Jun23

Xinyu Zhang

6/23/2020

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

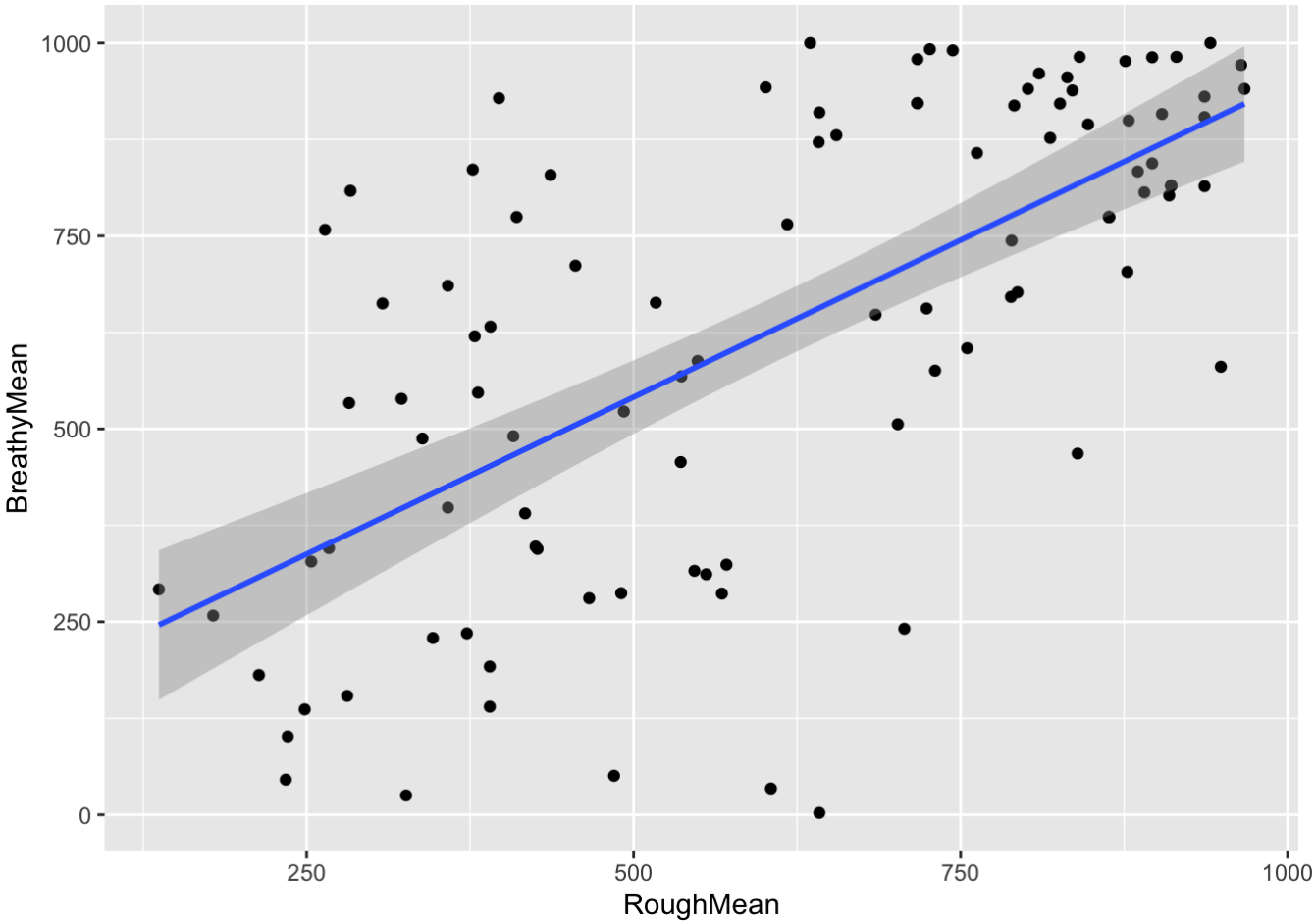
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
av_AVQI_B <- read.csv("av_AVQI_B.csv")
av_AVQI_R <- read.csv("av_AVQI_R.csv")
#merging mean breahtiness and mean roughness into one table
av_AVQI_BR <-left_join(av_AVQI_B,av_AVQI_R, by = "dummy") %>%
  glimpse
```

