Lab 4 MATH 342W

Loyd Flores

11:59PM February 29

Create a dataset D which we call `Xy` such that the linear model has R^2 about 0% but x, y are clearly associated.

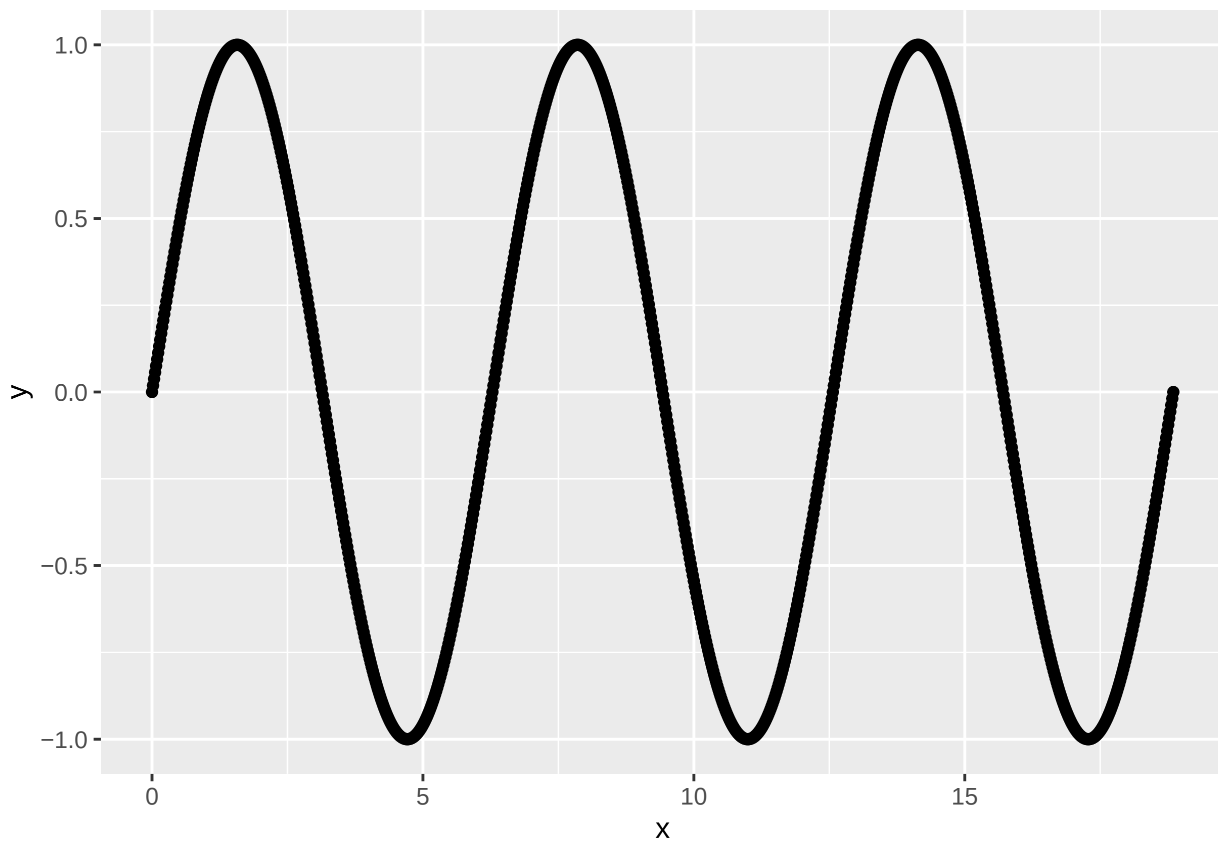
```
x = seq(0, 6 * pi, length.out=1000) # 1000 output
y = sin(x)

pacman::p_load(ggplot2)

#first check that Rsq is around zero
summary(lm(y ~ x))$r.squared
```

```
## [1] 0.06734109
```

```
#now check association visually
ggplot(data.frame(x = x, y = y)) + geom_point(aes(x = x, y = y))
```



Write a function `my_ols` that takes in `X`, a matrix with with p columns representing the feature measurements for each of the n units, a vector of n responses `y` and returns a list that contains the `b`, the p+1-sized column vector of OLS coefficients, `yhat` (the vector of n predictions), `e` (the vector of n residuals), `df` for degrees of freedom of the model, `SSE`, `SST`, `MSE`, `RMSE` and `Rsq` (for the R-squared metric). Internally, you cannot use `lm` or any other package; it must be done manually. You should throw errors if the inputs are non-numeric or not the same length. Or if `X` is not otherwise suitable. You should also name the class of the return value `my_ols` by using the `class` function as a setter. No need to create ROxygen documentation here.

df = degrees of freedom = p + 1 / number of dimensions / # of parameters

```
# X -> Columns / features
my_ols = function(X, y){
  # Step 1 concatenate 1 column
  X = cbind(1, X)

  # t(X) -> Transpose
  # Get best weights using OLS
  b = solve(t(X) %*% X) %*% t(X) %*% y

  # Get predictions
  y_hat = X %*% b

  # Get residuals
  e = y - y_hat

  # GET METRICS
  SSE = sum(e^2)
```

```
  SST = sum((y - mean(y))^2)

  df = ncol(X)
  n = nrow(X)
  MSE = SSE / (n - df)
  RMSE = sqrt(MSE)
  RSQ = (SST - SSE) / SST


  # RETURN
  lmobj = list(
    b = b,
    y_hat = y_hat,
    e = e,
    SSE = SSE,
    SST = SST,
    df = df,
    MSE = MSE,
    RSQ = RSQ
  )

  class(lmobj) = "my_ols"
  lmobj

}
```

Verify that the OLS coefficients for the `Type` of cars in the cars dataset gives you the same results as we did in class (i.e. the ybar's within group).

```
#TO-DO
cars = MASS::Cars93
colnames(cars)
```

```
##  [1] "Manufacturer"       "Model"              "Type"
##  [4] "Min.Price"          "Price"              "Max.Price"
##  [7] "MPG.city"           "MPG.highway"        "AirBags"
## [10] "DriveTrain"         "Cylinders"          "EngineSize"
## [13] "Horsepower"         "RPM"                "Rev.per.mile"
## [16] "Man.trans.avail"    "Fuel.tank.capacity" "Passengers"
## [19] "Length"             "Wheelbase"          "Width"
## [22] "Turn.circle"        "Rear.seat.room"     "Luggage.room"
## [25] "Weight"             "Origin"             "Make"
```

```
head(cars)
```

```
##   Manufacturer   Model    Type Min.Price Price Max.Price MPG.city MPG.highway
## 1        Acura Integra   Small      12.9  15.9      18.8       25          31
## 2        Acura  Legend Midsize      29.2  33.9      38.7       18          25
## 3         Audi      90 Compact      25.9  29.1      32.3       20          26
## 4         Audi     100 Midsize      30.8  37.7      44.6       19          26
## 5          BMW    535i Midsize      23.7  30.0      36.2       22          30
## 6        Buick Century Midsize      14.2  15.7      17.3       22          31
```

```
##              AirBags DriveTrain Cylinders EngineSize Horsepower  RPM
## 1               None      Front         4        1.8        140 6300
## 2 Driver & Passenger      Front         6        3.2        200 5500
## 3        Driver only      Front         6        2.8        172 5500
## 4 Driver & Passenger      Front         6        2.8        172 5500
## 5        Driver only       Rear         4        3.5        208 5700
## 6        Driver only      Front         4        2.2        110 5200
##   Rev.per.mile Man.trans.avail Fuel.tank.capacity Passengers Length Wheelbase
## 1         2890             Yes               13.2          5    177       102
## 2         2335             Yes               18.0          5    195       115
## 3         2280             Yes               16.9          5    180       102
## 4         2535             Yes               21.1          6    193       106
## 5         2545             Yes               21.1          4    186       109
## 6         2565              No               16.4          6    189       105
##   Width Turn.circle Rear.seat.room Luggage.room Weight  Origin           Make
## 1    68          37           26.5           11   2705 non-USA Acura Integra
## 2    71          38           30.0           15   3560 non-USA  Acura Legend
## 3    67          37           28.0           14   3375 non-USA       Audi 90
## 4    70          37           31.0           17   3405 non-USA      Audi 100
## 5    69          39           27.0           13   3640 non-USA      BMW 535i
## 6    69          41           28.0           16   2880     USA Buick Century
```

```r
# model.matrix => augments 1 and dummifies levels.
cars_X = model.matrix(~Type, cars)
head(cars_X)
```

```
##   (Intercept) TypeLarge TypeMidsize TypeSmall TypeSporty TypeVan
## 1           1         0           0         1          0       0
## 2           1         0           1         0          0       0
## 3           1         0           0         0          0       0
## 4           1         0           1         0          0       0
## 5           1         0           1         0          0       0
## 6           1         0           1         0          0       0
```

```r
# There are 6 types of cars, 1 became the intercept
head(cars_X)
```

```
##   (Intercept) TypeLarge TypeMidsize TypeSmall TypeSporty TypeVan
## 1           1         0           0         1          0       0
## 2           1         0           1         0          0       0
## 3           1         0           0         0          0       0
## 4           1         0           1         0          0       0
## 5           1         0           1         0          0       0
## 6           1         0           1         0          0       0
```

```r
# TRYING TO MODEL THE PRICE OF THE CAR FROM FEATURES
cars_y = cars$Price # Our label, y => Price, what we're guessing

# Initially it will fail because they the extra 1 column augmented makes the X not symmetrical
# -1 removes the intercept column or our 1 vector
my_ols(cars_X[, -1], cars_y)
```

```
## $b
##                   [,1]
##              18.212500
## TypeLarge     6.087500
## TypeMidsize   9.005682
## TypeSmall    -8.045833
## TypeSporty    1.180357
## TypeVan       0.887500
##
## $y_hat
##        [,1]
## 1  10.16667
## 2  27.21818
## 3  18.21250
## 4  27.21818
## 5  27.21818
## 6  27.21818
## 7  24.30000
## 8  24.30000
## 9  27.21818
## 10 24.30000
## 11 27.21818
## 12 18.21250
## 13 18.21250
## 14 19.39286
## 15 27.21818
## 16 19.10000
## 17 19.10000
## 18 24.30000
## 19 19.39286
## 20 24.30000
## 21 18.21250
## 22 24.30000
## 23 10.16667
## 24 10.16667
## 25 18.21250
## 26 19.10000
## 27 27.21818
## 28 19.39286
## 29 10.16667
## 30 24.30000
## 31 10.16667
## 32 10.16667
## 33 18.21250
## 34 19.39286
## 35 19.39286
## 36 19.10000
## 37 27.21818
## 38 24.30000
## 39 10.16667
## 40 19.39286
## 41 19.39286
## 42 10.16667
## 43 18.21250
```

```
## 44 10.16667
## 45 10.16667
## 46 19.39286
## 47 27.21818
## 48 27.21818
## 49 27.21818
## 50 27.21818
## 51 27.21818
## 52 24.30000
## 53 10.16667
## 54 10.16667
## 55 18.21250
## 56 19.10000
## 57 19.39286
## 58 18.21250
## 59 27.21818
## 60 19.39286
## 61 27.21818
## 62 10.16667
## 63 27.21818
## 64 10.16667
## 65 18.21250
## 66 19.10000
## 67 27.21818
## 68 18.21250
## 69 27.21818
## 70 19.10000
## 71 24.30000
## 72 19.39286
## 73 10.16667
## 74 18.21250
## 75 19.39286
## 76 27.21818
## 77 24.30000
## 78 18.21250
## 79 10.16667
## 80 10.16667
## 81 10.16667
## 82 18.21250
## 83 10.16667
## 84 10.16667
## 85 19.39286
## 86 27.21818
## 87 19.10000
## 88 10.16667
## 89 19.10000
## 90 18.21250
## 91 19.39286
## 92 18.21250
## 93 27.21818
##
## $e
##              [,1]
## 1   5.733333e+00
```

```
## 2   6.681818e+00
## 3   1.088750e+01
## 4   1.048182e+01
## 5   2.781818e+00
## 6  -1.151818e+01
## 7  -3.500000e+00
## 8  -6.000000e-01
## 9  -9.181818e-01
## 10  1.040000e+01
## 11  1.288182e+01
## 12 -4.812500e+00
## 13 -6.812500e+00
## 14 -4.292857e+00
## 15 -1.131818e+01
## 16 -2.800000e+00
## 17 -2.500000e+00
## 18 -5.500000e+00
## 19  1.860714e+01
## 20 -5.900000e+00
## 21 -2.412500e+00
## 22  5.200000e+00
## 23 -9.666667e-01
## 24  1.133333e+00
## 25 -4.912500e+00
## 26 -1.000000e-01
## 27 -1.161818e+01
## 28  6.407143e+00
## 29  2.033333e+00
## 30 -5.000000e+00
## 31 -2.766667e+00
## 32 -6.666667e-02
## 33 -6.912500e+00
## 34 -3.492857e+00
## 35 -5.392857e+00
## 36  8.000000e-01
## 37 -7.018182e+00
## 38 -3.400000e+00
## 39 -1.766667e+00
## 40 -6.892857e+00
## 41  4.071429e-01
## 42  1.933333e+00
## 43 -7.125000e-01
## 44 -2.166667e+00
## 45 -1.666667e-01
## 46 -9.392857e+00
## 47 -1.331818e+01
## 48  2.068182e+01
## 49  7.818182e-01
## 50  7.981818e+00
## 51  7.081818e+00
## 52  1.180000e+01
## 53 -1.866667e+00
## 54  1.433333e+00
## 55 -1.712500e+00
```

```
## 56 -3.552714e-15
## 57  1.310714e+01
## 58  1.368750e+01
## 59  3.468182e+01
## 60 -5.292857e+00
## 61 -1.231818e+01
## 62  1.333333e-01
## 63 -1.118182e+00
## 64  1.633333e+00
## 65 -2.512500e+00
## 66 -3.552714e-15
## 67 -5.718182e+00
## 68 -4.712500e+00
## 69 -1.091818e+01
## 70  4.000000e-01
## 71 -3.600000e+00
## 72 -4.992857e+00
## 73 -1.166667e+00
## 74 -7.112500e+00
## 75 -1.692857e+00
## 76 -8.718182e+00
## 77  1.000000e-01
## 78  1.048750e+01
## 79  9.333333e-01
## 80 -1.766667e+00
## 81  7.333333e-01
## 82  1.287500e+00
## 83 -1.566667e+00
## 84 -3.666667e-01
## 85 -9.928571e-01
## 86 -9.018182e+00
## 87  3.600000e+00
## 88 -1.066667e+00
## 89  6.000000e-01
## 90  1.787500e+00
## 91  3.907143e+00
## 92  4.487500e+00
## 93 -5.181818e-01
##
## $SSE
## [1] 5162.586
##
## $SST
## [1] 8584.021
##
## $df
## [1] 6
##
## $MSE
## [1] 59.34007
##
## $RSQ
## [1] 0.3985819
##
```

```
## attr(,"class")
## [1] "my_ols"
```

```
print("...")
```

```
## [1] "..."
```

