Final Project: Predicting Lifetime Expectancy

Group 4

2022-12-08

#Data Loading

```
#load necessary libraries
library(tidyverse) #readr, dplyr

#Visual
library(ggplot2)
library(corrplot)


#Analysis
library(leaps) #regsubsets
library(car) #boxcox
library(caret) #confusion matrix

#load processed dataset
df <- read_csv("df.csv")


#change outcome to categorical
df <- df %>%
  mutate(Life_Expectancy = case_when(
  Lifetime < 72  ~ 'Below Average',
  Lifetime < 100  ~ 'Above Average')) %>%
  mutate_at("Life_Expectancy", as.factor)

#delete innecessary variables
df <- df %>%
  select(-contains("Death")) %>% #not analyzing any of the cause of death factor this time
  rename("Depression"="Prevalence - Depressive disorders - Sex: Both - Age: Age-standardized (Percent)"
  select(-contains("Prevalence")) %>% #excluding all mental illness term except for depression
  rename("Diet_Animal"="Calories from animal protein (FAO (2017))") %>%
  rename("Diet_Plant"="Calories from plant protein (FAO (2017))") %>%
  select(-contains("Calories")) %>%
  select(-Population,-Continent)

#Change "Above Average" to the target level, since we are interested in longevity
df$Life_Expectancy <- factor(df$Life_Expectancy, levels = c("Below Average","Above Average"))
contrasts(df$Life_Expectancy)
```

```
##                 Above Average
## Below Average             0
## Above Average             1
```

#Feature Selection

```
#get the names of crucial features of Sequential, Forward and Backward Selection

#exclude two categorical variable, and year variable, which is predetermined to be included in the data
seqrep <-
  regsubsets(Life_Expectancy~Inventory+Exports+Consumption+Government+Household+Imports, data = df %>% s
                     nvmax = 3, method = "seqrep") %>%
  summary()
seqrep
```

```
## Subset selection object
## Call: regsubsets.formula(Life_Expectancy ~ Inventory + Exports + Consumption +
##     Government + Household + Imports, data = df %>% select(-1,
##     -2), nvmax = 3, method = "seqrep")
## 6 Variables  (and intercept)
##             Forced in Forced out
## Inventory       FALSE      FALSE
## Exports         FALSE      FALSE
## Consumption     FALSE      FALSE
## Government      FALSE      FALSE
## Household       FALSE      FALSE
## Imports         FALSE      FALSE
## 1 subsets of each size up to 3
## Selection Algorithm: 'sequential replacement'
##          Inventory Exports Consumption Government Household Imports
## 1  ( 1 ) " "       " "     " "         " "        "*"       " "
## 2  ( 1 ) " "       "*"     " "         " "        "*"       " "
## 3  ( 1 ) " "       "*"     " "         " "        "*"       "*"
```

```
#forward selection
forward <-
  regsubsets(Life_Expectancy~Inventory+Exports+Consumption+Government+Household+Imports, data = df %>% s
                      nvmax = 3, method = "forward") %>%
  summary()
forward
```

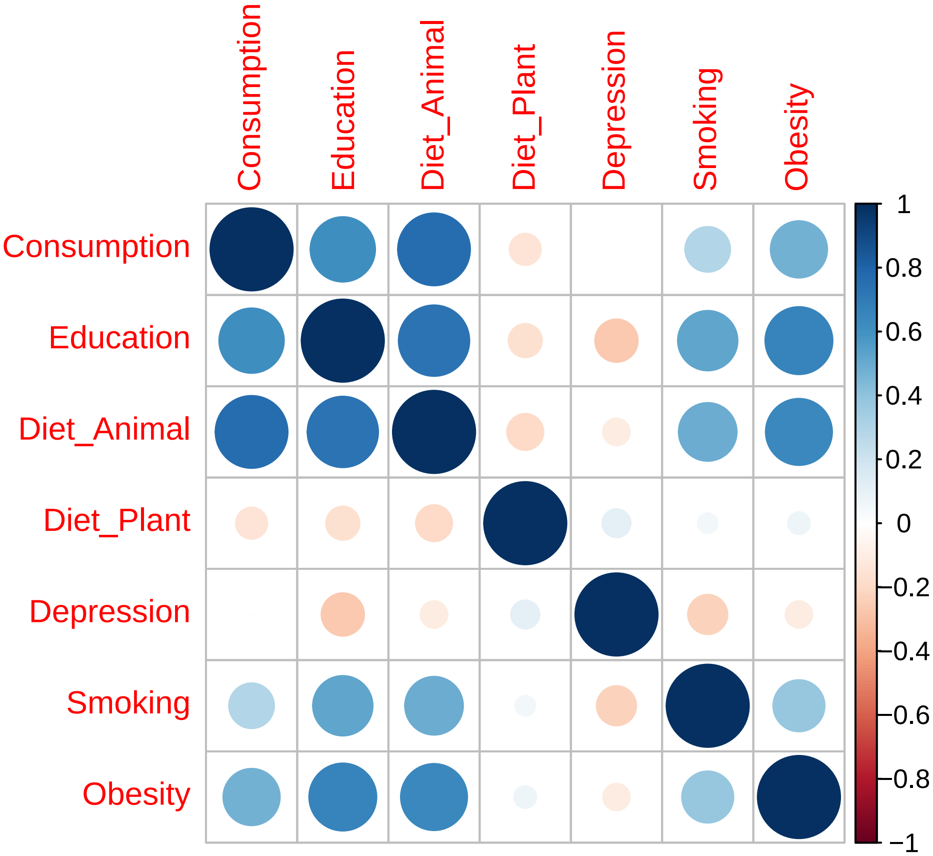
```
## Subset selection object
## Call: regsubsets.formula(Life_Expectancy ~ Inventory + Exports + Consumption +
##     Government + Household + Imports, data = df %>% select(-1,
##     -2), nvmax = 3, method = "forward")
## 6 Variables  (and intercept)
##             Forced in Forced out
## Inventory       FALSE      FALSE
## Exports         FALSE      FALSE
## Consumption     FALSE      FALSE
## Government      FALSE      FALSE
## Household       FALSE      FALSE
## Imports         FALSE      FALSE
## 1 subsets of each size up to 3
## Selection Algorithm: forward
##          Inventory Exports Consumption Government Household Imports
## 1  ( 1 ) " "       " "     " "         " "        "*"       " "    
## 2  ( 1 ) " "       "*"     " "         " "        "*"       " "    
## 3  ( 1 ) " "       "*"     " "         " "        "*"       "*"    
```

```
#backward selection
backward <-
  regsubsets(Life_Expectancy~Inventory+Exports+Consumption+Government+Household+Imports, data = df %>% s
                      nvmax = 3, method = "backward") %>%
  summary()

backward
```

```
## Subset selection object
## Call: regsubsets.formula(Life_Expectancy ~ Inventory + Exports + Consumption +
##     Government + Household + Imports, data = df %>% select(-1,
##     -2), nvmax = 3, method = "backward")
## 6 Variables  (and intercept)
##             Forced in Forced out
## Inventory       FALSE      FALSE
## Exports         FALSE      FALSE
## Consumption     FALSE      FALSE
## Government      FALSE      FALSE
## Household       FALSE      FALSE
## Imports         FALSE      FALSE
## 1 subsets of each size up to 3
## Selection Algorithm: backward
##          Inventory Exports Consumption Government Household Imports
## 1  ( 1 ) " "       " "     "*"         " "        " "       " "    
## 2  ( 1 ) " "       " "     "*"         "*"        " "       " "    
## 3  ( 1 ) " "       "*"     "*"         "*"        " "       " "    
```

#Confusion Matrix

```
df %>% select(Consumption, Education, Diet_Animal, Diet_Plant, Depression, Smoking, Obesity) %>%
  cor() %>%
  corrplot()
```



```
chisq.test(df$Education,df$Consumption) #testing some high correlation predictors, turns out good! no w
```

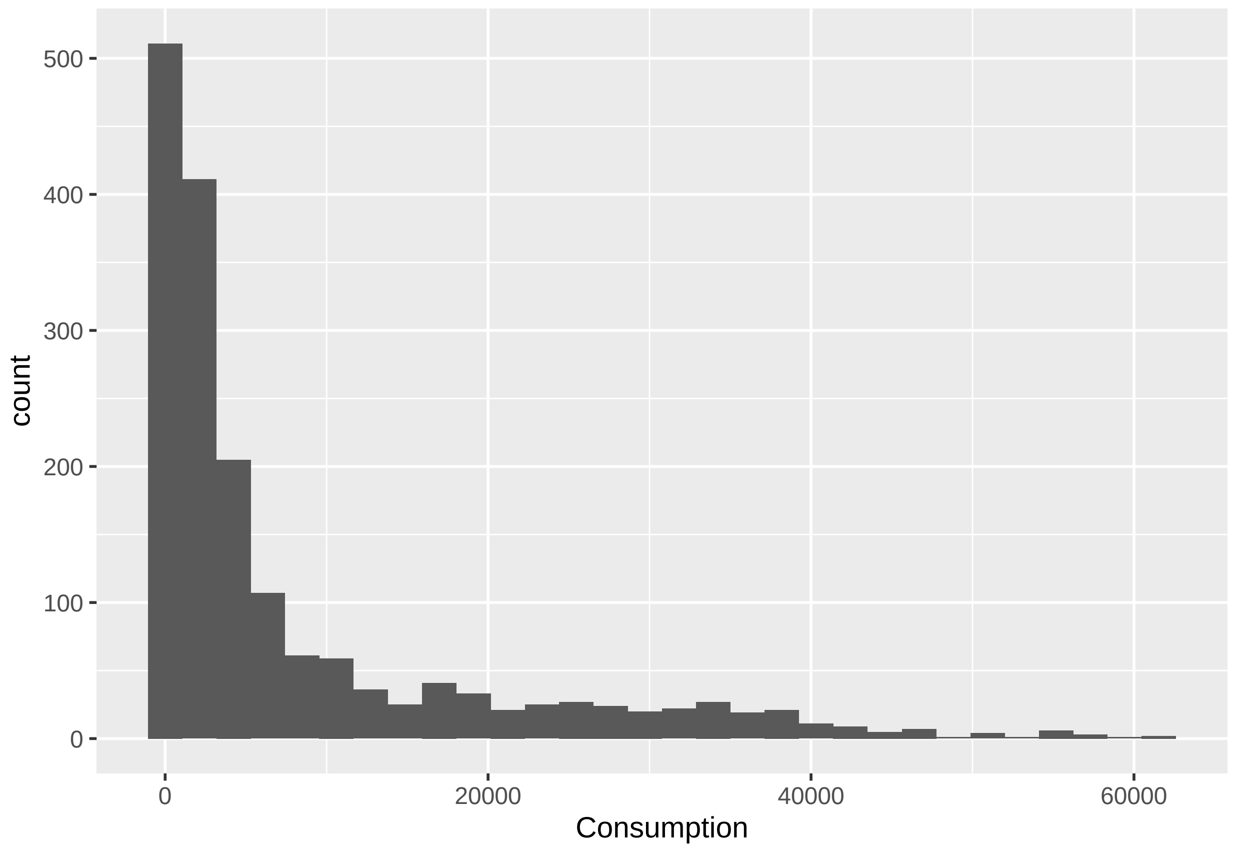
```
##
##  Pearson's Chi-squared test
##
## data:  df$Education and df$Consumption
## X-squared = 223360, df = 223232, p-value = 0.4237
```

follow the result above, we delete gdp terms except consumption, also delete diet_animal due to confusion matrix

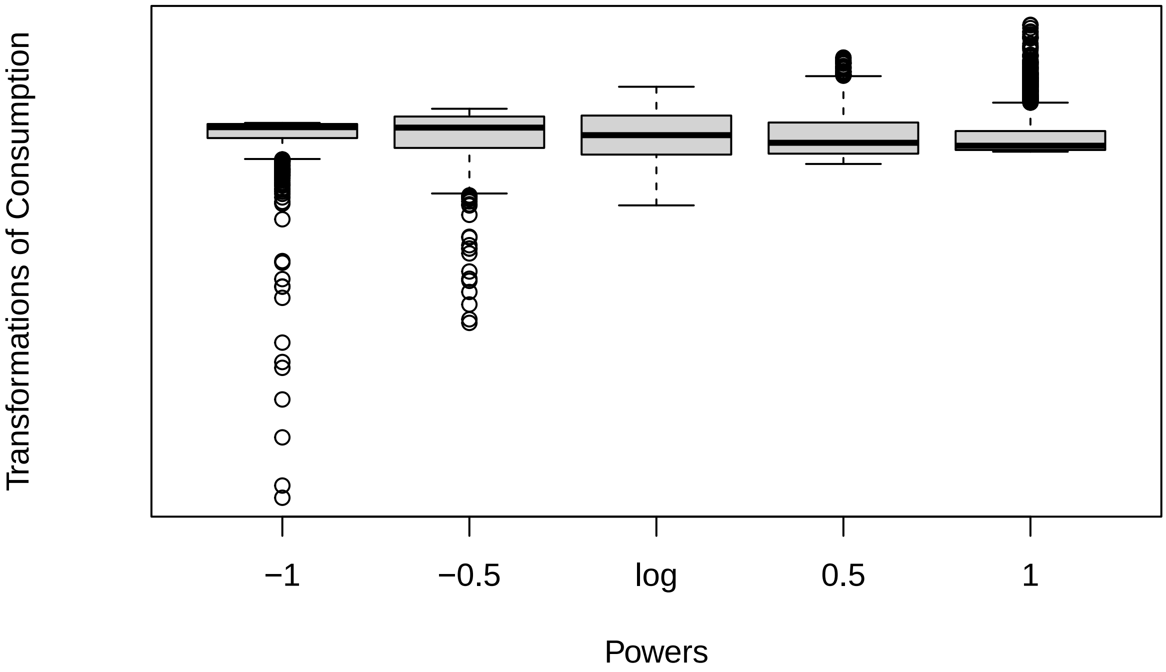
```
df <- df %>%
  select(-c(Inventory,Exports,Government,Household,Imports,Diet_Animal))
```

