Multiple and Logistic Regression in R_Evaluating and extending parallel slopes model

dizhen

5/3/2020

Model fit, residuals, and prediction

- $R^2 = 1 - \frac{SSE}{SST}$, SSE get smaller -> R^2 increases
- As p(number of explanatory variables) increases, the R^2 is always getting larger. Solution: $R^2_{adj} = 1 - \frac{SSE}{SST} \frac{n-1}{n-p-1}$

Fitted values

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(broom)
load("data/mario_kart.RData")

mario_kart <- mario_kart %>% filter(totalPr<=100)

# fit parallel slopes
mod <- lm(totalPr ~ wheels + cond, data = mario_kart)
summary(mod)
```

```
##
## Call:
## lm(formula = totalPr ~ wheels + cond, data = mario_kart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0078  -3.0754  -0.8254   2.9822  14.1646
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.3698     1.0651  39.780  < 2e-16 ***
## wheels        7.2328     0.5419  13.347  < 2e-16 ***
## condused     -5.5848     0.9245  -6.041 1.35e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.887 on 138 degrees of freedom
## Multiple R-squared:  0.7165, Adjusted R-squared:  0.7124
## F-statistic: 174.4 on 2 and 138 DF,  p-value: < 2.2e-16
```

```
# returns a vector
predict(mod)
```

```
##        1        2        3        4        5        6        7        8
## 49.60260 44.01777 49.60260 49.60260 56.83544 42.36976 36.78493 56.83544
##        9       10       11       12       13       14       15       16
## 44.01777 44.01777 56.83544 56.83544 56.83544 56.83544 44.01777 36.78493
##       17       18       19       20       21       22       23       24
## 49.60260 49.60260 56.83544 36.78493 56.83544 56.83544 56.83544 44.01777
##       25       26       27       28       29       30       31       32
## 56.83544 36.78493 36.78493 36.78493 49.60260 36.78493 36.78493 44.01777
##       33       34       35       36       37       38       39       40
## 51.25061 44.01777 44.01777 36.78493 44.01777 56.83544 56.83544 49.60260
##       41       42       43       44       45       46       47       48
## 44.01777 51.25061 56.83544 56.83544 44.01777 56.83544 36.78493 36.78493
##       49       50       51       52       53       54       55       56
## 44.01777 56.83544 36.78493 44.01777 42.36976 36.78493 36.78493 44.01777
##       57       58       59       60       61       62       63       64
## 44.01777 36.78493 36.78493 56.83544 36.78493 56.83544 36.78493 51.25061
##       65       66       67       68       69       70       71       72
## 56.83544 44.01777 58.48345 51.25061 49.60260 44.01777 49.60260 56.83544
##       73       74       75       76       77       78       79       80
## 56.83544 51.25061 44.01777 36.78493 36.78493 36.78493 44.01777 56.83544
##       81       82       83       84       85       86       87       88
## 44.01777 65.71629 44.01777 56.83544 36.78493 49.60260 49.60260 36.78493
##       89       90       91       92       93       94       95       96
## 44.01777 36.78493 51.25061 44.01777 36.78493 51.25061 42.36976 56.83544
##       97       98       99      100      101      102      103      104
## 51.25061 44.01777 51.25061 56.83544 56.83544 56.83544 36.78493 49.60260
##      105      106      107      108      109      110      111      112
## 51.25061 44.01777 56.83544 49.60260 36.78493 44.01777 51.25061 56.83544
##      113      114      115      116      117      118      119      120
## 64.06828 44.01777 49.60260 44.01777 49.60260 51.25061 42.36976 44.01777
##      121      122      123      124      125      126      127      128
## 56.83544 44.01777 49.60260 44.01777 51.25061 56.83544 56.83544 49.60260
##      129      130      131      132      133      134      135      136
## 56.83544 36.78493 44.01777 44.01777 36.78493 56.83544 36.78493 44.01777
##      137      138      139      140      141
## 36.78493 51.25061 49.60260 36.78493 56.83544
```

