Simple Linear Regression Assignment

Smitesh Patil, id=22223696

18 November, 2022

Predicting 3 km Running Times based on laboratory testing.

Study Description:

Sixteen well-trained male middle and long distance runners performed a 3 km time trial and a number of running tests in the laboratory including their running velocity at a blood lactate concentration of 4 mmol.l-1 (v4mM). Other variables measured were running velocity at their Lactate Threshold (vTlac), and VO2 max. All the laboratory testing took place on a motorised treadmill, and distance running performance was determined by 3 km time trials on an indoor 200m track.

Aims:

To investigate whether there is we can use linear regression to predict 3 km running time (minutes) from v-4mM (km per hour) in the population of well-trained male middle and long distance runners. Hence to predict 3km running time using running velocity at blood lactate concentration 4 mmol per litre.

- Response Variable: 3km running time (`Running.Time`) measured in minutes
- Explanatory Variable: running velocity at blood lactate concentration at 4mmol per litre (`v4mM`) measured in km/hr

```
library(tidyverse)
```

Read the data and see a few rows

```
running = read.csv("3krunning.csv", header = TRUE)
head(running)
```

```
##   Running.Time v4mM vTlac Rel.14.5 Rel.16.1 VO2Max
## 1         8.23 20.4  19.5     47.1     52.4   23.4
## 2         8.30 19.5  18.2     48.1     60.0   23.5
## 3         8.62 19.0  17.3     50.3     56.8   22.0
## 4         8.82 18.9  17.8     51.8     56.1   23.0
## 5         9.18 17.8  16.5     48.7     54.1   21.5
## 6         9.23 17.2  15.6     50.5     59.6   20.5
```

Summary Statistics

Task: Calculate the summary statistics for each column in the data and describe the key features of the data.

```
running %>%
  summarise(Mean_v4mM = mean(v4mM),
            SD_v4mM = sd(v4mM),
            Mean_vTlac = mean(vTlac),
            SD_vTlac = sd(vTlac)
            )
```

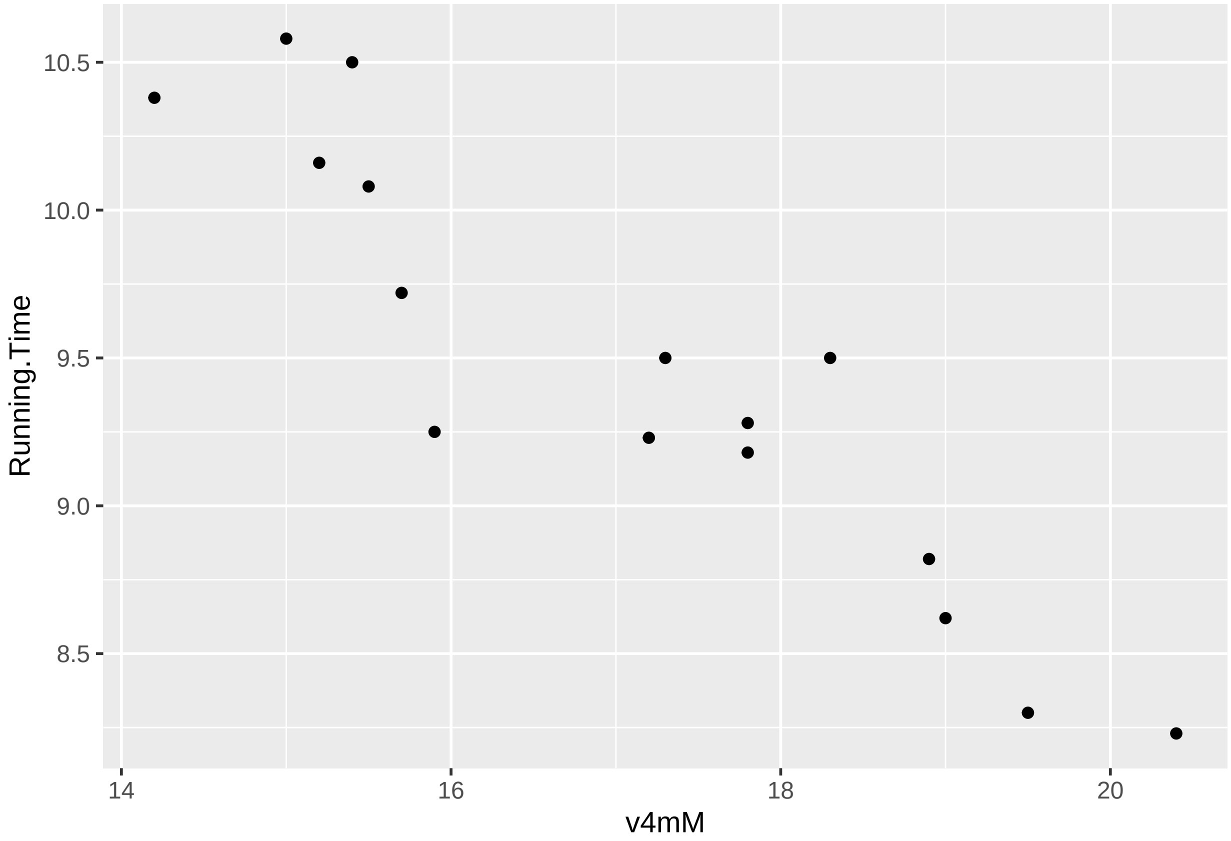
```
##   Mean_v4mM  SD_v4mM Mean_vTlac SD_vTlac
## 1  17.06875 1.848141      15.95 1.775763
```

