Stat364 ~ HW2

Mert Göksel

Q1

```
df <- read.xlsx("./accidents.xlsx")
df
```

```
  Gender Location Seat-Belt     1   2   3   4  5
1 Female    Urban        No  7287 175 720  91 10
2   <NA>     <NA>       Yes 11587 126 577  48  8
3   <NA>    Rural        No  3246  73 710 159 31
4   <NA>     <NA>       Yes  6134  94 564  82 17
5   Male    Urban        No 10381 136 566  96 14
6   <NA>     <NA>       Yes 10969  83 259  37  1
7   <NA>    Rural        No  6123 141 710 188 45
8   <NA>     <NA>       Yes  6693  74 353  74 12
```

This is pandas multiindex thus need reset_index().

```
library(reticulate)
use_python("C:\\Python\\envs\\myenv\\Scripts\\python.exe")
```

```
import pandas as pd
df = pd.read_excel("./accidents.xlsx", index_col=[0,1,2]).reset_index()
df.head()
```

```
   Gender Location Seat-Belt      1    2    3    4   5
0  Female    Urban        No   7287  175  720   91  10
1  Female    Urban       Yes  11587  126  577   48   8
2  Female    Rural        No   3246   73  710  159  31
```

```
3  Female    Rural       Yes   6134   94  564   82  17
4    Male    Urban        No  10381  136  566   96  14
```

```r
df <- py$df
head(df)
```

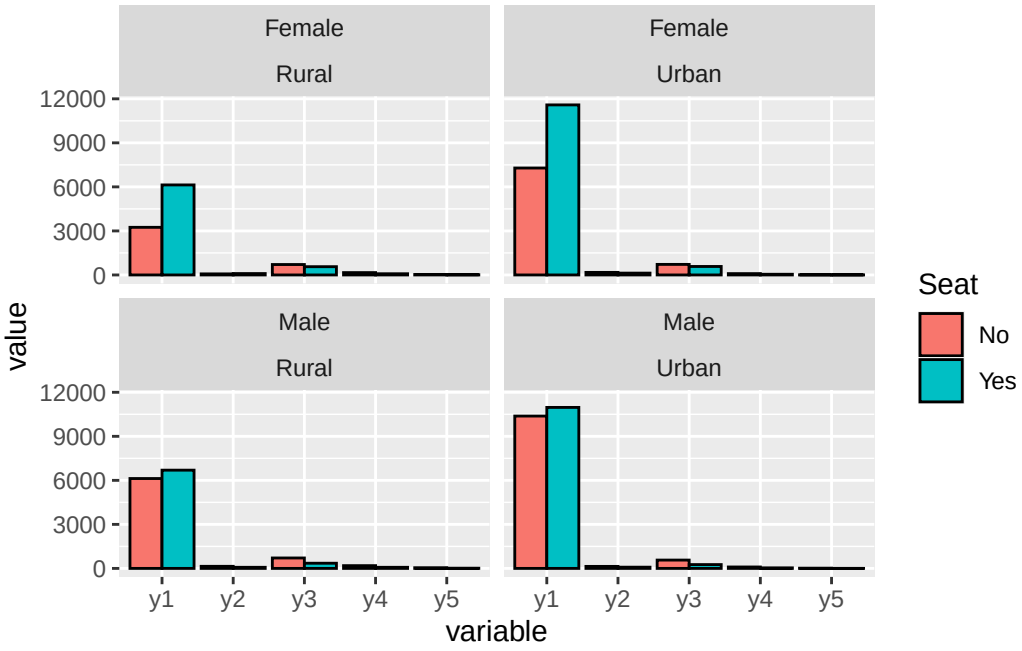
```
  Gender Location Seat-Belt     1   2   3   4  5
1 Female    Urban        No  7287 175 720  91 10
2 Female    Urban       Yes 11587 126 577  48  8
3 Female    Rural        No  3246  73 710 159 31
4 Female    Rural       Yes  6134  94 564  82 17
5   Male    Urban        No 10381 136 566  96 14
6   Male    Urban       Yes 10969  83 259  37  1
```

Our conversion to r has been completed. Now we can begin with the analysis.

