Sarah_Mya_Megi- Lab02

Sarah Mya Megi

9/20/2019

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.

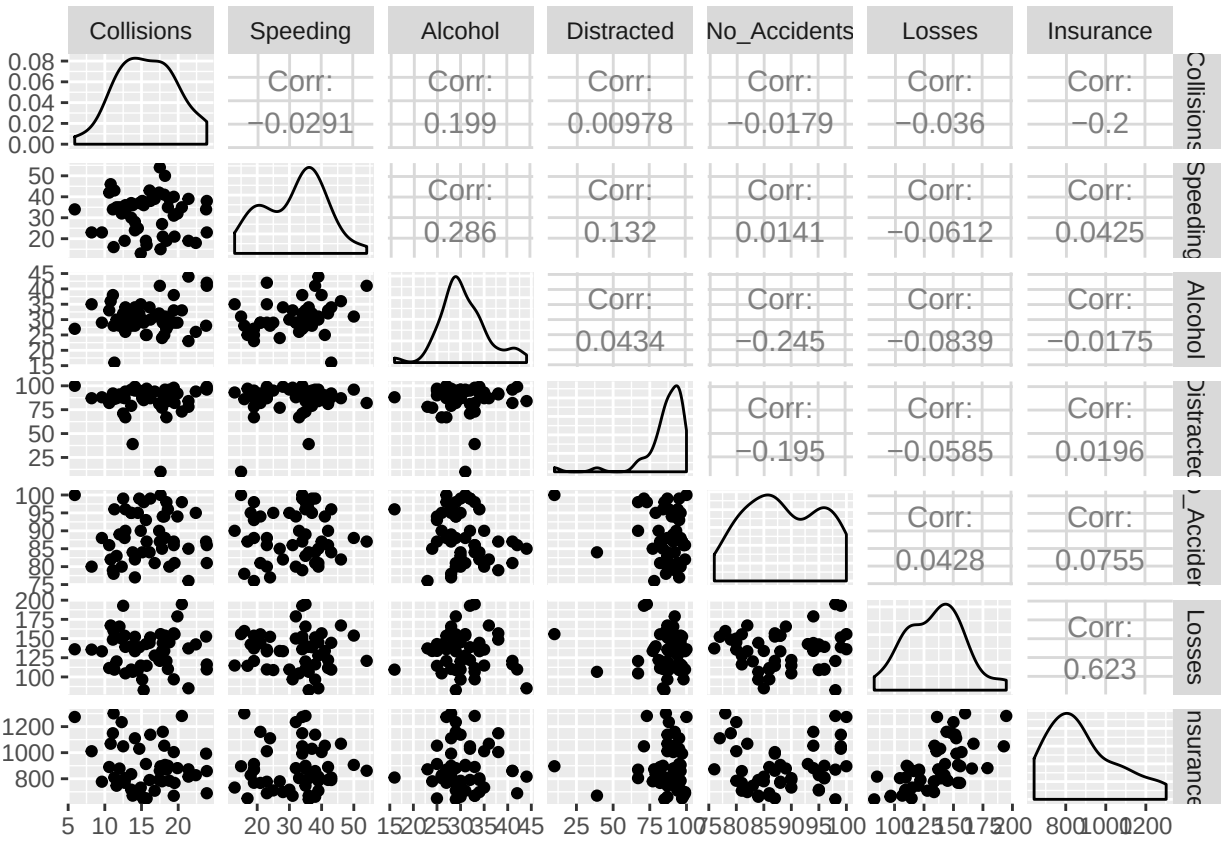
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
cars <- read.csv("data/bad-drivers.csv")
names(cars) <- c("State", "Collisions", "Speeding", "Alcohol", "Distracted", "No_Accidents", "Insurance
summary(cars)
```

```
##         State      Collisions       Speeding        Alcohol
##  Alabama   : 1   Min.   : 5.90   Min.   :13.00   Min.   :16.00
##  Alaska    : 1   1st Qu.:12.75   1st Qu.:23.00   1st Qu.:28.00
##  Arizona   : 1   Median :15.60   Median :34.00   Median :30.00
##  Arkansas  : 1   Mean   :15.79   Mean   :31.73   Mean   :30.69
##  California: 1   3rd Qu.:18.50   3rd Qu.:38.00   3rd Qu.:33.00
##  Colorado  : 1   Max.   :23.90   Max.   :54.00   Max.   :44.00
##  (Other)   :45
##    Distracted      No_Accidents      Insurance          Losses
##  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
##
```

Including Plots

You can also embed plots, for example:



We focused on the Losses (x) and Insurance(y) scatterplot; this plot was the only we we could observe that that a somewhat trend occurring.

Regression Analysis

