# Module 1: Introduction to PheCAP data

Siyue Yang, Jianhui Gao, and Jesse Gronsbell

The goal of phenotyping is to predict patients' disease status from electronic health record data.

In this module, we will go through a public released dataset from an R package PheCAP to get hands-on experience of phenotyping.

```
# Load the packages.
packages <- c("tidyverse", "PheCAP", "corrplot", "ggplot2")

# Check if the packages are missing or not.
# If missing, install automatically.
# If not missing, load the package.
package.check <- lapply(
      packages,
      FUN = function(x) {
        if (!require(x, character.only = TRUE)) {
          install.packages(x, dependencies = TRUE)
          library(x, character.only = TRUE)
        }
      }
    )
```

## PheCAP

https://celehs.github.io/PheCAP/

The most likely explanation is that this is a random sample of patients (for public release from a previous study) in the Partner's EHR database (4.6 million patients) with diabetes mellitus (DM) and who met

(i) an initial filter for CAD: $\geq 1$ ICD9 code for CAD (410.x, 411.x, 412.x, 414.x, 413.x), or
(ii) $\geq 1$ NLP mention for any CAD related concepts: CAD, CAD procedures, CAD biomarkers, positive stress test.

```
# Load helper functions.
source("../Rscripts/helper_function.R")
```

## PheCAP data

```
load('../data/CAD_norm_pub.rda')
```

## Elementary data exploration

```
sum(!is.na(y))
```

```
## [1] 181
```

```
x %>% head()
```

- Labels: "y", whether the patient has the disease, **extracted by clinicians' chart review**
- Features: "surrogates" refers to total number of billing codes + NLP mentions of the disease
- Features: "healthcare_utilization" refers to total number of notes the patient has
- Features: "CODx" (n = 10), "NLPx" (n = 574) refers to the counts of a specific code or NLP term, extracted by SQL or NLP

What do you observe?

- 4,164 patients and 586 features.
- Label is subjective to missing.

### What is the prevalence of labels?

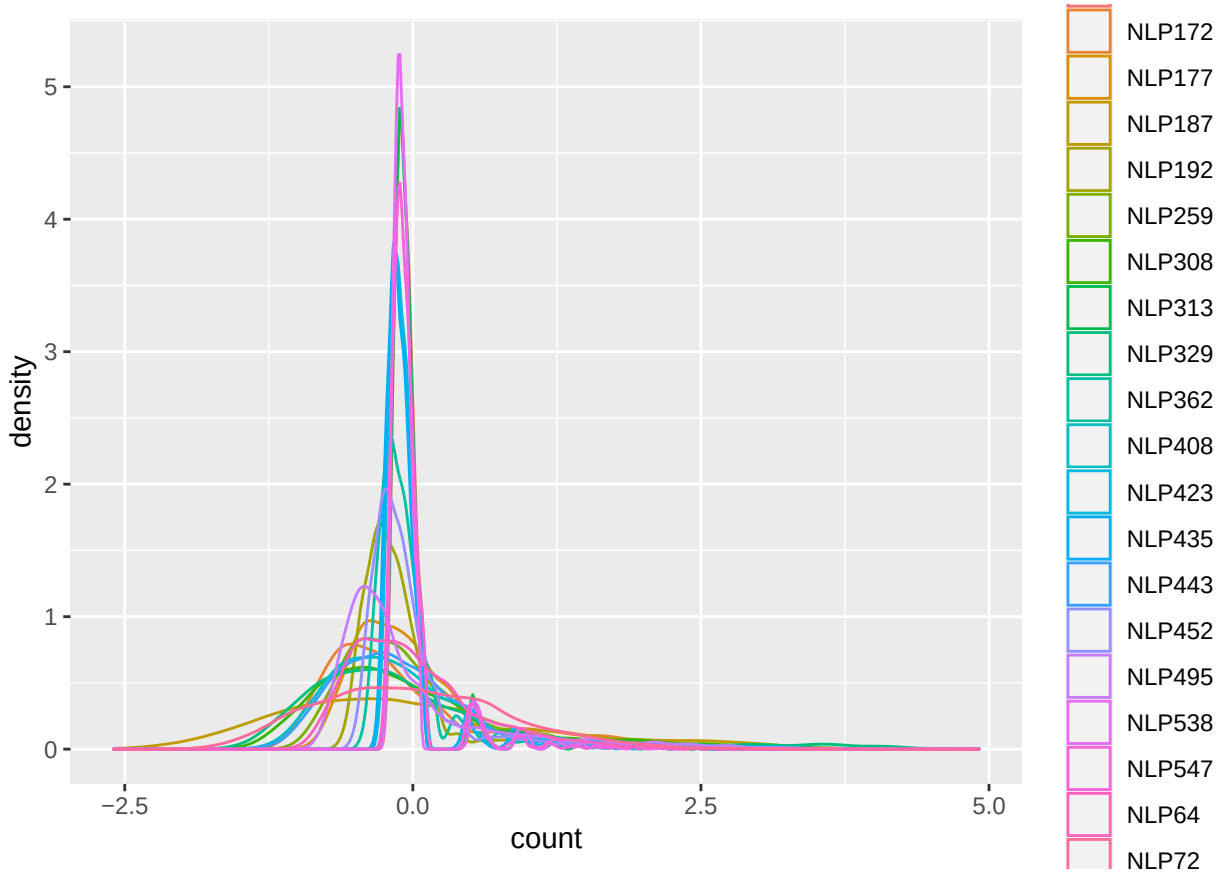
```
mean(y, na.rm = TRUE)
```

```
## [1] 0.6574586
```

