Sunderlab tutorial: clustering pt 2

Jahred Liddie

2024-04-05

Revisiting: What is clustering?

I don't think I've seen a very consistent definition of "clustering" either. It also probably varies depending on which field you're referring to and who you ask. A more important point (on which methods to consider) is probably what your research question is and which methods can help you answer that. Here are some very broad categories based on my understanding:

- **Methods that account for (pre-defined) grouping(s)**: this is where I would put multilevel/mixed-effect/hierarchical/random-effects models, clustered standard errors, cluster bootstrapping, generalized estimating equations, fixed-effects models (which themselves are pretty loosely defined), and probably more
  - In this case, you (sometimes) may not be as interested in the clusters themselves, but need to account for them because traditional methods do not.
- **Methods to help you figure out what clusters may exist**: this is where I would put KNN, DBSCAN, PCA, hierarchical clustering, and probably many other unsupervised methods
- **"Mixture" methods:** This definitely overlaps with the above, but these include a set of methods for estimating the effects of mixtures (often chemical mixtures) on some kind of outcome. Dimension reduction and variable selection may also be incorporated with these methods. Some methods in this group include: lasso/ridge/elastic net regression, weighted quantile sum regression (discussed here), and Bayesian Kernel Machine Regression
  - If you're interested in these methods, I'd recommend taking a look at Dr. Andrea Bellavia's free online introduction here.

Next is a loose, very preliminary introduction to scratch the surface of some of this material.

Clustered data exercise 1 ("multilevel" data)

Let's talk a bit more about the first group of methods. Traditional regression methods (e.g., linear models and the broad group of generalized linear models) make a few key assumptions, including an assumption that observations in the sample are independently distributed. Other assumptions may be violated (like homoscedasiticity) depending on the type of model as well. When these assumptions are violated, standard errors (related to precision of your estimates) and/or the coefficient estimates themselves may be biased. Here are some examples of what "clustering" could mean in this case:

- Students who share the same teacher may perform similarly on an exam - they may be more similar to each other than two students randomly chosen from different classrooms

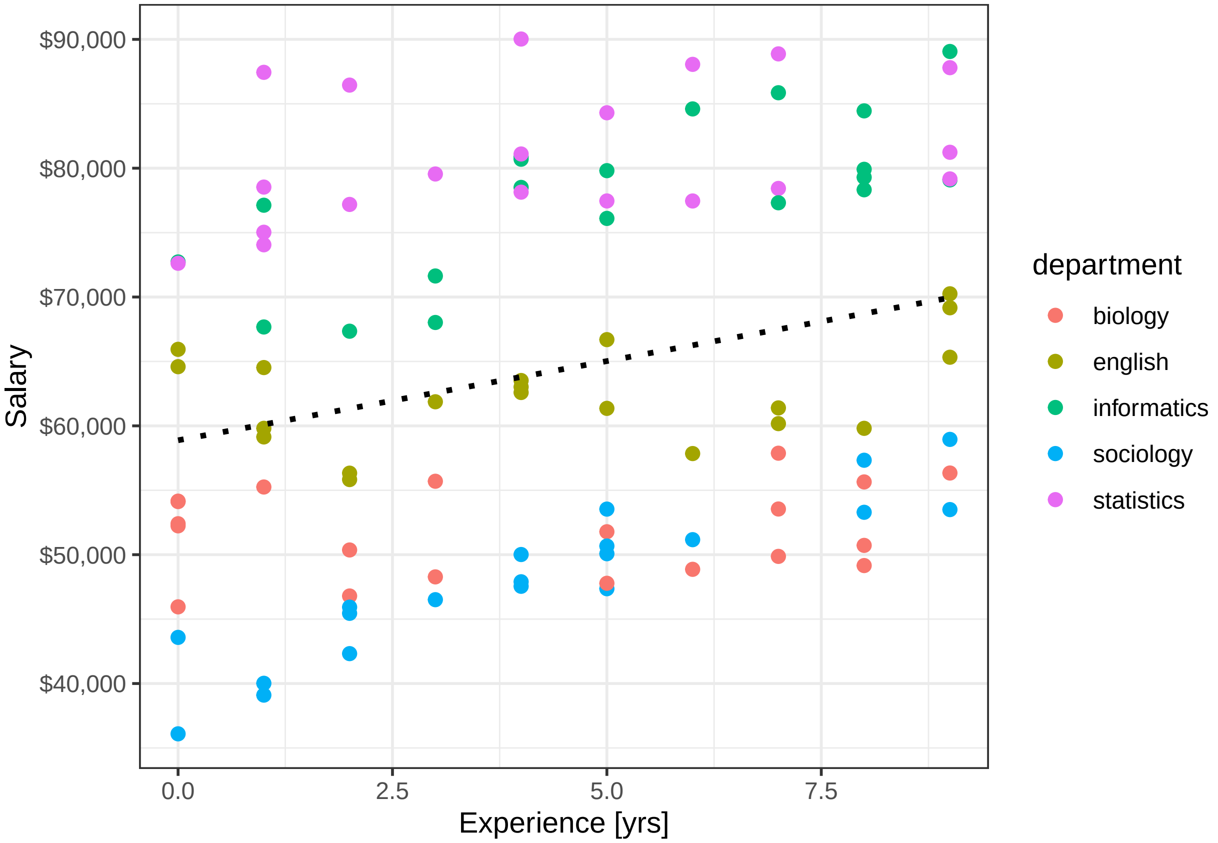
- $PM_{2.5}$ concentrations from different EPA monitors over time
- Water systems in the same county or state or region
- Silly example: you artificially duplicate your sample and run the same analysis

