Analyzing Sleep Through Smart Technology: A Case Study of Will Foote's Fitbit Data

[CODE FILE]

William Foote & Kyle Fang

6/12/2020

Loading sleep data

```
library(readxl)
library(chron)
```

```
## NOTE: The default cutoff when expanding a 2-digit year
## to a 4-digit year will change from 30 to 69 by Aug 2020
## (as for Date and POSIXct in base R.)
```

```
sleep <- read_excel("combined_data_final.xlsx")
class(sleep$Date)
```

```
## [1] "POSIXct" "POSIXt"
```

Clean start and end times for sleep

Need get_miltime function

```
get_miltime <- function (x) {
  if (grepl("-", x)) {
    x <- gsub("-", "", x)
    x <- unlist(strsplit(x, " "))
    if(length(x) > 2) {
      x <- paste(x[2:3], collapse = "")
    } else {
      x <- x[2]
    }
    if (grepl("AM", x)) {
      temp <- unlist(strsplit(x, ""))
      temp <- paste(temp[1:2], collapse = "")
      if (temp != 12) {
         x <- gsub("AM", "", x)
```

```
        x <- paste(x, ":", "00", sep = "")
      } else {
        x <- gsub("AM", "", x)
        split_minhr <- unlist(strsplit(x, ":"))
        hours <- as.numeric(split_minhr[1])
        mins <- split_minhr[2]
        mil_hr <- hours - 12
        x <- paste(mil_hr, ":", mins, ":", "00", sep = "")
      }
    }
    if (grepl("PM", x)) {
      x <- gsub("PM", "", x)
      temp <- unlist(strsplit(x, ""))
      temp <- paste(temp[1:2], collapse = "")
      if (temp != "12") {
        split_minhr <- unlist(strsplit(x, ":"))
        hours <- as.numeric(split_minhr[1])
        mins <- split_minhr[2]
        mil_hr <- hours + 12
        x <- paste(mil_hr, ":", mins, ":", "00", sep = "")
      } else {
        x <- paste(x, ":", "00", sep = "")
      }
    }
  }
  if (grepl("/", x)) {
    x <- gsub("/", "", x)
    x <- unlist(strsplit(x, " "))
    x <- x[2]
  }
  x
}
```

Use get_miltime function

```
start_mil_time <- character(76)
end_mil_time <- character(76)
for (i in seq_len(length(sleep$Start.Time))) {
  start_mil_time[i] <- get_miltime(sleep$Start.Time[i])
}
for (i in seq_len(length(sleep$End.Time))) {
  end_mil_time[i] <- get_miltime(sleep$End.Time[i])
}
for(i in 60:76) {
  if (!is.na(start_mil_time[i])) {
    start_mil_time[i] <- paste(start_mil_time[i], ":", "00", sep = "")
  }
}
for(i in 60:76) {
  if (!is.na(end_mil_time[i])) {
    end_mil_time[i] <- paste(end_mil_time[i], ":", "00", sep = "")
```

```
  }
}

start_asTime <- as.numeric(chron(times = start_mil_time))
end_asTime <- as.numeric(chron(times = end_mil_time))
```

Clean the times and dates and make final data frame

```
sleep_final <- sleep[, c(1, 21, 15, 27, 29, 30)]
sleep_final$start_mil_time <- start_mil_time
sleep_final$end_mil_time <- end_mil_time
sleep_final$overall_score <- as.numeric(sleep_final$overall_score)
sleep_final$Number.of.Awakenings <- as.numeric(sleep_final$Number.of.Awakenings)
sleep_final$restlessness <- as.numeric(sleep_final$restlessness)
sleep_final$start_asTime <- start_asTime
sleep_final$end_asTime <- end_asTime

six_oclock <- as.numeric(chron(times = "18:00:00"))
sleep_final$start_asTime <- ifelse(start_asTime > .88, start_asTime - six_oclock,
                                   start_asTime + .25)
# Times greater than 12 AM: add a day (1) then subtract hours of 6 PM (18:00 equals .75 of
# a day)... i.e. add .25 of a day, or 6 hours, to start_asTime[i] to get time past 6 PM

sleep_final$end_asTime <- end_asTime + .25

# Convert end_asTime to hours past 6 PM for uniformity.

prob_rem <- with(sleep, as.numeric(Minutes.REM.Sleep)/as.numeric(Minutes.Asleep))
# Above produces warning that NAs are introduced by coercion, because there
# are NAs in these data.
sleep_final$prob_rem <- prob_rem
```

