Homework 1: Simple Linear Regression an Logistic Regression

<MELTEM AKKOCA (31598286002)>

Last compiled on 05 Aralık, 2022

Contents

1 Part 1: Simple Linear Regression 1

2 Part 1: Solution 2

3 Part 2: Logistic Regression 9

4 Part 2: Solution 10

1 Part 1: Simple Linear Regression

Exam grades and weekly spent time on self study x (in hours) of 14 statistics students are given in the following table.

Self study	25.0	26.2	24.9	23.7	22.8	24.6	23.6	23.0	22.5	26.2	25.8	24.0	22.1	21.7
Exam Grades	63	53	52	46	34	47	43	37	40	45	53	42	32	49

1. Create a data frame in `R` with the above data. Plot the data with the weekly spent time on self study in the x-axis and exam grades on the y-axis (You should include labels for your axes and a title for the plot)

2. Obtain the least squares regression line of exam grades on weekly spent time on self study. Interpret your model result (Using the whole data set)

3. Fit the linear model after partitioning your data set into training and testing (round the number of observations when it is necessary). After fitting the model, compare your parameter estimates with the model result in Question 2. Then, make predictions on testing data and compare with the original observations.

4. Using the `plot` command, comment on the validity of the assumption of the model that you fit in Question 3 (Note before using the `plot` command you may wish to specify a 2x2 graphics window using `par(mfrow = c(2, 2))`).

5. Calculate a 95% confidence interval for the slope regression parameter for the last model you fit in Question 3. (Note that the number of degrees of freedom should be obtained from the `R` output). For this you can use a built-in function in R

2 Part 1: Solution

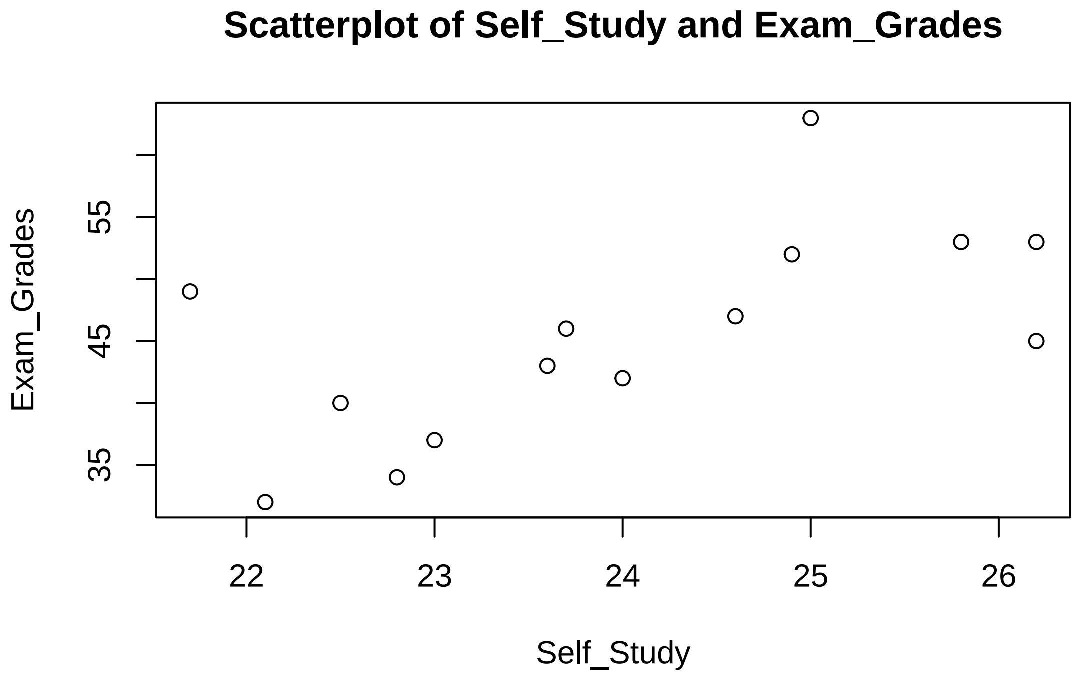
Use the given R-code chunk below to make your calculations and summarize your result thereafter by adding comments on it,

- MAKE SURE THAT ALL NECESSARY PACKAGES ARE ALREADY INSTALLED and READY TO USE
- You can use as many as Rcode chunks you want. In the final output, both Rcodes and your ouputs including your comments should appear in an order

```
# FOR REPRODUCIBILITY
set.seed(86002)
# ALERT: YOU NEED TO USE YOUR STUDENT NUMBER LAST 5 DIGITS
# HERE instead of 442 MAKE SURE THAT YOU CHANGED
# BEFORE STARTING TO YOUR ANALYSIS
```

