Onur Ordulu
2024808018

IE 582-HW 1

I have prepared the R script files with the necessary comments. I am preparing this pdf file in order to include the visual aids. Below, you will see how I formulized my solution in R through screenshots from my script, and then I will move on to comment on my findings.

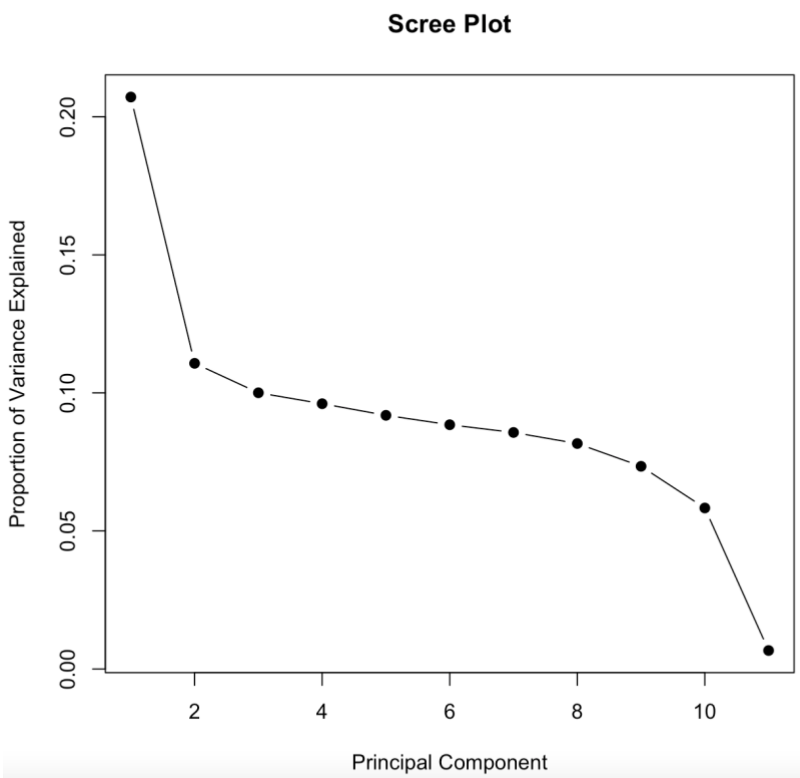
```
wd <- "/Users/oordulu/Documents/IE582/HW1/"
setwd(wd)
data_input<-read.csv("hw1_input.csv")

#Input data is imported. The PCA will be conducted on this data.
#I should scale all variables.

standardized_data=scale(data_input)
head(standardized_data)

colMeans(standardized_data)  # Should be approximately 0
apply(standardized_data, 2, sd)  # Should be approximately 1
```

```
> colMeans(standardized_data)  # Should be approximately 0
                        length.of.patch                          width.of.patch
                           1.003616e-16                            4.287696e-17
                        height.of.patch                     height.of.substrate
                           5.424953e-18                            5.277164e-17
            height.of.solder.resist.layer                     radius.of.the.probe
                          -1.300322e-16                           -1.492831e-16
                                  c_pad                               c_antipad
                          -6.046750e-17                            1.201938e-16
                                c_probe           dielectric.constant.of.substrate
                          -5.427206e-17                            1.392307e-16
dielectric.constant.of.solder.resist.layer
                           9.732475e-17
> apply(standardized_data, 2, sd)  # Should be approximately 1
                        length.of.patch                          width.of.patch
                                      1                                       1
                        height.of.patch                     height.of.substrate
                                      1                                       1
            height.of.solder.resist.layer                     radius.of.the.probe
                                      1                                       1
                                  c_pad                               c_antipad
                                      1                                       1
                                c_probe           dielectric.constant.of.substrate
                                      1                                       1
dielectric.constant.of.solder.resist.layer
                                      1
>
```

```r
# Answer to 3.1.1. Perform PCA. It is possible to reduce complexity of tha data
#with the cost of losing information to a great extent as it is shown in the
#next lines. Many PCs must be included in order to be able to explain, say,
#80% of the variance (8 PCs).
pca_result <- prcomp(standardized_data, scale. = FALSE)  # scale. = FALSE because data is already standardized

# View summary of PCA
summary(pca_result)
```

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8    PC9   PC10    PC11
Standard deviation     1.5095 1.1035 1.0490 1.02801 1.00528 0.98643 0.97070 0.94771 0.8985 0.8008 0.27132
Proportion of Variance 0.2072 0.1107 0.1000 0.09607 0.09187 0.08846 0.08566 0.08165 0.0734 0.0583 0.00669
Cumulative Proportion  0.2072 0.3179 0.4179 0.51397 0.60584 0.69430 0.77996 0.86161 0.9350 0.9933 1.00000
```

```r
#3.1.2: Proportion of variance each PCA covers can be seen in the second row of summary(pca_result)
#Example: It can be seen that the first 2 principal components cover 31.8% of the all variance.
#Then if picked up to 2 PCs, the dimensionality is reduced to 2 from 11 and
#31.8% of the variance is still explainable after the analyses.

explained_variance <- pca_result$sdev^2
explained_variance_ratio <- explained_variance / sum(explained_variance)

# Scree plot
plot(explained_variance_ratio, type = "b", pch = 19,
     xlab = "Principal Component", ylab = "Proportion of Variance Explained",
     main = "Scree Plot")
```



```r
# Absolute loadings for the first PC
loadings <- pca_result$rotation
abs_loadings_pc1 <- abs(loadings[, 1])