```
lm_fit <- lm(Insurance ~ Losses, data = cars)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = Insurance ~ Losses, data = cars)
##
## 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 *
## Losses        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
```

```r
reg01 <- function(x){
  predict(lm_fit, data.frame(Losses = x))}

lm_fit2 <- lm(Insurance ~ Losses + Collisions + Speeding + Alcohol + Distracted + No_Accidents, data = c
summary(lm_fit2)
```

```
##
## Call:
## lm(formula = Insurance ~ Losses + Collisions + Speeding + Alcohol +
##     Distracted + No_Accidents, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -209.09  -99.64  -28.18   86.58  303.57
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   68.5876   385.0879   0.178    0.859
## Losses         4.4929     0.8180   5.492 1.87e-06 ***
## Collisions    -8.2340     5.0278  -1.638    0.109
## Speeding       0.8149     2.2326   0.365    0.717
## Alcohol        2.6505     4.3730   0.606    0.548
## Distracted     0.7497     1.3747   0.545    0.588
## No_Accidents   1.9444     3.0739   0.633    0.530
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.8 on 44 degrees of freedom
## Multiple R-squared:  0.4355, Adjusted R-squared:  0.3585
## F-statistic: 5.658 on 6 and 44 DF,  p-value: 0.0002014
```

```r
reg02 <- function(x) {
  predict(quad_fit, data.frame(Losses = x))}
```

```r
set.seed(7304)
cars_val_inds <-  caret:: createDataPartition(
  y = cars$Insurance,
  p  = 0.8
  )

cars_val_inds
```

```
## $Resample1
##  [1]  1  3  4  5  7  8  9 10 11 12 14 15 17 18 19 20 21 22 23 24 26 27 28
## [24] 29 30 32 33 34 35 36 37 38 39 40 42 43 45 46 47 48 49 50 51
```

```r
cars_train_val <- cars %>% slice(cars_val_inds[[1]])
cars_test <- cars %>% slice(-cars_val_inds [[1]])

num_crossval_folds <- 5
crossval_folds_inds <- caret::createFolds(
  y = cars_train_val$Insurance,
  k = num_crossval_folds
)
```

```r
train_val_mse <- expand.grid(
  i = seq_len(5),
  simple_val_mse = NA,
  multiple_val_mse = NA
)

for(i in seq_len(5)) {

    cars_train <- cars_train_val %>% slice(-crossval_folds_inds[[i]])
    cars_val <- cars_train_val %>% slice(crossval_folds_inds[[i]])

    fit <- lm(Insurance ~ Losses, data = cars_train)
    train_resids <- cars_val$Insurance - predict(fit, newdata = cars_val)
    train_val_mse$simple_val_mse[i] <- mean(train_resids^2)

    fit <- lm(Insurance ~ Losses + Collisions + Speeding + Alcohol + Distracted + No_Accidents, data = c
    train_resids <- cars_val$Insurance - predict(fit, newdata = cars_val)
    train_val_mse$multiple_val_mse[i] <- mean(train_resids^2)
  }
```

```r
head(train_val_mse)
```

```
##   i simple_val_mse multiple_val_mse
## 1 1       19226.68         47489.86
## 2 2       25238.34         27274.11
## 3 3       29031.59         27420.36
## 4 4       12183.09         19962.51
## 5 5       16919.02         17602.00
```

```r
summarized_crossval_mse_results <- train_val_mse %>%
  summarize(
    crossval_mse = mean(simple_val_mse)
  )
summarized_crossval_mse_results
```

```
##   crossval_mse
## 1     20519.75
```

```r
summarized_crossval_mse_results <- train_val_mse %>%
  summarize(
    crossval_mse = mean(multiple_val_mse)
  )
summarized_crossval_mse_results
```

```
##   crossval_mse
## 1     27949.77
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.