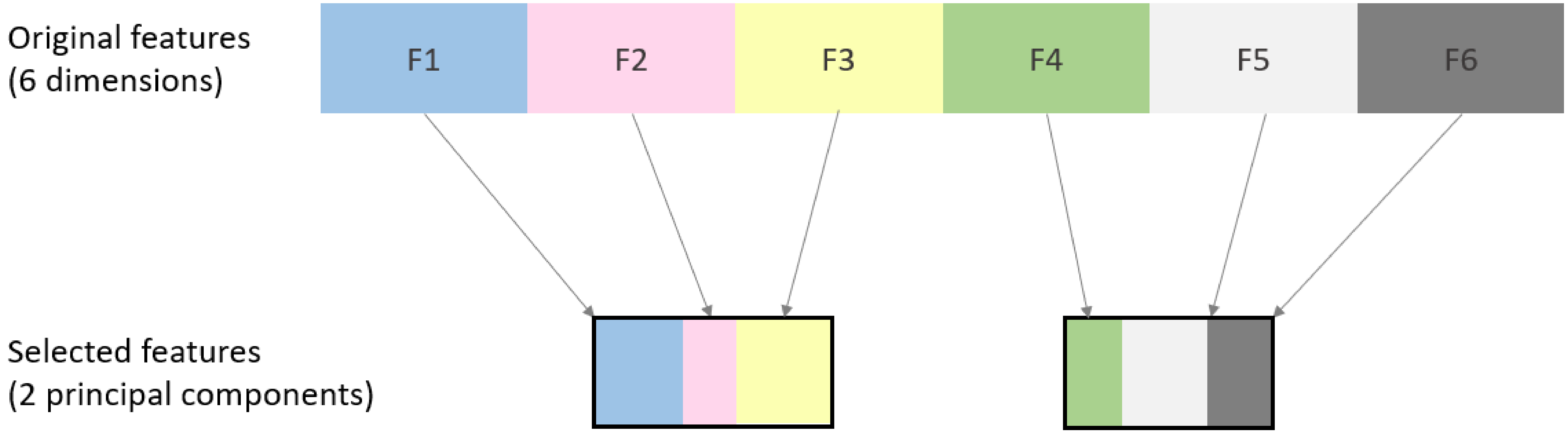
Foundations of feature extraction - principal components

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

Feature extraction review



Feature extraction review

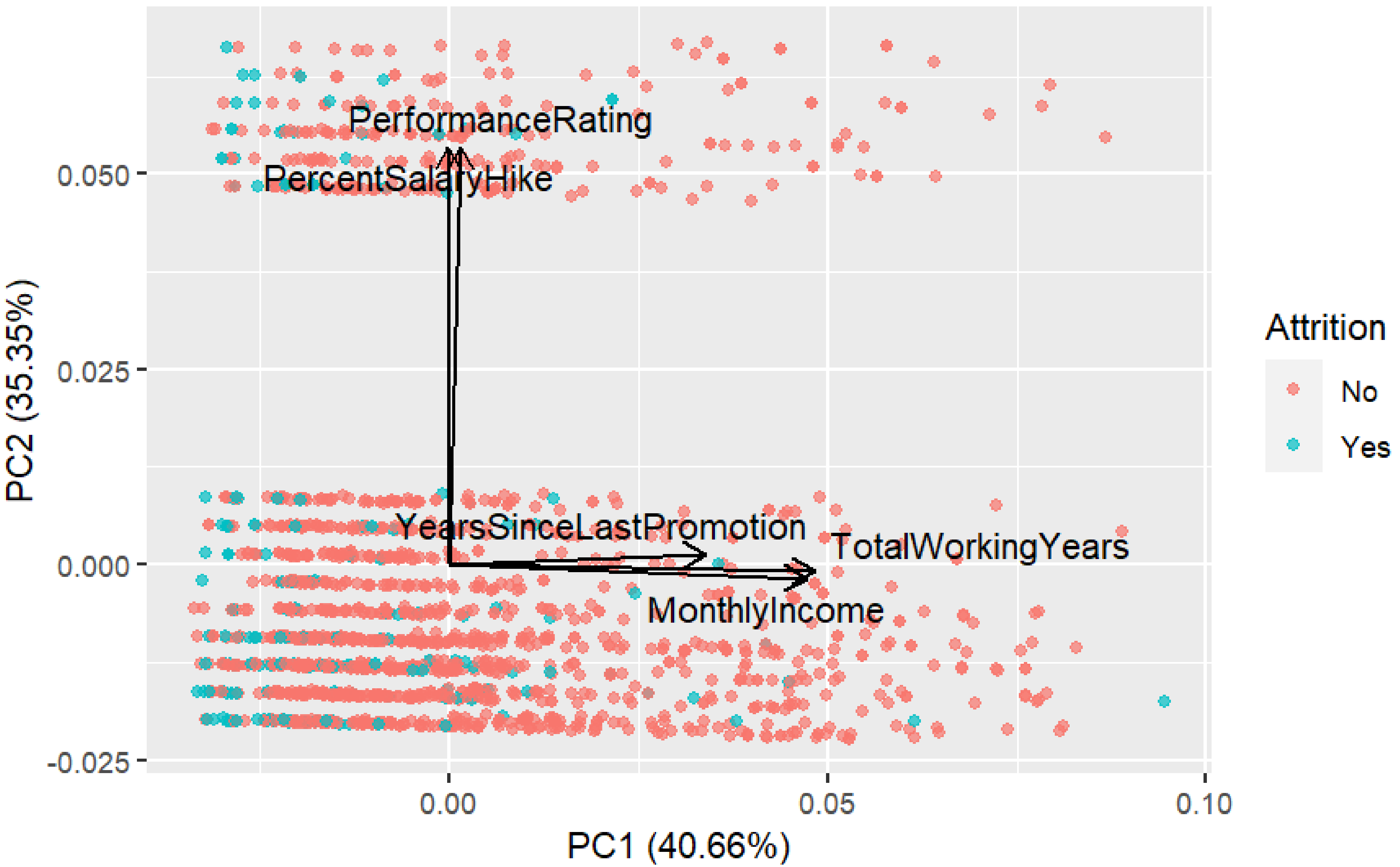
[1] Image Source: Daderot, CC0, via Wikimedia Commons

Salad recipe

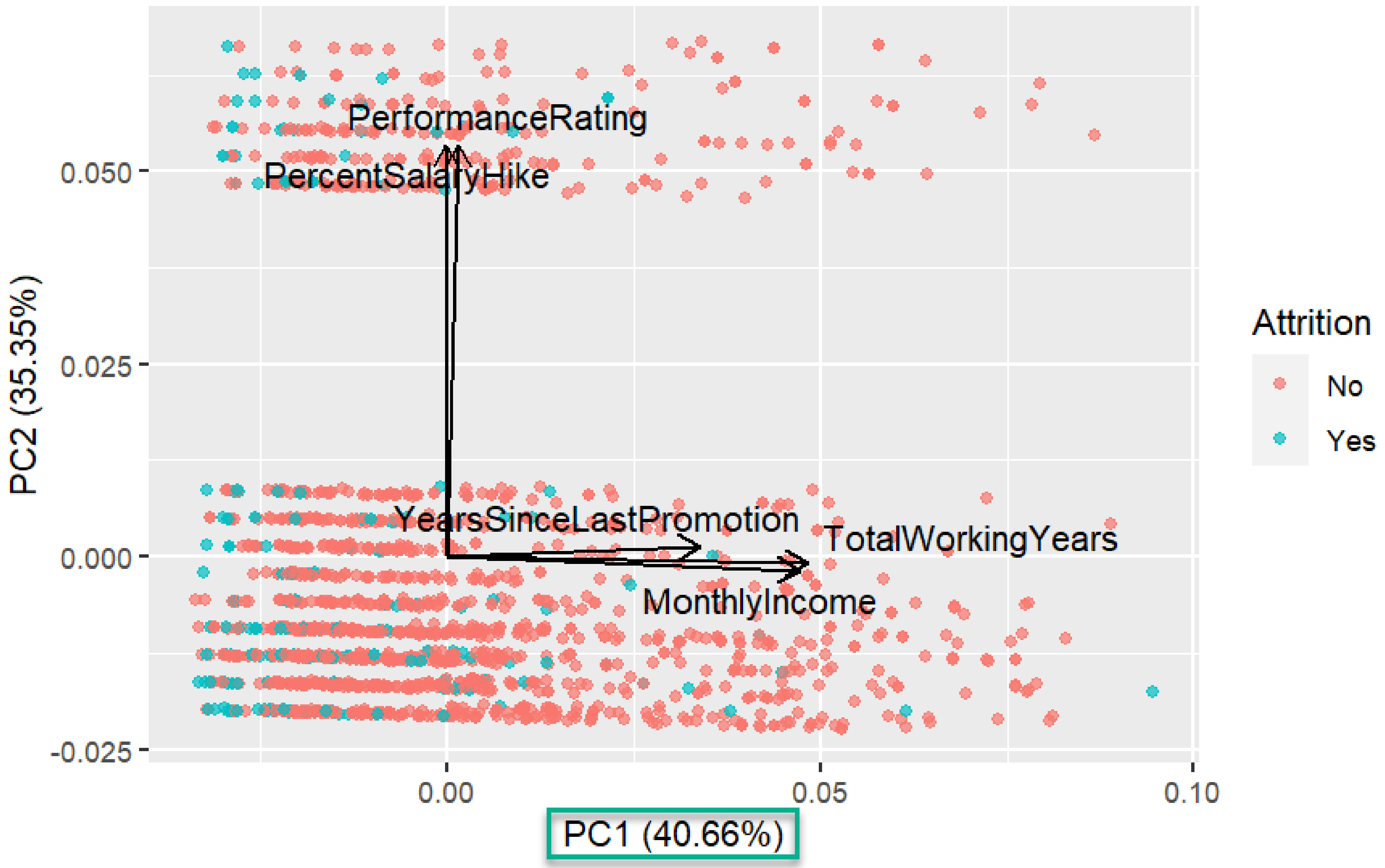
- 1 head of lettuce
- 3 carrots
- 2 tomatoes
- 1 cucumber