# Sort by contribution to PC1
key_variables_pc1 <- sort(abs_loadings_pc1, decreasing = TRUE)
print(key_variables_pc1)

#The importance of input variables can be determined by their absolute value in
#the first PC (the one that explains the most variance). Variables that have
#large coefficients in this PC can be stated to be critical for determination
#of S11 value. It is seen that width.of.patch, height.of.substrate,
#dielectric.constant.of.substrate are the 3 most important input variables
#according to PCA.
```

```
> print(key_variables_pc1)
                     width.of.patch                height.of.substrate    dielectric.constant.of.substrate
                         0.62417829                         0.62413737                          0.44510015
                    length.of.patch                    height.of.patch                               c_pad
                         0.10132741                         0.07105448                          0.05422779
                            c_probe dielectric.constant.of.solder.resist.layer             radius.of.the.probe
                         0.03921288                         0.03842190                          0.02599671
         height.of.solder.resist.layer                          c_antipad
                         0.02104923                         0.01872008
```

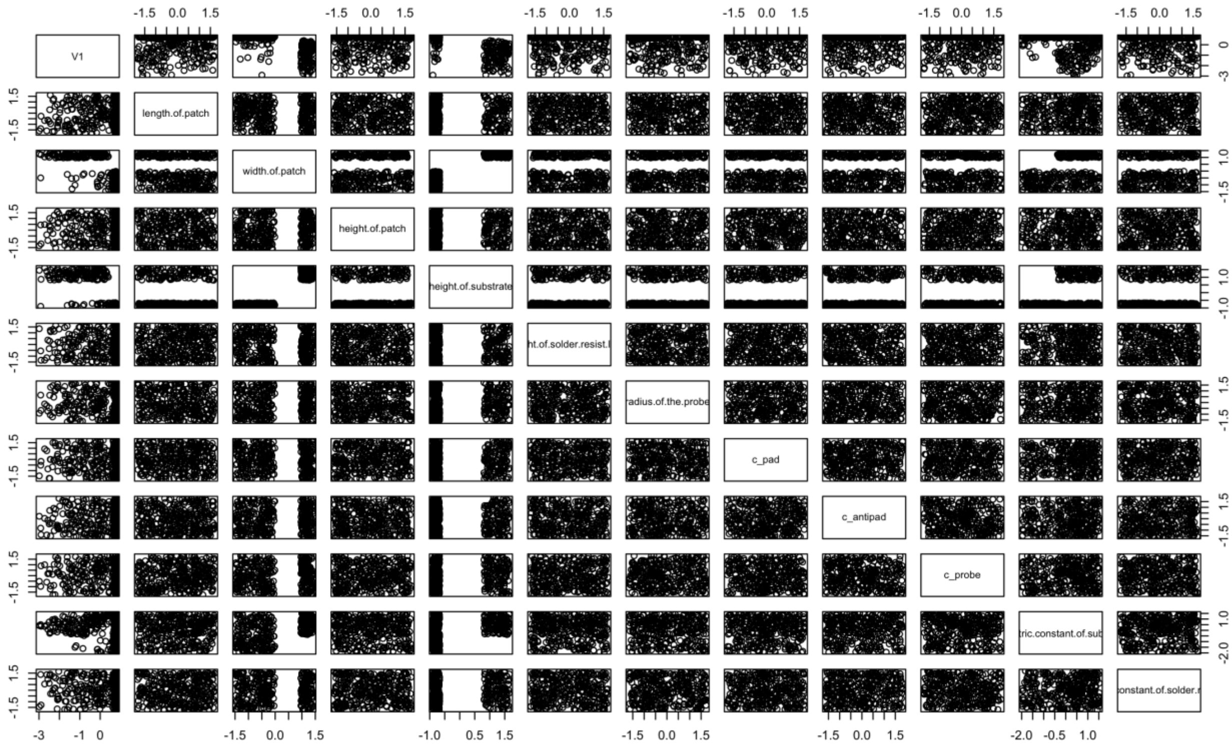
```r
#In general, PCA analysis by itself is not very useful in reducing the
#complexity of the interaction between S11 and the input variables since it
#costs large portion of variance to be unexplainable. However, it is useful in
#terms of understanding which input variables are more important compared to
#rest of the variables. I have conducted unsupervised PCA since that looked
#more meaningful to me although I searched the literature and PCA involving
#the response variable is also a thing. Thus, the conclusions I can draw
#from PCA is limited to identifying critical inputs, which was already done.

#Now let's start the regression portion of the work.
data_real<-read.csv("hw1_real.csv")
data_imag<-read.csv("hw1_img.csv")

freq=seq(0, 200)

#Calculate magnitudes of s parameter.
squared_matrix1 <- data_real^2
squared_matrix2 <- data_imag^2
```

```
sum_of_squares <- squared_matrix1 + squared_matrix2

result_matrix <- sqrt(sum_of_squares)
plot(freq,t(result_matrix[3,]))

# Find the column index of the minimum value in each row which correspond to resonance frequency for each design.
min_columns <- apply(result_matrix, 1, function(row) which.min(row))

#Find resonance frequencies that repeat the most so that the regression is carried out on those.
sorted_counts <- sort(table(min_columns), decreasing = TRUE)
top_5 <- head(sorted_counts, 5)
print(top_5)

