HW6

ASG

2023-03-29

3.1

a

```
library(AER) ; data(HousePrices) ; library(mgcv)
```

```
## Loading required package: car
```

```
## Loading required package: carData
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

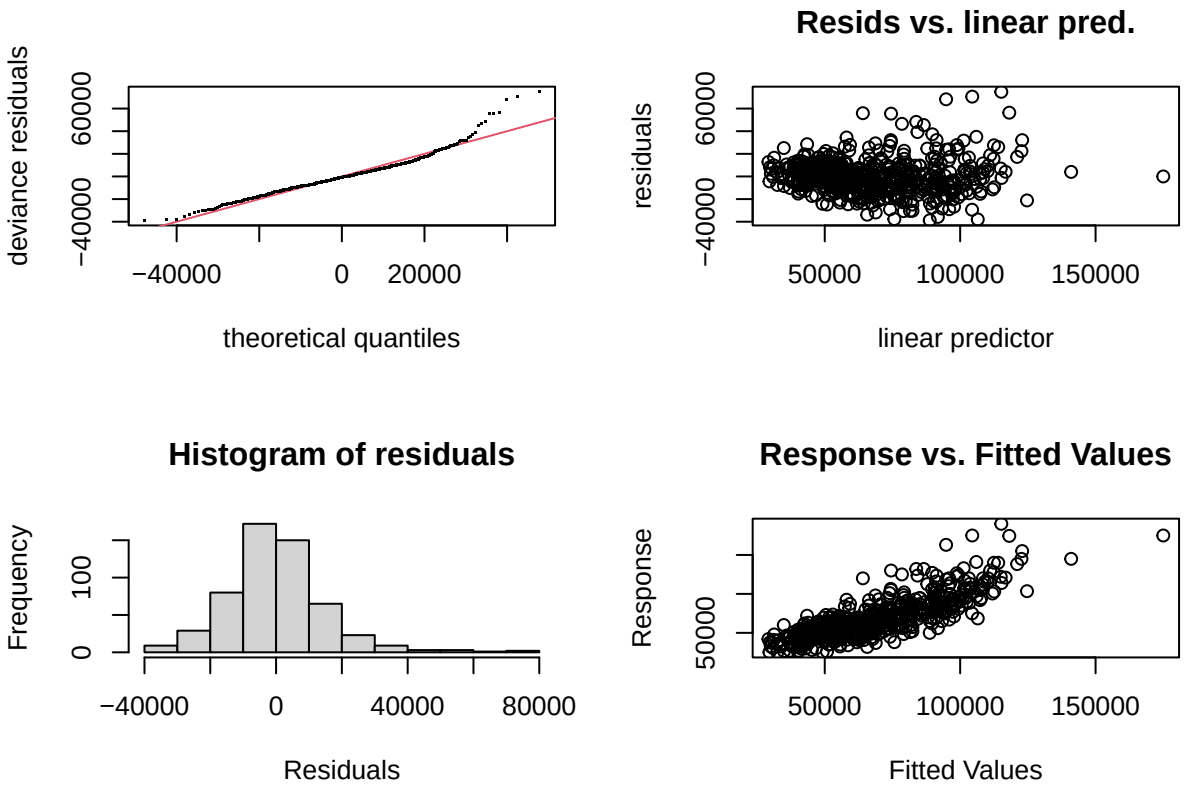
```
## Loading required package: nlme
```

```
## This is mgcv 1.8-42. For overview type 'help("mgcv-package")'.
```

```
fitGaussAM <- gam(price ~ s(lotsize,k = 27) + bedrooms + factor(bathrooms) + factor(stories) + factor(d
```

b