This is an example of simple linear regression. I only have one independent variable and one dependent variable.Independent variable is self_study and dependent variable is exam grade.Self Study and Exam grades are my observations in vectors. I created a data frame with these datas.

```
# 1. Create a data frame in `R` with the above data. Plot the data with the weekly
#spent time on self study in the $x$-axis and exam grades on the $y$-axis
#(You should include labels for your axes and a title for the plot)

#I am trying to explain exam grades with respect to self study that's why, self
#study is my predictor and exam grades are my response

Self_Study = c(25.0, 26.2, 24.9, 23.7, 22.8, 24.6, 23.6, 23.0, 22.5, 26.2, 25.8, 24.0, 22.1, 21.7)

Exam_Grades = c(63, 53, 52, 46, 34, 47, 43, 37, 40, 45, 53, 42, 32, 49)

data_frame = data.frame(Self_Study,Exam_Grades,stringsAsFactors = FALSE)


#this function obtain a plot which contains y axis is exam grades and x axis is
#weekly spent time on self study and plot name is Scatterplot of Self_Study and
#Exam_Grades
plot(data_frame$Self_Study, data_frame$Exam_Grades,
     # Modify title and axis labels
     xlab = "Self_Study",
     ylab = "Exam_Grades",
     main="Scatterplot of Self_Study and Exam_Grades")
```



```
### FIT LSRL ####

#2. Obtain the least squares regression line of exam grades on weekly spent time
#on self study. Interpret your model result (Using the whole data set)


#first looking at the data_frame information
#with this str function we learned that 14 obs. of  2 variables and all of them
#numerical variables.
str(data_frame)
```

```
## 'data.frame':    14 obs. of  2 variables:
##  $ Self_Study : num  25 26.2 24.9 23.7 22.8 24.6 23.6 23 22.5 26.2 ...
##  $ Exam_Grades: num  63 53 52 46 34 47 43 37 40 45 ...
```

```
head(data_frame)
```

```
##   Self_Study Exam_Grades
## 1       25.0          63
## 2       26.2          53
## 3       24.9          52
## 4       23.7          46
## 5       22.8          34
## 6       24.6          47
```

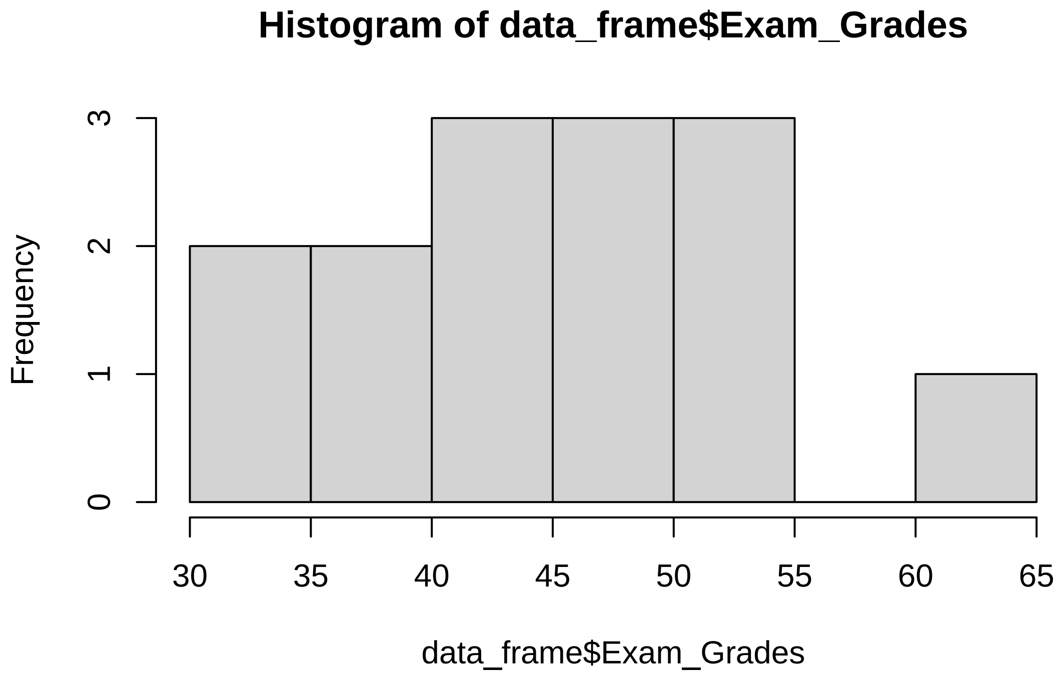
```
#looking for all info about data min,max,mean etc.
summary(data_frame)
```

```
##    Self_Study      Exam_Grades   
##  Min.   :21.70   Min.   :32.00  
##  1st Qu.:22.85   1st Qu.:40.50  
##  Median :23.85   Median :45.50  
##  Mean   :24.01   Mean   :45.43  
##  3rd Qu.:24.98   3rd Qu.:51.25  
##  Max.   :26.20   Max.   :63.00
```

```
# cor function is the correlation coefficient, and i see that higher values
#obviously
cor(data_frame[, c("Self_Study","Exam_Grades")])
```

