

# ggformula/lattice conversion examples

Nicholas Horton (nhorton@amherst.edu)

2017-07-19

## Introduction

This document is intended to help users of the `mosaic` package migrate their `lattice` package graphics to `ggformula`. The `mosaic` package provides a simplified and systematic introduction to the core functionality related to descriptive statistics, visualization, modeling, and simulation-based inference required in first and second courses in statistics. Originally, the `mosaic` package used `lattice` graphics but now support is also available for the `ggformula` system.

## References

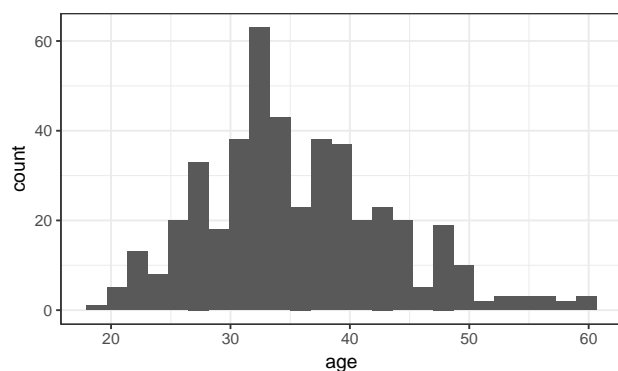
More information about `ggformula` can be found at <https://github.com/ProjectMOSAIC/ggformula>.

More information regarding Project MOSAIC (Kaplan, Pruim, and Horton) can be found at <http://www.mosaic-web.org>. Further information regarding the `mosaic` package can be found at <https://github.com/ProjectMOSAIC/mosaic> and <https://journal.r-project.org/archive/2017/RJ-2017-02>.

Examples of how to bring multidimensional graphics into day one of an introductory statistics course can be found at <http://escholarship.org/uc/item/84v3774z>.

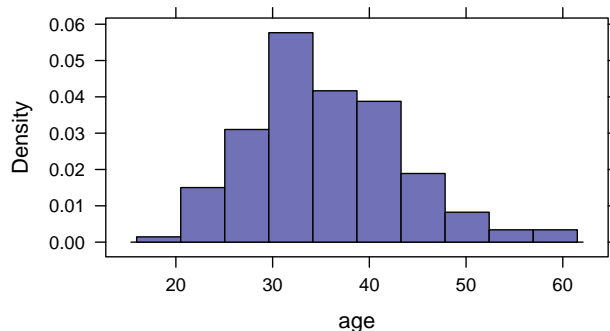
## Histograms (ggformula)

```
library(mosaic)
library(ggformula)
gf_histogram(~ age, data = HELPrct)
```



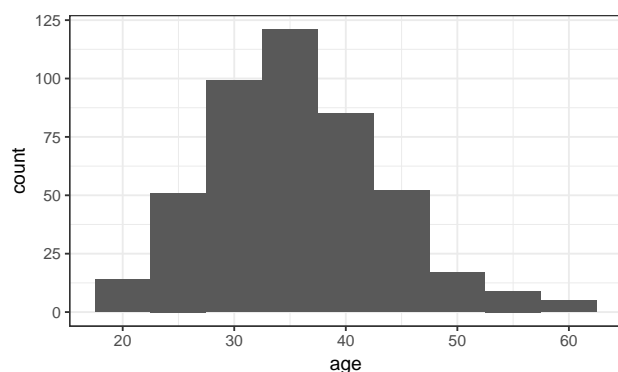
## Histograms (lattice)

```
library(mosaic)
library(lattice)
histogram(~ age, data = HELPrct)
```



