Solution to Series 8

1. Logistic Regression for Binary Data

a) We fit a logistic regression for the binary variable purchase with predictors income and age:

```
> car <- read.table("http://stat.ethz.ch/Teaching/Datasets/car.dat",header=T)
> fit <- glm(purchase ~ income + age, data=car, family=binomial)
> summary(fit)

Call:
glm(formula = purchase ~ income + age, family = binomial, data = car)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6189  -0.8949  -0.5880   0.9653   2.0846

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.73931    2.10195  -2.255   0.0242 *
income       0.06773    0.02806   2.414   0.0158 *
age          0.59863    0.39007   1.535   0.1249
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 44.987  on 32  degrees of freedom
Residual deviance: 36.690  on 30  degrees of freedom
AIC: 42.69

Number of Fisher Scoring iterations: 4
```

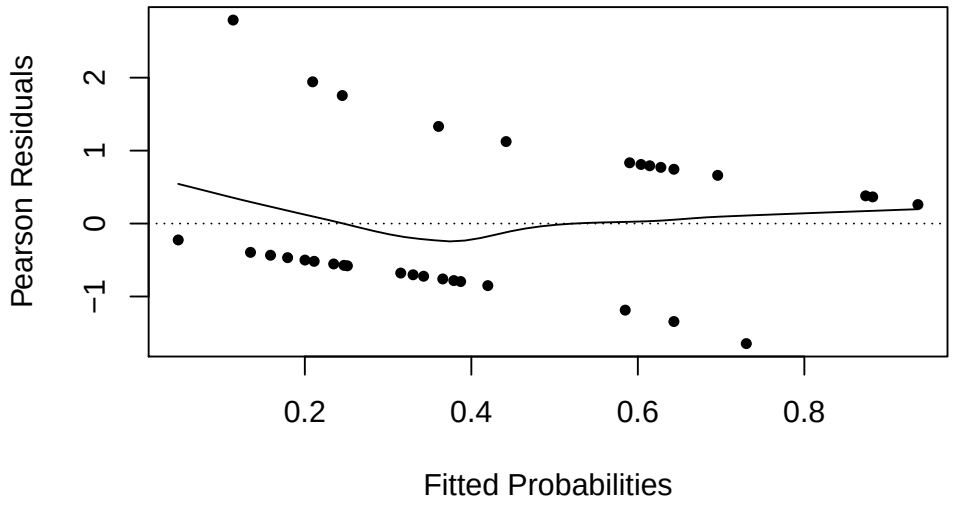
We can read off the coefficients from the regression output. The regression equation is:
$\log(\frac{\hat{p}}{1-\hat{p}}) = -4.74 + 0.068 \cdot income + 0.599 \cdot age$.

b) $\exp(\hat{\beta}_{income}) = e^{0.068} = 1.07$ and $\exp(\hat{\beta}_{age}) = e^{0.599} = 1.82$ are the relative changes of the odds of buying a new car for an increase of one unit in income and age respectively. I.e. the odds for buying a new car increase by 7% for each increase of income by 1000 USD. The odds increase by 82% for each additional year of age of the oldest car.

c)
```
> predict(fit, data.frame(age=3,income=50),type="response")
        1
0.6090245
```

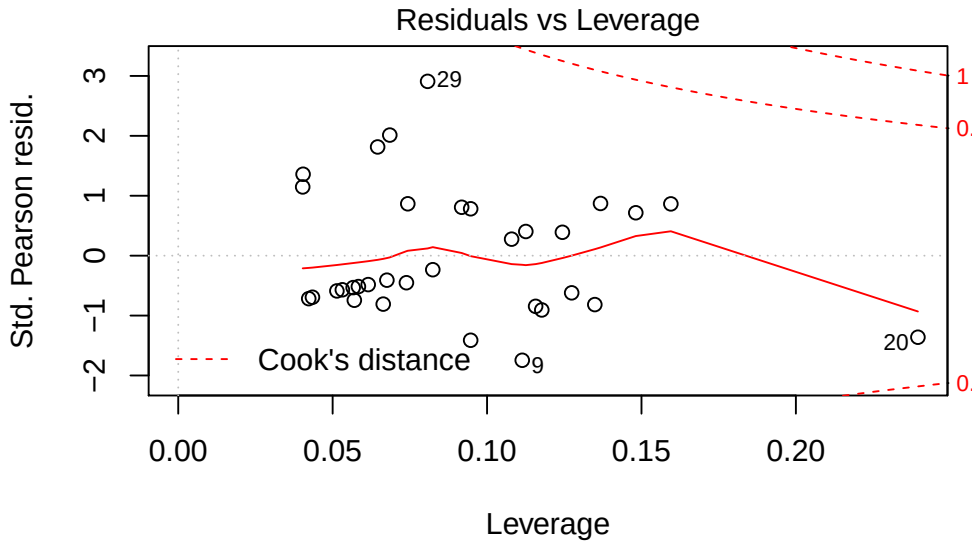
d) We first look at the Tukey-Anscombe plot with Pearson residuals:

```
> xx <- predict(fit, type="response")
> yy <- residuals(fit, type="pearson")
> scatter.smooth(xx, yy, family="gaussian", pch=20, xlab="Fitted Probabilities",
                 ylab="Pearson Residuals")
> abline(h=0, lty=3)
```



