Feature selection vs. feature extraction

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

Approaches to dimensionality reduction

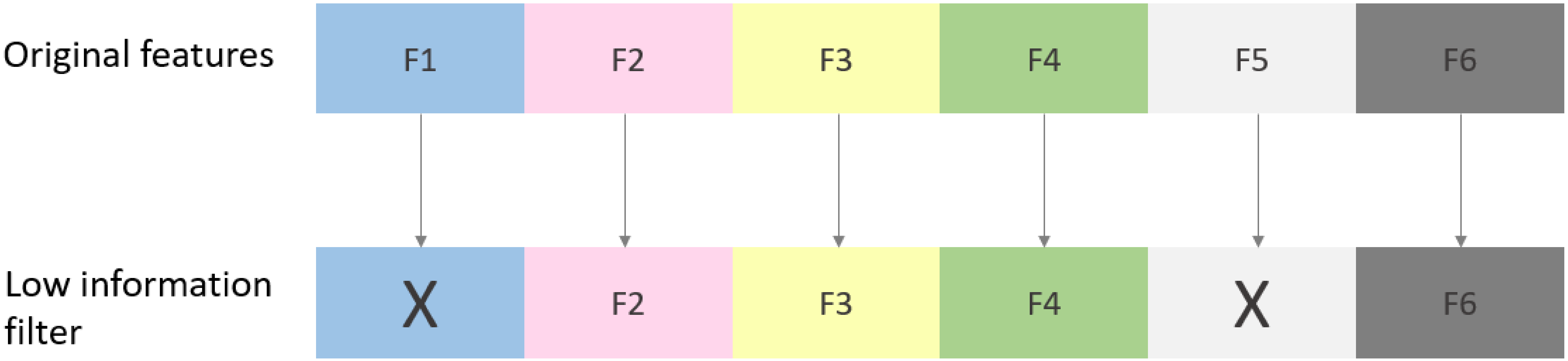
- **Feature selection** like pulling weeds
- **Feature extraction** like making a salad

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

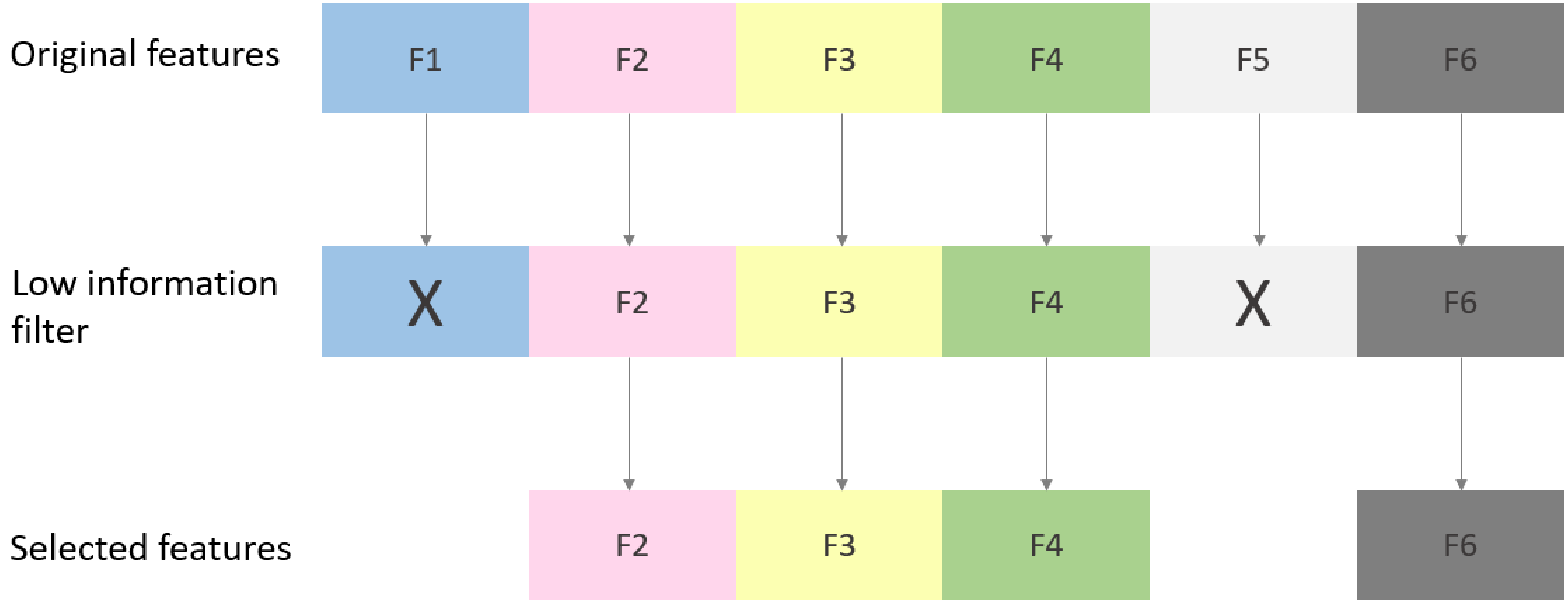
Feature selection

Original features	F1	F2	F3	F4	F5	F6

Feature selection



Feature selection



Example credit data

```
credit_df %>% head(n=5)
```

```
  annual_income num_bank_accounts num_credit_card outstanding_debt credit_history_months
          <dbl>             <dbl>           <dbl>            <dbl>                 <dbl>
1        87630.                 2               5             526.                   286
2        16574.                 2               5              NA                    122
3        24931.                 2               5              NA                    351
4       136680.                 2               5              NA                    216
5        76850.                 2               5            1112.                   272
```

Create an zero-variance filter

```
na_filter <- credit_df %>%
  summarize(across(everything(), ~ var(., na.rm = TRUE))) %>%
  pivot_longer(everything(), names_to = "feature", values_to = "variance") %>%
  filter(variance == 0) %>%
  pull(feature)
na_filter
```

```
"num_bank_accounts" "num_credit_card"
```

Create missing values filter

```
na_filter <- credit_df %>%
  summarize(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(everything(), names_to = "feature", values_to = "num_missing_values") %>%
  filter(num_missing_values > 0) %>%
  pull(feature)
na_filter
```

```
"outstanding_debt"
```

Applying the combined filter

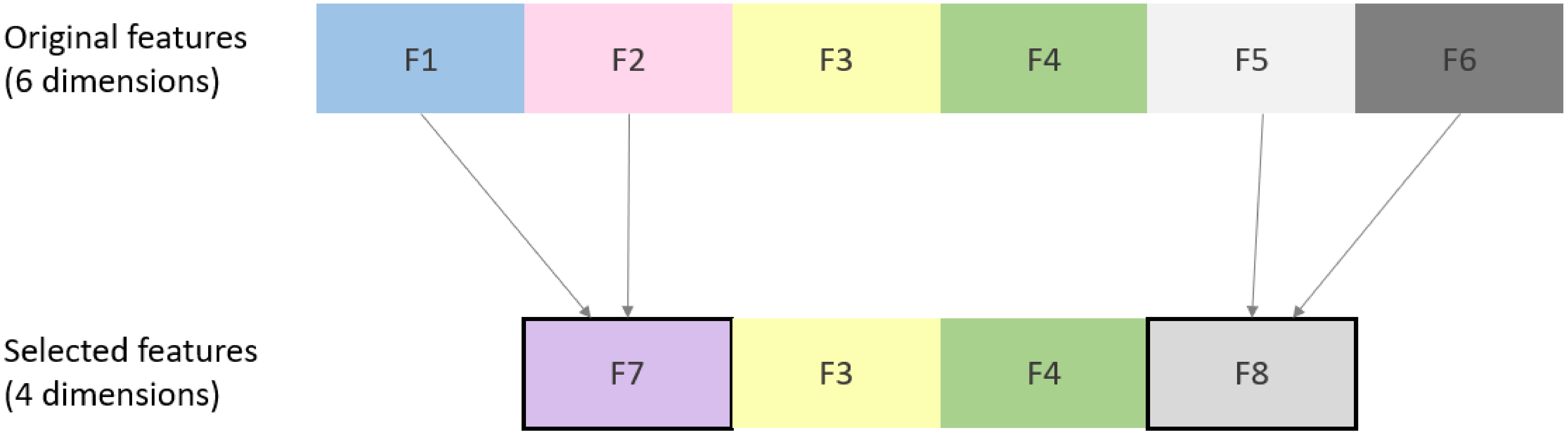
```
combined_filter <-
  c(low_var_filter, na_filter)

credit_df %>%
  select(-all_of(combined_filter)) %>%
  head(3)
```

```
  annual_income credit_history_months
          <dbl>                 <dbl>
1        87630.                   286
2        16574.                   122
3        24931.                   351
```

