Tone Classification

Giang Le

10/28/2021

Data Import and Preprocessing

Our data contains 15 voice reports from 15 recording sessions.

```
# Read in all the voice reports.
dataFiles <- lapply(Sys.glob("*/channel1/acoustic_measurements_unique_*.csv"), read.csv)
```

```
## Add the following categorical predictors.
# Gender: F and M (done)
# Noise type: quiet, 78 or 90 (done)
# Single token or token in a sentence
# Syllable type
# Tone

# Converting to DataFrames

f_1_78 <- as.data.frame(dataFiles[1])
f_1_90 <- as.data.frame(dataFiles[2])
f_1_q <- as.data.frame(dataFiles[3])

f_2_78 <- as.data.frame(dataFiles[4])
f_2_90 <- as.data.frame(dataFiles[5])
f_2_q <- as.data.frame(dataFiles[6])

m_1_78 <- as.data.frame(dataFiles[7])
m_1_90 <- as.data.frame(dataFiles[8])
m_1_q <- as.data.frame(dataFiles[9])

m_2_78 <- as.data.frame(dataFiles[10])
m_2_90 <- as.data.frame(dataFiles[11])
m_2_q <- as.data.frame(dataFiles[12])

m_3_78 <- as.data.frame(dataFiles[13])
m_3_90 <- as.data.frame(dataFiles[14])
m_3_q <- as.data.frame(dataFiles[15])
```

```
# Assigning gender variable (0 for female and 1 for male)
f_1_78$gender = 0
f_1_90$gender = 0
f_1_q$gender = 0

f_2_78$gender = 0
f_2_90$gender = 0
```

```
f_2_q$gender = 0

m_1_78$gender = 1
m_1_90$gender = 1
m_1_q$gender = 1

m_2_78$gender = 1
m_2_90$gender = 1
m_2_q$gender = 1

m_3_78$gender = 1
m_3_90$gender = 1
m_3_q$gender = 1

# Assigning noise level
f_1_78$noise = 78
f_1_90$noise = 90
f_1_q$noise = 0


f_2_78$noise = 78
f_2_90$noise = 90
f_2_q$noise = 0

m_1_78$noise = 78
m_1_90$noise = 90
m_1_q$noise = 0

m_2_78$noise = 78
m_2_90$noise = 90
m_2_q$noise = 0

m_3_78$noise = 78
m_3_90$noise = 90
m_3_q$noise = 0

# Assigning speaker code
f_1_78$speaker = "f-1"
f_1_90$speaker = "f-1"
f_1_q$speaker = "f-1"

f_2_78$speaker = "f-2"
f_2_90$speaker = "f-2"
f_2_q$speaker = "f-2"

m_1_78$speaker = "m-1"
m_1_90$speaker = "m-1"
m_1_q$speaker = "m-1"

m_2_78$speaker = "m-2"
m_2_90$speaker = "m-2"
m_2_q$speaker = "m-2"
```

```r
m_3_78$speaker = "m-3"
m_3_90$speaker = "m-3"
m_3_q$speaker = "m-3"
```

```r
### Concatenate all dataframes
voice_reports <- rbind(f_1_78, f_1_90, f_1_q,
                       f_2_78, f_2_90, f_2_q,
                       m_1_78, m_1_90, m_1_q,
                       m_2_78,m_2_90, m_2_q,
                       m_3_78, m_3_90, m_3_q)




## Drop intervals that don't matter
voice_reports <- voice_reports[!(endsWith(voice_reports$sound.name,"_")),]
dim(voice_reports)
```

```
## [1] 4732   37
```

```r
head(voice_reports,10)
```

```
##    sound.name total.duration intensity spectraltilt median.F0 mean.F0   sd.F0
## 2    11651_độ          0.229    60.102      -26.448    218.18 206.701  35.194
## 4    11659_độ          0.176    63.003      -27.852   207.319 209.281  18.505
## 6     11667_ễ          0.444    61.756      -14.646   235.861 188.774  72.344
## 8     11675_ễ          0.330    59.292      -11.698    224.04 224.782  12.365
## 10    11683_ê          0.430    60.768       -9.277   221.367 220.179   3.259
## 12    11691_ê          0.450    62.075      -10.194   216.535 217.309   3.446
## 14    11699_ệ          0.252    61.200      -12.242   214.422  264.94 135.012
## 16    11707_ệ          0.173    62.570      -11.646   216.864  213.11   7.844
## 18    11715_ể          0.469    60.697      -15.597   168.827 171.449  19.435
## 20    11723_ể          0.235    60.548      -16.422   166.075 169.669   14.28
##     min.F0  max.F0 number.pulses number.periods mean.periods sd.period
## 2   92.034 258.787            34             32        4.887     1.191
## 4  185.116 255.695            32             31        4.901      0.63
## 6  102.959 266.779            77             74        5.171      2.19
## 8  199.986 241.525            69             68        4.447     0.248
## 10  206.74 224.333            90             89        4.547     0.089
## 12  203.63 222.407            92             91        4.599      0.07
## 14 185.925 588.058            48             46        4.023     1.671
## 16 191.405 218.541            26             25        4.709     0.239
## 18 146.118 203.461            67             65        5.875     0.676
## 20 149.693 194.118            38             37        5.881     0.493
##    fraction.of.locally.unvoiced.frames
## 2                                8.333
## 4                                5.556
## 6                                4.000
## 8                                0.000
## 10                               0.000
## 12                               0.000
## 14                              18.519
## 16                              11.765
## 18                               9.434
## 20                               0.000
##                                           fraction number.of.voice.breaks
## 2   of locally unvoiced frames: 8.333%   (2 / 24)                      1
## 4   of locally unvoiced frames: 5.556%   (1 / 18)                      0
## 6   of locally unvoiced frames: 4.000%   (2 / 50)                      2
## 8        of locally unvoiced frames: 0   (0 / 36)                      0
## 10       of locally unvoiced frames: 0   (0 / 48)                      0
## 12       of locally unvoiced frames: 0   (0 / 50)                      0
## 14 of locally unvoiced frames: 18.519%   (5 / 27)                      0
## 16 of locally unvoiced frames: 11.765%   (2 / 17)                      0
## 18  of locally unvoiced frames: 9.434%   (5 / 53)                      1
## 20       of locally unvoiced frames: 0   (0 / 25)                      0
##    degree.of.voice.breaks
## 2                  22.773
## 4                   0.000
## 6                   7.426
## 8                   0.000
## 10                  0.000
## 12                  0.000
## 14                  0.000
## 16                  0.000
## 18                  7.559
## 20                  0.000
##                                                              degree
## 2  of voice breaks: 22.773%   (0.052155 seconds / 0.229018 seconds)
## 4               of voice breaks: 0   (0 seconds / 0.176274 seconds)
## 6   of voice breaks: 7.426%   (0.032980 seconds / 0.444099 seconds)
## 8               of voice breaks: 0   (0 seconds / 0.330119 seconds)
## 10              of voice breaks: 0   (0 seconds / 0.430139 seconds)
## 12              of voice breaks: 0   (0 seconds / 0.449857 seconds)
## 14              of voice breaks: 0   (0 seconds / 0.251984 seconds)
## 16              of voice breaks: 0   (0 seconds / 0.173127 seconds)
## 18  of voice breaks: 7.559%   (0.035468 seconds / 0.469197 seconds)
## 20              of voice breaks: 0   (0 seconds / 0.234767 seconds)
##    jitter.local jitter.local.abs jitter.rap jitter.ppq5 shimmer.local
## 2         2.926          142.986      1.605       2.147         8.542
## 4         6.202          303.958      4.035       3.564         8.893
## 6         1.554           80.339      0.477       0.552         4.136
## 8         0.626           27.832      0.241       0.253         1.327
## 10        0.411           18.682      0.148       0.159         1.432
## 12        0.449           20.659      0.222       0.175         1.752
## 14        3.447          138.645      0.889       0.951         6.189
## 16        1.245           58.639      0.272       0.435         2.246
## 18        2.373          139.393       0.93        0.57         5.078
## 20        1.864          109.602      0.884       1.008         5.029
##    shimmer.local.db shimmer.apq3 shimmer.apq5 shimmer.apq11 mean.autocorr
## 2               1.57         3.89         3.46         3.798         0.873
## 4              0.955        4.002        2.859         5.278          0.91
## 6              0.622        1.215        1.536         2.945         0.895
## 8              0.117        0.543        0.626          1.38          0.98
## 10             0.133        0.458        0.619         0.692         0.985
## 12             0.163        0.866        0.784         0.979         0.987
## 14             1.027        2.381        3.367         6.287         0.883
## 16             0.196        0.795        1.169         1.605         0.931
## 18             0.576        0.956        1.288         3.289         0.945
## 20             0.652         1.61        1.973          2.39         0.969
##    mean.NHR mean.HNR      F1       F2       F3       F4 gender noise speaker
## 2     0.193   12.731 495.881  853.512 3303.512 3880.690      0    78     f-1
## 4     0.145   16.212 484.495  894.356 3197.222 3737.516      0    78     f-1
## 6     0.161   15.621 558.530 2325.318 2993.339 4157.486      0    78     f-1
## 8     0.021   20.469 623.327 2293.245 2967.519 4045.199      0    78     f-1
## 10    0.017   25.995 559.097 2350.072 3034.048 4073.220      0    78     f-1
## 12    0.018   27.001 485.833 2290.109 2928.457 4066.557      0    78     f-1
## 14    0.173   12.823 613.222 2324.702 2985.978 4142.649      0    78     f-1
## 16    0.093   17.098 600.641 2282.071 2943.010 4134.439      0    78     f-1
## 18    0.076   15.586 553.022 2233.089 2817.136 4024.624      0    78     f-1
## 20    0.035   17.656 548.293 2310.878 2847.136 4136.784      0    78     f-1
```

```
# Assigning if the token is single (1) or not (0).
voice_reports$single <- ifelse(grepl("single", voice_reports$sound.name), 1, 0)

# Assign syllable shapes (do later)


# Assign tone values
# voice_reports$tone <- ifelse(grepl("a", voice_reports$sound.name, ignore.case=T), "A1",
#                       ifelse(grepl("à", voice_reports$sound.name, ignore.case=T), "A2",
#                       ifelse(grepl("á", voice_reports$sound.name, ignore.case=T), "B1",
#                       ifelse(grepl("ả", voice_reports$sound.name, ignore.case=T), "C1",
#                       ifelse(grepl("ã", voice_reports$sound.name, ignore.case=T), "C2",
#                       ifelse(grepl("ạ", voice_reports$sound.name, ignore.case=T), "B2",
#                       ifelse(grepl("ê", voice_reports$sound.name, ignore.case=T), "A1",
#                       ifelse(grepl("ề", voice_reports$sound.name, ignore.case=T), "A2",
#                       ifelse(grepl("ế", voice_reports$sound.name, ignore.case=T), "B1",
#                       ifelse(grepl("ể", voice_reports$sound.name, ignore.case=T), "C1",
#                       ifelse(grepl("ễ", voice_reports$sound.name, ignore.case=T), "C2",
#                       ifelse(grepl("ệ", voice_reports$sound.name, ignore.case=T), "B2",
#                       ifelse(grepl("u", voice_reports$sound.name, ignore.case=T), "A1",
#                       ifelse(grepl("ù", voice_reports$sound.name, ignore.case=T), "A2",
#                       ifelse(grepl("ú", voice_reports$sound.name, ignore.case=T), "B1",
#                       ifelse(grepl("ủ", voice_reports$sound.name, ignore.case=T), "C1",
#                       ifelse(grepl("ũ", voice_reports$sound.name, ignore.case=T), "C2",
#                       ifelse(grepl("ụ", voice_reports$sound.name, ignore.case=T), "B2",
#                       ifelse(grepl("ộ", voice_reports$sound.name, ignore.case=T), "B2","NA"))))))))))

voice_reports$tone <- ifelse(grepl("a", voice_reports$sound.name, ignore.case=T), "A1",
ifelse(grepl("_tát", voice_reports$sound.name, ignore.case=T), "D1",
ifelse(grepl("_tạt", voice_reports$sound.name, ignore.case=T), "D2",
ifelse(grepl("_tết", voice_reports$sound.name, ignore.case=T), "D1",
ifelse(grepl("_tệt", voice_reports$sound.name, ignore.case=T), "D2",
ifelse(grepl("_tút", voice_reports$sound.name, ignore.case=T), "D1",
ifelse(grepl("_tụt", voice_reports$sound.name, ignore.case=T), "D2",
ifelse(grepl("à", voice_reports$sound.name, ignore.case=T), "A2",
ifelse(grepl("á", voice_reports$sound.name, ignore.case=T), "B1",
ifelse(grepl("ả", voice_reports$sound.name, ignore.case=T), "C1",
ifelse(grepl("ã", voice_reports$sound.name, ignore.case=T), "C2",
ifelse(grepl("ạ", voice_reports$sound.name, ignore.case=T), "B2",
ifelse(grepl("ê", voice_reports$sound.name, ignore.case=T), "A1",
ifelse(grepl("ề", voice_reports$sound.name, ignore.case=T), "A2",
ifelse(grepl("ế", voice_reports$sound.name, ignore.case=T), "B1",
```

```r
ifelse(grepl("ể", voice_reports$sound.name, ignore.case=T), "C1",
ifelse(grepl("ễ", voice_reports$sound.name, ignore.case=T), "C2",
ifelse(grepl("ệ", voice_reports$sound.name, ignore.case=T), "B2",
ifelse(grepl("u", voice_reports$sound.name, ignore.case=T), "A1",
ifelse(grepl("ù", voice_reports$sound.name, ignore.case=T), "A2",
ifelse(grepl("ú", voice_reports$sound.name, ignore.case=T), "B1",
ifelse(grepl("ủ", voice_reports$sound.name, ignore.case=T), "C1",
ifelse(grepl("ũ", voice_reports$sound.name, ignore.case=T), "C2",
ifelse(grepl("ụ", voice_reports$sound.name, ignore.case=T), "B2",
ifelse(grepl("ộ", voice_reports$sound.name, ignore.case=T), "B2","NA")))))))))))))))))))))))))

# Assign phonation types
voice_reports$phonation <- ifelse(grepl("A1", voice_reports$tone, ignore.case=T), "modal",
                      ifelse(grepl("A2", voice_reports$tone, ignore.case=T), "breathy",
                      ifelse(grepl("B1", voice_reports$tone, ignore.case=T), "modal",
                      ifelse(grepl("B2", voice_reports$tone, ignore.case=T), "creaky",
                      ifelse(grepl("C1", voice_reports$tone, ignore.case=T), "creaky",
                      ifelse(grepl("C2", voice_reports$tone, ignore.case=T), "creaky","NA"))))))


# Assign creakiness or not
voice_reports$creaky <- ifelse(grepl("creaky", voice_reports$phonation, ignore.case=T), 1, 0)
voice_reports$creaky <- as.factor(voice_reports$creaky)
