## Demonstrating Train/Test Split for Model Validation on Credit Dataset

Libraries

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 3.5.2

## -- Attaching packages ---------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.2.5  
## v tibble 1.4.2 v dplyr 0.7.7  
## v tidyr 0.8.2 v stringr 1.3.1  
## v readr 1.1.1 v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(GGally) #for ggpairs function

## Warning: package 'GGally' was built under R version 3.5.2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':  
##   
## nasa

library(MASS) #access to forward and backward selection algorithms

## Warning: package 'MASS' was built under R version 3.5.2

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(leaps) #best subset selection

## Warning: package 'leaps' was built under R version 3.5.2

library(caret) #for splitting for validation

## Warning: package 'caret' was built under R version 3.5.2

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

Read-in dataset

credit = read\_csv("CreditData.csv")

## Parsed with column specification:  
## cols(  
## AnnualIncome = col\_double(),  
## HouseholdSize = col\_integer(),  
## YrsEdAfterHS = col\_integer(),  
## HrWkTV = col\_integer(),  
## AnnualCharges = col\_double()  
## )

Get rid of missing data rows (**any data cleaning/prepartion should take place before splitting**).

credit = credit %>% drop\_na() #delete any row with an NA value  
str(credit) #check structure after the drop

## Classes 'tbl\_df', 'tbl' and 'data.frame': 5000 obs. of 5 variables:  
## $ AnnualIncome : num 21.8 65.5 54.2 73.7 110.4 ...  
## $ HouseholdSize: int 4 7 3 6 7 8 5 8 1 3 ...  
## $ YrsEdAfterHS : int 5 3 2 0 5 3 4 5 4 1 ...  
## $ HrWkTV : int 29 46 18 44 39 39 40 27 15 3 ...  
## $ AnnualCharges: num 10024 11249 6115 9786 12634 ...

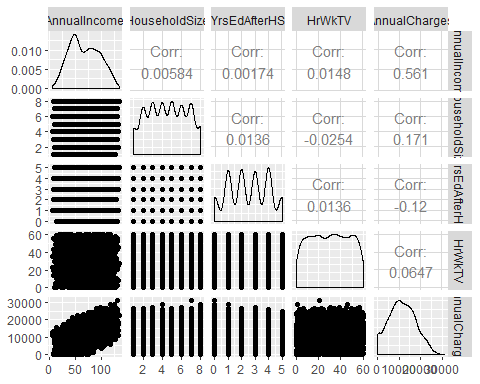
Split the data (training and testing)

set.seed(123)  
train.rows = createDataPartition(y = credit$AnnualCharges, p=0.7, list = FALSE) #70% in training  
train = credit[train.rows,]   
test = credit[-train.rows,]

AFTER you split, then do visualization and modeling with the **training set**.

Our Y (response) variable in this dataset is “AnnualCharges”. Let’s look at ggpairs plot for visualization and correlation.

ggpairs(train)



Model with best single variable (by correlation).

mod1 = lm(AnnualCharges ~ AnnualIncome, train) #create linear regression model  
summary(mod1) #examine the model

##   
## Call:  
## lm(formula = AnnualCharges ~ AnnualIncome, data = train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -12156.9 -3968.6 -39.8 3993.3 12778.9   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3192.94 221.06 14.44 <2e-16 \*\*\*  
## AnnualIncome 120.94 3.02 40.04 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5048 on 3498 degrees of freedom  
## Multiple R-squared: 0.3143, Adjusted R-squared: 0.3141   
## F-statistic: 1603 on 1 and 3498 DF, p-value: < 2.2e-16

Let’s assume (for the sake of time) that this model is our best model. The R squared value for this model on the training set is around 0.3. Now we need to evaluate its performance on the testing set. Typically, we will see performance degrade a bit. If we see severe degradation, we assume that may have overfit the training set.

Develop predictions on the testing set

test\_preds = predict(mod1, newdata = test)

Now we can manually calculate the R squared value.

SSE = sum((test$AnnualCharges - test\_preds)^2) #sum of squared residuals from model  
SST = sum((test$AnnualCharges - mean(test$AnnualCharges))^2) #sum of squared residuals from a "naive" model  
1 - SSE/SST #definition of R squared

## [1] 0.3176969