### How features are distributed?

- Let's randomly sample a few features first.
- Observe the densities.

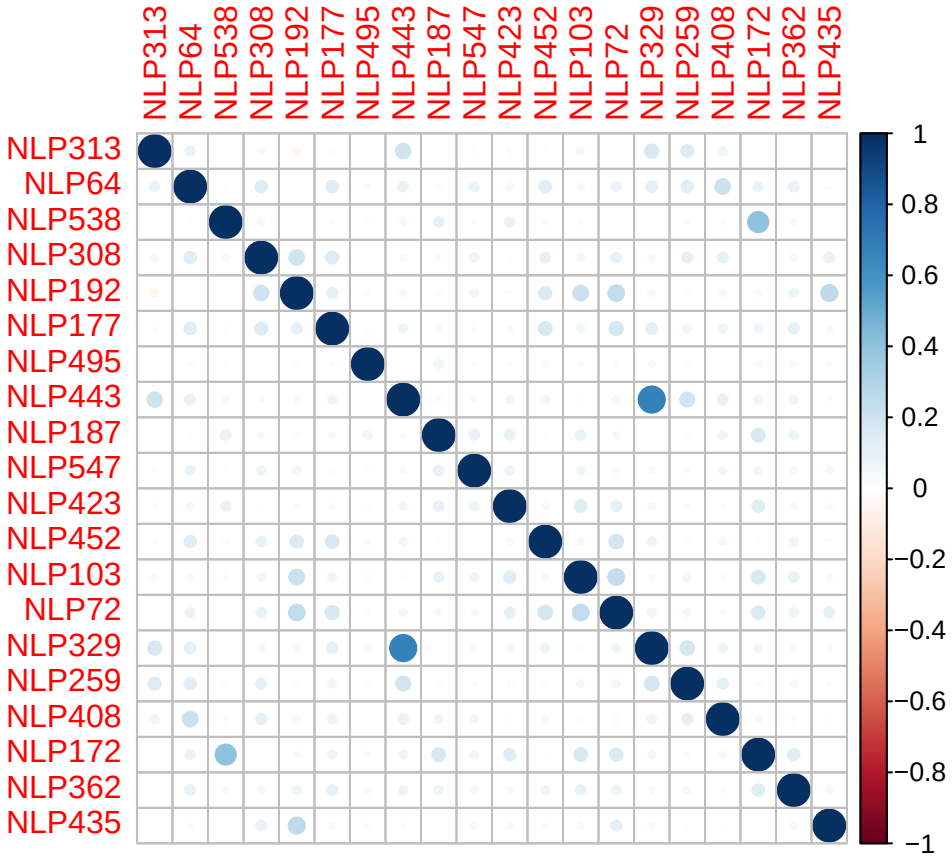
```
feature_index <- sample(c(1:ncol(x)), 20, replace = FALSE)

x[, feature_index] %>%
  pivot_longer(everything(), names_to = "feature", values_to = "count") %>%
  ggplot() +
  geom_density(aes(x = count, color = feature))
```




