Lab 02

Helena Littman and Katerina Alvarez & Remy Wang (kalva914 & helenalittman & RLWang)

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.

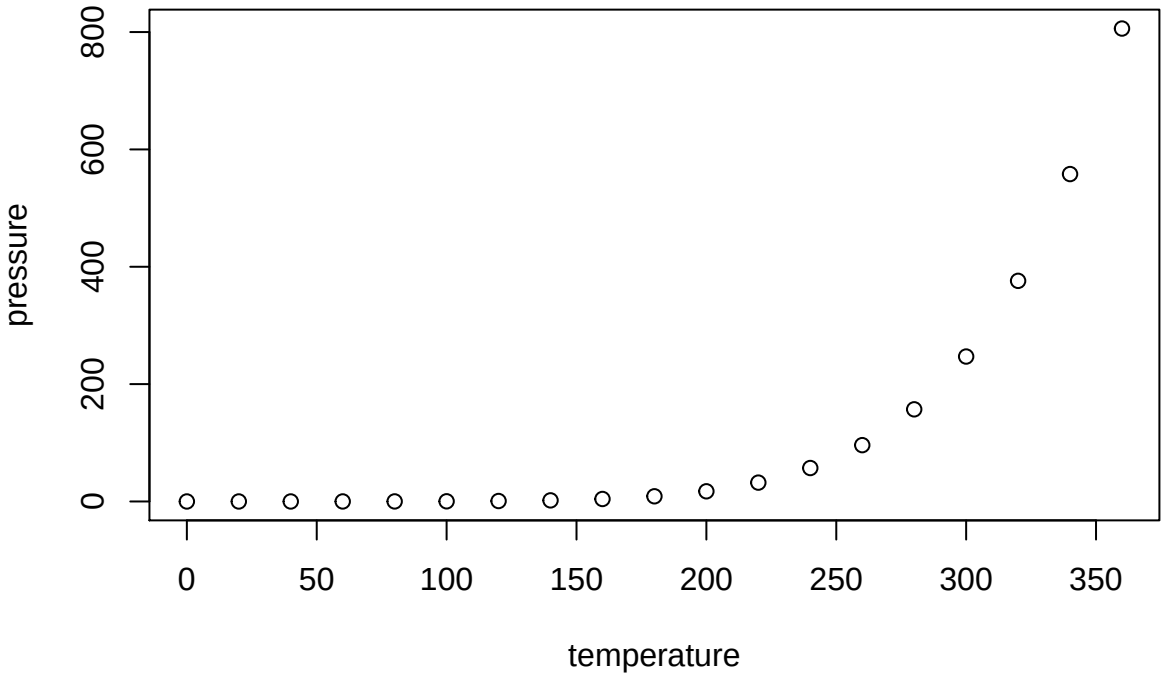
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



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

Lab Tasks

- read in the data set `data/bad-drivers.csv`
- (recommended) rename the columns to shorter nicknames (check out the `names` function)

bad_drivers <- read.csv("data/bad-drivers.csv")

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

- exploratory data analysis
- present some pictures and a brief description of trends you see in the data, and how they may influence fitting a model.

