# Exercise 3 R Markdown

Santanu Mukherjee

07/03/2022

# R Markdown

## Question

Infrared (IR) spectroscopy technology is used to determine the chemical makeup of a substance. The theory of IR spectroscopy holds that unique molecular structures absorb IR frequencies differently. In practice a spectrometer fires a series of IR frequencies into a sample material, and the device measures the absorbance of the sample at each individual frequency. This series of measurements creates a spectrum profile which can then be used to determine the chemical makeup of the sample material.

A Tecator Infratec Food and Feed Analyzer instrument was used to analyze 215 samples of meat across 100 frequencies. A sample of these frequency profiles is displayed in Fig. 6.20. In addition to an IR profile, analytical chemistry determined the percent content of water, fat, and protein for each sample. If we can establish a predictive relationship between IR spectrum and fat content, then food scientists could predict a sample's fat content with IR instead of using analytical chemistry. This would provide costs savings, since analytical chemistry is a more expensive, time-consuming process

### a

Start R and use these commands to load the data

## Answer (a)

```
library(caret)
data(tecator)
# use ?tecator to see more details
```

The matrix absorp contains the 100 absorbance values for the 215 samples, while matrix endpoints contains the percent of moisture, fat, and protein in columns 1–3, respectively.

### b

In this example the predictors are the measurements at the individual frequencies. Because the frequencies lie in a systematic order (850–1,050nm), the predictors have a high degree of correlation. Hence, the data lie in a smaller dimension than the total number of predictors (215). Use PCA to determine the effective dimension of these data. What is the effective dimension?

# Answer (b)

The function $prcomp$ is used for PCA.

```
pca.b = prcomp(absorp, center = TRUE, scale = TRUE )

# Determining percent of variance associated with each component

pct.var = pca.b$sdev^2/sum(pca.b$sdev^2) * 100
head(pct.var)
```

```
## [1] 98.626192582  0.969705229  0.279324276  0.114429868  0.006460911
## [6]  0.002624591
```

The $pct.var$ data shows that the first components accounts for almost 98% of the information in the data. So, reviewing this analysis, we can say that the true real dimensionality is much lower than the total number of predictors. But this is considering the linear combination of data. If we try to use non-linear combinations of the predictors, we might get a different result.

## c

Split the data into a training and a test set the response of the percentage of moisture, pre-process the data, and build each variety of models described in this chapter. For those models with tuning parameters, what are the optimal values of the tuning parameter(s)?

