Team 2 - Homework Two

Assignment Two Kuhn and Johnson 3.6

Bethany Poulin

10/23/19

Dependencies

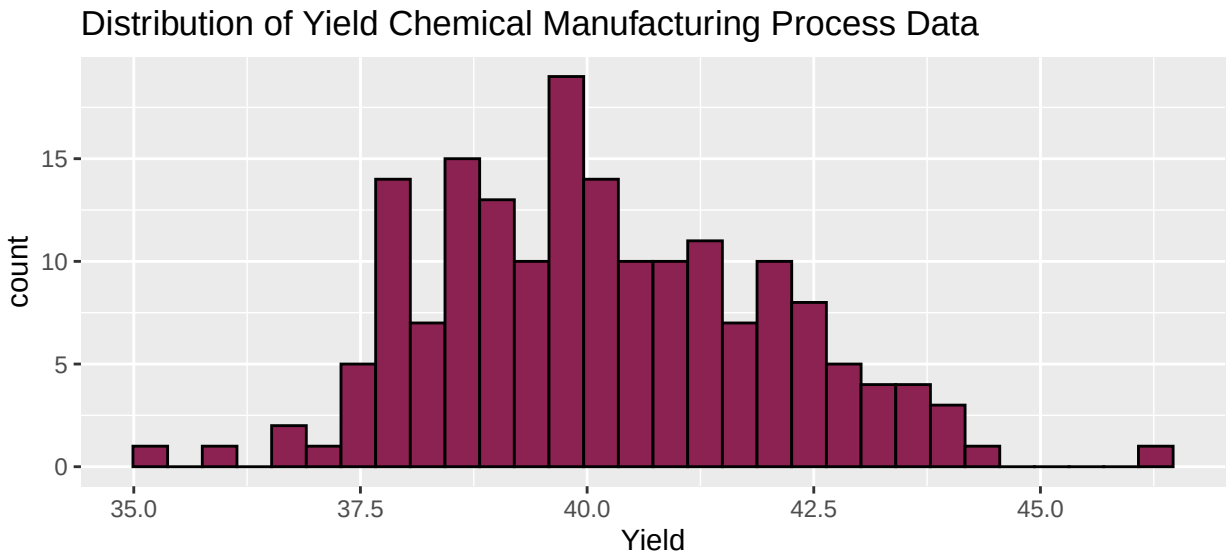
```
# Predicitve Modeling
libraries('AppliedPredictiveModeling', 'tidyverse', 'impute', 'caTools', 'pls')

# Formatting Libraries
libraries('default', 'knitr', 'kableExtra')

# Plotting Libraries
libraries('ggplot2', 'grid', 'ggfortify')
```

(1) Kuhn & Johnson 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:
>
> **(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.

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The outcome variable seems to be relatively normally distributed and a viable traget for a partial least squares regression.

(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).

Variable	Count	Variable	Count
ManufacturingProcess03	15	BiologicalMaterial01	0
ManufacturingProcess11	10	BiologicalMaterial02	0
ManufacturingProcess10	9	BiologicalMaterial03	0
ManufacturingProcess25	5	BiologicalMaterial04	0
ManufacturingProcess26	5	BiologicalMaterial05	0
ManufacturingProcess27	5	BiologicalMaterial06	0
ManufacturingProcess28	5	BiologicalMaterial07	0
ManufacturingProcess29	5	BiologicalMaterial08	0
ManufacturingProcess30	5	BiologicalMaterial09	0
ManufacturingProcess31	5	BiologicalMaterial10	0
ManufacturingProcess33	5	BiologicalMaterial11	0
ManufacturingProcess34	5	BiologicalMaterial12	0
ManufacturingProcess35	5	ManufacturingProcess09	0
ManufacturingProcess36	5	ManufacturingProcess13	0
ManufacturingProcess02	3	ManufacturingProcess15	0
ManufacturingProcess06	2	ManufacturingProcess16	0
ManufacturingProcess01	1	ManufacturingProcess17	0
ManufacturingProcess04	1	ManufacturingProcess18	0
ManufacturingProcess05	1	ManufacturingProcess19	0
ManufacturingProcess07	1	ManufacturingProcess20	0
ManufacturingProcess08	1	ManufacturingProcess21	0
ManufacturingProcess12	1	ManufacturingProcess32	0
ManufacturingProcess14	1	ManufacturingProcess37	0
ManufacturingProcess22	1	ManufacturingProcess38	0
ManufacturingProcess23	1	ManufacturingProcess39	0
ManufacturingProcess24	1	ManufacturingProcess42	0
ManufacturingProcess40	1	ManufacturingProcess43	0
ManufacturingProcess41	1	ManufacturingProcess44	0

