forchapter5

Haziq Jamil

18/03/2020

```
## Parsed with column specification:
## cols(
##   state = col_character(),
##   poverty = col_double(),
##   hs_grad = col_double(),
##   home_own = col_double(),
##   median_income = col_double(),
##   party_maj = col_character()
## )
```

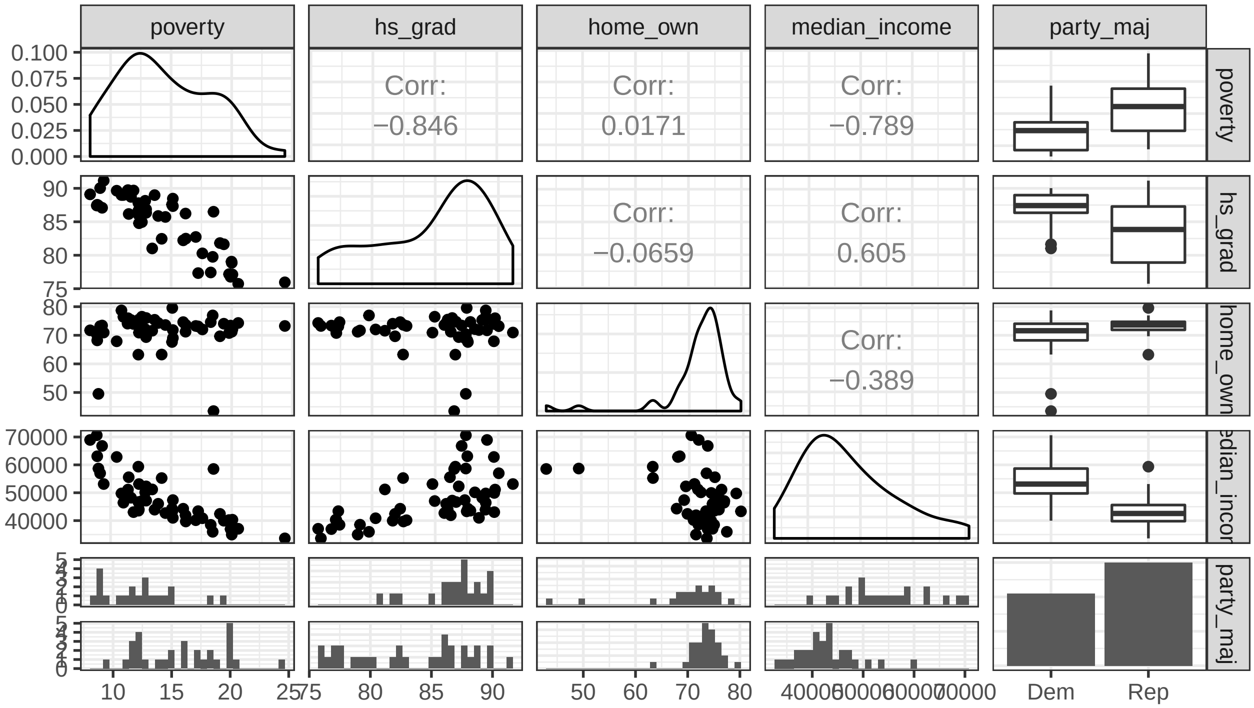
```
poverty
```

```
## # A tibble: 51 x 6
##    state                poverty hs_grad home_own median_income party_maj
##    <chr>                  <dbl>   <dbl>    <dbl>         <dbl> <chr>
##  1 Alabama                 19.9    76.8     73.4        36963. Rep
##  2 Alaska                  12.3    86.6     63.2        59351. Rep
##  3 Arizona                 19.0    81.8     69.7        42418. Rep
##  4 Arkansas                20.0    78.9     71.2        34983. Rep
##  5 California              14.2    82.5     63.3        55266. Dem
##  6 Colorado                12.9    88.1     72.1        50136. Dem
##  7 Connecticut              8.31   89.1     71.7        68935. Dem
##  8 Delaware                11.5    86.2     74.7        55568. Dem
##  9 District of Columbia    18.5    86.5     43.5        58526  Dem
## 10 Florida                 16.0    82.2     74.6        44269. Rep
## # ... with 41 more rows
```

```
GGally::ggpairs(poverty[, -1]) + theme_bw()
```