#Column numbers 201, 1, 115, 96 and 62 are found to be most repeated minimum entry columns across all designs.
#These column numbers correspond to frequencies 200, 0, 114, 95 and 61. 200 and 0 are minimum not because
#they are frequently the resonance frequency values but they are just the end values, that is why regression
#will not be made for those frequency values. Frequencies 114 and 95 are selected to carry out the regressions.

y_114=result_matrix[,115]
y_95=result_matrix[,96]
```

```
min_columns
201   1 115  96  62
 77  63   6   5   4
```

```
y114=scale(y_114)
y95=scale(y_95)

#Regression for magnitudes of s parameters.
#First let's set training and test sets

mydata=cbind(y114,standardized_data)
myobs<- sample(1:385,350);

mydata.tr <- mydata[myobs,];
mydata.te <- mydata[-myobs,]

rownames(mydata.tr)<- 1:350

mydata.tr_dataframe <- as.data.frame(mydata.tr)
mydata.te_dataframe <- as.data.frame(mydata.te)
```

```
#In doing regression, first I will get a linear regression model with only the meaningful variables
#(alpha=0.1). Then I will contain nonlinear terms of order 2 for those variables only.

quartz();plot(mydata.tr_dataframe)
#This plot is to see if there are any input variables that are correlated. The
#plot shows no clear correlation.
```



```
alt1<-lm(V1~length.of.patch + width.of.patch + height.of.patch +
          height.of.substrate + height.of.solder.resist.layer +
          radius.of.the.probe + c_pad + c_antipad + c_probe +
          dielectric.constant.of.substrate +
          dielectric.constant.of.solder.resist.layer,mydata.tr_dataframe)

summary(alt1)
#Eliminate variable with the most p-value until all remaining have p-values below 0.1.
```

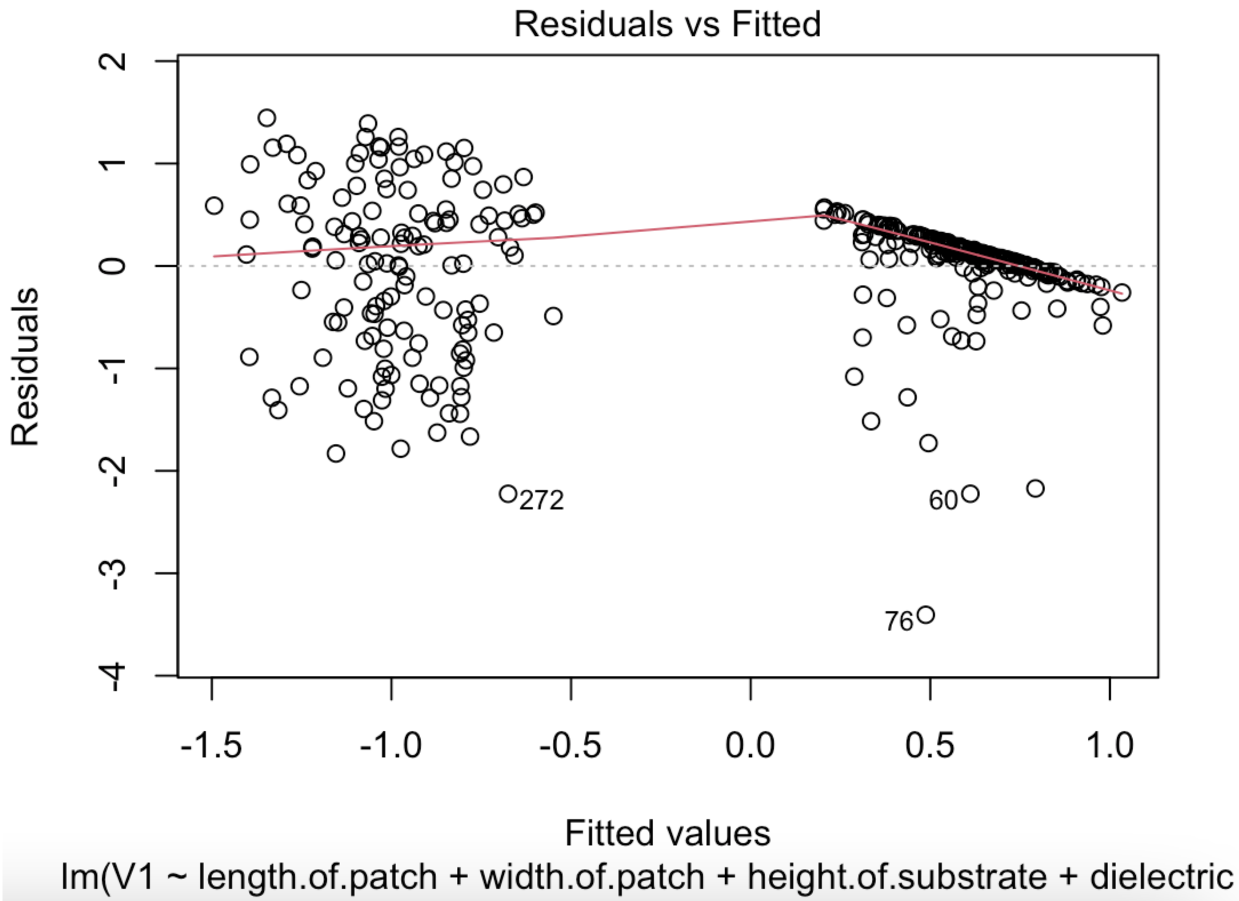
```
alt2<-lm(V1~length.of.patch + width.of.patch + height.of.patch +
          height.of.substrate + radius.of.the.probe + c_pad + c_antipad +
          c_probe + dielectric.constant.of.substrate +
          dielectric.constant.of.solder.resist.layer,mydata.tr_dataframe)

summary(alt2)
```

```
Call:
lm(formula = V1 ~ length.of.patch + width.of.patch + height.of.patch +
    height.of.substrate + height.of.solder.resist.layer + radius.of.the.probe +
    c_pad + c_antipad + c_probe + dielectric.constant.of.substrate +
    dielectric.constant.of.solder.resist.layer, data = mydata.tr_dataframe)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3670 -0.1450  0.0986  0.2890  1.4933

