Module 04: More Practice with Correlation & Simple Linear Regression

Patrick J. Rosopa, Ph.D.

September 27, 2023

Contents

Objective

- More practice using `R` for correlation and simple linear regression. However, instead of using `R` script, we use `R Markdown`.

Example: Maximum Heart Rate vs. Age

Based on medical research, the maximum heart rate (HR_{max}) of a person is purported to be related to age (Age) by the equation:

$$\text{HR}_{max} = 220 - \text{Age}$$

We will use a small dataset to investigate this relationship.

Load the dataset

There are several ways to load a dataset into `R`:

- If the data is not too large, you can type the data into `R`
- Importing data over the Internet

```
dat <- read.csv('insert_link_to_file_here', header = T)
```

- Read the dataset from your computer

```
# Download the data & ensure it is located in your project. In this example, the
# data file is located in a folder called Data in my current working directory.

dat <- read.csv('Data/maxHeartRate.csv', header = T)
str(dat)
```

```
## 'data.frame':    15 obs. of  2 variables:
##  $ Age         : int  18 23 25 35 65 54 34 56 72 19 ...
##  $ MaxHeartRate: int  202 186 187 180 156 169 174 172 153 199 ...
```

When using the `str()` function, we can see that there are 15 observations and 2 variables. Both variables are integer class.

Examine the data before fitting models

```
summary(dat)
```

```
##       Age         MaxHeartRate
##  Min.   :18.00   Min.   :153.0
##  1st Qu.:23.00   1st Qu.:173.0
##  Median :35.00   Median :180.0
##  Mean   :37.33   Mean   :180.3
##  3rd Qu.:48.00   3rd Qu.:190.0
##  Max.   :72.00   Max.   :202.0
```

```
sd(dat$Age); sd(dat$MaxHeartRate)
```

```
## [1] 17.48741
```

```
## [1] 14.63102
```

```
# Compute the covariance between Age and MaxHeartRate
cov(dat$Age, dat$MaxHeartRate)
```

```
## [1] -243.9524
```

```
# Compute the correlation between Age and MaxHeartRate
cor(dat$Age, dat$MaxHeartRate)
```

```
## [1] -0.9534656
```

We can see Q_1, Q_2, and Q_3 for each numeric variable as well as the mean. It looks like the mean of *Age* and *MaxHeartRate* is 37.33 and 180.3, respectively.

Although the covariance between the two variables is negative, we cannot interpret the strength or magnitude of the relationship because covariance is an unstandardized measure of linear association. Pearson's correlation is much more useful because it is a standardized measure of linear association. Because r = -.95, it appears that there is a strong negative linear association between *Age* and *MaxHeartRate*. As *Age* increases, *MaxHeartRate* tends to decrease.

From the formulas for calculating Pearson's correlation (see slides), we can confirm the estimate of the correlation.

```
cov.xy <- cov(dat$Age, dat$MaxHeartRate)

cov.xy/(sd(dat$Age) * sd(dat$MaxHeartRate))
```

```
## [1] -0.9534656
```

```
age.z <- scale(dat$Age)
hr.z <- scale(dat$MaxHeartRate)

sum(age.z*hr.z)/14
```

