Predicting sleep efficiency using linear regression

Brooke Stevens

2023-02-24

```
# Load libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
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
```

Data

```
# Sleep efficiency dataset
sleep <- read.csv("../data/Sleep_Efficiency.csv")

# Variables in dataset
colnames(sleep)
```

```
##  [1] "ID"                    "Age"                   "Gender"
##  [4] "Bedtime"               "Wakeup.time"           "Sleep.duration"
```

```
##  [7] "Sleep.efficiency"       "REM.sleep.percentage"   "Deep.sleep.percentage"
## [10] "Light.sleep.percentage" "Awakenings"             "Caffeine.consumption"
## [13] "Alcohol.consumption"    "Smoking.status"         "Exercise.frequency"
```

The `Sleep_Efficiency.csv` dataset was obtained through Kaggle. Each row represents a person in this study, and the 15 columns contain statistics related to that person's observed sleep session, including bedtime, wakeup time, sleep duration, and more.

Research Question

Using this data, I want to predict **sleep efficiency** using linear regression. The steps for wrangling the data are as follows:

1. Remove all missingness from the dataset
2. Obtain the columns relevant to the analysis

Variables of Interest

Dependent variable:

- `Sleep.efficiency`, the proportion of time in bed spent sleeping

Independent variables:

- `REM.sleep.percentage`, the percentage of sleep spent in REM
- `Deep.sleep.percentage`, the percentage of sleep spent in deep sleep
- `Light.sleep.percentage`, the percentage of sleep spent in light sleep
- `Awakenings`, the number of times woken up

Null and Alternative Hypotheses

Null Hypothesis: Sleep efficiency is **not** related to these variables

Alternative Hypothesis: Sleep efficiency **is** related to these variables

Formal Model:

$$Sleep\,Efficiency = \beta_0 + \beta_1 * REM\,Pct + \beta_2 * Deep\,Sleep\,Pct + \beta_3 * Light\,Sleep\,Pct + \beta_4 * Awakenings + \varepsilon$$

Formal Hypotheses:

$$H_0\,(null) : \beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$$

$$H_A\,(alternative) : Any\,\beta_i \neq 0$$

Data Wrangling

First, I want to remove all missingness from this dataset.

```
# Remove missingness in dataset
sleep <- sleep %>%
  drop_na()
```

Next, I want to modify the dataset to only include the columns relevant for the analysis.

```r
# Only including relevant data columns
sleep <- sleep %>%
  select(c("Sleep.efficiency",
           "REM.sleep.percentage",
           "Deep.sleep.percentage",
           "Light.sleep.percentage",
           "Awakenings"))
```

This data is now fully wrangled!