Create a prediction method `g` that takes in a vector `x_star` and the dataset D i.e. `X` and `y` and returns the OLS predictions. Let `X` be a matrix with with p columns representing the feature measurements for each of the n units

```
g = function(x_star, X, y){
  #TO-DO
  # c1 to x_star so it will be of same size to b since b = length(p+1)
  #x_star is our unseen data / unseen features
  # ols weights * x_star == y_hat
  c(1,x_star) %*% my_ols(X,y)$b
}
```

Load up the famous iris dataset. We are going to do a different prediction problem. Imagine the only input x is Species and you are trying to predict y which is Petal.Length. A reasonable prediction is the average petal length within each Species. Prove that this is the OLS model by fitting an appropriate `lm` and then using the predict function to verify.

```
data(iris)
#TO-DO
# We're trying tho show that this gives y_
coef(lm(Petal.Length ~ Species, iris))
```

```
##       (Intercept) Speciesversicolor  Speciesvirginica
##             1.462             2.798             4.090
```

```
# We try to pull petal length for all species
mean(iris$Petal.Length[iris$Species == "setosa"]) # PULLS OUT ALL THE PETAL.LENGTHS OF SETOSA -> Then w
```

```
## [1] 1.462
```

```
mean(iris$Petal.Length[iris$Species == "versicolor"])
```

```
## [1] 4.26
```

```
mean(iris$Petal.Length[iris$Species == "virginica"])
```

```
## [1] 5.552
```

```
# 5.552 = coefficient of Virginica -> 1.462 + 4.090
```

Construct the design matrix with an intercept, X without using `model.matrix`.

```r
# # design matrix == x matrix
cbind(1, ifelse(iris$Species == "versicolor", 1, 0), ifelse(iris$Species == "virginica", 1, 0))
```

```
##        [,1] [,2] [,3]
##   [1,]    1    0    0
##   [2,]    1    0    0
##   [3,]    1    0    0
##   [4,]    1    0    0
##   [5,]    1    0    0
##   [6,]    1    0    0
##   [7,]    1    0    0
##   [8,]    1    0    0
##   [9,]    1    0    0
##  [10,]    1    0    0
##  [11,]    1    0    0
##  [12,]    1    0    0
##  [13,]    1    0    0
##  [14,]    1    0    0
##  [15,]    1    0    0
##  [16,]    1    0    0
##  [17,]    1    0    0
##  [18,]    1    0    0
##  [19,]    1    0    0
##  [20,]    1    0    0
##  [21,]    1    0    0
##  [22,]    1    0    0
##  [23,]    1    0    0
##  [24,]    1    0    0
##  [25,]    1    0    0
##  [26,]    1    0    0
##  [27,]    1    0    0
##  [28,]    1    0    0
##  [29,]    1    0    0
##  [30,]    1    0    0
##  [31,]    1    0    0
##  [32,]    1    0    0
##  [33,]    1    0    0
##  [34,]    1    0    0
##  [35,]    1    0    0
##  [36,]    1    0    0
##  [37,]    1    0    0
##  [38,]    1    0    0
##  [39,]    1    0    0
##  [40,]    1    0    0
##  [41,]    1    0    0
##  [42,]    1    0    0
##  [43,]    1    0    0
##  [44,]    1    0    0
##  [45,]    1    0    0
##  [46,]    1    0    0
##  [47,]    1    0    0
##  [48,]    1    0    0
##  [49,]    1    0    0
```

```
##  [50,]    1    0    0
##  [51,]    1    1    0
##  [52,]    1    1    0
##  [53,]    1    1    0
##  [54,]    1    1    0
##  [55,]    1    1    0
##  [56,]    1    1    0
##  [57,]    1    1    0
##  [58,]    1    1    0
##  [59,]    1    1    0
##  [60,]    1    1    0
##  [61,]    1    1    0
##  [62,]    1    1    0
##  [63,]    1    1    0
##  [64,]    1    1    0
##  [65,]    1    1    0
##  [66,]    1    1    0
##  [67,]    1    1    0
##  [68,]    1    1    0
##  [69,]    1    1    0
##  [70,]    1    1    0
##  [71,]    1    1    0
##  [72,]    1    1    0
##  [73,]    1    1    0
##  [74,]    1    1    0
##  [75,]    1    1    0
##  [76,]    1    1    0
##  [77,]    1    1    0
##  [78,]    1    1    0
##  [79,]    1    1    0
##  [80,]    1    1    0
##  [81,]    1    1    0
##  [82,]    1    1    0
##  [83,]    1    1    0
##  [84,]    1    1    0
##  [85,]    1    1    0
##  [86,]    1    1    0
##  [87,]    1    1    0
##  [88,]    1    1    0
##  [89,]    1    1    0
##  [90,]    1    1    0
##  [91,]    1    1    0
##  [92,]    1    1    0
##  [93,]    1    1    0
##  [94,]    1    1    0
##  [95,]    1    1    0
##  [96,]    1    1    0
##  [97,]    1    1    0
##  [98,]    1    1    0
##  [99,]    1    1    0
## [100,]    1    1    0
## [101,]    1    0    1
## [102,]    1    0    1
## [103,]    1    0    1
```

```
## [104,]    1    0    1
## [105,]    1    0    1
## [106,]    1    0    1
## [107,]    1    0    1
## [108,]    1    0    1
## [109,]    1    0    1
## [110,]    1    0    1
## [111,]    1    0    1
## [112,]    1    0    1
## [113,]    1    0    1
## [114,]    1    0    1
## [115,]    1    0    1
## [116,]    1    0    1
## [117,]    1    0    1
## [118,]    1    0    1
## [119,]    1    0    1
## [120,]    1    0    1
## [121,]    1    0    1
## [122,]    1    0    1
## [123,]    1    0    1
## [124,]    1    0    1
## [125,]    1    0    1
## [126,]    1    0    1
## [127,]    1    0    1
## [128,]    1    0    1
## [129,]    1    0    1
## [130,]    1    0    1
## [131,]    1    0    1
## [132,]    1    0    1
## [133,]    1    0    1
## [134,]    1    0    1
## [135,]    1    0    1
## [136,]    1    0    1
## [137,]    1    0    1
## [138,]    1    0    1
## [139,]    1    0    1
## [140,]    1    0    1
## [141,]    1    0    1
## [142,]    1    0    1
## [143,]    1    0    1
## [144,]    1    0    1
## [145,]    1    0    1
## [146,]    1    0    1
## [147,]    1    0    1
## [148,]    1    0    1
## [149,]    1    0    1
## [150,]    1    0    1
```

We now load the diamonds dataset. Skim the dataset using skimr or summary. What is the datatype of the color feature? : ORDERED FACTOR originially before we turn it into an integer after one-hot-encoding the different levels.

```r
rm(list = ls())
pacman::p_load(ggplot2, skim)
```

```
## Installing package into ’C:/Users/lenovo/AppData/Local/R/win-library/4.3’
## (as ’lib’ is unspecified)

## Warning: package ’skim’ is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

## Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib
##   cannot open URL ’http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.3/PACKAGES’

## Warning: ’BiocManager’ not available.  Could not check Bioconductor.
##
## Please use ‘install.packages(’BiocManager’)‘ and then retry.

## Warning in p_install(package, character.only = TRUE, ...):

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called ’skim’

## Warning in pacman::p_load(ggplot2, skim): Failed to install/load:
## skim
```

```r
pacman::p_load(skim)
```

```
## Installing package into ’C:/Users/lenovo/AppData/Local/R/win-library/4.3’
## (as ’lib’ is unspecified)

## Warning: package ’skim’ is not available for this version of R
##
## A version of this package for your version of R might be available elsewhere,
## see the ideas at
## https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages

## Warning: unable to access index for repository http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib
##   cannot open URL ’http://www.stats.ox.ac.uk/pub/RWin/bin/windows/contrib/4.3/PACKAGES’

## Warning: ’BiocManager’ not available.  Could not check Bioconductor.
##
## Please use ‘install.packages(’BiocManager’)‘ and then retry.

## Warning in p_install(package, character.only = TRUE, ...):

## Warning in library(package, lib.loc = lib.loc, character.only = TRUE,
## logical.return = TRUE, : there is no package called ’skim’

## Warning in pacman::p_load(skim): Failed to install/load:
## skim
```

```r
diamonds = ggplot2::diamonds
#TO-DO
summary(diamonds)
```

```
##      carat                 cut        color       clarity          depth      
##  Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065   Min.   :43.00  
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258   1st Qu.:61.00  
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median :61.80  
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171   Mean   :61.75  
##  3rd Qu.:1.0400   Ideal    :21551   H: 8304   VVS2   : 5066   3rd Qu.:62.50  
##  Max.   :5.0100                     I: 5422   VVS1   : 3655   Max.   :79.00  
##                                     J: 2808   (Other): 2531                  
##      table           price             x                y         
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000  
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720  
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710  
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735  
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540  
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900  
##                                                                   
##        z         
##  Min.   : 0.000  
##  1st Qu.: 2.910  
##  Median : 3.530  
##  Mean   : 3.539  
##  3rd Qu.: 4.040  
##  Max.   :31.800  
## 
```

```r
colnames(diamonds)
```

```
##  [1] "carat"   "cut"     "color"   "clarity" "depth"   "table"   "price"  
##  [8] "x"       "y"       "z"
```

```r
typeof(diamonds$color)
```

```
## [1] "integer"
```

Find the levels of the color feature.

```r
levels(diamonds$color)
```

```
## [1] "D" "E" "F" "G" "H" "I" "J"
```

```r
# Different entries in color feature
```

Create new feature in the diamonds dataset, `color_as_numeric`, which is color expressed as a continuous interval value.

```r
#TO-DO
# NUMERIC
# one hot encoding color
diamonds$color_as_numeric = as.numeric(diamonds$color)
head(diamonds)
```

```
## # A tibble: 6 x 11
##   carat cut   color clarity depth table price     x     y     z color_as_numeric
##   <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>            <dbl>
## 1  0.23 Ideal E     SI2      61.5    55   326  3.95  3.98  2.43                2
## 2  0.21 Prem~ E     SI1      59.8    61   326  3.89  3.84  2.31                2
## 3  0.23 Good  E     VS1      56.9    65   327  4.05  4.07  2.31                2
## 4  0.29 Prem~ I     VS2      62.4    58   334  4.2   4.23  2.63                6
## 5  0.31 Good  J     SI2      63.3    58   335  4.34  4.35  2.75                7
## 6  0.24 Very~ J     VVS2     62.8    57   336  3.94  3.96  2.48                7
```

```r
# NOMINAL
diamonds$color_as_nominal = factor(diamonds$color)
head(diamonds)
```

```
## # A tibble: 6 x 12
##   carat cut   color clarity depth table price     x     y     z color_as_numeric
##   <dbl> <ord> <ord> <ord>   <dbl> <dbl> <int> <dbl> <dbl> <dbl>            <dbl>
## 1  0.23 Ideal E     SI2      61.5    55   326  3.95  3.98  2.43                2
## 2  0.21 Prem~ E     SI1      59.8    61   326  3.89  3.84  2.31                2
## 3  0.23 Good  E     VS1      56.9    65   327  4.05  4.07  2.31                2
## 4  0.29 Prem~ I     VS2      62.4    58   334  4.2   4.23  2.63                6
## 5  0.31 Good  J     SI2      63.3    58   335  4.34  4.35  2.75                7
## 6  0.24 Very~ J     VVS2     62.8    57   336  3.94  3.96  2.48                7
## # i 1 more variable: color_as_nominal <ord>
```

Use that converted feature as the one predictor in a regression. How well does this regression do as measured by RMSE?

```r
#TO-DO
# Trying to fit a linear line using price as a function of color_as_numeric
# Different colors of diamonds cost different amounts
diamonds_coeff = lm(price ~ color_as_numeric, diamonds) # gets w_0, w_1
diamonds_coeff
```

```
##
## Call:
## lm(formula = price ~ color_as_numeric, data = diamonds)
##
## Coefficients:
##      (Intercept)  color_as_numeric
##           2478.7             404.6
```

```r
summary (diamonds_coeff)$sigma # RSQ
```

```
## [1] 3929.665
```

Create new feature in the diamonds dataset, `color_as_nominal`, which is color expressed as a nominal categorical variable.

```
#TO-DO
diamonds_coeff_nominal = lm(price ~ color_as_nominal, diamonds)
summary (diamonds_coeff_nominal)$sigma
```

```
## [1] 3926.777
```

Use that converted feature as the one predictor in a regression. How well does this regression do as measured by RMSE?

```
#TO-DO

# Assuming the linear model has been fit and is named diamonds_coeff
predicted_prices = predict(diamonds_coeff, newdata = diamonds)

# Calculate residuals (differences between actual and predicted prices)
residuals = diamonds$price - predicted_prices

# Calculate MSE and then RMSE
mse = mean(residuals^2)
rmse = sqrt(mse)

# Print RMSE
print(rmse)
```

```
## [1] 3929.592
```

```
# We are off 3,929.59 in regards to price.
```

Which regression does better - `color_as_numeric` or `color_as_nominal`? Why?

