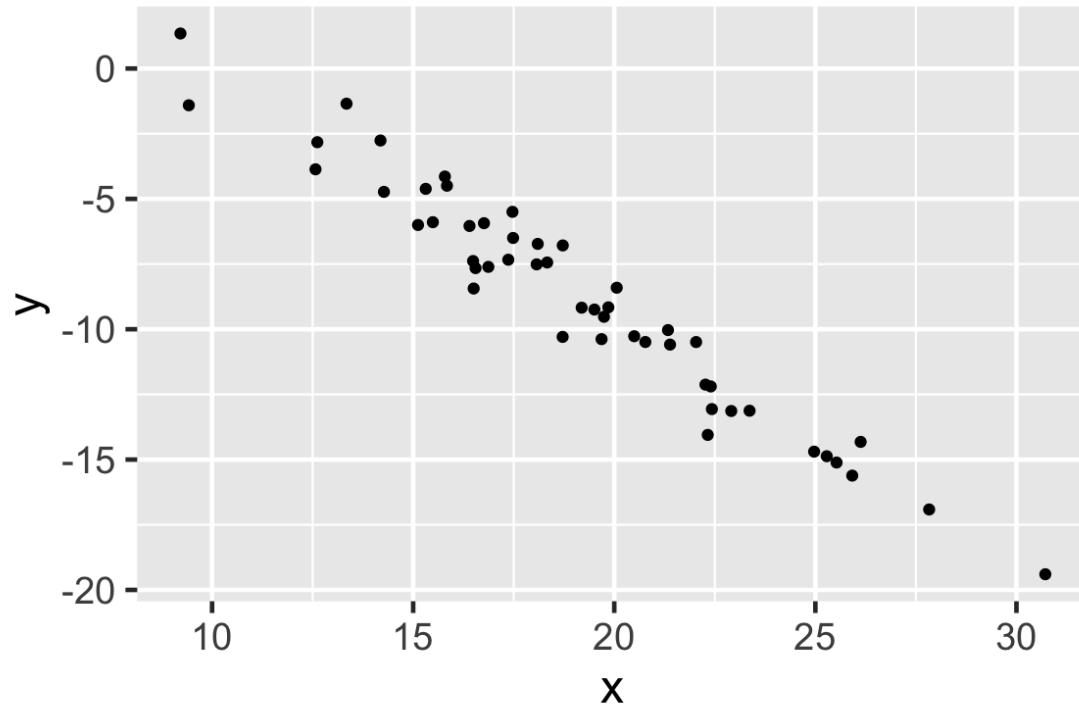


# Chapter 6: Multivariate Continuous Data

Joyce Robbins

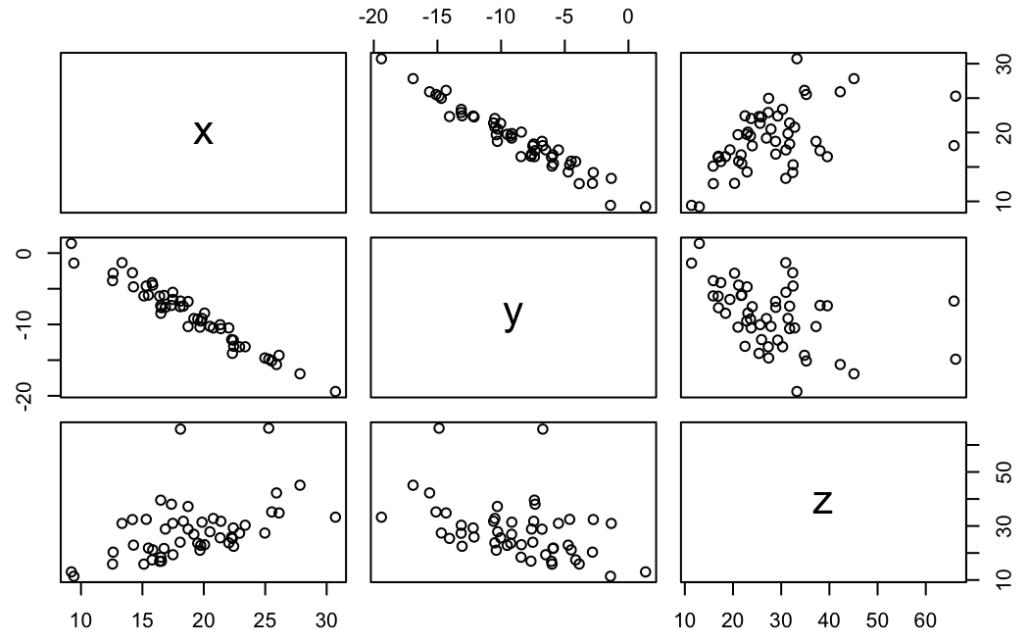
# Two continuous variables: scatterplot

```
library(tidyverse)
theme_set(theme_grey(18))
x <- rnorm(50, 20, 5)
y <- runif(50, 8, 12) - x
df <- data.frame(x, y)
gscat <- ggplot(df, aes(x, y)) + geom_point()
gscat
```



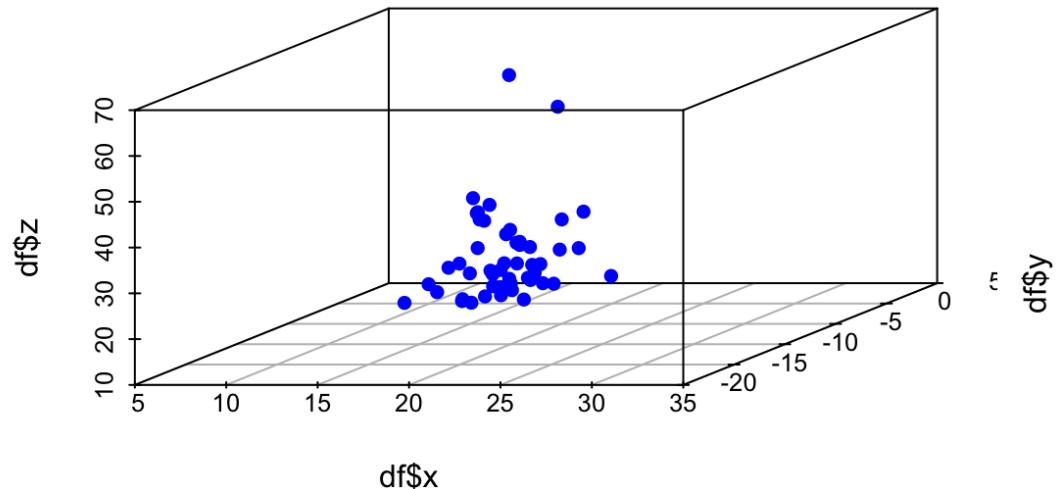
# Three continuous variables: scatterplot matrix

```
df <- df %>% mutate(z = rexp(50, .1) + x)
plot(df)
```



# Three continuous variables: 3D scatterplot

```
library(scatterplot3d)
scatterplot3d(df$x, df$y, df$z, pch = 16, color = "blue")
```

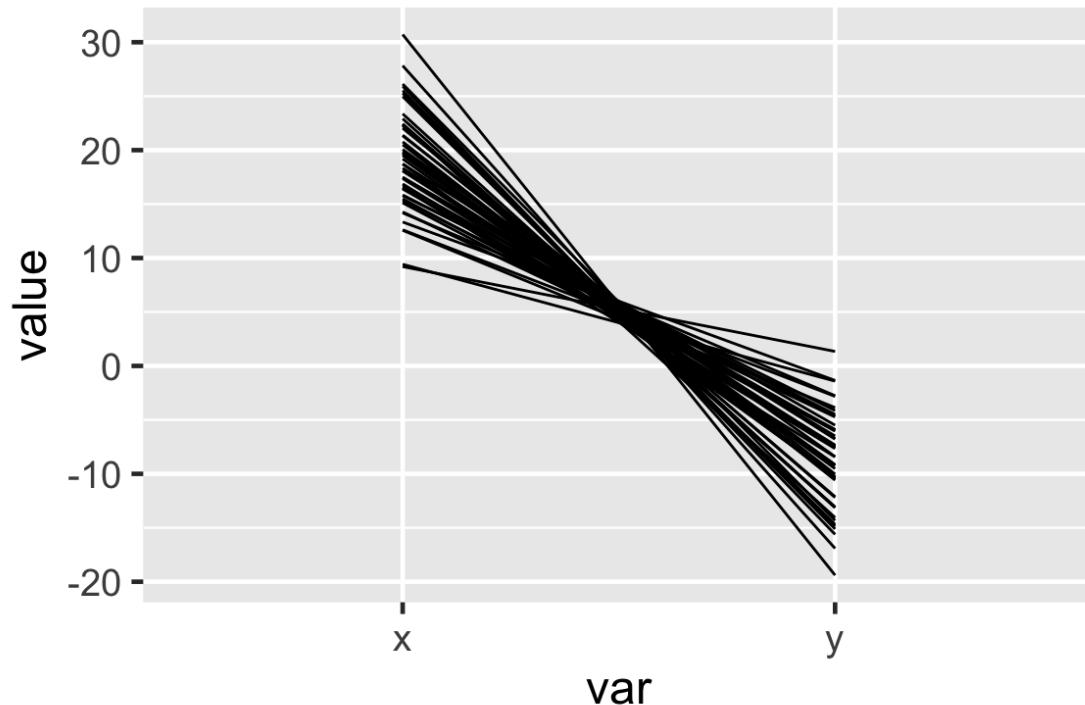


# Three continuous variables: interactive 3D scatterplot

```
library(plotly)
plot_ly(df, x = ~x, y = ~y, z = ~z, mode = "markers",
        marker = list(size = 4)) %>% add_markers()
```