All the features are already orthogonalized and standardized.

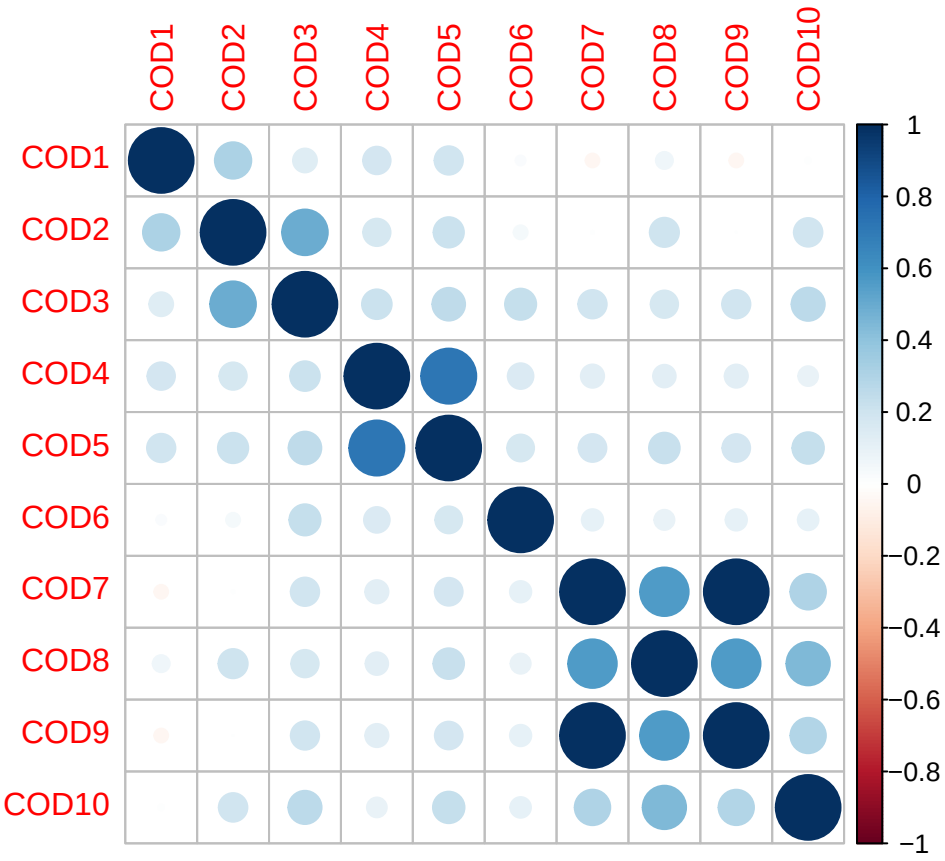
### What are correlations between features?

```
feature_cor <- cor(x[feature_index])
corrplot::corrplot(feature_cor)
```




### What about codified data?

```
feature_cor <- cor(x[3:12])
corrplot::corrplot(feature_cor)
```


COD1
COD2
COD3
COD4
COD5
COD6
COD7
COD8
COD9
COD10
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1