DATA_607_Project_2

Md Jalal Uddin

October 9, 2016

DATA Set 1:

```
# Load packages

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(tidyr)
library(stringr)
library(knitr)
library(ggplot2)
```

```
BPL_Data <- read.csv("C:/Users/sql_ent_svc/Google Drive/DATA_607/Project_2/BPL.csv", header = TRUE,  sti
head(BPL_Data)
```

```
##                  Teams Match Won Lost Tied N.R PTS    NRR  For Against
## 1   Comilla Victorea    10   7    3    0   0  14  0.788 1296    1231
## 2     Rangpur Riders    10   7    3    0   0  14  0.693 1233    1148
## 3      Barisal Bulls    10   7    3    0   0  14  0.063 1197    1147
## 4    Dhaka Dynamites    10   4    6    0   0   8 -0.010 1319    1321
## 5        Sylhet Super    10   3    7    0   0   6 -0.710 1190    1292
## 6 Chittagong Vikings    10   2    8    0   0   4 -0.828 1370    1466
```

```
#data Source:http://www.espncricinfo.com/bangladesh-premier-league-2015-16/engine/series/921139.html?vi
```

Rearrange the data with new column name using select statement

```
BPL_Data1<- BPL_Data %>%
select(Teams,Match, Won, PTS, For, Against) %>%
rename(BPL_Teams = Teams, Match_Played=Match, Match_Won = Won, Total_Points = PTS, Own_score = For, sco
BPL_Data1
```

```
##             BPL_Teams Match_Played Match_Won Total_Points Own_score
## 1   Comilla Victorea           10         7           14      1296
## 2     Rangpur Riders           10         7           14      1233
## 3      Barisal Bulls           10         7           14      1197
## 4    Dhaka Dynamites           10         4            8      1319
## 5        Sylhet Super           10         3            6      1190
## 6 Chittagong Vikings           10         2            4      1370
##   score_Against
## 1          1231
## 2          1148
## 3          1147
## 4          1321
## 5          1292
## 6          1466
```

By using select statement again I am selecting the only those variable whose we need for my analysis

```
BPL_Score_Points <- BPL_Data1%>%
select(BPL_Teams, Match_Won, Own_score, score_Against)
BPL_Score_Points
```

```
##             BPL_Teams Match_Won Own_score score_Against
## 1   Comilla Victorea         7      1296          1231
## 2     Rangpur Riders         7      1233          1148
## 3      Barisal Bulls         7      1197          1147
## 4    Dhaka Dynamites         4      1319          1321
## 5        Sylhet Super         3      1190          1292
## 6 Chittagong Vikings         2      1370          1466
```

Selecting only 3 variable

```
BPL_Score_Points1 <- BPL_Score_Points%>%
select(BPL_Teams, Own_score, score_Against)
BPL_Score_Points1
```

```
##             BPL_Teams Own_score score_Against
## 1   Comilla Victorea      1296          1231
## 2     Rangpur Riders      1233          1148
## 3      Barisal Bulls      1197          1147
## 4    Dhaka Dynamites      1319          1321
## 5        Sylhet Super      1190          1292
## 6 Chittagong Vikings      1370          1466
```

finding summary of BPL_Score_Points1 data

```
summary(BPL_Score_Points1)
```

```
##   BPL_Teams           Own_score    score_Against
##  Length:6           Min.   :1190   Min.   :1147
##  Class :character   1st Qu.:1206   1st Qu.:1169
##  Mode  :character   Median :1264   Median :1262
##                     Mean   :1268   Mean   :1268
##                     3rd Qu.:1313   3rd Qu.:1314
##                     Max.   :1370   Max.   :1466
```