#TO-DO

Now regress both `color_as_numeric` and `color_as_nominal` in a regression. Does this regression do any better (as gauged by RMSE) than either color_as_numericorcolor_as_nominal‘ alone?

```
#TO-DO

numeric_nominal_model = lm(diamonds$price ~ diamonds$color_as_numeric + diamonds$color_as_nominal, newda
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'newdata' will be disregarded
```

```
numeric_nominal_predictions = predict(numeric_nominal_model, newdata = diamonds)
numeric_nominal_residuals = diamonds$price - numeric_nominal_predictions
nn_mse = mean(numeric_nominal_residuals^2)
nn_rmse = sqrt(nn_mse)
cat("Color as Nominal only RMSE:", rmse, "Both Numeric and Nominal RMSE:", nn_rmse, "\n")
```

```
## Color as Nominal only RMSE: 3929.592 Both Numeric and Nominal RMSE: 3926.522
```

What are the coefficients (the b vector)?

```
# EXTRACT AND Y
X = subset(diamonds, select = -c(color_as_nominal, price))
X = as.matrix(X)
X = cbind(1, X)
y = diamonds$price

# Compute the coefficients vector 'b_vector' using the OLS formula
#b_vector <- solve(t(X) %*% X) %*% t(X) %*% y

# Print the coefficients vector
#print(b_vector)
```

Something appears to be anomalous in the coefficients. What is it? Why? 1. Intercept is really high, meaning if everything is 0 the diamond will cost insane amounts? 2. x has a negative coefficient which does not make sense since x is a measurement of the diamond. it should be that the larger the x the larger the price

#TO-DO

Return to the iris dataset. Find the hat matrix H for this regression of diamond price on diamond color. Use only the first 1,00 observations in the diamond dataset.

```
rm(list = ls())
diamonds1000 = ggplot2::diamonds[1:1000,]
# WE NEED TO FIND THE X MATRIX => H = X(X^T X)^-1 X^T
X_diamonds = model.matrix(price ~ color, diamonds1000) #Regress price to color
H = X_diamonds %*% solve(t(X_diamonds) %*% X_diamonds) %*% t(X_diamonds)
```

Verify this hat matrix is symmetric using the `expect_equal` function in the package `testthat`.

```
#TO-DO
pacman::p_load(testthat)
expect_equal(H, t(H)) #Test for symmetry ; NO RESPONSE HENCE IT IS SYMMETRICAL
```

Verify this hat matrix is idempotent using the `expect_equal` function in the package `testthat`.

```
pacman::p_load(testthat)
#TO-DO
expect_equal(H %*% H, H) # NO RESPONSE MEANING IT IS IDEMPOTENT
```

Using the `diag` function, find the trace of the hat matrix.

```
#TO-DO
sum(diag(H))
```

```
## [1] 7
```

```
# Trace of an orthogonal matrix is its rank
```

It turns out the trace of a hat matrix is the same as its rank! But we don't have time to prove these interesting and useful facts..

Using the hat matrix, compute the yhat vector and using the projection onto the residual space, compute the e vector and verify they are orthogonal to each other.

```
#TO-DO

y_diamond = diamonds1000$price
yhat_diamond = H%*%y_diamond #
yhat_diamond
```

```
##          [,1]
## 1    2542.571
## 2    2542.571
## 3    2542.571
## 4    1947.695
## 5    1990.217
## 6    1990.217
## 7    1947.695
## 8    2307.808
## 9    2542.571
## 10   2307.808
## 11   1990.217
## 12   1990.217
## 13   2692.438
## 14   1990.217
## 15   2542.571
## 16   2542.571
## 17   1947.695
## 18   1990.217
## 19   1990.217
## 20   1990.217
## 21   1947.695
## 22   2542.571
## 23   2307.808
## 24   1990.217
## 25   1990.217
## 26   2576.259
## 27   1947.695
## 28   1990.217
## 29   2594.380
## 30   2692.438
## 31   2692.438
## 32   2692.438
## 33   2542.571
## 34   2542.571
## 35   2594.380
## 36   2692.438
## 37   2542.571
## 38   2307.808
## 39   2594.380
## 40   1947.695
## 41   1947.695
```

```
## 42   1990.217
## 43   2594.380
## 44   2594.380
## 45   2307.808
## 46   2692.438
## 47   2307.808
## 48   2307.808
## 49   2542.571
## 50   2307.808
## 51   2692.438
## 52   2576.259
## 53   1947.695
## 54   2542.571
## 55   2594.380
## 56   1947.695
## 57   1990.217
## 58   1947.695
## 59   1947.695
## 60   1947.695
## 61   1947.695
## 62   2594.380
## 63   2594.380
## 64   2594.380
## 65   1947.695
## 66   2576.259
## 67   1947.695
## 68   2576.259
## 69   2576.259
## 70   2542.571
## 71   2594.380
## 72   2307.808
## 73   2307.808
## 74   2307.808
## 75   2307.808
## 76   2692.438
## 77   2542.571
## 78   2594.380
## 79   2594.380
## 80   2542.571
## 81   2542.571
## 82   2594.380
## 83   2542.571
## 84   1947.695
## 85   2542.571
## 86   2576.259
## 87   2307.808
## 88   2307.808
## 89   2307.808
## 90   1947.695
## 91   2542.571
## 92   2542.571
## 93   2576.259
## 94   2542.571
## 95   2576.259
```

```
## 96   2542.571
## 97   2692.438
## 98   2692.438
## 99   2542.571
## 100  2307.808
## 101  2594.380
## 102  2542.571
## 103  2576.259
## 104  2576.259
## 105  1947.695
## 106  2576.259
## 107  2576.259
## 108  1947.695
## 109  2692.438
## 110  2542.571
## 111  2692.438
## 112  2542.571
## 113  1947.695
## 114  2576.259
## 115  2692.438
## 116  2692.438
## 117  2692.438
## 118  2576.259
## 119  2542.571
## 120  2692.438
## 121  2594.380
## 122  2542.571
## 123  2692.438
## 124  2692.438
## 125  2692.438
## 126  2692.438
## 127  2307.808
## 128  2594.380
## 129  2307.808
## 130  2307.808
## 131  2307.808
## 132  2594.380
## 133  2594.380
## 134  2542.571
## 135  2307.808
## 136  2542.571
## 137  2692.438
## 138  2692.438
## 139  2307.808
## 140  2576.259
## 141  2576.259
## 142  2576.259
## 143  2594.380
## 144  2692.438
## 145  2594.380
## 146  2307.808
## 147  2576.259
## 148  2594.380
## 149  2594.380
```

```
## 150  2542.571
## 151  2594.380
## 152  1947.695
## 153  2594.380
## 154  2576.259
## 155  2692.438
## 156  2594.380
## 157  2692.438
## 158  2576.259
## 159  2576.259
## 160  2594.380
## 161  2576.259
## 162  2307.808
## 163  2692.438
## 164  2576.259
## 165  2692.438
## 166  2307.808
## 167  2692.438
## 168  2576.259
## 169  2692.438
## 170  2542.571
## 171  2594.380
## 172  2594.380
## 173  1990.217
## 174  2542.571
## 175  2542.571
## 176  1947.695
## 177  2692.438
## 178  2576.259
## 179  2542.571
## 180  2542.571
## 181  2542.571
## 182  2542.571
## 183  2576.259
## 184  2576.259
## 185  2576.259
## 186  2576.259
## 187  2594.380
## 188  2692.438
## 189  2692.438
## 190  2692.438
## 191  2692.438
## 192  2542.571
## 193  2542.571
## 194  2542.571
## 195  2542.571
## 196  2542.571
## 197  2542.571
## 198  2542.571
## 199  2542.571
## 200  2692.438
## 201  2542.571
## 202  2542.571
## 203  2542.571
```

```
## 204  2542.571
## 205  2307.808
## 206  2692.438
## 207  2542.571
## 208  2692.438
## 209  2307.808
## 210  2542.571
## 211  2692.438
## 212  2576.259
## 213  2692.438
## 214  2576.259
## 215  2576.259
## 216  2692.438
## 217  2307.808
## 218  2307.808
## 219  2307.808
## 220  2594.380
## 221  2576.259
## 222  2542.571
## 223  2542.571
## 224  2594.380
## 225  2594.380
## 226  2594.380
## 227  2594.380
## 228  2576.259
## 229  2692.438
## 230  2692.438
## 231  2692.438
## 232  2307.808
## 233  2692.438
## 234  2692.438
## 235  2594.380
## 236  2307.808
## 237  1947.695
## 238  1947.695
## 239  2594.380
## 240  2576.259
## 241  2692.438
## 242  2542.571
## 243  2307.808
## 244  2692.438
## 245  2542.571
## 246  2542.571
## 247  2542.571
## 248  1990.217
## 249  2576.259
## 250  2542.571
## 251  2576.259
## 252  2576.259
## 253  2542.571
## 254  2542.571
## 255  2307.808
## 256  1990.217
## 257  2576.259
```

```
## 258  1947.695
## 259  2692.438
## 260  2692.438
## 261  2692.438
## 262  1947.695
## 263  2542.571
## 264  2542.571
## 265  2542.571
## 266  2542.571
## 267  2692.438
## 268  2692.438
## 269  2307.808
## 270  2692.438
## 271  2576.259
## 272  2542.571
## 273  2542.571
## 274  1947.695
## 275  2307.808
## 276  2542.571
## 277  2542.571
## 278  1947.695
## 279  2692.438
## 280  2692.438
## 281  2594.380
## 282  1947.695
## 283  2307.808
## 284  2576.259
## 285  1947.695
## 286  2542.571
## 287  2307.808
## 288  2542.571
## 289  2594.380
## 290  2594.380
## 291  2594.380
## 292  2692.438
## 293  2594.380
## 294  2576.259
## 295  2307.808
## 296  2307.808
## 297  2576.259
## 298  2692.438
## 299  2542.571
## 300  2307.808
## 301  1947.695
## 302  2542.571
## 303  2692.438
## 304  2594.380
## 305  2576.259
## 306  1947.695
## 307  2542.571
## 308  1947.695
## 309  2576.259
## 310  2576.259
## 311  2307.808
```

```
## 312  2576.259
## 313  1947.695
## 314  2576.259
## 315  2576.259
## 316  2692.438
## 317  2692.438
## 318  2692.438
## 319  2307.808
## 320  2692.438
## 321  2692.438
## 322  2692.438
## 323  2692.438
## 324  2576.259
## 325  1990.217
## 326  2576.259
## 327  2692.438
## 328  2542.571
## 329  2692.438
## 330  2692.438
## 331  2542.571
## 332  2576.259
## 333  2542.571
## 334  2576.259
## 335  2692.438
## 336  2692.438
## 337  2576.259
## 338  2576.259
## 339  2576.259
## 340  2576.259
## 341  2576.259
## 342  2594.380
## 343  2542.571
## 344  2542.571
## 345  2542.571
## 346  2594.380
## 347  2692.438
## 348  2692.438
## 349  2307.808
## 350  2594.380
## 351  2594.380
## 352  2594.380
## 353  1947.695
## 354  2692.438
## 355  2692.438
## 356  2542.571
## 357  2542.571
## 358  2692.438
## 359  2576.259
## 360  2692.438
## 361  2576.259
## 362  2576.259
## 363  2307.808
## 364  2692.438
## 365  2692.438
```

```
## 366  2692.438
## 367  1990.217
## 368  2307.808
## 369  2692.438
## 370  2576.259
## 371  2576.259
## 372  2542.571
## 373  2594.380
## 374  2692.438
## 375  2594.380
## 376  2692.438
## 377  2692.438
## 378  2692.438
## 379  2692.438
## 380  2594.380
## 381  2692.438
## 382  2576.259
## 383  2692.438
## 384  2594.380
## 385  1990.217
## 386  1947.695
## 387  2576.259
## 388  2692.438
## 389  2542.571
## 390  2692.438
## 391  1947.695
## 392  1947.695
## 393  1947.695
## 394  1947.695
## 395  1947.695
## 396  1947.695
## 397  2307.808
## 398  2307.808
## 399  2307.808
## 400  2307.808
## 401  2307.808
## 402  2307.808
## 403  2307.808
## 404  1947.695
## 405  2307.808
## 406  2542.571
## 407  2542.571
## 408  2692.438
## 409  2307.808
## 410  2307.808
## 411  1947.695
## 412  1947.695
## 413  2576.259
## 414  2692.438
## 415  2576.259
## 416  2542.571
## 417  2692.438
## 418  1947.695
## 419  2576.259
```

```
## 420  2542.571
## 421  2692.438
## 422  2692.438
## 423  2594.380
## 424  1990.217
## 425  2542.571
## 426  2692.438
## 427  2576.259
## 428  2542.571
## 429  1947.695
## 430  2542.571
## 431  2542.571
## 432  2542.571
## 433  2594.380
## 434  2692.438
## 435  2576.259
## 436  2307.808
## 437  2307.808
## 438  2307.808
## 439  2307.808
## 440  1990.217
## 441  2692.438
## 442  2307.808
## 443  2542.571
## 444  2542.571
## 445  2576.259
## 446  1947.695
## 447  2576.259
## 448  2594.380
## 449  2692.438
## 450  2307.808
## 451  2307.808
## 452  2542.571
## 453  1947.695
## 454  2542.571
## 455  2542.571
## 456  2692.438
## 457  2542.571
## 458  2542.571
## 459  2542.571
## 460  2542.571
## 461  1990.217
## 462  2542.571
## 463  2542.571
## 464  2542.571
## 465  2542.571
## 466  2307.808
## 467  2692.438
## 468  2307.808
## 469  2542.571
## 470  2542.571
## 471  2692.438
## 472  2542.571
## 473  2307.808
```

```
## 474  2576.259
## 475  2307.808
## 476  2692.438
## 477  2692.438
## 478  2576.259
## 479  2542.571
## 480  2542.571
## 481  2542.571
## 482  2542.571
## 483  2542.571
## 484  2542.571
## 485  2542.571
## 486  2594.380
## 487  2542.571
## 488  2542.571
## 489  2542.571
## 490  2542.571
## 491  2542.571
## 492  2542.571
## 493  2692.438
## 494  1947.695
## 495  2692.438
## 496  2594.380
## 497  2594.380
## 498  2594.380
## 499  2594.380
## 500  1990.217
## 501  2594.380
## 502  2542.571
## 503  2542.571
## 504  2594.380
## 505  2542.571
## 506  2594.380
## 507  2594.380
## 508  2542.571
## 509  2692.438
## 510  2594.380
## 511  2594.380
## 512  1947.695
## 513  2542.571
## 514  2542.571
## 515  2307.808
## 516  2542.571
## 517  2594.380
## 518  1947.695
## 519  2542.571
## 520  2692.438
## 521  2542.571
## 522  2576.259
## 523  2307.808
## 524  2576.259
## 525  2307.808
## 526  1990.217
## 527  2307.808
```

```
## 528  2542.571
## 529  2542.571
## 530  1947.695
## 531  2542.571
## 532  2594.380
## 533  2542.571
## 534  1947.695
## 535  2692.438
## 536  2692.438
## 537  2594.380
## 538  2594.380
## 539  2307.808
## 540  2307.808
## 541  2307.808
## 542  2594.380
## 543  2307.808
## 544  2692.438
## 545  2542.571
## 546  2542.571
## 547  2692.438
## 548  2542.571
## 549  2542.571
## 550  2542.571
## 551  2307.808
## 552  2692.438
## 553  2692.438
## 554  2692.438
## 555  1947.695
## 556  2576.259
## 557  2542.571
## 558  2542.571
## 559  2542.571
## 560  2542.571
## 561  2576.259
## 562  2692.438
## 563  1947.695
## 564  2576.259
## 565  2576.259
## 566  2307.808
## 567  2692.438
## 568  1990.217
## 569  1947.695
## 570  2542.571
## 571  2594.380
## 572  2594.380
## 573  2692.438
## 574  2307.808
## 575  2692.438
## 576  2576.259
## 577  2307.808
## 578  2594.380
## 579  2692.438
## 580  1947.695
## 581  1990.217
```

```
## 582  2576.259
## 583  2542.571
## 584  2692.438
## 585  2692.438
## 586  2542.571
## 587  2542.571
## 588  2542.571
## 589  2692.438
## 590  2307.808
## 591  2542.571
## 592  2542.571
## 593  2542.571
## 594  2692.438
## 595  2307.808
## 596  2692.438
## 597  2692.438
## 598  2692.438
## 599  2692.438
## 600  2576.259
## 601  2692.438
## 602  2542.571
## 603  2692.438
## 604  2692.438
## 605  2576.259
## 606  2542.571
## 607  2692.438
## 608  2692.438
## 609  2692.438
## 610  2692.438
## 611  2692.438
## 612  2692.438
## 613  2692.438
## 614  2576.259
## 615  2542.571
## 616  2576.259
## 617  2576.259
## 618  2692.438
## 619  2576.259
## 620  2692.438
## 621  2307.808
## 622  2692.438
## 623  2692.438
## 624  2692.438
## 625  2594.380
## 626  2692.438
## 627  2307.808
## 628  2692.438
## 629  2594.380
## 630  2594.380
## 631  2692.438
## 632  2692.438
## 633  2692.438
## 634  2692.438
## 635  2692.438
```

```
## 636  2576.259
## 637  2594.380
## 638  2692.438
## 639  2576.259
## 640  1947.695
## 641  2692.438
## 642  2542.571
## 643  2542.571
## 644  2542.571
## 645  2692.438
## 646  2594.380
## 647  2576.259
## 648  2692.438
## 649  2542.571
## 650  2692.438
## 651  2576.259
## 652  2542.571
## 653  2542.571
## 654  1947.695
## 655  1947.695
## 656  2307.808
## 657  2542.571
## 658  2542.571
## 659  2307.808
## 660  2307.808
## 661  2594.380
## 662  2692.438
## 663  2307.808
## 664  2307.808
## 665  2307.808
## 666  2542.571
## 667  2542.571
## 668  2692.438
## 669  2542.571
## 670  2576.259
## 671  2576.259
## 672  2307.808
## 673  2542.571
## 674  2594.380
## 675  2576.259
## 676  2594.380
## 677  2594.380
## 678  2542.571
## 679  2542.571
## 680  2542.571
## 681  2692.438
## 682  1990.217
## 683  2307.808
## 684  2692.438
## 685  2594.380
## 686  2692.438
## 687  2307.808
## 688  2692.438
## 689  2542.571
```

```
## 690  1947.695
## 691  2576.259
## 692  2542.571
## 693  1947.695
## 694  2692.438
## 695  2576.259
## 696  2594.380
## 697  2594.380
## 698  2542.571
## 699  1947.695
## 700  2307.808
## 701  2542.571
## 702  2307.808
## 703  2542.571
## 704  1990.217
## 705  1990.217
## 706  1990.217
## 707  2307.808
## 708  2542.571
## 709  2692.438
## 710  2692.438
## 711  1990.217
## 712  2594.380
## 713  2542.571
## 714  1947.695
## 715  2576.259
## 716  2307.808
## 717  2692.438
## 718  2576.259
## 719  2692.438
## 720  1947.695
## 721  2542.571
## 722  2542.571
## 723  2307.808
## 724  2576.259
## 725  2542.571
## 726  2542.571
## 727  2542.571
## 728  2576.259
## 729  2542.571
## 730  2542.571
## 731  1947.695
## 732  1947.695
## 733  1947.695
## 734  1947.695
## 735  1947.695
## 736  1947.695
## 737  2576.259
## 738  2307.808
## 739  2307.808
## 740  2576.259
## 741  2692.438
## 742  2576.259
## 743  2576.259
```

```
## 744  2692.438
## 745  1990.217
## 746  1990.217
## 747  1990.217
## 748  1947.695
## 749  1947.695
## 750  1947.695
## 751  2307.808
## 752  2692.438
## 753  2692.438
## 754  2542.571
## 755  2576.259
## 756  2542.571
## 757  1947.695
## 758  2542.571
## 759  2576.259
## 760  2692.438
## 761  2692.438
## 762  2542.571
## 763  2542.571
## 764  2542.571
## 765  2542.571
## 766  1990.217
## 767  2542.571
## 768  2594.380
## 769  2594.380
## 770  2542.571
## 771  2576.259
## 772  2594.380
## 773  2594.380
## 774  2594.380
## 775  2692.438
## 776  2576.259
## 777  2594.380
## 778  2692.438
## 779  2542.571
## 780  2692.438
## 781  1990.217
## 782  2542.571
## 783  2594.380
## 784  2692.438
## 785  2594.380
## 786  2576.259
## 787  2576.259
## 788  1947.695
## 789  2692.438
## 790  2542.571
## 791  2594.380
## 792  2594.380
## 793  2692.438
## 794  2594.380
## 795  2692.438
## 796  2692.438
## 797  2307.808
```

```
## 798  2542.571
## 799  2307.808
## 800  2542.571
## 801  2542.571
## 802  2542.571
## 803  2542.571
## 804  1947.695
## 805  2594.380
## 806  2576.259
## 807  2692.438
## 808  2594.380
## 809  2594.380
## 810  2594.380
## 811  2594.380
## 812  2594.380
## 813  2594.380
## 814  1990.217
## 815  1990.217
## 816  2307.808
## 817  2542.571
## 818  2692.438
## 819  2542.571
## 820  2542.571
## 821  2542.571
## 822  1947.695
## 823  2594.380
## 824  2594.380
## 825  2594.380
## 826  2542.571
## 827  2307.808
## 828  1947.695
## 829  2307.808
## 830  2594.380
## 831  2307.808
## 832  2307.808
## 833  2692.438
## 834  2692.438
## 835  2692.438
## 836  2692.438
## 837  2692.438
## 838  2692.438
## 839  2576.259
## 840  2692.438
## 841  1947.695
## 842  2576.259
## 843  2542.571
## 844  2542.571
## 845  2594.380
## 846  2542.571
## 847  2576.259
## 848  2576.259
## 849  2307.808
## 850  2692.438
## 851  2542.571
```

```
## 852  2542.571
## 853  2576.259
## 854  2307.808
## 855  2692.438
## 856  2307.808
## 857  2542.571
## 858  1947.695
## 859  2542.571
## 860  2542.571
## 861  1990.217
## 862  1947.695
## 863  2576.259
## 864  2594.380
## 865  2594.380
## 866  1947.695
## 867  2576.259
## 868  2594.380
## 869  2542.571
## 870  2594.380
## 871  2594.380
## 872  2594.380
## 873  2542.571
## 874  2542.571
## 875  2576.259
## 876  2692.438
## 877  2542.571
## 878  2542.571
## 879  1947.695
## 880  2307.808
## 881  1947.695
## 882  1990.217
## 883  1990.217
## 884  1947.695
## 885  1947.695
## 886  1990.217
## 887  2692.438
## 888  2594.380
## 889  2692.438
## 890  2692.438
## 891  2576.259
## 892  2692.438
## 893  2692.438
## 894  2692.438
## 895  2576.259
## 896  2307.808
## 897  2307.808
## 898  1990.217
## 899  1947.695
## 900  2307.808
## 901  2542.571
## 902  2542.571
## 903  2307.808
## 904  2542.571
## 905  2576.259
```

```
## 906  2307.808
## 907  2692.438
## 908  2692.438
## 909  2692.438
## 910  2594.380
## 911  1947.695
## 912  2692.438
## 913  2692.438
## 914  2542.571
## 915  2692.438
## 916  2692.438
## 917  2576.259
## 918  2576.259
## 919  1990.217
## 920  2692.438
## 921  2692.438
## 922  1990.217
## 923  2594.380
## 924  2307.808
## 925  2542.571
## 926  2307.808
## 927  2307.808
## 928  2692.438
## 929  2692.438
## 930  2692.438
## 931  2692.438
## 932  2594.380
## 933  2307.808
## 934  2307.808
## 935  1947.695
## 936  2692.438
## 937  1947.695
## 938  2594.380
## 939  2542.571
## 940  2594.380
## 941  2542.571
## 942  2542.571
## 943  2594.380
## 944  2542.571
## 945  1947.695
## 946  2576.259
## 947  2542.571
## 948  2692.438
## 949  2307.808
## 950  2692.438
## 951  2542.571
## 952  2542.571
## 953  2692.438
## 954  2692.438
## 955  2542.571
## 956  2692.438
## 957  2692.438
## 958  2692.438
## 959  2692.438
```

```
## 960  2576.259
## 961  2542.571
## 962  2692.438
## 963  2542.571
## 964  2542.571
## 965  2594.380
## 966  2594.380
## 967  2307.808
## 968  2307.808
## 969  2692.438
## 970  2576.259
## 971  2576.259
## 972  2692.438
## 973  2307.808
## 974  2576.259
## 975  2576.259
## 976  2576.259
## 977  2576.259
## 978  2542.571
## 979  2576.259
## 980  2692.438
## 981  2576.259
## 982  2692.438
## 983  2576.259
## 984  2576.259
## 985  2576.259
## 986  2576.259
## 987  2692.438
## 988  2594.380
## 989  2542.571
## 990  2594.380
## 991  2542.571
## 992  1947.695
## 993  1947.695
## 994  2594.380
## 995  2542.571
## 996  2594.380
## 997  2542.571
## 998  2692.438
## 999  2594.380
## 1000 1990.217
```

```r
I = diag(nrow(H))
e = (I - H) %*% y_diamond


t(e) %*% yhat_diamond  # 0 -> but since the inversion the bits are off and giving us an error
```

```
##               [,1]
## [1,] 1.063338e-06
```

```r
print("...")
```

```
## [1] "..."
```