Do not use the whole plant, just the best parts

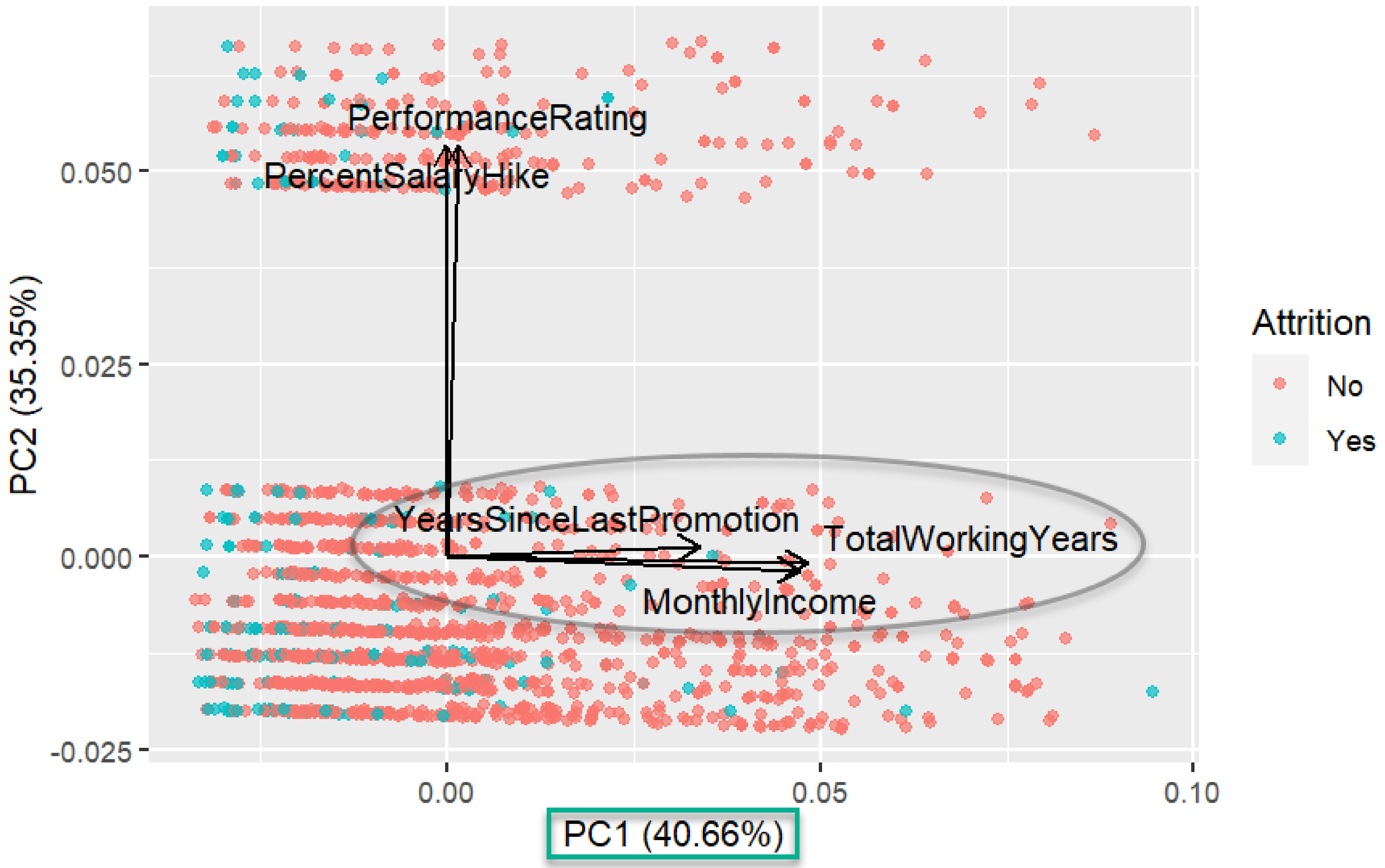
PCA plot


PerformanceRating
PercentSalaryHike
YearsSinceLastPromotion
TotalWorkingYears
MonthlyIncome
0.050
0.025
0.000
-0.025
PC2 (35.35%)
0.00
0.05
0.10
PC1 (40.66%)
Attrition
No
Yes