#EDA symbox

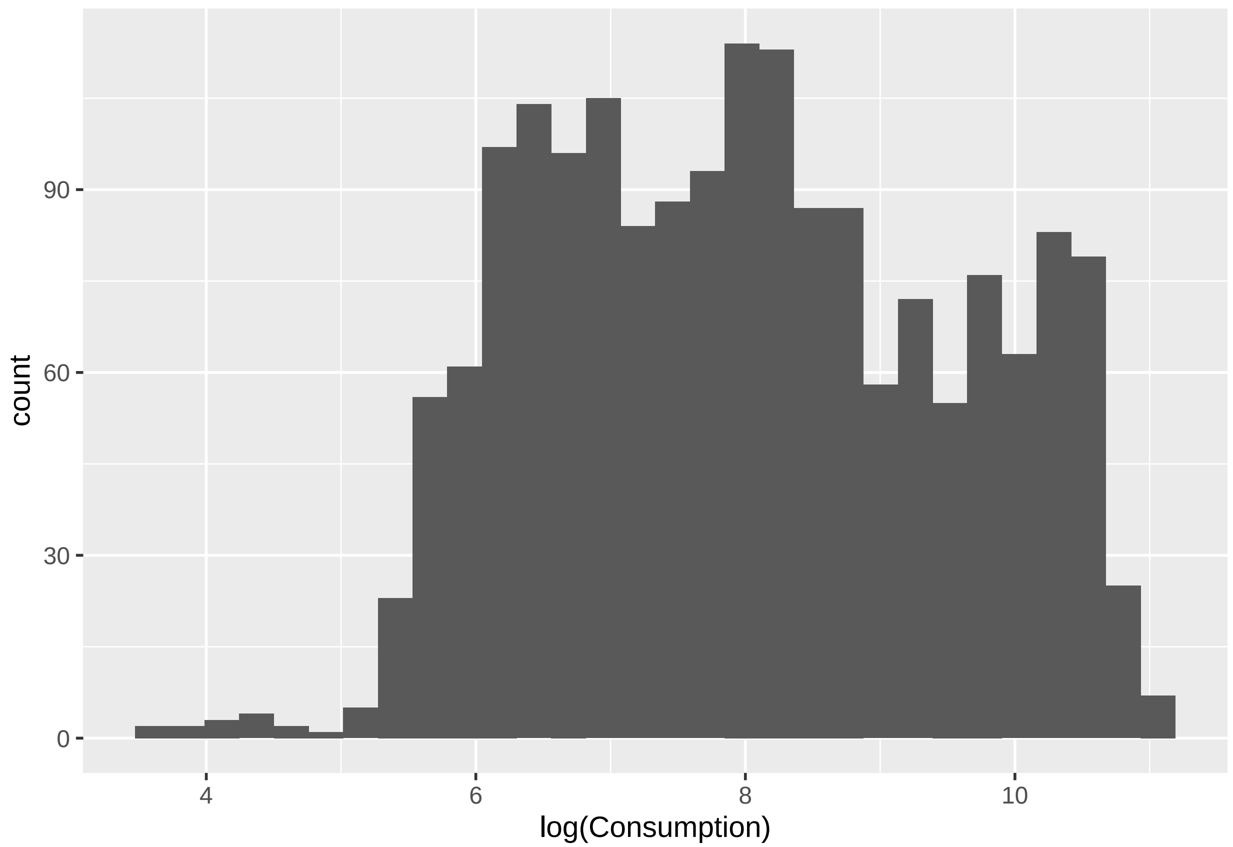
```
df %>% ggplot(aes(Consumption))+geom_histogram()
```



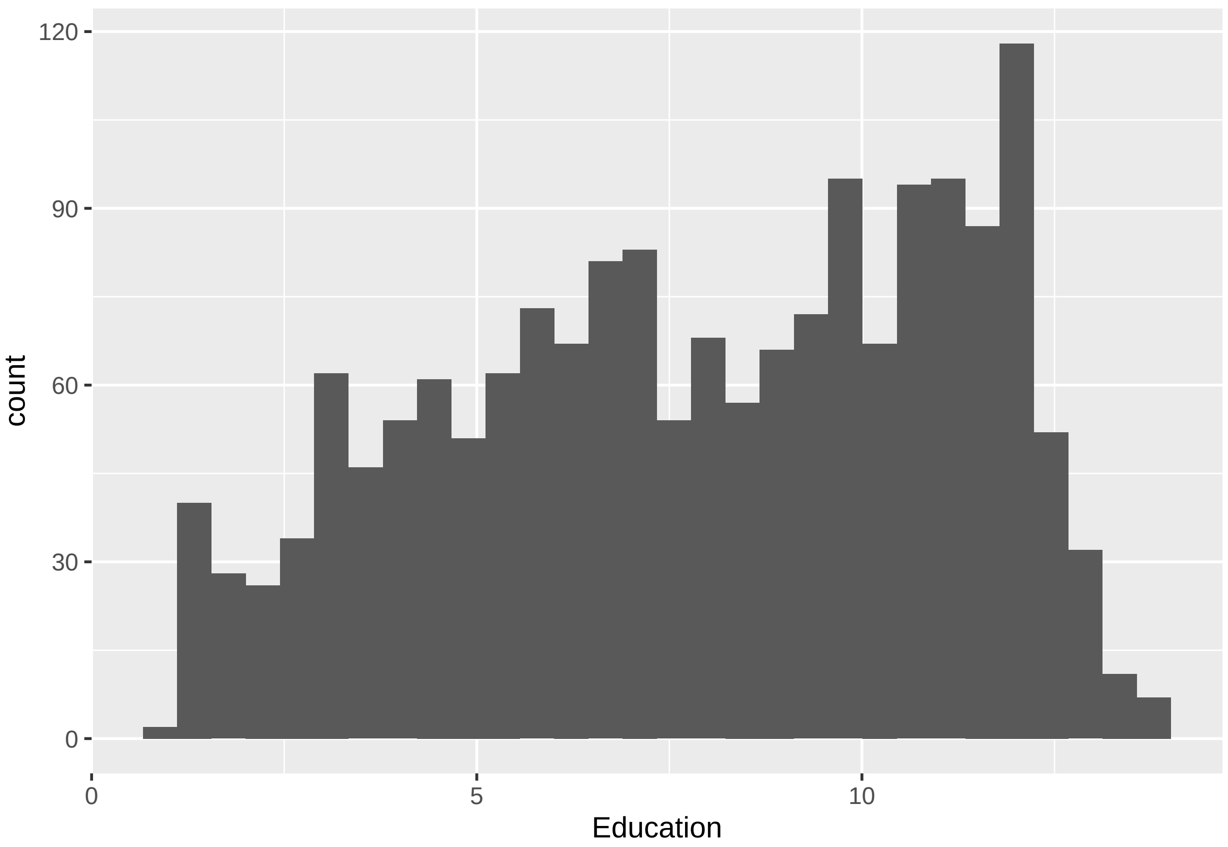
```
symbox(~Consumption, data=df)
```



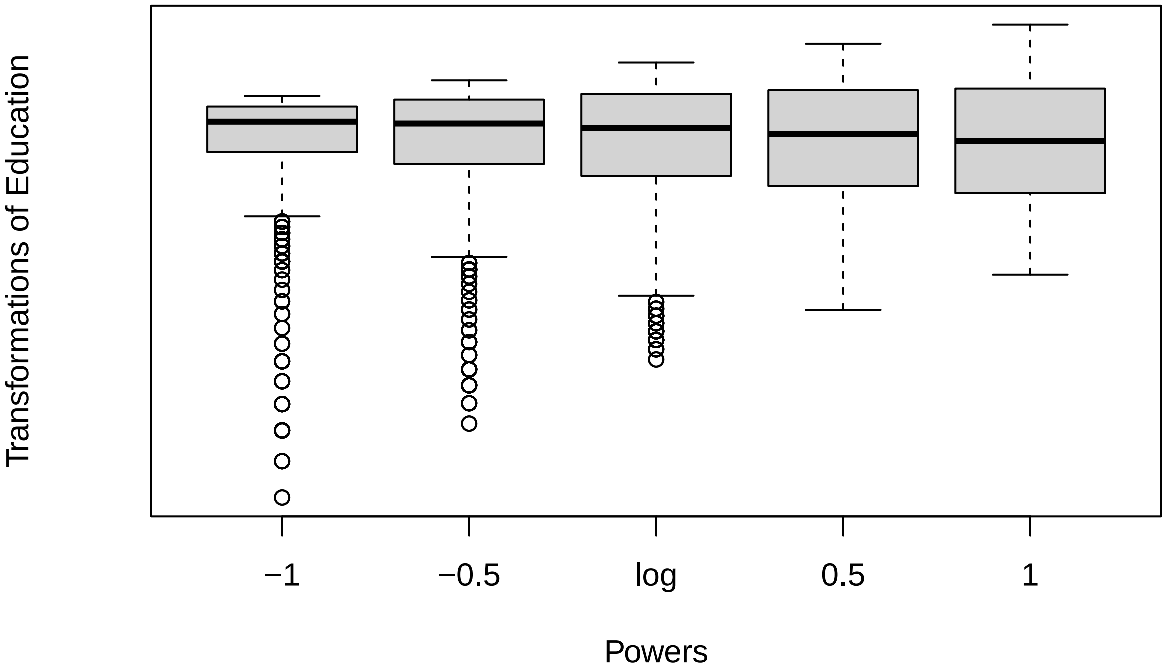
```
df %>% ggplot(aes(log(Consumption)))+geom_histogram()
```



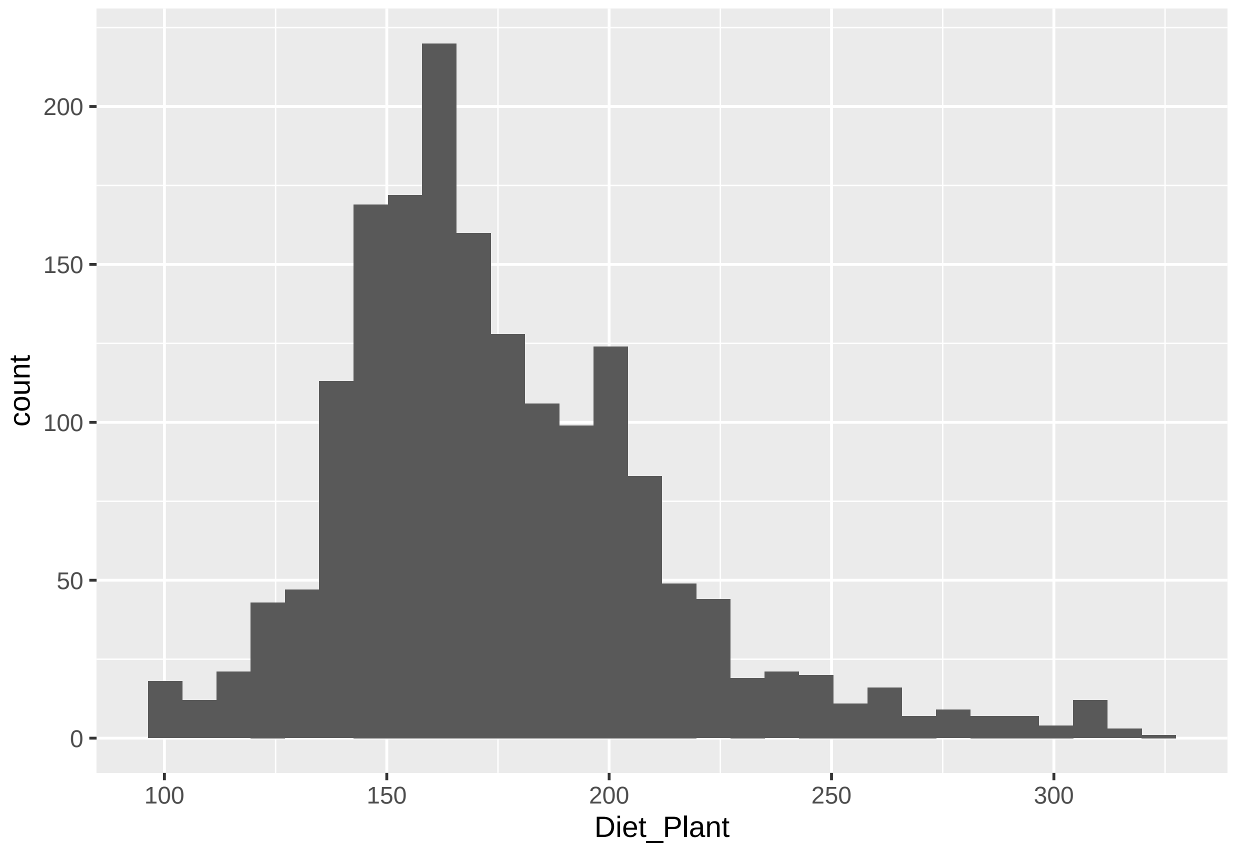
```
#log Consumption

df %>% ggplot(aes(Education))+geom_histogram()
```



