Lab Report 2

By: Abhishek Shah, Ravi Seth, Tanya Ralliaram

```r
# Insert necessary packages
library('tidyverse')
library('gridExtra')
library('ISLR')
library('plotly')
library('caret')
library('MASS')
library('glmnet')
library('gam')
library('splines')
library('foreach')
library("leaps")
library("AmesHousing")
```

Question 1: Nonlinear Regression

1.1. Process your data

```r
# Read the data
diamonds <- read.csv('data/diamonds.csv')

# Remove all rows that contain NA
diamonds <- na.omit(diamonds)

# Downsample data
diamonds <- diamonds[sample(5000), ]

# Select columns
diamonds <- diamonds[, c('X', 'carat', 'cut', 'depth', 'table', 'price', 'x', 'y', 'z')]

# Convert string column to categorical column
# Convert diamond string variables to categorical
diamonds$cut <- as.factor(diamonds$cut)

dim(diamonds)
```

```
## [1] 5000    9
```

```r
head(diamonds)
```

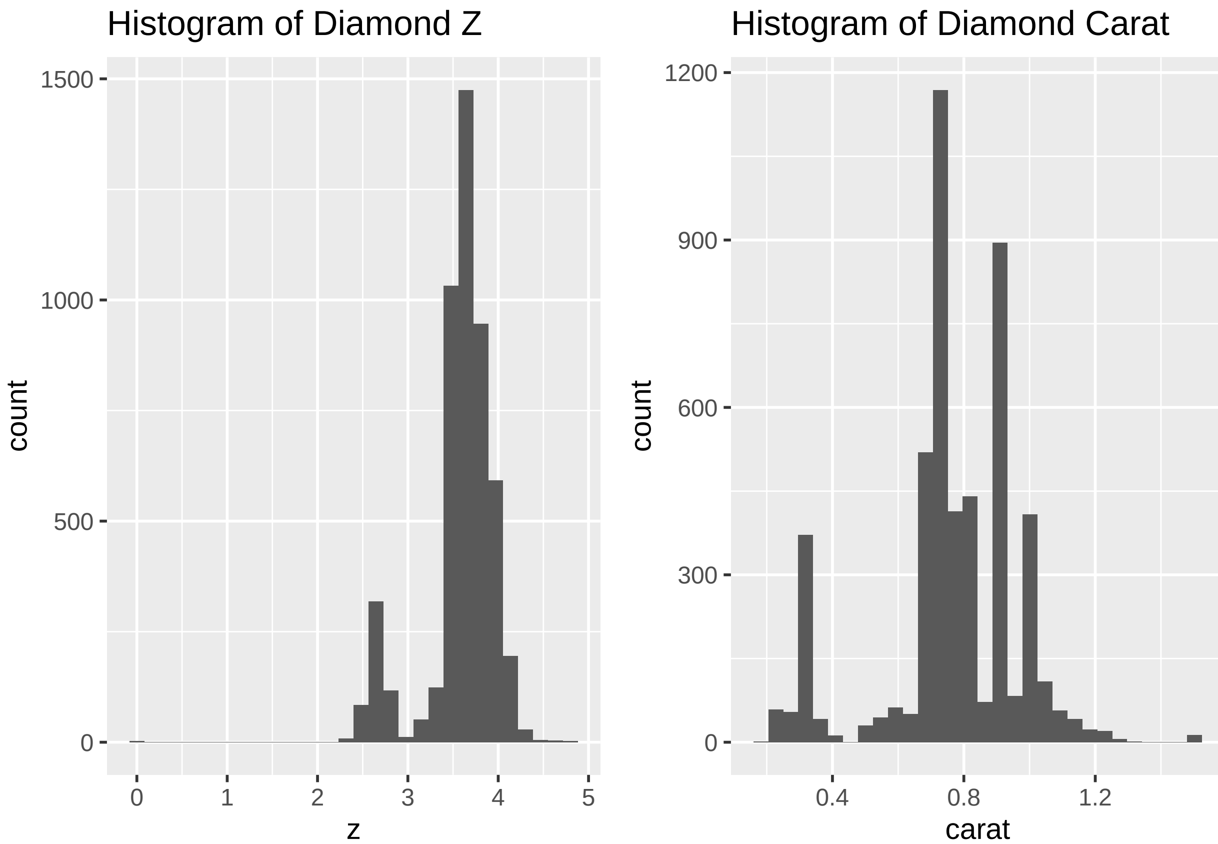
```
##         X carat     cut depth table price    x    y    z
## 1448 1448  0.72 Premium  62.7    58  2900 5.68 5.65 3.55
## 444   444  1.03    Good  63.8    54  3607 6.36 6.30 4.04
## 3698 3698  0.70   Ideal  60.5    56  3419 5.78 5.83 3.51
## 3354 3354  0.71 Premium  62.2    58  2832 5.72 5.66 3.54
## 4641 4641  1.00 Premium  62.4    58  3528 6.39 6.27 3.95
## 4856 4856  0.71   Ideal  61.2    56  2909 5.76 5.81 3.54
```

1.3. Visualize the data

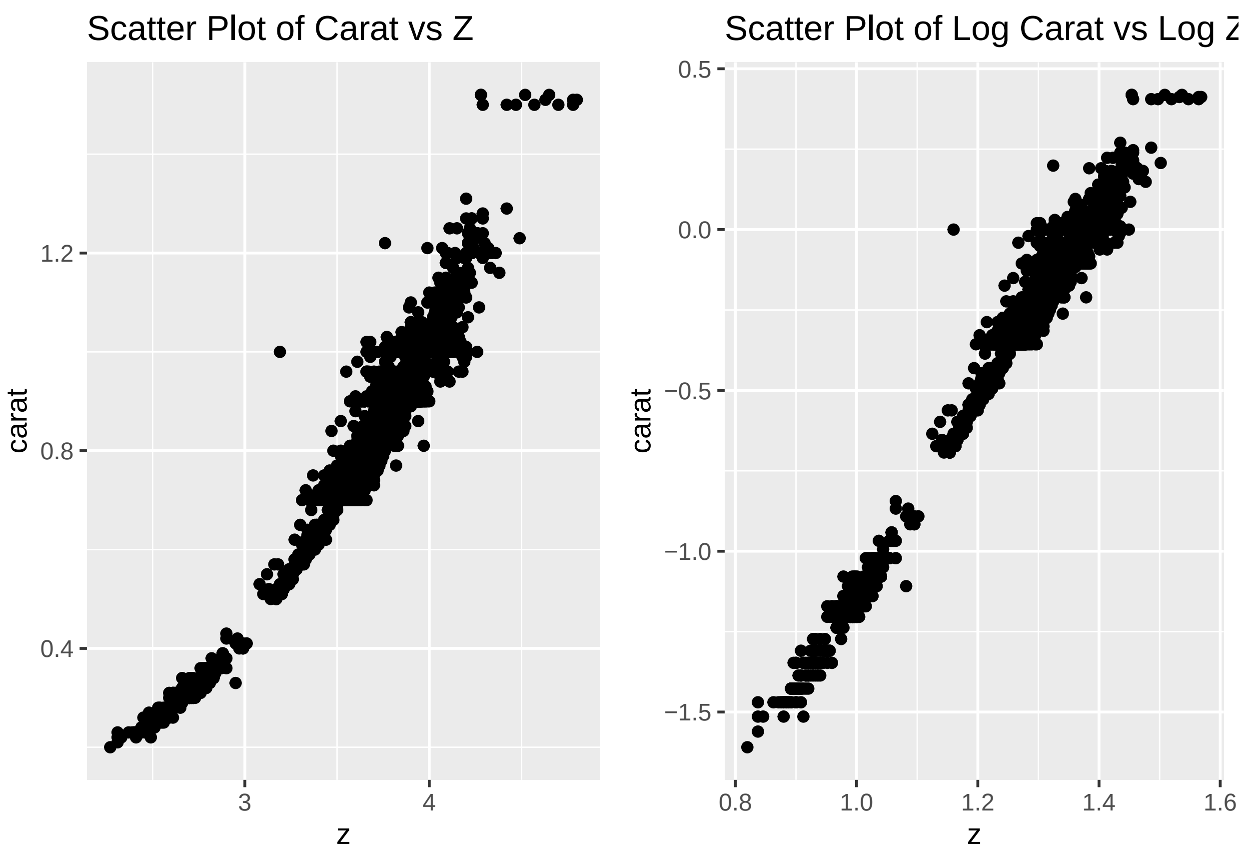
It was decided to do step 1.3. before step 1.2 because before splitting the input data into train and test, it makes more sense to apply the necessary transformations first.

```r
histo1 <- ggplot(data = diamonds) +
  geom_histogram(aes(x = z)) +
  ggtitle("Histogram of Diamond Z")

histo2 <- ggplot(data = diamonds) +
  geom_histogram(aes(x = carat)) +
  ggtitle("Histogram of Diamond Carat")

grid.arrange(histo1, histo2, ncol=2)
```

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




```r
# Remove outliers
diamonds <- diamonds[(diamonds$z>1),]

# Apply logarithmic transform
numeric.cols <- summarize_all(diamonds, is.numeric) %>% unlist()
logDiamonds <- diamonds
logDiamonds[,numeric.cols] <- log(diamonds[,numeric.cols])

scatter1 <- ggplot(data = diamonds) +
  geom_point(aes(x = z, y=carat)) +
  ggtitle("Scatter Plot of Carat vs Z")

scatter2 <- ggplot(data = logDiamonds) +
  geom_point(aes(x = z, y=carat)) +
  ggtitle("Scatter Plot of Log Carat vs Log Z")

grid.arrange(scatter1, scatter2, ncol=2)
```



After removing some outliers, it can be clearly seen that as z is increased, the carat is also increased. In terms of linearity, it is slightly non-linear and showing exponential form. When applying log transform to all the carat and x values, then the relationship is clearly linear.