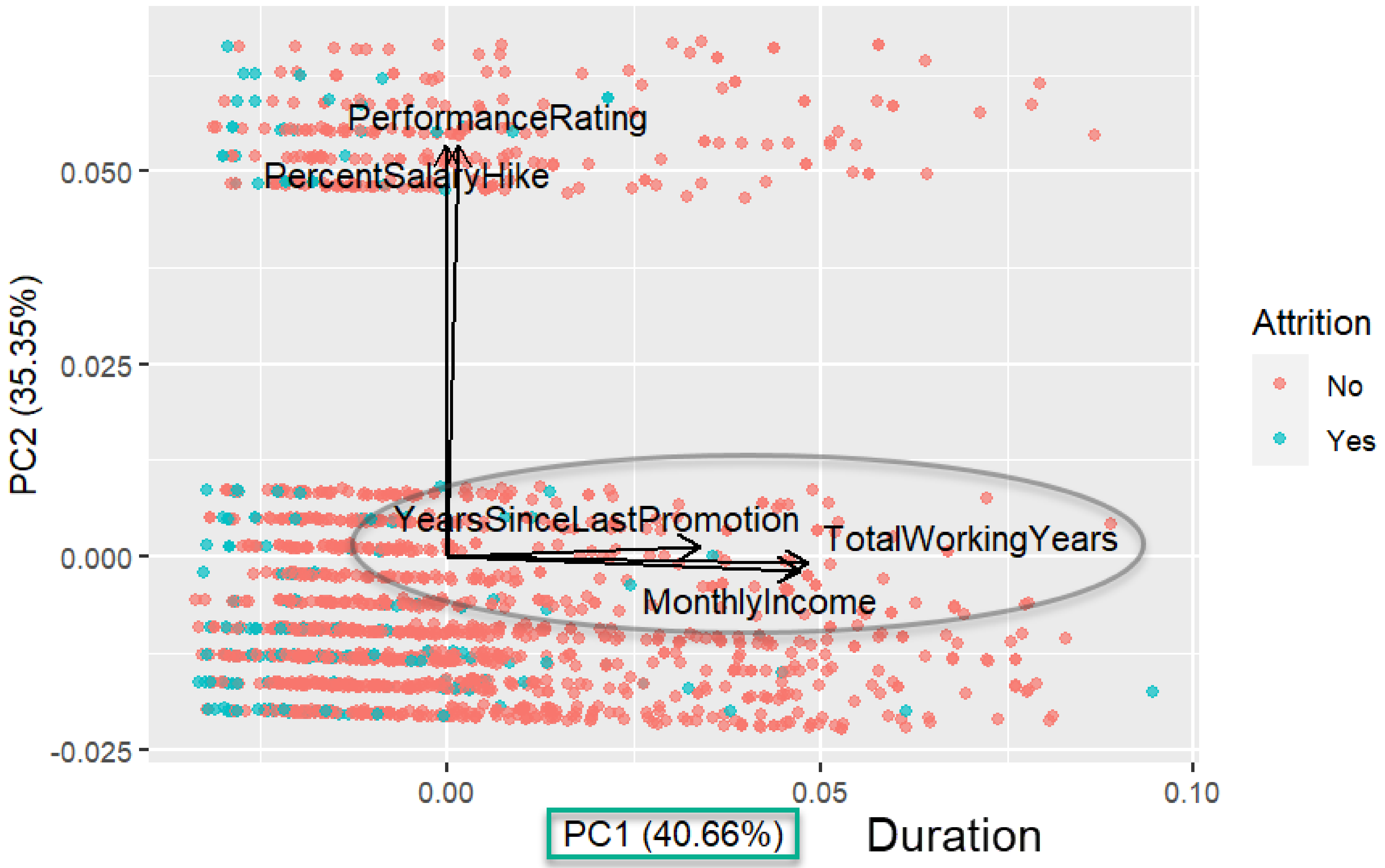
Principal component 1



PC1: feature vectors

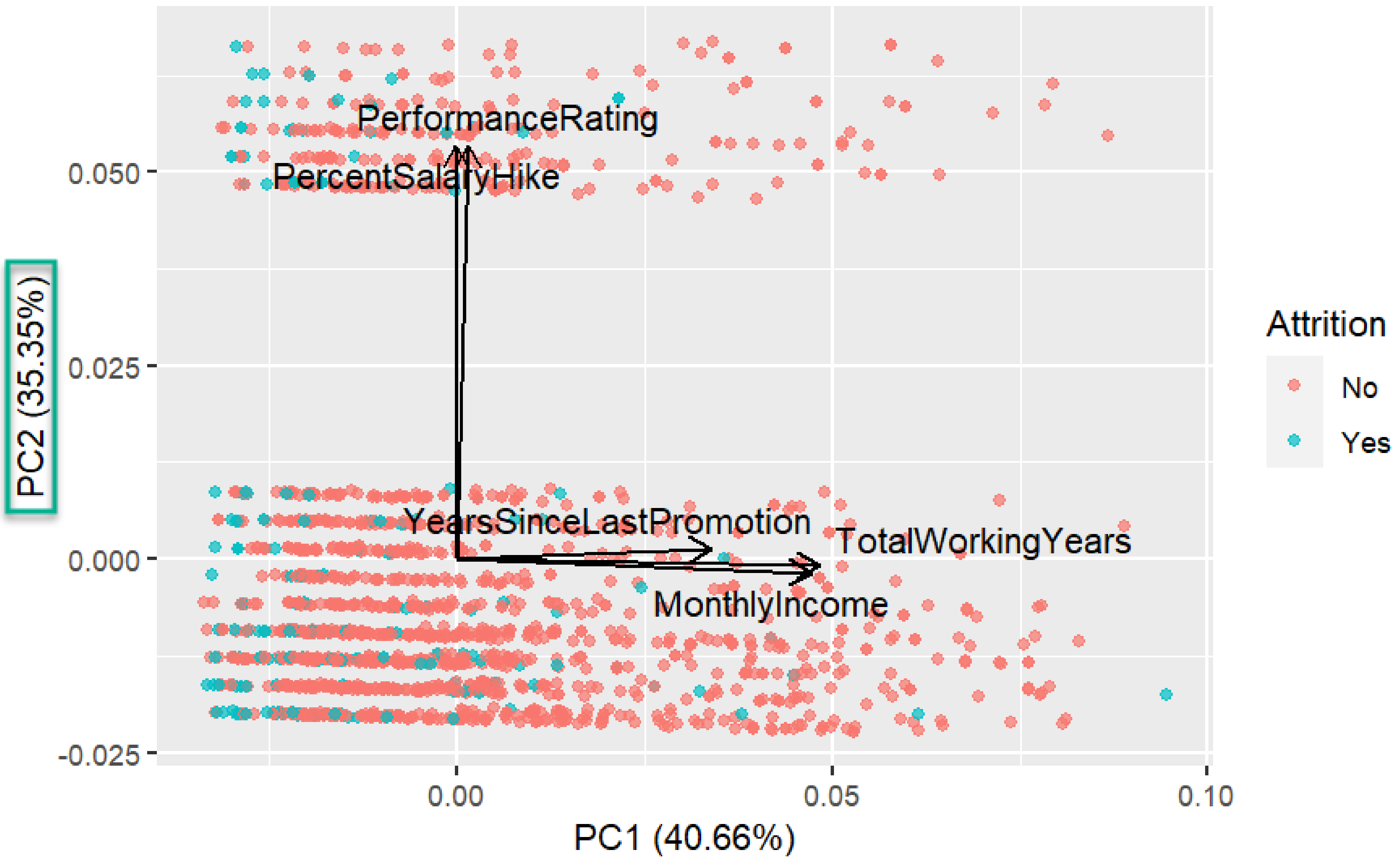


PC1: name

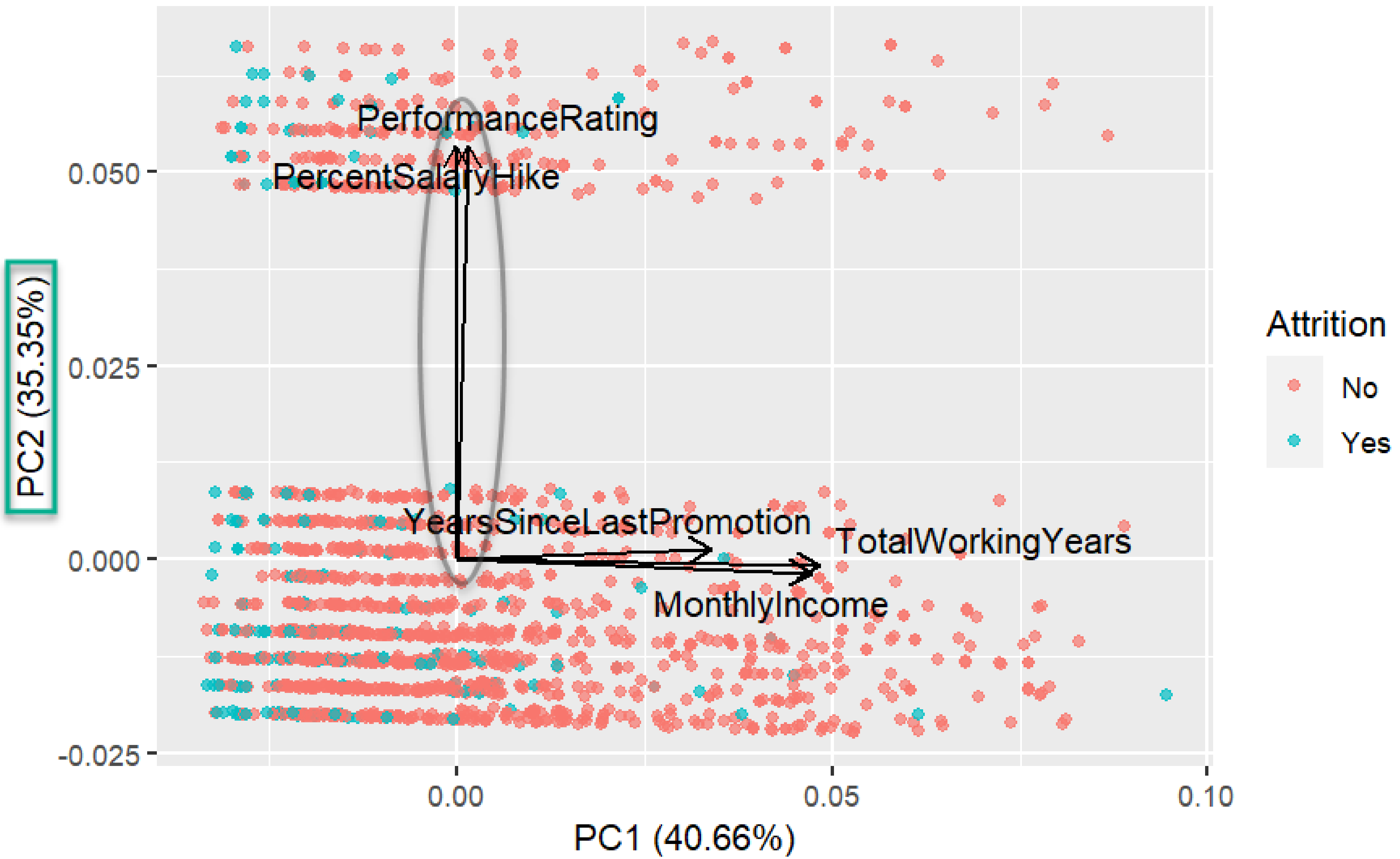


PerformanceRating
PercentSalaryHike
YearsSinceLastPromotion
TotalWorkingYears
MonthlyIncome
0.050
0.025
0.000
-0.025
0.00
0.05
0.10
PC2 (35.35%)
PC1 (40.66%)
Duration
Attrition
No
Yes

