

F0 Analysis

Giang Le

11/05/2021

Plan for Analysis

- Plot F0 and 95% confidence intervals on aggregate.
- Plot F0 and 95% confidence intervals by gender, by noise levels, by vowels, by vowel and noise levels, by environment (isolation or in a carrier sentence) and also create a regression line for each categorization.
- Summarize findings & email Yan Other notes Also leave one speaker out and try classification. If the result is good we can use it to predict tone produced on what kind of level (leave the noise level aside). The noise shouldn't matter much at this stage. (18 levels?) look at similarity of tones (using clustering).

```
# Read in all the F0 contour info. All of these f0.mat files have been downsampled
# in MatLab during linear interpolation (see getf0.mat in the matlab codes)

# matlab code:
# myFolder = '\client\h$\Desktop\ProsodyPro\m-3-78\channel1';
# myFiles = dir(fullfile(myFolder, '*.wav')); %gets all wav files
#
# for k = 1:length(myFiles)
#   baseFileName = myFiles(k).name;
#   fullFileName = fullfile(myFolder, baseFileName);
#   fprintf(1, 'Now reading %s\n', fullFileName);
#   [y, Fs] = audioread(fullFileName);
#   [f0, ~] = pitchRocco(y, Fs);
#   i = 1:length(f0);
#   i_new = linspace(min(i), max(i), 20);
#   f0_downsampled = interp1(i, f0, i_new, 'linear');
#   fid= fopen(fullfile(myFolder, 'F0.mat'), 'a');
#   fprintf(fid, '%s ', baseFileName);
#   fprintf(fid, '%f ', f0_downsampled);
#   fprintf(fid, '\n');
#   fclose(fid);
# end

# Step 1: Read in all F0.mat files into f0Files and assign a name for each

f0Files <- lapply(Sys.glob("*/channel1/F0.mat"), read.table)
length(f0Files)
```

```
## [1] 12
```

```
f0_f_1_78 <- as.data.frame(f0Files[1])
f0_f_1_90 <- as.data.frame(f0Files[2])
f0_f_1_q <- as.data.frame(f0Files[3])
f0_m_1_78 <- as.data.frame(f0Files[4])
f0_m_1_90 <- as.data.frame(f0Files[5])
f0_m_1_q <- as.data.frame(f0Files[6])
f0_m_2_78 <- as.data.frame(f0Files[7])
f0_m_2_90 <- as.data.frame(f0Files[8])
f0_m_2_q <- as.data.frame(f0Files[9])
f0_m_3_78 <- as.data.frame(f0Files[10])
f0_m_3_90 <- as.data.frame(f0Files[11])
f0_m_3_q <- as.data.frame(f0Files[12])
```

```
# Step 2: Add column names for all dataframes.
```

```
numbers <- 1:20
cols <- c("sound.name", numbers)
colnames(f0_f_1_78) <- cols
colnames(f0_f_1_90) <- cols
colnames(f0_f_1_q) <- cols
colnames(f0_m_1_78) <- cols
colnames(f0_m_1_90) <- cols
colnames(f0_m_1_q) <- cols
colnames(f0_m_2_78) <- cols
colnames(f0_m_2_90) <- cols
colnames(f0_m_2_q) <- cols
colnames(f0_m_3_78) <- cols
colnames(f0_m_3_90) <- cols
colnames(f0_m_3_q) <- cols
```

```
# Assigning noise level
```

```
f0_f_1_78$noise = 78
f0_f_1_90$noise = 90
f0_f_1_q$noise = 0
```

```
f0_m_1_78$noise = 78
f0_m_1_90$noise = 90
f0_m_1_q$noise = 0
```

```
f0_m_2_78$noise = 78
f0_m_2_90$noise = 90
f0_m_2_q$noise = 0
```

```
f0_m_3_78$noise = 78
f0_m_3_90$noise = 90
f0_m_3_q$noise = 0
```

```
# Assigning gender variable (0 for female and 1 for male)
```

```
f0_f_1_78$gender = 0
f0_f_1_90$gender = 0
f0_f_1_q$gender = 0
```

```
f0_m_1_78$gender = 1
f0_m_1_90$gender = 1
f0_m_1_q$gender = 1
```

```
f0_m_2_78$gender = 1
f0_m_2_90$gender = 1
f0_m_2_q$gender = 1
```

```
f0_m_3_78$gender = 1
f0_m_3_90$gender = 1
f0_m_3_q$gender = 1
```

```
### Concatenate all dataframes
```

```
f0_reports <- rbind(f0_f_1_78, f0_f_1_90, f0_f_1_q,
f0_m_1_78, f0_m_1_90, f0_m_1_q,
```

```
f0_m_2_78, f0_m_2_90, f0_m_2_q,  
f0_m_3_78, f0_m_3_90, f0_m_3_q)
```

According to the report, this dataframe has 3786 records. In total, from 4 speakers, we have 3786 sound segments to analyze.

```
# Assign tone values.  
  
# Extracting a substring that contains only the syllable names.  
nameswithoutwav <- sapply(strsplit(f0_reports[,1], split=". ", fixed=TRUE), "[", 1)  
f0_reports$syllable.names <- sapply(strsplit(nameswithoutwav, split="_ ", fixed=TRUE), "[", 1)  
  
f0_reports$tone <- ifelse(grepl("a", f0_reports$syllable.name, ignore.case=T), "A1",  
ifelse(grepl("^tát", f0_reports$syllable.name, ignore.case=T), "D1",  
ifelse(grepl("^tát", f0_reports$syllable.name, ignore.case=T), "D2",  
ifelse(grepl("^têt", f0_reports$syllable.name, ignore.case=T), "D1",  
ifelse(grepl("^têt", f0_reports$syllable.name, ignore.case=T), "D2",  
ifelse(grepl("^tút", f0_reports$syllable.name, ignore.case=T), "D1",  
ifelse(grepl("^tüt", f0_reports$syllable.name, ignore.case=T), "D2",  
ifelse(grepl("à", f0_reports$syllable.name, ignore.case=T), "A2",  
ifelse(grepl("á", f0_reports$syllable.name, ignore.case=T), "B1",  
ifelse(grepl("å", f0_reports$syllable.name, ignore.case=T), "C1",  
ifelse(grepl("ã", f0_reports$syllable.name, ignore.case=T), "C2",  
ifelse(grepl("ä", f0_reports$syllable.name, ignore.case=T), "B2",  
ifelse(grepl("ê", f0_reports$syllable.name, ignore.case=T), "A1",  
ifelse(grepl("è", f0_reports$syllable.name, ignore.case=T), "A2",  
ifelse(grepl("é", f0_reports$syllable.name, ignore.case=T), "B1",  
ifelse(grepl("ë", f0_reports$syllable.name, ignore.case=T), "C1",  
ifelse(grepl("ë", f0_reports$syllable.name, ignore.case=T), "C2",  
ifelse(grepl("ê", f0_reports$syllable.name, ignore.case=T), "B2",  
ifelse(grepl("u", f0_reports$syllable.name, ignore.case=T), "A1",  
ifelse(grepl("ù", f0_reports$syllable.name, ignore.case=T), "A2",  
ifelse(grepl("ú", f0_reports$syllable.name, ignore.case=T), "B1",  
ifelse(grepl("ú", f0_reports$syllable.name, ignore.case=T), "C1",  
ifelse(grepl("ü", f0_reports$syllable.name, ignore.case=T), "C2",  
ifelse(grepl("ü", f0_reports$syllable.name, ignore.case=T), "B2",  
ifelse(grepl("ö", f0_reports$syllable.name, ignore.case=T), "B2", "NA")))))))))))))))))))))  
  
# Assigning if the token is single (1) or not (0). Single tokens were produced in isolation.  
# Otherwise they were produced in carrier sentences.  
f0_reports$single <- ifelse(grepl("single", f0_reports$sound.name), 1, 0)  
  
# Convert categorical variables to factor levels.  
f0_reports$gender <- as.factor(f0_reports$gender)  
f0_reports$single <- as.factor(f0_reports$single)  
f0_reports$tone <- as.factor(f0_reports$tone)  
f0_reports$noise <- as.factor(f0_reports$noise)  
  
summary(f0_reports)
```

```

##   sound.name      1          2          3
## Length:3786    Min. : 0.00  Min. : 0.0  Min. : 0.0
## Class :character 1st Qu.: 0.00  1st Qu.:117.4  1st Qu.:125.1
## Mode  :character Median :115.91  Median :142.4  Median :147.2
##                   Mean  : 95.73  Mean  :140.6  Mean  :152.6
##                   3rd Qu.:148.73 3rd Qu.:169.8  3rd Qu.:175.9
##                   Max.  :376.52  Max.  :353.6  Max.  :369.0
##
##   4          5          6          7
## Min. : 0.0  Min. : 0.0  Min. : 0.0  Min. : 0.0
## 1st Qu.:125.9 1st Qu.:124.7 1st Qu.:122.4 1st Qu.:119.8
## Median :147.1  Median :146.9  Median :145.0  Median :143.4
## Mean   :154.6  Mean   :153.9  Mean   :151.7  Mean   :149.1
## 3rd Qu.:176.9 3rd Qu.:177.5 3rd Qu.:177.0 3rd Qu.:174.8
## Max.  :322.7  Max.  :308.9  Max.  :304.4  Max.  :306.9
##
##   8          9          10         11
## Min. : 0.0  Min. : 0.0  Min. : 0.0  Min. : 0.0
## 1st Qu.:117.5 1st Qu.:115.8 1st Qu.:114.2 1st Qu.:113.0
## Median :142.3  Median :141.7  Median :140.9  Median :139.3
## Mean   :147.5  Mean   :147.0  Mean   :146.3  Mean   :145.7
## 3rd Qu.:173.8 3rd Qu.:174.0 3rd Qu.:174.7 3rd Qu.:176.3
## Max.  :307.6  Max.  :327.2  Max.  :308.7  Max.  :312.0
##
##   12         13         14         15
## Min. : 0.0  Min. : 0.0  Min. : 0.0  Min. : 0.0
## 1st Qu.:111.0 1st Qu.:109.6 1st Qu.:108.5 1st Qu.:107.5
## Median :139.1  Median :139.8  Median :139.9  Median :140.6
## Mean   :144.3  Mean   :144.6  Mean   :144.7  Mean   :144.9
## 3rd Qu.:177.7 3rd Qu.:180.2 3rd Qu.:183.3 3rd Qu.:185.6
## Max.  :319.8  Max.  :327.8  Max.  :336.7  Max.  :356.8
##
##   16         17         18         19
## Min. : 0.0  Min. : 0.0  Min. : 0.0  Min. : 0.0
## 1st Qu.:107.3 1st Qu.:105.6 1st Qu.:102.3 1st Qu.: 96.9
## Median :141.2  Median :138.4  Median :134.2  Median :127.6
## Mean   :145.2  Mean   :142.8  Mean   :137.2  Mean   :129.0
## 3rd Qu.:186.7 3rd Qu.:184.8 3rd Qu.:180.4 3rd Qu.:170.6
## Max.  :398.6  Max.  :406.8  Max.  :413.7  Max.  :377.8
##
##   20      noise   gender syllable.names      tone   single
## Min. : 0.00  0 :1260  0: 947  Length:3786      B2     :624  0:1893
## 1st Qu.: 0.00 78:1263  1:2839  Class :character  A2     :575  1:1893
## Median : 97.47 90:1263                Mode  :character  B1     :575
## Mean   : 83.49                                C1     :575
## 3rd Qu.:135.47                                C2     :575
## Max.  :343.06                                A1     :574
##                                         (Other):288

```

Aggregate Tone Analysis

F0 Plots

```

# Note that I only plot the 2nd to 19th sampled points.
# Extract F0 of all tones A1 (574 instances)
par(mfrow=c(2,4))
A1 <- data.matrix(f0_reports[f0_reports$tone=="A1",-c(1,c(22:26))])
plot(colMeans(A1)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 200))

# # Do the same for the other tones
A2 <- data.matrix(f0_reports[f0_reports$tone=="A2",-c(1,c(22:26))])
plot(colMeans(A2)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 200))

B1 <- data.matrix(f0_reports[f0_reports$tone=="B1",-c(1,c(22:26))])
plot(colMeans(B1)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 200))

B2 <- data.matrix(f0_reports[f0_reports$tone=="B2",-c(1,c(22:26))])
plot(colMeans(B2)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 250))

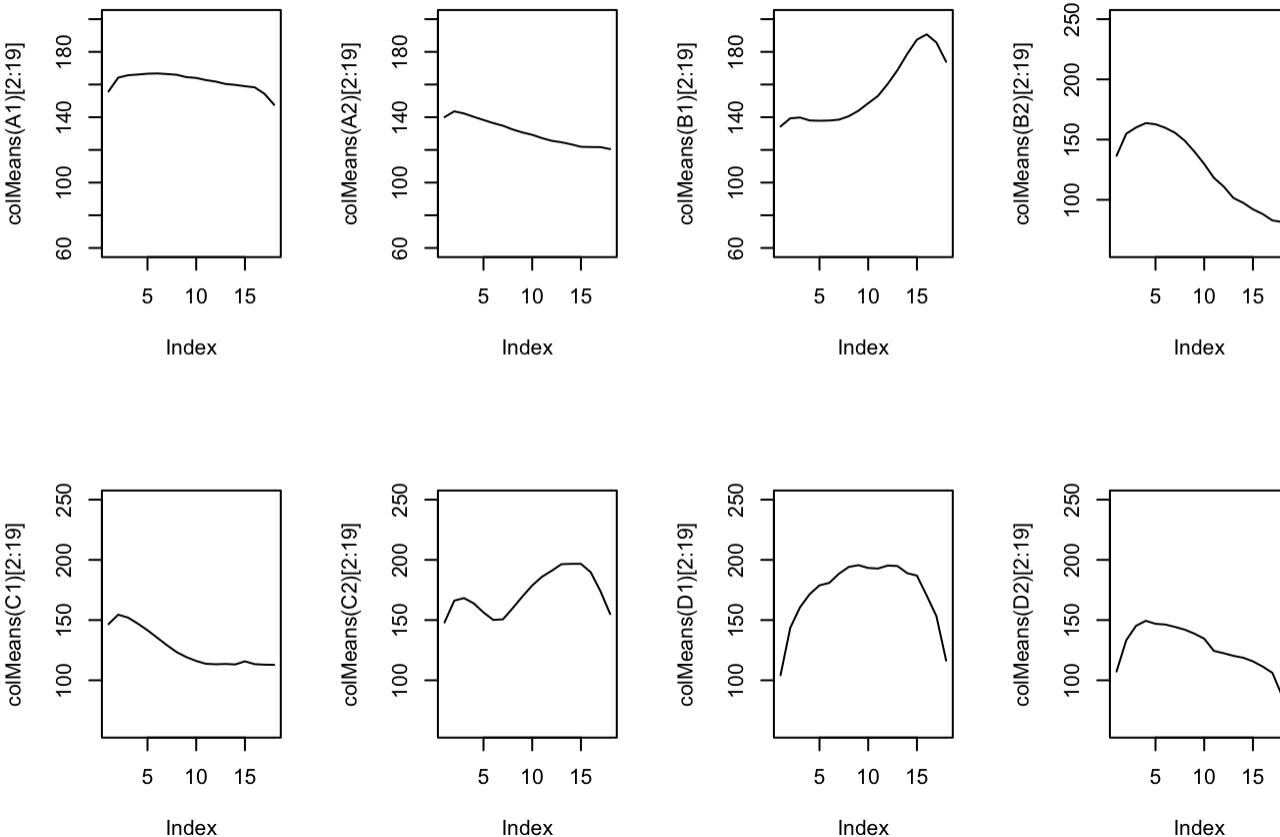
C1 <- data.matrix(f0_reports[f0_reports$tone=="C1",-c(1,c(22:26))])
plot(colMeans(C1)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 250))

C2 <- data.matrix(f0_reports[f0_reports$tone=="C2",-c(1,c(22:26))])
plot(colMeans(C2)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 250))

D1 <- data.matrix(f0_reports[f0_reports$tone=="D1",-c(1,c(22:26))])
plot(colMeans(D1)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 250))

D2 <- data.matrix(f0_reports[f0_reports$tone=="D2",-c(1,c(22:26))])
plot(colMeans(D2)[2:19], type="l", xlim=c(1, 18), ylim=c(60, 250))

```



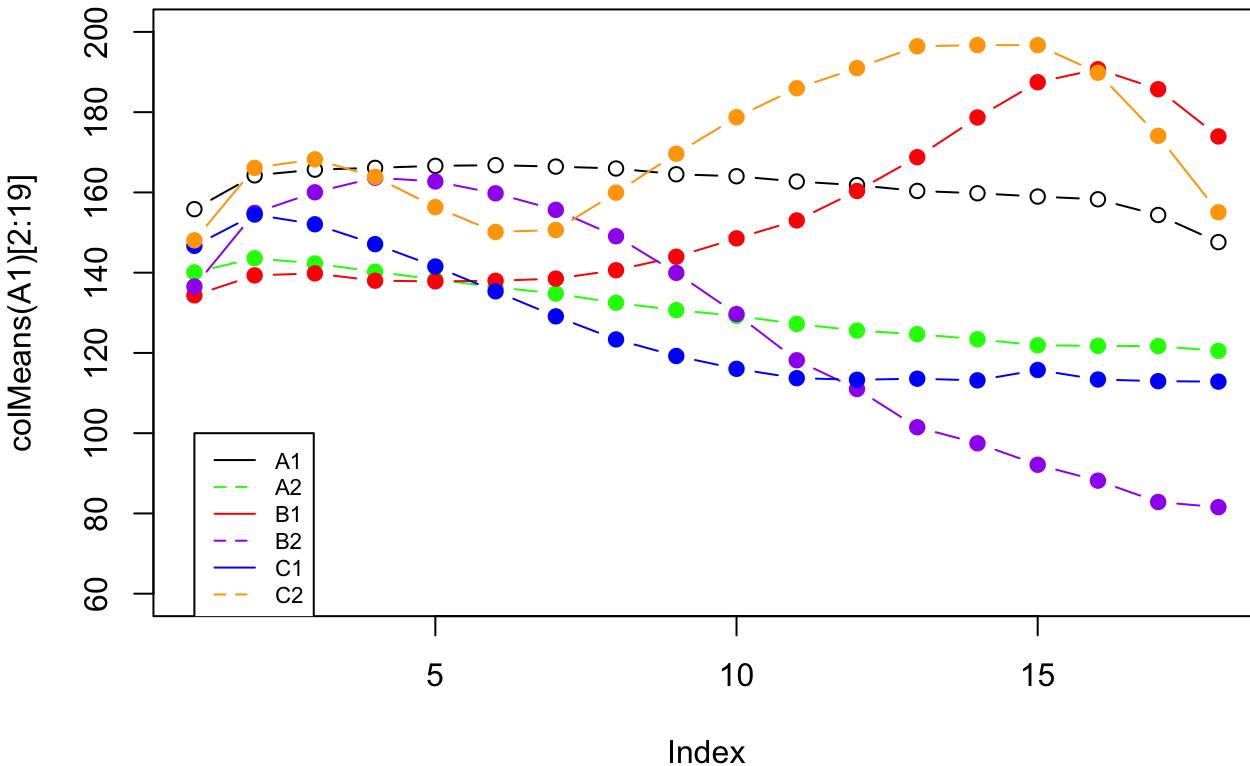
F0 contours on the same plot.

```

plot(colMeans(A1)[2:19], type="b", xlim=c(1, 18), ylim=c(60, 200))
lines(colMeans(A2)[2:19], col="green", type="b", pch=19)
lines(colMeans(B1)[2:19], col="red", type="b", pch=19)
lines(colMeans(B2)[2:19], col="purple", type="b", pch=19)
lines(colMeans(C1)[2:19], col="blue", type="b", pch=19)
lines(colMeans(C2)[2:19], col="orange", type="b", pch=19)

# Add a legend
legend(1, 100, legend=c("A1", "A2", "B1", "B2", "C1", "C2"),
       col=c("black", "green", "red", "purple", "blue", "orange"), lty=1:2, cex=0.7)

```



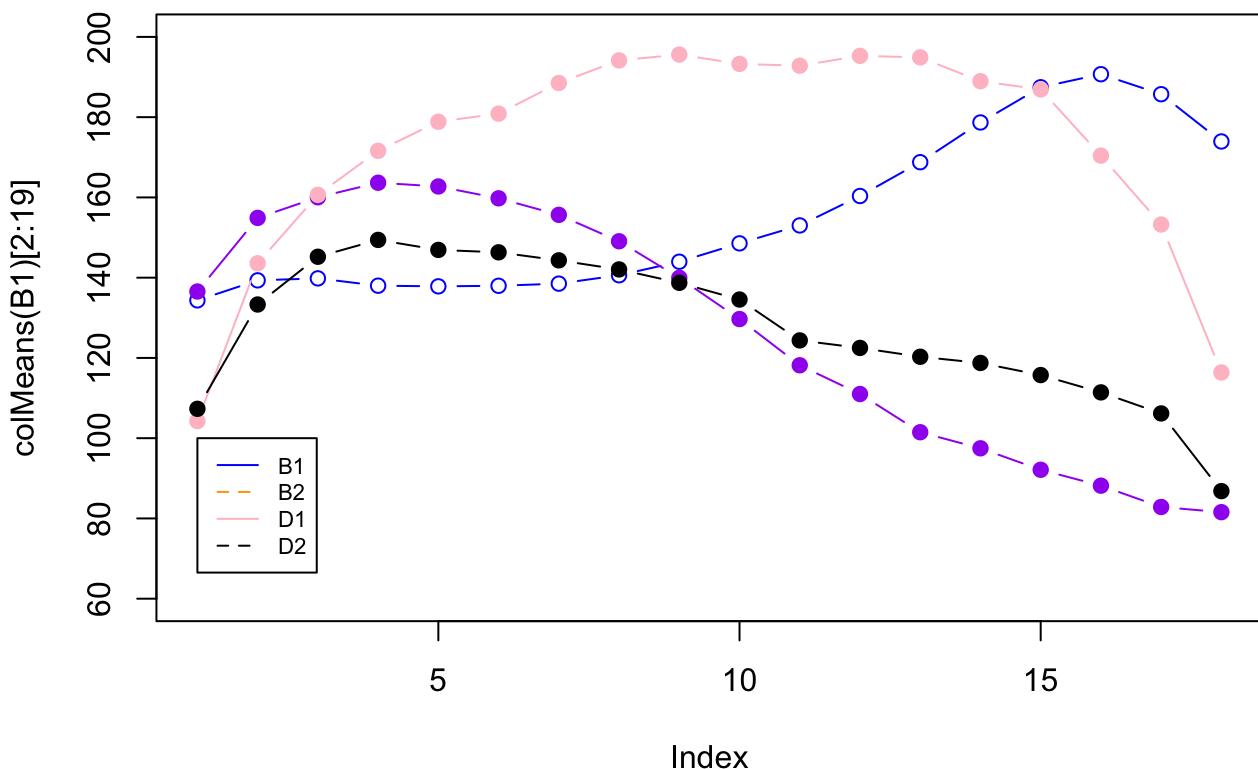
Plot B1, B2, D1, D2 together.

```

plot(colMeans(B1)[2:19], type="b", col="blue", xlim=c(1, 18), ylim=c(60, 200))
lines(colMeans(B2)[2:19], col="purple", type="b", pch=19)
lines(colMeans(D1)[2:19], col="pink", type="b", pch=19) ## D1 has a very strange contour.
lines(colMeans(D2)[2:19], col="black", type="b", pch=19)

# Add a legend
legend(1, 100, legend=c("B1", "B2", "D1", "D2"),
       col=c("blue", "orange", "pink", "black"), lty=1:2, cex=0.7)

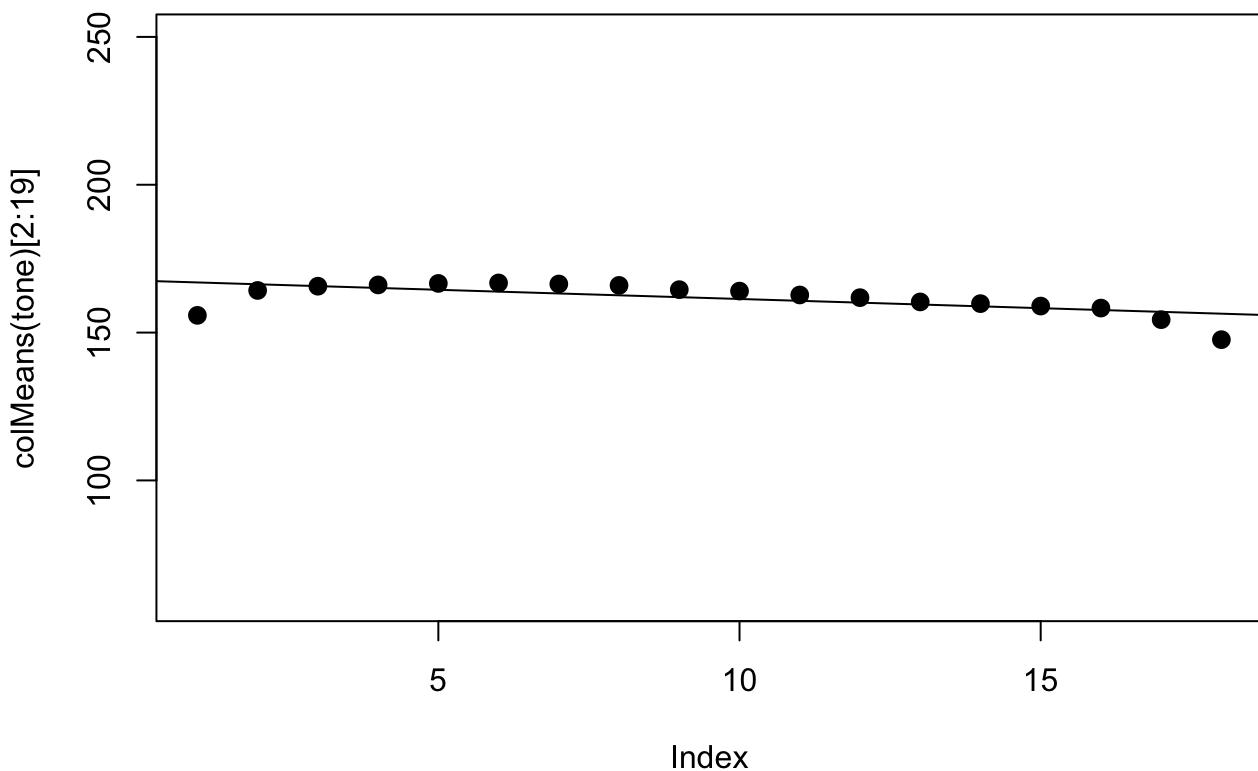
```



Estimate Regression Lines for Each Tone on Aggregate.

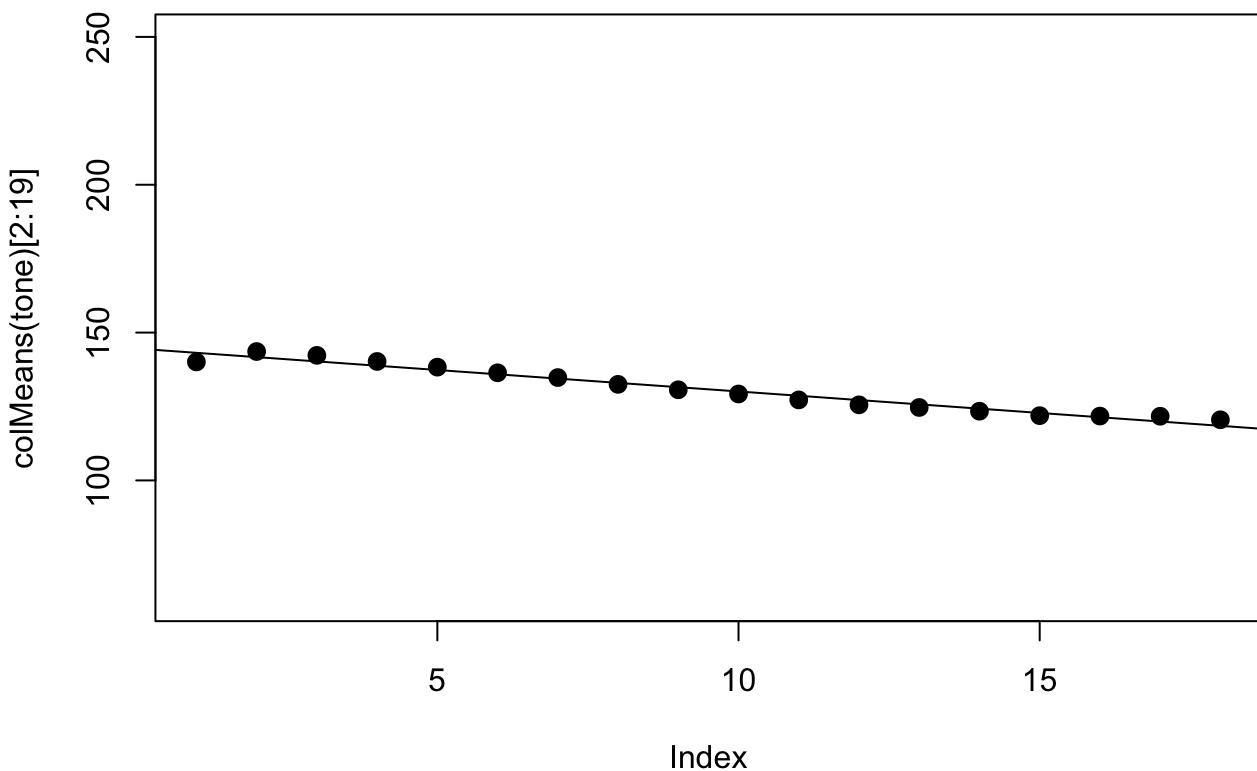
```
regression_report <- function(tone) {
  lm_tone <- lm(colMeans(tone)[2:19] ~ c(1:18))
  plot(colMeans(tone)[2:19], pch = 16, cex = 1.3, xlim=c(1, 18), ylim=c(60, 250))
  abline(lm(colMeans(tone)[2:19] ~ c(1:18)))
  return(lm_tone)
}
```

```
## Call regression_report on all tones.
regression_report(A1)
```



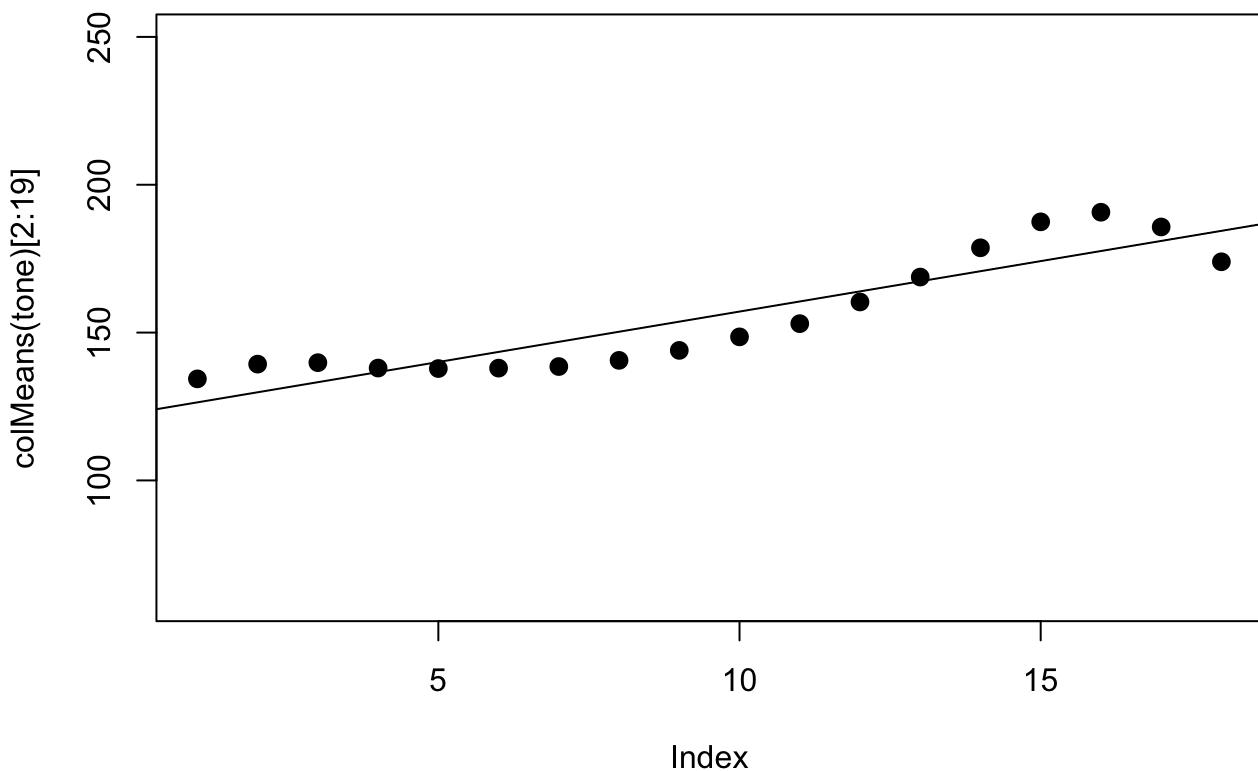
```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       167.5833     -0.6218
```

```
regression_report(A2)
```



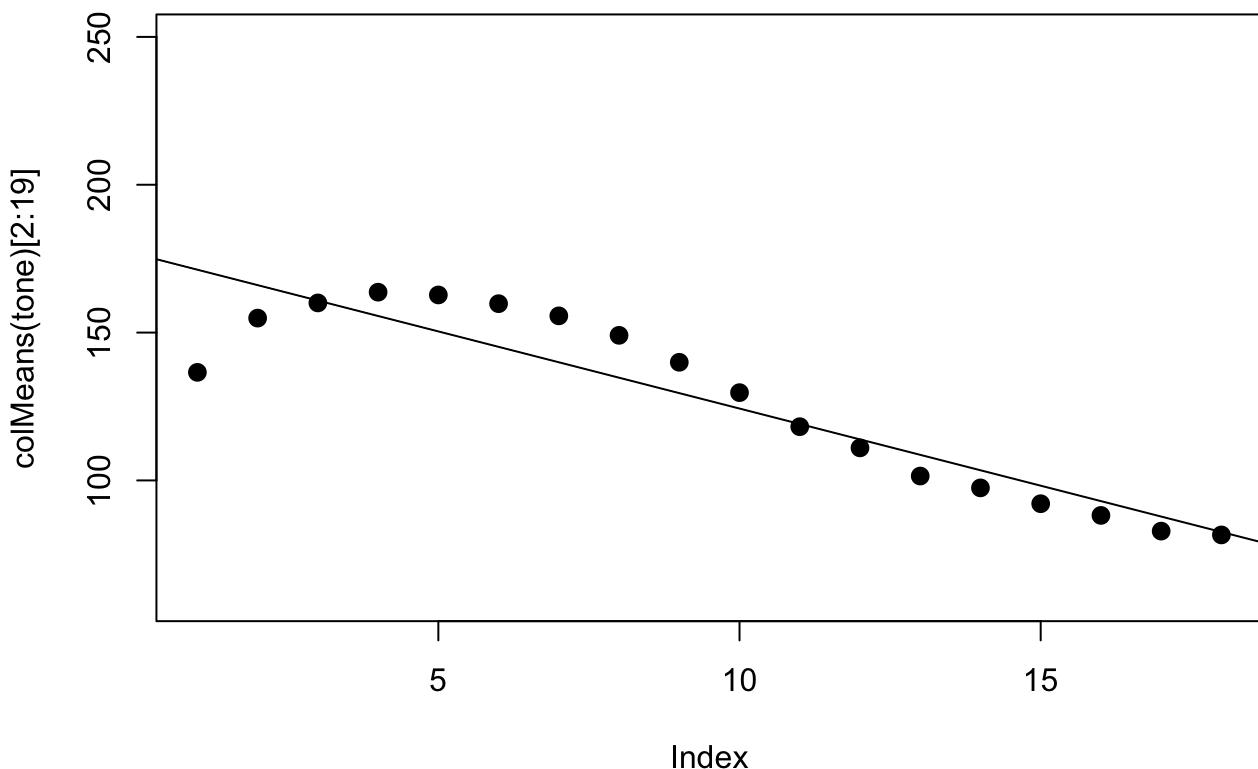
```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       144.641     -1.454
```

```
regression_report(B1)
```



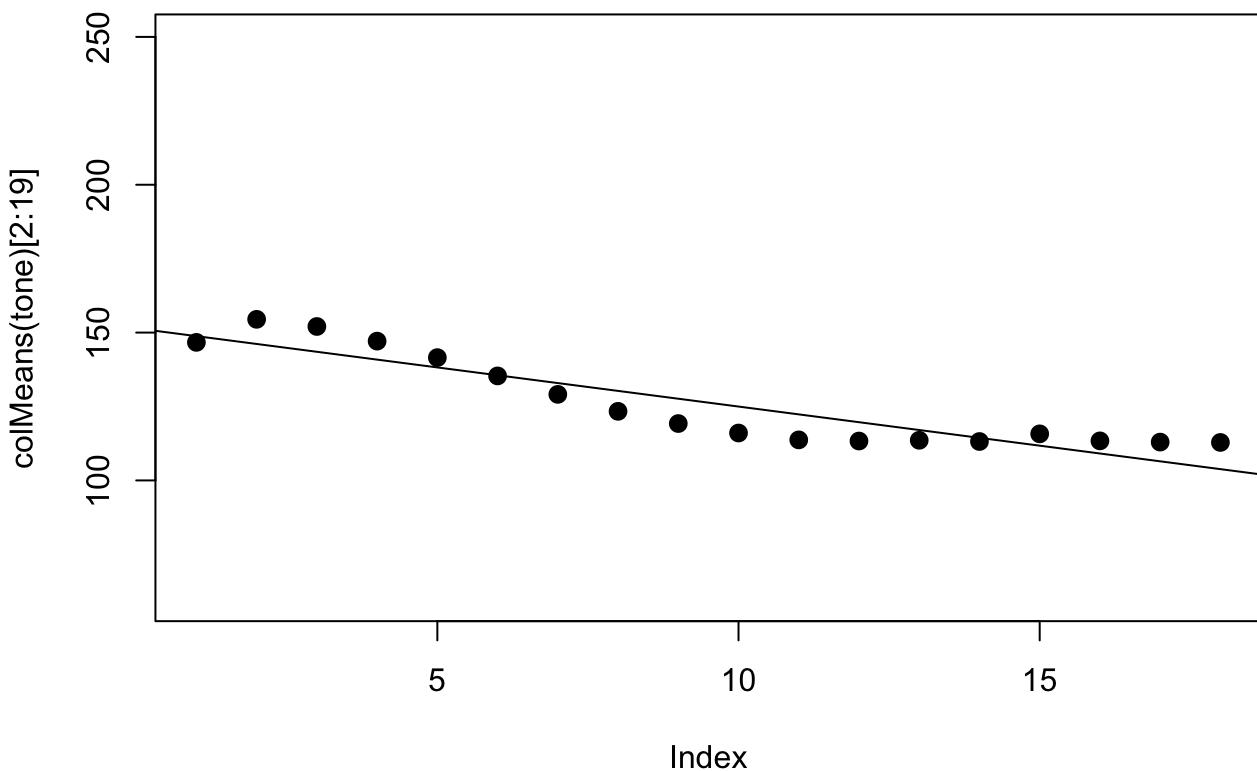
```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       122.987        3.414
```

```
regression_report(B2)
```



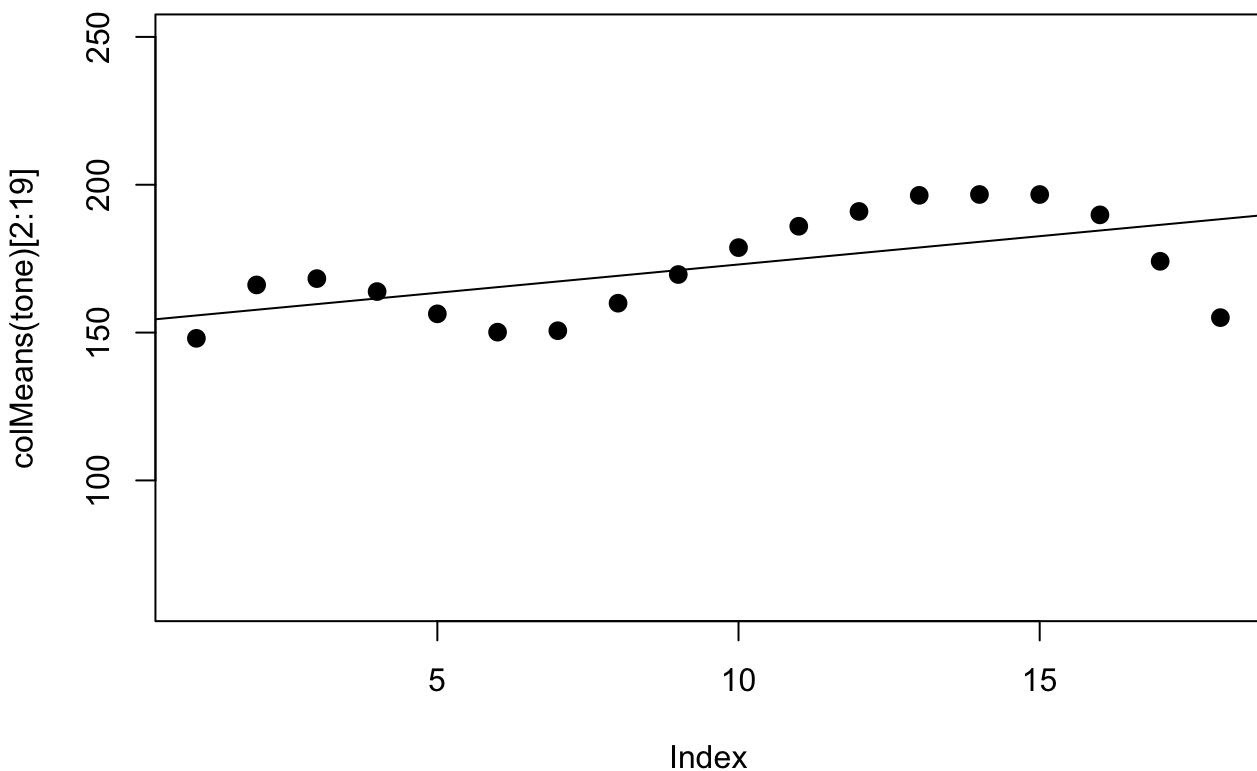
```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       176.53        -5.22
```

```
regression_report(C1)
```



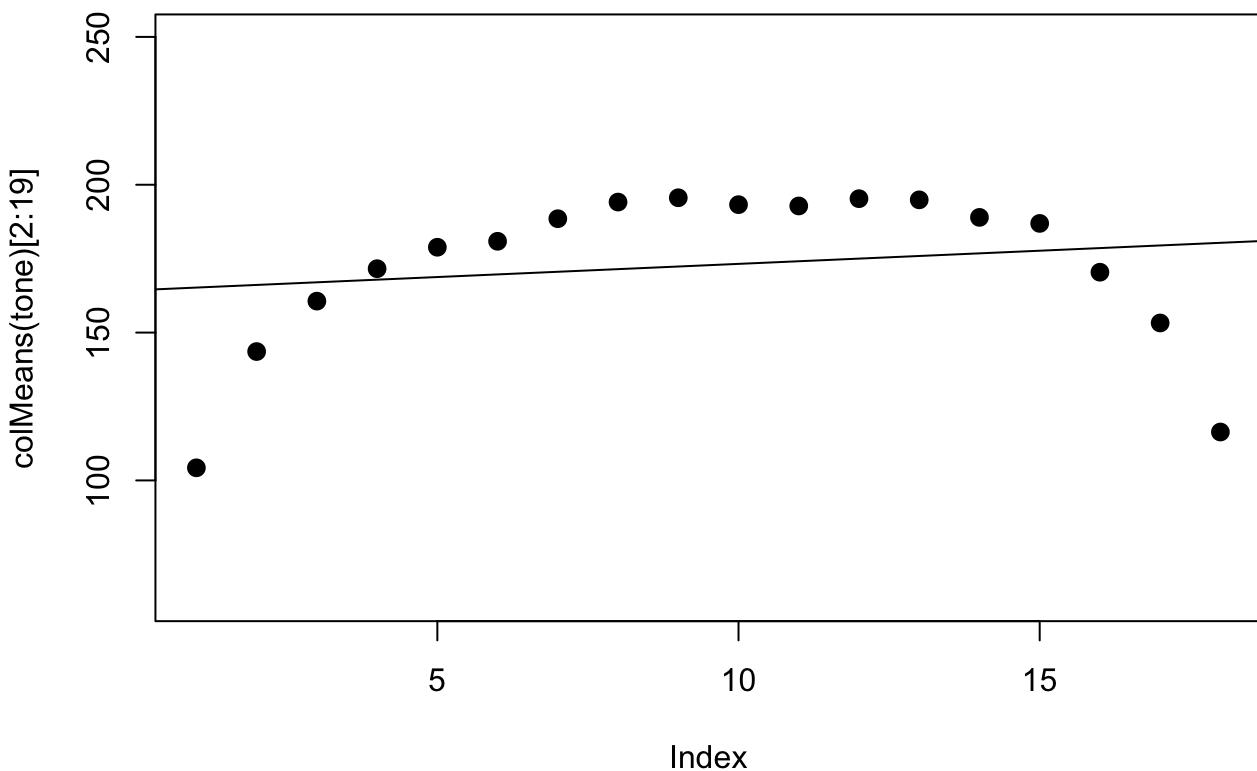
```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       151.462     -2.647
```

```
regression_report(C2)
```



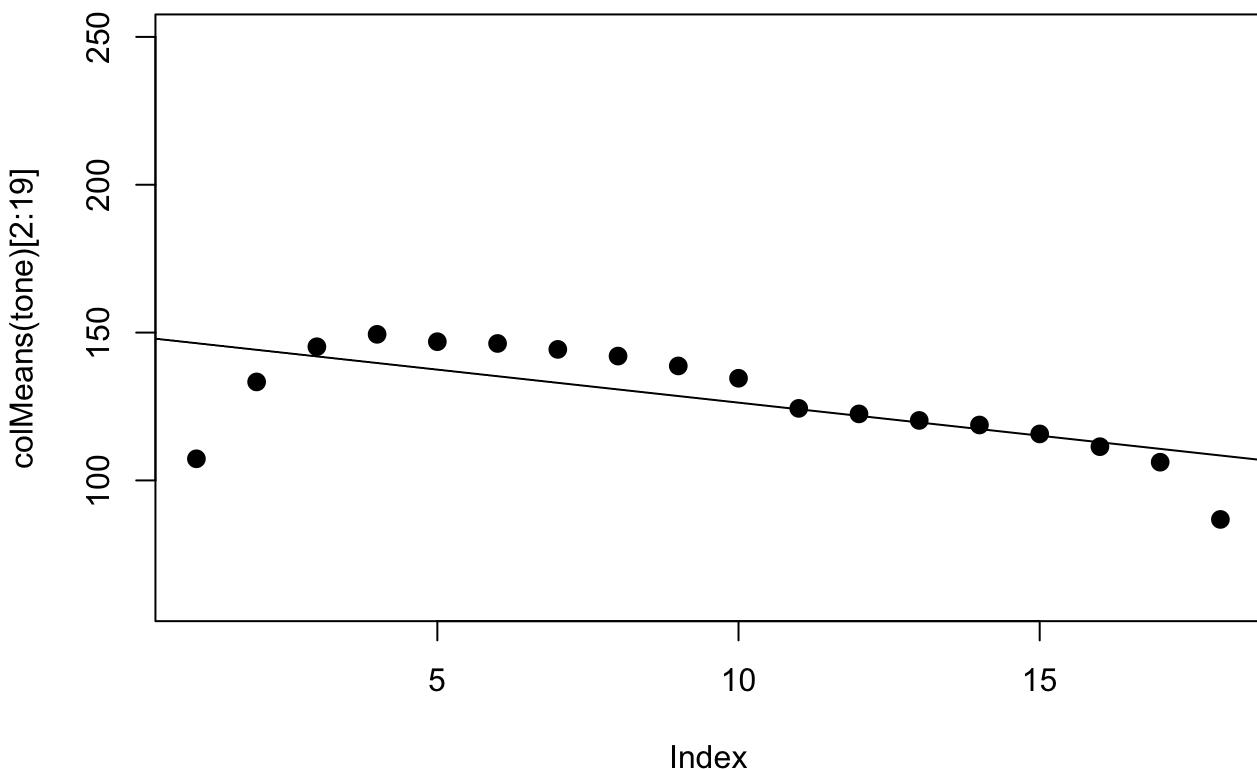
```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       153.883        1.916
```

```
regression_report(D1)
```



```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       164.3336      0.8907
```

```
regression_report(D2)
```



```
##  
## Call:  
## lm(formula = colMeans(tone)[2:19] ~ c(1:18))  
##  
## Coefficients:  
## (Intercept)      c(1:18)  
##       148.707     -2.237
```

Tone Contour Analysis on Different Noise Levels (Separately by Single vs. Tokens in Carriers)

```

# Plot F0 contours according to different levels

filter_f0 <- function(tone, noise, s) {
  if (s==TRUE) {
    tone_matrix <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise==noise & f0_reports$single=="1",-c(1,c(22:26))])
    return(tone_matrix)
  } else {
    tone_matrix <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise==noise & f0_reports$single=="0",-c(1,c(22:26))])
    return(tone_matrix)
  }
}

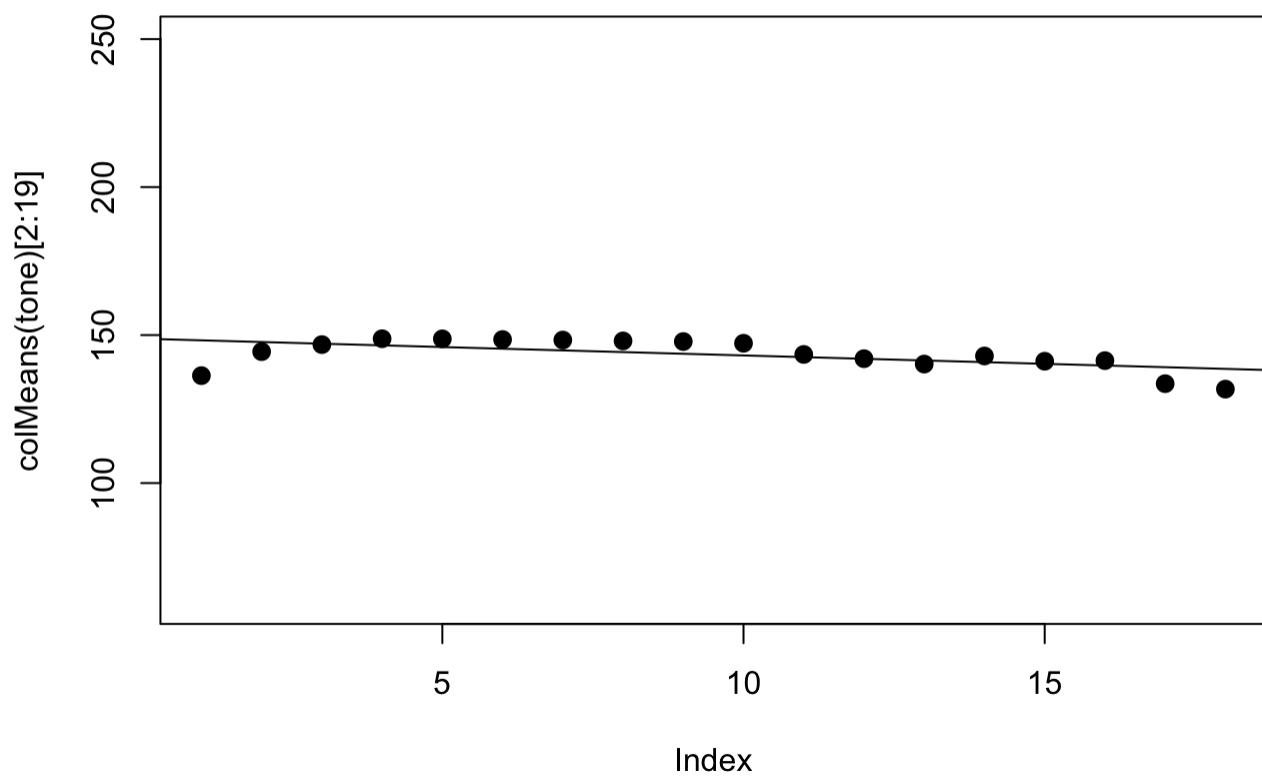
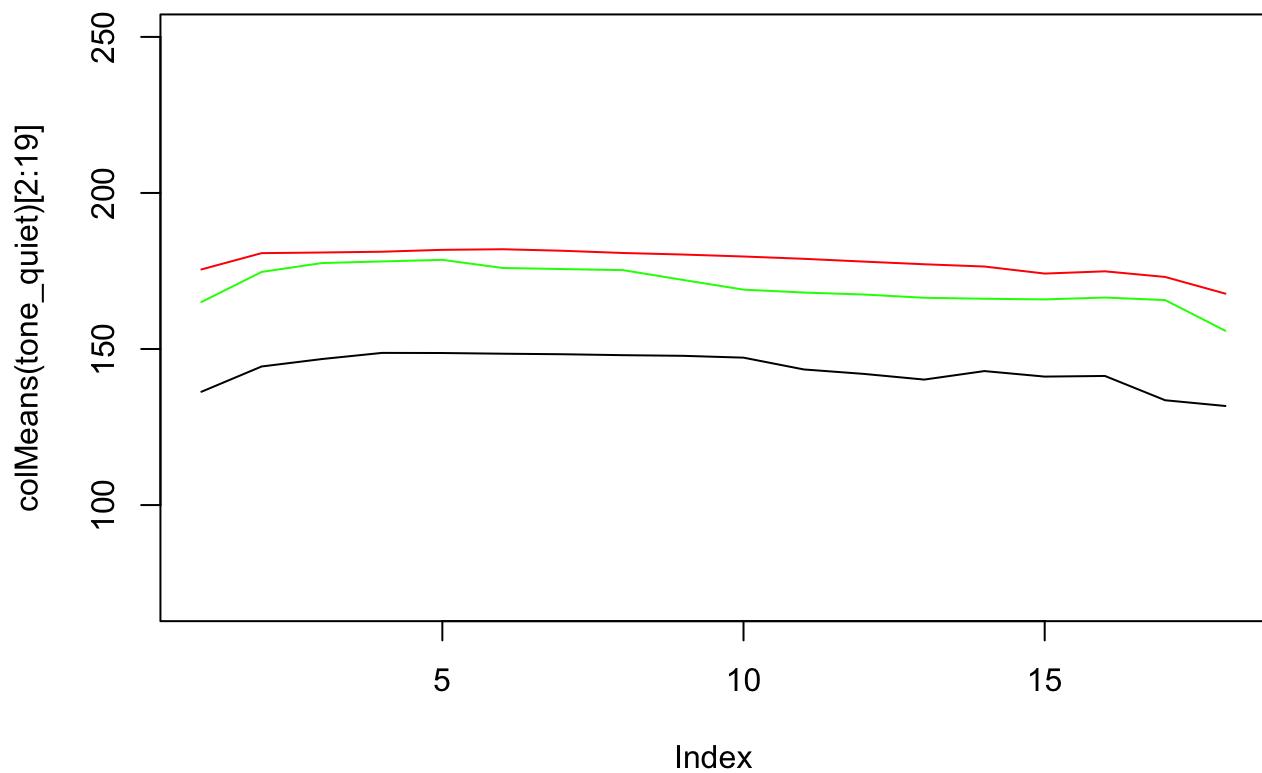
plot_f0_by_noise <- function(tone, s) {
  if (s==TRUE) {
    tone_quiet <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="0" & f0_reports$single=="1",-c(1,c(22:26))])
    tone_78 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="78" & f0_reports$single=="1",-c(1,c(22:26))])
    tone_90 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="90" & f0_reports$single=="1",-c(1,c(22:26))])
    plot(colMeans(tone_quiet)[2:19], type="l", xlim=c(1, 18), ylim=c(70, 250))
    lines(colMeans(tone_78)[2:19], col="green")
    lines(colMeans(tone_90)[2:19], col="red")
    # Find regression coefficients and return them.
    c(summary(regression_report(tone_quiet))$coefficients[2,1], summary(regression_report(tone_78))$coefficients[2,1], summary(regression_report(tone_90))$coefficients[2,1])
  } else {
    tone_quiet <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="0" & f0_reports$single=="0",-c(1,c(22:26))])
    tone_78 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="78" & f0_reports$single=="0",-c(1,c(22:26))])
    tone_90 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="90" & f0_reports$single=="0",-c(1,c(22:26))])
    plot(colMeans(tone_quiet)[2:19], type="l", xlim=c(1, 18), ylim=c(70, 250))
    lines(colMeans(tone_78)[2:19], col="green")
    lines(colMeans(tone_90)[2:19], col="red")
    c(summary(regression_report(tone_quiet))$coefficients[2,1], summary(regression_report(tone_78))$coefficients[2,1], summary(regression_report(tone_90))$coefficients[2,1])
  }
}

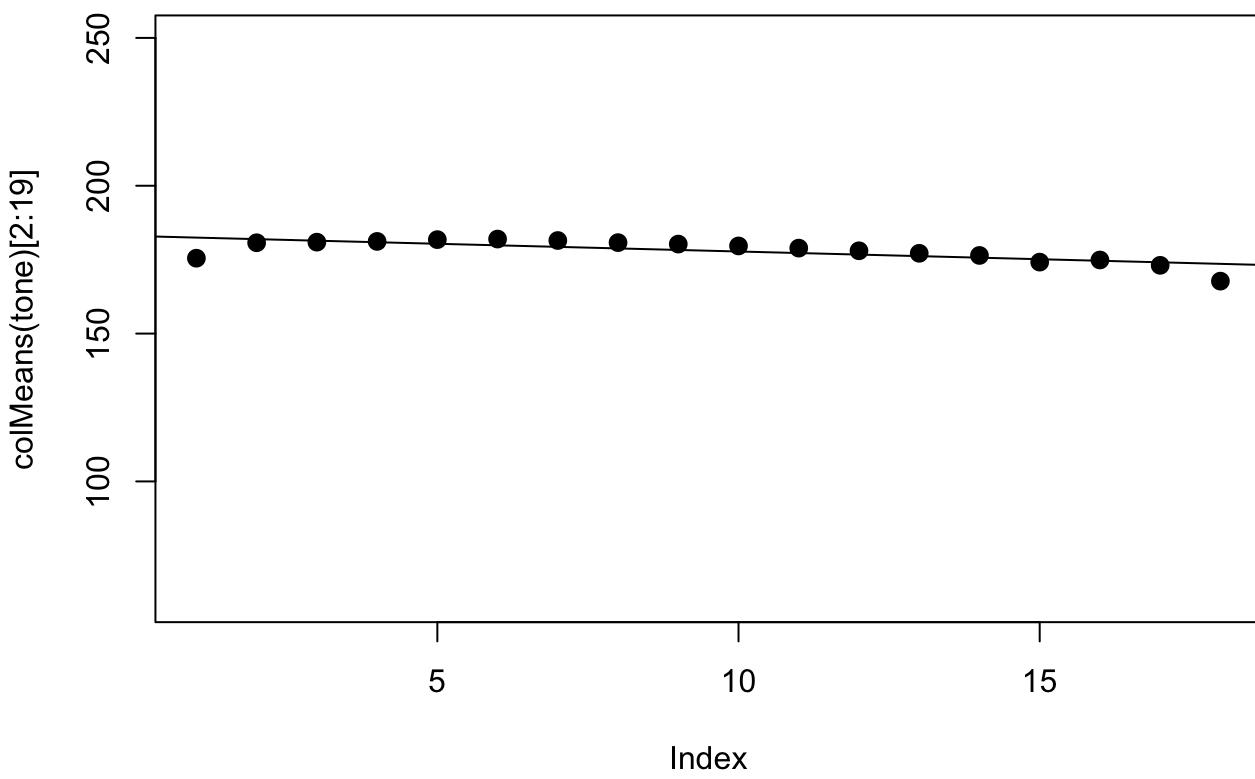
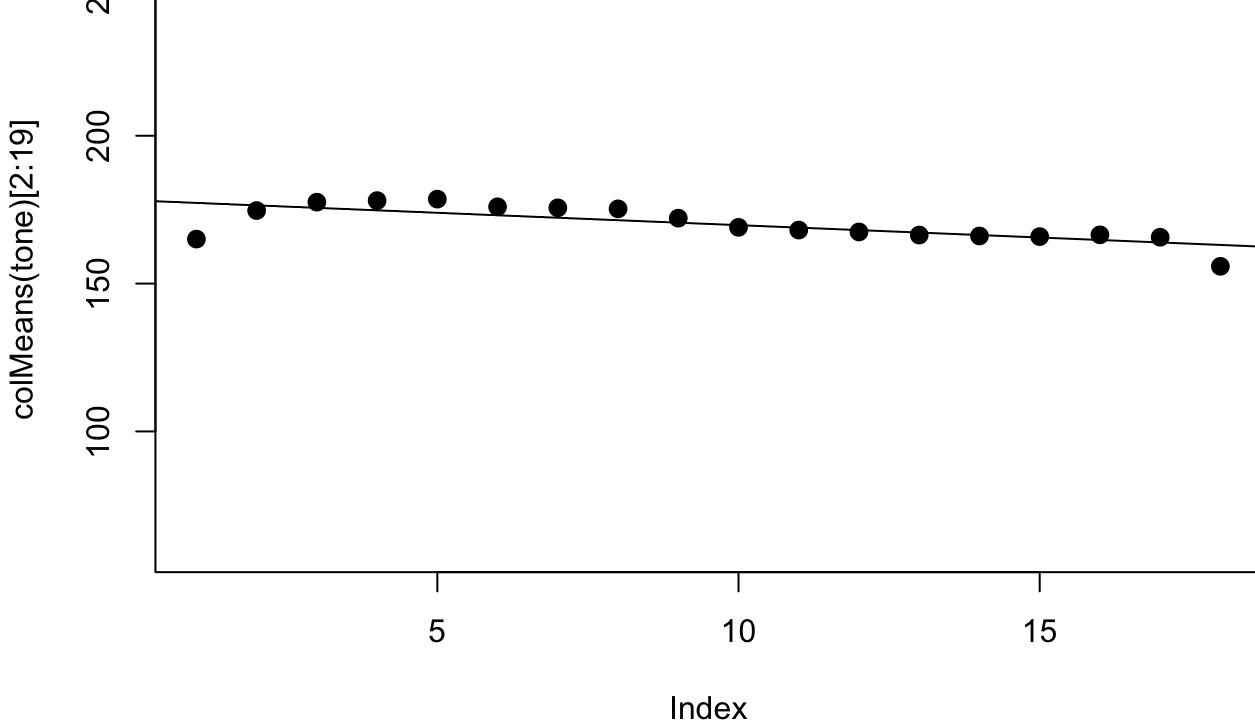
coefficient_reports <- function(tone_matrix) {
  lm_model <- lm(colMeans(tone_matrix)[2:19] ~ c(1:18))
  summary(lm_model)$coefficients[2,1]
}

```

F0 and regression line for tone A1, single tokens.

```
plot_f0_by_noise("A1", TRUE)
```

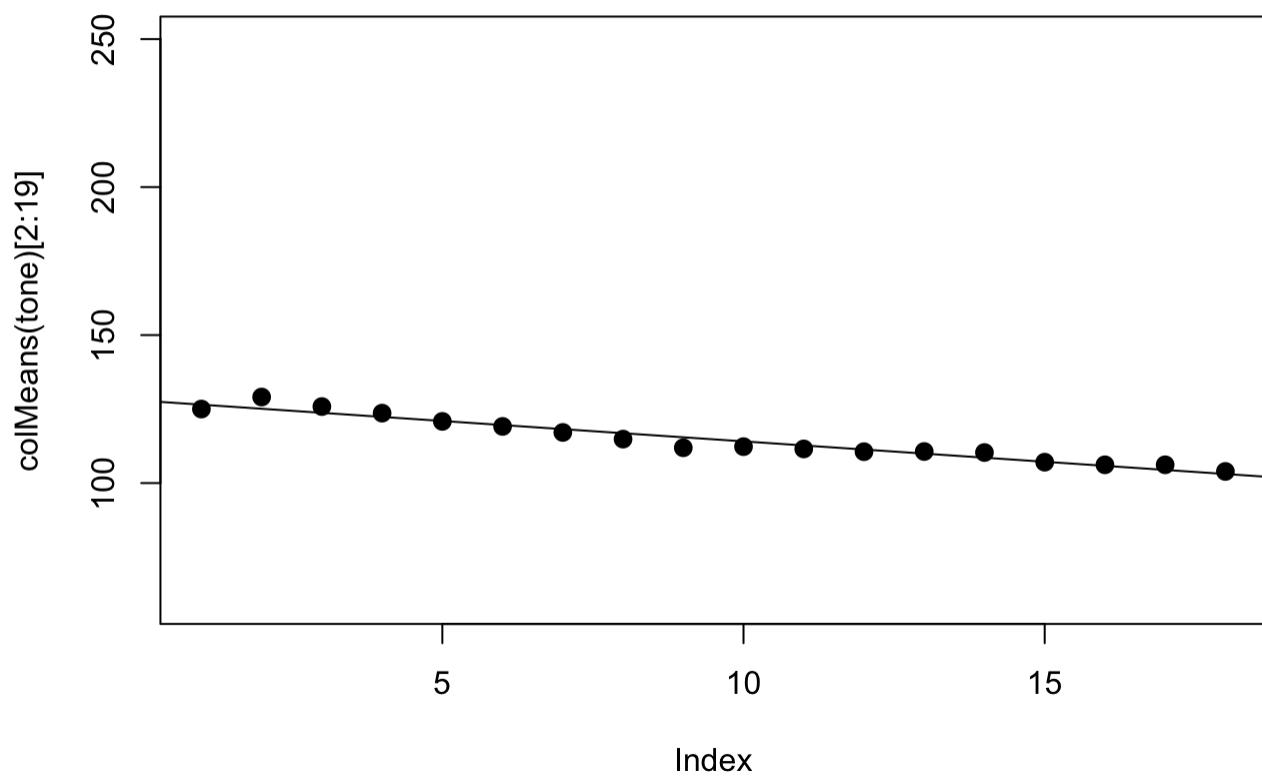
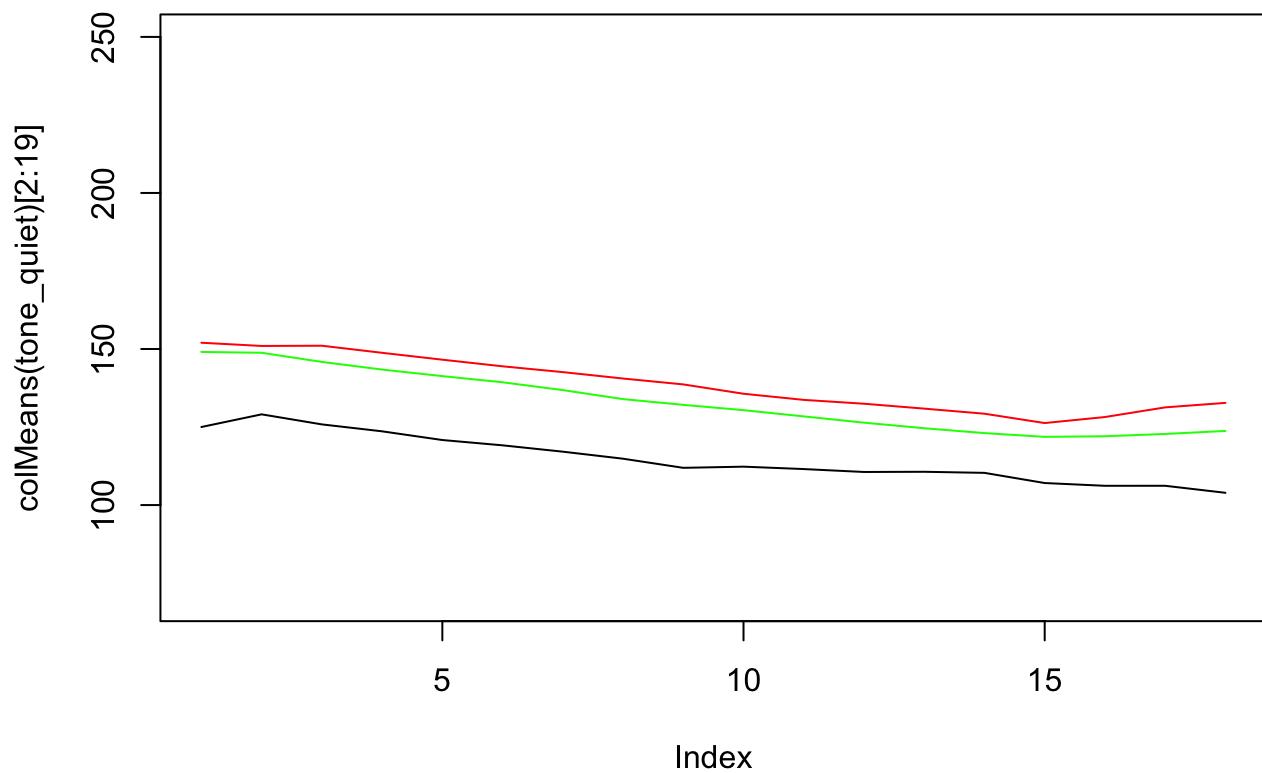