No column had more than 15 missing values, so imputation seemed reasonable in all 28 columns with missing values. Using the `imputeknn()` method from the impute package, values were estimated for the 106 missing values. With these values imputed, there are now 56 complete cases of 176 observations.

Out of curiosity, we looked at the summary statistics before and after imputing (ignoring missing values before imputation) to see if it seriously impacts the quantiles in a few of the imputed variables.

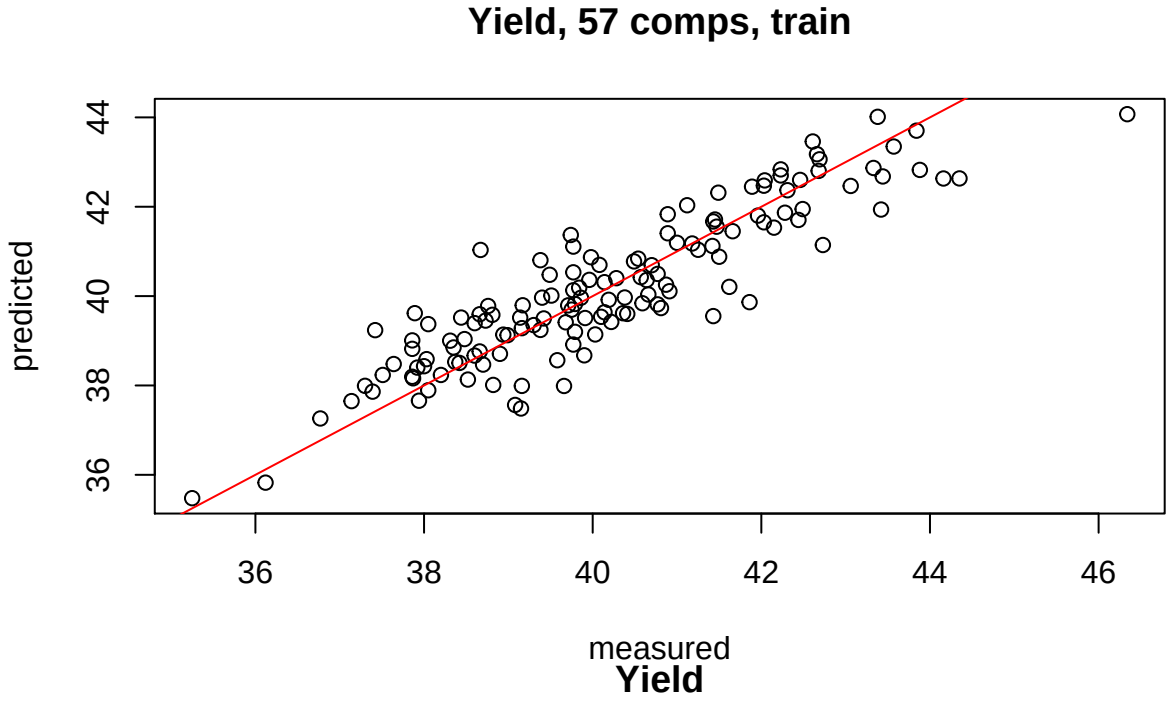
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
ManufacturingProcess01	175	11.207	1.822	0.000	10.800	12.150	14.100
ManufacturingProcess02	173	16.683	8.472	0.000	19.300	21.500	22.500
ManufacturingProcess03	161	1.540	0.022	1.470	1.530	1.550	1.600
ManufacturingProcess04	175	931.851	6.274	911.000	928.000	936.000	946.000
ManufacturingProcess05	175	1,001.693	30.527	923.000	986.750	1,008.850	1,175.300
ManufacturingProcess06	174	207.402	2.699	203.000	205.700	208.700	227.400
ManufacturingProcess07	175	177.480	0.501	177.000	177.000	178.000	178.000
ManufacturingProcess08	175	177.554	0.498	177.000	177.000	178.000	178.000
ManufacturingProcess09	176	45.660	1.546	39	44.9	46.5	49
ManufacturingProcess10	167	9.179	0.767	7.500	8.700	9.550	11.600
ManufacturingProcess11	166	9.386	0.716	7.500	9.000	9.900	11.500
ManufacturingProcess12	175	857.811	1,784.528	0.000	0.000	0.000	4,549.000
ManufacturingProcess13	176	34.508	1.015	32.100	33.900	35.200	38.600
ManufacturingProcess14	175	4,853.869	54.524	4,701.000	4,828.000	4,882.500	5,055.000
ManufacturingProcess15	176	6,038.920	58.313	5,904	6,010	6,061	6,233
ManufacturingProcess16	176	4,565.801	351.697	0	4,560.8	4,619	4,852
ManufacturingProcess17	176	34.344	1.248	31.300	33.500	35.100	40.000
ManufacturingProcess18	176	4,809.682	367.478	0	4,813	4,862	4,971
ManufacturingProcess19	176	6,028.199	45.579	5,890	6,000.8	6,050.2	6,146
ManufacturingProcess20	176	4,556.460	349.009	0	4,552.8	4,609.5	4,759
ManufacturingProcess21	176	−0.164	0.778	−2	−0.6	0	4
ManufacturingProcess22	175	5.406	3.331	0.000	3.000	8.000	12.000

