HarvardX: PH125.9x Data Science
IDV Learners Capstone Project

Cancer treatment survival analisis and prediction project

Denis Korolskii

November 23, 2019

Overview

IDV Learners Capstone Project of the HarvardX: PH125.9x Data Science: Capstone course. Current task is to create prediction system using a choosen dataset. Also, current task is to train a machine learning algorithm using the inputs in one subset to predict survival time in the test set.

Introduction

For this project we will focus on create a hypothesis testing and training prediction model system using Cameron and Pauling investigation dataset "Intravenous vitamin C in the supportive care of cancer patients: a review and rational approach", https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5927785/. Literature demonstrates that cancer patients experience vitamin C deficiency correlated with reduced oral intake, inflammation, infection, disease processes, and treatments such as radiation, chemotherapy, and surgery. Reaserch of the statistical significance of a possible difference in the effect of the Ascorbate treatment, as well as training the machine learning algorithm on the basis of the data obtained are the goals of this project.

Executive summary

The evaluation of algorithm performance is the Root Mean Square Error. RMSE is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSE represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences. These deviations are called residuals when the calculations are performed over the data sample that was used for estimation and are called errors (or prediction errors) when computed out-of-sample. The RMSE serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. RMSE is a measure of accuracy, to compare forecasting errors of different models for a particular dataset and not between datasets, as it is scale-dependent.

The function that computes the RMSE for vectors of ratings and their corresponding predictors will be the following:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{y}_{u,i} - y_{u,i})^2}$$

Dataset

Data open source: http://tunedit.org/repo/DASL/CancerSurvival.arff

The cancer_1 dataset is uploaded at GitHub repository https://github.com/DKorolski/homework-0/raw/8d656ba0abb62a8a48e611f8d5a2cebe4250bc97/cancer_1.xlsx

```r
#Loading libraries
if(!require(tidyverse)) install.packages ("tidyverse", repos = "http://cran.us.r-project.org")
if(!require(caret)) install.packages ("caret", repos = "http://cran.us.r-project.org")
if(!require(data.table)) install.packages ("data.table", repos = "http://cran.us.r-project.org")
if(!require(car)) install.packages("car", repos = "http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
if(!require(readxl)) install.packages("readxl", repos = "http://cran.us.r-project.org")

#Loading dataset using webscrape method
library(rvest)
url <- "http://tunedit.org/repo/DASL/CancerSurvival.arff"
web <- read_html(url)
t <- html_nodes(web,"textarea")
desc <- html_text(t)
desc1 <- str_sub(desc,-852,-4)
desc5 <- gsub(",","\t",desc1)
desc6 <- str_replace_all(desc5,"'","")
desc7 <- read_tsv(desc6,col_names = c("Survival", "Organ"))
z <- data.frame(desc7)
str(z)
```

```
## 'data.frame':    64 obs. of  2 variables:
##  $ Survival: num  124 42 25 45 412 ...
##  $ Organ   : chr  "Stomach" "Stomach" "Stomach" "Stomach" ...
```

```r
# alternative way of loading dataset from file
#(github repository link is provided) in case the url is broken
#z <- read_excel (path = 'cancer_1.xlsx', sheet = 'cancer')
```

Metadata Reference: Cameron, E. and Pauling, L. (1978) Supplemental ascorbate in the supportive treatment of cancer: re-evaluation of prolongation of survival times in terminal human cancer. Proceedings of the National Academy of Science USA. Also found in: Manly, B.F.J. (1986) Multivariate Statistical Methods: A Primer, New York: Chapman & Hall, 11. Also found in: Hand, D.J., et al. (1994) A Handbook of Small Data Sets, London: Chapman & Hall, 255. Description: Patients with advanced cancers of the stomach, bronchus, colon, ovary or breast were treated with ascorbate. The purpose of the study was to determine if the survival times differ with respect to the organ affected by the cancer. Number of cases: 64 Variable Names: Survival: Survival time (in days) Organ: Organ affected by the cancer relation 'Survival' numeric 'Organ' {"Breast","Bronchus","Colon","Ovary","Stomach"}

Methods and Analysis

Data Analysis

To get familiar with the dataset, we find the first rows of "cancer_1" subset as below. The subset contain the two variables "Survival", "Organ". Each row represent a single case.