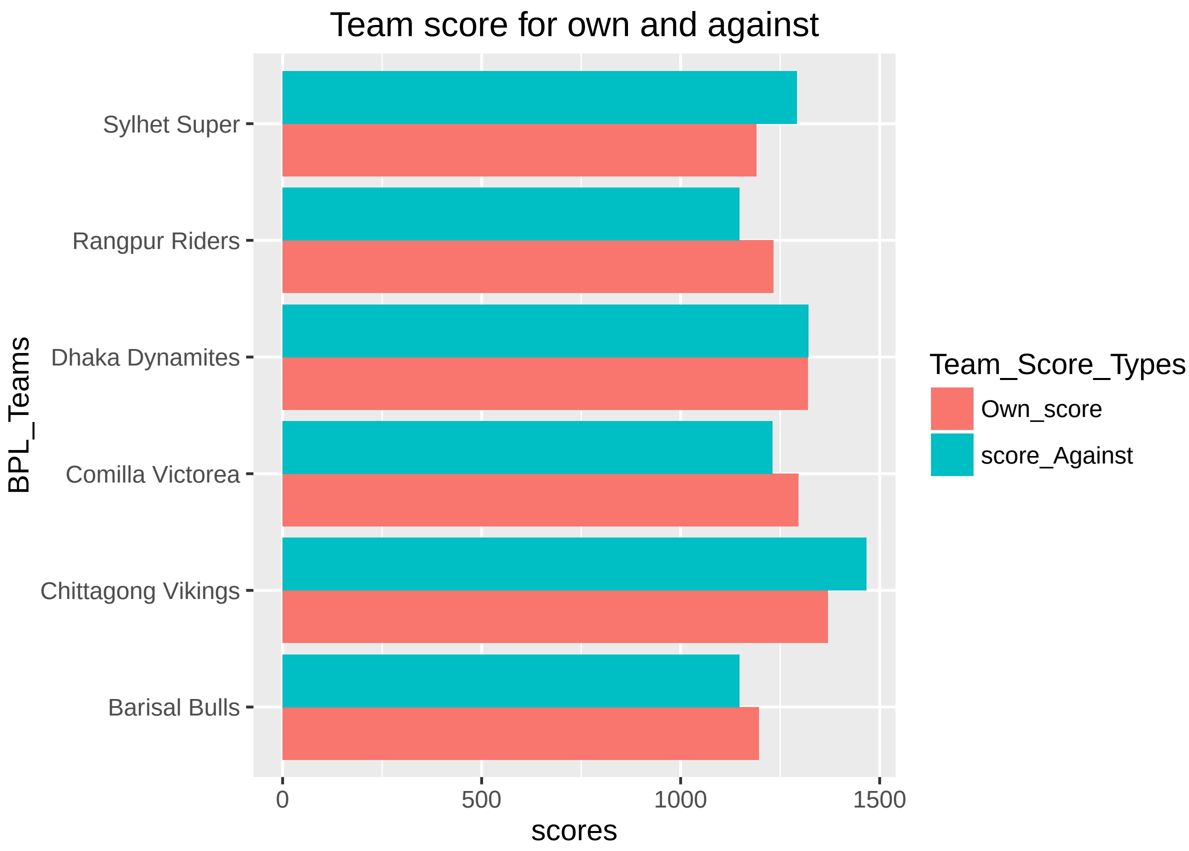
Rearrange data using gather statement

```
BPL_Score_Points3 <- gather(BPL_Score_Points1, "Team_Score_Types","Scores", 2:3)
BPL_Score_Points3
```

```
##              BPL_Teams Team_Score_Types Scores
## 1     Comilla Victorea        Own_score   1296
## 2       Rangpur Riders        Own_score   1233
## 3        Barisal Bulls        Own_score   1197
## 4      Dhaka Dynamites        Own_score   1319
## 5         Sylhet Super        Own_score   1190
## 6   Chittagong Vikings        Own_score   1370
## 7     Comilla Victorea    score_Against   1231
## 8       Rangpur Riders    score_Against   1148
## 9        Barisal Bulls    score_Against   1147
## 10     Dhaka Dynamites    score_Against   1321
## 11        Sylhet Super    score_Against   1292
## 12 Chittagong Vikings    score_Against   1466
```