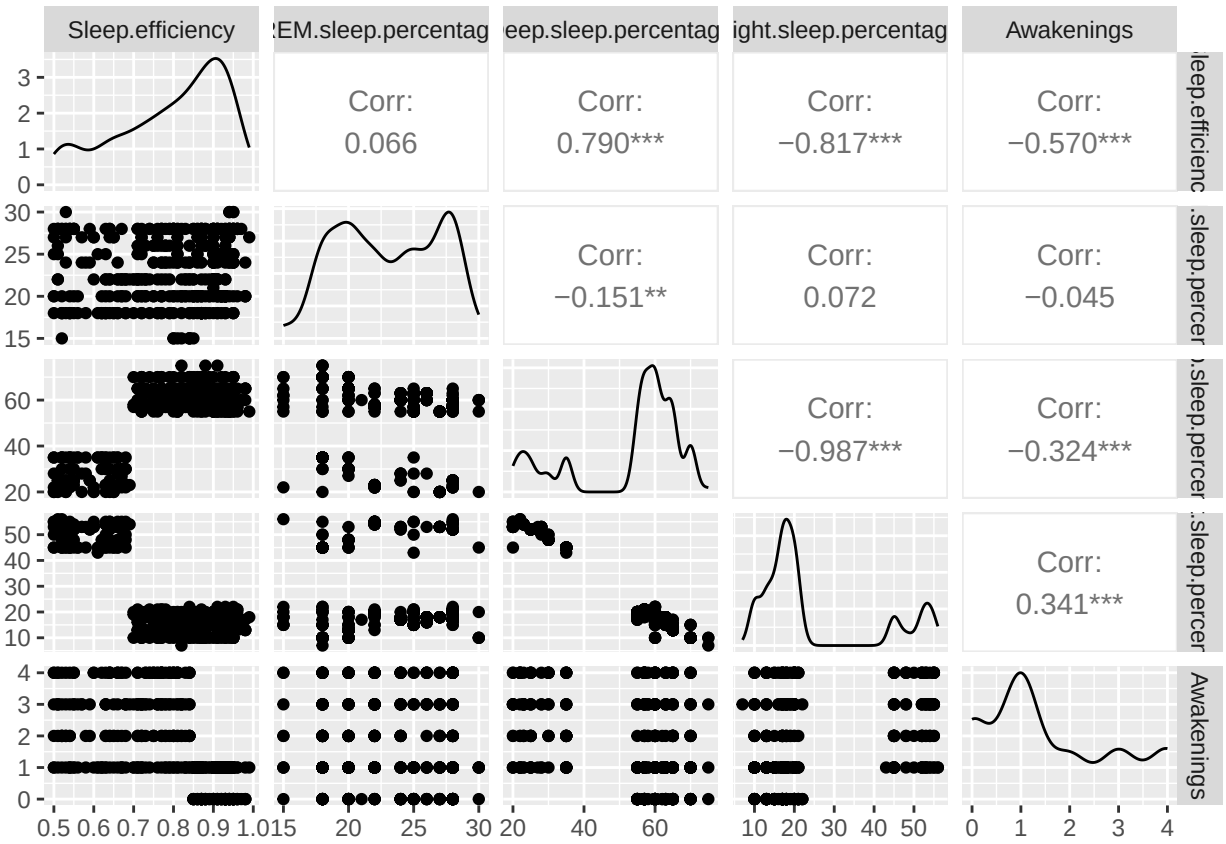
Analysis

The steps I will perform to create the linear regression are as follows:

1. Check normality and correlations of all variables of interest
2. Transform variables (if necessary)
3. Perform regression and look at model summary
4. Check for multicollinearity (using VIF > 5)
5. Remove non-significant predictors
6. Check normality of residuals (homoskedasticity) and correlation between residuals and predicted values
7. Repeat steps 3-6 as necessary

We will begin with step 1, which involves checking the normality and correlations of all variables of interest.

```r
# Checking normality and correlations using ggpairs
ggpairs(sleep)
```



Generally, these variables are not normally distributed. Some have right skew while others have left skew. The variables `Light.sleep.percentage` and `Deep.sleep.percentage` appear to have the highest correlation to the dependent variable `Sleep.efficiency`. There is a high correlation between the independent variables `Light.sleep.percentage` and `Deep.sleep.percentage`, and this may be cause for concern later. Many correlations are also statistically significant.

This calls for step 2, which involves transforming (normalizing) the variables.

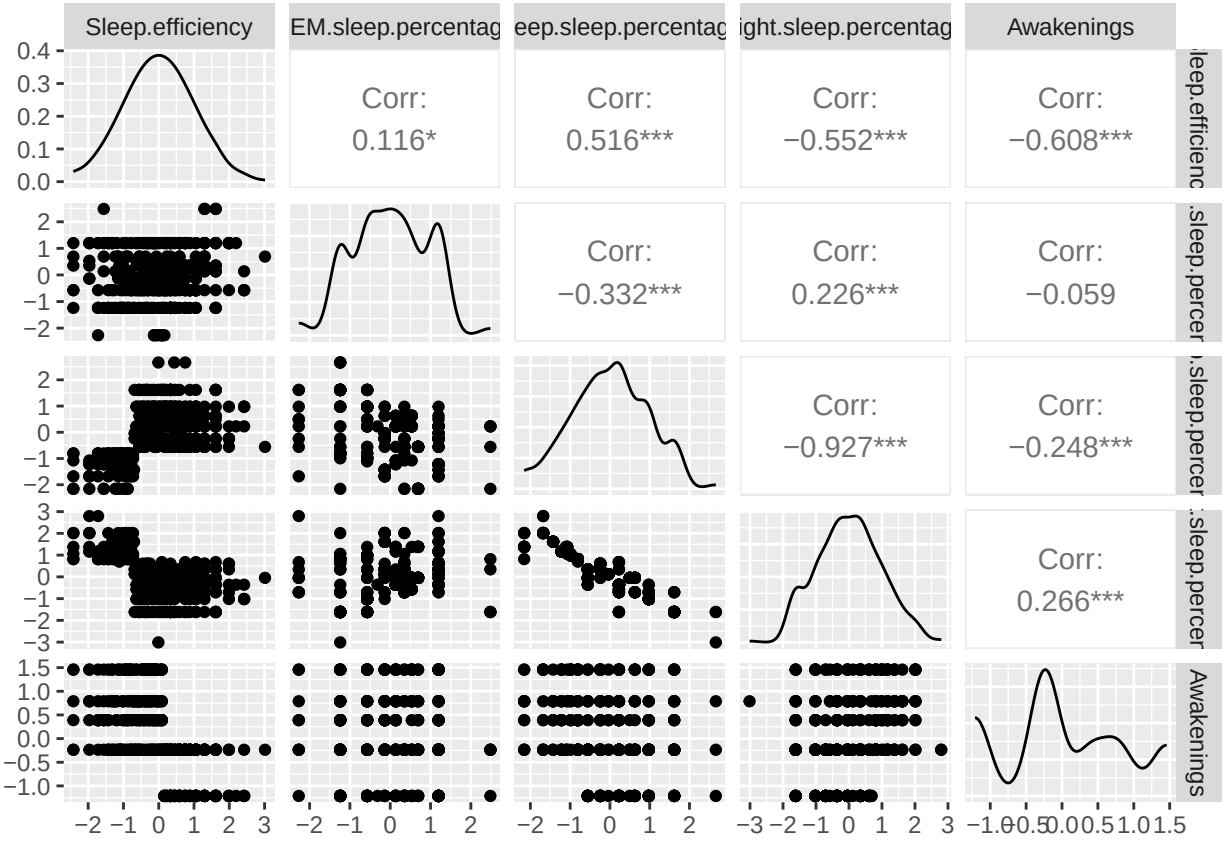
```
# Load bestNormalize library and set seed
library(bestNormalize)
set.seed(1234)

# Applying normalization
sleep_normal <- apply(sleep, 2, function(x){
  bestNormalize(x)$x.t
  }
)

# Convert to dataframe for ggpairs
sleep_normal <- as.data.frame(sleep_normal)

# Obtain transforms, to be used during backtransformation
sleep_transforms <- lapply(1:ncol(sleep), function(i){
    bestNormalize(
      sleep[,i]
      ) # obtain all output from function
  })
```

```
# Check out distributions again
ggpairs(sleep_normal)
```