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

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

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

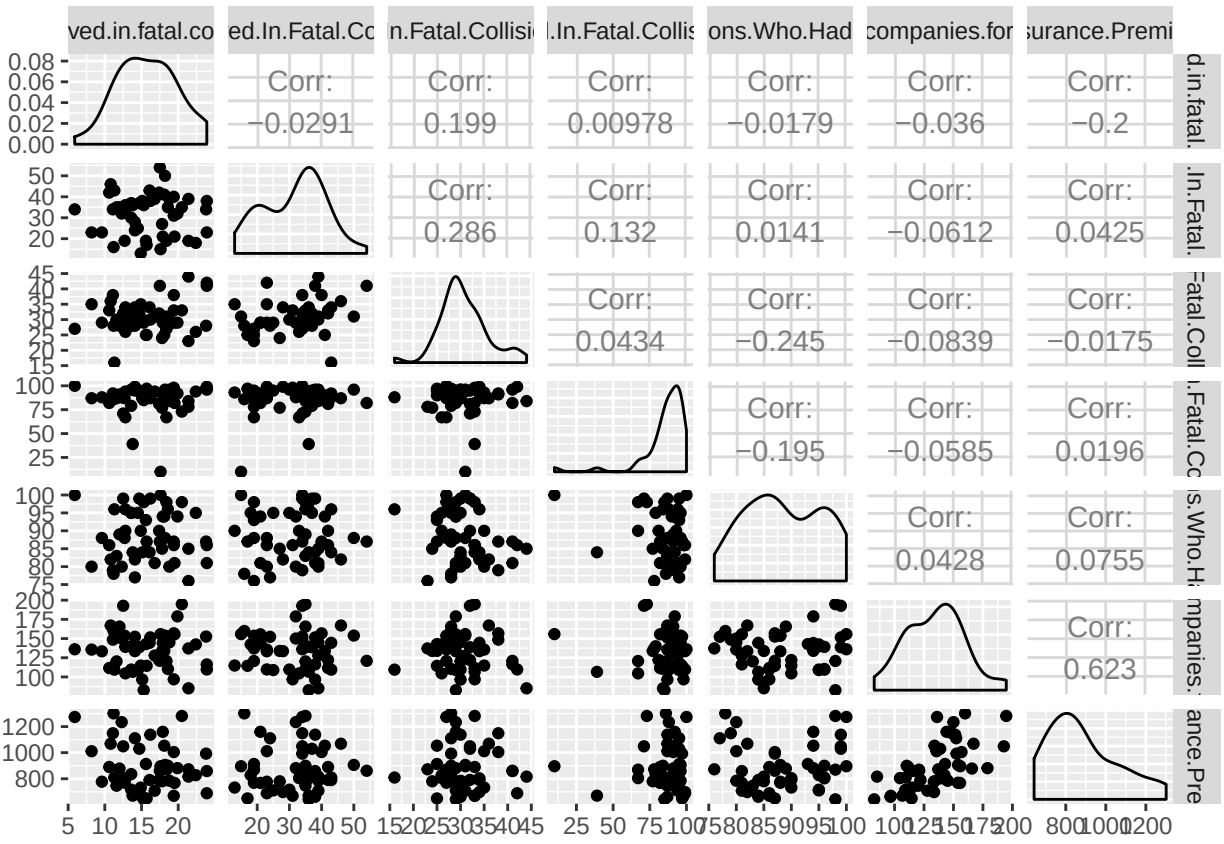
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
attach(bad_drivers)

vars_to_use <- c("Number.of.drivers.involved.in.fatal.collisions.per.billion.miles","Percentage.Of.Drive

ggpairs(bad_drivers %>% select(vars_to_use))
```



Only one plot shows a linear relationship (explanatory variable = Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....). The rest don't show relationships and are scatterred.

- regression analysis
- The target variable for our regression models is `Car Insurance Premiums ($)`
- fit a simple linear regression model and save this model as `reg01`.
- fit a multiple linear regression model that includes the variable you used in your simple linear regression and save this as `reg02`.

```
#Simple linear regression model = 'reg01'
reg01<-lm(Car.Insurance.Premiums....~Losses.incurred.by.insurance.companies.for.collisions.per.insured.

summary(reg01)
```

```
##
## Call:
## lm(formula = Car.Insurance.Premiums.... ~ Losses.incurred.by.insurance.companies.for.collisions.per.
##     data = bad_drivers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -213.33  -96.75  -40.11  112.24  379.97
##
## Coefficients:
```

```
##                                                                      Estimate
## (Intercept)                                                          285.3251
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....   4.4733
##                                                                      Std. Error
## (Intercept)                                                            109.6689
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....     0.8021
##                                                                      t value
## (Intercept)                                                            2.602
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....   5.577
##                                                                      Pr(>|t|)
## (Intercept)                                                            0.0122
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver.... 1.04e-06
##
## (Intercept)                                                          *
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver.... ***
## ---
## 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
#Multiple linear regression = 'reg02'
reg02<-lm(Car.Insurance.Premiums....~Losses.incurred.by.insurance.companies.for.collisions.per.insured.

