double(),
##   Jul = col_double(),
##   Aug = col_double(),
##   Sep = col_double(),
##   Oct = col_double(),
##   Nov = col_double(),
##   Dec = col_double()
## )
```

```r
snow <- read_tsv("../data/FauchaldEtAl2017/snow.csv")
```

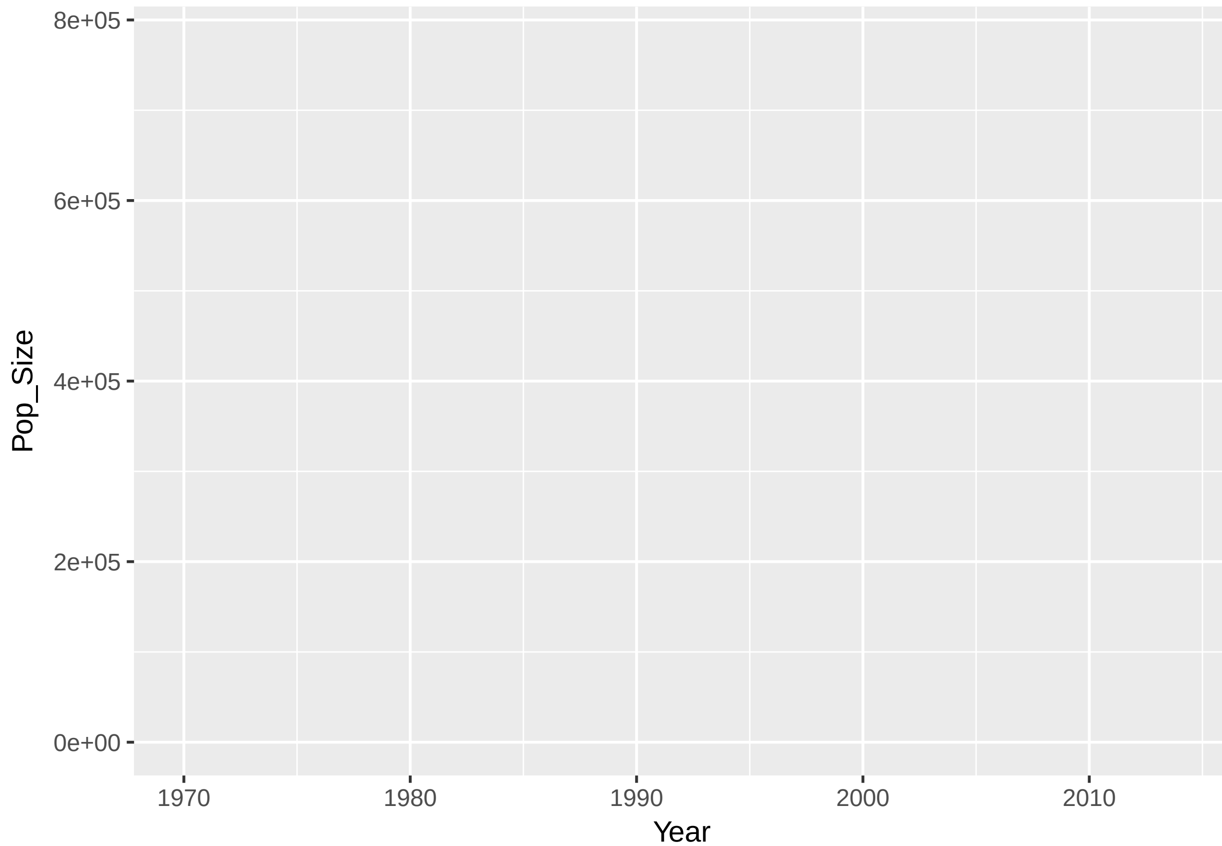
```
## Parsed with column specification:
## cols(
##   Year = col_integer(),
##   Herd = col_character(),
```

```
##   Perc_snowcover = col_double(),
##   Week_snowmelt = col_integer()
## )
```

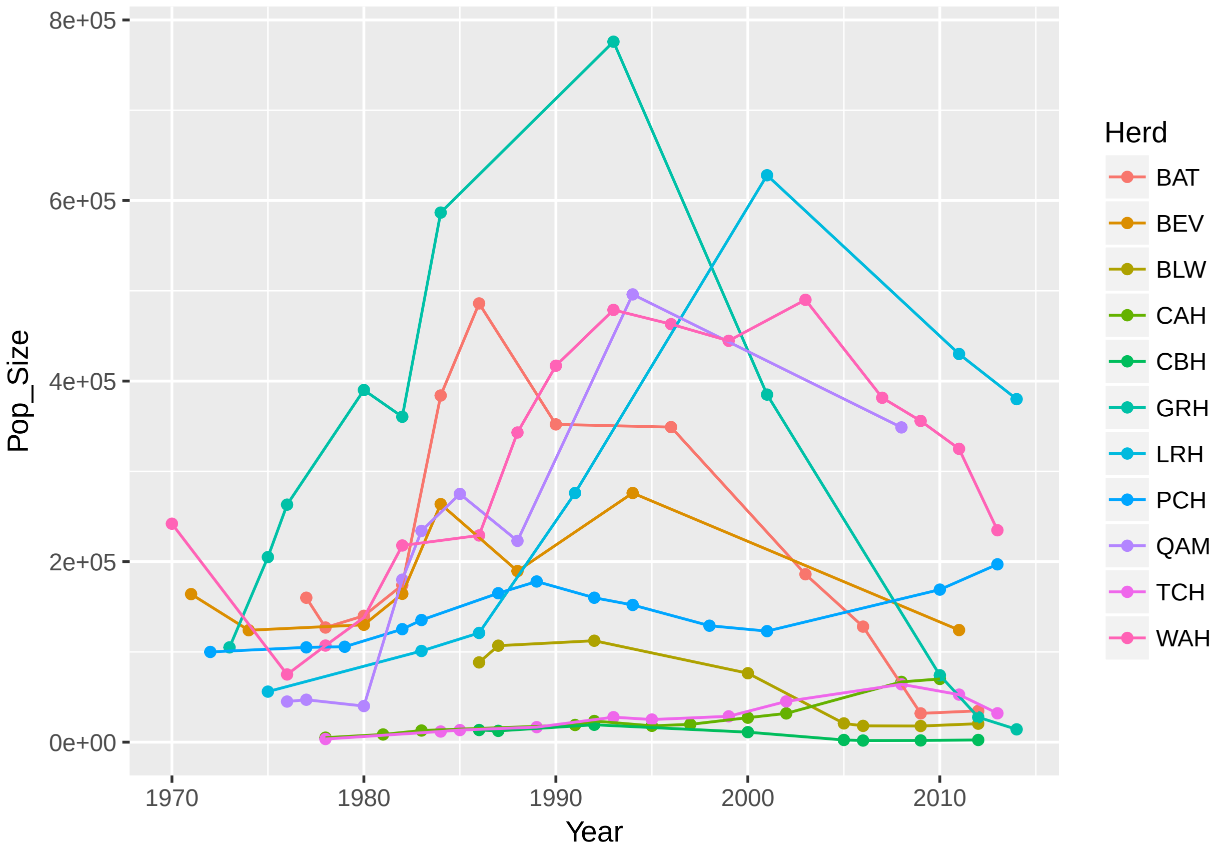
```r
# bring data into long format
seaice <- seaice %>% gather(Month, Cover, 3:14)





# build the first plot
ggplot(data = popsize)
```



```r
# add an aestethic mapping
ggplot(data = popsize) + aes(x = Year, y = Pop_Size, colour = Herd)
```

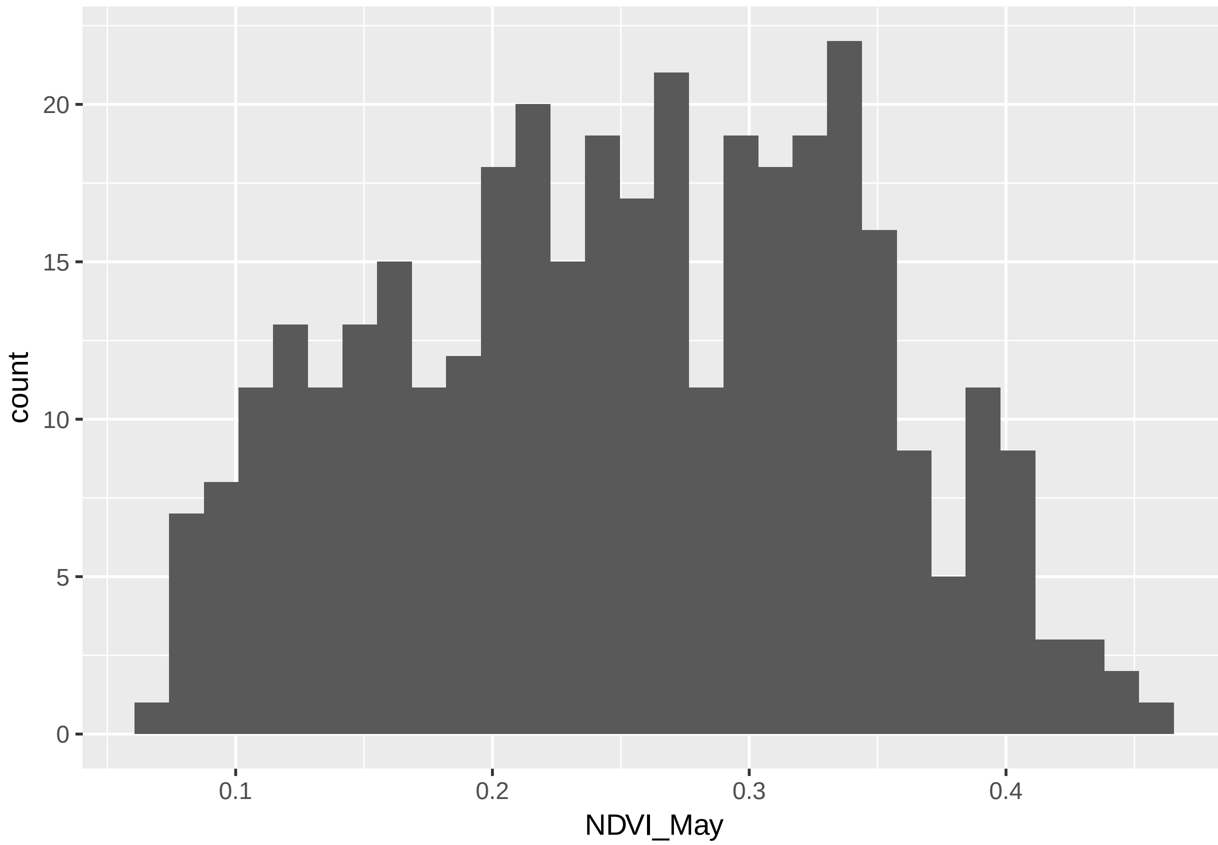



```r
# add geometries
ggplot(data = popsize) +
  aes(x = Year, y = Pop_Size, colour = Herd) +
  geom_point() +
  geom_line()
```



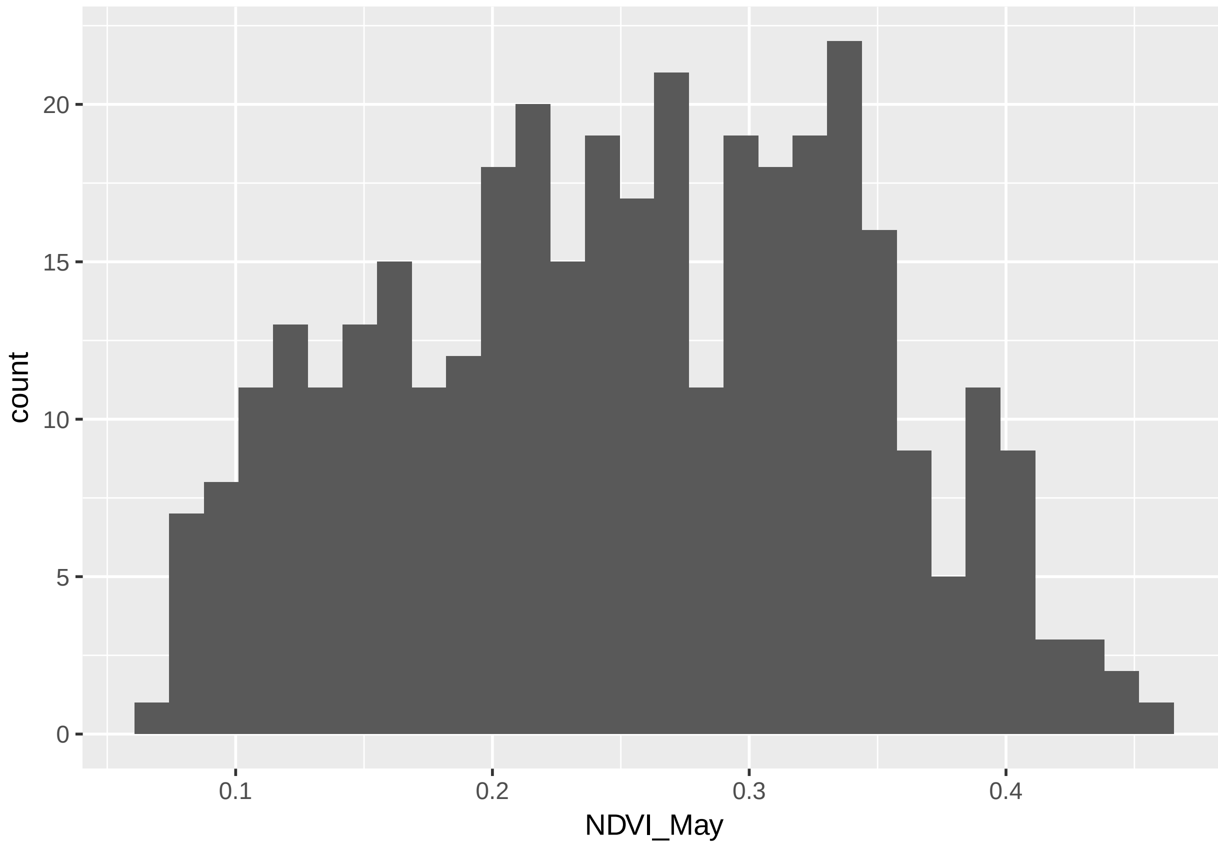

```
# plot frequency distribution (histogram)
ggplot(data = ndvi) + geom_histogram(aes(x = NDVI_May))
```

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



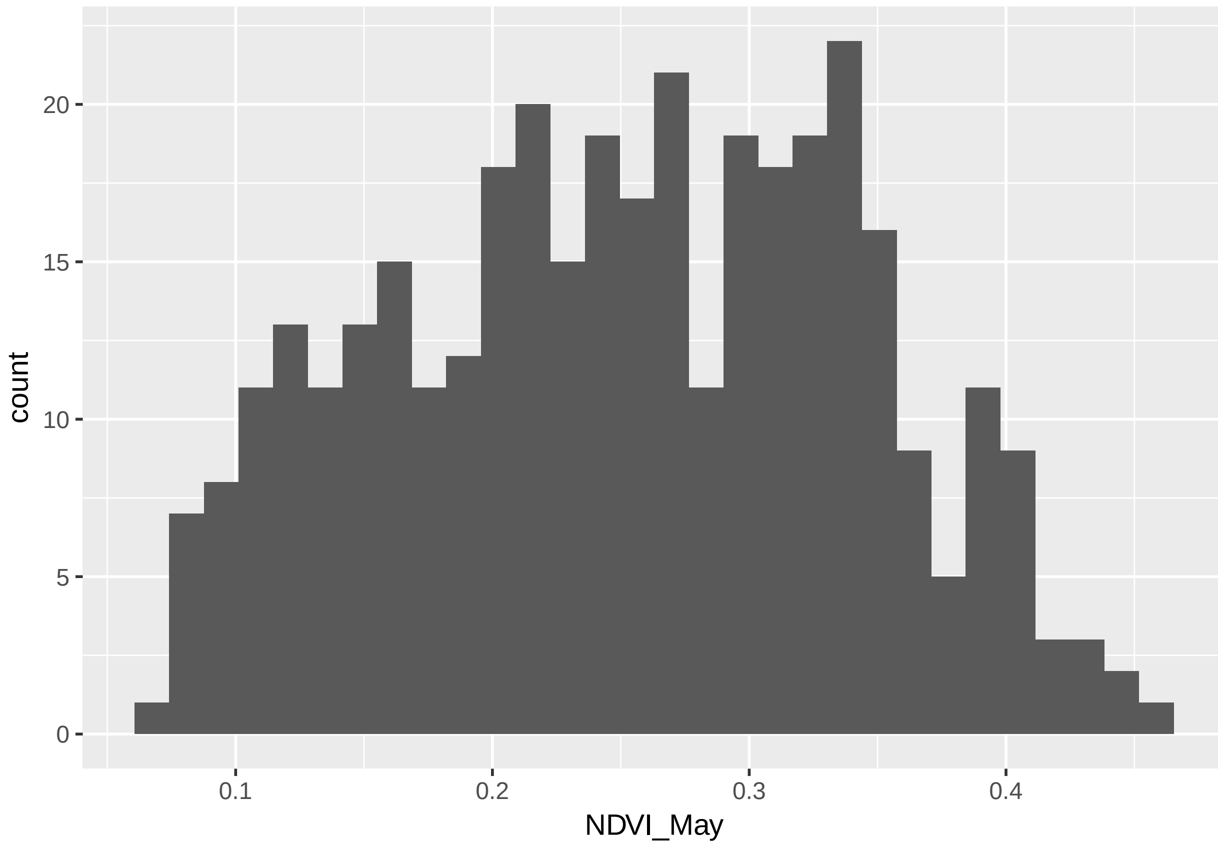

```r
# "unrolled" or nested
# These three commands produce the same graph:
ggplot(data = ndvi) + aes(x = NDVI_May) + geom_histogram()
```

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



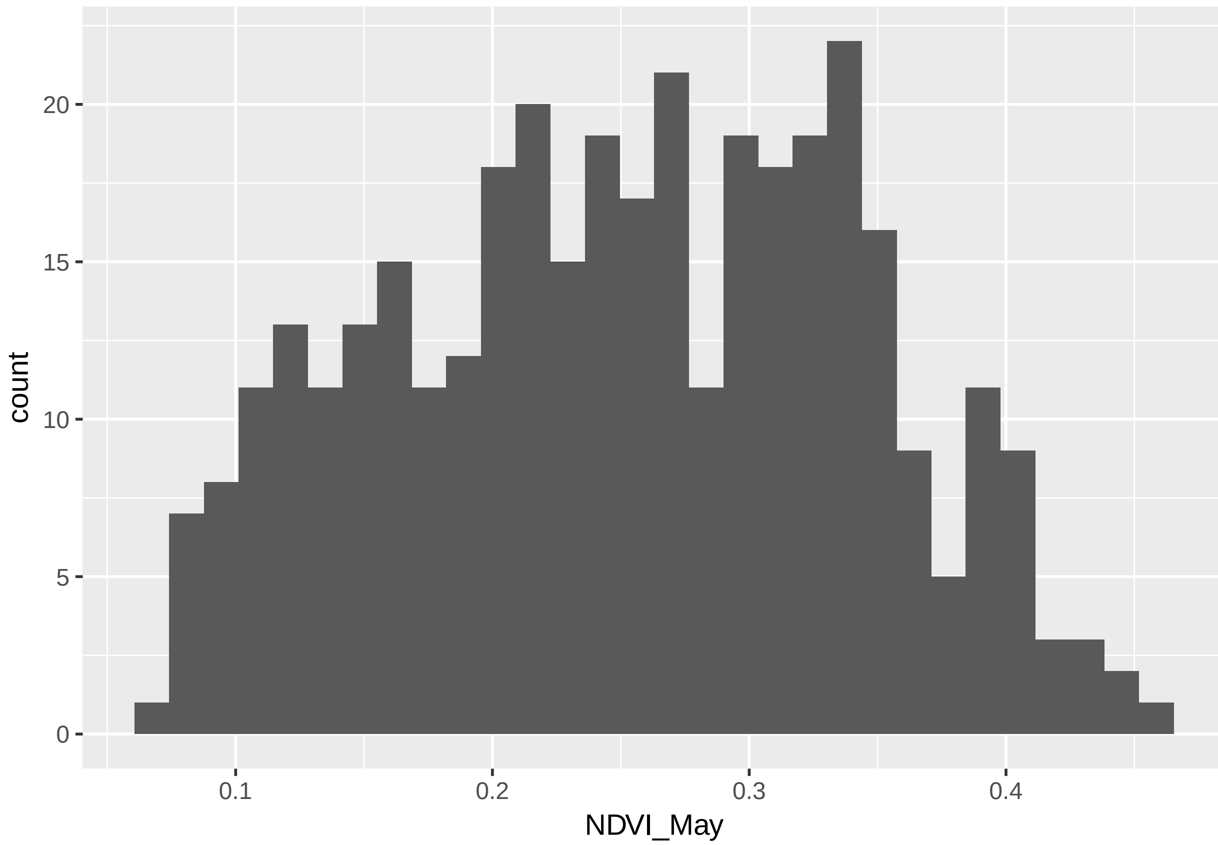