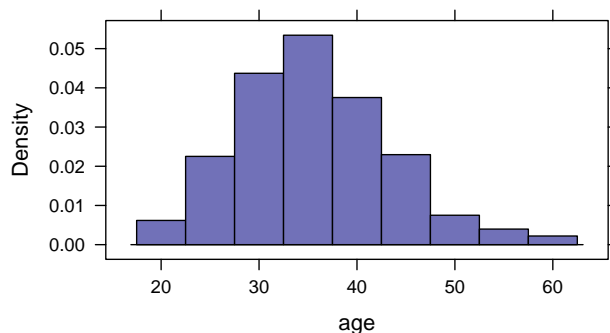
## Histogram options (ggformula)

```
gf_histogram(~ age, data = HELPrct,
             binwidth = 5)
```



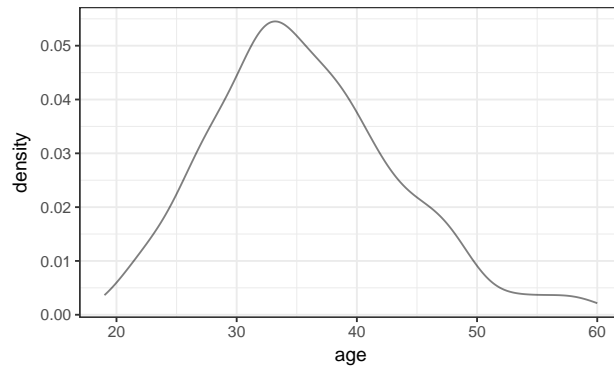
## Histogram options (lattice)

```
histogram(~ age, width = 5, data = HELPrct)
```



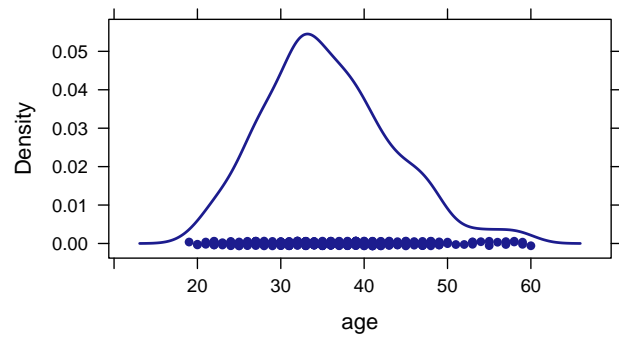
## Density plots (ggformula)

```
gf_dens(~ age, data = HELPrct)
```



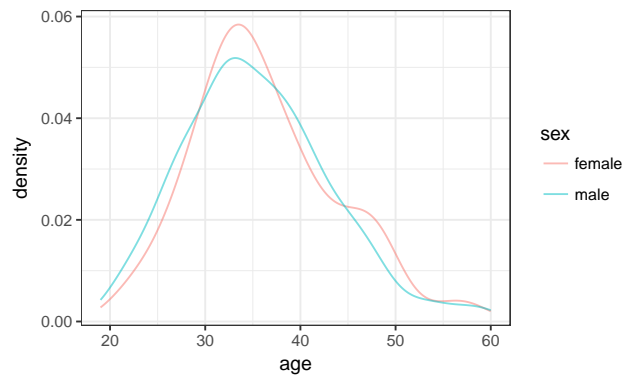
## Density plots (lattice)

```
densityplot(~ age, data = HELPrct)
```



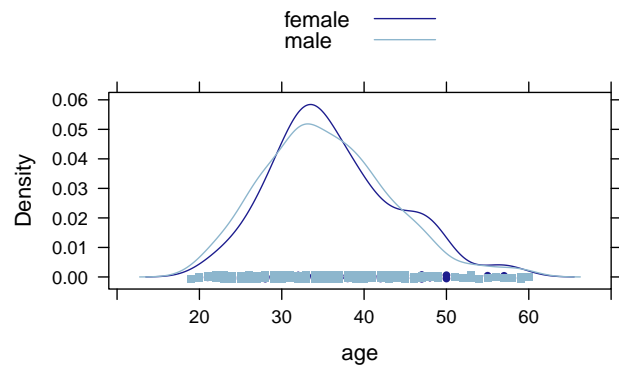
## Overlaid density plots (ggformula)

```
gf_dens(~ age, data = HELPrct,
        color = ~ sex)
```



