Homework 8

Kevin Linares and Jamila Sani

12 November, 2024

```
library(jtools)
library(faraway)
library(olsrr)
```

```
##
## Attaching package: 'olsrr'
```

```
## The following object is masked from 'package:faraway':
##
##     hsb
```

```
## The following object is masked from 'package:datasets':
##
##     rivers
```

```
library(knitr)
library(sandwich)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
options(scipen=999)

#_________________ handy functions ___________________
kappa_fun <- function(mod_name){
  # model matrix w/o intercept
  x_mod = model.matrix(mod_name)[,-1]
  #  matrix multiplication
  e=eigen(t(x_mod)%*%x_mod)
  message("Inspect wide range in eigenvalues")
  print(e$val)
```

```
  # calculate kappa values
  message("Inspect if Kappa conditional number value > 30")
  kappa_mod = sqrt(max(e$val)/min(e$val))
  print(kappa_mod)
  message("Inspect condition index, of at least one linear combination")
  nu = sqrt(max(e$val)/e$val)
  print(nu)
}



f_stat_lack_fit_fun <- function(mod_orig, mod_lack_fit){

  lack_fit_ss = anova(mod_orig)[[2]][2]
  pure_error = anova(mod_lack_fit)[[2]][2]
  # df lack of fit = # of unique x values - number of parameters
  df_lack_fit = length(unique(mod_orig$model$speed)) -
    length(mod_cars$coefficients)
  # df pure error = # of obs - # of unique x values
  df_pure_error = length(mod_orig$model$speed) -
    length(unique(mod_orig$model$speed))

  f_stat = ((
    # difference in residuals between models
    (lack_fit_ss - pure_error) /
      # df lack of fit - number number parameters
      df_lack_fit ) /
      # divide pure error by # of unique x values
      (pure_error / df_pure_error)
  )

  # Calculate p-value for lack of fit test
  p_value = 1 - pf(f_stat, df1=df_lack_fit, df2=df_pure_error)

  message(str_c("The F statistic is ", f_stat, " and the p_value is ", p_value))
}
# ______________________ END ________________________
```

Note: For each question, include R code and output pertinent to your answers.

1. Faraway Chapter 7. Exercise 5.

```
data("prostate")
?prostate
glimpse(prostate)
```

```
## Rows: 97
## Columns: 9
## $ lcavol  <dbl> -0.5798185, -0.9942523, -0.5108256, -1.2039728, 0.7514161, -1.~
## $ lweight <dbl> 2.7695, 3.3196, 2.6912, 3.2828, 3.4324, 3.2288, 3.4735, 3.5395~
## $ age     <int> 50, 58, 74, 58, 62, 50, 64, 58, 47, 63, 65, 63, 63, 67, 57, 66~
## $ lbph    <dbl> -1.386294, -1.386294, -1.386294, -1.386294, -1.386294, -1.3862~
## $ svi     <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
```

```
## $ lcp     <dbl> -1.38629, -1.38629, -1.38629, -1.38629, -1.38629, -1.38629, -1~
## $ gleason <int> 6, 6, 7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 6, 7, 6, 6, 6, 6,~
## $ pgg45   <int> 0, 0, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 30, 5, 5, 0, 30, 0, 0, 0,~
## $ lpsa    <dbl> -0.43078, -0.16252, -0.16252, -0.16252, 0.37156, 0.76547, 0.76~
```

- For the prostate data, fit a model with lpsa as the response and the other variables as predictors.

```
summ(
  mod <- lm(lpsa ~ lcavol + age + lbph + svi + lcp + gleason + pgg45,
            prostate))
```

```
## MODEL INFO:
## Observations: 97
## Dependent Variable: lpsa
## Type: OLS linear regression
##
## MODEL FIT:
## F(7,89) = 21.35, p = 0.00
## R² = 0.63
## Adj. R² = 0.60
##
## Standard errors:OLS
## ------------------------------------------------
##                       Est.   S.E.   t val.      p
## ----------------- ------- ------ -------- ------
## (Intercept)           2.30   1.18     1.95   0.05
## lcavol                0.62   0.09     6.91   0.00
## age                  -0.01   0.01    -1.27   0.21
## lbph                  0.17   0.06     3.02   0.00
## svi                   0.81   0.25     3.23   0.00
## lcp                  -0.11   0.09    -1.12   0.26
## gleason              -0.01   0.16    -0.05   0.96
## pgg45                 0.00   0.00     0.96   0.34
## ------------------------------------------------
```

1.A. Compute and comment on Kappa and the condition numbers.

