PLOT FUN

Ethan Allavarpu (UID: 405287603)

11/30/2020

```
training <- read.csv("training.csv", stringsAsFactors = FALSE)
dim(training)
```

```
## [1] 7242  260
```

```
any(is.na(training))
```

```
## [1] FALSE
```

```
var_types <- vapply(training, class, character(1))
names(training)[-which(var_types %in% c("integer", "numeric"))]
```

```
## [1] "PublishedDate"
```

```
dates <- training$PublishedDate
head(dates)
```

```
## [1] "4/17/2020 10:38" "8/31/2020 9:56"  "8/16/2020 12:15" "8/22/2020 9:00"
## [5] "8/22/2020 14:18" "7/24/2020 21:16"
```

```
split_dates <- strsplit(dates, "[:/ ]")
head(split_dates)
```

```
## [[1]]
## [1] "4"    "17"   "2020" "10"   "38"
##
## [[2]]
## [1] "8"    "31"   "2020" "9"    "56"
##
## [[3]]
## [1] "8"    "16"   "2020" "12"   "15"
##
## [[4]]
## [1] "8"    "22"   "2020" "9"    "00"
##
## [[5]]
## [1] "8"    "22"   "2020" "14"   "18"
##
## [[6]]
## [1] "7"    "24"   "2020" "21"   "16"
```

```
split_dates <- lapply(split_dates, as.numeric)
years <- function(date) {
  date[3]
}
yrs <- lapply(split_dates, years)
```

```r
unique(yrs)  # Only 2020
```

```
## [[1]]
## [1] 2020
```

```r
new_time <- function(old_date) {
  old_date <- as.numeric(old_date)
  month <- old_date[1]
  day <- old_date[2]
  hr <- old_date[4]
  minute <- old_date[5]
  month_days <- c(31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
  complete_months <- month - 1
  if (complete_months > 0) {
    complete_month_days <- sum(month_days[1:complete_months])