Preprocessing

Testing for any N/A

```
colSums(is.na(z))
```

```
## Survival    Organ
##        0        0
```

Exploring dataset

```
str(z)
```

```
## 'data.frame':    64 obs. of  2 variables:
##  $ Survival: num  124 42 25 45 412 ...
##  $ Organ   : chr  "Stomach" "Stomach" "Stomach" "Stomach" ...
```

```
summary(z)
```

```
##     Survival          Organ
##  Min.   :  20.0   Length:64
##  1st Qu.: 102.5   Class :character
##  Median : 265.5   Mode  :character
##  Mean   : 558.6
##  3rd Qu.: 721.0
##  Max.   :3808.0
```

Converting characters to factor

```
z$Organ <- factor(z$Organ)
```

```
table(z$Organ)
```

```
##
##   Breast Bronchus    Colon    Ovary  Stomach
##       11       17       17        6       13
```

```
table(z)
```

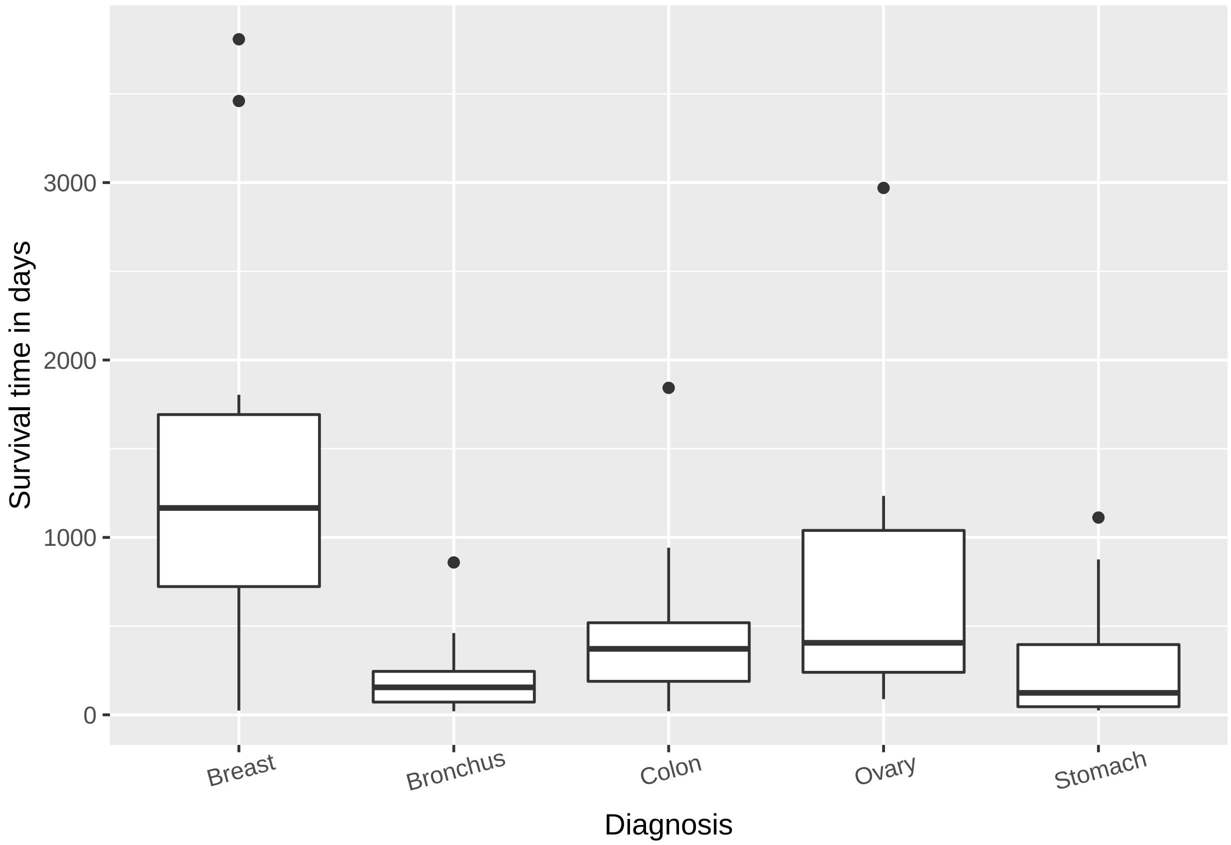
```
##         Organ
## Survival Breast Bronchus Colon Ovary Stomach
##     20        0        1     1     0       0
##     24        1        0     0     0       0
##     25        0        0     0     0       1
##     37        0        1     0     0       0
##     40        1        0     0     0       0
##     42        0        0     0     0       1
##     45        0        0     0     0       1
##     46        0        0     0     0       1
##     51        0        0     0     0       1
##     63        0        1     0     0       0
##     64        0        1     0     0       0
##     72        0        1     0     0       0
##     81        0        1     0     0       0
##     89        0        0     0     1       0
##     101       0        0     1     0       0
##     103       0        0     0     0       1
##     124       0        0     0     0       1
##     138       0        1     0     0       0
##     146       0        0     0     0       1
##     151       0        1     0     0       0
##     155       0        1     0     0       0
##     163       0        0     1     0       0
##     166       0        2     0     0       0
##     180       0        0     1     0       0
##     189       0        0     1     0       0
##     201       0        0     0     1       0
##     223       0        1     0     0       0
##     245       0        1     0     0       0
##     246       0        1     0     0       0
##     248       0        0     1     0       0
##     283       0        0     1     0       0
##     340       0        0     0     0       1
##     356       0        0     0     1       0
##     365       0        0     1     0       0
##     372       0        0     1     0       0
##     377       0        0     1     0       0
##     396       0        0     0     0       1
##     406       0        0     1     0       0
##     412       0        0     0     0       1
##     450       0        1     0     0       0
##     455       0        0     1     0       0
##     456       0        0     0     1       0
##     461       0        1     0     0       0
##     519       0        0     1     0       0
##     537       0        0     1     0       0
##     719       1        0     0     0       0
##     727       1        0     0     0       0
##     776       0        0     1     0       0
##     791       1        0     0     0       0
##     859       0        1     0     0       0
##     876       0        0     0     0       1
##     942       0        0     1     0       0
##     1112      0        0     0     0       1
##     1166      1        0     0     0       0
##     1234      0        0     0     1       0
##     1235      1        0     0     0       0
##     1581      1        0     0     0       0
##     1804      1        0     0     0       0
##     1843      0        0     1     0       0
##     2970      0        0     0     1       0
##     3460      1        0     0     0       0
##     3808      1        0     0     0       0
```

