qbs121_hw4_gibran

Gibran Erlangga

2/6/2022

1.1 Modelling Student Absences

Analyze the dataset quine which comes with the R library MASS. The dependent variable is number of student absences.

```
library(MASS)
data <- quine
```

1. Put together a table of univariable (1 covariate at at time) results on how each of the covariates relate to student absences.

```
attach(data)

dependent_var <- "Days"
vars <- names(data)[!names(data) %in% dependent_var]
ratios_df <- matrix(nrow=0, ncol=4)

df <- c()

for (var in vars) {
  model <- summary(glm(Days~data[, var]))
  coef <- model$coef[2, ]
  ratios_df <- rbind(ratios_df, c(exp(model$coef[2,1:2] %*% matrix(nrow=2, ncol=3, c(1,0,1,-2,1,+2))),
                                  model$coef[2,4]))

  df <- rbind(df, coef)
}

# set column name and row index
dimnames(ratios_df)[[2]] <- c("Odds Ratio", "95% CI", "Up", "P-value")
dimnames(ratios_df)[[1]] <- vars
print(ratios_df)
```

```
##       Odds Ratio       95% CI           Up      P-value
## Eth 1.173833e-04 6.523328e-07 2.112241e-02 0.0006508376
## Sex 1.532592e+01 6.887280e-02 3.410400e+03 0.3141892916
## Age 2.473149e-02 1.124885e-05 5.437413e+01 0.3379307571
## Lrn 4.403106e+00 1.901312e-02 1.019682e+03 0.5869582005
```

```
# store column name to col_names var
col_names <- colnames(df)

# set iindex, but it removes the column names
dimnames(df) <- list(vars)

# put column names back
colnames(df) <- col_names

print(df)
```

```
##      Estimate Std. Error    t value     Pr(>|t|)
## Eth -9.050066   2.596323 -3.4857248 0.0006508376
## Sex  2.729545   2.702520  1.0100002 0.3141892916
## Age -3.699678   3.847783 -0.9615089 0.3379307571
## Lrn  1.482310   2.722468  0.5444730 0.5869582005
```

2. Put together a table of multivariable results, i.e., run a multivariable model using all of the variables (or a subset if you choose).

```
model_multivariate <- glm(Days~Eth + Sex + Age + Lrn)
summary(model_multivariate)
```

```
##
## Call:
## glm(formula = Days ~ Eth + Sex + Age + Lrn)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -23.038  -10.027   -3.297    7.094   54.799
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   16.233      3.767   4.309 3.08e-05 ***
## EthN          -8.745      2.529  -3.458 0.000721 ***
## SexM           2.530      2.635   0.960 0.338631
## AgeF1         -4.457      3.929  -1.134 0.258547
## AgeF2          4.701      3.906   1.204 0.230778
## AgeF3          6.805      4.107   1.657 0.099771 .
## LrnSL          5.267      3.055   1.724 0.086934 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 231.9183)
##
##     Null deviance: 38304  on 145  degrees of freedom
## Residual deviance: 32237  on 139  degrees of freedom
## AIC: 1218.3
##
## Number of Fisher Scoring iterations: 2
```

3. Do this in two ways, (i) using Poisson regression in conjunction with sandwich variance to determine standard errors (or by selecting family=quasipoisson in the glm function) and (ii) negative binomial regression. Comment on the difference or similarity between the two sets of results.

```r
# poisson regression
model_poisson <- glm(Days~Eth + Sex + Age + Lrn, family=quasipoisson)
summary(model_poisson)
```

```
##
## Call:
## glm(formula = Days ~ Eth + Sex + Age + Lrn, family = quasipoisson)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -6.808  -3.065  -1.119   1.819   9.909
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.7154     0.2347  11.569  < 2e-16 ***
## EthN         -0.5336     0.1520  -3.511 0.000602 ***
## SexM          0.1616     0.1543   1.047 0.296914
## AgeF1        -0.3339     0.2543  -1.313 0.191413
## AgeF2         0.2578     0.2265   1.138 0.256938
## AgeF3         0.4277     0.2456   1.741 0.083831 .
## LrnSL         0.3489     0.1888   1.848 0.066760 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 13.16691)
##
##     Null deviance: 2073.5  on 145  degrees of freedom
## Residual deviance: 1696.7  on 139  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
# negative binomial regression
model_nb <- glm.nb(Days~Eth + Sex + Age + Lrn)
summary(model_nb)
```

```
##
## Call:
## glm.nb(formula = Days ~ Eth + Sex + Age + Lrn, init.theta = 1.274892646,
##     link = log)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7918  -0.8892  -0.2778   0.3797   2.1949
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.89458    0.22842  12.672  < 2e-16 ***
## EthN        -0.56937    0.15333  -3.713 0.000205 ***
## SexM         0.08232    0.15992   0.515 0.606710
## AgeF1       -0.44843    0.23975  -1.870 0.061425 .
## AgeF2        0.08808    0.23619   0.373 0.709211
## AgeF3        0.35690    0.24832   1.437 0.150651
```

```
## LrnSL        0.29211    0.18647   1.566 0.117236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1.2749) family taken to be 1)
##
##     Null deviance: 195.29  on 145  degrees of freedom
## Residual deviance: 167.95  on 139  degrees of freedom
## AIC: 1109.2
##
## Number of Fisher Scoring iterations: 1
##
##
##               Theta:  1.275
##           Std. Err.:  0.161
##
##  2 x log-likelihood:  -1093.151
```

```
detach(data)
```

