Data analysis - 236

9n7u6o

2022-08-05

P - prepare

p.1 - libraries

```
library(tidyverse)
library(MASS)
library(caret)
library(plotly)
library(ggrepel)
```

p.2 - function

Kruskal waillis test and pair wise test

```
kruskal_pairwise<- function(df, full = FALSE, col =3 ){

  g = as.factor(pull(df, col))
  pval = c()
  pairwise<- list()

  for (i in start:ncol(df) ){
    x =pull(df.1, i)
    if(full == TRUE){
      kruskal_result = kruskal.test( x , g)$p.value
      pval  = c(pval, kruskal_result)
    } else {pval = NULL}

    if(full==FALSE){
      pair_result = pairwise.wilcox.test(x, g,
                  p.adjust.method = "bonferroni")
      pairwise[[i-(start-1)]]<- pair_result
    } else{ pairwise = NULL}
  }
  return(list(pval, pairwise))
}
```

generate plot

```
group.colors=c("#999999", "#E69F00", "#56B4E9", "#009E73", "#0072B2", "#D55E00", "#CC79A7")

pred_train_plot<- function(data, fit, col =1, labels =c("a", "b","c") ){
  pred<- coefficients(fit)%*% t(data[,-col])
  train = unlist(data[,col])

  new_d<-  data.frame(train = train,pred = t(pred) )

  info<- new_d %>%
    group_by(train) %>%
    summarise(mean = mean(pred), median = median(pred))

  info.n<- info
  colnames(info.n)<-c("categories", "mean", "median")
  print(info.n)

  info<- info  %>%
    mutate(group = row_number()-1) %>%
    gather(type, value , mean, median, -train)

  gg<- new_d %>% ggplot(aes(x = train, y  = pred, color = train))+
    geom_point() +xlab("Categories")+ylab("Score")+
    scale_color_manual(name = c("Cateogories"),labels =labels, values=group.colors[1:length(leve
ls(train))])

  gg = gg+
    geom_point(data=info, aes(x = train, y = value,shape = type),fill="red", color="red",size =
2)

  print(ggplotly(gg))
  options(ggrepel.max.overlaps = Inf)

  gg<-gg+geom_text_repel(data = new_d,aes(label=round(pred,3)),size = 2)
  print(gg)

}
```

1. Import data

```
file = "Data analyis for Zhen.xlsx"

df.1<- readxl::read_xlsx(file, sheet = 1)
```

```
## New names:
## • `` -> `...15`
## • `` -> `...16`
```

```
df.2<- readxl::read_xlsx(file, sheet = 2)
```

```
## New names:
## • `` -> `...3`
## • `` -> `...16`
## • `` -> `...17`
```

```
df.1<- df.1[,-c(15,16)]
df.2<- df.2[,-c(16,17)]
```

2. Data cleaning

Remove no-use lines and columns

select columns

```
df.1<- df.1[,c(3:5,9:14)]
df.2<- df.2[,c(3:5,10:15)]
```

select rows

```
loc<- which(rowSums(is.na(df.1))>0)
df.1[loc,]
```

```
## # A tibble: 5 × 9
##   `Response type`    `Sample Name` AML S…¹ CD274 CTLA4  EZH2  TIM3  INFG PDCD1…²
##   <chr>              <chr>         <chr>   <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>
## 1 Remission          222-1         Diagno…    NA    NA    NA    NA    NA      NA
## 2 <NA>               <NA>          <NA>       NA    NA    NA    NA    NA      NA
## 3 <NA>               <NA>          <NA>       NA    NA    NA    NA    NA      NA
## 4 <NA>               <NA>          <NA>       NA    NA    NA    NA    NA      NA
## 5 Remission -Relapse 234-3         MRD+       NA    NA    NA    NA    NA      NA
## # … with abbreviated variable names ¹`AML STATUS`, ²PDCD1LG2
```

```
df.1<- df.1[-loc,]
```

```
loc<- which(rowSums(is.na(df.2))>0)
df.2[loc,]
```

```
## # A tibble: 4 × 9
##   ...3  `Analysis group`         Sampl…¹ CD274 CTLA4   EZH2   TIM3  INFG PDCD1…²
##   <chr> <chr>                    <chr>   <dbl> <dbl>  <dbl>  <dbl> <dbl>   <dbl>
## 1 <NA>  <NA>                     <NA>     NA       NA NA      NA        NA   NA
## 2 C     Remission diagnosis      222-1    NA       NA NA      NA        NA   NA
## 3 <NA>  <NA>                     <NA>     NA       NA NA      NA        NA   NA
## 4 <NA>  Persistent Disease day … 257-2    8.02     0  0.113  0.454     0    1.08
## # … with abbreviated variable names ¹`Sample Name`, ²PDCD1LG2
```

```
df.2<- df.2[-loc,]
```

