MAST 729 - Assignment #1

Francis Frégeau

Winter, 2021

Data description

The data were from the Medical Expenditure Panel Survey (MEPS), conducted by the U.S. Agency of Health Research and Quality. MEPS is a probability survey that provides nationally representative estimates of health care use, expenditures, sources of payment, and insurance coverage for the U.S. civilian population. This survey collects detailed information on individuals of each medical care episode by type of services including physician office visits, hospital emergency room visits, hospital outpatient visits, hospital inpatient stays, all other medical provider visits, and use of prescribed medicines. This detailed information allows one to develop models of health care utilization to predict future expenditures.

A more precise description of the variables of the database can be found at (page 10-12):

https://instruction.bus.wisc.edu/jfrees/jfreesbooks/regression%20modeling/bookwebdec2010/DataDescriptions.pdf

Below is the code to load the database, called MEPS:

```
path="https://instruction.bus.wisc.edu/jfrees/jfreesbooks/regression%20modeling/bookwebdec2010/CSVData/
download.file(path, "downloadeddata.csv")
MEPS=read.csv("downloadeddata.csv")
```

The aim

The aim of this project is to study how the number of inpatient visits (COUNTIP) and outpatient visits (COUNTOP) depend on individual characteristics, that are: AGE, ANYLIMIT, COLLEGE, HIGHSCH,GENDER,MNHPOOR,insure,USC, UNEMPLOY,MANAGEDCARE,famsize,RACE,REGION,EDUC,MARISTAT,INC

Note that we will not consider the following variables: EXPENDIP, EXPENDOP.

Preliminary analysis

1.Provide a table of counts, a histogram, and summary statistics of COUNTIP (the number of inpatient visits). Note the shape of the distribution and the relationship between the sample mean and sample variance.

Answer: The variance is roughly 1.7 times the mean, so the variable is over-dispersed with regards to a Poisson distribution.

```
#####################################################################
##############  INIT                  ###############################
#####################################################################

library("data.table")
library("dplyr")
```

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

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
library("ggplot2")
library("gridExtra")
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library("stringr")
library("glmnet")
```

```
## Loading required package: Matrix

## Loaded glmnet 4.0-2
```

```
#Load file
if(!file.exists("downloadeddata.csv")){

    path <- "https://instruction.bus.wisc.edu/jfrees/jfreesbooks/regression%20modeling/bookwebdec2010/CS

    download.file(path, "downloadeddata.csv")
    MEPS <- read.csv("downloadeddata.csv")

} else {

    MEPS <- fread("downloadeddata.csv")

}

#########################################################################

#Drop trash, format factors
#Remove duplicates
#Replace NA values with factor "Missing"
```

```r
names(MEPS) <- toupper(names(MEPS))
duplicates <- names(MEPS)[which(grepl("1", names(MEPS)))]
f <- function(x){if(any(x == "")){x[which(x == "")] <- "MISSING"}
                    return(x)}


MEPS %>%
    .[, (c("EXPENDIP", "EXPENDOP")) := NULL] %>%
    .[, (duplicates) := NULL] %>%
    .[, (names(MEPS)[which(sapply(MEPS, is.character))]) := lapply(.SD, f), .SDcols = names(MEPS)[which

#Organize the frame
#Identify the binary variables
is_binary <- function(x){

    vals <- sort(unique(x))
    if(length(vals) == 2){

        return(all.equal(sort(unique(x)), c(0,1)))

    } else {

        return(FALSE)

    }

}

binary_variales <- names(MEPS)[which(sapply(MEPS, is_binary))]

convert_binary_to_factor <- function(x){

    out <- rep("Yes", length(x))
    out[which(x == 0)] <- "No"
    return(as.factor(out))

}

blank_string_to_NA <- function(x){

    x[which(x == "")] <- NA
    return(x)

}


#Convert characters to factor
factor_variables <- names(MEPS)[which(sapply(MEPS, is.character))]
MEPS %>%
    .[, (factor_variables) := lapply(.SD, blank_string_to_NA), .SDcols = factor_variables] %>%
    .[, (factor_variables) := lapply(.SD, as.factor), .SDcols = factor_variables] %>%
    .[, (binary_variales) := lapply(.SD, convert_binary_to_factor), .SDcols = binary_variales]
```

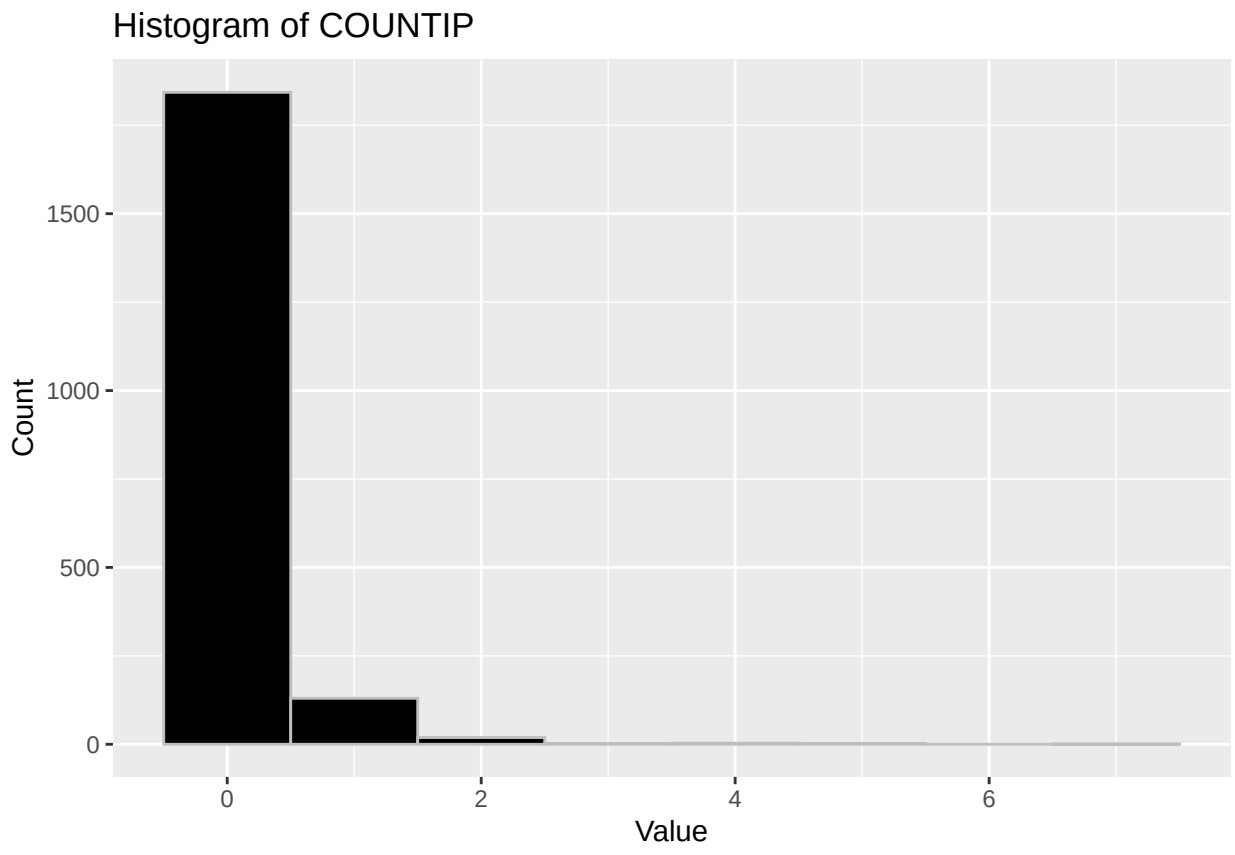
```
#Save backupframe for Q10
backup <- MEPS[, lapply(.SD, function(x){x}), .SDcols = names(MEPS)]