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

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
mod <- lm(formula = poverty ~ hs_grad + home_own + I(median_income/1000) +
            party_maj, data = poverty)
summary(mod)
```

```
##
## Call:
## lm(formula = poverty ~ hs_grad + home_own + I(median_income/1000) +
##     party_maj, data = poverty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.8794 -0.8104 -0.0762  0.8412  3.2827
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           76.07336    4.58210  16.602  < 2e-16 ***
## hs_grad               -0.45345    0.05599  -8.098 2.12e-10 ***
## home_own              -0.14614    0.03538  -4.130 0.000151 ***
## I(median_income/1000) -0.26101    0.03427  -7.616 1.10e-09 ***
## party_majRep          -0.51100    0.51347  -0.995 0.324857
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.354 on 46 degrees of freedom
## Multiple R-squared:  0.8859, Adjusted R-squared:  0.876
## F-statistic: 89.28 on 4 and 46 DF,  p-value: < 2.2e-16
```

```
X <- model.matrix(mod); head(X)   # n by (p + 1) matrix
```

```
##   (Intercept)  hs_grad home_own I(median_income/1000) party_majRep
## 1           1 76.78209 73.38657              36.96294            1
## 2           1 86.59310 63.22759              59.35103            1
## 3           1 81.82000 69.65333               42.41793            1
## 4           1 78.86400 71.23067               34.98271            1
## 5           1 82.45862 63.26724               55.26572            0
## 6           1 88.13906 72.14531               50.13584            0
```

```
y <- poverty$poverty; head(y)  # n by 1 vector
```

```
## [1] 19.90746 12.29310 19.03333 20.00533 14.22069 12.86094
```

```
XtX <- t(X) %*% X  # (p + 1) by (p + 1) matrix
colnames(XtX) <- rownames(XtX) <- NULL
XtX
```

```
##          [,1]       [,2]       [,3]       [,4]     [,5]
## [1,]   51.000   4323.498   3657.123   2430.910   30.000
## [2,] 4323.498 367497.623 309942.167 207273.903 2498.765
## [3,] 3657.123 309942.167 264098.108 173256.517 2202.333
## [4,] 2430.910 207273.903 173256.517 119871.817 1284.466
## [5,]   30.000   2498.765   2202.333   1284.466   30.000
```

```
Xty <- t(X) %*% y; head(Xty)  # (p + 1) by 1 vector
```

```
##                             [,1]
## (Intercept)             734.9980
## hs_grad               61590.5786
## home_own              52725.4769
## I(median_income/1000) 33676.2834
## party_majRep            476.7111
```

