Suicide Rates Project

Nick Wawee

10/16/2020

Libraries

```
rm(list=ls())
library(car)
library(ggplot2)
library(ggrepel)
library(reshape2)
library(ggpubr)
library(dplyr)
library(limma)
library(MASS)
knitr::opts_chunk$set(echo = TRUE)
options(width = 60)
plot_opts =   theme_bw()+
  theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_
```

Functions

```
plotdists = function(df, path){
  plotlist = list()
  for (col in colnames(df)){
    x = df[,col]
    if (class(x)=='factor'){
      dfnew = data.frame(col = factor(), count = integer())
      for (level in levels(x)){
        count = length(which(x==level))
        dfnew = rbind(data.frame(col= level, count = count), dfnew)
      }
      dfnew$col <- factor(dfnew$col, levels = dfnew$col[order(dfnew$count)])
      p = ggplot(dfnew, aes(x=col, y=count))+
        geom_bar(stat= 'identity')+
        plot_opts+
        labs(x = col, y = 'Count', title = paste(col, "Distribution"))+
        geom_text(aes(label = count), vjust = -0.3)
      plotlist[[col]] = p
    }else if (class(x) == 'numeric' | class(x) == 'integer'){
      dfnew =data.frame(col = class(x))
      histinfo = hist(x = x , breaks='Scott', plot = F)
      p = ggplot(as.data.frame(x), aes(x=x))+
        geom_histogram(bins = length(histinfo$breaks))+plot_opts+
        #geom_density(aes(y=..count..), size = 2)+plot_opts+
```

```
        geom_vline(aes(xintercept = median(x)),
             linetype = "dashed", size = 0.6)+
        labs(x = col, y = 'Count', title = paste(col, 'Distribution'))

      plotlist[[col]] = p
    }
  }
  pfinal =ggarrange(plotlist = plotlist)
  ggsave(path, pfinal, height=2.5*length(plotlist) , width=2.5*length(plotlist), units="in", limitsize =
  return(pfinal)
}

Assumption_Check = function(l_m, outp){
  df = l_m[['model']]
  df$residuals = l_m[["residuals"]]
  df = df[,-which(colnames(df)==l_m[["call"]][["formula"]][[2]])]
  #Assumption 1 - linearity check
  a1pls = list()#assumption 1 plotlist
  for (col in colnames(df)[-which(colnames(df)=='residuals')]){
    dfnew = data.frame(x = df[,col], y = df$residuals)
    a1pls[[col]] = ggplot(data = dfnew, aes(x =x, y=y))+
        geom_jitter()+plot_opts+labs(x = col, y = 'Residuals')+
      theme(axis.text.x = element_text(angle = 45, hjust = 1))+
      geom_hline(yintercept=0,linetype="dashed",color = "red", size=1)
  }
  p1 = ggarrange(plotlist= a1pls)
  p1 = annotate_figure(p1,text_grob("Linearity Checks", color = "red", face = "bold", size = 20))
  ggsave(filename = paste(outp,'linearity.png',sep=""), plot = p1, dpi = 600)

  #Assumption 2 - Bias and Scedasticity
  df2 = data.frame(Fit = l_m$fitted.values, Residuals = l_m$residuals)
  p2 = ggplot(data = df2, aes(x = Fit, y = Residuals))+
    geom_point()+plot_opts+geom_hline(yintercept=0,linetype="dashed",color = "red", size=1)+labs(title =
  ggsave(filename = paste(outp,'bias_sced.png',sep=""), plot = p2, dpi = 600)

  #Assumption 3 - Correlation in Errors
  a3pls = list()
  for (n in colnames(df[,-which(colnames(df)=='residuals')])){
    dfnew = data.frame(y = df$residuals[order(df[,n])], x = 1:nrow(df))
    a3pls[[n]] = ggplot(data = dfnew, aes(x = x, y = y))+geom_jitter()+
      plot_opts+labs(title = paste('Sorted By:', n), x = 'Index', y='Residuals')
  }
  p3 = ggarrange(plotlist = a3pls)
  p3 = annotate_figure(p3,text_grob("Error Independence Check", color = "red", face = "bold", size = 20
  ggsave(filename = paste(outp,'inderror.png',sep=""), plot = p3, dpi = 600)

  #Assumption 4 - Normality of Residuals
  shapres = shapiro.test(l_m$residuals)
  p4 = ggplot(df)+
    geom_qq(aes(sample = residuals))+geom_qq_line(aes(sample= residuals))+
    plot_opts+ labs(title = paste('Normality of Residuals\n', 'Shaprio Wilks Results: W = ', as.charact
  ggsave(filename = paste(outp,'normres.png',sep=""), plot = p4, dpi = 600)
```

```
  pfinal = ggarrange(plotlist = list(p1, p2, p3, p4))
  pfinal= annotate_figure(pfinal,text_grob("Model Assumption Check", face = "bold", size = 26))
  ggsave(filename = paste(outp,'all_assum.png',sep=""), plot = pfinal, dpi = 600, width = 8, height = 1
  return(pfinal)
}

infl_analysis = function(l_m, df){
  k = length(l_m$coefficients)-1
  n = nrow(df)
  row_num = 1:n
  #response_v = df[colnames(df)==l_m$terms[[2]]] #use if you would like to change the labels of the poi
  #Leverage points
  hatdf = data.frame(Values = hatvalues(l_m), Row_Num = row_num, Type = rep('Hat Values', length(row_nu
  hatdf$Label = NA
  inds = which(hatvalues(l_m)>2*(k+1)/n)
  if(length(inds)!= 0){hatdf$Label[inds] = row_num[inds]}
  #Outliers
  instdf = data.frame(Values = rstandard(l_m), Row_Num = row_num, Type = rep('Internally Standardized R
  instdf$Label = NA
  inds = which(rstandard(l_m) > 3 | rstandard(l_m) < -3)
  if(length(inds)!=0){instdf$Label[inds] = row_num[inds]}

  extdf = data.frame(Values = rstudent(l_m), Row_Num = row_num, Type = rep('Externally Standardized Resi
  extdf$Label = NA
  inds = which(rstudent(l_m) > 3 | rstudent(l_m) < -3)
  if(length(inds)!=0){extdf$Label[inds] = row_num[inds]}

  #Influential
  dfitsdf = data.frame(Values = dffits(l_m), Row_Num = row_num, Type = rep('DEFFITS', length(row_num)),
  dfitsdf$Label = NA
  inds = which(dffits(l_m) > 2*sqrt((k+2)/(n-k-2)) | dffits(l_m) < -2*sqrt((k+2)/(n-k-2)))
  if(length(inds)!=0){dfitsdf$Label[inds] = row_num[inds]}

  cddf = data.frame(Values = cooks.distance(l_m), Row_Num = row_num, Type = rep("Cook's Distance", leng
  cddf$Label = NA
  inds = cooks.distance(l_m) > 1
  if(length(inds)!=0){cddf$Label[inds] = row_num[inds]}

  cvdf = data.frame(Values = covratio(l_m), Row_Num = row_num, Type = rep("Covariance Ratio", length(ro
  cvdf$Label = NA
  inds = covratio(l_m) > 1 + 3*(k+1)/n | covratio(l_m) < 1 - 3*(k+1)/n
  if(length(inds)!=0){cvdf$Label[inds] = row_num[inds]}

  ret_df = rbind(hatdf, instdf, extdf, dfitsdf, cddf, cvdf)
  return(ret_df)
}
```