There is no evidence that assumptions aren't satisfied: the expectation does not seem to deviate much from zero. Also, most residuals have an absolute value that is smaller than 2.
Finally, we check for influential observations:

```
> plot(fit, which=5)
```



There seem to be no influential data points, just one outlier without influence.

e) We perform deviance-based significance tests using the function `drop1`:

```
> drop1(fit, test="Chisq")
Single term deletions

Model:
purchase ~ income + age
       Df Deviance    AIC    LRT Pr(>Chi)
<none>      36.690 42.690
income  1   44.987 48.987 8.2976  0.00397 **
age     1   39.305 43.305 2.6149  0.10586
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for age is quite high, so it might not be a significant predictor in this model. However, these tests are only approximate, so the results should not be overestimated.

f) We first fit a new model with an interaction term, and then perform an Anova:

```
> fit2 = glm(purchase ~ income + age + income:age, data=car, family=binomial)
> summary(fit2)
```

```
Call:
glm(formula = purchase ~ income + age + income:age, family = binomial,
    data = car)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6096  -0.8222  -0.5334   0.8731   1.9924

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.372993   2.862477  -0.829    0.407
income       0.001326   0.064770   0.020    0.984
age         -0.303860   0.890512  -0.341    0.733
income:age   0.028860   0.026493   1.089    0.276

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 44.987  on 32  degrees of freedom
Residual deviance: 35.404  on 29  degrees of freedom
AIC: 43.404

Number of Fisher Scoring iterations: 4
```

```
> anova(fit, fit2, test="Chisq")
```

```
Analysis of Deviance Table

Model 1: purchase ~ income + age
Model 2: purchase ~ income + age + income:age
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1        30     36.690
2        29     35.404  1   1.2855   0.2569
```

The p-value is large, so we cannot reject the null hypothesis that the two models are equal. Hence, there does not seem to be a significant interaction between income and age.

2. **a)**

```
> hyper.tbl <- cbind(n.hyper=n.hyper, n.nohyper=n.total-n.hyper)
> hyper.tbl
```

```
     n.hyper n.nohyper
[1,]       5        55
[2,]       2        15
[3,]       1         7
[4,]      35       152
[5,]      13        72
[6,]      15        36
[7,]       8        15
```

Note that the first column denotes the number of "successes", while the second column the number of "failures".

b)

```
> glm.hyp <- glm(hyper.tbl ~ smoking+obesity+snoring,family="binomial")
> summary(glm.hyp)
```

```
Call:
glm(formula = hyper.tbl ~ smoking + obesity + snoring, family = "binomial")

Deviance Residuals:
       1         2         3         4         5         6
 0.50780   0.10458   0.02847  -0.21903  -0.63361   0.32485
       7
 0.51753
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.6433     0.4090  -6.462 1.03e-10 ***
smokingYes    0.5488     0.3132   1.752  0.07976 .
obesityYes    0.6668     0.3455   1.930  0.05360 .
snoringYes    1.1184     0.3656   3.059  0.00222 **
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13.3181  on 6  degrees of freedom
Residual deviance:  1.0924  on 3  degrees of freedom
AIC: 34.011

Number of Fisher Scoring iterations: 4
```

Here, we model the expected value of the proportion of people with hypertension (using the logit function) as a function of the predictors `smoking`, `obesity`, and `snoring`.
Now, we use the residual deviance to assess the goodness-of-fit. Note that the number of observations in every batch is larger than 5. Therefore, the Chi-square test is valid in this case.