- We compute the conditional number kappa, which measures therelative sizes of the eigenvalues where l

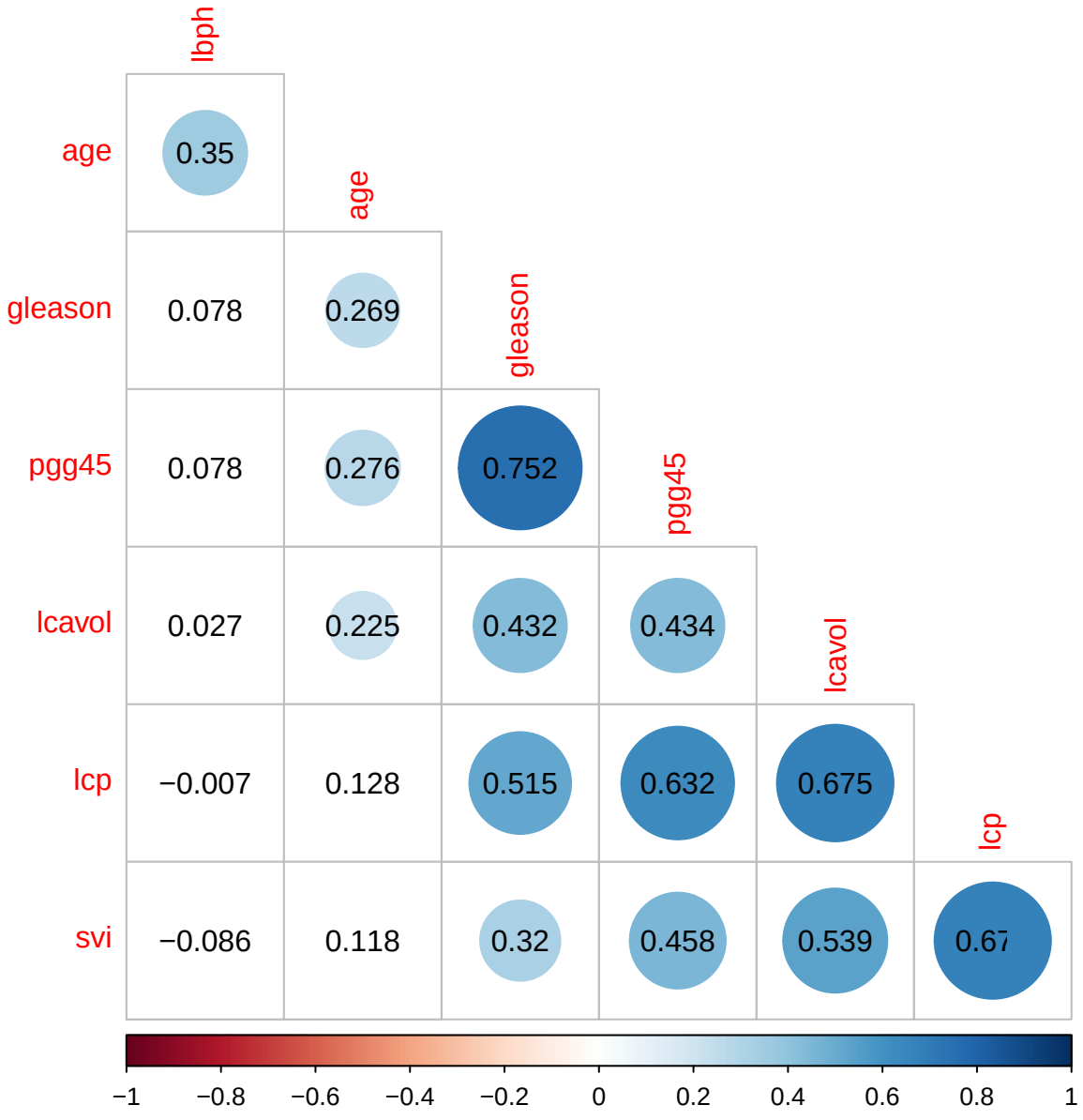
```
kappa_fun(mod)
```

```
## Inspect wide range in eigenvalues

## [1] 477834.78747  61864.84214    210.81849    174.87110     58.01914
## [6]     44.36002      8.29714

## Inspect if Kappa conditional number value > 30

## [1] 239.9798

## Inspect condition index, of at least one linear combination

## [1]   1.000000   2.779182  47.608504  52.273294  90.751383 103.787012 239.979789
```