Loading

```
df = read.csv("../Data/master.csv", stringsAsFactors = T)
str(df)
```

```
## 'data.frame':    27820 obs. of  12 variables:
```

```
##  $ country           : Factor w/ 101 levels "Albania","Antigua and Barbuda",..: 1 1 1 1 1 1 1 1 1 1
##  $ year              : int  1987 1987 1987 1987 1987 1987 1987 1987 1987 1987 ...
##  $ sex               : Factor w/ 2 levels "female","male": 2 2 1 2 2 1 1 1 2 1 ...
##  $ age               : Factor w/ 6 levels "15-24 years",..: 1 3 1 6 2 6 3 2 5 4 ...
##  $ suicides_no       : int  21 16 14 1 9 1 6 4 1 0 ...
##  $ population        : int  312900 308000 289700 21800 274300 35600 278800 257200 137500 311000 ...
##  $ suicides.100k.pop : num  6.71 5.19 4.83 4.59 3.28 2.81 2.15 1.56 0.73 0 ...
##  $ country.year      : Factor w/ 2321 levels "Albania1987",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ HDI.for.year      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ gdp_for_year....  : Factor w/ 2321 levels "1,002,219,052,968",..: 727 727 727 727 727 727 727 727
##  $ gdp_per_capita....: int  796 796 796 796 796 796 796 796 796 796 ...
##  $ generation        : Factor w/ 6 levels "Boomers","G.I. Generation",..: 3 6 3 2 1 2 6 1 2 3 ...
```

Cleaning

Below will examine the na values in each column:

```
for (col in colnames(df)){
  print(length(which(is.na(df[,col]))))
}
```

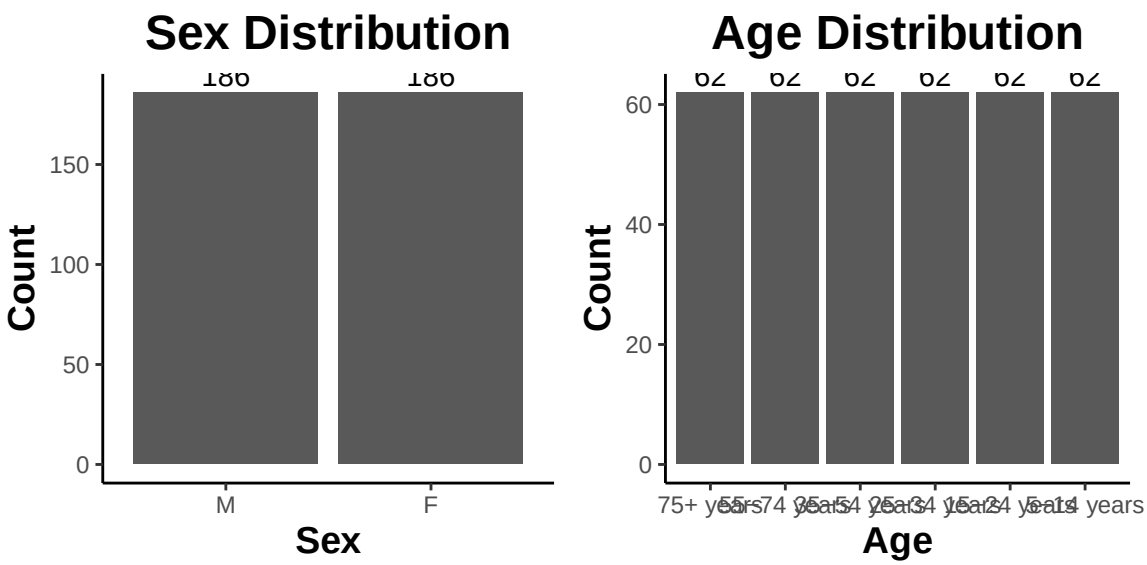
```
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 0
## [1] 19456
## [1] 0
## [1] 0
## [1] 0
```

