Workshop of Week 7

Saher Manaseer

2022-09-08

Week 7 Workshop

Regression Models with R

1. Regression modelling is a method to find relations among different variables (features) in a dataset. In this activity, we will work on the data provided to you in the file ***Hitters.csv***.

2. After creating a new project and a new Markdown file, the first task is to read the data into RStudio. This can be done by the command read_csv. Before using the command, you need to load the library called "tidyverse"

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

3. Now, run the read_csv commnd, After that, type the name of the dataframe (df) and run the code to see the contents loaded.

```
df <- read_csv("Hitters.csv")
```

```
## Rows: 322 Columns: 20
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): League, Division, NewLeague
## dbl (17): AtBat, Hits, HmRun, Runs, RBI, Walks, Years, CAtBat, CHits, CHmRun...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df
```

```
## # A tibble: 322 x 20
##    AtBat  Hits HmRun  Runs   RBI Walks Years CAtBat CHits CHmRun CRuns  CRBI
##    <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl>
##  1   293    66     1    30    29    14     1    293    66      1    30    29
##  2   315    81     7    24    38    39    14   3449   835     69   321   414
##  3   479   130    18    66    72    76     3   1624   457     63   224   266
##  4   496   141    20    65    78    37    11   5628  1575    225   828   838
##  5   321    87    10    39    42    30     2    396   101     12    48    46
##  6   594   169     4    74    51    35    11   4408  1133     19   501   336
##  7   185    37     1    23     8    21     2    214    42      1    30     9
##  8   298    73     0    24    24     7     3    509   108      0    41    37
##  9   323    81     6    26    32     8     2    341    86      6    32    34
## 10   401    92    17    49    66    65    13   5206  1332    253   784   890
## # ... with 312 more rows, and 8 more variables: CWalks <dbl>, League <chr>,
## #   Division <chr>, PutOuts <dbl>, Assists <dbl>, Errors <dbl>, Salary <dbl>,
## #   NewLeague <chr>
```

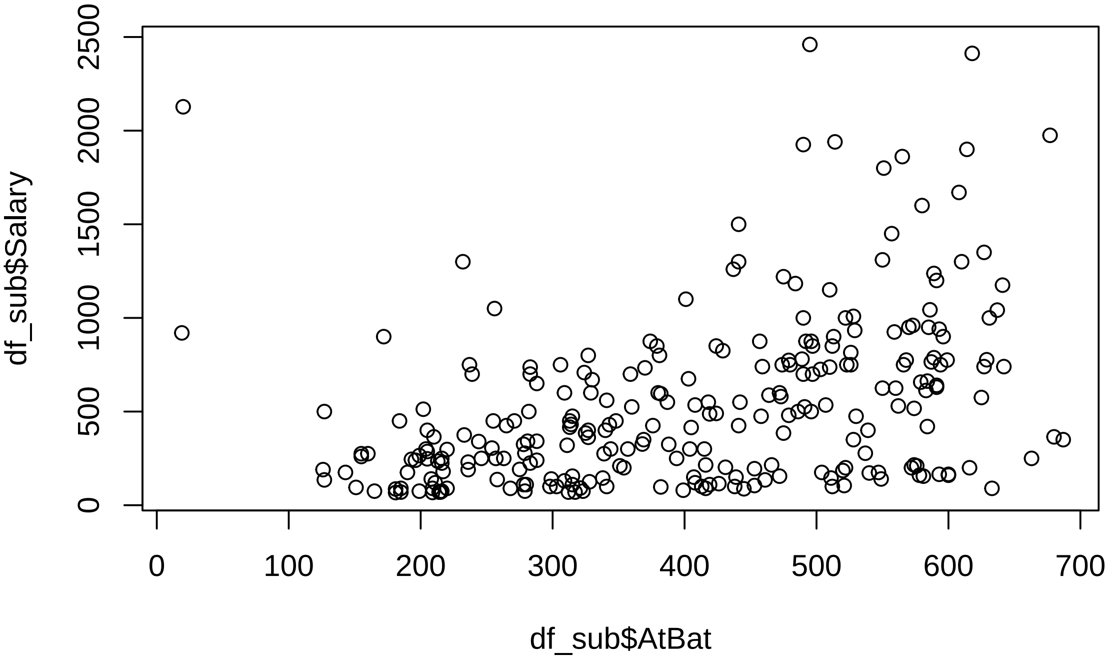
4. The outcome shows all the data. We can see that there are 20 columns (variables) and 322 rows (observations) in the data. However, This large number of columns is difficult to keep track of. Therefore, we will take some of them into new dataframe called df_sub. We will also show the contents as we did in the previous step.

```
df_sub <- select(df, AtBat, CHits, CRuns, CRBI, Salary)
df_sub
```