#Q1
print("Table of COUNTIP:", quote = FALSE)
```

```
## [1] Table of COUNTIP:
```

```
print(table(MEPS$COUNTIP))
```

```
##
##    0    1    2    3    4    5    7
## 1843  130   19    2    3    2    1
```

```
COUNTIP_histogram_frame <- MEPS[, c("COUNTIP"), with = FALSE]
COUNTIP_histogram <- ggplot(COUNTIP_histogram_frame, aes(x = COUNTIP)) + geom_histogram(color="grey",
                                                                                        fill="black",
                                                                                        binwidth = 1) +
                                                                         ggtitle("Histogram of COUNTIP") +
                                                                         xlab("Value") +
                                                                         ylab("Count")


print(COUNTIP_histogram)
```



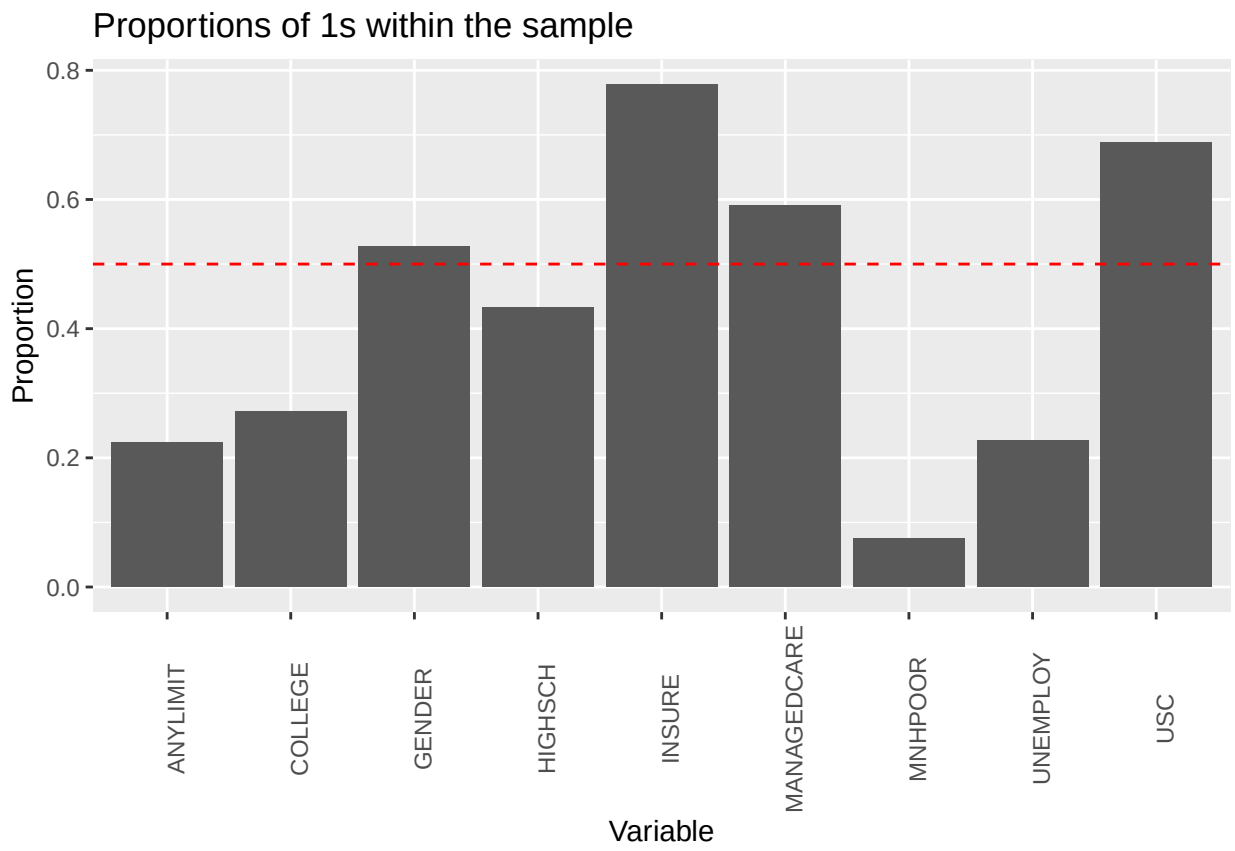
```
COUNTIP_stats <- c(mean(MEPS$COUNTIP),
                   var(MEPS$COUNTIP))
COUNTIP_stats <- as.data.table(t(COUNTIP_stats))
colnames(COUNTIP_stats) <- c("Mean", "Variance")
COUNTIP_stats[, ("Variance/Mean") := Variance / Mean]
print(COUNTIP_stats)
```

```
##      Mean  Variance Variance/Mean
## 1: 0.1015 0.1752854       1.72695
```

2.For each of the binary variables in the database, provide the proportion of patients having value 0 or 1. For instance, for "COLLEGE", determine the composition of patients with/without a college degree.

Answer: See code.

```
props <- MEPS[, lapply(.SD, function(x){sum(x == "Yes") / length(x)}), .SDcols = binary_variales]

props_plot_frame <- as.data.frame(t(props))
colnames(props_plot_frame) <- "Proportion"
props_plot_frame$Variable <- rownames(props_plot_frame)

props_plot <- ggplot(props_plot_frame, aes(x = Variable, y = Proportion)) + geom_bar(stat="identity") +
                                                                             theme(axis.text.x = element_
                                                                             ggtitle("Proportions of 1s
                                                                             geom_hline(yintercept = 0.5