```
> pchisq(deviance(glm.hyp), df.residual(glm.hyp), lower=FALSE)
[1] 0.7789051
```

The Chi-square test for the Residual deviance gives a p-value larger than 0.05, so we can conclude that the model fits well.

c) First we perform a Chi-squared test for the Null deviance to check whether any of the predictors have an influence on the response variable:

```
> pchisq(glm.hyp$null.deviance-glm.hyp$dev,df=(glm.hyp$df.null-glm.hyp$df.res),lower=FALSE)
[1] 0.006648823
```

The p-value is smaller than 0.05, which tells us that there is at least one significant predictor in our model.
Now we do deviance based individual tests for each of the predictors:

```
> D<- drop1(glm.hyp,test="Chisq")
> D
Single term deletions

Model:
hyper.tbl ~ smoking + obesity + snoring
        Df Deviance    AIC     LRT Pr(>Chi)
<none>       1.0924 34.011
smoking  1   4.2010 35.120  3.1086  0.07788 .
obesity  1   4.8781 35.797  3.7857  0.05169 .
snoring  1  11.4062 42.325 10.3138  0.00132 **
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The predictor `smoking` does not have a significant influence on the response, the p-value for `obesity` is smaller but still not significant at the 5% level. The predictor `snoring` is significant.

d) First we exclude `smoking` from the model.

```
> glm.hyp2 <- glm(hyper.tbl ~ obesity+snoring,family="binomial")
> summary(glm.hyp2)

Call:
glm(formula = hyper.tbl ~ obesity + snoring, family = "binomial")

Deviance Residuals:
```

```
       1         2         3         4         5         6
-0.28404   0.32506  -0.44798   0.13068  -1.21440   1.52066
       7
-0.09844

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.2676     0.3121  -7.267 3.69e-13 ***
obesityYes    0.7745     0.3225   2.401   0.0163 *
snoringYes    0.9075     0.3240   2.801   0.0051 **
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 13.318  on 6  degrees of freedom
Residual deviance:  4.201  on 4  degrees of freedom
AIC: 35.12

Number of Fisher Scoring iterations: 4
```

```
> drop1(glm.hyp2,test="Chisq")
```

```
Single term deletions

Model:
hyper.tbl ~ obesity + snoring
        Df Deviance    AIC    LRT Pr(>Chi)
<none>       4.201 35.120
obesity  1   10.251 39.170 6.0503 0.013904 *
snoring  1   12.303 41.222 8.1021 0.004421 **
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We note that smoking was covering some of the explanatory power of obesity, which is now significant. Now we check again the goodness of fit of the model without the predictor smoking.

```
> pchisq(deviance(glm.hyp2), df.residual(glm.hyp2), lower=FALSE)
```

```
[1] 0.3794888
```

```
> pchisq(glm.hyp2$null.deviance,glm.hyp2$df.null,lower=FALSE)
```

```
[1] 0.03825404
```

The model without the predictor smoking fits sufficiently well. Moreover, the result of the Chi-squared test for the Null deviance and both deviance based individual tests are significant. Therefore, we only include the variables obesity and snoring in our model.

e)

```
> fitted(glm.hyp2)-n.hyper/n.total
           1            2            3            4
 0.010508367 -0.023805358  0.058457380 -0.003708396
           5            6            7
 0.051280802 -0.089895669  0.009817867
```