All of our variables except for `Awakenings` are now normally distributed. The skew is gone. The correlation values between `Sleep.efficiency` and the independent variables have gone down. `Light.sleep.percentage` and `Deep.sleep.percentage` still have the highest correlation to the dependent variable `Sleep.efficiency`. All of these values are statistically significant except for the relationship between `Awakenings` and `Deep.sleep.percentage`.

We will now begin step 3, which involves performing the regression and analyzing the model summary.

```
# Creating linear model
lm_sleep = lm(Sleep.efficiency ~ ., sleep_normal)
summary(lm_sleep)
```

```
##
## Call:
## lm(formula = Sleep.efficiency ~ ., data = sleep_normal)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10098 -0.48856 -0.00382  0.46121  2.82687
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.01091    0.03311   0.330  0.74184
## REM.sleep.percentage     0.23848    0.03909   6.101 2.59e-09 ***
## Deep.sleep.percentage    0.22906    0.09678   2.367  0.01845 *
```

```
## Light.sleep.percentage -0.27737    0.09386  -2.955  0.00332 **
## Awakenings             -0.52938    0.03961 -13.366  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6504 on 381 degrees of freedom
## Multiple R-squared:  0.5749, Adjusted R-squared:  0.5705
## F-statistic: 128.8 on 4 and 381 DF,  p-value: < 2.2e-16
```

All of our variables appear to be statistically significant under $\alpha = 0.05$, so none of these variables should be removed from the regression as of now.

Each coefficient value represents the increase or decrease in `Sleep.efficiency` if the corresponding independent variable increases by one standard deviation. For example, when `REM.sleep.percentage` increases by one standard deviation, `Sleep.efficiency` increases by 0.23848.

The R^2 value is equal to **0.5749**. This means that 57.49% of the variance in `Sleep.efficiency` can be explained by the independent variables in the model.

We can now proceed to step 4, which involves checking these variables for multicollinearity. If we find multicollinearity, we may have to remove variables present in the model.

```
# Computing VIF values
vif(lm_sleep)
```

```
##   REM.sleep.percentage  Deep.sleep.percentage Light.sleep.percentage
##               1.209816              8.008846              7.522622
##              Awakenings
##               1.096146
```

Whoa! The `Deep.sleep.percentage` and `Light.sleep.percentage` variables exceed our VIF threshold of 5.

Because the correlation between `Sleep.efficiency` and `Light.sleep.percentage` is highest, I am going to remove `Deep.sleep.percentage` from the model and see if we obtain better VIF values.

```
# Creating linear model without Deep.sleep.percentage
lm_sleep_refined = lm(Sleep.efficiency ~ ., sleep_normal[-3])
summary(lm_sleep_refined)
```

```
##
## Call:
## lm(formula = Sleep.efficiency ~ ., data = sleep_normal[-3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.12263 -0.48476 -0.01841  0.47413  2.71014
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.01137    0.03331   0.341    0.733
## REM.sleep.percentage     0.20714    0.03700   5.599 4.13e-08 ***
## Light.sleep.percentage -0.48185    0.03689 -13.061  < 2e-16 ***
## Awakenings             -0.53405    0.03980 -13.420  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6543 on 382 degrees of freedom
## Multiple R-squared:  0.5687, Adjusted R-squared:  0.5653
```

```
## F-statistic: 167.9 on 3 and 382 DF,  p-value: < 2.2e-16
```

Again, all of our variables appear to be statistically significant under $\alpha = 0.05$. In fact, they are all significant under the 0 code.

The new R^2 value is equal to **0.5687**. This means that 56.87% of the variance in `Sleep.efficiency` can be explained by the independent variables in the model. This is slightly worse than our R^2 value last time.

We will now compute the new VIF values of this model.

```
# Computing VIF values
vif(lm_sleep_refined)
```

```
##   REM.sleep.percentage Light.sleep.percentage             Awakenings
##               1.070990               1.148344               1.093425
```

This looks good! All our VIF values are now under 5.

We may skip step 5, removing non-significant predictors, because all of our predictors are significant.