First, I want to see the visualizations in order to come up with an analysis plan.

```r
df <- df %>% rename(y1 = "1",
                    y2 = "2",
                    y3 = "3",
                    y4 = "4",
                    y5 = "5", Seat = `Seat-Belt`)
df %>% reshape2::melt() %>%
  ggplot(aes(x=variable, y=value, fill=Seat)) +
  geom_bar(stat="identity",position=position_dodge(),color='black') +
  facet_wrap(vars(Gender, Location))
```

```
Using Gender, Location, Seat as id variables
```



We know that 1st variable is the "not injured" variable and we see from the barplot that males have an approximately equal amount of non injured with seatbelt on and off. Other than that area type doesnt seem to affect the results of the crash.

now lets build our model

```
model <- vglm(cbind(y1,y2,y3,y4,y5)~Seat+Location+Gender,
              data=df, family = cumulative(parallel=TRUE))
```

```
Warning in vglm.fitter(x = x, y = y, w = w, offset = offset, Xm2 = Xm2, :
convergence not obtained in 30 IRLS iterations
```

```
summary(model)
```

```
Call:
vglm(formula = cbind(y1, y2, y3, y4, y5) ~ Seat + Location +
    Gender, family = cumulative(parallel = TRUE), data = df)

Coefficients:
```

```
              Estimate Std. Error z value            Pr(>|z|)
(Intercept):1 1.201115   0.008443  142.25 <0.0000000000000002 ***
(Intercept):2 1.376190   0.008583  160.33 <0.0000000000000002 ***
(Intercept):3 3.242534   0.029294  110.69 <0.0000000000000002 ***
(Intercept):4 5.150100   0.078849   65.32 <0.0000000000000002 ***
SeatYes       0.825203   0.007938  103.96 <0.0000000000000002 ***
LocationUrban 0.775308   0.007733  100.26 <0.0000000000000002 ***
GenderMale    0.545428   0.007815   69.79 <0.0000000000000002 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2]),
logitlink(P[Y<=3]), logitlink(P[Y<=4])

Residual deviance: 159.6304 on 25 degrees of freedom

Log-likelihood: -180.2704 on 25 degrees of freedom

Number of Fisher scoring iterations: 30

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):2', '(Intercept):3', '(Intercept):4'


Exponentiated coefficients:
      SeatYes LocationUrban    GenderMale
     2.282345      2.171260      1.725346
```

Our model is;

$$logit(P(Y \leq Not\ injured)) = 1.201115 - 0.825203.seatbelt + 0.775308.locationisurban + 0.545428.Genderismale$$
$$logit(P(Y \leq Injured\ faintly - nottransported)) = 1.376190 - 0.825203.seatbelt + 0.775308.locationisurban + 0.545428.Genderismale$$
$$logit(P(Y \leq Injured\ faintly - transported)) = 3.242534 - 0.825203.seatbelt + 0.775308.locationisurban + 0.545428.Genderismale$$
$$logit(P(Y \leq Injured\ heavily)) = 5.150100 - 0.825203.seatbelt + 0.775308.locationisurban + 0.545428.Genderismale$$

For any fixed j, the estimated odds that a drivers injury where the crash happened in an urban area is in the better side rather than the worse direction (i.e., Y j rather than Y > j) equal exp(_2)=exp(0.775308)=2.171261 times the estimated odds for rural.

```
kable(data.frame(gender=df$Gender, location=df$Location,
                 seatbelt = df$Seat, prob=fitted(model)))
```

gender	location	seatbelt	prob.y1	prob.y2	prob.y3	prob.y4	prob.y5
Female	Urban	No	0.8782992	0.0175094	0.0865176	0.0150102	0.0026636
Female	Urban	Yes	0.9427636	0.0087468	0.0406683	0.0066526	0.0011688
Female	Rural	No	0.7687230	0.0296553	0.1640256	0.0318307	0.0057654
Female	Rural	Yes	0.8835327	0.0168418	0.0827974	0.0142938	0.0025343
Male	Urban	No	0.9256595	0.0111855	0.0528348	0.0087748	0.0015455
Male	Urban	Yes	0.9660082	0.0053027	0.0241410	0.0038704	0.0006778
Male	Rural	No	0.8515161	0.0208025	0.1055409	0.0187907	0.0033497
Male	Rural	Yes	0.9290209	0.0107126	0.0504435	0.0083525	0.0014704