In addition to accounting for clustered observations, the methods described earlier may also be important for other settings, including:

- when there are predictors measured at multiple "levels" (e.g., personal income and census tract income) or the coefficient you're interested in is at a "higher" level
- when we want to assess whether the strength of an association or effect differs for different clusters/groups
- when we want to describe how variation is explained by the clusters/groups alone in comparison to total variation

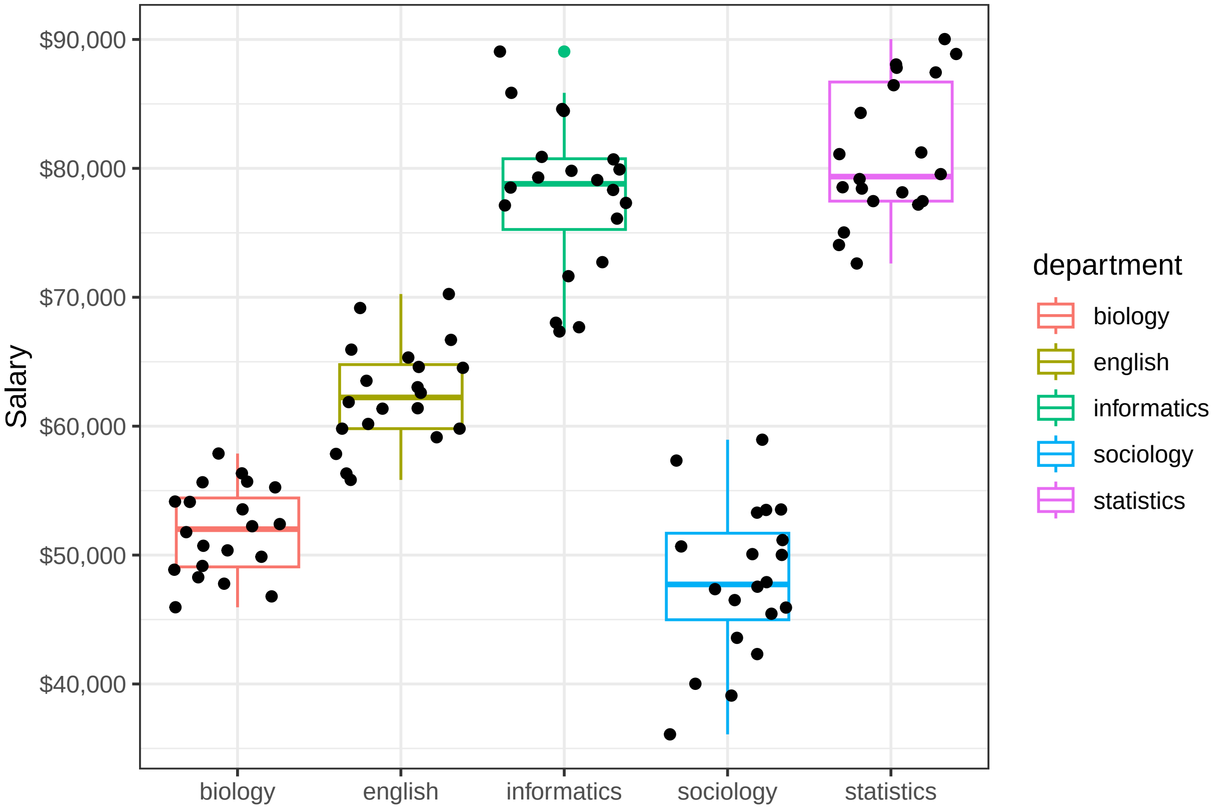
Example below uses fake (simulated) data on salaries and years of experience to give some more intuition - taken from this site that also visualizes it more interactively.

```r
# `salaries` has faculty data on salaries by department
# is experience associated with differences in salary?
# visualize salaries and experience, making depts different colors
ggplot(salaries, aes(x = experience, y = salary)) +
  geom_point(aes(color = department), size = 2) +
  geom_smooth(method = "lm", se = FALSE, color = "black", linetype = "dotted") +
  scale_y_continuous(labels = scales::dollar_format()) +
  labs(x = "Experience [yrs]", y = "Salary") +
  theme_bw()
```



```r
# same as the line in the plot
mod1 <- lm(salary ~ experience, data = salaries)
# summary(mod1)

# another quick visual check is a grouped boxplot:
ggplot(salaries, aes(x = department, y = salary)) +
  geom_boxplot(aes(color = department)) +
  geom_jitter() +
  scale_y_continuous(labels = scales::dollar_format()) +
  labs(x = "", y = "Salary") +
  theme_bw()
```