Coefficients:
                                            Estimate Std. Error t value Pr(>|t|)
(Intercept)                                -0.012602   0.034385  -0.366   0.7142
length.of.patch                             0.081532   0.035232   2.314   0.0213 *
width.of.patch                             -0.174549   0.090278  -1.933   0.0540 .
height.of.patch                             0.010415   0.034699   0.300   0.7642
height.of.substrate                        -0.654640   0.091065  -7.189 4.22e-12 ***
height.of.solder.resist.layer               0.004044   0.033936   0.119   0.9052
radius.of.the.probe                         0.020778   0.034825   0.597   0.5512
c_pad                                       0.022908   0.034697   0.660   0.5096
c_antipad                                   0.028601   0.034481   0.829   0.4074
c_probe                                     0.016675   0.035549   0.469   0.6393
dielectric.constant.of.substrate            0.098744   0.039288   2.513   0.0124 *
dielectric.constant.of.solder.resist.layer -0.079634   0.034421  -2.314   0.0213 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6429 on 338 degrees of freedom
Multiple R-squared:  0.6122,	Adjusted R-squared:  0.5996
F-statistic: 48.52 on 11 and 338 DF,  p-value: < 2.2e-16
```

This approach is carried out until all variables have p-values below 0.05.

```
alt7<-lm(V1~length.of.patch + width.of.patch + height.of.substrate +
           dielectric.constant.of.substrate +
           dielectric.constant.of.solder.resist.layer,mydata.tr_dataframe)
summary(alt7)
plot(alt7,which=1)
plot(alt7,which=2)
acf(alt7$residuals)

#This is the final model with only the linear terms.
#All p-values of variables are below alpha at linear regression model alternative
#7. Residuals vs. Fitted values grapgh shows existence of unexplained behaviour since
#residuals are not independently distributed between 0 and 1. That is why, nonlinear
#terms now will be involved.
```

```
Call:
lm(formula = V1 ~ length.of.patch + width.of.patch + height.of.substrate +
    dielectric.constant.of.substrate + dielectric.constant.of.solder.resist.layer,
    data = mydata.tr_dataframe)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4053 -0.1477  0.0997  0.2919  1.4457

Coefficients:
                                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                               -0.01307    0.03417  -0.383   0.7022
length.of.patch                            0.07914    0.03475   2.277   0.0234 *
width.of.patch                            -0.17424    0.08953  -1.946   0.0524 .
height.of.substrate                       -0.65357    0.09040  -7.230 3.16e-12 ***
dielectric.constant.of.substrate           0.09470    0.03878   2.442   0.0151 *
dielectric.constant.of.solder.resist.layer -0.07797    0.03412  -2.285   0.0229 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.639 on 344 degrees of freedom
Multiple R-squared:  0.6101,    Adjusted R-squared:  0.6045
F-statistic: 107.7 on 5 and 344 DF,  p-value: < 2.2e-16
```


Residuals vs Fitted
Residuals
2
1
0
-1
-2
-3
-4
272
60
76
-1.5
-1.0
-0.5
0.0
0.5
1.0
Fitted values
lm(V1 ~ length.of.patch + width.of.patch + height.of.substrate + dielectric .

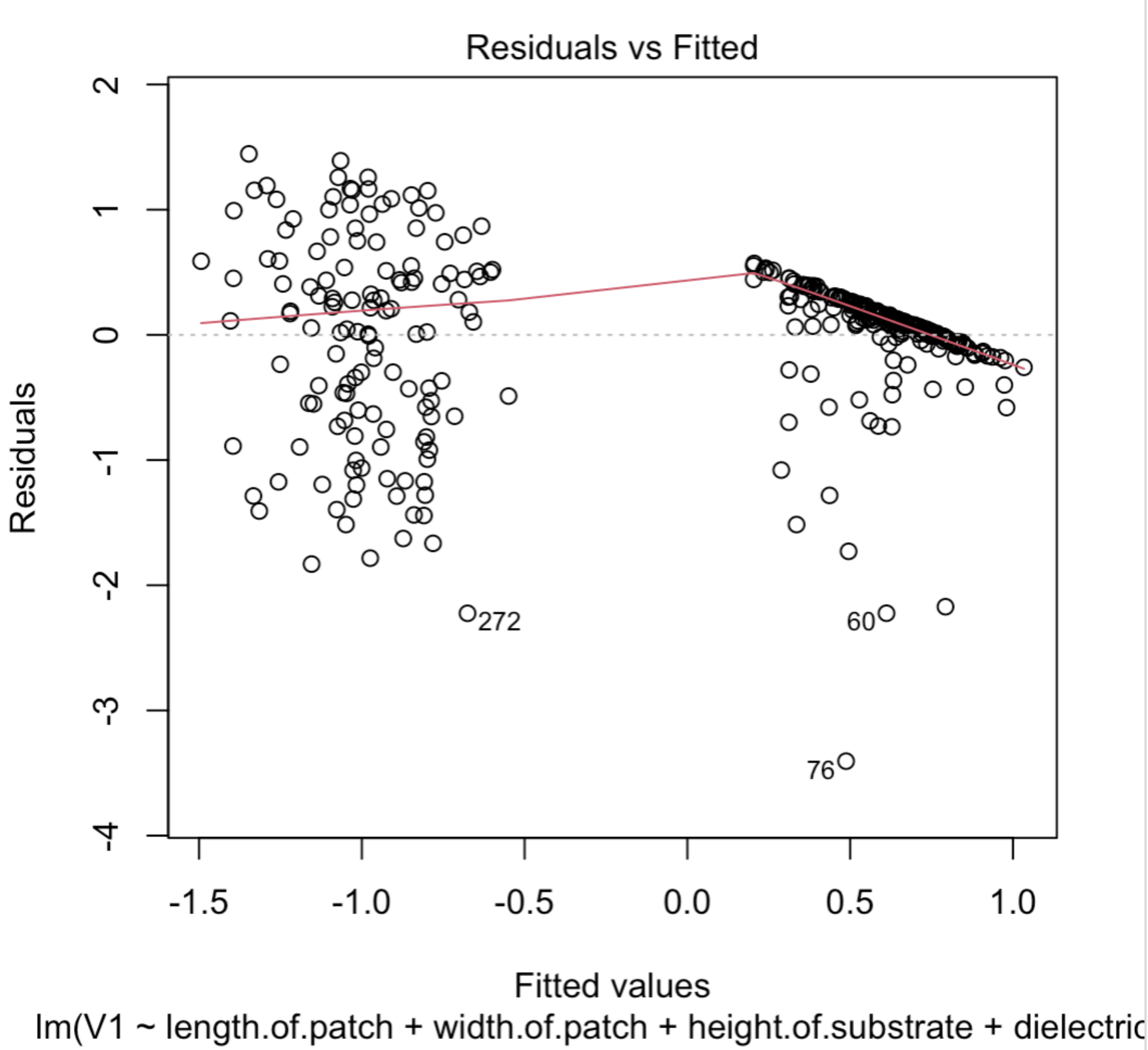
```
Call:
lm(formula = V1 ~ length.of.patch + width.of.patch + height.of.patch +
    height.of.substrate + radius.of.the.probe + c_pad + c_antipad +
    c_probe + dielectric.constant.of.substrate + dielectric.constant.of.solder.resist.layer,
    data = mydata.tr_dataframe)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3686 -0.1431  0.1000  0.2934  1.4900