Poisson regression results indicates that EthN is the only variable with a significant association with Days, with AgeF3 and LrnSL are the other variables which ranked 2nd and 3th in the p-value score (although not significant). Similarly, the result from negative binomial regression also gives the same notion, which says that EthN is the only variable with a significant association with Days, with AgeF1 as the variable in the 2nd lowest p-value score (although not significant).

1.2 Cancer Counts in Danish Cities

Access the data eba1977 in the R library ISwR. This is a small dataset on cancer counts by city and age group in Denmark.

```
library(ISwR)
data <- eba1977
head(data,3)
```

```
##         city   age  pop cases
## 1 Fredericia 40-54 3059    11
## 2    Horsens 40-54 2879    13
## 3    Kolding 40-54 3142     4
```

1. Which variable makes sense to use as an offset? Offset is the variable that is used to denote the exposure period in the Poisson regression. In this particular case, I think pop variable makes the most sense to be used as offset.

2. a. Use Poisson regression to model the association with age group. b. Test the significance of age using anova(o.glm, test="Chisq") c. Test the association of age as an ordinal variable (hint: create ageOrdinal = as.numeric(age)).

```
# a. Use Poisson regression to model the association with age group.
model_age_poisson <- glm(cases~age, offset=log(pop), family=poisson, data=data)
summary(model_age_poisson)
```

```
##
## Call:
## glm(formula = cases ~ age, family = poisson, data = data, offset = log(pop))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8520  -0.6424  -0.1067   0.7853   1.5468
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.8623     0.1741 -33.676  < 2e-16 ***
## age55-59      1.0823     0.2481   4.363 1.29e-05 ***
## age60-64      1.5017     0.2314   6.489 8.66e-11 ***
## age65-69      1.7503     0.2292   7.637 2.22e-14 ***
## age70-74      1.8472     0.2352   7.855 4.00e-15 ***
## age75+        1.4083     0.2501   5.630 1.80e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 129.908  on 23  degrees of freedom
## Residual deviance:  28.307  on 18  degrees of freedom
## AIC: 136.69
##
## Number of Fisher Scoring iterations: 5
```

```
# b. Test the significance of age using anova(o.glm, test="Chisq")
anova(model_age_poisson, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: cases
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                    23    129.908
## age   5    101.6        18     28.307 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# c. Test the association of age as an ordinal variable
ageOrdinal = as.numeric(data$age)
model_age_ordinal_poisson <- glm(data$cases~ageOrdinal, offset=log(data$pop), family=poisson)
summary(model_age_ordinal_poisson)
```

```
##
## Call:
## glm(formula = data$cases ~ ageOrdinal, family = poisson, offset = log(data$pop))
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3267  -0.3953   0.2912   1.0869   2.3017
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.63185    0.14058 -40.062  < 2e-16 ***
## ageOrdinal   0.28459    0.03498   8.135 4.11e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 129.908  on 23  degrees of freedom
## Residual deviance:  65.323  on 22  degrees of freedom
## AIC: 165.71
##
## Number of Fisher Scoring iterations: 4
```

3. a. Use Poisson regression to model the association with city. b. Test the significance of city.

```
# a. Use Poisson regression to model the association with city.
model_city_poisson <- glm(cases~city, offset=log(pop), family=poisson, data=data)
summary(model_city_poisson)
```

```
##
## Call:
## glm(formula = cases ~ city, family = poisson, data = data, offset = log(pop))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.8908  -0.3705   1.0893   2.2012   3.1090
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.5837     0.1250 -36.670   <2e-16 ***
## cityHorsens  -0.2286     0.1813  -1.261   0.2073
## cityKolding  -0.3357     0.1877  -1.789   0.0737 .
## cityVejle    -0.1883     0.1877  -1.003   0.3157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 129.91  on 23  degrees of freedom
## Residual deviance: 126.52  on 20  degrees of freedom
## AIC: 230.9
##
## Number of Fisher Scoring iterations: 5
```

```r
# b. Test the significance of age using anova(o.glm, test="Chisq")
anova(model_city_poisson, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: cases
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                    23     129.91
## city  3   3.3927        20     126.52    0.335
```