```
## Rows: 106
## Columns: 21
## $ X.x         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17…
## $ dummy       <fct> S9 - P, S18 - Pre, S34 - Pre, S6 - P6mnd, S38 - Pre, S17 …
## $ BreathyMean <dbl> 235.0, 1000.0, 2.5, 940.5, 136.5, 806.5, 328.0, 345.5, 88…
## $ CPPS.x      <dbl> 7.02, 13.51, 2.75, 15.37, 6.86, 12.57, 8.55, 10.65, 13.47…
## $ HNR.x       <dbl> 7.80, 16.86, 4.58, 23.69, 7.07, 19.70, 9.68, 13.98, 15.28…
## $ Jitter.x    <dbl> 1.98, 0.91, NA, 0.40, 2.89, 0.93, 1.96, 0.89, 0.39, 1.13,…
## $ Shimmer.x   <dbl> 13.20, 7.40, 20.61, 3.84, 12.02, 5.23, 14.47, 6.59, 6.23,…
## $ ShdB.x      <dbl> 1.18, 0.70, 1.74, 0.39, 1.14, 0.53, 1.27, 0.61, 0.56, 0.4…
## $ Slope.x     <dbl> -16.76, -22.19, -10.42, -21.07, -17.61, -26.12, -21.37, -…
## $ Tilt.x      <dbl> -7.69, -12.10, -10.15, -12.43, -10.85, -11.68, -9.29, -9.…
## $ Version.x   <dbl> 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.0…
## $ X.y         <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17…
## $ RoughMean   <dbl> 372.5, 941.0, 642.0, 967.0, 248.5, 890.5, 253.5, 267.0, 6…
## $ CPPS.y      <dbl> 7.02, 13.51, 2.75, 15.37, 6.86, 12.57, 8.55, 10.65, 13.47…
## $ HNR.y       <dbl> 7.80, 16.86, 4.58, 23.69, 7.07, 19.70, 9.68, 13.98, 15.28…
## $ Jitter.y    <dbl> 1.98, 0.91, NA, 0.40, 2.89, 0.93, 1.96, 0.89, 0.39, 1.13,…
## $ Shimmer.y   <dbl> 13.20, 7.40, 20.61, 3.84, 12.02, 5.23, 14.47, 6.59, 6.23,…
## $ ShdB.y      <dbl> 1.18, 0.70, 1.74, 0.39, 1.14, 0.53, 1.27, 0.61, 0.56, 0.4…
## $ Slope.y     <dbl> -16.76, -22.19, -10.42, -21.07, -17.61, -26.12, -21.37, -…
## $ Tilt.y      <dbl> -7.69, -12.10, -10.15, -12.43, -10.85, -11.68, -9.29, -9.…
## $ Version.y   <dbl> 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.03, 2.0…
```

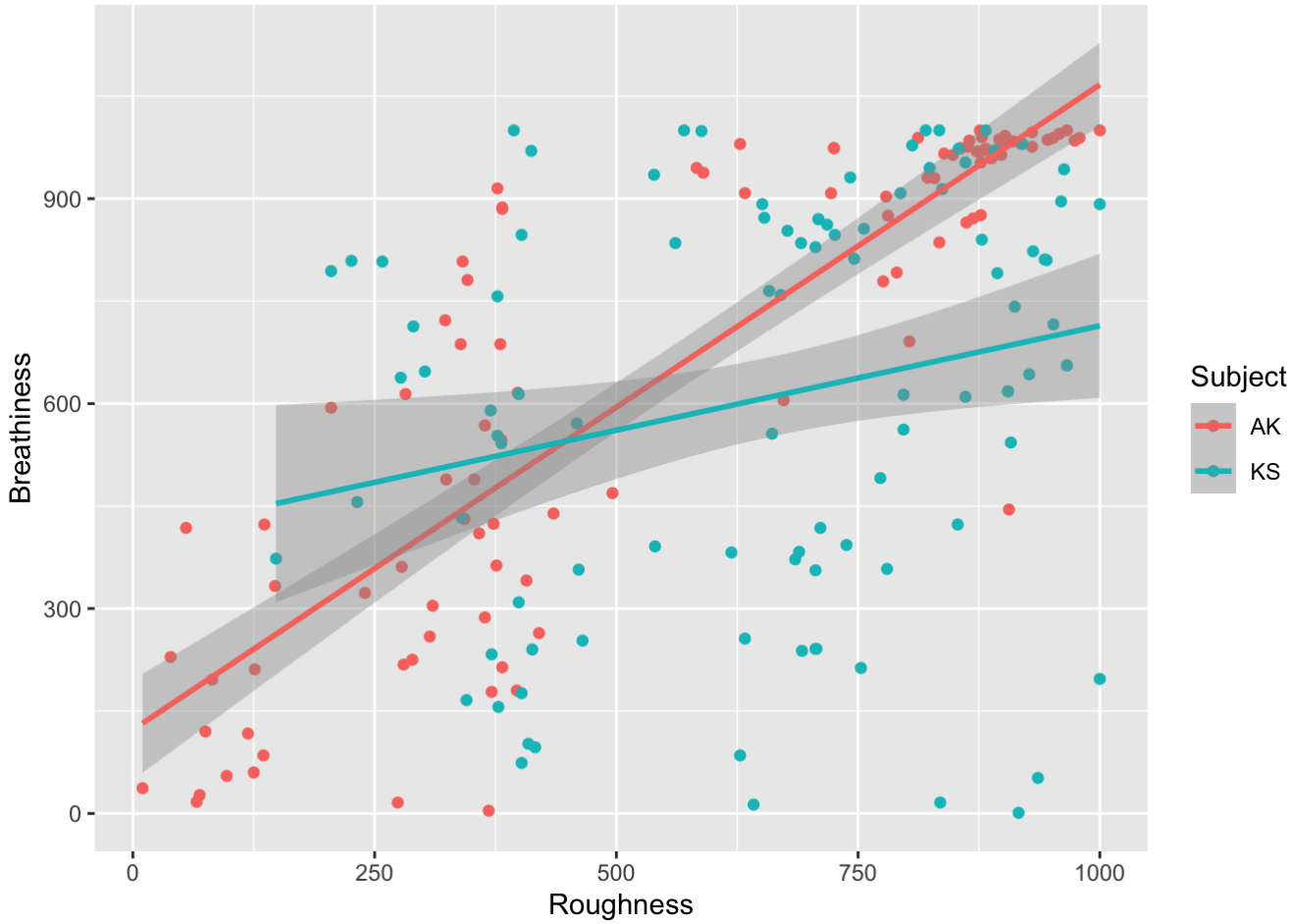
```
#plotting and fitting regression line
ggplot(data = av_AVQI_BR, aes(x = RoughMean, y = BreathyMean)) +
  geom_point() +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
#Comparing the above with the un-averaged data
merged <- read.csv("merged.csv")
library(ggplot2)
ggplot(data = merged, aes(x = Roughness, y = Breathiness, color = Subject)) +
  geom_point() +
  geom_smooth(method = "lm")
```

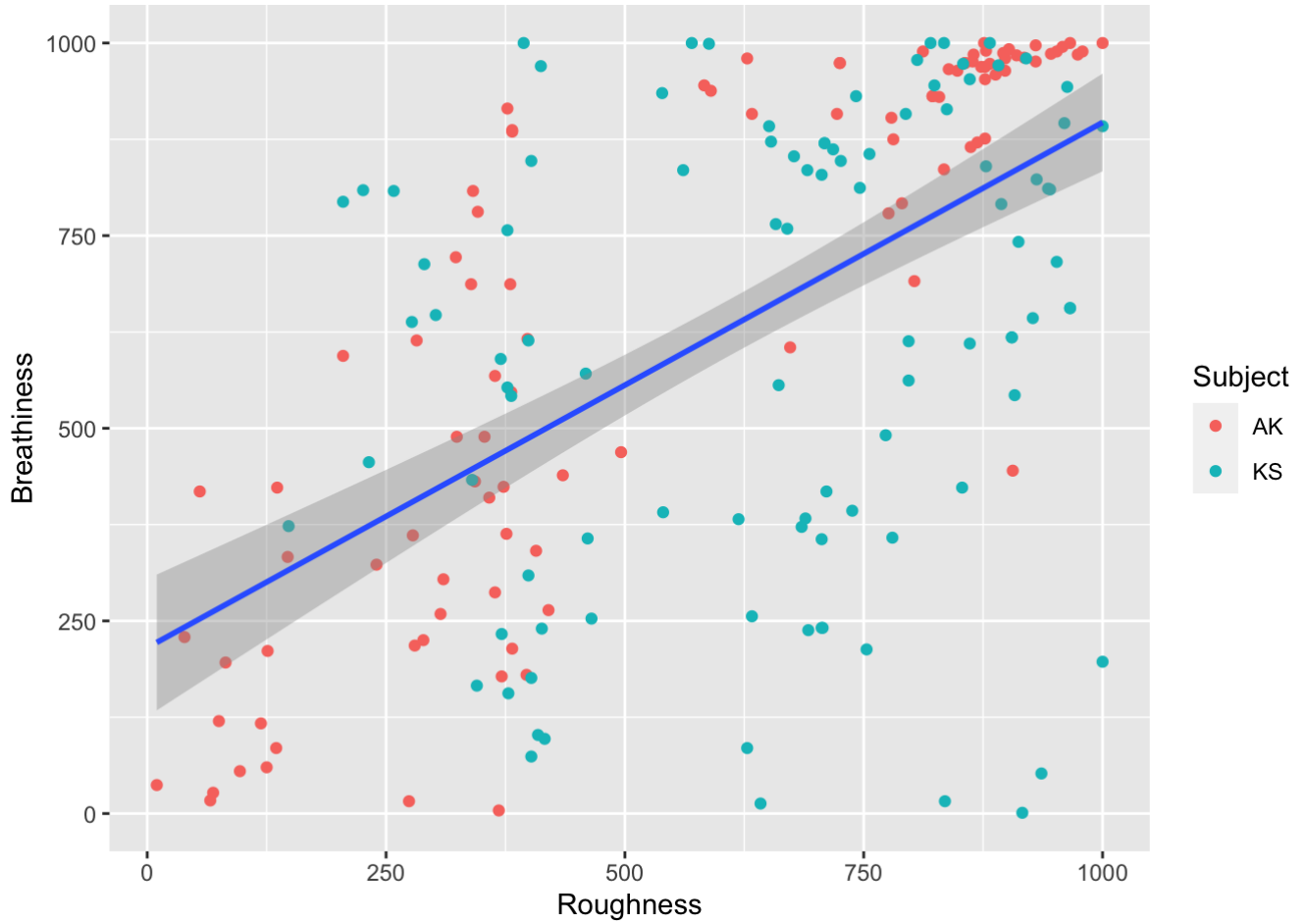
```
## `geom_smooth()` using formula 'y ~ x'
```




```
#ggsave("roughness-breathiness.pdf", plot = last_plot(), device = NULL)
```

```
#Only fitting one line over all points, regardless of subjects
merged <- read.csv("merged.csv")
library(ggplot2)
ggplot(data = merged, aes(x = Roughness, y = Breathiness)) +
  geom_point(aes(color = Subject)) +
  geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