```
gam.check(fitGaussAM)
```



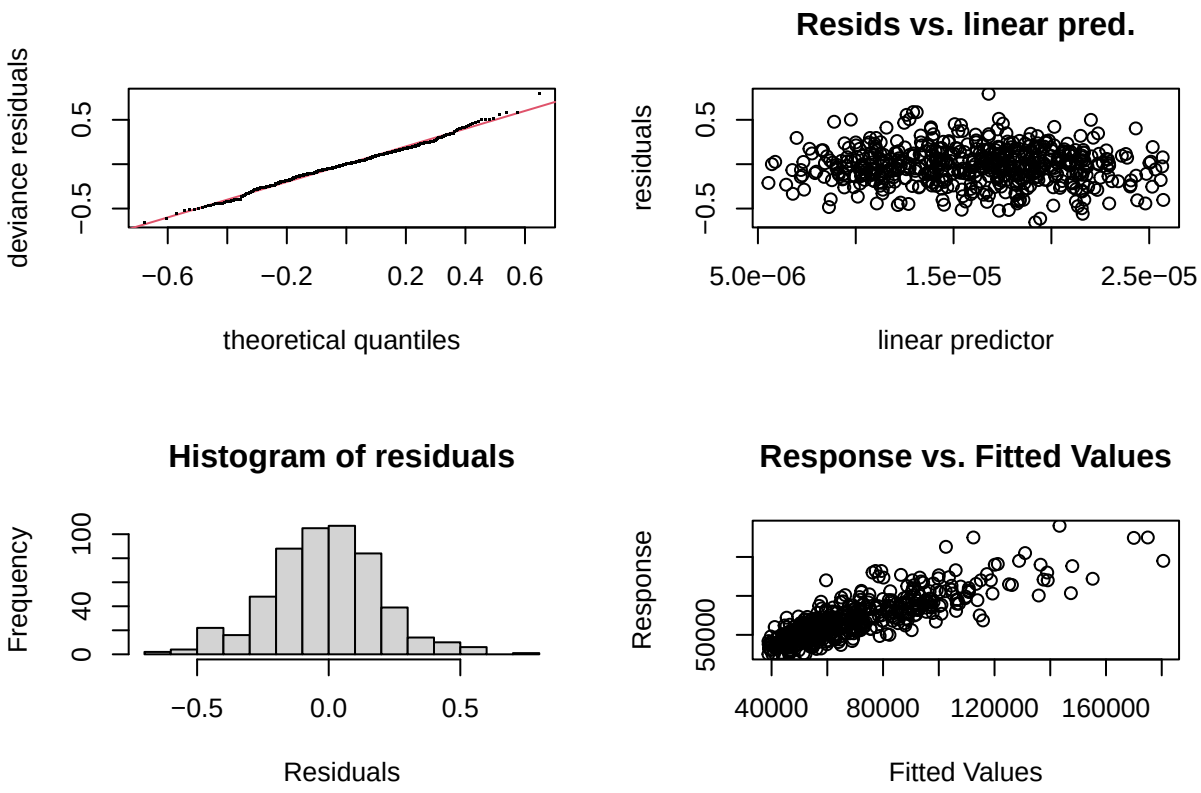

```
##
## Method: GCV   Optimizer: magic
## Smoothing parameter selection converged after 5 iterations.
## The RMS GCV score gradient at convergence was 84.05184 .
## The Hessian was positive definite.
## Model rank =  41 / 41
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##                k'   edf k-index p-value
## s(lotsize) 26.00  3.22    0.95    0.12
```

c

```
fitGammaAM <- gam(price ~ s(lotsize,k = 27) + bedrooms + factor(bathrooms) + factor(stories) + factor(di
```

d

```
gam.check(fitGammaAM)
```




```
##
## Method: GCV   Optimizer: outer newton
## full convergence after 4 iterations.
## Gradient range [1.14479e-11,1.14479e-11]
## (score 0.04663558 & scale 0.04516827).
## Hessian positive definite, eigenvalue range [0.0001882159,0.0001882159].
## Model rank =  41 / 41
##
## Basis dimension (k) checking results. Low p-value (k-index<1) may
## indicate that k is too low, especially if edf is close to k'.
##
##               k'   edf k-index p-value
## s(lotsize) 26.00  3.91    0.99    0.41
```

e