Coefficients:
                                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                                -0.01262    0.03433  -0.368   0.7134
length.of.patch                             0.08172    0.03514   2.325   0.0206 *
width.of.patch                             -0.17487    0.09011  -1.941   0.0531 .
height.of.patch                             0.01057    0.03463   0.305   0.7605
height.of.substrate                        -0.65438    0.09091  -7.198 3.95e-12 ***
radius.of.the.probe                         0.02083    0.03477   0.599   0.5495
c_pad                                       0.02291    0.03465   0.661   0.5089
c_antipad                                   0.02861    0.03443   0.831   0.4065
c_probe                                     0.01687    0.03546   0.476   0.6346
dielectric.constant.of.substrate            0.09873    0.03923   2.517   0.0123 *
dielectric.constant.of.solder.resist.layer -0.07973    0.03436  -2.320   0.0209 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.642 on 339 degrees of freedom
Multiple R-squared:  0.6122,	Adjusted R-squared:  0.6008
F-statistic: 53.52 on 10 and 339 DF,  p-value: < 2.2e-16
```

```
altnonlin1<-lm(V1~(length.of.patch + width.of.patch + height.of.substrate +
                  dielectric.constant.of.substrate +
                  dielectric.constant.of.solder.resist.layer)^2 +
                  I(length.of.patch^2) + I(width.of.patch^2) +
                  I(height.of.substrate^2) + I(dielectric.constant.of.substrate^2) +
                  I(dielectric.constant.of.solder.resist.layer^2),
                  mydata.tr_dataframe)
summary(altnonlin1)
```

```r
#width.of.patch became insignificant.

altnonlin2<-lm(V1~(length.of.patch + height.of.substrate +
                 dielectric.constant.of.substrate +
                 dielectric.constant.of.solder.resist.layer)^2 +
                 I(length.of.patch^2) + I(height.of.substrate^2) +
                 I(dielectric.constant.of.substrate^2) +
                 I(dielectric.constant.of.solder.resist.layer^2),
                 mydata.tr_dataframe)
summary(altnonlin2)

#I can see that only height.of.substrate:dielectric.constant.of.substrate and
#height.of.substrate:dielectric.constant.of.solder.resist.layer interaction
#terms are important.
#Also, squared terms of dielectric.constant.of.substrate and dielectric.constant.of.solder.resist.layer
#are not needed.

altnonlin3<-lm(V1~length.of.patch + height.of.substrate +
                   dielectric.constant.of.substrate +
                   dielectric.constant.of.solder.resist.layer +
                 I(length.of.patch^2) + I(height.of.substrate^2) +
                 height.of.substrate:dielectric.constant.of.substrate +
                 height.of.substrate:dielectric.constant.of.solder.resist.layer,
               mydata.tr_dataframe)
summary(altnonlin3)
```

