Case Study

Filippo Gambarota

2023-04-24

Introduction

The dataset `kid_school_inv.rds` contains data about performance of primary school children (6-12 years old) on the final exam (`y`, where `1 = passed` and `0 = failed`). There are some predictors in particular:

- the child age (`age`) in years
- the parents judgement about the child being involved in school related activities (`parent_score`) expressed in a scale from 0 (no involvement) to 10 (completely involved)
- the teacher judgement about the child being involved in school related activities (`teacher_score`) expressed in a scale from 0 (no involvement) to 10 (completely involved)
- if the child is attending to after-school activities such as private tutoring (`after_school`)

The idea is to understand the contribution of each variable to the probability of passing the exam building an additive model (i.e., no interactions) and interpreting the effects along with a general diagnostic.

Steps

1. Importing the `kid_school_inv.rds`
2. Check the dataset structure, type of variables, missing values, etc.
    1. exclude children with `NA` on the `after_school`, `age` or `y` variables
    2. impute the average value on other `NA` values
3. Mean-center the `age` variable and transform`after_school` into factor and set contrasts to `c(-0.5, 0.5)`
4. Analyze the relationships between predictors:
    1. Represent graphically the relationship between `after_school` and `teacher_score` and briefly comment the result
    2. Represent graphically the relationship between `after_school` and `parent_score` and briefly comment the result
    3. Represent graphically the relationship between `age` and `after_school` and briefly comment the result
5. Represent graphically the relationship between each predictor and the `y` variable
6. If you see some strange values on predictor try to solve the problem, are they impossible or just extreme?
7. Build a Gaussian GLM additive model with all variables:
    1. Interpret the results
    2. Plot the model effects

    3. Plot the model diagnostics and comment the results

8. Build a Binomial GLM with a logit link function:
    1. Build a null model
    2. Build a full model
    3. Build a series of nested models from the null to the full model
9. Check the model diagnostic:
    1. Plot the raw residuals against the fitted values and briefly comment the result
    2. Plot the raw residuals against each (numerical) predictor and briefly comment the result
    3. Compute the error rate for each model of the point 7 and briefly comment the result
10. Interpret the full model results:
    1. Briefly describe the meaning of each model coefficient
    2. Compute and interpret the marginal effect (divide by 4 rule) and the average marginal effect for each numerical predictor
    3. Compute and interpret the odds ratio for categorical predictors
11. Compare using a likelihood ratio test the nested models created at point 7 and report the best model
12. Plot the effects of the best model created at point 10

Solutions

1

```
dat <- readRDS("data/kid_school_inv.rds")
```

2

Dataset structure and type of variables:

```
str(dat) # dataset structure
```

```
## ’data.frame’:    100 obs. of  6 variables:
##  $ id           : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ age          : num  10 11 11 7 6 7 12 9 10 8 ...
##  $ after_school : Factor w/ 2 levels "no","yes": 2 1 1 2 1 2 1 1 1 1 ...
##   ..- attr(*, "contrasts")= num [1:2, 1] 0.5 -0.5
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:2] "no" "yes"
##   .. .. ..$ : NULL
##  $ parent_score : num  1 7 6 9 3 3 4 0 4 NA ...
##  $ teacher_score: int  9 0 7 4 8 6 1 5 9 6 ...
##  $ y            : int  1 0 1 1 0 1 0 1 1 0 ...
```

```
sapply(dat, class) # variables
```

```
##            id           age  after_school  parent_score teacher_score 
##     "integer"     "numeric"      "factor"     "numeric"     "integer" 
##             y 
##     "integer"
```

Number of NA values:

```
sapply(dat, function(col) sum(is.na(col)))
```

```
##            id           age  after_school  parent_score teacher_score 
##             0             2             2             3             4 
##             y 
##             4
```

Removing children with NA:

```
to_remove <- !complete.cases(dat[, c("after_school", "age", "y")])
dat <- dat[!to_remove, ]
```

Imputing values:

```
means <- sapply(dat, mean, na.rm = TRUE)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
dat$parent_score[is.na(dat$parent_score)] <- means["parent_score"]
dat$teacher_score[is.na(dat$teacher_score)] <- means["teacher_score"]
```