Compute SST, SSR and SSE and R^2 and then show that SST = SSR + SSE.

```r
SST = sum((y_diamond - mean(y_diamond))^2)
SSE = sum(e^2)
SSR = sum((yhat_diamond - mean(y_diamond))^2)

SST - sum(SSR + SSE)
```

```
## [1] 2.384186e-07
```

```r
RSQ = SSR / SST
RSQ
```

```
## [1] 0.07918666
```

Find the angle theta between y - ybar 1 and yhat - ybar 1 and then verify that its cosine squared is the same as the R^2 from the previous problem.

```r
#TO-DO

numer = sqrt(sum((yhat_diamond - mean(y_diamond))^2))
denom = sqrt(sum((y_diamond - mean(y_diamond))^2))
theta = acos(numer/denom)
cos(theta)^2
```

```
## [1] 0.07918666
```

Project the y vector onto each column of the X matrix and test if the sum of these projections is the same as yhat.

```r
n = nrow(X_diamonds)
sum_proj_y = matrix(0, nrow=n, ncol=1)

for (j in 1:ncol(X_diamonds)) {
  X_j = X_diamonds[, j, drop=FALSE]
  sum_proj_y = sum_proj_y + (X_j %*% t(X_j)/sum(X_j^2)) %*% y_diamond
}


sum_proj_y
```

```
##            [,1]
## 1     4476.1881
## 2     4476.1881
## 3     4476.1881
## 4     1062.8515
## 5      137.4839
## 6      137.4839
## 7     1062.8515
## 8     2020.1621
## 9     4476.1881
```

```
## 10   2020.1621
## 11    137.4839
## 12    137.4839
## 13   4449.3316
## 14    137.4839
## 15   4476.1881
## 16   4476.1881
## 17   1062.8515
## 18    137.4839
## 19    137.4839
## 20    137.4839
## 21   1062.8515
## 22   4476.1881
## 23   2020.1621
## 24    137.4839
## 25    137.4839
## 26   2708.5604
## 27   1062.8515
## 28    137.4839
## 29   2481.2024
## 30   4449.3316
## 31   4449.3316
## 32   4449.3316
## 33   4476.1881
## 34   4476.1881
## 35   2481.2024
## 36   4449.3316
## 37   4476.1881
## 38   2020.1621
## 39   2481.2024
## 40   1062.8515
## 41   1062.8515
## 42    137.4839
## 43   2481.2024
## 44   2481.2024
## 45   2020.1621
## 46   4449.3316
## 47   2020.1621
## 48   2020.1621
## 49   4476.1881
## 50   2020.1621
## 51   4449.3316
## 52   2708.5604
## 53   1062.8515
## 54   4476.1881
## 55   2481.2024
## 56   1062.8515
## 57    137.4839
## 58   1062.8515
## 59   1062.8515
## 60   1062.8515
## 61   1062.8515
## 62   2481.2024
## 63   2481.2024
```

```
## 64   2481.2024
## 65   1062.8515
## 66   2708.5604
## 67   1062.8515
## 68   2708.5604
## 69   2708.5604
## 70   4476.1881
## 71   2481.2024
## 72   2020.1621
## 73   2020.1621
## 74   2020.1621
## 75   2020.1621
## 76   4449.3316
## 77   4476.1881
## 78   2481.2024
## 79   2481.2024
## 80   4476.1881
## 81   4476.1881
## 82   2481.2024
## 83   4476.1881
## 84   1062.8515
## 85   4476.1881
## 86   2708.5604
## 87   2020.1621
## 88   2020.1621
## 89   2020.1621
## 90   1062.8515
## 91   4476.1881
## 92   4476.1881
## 93   2708.5604
## 94   4476.1881
## 95   2708.5604
## 96   4476.1881
## 97   4449.3316
## 98   4449.3316
## 99   4476.1881
## 100  2020.1621
## 101  2481.2024
## 102  4476.1881
## 103  2708.5604
## 104  2708.5604
## 105  1062.8515
## 106  2708.5604
## 107  2708.5604
## 108  1062.8515
## 109  4449.3316
## 110  4476.1881
## 111  4449.3316
## 112  4476.1881
## 113  1062.8515
## 114  2708.5604
## 115  4449.3316
## 116  4449.3316
## 117  4449.3316
```

```
## 118  2708.5604
## 119  4476.1881
## 120  4449.3316
## 121  2481.2024
## 122  4476.1881
## 123  4449.3316
## 124  4449.3316
## 125  4449.3316
## 126  4449.3316
## 127  2020.1621
## 128  2481.2024
## 129  2020.1621
## 130  2020.1621
## 131  2020.1621
## 132  2481.2024
## 133  2481.2024
## 134  4476.1881
## 135  2020.1621
## 136  4476.1881
## 137  4449.3316
## 138  4449.3316
## 139  2020.1621
## 140  2708.5604
## 141  2708.5604
## 142  2708.5604
## 143  2481.2024
## 144  4449.3316
## 145  2481.2024
## 146  2020.1621
## 147  2708.5604
## 148  2481.2024
## 149  2481.2024
## 150  4476.1881
## 151  2481.2024
## 152  1062.8515
## 153  2481.2024
## 154  2708.5604
## 155  4449.3316
## 156  2481.2024
## 157  4449.3316
## 158  2708.5604
## 159  2708.5604
## 160  2481.2024
## 161  2708.5604
## 162  2020.1621
## 163  4449.3316
## 164  2708.5604
## 165  4449.3316
## 166  2020.1621
## 167  4449.3316
## 168  2708.5604
## 169  4449.3316
## 170  4476.1881
## 171  2481.2024
```

```
## 172  2481.2024
## 173   137.4839
## 174  4476.1881
## 175  4476.1881
## 176  1062.8515
## 177  4449.3316
## 178  2708.5604
## 179  4476.1881
## 180  4476.1881
## 181  4476.1881
## 182  4476.1881
## 183  2708.5604
## 184  2708.5604
## 185  2708.5604
## 186  2708.5604
## 187  2481.2024
## 188  4449.3316
## 189  4449.3316
## 190  4449.3316
## 191  4449.3316
## 192  4476.1881
## 193  4476.1881
## 194  4476.1881
## 195  4476.1881
## 196  4476.1881
## 197  4476.1881
## 198  4476.1881
## 199  4476.1881
## 200  4449.3316
## 201  4476.1881
## 202  4476.1881
## 203  4476.1881
## 204  4476.1881
## 205  2020.1621
## 206  4449.3316
## 207  4476.1881
## 208  4449.3316
## 209  2020.1621
## 210  4476.1881
## 211  4449.3316
## 212  2708.5604
## 213  4449.3316
## 214  2708.5604
## 215  2708.5604
## 216  4449.3316
## 217  2020.1621
## 218  2020.1621
## 219  2020.1621
## 220  2481.2024
## 221  2708.5604
## 222  4476.1881
## 223  4476.1881
## 224  2481.2024
## 225  2481.2024
```

```
## 226  2481.2024
## 227  2481.2024
## 228  2708.5604
## 229  4449.3316
## 230  4449.3316
## 231  4449.3316
## 232  2020.1621
## 233  4449.3316
## 234  4449.3316
## 235  2481.2024
## 236  2020.1621
## 237  1062.8515
## 238  1062.8515
## 239  2481.2024
## 240  2708.5604
## 241  4449.3316
## 242  4476.1881
## 243  2020.1621
## 244  4449.3316
## 245  4476.1881
## 246  4476.1881
## 247  4476.1881
## 248   137.4839
## 249  2708.5604
## 250  4476.1881
## 251  2708.5604
## 252  2708.5604
## 253  4476.1881
## 254  4476.1881
## 255  2020.1621
## 256   137.4839
## 257  2708.5604
## 258  1062.8515
## 259  4449.3316
## 260  4449.3316
## 261  4449.3316
## 262  1062.8515
## 263  4476.1881
## 264  4476.1881
## 265  4476.1881
## 266  4476.1881
## 267  4449.3316
## 268  4449.3316
## 269  2020.1621
## 270  4449.3316
## 271  2708.5604
## 272  4476.1881
## 273  4476.1881
## 274  1062.8515
## 275  2020.1621
## 276  4476.1881
## 277  4476.1881
## 278  1062.8515
## 279  4449.3316
```

```
## 280  4449.3316
## 281  2481.2024
## 282  1062.8515
## 283  2020.1621
## 284  2708.5604
## 285  1062.8515
## 286  4476.1881
## 287  2020.1621
## 288  4476.1881
## 289  2481.2024
## 290  2481.2024
## 291  2481.2024
## 292  4449.3316
## 293  2481.2024
## 294  2708.5604
## 295  2020.1621
## 296  2020.1621
## 297  2708.5604
## 298  4449.3316
## 299  4476.1881
## 300  2020.1621
## 301  1062.8515
## 302  4476.1881
## 303  4449.3316
## 304  2481.2024
## 305  2708.5604
## 306  1062.8515
## 307  4476.1881
## 308  1062.8515
## 309  2708.5604
## 310  2708.5604
## 311  2020.1621
## 312  2708.5604
## 313  1062.8515
## 314  2708.5604
## 315  2708.5604
## 316  4449.3316
## 317  4449.3316
## 318  4449.3316
## 319  2020.1621
## 320  4449.3316
## 321  4449.3316
## 322  4449.3316
## 323  4449.3316
## 324  2708.5604
## 325   137.4839
## 326  2708.5604
## 327  4449.3316
## 328  4476.1881
## 329  4449.3316
## 330  4449.3316
## 331  4476.1881
## 332  2708.5604
## 333  4476.1881
```

```
## 334  2708.5604
## 335  4449.3316
## 336  4449.3316
## 337  2708.5604
## 338  2708.5604
## 339  2708.5604
## 340  2708.5604
## 341  2708.5604
## 342  2481.2024
## 343  4476.1881
## 344  4476.1881
## 345  4476.1881
## 346  2481.2024
## 347  4449.3316
## 348  4449.3316
## 349  2020.1621
## 350  2481.2024
## 351  2481.2024
## 352  2481.2024
## 353  1062.8515
## 354  4449.3316
## 355  4449.3316
## 356  4476.1881
## 357  4476.1881
## 358  4449.3316
## 359  2708.5604
## 360  4449.3316
## 361  2708.5604
## 362  2708.5604
## 363  2020.1621
## 364  4449.3316
## 365  4449.3316
## 366  4449.3316
## 367   137.4839
## 368  2020.1621
## 369  4449.3316
## 370  2708.5604
## 371  2708.5604
## 372  4476.1881
## 373  2481.2024
## 374  4449.3316
## 375  2481.2024
## 376  4449.3316
## 377  4449.3316
## 378  4449.3316
## 379  4449.3316
## 380  2481.2024
## 381  4449.3316
## 382  2708.5604
## 383  4449.3316
## 384  2481.2024
## 385   137.4839
## 386  1062.8515
## 387  2708.5604
```

```
## 388  4449.3316
## 389  4476.1881
## 390  4449.3316
## 391  1062.8515
## 392  1062.8515
## 393  1062.8515
## 394  1062.8515
## 395  1062.8515
## 396  1062.8515
## 397  2020.1621
## 398  2020.1621
## 399  2020.1621
## 400  2020.1621
## 401  2020.1621
## 402  2020.1621
## 403  2020.1621
## 404  1062.8515
## 405  2020.1621
## 406  4476.1881
## 407  4476.1881
## 408  4449.3316
## 409  2020.1621
## 410  2020.1621
## 411  1062.8515
## 412  1062.8515
## 413  2708.5604
## 414  4449.3316
## 415  2708.5604
## 416  4476.1881
## 417  4449.3316
## 418  1062.8515
## 419  2708.5604
## 420  4476.1881
## 421  4449.3316
## 422  4449.3316
## 423  2481.2024
## 424   137.4839
## 425  4476.1881
## 426  4449.3316
## 427  2708.5604
## 428  4476.1881
## 429  1062.8515
## 430  4476.1881
## 431  4476.1881
## 432  4476.1881
## 433  2481.2024
## 434  4449.3316
## 435  2708.5604
## 436  2020.1621
## 437  2020.1621
## 438  2020.1621
## 439  2020.1621
## 440   137.4839
## 441  4449.3316
```

```
## 442  2020.1621
## 443  4476.1881
## 444  4476.1881
## 445  2708.5604
## 446  1062.8515
## 447  2708.5604
## 448  2481.2024
## 449  4449.3316
## 450  2020.1621
## 451  2020.1621
## 452  4476.1881
## 453  1062.8515
## 454  4476.1881
## 455  4476.1881
## 456  4449.3316
## 457  4476.1881
## 458  4476.1881
## 459  4476.1881
## 460  4476.1881
## 461   137.4839
## 462  4476.1881
## 463  4476.1881
## 464  4476.1881
## 465  4476.1881
## 466  2020.1621
## 467  4449.3316
## 468  2020.1621
## 469  4476.1881
## 470  4476.1881
## 471  4449.3316
## 472  4476.1881
## 473  2020.1621
## 474  2708.5604
## 475  2020.1621
## 476  4449.3316
## 477  4449.3316
## 478  2708.5604
## 479  4476.1881
## 480  4476.1881
## 481  4476.1881
## 482  4476.1881
## 483  4476.1881
## 484  4476.1881
## 485  4476.1881
## 486  2481.2024
## 487  4476.1881
## 488  4476.1881
## 489  4476.1881
## 490  4476.1881
## 491  4476.1881
## 492  4476.1881
## 493  4449.3316
## 494  1062.8515
## 495  4449.3316
```