head(voice_reports, 20)
```

```
##          sound.name total.duration intensity spectraltilt median.F0 mean.F0
## 2          11651_độ          0.229    60.102      -26.448    218.18 206.701
## 4          11659_độ          0.176    63.003      -27.852   207.319 209.281
## 6           11667_ễ          0.444    61.756      -14.646   235.861 188.774
## 8           11675_ễ          0.330    59.292      -11.698    224.04 224.782
## 10          11683_ê          0.430    60.768       -9.277   221.367 220.179
## 12          11691_ê          0.450    62.075      -10.194   216.535 217.309
## 14          11699_ệ          0.252    61.200      -12.242   214.422  264.94
## 16          11707_ệ          0.173    62.570      -11.646   216.864  213.11
## 18          11715_ể          0.469    60.697      -15.597   168.827 171.449
## 20          11723_ể          0.235    60.548      -16.422   166.075 169.669
## 22          11731_ế          0.419    61.703      -15.343   187.562 205.239
## 24          11739_ế          0.382    60.027      -13.377   180.888  191.02
## 26          11747_ề          0.537    61.288      -13.552   178.122 177.682
## 28          11755_ề          0.450    61.082      -11.035   181.129 180.049
## 30 11763_TÚT_single          0.175    64.908      -26.815   263.384 260.847
## 32 11771_TÚT_single          0.183    65.207      -30.200   251.906 254.757
## 34 11779_TỤT_single          0.194    65.341      -33.129   197.522   195.4
## 36 11787_TỤT_single          0.229    63.707      -31.444   193.629 194.373
## 38        11795_thề          0.417    61.015      -23.340   173.842 148.532
## 40        11803_thề          0.426    62.959      -18.748   174.036 175.839
##      sd.F0  min.F0  max.F0 number.pulses number.periods mean.periods sd.period
## 2   35.194  92.034 258.787            34             32        4.887     1.191
## 4   18.505 185.116 255.695            32             31        4.901      0.63
## 6   72.344 102.959 266.779            77             74        5.171      2.19
## 8   12.365 199.986 241.525            69             68        4.447     0.248
## 10   3.259  206.74 224.333            90             89        4.547     0.089
## 12   3.446  203.63 222.407            92             91        4.599      0.07
## 14 135.012 185.925 588.058            48             46        4.023     1.671
## 16   7.844 191.405 218.541            26             25        4.709     0.239
## 18  19.435 146.118 203.461            67             65        5.875     0.676
## 20   14.28 149.693 194.118            38             37        5.881     0.493
## 22  32.736 174.951 279.874            82             81        4.863     0.728
## 24  18.394 175.463 236.876            69             68        5.232     0.485
## 26   7.788 155.226 191.438            92             91        5.658     0.353
## 28   7.246 165.019  190.82            78             77        5.551     0.231
## 30    9.12 236.192 269.519            40             39        3.878     0.281
## 32   6.494 248.721 269.103            40             39        3.925     0.106
## 34   6.138 185.387 201.502            33             32        5.103     0.181
## 36   5.907 185.754 204.359            40             39        5.146     0.153
## 38  39.939  81.623 185.182            59             57        6.627     1.938
## 40  12.119 159.605 210.451            72             71         5.68     0.396
##    fraction.of.locally.unvoiced.frames
## 2                                8.333
## 4                                5.556
## 6                                4.000
## 8                                0.000
## 10                               0.000
## 12                               0.000
## 14                              18.519
## 16                              11.765
## 18                               9.434
## 20                               0.000
## 22                               0.000
## 24                               0.000
## 26                               0.000
## 28                               0.000
## 30                               5.556
## 32                               0.000
## 34                               0.000
## 36                               0.000
## 38                               0.000
## 40                               0.000
##                                           fraction number.of.voice.breaks
## 2   of locally unvoiced frames: 8.333%   (2 / 24)                      1
## 4   of locally unvoiced frames: 5.556%   (1 / 18)                      0
## 6   of locally unvoiced frames: 4.000%   (2 / 50)                      2
## 8        of locally unvoiced frames: 0   (0 / 36)                      0
## 10       of locally unvoiced frames: 0   (0 / 48)                      0
## 12       of locally unvoiced frames: 0   (0 / 50)                      0
## 14 of locally unvoiced frames: 18.519%   (5 / 27)                      0
## 16 of locally unvoiced frames: 11.765%   (2 / 17)                      0
## 18  of locally unvoiced frames: 9.434%   (5 / 53)                      1
## 20       of locally unvoiced frames: 0   (0 / 25)                      0
## 22       of locally unvoiced frames: 0   (0 / 47)                      0
## 24       of locally unvoiced frames: 0   (0 / 42)                      0
## 26       of locally unvoiced frames: 0   (0 / 61)                      0
## 28       of locally unvoiced frames: 0   (0 / 51)                      0
## 30  of locally unvoiced frames: 5.556%   (1 / 18)                      0
## 32       of locally unvoiced frames: 0   (0 / 18)                      0
## 34       of locally unvoiced frames: 0   (0 / 20)                      0
## 36       of locally unvoiced frames: 0   (0 / 24)                      0
## 38       of locally unvoiced frames: 0   (0 / 46)                      1
## 40       of locally unvoiced frames: 0   (0 / 48)                      0
##    degree.of.voice.breaks
## 2                  22.773
## 4                   0.000
## 6                   7.426
## 8                   0.000
## 10                  0.000
## 12                  0.000
## 14                  0.000
## 16                  0.000
## 18                  7.559
## 20                  0.000
## 22                  0.000
## 24                  0.000
## 26                  0.000
## 28                  0.000
## 30                  0.000
## 32                  0.000
## 34                  0.000
## 36                  0.000
## 38                  4.272
## 40                  0.000
##                                                               degree
## 2  of voice breaks: 22.773%   (0.052155 seconds / 0.229018 seconds)
## 4               of voice breaks: 0   (0 seconds / 0.176274 seconds)
## 6   of voice breaks: 7.426%   (0.032980 seconds / 0.444099 seconds)
## 8               of voice breaks: 0   (0 seconds / 0.330119 seconds)
## 10              of voice breaks: 0   (0 seconds / 0.430139 seconds)
## 12              of voice breaks: 0   (0 seconds / 0.449857 seconds)
## 14              of voice breaks: 0   (0 seconds / 0.251984 seconds)
## 16              of voice breaks: 0   (0 seconds / 0.173127 seconds)
## 18  of voice breaks: 7.559%   (0.035468 seconds / 0.469197 seconds)
## 20              of voice breaks: 0   (0 seconds / 0.234767 seconds)
## 22              of voice breaks: 0   (0 seconds / 0.419245 seconds)
## 24              of voice breaks: 0   (0 seconds / 0.381955 seconds)
## 26              of voice breaks: 0   (0 seconds / 0.536538 seconds)
## 28              of voice breaks: 0   (0 seconds / 0.450367 seconds)
## 30              of voice breaks: 0   (0 seconds / 0.175317 seconds)
## 32              of voice breaks: 0   (0 seconds / 0.182882 seconds)
## 34              of voice breaks: 0   (0 seconds / 0.194113 seconds)
## 36              of voice breaks: 0   (0 seconds / 0.229112 seconds)
## 38  of voice breaks: 4.272%   (0.017797 seconds / 0.416591 seconds)
## 40              of voice breaks: 0   (0 seconds / 0.425752 seconds)
##    jitter.local jitter.local.abs jitter.rap jitter.ppq5 shimmer.local
## 2         2.926          142.986      1.605       2.147         8.542
## 4         6.202          303.958      4.035       3.564         8.893
## 6         1.554           80.339      0.477       0.552         4.136
## 8         0.626           27.832      0.241       0.253         1.327
## 10        0.411           18.682      0.148       0.159         1.432
## 12        0.449           20.659      0.222       0.175         1.752
## 14        3.447          138.645      0.889       0.951         6.189
## 16        1.245           58.639      0.272       0.435         2.246
## 18        2.373          139.393       0.93        0.57         5.078
## 20        1.864          109.602      0.884       1.008         5.029
## 22        0.721           35.045      0.151       0.211         2.375
## 24         0.59           30.895      0.195       0.248         1.926
## 26         1.07           60.565      0.328       0.466         2.495
## 28        0.498           27.664      0.214       0.226         2.159
## 30         2.91           112.86      1.372       0.982         5.269
## 32        0.978           38.398      0.563       0.606         2.131
## 34        1.031           52.603      0.397       0.395         5.463
## 36        0.421           21.655      0.182       0.166         3.801
## 38        2.588           171.52      1.419       1.359        11.965
## 40        0.777           44.153      0.369       0.315         1.736
##    shimmer.local.db shimmer.apq3 shimmer.apq5 shimmer.apq11 mean.autocorr
## 2              1.57         3.89         3.46         3.798         0.873
## 4             0.955        4.002        2.859         5.278          0.91
## 6             0.622        1.215        1.536         2.945         0.895
## 8             0.117        0.543        0.626          1.38          0.98
## 10            0.133        0.458        0.619         0.692         0.985
## 12            0.163        0.866        0.784         0.979         0.987
## 14            1.027        2.381        3.367         6.287         0.883
## 16            0.196        0.795        1.169         1.605         0.931
## 18            0.576        0.956        1.288         3.289         0.945
## 20            0.652         1.61        1.973          2.39         0.969
## 22            0.209        0.585        0.966          1.41         0.981
## 24            0.175        0.551        0.642          1.09         0.989
## 26            0.309        0.509        0.567         1.251         0.978
## 28            0.214        0.461        0.605         1.299         0.989
## 30            0.471        2.117        1.963         2.308         0.928
## 32            0.196        0.824        1.047         1.702         0.989
## 34            0.877        2.668        3.658         3.035          0.94
## 36            0.344        2.123        1.871         1.923         0.991
## 38             1.73        6.748        5.364         8.823         0.938
## 40            0.183        0.654        0.665         1.184         0.988
##    mean.NHR mean.HNR      F1       F2       F3       F4 gender noise speaker
## 2     0.193   12.731 495.881  853.512 3303.512 3880.690      0    78     f-1
## 4     0.145   16.212 484.495  894.356 3197.222 3737.516      0    78     f-1
## 6     0.161   15.621 558.530 2325.318 2993.339 4157.486      0    78     f-1
## 8     0.021   20.469 623.327 2293.245 2967.519 4045.199      0    78     f-1
## 10    0.017   25.995 559.097 2350.072 3034.048 4073.220      0    78     f-1
## 12    0.018   27.001 485.833 2290.109 2928.457 4066.557      0    78     f-1
## 14    0.173   12.823 613.222 2324.702 2985.978 4142.649      0    78     f-1
## 16    0.093   17.098 600.641 2282.071 2943.010 4134.439      0    78     f-1
## 18    0.076   15.586 553.022 2233.089 2817.136 4024.624      0    78     f-1
## 20    0.035   17.656 548.293 2310.878 2847.136 4136.784      0    78     f-1
## 22    0.021    19.94 480.823 2361.993 2873.480 4108.998      0    78     f-1
## 24    0.011   21.446 490.250 2372.917 2982.956 3979.519      0    78     f-1
## 26    0.036   25.215 509.805 2312.655 2914.561 4068.288      0    78     f-1
## 28    0.012   23.782 517.387 2335.944 2849.333 4062.744      0    78     f-1
## 30     0.13   19.408 442.994  901.428 3194.200 3816.831      0    78     f-1
## 32    0.012   24.027 437.966  834.338 3208.140 3890.951      0    78     f-1
## 34    0.085   19.242 379.499  878.449 2928.371 3664.029      0    78     f-1
## 36     0.01   23.756 401.746  942.120 2915.999 3676.266      0    78     f-1
## 38    0.089   16.976 525.561 2288.670 2945.335 4081.089      0    78     f-1
## 40    0.012   21.481 517.043 2221.732 2923.273 3969.970      0    78     f-1
##    single tone phonation creaky
## 2       0   B2    creaky      1
## 4       0   B2     creaky      1
## 6       0   C2     creaky      1
## 8       0   C2     creaky      1
## 10      0   A1      modal      0
## 12      0   A1      modal      0
## 14      0   B2     creaky      1
## 16      0   B2     creaky      1
## 18      0   C1     creaky      1
## 20      0   C1     creaky      1
## 22      0   B1      modal      0
## 24      0   B1      modal      0
## 26      0   A2    breathy      0
## 28      0   A2    breathy      0
## 30      1   D1         NA      0
## 32      1   D1         NA      0
## 34      1   D2         NA      0
## 36      1   D2         NA      0
## 38      0   A2    breathy      0
## 40      0   A2    breathy      0
```