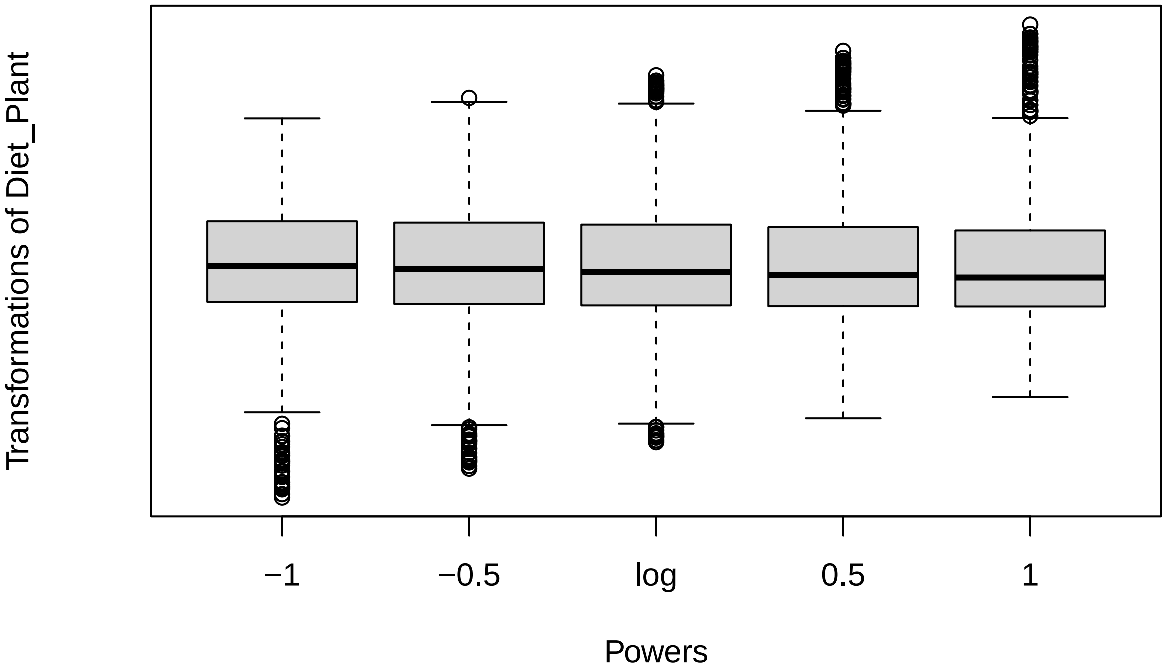
```
symbox(~Education, data=df)
```



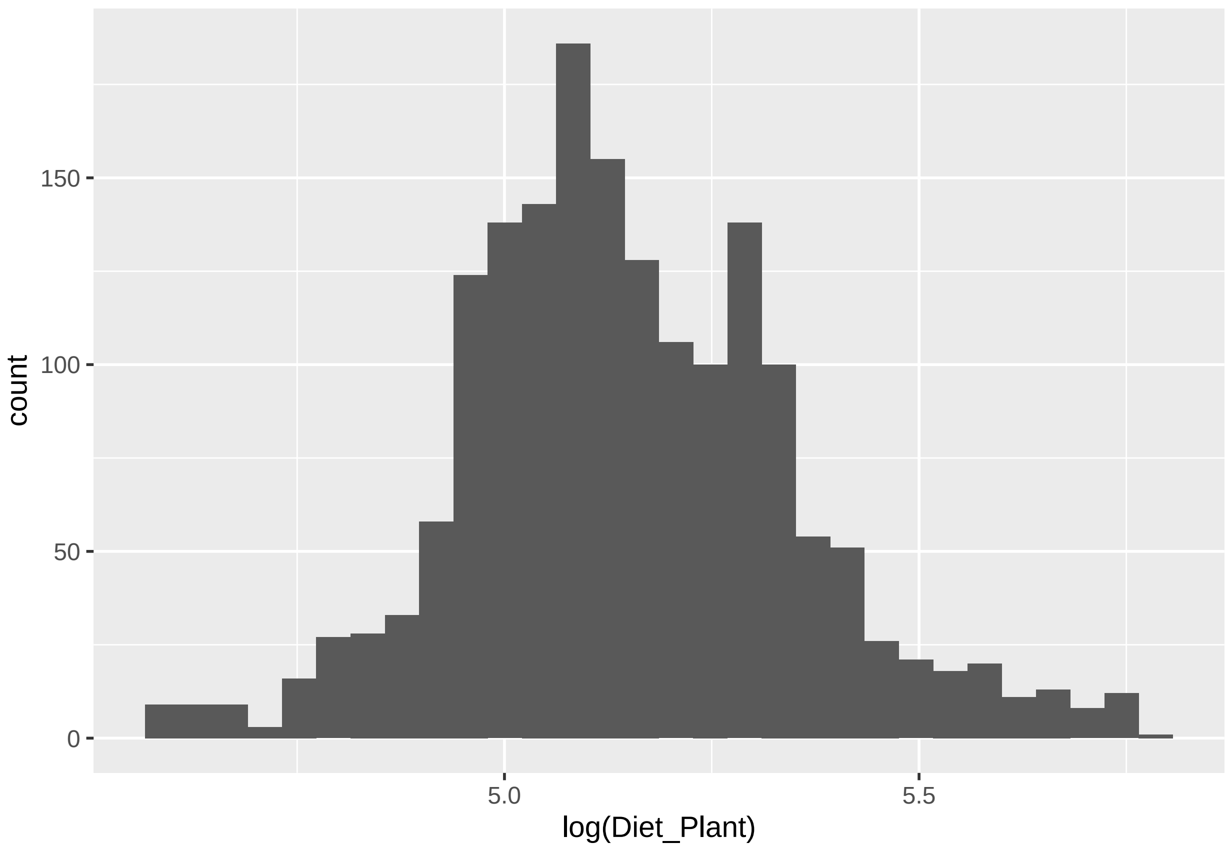
```
#keep education the same

df %>% ggplot(aes(Diet_Plant))+geom_histogram()
```



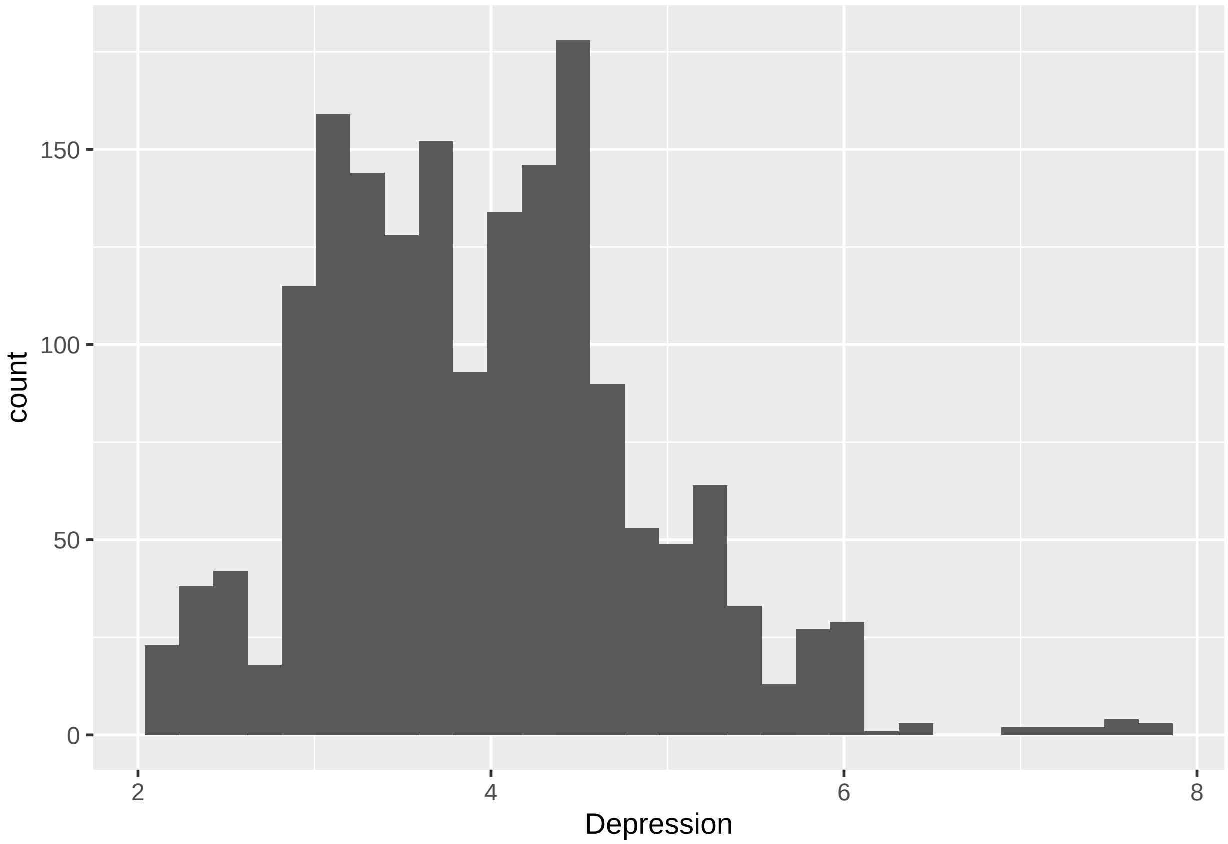
```
symbox(~Diet_Plant, data=df)
```



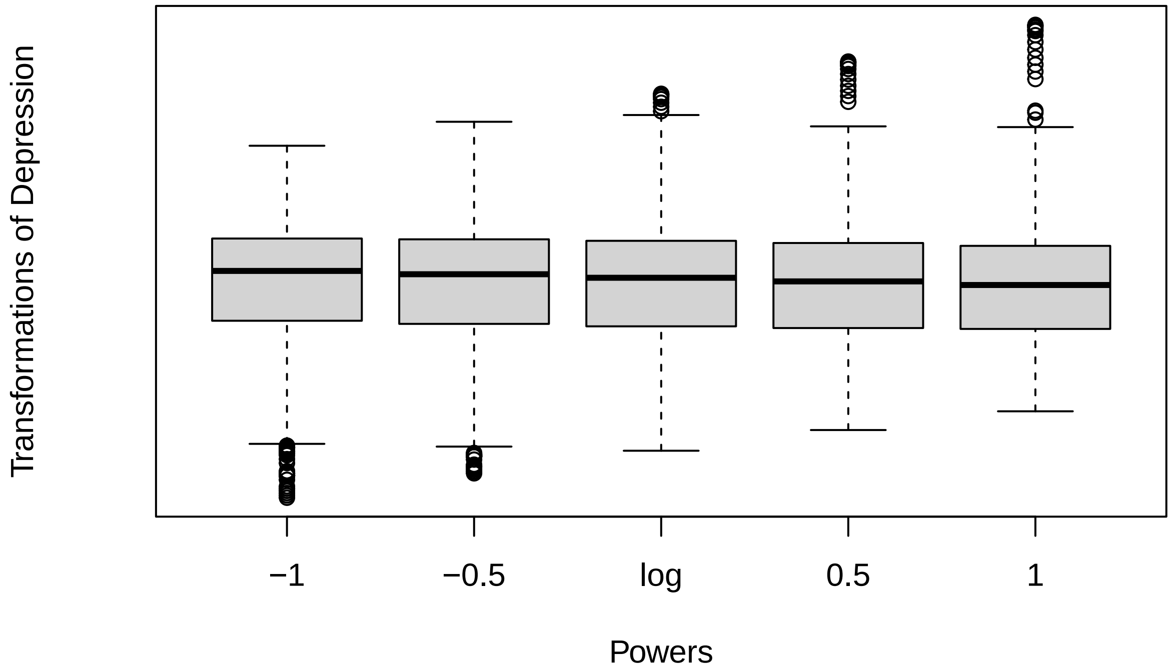
```
df %>% ggplot(aes(log(Diet_Plant)))+geom_histogram()
```



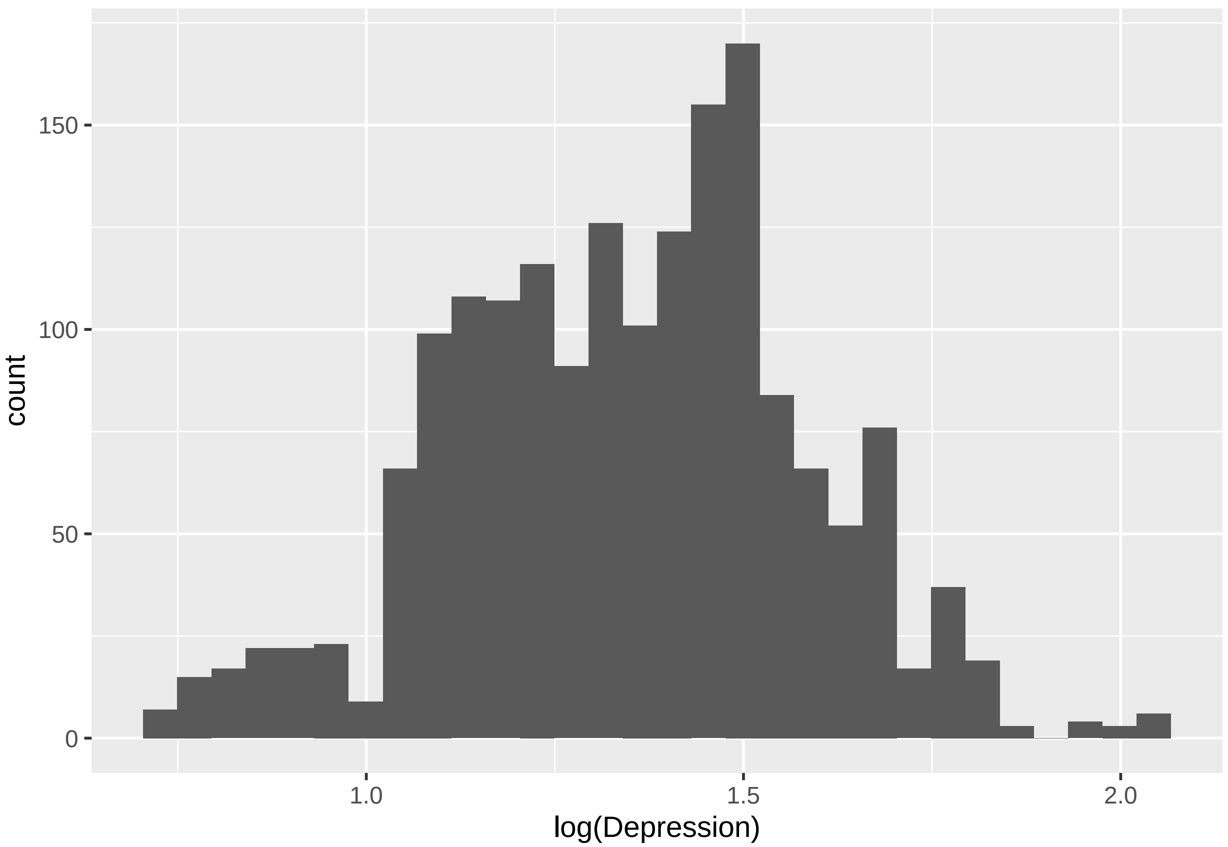
```
#data is large 1700+ so we are good

df %>% ggplot(aes(Depression))+geom_histogram()
```