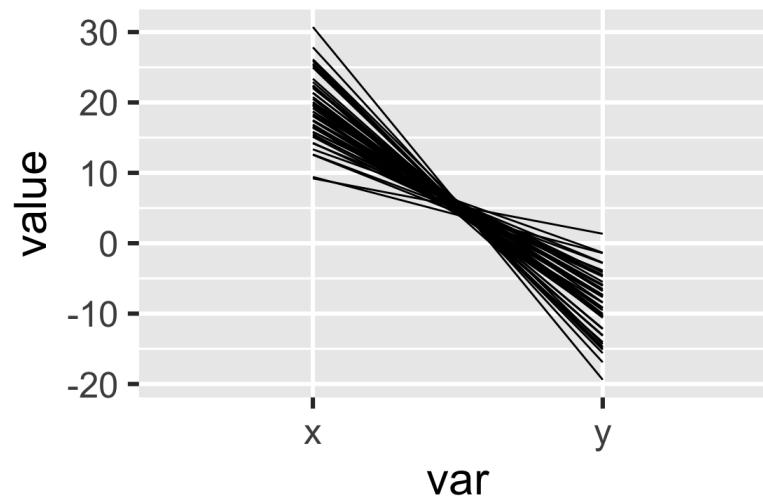
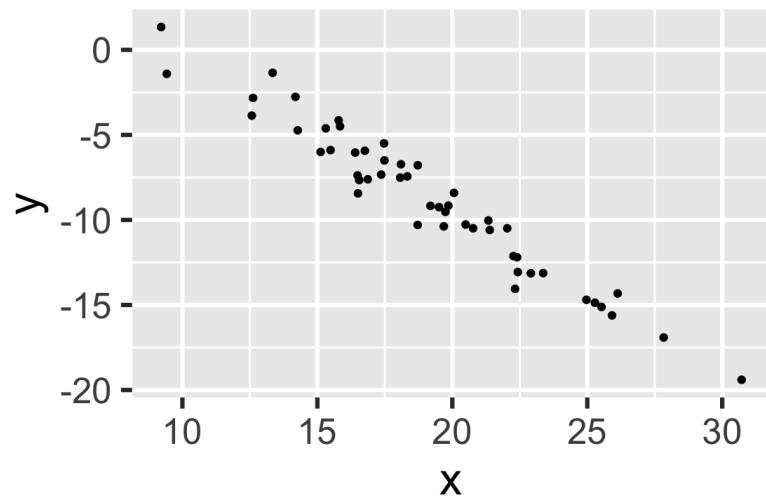
# Slope graph

```
tidydf <- df %>% select(x, y) %>% rownames_to_column("ID") %>%
  gather(var, value, -ID)
gslope <- ggplot(tidydf, aes(x = var, y = value, group = ID)) +
  geom_line()
gslope
```



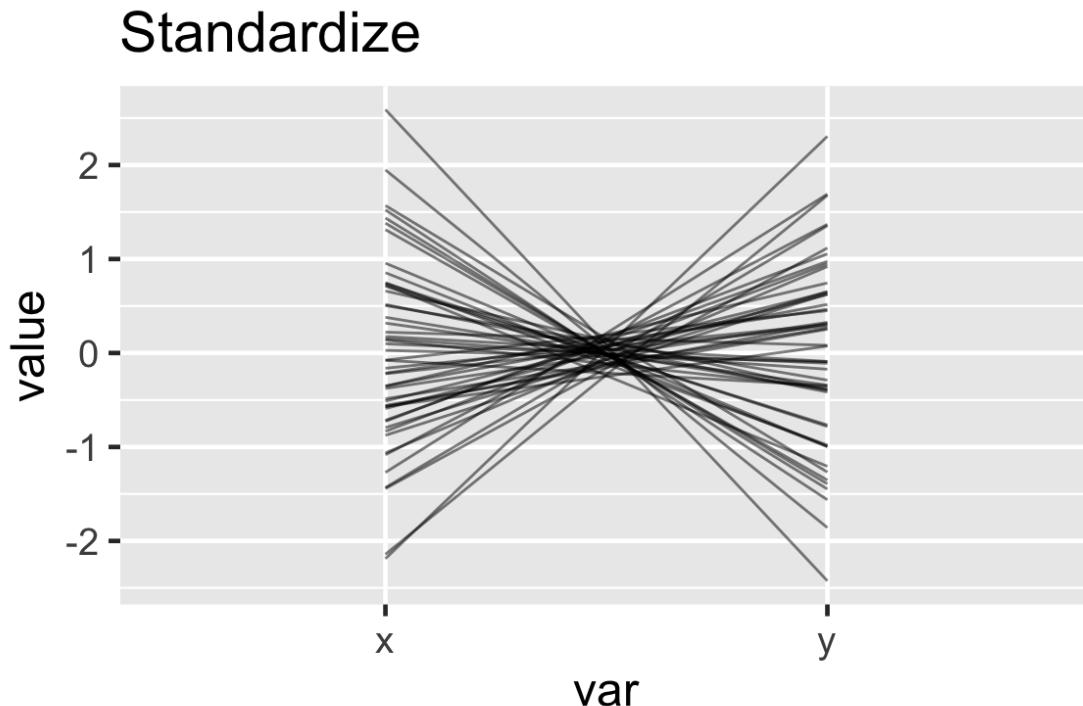
```
geom_line()
```

# Scatterplot vs slope graph



# Standardize data

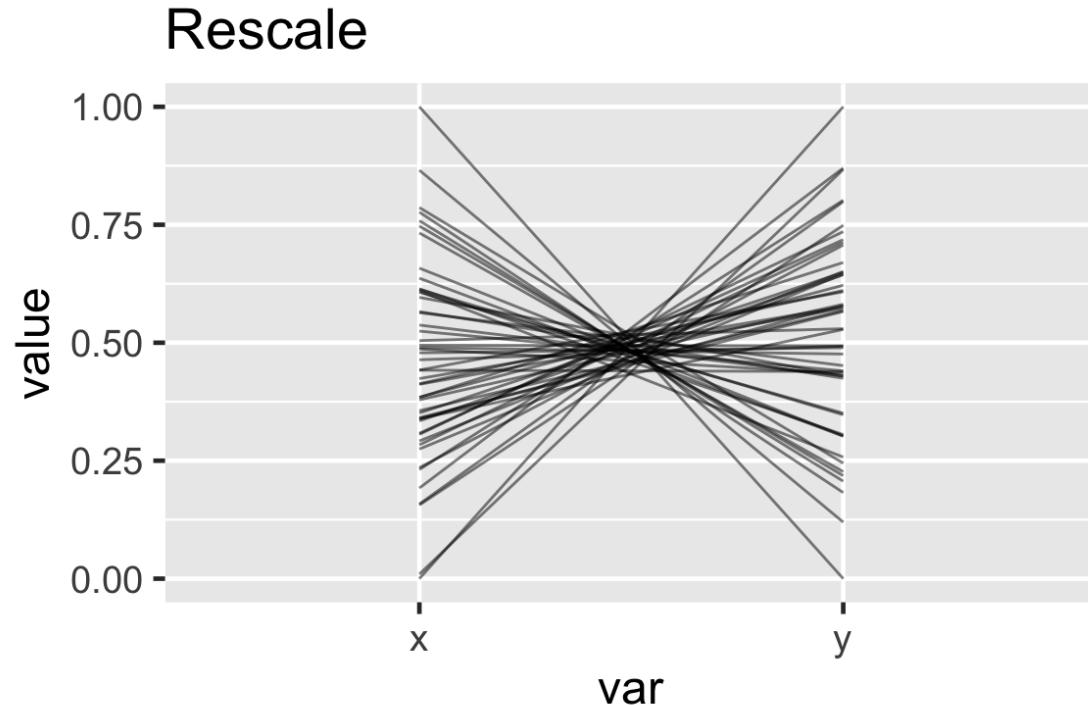
```
standardize <- function(x) (x-mean(x))/sd(x)
df2 <- tidydf %>% group_by(var) %>%
  mutate(value = standardize(value)) %>% ungroup()
g0 <- ggplot(df2, aes(x = var, y = value, group = ID)) +
  geom_line(alpha = .5) + ggttitle("Standardize")
g0
```



```
geom_line()
```

# Rescale data to [0,1]

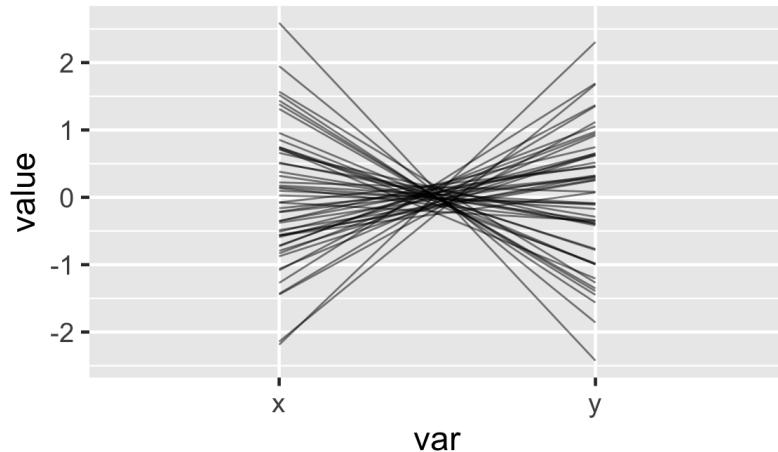
```
df2 <- tidydf %>% group_by(var) %>%
  mutate(value = scales::rescale(value)) %>% ungroup()
g1 <- ggplot(df2, aes(x = var, y = value, group = ID)) +
  geom_line(alpha = .5) + ggtitle("Rescale")
g1
```