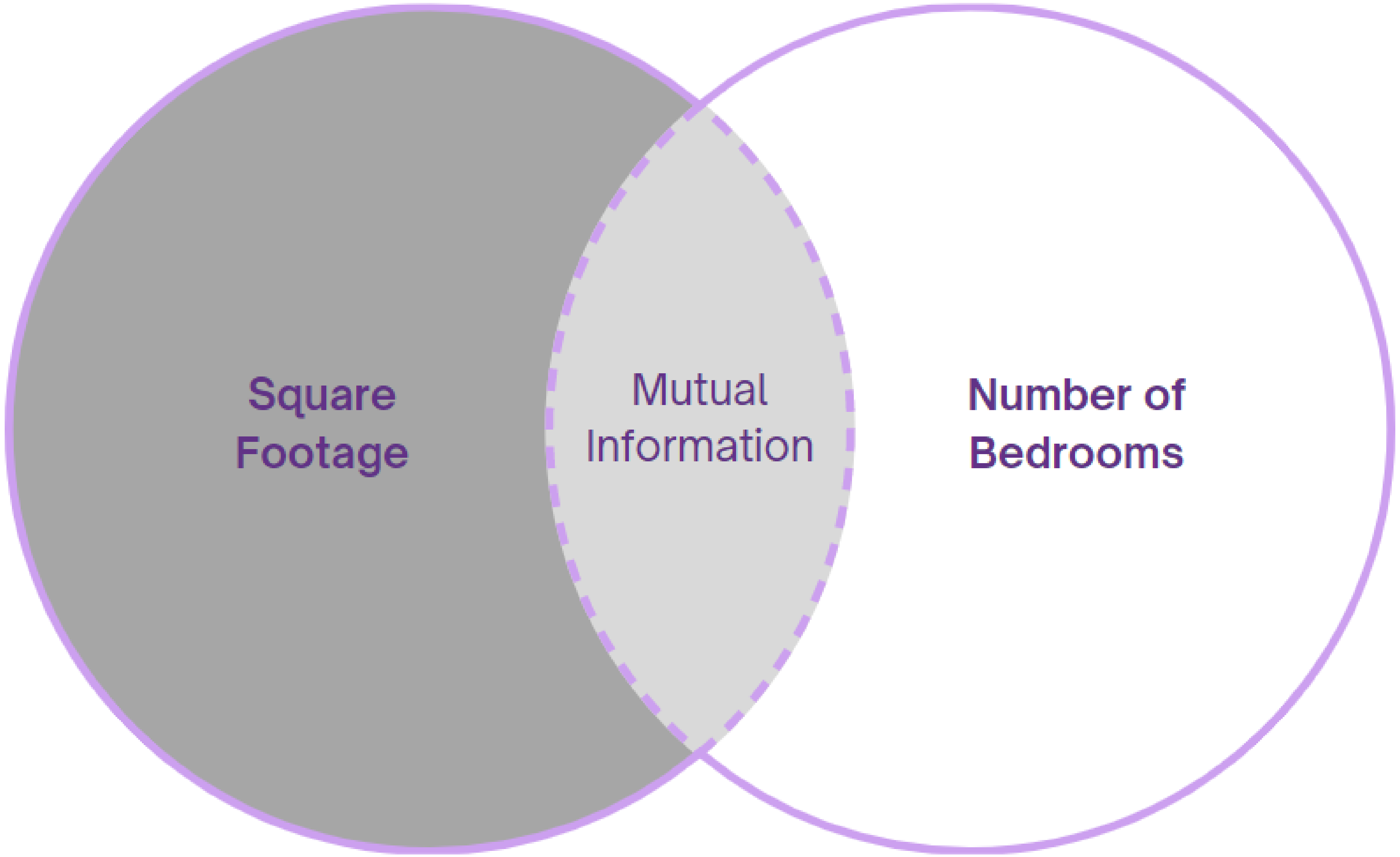
Feature extraction

Original features (6 dimensions)	F1	F2	F3	F4	F5	F6

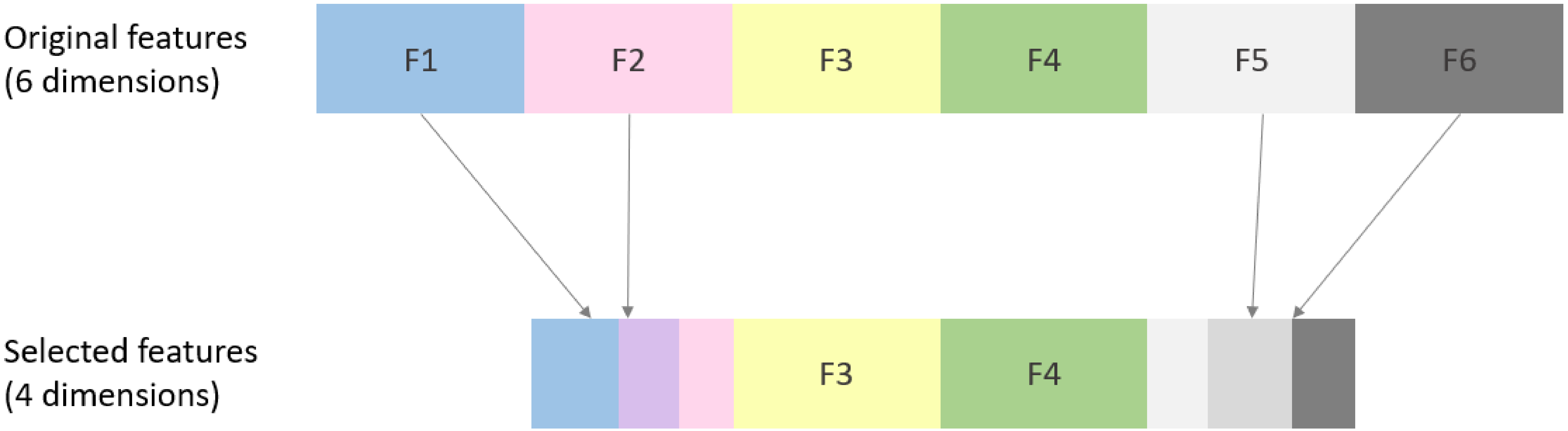
Feature extraction



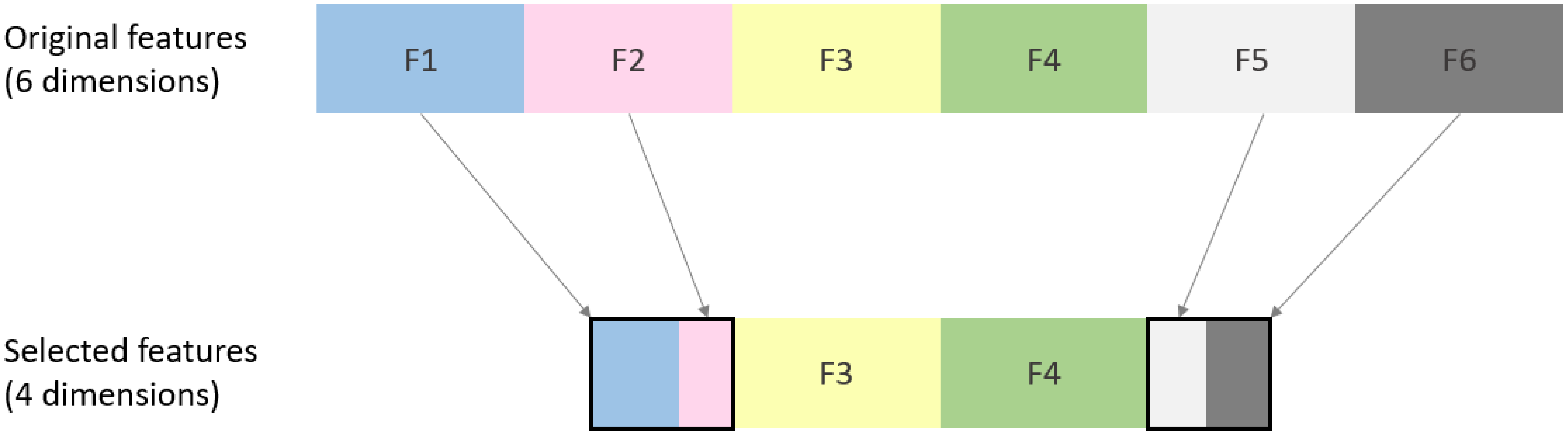
Feature extraction and mutual information