  } else {
    complete_month_days <- 0
  }
  total_days <- complete_month_days + day
  pre_hrs <- total_days * 24
  total_hrs <- pre_hrs + hr
  pre_min <- total_hrs * 60
  final_time <- pre_min + minute
  final_time
}
processed_dates <- vapply(split_dates, new_time, numeric(1))
training$PublishedDate <- processed_dates
```

```r
cor_mtx <- round(cor(training[, -ncol(training)]), 2)
```

```
## Warning in cor(training[, -ncol(training)]): the standard deviation is zero
```

```r
library(reshape2)
library(ggplot2)
melted_cor_mtx <- melt(cor_mtx)
cor_vars <- which(abs(cor_mtx) > 0.5, arr.ind = TRUE)
for (i in 1:nrow(cor_vars)) {
  if (cor_vars[i, 1] >= cor_vars[i, 2]) {
    cor_vars[i, ] <- rep(NA, 2)
  }
}

all_na <- function(data) {
  all(is.na(data))
}
unique_cor_vars <- apply(cor_vars, 1, all_na)
cor_vars <- cor_vars[!unique_cor_vars, ]
corr_vars <- unique(rownames(cor_vars))
```

```r
outlier <- function(data) {
  low <- mean(data) - 1.5 * IQR(data)
  high <- mean(data) + 1.5 * IQR(data)
  which(data < low | data > high)
}
outliers <- table(unlist(lapply(training[,-ncol(training)], outlier)))
```

```
outlier_index <- sort(outliers, decreasing = TRUE)
training <- training[-as.numeric(names(outlier_index)[1:50]), ]
```

```
subs <- training[, 248:250]
sub_final <- 3 - 3 * subs[[1]] - 2 * subs[[2]] - subs[[3]]
views <- training[, 251:253]
views_final <- 3 - 3 * views[[1]] - 2 * views[[2]] - views[[3]]
growth <- training[, 254:256]