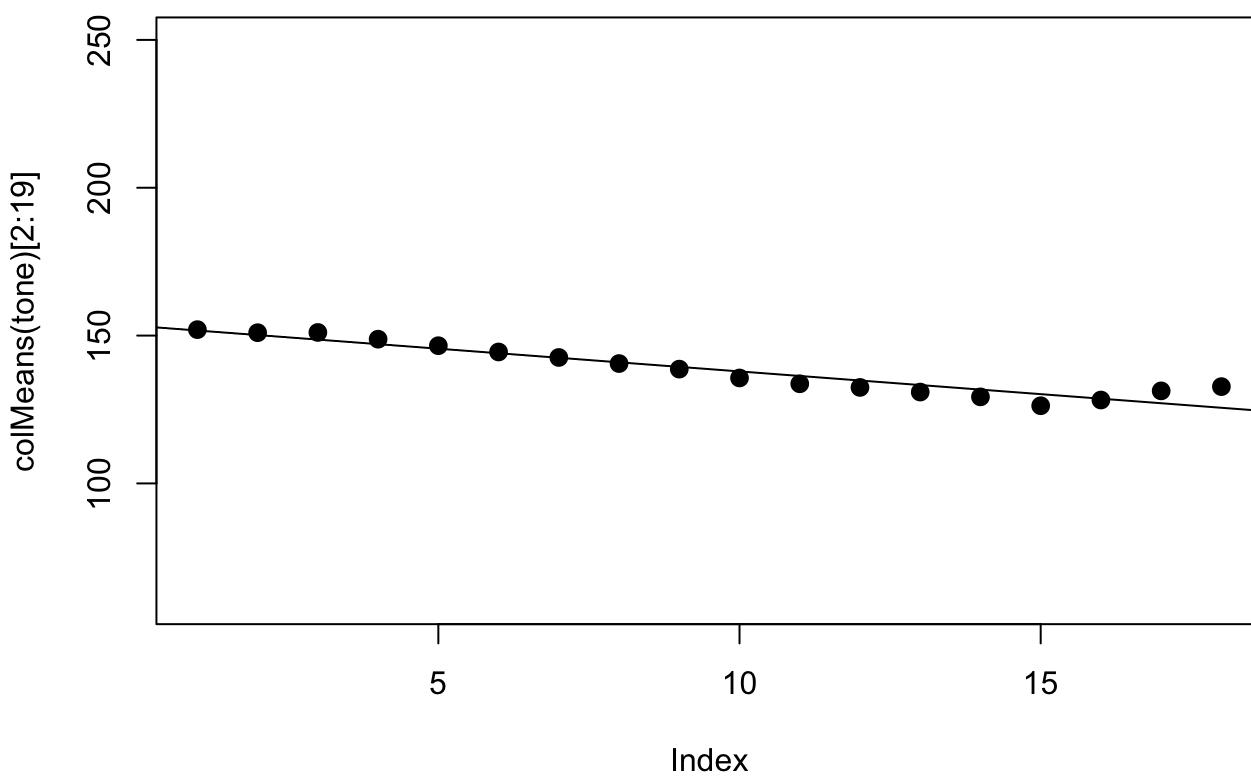
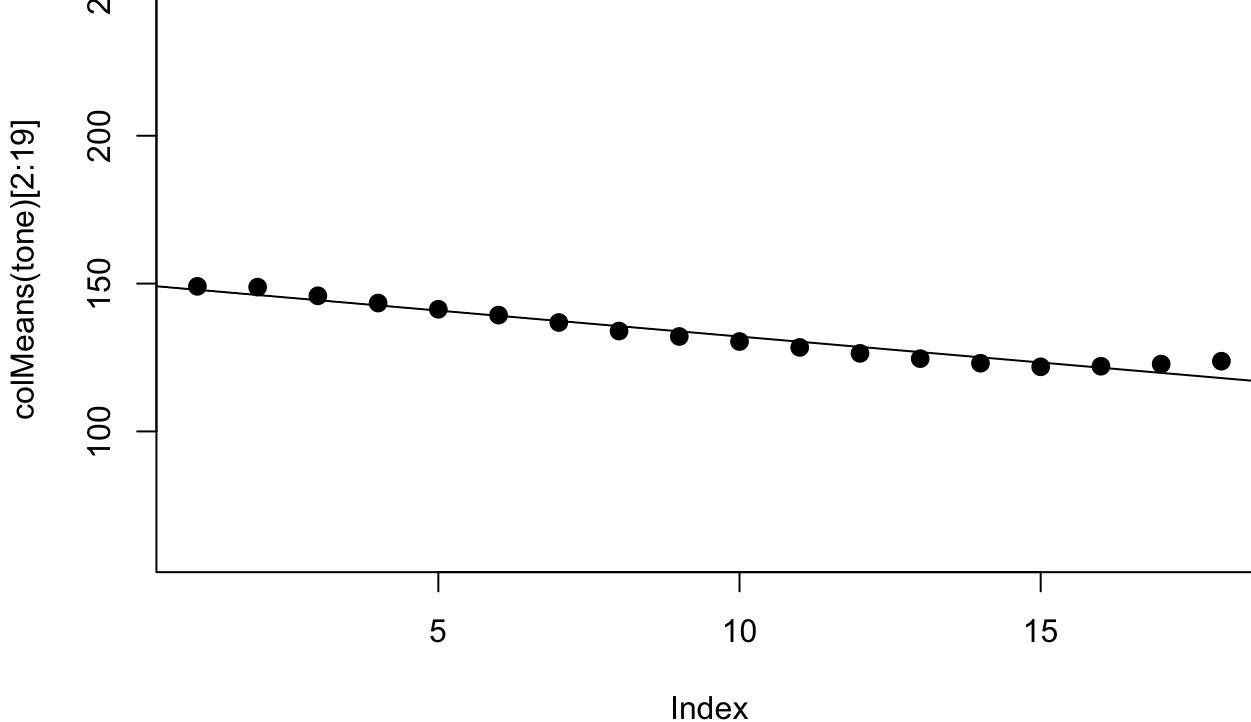


```
## [1] -0.5672417 -0.8400165 -0.5233609
```

F0 and regression line for tone A2, single tokens.

```
plot_f0_by_noise("A2", TRUE)
```

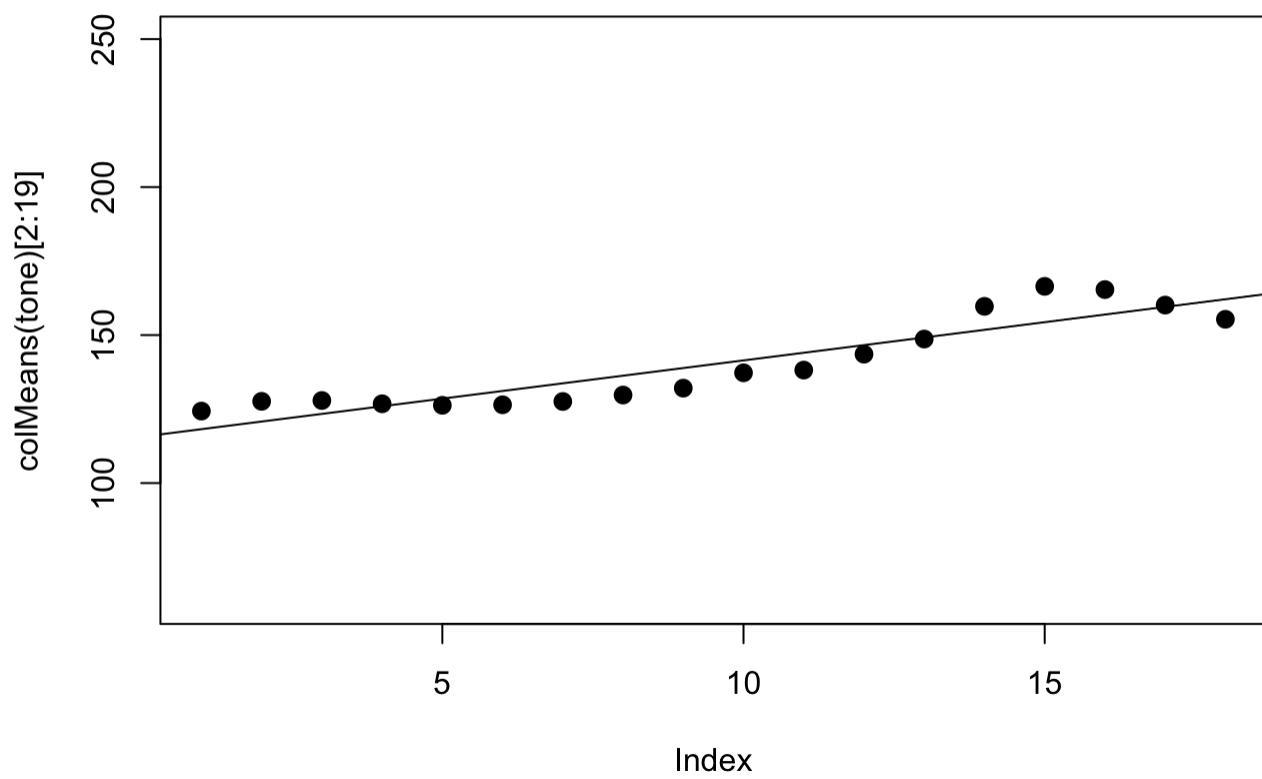
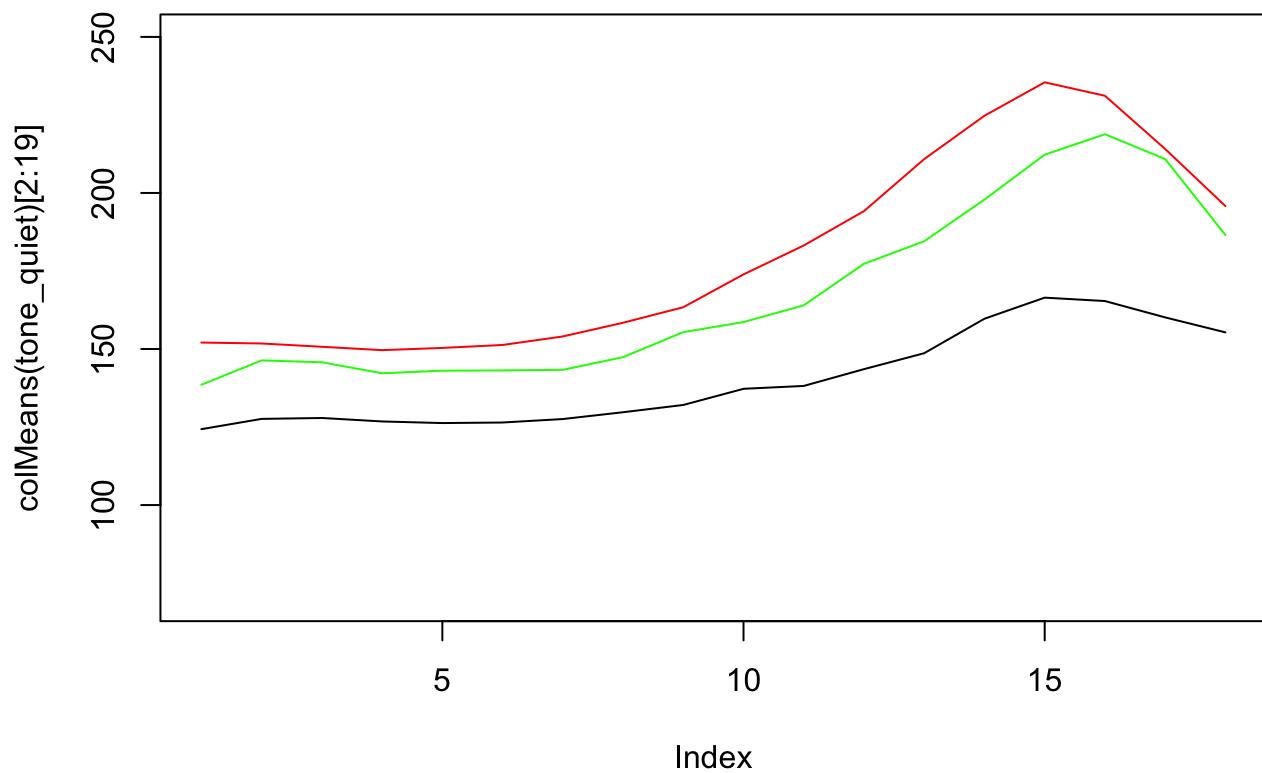



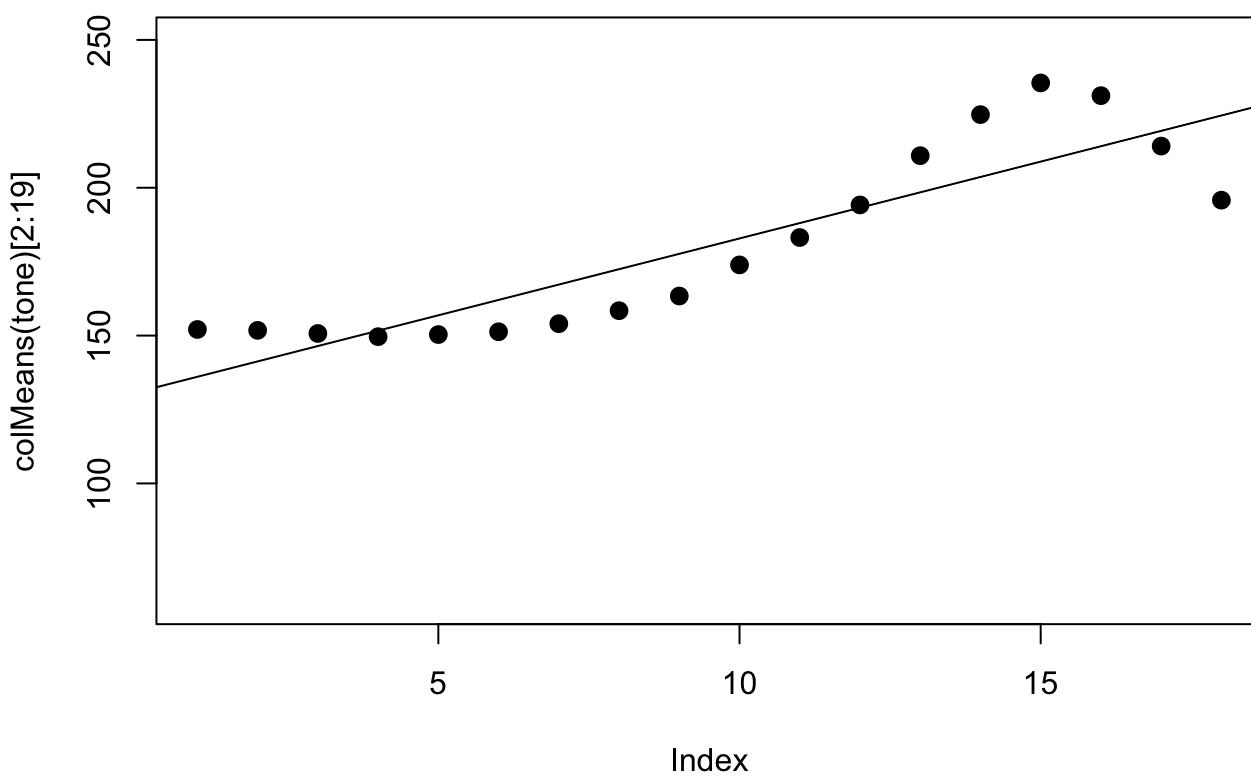
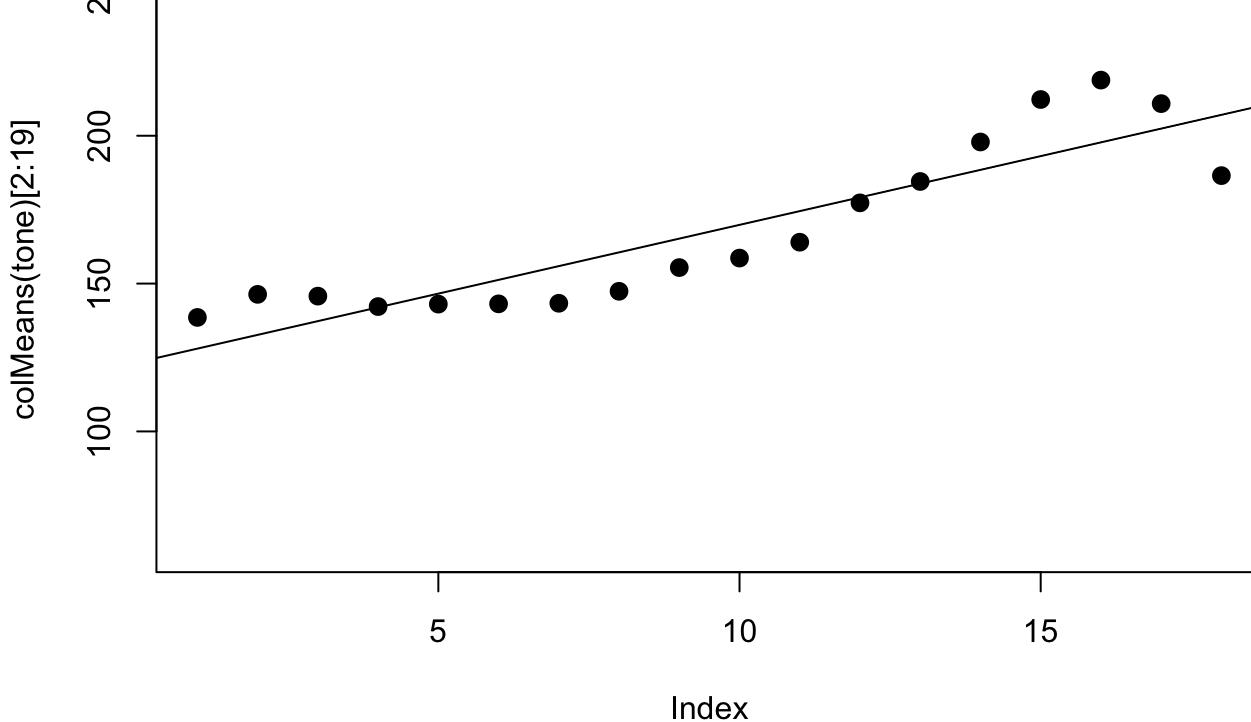


```
## [1] -1.378635 -1.759775 -1.540135
```

F0 and regression line for tone B1, single tokens.

```
plot_f0_by_noise("B1", TRUE)
```

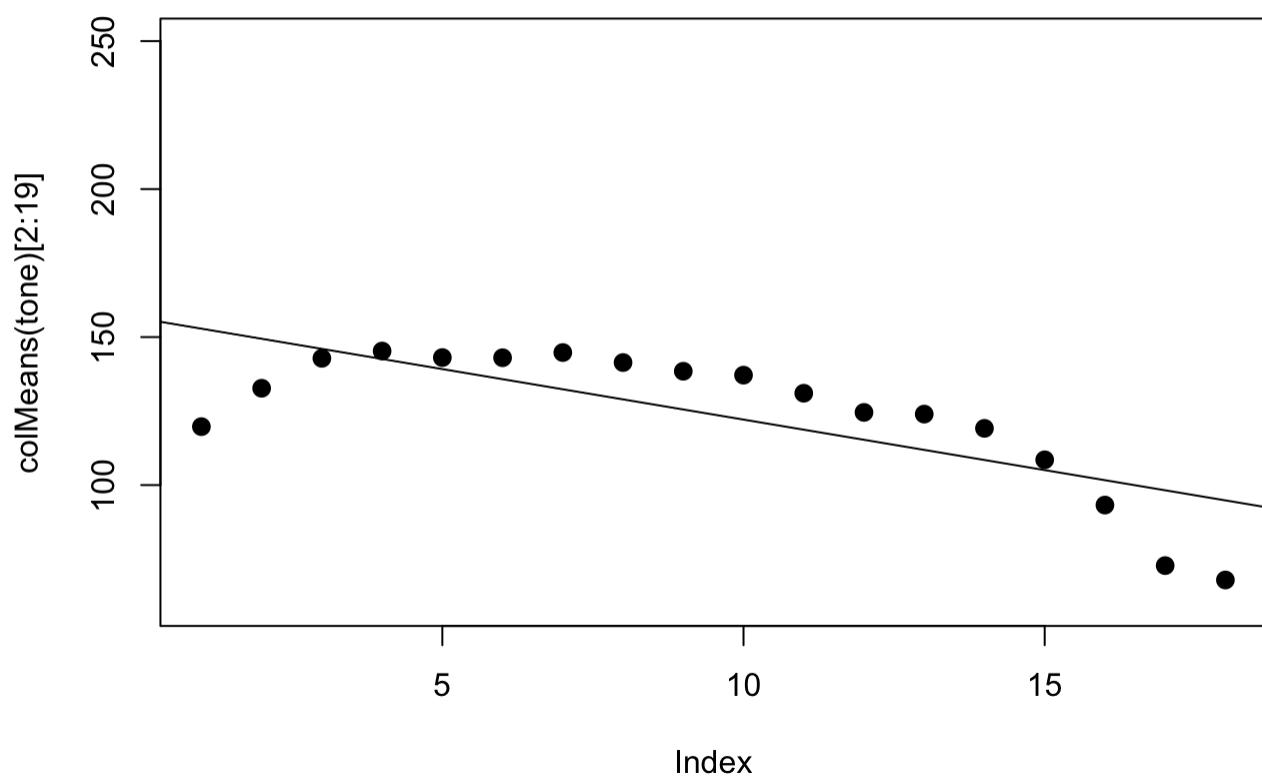
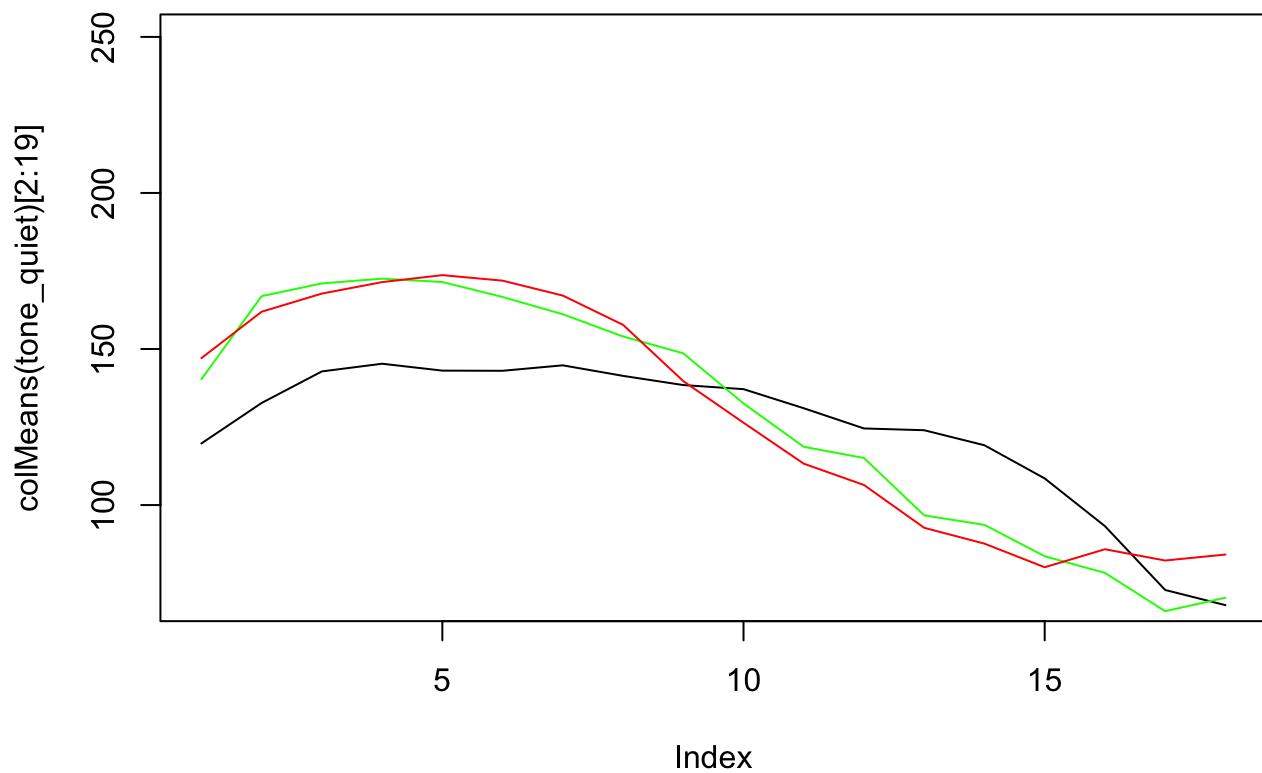



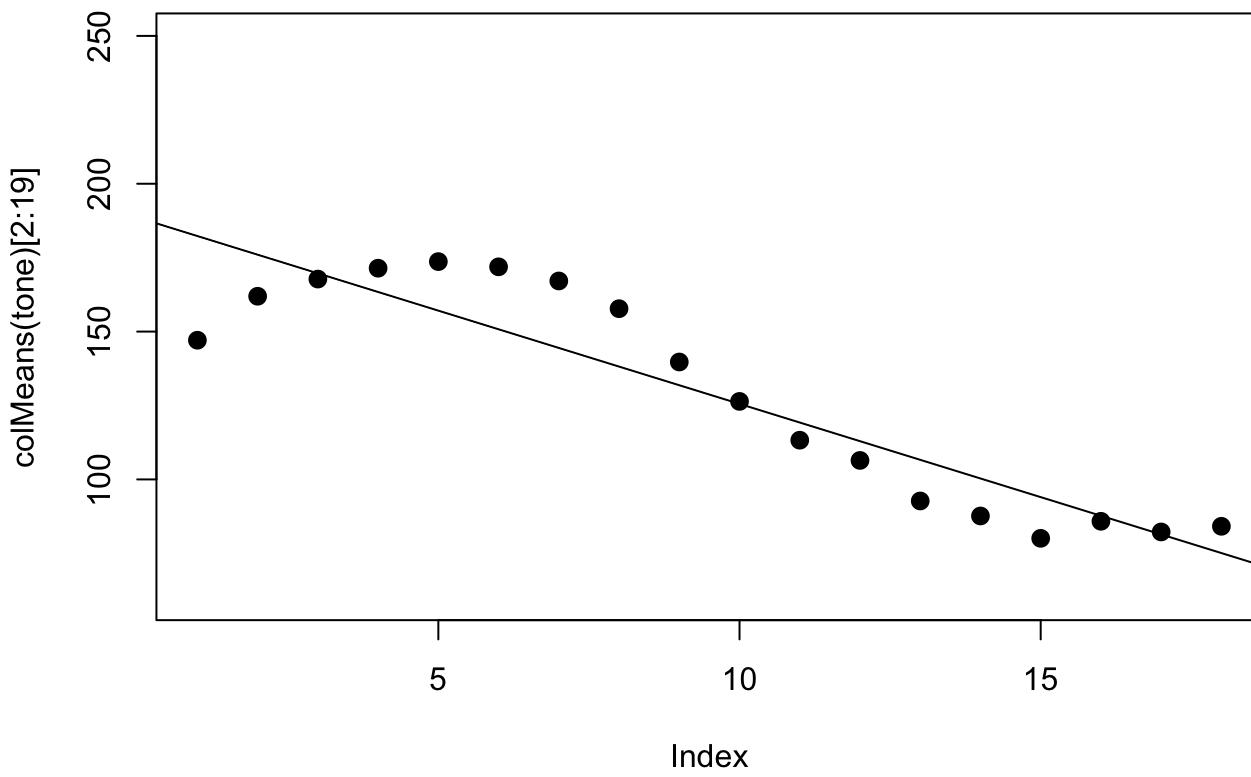
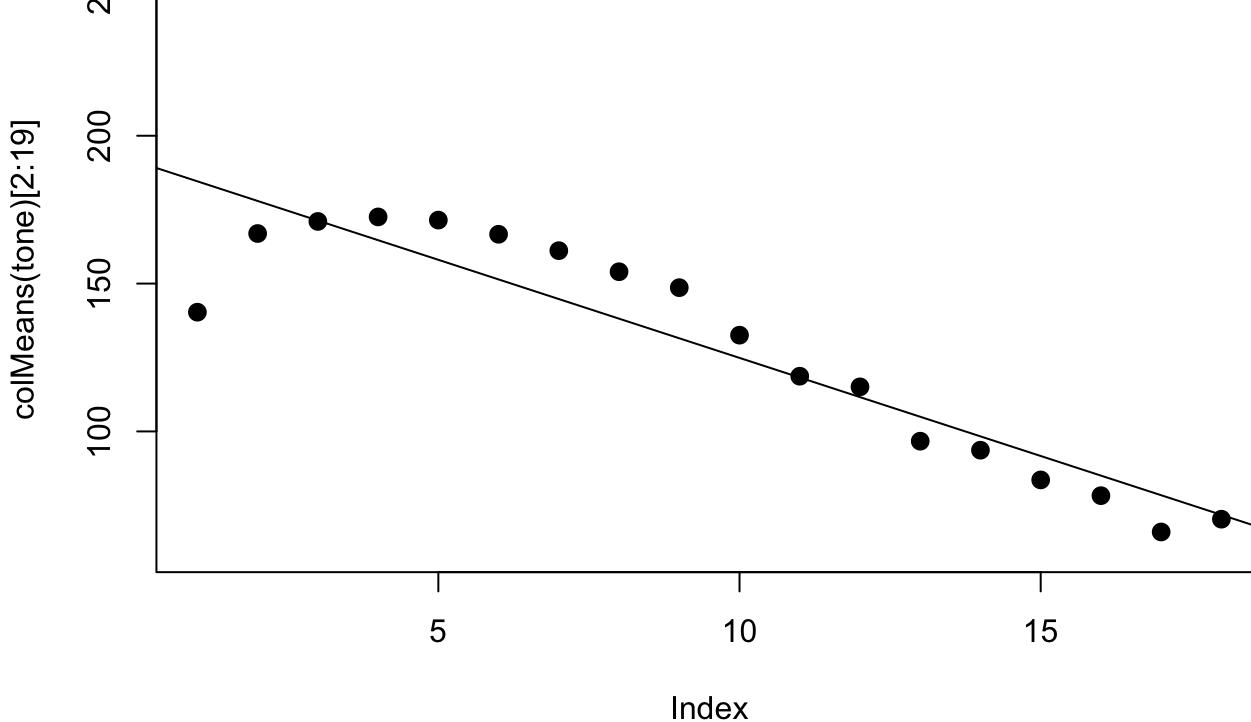


```
## [1] 2.585417 4.653510 5.201106
```

F0 and regression line for tone B2, single tokens.

```
plot_f0_by_noise("B2", TRUE)
```

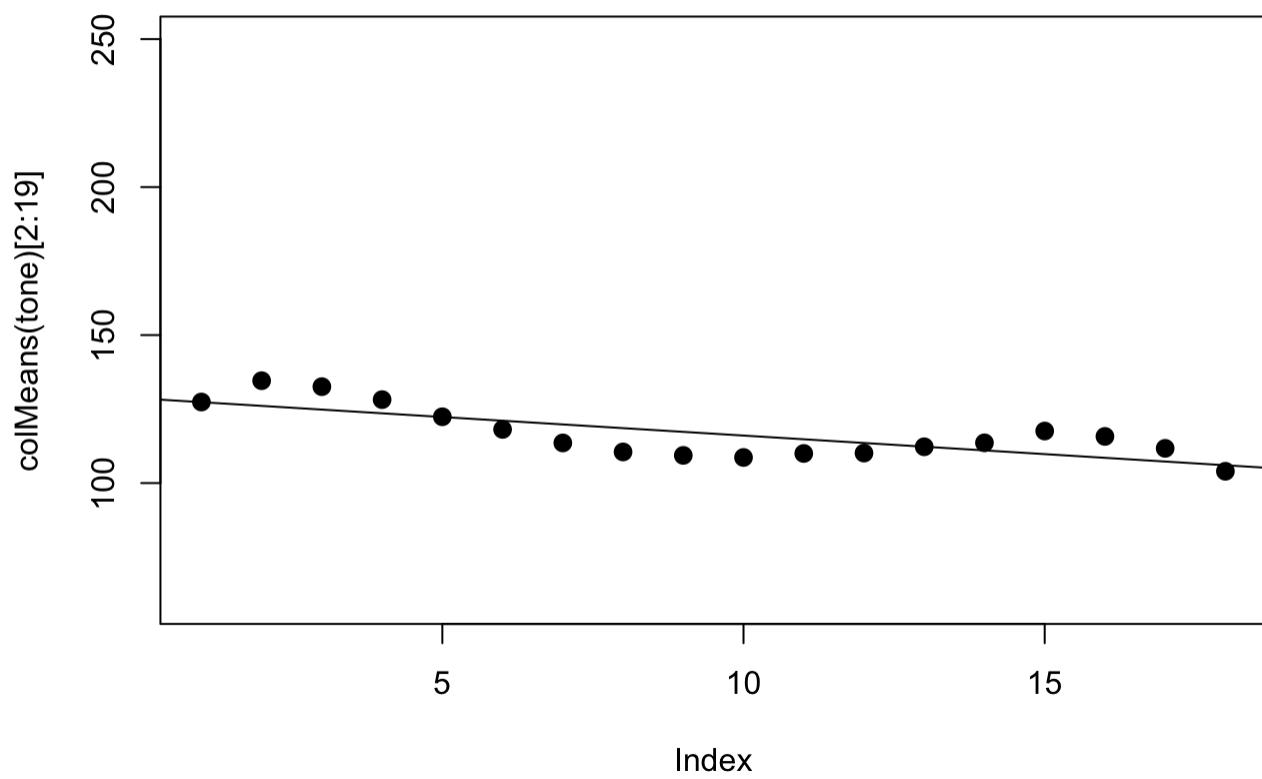
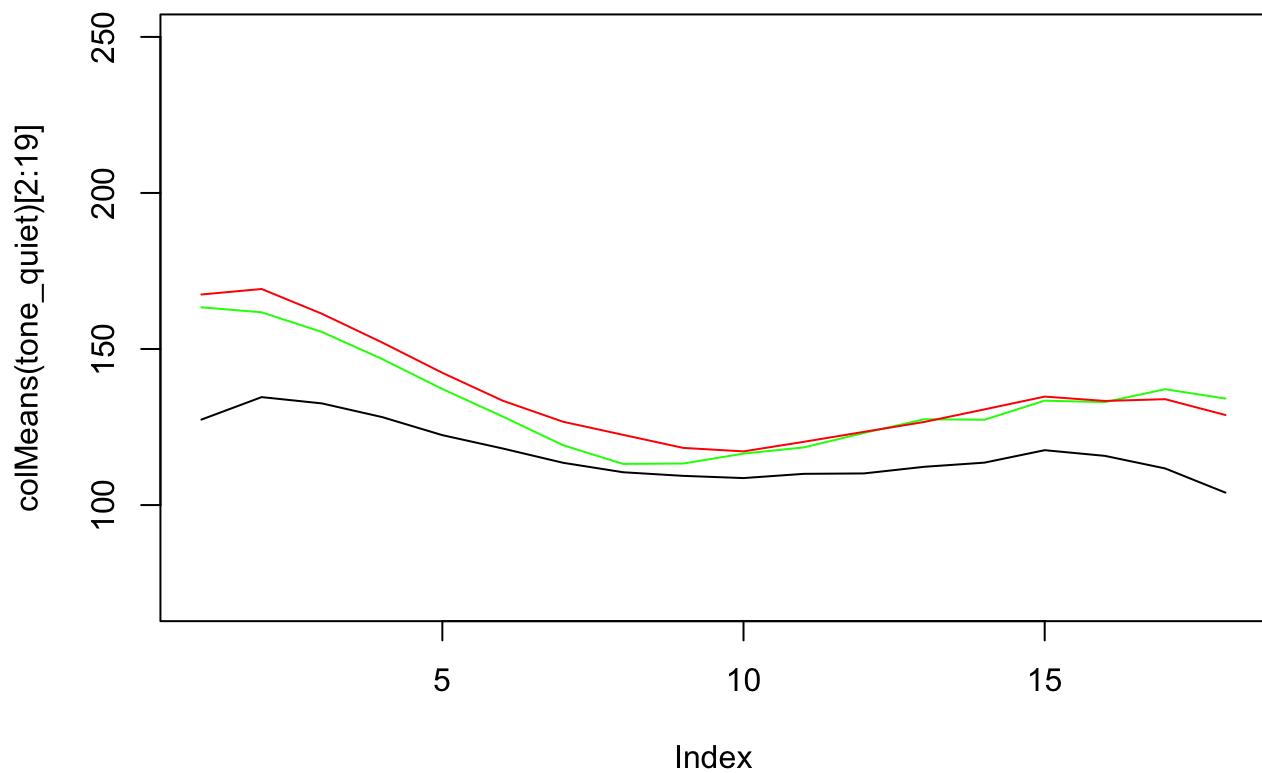



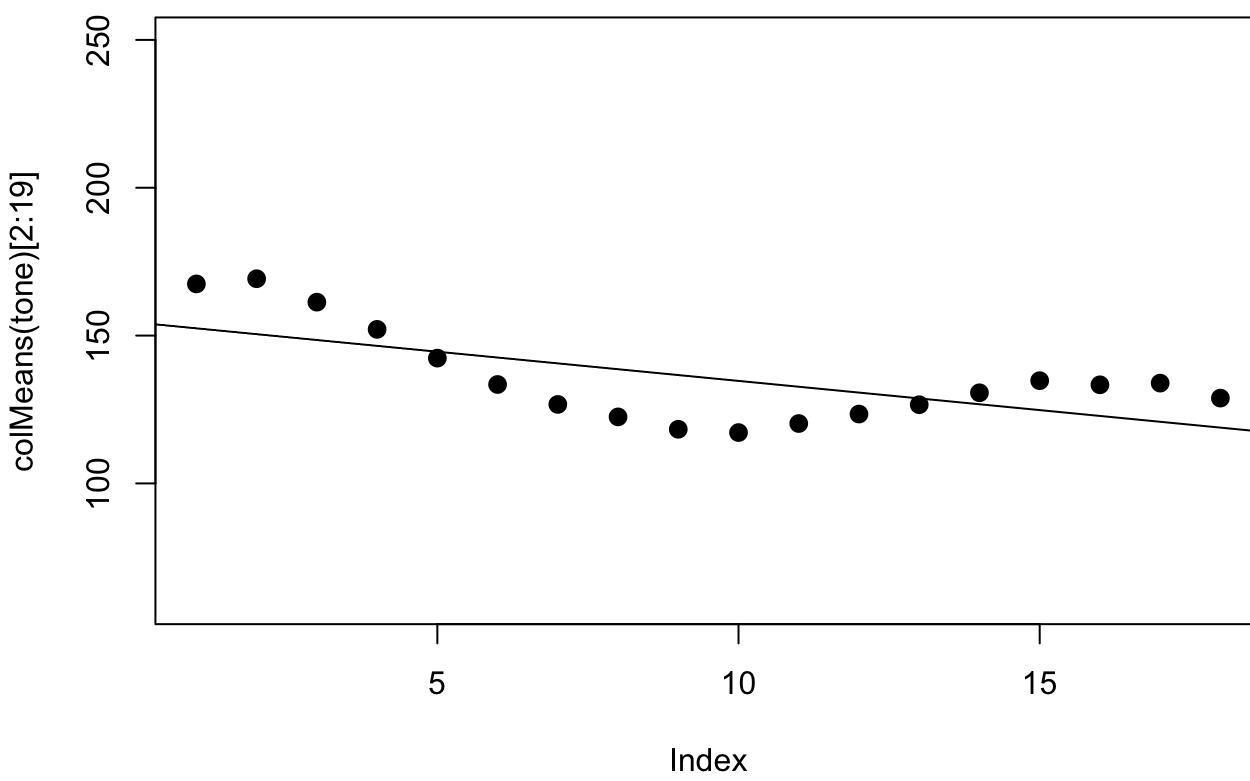
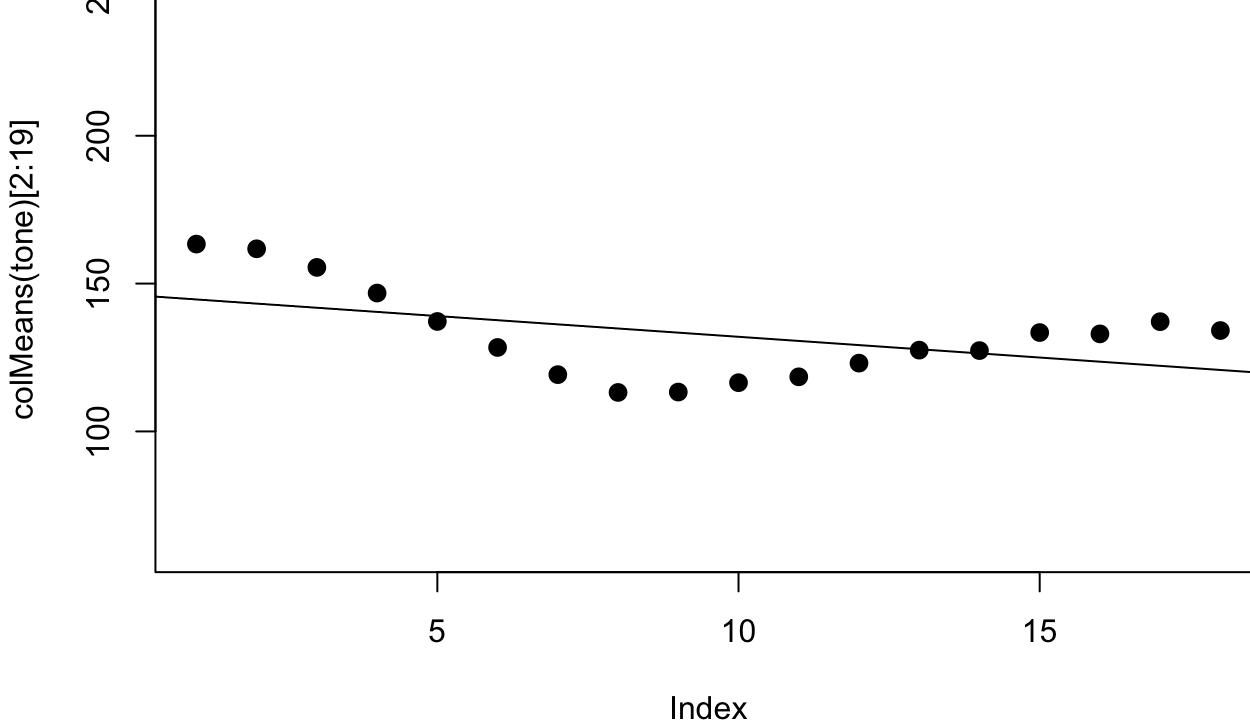


```
## [1] -3.416345 -6.636740 -6.309881
```

F0 and regression line for tone C1, single tokens.

```
plot_f0_by_noise("C1", TRUE)
```

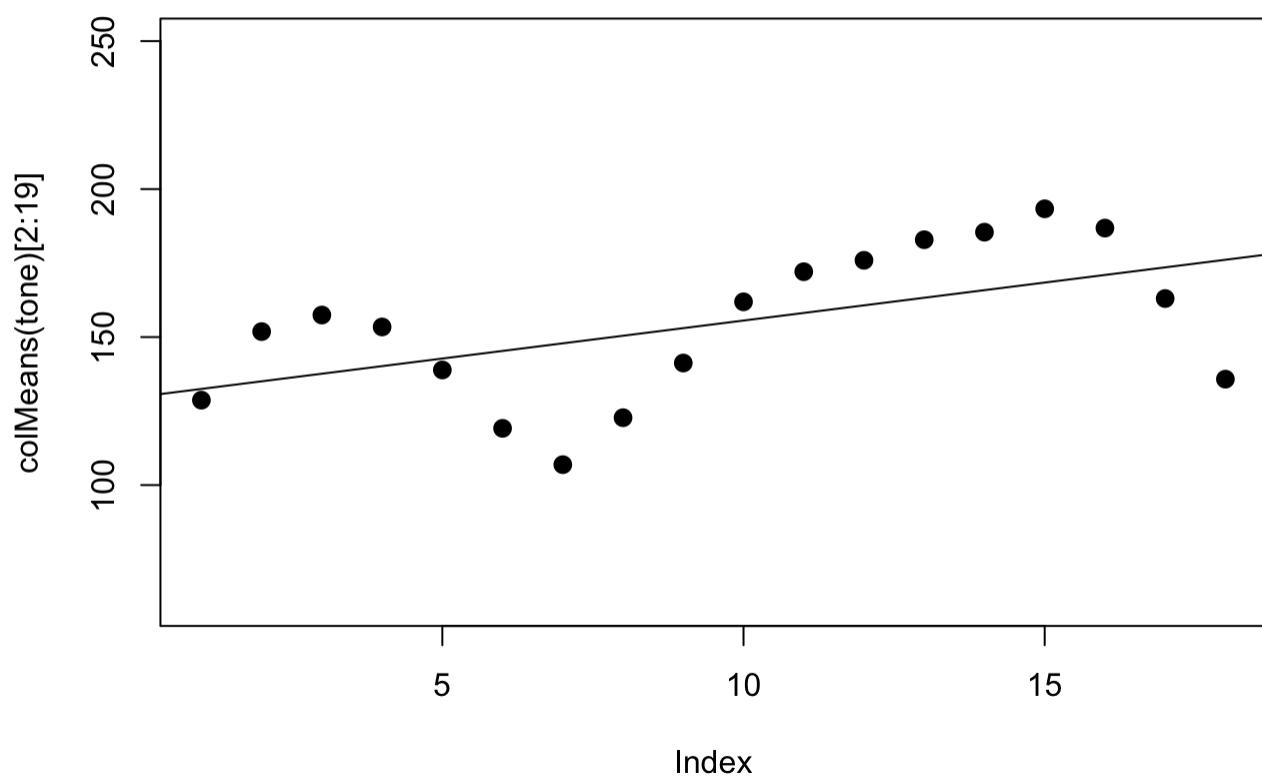
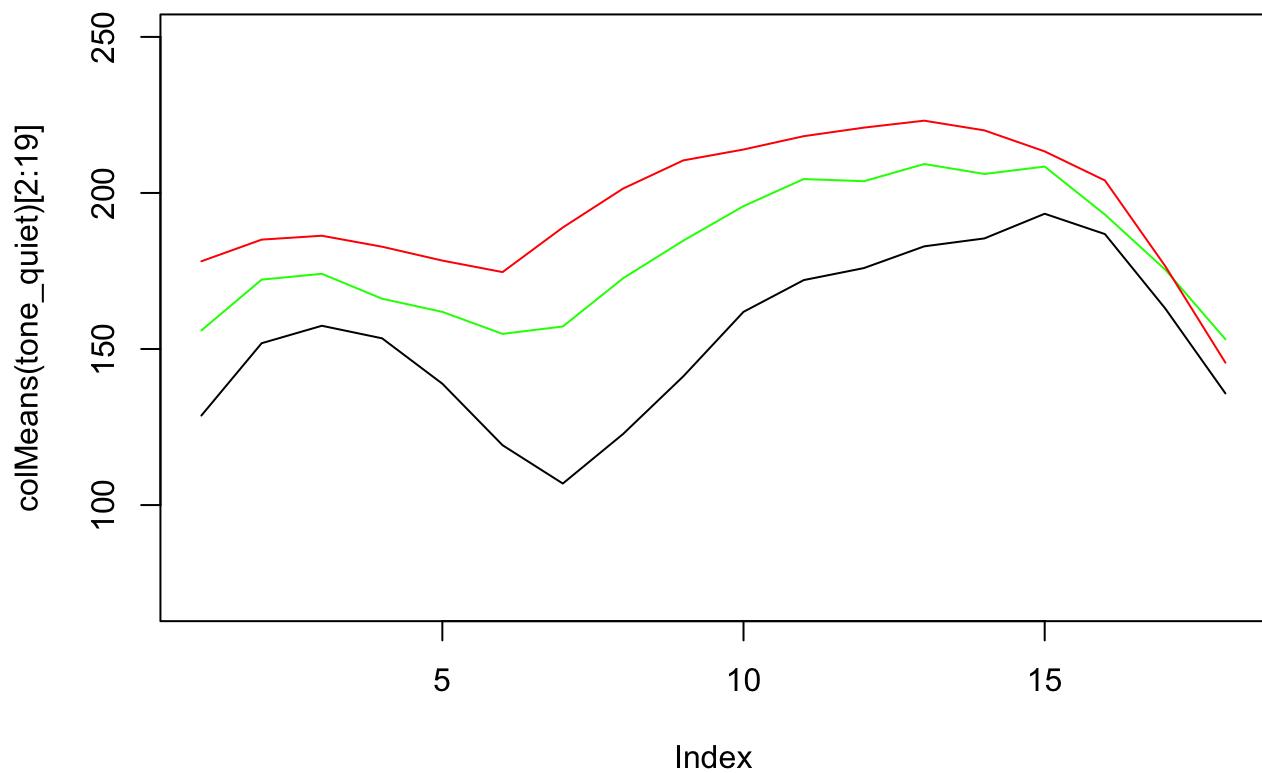



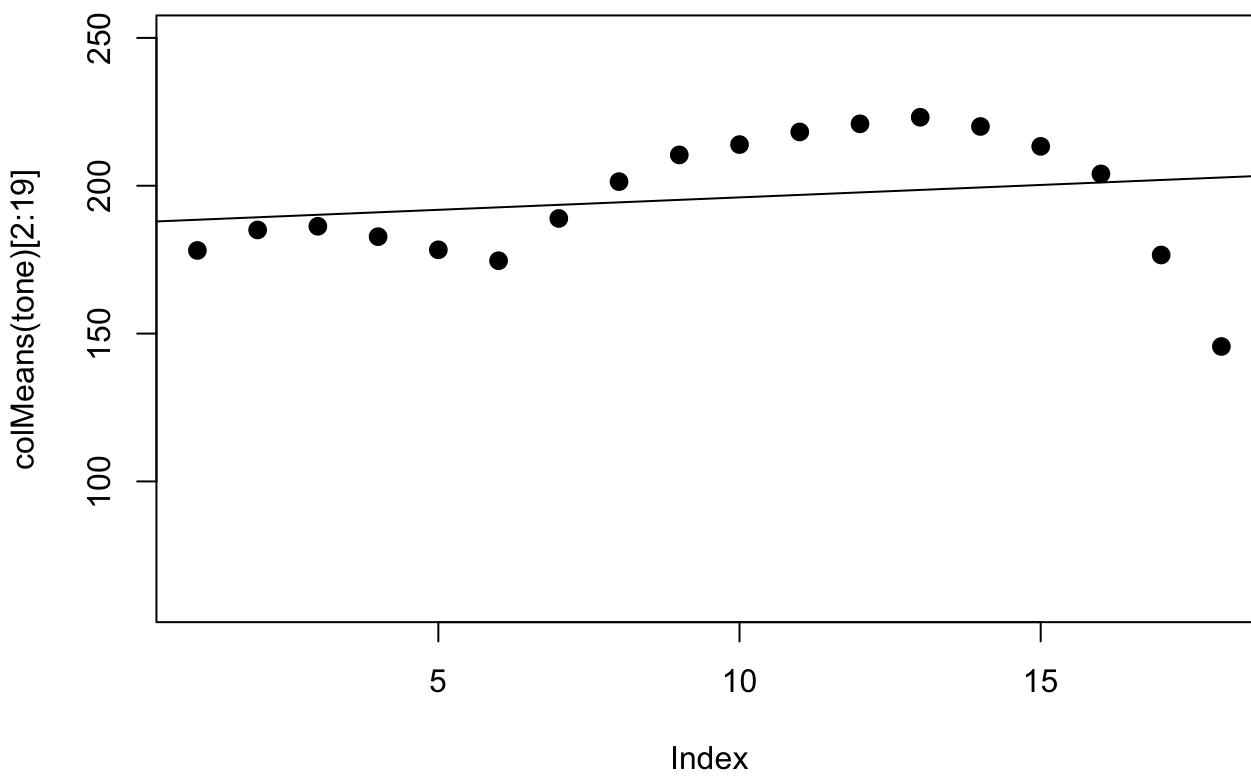
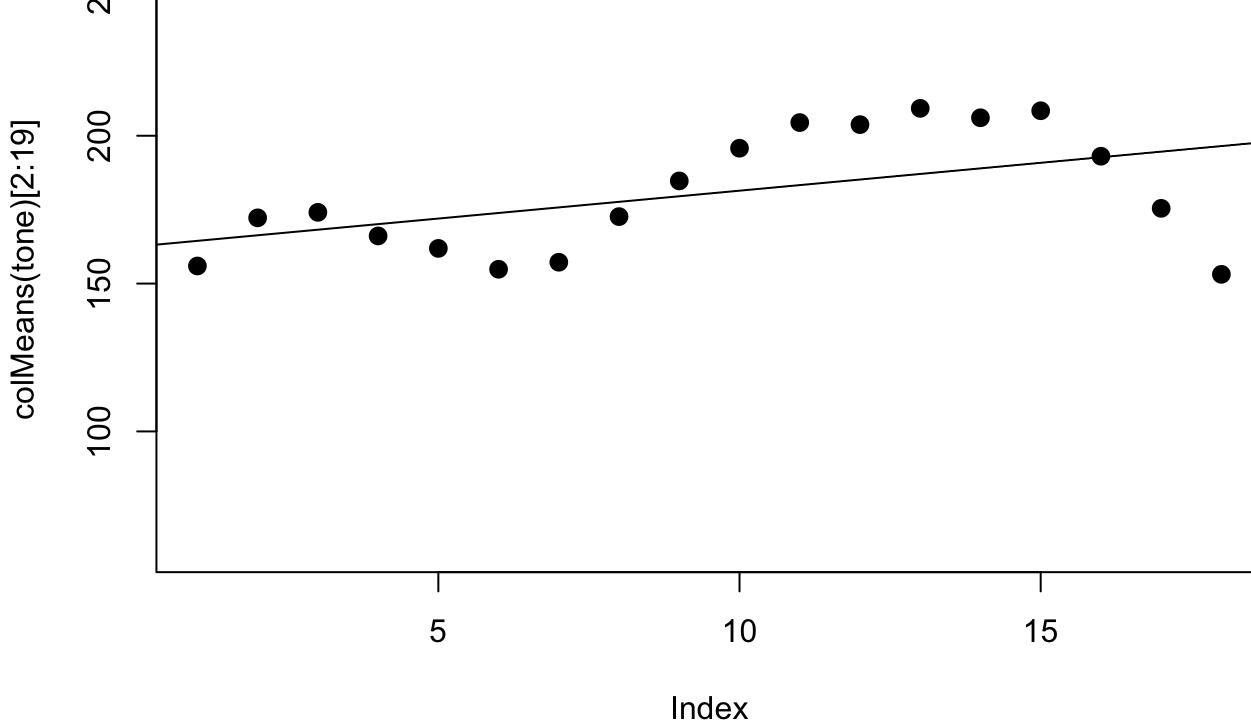


```
## [1] -1.254551 -1.406069 -1.978960
```

###F0 and regression line for tone C2, single tokens.

```
plot_f0_by_noise("C2", TRUE)
```

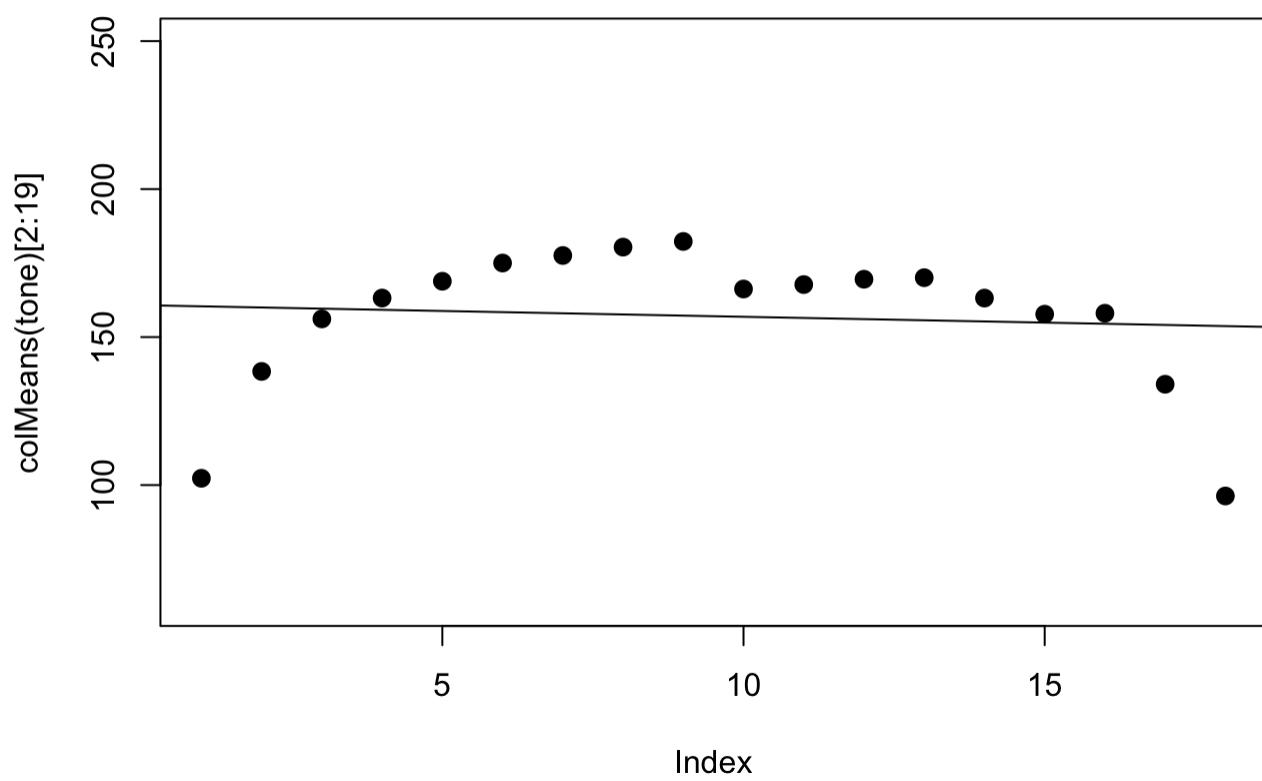
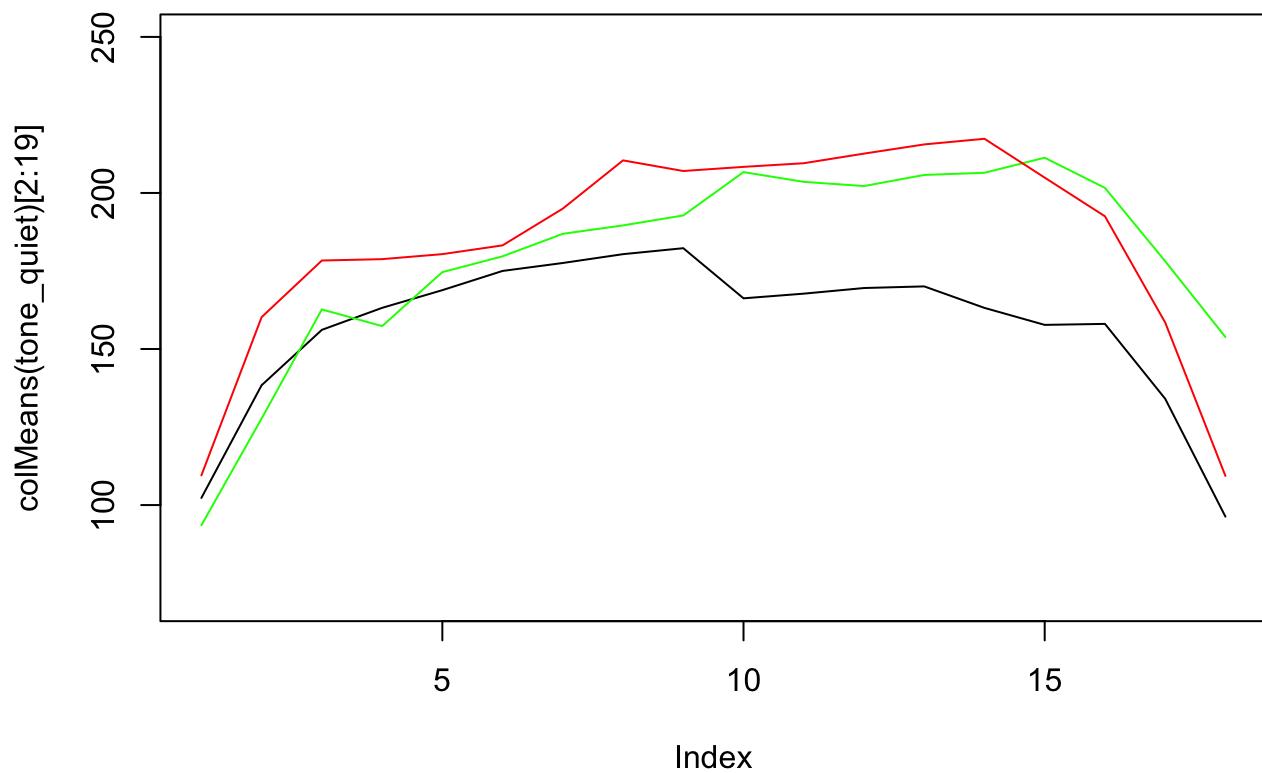



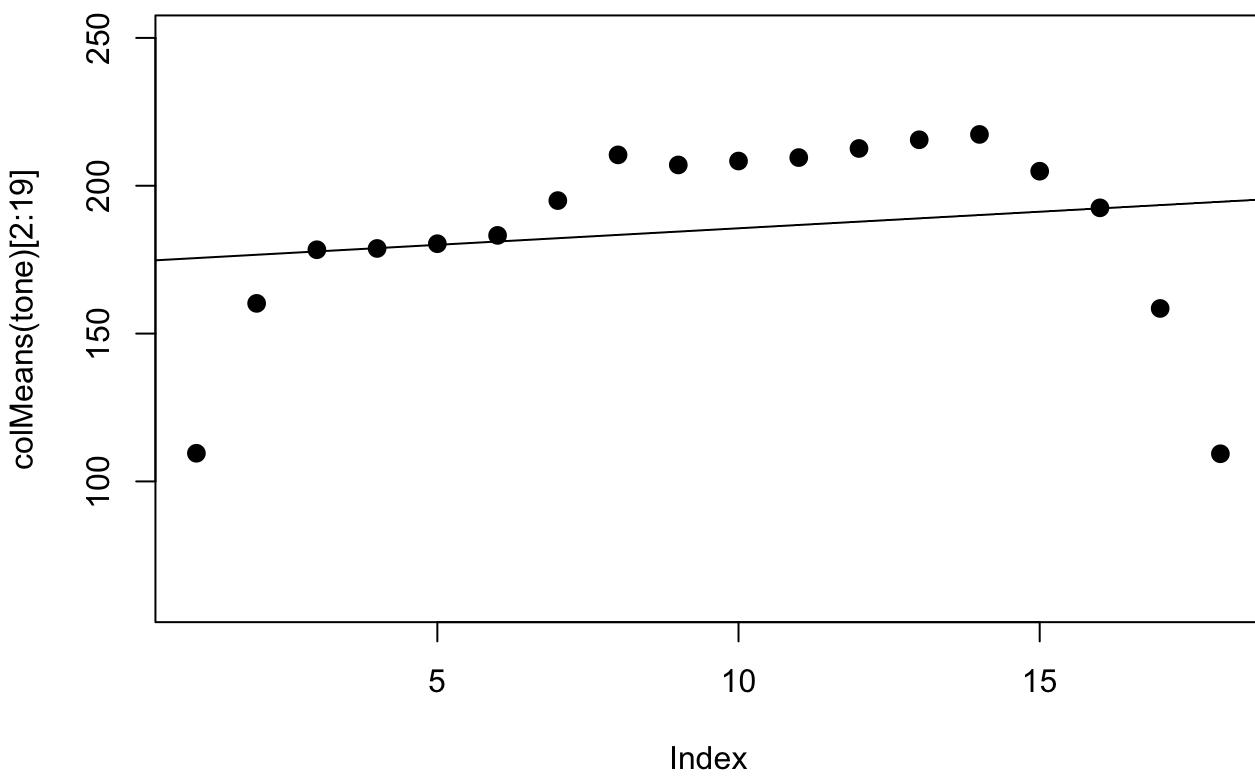
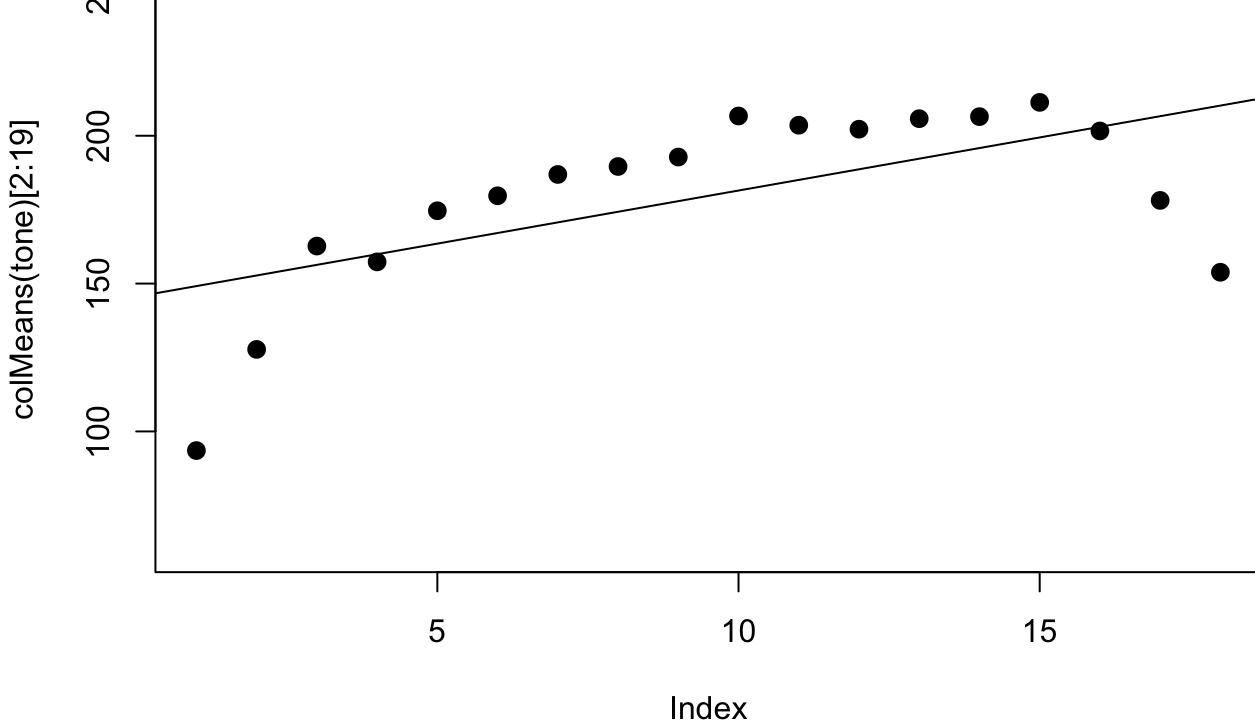


```
## [1] 2.5676060 1.8890026 0.8420454
```

F0 and regression line for tone D1, single tokens.

```
plot_f0_by_noise("D1", TRUE)
```

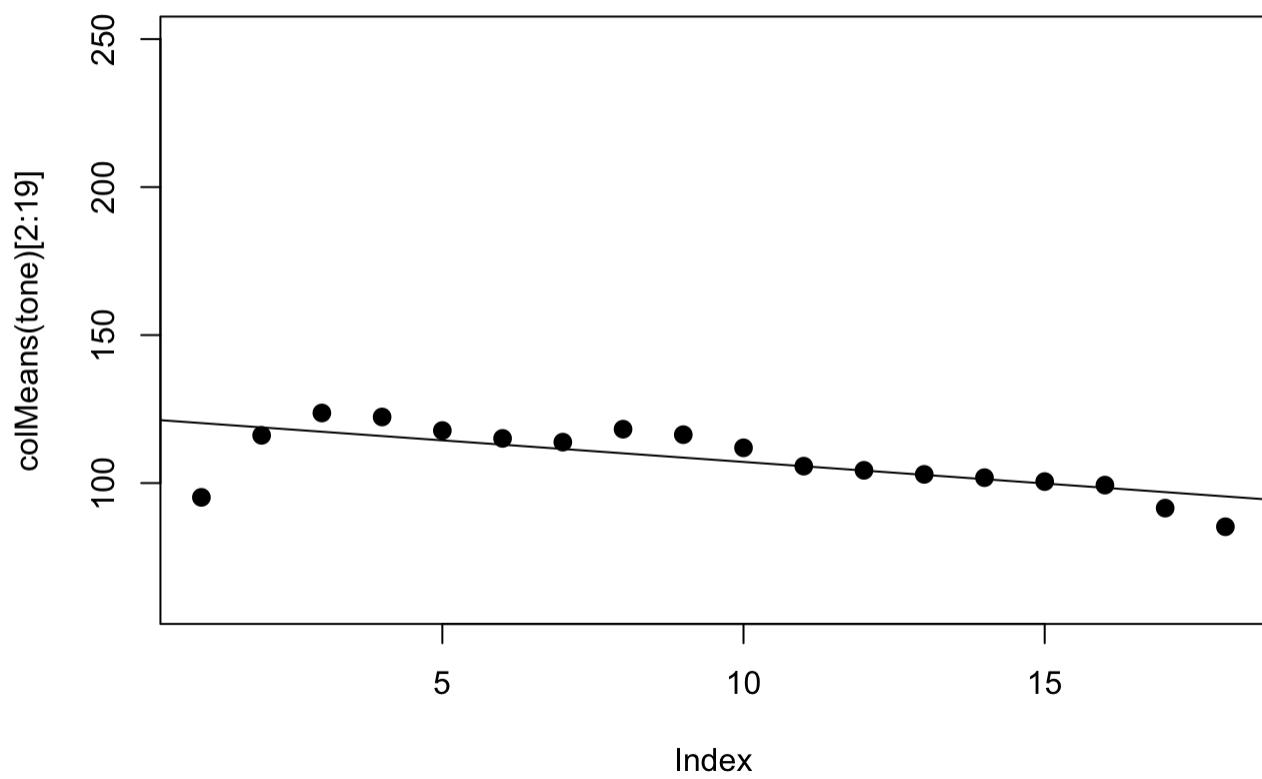
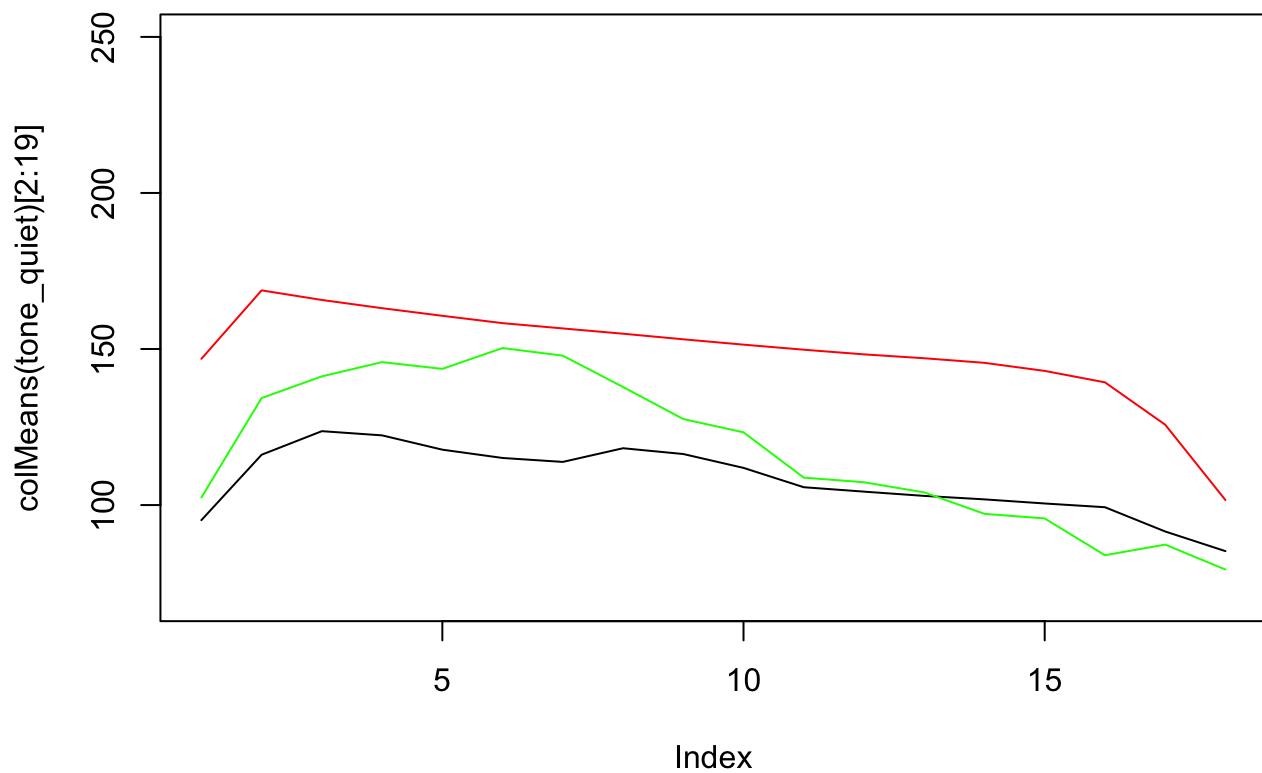