1.2. Train / Test Split

```
set.seed(156)
train_inds <- sample(1:nrow(logDiamonds), floor(nrow(logDiamonds)*0.8))
train <- logDiamonds[ train_inds, ]
test  <- logDiamonds[-train_inds, ]

cat('train: ', nrow(train), ', test: ', nrow(test))
```

```
## train:  3997 , test:  1000
```

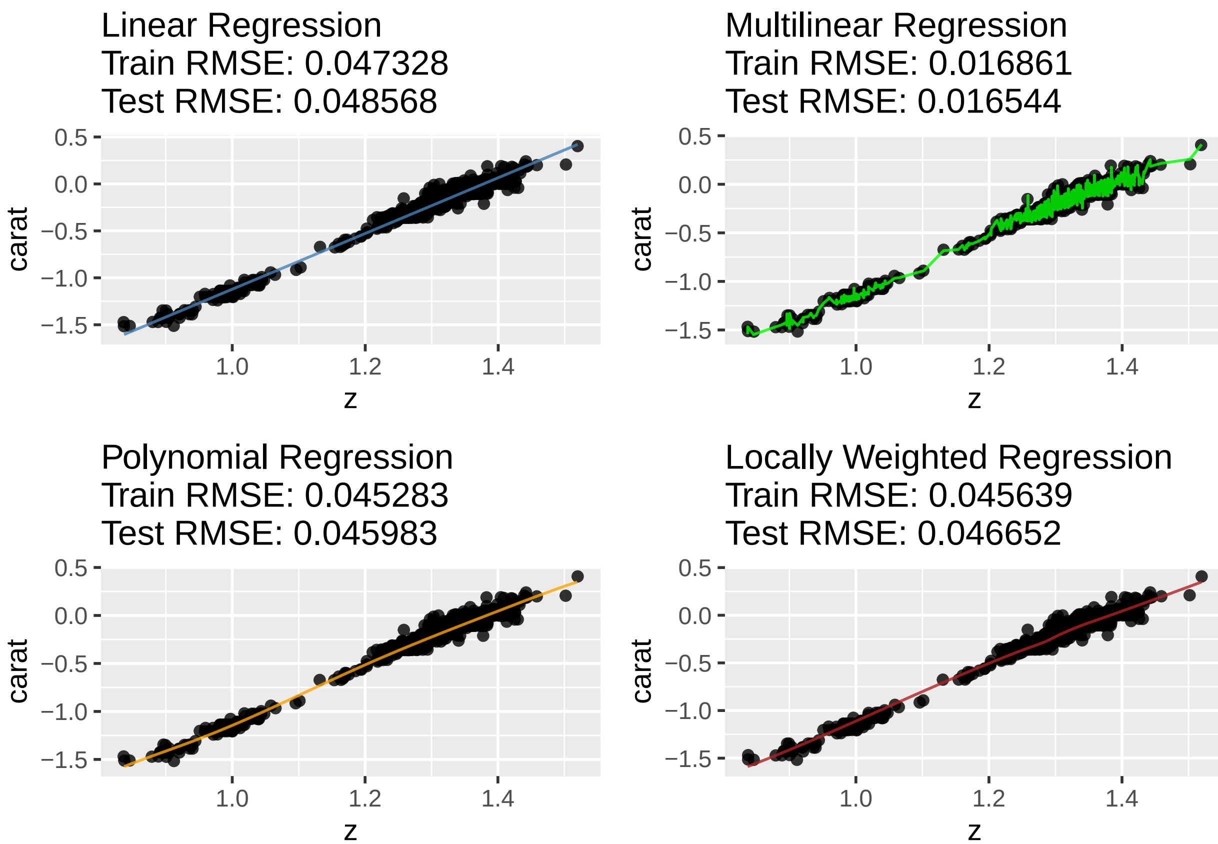
1.4. Fit 4 Models

```
# Linear Regression
fit.lm  <- lm(carat ~ z, data = train)

# Predictions
preds.lm_train <- predict(fit.lm, train)
preds.lm_test <- predict(fit.lm, test)
```

```
# RMSE
rmse.lm_train  <- RMSE(preds.lm_train, train$carat)
rmse.lm_test  <- RMSE(preds.lm_test, test$carat)

# Multilinear Regression
fit.mlm <- lm(carat ~ ., data = train)

# Predictions
preds.mlm_train <- predict(fit.mlm, train)
preds.mlm_test <- predict(fit.mlm, test)

# RMSE
rmse.mlm_train  <- RMSE(preds.mlm_train, train$carat)
rmse.mlm_test  <- RMSE(preds.mlm_test, test$carat)

# Polynomial Regression
fit.poly <- lm(carat ~ poly(z,6), data = train)

# Predictions
preds.poly_train <- predict(fit.poly, train)
preds.poly_test <- predict(fit.poly, test)

# RMSE
rmse.poly_train  <- RMSE(preds.poly_train, train$carat)
rmse.poly_test  <- RMSE(preds.poly_test, test$carat)

# Locally Weighted Regression
fit.wlm <- loess(carat ~ z, data=train)

# Predictions
preds.wlm_train <- predict(fit.wlm, train)
preds.wlm_test <- predict(fit.wlm, test)

# RMSE
rmse.wlm_train  <- RMSE(preds.wlm_train, train$carat)
rmse.wlm_test  <- RMSE(preds.wlm_test, test$carat)
```

```
# Plots
linear_plot <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.lm_test), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Linear Regression \nTrain RMSE: ',round(rmse.lm_train, 6), "\nTest RMSE: ", rour

multilinear_plot <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.mlm_test), colour = 'green', alpha=.8) +
  ggtitle(paste0('Multilinear Regression \nTrain RMSE: ',round(rmse.mlm_train, 6), "\nTest RMSE: '

poly_plot <- ggplot(test, aes(y=carat, x=z)) +
```

```
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.poly_test), colour = 'orange', alpha=.8) +
  ggtitle(paste0('Polynomial Regression \nTrain RMSE: ',round(rmse.poly_train, 6), "\nTest RMSE: '

weighted_plot <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.wlm_test), colour = 'firebrick', alpha=.8) +
  ggtitle(paste0('Locally Weighted Regression \nTrain RMSE: ',round(rmse.wlm_train, 6), "\nTest RM

grid.arrange(linear_plot, multilinear_plot, poly_plot, weighted_plot, ncol=2)
```



Based off of Train RMSE, the order of models from best (lowest RMSE) to worst (highest RMSE) is: 1. Multilinear Regression 2. Polynomial Regression 3. Locally Weighted Regression 4. Linear Regression

Based off of the Test RMSE, the order of models from best (lowest RMSE) to worst (highest RMSE) is: 1. Multilinear Regression 2. Polynomial Regression 3. Locally Weighted Regression 4. Linear Regression

The order of models from best to worst did not change when ordering by train RMSE or test RMSE.

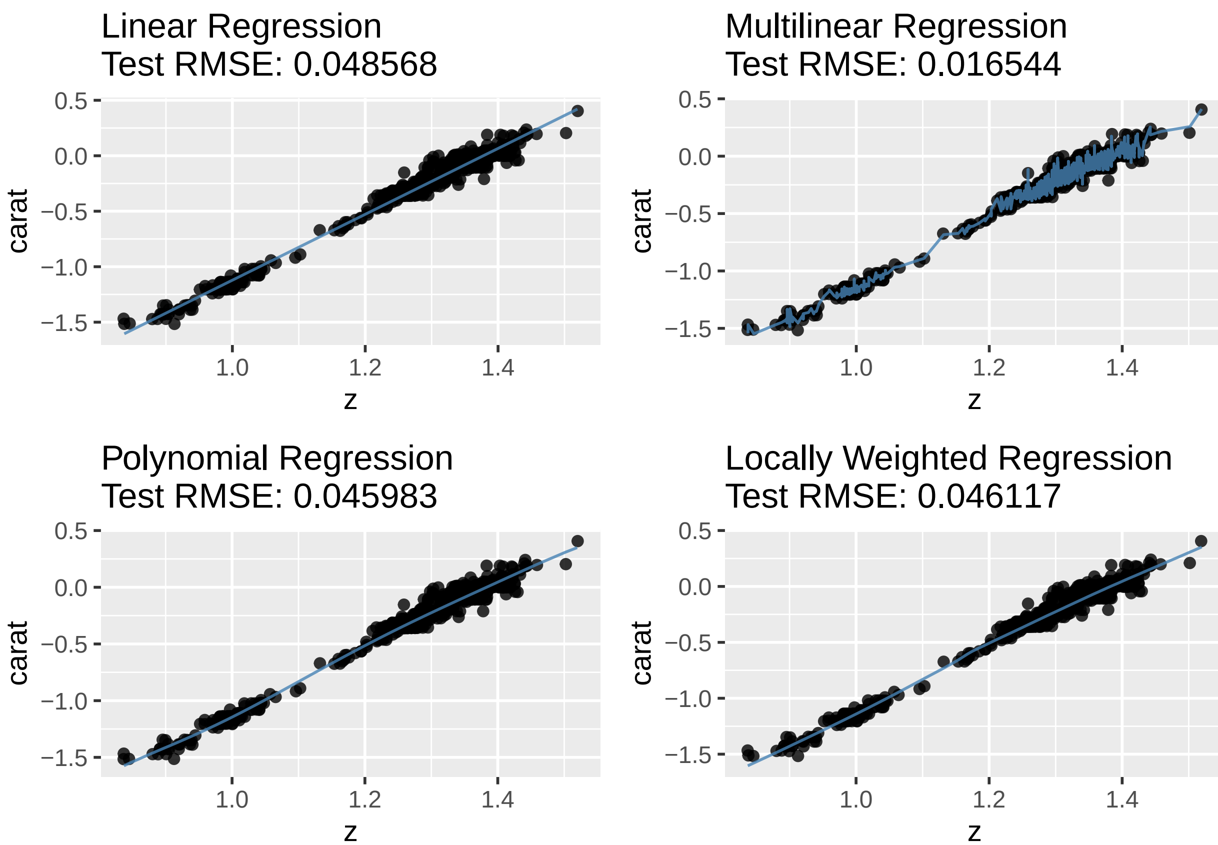
1.5. Cross Validation

```
ctrl <- trainControl(method = "repeatedcv", number = 10, repeats=5)

# Linear regression
cv_fit.lm <- train(
```

```r
  form = carat ~ z,
  data = train,
  method = "lm",
  trControl = ctrl
)

preds.cv_lm <- predict(cv_fit.lm,test)
rmse.cv_lm  <- RMSE(preds.cv_lm,test$carat)

# Multilinear regression
cv_fit.mlm <- train(
  form = carat ~ .,
  data = train,
  method = "lm",
  trControl = ctrl
)

preds.cv_mlm <- predict(cv_fit.mlm,test)
rmse.cv_mlm  <- RMSE(preds.cv_mlm,test$carat)

# Polynomial regression

cv_fit.poly <- train(
  form = carat ~ poly(z, 6),
  data = train,
  method = "lm",
  trControl = ctrl
)

preds.cv_poly <- predict(cv_fit.poly,test)
rmse.cv_poly  <- RMSE(preds.cv_poly,test$carat)

# Locally weighted regression
cv_fit.wlm <- train(
  form = carat ~ z,
  data = train,
  method = "gamLoess",
  trControl = ctrl
)

preds.cv_wlm <- predict(cv_fit.wlm,test)
rmse.cv_wlm  <- RMSE(preds.cv_wlm,test$carat)
```

```r
cv_linear_plot <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.cv_lm), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Linear Regression \nTest RMSE: ',round(rmse.cv_lm, 6)))

cv_multilinear_plot <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
```

```
  geom_line(aes(y=preds.cv_mlm), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Multilinear Regression \nTest RMSE: ',round(rmse.cv_mlm, 6)))

cv_poly_plot <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.cv_poly), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Polynomial Regression \nTest RMSE: ',round(rmse.cv_poly, 6)))

cv_wlm_plot <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.cv_wlm), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Locally Weighted Regression \nTest RMSE: ',round(rmse.cv_wlm, 6)))

grid.arrange(cv_linear_plot, cv_multilinear_plot, cv_poly_plot, cv_wlm_plot, ncol=2)
```



