Least Squares Regression

Alan T. Arnholt

Last modified on August 15, 2023 11:01:37 Eastern Daylight Time

Correlation

The **correlation coefficient**, denoted by r, measures the direction and strength of the linear relationship between two numerical variables. Is is given by the equation

$$r = \frac{1}{(n-1)} \sum_{i=1}^{n} \left(\frac{x_i - \bar{x}}{s_x}\right) \left(\frac{y_i - \bar{y}}{s_y}\right) \tag{1}$$

Following are the high school GPAs and the college GPAs at the end of the freshman year for ten different students from the `Gpa` data set of the `BSDA` package.

hsgpa	collgpa
2.7	2.2
3.1	2.8
2.1	2.4
3.2	3.8
2.4	1.9
3.4	3.5
2.6	3.1
2.0	1.4
3.1	3.4
2.5	2.5

Create a scatterplot and then comment on the relationship between the two variables.

R Code

```r
library(tidyverse)
library(BSDA)
ggplot(data = Gpa, aes(x = hsgpa, y = collgpa)) +
  labs(x = "High School GPA", y = "College GPA") +
  geom_point() +
  theme_bw()
```

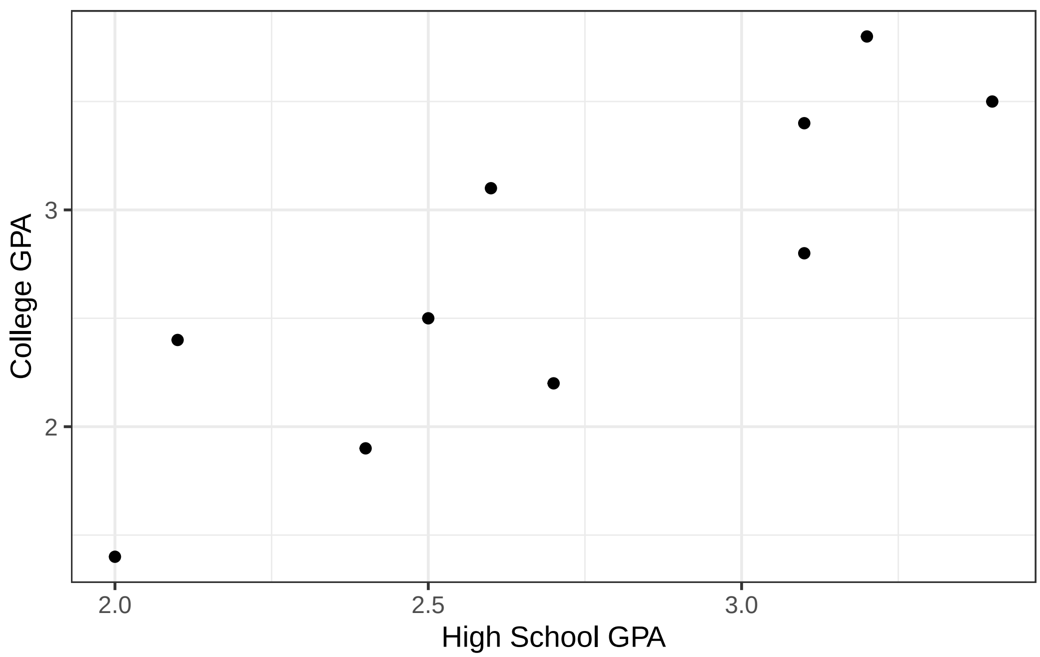


Figure 1: Scatterplot of College GPA versus High School GPA

The college GPA is the response variable and is labeled on the vertical axis. The scatterplot in Figure 1 shows that the college GPA increases as the high school GPA increases. In fact, the the dots appear to cluster along a straight line. The correlation coefficient is $r = 0.844$, which indicates that a straight line is a reasonable relationship between the two variables.

- Compute the correlation coefficient using the equation presented earlier.