Q2

```
cereal <- read.csv("./cereal_dillons.csv")
cereal %>% count(Cereal) %>% nrow() #38 different cereals
```

```
[1] 38
```

lets re-format our data:

```
stand01 <- function(x){(x-min(x))/(max(x)-min(x))}
cereal2 <- data.frame(Shelf=cereal$Shelf,
                      sugar=stand01(x=cereal$sugar_g/cereal$size_g),
                      fat=stand01(x=cereal$fat_g/cereal$size_g),
                      sodium=stand01(x=cereal$sodium_mg/cereal$size_g))
head(cereal2) #much better
```

```
  Shelf     sugar   fat    sodium
1     1 0.6428571 0.000 0.5666667
2     1 0.1285714 0.000 0.9000000
3     1 0.1285714 0.000 1.0000000
4     1 0.1125000 0.675 0.8166667
5     1 0.7800000 0.360 0.6533333
6     1 0.6387097 0.000 0.5419355
```

Part A

```r
#Estimate a multinomial regression model with linear forms of the sugar,
#fat, and sodium variables.

model <- vglm(Shelf~., data=cereal2, family = multinomial)
summary(model)
```

```
Call:
vglm(formula = Shelf ~ ., family = multinomial, data = cereal2)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept):1 -21.3002     7.4388      NA       NA
(Intercept):2 -14.3886     5.4947  -2.619  0.00883 **
(Intercept):3   0.3925     1.3487   0.291  0.77102
sugar:1        11.4012     4.8733   2.340  0.01931 *
sugar:2        14.0867     4.9890   2.824  0.00475 **
sugar:3        -0.8226     1.9541  -0.421  0.67379
fat:1           0.8703     2.4060   0.362  0.71755
fat:2           4.9349     2.7448   1.798  0.07219 .
fat:3           0.3128     1.7531   0.178  0.85839
sodium:1       24.6861     8.0661   3.060  0.00221 **
sodium:2        7.1831     5.5209   1.301  0.19324
sodium:3       -0.3051     2.1550  -0.142  0.88741
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: log(mu[,1]/mu[,4]), log(mu[,2]/mu[,4]),
log(mu[,3]/mu[,4])

Residual deviance: 67.1903 on 108 degrees of freedom

Log-likelihood: -33.5951 on 108 degrees of freedom

Number of Fisher scoring iterations: 7

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):1'
```

```
Reference group is level  4  of the response
```

```
#perform lrt to each variable

#for sugar
VGAM::lrtest(model, vglm(Shelf~fat+sodium, cereal2, family = multinomial))
```

```
Likelihood ratio test

Model 1: Shelf ~ .
Model 2: Shelf ~ fat + sodium
  #Df  LogLik Df  Chisq Pr(>Chisq)
1 108 -33.595
2 111 -44.978  3 22.765 0.00004521 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Sugar is significant

```
#for fat
VGAM::lrtest(model, vglm(Shelf~.-fat, cereal2, family = multinomial))
```

```
Likelihood ratio test

Model 1: Shelf ~ .
Model 2: Shelf ~ . - fat
  #Df  LogLik Df  Chisq Pr(>Chisq)
1 108 -33.595
2 111 -36.237  3 5.2836     0.1522
```

Fat seems to be not significant

```
#for sodium
VGAM::lrtest(model, vglm(Shelf~.-sodium, cereal2, family = multinomial))
```

```
Likelihood ratio test

Model 1: Shelf ~ .
Model 2: Shelf ~ . - sodium
  #Df  LogLik Df Chisq  Pr(>Chisq)
1 108 -33.595
2 111 -46.905  3 26.62 0.000007073 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

And sodium is significant.