# Answer (c)

So, there are 3 endpoints and based on the question, we will model the percentage of moisture (first column). We will do the split of the dataset into 80% (train) and 20% (test), by using $createDataPartition$ function. We will also perform 10-fold Cross validation (CV) with 5 repeats to tune the models.

```
set.seed(10)
train.part = createDataPartition(endpoints[,1], p = 0.8, list=FALSE)
absorp = data.frame(absorp)

absorp.train = absorp[train.part,]
absorp.test = absorp[-train.part,]
moisture.train = endpoints[train.part,1]
moisture.test = endpoints[-train.part,1]

cntrl = trainControl(method = "repeatedcv", repeats = 5)
```

```
# Build various models and then compare performance:

# Simple Linear Model
set.seed(15)
lm.model = train( absorp.train, moisture.train, method="lm", preProcess=c("center","scale"), trControl=trainControl(method=
"repeatedcv",repeats=5) )
lm.model
```

```
## Linear Regression
##
## 175 samples
## 100 predictors
##
## Pre-processing: centered (100), scaled (100)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 158, 158, 157, 157, 158, 158, ...
## Resampling results:
##
##   RMSE     Rsquared  MAE
##   3.32623  0.88888   1.979368
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
# For Robust Linear Model (rlm) we cannot have a singular predictor covariance matrix thus we preprocess with PCA:
#
set.seed(0)
rlm.model = train( absorp.train, moisture.train, method="rlm", preProcess=c("pca"), trControl=trainControl(method="repeatedcv",repeats=5) )
rlm.model
```

```
## Robust Linear Model
##
## 175 samples
## 100 predictors
##
## Pre-processing: principal component signal extraction (100), centered
##  (100), scaled (100)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 157, 158, 156, 156, 158, 159, ...
## Resampling results across tuning parameters:
##
##   intercept  psi           RMSE       Rsquared   MAE
##   FALSE      psi.huber     63.833226  0.3325478  63.304275
##   FALSE      psi.hampel    63.833226  0.3325478  63.304275
##   FALSE      psi.bisquare  63.834660  0.3324373  63.305393
##    TRUE      psi.huber      8.295690  0.3373398   6.521853
##    TRUE      psi.hampel     8.242311  0.3338106   6.673744
##    TRUE      psi.bisquare   8.358805  0.3371711   6.485898
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were intercept = TRUE and psi = psi.hampel.
```

```
# Ridge regression model
#
ridgeGrid = data.frame(.lambda=seq(0,1,length=20))
set.seed(0)
ridge.model = train( absorp.train, moisture.train, method="ridge",
                         # fit the model over many penalty values
                         tuneGrid = ridgeGrid,
                         preProcess=c("center","scale"), trControl=trainControl(method="repeatedcv",repeats=5) )

ridge.model
```

```
## Ridge Regression
##
## 175 samples
## 100 predictors
##
## Pre-processing: centered (100), scaled (100)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 157, 158, 156, 156, 158, 159, ...
## Resampling results across tuning parameters:
##
##   lambda      RMSE      Rsquared   MAE
##   0.00000000  2.880041  0.9120609  1.804162
##   0.05263158  3.530150  0.8820442  2.875805
##   0.10526316  3.989872  0.8505225  3.218088
##   0.15789474  4.370046  0.8194993  3.467830
##   0.21052632  4.689266  0.7897057  3.666993
##   0.26315789  4.965211  0.7616509  3.848756
##   0.31578947  5.210878  0.7355433  4.004838
##   0.36842105  5.435341  0.7114154  4.151095
##   0.42105263  5.644948  0.6892022  4.294244
##   0.47368421  5.844200  0.6687891  4.433754
##   0.52631579  6.036329  0.6500396  4.573339
##   0.57894737  6.223692  0.6328114  4.714598
##   0.63157895  6.408019  0.6169655  4.858849
##   0.68421053  6.590592  0.6023708  5.003223
##   0.73684211  6.772359  0.5889068  5.148920
##   0.78947368  6.954028  0.5764636  5.294776
##   0.84210526  7.136120  0.5649424  5.440962
##   0.89473684  7.319017  0.5542544  5.587667
##   0.94736842  7.502997  0.5443204  5.736594
##   1.00000000  7.688255  0.5350697  5.891630
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was lambda = 0.
```

```
# Elastic net model
#
enetGrid = expand.grid(.lambda=seq(0,1,length=20), .fraction=seq(0.05, 1.0, length=20))
set.seed(0)
enet.model = train( absorp.train, moisture.train, method="enet",
                    # fit the model over many penalty values
                    tuneGrid = enetGrid,
                    preProcess=c("center","scale"), trControl=trainControl(method="repeatedcv",repeats=5) )


enet.model
```

```
## Elasticnet
##
## 175 samples
## 100 predictors
##
## Pre-processing: centered (100), scaled (100)
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 157, 158, 156, 156, 158, 159, ...