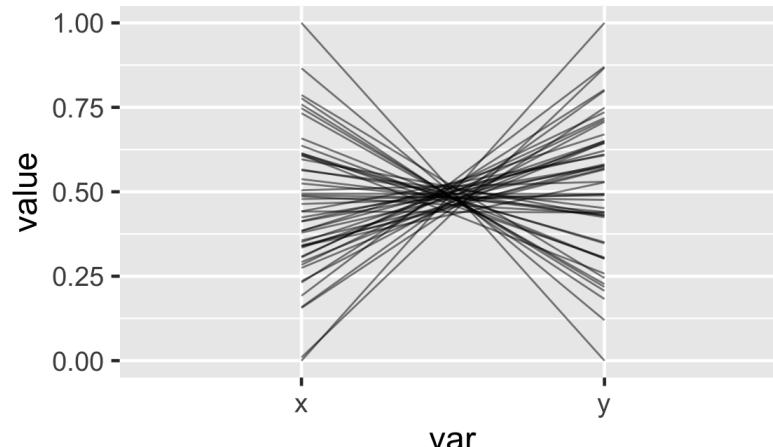
```
geom_line()
```

# Compare

Standardize

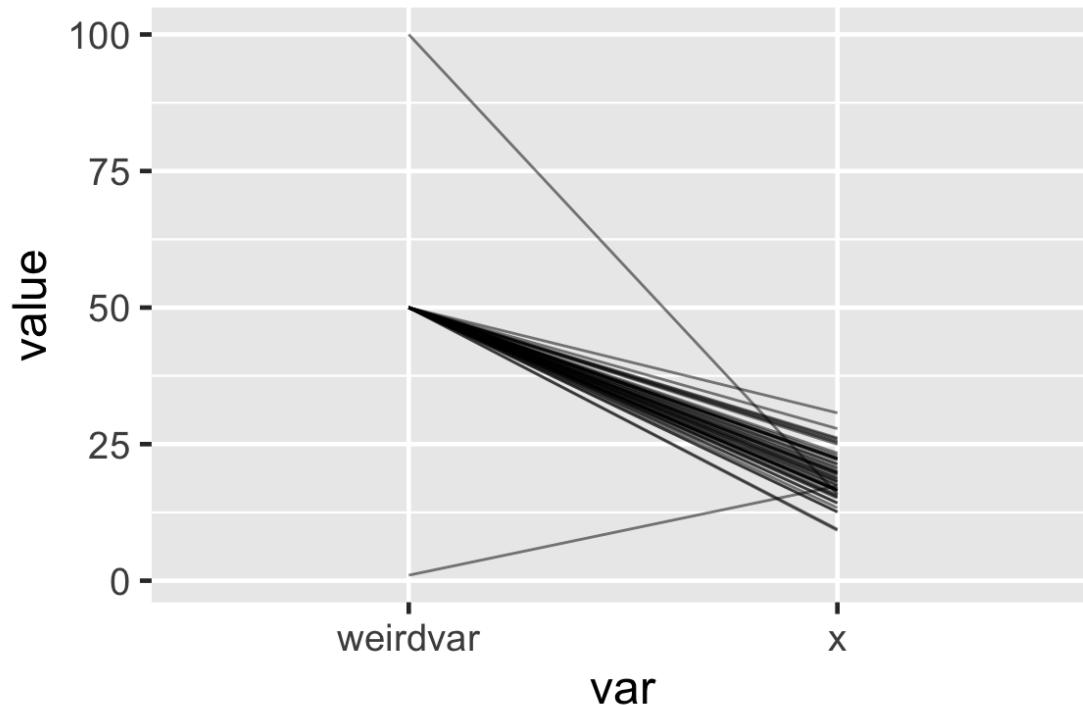


Rescale

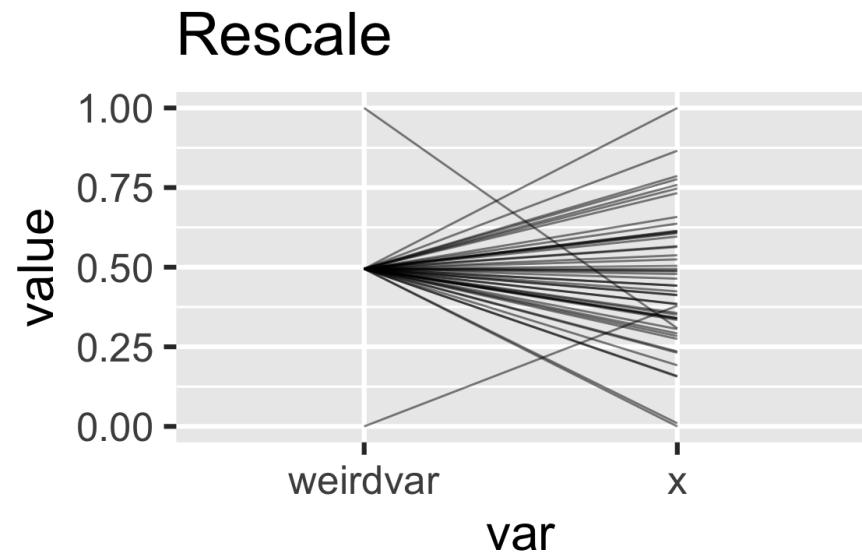
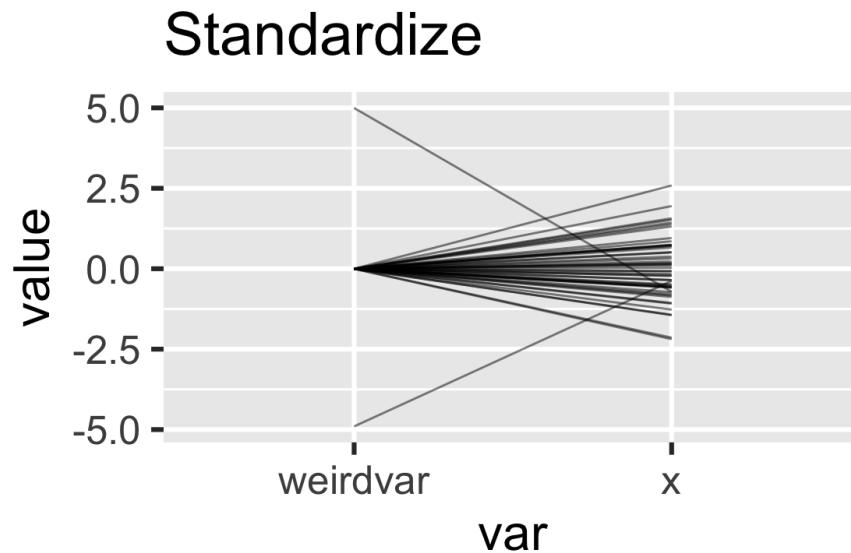


# Compare different distributions

```
weirdvar <- c(1, rep(50, 48), 100)
df <- data.frame(x, weirdvar)
tidydf <- df %>% rownames_to_column("ID") %>% gather(var, value, -ID)
ggplot(tidydf, aes(x = var, y = value, group = ID)) +
  geom_line(alpha = .5)
```

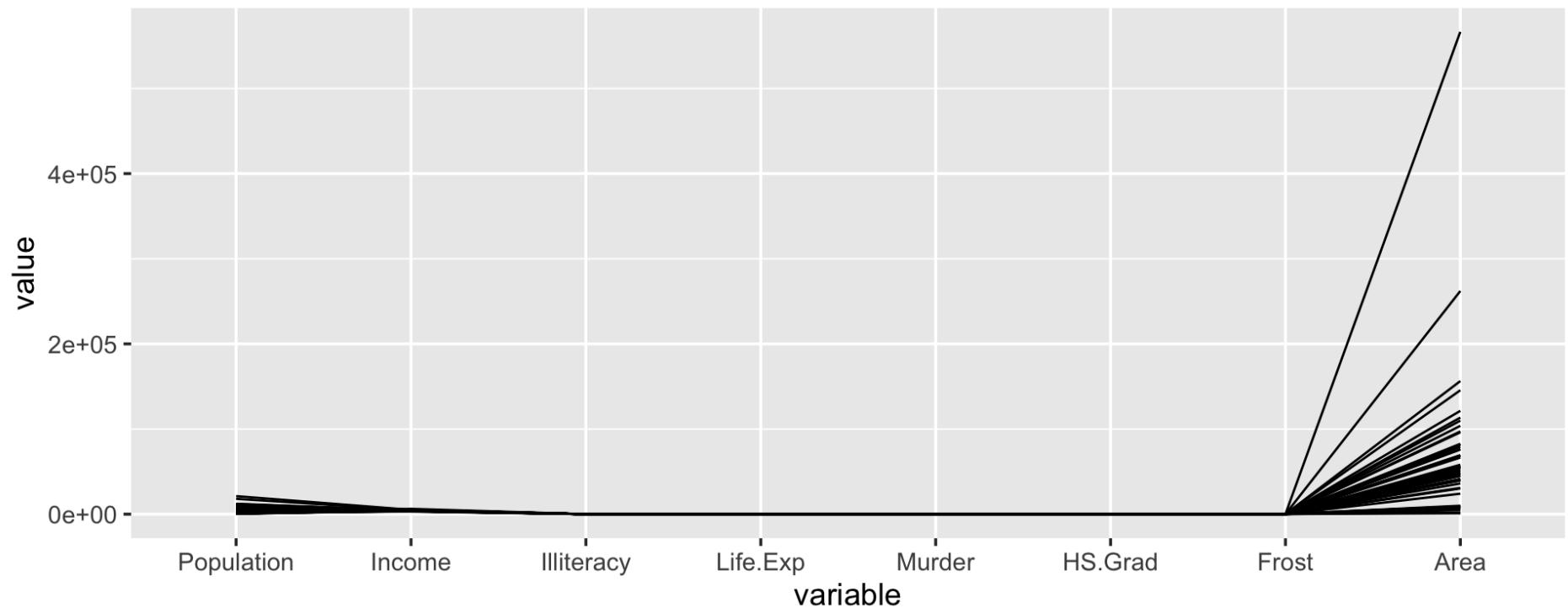


# Compare



# Scale = “globalminmax”

```
library(GGally)
theme_set(theme_grey(14))
# scale = globalminmax
mystates <- data.frame(state.x77) %>%
  rownames_to_column("State") %>%
  mutate(Region = factor(state.region))
# state.region is a separate vector -- see: ?state
mystates$Region <- factor(mystates$Region,
                           levels = c("Northeast", "North Central",
                                     "South", "West"))
ggparcoord(mystates, columns = 2:9, scale = "globalminmax")
```