Part B

```
stand02 <- \(x, y, z){
  z <- (cereal %>% select(z))/(cereal %>% select(size_g))
  (x/y - min(z))/(max(z)-min(z))
}

pred <- data.frame(sugar = stand02(12,28, "sugar_g"),
                   fat = stand02(0.5, 28, "fat_g"),
                   sodium = stand02(130, 28, "sodium_mg"))
```

```
Note: Using an external vector in selections is ambiguous.
i Use `all_of(z)` instead of `z` to silence this message.
i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
This message is displayed once per session.
```

```
prob <- predict(model, pred, type = "response", se.fit = F)
prob
```

```
          1         2         3         4
1 0.05322364 0.4720136 0.2004235 0.2743392
```

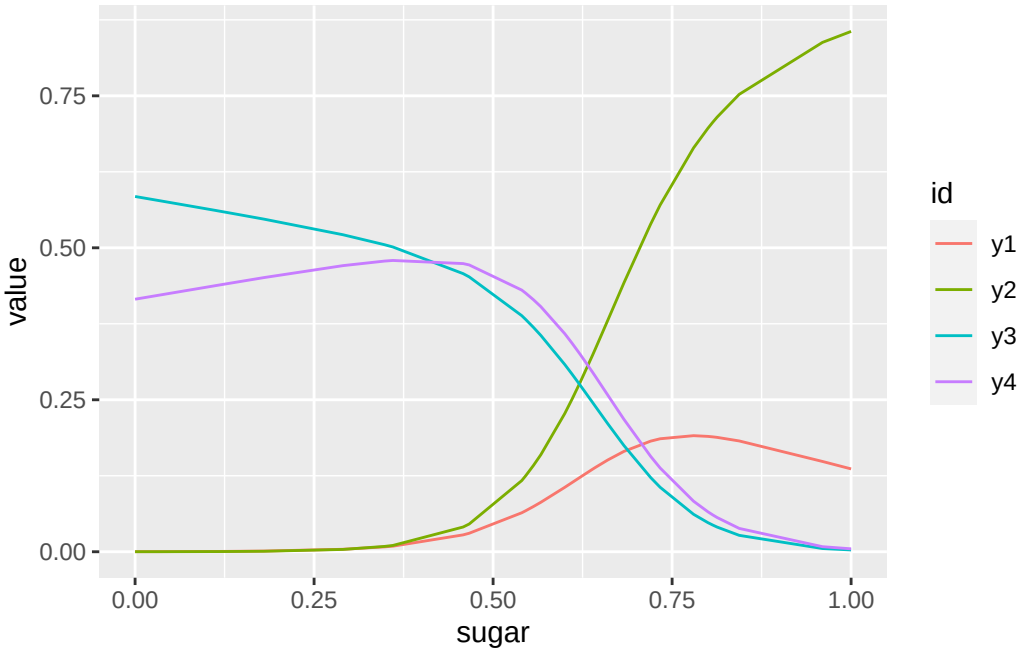
Most probable shelf is the second.

Part C

```
data_prob <- data.frame(Shelf = cereal$Shelf,
                        sugar = stand01(x = cereal$sugar_g/cereal$size_g),
                        fat = mean(stand01( x = cereal$fat_g/cereal$size_g)),
                        sodium = mean(stand01(x = cereal$sodium_mg/cereal$size_g )))

probs <- as.data.frame(predict(model, data_prob, type = "response", se.fit = F))
colnames(probs) <- c("y1","y2","y3","y4")

probs %>% mutate(sugar = data_prob$sugar) %>%
  arrange(sugar) %>% tidyr::gather("id", "value", 1:4) %>%
  ggplot(aes(x=sugar, y=value, color=id)) + geom_line()
```



Part D

```
int <- confint(model, level = .95)
round(exp(int), 4)
```

```
                2.5 %                97.5 %
```

```
(Intercept):1    0.0000                  0.0012
(Intercept):2    0.0000                  0.0268
(Intercept):3    0.1053                 20.8195
sugar:1          6.3574         1257897623.8851
sugar:2         74.3163        23144659186.1285
sugar:3          0.0095                 20.2333
fat:1            0.0214                266.6726
fat:2            0.6409              30171.3543
fat:3            0.0440                 42.4721
sodium:1      7164.0234 386297754496574400.0000
sodium:2         0.0263           65915945.0354
sodium:3         0.0108                 50.3294
```

Q3

```
df <- read.xlsx("heart.xlsx")
kable(head(df))
```

id	totalcost	age	gender	interventions	drugs	visits	complications	comorbidities	duration
1	179.1	63	0	2	1	4	0	3	300
2	319.0	59	0	2	0	6	0	0	120
3	9310.7	62	0	17	0	2	0	5	353
4	280.9	60	1	9	0	7	0	2	332
5	18727.1	55	0	5	2	7	0	0	18
6	453.4	66	0	1	0	3	0	4	296