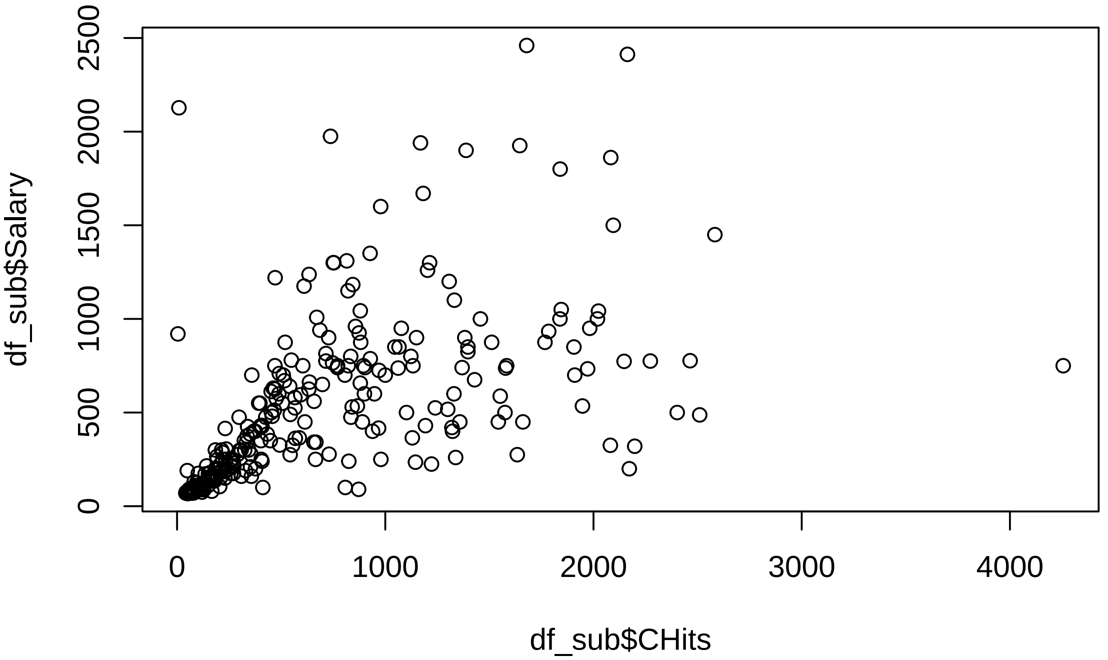
```
## # A tibble: 322 x 5
##    AtBat CHits CRuns  CRBI Salary
##    <dbl> <dbl> <dbl> <dbl>  <dbl>
##  1   293    66    30    29   NA
##  2   315   835   321   414  475
##  3   479   457   224   266  480
##  4   496  1575   828   838  500
##  5   321   101    48    46   91.5
##  6   594  1133   501   336  750
##  7   185    42    30     9   70
##  8   298   108    41    37  100
##  9   323    86    32    34   75
## 10   401  1332   784   890 1100
## # ... with 312 more rows
```