```r
ggplot(data = ndvi, aes(x = NDVI_May)) + geom_histogram()
```

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

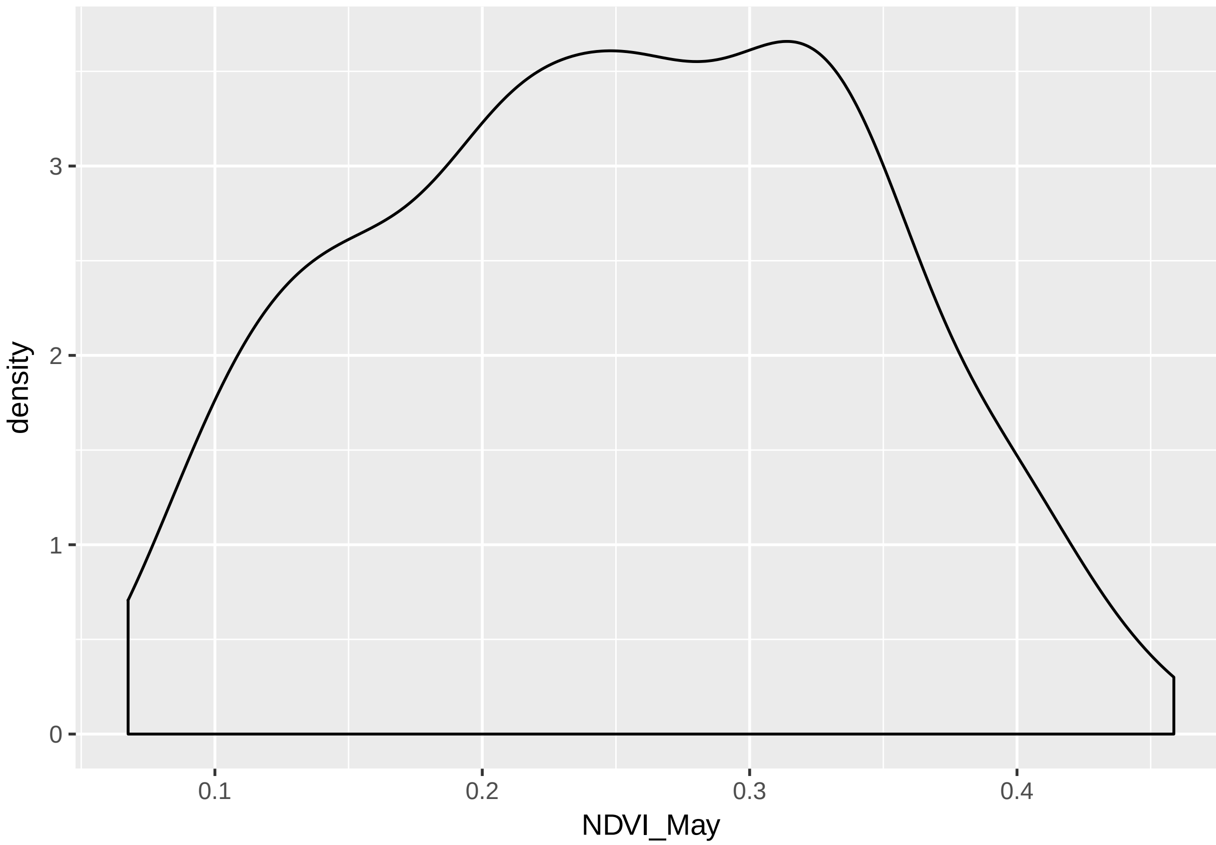



```r
ggplot(data = ndvi) + geom_histogram(aes(x = NDVI_May))
```

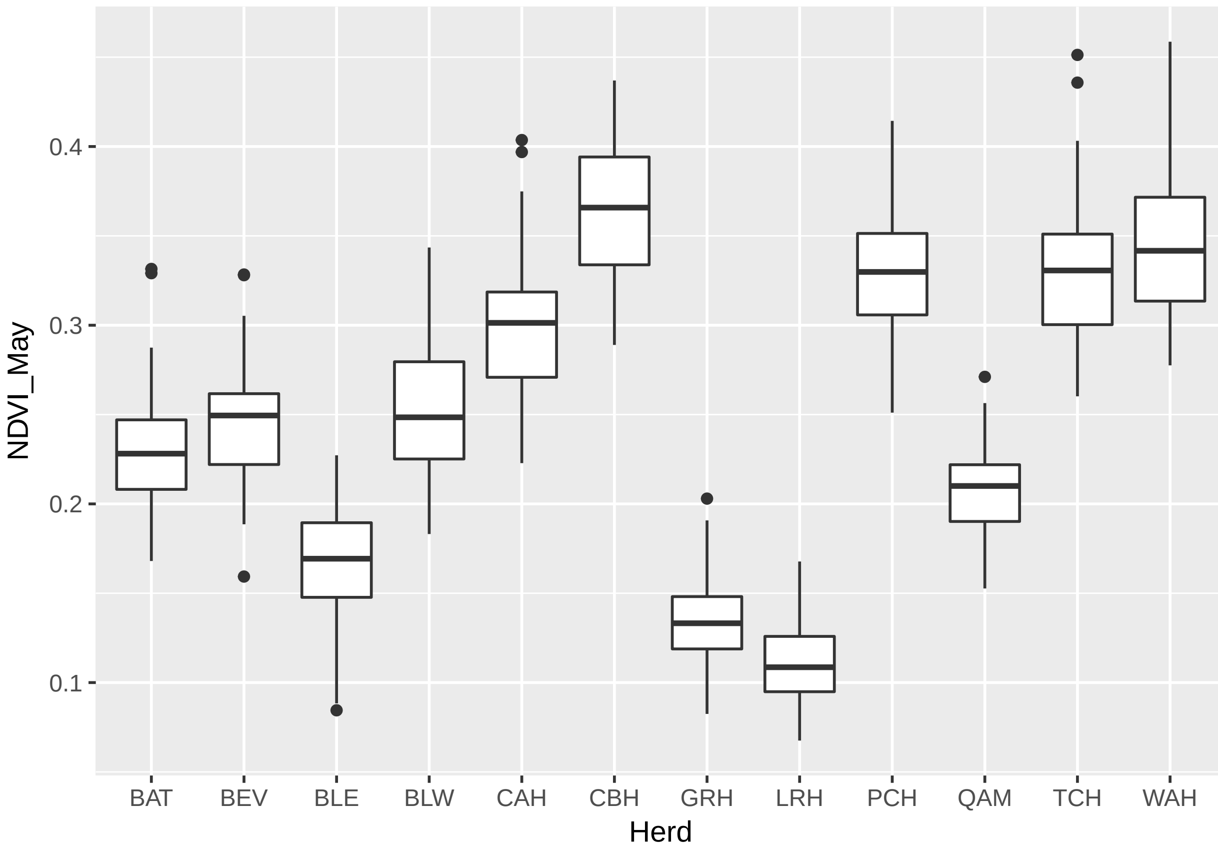
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



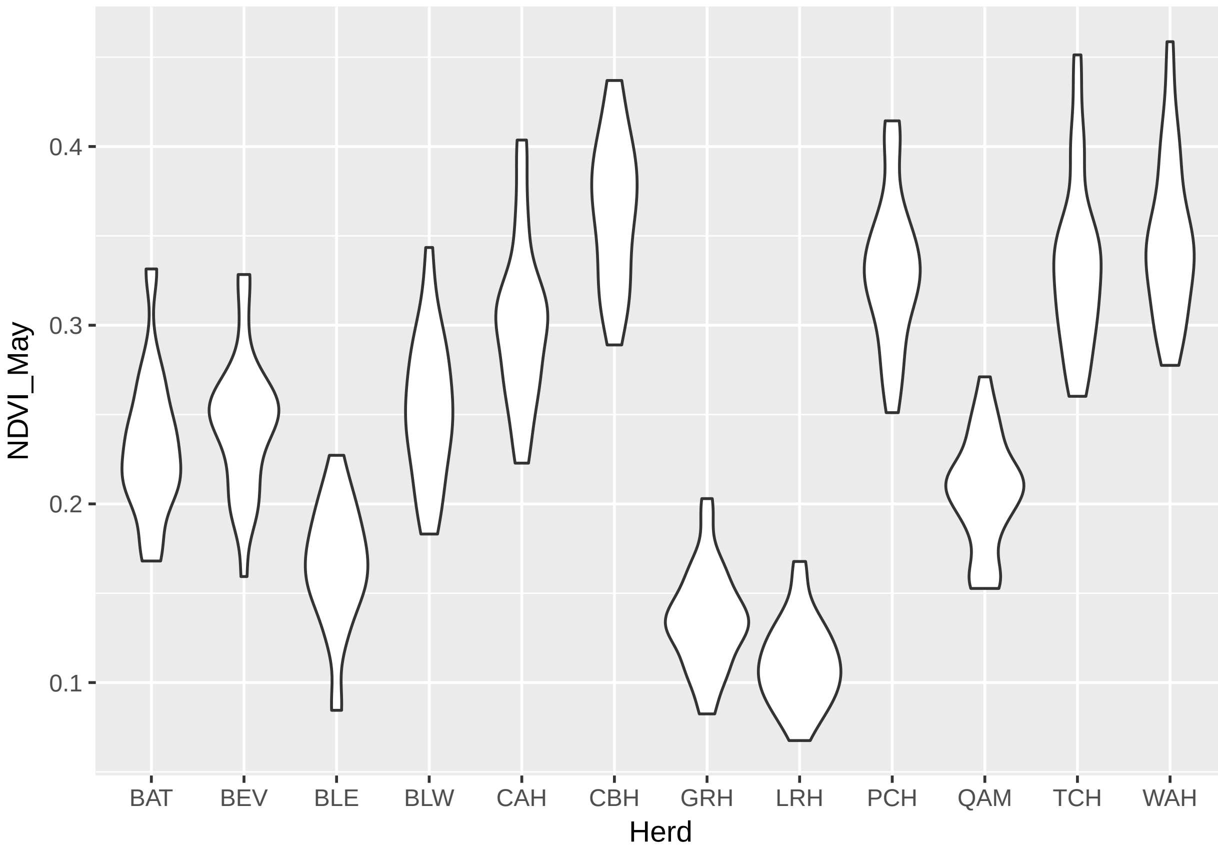

```r
# density plot
ggplot(data = ndvi) + aes(x = NDVI_May) + geom_density()
```



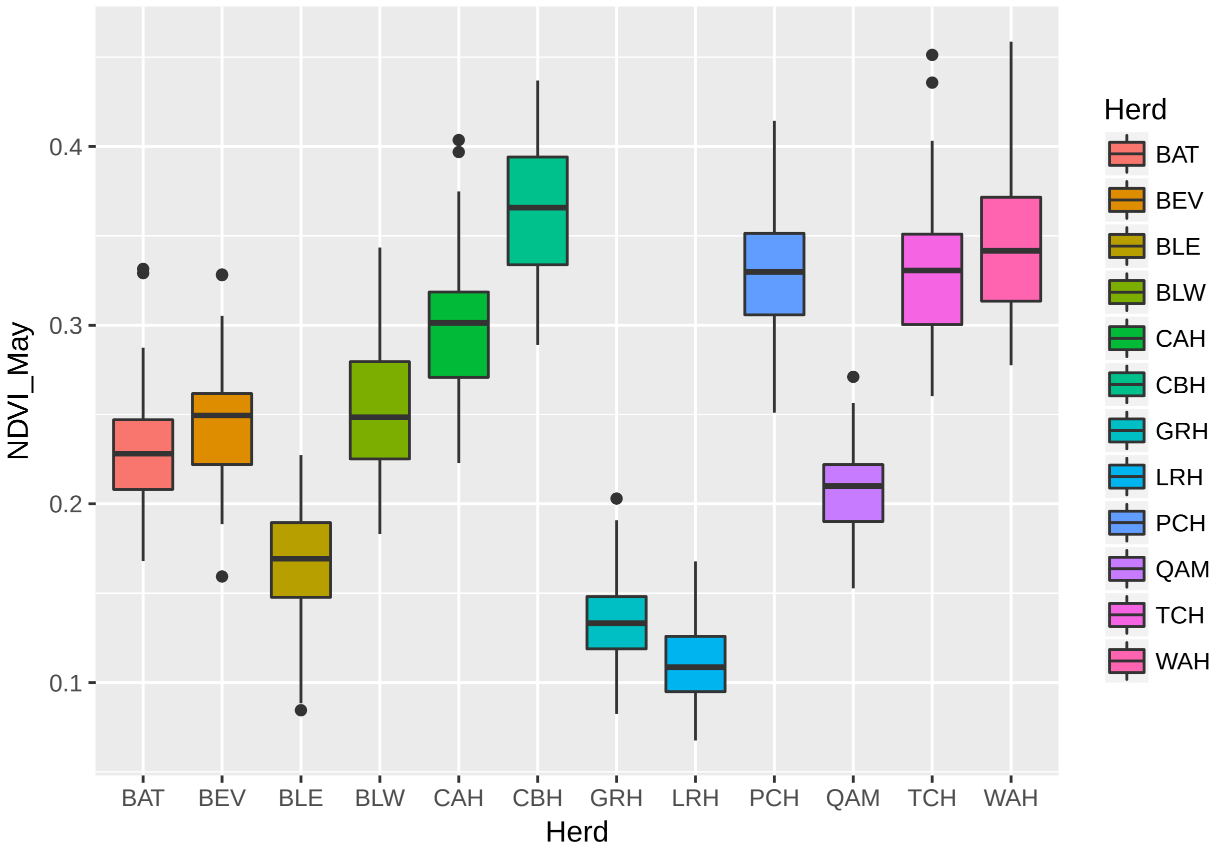

```
# assign plot to a variable
pl <- ggplot(data = ndvi) + aes(x = Herd, y = NDVI_May)
# add components to existing plot
pl + geom_boxplot()
```



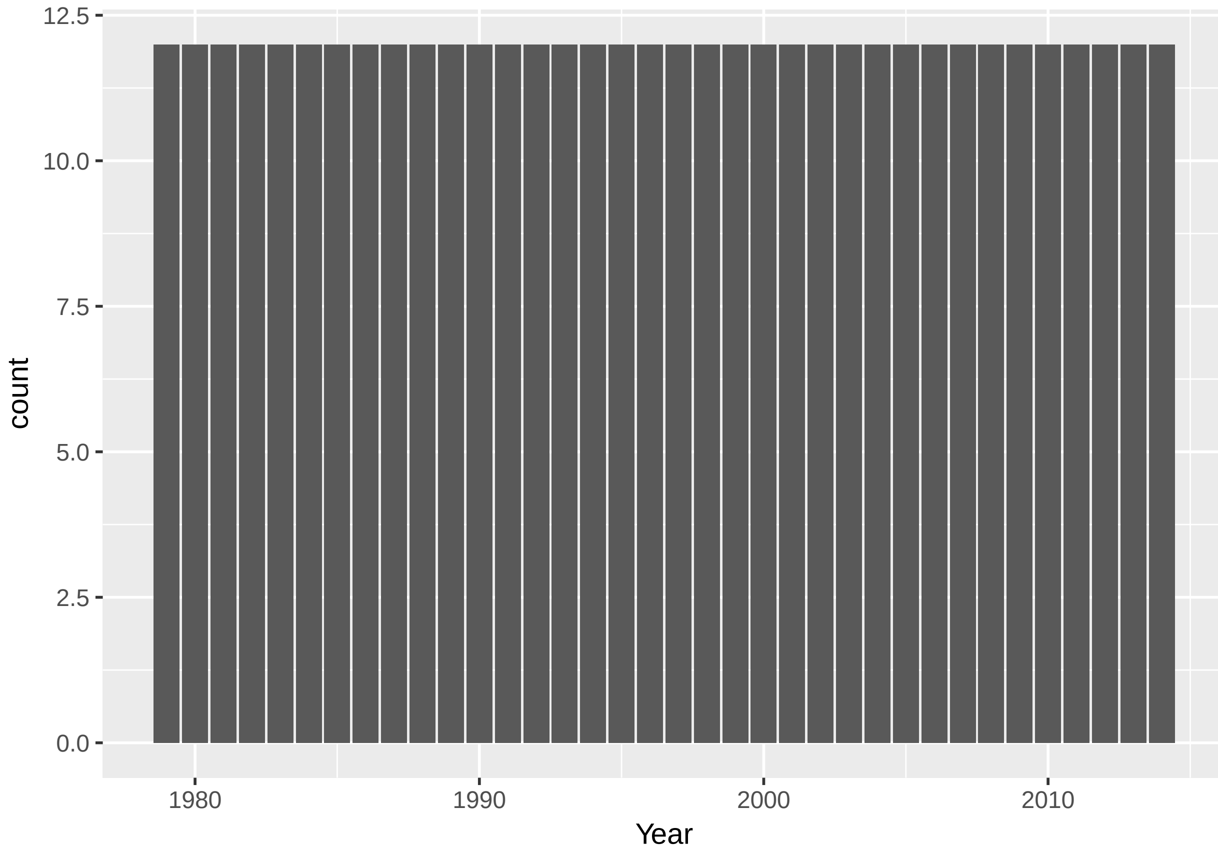