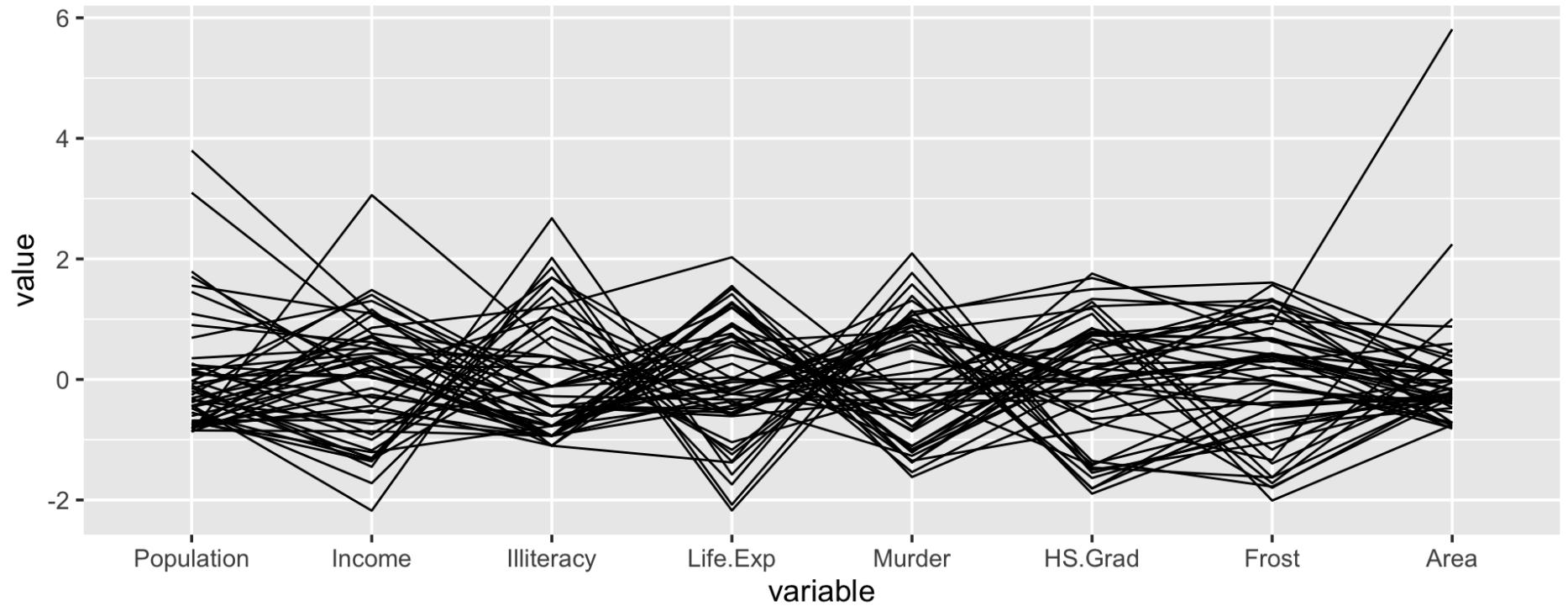


```
GGally:::ggparcoord()
```



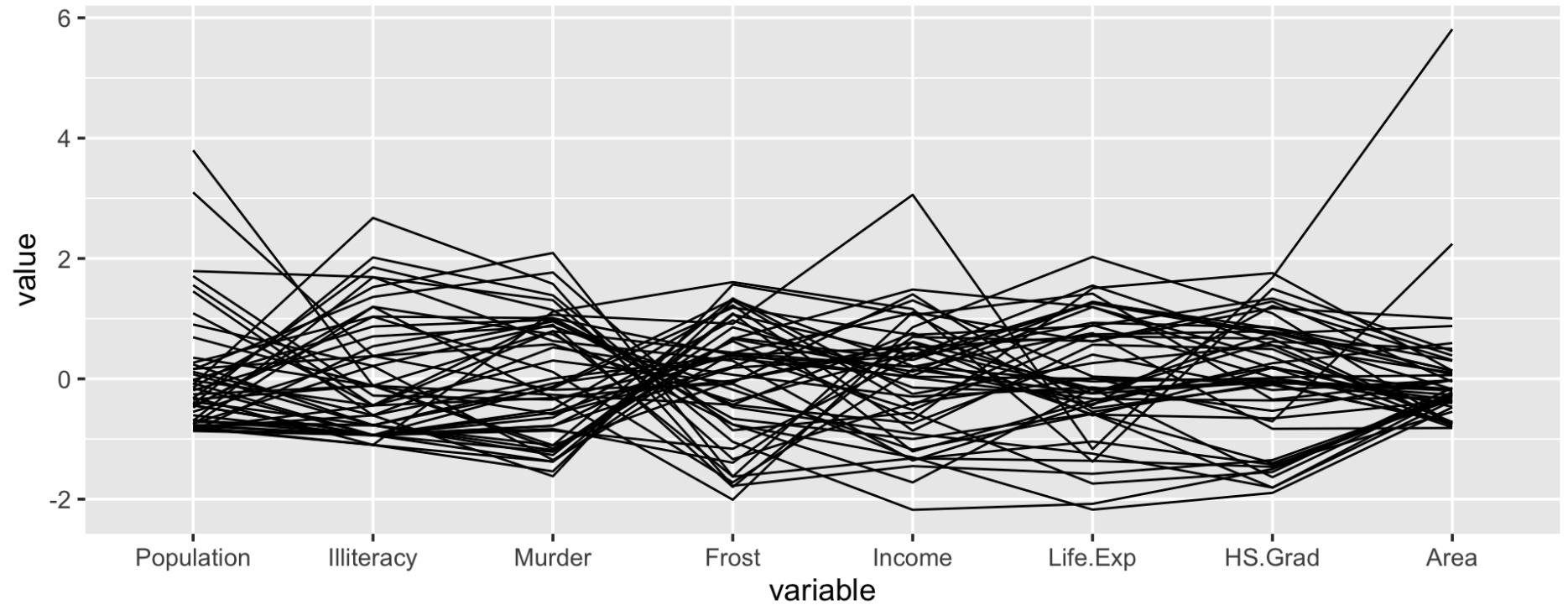
# Scale = “std” (default)

```
# scale = std (default)
ggparcoord(mystates, columns = 2:9)
```



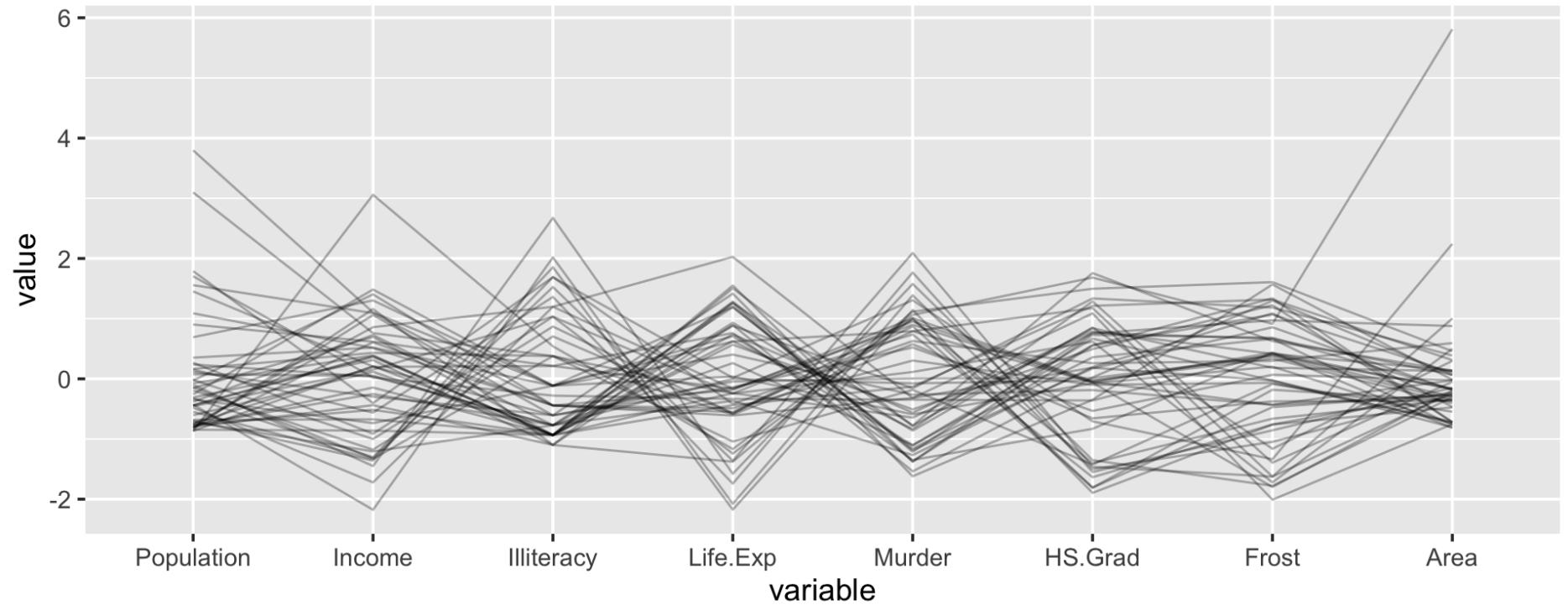
# Reordered

```
# scale = std (default)
ggparcoord(mystates, columns = c(2, 4, 6, 8, 3, 5, 7, 9))
```



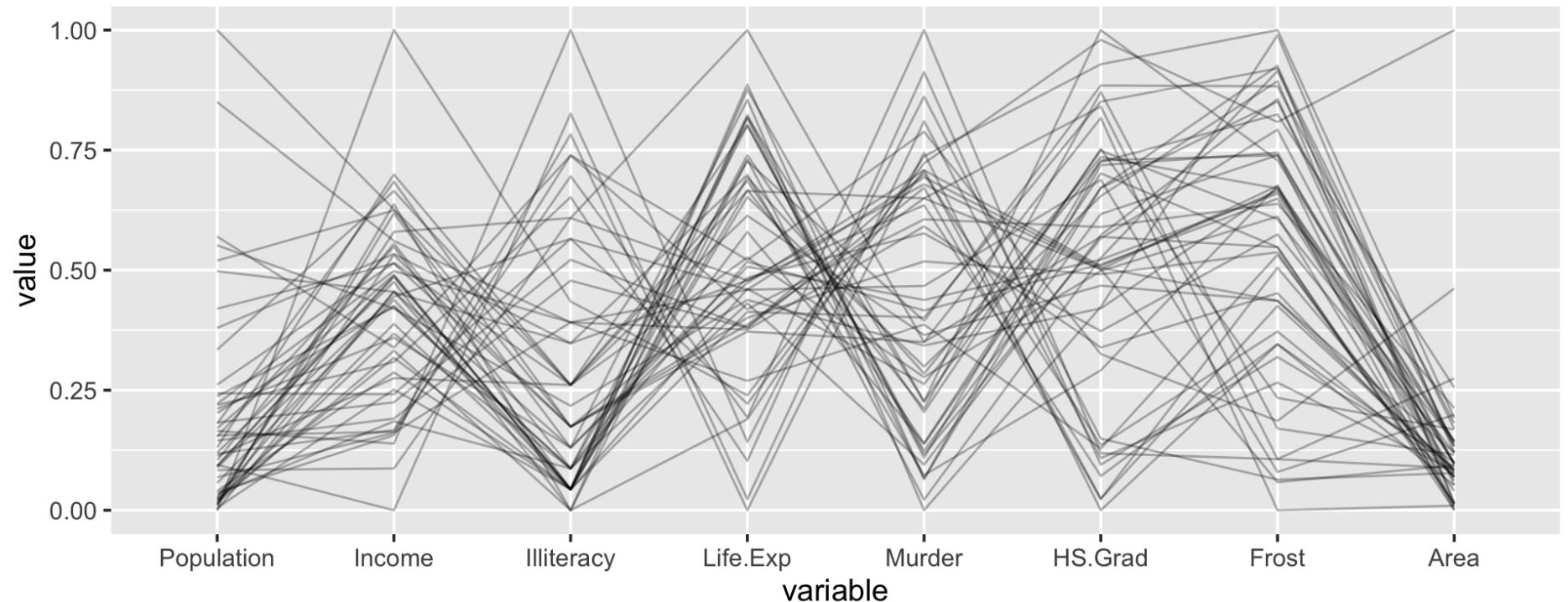
# Alpha

```
# scale = std (default)
ggparcoord(mystates, columns = 2:9, alphaLines = .3)
```