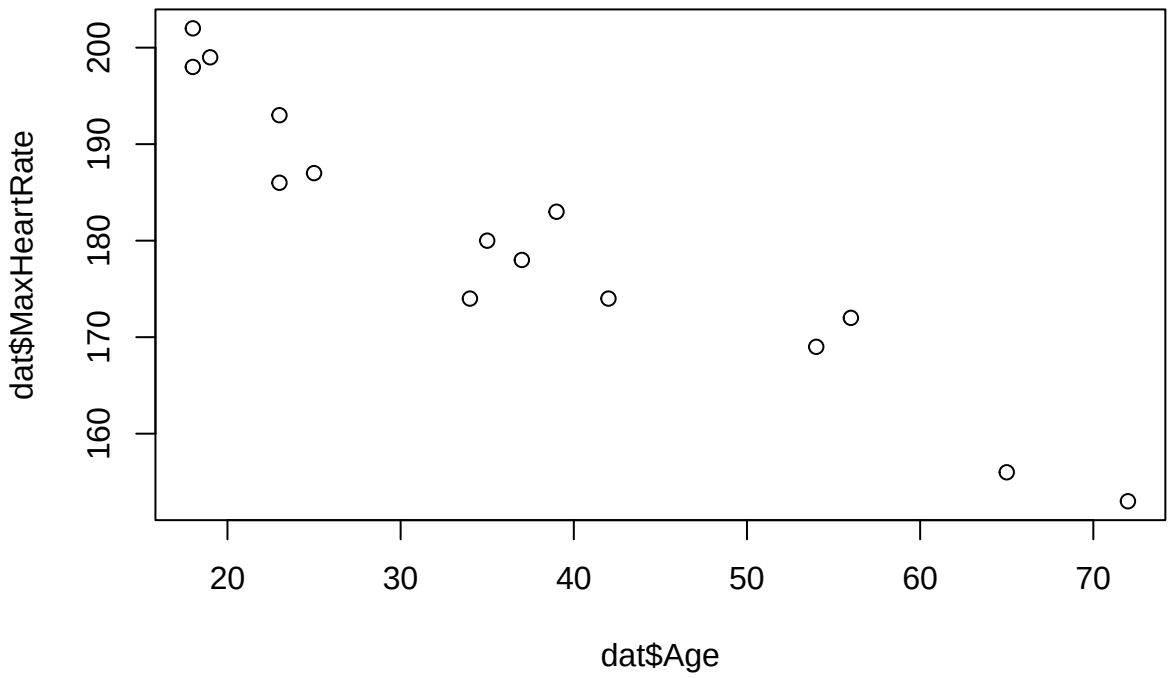
```
## [1] -0.9534656
```

Recall from the readings on correlation and regression, when the correlation is squared, this is referred to as the "coefficient of determination." The coefficient of determination represents the proportion of variance in the outcome that can be attributed to the predictor (or predictors, in the case of multiple regression). Thus, it appears that *Age* explains approximately 90.9% of the variance in *MaxHeartRate*.

Plot the data before fitting models

Below is a scatterplot using the `plot()` function. The first argument is the predictor (*Age*) and the second argument is for the response (*MaxHeartRate*).

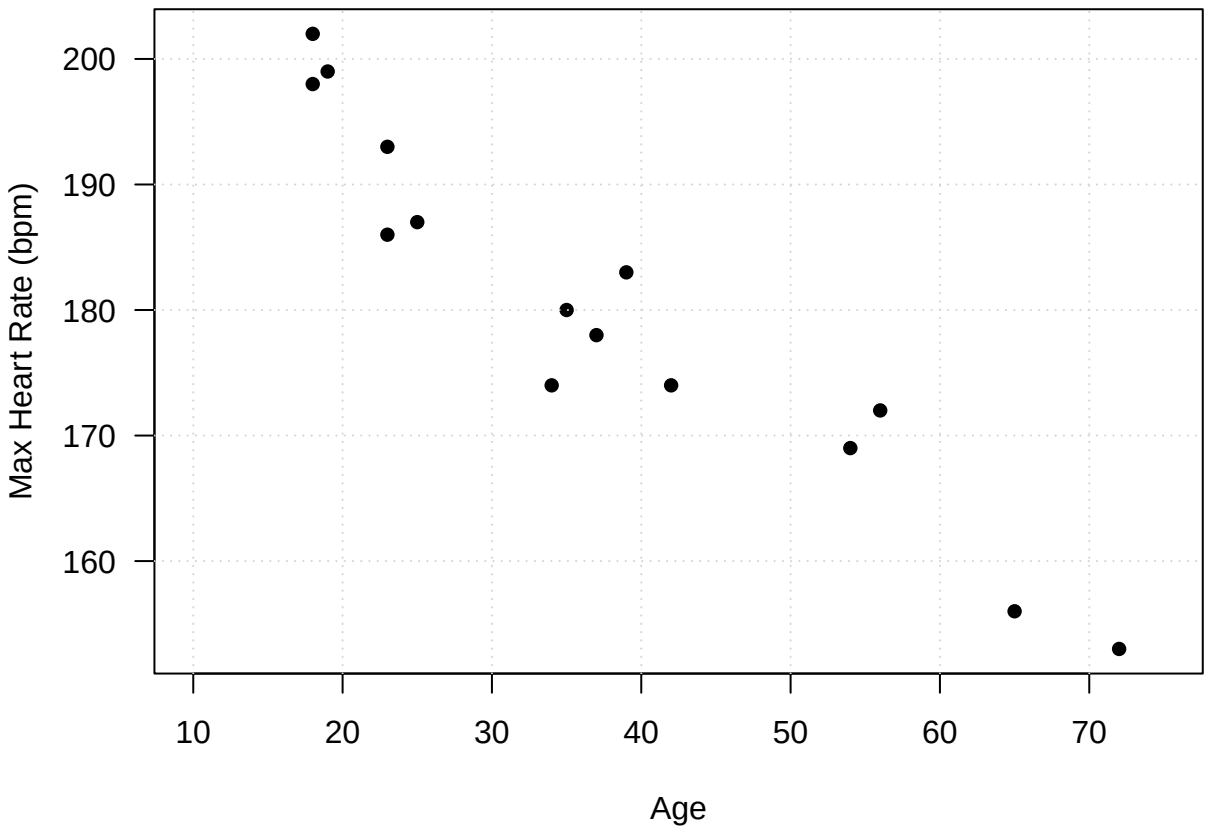
```
plot(dat$Age, dat$MaxHeartRate)
```




```
# with(dat, plot(Age, MaxHeartRate))
```

