# Fitbit Feature Engineering 2 : Predict sleep_score without stress_score

GOAL : predict sleep_score without stress_score\ CONCLUSION : Model m4 has the highest rsq and lowest mrse. m4 is a linear regression model that uses predictors date and deep_sleep_min only.

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.2     v purrr   1.0.2
## v tibble  3.2.1     v dplyr   1.1.2
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(dplyr)
library(ggplot2)
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.1.0 --
## v broom        1.0.5     v rsample      1.1.1
## v dials        1.2.0     v tune         1.1.1
## v infer        1.0.4     v workflows    1.1.3
## v modeldata    1.2.0     v workflowsets 1.0.1
## v parsnip      1.1.0     v yardstick    1.2.0
## v recipes      1.0.7
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use suppressPackageStartupMessages() to eliminate package startup messages
```

```r
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
```

```
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(yardstick)
```

```r
fitbit_df <- read.csv('fitbit_data.csv')
fitbit_df <- fitbit_df %>% select(-stress_score)
fitbit_df$date <- as.Date(fitbit_df$date)
head(fitbit_df)
```

```
##         date AZM_minutes    rmssd    nremhr   entropy sleep_score
## 1 2023-06-29         157 67.89393 0.9697126 1106.6132          68
## 2 2023-06-30          34 63.09258 0.9740137  930.9208          65
## 3 2023-07-01           1 87.91776 0.9673021 1320.8890          85
## 4 2023-07-02          26 60.61797 0.9711250  950.8540          84
## 5 2023-07-03          44 96.20780 0.9771325 1310.1257          80
## 6 2023-07-04          44 89.09386 0.9704167 1309.5501          72
##   deep_sleep_min resting_heart_rate   o2_avg o2_lower_bound o2_upper_bound
## 1             96                 58 84.79727          70.70           98.8
## 2             65                 57 83.35863          93.05           98.4
## 3            106                 57 84.84333          86.35           98.6
## 4             90                 56 84.86729          86.75           98.2
## 5             78                 56 83.33722          90.85           97.6
## 6             63                 54 78.59688          71.75           96.8
##   calories
## 1  2345.97
## 2  1772.70
## 3  1669.63
## 4  1591.05
## 5  2095.86
## 6  1463.32
```

## Split Data

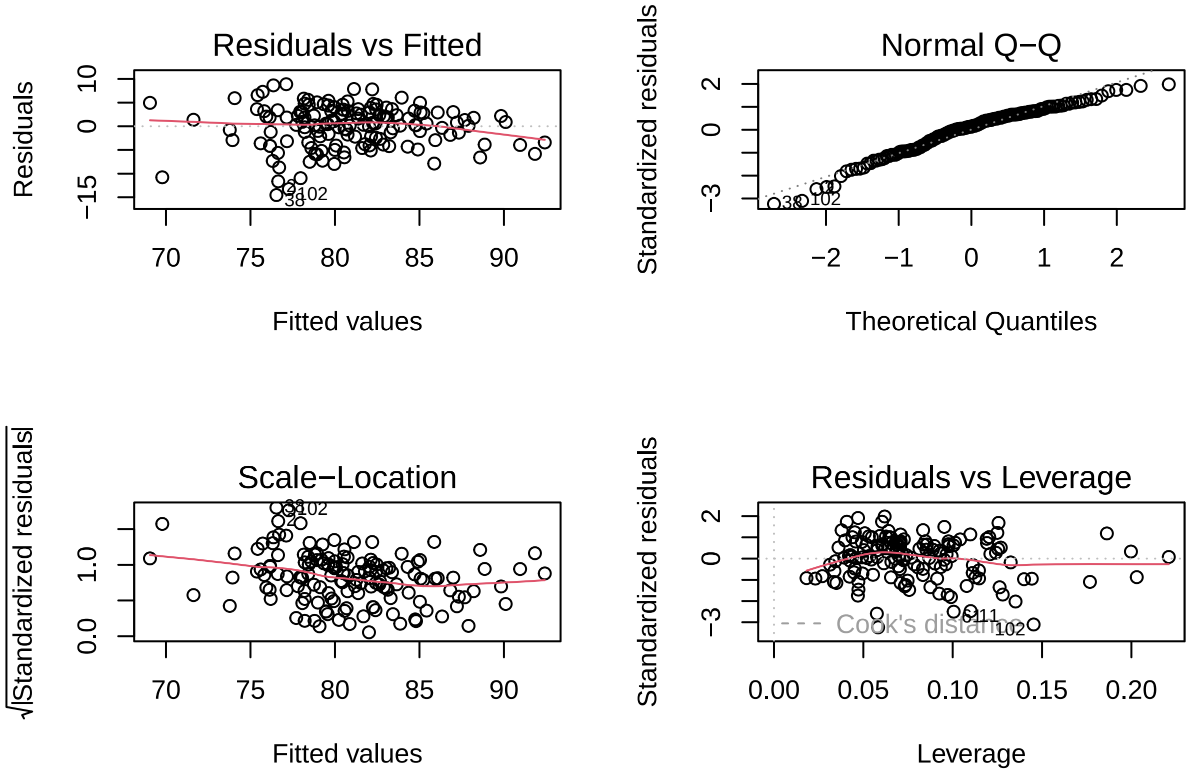
```r
set.seed(123)
split <- initial_split(fitbit_df, prop=0.9)
train <- training(split)
test <- testing(split)
```

## Influential Points