```
Call:
lm(formula = V1 ~ length.of.patch + height.of.substrate + dielectric.constant.of.substrate +
    dielectric.constant.of.solder.resist.layer + I(length.of.patch^2) +
    I(height.of.substrate^2) + height.of.substrate:dielectric.constant.of.substrate +
    height.of.substrate:dielectric.constant.of.solder.resist.layer,
    data = mydata.tr_dataframe)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3773 -0.1543  0.0782  0.3150  1.7087

Coefficients:
                                                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                                                    -0.47951    0.10959  -4.375 1.61e-05 ***
length.of.patch                                                 0.09590    0.03263   2.939 0.003518 **
height.of.substrate                                            -1.14309    0.06590 -17.346  < 2e-16 ***
dielectric.constant.of.substrate                                0.25295    0.04368   5.791 1.59e-08 ***
dielectric.constant.of.solder.resist.layer                     -0.07724    0.03182  -2.427 0.015727 *
I(length.of.patch^2)                                           -0.04698    0.03480  -1.350 0.177916
I(height.of.substrate^2)                                        0.37032    0.09825   3.769 0.000193 ***
height.of.substrate:dielectric.constant.of.substrate            0.31790    0.05045   6.301 9.12e-10 ***
height.of.substrate:dielectric.constant.of.solder.resist.layer  0.02588    0.03196   0.810 0.418639
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5952 on 341 degrees of freedom
Multiple R-squared:  0.6647,	Adjusted R-squared:  0.6568
F-statistic: 84.49 on 8 and 341 DF,  p-value: < 2.2e-16
```

```
altnonlin4<-lm(V1~length.of.patch + height.of.substrate +
                 dielectric.constant.of.substrate +
                 dielectric.constant.of.solder.resist.layer +
                 I(length.of.patch^2) + I(height.of.substrate^2) +
                 height.of.substrate:dielectric.constant.of.substrate,
               mydata.tr_dataframe)
summary(altnonlin4)
plot(altnonlin4,which=1)
plot(altnonlin4,which=2)
acf(altnonlin4$residuals)

#This is the final model including nonlinear terms.
```



```
AIC(alt1,alt2,alt3,alt4,alt5,alt6,alt7,altnonlin1,altnonlin2,altnonlin3,altnonlin4)
#Akeike value is like a goodness of fit value that takes degree of freedom involved
#into consideration. Minimum akeike is desired. Final nonlinear model is to be selected
#looking at this. It is also worth noting that every operation decreased AIC value.
#Looking at this is not sufficient. I should also look at test error.
```

```
           df      AIC
alt1       13 680.0847
alt2       12 678.6165
alt3       11 677.0960
alt4       10 675.5968
alt5        9 673.7196
alt6        8 672.2836
alt7        7 670.7559
altnonlin1 22 628.5422
altnonlin2 16 622.2869
altnonlin3 10 614.2979
altnonlin4  9 612.3296
```

```
errorTe_altnonlin4 <-  mydata.te_dataframe$V1 - predict(altnonlin4,new=mydata.te_dataframe) # error Testset
sum(errorTe_altnonlin4^2)/length(errorTe_altnonlin4) #   MSE test set

errorTe_alt7 <-  mydata.te_dataframe$V1 - predict(alt7,new=mydata.te_dataframe) # error Testset
sum(errorTe_alt7^2)/length(errorTe_alt7) #   MSE test set

errorTe_altnonlin3 <-  mydata.te_dataframe$V1 - predict(altnonlin3,new=mydata.te_dataframe) # error Testset
sum(errorTe_altnonlin3^2)/length(errorTe_altnonlin3) #   MSE test set
> sum(errorTe_altnonlin4^2)/length(errorTe_altnonlin4) #   MSE test set
[1] 0.5696322
> errorTe_alt7 <-  mydata.te_dataframe$V1 - predict(alt7,new=mydata.te_dataframe) # error Testset
> sum(errorTe_alt7^2)/length(errorTe_alt7) #   MSE test set
[1] 0.5740538
> errorTe_altnonlin3 <-  mydata.te_dataframe$V1 - predict(altnonlin3,new=mydata.te_dataframe) # error Testset
> sum(errorTe_altnonlin3^2)/length(errorTe_altnonlin3) #   MSE test set
[1] 0.5663291
```

```
#It is seen that altnonlin3 has less training error compared to altnonlin4, however,
#the decision should be done regarding the test errors, which is minimum for
#altnonlin4. That is why altnonlin4 is selected although clearly there is still behaviour
#must be incorporated for a model that is reliable.

#Until now, regression was made for s11 performance, which is described as magnitude of the s parameter.
#Let's if we can do regression to predict real and imaginry parts of the s parameters.
y_114_real=data_real[,115]
y_114_imag=data_imag[,115]

y114_real=scale(y_114_real)
y114_imag=scale(y_114_imag)

mydata_real=cbind(y114_real,standardized_data)
mydata_imag=cbind(y114_imag,standardized_data)

mydata_real.tr <- mydata_real[myobs,];
mydata_real.te <- mydata_real[-myobs,]
```

```
mydata_imag.tr <- mydata_imag[myobs,];
mydata_imag.te <- mydata_imag[-myobs,]

rownames(mydata_real.tr)<- 1:350
rownames(mydata_imag.tr)<- 1:350

mydata_real.tr_dataframe <- as.data.frame(mydata_real.tr)
mydata_real.te_dataframe <- as.data.frame(mydata_real.te)

mydata_imag.tr_dataframe <- as.data.frame(mydata_imag.tr)
mydata_imag.te_dataframe <- as.data.frame(mydata_imag.te)

alt1_real<-lm(V1~length.of.patch + width.of.patch + height.of.patch +
           height.of.substrate + height.of.solder.resist.layer +
           radius.of.the.probe + c_pad + c_antipad + c_probe +
           dielectric.constant.of.substrate +
           dielectric.constant.of.solder.resist.layer,mydata_real.tr_dataframe)
summary(alt1_real)
```