We can make the plot look nicer by using other optional arguments. (type ?plot in the console to learn more)

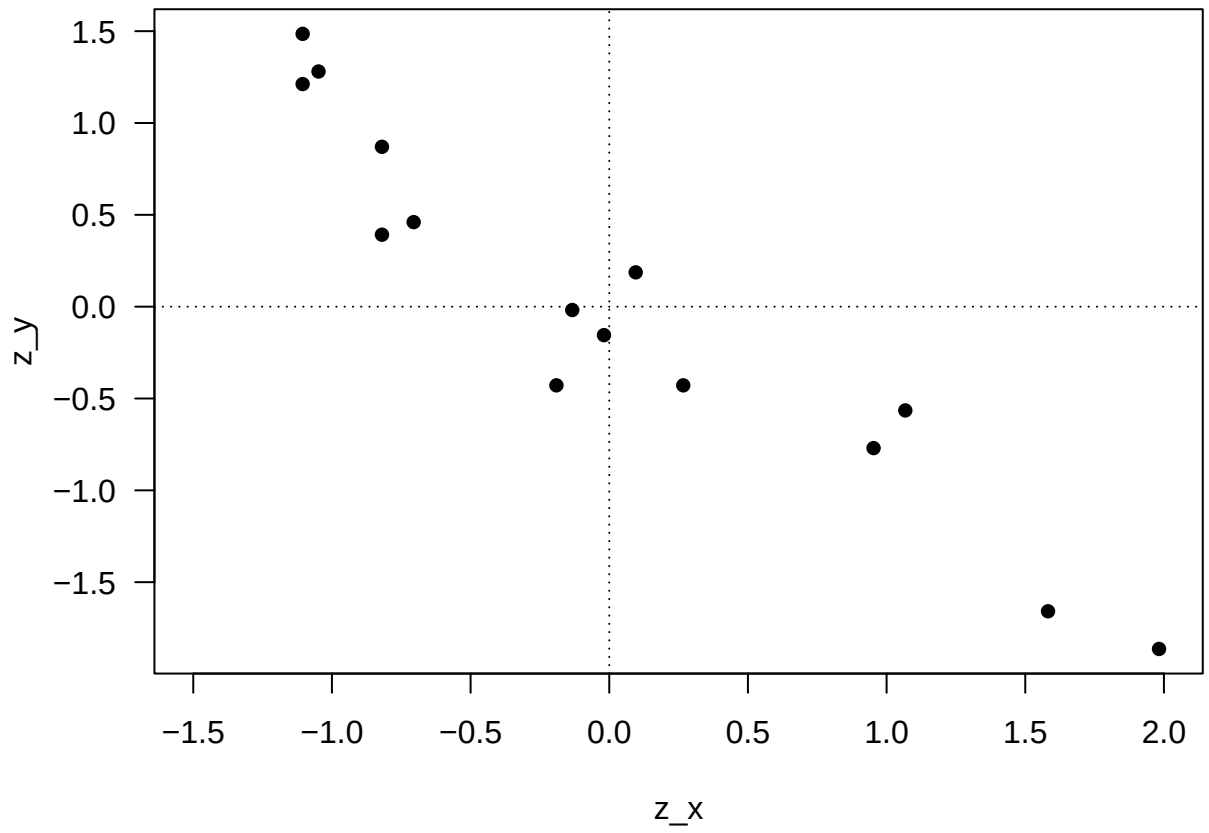
```
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(dat$Age, dat$MaxHeartRate,
     pch = 16,
     xlab = "Age",
     ylab = "Max Heart Rate (bpm)",
     xlim = c(10,75))
grid()
```