4. Run the multivariable model with city and age.

```r
model_city_age <- glm(cases~city+age, offset=log(pop), family=poisson, data=data)
summary(model_city_age)
```

```
##
## Call:
## glm(formula = cases ~ city + age, family = poisson, data = data,
##     offset = log(pop))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.63573  -0.67296  -0.03436   0.37258   1.85267
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -5.6321     0.2003 -28.125  < 2e-16 ***
## cityHorsens  -0.3301     0.1815  -1.818   0.0690 .
## cityKolding  -0.3715     0.1878  -1.978   0.0479 *
## cityVejle    -0.2723     0.1879  -1.450   0.1472
## age55-59      1.1010     0.2483   4.434 9.23e-06 ***
## age60-64      1.5186     0.2316   6.556 5.53e-11 ***
## age65-69      1.7677     0.2294   7.704 1.31e-14 ***
## age70-74      1.8569     0.2353   7.891 3.00e-15 ***
## age75+        1.4197     0.2503   5.672 1.41e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 129.908  on 23  degrees of freedom
## Residual deviance:  23.447  on 15  degrees of freedom
## AIC: 137.84
##
## Number of Fisher Scoring iterations: 5
```

5. For interest, instead of using an offset include log(population) as a covariate. Is the coefficient significantly different from 1.0 ?

```
model_offset <- glm(cases~city+age+log(pop), family=poisson, data=data)
summary(model_offset)
```

```
##
## Call:
## glm(formula = cases ~ city + age + log(pop), family = poisson,
##     data = data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.44001  -0.64195  -0.04286   0.50052   1.51893
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  11.7496     8.8151   1.333    0.183
## cityHorsens   0.1833     0.3193   0.574    0.566
## cityKolding  -0.0483     0.2520  -0.192    0.848
## cityVejle    -0.1679     0.1965  -0.855    0.393
## age55-59     -1.3842     1.2729  -1.087    0.277
## age60-64     -1.2367     1.4049  -0.880    0.379
## age65-69     -1.4378     1.6310  -0.882    0.378
## age70-74     -1.8049     1.8608  -0.970    0.332
## age75+       -1.8383     1.6588  -1.108    0.268
## log(pop)     -1.2096     1.1227  -1.077    0.281
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 27.704  on 23  degrees of freedom
## Residual deviance: 19.498  on 14  degrees of freedom
## AIC: 135.89
##
## Number of Fisher Scoring iterations: 4
```

2.1 Large Counts: Linear Regression vs Poisson

If the dependent variable is a count that takes large values (e.g. counts that are zero with very low frequency) it may be preferable to use linear regression. 1. Choose a sample size, e.g. n=500

```
n=500
```