## Resampling results across tuning parameters:
##
##   lambda      fraction  RMSE      Rsquared   MAE
##   0.00000000  0.05      1.744305  0.9663799  1.314916
##   0.00000000  0.10      1.987830  0.9507733  1.352790
##   0.00000000  0.15      2.238584  0.9307347  1.404891
##   0.00000000  0.20      2.398709  0.9202195  1.458842
##   0.00000000  0.25      2.446852  0.9187633  1.483858
##   0.00000000  0.30      2.495782  0.9166490  1.513347
##   0.00000000  0.35      2.536271  0.9149973  1.536627
##   0.00000000  0.40      2.564876  0.9146263  1.558074
##   0.00000000  0.45      2.574608  0.9154113  1.574644
##   0.00000000  0.50      2.577263  0.9165863  1.587402
##   0.00000000  0.55      2.582686  0.9177286  1.597767
##   0.00000000  0.60      2.598768  0.9182389  1.610183
##   0.00000000  0.65      2.616773  0.9187171  1.624001
##   0.00000000  0.70      2.632508  0.9193363  1.637327
##   0.00000000  0.75      2.660434  0.9192143  1.654820
##   0.00000000  0.80      2.699164  0.9182937  1.679466
##   0.00000000  0.85      2.741733  0.9170207  1.706573
##   0.00000000  0.90      2.779952  0.9159437  1.736602
##   0.00000000  0.95      2.825053  0.9143438  1.768703
##   0.00000000  1.00      2.880041  0.9120609  1.804162
##   0.05263158  0.05      7.908420  0.3646641  6.612717
##   0.05263158  0.10      7.503058  0.4386488  6.284175
##   0.05263158  0.15      7.105665  0.5103555  5.955666
##   0.05263158  0.20      6.723931  0.5758946  5.631392
##   0.05263158  0.25      6.356563  0.6340717  5.310123
##   0.05263158  0.30      6.014462  0.6826842  5.005477
##   0.05263158  0.35      5.704208  0.7217144  4.723571
##   0.05263158  0.40      5.442804  0.7508481  4.490330
```

```
##   0.05263158  0.45      5.184101  0.7767216  4.271323
##   0.05263158  0.50      4.930760  0.7993993  4.059938
##   0.05263158  0.55      4.695380  0.8181647  3.860436
##   0.05263158  0.60      4.478457  0.8335743  3.672244
##   0.05263158  0.65      4.278740  0.8462752  3.505543
##   0.05263158  0.70      4.105381  0.8561780  3.368196
##   0.05263158  0.75      3.953677  0.8640343  3.248989
##   0.05263158  0.80      3.825408  0.8701078  3.141000
##   0.05263158  0.85      3.722159  0.8745973  3.049421
##   0.05263158  0.90      3.641387  0.8778492  2.976094
##   0.05263158  0.95      3.577969  0.8803041  2.917240
##   0.05263158  1.00      3.530150  0.8820442  2.875805
##   0.10526316  0.05      7.944898  0.3593900  6.686060
##   0.10526316  0.10      7.631695  0.4123601  6.345314
##   0.10526316  0.15      7.285697  0.4730805  6.056698
##   0.10526316  0.20      6.955134  0.5297627  5.774763
##   0.10526316  0.25      6.638870  0.5814873  5.498273
##   0.10526316  0.30      6.339703  0.6271315  5.231085
##   0.10526316  0.35      6.062613  0.6660662  4.980584
##   0.10526316  0.40      5.834205  0.6954886  4.775073
##   0.10526316  0.45      5.615583  0.7212065  4.586581
##   0.10526316  0.50      5.395749  0.7448926  4.403161
##   0.10526316  0.55      5.183540  0.7658040  4.224909
##   0.10526316  0.60      4.988228  0.7834137  4.057174
##   0.10526316  0.65      4.806938  0.7983930  3.898277
##   0.10526316  0.70      4.641378  0.8109577  3.753137
##   0.10526316  0.75      4.490576  0.8214708  3.623772
##   0.10526316  0.80      4.360743  0.8298460  3.516600
##   0.10526316  0.85      4.245625  0.8367660  3.425527
##   0.10526316  0.90      4.145463  0.8424154  3.344489
##   0.10526316  0.95      4.061266  0.8469000  3.276108
##   0.10526316  1.00      3.989872  0.8505225  3.218088
##   0.15789474  0.05      8.001602  0.3587179  6.767053
##   0.15789474  0.10      7.727818  0.3949224  6.375768
##   0.15789474  0.15      7.418607  0.4471956  6.115362
##   0.15789474  0.20      7.122506  0.4969570  5.861879
##   0.15789474  0.25      6.839011  0.5432761  5.614023
##   0.15789474  0.30      6.568859  0.5854880  5.374844
##   0.15789474  0.35      6.319379  0.6223140  5.150356
```

```
##   0.15789474  0.40      6.107815  0.6517115  4.957130
##   0.15789474  0.45      5.914579  0.6767215  4.788174
##   0.15789474  0.50      5.717692  0.7002561  4.620917
##   0.15789474  0.55      5.529666  0.7211654  4.462053
##   0.15789474  0.60      5.351328  0.7396810  4.309523
##   0.15789474  0.65      5.185575  0.7557493  4.165679
##   0.15789474  0.70      5.030563  0.7697741  4.028461
##   0.15789474  0.75      4.890191  0.7816851  3.902774
##   0.15789474  0.80      4.763045  0.7918198  3.791886
##   0.15789474  0.85      4.646570  0.8005872  3.690504
##   0.15789474  0.90      4.541806  0.8080610  3.600488
##   0.15789474  0.95      4.450838  0.8142432  3.528591
##   0.15789474  1.00      4.370046  0.8194993  3.467830
##   0.21052632  0.05      8.058392  0.3584211  6.834024
##   0.21052632  0.10      7.812313  0.3819245  6.392380
##   0.21052632  0.15      7.530735  0.4277685  6.153995
##   0.21052632  0.20      7.259170  0.4721462  5.920344
##   0.21052632  0.25      7.000014  0.5139262  5.695614
##   0.21052632  0.30      6.754877  0.5522996  5.481340
##   0.21052632  0.35      6.523741  0.5871525  5.276181
##   0.21052632  0.40      6.322691  0.6161904  5.093375
##   0.21052632  0.45      6.146589  0.6403668  4.933910
##   0.21052632  0.50      5.972526  0.6625844  4.782503
##   0.21052632  0.55      5.800590  0.6831895  4.635650
##   0.21052632  0.60      