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

```
data(ehr_data)
data <- PhecapData(ehr_data, "healthcare_utilization", "label", 0.4, patient_id = "patient_id")
data
```

```
## PheCAP Data
## Feature: 10000 observations of 587 variables
## Label: 119 yes, 62 no, 9819 missing
## Size of training samples: 109
## Size of validation samples: 72
```

What do you observe?

- 10,000 patients and 587 features.
- Label is subjective to missing.
- Split into training and validation set.

Elementary data exploration

```
ehr_data %>% head()
```

- Labels: "label", whether the patient has the disease, **extracted by clinicians' chart review**
- Features: "main_ICD", "main_NLP" refers to total number of billing codes or 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

Missingness

```
colnames(ehr_data)[which( colMeans(is.na(ehr_data)) > 0)]
```

```
## [1] "label"
```

Only label is missing.

What is the prevalence of labels?

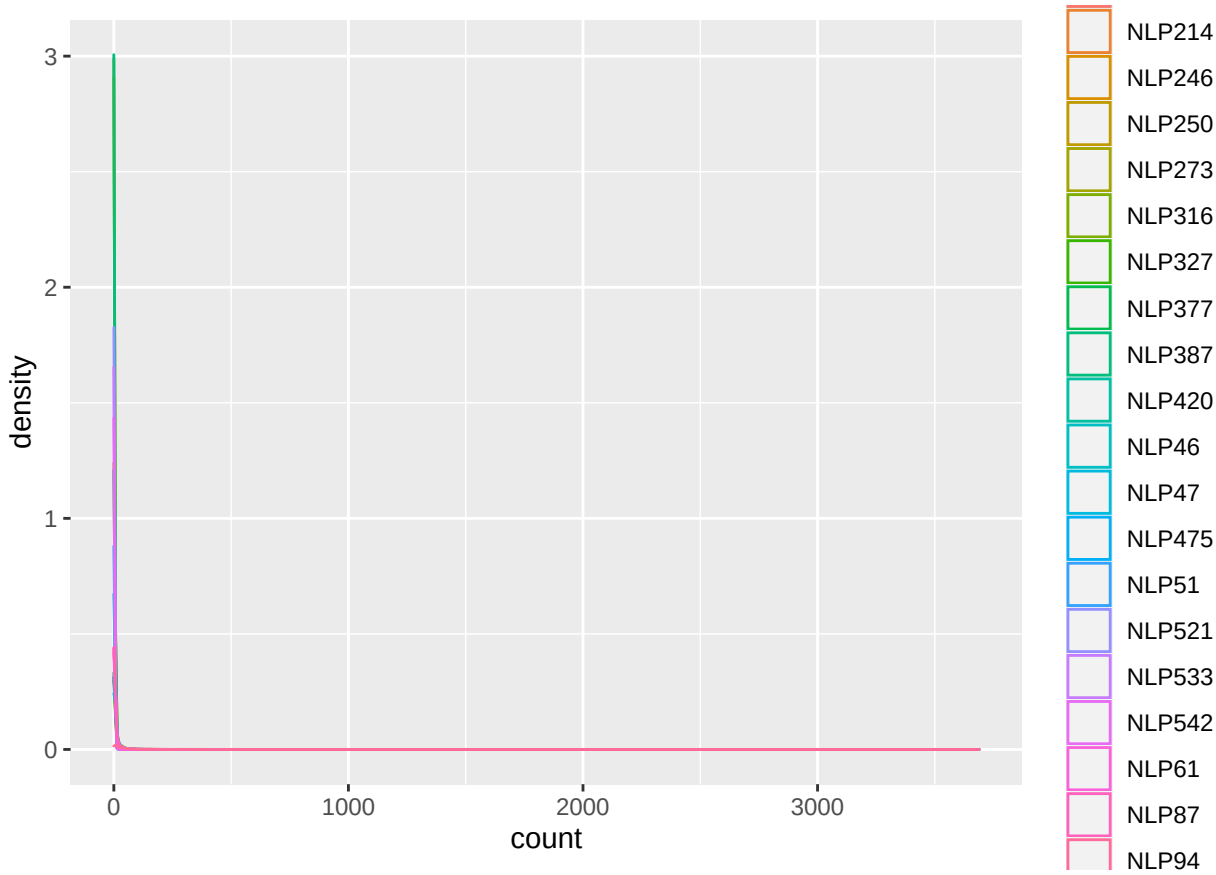
```
mean(ehr_data$label, na.rm = TRUE)
```

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

How features are distributed?