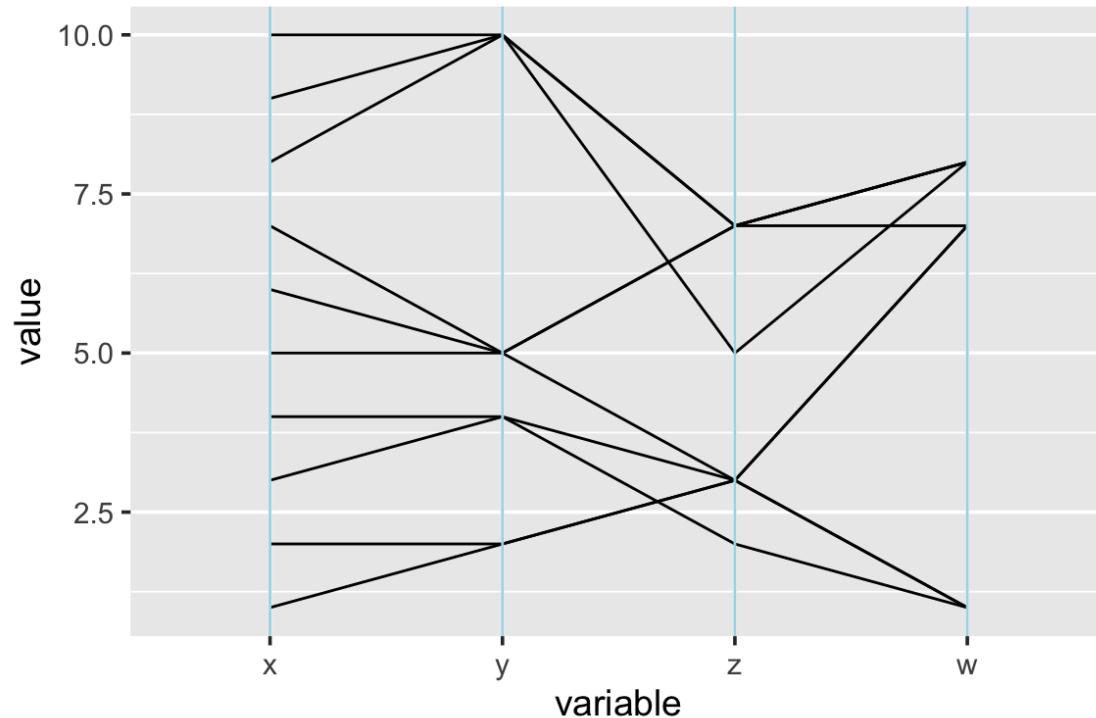
# Alpha + rescale

```
# scale = std (default)
ggparcoord(mystates, columns = 2:9, alphaLines = .3,
           scale = "uniminmax")
```



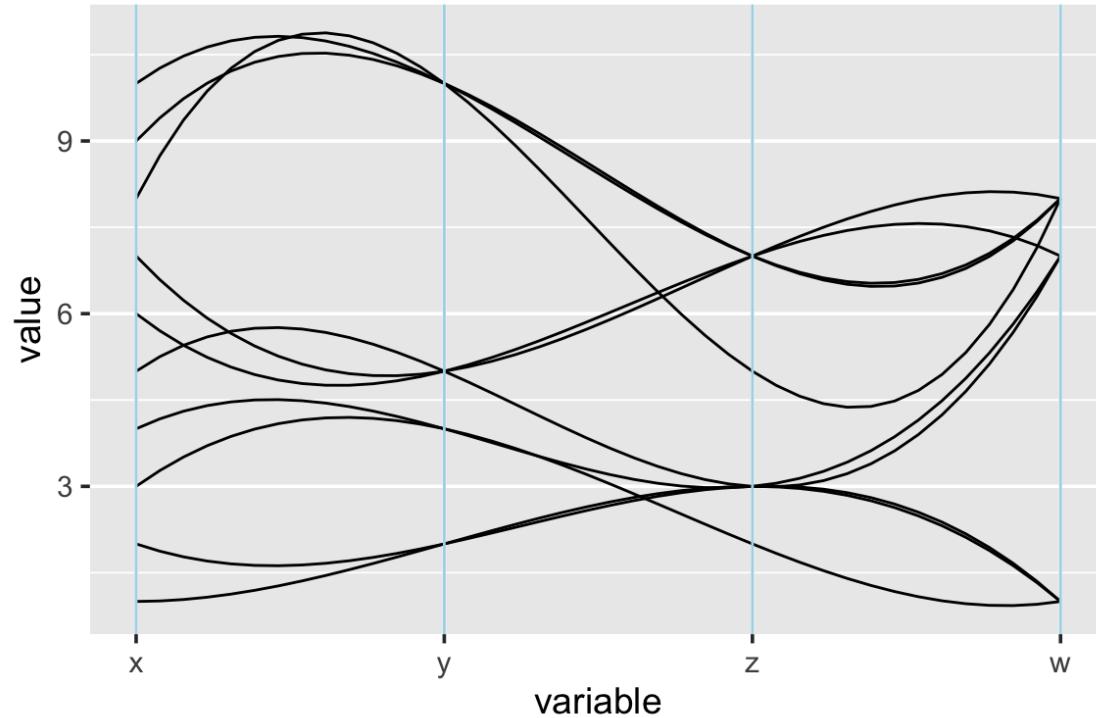
# Dataset with repeats

```
x <- 1:10
y <- c(2,2,4,4,5,5,5,10,10,10)
z <- c(3,3,2,3,3,7,7,5,7,7)
w <- c(1, 1, 1, 7, 7, 7, 8, 8, 8, 8)
df <- data.frame(x,y,z, w)
g0 <- ggparcoord(df, columns = 1:4, scale = "globalminmax") + geom_vline(xintercept = 1:4, color = "lightblue")
g0
```

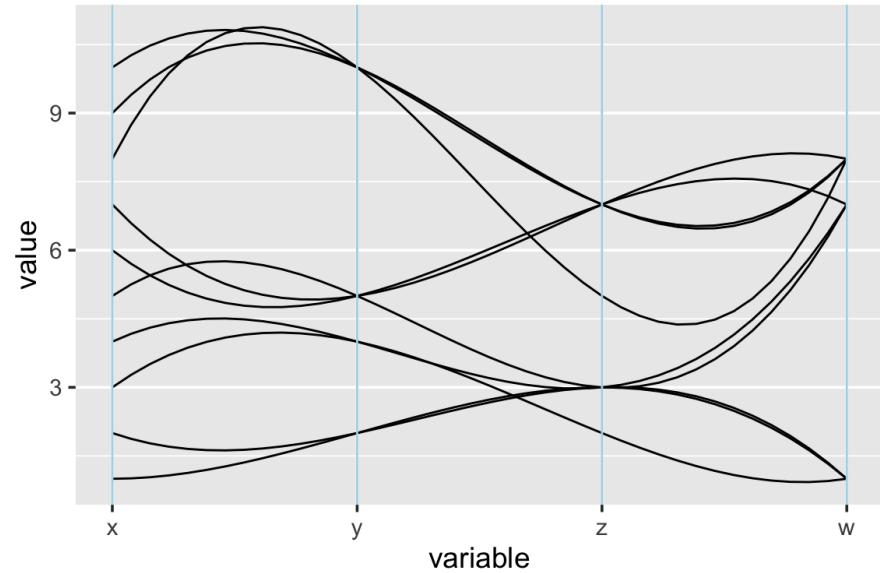
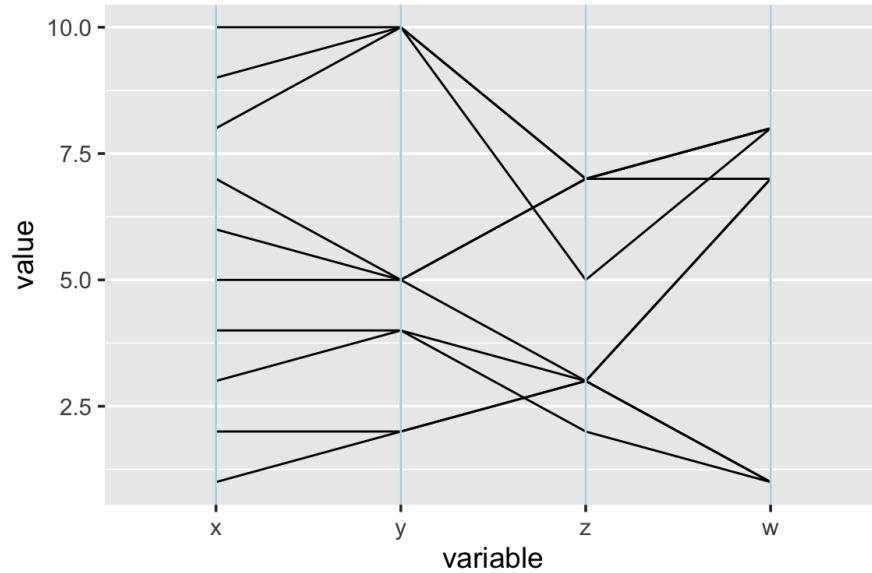


# Splines

```
g1 <- ggparcoord(df, columns = 1:4, scale = "globalminmax",
                  splineFactor = 10) +
  geom_vline(xintercept = 1:4, color = "lightblue")
g1
```