5.637098  0.7017328  4.495070
##   0.21052632  0.65      5.483389  0.7182421  4.361926
##   0.21052632  0.70      5.338659  0.7329738  4.234586
##   0.21052632  0.75      5.206925  0.7457040  4.117601
##   0.21052632  0.80      5.083736  0.7570352  4.007611
##   0.21052632  0.85      4.969992  0.7670070  3.908316
##   0.21052632  0.90      4.865913  0.7757359  3.819676
##   0.21052632  0.95      4.773186  0.7832025  3.739952
##   0.21052632  1.00      4.689266  0.7897057  3.666993
##   0.26315789  0.05      8.111733  0.3582016  6.891358
##   0.26315789  0.10      7.892145  0.3717939  6.404541
##   0.26315789  0.15      7.631570  0.4127139  6.183572
##   0.26315789  0.20      7.380351  0.4525554  5.968169
##   0.26315789  0.25      7.140134  0.4904863  5.762099
##   0.26315789  0.30      6.913896  0.5256299  5.566658
```

```
##   0.26315789  0.35      6.698563  0.5582372  5.376295
##   0.26315789  0.40      6.505278  0.5866678  5.201073
##   0.26315789  0.45      6.341993  0.6098910  5.050005
##   0.26315789  0.50      6.185633  0.6308149  4.909866
##   0.26315789  0.55      6.027028  0.6507658  4.771437
##   0.26315789  0.60      5.875943  0.6689081  4.641330
##   0.26315789  0.65      5.731827  0.6854752  4.516416
##   0.26315789  0.70      5.596854  0.7003461  4.398564
##   0.26315789  0.75      5.471356  0.7136009  4.289733
##   0.26315789  0.80      5.352936  0.7256175  4.186553
##   0.26315789  0.85      5.243012  0.7363399  4.089921
##   0.26315789  0.90      5.141783  0.7458564  4.000088
##   0.26315789  0.95      5.049577  0.7542327  3.920853
##   0.26315789  1.00      4.965211  0.7616509  3.848756
##   0.31578947  0.05      8.160208  0.3580197  6.941818
##   0.31578947  0.10      7.970343  0.3636735  6.420612
##   0.31578947  0.15      7.726982  0.4006894  6.213503
##   0.31578947  0.20      7.492543  0.4367731  6.013743
##   0.31578947  0.25      7.268721  0.4713032  5.821474
##   0.31578947  0.30      7.056784  0.5037799  5.637406
##   0.31578947  0.35      6.855006  0.5342011  5.459082
##   0.31578947  0.40      6.669358  0.5616595  5.290759
##   0.31578947  0.45      6.515491  0.5839976  5.147423
##   0.31578947  0.50      6.371963  0.6038703  5.016395
##   0.31578947  0.55      6.225764  0.6228918  4.887406
##   0.31578947  0.60      6.084348  0.6405199  4.763884
##   0.31578947  0.65      5.948477  0.6568658  4.646034
##   0.31578947  0.70      5.821483  0.6716231  4.538283
##   0.31578947  0.75      5.701770  0.6850727  4.436208
##   0.31578947  0.80      5.588562  0.6973778  4.338489
##   0.31578947  0.85      5.482360  0.7085539  4.245677
##   0.31578947  0.90      5.384337  0.7185589  4.159120
##   0.31578947  0.95      5.294047  0.7275158  4.078372
##   0.31578947  1.00      5.210878  0.7355433  4.004838
##   0.36842105  0.05      8.203852  0.3578914  6.988055
##   0.36842105  0.10      8.030675  0.3580990  6.438783
##   0.36842105  0.15      7.820926  0.3907619  6.249809
##   0.36842105  0.20      7.600492  0.4237406  6.060949
##   0.36842105  0.25      7.390685  0.4553581  5.879918
```

```
##   0.36842105  0.30      7.190627  0.4855012  5.704135
##   0.36842105  0.35      7.000752  0.5138705  5.533357
##   0.36842105  0.40      6.823161  0.5401137  5.371538
##   0.36842105  0.45      6.675156  0.5617911  5.234171
##   0.36842105  0.50      6.541796  0.5807303  5.110890
##   0.36842105  0.55      6.406635  0.5987660  4.989813
##   0.36842105  0.60      6.273234  0.6158110  4.874389
##   0.36842105  0.65      6.144794  0.6317373  4.764687
##   0.36842105  0.70      6.024608  0.6462247  4.663095
##   0.36842105  0.75      5.910087  0.6596587  4.565744
##   0.36842105  0.80      5.801589  0.6720438  4.473073
##   0.36842105  0.85      5.699486  0.6833987  4.384817
##   0.36842105  0.90      5.604896  0.6936553  4.301927
##   0.36842105  0.95      5.516827  0.7029813  4.224013
##   0.36842105  1.00      5.435341  0.7114154  4.151095
##   0.42105263  0.05      8.242997  0.3577889  7.028577
##   0.42105263  0.10      8.018888  0.3567174  6.450886
##   0.42105263  0.15      7.915304  0.3824400  6.291505
##   0.42105263  0.20      7.707059  0.4127716  6.112493
##   0.42105263  0.25      7.509073  0.4419314  5.939309
##   0.42105263  0.30      7.319525  0.4699822  5.771229
##   0.42105263  0.35      7.140061  0.4964956  5.609098
##   0.42105263  0.40      6.970633  0.5213935  5.454195
##   0.42105263  0.45      6.826507  0.5425352  5.319588
##   0.42105263  0.50      6.700914  0.5606963  5.202119
##   0.42105263  0.55      6.575311  0.5777977  5.091436
##   0.42105263  0.60      6.449332  0.5941785  4.982903
##   0.42105263  0.65      6.327682  0.6095684  4.878868
##   0.42105263  0.70      6.213063  0.6237378  4.781169
##   0.42105263  0.75      6.103518  0.6369830  4.688705
##   0.42105263  0.80      5.999485  0.6492851  4.600215
##   0.42105263  0.85      5.901147  0.6606699  4.515769
##   0.42105263  0.90      5.810011  0.6710062  4.437303
##   0.42105263  0.95      5.724466  0.6805220  4.363419
##   0.42105263  1.00      5.644948  0.6892022  4.294244
##   0.47368421  0.05      8.278081  0.3576996  7.063938
##   0.47368421  0.10      7.993504  0.3565425  6.470022
##   0.47368421  0.15      8.011239  0.3753659  6.335003
##   0.47368421  0.20      7.813852  0.4034210  6.165032
```