The line above is equivalent to the following simple linear regression:

$$Salary_i = \beta_0 \ + \ \beta_1 experience_i \ + \ \epsilon_i$$
$$\epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

However, you may notice that it's likely in these data that different departments have different "baseline" salaries (for example when experience is equal to 0), even if the change in salary for each year of experience is similar across departments. This is when we could use a multilevel model (with random intercepts for each department) - this can be written like:

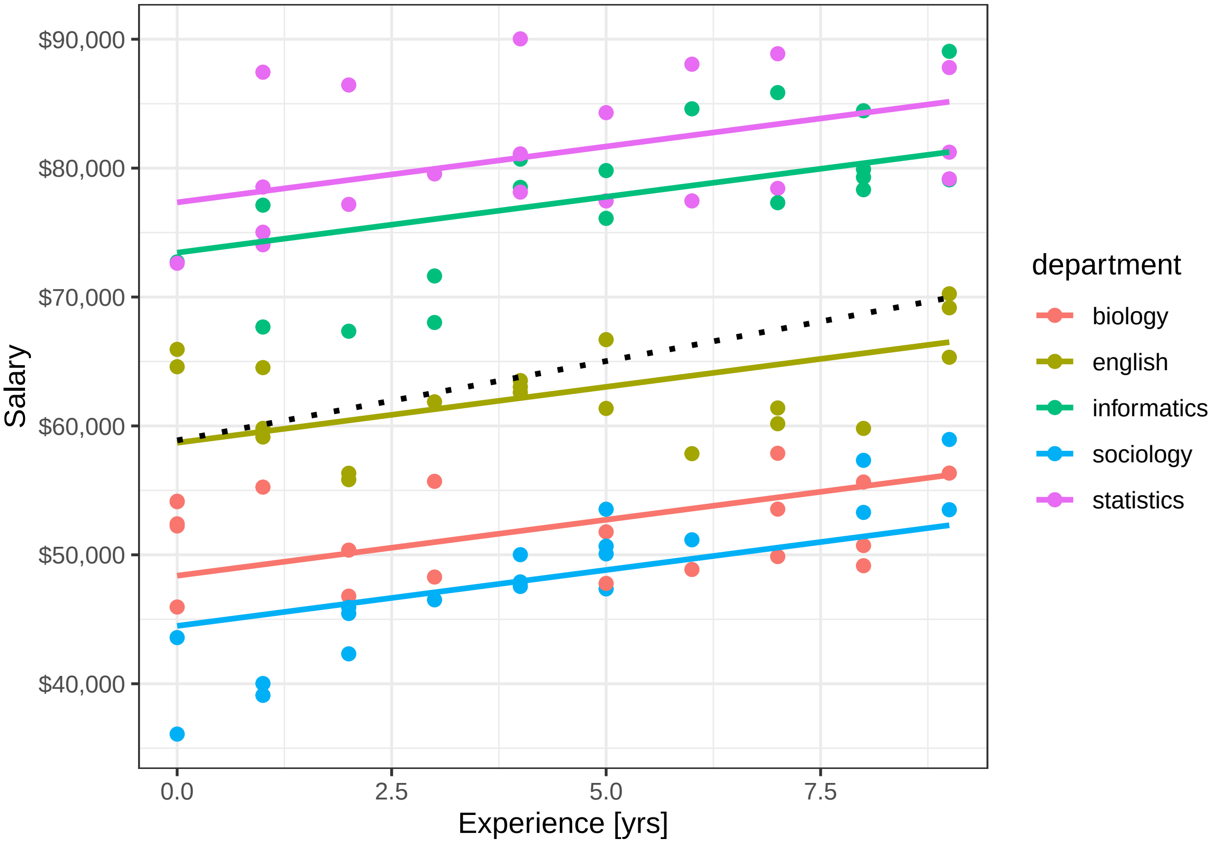
$$Salary_{ij} = \beta_{00} \ + \ \beta_1 experience_{ij} \ + \ \epsilon_{ij} \ + \ u_j$$
$$\epsilon_{ij} \sim \mathcal{N}(0, \sigma^2)$$
$$u_j \sim \mathcal{N}(0, \tau^2)$$

where u_j serves as an "offset" term for each department j from the "grand mean" across departments. Note the additional notation for how this offset term is assumed to be distributed. Note: In this case, we may also instead turn to a fixed-effects regression model (which actually gives very similar results in this case), especially since the number of groups/clusters is not very large.

```
# example with a random intercept model - this has lmer notation for the intercept
mod2 <- lmer(salary ~ experience + (1 | department), data = salaries)

salaries$pred_mod1 <- predict(mod1)
salaries$pred_mod2 <- predict(mod2)

ggplot(salaries, aes(x = experience, y = salary)) +
  geom_point(aes(color = department), size = 2) +
  geom_line(aes(y = pred_mod2, color = department),
            linewidth = 1) +
  geom_line(aes(y = pred_mod1), linetype = "dotted", linewidth = 1) +
  scale_y_continuous(labels = scales::dollar_format()) +
  labs(x = "Experience [yrs]", y = "Salary") +
  theme_bw()
```



Now, conceptually, also consider what a "random slope" model (without random intercepts) would allow for or what a random slope *and* random intercept model would allow. *Hint: what do you notice about the regression lines for the random intercept model?*

We could also use this model to see what proportion of the total variation in salaries is explained by differences between departments.

If you're interested in using these methods for your work, there are some good textbooks out there (see below). Harvard also has several classes that focus on these.

- Gelman and Hill, 2007

- Raudenbush and Bryk, 2002

Clustered data 2 (mixtures)