3. Observe the data

plots 2d

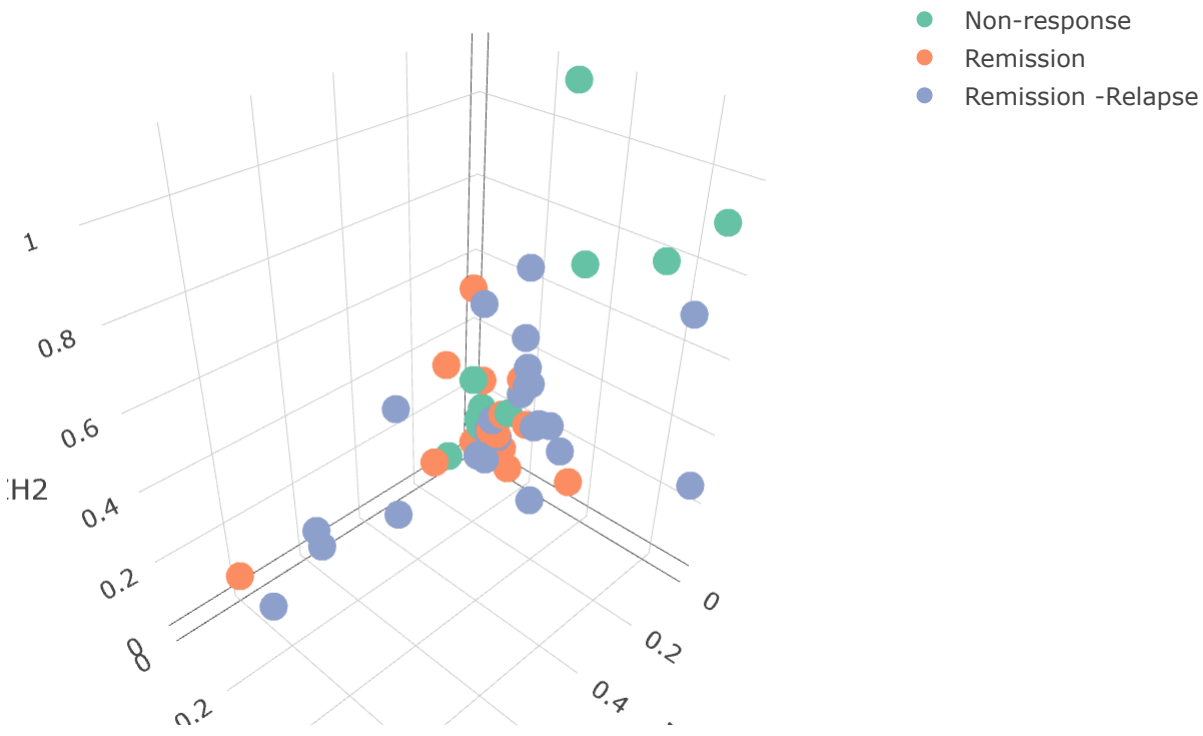
Normality simple check: Somewhat normal distributed groups: CD274: R-R, N-R PDCD1LG2: R, N-R

Other data: heavy lower tails

3d review of the data

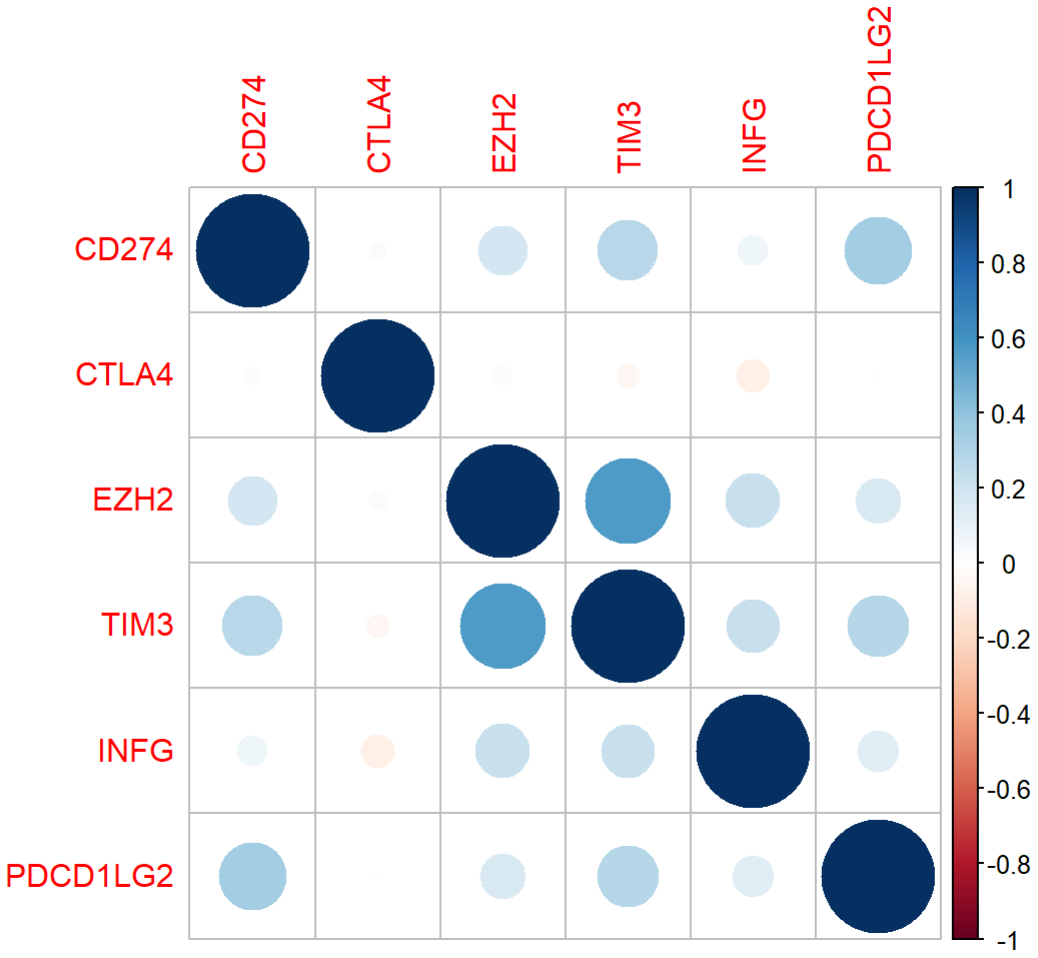
Feel free to change the x,y,z, color, and data to have a quick view of the relationship between groups data and variables

```
fig <- plot_ly(df.1.log, x = ~CTLA4, y = ~EZH2, z = ~TIM3, color = ~`Response type`)
fig <- fig %>% add_markers()
fig <- fig %>% layout(scene = list(xaxis = list(title = 'CD274'),
                     yaxis = list(title = 'CTLA4'),
                     zaxis = list(title = 'EZH2')))
fig
```



correlation plot – low correlation between variables

```
corrplot::corrplot(corr = cor(df.1.log[,c(4:9)]) )
```



4. Hypothesis testing for groups

Are there any differences between groups?