5. Lets plot the relation between salary and each of the other 4 columns

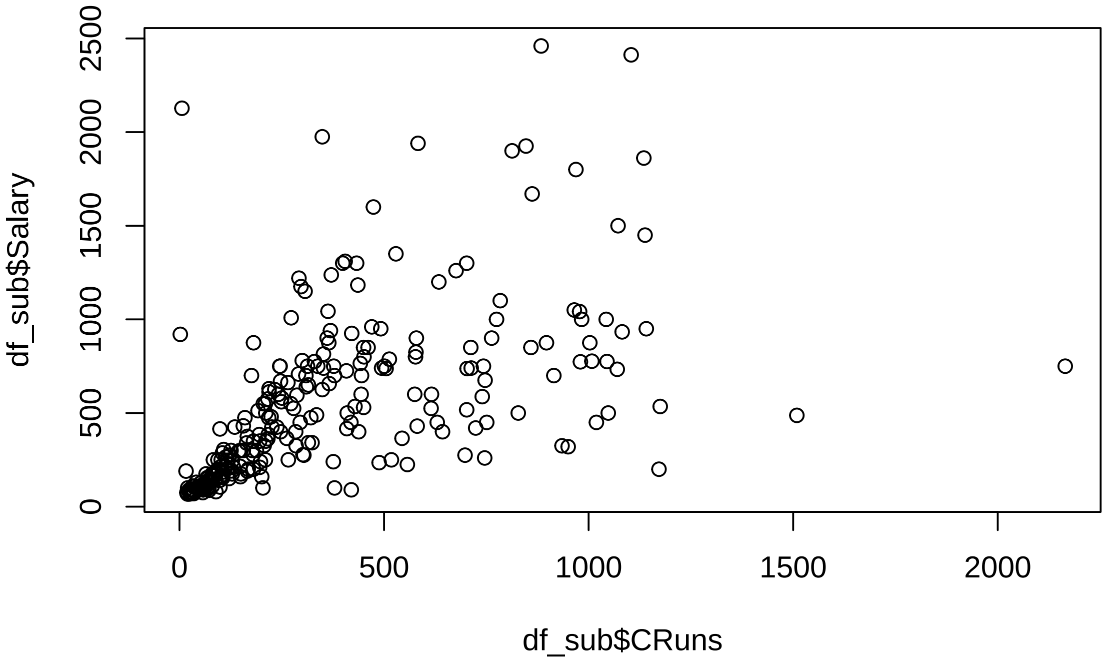
```
plot(df_sub$AtBat,df_sub$Salary)
```



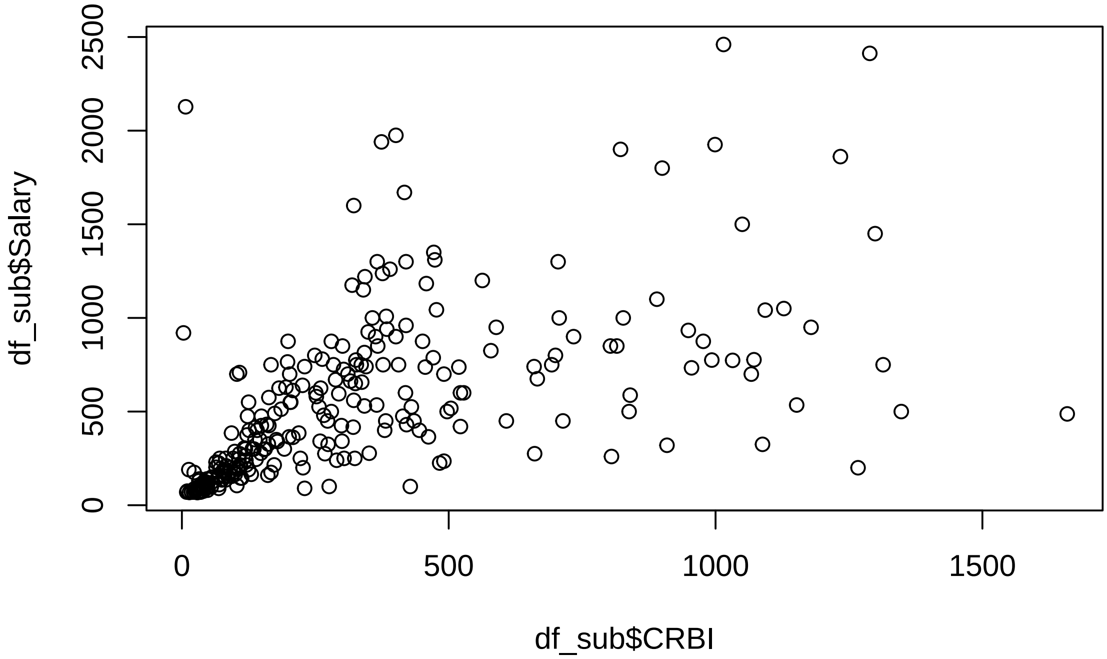
```
plot(df_sub$CHits,df_sub$Salary)
```



```
plot(df_sub$CRuns,df_sub$Salary)
```



```
plot(df_sub$CRBI,df_sub$Salary)
```



6. Now, lets try to fit a model (linear model) to predict the salary based on AtBat column

```
model1 <- lm(Salary ~ AtBat, data = df_sub)
```

7. To have a look at the contents of the model (details of the equation) lets use the command tidy(). This will need a library called **broom**

```
library(broom)
tidy(model1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    47.9     74.8       0.641 5.22e- 1
## 2 AtBat           1.21     0.174     6.94  3.07e-11
```