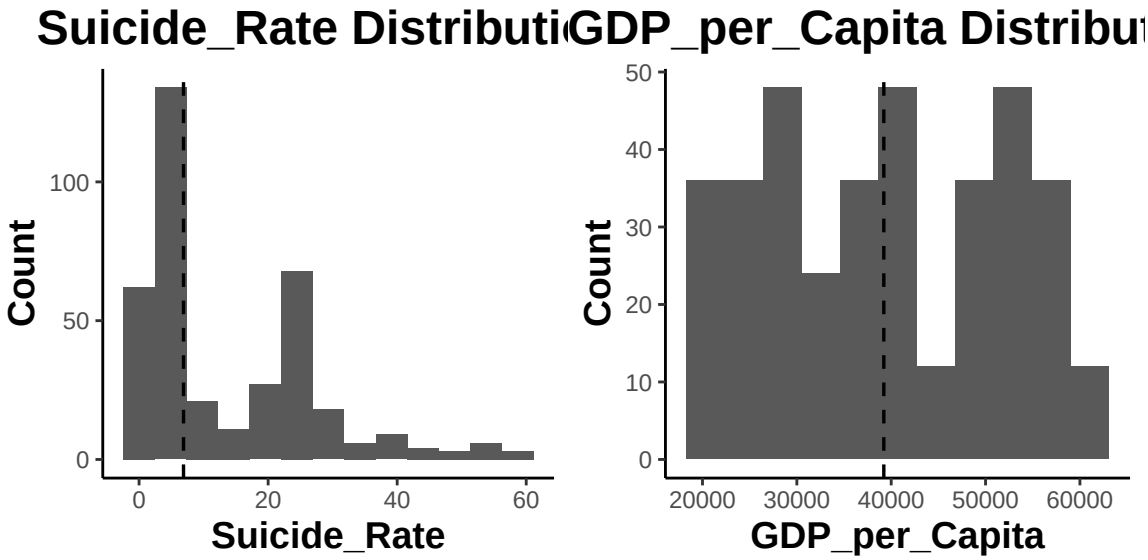
It appears that the HDI for year column has the majority of NAs and will be discluded from the remaining of the analysis. The data will be filtered for the United States only.

```
## 'data.frame':    372 obs. of  6 variables:
##  $ year              : int  1985 1985 1985 1985 1985 1985 1985 1985 1985 1985 ...
##  $ sex               : Factor w/ 2 levels "female","male": 2 2 2 2 2 1 1 1 1 1 ...
##  $ age               : Factor w/ 6 levels "15-24 years",..: 6 5 2 3 1 3 5 6 2 1 ...
##  $ suicides.100k.pop : num  53.6 29.5 24.5 22.8 21.4 ...
##  $ gdp_for_year....  : num  4.35e+12 4.35e+12 4.35e+12 4.35e+12 4.35e+12 ...
##  $ gdp_per_capita....: int  19693 19693 19693 19693 19693 19693 19693 19693 19693 19693 ...
```

Distributions

```
colnames(df) = c('Sex', 'Age', 'Suicide_Rate', 'GDP_per_Capita')
df$Age <- relevel(df$Age, ref = '5-14 years')
df$Sex = factor(paste(toupper(strsplit2(df$Sex, split ="")[,1])))
path = "../Plots/distplots.png"
plotdists(df, path)
```