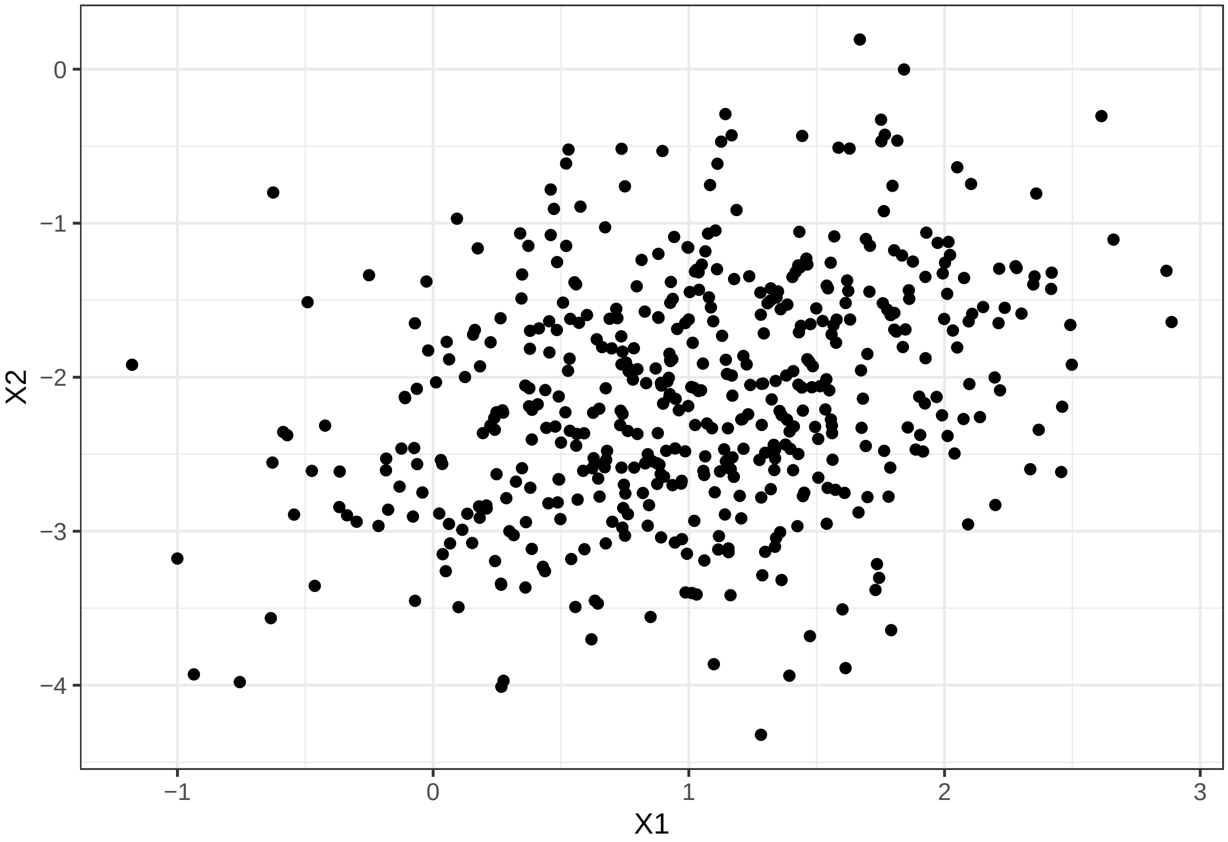
Let's change our focus now to "clustering" in the form of mixtures, which is probably more related to what we talked about last time. In environmental health, a "mixture" would refer to at least 3 (likely) correlated chemical exposures, but the idea of a mixture can show up other disciplines too. This section refers heavily to Dr. Andrea Bellavia's course and online material. Research questions related to mixtures in environmental health could include the following and different methods can address these questions:

- What are common exposure patterns?
- Which exposures are toxic?
- What is the cumulative effect of the mixture?*
- Is there synergy/interaction between the exposures?
- What is the exposure-response function for an exposure in a mixture?