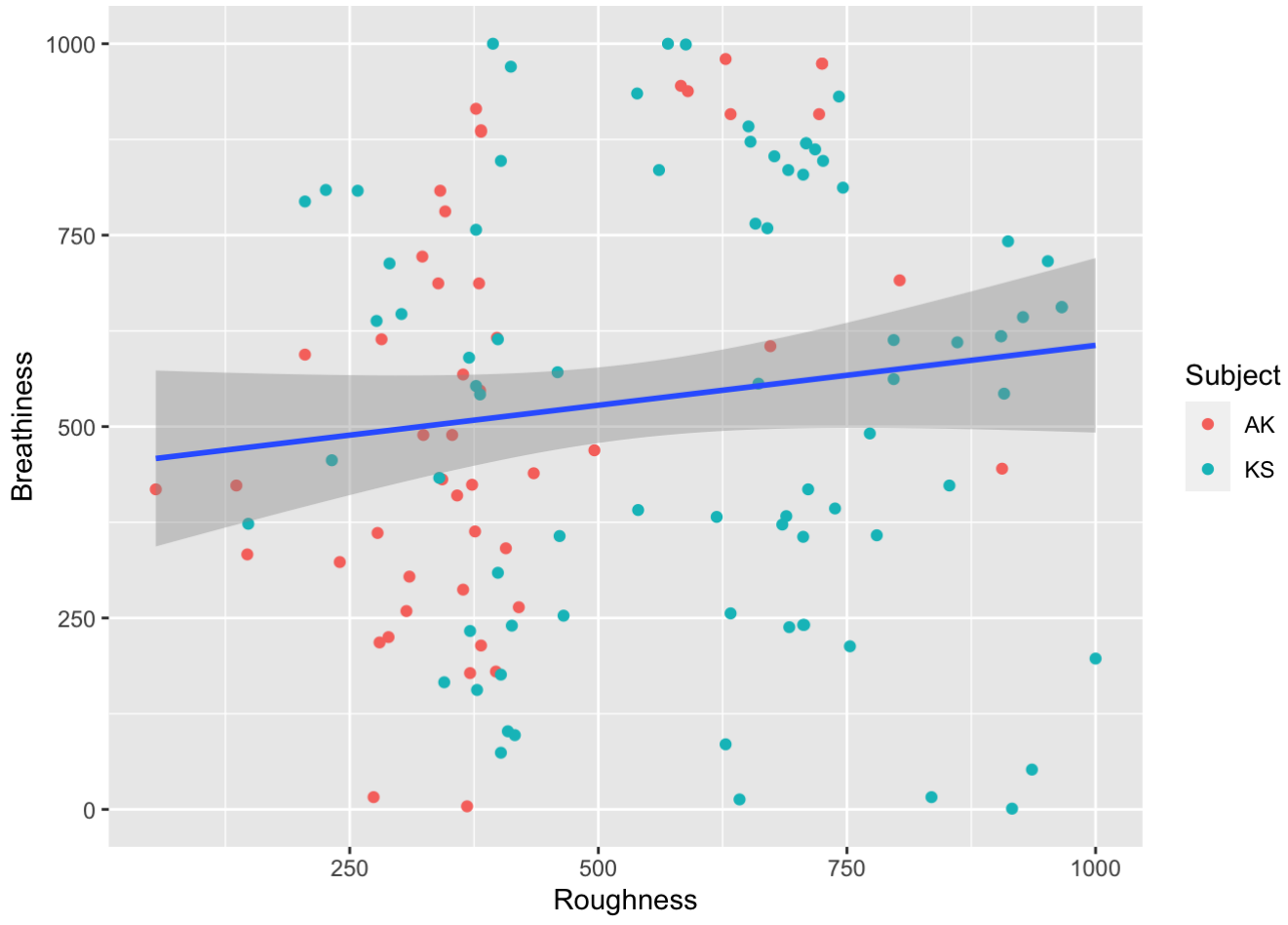

```
#ggsave("roughness-breathiness.pdf", plot = last_plot(), device = NULL)
```

```
#excluding the upper-right-most and lower-left-most squares
merged2 <- merged %>%
  filter(!(Roughness > 750 & Breathiness > 750) & !(Roughness < 250 & Breathiness <250))
%>%
  glimpse
```

```
## Rows: 127
## Columns: 15
## $ X           <int> 3, 7, 9, 10, 12, 13, 18, 19, 21, 24, 26, 27, 28, 31, 32, …
## $ Roughness   <int> 368, 136, 633, 282, 55, 382, 307, 398, 380, 280, 278, 310…
## $ Breathiness <int> 4, 423, 908, 614, 418, 885, 259, 616, 687, 218, 361, 304,…
## $ Speaker     <fct> S34, S9, S10, S46, S1, S29, S38, S44, S25, S5, S10, S43, …
## $ T.x         <fct> Pre, Pre, P6mnd, P6mnd, Pre, Pre, P6mnd, Pre, Pre, Pre, P…
## $ Subject     <fct> AK, AK, AK, AK, AK, AK, AK, AK, AK, AK, AK, AK, AK, AK, A…
## $ token       <fct> S34 - Pre, S9 - Pre, S10 - P6mnd, S46 - P6mnd, S1 - Pre, …
## $ AVQI        <dbl> 7.67, 6.36, 3.36, 5.62, NA, 3.30, 7.11, 2.63, 4.23, 4.19,…
## $ CPPS        <dbl> 2.75, 8.55, 13.47, 8.67, NA, 12.36, 7.13, 13.72, 11.60, 1…
## $ HNR         <dbl> 4.58, 9.68, 15.28, 10.80, NA, 20.41, 4.20, 19.08, 17.45, …
## $ Jitter      <fct> --undefined--, 1.96, 0.39, 1.13, NA, 0.43, 1.21, 0.31, 0.…
## $ Shimmer     <dbl> 20.61, 14.47, 6.23, 4.70, NA, 4.17, 12.95, 3.84, 5.81, 5.…
## $ ShdB        <dbl> 1.74, 1.27, 0.56, 0.48, NA, 0.38, 1.19, 0.37, 0.56, 0.50,…
## $ Slope       <dbl> -10.42, -21.37, -20.79, -19.10, NA, -25.90, -12.63, -21.1…
## $ Tilt        <dbl> -10.15, -9.29, -11.42, -10.01, NA, -9.94, -9.92, -11.76, …
```

```
ggplot(data = merged2, aes(x = Roughness, y = Breathiness))+
         geom_point(aes(color = Subject))+
         geom_smooth(method = "lm")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
#not much correlation here anymore
```

```
library(dplyr)
#reading in listening test results
listener1 <- read.csv("Results_AK.csv", header = TRUE, sep = ";")
listener2 <- read.csv("Results_KS.csv", header = TRUE, sep = ";")
#renaming the two answers, removing column "Num"
SLP1 <- listener1 %>%
  rename(Roughness = Answer1, Breathiness = Answer2)%>%
  dplyr::select(-Num)

SLP2 <- listener2 %>%
  rename(Roughness = Answer1, Breathiness = Answer2)%>%
  dplyr::select(-Num)

#inserting a column with Speaker and T merged together in the listening experiment resul
ts table
#SLP1$dummy <- paste(SLP1$Speaker, "-", SLP1$T)
#SLP2$dummy <- paste(SLP2$Speaker, "-", SLP2$T)

#as_tibble(SLP1)
#as_tibble(SLP2)

#join the two listening results tables by matching both Speaker and T
SLP <-rbind(SLP1,SLP2) %>%
  dplyr::select(-Z, -A) %>%
  filter(!(Roughness > 750 & Breathiness > 750) & !(Roughness < 250 & Breathiness <250))

SLP$dummy <- paste(SLP$Speaker, "-", SLP$T)

str(SLP)
```

```
## 'data.frame':    123 obs. of  6 variables:
##  $ Roughness  : int  368 136 633 282 55 382 307 398 380 280 ...
##  $ Breathiness: int  4 423 908 614 418 885 259 616 687 218 ...
##  $ Speaker    : Factor w/ 41 levels "S1","S10","S11",..: 27 41 2 37 1 22 29 35 18 38
...
##  $ T          : Factor w/ 10 levels "-","P","P1","P12mnd",..: 9 9 8 8 9 9 8 9 9 9 ...
##  $ Subject    : Factor w/ 2 levels "AK","KS": 1 1 1 1 1 1 1 1 1 1 ...
##  $ dummy      : chr  "S34 - Pre" "S9 - Pre" "S10 - P6mnd" "S46 - P6mnd" ...
```

```
AVQI <- read.csv("AVQI_results_new.csv")
#create a dummy column as an identifier for each "Speaker-T" entry
AVQI$dummy <- paste(AVQI$Speaker, "-", AVQI$T)

SLPRough <- SLP %>%
  dplyr::select(Roughness, dummy)

#merging ("left-join" only appends AVQI entries that has SLP rating correspondents)
library(dplyr)
av_AVQI_R <-left_join(SLPRough,AVQI, by = "dummy")%>%
  dplyr::select(-Speaker, -T, -AVQI, -dummy)

#converting jitter into a factor
av_AVQI_R$Jitter <- as.numeric(levels(av_AVQI_R$Jitter))[av_AVQI_R$Jitter]
```