```
summary(fitGammaAM)
```

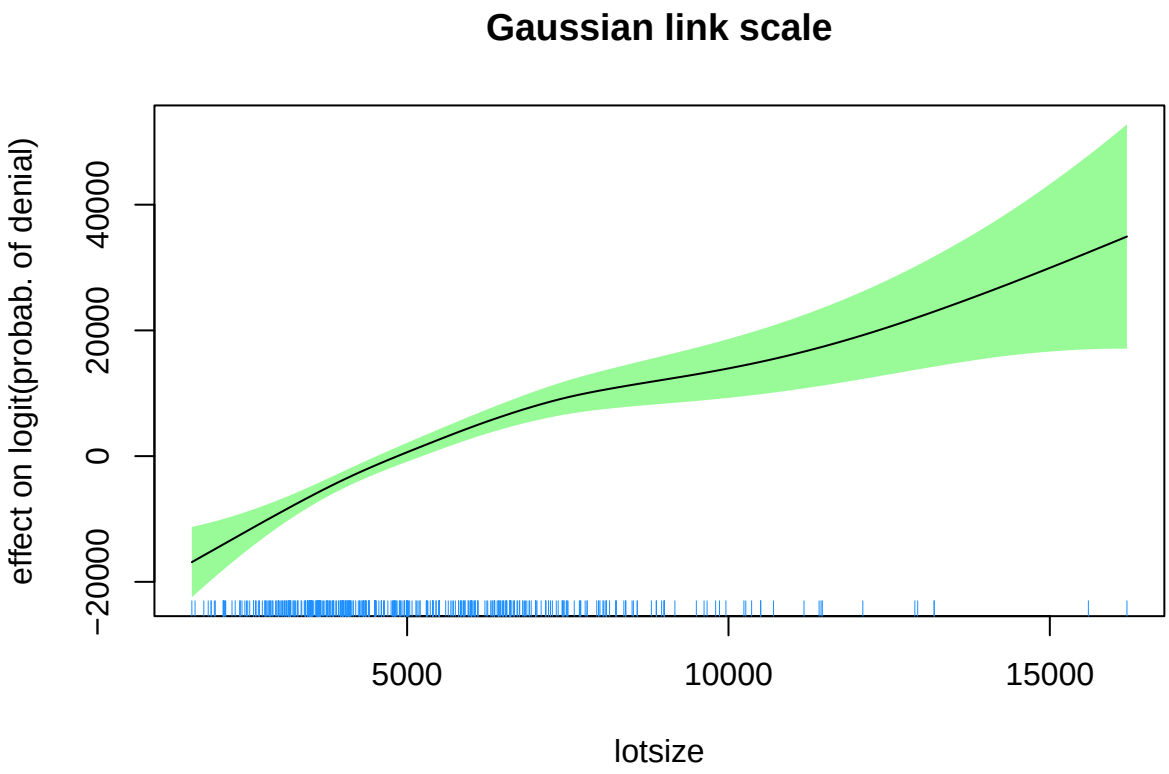
```
##
## Family: Gamma
## Link function: inverse
##
## Formula:
## price ~ s(lotsize, k = 27) + bedrooms + factor(bathrooms) + factor(stories) +
##     factor(driveway) + factor(recreation) + factor(fullbase) +
##     factor(gasheat) + factor(aircon) + garage + factor(prefer)
##
## Parametric coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            2.381e-05  8.087e-07  29.448  < 2e-16 ***
## bedrooms              -5.107e-07  2.236e-07  -2.284   0.0228 *
## factor(bathrooms)2    -2.223e-06  3.037e-07  -7.321 9.25e-13 ***
## factor(bathrooms)3    -3.306e-06  6.892e-07  -4.797 2.10e-06 ***
## factor(bathrooms)4    -3.656e-06  1.343e-06  -2.722   0.0067 **
## factor(stories)2      -1.542e-06  3.509e-07  -4.394 1.35e-05 ***
## factor(stories)3      -2.917e-06  5.264e-07  -5.542 4.72e-08 ***
## factor(stories)4      -3.084e-06  4.942e-07  -6.241 8.94e-10 ***
## factor(driveway)yes   -2.494e-06  5.497e-07  -4.538 7.04e-06 ***
## factor(recreation)yes -5.384e-07  3.274e-07  -1.644   0.1007
## factor(fullbase)yes   -1.669e-06  3.140e-07  -5.314 1.58e-07 ***
## factor(gasheat)yes    -2.512e-06  5.639e-07  -4.455 1.03e-05 ***
## factor(aircon)yes     -2.242e-06  2.977e-07  -7.530 2.22e-13 ***
## garage                -7.829e-07  1.589e-07  -4.927 1.12e-06 ***
## factor(prefer)yes     -1.381e-06  3.017e-07  -4.578 5.87e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df     F p-value
## s(lotsize) 3.907  4.933 25.12  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.688   Deviance explained =   69%
## GCV = 0.046636  Scale est. = 0.045168  n = 546
```