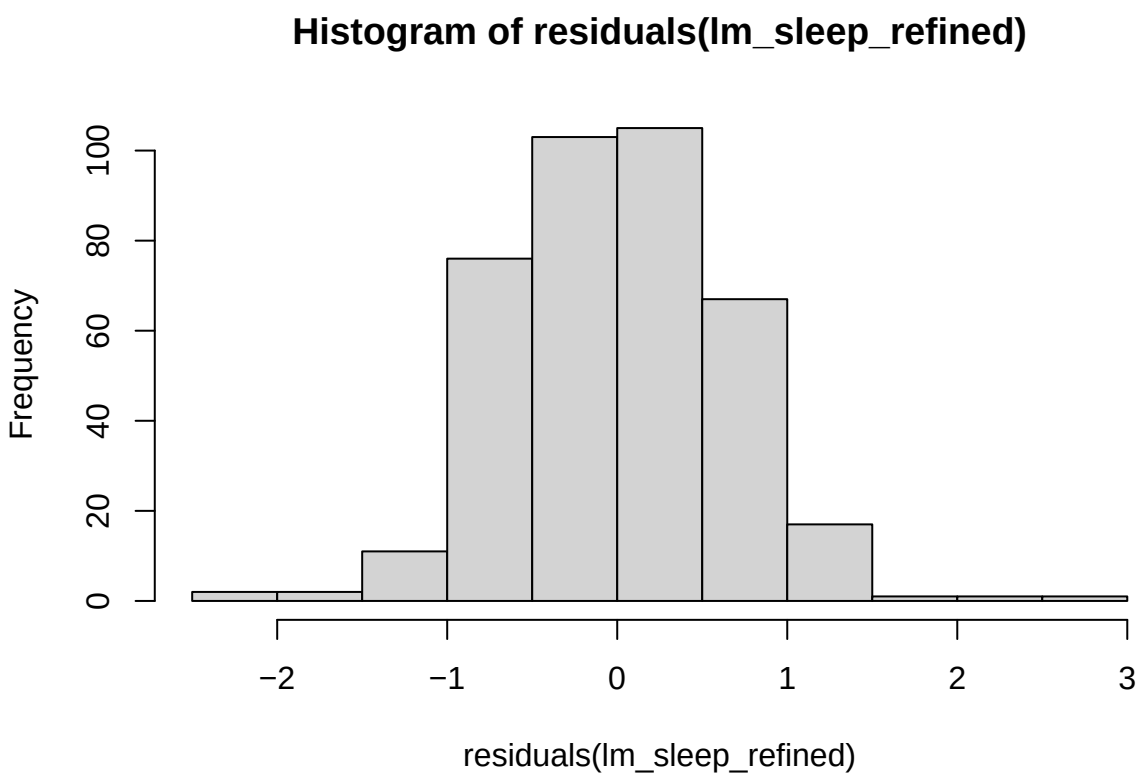
Finally, we can proceed to step 6, which involves checking the normality of residuals (homoskedasticity) and correlation between residuals and predicted values.

We can check the normality of the residuals in 3 ways:

1. Plotting a histogram of the residuals
2. Shapiro-Wilk test
3. QQ plot

First, we can visually check normality by plotting the residuals on a histogram:

```
# Histogram of residuals
hist(residuals(lm_sleep_refined))
```



At first glance, these residuals appear to be normally distributed. But let's investigate further.

To check normality using the Shapiro-Wilk test:

- **Null Hypothesis:** Residuals **are** normally distributed ($p > 0.05$)
- **Alternative Hypothesis:** Residuals are **not** normally distributed ($p < 0.05$)

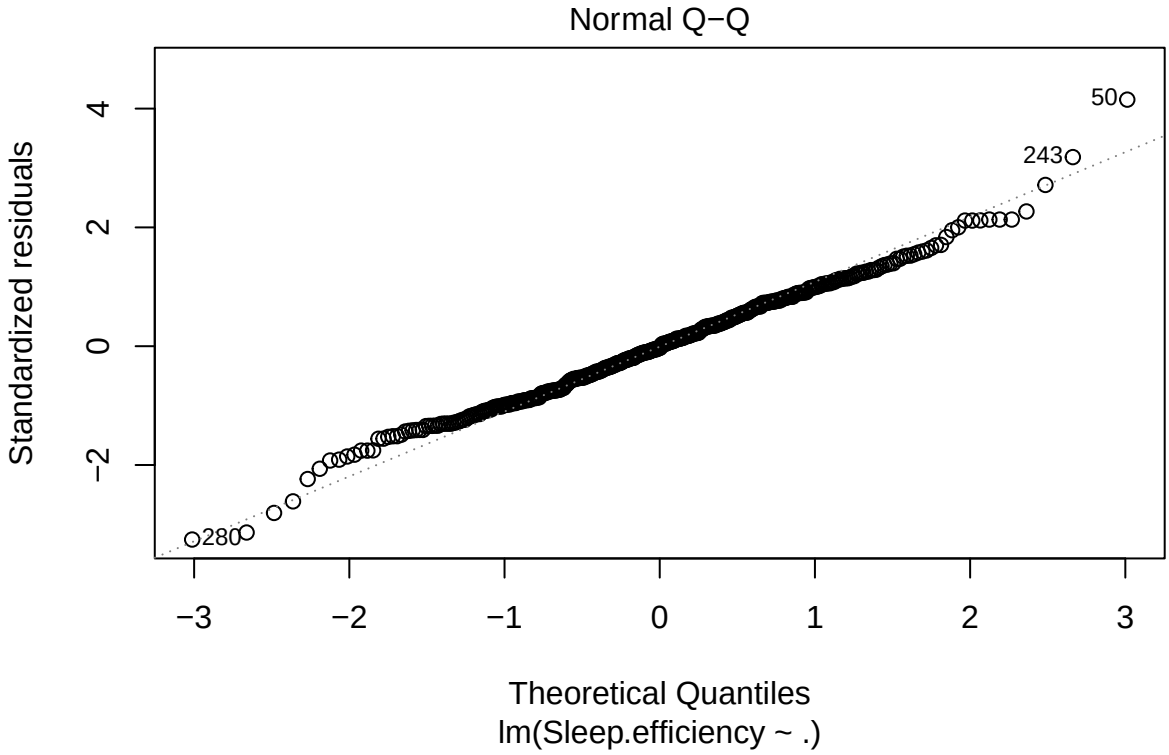
```
# Shapiro-Wilk test
shapiro.test(residuals(lm_sleep_refined))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm_sleep_refined)
## W = 0.99227, p-value = 0.04281
```