Creating a geomatric bar plot by using ggplot

```
ggplot(data = BPL_Score_Points3, aes(x = BPL_Teams, y = Scores, fill = Team_Score_Types))+ geom_bar(sta
```



```
#source: http://docs.ggplot2.org/0.9.3.1/geom_bar.html
```

DATA Set 2:

```
Animal_Data <- read.csv("C:/Users/sql_ent_svc/Google Drive/DATA_607/Project_2/Animal_Sleep.csv", header
head(Animal_Data)
```

```
##                  Species   BodyWt BrainWt NonDreaming Dreaming TotalSleep
## 1        Africanelephant 6654.000  5712.0          NA       NA        3.3
## 2 Africangiantpouchedrat    1.000     6.6         6.3      2.0        8.3
## 3              ArcticFox    3.385    44.5          NA       NA       12.5
## 4   Arcticgroundsquirrel    0.920     5.7          NA       NA       16.5
## 5          Asianelephant 2547.000  4603.0         2.1      1.8        3.9
## 6                Baboon   10.550   179.5         9.1      0.7        9.8
##   LifeSpan Gestation Predation Exposure Danger
## 1     38.6       645         3        5      3
## 2      4.5        42         3        1      3
## 3     14.0        60         1        1      1
## 4       NA        25         5        2      3
## 5     69.0       624         3        5      4
## 6     27.0       180         4        4      4
```

```
#data Source:http://www.statsci.org/data/general/sleep.txt
```

Rearrange the data by using select statement

```
Animal_Data1 <- Animal_Data%>%
select(Species, BodyWt, BrainWt, TotalSleep,LifeSpan, Danger)
head(Animal_Data1)
```

```
##                  Species   BodyWt BrainWt TotalSleep LifeSpan Danger
## 1        Africanelephant 6654.000  5712.0        3.3     38.6      3
## 2 Africangiantpouchedrat    1.000     6.6        8.3      4.5      3
## 3              ArcticFox    3.385    44.5       12.5     14.0      1
## 4   Arcticgroundsquirrel    0.920     5.7       16.5       NA      3
## 5          Asianelephant 2547.000  4603.0        3.9     69.0      4
## 6                Baboon   10.550   179.5        9.8     27.0      4
```

we can find mean of a specific variable or column by using followoing mean function

```
mean(Animal_Data1$BodyWt)
```

```
## [1] 198.79
```

Finding correlation of different variables

```
(Cor.BodyWt_BrainWt <- cor.test( Animal_Data1$BodyWt,Animal_Data1$BrainWt))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Animal_Data1$BodyWt and Animal_Data1$BrainWt
## t = 20.278, df = 60, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8926796 0.9599518
## sample estimates:
##       cor
## 0.9341638
```

```
(Cor.BodyWt_LifeSpan <- cor.test( Animal_Data1$BodyWt,Animal_Data1$LifeSpan))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Animal_Data1$BodyWt and Animal_Data1$LifeSpan
## t = 2.3745, df = 56, p-value = 0.02102
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.04789655 0.52011413
## sample estimates:
##       cor
## 0.3024506
```

```
(Cor.TotalSleep_BrainWt <- cor.test( Animal_Data1$TotalSleep,Animal_Data1$BrainWt))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Animal_Data1$TotalSleep and Animal_Data1$BrainWt
## t = -2.8701, df = 56, p-value = 0.00578
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.5642105 -0.1099789
## sample estimates:
##       cor
## -0.358102
```

```
(Cor.LifeSpan_BrainWt <- cor.test( Animal_Data1$LifeSpan,Animal_Data1$BrainWt))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Animal_Data1$LifeSpan and Animal_Data1$BrainWt
## t = 4.4281, df = 56, p-value = 4.457e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2889670 0.6783233
## sample estimates:
##       cor
## 0.5092527
```

Here we can see variables Bodywt and BrainWt have very strong positive relationship by seeing the correlation coefficient between them.