2. Generate a couple continuous variables, Z1=rnorm(n) and Z2=rnorm(n)

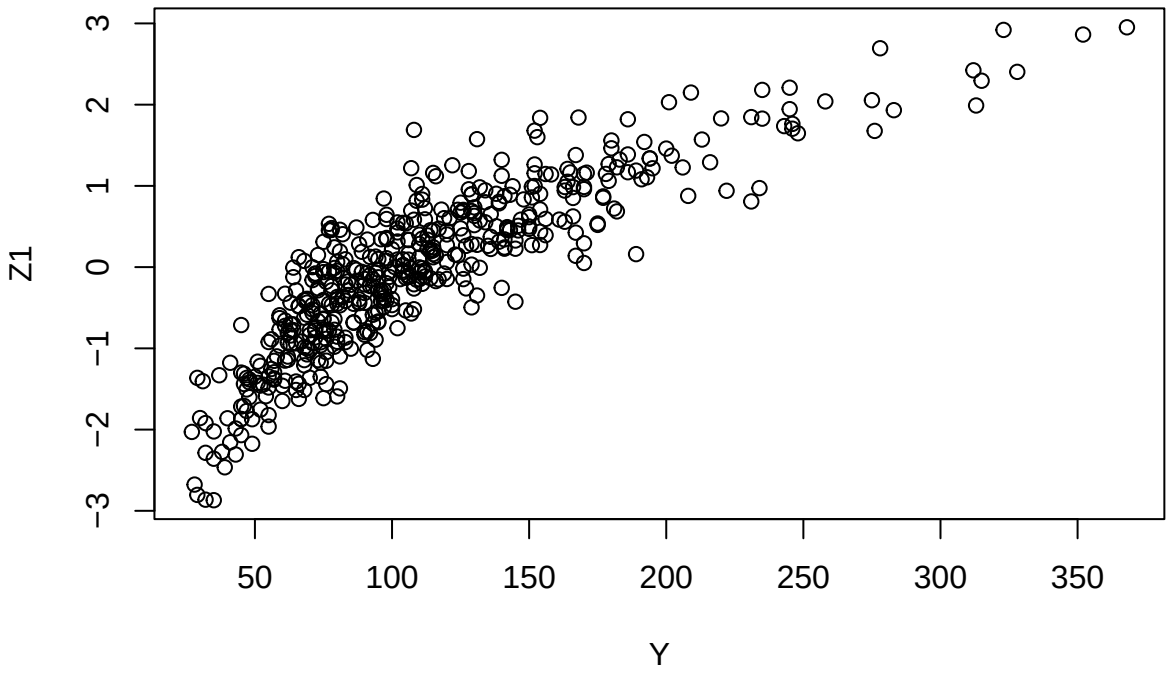
```
Z1=rnorm(n)
Z2=rnorm(n)
```

3. Generate a large count Y=rpois(n, lambda=100*1.5**Z1/1.2**Z2)

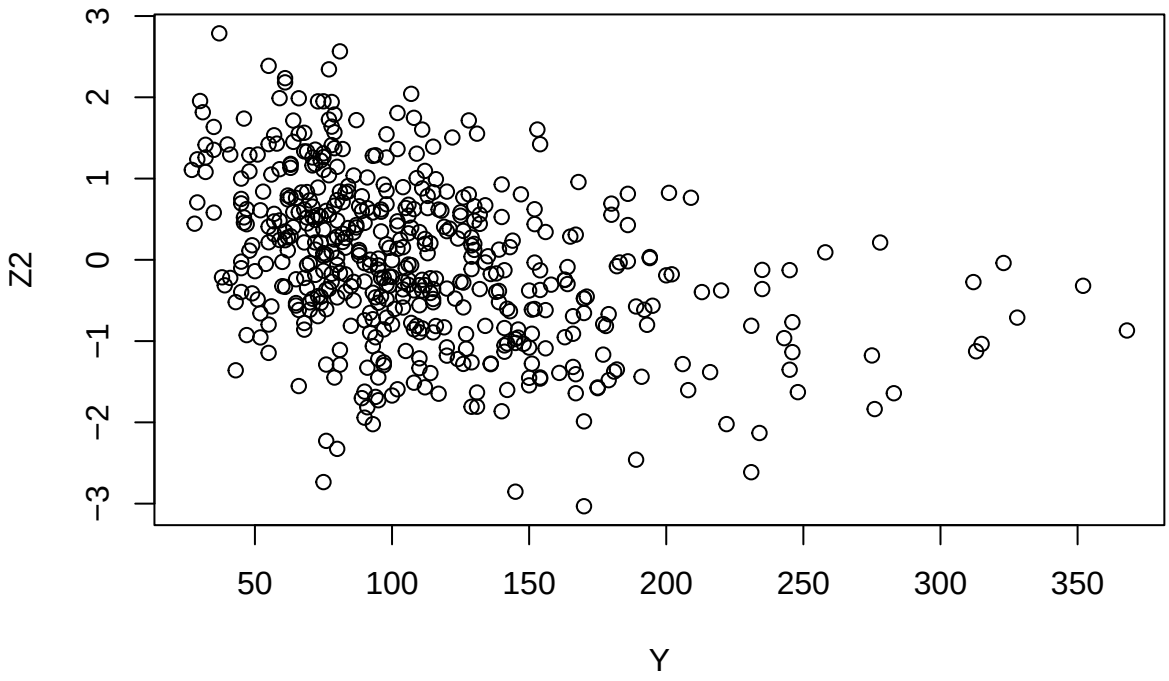
```
Y=rpois(n, lambda=100*1.5**Z1/1.2**Z2)
```

4. Plot this count vs Z1, and then versus Z2

```
plot(Y, Z1)
```




```
plot(Y, Z2)
```



5. Use multivariable Poisson regression to model Y vs Z1 and Z2.

```
model_poisson <- glm(Y~Z1+Z2, family=poisson)
summary(model_poisson)
```

```
##
## Call:
## glm(formula = Y ~ Z1 + Z2, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2976  -0.7023   0.0039   0.7001   3.3291
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.608798   0.004653  990.53   <2e-16 ***
## Z1           0.405791   0.004141   97.98   <2e-16 ***
## Z2          -0.177784   0.004353  -40.84   <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: 12180.44  on 499  degrees of freedom
## Residual deviance:   479.43  on 497  degrees of freedom
## AIC: 3703.3
```

```
##
## Number of Fisher Scoring iterations: 4
```

6. Use multivariable linear regression to model Y vz Z1 and Z2

```
model_linear <- glm(Y~Z1+Z2, family=gaussian)
summary(model_poisson)
```

```
##
## Call:
## glm(formula = Y ~ Z1 + Z2, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.2976  -0.7023   0.0039   0.7001   3.3291
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.608798   0.004653  990.53   <2e-16 ***
## Z1           0.405791   0.004141   97.98   <2e-16 ***
## Z2          -0.177784   0.004353  -40.84   <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: 12180.44  on 499  degrees of freedom
## Residual deviance:   479.43  on 497  degrees of freedom
## AIC: 3703.3
##
## Number of Fisher Scoring iterations: 4
```