```
pl + geom_violin()
```



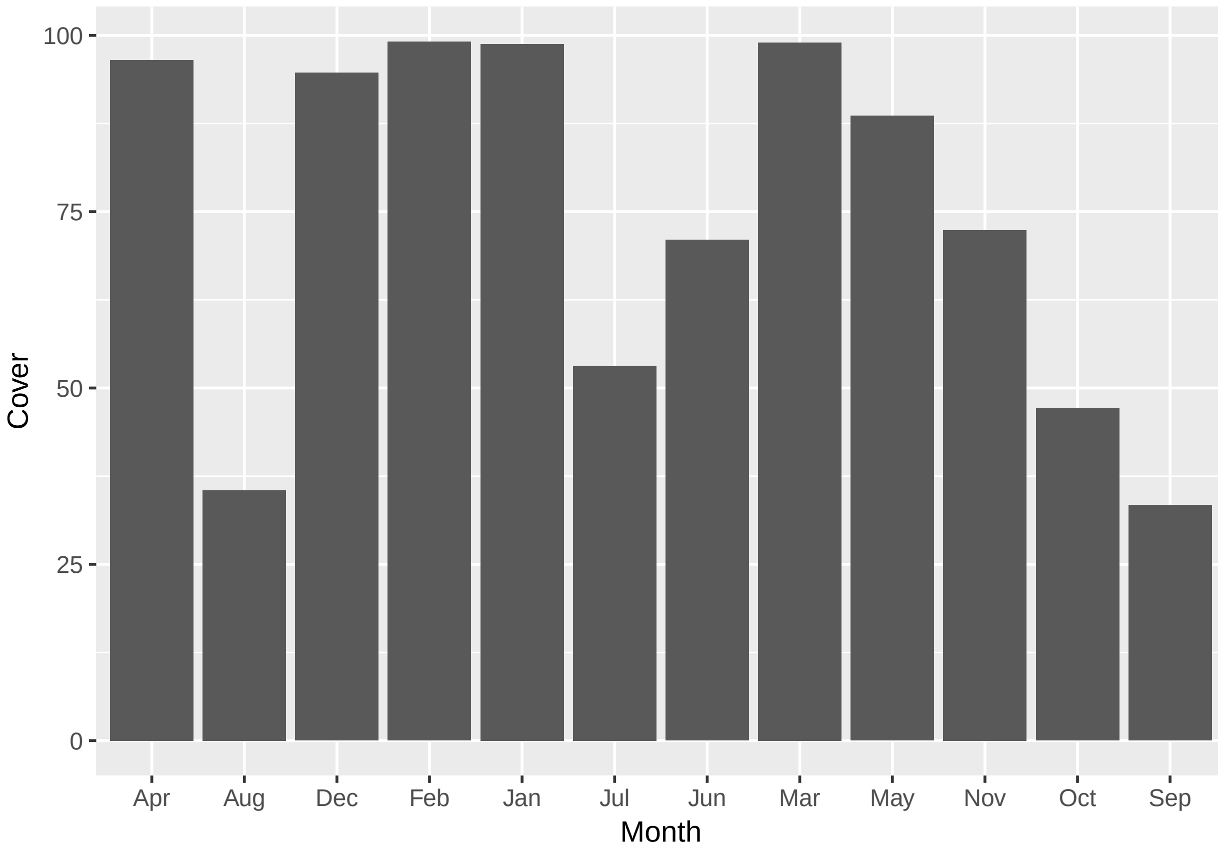

```
# change color of boxes
pl + geom_boxplot() + aes(fill = Herd)
```



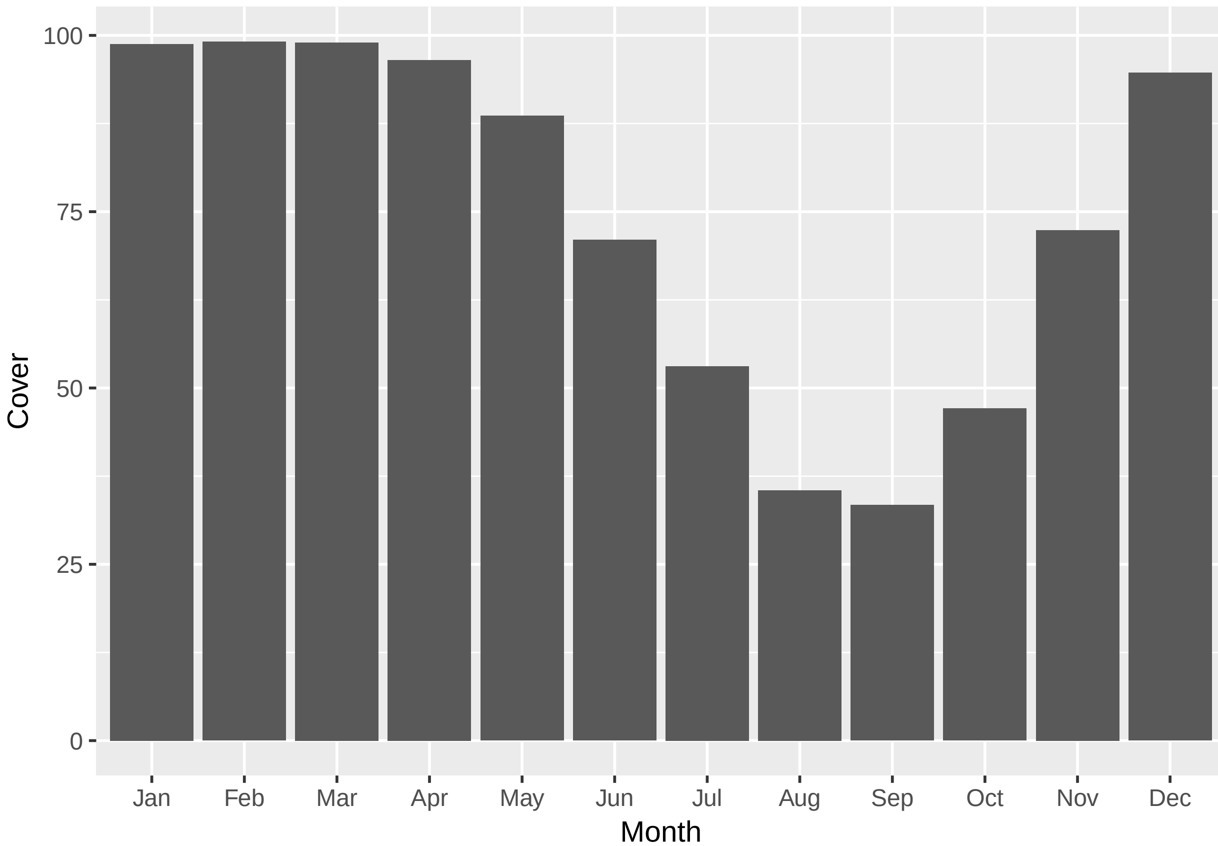

```
# barplot (count data)
ggplot(data = seaice %>% filter(Herd == "WAH")) + aes(x = Year) + geom_bar()
```



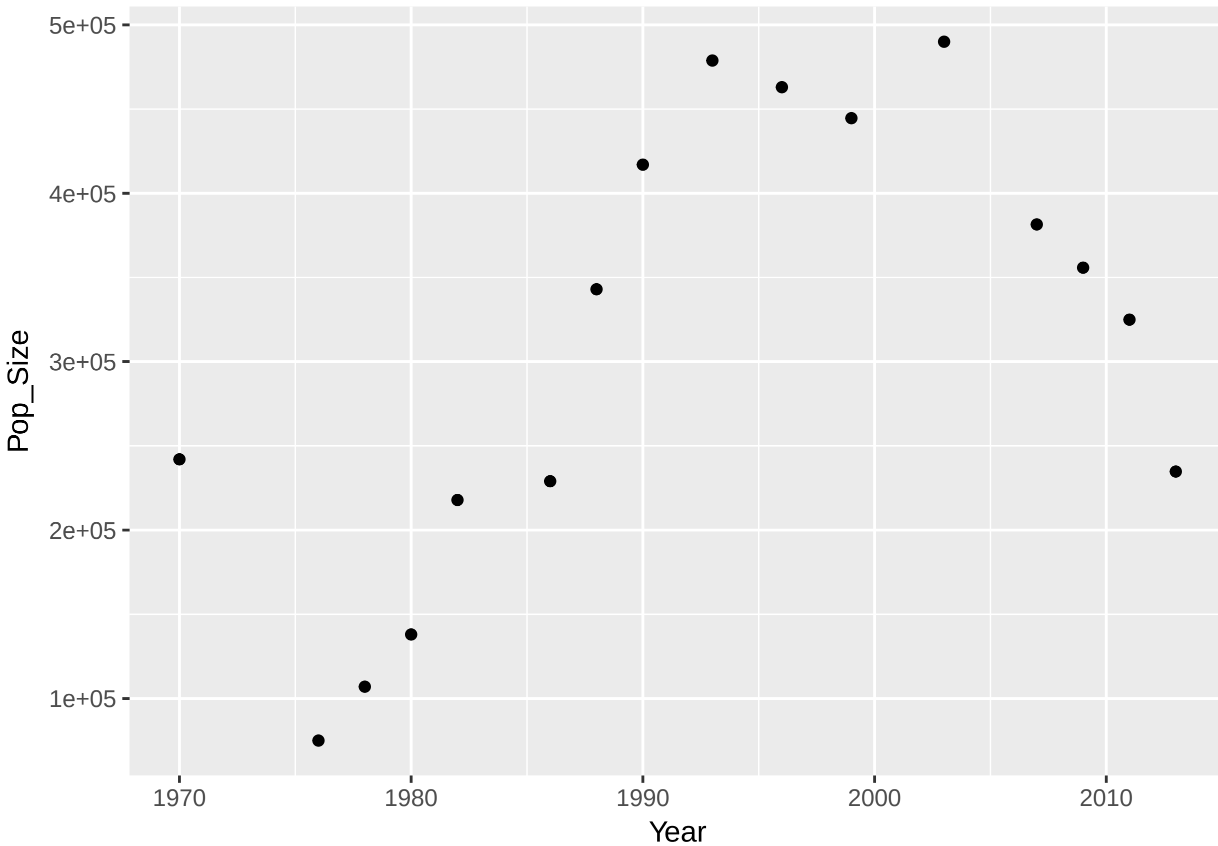

```r
# map data to columns (note alphabetical order of x-axis)
ggplot(data = seaice %>% filter(Herd == "WAH", Year == 1990)) + aes(x = Month, y = Cover) + geom_col()
```




```r
# display bars in chronolocigal order
# convert data into factor and set to three-letter abbreviation of months
seaice$Month <- factor(seaice$Month, month.abb)
ggplot(data = seaice %>% filter(Herd == "WAH", Year == 1990)) + aes(x = Month, y = Cover) + geom_col()
```

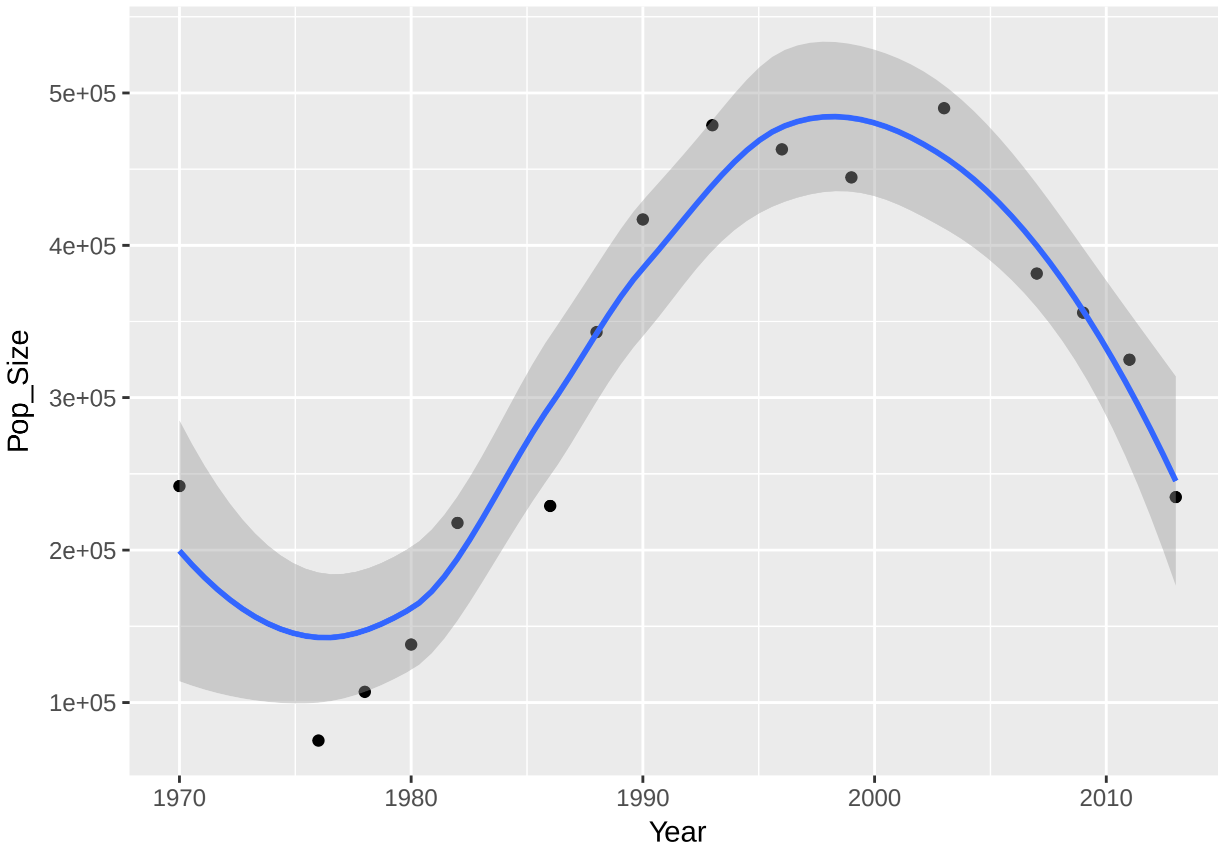



```
# scatterplots
pl <- ggplot(data = popsize %>% filter(Herd == "WAH")) + aes(x = Year, y = Pop_Size) + geom_point()
# show plot assigned to variable
show(pl)
```

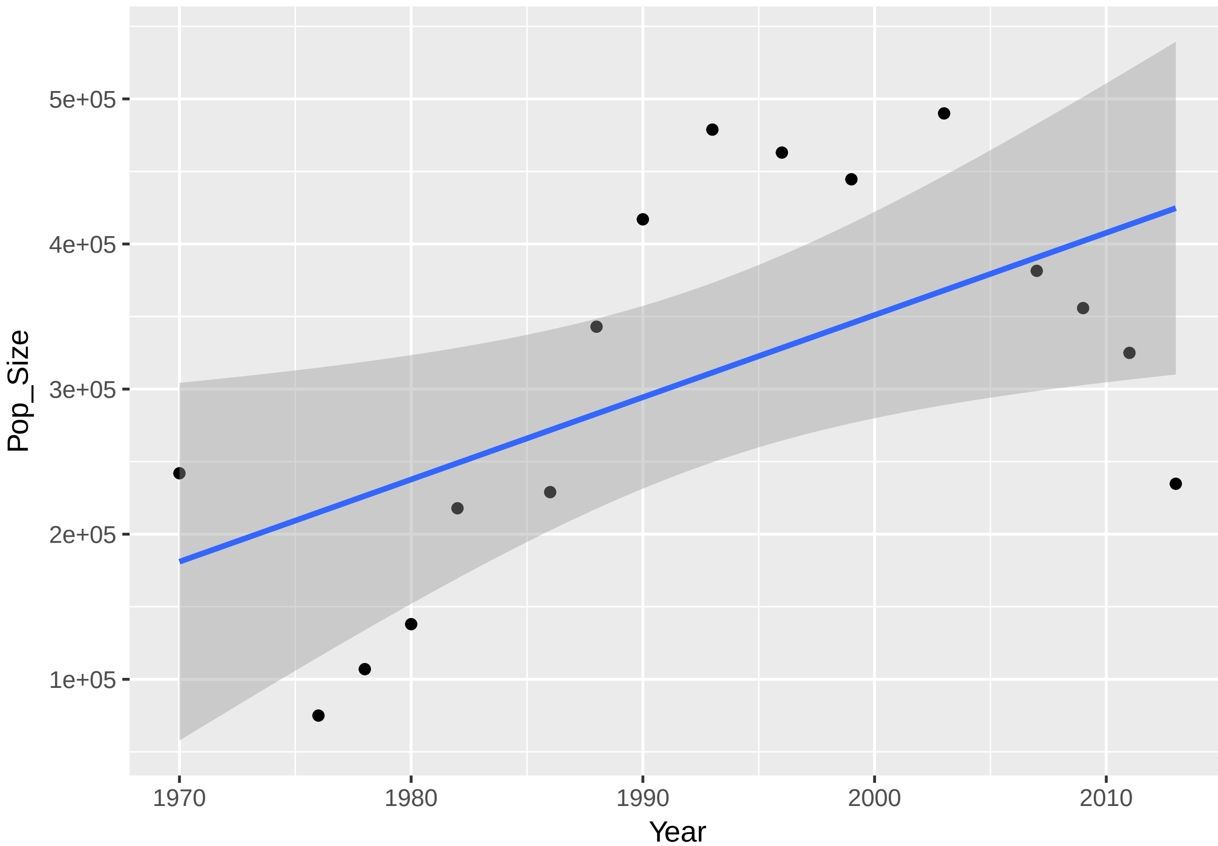



```
# add smoothing funtion
pl + geom_smooth()
```

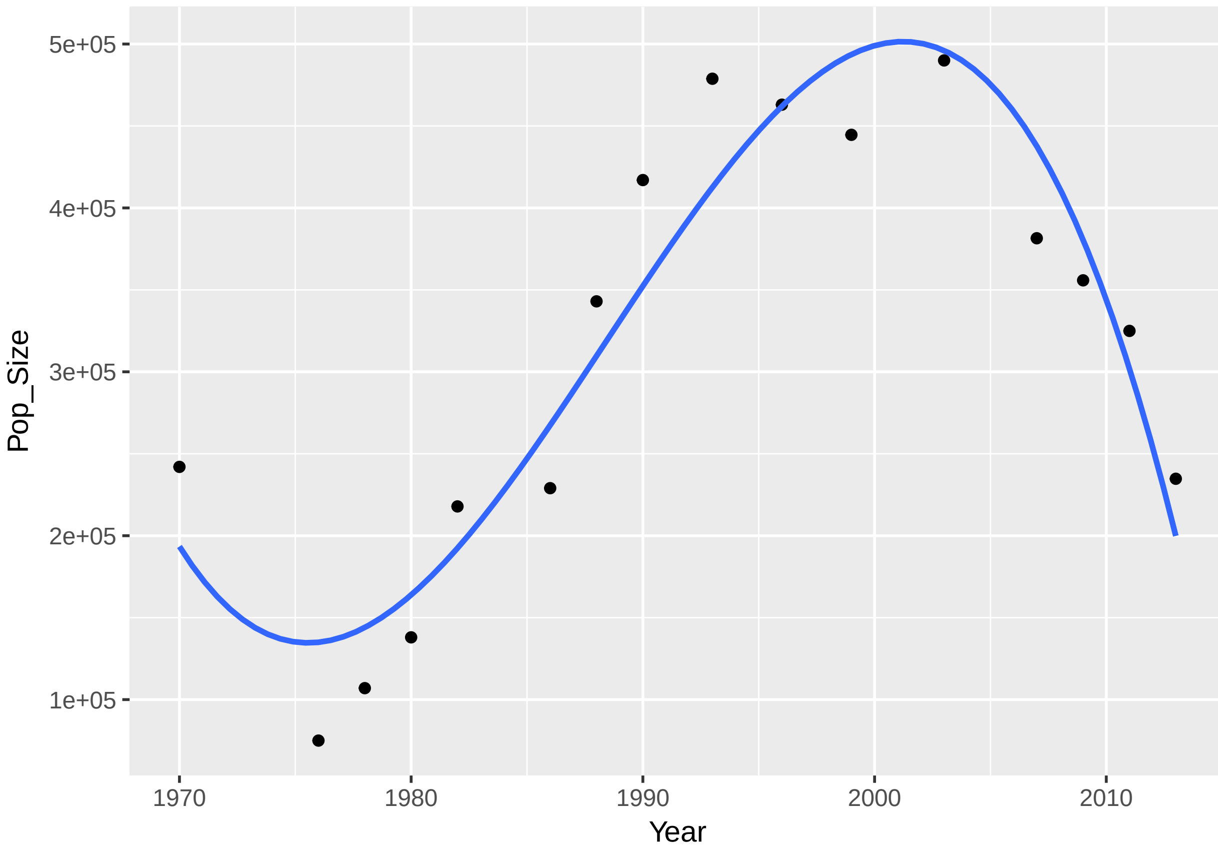
```
## `geom_smooth()` using method = 'loess'
```



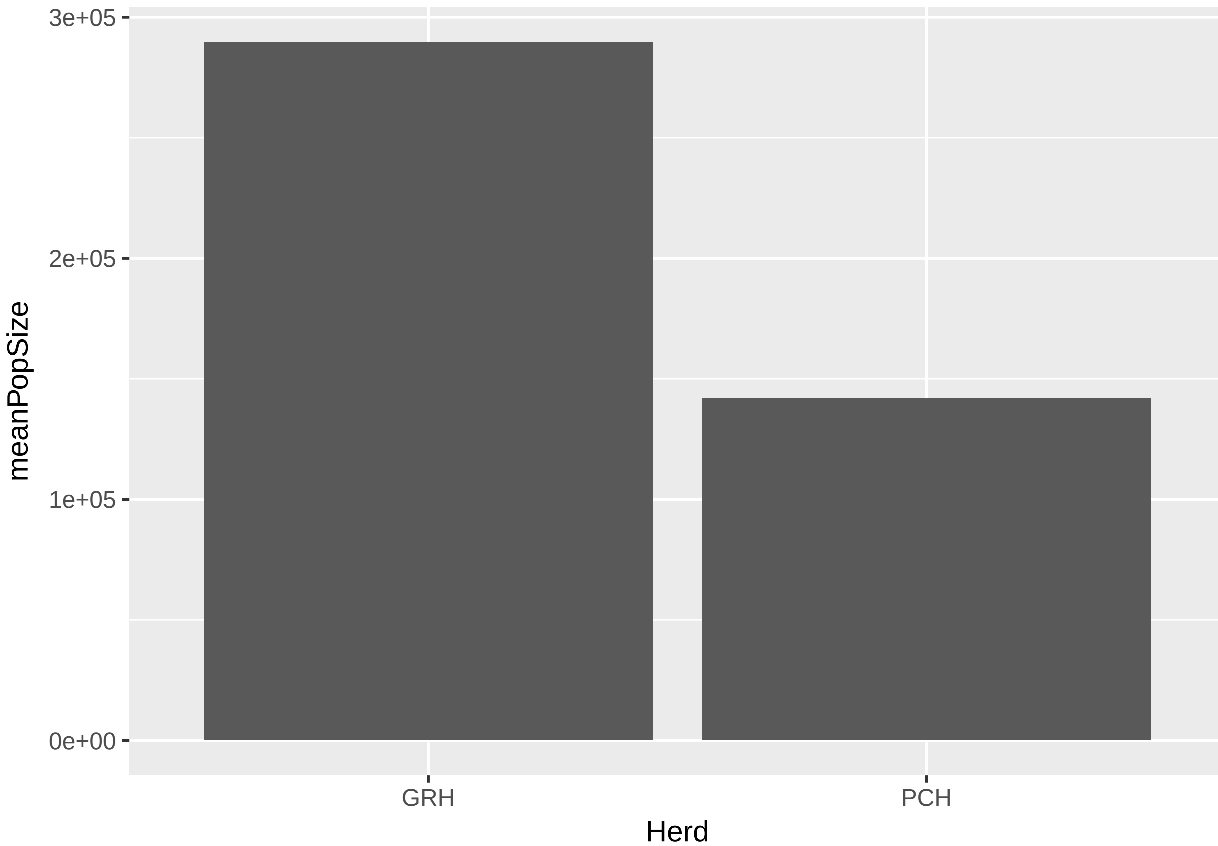