This section will provide a hypothesis test to check if there is strong evidence showing that every two groups are different from each other

Method: pairwise wilcox test

pair wise wilcox test p< 0.05: significant diff between groups

4.1 - df. 1

4.1.1 - AML STATUS

No significant differences between any groups

```
### no significant differences between groups
start = 4
kruskal_pairwise(df.1, col = 3  )[[2]]
```

```
## [[1]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##                Diagnosis MRD- MRD+ MRID-
## MRD-           1.00      -    -    -
## MRD+           1.00      1.00 -    -
## MRID-          1.00      1.00 1.00 -
## Persistant AML 1.00      0.14 1.00 1.00
##
## P value adjustment method: bonferroni
##
## [[2]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##                Diagnosis MRD- MRD+ MRID-
## MRD-           1         -    -    -
## MRD+           1         1    -    -
## MRID-          1         1    1    -
## Persistant AML 1         1    1    1
##
## P value adjustment method: bonferroni
##
## [[3]]
##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data:  x and g
##
##                Diagnosis MRD- MRD+ MRID-
## MRD-           1         -    -    -
## MRD+           1         1    -    -
## MRID-          1         1    1    -
## Persistant AML 1         1    1    1
##
## P value adjustment method: bonferroni
##
## [[4]]
##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data:  x and g
##
##                Diagnosis MRD- MRD+ MRID-
## MRD-           0.14      -    -    -
## MRD+           1.00      1.00 -    -
## MRID-          1.00      1.00 1.00 -
```

```
## Persistant AML 0.94      1.00 1.00 1.00
##
## P value adjustment method: bonferroni
##
## [[5]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##                Diagnosis MRD- MRD+ MRID-
## MRD-           1         -    -    -
## MRD+           1         1    -    -
## MRID-          1         1    1    -
## Persistant AML 1         1    1    1
##
## P value adjustment method: bonferroni
##
## [[6]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##                Diagnosis MRD- MRD+ MRID-
## MRD-           1.00      -    -    -
## MRD+           0.64      0.32 -    -
## MRID-          1.00      1.00 1.00 -
## Persistant AML 1.00      0.50 1.00 1.00
##
## P value adjustment method: bonferroni
```

4.1.2 - Response type

Some groups are significantly different

CD274: Remission vs. Non-response

EZH2: Remission vs. Remission -Relapse

```
### no significant differences between groups
kruskal_pairwise(df.1, col = 1  )[[2]][[1]]
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##                    Non-response Remission
## Remission          0.028        -
## Remission -Relapse 1.000        0.055
##
## P value adjustment method: bonferroni
```

```
kruskal_pairwise(df.1, col = 1  )[[2]][[3]]
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data:  x and g
##
##                    Non-response Remission
## Remission          1.000        -
## Remission -Relapse 1.000        0.041
##
## P value adjustment method: bonferroni
```

4.2 - df.2

4.2.1 - category

No significant differences between any groups

```
kruskal_pairwise(df.2, col = 1  )[[2]]
```

```
## [[1]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##   A    B    C    D
## B 1.00 -    -    -
## C 1.00 0.82 -    -
## D 1.00 1.00 1.00 -
## E 1.00 0.32 1.00 1.00
##
## P value adjustment method: bonferroni
##
## [[2]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##   A B C D
## B 1 - - -
## C 1 1 - -
## D 1 1 1 -
## E 1 1 1 1
##
## P value adjustment method: bonferroni
##
## [[3]]
##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data:  x and g
##
##   A B C D
## B 1 - - -
## C 1 1 - -
## D 1 1 1 -
## E 1 1 1 1
##
## P value adjustment method: bonferroni
##
## [[4]]
##
##  Pairwise comparisons using Wilcoxon rank sum exact test
##
## data:  x and g
##
##   A B C D
## B 1 - - -
## C 1 1 - -
## D 1 1 1 -
```

```
## E 1 1 1 1
##
## P value adjustment method: bonferroni
##
## [[5]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##   A    B    C    D
## B 1.00 -    -    -
## C 1.00 1.00 -    -
## D 1.00 1.00 1.00 -
## E 0.64 0.64 1.00 1.00
##
## P value adjustment method: bonferroni
##
## [[6]]
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  x and g
##
##   A    B    C    D
## B 1.00 -    -    -
## C 0.70 0.49 -    -
## D 1.00 1.00 1.00 -
## E 1.00 1.00 1.00 1.00
##
## P value adjustment method: bonferroni
```

5. Prediction methods

Method multinational logistic (polytomous Logistic Regression)

Logistic methods and interpretation:

$\pi = \frac{P(x)}{1-P(x)}$, $P(x)$ is the probability of a event x (a patient categorized as response / Diagnosis / A) occur

Logistic regression function:

$$logit(\pi) = log(\frac{\pi}{1-\pi}) = X'\beta$$

Function transformation:

$$\pi = \frac{P(x)}{1-P(x)} = e^{X'\beta}$$

$$P(x) = \frac{e^{X'\beta}}{1+e^{X'\beta}}$$

5.1 Assign socres for each class

class1: 1

class2: 2

…

classn: n

```
df.1<- df.1 %>% mutate(`AML STATUS` =ifelse(`AML STATUS` ==  "MRID-" , "MRD-", `AML STATUS` )  )

df.1$`Response type` <- factor(df.1$`Response type`,
                               levels = c("Non-response", "Remission"   ,"Remission -Relapse" ),
                               labels = c("0","1","2"), ordered = T)



df.1$`AML STATUS`<- factor(df.1$`AML STATUS`,
                            levels = c( "Diagnosis","Persistant AML", "MRD-", "MRD+" ),
                           labels = c("0","1","2", "3"), ordered = T)


df.2$...3 <- factor(df.2$...3,
                               levels = c( "A", "B", "C", "D", "E" ),
                               labels = c("0","1","2", "3", "4"), ordered = T)
```