Based off of the Test RMSE, the order of models from best (lowest RMSE) to worst (highest RMSE) is: 1. Multilinear Regression 2. Locally Weighted Regression 3. Polynomial Regression 4. Linear Regression

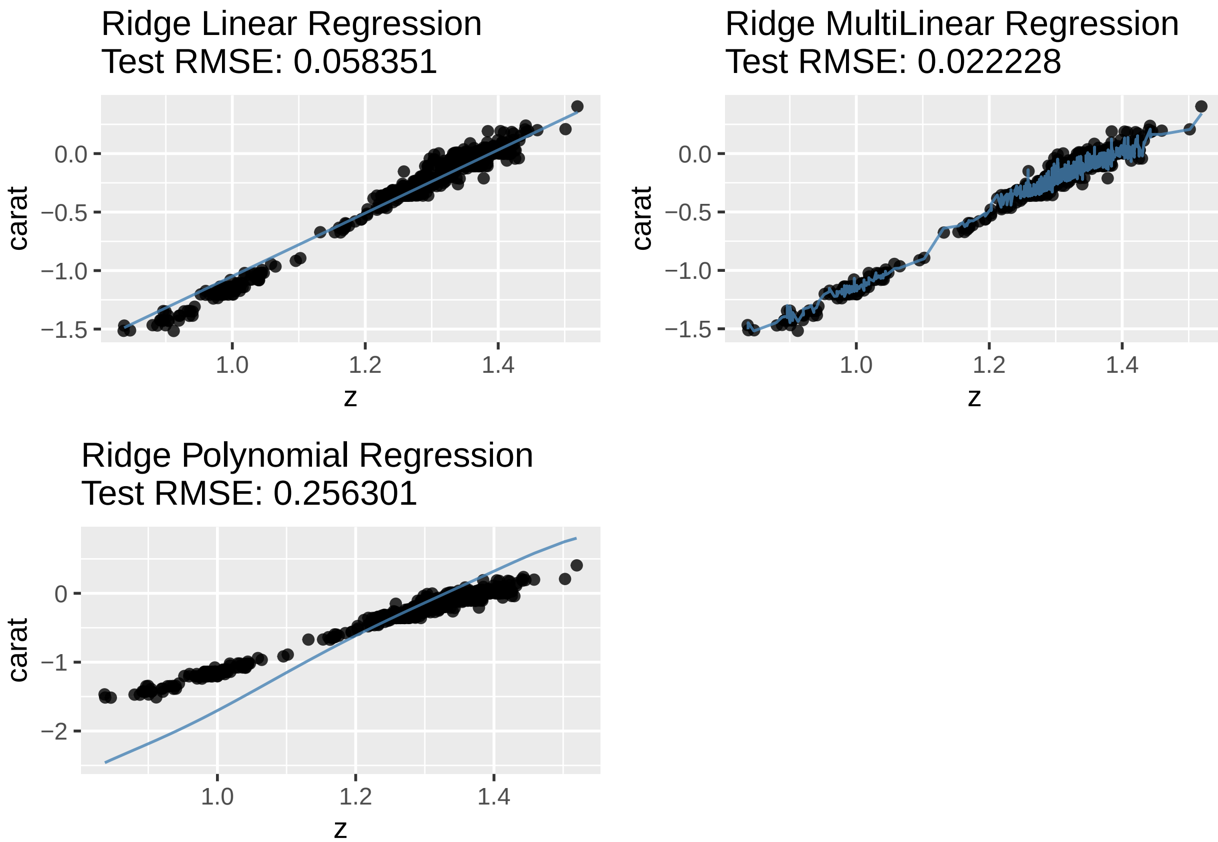
The order of models have changed when adding k fold cross validation. The Locally Weighted Regression is now second best and moved in front of Polynomial Regression. Therefore, by applying cross validation, more precise performance metrics have been measured for the models.

1.6. Shrinkage

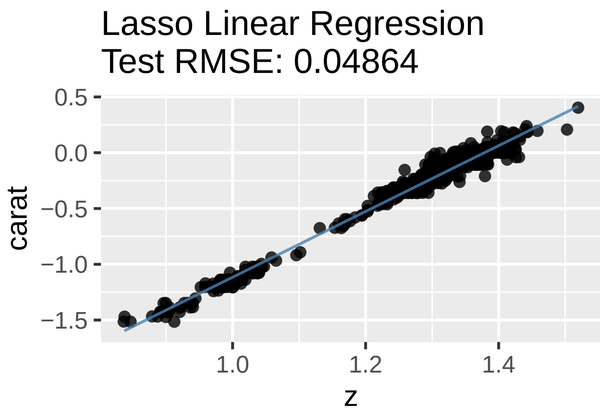
```r
x_train <- model.matrix(carat ~ z,train)
x_train_multi <- model.matrix(carat ~ .,train)
x_train_poly <- model.matrix(carat ~ poly(z,6),train)
y_train <- train$carat

x_test <- model.matrix(carat ~ z,test)
x_test_multi <- model.matrix(carat ~ .,test)
x_test_poly <- model.matrix(carat ~ poly(z, 6),test)

# Ridge
fit.ridge_lm  <- cv.glmnet(x_train,y_train,alpha=0, nfolds = 10)
fit.ridge_lm <- glmnet(x_train,y_train,alpha=0, lambda=fit.ridge_lm$lambda.min)

fit.ridge_mlm  <- cv.glmnet(x_train_multi,y_train,alpha=0, nfolds = 10)
fit.ridge_mlm <- glmnet(x_train_multi,y_train,alpha=0, lambda=fit.ridge_mlm$lambda.min)

fit.ridge_poly  <- cv.glmnet(x_train_poly,y_train,alpha=0, nfolds = 10)
fit.ridge_poly <- glmnet(x_train_poly,y_train,alpha=0, lambda=fit.ridge_poly$lambda.min)


preds.ridge_lm <- predict(fit.ridge_lm, x_test)
preds.ridge_mlm <- predict(fit.ridge_mlm, x_test_multi)
preds.ridge_poly <- predict(fit.ridge_poly, x_test_poly)

rmse.ridge_lm <- RMSE(preds.ridge_lm, test$carat)
rmse.ridge_mlm <- RMSE(preds.ridge_mlm, test$carat)
rmse.ridge_poly <- RMSE(preds.ridge_poly, test$carat)

# Lasso
fit.lasso_lm  <- cv.glmnet(x_train,y_train,alpha=1, nfolds = 10)
fit.lasso_lm <- glmnet(x_train,y_train,alpha=1, lambda=fit.lasso_lm$lambda.min)

fit.lasso_mlm  <- cv.glmnet(x_train_multi,y_train,alpha=1, nfolds = 10)
fit.lasso_mlm <- glmnet(x_train_multi,y_train,alpha=1, lambda=fit.lasso_mlm$lambda.min)

fit.lasso_poly  <- cv.glmnet(x_train_poly,y_train,alpha=1, nfolds = 10)
fit.lasso_poly <- glmnet(x_train_poly,y_train,alpha=1, lambda=fit.lasso_poly$lambda.min)

preds.lasso_lm <- predict(fit.lasso_lm, x_test)
preds.lasso_mlm <- predict(fit.lasso_mlm, x_test_multi)
preds.lasso_poly <- predict(fit.lasso_poly, x_test_poly)

rmse.lasso_lm <- RMSE(preds.lasso_lm, test$carat)
rmse.lasso_mlm <- RMSE(preds.lasso_mlm, test$carat)
rmse.lasso_poly <- RMSE(preds.lasso_poly, test$carat)
```

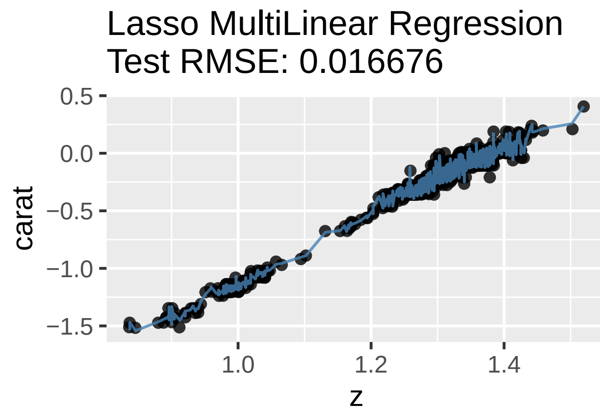
```r
# Ridge
ridge_lm <-ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
```

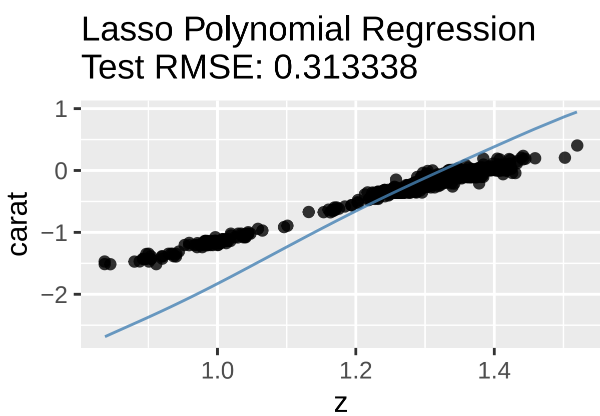
```r
  geom_line(aes(y=preds.ridge_lm), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Ridge Linear Regression \nTest RMSE: ',round(rmse.ridge_lm, 6)))

ridge_mlm <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.ridge_mlm), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Ridge MultiLinear Regression \nTest RMSE: ',round(rmse.ridge_mlm, 6)))

ridge_poly <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.ridge_poly), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Ridge Polynomial Regression \nTest RMSE: ',round(rmse.ridge_poly, 6)))

# Lasso
lasso_lm <-ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.lasso_lm), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Lasso Linear Regression \nTest RMSE: ',round(rmse.lasso_lm, 6)))

lasso_mlm <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.lasso_mlm), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Lasso MultiLinear Regression \nTest RMSE: ',round(rmse.lasso_mlm, 6)))

lasso_poly <- ggplot(test, aes(y=carat, x=z)) +
  geom_point(alpha=.8, position = position_jitter()) +
  geom_line(aes(y=preds.lasso_poly), colour = 'steelblue', alpha=.8) +
  ggtitle(paste0('Lasso Polynomial Regression \nTest RMSE: ',round(rmse.lasso_poly, 6)))

grid.arrange(ridge_lm, ridge_mlm, ridge_poly, ncol=2)
```




```
grid.arrange(lasso_lm, lasso_mlm, lasso_poly, ncol=2)
```