```r
lm <- lm(data=fitbit_df, sleep_score ~ .)
summary(lm)
```

```
##
## Call:
## lm(formula = sleep_score ~ ., data = fitbit_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5355  -3.0483   0.6008   3.1232   8.8847
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -1.209e+03  2.039e+02  -5.927 2.31e-08 ***
## date                5.822e-02  9.525e-03   6.113 9.32e-09 ***
## AZM_minutes         1.273e-02  1.238e-02   1.029  0.30527
## rmssd              -1.110e-01  5.456e-02  -2.035  0.04372 *
## nremhr              9.971e+01  4.769e+01   2.091  0.03836 *
## entropy             6.198e-03  3.191e-03   1.943  0.05409 .
## deep_sleep_min      9.932e-02  2.165e-02   4.587 9.92e-06 ***
## resting_heart_rate  1.351e-01  1.632e-01   0.828  0.40924
## o2_avg              2.081e-01  1.849e-01   1.126  0.26223
## o2_lower_bound      1.903e-02  4.192e-02   0.454  0.65050
## o2_upper_bound      3.345e-01  2.909e-01   1.150  0.25229
## calories           -8.376e-03  2.550e-03  -3.284  0.00129 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.623 on 139 degrees of freedom
## Multiple R-squared:  0.4528, Adjusted R-squared:  0.4095
## F-statistic: 10.46 on 11 and 139 DF,  p-value: 8.55e-14
```

```
par(mfrow=c(2,2))
plot(lm)
```

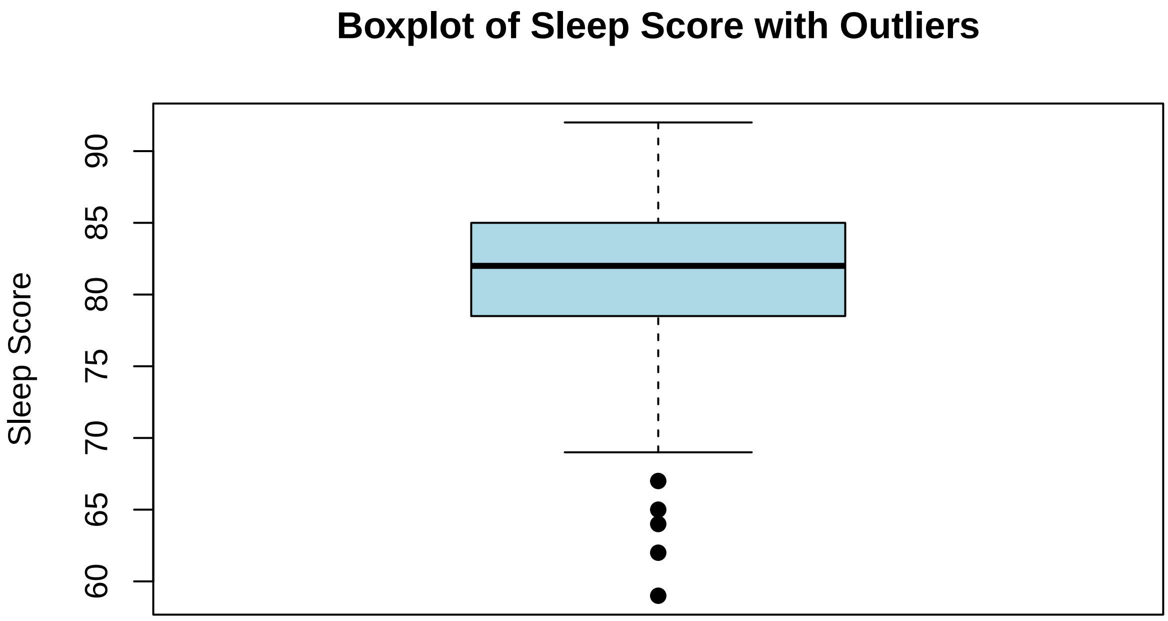


### Outliers : 36 122 124 132 133