7. Use multivariable linear regression to model log(Y) vz Z1 and Z2

```
model_poisson <- glm(log(Y)~Z1+Z2, family=gaussian)
summary(model_poisson)
```

```
##
## Call:
## glm(formula = log(Y) ~ Z1 + Z2, family = gaussian)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.46905  -0.05979   0.00606   0.07022   0.39783
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.604482   0.004630  994.42   <2e-16 ***
## Z1           0.404450   0.004548   88.93   <2e-16 ***
## Z2          -0.175221   0.004657  -37.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for gaussian family taken to be 0.01069727)
##
##     Null deviance: 110.2783  on 499  degrees of freedom
## Residual deviance:   5.3165  on 497  degrees of freedom
## AIC: -844.95
##
## Number of Fisher Scoring iterations: 2
```

8. Assess similarities and differences of the estimates, standard errors and Z-values from these three models.

Similarities:
Most obvious one is the fact that both variables have highly significant p-values. Another similarity I can spot is the estimated coefficients of Z1 and Z2, as well as the t-value and z-value in log linear regression and poisson regression, respectively. Additionally, the value of standard error in both log linear regression and poisson regression are also similar, with the standard error value of log linear regression being slightly higher than the poisson regression. Differences:
The coefficient estimates of Z1 and Z2 and its standard error from linear regression is significantly different from the other regression methods.

3.1 AUROC as Measure of Difference of Two Distributions

The AUROC of a score that predicts an event equals the probability that a subject with the event will have a higher score than a person without the event. If the distribution of the scores in subjects with the event is normal with mean m1 and s1 and the distribution of scores in subjects without the event is normal with mean m0 and s0, then the following R line of code estimates the concordancy.

a. Create a table of Concordancy vs the following choices, m0=0,sd=1, m1 = 0.0, 0.25,0.5,0.75,1,1.5,2,3 and s1=1.

```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 4.1.1
```

```
m0 <- 0
s0 <- 1
m1 <- c(0, 0.25, 0.5, 0.75, 1.0, 1.5, 2, 3)
s1 <- 1
n <- 1000

m0.values <- rep(0, 8)
s0.values <- rep(1, 8)
s1.values <- rep(1, 8)
auc <- rep(0, 8)

for (i in 1:8) {
  auc[i] = mean(rnorm(n=n<-10^6, mean=m0, sd=s0) < rnorm(n=n, mean=m1[i], sd=s1))
  }

df <- data.frame(cbind(m0.values, s0.values, m1, s1.values, auc))
df
```

```
##   m0.values s0.values   m1 s1.values      auc
## 1         0         1 0.00         1 0.500561
## 2         0         1 0.25         1 0.570550
## 3         0         1 0.50         1 0.638623
## 4         0         1 0.75         1 0.703016
## 5         0         1 1.00         1 0.760326
## 6         0         1 1.50         1 0.855969
## 7         0         1 2.00         1 0.921360
## 8         0         1 3.00         1 0.982995
```

b. Suppose a score for the risk of an event is such that its distribution in those who will have the event is normal with mean m1 and standard deviation of s1, and its distribution in those who will not have the event is mean 0 and standard deviation 1. Simulate the score of 1000 events (cases) and 1000 controls and plot the corresponding ROC curve for the following 4 scenarios (m1=0.5, s1=1), (m1=0.5, s1=2), (m1=2.0, s1=1), (m1=2.0, s1=2). To do this the following code could be helpful:

```
# m1=0.5, s1=1
m1 <- 0.5
s1 <- 1.0
n <- 1000

score_1 <- rnorm(n=n, mean=m1, sd=s1)
score_base <- rnorm(n=n, mean=0, sd=1)
event <- rep(c(1, 0), n)
scores <- c(score_1, score_base)
result <- roc(event, scores)
```

```
## Setting levels: control = 0, case = 1
```

