BUS/CSC 328 Homework #5
Cross-Validation
Aaron, Hamza, Selemawit

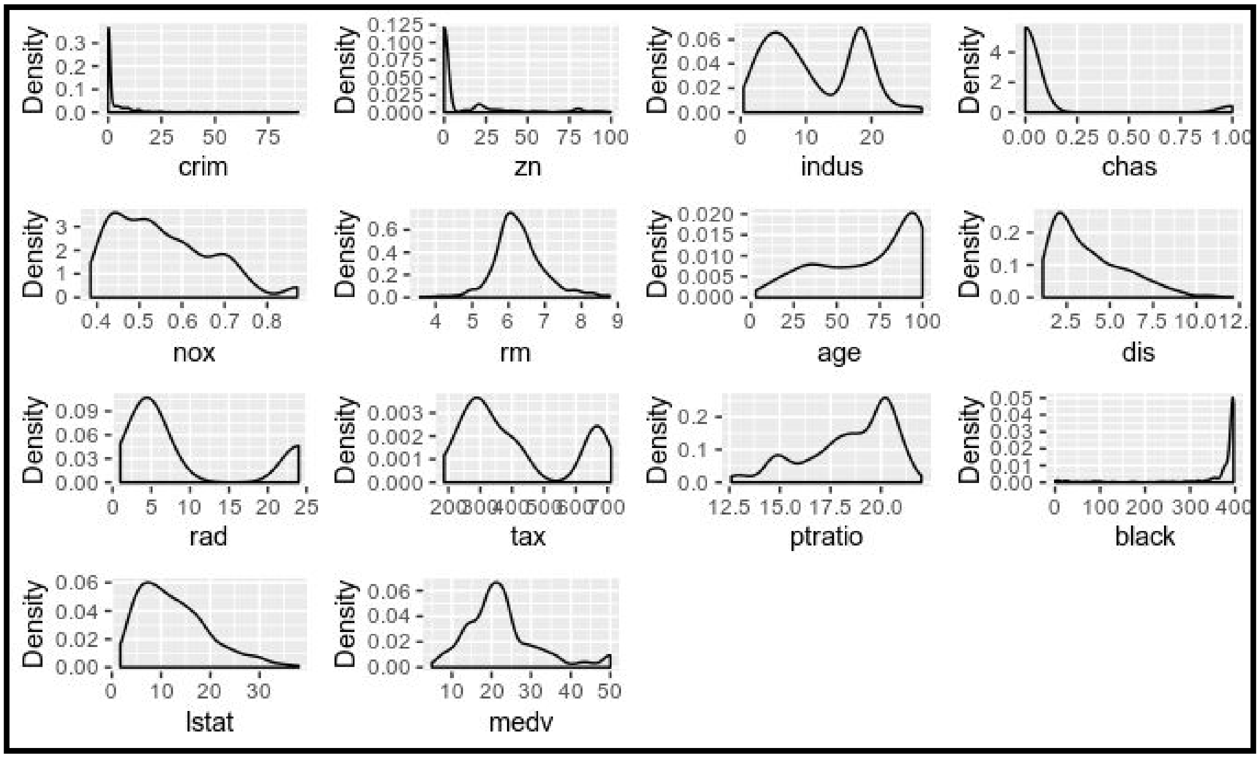
Step 1: Import the Boston dataset from the MASS package, and inspect for number of observations, number of variables, and data types of variables. NOTE: medv, the median price of a home, is the dependent variable. Put this information into your homework report.

Number of observations: 506
Number of variables: 13
Data types of variables: Doubles and Integers

```
> glimpse(data_raw)
Observations: 506
Variables: 14
$ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.…
$ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5…
$ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87,…
$ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
$ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524…
$ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012…
$ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1,…
$ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622,…
$ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4,…
$ tax     <dbl> 296, 242, 242, 222, 222, 222, 311, 311, 311, 31…
$ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2,…
$ black   <dbl> 396.90, 396.90, 392.83, 394.63, 396.90, 394.12,…
$ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.1…
$ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1,…
```