Dataset is small. Data is set of independent medical cases H0_hypothesis - survival time is equally dependent on organ (traditional) H1_hypothesis - survival time is not equally dependent on organ

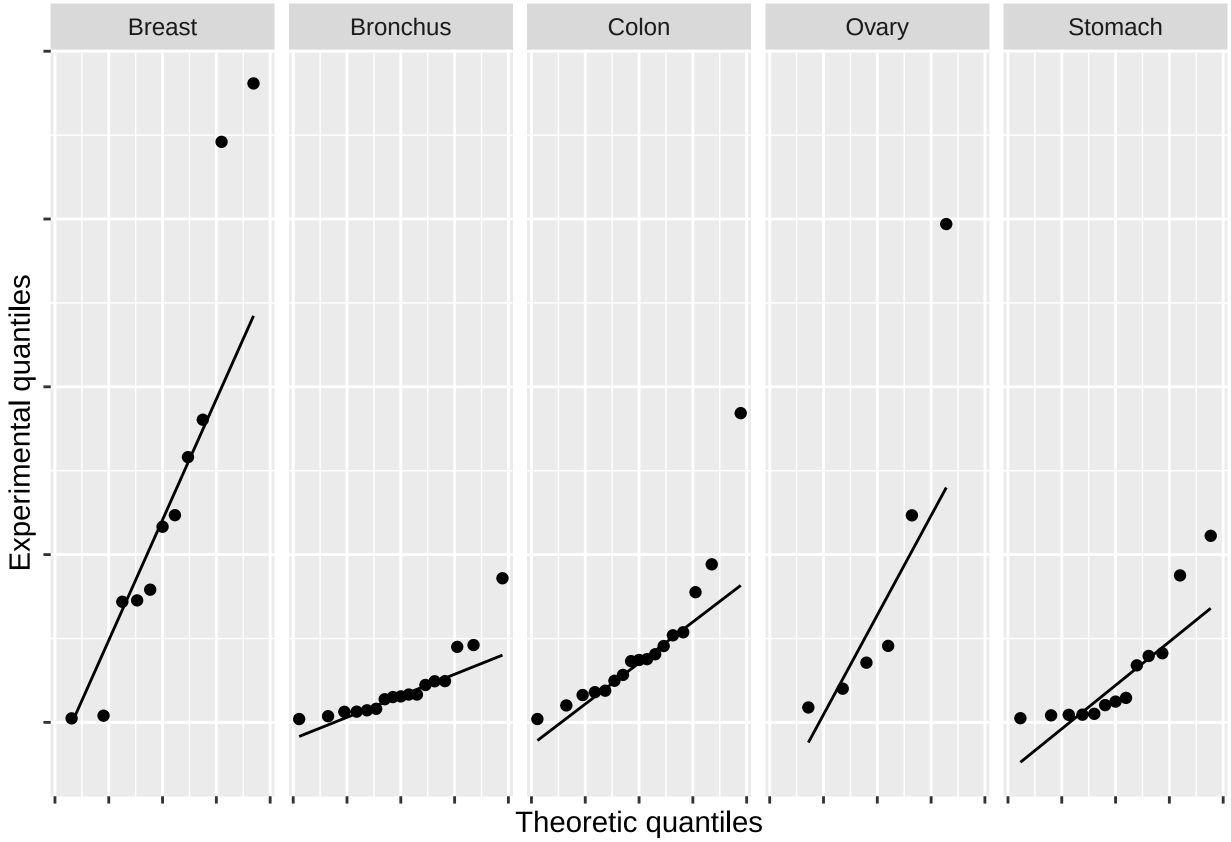
Plotting dataset distibution

The plotted boxplot shows the presence of extremely high survival rates in each group Since these outliers are uniformly distributed across all groups, it is more likely to conclude that these are not sample artifacts, but a strong sign of an asymmetric distribution. Their exclusion will lead to a distortion of the initial nature of the distribution, so it was decided to leave all the data in the array for further analysis.

```r
#check for normal distribution
ggplot(data = z, aes(x = Organ, y = Survival)) +
  geom_boxplot() +
  labs(x = "Diagnosis", y = "Survival time in days") +
  theme(axis.text.x = element_text(angle = 15, vjust = 0.9, hjust = 0.5))
```



```r