R Code

```r
head(Gpa)
```

```
# A tibble: 6 x 2
  hsgpa collgpa
  <dbl>   <dbl>
1   2.7     2.2
2   3.1     2.8
3   2.1     2.4
4   3.2     3.8
5   2.4     1.9
6   3.4     3.5
```

```r
values <- Gpa %>%
  mutate(y_ybar = collgpa - mean(collgpa),
         x_xbar = hsgpa - mean(hsgpa),
         zx = x_xbar/sd(hsgpa),
         zy = y_ybar/sd(collgpa))
knitr::kable(values)
```

hsgpa	collgpa	y_ybar	x_xbar	zx	zy
2.7	2.2	-0.5	-0.01	-0.0209580	-0.6565322
3.1	2.8	0.1	0.39	0.8173628	0.1313064
2.1	2.4	-0.3	-0.61	-1.2784393	-0.3939193
3.2	3.8	1.1	0.49	1.0269430	1.4443708
2.4	1.9	-0.8	-0.31	-0.6496987	-1.0504515
3.4	3.5	0.8	0.69	1.4461035	1.0504515
2.6	3.1	0.4	-0.11	-0.2305382	0.5252257
2.0	1.4	-1.3	-0.71	-1.4880195	-1.7069836
3.1	3.4	0.7	0.39	0.8173628	0.9191450
2.5	2.5	-0.2	-0.21	-0.4401184	-0.2626129

```r
#
values %>%
  summarize(r = (1/9)*sum(zx*zy))
```

```
# A tibble: 1 x 1
      r
  <dbl>
1 0.844
```

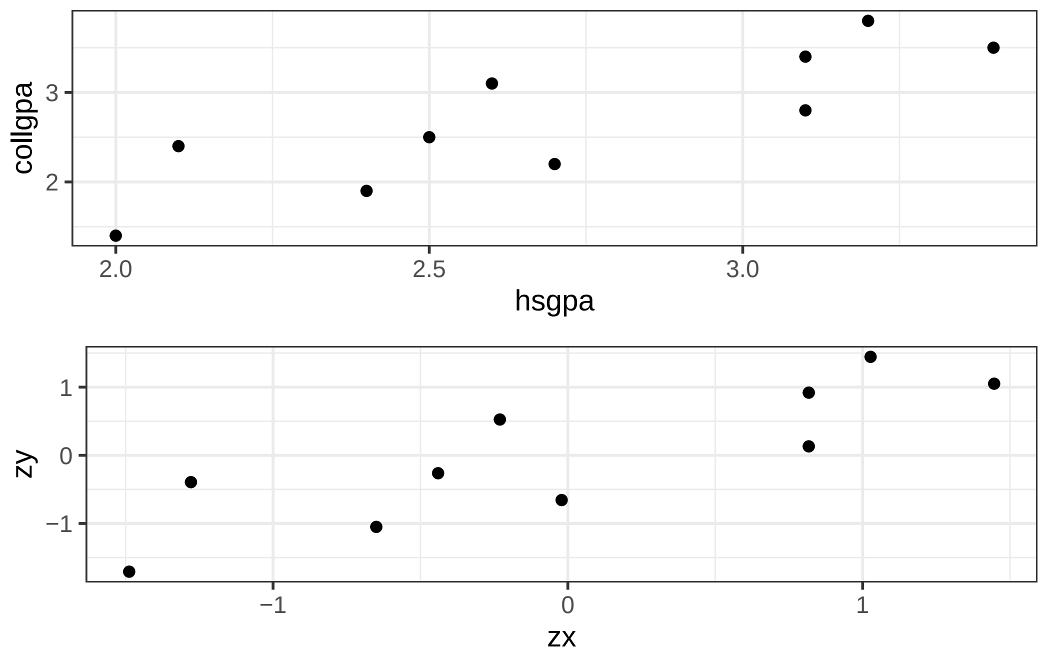
Using the build in `cor()` function:

```r
Gpa %>%
  summarize(r = cor(collgpa, hsgpa))
```