5.2 - df1

5.2.1 - AML STATUS

AML STATUS	Score
Diagnosis	0
Persistant AML	1
MRD-	2
MRD+	3

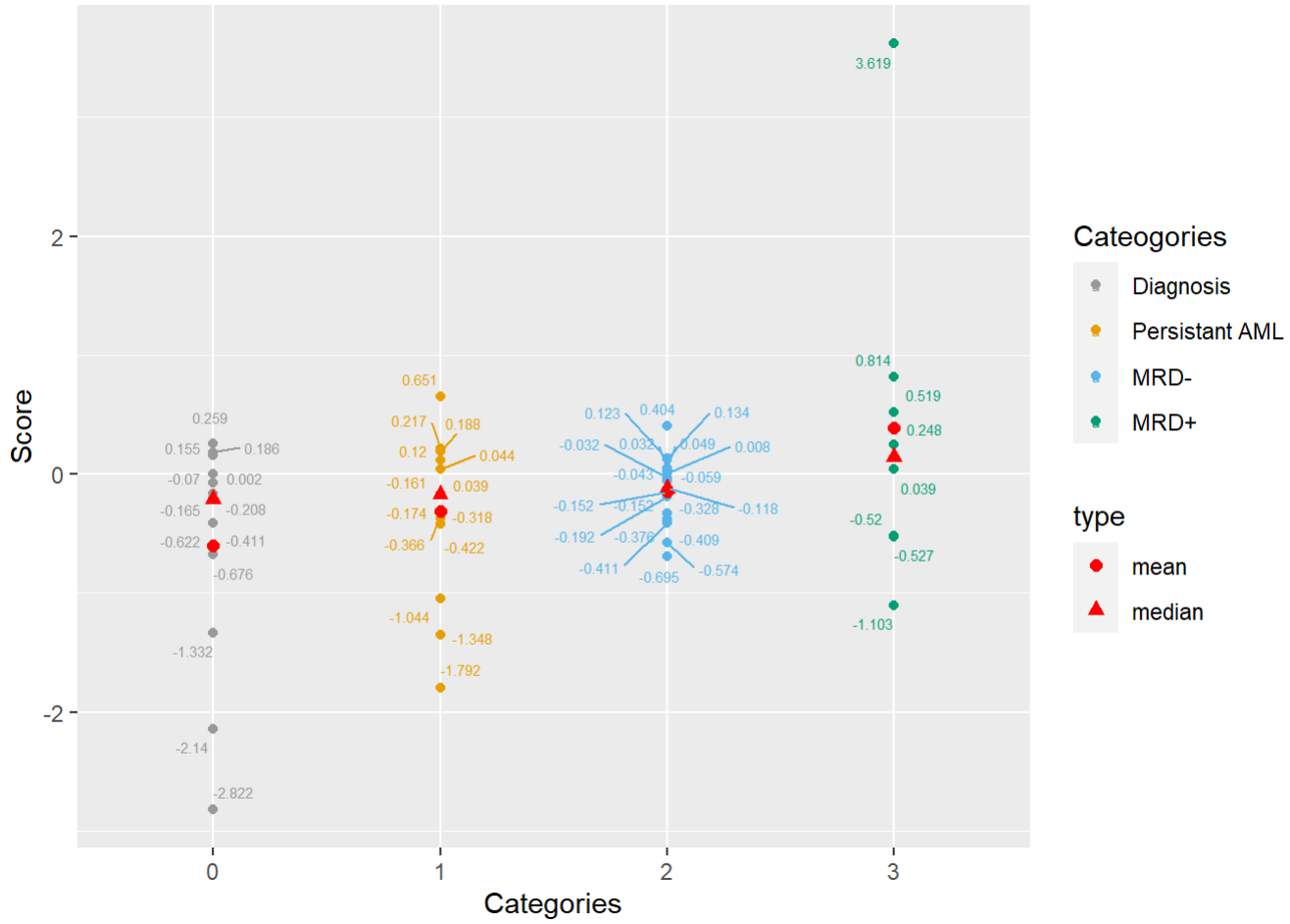
```
data =(df.1[,c(3, 4:9)])
r.fit = polr(data$`AML STATUS` ~ ., data = data)

summary(r.fit, digits = 3)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = data$`AML STATUS` ~ ., data = data)
##
## Coefficients:
##            Value Std. Error t value
## CD274    -0.0126     0.0344  -0.365
## CTLA4     0.1030     0.3046   0.338
## EZH2     -0.7911     0.4961  -1.595
## TIM3      0.2246     0.3058   0.734
## INFG     -0.0524     0.0387  -1.353
## PDCD1LG2  0.1178     0.0811   1.453
##
## Intercepts:
##     Value  Std. Error t value
## 0|1 -1.505  0.518     -2.904
## 1|2 -0.241  0.453     -0.532
## 2|3  1.676  0.537      3.119
##
## Residual Deviance: 136.4506
## AIC: 154.4506
```

```
pred_train_plot(data, r.fit,1,  labels = c( "Diagnosis","Persistant AML", "MRD-", "MRD+"   ))
```

```
## # A tibble: 4 × 3
##   categories   mean median
##   <fct>       <dbl>  <dbl>
## 1 0          -0.603 -0.208
## 2 1          -0.312 -0.167
## 3 2          -0.147 -0.118
## 4 3           0.386  0.143
```