Step 2: Examine the correlations and density plots for the predictors. Put this information into your homework report and note any problem areas regarding linear regression.
Some of the predictors like rad and tax are highly correlated with each other, so we need to be careful as this might affect the accuracy of the model. The predictors are also non normal, so we should normalize them.

```
Correlations/Type of Correlation:
           medv     crim      zn   indus      chas     nox      rm     age     dis     rad     tax ptratio   black   lstat
medv          1  Pearson Pearson Pearson   Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson
crim    -0.3883        1 Pearson Pearson   Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson
zn       0.3604  -0.2005       1 Pearson   Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson
indus   -0.4837   0.4066 -0.5338       1   Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson
chas     0.1753 -0.05589 -0.0427 0.06294         1 Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson
nox     -0.4273    0.421 -0.5166  0.7637    0.0912       1 Pearson Pearson Pearson Pearson Pearson Pearson Pearson Pearson
rm       0.6954  -0.2192   0.312 -0.3917   0.09125 -0.3022       1 Pearson Pearson Pearson Pearson Pearson Pearson Pearson
age      -0.377   0.3527 -0.5695  0.6448   0.08652  0.7315 -0.2403       1 Pearson Pearson Pearson Pearson Pearson Pearson
dis      0.2499  -0.3797  0.6644  -0.708  -0.09918 -0.7692  0.2052 -0.7479       1 Pearson Pearson Pearson Pearson Pearson
rad     -0.3816   0.6255 -0.3119  0.5951 -0.007368  0.6114 -0.2098   0.456 -0.4946       1 Pearson Pearson Pearson Pearson
tax     -0.4685   0.5828 -0.3146  0.7208  -0.03559   0.668  -0.292  0.5065 -0.5344  0.9102       1 Pearson Pearson Pearson
ptratio -0.5078   0.2899 -0.3917  0.3832   -0.1215  0.1889 -0.3555  0.2615 -0.2325  0.4647  0.4609       1 Pearson Pearson
black    0.3335  -0.3851  0.1755  -0.357   0.04879 -0.3801  0.1281 -0.2735  0.2915 -0.4444 -0.4418 -0.1774       1 Pearson
lstat   -0.7377   0.4556  -0.413  0.6038  -0.05393  0.5909 -0.6138  0.6023  -0.497  0.4887   0.544   0.374 -0.3661       1
```



Step 3: Construct a recipe that addresses any issues with the data. Put this information into your homework report along with the recipe output.

```
Data Recipe

Inputs:

      role #variables
   outcome          1
 predictor         13

Training data contained 506 data points and no missing data.

Operations:

Box-Cox transformation on crim, indus, nox, rm, age, dis, rad, ... [trained]
```

Step 4: Split the data into training and test sets, with a 70/30 split function using a seed value of your choice. Put this information into your homework report.

```
> train_test_split
<355/151/506>
>
> train_clean = training(train_test_split)
> dim(train_clean)
[1] 355  14
> test_clean = testing(train_test_split)
> dim(test_clean)
[1] 151  14
```

Step 5: Build and run a linear regression model using the training data, and put the estimates, standard error, R-squared, and F statistics into table titled "Run 1." Put this information into your homework report.

Run	Estimates	Standard Error	R-squared	F-statistics
Run 1	Estimate 2.270e+02 3.099e-01 1.480e-02 -3.124e-01 2.558e+00 -4.703e+00 6.805e+00 4.854e-03 -8.040e+00 1.957e+00 -1.102e+02 -3.740e-05 1.700e-09 -6.389e+00	Std. Error 4.887e+01 2.851e-01 1.500e-02 2.572e-01 1.024e+00 1.823e+00 1.269e+00 4.303e-03 1.242e+00 7.239e-01 2.689e+01 8.988e-06 1.064e-09 4.134e-01	**0.7715**	**92.93**

Step 6: Change the seed value of the split function and create new training and test datasets.

```
set.seed(4534)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_clean2 = training(train_test_split)
dim(train_clean2)
test_clean2 = testing(train_test_split)
dim(test_clean2)
```

Step 7: Build and run a linear regression model using the new training data, and put the estimates, standard error, RSE, R-squared, and F statistics into table titled "Run 2."

Repeat this process for a total of five runs. Put this information into your homework report.