3

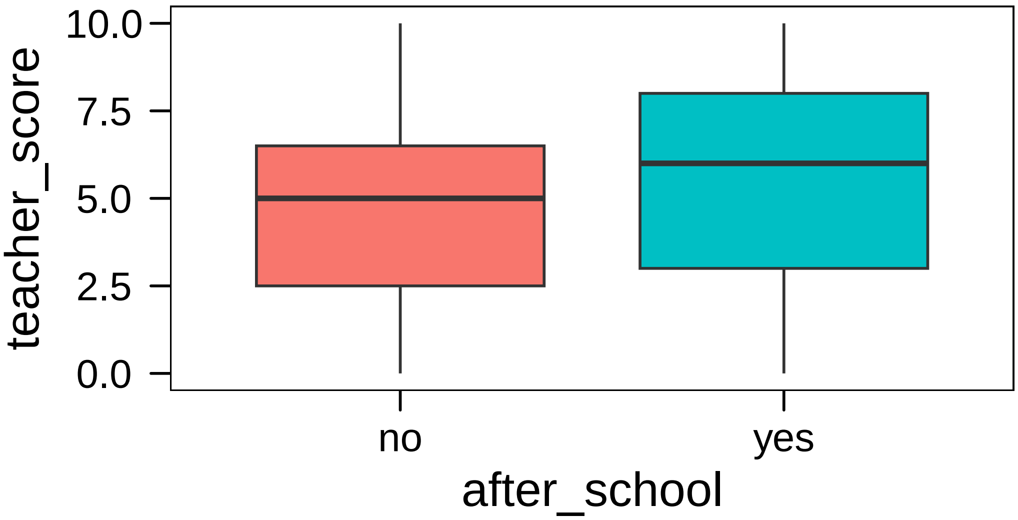
For teacher_score and parent_score the 0 point is meaningful thus we can avoid centering. For the age the 0 point is not meaningful thus we center it:

```
dat$age0 <- dat$age - mean(dat$age)
dat$after_school <- factor(dat$after_school)
contrasts(dat$after_school) <- contr.sum(2)/2
```

4

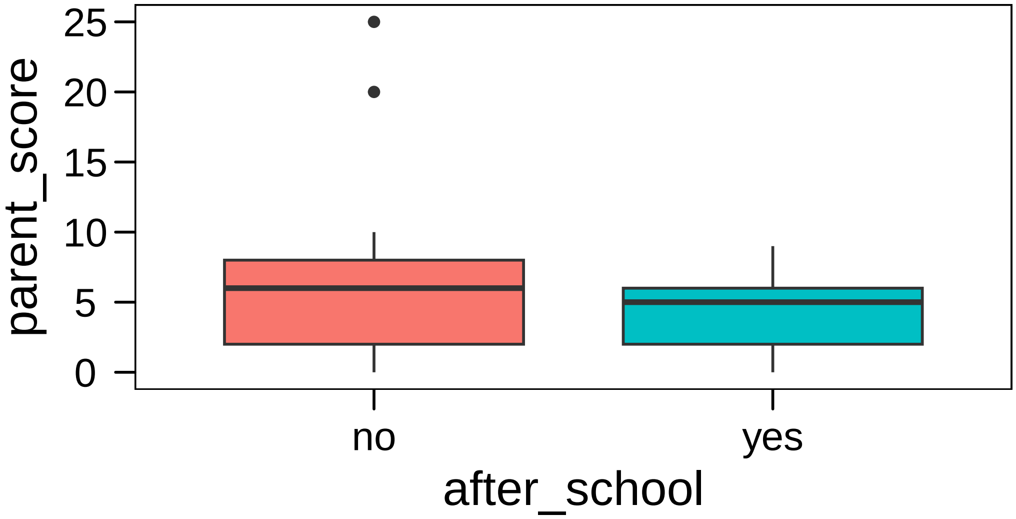
Represent graphically the relationship between after_school and teacher_score and briefly comment the result:

```
dat |>
    ggplot(aes(x = after_school, y = teacher_score, fill = after_school)) +
    geom_boxplot(show.legend = FALSE) +
    mytheme()
```