Scatterplot

Task: Make a labelled scatterplot of `v4mM` vs `Running.Time` and interpret it.

Interpretation : On a look, it seems that the variables Running.Time and v4mM have a negative correlation. I.E as the value of Running.Time increases value of v4mM decreases.

```
ggplot(running, aes(x = v4mM , y = Running.Time))+
  geom_point()
```



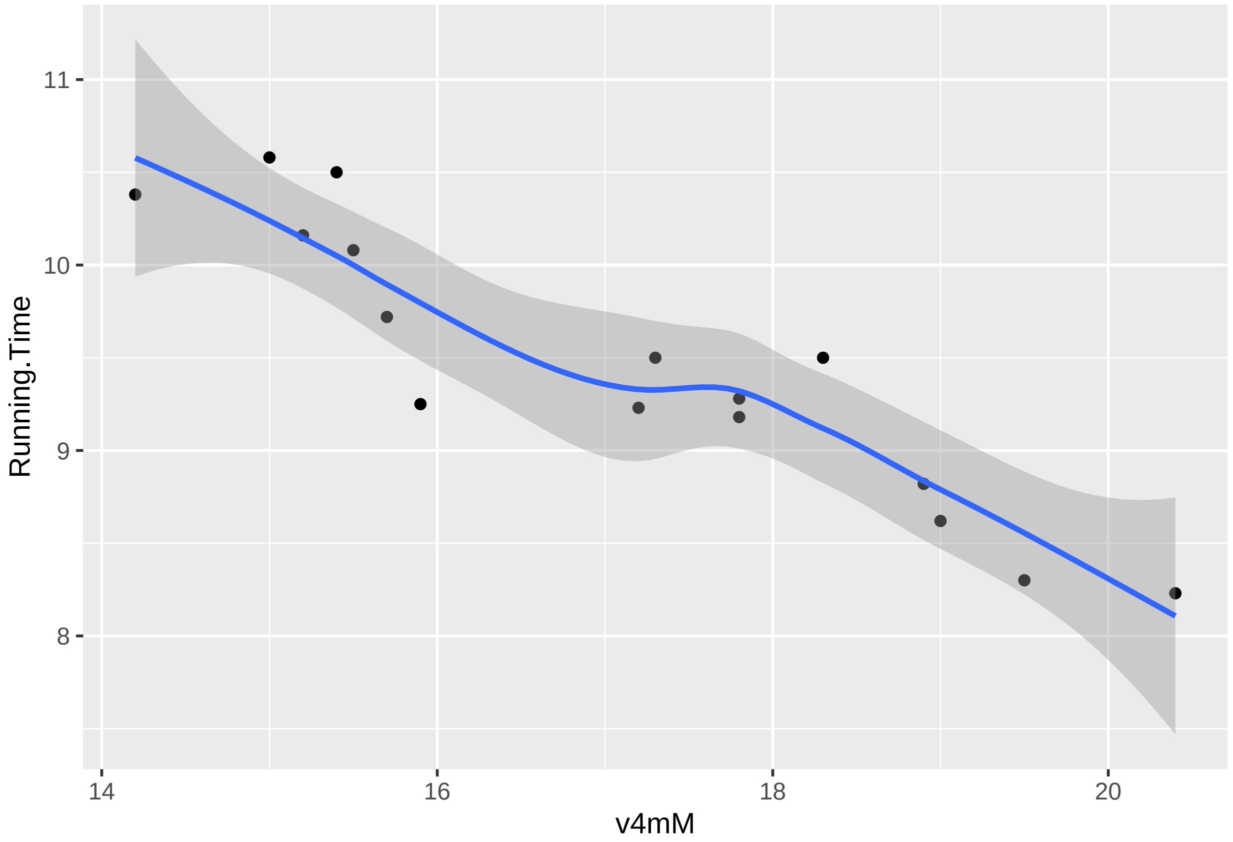
Scatterplot with smoother.