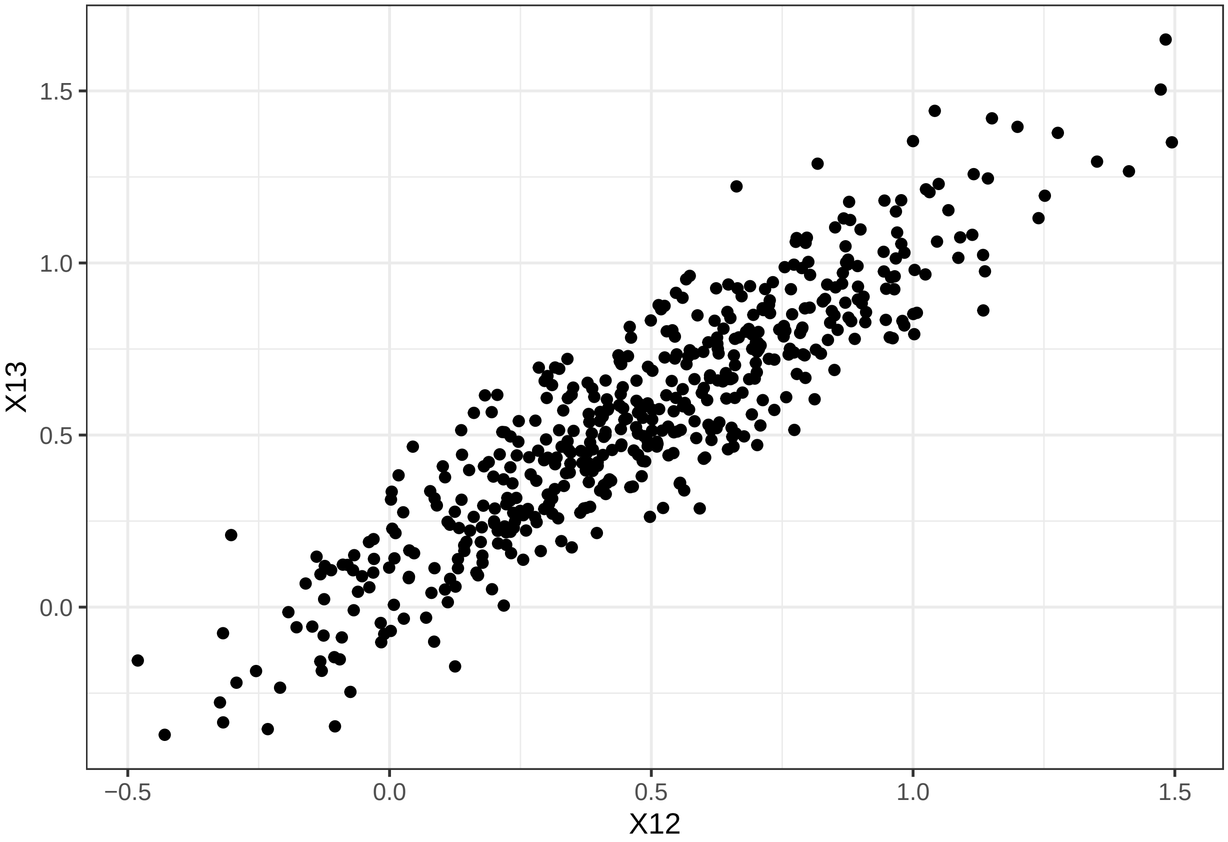
Generally, unsupervised methods (including a few introduced in the prior tutorial) can help answer questions related to exposure patterns, while supervised methods will be used to understand health outcomes due to exposures (or whatever outcome you're interested in). Including all exposures in one regression model is an option, but can become unfeasible if there is: (1) significant multi-collinearity among exposures, (2) overfitting (esp. with a lot of exposure variables and a small sample size), and (3) bias amplification can also occur (not going more into this specific point - examples of this I have seen include DAGs).

```r
# check columns in dataset
# names(mixture_dat) # there are 14 "exposure" variables, one health outcome y,
# and three other covariates (z1-z3)

# often advisable to first visualize correlated variables,
  # like we did at last tutorial
  # one option is just a scatterplot with two dimensions:
  ggplot(mixture_dat, aes(x = x1, y = x2)) +
    geom_point() +
    labs(x = "X1", y = "X2") +
    theme_bw()
```

