Lab02_tm1

Load Packages

The following R code loads packages needed in this assignment.

```
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(GGally)
```

```
##
## Attaching package: 'GGally'
```

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

```
library(caret)
```

```
## Loading required package: lattice
```

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
drivers <- read_csv("data/bad-drivers.csv")
```

```
## Parsed with column specification:
## cols(
##   State = col_character(),
##   `Number of drivers involved in fatal collisions per billion miles` = col_double(),
##   `Percentage Of Drivers Involved In Fatal Collisions Who Were Speeding` = col_integer(),
##   `Percentage Of Drivers Involved In Fatal Collisions Who Were Alcohol-Impaired` = col_integer(),
##   `Percentage Of Drivers Involved In Fatal Collisions Who Were Not Distracted` = col_integer(),
##   `Percentage Of Drivers Involved In Fatal Collisions Who Had Not Been Involved In Any Previous Acci
##   `Car Insurance Premiums ($)` = col_double(),
##   `Losses incurred by insurance companies for collisions per insured driver ($)` = col_double()
## )
```

```
names(drivers)[2] <-"DriverNum"
names(drivers)[3] <-"Speeding"
names(drivers)[4] <-"Alcohol"
names(drivers)[5] <-"Distraction"
names(drivers)[6] <-"History"
names(drivers)[7] <-"CIP"
names(drivers)[8] <-"Loss"

names(drivers)
```

```
## [1] "State"       "DriverNum"   "Speeding"    "Alcohol"     "Distraction"
## [6] "History"     "CIP"         "Loss"
```

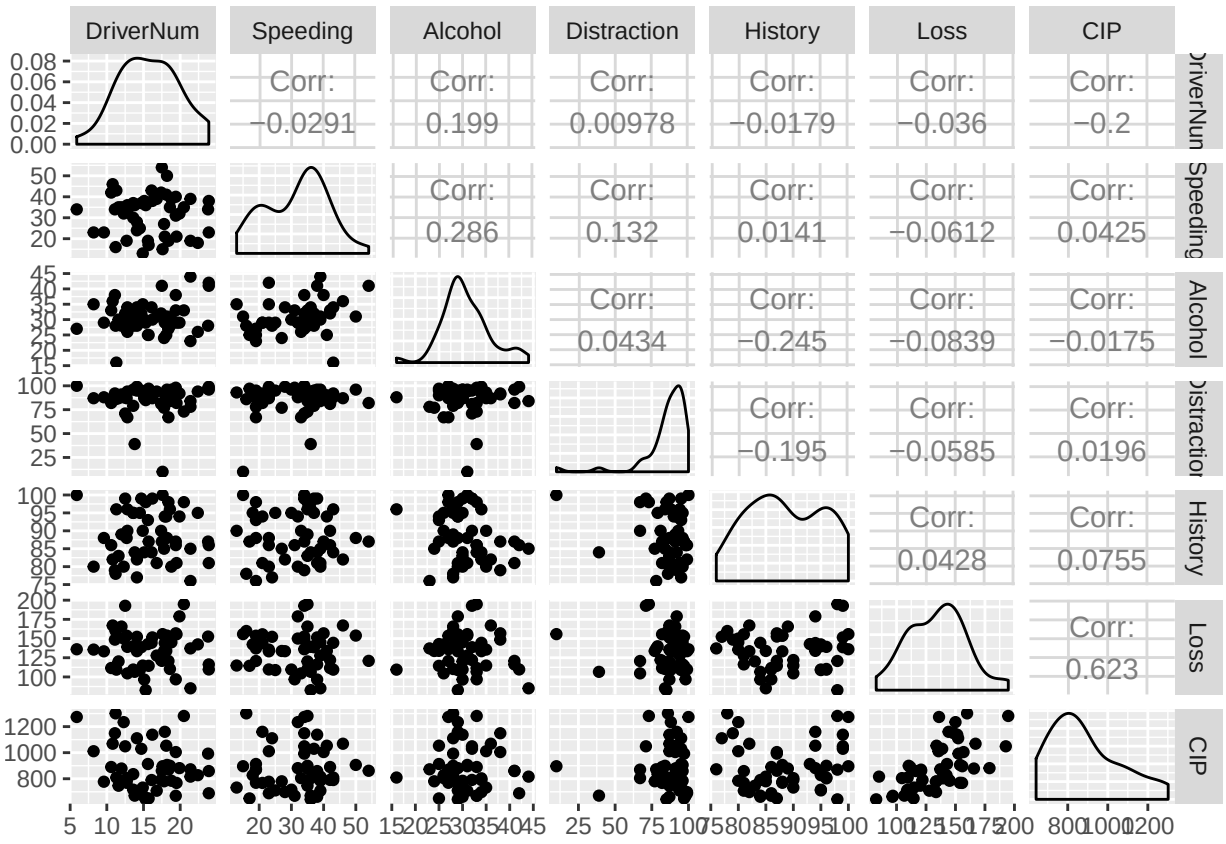
Including Plots

You can also embed plots, for example:

```
##     State             DriverNum        Speeding        Alcohol
##  Length:51          Min.   : 5.90   Min.   :13.00   Min.   :16.00
##  Class :character   1st Qu.:12.75   1st Qu.:23.00   1st Qu.:28.00
##  Mode  :character   Median :15.60   Median :34.00   Median :30.00
##                     Mean   :15.79   Mean   :31.73   Mean   :30.69
##                     3rd Qu.:18.50   3rd Qu.:38.00   3rd Qu.:33.00
##                     Max.   :23.90   Max.   :54.00   Max.   :44.00
##   Distraction        History           CIP             Loss
##  Min.   : 10.00   Min.   : 76.00   Min.   : 642.0   Min.   : 82.75
##  1st Qu.: 83.00   1st Qu.: 83.50   1st Qu.: 768.4   1st Qu.:114.64
##  Median : 88.00   Median : 88.00   Median : 859.0   Median :136.05
##  Mean   : 85.92   Mean   : 88.73   Mean   : 887.0   Mean   :134.49
##  3rd Qu.: 95.00   3rd Qu.: 95.00   3rd Qu.:1007.9   3rd Qu.:151.87
##  Max.   :100.00   Max.   :100.00   Max.   :1301.5   Max.   :194.78
```

```
## [1] 51  8
```

```
## # A tibble: 6 x 8
##   State      DriverNum Speeding Alcohol Distraction History   CIP  Loss
##   <chr>          <dbl>    <int>   <int>       <int>   <int> <dbl> <dbl>
## 1 Alabama         18.8       39      30          96      80  785.  145.
## 2 Alaska          18.1       41      25          90      94 1053.  134.
## 3 Arizona         18.6       35      28          84      96  899.  110.
## 4 Arkansas        22.4       18      26          94      95  827.  142.
## 5 California      12         35      28          91      89  878.  166.
## 6 Colorado        13.6       37      28          79      95  836.  140.
```




```
reg01 <- lm(CIP~Loss, data = drivers)
summary(reg01)
```

```
##
## Call:
## lm(formula = CIP ~ Loss, data = drivers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -213.33  -96.75  -40.11  112.24  379.97
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 285.3251   109.6689   2.602   0.0122 *
## Loss          4.4733     0.8021   5.577 1.04e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 140.9 on 49 degrees of freedom
## Multiple R-squared:  0.3883, Adjusted R-squared:  0.3758
## F-statistic:  31.1 on 1 and 49 DF,  p-value: 1.043e-06
```

```
confint(reg01,level = 0.95)
```

```
##                 2.5 %     97.5 %
## (Intercept) 64.937209 505.712968
## Loss         2.861401   6.085265
```

```r
reg02 <-lm(CIP~(Loss+Alcohol), data = drivers)
summary(reg02)
```

```
##
## Call:
## lm(formula = CIP ~ (Loss + Alcohol), data = drivers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -211.72  -97.95  -41.45  108.43  384.43
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 245.0788   170.6089   1.436    0.157
## Loss          4.4945     0.8125   5.532 1.29e-06 ***
## Alcohol       1.2189     3.9318   0.310    0.758
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.2 on 48 degrees of freedom
## Multiple R-squared:  0.3895, Adjusted R-squared:  0.3641
## F-statistic: 15.31 on 2 and 48 DF,  p-value: 7.186e-06
```