```
## 496  2481.2024
## 497  2481.2024
## 498  2481.2024
## 499  2481.2024
## 500   137.4839
## 501  2481.2024
## 502  4476.1881
## 503  4476.1881
## 504  2481.2024
## 505  4476.1881
## 506  2481.2024
## 507  2481.2024
## 508  4476.1881
## 509  4449.3316
## 510  2481.2024
## 511  2481.2024
## 512  1062.8515
## 513  4476.1881
## 514  4476.1881
## 515  2020.1621
## 516  4476.1881
## 517  2481.2024
## 518  1062.8515
## 519  4476.1881
## 520  4449.3316
## 521  4476.1881
## 522  2708.5604
## 523  2020.1621
## 524  2708.5604
## 525  2020.1621
## 526   137.4839
## 527  2020.1621
## 528  4476.1881
## 529  4476.1881
## 530  1062.8515
## 531  4476.1881
## 532  2481.2024
## 533  4476.1881
## 534  1062.8515
## 535  4449.3316
## 536  4449.3316
## 537  2481.2024
## 538  2481.2024
## 539  2020.1621
## 540  2020.1621
## 541  2020.1621
## 542  2481.2024
## 543  2020.1621
## 544  4449.3316
## 545  4476.1881
## 546  4476.1881
## 547  4449.3316
## 548  4476.1881
## 549  4476.1881
```

```
## 550  4476.1881
## 551  2020.1621
## 552  4449.3316
## 553  4449.3316
## 554  4449.3316
## 555  1062.8515
## 556  2708.5604
## 557  4476.1881
## 558  4476.1881
## 559  4476.1881
## 560  4476.1881
## 561  2708.5604
## 562  4449.3316
## 563  1062.8515
## 564  2708.5604
## 565  2708.5604
## 566  2020.1621
## 567  4449.3316
## 568   137.4839
## 569  1062.8515
## 570  4476.1881
## 571  2481.2024
## 572  2481.2024
## 573  4449.3316
## 574  2020.1621
## 575  4449.3316
## 576  2708.5604
## 577  2020.1621
## 578  2481.2024
## 579  4449.3316
## 580  1062.8515
## 581   137.4839
## 582  2708.5604
## 583  4476.1881
## 584  4449.3316
## 585  4449.3316
## 586  4476.1881
## 587  4476.1881
## 588  4476.1881
## 589  4449.3316
## 590  2020.1621
## 591  4476.1881
## 592  4476.1881
## 593  4476.1881
## 594  4449.3316
## 595  2020.1621
## 596  4449.3316
## 597  4449.3316
## 598  4449.3316
## 599  4449.3316
## 600  2708.5604
## 601  4449.3316
## 602  4476.1881
## 603  4449.3316
```

```
## 604  4449.3316
## 605  2708.5604
## 606  4476.1881
## 607  4449.3316
## 608  4449.3316
## 609  4449.3316
## 610  4449.3316
## 611  4449.3316
## 612  4449.3316
## 613  4449.3316
## 614  2708.5604
## 615  4476.1881
## 616  2708.5604
## 617  2708.5604
## 618  4449.3316
## 619  2708.5604
## 620  4449.3316
## 621  2020.1621
## 622  4449.3316
## 623  4449.3316
## 624  4449.3316
## 625  2481.2024
## 626  4449.3316
## 627  2020.1621
## 628  4449.3316
## 629  2481.2024
## 630  2481.2024
## 631  4449.3316
## 632  4449.3316
## 633  4449.3316
## 634  4449.3316
## 635  4449.3316
## 636  2708.5604
## 637  2481.2024
## 638  4449.3316
## 639  2708.5604
## 640  1062.8515
## 641  4449.3316
## 642  4476.1881
## 643  4476.1881
## 644  4476.1881
## 645  4449.3316
## 646  2481.2024
## 647  2708.5604
## 648  4449.3316
## 649  4476.1881
## 650  4449.3316
## 651  2708.5604
## 652  4476.1881
## 653  4476.1881
## 654  1062.8515
## 655  1062.8515
## 656  2020.1621
## 657  4476.1881
```

```
## 658  4476.1881
## 659  2020.1621
## 660  2020.1621
## 661  2481.2024
## 662  4449.3316
## 663  2020.1621
## 664  2020.1621
## 665  2020.1621
## 666  4476.1881
## 667  4476.1881
## 668  4449.3316
## 669  4476.1881
## 670  2708.5604
## 671  2708.5604
## 672  2020.1621
## 673  4476.1881
## 674  2481.2024
## 675  2708.5604
## 676  2481.2024
## 677  2481.2024
## 678  4476.1881
## 679  4476.1881
## 680  4476.1881
## 681  4449.3316
## 682   137.4839
## 683  2020.1621
## 684  4449.3316
## 685  2481.2024
## 686  4449.3316
## 687  2020.1621
## 688  4449.3316
## 689  4476.1881
## 690  1062.8515
## 691  2708.5604
## 692  4476.1881
## 693  1062.8515
## 694  4449.3316
## 695  2708.5604
## 696  2481.2024
## 697  2481.2024
## 698  4476.1881
## 699  1062.8515
## 700  2020.1621
## 701  4476.1881
## 702  2020.1621
## 703  4476.1881
## 704   137.4839
## 705   137.4839
## 706   137.4839
## 707  2020.1621
## 708  4476.1881
## 709  4449.3316
## 710  4449.3316
## 711   137.4839
```

```
## 712  2481.2024
## 713  4476.1881
## 714  1062.8515
## 715  2708.5604
## 716  2020.1621
## 717  4449.3316
## 718  2708.5604
## 719  4449.3316
## 720  1062.8515
## 721  4476.1881
## 722  4476.1881
## 723  2020.1621
## 724  2708.5604
## 725  4476.1881
## 726  4476.1881
## 727  4476.1881
## 728  2708.5604
## 729  4476.1881
## 730  4476.1881
## 731  1062.8515
## 732  1062.8515
## 733  1062.8515
## 734  1062.8515
## 735  1062.8515
## 736  1062.8515
## 737  2708.5604
## 738  2020.1621
## 739  2020.1621
## 740  2708.5604
## 741  4449.3316
## 742  2708.5604
## 743  2708.5604
## 744  4449.3316
## 745   137.4839
## 746   137.4839
## 747   137.4839
## 748  1062.8515
## 749  1062.8515
## 750  1062.8515
## 751  2020.1621
## 752  4449.3316
## 753  4449.3316
## 754  4476.1881
## 755  2708.5604
## 756  4476.1881
## 757  1062.8515
## 758  4476.1881
## 759  2708.5604
## 760  4449.3316
## 761  4449.3316
## 762  4476.1881
## 763  4476.1881
## 764  4476.1881
## 765  4476.1881
```

```
## 766   137.4839
## 767  4476.1881
## 768  2481.2024
## 769  2481.2024
## 770  4476.1881
## 771  2708.5604
## 772  2481.2024
## 773  2481.2024
## 774  2481.2024
## 775  4449.3316
## 776  2708.5604
## 777  2481.2024
## 778  4449.3316
## 779  4476.1881
## 780  4449.3316
## 781   137.4839
## 782  4476.1881
## 783  2481.2024
## 784  4449.3316
## 785  2481.2024
## 786  2708.5604
## 787  2708.5604
## 788  1062.8515
## 789  4449.3316
## 790  4476.1881
## 791  2481.2024
## 792  2481.2024
## 793  4449.3316
## 794  2481.2024
## 795  4449.3316
## 796  4449.3316
## 797  2020.1621
## 798  4476.1881
## 799  2020.1621
## 800  4476.1881
## 801  4476.1881
## 802  4476.1881
## 803  4476.1881
## 804  1062.8515
## 805  2481.2024
## 806  2708.5604
## 807  4449.3316
## 808  2481.2024
## 809  2481.2024
## 810  2481.2024
## 811  2481.2024
## 812  2481.2024
## 813  2481.2024
## 814   137.4839
## 815   137.4839
## 816  2020.1621
## 817  4476.1881
## 818  4449.3316
## 819  4476.1881
```

```
## 820  4476.1881
## 821  4476.1881
## 822  1062.8515
## 823  2481.2024
## 824  2481.2024
## 825  2481.2024
## 826  4476.1881
## 827  2020.1621
## 828  1062.8515
## 829  2020.1621
## 830  2481.2024
## 831  2020.1621
## 832  2020.1621
## 833  4449.3316
## 834  4449.3316
## 835  4449.3316
## 836  4449.3316
## 837  4449.3316
## 838  4449.3316
## 839  2708.5604
## 840  4449.3316
## 841  1062.8515
## 842  2708.5604
## 843  4476.1881
## 844  4476.1881
## 845  2481.2024
## 846  4476.1881
## 847  2708.5604
## 848  2708.5604
## 849  2020.1621
## 850  4449.3316
## 851  4476.1881
## 852  4476.1881
## 853  2708.5604
## 854  2020.1621
## 855  4449.3316
## 856  2020.1621
## 857  4476.1881
## 858  1062.8515
## 859  4476.1881
## 860  4476.1881
## 861   137.4839
## 862  1062.8515
## 863  2708.5604
## 864  2481.2024
## 865  2481.2024
## 866  1062.8515
## 867  2708.5604
## 868  2481.2024
## 869  4476.1881
## 870  2481.2024
## 871  2481.2024
## 872  2481.2024
## 873  4476.1881
```

```
## 874  4476.1881
## 875  2708.5604
## 876  4449.3316
## 877  4476.1881
## 878  4476.1881
## 879  1062.8515
## 880  2020.1621
## 881  1062.8515
## 882   137.4839
## 883   137.4839
## 884  1062.8515
## 885  1062.8515
## 886   137.4839
## 887  4449.3316
## 888  2481.2024
## 889  4449.3316
## 890  4449.3316
## 891  2708.5604
## 892  4449.3316
## 893  4449.3316
## 894  4449.3316
## 895  2708.5604
## 896  2020.1621
## 897  2020.1621
## 898   137.4839
## 899  1062.8515
## 900  2020.1621
## 901  4476.1881
## 902  4476.1881
## 903  2020.1621
## 904  4476.1881
## 905  2708.5604
## 906  2020.1621
## 907  4449.3316
## 908  4449.3316
## 909  4449.3316
## 910  2481.2024
## 911  1062.8515
## 912  4449.3316
## 913  4449.3316
## 914  4476.1881
## 915  4449.3316
## 916  4449.3316
## 917  2708.5604
## 918  2708.5604
## 919   137.4839
## 920  4449.3316
## 921  4449.3316
## 922   137.4839
## 923  2481.2024
## 924  2020.1621
## 925  4476.1881
## 926  2020.1621
## 927  2020.1621
```

```
## 928  4449.3316
## 929  4449.3316
## 930  4449.3316
## 931  4449.3316
## 932  2481.2024
## 933  2020.1621
## 934  2020.1621
## 935  1062.8515
## 936  4449.3316
## 937  1062.8515
## 938  2481.2024
## 939  4476.1881
## 940  2481.2024
## 941  4476.1881
## 942  4476.1881
## 943  2481.2024
## 944  4476.1881
## 945  1062.8515
## 946  2708.5604
## 947  4476.1881
## 948  4449.3316
## 949  2020.1621
## 950  4449.3316
## 951  4476.1881
## 952  4476.1881
## 953  4449.3316
## 954  4449.3316
## 955  4476.1881
## 956  4449.3316
## 957  4449.3316
## 958  4449.3316
## 959  4449.3316
## 960  2708.5604
## 961  4476.1881
## 962  4449.3316
## 963  4476.1881
## 964  4476.1881
## 965  2481.2024
## 966  2481.2024
## 967  2020.1621
## 968  2020.1621
## 969  4449.3316
## 970  2708.5604
## 971  2708.5604
## 972  4449.3316
## 973  2020.1621
## 974  2708.5604
## 975  2708.5604
## 976  2708.5604
## 977  2708.5604
## 978  4476.1881
## 979  2708.5604
## 980  4449.3316
## 981  2708.5604
```

```
## 982  4449.3316
## 983  2708.5604
## 984  2708.5604
## 985  2708.5604
## 986  2708.5604
## 987  4449.3316
## 988  2481.2024
## 989  4476.1881
## 990  2481.2024
## 991  4476.1881
## 992  1062.8515
## 993  1062.8515
## 994  2481.2024
## 995  4476.1881
## 996  2481.2024
## 997  4476.1881
## 998  4449.3316
## 999  2481.2024
## 1000  137.4839
```

```r
print("...")
```

```
## [1] "..."
```