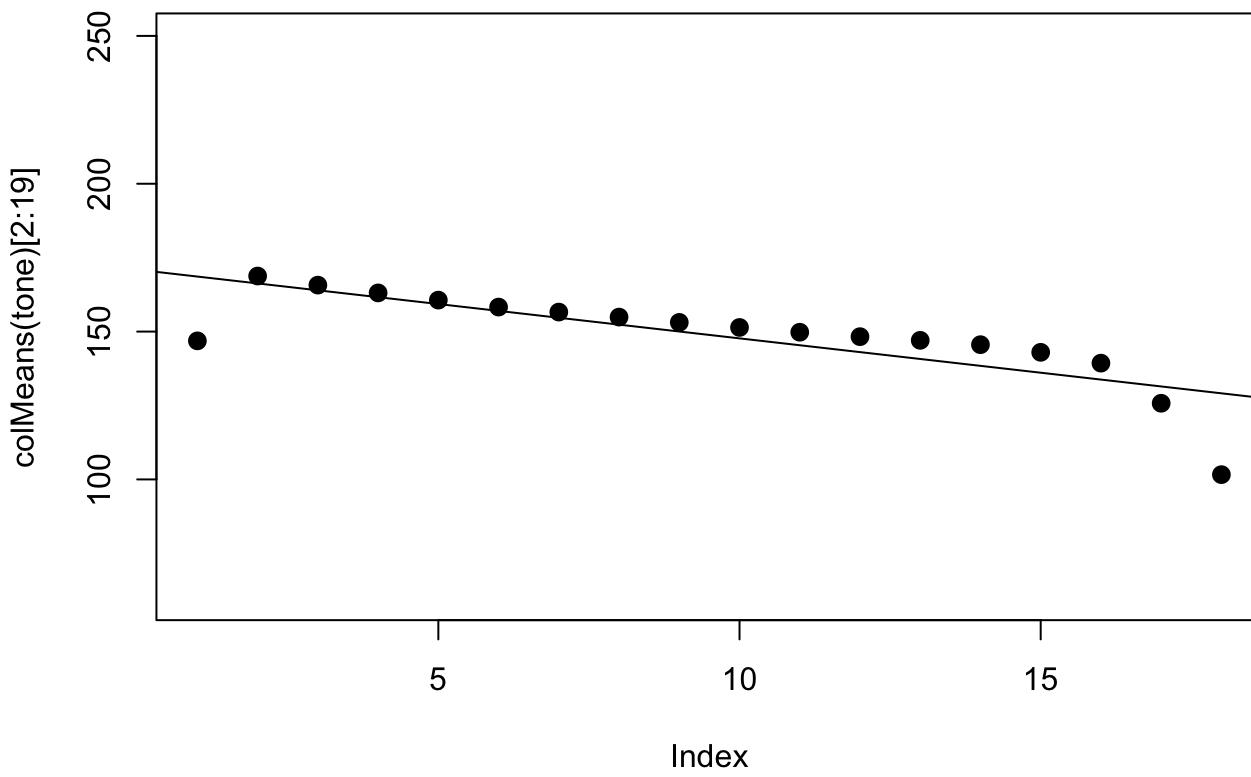
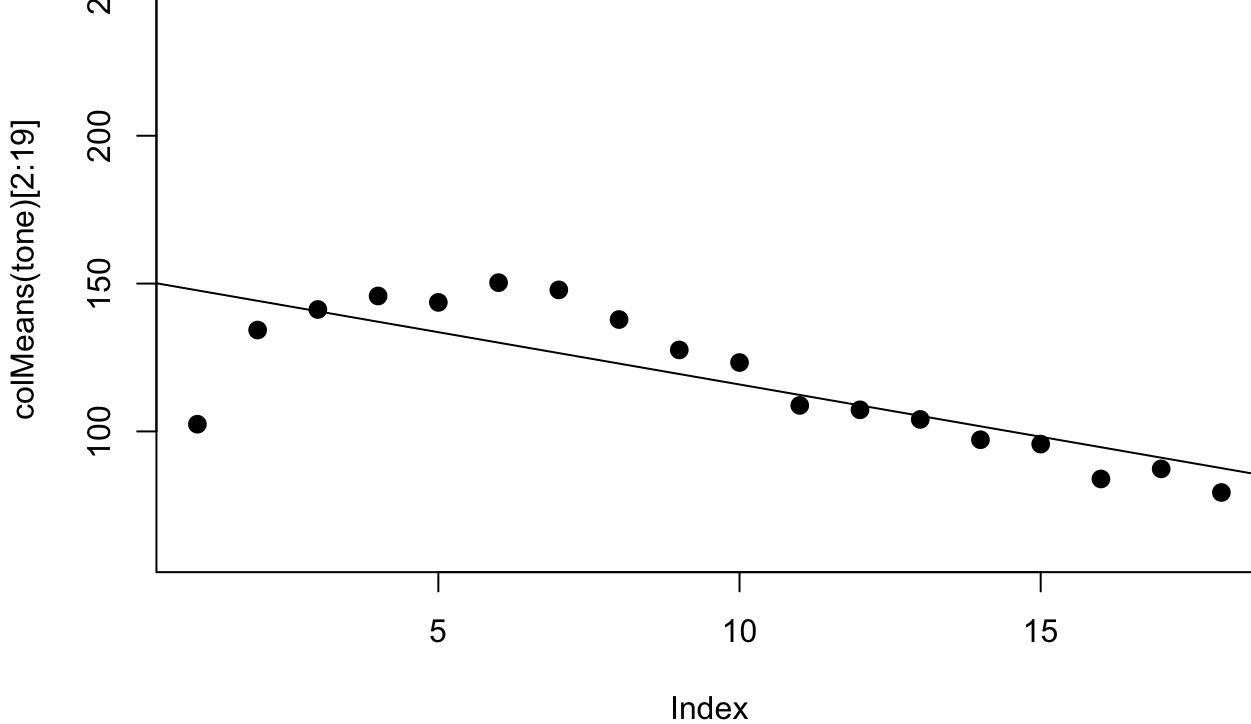


```
## [1] -0.3939798 3.5934800 1.1236231
```

F0 and regression line for tone D2, single tokens.

```
plot_f0_by_noise("D2", TRUE)
```

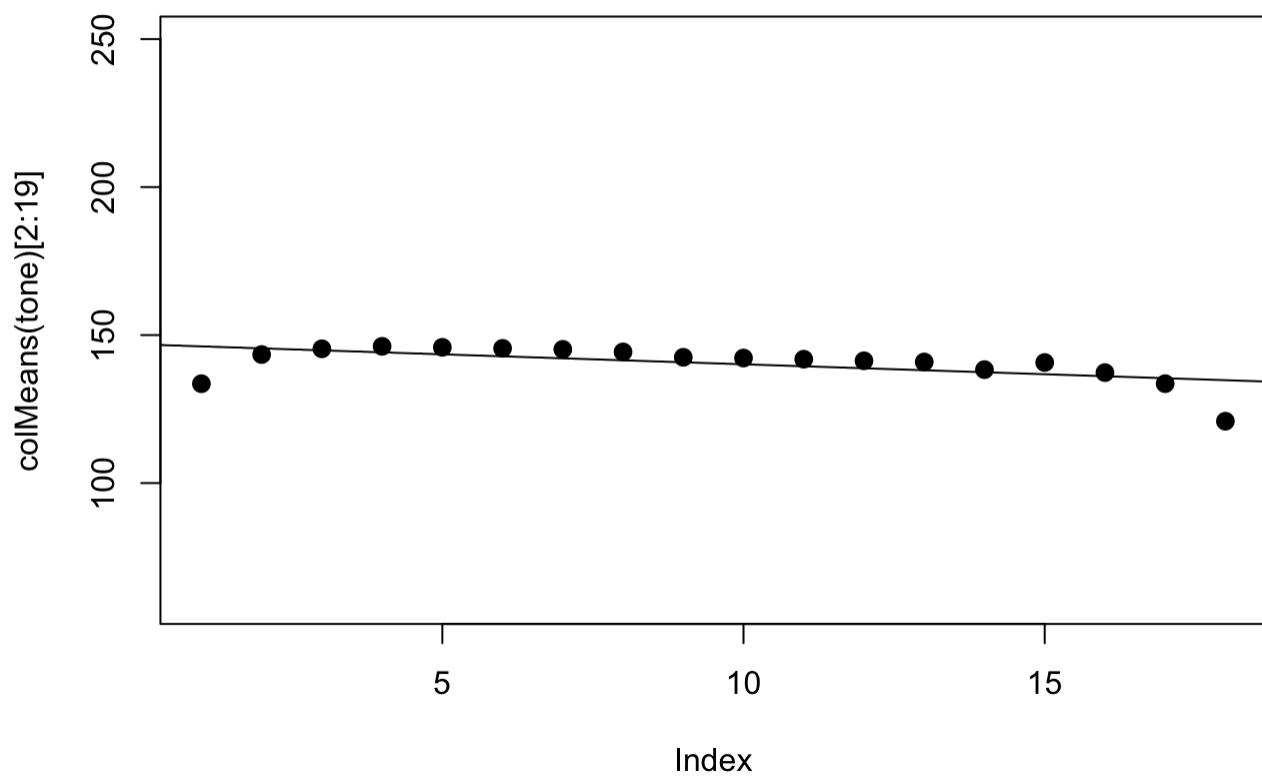
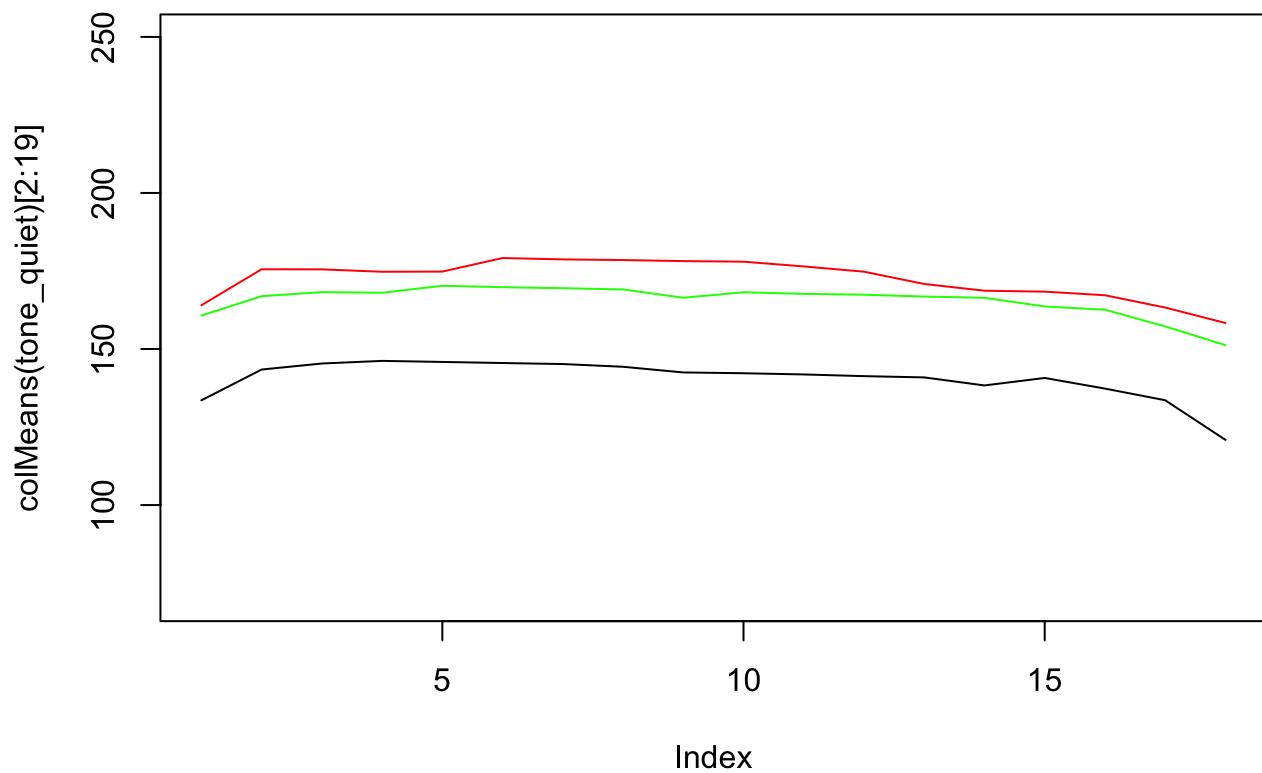



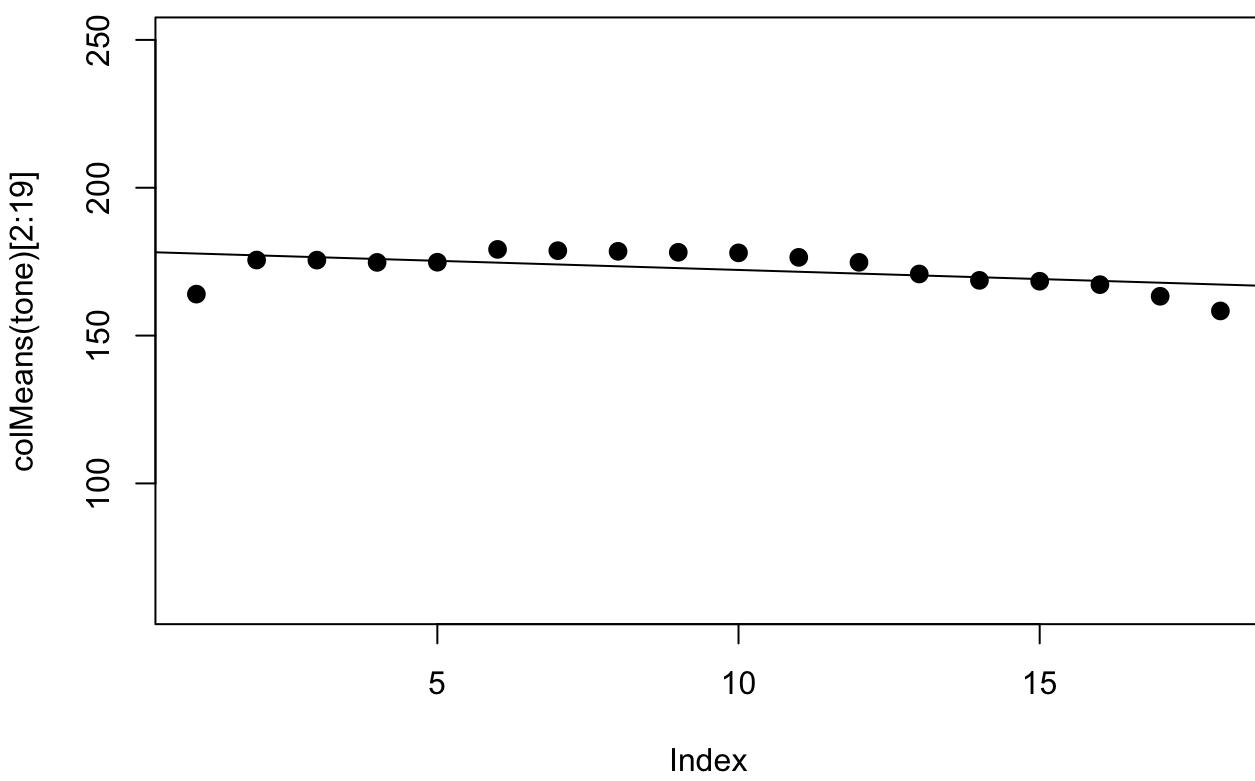
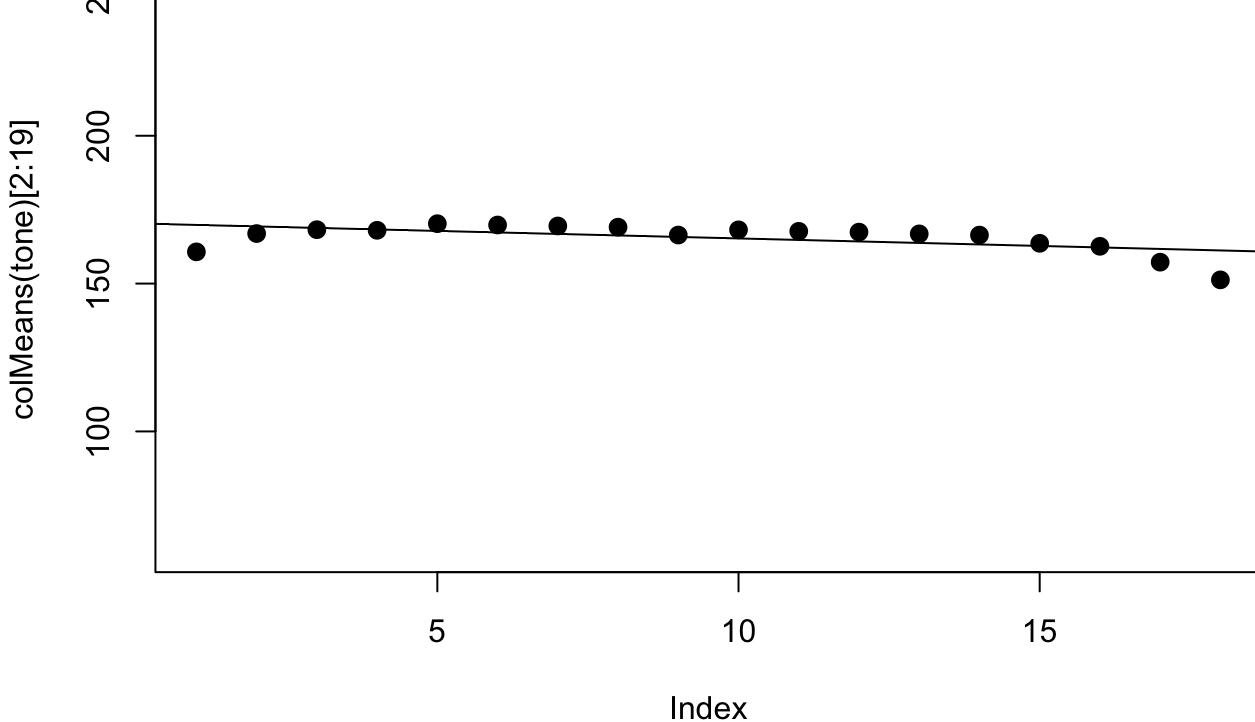


```
## [1] -1.457687 -3.538611 -2.323780
```

F0 and regression line for tone A1, tokens in carriers.

```
plot_f0_by_noise("A1", FALSE)
```

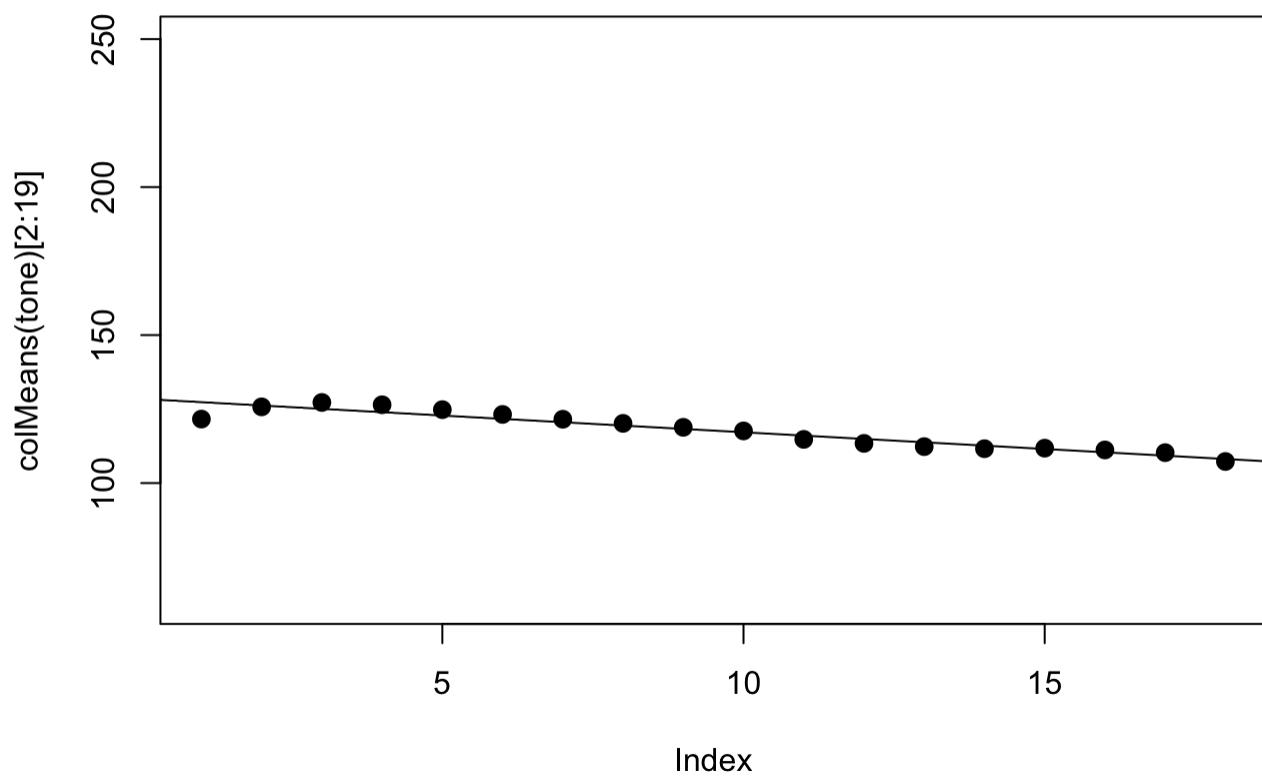
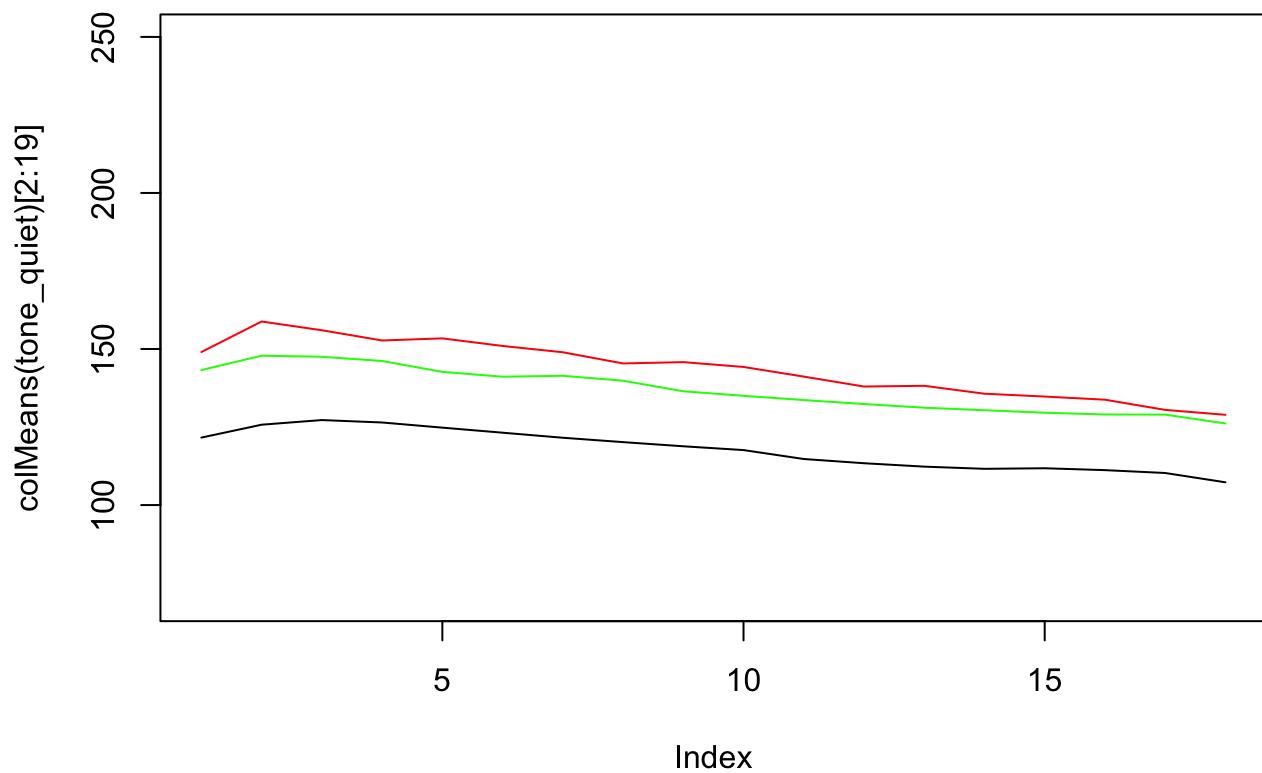



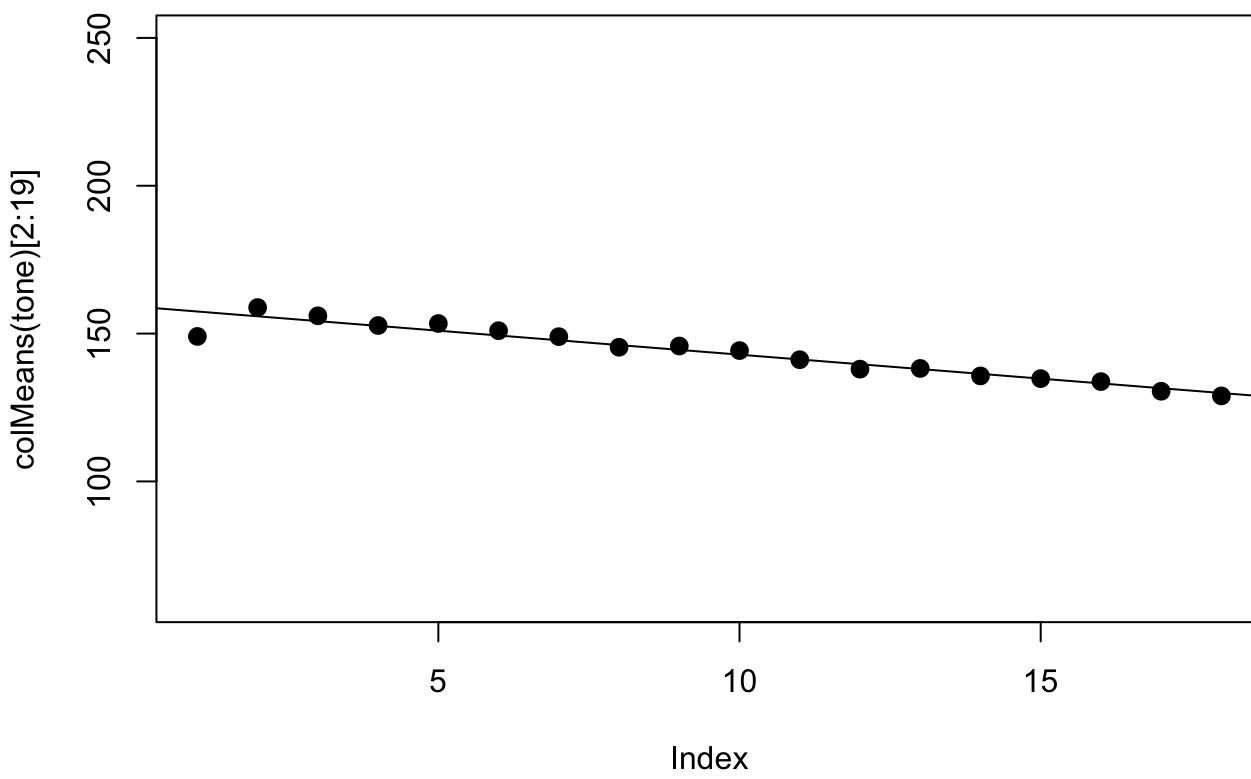
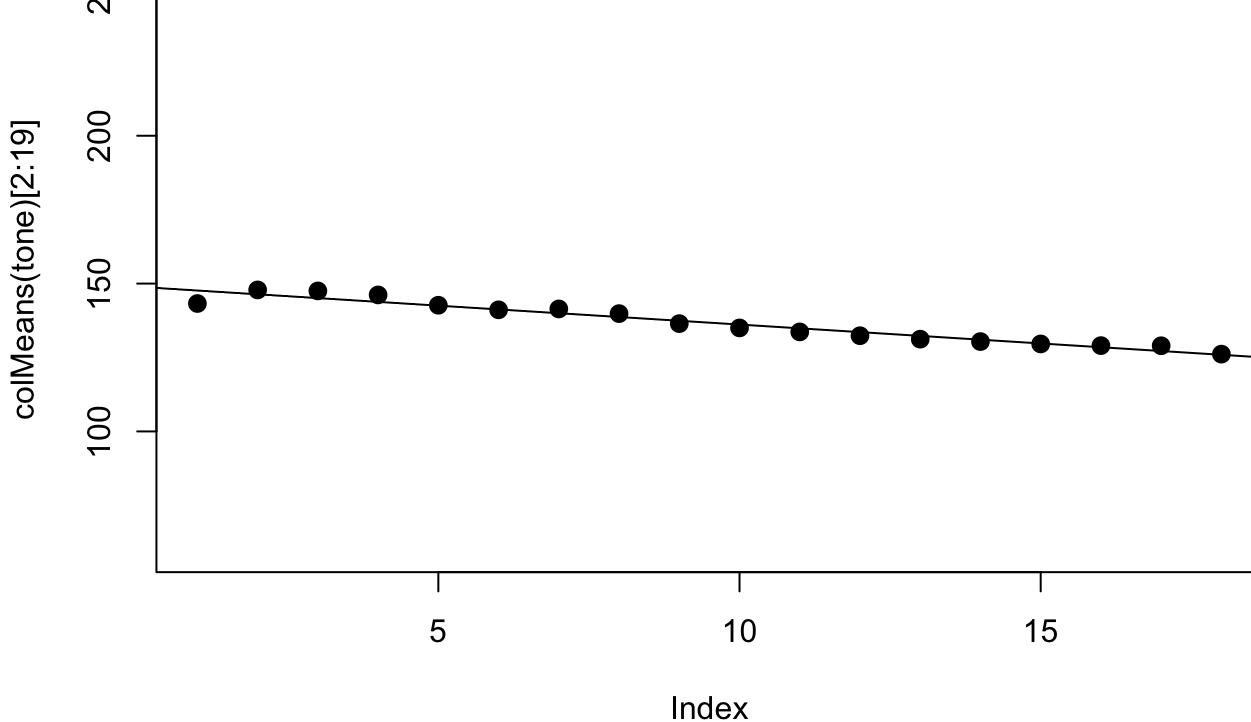


```
## [1] -0.6754738 -0.5120815 -0.6169642
```

F0 and regression line for tone A2, tokens in carriers.

```
plot_f0_by_noise("A2", FALSE)
```

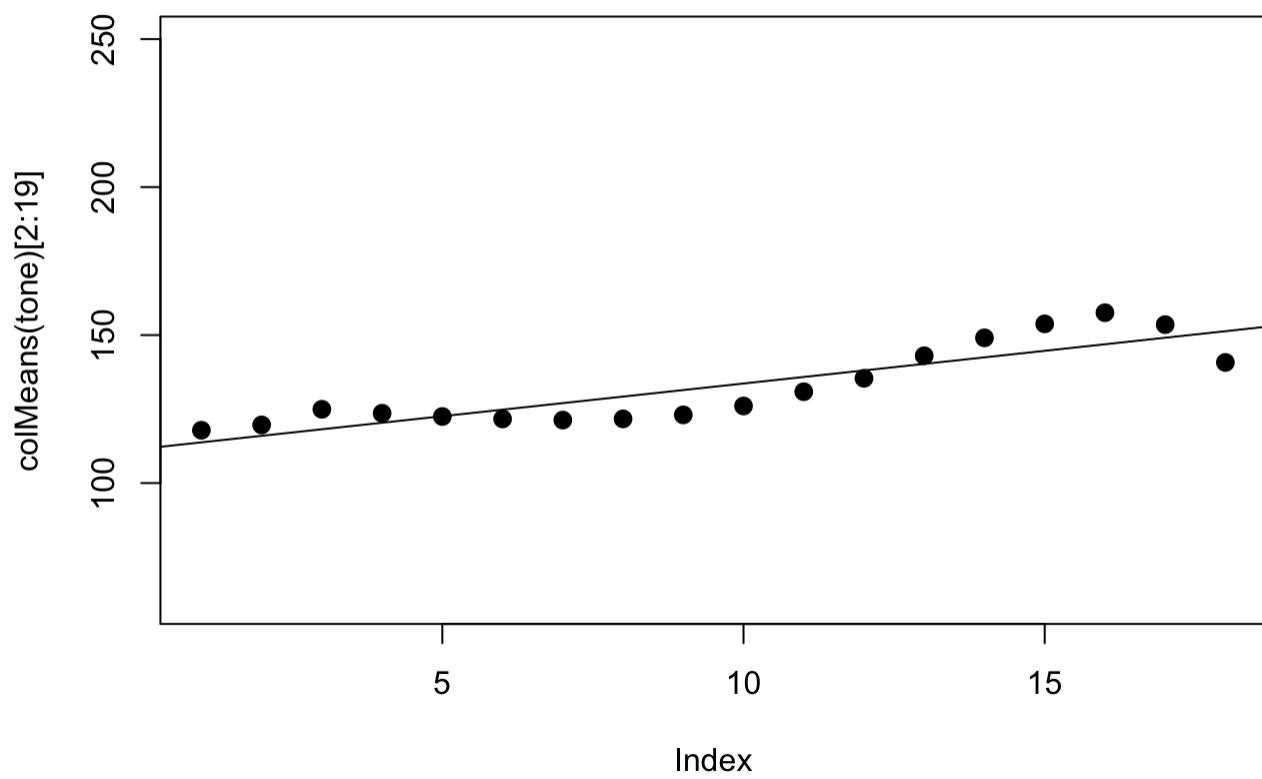
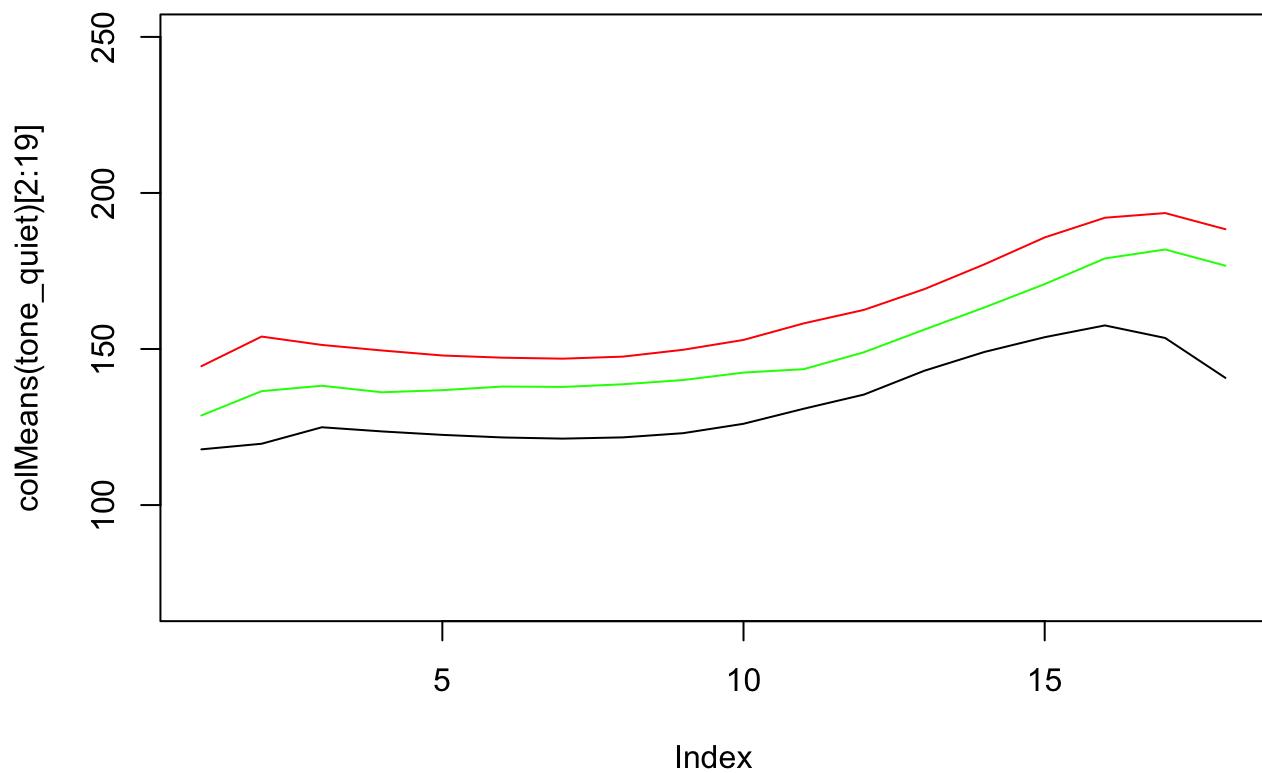



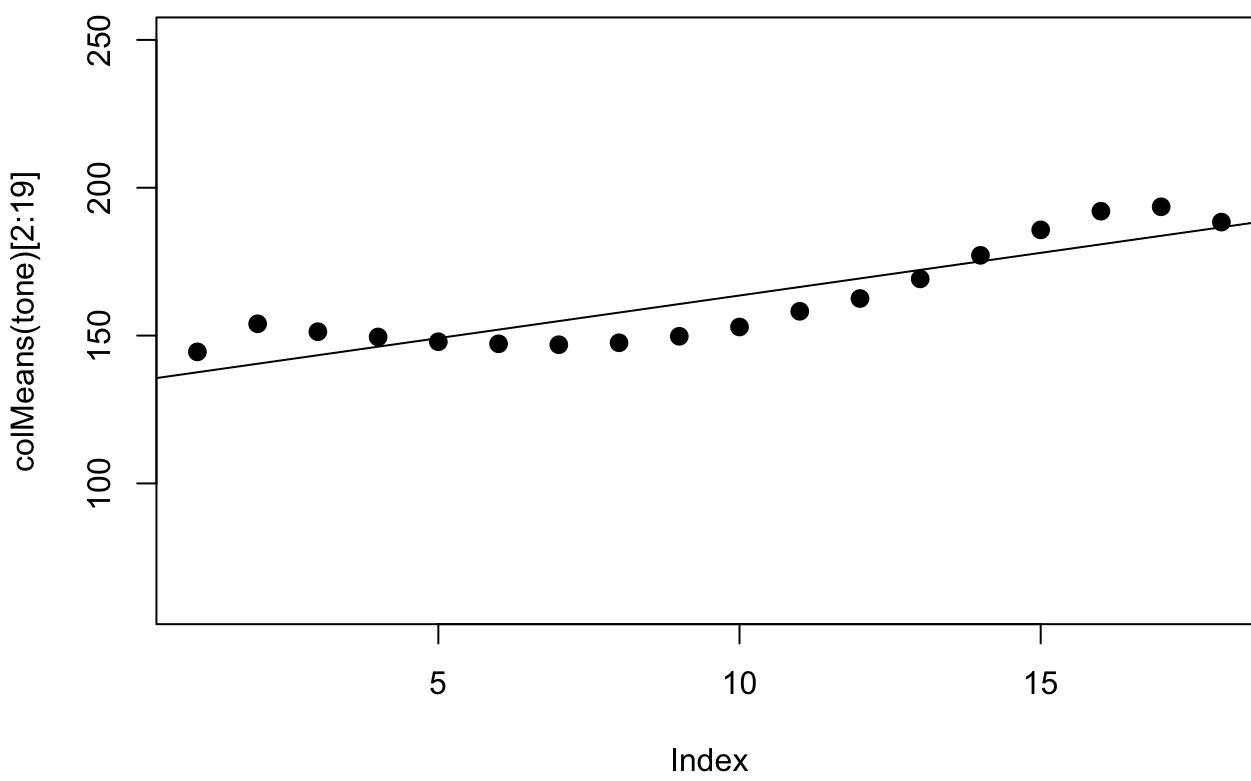
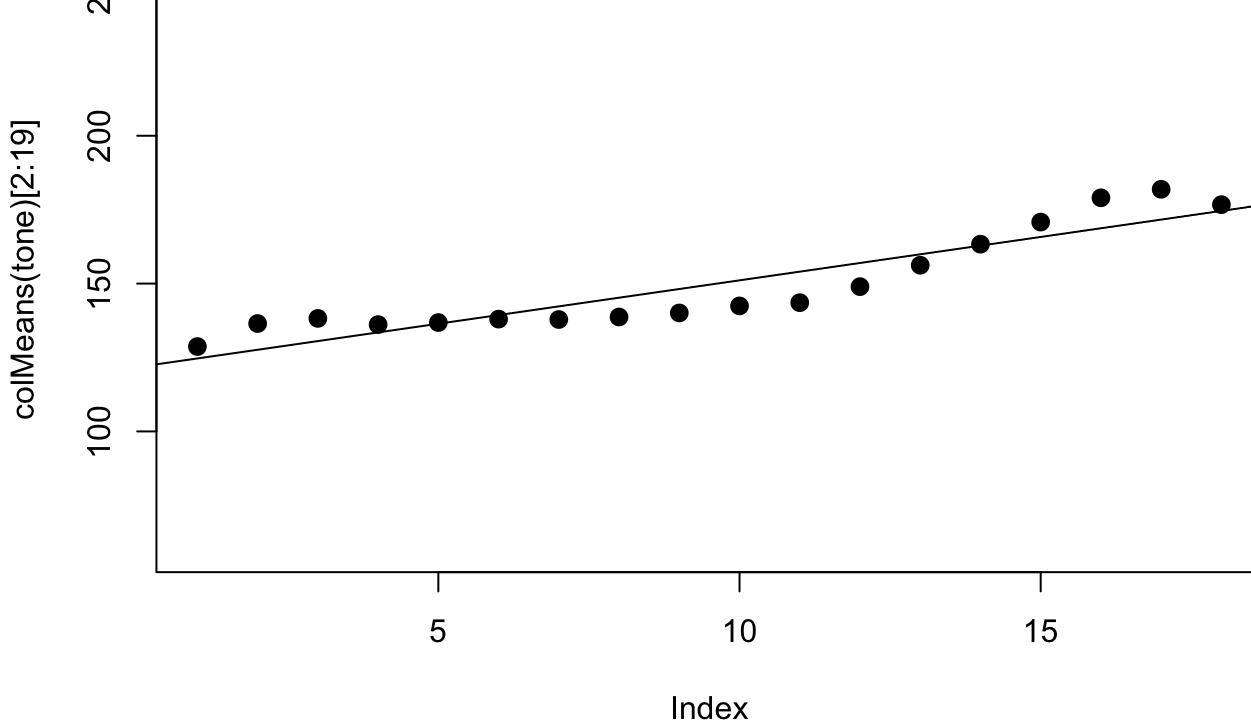


```
## [1] -1.133775 -1.282178 -1.621849
```

F0 and regression line for tone B1, tokens in carriers.

```
plot_f0_by_noise("B1", FALSE)
```

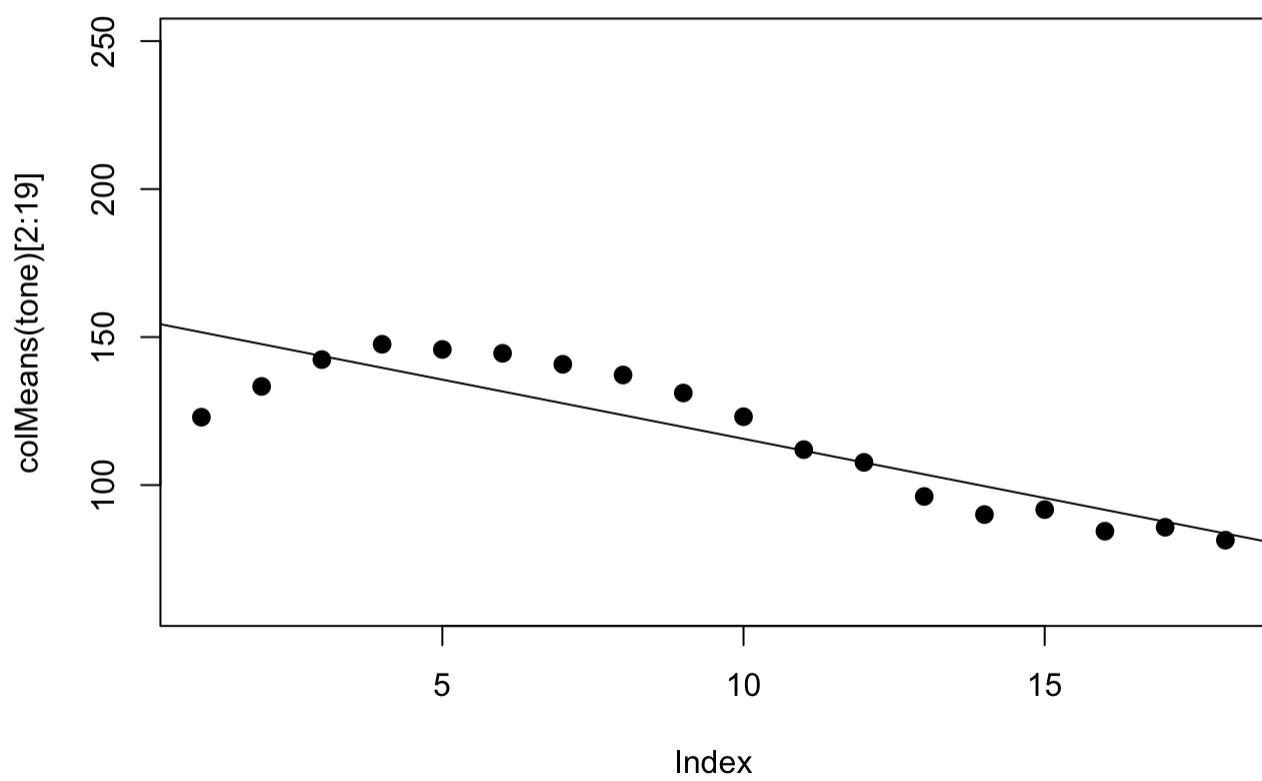
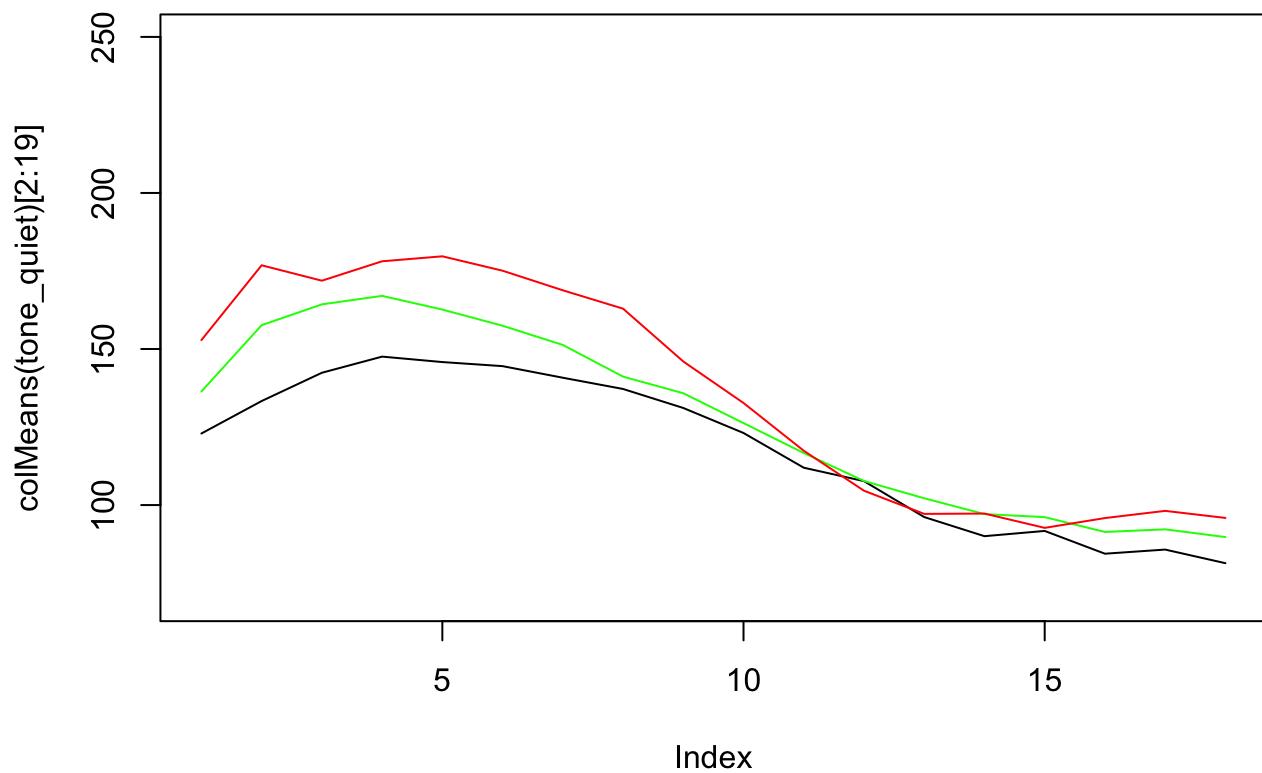



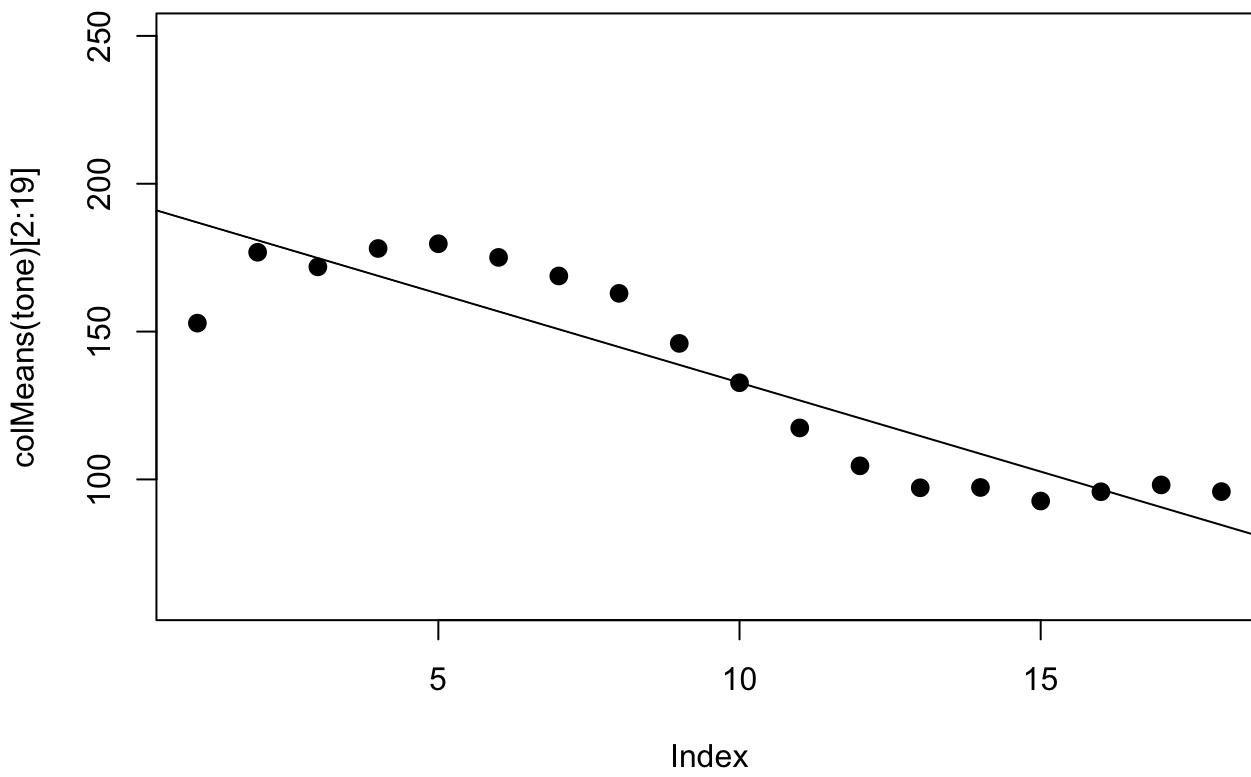
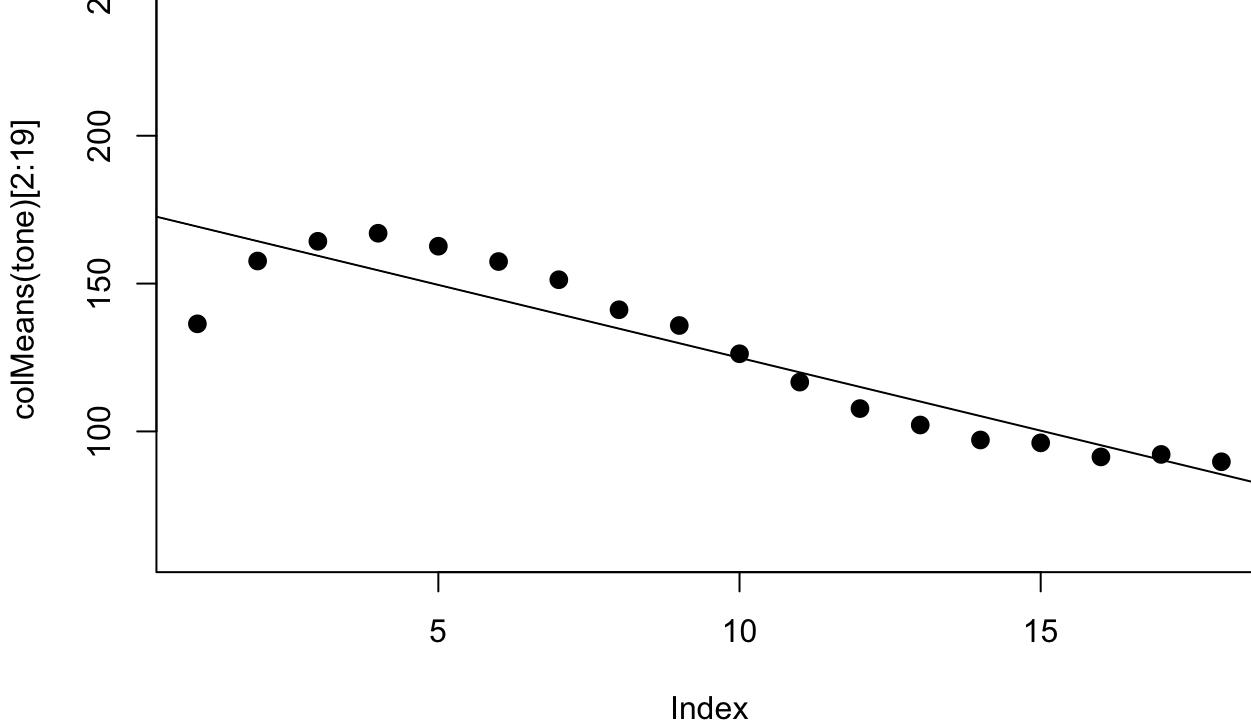


```
## [1] 2.212407 2.938129 2.887313
```

F0 and regression line for tone B2, tokens in carriers.

```
plot_f0_by_noise("B2", FALSE)
```

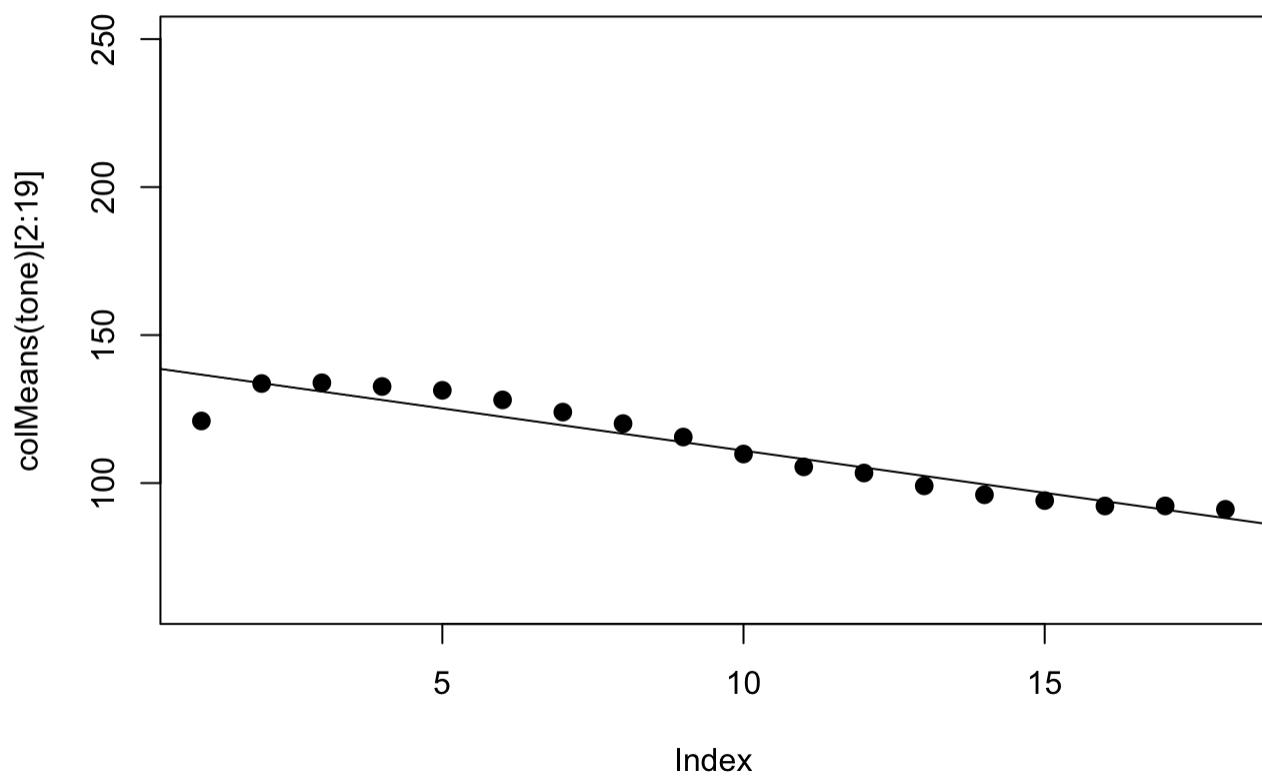
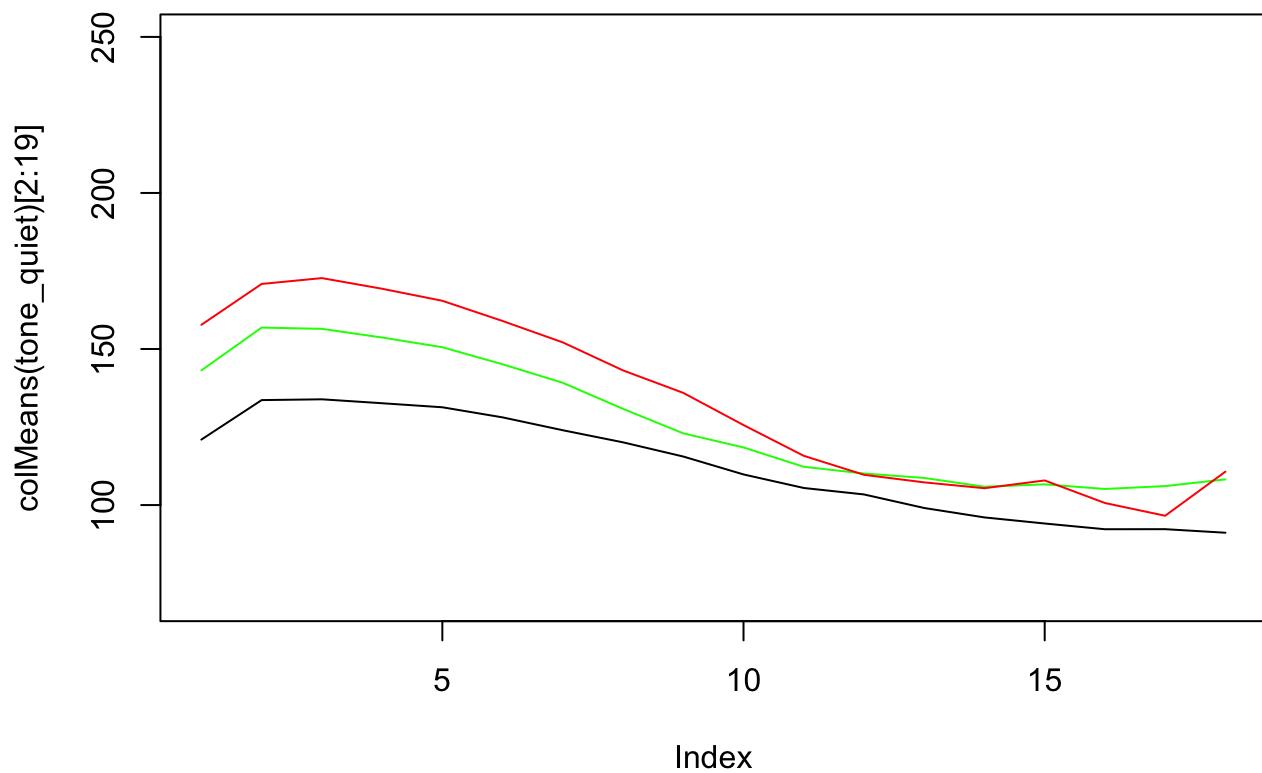



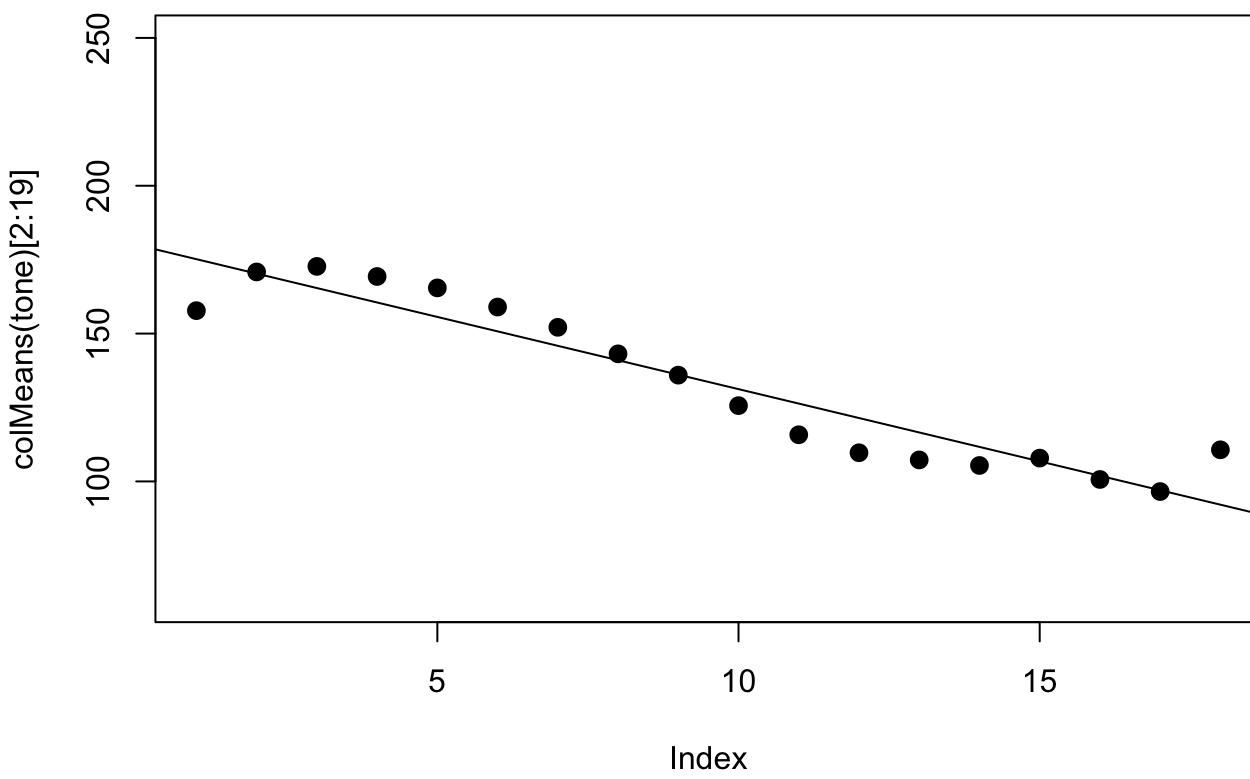
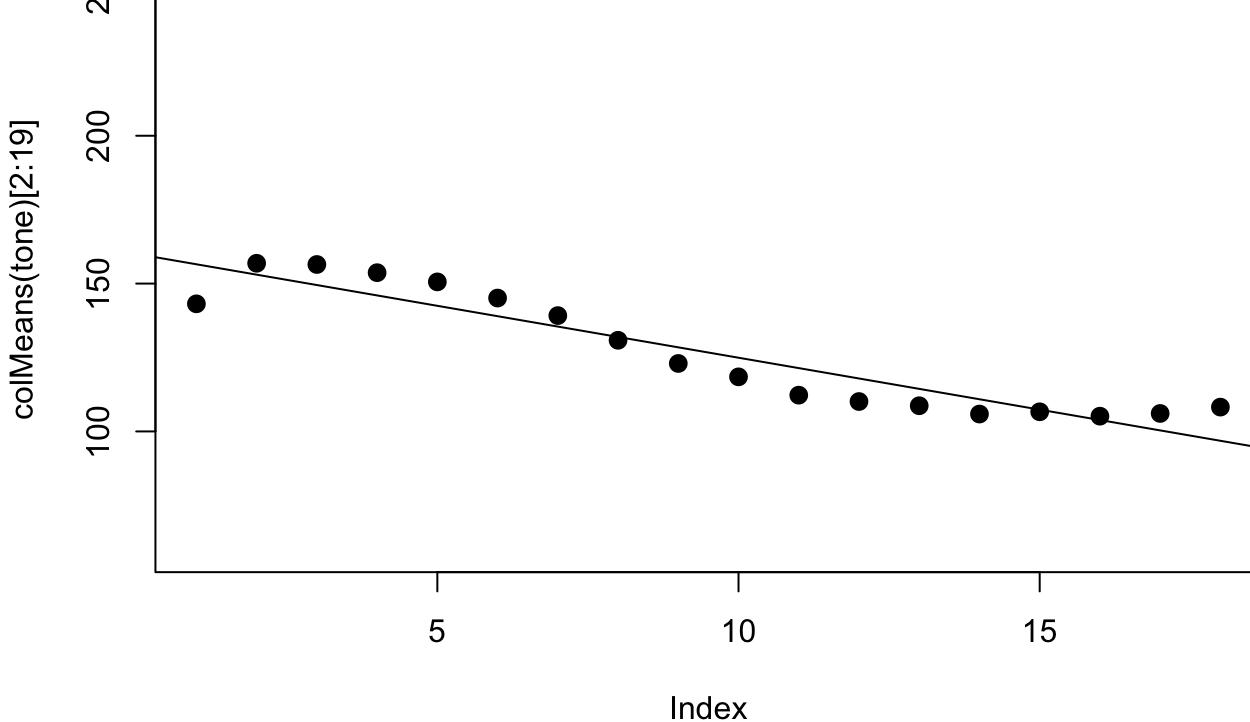


```
## [1] -4.002273 -4.932588 -6.022167
```