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

```
#have a look at the whole table
av_AVQI_R
```

```
##     Roughness  CPPS   HNR Jitter Shimmer ShdB  Slope   Tilt Version
## 1         368  2.75  4.58     NA   20.61 1.74 -10.42 -10.15    2.03
## 2         136  8.55  9.68   1.96   14.47 1.27 -21.37  -9.29    2.03
## 3         633 13.47 15.28   0.39    6.23 0.56 -20.79 -11.42    2.03
## 4         282  8.67 10.80   1.13    4.70 0.48 -19.10 -10.01    2.03
## 5          55    NA    NA     NA      NA   NA     NA     NA      NA
## 6         382 12.36 20.41   0.43    4.17 0.38 -25.90  -9.94    2.03
## 7         307  7.13  4.20   1.21   12.95 1.19 -12.63  -9.92    2.03
## 8         398 13.72 19.08   0.31    3.84 0.37 -21.13 -11.76    2.03
## 9         380 11.60 17.45   0.93    5.81 0.56 -26.57 -10.32    2.03
## 10        280 10.82 15.73   0.44    5.57 0.50 -21.40  -9.66    2.03
## 11        278  9.81 12.74   2.35   10.02 0.91 -23.62 -11.35    2.03
## 12        310  8.59 13.40   0.80    8.73 0.80 -24.35  -9.33    2.03
## 13        358 10.91 18.66   0.78    5.44 0.48 -26.53 -11.17    2.03
## 14        364 10.64 12.66   2.40    8.93 0.81 -21.52  -9.37    2.03
## 15        339  6.77 12.02   0.86   11.24 0.96 -23.67  -9.98    2.03
## 16        583 13.83 16.65   0.74    6.40 0.48 -21.75 -11.60    2.03
## 17        803 11.34 20.85   0.55    4.73 0.43 -27.35 -11.09    2.03
## 18        240  8.27 12.42   1.05    9.93 0.77 -21.09 -10.61    2.03
## 19        673 14.21 20.79   0.36    4.64 0.49 -22.40 -11.31    2.03
## 20        373  8.48  9.31   0.94    9.80 0.89 -17.35 -10.47    2.03
## 21        420  7.81 12.09   2.06   10.63 0.93 -23.14  -8.36    2.03
## 22        323 10.96 10.00   0.51    6.87 0.68 -17.86  -6.89    2.03
## 23        381    NA    NA     NA      NA   NA     NA     NA      NA
## 24        376  4.90  7.15   5.53   12.56 1.12 -21.02 -11.21    2.03
## 25        906 11.53 13.62   1.10    7.89 0.73 -21.34 -11.02    2.03
## 26        341 13.36  8.50   9.34    8.93 0.84 -15.28 -12.11    2.03
## 27        590 11.39 13.74   0.73   10.19 0.93 -18.97 -10.41    2.03
## 28        382 12.37 16.70   1.84    6.14 0.46 -21.10 -11.60    2.03
## 29        346 10.63 12.23   1.86    8.37 0.76 -22.18  -9.16    2.03
## 30        435 12.80 15.17   1.26    5.94 0.58 -20.81 -11.41    2.03
## 31        343 12.22 17.04   0.44    6.48 0.62 -20.48 -10.76    2.03
## 32        274  6.92  7.82   1.69   12.48 1.14 -18.10  -7.54    2.03
## 33        364  7.09 12.26   2.35   13.23 1.25 -25.91  -9.08    2.03
## 34        382  8.70 15.85   1.73    6.84 0.70 -28.16 -10.81    2.03
## 35        496 10.49 13.82   1.05    6.17 0.54 -21.57  -9.57    2.03
## 36        289  8.68 12.31   1.46    9.48 0.88 -19.44  -9.95    2.03
## 37        147 11.76 15.80   0.56    6.20 0.55 -23.56 -10.44    2.03
## 38        353  8.15 13.58   1.14    9.80 0.86 -22.64  -9.95    2.03
## 39        205  9.39  6.81   0.98   10.43 0.96 -14.51  -6.15    2.03
## 40        407  7.84 11.42   2.16    9.78 1.08 -22.61  -8.92    2.03
## 41        324  9.82 11.33   1.72   12.41 1.09 -20.43 -11.45    2.03
## 42        722 11.89 15.34   0.46    7.85 0.65 -24.24 -10.10    2.03
## 43        397 10.89 17.73   0.82    6.15 0.65 -23.62  -9.26    2.03
## 44        725 14.47 20.21   0.37    3.65 0.30 -22.96 -11.72    2.03
## 45        725 14.64 20.71   0.37    3.73 0.32 -23.00 -11.67    2.03
## 46        628 13.37 14.64   0.63    6.50 0.53 -18.32 -11.13    2.03
## 47        377 11.63 15.32   0.80    8.17 0.68 -19.86 -10.66    2.03
## 48        371  8.44 14.65   0.62    7.62 0.64 -23.33  -9.75    2.03
## 49        742 15.41 19.00   0.40    5.67 0.53 -17.80 -12.51    2.03
## 50        399 14.21 20.79   0.36    4.64 0.49 -22.40 -11.31    2.03
## 51        345  9.09 10.48   1.29    9.17 0.82 -19.10 -10.10    2.03
## 52        371  8.55  9.68   1.96   14.47 1.27 -21.37  -9.29    2.03
```

```
## 53        685  4.61  5.29   8.92   17.93 1.54 -25.90  -9.06    2.03
## 54        952 11.34 20.85   0.55    4.73 0.43 -27.35 -11.09    2.03
## 55        416  7.09 12.26   2.35   13.23 1.25 -25.91  -9.08    2.03
## 56        628  8.15 13.58   1.14    9.80 0.86 -22.64  -9.95    2.03
## 57        402  3.53  4.70   5.72   17.84 1.47 -14.64  -8.65    2.03
## 58        835  8.58 10.45   1.75    8.30 0.86 -19.23  -9.46    2.03
## 59        205 10.96 10.00   0.51    6.87 0.68 -17.86  -6.89    2.03
## 60        966 11.65 18.49   0.64    6.05 0.55 -23.53 -11.80    2.03
## 61        966 11.65 18.49   0.64    6.05 0.55 -23.53 -11.80    2.03
## 62       1000  9.73 18.07   1.22    5.10 0.51 -27.15 -10.18    2.03
## 63        570 10.14  6.24   1.36   13.16 1.11 -13.57  -9.87    2.03
## 64        916  2.75  4.58     NA   20.61 1.74 -10.42 -10.15    2.03
## 65        232  7.71 11.76   2.70    7.67 0.68 -25.92 -10.15    2.03
## 66        409  8.44 14.65   0.62    7.62 0.64 -23.33  -9.75    2.03
## 67        738 10.89 17.73   0.82    6.15 0.65 -23.62  -9.26    2.03
## 68        936  6.92  7.82   1.69   12.48 1.14 -18.10  -7.54    2.03
## 69        394 17.09 17.57   0.84    5.38 0.48 -17.83 -12.84    2.03
## 70        258  7.13  4.20   1.21   12.95 1.19 -12.63  -9.92    2.03
## 71        399  9.81 12.74   2.35   10.02 0.91 -23.62 -11.35    2.03
## 72        692  8.27 12.42   1.05    9.93 0.77 -21.09 -10.61    2.03
## 73        927 14.39 19.61   0.58    4.95 0.40 -21.27 -11.54    2.03
## 74        677 13.47 15.28   0.39    6.23 0.56 -20.79 -11.42    2.03
## 75        780 11.39 13.74   0.73   10.19 0.93 -18.97 -10.41    2.03
## 76        706  7.02  7.80   1.98   13.20 1.18 -16.76  -7.69    2.03
## 77        653  7.81 12.09   2.06   10.63 0.93 -23.14  -8.36    2.03
## 78        706 11.76 15.80   0.56    6.20 0.55 -23.56 -10.44    2.03
## 79        853  8.68 12.31   1.46    9.48 0.88 -19.44  -9.95    2.03
## 80        773 11.53 13.62   1.10    7.89 0.73 -21.34 -11.02    2.03
## 81        908 10.49 13.82   1.05    6.17 0.54 -21.57  -9.57    2.03
## 82        412 12.37 16.70   1.84    6.14 0.46 -21.10 -11.60    2.03
## 83        861 13.66 16.61   0.74    7.26 0.69 -20.72 -11.59    2.03
## 84        277  6.77 12.02   0.86   11.24 0.96 -23.67  -9.98    2.03
## 85        642  2.95  4.75   0.44   18.66 1.70 -10.46  -8.10    2.03
## 86        588 15.21 18.40   0.88    5.67 0.47 -20.39 -11.93    2.03
## 87        905  9.69 13.93   3.04    5.96 0.51 -24.23 -11.66    2.03
## 88        905  9.77 14.09   3.04    5.75 0.51 -24.06 -11.24    2.03
## 89        797 13.76 16.38   0.38    5.18 0.46 -19.18  -9.63    2.03
## 90        651 13.75 14.50   2.68    7.48 0.67 -21.64 -11.34    2.03
## 91        539  8.67 10.80   1.13    4.70 0.48 -19.10 -10.01    2.03
## 92        746 12.14 13.39   0.83    9.94 0.83 -20.05 -10.35    2.03
## 93        912 12.57 19.70   0.93    5.23 0.53 -26.12 -11.68    2.03
## 94        718 14.27 19.85   0.42    4.47 0.32 -21.44 -11.31    2.03
## 95        709 14.47 20.21   0.37    3.65 0.30 -22.96 -11.72    2.03
## 96        709 14.64 20.71   0.37    3.73 0.32 -23.00 -11.67    2.03
## 97        753 10.91 18.66   0.78    5.44 0.48 -26.53 -11.17    2.03
## 98        226 13.36  8.50   9.34    8.93 0.84 -15.28 -12.11    2.03
## 99        561 11.89 15.34   0.46    7.85 0.65 -24.24 -10.10    2.03
## 100       711  8.70 15.85   1.73    6.84 0.70 -28.16 -10.81    2.03
## 101       691  7.84 11.42   2.16    9.78 1.08 -22.61  -8.92    2.03
## 102       670 10.64 12.66   2.40    8.93 0.81 -21.52  -9.37    2.03
## 103       148  7.36 11.83   1.41    9.14 0.82 -17.20 -10.54    2.03
## 104       377 11.60 17.45   0.93    5.81 0.56 -26.57 -10.32    2.03
## 105       381 12.80 15.17   1.26    5.94 0.58 -20.81 -11.41    2.03
## 106       377 11.63 15.32   0.80    8.17 0.68 -19.86 -10.66    2.03
```

```
## 107       633 13.35 15.94   0.79    5.35 0.49 -17.24 -11.88    2.03
## 108       461  8.48  9.31   0.94    9.80 0.89 -17.35 -10.47    2.03
## 109       413 10.82 15.73   0.44    5.57 0.50 -21.40  -9.66    2.03
## 110       689 13.01 16.18   1.13    6.94 0.62 -23.59 -10.79    2.03
## 111       370 10.63 12.23   1.86    8.37 0.76 -22.18  -9.16    2.03
## 112       402  3.38  1.93   3.65   15.24 1.26  -8.31  -9.04    2.03
## 113       797 13.01 18.92   0.33    4.23 0.44 -22.42 -10.85    2.03
## 114       340  4.90  7.15   5.53   12.56 1.12 -21.02 -11.21    2.03
## 115       661  9.82 11.33   1.72   12.41 1.09 -20.43 -11.45    2.03
## 116       540  8.59 13.40   0.80    8.73 0.80 -24.35  -9.33    2.03
## 117       459 10.65 13.98   0.89    6.59 0.61 -19.32  -9.93    2.03
## 118       707    NA    NA     NA      NA   NA     NA     NA      NA
## 119       619 10.70 15.94   0.93    7.31 0.62 -23.53 -11.14    2.03
## 120       465  7.17 11.74   2.10   10.44 0.87 -22.14  -9.08    2.03
## 121       402 13.88 17.34   0.68    5.95 0.47 -19.37 -12.44    2.03
## 122       302 12.22 17.04   0.44    6.48 0.62 -20.48 -10.76    2.03
## 123       658 13.29 18.86   0.26    4.03 0.39 -24.82 -11.00    2.03
## 124       378  6.86  7.07   2.89   12.02 1.14 -17.61 -10.85    2.03
## 125       706  9.39  6.81   0.98   10.43 0.96 -14.51  -6.15    2.03
## 126       290 13.83 16.65   0.74    6.40 0.48 -21.75 -11.60    2.03
## 127       726    NA    NA     NA      NA   NA     NA     NA      NA
```

```
SLPBreathy <- SLP %>%
  dplyr::select(Breathiness, dummy)

#merging ("left-join" only appends AVQI entries that has SLP rating correspondents)
library(dplyr)
av_AVQI_B <-left_join(SLPBreathy,AVQI, by = "dummy")%>%
  dplyr::select(-Speaker, -T, -AVQI, -dummy)

#converting jitter into a factor
av_AVQI_B$Jitter <- as.numeric(levels(av_AVQI_B$Jitter))[av_AVQI_B$Jitter]
```