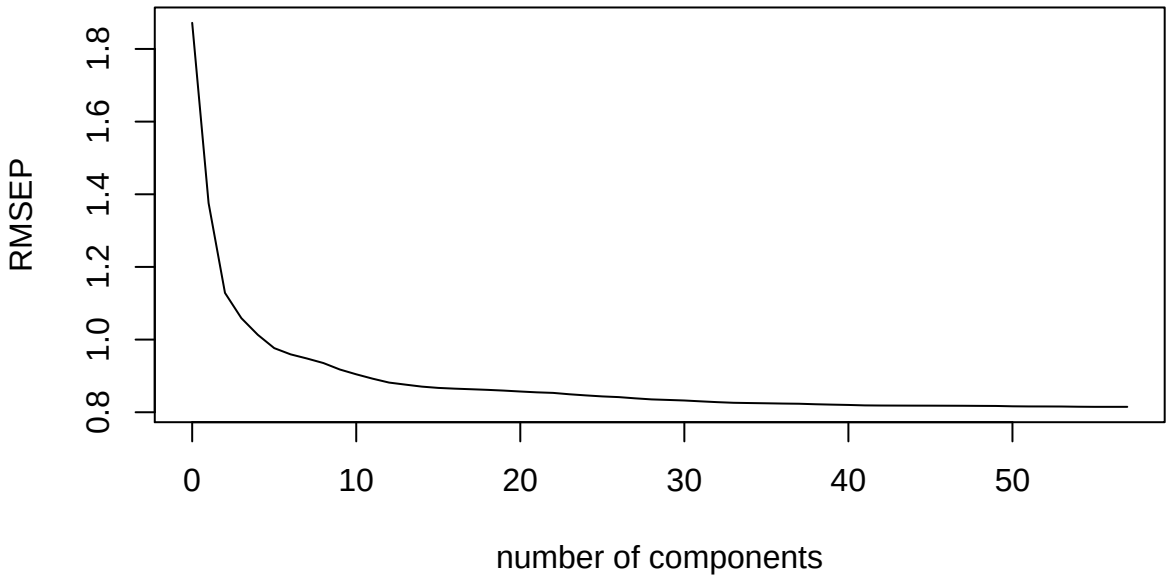
Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
ManufacturingProcess01	176	11.196	1.824	0.000	10.775	12.125	14.100
ManufacturingProcess02	176	16.664	8.446	0.000	19.172	21.500	22.500
ManufacturingProcess03	176	1.540	0.022	1.470	1.530	1.550	1.600
ManufacturingProcess04	176	931.828	6.264	911.000	927.950	936.000	946.000
ManufacturingProcess05	176	1,001.818	30.485	923.000	986.825	1,009.200	1,175.300
ManufacturingProcess06	176	207.393	2.688	203.000	205.700	208.700	227.400
ManufacturingProcess07	176	177.480	0.500	177.000	177.000	178.000	178.000
ManufacturingProcess08	176	177.552	0.498	177.000	177.000	178.000	178.000
ManufacturingProcess09	176	45.660	1.546	39	44.9	46.5	49
ManufacturingProcess10	176	9.186	0.751	7.500	8.700	9.525	11.600
ManufacturingProcess11	176	9.396	0.702	7.500	9.000	9.900	11.500
ManufacturingProcess12	176	852.938	1,780.597	0	0	0	4,549
ManufacturingProcess13	176	34.508	1.015	32.100	33.900	35.200	38.600
ManufacturingProcess14	176	4,853.500	54.587	4,701	4,827.2	4,882.2	5,055
ManufacturingProcess15	176	6,038.920	58.313	5,904	6,010	6,061	6,233
ManufacturingProcess16	176	4,565.801	351.697	0	4,560.8	4,619	4,852
ManufacturingProcess17	176	34.344	1.248	31.300	33.500	35.100	40.000
ManufacturingProcess18	176	4,809.682	367.478	0	4,813	4,862	4,971
ManufacturingProcess19	176	6,028.199	45.579	5,890	6,000.8	6,050.2	6,146
ManufacturingProcess20	176	4,556.460	349.009	0	4,552.8	4,609.5	4,759
ManufacturingProcess21	176	−0.164	0.778	−2	−0.6	0	4
ManufacturingProcess22	176	5.406	3.321	0.000	3.000	8.000	12.000