Task: Add a smooth line to the scatter plot produced in the previous task, and include the new plot below.

Interpretation : Smoothing aids us to see a pattern by plotting a line and intervals to a certain width, here we can obeserve for certain that there is a negative correlation between Running.Time and v4mM. Using pearson correlation would give us concrete proof of a negative correlation.

```
ggplot(running, aes(x = v4mM , y = Running.Time))+
  geom_point()+
  geom_smooth()
```

```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



Task: What does the smoother suggest regarding the suitability of a simple linear regression model for this relationship?

Interpretation : In simple linear regression, we try to fit our hypothesis (a line) by calculating error on existing hypothesis and recalculating slope and intercept. The smoother suggest we can fit a line to relatively good accuracy to predict Running.Time based on its v4mM value. A simple linear regression model, basically refers to the fact that there is a single dependent variable (v4mM) for predicting the target variable (Running.Time)

Correlation coefficient

Task: calculate the correlation coefficient between `v4mM` vs `Running.Time` and interpret it.

Interpretation : Correlation value of -0.925857 between v4mM and Running.Time supports our interpretation that the two variables are highy correlated negatively.

```
running %>% select(v4mM, Running.Time) %>% cor()
```

```
##                   v4mM Running.Time
## v4mM          1.000000    -0.925857
## Running.Time -0.925857     1.000000
```

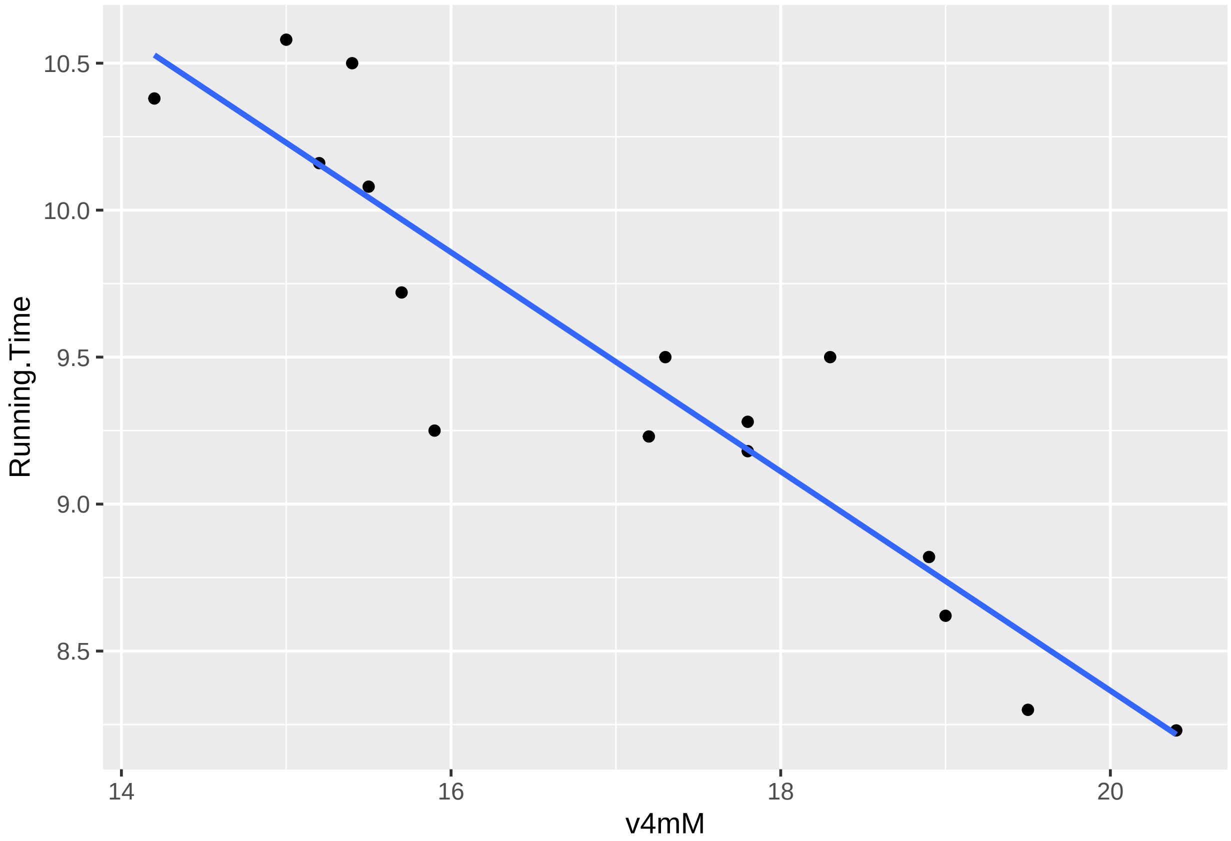
Scatterplot with line of best fit

Task: Add the line of best fit to the scatter plot produced above and interpret it.

Interpretation :geom_smooth or stats_smooth are used to plot a hypothesis for the input data. The se=FALSE argument causes the line to not plot confidence interval and in method we pass lm to plot linee based on linear model.

```
ggplot(running, aes(x = v4mM , y = Running.Time))+
