Module 1: Introduction

In this module, we will go through a publicly released dataset from the PheCAP R package to get hands-on experience with phenotyping. The goal of phenotyping is to infer a patients phenotype based on the information in their electronic health record (EHR).

```
# Packages required for this module.
packages <- c("tidyverse", "PheCAP", "corrplot", "ggplot2")

# Load packages, or install if missing.
package.check <- lapply(
      packages,
      FUN = function(x) {
        if (!require(x, character.only = TRUE)) {
          install.packages(x, dependencies = TRUE)
          library(x, character.only = TRUE)
        }
      }
    )
```

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

PheCAP

Information about the PheCAP package can be found here: https://celehs.github.io/PheCAP/.

This data mart is a random sample of patients in the Mass General Brigham (formerly Partner's Healthcare) EHR database who had at least one note of 500 characters and met an initial filter for coronary artery disease (CAD) defined as:

(i) $\geq$ 1 ICD9 code related to CAD (410.x, 411.x, 412.x, 413., 414.x)

or

(ii) $\geq$ 1 mention for any CAD related concepts (eg. CAD, CAD procedures, CAD biomarkers, positive stress test)

PheCAP data

```
data(ehr_data)
data <- PhecapData(ehr_data, "healthcare_utilization", "label", 75,
                   "patient_id", seed = 123)
data
```

```
## PheCAP Data
## Feature: 10000 observations of 587 variables
## Label: 119 yes, 62 no, 9819 missing
## Size of training samples: 106
## Size of validation samples: 75
```

The data contains 10,000 observations, but only 181 labeled examples split into a training and validation set.

Exploratory data analysis

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

- Labels: "label", whether the patient has CAD, **extracted from chart review by a clinician**
- Features:
  - "main_ICD", "main_NLP" refers to total number of billing codes or NLP mentions of the disease
  - "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, respectively

Missingness

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

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

There is no missing data, except what is in the label.