8. Lets display model summary to evaluate the goodness of fit.

```
summary(model1)
```

```
##
## Call:
## lm(formula = Salary ~ AtBat, data = df_sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -723.21 -237.53  -58.98  176.82 2055.22
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  47.9381    74.8184   0.641    0.522
## AtBat         1.2090     0.1742   6.942 3.07e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 415.3 on 261 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.1558, Adjusted R-squared:  0.1526
## F-statistic: 48.18 on 1 and 261 DF,  p-value: 3.065e-11
```

9. As seen from the previous command, R-squared is not very high. let us try the other variables to build other models.

```
model2 <- lm(Salary ~ CHits, data = df_sub)
summary(model2)
```

```
##
## Call:
## lm(formula = Salary ~ CHits, data = df_sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1135.90  -195.71   -99.43   132.59  1863.86
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 260.03835   34.91386   7.448 1.39e-12 ***
## CHits         0.38202    0.03601  10.609  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 377.8 on 261 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.3013, Adjusted R-squared:  0.2986
## F-statistic: 112.6 on 1 and 261 DF,  p-value: < 2.2e-16
```

```
#AtBat, CHits, CRuns, CRBI, Salary
```

```
model3 <- lm(Salary ~ CRuns, data = df_sub)
summary(model3)
```

```
##
## Call:
## lm(formula = Salary ~ CRuns, data = df_sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1168.36  -197.05   -91.06   134.53  1863.65
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 259.0823    34.1273   7.592 5.62e-13 ***
## CRuns         0.7664     0.0697  10.996  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 373.6 on 261 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.3166, Adjusted R-squared:  0.314
## F-statistic: 120.9 on 1 and 261 DF,  p-value: < 2.2e-16
```

```
#AtBat, CHits, CRuns, CRBI, Salary
```

```
model4 <- lm(Salary ~ CRBI, data = df_sub)
summary(model4)
```

```
##
## Call:
## lm(formula = Salary ~ CRBI, data = df_sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1099.27  -203.45   -97.43   146.37  1847.22
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 274.58039   32.85537   8.357 3.85e-15 ***
## CRBI          0.79095    0.07113  11.120  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 372.3 on 261 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.3215, Adjusted R-squared:  0.3189
## F-statistic: 123.6 on 1 and 261 DF,  p-value: < 2.2e-16
```

10. Lets try the two variables with the highest R-squared values to build a new model

```
model5 <- lm(Salary ~ CHits+CRuns, data = df_sub)
summary(model5)
```

```
##
## Call:
## lm(formula = Salary ~ CHits + CRuns, data = df_sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1161.7  -199.2   -98.4   135.2  1860.3
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 262.0911    34.5844   7.578 6.17e-13 ***
## CHits        -0.1150     0.2036  -0.565   0.5726
## CRuns         0.9880     0.3985   2.480   0.0138 *
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 374.1 on 260 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.3174, Adjusted R-squared:  0.3122
## F-statistic: 60.46 on 2 and 260 DF,  p-value: < 2.2e-16
```

```
#AtBat, CHits, CRuns, CRBI, Salary
```