Checking

```
# How many values are of each category
length(voice_reports$tone[voice_reports$tone == "A1"])
```

```
## [1] 716
```

```
## [1] 574
length(voice_reports$tone[voice_reports$tone == "A2"])
```

```
## [1] 719
```

```
## [1] 575
length(voice_reports$tone[voice_reports$tone == "B1"])
```

```
## [1] 719
```

```
## [1] 719
length(voice_reports$tone[voice_reports$tone == "B2"])
```

```
## [1] 780
```

```
## [1] 768
length(voice_reports$tone[voice_reports$tone == "C1"])
```

```
## [1] 719
```

```
## [1] 575
length(voice_reports$tone[voice_reports$tone == "C2"])
```

```
## [1] 719
```

```
## [1] 575

length(voice_reports$tone[voice_reports$tone == "D1"])
```

```
## [1] 180
## [1] 575
length(voice_reports$tone[voice_reports$tone == "D2"])
```

```
## [1] 180
```

```
## [1] 575
length(voice_reports$tone[voice_reports$tone == "NA"])
```

```
## [1] 0
```

```
## [1] 0
```

Convert categorical values to factors

```
## Not sure if this is necessary for variables already binarily coded.
voice_reports$gender <- as.factor(voice_reports$gender)
voice_reports$noise <- as.factor(voice_reports$noise)
voice_reports$tone <- as.factor(voice_reports$tone)
voice_reports$single <- as.factor(voice_reports$single)
voice_reports$phonation <- as.factor(voice_reports$phonation)
voice_reports$creaky <- as.factor(voice_reports$creaky)
voice_reports$speaker <- as.factor(voice_reports$speaker)
```

Summary of current data

```
summary(voice_reports)
```

```
##   sound.name        total.duration     intensity      spectraltilt
##  Length:4732        Min.   :0.0340   Min.   :40.23   Min.   :-46.128
##  Class :character   1st Qu.:0.2110   1st Qu.:56.61   1st Qu.:-26.751
##  Mode  :character   Median :0.3000   Median :63.24   Median :-17.625
##                     Mean   :0.3173   Mean   :61.96   Mean   :-19.739
##                     3rd Qu.:0.4170   3rd Qu.:68.01   3rd Qu.:-13.302
##                     Max.   :0.7850   Max.   :80.76   Max.   :  2.476
##
##   median.F0           mean.F0             sd.F0               min.F0
##  Length:4732        Length:4732        Length:4732        Length:4732
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     max.F0          number.pulses    number.periods   mean.periods
##  Length:4732        Min.   :  1.00   Min.   :  0.00   Length:4732
##  Class :character   1st Qu.: 26.00   1st Qu.: 25.00   Class :character
##  Mode  :character   Median : 41.00   Median : 40.00   Mode  :character
##                     Mean   : 47.74   Mean   : 46.51
##                     3rd Qu.: 66.00   3rd Qu.: 65.00
##                     Max.   :152.00   Max.   :151.00
##
##   sd.period         fraction.of.locally.unvoiced.frames   fraction
##  Length:4732        Min.   : 0.000                      Length:4732
##  Class :character   1st Qu.: 0.000                      Class :character
##  Mode  :character   Median : 0.000                      Mode  :character
##                     Mean   : 4.686
##                     3rd Qu.: 4.000
##                     Max.   :96.875
##
##  number.of.voice.breaks degree.of.voice.breaks    degree
##  Min.   :0.000          Min.   : 0.000         Length:4732
##  1st Qu.:0.000          1st Qu.: 0.000         Class :character
##  Median :0.000          Median : 0.000         Mode  :character
##  Mean   :0.142          Mean   : 2.148
##  3rd Qu.:0.000          3rd Qu.: 0.000
##  Max.   :3.000          Max.   :56.526
##
##  jitter.local       jitter.local.abs    jitter.rap        jitter.ppq5
##  Length:4732        Length:4732        Length:4732        Length:4732
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  shimmer.local      shimmer.local.db   shimmer.apq3       shimmer.apq5
##  Length:4732        Length:4732        Length:4732        Length:4732
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  shimmer.apq11      mean.autocorr        mean.NHR           mean.HNR
##  Length:4732        Length:4732        Length:4732        Length:4732
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##        F1               F2               F3             F4       gender
##  Min.   : 201.1   Min.   : 462.1   Min.   :1767   Min.   :2688   0:1893
##  1st Qu.: 383.2   1st Qu.: 882.0   1st Qu.:2557   1st Qu.:3452   1:2839
##  Median : 491.9   Median :1599.7   Median :2711   Median :3713
##  Mean   : 574.0   Mean   :1515.6   Mean   :2737   Mean   :3708
##  3rd Qu.: 803.7   3rd Qu.:1980.4   3rd Qu.:2907   3rd Qu.:3970
##  Max.   :1155.4   Max.   :2656.6   Max.   :3518   Max.   :4681
##
##  noise     speaker   single        tone       phonation    creaky
##  0 :1575   f-1:947   0:2366   B2     :780   breathy: 719   0:2514
##  78:1578   f-2:946   1:2366   A2     :719   creaky :2218   1:2218
##  90:1579   m-1:945            B1     :719   modal  :1435
##            m-2:947            C1     :719   NA     : 360
##            m-3:947            C2     :719
##                               A1     :716
##                               (Other):360
```

Clean up undefined values to prepare for Classification

```
## Method 1: Simply drop values that are undefined in jitter and shimmer variables
voice_reports_clean <- voice_reports[!(voice_reports$jitter.local==" --undefined-- " | voice_reports$shi

# Convert two variables to numeric
voice_reports_clean$jitter.local <- as.numeric(voice_reports_clean$jitter.local)
voice_reports_clean$shimmer.local <- as.numeric(voice_reports_clean$shimmer.local)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$median.F0 <- as.numeric(voice_reports_clean$median.F0)
voice_reports_clean$mean.F0 <- as.numeric(voice_reports_clean$mean.F0)
voice_reports_clean$sd.F0 <- as.numeric(voice_reports_clean$sd.F0)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$min.F0 <- as.numeric(voice_reports_clean$min.F0)
voice_reports_clean$max.F0 <- as.numeric(voice_reports_clean$max.F0)
voice_reports_clean$number.pulses<- as.numeric(voice_reports_clean$number.pulses)
voice_reports_clean$number.periods <- as.numeric(voice_reports_clean$number.periods)
voice_reports_clean$mean.periods <- as.numeric(voice_reports_clean$mean.periods)
#voice_reports_clean$sd.periods <- as.numeric(voice_reports_clean$sd.periods)
voice_reports_clean$jitter.local.abs <- as.numeric(voice_reports_clean$jitter.local.abs)
voice_reports_clean$jitter.rap <- as.numeric(voice_reports_clean$jitter.rap)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$jitter.ppq5 <- as.numeric(voice_reports_clean$jitter.ppq5)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$shimmer.local.db <- as.numeric(voice_reports_clean$shimmer.local.db)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$shimmer.apq3 <- as.numeric(voice_reports_clean$shimmer.apq3)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$shimmer.apq5 <- as.numeric(voice_reports_clean$shimmer.apq5)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$shimmer.apq11 <- as.numeric(voice_reports_clean$shimmer.apq11)
```

```
## Warning: NAs introduced by coercion
```

```
voice_reports_clean$mean.autocorr <- as.numeric(voice_reports_clean$mean.autocorr)
voice_reports_clean$mean.NHR <- as.numeric(voice_reports_clean$mean.NHR)
voice_reports_clean$mean.HNR <- as.numeric(voice_reports_clean$mean.HNR)
voice_reports_clean$tone <- as.factor(voice_reports_clean$tone)
summary(voice_reports_clean)
```

```
##   sound.name        total.duration     intensity      spectraltilt
##  Length:4729        Min.   :0.0340   Min.   :40.23   Min.   :-46.128
##  Class :character   1st Qu.:0.2110   1st Qu.:56.60   1st Qu.:-26.748
##  Mode  :character   Median :0.3010   Median :63.25   Median :-17.626
##                     Mean   :0.3174   Mean   :61.96   Mean   :-19.738
##                     3rd Qu.:0.4170   3rd Qu.:68.01   3rd Qu.:-13.302
##                     Max.   :0.7850   Max.   :80.76   Max.   :  2.476
##
##    median.F0        mean.F0          sd.F0            min.F0
##  Min.   : 75.86   Min.   : 77.42   Min.   :  0.244   Min.   : 63.42
##  1st Qu.:127.09   1st Qu.:129.66   1st Qu.:  5.770   1st Qu.:101.03
##  Median :159.35   Median :163.61   Median : 12.564   Median :130.39
##  Mean   :164.91   Mean   :169.06   Mean   : 23.939   Mean   :138.15
##  3rd Qu.:191.31   3rd Qu.:195.79   3rd Qu.: 26.153   3rd Qu.:168.47
##  Max.   :571.98   Max.   :506.61   Max.   :222.776   Max.   :489.12
##                                    NA's   :2
##      max.F0       number.pulses    number.periods    mean.periods
##  Min.   : 80.87   Min.   :  3.00   Min.   :  2.00   Min.   : 2.036
##  1st Qu.:149.56   1st Qu.: 26.00   1st Qu.: 25.00   1st Qu.: 5.121
##  Median :188.80   Median : 41.00   Median : 40.00   Median : 6.117
##  Mean   :209.14   Mean   : 47.77   Mean   : 46.54   Mean   : 6.458
##  3rd Qu.:234.05   3rd Qu.: 66.00   3rd Qu.: 65.00   3rd Qu.: 7.734
##  Max.   :643.05   Max.   :152.00   Max.   :151.00   Max.   :12.909
##
##   sd.period         fraction.of.locally.unvoiced.frames   fraction
##  Length:4729        Min.   : 0.000                      Length:4729
##  Class :character   1st Qu.: 0.000                      Class :character
##  Mode  :character   Median : 0.000                      Mode  :character
##                     Mean   : 4.647
##                     3rd Qu.: 4.000
##                     Max.   :88.372
##
##  number.of.voice.breaks degree.of.voice.breaks    degree
##  Min.   :0.0000         Min.   : 0.00          Length:4729
##  1st Qu.:0.0000         1st Qu.: 0.00          Class :character
##  Median :0.0000         Median : 0.00          Mode  :character
##  Mean   :0.1421         Mean   : 2.15
##  3rd Qu.:0.0000         3rd Qu.: 0.00
##  Max.   :3.0000         Max.   :56.53
##
##   jitter.local    jitter.local.abs     jitter.rap       jitter.ppq5
##  Min.   : 0.130   Min.   :   6.221   Min.   : 0.0500   Min.   : 0.0720
##  1st Qu.: 0.757   1st Qu.:  44.246   1st Qu.: 0.2150   1st Qu.: 0.2740
##  Median : 1.353   Median :  88.445   Median : 0.3680   Median : 0.4460
##  Mean   : 2.030   Mean   : 133.265   Mean   : 0.7642   Mean   : 0.8815
##  3rd Qu.: 2.608   3rd Qu.: 171.955   3rd Qu.: 0.8233   3rd Qu.: 0.9030
##  Max.   :23.746   Max.   :1855.965   Max.   :14.3880   Max.   :28.6280
##                                      NA's   :5         NA's   :18
##  shimmer.local    shimmer.local.db  shimmer.apq3     shimmer.apq5
##  Min.   : 0.890   Min.   :0.0780   Min.   : 0.065   Min.   : 0.386
##  1st Qu.: 3.478   1st Qu.:0.3558   1st Qu.: 1.057   1st Qu.: 1.453
##  Median : 5.184   Median :0.5340   Median : 1.701   Median : 2.348
##  Mean   : 6.640   Mean   :0.7584   Mean   : 2.510   Mean   : 3.331
##  3rd Qu.: 8.120   3rd Qu.:0.8710   3rd Qu.: 2.988   3rd Qu.: 4.033
##  Max.   :74.441   Max.   :7.0640   Max.   :49.231   Max.   :55.763
##  NA's   :5        NA's   :5        NA's   :9        NA's   :25
##  shimmer.apq11    mean.autocorr      mean.NHR         mean.HNR
##  Min.   : 0.241   Min.   :0.4930   Min.   :0.0020   Min.   :-0.126
##  1st Qu.: 2.337   1st Qu.:0.8930   1st Qu.:0.0190   1st Qu.:12.259
##  Median : 3.767   Median :0.9560   Median :0.0520   Median :16.311
##  Mean   : 5.400   Mean   :0.9275   Mean   :0.1051   Mean   :16.235
##  3rd Qu.: 6.457   3rd Qu.:0.9820   3rd Qu.:0.1490   3rd Qu.:20.297
##  Max.   :61.317   Max.   :0.9980   Max.   :1.1340   Max.   :31.943
##  NA's   :238
##        F1               F2               F3              F4        gender
##  Min.   : 201.1   Min.   : 462.1   Min.   :1767   Min.   :2688   0:1893
##  1st Qu.: 383.2   1st Qu.: 882.0   1st Qu.:2557   1st Qu.:3452   1:2836
##  Median : 491.9   Median :1599.9   Median :2711   Median :3713
##  Mean   : 573.9   Mean   :1515.8   Mean   :2737   Mean   :3708
##  3rd Qu.: 803.7   3rd Qu.:1980.4   3rd Qu.:2907   3rd Qu.:3970
##  Max.   :1155.4   Max.   :2656.6   Max.   :3518   Max.   :4681
##
##  noise     speaker   single        tone       phonation    creaky
##  0 :1573   f-1:947   0:2364   B2     :779   breathy: 718   0:2512
##  78:1577   f-2:946   1:2365   C1     :719   creaky :2217   1:2217
##  90:1579   m-1:944            C2     :719   modal  :1434
##            m-2:947            A2     :718   NA     : 360
##            m-3:945            B1     :718
##                               A1     :716
##                               (Other):360
```

```
dim(voice_reports_clean)
```

```
## [1] 4729   41
```

Logistic Regression on Gender

```
logit_gender = glm(gender ~ mean.F0 + total.duration + intensity +  mean.HNR, family = "binomial", data
summary(logit_gender)
```

```
##
## Call:
## glm(formula = gender ~ mean.F0 + total.duration + intensity +
##     mean.HNR, family = "binomial", data = voice_reports_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.8737  -0.1990   0.0663   0.3090   4.7545
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -12.460987   0.623470 -19.987   <2e-16 ***
## mean.F0         -0.038406   0.001306 -29.400   <2e-16 ***
## total.duration  -4.233214   0.468622  -9.033   <2e-16 ***
## intensity        0.389374   0.013315  29.244   <2e-16 ***
## mean.HNR        -0.174024   0.011453 -15.194   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6366.5  on 4728  degrees of freedom
## Residual deviance: 2226.3  on 4724  degrees of freedom
## AIC: 2236.3
##
```

```
## Number of Fisher Scoring iterations: 7
```