Finding Regression by using following statement

```
Reg_BodyWt_BrainWt <- lm(BodyWt~BrainWt+LifeSpan+Danger, data=Animal_Data1)
summary(Reg_BodyWt_BrainWt)
```

```
##
## Call:
## lm(formula = BodyWt ~ BrainWt + LifeSpan + Danger, data = Animal_Data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1462.41   -86.28   -18.02    68.08  1147.80
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 148.65532   86.78234   1.713   0.0925 .
## BrainWt       1.02005    0.04553  22.406  < 2e-16 ***
## LifeSpan    -11.93203    2.37887  -5.016 6.06e-06 ***
## Danger       -2.81257   25.59548  -0.110   0.9129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 281.4 on 54 degrees of freedom
##   (4 observations deleted due to missingness)
## Multiple R-squared:  0.913,  Adjusted R-squared:  0.9082
## F-statistic: 188.9 on 3 and 54 DF,  p-value: < 2.2e-16
```

Rearrange data selecting only 3 variable for my analysis

```
Animal_Data2 <- Animal_Data1%>%
select(Species, BodyWt, BrainWt)
head(Animal_Data2)
```

```
##                  Species   BodyWt BrainWt
## 1        Africanelephant 6654.000  5712.0
## 2 Africangiantpouchedrat    1.000     6.6
## 3              ArcticFox    3.385    44.5
## 4   Arcticgroundsquirrel    0.920     5.7
## 5          Asianelephant 2547.000  4603.0
## 6                 Baboon   10.550   179.5
```

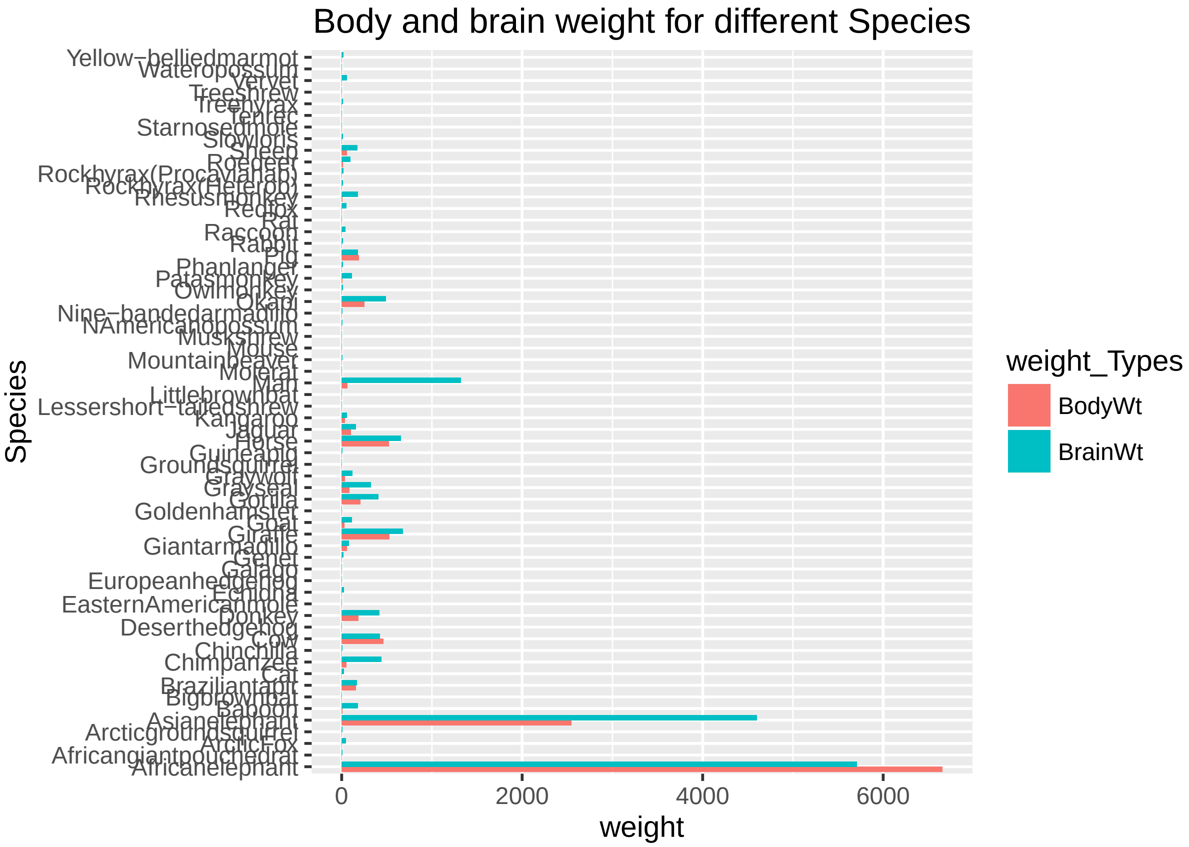
rearange data using gather function

```
Animal_Data3 <- gather(Animal_Data2, "weight_Types","weight", 2:3)
head(Animal_Data3)
```

```
##                  Species weight_Types   weight
## 1        Africanelephant       BodyWt 6654.000
## 2 Africangiantpouchedrat       BodyWt    1.000
## 3              ArcticFox       BodyWt    3.385
## 4   Arcticgroundsquirrel       BodyWt    0.920
## 5          Asianelephant       BodyWt 2547.000
## 6                 Baboon       BodyWt   10.550
```