F0 and regression line for tone C1, tokens in carriers.

```
plot_f0_by_noise("C1", FALSE)
```

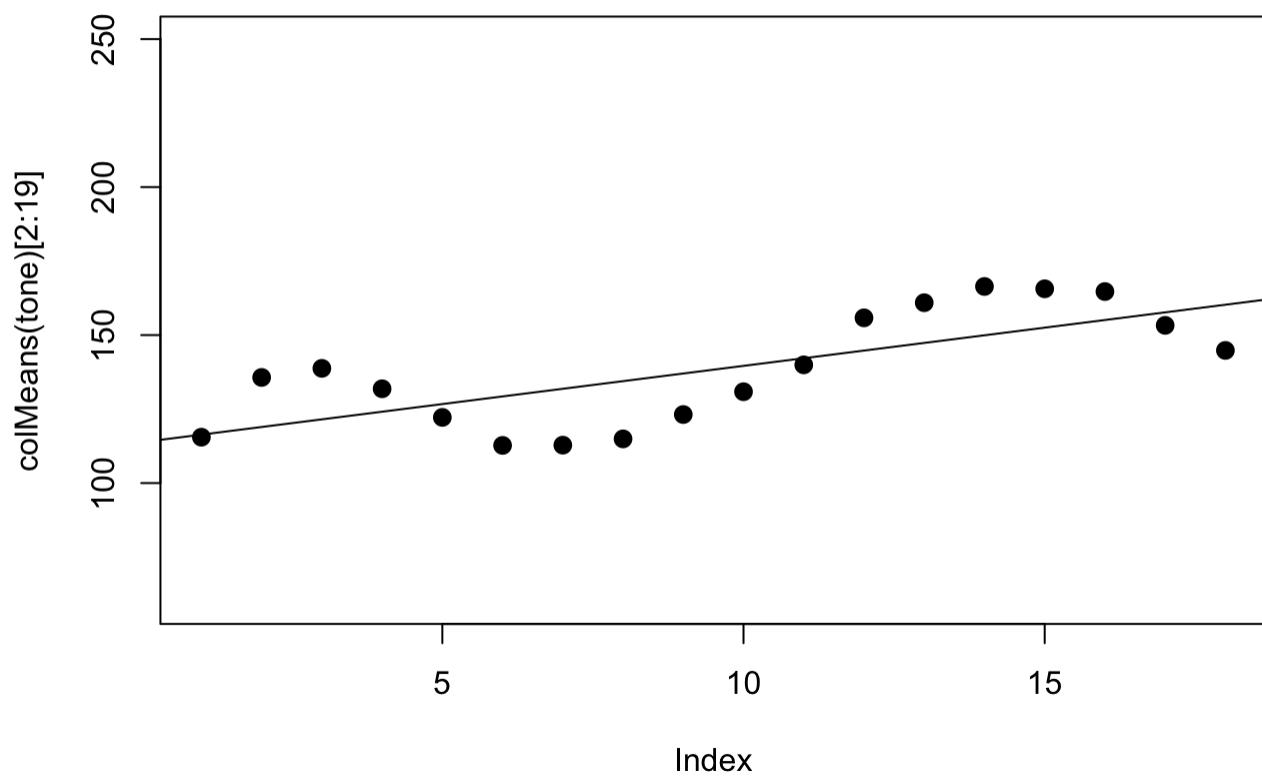
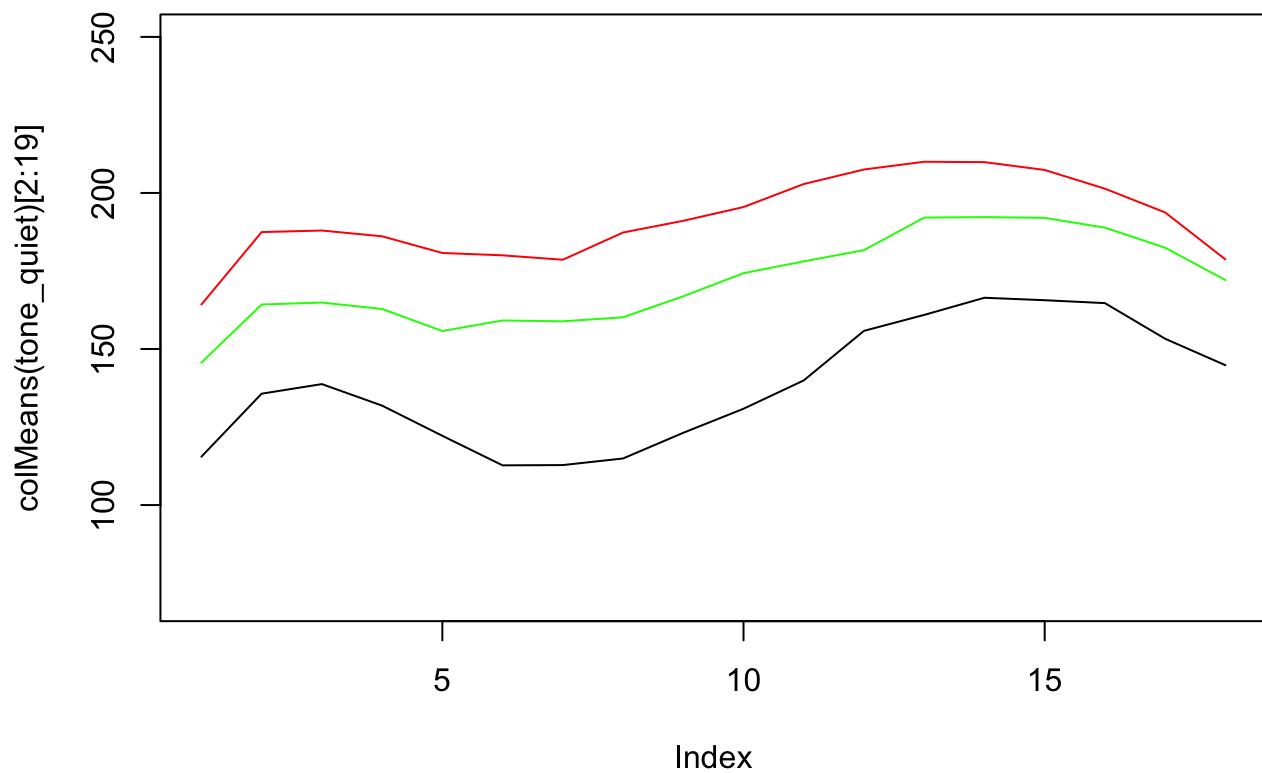



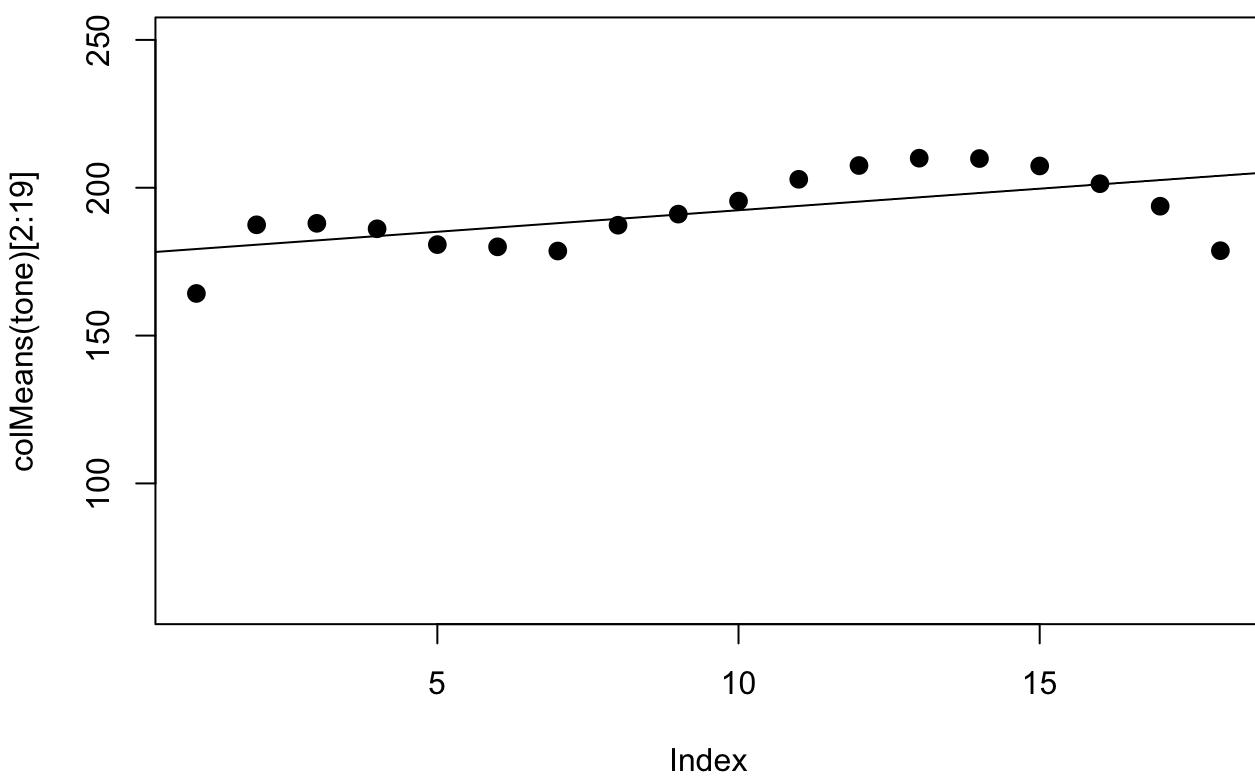
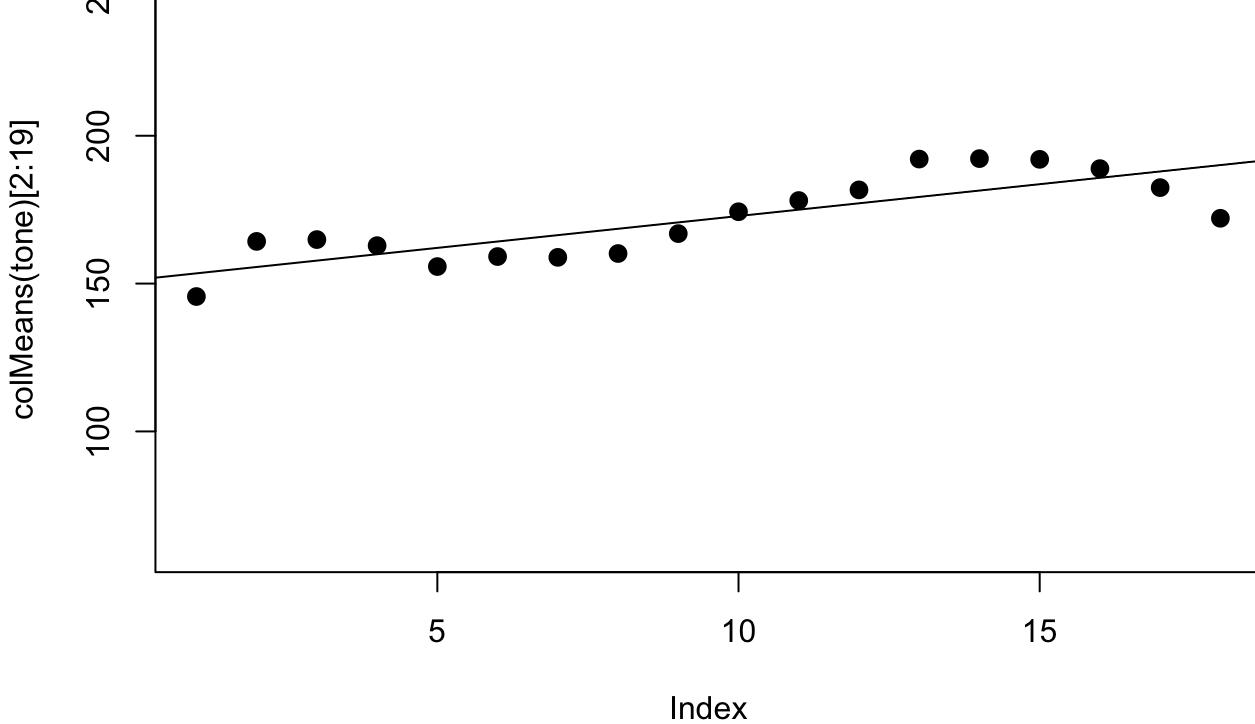


```
## [1] -2.854671 -3.511772 -4.884935
```

F0 and regression line for tone C2, tokens in carriers.

```
plot_f0_by_noise("C2", FALSE)
```

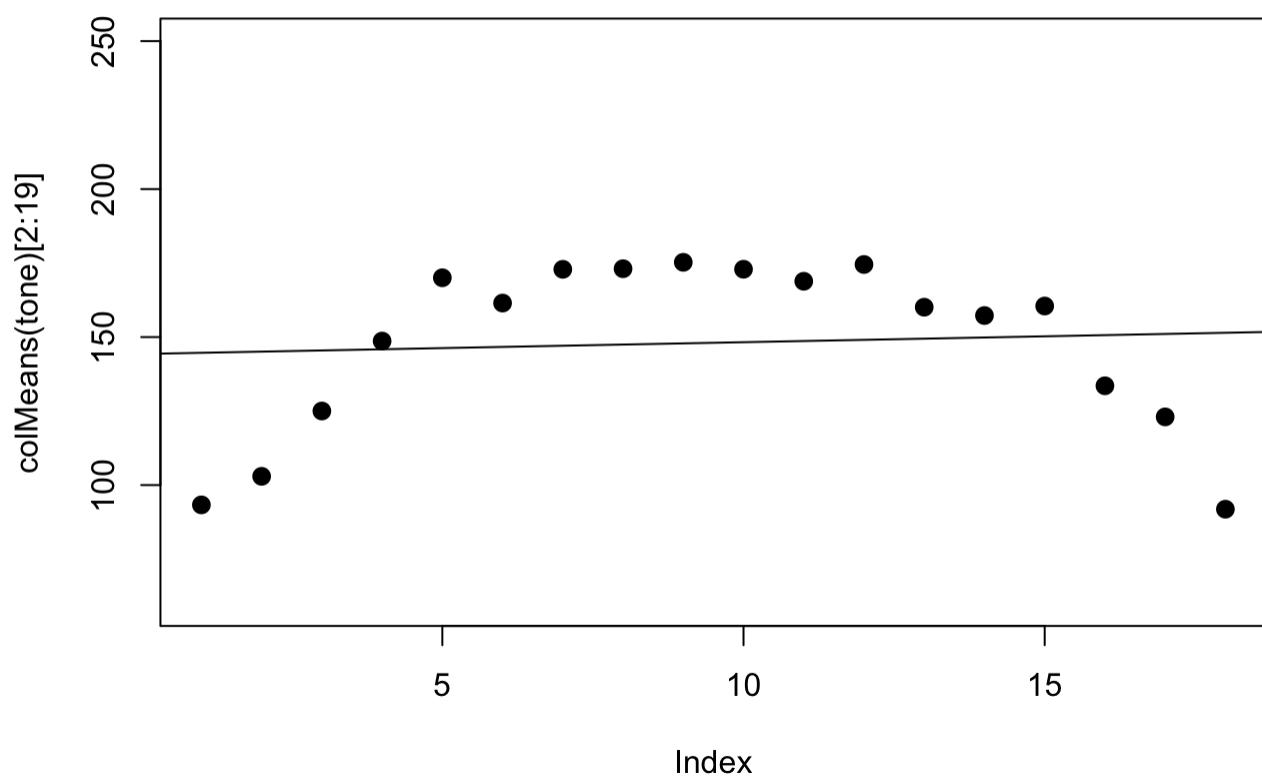
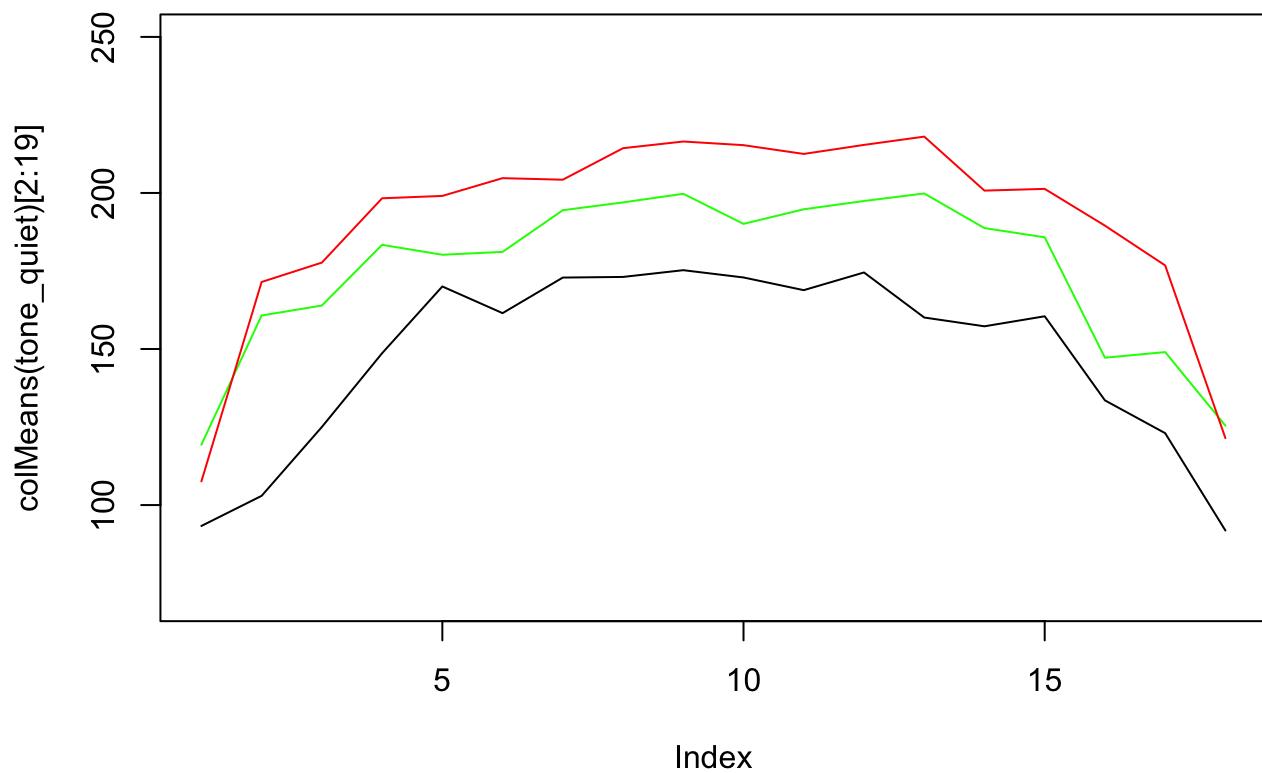



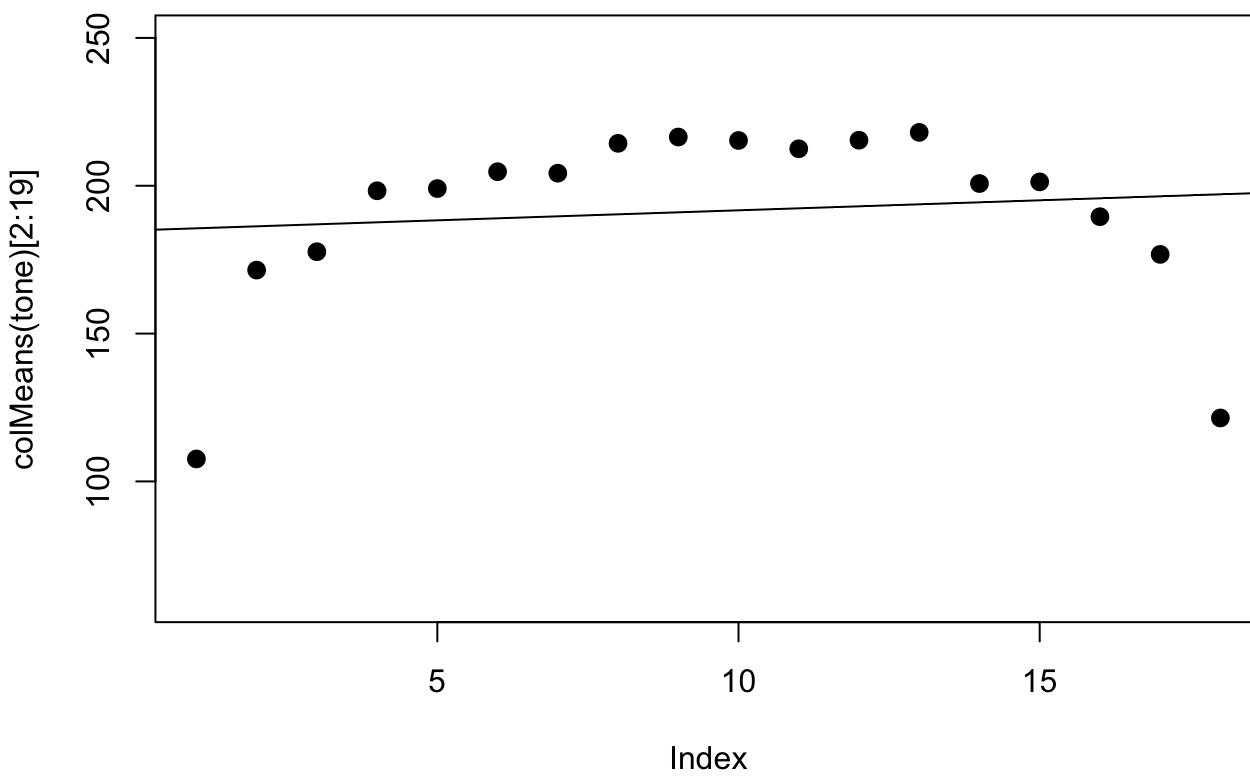
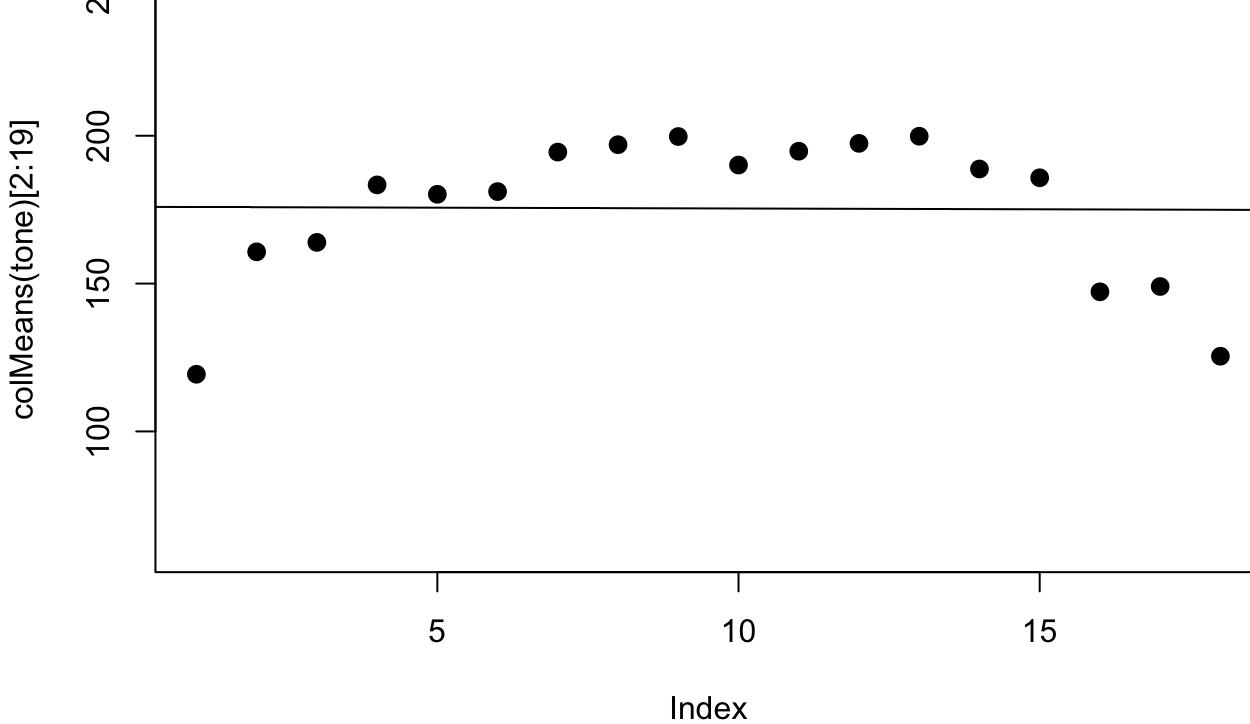


```
## [1] 2.584509 2.157254 1.460962
```

F0 and regression line for tone D1, tokens in carriers.

```
plot_f0_by_noise("D1", FALSE)
```

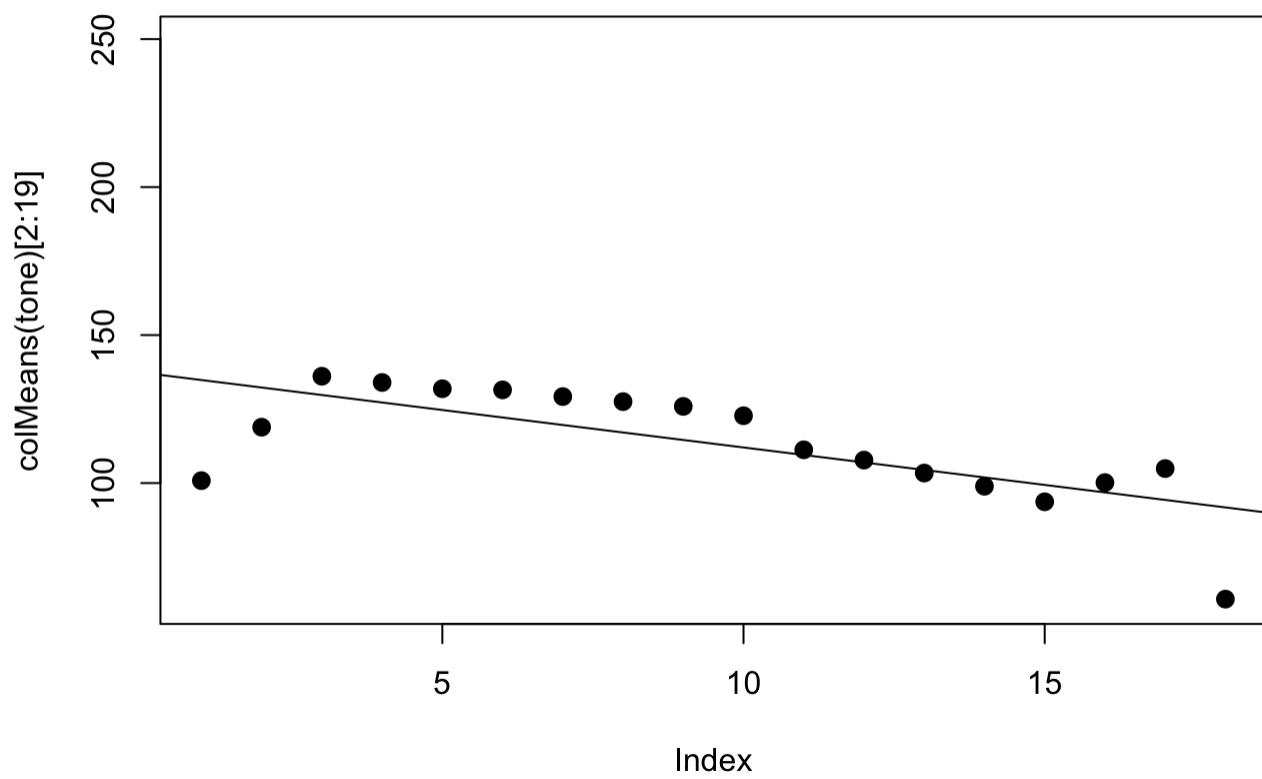
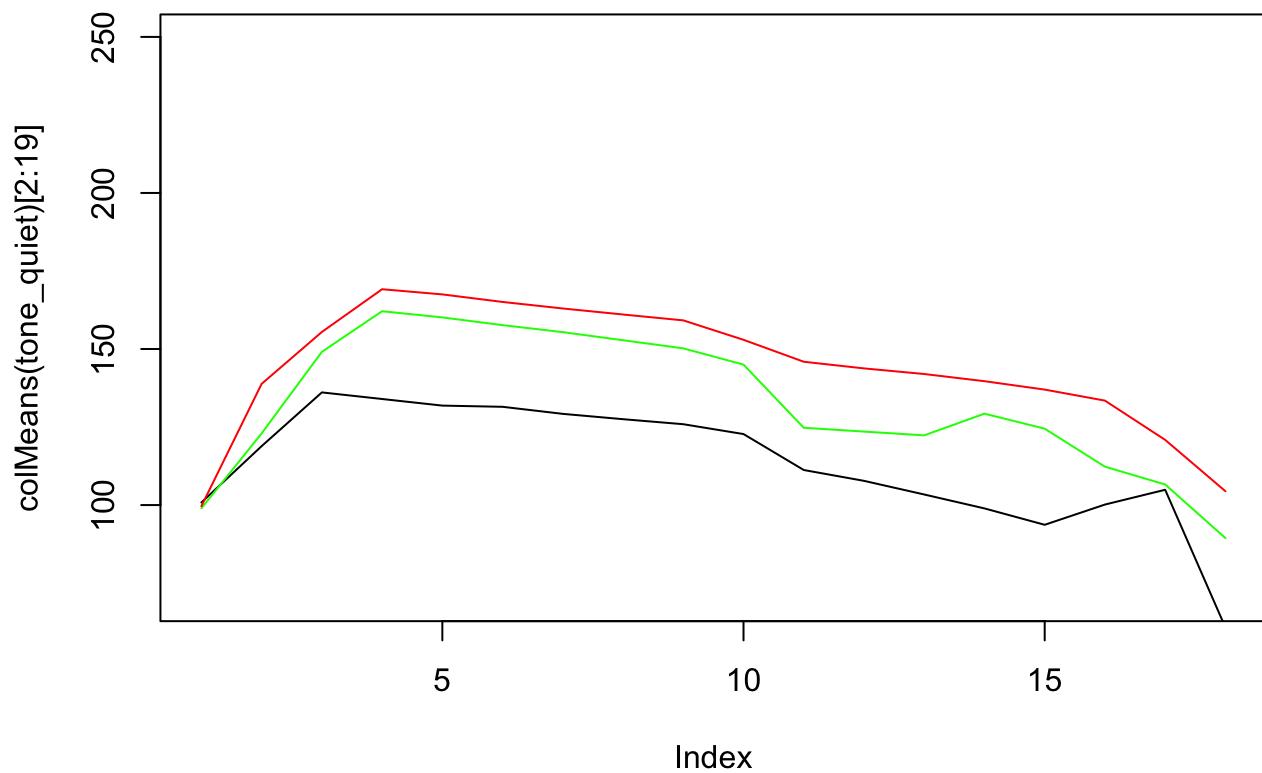



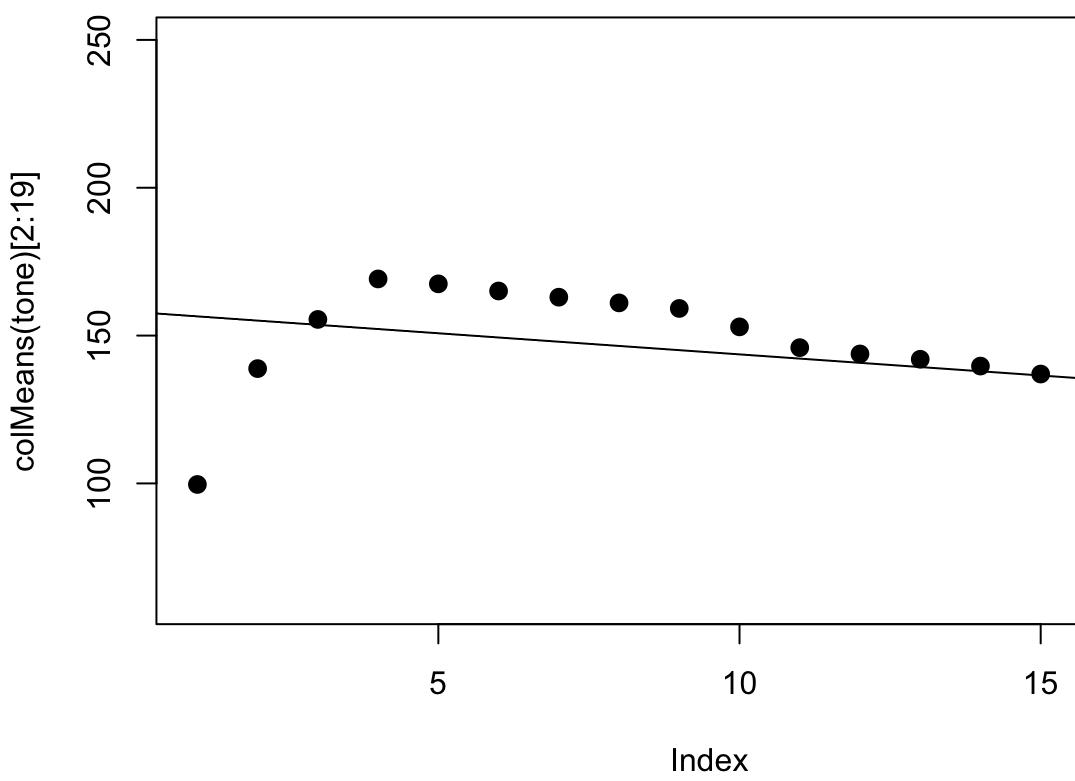
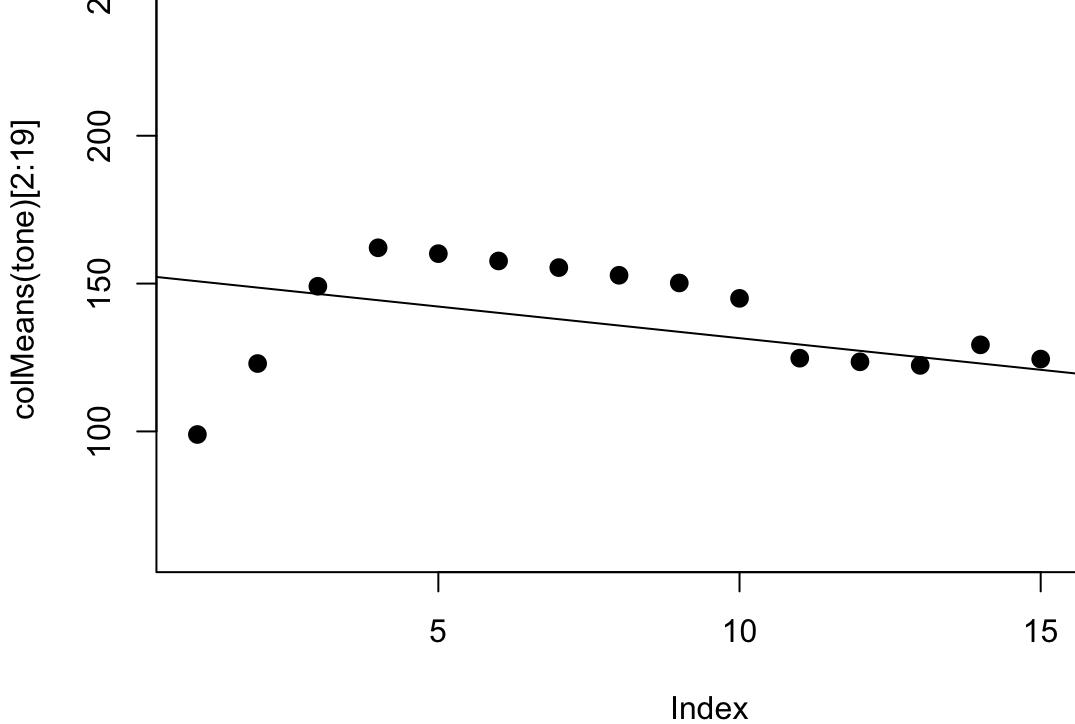


```
## [1] 0.3983035 -0.0583307 0.6809495
```

F0 and regression line for tone D2, tokens in carriers.

```
plot_f0_by_noise("D2", FALSE)
```



```
## [1] -2.531688 -2.139725 -1.431408
```

All coefficients together in a table.

```

# Table of tone * noise (20x3)
tab <- matrix(,nrow=16, ncol=3, byrow=TRUE)
colnames(tab) <- c('quiet','noise 78','noise 90')
rownames(tab) <- c('A1 single','A2 single','B1 single', 'B2 single',
                    'C1 single','C2 single','D1 single', 'D2 single',
                    'A1 carrier','A2 carrier','B1 carrier', 'B2 carrier',
                    'C1 carrier','C2 carrier','D1 carrier', 'D2 carrier')
tab <- as.table(tab)
tab[1,1] = coefficient_reports(filter_f0("A1", "0", TRUE))
tab[1,2] = coefficient_reports(filter_f0("A1", "78", TRUE))
tab[1,3] = coefficient_reports(filter_f0("A1", "90", TRUE))

tab[2,1] = coefficient_reports(filter_f0("A2", "0", TRUE))
tab[2,2] = coefficient_reports(filter_f0("A2", "78", TRUE))
tab[2,3] = coefficient_reports(filter_f0("A2", "90", TRUE))

tab[3,1] = coefficient_reports(filter_f0("B1", "0", TRUE))
tab[3,2] = coefficient_reports(filter_f0("B1", "78", TRUE))
tab[3,3] = coefficient_reports(filter_f0("B1", "90", TRUE))

tab[4,1] = coefficient_reports(filter_f0("B2", "0", TRUE))
tab[4,2] = coefficient_reports(filter_f0("B2", "78", TRUE))
tab[4,3] = coefficient_reports(filter_f0("B2", "90", TRUE))

tab[5,1] = coefficient_reports(filter_f0("C1", "0", TRUE))
tab[5,2] = coefficient_reports(filter_f0("C1", "78", TRUE))
tab[5,3] = coefficient_reports(filter_f0("C1", "90", TRUE))

tab[6,1] = coefficient_reports(filter_f0("C2", "0", TRUE))
tab[6,2] = coefficient_reports(filter_f0("C2", "78", TRUE))
tab[6,3] = coefficient_reports(filter_f0("C2", "90", TRUE))

tab[7,1] = coefficient_reports(filter_f0("D1", "0", TRUE))
tab[7,2] = coefficient_reports(filter_f0("D1", "78", TRUE))
tab[7,3] = coefficient_reports(filter_f0("D1", "90", TRUE))

tab[8,1] = coefficient_reports(filter_f0("D2", "0", TRUE))
tab[8,2] = coefficient_reports(filter_f0("D2", "78", TRUE))
tab[8,3] = coefficient_reports(filter_f0("D2", "90", TRUE))

# carrier
tab[9,1] = coefficient_reports(filter_f0("A1", "0", FALSE))
tab[9,2] = coefficient_reports(filter_f0("A1", "78", FALSE))
tab[9,3] = coefficient_reports(filter_f0("A1", "90", FALSE))

tab[10,1] = coefficient_reports(filter_f0("A2", "0", FALSE))
tab[10,2] = coefficient_reports(filter_f0("A2", "78", FALSE))
tab[10,3] = coefficient_reports(filter_f0("A2", "90", FALSE))

tab[11,1] = coefficient_reports(filter_f0("B1", "0", FALSE))
tab[11,2] = coefficient_reports(filter_f0("B1", "78", FALSE))
tab[11,3] = coefficient_reports(filter_f0("B1", "90", FALSE))

tab[12,1] = coefficient_reports(filter_f0("B2", "0", FALSE))
tab[12,2] = coefficient_reports(filter_f0("B2", "78", FALSE))
tab[12,3] = coefficient_reports(filter_f0("B2", "90", FALSE))

tab[13,1] = coefficient_reports(filter_f0("C1", "0", FALSE))

```

```

tab[13,2] = coefficient_reports(filter_f0("C1", "78", FALSE))
tab[13,3] = coefficient_reports(filter_f0("C1", "90", FALSE))

tab[14,1] = coefficient_reports(filter_f0("C2", "0", FALSE))
tab[14,2] = coefficient_reports(filter_f0("C2", "78", FALSE))
tab[14,3] = coefficient_reports(filter_f0("C2", "90", FALSE))

tab[15,1] = coefficient_reports(filter_f0("D1", "0", FALSE))
tab[15,2] = coefficient_reports(filter_f0("D1", "78", FALSE))
tab[15,3] = coefficient_reports(filter_f0("D1", "90", FALSE))

tab[16,1] = coefficient_reports(filter_f0("D2", "0", FALSE))
tab[16,2] = coefficient_reports(filter_f0("D2", "78", FALSE))
tab[16,3] = coefficient_reports(filter_f0("D2", "90", FALSE))

tab

```

```

##          quiet    noise 78    noise 90
## A1 single -0.5672417 -0.8400165 -0.5233609
## A2 single -1.3786352 -1.7597751 -1.5401346
## B1 single  2.5854175  4.6535103  5.2011062
## B2 single -3.4163446 -6.6367399 -6.3098807
## C1 single -1.2545513 -1.4060692 -1.9789597
## C2 single  2.5676060  1.8890026  0.8420454
## D1 single -0.3939798  3.5934800  1.1236231
## D2 single -1.4576871 -3.5386107 -2.3237797
## A1 carrier -0.6754738 -0.5120815 -0.6169642
## A2 carrier -1.1337752 -1.2821783 -1.6218491
## B1 carrier  2.2124070  2.9381288  2.8873133
## B2 carrier -4.0022734 -4.9325882 -6.0221675
## C1 carrier -2.8546710 -3.5117715 -4.8849348
## C2 carrier  2.5845091  2.1572536  1.4609616
## D1 carrier  0.3983035 -0.0583307  0.6809495
## D2 carrier -2.5316879 -2.1397249 -1.4314081

```

Tone Contour Analysis on Different Noise Levels on Vowel A.

```
# Add vowel annotation.
```

```
f0_reports$vowel <- ifelse(grepl("a", f0_reports$syllable.name, ignore.case=T), "A",
ifelse(grepl("à", f0_reports$syllable.name, ignore.case=T), "A",
ifelse(grepl("á", f0_reports$syllable.name, ignore.case=T), "A",
ifelse(grepl("å", f0_reports$syllable.name, ignore.case=T), "A",
ifelse(grepl("ã", f0_reports$syllable.name, ignore.case=T), "A",
ifelse(grepl("ä", f0_reports$syllable.name, ignore.case=T), "A",
ifelse(grepl("ê", f0_reports$syllable.name, ignore.case=T), "E",
ifelse(grepl("è", f0_reports$syllable.name, ignore.case=T), "E",
ifelse(grepl("é", f0_reports$syllable.name, ignore.case=T), "E",
ifelse(grepl("ë", f0_reports$syllable.name, ignore.case=T), "E",
ifelse(grepl("ë", f0_reports$syllable.name, ignore.case=T), "E",
ifelse(grepl("ë", f0_reports$syllable.name, ignore.case=T), "E",
ifelse(grepl("ù", f0_reports$syllable.name, ignore.case=T), "U",
ifelse(grepl("ú", f0_reports$syllable.name, ignore.case=T), "U",
ifelse(grepl("ô", f0_reports$syllable.name, ignore.case=T), "O", "NA")))))))))))))))))
```

```
# Convert vowel types to a factor variable
```

```
f0_reports$vowel <- as.factor(f0_reports$vowel)
head(f0_reports, 30)
```

sound.name	1	2	3	4	5	6	7	8
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1 A_single.wav	0.0000	203.65147	215.6574	220.6682	218.4150	216.3874	217.3327	218.4298
2 A_single1.wav	0.0000	203.65147	215.6574	220.6682	218.4150	216.3874	217.3327	218.4298
3 TA_single.wav	178.7868	225.96398	231.4816	231.0169	229.4844	231.4189	229.7609	229.6564
4 TA_single1.wav	0.0000	0.00000	0.0000	229.0033	230.5813	229.9129	229.7844	228.6812
5 THA_single.wav	266.1095	260.00300	246.5207	236.1820	230.7340	223.6295	218.8480	216.4334
6 THA_single1.wav	225.8704	261.01095	241.9743	232.6051	230.3699	229.7041	229.5388	225.6905
7 THU_single.wav	301.9594	321.52027	309.8326	286.7558	284.3097	283.6454	281.3352	282.6425
8 THU_single1.wav	0.0000	18.49505	303.7577	287.5523	281.9481	279.9251	277.8155	277.3324
9 THÀ_single.wav	146.0353	198.82998	191.8680	189.8348	185.8575	185.5470	184.3921	180.8341
10 THÀ_single1.wav	203.9195	228.61828	220.6974	206.7640	197.1366	189.8947	184.0280	177.4527

1-10 of 30 rows | 1-10 of 28 columns

Previous **1** 2 3 Next

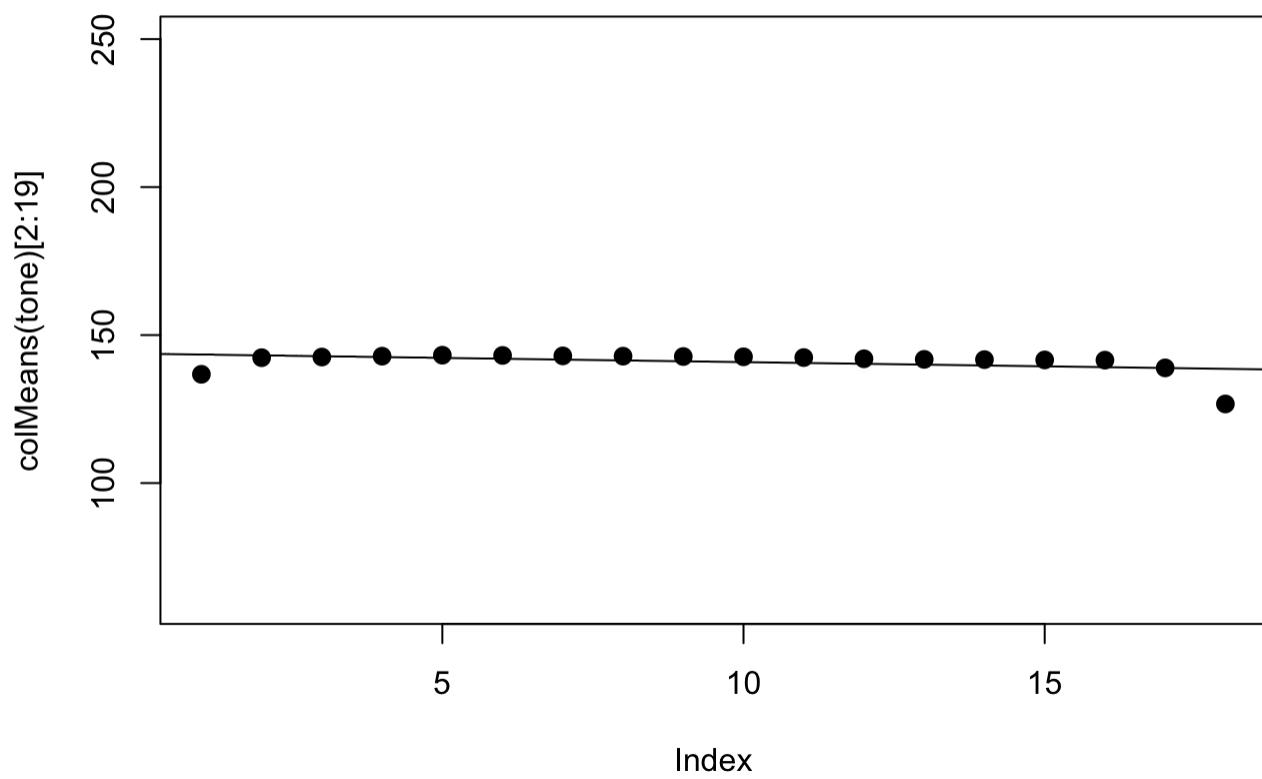
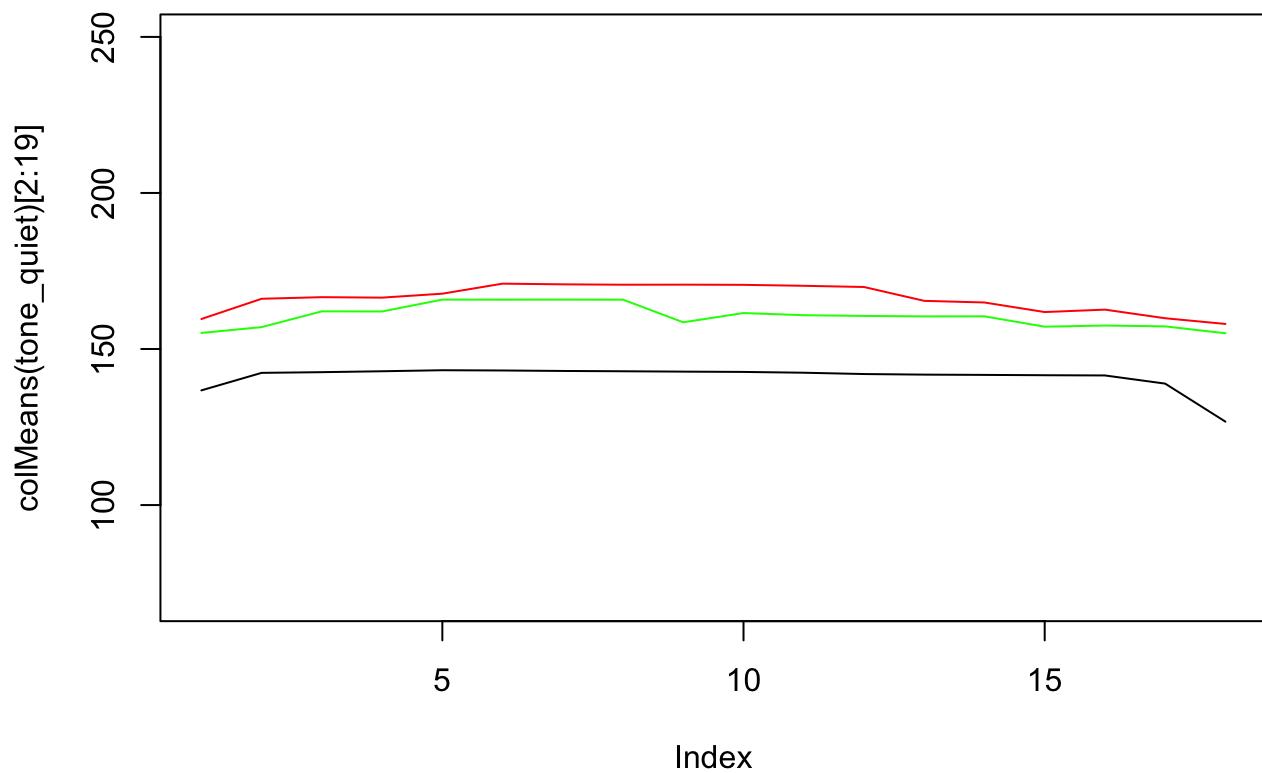
```
# Plot F0 contour by vowels (not distinguishing single tokens or tokens in carriers.)
```

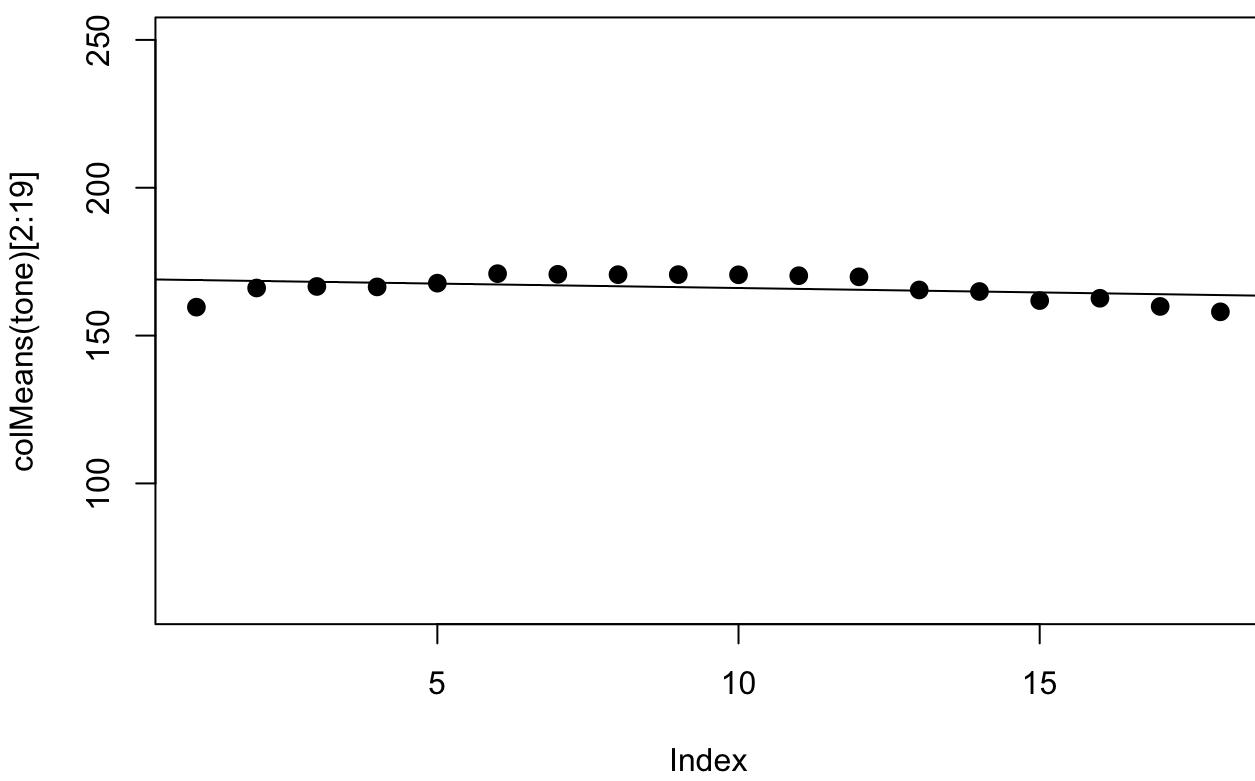
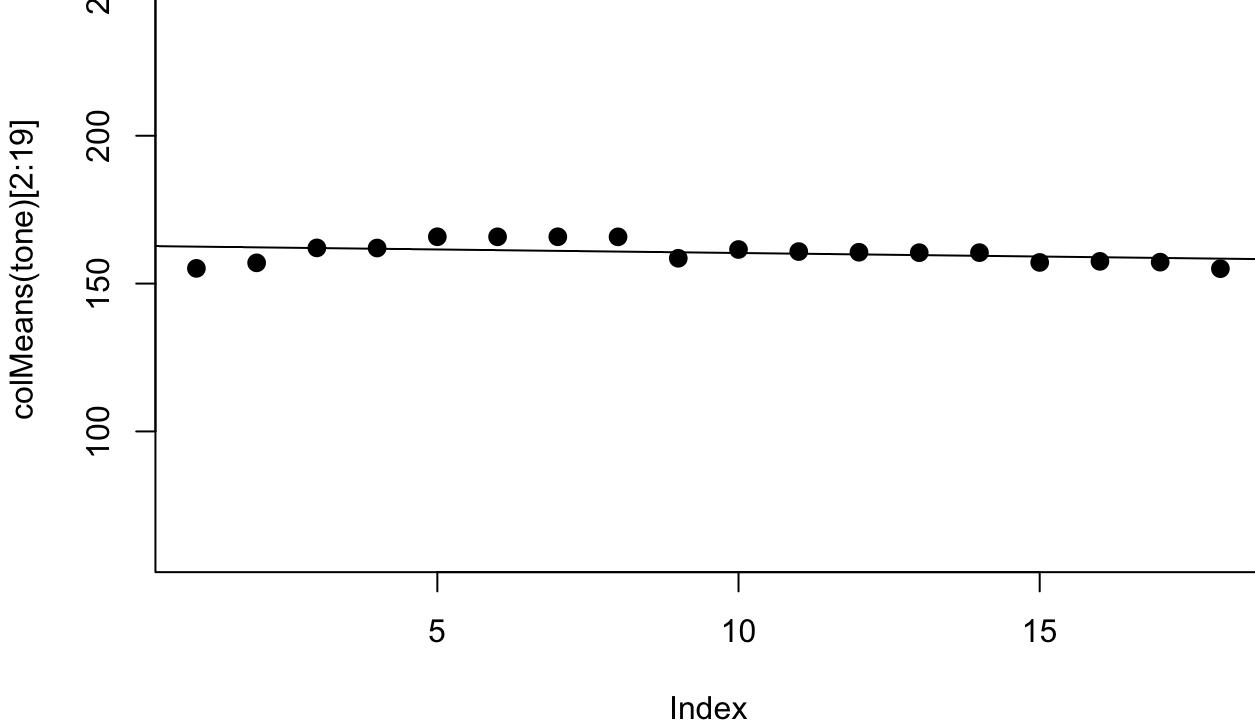
```
plot_f0_by_vowels <- function(tone, vowel) {
  if (vowel=="A") {
    tone_quiet <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="0" & f0_reports$vowel=="A",-c(1,c(22:26))])
    tone_78 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="78" & f0_reports$vowel=="A",-c(1,c(22:26))])
    tone_90 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="90" & f0_reports$vowel=="A",-c(1,c(22:26))])
    plot(colMeans(tone_quiet)[2:19], type="l", xlim=c(1, 18), ylim=c(70, 250))
    lines(colMeans(tone_78)[2:19], col="green")
    lines(colMeans(tone_90)[2:19], col="red")
    # Find regression coefficients and return them.
    c(summary(regression_report(tone_quiet))$coefficients[2,1], summary(regression_report(tone_78))$coefficients[2,1], summary(regression_report(tone_90))$coefficients[2,1])
  } else if (vowel=="E") {
    tone_quiet <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="0" & f0_reports$vowel=="E",-c(1,c(22:26))])
    tone_78 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="78" & f0_reports$vowel=="E",-c(1,c(22:26))])
    tone_90 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="90" & f0_reports$vowel=="E",-c(1,c(22:26))])
    plot(colMeans(tone_quiet)[2:19], type="l", xlim=c(1, 18), ylim=c(70, 250))
    lines(colMeans(tone_78)[2:19], col="green")
    lines(colMeans(tone_90)[2:19], col="red")
    # Find regression coefficients and return them.
    c(summary(regression_report(tone_quiet))$coefficients[2,1], summary(regression_report(tone_78))$coefficients[2,1], summary(regression_report(tone_90))$coefficients[2,1])
  } else {
    tone_quiet <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="0" & f0_reports$vowel=="U",-c(1,c(22:26))])
    tone_78 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="78" & f0_reports$vowel=="U",-c(1,c(22:26))])
    tone_90 <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise=="90" & f0_reports$vowel=="U",-c(1,c(22:26))])
    plot(colMeans(tone_quiet)[2:19], type="l", xlim=c(1, 18), ylim=c(70, 250))
    lines(colMeans(tone_78)[2:19], col="green")
    lines(colMeans(tone_90)[2:19], col="red")
    # Find regression coefficients and return them.
    c(summary(regression_report(tone_quiet))$coefficients[2,1], summary(regression_report(tone_78))$coefficients[2,1], summary(regression_report(tone_90))$coefficients[2,1])
  }
}

# Call functions
```

Tone A1 on A

```
plot_f0_by_vowels("A1", "A")
```

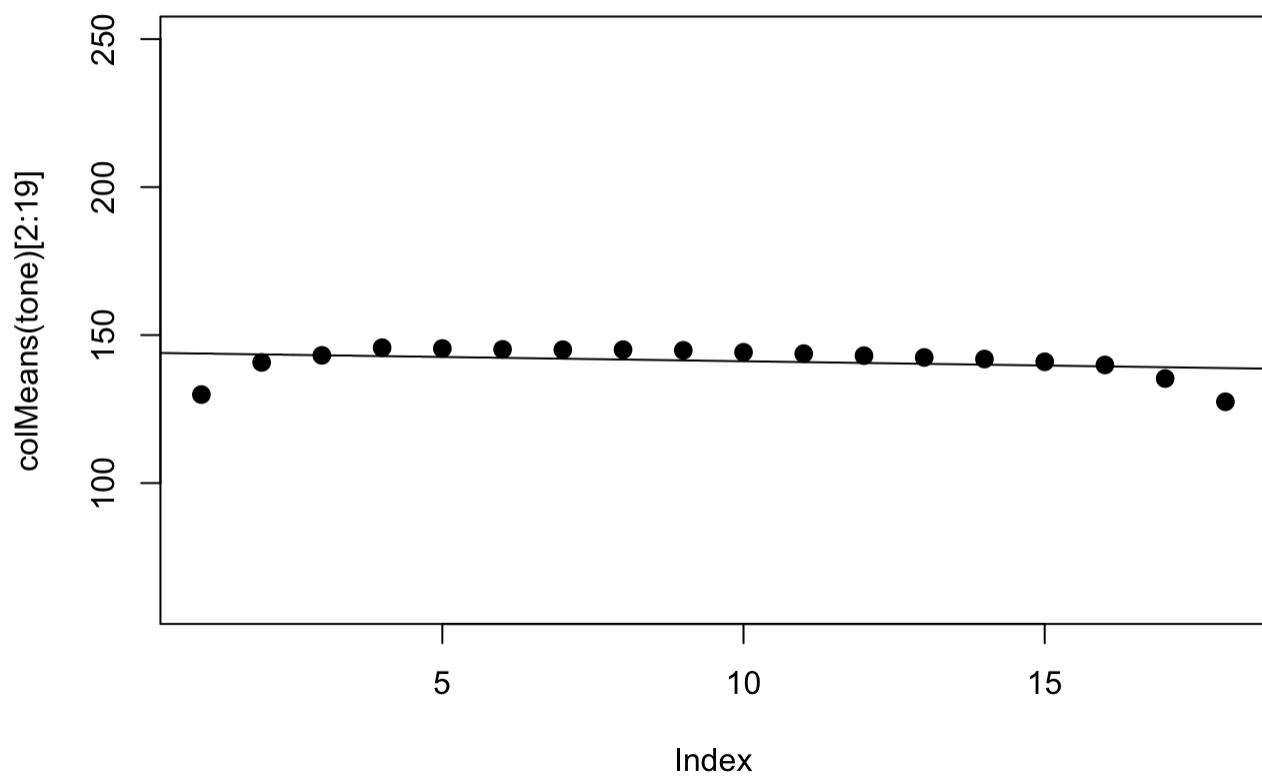
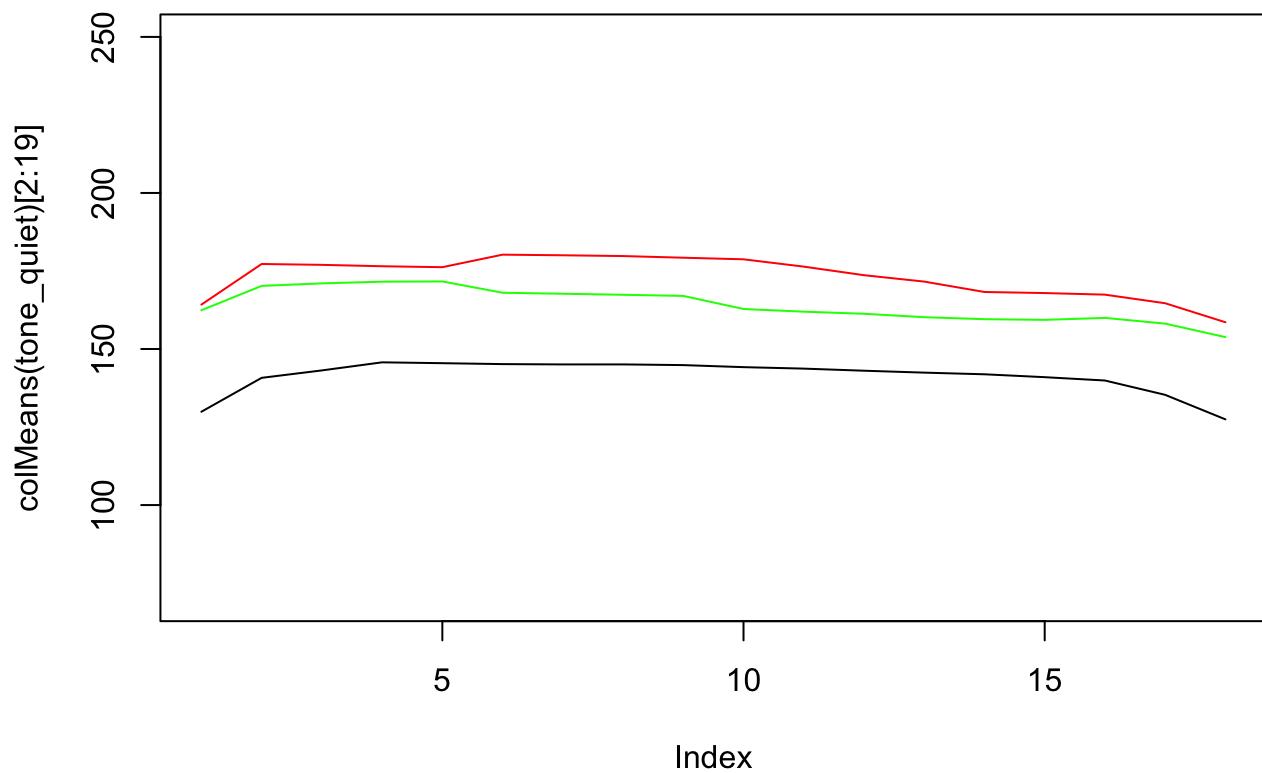



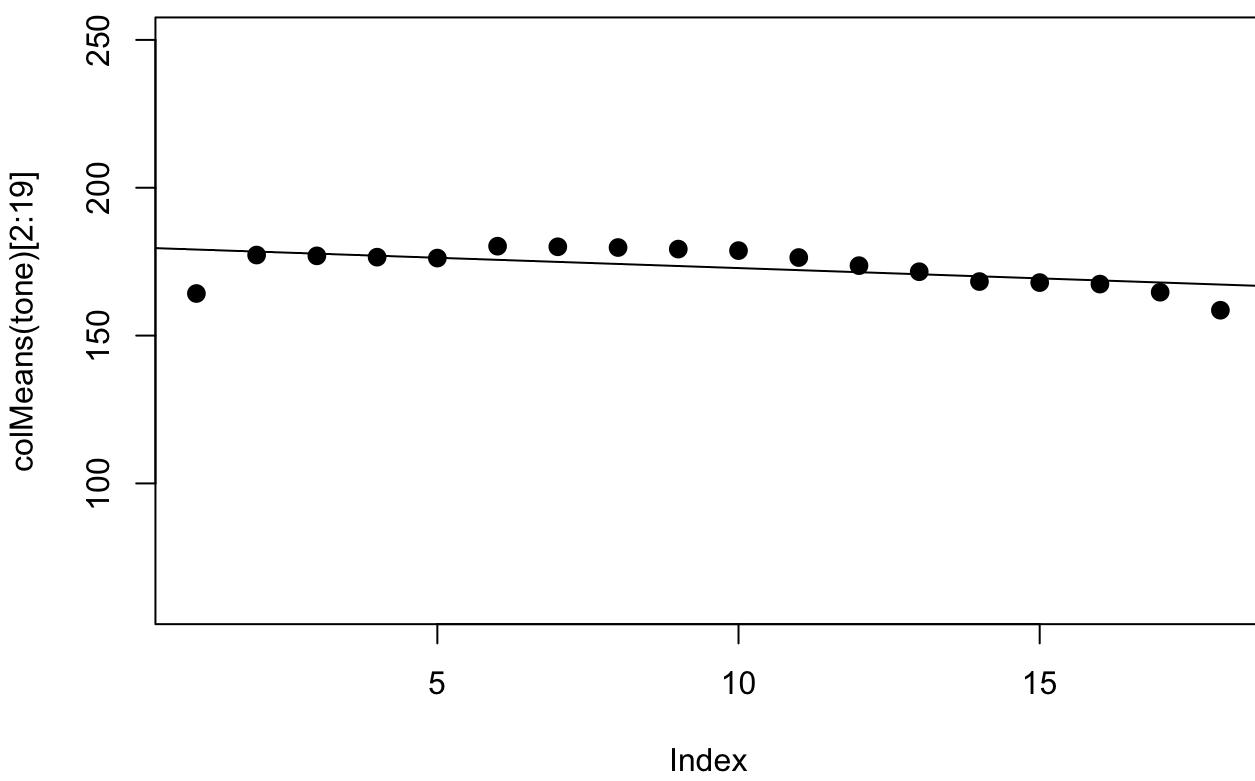
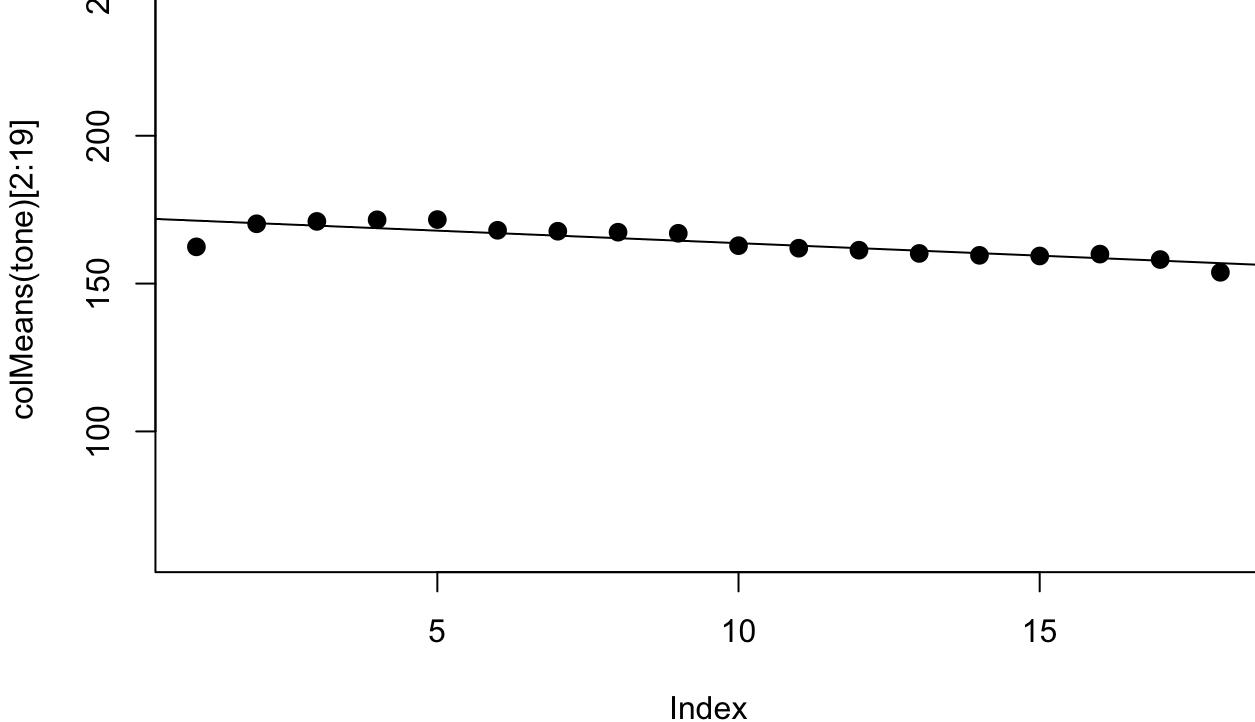


```
## [1] -0.2878660 -0.2424775 -0.3018372
```