1.B. Compute and comment on the correlations between the predictors. Round to 3 decimal places.

- In the correlation matrix contour plot below we take note on the strong positive association betwee

```
testRes = cor.mtest(prostate |> select(-lpsa, -lweight), conf.level = 0.95)

corrplot(cor(prostate |> select(-lpsa, -lweight)),
         p.mat = testRes$p, method = 'circle', type = 'lower', insig='blank',
         order = 'AOE', diag = FALSE)$corrPos -> p1
text(p1$x, p1$y, round(p1$corr, 3))
```



1.C. Compute the variance inflation factors. Comment on whether any appear problematic and why.

- The variance inflation factor (VIF) assesses multicollinearity, which is the ratio of the variance of $\hat{\beta}_j$: when fitting the full model divided by the variance of $\hat{\beta}_j$ if fit on its own. The smallest possible value is 1 indicating absence of collinearity. A rule of thumb of a VIF of over 5 is an indication of a problematic amount of collinearity. VIF is expressed as:

$$VIF_j = \frac{1}{1 = R_j^2}$$

- We see below that our VIF for each variable is below 5 and conclude that just we saw in our correlation plot above, we are not seeing evidence for multicollinearity.

```
ols_vif_tol(mod) |> select(Variables, VIF)
```

```
##   Variables      VIF
## 1    lcavol 2.010442
## 2       age 1.284325
## 3      lbph 1.165151
## 4       svi 1.946040
## 5       lcp 3.097950
## 6   gleason 2.433708
## 7     pgg45 2.974017
```

2. Faraway Chapter 8. Exercise 4.

- For the cars dataset, fit a linear model with distance as the response and speed as the predictor.

```
data("cars")
?cars
glimpse(cars)
```

```
## Rows: 50
## Columns: 2
## $ speed <dbl> 4, 4, 7, 7, 8, 9, 10, 10, 10, 11, 11, 12, 12, 12, 12, 13, 13, 13~
## $ dist  <dbl> 2, 10, 4, 22, 16, 10, 18, 26, 34, 17, 28, 14, 20, 24, 28, 26, 34~
```