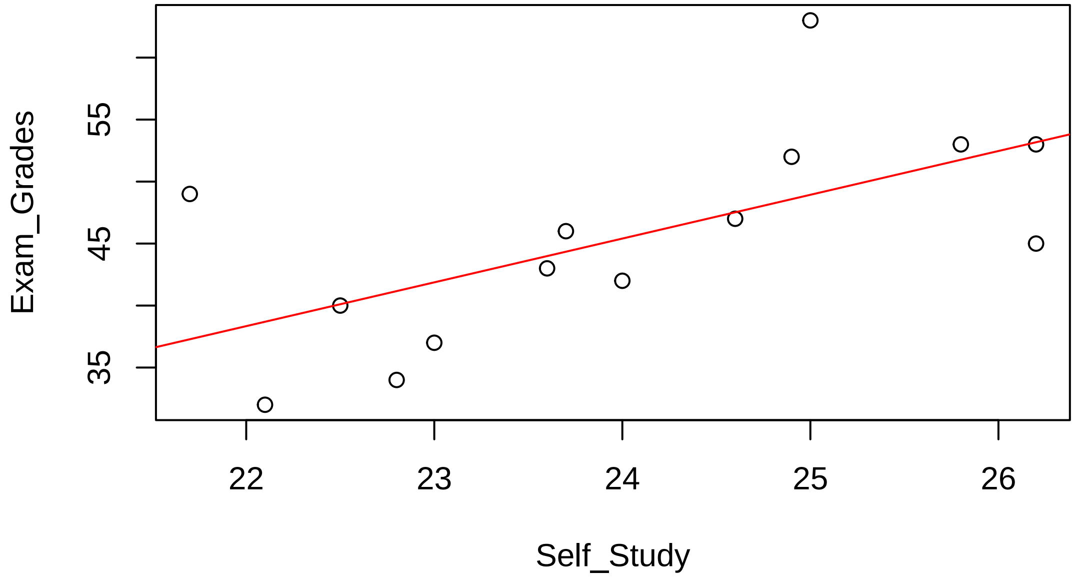
```
##             Self_Study Exam_Grades
## Self_Study   1.0000000   0.6291273
## Exam_Grades  0.6291273   1.0000000
```

```
#to draw a histogram
hist(data_frame$Exam_Grades)
```



```
#----------------------------------------

#since lm function extract exam grades and self study from original data set,
#we don't need to write like this data_frame$Exam_Grades. We assigned output of
#the lm function to an object.
lm.fit <- lm(formula = Exam_Grades ~ Self_Study, data = data_frame)


# To understand the relationship between two variables, its plot of the data like
```

```
#above
#by looking at the plot, we can say that there is a kind of linear relationship
plot(data_frame[, c("Self_Study","Exam_Grades")])

#same scatterplot with the previous, but with the least-squares regression line
#"fit" to the data to describe exam grades by self study
abline(lm.fit, col="red")
```



```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Exam_Grades ~ Self_Study, data = data_frame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1721 -4.5049 -0.3471  1.5521 14.0654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -39.347     30.291  -1.299   0.2184
## Self_Study     3.531      1.259   2.804   0.0159 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.774 on 12 degrees of freedom
## Multiple R-squared:  0.3958, Adjusted R-squared:  0.3455
## F-statistic: 7.861 on 1 and 12 DF,  p-value: 0.01593
```

```
#if i want to make a prediction, my new value should be lying between this two
#output value 21.7 and 26.2
#range(data_frame$Self_Study)

#we can understand our estimation significant or not simply evaluating p values
#stars illustrates differenet significant levels if we find bigger p values we
#need to be suspicious for model reliability
#final p value is overall significant level of our linear model
```

I can say that; Explanatory- Predictor variable : Self_Study Response variable : Exam_Grades

I can use histogram function to determine and to check the dependent(Exam Grades) variable follows a normal distribution. I see that my histogram follows a roughly bell-shaped curve that's why, it is following the normal distribution. That's why i can go into progress with linear regression.