Model Fitting

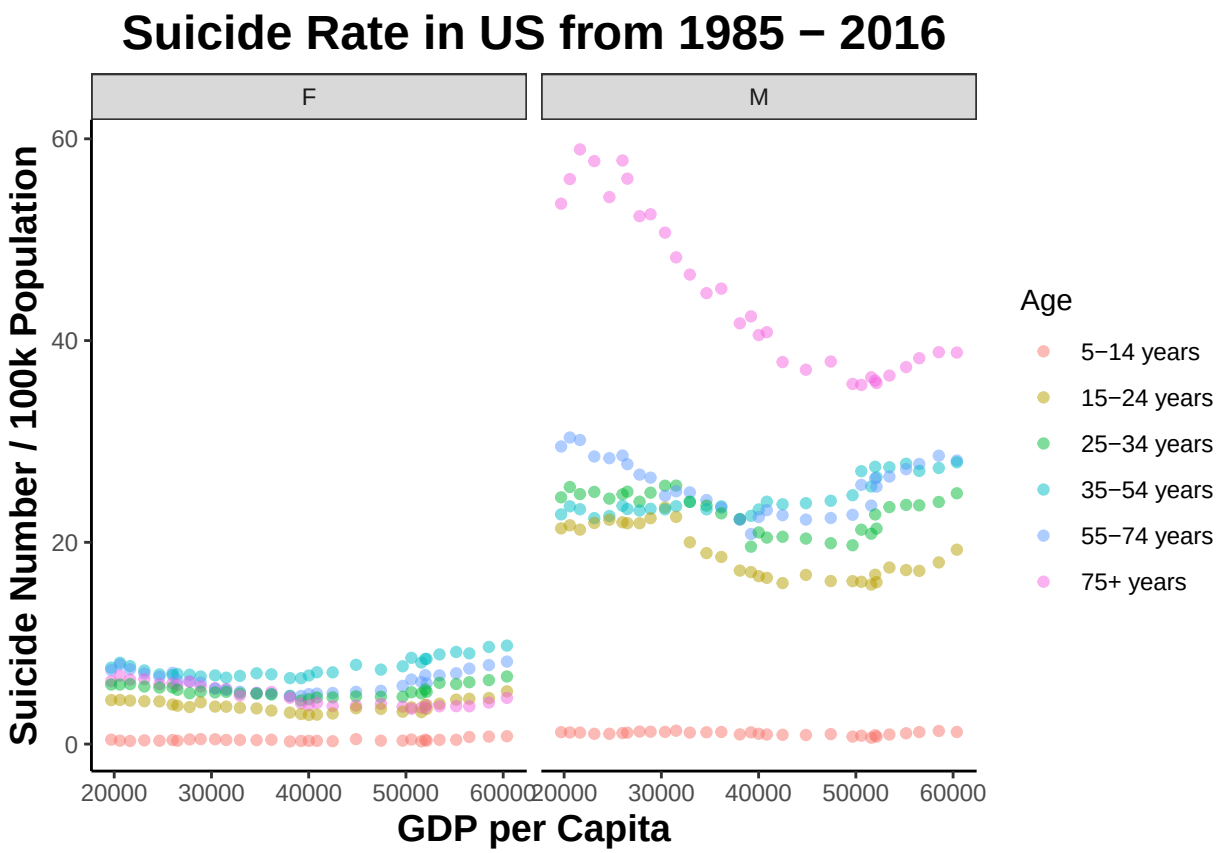
```
mlr2 = lm(Suicide_Rate ~ Sex*Age + GDP_per_Capita, data = df)
summary(mlr2)
```

```
##
## Call:
## lm(formula = Suicide_Rate ~ Sex * Age + GDP_per_Capita, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2498 -1.0883 -0.2779  0.9381 13.2023
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.989e+00  6.393e-01   4.675 4.16e-06
## SexM                 6.410e-01  6.653e-01   0.963    0.336
```

```
## Age15-24 years       3.361e+00  6.653e-01   5.052 6.96e-07
## Age25-34 years       4.889e+00  6.653e-01   7.349 1.36e-12
## Age35-54 years       7.200e+00  6.653e-01  10.822  < 2e-16
## Age55-74 years       5.767e+00  6.653e-01   8.669  < 2e-16
## Age75+ years         4.363e+00  6.653e-01   6.558 1.90e-10
## GDP_per_Capita      -6.560e-05  1.102e-05  -5.950 6.37e-09
## SexM:Age15-24 years  1.450e+01  9.409e-01  15.414  < 2e-16
## SexM:Age25-34 years  1.710e+01  9.409e-01  18.174  < 2e-16
## SexM:Age35-54 years  1.621e+01  9.409e-01  17.234  < 2e-16
## SexM:Age55-74 years  1.889e+01  9.409e-01  20.072  < 2e-16
## SexM:Age75+ years    3.917e+01  9.409e-01  41.636  < 2e-16
##
## (Intercept)         ***
## SexM
## Age15-24 years      ***
## Age25-34 years      ***
## Age35-54 years      ***
## Age55-74 years      ***
## Age75+ years        ***
## GDP_per_Capita      ***
## SexM:Age15-24 years ***
## SexM:Age25-34 years ***
## SexM:Age35-54 years ***
## SexM:Age55-74 years ***
## SexM:Age75+ years   ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.619 on 359 degrees of freedom
## Multiple R-squared:  0.9621, Adjusted R-squared:  0.9608
## F-statistic: 758.9 on 12 and 359 DF,  p-value: < 2.2e-16
```

Scatter Plot

```
outp = "../Plots/"
p = ggplot(data = df, aes(x = GDP_per_Capita, y = Suicide_Rate))+
  geom_point(aes(color = Age), alpha = 0.5)+plot_opts+facet_wrap(~Sex)+
  labs(x = 'GDP per Capita', y = 'Suicide Number / 100k Population', title = 'Suicide Rate in US from 19
p
```