```
as.numeric(beta.hat <- solve(XtX, Xty))  # regression coefficients
```

```
## [1] 76.0733587 -0.4534462 -0.1461374 -0.2610123 -0.5109967
```

```
y.hat <- as.numeric(X %*% beta.hat); head(y.hat)  # fitted values
```

```
## [1] 20.37351 11.56579 17.21084 20.26140 15.01207 12.47784
```

```
eps.hat <- y - y.hat; head(eps.hat)  # residuals
```

```
## [1] -0.4660507  0.7273168  1.8224961 -0.2560688 -0.7913801  0.3830990
```

```
(sigma.hat <- sqrt(sum(eps.hat ^ 2) / (51 - 4 - 1)))  # residual SE
```

```
## [1] 1.353753
```

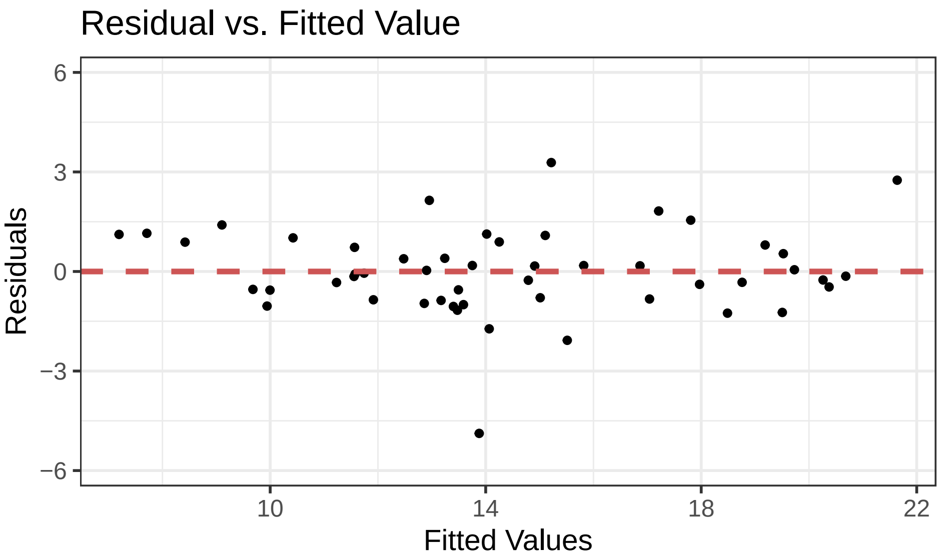
```
fitted(mod)  # obtain fitted values
```

```
##         1         2         3         4         5         6         7         8 
## 20.373513 11.565787 17.210837 20.261402 15.012070 12.477839  7.194859 11.576235 
##         9        10        11        12        13        14        15        16 
## 15.217284 15.819316 19.729882 13.879387 14.791221 13.493640 13.587406 11.915343 
##        17        18        19        20        21        22        23        24 
## 13.474400 20.683055 19.524389 13.240073  8.420077  9.104415 12.953813 11.231420 
##        25        26        27        28        29        30        31        32 
## 21.637436 17.040754 14.251766 12.859763 13.400985  9.679679  7.710733 17.805235 
##        33        34        35        36        37        38        39        40 
## 12.902249 17.969372 14.065428 13.752141 16.863705 14.909914 13.171882  9.942183 
##        41        42        43        44        45        46        47        48 
## 19.186426 15.105553 19.505661 18.487914 10.424302 11.556045 15.513875 14.018749 
##        49        50        51 
## 18.759122 11.743215  9.996266
```

```
resid(mod)  # obtain residuals
```

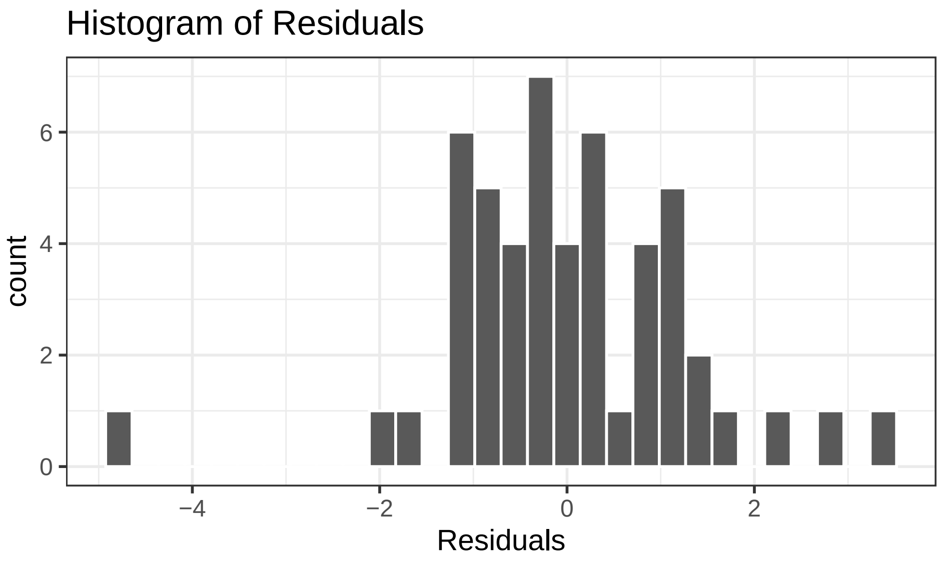
```
##           1           2           3           4           5           6 
## -0.46605065  0.72731679  1.82249609 -0.25606877 -0.79138008  0.38309895 
##           7           8           9          10          11          12 
##  1.11764146 -0.07623466  3.28271639  0.17919156  0.05250815 -4.87938690 
##          13          14          15          16          17          18 
## -0.26394784 -0.55344392 -0.99936285 -0.85271642 -1.16773370 -0.14222118 
##          19          20          21          22          23          24 
##  0.53654811  0.39742676  0.88408973  1.40272776  2.14377701 -0.33141953 
##          25          26          27          28          29          30 
##  2.75158873 -0.82944933  0.89109102 -0.96191335 -1.05392611 -0.53967877 
##          31          32          33          34          35          36 
##  1.15117164  1.54627966  0.03323485 -0.38737151 -1.72769199  0.18081322 
##          37          38          39          40          41          42 
##  0.17136036  0.16508568 -0.87188216 -1.04218336  0.79835675  1.08687105 
##          43          44          45          46          47          48 
## -1.23408238 -1.25563045  1.01362932 -0.14890176 -2.07432310  1.12740532 
##          49          50          51 
## -0.32639451 -0.05154804 -0.56148303
```