By looking at the plot we can say that the relationship between dependent(Exam Grades) and independent(Self Study) variable is Linear, Positive and Weak.

I obtain significant parameter estimate but at the same time r squared value (0.3958) smaller than trashold(widely considered as 0.5)

The slope explain that how much increase there will be in the y axis for one unit increase in x axis thats why in our example, for every additional self study, exam grade will be 3.531 more points In our example value of y intercept i mean, for 0 hour self study , the exam grade should be -39.347, it does not make sense.

If I look at the p value the corresponding p-value is 0.0159, which is statistically significant at an alpha level of 0.05.

This tells us that that the average change in exam score for each additional weekly spent time is statistically significantly different than zero.

```
#3. Fit the linear model after partitioning your data set into training and
#testing (round the number of observations when it is necessary). After fitting
#the model, compare your parameter estimates with the model result in Question
#2. Then, make predictions on testing data and compare with the original
#observations.


#80% (11) should go the training subset,and rest should belong to the testing subset
sample.size <- floor(0.80 * nrow(data_frame)) # %80 for training, %20 for testing

#we are sampling from the whole set of data to create the index for training
train.index <- sample(seq_len(nrow(data_frame)), size = sample.size)

# Partitioning on training and testing
#in train.index, we are picking up  some of the observations for trainig set
train <- data_frame[train.index, ]
#it is just dropping spesific rows , this is our testing data
test <- data_frame[-train.index, ]

#dimension of the train and test data
dim(train)
```

```
## [1] 11  2
```

```
dim(test)
```

```
## [1] 3 2
```

```
# MODEL BUILDING
# Simple linear regression
#estimations are diffrent but they are really close to each other,
#according to which observations picking up  for the training set,the model
#output will be change a little bit
lm.fit = lm(formula = Exam_Grades ~ Self_Study, data = train)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Exam_Grades ~ Self_Study, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.881 -4.730 -1.124  1.901 14.110
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -28.776     36.037  -0.799   0.4451
## Self_Study     3.107      1.500   2.071   0.0682 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.371 on 9 degrees of freedom
## Multiple R-squared:  0.3228, Adjusted R-squared:  0.2476
## F-statistic: 4.291 on 1 and 9 DF,  p-value: 0.06821
```

```
# PREDICTION : Use the testing data
class(lm.fit)
```

```
## [1] "lm"
```

```
# Make prediction on training data because we obtain lm on training data to
#lm fit
predict(lm.fit)
```

```
##        6        2       12       14       10       13        9        1
## 47.64775 52.61840 45.78376 38.63845 52.61840 39.88112 41.12378 48.89041
##        3        7        8
## 48.57975 44.54110 42.67710
```

```
# Make predictions on testing only, newdata set  (testing subset playing a role
#new observation for the fitted model)
predict_data_frame <- predict(lm.fit, newdata = test)
#the results are my predictions
head(predict_data_frame)
```

```
##        4        5       11
## 44.85176 42.05577 51.37573
```

```
#these results are true observations under the test data
head(test$Exam_Grades)
```

```
## [1] 46 34 53
```

```
# Looking at RMSE basically
# difference between original and predicted values
MSE_fit <- mean((test$Exam_Grades - predict_data_frame)^2)
MSE_fit
```

```
## [1] 22.95072
```