Tone A1 on E

```
plot_f0_by_vowels("A1", "E")
```

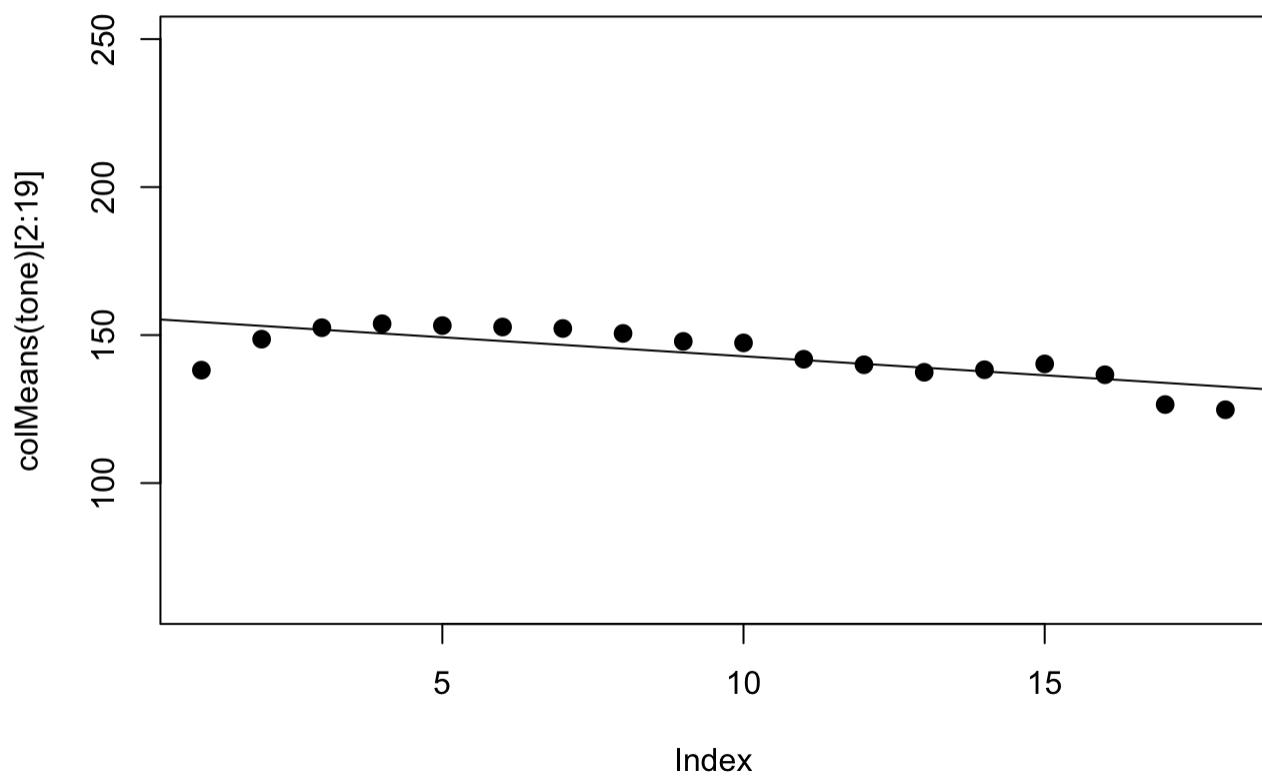
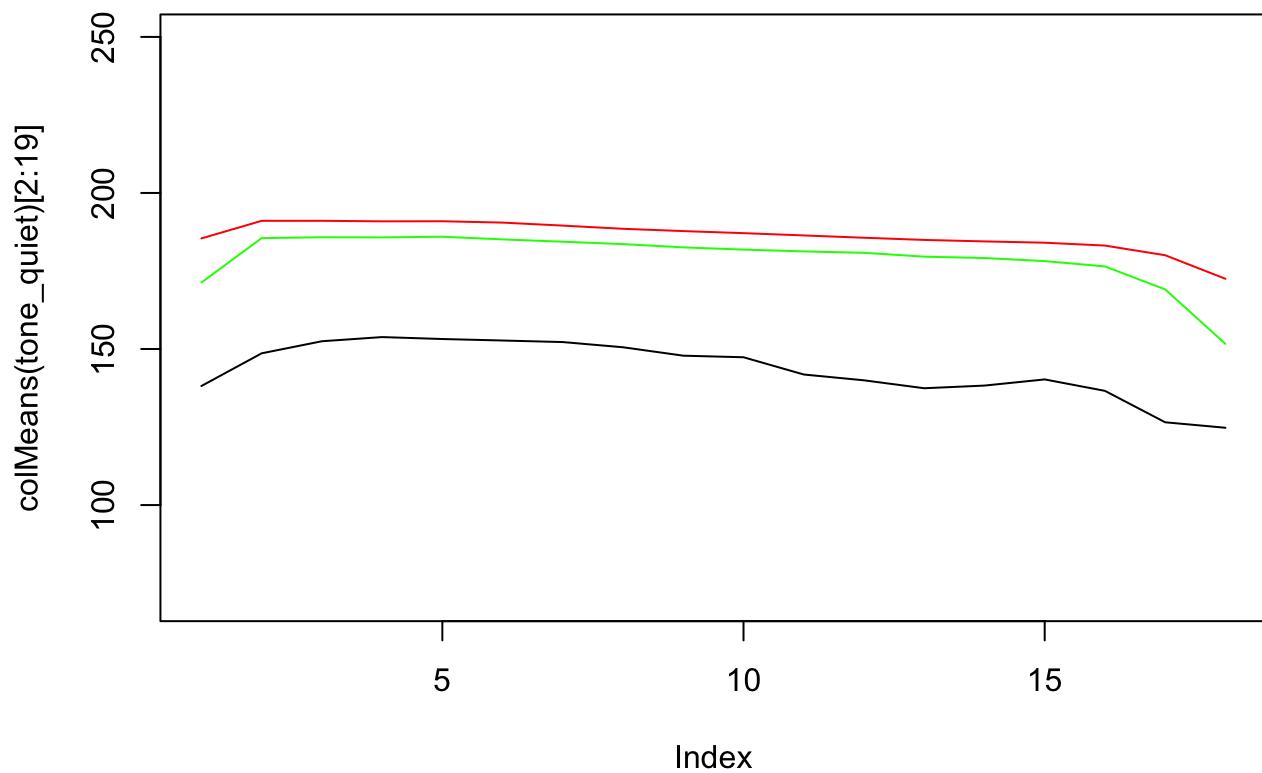



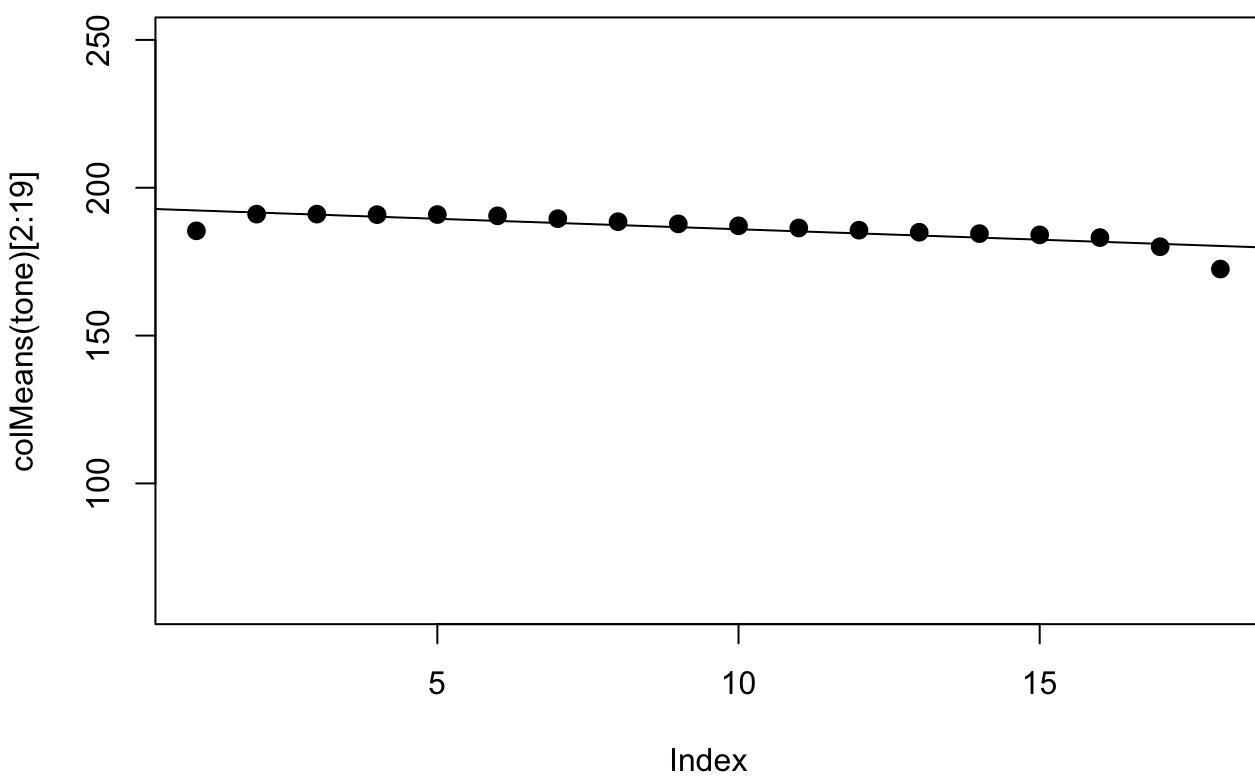
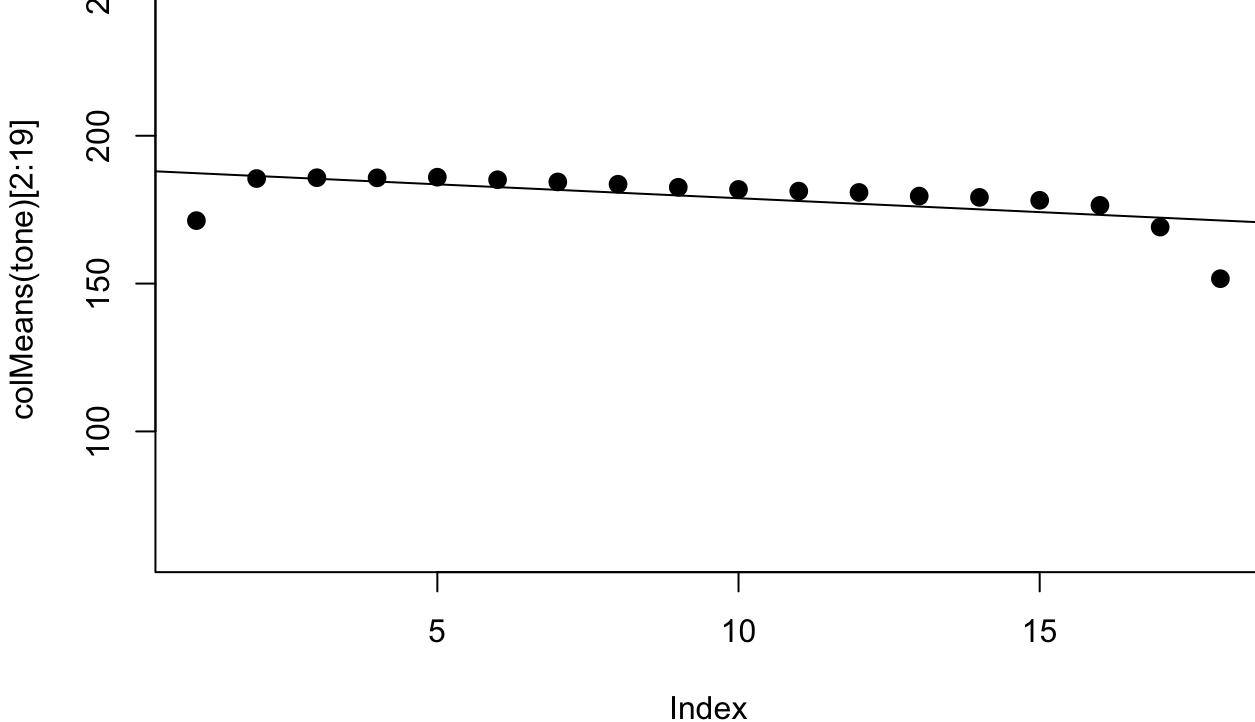


```
## [1] -0.2925254 -0.8486117 -0.6995807
```

Tone A1 on U

```
plot_f0_by_vowels("A1", "U")
```

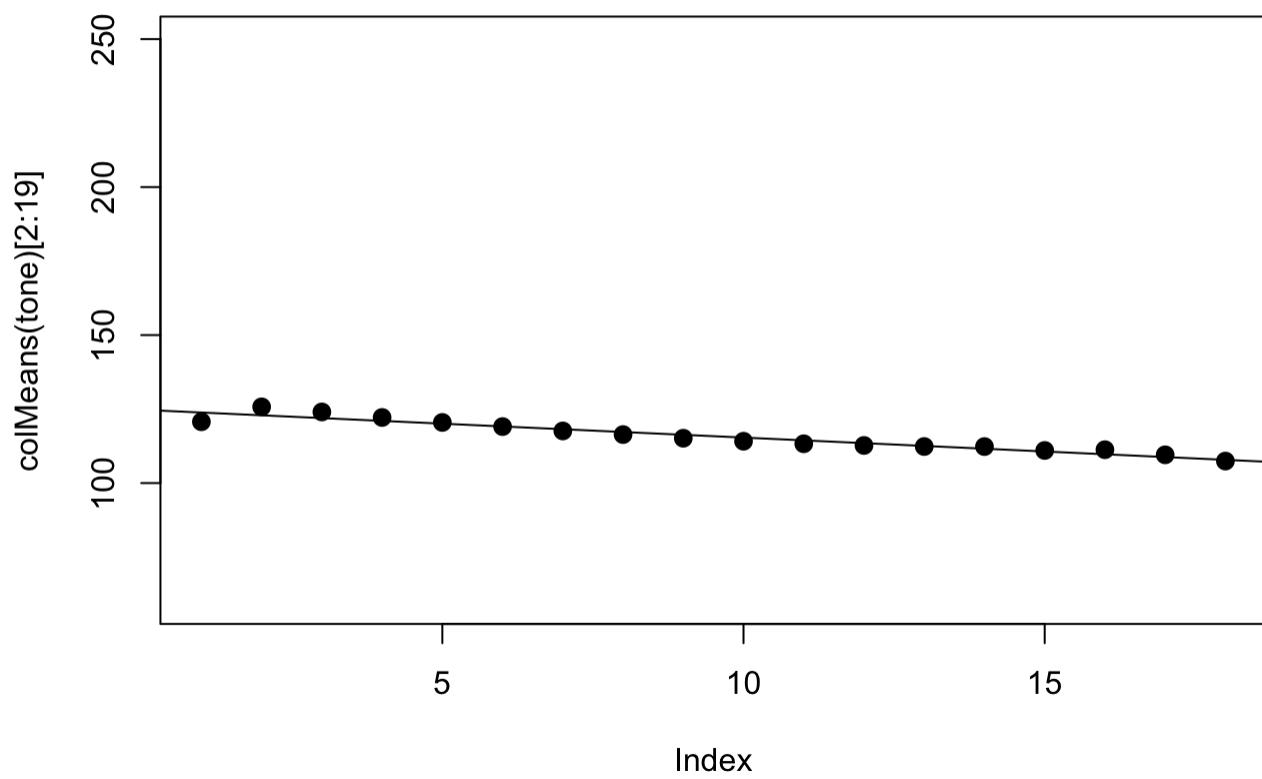
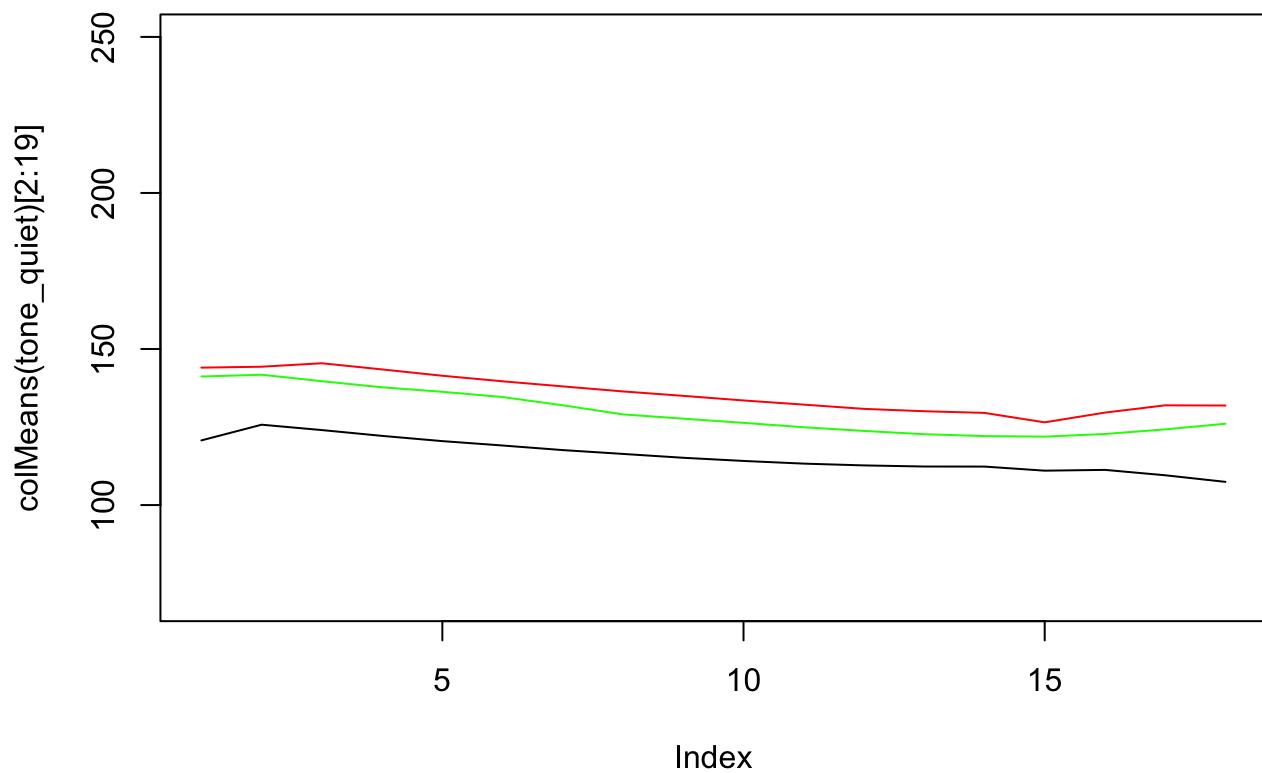



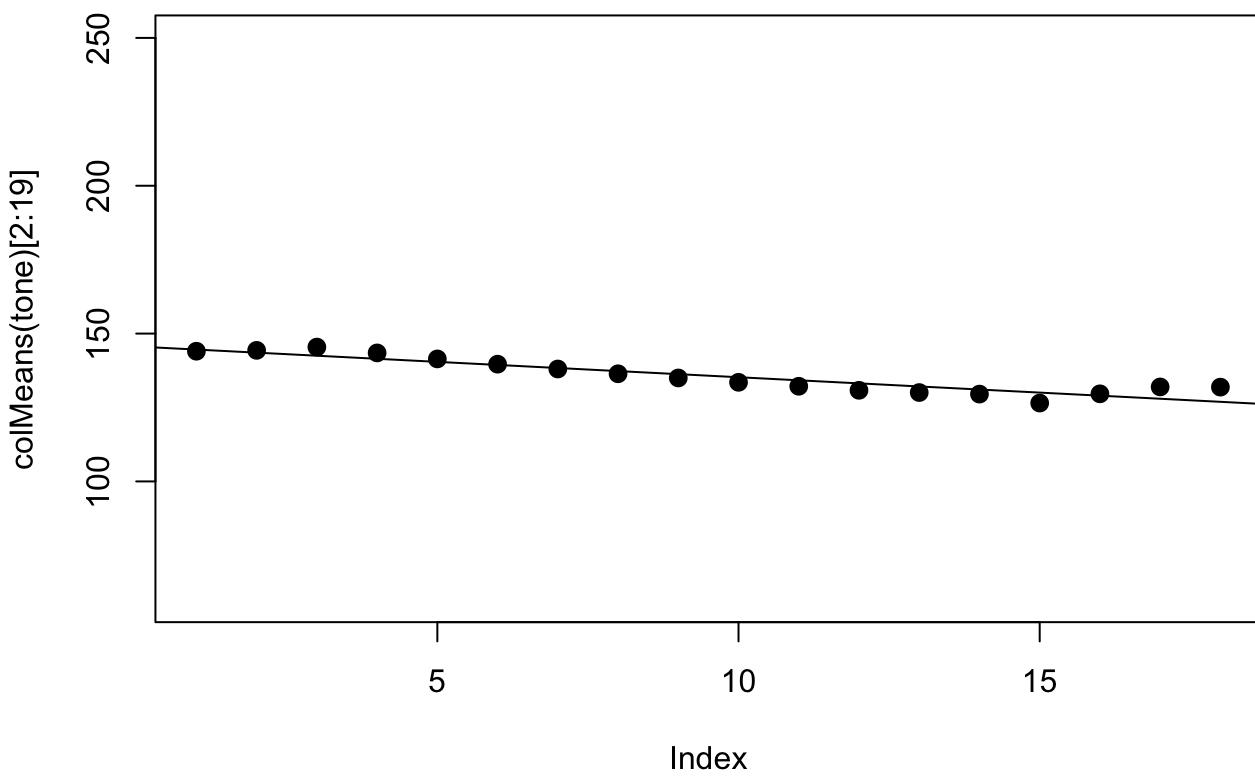
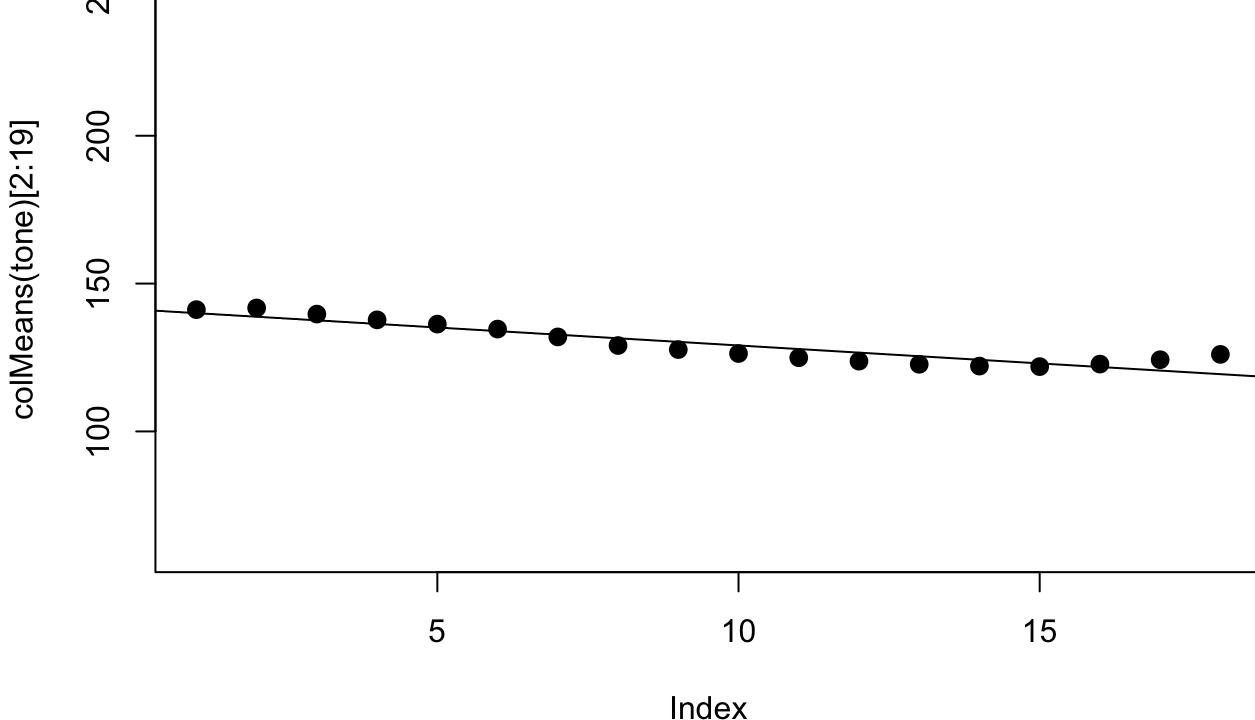


```
## [1] -1.2836820 -0.9401882 -0.7090698
```

Tone A2 on A

```
plot_f0_by_vowels("A2", "A")
```

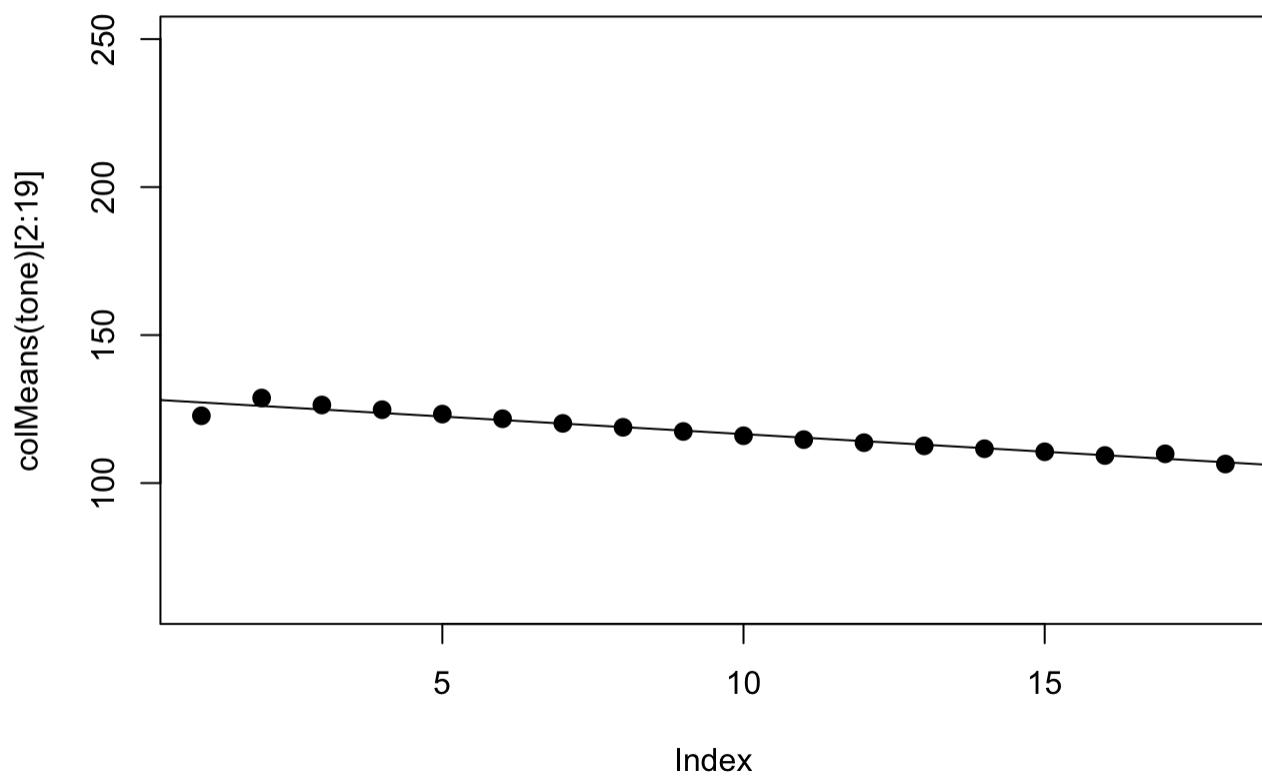
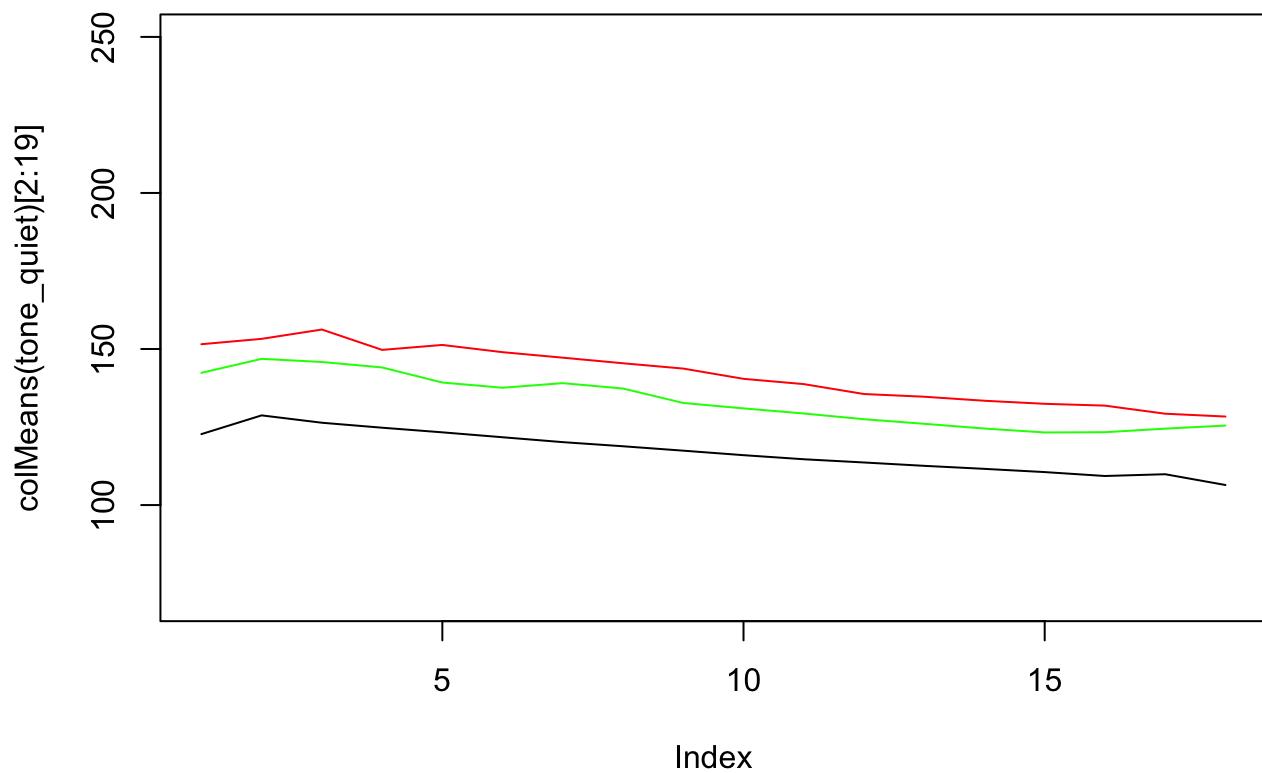



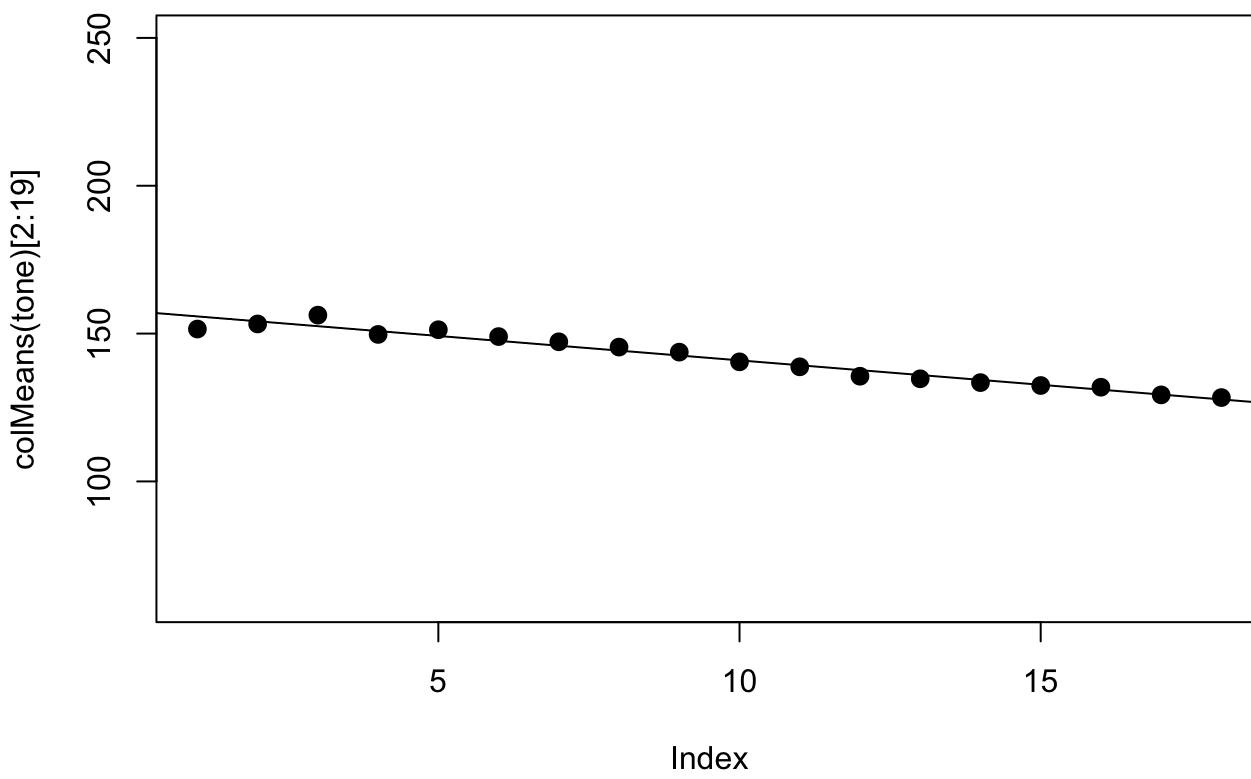
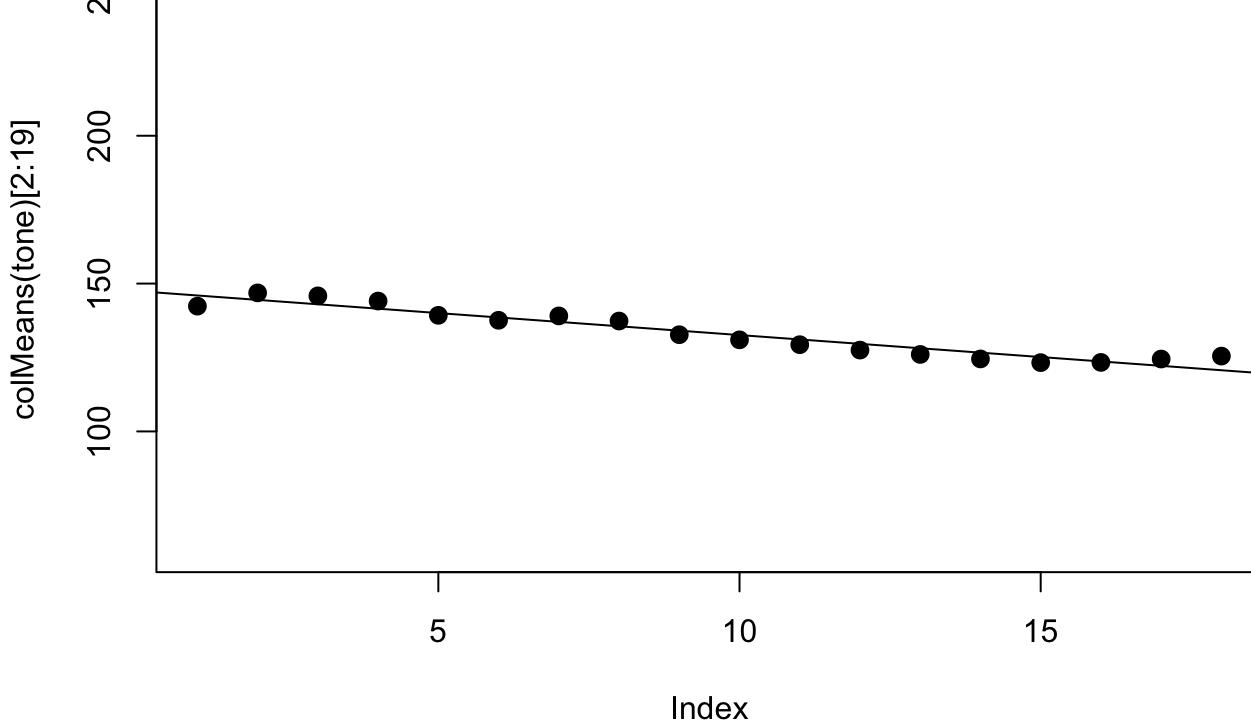


```
## [1] -0.9418938 -1.2171543 -1.0407443
```

Tone A2 on E

```
plot_f0_by_vowels("A2", "E")
```

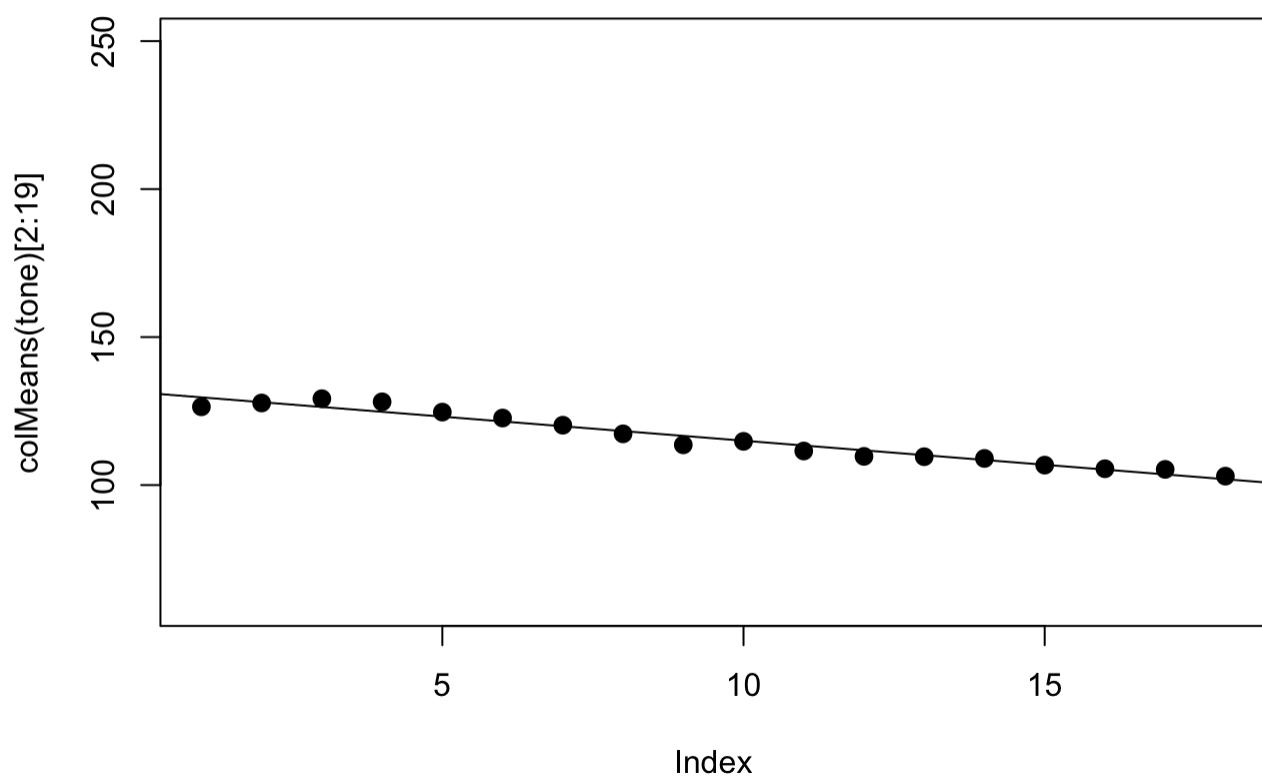
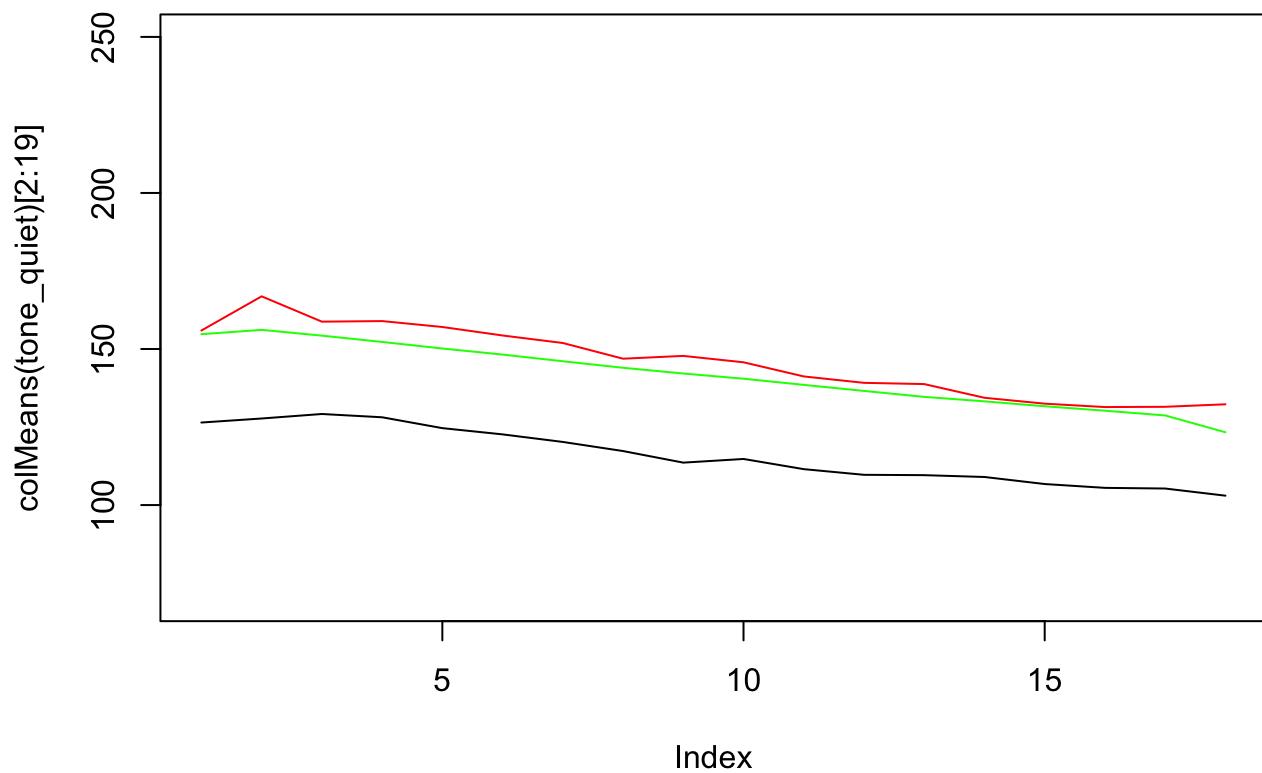



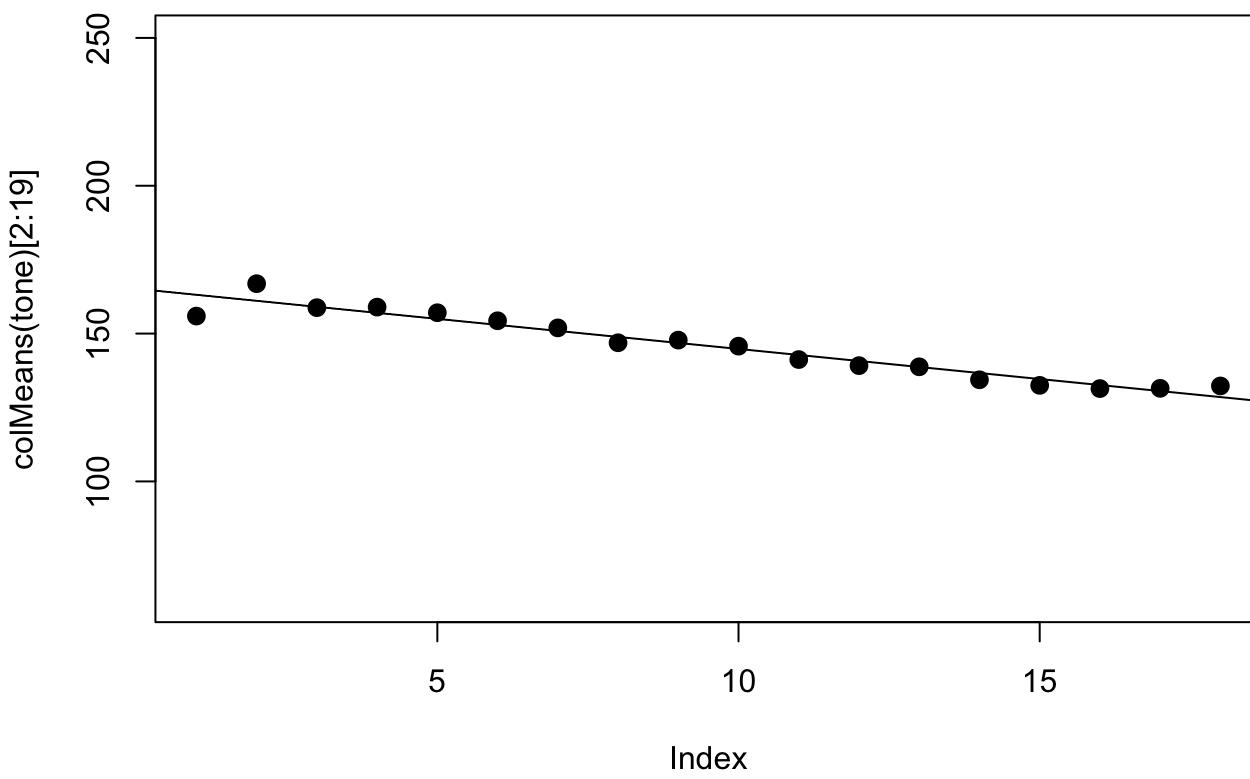
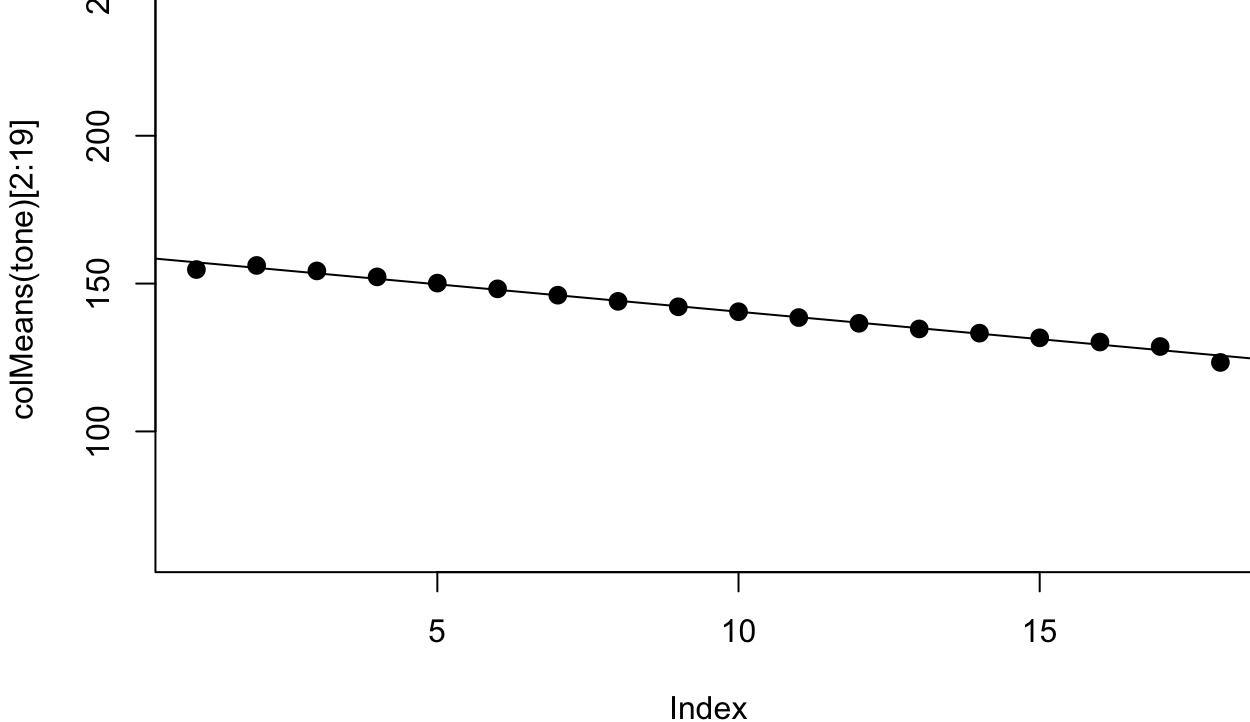


```
## [1] -1.191150 -1.487463 -1.654423
```

Tone A2 on U

```
plot_f0_by_vowels("A2", "U")
```

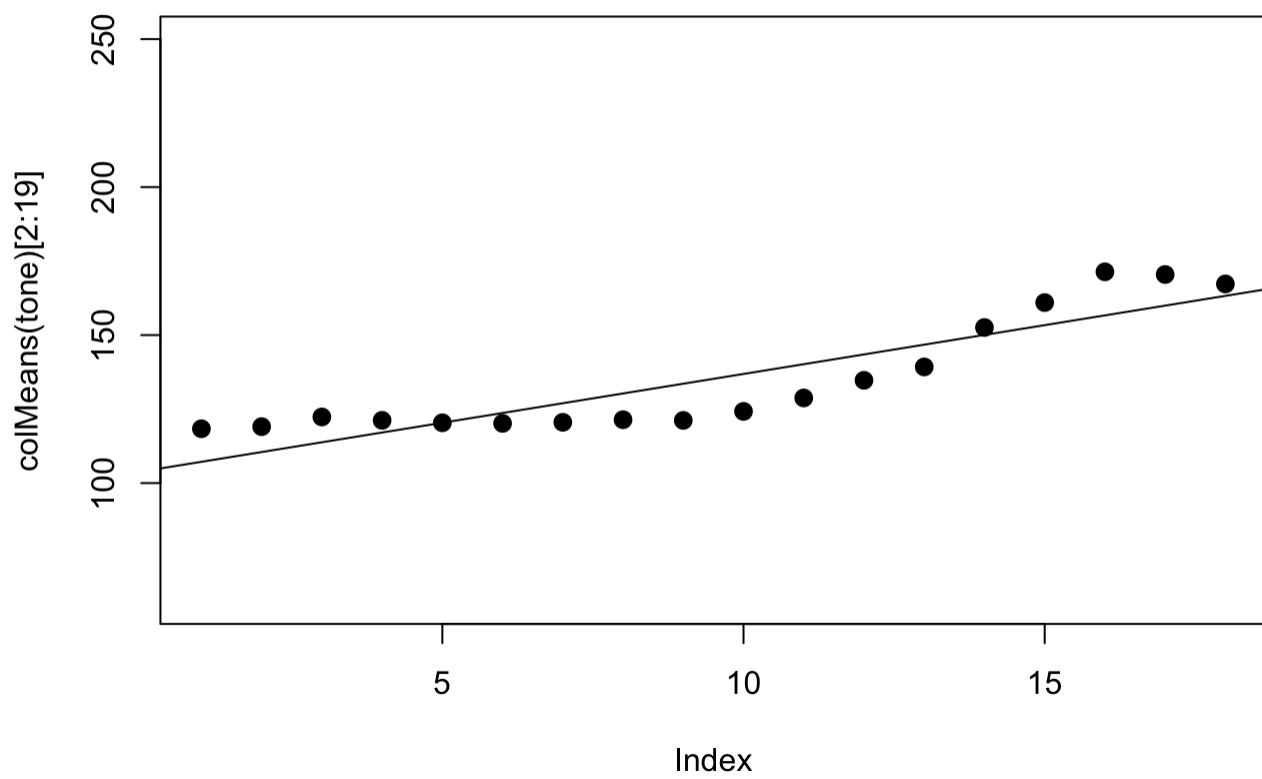
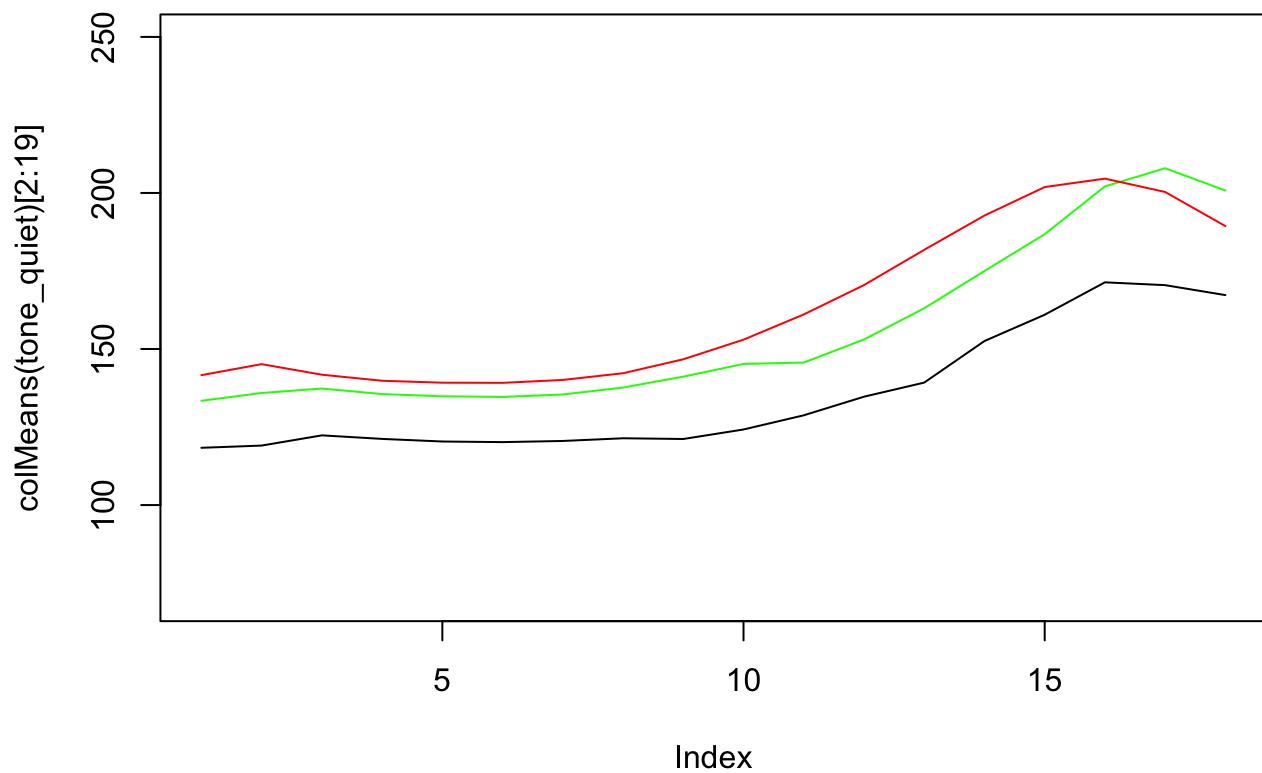



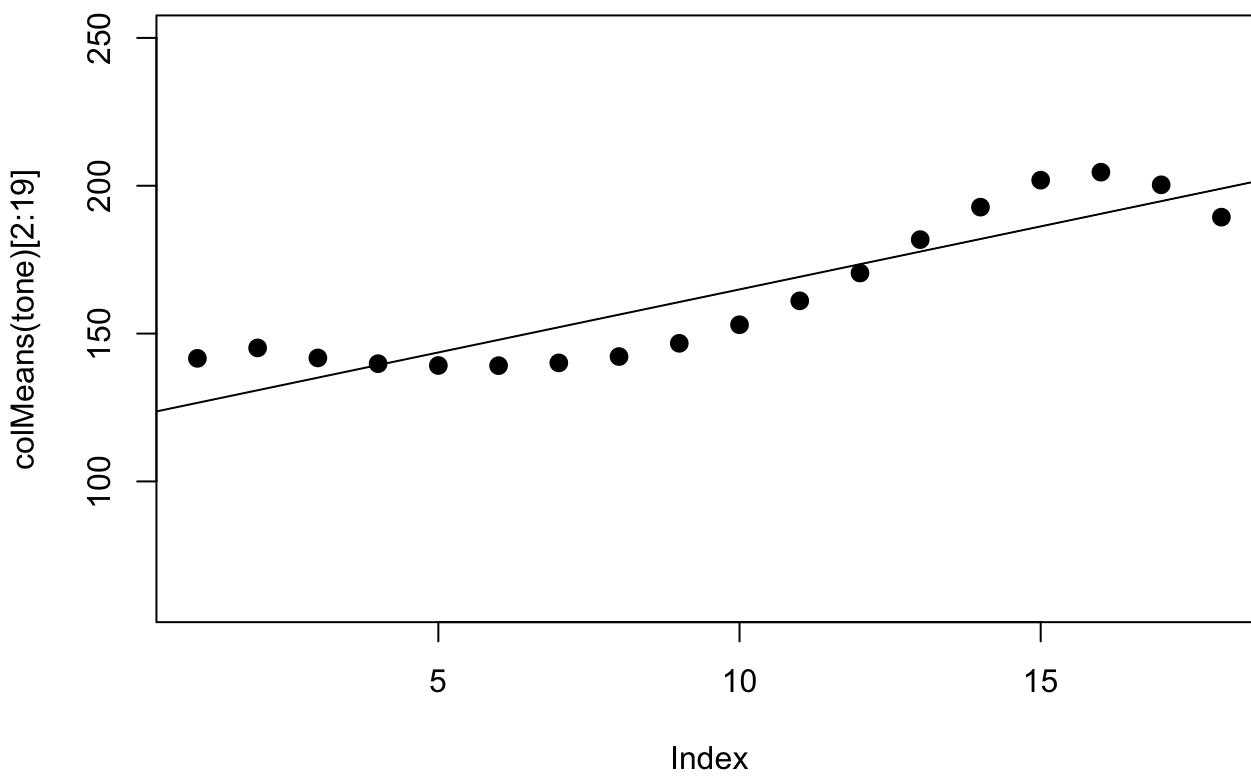
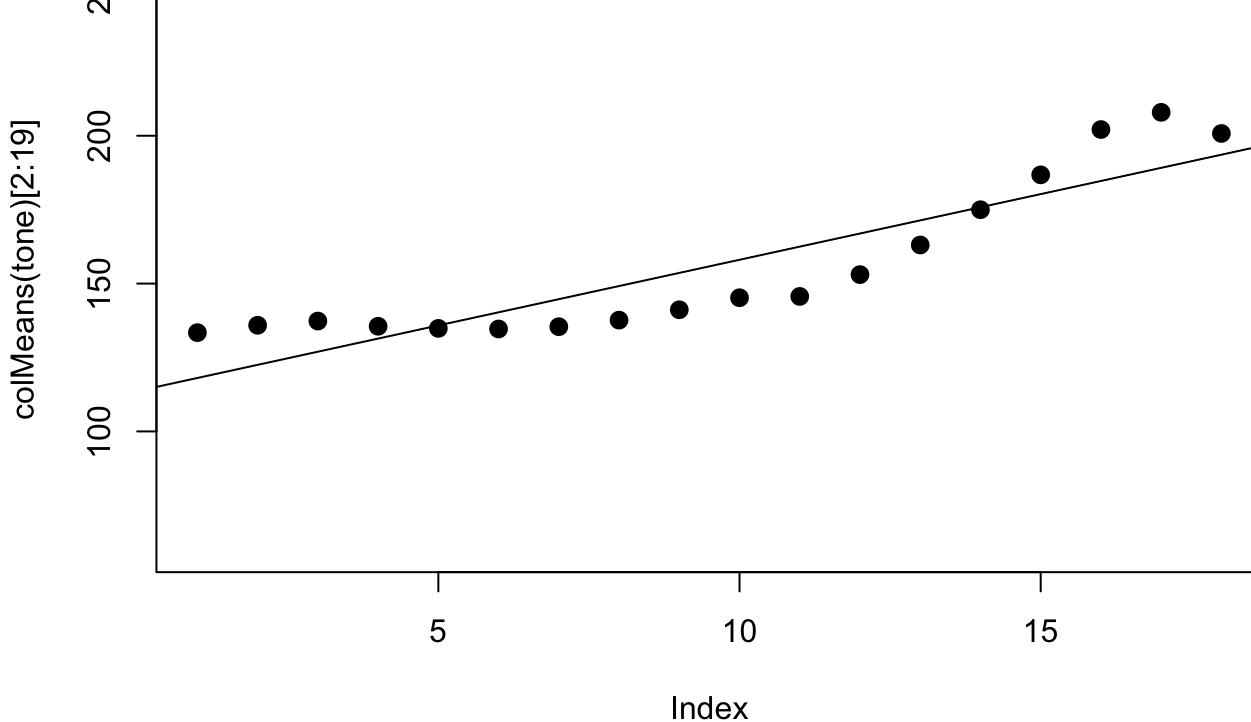


```
## [1] -1.629644 -1.855327 -2.038729
```

Tone B1 on A

```
plot_f0_by_vowels("B1", "A")
```

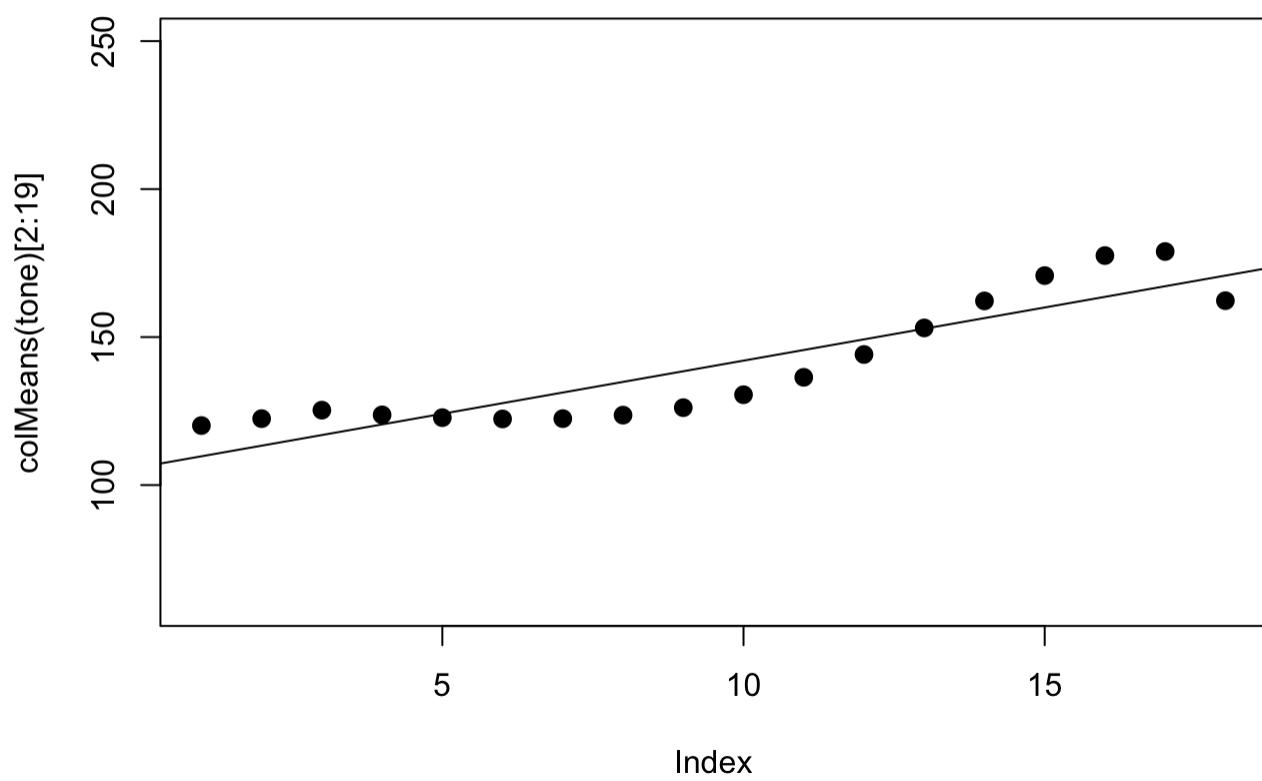
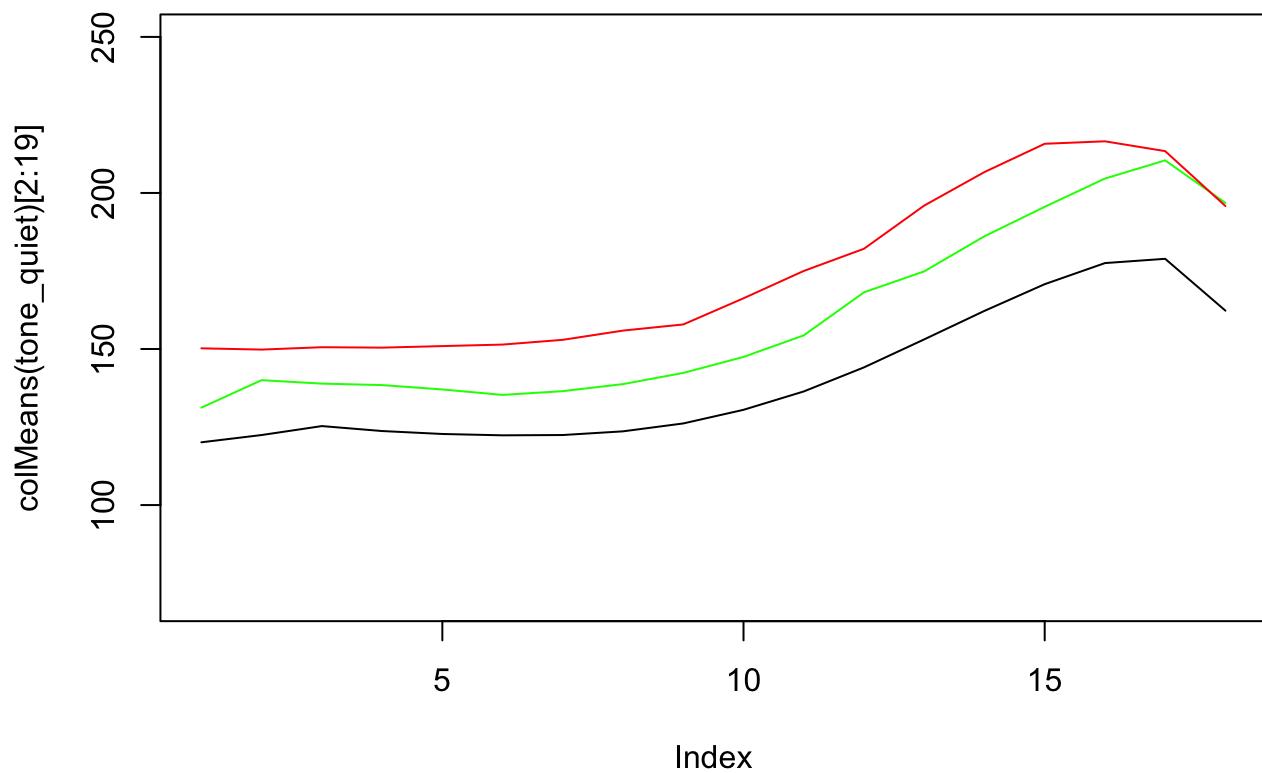



