DATA 624 - Homework 7

Richie Rivera

Question 6.2

6.2. Developing a model to predict permeability (see Sect. 1.4) could save significant resources for a pharmaceutical company, while at the same time more rapidly identifying molecules that have a sufficient permeability to become a drug:

```r
library(AppliedPredictiveModeling)
data(permeability)
```

(a) Start R and use these commands to load the data: The matrix fingerprints contains the 1,107 binary molecular predictors for the 165 compounds, while permeability contains permeability response.

```r
library(ggplot2)
library(caret)
library(dplyr)
library(pls)
library(RANN)

dim(permeability)
```

(b) The fingerprint predictors indicate the presence or absence of substructures of a molecule and are often sparse meaning that relatively few of the molecules contain each substructure. Filter out the predictors that have low frequencies using the nearZeroVar function from the caret package. How many predictors are left for modeling?

```
## [1] 165   1
```

```r
dim(fingerprints)
```

```
## [1]  165 1107
```

```r
no_variance_62b <- nearZeroVar(fingerprints)

filter_finger <- fingerprints[, -no_variance_62b]
```

The `nearZeroVar()` function from the caert package has removed 719 predictors that had little to no variance.

```r
set.seed(19940211)
# Partition the data into a sample of 80% of the full dataset
train_rows <- createDataPartition(
  permeability,
  p = 0.8,
  list = FALSE
)

# performing pre-processing on filter_finger
pp_filter_finger <- prcomp(
  filter_finger,
  center = TRUE,
  scale. = TRUE
)$x

# Use the sample to create a training dataset
train_independent <- pp_filter_finger[train_rows, ]
train_dependent <- permeability[train_rows]

str(train_independent)
```

(c) Split the data into a training and a test set, pre-process the data, and tune a PLS model. How many latent variables are optimal and what is the corresponding resampled estimate of R2?

```
##  num [1:133, 1:165] -5.87 -4.97 -2.78 -4.61 -4.96 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:133] "1" "2" "3" "4" ...
##   ..$ : chr [1:165] "PC1" "PC2" "PC3" "PC4" ...
```

```r
# Use the sample to create a testing dataset
test_independent <- pp_filter_finger[-train_rows, ]
test_dependent <- permeability[-train_rows]

str(test_independent)
```

```
##  num [1:32, 1:165] -7.15 -10 -5.09 15.31 -6.8 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:32] "8" "9" "10" "20" ...
##   ..$ : chr [1:165] "PC1" "PC2" "PC3" "PC4" ...
```

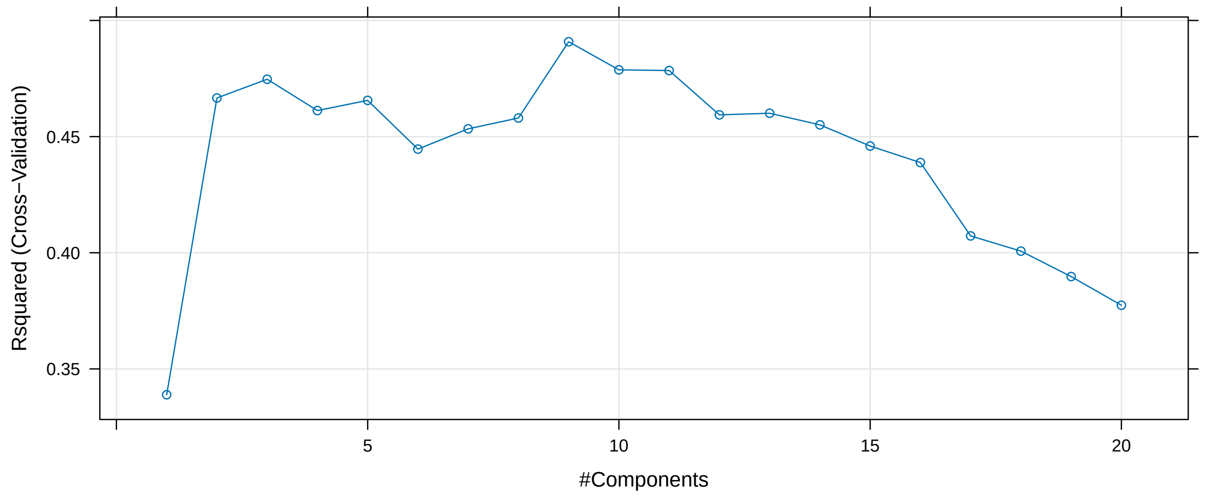
```r
# Tuning/Fitting our model
pls_fit <- train(
  x = train_independent,
  y = train_dependent,
  method = "pls",
  metric = "Rsquared",
  tuneLength = 20,
  trControl = trainControl(method = "cv", number = 10)
)
```

```r
optimal_result <- pls_fit$results |>
  arrange(desc(Rsquared)) |>
  head(1)

pls_fit$results
```

```
##    ncomp     RMSE  Rsquared       MAE   RMSESD RsquaredSD    MAESD
## 1      1 13.43901 0.3388749 10.282760 2.978579  0.2815885 1.767056
## 2      2 12.20110 0.4666269  9.006051 3.869627  0.2972117 2.397140
## 3      3 12.29694 0.4747034  9.434779 3.017306  0.2603653 1.981214
## 4      4 12.61021 0.4612168  9.784650 2.542258  0.2242891 1.761502
## 5      5 12.64158 0.4655996  9.490042 2.658704  0.2306457 1.851352
## 6      6 12.61142 0.4446318  9.571305 2.282995  0.1971172 1.527010
## 7      7 12.32710 0.4533517  9.513578 2.290700  0.1935326 1.535155
## 8      8 12.07100 0.4580050  9.521681 2.604024  0.1991950 1.960332
## 9      9 11.69082 0.4908505  8.989669 2.902400  0.2121236 2.016270
## 10    10 11.92369 0.4787437  9.276626 2.919607  0.2171244 2.050739
## 11    11 11.98937 0.4784396  9.196496 3.074471  0.2228767 2.082624
## 12    12 12.34770 0.4593690  9.498074 3.042321  0.2221941 2.099994
## 13    13 12.37582 0.4600611  9.489046 2.992542  0.2185876 2.123577
## 14    14 12.46653 0.4550778  9.397099 2.865548  0.1973506 1.898062
## 15    15 12.67118 0.4459445  9.652758 2.686416  0.1824183 1.739909
## 16    16 12.75992 0.4388306  9.611897 2.628752  0.1727606 1.573323
## 17    17 13.29061 0.4072524 10.109464 2.855286  0.1766001 1.810778
## 18    18 13.46613 0.4007145 10.279255 2.849763  0.1737949 1.865656
## 19    19 13.58891 0.3897567 10.377628 2.799085  0.1697751 1.921757
## 20    20 13.75721 0.3774046 10.463378 2.782364  0.1666304 1.959115
```

```r
plot(pls_fit)
```



```r
pls_fit
```

```
## Partial Least Squares
```

```
##
## 133 samples
## 165 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 120, 120, 120, 119, 120, 119, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE      Rsquared   MAE
##    1     13.43901  0.3388749  10.282760
##    2     12.20110  0.4666269   9.006051
##    3     12.29694  0.4747034   9.434779
##    4     12.61021  0.4612168   9.784650
##    5     12.64158  0.4655996   9.490042
##    6     12.61142  0.4446318   9.571305
##    7     12.32710  0.4533517   9.513578
##    8     12.07100  0.4580050   9.521681
##    9     11.69082  0.4908505   8.989669
##   10     11.92369  0.4787437   9.276626
##   11     11.98937  0.4784396   9.196496
##   12     12.34770  0.4593690   9.498074
##   13     12.37582  0.4600611   9.489046
##   14     12.46653  0.4550778   9.397099
##   15     12.67118  0.4459445   9.652758
##   16     12.75992  0.4388306   9.611897
##   17     13.29061  0.4072524  10.109464
##   18     13.46613  0.4007145  10.279255
##   19     13.58891  0.3897567  10.377628
##   20     13.75721  0.3774046  10.463378
##
## Rsquared was used to select the optimal model using the largest value.
## The final value used for the model was ncomp = 9.
```

From the model above, we can see that the optimal number of components is 9 with an R^2 of 0.4909

```r
# Get predictions on the testing dataset
test_predictions <- predict(
  pls_fit,
  test_independent,
  ncomp = optimal_result$ncomp
)

# Use postResample to compare results
test_results <- postResample(
  pred = test_predictions,
  obs = test_dependent
)

test_results
```

(d) Predict the response for the test set. What is the test set estimate of R2?

```
##       RMSE   Rsquared        MAE
## 10.2106903  0.6890925  7.6501522
```

Using the postResample function, we can see that the test set estimate of R^2 is 0.6891

(e) Try building other models discussed in this chapter. Do any have better predictive performance? A few of the other types of models discussed in this chapter are the penalized models:

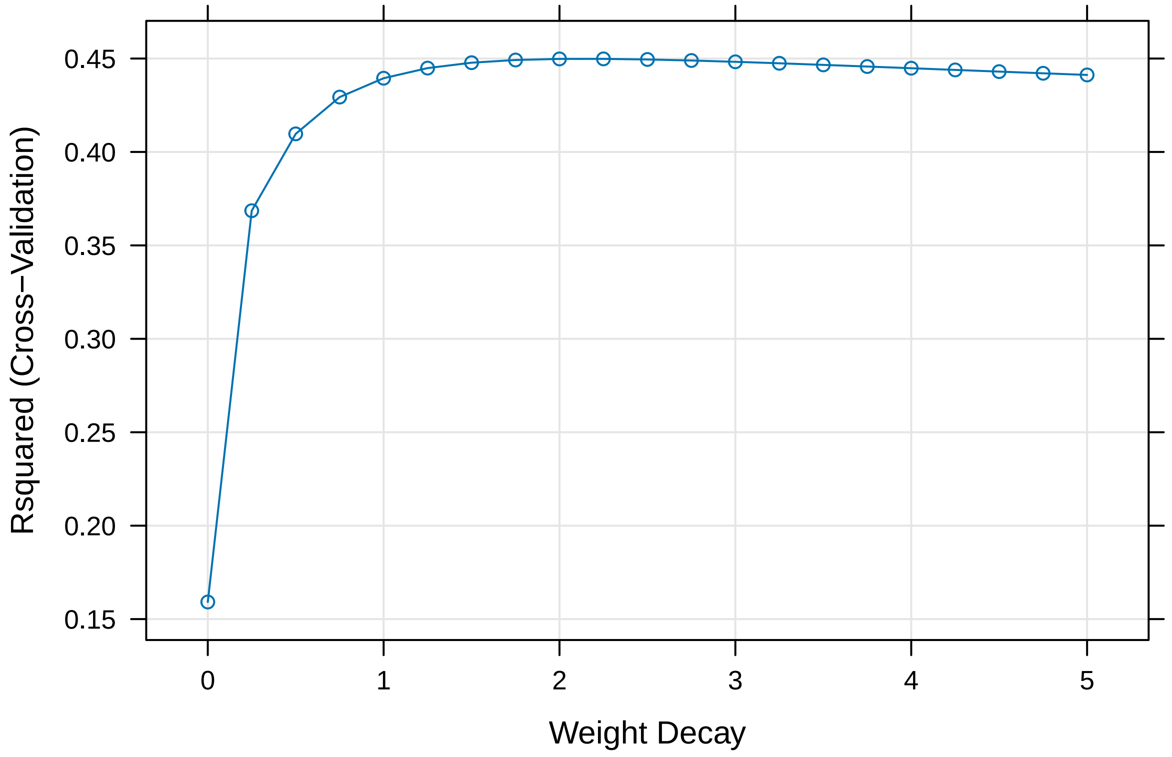
1. Ridge Method

```
ridge_fit <- train(
  x = train_independent,
  y = train_dependent,
  method = "ridge",
  metric = "Rsquared",
  tuneGrid =  data.frame(.lambda = seq(0, 5, by = .25)),
  trControl = trainControl(method = "cv", number = 10)
)

ridge_optimal <- ridge_fit$results |>
  arrange(Rsquared) |>
  tail(1)

ridge_test_results <- postResample(
  pred = predict(
    ridge_fit,
    test_independent
  ),
  test_dependent
)

plot(ridge_fit)
```



```
ridge_fit
```

```
## Ridge Regression
##
## 133 samples
## 165 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 121, 119, 120, 119, 119, 120, ...
## Resampling results across tuning parameters:
##
##   lambda  RMSE          Rsquared   MAE
##   0.00    6.022699e+15  0.1591556  3.230232e+15
##   0.25    1.381489e+01  0.3685460  1.037037e+01
##   0.50    1.348224e+01  0.4096806  1.019960e+01
##   0.75    1.354753e+01  0.4293616  1.029477e+01
##   1.00    1.381121e+01  0.4394675  1.061306e+01
##   1.25    1.419038e+01  0.4448779  1.098575e+01
##   1.50    1.463988e+01  0.4477712  1.138778e+01
##   1.75    1.513207e+01  0.4492185  1.178317e+01
##   2.00    1.564900e+01  0.4497921  1.219235e+01
##   2.25    1.617864e+01  0.4498188  1.260428e+01
##   2.50    1.671272e+01  0.4494952  1.300034e+01
##   2.75    1.724552e+01  0.4489442  1.337997e+01
##   3.00    1.777304e+01  0.4482456  1.375231e+01
##   3.25    1.829254e+01  0.4474524  1.411132e+01
##   3.50    1.880214e+01  0.4466002  1.445442e+01
##   3.75    1.930059e+01  0.4457139  1.480221e+01
##   4.00    1.978709e+01  0.4448106  1.514356e+01
##   4.25    2.026119e+01  0.4439024  1.547157e+01
##   4.50    2.072269e+01  0.4429977  1.579973e+01
##   4.75    2.117155e+01  0.4421026  1.612306e+01
##   5.00    2.160787e+01  0.4412211  1.643511e+01
##
## Rsquared was used to select the optimal model using the largest value.
## The final value used for the model was lambda = 2.25.
```

Across this ridge model, our best R^2 is 0.45 with a lambda of 2.25

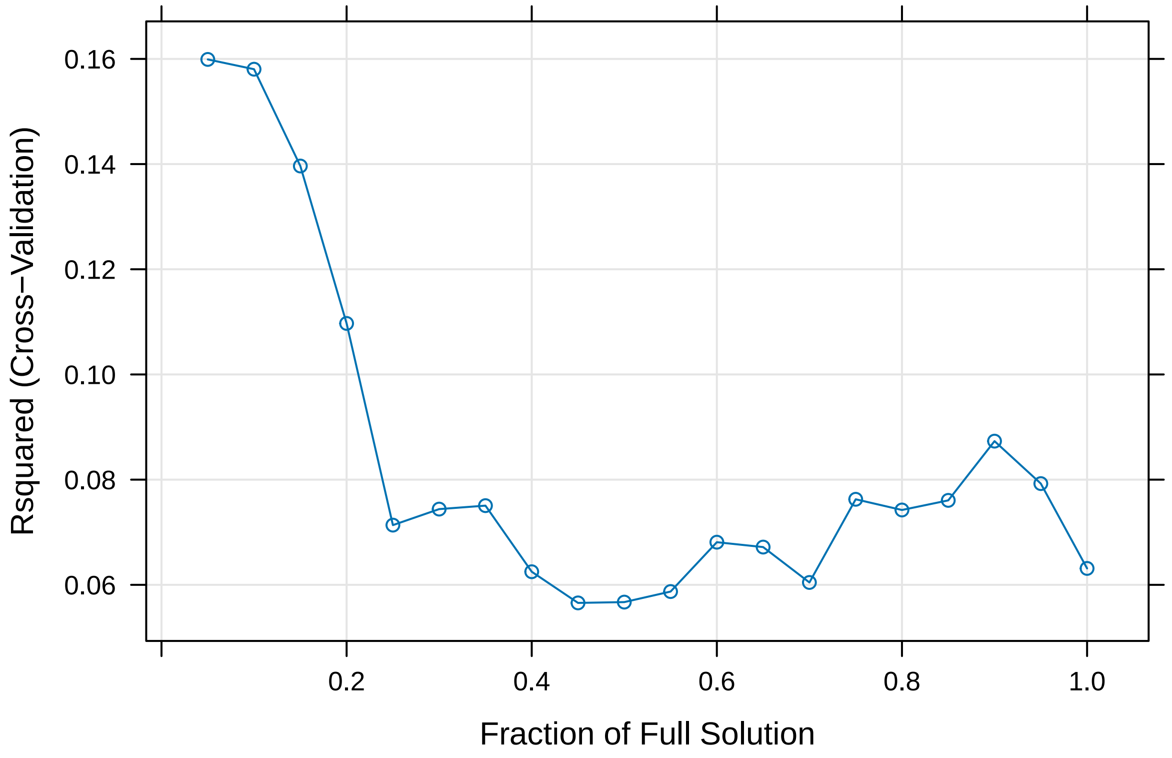
2. Lasso Method

```r
lasso_fit <- train(
  x = train_independent,
  y = train_dependent,
  method = "lasso",
  metric = "Rsquared",
  tuneGrid =  expand.grid(
    .fraction = seq(.05, 1, by = 0.05)
  ),
  trControl = trainControl(method = "cv", number = 10)
)

lasso_optimal <- lasso_fit$results |>
  arrange(Rsquared) |>
  tail(1)

lasso_test_results <- postResample(
  pred = predict(
    lasso_fit,
    test_independent
  ),
  test_dependent
)

plot(lasso_fit)
```



```
lasso_fit
```

```
## The lasso
##
## 133 samples
## 165 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 119, 118, 121, 120, 118, 120, ...
## Resampling results across tuning parameters:
##
##   fraction  RMSE         Rsquared    MAE
##   0.05      6.74300e+14  0.15990369  3.576401e+14
##   0.10      1.34860e+15  0.15804040  7.152803e+14
##   0.15      2.02290e+15  0.13963397  1.072920e+15
##   0.20      2.69720e+15  0.10970104  1.430561e+15
##   0.25      3.37150e+15  0.07135457  1.788201e+15
##   0.30      4.04580e+15  0.07439705  2.145841e+15
##   0.35      4.72010e+15  0.07505794  2.503481e+15
##   0.40      5.39440e+15  0.06249289  2.861121e+15
##   0.45      6.06870e+15  0.05656607  3.218761e+15
##   0.50      6.74300e+15  0.05671718  3.576401e+15
##   0.55      7.41730e+15  0.05872035  3.934041e+15
##   0.60      8.09160e+15  0.06810702  4.291682e+15
##   0.65      8.76590e+15  0.06717880  4.649322e+15
##   0.70      9.44020e+15  0.06045243  5.006962e+15
##   0.75      1.01145e+16  0.07626538  5.364602e+15
##   0.80      1.07888e+16  0.07423293  5.722242e+15
##   0.85      1.14631e+16  0.07605900  6.079882e+15
##   0.90      1.21374e+16  0.08732259  6.437522e+15
##   0.95      1.28117e+16  0.07924761  6.795163e+15
##   1.00      1.34860e+16  0.06312166  7.152803e+15
##
## Rsquared was used to select the optimal model using the largest value.
## The final value used for the model was fraction = 0.05.
```