Logistic Regression on Creaky

```
logit_creaky = glm(creaky ~  mean.F0 + total.duration + intensity + spectraltilt + number.pulses + mean
```

```
## Warning: glm.fit: algorithm did not converge
```

```
summary(logit_creaky)
```

```
##
## Call:
## glm(formula = creaky ~ mean.F0 + total.duration + intensity +
##     spectraltilt + number.pulses + mean.HNR + jitter.local +
##     shimmer.local, family = "binomial", data = voice_reports_clean)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.6154  -0.5712  -0.1713   0.5725   2.4249
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.259639   0.591364  -3.821 0.000133 ***
## mean.F0         0.023476   0.001967  11.933  < 2e-16 ***
## total.duration  2.959517   0.936577   3.160 0.001578 **
## intensity       0.008285   0.005075   1.633 0.102565
## spectraltilt   -0.069450   0.006075 -11.432  < 2e-16 ***
## number.pulses  -0.012641   0.005496  -2.300 0.021457 *
## mean.HNR       -0.333977   0.016911 -19.749  < 2e-16 ***
## jitter.local    0.493546   0.054695   9.024  < 2e-16 ***
## shimmer.local   0.086665   0.019847   4.367 1.26e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6530.3  on 4723  degrees of freedom
## Residual deviance: 3660.2  on 4715  degrees of freedom
##   (5 observations deleted due to missingness)
## AIC: 3678.2
##
## Number of Fisher Scoring iterations: 25
```

Regression with Interaction

```
lm_creaky = lm(jitter.local ~ mean.HNR + noise + mean.HNR*noise, data = voice_reports_clean)
summary(lm_creaky)
```

```
##
## Call:
## lm(formula = jitter.local ~ mean.HNR + noise + mean.HNR * noise,
##     data = voice_reports_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.4035 -0.8150 -0.1562  0.4947 16.3821
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        7.402153   0.104585   70.78   <2e-16 ***
## mean.HNR          -0.305684   0.006352  -48.13   <2e-16 ***
## noise78           -2.527307   0.155060  -16.30   <2e-16 ***
## noise90           -2.981343   0.156652  -19.03   <2e-16 ***
## mean.HNR:noise78   0.117111   0.009220   12.70   <2e-16 ***
## mean.HNR:noise90   0.139891   0.009102   15.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.437 on 4723 degrees of freedom
## Multiple R-squared:  0.4742, Adjusted R-squared:  0.4736
## F-statistic: 851.7 on 5 and 4723 DF,  p-value: < 2.2e-16
```

Multinomial Regression to predict the Noise Level.

```
# Use the multinom function from the nnet package (Ref: https://stats.idre.ucla.edu/r/dae/multinomial-l

library("nnet")
# Use the 78 noise level as the reference level
voice_reports_clean$noise2 <- relevel(voice_reports_clean$noise, ref = "78")
multinom_noise <- multinom(noise2 ~ mean.F0 + total.duration + intensity + spectraltilt, data=voice_repo
```

```
## # weights:  18 (10 variable)
## initial  value 5195.337513
## iter  10 value 3975.790276
## iter  20 value 3796.196458
## iter  20 value 3796.196457
## iter  20 value 3796.196457
## final  value 3796.196457
## converged
```

```
summary(multinom_noise)
```

```
## Call:
## multinom(formula = noise2 ~ mean.F0 + total.duration + intensity +
##     spectraltilt, data = voice_reports_clean)
##
## Coefficients:
##    (Intercept)      mean.F0 total.duration  intensity spectraltilt
## 0    11.096456 -0.017515359      -5.053825 -0.1372098 -0.066048629
## 90   -9.666187  0.005532275       1.529608  0.1259355  0.003031549
##
## Std. Errors:
##    (Intercept)      mean.F0 total.duration   intensity spectraltilt
## 0    0.4754404 0.0010976356      0.3722000 0.006129938  0.005180976
## 90   0.4945335 0.0007262764      0.2924905 0.006466712  0.004398994
##
## Residual Deviance: 7592.393
## AIC: 7612.393
```

```
# The result in general supports our predictions regarding the relationship
#between relative noise levels
# and F0, duration, intensity, etc.

# For instance,
# A one-unit increase in mean F0 is associated with the decrease in the
#log odds of quiet vs. 78 noise level in the amount of 0.0133
# A one-unit increase in mean F0 is associated with the increase in the
#log odds of 90 noise vs. 78 noise in the amount of 0.006

# A one-unit increase in duration is associated with the decrease in the
#log odds of quiet vs. 78 noise level in the amount of 5.795
# A one-unit increase in duration is associated with the increase in the
#log odds of 90 noise vs. 78 noise in the amount of 2.80

# A one-unit increase in intensity is associated with the decrease in the
#log odds of quiet vs. 78 noise level in the amount of 0.35
# A one-unit increase in intensity is associated with the increase in the
#*log odds of 90 noise vs. 78 noise in the amount of 0.24

## Giang to double check this result
# A one-unit increase in spectraltilt is associated with the decrease in the
#log odds of quiet vs. 78 noise level in the amount of 0.066
# A one-unit increase in spectraltilt is associated with the increase in the
#log odds of 90 noise vs. 78 noise in the amount of 0.011
```

Classification using SMV (ref https://medium.com/@ODSC/build-a-multi-class-support-vector-machine-in-r-abcdd4b7dab6)

```
library(e1071)
set.seed(777)
n <- nrow(voice_reports_clean)
ntrain <- round(n*0.75)  # 75% for training set
tindex <- sample(n, ntrain)
# Do not include noise predictor in this model yet.
train <- voice_reports_clean[tindex,c("total.duration", "intensity",
                                      "spectraltilt", "mean.F0", "jitter.local",
                                      "shimmer.local", "mean.HNR", "gender",
                                      "tone")]
test <- voice_reports_clean[-tindex,c("total.duration", "intensity",
                                      "spectraltilt", "mean.F0", "jitter.local",
                                      "shimmer.local", "mean.HNR", "gender",
                                        "tone")]

# Some factors cause any error probably due to not having the same levels between train and test?
svm_model <- svm(tone ~ total.duration + intensity + spectraltilt + mean.F0 + jitter.local
                 + shimmer.local + mean.HNR + gender, data=train,
          method="C-classification", kernal="radial",
          gamma=0.1, cost=10)

summary(svm_model)
```

```
##
```

```
## Call:
## svm(formula = tone ~ total.duration + intensity + spectraltilt +
##     mean.F0 + jitter.local + shimmer.local + mean.HNR + gender, data = train,
##     method = "C-classification", kernal = "radial", gamma = 0.1,
##     cost = 10)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  2389
##
##  ( 336 364 283 373 428 405 85 115 )
##
##
## Number of Classes:  8
##
## Levels:
##  A1 A2 B1 B2 C1 C2 D1 D2
```

```
prediction <- predict(svm_model, test)
confusion <- table(test$tone, prediction)
confusion
```

```
##     prediction
##        A1  A2  B1  B2  C1  C2  D1  D2
##   A1  130  16  11   6   2   1   0   0
##   A2    9 132  19   2   8   0   0   2
##   B1    9  24 124   1  11  13   0   0
##   B2    1   3   3 148  24  25   3   7
##   C1    1  11   3  11 142   6   0   3
##   C2    6   1   5  30  10 130   2   2
##   D1    2   0   0   1   0   2  31   2
##   D2    0   5   7   1   6   0   0  28
```

```
# Accuracy
sum(diag(confusion))/sum(confusion)
```

```
## [1] 0.7318105
```

Classification using k-means clustering

```
install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/3_mgdyf12z7dvb8rt4d60nt80000gn/T//Rtmp3eURqD/downloaded_packages
```

```
library("caret")
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
set.seed(777)
n <- nrow(voice_reports_clean)
ntrain <- round(n*0.75)  # 75% for training set
tindex <- sample(n, ntrain)
# Do not include noise predictor in this model yet.
train <- voice_reports_clean[tindex,c("total.duration", "intensity",
                                      "spectraltilt", "mean.F0", "jitter.local",
                                      "shimmer.local", "mean.HNR", "gender", "tone")]
test <- voice_reports_clean[-tindex,c("total.duration", "intensity",
                                      "spectraltilt", "mean.F0", "jitter.local",
                                          "shimmer.local", "mean.HNR", "gender", "tone")]
summary(train)
```

```
##  total.duration     intensity      spectraltilt        mean.F0
##  Min.   :0.0340   Min.   :40.23   Min.   :-46.128   Min.   : 77.42
##  1st Qu.:0.2110   1st Qu.:56.74   1st Qu.:-26.811   1st Qu.:129.58
##  Median :0.3000   Median :63.20   Median :-17.577   Median :163.94
##  Mean   :0.3174   Mean   :62.00   Mean   :-19.716   Mean   :168.95
##  3rd Qu.:0.4170   3rd Qu.:68.02   3rd Qu.:-13.286   3rd Qu.:195.47
##  Max.   :0.7850   Max.   :80.76   Max.   :  2.476   Max.   :506.61
##
##   jitter.local    shimmer.local       mean.HNR      gender        tone
##  Min.   : 0.130   Min.   : 0.890   Min.   :-0.015   0:1429   B2     :565
##  1st Qu.: 0.754   1st Qu.: 3.453   1st Qu.:12.287   1:2118   A1     :550
##  Median : 1.337   Median : 5.091   Median :16.428            A2     :546
##  Mean   : 2.007   Mean   : 6.553   Mean   :16.300            C1     :542
##  3rd Qu.: 2.561   3rd Qu.: 8.014   3rd Qu.:20.405            B1     :536
##  Max.   :21.610   Max.   :67.317   Max.   :31.943            C2     :533
##                   NA's   :5                                  (Other):275
```

```
# Use repeated 5-fold cross-validation, with 3 repeats.
install.packages("kknn", repos = "http://cran.us.r-project.org")
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/3_mgdyf12z7dvb8rt4d60nt80000gn/T//Rtmp3eURqD/downloaded_packages
```

```
library("kknn")
```

```
##
## Attaching package: 'kknn'

## The following object is masked from 'package:caret':
##
##     contr.dummy
```

```
# # I set the parameters of trainControl to be method ~ repeatedcv, 5 folds, # and 2 repeats.
# control <- trainControl(method = "repeatedcv", number = 5, repeats=2)
# # Here I train the model using knn, and k values from 1 to 10.
# knn.cvfit <- train(tone ~ ., method = "knn", data = train,
# tuneGrid = data.frame(k = seq(1, 6, 1)), trControl = control)
#
# plot(knn.cvfit$results$k, 1-knn.cvfit$results$Accuracy,
# xlab = "K", ylab = "Classification Error", type = "b", pch = 19, col = "darkorange")
```

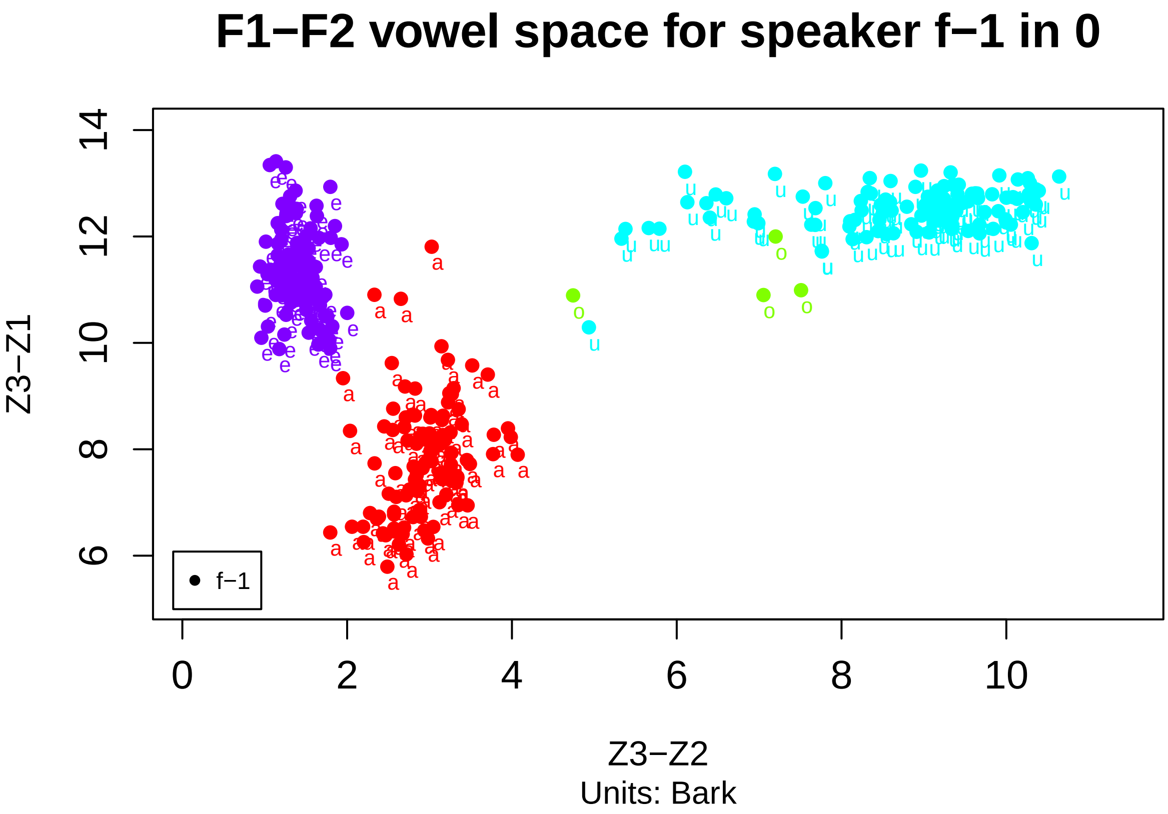
Vowels plotting

```
#http://lingtools.uoregon.edu/norm/about_norm1.php

install.packages("vowels", repos='http://cran.us.r-project.org')
```

