# Linear Regression with Medical Insurance

Johann Piedras

11/28/2020

# Contents

# Which Factors Influce the Price Of Health Insurance?

First and foremost, this RMD file is inspired and founded thanks to a Python Notebook by mariapushkareva on Kaggle. The purpose here is to showcase an example of linear regression with medical data, but also to parallel data analysis from Python using R.

With that out of the way lets, look at the insurance.csv file sourced from Kaggle. Although I don't know exactly where Kaggle got the csv from, the purpose derived from the data is is to attempt to understand what factors contribute to the cost of health insurance.

#### Factors In Health Insurance Cost

**Age** : The age of a beneficiary of health insurance.
**Sex** : Insurance companies use the binary of female/male
**BMI** : Short for Body Mass Index. $BMI = \frac{m}{h^2}$ where m is mass in kilograms, and h is height in meters.
**Children** : Number of dependents for the beneficiaries insurance plan.
**Smoker** : Whether the beneficiary smokes or not.
**Region** : The beneficiaries area within the US.
**Charges** : Amount they've been charged.

## EDA

```
# importing the data
df <- read.csv('insurance.csv', header = TRUE)
head(df)
```

```
##   age    sex    bmi children smoker    region   charges
## 1  19 female 27.900        0    yes southwest 16884.924
## 2  18   male 33.770        1     no southeast  1725.552
## 3  28   male 33.000        3     no southeast  4449.462
## 4  33   male 22.705        0     no northwest 21984.471
## 5  32   male 28.880        0     no northwest  3866.855
## 6  31 female 25.740        0     no southeast  3756.622
```

#### Dimension: dim() <– shape()

```
dim(df)
```

```
## [1] 1338    7
```

#### Summary Statistics: summary() <– describe()

```
summary(df)
```

```
##       age            sex           bmi           children     smoker
##  Min.   :18.00   female:662   Min.   :15.96   Min.   :0.000   no :1064
##  1st Qu.:27.00   male  :676   1st Qu.:26.30   1st Qu.:0.000   yes: 274
##  Median :39.00                Median :30.40   Median :1.000
##  Mean   :39.21                Mean   :30.66   Mean   :1.095
##  3rd Qu.:51.00                3rd Qu.:34.69   3rd Qu.:2.000
##  Max.   :64.00                Max.   :53.13   Max.   :5.000
##        region       charges
##  northeast:324   Min.   : 1122
##  northwest:325   1st Qu.: 4740
##  southeast:364   Median : 9382
##  southwest:325   Mean   :13270
##                  3rd Qu.:16640
##                  Max.   :63770
```

**Data Types: str() <– .types()**

```
str(df)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : Factor w/ 2 levels "female","male": 1 2 2 2 2 1 1 1 2 1 ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : Factor w/ 2 levels "no","yes": 2 1 1 1 1 1 1 1 1 1 ...
##  $ region  : Factor w/ 4 levels "northeast","northwest",..: 4 3 3 2 2 3 3 2 1 2 ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

Alternatively you can use the sapply function to find the datatypes of each column.