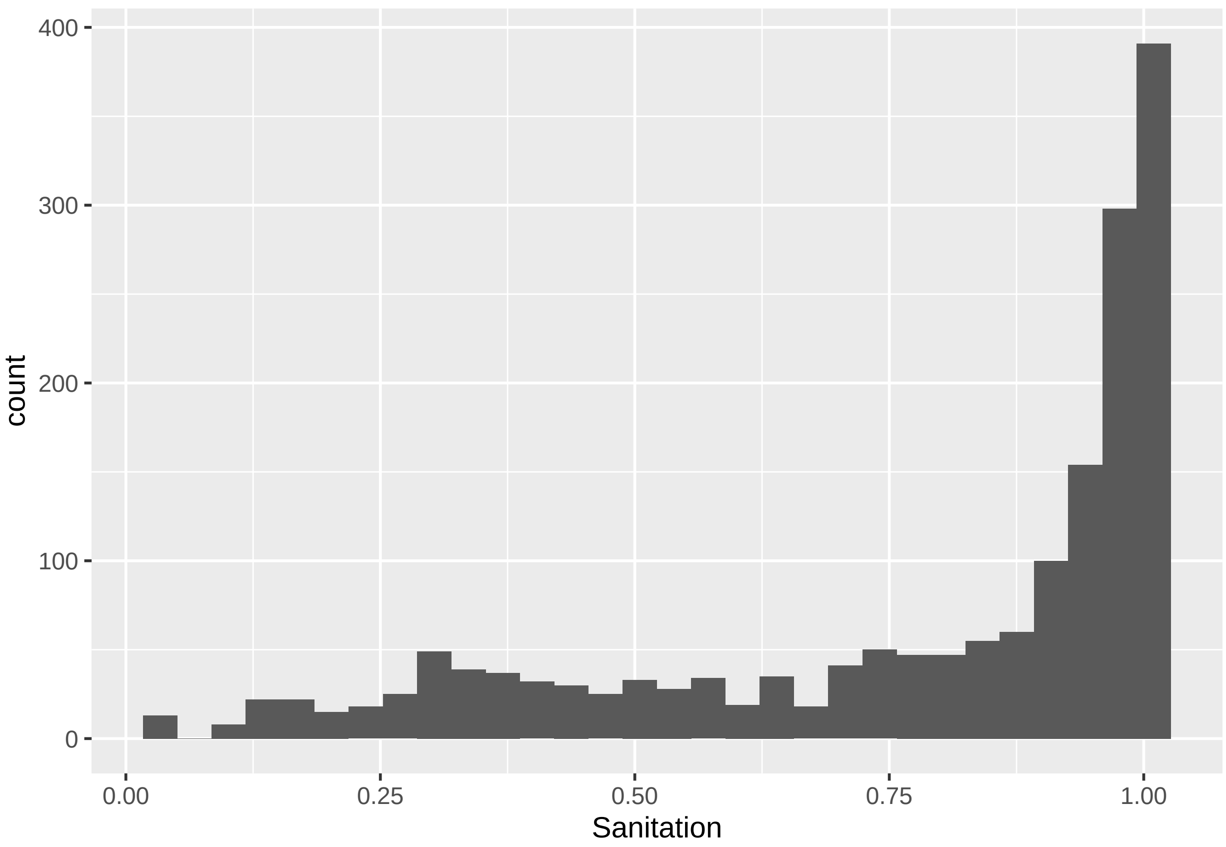
```
symbox(~Depression, data=df)
```



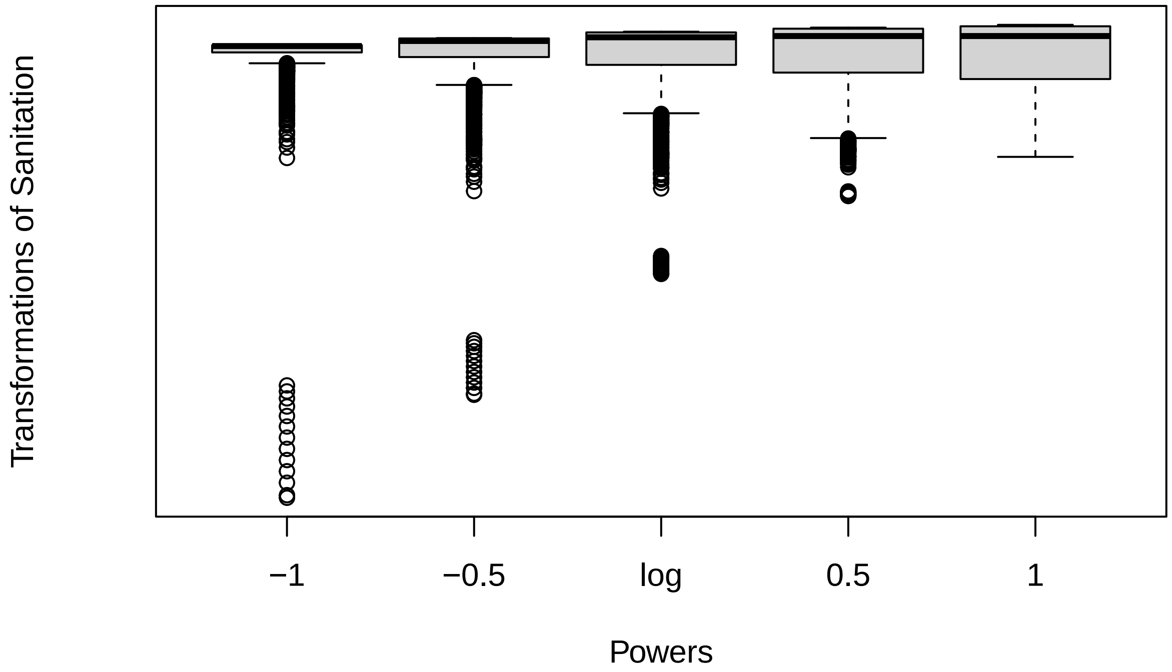
```
df %>% ggplot(aes(log(Depression)))+geom_histogram()
```



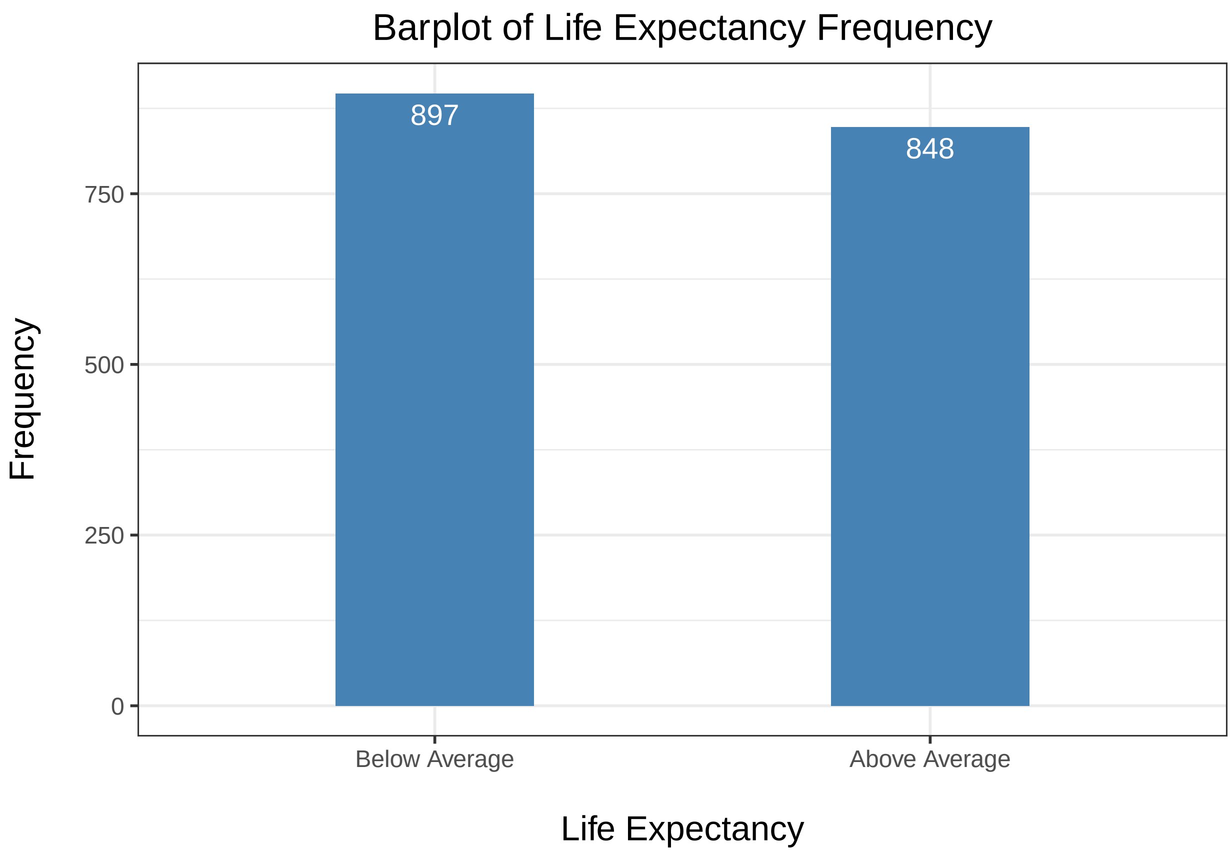
```
#data is large 1700+ so we are good

df %>% ggplot(aes(Sanitation))+geom_histogram()
```



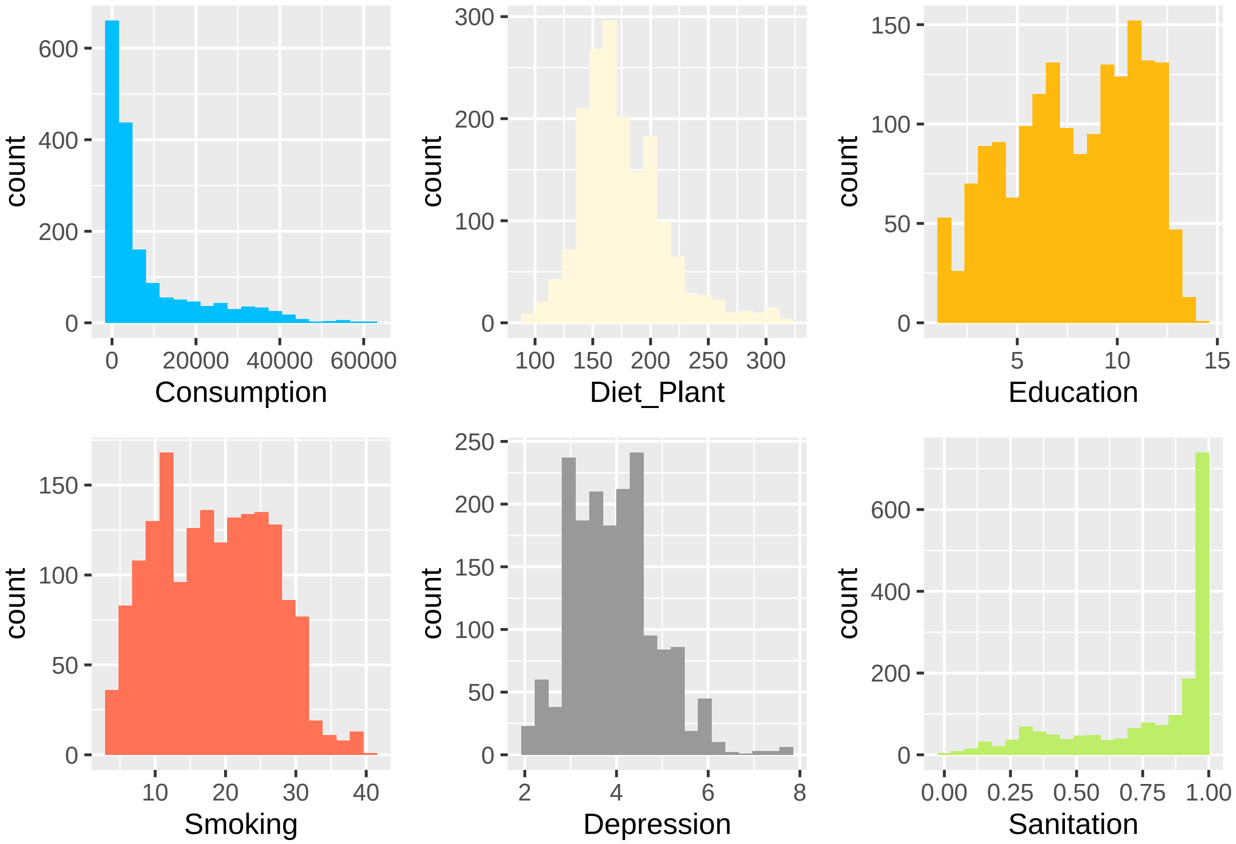
```
symbox(~Sanitation, data=df)
```



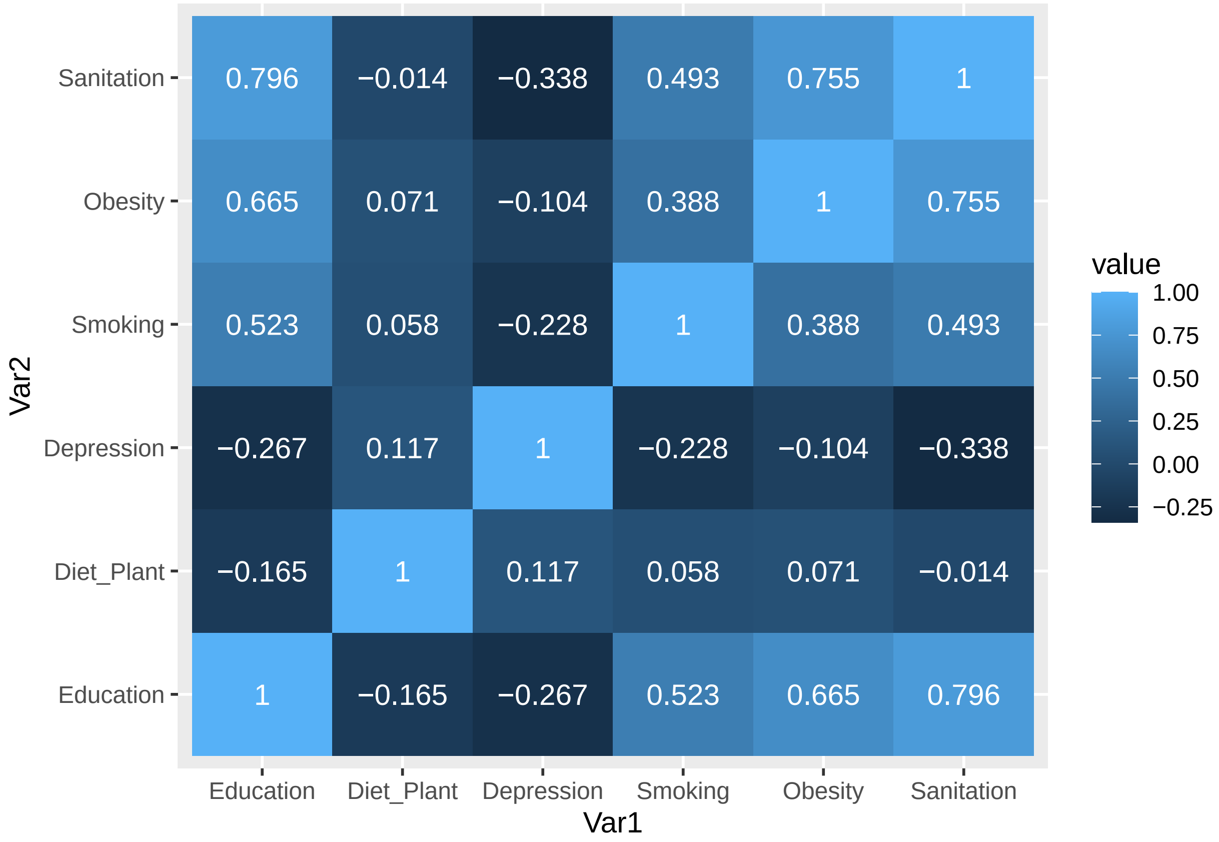
```
#view response variable
tab <- data.frame(table(df$Life_Expectancy))
a <- ggplot(tab,aes(x=Var1,y=Freq)) +
  geom_bar(stat="identity", fill="steelblue",width=0.4) +
  theme_bw() + xlab("\nLife Expectancy") + ylab("Frequency\n") +
  ggtitle("Barplot of Life Expectancy Frequency") +
  theme(plot.title = element_text(hjust = 0.5,size=14),
        axis.title.x = element_text(size = 12.5),
        axis.title.y = element_text(size = 12.5)) +
  geom_text(aes(label = Freq), vjust = 1.5, color="white",
             position = position_dodge(.9), size = 4)
a
```