```
##   0.47368421  0.25      7.626134  0.4304817  6.002004
##   0.47368421  0.30      7.446032  0.4566573  5.843616
##   0.47368421  0.35      7.275603  0.4815224  5.691150
##   0.47368421  0.40      7.114395  0.5050244  5.544031
##   0.47368421  0.45      6.973251  0.5256674  5.413777
##   0.47368421  0.50      6.853114  0.5432163  5.301834
##   0.47368421  0.55      6.736090  0.5594550  5.196937
##   0.47368421  0.60      6.616877  0.5751604  5.093137
##   0.47368421  0.65      6.501479  0.5899522  4.993082
##   0.47368421  0.70      6.391875  0.6037390  4.899090
##   0.47368421  0.75      6.286692  0.6167288  4.809268
##   0.47368421  0.80      6.186995  0.6288214  4.724462
##   0.47368421  0.85      6.092317  0.6401127  4.644373
##   0.47368421  0.90      6.004319  0.6504339  4.569491
##   0.47368421  0.95      5.921434  0.6600044  4.498872
##   0.47368421  1.00      5.844200  0.6687891  4.433754
##   0.52631579  0.05      8.309604  0.3576178  7.095327
##   0.52631579  0.10      7.975673  0.3564301  6.495131
##   0.52631579  0.15      8.109505  0.3692834  6.386438
##   0.52631579  0.20      7.922099  0.3953447  6.226803
##   0.52631579  0.25      7.743381  0.4206007  6.073005
##   0.52631579  0.30      7.571888  0.4450970  5.922840
##   0.52631579  0.35      7.409496  0.4684846  5.779077
##   0.52631579  0.40      7.255978  0.4906692  5.640934
##   0.52631579  0.45      7.118004  0.5107351  5.513472
##   0.52631579  0.50      7.001615  0.5278054  5.404604
##   0.52631579  0.55      6.891937  0.5432880  5.305092
##   0.52631579  0.60      6.779274  0.5583062  5.206162
##   0.52631579  0.65      6.669469  0.5725077  5.110781
##   0.52631579  0.70      6.564517  0.5858632  5.019782
##   0.52631579  0.75      6.463390  0.5985394  4.932998
##   0.52631579  0.80      6.367534  0.6103802  4.851522
##   0.52631579  0.85      6.276556  0.6214723  4.774650
##   0.52631579  0.90      6.191542  0.6317022  4.703105
##   0.52631579  0.95      6.111270  0.6412409  4.635799
##   0.52631579  1.00      6.036329  0.6500396  4.573339
##   0.57894737  0.05      8.338036  0.3575475  7.123754
##   0.57894737  0.10      7.964508  0.3563342  6.522768
##   0.57894737  0.15      8.210648  0.3639794  6.449433
```

```
##   0.57894737  0.20      8.032484  0.3883015  6.298615
##   0.57894737  0.25      7.861914  0.4119776  6.152532
##   0.57894737  0.30      7.698361  0.4349716  6.011045
##   0.57894737  0.35      7.543147  0.4570408  5.873520
##   0.57894737  0.40      7.396646  0.4780143  5.740904
##   0.57894737  0.45      7.262584  0.4973974  5.615787
##   0.57894737  0.50      7.148471  0.5141099  5.508075
##   0.57894737  0.55      7.044985  0.5289424  5.413189
##   0.57894737  0.60      6.938608  0.5432832  5.319434
##   0.57894737  0.65      6.834059  0.5569044  5.228922
##   0.57894737  0.70      6.733252  0.5698304  5.142339
##   0.57894737  0.75      6.636123  0.5821325  5.059553
##   0.57894737  0.80      6.543780  0.5936920  4.981332
##   0.57894737  0.85      6.456161  0.6045474  4.907027
##   0.57894737  0.90      6.374171  0.6146084  4.838549
##   0.57894737  0.95      6.296398  0.6240583  4.774250
##   0.57894737  1.00      6.223692  0.6328114  4.714598
##   0.63157895  0.05      8.363793  0.3574829  7.149372
##   0.63157895  0.10      7.958013  0.3562532  6.549056
##   0.63157895  0.15      8.310642  0.3595091  6.517643
##   0.63157895  0.20      8.145528  0.3821101  6.380547
##   0.63157895  0.25      7.982389  0.4043911  6.241685
##   0.63157895  0.30      7.826116  0.4260446  6.105353
##   0.63157895  0.35      7.677576  0.4469075  5.972651
##   0.63157895  0.40      7.537503  0.4667866  5.844606
##   0.63157895  0.45      7.407634  0.4854460  5.723432
##   0.63157895  0.50      7.295307  0.5018327  5.616437
##   0.63157895  0.55      7.196860  0.5161248  5.524929
##   0.63157895  0.60      7.096351  0.5298330  5.436136
##   0.63157895  0.65      6.996861  0.5428775  5.350204
##   0.63157895  0.70      6.899920  0.5553763  5.267186
##   0.63157895  0.75      6.806379  0.5673048  5.187587
##   0.63157895  0.80      6.717493  0.5785418  5.112410
##   0.63157895  0.85      6.633057  0.5891306  5.041354
##   0.63157895  0.90      6.553830  0.5989954  4.975795
##   0.63157895  0.95      6.478540  0.6083043  4.915135
##   0.63157895  1.00      6.408019  0.6169655  4.858849
##   0.68421053  0.05      8.387162  0.3574258  7.172338
##   0.68421053  0.10      7.954926  0.3561842  6.572971
```

```