```r
q <- quantile(train$sleep_score, c(0.25, 0.75))
iqr <- IQR(train$sleep_score)
threshold <- 1.5 * iqr
outliers <- which(train$sleep_score < (q[1] - threshold) | train$sleep_score > (q[2] + threshold))
print(outliers)
```

```
## [1]  36 122 124 132 133
```

```r
boxplot(train$sleep_score, main = "Boxplot of Sleep Score with Outliers",
        ylab = "Sleep Score", col = "lightblue", pch = 19)
points(outliers, train$sleep_score[outliers], col = "red", pch = 19)
```



# New train set after removing outliers

```
train <- train[-outliers]
```

## m1 : all predictors

```
m1 <- linear_reg()

m1_recipe <- recipe(data=train, sleep_score ~ .) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_date(date, features = c("dow", "month", "year")) %>%
  step_holiday(date) %>%
  step_corr(all_numeric_predictors(), threshold = 0.5) %>%
  step_YeoJohnson(all_numeric_predictors())


m1_wkfl <- workflow() %>%
  add_model(m1) %>%
  add_recipe(m1_recipe)

m1_fit <- m1_wkfl %>%
  fit(data=train)
```

```
## Warning in stats::cor(x, use = use, method = method): the standard deviation is
## zero

## Warning: The correlation matrix has missing values. 4 columns were excluded from
## the filter.
```

```
m1_aug <- m1_fit %>%
  augment(test)
```

```
## Warning in predict.lm(object = object$fit, newdata = new_data, type =
## "response"): prediction from a rank-deficient fit may be misleading
```

```
m1_aug %>%
  metrics(truth = sleep_score, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.97
## 2 rsq     standard       0.702
## 3 mae     standard       2.44
```

## m2 : 5 selected predictors based on my expectation

```
m2 <- linear_reg()

m2_recipe <- recipe(data=train, sleep_score ~ date+deep_sleep_min+AZM_minutes+o2_avg+resting_heart_rate)
    step_normalize(all_numeric_predictors()) %>%
    step_date(date, features = c("dow", "month", "year")) %>%
    step_holiday(date) %>%
    step_corr(all_numeric_predictors(), threshold = 0.5) %>%
    step_YeoJohnson(all_numeric_predictors())

m2_wkfl <- workflow() %>%
  add_model(m2) %>%
  add_recipe(m2_recipe)

m2_fit <- m2_wkfl %>%
  fit(data=train)
```

```
## Warning in stats::cor(x, use = use, method = method): the standard deviation is
## zero

## Warning: The correlation matrix has missing values. 4 columns were excluded from
## the filter.
```

```
m2_aug <- m2_fit %>%
  augment(test)
```

```
## Warning in predict.lm(object = object$fit, newdata = new_data, type =
## "response"): prediction from a rank-deficient fit may be misleading
```