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

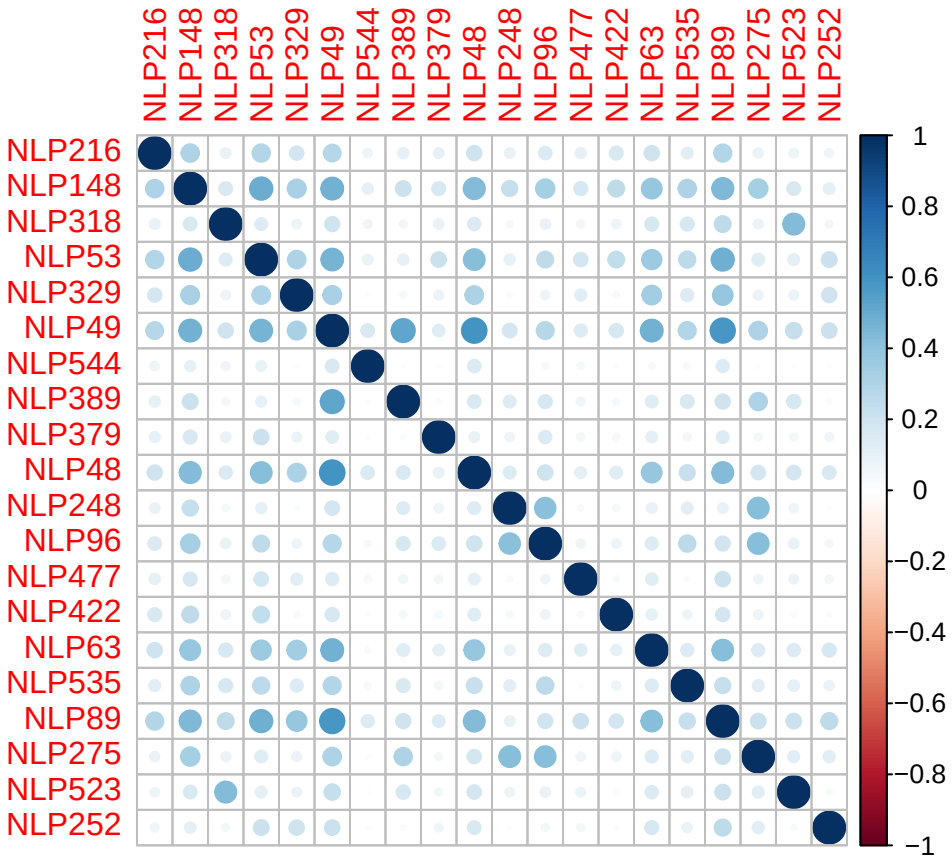
```
feature_index <- sample(c(3:ncol(ehr_data)), 20, replace = FALSE)
ehr_data[, feature_index] %>%
  pivot_longer(everything(), names_to = "feature", values_to = "count") %>%
  ggplot() +
  geom_density(aes(x = count, color = feature))
```



Too skewed.