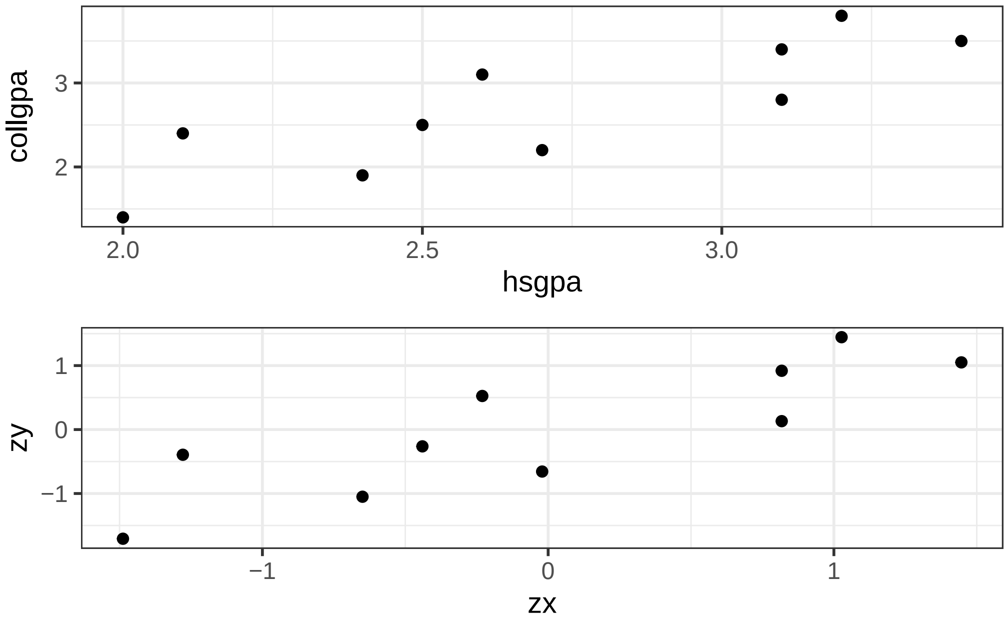
```
# A tibble: 1 x 1
      r
  <dbl>
1 0.844
```

Note:

```r
p1 <- ggplot(data = Gpa, aes(x = hsgpa, y = collgpa)) +
  geom_point() +
  theme_bw()
p2 <- ggplot(data = values, aes(x = zx, y = zy)) +
  geom_point() +
  theme_bw()
library(gridExtra)
grid.arrange(p1, p2, ncol = 1, nrow = 2)
```




```r
# Or better yet
library(patchwork)
p1/p2
```



Least Squares Regression

The equation of a straight line is

$$y = b_0 + b_1 x$$

where b_0 is the y-intercept and b_1 is the slope of the line. From the equation of the line that best fits the data,

$$\hat{y} = b_0 + b_1 x$$

we can compute a predicted y for each value of x and then measure the error of the prediction. The error of the prediction, e_i (also called the residual) is the difference in the actual y_i and the predicted $\hat{y}_i$. That is, the residual associated with the data point (x_i, y_i) is

$$e_i = y_i - \hat{y}_i.$$

The least squares regression line is

$$\hat{y} = b_0 + b_1 x$$

where

$$b_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} = r\frac{s_y}{s_x} \tag{2}$$

and

$$b_0 = \bar{y} - b_1\bar{x} \tag{3}$$

Find the least squares regression line $\hat{y} = b_0 + b_1 x$ for the `Gpa` data.

R Code

```
Gpa %>%
  summarize(b1 = cor(hsgpa, collgpa)*sd(collgpa)/sd(hsgpa),
            b0 = mean(collgpa) - b1*mean(hsgpa))
```

```
# A tibble: 1 x 2
     b1     b0
  <dbl>  <dbl>
1  1.35 -0.950
```

The coefficients are also computed when using the `lm()` function.