Looking at variables ManufacturingProcess11, with 10 missing values & ManufacturingProcess03 with 15 missing values in the original set and comparing them with the imputed sets, the quartiles are very nearly the same, so it seems that using `impute.knn( )` is a reasonable method.

(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?

After separating the data in training (80%) and testing sets (20%), using `sample.split( )` from the `caTools` package, we chose to create a partial least squares model using `kernelpls` from the `plsr` package, after centering and scaling. The intent with Partial Least Squares was to maximize covariance between the variables and the outcome variable.





The following are the root mean squared areas for models built with increasing numbers of principal components. Based on these errors, the 55 components seems to be the best possible model, however, at 41 components, the model's improvement becomes infinitesimal. So that is what we will build the final model upon.

```
(Intercept)      1 comps      2 comps      3 comps      4 comps
     1.8719       1.3758       1.1285       1.0589       1.0132
    5 comps      6 comps      7 comps      8 comps      9 comps
     0.9765       0.9593       0.9479       0.9354       0.9176
   10 comps     11 comps     12 comps     13 comps     14 comps
     0.9044       0.8924       0.8817       0.8760       0.8705
   15 comps     16 comps     17 comps     18 comps     19 comps
     0.8670       0.8649       0.8633       0.8616       0.8596
   20 comps     21 comps     22 comps     23 comps     24 comps
     0.8571       0.8548       0.8533       0.8494       0.8464
   25 comps     26 comps     27 comps     28 comps     29 comps
     0.8436       0.8417       0.8383       0.8353       0.8339
   30 comps     31 comps     32 comps     33 comps     34 comps
     0.8324       0.8300       0.8278       0.8260       0.8253
   35 comps     36 comps     37 comps     38 comps     39 comps
     0.8246       0.8239       0.8234       0.8220       0.8210
   40 comps     41 comps     42 comps     43 comps     44 comps
     0.8201       0.8189       0.8185       0.8183       0.8181
   45 comps     46 comps     47 comps     48 comps     49 comps
     0.8181       0.8180       0.8178       0.8175       0.8173
   50 comps     51 comps     52 comps     53 comps     54 comps
     0.8163       0.8160       0.8159       0.8158       0.8151
   55 comps     56 comps     57 comps
     0.8149       0.8149       0.8149
```