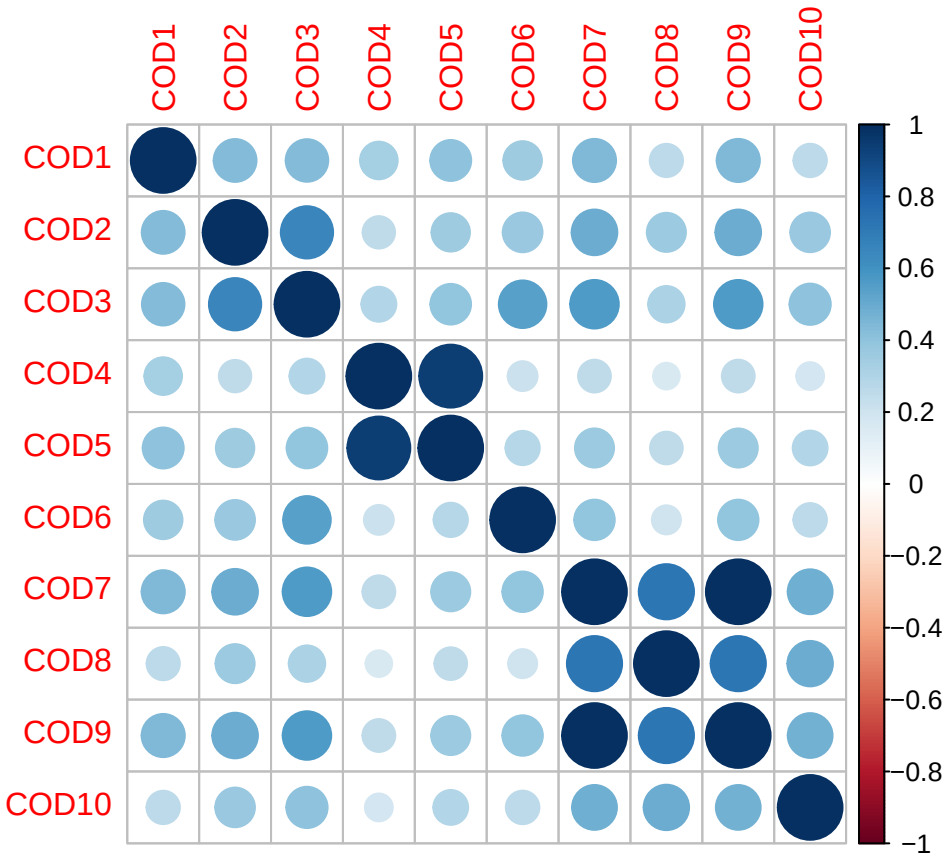
What are correlations between features?

```
features <- ehr_data[, c(3:ncol(ehr_data))]
feature_cor <- cor(features[feature_index])
corrplot::corrplot(feature_cor)
```



What about codified data?

```
feature_cor <- cor(features[4:13])
corrplot::corrplot(feature_cor)
```