Exploring the data

Look for transformations needed

Y variable

```
sleep_final_final <- sleep_final[, c(1, 2, 3, 4, 9, 10, 11)]
sleep_lm <- lm(overall_score ~ . - Date, data = sleep_final_final)
require(car)
```

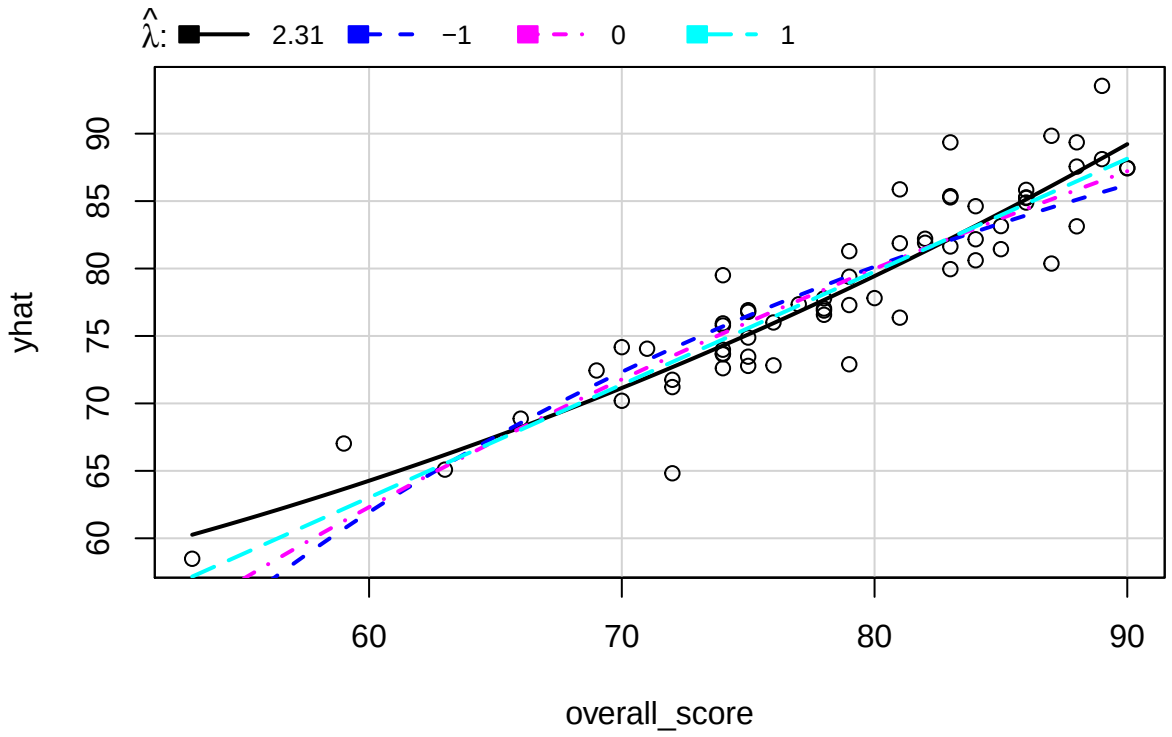
```
## Loading required package: car
```

```
## Loading required package: carData
```

```r
library(alr3)
summary(powerTransform(sleep_lm))
```