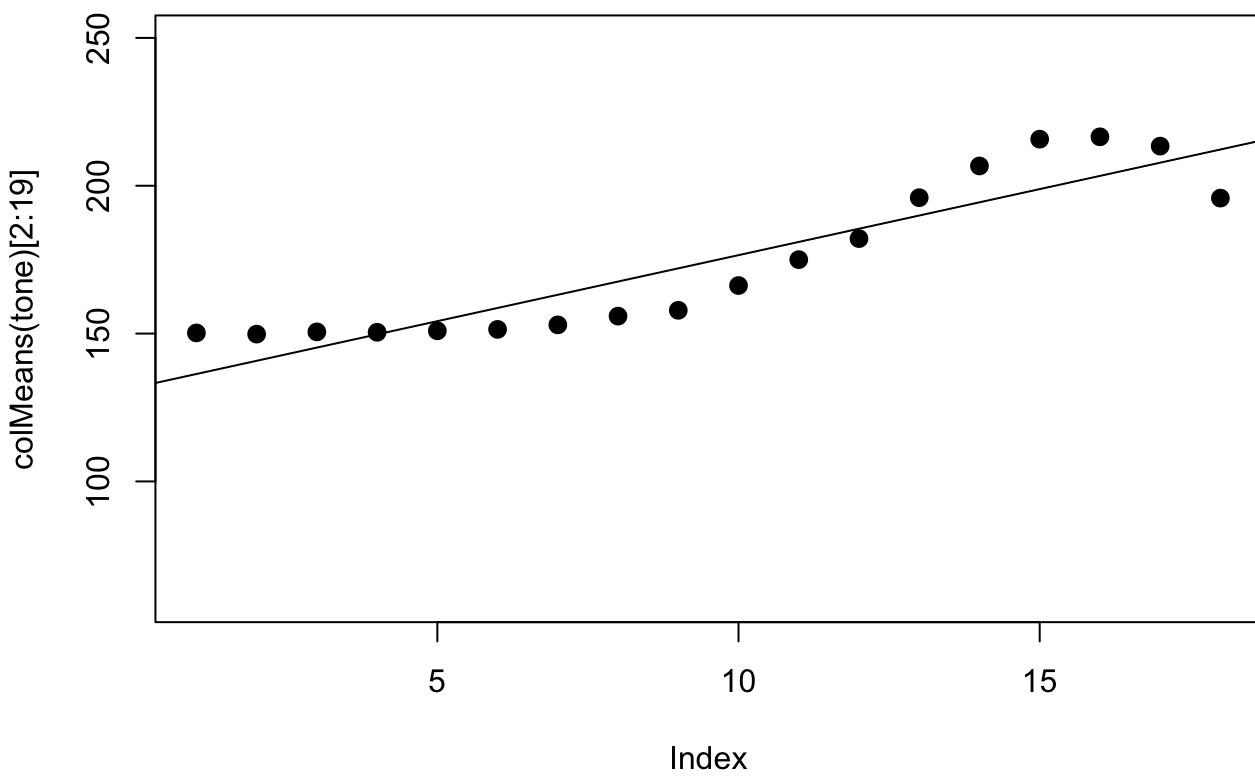
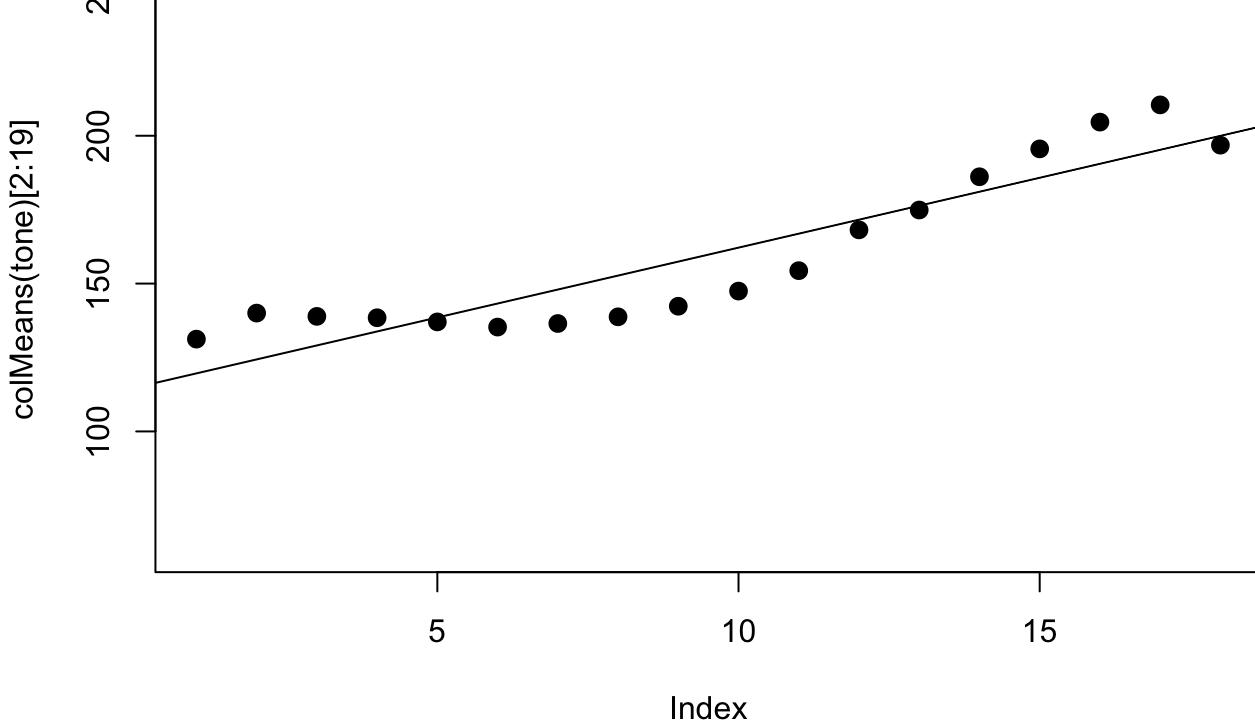


```
## [1] 3.299303 4.444733 4.265991
```

Tone B1 on E

```
plot_f0_by_vowels("B1", "E")
```

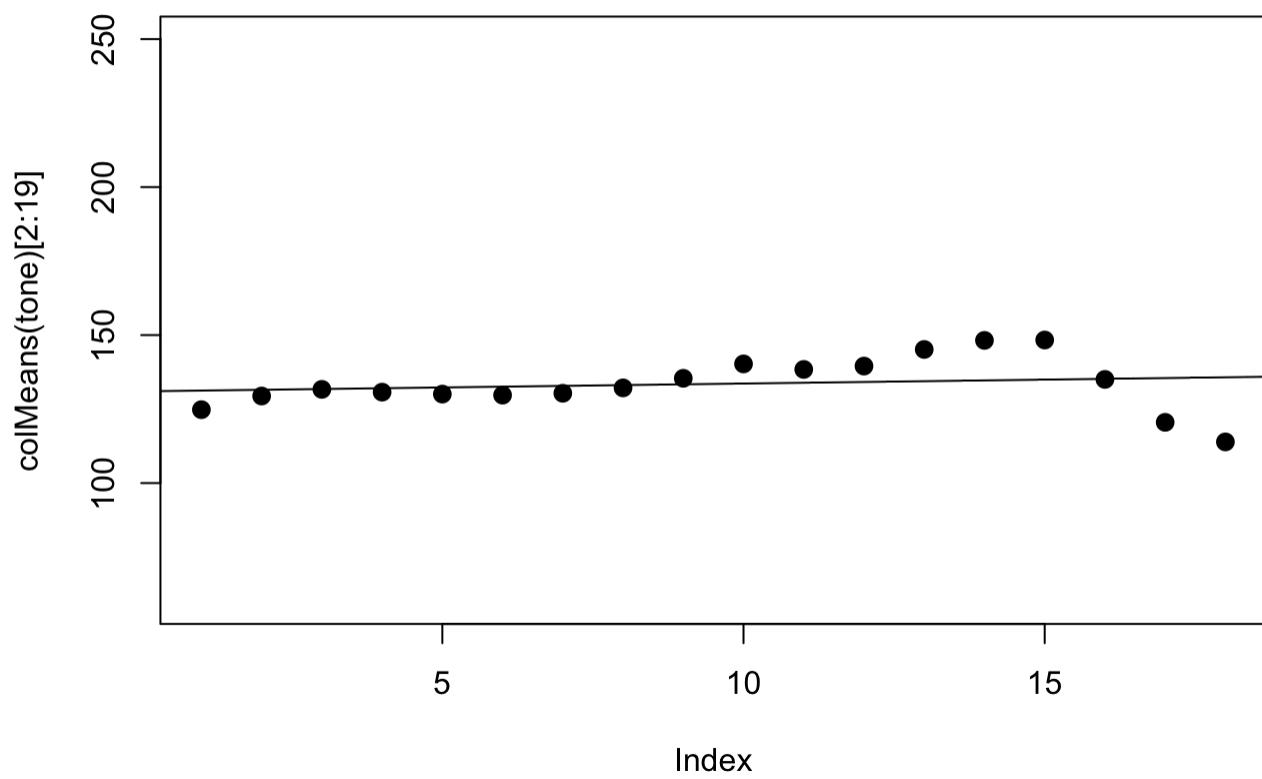
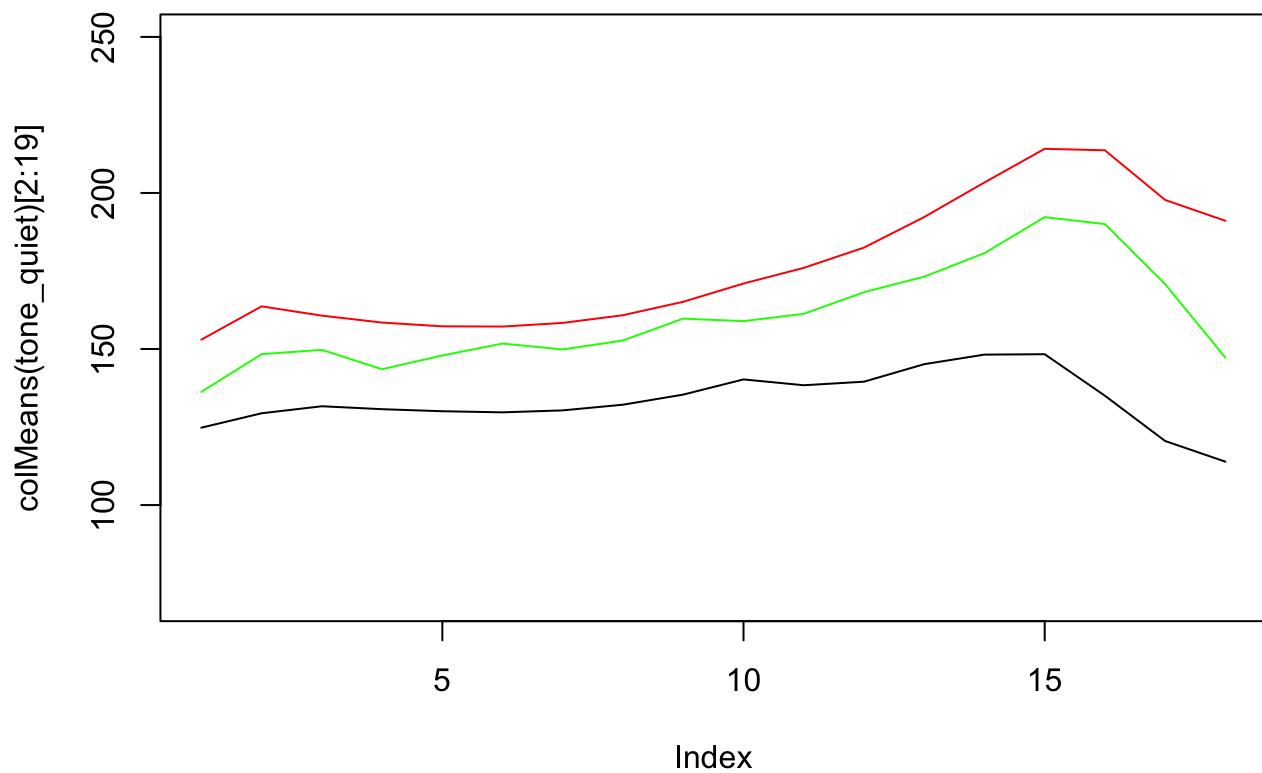



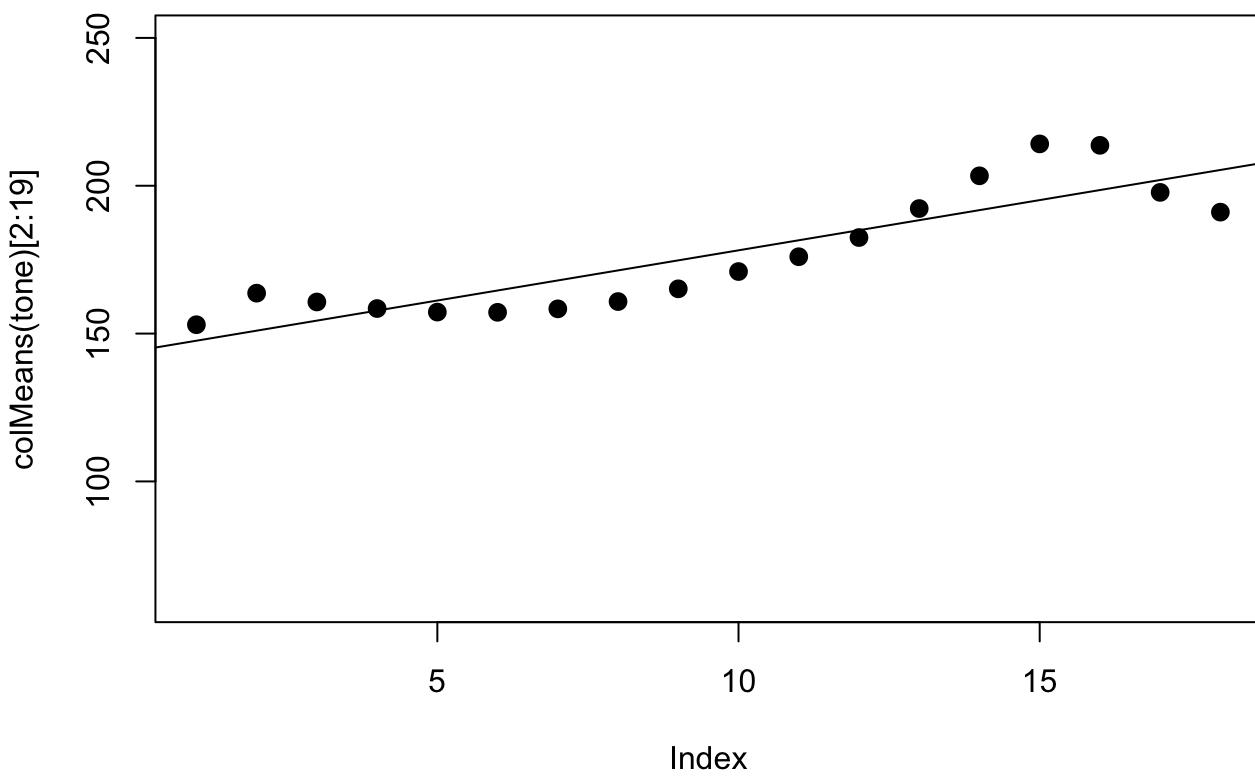
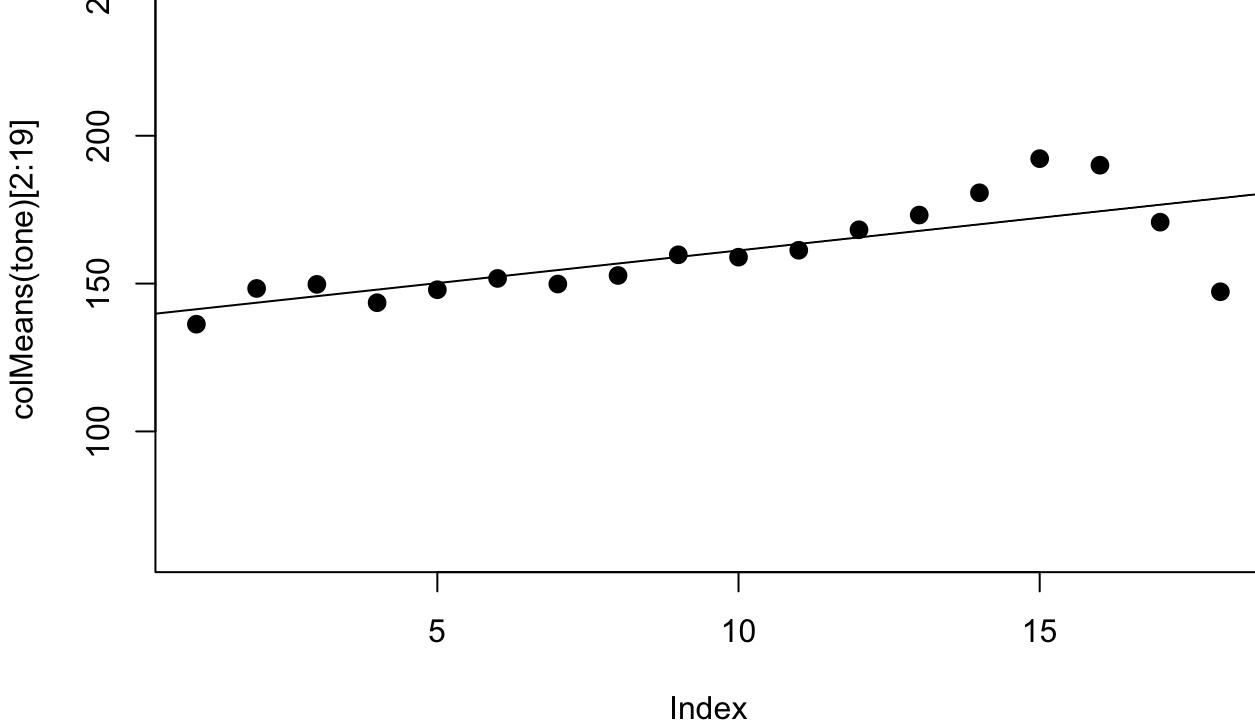


```
## [1] 3.593280 4.729655 4.469133
```

Tone B1 on U

```
plot_f0_by_vowels("B1", "U")
```

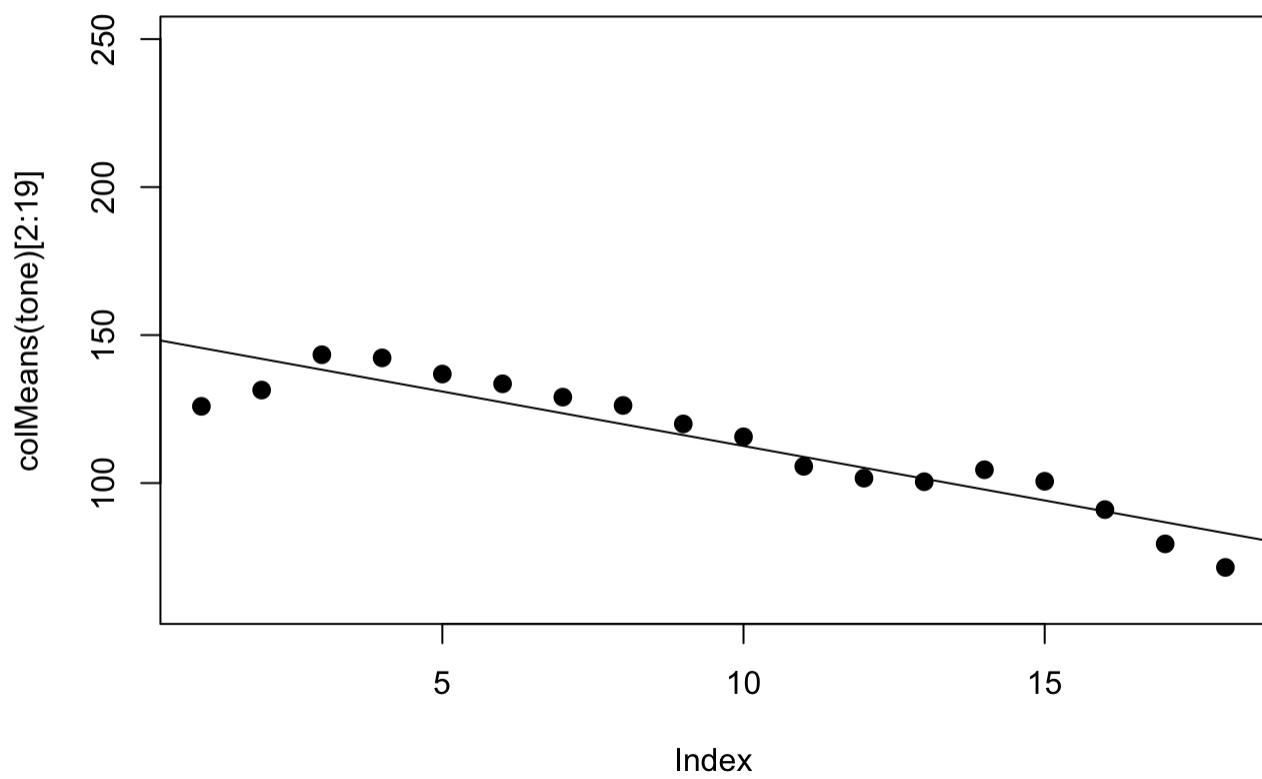
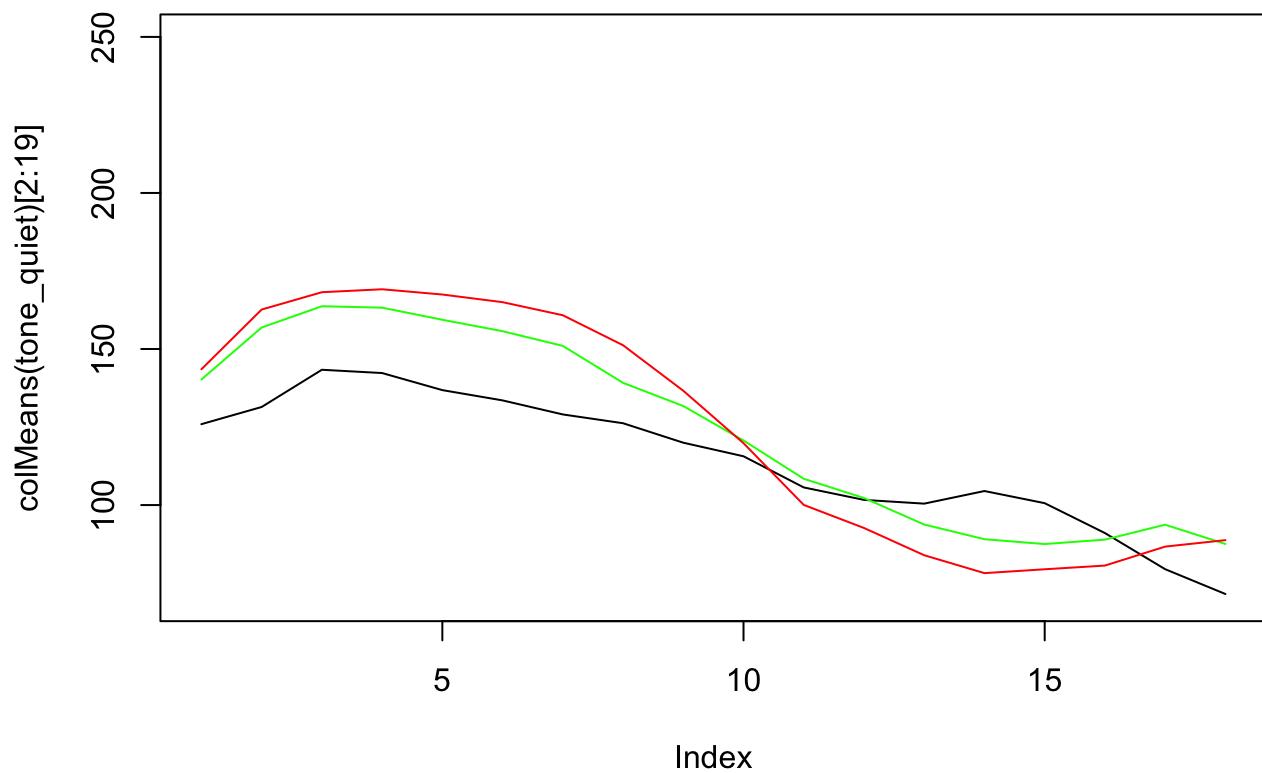



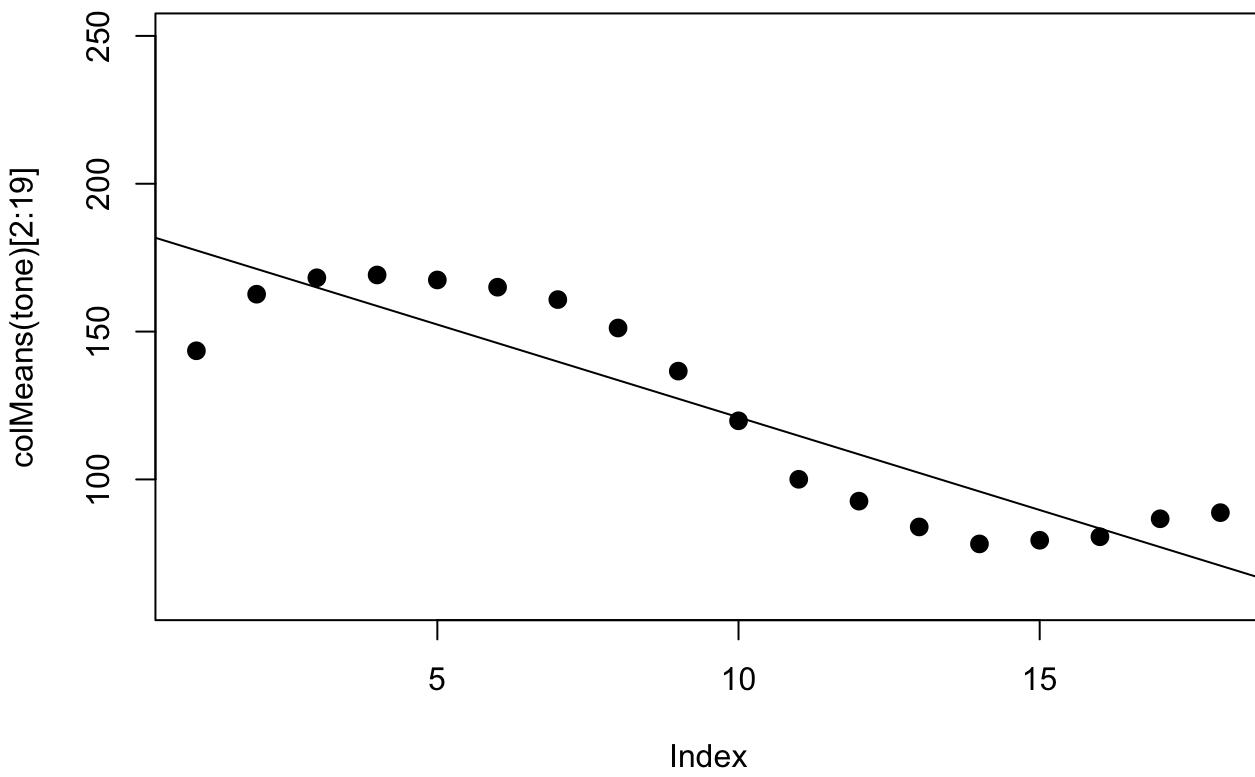
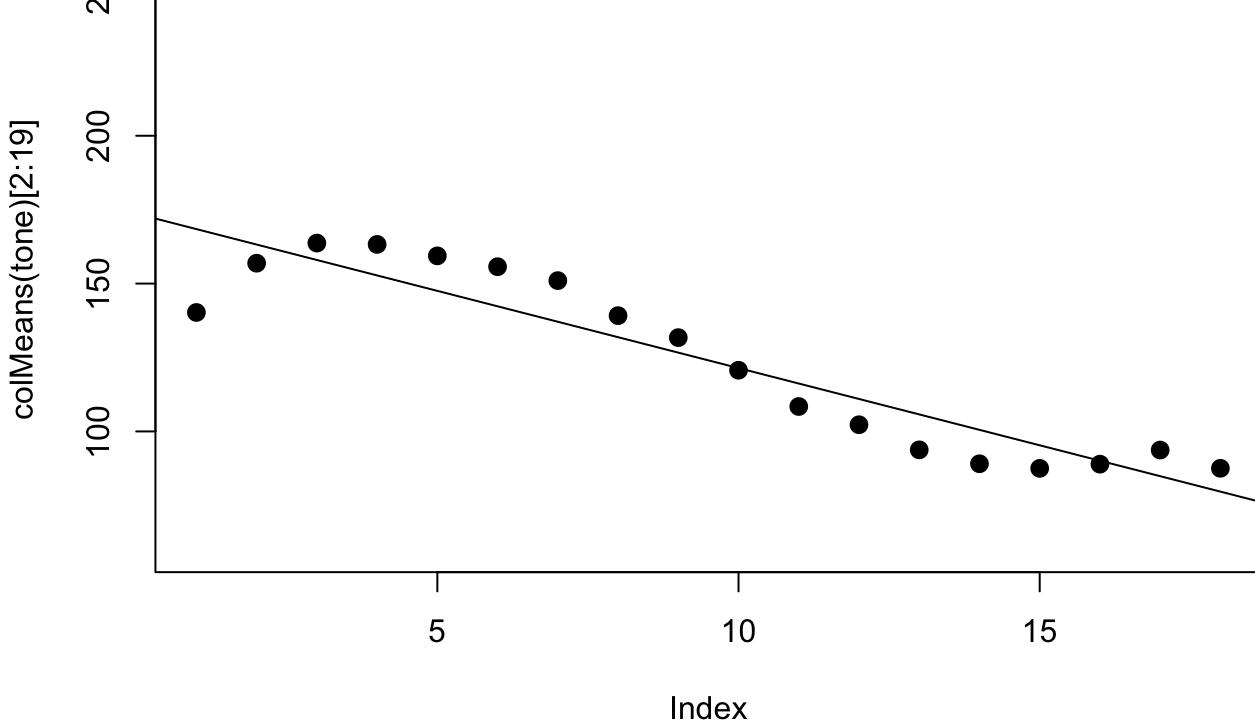


```
## [1] 0.2679428 2.2130707 3.3975057
```

Tone B2 on A

```
plot_f0_by_vowels("B2", "A")
```

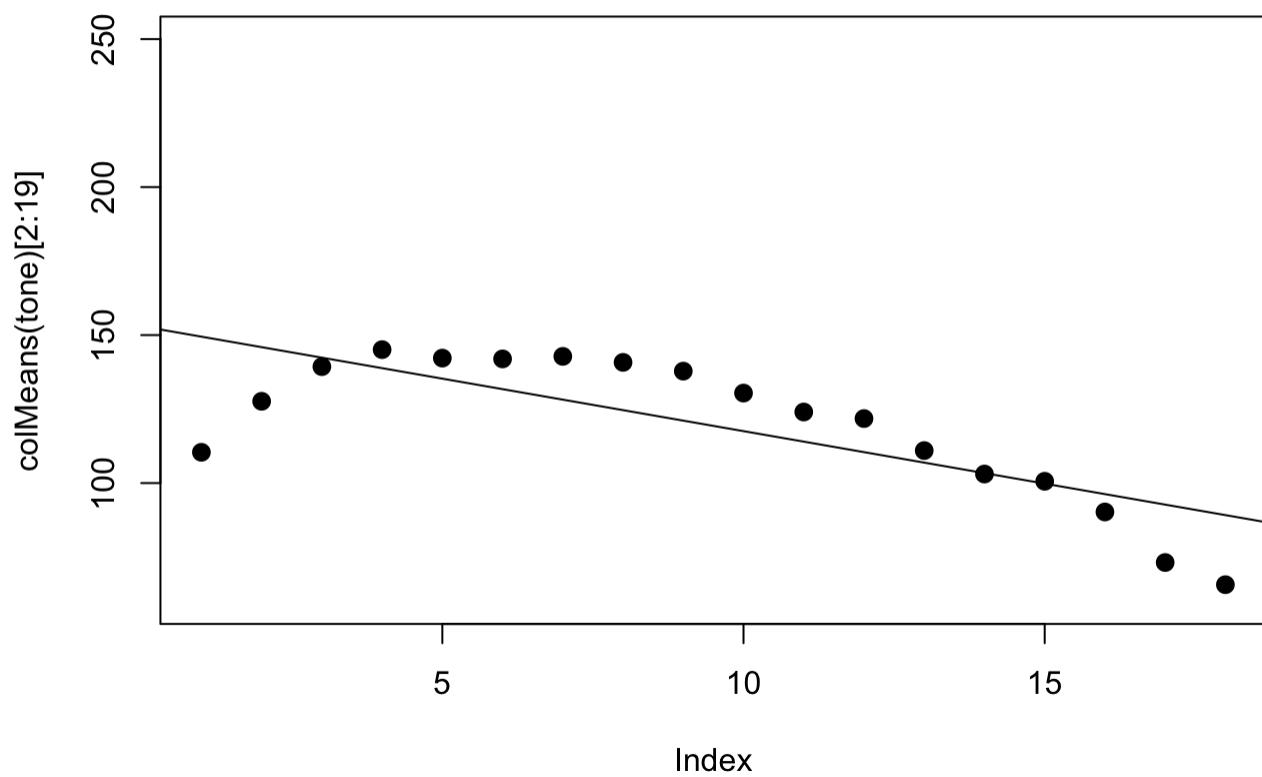
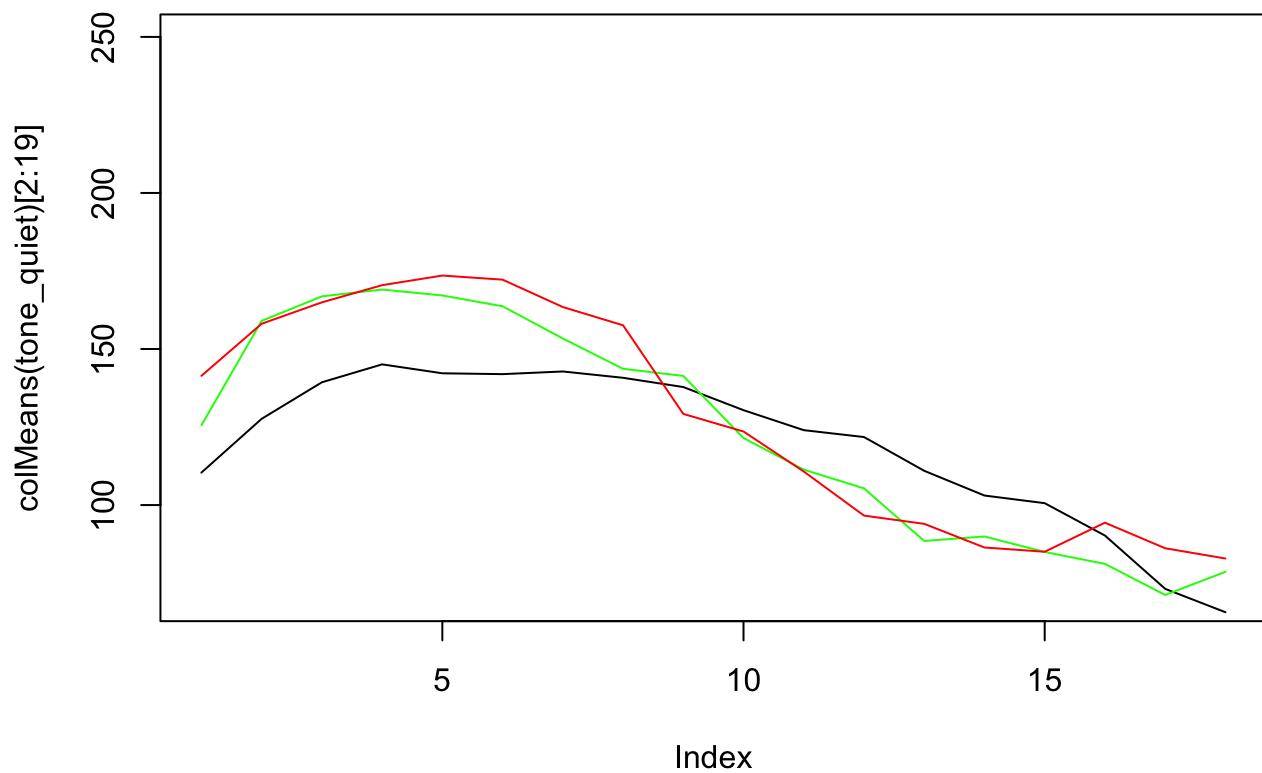



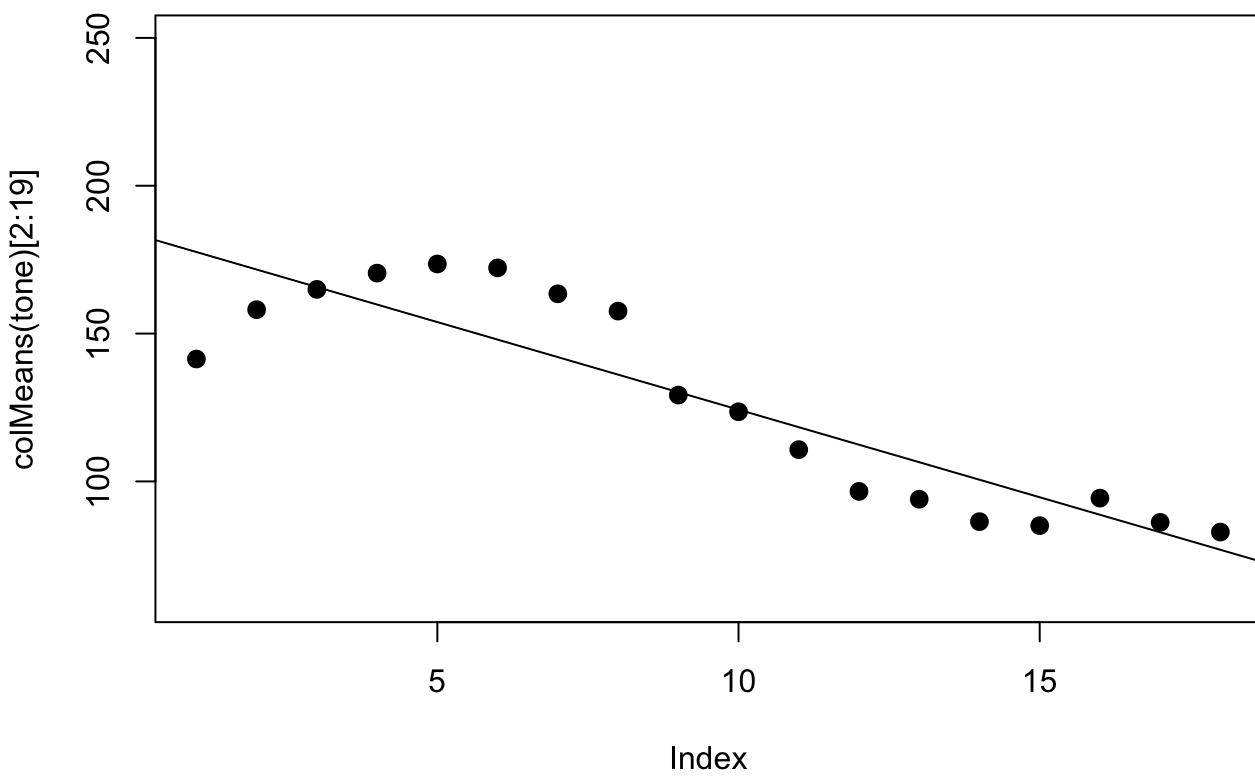
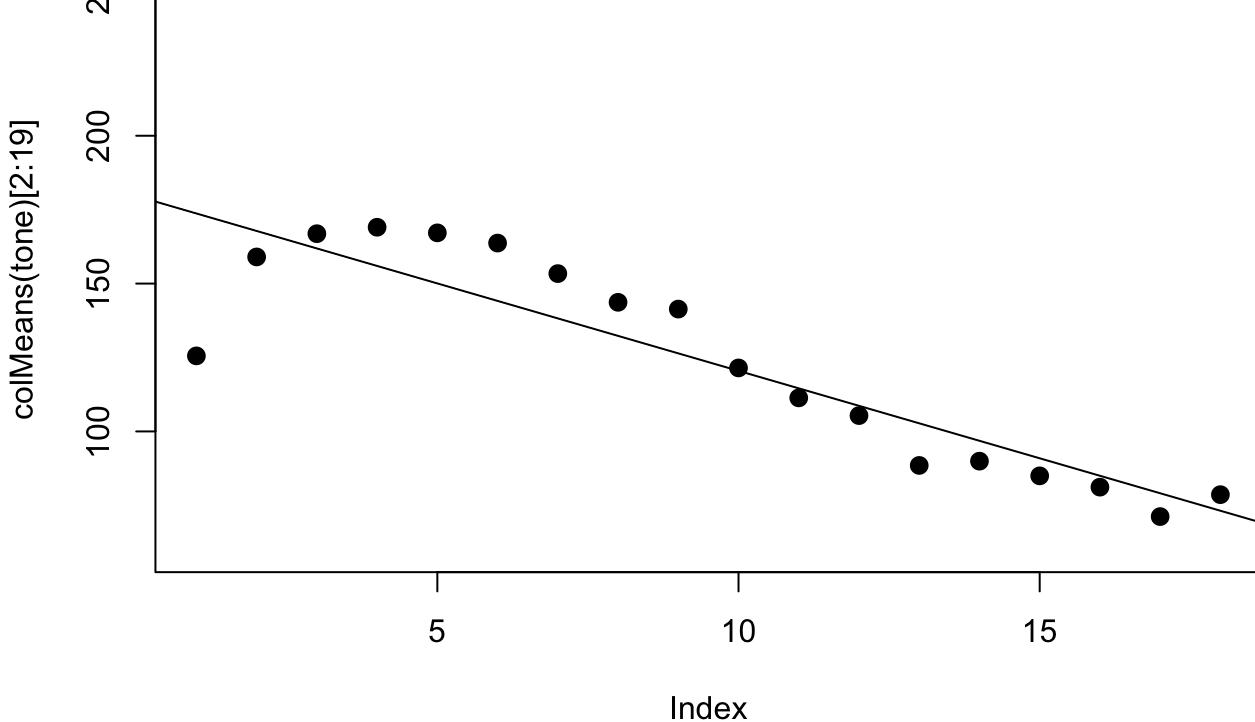


```
## [1] -3.683201 -5.224309 -6.272141
```

Tone B2 on E

```
plot_f0_by_vowels("B2", "E")
```

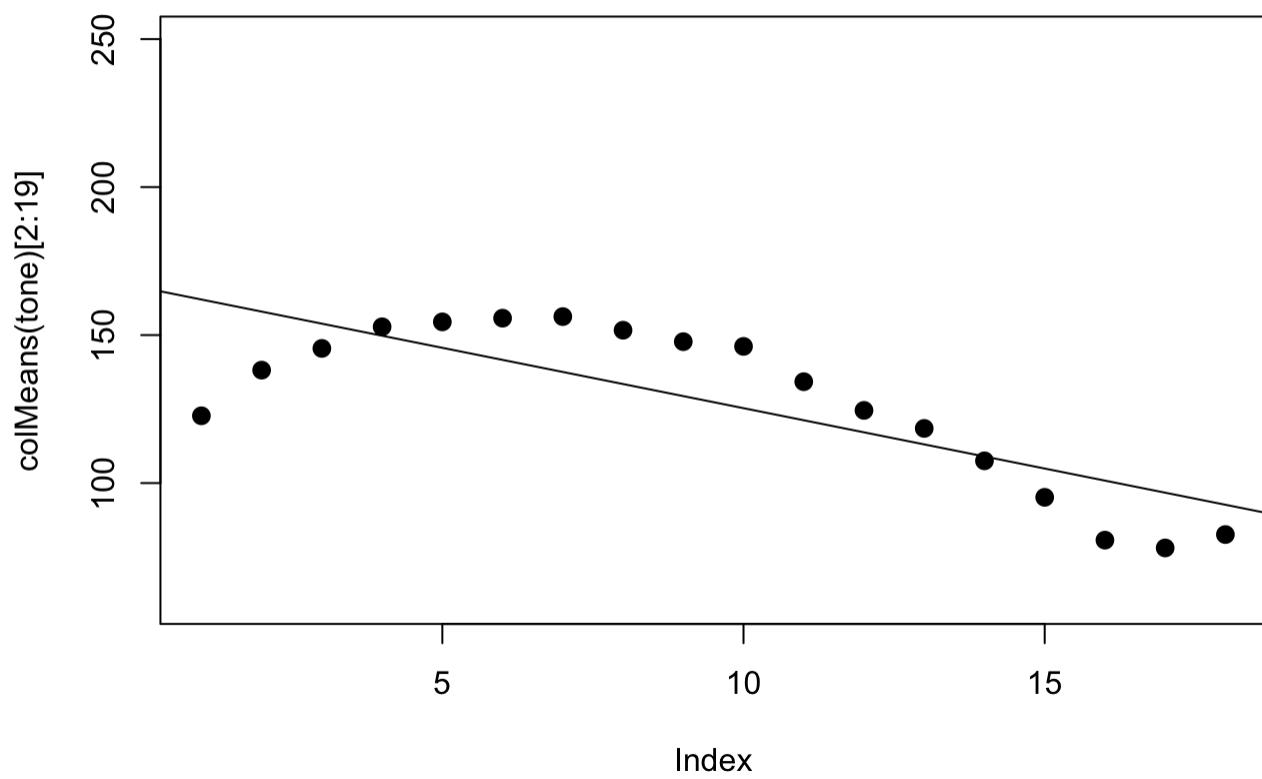
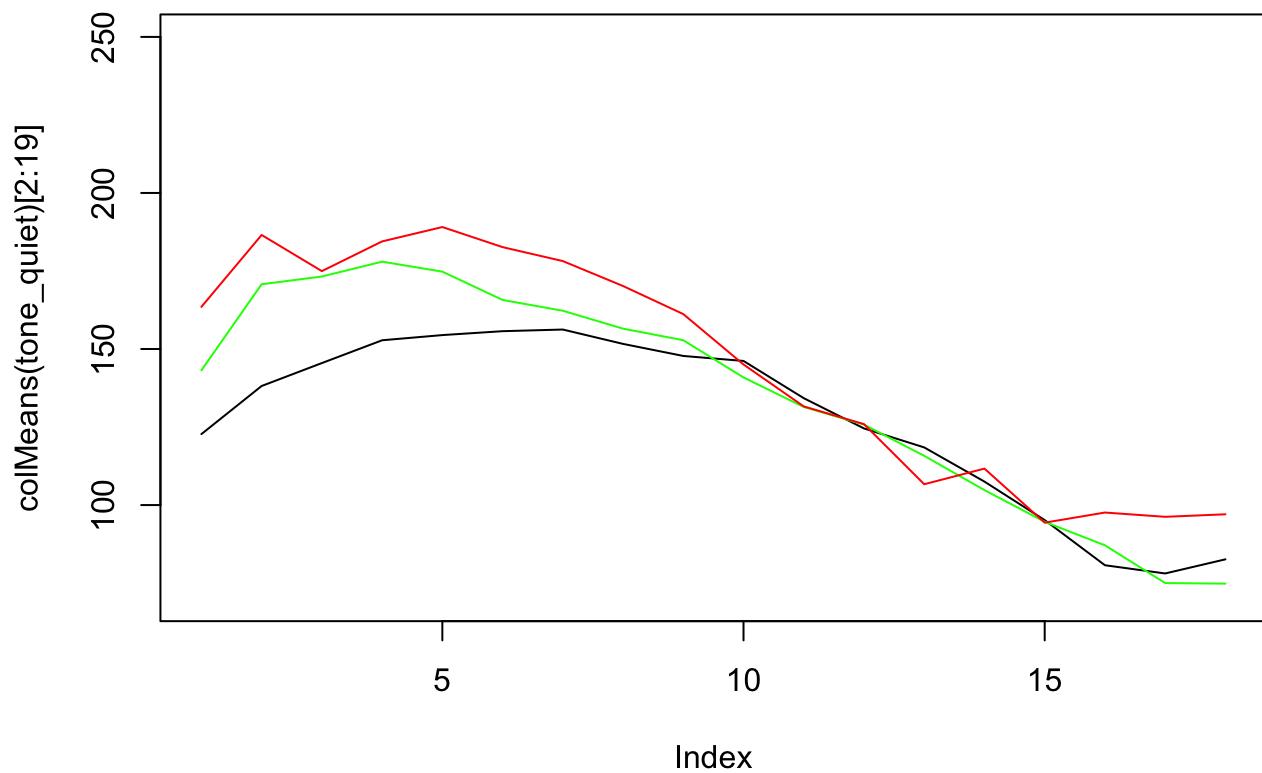



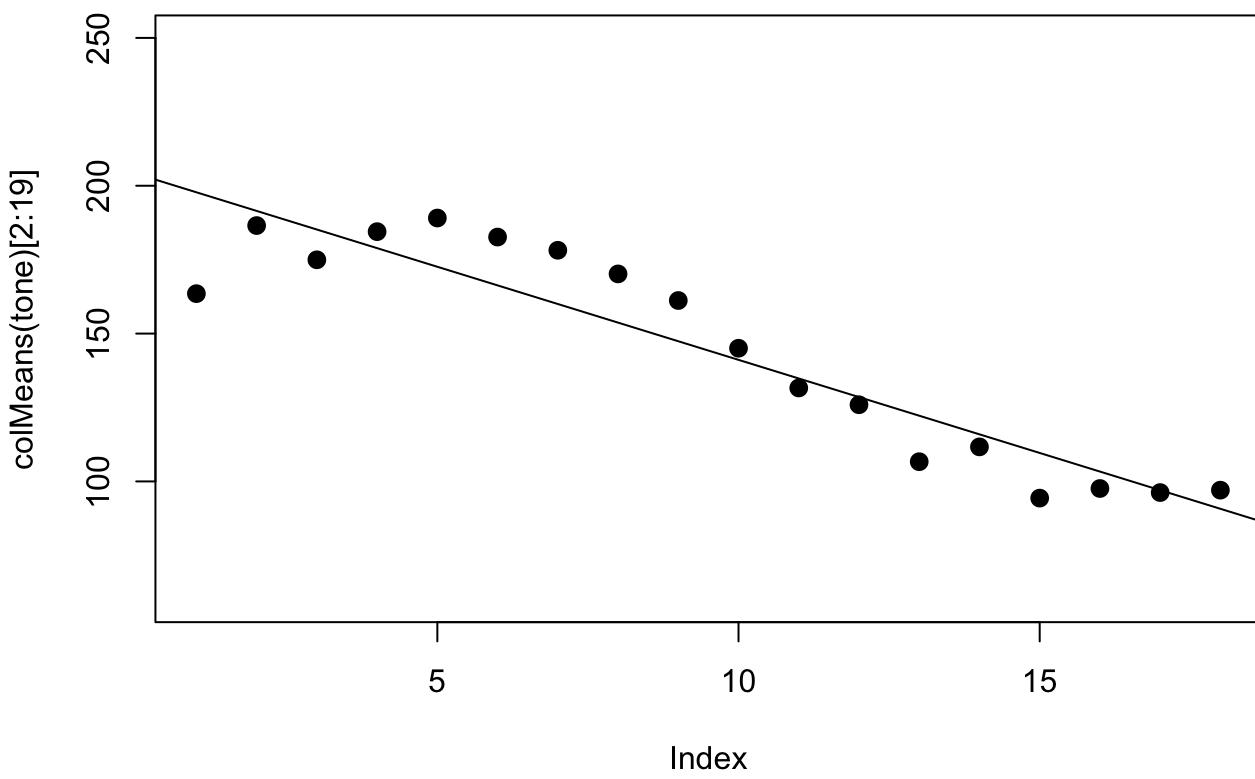
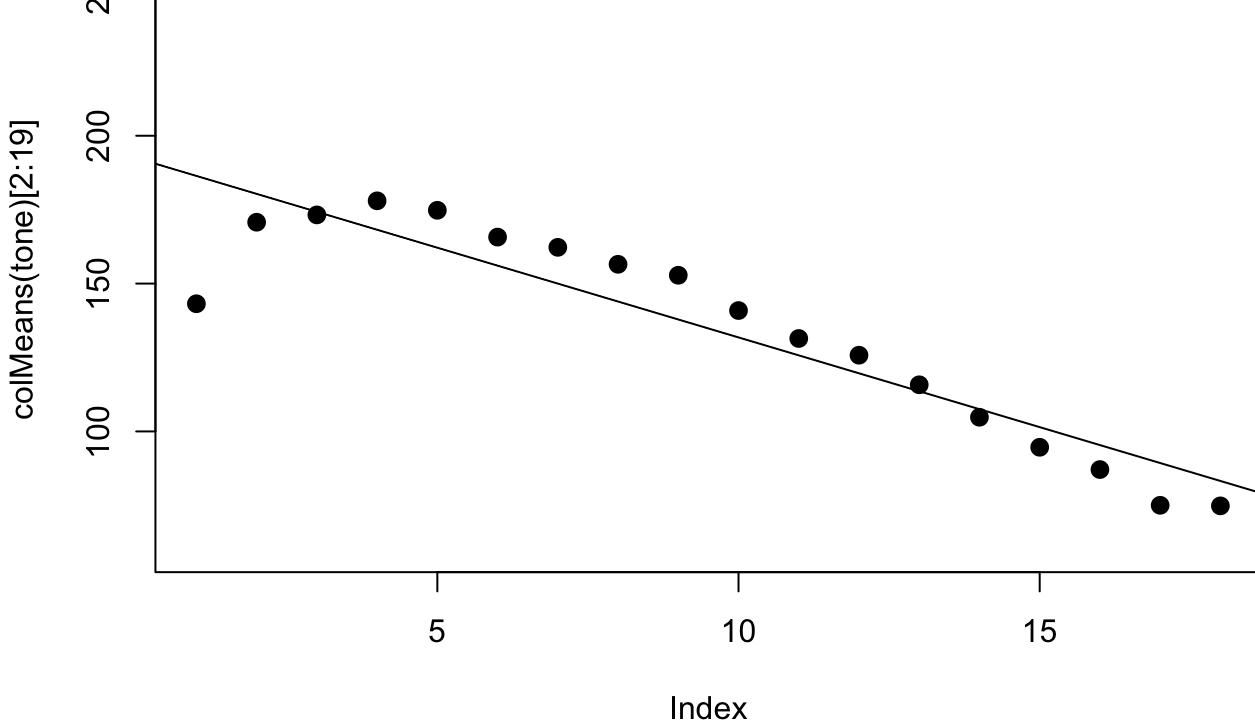


```
## [1] -3.547011 -5.916156 -5.927092
```

Tone B2 on U

```
plot_f0_by_vowels("B2", "U")
```

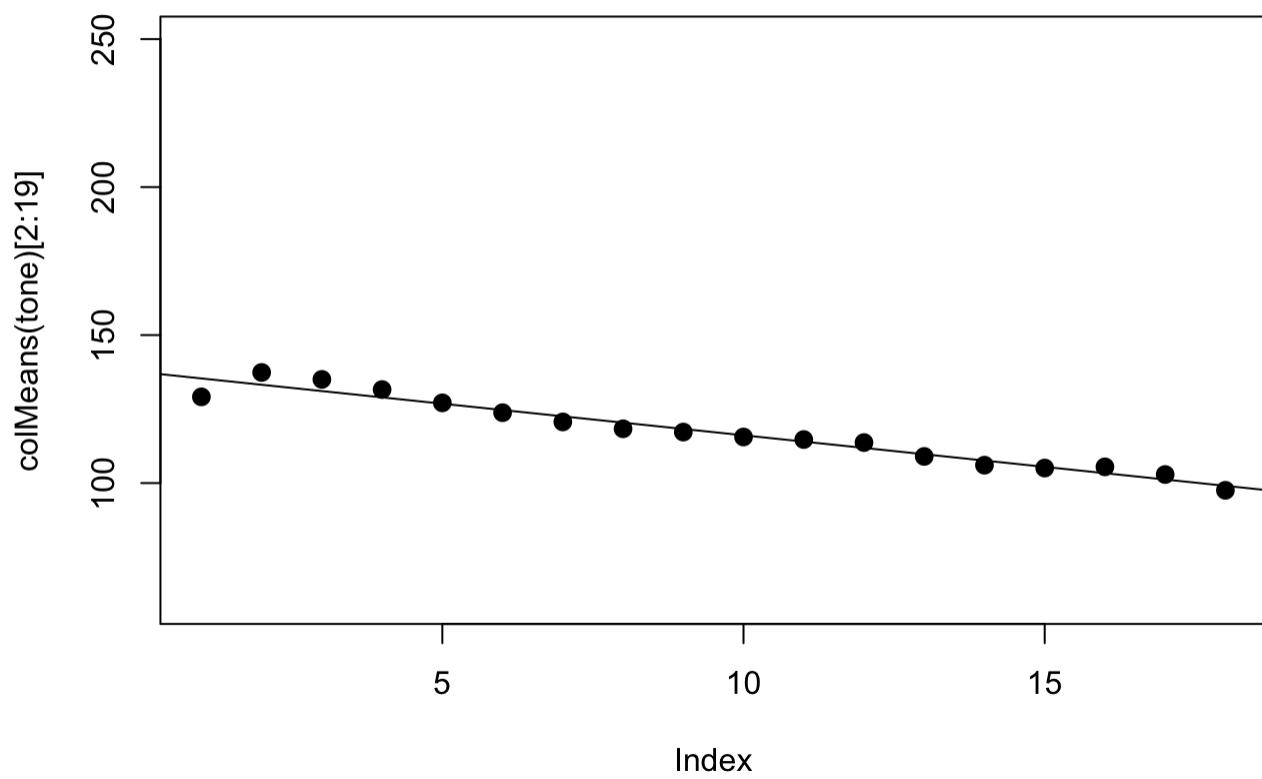
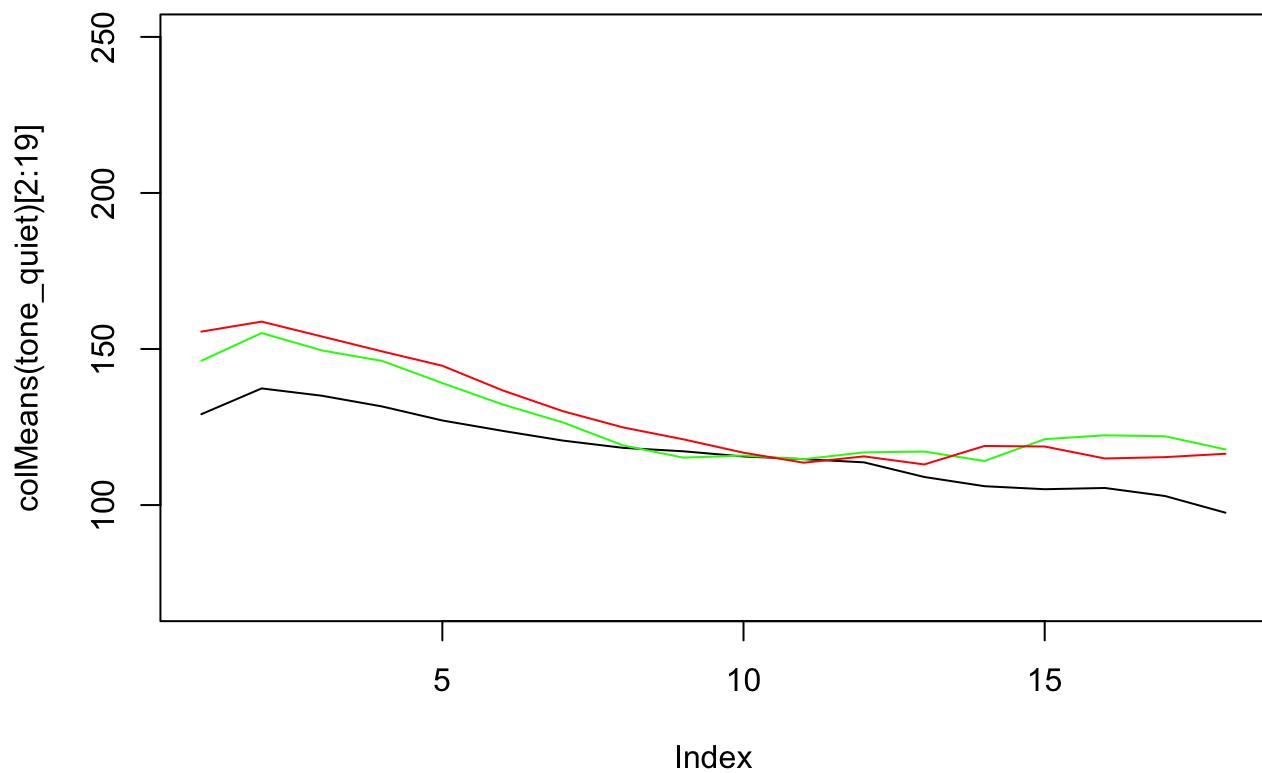



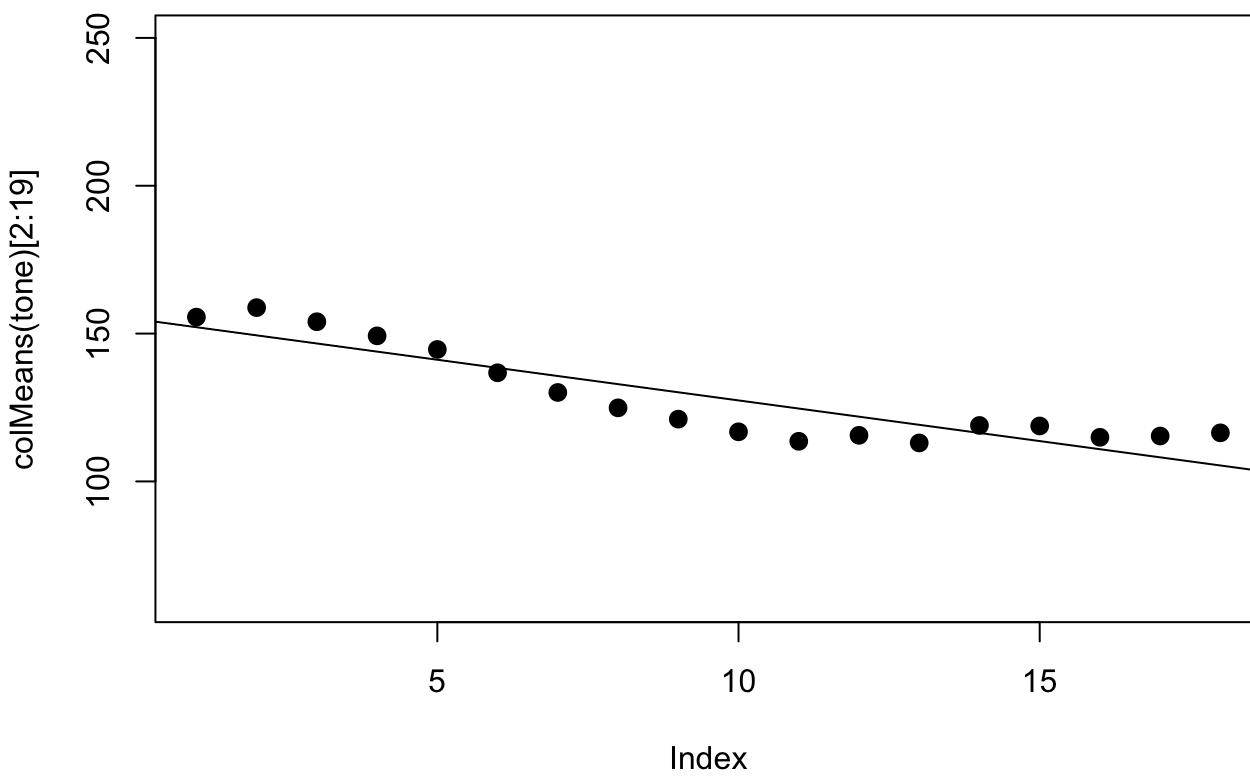
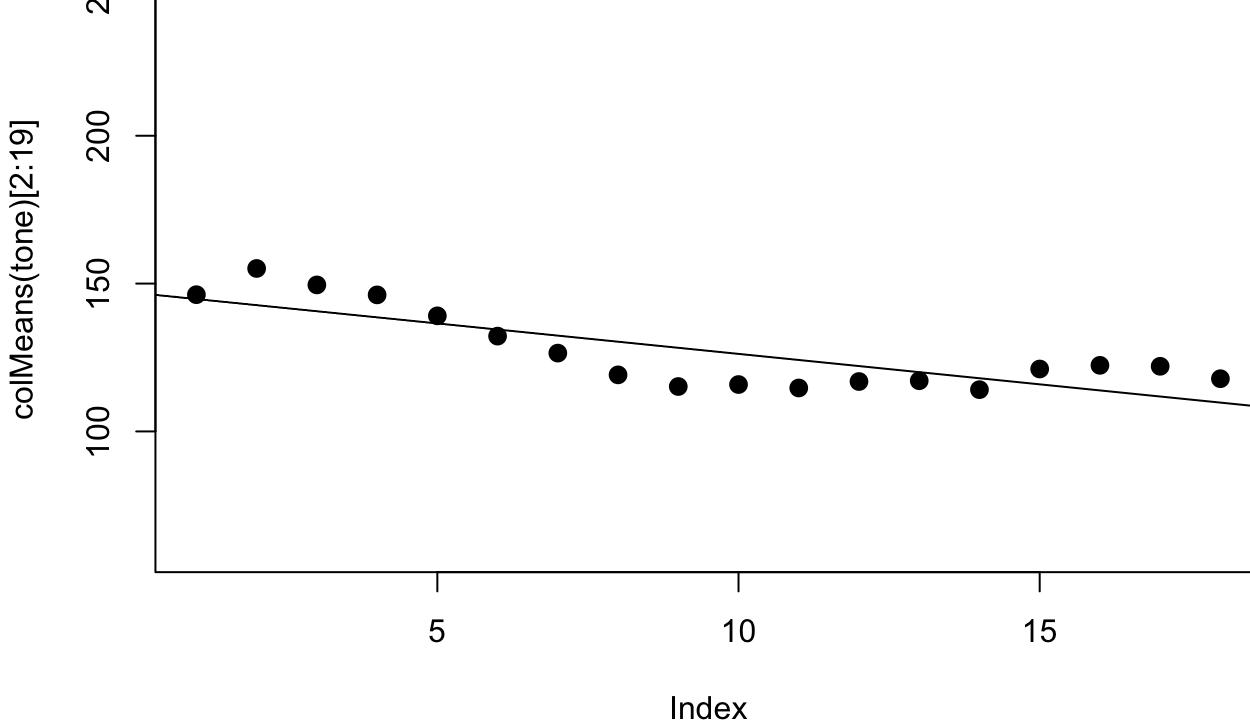


```
## [1] -4.080879 -6.072111 -6.298078
```

Tone C1 on A

```
plot_f0_by_vowels("C1", "A")
```

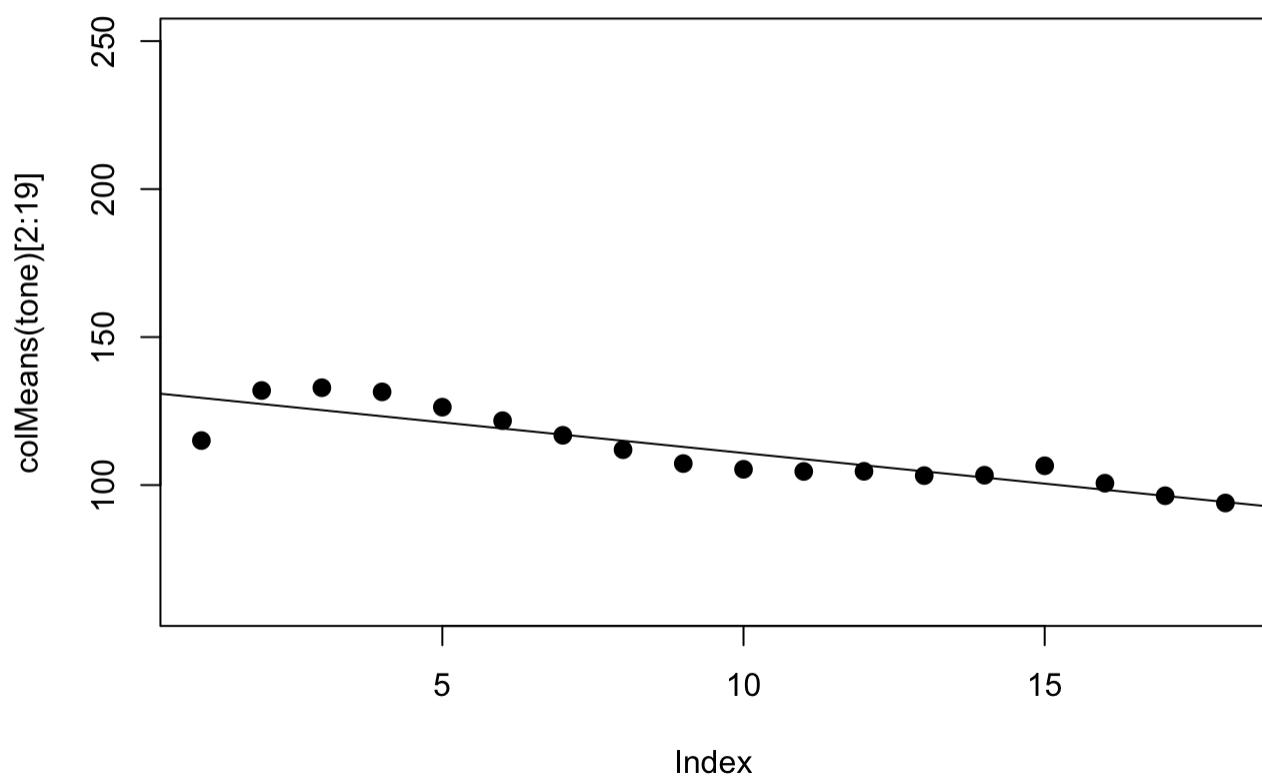
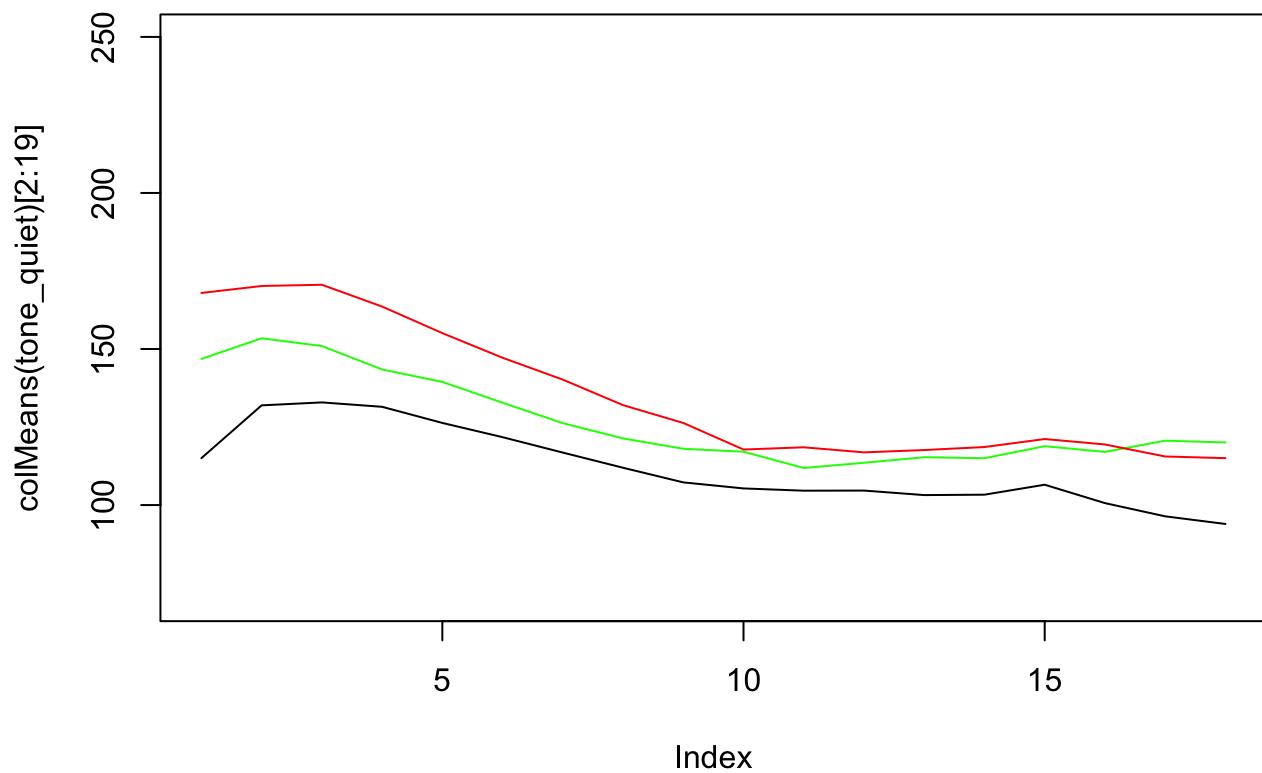