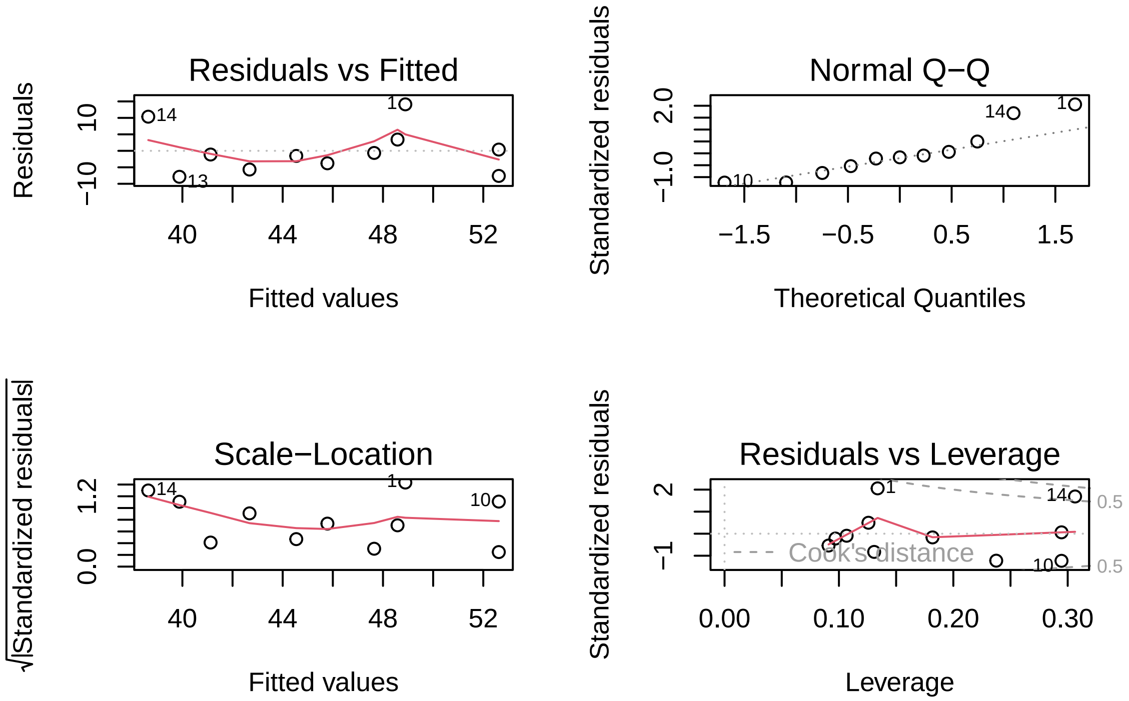
```
#error rate by the sqrt of MSE
RMSE_fit <- sqrt(MSE_fit)
RMSE_fit
```

```
## [1] 4.790691
```

```
cor(predict_data_frame, test$Exam_Grades)^2
```

```
## [1] 0.8639697
```

```
#4. Using the `plot` command, comment on the validity of the assumption of the
#model that you fit in Question 3 (Note before using the `plot` command
#you may wish to specify a 2x2 graphics window using `par(mfrow = c(2, 2))`).

#partition of the window 2 row and 2 column
par(mfrow = c(2, 2))
#creates for different plots(residuals,Q-Q etc.)
plot(lm.fit)
```



residuals should following normal distibution, red line should be approximately horizontal at zero. in our example, the linear assumption is not met, because the red line is not flatting on the zero at some points.

normal Q-Q plot, realize that upper tail, there is a kind of violation when we compare with the lower tail from the normal assumption. The points should be follow the diagonal line.

scale location,we need to see almost horizantal line, but there is no horizontal line in our example

residuals vs leverage, extreme values that might influence the regression results coming from the observation which could be out of the range.

```
#5. Calculate a $95\%$ confidence interval for the slope regression parameter
#for the last model you fit in Question 3. (Note that the number of degrees of
#freedom should be obtained from the `R` output). For this you can use a built-in function in R

confint(lm.fit, level=0.95)
```

```
##                     2.5 %    97.5 %
## (Intercept) -110.2967055 52.744846
## Self_Study    -0.2861493  6.499457
```

I made calculation because of the confidence interval with using confint() function.

3 Part 2: Logistic Regression

Consider the available example data set below

```r
# install.packages("mlbench")
library(mlbench)
data(BreastCancer)

summary(BreastCancer)
```

```
##       Id             Cl.thickness   Cell.size     Cell.shape  Marg.adhesion
##  Length:699         1      :145   1      :384   1      :353   1      :407
##  Class :character   5      :130   10     : 67   2      : 59   2      : 58
##  Mode  :character   3      :108   3      : 52   10     : 58   3      : 58
##                     4      : 80   2      : 45   3      : 56   10     : 55
##                     10     : 69   4      : 40   4      : 44   4      : 33
##                     2      : 50   5      : 30   5      : 34   8      : 25
##                     (Other):117   (Other): 81   (Other): 95   (Other): 63
##   Epith.c.size  Bare.nuclei   Bl.cromatin  Normal.nucleoli    Mitoses
##  2      :386   1      :402   2      :166   1      :443     1      :579
##  3      : 72   10     :132   3      :165   10     : 61     2      : 35
##  4      : 48   2      : 30   1      :152   3      : 44     3      : 33
##  1      : 47   5      : 30   7      : 73   2      : 36     10     : 14
##  6      : 41   3      : 28   4      : 40   8      : 24     4      : 12
##  5      : 39   (Other): 61   5      : 34   6      : 22     7      :  9
##  (Other): 66   NA’s   : 16   (Other): 69   (Other): 69     (Other): 17
##        Class
##  benign   :458
##  malignant:241
##
##
##
##
##
```

```r
# You can check the details here
# https://www.rdocumentation.org/packages/mlbench/versions/2.1-3/topics/BreastCancer
```

1. Convert your Class variable into a numerical one since you have two classes (benign malignant) you can make it one of them as 0 and the other one is 1

2. Fit a logistic regression model to classify **Class** using Mitoses (DO NOT FORGET TO PARTITION YOUR DATA INTO TRAINING AND TESTING DATA SETS, DO NOT FORGET THAT THIS DATA SET INCLUDES QUALITATIVE PREDICTORS !)