The final model is Partial Least Squares with 41 centered and scaled, principal components using the `kernelpls`

** Evaluating the Training Set**

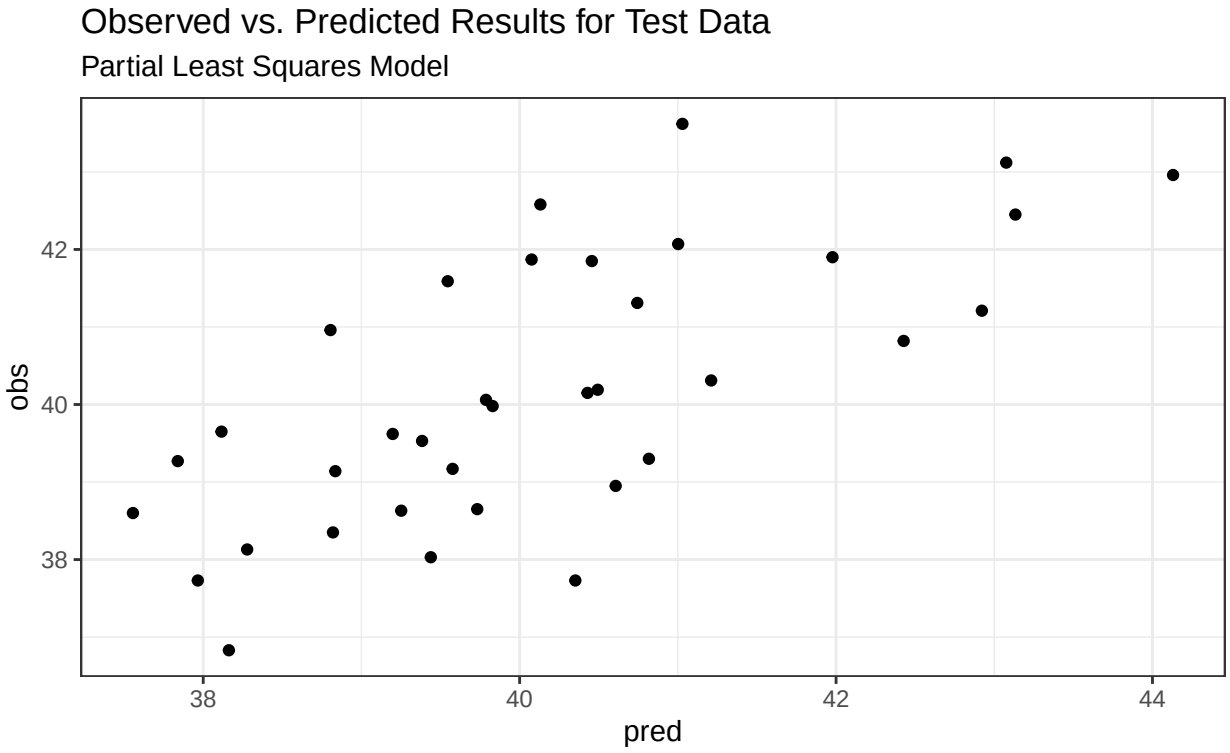
```
     RMSE  Rsquared       MAE
1.3757981 0.4598005 1.1110483
```

Clearly this is not a fabulous model, as the r-squared suggests that only 45% of variation in out outcome variable, Yield, is explained by this model, and this is just the training set.