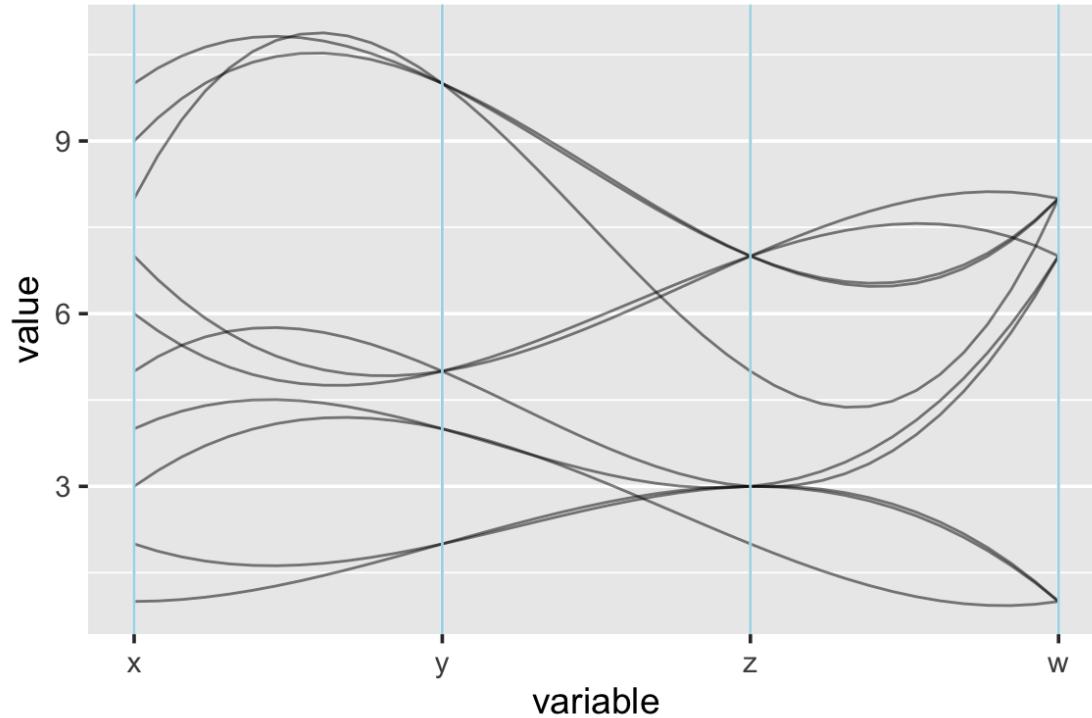


# Compare



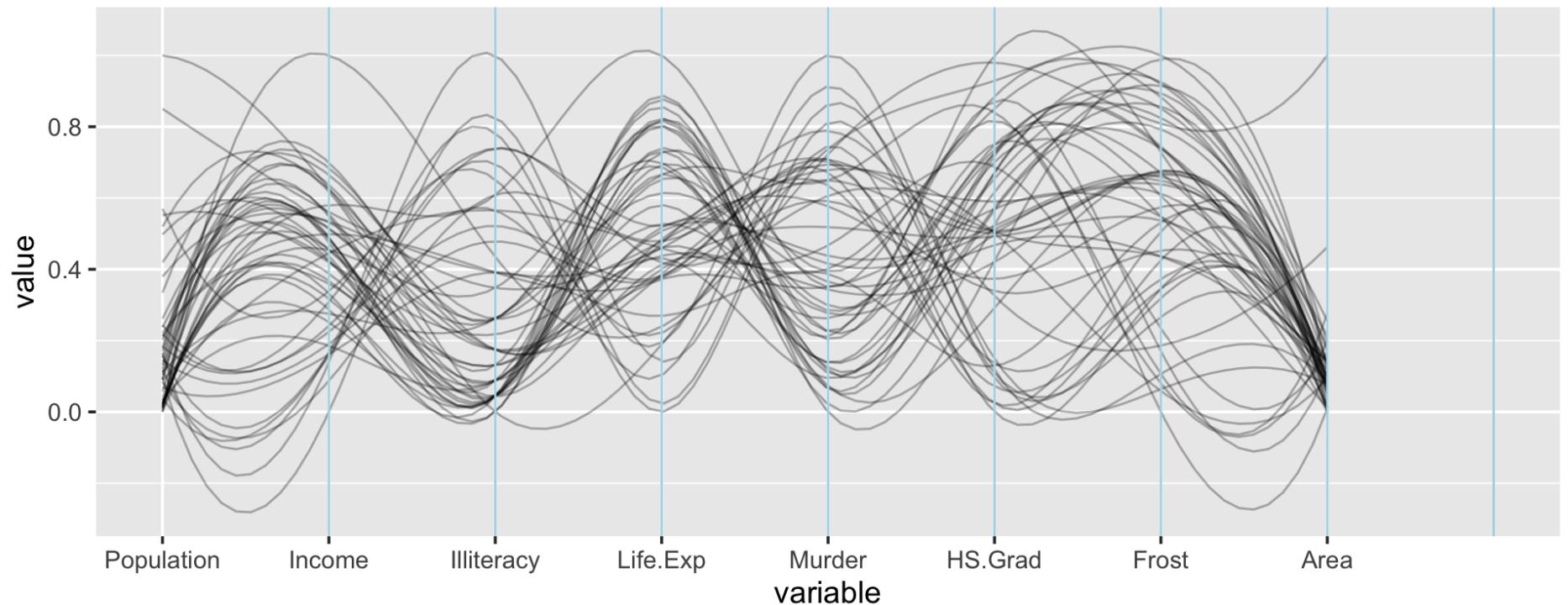
# Alpha

```
g1 <- ggparcoord(df, columns = 1:4, scale = "globalminmax",
                  splineFactor = 10, alphaLines = .5) +
  geom_vline(xintercept = 1:4, color = "lightblue")
g1
```



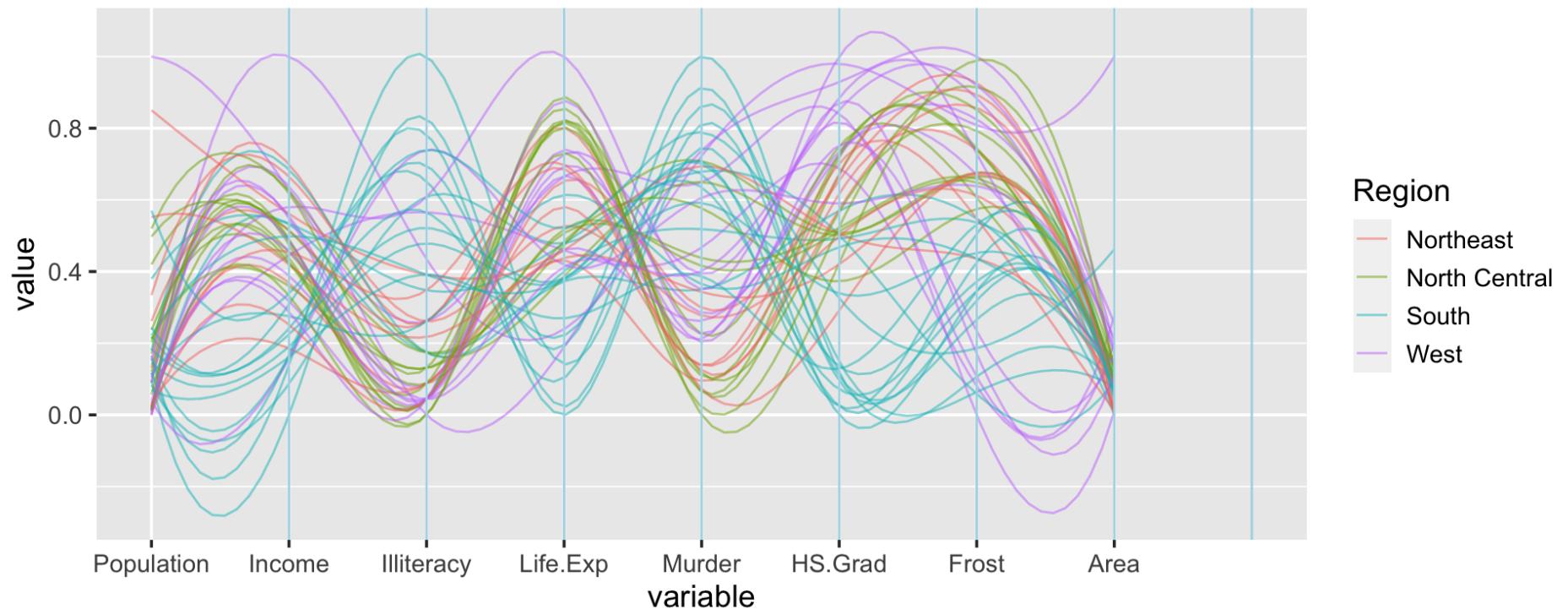
# Alpha + rescale + splines

```
# scale = std (default)
ggparcoord(mystates, columns = 2:9, alphaLines = .3,
            scale = "uniminmax", splineFactor = 10) +
  geom_vline(xintercept = 2:9, color = "lightblue")
```



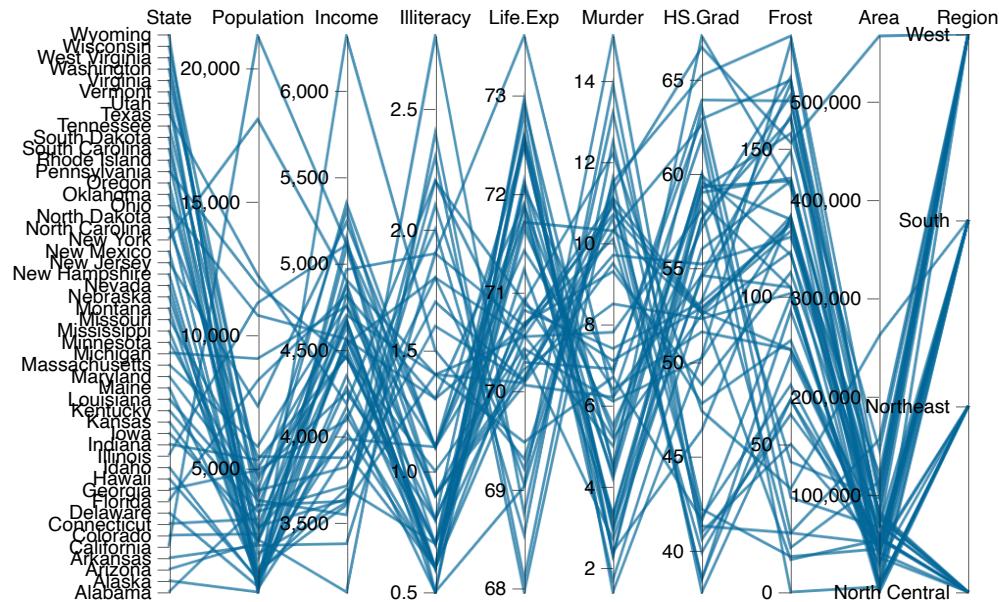
# Alpha + rescale + splines + group

```
# scale = std (default)
ggparcoord(mystates, columns = 2:9, alphaLines = .5,
            scale = "uniminmax", splineFactor = 10, groupColumn = 10) +
  geom_vline(xintercept = 2:9, color = "lightblue")
```



# Html widget: parcoords

```
# See: http://www.buildingwidgets.com/blog/2015/1/30/week-04-interactive-parallel-coordinates-1
# devtools::install_github("timelyportfolio/parcoords")
library(parcoords)
mystates %>% arrange(Region) %>%
  parcoords(
    rownames = F
    , brushMode = "1D-axes"
    , reorderable = T
    , queue = T
    )
```

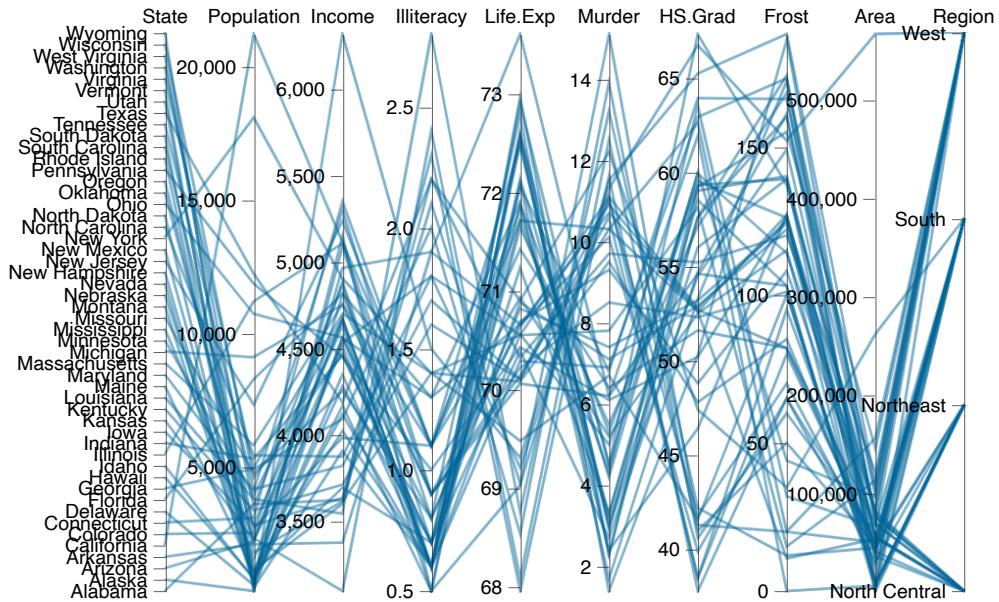


```
parcoords::parcoords()
```

<http://www.buildingwidgets.com/blog/2015/1/30/week-04-interactive-parallel-coordinates-1>