Part A

```
model <- glm(visits ~., df, family = poisson(link = "log"))
summary(model)
```

```
Call:
glm(formula = visits ~ ., family = poisson(link = "log"), data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
```

```
-2.7010  -1.0393  -0.2316   0.5728   5.7162

Coefficients:
                  Estimate   Std. Error z value             Pr(>|z|)
(Intercept)    0.478391277  0.178463867   2.681              0.00735 **
id             0.000060080  0.000084904   0.708              0.47918
totalcost      0.000014900  0.000002864   5.202          0.000000197 ***
age            0.006691924  0.002965790   2.256              0.02405 *
gender         0.180867765  0.044018849   4.109          0.000039760 ***
interventions  0.010149667  0.003822211   2.655              0.00792 **
drugs          0.193380206  0.012674308  15.258 < 0.0000000000000002 ***
complications  0.061969071  0.060085040   1.031              0.30237
comorbidities -0.000920248  0.003686753  -0.250              0.80289
duration       0.000347794  0.000190035   1.830              0.06723 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1485.0  on 787  degrees of freedom
Residual deviance: 1043.1  on 778  degrees of freedom
AIC: 3272.5

Number of Fisher Scoring iterations: 5
```

The response function is:

$$ln(Y) = e^{0.4784 + id*0.0001 + totalcost*0 + age*0.0067 + gender*0.1809 + interventions*0.0101} * e^{drugs*0.1934 + complications*0.062 + comorbidities*-0.0009 + duration*0.0003}$$

Part B

```r
best.model <- step(model, direction = "backward", trace = F)$call
best.model <- glm(best.model, family = poisson(link = "log"), data = df)
summary(best.model)
```

```
Call:
glm(formula = best.model, family = poisson(link = "log"), data = df)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6057  -1.0366  -0.2380   0.5763   5.7457

Coefficients:
                 Estimate  Std. Error z value             Pr(>|z|)
(Intercept)   0.520840362 0.174455011   2.986              0.00283 **
totalcost     0.000014930 0.000002844   5.251          0.000000152 ***
age           0.006334003 0.002938392   2.156              0.03111 *
gender        0.185713956 0.043792723   4.241          0.000022277 ***
interventions 0.010247319 0.003781234   2.710              0.00673 **
drugs         0.196255771 0.012214620  16.067 < 0.0000000000000002 ***
duration      0.000345292 0.000168573   2.048              0.04053 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 1485.0  on 787  degrees of freedom
Residual deviance: 1044.7  on 781  degrees of freedom
AIC: 3268.1

Number of Fisher Scoring iterations: 5
```

Part C

```r
deviance(model) > qchisq(0.05, df.residual(model), lower.tail = F)
```

```
[1] TRUE
```

Since test statistic is more extreme this is not a good fit.

Part D

```r
dispersiontest(model)
```

```
    Overdispersion test
```

```
data:  model
z = 3.1334, p-value = 0.000864
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  1.327239
```

