R Notebook

This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

#load necessary libraries  
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

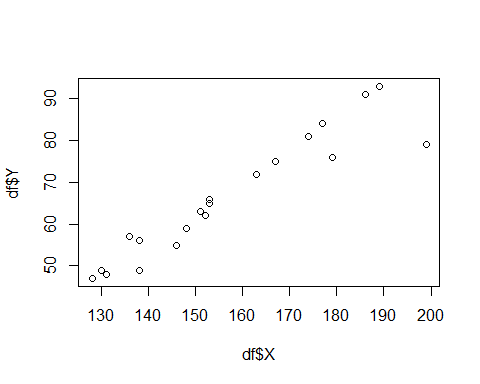
library(readxl)  
library(ggplot2)  
library(ggpubr)

# 1+2) Solution:

# Importing the dataset into a dataframe  
file\_name = "simpledata.csv"  
df = read.csv(file = file\_name)  
df

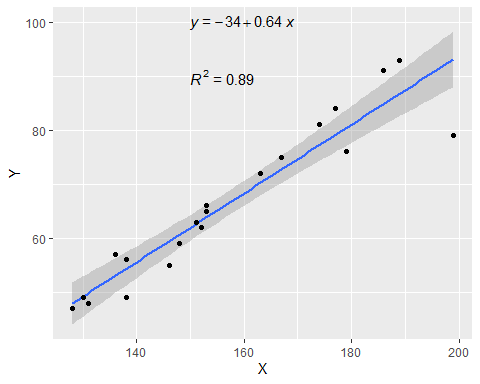
## X Y  
## 1 151 63  
## 2 174 81  
## 3 138 56  
## 4 186 91  
## 5 128 47  
## 6 136 57  
## 7 179 76  
## 8 163 72  
## 9 152 62  
## 10 131 48  
## 11 153 65  
## 12 177 84  
## 13 148 59  
## 14 189 93  
## 15 138 49  
## 16 146 55  
## 17 199 79  
## 18 167 75  
## 19 153 66  
## 20 130 49

# Scatterplot  
plot(df$X, df$Y)



#create plot with regression line, regression equation, and R-squared  
ggplot(data=df, aes(x=X, y=Y)) +  
 geom\_smooth(method="lm") +  
 geom\_point() +  
 stat\_regline\_equation(label.x=150, label.y=100) +  
 stat\_cor(aes(label=..rr.label..), label.x=150, label.y=90)

## `geom\_smooth()` using formula 'y ~ x'



# Dataframe summary  
summary(df)

## X Y   
## Min. :128.0 Min. :47.00   
## 1st Qu.:138.0 1st Qu.:55.75   
## Median :152.5 Median :64.00   
## Mean :156.9 Mean :66.35   
## 3rd Qu.:174.8 3rd Qu.:76.75   
## Max. :199.0 Max. :93.00

# Dataframe structure  
str(df)

## 'data.frame': 20 obs. of 2 variables:  
## $ X: int 151 174 138 186 128 136 179 163 152 131 ...  
## $ Y: int 63 81 56 91 47 57 76 72 62 48 ...

# Correlation  
cor(df$X,df$Y)

## [1] 0.944644

*So the the variables have a strong, positive, linear relationship.*

# Linear regression  
fit<- lm(df$Y ~ df$X)  
summary(fit)

##   
## Call:  
## lm(formula = df$Y ~ df$X)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -14.1573 -1.7267 0.7701 2.6045 6.2102   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -33.55669 8.25032 -4.067 0.000723 \*\*\*  
## df$X 0.63675 0.05213 12.215 3.79e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.846 on 18 degrees of freedom  
## Multiple R-squared: 0.8924, Adjusted R-squared: 0.8864   
## F-statistic: 149.2 on 1 and 18 DF, p-value: 3.788e-10

# Intercept and slope  
a = unname(fit$coefficients[1])  
b = unname(fit$coefficients[2])  
paste0("Intercept = ",b)

## [1] "Intercept = 0.636753917008031"

paste0("Slope = ",a)

## [1] "Slope = -33.55668957856"

# Linear Equation  
paste0("y = ",round(a,2),"x + ",round(b,2))

## [1] "y = -33.56x + 0.64"

# 3) Solution:

*The results are same but the calculations take a lot less time in R.*