print(props_plot)
```

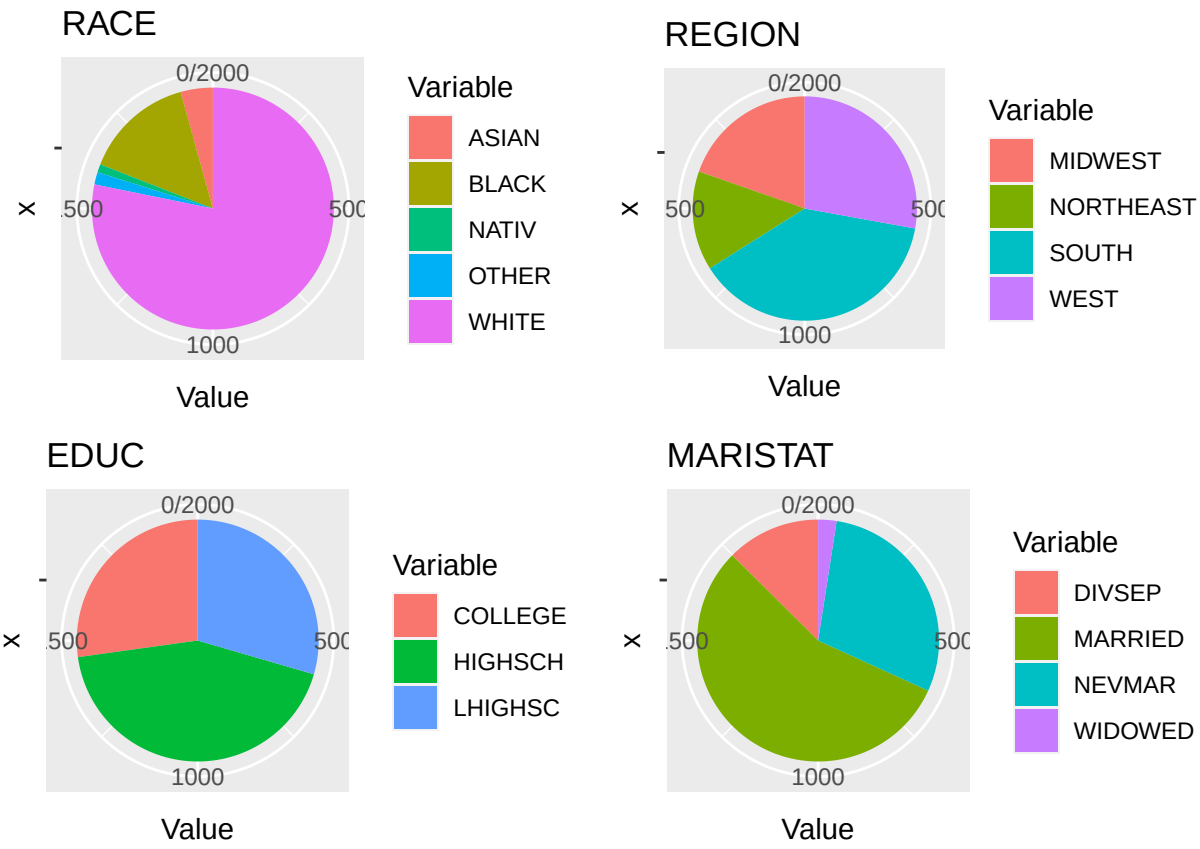


3.For each of the other categorical variables, use pie chart to provide the breakdown of the patients taking different values of this variable.

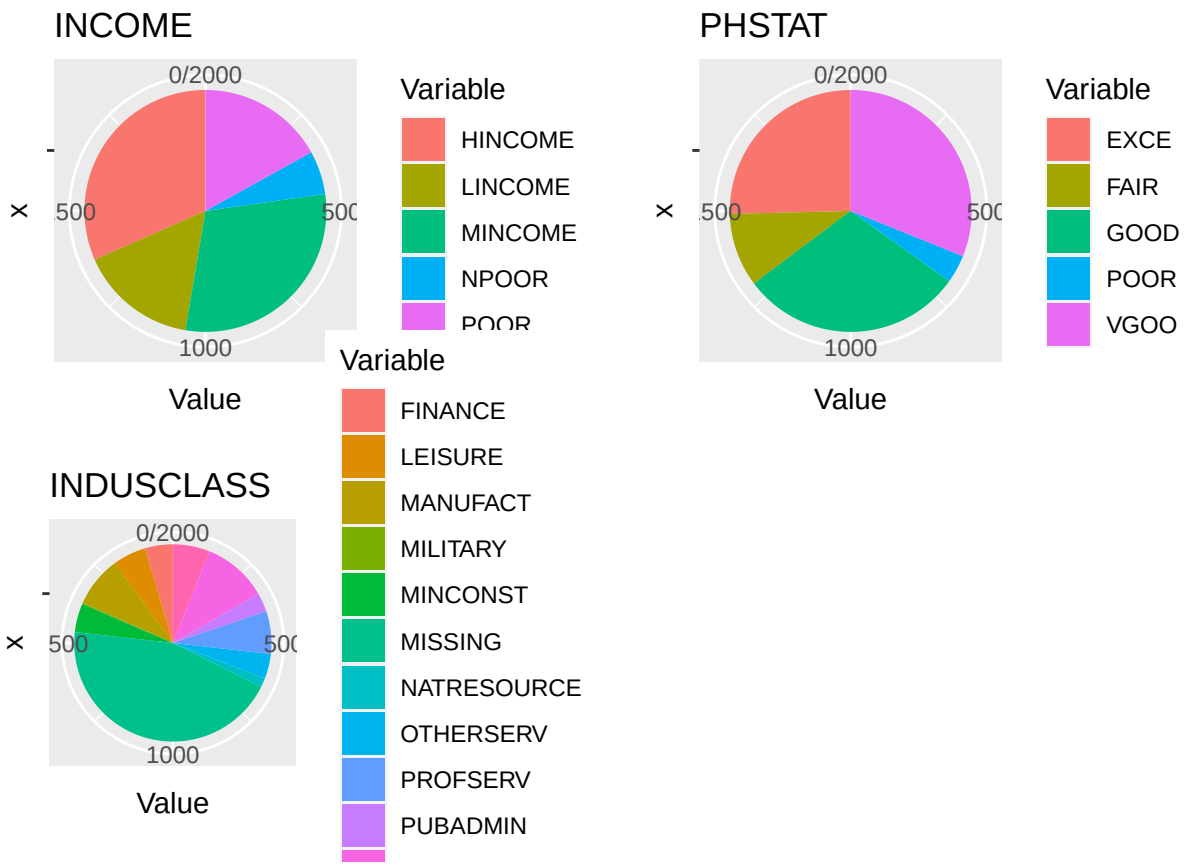
Answer: See code.