Across this lasso model, our best R^2 is 0.16 with a fraction of 0.05.

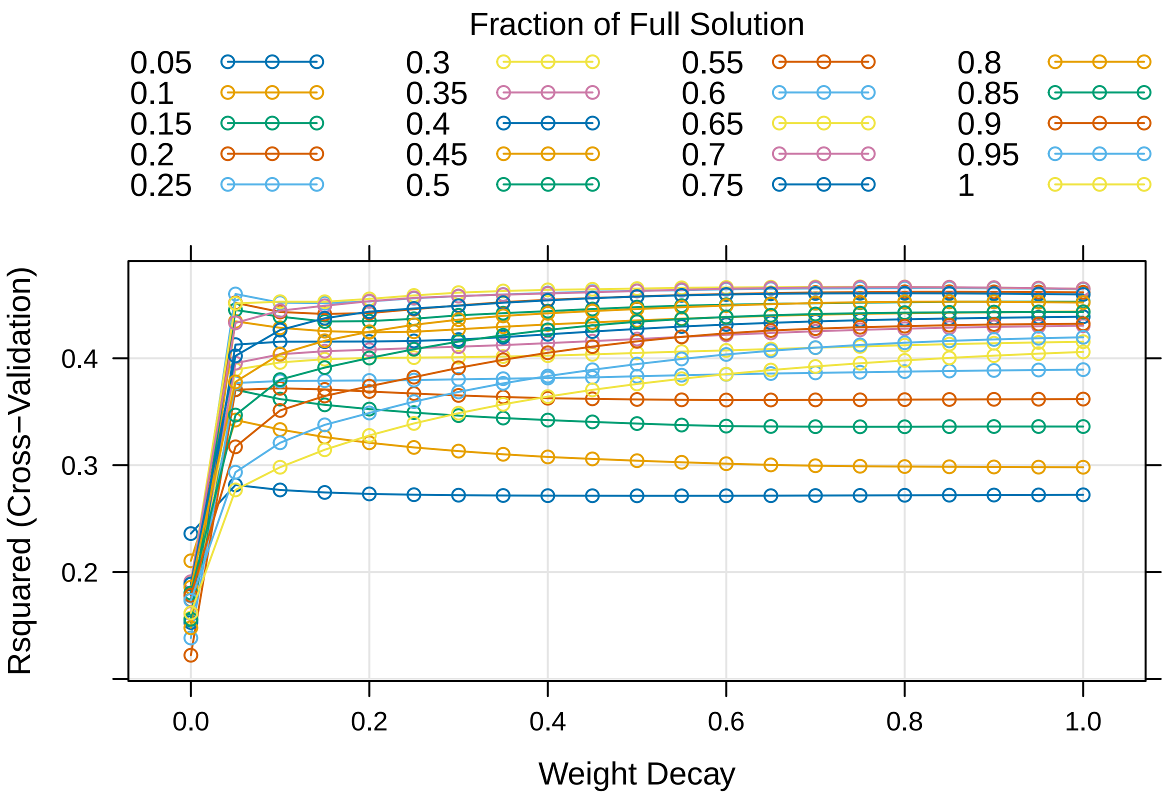
3. Elasticnet Method

```r
enet_fit <- train(
  x = train_independent,
  y = train_dependent,
  method = "enet",
  metric = "Rsquared",
  tuneGrid =  expand.grid(
    .fraction = seq(0.05, 1, by = 0.05),
    .lambda = seq(.00, 1, by = .05)
  ),
  trControl = trainControl(method = "cv", number = 10)
)

enet_test_results <- postResample(
  pred = predict(
    enet_fit,
    test_independent
  ),
  test_dependent
)

enet_optimal <- enet_fit$results |>
  arrange(Rsquared) |>
  tail(1)

plot(enet_fit)
```



```
enet_fit
```

```
## Elasticnet
##
## 133 samples
## 165 predictors
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 119, 121, 121, 120, 118, 120, ...
## Resampling results across tuning parameters:
##
##   lambda  fraction  RMSE          Rsquared   MAE
##   0.00    0.05      6.486159e+12  0.2359908  3.911765e+12
##   0.00    0.10      1.297231e+13  0.2105106  7.823523e+12
##   0.00    0.15      1.945846e+13  0.1549982  1.173528e+13
##   0.00    0.20      2.594462e+13  0.1221946  1.564704e+13
##   0.00    0.25      3.243077e+13  0.1383184  1.955880e+13
##   0.00    0.30      3.891692e+13  0.1569139  2.347055e+13
##   0.00    0.35      4.540307e+13  0.1612820  2.738231e+13
##   0.00    0.40      5.188923e+13  0.1527657  3.129407e+13
##   0.00    0.45      5.837538e+13  0.1478469  3.520583e+13
##   0.00    0.50      6.486153e+13  0.1559127  3.911758e+13
##   0.00    0.55      7.134768e+13  0.1740038  4.302934e+13
##   0.00    0.60      7.783384e+13  0.1775103  4.694110e+13
##   0.00    0.65      8.431999e+13  0.1901867  5.085286e+13
##   0.00    0.70      9.080614e+13  0.1909979  5.476461e+13
##   0.00    0.75      9.729229e+13  0.1887440  5.867637e+13
##   0.00    0.80      1.037784e+14  0.1858333  6.258813e+13
##   0.00    0.85      1.102646e+14  0.1803685  6.649989e+13
##   0.00    0.90      1.167508e+14  0.1782691  7.041165e+13
##   0.00    0.95      1.232369e+14  0.1735499  7.432340e+13
##   0.00    1.00      1.297231e+14  0.1617515  7.823516e+13
##   0.05    0.05      1.422682e+01  0.2814623  1.112783e+01
##   0.05    0.10      1.350199e+01  0.3422131  1.045710e+01
##   0.05    0.15      1.302966e+01  0.3705478  9.969193e+00
##   0.05    0.20      1.295208e+01  0.3705223  9.888175e+00
##   0.05    0.25      1.286096e+01  0.3768341  9.888698e+00
##   0.05    0.30      1.267456e+01  0.3893027  9.849772e+00
##   0.05    0.35      1.254085e+01  0.3954106  9.772786e+00
##   0.05    0.40      1.232038e+01  0.4127619  9.541054e+00
##   0.05    0.45      1.210752e+01  0.4345488  9.270514e+00
##   0.05    0.50      1.202693e+01  0.4451543  9.108616e+00
##   0.05    0.55      1.200814e+01  0.4518738  9.058280e+00
##   0.05    0.60      1.195417e+01  0.4601368  8.947780e+00
##   0.05    0.65      1.209661e+01  0.4513114  8.972336e+00
##   0.05    0.70      1.236352e+01  0.4331184  9.105007e+00
##   0.05    0.75      1.277803e+01  0.4021985  9.294780e+00
##   0.05    0.80      1.317977e+01  0.3778531  9.446969e+00
##   0.05    0.85      1.371441e+01  0.3468425  9.803302e+00
##   0.05    0.90      1.424772e+01  0.3171495  1.020339e+01
##   0.05    0.95      1.469349e+01  0.2933905  1.049369e+01
##   0.05    1.00      1.506614e+01  0.2768646  1.074521e+01
##   0.10    0.05      1.425049e+01  0.2767915  1.113858e+01
##   0.10    0.10      1.354657e+01  0.3333271  1.050347e+01
##   0.10    0.15      1.309137e+01  0.3618009  1.003830e+01
##   0.10    0.20      1.291774e+01  0.3718858  9.871365e+00
##   0.10    0.25      1.282885e+01  0.3787381  9.867547e+00
##   0.10    0.30      1.260950e+01  0.3961077  9.797797e+00
##   0.10    0.35      1.246346e+01  0.4037143  9.710611e+00
##   0.10    0.40      1.229724e+01  0.4154738  9.546394e+00
##   0.10    0.45      1.214797e+01  0.4284943  9.349724e+00
##   0.10    0.50      1.205263e+01  0.4389180  9.170933e+00
##   0.10    0.55      1.203887e+01  0.4433010  9.098789e+00
##   0.10    0.60      1.198980e+01  0.4520083  9.028938e+00
##   0.10    0.65      1.201851e+01  0.4529295  8.999594e+00
##   0.10    0.70      1.214234e+01  0.4445262  9.019318e+00
##   0.10    0.75      1.237643e+01  0.4263929  9.102084e+00
##   0.10    0.80      1.269587e+01  0.4040018  9.271346e+00
##   0.10    0.85      1.308828e+01  0.3800239  9.488056e+00
##   0.10    0.90      1.359586e+01  0.3511724  9.802332e+00
##   0.10    0.95      1.413050e+01  0.3209358  1.015429e+01
##   0.10    1.00      1.458993e+01  0.2980635  1.045878e+01
##   0.15    0.05      1.425360e+01  0.2744732  1.113204e+01
##   0.15    0.10      1.356670e+01  0.3262803  1.052030e+01
##   0.15    0.15      1.311501e+01  0.3564305  1.006650e+01
##   0.15    0.20      1.289734e+01  0.3708873  9.865388e+00
##   0.15    0.25      1.280382e+01  0.3789815  9.845570e+00
##   0.15    0.30      1.256444e+01  0.3991259  9.750142e+00
##   0.15    0.35      1.242287e+01  0.4065924  9.673238e+00
##   0.15    0.40      1.228823e+01  0.4155548  9.549510e+00
##   0.15    0.45      1.217153e+01  0.4253577  9.403388e+00
##   0.15    0.50      1.207280e+01  0.4344163  9.245601e+00
##   0.15    0.55      1.202627e+01  0.4414274  9.137193e+00
##   0.15    0.60      1.196693e+01  0.4515377  9.051883e+00
##   0.15    0.65      1.198048e+01  0.4528461  9.021159e+00
##   0.15    0.70      1.203489e+01  0.4491287  9.002398e+00
##   0.15    0.75      1.220145e+01  0.4374253  9.056807e+00
##   0.15    0.80      1.250042e+01  0.4163587  9.202649e+00
##   0.15    0.85      1.287017e+01  0.3912685  9.429015e+00
##   0.15    0.90      1.332520e+01  0.3647082  9.721223e+00
##   0.15    0.95      1.382304e+01  0.3376930  1.003575e+01
##   0.15    1.00      1.428838e+01  0.3144289  1.032202e+01
##   0.20    0.05      1.424905e+01  0.2730773  1.111890e+01
##   0.20    0.10      1.357622e+01  0.3209401  1.052813e+01
##   0.20    0.15      1.312531e+01  0.3524436  1.008035e+01
##   0.20    0.20      1.289022e+01  0.3688889  9.861646e+00
##   0.20    0.25      1.277865e+01  0.3792539  9.819867e+00
##   0.20    0.30      1.253083e+01  0.4002767  9.711098e+00
##   0.20    0.35      1.239525e+01  0.4079929  9.641489e+00
##   0.20    0.40      1.227970e+01  0.4156741  9.554439e+00
##   0.20    0.45      1.216906e+01  0.4242801  9.420796e+00
##   0.20    0.50      1.205854e+01  0.4347419  9.265947e+00
##   0.20    0.55      1.199991e+01  0.4422617  9.145311e+00
##   0.20    0.60      1.193265e+01  0.4531574  9.055606e+00
##   0.20    0.65      1.193371e+01  0.4555256  9.006779e+00
##   0.20    0.70      1.197445e+01  