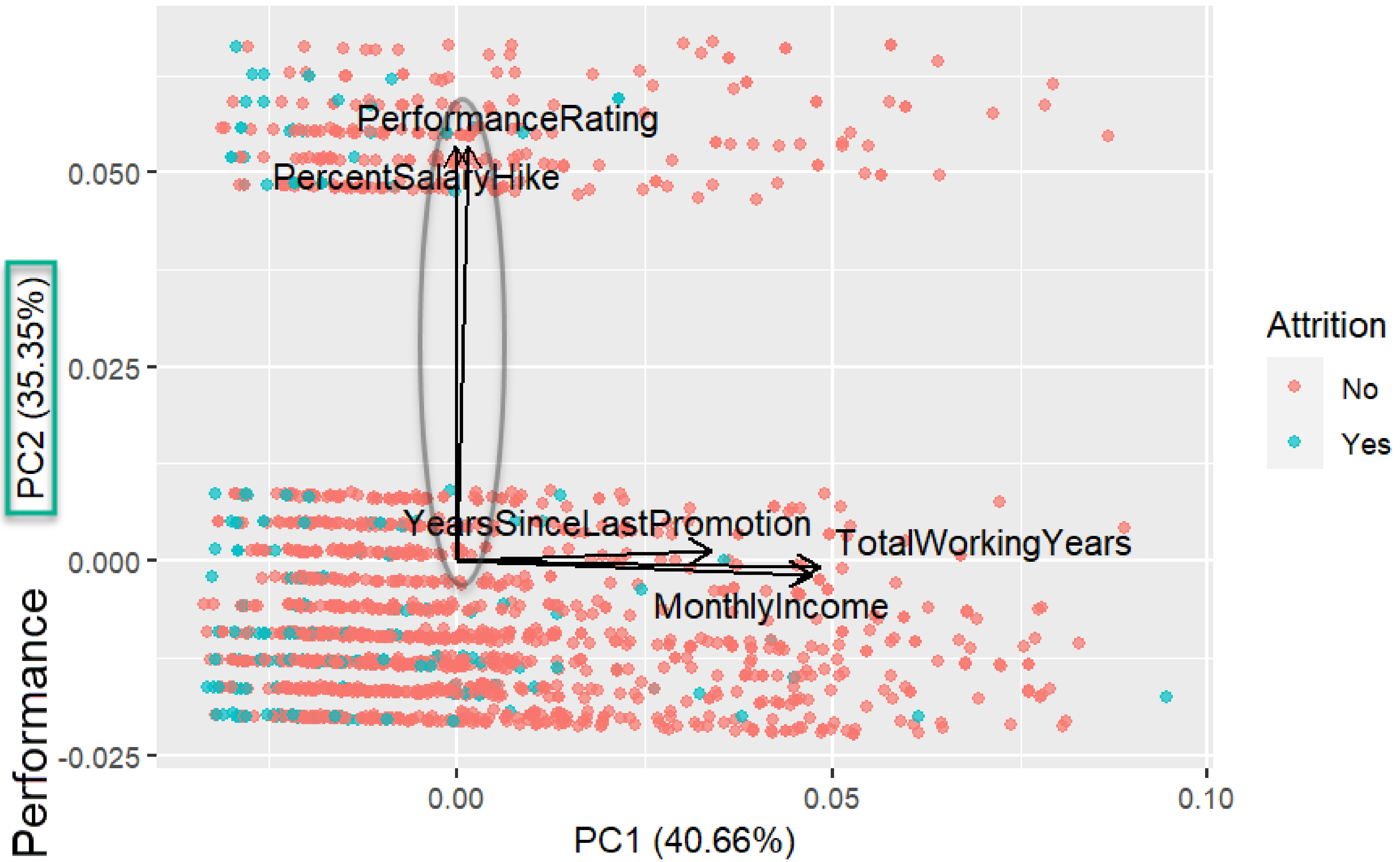
Principal component 2



PC2: feature vectors



PC2: name


PerformanceRating
PercentSalaryHike
YearsSinceLastPromotion
TotalWorkingYears
MonthlyIncome
0.050
0.025
0.000
-0.025
0.00
0.05
0.10
PC1 (40.66%)
PC2 (35.35%)
Performance
Attrition
No
Yes

Code for a PCA plot

```
library(ggfortify)
pca_res <- prcomp(attrition_df %>% select(-Attrition), scale. = TRUE)
autoplot(pca_res,
         data = attrition_df,
         colour = "Attrition",
         alpha = 0.7,
         loadings = TRUE,
         loadings.label = TRUE,
         loadings.colour = "black",
         loadings.label.colour = "black",
         loadings.label.repel = TRUE)
```

Let's practice!

DIMENSIONALITY REDUCTION IN R

Principal Component Analysis (PCA)

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

Performing a PCA

```
pca_res <- prcomp(attrition_df %>% select(-Attrition), scale. = TRUE)
summary(pca_res)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5
Standard deviation     1.4259 1.3295 0.8618 0.48401 0.47138
Proportion of Variance 0.4067 0.3535 0.1485 0.04685 0.04444
Cumulative Proportion  0.4067 0.7602 0.9087 0.95556 1.00000
```

PC loadings

```
pca_res
```

```
Standard deviations (1, .., p=5):
[1] 1.43 1.33 0.86 0.48 0.47

Rotation (n x k) = (5 x 5):
                            PC1    PC2     PC3    PC4    PC5
MonthlyIncome            0.6244 -0.024  0.3665 -0.280 -0.630
TotalWorkingYears        0.6390 -0.011  0.2674  0.293  0.659
YearsSinceLastPromotion  0.4488  0.018 -0.8902 -0.061 -0.047
PercentSalaryHike       -0.0018  0.707  0.0426 -0.647  0.284
PerformanceRating        0.0210  0.707 -0.0033  0.643 -0.294
```

PC loadings

	PC1	PC2
MonthlyIncome	0.6244	-0.024
TotalWorkingYears	0.6390	-0.011
YearsSinceLastPromotion	0.4488	0.018
PercentSalaryHike	-0.0018	0.707
PerformanceRating	0.0210	0.707

PC loadings

	PC1	PC2
MonthlyIncome	0.6244	-0.024
TotalWorkingYears	0.6390	-0.011
YearsSinceLastPromotion	0.4488	0.018
PercentSalaryHike	-0.0018	0.707
PerformanceRating	0.0210	0.707

PC loadings

	PC1	PC2
MonthlyIncome	0.6244	-0.024
TotalWorkingYears	0.6390	-0.011
YearsSinceLastPromotion	0.4488	0.018
PercentSalaryHike	-0.0018	0.707
PerformanceRating	0.0210	0.707

PCA with tidymodels

```
pca_recipe <- recipe(Attrition ~ . , data = train) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_pca(all_numeric_predictors(), num_comp = 2)

attrition_fit <- workflow(preprocessor = pca_recipe, spec = logistic_reg()) %>%
  fit(train)

attrition_pred_df <- predict(attrition_fit, test) %>%
  bind_cols(test %>% select(Attrition))

f_meas(attrition_pred_df, Attrition, .pred_class)
```

See the PCs in the model details

```
attrition_fit
```

```
Call:  stats::glm(formula = ..y ~ ., family = stats::binomial, data = data)

Coefficients:
(Intercept)          PC1          PC2
   -2.42067      0.80493     -0.03429

Degrees of Freedom: 1339 Total (i.e. Null);  1337 Residual
Null Deviance:        951.8
Residual Deviance: 870.6     AIC: 876.6
```

Let's practice!

DIMENSIONALITY REDUCTION IN R

t-Distributed Stochastic Neighborhood Embedding (t-SNE)

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

PCA	t-SNE
Linear	Non-linear

PCA	t-SNE
Linear	Non-linear
Deterministic	Non-deterministic (random)