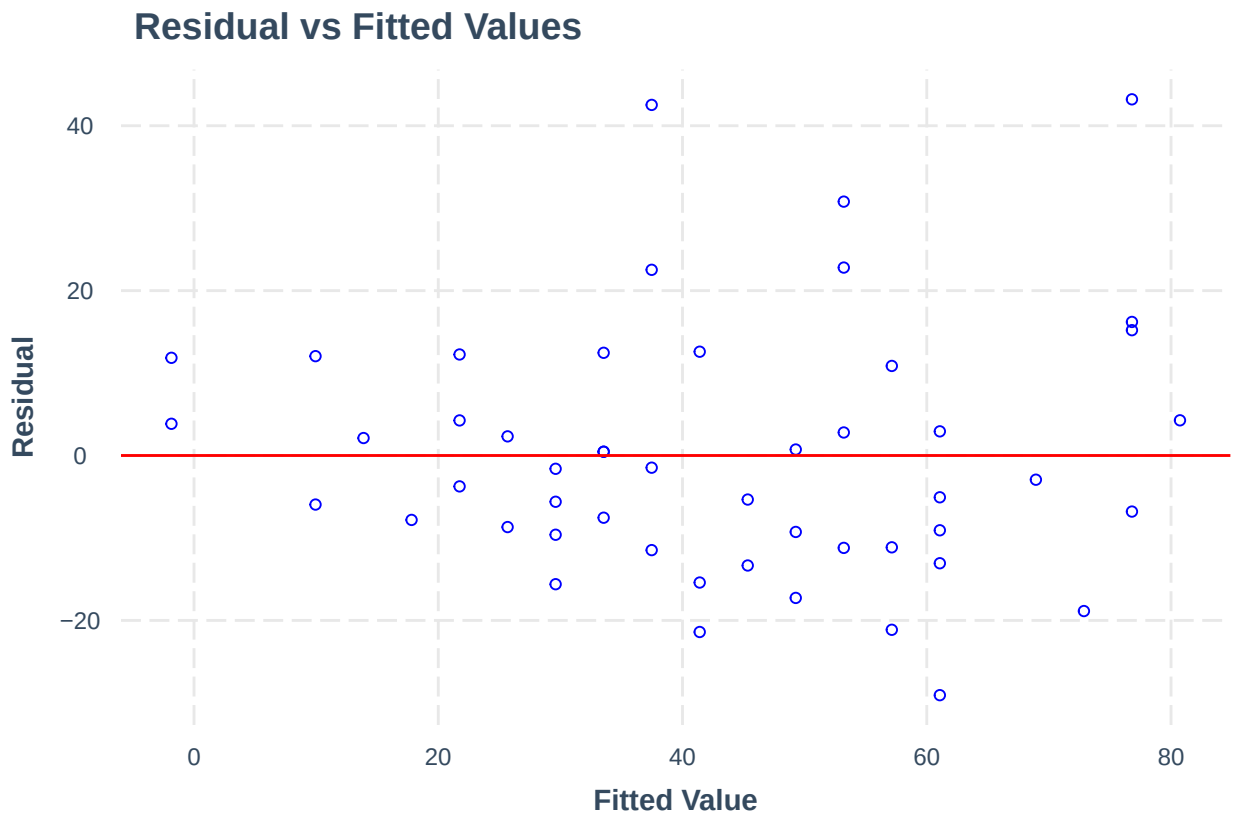
```
summ(mod_cars <- lm(dist ~ speed, cars))
```

```
## MODEL INFO:
## Observations: 50
## Dependent Variable: dist
## Type: OLS linear regression
##
## MODEL FIT:
## F(1,48) = 89.57, p = 0.00
## R² = 0.65
## Adj. R² = 0.64
##
## Standard errors:OLS
## ------------------------------------------------
##                       Est.   S.E.   t val.      p
## ----------------- -------- ------ -------- ------
## (Intercept)         -17.58   6.76    -2.60   0.01
## speed                 3.93   0.42     9.46   0.00
## ------------------------------------------------
```

Test the homoscedasticity assumption using both a scatter plot between the residuals and fitted values and an F-test of equal variance below and above the fitted value of 30. What do you conclude about whether the assumption is met?

- We can see from the residual plot below that as fitted values increase, variation in the residuals widens such as like a cone shape. We can also see that lower fitted values have residuals higher than 0, while some mid point values are below 0. Therefore, we are seeing evidence for heteroscadasticity.

```
ols_plot_resid_fit(mod_cars, print_plot = FALSE)  +
  theme_nice()
```



- We examine statistically equal variances for fitted values greater than 30 and values less than or equal to 30. The F-test for equal variances suggest that the variances in the residuals are not equal based on the confidence intervals (95%[1.53, 9.42]) does not contain 0, meaning that we reject the null that the true ration of variances =1 in favor of the alternative hypothesis of unequal variances, according to our F-test. We conclude that we have evidence for non-constant variance both graphically and statistically.

```
var.test(resid(mod_cars)[fitted(mod_cars)>30],
         resid(mod_cars)[fitted(mod_cars)<=30])
```

```
##
##  F test to compare two variances
##
## data:  resid(mod_cars)[fitted(mod_cars) > 30] and resid(mod_cars)[fitted(mod_cars) <= 30]
## F = 4.1325, num df = 34, denom df = 14, p-value = 0.006658
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  1.527594 9.415644
```

```
## sample estimates:
## ratio of variances
##           4.132502
```

2.B. Report the estimate of the heteroscedastic consistent variance for the regression slope.