finding the geomatric bar plot using ggplot

```
ggplot(data = Animal_Data3, aes(x = Species, y = weight, fill = weight_Types))+ geom_bar(stat="identity
```



DATA SEt 3:

```
Pima_Indian_Data <- read.csv("C:/Users/sql_ent_svc/Google Drive/DATA_607/Project_2/Pima_Indian_diabetes_
head(Pima_Indian_Data)
```

```
##   V1  V2 V3 V4  V5   V6    V7 V8 V9
## 1  6 148 72 35   0 33.6 0.627 50  1
## 2  1  85 66 29   0 26.6 0.351 31  0
## 3  8 183 64  0   0 23.3 0.672 32  1
## 4  1  89 66 23  94 28.1 0.167 21  0
## 5  0 137 40 35 168 43.1 2.288 33  1
## 6  5 116 74  0   0 25.6 0.201 30  0
```

```
#data Source: http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes
```

Rearrange the data set by changing the variable name/column name by folloing statement

```
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V1"] <- "Number_of_Times_Pregnant"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V2"] <- "Plasma_glucose"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V3"] <- "Diastolic_Pressure"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V4"] <- "Triceps_skin_thickness"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V5"] <- "Serum_insulin"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V6"] <- "BMI"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V7"] <- "Pedigree_function"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V8"] <- "Age"
names(Pima_Indian_Data)[names(Pima_Indian_Data) == "V9"] <- "Class_variable"

head(Pima_Indian_Data)
```

```
##   Number_of_Times_Pregnant Plasma_glucose Diastolic_Pressure
## 1                        6            148                 72
## 2                        1             85                 66
## 3                        8            183                 64
## 4                        1             89                 66
## 5                        0            137                 40
## 6                        5            116                 74
##   Triceps_skin_thickness Serum_insulin  BMI Pedigree_function Age
## 1                     35             0 33.6             0.627  50
## 2                     29             0 26.6             0.351  31
## 3                      0             0 23.3             0.672  32
## 4                     23            94 28.1             0.167  21
## 5                     35           168 43.1             2.288  33
## 6                      0             0 25.6             0.201  30
##   Class_variable
## 1              1
## 2              0
## 3              1
## 4              0
## 5              1
## 6              0
```

Arranging the data to see who has hightest BMI

```
Pima_Indian_Data1 <- Pima_Indian_Data %>%
           select(Number_of_Times_Pregnant,Plasma_glucose, Diastolic_Pressure, BMI, Age)

head(Pima_Indian_Data1)
```

```
##   Number_of_Times_Pregnant Plasma_glucose Diastolic_Pressure  BMI Age
## 1                        6            148                 72 33.6  50
## 2                        1             85                 66 26.6  31
## 3                        8            183                 64 23.3  32
## 4                        1             89                 66 28.1  21
## 5                        0            137                 40 43.1  33
## 6                        5            116                 74 25.6  30
```