The model that yielded the lowest RMSE was Lasso Multilinear Regression. Based off of the Test RMSE, the order of models from best (lowest RMSE) to worst (highest RMSE) is: 1. Lasso Multilinear Regression 2. Ridge Multilinear Regression 3. Lasso Linear Regression 4. Ridge Linear Regression 5. Ridge Polynomial Regression 6. Lasso Polynomial Regression

Question 2

```
# read in data
health <- read.csv("data/mental_health.csv")[,-1]
```

2.1 Train / Test Split

```
set.seed(123)
train_inds <- sample(1:nrow(health), floor(nrow(health)*0.8))
train <- health[ train_inds, ]
test  <- health[-train_inds, ]
X_train   <- model.matrix(IsMentalHealthRelated ~ .,train)
y_train   <- train$IsMentalHealthRelated
X_test   <- model.matrix(IsMentalHealthRelated ~ .,test)
y_test   <- test$IsMentalHealthRelated
cat('train: ', dim(train), ', test: ', dim(test))
```

```
## train:  5049 489 , test:  1263 489
```

2.2 Fit models

```
# Logistic Regression model
fit.logreg <- glm(formula = IsMentalHealthRelated ~ ., data=train, family = binomial())
# L1 Model
cv.fit  <- cv.glmnet(X_train, y_train, alpha=1, family="binomial", nfolds = 5)
lambda.l1 <- cv.fit$lambda.min
fit.l1 <- glmnet(X_train, y_train, alpha=1, family="binomial", lambda=lambda.l1)
# L2 Model
cv.fit  <- cv.glmnet(X_train, y_train, alpha=0, family="binomial", nfolds = 5)
lambda.l2 = cv.fit$lambda.min
fit.l2 <- glmnet(X_train, y_train, alpha=0, family="binomial", lambda=lambda.l2)
```

2.3 Compare Performances

```
# Logistic Regression (LR)
probs.logreg <- predict(fit.logreg, as.data.frame(X_test), type="response")
preds.logreg  <- ifelse(probs.logreg >= 0.5, 1, 0)
acc.logreg <- mean(preds.logreg == y_test)
# L1 Model
probs.l1 <- predict(fit.l1, X_test, type="response")
preds.l1  <- ifelse(probs.l1 >= 0.5, 1, 0)
acc.l1 <- mean(preds.l1 == y_test)
# L2 Model
probs.l2 <- predict(fit.l2, X_test, type="response")
```

```r
preds.l2  <- ifelse(probs.l2 >= 0.5, 1, 0)
acc.l2 <- mean(preds.l2 == y_test)
cat(sprintf("Logisitc Regression Accuracy: %f \nL1 Accuracy: %f \nL2 Accuracy: %f", acc.logreg, ac
```

```
## Logisitc Regression Accuracy: 0.855107
## L1 Accuracy: 0.866983
## L2 Accuracy: 0.869359
```

The L2 model had the best accuracy and L1 has the second best accuracy. Logistic Regression without any regularization had the worst accuracy out of the three.

2.4 Interpret the models

```r
sorted.l1 <- sort(coef(fit.l1)[,1])
cat('The words that have the highest coefficients with L1 are: \n')
```

```
## The words that have the highest coefficients with L1 are:
```

```r
sort(tail(sorted.l1, 5), decreasing=TRUE)
```

```
##          term       counsel mental.health            op    university
##      7.943078      6.930450      4.722108      4.580570      3.966913
```

```r
cat('\nThe words that have the smallest coefficients with L1 are: \n')
```

```
##
## The words that have the smallest coefficients with L1 are:
```

```r
head(sorted.l1, 5)
```

```
##    fitness    workout     muscle      squat   workouts
## -11.431782 -10.222683  -9.380444  -7.980440  -7.517964
```

```r
sorted.l2 <- sort(coef(fit.l2)[,1])
cat('\nThe words that have the highest coefficients with L2 are: \n')
```

```
##
## The words that have the highest coefficients with L2 are:
```

```r
sort(tail(sorted.l2, 5), decreasing=TRUE)
```

```
##       term    counsel university         op    service
##   4.485622   3.948127   3.516435   2.921839   2.837861
```

```
cat('\nThe words that have the smallest coefficients with L2 are: \n')
```

```
##
## The words that have the smallest coefficients with L2 are:
```

```
head(sorted.l2, 5)
```

```
##   fitness   workout time.week     sugar  workouts
## -6.076931 -5.249456 -5.140047 -4.867361 -4.832190
```

L1 tends to tends to zero many coefficients while keeping the rest as they are. L2 tends to shrink all the coefficients and doesn't zero any.

Question 3

```r
ames        <- AmesHousing::make_ames()
numericVars <- ames %>% summarise_all(is.numeric) %>% unlist()
ames        <- ames[, numericVars]
dim(ames)
```

```
## [1] 2930   35
```

```r
head(ames)
```

```
## # A tibble: 6 x 35
##   Lot_Frontage Lot_Area Year_Built Year_Remod_Add Mas_Vnr_Area BsmtFin_SF_1
##          <dbl>    <int>      <int>          <int>        <dbl>        <dbl>
## 1          141    31770       1960           1960          112            2
## 2           80    11622       1961           1961            0            6
## 3           81    14267       1958           1958          108            1
## 4           93    11160       1968           1968            0            1
## 5           74    13830       1997           1998            0            3
## 6           78     9978       1998           1998           20            3
## # ... with 29 more variables: BsmtFin_SF_2 <dbl>, Bsmt_Unf_SF <dbl>,
## #   Total_Bsmt_SF <dbl>, First_Flr_SF <int>, Second_Flr_SF <int>,
## #   Low_Qual_Fin_SF <int>, Gr_Liv_Area <int>, Bsmt_Full_Bath <dbl>,
## #   Bsmt_Half_Bath <dbl>, Full_Bath <int>, Half_Bath <int>,
## #   Bedroom_AbvGr <int>, Kitchen_AbvGr <int>, TotRms_AbvGrd <int>,
## #   Fireplaces <int>, Garage_Cars <dbl>, Garage_Area <dbl>, Wood_Deck_SF <int>,
## #   Open_Porch_SF <int>, Enclosed_Porch <int>, Three_season_porch <int>,
## #   Screen_Porch <int>, Pool_Area <int>, Misc_Val <int>, Mo_Sold <int>,
## #   Year_Sold <int>, Sale_Price <int>, Longitude <dbl>, Latitude <dbl>
```

Forward Selection

Using forward selection, find the best model coefficients that predict Sale_Price