3. Make predictions and compare with the true observations (using TEST DATA SET). Calculate and intepret the Confusion Matrix results

4. Fit a multiple logistic regression to classify **Class** by using more than one predictor

5. Compare simple logistic and multiple logistic regression models using F1-score to make a decision on the best model. Why the overall accuracy is not enough as a performance measure ? Explain shortly

4 Part 2: Solution

Use the given R-code chunk below to make your calculations and summarize your result thereafter by adding comments on it,

- MAKE SURE THAT ALL NECESSARY PACKAGES ARE ALREADY INSTALLED and READY TO USE
- You can use as many as Rcode chunks you want. In the final output, both Rcodes and your ouputs including your comments should appear in an order

```
# 1. Convert your Class variable into a numerical one since you have two classes
#(benign malignant) you can make it one of them as 0 and the other one is 1

#provides access to the levels attribute of a variable. The first form
#returns the value of the levels of its argument and the second sets the attribute.
levels(BreastCancer$Class) <- c(1,0)
# Benign = 1, malignant = 0

head(BreastCancer)
```

```
##        Id Cl.thickness Cell.size Cell.shape Marg.adhesion Epith.c.size
## 1 1000025            5         1          1             1            2
## 2 1002945            5         4          4             5            7
## 3 1015425            3         1          1             1            2
## 4 1016277            6         8          8             1            3
## 5 1017023            4         1          1             3            2
## 6 1017122            8        10         10             8            7
##   Bare.nuclei Bl.cromatin Normal.nucleoli Mitoses Class
## 1           1           3               1       1     1
## 2          10           3               2       1     1
## 3           2           3               1       1     1
## 4           4           3               7       1     1
## 5           1           3               1       1     1
## 6          10           9               7       1     0
```

```
#2. Fit a logistic regression model to classify **Class** using Mitoses
#(DO NOT FORGET TO PARTITION YOUR DATA INTO TRAINING AND TESTING DATA SETS,
#DO NOT FORGET THAT THIS DATA SET INCLUDES QUALITATIVE PREDICTORS !)



#BreastCancer data is divided into 80% training and %20 testing test
BC_idx = sample(nrow(BreastCancer), 0.8 * nrow(BreastCancer))
BC_idx
```

```
##   [1] 344 569 116 323 548 620 298 247 276  21 500 310 644 544 381 649  82 462
##  [19] 508 632 425 492 306 250 483 346  62 592 546  37 246 579  25  74 422 426
##  [37] 209 146 605 175 312 289 121 223 668 615 562 391 300 497  95 641 533 299
##  [55] 607 263 371 586 361 637 144 390 315 160 114 215 105 573 545 156 241 695
##  [73] 659 434 527 321  52 179 172 196 433 699 185 688 355  92 374 260   1 651
##  [91] 583 550 255  12 507  63 643 176 161 506  84 394 461 317  27 353  28 225
## [109] 265 537  76 471  42 481 290 676 259 151 538 623 603 597 254 138 127  48
## [127] 357 613 657 582 574 625 320 305 530 365  59 480 327 362 558  50 543 681
## [145] 516  20 359  73  44 239 294 413 354 661 399 677 631  99 379 512   3 629
## [163] 140 427   4 624 193 585 325 571 283 221 229  89 256 324 514 133 147 228
## [181] 521  11 351 490 118 403 232 616 565 667 698 630 477 262  98 614   9 285
## [199] 153 447 329 380 377 135 369  94 224 588 684 648 555 693 337 577 108  33
## [217] 197 281  81 611 419 430 301  17 142 318 261   5 376 384 626 206 101 314
## [235] 103  88 278 566 458 457 358 656  69 602 293 416 204 165 671 309 404  49
## [253] 600  43 389 463  23 452 326 552 257  77 222  71 528 154  78 373 249 322
## [271] 438 535 511 547 441 169  60 198 599 291  91 627  58  30 557 542 450 331
## [289] 234 409 145  19 188 364 560  14 431 387  55 345 167  24 271 159 284 183
## [307] 464 386 303 126 102 446 155 124 363 406 494 608  35 171 488 330 594 540
## [325] 243 645 467 690 474 604 370 692 435 479  32   8 650 439 418 578 157 664
## [343] 532 487 675  72 589 130 174 429 338 628 352 258 218 691 235 414 549 541
## [361] 679 420 136 368 113 273 639  67 472 240 696 590  10 453 287 270 132  79
## [379] 208 411 634 640 187  56 612 570 148 189 694 407  61 307 396  75 125 601
## [397]  34 181 192 217  93 173  66 498 499  45 360 596 493 553 534 504 375 635
## [415] 233  80 385 476 559 697 280  57  54 591 646 670  47 115  36 170 436 367
## [433] 264 470 466 448 397  97 106 282 595 621 509 496 412 522 442 475 272 199
## [451] 266 598  90 205 680 201 214  40 203 654 268 408 109 686 636 139 478 485
## [469] 687 348 339 660 120 421 662 444 401 580 182 683 207 513  39 575 316 304
## [487] 520 678 652 469  83 149 428 674 347 277 245 445 248 119  64 128 568 685
## [505] 190 100 526 539 158 515 231 366 343 489  26 212 402 655 486 134 162 342
## [523]  16 184 400 460 501 449 295  96 251 503 349 163 110 610 531 619 275 609
## [541] 104 213 456 350 334 141 296 123 180 529 459 536 111 658 505 617 689 415
## [559]   7
```

```r
#---------------------------
# Training data set
BC_train = BreastCancer[BC_idx, ]
# Testing data set
BC_test = BreastCancer[-BC_idx, ]

#install glm2 library
library("glm2")
#fitting logistic reg. model while using mitosis a predictor
BC_glm = glm(Class~Mitoses, data=BC_train, family="binomial")






summary(BC_glm)
```

```
##
## Call:
## glm(formula = Class ~ Mitoses, family = "binomial", data = BC_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5211  -0.7294  -0.7294   0.2918   1.7055
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.1884     0.1089 -10.912  < 2e-16 ***
## Mitoses2       2.5234     0.5143   4.907 9.27e-07 ***
## Mitoses3       4.3239     1.0273   4.209 2.57e-05 ***
## Mitoses4      18.7545  1318.7268   0.014  0.98865
## Mitoses5       2.5747     1.1233   2.292  0.02190 *
## Mitoses6      18.7545  2797.4419   0.007  0.99465
## Mitoses7      18.7545  1615.1039   0.012  0.99074
## Mitoses8       2.9802     1.0856   2.745  0.00605 **
## Mitoses10     18.7545  1192.8333   0.016  0.98746
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 716.60  on 558  degrees of freedom
## Residual deviance: 555.62  on 550  degrees of freedom
## AIC: 573.62
##
## Number of Fisher Scoring iterations: 16
```