Finding the correlation between different variables

```
(Cor.BMI_Age <- cor.test( Pima_Indian_Data1$BMI,Pima_Indian_Data1$Age))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Pima_Indian_Data1$BMI and Pima_Indian_Data1$Age
## t = 1.0037, df = 766, p-value = 0.3158
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.03459109  0.10671254
## sample estimates:
##        cor
## 0.03624187
```

```
(Cor.BMI_Diastolic_Pressure <- cor.test( Pima_Indian_Data1$BMI,Pima_Indian_Data1$Diastolic_Pressure))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Pima_Indian_Data1$BMI and Pima_Indian_Data1$Diastolic_Pressure
## t = 8.1289, df = 766, p-value = 1.738e-15
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2153543 0.3456585
## sample estimates:
##       cor
## 0.2818053
```

```
(Cor.Age_Diastolic_Pressure <- cor.test( Pima_Indian_Data1$Age,Pima_Indian_Data1$Diastolic_Pressure))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Pima_Indian_Data1$Age and Pima_Indian_Data1$Diastolic_Pressure
## t = 6.8281, df = 766, p-value = 1.752e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1716931 0.3051022
## sample estimates:
##       cor
## 0.2395279
```

```
(Cor.Age_Diastolic_Plasma_glucose <- cor.test( Pima_Indian_Data1$Plasma_glucose ,Pima_Indian_Data1$Dias
```

```
##
##  Pearson's product-moment correlation
##
## data:  Pima_Indian_Data1$Plasma_glucose and Pima_Indian_Data1$Diastolic_Pressure
## t = 4.2732, df = 766, p-value = 2.17e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  0.08273846 0.22094875
## sample estimates:
##       cor
## 0.1525896
```

```
(Cor.BMI_Plasma_glucose <- cor.test( Pima_Indian_Data1$BMI,Pima_Indian_Data1$Plasma_glucose))
```

```
##
##  Pearson's product-moment correlation
##
## data:  Pima_Indian_Data1$BMI and Pima_Indian_Data1$Plasma_glucose
## t = 6.2737, df = 766, p-value = 5.891e-10
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1527152 0.2873218
## sample estimates:
##       cor
## 0.2210711
```

finding Regression

```
Reg_Plasma_glucose <- lm(Plasma_glucose~Diastolic_Pressure+Age+BMI, data=Pima_Indian_Data1)
summary(Reg_Plasma_glucose)
```

```
##
## Call:
## lm(formula = Plasma_glucose ~ Diastolic_Pressure + Age + BMI,
##     data = Pima_Indian_Data1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -131.784  -19.112   -2.026   18.401   84.459
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        68.18492    5.80137  11.753  < 2e-16 ***
## Diastolic_Pressure  0.06018    0.06033   0.998    0.319
## Age                 0.67280    0.09534   7.057 3.82e-12 ***
## BMI                 0.81850    0.14390   5.688 1.83e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.13 on 764 degrees of freedom
## Multiple R-squared:  0.1154, Adjusted R-squared:  0.1119
## F-statistic: 33.22 on 3 and 764 DF,  p-value: < 2.2e-16
```

Selecting only 3 variables for my analysis

```
Pima_Indian_Data2 <- Pima_Indian_Data1%>%
select(Number_of_Times_Pregnant, Diastolic_Pressure, BMI)
head(Pima_Indian_Data2)
```

```
##   Number_of_Times_Pregnant Diastolic_Pressure  BMI
## 1                        6                 72 33.6
## 2                        1                 66 26.6
## 3                        8                 64 23.3
## 4                        1                 66 28.1
## 5                        0                 40 43.1
## 6                        5                 74 25.6
```