Feature extraction: Combining mutual exclusive info



Feature extraction: Combining mutual exclusive info



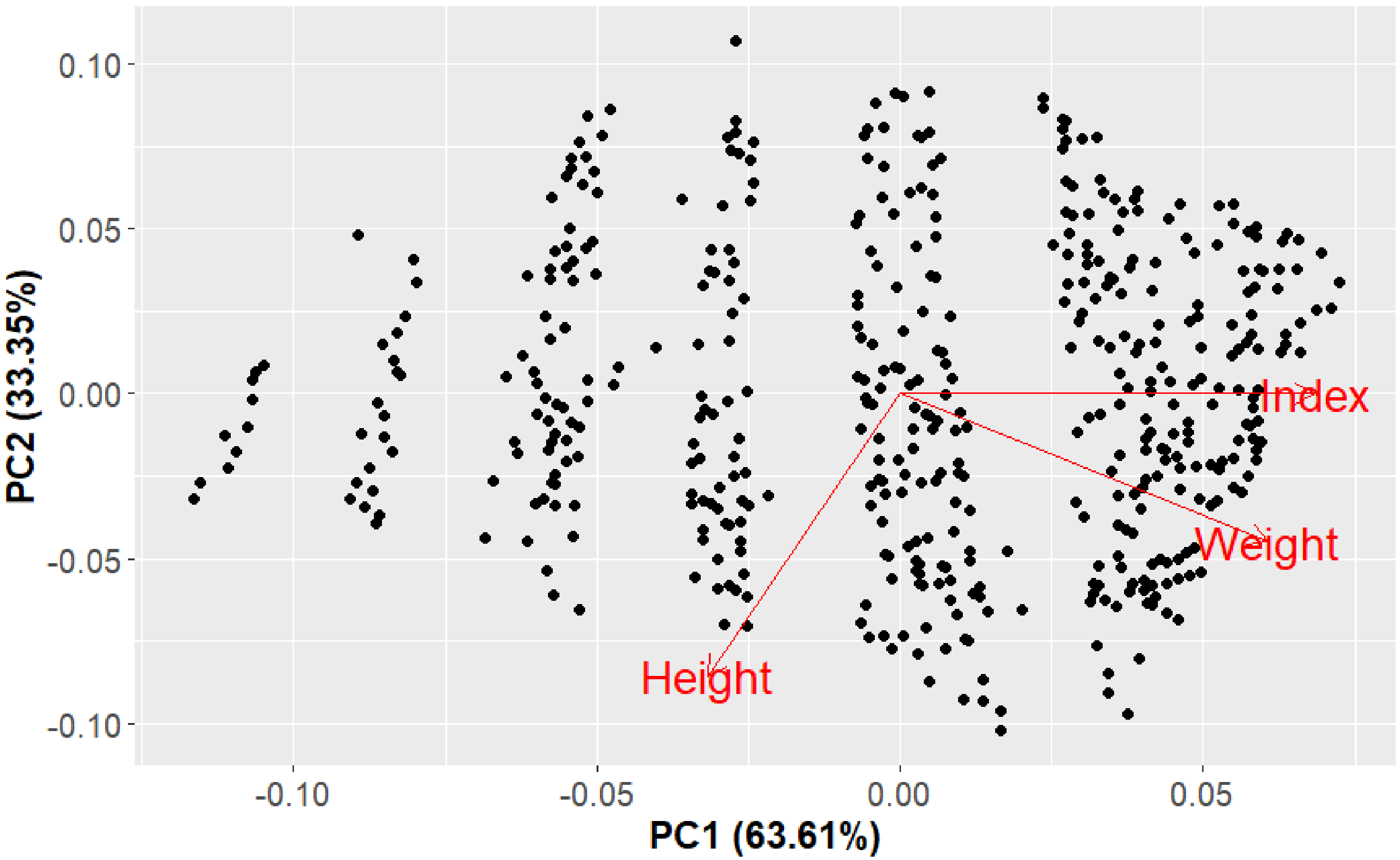
Advantages and disadvantages of feature extraction

Advantages

- can combine information into new features

Disadvantages

- implementation is more complicated
- new features are difficult to interpret



Let's practice!

DIMENSIONALITY REDUCTION IN R

Selecting based on missing values

DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

Calculate missing values ratio

$$Missing\ Value\ Ratio = \frac{\#\ of\ missing\ values}{\#\ of\ observations}$$

```
n <- nrow(credit_df)

missing_vals_df <- credit_df %>%
  summarize(across(everything(), ~ sum(is.na(.)))) %>%
  pivot_longer(everything(), names_to = "feature", values_to = "num_missing_values") %>%
  mutate(missing_val_ratio = num_missing_values / n)
```

Missing values ratio output

```
missing_vals_df
```

```
# A tibble: 5 × 3
  feature          num_missing_values missing_val_ratio
  <chr>                         <int>             <dbl>
1 credit_score                      0             0
2 annual_income                     0             0
3 age                              84             0.613
4 outstanding_debt                129             0.942
5 num_of_loan                       0             0
```

Rules of thumb for missing value ratio threshold

- No objective cutoffs
- Depends on feature importance
  - Example: `outstanding_debt` vs. `age`

Threshold	Rule of Thumb
< 0.20	Keep
0.2 to 0.8	Keep if feature is important
> 0.8	Discard

Create the missing values filter

```
missing_vals_filter <- missing_vals_df %>%
  filter(missing_val_ratio <= 0.5) %>%
  pull(feature)

missing_vals_filter
```

```
[1] "credit_score"  "annual_income" "num_of_loan"
```

Apply missing values filter

```
filtered_credit_df <- credit_df %>%
  select(missing_vals_filter)

filtered_credit_df %>% head(3)
```

```
# A tibble: 5 × 3
  credit_score annual_income num_of_loan
  <chr>                <dbl>       <dbl>
1 Standard            87630.           4
2 Standard            16574.           7
3 Standard            24931.           2
```

The tidymodel approach

Create the recipe

```
missing_vals_recipe <-
  recipe(credit_score ~ ., data = credit_df) %>%
  step_filter_missing(all_predictors(), threshold = 0.5) %>%
  prep()
```

Apply the recipe

```
filtered_credit_df <-
  bake(missing_vals_recipe, new_data = NULL)
```

Baked recipe output

```
filtered_credit_df %>% head(5)
```

```
# A tibble: 5 × 3
  annual_income num_of_loan credit_score
          <dbl>       <dbl> <fct>
1        87630.           4 Standard
2        16574.           7 Standard
3        24931.           2 Standard
4       136680.           1 Good
5        76850.           3 Standard
```

Let's practice!