0.4533858  9.002366e+00
##   0.20    0.75      1.212558e+01  0.4435782  9.067344e+00
##   0.20    0.80      1.239509e+01  0.4248217  9.177479e+00
##   0.20    0.85      1.275070e+01  0.4002271  9.397691e+00
##   0.20    0.90      1.317726e+01  0.3738347  9.676120e+00
##   0.20    0.95      1.363679e+01  0.3487916  9.977786e+00
##   0.20    1.00      1.407981e+01  0.3277891  1.023908e+01
##   0.25    0.05      1.424001e+01  0.2723617  1.110327e+01
##   0.25    0.10      1.358064e+01  0.3166042  1.053153e+01
##   0.25    0.15      1.313170e+01  0.3491621  1.008532e+01
##   0.25    0.20      1.288311e+01  0.3669983  9.860750e+00
##   0.25    0.25      1.275303e+01  0.3795893  9.795708e+00
##   0.25    0.30      1.250743e+01  0.4005839  9.673928e+00
##   0.25    0.35      1.237064e+01  0.4094882  9.615319e+00
##   0.25    0.40      1.226648e+01  0.4162069  9.549744e+00
##   0.25    0.45      1.215411e+01  0.4247966  9.420621e+00
##   0.25    0.50      1.203168e+01  0.4369339  9.263026e+00
##   0.25    0.55      1.195411e+01  0.4459258  9.136328e+00
##   0.25    0.60      1.189507e+01  0.4561073  9.043075e+00
##   0.25    0.65      1.189893e+01  0.4587580  8.992247e+00
##   0.25    0.70      1.194837e+01  0.4564685  9.003746e+00
##   0.25    0.75      1.210069e+01  0.4465950  9.088949e+00
##   0.25    0.80      1.233557e+01  0.4312883  9.192541e+00
##   0.25    0.85      1.266507e+01  0.4084546  9.381086e+00
##   0.25    0.90      1.307438e+01  0.3823079  9.648976e+00
##   0.25    0.95      1.349957e+01  0.3594296  9.927710e+00
##   0.25    1.00      1.393130e+01  0.3390198  1.018148e+01
##   0.30    0.05      1.422919e+01  0.2719163  1.108686e+01
##   0.30   0.10    1.358050e+01  0.3131559  1.053111e+01
##   0.30   0.15    1.313467e+01  0.3463637  1.008455e+01
##   0.30   0.20    1.287434e+01  0.3652437  9.857395e+00
##   0.30   0.25    1.272552e+01  0.3803048  9.772766e+00
##   0.30   0.30    1.248648e+01  0.4010210  9.644426e+00
##   0.30   0.35    1.234870e+01  0.4108460  9.599024e+00
##   0.30   0.40    1.224655e+01  0.4175061  9.541293e+00
##   0.30   0.45    1.212524e+01  0.4271161  9.404701e+00
##   0.30   0.50    1.199673e+01  0.4401996  9.252838e+00
##   0.30   0.55    1.192132e+01  0.4494097  9.124667e+00
##   0.30   0.60    1.187267e+01  0.4581205  9.032867e+00
##   0.30   0.65    1.187689e+01  0.4613977  8.982668e+00
##   0.30   0.70    1.194413e+01  0.4582001  9.008495e+00
##   0.30   0.75    1.209313e+01  0.4491122  9.107560e+00
##   0.30   0.80    1.230437e+01  0.4361058  9.215194e+00
##   0.30   0.85    1.260785e+01  0.4156077  9.384070e+00
##   0.30   0.90    1.298569e+01  0.3910291  9.623145e+00
##   0.30   0.95    1.340578e+01  0.3685020  9.890398e+00
##   0.30   1.00    1.382528e+01  0.3486172  1.014501e+01
##   0.35   0.05    1.421740e+01  0.2716261  1.106992e+01
##   0.35   0.10    1.357831e+01  0.3101750  1.052716e+01
##   0.35   0.15    1.313230e+01  0.3440725  1.008234e+01
##   0.35   0.20    1.286304e+01  0.3637823  9.848340e+00
##   0.35   0.25    1.270275e+01  0.3808673  9.752162e+00
##   0.35   0.30    1.246580e+01  0.4016057  9.618541e+00
##   0.35   0.35    1.232669e+01  0.4122882  9.578115e+00
##   0.35   0.40    1.221998e+01  0.4197100  9.519786e+00
##   0.35   0.45    1.209891e+01  0.4294831  9.385344e+00
##   0.35   0.50    1.197722e+01  0.4421840  9.237482e+00
##   0.35   0.55    1.189843e+01  0.4524923  9.112964e+00
##   0.35   0.60    1.186253e+01  0.4596906  9.022559e+00
##   0.35   0.65    1.187230e+01  0.4629308  8.985431e+00
##   0.35   0.70    1.194998e+01  0.4595131  9.015973e+00
##   0.35   0.75    1.208997e+01  0.4519431  9.119503e+00
##   0.35   0.80    1.229607e+01  0.4396393  9.238484e+00
##   0.35   0.85    1.257562e+01  0.4215254  9.401561e+00
##   0.35   0.90    1.293154e+01  0.3985872  9.621415e+00
##   0.35   0.95    1.333577e+01  0.3765489  9.865997e+00
##   0.35   1.00    1.375111e+01  0.3569052  1.012085e+01
##   0.40   0.05    1.420486e+01  0.2714768  1.105285e+01
##   0.40   0.10    1.357537e+01  0.3076859  1.052369e+01
##   0.40   0.15    1.312696e+01  0.3421703  1.008147e+01
##   0.40   0.20    1.285085e+01  0.3626398  9.837384e+00
##   0.40   0.25    1.268220e+01  0.3815176  9.729893e+00
##   0.40   0.30    1.244666e+01  0.4024383  9.595104e+00
##   0.40   0.35    1.230207e+01  0.4141493  9.553942e+00
##   0.40   0.40    1.218991e+01  0.4224150  9.490808e+00
##   0.40   0.45    1.207746e+01  0.4316486  9.362053e+00
##   0.40   0.50    1.196299e+01  0.4440572  9.224483e+00
##   0.40   0.55    1.188813e+01  0.4547328  9.106509e+00
##   0.40   0.60    1.186147e+01  0.4610512  9.017930e+00
##   0.40   0.65    1.187880e+01  0.4639841  8.993455e+00
##   0.40   0.70    1.196443e+01  0.4606111  9.024255e+00
##   0.40   0.75    1.209900e+01  0.4541734  9.137969e+00
##   0.40    0.80      1.230844e+01  0.4420592  9.269804e+00
##   0.40    0.85      1.255994e+01  0.4265246  9.419653e+00
##   0.40    0.90      1.290003e+01  0.4051392  9.631608e+00
##   0.40    0.95      1.329135e+01  0.3833453  9.866533e+00
##   0.40    1.00      1.370180e+01  0.3641148  1.010830e+01
##   0.45    0.05      1.419237e+01  0.2713785  1.103628e+01
##   0.45    0.10      1.356955e+01  0.3058328  1.051727e+01
##   0.45    0.15      1.312108e+01  0.3404672  1.008200e+01
##   0.45    0.20      1.283584e+01  0.3620387  9.821614e+00
##   0.45    0.25      1.266344e+01  0.3820801  9.709709e+00
##   0.45    0.30      1.242617e+01  0.4036200  9.573151e+00
##   0.45    0.35      1.227770e+01  0.4160380  9.528449e+00
##   0.45    0.40      1.216345e+01  0.4249883  9.462123e+00
##   0.45    0.45      1.206271e+01  0.4334666  9.339668e+00
##   0.45    0.50      1.195186e+01  0.4460584  9.209683e+00
##   0.45    0.55      1.188687e+01  0.4563535  9.101481e+00
##   0.45    0.60      1.186621e+01  0.4623181  9.019715e+00
##   0.45    0.65      1.189470e+01  0.4645587  9.006514e+00
##   0.45    0.70      1.198423e+01  0.4617210  9.037137e+00
##   0.45    0.75      1.211545e+01  0.4561719  9.161123e+00
##   0.45    0.80      1.232902e+01  0.4441404  9.304633e+00
##   0.45    0.85      1.255874e+01  0.4307100  9.448665e+00
##   0.45    0.90      1.288373e+01  0.4108955  9.645163e+00
##   0.45    0.95      1.326946e+01  0.3892165  9.877482e+00
##   0.45    1.00      1.367255e+01  0.3704199  1.011931e+01
##   0.50    0.05      1.417876e+01  0.2713463  1.101985e+01
##   0.50    0.10      1.356315e+01  0.3041725  1.050970e+01
##   0.50    0.15      1.311506e+01  0.3389170  1.008095e+01
##   0.50    0.20      1.282225e+01  0.3614605  9.806086e+00
##   0.50    0.25      1.263982e+01  