ggplot(aes(sample = Survival), data = z) +
  geom_qq() + geom_qq_line() +
  scale_x_continuous(labels = NULL, name = "Theoretic quantiles") +
  scale_y_continuous(labels = NULL, name = "Experimental quantiles") +
  facet_wrap(~Organ, ncol = 5)
```



Performing t-test

```r
#Performing t-test comparing selected organ (breast) with others
Stomach <- (z$Survival[z$Organ == "Stomach"])
Breast <- (z$Survival[z$Organ == "Breast"])
Bronchus <- (z$Survival[z$Organ == "Bronchus"])
Colon <- (z$Survival[z$Organ == "Colon"])
Ovary <- (z$Survival[z$Organ == "Ovary"])
t_St <- t.test(Breast, Stomach)
t_Br <- t.test(Breast, Bronchus)
t_C <- t.test(Breast, Colon)
t_O <- t.test(Breast, Ovary)
p_vals <- c(t_St$p.value, t_Br$p.value, t_C$p.value, t_O$p.value)
#Performing Holm adjustment
p_holm <- p.adjust(p_vals, method = 'holm')
sum(p_holm <= 0.05)
```

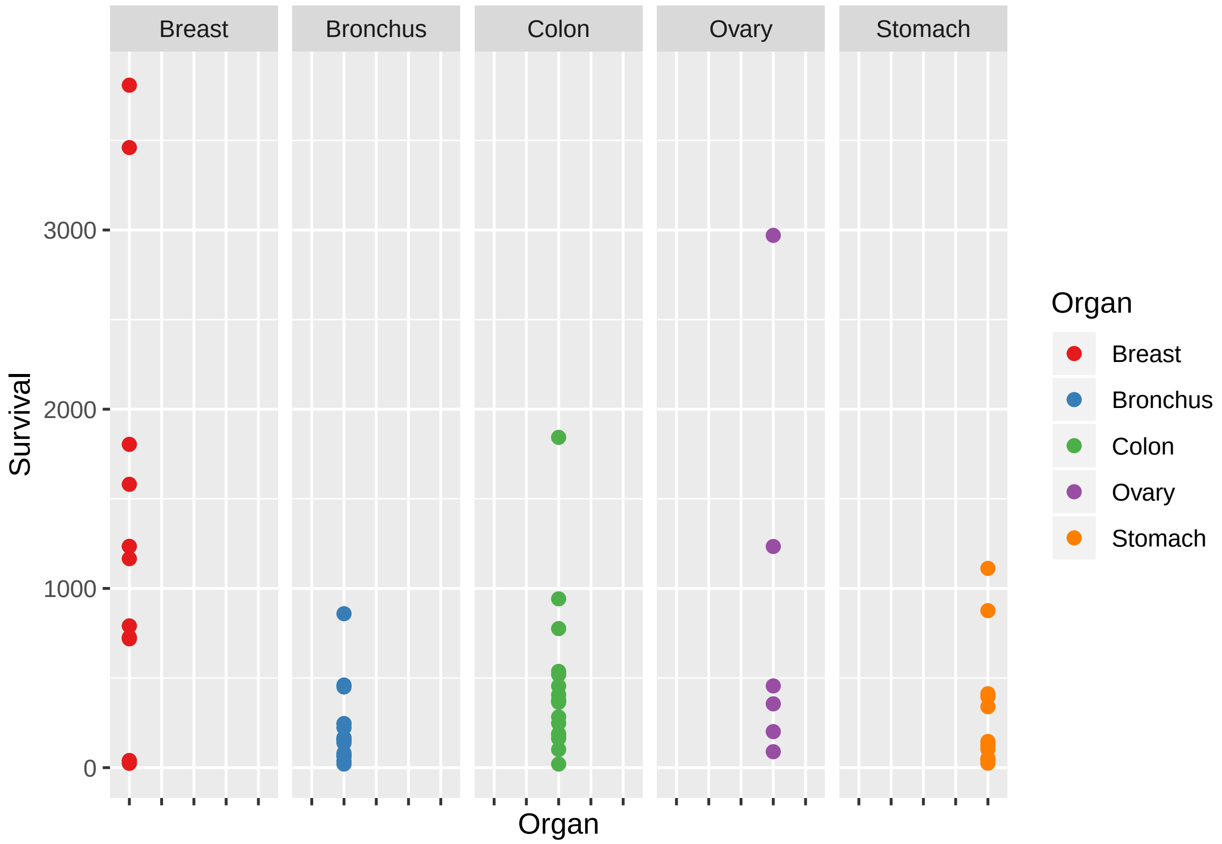
```
## [1] 2
```

```r
p_holm
```

```
## [1] 0.04391906 0.04018233 0.06595352 0.39862981
```