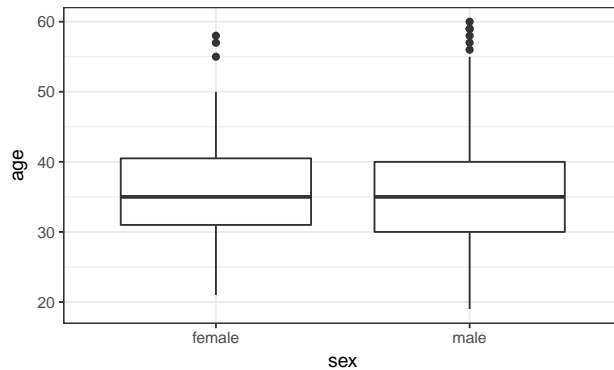
## Overlaid density plots (lattice)

```
densityplot(~ age, data = HELPrct,
            groups = sex, auto.key = TRUE)
```



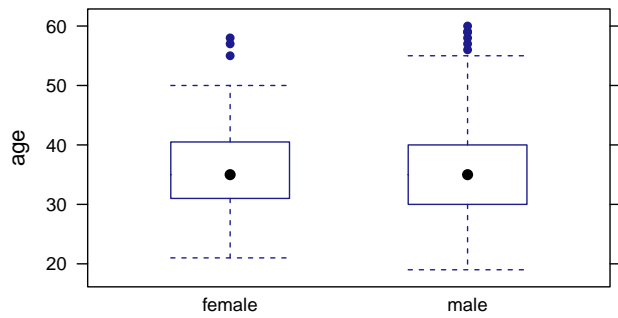
## Side by side box plots (ggformula)

```
gf_boxplot(age ~ sex, data = HELPrct)
```



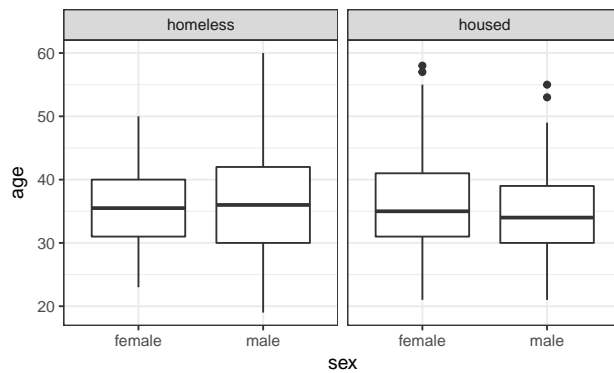
## Side by side plots (lattice)

```
bwplot(age ~ sex, data = HELPrct)
```



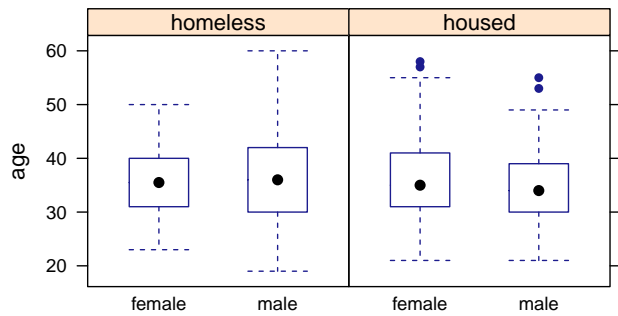
## Faceted side by side box plots (ggformula)

```
gf_boxplot(age ~ sex | homeless, data = HELPrct)
```