Convert this design matrix into Q, an orthonormal matrix.

```r
#TO-DO
Q = matrix(NA, nrow=nrow(X_diamonds), ncol=ncol(X_diamonds))

Q[,1] = X_diamonds[,1]

for(j in 2:ncol(X_diamonds)){
  Q[,j] = X_diamonds[,j]
  for (k in 1: (j-1)){
    q_k = Q[,k , drop=FALSE]
    Q[,j] = X_diamonds[,j] - (q_k %*% t(q_k)/sum(q_k^2)) %*% X_diamonds[,j]
  }
}

# NORMALIZE
for (j in 1:ncol(X_diamonds)){
  Q[,j] = Q[,j] / sqrt(sum(Q[,j]^2))
}
```

Project the y vector onto each column of the Q matrix and test if the sum of these projections is the same as yhat.

```r
#TO-DO
n = nrow(X_diamonds)
sum_proj_y = matrix(0, nrow=n, ncol=1)

for (j in 1:ncol(Q)) {
  q_j = Q[, j, drop=FALSE]
```

```
  sum_proj_y = sum_proj_y + (q_j %*% t(q_j)/sum(q_j^2)) %*% y_diamond
}


yhat_diamond - sum_proj_y
```

```
##              [,1]
## 1      -670.15854
## 2      -670.15854
## 3      -670.15854
## 4        72.18379
## 5      1134.59896
## 6      1134.59896
## 7        72.18379
## 8      -202.61394
## 9      -670.15854
## 10     -202.61394
## 11     1134.59896
## 12     1134.59896
## 13    -1003.83845
## 14     1134.59896
## 15     -670.15854
## 16     -670.15854
## 17       72.18379
## 18     1134.59896
## 19     1134.59896
## 20     1134.59896
## 21       72.18379
## 22     -670.15854
## 23     -202.61394
## 24     1134.59896
## 25     1134.59896
## 26     -108.90481
## 27       72.18379
## 28     1134.59896
## 29     1044.98173
## 30    -1003.83845
## 31    -1003.83845
## 32    -1003.83845
## 33     -670.15854
## 34     -670.15854
## 35     1044.98173
## 36    -1003.83845
## 37     -670.15854
## 38     -202.61394
## 39     1044.98173
## 40       72.18379
## 41       72.18379
## 42     1134.59896
## 43     1044.98173
## 44     1044.98173
## 45     -202.61394
## 46    -1003.83845
```

```
## 47    -202.61394
## 48    -202.61394
## 49    -670.15854
## 50    -202.61394
## 51   -1003.83845
## 52    -108.90481
## 53      72.18379
## 54    -670.15854
## 55    1044.98173
## 56      72.18379
## 57    1134.59896
## 58      72.18379
## 59      72.18379
## 60      72.18379
## 61      72.18379
## 62    1044.98173
## 63    1044.98173
## 64    1044.98173
## 65      72.18379
## 66    -108.90481
## 67      72.18379
## 68    -108.90481
## 69    -108.90481
## 70    -670.15854
## 71    1044.98173
## 72    -202.61394
## 73    -202.61394
## 74    -202.61394
## 75    -202.61394
## 76   -1003.83845
## 77    -670.15854
## 78    1044.98173
## 79    1044.98173
## 80    -670.15854
## 81    -670.15854
## 82    1044.98173
## 83    -670.15854
## 84      72.18379
## 85    -670.15854
## 86    -108.90481
## 87    -202.61394
## 88    -202.61394
## 89    -202.61394
## 90      72.18379
## 91    -670.15854
## 92    -670.15854
## 93    -108.90481
## 94    -670.15854
## 95    -108.90481
## 96    -670.15854
## 97   -1003.83845
## 98   -1003.83845
## 99    -670.15854
## 100   -202.61394
```

```
## 101   1044.98173
## 102   -670.15854
## 103   -108.90481
## 104   -108.90481
## 105     72.18379
## 106   -108.90481
## 107   -108.90481
## 108     72.18379
## 109  -1003.83845
## 110   -670.15854
## 111  -1003.83845
## 112   -670.15854
## 113     72.18379
## 114   -108.90481
## 115  -1003.83845
## 116  -1003.83845
## 117  -1003.83845
## 118   -108.90481
## 119   -670.15854
## 120  -1003.83845
## 121   1044.98173
## 122   -670.15854
## 123  -1003.83845
## 124  -1003.83845
## 125  -1003.83845
## 126  -1003.83845
## 127   -202.61394
## 128   1044.98173
## 129   -202.61394
## 130   -202.61394
## 131   -202.61394
## 132   1044.98173
## 133   1044.98173
## 134   -670.15854
## 135   -202.61394
## 136   -670.15854
## 137  -1003.83845
## 138  -1003.83845
## 139   -202.61394
## 140   -108.90481
## 141   -108.90481
## 142   -108.90481
## 143   1044.98173
## 144  -1003.83845
## 145   1044.98173
## 146   -202.61394
## 147   -108.90481
## 148   1044.98173
## 149   1044.98173
## 150   -670.15854
## 151   1044.98173
## 152     72.18379
## 153   1044.98173
## 154   -108.90481
```

```
## 155  -1003.83845
## 156   1044.98173
## 157  -1003.83845
## 158   -108.90481
## 159   -108.90481
## 160   1044.98173
## 161   -108.90481
## 162   -202.61394
## 163  -1003.83845
## 164   -108.90481
## 165  -1003.83845
## 166   -202.61394
## 167  -1003.83845
## 168   -108.90481
## 169  -1003.83845
## 170   -670.15854
## 171   1044.98173
## 172   1044.98173
## 173   1134.59896
## 174   -670.15854
## 175   -670.15854
## 176     72.18379
## 177  -1003.83845
## 178   -108.90481
## 179   -670.15854
## 180   -670.15854
## 181   -670.15854
## 182   -670.15854
## 183   -108.90481
## 184   -108.90481
## 185   -108.90481
## 186   -108.90481
## 187   1044.98173
## 188  -1003.83845
## 189  -1003.83845
## 190  -1003.83845
## 191  -1003.83845
## 192   -670.15854
## 193   -670.15854
## 194   -670.15854
## 195   -670.15854
## 196   -670.15854
## 197   -670.15854
## 198   -670.15854
## 199   -670.15854
## 200  -1003.83845
## 201   -670.15854
## 202   -670.15854
## 203   -670.15854
## 204   -670.15854
## 205   -202.61394
## 206  -1003.83845
## 207   -670.15854
## 208  -1003.83845
```

```
## 209   -202.61394
## 210   -670.15854
## 211  -1003.83845
## 212   -108.90481
## 213  -1003.83845
## 214   -108.90481
## 215   -108.90481
## 216  -1003.83845
## 217   -202.61394
## 218   -202.61394
## 219   -202.61394
## 220   1044.98173
## 221   -108.90481
## 222   -670.15854
## 223   -670.15854
## 224   1044.98173
## 225   1044.98173
## 226   1044.98173
## 227   1044.98173
## 228   -108.90481
## 229  -1003.83845
## 230  -1003.83845
## 231  -1003.83845
## 232   -202.61394
## 233  -1003.83845
## 234  -1003.83845
## 235   1044.98173
## 236   -202.61394
## 237     72.18379
## 238     72.18379
## 239   1044.98173
## 240   -108.90481
## 241  -1003.83845
## 242   -670.15854
## 243   -202.61394
## 244  -1003.83845
## 245   -670.15854
## 246   -670.15854
## 247   -670.15854
## 248   1134.59896
## 249   -108.90481
## 250   -670.15854
## 251   -108.90481
## 252   -108.90481
## 253   -670.15854
## 254   -670.15854
## 255   -202.61394
## 256   1134.59896
## 257   -108.90481
## 258     72.18379
## 259  -1003.83845
## 260  -1003.83845
## 261  -1003.83845
## 262     72.18379
```

```
## 263   -670.15854
## 264   -670.15854
## 265   -670.15854
## 266   -670.15854
## 267  -1003.83845
## 268  -1003.83845
## 269   -202.61394
## 270  -1003.83845
## 271   -108.90481
## 272   -670.15854
## 273   -670.15854
## 274     72.18379
## 275   -202.61394
## 276   -670.15854
## 277   -670.15854
## 278     72.18379
## 279  -1003.83845
## 280  -1003.83845
## 281   1044.98173
## 282     72.18379
## 283   -202.61394
## 284   -108.90481
## 285     72.18379
## 286   -670.15854
## 287   -202.61394
## 288   -670.15854
## 289   1044.98173
## 290   1044.98173
## 291   1044.98173
## 292  -1003.83845
## 293   1044.98173
## 294   -108.90481
## 295   -202.61394
## 296   -202.61394
## 297   -108.90481
## 298  -1003.83845
## 299   -670.15854
## 300   -202.61394
## 301     72.18379
## 302   -670.15854
## 303  -1003.83845
## 304   1044.98173
## 305   -108.90481
## 306     72.18379
## 307   -670.15854
## 308     72.18379
## 309   -108.90481
## 310   -108.90481
## 311   -202.61394
## 312   -108.90481
## 313     72.18379
## 314   -108.90481
## 315   -108.90481
## 316  -1003.83845
```

```
## 317  -1003.83845
## 318  -1003.83845
## 319   -202.61394
## 320  -1003.83845
## 321  -1003.83845
## 322  -1003.83845
## 323  -1003.83845
## 324   -108.90481
## 325   1134.59896
## 326   -108.90481
## 327  -1003.83845
## 328   -670.15854
## 329  -1003.83845
## 330  -1003.83845
## 331   -670.15854
## 332   -108.90481
## 333   -670.15854
## 334   -108.90481
## 335  -1003.83845
## 336  -1003.83845
## 337   -108.90481
## 338   -108.90481
## 339   -108.90481
## 340   -108.90481
## 341   -108.90481
## 342   1044.98173
## 343   -670.15854
## 344   -670.15854
## 345   -670.15854
## 346   1044.98173
## 347  -1003.83845
## 348  -1003.83845
## 349   -202.61394
## 350   1044.98173
## 351   1044.98173
## 352   1044.98173
## 353     72.18379
## 354  -1003.83845
## 355  -1003.83845
## 356   -670.15854
## 357   -670.15854
## 358  -1003.83845
## 359   -108.90481
## 360  -1003.83845
## 361   -108.90481
## 362   -108.90481
## 363   -202.61394
## 364  -1003.83845
## 365  -1003.83845
## 366  -1003.83845
## 367   1134.59896
## 368   -202.61394
## 369  -1003.83845
## 370   -108.90481
```

```
## 371   -108.90481
## 372   -670.15854
## 373   1044.98173
## 374  -1003.83845
## 375   1044.98173
## 376  -1003.83845
## 377  -1003.83845
## 378  -1003.83845
## 379  -1003.83845
## 380   1044.98173
## 381  -1003.83845
## 382   -108.90481
## 383  -1003.83845
## 384   1044.98173
## 385   1134.59896
## 386     72.18379
## 387   -108.90481
## 388  -1003.83845
## 389   -670.15854
## 390  -1003.83845
## 391     72.18379
## 392     72.18379
## 393     72.18379
## 394     72.18379
## 395     72.18379
## 396     72.18379
## 397   -202.61394
## 398   -202.61394
## 399   -202.61394
## 400   -202.61394
## 401   -202.61394
## 402   -202.61394
## 403   -202.61394
## 404     72.18379
## 405   -202.61394
## 406   -670.15854
## 407   -670.15854
## 408  -1003.83845
## 409   -202.61394
## 410   -202.61394
## 411     72.18379
## 412     72.18379
## 413   -108.90481
## 414  -1003.83845
## 415   -108.90481
## 416   -670.15854
## 417  -1003.83845
## 418     72.18379
## 419   -108.90481
## 420   -670.15854
## 421  -1003.83845
## 422  -1003.83845
## 423   1044.98173
## 424   1134.59896
```

```
## 425   -670.15854
## 426  -1003.83845
## 427   -108.90481
## 428   -670.15854
## 429     72.18379
## 430   -670.15854
## 431   -670.15854
## 432   -670.15854
## 433   1044.98173
## 434  -1003.83845
## 435   -108.90481
## 436   -202.61394
## 437   -202.61394
## 438   -202.61394
## 439   -202.61394
## 440   1134.59896
## 441  -1003.83845
## 442   -202.61394
## 443   -670.15854
## 444   -670.15854
## 445   -108.90481
## 446     72.18379
## 447   -108.90481
## 448   1044.98173
## 449  -1003.83845
## 450   -202.61394
## 451   -202.61394
## 452   -670.15854
## 453     72.18379
## 454   -670.15854
## 455   -670.15854
## 456  -1003.83845
## 457   -670.15854
## 458   -670.15854
## 459   -670.15854
## 460   -670.15854
## 461   1134.59896
## 462   -670.15854
## 463   -670.15854
## 464   -670.15854
## 465   -670.15854
## 466   -202.61394
## 467  -1003.83845
## 468   -202.61394
## 469   -670.15854
## 470   -670.15854
## 471  -1003.83845
## 472   -670.15854
## 473   -202.61394
## 474   -108.90481
## 475   -202.61394
## 476  -1003.83845
## 477  -1003.83845
## 478   -108.90481
```

```