```r
# use a linear model
pl + geom_smooth(method = "lm")
```



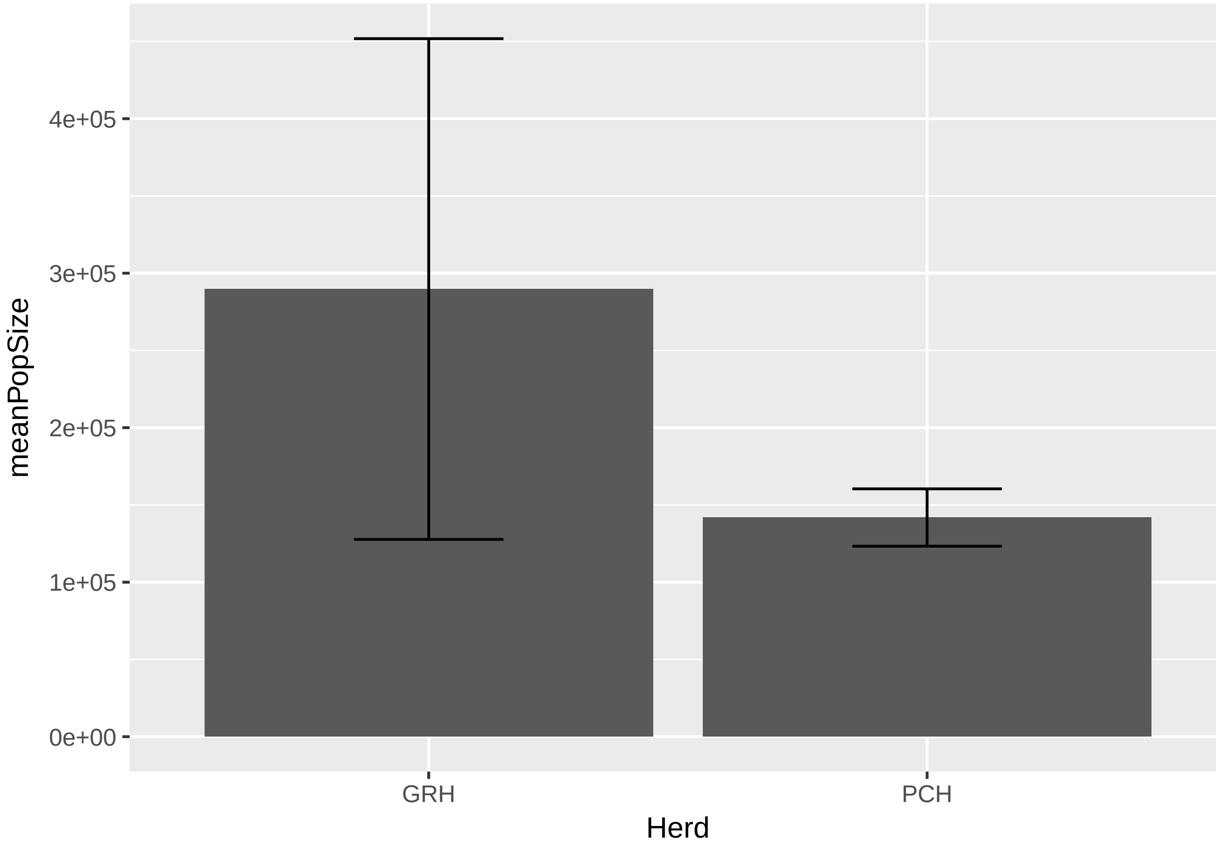

```r
# use a polynomial regression
pl + geom_smooth(method = "glm", formula = y ~poly(x, 3), se = FALSE)
```



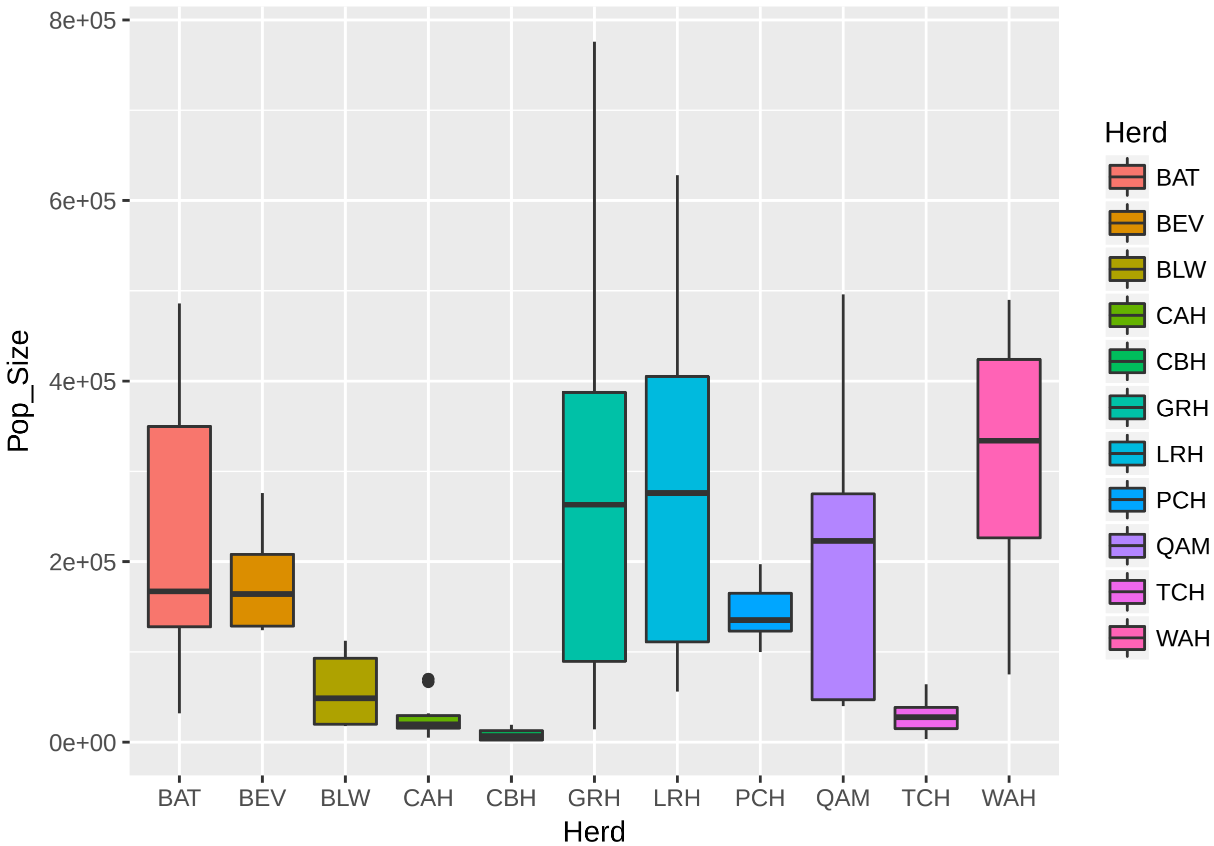

```r
# calculate summary stats and errors
stats <- popsize %>% filter(Herd %in% c("GRH", "PCH")) %>%
  group_by(Herd) %>%
  summarise(
    meanPopSize= mean(Pop_Size),
    SD = sd(Pop_Size),
    N = n(),
    SEM = SD/sqrt(N),
    CI = SEM * qt(0.975, N-1))
# bar plot without error bars
ggplot(data = stats) +
  aes(x = Herd, y = meanPopSize) +
  geom_col()
```



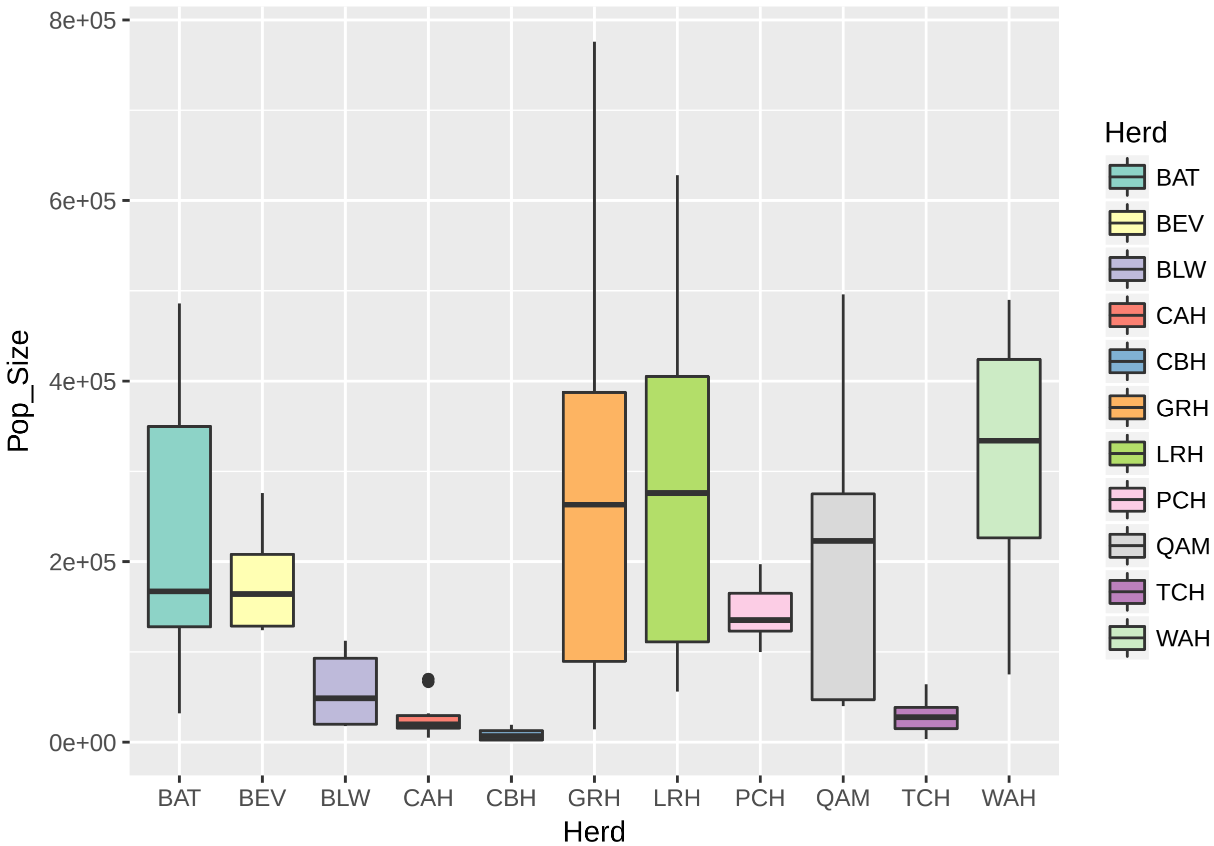

```r
# set up aesthetic mapping for confidence intervals
limits <- aes(ymax = stats$meanPopSize + stats$CI,
              ymin = stats$meanPopSize - stats$CI)
# plot including confidence intervals
ggplot(data = stats) +
  aes(x = Herd, y = meanPopSize) +
  geom_col() +
  geom_errorbar(limits, width = .3)
```



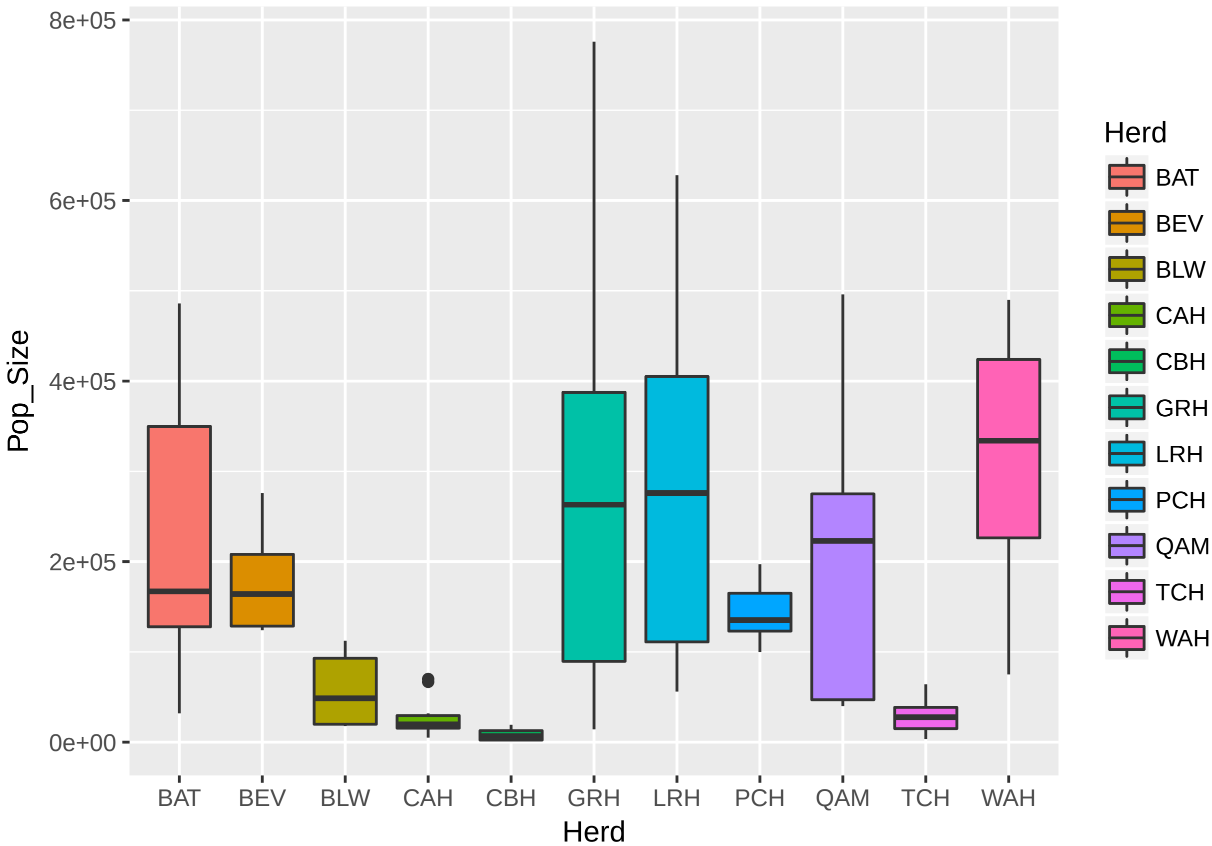

```
# set fill color of boxes by using scales
pl <- ggplot(data = popsize, aes(x = Herd, y = Pop_Size, fill = Herd)) + geom_boxplot()
show(pl)
```



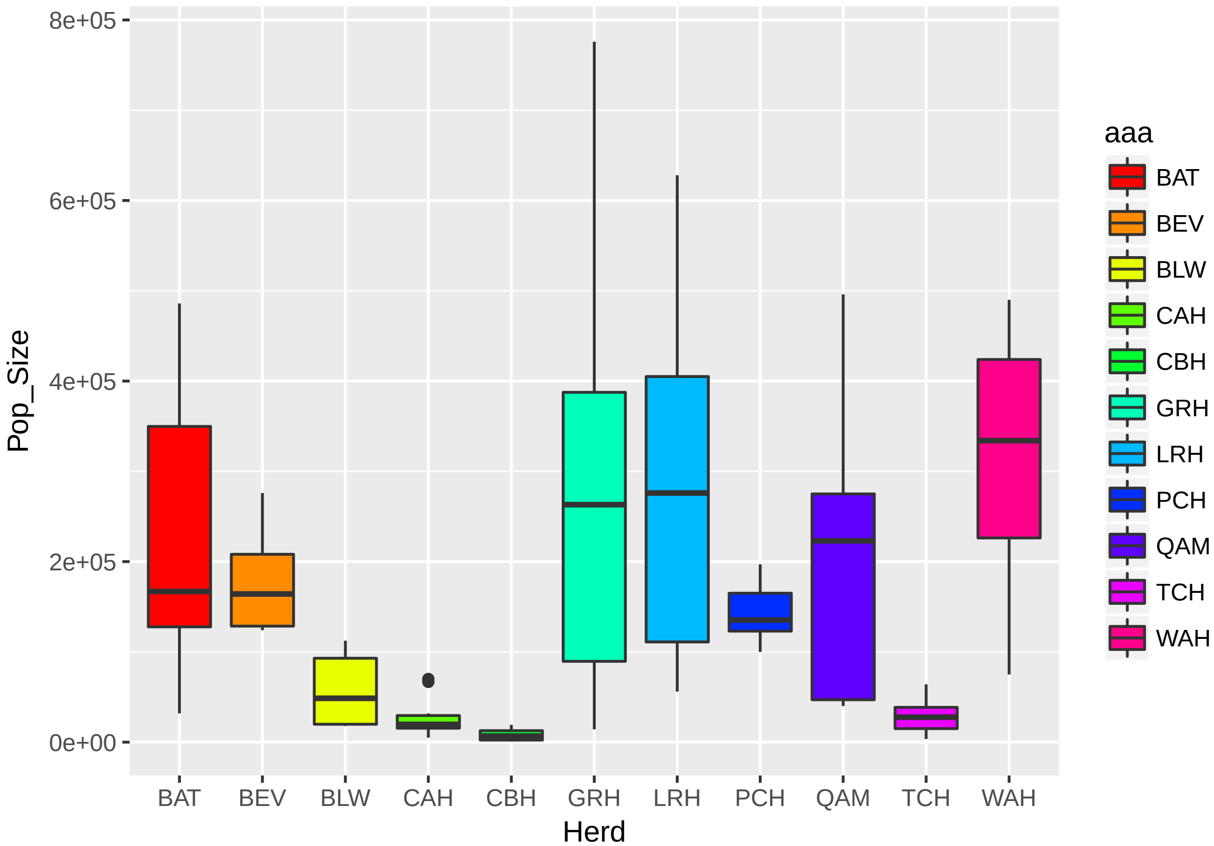