The p-value is just barely under 0.05, indicating that the residuals might not be normally distributed!

Let's now assess the linearity of the QQ plot:

```
# Plot fitted values on residuals
plot(lm_sleep_refined, which = 2)
```



The ends of the QQ plot do not lie on the line, indicating there is a collection of residuals that are not normally distributed.

Let's attempt to fix the normality of our residuals by removing outliers in this dataset. Similar to the lab, I will consider residuals of magnitude greater than 2 to be outliers. I will add a column to the sleep dataset called `outliers` which will be used to filter outliers (which correspond to a value of 1).

```
# Adding a variable called outliers to sleep dataset
sleep_normal <- sleep_normal %>%
  mutate(outliers = ifelse(
    residuals(lm_sleep_refined) <= -2 | residuals(lm_sleep_refined) >= 2, 1, 0))
```

```r
# Removing outliers
sleep_normal <- sleep_normal %>%
  filter(outliers == 0) %>%
  select(!outliers)

# Creating linear model without outliers
lm_sleep_outliers <- lm(Sleep.efficiency ~ ., sleep_normal[-3])
summary(lm_sleep_outliers)
```

```
##
## Call:
## lm(formula = Sleep.efficiency ~ ., data = sleep_normal[-3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80078 -0.47019 -0.00879  0.46051  1.77884
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              0.009684   0.031481   0.308    0.759
## REM.sleep.percentage     0.195042   0.034851   5.596 4.21e-08 ***
## Light.sleep.percentage -0.464606   0.034813 -13.346  < 2e-16 ***
## Awakenings             -0.539362   0.037457 -14.400  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6152 on 378 degrees of freedom
## Multiple R-squared:  0.5941, Adjusted R-squared:  0.5909
## F-statistic: 184.5 on 3 and 378 DF,  p-value: < 2.2e-16
```

Our R^2 value seems to have improved slightly. It is now equal to **0.5941**.

We will now compute the new VIF values of this model.

```r
# Computing VIF values
vif(lm_sleep_outliers)
```

```
##   REM.sleep.percentage Light.sleep.percentage             Awakenings
##               1.071872               1.150605               1.094784
```

Looks good!

Again, we do not need to remove insignificant predictors because there are none.

Let's again run through the 3 steps of determining normality of the residuals.

```r
# Histogram of residuals
hist(residuals(lm_sleep_outliers))
```