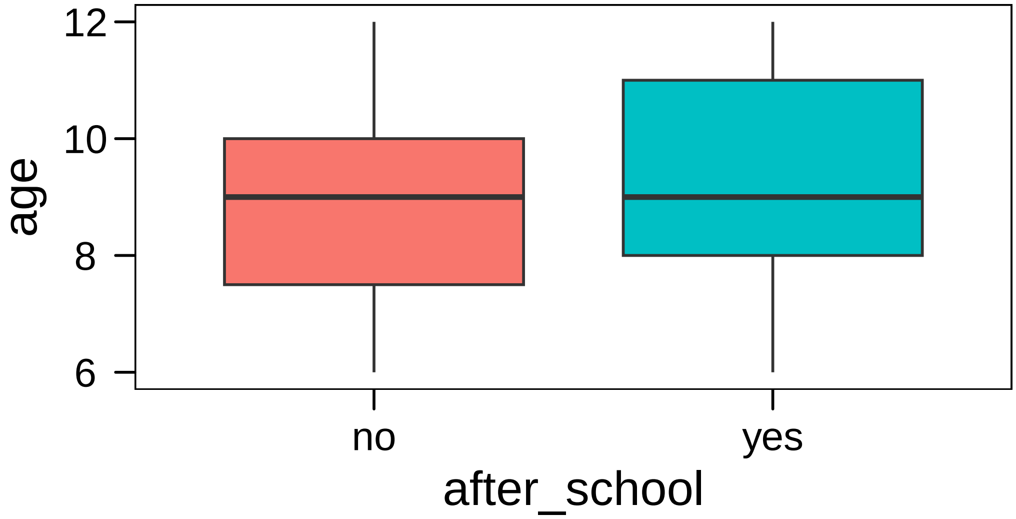
Represent graphically the relationship between `after_school` and `parent_score` and briefly comment the result:

```
dat |>
    ggplot(aes(x = after_school, y = parent_score, fill = after_school)) +
    geom_boxplot(show.legend = FALSE) +
    mytheme()
```



Represent graphically the relationship between `age` and `after_school` and briefly comment the result:

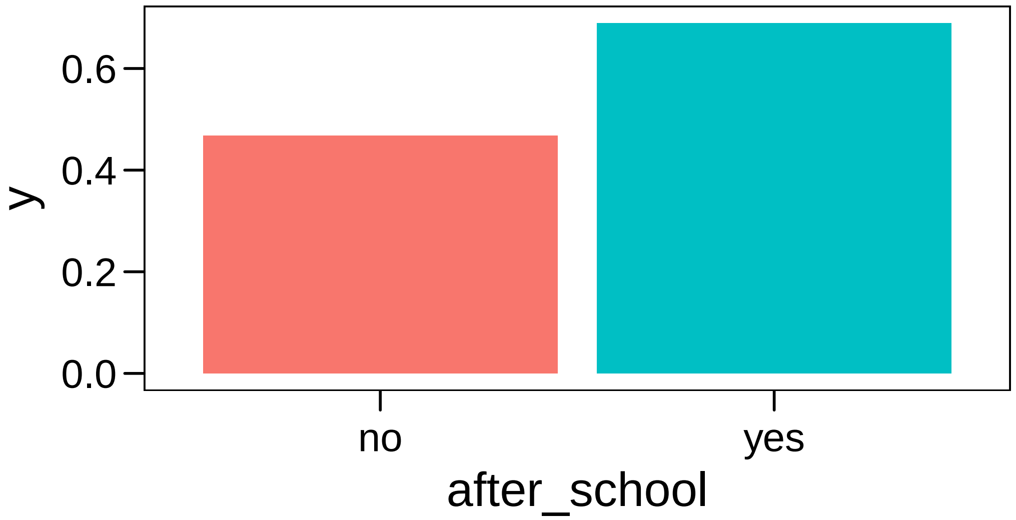
```
dat |>
    ggplot(aes(x = after_school, y = age, fill = after_school)) +
    geom_boxplot(show.legend = FALSE) +
    mytheme()
```