H1_hypothesis is correct for Breast-Stomach, Breast-Bronchus pairs

```
#plotting results
ggplot(z, aes(x = Organ, y = Survival, colour = Organ)) + geom_point(size = 2) + scale_colour_brewer( pa
```



Regression model

Preparing train and test datasets for small dataset with normal distribution

```
# Set seed
set.seed (42)
#making permutations
n_obs <- nrow(z)
permuted_rows<-sample(n_obs)

z_shuffled <- z[permuted_rows, ]

# Identify row to split on: split
split <- round(n_obs * 0.6)

# Create train
train <- z_shuffled[1:split, ]
str(train)
```

```
## 'data.frame':    38 obs. of  2 variables:
##  $ Survival: num  89 519 124 166 876 537 246 719 283 151 ...
##  $ Organ   : Factor w/ 5 levels "Breast","Bronchus",..: 4 3 5 2 5 3 2 1 3 2 ...
```

```
# Create test
test <- z_shuffled[(split+1):n_obs, ]
str(test)
```

```
## 'data.frame':    26 obs. of  2 variables:
##  $ Survival: num  45 776 72 40 396 406 163 42 859 180 ...
##  $ Organ   : Factor w/ 5 levels "Breast","Bronchus",..: 5 3 2 1 5 3 3 5 2 3 ...
```

1. Fit lm model on train: model

```
model <- lm (Survival ~ Organ , train)
head (model)
```

```
## $coefficients
##   (Intercept) OrganBronchus    OrganColon    OrganOvary  OrganStomach
##     1209.7143    -1021.2143     -637.0893     -325.3810     -818.1429
##
## $residuals
##          49          37           1          25          10          36
##  -795.33333   -53.62500  -267.57143   -22.50000   484.42857   -35.62500
##          18          64          47          24           7          59
##    57.50000  -490.71429  -289.62500   -37.50000   720.42857  -482.71429
##          61          63          46          20          26          56
##  -418.71429  2250.28571  -552.62500  -125.50000  -151.50000   371.28571
##           3          41          52          27          53          51
##  -366.57143   369.37500  2085.66667    34.50000  -428.33333  -528.33333
##          31           5          57          34          28          50
##  -324.62500    20.42857   -43.71429  1270.37500   -50.50000  -683.33333
##          33          55          30          11          15          22
##  -383.62500 -1185.71429    56.50000  -245.57143   272.50000   -33.50000
##          48           8
##   349.66667  -345.57143
##
## $effects
##   (Intercept) OrganBronchus    OrganColon    OrganOvary  OrganStomach
##   -3728.01048   -1533.48933    -605.08448     171.51444    1530.60513
##
##     -10.68558     152.39015    -234.68337    -264.68558      57.39015
##
##     718.53414    -226.68337    -162.68337    2506.31663    -527.68558
##
##     -30.60985     -56.60985     627.31663    -368.46586     394.31442
##
##    2248.05769     129.39015    -265.94231    -365.94231    -299.68558
##
##      18.53414     212.31663    1295.31442      44.39015    -520.94231
##
##    -358.68558    -929.68337     151.39015    -247.46586     367.39015
##
##      61.39015     512.05769    -347.46586
##
## $rank
## [1] 5
##
## $fitted.values
##        49        37         1        25        10        36        18 
##  884.3333  572.6250  391.5714  188.5000  391.5714  572.6250  188.5000 
##        64        47        24         7        59        61        63 
## 1209.7143  572.6250  188.5000  391.5714 1209.7143 1209.7143 1209.7143 
##        46        20        26        56         3        41        52 
##  572.6250  188.5000  188.5000 1209.7143  391.5714  572.6250  884.3333 
##        27        53        51        31         5        57        34 
##  188.5000  884.3333  884.3333  572.6250  391.5714 1209.7143  572.6250 
##        28        50        33        55        30        11        15 
##  188.5000  884.3333  572.6250 1209.7143  188.5000  391.5714  188.5000 
##        22        48         8 
##  188.5000  884.3333  391.5714 
##
## $assign
## [1] 0 1 1 1 1
```

```r
# Predict on test: p
p <- predict(model,test)
length(p)
```

```
## [1] 26
```

```r
# Compute errors: error
error <- p - test[["Survival"]]
length(p)
```

```
## [1] 26
```

```r
nrow(test)
```

```
## [1] 26
```

```r
length(error)
```

```
## [1] 26
```

```r
# Calculate RMSE

rmse1<-sqrt(mean(error^2))
rmse_result <- data_frame(method = "Linear regression model 1", RMSE = rmse1)
```

```
## Warning: `data_frame()` is deprecated, use `tibble()`.
## This warning is displayed once per session.
```

```r
rmse_result %>% knitr::kable()
```