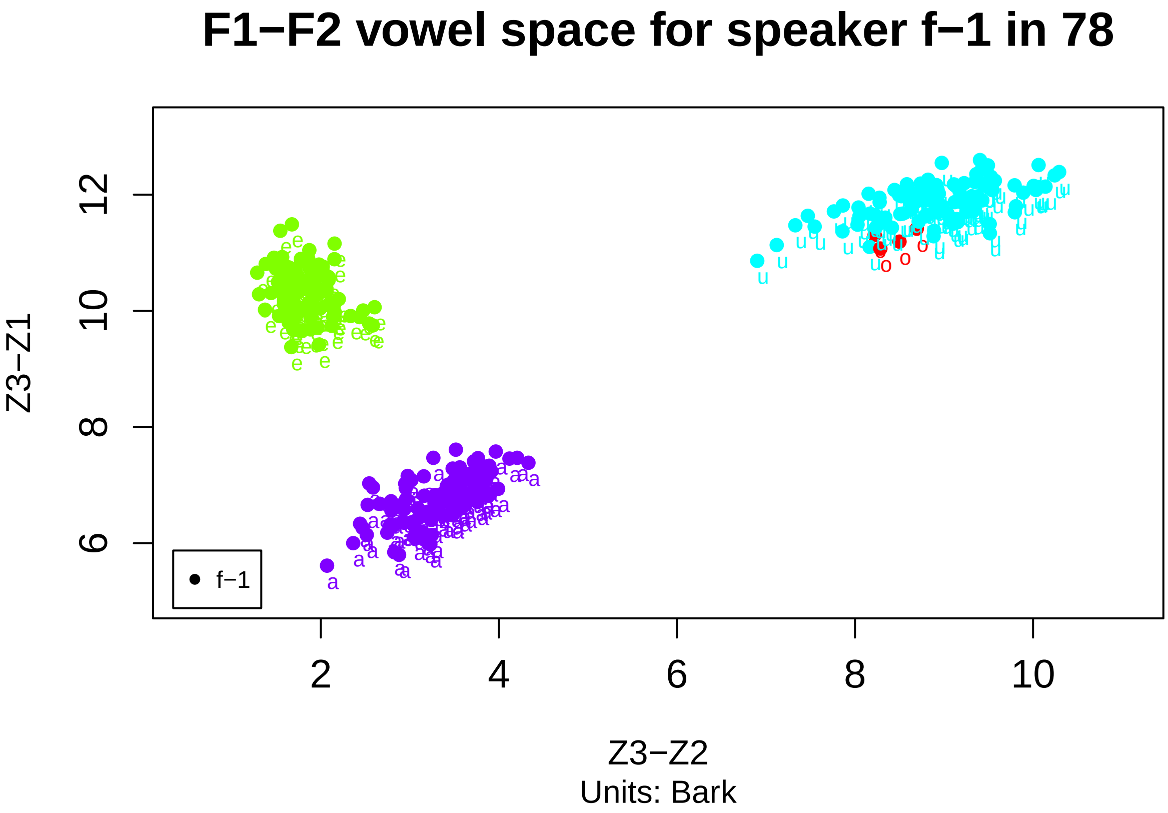
```
##
## The downloaded binary packages are in
##  /var/folders/9c/3_mgdyf12z7dvb8rt4d60nt80000gn/T//Rtmp3eURqD/downloaded_packages
```

```
library(vowels)
```

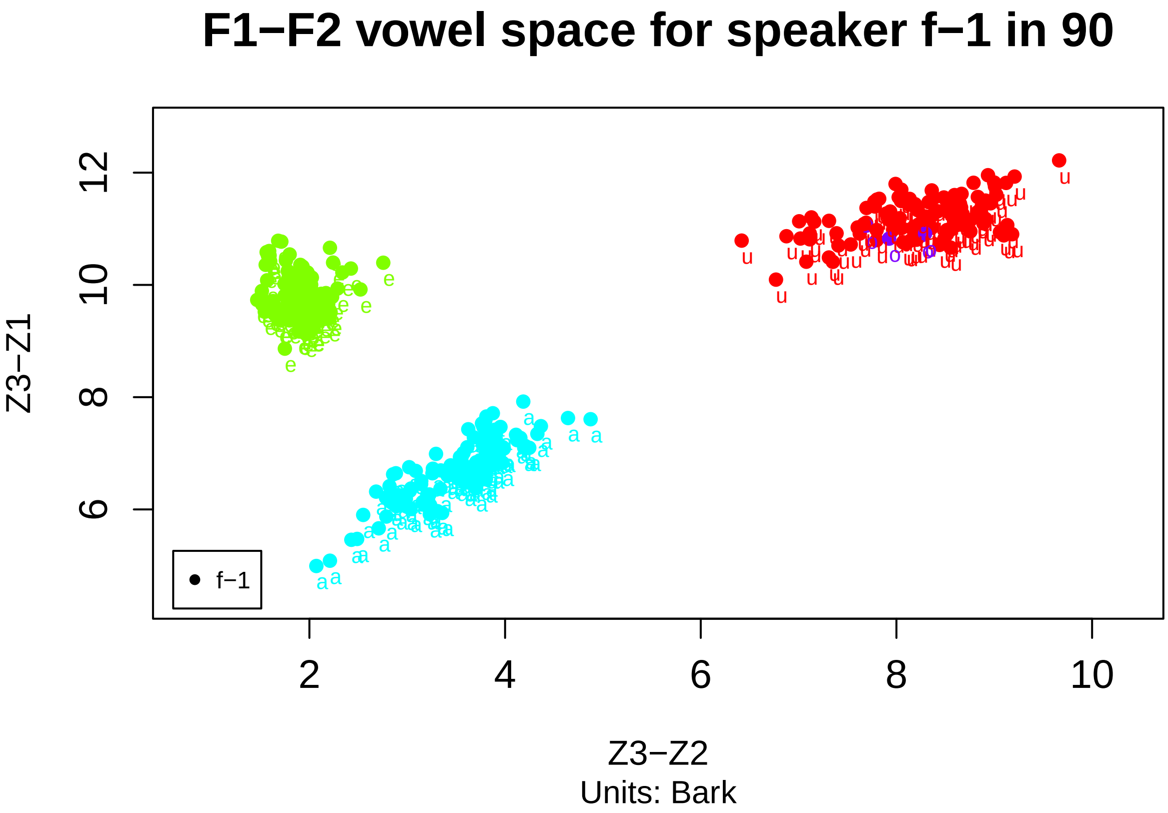
```
# Prepare vowels data
vowels <- voice_reports_clean[, c(37, 1, 36, 31, 32, 33)]
vowels$gl.F1 <- NA
vowels$gl.F2 <- NA
vowels$gl.F3 <- NA
# Extracting a substring that contains only the syllable names.
vowels$sound.name <- sapply(strsplit(vowels[,2], split="_", fixed=TRUE), "[", 2)

# Add vowel annotation.
vowels$vowel <- ifelse(grepl("a", vowels$sound.name, ignore.case=T), "a",
ifelse(grepl("à", vowels$sound.name, ignore.case=T), "a",
ifelse(grepl("á", vowels$sound.name, ignore.case=T), "a",
ifelse(grepl("ả", vowels$sound.name, ignore.case=T), "a",
ifelse(grepl("ã", vowels$sound.name, ignore.case=T), "a",
ifelse(grepl("ạ", vowels$sound.name, ignore.case=T), "a",
ifelse(grepl("ê", vowels$sound.name, ignore.case=T), "e",
ifelse(grepl("ề", vowels$sound.name, ignore.case=T), "e",
ifelse(grepl("ế", vowels$sound.name, ignore.case=T), "e",
ifelse(grepl("ể", vowels$sound.name, ignore.case=T), "e",
ifelse(grepl("ễ", vowels$sound.name, ignore.case=T), "e",
ifelse(grepl("ệ", vowels$sound.name, ignore.case=T), "e",
ifelse(grepl("u", vowels$sound.name, ignore.case=T), "u",
ifelse(grepl("ù", vowels$sound.name, ignore.case=T), "u",
ifelse(grepl("ú", vowels$sound.name, ignore.case=T), "u",
ifelse(grepl("ủ", vowels$sound.name, ignore.case=T), "u",
ifelse(grepl("ũ", vowels$sound.name, ignore.case=T), "u",
ifelse(grepl("ụ", vowels$sound.name, ignore.case=T), "u",
ifelse(grepl("ộ", vowels$sound.name, ignore.case=T), "o","NA"))))))))))))))))))

# Convert vowel types to a factor variable
vowels$vowel <- as.factor(vowels$vowel)
vowels$noise <- as.factor(vowels$noise)
vowels <- vowels[,c("speaker", "vowel", "noise", "F1", "F2", "F3", "gl.F1", "gl.F2", "gl.F3")]

# plot only sub-dataframes
vowels_plotting <- function(datamat, noise, speaker) {
  if (speaker == "all") {
    vowels <- datamat[datamat$noise==noise,]
    vowelplot(norm.bark(vowels), title=paste("F1-F2 vowel space for", speaker, "in", noise), color="vowe
  } else {
    vowels <- datamat[datamat$noise==noise & datamat$speaker==speaker,]
  vowelplot(norm.bark(vowels), title=paste("F1-F2 vowel space for speaker", speaker, "in", noise), colo
  }
}

vowels_plotting(vowels, 0, "f-1")
```



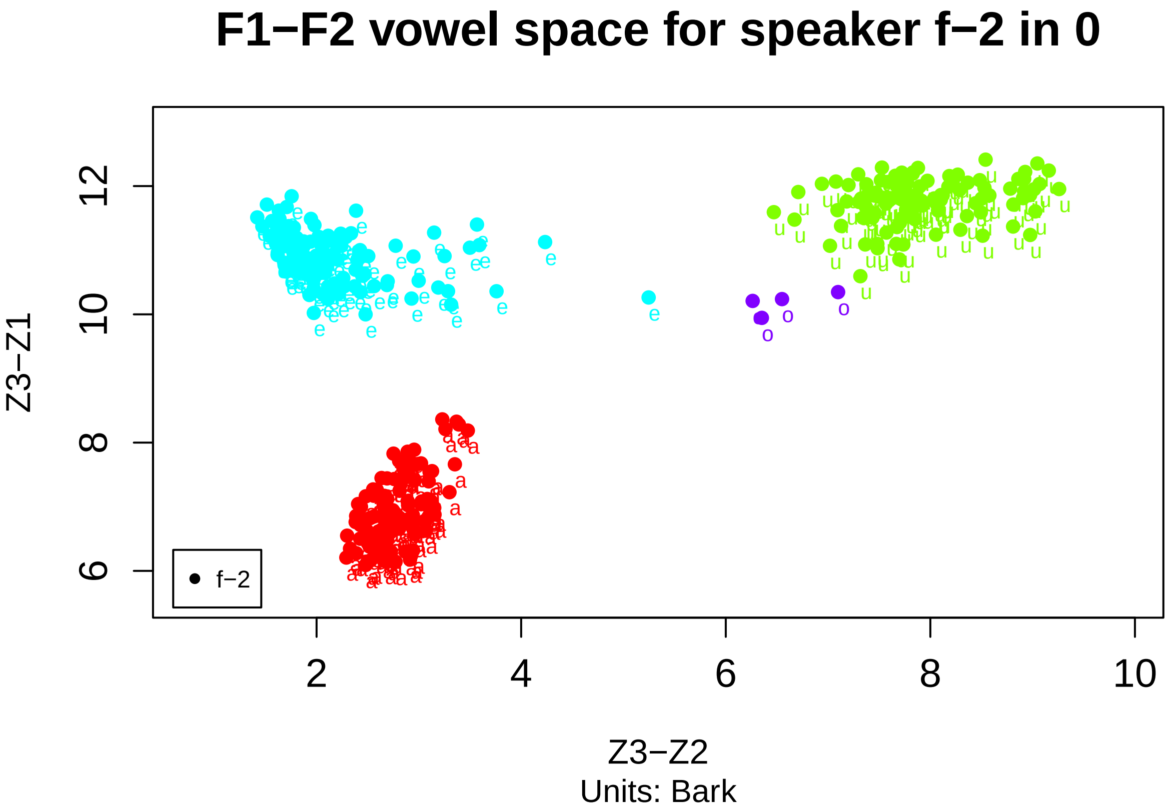
```
vowels_plotting(vowels, 78, "f-1")
```



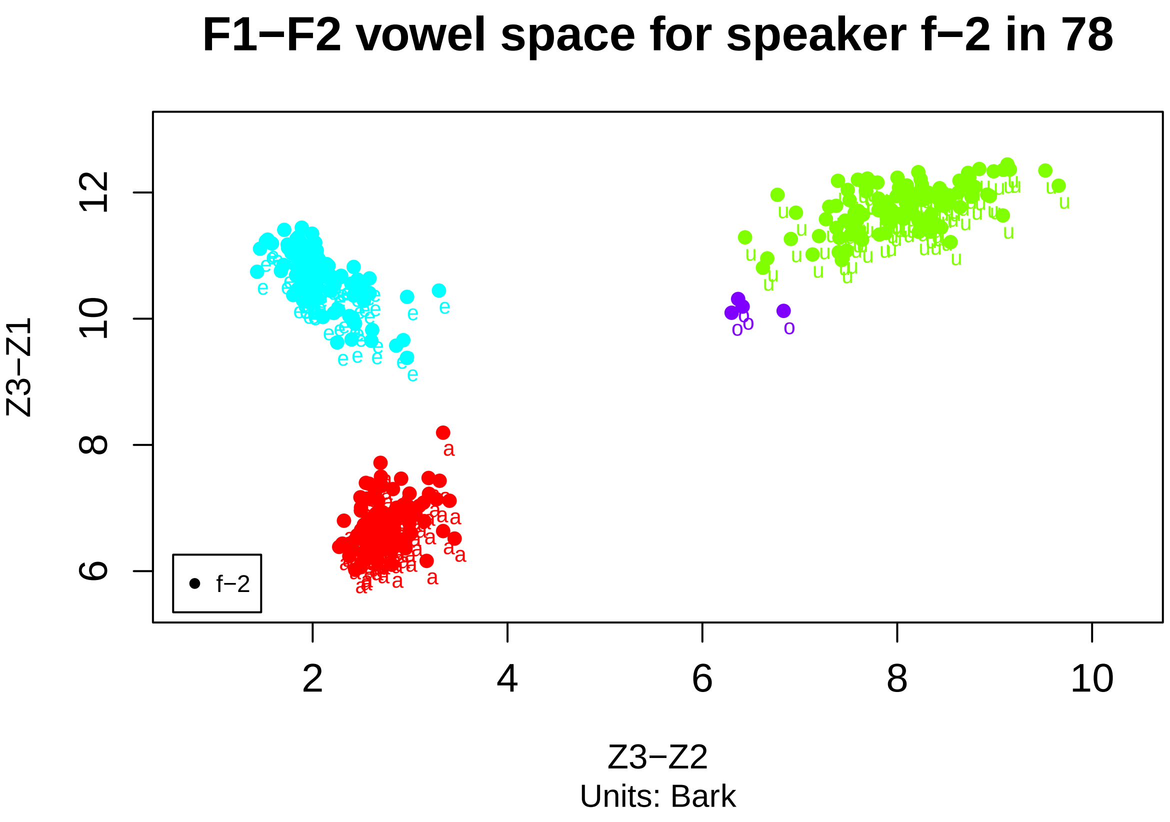
```
vowels_plotting(vowels, 90, "f-1")
```