```
m2_aug %>%
  metrics(truth = sleep_score, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       3.02
## 2 rsq     standard       0.699
## 3 mae     standard       2.26
```

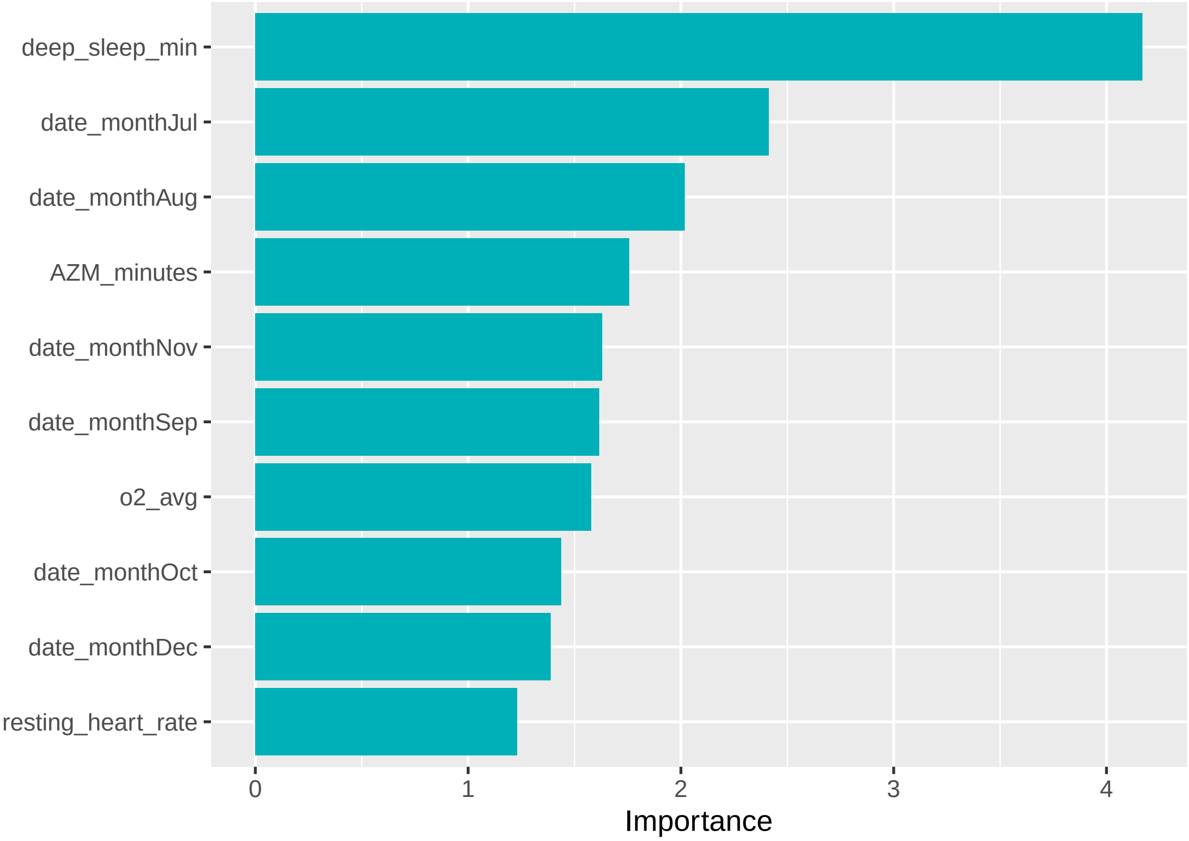
### m3 : 3 selected predictors based on importance

```
library(vip)
```

```
##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

```
m1_fit %>%
  extract_fit_parsnip() %>%
  vip(aesthetics = list(fill = "#00B0B9"))
```



```
m3 <- linear_reg()

m3_recipe <- recipe(data=train, sleep_score ~ date+deep_sleep_min+AZM_minutes) %>%
    step_normalize(all_numeric_predictors()) %>%
    step_date(date, features = c("dow", "month", "year")) %>%
    step_holiday(date) %>%
    step_corr(all_numeric_predictors(), threshold = 0.5) %>%
    step_YeoJohnson(all_numeric_predictors())
```

```
m3_wkfl <- workflow() %>%
  add_model(m3) %>%
  add_recipe(m3_recipe)

m3_fit <- m3_wkfl %>%
  fit(data=train)
```

```
## Warning in stats::cor(x, use = use, method = method): the standard deviation is
## zero
```

```
## Warning: The correlation matrix has missing values. 4 columns were excluded from
## the filter.
```

```
m3_aug <- m3_fit %>%
  augment(test)
```

```
## Warning in predict.lm(object = object$fit, newdata = new_data, type =
## "response"): prediction from a rank-deficient fit may be misleading
```

```
m3_aug %>%
  metrics(truth = sleep_score, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.63
## 2 rsq     standard       0.771
## 3 mae     standard       1.88
```

## m4 : 2 selected predictors

rsq = 0.8137866 Highest rsq : after k=2 and remove influential -> 0.8291957

```
m4 <- linear_reg()

m4_recipe <- recipe(data=train, sleep_score ~ date+deep_sleep_min) %>%
    step_normalize(all_numeric_predictors()) %>%
    step_date(date, features = c("dow", "month", "year")) %>%
    step_holiday(date) %>%
    step_corr(all_numeric_predictors(), threshold = 0.5) %>%
    step_YeoJohnson(all_numeric_predictors())

m4_wkfl <- workflow() %>%
  add_model(m4) %>%
  add_recipe(m4_recipe)

m4_fit <- m4_wkfl %>%
  fit(data=train)
```

```
## Warning in stats::cor(x, use = use, method = method): the standard deviation is
## zero
```

```
## Warning: Too many correlations are ‘NA‘; skipping correlation filter.
```

```
m4_aug <- m4_fit %>%
  augment(test)
```

```
## Warning in predict.lm(object = object$fit, newdata = new_data, type =
## "response"): prediction from a rank-deficient fit may be misleading
```

```
m4_aug %>%
  metrics(truth = sleep_score, estimate = .pred)
```

```
## # A tibble: 3 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard       2.36
## 2 rsq     standard       0.814
## 3 mae     standard       1.83
```