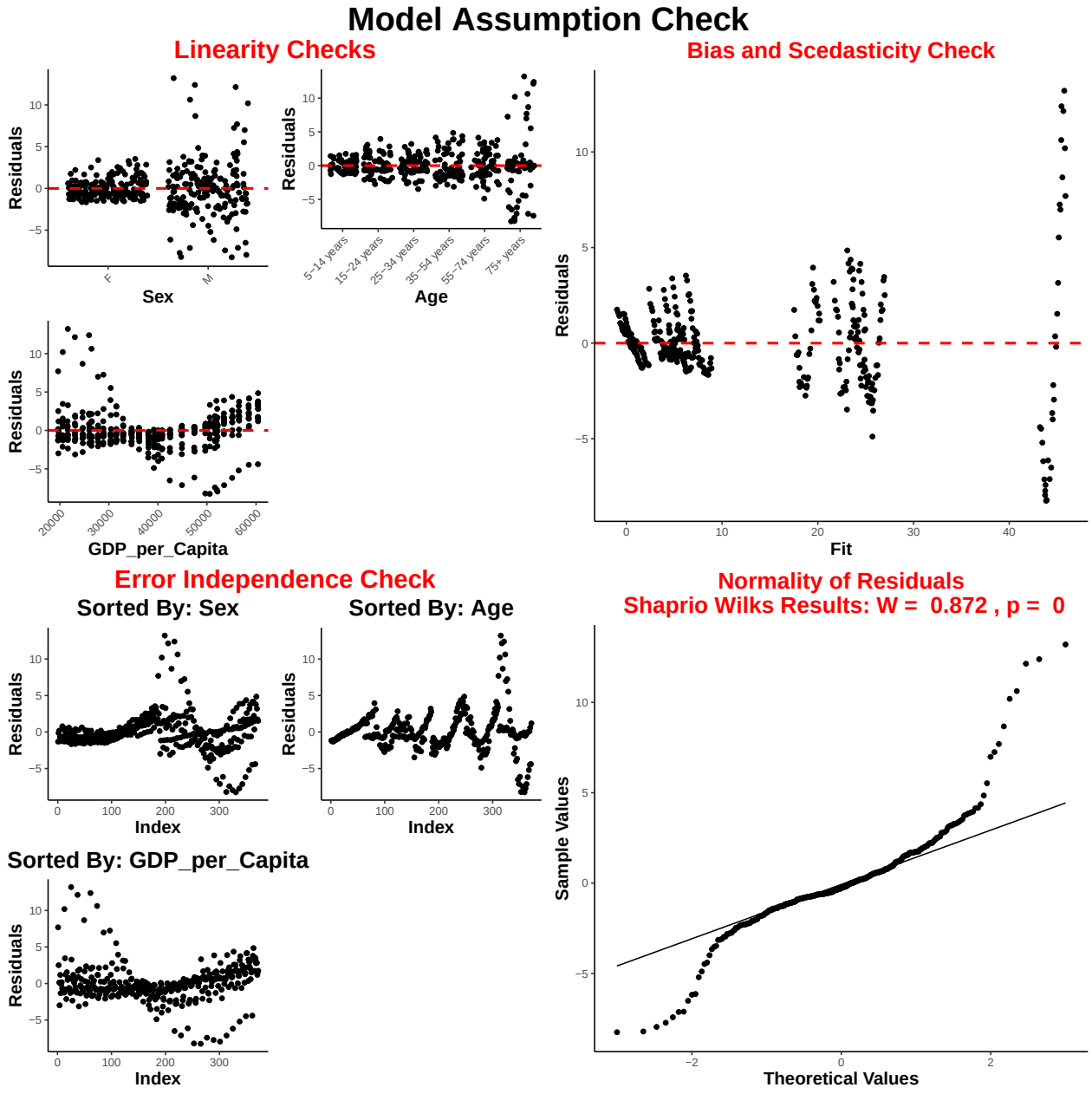

```
ggsave(filename = paste(outp,'scatter.png',sep=""), plot = p, dpi = 600, width = 8, height = 4, units =
```

Assumption Check 1

```
pf = Assumption_Check(mlr2, outp)
```

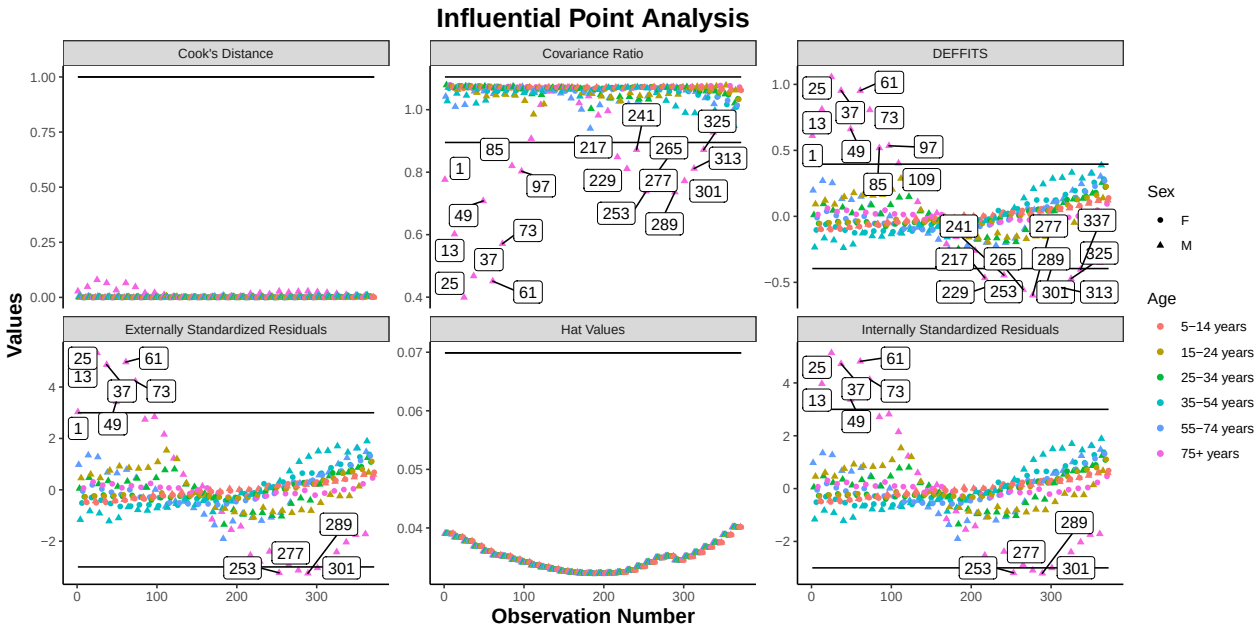
```
## Saving 12 x 12 in image
## Saving 12 x 12 in image
## Saving 12 x 12 in image
## Saving 12 x 12 in image
```

```
pf
```