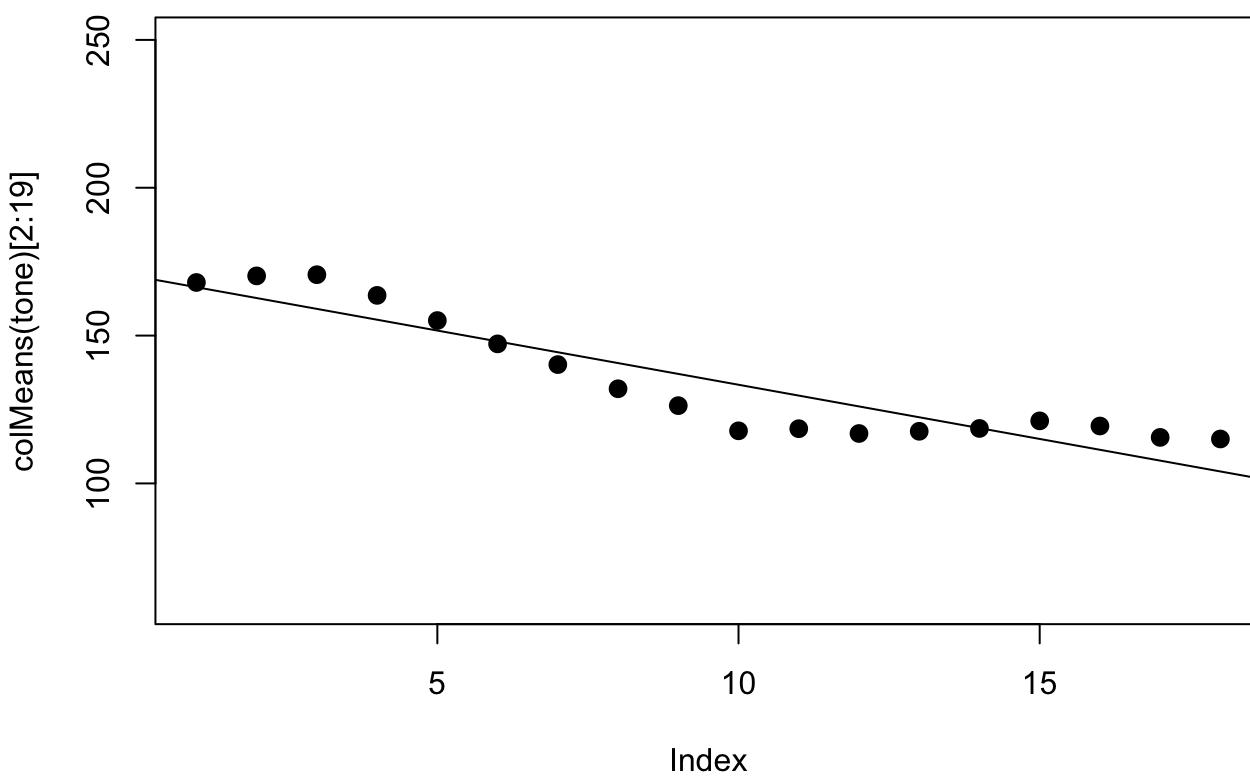
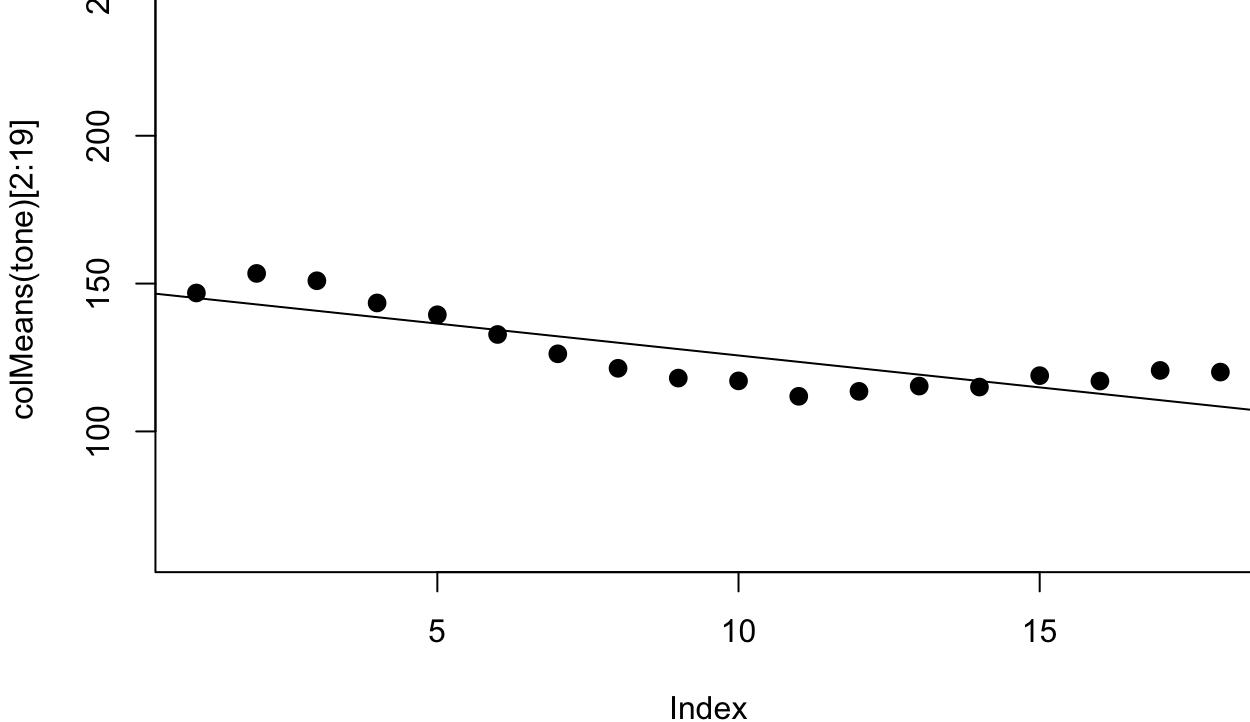


```
## [1] -2.136312 -2.063681 -2.753528
```

Tone C1 on E

```
plot_f0_by_vowels("C1", "E")
```

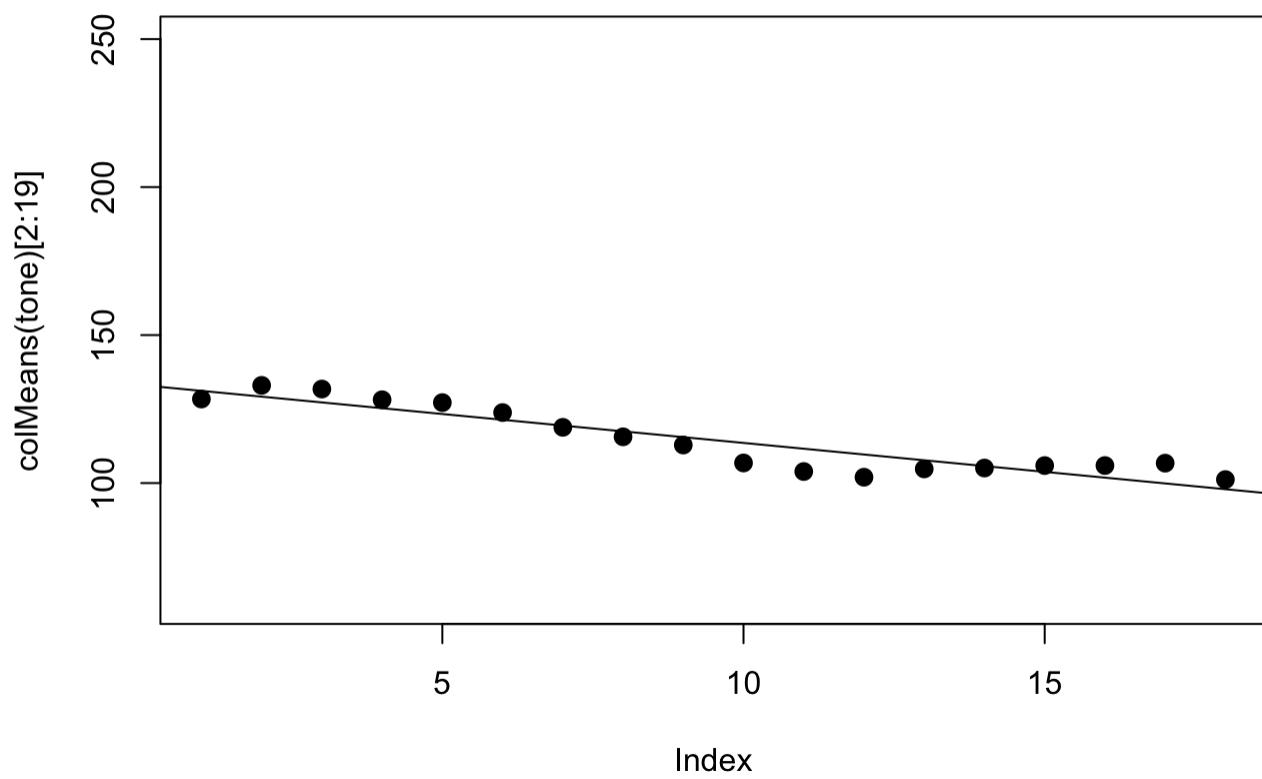
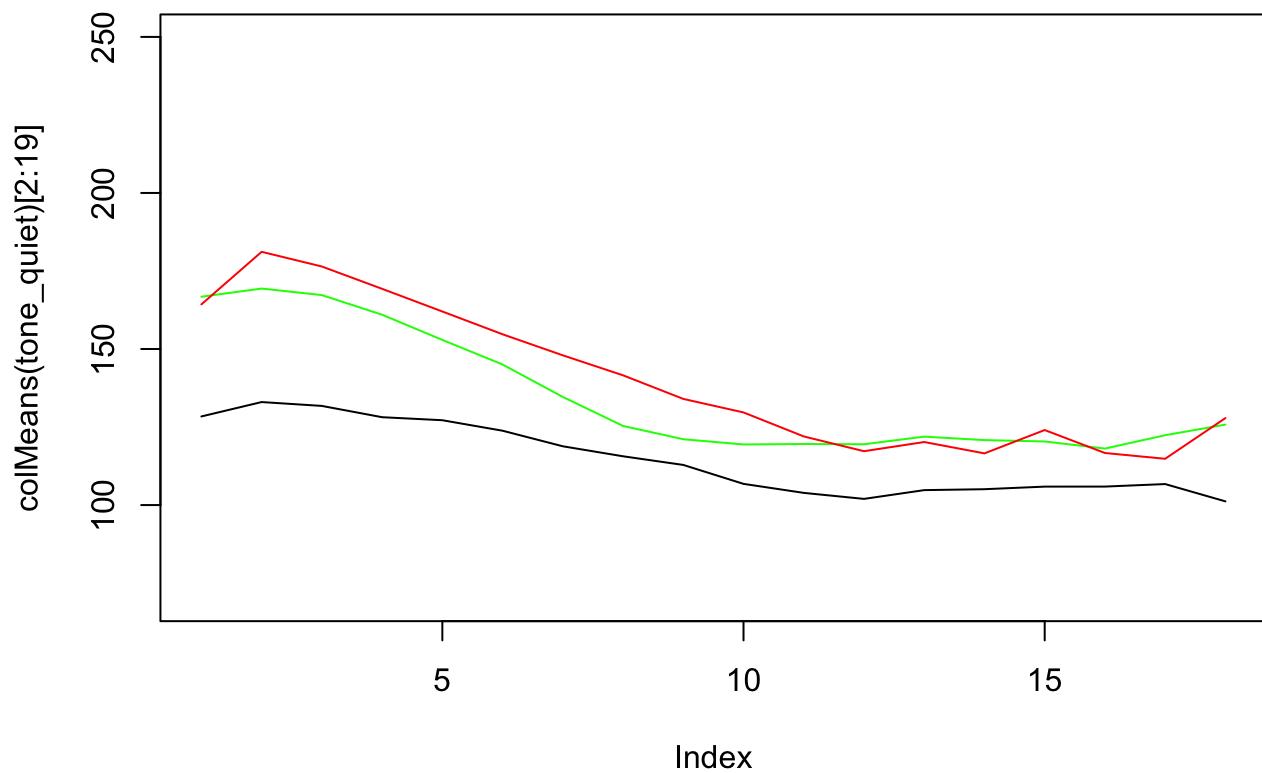



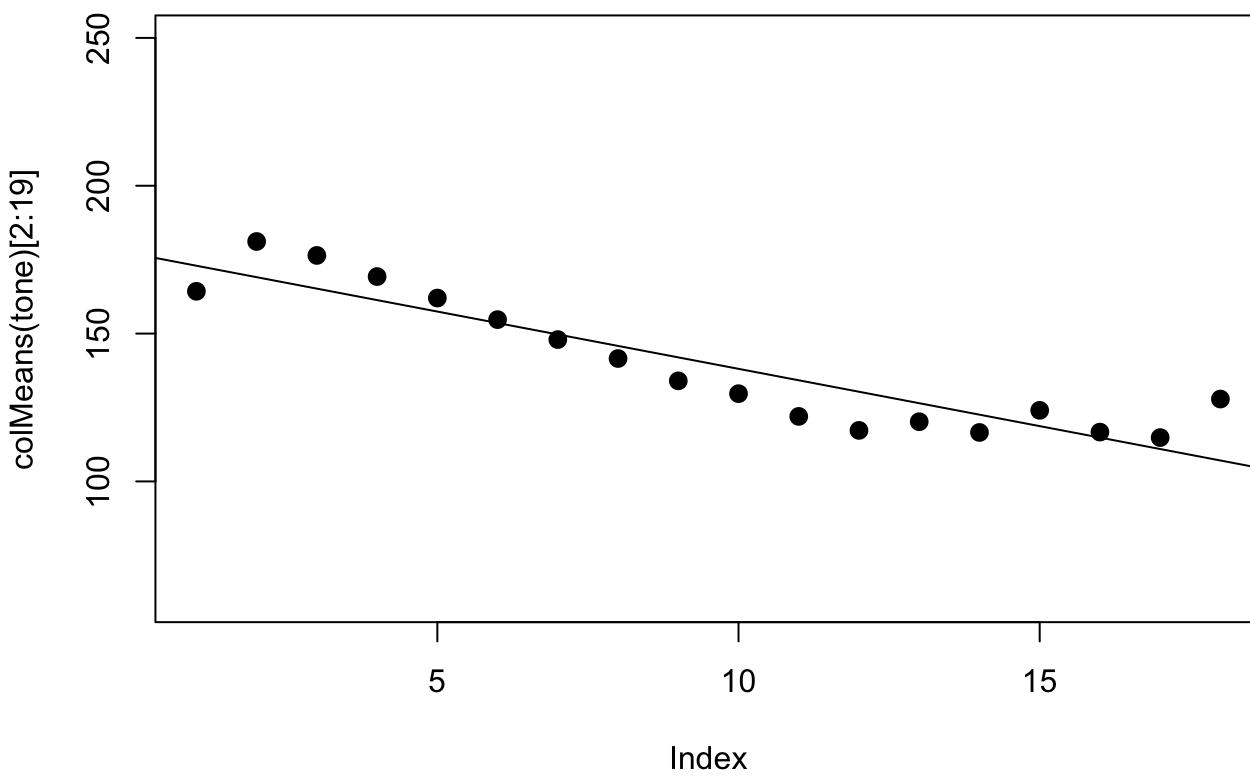
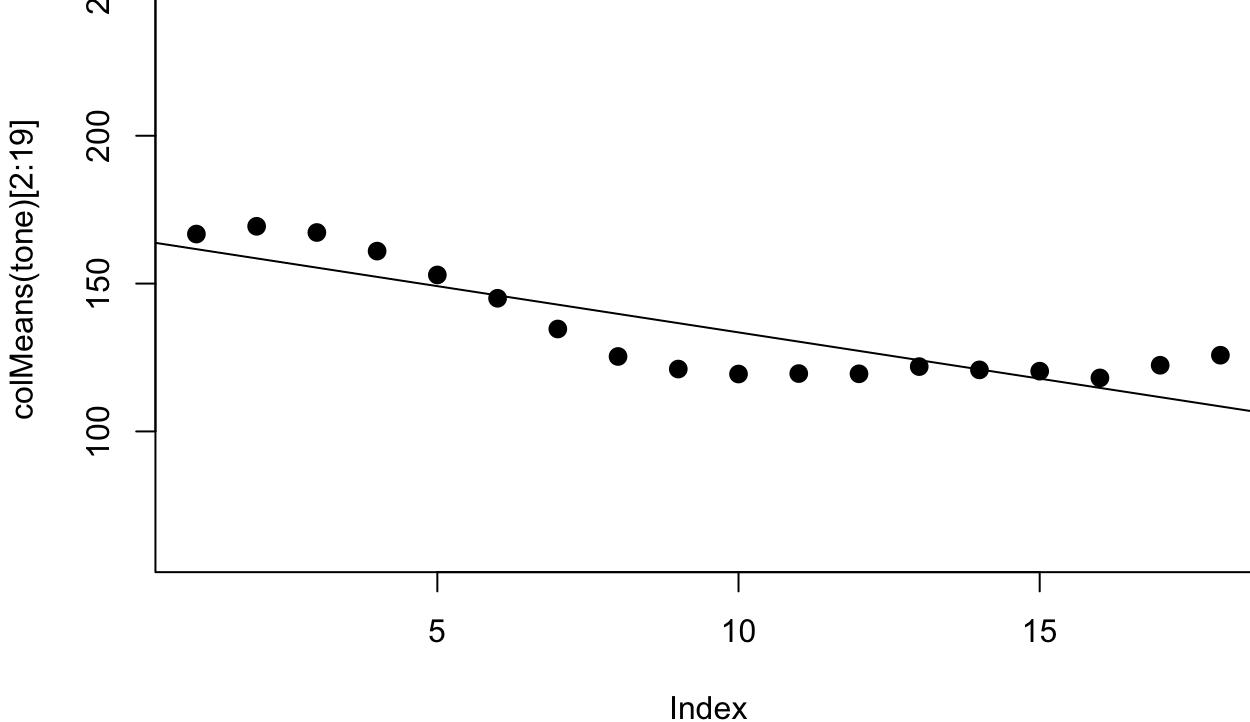


```
## [1] -2.071855 -2.160343 -3.665938
```

Tone C1 on U

```
plot_f0_by_vowels("C1", "U")
```

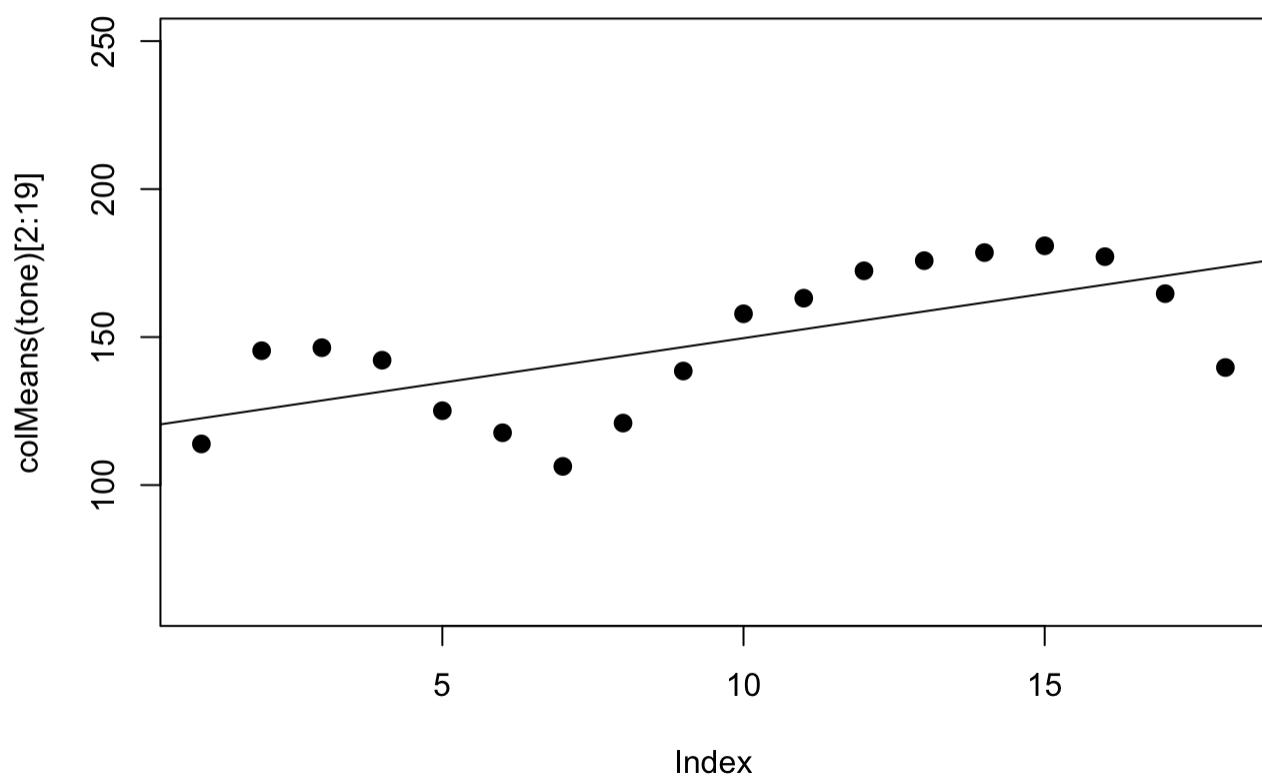
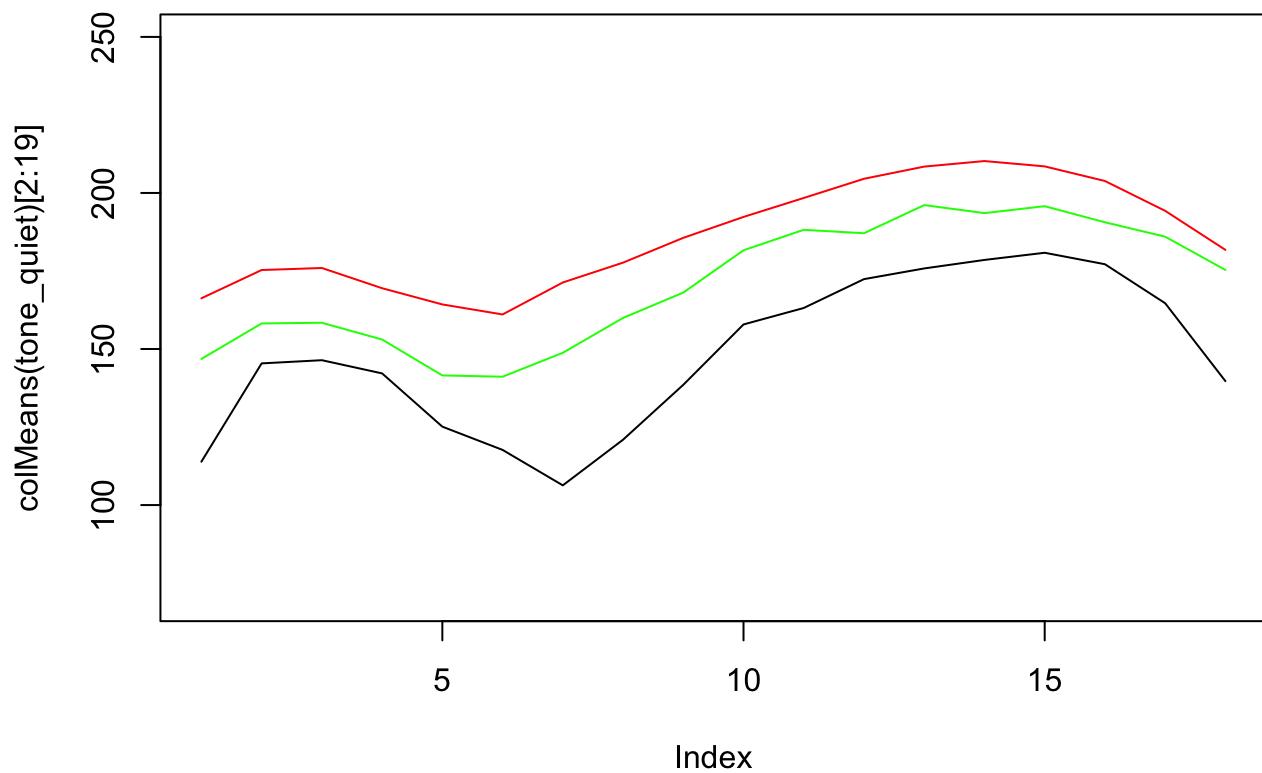



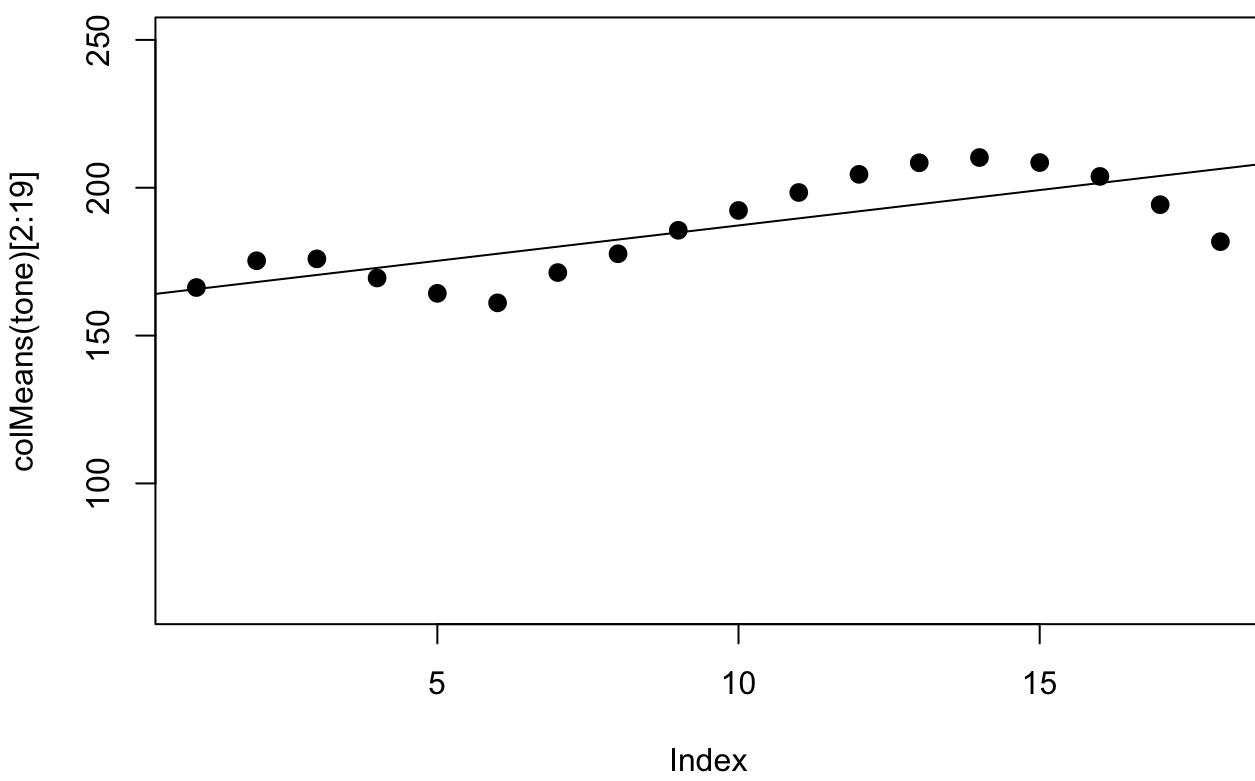
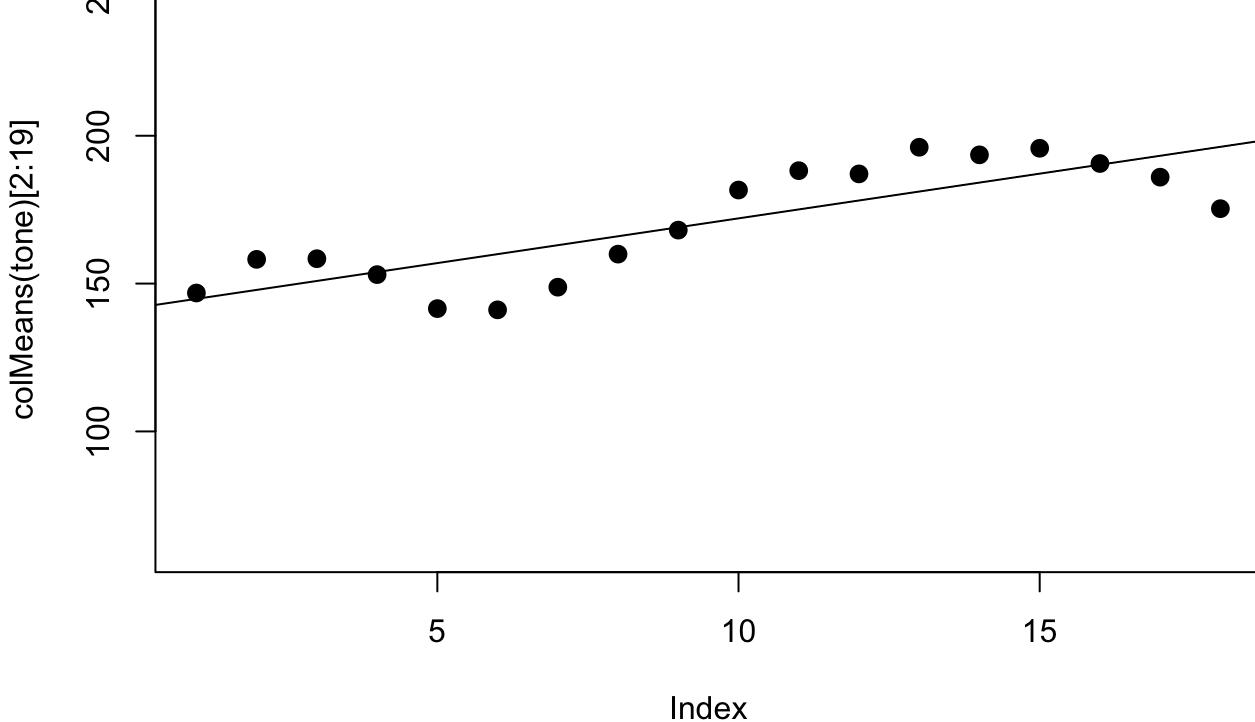


```
## [1] -1.955666 -3.130111 -3.876376
```

Tone C2 on A

```
plot_f0_by_vowels("C2", "A")
```

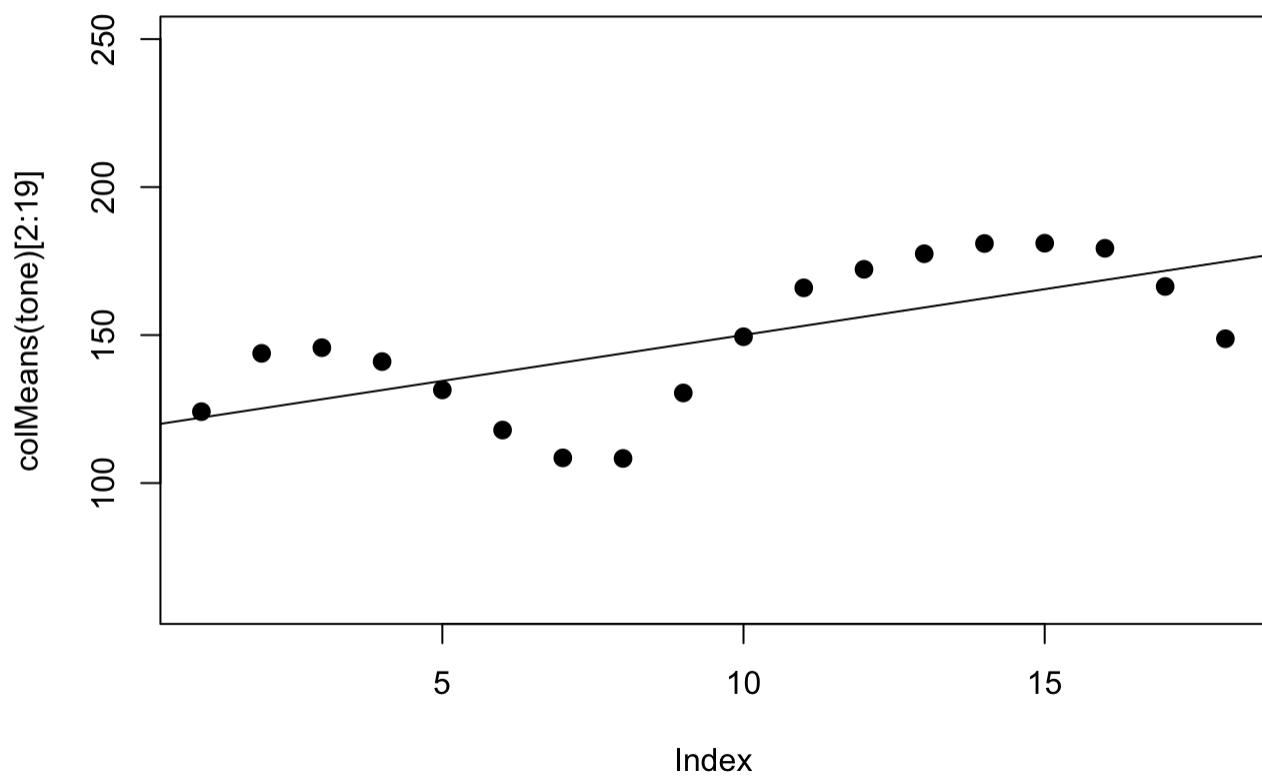
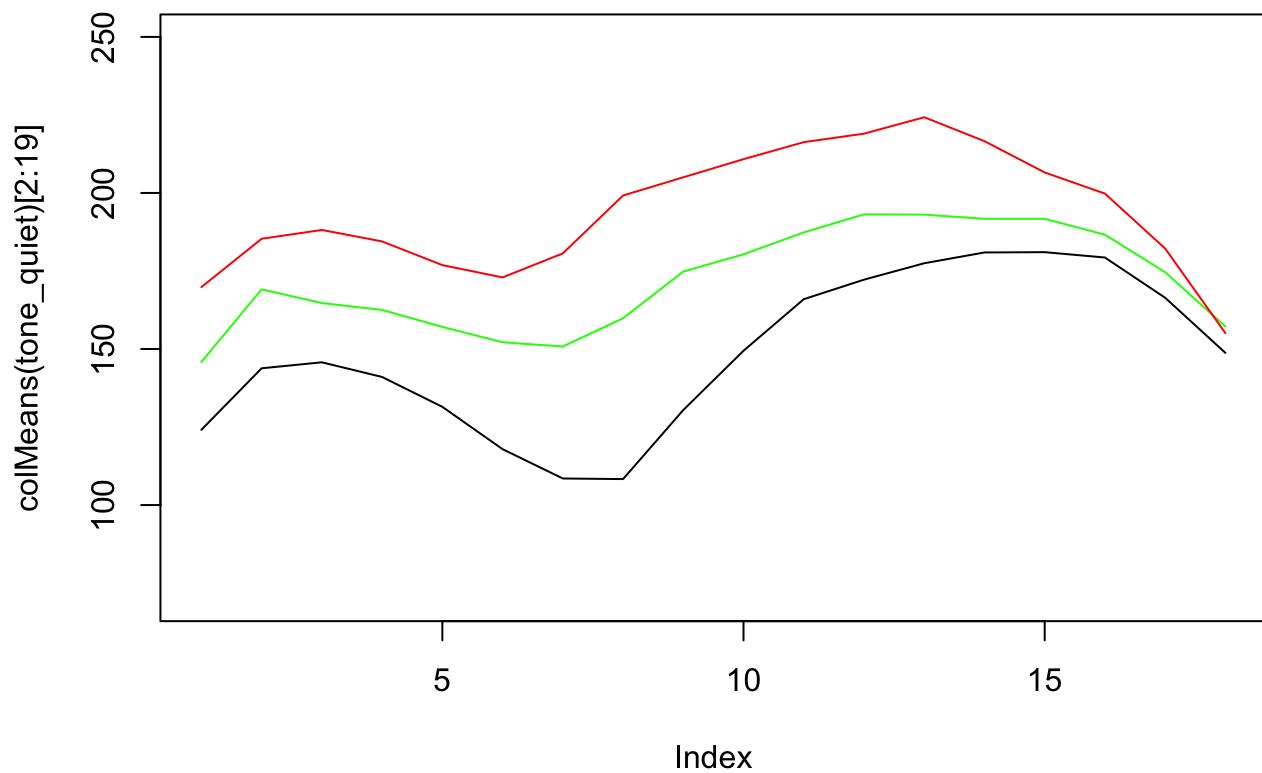



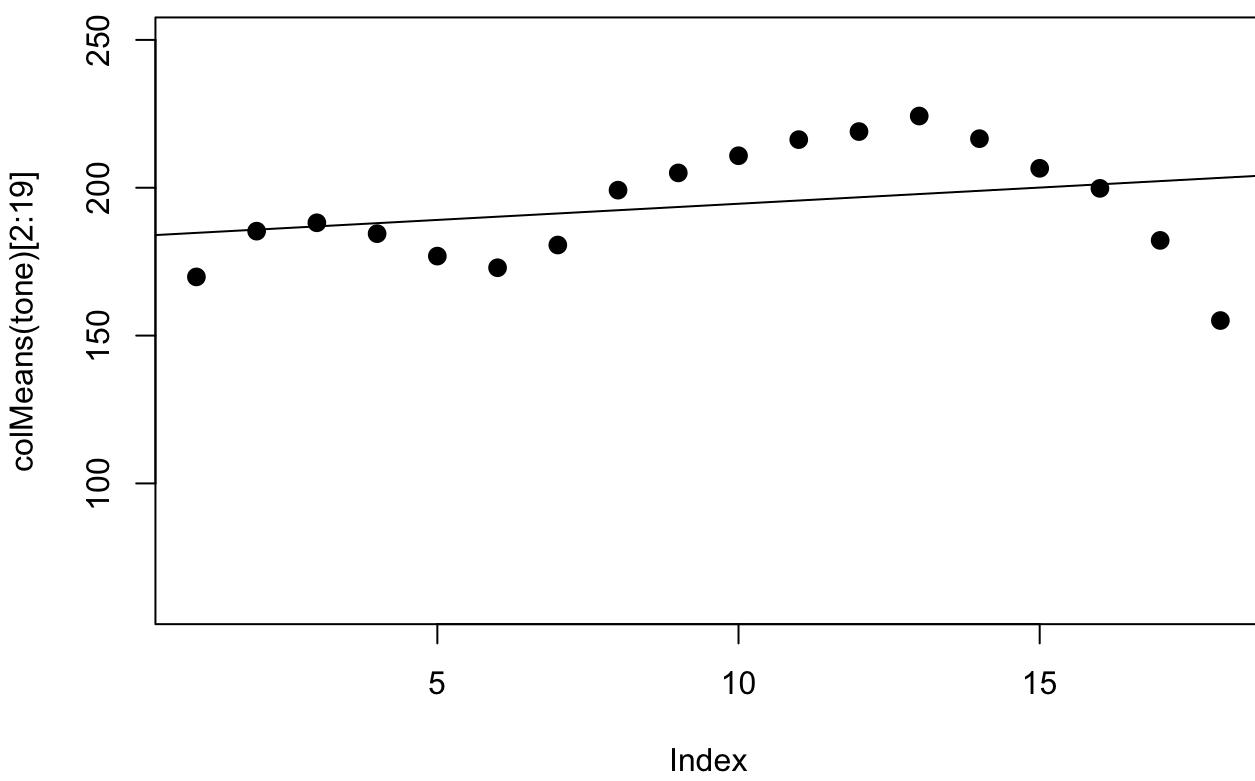
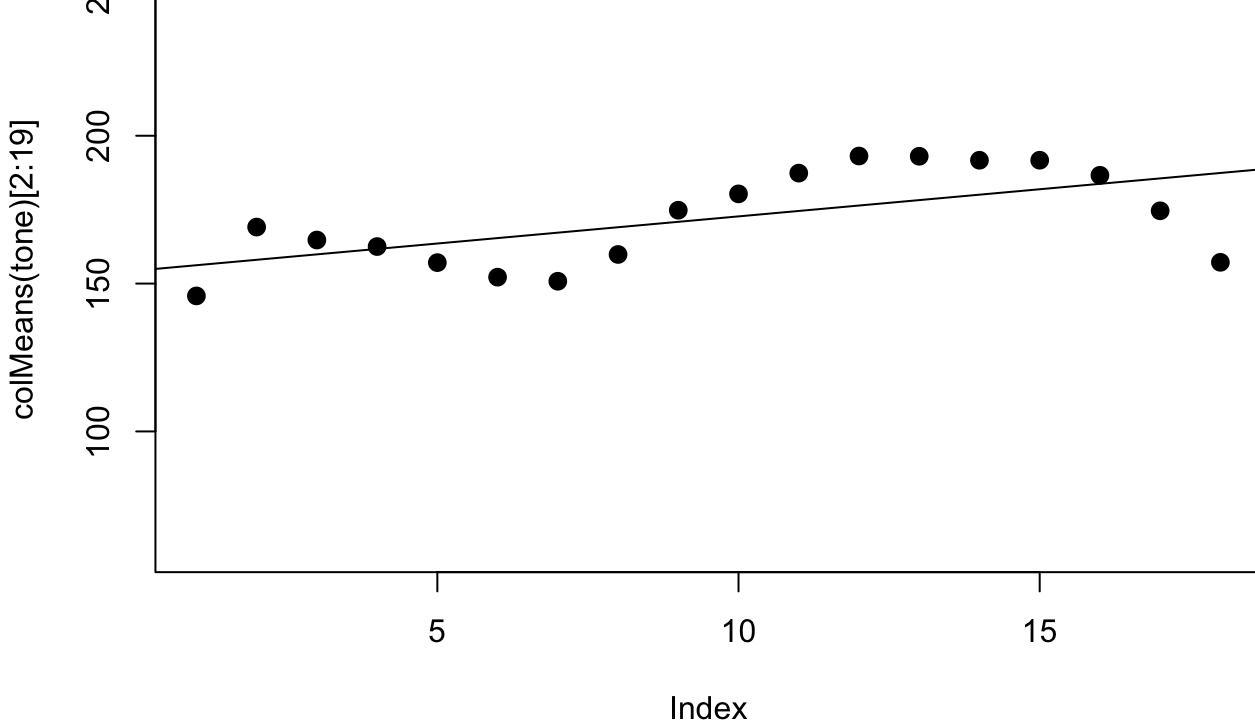


```
## [1] 3.010333 3.027028 2.394154
```

Tone C2 on E

```
plot_f0_by_vowels("C2", "E")
```

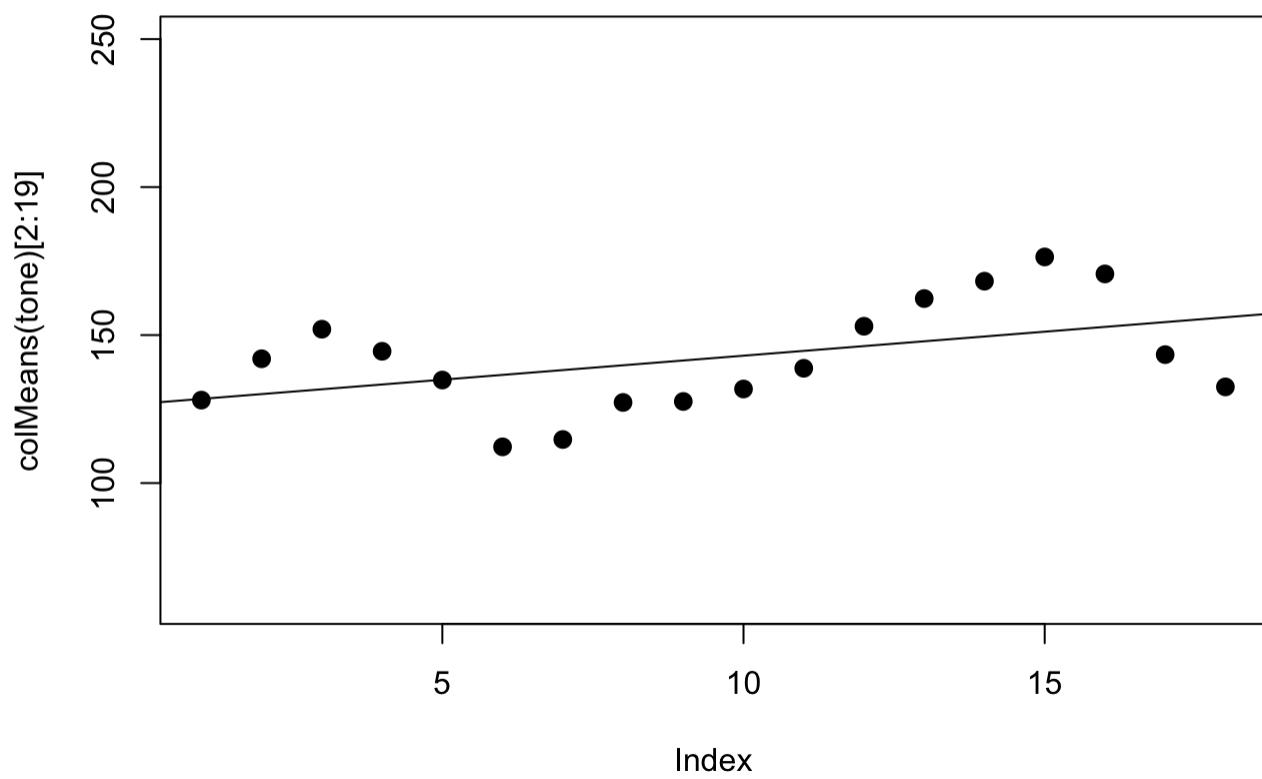
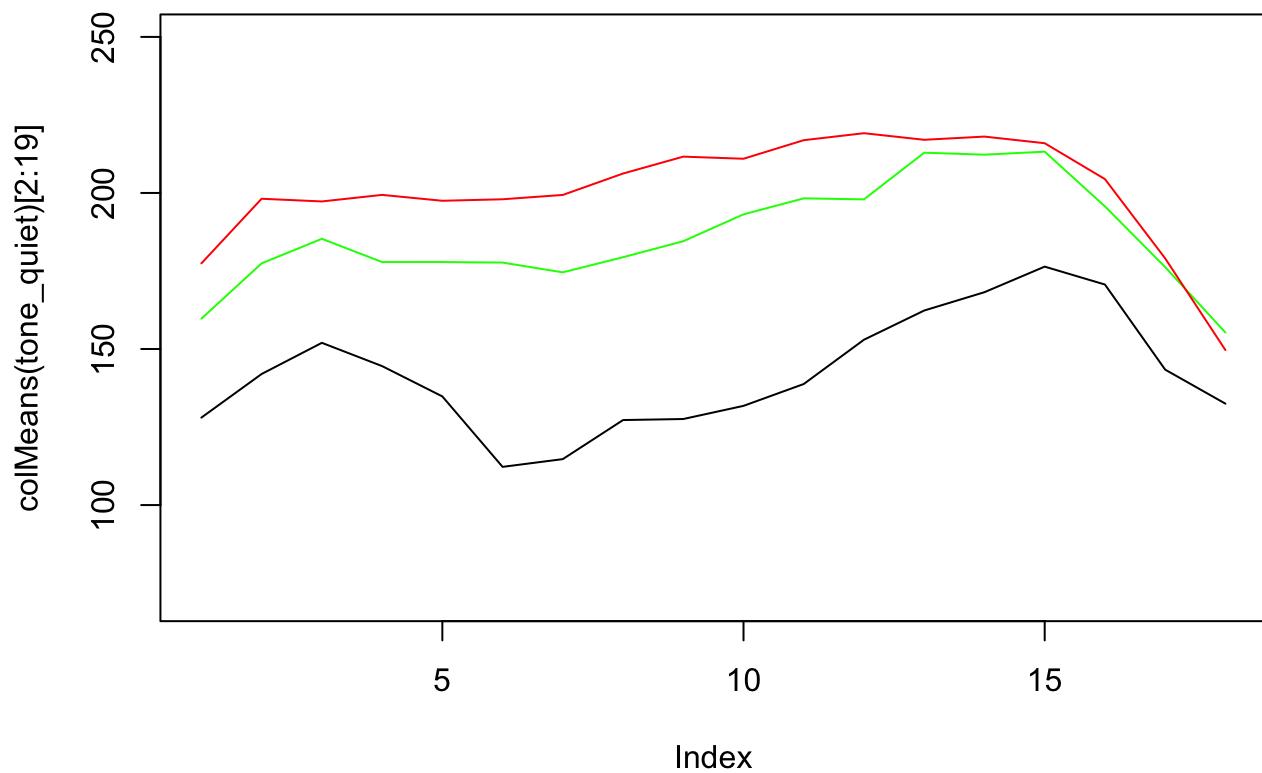



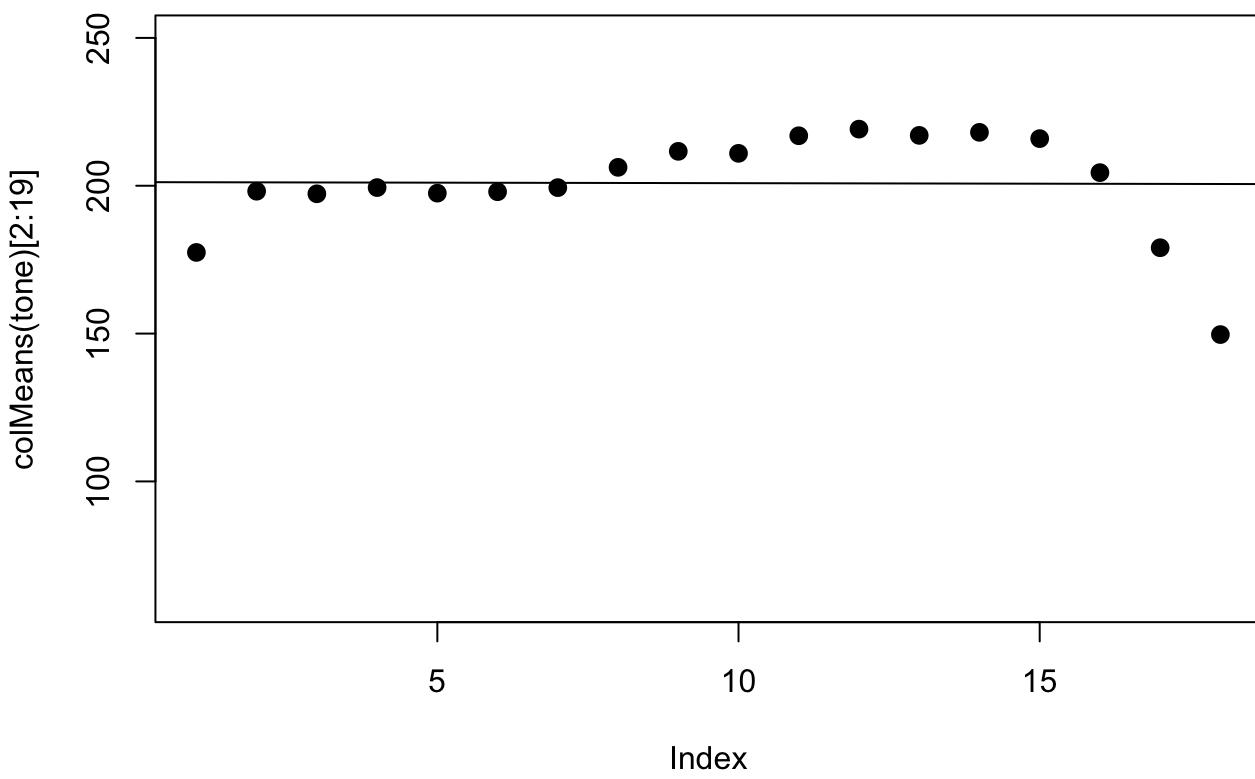
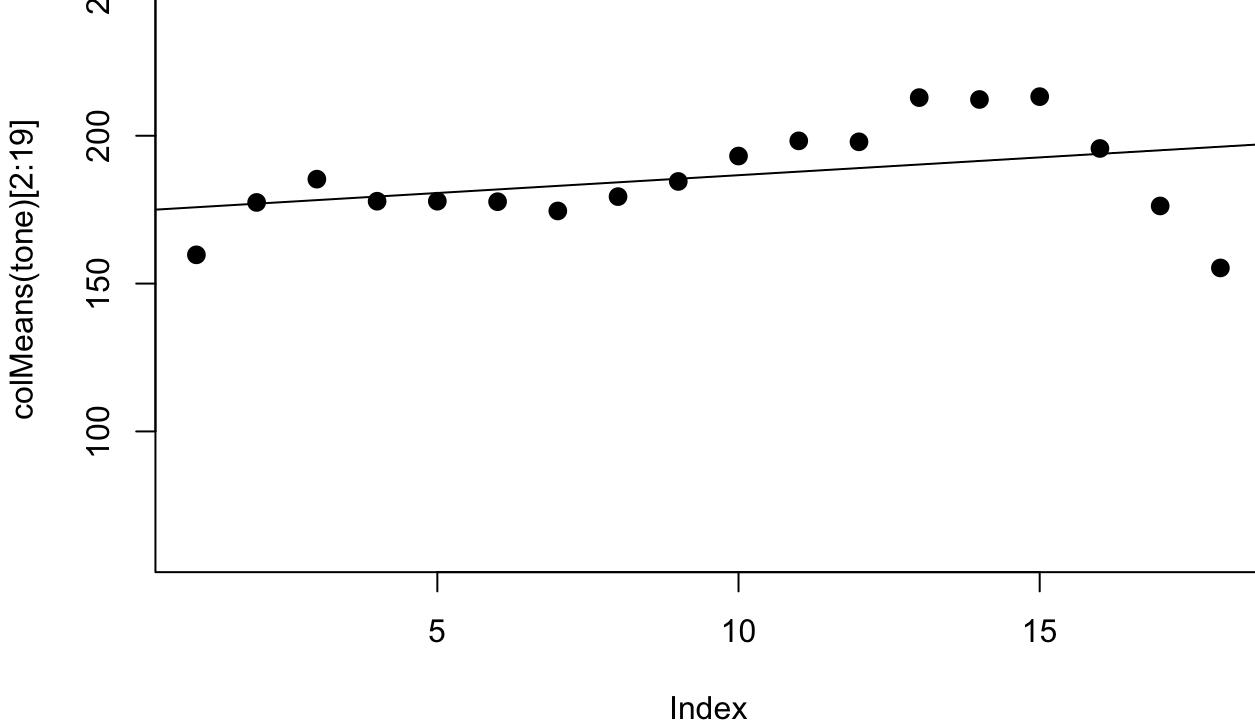


```
## [1] 3.102830 1.836030 1.096396
```

Tone C2 on U

```
plot_f0_by_vowels("C2", "U")
```

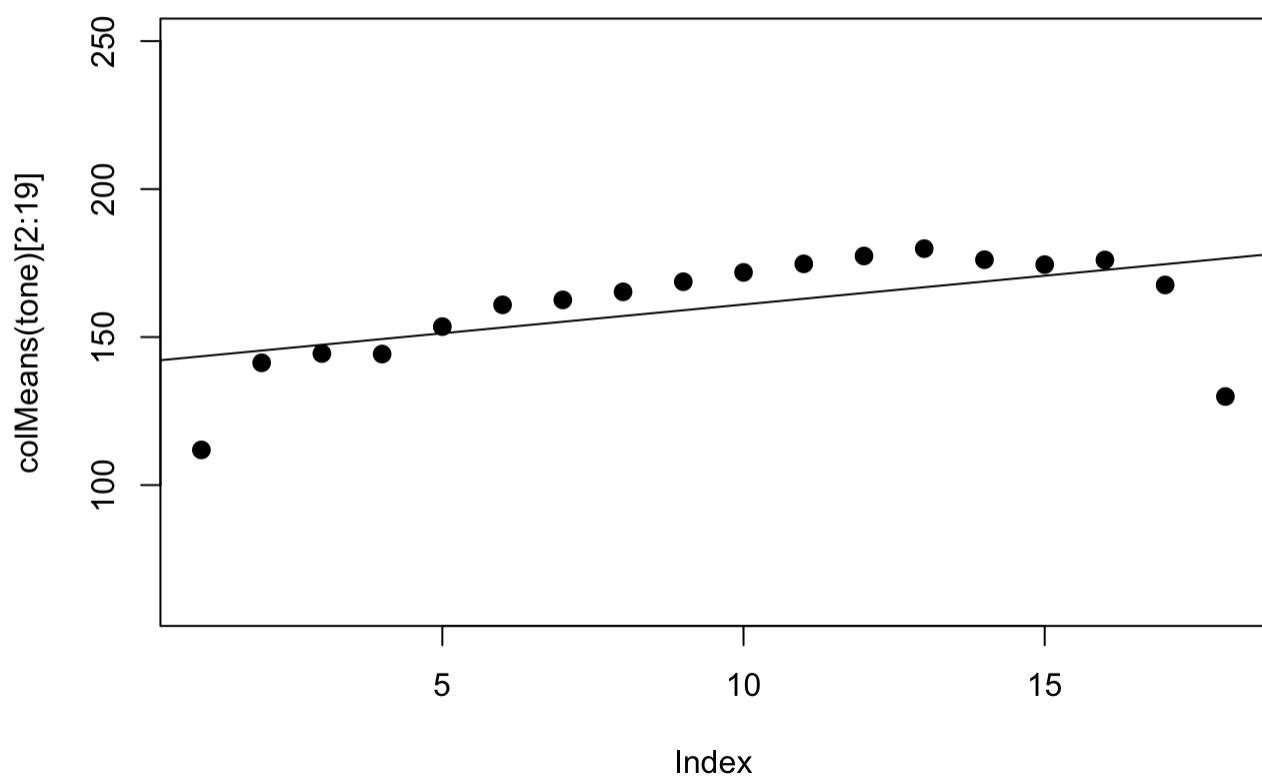
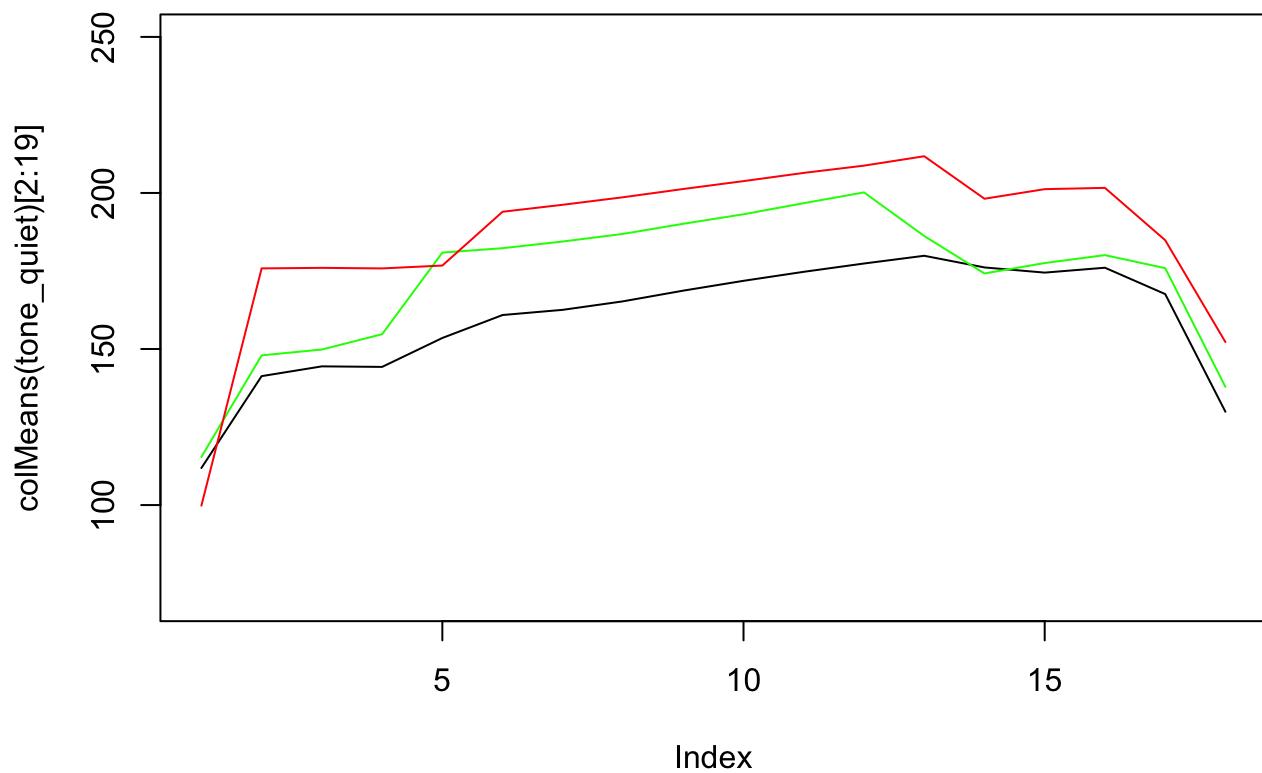



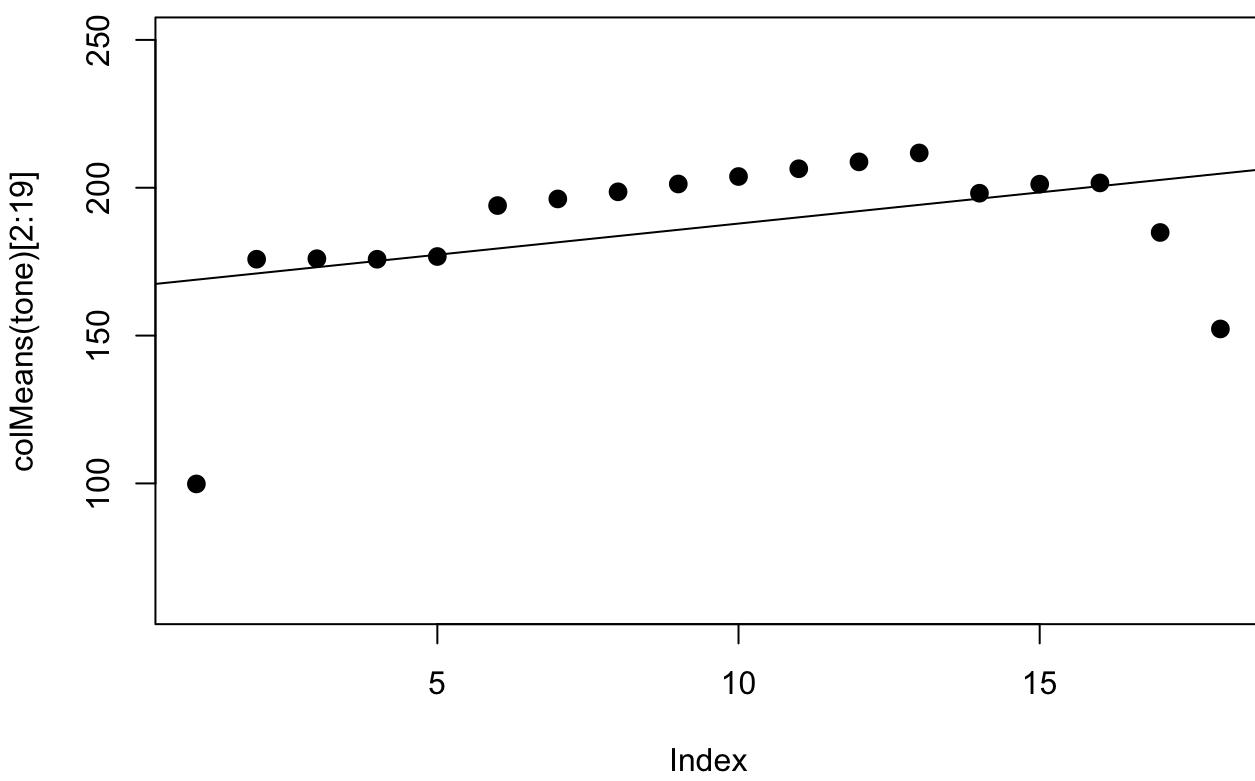
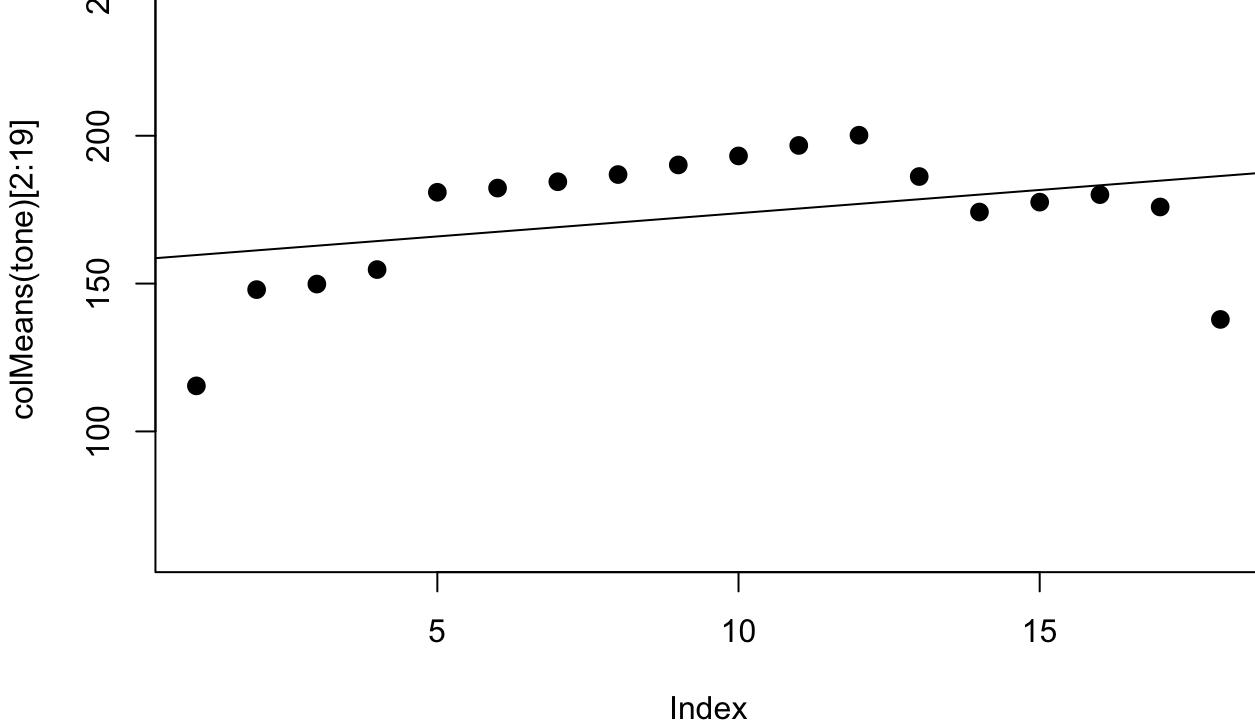


```
## [1] 1.62192705 1.20632671 -0.03604034
```