0.3831650  9.687057e+00
##   0.50    0.30      1.240673e+01  0.4049849  9.554325e+00
##   0.50    0.35      1.225489e+01  0.4179739  9.502187e+00
##   0.50    0.40      1.214079e+01  0.4274323  9.431991e+00
##   0.50    0.45      1.204934e+01  0.4354047  9.320188e+00
##   0.50    0.50      1.194732e+01  0.4478721  9.204056e+00
##   0.50    0.55      1.188909e+01  0.4578192  9.098945e+00
##   0.50    0.60      1.188177e+01  0.4630685  9.026497e+00
##   0.50    0.65      1.191633e+01  0.4651681  9.019073e+00
##   0.50    0.70      1.200816e+01  0.4629472  9.056131e+00
##   0.50    0.75      1.213939e+01  0.4577183  9.187067e+00
##   0.50    0.80      1.235596e+01  0.4460494  9.338312e+00
##   0.50    0.85      1.257154e+01  0.4341705  9.482722e+00
##   0.50    0.90      1.288217e+01  0.4158390  9.679571e+00
##   0.50    0.95      1.326536e+01  0.3945442  9.906129e+00
##   0.50    1.00      1.365985e+01  0.3759564  1.014127e+01
##   0.55    0.05      1.416526e+01  0.2713350  1.100363e+01
##   0.55    0.10      1.355630e+01  0.3026927  1.050101e+01
##   0.55    0.15      1.310844e+01  0.3375299  1.007760e+01
##   0.55    0.20      1.280737e+01  0.3611299  9.791620e+00
##   0.55    0.25      1.261934e+01  0.3841388  9.667271e+00
##   0.55    0.30      1.239008e+01  0.4062281  9.537570e+00
##   0.55    0.35      1.223422e+01  0.4199943  9.475040e+00
##   0.55    0.40      1.212244e+01  0.4296779  9.398963e+00
##   0.55    0.45      1.204364e+01  0.4370219  9.304922e+00
##   0.55   0.50    1.195078e+01  0.4491805  9.199793e+00
##   0.55   0.55    1.189613e+01  0.4591622  9.101283e+00
##   0.55   0.60    1.190287e+01  0.4636237  9.033254e+00
##   0.55   0.65    1.194226e+01  0.4659004  9.031571e+00
##   0.55   0.70    1.203395e+01  0.4640803  9.083638e+00
##   0.55   0.75    1.217145e+01  0.4588406  9.218203e+00
##   0.55   0.80    1.238805e+01  0.4477529  9.379685e+00
##   0.55   0.85    1.260067e+01  0.4366286  9.528911e+00
##   0.55   0.90    1.289440e+01  0.4199969  9.727948e+00
##   0.55   0.95    1.327354e+01  0.3993378  9.950649e+00
##   0.55   1.00    1.366106e+01  0.3808334  1.016793e+01
##   0.60   0.05    1.415176e+01  0.2713679  1.098762e+01
##   0.60   0.10    1.354929e+01  0.3013699  1.049146e+01
##   0.60   0.15    1.309918e+01  0.3365364  1.007133e+01
##   0.60   0.20    1.279308e+01  0.3609592  9.779850e+00
##   0.60   0.25    1.260295e+01  0.3849199  9.649483e+00
##   0.60   0.30    1.237624e+01  0.4073553  9.522908e+00
##   0.60   0.35    1.221668e+01  0.4219223  9.450249e+00
##   0.60   0.40    1.210981e+01  0.4315571  9.373133e+00
##   0.60   0.45    1.204408e+01  0.4383838  9.294914e+00
##   0.60   0.50    1.196125e+01  0.4500755  9.197140e+00
##   0.60   0.55    1.191012e+01  0.4601186  9.105243e+00
##   0.60   0.60    1.192582e+01  0.4645178  9.043657e+00
##   0.60   0.65    1.197558e+01  0.4662452  9.054592e+00
##   0.60   0.70    1.206419e+01  0.4650583  9.121107e+00
##   0.60   0.75    1.221138e+01  0.4596620  9.257167e+00
##   0.60   0.80    1.242553e+01  0.4493156  9.420949e+00
##   0.60   0.85    1.263866e+01  0.4386194  9.582740e+00
##   0.60   0.90    1.292086e+01  0.4233542  9.779013e+00
##   0.60   0.95    1.329184e+01  0.4035649  9.999209e+00
##   0.60   1.00    1.367415e+01  0.3851401  1.020974e+01
##   0.65   0.05    1.413815e+01  0.2714450  1.097184e+01
##   0.65   0.10    1.354179e+01  0.3002460  1.048109e+01
##   0.65   0.15    1.308485e+01  0.3361956  1.006331e+01
##   0.65   0.20    1.277871e+01  0.3610039  9.766928e+00
##   0.65   0.25    1.258810e+01  0.3857077  9.632716e+00
##   0.65   0.30    1.236282e+01  0.4086223  9.506410e+00
##   0.65   0.35    1.220302e+01  0.4236624  9.426946e+00
##   0.65   0.40    1.210233e+01  0.4331339  9.350344e+00
##   0.65   0.45    1.204795e+01  0.4396968  9.285443e+00
##   0.65   0.50    1.197553e+01  0.4508995  9.198436e+00
##   0.65   0.55    1.192923e+01  0.4609567  9.112733e+00
##   0.65   0.60    1.195556e+01  0.4650075  9.058026e+00
##   0.65   0.65    1.201232e+01  0.4664763  9.084537e+00
##   0.65   0.70    1.210186e+01  0.4656245  9.168137e+00
##   0.65   0.75    1.225599e+01  0.4603059  9.306675e+00
##   0.65   0.80    1.246916e+01  0.4506177  9.460931e+00
##   0.65   0.85    1.268315e+01  0.4401998  9.636466e+00
##   0.65   0.90    1.295996e+01  0.4259583  9.832392e+00
##   0.65   0.95    1.332182e+01  0.4069952  1.004947e+01
##   0.65   1.00    1.369750e+01  0.3889504  1.025597e+01
##   0.70   0.05    1.412437e+01  0.2715813  1.095542e+01
##   0.70   0.10    1.353226e+01  0.2994622  1.046847e+01
##   0.70   0.15    1.307019e+01  0.3359978  1.005438e+01
##   0.70    0.20      1.276563e+01  0.3610738  9.753943e+00
##   0.70    0.25      1.257601e+01  0.3863140  9.616926e+00
##   0.70    0.30      1.235179e+01  0.4098760  9.489591e+00
##   0.70    0.35      1.219311e+01  0.4252529  9.408943e+00
##   0.70    0.40      1.209978e+01  0.4345012  9.330757e+00
##   0.70    0.45      1.205706e+01  0.4407439  9.277250e+00
##   0.70    0.50      1.199469e+01  0.4514727  9.204530e+00
##   0.70    0.55      1.195474e+01  0.4615495  9.122790e+00
##   0.70    0.60      1.198953e+01  0.4653074  9.073962e+00
##   0.70    0.65      1.205074e+01  0.4667139  9.124855e+00
##   0.70    0.70      1.214441e+01  0.4661517  9.214343e+00
##   0.70    0.75      1.230641e+01  0.4606445  9.357193e+00
##   0.70    0.80      1.251719e+01  0.4516695  9.514891e+00
##   0.70    0.85      1.273401e+01  0.4414017  9.691269e+00
##   0.70    0.90      1.301127e+01  0.4277042  9.887712e+00
##   0.70    0.95      1.335926e+01  0.4099463  1.009894e+01
##   0.70    1.00      1.372979e+01  0.3923267  1.030283e+01
##   0.75    0.05      1.411086e+01  0.2717167  1.093977e+01
##   0.75    0.10      1.352088e+01  0.2989788  1.045620e+01
##   0.75    0.15      1.305572e+01  0.3359075  1.004436e+01
##   0.75    0.20      1.275473e+01  0.3611122  9.741930e+00
##   0.75    0.25      1.256510e+01  0.3869253  9.600354e+00
##   0.75    0.30      1.234294e+01  0.4110370  9.473296e+00
##   0.75    0.35      1.218766e+01  0.4265823  9.392198e+00
##   0.75    0.40      1.210260e+01  0.4355845  9.315367e+00
##   0.75    0.45      1.207097e+01  0.4415385  9.273016e+00
##   0.75    0.50      1.201749e+01  0.4519765  9.214470e+00
##   0.75    0.55      1.198520e+01  0.4618739  9.139223e+00
##   0.75    0.60      1.202707e+01  0.4654667  9.111069e+00
##   0.75    0.65      1.209361e+01  0.4667958  9.166019e+00
##   0.75    0.70      1.219206e+01  0.4664171  9.260517e+00
##   0.75    0.75      1.236149e+01  0.4607721  9.405860e+00
##   0.75    0.80      