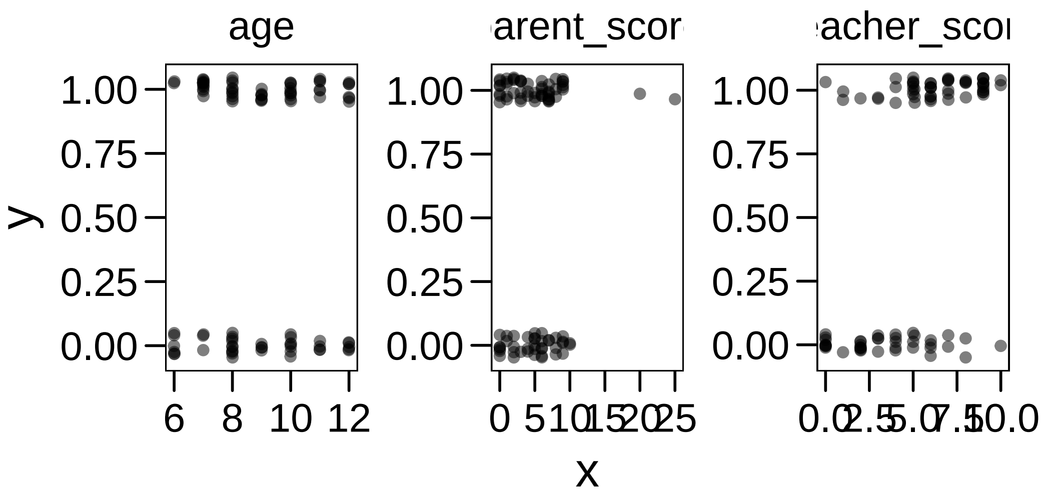
5

Represent graphically the relationship between each predictor and the y variable:

```
dat |>
    group_by(after_school) |>
    summarise(y = mean(y)) |>
    ggplot(aes(x = after_school, y = y, fill = after_school)) +
    geom_col(show.legend = FALSE) +
    mytheme()
```



```
dat |>
    pivot_longer(c(parent_score, teacher_score, age), names_to = "var", values_to = "x") |>
    ggplot(aes(x = x, y = y)) +
    geom_point(position = position_jitter(height = 0.05),
               alpha = 0.5) +
    facet_wrap(~var, scales = "free") +
    mytheme()
```



6

If you see some strange values on predictor try to solve the problem, are they impossible or just extreme?

There are two impossible values for the `parent_score` because the range is only 0-10, we need to remove them:

```
dat[dat$parent_score > 10, ]
```

```
##      id age after_school parent_score teacher_score y        age0
## 99   99   9           no           20             6 1  0.04347826
## 100 100   7           no           25             6 1 -1.95652174
```

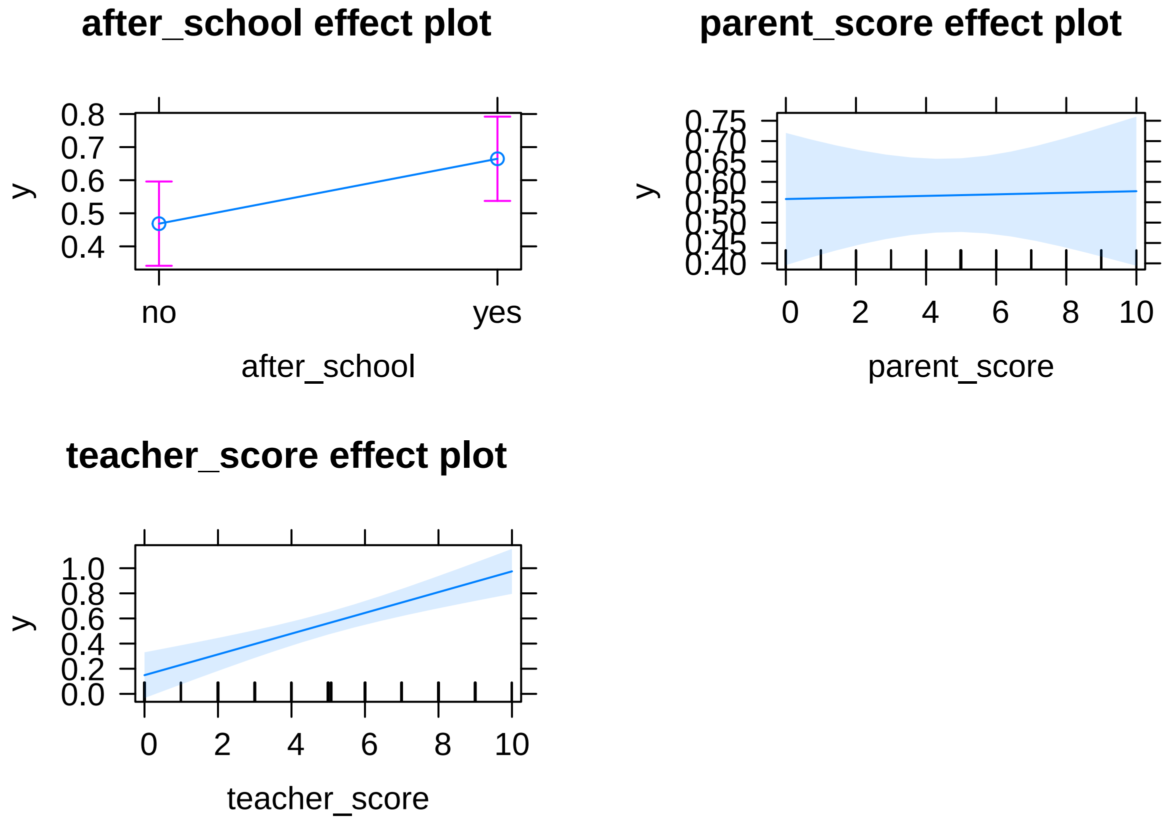
```
dat <- dat[dat$parent_score <= 10, ]
```