I divided into data set with (%80 training and %20 test) two parts training and test. My predictor is Mitosis and i gained the data from the trainig data while using for my logistic regression model

```
#3. Make predictions and compare with the true observations (using TEST DATA SET).
#Calculate and intepret the Confusion Matrix results

#installing the necessery package and call it
library(caret)
```

```
## Zorunlu paket yükleniyor: ggplot2
```

```
## Zorunlu paket yükleniyor: lattice
```

```
#to create prediction
predicted_values <- predict(BC_glm,type="response",newdata = BC_test)
#to make classification
BC_glm_predictor = ifelse(predicted_values >= 0.5,1,0)

#creating confusion matrix
confusion_matrix_test = confusionMatrix(as.factor(BC_glm_predictor),as.factor(BC_test$Class))
```

```
## Warning in confusionMatrix.default(as.factor(BC_glm_predictor),
## as.factor(BC_test$Class)): Levels are not in the same order for reference and
## data. Refactoring data to match.
```

```
confusion_matrix_test
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  0
##          1  5 27
##          0 84 24
##
##                Accuracy : 0.2071
##                  95% CI : (0.1433, 0.2838)
##     No Information Rate : 0.6357
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : -0.3821
##
##  Mcnemar's Test P-Value : 1.065e-07
##
##             Sensitivity : 0.05618
##             Specificity : 0.47059
##          Pos Pred Value : 0.15625
##          Neg Pred Value : 0.22222
##              Prevalence : 0.63571
##          Detection Rate : 0.03571
##    Detection Prevalence : 0.22857
##       Balanced Accuracy : 0.26338
##
##        'Positive' Class : 1
##
```

True positive (TP) (in our model it is 5 times) is the number of true results when the actual observation is positive.

False positive (FP) (in our model it is 27 times) is the number of incorrect predictions when the actual observation is positive.