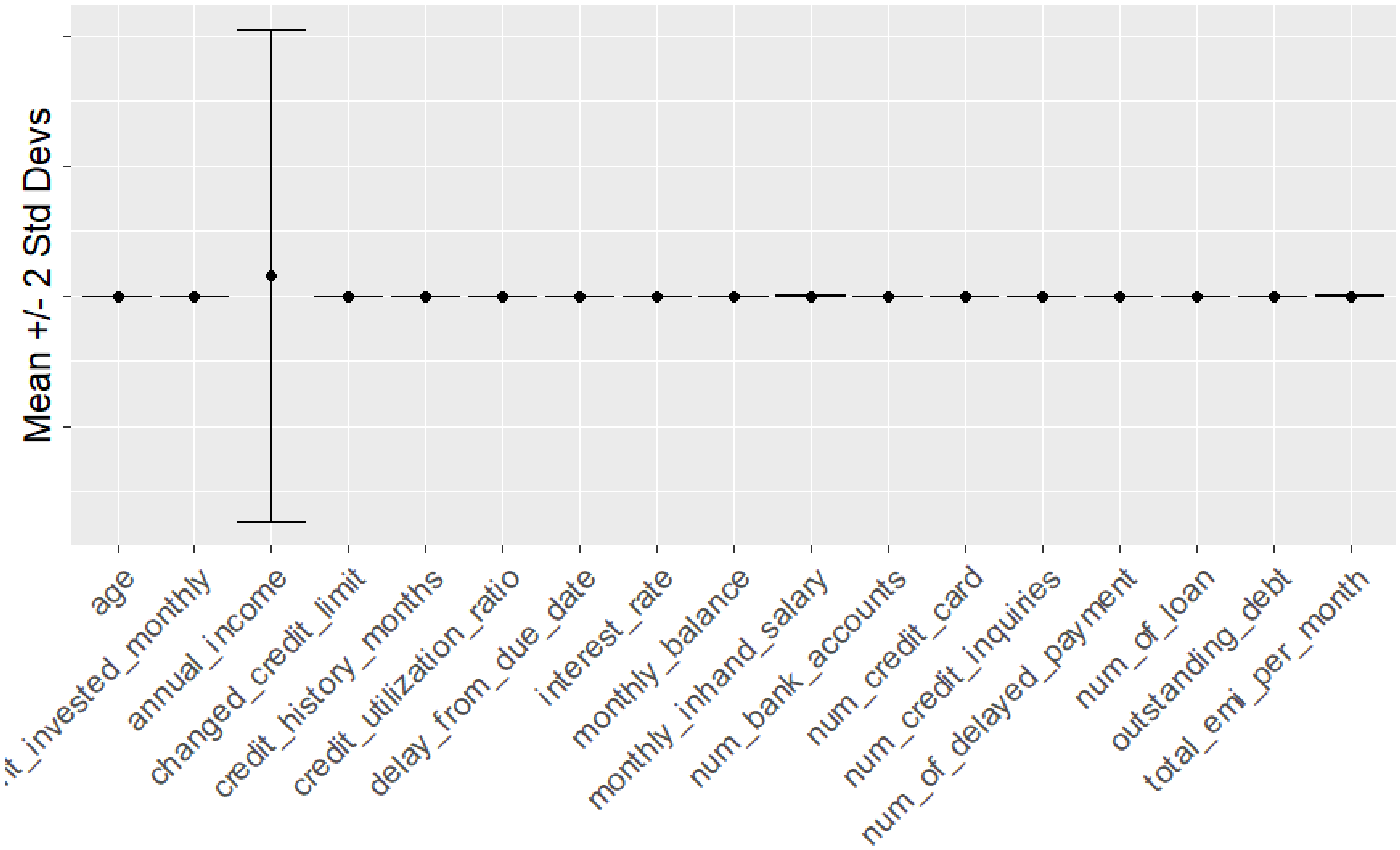
DIMENSIONALITY REDUCTION IN R

Selecting based on variance

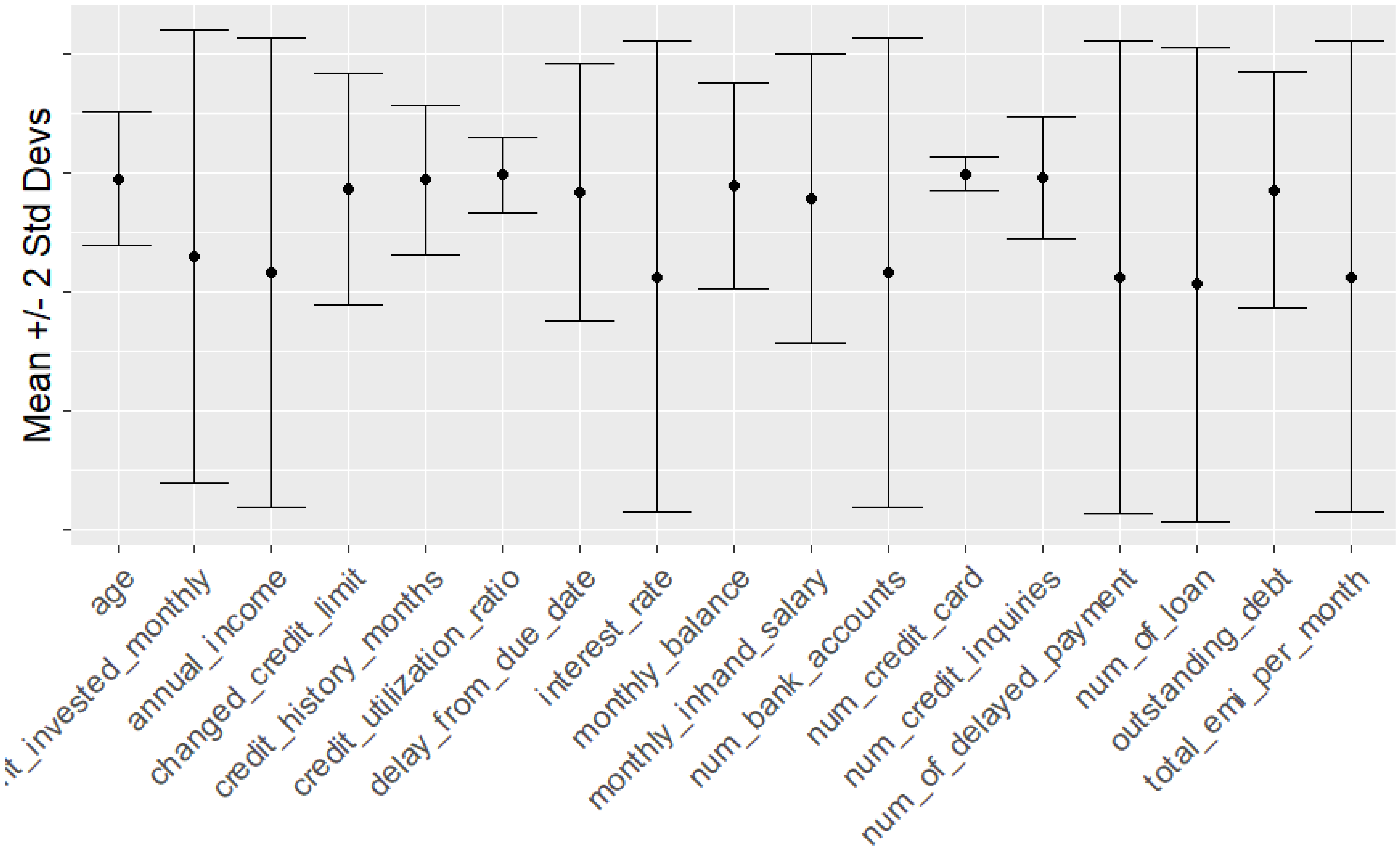
DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

Variance of unscaled data



Variance of scaled data



Calculate scaled variances

```
credit_variances <- credit_df %>%
  summarize(across(everything(), ~ var(scale(., center = FALSE)), na.rm = TRUE)) %>%
  pivot_longer(everything(), names_to = "feature", values_to = "variance") %>%
  arrange(desc(variance))

credit_variances
```

```
# A tibble: 17 × 2
   feature                  variance
   <chr>                       <dbl>
 1 num_of_loan                0.996
 2 num_of_delayed_payment     0.986
 ...
```