```
pie_chart <- function(x){

    if(any(is.na(x))){

        x <- as.character(x)
        x[which(is.na(x))] <- "MISSING"
        x <- as.factor(x)

    }
    values <- sort(unique(x))
    counts <- lapply(values, function(a){length(which(x == a))})


    pie_frame <- as.data.frame(values)
    colnames(pie_frame) <- "Variable"
    pie_frame$Value <- counts

    chart <- ggplot(pie_frame, aes(x="", y = Value, fill = Variable))+
            geom_bar(width = 1, stat = "identity") +
            coord_polar("y", start=0)

    return(chart)
```

```
}

pie_plots <- lapply(MEPS[, factor_variables, with = FALSE], pie_chart)
for(i in 1:length(pie_plots)){pie_plots[[i]] <- pie_plots[[i]] + ggtitle(factor_variables[i])}

gridExtra::grid.arrange(grobs = pie_plots[1:4], ncol=2, nrow=2)
```




```
gridExtra::grid.arrange(grobs = pie_plots[5:length(pie_plots)], ncol = 2, nrow=2)
```

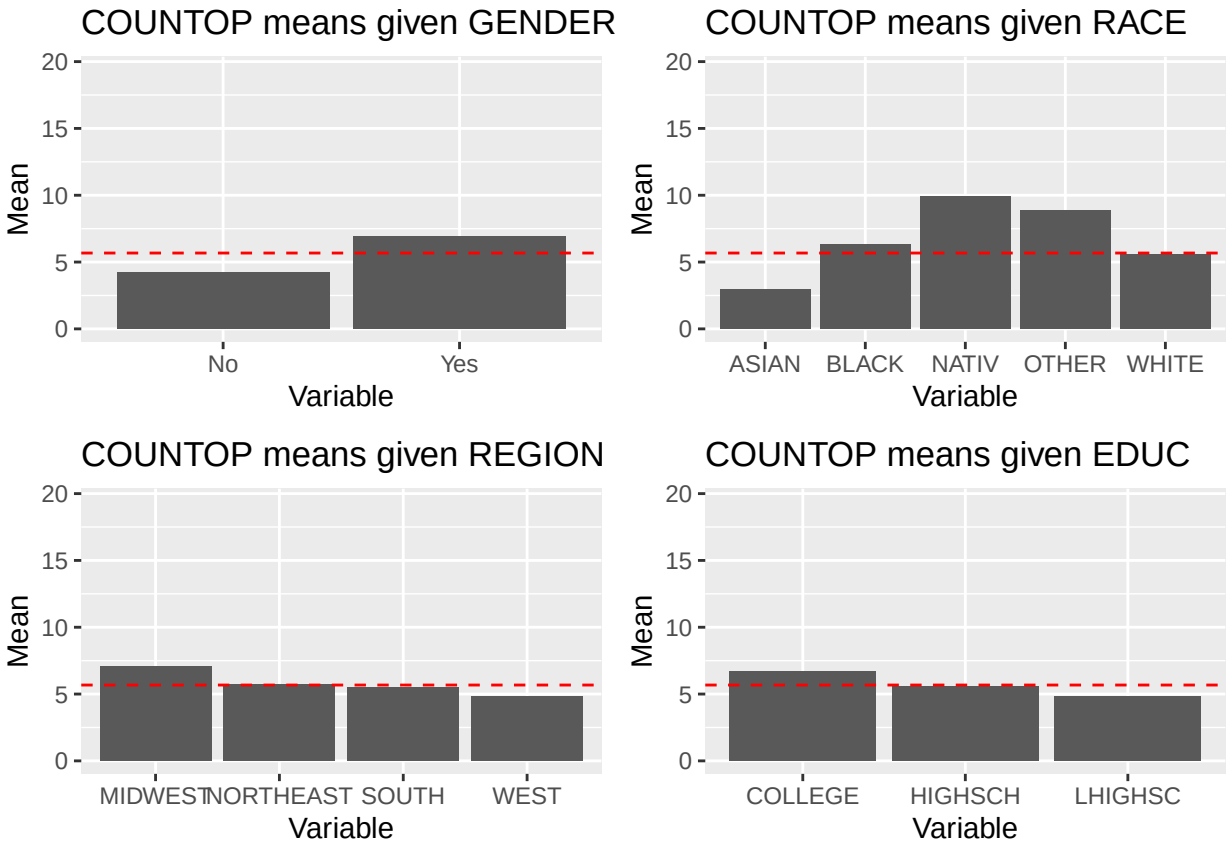


4.Create tables of means of COUNTOP by level of GENDER, ethnicity, region, education, self-rated physical health, self-rated mental health activity limitation, income, and insurance. Do the tables suggest that the explanatory variables have an impact on COUNTOP?

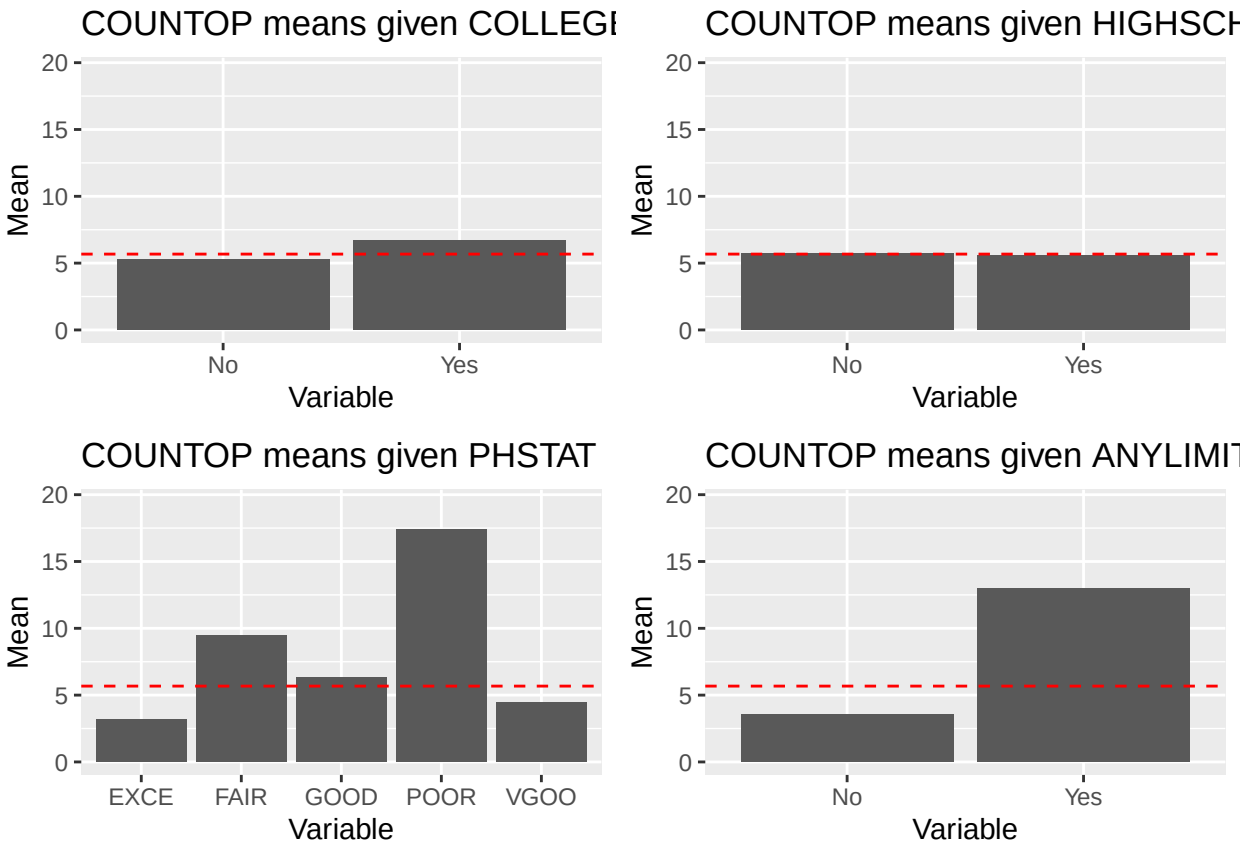
Answer: the tables do suggest that some of the variables seem to have an effect on COUNTOP.