Below, we plot the standardized variables. Note that when the variables are standardized the scatter of data is now centered at the origin: (0,0).

```
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(scale(dat$Age), scale(dat$MaxHeartRate),
     pch = 16,
     xlab = "z_x",
     ylab = "z_y",
     xlim = c(-1.5,2))

abline(v = 0, h = 0, lty = 3)
```



Fit a simple linear regression model

With simple linear regression, the number of predictor variables (p) is 1. In this example, our predictor is *Age*. We can calculate the regression coefficients as well as the standard deviation of the random error. Here, *Age* is assigned to `x` and *MaxHeartRate* is assigned to `y`.

- Estimated slope

```
x <- dat$Age; y <- dat$MaxHeartRate
y_diff <- y - mean(y)
x_diff <- x - mean(x)
b_1 <- sum(y_diff * x_diff) / sum((x_diff)^2)
b_1
```

```
## [1] -0.7977266
```

```
# b1 <- cor(x,y)*(sd(y)/sd(x))
```

- Estimated intercept

```
b_0 <- mean(y) - mean(x) * b_1
b_0
```

```
## [1] 210.0485
```

- Fitted values

When we plug in scores for the predictor (*Age*) into the regression equation, we obtain the predicted values, or fitted values, on the outcome (*MaxHeartRate*). Predicted values can be denoted by the symbol $\hat{y}$. For each of the 15 observations, we can obtain predicted values.

```
y_hat <- b_0 + b_1 * x
y_hat
```

```
##  [1] 195.6894 191.7007 190.1053 182.1280 158.1962 166.9712 182.9258 165.3758
##  [9] 152.6121 194.8917 191.7007 176.5439 195.6894 178.9371 180.5326
```

- Least squares residuals

Recall that for a linear regression, the least squares residuals are calculated as the difference between the observed scores on y and the predicted scores on y ($\hat{y}$). Because there are 15 observations, there are 15 residuals.

```
y - y_hat
```

```
##  [1]  6.3106197 -5.7007474 -3.1052943 -2.1280287 -2.1962317  2.0287761
##  [7] -8.9257552  6.6242292  0.3878543  4.1083463  1.2992526 -2.5439427
## [13]  2.3106197  4.0628776 -2.5325755
```

- $\hat{\sigma}$

The estimated variance due to random error is also referred to as *mean square error*. When we take the square root, we get $\hat{\sigma}$. In some textbooks, it is also referred to as root mean square error. It is the square root of the sum of the squared residuals divided by the error degrees of freedom.