method	RMSE
Linear regression model 1	627.9954

2.Cross-Validation. This method is a good choice when we have a minimum amount of data and we get sufficiently big difference in quality or different optimal parameters between folds. As a general rule, we choose k=5 or k=10, as these values have been shown empirically to yield test error estimates that suffer neither from excessively high bias nor high variance.

```
#CROSS-VALIDATION
# Fit lm model using 10-fold CV: model
model <- train(
  Survival~Organ ,
  z,
  method = "lm",
  trControl = trainControl(            #train-control func
    method = "cv",
    number = 10, #10-fold cross validation
    verboseIter = TRUE
  )
)
```

```
## + Fold01: intercept=TRUE
## - Fold01: intercept=TRUE
## + Fold02: intercept=TRUE
## - Fold02: intercept=TRUE
## + Fold03: intercept=TRUE
## - Fold03: intercept=TRUE
## + Fold04: intercept=TRUE
## - Fold04: intercept=TRUE
## + Fold05: intercept=TRUE
## - Fold05: intercept=TRUE
## + Fold06: intercept=TRUE
## - Fold06: intercept=TRUE
## + Fold07: intercept=TRUE
## - Fold07: intercept=TRUE
## + Fold08: intercept=TRUE
## - Fold08: intercept=TRUE
## + Fold09: intercept=TRUE
## - Fold09: intercept=TRUE
## + Fold10: intercept=TRUE
## - Fold10: intercept=TRUE
## Aggregating results
## Fitting final model on full training set
```

```
rmse2<-model$results$RMSE
rmse_result <- data_frame(method = "Linear regression+ 10 fold cross validation", RMSE = rmse2)
rmse_result %>% knitr::kable()
```