```
mean_by <- function(x){

    out <- MEPS[, lapply(.SD, mean), .SDcols = "COUNTOP", by = x]
    names(out)[2] <- paste(names(out)[2], "_Mean", sep = "")
    return(out)

}
variables <- c("GENDER", "RACE", "REGION", "EDUC", "COLLEGE", "HIGHSCH", "PHSTAT", "ANYLIMIT", "INCOME"
tables_of_means <- lapply(as.list(variables), mean_by)

tables_of_means_max <- max(unlist(lapply(tables_of_means, function(x){max(x$COUNTOP_Mean)})))

bar_plots <- function(x){

    title <- paste("COUNTOP means given", names(x)[1])
    names(x)[1] <- "Variable"
    names(x)[2] <- "Mean"

    out <- ggplot(x, aes(x = Variable, y = Mean)) + geom_bar(stat="identity") +
                                                                    ggtitle(title) +
                                                                    geom_hline(yintercept = mean(ME
```

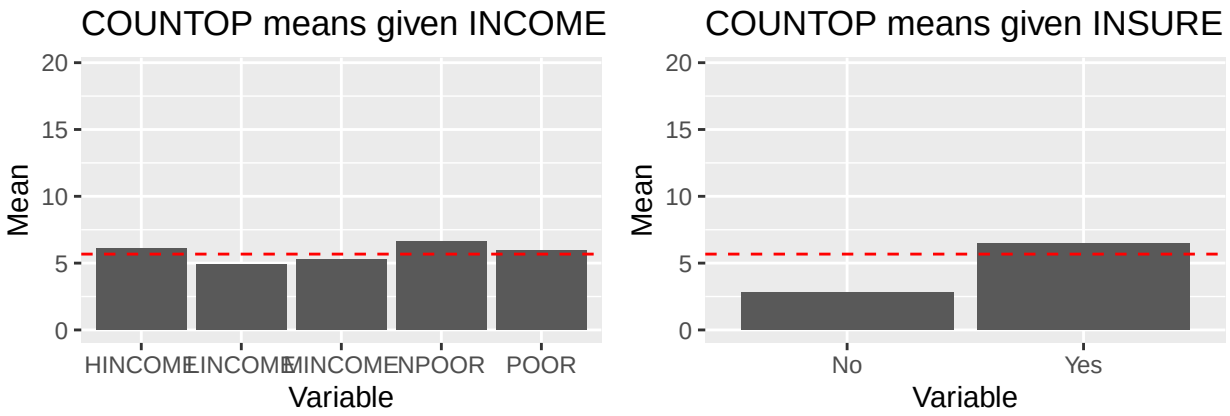
```
                                                                ylim(0, tables_of_means_max + 2)

    return(out)

}

tables_of_means_barplots <- lapply(tables_of_means, bar_plots)
gridExtra::grid.arrange(grobs = tables_of_means_barplots[1:4], ncol=2, nrow=2)
```




```
gridExtra::grid.arrange(grobs = tables_of_means_barplots[5:8], ncol=2, nrow=2)
```




```
gridExtra::grid.arrange(grobs = tables_of_means_barplots[9:length(tables_of_means_barplots)], ncol=2, n
```



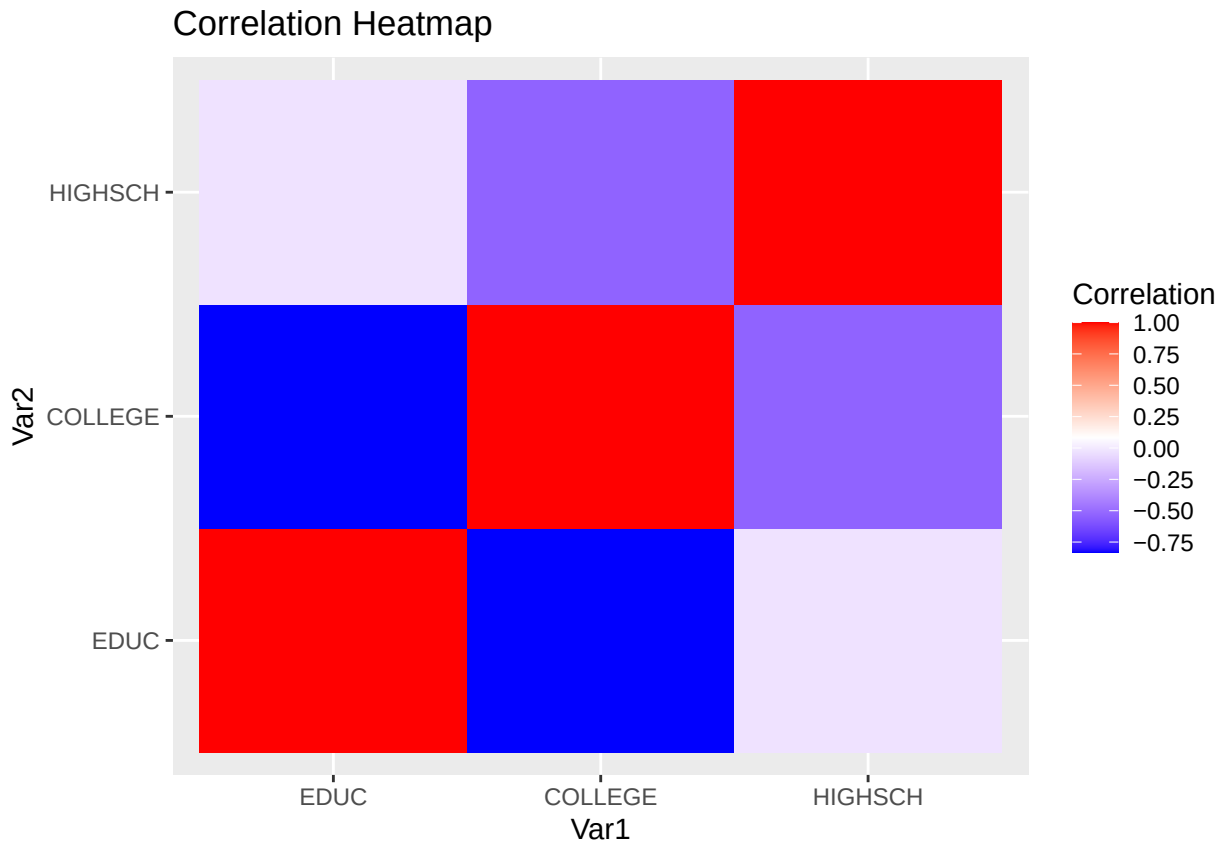
5.Show that there is a colinearity relationship between the variable "EDUC", "COLLEGE", "HIGHSCH". Hence in the following, we will not use the variable "EDUC".

Answer: See code.