Influential Point Analysis

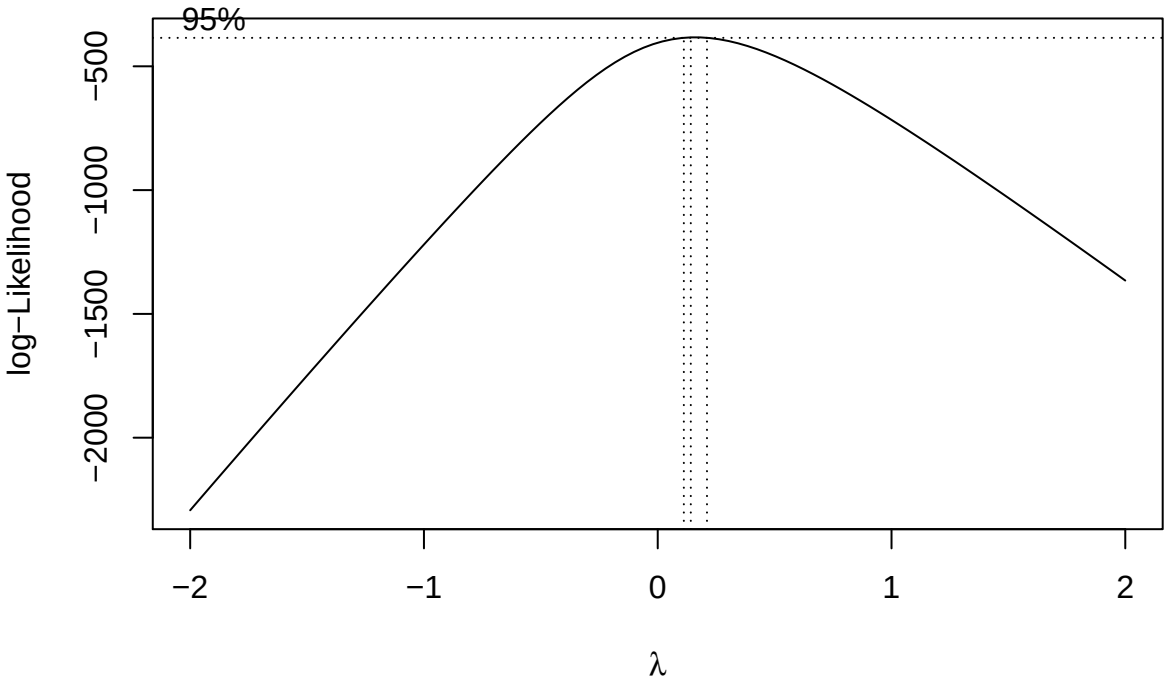
```
ret_df = infl_analysis(mlr2, df =df)
ret_df = cbind(ret_df, df)
p = ggplot(data= ret_df, aes(x= Row_Num, y = Values))+
  geom_point(aes(color = Age, shape = Sex))+
  facet_wrap(~Type, scales = "free_y")+plot_opts+geom_line(aes(y=Bound1))+geom_line(aes(y=Bound2))+
  geom_label_repel(aes(label=Label))+
  labs(title = 'Influential Point Analysis', x = 'Observation Number')
p
```




```
ggsave(filename = paste(outp,'influential.png',sep=""), plot = p, dpi = 600, width = 12, height = 6, uni
```

BoxCox Transformation

```
bc = boxcox(mlr2, data = df)
```




```
p = bc$x[which.max(bc$y)]
df$Suicide_Rate = df$Suicide_Rate**p
```