growth_final <- 3 - 3 * growth[[1]] - 2 * growth[[2]] - growth[[3]]
vids <- training[, 257:259]
vid_final <- 3 - 3 * vids[[1]] - 2 * vids[[2]] - vids[[3]]
growth_2_6 <- training$growth_2_6
names(training)[c(1, 248:260)]
```

```
##  [1] "id"                          "Num_Subscribers_Base_low"
##  [3] "Num_Subscribers_Base_low_mid"  "Num_Subscribers_Base_mid_high"
##  [5] "Num_Views_Base_low"          "Num_Views_Base_low_mid"
##  [7] "Num_Views_Base_mid_high"     "avg_growth_low"
##  [9] "avg_growth_low_mid"          "avg_growth_mid_high"
## [11] "count_vids_low"              "count_vids_low_mid"
## [13] "count_vids_mid_high"         "growth_2_6"
```

```
training <- training[, -c(1, 248:260)]
training <- data.frame(training, Num_Subscribers_Base = sub_final,
                       Num_Views_Base = views_final, avg_growth = growth_final,
                       count_vids = vid_final, growth_2_6 = growth_2_6)
```

```
factor_vars <- training[, 247:250]
factor_vars <- data.frame(lapply(factor_vars, factor))
training[, 247:250] <- factor_vars
```

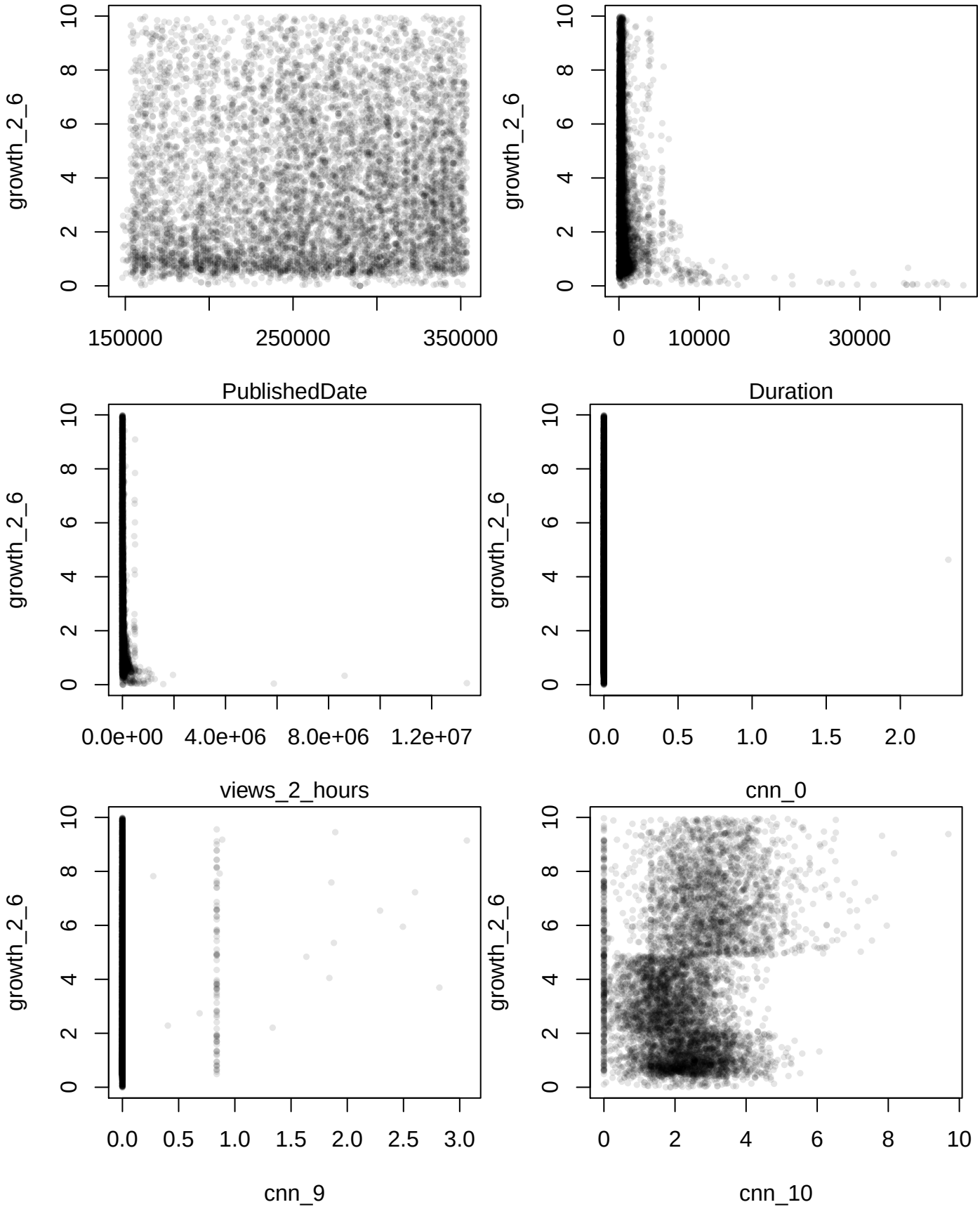
```
library(dplyr)
```

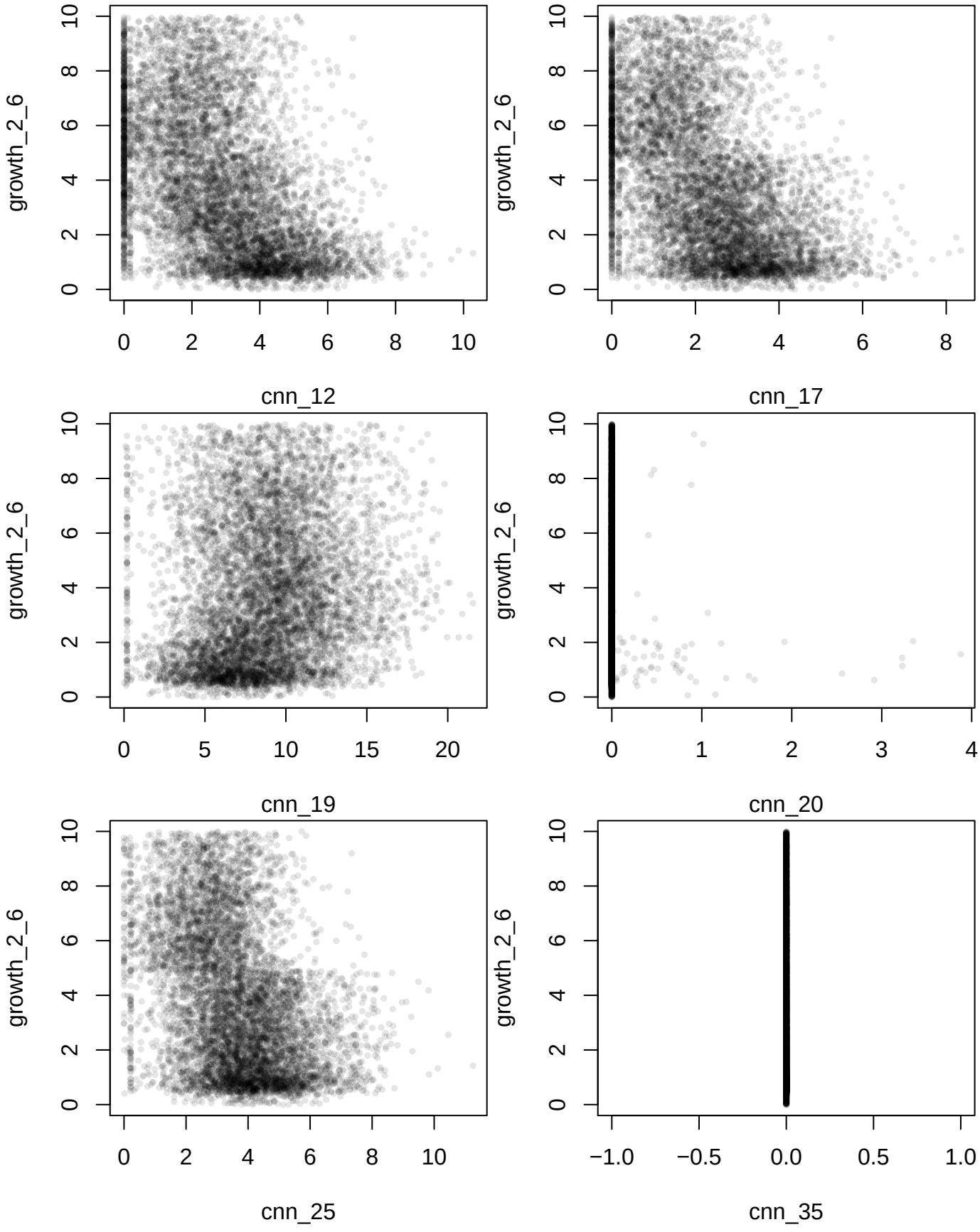
```
##
## Attaching package: 'dplyr'
```

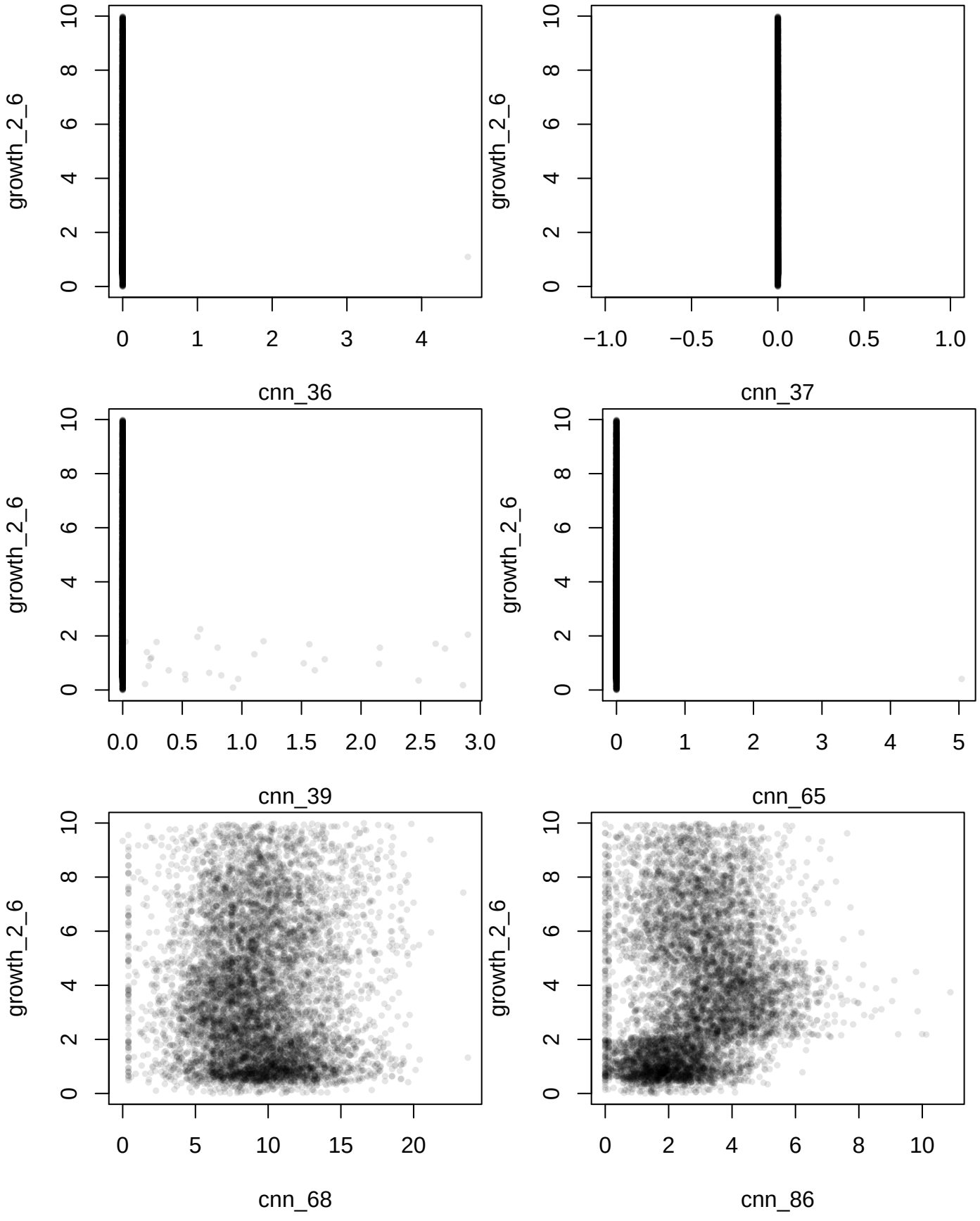
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

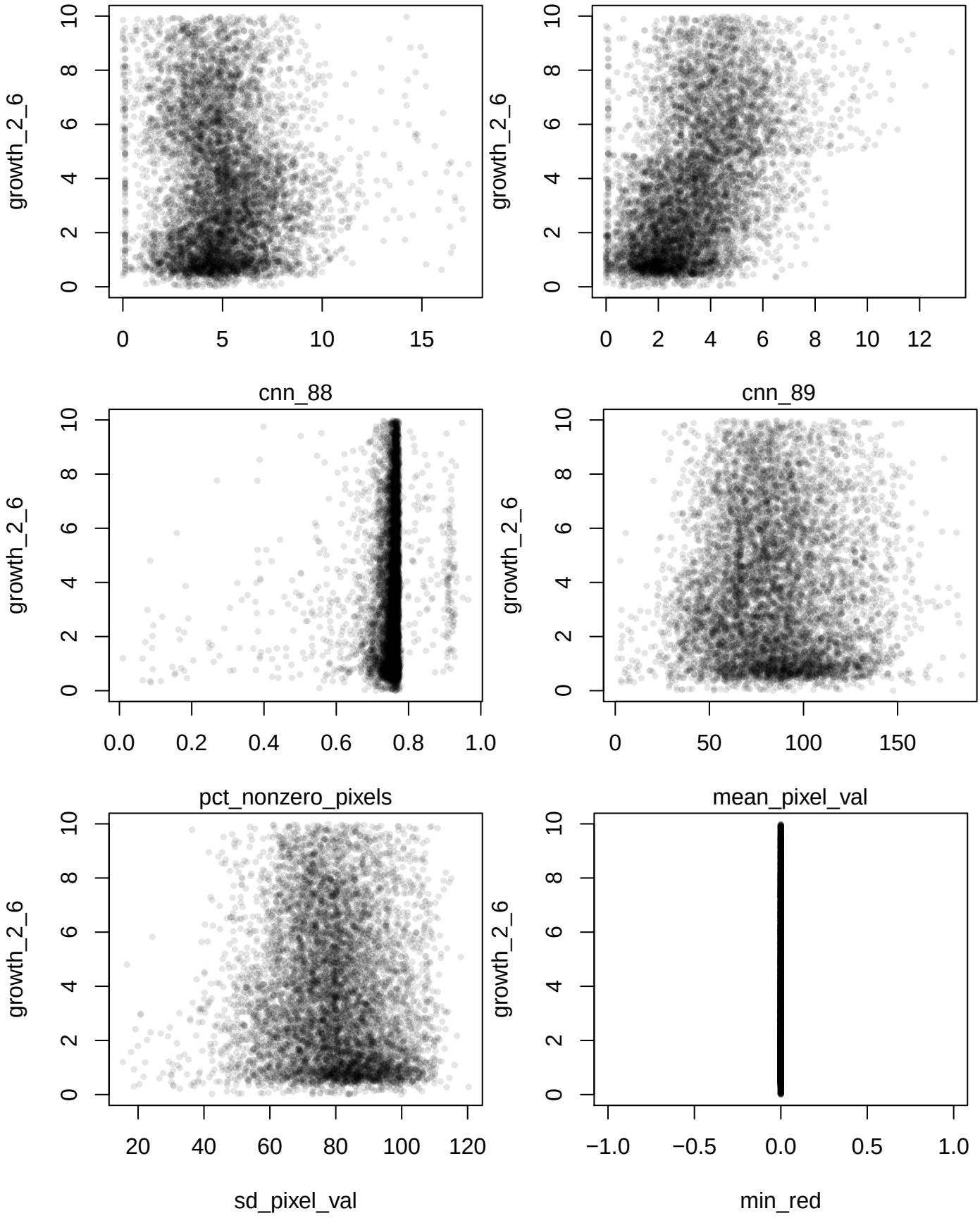
```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

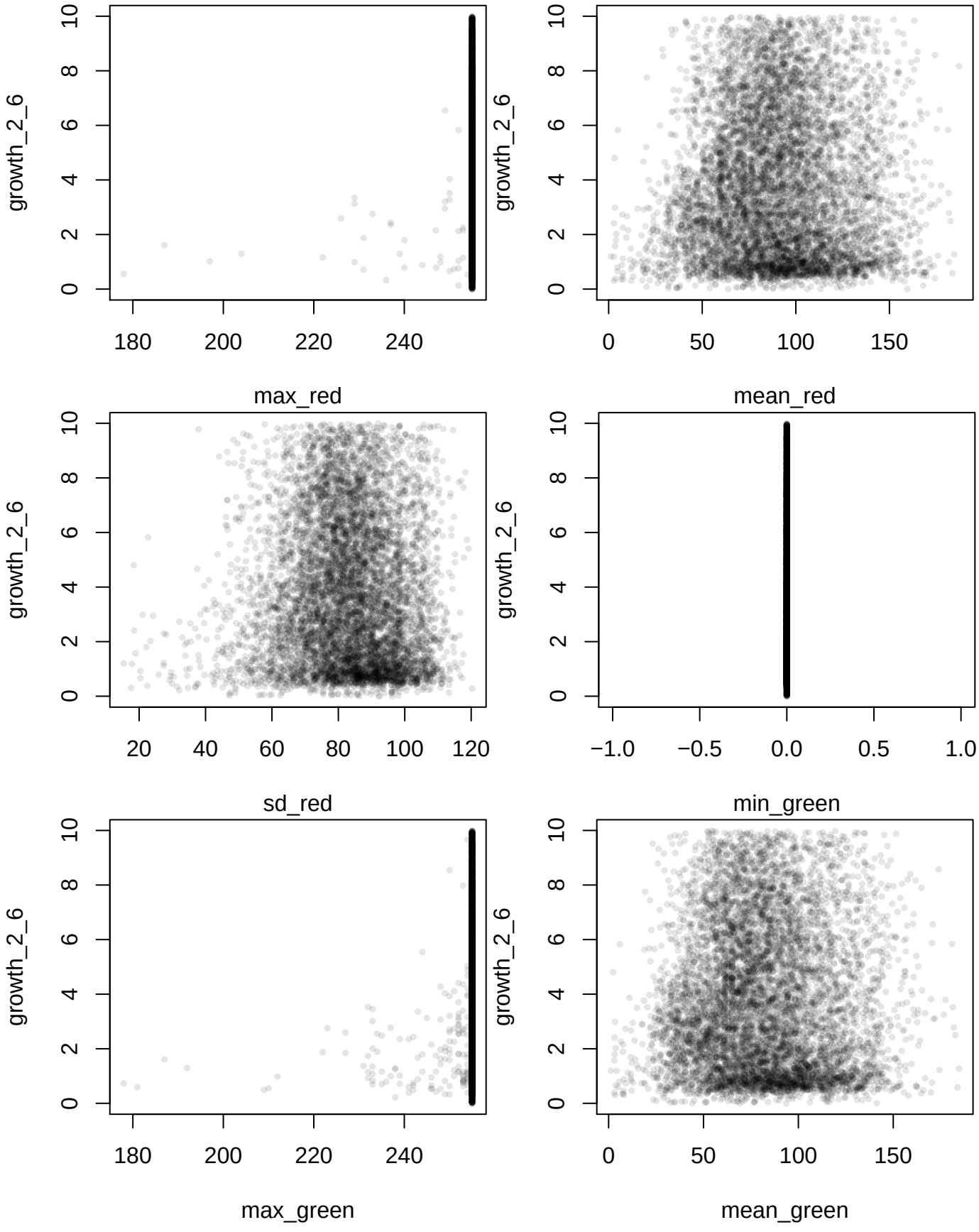
```
no_hog <- training %>% select(-contains("hog"))