```
numerical_vars <- MEPS[, lapply(.SD, as.numeric), .SDcols = c("EDUC", "COLLEGE", "HIGHSCH")]
numerical_vars_cor <- cor(numerical_vars)

print(round(numerical_vars_cor, 2))
```

```
##          EDUC COLLEGE HIGHSCH
## EDUC     1.00   -0.83   -0.03
## COLLEGE -0.83    1.00   -0.53
## HIGHSCH -0.03   -0.53    1.00
```

```
tile <- expand.grid(numerical_vars_cor)
names <- expand.grid(x = rownames(numerical_vars_cor), y = colnames(numerical_vars_cor))
tile <- as.data.table(cbind(tile, names))
names(tile) <- c("Correlation", "Var1", "Var2")

cor_plot <- ggplot(tile, aes(x = Var1, y = Var2, fill = Correlation)) + geom_tile() +
                                                                        ggtitle("Correlation Heatmap") +
                                                                        scale_fill_gradientn(colours = c
                                                                                             breaks = seq(-1, 1,


cor_plot
```




```
#Drop EDUC
MEPS[, EDUC := NULL]
variables <- variables[-which(variables == "EDUC")]
```

Poisson regression

6.Estimate a first model which includes all the explanatory variables mentioned above. Compute the AIC and BIC.

Answer: AIC = 1268.053, BIC = 1497.690

```
COUNTOP <- MEPS[, c("COUNTOP"), with = FALSE]
MEPS[, COUNTOP := NULL]

variables <- names(MEPS)
poisson_model_Q6 <- glm(COUNTIP ~., MEPS, family = poisson())
poisson_model_Q6_summary <- summary(poisson_model_Q6)

aic <- poisson_model_Q6_summary$aic

answer_table <- as.data.frame(aic)
colnames(answer_table) <- c("AIC")

p <- nrow(MEPS) - poisson_model_Q6_summary$df.residual
LL <- (aic - 2*p) / (-2)
```

```
answer_table$BIC <- p*log(nrow(MEPS)) - 2*LL
print(answer_table)
```

```
##        AIC     BIC
## 1 1268.053 1497.69
```

7.We can check that for "REGION" (resp. "PHSTAT", "MARISTAT", "INDUSCLASS"), only the coefficient before "WEST" (resp. "POOR", "NEVMAR","TRANSINFO") is significant. Replace these three categorical variables by three dummy variables. Remove also other variables whose coefficients are not significant at 1 % level. Re-estimate the Poisson regression model. Compute the AIC and BIC.

Answer: AIC = 1240.245, BIC = 1296.254

```
model_Q6_coefs <- as.data.table(poisson_model_Q6_summary$coefficients)


split_name <- function(x){

    matches <- sapply(variables, function(a){grepl(a, x, fixed = TRUE)})
    matches <- apply(matches, 1, function(a){
                                                k <- which(a)
                                                if(length(k) == 0){return(NA)} else {return(k[length(k)]
                                            })

    work_with <- which(!is.na(matches))
    matches <- matches[work_with]
    var_names <- variables[matches]

    x[work_with] <- sapply(c(1:length(var_names)), function(a){

                                                        suffix <- str_remove(x[work_with][a], var_names
                                                        return(paste(var_names[a], "_=_", suffix, sep =

        })

    return(x)

}

#Format the names of the coefficients, and print the rounded p-values
model_Q6_coefs %>%
    .[, Variable := rownames(poisson_model_Q6_summary$coefficients)] %>%
    .[, Variable := lapply(.SD, split_name), .SDcols = c("Variable")] %>%
    .[, (names(model_Q6_coefs)[1:3]) := NULL] %>%
    .[, P_val := lapply(.SD, function(x){x}), .SDcols = c("Pr(>|z|)")] %>%
    .[, ("Pr(>|z|)") := NULL] %>%
    .[, P_val := round(P_val, 4)] %>%
    .[, Coef := coef(poisson_model_Q6)]

print(model_Q6_coefs)
```

```
##                     Variable  P_val          Coef
```

```
##  1:              (Intercept) 0.0001  -3.288554171
##  2:                    AGE_=_ 0.0267  -0.015194178
##  3:           ANYLIMIT_=_Yes 0.0138   0.454807838
##  4:            COLLEGE_=_Yes 0.9366  -0.017856176
##  5:            HIGHSCH_=_Yes 0.5189  -0.112591444
##  6:             GENDER_=_Yes 0.0452   0.334791757
##  7:            MNHPOOR_=_Yes 0.4634  -0.164737751
##  8:             INSURE_=_Yes 0.0002   1.184646643
##  9:                USC_=_Yes 0.2784   0.221009477
## 10:           UNEMPLOY_=_Yes 0.0668   0.378234039
## 11:        MANAGEDCARE_=_Yes 0.5670   0.100155090
## 12:              FAMSIZE_=_ 0.3654  -0.046681685
## 13:            RACE_=_BLACK 0.4660   0.356527397
## 14:            RACE_=_NATIV 0.5219   0.477211510
## 15:            RACE_=_OTHER 0.9187  -0.077271742
## 16:            RACE_=_WHITE 0.9882   0.006896122
## 17:       REGION_=_NORTHEAST 0.3986  -0.198312575
## 18:           REGION_=_SOUTH 0.5647  -0.108248747
## 19:            REGION_=_WEST 0.0041  -0.682589626
## 20:       MARISTAT_=_MARRIED 0.3881  -0.179829434
## 21:        MARISTAT_=_NEVMAR 0.0102  -0.608066588
## 22:       MARISTAT_=_WIDOWED 0.1066  -0.737661113
## 23:         INCOME_=_LINCOME 0.3027   0.268454987
## 24:         INCOME_=_MINCOME 0.5091   0.139920565
## 25:           INCOME_=_NPOOR 0.7652   0.108120774
## 26:            INCOME_=_POOR 0.3253   0.263415937
## 27:            PHSTAT_=_FAIR 0.5735   0.180746523
## 28:            PHSTAT_=_GOOD 0.1023   0.387588790
## 29:            PHSTAT_=_POOR 0.0000   1.657554701
## 30:            PHSTAT_=_VGOO 0.1304   0.356350170
## 31:     INDUSCLASS_=_LEISURE 0.4208  -0.469261819
## 32:    INDUSCLASS_=_MANUFACT 0.4904  -0.332848303
## 33:    INDUSCLASS_=_MILITARY 0.9818 -12.512322948
## 34:    INDUSCLASS_=_MINCONST 0.3234   0.498312670
## 35:     INDUSCLASS_=_MISSING 0.7333   0.133049872
## 36: INDUSCLASS_=_NATRESOURCE 0.3573   0.635017629
## 37:   INDUSCLASS_=_OTHERSERV 0.2165  -0.985586592
## 38:    INDUSCLASS_=_PROFSERV 0.3641  -0.494833546
## 39:    INDUSCLASS_=_PUBADMIN 0.1192  -1.657333100
## 40:       INDUSCLASS_=_SALES 0.5880  -0.243909585
## 41:   INDUSCLASS_=_TRANSINFO 0.0618  -1.486996007
##                     Variable  P_val          Coef
```