Variance cutoff

```
# A tibble: 17 × 2
   feature                  variance
   <chr>                       <dbl>
 1 num_of_loan               0.996
 2 num_of_delayed_payment    0.986
 3 total_emi_per_month       0.984
 4 interest_rate             0.983
 5 num_bank_accounts         0.975
 6 annual_income             0.973
 7 amount_invested_monthly   0.909
 8 monthly_inhand_salary     0.369
 9 delay_from_due_date       0.293
10 outstanding_debt          0.248
11 changed_credit_limit      0.238
12 monthly_balance           0.186
13 credit_history_months     0.0980
14 age                       0.0783
15 num_credit_inquiries      0.0647
16 credit_utilization_ratio  0.0251
17 num_credit_card           0.00523
```

Variance cutoff

```
# A tibble: 17 × 2
   feature                   variance
   <chr>                        <dbl>
 1 num_of_loan                0.996
 2 num_of_delayed_payment     0.986
 3 total_emi_per_month        0.984
 4 interest_rate              0.983
 5 num_bank_accounts          0.975
 6 annual_income              0.973
 7 amount_invested_monthly    0.909
 8 monthly_inhand_salary      0.369
 9 delay_from_due_date        0.293
10 outstanding_debt           0.248
11 changed_credit_limit       0.238
12 monthly_balance            0.186
13 credit_history_months      0.0980
14 age                        0.0783
15 num_credit_inquiries       0.0647
16 credit_utilization_ratio   0.0251
17 num_credit_card            0.00523
```

Variance cutoff

```
# A tibble: 17 × 2
   feature                  variance
   <chr>                       <dbl>
 1 num_of_loan                0.996
 2 num_of_delayed_payment     0.986
 3 total_emi_per_month        0.984
 4 interest_rate              0.983
 5 num_bank_accounts          0.975
 6 annual_income              0.973
 7 amount_invested_monthly    0.909
 8 monthly_inhand_salary      0.369
 9 delay_from_due_date        0.293
10 outstanding_debt           0.248
```

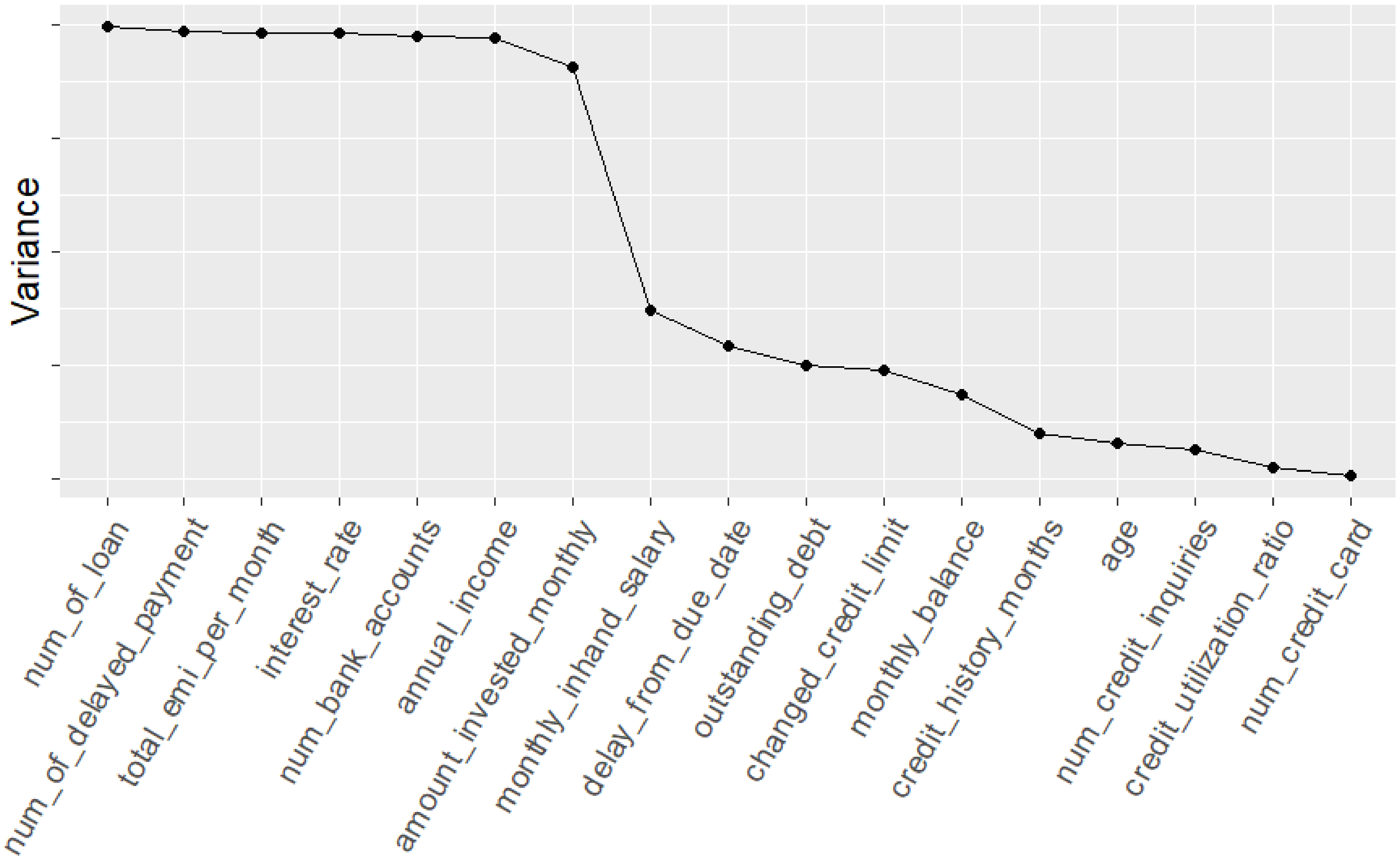
```
11 changed_credit_limit       0.238
12 monthly_balance            0.186
13 credit_history_months      0.0980
14 age                        0.0783
15 num_credit_inquiries       0.0647
16 credit_utilization_ratio   0.0251
17 num_credit_card            0.00523
```

Variance cutoff

```
# A tibble: 17 × 2
   feature                  variance
   <chr>                       <dbl>
 1 num_of_loan               0.996
 2 num_of_delayed_payment    0.986
 3 total_emi_per_month       0.984
 4 interest_rate             0.983
 5 num_bank_accounts         0.975
 6 annual_income             0.973
 7 amount_invested_monthly   0.909
 8 monthly_inhand_salary     0.369
 9 delay_from_due_date       0.293
10 outstanding_debt          0.248
```

```
11 changed_credit_limit      0.238
12 monthly_balance           0.186
13 credit_history_months     0.0980
14 age                       0.0783
15 num_credit_inquiries      0.0647
16 credit_utilization_ratio  0.0251
17 num_credit_card           0.00523
```

Variance cutoff plot



Create variance filter

```
low_var_filter <- credit_variances %>%
  filter(variance < 0.1) %>%
  pull(feature)

low_var_filter
```

```
[1] "credit_history_months"    "age"
[3] "num_credit_inquiries"     "credit_utilization_ratio"
[5] "num_credit_card"
```

The tidymodel approach

Create the recipe

```
low_variance_recipe <- recipe(credit_score ~ ., data = credit_df) %>%
  step_zv(all_predictors()) %>%
  step_scale(all_numeric_predictors()) %>%
  step_nzv(all_predictors()) %>%
  prep()
```

Apply the recipe

```
filtered_credit_df <- bake(low_variance_recipe, new_data = NULL)
```

Investigating effect of a specific step

```
low_variance_recipe <- recipe(credit_score ~ ., data = credit_df) %>%
  step_zv(all_predictors()) %>%
  step_scale(all_numeric_predictors()) %>%
  step_nzv(all_predictors()) %>%
  prep()

tidy(low_variance_recipe, number = 3)
```

```
  terms                 id
  <chr>                 <chr>
1 num_credit_card       nzv_ni8L7
2 num_credit_inquiries nzv_ni8L7
```

Let's practice!