- The heteroscedastic consistent variance for the slope is .183.

```
hetvar <- mod_cars |>
  # calculate Heteroscedasticity-consistent estimation of the covariance matrix for coefficients
  vcovHC()  |>
  # gives variances as they are the diagonal of the covariance matrix
  diag()

hetvar
```

```
## (Intercept)       speed
##  35.1862906   0.1827881
```

2.C. Construct 95% confidence interval of the regression slope assuming homoscedasticity and using the results in 2.B. How do they compare?

```
coef(mod_cars)[2]
```

```
##    speed
## 3.932409
```

```
confint(mod_cars)[2,]
```

```
##    2.5 %   97.5 %
## 3.096964 4.767853
```

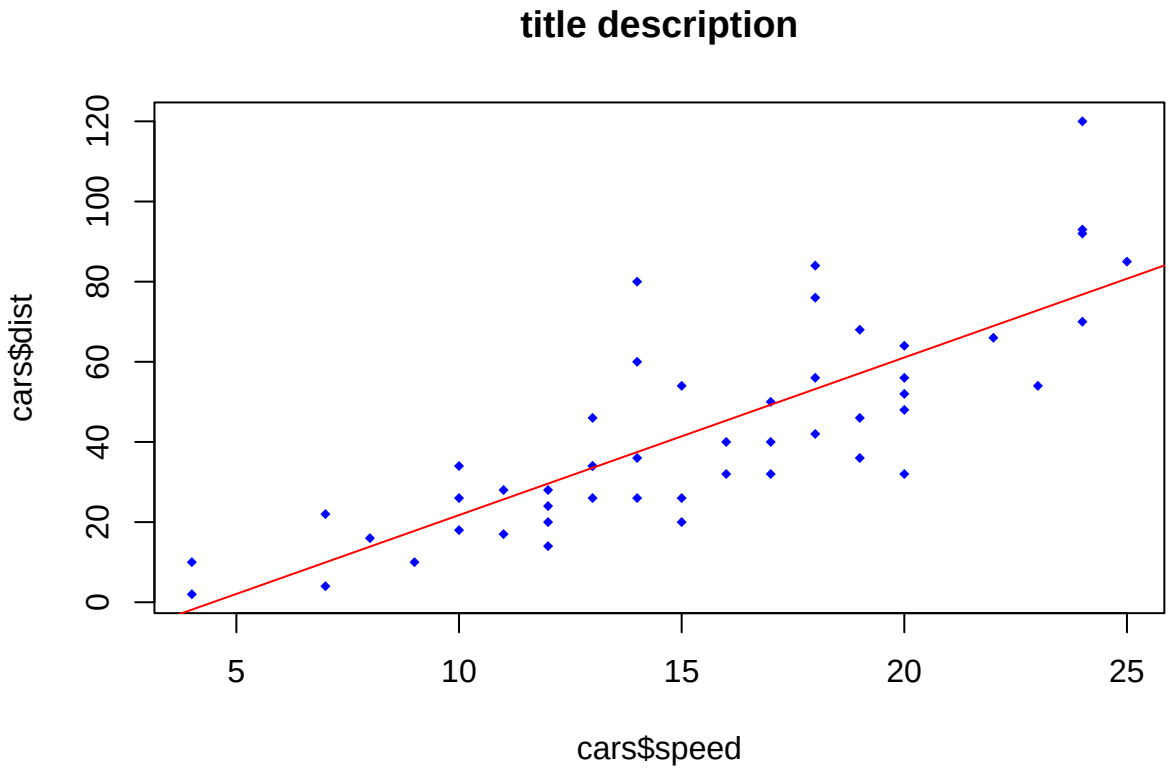
```
mod_cars$coefficients[2]+c(-1,1)*qt(0.975,mod_cars$df.residual)*sqrt(hetvar[2])
```

```
## [1] 3.072788 4.792030
```

2.D. Check for the lack of fit of the model.