```
sapply(df,class)
```

```
##       age       sex       bmi  children    smoker    region   charges
## "integer"  "factor" "numeric" "integer"  "factor"  "factor" "numeric"
```

**Apply() family tangent** **apply()**: apply(Array, margin =(1 or 2), function...) -> For arrays. 1 is rows. 2 is columns. c(1,2) is columns and rows.
**lapply()**: lapply(Dataframe/List/Vectors, function...) -> Returns a list of the same size of the object inputted/
**sapply()**: sapply(Dataframe/List/Vectors, function...) -> Returns a vector, or a simplified version of the object class.

**Nulls in each column**

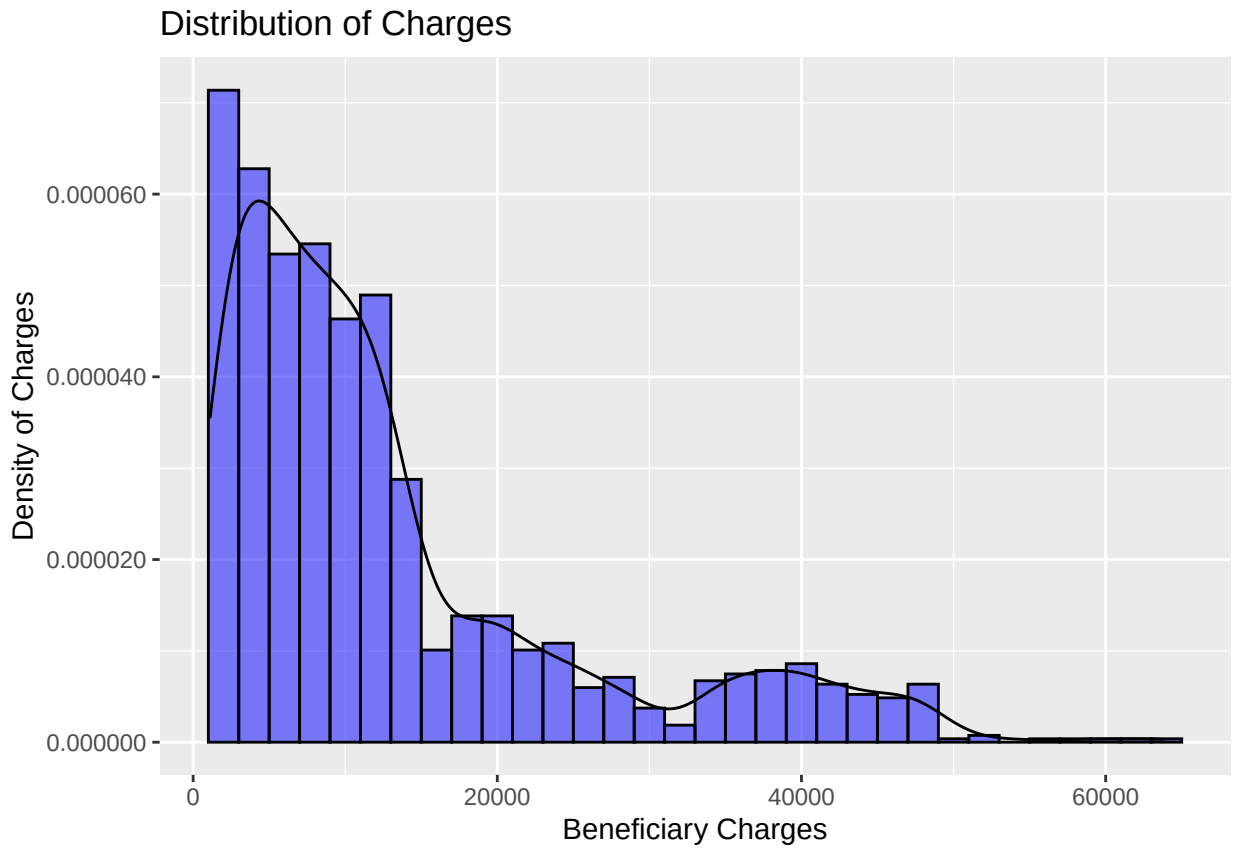
```
# Python would use... df.isnull().sum()
colSums(is.na(df))
```

```
##      age      sex      bmi children   smoker   region  charges
##        0        0        0        0        0        0        0
```

**Basic EDA report**

So after some basic review the first things that come to mind are: - No Null.
- Sex, Smoker, and Region are factors. We could potentially make dummies from these.
- There's a significant amount of non smokers compared to smokers.
- Each region has about 320 people.
- Charges range from 1k to 63k so there is a lot of variance there.

```
# Histogram overlaid with kernel density curve
ggplot(df, aes(x=charges)) +
    geom_histogram(aes(y=..density..),      # Histogram with density instead of count on y-axis
                   binwidth=2000,
                   colour="black", fill="blue",alpha=.5) +
    geom_density() + # Overlay with transparent density plot
    scale_y_continuous(name = "Density of Charges", labels = comma) +
    ggtitle("Distribution of Charges") + xlab("Beneficiary Charges")
```




```
# Histogram overlaid with kernel density curve
ggplot(df, aes(x=charges)) +
    geom_histogram(aes(y=..density..),      # Histogram with density instead of count on y-axis
                   #binwidth=2000,
                   colour="black", fill="blue",alpha=.5) +
    geom_density() + # Overlay with transparent density plot
    scale_x_log10()+
```

```
    scale_y_continuous(name = "Density of Charges", labels = comma) +
    ggtitle("Distribution of Charges") + xlab("Beneficiary Charges")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```