1.257010e+01  0.4524584  9.572971e+00
##   0.75    0.85      1.279279e+01  0.4421540  9.757491e+00
##   0.75    0.90      1.307049e+01  0.4289938  9.951357e+00
##   0.75    0.95      1.340383e+01  0.4124697  1.014951e+01
##   0.75    1.00      1.376994e+01  0.3953217  1.036101e+01
##   0.80    0.05      1.409788e+01  0.2718347  1.092593e+01
##   0.80    0.10      1.350904e+01  0.2987043  1.044337e+01
##   0.80    0.15      1.304164e+01  0.3359224  1.003329e+01
##   0.80    0.20      1.274406e+01  0.3612872  9.730552e+00
##   0.80    0.25      1.255533e+01  0.3875245  9.583756e+00
##   0.80    0.30      1.233659e+01  0.4121805  9.457841e+00
##   0.80    0.35      1.218612e+01  0.4276802  9.376752e+00
##   0.80    0.40      1.210971e+01  0.4364607  9.302328e+00
##   0.80    0.45      1.208922e+01  0.4421376  9.273826e+00
##   0.80    0.50      1.204313e+01  0.4524331  9.226017e+00
##   0.80    0.55      1.201877e+01  0.4621520  9.159563e+00
##   0.80    0.60      1.206703e+01  0.4656339  9.150028e+00
##   0.80    0.65      1.214217e+01  0.4666457  9.209271e+00
##   0.80    0.70      1.224365e+01  0.4665347  9.306102e+00
##   0.80    0.75      1.241966e+01  0.4608078  9.457795e+00
##   0.80    0.80      1.262889e+01  0.4528555  9.632661e+00
##   0.80    0.85      1.285785e+01  0.4426655  9.823891e+00
##   0.80    0.90       1.313449e+01  0.4300317  1.002410e+01
##   0.80    0.95       1.345670e+01  0.4145388  1.023242e+01
##   0.80    1.00       1.381702e+01  0.3979808  1.044330e+01
##   0.85    0.05       1.408533e+01  0.2719397  1.091310e+01
##   0.85    0.10       1.349754e+01  0.2984865  1.043153e+01
##   0.85    0.15       1.302782e+01  0.3360451  1.002104e+01
##   0.85    0.20       1.273474e+01  0.3614543  9.719505e+00
##   0.85    0.25       1.254750e+01  0.3880937  9.570459e+00
##   0.85    0.30       1.233261e+01  0.4131874  9.446028e+00
##   0.85    0.35       1.218759e+01  0.4286189  9.361702e+00
##   0.85    0.40       1.212091e+01  0.4371934  9.290656e+00
##   0.85    0.45       1.211114e+01  0.4425855  9.277916e+00
##   0.85    0.50       1.207195e+01  0.4528243  9.242991e+00
##   0.85    0.55       1.205702e+01  0.4622118  9.186019e+00
##   0.85    0.60       1.211152e+01  0.4656653  9.192798e+00
##   0.85    0.65       1.219453e+01  0.4663699  9.255643e+00
##   0.85    0.70       1.229973e+01  0.4664435  9.351648e+00
##   0.85    0.75       1.248083e+01  0.4607068  9.508907e+00
##   0.85    0.80       1.269309e+01  0.4529570  9.693419e+00
##   0.85    0.85       1.292619e+01  0.4429716  9.902224e+00
##   0.85    0.90       1.320279e+01  0.4309107  1.011490e+01
##   0.85    0.95       1.351636e+01  0.4162468  1.031970e+01
##   0.85    1.00       1.387026e+01  0.4003427  1.053046e+01
##   0.90    0.05       1.407313e+01  0.2720397  1.090060e+01
##   0.90    0.10       1.348592e+01  0.2983037  1.042023e+01
##   0.90    0.15       1.301507e+01  0.3361411  1.000893e+01
##   0.90    0.20       1.272686e+01  0.3615845  9.709565e+00
##   0.90    0.25       1.254165e+01  0.3885873  9.557536e+00
##   0.90    0.30       1.233173e+01  0.4140532  9.434361e+00
##   0.90    0.35       1.219397e+01  0.4293254  9.347821e+00
##   0.90    0.40       1.213476e+01  0.4378298  9.287289e+00
##   0.90    0.45       1.213513e+01  0.4430559  9.289114e+00
##   0.90    0.50       1.210605e+01  0.4529381  9.265730e+00
##   0.90    0.55       1.209935e+01  0.4620848  9.216574e+00
##   0.90    0.60       1.216119e+01  0.4654237  9.237757e+00
##   0.90    0.65       1.225139e+01  0.4659597  9.304433e+00
##   0.90    0.70       1.236064e+01  0.4660662  9.398283e+00
##   0.90    0.75       1.254537e+01  0.4604617  9.569852e+00
##   0.90    0.80       1.276227e+01  0.4528043  9.773193e+00
##   0.90    0.85       1.299801e+01  0.4431595  9.989475e+00
##   0.90    0.90       1.327578e+01  0.4315608  1.020810e+01
##   0.90    0.95       1.358271e+01  0.4176284  1.040905e+01
##   0.90    1.00       1.392900e+01  0.4024412  1.061775e+01
##   0.95    0.05       1.406130e+01  0.2721343  1.088901e+01
##   0.95    0.10       1.347458e+01  0.2981677  1.040948e+01
##   0.95    0.15       1.300439e+01  0.3361461  9.997606e+00
##   0.95    0.20       1.272018e+01  0.3617018  9.699875e+00
##   0.95    0.25       1.253772e+01  0.3890145  9.546228e+00
##   0.95    0.30       1.233346e+01  0.4147976  9.424205e+00
##   0.95    0.35       1.220316e+01  0.4299316  9.340350e+00
##   0.95    0.40       1.215126e+01  0.4384123  9.289483e+00
##   0.95    0.45       1.216199e+01  0.4434280  9.303680e+00
##   0.95    0.50       1.214366e+01  0.4529188  9.290544e+00
##   0.95    0.55       1.214365e+01  0.4619606  9.253977e+00
##   0.95    0.60      1.221397e+01  0.4650698  9.288722e+00
##   0.95    0.65      1.231093e+01  0.4655243  9.359203e+00
##   0.95    0.70      1.242430e+01  0.4655866  9.457968e+00
##   0.95    0.75      1.261330e+01  0.4600917  9.649244e+00
##   0.95    0.80      1.283538e+01  0.4524941  9.856989e+00
##   0.95    0.85      1.307279e+01  0.4432776  1.007749e+01
##   0.95    0.90      1.335242e+01  0.4320004  1.029906e+01
##   0.95    0.95      1.365481e+01  0.4187212  1.049786e+01
##   0.95    1.00      1.399264e+01  0.4043056  1.070623e+01
##   1.00    0.05      1.404926e+01  0.2722558  1.087756e+01
##   1.00    0.10      1.346367e+01  0.2980677  1.039964e+01
##   1.00    0.15      1.299485e+01  0.3361299  9.986480e+00
##   1.00    0.20      1.271460e+01  0.3618324  9.690580e+00
##   1.00    0.25      1.253578e+01  0.3893695  9.536520e+00
##   1.00    0.30      1.233732e+01  0.4154598  9.417740e+00
##   1.00    0.35      1.221505e+01  0.4304367  9.335589e+00
##   1.00    0.40      1.217095e+01  0.4388632  9.292716e+00
##   1.00    0.45      1.219242e+01  0.4436702  9.323728e+00
##   1.00    0.50      1.218298e+01  0.4528537  9.323656e+00
##   1.00    0.55      1.219126e+01  0.4616994  9.294066e+00
##   1.00    0.60      1.227063e+01  0.4645094  9.342434e+00
##   1.00    0.65      1.237371e+01  0.4649390  9.415809e+00
##   1.00    0.70      1.249146e+01  0.4649448  9.527784e+00
##   1.00    0.75      1.268384e+01  0.4597256  9.730878e+00
##   1.00    0.80      1.291063e+01  0.4521243  9.938872e+00
##   1.00    0.85      1.315002e+01  0.4432985  1.016450e+01
##   1.00    0.90      1.343142e+01  0.4322802  1.038747e+01
##   1.00    0.95      1.373102e+01  0.4196356  1.058735e+01
##   1.00    1.00      1.406068e+01  0.4059613  1.079379e+01
##
## Rsquared was used to select the optimal model using the largest value.
## The final values used for the model were fraction = 0.65 and lambda = 0.75.
```