```
#Filter dummy variables with p_val > 0.1


model_Q6_coefs_low_pval <- model_Q6_coefs[P_val > 0.1] %>%
                                                .[, P_val := round(P_val, 4)]



#Get p-values for factors as a whole
model_Q6_pval <- as.data.table(drop1(poisson_model_Q6))[-1, ] %>%
                    .[, Chi_Score := Deviance - poisson_model_Q6$deviance] %>%
```

```
                    .[, P_val := 1 - pchisq(Chi_Score, Df)]

model_Q6_pval[, names(model_Q6_pval) := lapply(.SD, function(x){round(x, 6)}), .SDcols = names(model_Q6

model_Q6_pval <- cbind(all.vars(formula(poisson_model_Q6)[-2]), model_Q6_pval)
names(model_Q6_pval)[1] <- "Variable"

print(model_Q6_pval)
```

```
##         Variable Df Deviance      AIC Chi_Score    P_val
##  1:          AGE  1 854.7420 1270.952  4.899070 0.026871
##  2:    ANYLIMIT  1 855.7617 1271.972  5.918714 0.014981
##  3:     COLLEGE  1 849.8493 1266.059  0.006340 0.936536
##  4:     HIGHSCH  1 850.2577 1266.468  0.414721 0.519583
##  5:      GENDER  1 853.9859 1270.196  4.142894 0.041810
##  6:     MNHPOOR  1 850.3891 1266.599  0.546123 0.459906
##  7:      INSURE  1 866.4126 1282.623 16.569613 0.000047
##  8:         USC  1 851.0591 1267.269  1.216154 0.270117
##  9:    UNEMPLOY  1 853.2613 1269.471  3.418332 0.064476
## 10: MANAGEDCARE  1 850.1742 1266.384  0.331176 0.564968
## 11:     FAMSIZE  1 850.6806 1266.891  0.837608 0.360082
## 12:        RACE  4 853.8979 1264.108  4.054901 0.398627
## 13:      REGION  3 860.0325 1272.243 10.189562 0.017022
## 14:    MARISTAT  3 858.8343 1271.044  8.991287 0.029407
## 15:      INCOME  4 851.1692 1261.379  1.326263 0.856904
## 16:      PHSTAT  4 890.5904 1300.801 40.747397 0.000000
## 17:  INDUSCLASS 11 871.7306 1267.941 21.887627 0.025258
```

```
#Drop the levels with low p-values
#I will set dropped variables to their reference (i.e.: the first factor of their class)
#This ammounts to dropping their one-hot-encoded column
#(Dropped variable, and reference level, are tagged as "A_Base")
to_trim <- unique(str_split(model_Q6_coefs_low_pval$Variable, "_", simplify = TRUE)[, 1])
f <- function(x){

    index <- which(str_split(model_Q6_coefs_low_pval$Variable, "_", simplify = TRUE)[, 1] == x)
    base_level <- levels(MEPS[, x, with = FALSE][[1]])[1]
    base_level <- unique(c(base_level, str_split(model_Q6_coefs_low_pval$Variable, "_", simplify = TRUE)

    return(base_level)

}
to_remove <- lapply(as.list(to_trim), f)

for(i in 1:length(to_trim)){

    if("" %in% to_remove[[i]]){next}
    setkeyv(MEPS, (to_trim[i]))
    MEPS %>%
        .[to_remove[[i]], (to_trim[i]) := "A_Base"] %>%
        .[, (to_trim[i]) := lapply(.SD, function(x){as.factor(as.character(x))}), .SDcols = to_trim[i]]

}
```

```
#Drop the variables with low p-values
rmv <- model_Q6_pval[P_val > 0.1, Variable]
MEPS[, (rmv) := NULL]




#Fit the new Poisson model
variables <- names(MEPS)
poisson_model_Q7 <- glm(COUNTIP ~., MEPS, family = poisson())
poisson_model_Q7_summary <- summary(poisson_model_Q7)

aic <- poisson_model_Q7_summary$aic

answer_table_Q7 <- as.data.frame(aic)
colnames(answer_table_Q7) <- c("AIC")

p <- nrow(MEPS) - poisson_model_Q7_summary$df.residual
LL <- (aic - 2*p) / (-2)

answer_table_Q7$BIC <- p*log(nrow(MEPS)) - 2*LL
answer_table_Q7 <- rbind(answer_table, answer_table_Q7)
rownames(answer_table_Q7) <- c("Full Model", "Reduced Model")
print(answer_table_Q7)
```

```
##                     AIC      BIC
## Full Model    1268.053 1497.690
## Reduced Model 1240.245 1296.254
```

8.Provide an interpretation for the GENDER coefficient.

Answer: The Gender variable multiplies the predicted mean of COUNTIP by a factor of 1.49

```
model_Q7_coefs <- as.data.table(poisson_model_Q7_summary$coefficients)
model_Q7_coefs %>%
    .[, Variable := rownames(poisson_model_Q7_summary$coefficients)] %>%
    .[, Variable := lapply(.SD, split_name), .SDcols = c("Variable")] %>%
    .[, (names(model_Q7_coefs)[1:3]) := NULL] %>%
    .[, P_val := lapply(.SD, function(x){x}), .SDcols = c("Pr(>|z|)")] %>%
    .[, ("Pr(>|z|)") := NULL] %>%
    .[, P_val := round(P_val, 10)] %>%
    .[, Coef := coef(poisson_model_Q7)]

model_Q7_coefs_groupnames <- str_split(model_Q7_coefs$Variable, "_", simplify = TRUE)[, 1]

IsMale <- model_Q7_coefs[which(model_Q7_coefs_groupnames == "GENDER")] %>%
            .[, Multiplicative_Factor := exp(Coef)]