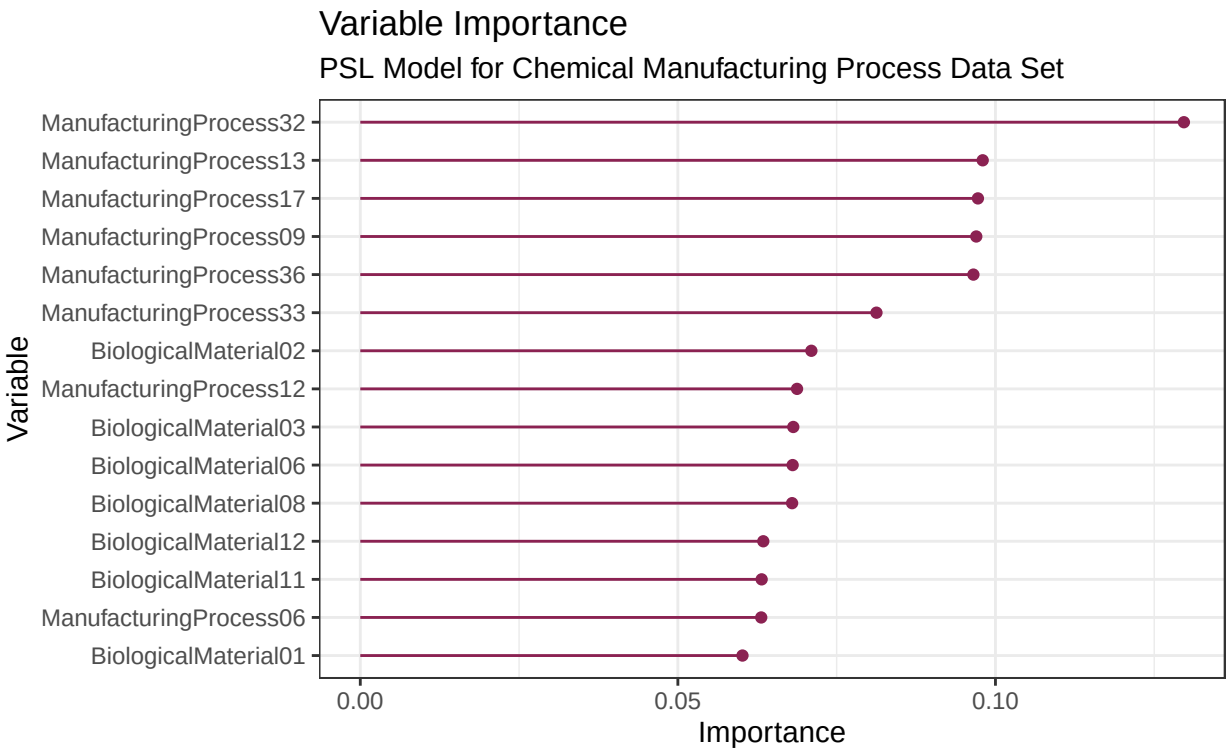
(d). Predict the response for the test set. What is the value of the performance metric and how does this compare with the resampled performance metric on the training set?