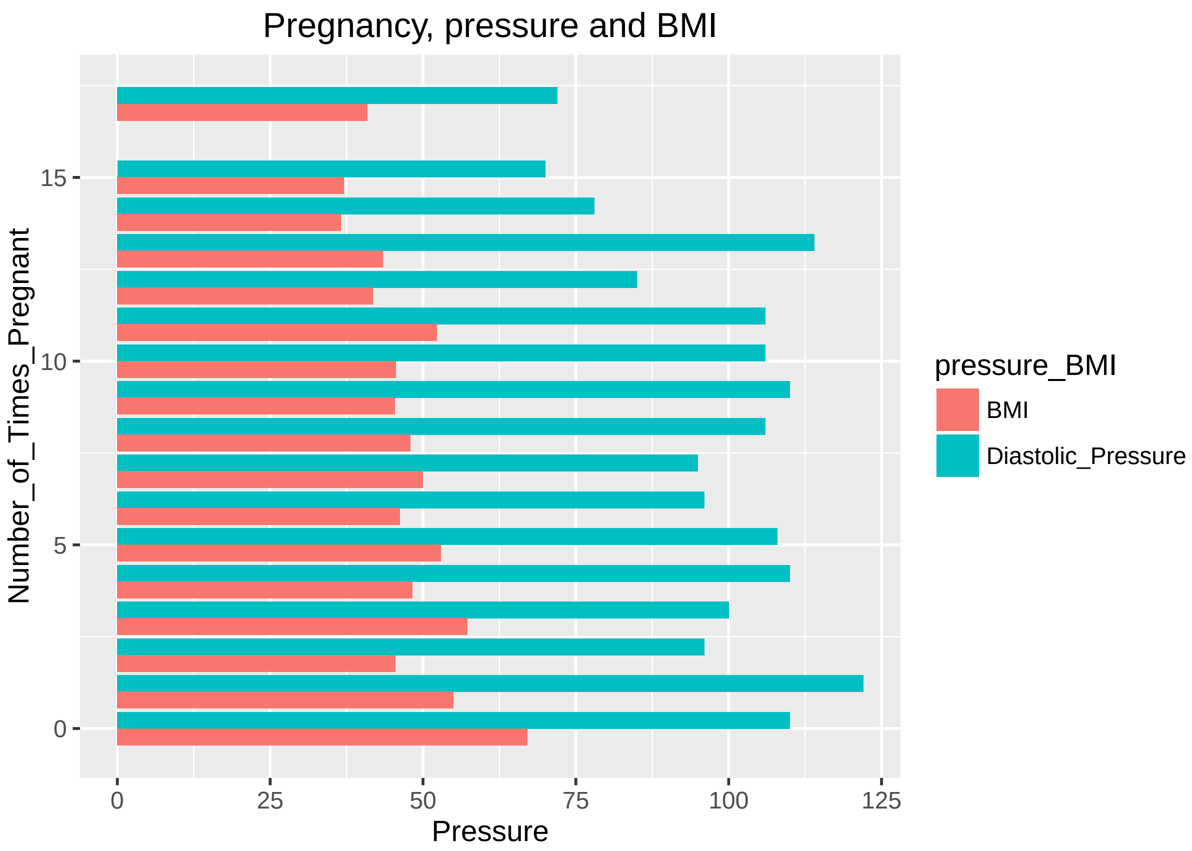
rearrange data using gather function

```
Pima_Indian_Data3 <- gather(Pima_Indian_Data2, "pressure_BMI","pressure", 2:3)
head(Pima_Indian_Data3)
```

```
##   Number_of_Times_Pregnant       pressure_BMI pressure
## 1                        6 Diastolic_Pressure       72
## 2                        1 Diastolic_Pressure       66
## 3                        8 Diastolic_Pressure       64
## 4                        1 Diastolic_Pressure       66
## 5                        0 Diastolic_Pressure       40
## 6                        5 Diastolic_Pressure       74
```

finding geomaric bar plot using ggplot

```
ggplot(data = Pima_Indian_Data3, aes(x = Number_of_Times_Pregnant, y = pressure, fill =  pressure_BMI))
```


Pregnancy, pressure and BMI
Number_of_Times_Pregnant
Pressure
pressure_BMI
BMI
Diastolic_Pressure