```
> data.frame(fit=fitted(glm.hyp2) * n.total, n.hyper, n.total)
        fit n.hyper n.total
1  5.630502       5      60
2  1.595309       2      17
3  1.467659       1       8
4 34.306530      35     187
5 17.358868      13      85
6 10.415321      15      51
7  8.225811       8      23
```

3. Poisson Regression

a) Since we have discrete count data (and an unknown maximum), we fit a Poisson regression model. I.e. we model the logarithm of the rate λ as a linear function of the predictors. We start with the categorical variable `sample` as the only predictor, i.e. we estimate the rate in each of the three samples batches.

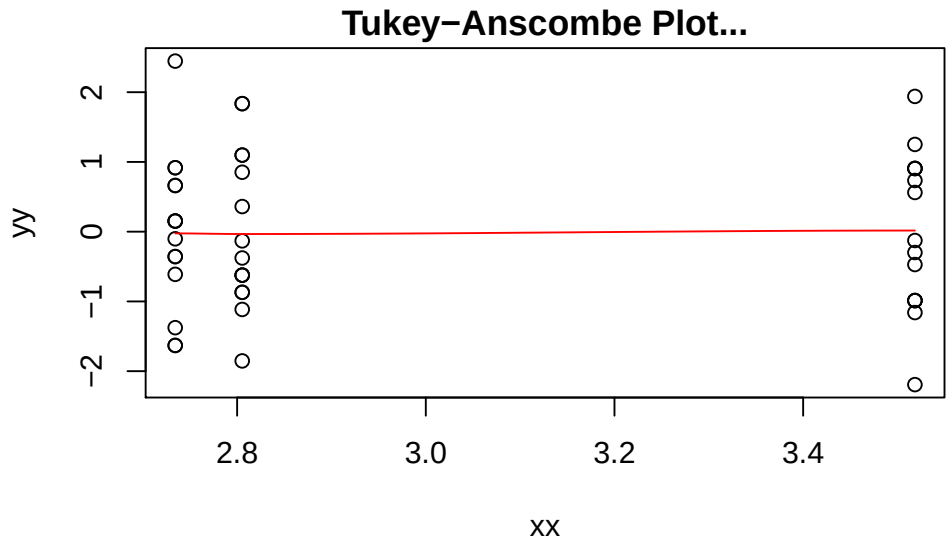
```
> # Read in the data
> count <- c(31,28,33,38,28,32,39,27,28,39,21,39,45,37,
            41,14,16,18,9,21,21,14,12,13,13,14,20,24,
            15,24,18,13,19,14,15,16,14,19,25,16,16,18,9,10,9)
> probe <- factor(rep(1:3, each = 15))
> vol <- c(rep(40,15), rep(20,30))
> nema <- data.frame(probe,count,vol)
> # Fit Poisson Regression Model
> mod1 <- glm(count~probe, family=poisson, data=nema)
> summary(mod1)

Call:
glm(formula = count ~ probe, family = poisson, data = nema)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3580  -0.9031  -0.1267   0.8846   2.2417

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.51849    0.04446  79.146   <2e-16 ***
probe2      -0.71311    0.07751  -9.200   <2e-16 ***
probe3      -0.78412    0.07941  -9.875   <2e-16 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 188.602  on 44  degrees of freedom
Residual deviance:  52.528  on 42  degrees of freedom
AIC: 276.14

Number of Fisher Scoring iterations: 4
```

The residual deviance is on the order of the degrees of freedom, suggesting that the model fits well. Indeed, we cannot reject the null hypothesis, that the model fits:

```
> pchisq(deviance(mod1), df.residual(mod1), lower=FALSE)

[1] 0.127964
```

Also, looking at the Tukey-Anscombe plot, it seems plausible that the Pearson residuals follow a standard Normal distribution:

```
> xx <- predict(mod1, type="link")
> yy <- resid(mod1, type="pearson")
> plot(xx, yy, main="Tukey-Anscombe Plot...")
> lines(loess.smooth(xx, yy), col="red")
```



b) There is a large difference between probe 1 and the other two. However, probe 1 has a different volume which could account for the observed difference.

c) We now model the log-rate as a linear function of the log-volume, i.e.

$$\lambda_i = \exp(\beta_0 + \beta_1 \log \mathrm{vol}_i) \tag{1}$$

```
> mod2 <- glm(count~log(vol), family=poisson, data=nema)
> summary(mod2)

Call:
glm(formula = count ~ log(vol), family = poisson, data = nema)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3580  -0.7674  -0.1267   0.7368   2.0861

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.46223    0.30991  -1.491    0.136
log(vol)     1.07911    0.09197  11.733   <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 188.602  on 44  degrees of freedom
Residual deviance:  53.131  on 43  degrees of freedom
AIC: 274.74