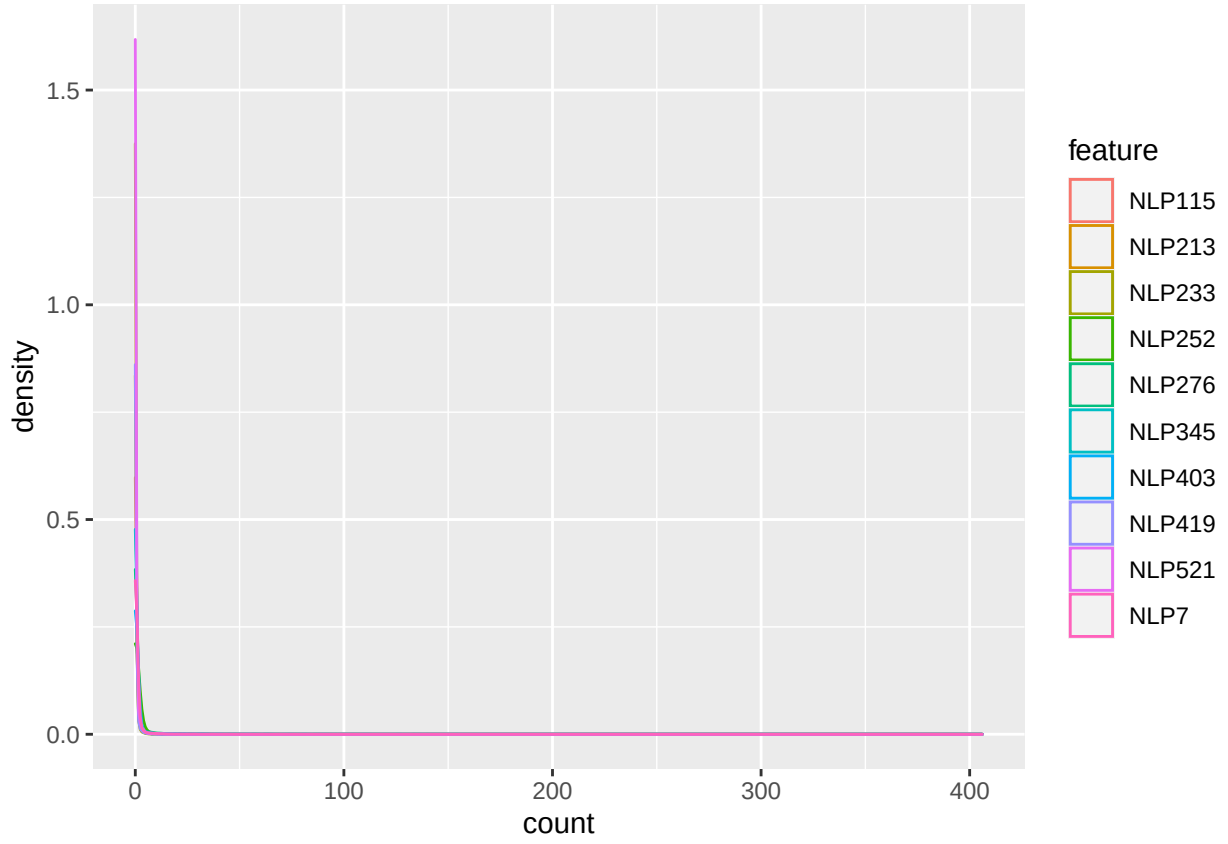
Prevalence of CAD

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

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

Distributions of the features

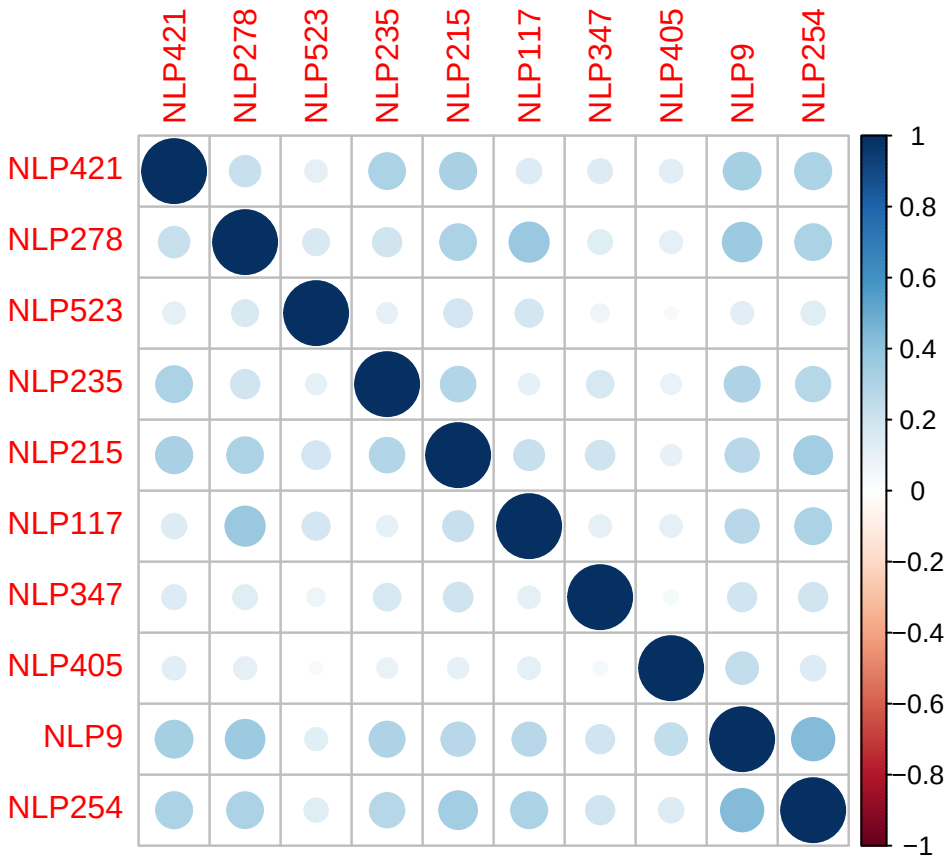
```
set.seed(99)
feature_index <- sample(c(3:ncol(ehr_data)), 10, replace = FALSE)
ehr_data[, feature_index] %>%
  pivot_longer(everything(), names_to = "feature", values_to = "count") %>%
  ggplot() +
  geom_density(aes(x = count, color = feature))
```



The features are highly skewed.