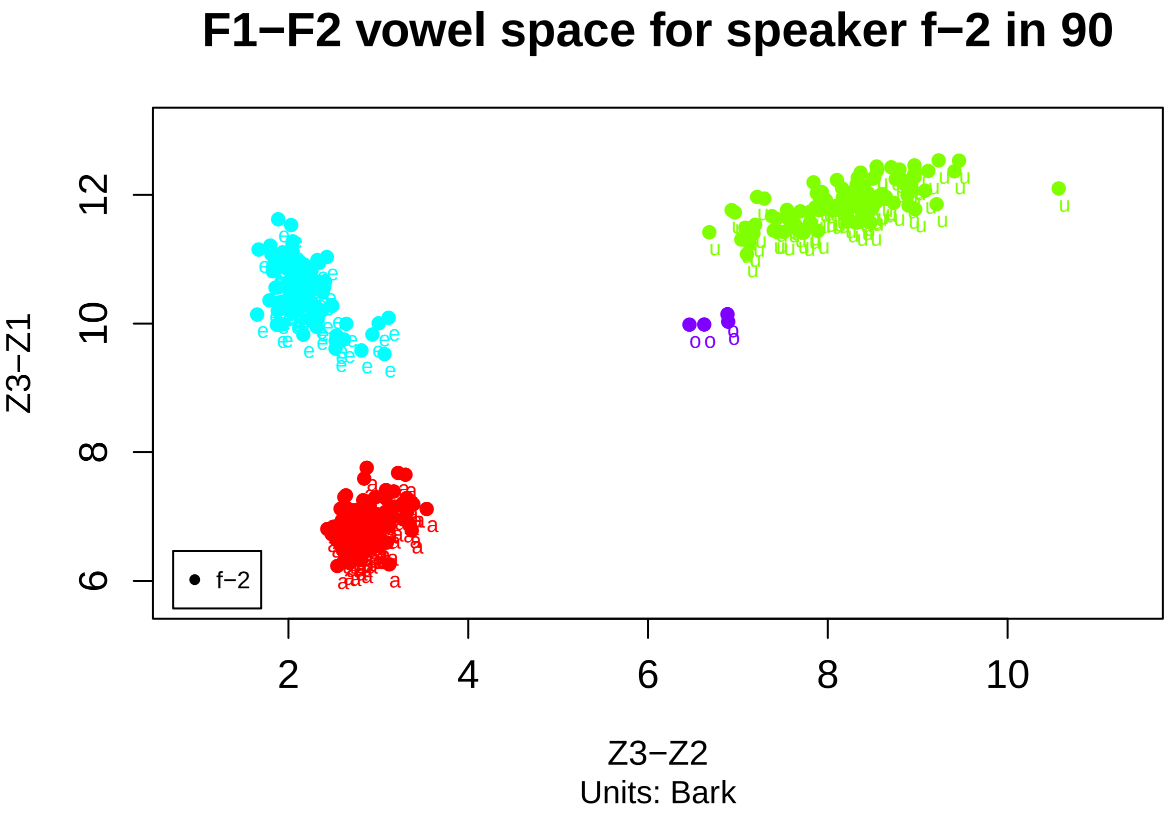
```
vowels_plotting(vowels, 0, "f-2")
```



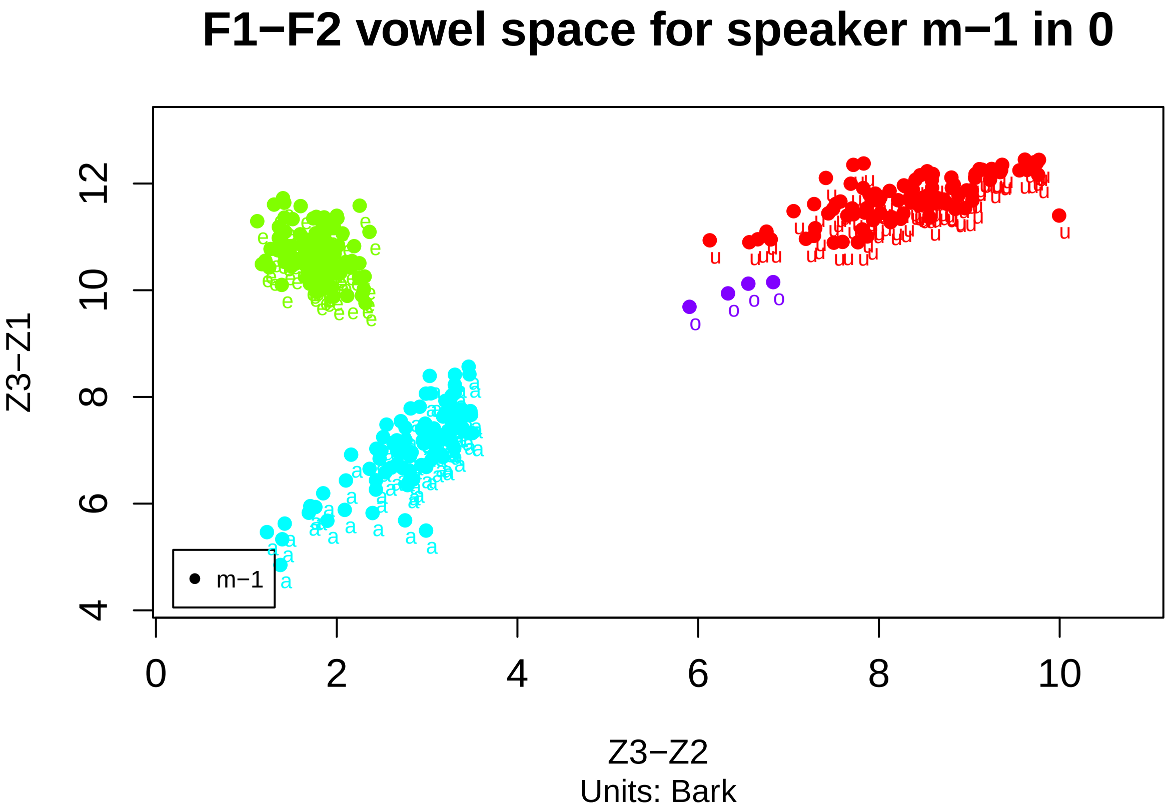
```
vowels_plotting(vowels, 78, "f-2")
```



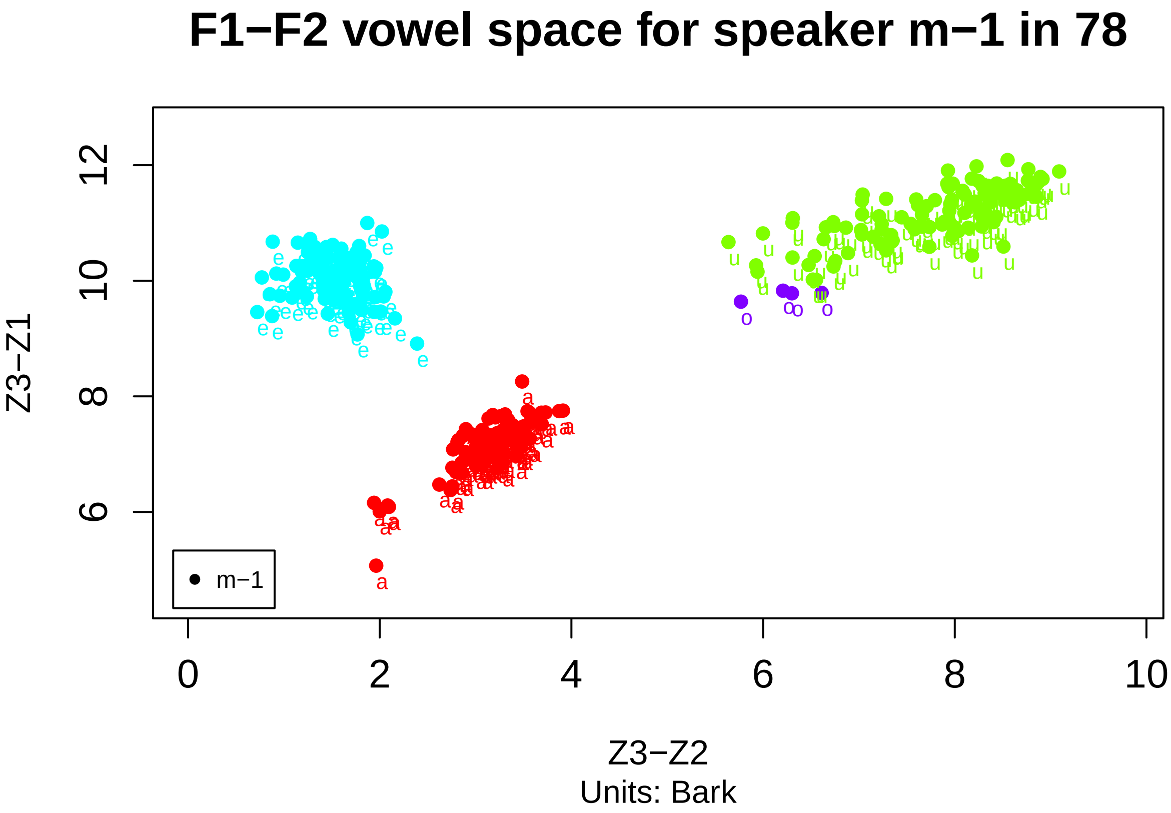
```r
vowels_plotting(vowels, 90, "f-2")
```



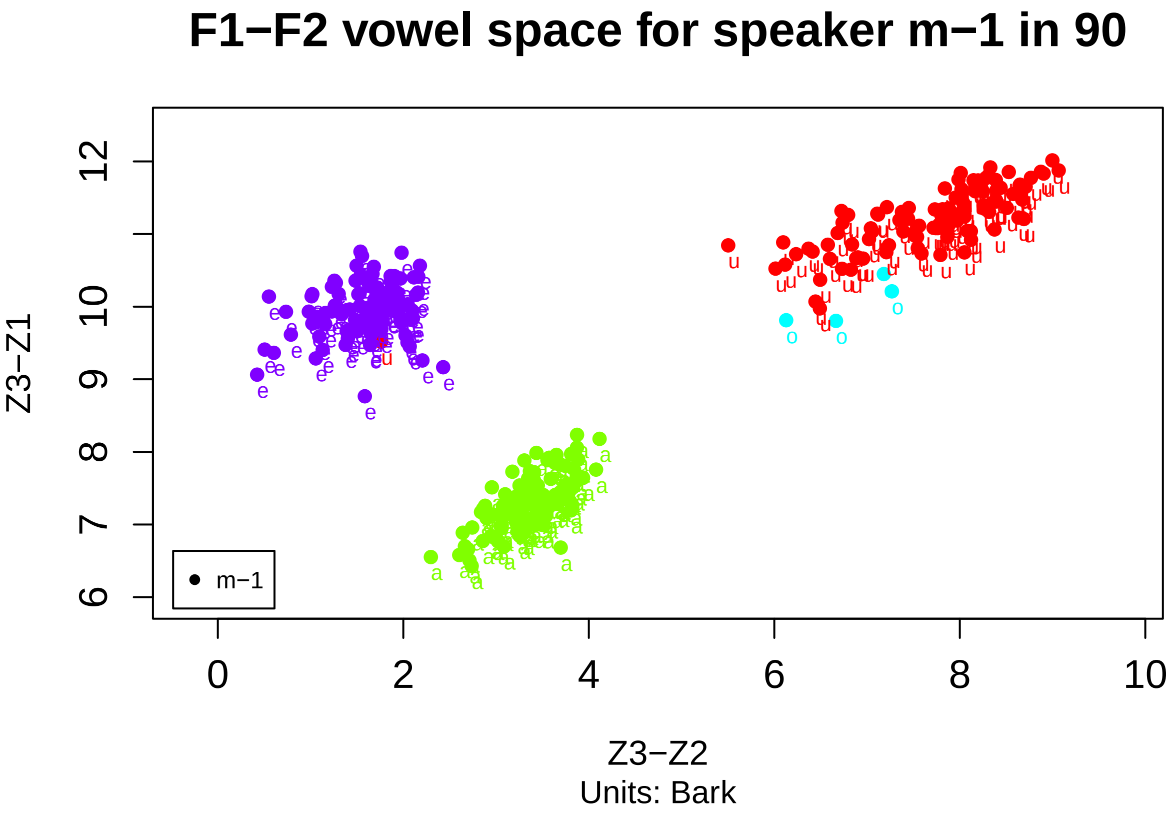
```
vowels_plotting(vowels, 0, "m-1")
```