1. Run forward selection using `regsubsets` function.

```r
res <- regsubsets(Sale_Price ~ .,data=ames, method = "forward", nvmax=34)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 1 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to replace is
## not a multiple of replacement length
```

```r
smm <- summary(res)
smm
```

```
## Subset selection object
## Call: regsubsets.formula(Sale_Price ~ ., data = ames, method = "forward",
##     nvmax = 34)
## 34 Variables  (and intercept)
##                    Forced in Forced out
## Lot_Frontage           FALSE      FALSE
## Lot_Area               FALSE      FALSE
## Year_Built             FALSE      FALSE
## Year_Remod_Add         FALSE      FALSE
## Mas_Vnr_Area           FALSE      FALSE
## BsmtFin_SF_1           FALSE      FALSE
## BsmtFin_SF_2           FALSE      FALSE
## Bsmt_Unf_SF            FALSE      FALSE
## Total_Bsmt_SF          FALSE      FALSE
## First_Flr_SF           FALSE      FALSE
## Second_Flr_SF          FALSE      FALSE
## Low_Qual_Fin_SF        FALSE      FALSE
## Bsmt_Full_Bath         FALSE      FALSE
## Bsmt_Half_Bath         FALSE      FALSE
## Full_Bath              FALSE      FALSE
## Half_Bath              FALSE      FALSE
## Bedroom_AbvGr          FALSE      FALSE
## Kitchen_AbvGr          FALSE      FALSE
## TotRms_AbvGrd          FALSE      FALSE
## Fireplaces             FALSE      FALSE
## Garage_Cars            FALSE      FALSE
## Garage_Area            FALSE      FALSE
## Wood_Deck_SF           FALSE      FALSE
## Open_Porch_SF          FALSE      FALSE
## Enclosed_Porch         FALSE      FALSE
## Three_season_porch     FALSE      FALSE
## Screen_Porch           FALSE      FALSE
## Pool_Area              FALSE      FALSE
## Misc_Val               FALSE      FALSE
## Mo_Sold                FALSE      FALSE
## Year_Sold              FALSE      FALSE
## Longitude              FALSE      FALSE
## Latitude               FALSE      FALSE
## Gr_Liv_Area            FALSE      FALSE
## 1 subsets of each size up to 33
## Selection Algorithm: forward
##           Lot_Frontage Lot_Area Year_Built Year_Remod_Add Mas_Vnr_Area
## 1  ( 1 )  " "          " "      " "        " "            " "
## 2  ( 1 )  " "          " "      "*"        " "            " "
## 3  ( 1 )  " "          " "      "*"        " "            " "
## 4  ( 1 )  " "          " "      "*"        " "            " "
## 5  ( 1 )  " "          " "      "*"        " "            " "
```

```
## 6  ( 1 )  " "          " "       "*"        " "           " "
## 7  ( 1 )  " "          " "       "*"        "*"           " "
## 8  ( 1 )  " "          " "       "*"        "*"           "*"
## 9  ( 1 )  " "          " "       "*"        "*"           "*"
## 10  ( 1 ) " "          " "       "*"        "*"           "*"
## 11  ( 1 ) " "          " "       "*"        "*"           "*"
## 12  ( 1 ) " "          " "       "*"        "*"           "*"
## 13  ( 1 ) "*"          " "       "*"        "*"           "*"
## 14  ( 1 ) "*"          " "       "*"        "*"           "*"
## 15  ( 1 ) "*"          " "       "*"        "*"           "*"
## 16  ( 1 ) "*"          " "       "*"        "*"           "*"
## 17  ( 1 ) "*"          " "       "*"        "*"           "*"
## 18  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 19  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 20  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 21  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 22  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 23  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 24  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 25  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 26  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 27  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 28  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 29  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 30  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 31  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 32  ( 1 ) "*"          "*"       "*"        "*"           "*"
## 33  ( 1 ) "*"          "*"       "*"        "*"           "*"
##           BsmtFin_SF_1 BsmtFin_SF_2 Bsmt_Unf_SF Total_Bsmt_SF First_Flr_SF
## 1  ( 1 )  " "          " "          " "         " "           " "
## 2  ( 1 )  " "          " "          " "         " "           " "
## 3  ( 1 )  " "          " "          " "         "*"           " "
## 4  ( 1 )  " "          " "          " "         "*"           " "
## 5  ( 1 )  " "          " "          " "         "*"           " "
## 6  ( 1 )  " "          " "          " "         "*"           " "
## 7  ( 1 )  " "          " "          " "         "*"           " "
## 8  ( 1 )  " "          " "          " "         "*"           " "
## 9  ( 1 )  " "          " "          "*"         "*"           " "
## 10  ( 1 ) " "          " "          "*"         "*"           " "
## 11  ( 1 ) " "          " "          "*"         "*"           " "
## 12  ( 1 ) " "          " "          "*"         "*"           " "
## 13  ( 1 ) " "          " "          "*"         "*"           " "
## 14  ( 1 ) " "          " "          "*"         "*"           " "
## 15  ( 1 ) " "          " "          "*"         "*"           " "
## 16  ( 1 ) " "          " "          "*"         "*"           " "
## 17  ( 1 ) " "          " "          "*"         "*"           " "
## 18  ( 1 ) " "          " "          "*"         "*"           " "
## 19  ( 1 ) " "          "*"          "*"         "*"           " "
## 20  ( 1 ) " "          "*"          "*"         "*"           " "
## 21  ( 1 ) " "          "*"          "*"         "*"           " "
```

```
## 22  ( 1 ) " "          "*"          "*"         "*"           " "
## 23  ( 1 ) " "          "*"          "*"         "*"           " "
## 24  ( 1 ) " "          "*"          "*"         "*"           " "
## 25  ( 1 ) " "          "*"          "*"         "*"           " "
## 26  ( 1 ) " "          "*"          "*"         "*"           " "
## 27  ( 1 ) "*"          "*"          "*"         "*"           " "
## 28  ( 1 ) "*"          "*"          "*"         "*"           " "
## 29  ( 1 ) "*"          "*"          "*"         "*"           " "
## 30  ( 1 ) "*"          "*"          "*"         "*"           " "
## 31  ( 1 ) "*"          "*"          "*"         "*"           " "
## 32  ( 1 ) "*"          "*"          "*"         "*"           " "
## 33  ( 1 ) "*"          "*"          "*"         "*"           "*"
##           Second_Flr_SF Low_Qual_Fin_SF Gr_Liv_Area Bsmt_Full_Bath
## 1  ( 1 )  " "           " "             "*"         " "
## 2  ( 1 )  " "           " "             "*"         " "
## 3  ( 1 )  " "           " "             "*"         " "
## 4  ( 1 )  " "           " "             "*"         " "
## 5  ( 1 )  " "           " "             "*"         " "
## 6  ( 1 )  " "           " "             "*"         " "
## 7  ( 1 )  " "           " "             "*"         " "
## 8  ( 1 )  " "           " "             "*"         " "
## 9  ( 1 )  " "           " "             "*"         " "
## 10  ( 1 ) " "           " "             "*"         " "
## 11  ( 1 ) " "           " "             "*"         " "
## 12  ( 1 ) " "           " "             "*"         " "
## 13  ( 1 ) " "           " "             "*"         " "
## 14  ( 1 ) " "           " "             "*"         " "
## 15  ( 1 ) " "           " "             "*"         " "
## 16  ( 1 ) " "           " "             "*"         "*"
## 17  ( 1 ) " "           " "             "*"         "*"
## 18  ( 1 ) " "           " "             "*"         "*"
## 19  ( 1 ) " "           " "             "*"         "*"
## 20  ( 1 ) " "           " "             "*"         "*"
## 21  ( 1 ) " "           " "             "*"         "*"
## 22  ( 1 ) " "           "*"             "*"         "*"
## 23  ( 1 ) " "           "*"             "*"         "*"
## 24  ( 1 ) " "           "*"             "*"         "*"
## 25  ( 1 ) " "           "*"             "*"         "*"
## 26  ( 1 ) " "           "*"             "*"         "*"
## 27  ( 1 ) " "           "*"             "*"         "*"
## 28  ( 1 ) " "           "*"             "*"         "*"
## 29  ( 1 ) " "           "*"             "*"         "*"
## 30  ( 1 ) " "           "*"             "*"         "*"
## 31  ( 1 ) " "           "*"             "*"         "*"
## 32  ( 1 ) " "           "*"             "*"         "*"
## 33  ( 1 ) " "           "*"             "*"         "*"
##           Bsmt_Half_Bath Full_Bath Half_Bath Bedroom_AbvGr Kitchen_AbvGr
## 1  ( 1 )  " "            " "       " "       " "           " "
## 2  ( 1 )  " "            " "       " "       " "           " "
## 3  ( 1 )  " "            " "       " "       " "           " "
```

```
## 4  ( 1 )  " "            " "        " "        " "          " "
## 5  ( 1 )  " "            " "        " "        "*"          " "
## 6  ( 1 )  " "            " "        " "        "*"          "*"
## 7  ( 1 )  " "            " "        " "        "*"          "*"
## 8  ( 1 )  " "            " "        " "        "*"          "*"
## 9  ( 1 )  " "            " "        " "        "*"          "*"
## 10  ( 1 ) " "            " "        " "        "*"          "*"
## 11  ( 1 ) " "            " "        " "        "*"          "*"
## 12  ( 1 ) " "            " "        " "        "*"          "*"
## 13  ( 1 ) " "            " "        " "        "*"          "*"
## 14  ( 1 ) " "            " "        " "        "*"          "*"
## 15  ( 1 ) " "            " "        " "        "*"          "*"
## 16  ( 1 ) " "            " "        " "        "*"          "*"
## 17  ( 1 ) " "            " "        " "        "*"          "*"
## 18  ( 1 ) " "            " "        " "        "*"          "*"
## 19  ( 1 ) " "            " "        " "        "*"          "*"
## 20  ( 1 ) " "            " "        " "        "*"          "*"
## 21  ( 1 ) " "            " "        " "        "*"          "*"
## 22  ( 1 ) " "            " "        " "        "*"          "*"
## 23  ( 1 ) " "            " "        " "        "*"          "*"
## 24  ( 1 ) " "            " "        "*"        "*"          "*"
## 25  ( 1 ) " "            " "        "*"        "*"          "*"
## 26  ( 1 ) " "            "*"        "*"        "*"          "*"
## 27  ( 1 ) " "            "*"        "*"        "*"          "*"
## 28  ( 1 ) "*"            "*"        "*"        "*"          "*"
## 29  ( 1 ) "*"            "*"        "*"        "*"          "*"
## 30  ( 1 ) "*"            "*"        "*"        "*"          "*"
## 31  ( 1 ) "*"            "*"        "*"        "*"          "*"
## 32  ( 1 ) "*"            "*"        "*"        "*"          "*"
## 33  ( 1 ) "*"            "*"        "*"        "*"          "*"
##           TotRms_AbvGrd Fireplaces Garage_Cars Garage_Area Wood_Deck_SF
## 1  ( 1 )  " "           " "        " "         " "         " "
## 2  ( 1 )  " "           " "        " "         " "         " "
## 3  ( 1 )  " "           " "        " "         " "         " "
## 4  ( 1 )  " "           " "        "*"         " "         " "
## 5  ( 1 )  " "           " "        "*"         " "         " "
## 6  ( 1 )  " "           " "        "*"         " "         " "
## 7  ( 1 )  " "           " "        "*"         " "         " "
## 8  ( 1 )  " "           " "        "*"         " "         " "
## 9  ( 1 )  " "           " "        "*"         " "         " "
## 10  ( 1 ) " "           " "        "*"         " "         " "
## 11  ( 1 ) " "           "*"        "*"         " "         " "
## 12  ( 1 ) " "           "*"        "*"         " "         " "
## 13  ( 1 ) " "           "*"        "*"         " "         " "
## 14  ( 1 ) "*"           "*"        "*"         " "         " "
## 15  ( 1 ) "*"           "*"        "*"         " "         " "
## 16  ( 1 ) "*"           "*"        "*"         " "         " "
## 17  ( 1 ) "*"           "*"        "*"         " "         "*"
## 18  ( 1 ) "*"           "*"        "*"         " "         "*"
## 19  ( 1 ) "*"           "*"        "*"         " "         "*"
```

```
## 20  ( 1 ) "*"           "*"         "*"         " "         "*"
## 21  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 22  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 23  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 24  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 25  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 26  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 27  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 28  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 29  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 30  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 31  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 32  ( 1 ) "*"           "*"         "*"         "*"         "*"
## 33  ( 1 ) "*"           "*"         "*"         "*"         "*"
##           Open_Porch_SF Enclosed_Porch Three_season_porch Screen_Porch
## 1  ( 1 )  " "           " "            " "                " "
## 2  ( 1 )  " "           " "            " "                " "
## 3  ( 1 )  " "           " "            " "                " "
## 4  ( 1 )  " "           " "            " "                " "
## 5  ( 1 )  " "           " "            " "                " "
## 6  ( 1 )  " "           " "            " "                " "
## 7  ( 1 )  " "           " "            " "                " "
## 8  ( 1 )  " "           " "            " "                " "
## 9  ( 1 )  " "           " "            " "                " "
## 10  ( 1 ) " "           " "            " "                " "
## 11  ( 1 ) " "           " "            " "                " "
## 12  ( 1 ) " "           " "            " "                " "
## 13  ( 1 ) " "           " "            " "                " "
## 14  ( 1 ) " "           " "            " "                " "
## 15  ( 1 ) " "           " "            " "                "*"
## 16  ( 1 ) " "           " "            " "                "*"
## 17  ( 1 ) " "           " "            " "                "*"
## 18  ( 1 ) " "           " "            " "                "*"
## 19  ( 1 ) " "           " "            " "                "*"
## 20  ( 1 ) " "           " "            " "                "*"
## 21  ( 1 ) " "           " "            " "                "*"
## 22  ( 1 ) " "           " "            " "                "*"
## 23  ( 1 ) " "           "*"            " "                "*"
## 24  ( 1 ) " "           "*"            " "                "*"
## 25  ( 1 ) " "           "*"            " "                "*"
## 26  ( 1 ) " "           "*"            " "                "*"
## 27  ( 1 ) " "           "*"            " "                "*"
## 28  ( 1 ) " "           "*"            " "                "*"
## 29  ( 1 ) " "           "*"            " "                "*"
## 30  ( 1 ) "*"           "*"            " "                "*"
## 31  ( 1 ) "*"           "*"            "*"                "*"
## 32  ( 1 ) "*"           "*"            "*"                "*"
## 33  ( 1 ) "*"           "*"            "*"                "*"
##           Pool_Area Misc_Val Mo_Sold Year_Sold Longitude Latitude
## 1  ( 1 )  " "       " "      " "     " "       " "       " "
```