7

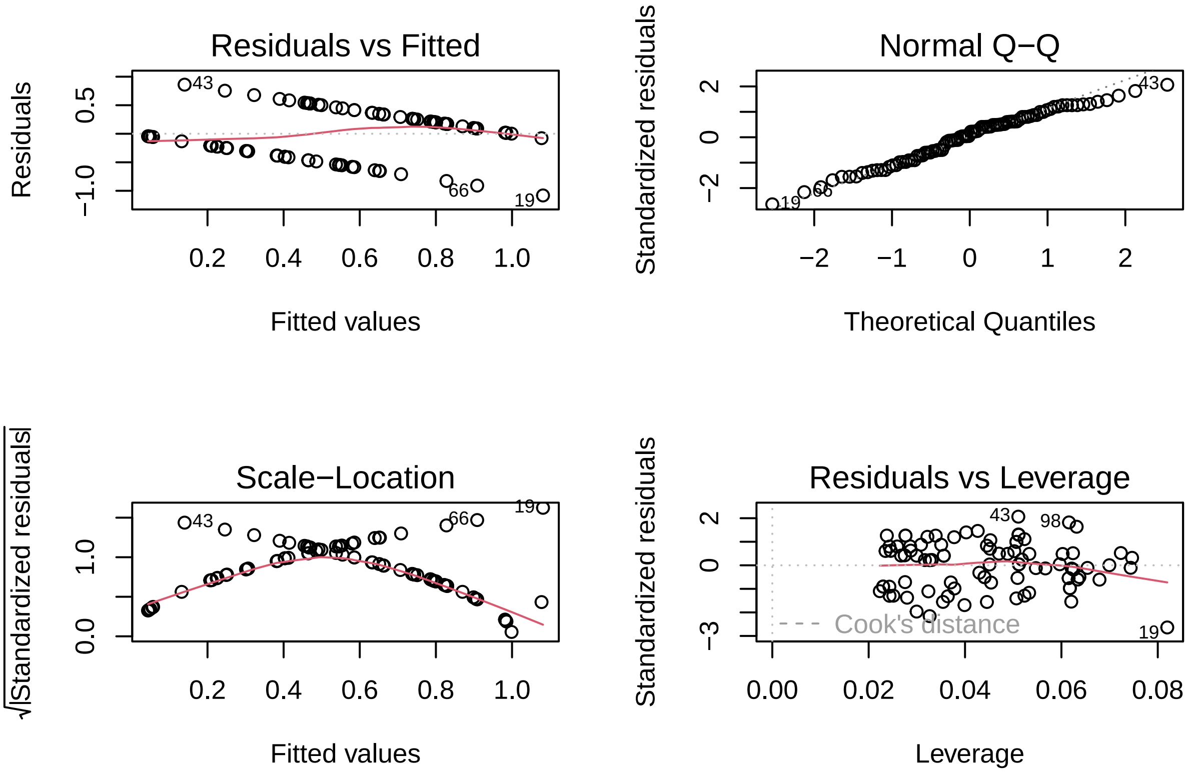
```
# gaussian model with all variables, no interactions
fit_lm <- lm(y ~ after_school + parent_score + teacher_score, data = dat)

# model result
summary(fit_lm)
```

```
##
## Call:
## lm(formula = y ~ after_school + parent_score + teacher_score,
##     data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.08137 -0.30649  0.05502  0.32583  0.86018
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.139915   0.117068   1.195   0.2353
## after_school1 -0.196110   0.091087  -2.153   0.0341 *
## parent_score   0.001918   0.014833   0.129   0.8974
## teacher_score  0.082614   0.015766   5.240 1.13e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4276 on 86 degrees of freedom
## Multiple R-squared:  0.2884, Adjusted R-squared:  0.2635
## F-statistic: 11.62 on 3 and 86 DF,  p-value: 1.826e-06
```

```
# plotting results
plot(effects::allEffects(fit_lm))
```



```
# plotting diagnostics
par(mfrow = c(2,2))
plot(fit_lm)
```


Residuals vs Fitted
Residuals
0.5
−1.0
0.2
0.4
0.6
0.8
1.0
Fitted values
43
66
19
Normal Q−Q
Standardized residuals
2
0
−2
−1
1
Theoretical Quantiles
43
66
19
Scale−Location
√|Standardized residuals|
1.0
0.0
Fitted values
43
66
19
Residuals vs Leverage
Standardized residuals
2
0
−3
0.00
0.02
0.04
0.06
0.08
Leverage
Cook's distance
43
98
19