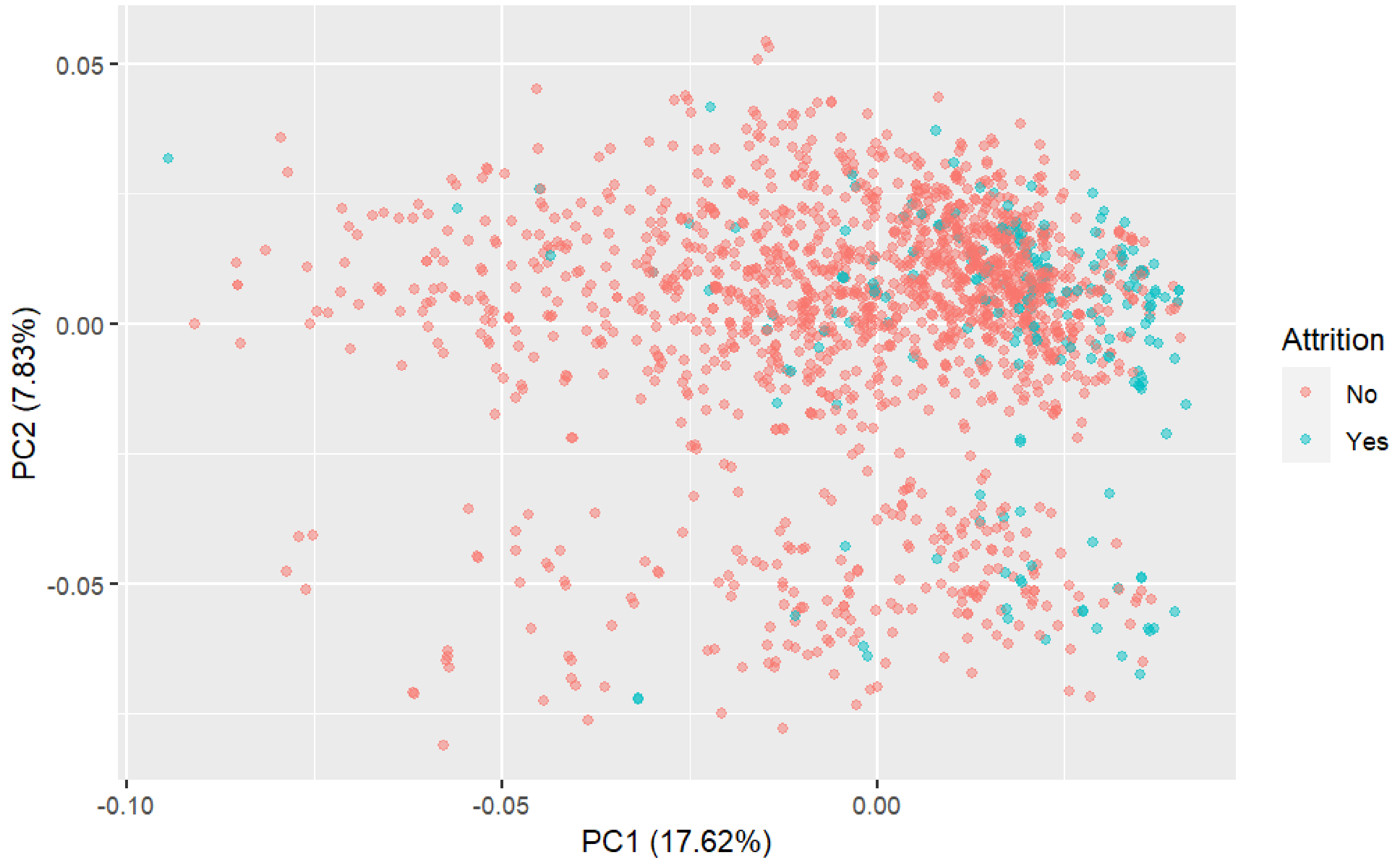
PCA	t-SNE
Linear	Non-linear
Deterministic	Non-deterministic (random)
Does not handle outliers well	Handles outliers well

PCA	t-SNE
Linear	Non-linear
Deterministic	Non-deterministic (random)
Does not handle outliers well	Handles outliers well
Computationally cheap	Computationally expensive

PCA	t-SNE
Linear	Non-linear
Deterministic	Non-deterministic (random)
Does not handle outliers well	Handles outliers well
Computationally cheap	Computationally expensive
No hyperparameters	Hyperparameters