```r
# returns a data.frame
augment(mod)
```

```
## # A tibble: 141 x 10
##    totalPr wheels cond  .fitted .se.fit .resid   .hat .sigma  .cooksd .std.resid
##      <dbl>  <int> <chr>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>    <dbl>      <dbl>
##  1    51.6      1 new      49.6   0.709  1.95  0.0210   4.90  1.16e-3     0.403
##  2    37.0      1 used     44.0   0.547 -6.98  0.0125   4.87  8.71e-3    -1.44
##  3    45.5      1 new      49.6   0.709 -4.10  0.0210   4.89  5.15e-3    -0.848
##  4    44        1 new      49.6   0.709 -5.60  0.0210   4.88  9.61e-3    -1.16
##  5    71        2 new      56.8   0.676 14.2   0.0192   4.75  5.57e-2     2.93
##  6    45        0 new      42.4   1.07   2.63  0.0475   4.90  5.05e-3     0.551
##  7    37.0      0 used     36.8   0.707  0.235 0.0209   4.91  1.68e-5     0.0486
##  8    54.0      2 new      56.8   0.676 -2.85  0.0192   4.90  2.25e-3    -0.588
##  9    47        1 used     44.0   0.547  2.98  0.0125   4.90  1.59e-3     0.614
## 10    50        1 used     44.0   0.547  5.98  0.0125   4.88  6.40e-3     1.23
## # ... with 131 more rows
```

Predictions

```r
new_obs <- data.frame(wheels = 1, cond = "used")
# returns a vector
predict(mod, newdata = new_obs)
```

```
##        1
## 44.01777
```

```r
# returns a data.frame
augment(mod, newdata = new_obs)
```

```
## # A tibble: 1 x 4
##   wheels cond  .fitted .se.fit
##    <dbl> <fct>   <dbl>   <dbl>
## 1      1 used     44.0   0.547
```

```r
# R^2 and adjusted R^2
summary(mod)
```

```
##
## Call:
## lm(formula = totalPr ~ wheels + cond, data = mario_kart)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.0078  -3.0754  -0.8254   2.9822  14.1646
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.3698     1.0651  39.780  < 2e-16 ***
## wheels        7.2328     0.5419  13.347  < 2e-16 ***
## condused     -5.5848     0.9245  -6.041 1.35e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.887 on 138 degrees of freedom
## Multiple R-squared:  0.7165, Adjusted R-squared:  0.7124
## F-statistic: 174.4 on 2 and 138 DF,  p-value: < 2.2e-16
```

```
# add random noise
mario_kart_noisy <- mutate(mario_kart, noise = rnorm(dim(mario_kart)[1],0,1))