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

```
av_AVQI_B
```

```
##     Breathiness  CPPS   HNR Jitter Shimmer ShdB  Slope   Tilt Version
## 1             4  2.75  4.58     NA   20.61 1.74 -10.42 -10.15    2.03
## 2           423  8.55  9.68   1.96   14.47 1.27 -21.37  -9.29    2.03
## 3           908 13.47 15.28   0.39    6.23 0.56 -20.79 -11.42    2.03
## 4           614  8.67 10.80   1.13    4.70 0.48 -19.10 -10.01    2.03
## 5           418    NA    NA     NA      NA   NA     NA     NA      NA
## 6           885 12.36 20.41   0.43    4.17 0.38 -25.90  -9.94    2.03
## 7           259  7.13  4.20   1.21   12.95 1.19 -12.63  -9.92    2.03
## 8           616 13.72 19.08   0.31    3.84 0.37 -21.13 -11.76    2.03
## 9           687 11.60 17.45   0.93    5.81 0.56 -26.57 -10.32    2.03
## 10          218 10.82 15.73   0.44    5.57 0.50 -21.40  -9.66    2.03
## 11          361  9.81 12.74   2.35   10.02 0.91 -23.62 -11.35    2.03
## 12          304  8.59 13.40   0.80    8.73 0.80 -24.35  -9.33    2.03
## 13          410 10.91 18.66   0.78    5.44 0.48 -26.53 -11.17    2.03
## 14          568 10.64 12.66   2.40    8.93 0.81 -21.52  -9.37    2.03
## 15          687  6.77 12.02   0.86   11.24 0.96 -23.67  -9.98    2.03
## 16          945 13.83 16.65   0.74    6.40 0.48 -21.75 -11.60    2.03
## 17          691 11.34 20.85   0.55    4.73 0.43 -27.35 -11.09    2.03
## 18          323  8.27 12.42   1.05    9.93 0.77 -21.09 -10.61    2.03
## 19          605 14.21 20.79   0.36    4.64 0.49 -22.40 -11.31    2.03
## 20          424  8.48  9.31   0.94    9.80 0.89 -17.35 -10.47    2.03
## 21          264  7.81 12.09   2.06   10.63 0.93 -23.14  -8.36    2.03
## 22          722 10.96 10.00   0.51    6.87 0.68 -17.86  -6.89    2.03
## 23          547    NA    NA     NA      NA   NA     NA     NA      NA
## 24          363  4.90  7.15   5.53   12.56 1.12 -21.02 -11.21    2.03
## 25          445 11.53 13.62   1.10    7.89 0.73 -21.34 -11.02    2.03
## 26          808 13.36  8.50   9.34    8.93 0.84 -15.28 -12.11    2.03
## 27          938 11.39 13.74   0.73   10.19 0.93 -18.97 -10.41    2.03
## 28          887 12.37 16.70   1.84    6.14 0.46 -21.10 -11.60    2.03
## 29          781 10.63 12.23   1.86    8.37 0.76 -22.18  -9.16    2.03
## 30          439 12.80 15.17   1.26    5.94 0.58 -20.81 -11.41    2.03
## 31          431 12.22 17.04   0.44    6.48 0.62 -20.48 -10.76    2.03
## 32           16  6.92  7.82   1.69   12.48 1.14 -18.10  -7.54    2.03
## 33          287  7.09 12.26   2.35   13.23 1.25 -25.91  -9.08    2.03
## 34          214  8.70 15.85   1.73    6.84 0.70 -28.16 -10.81    2.03
## 35          469 10.49 13.82   1.05    6.17 0.54 -21.57  -9.57    2.03
## 36          225  8.68 12.31   1.46    9.48 0.88 -19.44  -9.95    2.03
## 37          333 11.76 15.80   0.56    6.20 0.55 -23.56 -10.44    2.03
## 38          489  8.15 13.58   1.14    9.80 0.86 -22.64  -9.95    2.03
## 39          594  9.39  6.81   0.98   10.43 0.96 -14.51  -6.15    2.03
## 40          341  7.84 11.42   2.16    9.78 1.08 -22.61  -8.92    2.03
## 41          489  9.82 11.33   1.72   12.41 1.09 -20.43 -11.45    2.03
## 42          908 11.89 15.34   0.46    7.85 0.65 -24.24 -10.10    2.03
## 43          180 10.89 17.73   0.82    6.15 0.65 -23.62  -9.26    2.03
## 44          974 14.47 20.21   0.37    3.65 0.30 -22.96 -11.72    2.03
## 45          974 14.64 20.71   0.37    3.73 0.32 -23.00 -11.67    2.03
## 46          980 13.37 14.64   0.63    6.50 0.53 -18.32 -11.13    2.03
## 47          915 11.63 15.32   0.80    8.17 0.68 -19.86 -10.66    2.03
## 48          178  8.44 14.65   0.62    7.62 0.64 -23.33  -9.75    2.03
## 49          931 15.41 19.00   0.40    5.67 0.53 -17.80 -12.51    2.03
## 50          309 14.21 20.79   0.36    4.64 0.49 -22.40 -11.31    2.03
## 51          166  9.09 10.48   1.29    9.17 0.82 -19.10 -10.10    2.03
## 52          233  8.55  9.68   1.96   14.47 1.27 -21.37  -9.29    2.03
```

```
## 53          372  4.61  5.29   8.92   17.93 1.54 -25.90  -9.06    2.03
## 54          716 11.34 20.85   0.55    4.73 0.43 -27.35 -11.09    2.03
## 55           97  7.09 12.26   2.35   13.23 1.25 -25.91  -9.08    2.03
## 56           85  8.15 13.58   1.14    9.80 0.86 -22.64  -9.95    2.03
## 57           74  3.53  4.70   5.72   17.84 1.47 -14.64  -8.65    2.03
## 58           16  8.58 10.45   1.75    8.30 0.86 -19.23  -9.46    2.03
## 59          794 10.96 10.00   0.51    6.87 0.68 -17.86  -6.89    2.03
## 60          656 11.65 18.49   0.64    6.05 0.55 -23.53 -11.80    2.03
## 61          656 11.65 18.49   0.64    6.05 0.55 -23.53 -11.80    2.03
## 62          197  9.73 18.07   1.22    5.10 0.51 -27.15 -10.18    2.03
## 63         1000 10.14  6.24   1.36   13.16 1.11 -13.57  -9.87    2.03
## 64            1  2.75  4.58     NA   20.61 1.74 -10.42 -10.15    2.03
## 65          456  7.71 11.76   2.70    7.67 0.68 -25.92 -10.15    2.03
## 66          102  8.44 14.65   0.62    7.62 0.64 -23.33  -9.75    2.03
## 67          393 10.89 17.73   0.82    6.15 0.65 -23.62  -9.26    2.03
## 68           52  6.92  7.82   1.69   12.48 1.14 -18.10  -7.54    2.03
## 69         1000 17.09 17.57   0.84    5.38 0.48 -17.83 -12.84    2.03
## 70          808  7.13  4.20   1.21   12.95 1.19 -12.63  -9.92    2.03
## 71          614  9.81 12.74   2.35   10.02 0.91 -23.62 -11.35    2.03
## 72          238  8.27 12.42   1.05    9.93 0.77 -21.09 -10.61    2.03
## 73          643 14.39 19.61   0.58    4.95 0.40 -21.27 -11.54    2.03
## 74          853 13.47 15.28   0.39    6.23 0.56 -20.79 -11.42    2.03
## 75          358 11.39 13.74   0.73   10.19 0.93 -18.97 -10.41    2.03
## 76          241  7.02  7.80   1.98   13.20 1.18 -16.76  -7.69    2.03
## 77          872  7.81 12.09   2.06   10.63 0.93 -23.14  -8.36    2.03
## 78          356 11.76 15.80   0.56    6.20 0.55 -23.56 -10.44    2.03
## 79          423  8.68 12.31   1.46    9.48 0.88 -19.44  -9.95    2.03
## 80          491 11.53 13.62   1.10    7.89 0.73 -21.34 -11.02    2.03
## 81          543 10.49 13.82   1.05    6.17 0.54 -21.57  -9.57    2.03
## 82          970 12.37 16.70   1.84    6.14 0.46 -21.10 -11.60    2.03
## 83          610 13.66 16.61   0.74    7.26 0.69 -20.72 -11.59    2.03
## 84          638  6.77 12.02   0.86   11.24 0.96 -23.67  -9.98    2.03
## 85           13  2.95  4.75   0.44   18.66 1.70 -10.46  -8.10    2.03
## 86          999 15.21 18.40   0.88    5.67 0.47 -20.39 -11.93    2.03
## 87          618  9.69 13.93   3.04    5.96 0.51 -24.23 -11.66    2.03
## 88          618  9.77 14.09   3.04    5.75 0.51 -24.06 -11.24    2.03
## 89          613 13.76 16.38   0.38    5.18 0.46 -19.18  -9.63    2.03
## 90          892 13.75 14.50   2.68    7.48 0.67 -21.64 -11.34    2.03
## 91          935  8.67 10.80   1.13    4.70 0.48 -19.10 -10.01    2.03
## 92          812 12.14 13.39   0.83    9.94 0.83 -20.05 -10.35    2.03
## 93          742 12.57 19.70   0.93    5.23 0.53 -26.12 -11.68    2.03
## 94          862 14.27 19.85   0.42    4.47 0.32 -21.44 -11.31    2.03
## 95          870 14.47 20.21   0.37    3.65 0.30 -22.96 -11.72    2.03
## 96          870 14.64 20.71   0.37    3.73 0.32 -23.00 -11.67    2.03
## 97          213 10.91 18.66   0.78    5.44 0.48 -26.53 -11.17    2.03
## 98          809 13.36  8.50   9.34    8.93 0.84 -15.28 -12.11    2.03
## 99          835 11.89 15.34   0.46    7.85 0.65 -24.24 -10.10    2.03
## 100         418  8.70 15.85   1.73    6.84 0.70 -28.16 -10.81    2.03
## 101         835  7.84 11.42   2.16    9.78 1.08 -22.61  -8.92    2.03
## 102         759 10.64 12.66   2.40    8.93 0.81 -21.52  -9.37    2.03
## 103         373  7.36 11.83   1.41    9.14 0.82 -17.20 -10.54    2.03
## 104         553 11.60 17.45   0.93    5.81 0.56 -26.57 -10.32    2.03
## 105         542 12.80 15.17   1.26    5.94 0.58 -20.81 -11.41    2.03
## 106         757 11.63 15.32   0.80    8.17 0.68 -19.86 -10.66    2.03
```

```
## 107         256 13.35 15.94   0.79    5.35 0.49 -17.24 -11.88    2.03
## 108         357  8.48  9.31   0.94    9.80 0.89 -17.35 -10.47    2.03
## 109         240 10.82 15.73   0.44    5.57 0.50 -21.40  -9.66    2.03
## 110         383 13.01 16.18   1.13    6.94 0.62 -23.59 -10.79    2.03
## 111         590 10.63 12.23   1.86    8.37 0.76 -22.18  -9.16    2.03
## 112         176  3.38  1.93   3.65   15.24 1.26  -8.31  -9.04    2.03
## 113         562 13.01 18.92   0.33    4.23 0.44 -22.42 -10.85    2.03
## 114         433  4.90  7.15   5.53   12.56 1.12 -21.02 -11.21    2.03
## 115         556  9.82 11.33   1.72   12.41 1.09 -20.43 -11.45    2.03
## 116         391  8.59 13.40   0.80    8.73 0.80 -24.35  -9.33    2.03
## 117         571 10.65 13.98   0.89    6.59 0.61 -19.32  -9.93    2.03
## 118         241    NA    NA     NA      NA   NA     NA     NA      NA
## 119         382 10.70 15.94   0.93    7.31 0.62 -23.53 -11.14    2.03
## 120         253  7.17 11.74   2.10   10.44 0.87 -22.14  -9.08    2.03
## 121         847 13.88 17.34   0.68    5.95 0.47 -19.37 -12.44    2.03
## 122         647 12.22 17.04   0.44    6.48 0.62 -20.48 -10.76    2.03
## 123         765 13.29 18.86   0.26    4.03 0.39 -24.82 -11.00    2.03
## 124         156  6.86  7.07   2.89   12.02 1.14 -17.61 -10.85    2.03
## 125         829  9.39  6.81   0.98   10.43 0.96 -14.51  -6.15    2.03
## 126         713 13.83 16.65   0.74    6.40 0.48 -21.75 -11.60    2.03
## 127         847    NA    NA     NA      NA   NA     NA     NA      NA
```