```
diag.plots <- lindia::gg_diagnose(mod, theme = theme_bw(), plot.all = FALSE,
                                  scale.factor = 1)
diag.plots$res_fitted
```

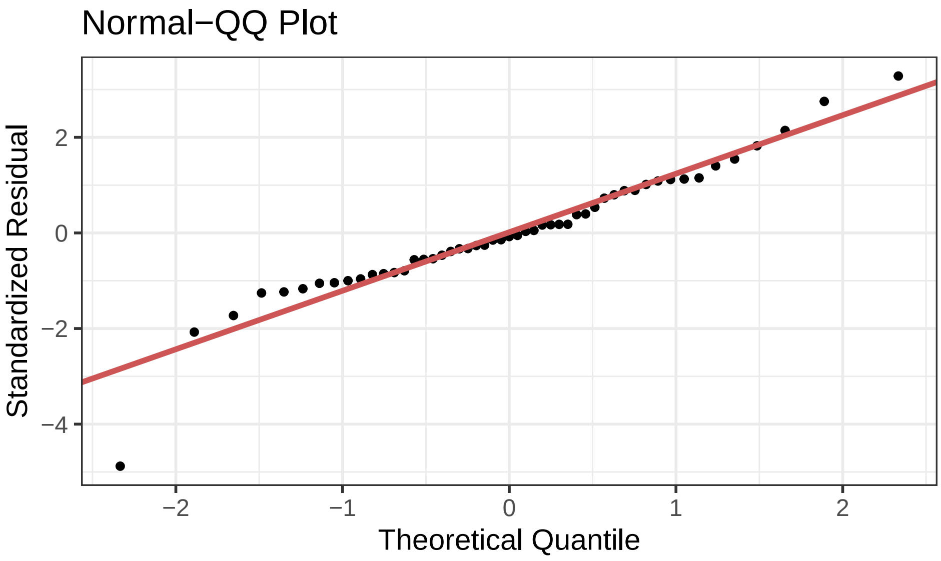


```
diag.plots$residual_hist
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
diag.plots$qqplot
```



```
(var.beta.hat <- sigma.hat ^ 2 * solve(XtX))   # Estimate of Var(beta.hat)
```

```
##               [,1]          [,2]          [,3]          [,4]         [,5]
## [1,] 20.995620883 -0.1849707465 -0.0674556619 -0.0029825145 -0.509316515
## [2,] -0.184970747  0.0031353530 -0.0004649709 -0.0010121404  0.001289793
## [3,] -0.067455662 -0.0004649709  0.0012520320  0.0003869122 -0.002294815
## [4,] -0.002982514 -0.0010121404  0.0003869122  0.0011746260  0.008589991
## [5,] -0.509316515  0.0012897933 -0.0022948145  0.0085899906  0.263655017
```

```
(se.beta.hat <- sqrt(diag(var.beta.hat)))  # SE beta.hat
```

```
## [1] 4.58209787 0.05599422 0.03538406 0.03427282 0.51347348
```

```
as.numeric(beta.hat / se.beta.hat)  # test statistic value
```

```
## [1] 16.6022990 -8.0980881 -4.1300351 -7.6157228 -0.9951764
```

```
as.numeric(  # p-values
  pt(abs(beta.hat / se.beta.hat), df = 50, lower.tail = FALSE)
```