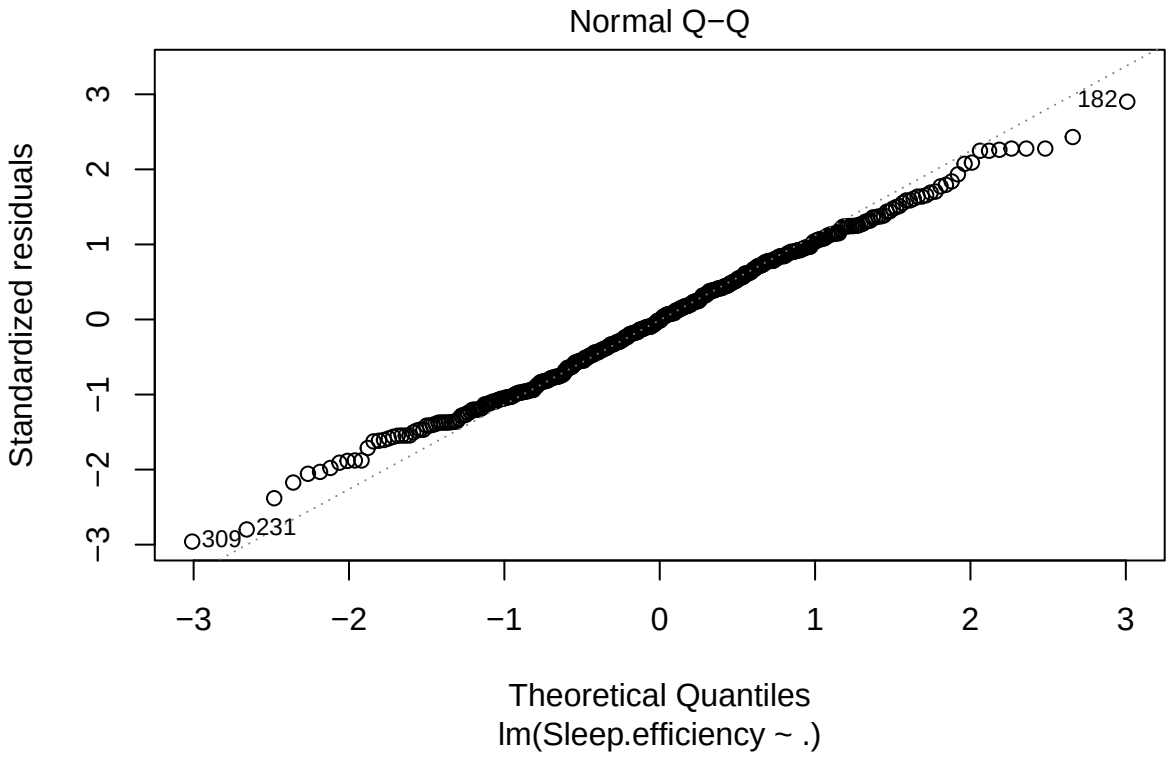
These appear to be normally distributed at first glance.

```
# Shapiro-Wilk test
shapiro.test(residuals(lm_sleep_outliers))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(lm_sleep_outliers)
## W = 0.99615, p-value = 0.4832
```

This p-value is now greater than 0.05, so we can accept the Null Hypothesis that these residuals **are** normally distributed!

```
# Plot fitted values on residuals
plot(lm_sleep_outliers, which = 2)
```

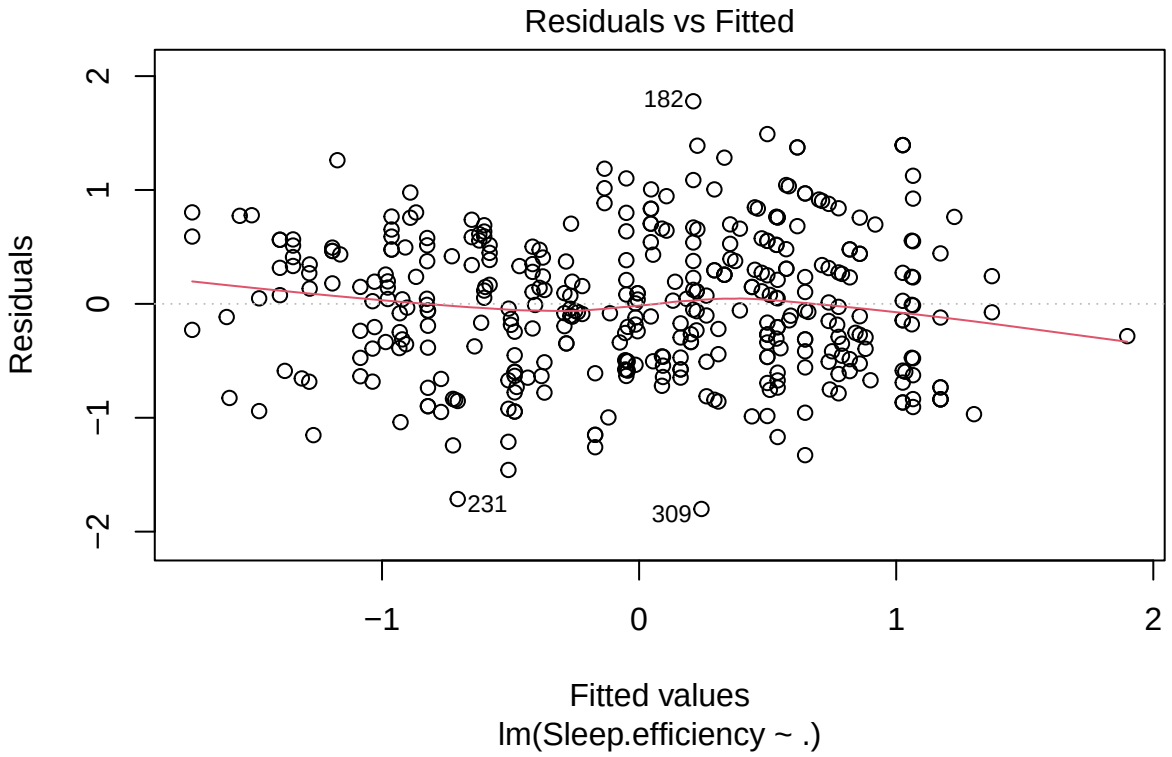


This QQ plot appears slightly more linear than the last. However, it is still not perfect.