Run	Estimates	Standard Error	RSE	R-squared	F-statistics
Run 2	Estimate 2.266e+02 8.062e-01 1.581e-02 -4.873e-01 1.920e+00 -6.905e+00 7.374e+00 -8.296e-05 -9.287e+00 1.313e+00 -1.114e+02 -4.073e-05 3.707e-09 -5.481e+00	Std. Error 5.074e+01 2.810e-01 1.498e-02 2.593e-01 1.040e+00 1.823e+00 1.232e+00 4.024e-03 1.218e+00 6.900e-01 2.815e+01 8.554e-06 1.049e-09 3.969e-01	**4.477**	**0.7729**	**93.67**
Run 3	Estimate 2.318e+02 4.852e-01 2.206e-02 -3.676e-01 3.179e+00 -6.573e+00 6.281e+00 4.309e-03 -8.184e+00 1.677e+00 -1.135e+02 -4.406e-05 2.026e-09 -5.586e+00	Std. Error 5.011e+01 2.703e-01 1.533e-02 2.462e-01 1.027e+00 1.754e+00 1.363e+00 3.933e-03 1.184e+00 7.129e-01 2.773e+01 8.723e-06 1.044e-09 4.205e-01	**4.394**	**0.7639**	**89.11**

Run 4	Estimate 1.962e+02 4.029e-01 2.066e-02 -4.019e-01 2.560e+00 -4.148e+00 7.559e+00 1.935e-03 -7.686e+00 9.172e-01 -9.460e+01 -4.084e-05 2.705e-09 -5.445e+00	Std. Error 5.057e+01 2.644e-01 1.466e-02 2.457e-01 9.362e-01 1.747e+00 1.304e+00 3.963e-03 1.150e+00 6.792e-01 2.805e+01 8.448e-06 1.036e-09 4.137e-01	**4.318**	**0.7735**	**93.99**
Run 5	Estimate) 2.125e+02 5.147e-01 1.744e-02 -4.161e-01 2.659e+00 -6.429e+00 7.578e+00 7.721e-03 -6.991e+00 1.133e+00 -1.058e+02 -4.229e-05 2.276e-09 -5.423e+00	Std. Error 4.647e+01 2.717e-01 1.436e-02 2.430e-01 9.278e-01 1.760e+00 1.394e+00 4.085e-03 1.213e+00 6.894e-01 2.549e+01 8.512e-06 1.085e-09 4.368e-01	**4.413**	**0.7708**	**92.6**
Run 6	Estimate 2.005e+02 2.794e-01 2.682e-02 -3.370e-01 2.772e+00 -6.103e+00 6.844e+00 3.395e-03 -7.664e+00 1.753e+00 -9.775e+01 -3.513e-05 1.701e-09 -5.588e+00	Std. Error 4.720e+01 2.581e-01 1.369e-02 2.492e-01 8.848e-01 1.614e+00 1.163e+00 3.745e-03 1.138e+00 6.270e-01 2.621e+01 7.799e-06 9.904e-10 3.840e-01	**4.092**	**0.7856**	**100.8**
Mean			**4.3388**	**0.77334**	**93.85**

Step 8: Make a new table that shows the five runs for the coefficient estimates, standard errors, RSE, R-squared, and F statistics. Compute the mean for each of these statistics and put those into a column. Put this information into your homework report.

Step 9: Split the data again into training and test sets, with a 70/30 split function using a new seed value of your choice.

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.106e+02  4.519e+01   4.661 4.52e-06 ***
crim         5.549e-01  2.641e-01   2.101 0.036382 *  
zn           1.935e-02  1.458e-02   1.327 0.185425    
indus       -3.106e-01  2.387e-01  -1.301 0.194097    
chas         2.972e+00  1.007e+00   2.952 0.003377 ** 
nox         -5.920e+00  1.689e+00  -3.506 0.000516 ***
rm           6.689e+00  1.221e+00   5.480 8.29e-08 ***
age          1.851e-03  3.886e-03   0.476 0.634208    
dis         -7.058e+00  1.136e+00  -6.213 1.51e-09 ***
rad          1.385e+00  6.880e-01   2.013 0.044931 *  
tax         -1.022e+02  2.491e+01  -4.101 5.14e-05 ***
ptratio     -4.038e-05  8.675e-06  -4.655 4.65e-06 ***
black        1.508e-09  1.007e-09   1.498 0.134969    
lstat       -5.622e+00  3.961e-01 -14.194  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.277 on 341 degrees of freedom
Multiple R-squared:  0.7745,	Adjusted R-squared:  0.7659 
F-statistic: 90.07 on 13 and 341 DF,  p-value: < 2.2e-16
```

Step 10: Build and run a cross-validated linear regression model using the training data with k=10 and three repeats, and put the estimates, standard error, R-squared, and F statistics into table titled “CV Run.” Put this information into your homework report.

CV Run	Estimates	Standard Error	R-squared	F-statistics

	Estimate 2.302e+02 3.600e-01 1.882e-02 -2.813e-01 2.898e+00 -6.559e+00 1.039e+01 -1.829e-03 -8.508e+00 1.165e+00 -1.196e+02 -4.494e-05 1.730e-09 -4.497e+00	Std. Error 4.757e+01 2.642e-01 1.496e-02 2.484e-01 1.094e+00 1.870e+00 1.498e+00 4.177e-03 1.218e+00 7.102e-01 2.628e+01 8.896e-06 1.078e-09 4.904e-01	**0.773**	**93.73**

Step 11: Compare the means of the standard models with the cross-validated model and put this evaluation into your homework report.

The response from the CV run shows that the cross validated model has a lower F-statistic and a lower R-squared value, which makes the standard models slightly better. This means that the average of 5 previous models is better than the CV values but the values are in close proximity explaining why a single CV model would be much better than a singular linear regression model.

Step 12: Using the cross-validated model, create a prediction against the test dataset and put this information into your homework report.