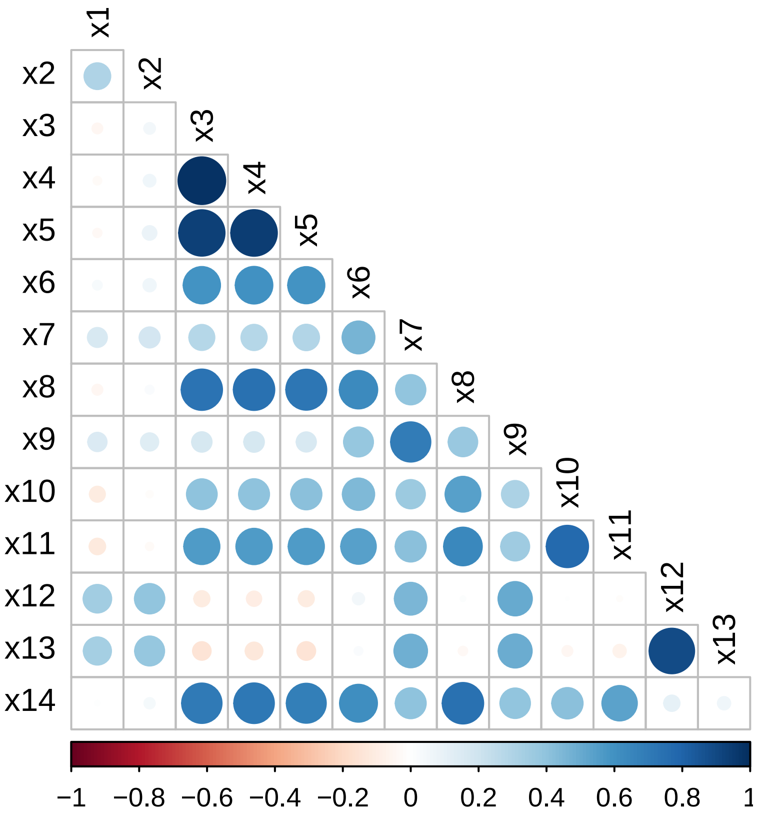


```r
ggplot(mixture_dat, aes(x = x12, y = x13)) +
  geom_point() +
  labs(x = "X12", y = "X13") +
  theme_bw()
```



```
# correlation matrices are sometimes used to visualize all
  # bivariate correlation estimates
  # construct matrix:
  cor.matrix <- cor(mixture_dat[,3:16], method = "spearman")

  corrplot(cor.matrix,
           method = "circle",
           tl.col = "black",
           type = "lower",
           diag = FALSE,
           sig.level = 0.05)
```



An approach that is somewhat intuitive is weighted quantile sum regression (WQS). WQS is an approach that summarizes the exposures in the mixture when estimating effects on an outcome. It does this by weighting each exposures contribution to the effect of the overall mixture. Training and validation steps are typically used for this method. Here is a representation of WQS for the `mixture_dat` dataframe:

$$Y = \beta_0 \ + \ \beta_1 \left(\sum_{j=1}^{p} w_j q_j \right) \ + \ Z^T \beta \ + \ \epsilon$$

The weights are further constrained such that their sum equals 1 and that $0 \leq w_j \leq 1$. When splitting into training and testing data, the weights for each component of the mixture are constructed using the averages across all bootstrapped datasets.

A very important assumption of this method is that β_1 is assumed to be either positive or negative, either in the estimation step for each bootstrapped sample or when constructing final weights after bootstrapping.

We will try this in the case of a continuous outcome, which is fitted similarly to a linear model, although WQS can be extended for other kinds of outcomes. (Upon further reading in Carrico et al., it seems that bootstrapped samples in which β_1 are larger in magnitude get more weight in constructing the final weights, although I'm not sure why. If you're interested more in how parameters in WQS are estimated mathematically, I'd recommend looking at Carrico et al., 2015.)

```
# define the names of the exposures in the mixture
exposures <- names(mixture_dat[,3:16]) # can also get them w/ grepl here

WQS_results <- gwqs(y ~ wqs,
                    mix_name = exposures,
                    data = mixture_dat,
                    # this specifies what quantile of the exposure
```