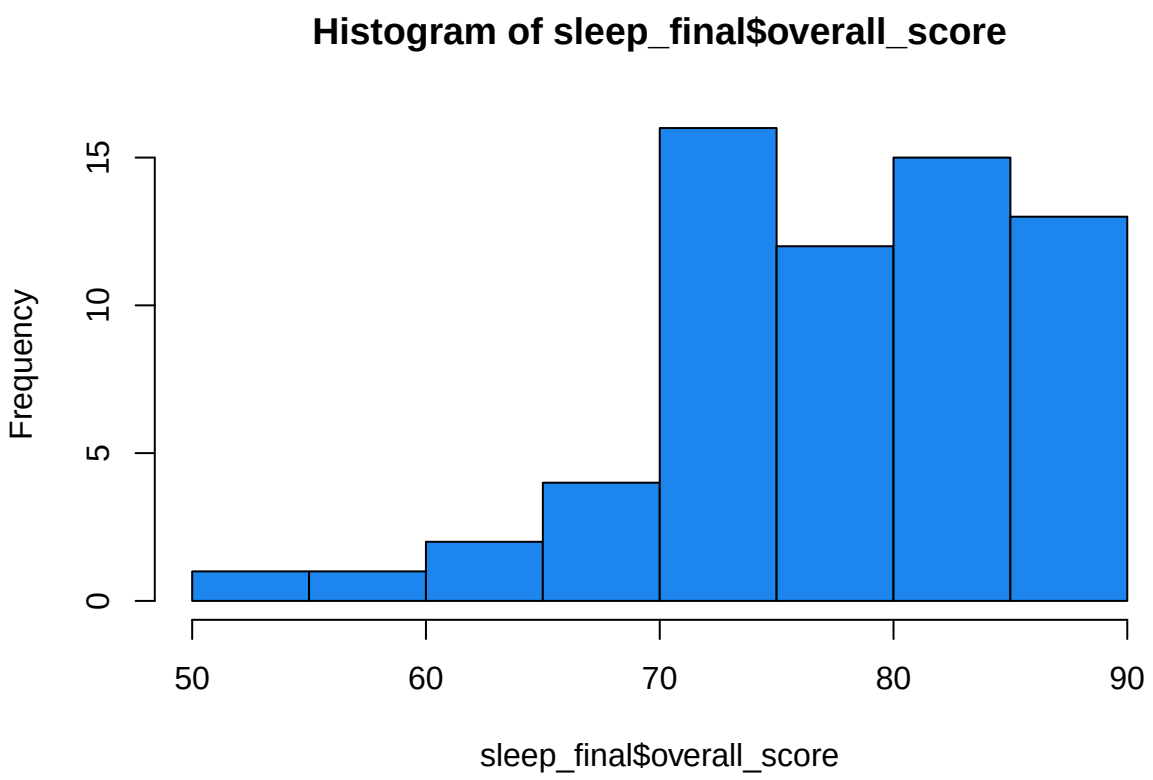
```
## bcPower Transformation to Normality
##    Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Y1    2.9266           2       1.5946       4.2586
##
## Likelihood ratio test that transformation parameter is equal to 0
##  (log transformation)
##                            LRT df       pval
## LR test, lambda = (0) 22.91447  1 1.6937e-06
##
## Likelihood ratio test that no transformation is needed
##                            LRT df      pval
## LR test, lambda = (1) 9.456782  1 0.0021037
```

```r
invResPlot(sleep_lm)
```




```
##      lambda      RSS
## 1  2.312679 454.1058
## 2 -1.000000 628.1888
## 3  0.000000 533.8310
## 4  1.000000 477.8276
```

```r
hist(sleep_final$overall_score, col = "dodgerblue2")
```



X variables

```
summary(powerTransform(as.matrix(sleep_final_final[, c(3:7)]) ~ 1))
```

```
## bcPower Transformations to Multinormality
##                       Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## Number.of.Awakenings    0.7599           1       0.1437       1.3760
## restlessness            0.0111           1      -1.0251       1.0474
## start_asTime            0.8799           1       0.0913       1.6685
## end_asTime             -0.0429           1      -1.6813       1.5954
## prob_rem                0.8028           1      -0.0311       1.6368
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                   LRT df    pval
## LR test, lambda = (0 0 0 0 0) 14.01569  5 0.01551
##
## Likelihood ratio test that no transformations are needed
##                                   LRT df    pval
## LR test, lambda = (1 1 1 1 1) 5.884141  5 0.31766
```

Do transforms

```
sleep_final_final$overall_score <- sleep_final_final$overall_score ^ 3
```