5.2.1.a- remove CTLA4

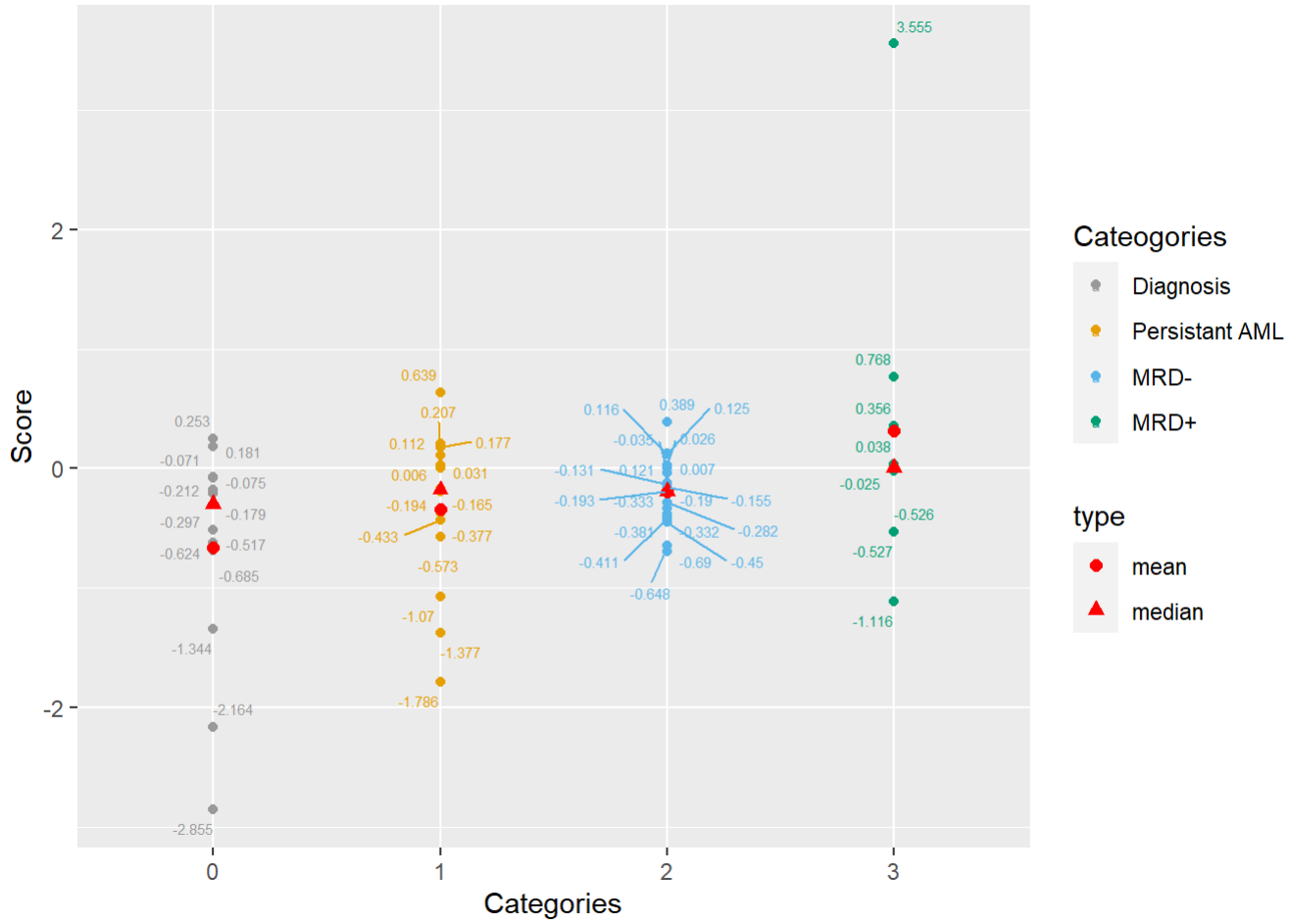
```
r.fit = polr(data$`AML STATUS` ~ ., data = data[,-3])

summary(r.fit, digits = 3)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = data$`AML STATUS` ~ ., data = data[, -3])
##
## Coefficients:
##            Value Std. Error t value
## CD274    -0.0131     0.0344  -0.381
## EZH2     -0.7841     0.4938  -1.588
## TIM3      0.2162     0.3045   0.710
## INFG     -0.0527     0.0386  -1.365
## PDCD1LG2  0.1171     0.0810   1.446
##
## Intercepts:
##     Value  Std. Error t value
## 0|1 -1.549  0.503     -3.083
## 1|2 -0.285  0.433     -0.658
## 2|3  1.625  0.514      3.162
##
## Residual Deviance: 136.5662
## AIC: 152.5662
```

```
pred_train_plot(data[,-3], r.fit,1,  labels = c( "Diagnosis","Persistant AML", "MRD-", "MRD+"))
```

```
## # A tibble: 4 × 3
##   categories   mean   median
##   <fct>       <dbl>    <dbl>
## 1 0          -0.661 -0.297
## 2 1          -0.343 -0.179
## 3 2          -0.194 -0.190
## 4 3           0.315  0.00644
```