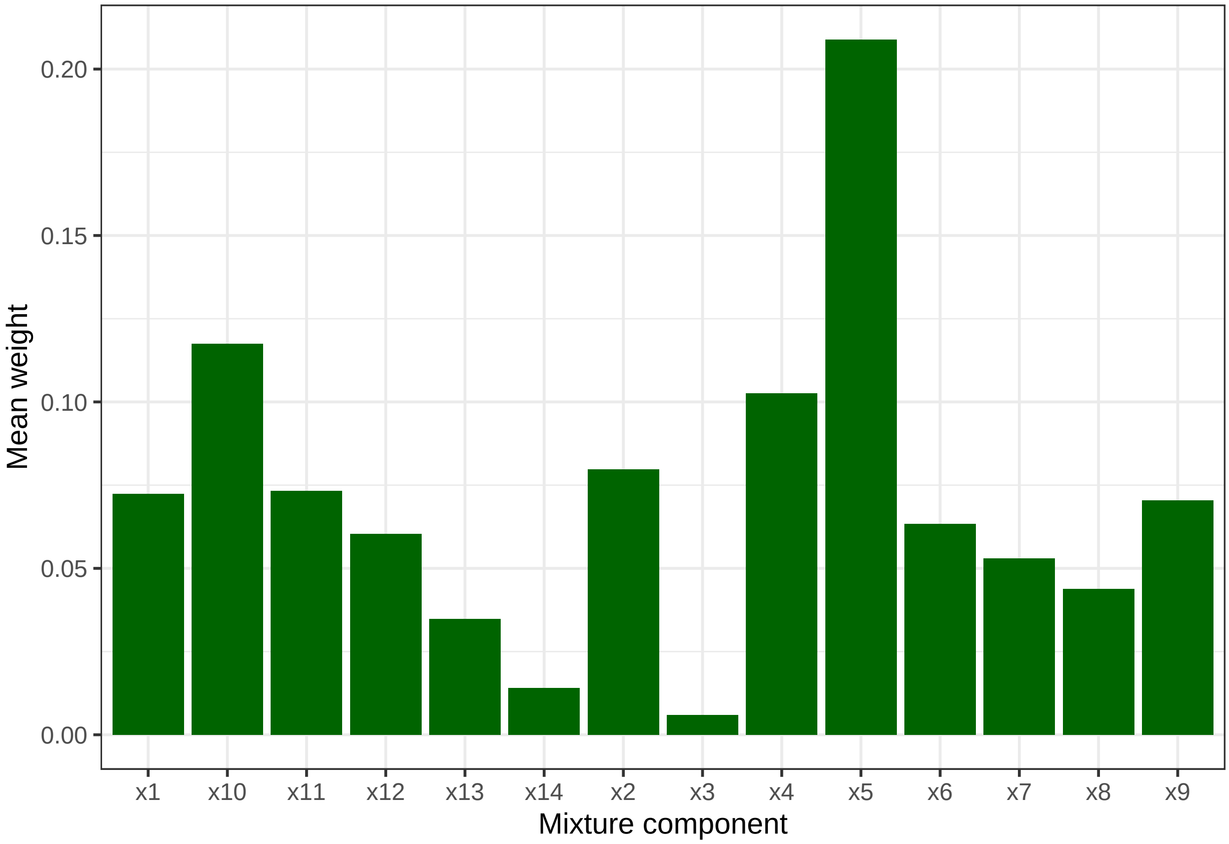
```r
                      # variables will be used
                      # here we're using quartiles
                      # q = NULL will consider actual values,
                      # but is more sensitive to outliers
                      # and variables should be standardized first
                    q = 4,
                    validation = 0.6,
                    b = 100,
                    b1_pos = T,
                    family = "gaussian",
                    seed = 123)

# summary of result on held-out data
summary(WQS_results)
```

```
##
## Call:
## gwqs(formula = y ~ wqs, data = mixture_dat, mix_name = exposures,
##     b = 100, b1_pos = T, q = 4, validation = 0.6, family = "gaussian",
##     seed = 123)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.45766  -0.37160  -0.00197   0.36409   1.62019
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.08339    0.08317   37.07   <2e-16 ***
## wqs          0.52746    0.05037   10.47   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.3315459)
##
##     Null deviance: 136.81  on 304  degrees of freedom
## Residual deviance: 100.46  on 303  degrees of freedom
## AIC: 532.83
##
## Number of Fisher Scoring iterations: 2
```

```r
WQS_plot <- as_tibble(WQS_results$final_weights)

ggplot(WQS_plot, aes(x = mix_name, y = mean_weight)) +
  geom_bar(fill = "darkgreen", stat = "identity") +
  labs(x = "Mixture component", y = "Mean weight") +
  theme_bw()
```



How would you interpret this plot?

What seem to be the disadvantages of WQS?

If you're interested in using methods like WQS, there are also more recent advances in these methods to relax some of the assumptions it makes.