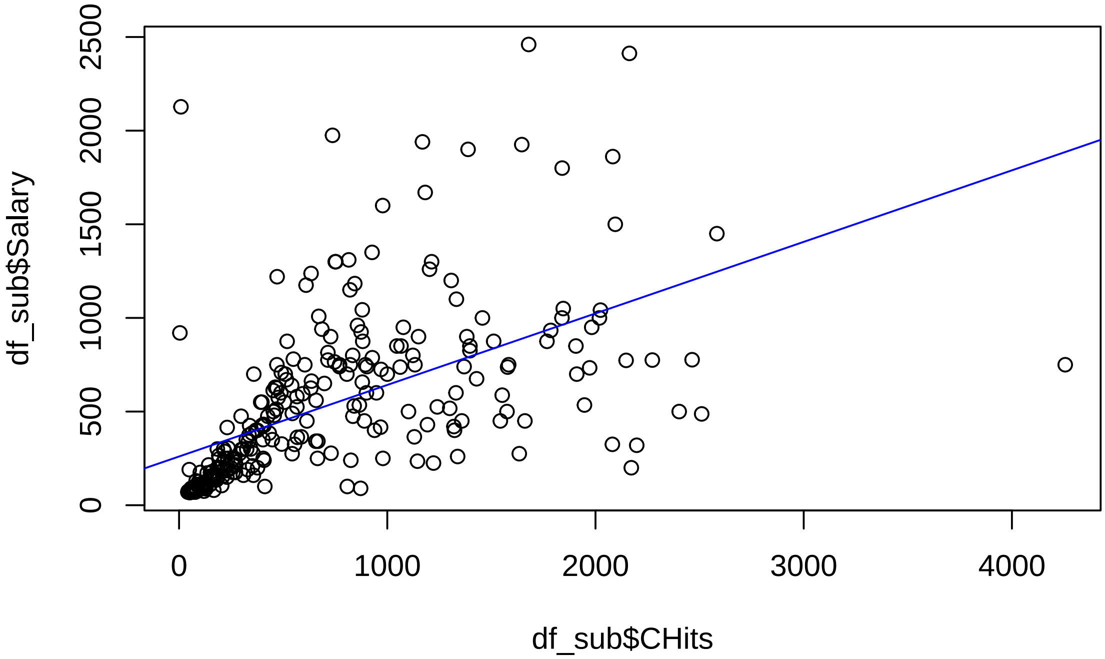
11. Maybe another version of multiplication instead of addition

```
model6 <- lm(Salary ~ CHits*CRuns, data = df_sub)
summary(model6)
```

```
##
## Call:
## lm(formula = Salary ~ CHits * CRuns, data = df_sub)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -896.20 -135.86  -74.31   91.09 1996.98
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.207e+02  4.037e+01   2.990  0.00306 **
## CHits        2.272e-02  1.929e-01   0.118  0.90632
## CRuns        1.577e+00  3.877e-01   4.066 6.35e-05 ***
## CHits:CRuns -3.620e-04  6.125e-05  -5.911 1.07e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 351.9 on 259 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.3986, Adjusted R-squared:  0.3916
## F-statistic: 57.21 on 3 and 259 DF,  p-value: < 2.2e-16
```

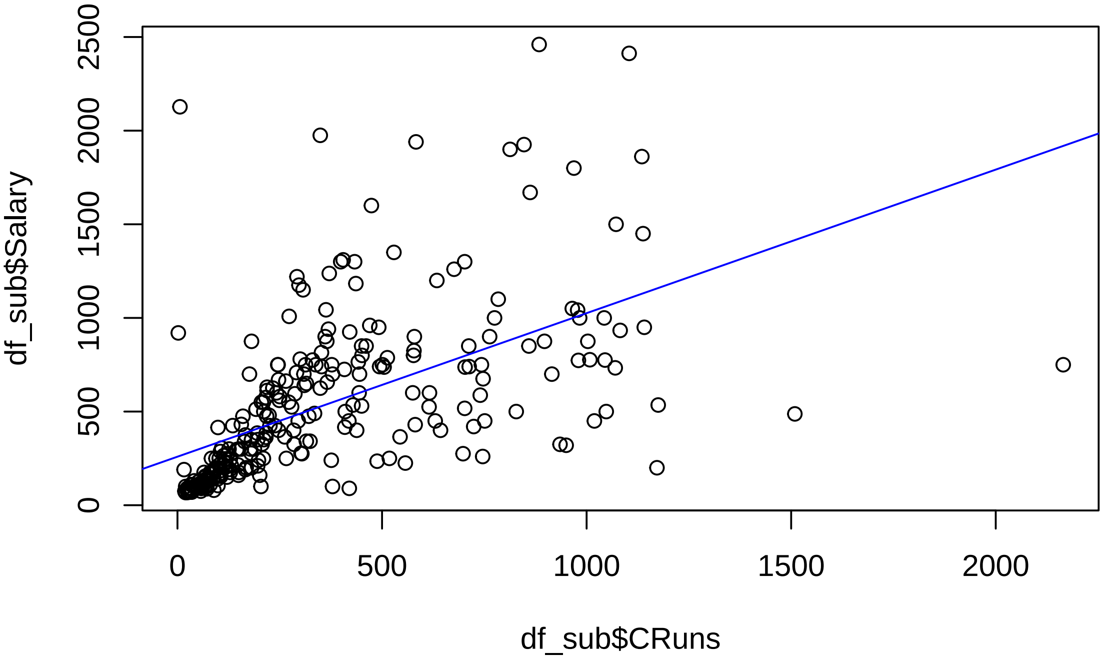
12. Lets fit a line to the plot with the values predicted from model3

```
plot(df_sub$CHits, df_sub$Salary)
abline(model2, col = "blue")
```



13. Lets try model3

```
plot(df_sub$CRuns, df_sub$Salary)
abline(model3, col = "blue")
```



14. State your observations on the goodness of your models.
15. Congratulations! You have just our R regression example.