##   0.68421053  0.15      8.389014  0.3563810  6.576129
##   0.68421053  0.20      8.261429  0.3766278  6.468854
##   0.68421053  0.25      8.105220  0.3976645  6.334608
##   0.68421053  0.30      7.955691  0.4181232  6.203070
##   0.68421053  0.35      7.813311  0.4378968  6.074556
##   0.68421053  0.40      7.679299  0.4567652  5.951143
##   0.68421053  0.45      7.553673  0.4747016  5.832638
##   0.68421053  0.50      7.443114  0.4907525  5.727012
##   0.68421053  0.55      7.348533  0.5046088  5.639426
##   0.68421053  0.60      7.253597  0.5177303  5.555823
##   0.68421053  0.65      7.158828  0.5302379  5.475329
##   0.68421053  0.70      7.065693  0.5422945  5.396182
##   0.68421053  0.75      6.975505  0.5538512  5.319719
##   0.68421053  0.80      6.889807  0.5647575  5.247574
##   0.68421053  0.85      6.808425  0.5750594  5.180306
##   0.68421053  0.90      6.731848  0.5847050  5.117582
##   0.68421053  0.95      6.658952  0.5938404  5.058396
##   0.68421053  1.00      6.590592  0.6023708  5.003223
##   0.73684211  0.05      8.408412  0.3573751  7.193138
##   0.73684211  0.10      7.954313  0.3561226  6.594589
##   0.73684211  0.15      8.408567  0.3550531  6.592216
##   0.73684211  0.20      8.380351  0.3717425  6.558115
##   0.73684211  0.25      8.230653  0.3916669  6.427974
##   0.73684211  0.30      8.087493  0.4110438  6.300604
##   0.73684211  0.35      7.950906  0.4298275  6.176948
##   0.73684211  0.40      7.822417  0.4477863  6.057965
##   0.73684211  0.45      7.701256  0.4649892  5.944222
##   0.73684211  0.50      7.592464  0.4806996  5.842303
##   0.73684211  0.55      7.500797  0.4942078  5.757685
##   0.73684211  0.60      7.411159  0.5067810  5.678640
##   0.73684211  0.65      7.320946  0.5187879  5.602036
##   0.73684211  0.70      7.231368  0.5304163  5.526607
##   0.73684211  0.75      7.144431  0.5415987  5.453776
##   0.73684211  0.80      7.061674  0.5521810  5.385305
##   0.73684211  0.85      6.983234  0.5621773  5.320449
##   0.73684211  0.90      6.909170  0.5715929  5.259859
##   0.73684211  0.95      6.838616  0.5805322  5.202747
##   0.73684211  1.00      6.772359  0.5889068  5.148920
##   0.78947368  0.05      8.427826  0.3573290  7.212111
```

```
##   0.78947368  0.10      7.955445  0.3560645  6.614010
##   0.78947368  0.15      8.380883  0.3546497  6.572639
##   0.78947368  0.20      8.502424  0.3673589  6.651559
##   0.78947368  0.25      8.358994  0.3862751  6.524172
##   0.78947368  0.30      8.221741  0.4046842  6.400103
##   0.78947368  0.35      8.090715  0.4225563  6.279995
##   0.78947368  0.40      7.967390  0.4396866  6.166264
##   0.78947368  0.45      7.850628  0.4561807  6.058719
##   0.78947368  0.50      7.744023  0.4715097  5.960920
##   0.78947368  0.55      7.654362  0.4847482  5.878834
##   0.78947368  0.60      7.569582  0.4968359  5.804961
##   0.78947368  0.65      7.483781  0.5083714  5.732069
##   0.78947368  0.70      7.397644  0.5195907  5.659915
##   0.78947368  0.75      7.313873  0.5303948  5.590015
##   0.78947368  0.80      7.233941  0.5406566  5.523670
##   0.78947368  0.85      7.158225  0.5503535  5.461520
##   0.78947368  0.90      7.086555  0.5595292  5.403151
##   0.78947368  0.95      7.018264  0.5682577  5.347242
##   0.78947368  1.00      6.954028  0.5764636  5.294776
##   0.84210526  0.05      8.445650  0.3572872  7.229251
##   0.84210526  0.10      7.957829  0.3560107  6.631543
##   0.84210526  0.15      8.341553  0.3545403  6.547104
##   0.84210526  0.20      8.627572  0.3634045  6.751987
##   0.84210526  0.25      8.490326  0.3813999  6.626245
##   0.84210526  0.30      8.358605  0.3989383  6.504664
##   0.84210526  0.35      8.232870  0.4159748  6.388576
##   0.84210526  0.40      8.114426  0.4323474  6.279342
##   0.84210526  0.45      8.001858  0.4481781  6.177674
##   0.84210526  0.50      7.898003  0.4630745  6.083132
##   0.84210526  0.55      7.809802  0.4760946  6.004227
##   0.84210526  0.60      7.729278  0.4877653  5.933394
##   0.84210526  0.65      7.647769  0.4988549  5.863813
##   0.84210526  0.70      7.565033  0.5096820  5.794190
##   0.84210526  0.75      7.484241  0.5201235  5.727258
##   0.84210526  0.80      7.407050  0.5300647  5.663633
##   0.84210526  0.85      7.333936  0.5394682  5.603814
##   0.84210526  0.90      7.264565  0.5483982  5.546718
##   0.84210526  0.95      7.198408  0.5569146  5.492499
##   0.84210526  1.00      7.136120  0.5649424  5.440962
```

```
##   0.89473684  0.05      8.462032  0.3572497  7.244773
##   0.89473684  0.10      7.961150  0.3559608  6.647431
##   0.89473684  0.15      8.303165  0.3544626  6.523473
##   0.89473684  0.20      8.754897  0.3598759  6.863013
##   0.89473684  0.25      8.624559  0.3769806  6.741322
##   0.89473684  0.30      8.498225  0.3937175  6.622184
##   0.89473684  0.35      8.377541  0.4099890  6.508283
##   0.89473684  0.40      8.263710  0.4256660  6.403656
##   0.89473684  0.45      8.155192  0.4408792  6.305428
##   0.89473684  0.50      8.054424  0.4553170  6.214286
##   0.89473684  0.55      7.967557  0.4681306  6.135132
##   0.89473684  0.60      7.890636  0.4794516  6.065820
##   0.89473684  0.65      7.813122  0.4901411  5.998708
##   0.89473684  0.70      7.733806  0.5005860  5.931620