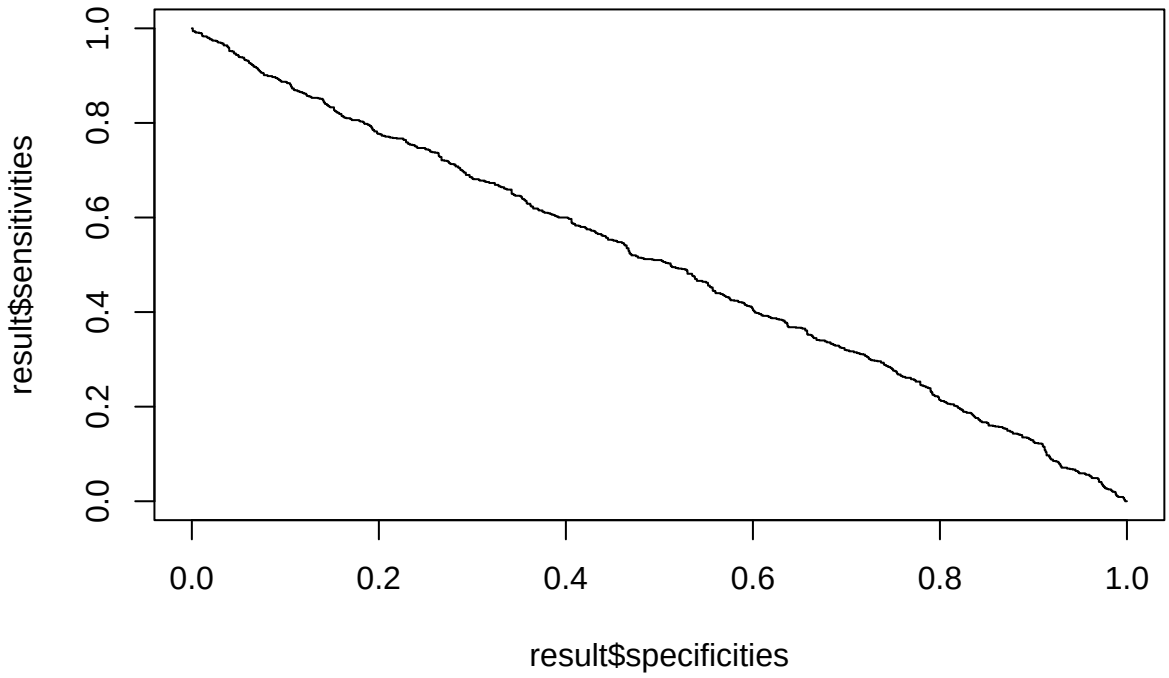
```
## Setting direction: controls > cases
```

```
print(paste(m1, s1))
```

```
## [1] "0.5 1"
```

```
plot(result$specificities, result$sensitivities, type="line")
```

```
## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to first
## character
```




```
# m1=2.0, s1=1
m1 <- 2.0
s1 <- 1.0
n <- 1000

score_1 <- rnorm(n=n, mean=m1, sd=s1)
score_base <- rnorm(n=n, mean=0, sd=1)
event <- rep(c(1, 0), n)
scores <- c(score_1, score_base)
result <- roc(event, scores)
```

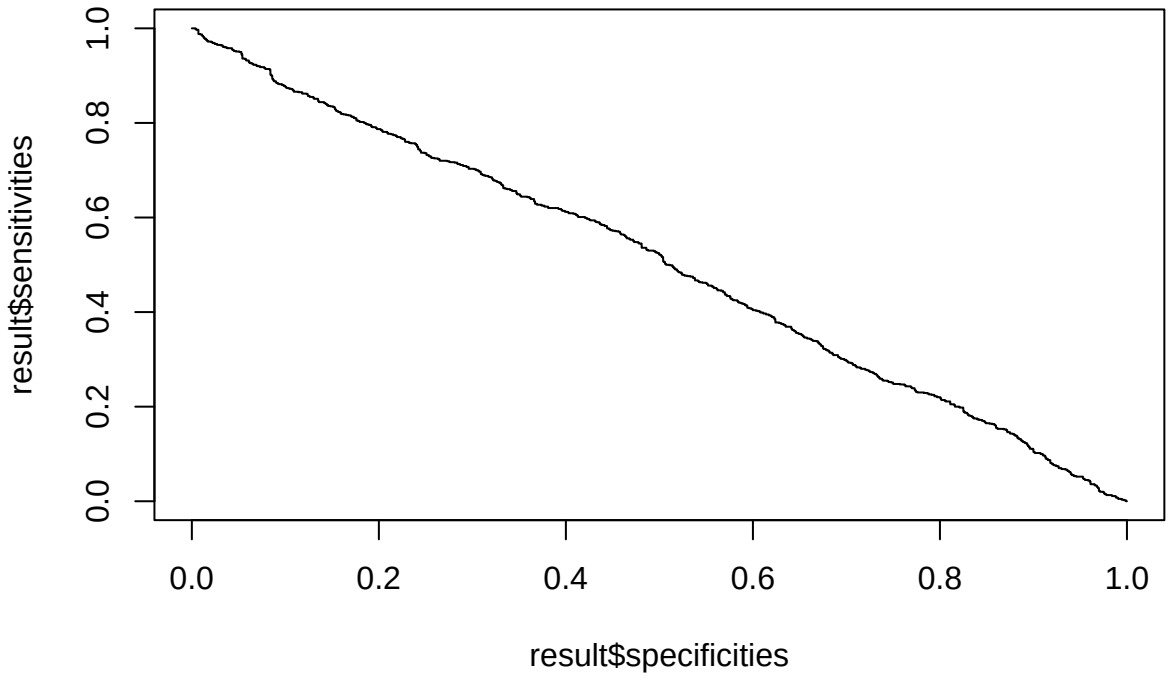
```
## Setting levels: control = 0, case = 1
## Setting direction: controls > cases
```

```
print(paste(m1, s1))
```

```
## [1] "2 1"
```

```
plot(result$specificities, result$sensitivities, type="line")
```

```
## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to first
## character
```




```
# m1=2.0, s1=2
m1 <- 2.0
s1 <- 2.0
n <- 1000

score_1 <- rnorm(n=n, mean=m1, sd=s1)
score_base <- rnorm(n=n, mean=0, sd=1)
event <- rep(c(1, 0), n)
scores <- c(score_1, score_base)
result <- roc(event, scores)
```