```
summary(fitGaussAM)
```

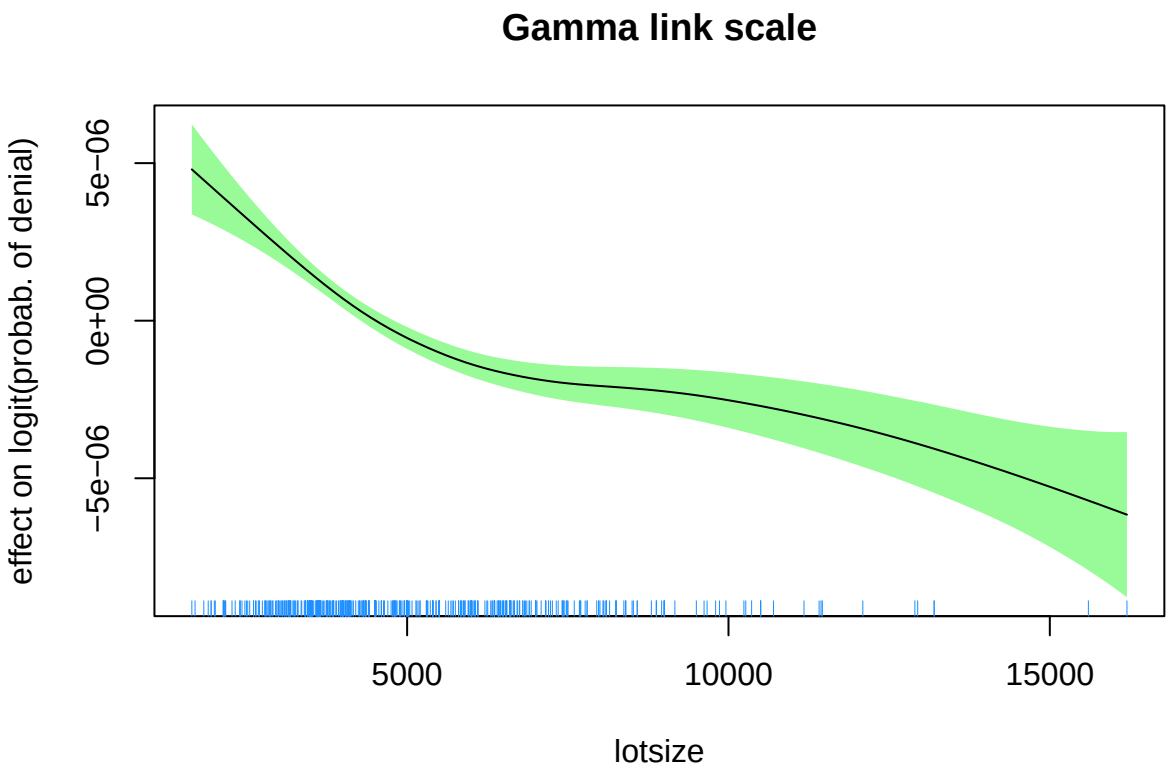
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## price ~ s(lotsize, k = 27) + bedrooms + factor(bathrooms) + factor(stories) +
##     factor(driveway) + factor(recreation) + factor(fullbase) +
##     factor(gasheat) + factor(aircon) + garage + factor(prefer)
##
## Parametric coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            35832.7     3550.2  10.093  < 2e-16 ***
## bedrooms                2102.8     1122.8   1.873 0.061638 .
## factor(bathrooms)2     13096.4     1735.0   7.548 1.95e-13 ***
```

```
## factor(bathrooms)3     29762.6     5108.0   5.827 9.84e-09 ***
## factor(bathrooms)4     68761.3    15758.7   4.363 1.54e-05 ***
## factor(stories)2        5715.0     1690.8   3.380 0.000778 ***
## factor(stories)3       12901.3     2903.8   4.443 1.08e-05 ***
## factor(stories)4       19730.6     3075.1   6.416 3.11e-10 ***
## factor(driveway)yes     5881.1     2074.3   2.835 0.004755 **
## factor(recreation)yes   3769.7     1926.7   1.957 0.050926 .
## factor(fullbase)yes     5843.3     1595.6   3.662 0.000275 ***
## factor(gasheat)yes     13018.6     3226.3   4.035 6.27e-05 ***
## factor(aircon)yes      11988.9     1582.8   7.575 1.62e-13 ***
## garage                  4097.8      852.7   4.806 2.01e-06 ***
## factor(prefer)yes       9713.8     1709.7   5.682 2.21e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##              edf Ref.df     F p-value
## s(lotsize) 3.218  4.103 24.75  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =   0.67   Deviance explained = 68.1%
## GCV = 2.4331e+08  Scale est. = 2.3519e+08  n = 546
```

```
plot(fitGaussAM,shade = TRUE,shade.col = "palegreen",
     select = 1,xlim = range(HousePrices$lotsize),ylab = "effect on logit(probab. of denial)",
     xlab = "lotsize",
     main = "Gaussian link scale",rug = FALSE)
rug(HousePrices$lotsize,col = "dodgerblue",quiet = TRUE)
```