5.2.1.b- remove CTLA4 and INFG

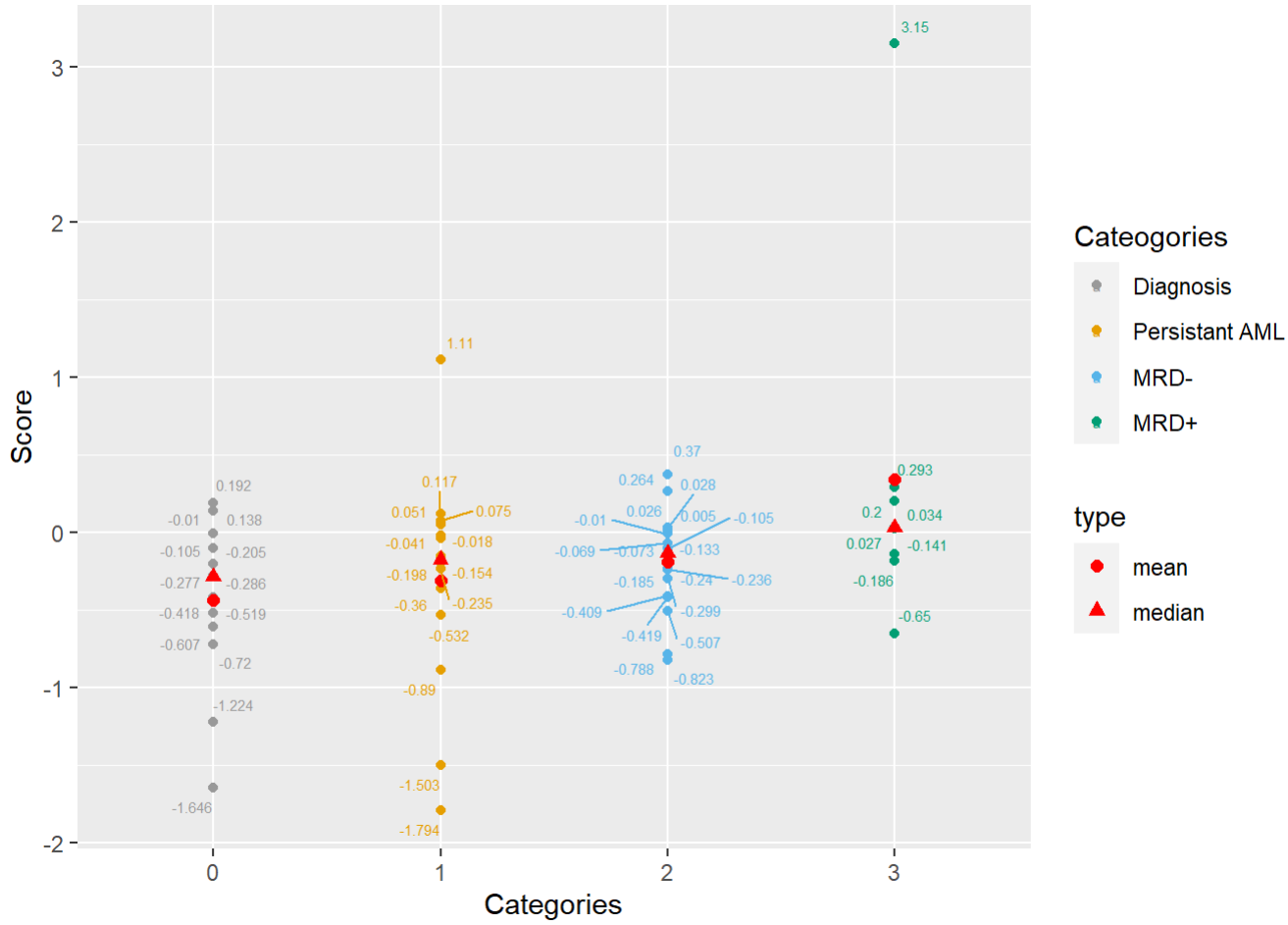
```
r.fit = polr(data$`AML STATUS` ~ ., data = data[,-c(3,6)])

summary(r.fit)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = data$`AML STATUS` ~ ., data = data[, -c(3, 6)])
##
## Coefficients:
##             Value Std. Error t value
## CD274    -0.01707    0.03388 -0.5040
## EZH2     -0.85338    0.49550 -1.7223
## TIM3      0.14672    0.31146  0.4711
## PDCD1LG2  0.11625    0.07629  1.5238
##
## Intercepts:
##     Value   Std. Error t value
## 0|1 -1.4665  0.4872    -3.0099
## 1|2 -0.2422  0.4234    -0.5721
## 2|3  1.6449  0.5042     3.2622
##
## Residual Deviance: 138.5967
## AIC: 152.5967
```

```
pred_train_plot(data[,-c(3,6)], r.fit,1,  labels = c( "Diagnosis","Persistant AML", "MRD-", "MRD
+"))
```

```
## # A tibble: 4 × 3
##   categories   mean  median
##   <fct>       <dbl>   <dbl>
## 1 0          -0.437 -0.286
## 2 1          -0.312 -0.176
## 3 2          -0.190 -0.133
## 4 3           0.341  0.0304
```



5.2.2 - Remission

Non-response: 1 Remission Remission -Relapse

Response type	score
Non-response	0
Remission	1
Remission -Relapse	2

```
data =(df.1[,c(1,3, 4:9)])

r.fit<- polr(`Response type` ~ ., data = data[,c(-2)])
coef(r.fit)
```

```
##       CD274       CTLA4        EZH2        TIM3        INFG    PDCD1LG2
##  0.03541407  0.42269383  1.02707127 -1.09036478  0.02341016  0.10791693
```

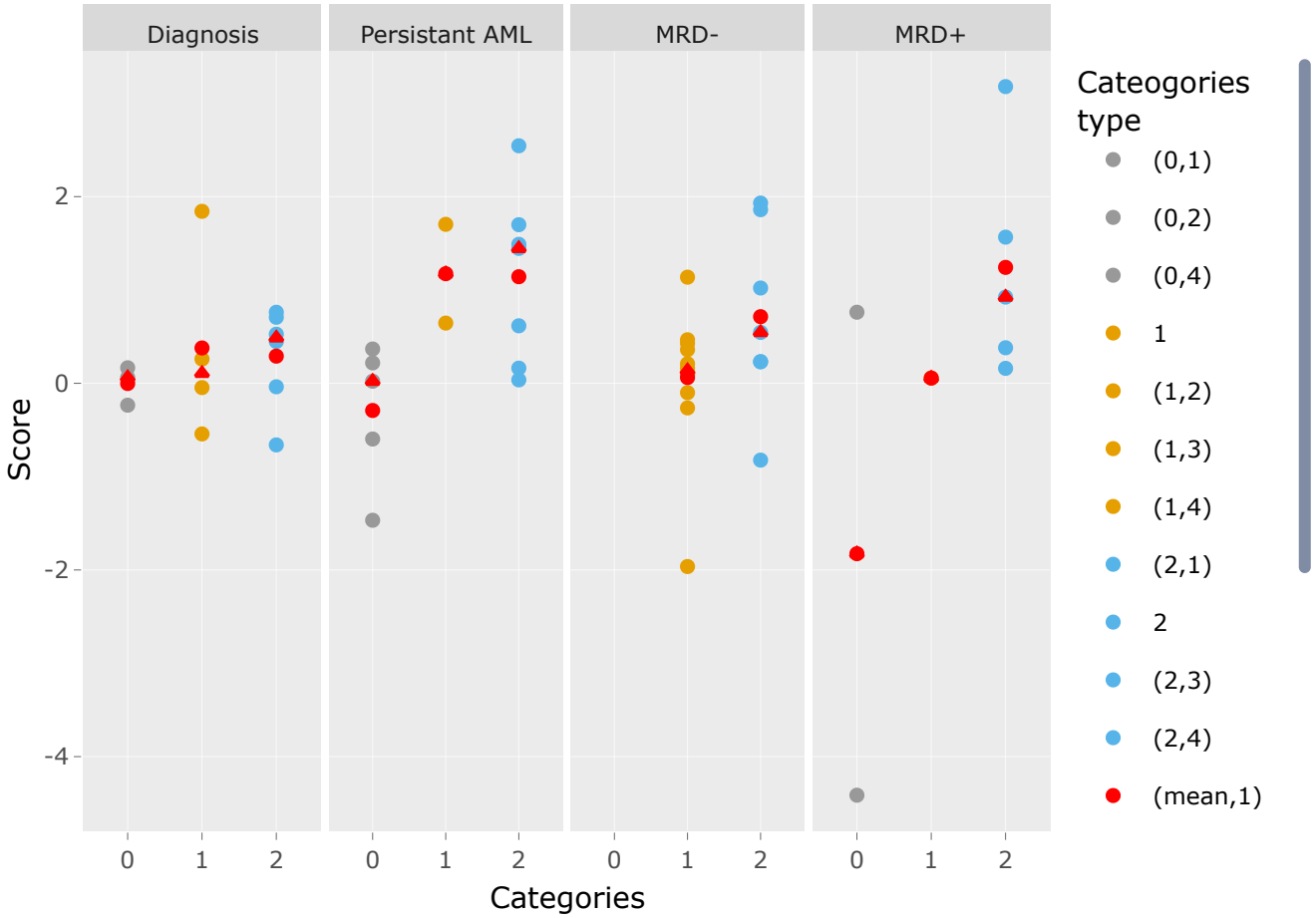
```
data = data
fit = r.fit
labels  =  c("Non-response", "Remission"   ,"Remission -Relapse" )
facet_label = c(`0`=  "Diagnosis",
                `1`= "Persistant AML",
                `2`="MRD-",
                `3`="MRD+")
col= c(1,2)

pred<- coefficients(fit)%*% t(data[,-col ])
train = (data[,col])

new_d<-  data.frame(train = train,pred = t(pred) )
colnames(new_d) =c( "Response.type", "AML.STATUS", "val")

info<- new_d %>%
  group_by(Response.type, AML.STATUS) %>%
  summarise(mean = mean(val), median = median(val)) %>%
  mutate(group = row_number()-1)
```

```
## `summarise()` has grouped output by 'Response.type'. You can override using the
## `.groups` argument.
```