  geom_point()+
  geom_smooth(method = "lm", se = FALSE)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```



### Fitting a Simple Linear Regression Model

Task: Estimate the parameters of the line of best fit for the linear relationship between 3 km running time and v-4mM in the sample. This can then be used for inference about the linear relationship in the population of well-trained male middle and long distance runners.

Interpretation : by running the lm function and passing Running.Time and v4mM, you can predict Running.Time values based on the v4mM values. This gives us the intercept and slope values which can be used to plot our hypothesis with value y = mx+c. The values of paramters are the slope value m = -0.3729 and value of y-intercept 15.82223

```
model = lm(Running.Time ~ v4mM, running)
model
```

```
##
## Call:
## lm(formula = Running.Time ~ v4mM, data = running)
##
## Coefficients:
## (Intercept)         v4mM
##     15.8223      -0.3729
```

Equation of line of best fit

Task: Write down the equation of the line of best fit and also provide an interpretation of the slope and intercept. Does the intercept have a physically meaningful interpretation?

Interpretation : Here the line equation is of the format y = mx +c where m is the slope of the line and c is the y-intercept of the line. by using slope and the y intercept you can plot any line on a number-plane.

```
equation = paste("Y = ", model$coefficients[-1],"*X + ",  model$coefficients[1])
equation
```

```
## [1] "Y =  -0.372853981633588 *X +  15.8222762322351"
```

Make some point predictions

Task: Predict the running time (i.e. `Running.Time`) when running speed at blood lactate concentration 4 mmol/litre (i.e. `v4mM`) are 14, 15, 16, 17, 18, 19 and 20 km per hour.

```
v4mM_pred = data.frame(
  v4mM = c(14, 15, 16, 17, 18, 19, 20))
```

Interval estimation for predicted running times

For each of the predictions produce a 95% confidence interval and 95% prediction interval, and interpret the results carefully.