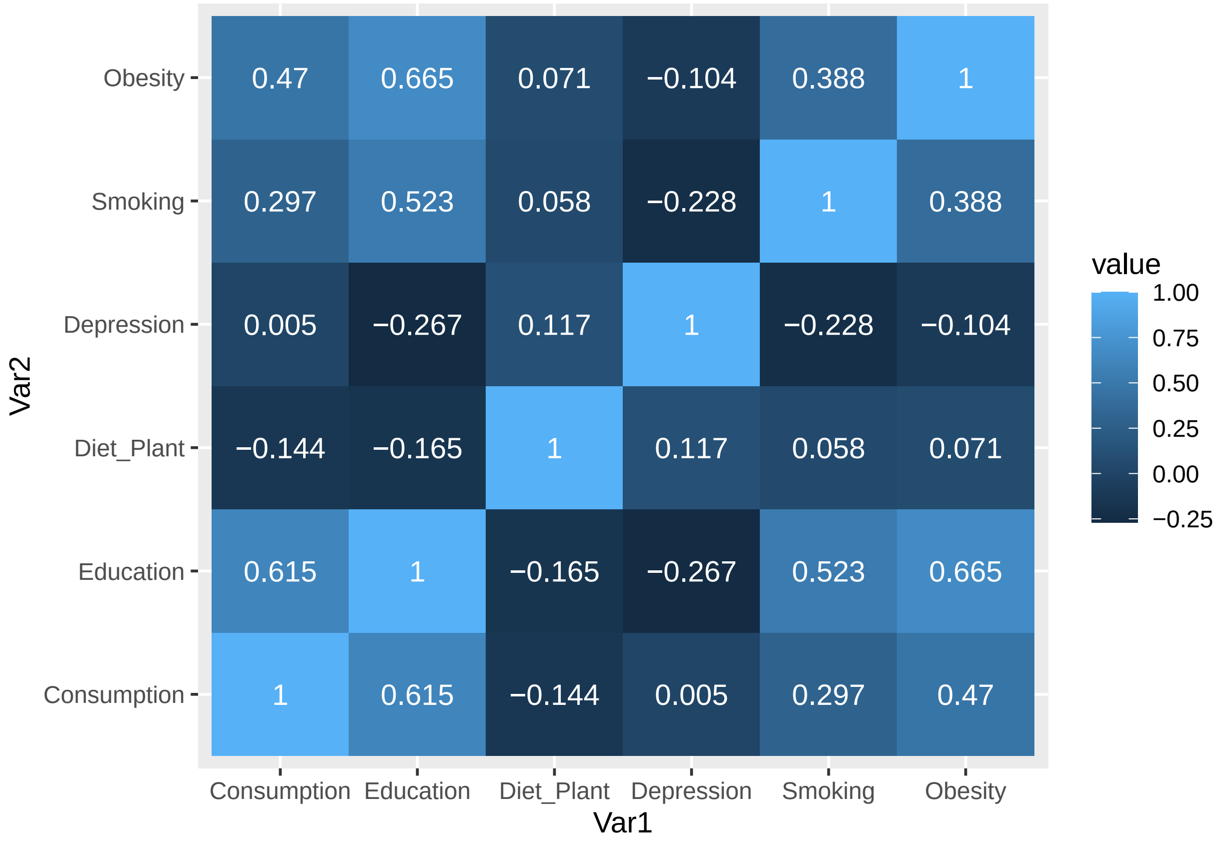
```
library(ggpubr)
c <- ggplot(df,aes(x=Consumption)) +
  geom_histogram(fill="deepskyblue",bins=20)
d <- ggplot(df,aes(x=Diet_Plant)) +
  geom_histogram(fill="cornsilk",bins=20)
e <- ggplot(df,aes(x=Education)) +
  geom_histogram(fill="darkgoldenrod1",bins=20)
f <- ggplot(df,aes(x=Smoking)) +
  geom_histogram(fill="coral1",bins=20)
g <- ggplot(df,aes(x=Depression)) +
  geom_histogram(fill="#999999",bins=20)
h <- ggplot(df,aes(x=Sanitation)) +
  geom_histogram(bins=20,fill="darkolivegreen2")
ggarrange(c,d,e,f,g,h)
```



```r
#heatmap for GDP-related
library(reshape2)
corr_mat <- round(cor(df[,4:9]),3)
melted_corr_mat <- melt(corr_mat)
ggplot(data = melted_corr_mat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value),
          color = "white", size = 4)
```



```
#heatmap for others
corr_mat <- df %>%
  select(Consumption, Education,Diet_Plant,Depression, Smoking, Obesity) %>%
  cor() %>% round(3)
melted_corr_mat <- melt(corr_mat)
ggplot(data = melted_corr_mat, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile() +
  geom_text(aes(Var2, Var1, label = value),
          color = "white", size = 4)
```

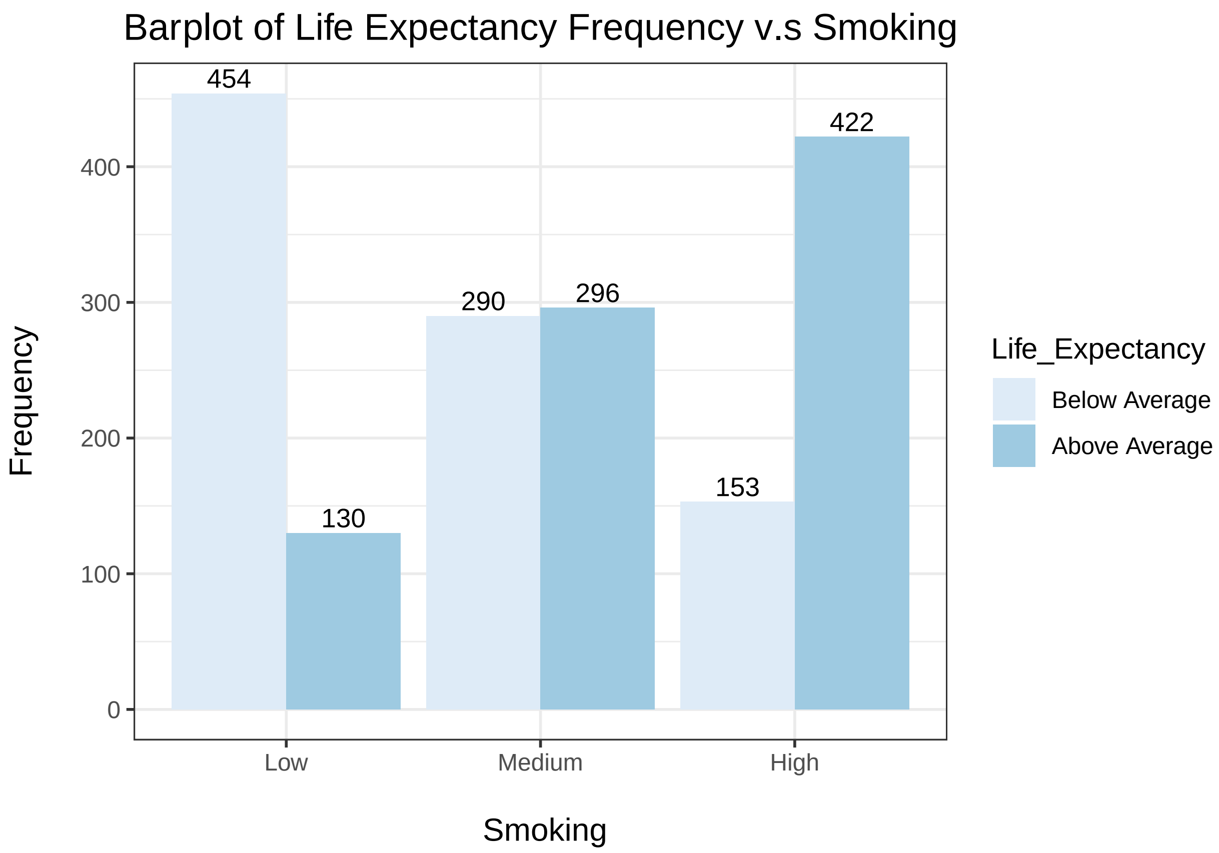


```
#convert smoking and sanitation
Smoking1 = cut(df$Smoking,breaks=(c(0,13.6,22.6,40)),labels=c("Low","Medium","High"))
Sanitation1 = cut(df$Sanitation,breaks=(c(0,0.95,1)),labels=c("Lack Access","Safe Access"),right = TRUE
df <- df %>%
  mutate(Smoking = as.factor(Smoking1)) %>%
  mutate(Sanitation = as.factor(Sanitation1)) %>%
  #change order
  select(-Smoking,-Sanitation,-Life_Expectancy,Smoking,Sanitation, Life_Expectancy)
```

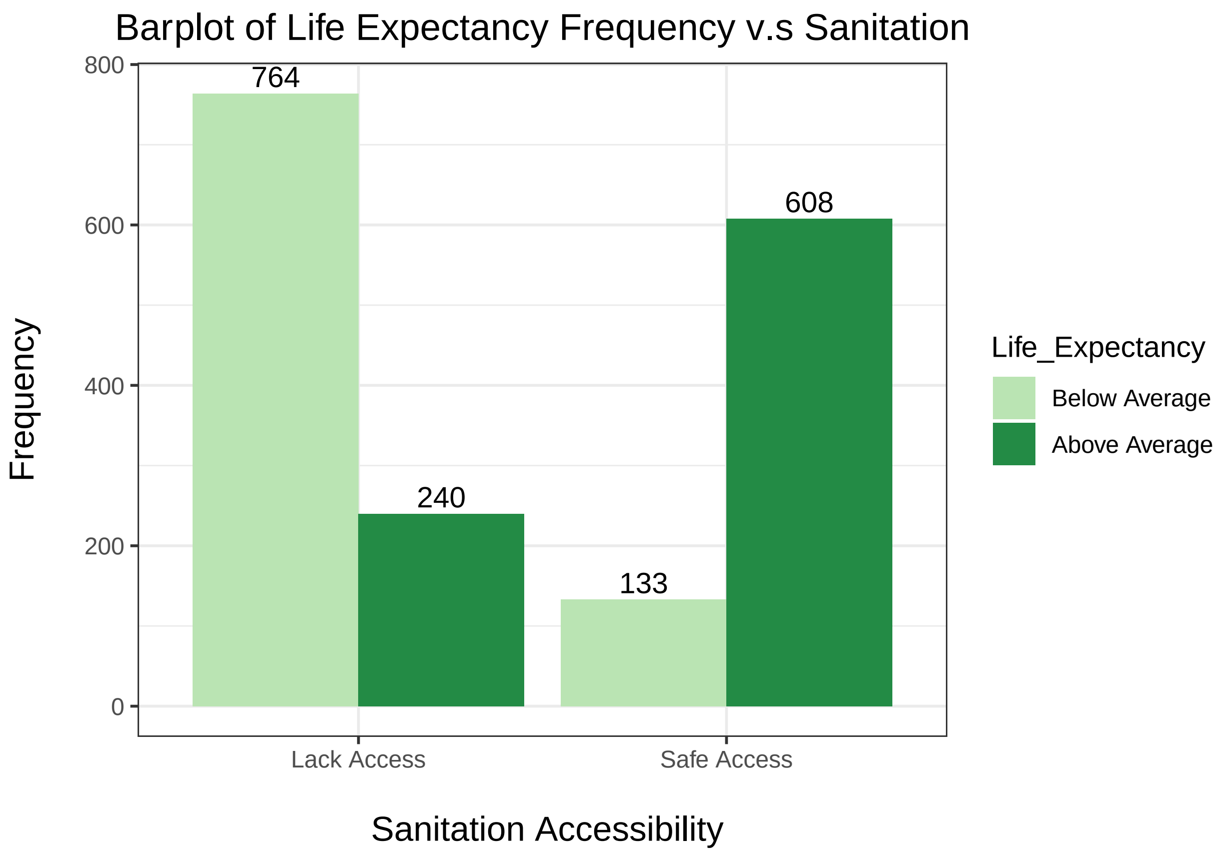
```
library(formattable)
formattable(table(df$Smoking,df$Life_Expectancy))
```

```
##
##          Below Average Above Average
##   Low    454           130
##   Medium 290           296
##   High   153           422
```

```
#versus continent
tab1 <- data.frame(table(df[,c("Smoking","Life_Expectancy")]))
b <- ggplot(tab1,aes(x=Smoking,y=Freq,fill=Life_Expectancy)) +
  geom_bar(stat="identity",alpha = 1.5,position = position_dodge()) +
  theme_bw() + ylab("Frequency\n") + scale_fill_brewer(palette="Blues") +
  ggtitle("Barplot of Life Expectancy Frequency v.s Smoking") +
  theme(plot.title = element_text(hjust = 0.5,size=14),
        axis.title.x = element_text(size = 12),
        axis.title.y = element_text(size = 12)) + xlab("\n Smoking") +
  geom_text(aes(label = Freq), vjust = -0.3,
             position = position_dodge(.9), size = 3.5)
b
```



```
#year
#table(df$Year)
tab2 <- data.frame(table(df[,c("Sanitation","Life_Expectancy")]))
y <- ggplot(tab2,aes(x=Sanitation,y=Freq,fill=Life_Expectancy)) +
  geom_bar(stat="identity",alpha = 1,position = position_dodge()) +
  theme_bw() + ylab("Frequency\n") +
  scale_fill_manual(values=c("#bae4b3","#238b45")) +
  ggtitle("Barplot of Life Expectancy Frequency v.s Sanitation") +
  theme(plot.title = element_text(hjust = 0.5,size=14),
        axis.title.x = element_text(size = 12.5),
        axis.title.y = element_text(size = 12.5)) +
  xlab("\n Sanitation Accessibility") +
  geom_text(aes(label = Freq), vjust = -0.3,
             position = position_dodge(.9), size = 4)
y
```



#Model 1

```
mod0 <- glm(Life_Expectancy~log(Consumption)+Education+Diet_Plant+Depression+Obesity+Smoking+Sanitation
summary(mod0)
```

```
##
## Call:
## glm(formula = Life_Expectancy ~ log(Consumption) + Education +
##     Diet_Plant + Depression + Obesity + Smoking + Sanitation,
##     family = "binomial", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.62410  -0.33860  -0.01957   0.23489   2.95427
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -17.884431   1.187969 -15.055  < 2e-16 ***
## log(Consumption)       2.792139   0.175540  15.906  < 2e-16 ***
## Education             -0.204527   0.048549  -4.213 2.52e-05 ***
## Diet_Plant             0.009304   0.002427   3.833 0.000127 ***
## Depression            -1.026707   0.124098  -8.273  < 2e-16 ***
## Obesity               -0.019390   0.008624  -2.248 0.024550 *
## SmokingMedium         -0.851409   0.246689  -3.451 0.000558 ***
## SmokingHigh            0.494269   0.238370   2.074 0.038123 *
## SanitationSafe Access  1.117511   0.213960   5.223 1.76e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2417.71  on 1744  degrees of freedom
## Residual deviance:  887.42  on 1736  degrees of freedom
## AIC: 905.42
##
## Number of Fisher Scoring iterations: 7
```

#VIF Model 1

```
vif(glm(Life_Expectancy ~ log(Consumption)+ Education + Diet_Plant + Depression + Obesity, df, family="
```

```
## log(Consumption)        Education       Diet_Plant       Depression
##         2.008312         1.585473         1.269884         1.300972
##          Obesity
##         1.647141
```