Number of Fisher Scoring iterations: 4
```

Again, judging from the residual deviance, the model fits well.

d) If we re-write the model equation (1) from the last part, we get:

$$\lambda_i = e^{\beta_0} \cdot \mathrm{vol}_i^{\beta_1}$$

Hence, for $\beta_1 = 1$, λ is proportional to vol. We check whether $\beta_1 = 1$ is reasonable by computing its confidence interval:

```
> confint(mod2)
                 2.5 %    97.5 %
(Intercept) -1.0721154 0.1430996
log(vol)     0.8988966 1.2595331
```

The confidence interval for β_1 does include 1, so $\lambda = c \cdot \text{vol}$ seems to be a reasonable approximation (where $c = e^{\beta_0}$).

e) We now fit the model $\lambda = c \cdot \text{vol}$ by constraining β_1 to 1:

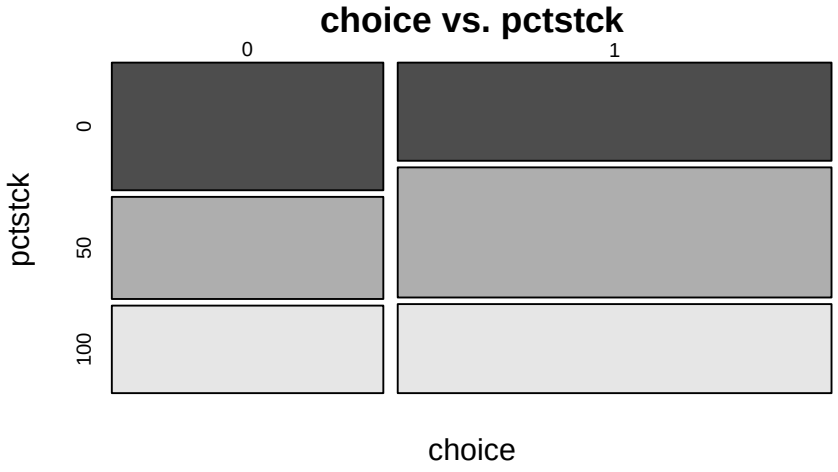
```
> mod3 <- glm(count~offset(log(vol)), family=poisson, data=nema)
> summary(mod3)

Call:
glm(formula = count ~ offset(log(vol)), family = poisson, data = nema)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2127  -0.8656  -0.1033   0.8548   2.0091

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.19744    0.03186  -6.196 5.78e-10 ***
---
Signif. codes:
0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 53.871  on 44  degrees of freedom
Residual deviance: 53.871  on 44  degrees of freedom
AIC: 273.48

Number of Fisher Scoring iterations: 4
```

We get a very similar residual deviance as before, so also this model fits well.

4. a)

```
> library(foreign)
> pension <- read.dta("http://fmwww.bc.edu/ec-p/data/wooldridge2k/pension.dta")
> pension$pctstck <- factor(pension$pctstck)
> pension$choice <- factor(pension$choice)
> pension$female <- factor(pension$female)
> pension$married <- factor(pension$married)
> pension$black <- factor(pension$black)
> pension$prftshr <- factor(pension$prftshr)
```

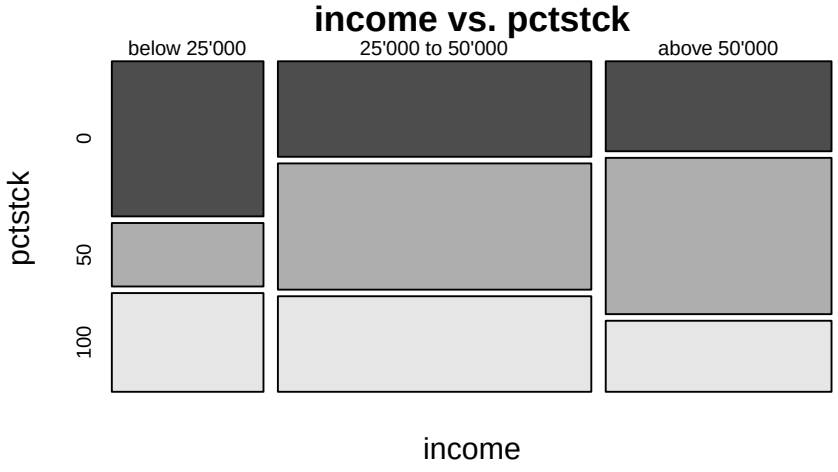
```
> table(pension$choice,pension$pctstck)