True negative (TN) (in our model it is 24 times) is the number of true predictions when the observation is negative.

False negative (FN) (in our model it is 84 times)is the number of incorrect predictions when the observation is negative.

```
#4. Fit a multiple logistic regression to classify **Class** by using more than one predictor

BC_glm_multiple <- glm(Class~Cl.thickness +Epith.c.size, data=BC_train,family="binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
predicted_values_multiple <- predict(BC_glm_multiple,type="response",newdata=BC_test)
BC_glm_multiple_predictor = ifelse(predicted_values_multiple >= 0.5,1,0)

#creating confusion matrix
confusion_matrix_test_multiple = confusionMatrix(as.factor(BC_glm_multiple_predictor),as.factor(BC_test$
```

```
## Warning in confusionMatrix.default(as.factor(BC_glm_multiple_predictor), :
## Levels are not in the same order for reference and data. Refactoring data to
## match.
```

```
confusion_matrix_test_multiple
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  0
##          1  8 49
##          0 81  2
##
##                Accuracy : 0.0714
##                  95% CI : (0.0348, 0.1274)
##     No Information Rate : 0.6357
##     P-Value [Acc > NIR] : 1.00000
##
##                   Kappa : -0.768
##
##  Mcnemar's Test P-Value : 0.00655
##
##             Sensitivity : 0.08989
##             Specificity : 0.03922
##          Pos Pred Value : 0.14035
##          Neg Pred Value : 0.02410
##              Prevalence : 0.63571
##          Detection Rate : 0.05714
##    Detection Prevalence : 0.40714
##       Balanced Accuracy : 0.06455
##
##        'Positive' Class : 1
##
```

I took Cl.thickness and Epith.c.size as two predictors because of multiple predictors. Then also, I created the confusion matrix with these predictors and my newdata=BC_test. And I compare with accuracy of the one predictor and multiple predictor, i can say that accuracy level of simple logistic regression is better than multiple.

True positive (TP) (in our model it is 8 times) is the number of true results when the actual observation is positive.

False positive (FP) (in our model it is 81 times) is the number of incorrect predictions when the actual observation is positive.

True negative (TN) (in our model it is 2 times) is the number of true predictions when the observation is negative.

False negative (FN) (in our model it is 49 times)is the number of incorrect predictions when the observation is negative.

```
#5. Compare simple logistic and multiple logistic regression models using F1-score
#to make a decision on the best model. Why the overall accuracy is not enough as
#a performance measure ? Explain shortly


library(MLmetrics)
```

```
##
## Attaching package: 'MLmetrics'

## The following objects are masked from 'package:caret':
##
##     MAE, RMSE

## The following object is masked from 'package:base':
```

```
##
##      Recall
```

```
#f1 score calculations
F1_Score(as.factor(BC_glm_predictor),as.factor(BC_test$Class),positive ="0")
```

```
## [1] 0.3018868
```

```
F1_Score(as.factor(BC_glm_multiple_predictor),as.factor(BC_test$Class),positive="0")
```

```
## [1] 0.02985075
```

I have obtain less F1_score with multiple predictor than F1_score with one predictor.That's mean that model will obtain a low F1 score and it's both Precision and Recall are low. F1_score with one predictor are better than F1_score with multiple predictor. Because all of this reasons i can say in my data samples that simple logistic regression is better than the multiple logistic regression because our F1 score for multiple logistic regresssion is lower than simple logistic regression.

F1 Score value shows us the harmonic mean of (Precision) and (Recall) values.Our F1 scores were also low because both of our values were low.

We can't directly say accuracy is poor measure to evaluate. When the data is balanced accuracy is a good measure of evaluating our model. In other hand if data is imbalanced then accuracy is not a correct measure of evaluation.

Actually i can say that my observation and my data give a little bit bad value, i really get confused and force to comment on it. # References

Give a list of the available sources that you use while preparing your home-work (If you use other resources, you can make a list here for checking & reproducibility).

http://www.sthda.com/english/articles/39-regression-model-diagnostics/161-linear-regression-assumptions-and-diagnostics-in-r-essentials/

https://rstudio-pubs-static.s3.amazonaws.com/199692_d02c8f7b352e4ec1b85544432ac28896.html

https://medium.com/@KrishnaRaj_Parthasarathy/ml-classification-why-accuracy-is-not-a-best-measure-for-assessing-ceeb964ae47c

https://medium.com/@gulcanogundur/do%C4%9Fruluk-accuracy-kesinlik-precision-duyarl%C4%B1l%C4%B1k-recall-ya-da-f1-score-300c925feb38

https://www.turing.com/kb/how-to-plot-confusion-matrix