plot(growth_2_6 ~ ., data = no_hog,
     cex = 0.5, pch = 19, col = rgb(0, 0, 0, 0.1))
```

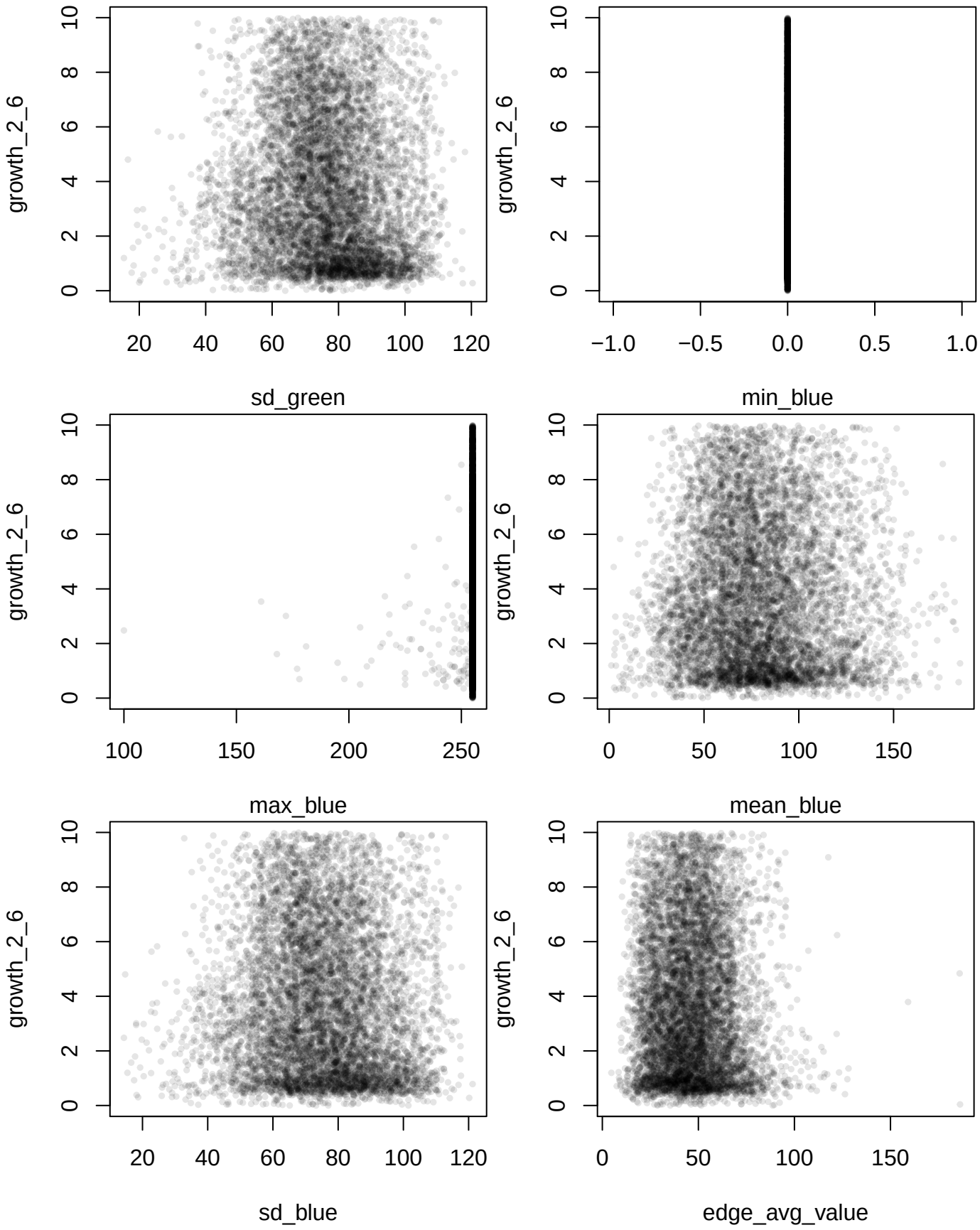

growth_2_6
0
2
4
6
8
10
150000
250000
350000
0
10000
30000
PublishedDate
0.0e+00
4.0e+06
8.0e+06
1.2e+07
Duration
0.0
0.5
1.0
1.5
2.0
views_2_hours
0.0
0.5
1.0
1.5
2.0
2.5
3.0
cnn_9
cnn_0
0
2
4
6
8
10
cnn_10

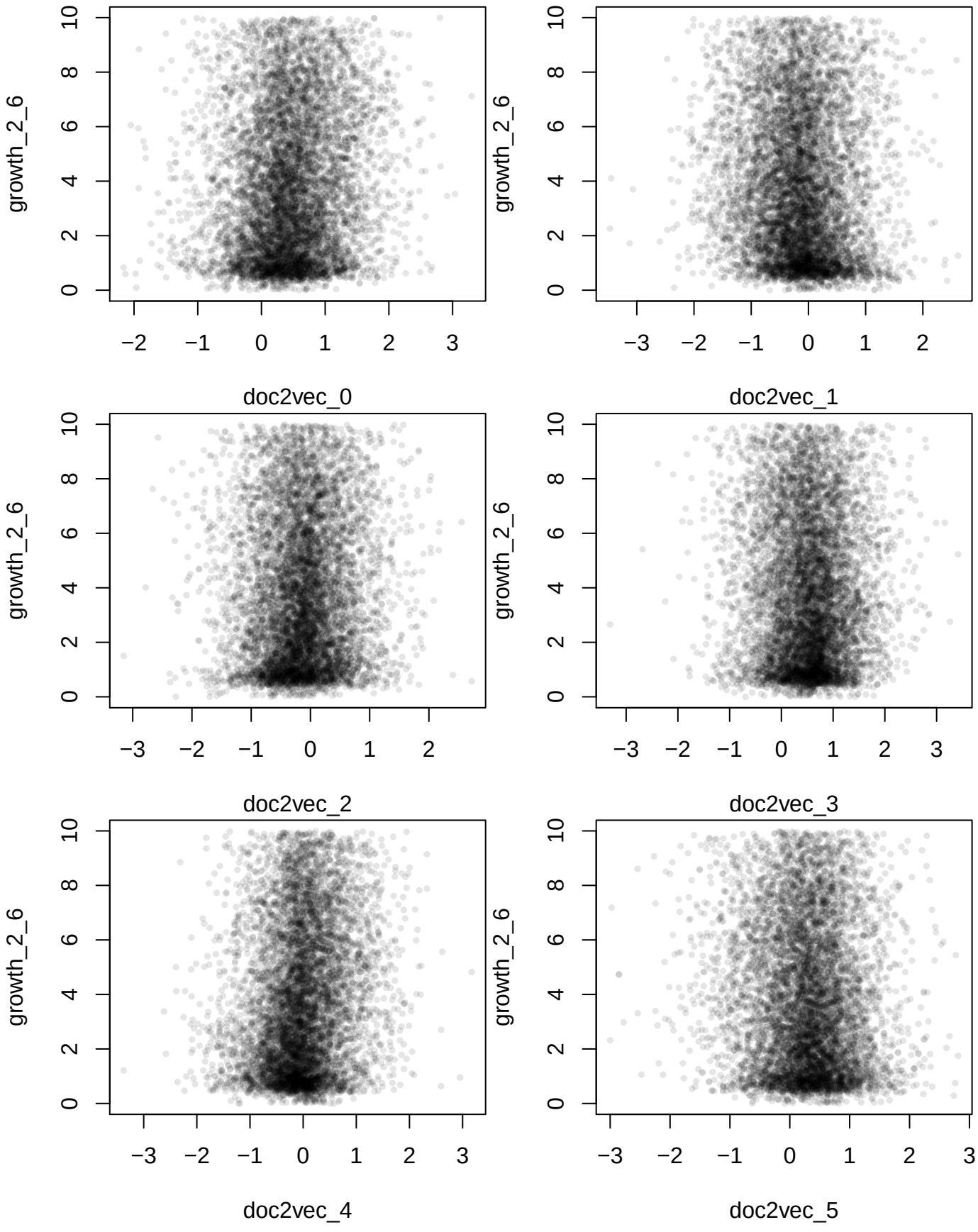

growth_2_6
cnn_12
cnn_17
cnn_19
cnn_20
cnn_25
cnn_35

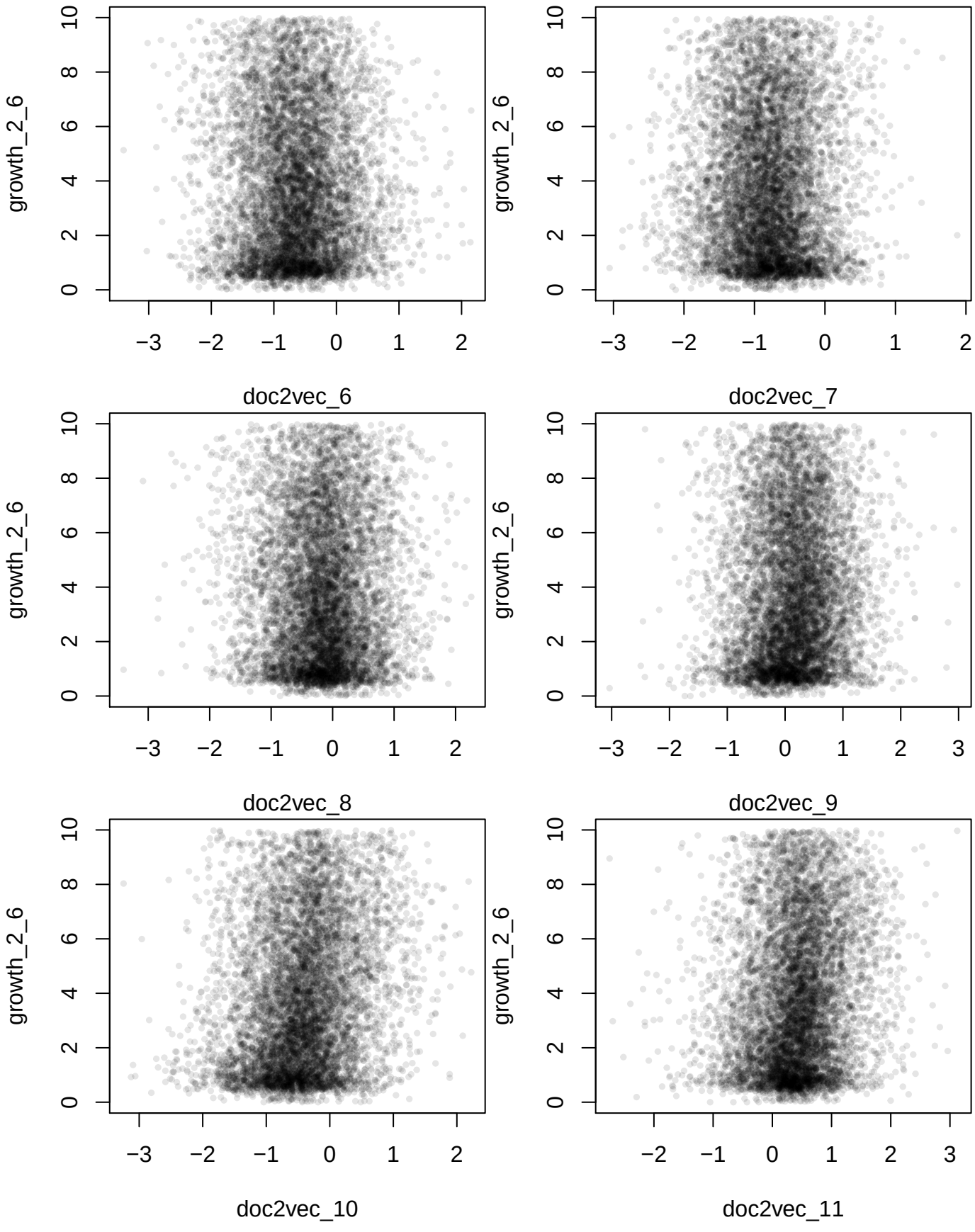

growth_2_6
cnn_36
cnn_37
cnn_39
cnn_65
cnn_68
cnn_86

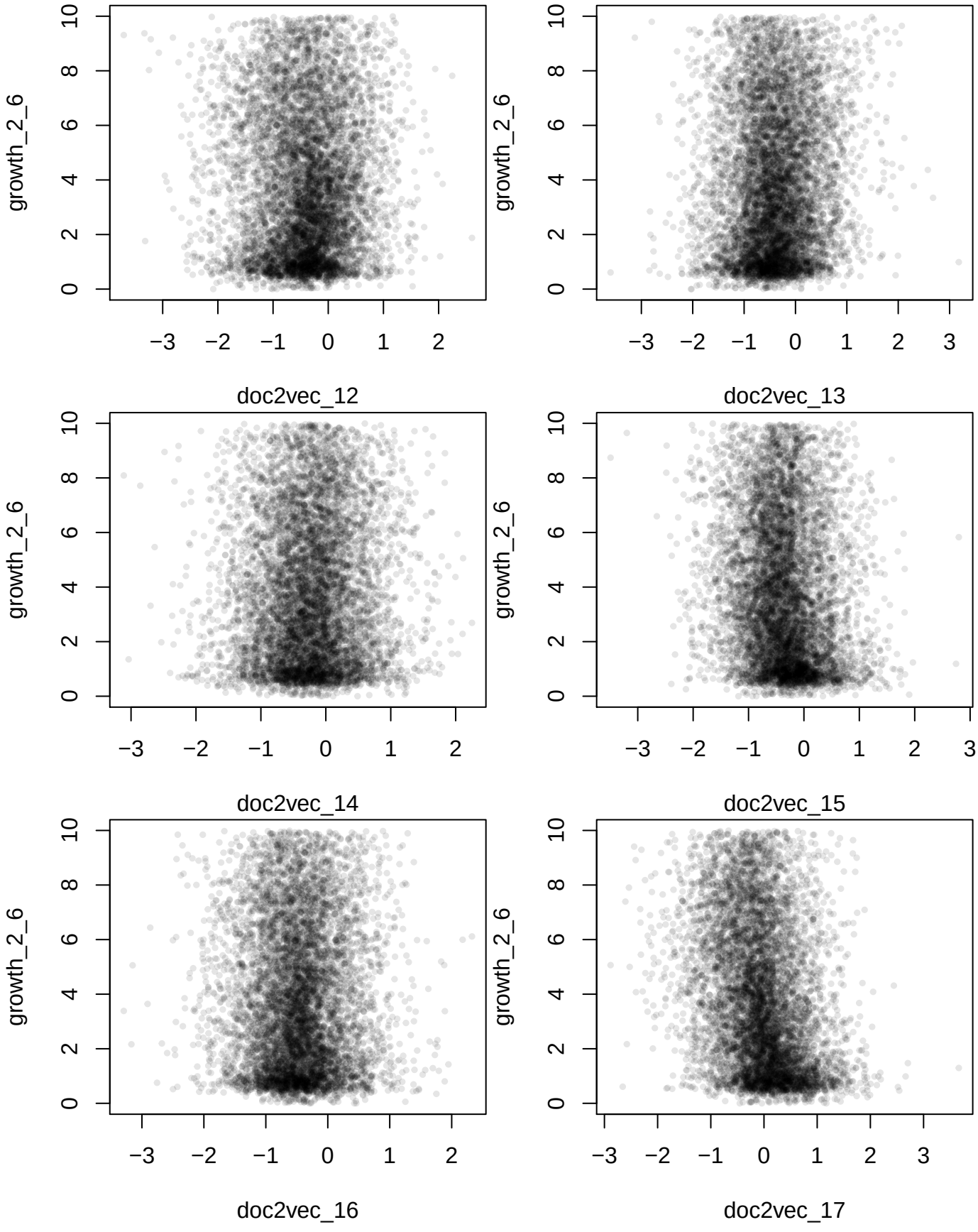

growth_2_6
0
2
4
6
8
10
0
5
10
15
0
2
4
6
8
10
12
cnn_88
0.0
0.2
0.4
0.6
0.8
1.0
cnn_89
0
50
100
150
pct_nonzero_pixels
20
40
60
80
100
120
sd_pixel_val