We will proceed with `lm_sleep_outliers` as our final version of the model. The residuals are normally distributed based on this analysis!

We can see where we are underpredicting and overpredicting by plotting the residuals and the fitted values:

```
# Plotting residuals and fitted values
plot(lm_sleep_outliers, which = 1)
```



We seem to be underpredicting for lower fitted values and overpredicting for higher fitted values. Otherwise, these residuals are hovering around 0.

To conclude our analysis, we can compute the **RMSE** (root mean squared error) of this linear model.

```
# Compute RMSE
sqrt(mean(residuals(lm_sleep_outliers)^2))
```

```
## [1] 0.6119604
```

The RMSE of this model is equal to **0.6119604**. This means that on average, our predicted values are about 0.6119604 standard deviations away from the actual values.

Data Visualization

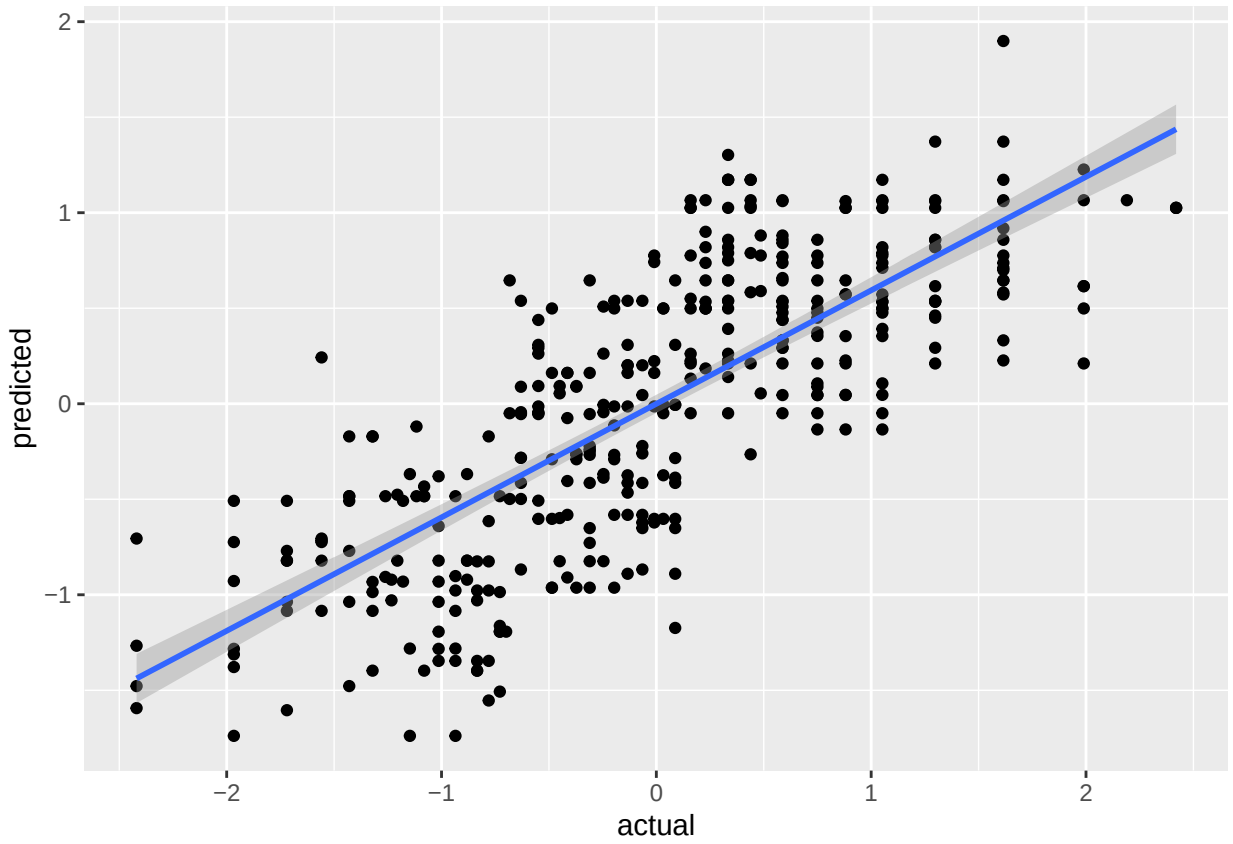
To conclude, I will plot:

- The predicted values on the actual values
- The backtransformed predicted distances on the actual distances

```
# Plotting predicted values on actual values
final_df <- data.frame(
  actual = sleep_normal$Sleep.efficiency,
  predicted = predict(lm_sleep_outliers)
)

ggplot(data = final_df, aes(x = actual, y = predicted)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



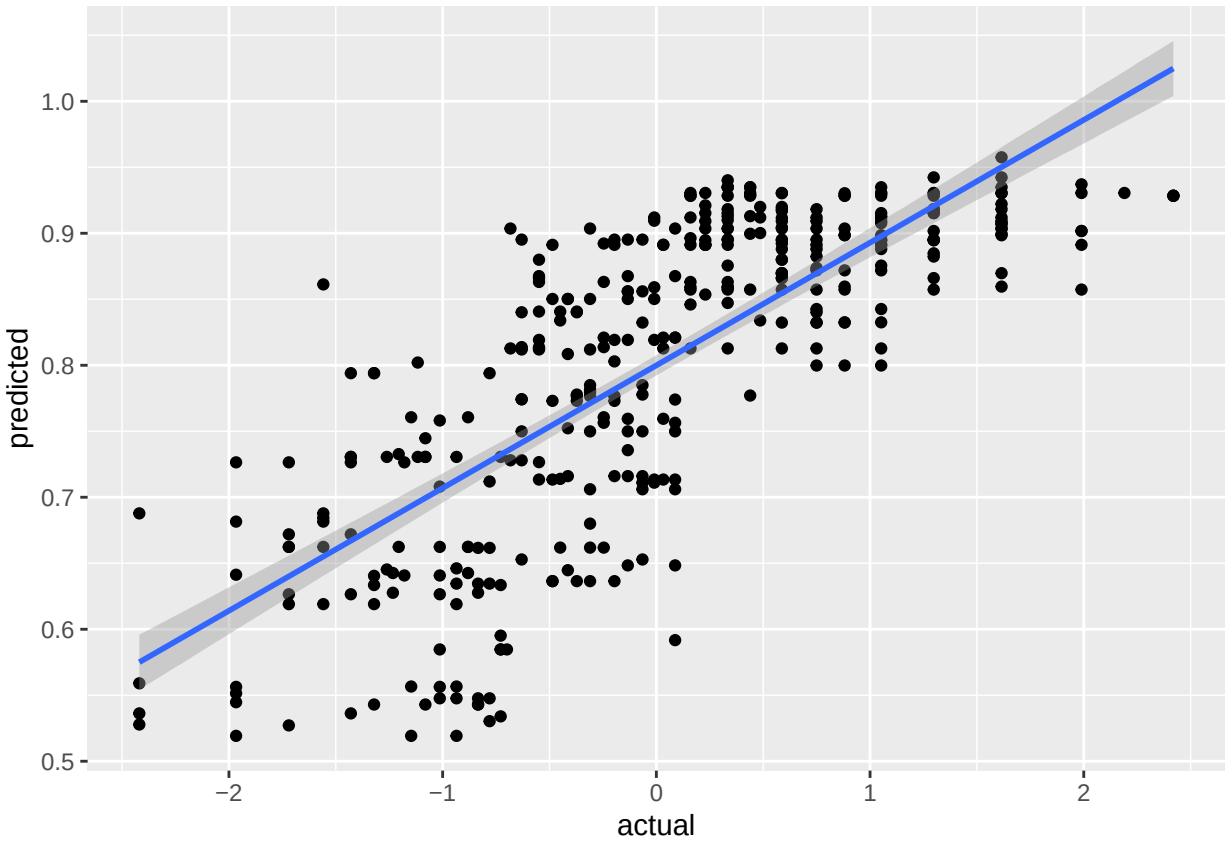

```r
# Perform backtransformation on predicted values
predicted_values <- predict(
  sleep_transforms[[1]],
  # corresponding {bestNormalize} output
  newdata = predict(lm_sleep_outliers),
  # get predictions from model
  inverse = TRUE
  # backtransform
)

# Obtain actual values
actual_values <- sleep_normal$Sleep.efficiency

# Plot data frame
plot_df <- data.frame(
  predicted = predicted_values,
  actual = actual_values
)

ggplot(data = plot_df, aes(x = actual, y = predicted)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Discussion

The purpose of this analysis was to determine if **sleep efficiency** could be predicted using numeric variables within the `Sleep_Efficiency.csv` dataset. Specifically, these independent variables were REM sleep percentage, deep sleep percentage (removed), light sleep percentage, and the number of awakenings. Our **Formal Null Hypothesis** specified that the coefficients in our linear regression model were all equal to 0, while our **Formal Alternative Hypothesis** specified that at least one of these coefficients was not equal to 0. Through the analysis performed above, the coefficients were determined to be non-zero:

$$Sleep\,Efficiency = 0.009684 + 0.195042 * REM\,Pct - 0.464606 * Light\,Sleep\,Pct - 0.539362 * Awakenings + \varepsilon$$

Therefore, we may accept the **Alternative Hypothesis** that **sleep efficiency is related to these independent variables.**

The final model `lm_sleep_outliers` was obtained by normalizing the variables, removing insignificant variables, removing instances of multicollinearity, and removing outliers. This model had an R^2 **value** of **0.5941**, meaning that 59.41% of the variance in `Sleep.efficiency` can be explained by the independent variables in the model. This is a pretty good R^2 value, so this relationship certainly deserves our attention. Our residuals were also determined to be normally distributed, which is favorable for our model. Finally, the **RMSE** of this model was equal to **0.6119604**, meaning that on average, our predicted values were about 0.6119604 standard deviations away from the actual values.

In summary, this model is a good one, and it should be ready to deploy in a real world setting.