#Final Model

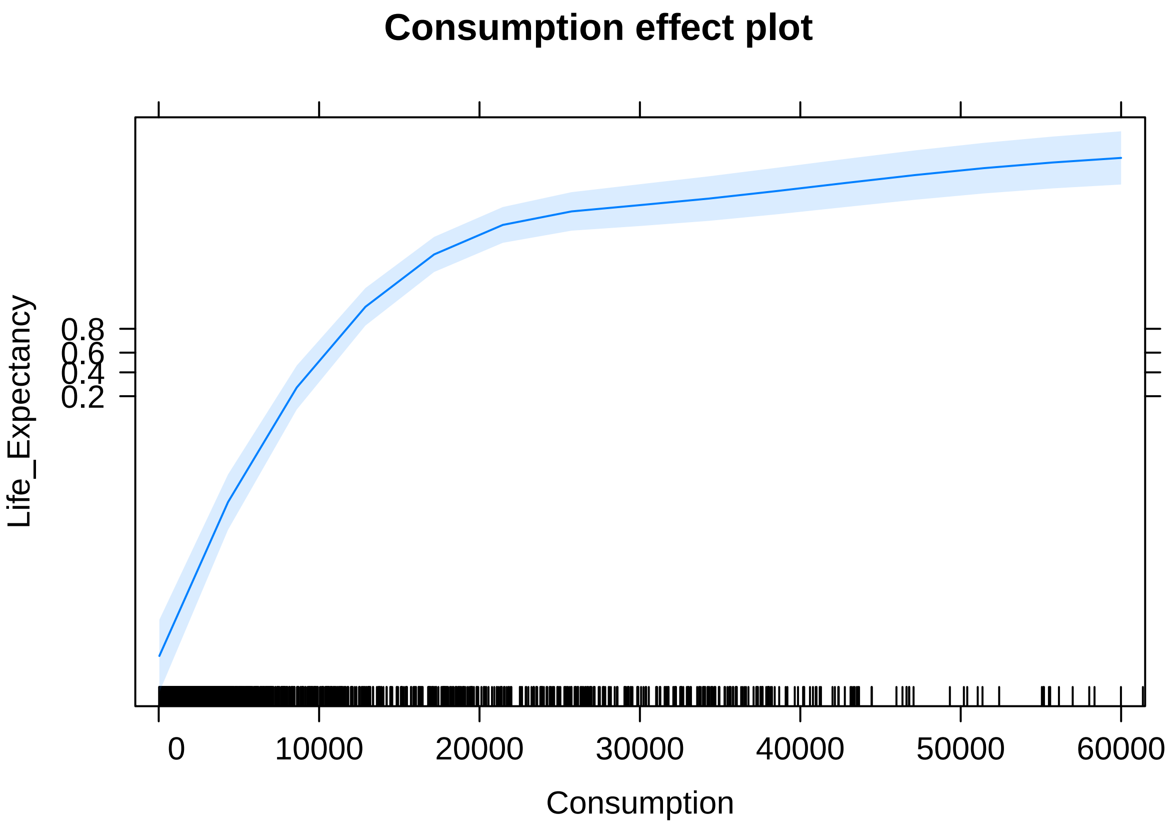
```
mod <- glm(Life_Expectancy~log(Consumption)+Education+Diet_Plant+Obesity+Depression+Smoking+Sanitation+
summary(mod)
```

```
##
## Call:
## glm(formula = Life_Expectancy ~ log(Consumption) + Education +
##     Diet_Plant + Obesity + Depression + Smoking + Sanitation +
##     Smoking * Sanitation, family = "binomial", data = df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.61214  -0.33630  -0.01955   0.22737   2.92287
##
## Coefficients:
##                                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)                           -17.948375   1.192210 -15.055  < 2e-16 ***
## log(Consumption)                        2.801324   0.177196  15.809  < 2e-16 ***
## Education                              -0.206832   0.048714  -4.246 2.18e-05 ***
## Diet_Plant                              0.009320   0.002471   3.772 0.000162 ***
## Obesity                                -0.020201   0.008803  -2.295 0.021741 *
## Depression                             -1.023018   0.125265  -8.167 3.17e-16 ***
## SmokingMedium                          -0.738441   0.274962  -2.686 0.007240 **
## SmokingHigh                             0.497425   0.265360   1.875 0.060857 .
## SanitationSafe Access                   1.515687   0.609856   2.485 0.012943 *
## SmokingMedium:SanitationSafe Access    -0.564689   0.671941  -0.840 0.400693
## SmokingHigh:SanitationSafe Access      -0.310895   0.689670  -0.451 0.652142
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

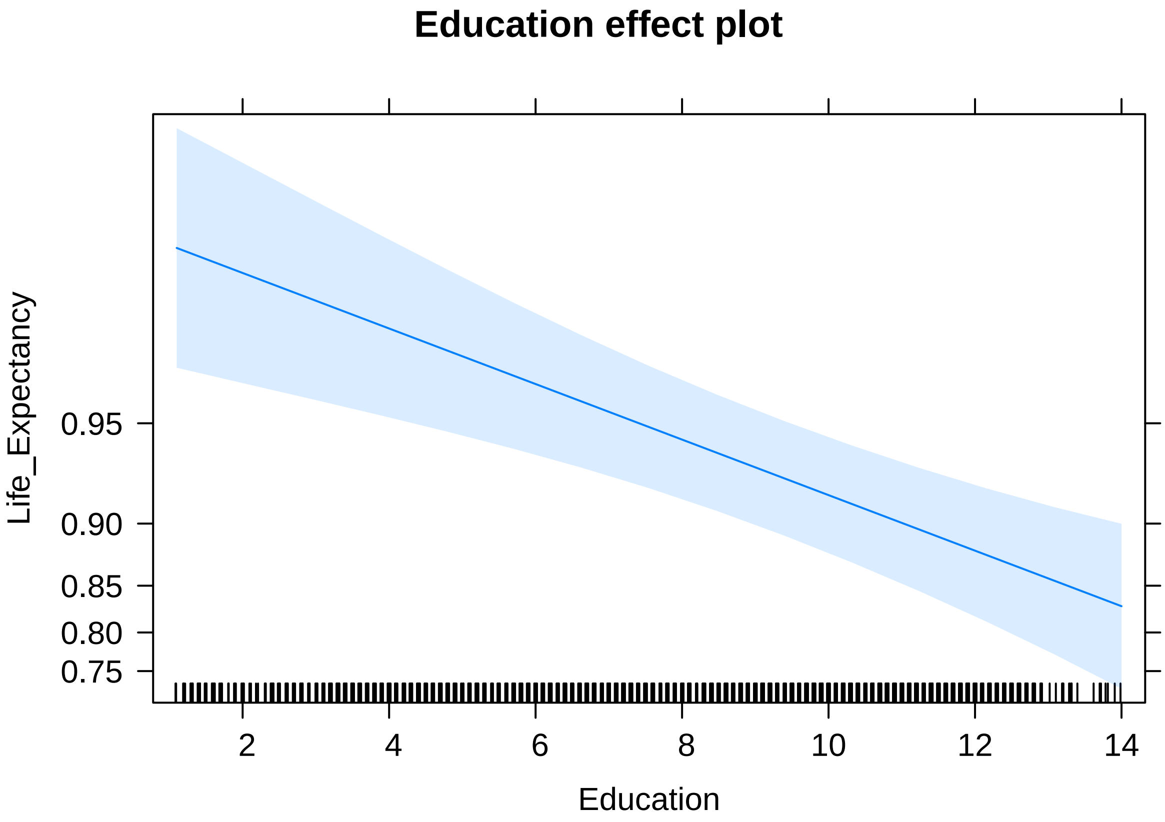
```
##
##     Null deviance: 2417.71  on 1744  degrees of freedom
## Residual deviance:  886.58  on 1734  degrees of freedom
## AIC: 908.58
##
## Number of Fisher Scoring iterations: 7
```

#Effect Plot and Interaction Plot

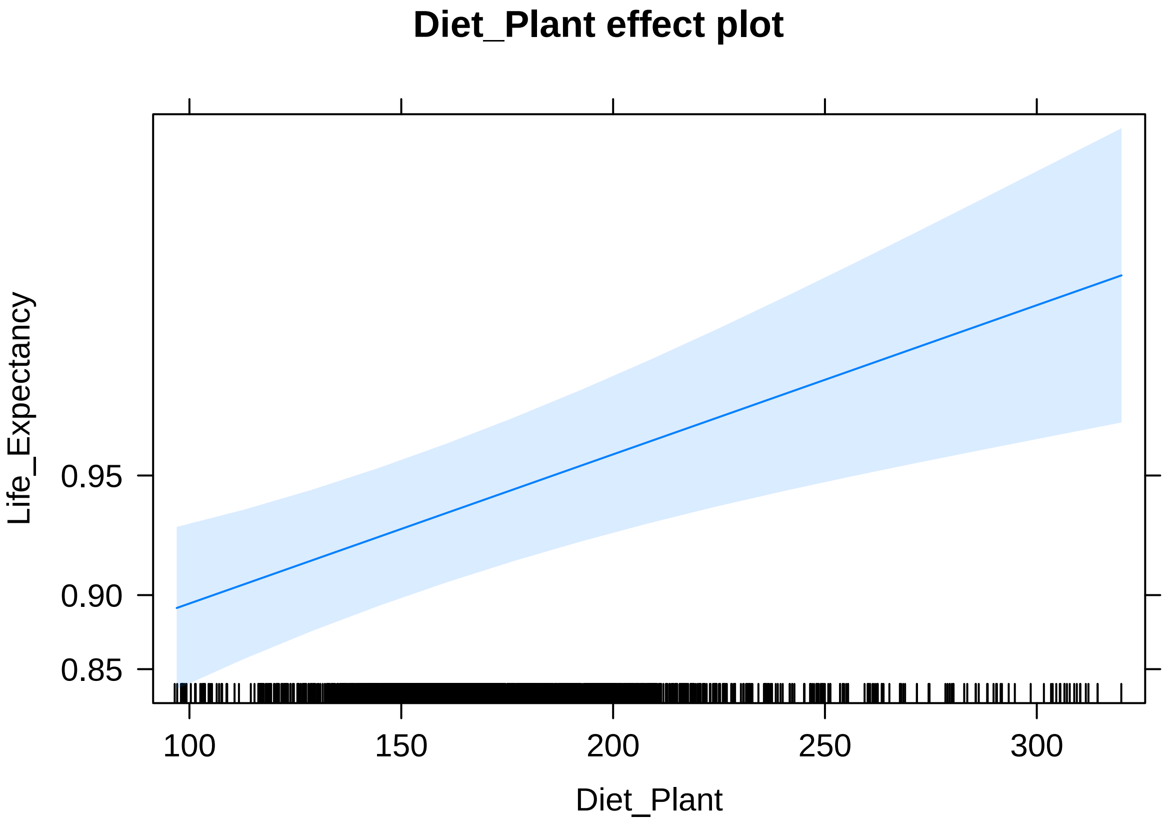
```
library(effects)
plot(Effect("Consumption",mod))
```



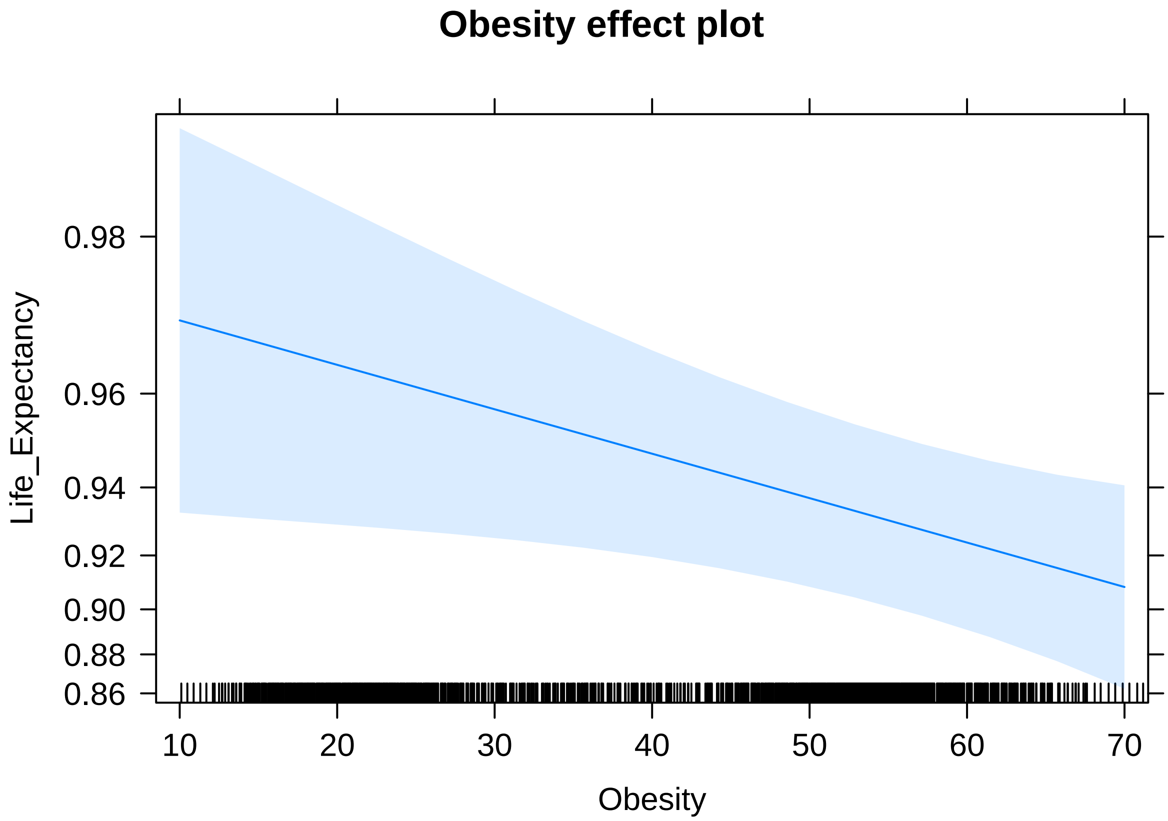
```
plot(Effect("Education",mod))
```



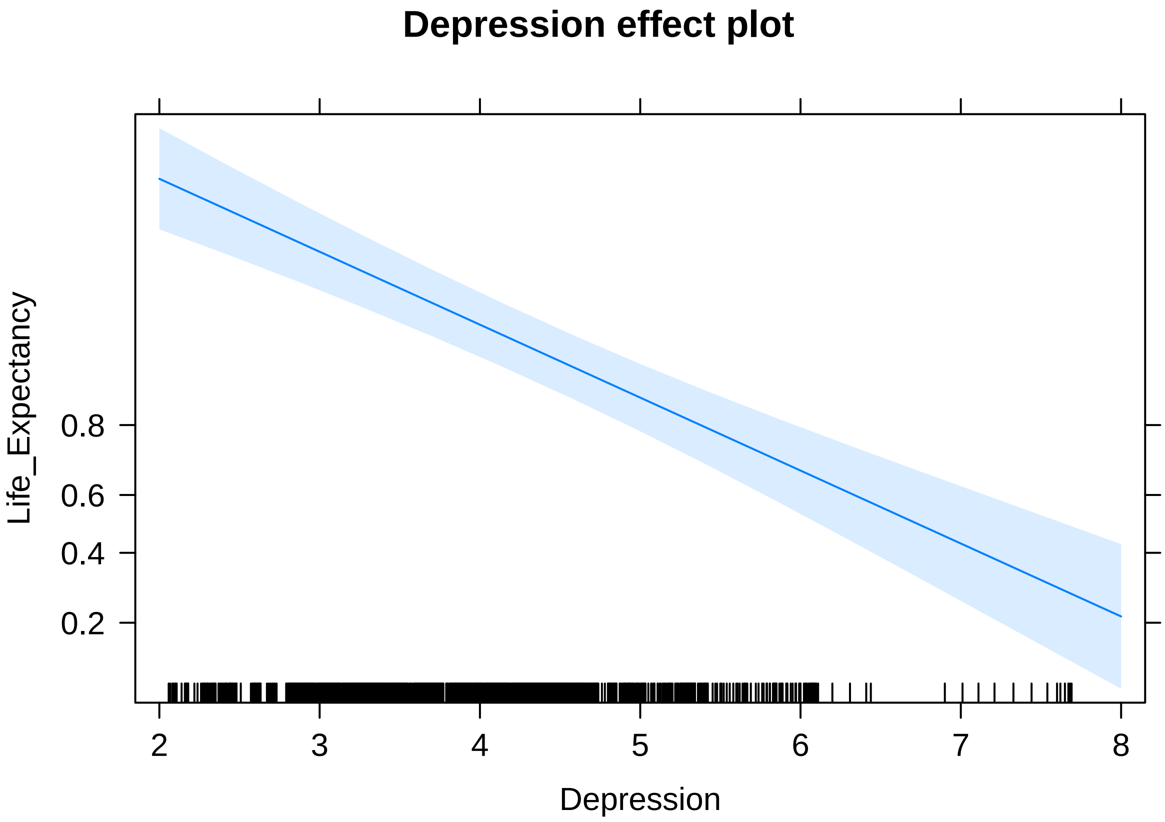
```
plot(Effect("Diet_Plant",mod))
```



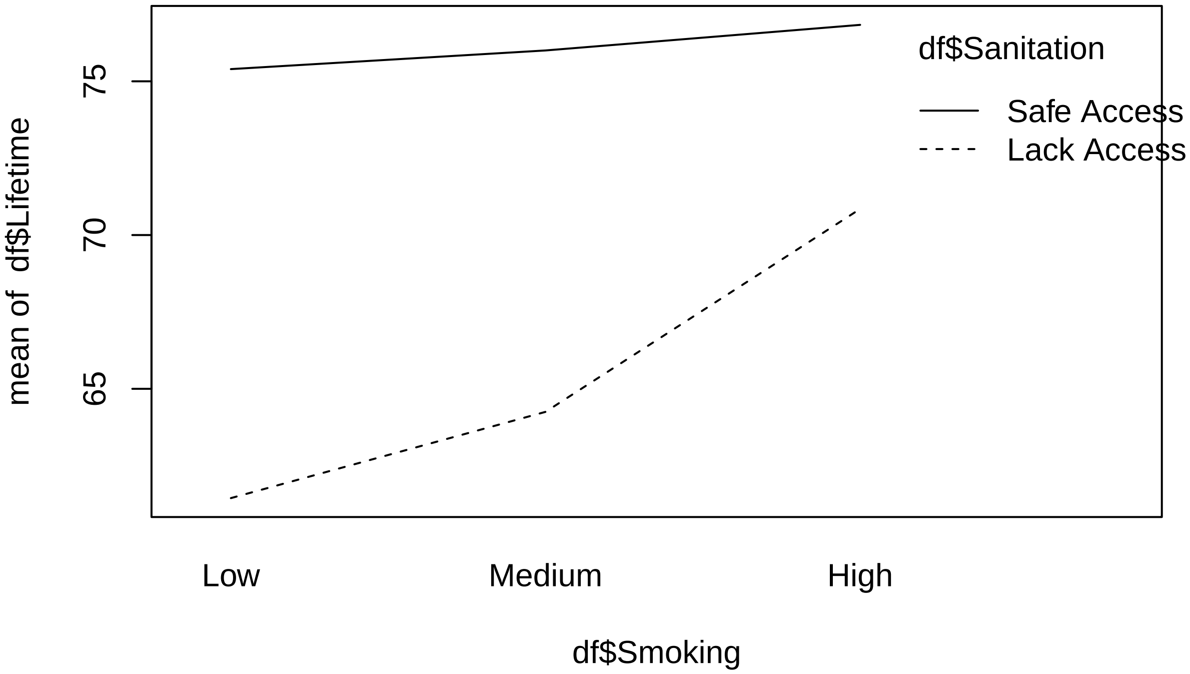
```
plot(Effect("Obesity",mod))
```



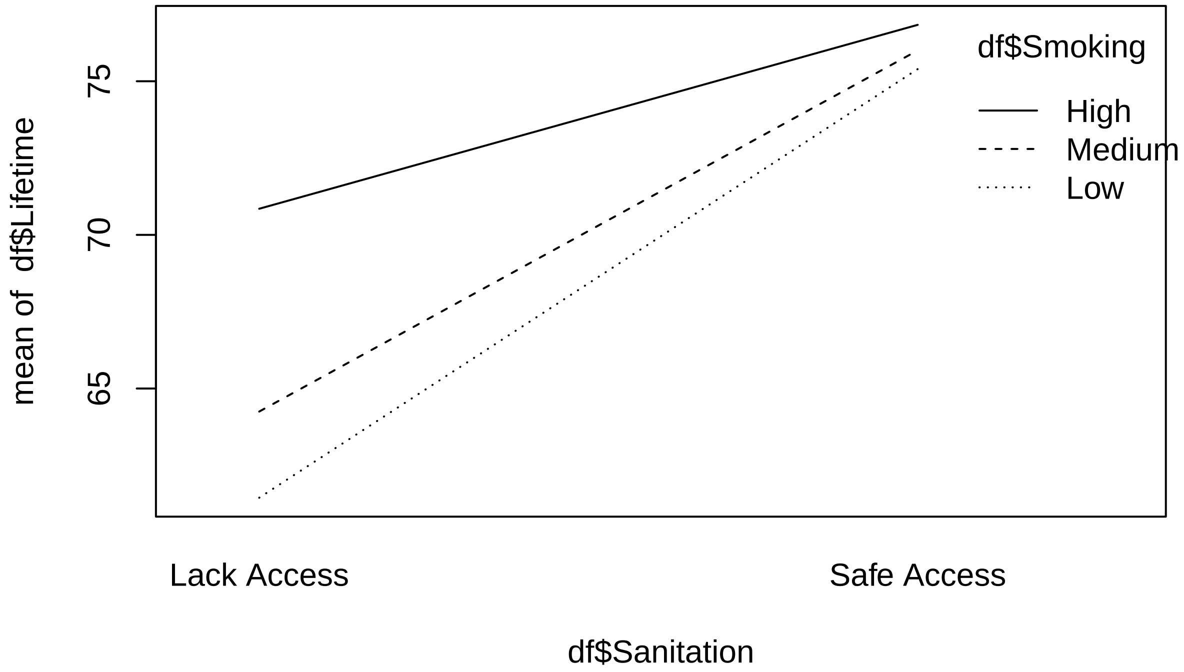
```
plot(Effect("Depression",mod))
```