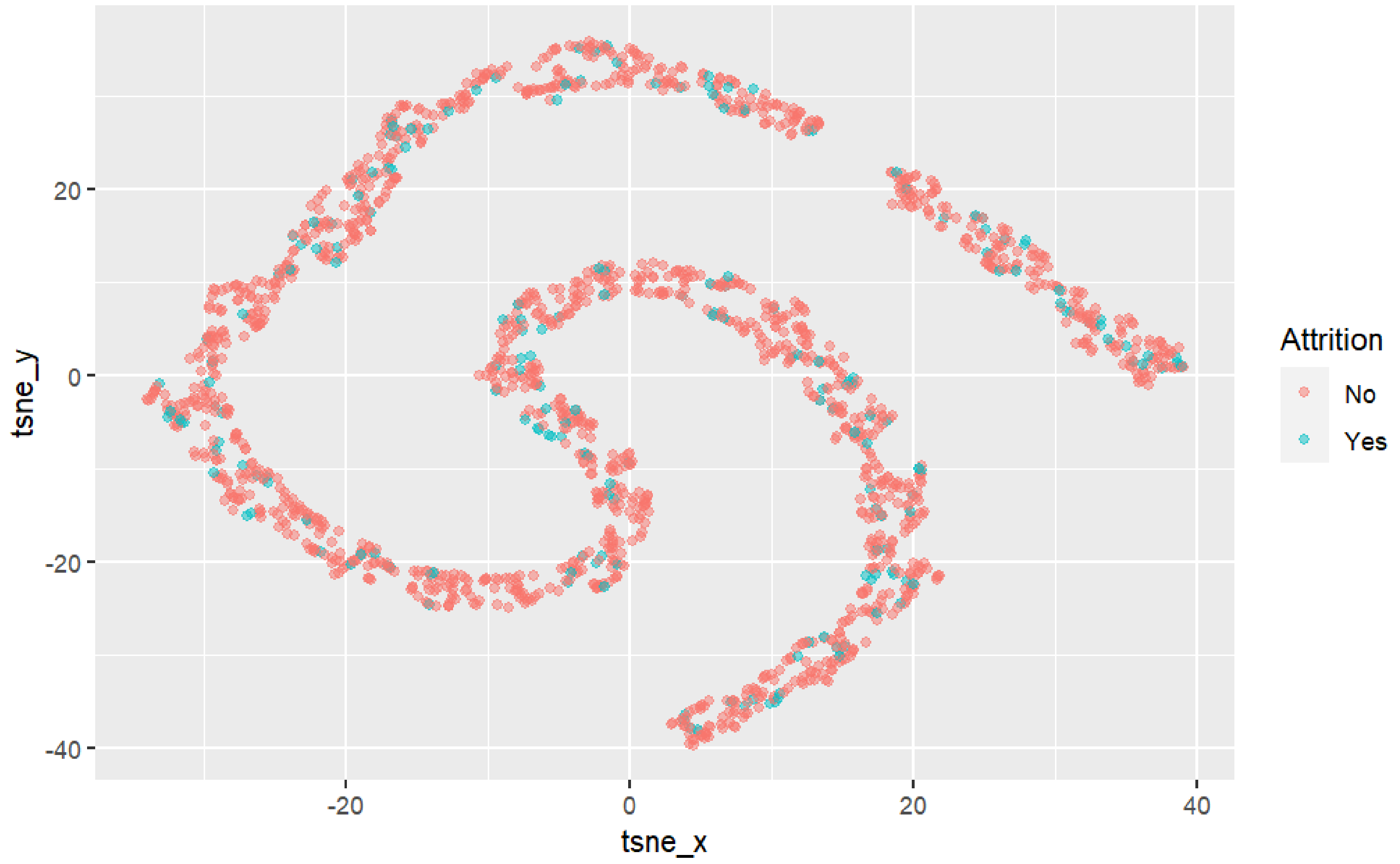
Plotting PCA and t-SNE

PCA



Preserves global structure

t-SNE

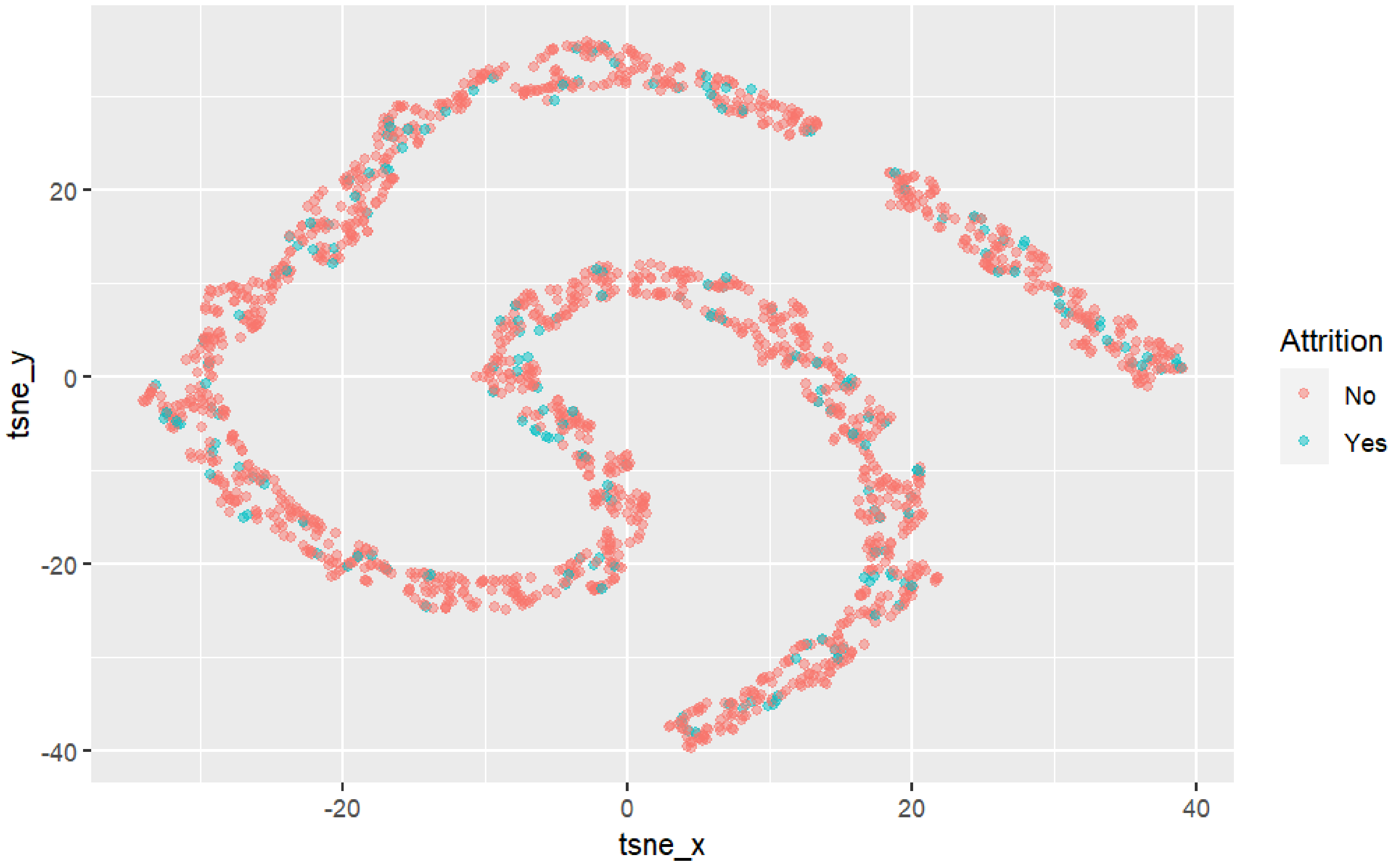


Preserves local structure (keeps neighbors next to each other)

t-SNE hyperparameters

- **Perplexity** - determines the number of nearest neighbors considered
- **Learning rate** - rate the weights of the neural network are adjusted
- **Iterations** - number of backpropogation iterations

t-SNE



t-SNE in R

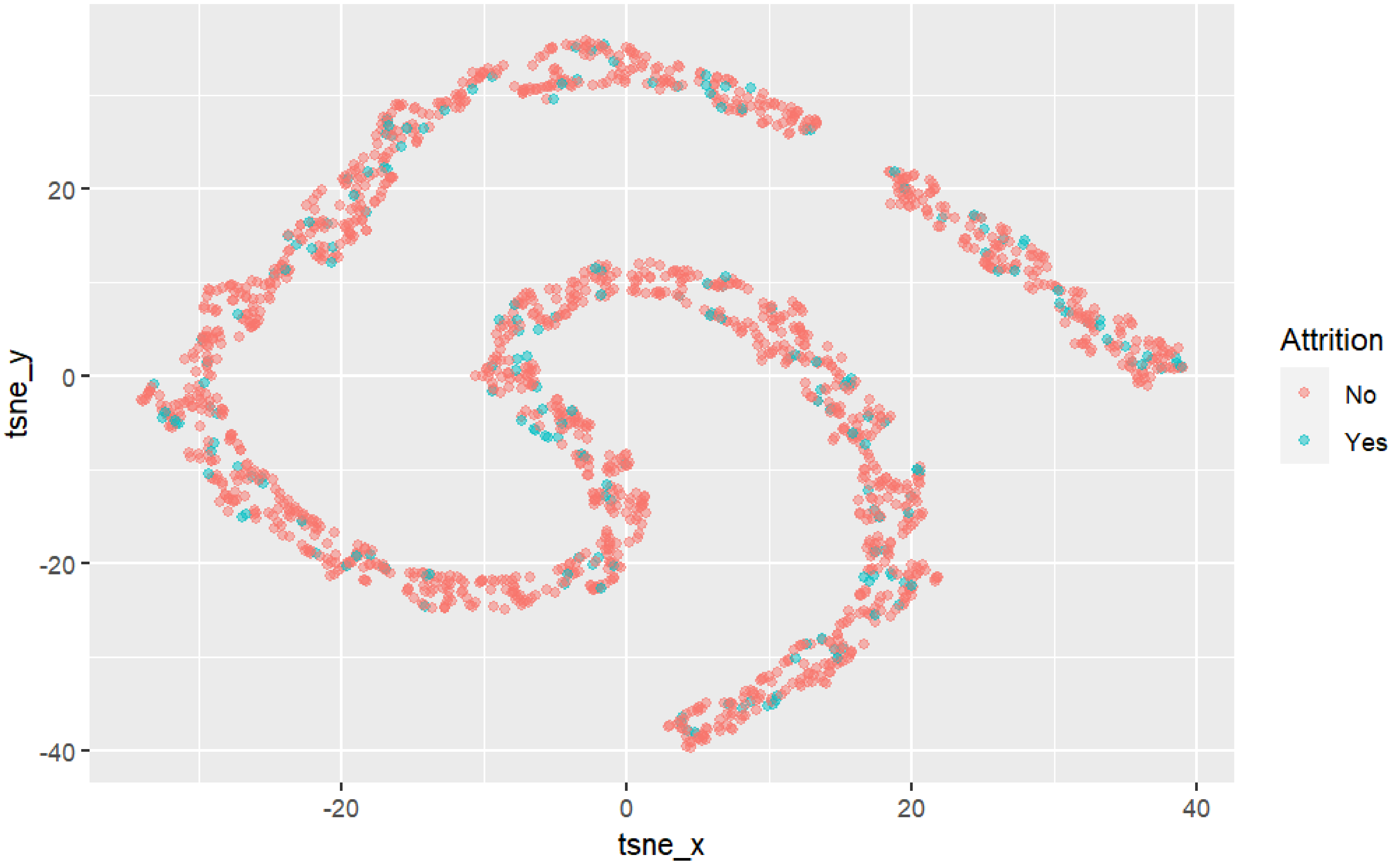
```
library(Rtsne)

set.seed(1234)
tsne <- Rtsne(attrition_df %>% select(-Attrition))

tsne_df <- attrition_df %>%
  bind_cols(tsne_x = tsne$Y[,1], tsne_y = tsne$Y[,2])

tsne_df %>%
  ggplot(aes(x = tsne_x, y = tsne_y, color = Attrition)) +
  geom_point(alpha = 0.5)
```

t-SNE plot



Let's practice!

DIMENSIONALITY REDUCTION IN R

Uniform Manifold Approximation and Projection (UMAP)

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

PCA, t-SNE, and UMAP

PCA	t-SNE	UMAP
Linear	Non-linear	Non-linear

PCA, t-SNE, and UMAP

PCA	t-SNE	UMAP
Linear	Non-linear	Non-linear
Deterministic	Non-deterministic	Non-deterministic

PCA, t-SNE, and UMAP

PCA	t-SNE	UMAP
Linear	Non-linear	Non-linear
Deterministic	Non-deterministic	Non-deterministic
Computationally cheap	Computationally expensive	Computationally efficient

PCA, t-SNE, and UMAP

PCA	t-SNE	UMAP
Linear	Non-linear	Non-linear
Deterministic	Non-deterministic	Non-deterministic
Computationally cheap	Computationally expensive	Computationally efficient
Preserves global structure	Preserves local structure	Preserves local and global structure

PCA, t-SNE, and UMAP

PCA	t-SNE	UMAP
Linear	Non-linear	Non-linear
Deterministic	Non-deterministic	Non-deterministic
Computationally cheap	Computationally expensive	Computationally efficient
Preserves global structure	Preserves local structure	Preserves local and global structure
No hyperparameters	Hyperparameters	Hyperparameters

UMAP has similar hyperparameters that can be tuned.