Across these Elasticnet models, our best R^2 on the test dataset is 0.47 using the PLS model.

I wouldn't recommend using any of these models as their testing MAE is very high relative to the predicted values, meaning that the actual value can greatly vary from our predictions.

```
test_results
```

(f) Would you recommend any of your models to replace the permeability laboratory experiment?

```
##       RMSE   Rsquared        MAE
## 10.2106903  0.6890925  7.6501522
```

```
ridge_test_results
```

```
##       RMSE   Rsquared        MAE
## 17.6706208  0.6859879 13.9072273
```

```
lasso_test_results
```

```
##         RMSE     Rsquared          MAE
## 4.038259e+17 3.112027e-03 1.643742e+17
```

```
enet_test_results
```

```
##       RMSE   Rsquared        MAE
## 10.6033771  0.6715131  7.4653516
```

From our models above, the best performing one is the PLS model performed the best with a R^2 of 0.69.

Question 6.3

6.3. A chemical manufacturing process for a pharmaceutical product was discussed in Sect. 1.4. In this problem, the objective is to understand the relationship between biological measurements of the raw materials (predictors), measurements of the manufacturing process (predictors), and the response of product yield. Biological predictors cannot be changed but can be used to assess the quality of the raw material before processing. On the other hand, manufacturing process predictors can be changed in the manufacturing process. Improving product yield by 1 % will boost revenue by approximately one hundred thousand dollars per batch:

```r
library(AppliedPredictiveModeling)
data(ChemicalManufacturingProcess)
```

(a) Start R and use these commands to load the data: The matrix processPredictors contains the 57 predictors (12 describing the input biological material and 45 describing the process predictors) for the 176 manufacturing runs. yield contains the percent yield for each run.

(b) A small percentage of cells in the predictor set contain missing values. Use an imputation function to fill in these missing values (e.g., see Sect. 3.8). In the `preprocess` function, we can specify a NA fill method. I believe using a KNN would be a good method:

```r
sum(is.na(ChemicalManufacturingProcess[, -c(1)]))
```

```
## [1] 106
```

```r
knn_impute <- preProcess(
  ChemicalManufacturingProcess[, -c(1)],
  method = "knnImpute"
)

cmp_independent <- predict(
  knn_impute,
  ChemicalManufacturingProcess[, -c(1)]
)
```

```r
cmp_dependent <- ChemicalManufacturingProcess[, c(1), drop=FALSE]

sum(is.na(cmp_independent[, -c(1)]))
```

```
## [1] 0
```

```r


set.seed(19940211)
# Partition the data into a sample of 80% of the full dataset
cmp_train_rows <- createDataPartition(
  cmp_independent$BiologicalMaterial01,
  p = 0.8,
  list = FALSE
)

# Use the sample to create a training dataset
cmp_train_ind <- cmp_independent[cmp_train_rows, ]
cmp_train_dep <- cmp_dependent[cmp_train_rows]

str(train_independent)
```

(c) Split the data into a training and a test set, pre-process the data, and tune a model of your choice from this chapter. What is the optimal value of the performance metric?

```
##  num [1:133, 1:165] -5.87 -4.97 -2.78 -4.61 -4.96 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:133] "1" "2" "3" "4" ...
##   ..$ : chr [1:165] "PC1" "PC2" "PC3" "PC4" ...
```

```r
# Use the sample to create a testing dataset
cmp_test_ind <- cmp_independent[-cmp_train_rows, ]
cmp_test_dep <- cmp_dependent[-cmp_train_rows]

str(test_independent)
```

```
##  num [1:32, 1:165] -7.15 -10 -5.09 15.31 -6.8 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:32] "8" "9" "10" "20" ...
##   ..$ : chr [1:165] "PC1" "PC2" "PC3" "PC4" ...
```