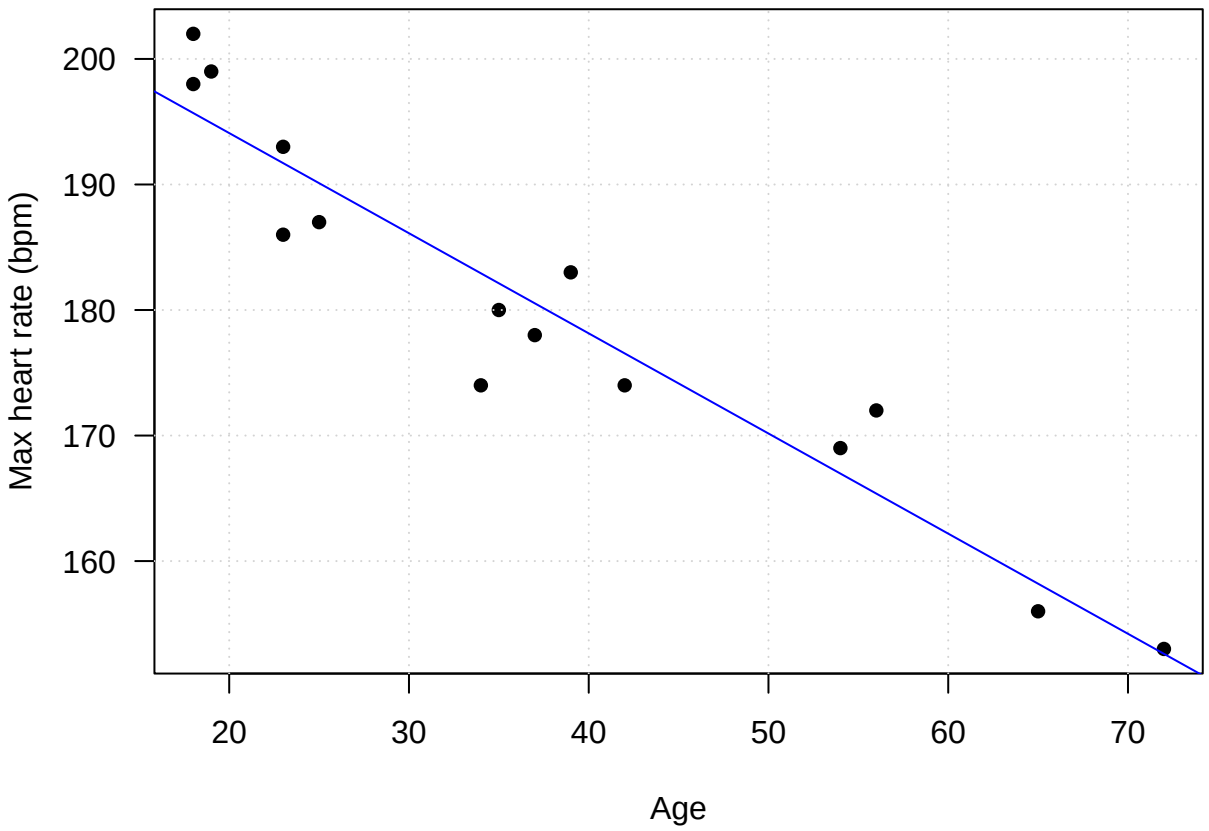
```
sigma2 <- sum((y - y_hat)^2) / (length(y) - 2)
sqrt(sigma2)
```

```
## [1] 4.577799
```

Add the fitted regression line to the scatterplot

```
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(dat$Age, dat$MaxHeartRate,
     pch = 16,
     xlab = "Age",
     ylab = "Max heart rate (bpm)")
grid()

# abline(v = mean(dat$Age), lty=2, col="red")
# abline(h = mean(dat$MaxHeartRate), lty=2, col="red")
abline(a = b_0, b = b_1,
       col = "blue")
```



We will learn about some common approaches to check the fit of our model as well as the normality and constant variance (homoscedasticity) assumptions. The least squares residuals are needed to check these assumptions.