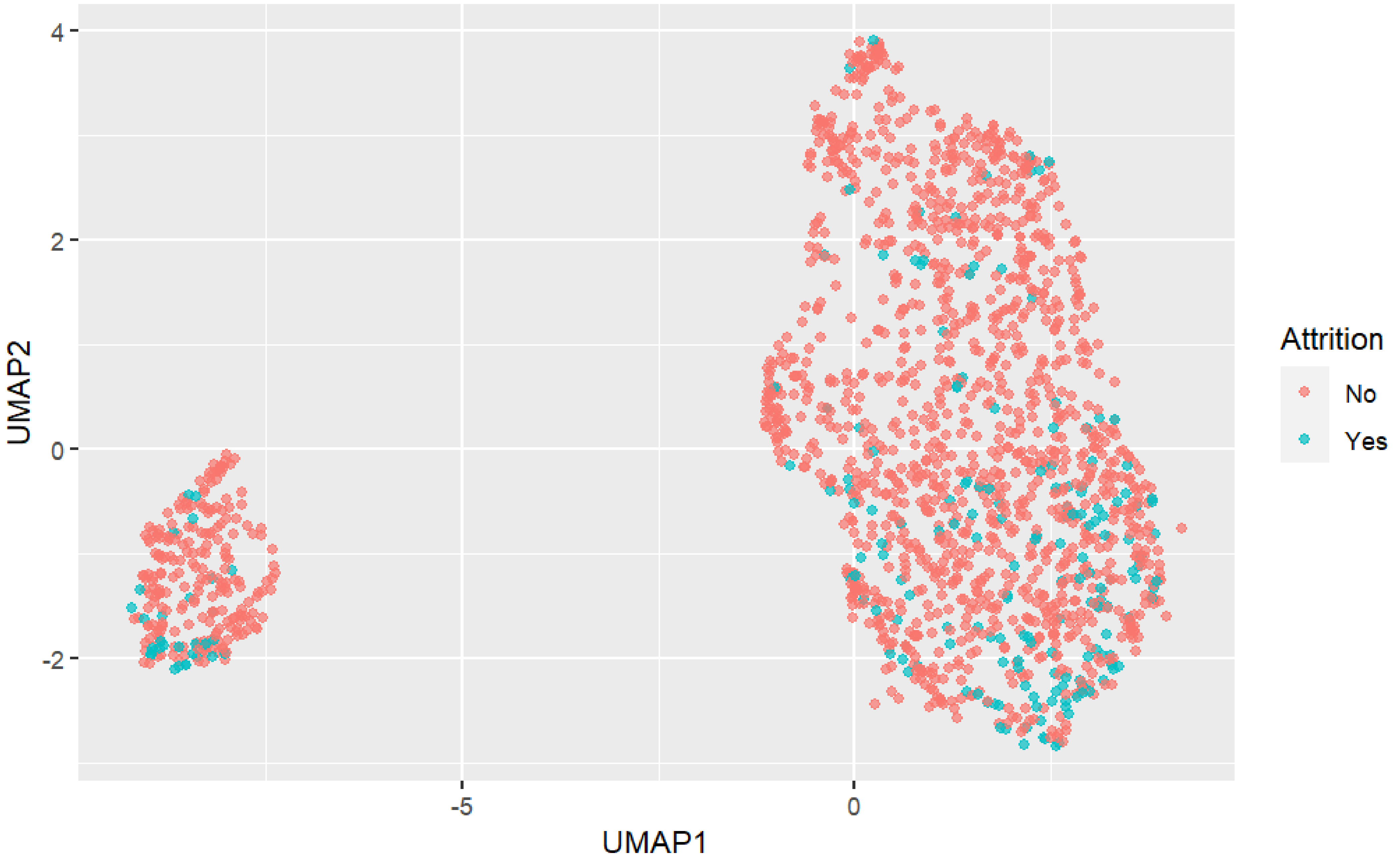
UMAP plot

```
library(embed)

set.seed(1234)
umap_df <- recipe(Attrition ~ ., data = attrition_df) %>%
  step_normalize(all_predictors()) %>%
  step_umap(all_predictors(), num_comp = 2) %>%
  prep() %>%
  juice()

umap_df %>%
  ggplot(aes(x = UMAP1, y = UMAP2, color = Attrition)) +
  geom_point(alpha = 0.7)
```

UMAP: employee attrition



UMAP in tidymodels

Create recipe

```
umap_recipe <-  recipe(Attrition ~ ., data = train) %>%
  step_normalize(all_predictors()) %>%
  step_umap(all_predictors(), num_comp = 4)
```

Create model spec

```
umap_lr_model <- linear_reg()
```

UMAP in tidymodels

Create workflow

```
umap_lr_workflow <-  workflow() %>%
  add_recipe(umap_recipe) %>%
  add_model(umap_lr_model)
```

Fit the workflow

```
umap_lr_fit <- umap_lr_workflow %>%
  fit(data = train)
```

UMAP in tidymodels

Evaluate the model

```
predict_umap_df <- test %>%
  bind_cols(predict = predict(umap_lr_fit, test))

rmse(predict_umap_df, Attrition, .pred_class)
```

Let's practice!

DIMENSIONALITY REDUCTION IN R

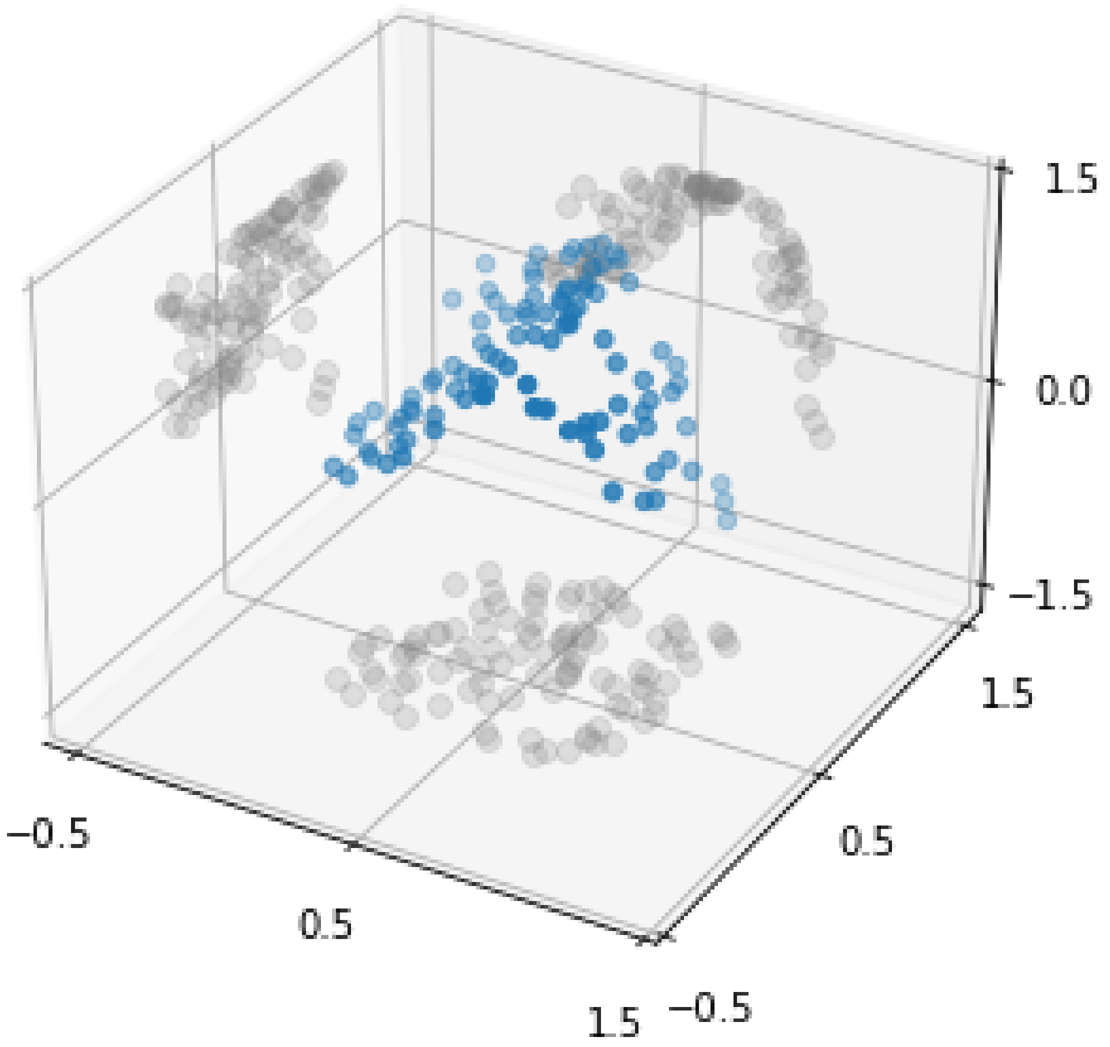
Wrap up

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

Chapter 1 - Dimensionality reduction, feature information

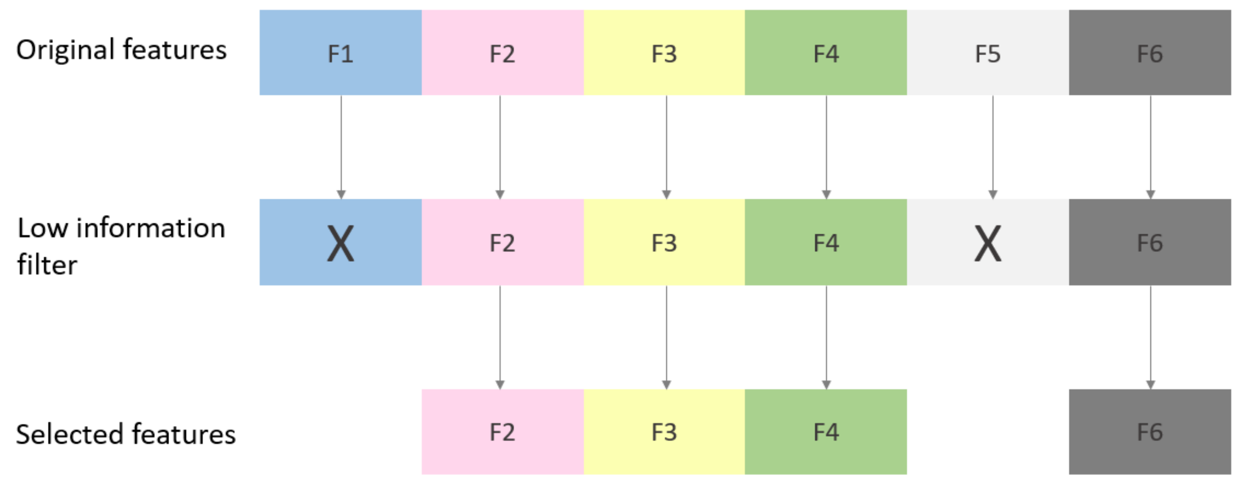
- Information - missing values, low variance, and correlation
- Information gain and feature importance
- Curse of dimensionality