```
# choose a palette from Color Brewer
pl + scale_fill_brewer(palette = "Set3")
```



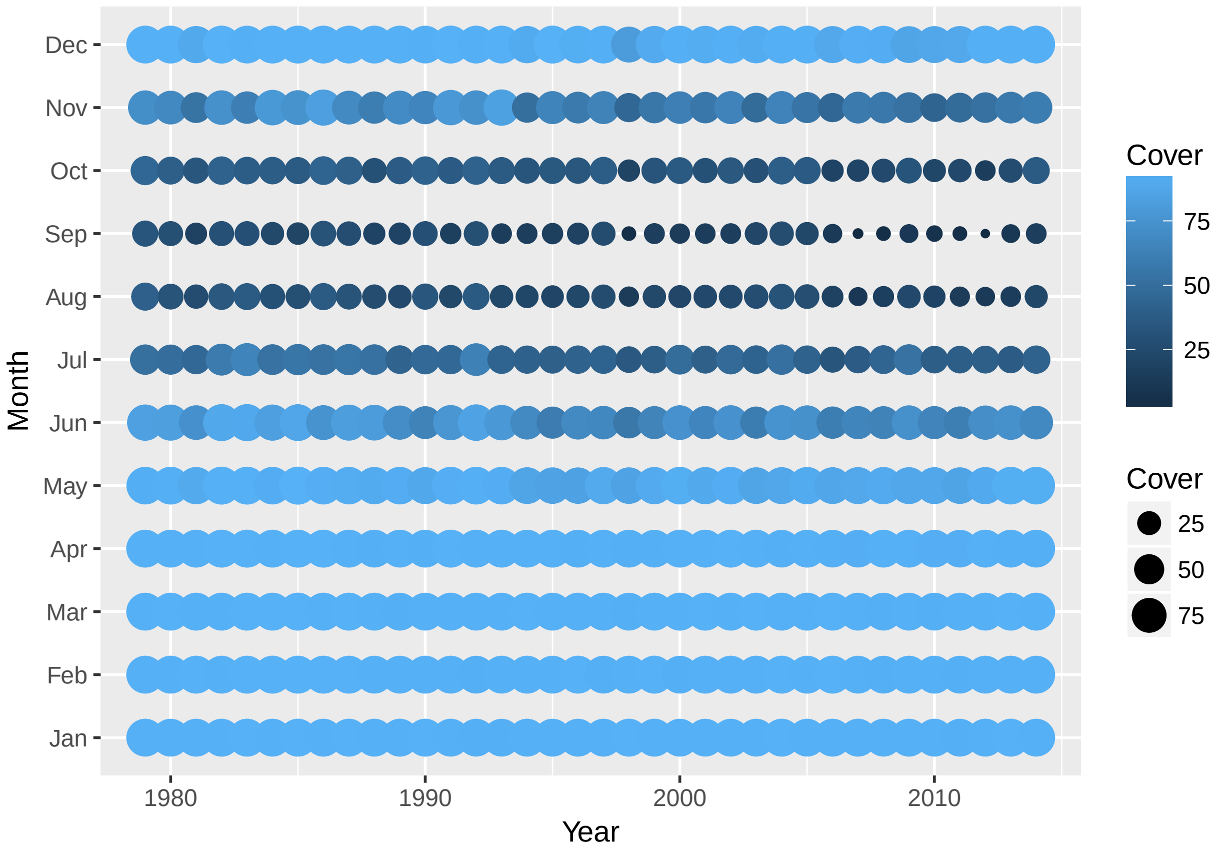

```
# palette based on hue
pl + scale_fill_hue()
```



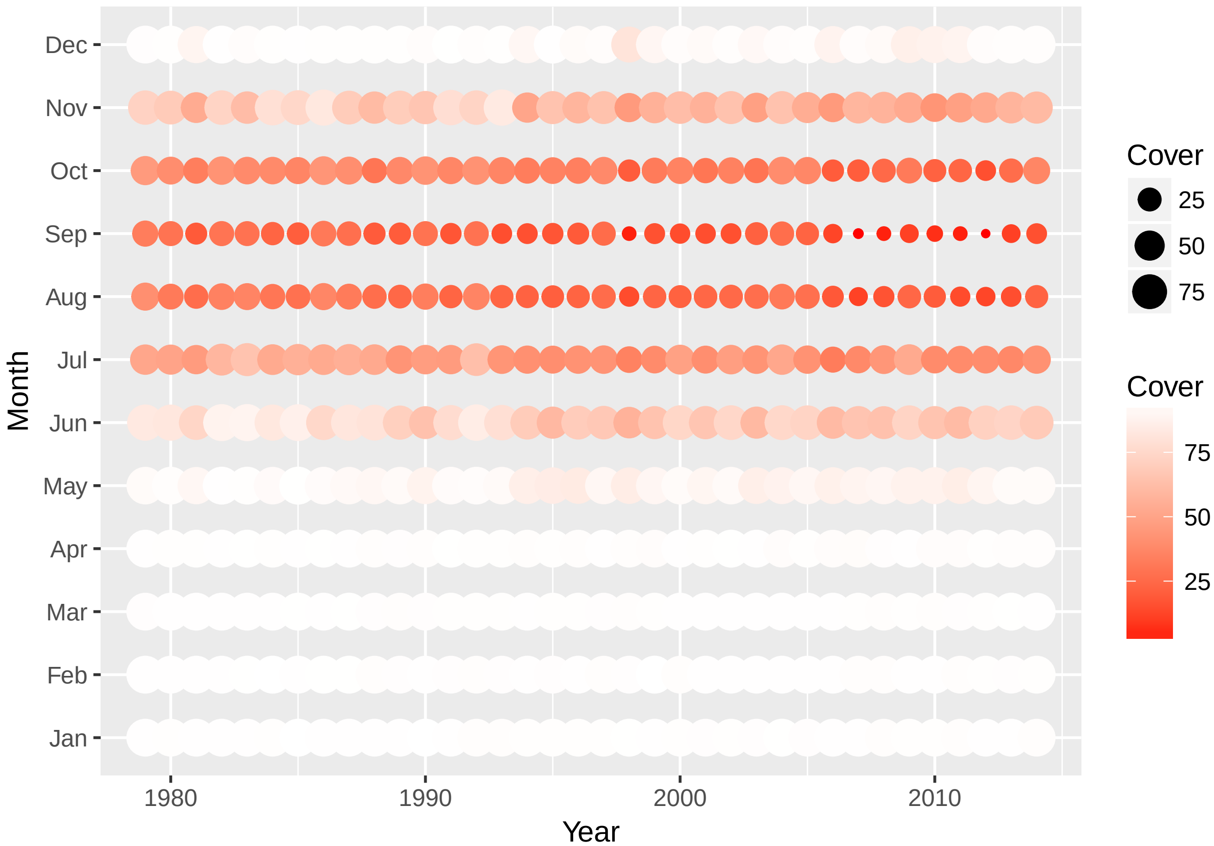

```r
# manually set values and rename the legend
pl + scale_fill_manual(values = rainbow(11), name = "aaa")
```



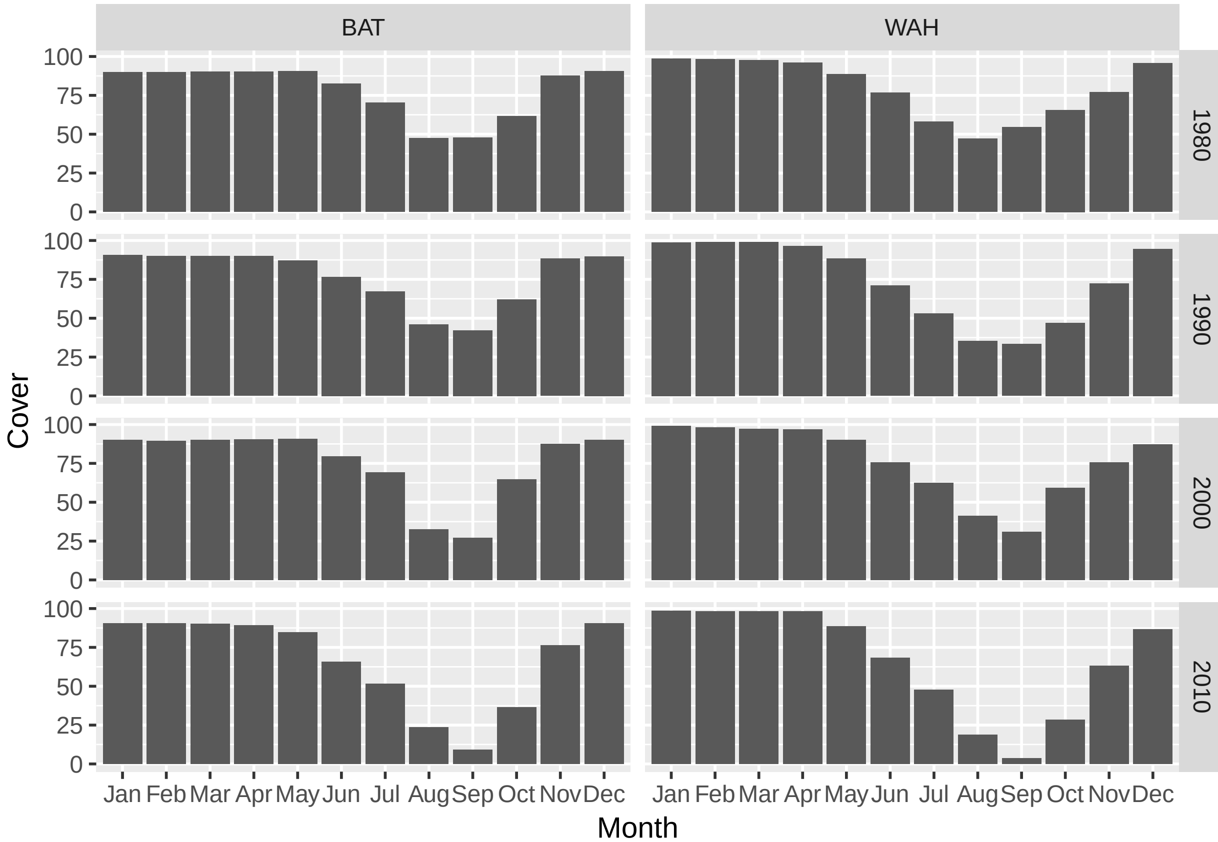

```
# apply scales to manipulate color and size of aestethic mappings
pl <- ggplot(data = seaice %>% filter(Herd == "BEV")) + aes(x = Year, y = Month, colour = Cover, size =
show(pl)
```



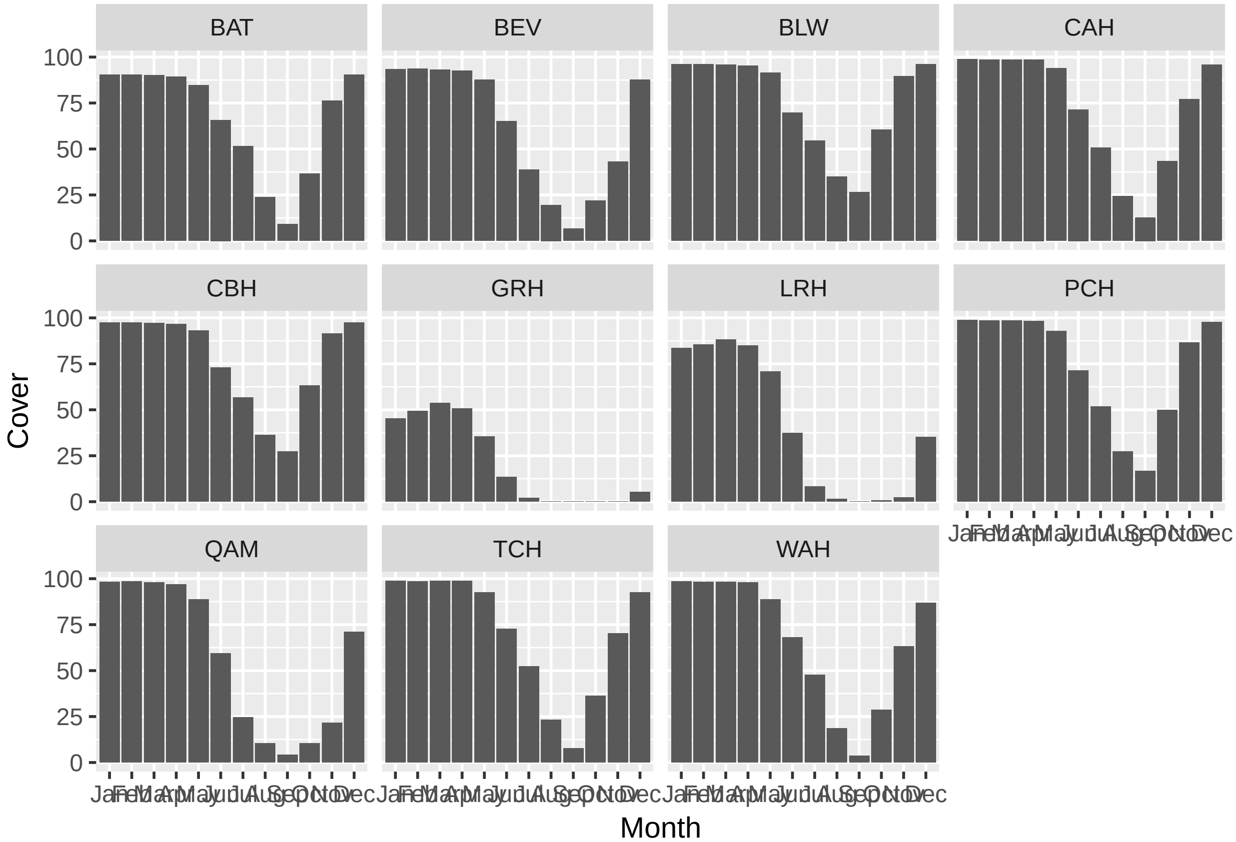

```
# change color of continuous gradient
pl + scale_color_gradient(high = "white", low = "red")
```



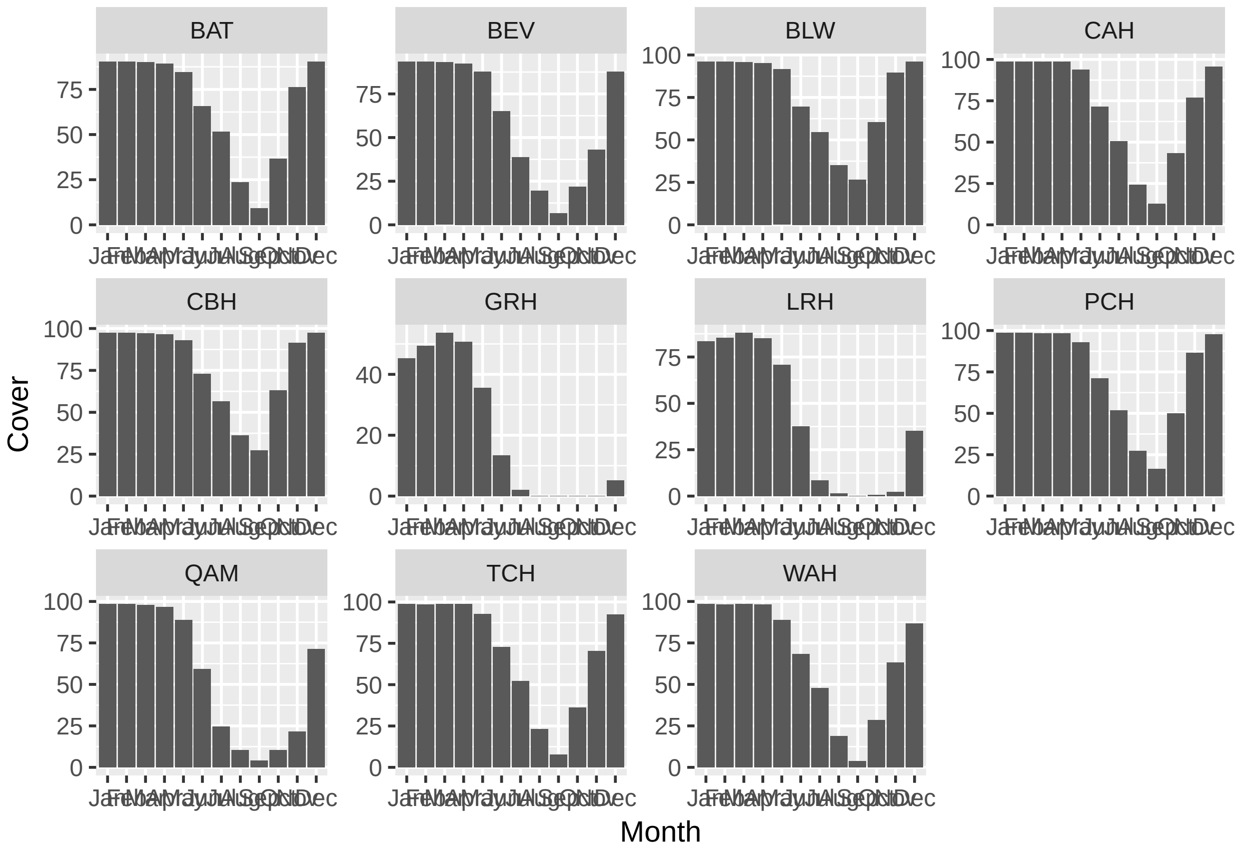