DIMENSIONALITY REDUCTION IN R

Selecting based on correlation with other features

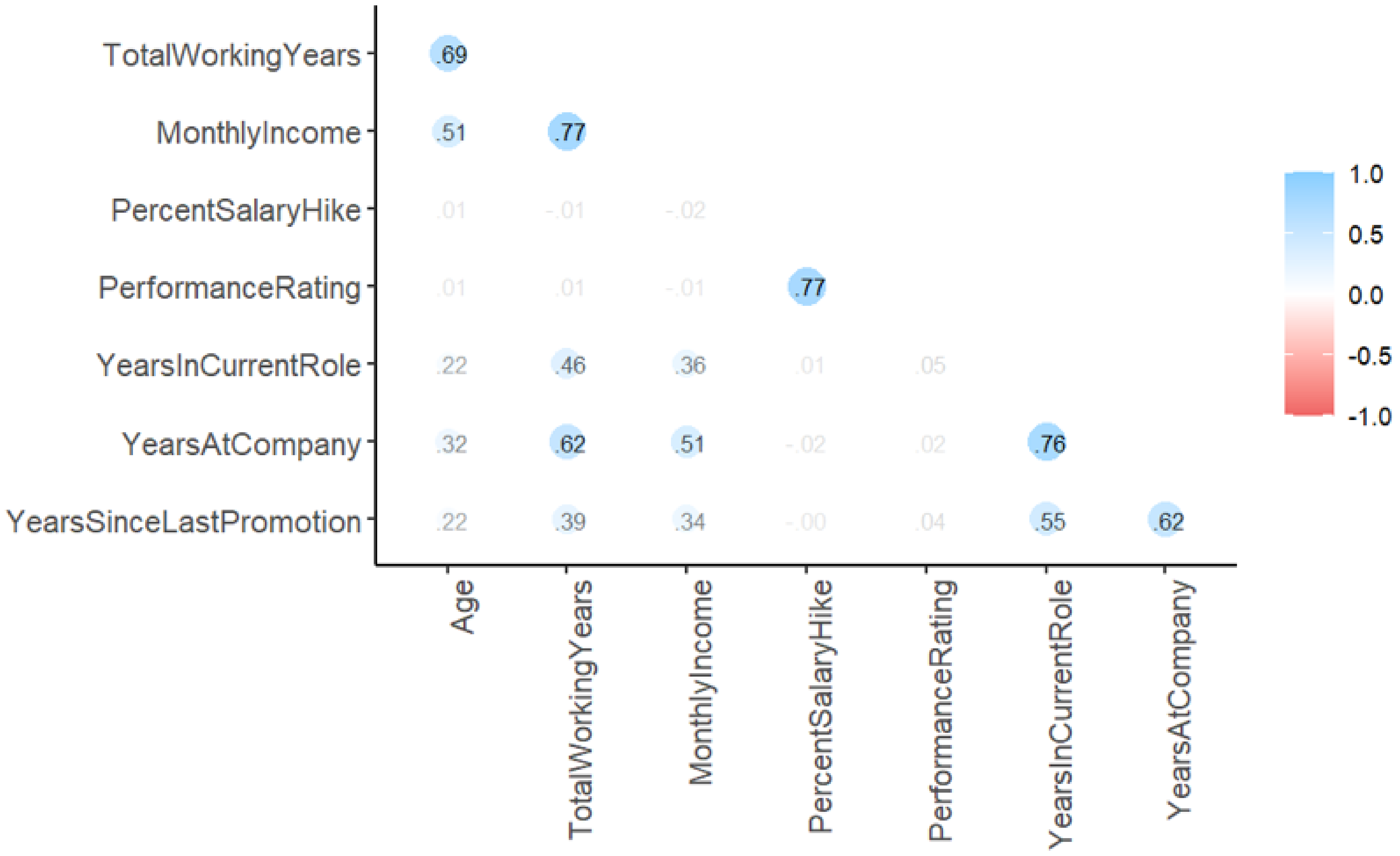
DIMENSIONALITY REDUCTION IN R

Matt Pickard
Owner, Pickard Predictives, LLC

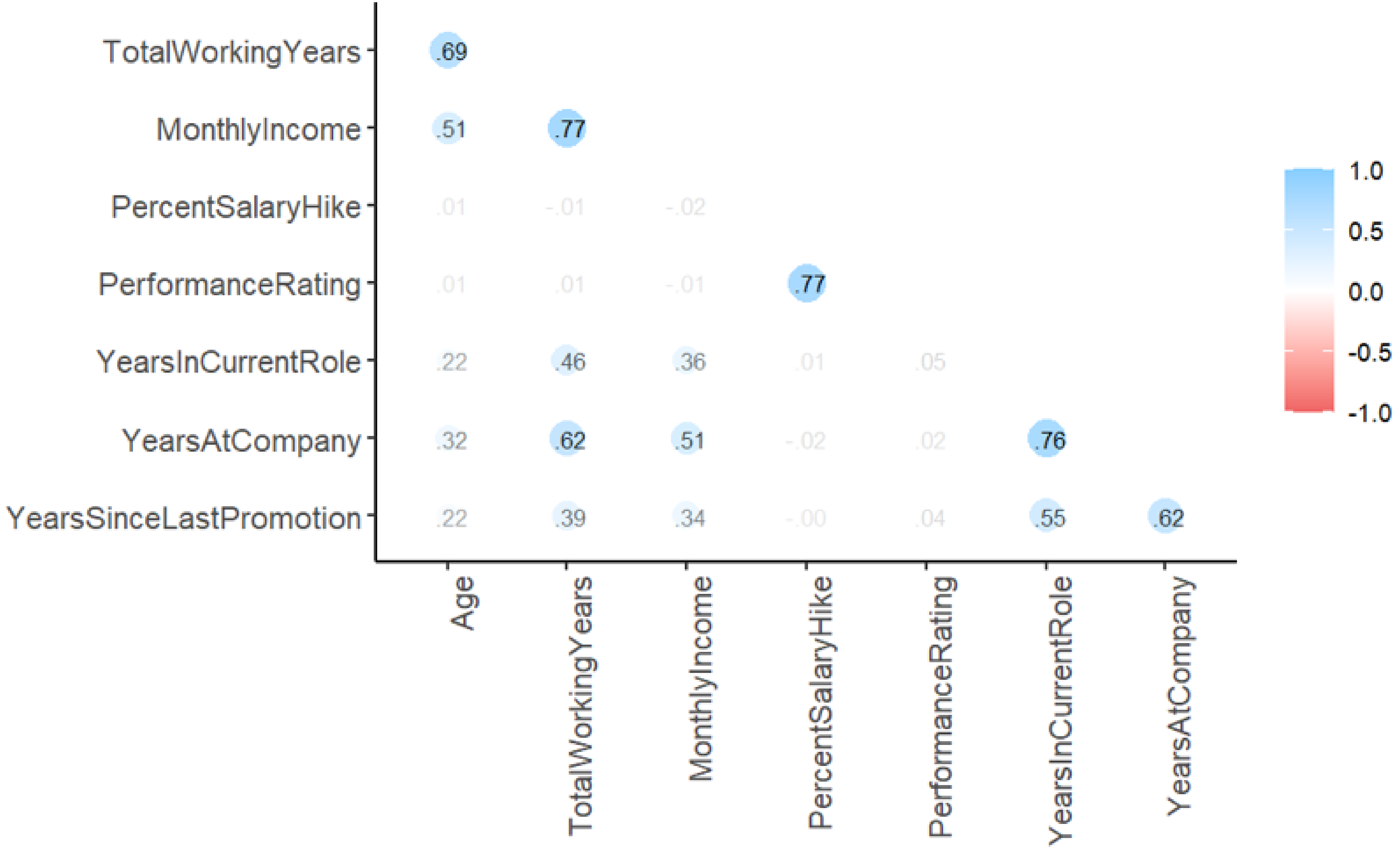
Review correlation plot creation

```
healthcare_df %>%
  select(where(is.numeric)) %>%
  correlate() %>%
  shave() %>%
  rplot(print_cor = TRUE) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