```
Call:
lm(formula = V1 ~ length.of.patch + width.of.patch + height.of.patch +
    height.of.substrate + height.of.solder.resist.layer + radius.of.the.probe +
    c_pad + c_antipad + c_probe + dielectric.constant.of.substrate +
    dielectric.constant.of.solder.resist.layer, data = mydata_real.tr_dataframe)

Residuals:
     Min       1Q   Median       3Q      Max
-1.02527 -0.25552 -0.07017  0.19753  2.08621

Coefficients:
                                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                                 0.0003093  0.0232746   0.013  0.98940
length.of.patch                             0.0516965  0.0238578   2.167  0.03094 *
width.of.patch                              0.1728751  0.0616128   2.806  0.00531 **
height.of.patch                             0.0115587  0.0233658   0.495  0.62114
height.of.substrate                         0.7337692  0.0623977  11.760  < 2e-16 ***
height.of.solder.resist.layer               0.0186856  0.0236518   0.790  0.43007
radius.of.the.probe                        -0.0431532  0.0233215  -1.850  0.06513 .
c_pad                                      -0.0040702  0.0236464  -0.172  0.86344
c_antipad                                  -0.0180965  0.0234527  -0.772  0.44088
c_probe                                     0.0459052  0.0237470   1.933  0.05406 .
dielectric.constant.of.substrate            0.0131331  0.0258968   0.507  0.61239
dielectric.constant.of.solder.resist.layer  0.0243644  0.0235847   1.033  0.30231
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4349 on 338 degrees of freedom
Multiple R-squared:  0.8149,    Adjusted R-squared:  0.8089
F-statistic: 135.3 on 11 and 338 DF,  p-value: < 2.2e-16
```

After standardization of all variables, the methodology is exactly the same for the regression analysis for the magnitude values of the s parameter.
I reduced number of varables until all have p-values below one and then included nonlinear terms (only squares and interaction by two variables). And finally, I decided on a final model looking at AIC values and test errors. Thus, I think it would be much more interesting if I jumped directly to discuss my results. The script file will be added to the website.

Below is the fundamental information about 6 regression models. 3 are for frequency 114 and the other 3 are for resonance frequency 95. 114 and 95 are found by calculating magnitudes for each design and then checking the common resonance frequencies through those magnitude values. Resonance frequencies are frequencies where return loss is at minimum. 3 models for each frequency are for their magnitude, real parts and imaginary parts.

```
summary(altnonlin4)          #Model for magnitudes of s11 at freq=114
summary(altnonlin3_real)     #Model for real parts of s11 at freq=114
summary(altnonlin2_imag)     #Model for imaginary parts of s11 at freq=114

summary(alt_nonlin2_2)       #Model for magnitudes of s11 at freq=95
summary(alt4_real_2)         #Model for real parts of s11 at freq=95
summary(altnonlin2_imag_2)   #Model for imaginary parts of s11 at freq=95
```

Full summaries of the first 3 models are put here as sample. The other three can be reached from the script.

```
> summary(altnonlin4)         #Model for magnitudes of s11 at freq=114

Call:
lm(formula = V1 ~ length.of.patch + height.of.substrate + dielectric.constant.of.substrate +
    dielectric.constant.of.solder.resist.layer + I(length.of.patch^2) +
    I(height.of.substrate^2) + height.of.substrate:dielectric.constant.of.substrate,
    data = mydata.tr_dataframe)

Residuals:
    Min      1Q  Median      3Q     Max
-3.3527 -0.1482  0.0901  0.2942  1.4792

Coefficients:
                                                      Estimate Std. Error t value Pr(>|t|)
(Intercept)                                           -0.51210    0.10926  -4.687 4.01e-06 ***
length.of.patch                                        0.11829    0.03243   3.648 0.000305 ***
height.of.substrate                                   -1.15359    0.06536 -17.650  < 2e-16 ***
dielectric.constant.of.substrate                       0.25737    0.04240   6.070 3.39e-09 ***
dielectric.constant.of.solder.resist.layer            -0.08229    0.03177  -2.591 0.009992 **
I(length.of.patch^2)                                  -0.05946    0.03456  -1.721 0.086190 .
I(height.of.substrate^2)                               0.41248    0.09760   4.226 3.05e-05 ***
height.of.substrate:dielectric.constant.of.substrate   0.33366    0.04852   6.877 2.92e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.5914 on 342 degrees of freedom
Multiple R-squared:  0.6649,    Adjusted R-squared:  0.658
F-statistic: 96.94 on 7 and 342 DF,  p-value: < 2.2e-16
```

```
> summary(altnonlin3_real)    #Model for real parts of s11 at freq=114

Call:
lm(formula = V1 ~ length.of.patch + width.of.patch + height.of.substrate +
    radius.of.the.probe + c_probe + length.of.patch:height.of.substrate,
    data = mydata_real.tr_dataframe)

Residuals:
     Min       1Q   Median       3Q      Max
-0.89344 -0.24649 -0.07952  0.15714  2.46349