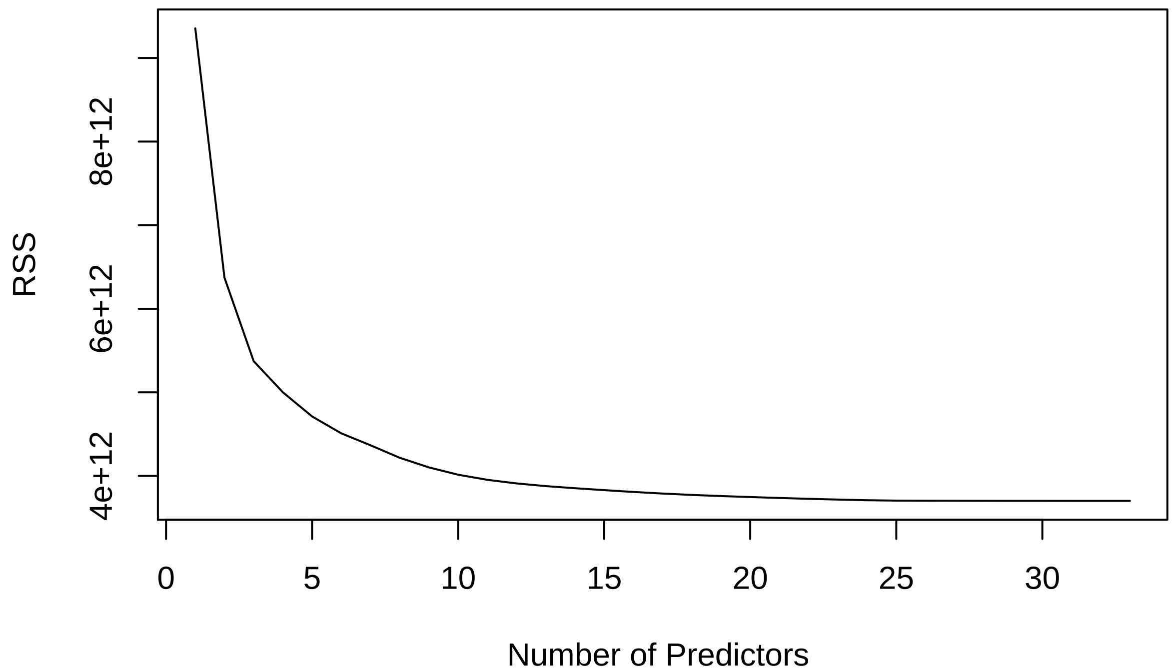
```
## 2  ( 1 )  " "        " "       " "      " "        " "        " "
## 3  ( 1 )  " "        " "       " "      " "        " "        " "
## 4  ( 1 )  " "        " "       " "      " "        " "        " "
## 5  ( 1 )  " "        " "       " "      " "        " "        " "
## 6  ( 1 )  " "        " "       " "      " "        " "        " "
## 7  ( 1 )  " "        " "       " "      " "        " "        " "
## 8  ( 1 )  " "        " "       " "      " "        " "        " "
## 9  ( 1 )  " "        " "       " "      " "        " "        " "
## 10  ( 1 ) " "        "*"       " "      " "        " "        " "
## 11  ( 1 ) " "        "*"       " "      " "        " "        " "
## 12  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 13  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 14  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 15  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 16  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 17  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 18  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 19  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 20  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 21  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 22  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 23  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 24  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 25  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 26  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 27  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 28  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 29  ( 1 ) "*"        "*"       " "      "*"        "*"        "*"
## 30  ( 1 ) "*"        "*"       " "      "*"        "*"        "*"
## 31  ( 1 ) "*"        "*"       " "      "*"        "*"        "*"
## 32  ( 1 ) "*"        "*"       "*"      "*"        "*"        "*"
## 33  ( 1 ) "*"        "*"       "*"      "*"        "*"        "*"
```

2. Extract the RSS of each model and plot. Your plot must have number of predictors on x axis and RSS on y axis.

```
smm$rss
```

```
##  [1] 9.354907e+12 6.372705e+12 5.372622e+12 5.000405e+12 4.711132e+12
##  [6] 4.509022e+12 4.366282e+12 4.216771e+12 4.101703e+12 4.014448e+12
## [11] 3.952959e+12 3.910112e+12 3.877808e+12 3.852701e+12 3.829707e+12
## [16] 3.808074e+12 3.788825e+12 3.772223e+12 3.759006e+12 3.747105e+12
## [21] 3.736053e+12 3.725905e+12 3.716953e+12 3.708821e+12 3.704526e+12
## [26] 3.703314e+12 3.702500e+12 3.701952e+12 3.701714e+12 3.701525e+12
## [31] 3.701381e+12 3.701365e+12 3.701352e+12
```

```
plot(smm$rss ,xlab="Number of Predictors ",ylab="RSS", type="l")
```



3. What number of predictors were used in the best model? What are the coefficients?

We can see that the RSS when only 1 variable `Gr_Living_Area` is included is the highest, at 9.35e+12, and when all predictors are included it is the lowest at 3.7e+12.

4. A friend of yours said that RSS is not a reliable measure and one must use BIC instead. Do all the steps you did for RSS. How many predictors resulted in the best model that yielded the minimum BIC?

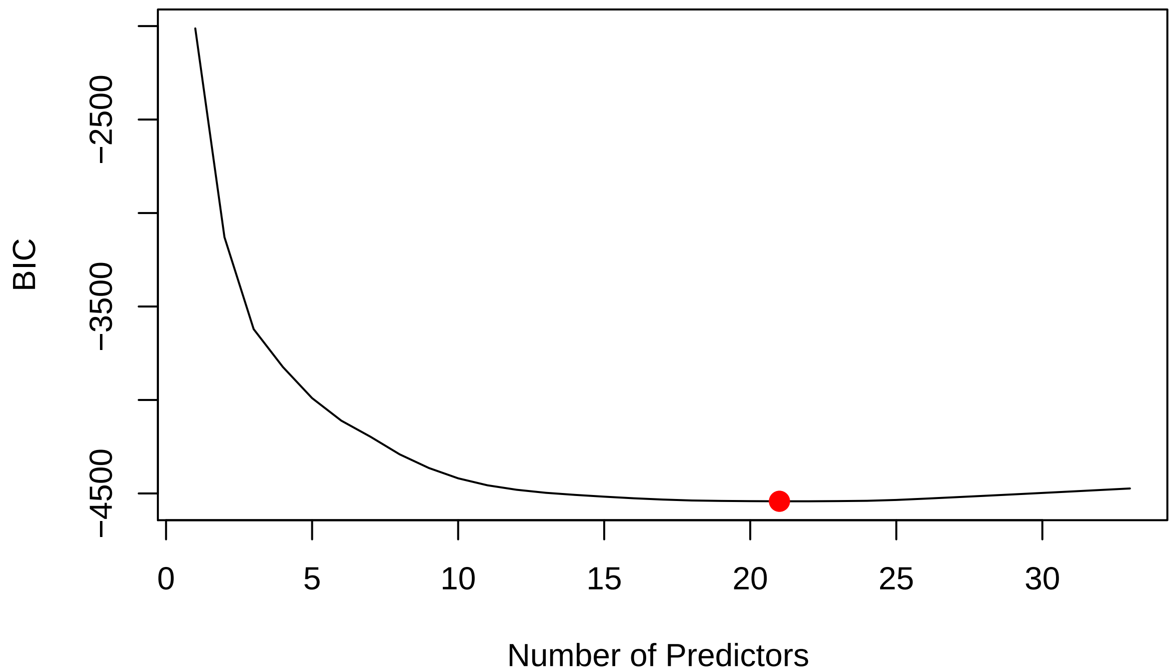
```
smm$bic
```

```
##  [1] -2012.249 -3129.026 -3621.217 -3823.600 -3990.218 -4110.710 -4196.981
##  [8] -4291.086 -4364.168 -4419.188 -4456.431 -4480.380 -4496.705 -4507.754
## [15] -4517.310 -4525.926 -4532.791 -4537.675 -4539.977 -4541.285 -4541.957
## [22] -4541.943 -4541.009 -4539.443 -4534.856 -4527.831 -4520.493 -4512.944
## [29] -4505.149 -4497.317 -4489.447 -4481.478 -4473.505
```

```
which.min(smm$bic )
```

```
## [1] 21
```

```r
plot(smm$bic ,xlab="Number of Predictors ",ylab="BIC", type="l")
points(21,smm$bic[21],col="red",cex=2,pch =20)
```



Using the BIC measure, the best model occurs when 21 predictors are used, and the minimum BIC is -4541.957