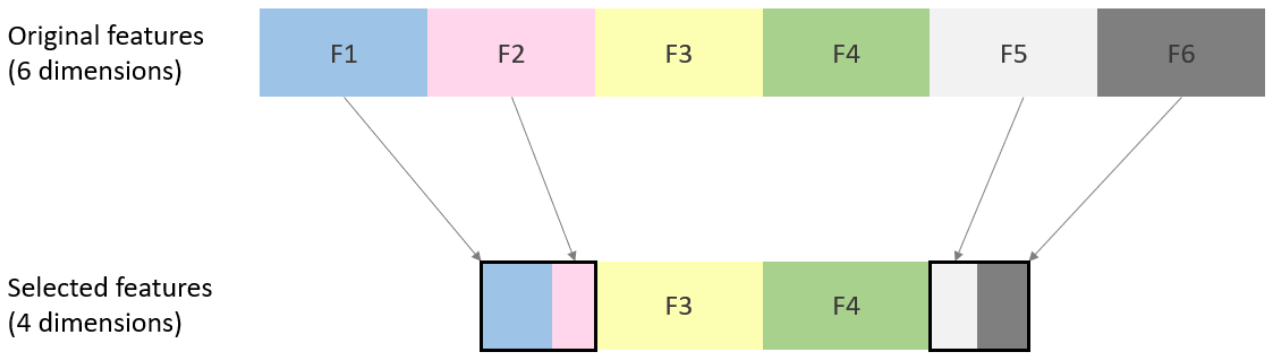
Chapter 2 - Unsupervised feature selection

- Feature selection vs. feature extraction
- Unsupervised feature selection:
  - missing value ratio filter
  - low-variance filter
  - correlation filter
- `tidymodels` recipe steps

Feature Selection

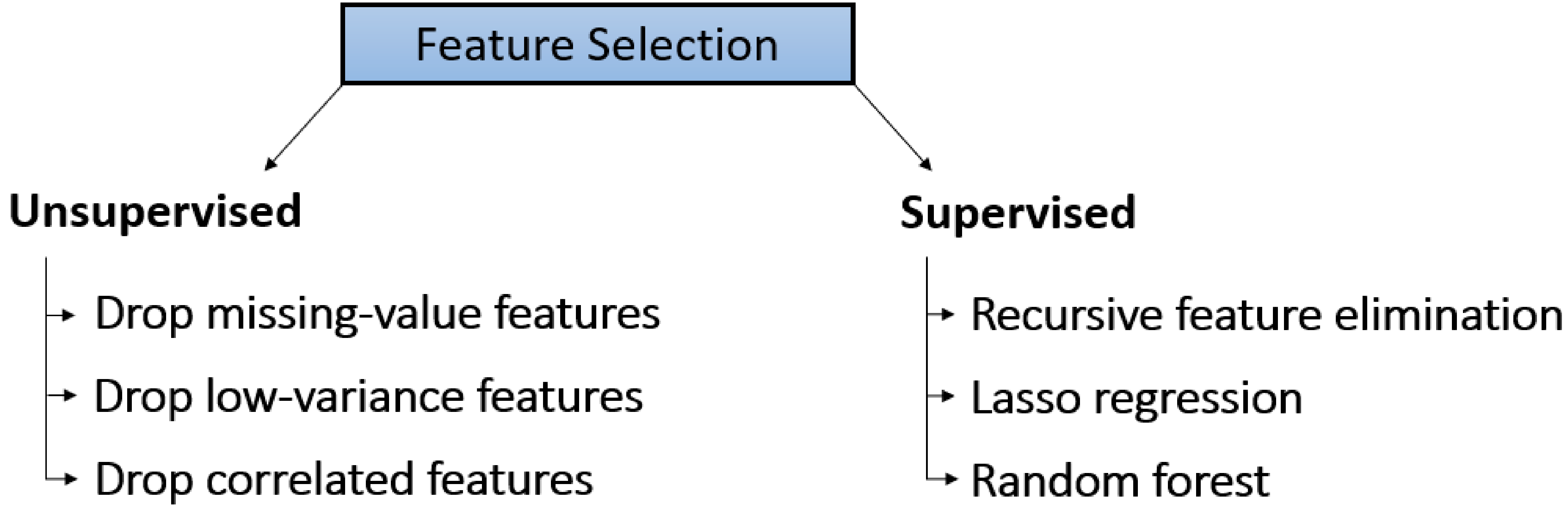


Feature Extraction



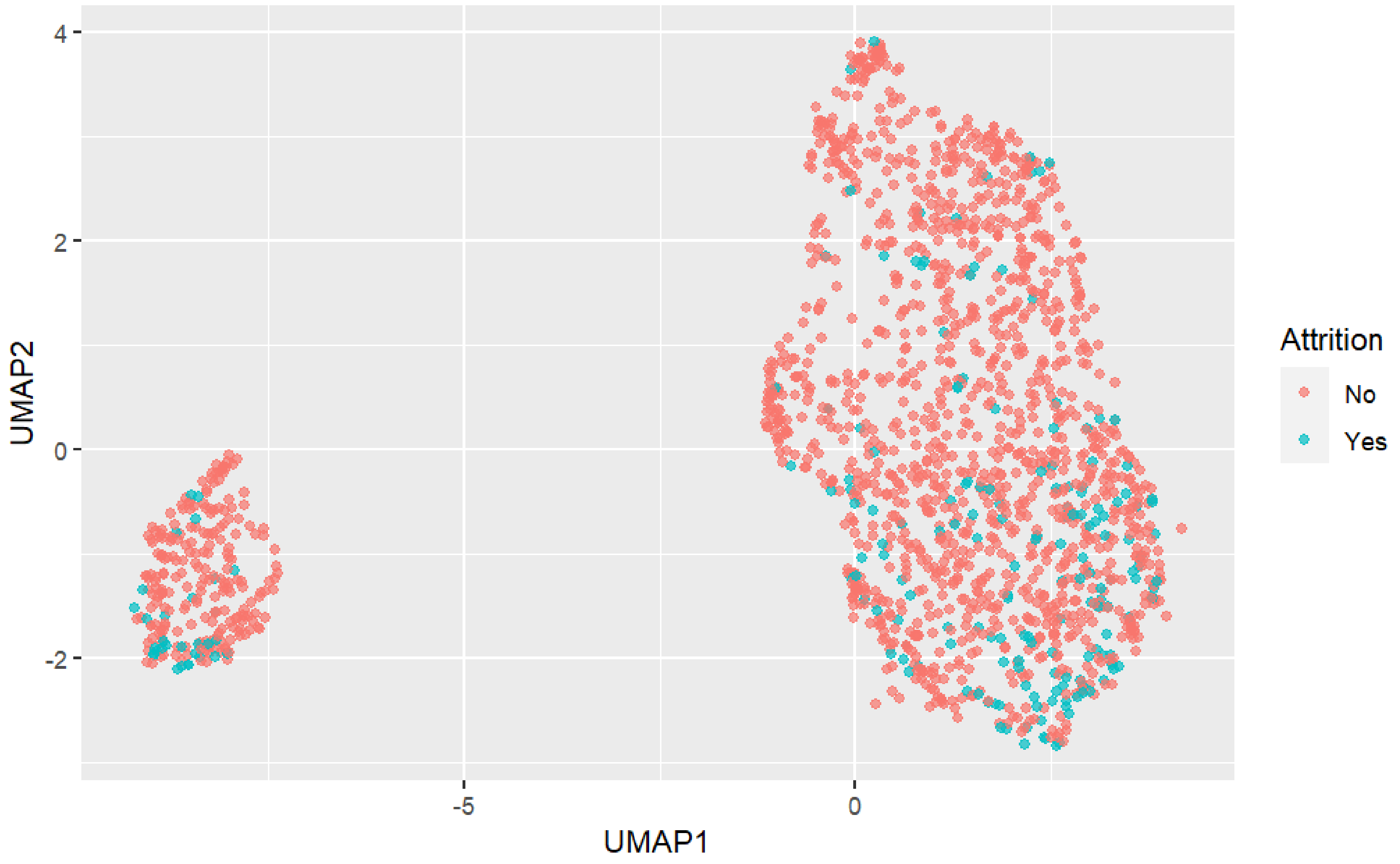
Chapter 3 - Supervised feature selection

- Reviewed model building with `tidymodels`
- Supervised feature selection methods: lasso regression, random forest
- Evaluated reduced model performance



Chapter 4 - Feature extraction

- Principal components and feature vectors
- Principal component analysis
- t-SNE
- UMAP



Congratulations!

DIMENSIONALITY REDUCTION IN R