Optional: Matrix approach to linear regression

To perform the matrix calculations, we need the model matrix ($\mathbf{X}$) and the response vector ($\mathbf{y}$).

```
X.mod <- as.matrix(cbind(1,dat[,1]))
y <- matrix(dat[,2], ncol=1)

dim(X.mod)
```

```
## [1] 15  2
```

```
dim(y)
```

```
## [1] 15  1
```

- Estimated intercept and slope

The estimated intercept and slope are now straightforward to calculate using matrices.

```
est <- solve(t(X.mod) %*% X.mod) %*% t(X.mod) %*% y
est
```

```
##              [,1]
## [1,] 210.0484584
## [2,]  -0.7977266
```

- Fitted values

We can calculate all the predicted values ($\hat{y}$) using the model matrix ($\mathbf{X}$) and the estimated regression coefficients (intercept and slope). Because there are 15 observations, we will have 15 predicted values. In terms of matrices, we have a 15 x 1 vector of predicted values.

```
y_hat <- X.mod %*% est
y_hat
```

```
##           [,1]
##  [1,] 195.6894
##  [2,] 191.7007
##  [3,] 190.1053
##  [4,] 182.1280
##  [5,] 158.1962
##  [6,] 166.9712
##  [7,] 182.9258
##  [8,] 165.3758
##  [9,] 152.6121
## [10,] 194.8917
## [11,] 191.7007
## [12,] 176.5439
## [13,] 195.6894
## [14,] 178.9371
## [15,] 180.5326
```

- Least squares residuals

Recall that for a linear regression, the least squares residuals are calculated as the difference between the observed scores on y and the predicted scores on y ($\hat{y}$). In terms of matrices, we have a 15 x 1 vector of residuals.

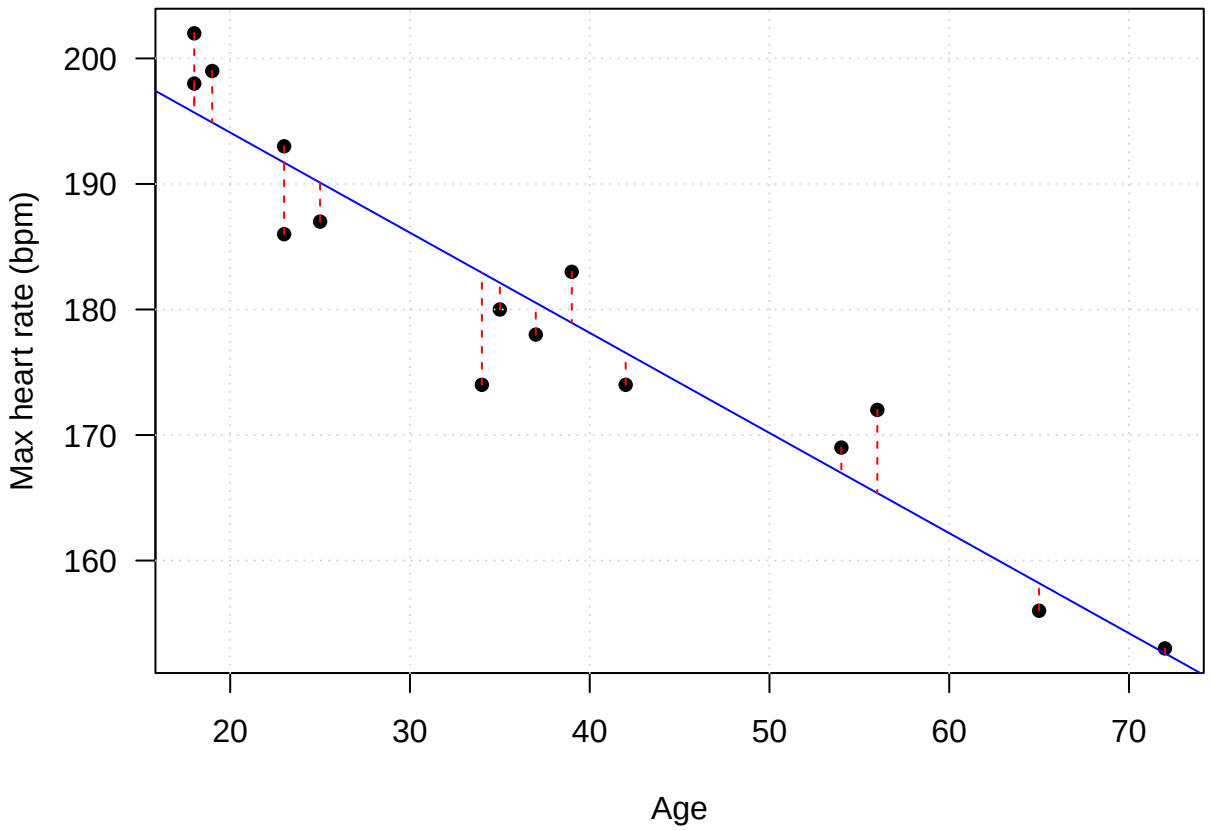
```
y_resid <- dat[,2] - y_hat
y_resid
```

```
##              [,1]
##  [1,]  6.3106197
##  [2,] -5.7007474
##  [3,] -3.1052943
##  [4,] -2.1280287
##  [5,] -2.1962317
##  [6,]  2.0287761
##  [7,] -8.9257552
##  [8,]  6.6242292
```

```
##  [9,]  0.3878543
## [10,]  4.1083463
## [11,]  1.2992526
## [12,] -2.5439427
## [13,]  2.3106197
## [14,]  4.0628776
## [15,] -2.5325755
```