```
# life <- ifelse(df$Life_Expectancy=="Above Average",1,0)
# Obe <- ifelse(df$Obesity>=49.7,"High","Low")
# Edu <- ifelse(df$Education >8,"High","Low")
# Dep <- ifelse(df$Depression >3.9,"High","Low")
interaction.plot(
    df$Smoking,#x-axis variable
    df$Sanitation,#variable for line
    df$Lifetime) #y-axis variable
```

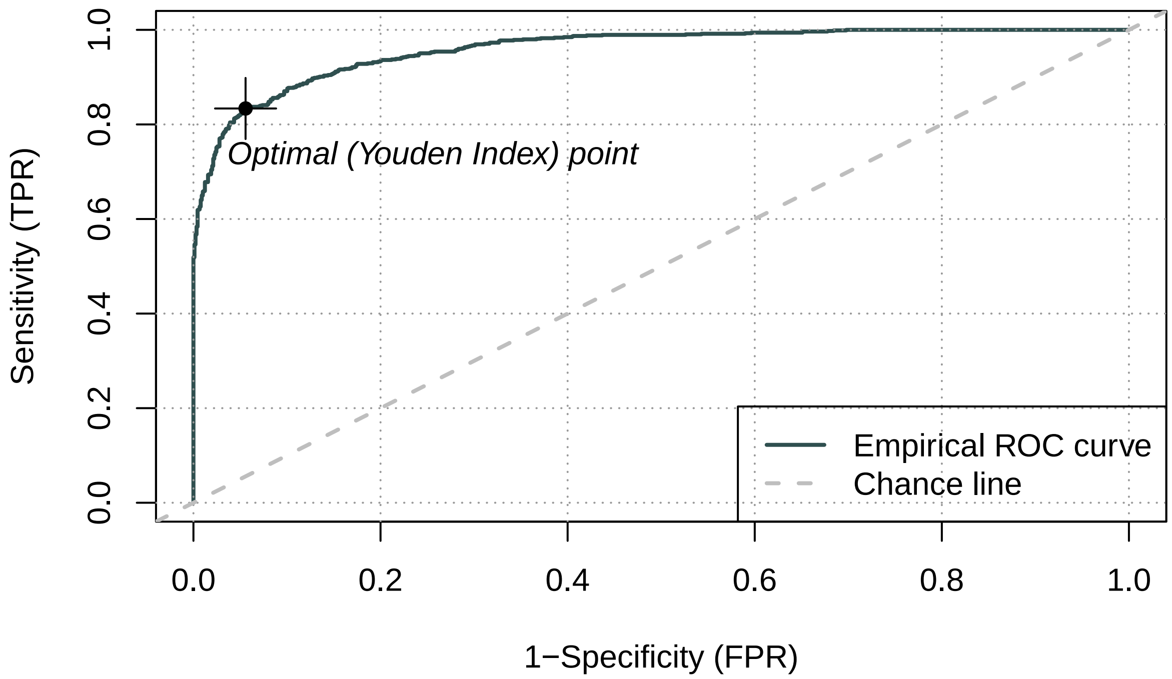


```
interaction.plot(
    df$Sanitation,#x-axis variable
    df$Smoking,#variable for line
    df$Lifetime) #y-axis variable
```



#Measure of Accuracy ROC Curve and AUC value

```
library(ROCit)
## Warning: package 'ROCit' was built under R version 3.5.2
ROCit_obj <- rocit(mod$fitted.values,df$Life_Expectancy)
plot(ROCit_obj)
```



```
ROCit_obj$AUC
```

```
## [1] 0.9595257
```

Confusion Matrix

```
predict_life <- predict(mod,df,"response")
predict_life <- as.factor(ifelse(predict_life > 0.5, "Above Average", "Below Average"))
confusionMatrix(predict_life,df$Life_Expectancy,mode = "everything")
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      Below Average Above Average
##   Below Average           814           119
##   Above Average            83           729
##
##                Accuracy : 0.8842
##                  95% CI : (0.8683, 0.8989)
##     No Information Rate : 0.514
##     P-Value [Acc > NIR] : < 2e-16
##
##                   Kappa : 0.768
##
```

```
##  Mcnemar's Test P-Value : 0.01379
##
##             Sensitivity : 0.9075
##             Specificity : 0.8597
##          Pos Pred Value : 0.8725
##          Neg Pred Value : 0.8978
##               Precision : 0.8725
##                  Recall : 0.9075
##                      F1 : 0.8896
##              Prevalence : 0.5140
##          Detection Rate : 0.4665
##    Detection Prevalence : 0.5347
##       Balanced Accuracy : 0.8836
##
##        'Positive' Class : Below Average
##
```

Using CV to test for overfitting

```
set.seed(827)

#Train test split; using 80% as training
sample <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.8,0.2))
train  <- df[sample, ]
test   <- df[!sample, ]

# Cross-Validation
set.seed(123)
train.control <- trainControl(method = "cv",
                              number = 10, repeats = 3)
# Train the model
model <- train(Life_Expectancy~log(Consumption)+Education+Depression+Smoking+Sanitation+Smoking*Educati
               method = "glm",family="binomial",
               trControl = train.control)

model$resample
```

```
##     Accuracy     Kappa Resample
## 1  0.9290780 0.8580918   Fold01
## 2  0.8928571 0.7858017   Fold02
## 3  0.8714286 0.7426471   Fold03
## 4  0.9219858 0.8440109   Fold04
## 5  0.8642857 0.7284606   Fold05
## 6  0.8642857 0.7277937   Fold06
## 7  0.8500000 0.6993865   Fold07
## 8  0.9078014 0.8155750   Fold08
## 9  0.8428571 0.6857143   Fold09
## 10 0.8936170 0.7869447   Fold10
```

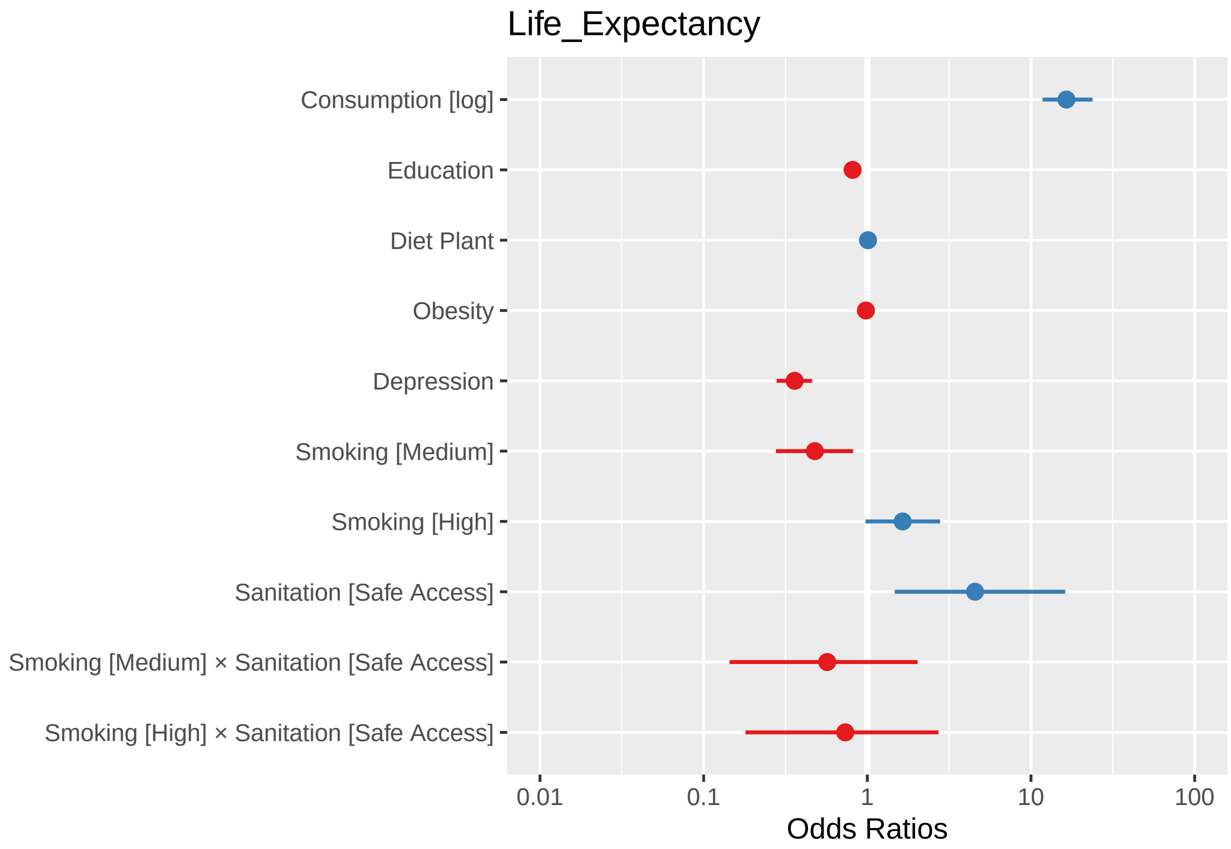
#Interpretation With confidence interval of exponentiated coefficients, we can interpret the Odds and Logit Scores