print(IsMale)
```

```
##        Variable       P_val     Coef Multiplicative_Factor
## 1: GENDER_=_Yes 0.009691049 0.401274              1.493726
```

9.Predict the number of inpatient visit for a maried, insured, employed female of age 30 who has no activity limitation, "Excellent" self-rated health status, and who lives in "WEST" and works in the "TRANSINFO" sector.

Answer: 0.017

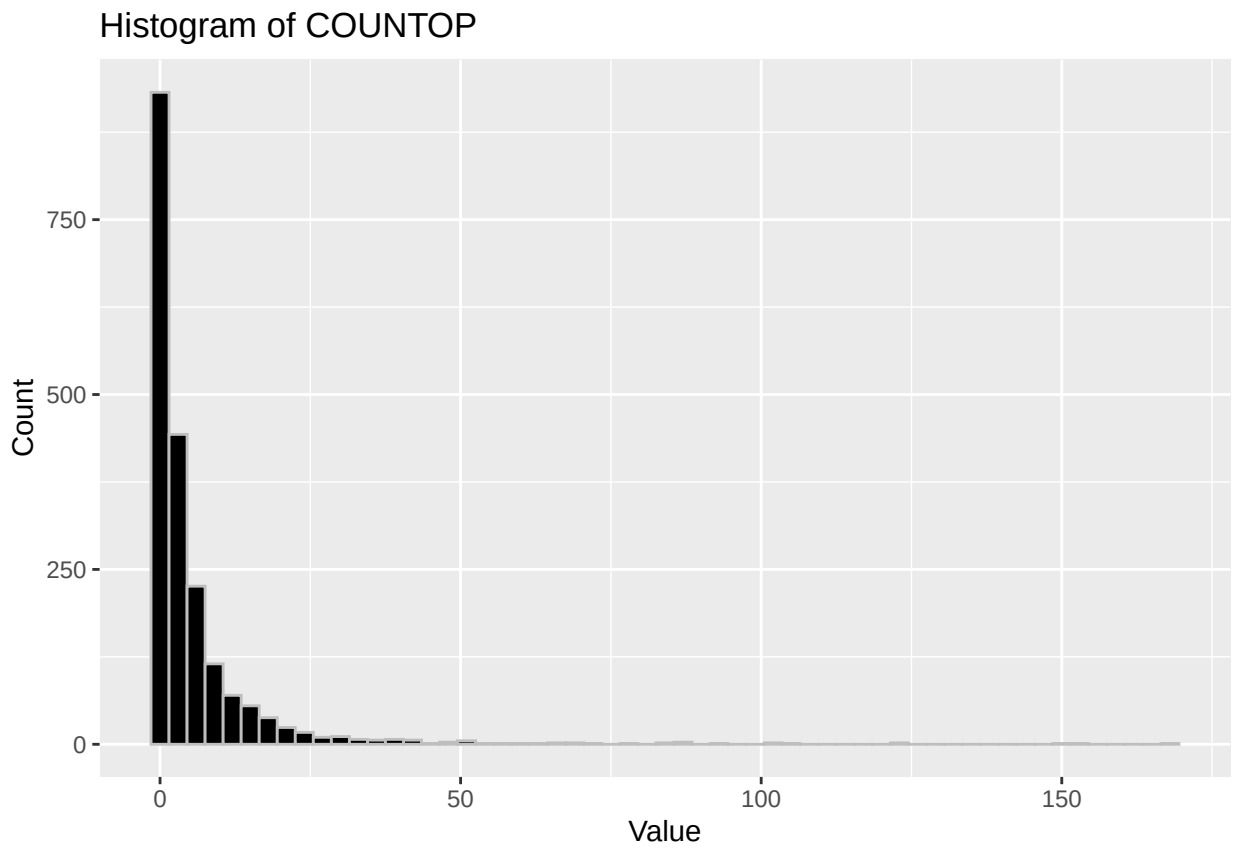
```
individual <- MEPS[1]
individual$MARISTAT <- "A_Base"
individual$INSURE <- "Yes"
individual$UNEMPLOY <- "No"
individual$AGE <- 30
individual$ANYLIMIT <- "No"
individual$PHSTAT <- "A_Base"
individual$REGION <- "WEST"
individual$INDUSCLASS <- "TRANSINFO"

answer <- predict(poisson_model_Q7, individual, type = "response")
print(paste("Predicted mean IP:", round(answer, 4)), quote = FALSE)
```

```
## [1] Predicted mean IP: 0.017
```

10.Estimate a regression model for variable "COUNTOP". Comment.

Answer: the variable is strongly over-dispersed with respect to a Poisson distribution, as the variance is roughly equal to 27.5 times the mean. First, a L1-penalized Poisson glm was fitted with glmnet, where the optimal regularization parameter (lambda) was computed using cross-validation. Then, a Quasi-Poisson glm was fitted using only the retained predictors. The dispertion parameter came out at 19.4, and the following variables were significant at a level of 0.05: ANYLIMIT = Yes, GENDER = Male, MNHPOOR = Yes, INSURE = Yes, USC = Yes, FAMSIZE.

```
#Use the original frame, since the MEPS one has been altered
#Remove COUNTIP
backup[, COUNTIP := NULL]

#Plot density
ggplot(backup, aes(x = COUNTOP)) + geom_histogram(color="grey",
                                                  fill="black",
                                                  binwidth = 3) +
                                   ggtitle("Histogram of COUNTOP") +
                                   xlab("Value") +
                                   ylab("Count")
```




```
COUNTOP_stats <- c(mean(backup$COUNTOP),
                   var(backup$COUNTOP))
COUNTOP_stats <- as.data.table(t(COUNTOP_stats))
colnames(COUNTOP_stats) <- c("Mean", "Variance")
COUNTOP_stats[, ("Variance/Mean") := Variance / Mean]
print(COUNTOP_stats)
```

```
##      Mean Variance Variance/Mean
## 1: 5.6745 156.2507      27.53558
```

```
#The variable is over-dispersed
#Use glmnet with L1 penalty for variable selection (1se rule)

X <- model.matrix(COUNTOP ~. , backup)
y <- as.matrix(backup$COUNTOP)
colnames(y) <- "COUNTOP"

lambda <- cv.glmnet(X, y, family = "poisson")

#Fit a regular GLM with the selected variable, but with quasi-poisson because of over-dispersion
#glmnet was only poisson, but the estimate for the mean is the same in both case, so it should be fine
X <- X[, which(coef(lambda)[-1] != 0)]
X <- as.data.table(cbind(X, y))

COUNTOP_model <- glm(COUNTOP ~., X, family = quasipoisson())
print(summary(COUNTOP_model))
```

```
##
## Call:
## glm(formula = COUNTOP ~ ., family = quasipoisson(), data = X)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -8.0100  -2.1900  -1.3745   0.1079  23.2706
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.226714   0.228788   0.991 0.321839
## AGE                 0.004812   0.003471   1.386 0.165820
## ANYLIMITYes         0.939877   0.095038   9.890  < 2e-16 ***
## COLLEGEYes          0.302812   0.091147   3.322 0.000909 ***
## GENDERYes           0.355550   0.088447   4.020 6.04e-05 ***
## MNHPOORYes          0.331060   0.124135   2.667 0.007717 **
## INSUREYes           0.383163   0.135128   2.836 0.004621 **
## USCYes              0.611612   0.126601   4.831 1.46e-06 ***
## FAMSIZE            -0.076164   0.027991  -2.721 0.006565 **
## PHSTATFAIR          0.330722   0.119879   2.759 0.005855 **
## PHSTATPOOR          0.361335   0.151796   2.380 0.017388 *
## INDUSCLASSMANUFACT -0.470460   0.205438  -2.290 0.022124 *
## INDUSCLASSMISSING   0.051165   0.091937   0.557 0.577913
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 18.59421)
##
##     Null deviance: 25325  on 1999  degrees of freedom
## Residual deviance: 17913  on 1987  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```