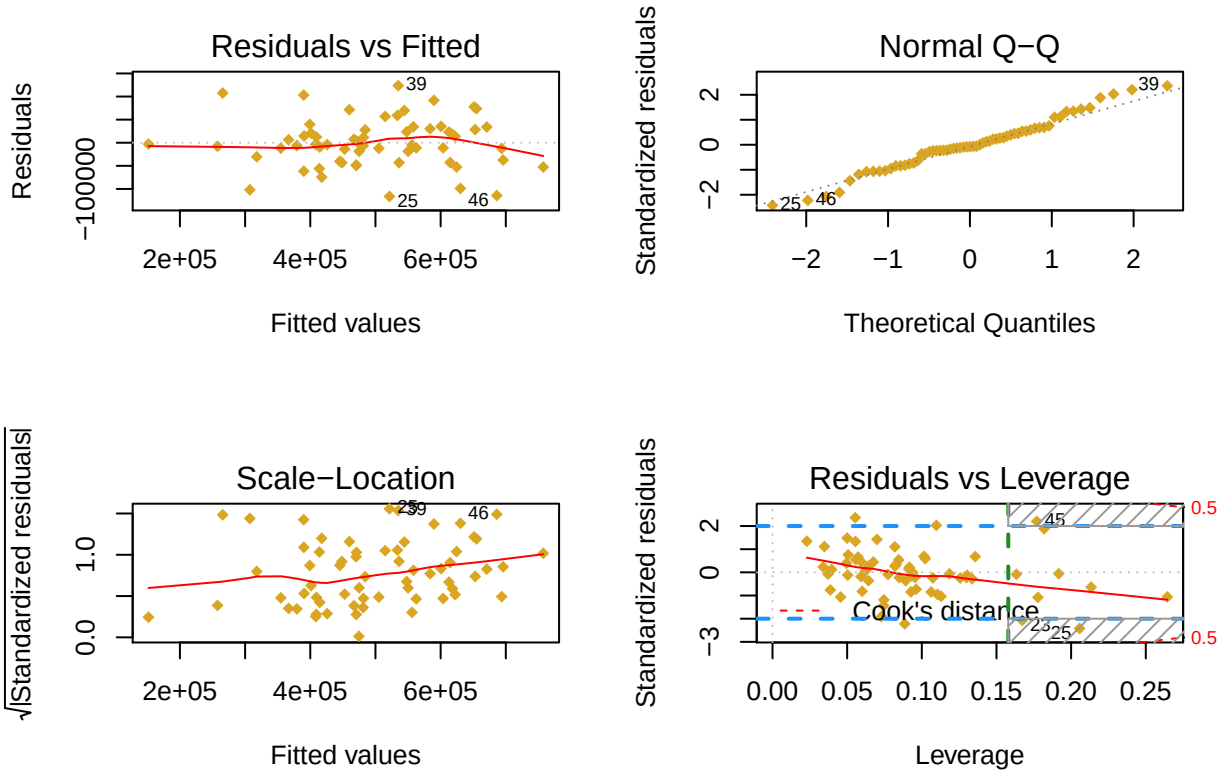
Fit the new model, output the summary

```
sleep_lm_2 <- lm(overall_score ~ . - Date, data = sleep_final_final)
summary(sleep_lm_2)
```

```
##
## Call:
## lm(formula = overall_score ~ . - Date, data = sleep_final_final)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116274  -33880   -4058   28141  123543
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            183600.5    67275.3   2.729 0.008431 **
## Number.of.Awakenings     5428.1      971.4   5.588 6.74e-07 ***
## restlessness         -2191853.9   303169.1  -7.230 1.32e-09 ***
## start_asTime          -539076.9   154754.9  -3.483 0.000958 ***
## end_asTime             563237.1   124335.7   4.530 3.06e-05 ***
## prob_rem               881436.2   143291.4   6.151 8.12e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53860 on 57 degrees of freedom
##   (13 observations deleted due to missingness)
## Multiple R-squared:  0.8452, Adjusted R-squared:  0.8316
## F-statistic: 62.24 on 5 and 57 DF,  p-value: < 2.2e-16
```