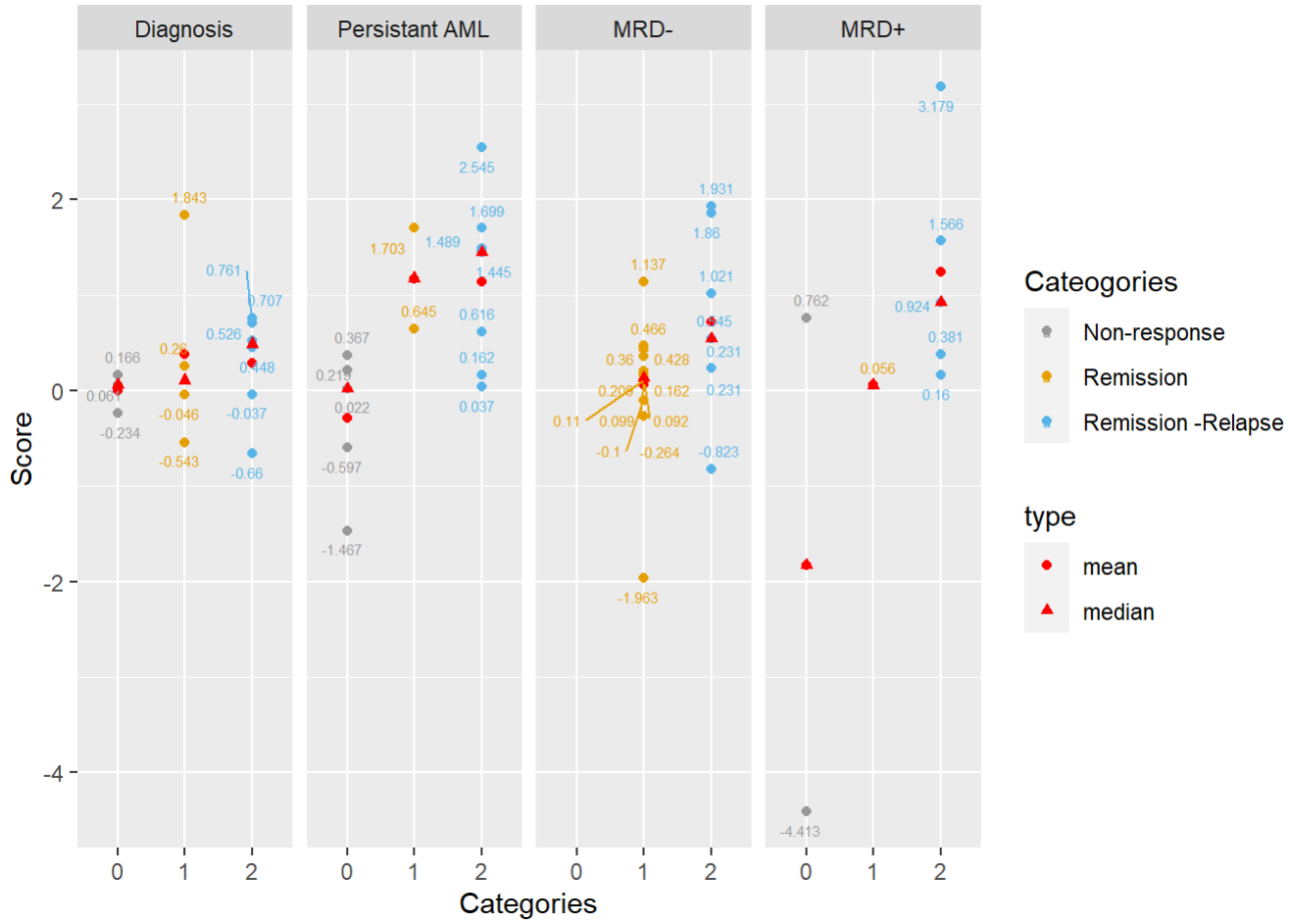
```
info %>% select(-group)
```

```
## # A tibble: 11 × 4
## # Groups:   Response.type [3]
##    Response.type AML.STATUS     mean  median
##    <ord>         <ord>         <dbl>   <dbl>
##  1 0             0          -0.00234  0.0611
##  2 0             1          -0.291    0.0224
##  3 0             3          -1.83    -1.83
##  4 1             0           0.378    0.107
##  5 1             1           1.17     1.17
##  6 1             2           0.0615   0.136
##  7 1             3           0.0558   0.0558
##  8 2             0           0.291    0.487
##  9 2             1           1.14     1.45
## 10 2             2           0.714    0.545
## 11 2             3           1.24     0.924
```

```
info<- info %>%
  gather(type, value , mean, median,-group)
options(ggrepel.max.overlaps = Inf)
gg<- ggplot(data= new_d,
            aes(x = Response.type, y = val, color =Response.type ))+
  geom_point()+
  scale_color_manual(name = c("Cateogories"),
                     labels =labels,
                     values=group.colors[1:3])+
  facet_grid(~AML.STATUS, labeller =as_labeller(facet_label)) +
  xlab("Categories")+ylab("Score")

gg<- gg+geom_point(data = info,
                   aes(x = Response.type, y = value, shape = type),
                   color = "red")

ggplotly(gg)
```




```
gg+
  geom_text_repel(aes(label=round(val,3)),size = 2)
```