##Re-training the roughness tree

Partitioning:

```
#setting seed for reproducibility
set.seed(1606)

#deviding the data into three parts

#setting proportions, 70% for training, devide the remainder into two halves
assignment <- sample(1:3, size = nrow(av_AVQI_R), prob = c(0.7, 0.15, 0.15), replace = T
RUE)

#subsetting the data to training indices
roughness_train <- av_AVQI_R[assignment == 1, ]

#subsetting the data to validation indices
roughness_valid <- av_AVQI_R[assignment == 2, ]

#subsetting the data to test indices
roughness_test <- av_AVQI_R[assignment == 3, ]
```

Training the model:

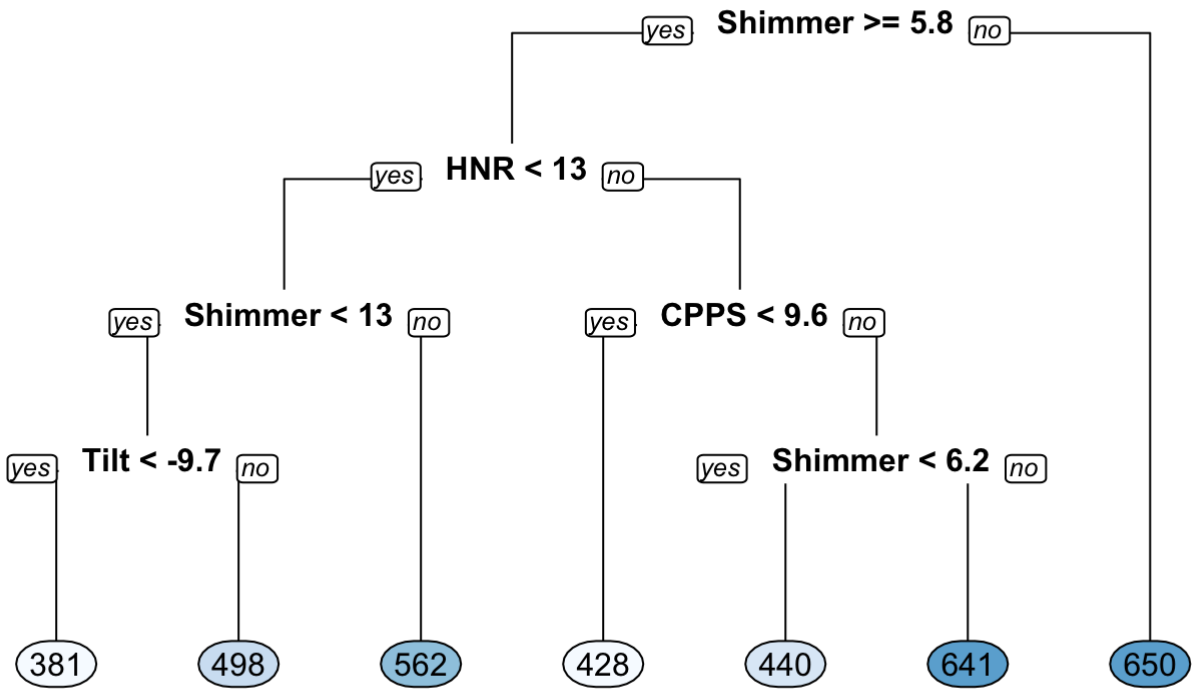
```
#the model
library(rpart)
roughness_model <- rpart(formula = Roughness ~.,
                         data = roughness_train,
                         method = "anova")

roughness_model
```

```
## n=81 (4 observations deleted due to missingness)
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 81 3172902.00 524.7284
##    2) Shimmer>=5.78 67 2355928.00 498.5672
##      4) HNR< 13.065 36 1349211.00 461.9722
##        8) Shimmer< 13.055 27  934259.20 428.7407
##         16) Tilt< -9.69 16  247554.90 381.0625 *
##         17) Tilt>=-9.69 11  597428.90 498.0909 *
##        9) Shimmer>=13.055 9  295684.00 561.6667 *
##      5) HNR>=13.065 31  902519.90 541.0645
##       10) CPPS< 9.595 7   77817.71 427.5714 *
##       11) CPPS>=9.595 24  708239.30 574.1667
##         22) Shimmer< 6.16 8  103955.50 440.2500 *
##         23) Shimmer>=6.16 16  389079.80 641.1250 *
##    3) Shimmer< 5.78 14  551666.90 649.9286 *
```