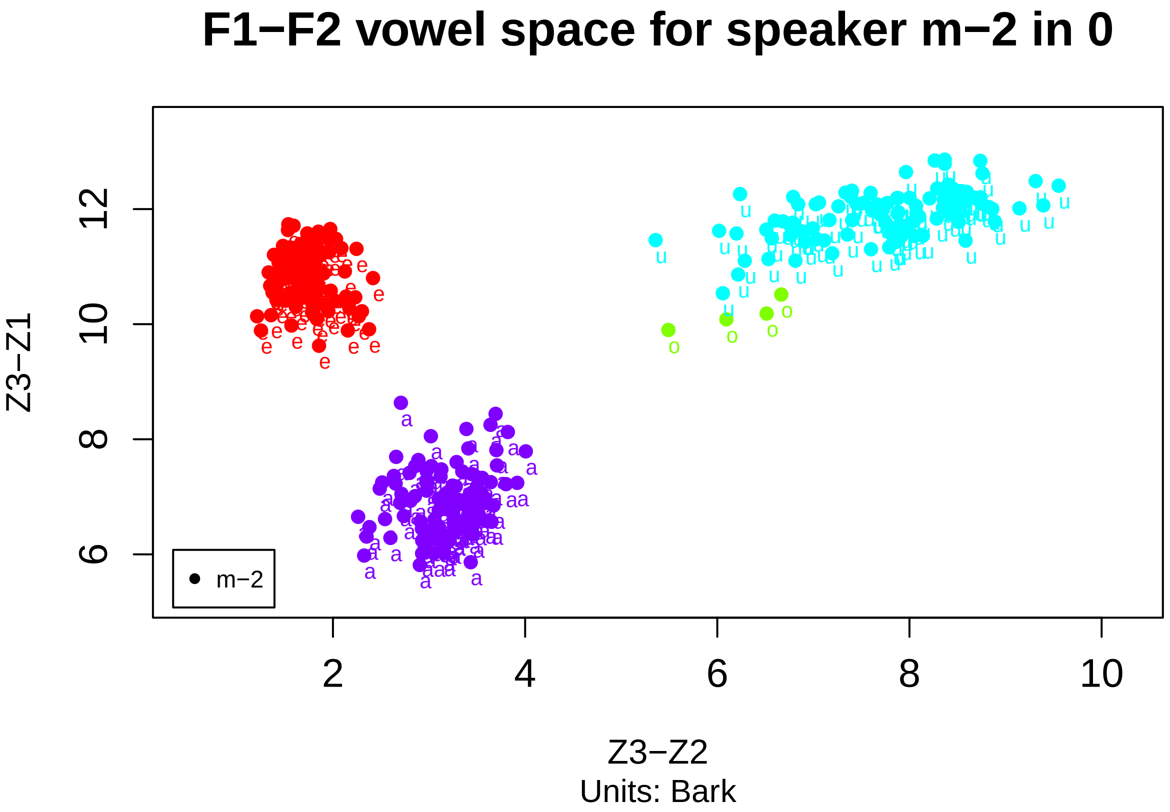
```
vowels_plotting(vowels, 78, "m-1")
```



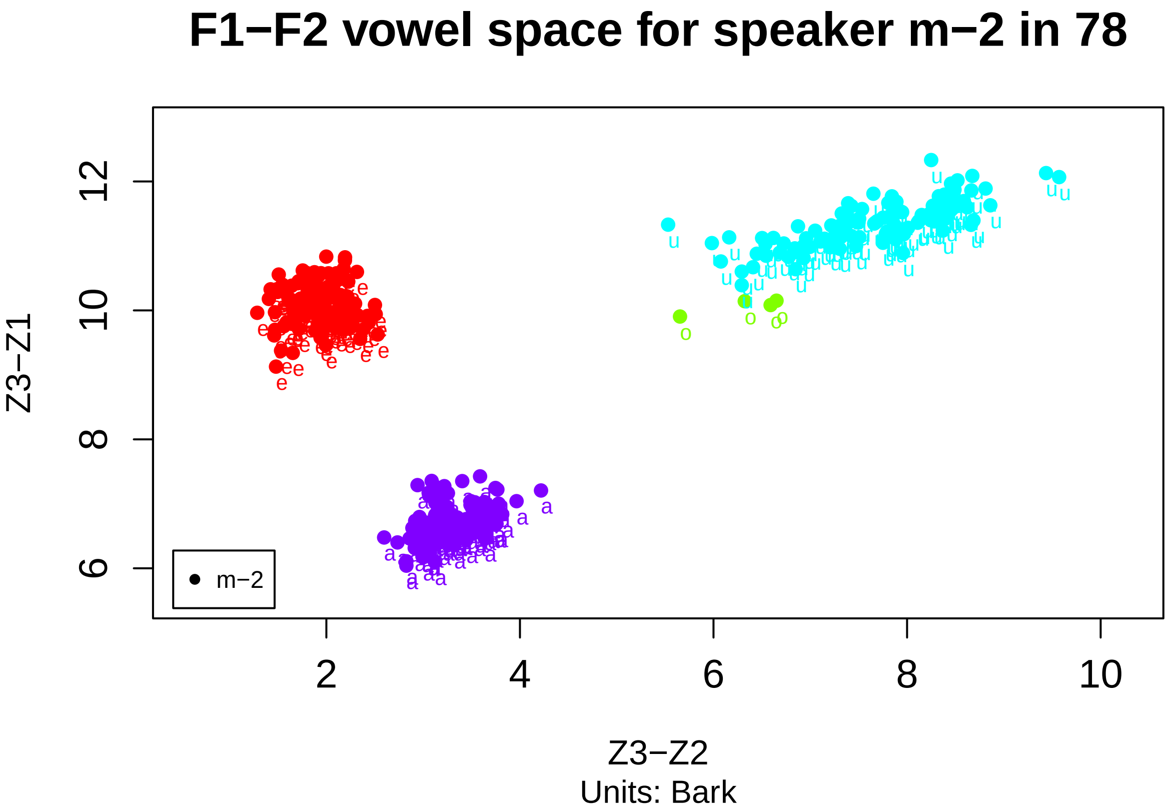
```r
vowels_plotting(vowels, 90, "m-1")
```



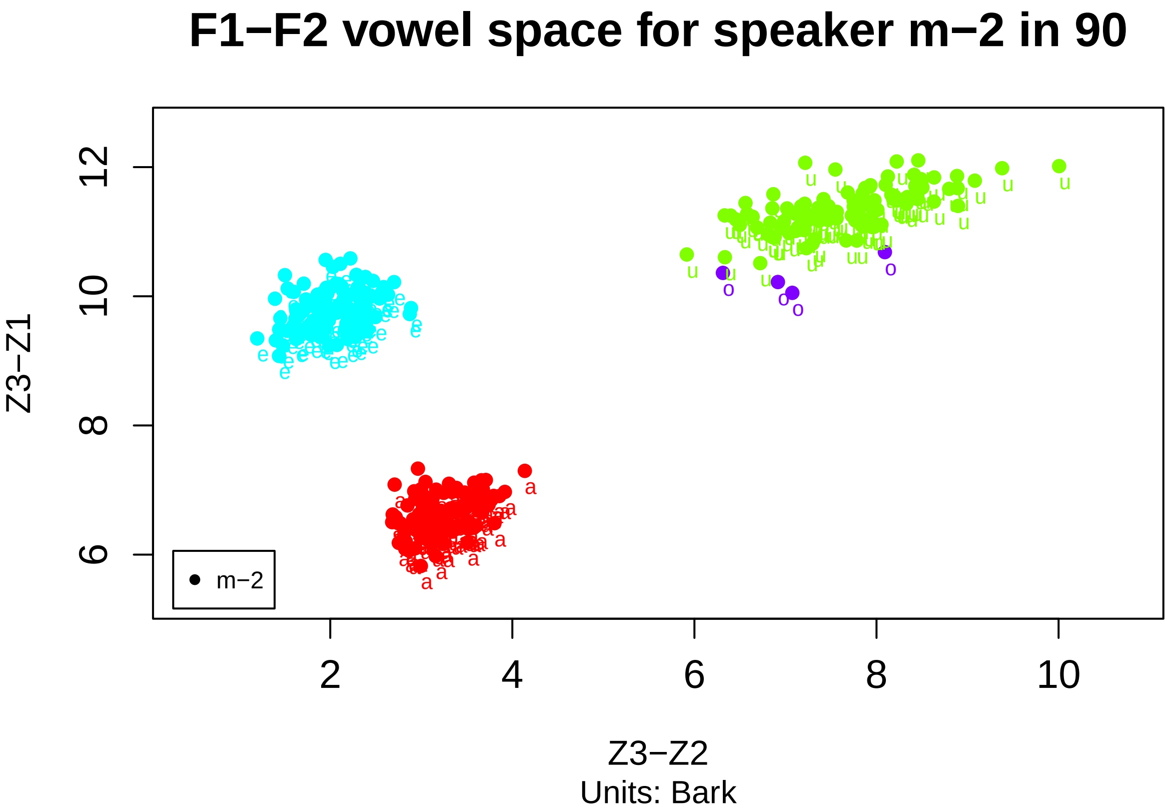
```
vowels_plotting(vowels, 0, "m-2")
```



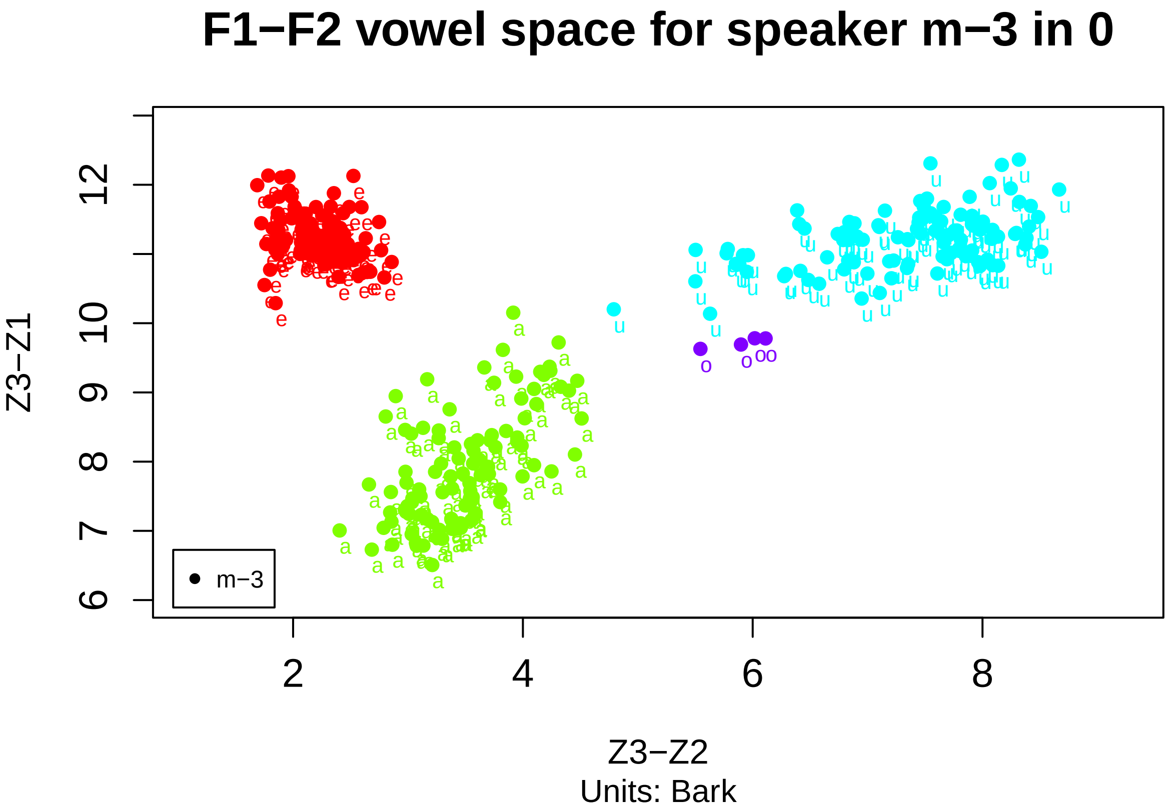
```
vowels_plotting(vowels, 78, "m-2")
```



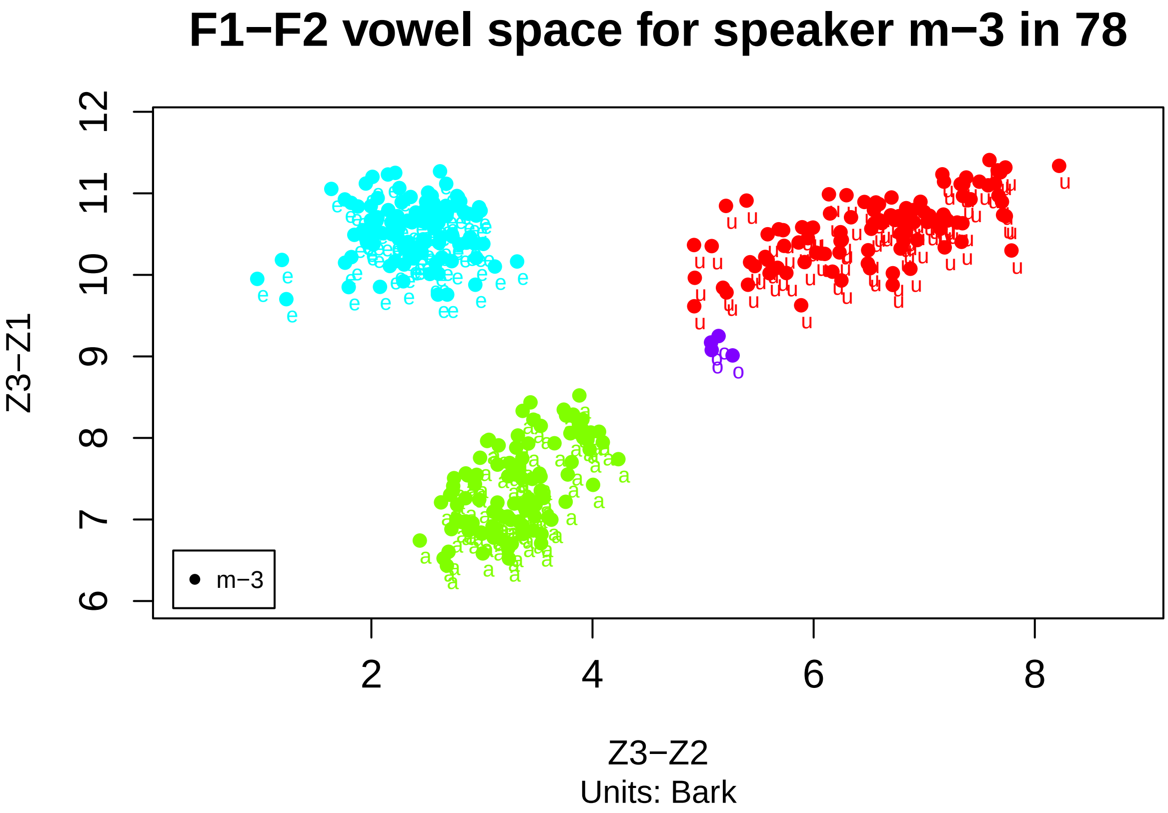
```
vowels_plotting(vowels, 90, "m-2")
```