Having made predictions on the `test` data with an RMSE of 1.29 which compared the to the training RMSE of 1.37, unexpectedly, this test set outperforms the training set. However a closer looc at the

```
     RMSE  Rsquared       MAE
1.2918866 0.4853202 1.0457651
```

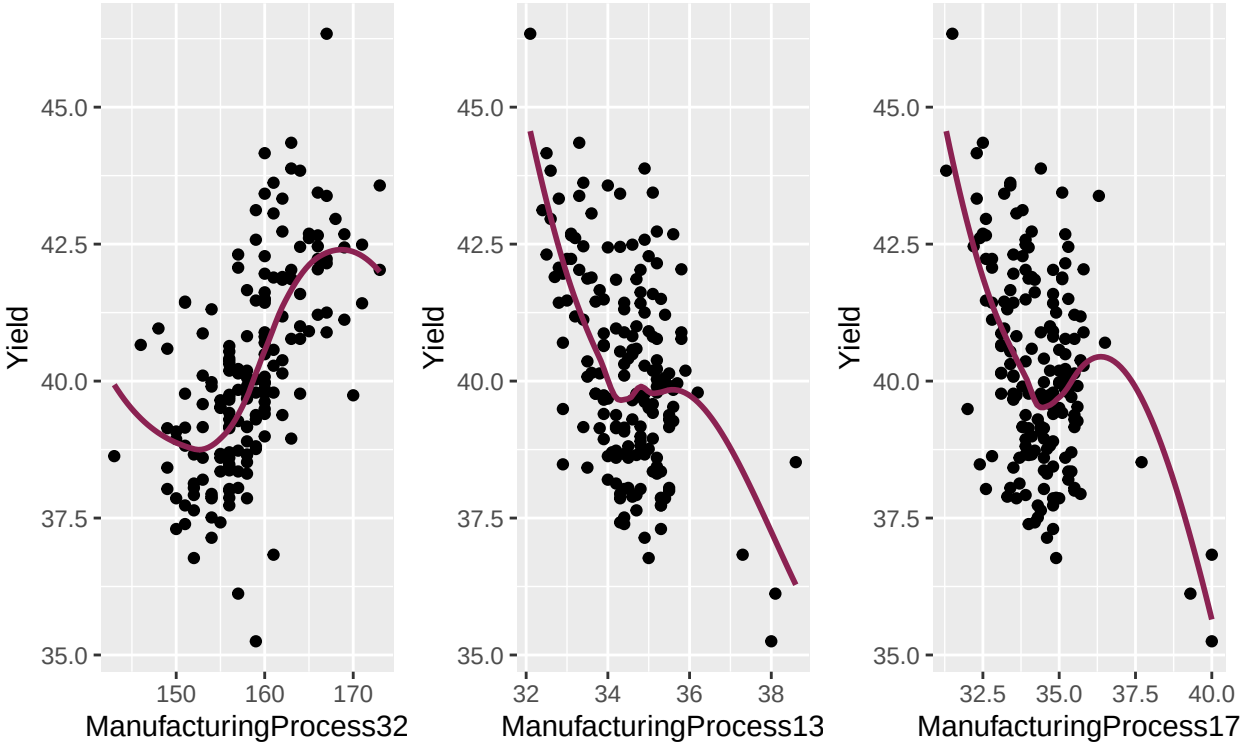


(e). Which predictors are most important in the model you have trained? Do either the biological or process predictors dominate the list?



Based on the top seven predictors are all ManufacturingProcesses 32, 13, 17, 09, 36 & 33. Four out of the next six variables are Biological Materials variables.

(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?



In looking at the plots of the top three variables, ManufacturingProcesses 32 is somewhat positively associated with Yield, while ManufacturingProcesses 13 & ManufacturingProcesses 17 are slightly negatively associated with Yield. Given that these are th three most associated variables and they are only slightly correlated with Yield, is makes sense that the model is only modestly predictive and requires the majority of variables even with