method	RMSE
Linear regression+ 10 fold cross validation	638.0567

```
# Fit lm model using 5 x 5-fold CV: model
model <- train(
  Survival~Organ,
  z,
  method = "lm",
  trControl = trainControl(
    method = "repeatedcv",
    number = 5,
    repeats = 5,
    verboseIter = TRUE
  )
)
```

```
## + Fold1.Rep1: intercept=TRUE
## - Fold1.Rep1: intercept=TRUE
## + Fold2.Rep1: intercept=TRUE
## - Fold2.Rep1: intercept=TRUE
## + Fold3.Rep1: intercept=TRUE
## - Fold3.Rep1: intercept=TRUE
## + Fold4.Rep1: intercept=TRUE
## - Fold4.Rep1: intercept=TRUE
## + Fold5.Rep1: intercept=TRUE
## - Fold5.Rep1: intercept=TRUE
## + Fold1.Rep2: intercept=TRUE
## - Fold1.Rep2: intercept=TRUE
## + Fold2.Rep2: intercept=TRUE
## - Fold2.Rep2: intercept=TRUE
## + Fold3.Rep2: intercept=TRUE
## - Fold3.Rep2: intercept=TRUE
## + Fold4.Rep2: intercept=TRUE
## - Fold4.Rep2: intercept=TRUE
## + Fold5.Rep2: intercept=TRUE
## - Fold5.Rep2: intercept=TRUE
## + Fold1.Rep3: intercept=TRUE
## - Fold1.Rep3: intercept=TRUE
## + Fold2.Rep3: intercept=TRUE
## - Fold2.Rep3: intercept=TRUE
## + Fold3.Rep3: intercept=TRUE
## - Fold3.Rep3: intercept=TRUE
## + Fold4.Rep3: intercept=TRUE
## - Fold4.Rep3: intercept=TRUE
## + Fold5.Rep3: intercept=TRUE
## - Fold5.Rep3: intercept=TRUE
## + Fold1.Rep4: intercept=TRUE
## - Fold1.Rep4: intercept=TRUE
## + Fold2.Rep4: intercept=TRUE
## - Fold2.Rep4: intercept=TRUE
## + Fold3.Rep4: intercept=TRUE
## - Fold3.Rep4: intercept=TRUE
## + Fold4.Rep4: intercept=TRUE
## - Fold4.Rep4: intercept=TRUE
## + Fold5.Rep4: intercept=TRUE
## - Fold5.Rep4: intercept=TRUE
## + Fold1.Rep5: intercept=TRUE
## - Fold1.Rep5: intercept=TRUE
## + Fold2.Rep5: intercept=TRUE
## - Fold2.Rep5: intercept=TRUE
## + Fold3.Rep5: intercept=TRUE
## - Fold3.Rep5: intercept=TRUE
## + Fold4.Rep5: intercept=TRUE
## - Fold4.Rep5: intercept=TRUE
## + Fold5.Rep5: intercept=TRUE
## - Fold5.Rep5: intercept=TRUE
## Aggregating results
## Fitting final model on full training set
```

```r
rmse3<-model$results$RMSE
rmse_result <- data_frame(method = "Linear regression+ 5x5 fold repeated cross validation", RMSE = rmse3
rmse_result %>% knitr::kable()
```

method	RMSE
Linear regression+ 5x5 fold repeated cross validation Calculating RMSE	668.4124

```r
model
```

```
## Linear Regression
##
## 64 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 52, 50, 51, 51, 52, 51, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   668.4124  0.2849913  430.9667
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
# Show the coefficients
mod <- lm(Survival~Organ, z)
coef(mod)
```

```
##   (Intercept) OrganBronchus    OrganColon    OrganOvary  OrganStomach
##     1395.9091    -1184.3209     -938.4973     -511.5758    -1109.9091
```

```r
# Show the full output
summary(mod)
```

```
##
## Call:
```

```
## lm(formula = Survival ~ Organ, data = z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1371.91  -241.75  -111.50    87.19  2412.09
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1395.9      201.9   6.915 3.77e-09 ***
## OrganBronchus  -1184.3      259.1  -4.571 2.53e-05 ***
## OrganColon      -938.5      259.1  -3.622 0.000608 ***
## OrganOvary      -511.6      339.8  -1.506 0.137526
## OrganStomach   -1109.9      274.3  -4.046 0.000153 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 669.5 on 59 degrees of freedom
## Multiple R-squared:  0.3037, Adjusted R-squared:  0.2565
## F-statistic: 6.433 on 4 and 59 DF,  p-value: 0.0002295
```

```
# View summary of model
summary(mod)
```

```
##
## Call:
## lm(formula = Survival ~ Organ, data = z)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1371.91  -241.75  -111.50    87.19  2412.09
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1395.9      201.9   6.915 3.77e-09 ***
## OrganBronchus  -1184.3      259.1  -4.571 2.53e-05 ***
## OrganColon      -938.5      259.1  -3.622 0.000608 ***
## OrganOvary      -511.6      339.8  -1.506 0.137526
## OrganStomach   -1109.9      274.3  -4.046 0.000153 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 669.5 on 59 degrees of freedom
## Multiple R-squared:  0.3037, Adjusted R-squared:  0.2565
## F-statistic: 6.433 on 4 and 59 DF,  p-value: 0.0002295
```