Interpretation : 95% confidence interval tells us that we are 95% confident that true mean lies between two values, whereas 95% prediction interval tells us that individual values of data lies between two values. Consequently, prediction intervals are larger than confidence interval.

```
predict(model, newdata = v4mM_pred, interval = "confidence")
```

```
##         fit       lwr       upr
## 1 10.602320 10.292450 10.912191
## 2 10.229467  9.990868 10.468065
## 3  9.856613  9.674799 10.038426
## 4  9.483759  9.327551  9.639966
## 5  9.110905  8.934940  9.286869
## 6  8.738051  8.508390  8.967712
## 7  8.365197  8.065626  8.664767
```

```
predict(model, newdata = v4mM_pred, interval = "prediction")
```

```
##         fit      lwr       upr
## 1 10.602320 9.905285 11.299356
## 2 10.229467 9.561059 10.897874
## 3  9.856613 9.206309 10.506916
## 4  9.483759 8.840144 10.127373
## 5  9.110905 8.462212  9.759598
## 6  8.738051 8.072781  9.403320
## 7  8.365197 7.672678  9.057715
```

Plots with confidence and prediction intervals

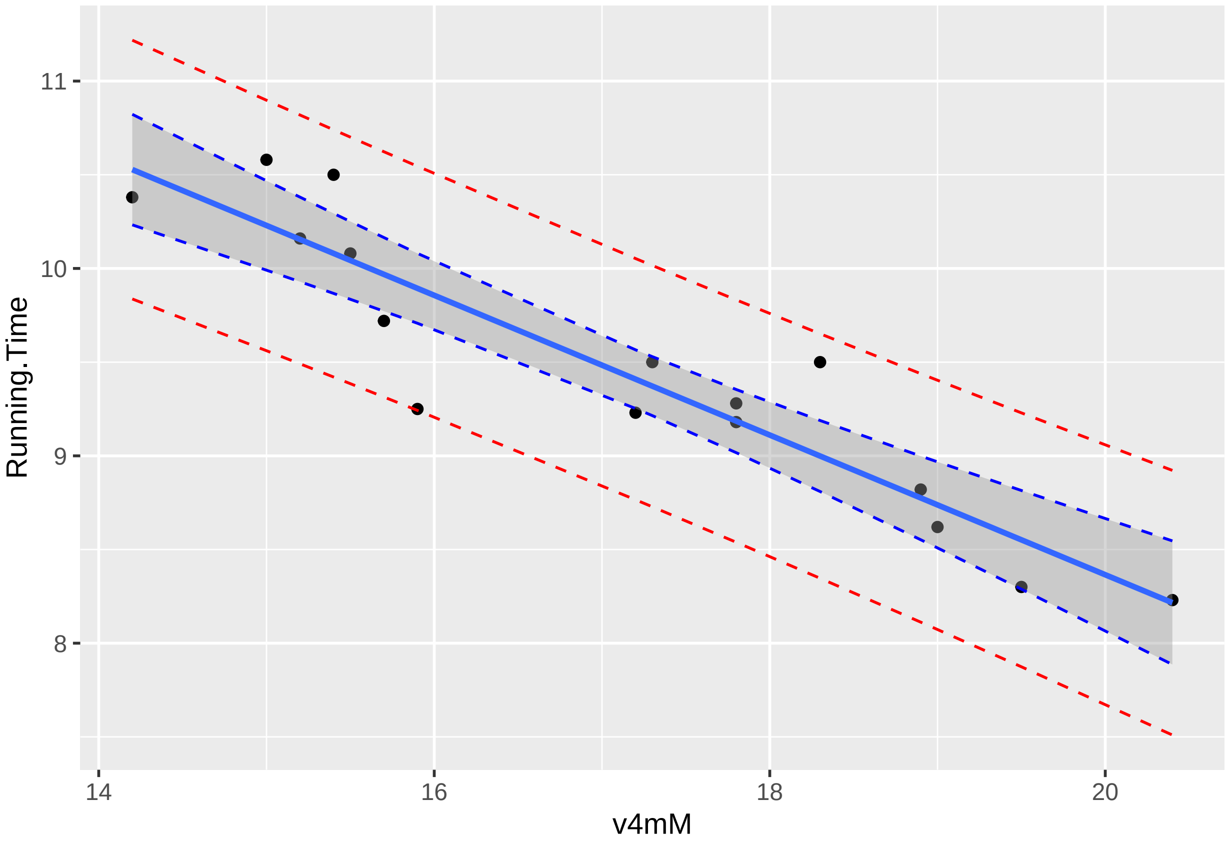
Task: Add the 95% confidence and 95% prediction intervals to the scatter plot with the line of best fit, and interpret.