Appendix

```
require(AppliedPredictiveModeling)
require(tidyverse)
require(impute)
require(caTools)
require(pls)
require(kableExtra)
require(ggplot2)
require(stargazer)
require(caret)
require(tidyverse)
require(gridExtra)
options(scipen = 999)
# a.
data("ChemicalManufacturingProcess")

# Total NA Values
#na_table<- table(is.na(ChemicalManufacturingProcess))
total_na<-sapply(ChemicalManufacturingProcess[2:57], function(x) sum(is.na(x)))
```

```r
na_table<-sapply(ChemicalManufacturingProcess, function(x) table(is.na(x)))

total_na<- data.frame(sort(total_na, decreasing = TRUE))
total_na<- cbind(Variable = rownames(total_na), total_na)
rownames(total_na) <- 1:nrow(total_na)
colnames(total_na)<-  c("Variable", "Count")
total_na<-cbind(total_na[1:28,],total_na[29:56,])

ggplot(ChemicalManufacturingProcess, aes(x = Yield))+
  geom_histogram(colour ='black', fill = 'violetred4')

# b. Imputing Values
summary(ChemicalManufacturingProcess)
imputed_data = data.frame(impute.knn(as.matrix(ChemicalManufacturingProcess),
       k =10,
       rowmax =.30,
       colmax =.85,
       rng.seed =1942)$data)



imp_proc_34 <- summary(imputed_data$ManufacturingProcess34)
proc_34 <- summary(ChemicalManufacturingProcess$ManufacturingProcess34)
imp_proc_03 <- summary(imputed_data$ManufacturingProcess03)
proc_03 <- summary(ChemicalManufacturingProcess$ManufacturingProcess03)

# c. tts, train and evaluate
set.seed(1492)   #  set seed to ensure you always have same random numbers generated
sample = sample.split(imputed_data, SplitRatio = 0.80) # splits the data in the ratio mentioned in Spli
train =subset(imputed_data,sample ==TRUE) # creates a training dataset named train1 with rows which are
test=subset(imputed_data, sample==FALSE)


#
fit <- plsr(Yield~., data=train,
            method = 'kernelpls',
            scale = TRUE,
            center = TRUE)



# 57 is the best lowest number of components

# Best Train

fit_41 <- plsr(Yield~., data=train,
            method = 'kernelpls',
            scale = TRUE,
            center = TRUE,
            ncomp =41)

#  Train Metrics
train_eval=data.frame('obs' = train$Yield, 'pred' =fit$fitted.values)
```

```r
colnames(train_eval) <- c('obs', 'pred')
caret::defaultSummary(train_eval)
#     RMSE  Rsquared       MAE
# 1.3757981 0.4598005 1.1110483
#
# # d.
#
# #Test Predictions & Metrics
test_pred_41 <- predict(fit_41, test, ncomp=41)
test_eval_41=data.frame('obs' = test$Yield, 'pred' =test_pred_41)
colnames(test_eval_41) <- c('obs', 'pred')
caret::defaultSummary(test_eval_41)


eval_plot <- ggplot(test_eval_41, aes(obs, pred)) +
  labs(title="Observed vs. Predicted Results for Test Data",
       subtitle="Partial Least Squares Model")+
  geom_point()+
  coord_flip()+
  theme_bw()+
  theme()
# e Importance

importance <- caret::varImp(fit_41, scale=FALSE)

importance<-importance%>%
    mutate(Variable = row.names(importance))%>%
    remove_rownames()%>%
    select(Variable, Overall)%>%
    arrange(desc(Overall))

imp_plot <- ggplot(head(importance, 15), aes(x=reorder(Variable, Overall), y=Overall)) +
    geom_point(colour = 'violetred4') +
    geom_segment(aes(x=Variable,xend=Variable,y=0,yend=Overall),colour = 'violetred4') +
    labs(title="Variable Importance",
         subtitle="PSL Model for Chemical Manufacturing Process Data Set", x="Variable", y="Importance"
    coord_flip()+
    theme_bw()+
    theme()


# F Comparison

p1 <-qplot(ManufacturingProcess32,Yield,  data =ChemicalManufacturingProcess)+
  geom_smooth(method = "loess", se =FALSE)
p2 <-qplot(ManufacturingProcess13,Yield,   data =ChemicalManufacturingProcess)+
  geom_smooth(method = "loess", se =FALSE)
p3 <-qplot( ManufacturingProcess17, Yield, data =ChemicalManufacturingProcess)+
  geom_smooth(method = "loess", se =FALSE)
```