summary(reg02)
```

```
##
## Call:
## lm(formula = Car.Insurance.Premiums.... ~ Losses.incurred.by.insurance.companies.for.collisions.per.
##     Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Had.Not.Been.Involved.In.Any.Previous.Acc
##     Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Were.Speeding,
##     data = bad_drivers)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -211.67  -91.90  -40.02   98.77  362.80
##
## Coefficients:
##
## (Intercept)
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Had.Not.Been.Involved.In.Any.Previous.Acciden
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Were.Speeding
##
## (Intercept)
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Had.Not.Been.Involved.In.Any.Previous.Acciden
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Were.Speeding
##
## (Intercept)
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Had.Not.Been.Involved.In.Any.Previous.Acciden
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Were.Speeding
##
```

```
## (Intercept)
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Had.Not.Been.Involved.In.Any.Previous.Accident
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Were.Speeding
##
## (Intercept)
## Losses.incurred.by.insurance.companies.for.collisions.per.insured.driver....
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Had.Not.Been.Involved.In.Any.Previous.Accident
## Percentage.Of.Drivers.Involved.In.Fatal.Collisions.Who.Were.Speeding
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 142.8 on 47 degrees of freedom
## Multiple R-squared:  0.3971, Adjusted R-squared:  0.3586
## F-statistic: 10.32 on 3 and 47 DF,  p-value: 2.474e-05
```

- Cross-validation
- **For both reg01 and reg02**
  - split your data into 5 cross-validation folds.

```
set.seed(13)

train_val_inds <- caret::createDataPartition(
  y=Car.Insurance.Premiums....,
  p= 0.8
)

train_val_inds
```

```
## $Resample1
##  [1]  1  2  3  4  5  6  7  8  9 10 12 13 14 15 16 17 18 19 21 22 24 25 26
## [24] 28 29 30 31 32 33 34 35 36 38 39 41 42 43 44 45 47 48 49 50
```

```
cars_train_val <- bad_drivers %>% slice (train_val_inds[[1]])
cars_test <- bad_drivers %>% slice (-train_val_inds[[1]])

num_crossval_folds <- 5
crossval_fold_inds <- caret::createFolds (
  y=cars_train_val$Car.Insurance.Premiums....,
  k= num_crossval_folds
)
```

```
* write a for loop that trains your model on 4 of the folds and evaluates on the "held-out" fold.  (This
* compute the MSE for each validation fold
* compute the MSE averaged across all 5 folds.
```

```
train_val_mse <- expand.grid(
  poly_degree = seq_len(7),
  val_fold_num = seq_len(num_crossval_folds),
  train_mse = NA,
  val_mse = NA
)

for(poly_degree in seq_len(7)) {
  for(val_fold_num in seq_len(num_crossval_folds)) {
    results_index <- which (
```

```
      train_val_mse$poly_degree == poly_degree &
        train_val_mse$val_fold_num == val_fold_num
    )
    cars_train <- cars_train_val %>% slice(-crossval_fold_inds[[val_fold_num]])
    cars_val <- cars_train_val %>% slice(crossval_fold_inds[[val_fold_num]])

    fit <- lm(Car.Insurance.Premiums.... ~ poly(Losses.incurred.by.insurance.companies.for.collisions.pe

    train_resids <- cars_train$Car.Insurance.Premiums.... - predict(fit)
    train_val_mse$train_mse[results_index] <- mean(train_resids^2)

    val_resids<-cars_val$Car.Insurance.Premiums....-predict(fit, cars_val)
    train_val_mse$val_mse[results_index] <-mean(val_resids^2)
  }
}
```

```
## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length
```

```
## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length
```

```
## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length

## Warning in cars_train$Car.Insurance.Premiums.... - predict(fit): longer
## object length is not a multiple of shorter object length
```

```
head(train_val_mse)
```

```
##   poly_degree val_fold_num train_mse  val_mse
## 1           1            1  45512.77 27121.94
## 2           2            1  45466.17 27205.10
## 3           3            1  45416.98 28406.08
## 4           4            1  47795.36 28230.73
## 5           5            1  48411.07 29299.00
## 6           6            1  48922.09 30016.96
```

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

summarized_crossval_mse_results
```

```
## # A tibble: 7 x 2
##   poly_degree crossval_mse
##         <int>        <dbl>
## 1           1       19432.
## 2           2       19457.
## 3           3       19612.
## 4           4       19694.
## 5           5       19465.
## 6           6       19215.
## 7           7       19088.
```