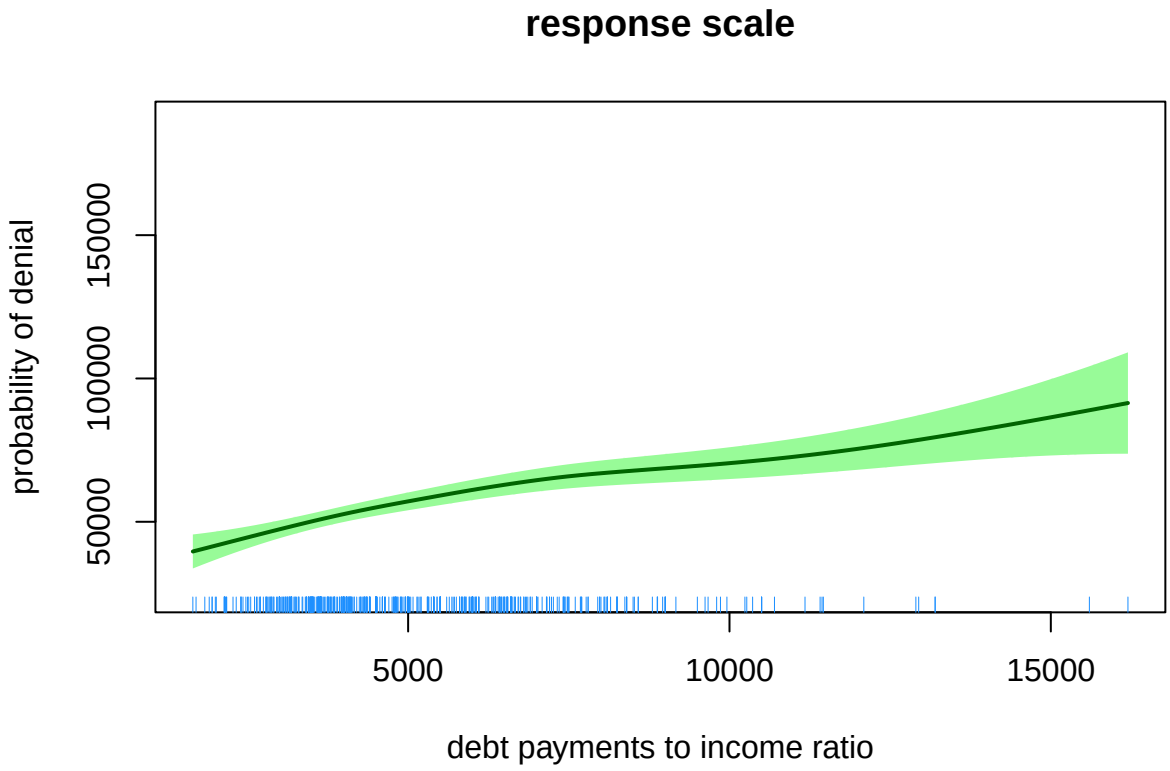



```
plot(fitGammaAM,shade = TRUE,shade.col = "palegreen",
     select = 1,xlim = range(HousePrices$lotsize),ylab = "effect on logit(probab. of denial)",
     xlab = "lotsize",
     main = "Gamma link scale",rug = FALSE)
rug(HousePrices$lotsize,col = "dodgerblue",quiet = TRUE)
```