Since p value less than 0.05 we can say that there is overdispersion.

```r
model.new1 <- glm(visits~., family = quasipoisson, df)
summary(model.new1)
```

```
Call:
glm(formula = visits ~ ., family = quasipoisson, data = df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7010  -1.0393  -0.2316   0.5728   5.7162

Coefficients:
                  Estimate   Std. Error t value             Pr(>|t|)
(Intercept)    0.478391277  0.206041207   2.322             0.020500 *
id             0.000060080  0.000098024   0.613             0.540116
totalcost      0.000014900  0.000003307   4.506          0.00000762 ***
age            0.006691924  0.003424083   1.954             0.051015 .
gender         0.180867765  0.050820913   3.559             0.000395 ***
interventions  0.010149667  0.004412842   2.300             0.021710 *
drugs          0.193380206  0.014632821  13.216 < 0.0000000000000002 ***
complications  0.061969071  0.069369752   0.893             0.371965
comorbidities -0.000920248  0.004256453  -0.216             0.828888
duration       0.000347794  0.000219400   1.585             0.113326
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasipoisson family taken to be 1.332931)

    Null deviance: 1485.0  on 787  degrees of freedom
Residual deviance: 1043.1  on 778  degrees of freedom
AIC: NA
```

```
Number of Fisher Scoring iterations: 5
```

```r
model.new2 <- glm.nb(visits~., df)
summary(model.new2)
```

```
Call:
glm.nb(formula = visits ~ ., data = df, init.theta = 11.96788793,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5256  -0.9474  -0.2110   0.4680   4.4876

Coefficients:
                 Estimate  Std. Error z value             Pr(>|z|)
(Intercept)    0.42446736  0.20467992   2.074             0.038097 *
id             0.00005014  0.00009776   0.513             0.608044
totalcost      0.00001612  0.00000381   4.233            0.0000231 ***
age            0.00744636  0.00340620   2.186             0.028807 *
gender         0.18428100  0.05107467   3.608             0.000308 ***
interventions  0.01103132  0.00498259   2.214             0.026831 *
drugs          0.20901202  0.01652045  12.652 < 0.0000000000000002 ***
complications  0.08416440  0.07587465   1.109             0.267320
comorbidities -0.00098892  0.00423808  -0.233             0.815496
duration       0.00030265  0.00021758   1.391             0.164226
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(11.9679) family taken to be 1)

    Null deviance: 1153.23  on 787  degrees of freedom
Residual deviance:  819.45  on 778  degrees of freedom
AIC: 3241.2

Number of Fisher Scoring iterations: 1

              Theta:  11.97
          Std. Err.:  2.57
```

```
 2 x log-likelihood:  -3219.199
```

```
data.frame(quassi =
             summary(model.new1)$deviance/summary(model.new1)$df.residual,
           negbinom =
             summary(model.new2)$deviance/summary(model.new2)$df.residual) %>%
  mutate(whichbetter = ifelse(quassi > negbinom, "negbinom", "quassi"))
```

```
    quassi negbinom whichbetter
1 1.340769  1.05328    negbinom
```

Model fits better with negative binomial.

Q4

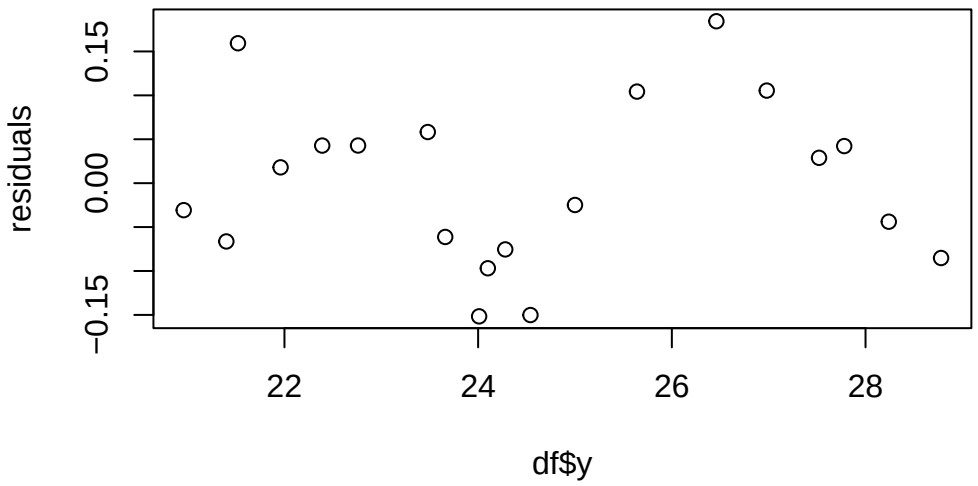
```
df <- read.xlsx("mcgill.xlsx")
kable(head(df))
```

x	y
127.3	20.96
130.0	21.40
132.7	21.96
129.4	21.52
135.0	22.39
137.1	22.76

Part A

```
model <- lm(y~x, df)
residuals <- model$residuals

plot(df$y, residuals)
```



I think there is some seasonal trend almost like $sin(x)$ like.

Part B

```
durbinWatsonTest(model, alternative = "positive")
```

```
 lag Autocorrelation D-W Statistic p-value
   1        0.644368     0.6632531       0
 Alternative hypothesis: rho > 0
```

Since p value is $0 < 0.01$ we reject the null hypothesis so there is positive autocorrelation.