Transformed Response Variable

```
##
```

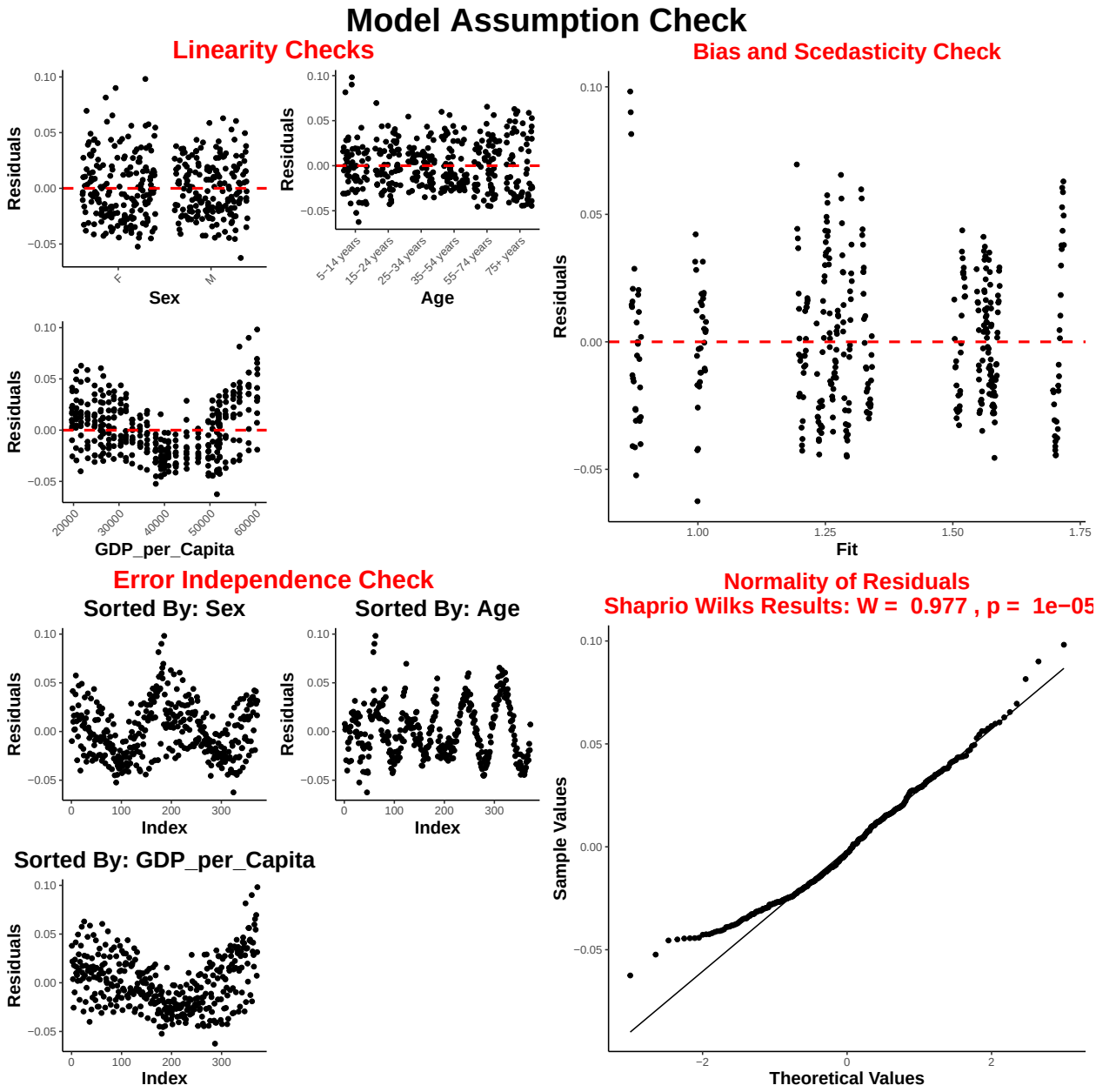
```
## Call:
## lm(formula = Suicide_Rate ~ Sex * Age + GDP_per_Capita, data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.062511 -0.021512 -0.002907  0.018208  0.098173
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)          8.987e-01  6.750e-03 133.140  < 2e-16
## SexM                 1.274e-01  7.024e-03  18.134  < 2e-16
## Age15-24 years       3.267e-01  7.024e-03  46.515  < 2e-16
## Age25-34 years       3.868e-01  7.024e-03  55.068  < 2e-16
## Age35-54 years       4.530e-01  7.024e-03  64.494  < 2e-16
## Age55-74 years       4.133e-01  7.024e-03  58.842  < 2e-16
## Age75+ years         3.656e-01  7.024e-03  52.046  < 2e-16
## GDP_per_Capita      -5.207e-07  1.164e-07  -4.473 1.04e-05
## SexM:Age15-24 years  1.815e-01  9.934e-03  18.271  < 2e-16
## SexM:Age25-34 years  1.652e-01  9.934e-03  16.633  < 2e-16
## SexM:Age35-54 years  1.125e-01  9.934e-03  11.324  < 2e-16
## SexM:Age55-74 years  1.627e-01  9.934e-03  16.377  < 2e-16
## SexM:Age75+ years    3.364e-01  9.934e-03  33.868  < 2e-16
##
## (Intercept)         ***
## SexM                ***
## Age15-24 years      ***
## Age25-34 years      ***
## Age35-54 years      ***
## Age55-74 years      ***
## Age75+ years        ***
## GDP_per_Capita      ***
## SexM:Age15-24 years ***
## SexM:Age25-34 years ***
## SexM:Age35-54 years ***
## SexM:Age55-74 years ***
## SexM:Age75+ years   ***
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02766 on 359 degrees of freedom
## Multiple R-squared:  0.9872, Adjusted R-squared:  0.9868
## F-statistic:  2314 on 12 and 359 DF,  p-value: < 2.2e-16
```

Assumption Check 2

```
outp = "../Plots/trans_"
pf = Assumption_Check(mlr3, outp)
```

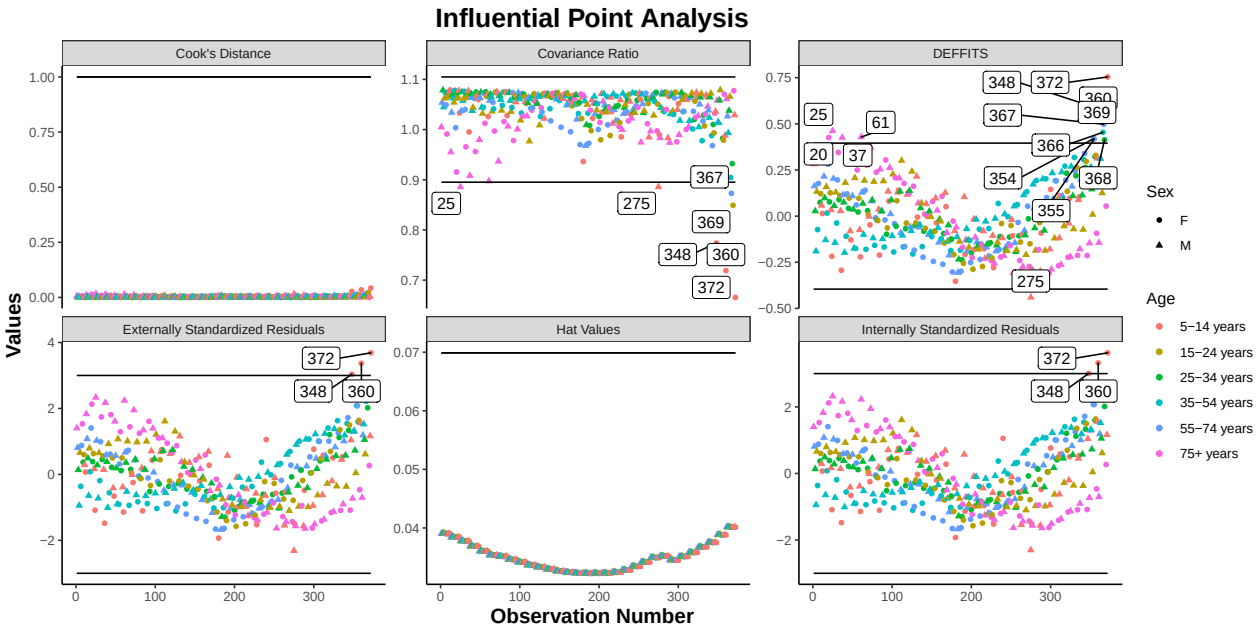
```
## Saving 12 x 12 in image
## Saving 12 x 12 in image
## Saving 12 x 12 in image
## Saving 12 x 12 in image
```

```
pf
```