Plotting the tree:

```
library(rpart.plot)
rpart.plot(x = roughness_model, yesno = 2, type = 0, extra = 0)
```



Evaluating the model:

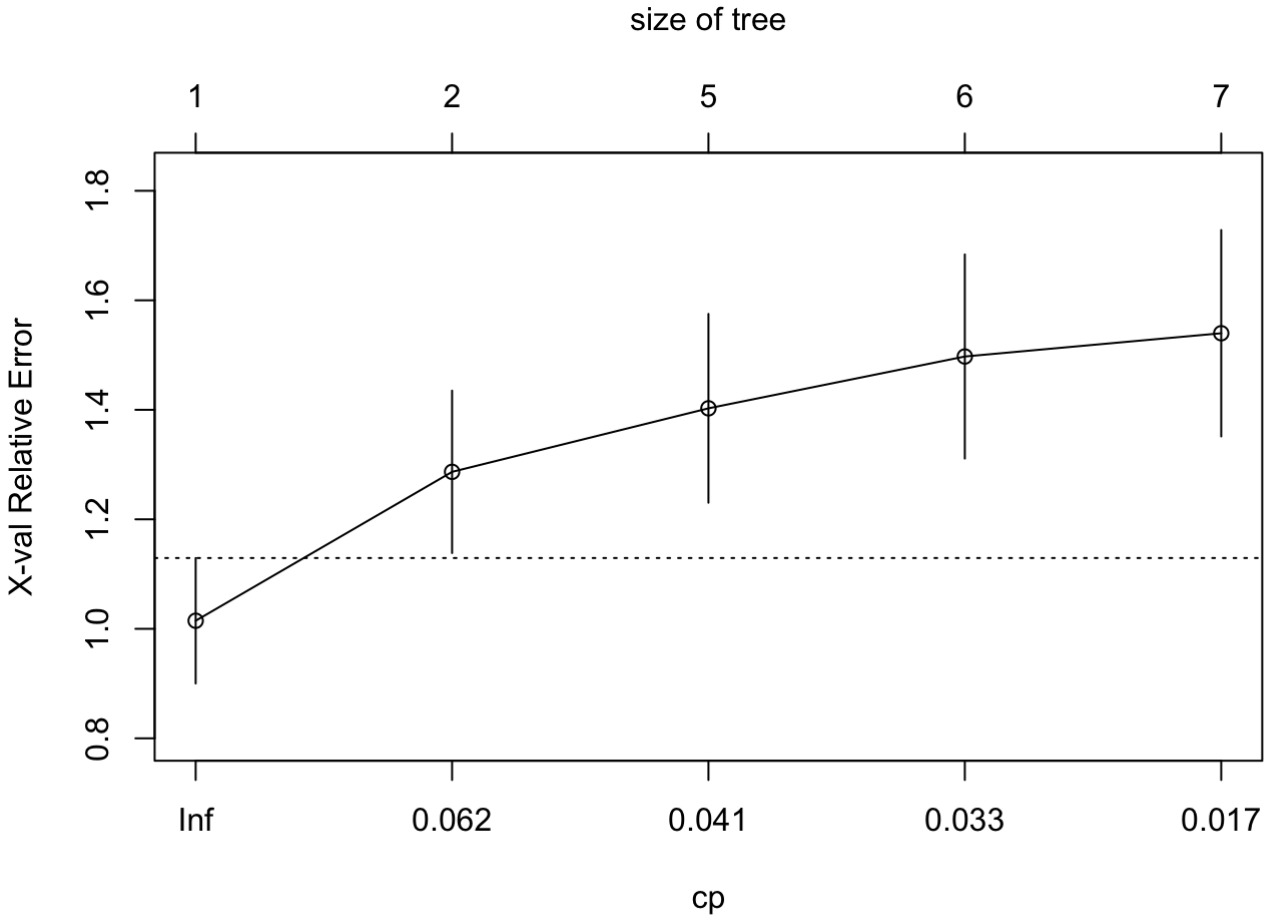
```
#generating preedictions on the test set
pred <- predict(object = roughness_model,
                newdata = roughness_test)

#computing the RMSE
library(Metrics)
rmse(actual = roughness_test$Roughness,
     predicted = pred)
```

```
## [1] 244.7803
```

Tuning the model:

```
#plot the Complexity Parameter table
plotcp(roughness_model)
```




```
#print the "CP Table"
print(roughness_model$cptable)
```

```
##           CP nsplit rel error   xerror      xstd
## 1 0.08361640      0 1.0000000 1.014744 0.1145782
## 2 0.04579031      1 0.9163836 1.286689 0.1483205
## 3 0.03758950      4 0.7790127 1.402721 0.1724188
## 4 0.02813681      5 0.7414232 1.497368 0.1863977
## 5 0.01000000      6 0.7132864 1.539853 0.1884773
```

```
#retreaving the optimal cp value based on cross-validation error
opt_index <- which.min(roughness_model$cptable[, "xerror"])
cp_opt <- roughness_model$cptable[opt_index, "CP"]

#prune the tree to optimazed cp value
roughness_model_opt <- prune(tree = roughness_model, cp = cp_opt)

#plot the optimized model
#rpart.plot(x = roughness_model_opt, yesno = 2, type = 0)
#^not plottable

#trying this:
rpart.plot(x = roughness_model_opt, type = 0)
```

525
100%

:/

Grid search:

Creating a grid of possible models:

```
#establishing a list of possible values for minsplit and maxdepth
splits <- seq(1, 5, 1)
depths <- seq(1, 5, 1)

#creating a data frame containing all combinations
hyper_grid <- expand.grid(minsplit = splits, maxdepth = depths)

#checking out the grid
head(hyper_grid)
```

```
##   minsplit maxdepth
## 1        1        1
## 2        2        1
## 3        3        1
## 4        4        1
## 5        5        1
## 6        1        2
```

```
#printing the number of grid combinations
nrow(hyper_grid)
```

```
## [1] 25
```

```
#number of potential models in the grid
num_models <- nrow(hyper_grid)

#creating an empty list to store models
roughness_models <- list()

#a for loop to loop over the rows of hyper_grid to train the grid of models

for (i in 1:num_models) {
  #getting minsplit, maxdepth values at row i
  minsplit <- hyper_grid$minsplit[i]
  maxdepth <- hyper_grid$maxdepth[i]

  #training a model and storing it in the list
  roughness_models[[i]] <- rpart(formula = Roughness ~ .,
                                 data = roughness_valid,
                                 method = "anova",
                                 minsplit = minsplit,
                                 maxdepth = maxdepth)
}
```

Evaluating the grid:

```
#number of potential models in the grid
num_models <- lengths(roughness_models)

#creating an empty vector to store RMSE values
rmse_values <- c()

#a loop over the models to compute validation RMSE
for (i in 1:num_models){

  #retrieving the i^th model from the list
  model <- roughness_models[[i]]

  #generating predictions on roughness_valid
  pred <- predict(object = model,
                  newdata = roughness_valid)

  #computing validation RMSE and add to the vector
  rmse_values[i] <- rmse(actual = roughness_valid$Roughness,
                         predicted = pred)
}
```

```
## Warning in 1:num_models: numerical expression has 25 elements: only the first
## used
```

```
#idenetifying the model with smallest validation set RMSE
best_model <- roughness_models[[which.min(rmse_values)]]

#printing the parameters of the best model
best_model$control
```