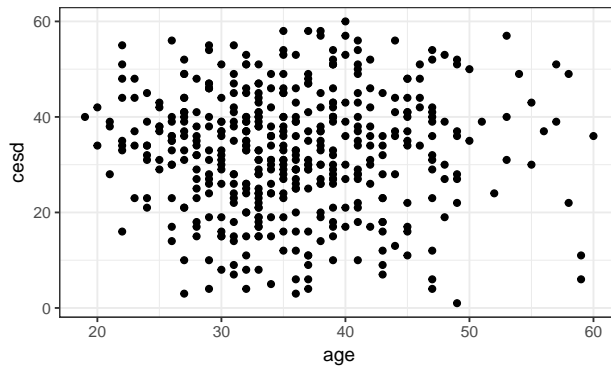
## Faceted side by side plots (lattice)

```
bwplot(age ~ sex | homeless, data = HELPrct)
```



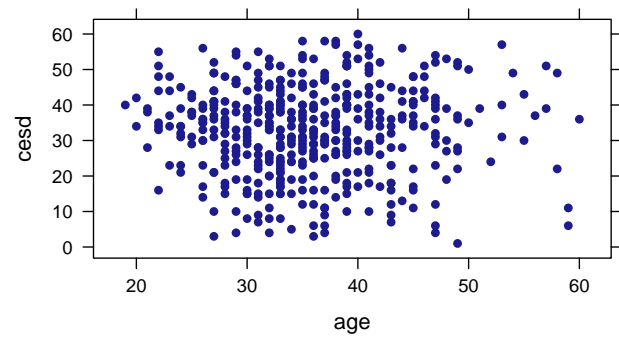
## Scatterplot (ggformula)

```
gf_point(cesd ~ age, data = HELPrct)
```



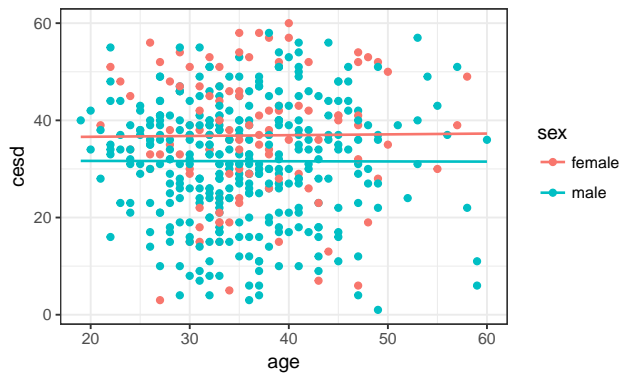
## Scatterplot (lattice)

```
xyplot(cesd ~ age, data = HELPrct)
```



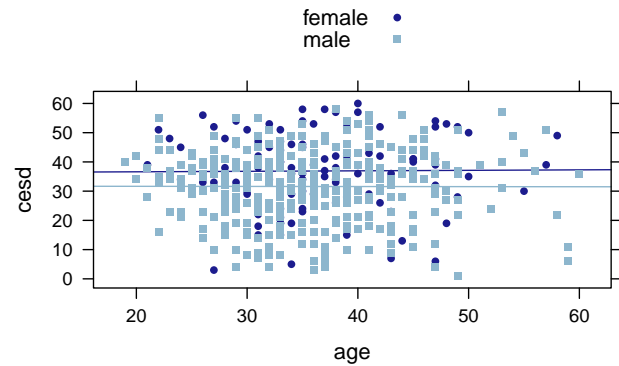
## Overlaid scatterplot with linear fit (ggformula)

```
gf_point(cesd ~ age, data = HELPrct,
         color = ~ sex) %>%
  gf_lm()
```



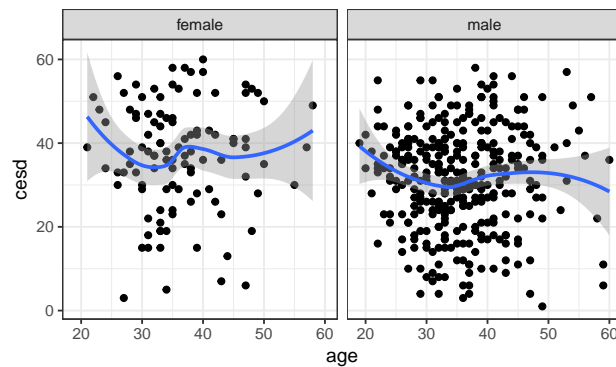
## Overlaid scatterplot with linear fit (lattice)

```
xyplot(cesd ~ age, data = HELPrct,
       groups = sex,
       type = c("p", "r"),
       auto.key = TRUE)
```