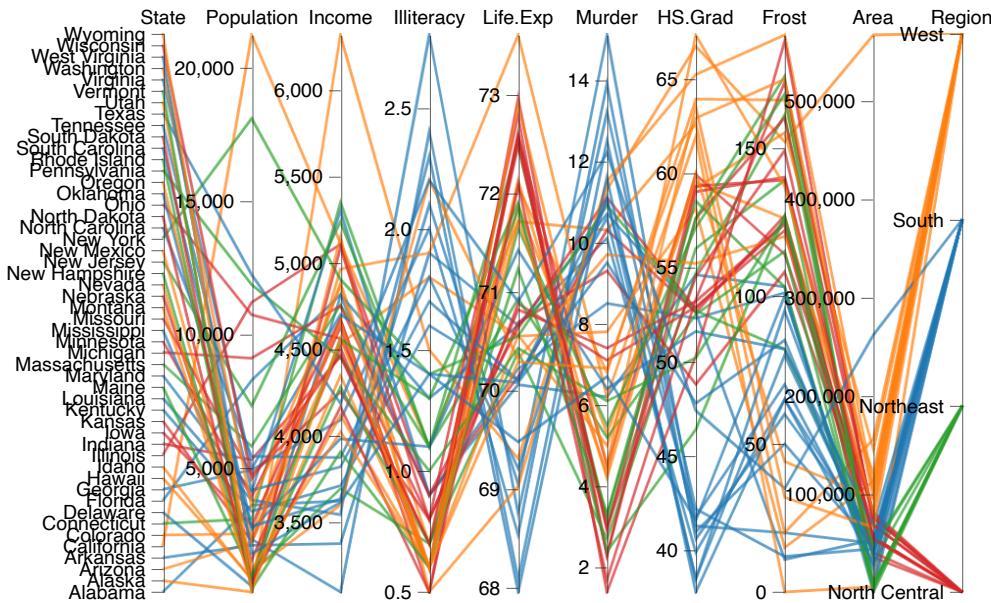
# Add alpha blending

```
# with alpha
mystates %>% arrange(Region) %>%
  parcoords(
    rownames = F
    , brushMode = "1D-axes"
    , reorderable = T
    , queue = T
    , alpha = .5
  )
```



# Color

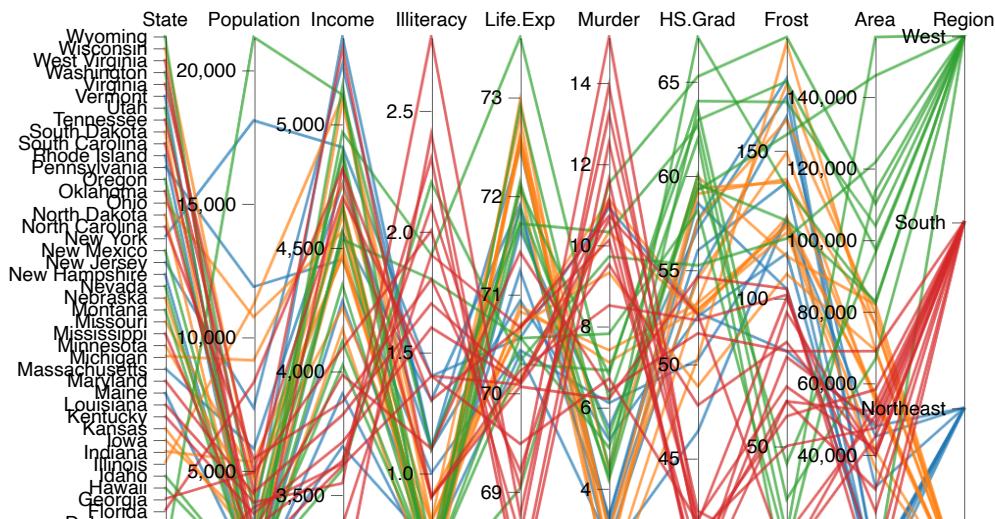
```
parcoords(mystates  
  , rownames = F  
  , brushMode = "1D-axes"  
  , reorderable = T  
  , queue = T  
  , color = list(  
    colorBy = "Region"  
    ,colorScale = "scaleOrdinal"  
    ,colorScheme = "schemeCategory10"  
  )  
  , withD3 = TRUE  
)
```

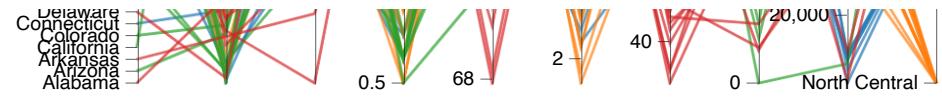




# Filter out Alaska / Texas

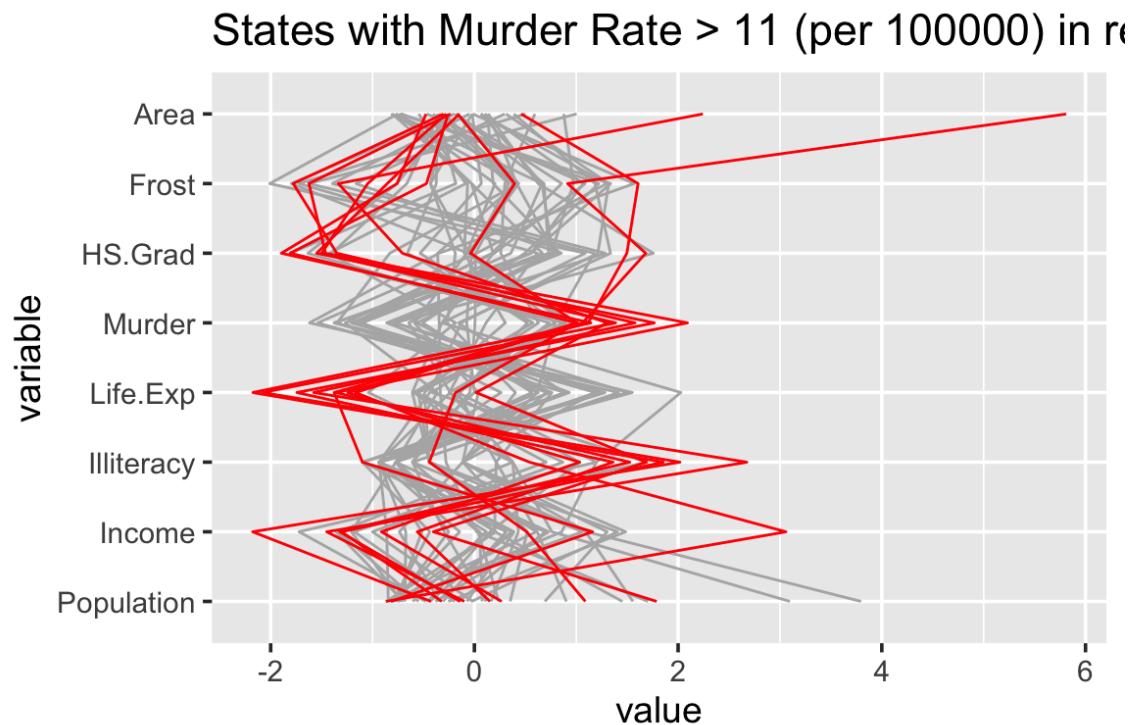
```
# filter out Alaska, Texas, group by Region
myregionorder <- mystates %>% group_by(Region) %>%
  summarize(n = n()) %>% arrange(n)
mystates$Region <- factor(mystates$Region,
  levels = myregionorder$Region)
data.frame(mystates) %>% filter(State != "Alaska") %>%
  filter(State != "Texas") %>% arrange(Region) %>%
  parcoords(
    rownames = F
    , brushMode = "1D-axes"
    , reorderable = T
    , queue = T
    , color = list(
      colorBy = "Region"
      ,colorScale = "scaleOrdinal"
      ,colorScheme = "schemeCategory10"
      )
    , withD3 = TRUE
  )
```





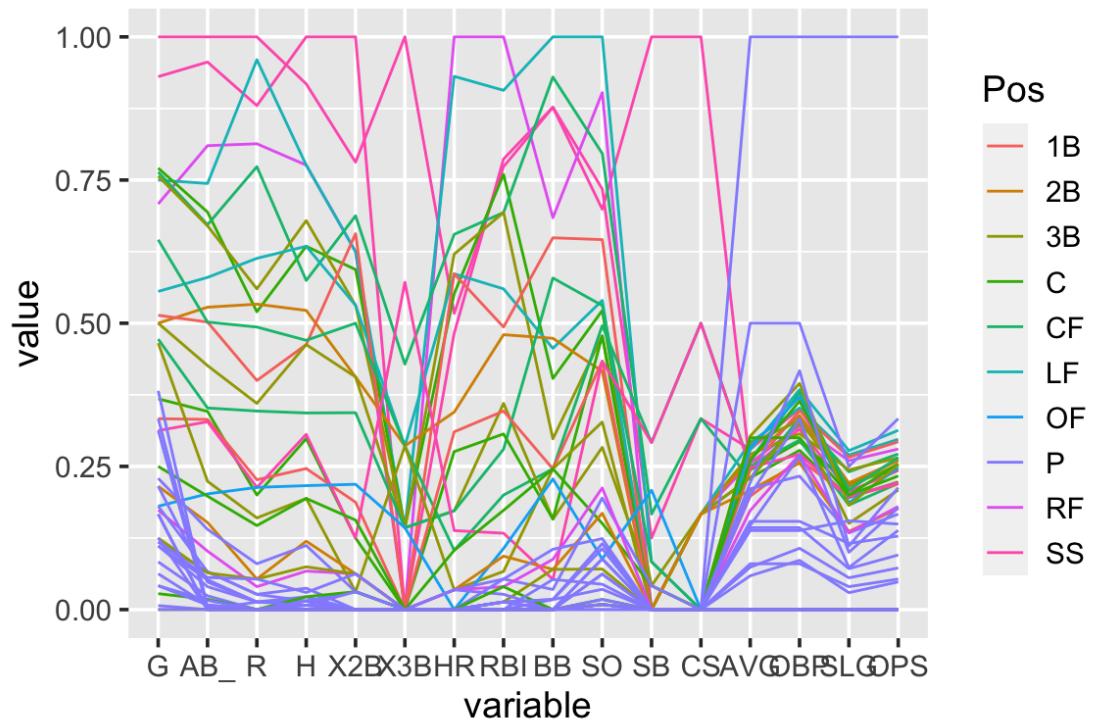
# Highlighting a trend

```
mystates <- mystates %>%
  mutate(color = factor(ifelse(Murder > 11, 1, 0))) %>%
  arrange(color)
ggparcoord(mystates, columns = 2:9, groupColumn = "color") +
  scale_color_manual(values = c("grey70", "red")) +
  coord_flip() + guides(color = FALSE) +
  ggtitle("States with Murder Rate > 11 (per 100000) in red")
```