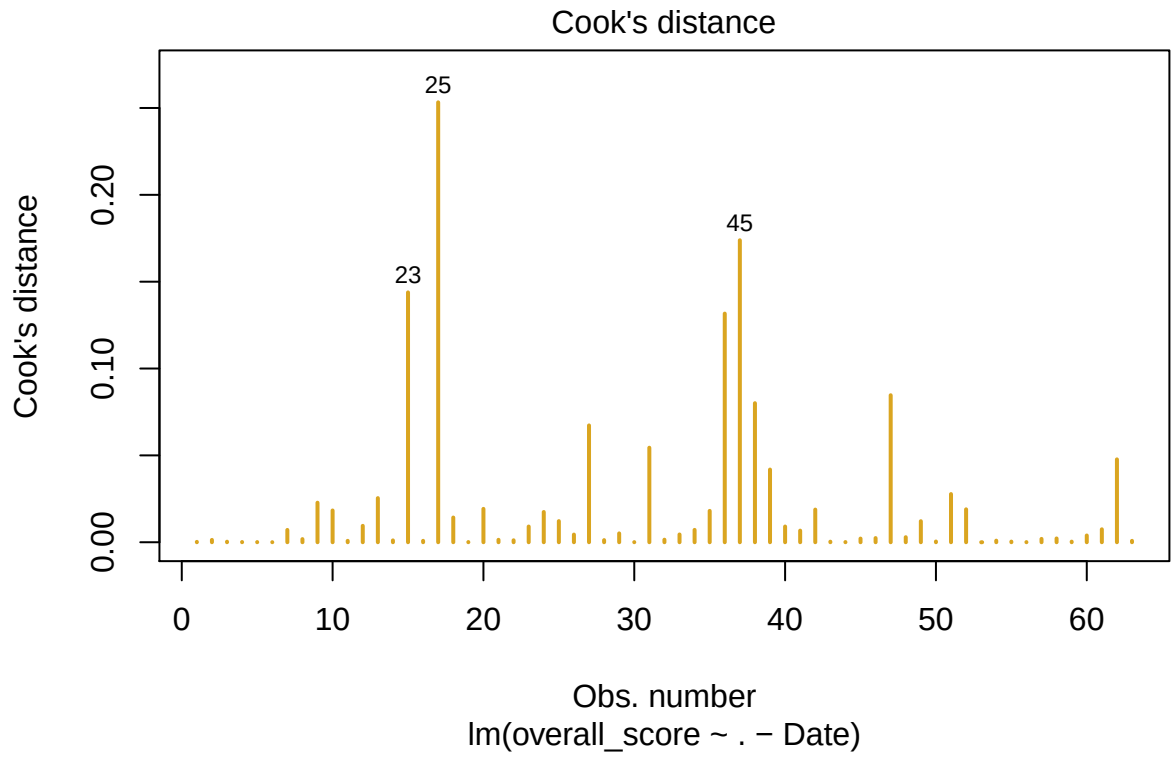
Check the diagnostics for validity/model weaknesses

```
par(mfrow = c(2, 2))
plot(sleep_lm_2, col = "goldenrod", pch = 18, which = c(1:3))
n <- dim(sleep_final_final)[1]
plot(sleep_lm_2, col = "goldenrod", pch = 18, which = 5)
abline(v = 2 * (5 + 1) / n, lty = 2, col = "forestgreen", lwd = 2)
abline(h = c(2, -2), lty = 2, col = "dodgerblue", lwd = 2)
rect(xleft = 2 * (5 + 1) / n, xright = .3, ybottom = c(2, -3.5), ytop = c(3.5, -2),
     border = NULL, col = "gray60", density = 15)
```



Bad Leverage Points and Influential Points

```
plot(sleep_lm_2, which = 4, col = "goldenrod", lwd = 2)
```



```r
hist(cooks.distance(sleep_lm_2), col = "dodgerblue3",
     main = "Distribution of Cook's Distances:\nWhich points, if removed, would change the regression mo
     xlab = "Cook's Distance",
     cex.main = .89)
```