```r
)
```

```
## [1] 2.753456e-22 5.867138e-11 6.883705e-05 3.281345e-10 1.622212e-01
```

```r
(total.SS <- sum((y - mean(y)) ^ 2))  # Total SS
```

```
## [1] 738.7815
```

```r
(resid.SS <- sum(eps.hat ^ 2))  # Resid SS
```

```
## [1] 84.30172
```

```r
(reg.SS <- total.SS - resid.SS)  # Reg SS
```

```
## [1] 654.4798
```

```r
(reg.SS / 4) / (resid.SS / (51 - 4 - 1))  # F-statistic
```

```
## [1] 89.28071
```

```r
1 - resid.SS / total.SS  # R^2 value
```

```
## [1] 0.8858909
```

```r
(tmp <- apply(X, 2, mean))
```

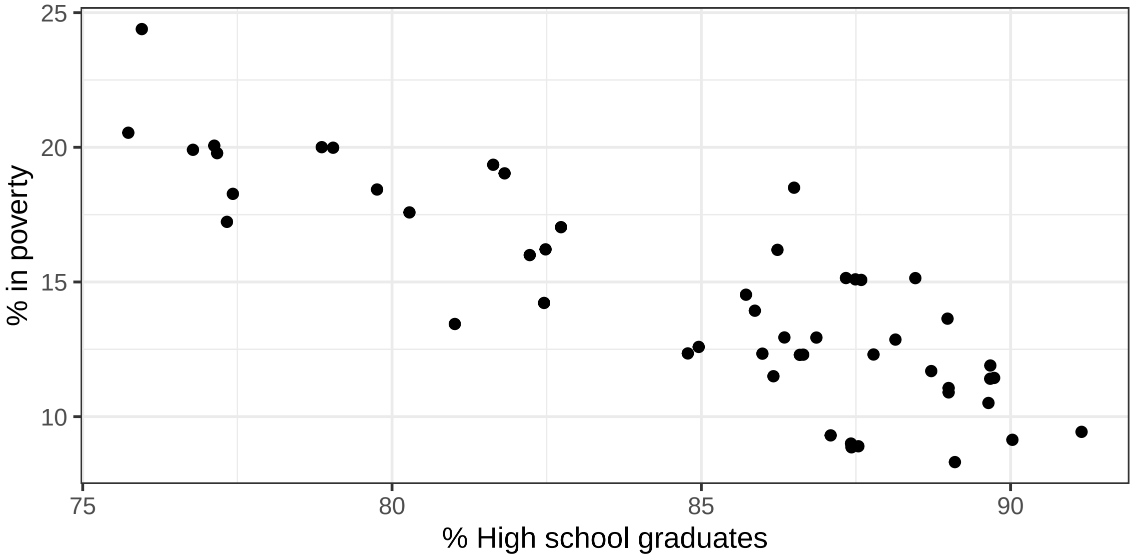
```
##           (Intercept)               hs_grad              home_own 
##             1.0000000            84.7744689            71.7082973 
## I(median_income/1000)          party_majRep 
##            47.6648981             0.5882353
```