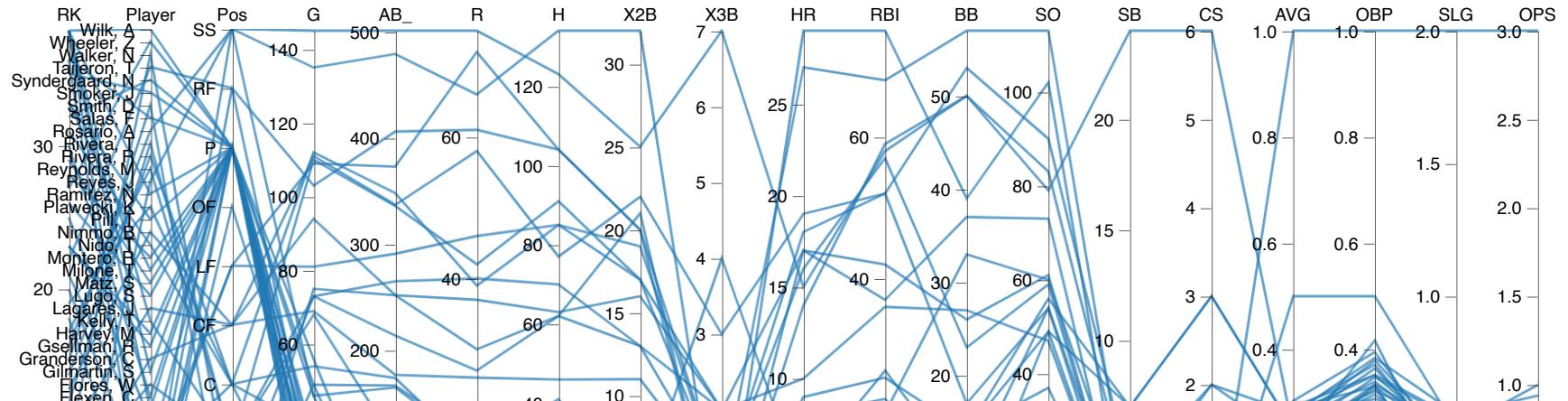
# 2017 Mets

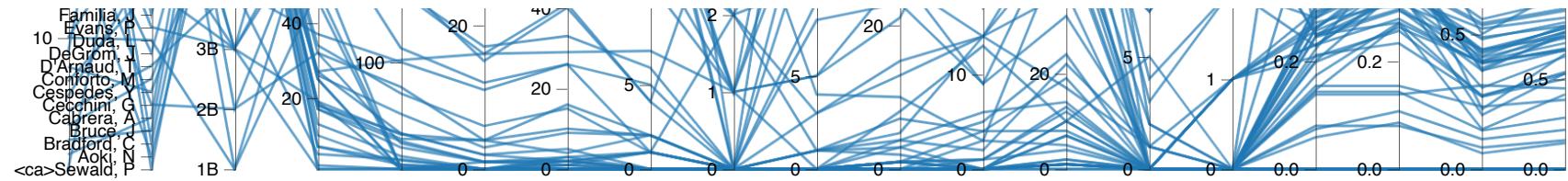
```
# Data from http://mlb.mlb.com/stats/
baseball <- read.csv("MetsStats2017.csv")
ggparcoord(baseball, columns = 5:20, groupColumn = "Pos",
           scale = "uniminmax")
```



# 2017 Mets (Interactive)

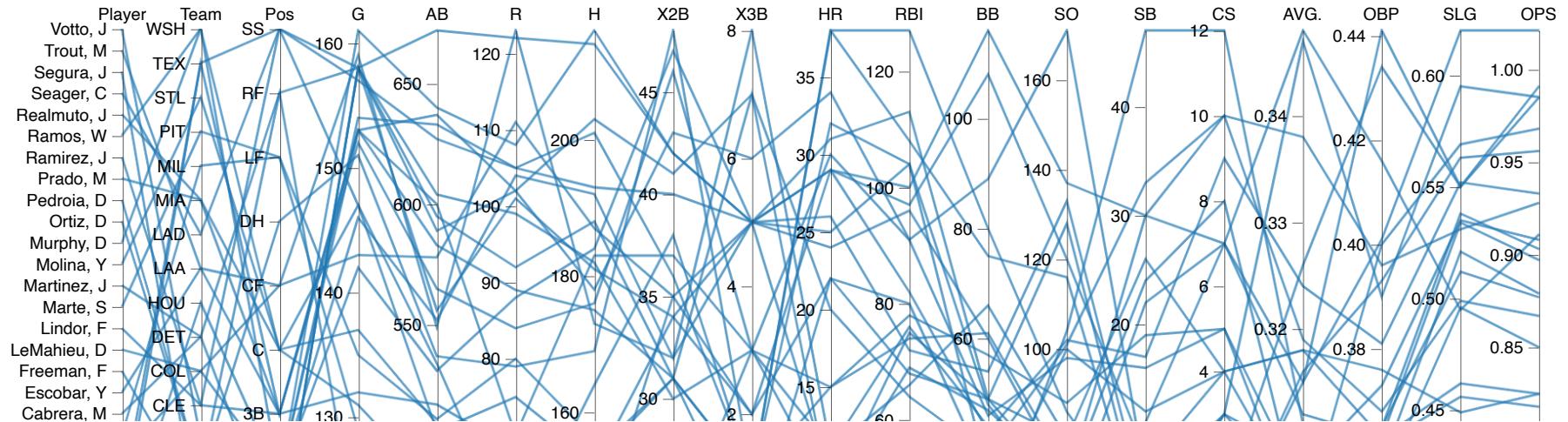
```
baseball <- read.csv("MetsStats2017.csv")
myposorder <- baseball %>% group_by(Pos) %>%
  summarize(n = n()) %>% arrange(n)
baseball$Pos <- factor(baseball$Pos,
  levels = myposorder$Pos)
data.frame(baseball) %>% arrange(Pos) %>%
  select(-Team) %>%
  parcoords(
    rownames = F # turn off rownames from the data.frame
    , brushMode = "1D-axes"
    , reorderable = T
    , queue = T
    , color = list(
      colorBy = "Region"
      ,colorScale = "scaleOrdinal"
      ,colorScheme = "schemeCategory10"
      )
    , withD3 = TRUE
    , width = 1000
    , height = 400
  )
)
```

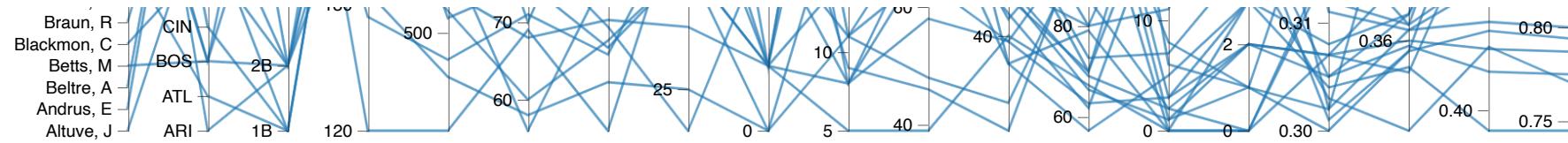




# 2016 Top Hitters (25 highest AVG)

```
baseball <- read.csv("TopHitters2016.csv")
myteamorder <- baseball %>% group_by(Team) %>%
  summarize(n = n()) %>% arrange(n)
baseball$Team <- factor(baseball$Team,
  levels = myteamorder$Team)
data.frame(baseball) %>% arrange(Team) %>%
  parcoords(
    rownames = F # turn off rownames from the data.frame
    , brushMode = "1D-axes"
    , reorderable = T
    , queue = T
    , color = list(
      colorBy = "Region"
      ,colorScale = "scaleOrdinal"
      ,colorScheme = "schemeCategory10"
      )
    , withD3 = TRUE
    , width = 1000
    , height = 400
  )
```



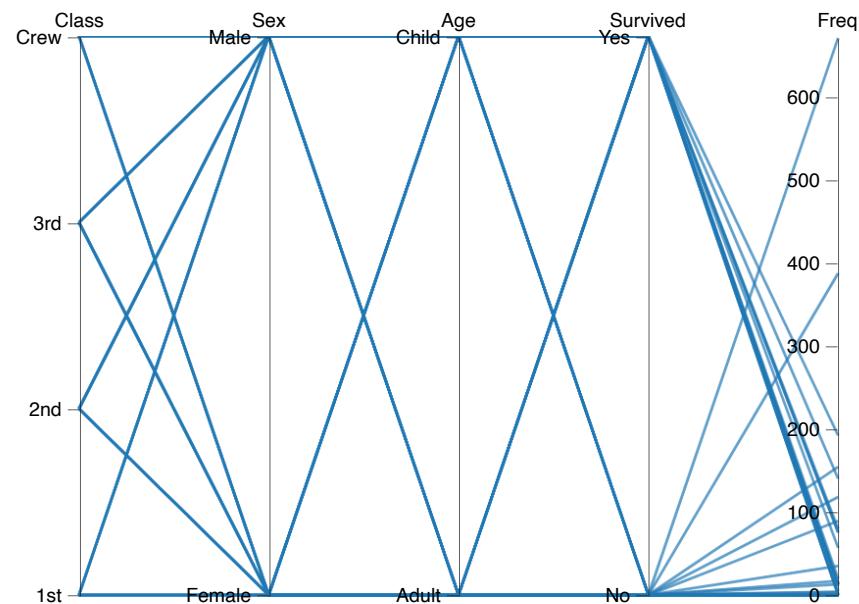


# SAT data

<https://data.cityofnewyork.us/Education/SAT-College-Board-2010-School-Level-Results/zt9s-n5aj/data>

# Watch out for categorical variables

```
# parcoords package (htmlwidget)
data.frame(Titanic) %>%
  parcoords(
    rownames = F # turn off rownames from the data.frame
    , brushMode = "1D-axes"
    , reorderable = T
    , queue = T
    , color = list(
      colorBy = "Region"
      ,colorScale = "scaleOrdinal"
      ,colorScheme = "schemeCategory10"
      )
    , withD3 = TRUE
  )
```





# Parallel coordinate plots

- `ggplot2::geom_line()`
- `GGally::ggparcoord()` (static, `ggplot2`)
- `MASS:: parcoord()` (static, `base`)
- `parcoords::parcoords()` (interactive)  
`devtools::install_github("timelyportfolio/parcoords")`  
<http://www.buildingwidgets.com/blog/2015/1/30/week-04-interactive-parallel-coordinates-1>