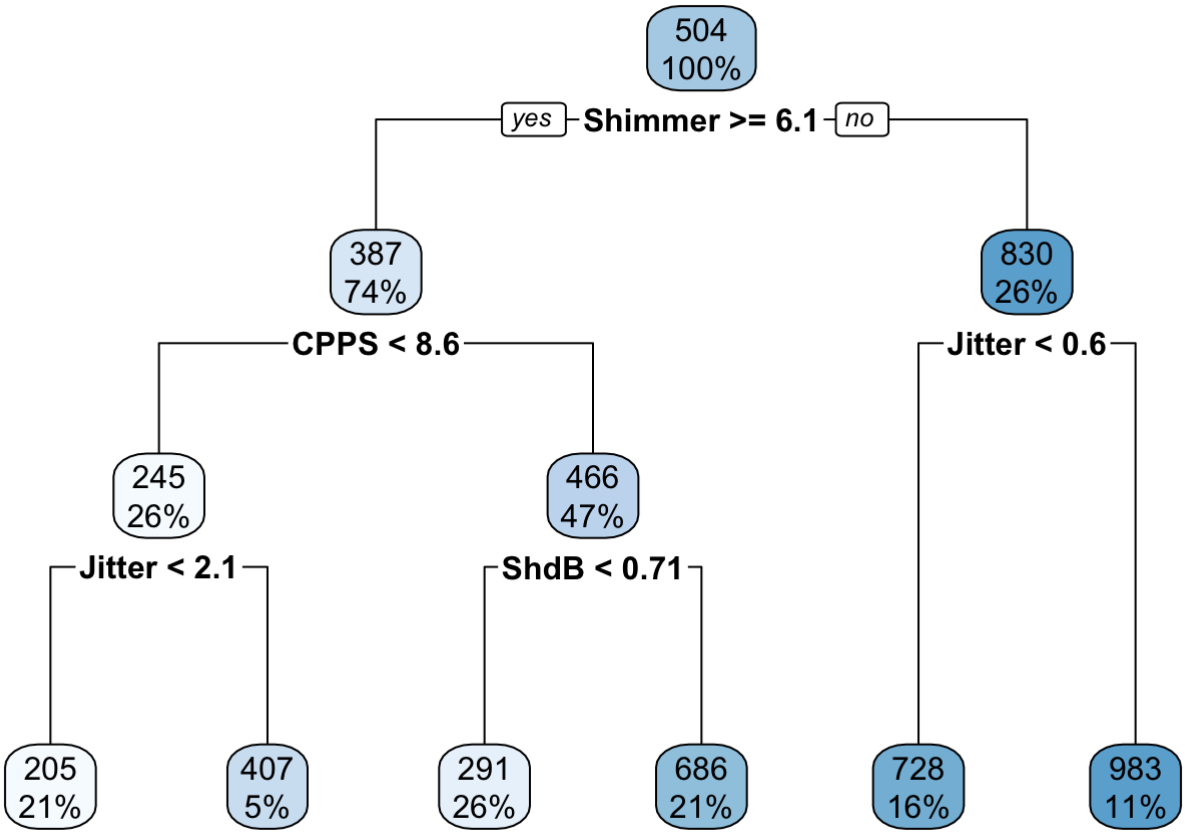
```
## $minsplit
## [1] 2
##
## $minbucket
## [1] 1
##
## $cp
## [1] 0.01
##
## $maxcompete
## [1] 4
##
## $maxsurrogate
## [1] 5
##
## $usesurrogate
## [1] 2
##
## $surrogatestyle
## [1] 0
##
## $maxdepth
## [1] 3
##
## $xval
## [1] 10
```

```
#computing test set RMSE on best_model
pred <- predict(object = best_model,
                newdata = roughness_test)
rmse(actual = roughness_test$Roughness,
     predicted = pred)
```

```
## [1] 284.1058
```

Plotting the new best model

```
rpart.plot(best_model)
```



Training a bagged tree model:

```
library(ipred)
#setting seed for reproducibility
set.seed(1557)

#training a bagged model
roughness_model_bag <- bagging(formula = Roughness ~ .,
                           data = roughness_train,
                           coob = TRUE) #=using out-of-bag samples to estimate model acc
uracy

#printing the model
roughness_model_bag
```

```
##
## Bagging regression trees with 25 bootstrap replications
##
## Call: bagging.data.frame(formula = Roughness ~ ., data = roughness_train,
##     coob = TRUE)
##
## Out-of-bag estimate of root mean squared error:  220.4213
```

Assessing the bagged trees:

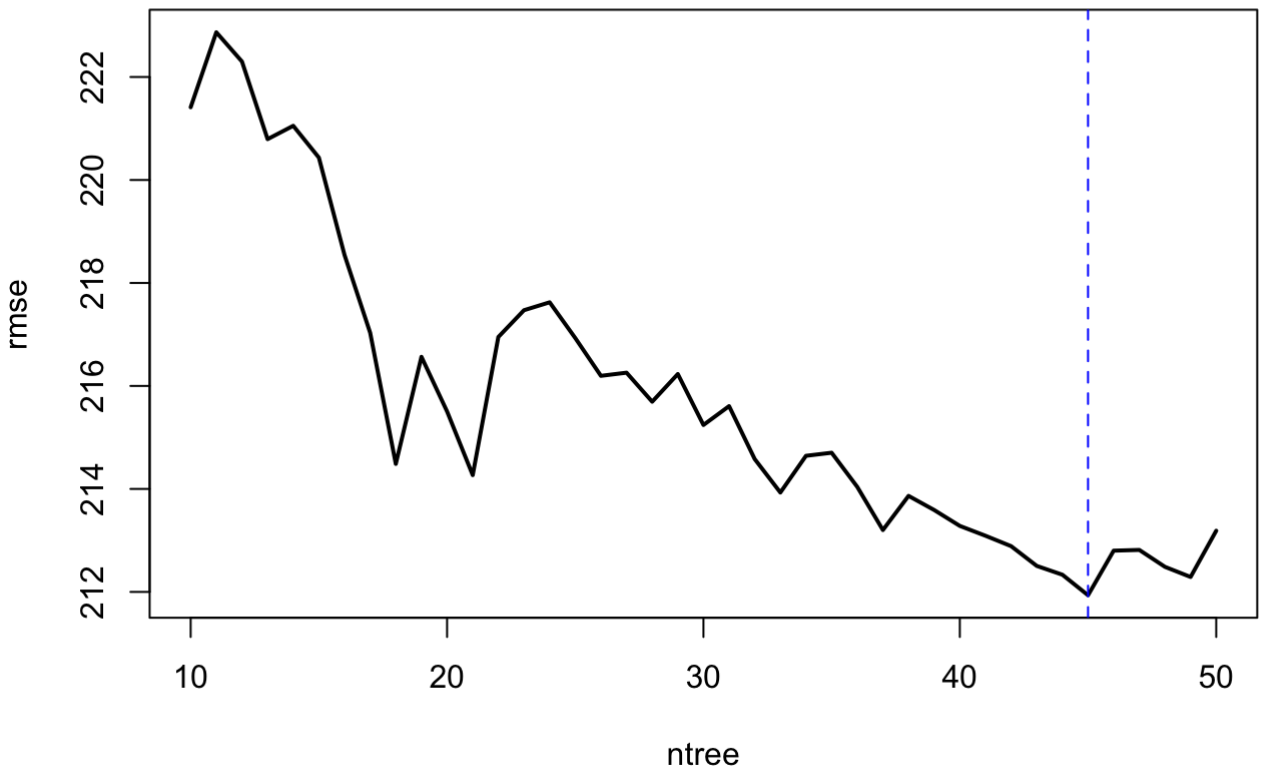
```
#assess 10-25 bagged trees
ntree <- 10:50

#create empty vector to store out of bag RMSE values
rmse <- vector(mode = "numeric", length = length(ntree))

for (i in seq_along(ntree)) {
  #set seed for reproducibility
  set.seed(1555)

  #perform bagged model
  model <- bagging(
    formula = Roughness ~ .,
    data = roughness_train,
    coob = TRUE,
    nbagg = ntree[i]
  )
  #get out of bag error
  rmse[i] <- model$err
}

plot(ntree, rmse, type = 'l', lwd = 2)
abline(v = 45, col = "blue", lty = "dashed")
```



^the lowest rmse appeared when the number of trees is 45.

Bagging with Caret:

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following objects are masked from 'package:Metrics':
##
##     precision, recall
```

```
#specify a 10-fold cross validation
ctrl <- trainControl(method = "cv", number = 10)

#cross-validation bagged model
bagged_cv <- train(
  Roughness ~ .,
  data = roughness_train,
  method = "treebag",
  trControl = ctrl,
  mtry = 45, #according to the above step
  importance = TRUE,
  na.action = na.pass #ignore N/A's
)
```

```
## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length
```

```
## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length
```

```
## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length
```

```
## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length
```

```
## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length

## Warning in cprob[tindx] + pred: longer object length is not a multiple of
## shorter object length
```

```
plot(varImp(bagged_cv), 8)
```