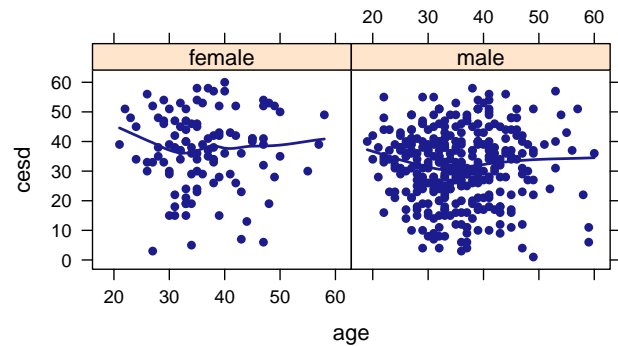
## Faceted scatterplot with smooth fit (ggformula)

```
gf_point(cesd ~ age | sex,
         data = HELPrct) %>%
  gf_smooth()
```



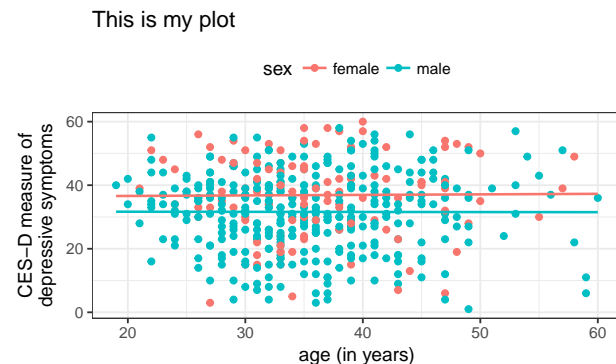
## Faceted scatterplot with smooth fit (lattice)

```
xyplot(cesd ~ age | sex, data = HELPrct,
       type = c("p", "smooth"),
       auto.key = TRUE)
```



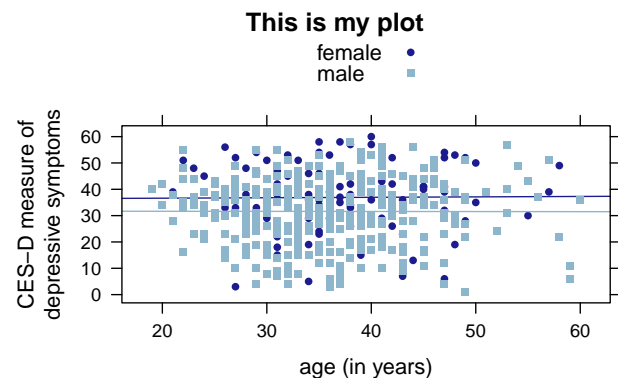
## More options for scatterplot with linear fit (ggformula)

```
gf_point(cesd ~ age, data = HELPrct,
         color = ~ sex) %>%
  gf_lm() %>%
  gf_theme(legend.position = "top") %>%
  gf_labs(title = "This is my plot",
         x = "age (in years)",
         y = "CES-D measure of
depressive symptoms")
```



## More options for scatterplot with linear fit (lattice)

```
xyplot(cesd ~ age, groups = sex,
       type = c("p", "r"),
       auto.key = TRUE,
       main = "This is my plot",
       xlab = "age (in years)",
       ylab = "CES-D measure of
depressive symptoms",
       data = HELPrct)
```



## Want to explore more?

Within RStudio, after loading the `mosaic` package, try running the command `mplot(ds)` where `ds` is a dataframe. This will open up an interactive visualizer that will output the code to generate the figure (using `lattice`, `ggplot2`, or `ggformula`) when you click on Show Expression.