```
c1 = exp(coef(mod))
c2 = exp(confint(mod))
cbind(Estimate=c1,c2)
```

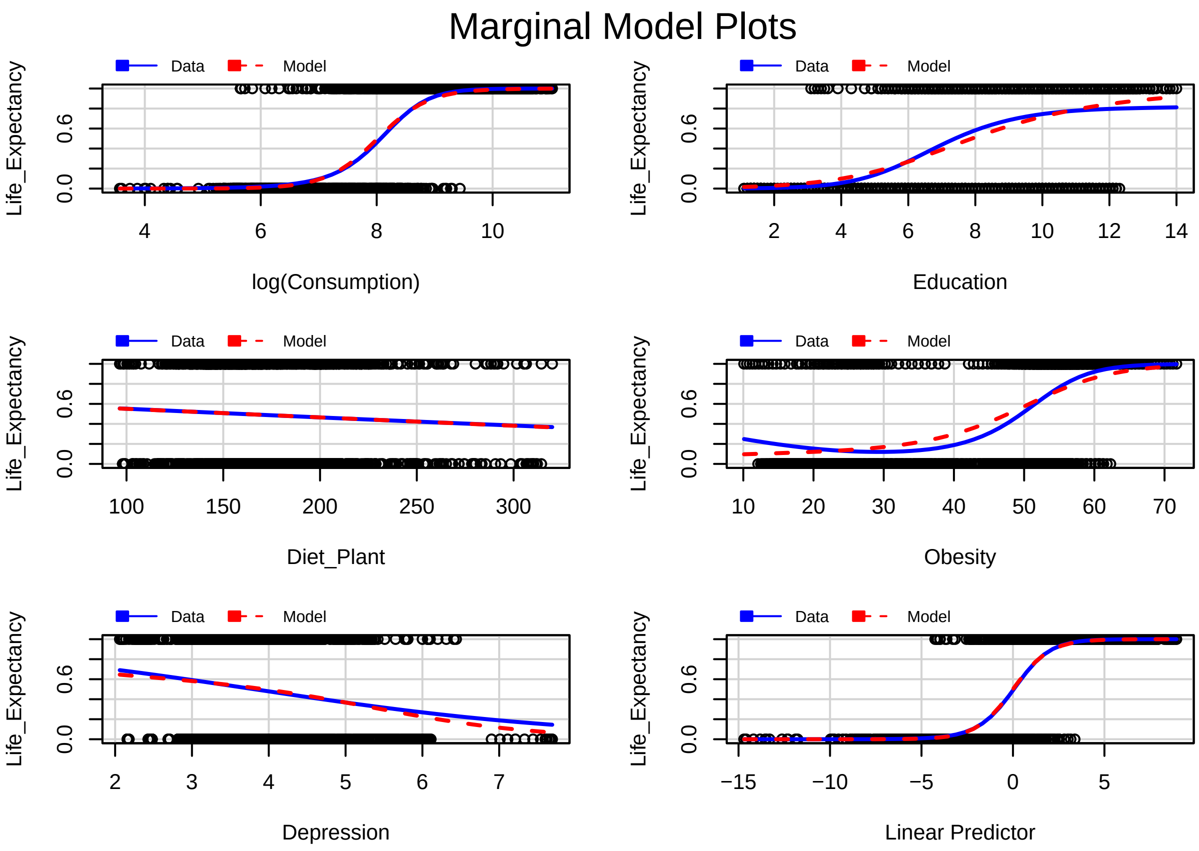
```
##                                        Estimate        2.5 %       97.5 %
## (Intercept)                        1.603688e-08 1.418983e-09 1.527485e-07
## log(Consumption)                   1.646644e+01 1.179431e+01 2.364041e+01
## Education                          8.131563e-01 7.381660e-01 8.936830e-01
## Diet_Plant                         1.009364e+00 1.004525e+00 1.014313e+00
## Obesity                            9.800019e-01 9.631695e-01 9.970274e-01
## Depression                         3.595081e-01 2.797131e-01 4.572925e-01
## SmokingMedium                      4.778584e-01 2.768043e-01 8.145063e-01
## SmokingHigh                        1.644482e+00 9.758110e-01 2.765065e+00
## SanitationSafe Access              4.552548e+00 1.472410e+00 1.610509e+01
## SmokingMedium:SanitationSafe Access 5.685370e-01 1.441786e-01 2.017302e+00
## SmokingHigh:SanitationSafe Access   7.327906e-01 1.805250e-01 2.709229e+00
```

Plot of Odds

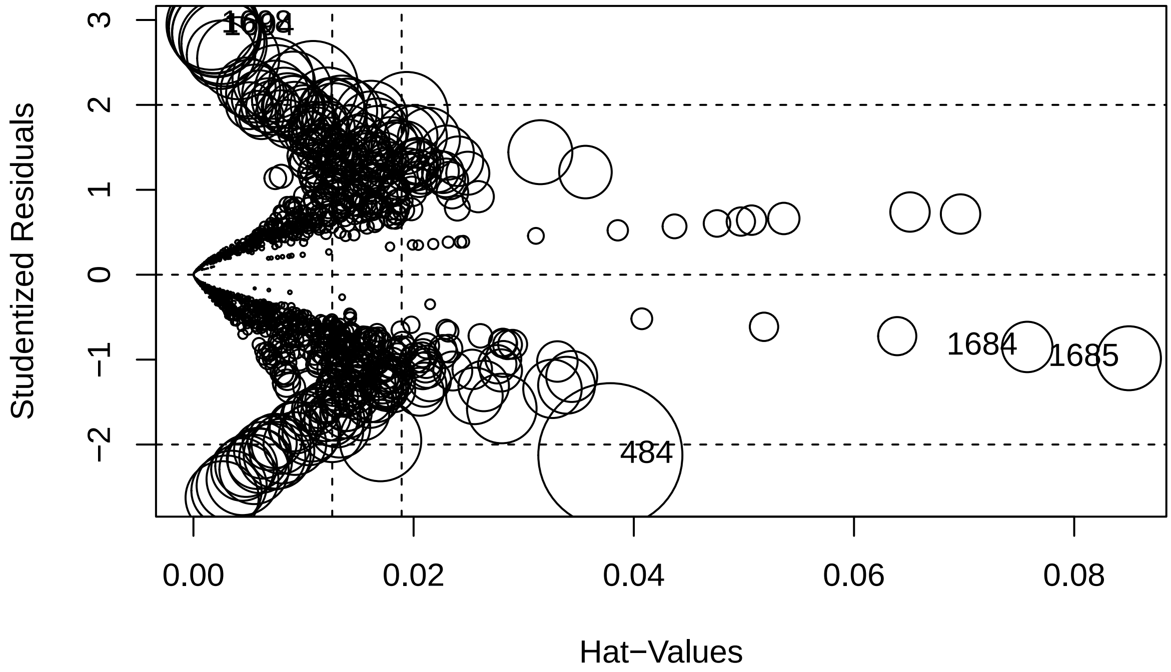
```
library(sjPlot)
plot_model(mod)
```



```
mmps(mod,~log(Consumption)+Education+Diet_Plant+Obesity+Depression+Smoking+Sanitation+ Smoking*Sanitati
```



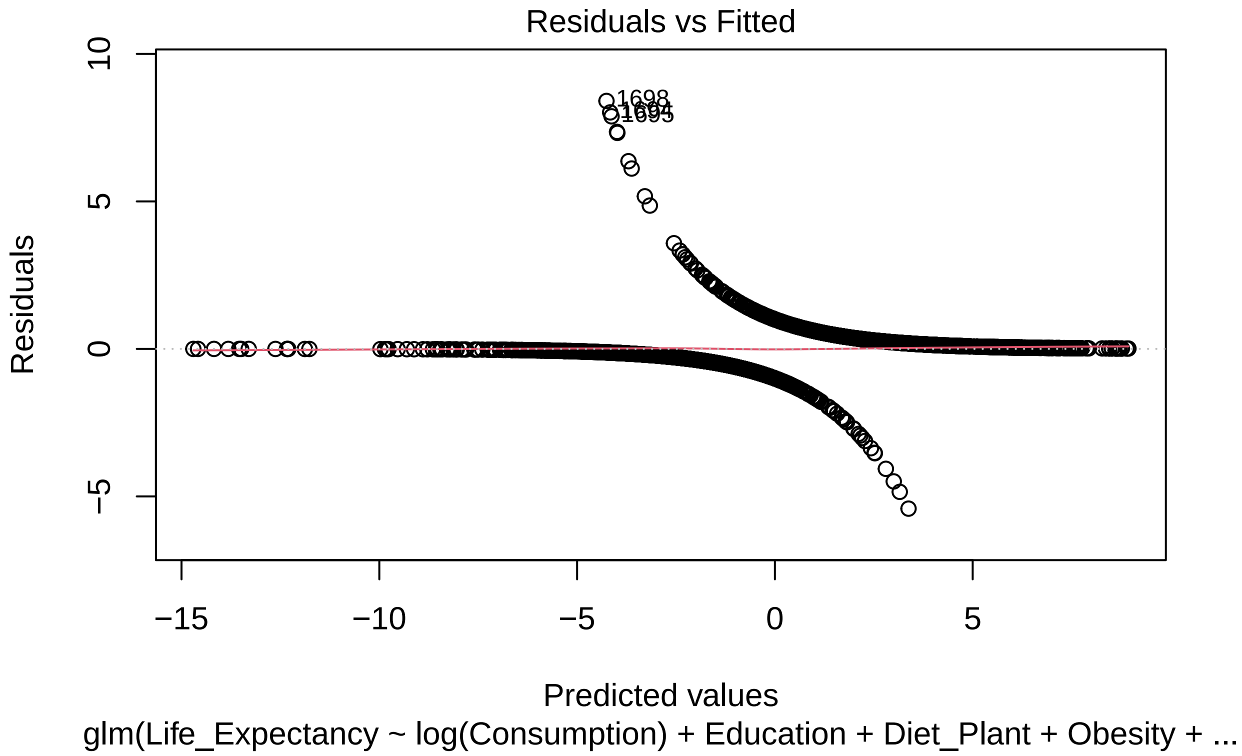
```
influencePlot(mod)
```



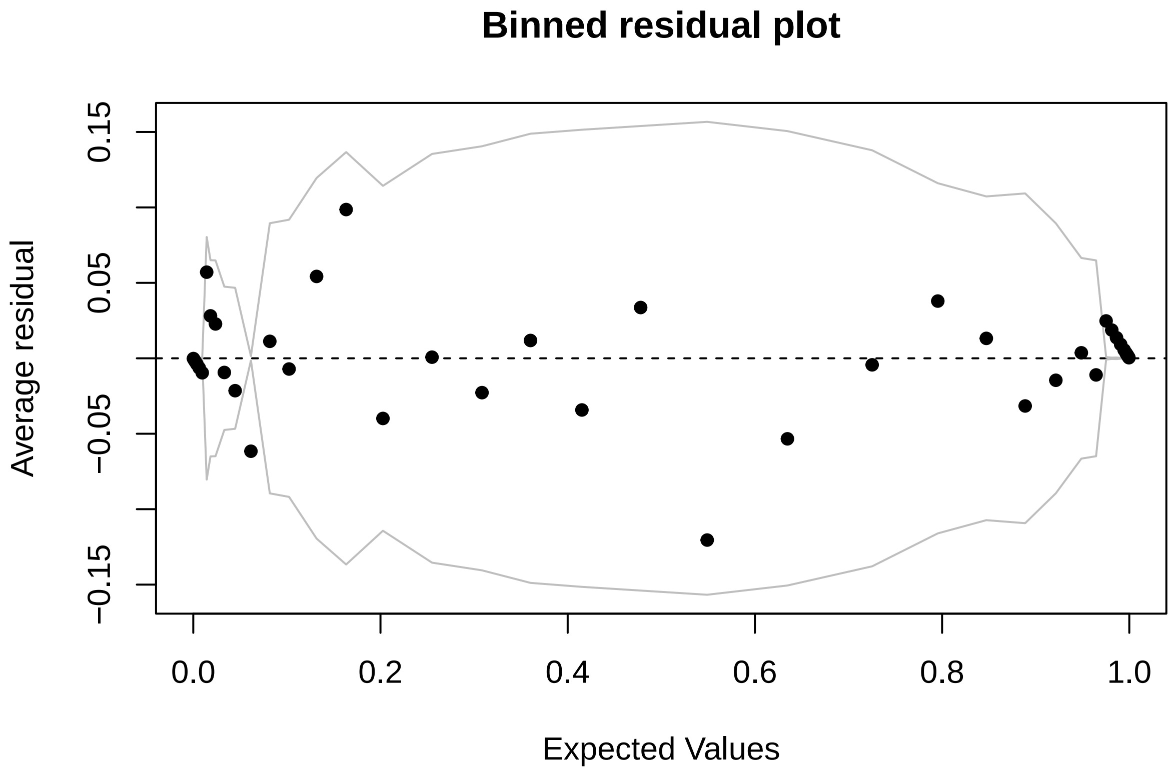
```
##          StudRes         Hat       CookD
## 484  -2.1255754 0.037869796 0.027125611
## 1684 -0.8507873 0.075737817 0.003321023
## 1685 -0.9841300 0.084971046 0.005349679
## 1694  2.9114524 0.001853836 0.010872369
## 1698  2.9426353 0.001638488 0.010555942
```

#Variance Analysis

```
#better residual plot by binning into categories
library(arm)
plot(mod,1)
```



```
binnedplot(fitted(mod),
           residuals(mod, type = "response"),
           nclass = NULL,
           xlab = "Expected Values",
           ylab = "Average residual",
           main = "Binned residual plot",
           cex.pts = 0.8,
           col.pts = 1,
           col.int = "gray")
```


Binned residual plot
Average residual
0.15
0.05
−0.05
−0.15
0.0
0.2
0.4
0.6
0.8
1.0
Expected Values