Influential Point Analysis 2

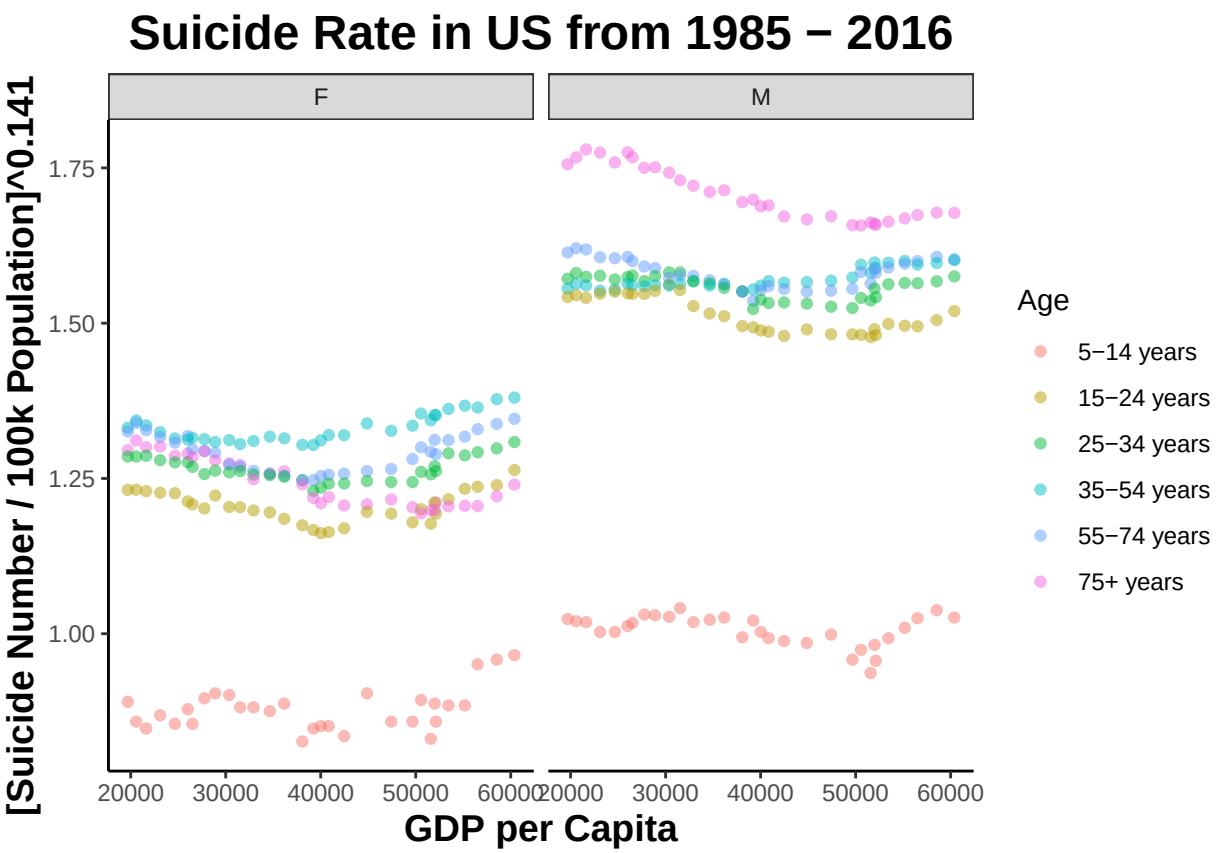
```
ret_df = infl_analysis(mlr3, df =df)
ret_df = cbind(ret_df, df)
p = ggplot(data= ret_df, aes(x= Row_Num, y = Values))+
  geom_point(aes(color = Age, shape = Sex))+
  facet_wrap(~Type, scales = "free_y")+plot_opts+geom_line(aes(y=Bound1))+geom_line(aes(y=Bound2))+
  geom_label_repel(aes(label=Label))+
  labs(title = 'Influential Point Analysis', x = 'Observation Number')
p
```




```
ggsave(filename = paste(outp,'influential.png',sep=""), plot = p, dpi = 600, width = 12, height = 6, uni
```

Transformed Response Plot

```
p = ggplot(data = df, aes(x = GDP_per_Capita, y = Suicide_Rate))+
  geom_point(aes(color = Age), alpha = 0.5)+plot_opts+facet_wrap(~Sex)+
  labs(x = 'GDP per Capita', y = '[Suicide Number / 100k Population]^0.141', title = 'Suicide Rate in US
p
```




```
ggsave(filename = paste(outp,'scatter.png',sep=""), plot = p, dpi = 600, width = 10, height = 5, units =
```