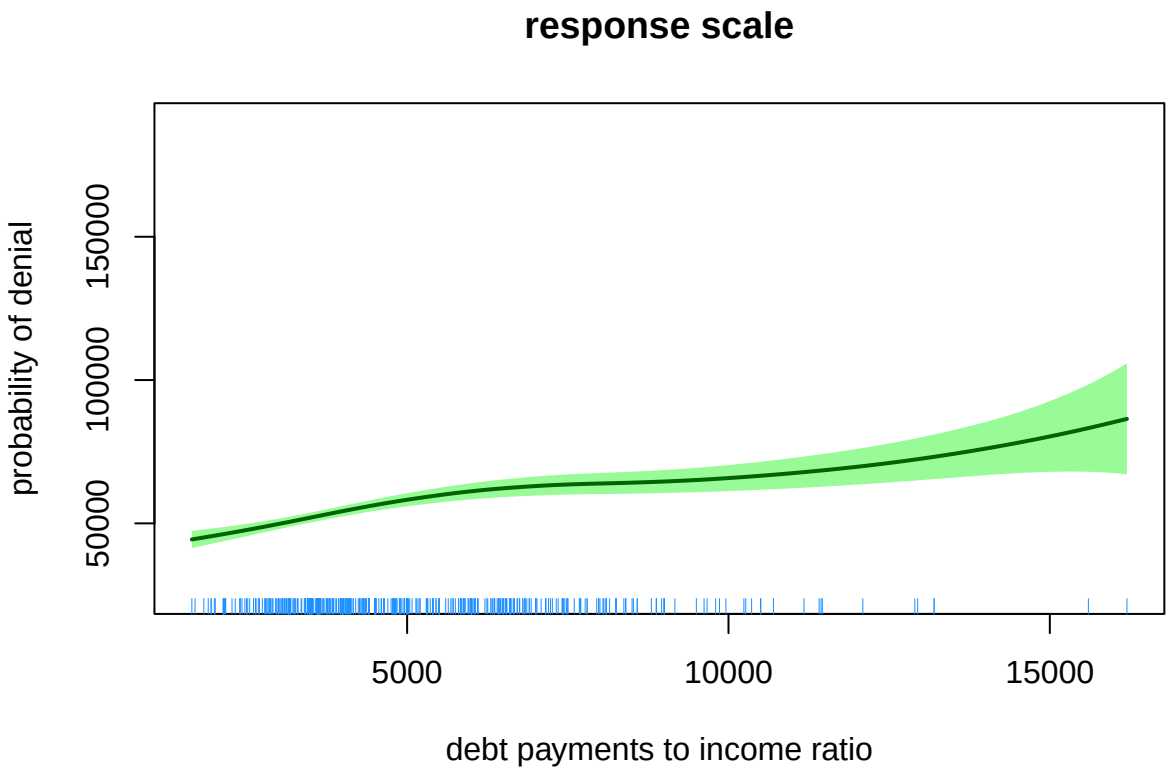

```
modalValue <- function(x)
   return(unique(x)[which.max(tabulate(match(x,unique(x))))])

# Set grids for `dir' and `lvg':

ng <- 401 ; dirg <- seq(min(HousePrices$lotsize),max(HousePrices$lotsize),length = ng) ; lvrg <- seq(0,

# Obtain and plot slice of the probability surface
# in the `dir' direction corresponding to the modal
# values of categorical predictors and the mean of
# other continuous predictors:



newdataDF <- data.frame(lotsize = dirg,
                        bedrooms=mean(HousePrices$bedrooms),
                        bathrooms=modalValue(HousePrices$bathrooms),
                        stories=modalValue(HousePrices$stories),
                        driveway=modalValue(HousePrices$driveway),
                        recreation=modalValue(HousePrices$recreation),
                        fullbase=modalValue(HousePrices$fullbase),
                        gasheat=modalValue(HousePrices$gasheat),
                        aircon=modalValue(HousePrices$aircon),
                        garage=mean(HousePrices$garage),
                        prefer=modalValue(HousePrices$prefer)
                        )
```

```
predObjdir <- predict(fitGaussAM,newdata = newdataDF,
                      type = "response",se.fit = TRUE)
etahatdirg <- predObjdir$fit
lowdirg <- etahatdirg - qnorm(0.975)*predObjdir$se.fit
uppdirg <- etahatdirg + qnorm(0.975)*predObjdir$se.fit
#lowdirg[lowdirg<0] <- 0 ; lowdirg[lowdirg>0.5] <- 0.5

plot(0,type = "n",ylim = range(HousePrices$price),xlim = range(HousePrices$lotsize),xlab  =  "debt payme
     ylab  =  "probability of denial",main  =  "response scale")
polygon(c(dirg,rev(dirg)),c(lowdirg,rev(uppdirg)),col = "palegreen",border = FALSE)
lines(dirg,etahatdirg,col = "darkgreen",lwd = 2)

rug(HousePrices$lotsize,col = "dodgerblue",quiet = TRUE)
```




```
predObjdir <- predict(fitGammaAM,newdata = newdataDF,
                      type = "response",se.fit = TRUE)
etahatdirg <- predObjdir$fit
lowdirg <- etahatdirg - qnorm(0.975)*predObjdir$se.fit
uppdirg <- etahatdirg + qnorm(0.975)*predObjdir$se.fit
#lowdirg[lowdirg<0] <- 0 ; lowdirg[lowdirg>0.5] <- 0.5

plot(0,type = "n",ylim = range(HousePrices$price),xlim = range(HousePrices$lotsize),xlab  =  "debt payme
     ylab  =  "probability of denial",main  =  "response scale")
polygon(c(dirg,rev(dirg)),c(lowdirg,rev(uppdirg)),col = "palegreen",border = FALSE)
lines(dirg,etahatdirg,col = "darkgreen",lwd = 2)
```

```
rug(HousePrices$lotsize,col = "dodgerblue",quiet = TRUE)
```