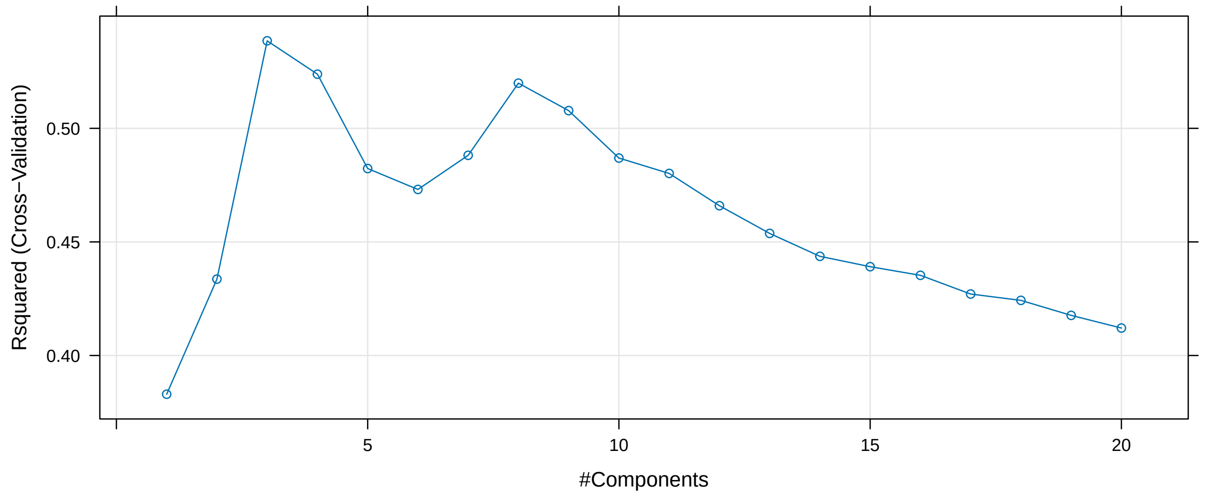
```r
# Tuning/Fitting our model
cmp_pls_fit <- train(
  x = cmp_train_ind,
  y = cmp_train_dep,
  method = "pls",
  metric = "Rsquared",
  tuneLength = 20,
  trControl = trainControl(method = "cv", number = 10),
  preProcess = c(
    "center",
    "scale"
  )
)

cmp_pls_optimal_result <- cmp_pls_fit$results |>
  arrange(desc(Rsquared)) |>
  head(1)

cmp_pls_fit$results
```

```
##    ncomp     RMSE  Rsquared      MAE    RMSESD RsquaredSD     MAESD
## 1      1 1.580071 0.3829472 1.192772 0.5123706  0.2334273 0.2079890
## 2      2 2.126260 0.4336151 1.252441 1.9190862  0.2913773 0.6113283
## 3      3 1.312699 0.5385236 1.004375 0.4835958  0.2009453 0.2339276
## 4      4 1.347443 0.5238455 1.050064 0.4267744  0.1920633 0.2535925
## 5      5 1.655898 0.4823007 1.144655 1.0593779  0.2372707 0.4501137
## 6      6 1.669474 0.4730924 1.151734 1.0482871  0.2421177 0.4428813
## 7      7 1.616966 0.4881266 1.137010 1.0832847  0.2212372 0.4149625
## 8      8 1.655856 0.5198908 1.136405 1.4280055  0.2076757 0.4710184
## 9      9 1.851990 0.5077873 1.183778 1.9787627  0.2200201 0.6051466
## 10    10 2.020882 0.4869231 1.224026 2.3558178  0.2526423 0.7062781
## 11    11 2.178807 0.4801467 1.279488 2.6407203  0.2665653 0.7812925
## 12    12 2.354369 0.4659388 1.350860 2.8627962  0.2709910 0.8501395
## 13    13 2.414094 0.4537372 1.384310 2.9633345  0.2691560 0.8791279
## 14    14 2.484869 0.4436561 1.406640 3.0394217  0.2753261 0.9092997
## 15    15 2.512374 0.4390887 1.414258 3.0040594  0.2701615 0.8947635
## 16    16 2.523327 0.4352910 1.408737 2.9281151  0.2701676 0.8915870
## 17    17 2.466160 0.4270522 1.390705 2.8232428  0.2726033 0.8675907
## 18    18 2.420862 0.4242607 1.383694 2.7553439  0.2706446 0.8533477
## 19    19 2.459239 0.4176530 1.401009 2.7390655  0.2732458 0.8558153
## 20    20 2.543295 0.4120774 1.428165 2.7622367  0.2777628 0.8690816
```

```r
plot(cmp_pls_fit)
```



```
cmp_pls_fit
```

```
## Partial Least Squares
##
## 143 samples
##  57 predictor
##
## Pre-processing: centered (57), scaled (57)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 130, 127, 129, 129, 128, 129, ...
## Resampling results across tuning parameters:
##
##   ncomp  RMSE      Rsquared   MAE
##    1     1.580071  0.3829472  1.192772
##    2     2.126260  0.4336151  1.252441
##    3     1.312699  0.5385236  1.004375
##    4     1.347443  0.5238455  1.050064
##    5     1.655898  0.4823007  1.144655
##    6     1.669474  0.4730924  1.151734
##    7     1.616966  0.4881266  1.137010
##    8     1.655856  0.5198908  1.136405
##    9     1.851990  0.5077873  1.183778
##   10     2.020882  0.4869231  1.224026
##   11     2.178807  0.4801467  1.279488
##   12     2.354369  0.4659388  1.350860
##   13     2.414094  0.4537372  1.384310
##   14     2.484869  0.4436561  1.406640
##   15     2.512374  0.4390887  1.414258
##   16     2.523327  0.4352910  1.408737
##   17     2.466160  0.4270522  1.390705
##   18     2.420862  0.4242607  1.383694
##   19     2.459239  0.4176530  1.401009
##   20     2.543295  0.4120774  1.428165
##
## Rsquared was used to select the optimal model using the largest value.
## The final value used for the model was ncomp = 3.
```

The above chunk of code uses a PLS model and we can see that the optimal value of this model is 3 with an R^2 of 0.54.

```
# Get predictions on the testing dataset
cmp_pls_test_pred <- predict(
  cmp_pls_fit,
  cmp_test_ind,
  ncomp = cmp_pls_optimal_result$ncomp
)

# Use postResample to compare results
cmp_pls_test_results <- postResample(
  pred = cmp_pls_test_pred,
```

```r
  obs = cmp_test_dep
)

cmp_pls_test_results[["Rsquared"]]
```

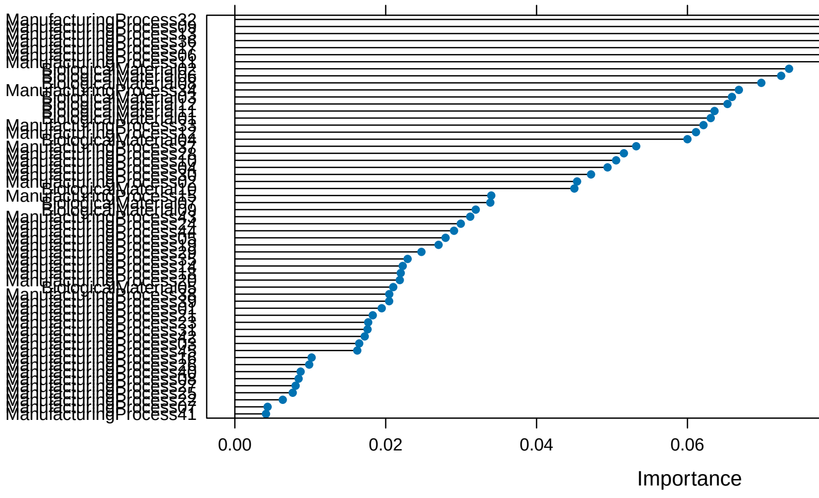
(d) Predict the response for the test set. What is the value of the performance metric and how does this compare with the resampled performance metric on the training set?

```
## [1] 0.6150737
```

From the above test, we can see that the test R^2 is 0.62 which is greater but similar to that of the training set.

```r
cmp_pls_importance <- varImp(cmp_pls_fit, scale = FALSE)
plot(cmp_pls_importance)
```

(e) Which predictors are most important in the model you have trained? Do either the biological or process predictors dominate the list?



From the above plot, we can see that the most important predictors are (in order): 1. `ManufacturingProcess32` 2. `ManufacturingProcess36` 3. `ManufacturingProcess13` 4. `ManufacturingProcess17` 5. `ManufacturingProcess09`

The `ManufacturingProcess` predictors comprise 8 of the top 10 predictors. With the top 5 accounting for 60.12% of total importance

(f) Explore the relationships between each of the top predictors and the response. How could this information be helpful in improving yield in future runs of the manufacturing process? As a rule of thumb, a larger the coefficient is the more important it is to the model. So we can use that information to indicate how important certain steps or materials are.

If it is negative, then it's an indication that we may be able to better identify poor runs or highlight areas that need improvement/change to increase the reliability of the process.

```
coef(cmp_pls_fit$finalModel)
```

```
## , , 3 comps
##
##                             .outcome
## BiologicalMaterial01   -0.006555801
## BiologicalMaterial02    0.069590522
## BiologicalMaterial03    0.111233335
## BiologicalMaterial04    0.048986151
## BiologicalMaterial05   -0.008553501
## BiologicalMaterial06    0.099152494
## BiologicalMaterial07   -0.139489082
## BiologicalMaterial08   -0.008167224
## BiologicalMaterial09   -0.074555620
## BiologicalMaterial10   -0.049042032
## BiologicalMaterial11   -0.012086241
## BiologicalMaterial12    0.019828331
## ManufacturingProcess01  0.015822002
## ManufacturingProcess02 -0.032997234
## ManufacturingProcess03 -0.052999764
## ManufacturingProcess04  0.068489058
## ManufacturingProcess05 -0.005791908
## ManufacturingProcess06  0.190573944
## ManufacturingProcess07 -0.020653639
## ManufacturingProcess08 -0.033037880
## ManufacturingProcess09  0.266492898
## ManufacturingProcess10  0.044729941
## ManufacturingProcess11  0.156517092
## ManufacturingProcess12  0.070622068
## ManufacturingProcess13 -0.211628559
## ManufacturingProcess14  0.034086884
## ManufacturingProcess15  0.126656607
## ManufacturingProcess16 -0.023356789
## ManufacturingProcess17 -0.233447504
## ManufacturingProcess18  0.058626531
## ManufacturingProcess19  0.019076617
## ManufacturingProcess20  0.050343576
## ManufacturingProcess21 -0.098824095
## ManufacturingProcess22 -0.039751293
## ManufacturingProcess23 -0.013417105
## ManufacturingProcess24 -0.028694802
## ManufacturingProcess25 -0.006201066
## ManufacturingProcess26  0.009627655
## ManufacturingProcess27 -0.008743357
## ManufacturingProcess28 -0.036866468
## ManufacturingProcess29  0.046591164
## ManufacturingProcess30  0.104108850
## ManufacturingProcess31 -0.027971305
## ManufacturingProcess32  0.342895324
## ManufacturingProcess33  0.137938297
## ManufacturingProcess34  0.261267222
## ManufacturingProcess35 -0.038464692
## ManufacturingProcess36 -0.273358790
## ManufacturingProcess37 -0.195304087
## ManufacturingProcess38 -0.040014732
## ManufacturingProcess39  0.111977542
## ManufacturingProcess40 -0.030604627
## ManufacturingProcess41 -0.019269842
## ManufacturingProcess42  0.095404293
## ManufacturingProcess43  0.075813284
## ManufacturingProcess44  0.131529404
## ManufacturingProcess45  0.096036821
```