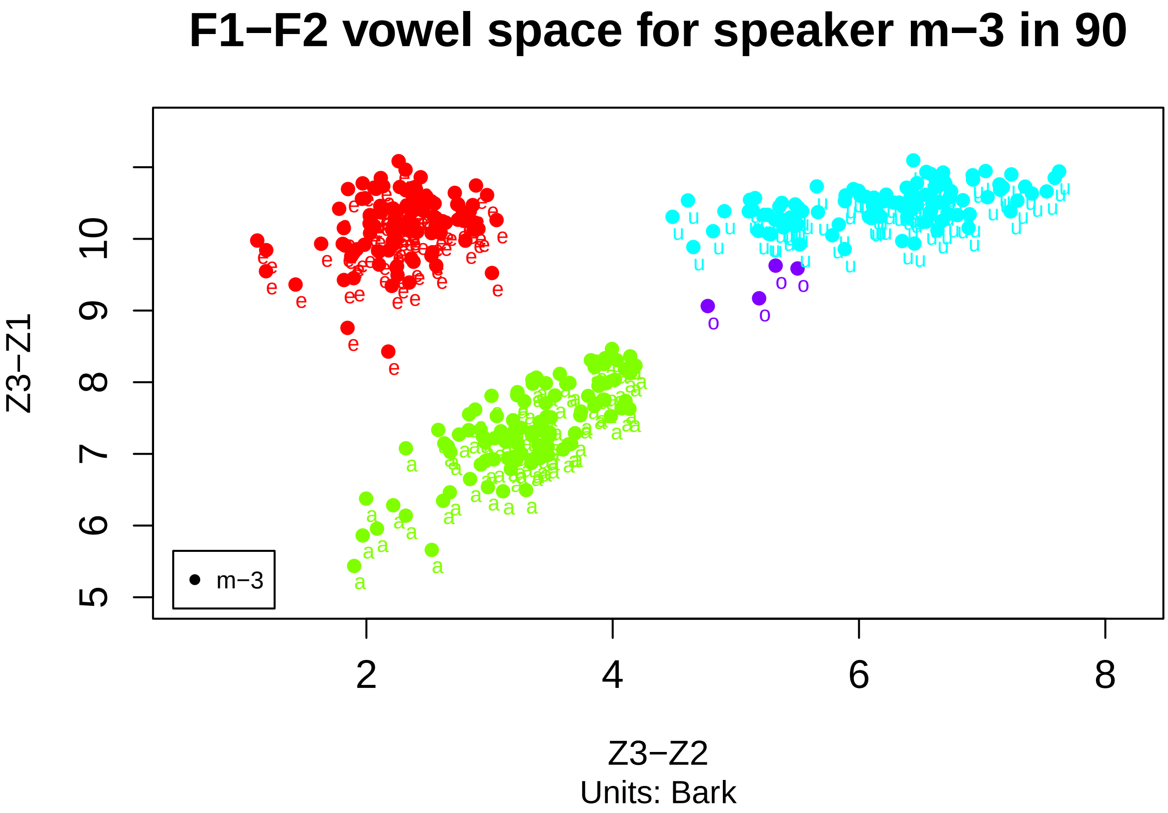
```
vowels_plotting(vowels, 0, "m-3")
```



```
vowels_plotting(vowels, 78, "m-3")
```

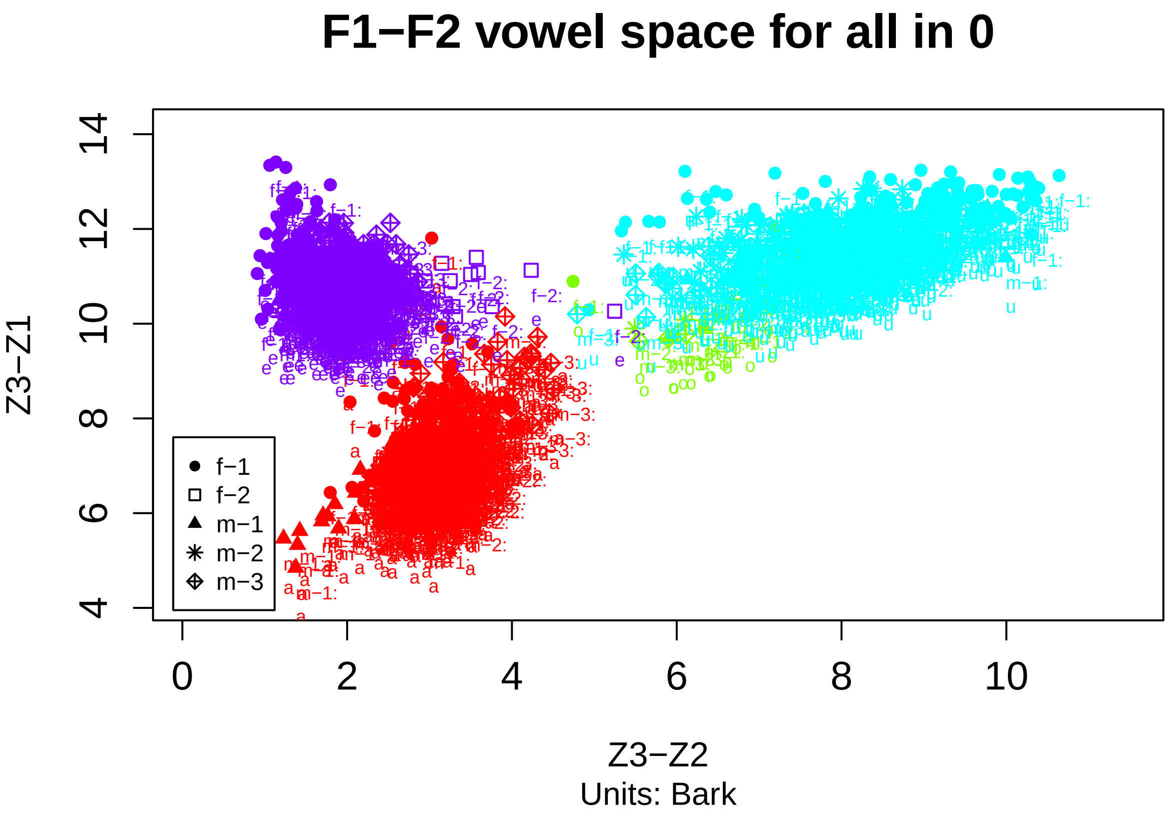


```
vowels_plotting(vowels, 90, "m-3")
```

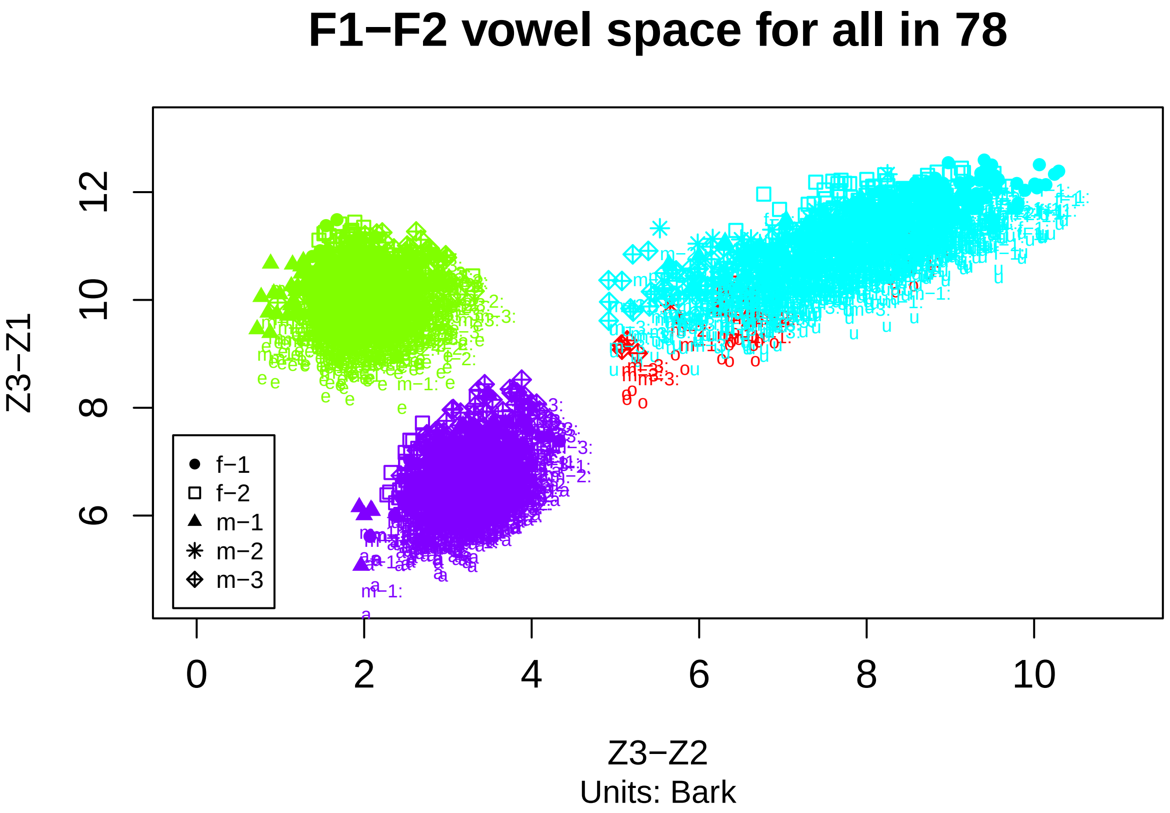


```
# par(mfrow=c(2,1))
# vowelplot(compute.means(vowels), shape="vowels")
# vowelplot(compute.means(norm.lobanov(vowels)), shape="vowels")

# par(mfrow=c(1,1))
# g09.means <- compute.means(vowels, speaker="f-1")
# vowelplot(g09.means, color="vowels", labels="none")
# add.spread.vowelplot(vowels, speaker="f-1", sd.mult=1, color="vowels", labels="none")
# # can add annotations to the vowel plots as any other R graph, eg:
# legend("top", legend="Can you guess which vowel is 'BOY'?", col='lightslategrey', bty="n")
```

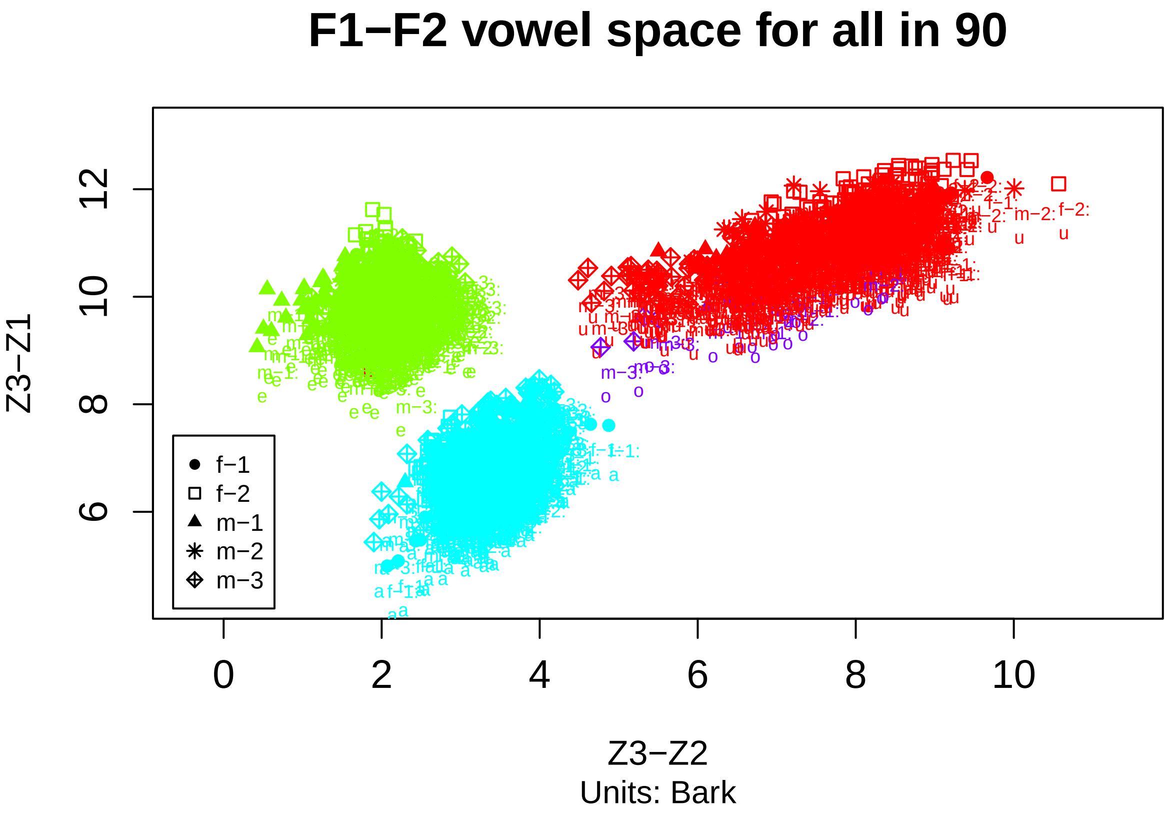
```
vowels_plotting(vowels, 0, "all")
```



```
vowels_plotting(vowels, 78, "all")
```



```
vowels_plotting(vowels, 90, "all")
```



Use phonR to calculate the hull area. Decreasing area found.

```
install.packages("phonR", repos = 'http://cran.us.r-project.org')
```

```
##
## The downloaded binary packages are in
##  /var/folders/9c/3_mgdyf12z7dvb8rt4d60nt80000gn/T//Rtmp3eURqD/downloaded_packages
```

```
library(phonR)
```

```
#head(vowels)

convexHullArea(vowels$F1, vowels$F2, group=vowels$speaker)
#vowelMeansPolygonArea(vowels$F1, vowels$F2, vowel, poly.order, group=NULL)

# reduced hull.area
hull.area <- with(vowels[vowels$speaker=="f-2",], convexHullArea(F1, F2, group=noise))
#    poly.area <- with(indo, vowelMeansPolygonArea(f1, f2, vowel,
#       poly.order=c("i", "e", "a", "o", "u"), group=subj))
hull.area
```

```
##        0       78       90
## 865452.4 759835.5 719835.8
```