```
# facetting with identical scale of axis, including missing data
ggplot(data = seaice %>% filter(Herd %in% c("WAH", "BAT"), Year %in% c(1980, 1990, 2000, 2010))) + aes(
```



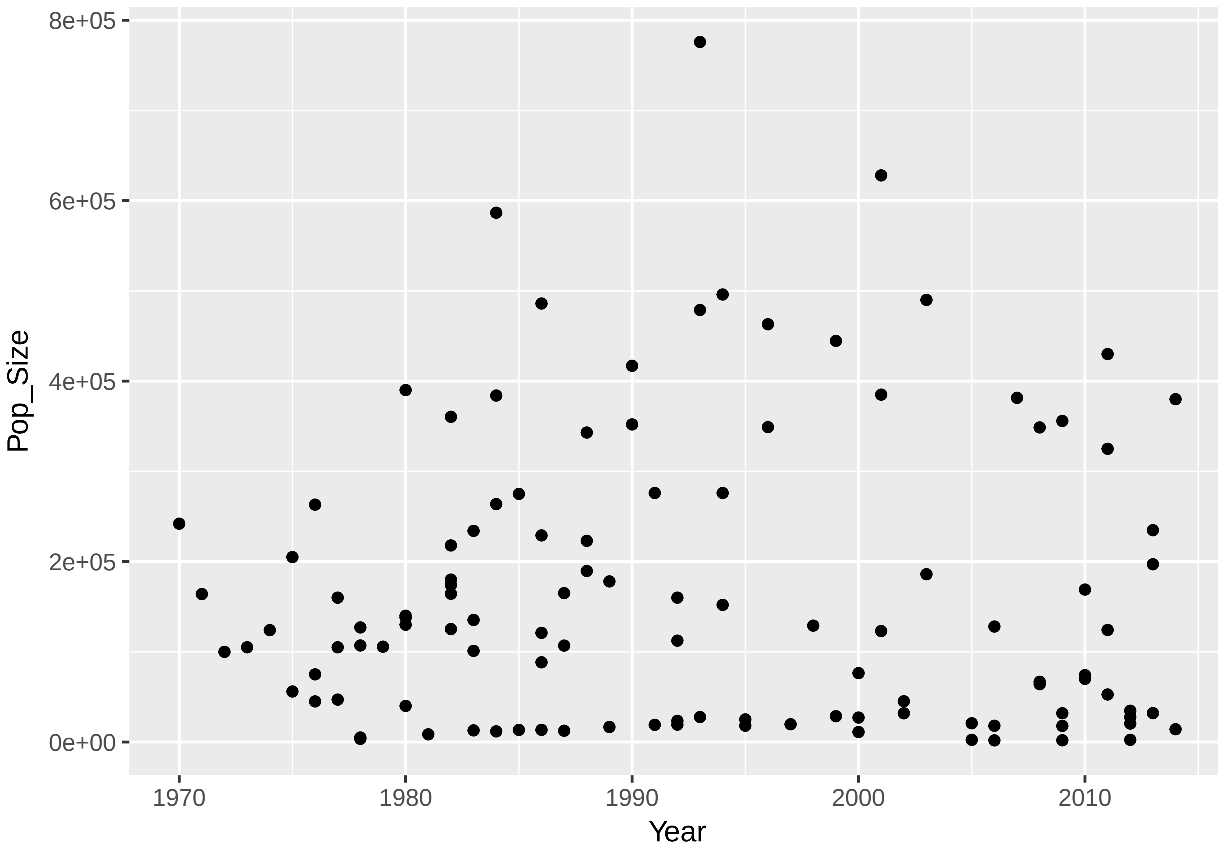

```
# facetting, ommit missing data
ggplot(data = seaice %>% filter(Year == 2010)) + aes(x = Month, y = Cover) + geom_col() + facet_wrap(~Ho
```



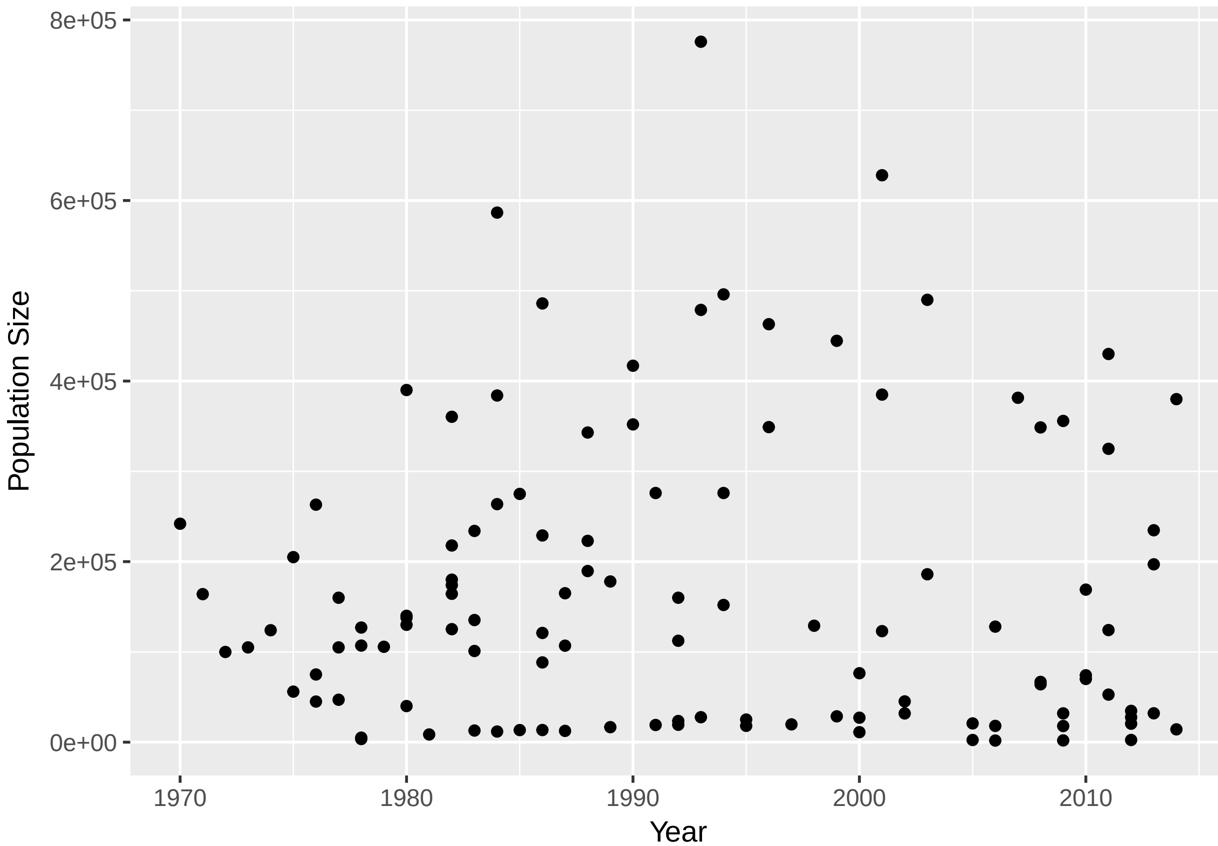

```r
# facetting, ommit missing data, adjusted scale of axes
ggplot(data = seaice %>% filter(Year == 2010)) + aes(x = Month, y = Cover) + geom_col() + facet_wrap(~H
```



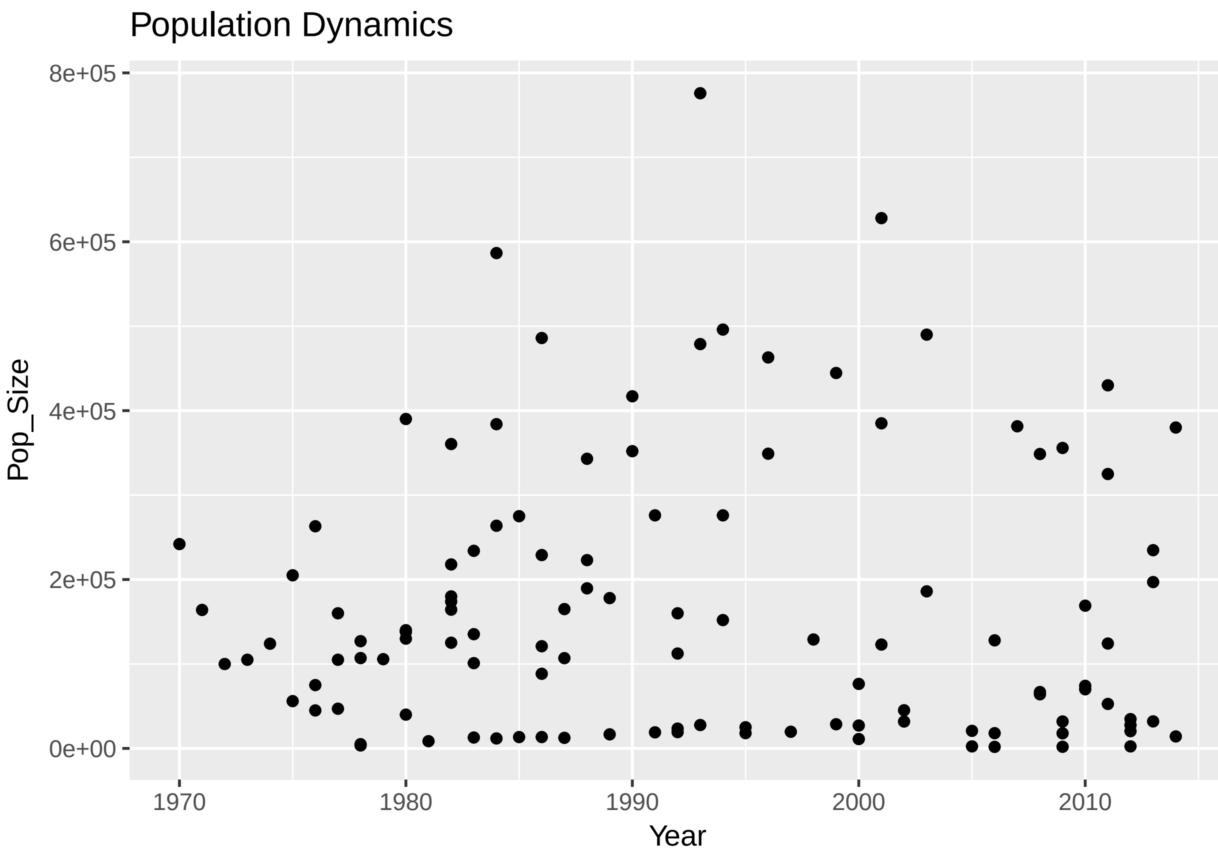

```r
# changing labels and title
pl <- ggplot(data = popsize) + aes(x = Year, y = Pop_Size) + geom_point()
pl + xlab("Year")
```



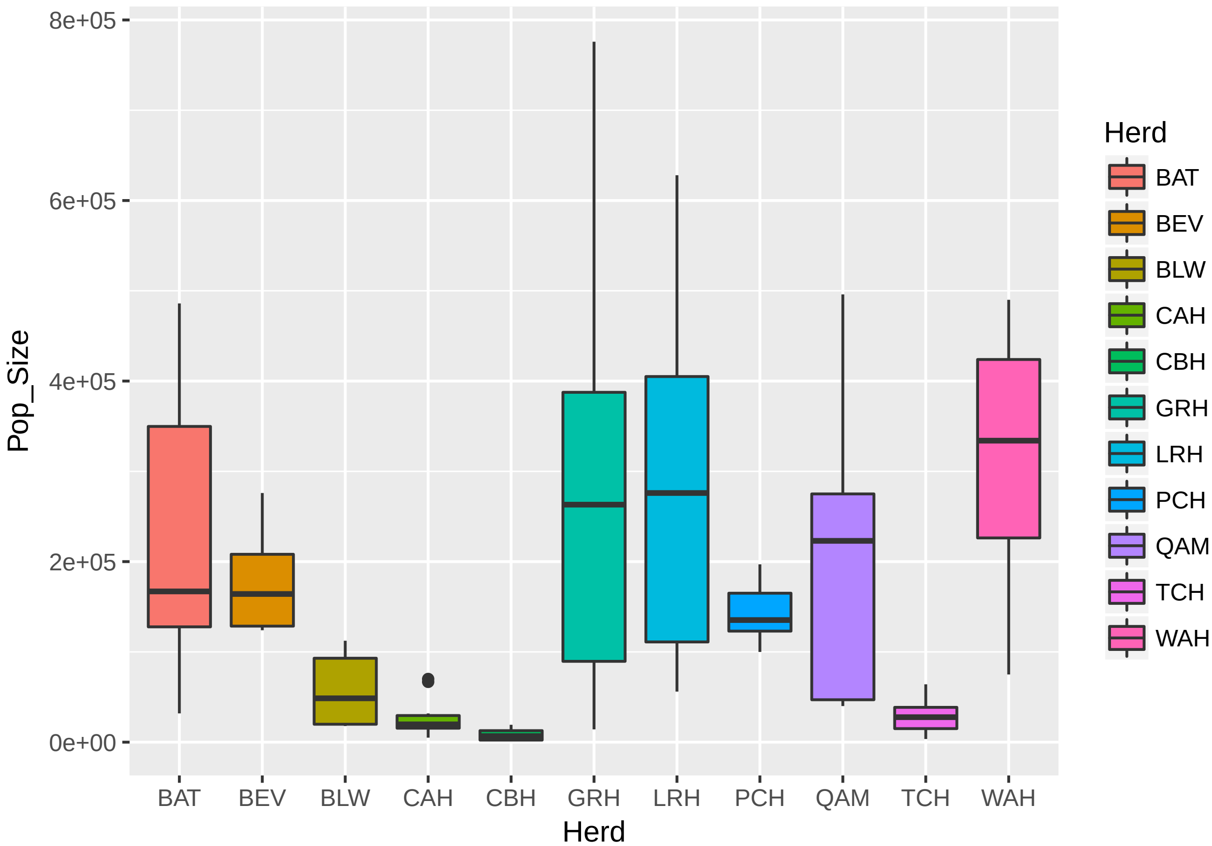

```
pl + ylab("Population Size")
```



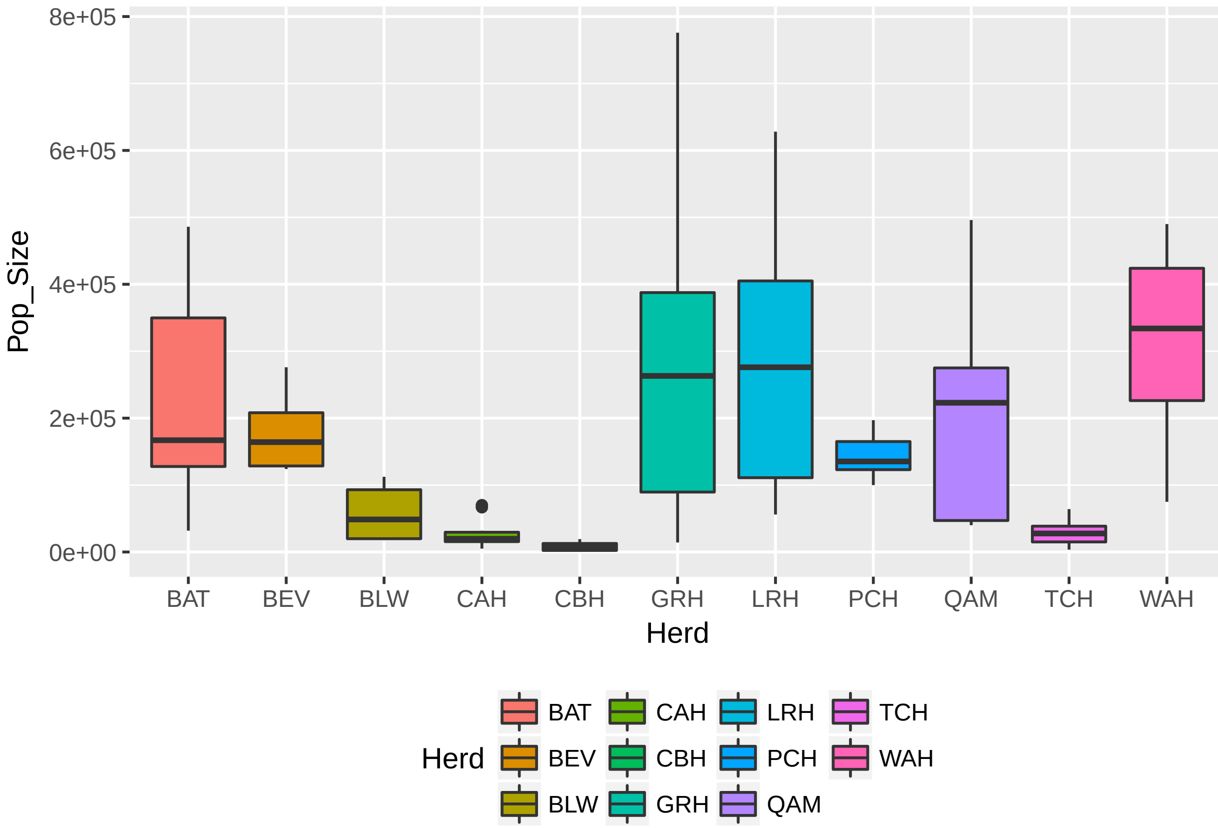

```
pl + ggtitle("Population Dynamics")
```



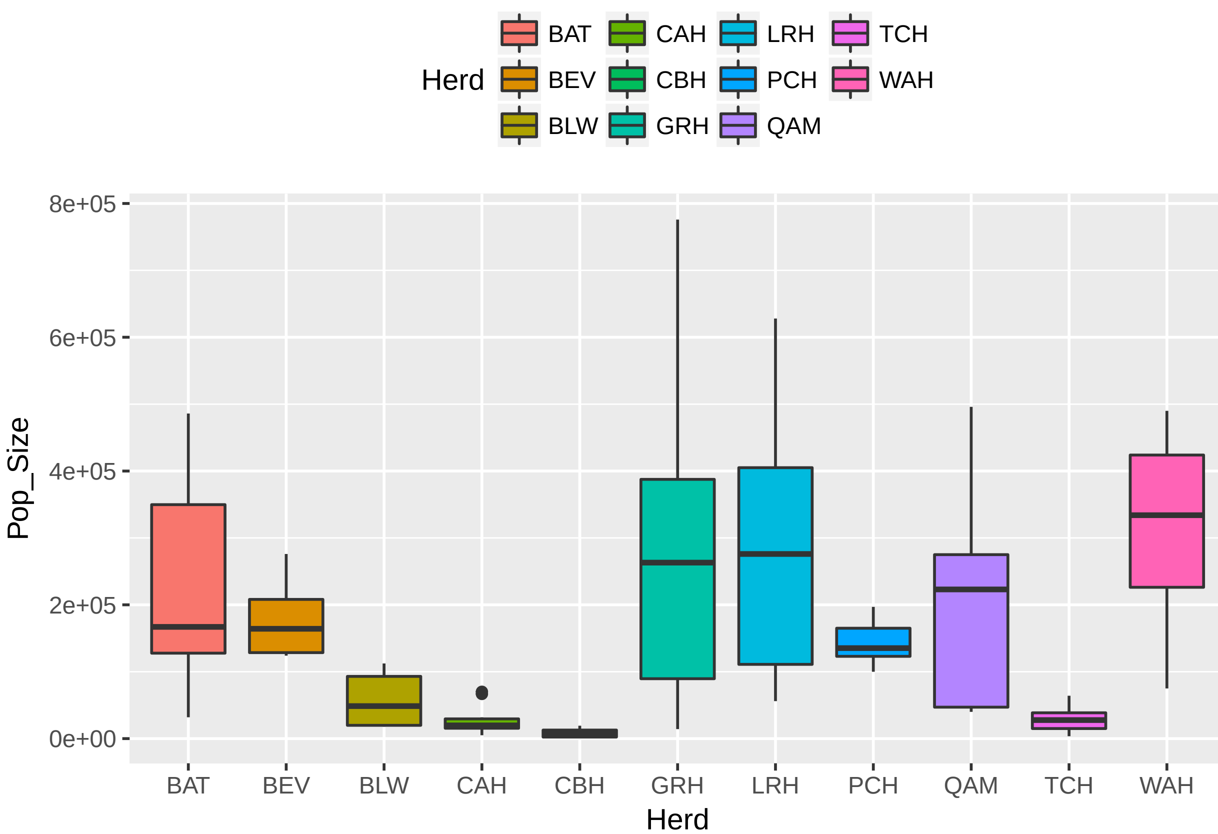