Let's recreate the scatterplot above, but this time we will add red dashed lines representing the residuals for each observation. A residual is calculated as y - $\hat{y}$. Residuals are represented by red dashed lines. The residuals are vertical distances parallel to the y-axis that represent how much y differs from $\hat{y}$.

```
par(las = 1, mar = c(4.1, 4.1, 1.1, 1.1))
plot(dat$Age, dat$MaxHeartRate,
     pch = 16, xlab = "Age",
     ylab = "Max heart rate (bpm)")
grid()
abline(a = b_0, b = b_1, col = "blue")
segments(dat$Age, dat$MaxHeartRate,
         dat$Age, y_hat, lty=2, col='red')
```



- $\hat{\sigma}$

The estimated variance due to random error is also referred to as *mean square error*. When we take the square root, we get $\hat{\sigma}$. In some textbooks, it is also referred to as root mean square error. It is the square root of the sum of the squared residuals divided by the error degrees of freedom.

```
sqrt(t(y_resid) %*% y_resid / (dim(X.mod)[1] - 2))
```

```
##          [,1]
## [1,] 4.577799
```

Let R do the work

In PSYC 8100, we will use the `lm` function often for analyses. `lm` is short for (general) linear model. Remember that the dependent variable is the first argument. Then, the predictors should appear to the right of the ~ symbol. I recommend using the `data` argument. Avoid attaching a data frame using `attach` because it is easy to forget to `detach` the data frame.

```
fit <- lm(MaxHeartRate ~ Age, data = dat)
summary(fit)
```

```
##
## Call:
## lm(formula = MaxHeartRate ~ Age, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9258 -2.5383  0.3879  3.1867  6.6242
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 210.04846    2.86694   73.27  < 2e-16 ***
## Age          -0.79773    0.06996  -11.40 3.85e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.578 on 13 degrees of freedom
## Multiple R-squared:  0.9091, Adjusted R-squared:  0.9021
## F-statistic:   130 on 1 and 13 DF,  p-value: 3.848e-08
```

We readily obtain the estimated intercept and slope for *Age*. Note that the standard deviation of the random error appears below the estimated parameters, $\hat{\sigma} = 4.578$. The $R^2 = .9091$, suggesting that about 91% of variance in Max heart rate can be attributed to a person's age. The overall model is statistically significant, $F(1,13) = 130.01$, $p < .001$.

- Regression coefficients (estimated intercept and slope)

```
fit$coefficients
```

```
## (Intercept)         Age
## 210.0484584  -0.7977266
```

```
confint(fit)
```

```
##                  2.5 %      97.5 %
## (Intercept) 203.854813 216.2421034
## Age          -0.948872  -0.6465811
```

- Fitted values

```
fit$fitted.values    # shorter to use fit$fitted
```

```
##        1        2        3        4        5        6        7        8
## 195.6894 191.7007 190.1053 182.1280 158.1962 166.9712 182.9258 165.3758
##        9       10       11       12       13       14       15
## 152.6121 194.8917 191.7007 176.5439 195.6894 178.9371 180.5326
```