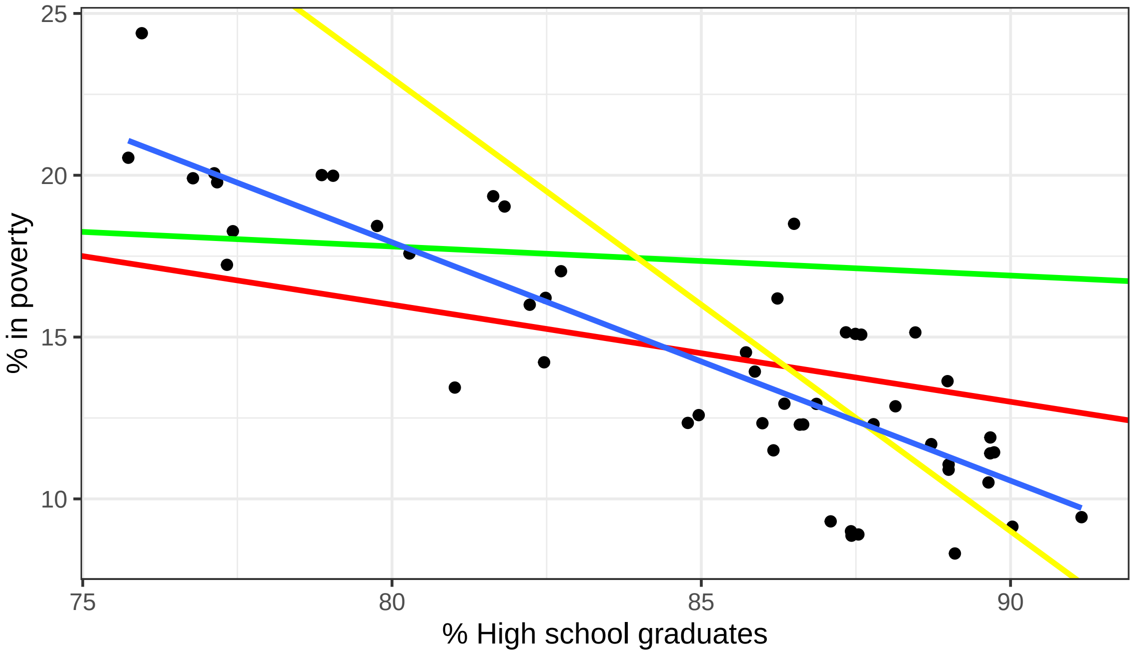
```r
newx <- data.frame(
  hs_grad = tmp[2],
  home_own = tmp[3],
  median_income = tmp[4] * 1000,
  party_maj = "Dem"
)
predict(mod, newx, interval = "confidence", level = 0.95)  # narrow
```

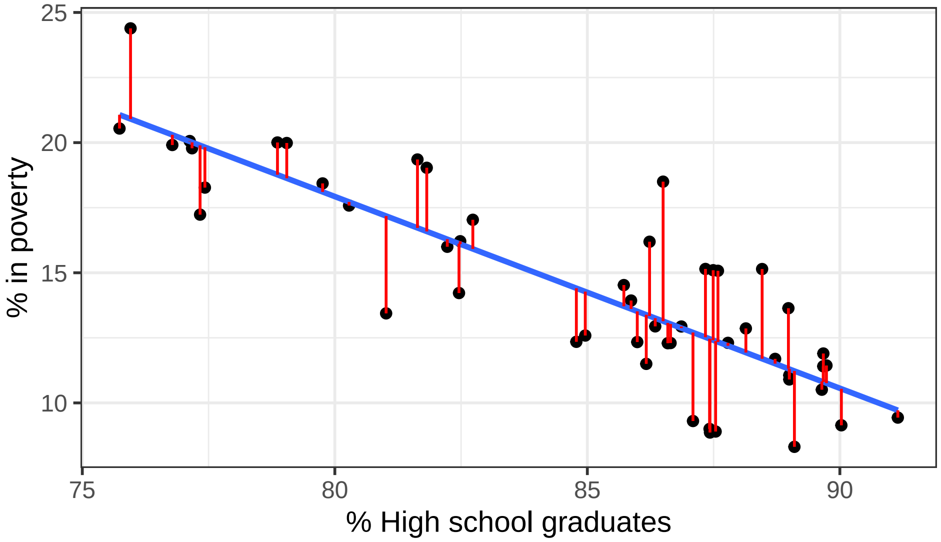
```
##              fit      lwr      upr
## hs_grad 14.71231 13.99451 15.43011
```

```r
predict(mod, newx, interval = "prediction", level = 0.95)  # wider
```

```
##              fit      lwr      upr
## hs_grad 14.71231 11.89439 17.53023
```


25
20
15
10
% in poverty
75
80
85
90
% High school graduates


25
20
15
10
% in poverty
75
80
85
90
% High school graduates


25
20
15
10
% in poverty
75
80
85
90
% High school graduates