```
# legends
pl <- ggplot(data = popsize) + aes(x = Herd, y = Pop_Size, fill = Herd) + geom_boxplot()
# default
show(pl)
```



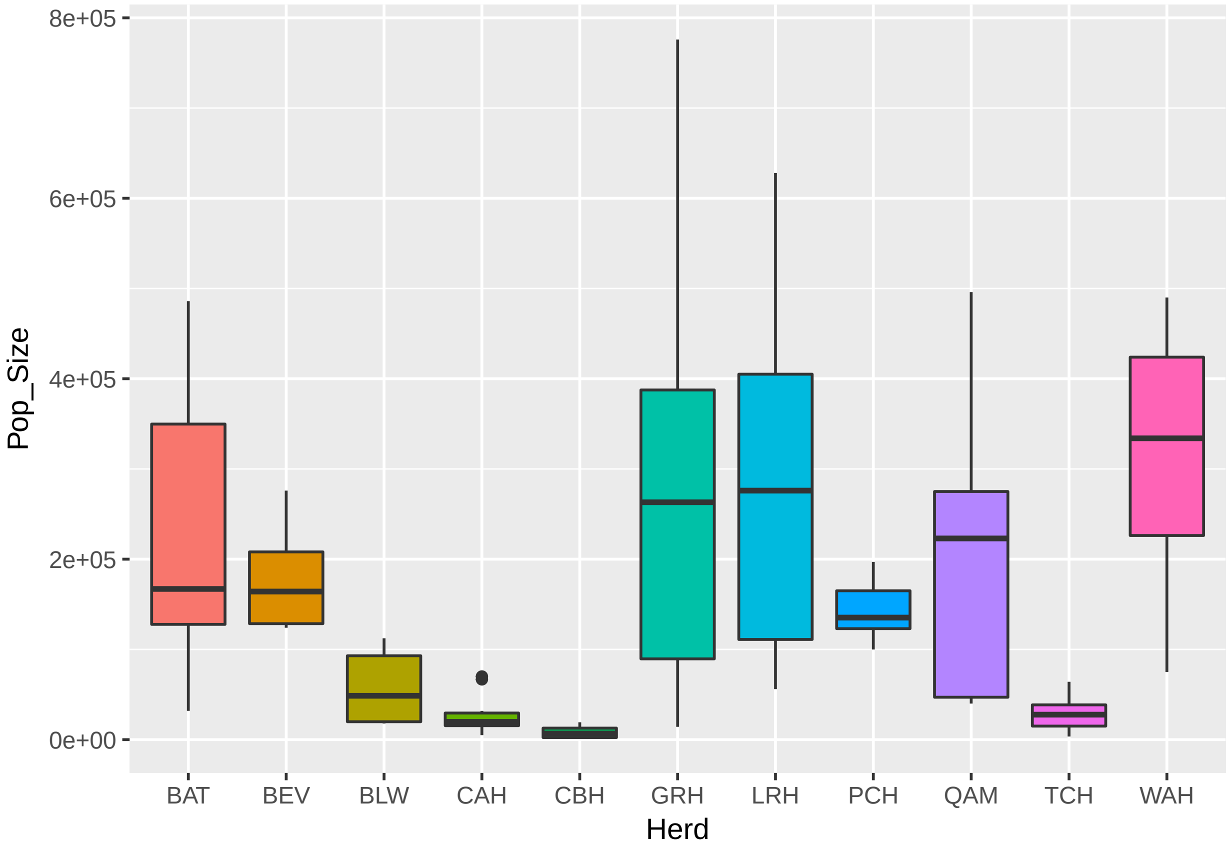

```
# move legend
pl + theme(legend.position = "bottom")
```



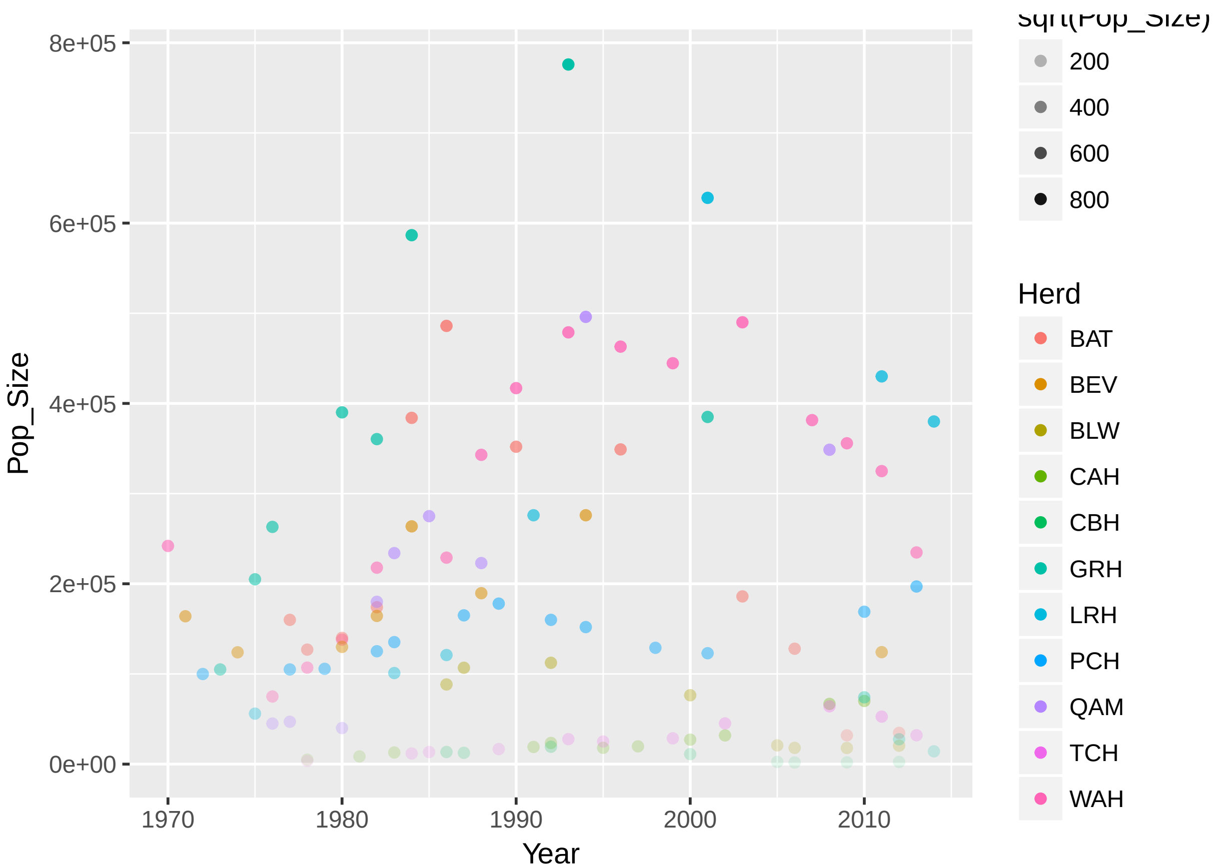

```
pl + theme(legend.position = "top")
```



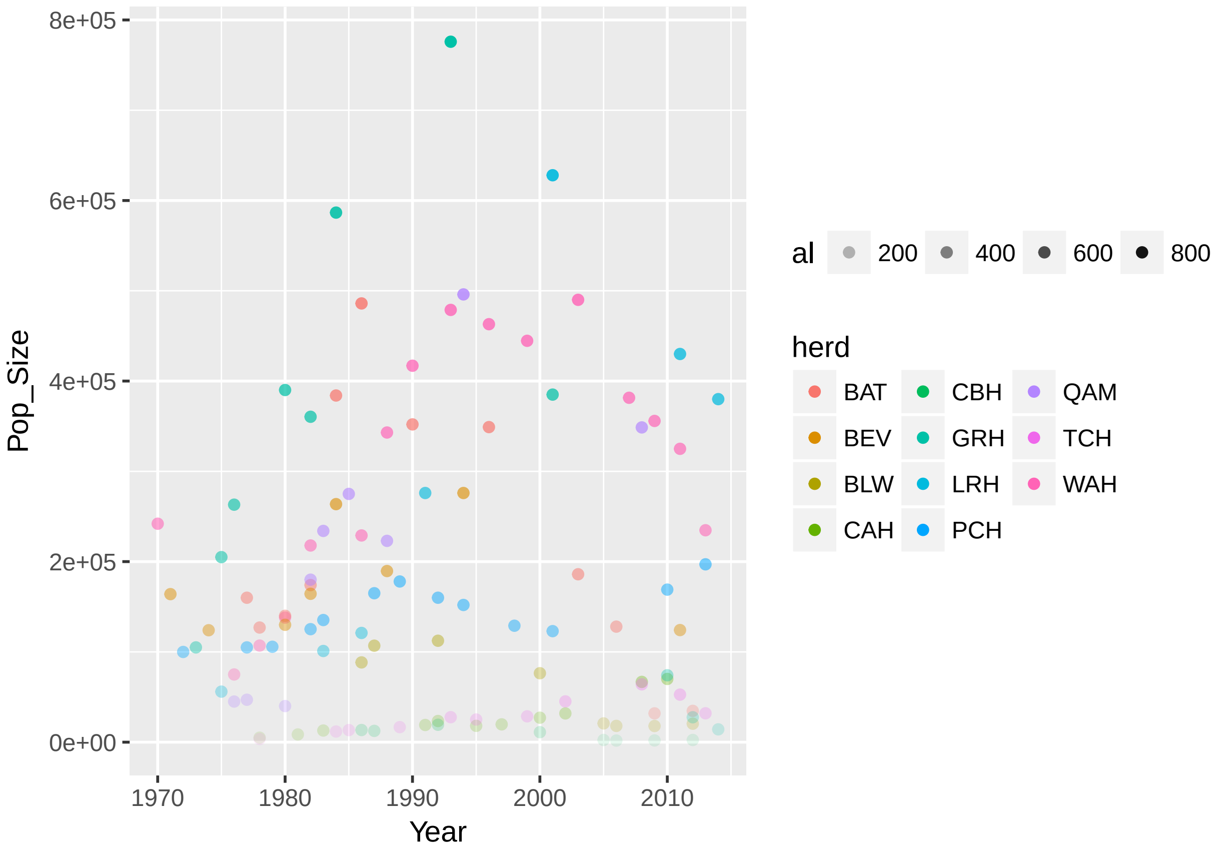
```r
# remove legend
pl + theme(legend.position = "none")
```



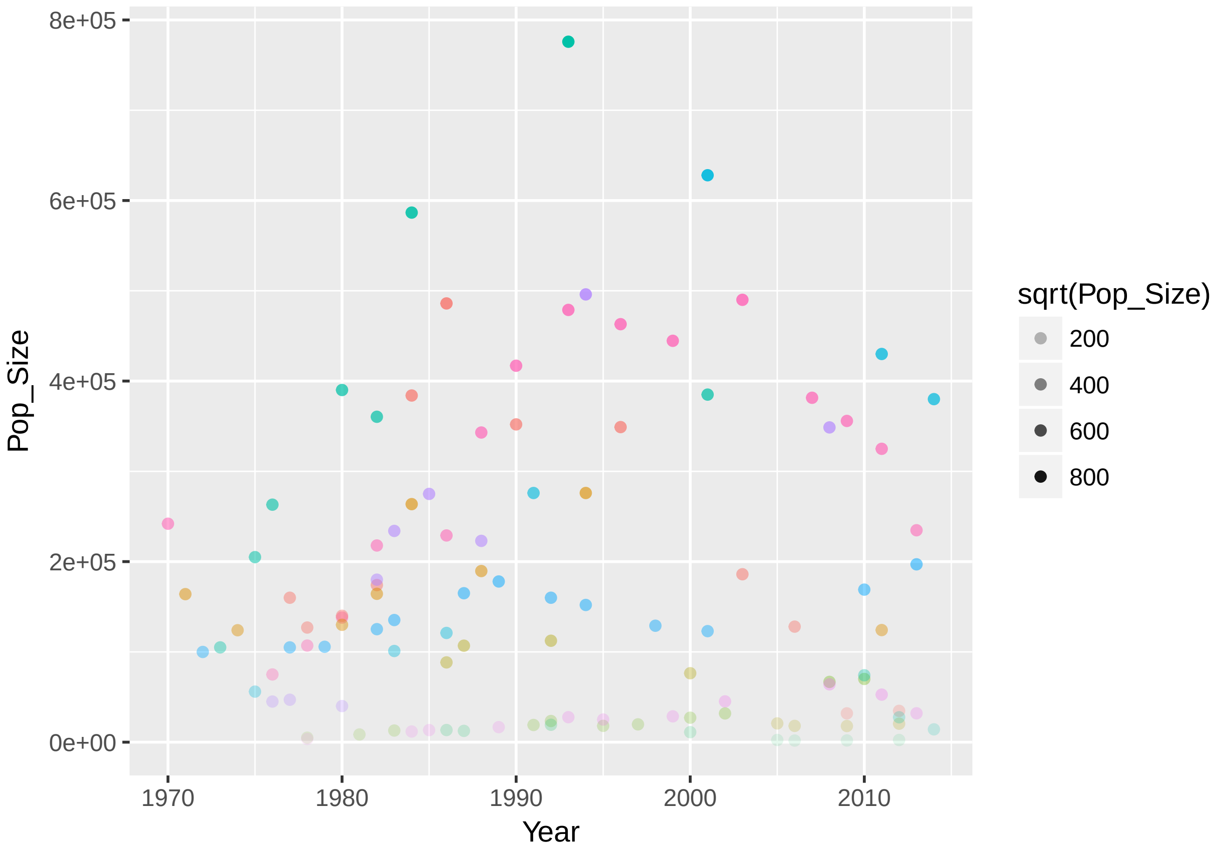

```r
# legend guides
pl <- ggplot(data = popsize) + aes(x = Year, y = Pop_Size, colour = Herd, alpha = sqrt(Pop_Size)) + geom
show(pl)
```




```
pl + guides(colour = guide_legend(nrow = 4, title = "herd"), alpha = guide_legend(direction = "horizonta
```

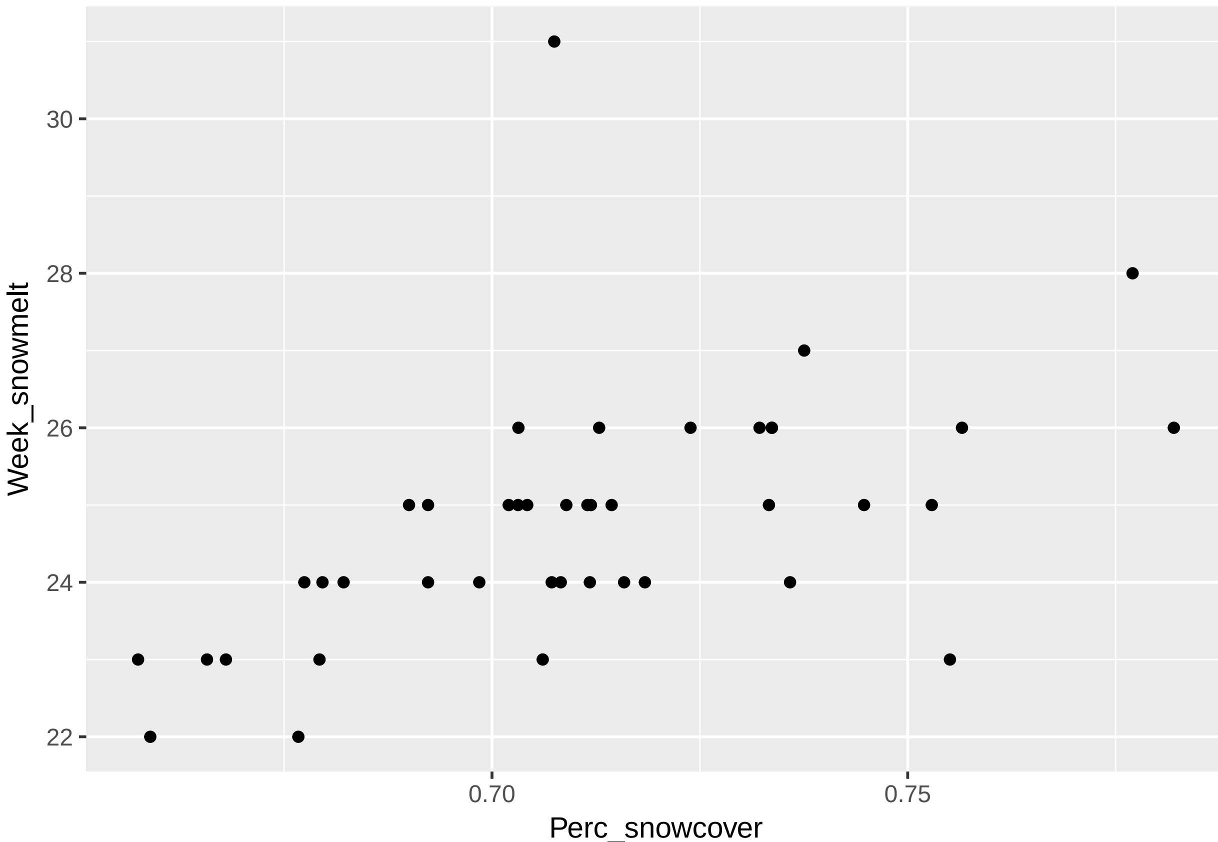



```
# suppress only one legend
pl + guides(colour = "none")
```



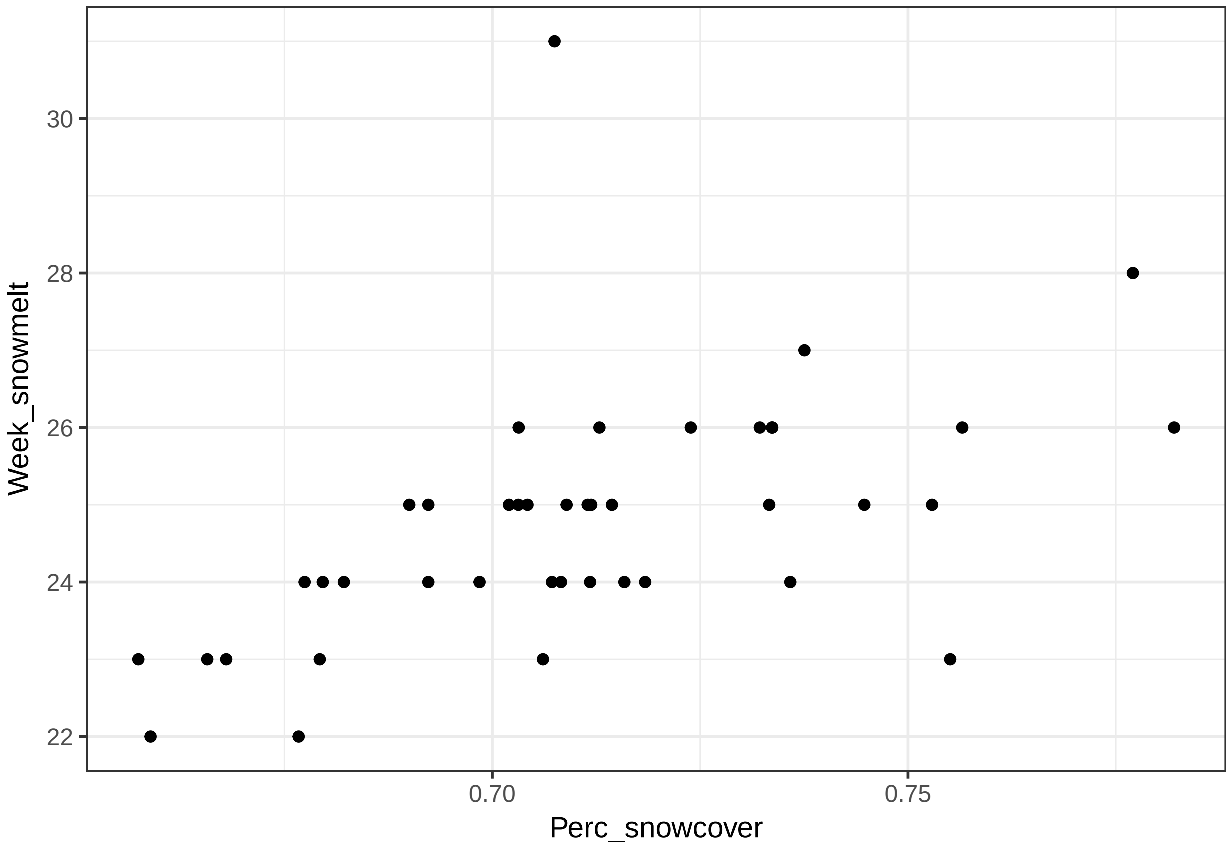

```
# themes
pl <- ggplot(data = snow %>%
               filter(Herd == "CAH"),
             aes(y = Week_snowmelt, x = Perc_snowcover)) +
  geom_point()
# default theme with grey background and white gridlines
show(pl)
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