- We use the function we created at the begining of this rmarkdown and pass it two models, the orignal model and one where we convert the continous variable into a factor.

```
plot(cars$speed,cars$dist,pch=18,col="blue",cex=.7,
     main="title description") +
  abline(lm(dist~speed, cars),col="red")
```




```
## integer(0)
```

```
summ(mod_cars_a<-lm(resid(mod_cars) ~ factor(speed), cars))
```

```
## MODEL INFO:
## Observations: 50
## Dependent Variable: resid(mod_cars)
## Type: OLS linear regression
##
## MODEL FIT:
## F(18,31) = 1.17, p = 0.34
## R² = 0.40
## Adj. R² = 0.06
##
## Standard errors:OLS
## ------------------------------------------------------
##                         Est.    S.E.   t val.      p
## -------------------- -------- ------- -------- ------
## (Intercept)              7.85   10.45     0.75   0.46
## factor(speed)7          -4.80   14.77    -0.32   0.75
## factor(speed)8          -5.73   18.09    -0.32   0.75
## factor(speed)9         -15.66   18.09    -0.87   0.39
## factor(speed)10         -3.59   13.49    -0.27   0.79
## factor(speed)11        -11.03   14.77    -0.75   0.46
## factor(speed)12        -15.96   12.79    -1.25   0.22
## factor(speed)13         -6.39   12.79    -0.50   0.62
## factor(speed)14          5.18   12.79     0.40   0.69
```

```
## factor(speed)15         -15.92   13.49    -1.18   0.25
## factor(speed)16         -17.19   14.77    -1.16   0.25
## factor(speed)17         -16.45   13.49    -1.22   0.23
## factor(speed)18           3.45   12.79     0.27   0.79
## factor(speed)19         -14.99   13.49    -1.11   0.27
## factor(speed)20         -18.52   12.36    -1.50   0.14
## factor(speed)22         -10.78   18.09    -0.60   0.56
## factor(speed)23         -26.72   18.09    -1.48   0.15
## factor(speed)24           9.10   12.79     0.71   0.48
## factor(speed)25          -3.58   18.09    -0.20   0.84
## ------------------------------------------------------
```

- There is little evidence for lack of fit in this data or model. We failed to reject the null hypothesis, p >.05, which might be a limitation of the data. We took note in the R^2 of these models and report that for mod_cars (original model) the R^2 is .65, while the R^2 for mod_cars_a (lack of fit model) it dropped to .40, suggesting that the data may not support these many levels in the factor considering that there are only 50 observations and 25 factor levels. In this case, we may not want to rely on R^2 as much.

```
f_stat_lack_fit_fun(mod_cars, mod_cars_a)
```

```
## The F statistic is 1.23694991825985 and the p_value is 0.294837396797043
```

- We can also get the f-statistic from the `olsrr::ols_pure_error()`

```
## use the olsrr package to conduct a lack of fit test
ols_pure_error_anova(mod_cars)
```

```
## Lack of Fit F Test
## ------------------
## Response :   dist
## Predictor:   speed
##
##                          Analysis of Variance Table
## ---------------------------------------------------------------------------
##                 DF     Sum Sq     Mean Sq     F Value            Pr(>F)
## ---------------------------------------------------------------------------
## speed            1    21185.46    21185.46    97.08356    0.0000000000004102508
## Residual        48    11353.52    236.5317
##  Lack of fit    17    4588.738    269.9257     1.23695               0.2948374
##  Pure Error     31    6764.783    218.2188
## ---------------------------------------------------------------------------
```