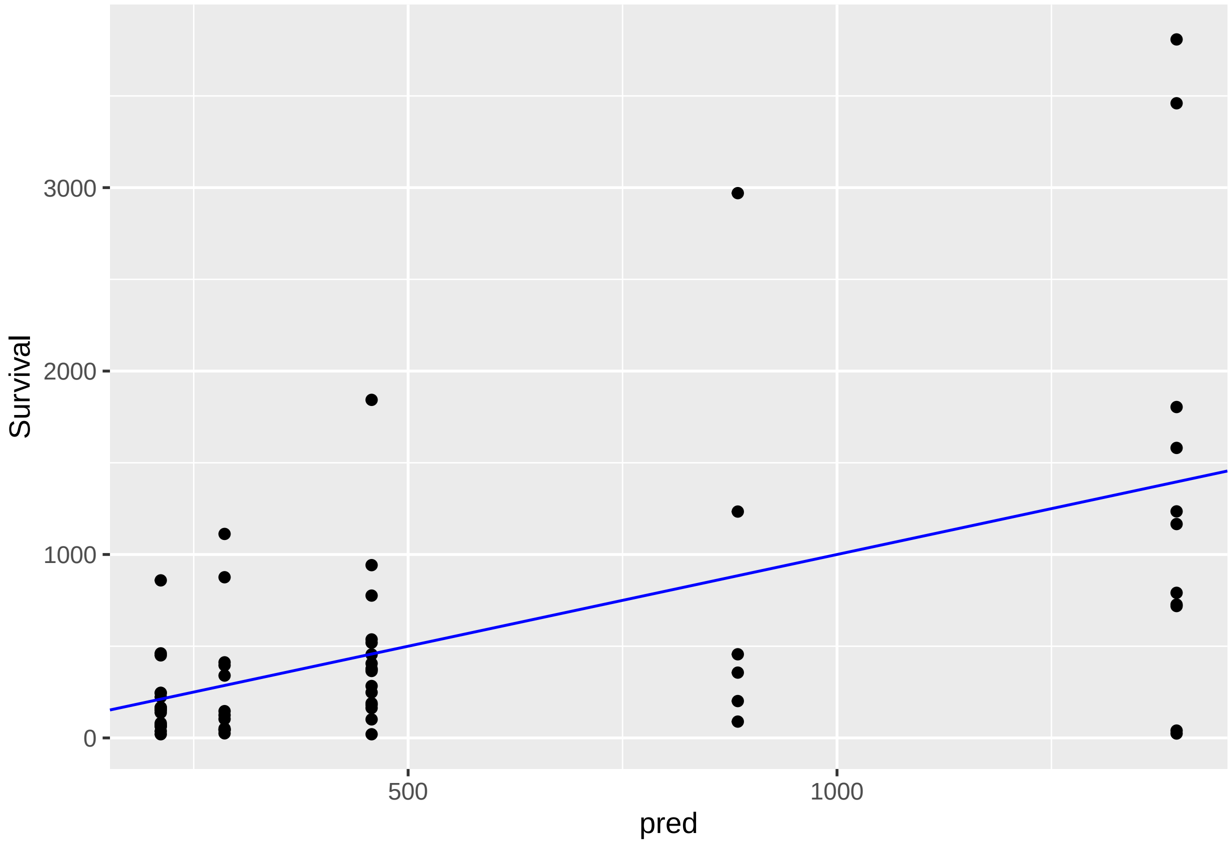
```
# Compute the mean of the residuals
mean(residuals(mod))
```

```
## [1] -2.4712e-14
```

```
# Compute RMSE
sqrt(sum(residuals(mod)^2) / df.residual(mod))
```

```
## [1] 669.5324
```

```
z$pred <- predict(model)
# Make a plot to compare predictions to actual (prediction on x axis).
ggplot(z, aes(x = pred, y = Survival)) +
  geom_point() +
  geom_abline(color = "blue")
```



3.Average+Organ effect system

```
mu <- mean(train$Survival)
survival_avgs <- train %>%
  group_by(Organ) %>%
  summarize(b_i = mean(Survival - mu))
# predicted ratings
predicted_ratings_bi <- mu + test %>%
  left_join(survival_avgs, by = "Organ") %>%
  .$b_i

rmse_4 <- RMSE(test$Survival, predicted_ratings_bi)
rmse_result <- data_frame(method = "Linear regression+ regular model", RMSE = rmse_4)
rmse_result %>% knitr::kable()
```

method	RMSE
Linear regression+ regular model	627.9954

#Results

method	RMSE
Linear regression model 1	627.9954
Linear regression+ 10 fold cross validation	638.0567
Linear regression+ 5x5 fold repeated cross validation	668.4124
Linear regression+ regular model	627.9954

#Conclusion.This IDV project was examined to observe data, check hypothesis and to predict survival time. The model evaluation performance through the RMSE (root mean squared error) showed that the Linear regression models are useful to predict survival time on the test set.
#Appendix - Enviroment

```
print("Operating System:")
```

```
## [1] "Operating System:"
```

```
version
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          6.1
## year           2019
## month          07
## day            05
## svn rev        76782
## language       R
## version.string R version 3.6.1 (2019-07-05)
## nickname       Action of the Toes
```