Part C

```
res <- model$residuals
sum(res[2:20]*res[1:19])/sum(res[1:19]^2)
```

```
[1] 0.6729603
```

Part D

```
yt <- df$y[2:20] - 0.6729603*df$y[1:19]
xt <- df$y[2:20] - 0.6729603*df$x[1:19]

transformed_model <- lm(yt~xt)
summary(transformed_model)
```

```
Call:
lm(formula = yt ~ xt)

Residuals:
     Min       1Q   Median       3Q      Max
-0.93484 -0.09657  0.06376  0.21581  0.59464

Coefficients:
            Estimate Std. Error t value    Pr(>|t|)
(Intercept) -0.29232    1.15744  -0.253       0.804
xt          -0.11757    0.01564  -7.515 0.000000846 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4197 on 17 degrees of freedom
Multiple R-squared:  0.7686,    Adjusted R-squared:  0.755
F-statistic: 56.47 on 1 and 17 DF,  p-value: 0.0000008464
```

Part E

```
durbinWatsonTest(transformed_model, alternative = "positive")
```

```
 lag Autocorrelation D-W Statistic p-value
   1      -0.2857218      2.567488   0.844
 Alternative hypothesis: rho > 0
```

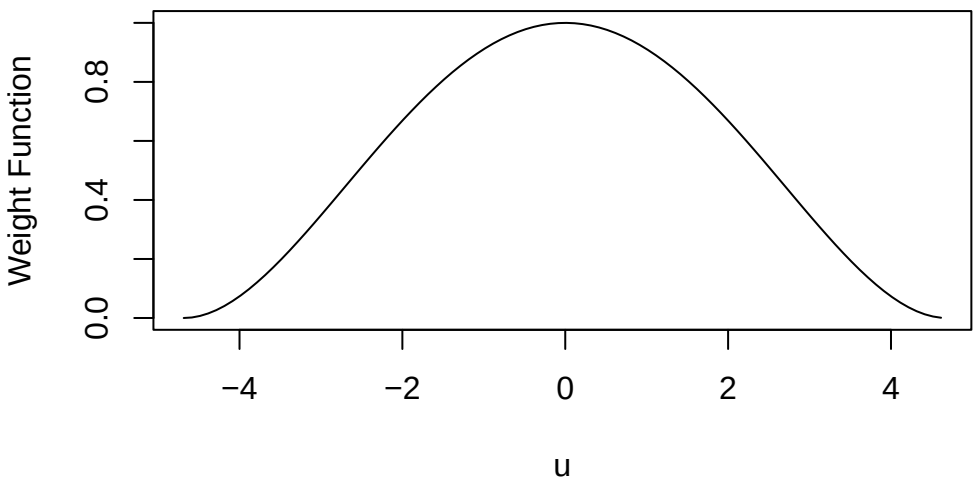
Since p value is now above 0.05 there is no autocorrelation.

Q5

If we have influential/leverage points then we need to weigth each observation where higher the residual lesser the weight. In order to do this we first select a weight function. I will select the bisquare function for example where;

$$\begin{cases} [1-(\frac{u}{4.685})^2]^2 & ,|u| \leq 4.685 \\ 0 & ,|u| > 4.685 \end{cases}$$

```
bisq <- \(x) (1-(x/4.685)^2)^2 #func
u = seq(-4.685,4.685,by=.1) #range
plot(u,bisq(u),type='l',ylab='Weight Function') #plot
```



Values of u near 0 has bigger weights and values farther away has less and less.

Then we initialize the weights. If all weights are same then its the same idea with ordinary least squares

3rdly we do the weighted least squares with the weights we just initialized and get the fitted model. The formula for coefficients then will be

$$b_w = (X'WX)^{-1}X'Wy$$

we use this initial model to get an initial set of residuals. For each residual we scale it via the formula

$$u_i = \frac{e_i}{MAD}$$

where MAD is the Median Absolute Deviation. Which is;

$$MAD = \frac{1}{.6745} median(|e_i - median(e_i)|)$$

- $\frac{1}{.6745}$ is the coefficient which makes mad unbaised.

then we plug our u_i vector into bisq function again to obtain weight vector.

we repeat the last 2 process many times until it starts to stabilize.