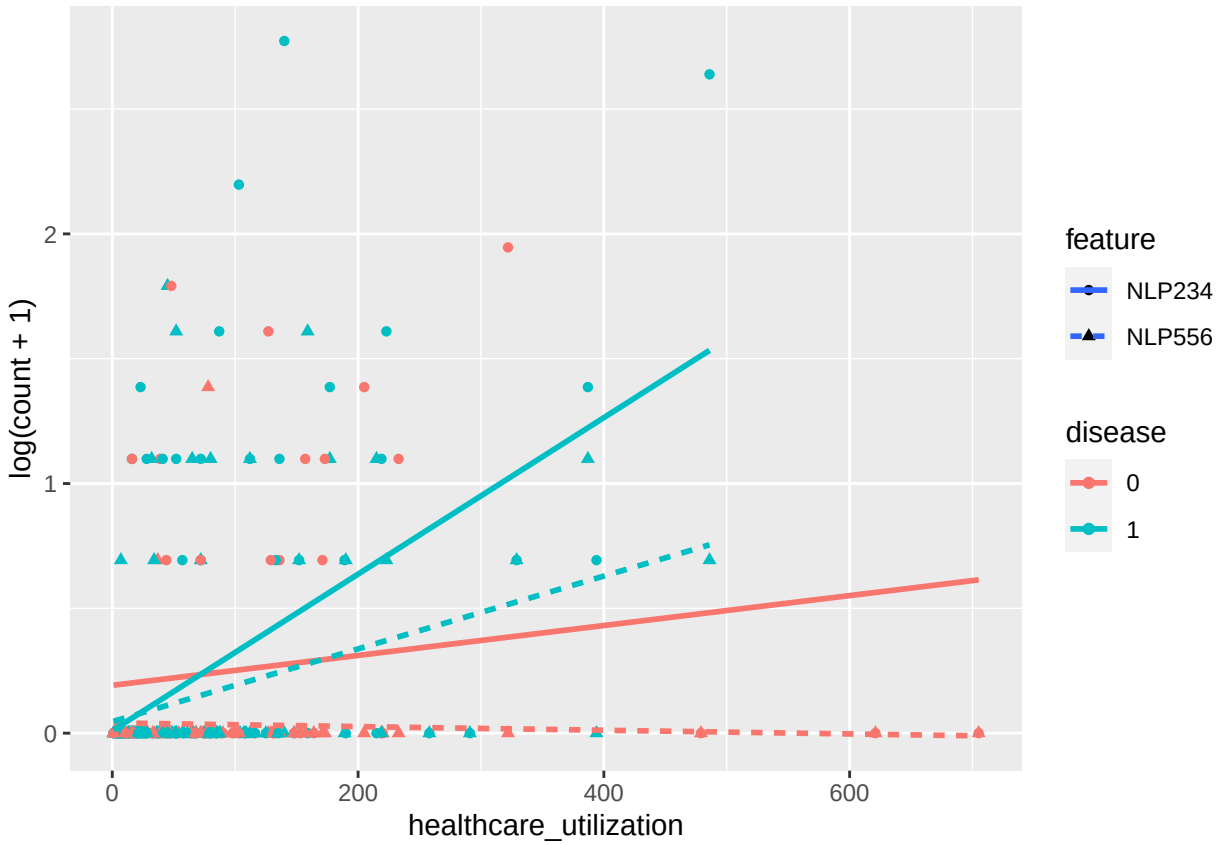


Healthcare utilization

"Healthcare_utilization" refers to total number of notes the patient has.

```
feature_index <- sample(c(3:ncol(ehr_data)), 2, replace = FALSE)

ehr_data %>%
  dplyr::select(label, healthcare_utilization, feature_index) %>%
  filter(!is.na(label)) %>%
  mutate(disease = factor(label)) %>%
  dplyr::select(-label) %>%
  pivot_longer(c(everything(), -disease, -healthcare_utilization),
               names_to = "feature", values_to = "count") %>%
  ggplot(aes(x = healthcare_utilization, y = log(count + 1), color = disease, shape = feature, linetype
  geom_point() +
  geom_smooth(method='lm', se = FALSE, size=1)
```



More healthcare utilization, more feature counts; but not necessary the patient get the disease.

Therefore, we need to adjust for the healthcare utilization when model fitting.

One option is to orthogonalize the healthcare utilization and other features.

Prepare the data for model fitting.

To do this, we first need to transform all the features as they are count variables.

Then orthogonalize other features X against healthcare utilization H, by performing a linear regression of X against H and taking the residual from the fitting to obtain the new X.

```
features$main_ICDNLP <- features$main_ICD + features$main_NLP
features <- log(features + 1)
```

```
# Features other than healthcare utilization.
other_features <- features[, -3]

# Orthogonalize.
orthognalized_features <- qr.resid(qr(cbind(1, features$healthcare_utilization)),
                                   as.matrix(other_features))
```

```
orthognalized_features <- data.frame(orthognalized_features)

ehr_data <- cbind(healthcare_utilization = features$healthcare_utilization,
```

```
                orthognalized_features,
                label = ehr_data$label,
                patient_id = ehr_data$patient_id) %>%
            dplyr::select(patient_id, label, main_ICD, main_NLP, main_ICDNLP,
                          healthcare_utilization, everything())
```

Save the data for module 2 and model fitting.

```
save(list = ls(), file = "../module2/environment.RData")
```