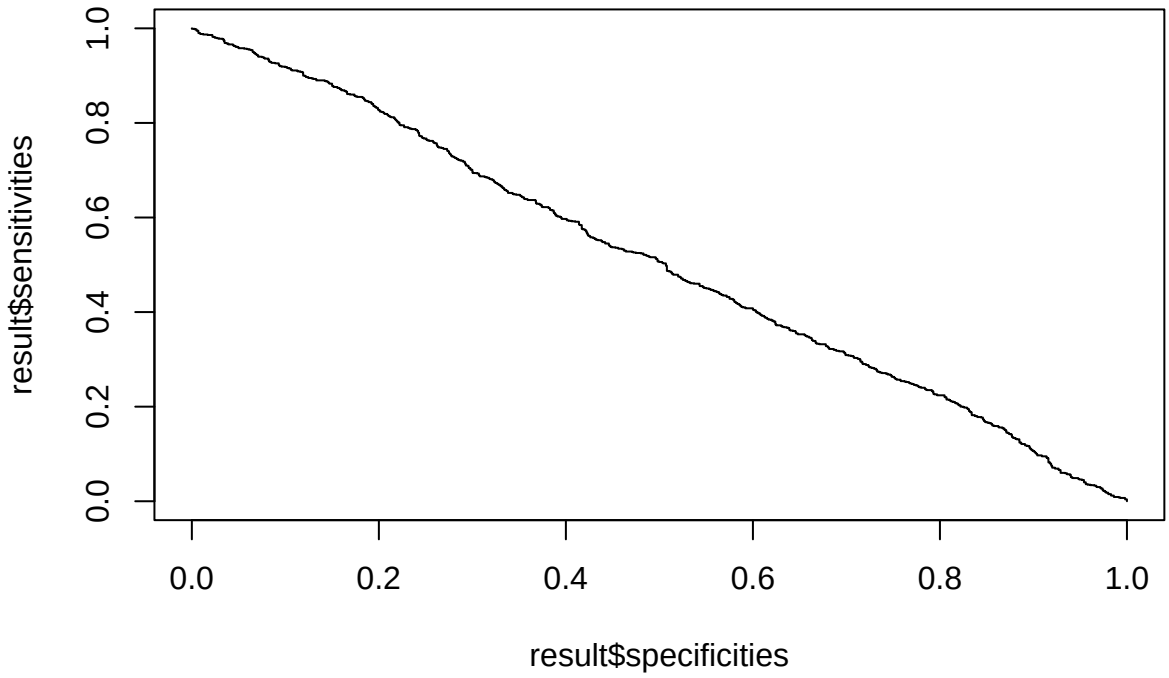
```
## Setting levels: control = 0, case = 1
## Setting direction: controls > cases
```

```
print(m1, s1)
```

```
## [1] 2
```

```
plot(result$specificities, result$sensitivities, type="line")
```

```
## Warning in plot.xy(xy, type, ...): plot type 'line' will be truncated to first
## character
```