##   0.89473684  0.75      7.655900  0.5106793  5.866393
##   0.89473684  0.80      7.581348  0.5203027  5.805378
##   0.89473684  0.85      7.510684  0.5294231  5.747035
##   0.89473684  0.90      7.443523  0.5381099  5.691212
##   0.89473684  0.95      7.379442  0.5464086  5.638018
##   0.89473684  1.00      7.319017  0.5542544  5.587667
##   0.94736842  0.05      8.477119  0.3572165  7.258883
##   0.94736842  0.10      7.965123  0.3559154  6.661897
##   0.94736842  0.15      8.270103  0.3543900  6.505808
##   0.94736842  0.20      8.877963  0.3568571  6.971475
##   0.94736842  0.25      8.761757  0.3729515  6.862011
##   0.94736842  0.30      8.640568  0.3889603  6.747820
##   0.94736842  0.35      8.524708  0.4045301  6.640919
##   0.94736842  0.40      8.415247  0.4195688  6.540802
##   0.94736842  0.45      8.310707  0.4342008  6.444384
##   0.94736842  0.50      8.213137  0.4481780  6.353601
##   0.94736842  0.55      8.127740  0.4607743  6.273628
##   0.94736842  0.60      8.053807  0.4718068  6.204644
##   0.94736842  0.65      7.980085  0.4821342  6.138893
##   0.94736842  0.70      7.904127  0.4922146  6.073228
##   0.94736842  0.75      7.829059  0.5019725  6.010499
##   0.94736842  0.80      7.757037  0.5112892  5.950441
##   0.94736842  0.85      7.688668  0.5201386  5.893160
##   0.94736842  0.90      7.623650  0.5285827  5.838570
##   0.94736842  0.95      7.561598  0.5366611  5.786226
```

```
##   0.94736842  1.00      7.502997  0.5443204  5.736594
##   1.00000000  0.05      8.491033  0.3571875  7.271745
##   1.00000000  0.10      7.969509  0.3558725  6.675172
##   1.00000000  0.15      8.241416  0.3543226  6.493832
##   1.00000000  0.20      8.978500  0.3546486  7.057720
##   1.00000000  0.25      8.901800  0.3692684  6.985117
##   1.00000000  0.30      8.785621  0.3846070  6.878862
##   1.00000000  0.35      8.674395  0.3995338  6.777822
##   1.00000000  0.40      8.569132  0.4139830  6.681123
##   1.00000000  0.45      8.468438  0.4280718  6.588225
##   1.00000000  0.50      8.374185  0.4415874  6.500128
##   1.00000000  0.55      8.290397  0.4539610  6.419677
##   1.00000000  0.60      8.218849  0.4647512  6.350645
##   1.00000000  0.65      8.148802  0.4747531  6.286084
##   1.00000000  0.70      8.076131  0.4844898  6.222417
##   1.00000000  0.75      8.003863  0.4939243  6.161013
##   1.00000000  0.80      7.934291  0.5029478  6.102256
##   1.00000000  0.85      7.868152  0.5115315  6.045766
##   1.00000000  0.90      7.805134  0.5197444  5.991972
##   1.00000000  0.95      7.745043  0.5276041  5.940330
##   1.00000000  1.00      7.688255  0.5350697  5.891630
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were fraction = 0.05 and lambda = 0.
```

```
resamp = resamples(list(lm=lm.model,rlm=rlm.model,ridge=ridge.model,enet=enet.model) )
print( summary(resamp) )
```

```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lm, rlm, ridge, enet
## Number of resamples: 50
##
## MAE
##            Min.  1st Qu.   Median     Mean  3rd Qu.     Max. NA's
## lm    1.0220290 1.469033 1.899226 1.979368 2.320623 4.284284    0
## rlm   3.6990323 6.075221 6.650738 6.673744 7.208466 8.915632    0
## ridge 0.6994013 1.325802 1.726125 1.804162 2.117850 3.291893    0
## enet  0.6745766 1.118898 1.260453 1.314916 1.440572 2.343691    0
##
## RMSE
##            Min.  1st Qu.   Median     Mean  3rd Qu.      Max. NA's
## lm    1.3300316 2.002064 2.659664 3.326230 3.896379 10.927391    0
## rlm   4.3541481 7.424318 8.172840 8.242311 9.092080 11.330397    0
## ridge 0.9248001 1.923357 2.416120 2.880041 3.769823  8.310639    0
## enet  0.8234350 1.410570 1.562760 1.744305 1.808257  3.787166    0
##
## Rsquared
##               Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lm    4.029324e-01 0.8828940 0.9403876 0.8888800 0.9623534 0.9866768    0
## rlm   1.306027e-05 0.2451909 0.3286024 0.3338106 0.3886462 0.8641637    0
## ridge 4.123183e-01 0.9023163 0.9513751 0.9120609 0.9674778 0.9882256    0
## enet  8.212380e-01 0.9674414 0.9767300 0.9663799 0.9837908 0.9924534    0
```

```
print( summary(diff(resamp)) )
```

```
##
## Call:
## summary.diff.resamples(object = diff(resamp))
##
## p-value adjustment: bonferroni
## Upper diagonal: estimates of the difference
## Lower diagonal: p-value for H0: difference = 0
##
## MAE
##       lm        rlm       ridge     enet
## lm              -4.6944    0.1752    0.6645
## rlm   < 2.2e-16            4.8696    5.3588
## ridge 0.7081    < 2.2e-16            0.4892
## enet  3.751e-06 < 2.2e-16 1.111e-06
##
## RMSE
##       lm        rlm       ridge     enet
## lm              -4.9161    0.4462    1.5819
## rlm   < 2.2e-16            5.3623    6.4980
## ridge 0.9872    < 2.2e-16            1.1357
## enet  4.065e-05 < 2.2e-16 3.757e-06
##
## Rsquared
##       lm        rlm       ridge    enet
## lm               0.55507  -0.02318 -0.07750
## rlm   < 2.2e-16           -0.57825 -0.63257
## ridge 1.000000  < 2.2e-16          -0.05432
## enet  0.002415  < 2.2e-16 0.001826
```