Correlation plot

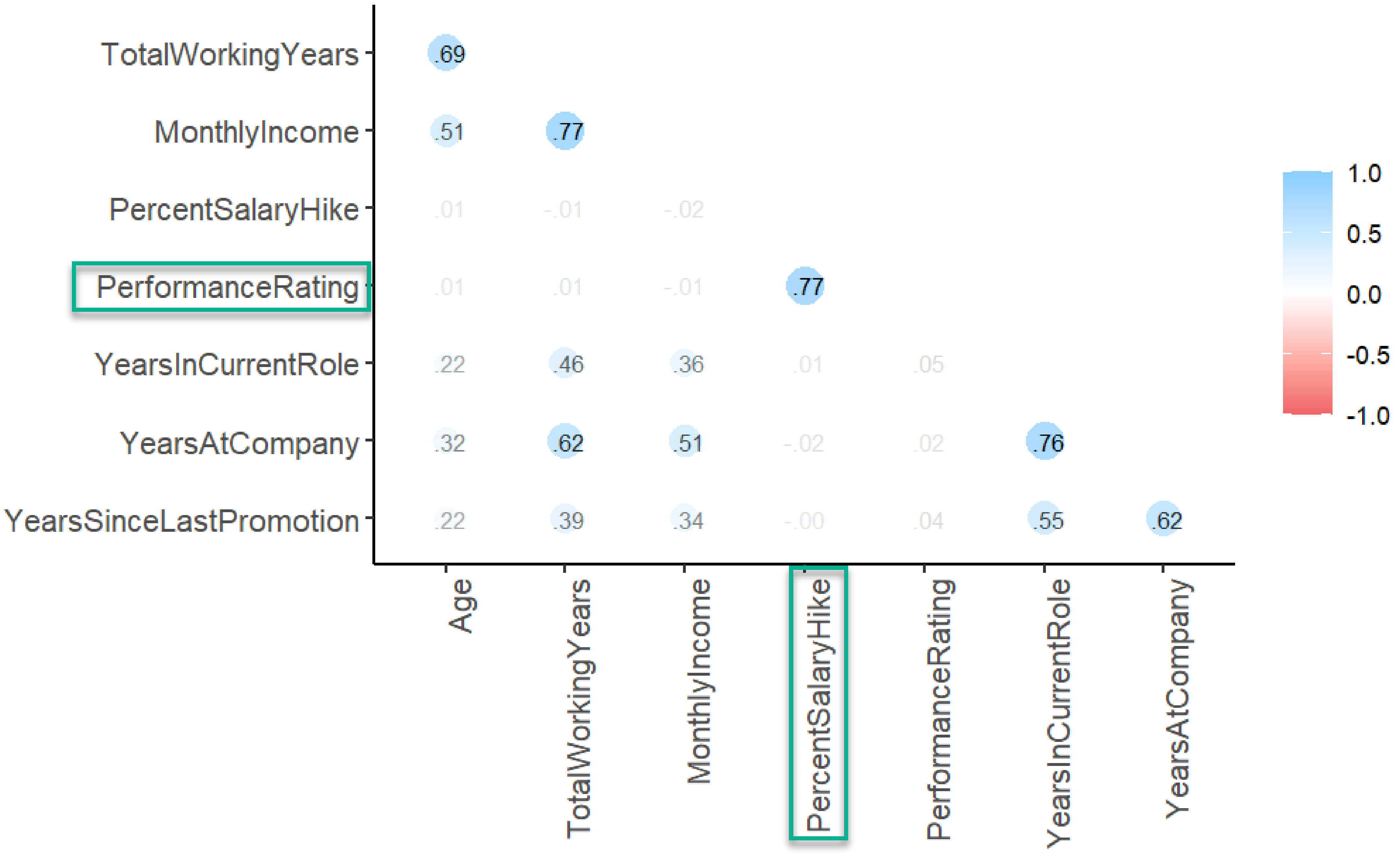


Correlation strength

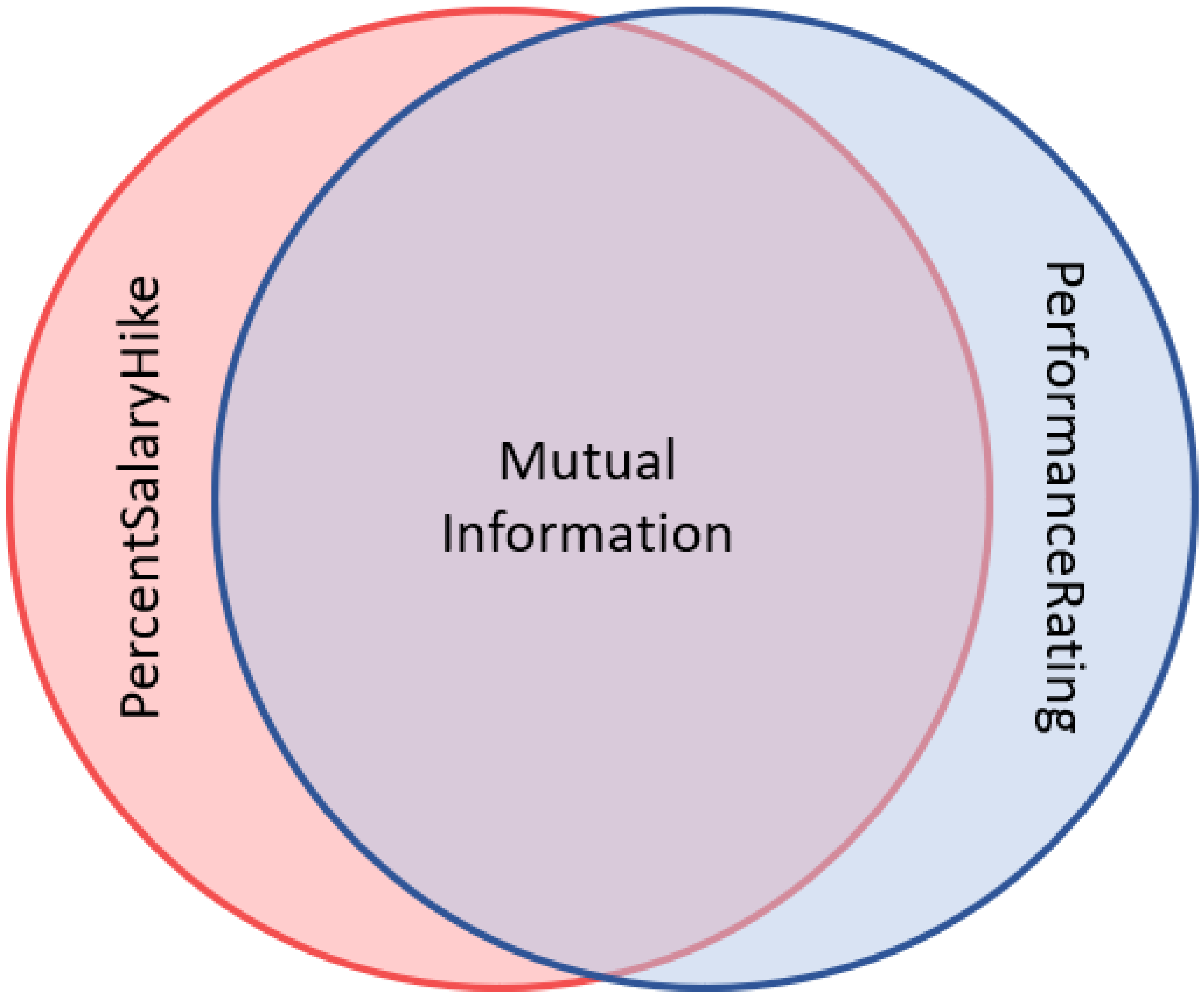


< 0.3	Low
0.3 to 0.7	Medium
> 0.7	High

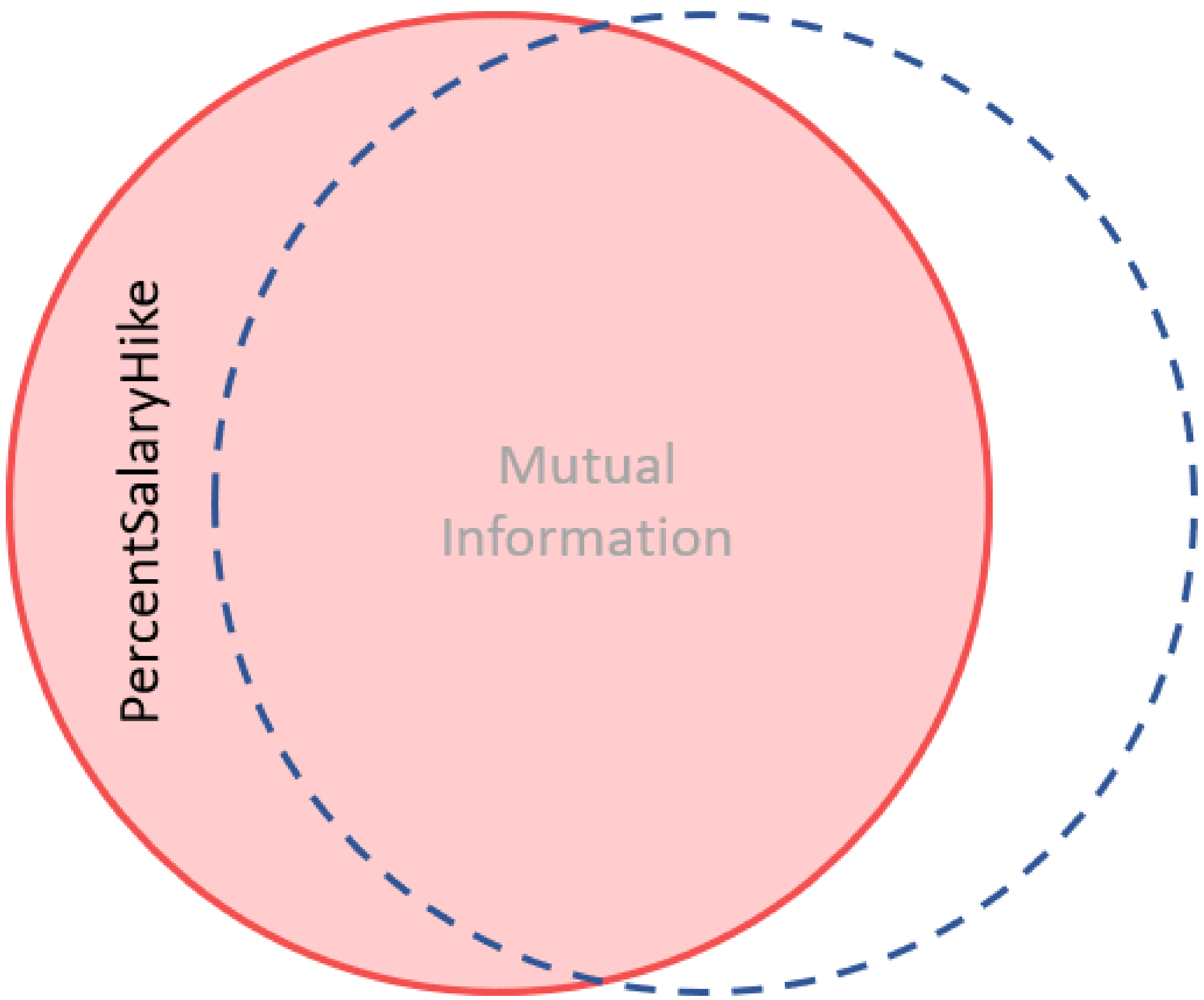
A correlation filter?



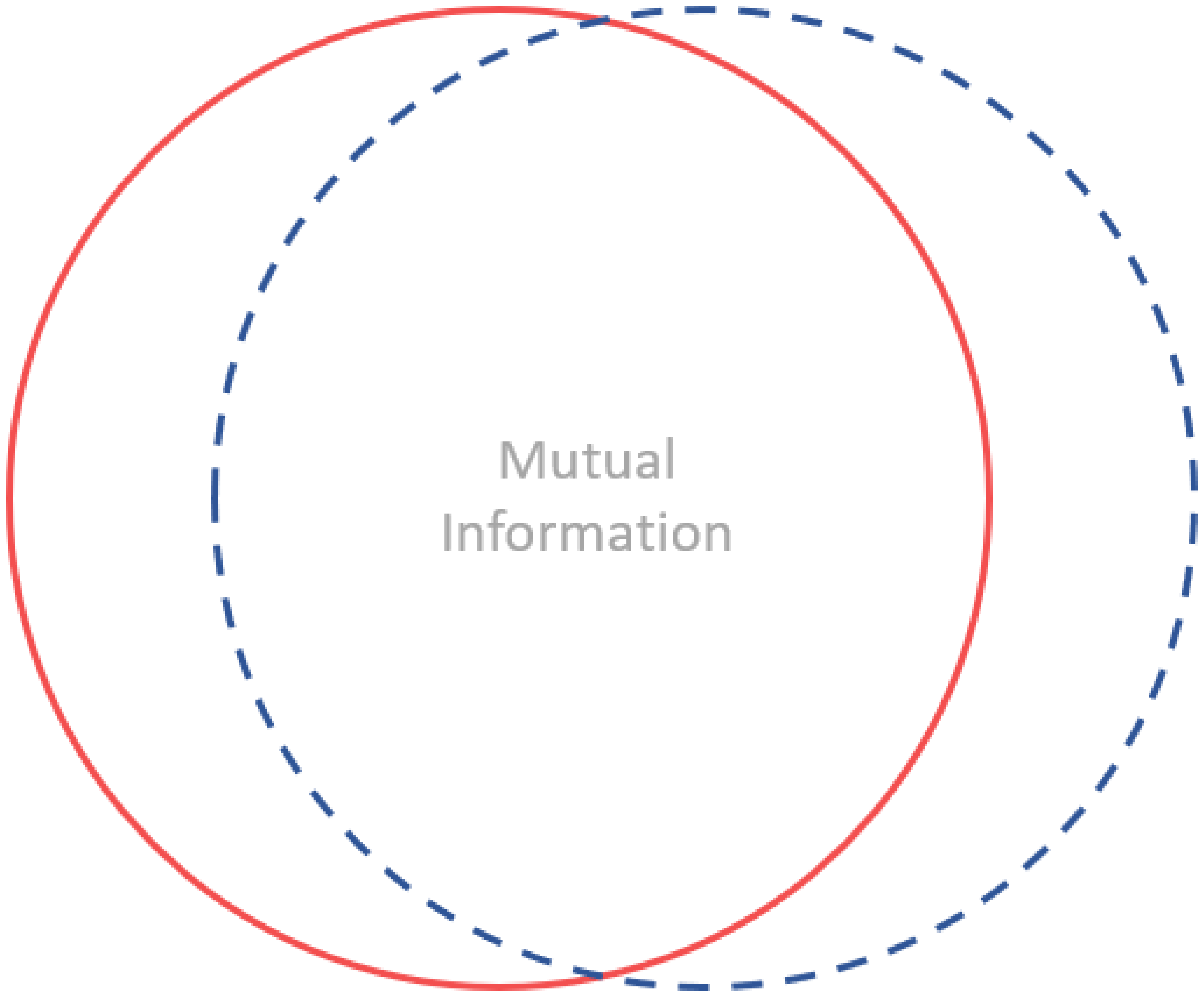
A correlation filter?



A correlation filter?



A correlation filter?



A correlation filter recipe

```
# create and prep the recipe
corr_recipe <-
  recipe(Attrition ~ ., data = healthcare_df) %>%
  step_corr(all_numeric_predictors(), threshold = 0.7) %>%
  prep()

# Apply the recipe to the data
filtered_healthcare_df <-
  corr_recipe %>%
  bake(new_data = NULL)

# Identify the features that were removed
tidy(corr_recipe, number = 1)
```

Let's practice!

DIMENSIONALITY REDUCTION IN R