- Least squares residuals

```
fit$residuals    # shorter to use fit$resid
```

```
##          1          2          3          4          5          6          7
##  6.3106197 -5.7007474 -3.1052943 -2.1280287 -2.1962317  2.0287761 -8.9257552
##          8          9         10         11         12         13         14
##  6.6242292  0.3878543  4.1083463  1.2992526 -2.5439427  2.3106197  4.0628776
##         15
## -2.5325755
```

- $\hat{\sigma}$

```
summary(fit)$sigma
```

```
## [1] 4.577799
```

The estimated regression equation is as follows:

$$\hat{y}_i = 210.048 - .798x_i$$

Let's consider conceptually what we accomplished in simple linear regression. We estimated a set of parameters (population intercept & population slope) such that the linear combination on the predictor side of the equation is **optimally** related to the outcome. We calculated the intercept and slope and we computed the predicted scores on the outcome.

Let's correlate the observed scores on y with the predicted scores on y ($\hat{y}$). Using the plain old Pearson's correlation (r), $r = .9534656$. When squared, we get the coefficient of determination. See the output above when using the `summary` function on the `lm` object `fit` and compare to the calculations below.

```
cor(y, y_hat)
```

```
##           [,1]
## [1,] 0.9534656
```

```
cor(y, y_hat)^2
```

```
##           [,1]
## [1,] 0.9090967
```

Amazing! The Pearson correlation between the observed scores and the predicted scores ($r_{y\hat{y}}$) is actually the multiple R. When squared, we obtain R^2. Another amazing concept is that this is true not only for simple linear regression ($p = 1$), but also multiple linear regression where $p > 1$. Thus, with multiple linear regression, we are trying to discover the $(p + 1)$ regression coefficients that result in an optimal linear combination of our predictors that will maximally correlate with the outcome.

Confidence Intervals vs. Prediction Intervals

We have estimated a simple linear regression. We have one continuous predictor. We found that the predictor was significantly related to the outcome. Depending on the context, we may want to use our simple linear regression equation to make predictions. For example, given a person's *Age*, what is the person's predicted *MaxHeartRate*? We can just plug this value into the equation.

$$\hat{y}_i = 210.048 - .798x_i$$

If a person's Age is 25, their predicted MaxHeartRate is 190.1. If a person's Age is 70, their predicted MaxHeartRate is 154.2.

Do Not Extrapolate

We should not attempt to make predictions outside the observed range of our predictor variable. Because the predictor, Age, has a minimum of 18 and a maximum of 72, we should not use this equation to predict the MaxHeartRate of someone who is 85 years old. That is, do not extrapolate. Does the relationship remain the same outside the observed range of the predictor? We do not know whether the functional form of the relationship changes outside the range of our predictor. The relationship could change direction from a negative slope to a positive slope or perhaps the relationship is close to 0 outside the observed range of the predictor. What does predicted MaxHeartRate mean if we plugged in a value of 250 for Age? (Maybe the 250 year old person could be a vampire or simply immortal.) What does predicted MaxHeartRate mean if we plugged in a value of 0 for Age? A person who is 0 years old has a MaxHeartRate of 210.048. (Insert confused emoji here.)

```
# Use the predict() function. The first argument is the name of
# your lm object. In this case, when I conducted the regression,
# I assigned the result to an object called fit.

# To obtain predicted values using specific values of predictors,
# pass a list containing the vector of values on the predictor.
# The predictor from our model was called Age. We want to predict
# max heart rate when age is 25 vs. 70.
predict(fit, new=list(Age=c(25,70)))
```

```
##        1        2
## 190.1053 154.2076
```

We calculated a predicted score on the outcome using a value on the predictor. What if we wanted a confidence interval around the predicted score? There is an important difference between confidence intervals and prediction intervals. Confidence intervals around the predicted value provide an estimated interval around the "mean function" (i.e., the regression surface at the specific values on the predictor(s)) while prediction intervals around the predicted value provide an estimated interval assuming we have some new observation with the specific set of value(s) on the predictor(s). For now, recognize that confidence intervals will be narrower than prediction intervals.

We cover this topic in more detail when we get to multiple linear regression.