So, the different models that I have tried are **Simple Linear Regression**, **Robust Linear Regression**, **Ridge Regression** and **Elastic Net**. The models **Ridge Regression** and **Elastic Net* have tuning parameters. For** Ridge Regression**, the optimal value for the tuning parameter ($\lambda$) is $0$, $RMSE$ is the lowest $3.246823$, $R-squared$ is the highest $0.8836765$.

For **Elastic Net*, the optimal values for $fraction$ = $0.05$ and $\lambda$ = $0$, $RMSE$ is the lowest $1.816646$, $R-squared$ is the highest $0.9664863$.

## d

Which model has the best predictive ability? Is any model significantly better or worse than the others?

# Answer (d)

```
# Prediction for Simple Linear Model and MSE

MSE.lm.test <- mean((predict(lm.model, data=absorp.test) - moisture.test)^2)
MSE.lm.test
```

```
## [1] 202.9323
```

```
RMSE.lm.test = sqrt(MSE.lm.test)
RMSE.lm.test
```

```
## [1] 14.24543
```

```
# Prediction for Robust Linear Regression Model and MSE

MSE.rlm.test <- mean((predict(rlm.model, data=absorp.test) - moisture.test)^2)
MSE.rlm.test
```

```
## [1] 140.7932
```

```
RMSE.rlm.test = sqrt(MSE.rlm.test)
RMSE.rlm.test
```

```
## [1] 11.86563
```

```
# Prediction for Ridge Regression Model and RMSE

MSE.ridge.test <- mean((predict(ridge.model, data=absorp.test) - moisture.test)^2)
MSE.ridge.test
```

```
## [1] 202.9324
```

```
RMSE.ridge.test = sqrt(MSE.ridge.test)
RMSE.ridge.test
```

```
## [1] 14.24543
```

```
# Prediction for Elastic Net Model and RMSE

MSE.enet.test <- mean((predict(enet.model, data=absorp.test) - moisture.test)^2)
MSE.enet.test
```

```
## [1] 200.3792
```

```
RMSE.enet.test = sqrt(MSE.enet.test)
RMSE.enet.test
```

```
## [1] 14.15554
```

Based on the results of the predictions, **Ridge Regression Model** has the lowest MSE of $11.865$ and so this is the best model.

e

Explain which model you would use for predicting the percentage of moisture of a sample.

# Answer (e)

Based on the results obtained, I would use the **Ridge Regression Model** for predicting the moisture of a sample as we got the lowest $MSE$ for that model.