5.3 - df.2

5.3.1 - Category

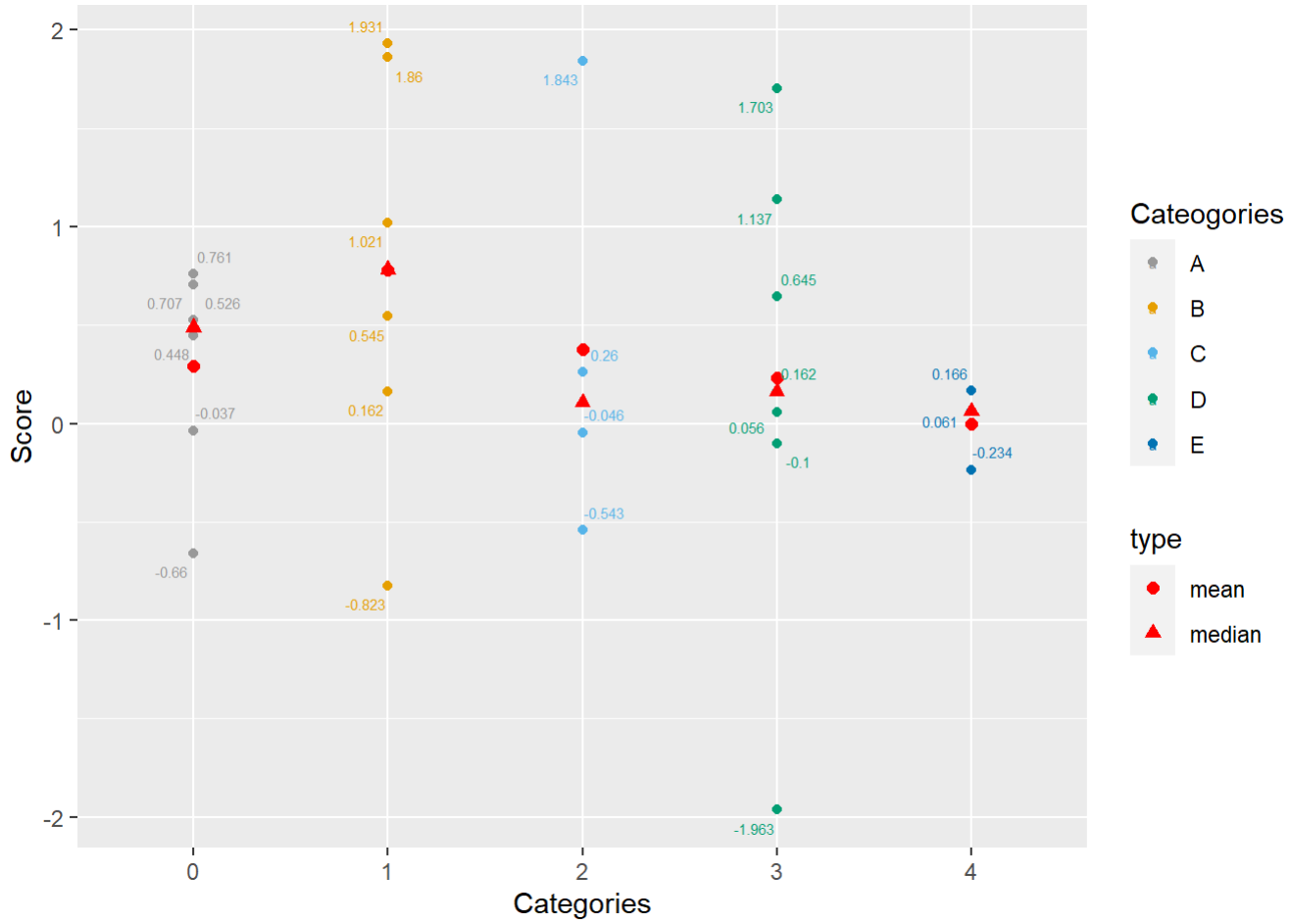
CATEGORY	Score
A	0
B	1
C	2
D	3
E	4

```
data =df.2[,c(1,4:9)]

polr(data$...3 ~ ., data = data)
```

```
## Call:
## polr(formula = data$...3 ~ ., data = data)
##
## Coefficients:
##       CD274       CTLA4        EZH2        TIM3        INFG    PDCD1LG2
## -0.25981110 -0.10478422 -0.98575383  0.04394299 -0.01137643  0.98728861
##
## Intercepts:
##        0|1        1|2        2|3        3|4
## -1.9169079 -0.4848115  0.3423635  2.1803313
##
## Residual Deviance: 70.75222
## AIC: 90.75222
```

```
pred_train_plot(data, r.fit,1, labels  = c( "A", "B", "C", "D", "E" ))
```

```
## # A tibble: 5 × 3
##   categories     mean median
##   <fct>         <dbl>  <dbl>
## 1 0           0.291   0.487
## 2 1           0.783   0.783
## 3 2           0.378   0.107
## 4 3           0.234   0.162
## 5 4          -0.00234 0.0611
```



6. Limitation of this simple logistic regression:

Logistic regression does not provide a very well prediction for the data, but still could be used. Training error might be high.

I used the simplest logistic regression methods. In the future, interaction terms could be introduced into the logistic regression to improve the prediction

other methods I considered using but was not able to perform due to the small sample size or the complexity of the expression functions:

1. lda
2. qda
3. random forest
4. knn could also be performed, but it does not provide a very straightforward function for each group.