3.2 ADR vs APC

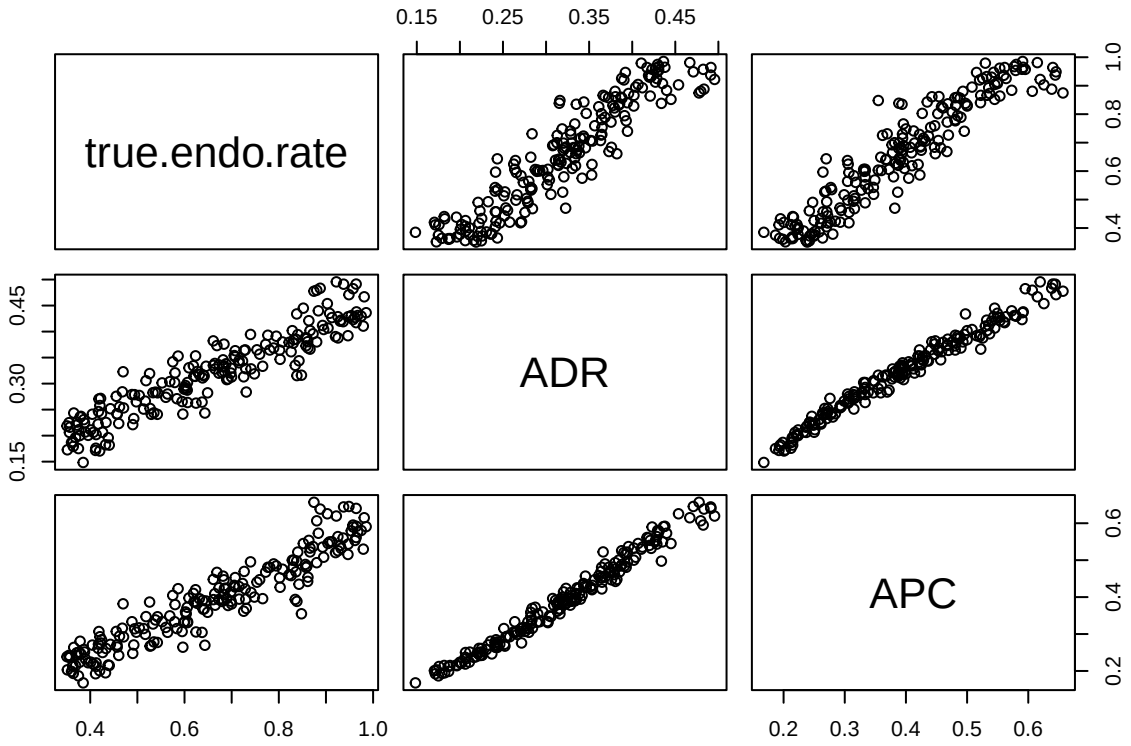
The most recommended modality for colorectal cancer screening in the USA is colonoscopy. During a colonoscopy a clinician uses a camera at the end of a tube (colonoscope) to examine the colon. The colonoscope is also equipped with features to remove pre-cancerous lesions (polyps, adenomas). Colonoscopists vary in their ability to detect polyps. One measure of detection ability is the Adenoma Detection Rate (ADR). It is defined as the proportion of colonoscopies in which at least one adenoma is detected; like the proportion of games in which an athlete gets at least one point. An alternative metric is the APC (adenomas per colonoscopies); like the average number of points per game. Explain what the following simulation is doing and interpret the results.

```
R <- 1000
cor.ADR.true <- cor.APC.true <- R
n.endoscopists <- 200 # number of endoscopists in the cohort

for (r in 1:R) {
  # number of patients each endoscopists scopes in a year
  n.pt.endoscopist <- ceiling(rgamma(n=n.endoscopists, shape=10, scale=30))
  N <- sum(n.pt.endoscopist)
  ID.Endo <- rep(1:n.endoscopists, times=n.pt.endoscopist)
  true.endo.rate <- runif(n.endoscopists , min=0.35,max=0.99) # given a uniform distribution
  long.true.endo.rate <- rep(true.endo.rate, times=n.pt.endoscopist)
  n.polyps <- rpois(n=N, lambda=0.6) # lambda is the average actual adenomas
  n.polyps.detected <- rbinom(n=N, size=n.polyps, prob=long.true.endo.rate)
  at.least.one <- n.polyps.detected>0
```

```
  ADR <- tapply(at.least.one, ID.Endo, mean)
  APC <- tapply(n.polyps.detected , ID.Endo, mean)
  cor.ADR.true[r] <- cor(ADR, true.endo.rate)
  cor.APC.true[r] <- cor(APC, true.endo.rate)}

pairs(cbind(true.endo.rate, ADR, APC))
```




```
summary(cbind(cor.ADR.true, cor.APC.true))
```

```
##   cor.ADR.true     cor.APC.true
##  Min.   :0.9060   Min.   :0.9204
##  1st Qu.:0.9286   1st Qu.:0.9403
##  Median :0.9348   Median :0.9450
##  Mean   :0.9343   Mean   :0.9446
##  3rd Qu.:0.9401   3rd Qu.:0.9495
##  Max.   :0.9592   Max.   :0.9653
```

The simulation is about Adenoma Detection Rate (ADR) and Adenomas per Colonoscopies (APR). The graph above is showing the correlation between ADR and true endoscopist rate, and also between APR and true endoscopist rate. APR seems to be fractionally better than ADR in its association with the value of true endo rate, but the discrepancy is not huge. Overall, I do not see any significant difference between both relational graphs (ADR vs true endo rate and APR vs true endo rate). This fact also supported by the strong linear association showed between ADR and APR in the graph.