R Code

```
mod1 <- lm(collgpa ~ hsgpa, data = Gpa)
mod1
```

```

Call:
lm(formula = collgpa ~ hsgpa, data = Gpa)

Coefficients:
(Intercept)        hsgpa
    -0.9504       1.3470
```

```
summary(mod1)
```

```

Call:
```

```
lm(formula = collgpa ~ hsgpa, data = Gpa)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.48653 -0.37273 -0.02328  0.37365  0.54817 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -0.9504     0.8318  -1.143  0.28625   
hsgpa         1.3470     0.3027   4.449  0.00214 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4333 on 8 degrees of freedom
Multiple R-squared:  0.7122,	Adjusted R-squared:  0.6762 
F-statistic:  19.8 on 1 and 8 DF,  p-value: 0.002141
```

```r
library(moderndive)
get_regression_table(mod1)
```

```
# A tibble: 2 x 7
  term      estimate std_error statistic p_value lower_ci upper_ci
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept    -0.95     0.832     -1.14   0.286   -2.87     0.968
2 hsgpa         1.35     0.303      4.45   0.002    0.649    2.04 
```

Find the residuals for `mod1`.

R Code

```r
get_regression_points(mod1)
```

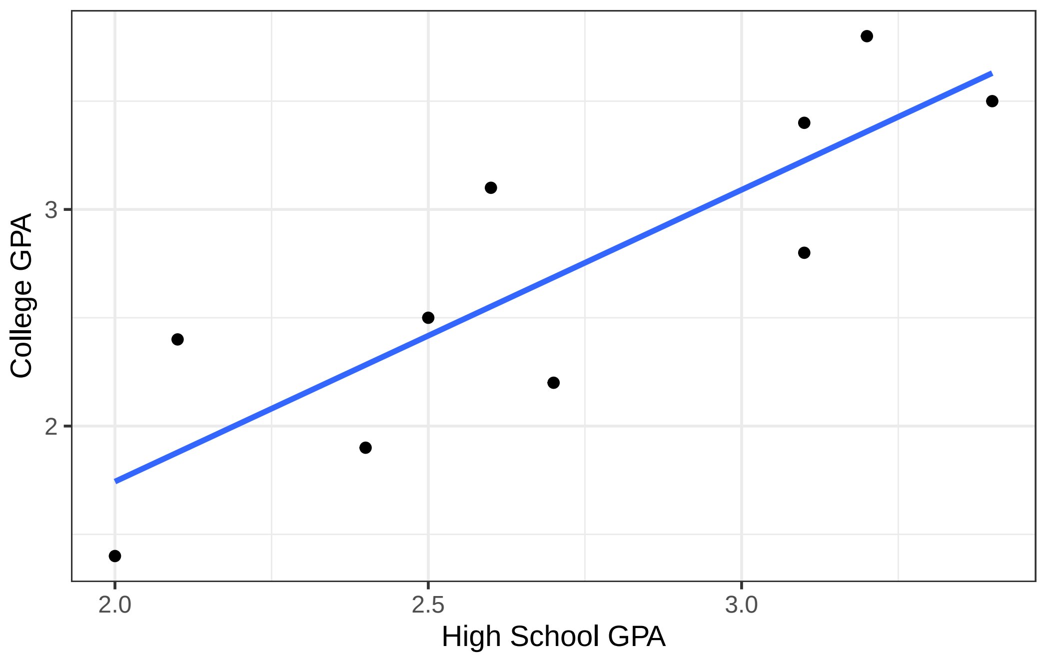
```
# A tibble: 10 x 5
      ID collgpa hsgpa collgpa_hat residual
   <int>   <dbl> <dbl>       <dbl>    <dbl>
 1     1     2.2   2.7        2.69   -0.487
 2     2     2.8   3.1        3.22   -0.425
 3     3     2.4   2.1        1.88    0.522
 4     4     3.8   3.2        3.36    0.44 
 5     5     1.9   2.4        2.28   -0.382
 6     6     3.5   3.4        3.63   -0.129
```

```
 7     7      3.1    2.6          2.55     0.548
 8     8      1.4    2            1.74    -0.344
 9     9      3.4    3.1          3.22     0.175
10    10      2.5    2.5          2.42     0.083
```

Add the least squares line to the scatterplot for `collgpa` versus `hsgpa`.

R Code

```
ggplot(data = Gpa, aes(x = hsgpa, y = collgpa)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "High School GPA", y = "College GPA") +
  theme_bw()
```



Assessing the fit

R Code

```
library(ggfortify)
autoplot(mod1, ncol = 2, nrow = 1, which = 1:2) + theme_bw()
```