```r
confint(reg02,level = 0.95)
```

```
##                   2.5 %     97.5 %
## (Intercept) -97.953427 588.111062
## Loss          2.860841   6.128098
## Alcohol      -6.686590   9.124392
```

```r
set.seed(67)
train_val_inds <- caret::createDataPartition(
  y = drivers$CIP,
  p = 0.8
)
train_val_inds
```

```
## $Resample1
##  [1]  1  2  3  5  6  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [24] 26 27 28 30 31 32 33 34 37 38 39 40 41 42 43 46 47 49 50 51
```

```r
driver_train_val <- drivers %>% slice(train_val_inds[[1]])
driver_test <- drivers %>% slice(-train_val_inds[[1]])

crossval_fold_inds <- caret::createFolds(
  y = driver_train_val$CIP,
  k = 5
)


train_val_mse <- expand.grid(
  reg = seq_len(2),
  val_fold_num = seq_len(5),
  train_mse = NA,
  val_mse = NA
)
```

```
for(reg in seq_len(2)){
  for (val_fold_num in seq_len(5)){
    results_index <- which(
      train_val_mse$reg == reg &
      train_val_mse$val_fold_num == val_fold_num
    )
    driver_train <- driver_train_val %>% slice(-crossval_fold_inds[[val_fold_num]])
    driver_val <- driver_train_val %>% slice(crossval_fold_inds[[val_fold_num]])
    if (reg == 1){
      fit <- lm(CIP~Loss,data = driver_train)
    }else{
      fit <- lm(CIP~(Loss+Alcohol),data = driver_train)
    }

    train_resids<- driver_train$CIP - predict(fit)
    train_val_mse$train_mse[results_index] <- mean(train_resids^2)

    val_resids<- driver_val$CIP - predict(fit , driver_val)
    train_val_mse$val_mse[results_index] <- mean(val_resids^2) #mean(val_resids^2)

  }
}

train_val_mse
```

```
##    reg val_fold_num train_mse  val_mse
## 1    1            1  18680.18 28665.50
## 2    2            1  18357.00 29087.68
## 3    1            2  21144.13 19398.36
## 4    2            2  20845.58 19918.06
## 5    1            3  19626.75 25349.54
## 6    2            3  19537.27 27610.87
## 7    1            4  22260.93 14168.78
## 8    2            4  21621.71 16529.61
## 9    1            5  21281.34 21750.94
## 10   2            5  21086.00 21919.32
```

```
summarized_crossval_mse_results <- train_val_mse %>%
  group_by(reg) %>%
  summarize(
    crossval_mse = mean(val_mse)
  )
summarized_crossval_mse_results
```

```
## # A tibble: 2 x 2
##     reg crossval_mse
##   <int>        <dbl>
## 1     1       21867.
## 2     2       23013.
```

Discussion

Please explain your model, making sure to reference the coefficients of the model. You should discuss any relevant hypothesis tests or confidence intervals as appropriate. reg01: coefficeients: predictive: Loss, response:CIP hypothesis: a p-value of 1.043e-06 shows strong rejection confintL 2.5 % 97.5 % (Intercept) 64.937209 505.712968 Loss 2.861401 6.085265

reg02: coefficeients: predictive: Loss, response:CIP+Alcohol hypothesis: a p-value of 7.186e-06 shows strong rejection 2.5 % 97.5 % (Intercept) -97.953427 588.111062 Loss 2.860841 6.128098 Alcohol -6.686590 9.124392

How does your multiple regression model compare to the simple linear regression model, and how would you communicate these results to an audience? From the summary output of both models, the simple linear model is better because the second predictive variable in multiple regression model is not significant, although both models are significant.

How does the cross-validation MSE compare between your simple and multiple regression models? What does this mean? The multiple regression model has better training data performence, but preforms worse in validation data; it also shows the same pattern in average MSE. Based on the limited info provided, a simple linear regression is better.