Coefficients:
                                     Estimate Std. Error t value Pr(>|t|)
(Intercept)                         -0.001127   0.023678  -0.048  0.96207
length.of.patch                      0.046713   0.024165   1.933  0.05405 .
width.of.patch                       0.188184   0.062490   3.011  0.00279 **
height.of.substrate                  0.710115   0.062739  11.319  < 2e-16 ***
radius.of.the.probe                 -0.074182   0.023880  -3.106  0.00205 **
c_probe                              0.055635   0.024071   2.311  0.02141 *
length.of.patch:height.of.substrate  0.146126   0.023660   6.176 1.86e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4413 on 343 degrees of freedom
Multiple R-squared:  0.8062,    Adjusted R-squared:  0.8028
F-statistic: 237.8 on 6 and 343 DF,  p-value: < 2.2e-16
```

```
> summary(altnonlin2_imag)    #Model for imaginary parts of s11 at freq=114

Call:
lm(formula = V1 ~ length.of.patch + width.of.patch + height.of.substrate + 
    radius.of.the.probe + c_probe + dielectric.constant.of.substrate + 
    length.of.patch:c_probe + width.of.patch:c_probe, data = mydata_imag.tr_dataframe)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.73663 -0.42991  0.05604  0.50776  2.80019 

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)                      -0.02330    0.04375  -0.533 0.594717    
length.of.patch                   0.13896    0.04486   3.097 0.002115 ** 
width.of.patch                    0.35405    0.11591   3.055 0.002431 ** 
height.of.substrate              -0.78958    0.11689  -6.755 6.18e-11 ***
radius.of.the.probe              -0.12136    0.04416  -2.748 0.006308 ** 
c_probe                          -0.28198    0.04448  -6.339 7.33e-10 ***
dielectric.constant.of.substrate  0.19965    0.04892   4.081 5.58e-05 ***
length.of.patch:c_probe           0.16700    0.04711   3.545 0.000448 ***
width.of.patch:c_probe            0.07842    0.04532   1.730 0.084484 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8137 on 341 degrees of freedom
Multiple R-squared:  0.3518,    Adjusted R-squared:  0.3366 
F-statistic: 23.14 on 8 and 341 DF,  p-value: < 2.2e-16
```

Table 1. Models and Variables Used

Output	Model	Variables Involved from Input Data
Freq=114, Magnitude	altnonlin4	length.of.patch, height.of.substrate, dielectric.constant.of.substrate, dielectric.constant.of.solder.resist.layer
Freq=114, Real Part	altnonlin3_real	length.of.patch, width.of.patch, height.of.substrate, radius.of.the.probe, c_probe
Freq=114, Imaginary Part	altnonlin2_imag	length.of.patch, width.of.patch, height.of.substrate, radius.of.the.probe, c_probe, dielectric.constant.of.substrate
Freq=95, Magnitude	alt_nonlin2_2	width.of.patch, height.of.substrate
Freq=95, Real Part	alt4_real_2	width.of.patch, height.of.substrate, radius.of.the.probe, c_probe
Freq=95, Imaginary Part	altnonlin2_imag_2	length.of.patch, width.of.patch, height.of.substrate, radius.of.the.probe, c_antipad, c_probe, dielectric.constant.of.substrate

The variable height.of.substrate is present in all models, it can be stated that it is an important predictor to s-parameter performance. On the other hand variables like height.of.patch, height.of.solder.resist.layer and c_pad were not present in any models. This observation underlines the idea that these variables can be considered as fairly ineffective in determination of s11 magnitude, real parts or imaginary parts. The variables length.of.patch and width.of.patch can also be considered as important since they are present in most of the models created.

Let's see if these observations are supported by PCA that was done at the beginning. The following is the loading of variables in the first principal component:

```
> print(key_variables_pc1)
                  width.of.patch                       height.of.substrate
                      0.62417829                                0.62413737
  dielectric.constant.of.substrate                         length.of.patch
                      0.44510015                                0.10132741
                 height.of.patch                                     c_pad
                      0.07105448                                0.05422779
                         c_probe dielectric.constant.of.solder.resist.layer
                      0.03921288                                0.03842190
             radius.of.the.probe             height.of.solder.resist.layer
                      0.02599671                                0.02104923
                       c_antipad
                      0.01872008
```

It is seen that width.of.patch and height.of.substrate are the two variables that are most effective in PC1, these two were also found to be important after interpreting regression results. The variable radius.of.the.probe was present in multiple models but PCA shows very low loading for this variable. Since PCA does not have any information about the response variable in this case, a strong relation between PCA and regression results was not expected. PCA only explains which variables may be more useful while doing dimensionality reduction in explaining the variance of the original input data.

Please find comments on answers to the question 3.2 and 3.3 below.

```
#One key observation is that height of substrate is present in all models that
#are developed.
#Width of patch and length of patch seem to be important in determining magnitudes,
#real parts and imaginary parts of the s11 parameter.
#In all models developed, Residuals vs Fitted grapgh shows some behaviour of
#residuals that must be addressed. This can normally be done by selecting
#a generalized linear model with a link function (from Gamma family, probably)
#However this was not done since regression was the topic for the time being.
#Yet, it should be stated that the current models are not reliable in fully
#understanding the response variable in relation with the input variables.
#It is unlikely that a regression model will be suitable for all frequency values since
#this analysis showed very different models (with different selection of variables)
#for the two studied common resonance frequency values.
#This analysis lacked presence of k-fold cross validation. While comparing
#different models, AIC values and test errors are studied but in abscence of
#k-fold cross validation, selection of test data may influenced the value of
#test errors for different types of models.

#PCA was not used as a predictive tool in this work. PCA was conducted only on
#input variables.
```