Interpretation : In the below plot, blue dashed line indicate the confidence interval, whereas the red dashed line indicate prediction interval. As stated before, the prediction interval is larger in range than confidence interval as it gives probability of values lying between two values whereas prediction interval whereas confidence does the same with the true mean.

```
preds_prediction = predict(model, newdata = running, interval = "prediction")

preds_prediction <- data.frame(preds_prediction)

preds_prediction <- preds_prediction %>% rename(
  fit = fit,
  lwr_pred = lwr,
  upr_pred = upr
)

preds_confidence = predict(model, newdata = running, interval = "confidence")

preds_confidence <- data.frame(preds_confidence)

preds_confidence <- preds_confidence %>% rename(
  lwr_con = lwr,
  upr_con = upr
) %>%
  select(lwr_con, upr_con)
```

```
df2 = data.frame(cbind(running, preds_confidence, preds_prediction))

ggplot(df2, aes(y = Running.Time, x = v4mM)) +
         geom_point() +
          stat_smooth(method = lm) +
          geom_line(aes(y = lwr_pred), color = "red", linetype = "dashed")+
          geom_line(aes(y = upr_pred), color = "red", linetype = "dashed")+
          geom_line(aes(y = upr_con), color = "blue", linetype = "dashed")+
          geom_line(aes(y = lwr_con), color = "blue", linetype = "dashed")
```

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



More prediction

Task: Predict the running time (i.e. `Running.Time`) when running speed at blood lactate concentration 4 mmol/litre (i.e. `v4mM`) is 18.9 km per hour.

```
predict(model, newdata = data.frame(v4mM = 18.9))
```

```
##        1
## 8.775336
```

Task: Why is the result here is different from 8.82, the observed running time when running speed at blood lactate concentration 4 mmol/litre (v4mM) is 18.9 mmol.l-1? (see observation row 4)

Interpretation : This is because the value given as prediction by our linear model is just an estimated value that it learns by training on the data provided. The model accuracy depends on many factors like the size of the dataset, the features of the dataset, hyperparamters of the model, etc. Hence, the value of running speed is diffrent than the one predicted.

Task: Predict the running time (i.e. Running.Time) when v4mM is 2.6 km per hour. Explain if you have any concern related to this prediction.

Interpretation : Yes, I am concerned with this prediciton. The value of 'v4mM' provided is an outlier. The values of 'v4mM' are provided in range of 14.2 to 20.4 as 2.6 value is an outlier the value that our hypothesis predicts would not be considered accurate as our model has not trained on data around that value.

```
predict(model, newdata = data.frame(v4mM = 2.6))
```

```
##        1
## 14.85286
```

Overall Conclusion

Task: State your overall conclusions from fitting a linear model for the relationship between 3k running time and the running speed at blood lactate concentration 4 mmol/litre.

Answer : In brief, we found out that variables v4mM and Running.Time are highly negatively correlated hence, we can use the value of v4mM to predict running.time. We ran a simple linear regression model and used the model trained on data to make some predictions. We later stated out that prediction on outliers could be ambigious as the dataset doesn't contain any observed value in that range.