Backward Selection

1. Run backward selection using `regsubsets` function.

```r
res <- regsubsets(Sale_Price ~ .,data=ames, method = "backward", nvmax=34)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax, force.in =
## force.in, : 1 linear dependencies found

## Reordering variables and trying again:

## Warning in rval$lopt[] <- rval$vorder[rval$lopt]: number of items to replace is
## not a multiple of replacement length
```

```r
smm_bw <- summary(res)
smm_bw
```

```
## Subset selection object
## Call: regsubsets.formula(Sale_Price ~ ., data = ames, method = "backward",
##     nvmax = 34)
## 34 Variables  (and intercept)
##                    Forced in Forced out
## Lot_Frontage           FALSE      FALSE
## Lot_Area               FALSE      FALSE
## Year_Built             FALSE      FALSE
## Year_Remod_Add         FALSE      FALSE
## Mas_Vnr_Area           FALSE      FALSE
## BsmtFin_SF_1           FALSE      FALSE
## BsmtFin_SF_2           FALSE      FALSE
## Bsmt_Unf_SF            FALSE      FALSE
## Total_Bsmt_SF          FALSE      FALSE
## First_Flr_SF           FALSE      FALSE
## Second_Flr_SF          FALSE      FALSE
## Low_Qual_Fin_SF        FALSE      FALSE
## Bsmt_Full_Bath         FALSE      FALSE
## Bsmt_Half_Bath         FALSE      FALSE
## Full_Bath              FALSE      FALSE
## Half_Bath              FALSE      FALSE
## Bedroom_AbvGr          FALSE      FALSE
## Kitchen_AbvGr          FALSE      FALSE
## TotRms_AbvGrd          FALSE      FALSE
## Fireplaces             FALSE      FALSE
## Garage_Cars            FALSE      FALSE
## Garage_Area            FALSE      FALSE
## Wood_Deck_SF           FALSE      FALSE
## Open_Porch_SF          FALSE      FALSE
## Enclosed_Porch         FALSE      FALSE
## Three_season_porch     FALSE      FALSE
## Screen_Porch           FALSE      FALSE
## Pool_Area              FALSE      FALSE
## Misc_Val               FALSE      FALSE
## Mo_Sold                FALSE      FALSE
## Year_Sold              FALSE      FALSE
## Longitude              FALSE      FALSE
## Latitude               FALSE      FALSE
## Gr_Liv_Area            FALSE      FALSE
## 1 subsets of each size up to 33
## Selection Algorithm: backward
##          Lot_Frontage Lot_Area Year_Built Year_Remod_Add Mas_Vnr_Area
## 1  ( 1 ) " "          " "      " "        " "            " "
## 2  ( 1 ) " "          " "      " "        " "            " "
## 3  ( 1 ) " "          " "      "*"        " "            " "
## 4  ( 1 ) " "          " "      "*"        " "            " "
## 5  ( 1 ) " "          " "      "*"        " "            " "
## 6  ( 1 ) " "          " "      "*"        " "            " "
## 7  ( 1 ) " "          " "      "*"        "*"            " "
## 8  ( 1 ) " "          " "      "*"        "*"            " "
```

```
## 9  ( 1 )  " "           " "       "*"        "*"           "*"
## 10  ( 1 ) " "           " "       "*"        "*"           "*"
## 11  ( 1 ) " "           " "       "*"        "*"           "*"
## 12  ( 1 ) " "           " "       "*"        "*"           "*"
## 13  ( 1 ) " "           " "       "*"        "*"           "*"
## 14  ( 1 ) " "           " "       "*"        "*"           "*"
## 15  ( 1 ) "*"           " "       "*"        "*"           "*"
## 16  ( 1 ) "*"           " "       "*"        "*"           "*"
## 17  ( 1 ) "*"           " "       "*"        "*"           "*"
## 18  ( 1 ) "*"           " "       "*"        "*"           "*"
## 19  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 20  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 21  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 22  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 23  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 24  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 25  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 26  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 27  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 28  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 29  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 30  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 31  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 32  ( 1 ) "*"           "*"       "*"        "*"           "*"
## 33  ( 1 ) "*"           "*"       "*"        "*"           "*"
##           BsmtFin_SF_1 BsmtFin_SF_2 Bsmt_Unf_SF Total_Bsmt_SF First_Flr_SF
## 1  ( 1 )  " "          " "          " "         " "           "*"
## 2  ( 1 )  " "          " "          " "         " "           "*"
## 3  ( 1 )  " "          " "          " "         " "           "*"
## 4  ( 1 )  " "          " "          " "         " "           "*"
## 5  ( 1 )  " "          " "          " "         " "           "*"
## 6  ( 1 )  " "          " "          " "         "*"           "*"
## 7  ( 1 )  " "          " "          " "         "*"           "*"
## 8  ( 1 )  " "          " "          "*"         "*"           "*"
## 9  ( 1 )  " "          " "          "*"         "*"           "*"
## 10  ( 1 ) " "          " "          "*"         "*"           "*"
## 11  ( 1 ) " "          " "          "*"         "*"           "*"
## 12  ( 1 ) " "          " "          "*"         "*"           "*"
## 13  ( 1 ) " "          " "          "*"         "*"           "*"
## 14  ( 1 ) " "          " "          "*"         "*"           "*"
## 15  ( 1 ) " "          " "          "*"         "*"           "*"
## 16  ( 1 ) " "          " "          "*"         "*"           "*"
## 17  ( 1 ) " "          " "          "*"         "*"           "*"
## 18  ( 1 ) " "          " "          "*"         "*"           "*"
## 19  ( 1 ) " "          " "          "*"         "*"           "*"
## 20  ( 1 ) " "          "*"          "*"         "*"           "*"
## 21  ( 1 ) " "          "*"          "*"         "*"           "*"
## 22  ( 1 ) " "          "*"          "*"         "*"           "*"
## 23  ( 1 ) " "          "*"          "*"         "*"           "*"
## 24  ( 1 ) " "          "*"          "*"         "*"           "*"
```

```
## 25  ( 1 ) " "          "*"          "*"         "*"           "*"
## 26  ( 1 ) " "          "*"          "*"         "*"           "*"
## 27  ( 1 ) " "          "*"          "*"         "*"           "*"
## 28  ( 1 ) "*"          "*"          "*"         "*"           "*"
## 29  ( 1 ) "*"          "*"          "*"         "*"           "*"
## 30  ( 1 ) "*"          "*"          "*"         "*"           "*"
## 31  ( 1 ) "*"          "*"          "*"         "*"           "*"
## 32  ( 1 ) "*"          "*"          "*"         "*"           "*"
## 33  ( 1 ) "*"          "*"          "*"         "*"           "*"
##           Second_Flr_SF Low_Qual_Fin_SF Gr_Liv_Area Bsmt_Full_Bath
## 1  ( 1 )  " "           " "             " "         " "
## 2  ( 1 )  "*"           " "             " "         " "
## 3  ( 1 )  "*"           " "             " "         " "
## 4  ( 1 )  "*"           " "             " "         " "
## 5  ( 1 )  "*"           " "             " "         " "
## 6  ( 1 )  "*"           " "             " "         " "
## 7  ( 1 )  "*"           " "             " "         " "
## 8  ( 1 )  "*"           " "             " "         " "
## 9  ( 1 )  "*"           " "             " "         " "
## 10  ( 1 ) "*"           " "             " "         " "
## 11  ( 1 ) "*"           " "             " "         " "
## 12  ( 1 ) "*"           " "             " "         " "
## 13  ( 1 ) "*"           " "             " "         " "
## 14  ( 1 ) "*"           " "             " "         " "
## 15  ( 1 ) "*"           " "             " "         " "
## 16  ( 1 ) "*"           " "             " "         " "
## 17  ( 1 ) "*"           " "             " "         "*"
## 18  ( 1 ) "*"           " "             " "         "*"
## 19  ( 1 ) "*"           " "             " "         "*"
## 20  ( 1 ) "*"           " "             " "         "*"
## 21  ( 1 ) "*"           " "             " "         "*"
## 22  ( 1 ) "*"           " "             " "         "*"
## 23  ( 1 ) "*"           " "             " "         "*"
## 24  ( 1 ) "*"           " "             " "         "*"
## 25  ( 1 ) "*"           " "             " "         "*"
## 26  ( 1 ) "*"           "*"             " "         "*"
## 27  ( 1 ) "*"           "*"             " "         "*"
## 28  ( 1 ) "*"           "*"             " "         "*"
## 29  ( 1 ) "*"           "*"             " "         "*"
## 30  ( 1 ) "*"           "*"             " "         "*"
## 31  ( 1 ) "*"           "*"             " "         "*"
## 32  ( 1 ) "*"           "*"             " "         "*"
## 33  ( 1 ) "*"           "*"             " "         "*"
##           Bsmt_Half_Bath Full_Bath Half_Bath Bedroom_AbvGr Kitchen_AbvGr
## 1  ( 1 )  " "            " "       " "       " "           " "
## 2  ( 1 )  " "            " "       " "       " "           " "
## 3  ( 1 )  " "            " "       " "       " "           " "
## 4  ( 1 )  " "            " "       " "       " "           "*"
## 5  ( 1 )  " "            " "       " "       " "           "*"
## 6  ( 1 )  " "            " "       " "       " "           "*"
```

```
## 7  ( 1 )  " "            " "        " "        " "           "*"
## 8  ( 1 )  " "            " "        " "        " "           "*"
## 9  ( 1 )  " "            " "        " "        " "           "*"
## 10  ( 1 ) " "            " "        " "        "*"           "*"
## 11  ( 1 ) " "            " "        " "        "*"           "*"
## 12  ( 1 ) " "            " "        " "        "*"           "*"
## 13  ( 1 ) " "            " "        " "        "*"           "*"
## 14  ( 1 ) " "            " "        " "        "*"           "*"
## 15  ( 1 ) " "            " "        " "        "*"           "*"
## 16  ( 1 ) " "            " "        " "        "*"           "*"
## 17  ( 1 ) " "            " "        " "        "*"           "*"
## 18  ( 1 ) " "            " "        " "        "*"           "*"
## 19  ( 1 ) " "            " "        " "        "*"           "*"
## 20  ( 1 ) " "            " "        " "        "*"           "*"
## 21  ( 1 ) " "            " "        " "        "*"           "*"
## 22  ( 1 ) " "            " "        " "        "*"           "*"
## 23  ( 1 ) " "            " "        " "        "*"           "*"
## 24  ( 1 ) " "            " "        "*"        "*"           "*"
## 25  ( 1 ) " "            " "        "*"        "*"           "*"
## 26  ( 1 ) " "            " "        "*"        "*"           "*"
## 27  ( 1 ) " "            "*"        "*"        "*"           "*"
## 28  ( 1 ) " "            "*"        "*"        "*"           "*"
## 29  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 30  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 31  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 32  ( 1 ) "*"            "*"        "*"        "*"           "*"
## 33  ( 1 ) "*"            "*"        "*"        "*"           "*"
##           TotRms_AbvGrd Fireplaces Garage_Cars Garage_Area Wood_Deck_SF
## 1  ( 1 )  " "           " "        " "         " "         " "
## 2  ( 1 )  " "           " "        " "         " "         " "
## 3  ( 1 )  " "           " "        " "         " "         " "
## 4  ( 1 )  " "           " "        " "         " "         " "
## 5  ( 1 )  " "           " "        "*"         " "         " "
## 6  ( 1 )  " "           " "        "*"         " "         " "
## 7  ( 1 )  " "           " "        "*"         " "         " "
## 8  ( 1 )  " "           " "        "*"         " "         " "
## 9  ( 1 )  " "           " "        "*"         " "         " "
## 10  ( 1 ) " "           " "        "*"         " "         " "
## 11  ( 1 ) " "           " "        "*"         " "         " "
## 12  ( 1 ) " "           "*"        "*"         " "         " "
## 13  ( 1 ) " "           "*"        "*"         " "         " "
## 14  ( 1 ) "*"           "*"        "*"         " "         " "
## 15  ( 1 ) "*"           "*"        "*"         " "         " "
## 16  ( 1 ) "*"           "*"        "*"         " "         " "
## 17  ( 1 ) "*"           "*"        "*"         " "         " "
## 18  ( 1 ) "*"           "*"        "*"         " "         "*"
## 19  ( 1 ) "*"           "*"        "*"         " "         "*"
## 20  ( 1 ) "*"           "*"        "*"         " "         "*"
## 21  ( 1 ) "*"           "*"        "*"         " "         "*"
## 22  ( 1 ) "*"           "*"        "*"         "*"         "*"
```

```
## 23  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 24  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 25  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 26  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 27  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 28  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 29  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 30  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 31  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 32  ( 1 ) "*"           "*"         "*"          "*"          "*"
## 33  ( 1 ) "*"           "*"         "*"          "*"          "*"
##           Open_Porch_SF Enclosed_Porch Three_season_porch Screen_Porch
## 1  ( 1 )  " "           " "            " "                " "
## 2  ( 1 )  " "           " "            " "                " "
## 3  ( 1 )  " "           " "            " "                " "
## 4  ( 1 )  " "           " "            " "                " "
## 5  ( 1 )  " "           " "            " "                " "
## 6  ( 1 )  " "           " "            " "                " "
## 7  ( 1 )  " "           " "            " "                " "
## 8  ( 1 )  " "           " "            " "                " "
## 9  ( 1 )  " "           " "            " "                " "
## 10  ( 1 ) " "           " "            " "                " "
## 11  ( 1 ) " "           " "            " "                " "
## 12  ( 1 ) " "           " "            " "                " "
## 13  ( 1 ) " "           " "            " "                " "
## 14  ( 1 ) " "           " "            " "                " "
## 15  ( 1 ) " "           " "            " "                " "
## 16  ( 1 ) " "           " "            " "                "*"
## 17  ( 1 ) " "           " "            " "                "*"
## 18  ( 1 ) " "           " "            " "                "*"
## 19  ( 1 ) " "           " "            " "                "*"
## 20  ( 1 ) " "           " "            " "                "*"
## 21  ( 1 ) " "           " "            " "                "*"
## 22  ( 1 ) " "           " "            " "                "*"
## 23  ( 1 ) " "           "*"            " "                "*"
## 24  ( 1 ) " "           "*"            " "                "*"
## 25  ( 1 ) " "           "*"            " "                "*"
## 26  ( 1 ) " "           "*"            " "                "*"
## 27  ( 1 ) " "           "*"            " "                "*"
## 28  ( 1 ) " "           "*"            " "                "*"
## 29  ( 1 ) " "           "*"            " "                "*"
## 30  ( 1 ) " "           "*"            " "                "*"
## 31  ( 1 ) "*"           "*"            " "                "*"
## 32  ( 1 ) "*"           "*"            "*"                "*"
## 33  ( 1 ) "*"           "*"            "*"                "*"
##           Pool_Area Misc_Val Mo_Sold Year_Sold Longitude Latitude
## 1  ( 1 )  " "       " "      " "     " "       " "       " "
## 2  ( 1 )  " "       " "      " "     " "       " "       " "
## 3  ( 1 )  " "       " "      " "     " "       " "       " "
## 4  ( 1 )  " "       " "      " "     " "       " "       " "
```