```
> defaultSummary(modelvalues)
     RMSE  Rsquared       MAE
4.5088760 0.7359841 3.2725307
>
```

```
library(MASS)
library(recipes)
library(rsample)
library(car)
library(DataExplorer)
library(polycor)
library(tidyverse)
library(ROCR)

data("Boston")
data_raw = Boston
glimpse(data_raw)

data_raw = data_raw %>% select(medv,everything())

plot_density(data_raw)
hetcor(data_raw)

pancakes = recipe(medv ~ ., data=data_raw) %>%
  step_BoxCox(all_predictors(), -all_outcomes()) %>%
  prep(data=data_raw)

pancakes

data_clean = bake(pancakes, new_data=data_raw)

#First linear model
set.seed(20000)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_clean = training(train_test_split)
dim(train_clean)
test_clean = testing(train_test_split)
dim(test_clean)

lm.fit = lm(medv ~ ., data=train_clean)
summary(lm.fit)

#Second linear model
set.seed(4534)
```

```
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_clean2 = training(train_test_split)
test_clean2 = testing(train_test_split)

lm.fit2 = lm(medv ~ ., data=train_clean2)
summary(lm.fit2)

#Third linear model
set.seed(76574)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_clean3 = training(train_test_split)
test_clean3 = testing(train_test_split)

lm.fit3 = lm(medv ~ ., data=train_clean3)
summary(lm.fit3)

#Fourth linear model
set.seed(9375)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_clean4 = training(train_test_split)
test_clean4 = testing(train_test_split)

lm.fit4 = lm(medv ~ ., data=train_clean4)
summary(lm.fit4)

#Fifth linear model
set.seed(1)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_clean5 = training(train_test_split)
test_clean5 = testing(train_test_split)

lm.fit5 = lm(medv ~ ., data=train_clean5)
summary(lm.fit5)

#Sixth linear model
```

```
set.seed(986754)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_clean6 = training(train_test_split)
test_clean6 = testing(train_test_split)

lm.fit6 = lm(medv ~ ., data=train_clean6)
summary(lm.fit6)




#cross validated linear model
set.seed(2500)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

B_train_clean = training(train_test_split)
B_test_clean = testing(train_test_split)

# Build the linear regression model with embedded cross validation

ctrl<-trainControl(method = "repeatedcv", number = 10, repeats = 3, summaryFunction =
defaultSummary)
BTown.cv.fit <- train(medv ~ ., data = B_train_clean , method = "lm", trControl = ctrl,
metric= "Rsquared")

summary(BTown.cv.fit)

#Cross validation
set.seed(2345)
train_test_split = initial_split(data_clean, prop=0.70)
train_test_split

train_cleanCV = training(train_test_split)
test_cleanCV = testing(train_test_split)

ctrl<-trainControl(method = "repeatedcv" ,number = 10, repeats = 3, summaryFunction =
defaultSummary)
Beantown.cv.fit <-train(medv ~ ., data = train_cleanCV, method = "lm", trControl = ctrl,
metric= "Rsquared")
```

```
summary(Beantown.cv.fit)
names(Beantown.cv.fit)

predCV = predict(Beantown.cv.fit, test_cleanCV)
modelvalues<-data.frame(obs = test_cleanCV$medv, pred=predCV)
defaultSummary(modelvalues)
```