Tone D1 on A

```
plot_f0_by_vowels("D1", "A")
```

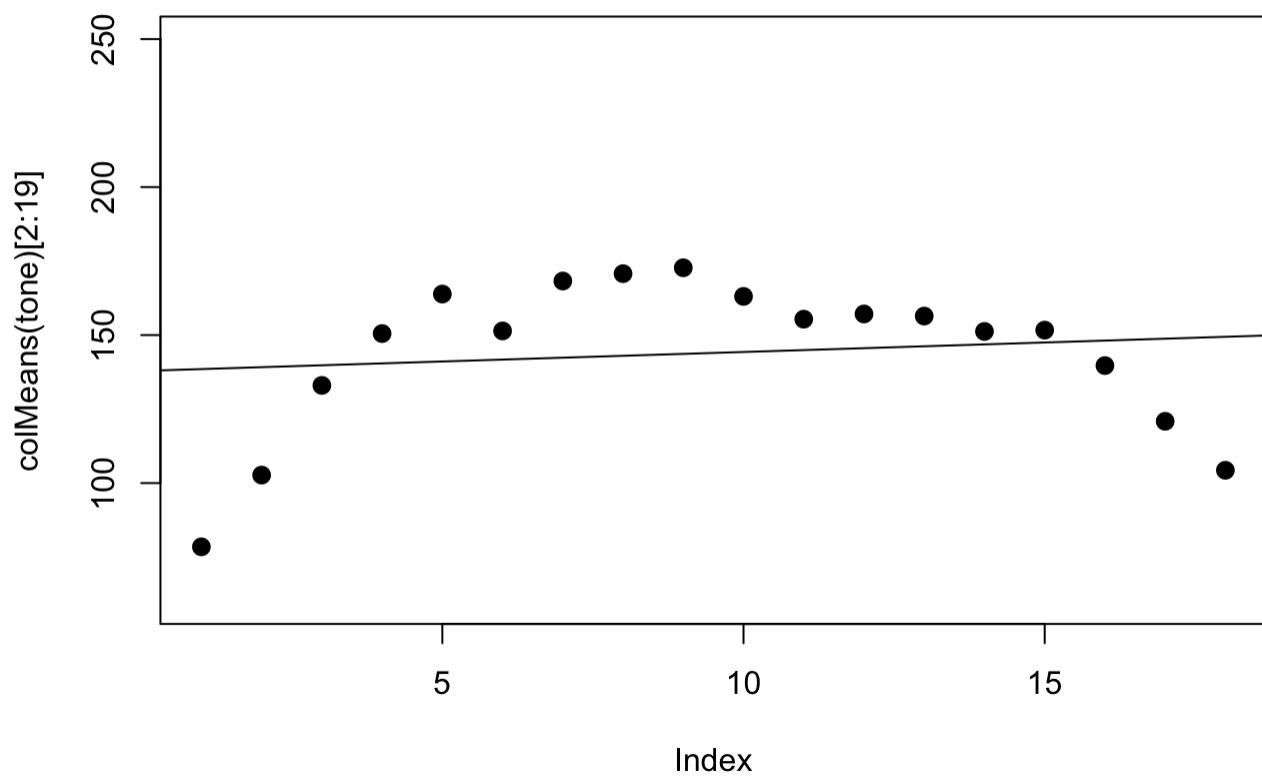
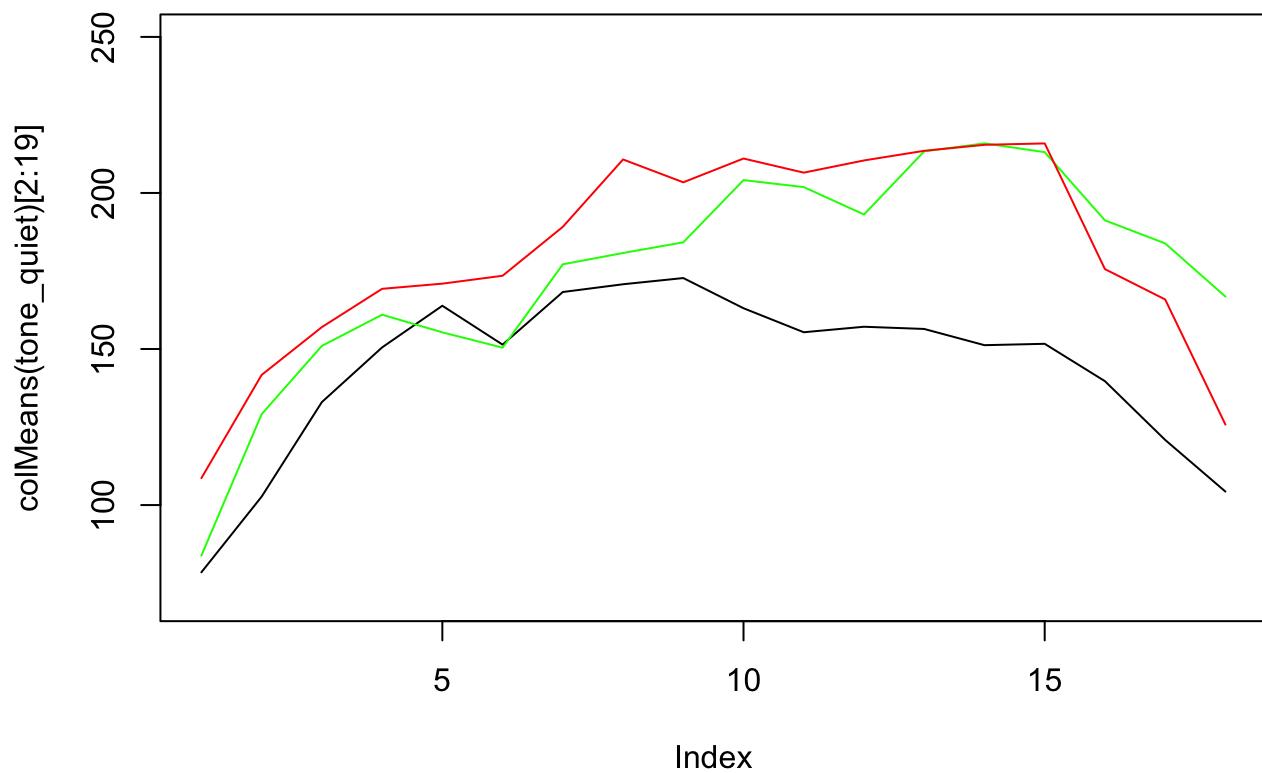



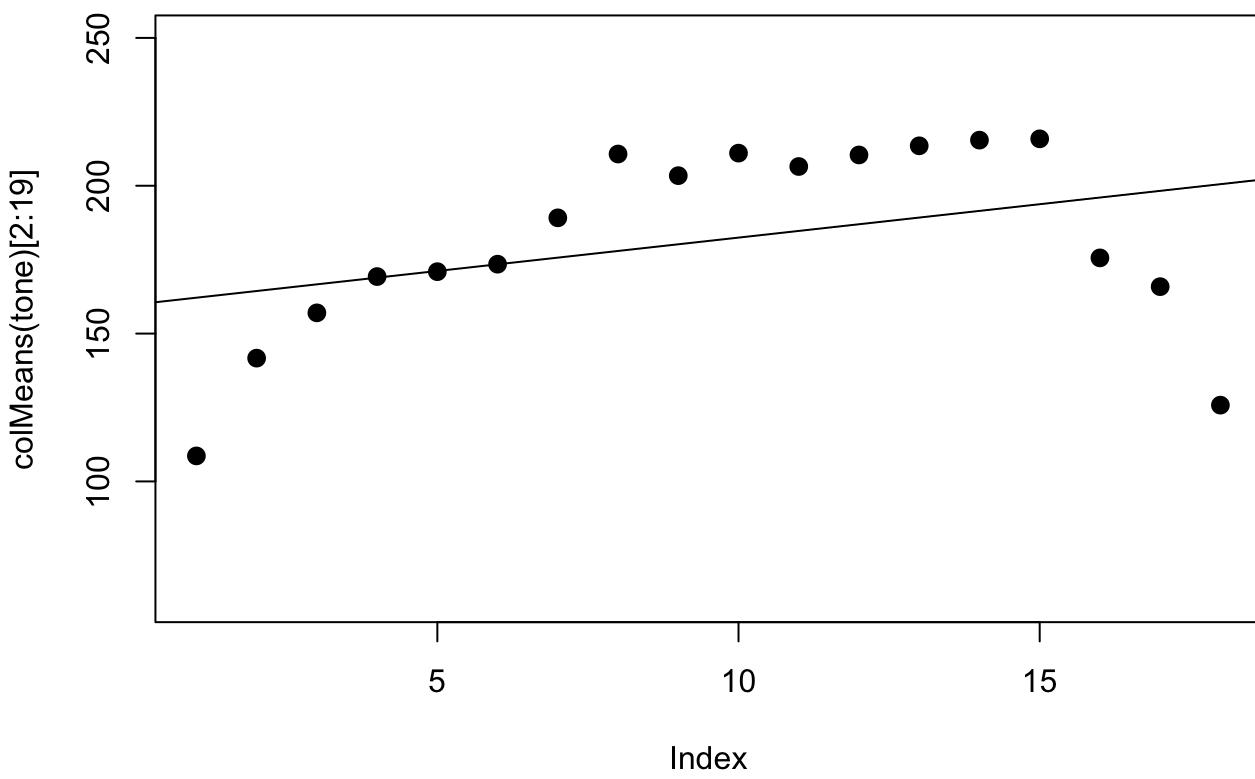
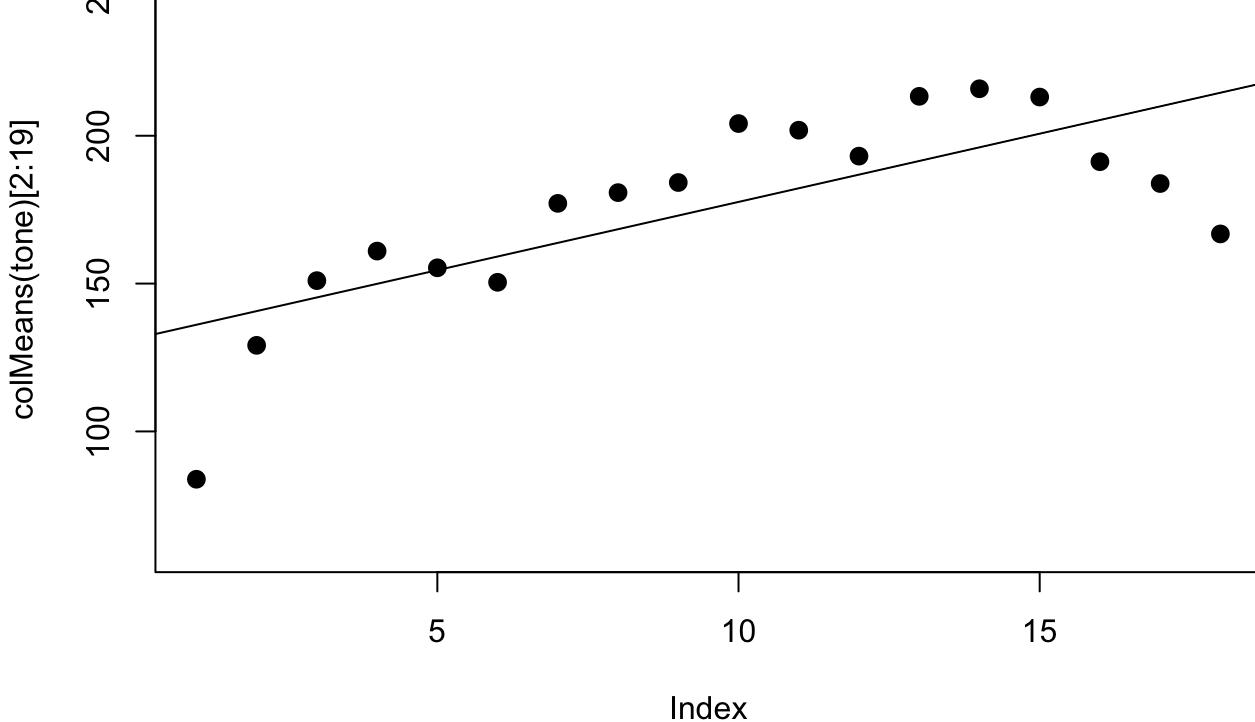


```
## [1] 1.946698 1.572546 2.110463
```

Tone D1 on E

```
plot_f0_by_vowels("D1", "E")
```

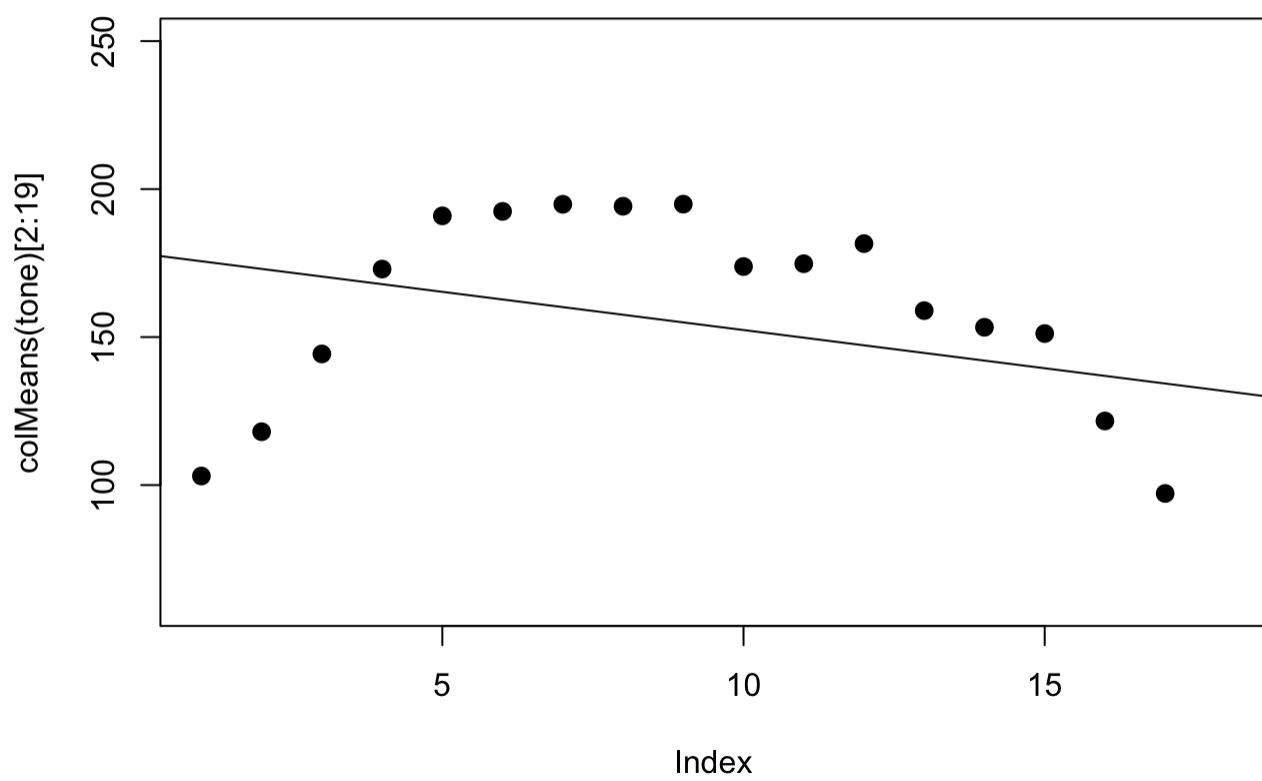
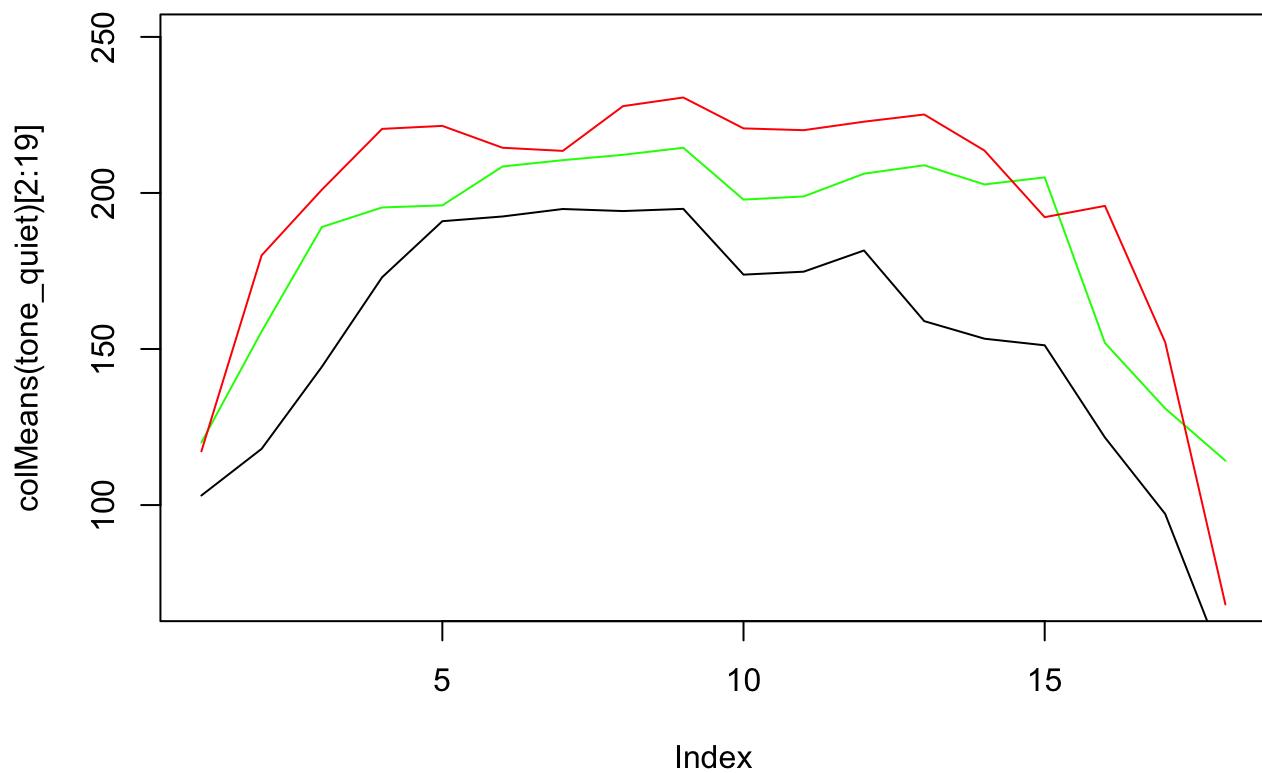



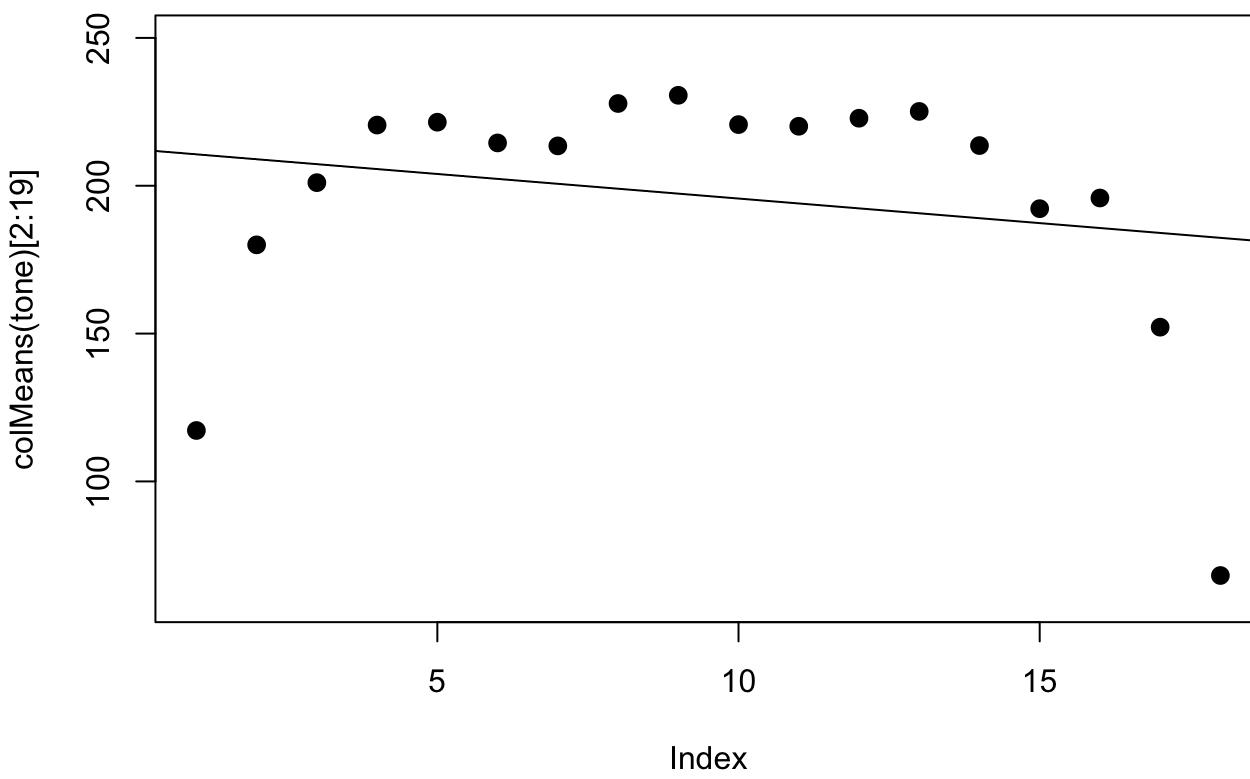
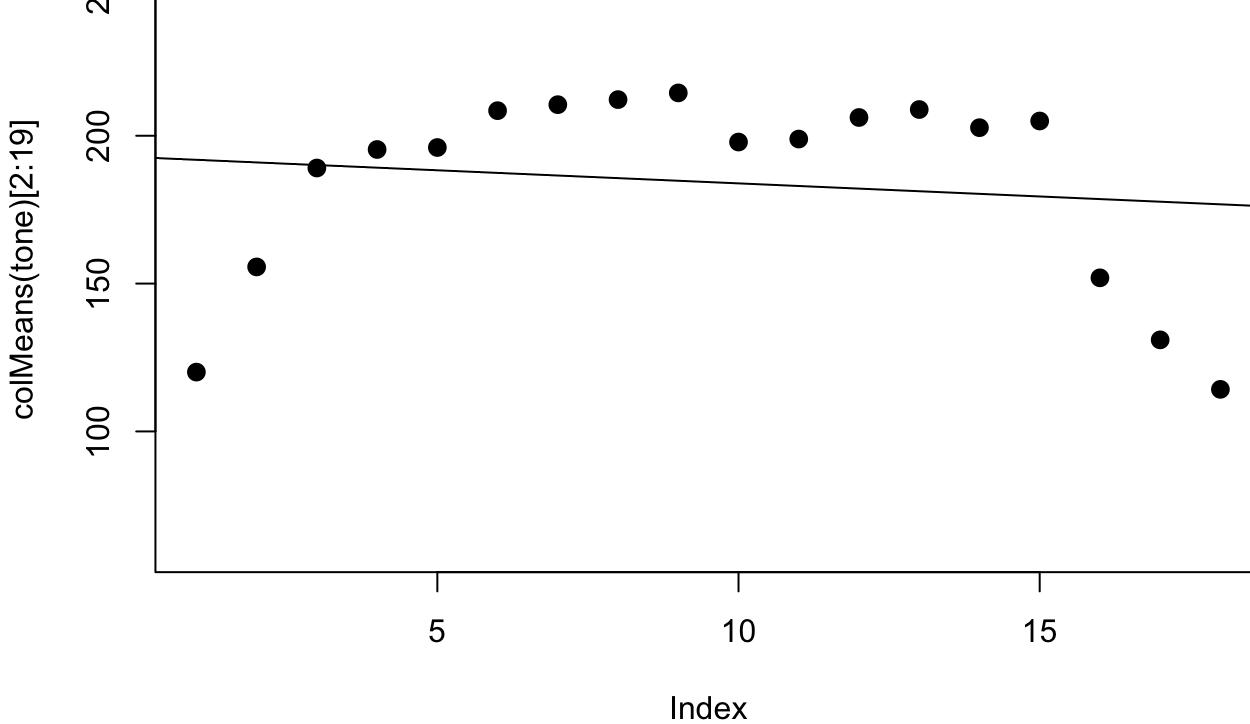


```
## [1] 0.6420803 4.6188919 2.2607038
```

Tone D1 on U

```
plot_f0_by_vowels("D1", "U")
```

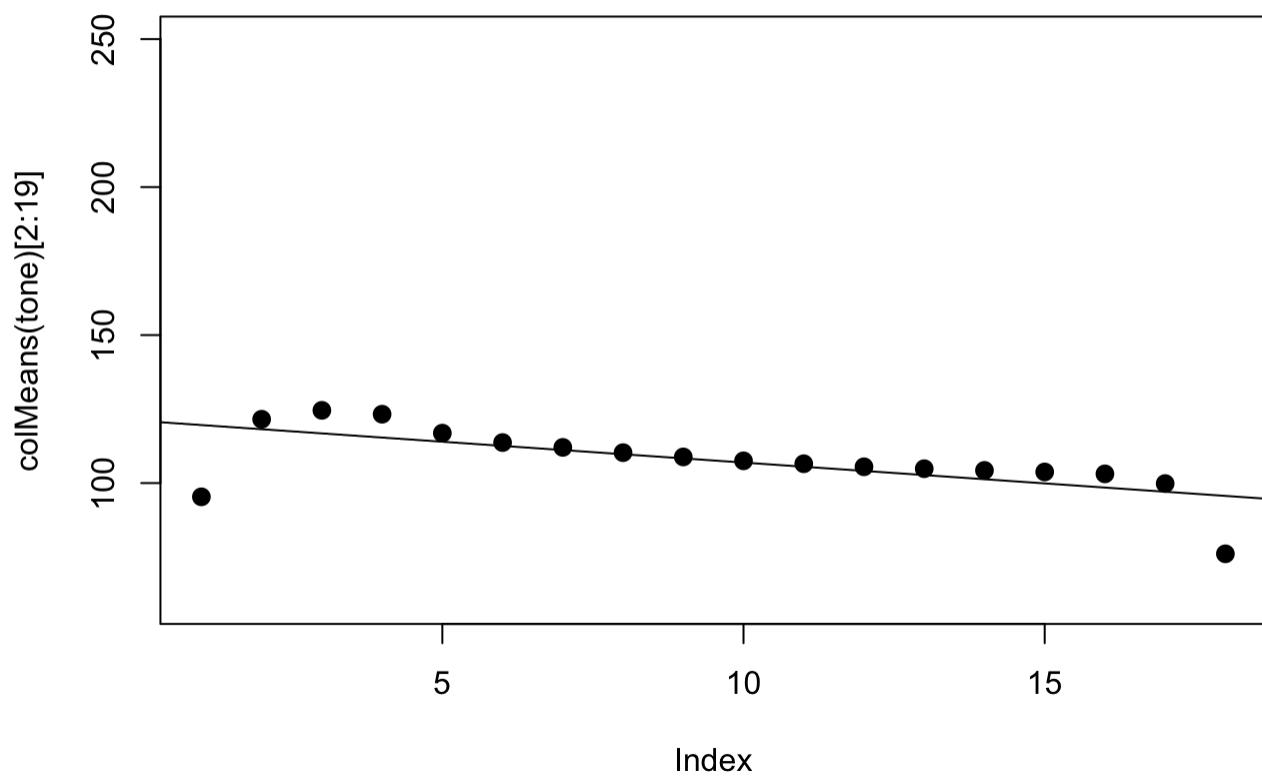
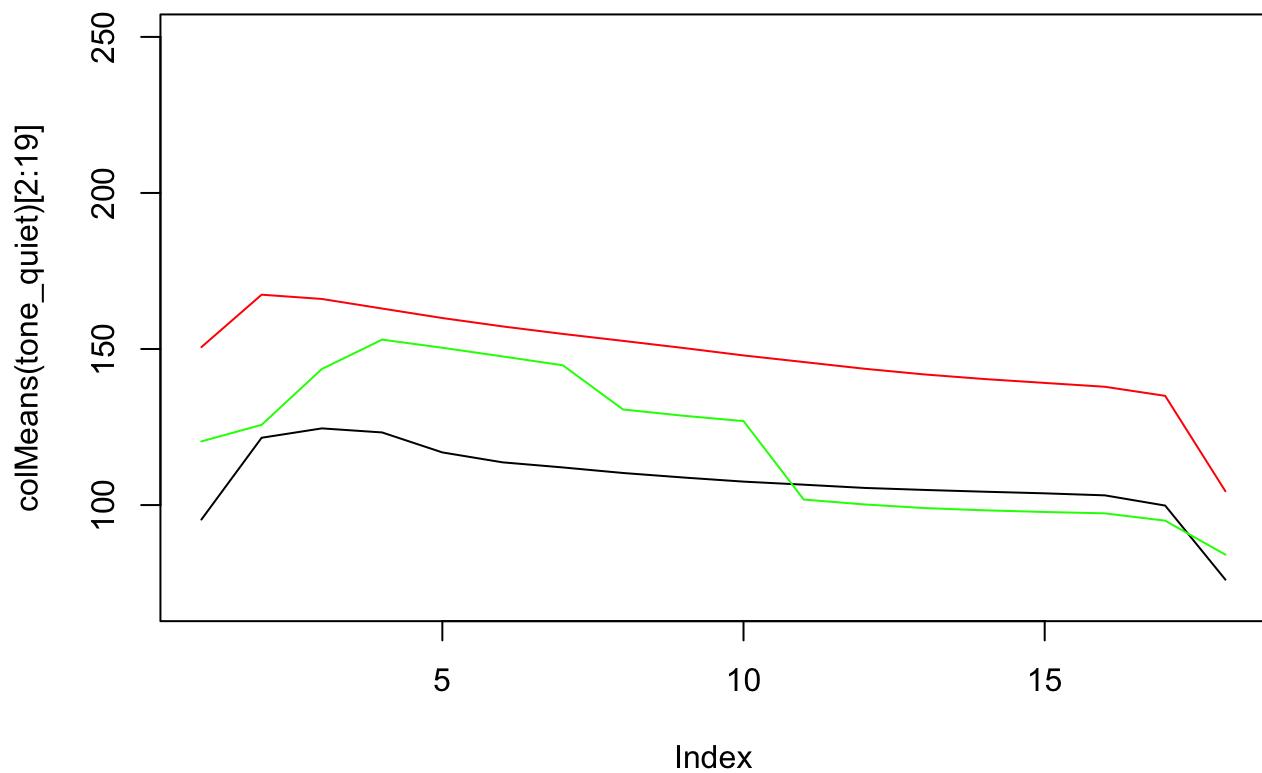



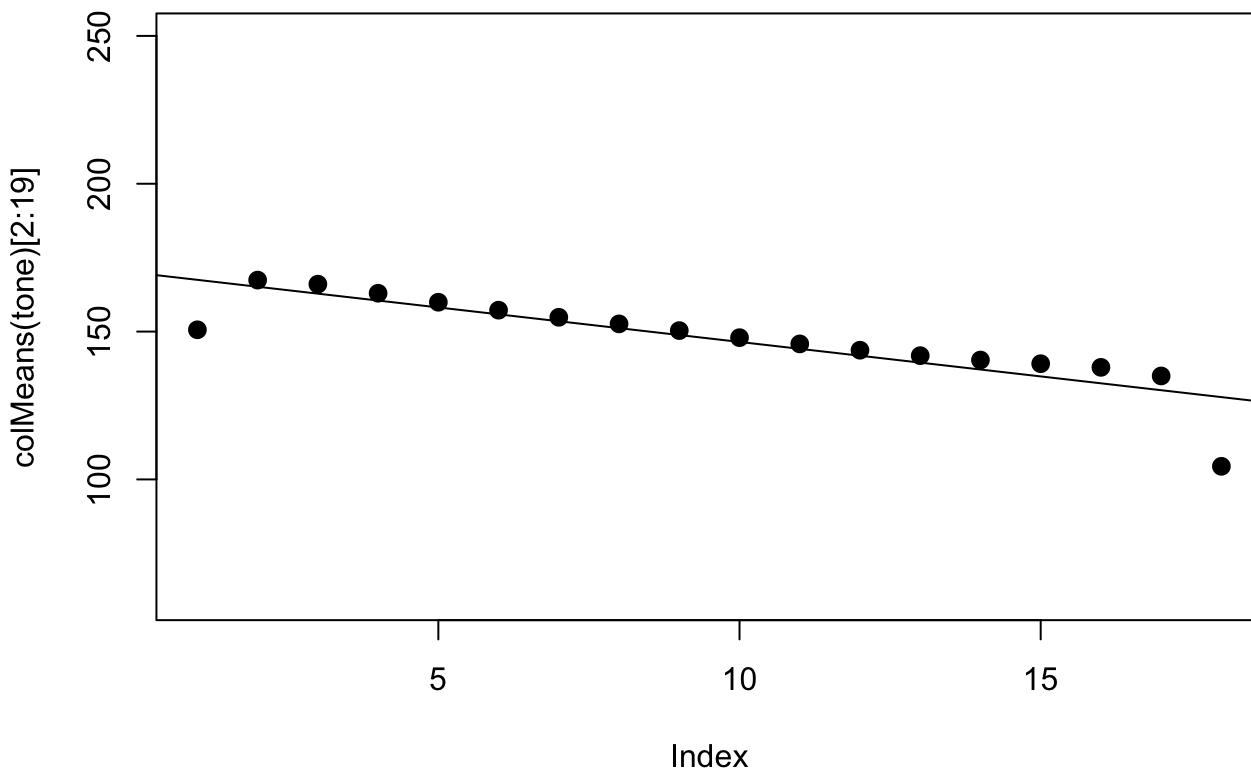
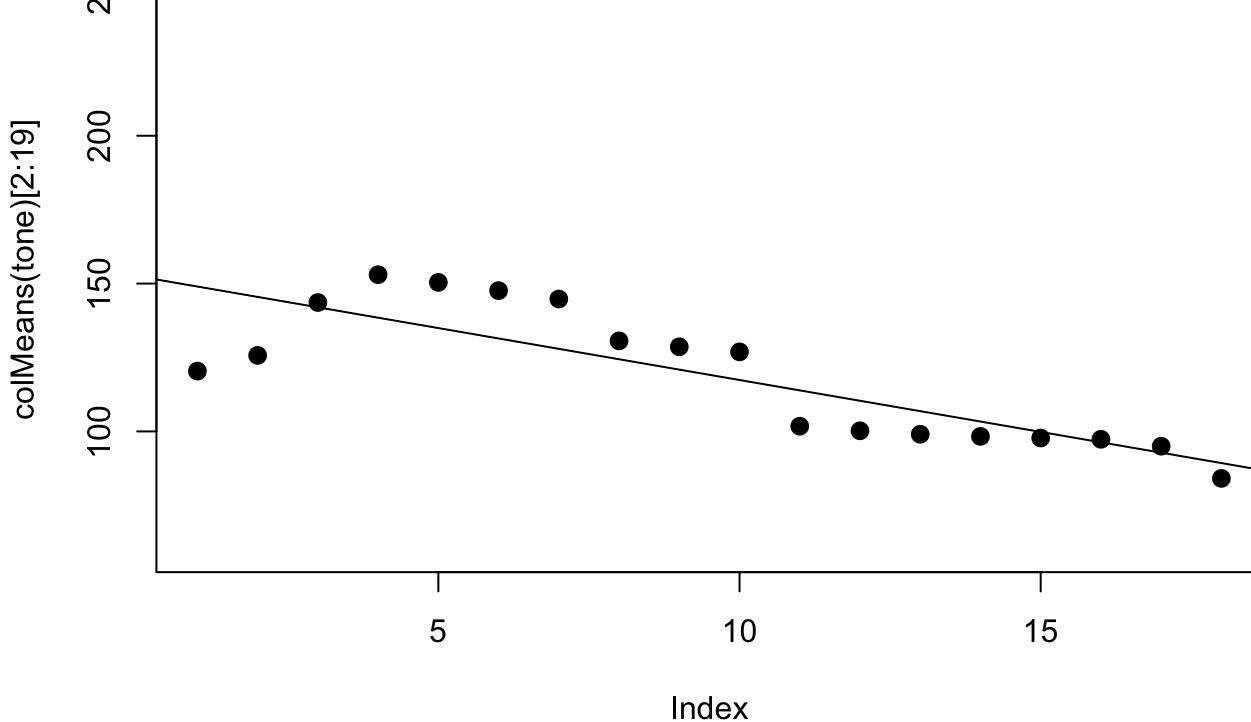


```
## [1] -2.5822926 -0.8887144 -1.6643078
```

Tone D2 on A

```
plot_f0_by_vowels("D2", "A")
```

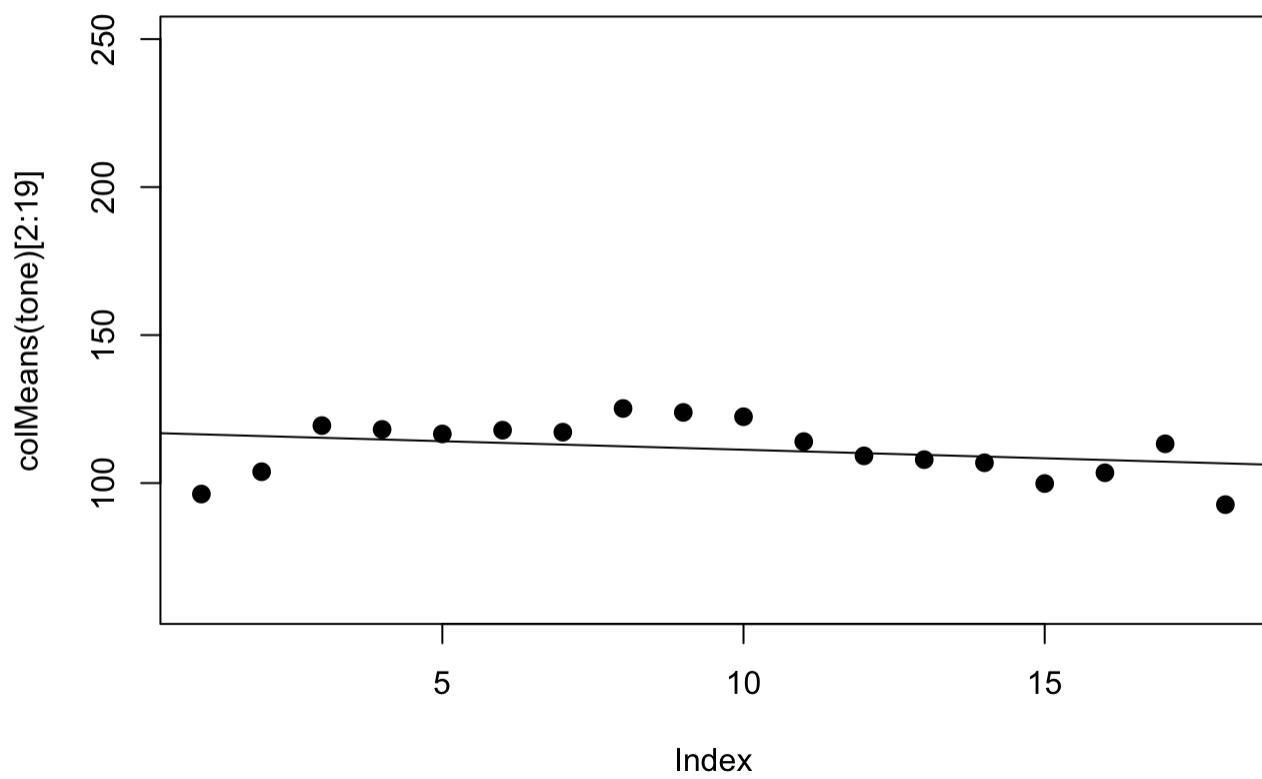
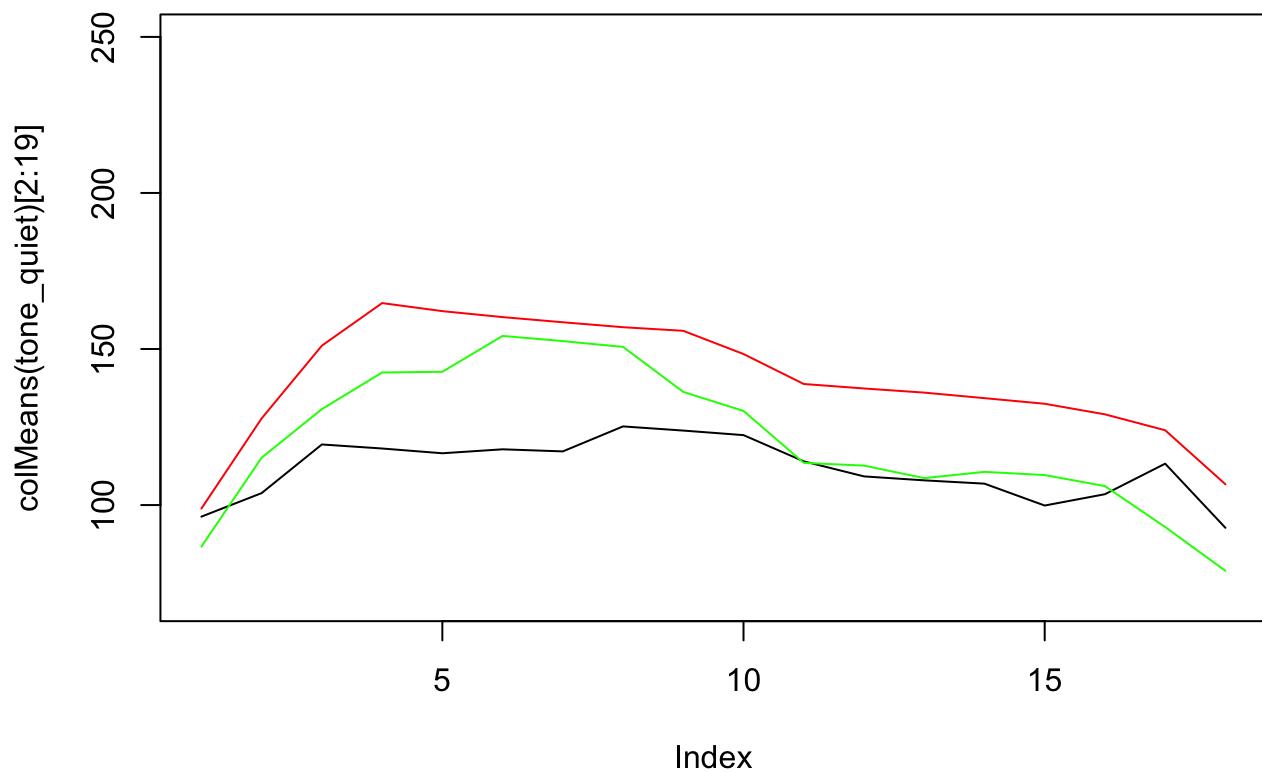



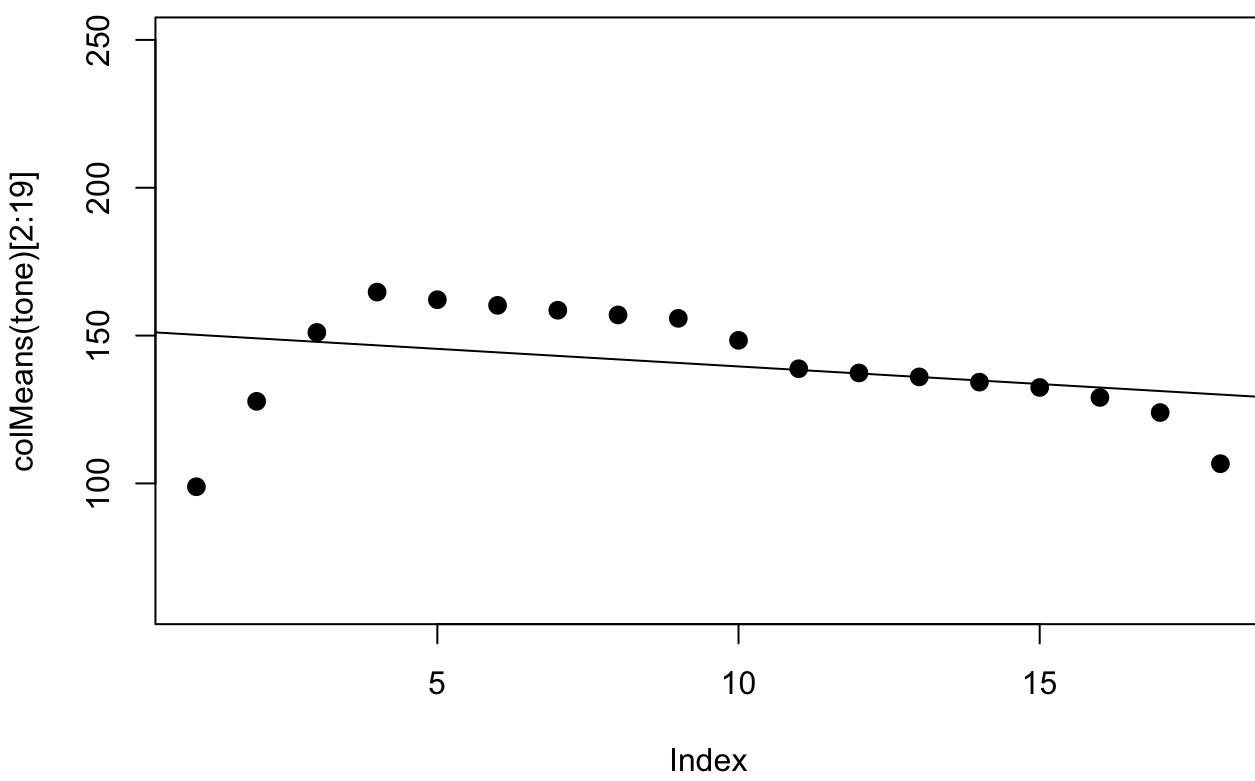
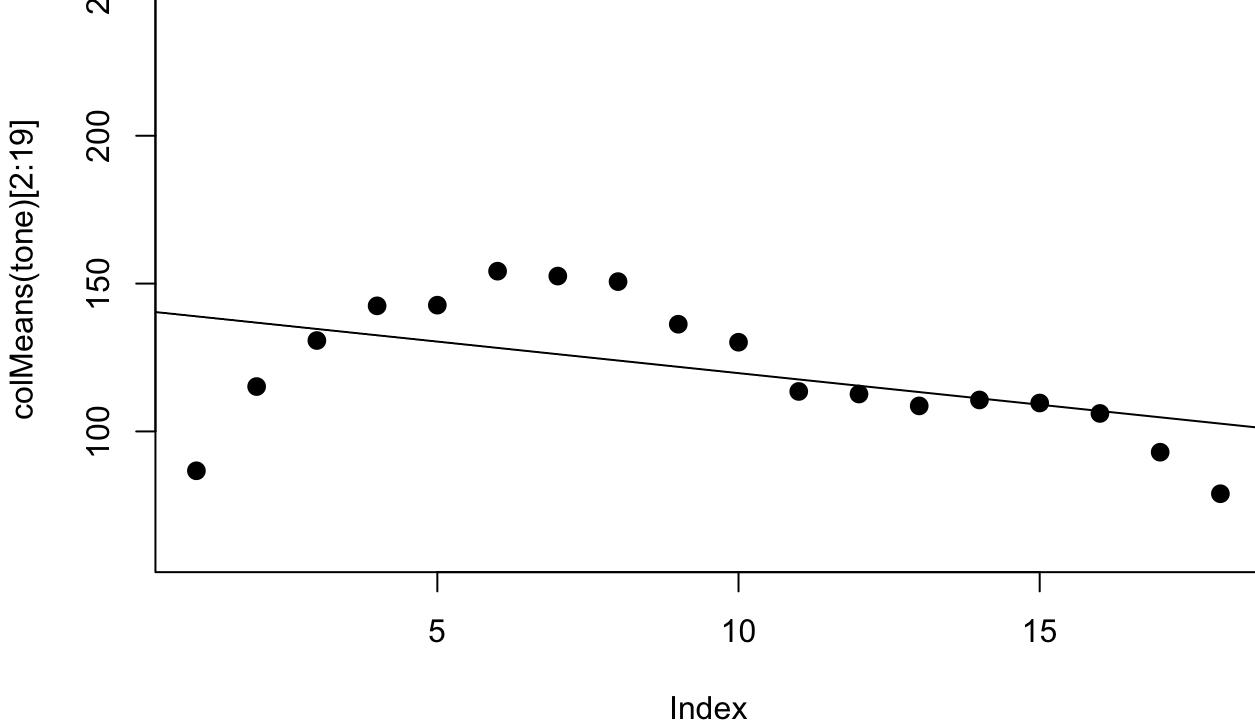


```
## [1] -1.411010 -3.515305 -2.333420
```

Tone D2 on E

```
plot_f0_by_vowels("D2", "E")
```

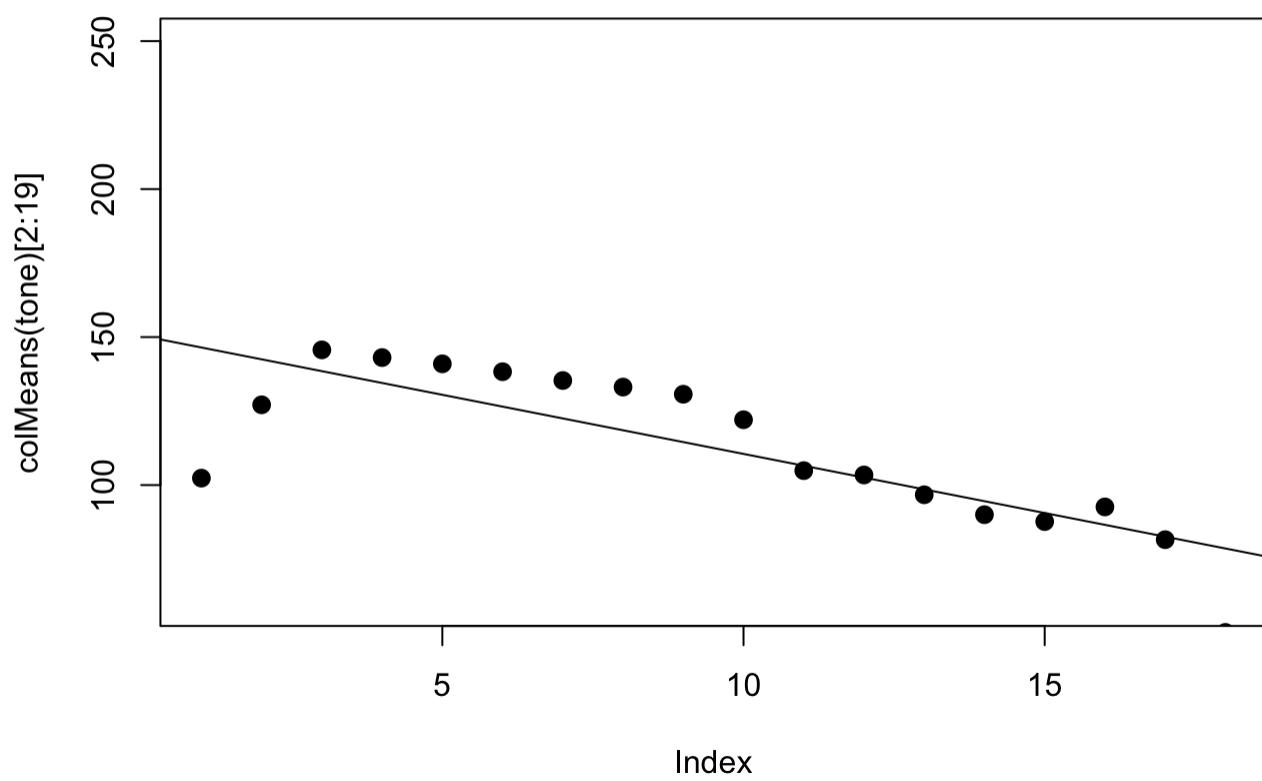
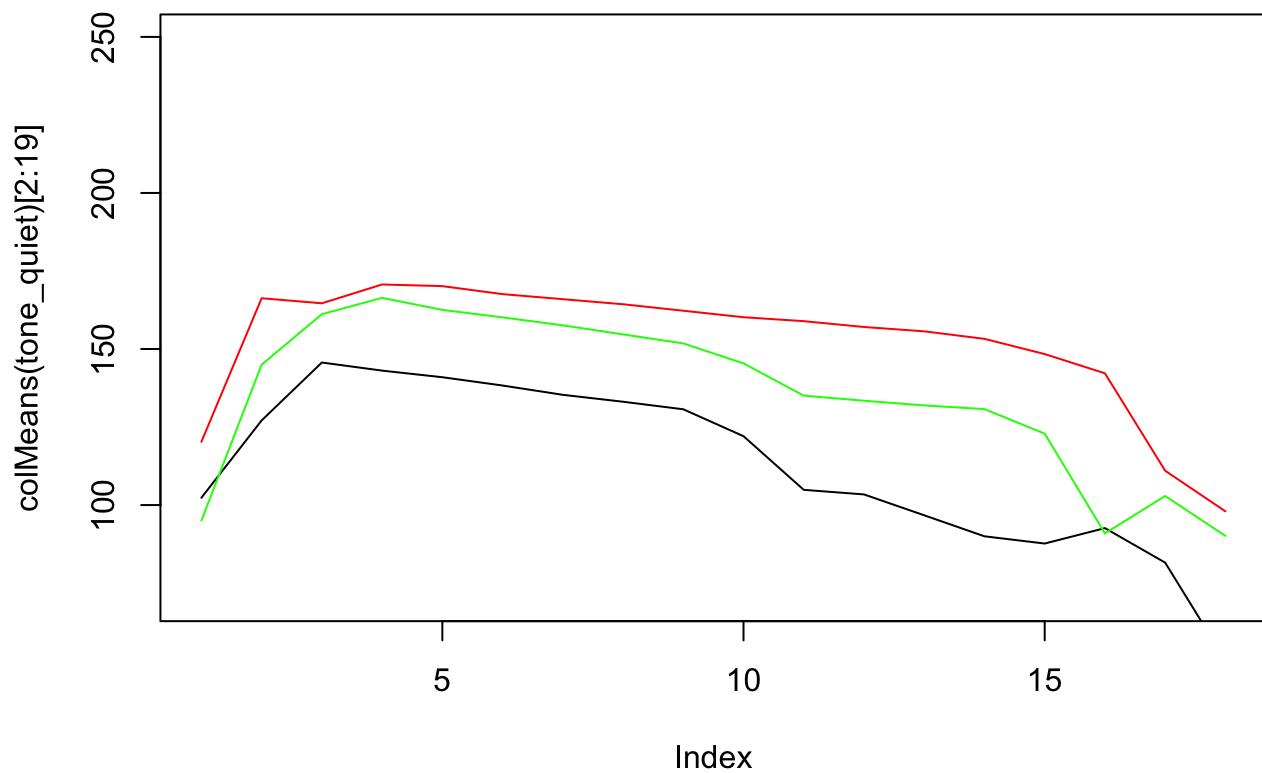



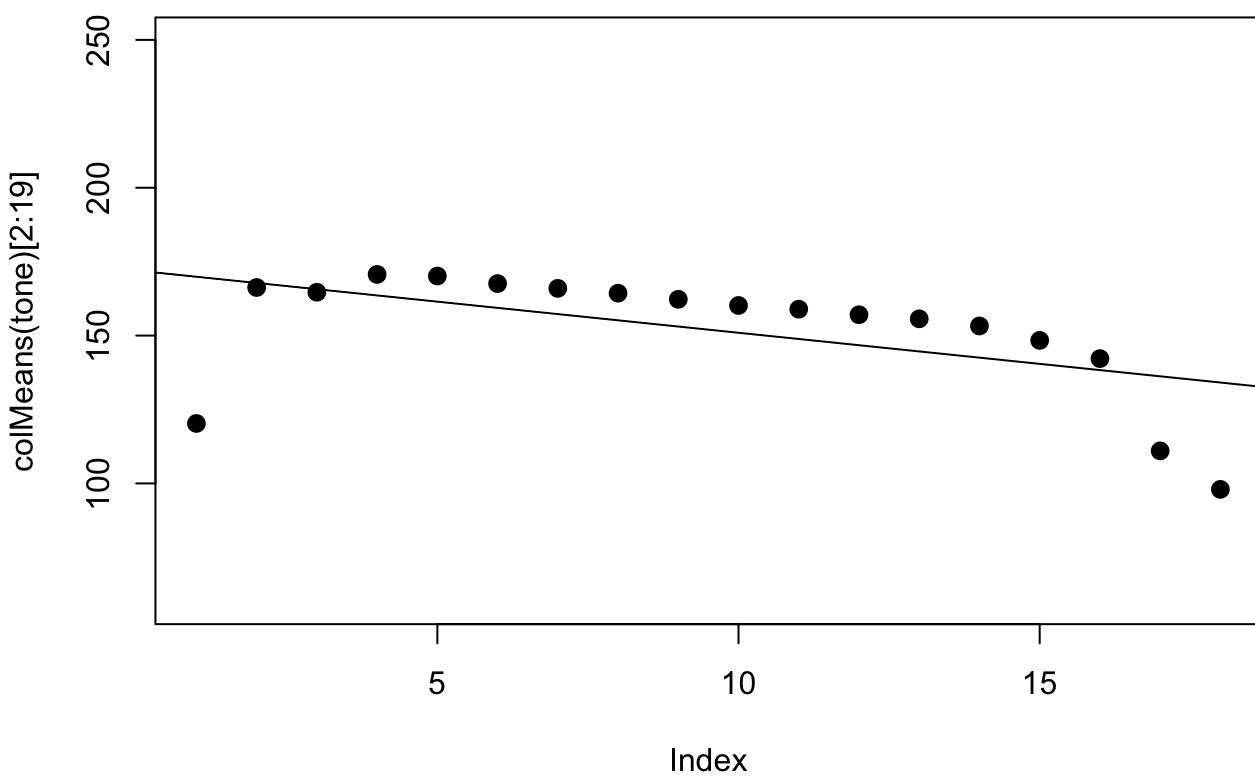
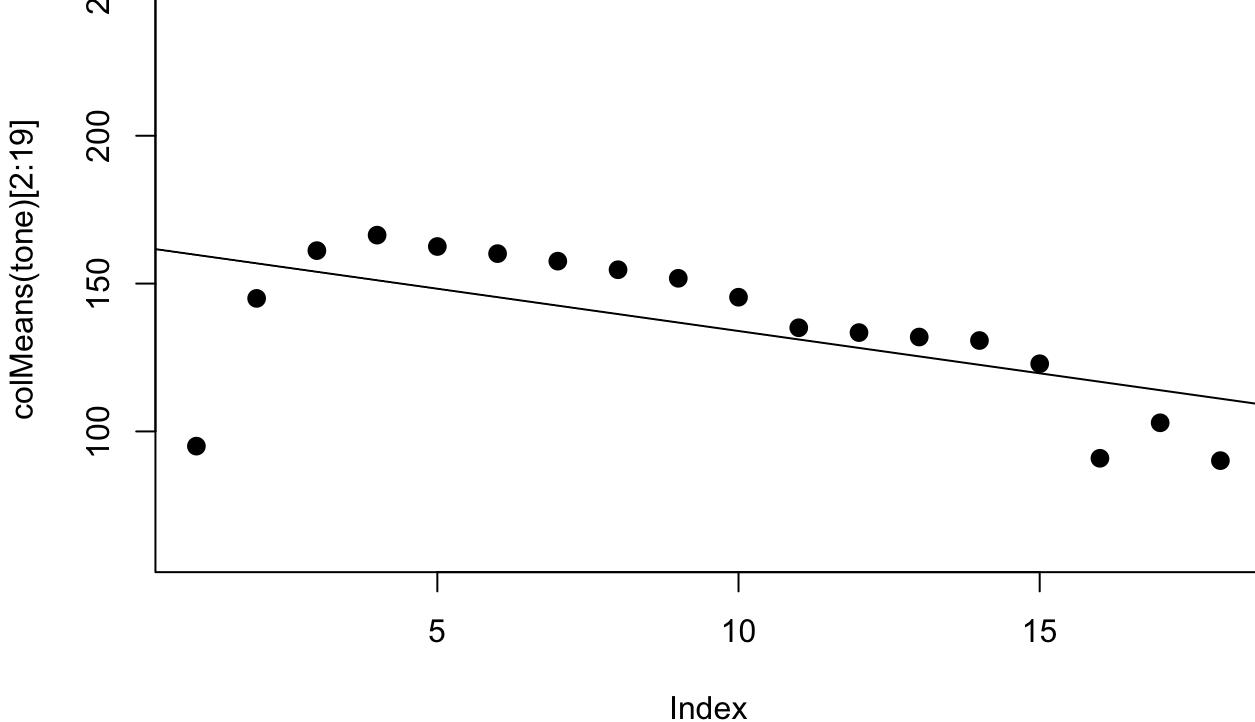


```
## [1] -0.5778161 -2.1378543 -1.1907024
```

Tone D2 on U

```
plot_f0_by_vowels("D2", "U")
```



```
## [1] -3.995236 -2.864344 -2.108659
```

Slope coefficients in a table

Plot F0 contours according to different levels

```

filter_f0 <- function(tone, vowel, noise) {
  tone_matrix <- data.matrix(f0_reports[f0_reports$tone==tone & f0_reports$noise==noise & f0_reports$vowel==vowel,-c(1,c(22:26))])
  return(tone_matrix)
}

# Table of tone * noise (24x3)
tab_vowel <- matrix(nrow=24, ncol=3, byrow=TRUE)
colnames(tab_vowel) <- c('quiet','noise 78','noise 90')
rownames(tab_vowel) <- c('A1 A','A2 A','B1 A', 'B2 A',
                          'C1 A','C2 A','D1 A', 'D2 A',
                          'A1 E','A2 E','B1 E', 'B2 E',
                          'C1 E','C2 E','D1 E', 'D2 E',
                          'A1 U','A2 U','B1 U', 'B2 U',
                          'C1 U','C2 U','D1 U', 'D2 U')
tab_vowel <- as.table(tab_vowel)
tab_vowel[1,1] = coefficient_reports(filter_f0("A1", "A", "0"))
tab_vowel[1,2] = coefficient_reports(filter_f0("A1", "A", "78"))
tab_vowel[1,3] = coefficient_reports(filter_f0("A1", "A", "90"))

tab_vowel[2,1] = coefficient_reports(filter_f0("A2", "A", "0"))
tab_vowel[2,2] = coefficient_reports(filter_f0("A2", "A", "78"))
tab_vowel[2,3] = coefficient_reports(filter_f0("A2", "A", "90"))

tab_vowel[3,1] = coefficient_reports(filter_f0("B1", "A", "0"))
tab_vowel[3,2] = coefficient_reports(filter_f0("B1", "A", "78"))
tab_vowel[3,3] = coefficient_reports(filter_f0("B1", "A", "90"))

tab_vowel[4,1] = coefficient_reports(filter_f0("B2", "A", "0"))
tab_vowel[4,2] = coefficient_reports(filter_f0("B2", "A", "78"))
tab_vowel[4,3] = coefficient_reports(filter_f0("B2", "A", "90"))

tab_vowel[5,1] = coefficient_reports(filter_f0("C1", "A", "0"))
tab_vowel[5,2] = coefficient_reports(filter_f0("C1", "A", "78"))
tab_vowel[5,3] = coefficient_reports(filter_f0("C1", "A", "90"))

tab_vowel[6,1] = coefficient_reports(filter_f0("C2", "A", "0"))
tab_vowel[6,2] = coefficient_reports(filter_f0("C2", "A", "78"))
tab_vowel[6,3] = coefficient_reports(filter_f0("C2", "A", "90"))

tab_vowel[7,1] = coefficient_reports(filter_f0("D1", "A", "0"))
tab_vowel[7,2] = coefficient_reports(filter_f0("D1", "A", "78"))
tab_vowel[7,3] = coefficient_reports(filter_f0("D1", "A", "90"))

tab_vowel[8,1] = coefficient_reports(filter_f0("D2", "A", "0"))
tab_vowel[8,2] = coefficient_reports(filter_f0("D2", "A", "78"))
tab_vowel[8,3] = coefficient_reports(filter_f0("D2", "A", "90"))

tab_vowel[9,1] = coefficient_reports(filter_f0("A1", "E", "0"))
tab_vowel[9,2] = coefficient_reports(filter_f0("A1", "E", "78"))
tab_vowel[9,3] = coefficient_reports(filter_f0("A1", "E", "90"))

tab_vowel[10,1] = coefficient_reports(filter_f0("A2", "E", "0"))
tab_vowel[10,2] = coefficient_reports(filter_f0("A2", "E", "78"))
tab_vowel[10,3] = coefficient_reports(filter_f0("A2", "E", "90"))

tab_vowel[11,1] = coefficient_reports(filter_f0("B1", "E", "0"))
tab_vowel[11,2] = coefficient_reports(filter_f0("B1", "E", "78"))

```

```
tab_vowel[11,3] = coefficient_reports(filter_f0("B1", "E", "90"))

tab_vowel[12,1] = coefficient_reports(filter_f0("B2", "E", "0"))
tab_vowel[12,2] = coefficient_reports(filter_f0("B2", "E", "78"))
tab_vowel[12,3] = coefficient_reports(filter_f0("B2", "E", "90"))

tab_vowel[13,1] = coefficient_reports(filter_f0("C1", "E", "0"))
tab_vowel[13,2] = coefficient_reports(filter_f0("C1", "E", "78"))
tab_vowel[13,3] = coefficient_reports(filter_f0("C1", "E", "90"))

tab_vowel[14,1] = coefficient_reports(filter_f0("C2", "E", "0"))
tab_vowel[14,2] = coefficient_reports(filter_f0("C2", "E", "78"))
tab_vowel[14,3] = coefficient_reports(filter_f0("C2", "E", "90"))

tab_vowel[15,1] = coefficient_reports(filter_f0("D1", "E", "0"))
tab_vowel[15,2] = coefficient_reports(filter_f0("D1", "E", "78"))
tab_vowel[15,3] = coefficient_reports(filter_f0("D1", "E", "90"))

tab_vowel[16,1] = coefficient_reports(filter_f0("D2", "E", "0"))
tab_vowel[16,2] = coefficient_reports(filter_f0("D2", "E", "78"))
tab_vowel[16,3] = coefficient_reports(filter_f0("D2", "E", "90"))

tab_vowel[17,1] = coefficient_reports(filter_f0("A1", "U", "0"))
tab_vowel[17,2] = coefficient_reports(filter_f0("A1", "U", "78"))
tab_vowel[17,3] = coefficient_reports(filter_f0("A1", "U", "90"))

tab_vowel[18,1] = coefficient_reports(filter_f0("A2", "U", "0"))
tab_vowel[18,2] = coefficient_reports(filter_f0("A2", "U", "78"))
tab_vowel[18,3] = coefficient_reports(filter_f0("A2", "U", "90"))

tab_vowel[19,1] = coefficient_reports(filter_f0("B1", "U", "0"))
tab_vowel[19,2] = coefficient_reports(filter_f0("B1", "U", "78"))
tab_vowel[19,3] = coefficient_reports(filter_f0("B1", "U", "90"))

tab_vowel[20,1] = coefficient_reports(filter_f0("B2", "U", "0"))
tab_vowel[20,2] = coefficient_reports(filter_f0("B2", "U", "78"))
tab_vowel[20,3] = coefficient_reports(filter_f0("B2", "U", "90"))

tab_vowel[21,1] = coefficient_reports(filter_f0("C1", "U", "0"))
tab_vowel[21,2] = coefficient_reports(filter_f0("C1", "U", "78"))
tab_vowel[21,3] = coefficient_reports(filter_f0("C1", "U", "90"))

tab_vowel[22,1] = coefficient_reports(filter_f0("C2", "U", "0"))
tab_vowel[22,2] = coefficient_reports(filter_f0("C2", "U", "78"))
tab_vowel[22,3] = coefficient_reports(filter_f0("C2", "U", "90"))

tab_vowel[23,1] = coefficient_reports(filter_f0("D1", "U", "0"))
tab_vowel[23,2] = coefficient_reports(filter_f0("D1", "U", "78"))
tab_vowel[23,3] = coefficient_reports(filter_f0("D1", "U", "90"))

tab_vowel[24,1] = coefficient_reports(filter_f0("D2", "U", "0"))
tab_vowel[24,2] = coefficient_reports(filter_f0("D2", "U", "78"))
tab_vowel[24,3] = coefficient_reports(filter_f0("D2", "U", "90"))

tab_vowel
```

```
## quiet noise 78 noise 90
## A1 A -0.28786597 -0.24247753 -0.30183718
## A2 A -0.94189377 -1.21715432 -1.04074426
## B1 A 3.29930310 4.44473319 4.26599052
## B2 A -3.68320100 -5.22430946 -6.27214142
## C1 A -2.13631231 -2.06368135 -2.75352823
## C2 A 3.01033315 3.02702809 2.39415449
## D1 A 1.94669799 1.57254633 2.11046287
## D2 A -1.41100990 -3.51530530 -2.33342025
## A1 E -0.29252540 -0.84861170 -0.69958066
## A2 E -1.19115023 -1.48746256 -1.65442259
## B1 E 3.59328024 4.72965482 4.46913289
## B2 E -3.54701137 -5.91615578 -5.92709176
## C1 E -2.07185516 -2.16034262 -3.66593772
## C2 E 3.10283002 1.83602960 1.09639647
## D1 E 0.64208026 4.61889193 2.26070381
## D2 E -0.57781611 -2.13785431 -1.19070239
## A1 U -1.28368202 -0.94018817 -0.70906976
## A2 U -1.62964409 -1.85532732 -2.03872891
## B1 U 0.26794278 2.21307069 3.39750573
## B2 U -4.08087893 -6.07211141 -6.29807801
## C1 U -1.95566596 -3.13011073 -3.87637567
## C2 U 1.62192705 1.20632671 -0.03604034
## D1 U -2.58229256 -0.88871437 -1.66430780
## D2 U -3.99523644 -2.86434374 -2.10865905
```