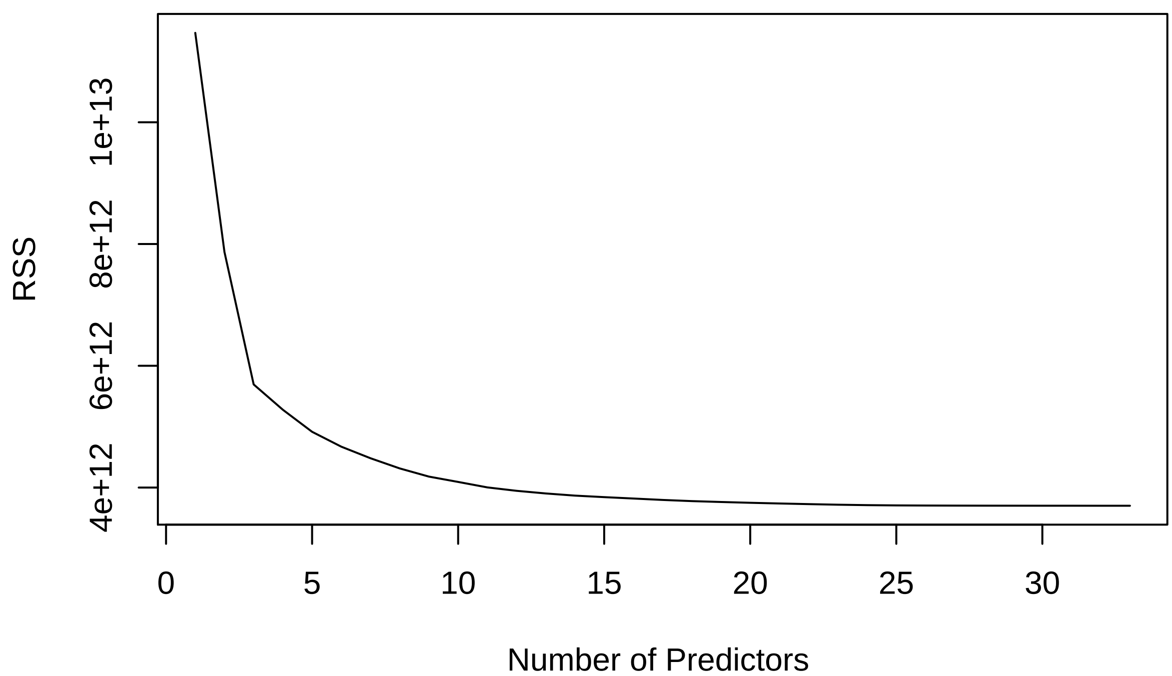
```
## 5  ( 1 )  " "        " "       " "      " "        " "        " "
## 6  ( 1 )  " "        " "       " "      " "        " "        " "
## 7  ( 1 )  " "        " "       " "      " "        " "        " "
## 8  ( 1 )  " "        " "       " "      " "        " "        " "
## 9  ( 1 )  " "        " "       " "      " "        " "        " "
## 10  ( 1 ) " "        " "       " "      " "        " "        " "
## 11  ( 1 ) " "        "*"       " "      " "        " "        " "
## 12  ( 1 ) " "        "*"       " "      " "        " "        " "
## 13  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 14  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 15  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 16  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 17  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 18  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 19  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 20  ( 1 ) " "        "*"       " "      " "        " "        "*"
## 21  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 22  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 23  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 24  ( 1 ) "*"        "*"       " "      " "        " "        "*"
## 25  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 26  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 27  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 28  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 29  ( 1 ) "*"        "*"       " "      "*"        " "        "*"
## 30  ( 1 ) "*"        "*"       " "      "*"        "*"        "*"
## 31  ( 1 ) "*"        "*"       " "      "*"        "*"        "*"
## 32  ( 1 ) "*"        "*"       " "      "*"        "*"        "*"
## 33  ( 1 ) "*"        "*"       "*"      "*"        "*"        "*"
```

2. Extract the RSS of each model and plot. Your plot must have number of predictors on x axis and RSS on y axis.

```
smm_bw$rss
```

```
##  [1] 1.146822e+13 7.869601e+12 5.693659e+12 5.277521e+12 4.915896e+12
##  [6] 4.671204e+12 4.481447e+12 4.314304e+12 4.179940e+12 4.092241e+12
## [11] 4.002680e+12 3.946271e+12 3.902998e+12 3.867500e+12 3.842522e+12
## [16] 3.819776e+12 3.796777e+12 3.777550e+12 3.763157e+12 3.750030e+12
## [21] 3.738591e+12 3.727819e+12 3.718299e+12 3.711271e+12 3.707002e+12
## [26] 3.704526e+12 3.703298e+12 3.702482e+12 3.701936e+12 3.701699e+12
## [31] 3.701509e+12 3.701367e+12 3.701352e+12
```

```
plot(smm_bw$rss ,xlab="Number of Predictors ",ylab="RSS", type="l")
```



3.

What number of predictors were used in the best model? What are the coefficients?

We can see that the RSS when only 1 variable `First_Flr_SF` is included is the highest, at 1.15+e13, and when all predictors are included it is the lowest at 3.7e+12.

4. A friend of yours said that RSS is not a reliable measure and one must use BIC instead. Do all the steps you did for RSS. How many predictors resulted in the best model that yielded the minimum BIC?

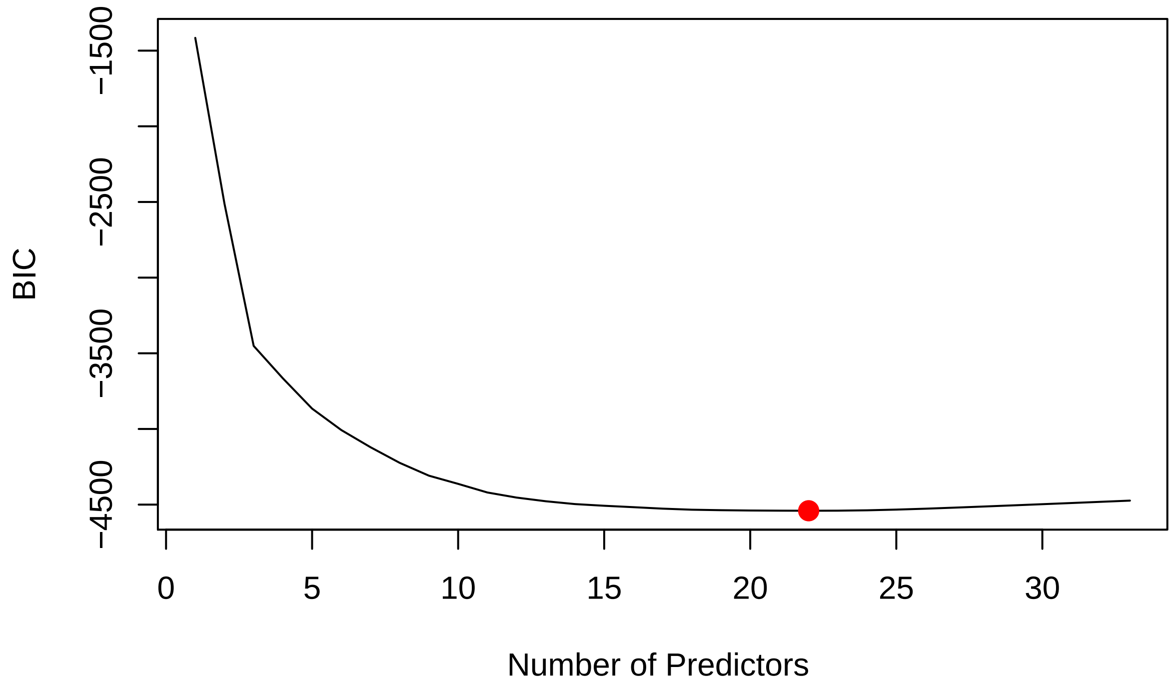
```
smm_bw$bic
```

```
##  [1] -1415.469 -2510.844 -3451.169 -3665.563 -3865.559 -4007.173 -4120.700
##  [8] -4224.087 -4308.807 -4362.953 -4419.806 -4453.409 -4477.733 -4496.521
## [15] -4507.522 -4516.935 -4526.648 -4533.540 -4536.742 -4538.999 -4539.967
## [22] -4540.439 -4539.948 -4537.508 -4532.898 -4526.873 -4519.862 -4512.524
## [29] -4504.974 -4497.179 -4489.346 -4481.476 -4473.505
```

```
which.min(smm_bw$bic )
```

```
## [1] 22
```

```
plot(smm_bw$bic ,xlab="Number of Predictors ",ylab="BIC", type="l")
points(22,smm_bw$bic[22],col="red",cex=2,pch =20)
```



Using the BIC measure with backwards selection, the best model occurs when 22 predictors are used, and the minimum BIC is -4540.439.