## 479   -670.15854
## 480   -670.15854
## 481   -670.15854
## 482   -670.15854
## 483   -670.15854
## 484   -670.15854
## 485   -670.15854
## 486   1044.98173
## 487   -670.15854
## 488   -670.15854
## 489   -670.15854
## 490   -670.15854
## 491   -670.15854
## 492   -670.15854
## 493  -1003.83845
## 494     72.18379
## 495  -1003.83845
## 496   1044.98173
## 497   1044.98173
## 498   1044.98173
## 499   1044.98173
## 500   1134.59896
## 501   1044.98173
## 502   -670.15854
## 503   -670.15854
## 504   1044.98173
## 505   -670.15854
## 506   1044.98173
## 507   1044.98173
## 508   -670.15854
## 509  -1003.83845
## 510   1044.98173
## 511   1044.98173
## 512     72.18379
## 513   -670.15854
## 514   -670.15854
## 515   -202.61394
## 516   -670.15854
## 517   1044.98173
## 518     72.18379
## 519   -670.15854
## 520  -1003.83845
## 521   -670.15854
## 522   -108.90481
## 523   -202.61394
## 524   -108.90481
## 525   -202.61394
## 526   1134.59896
## 527   -202.61394
## 528   -670.15854
## 529   -670.15854
## 530     72.18379
## 531   -670.15854
## 532   1044.98173
```

```
## 533   -670.15854
## 534     72.18379
## 535  -1003.83845
## 536  -1003.83845
## 537   1044.98173
## 538   1044.98173
## 539   -202.61394
## 540   -202.61394
## 541   -202.61394
## 542   1044.98173
## 543   -202.61394
## 544  -1003.83845
## 545   -670.15854
## 546   -670.15854
## 547  -1003.83845
## 548   -670.15854
## 549   -670.15854
## 550   -670.15854
## 551   -202.61394
## 552  -1003.83845
## 553  -1003.83845
## 554  -1003.83845
## 555     72.18379
## 556   -108.90481
## 557   -670.15854
## 558   -670.15854
## 559   -670.15854
## 560   -670.15854
## 561   -108.90481
## 562  -1003.83845
## 563     72.18379
## 564   -108.90481
## 565   -108.90481
## 566   -202.61394
## 567  -1003.83845
## 568   1134.59896
## 569     72.18379
## 570   -670.15854
## 571   1044.98173
## 572   1044.98173
## 573  -1003.83845
## 574   -202.61394
## 575  -1003.83845
## 576   -108.90481
## 577   -202.61394
## 578   1044.98173
## 579  -1003.83845
## 580     72.18379
## 581   1134.59896
## 582   -108.90481
## 583   -670.15854
## 584  -1003.83845
## 585  -1003.83845
## 586   -670.15854
```

```
## 587   -670.15854
## 588   -670.15854
## 589  -1003.83845
## 590   -202.61394
## 591   -670.15854
## 592   -670.15854
## 593   -670.15854
## 594  -1003.83845
## 595   -202.61394
## 596  -1003.83845
## 597  -1003.83845
## 598  -1003.83845
## 599  -1003.83845
## 600   -108.90481
## 601  -1003.83845
## 602   -670.15854
## 603  -1003.83845
## 604  -1003.83845
## 605   -108.90481
## 606   -670.15854
## 607  -1003.83845
## 608  -1003.83845
## 609  -1003.83845
## 610  -1003.83845
## 611  -1003.83845
## 612  -1003.83845
## 613  -1003.83845
## 614   -108.90481
## 615   -670.15854
## 616   -108.90481
## 617   -108.90481
## 618  -1003.83845
## 619   -108.90481
## 620  -1003.83845
## 621   -202.61394
## 622  -1003.83845
## 623  -1003.83845
## 624  -1003.83845
## 625   1044.98173
## 626  -1003.83845
## 627   -202.61394
## 628  -1003.83845
## 629   1044.98173
## 630   1044.98173
## 631  -1003.83845
## 632  -1003.83845
## 633  -1003.83845
## 634  -1003.83845
## 635  -1003.83845
## 636   -108.90481
## 637   1044.98173
## 638  -1003.83845
## 639   -108.90481
## 640     72.18379
```

```
## 641  -1003.83845
## 642   -670.15854
## 643   -670.15854
## 644   -670.15854
## 645  -1003.83845
## 646   1044.98173
## 647   -108.90481
## 648  -1003.83845
## 649   -670.15854
## 650  -1003.83845
## 651   -108.90481
## 652   -670.15854
## 653   -670.15854
## 654     72.18379
## 655     72.18379
## 656   -202.61394
## 657   -670.15854
## 658   -670.15854
## 659   -202.61394
## 660   -202.61394
## 661   1044.98173
## 662  -1003.83845
## 663   -202.61394
## 664   -202.61394
## 665   -202.61394
## 666   -670.15854
## 667   -670.15854
## 668  -1003.83845
## 669   -670.15854
## 670   -108.90481
## 671   -108.90481
## 672   -202.61394
## 673   -670.15854
## 674   1044.98173
## 675   -108.90481
## 676   1044.98173
## 677   1044.98173
## 678   -670.15854
## 679   -670.15854
## 680   -670.15854
## 681  -1003.83845
## 682   1134.59896
## 683   -202.61394
## 684  -1003.83845
## 685   1044.98173
## 686  -1003.83845
## 687   -202.61394
## 688  -1003.83845
## 689   -670.15854
## 690     72.18379
## 691   -108.90481
## 692   -670.15854
## 693     72.18379
## 694  -1003.83845
```

```
## 695   -108.90481
## 696   1044.98173
## 697   1044.98173
## 698   -670.15854
## 699     72.18379
## 700   -202.61394
## 701   -670.15854
## 702   -202.61394
## 703   -670.15854
## 704   1134.59896
## 705   1134.59896
## 706   1134.59896
## 707   -202.61394
## 708   -670.15854
## 709  -1003.83845
## 710  -1003.83845
## 711   1134.59896
## 712   1044.98173
## 713   -670.15854
## 714     72.18379
## 715   -108.90481
## 716   -202.61394
## 717  -1003.83845
## 718   -108.90481
## 719  -1003.83845
## 720     72.18379
## 721   -670.15854
## 722   -670.15854
## 723   -202.61394
## 724   -108.90481
## 725   -670.15854
## 726   -670.15854
## 727   -670.15854
## 728   -108.90481
## 729   -670.15854
## 730   -670.15854
## 731     72.18379
## 732     72.18379
## 733     72.18379
## 734     72.18379
## 735     72.18379
## 736     72.18379
## 737   -108.90481
## 738   -202.61394
## 739   -202.61394
## 740   -108.90481
## 741  -1003.83845
## 742   -108.90481
## 743   -108.90481
## 744  -1003.83845
## 745   1134.59896
## 746   1134.59896
## 747   1134.59896
## 748     72.18379
```

```
## 749     72.18379
## 750     72.18379
## 751   -202.61394
## 752  -1003.83845
## 753  -1003.83845
## 754   -670.15854
## 755   -108.90481
## 756   -670.15854
## 757     72.18379
## 758   -670.15854
## 759   -108.90481
## 760  -1003.83845
## 761  -1003.83845
## 762   -670.15854
## 763   -670.15854
## 764   -670.15854
## 765   -670.15854
## 766   1134.59896
## 767   -670.15854
## 768   1044.98173
## 769   1044.98173
## 770   -670.15854
## 771   -108.90481
## 772   1044.98173
## 773   1044.98173
## 774   1044.98173
## 775  -1003.83845
## 776   -108.90481
## 777   1044.98173
## 778  -1003.83845
## 779   -670.15854
## 780  -1003.83845
## 781   1134.59896
## 782   -670.15854
## 783   1044.98173
## 784  -1003.83845
## 785   1044.98173
## 786   -108.90481
## 787   -108.90481
## 788     72.18379
## 789  -1003.83845
## 790   -670.15854
## 791   1044.98173
## 792   1044.98173
## 793  -1003.83845
## 794   1044.98173
## 795  -1003.83845
## 796  -1003.83845
## 797   -202.61394
## 798   -670.15854
## 799   -202.61394
## 800   -670.15854
## 801   -670.15854
## 802   -670.15854
```

```
## 803   -670.15854
## 804     72.18379
## 805   1044.98173
## 806   -108.90481
## 807  -1003.83845
## 808   1044.98173
## 809   1044.98173
## 810   1044.98173
## 811   1044.98173
## 812   1044.98173
## 813   1044.98173
## 814   1134.59896
## 815   1134.59896
## 816   -202.61394
## 817   -670.15854
## 818  -1003.83845
## 819   -670.15854
## 820   -670.15854
## 821   -670.15854
## 822     72.18379
## 823   1044.98173
## 824   1044.98173
## 825   1044.98173
## 826   -670.15854
## 827   -202.61394
## 828     72.18379
## 829   -202.61394
## 830   1044.98173
## 831   -202.61394
## 832   -202.61394
## 833  -1003.83845
## 834  -1003.83845
## 835  -1003.83845
## 836  -1003.83845
## 837  -1003.83845
## 838  -1003.83845
## 839   -108.90481
## 840  -1003.83845
## 841     72.18379
## 842   -108.90481
## 843   -670.15854
## 844   -670.15854
## 845   1044.98173
## 846   -670.15854
## 847   -108.90481
## 848   -108.90481
## 849   -202.61394
## 850  -1003.83845
## 851   -670.15854
## 852   -670.15854
## 853   -108.90481
## 854   -202.61394
## 855  -1003.83845
## 856   -202.61394
```

```
## 857   -670.15854
## 858     72.18379
## 859   -670.15854
## 860   -670.15854
## 861   1134.59896
## 862     72.18379
## 863   -108.90481
## 864   1044.98173
## 865   1044.98173
## 866     72.18379
## 867   -108.90481
## 868   1044.98173
## 869   -670.15854
## 870   1044.98173
## 871   1044.98173
## 872   1044.98173
## 873   -670.15854
## 874   -670.15854
## 875   -108.90481
## 876  -1003.83845
## 877   -670.15854
## 878   -670.15854
## 879     72.18379
## 880   -202.61394
## 881     72.18379
## 882   1134.59896
## 883   1134.59896
## 884     72.18379
## 885     72.18379
## 886   1134.59896
## 887  -1003.83845
## 888   1044.98173
## 889  -1003.83845
## 890  -1003.83845
## 891   -108.90481
## 892  -1003.83845
## 893  -1003.83845
## 894  -1003.83845
## 895   -108.90481
## 896   -202.61394
## 897   -202.61394
## 898   1134.59896
## 899     72.18379
## 900   -202.61394
## 901   -670.15854
## 902   -670.15854
## 903   -202.61394
## 904   -670.15854
## 905   -108.90481
## 906   -202.61394
## 907  -1003.83845
## 908  -1003.83845
## 909  -1003.83845
## 910   1044.98173
```

```
## 911     72.18379
## 912  -1003.83845
## 913  -1003.83845
## 914   -670.15854
## 915  -1003.83845
## 916  -1003.83845
## 917   -108.90481
## 918   -108.90481
## 919   1134.59896
## 920  -1003.83845
## 921  -1003.83845
## 922   1134.59896
## 923   1044.98173
## 924   -202.61394
## 925   -670.15854
## 926   -202.61394
## 927   -202.61394
## 928  -1003.83845
## 929  -1003.83845
## 930  -1003.83845
## 931  -1003.83845
## 932   1044.98173
## 933   -202.61394
## 934   -202.61394
## 935     72.18379
## 936  -1003.83845
## 937     72.18379
## 938   1044.98173
## 939   -670.15854
## 940   1044.98173
## 941   -670.15854
## 942   -670.15854
## 943   1044.98173
## 944   -670.15854
## 945     72.18379
## 946   -108.90481
## 947   -670.15854
## 948  -1003.83845
## 949   -202.61394
## 950  -1003.83845
## 951   -670.15854
## 952   -670.15854
## 953  -1003.83845
## 954  -1003.83845
## 955   -670.15854
## 956  -1003.83845
## 957  -1003.83845
## 958  -1003.83845
## 959  -1003.83845
## 960   -108.90481
## 961   -670.15854
## 962  -1003.83845
## 963   -670.15854
## 964   -670.15854
```

```
## 965   1044.98173
## 966   1044.98173
## 967   -202.61394
## 968   -202.61394
## 969  -1003.83845
## 970   -108.90481
## 971   -108.90481
## 972  -1003.83845
## 973   -202.61394
## 974   -108.90481
## 975   -108.90481
## 976   -108.90481
## 977   -108.90481
## 978   -670.15854
## 979   -108.90481
## 980  -1003.83845
## 981   -108.90481
## 982  -1003.83845
## 983   -108.90481
## 984   -108.90481
## 985   -108.90481
## 986   -108.90481
## 987  -1003.83845
## 988   1044.98173
## 989   -670.15854
## 990   1044.98173
## 991   -670.15854
## 992     72.18379
## 993     72.18379
## 994   1044.98173
## 995   -670.15854
## 996   1044.98173
## 997   -670.15854
## 998  -1003.83845
## 999   1044.98173
## 1000  1134.59896
```

```r
print("...")
```

```
## [1] "..."
```