According to the summaries and the residual plots checked earlier, It can be seen that the Gamma model fit slightly outperforms the gaussian fit. It has a higher Rsquared value and a little higher deviance explained value.

f

```
newdataDF <- data.frame(lotsize = 5000,
                        bedrooms=3,
                        bathrooms=2,
                        stories=2,
                        driveway="yes",
                        recreation="no",
                        fullbase="yes",
                        gasheat="no",
                        aircon="no",
                        garage=2,
                        prefer="no"
                        )

predObjdir <- predict(fitGaussAM,newdata = newdataDF,
                      type = "response",se.fit = TRUE)
print(predObjdir)
```

```
## $fit
##        1
## 81492.52
##
## $se.fit
##        1
## 2572.305
```

g

```
etahatdirg <- predObjdir$fit
lowdirg <- etahatdirg - qnorm(0.975)*predObjdir$se.fit
uppdirg <- etahatdirg + qnorm(0.975)*predObjdir$se.fit
print(lowdirg)
```

```
##        1
## 76450.89
```

```
print(uppdirg)
```

```
##        1
## 86534.14
```

2

The IID assumption i.e Independence of Independent variable assumption is not mentioned in the textbook which is one of the assumption that we usually use in OLS regression. Especially for the GLM the iid assumption is relaxed for the predictors but still the residuals needs to be iid otherwise they will provide biased estimates.

3

a

Scaled Deviance of a model is essentially a measure of the "goodness of fit" of the data. It is often measured by comparing the deviance or the fit of the target model with that of the saturated model. The value of scaled deviance is equivalent to the $-2 * log(\frac{L(\theta_{MLE}/Y)}{L(\theta_S/y)})$

b

As seen from the above equation , when we subtract two scaled deviance of two different model, due to the logarithmic rules, the likelihood ration inside the two logs get multipled to the inverse of the other model such as :

$D0 - D1 = -2 * log(\frac{L(\theta_{10}/Y)}{L(\theta_S/y)} * \frac{L(\theta_S/y}{\theta/Y}) = \lambda$

Thus the likelihood of saturated model gets cancelled and we are left with the log likelihood ratio test statistic lambda.