# compute new model
mod2 <- lm(totalPr ~ wheels + cond + noise, data = mario_kart_noisy)

# new R^2 and adjusted R^2
summary(mod2)
```

```
##
## Call:
## lm(formula = totalPr ~ wheels + cond + noise, data = mario_kart_noisy)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.359  -3.208  -0.934   2.949  13.331
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.5928     1.0410  40.917  < 2e-16 ***
## wheels        7.1248     0.5295  13.457  < 2e-16 ***
## condused     -5.6461     0.9013  -6.264 4.52e-09 ***
## noise        -1.1852     0.4118  -2.878  0.00464 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.763 on 137 degrees of freedom
## Multiple R-squared:  0.7327, Adjusted R-squared:  0.7268
## F-statistic: 125.2 on 3 and 137 DF,  p-value: < 2.2e-16
```

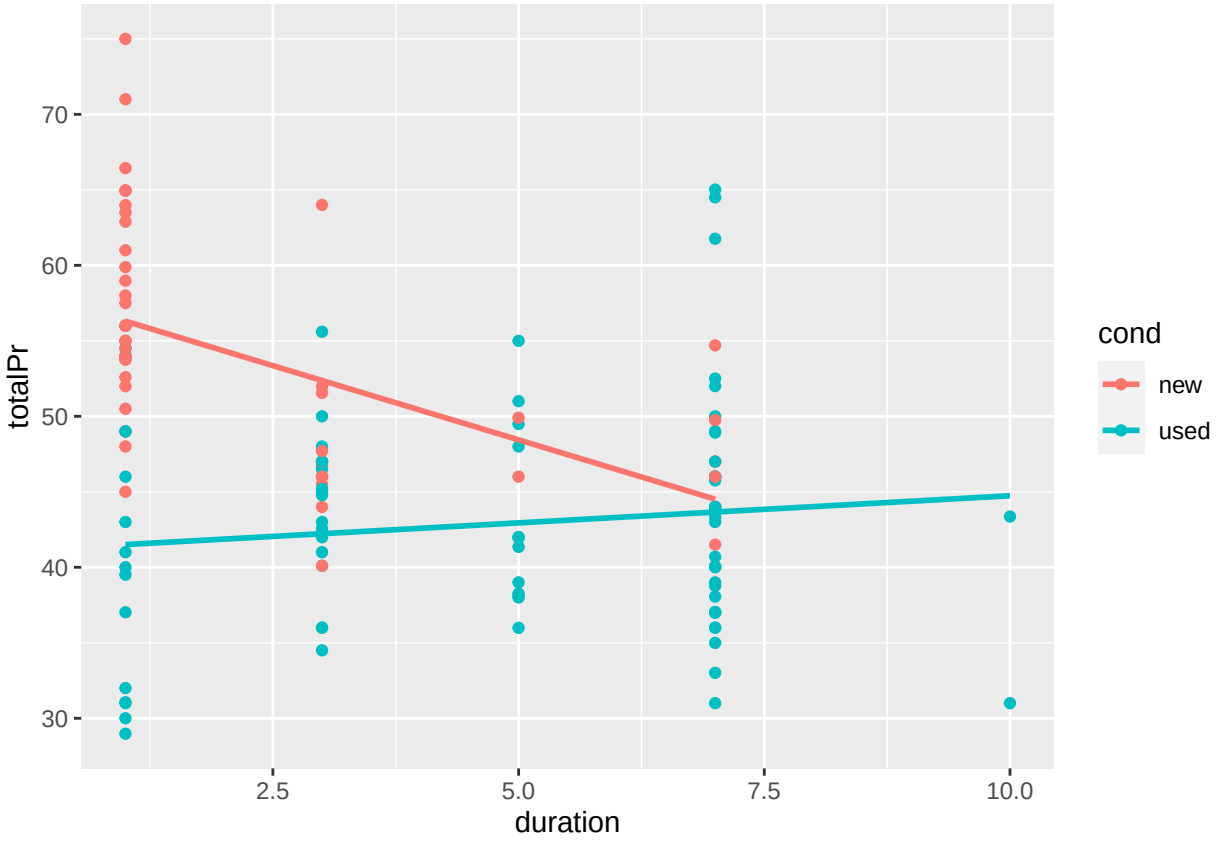
Understanding interaction

```
# include interaction
lm(totalPr ~ cond + duration + cond:duration, data = mario_kart)
```

```
##
## Call:
## lm(formula = totalPr ~ cond + duration + cond:duration, data = mario_kart)
##
## Coefficients:
##       (Intercept)           condused           duration  condused:duration
##            58.268            -17.122             -1.966              2.325
```

```
library(ggplot2)
# interaction plot
ggplot(mario_kart, aes(y = totalPr, x = duration, color = cond)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)
```

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



Simpson's Paradox

```
slr <- ggplot(mario_kart, aes(y = totalPr, x = duration)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE)

# model with one slope
lm(totalPr ~ duration, data=mario_kart)
```

```
##
## Call:
## lm(formula = totalPr ~ duration, data = mario_kart)
##
## Coefficients:
```

```
## (Intercept)     duration
##      52.374       -1.317
```

```
# plot with two slopes
slr + aes(color = cond)
```

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