Find linear OLS estimates if Q is used as the design matrix using the `lm` method. Is the OLS solution the same as the OLS solution for X?

```r
#TO-DO
model_vanilla = lm(y_diamond ~ 0 + X_diamonds)
b = coef(model_vanilla)

model_ortho = lm(y_diamond ~ 0 + Q)
b_q = coef(model_ortho)
print(b)
```

```
## X_diamonds(Intercept)     X_diamondscolor.L     X_diamondscolor.Q
```

```
##           2378.76684           -640.05819           -259.98348
##      X_diamondscolor.C     X_diamondscolor^4     X_diamondscolor^5
##            153.23342            223.99725            -16.12721
##      X_diamondscolor^6
##             36.89513
```

```r
print(b_q)
```

```
##          Q1          Q2          Q3          Q4          Q5          Q6          Q7
## 77671.2630 -5715.4205 -3299.1850   887.8146  2685.8521  -310.5586   452.0611
```

Use the predict function and ensure that the predicted values are the same for both linear models: the one created with X as its design matrix and the one created with Q as its design matrix.

```r
#TO-DO
y_hat_vanilla = predict(model_vanilla, data.frame(X_diamonds))
y_hat_ortho = predict(model_ortho, data.frame(Q))


table(sum(abs(y_hat_vanilla - y_hat_ortho)))
```

```
##
## 2.55363374890294e-09
##                    1
```

Clear the workspace and load the boston housing data and extract X and y. The dimensions are n = 506 and p = 13. Create a matrix that is (p + 1) x (p + 1) full of NA's. Label the columns the same columns as X. Do not label the rows. For the first row, find the OLS estimate of the y regressed on the first column only and put that in the first entry. For the second row, find the OLS estimates of the y regressed on the first and second columns of X only and put them in the first and second entries. For the third row, find the OLS estimates of the y regressed on the first, second and third columns of X only and put them in the first, second and third entries, etc. For the last row, fill it with the full OLS estimates.

```r
#TO-DO
rm(list = ls())

# Load the MASS package
library(MASS)
# Load the Boston housing data
data(Boston)

# EXTRACT X AND y
X = Boston[, !(names(Boston) %in% 'medv')]
y = Boston$medv

# Extract Dimensions
n = nrow(X)
p = ncol(X)

# Create NA matrix of p+1 dim
na_matrix = matrix(NA, nrow=p+1, ncol=p+1)
```

```r
# rename matrix colnames
col_names = colnames(X)
colnames(na_matrix) = c("Intercept", col_names)


# Sequentially regress y on an increasing number of predictors
for (i in 1:p) {
  # Regress y on the first i predictors
  X_sub = X[, 1:i, drop = FALSE] # TAKE ONE COLUMN AT A TIME
  model = lm(y ~ ., data = as.data.frame(X_sub)) # FIT A MODEL USINGTHE COLUMNS REMOVED

  # i + 1 => Accounts for intercept
  # Fill out per column
  na_matrix[i + 1, 1:(i + 1)] = coef(model)
}


# Regress y on all predictors for the last row
model_full = lm(y ~ ., data = as.data.frame(X))
na_matrix[p + 1, ] = coef(model_full)

head(na_matrix)
```

```
##      Intercept       crim          zn      indus     chas       nox rm age dis
## [1,]        NA         NA          NA         NA       NA        NA NA  NA  NA
## [2,]  24.03311 -0.4151903          NA         NA       NA        NA NA  NA  NA
## [3,]  22.48563 -0.3520783 0.11610909         NA       NA        NA NA  NA  NA
## [4,]  27.39465 -0.2486283 0.05850082 -0.4155778       NA        NA NA  NA  NA
## [5,]  27.11280 -0.2287981 0.05928665 -0.4403251 6.894059        NA NA  NA  NA
## [6,]  29.48994 -0.2185190 0.05511047 -0.3834805 7.026223 -5.424659 NA  NA  NA
##      rad tax ptratio black lstat
## [1,]  NA  NA      NA    NA    NA
## [2,]  NA  NA      NA    NA    NA
## [3,]  NA  NA      NA    NA    NA
## [4,]  NA  NA      NA    NA    NA
## [5,]  NA  NA      NA    NA    NA
## [6,]  NA  NA      NA    NA    NA
```

Why are the estimates changing from row to row as you add in more predictors?

The line fitted has to adjust, with more features to consider the fit will be different leading to different coefficients.

Create a vector of length p+1 and compute the R^2 values for each of the above models.

```r
#TO-DO
r_squared = numeric(p + 1)

# Loop through each subset of predictors and compute R^2
for (i in 1:p) {
  X_sub <- X[, 1:i, drop = FALSE]
```

```r
  model <- lm(y ~ ., data = as.data.frame(X_sub))  # Fit the model

  # Calculate R^2 and store it
  r_squared[i + 1] <- summary(model)$r.squared
}

model_full <- lm(y ~ ., data = Boston)  # Fit the full model
r_squared[1] <- summary(model_full)$r.squared  # Store
```

```
## Warning in summary.lm(model_full): essentially perfect fit: summary may be
## unreliable
```

```r
print(r_squared)
```

```
##  [1] 1.0000000 0.1507805 0.2339884 0.2937136 0.3295277 0.3313127 0.5873770
##  [8] 0.5894902 0.6311488 0.6319479 0.6396628 0.6703141 0.6842043 0.7406427
```

Is R^2 monotonically increasing? Why?

It is increasing because the more features in our model seems to help the model understand the relationship by accounting for the variance. The more features the better we can fit to our data

Create a 2x2 matrix with the first column 1's and the next column iid normals. Find the absolute value of the angle (in degrees, not radians) between the two columns in absolute difference from 90 degrees.

```r
n = 100

X = matrix(rnorm(2 * n), ncol = 2)
acos(t(X[,1]) %*% X[,2] / sqrt(sum(X[, 1]^2) * sum(X[, 2]^2))) * 180 / pi
```

```
##          [,1]
## [1,] 82.10683
```

Repeat this exercise `Nsim = 1e5` times and report the average absolute angle.

```r
Nsim = 1e5
n = 100
angles = numeric(Nsim)

for (i in 1:Nsim) {
  X = matrix(rnorm(2 * n), ncol = 2) # Create random matrix of size (2*n)(2)
  angle = acos(t(X[,1]) %*% X[,2] / sqrt(sum(X[, 1]^2) * sum(X[, 2]^2))) * 180 / pi # This line calcula
  angles[i] = abs(angle - 90)
}

average_angle = mean(angles)
print(average_angle)
```

```
## [1] 4.6029
```

Create a n x 2 matrix with the first column 1's and the next column iid normals. Find the absolute value of the angle (in degrees, not radians) between the two columns. For n = 10, 50, 100, 200, 500, 1000, report the average absolute angle over `Nsim = 1e5` simulations.

```
#TO-DO
n_values = c(10, 50, 100, 200, 500, 1000)
average_angles = numeric(length(n_values))

for (i in seq_along(n_values)) {
  n = n_values[i]
  angles = numeric(Nsim)

  for (j in 1:Nsim) {
    X = cbind(1, rnorm(n))  # Create the n x 2 matrix with 1's in the first column and iid normals in t
    angle = acos(t(X[,1]) %*% X[,2] / sqrt(sum(X[, 1]^2) * sum(X[, 2]^2))) * 180 / pi
    angles[j] <- abs(angle)  # Absolute value of the angle
  }

  average_angles[i] <- mean(angles)
}

# Print the results
result = data.frame(n_values, average_angles)
print(result)
```

```
##   n_values average_angles
## 1       10       89.92826
## 2       50       89.98565
## 3      100       89.95665
## 4      200       89.99340
## 5      500       89.98811
## 6     1000       89.99306
```

What is this absolute angle difference from 90 degrees converging to? Why does this make sense?

the convergence of the average absolute angle to 0 as n increases makes sense because it reflects the fact that the angle between the two columns of the matrix tends to be close to 90 degrees when n is large.