Correlation among the features

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

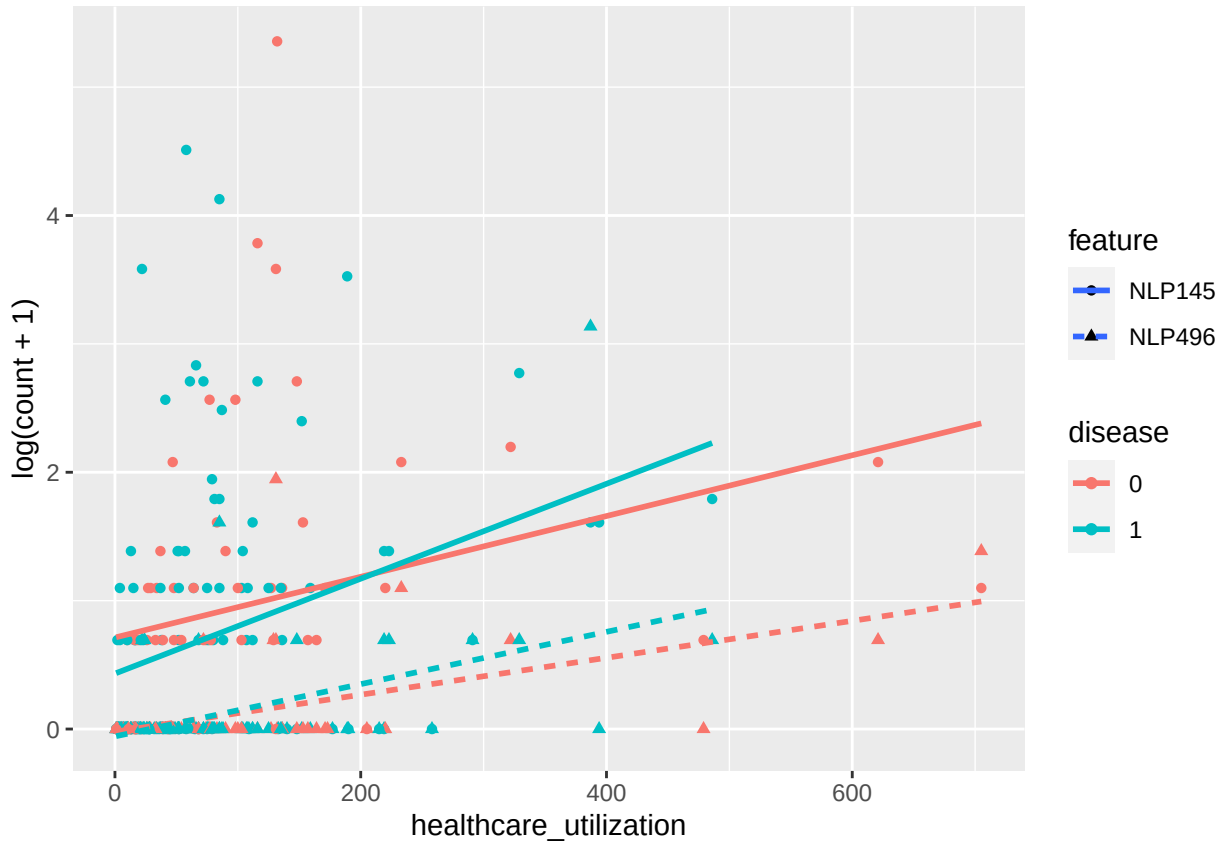


Healthcare utilization

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

```
surrogate_index <- sample(3:ncol(ehr_data), 2)

ehr_data %>%
  dplyr::select(label, healthcare_utilization, surrogate_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)
```



Increased healthcare utilization leads to higher features counts in both the cases and controls.

Prepare the data for model fitting.

We first transform all the features as they are count variables. We then orthogonalize the 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 feature for modeling.

```
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")
```