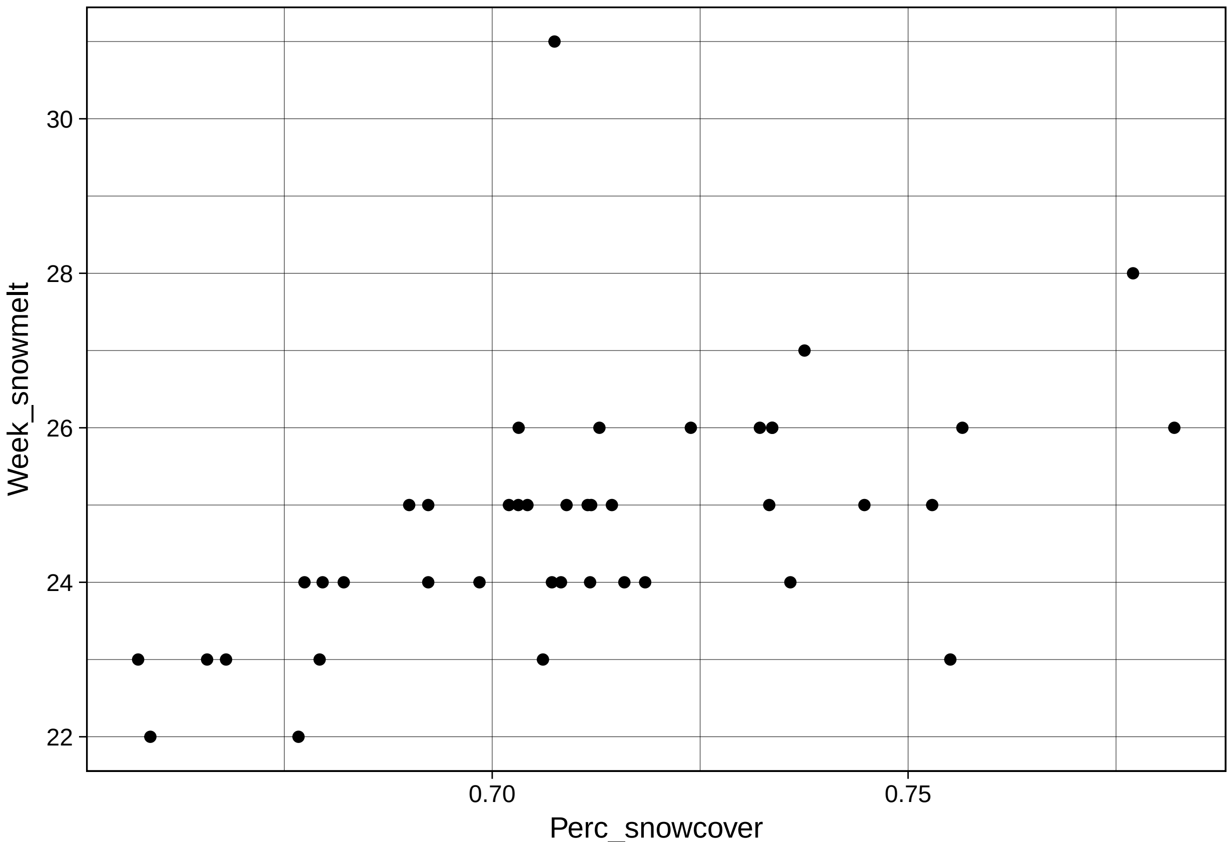

```
# black and white (light background)
pl + theme_bw()
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



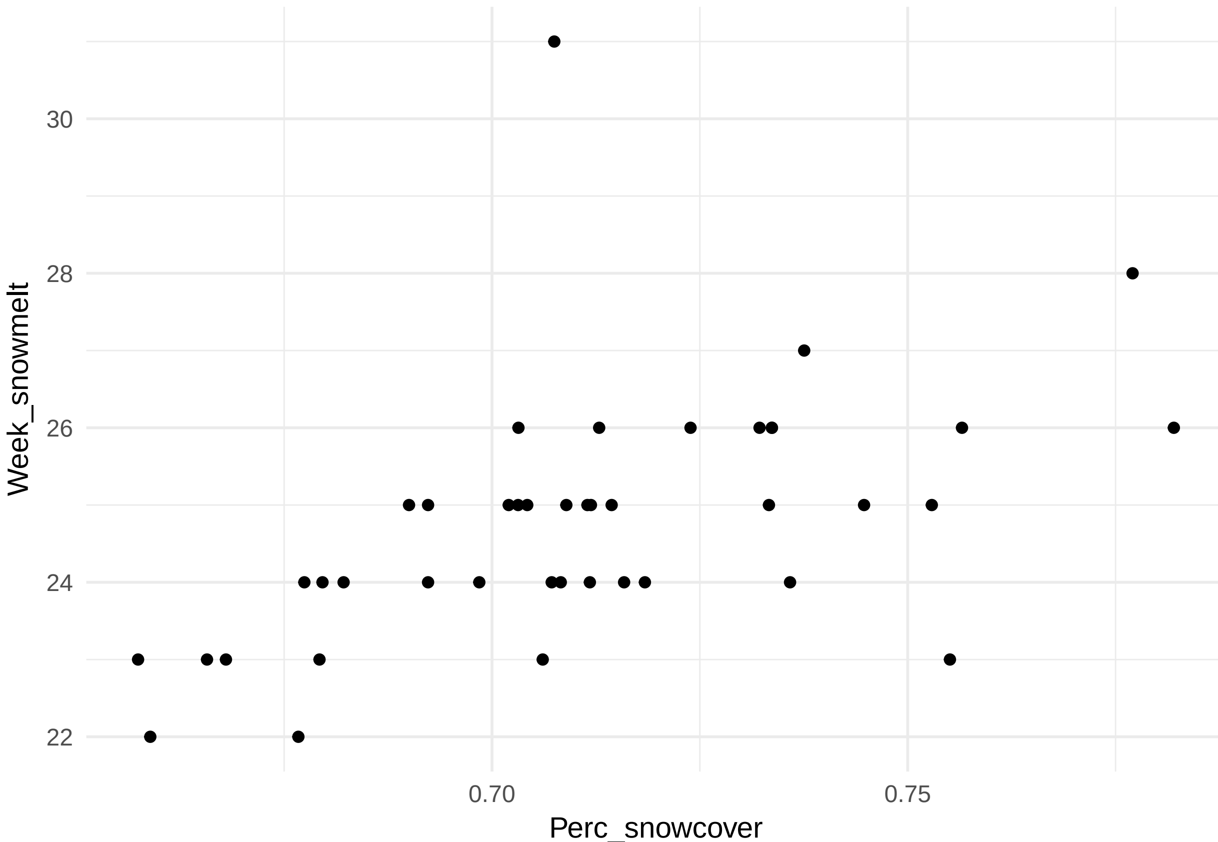

```
# line draw
pl + theme_linedraw()
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



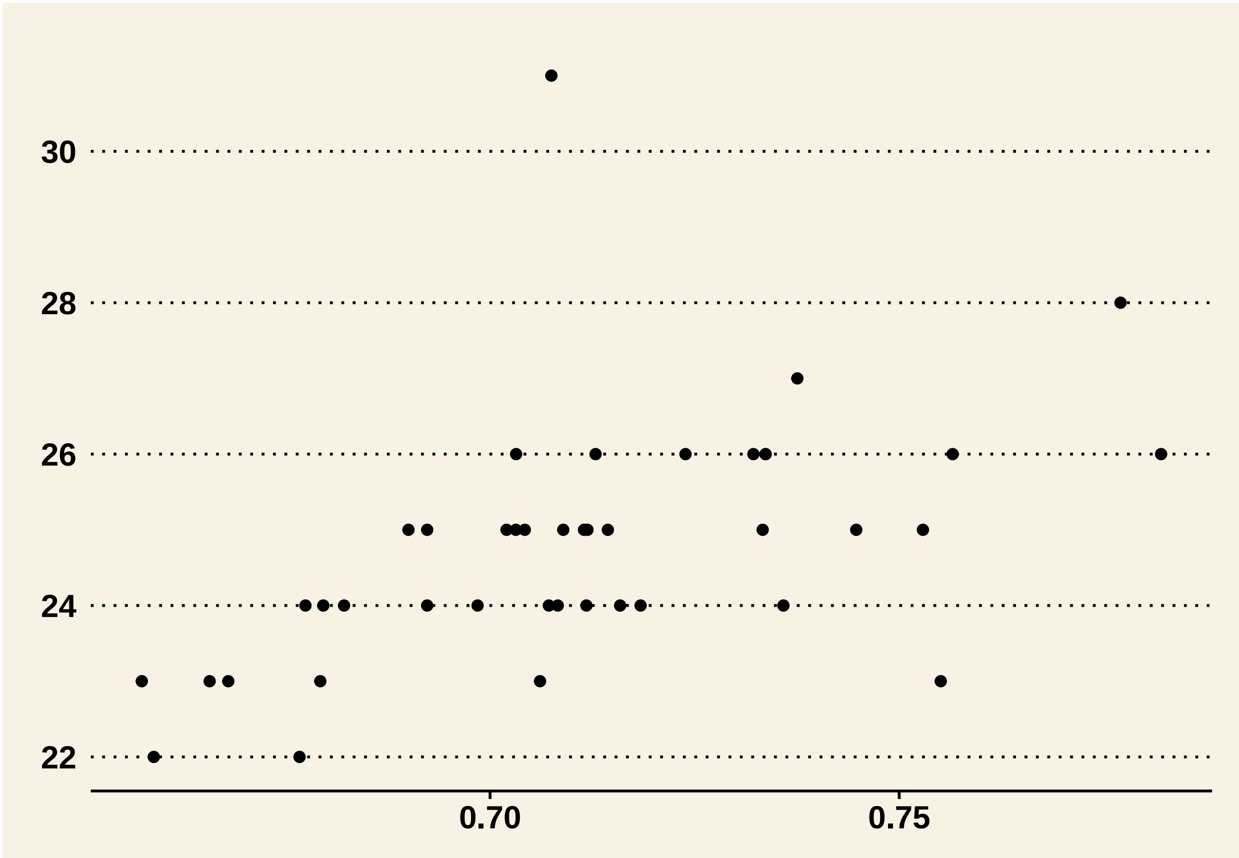

```r
# minimalistic theme
pl + theme_minimal()
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



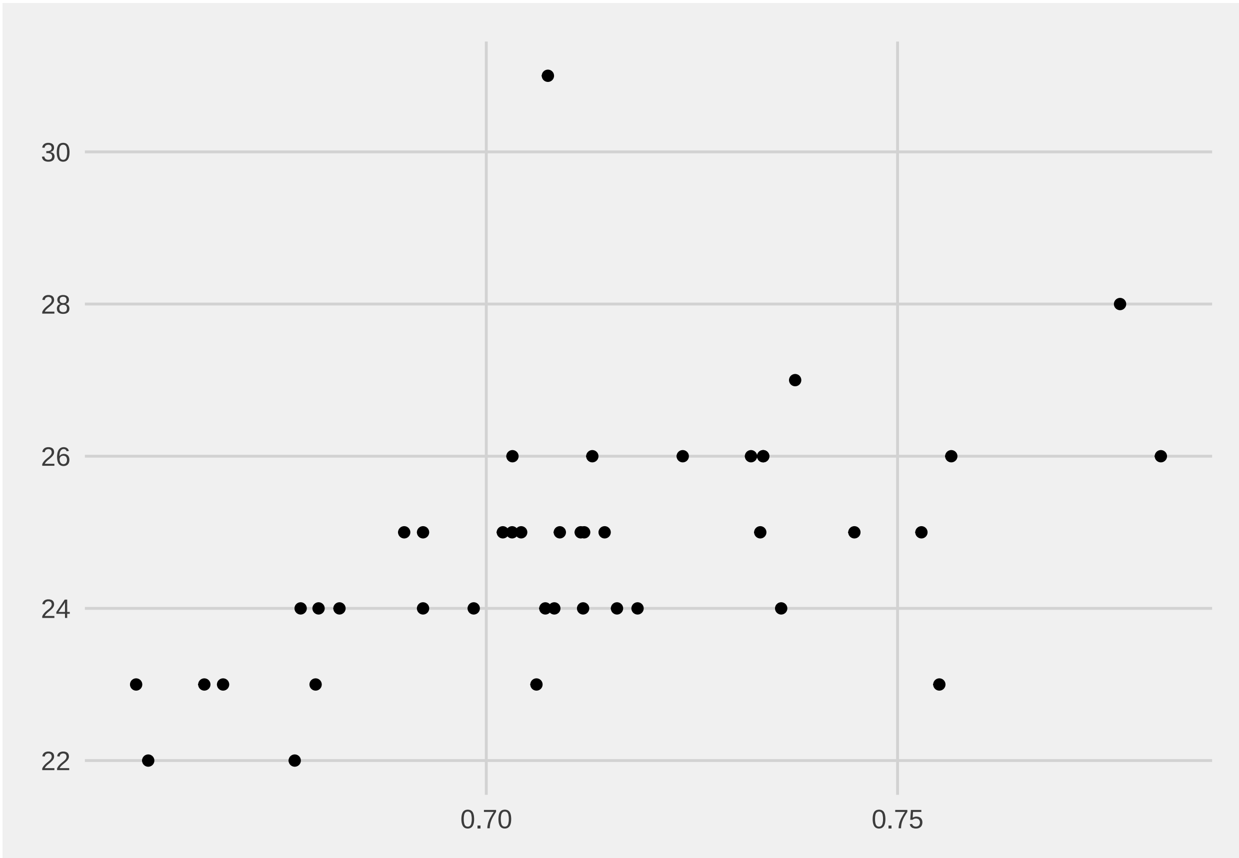

```r
# load additional themes
library(ggthemes)
# Wall Street Journal
show(pl + theme_wsj())
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```

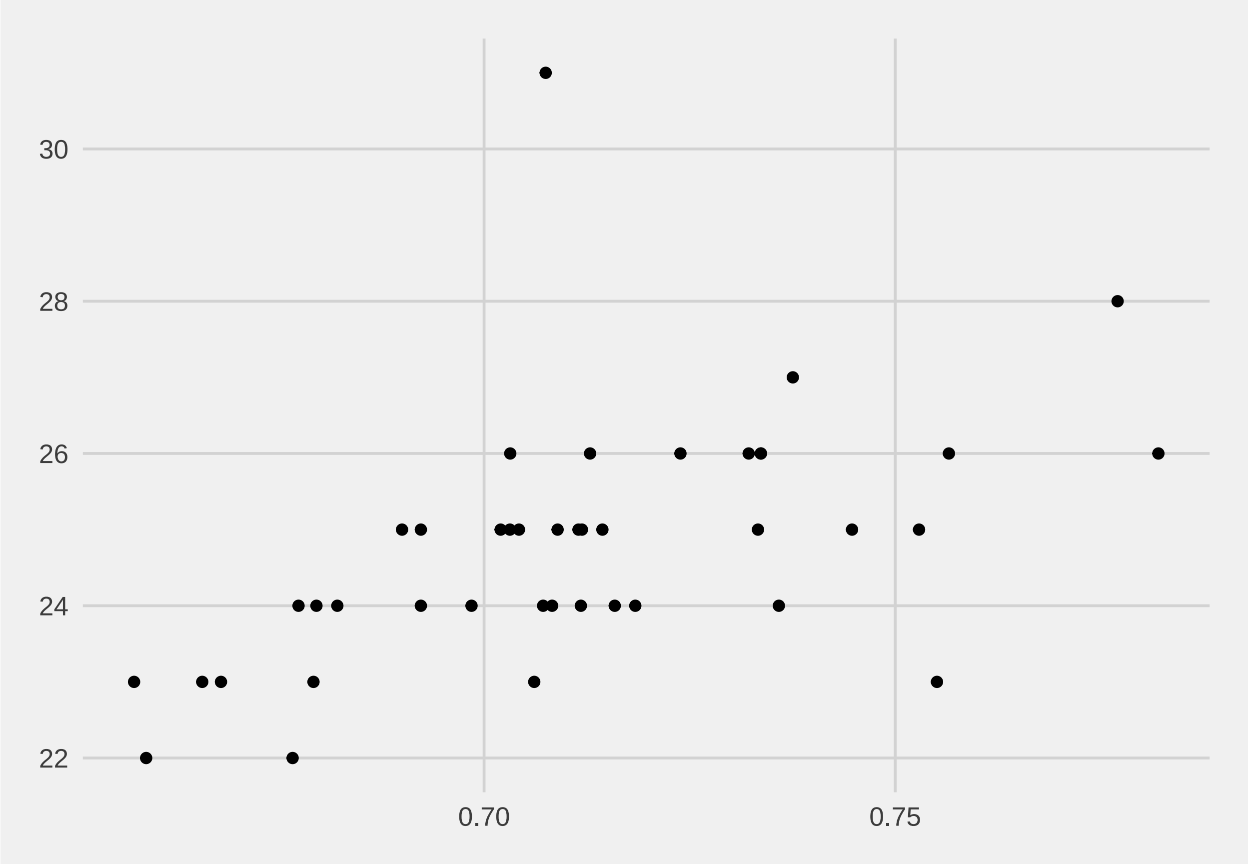



```r
# Five thirty-eight
show(pl + theme_fivethirtyeight())
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```




```
# setting features
# use color as an aesthetic mapping, associated with Herd
pl <- ggplot(data = popsize) + aes(x = Year, y = Pop_Size, colour = Herd) + geom_point()
# set color to be red for all points
pl <- ggplot(data = popsize) + aes(x = Year, y = Pop_Size) + geom_point(colour = "red")



# saving plot as test.pdf in the sandbox
ggsave(filename = "../sandbox/test.pdf", plot = pl, width = 3, height = 4)



# select numerical column headers, or headers with white space using back ticks
popsize %>% filter(Year > 1979, Year < 1985) %>% spread(Year, Pop_Size) %>%  select(Herd, `1980`)
```




```
## # A tibble: 9 x 2
##    Herd `1980`
## * <chr>  <int>
## 1   BAT 140000
## 2   BEV 130000
## 3   CAH     NA
## 4   GRH 390100
## 5   LRH     NA
## 6   PCH     NA
## 7   QAM  40000
## 8   TCH     NA
## 9   WAH 138000
```

```
# ungroup elements
popsize %>% group_by(Herd, Year) %>% tally() %>% ungroup()
```

```
## # A tibble: 114 x 3
##     Herd  Year     n
##    <chr> <int> <int>
##  1   BAT  1977     1
##  2   BAT  1978     1
##  3   BAT  1980     1
##  4   BAT  1982     1
##  5   BAT  1984     1
##  6   BAT  1986     1
##  7   BAT  1990     1
##  8   BAT  1996     1
```

```
##  9   BAT  2003     1
## 10   BAT  2006     1
## # ... with 104 more rows
```

```r
# operation on columns vs rowwise
ndvi %>% mutate(maxndvi = max(NDVI_May, NDVI_June_August)) %>% head(4)
```

```
## # A tibble: 4 x 5
##    Herd  Year NDVI_May NDVI_June_August  maxndvi
##   <chr> <int>    <dbl>            <dbl>    <dbl>
## 1   BAT  1982  0.21440        0.3722679 6.308922
## 2   BAT  1983  0.20448       -0.9977483 6.308922
## 3   BAT  1984  0.24650        1.5864094 6.308922
## 4   BAT  1985  0.24444        0.6420830 6.308922
```

```r
ndvi %>% rowwise() %>% mutate(maxndvi = max(NDVI_May, NDVI_June_August)) %>% head(4)
```

```
## # A tibble: 4 x 5
##    Herd  Year NDVI_May NDVI_June_August   maxndvi
##   <chr> <int>    <dbl>            <dbl>     <dbl>
## 1   BAT  1982  0.21440        0.3722679 0.3722679
## 2   BAT  1983  0.20448       -0.9977483 0.2044800
## 3   BAT  1984  0.24650        1.5864094 1.5864094
## 4   BAT  1985  0.24444        0.6420830 0.6420830
```