     0 50 100
  0 35 28  24
  1 43 57  39

> prop.table(table(pension$choice,pension$pctstck),1)

            0        50       100
  0 0.4022989 0.3218391 0.2758621
  1 0.3093525 0.4100719 0.2805755

> mosaicplot(table(pension$choice,pension$pctstck), color=TRUE,
            main="choice vs. pctstck",xlab="choice",ylab="pctstck")
```



People with freedom to choose their investment strategy avoid portfolios mainly consisting on obligations.

b)

```
> pension$inc <- rep(NA,226)
> pension$inc[pension$finc25==1] <- 1
> pension$inc[pension$finc35==1 | pension$finc50==1] <- 2
> pension$inc[pension$finc75==1 | pension$finc100==1 | pension$finc101==1] <- 3
> pension$inc <- factor(pension$inc,labels=
                        c("below 25'000","25'000 to 50'000", "above 50'000"))
> table(pension$inc,pension$pctstck)
                     0 50 100
  below 25'000      22  9  14
  25'000 to 50'000 28 37  28
  above 50'000      19 33  15
> prop.table(table(pension$inc,pension$pctstck),1)
                            0        50       100
  below 25'000      0.4888889 0.2000000 0.3111111
  25'000 to 50'000 0.3010753 0.3978495 0.3010753
  above 50'000      0.2835821 0.4925373 0.2238806
> mosaicplot(table(pension$inc,pension$pctstck), color=TRUE,
           main="income vs. pctstck",xlab="income",ylab="pctstck")
```



From the mosaic plots we can clearly see that people with a higher income are more likely to have mixed investment strategies.

c)
```
> library(nnet)
> pension$pct <- factor(pension$pctstck, levels = c("50","0","100"),
                        ordered = FALSE)
> str(pension)
'data.frame':        226 obs. of  21 variables:
 $ id      : int  38 152 152 182 222 226 233 233 253 314 ...
 $ pyears  : int  1 6 25 20 35 13 2 10 26 5 ...
 $ prftshr : Factor w/ 2 levels "0","1": 1 2 2 2 1 2 1 2 1 1 ...
 $ choice  : Factor w/ 2 levels "0","1": 2 2 2 1 2 1 2 1 1 2 ...
 $ female  : Factor w/ 2 levels "0","1": 1 2 1 2 1 1 2 2 2 1 ...
 $ married : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ age     : int  64 56 56 63 67 64 64 64 69 60 ...
 $ educ    : int  12 13 12 12 12 11 12 12 12 14 ...
 $ finc25  : int  0 0 0 1 0 0 1 1 0 0 ...
 $ finc35  : int  0 0 0 0 1 0 0 0 1 0 ...
 $ finc50  : int  1 0 0 0 0 0 0 0 0 0 ...
 $ finc75  : int  0 1 1 0 0 1 0 0 0 0 ...
 $ finc100 : int  0 0 0 0 0 0 0 0 0 0 ...
 $ finc101 : int  0 0 0 0 0 0 0 0 0 1 ...
 $ wealth89: num  77.9 154.9 154.9 232.5 179 ...
 $ black   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ stckin89: int  1 1 1 1 0 1 0 0 0 1 ...
 $ irain89 : int  1 1 1 1 1 0 1 1 0 1 ...
 $ pctstck : Factor w/ 3 levels "0","50","100": 1 2 2 3 3 1 3 3 2 2 ...
 $ inc     : Factor w/ 3 levels "below 25'000",..: 2 3 3 1 2 3 1 1 2 3 ...
 $ pct     : Factor w/ 3 levels "50","0","100": 2 1 1 3 3 2 3 3 1 1 ...
 - attr(*, "datalabel")= chr ""
 - attr(*, "time.stamp")= chr " 3 Dec 2001 14:45"
 - attr(*, "formats")= chr  "%9.0g" "%9.0g" "%9.0g" "%9.0g" ...
 - attr(*, "types")= int  105 98 98 98 98 98 98 98 98 98 ...
 - attr(*, "val.labels")= chr  "" "" "" "" ...
 - attr(*, "var.labels")= chr  "family identifier" "years in pension plan" "=1 if profit shari
 - attr(*, "version")= int 6
```

Here, we re-arrange the levels because R automatically takes the first one as the baseline.
The predictors `age` and `educ` are counts, therefore we transform them:

```
> pension$age <- sqrt(pension$age)
> pension$educ <- sqrt(pension$educ)
```

Now we fit a multinomial logit model:

```
> mod1 <- multinom(pct~choice+age+educ+female+married+black+inc
                   +wealth89+prftshr, data=pension)
# weights:  36 (22 variable)
initial  value 202.144661
iter  10 value 184.319612
iter  20 value 182.233592
iter  30 value 182.021332
final  value 182.021109
converged
> summary(mod1)
Call:
multinom(formula = pct ~ choice + age + educ + female + married +
    black + inc + wealth89 + prftshr, data = pension)

Coefficients:
    (Intercept)    choice1        age       educ    female1
0     -6.506605 -0.4755586  1.5645957 -1.1083136 -0.2291351
100    5.305235  0.1169744 -0.3346569 -0.6171987 -0.2234747
      married1      black1 inc25'000 to 50'000
```

```
0   -0.7207851 -0.45888909           -1.274006
100 -0.5213525  0.02713282           -0.557277
    incabove 50'000     wealth89  prftshr1
0        -1.0767704 0.0008916188 0.2189576
100      -0.9811298 0.0005537250 1.2438281

Std. Errors:
    (Intercept)   choice1       age      educ   female1
0    0.02298326 0.4197700 0.2780089 0.5681811 0.4238711
100  0.02599240 0.4309903 0.2866715 0.5774954 0.4299014
     married1    black1 inc25'000 to 50'000 incabove 50'000
0   0.5605508 0.6947724           0.5548439       0.6468224
100 0.5507528 0.6439294           0.5767501       0.6878132
        wealth89  prftshr1
0   0.0008073750 0.5448665
100 0.0008984634 0.5120854

Residual Deviance: 364.0422
AIC: 408.0422
```

d) `> mod2 <- multinom(pct~age+educ+female+married+black+inc+wealth89+prftshr, data=pension)`

```
# weights:  33 (20 variable)
initial  value 202.144661
iter  10 value 185.178630
iter  20 value 183.167364
iter  30 value 183.084823
final  value 183.084802
converged
```

`> pchisq(deviance(mod2) - deviance(mod1), mod1$edf - mod2$edf, lower=FALSE)`

```
[1] 0.3451787
```

From the deviance differences based Chi-squared test we can see that the predictor `choice` is not significant.

Looking at the summary output of `mod1`, we can see that, on the one hand, the odds of an investment strategy consisting mainly in obligations versus a mixed one decrease by 42% ($e^{-0.53} = 0.58$) when the people have the freedom to choose their investment strategy. On the other hand, the odds of investment strategies consisting mainly in stocks versus mixed one increase about 14% ($e^{0.13} = 1.14$) when the people can choose their strategy.

e)

```
> predict(mod1,type="probs",newdata=data.frame(choice="0",age=
                            sqrt(60), educ=sqrt(13.5),
                            female="0",married="0",
                            black="0",inc="above 50'000",
                            wealth89=200,prftshr="1"))
       50         0       100
0.1775220 0.4199362 0.4025418
> predict(mod1,type="probs",newdata=data.frame(choice="1",age=
                            sqrt(60), educ= sqrt(13.5),
                            female="0",married="0",
                            black="0",inc="above 50'000",
                            wealth89=200,prftshr="1"))
       50         0       100
0.1992342 0.2929292 0.5078366
```

Or we can also obtain from R the level of the response for which the probability is maximized:

```
> predict(mod1,type="class",newdata=data.frame(choice="0",age=
                            sqrt(60), educ=sqrt(13.5),
                            female="0",married="0",
                            black="0",inc="above 50'000",
                            wealth89=200,prftshr="1"))
```

```
[1] 0
Levels: 50 0 100
```

```
> predict(mod1,type="class",newdata=data.frame(choice="1",age=
                              sqrt(60), educ= sqrt(13.5),
                              female="0",married="0",
                              black="0",inc="above 50'000",
                              wealth89=200,prftshr="1"))
```

```
[1] 100
Levels: 50 0 100
```

For this person the probability of having a mixed investment strategy increases and the probability of having a strategy mainly consisting of obligations decreases when we specify that he had the freedom to choose his strategy.