mean_pixel_val
−1.0
−0.5
0.0
0.5
1.0
min_red

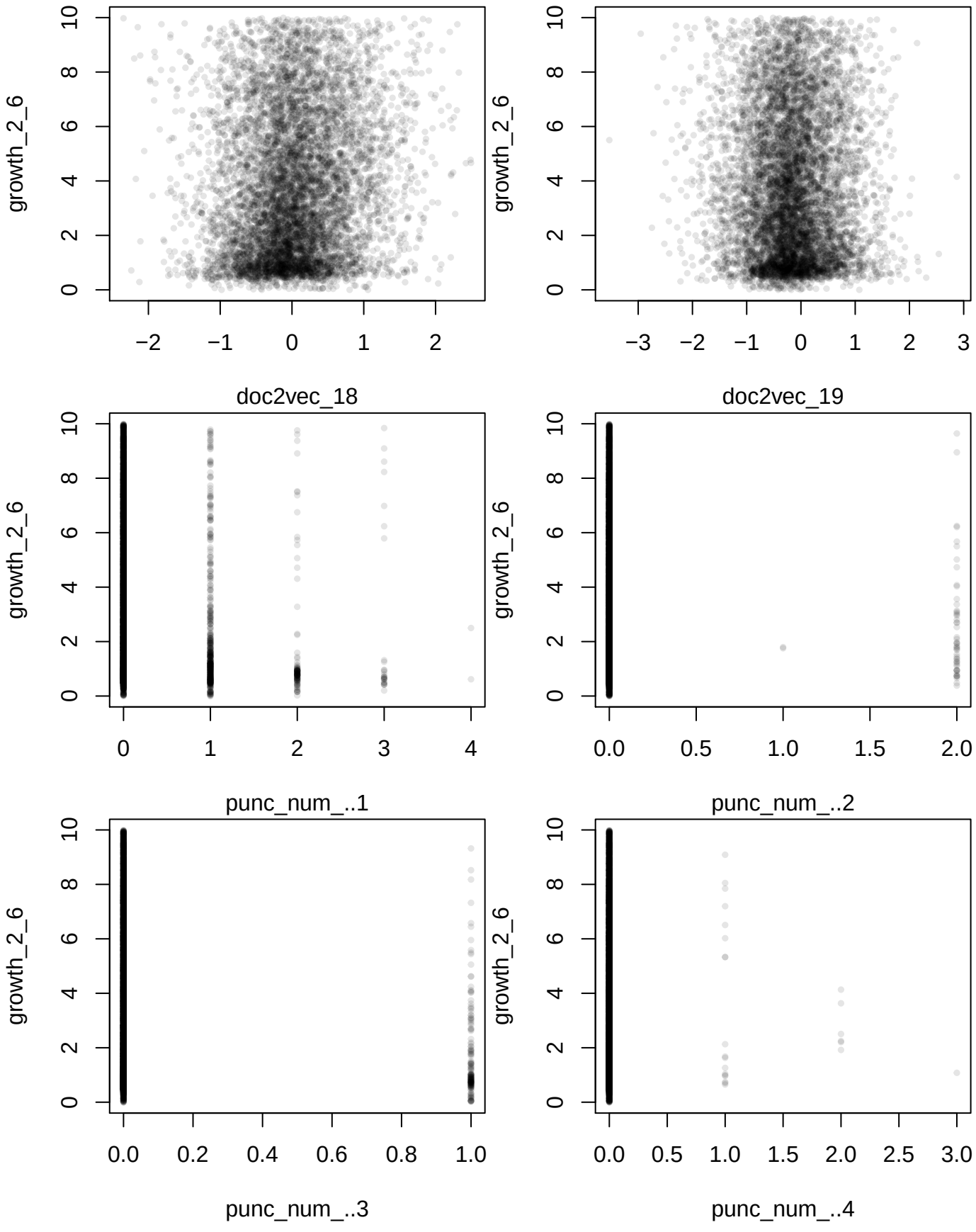

growth_2_6
max_red
mean_red
sd_red
min_green
max_green
mean_green

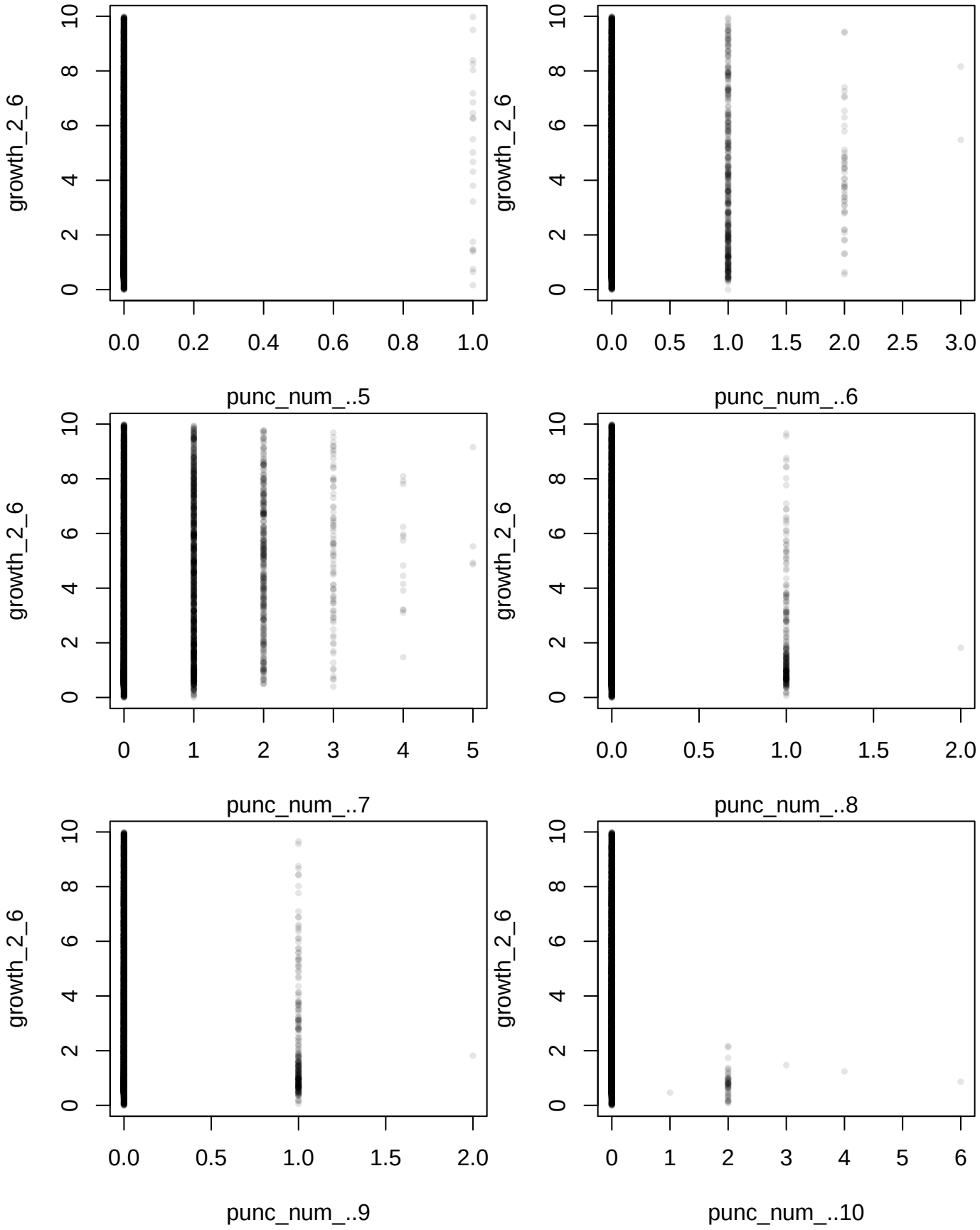

growth_2_6
sd_green
min_blue
max_blue
mean_blue
sd_blue
edge_avg_value

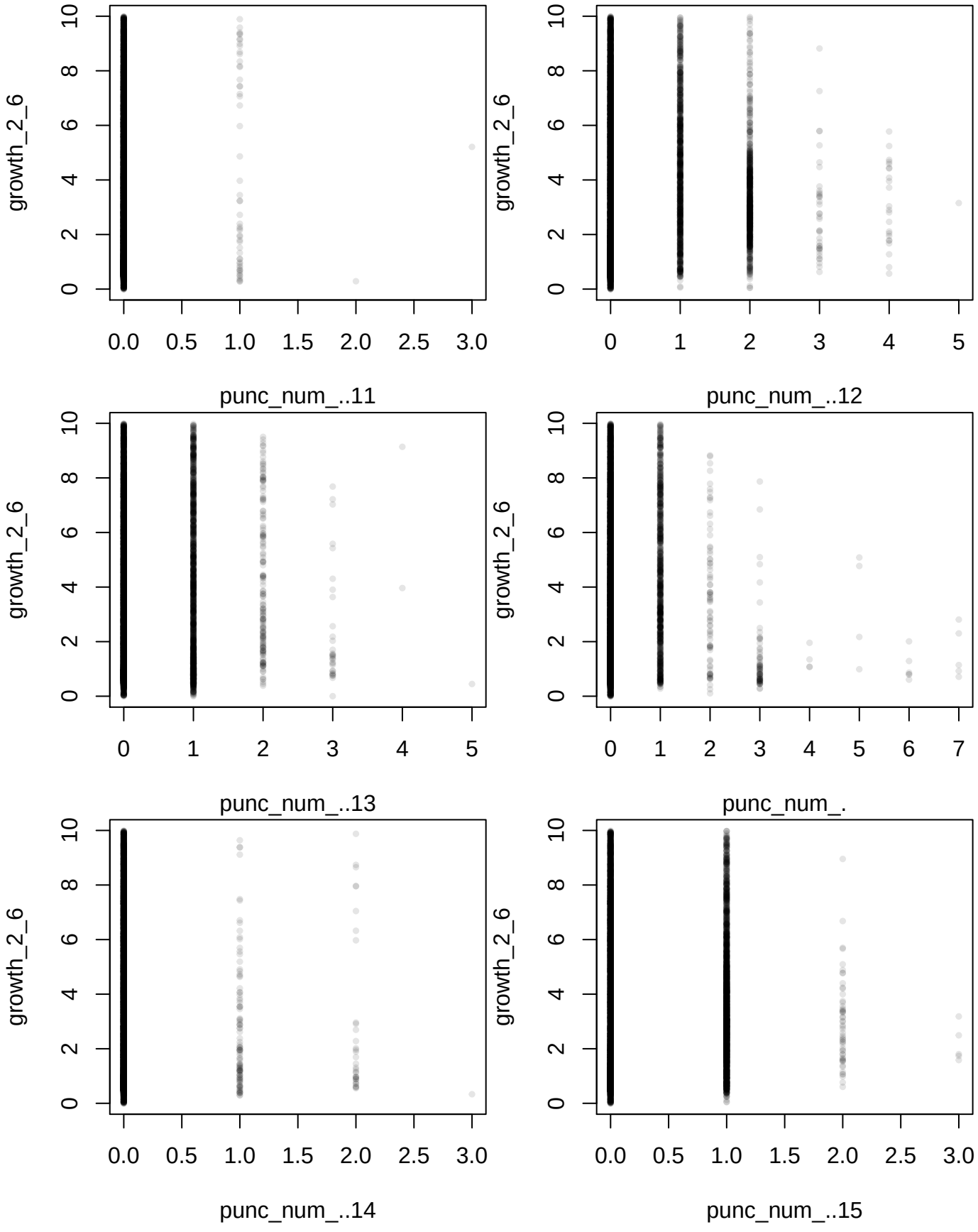

growth_2_6
doc2vec_0
doc2vec_1
doc2vec_2
doc2vec_3
doc2vec_4
doc2vec_5

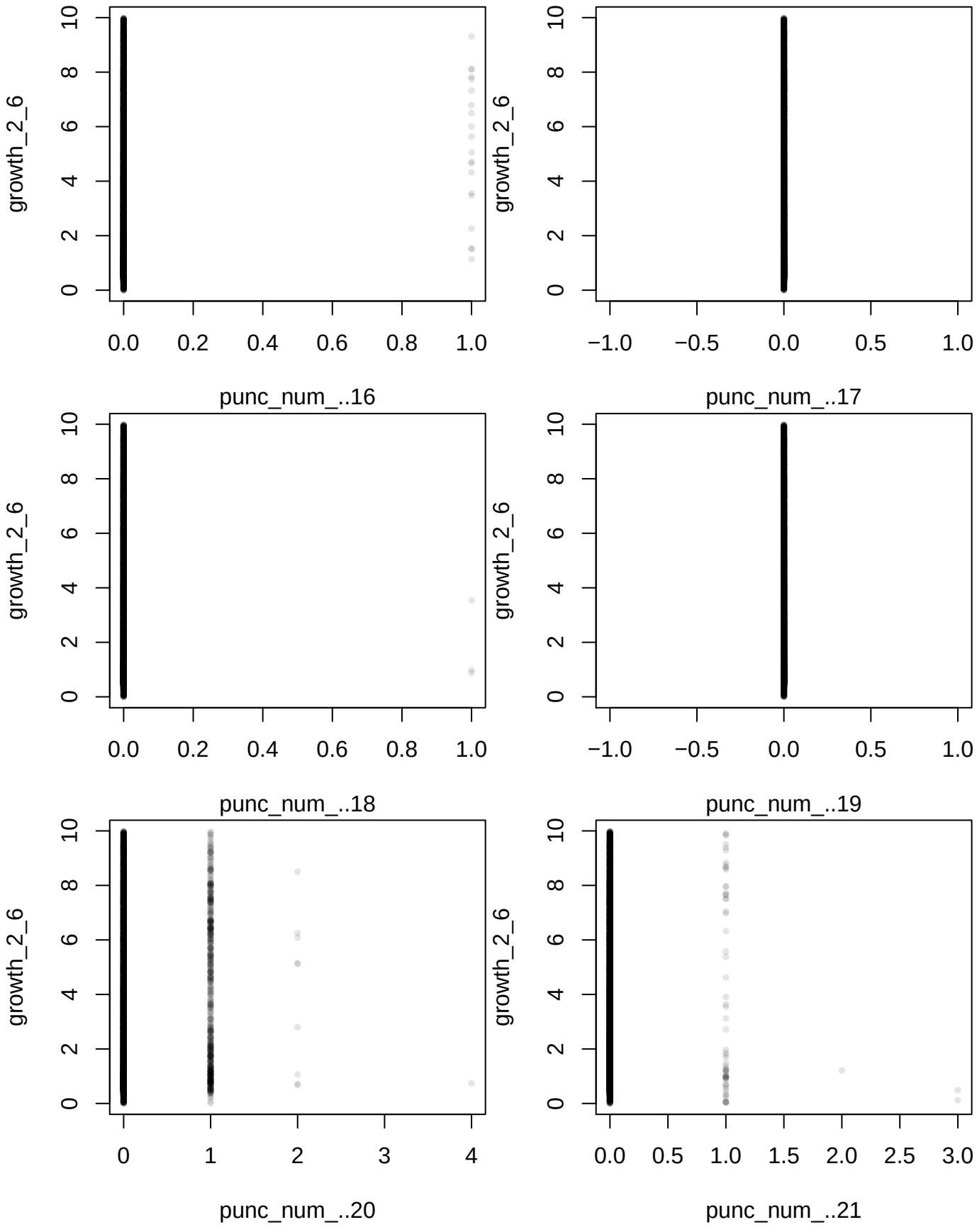

growth_2_6
doc2vec_6
doc2vec_7
doc2vec_8
doc2vec_9
doc2vec_10
doc2vec_11

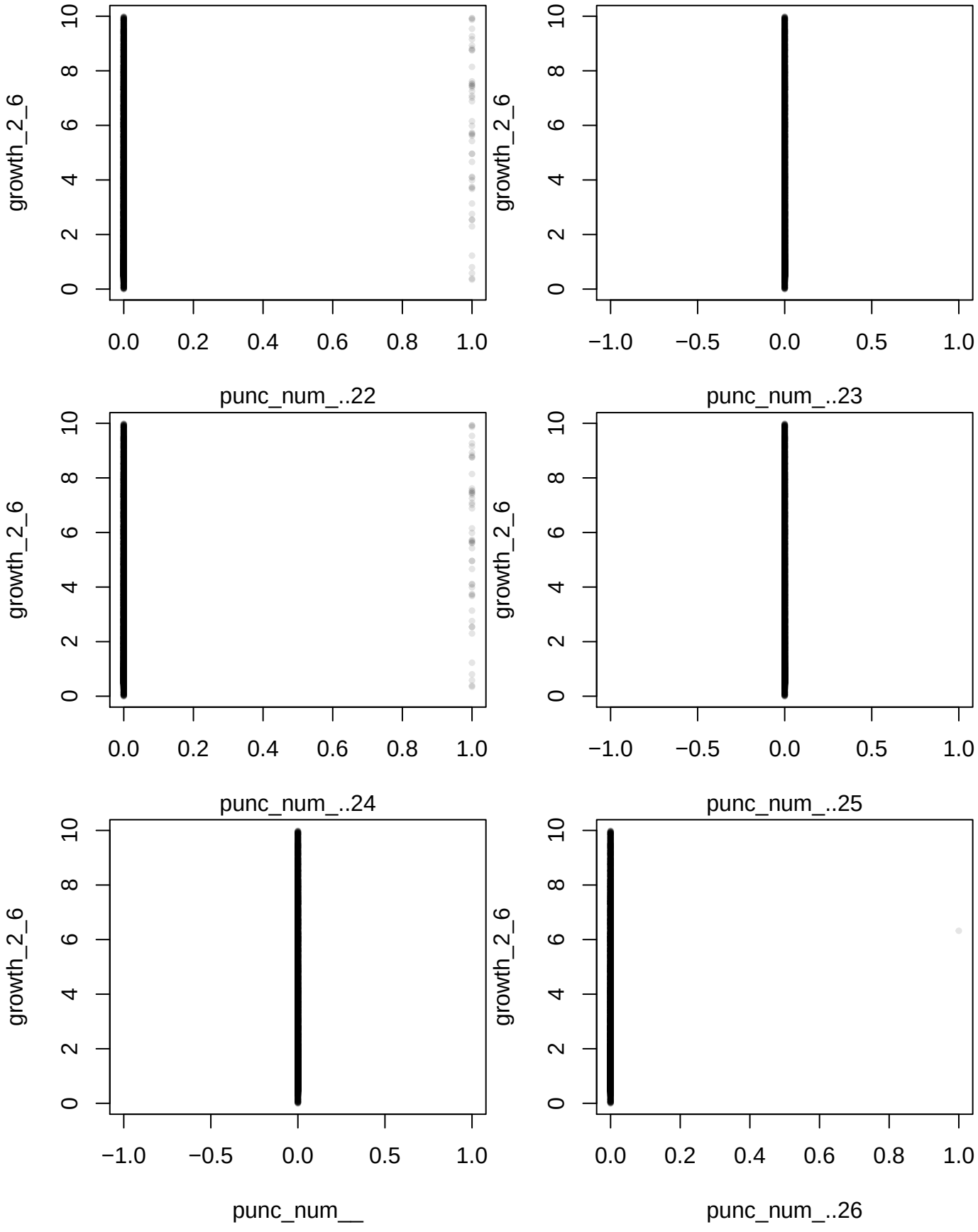

growth_2_6
doc2vec_12
doc2vec_13
doc2vec_14
doc2vec_15
doc2vec_16
doc2vec_17

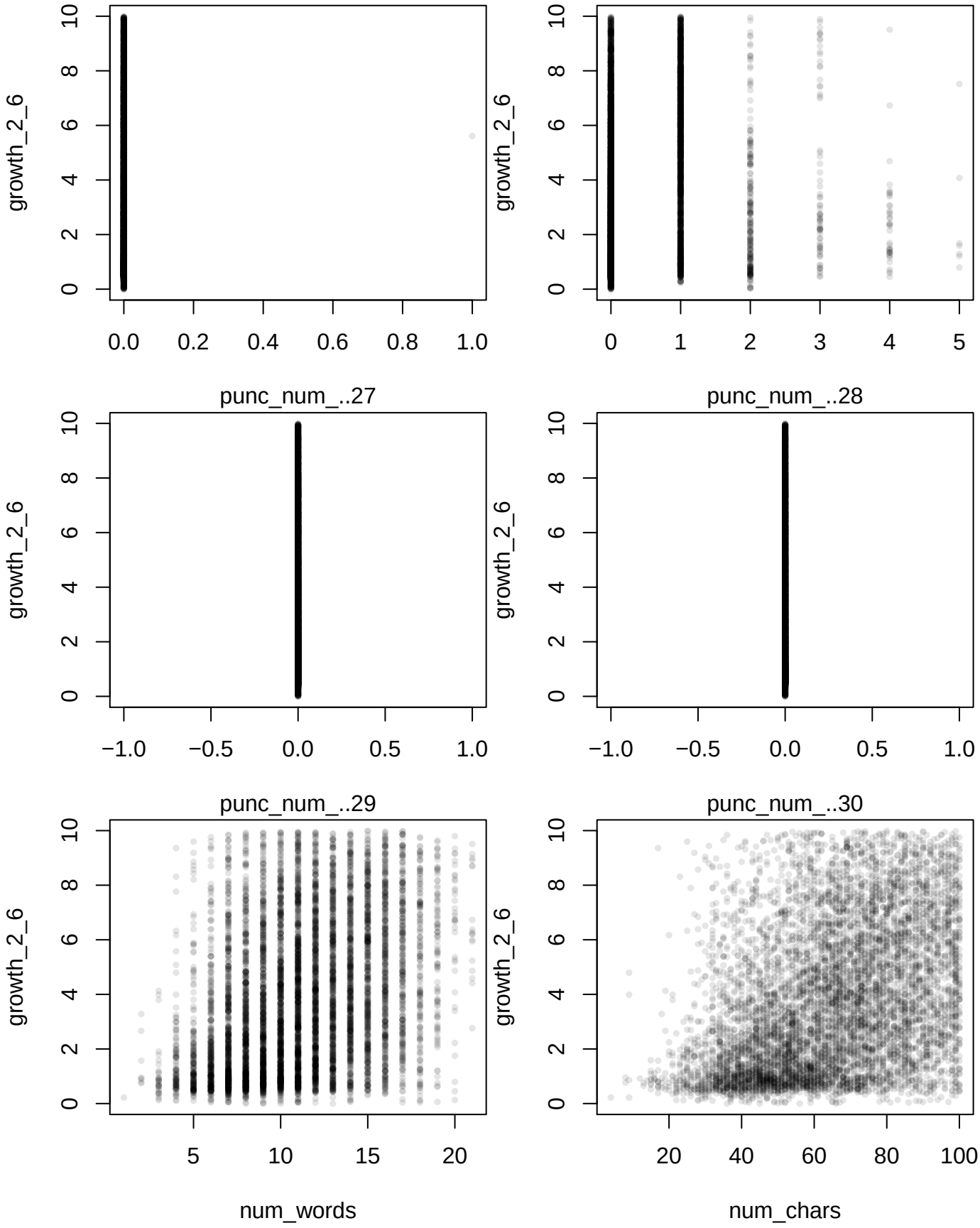

growth_2_6
doc2vec_18
doc2vec_19
punc_num_..1
punc_num_..2
punc_num_..3
punc_num_..4

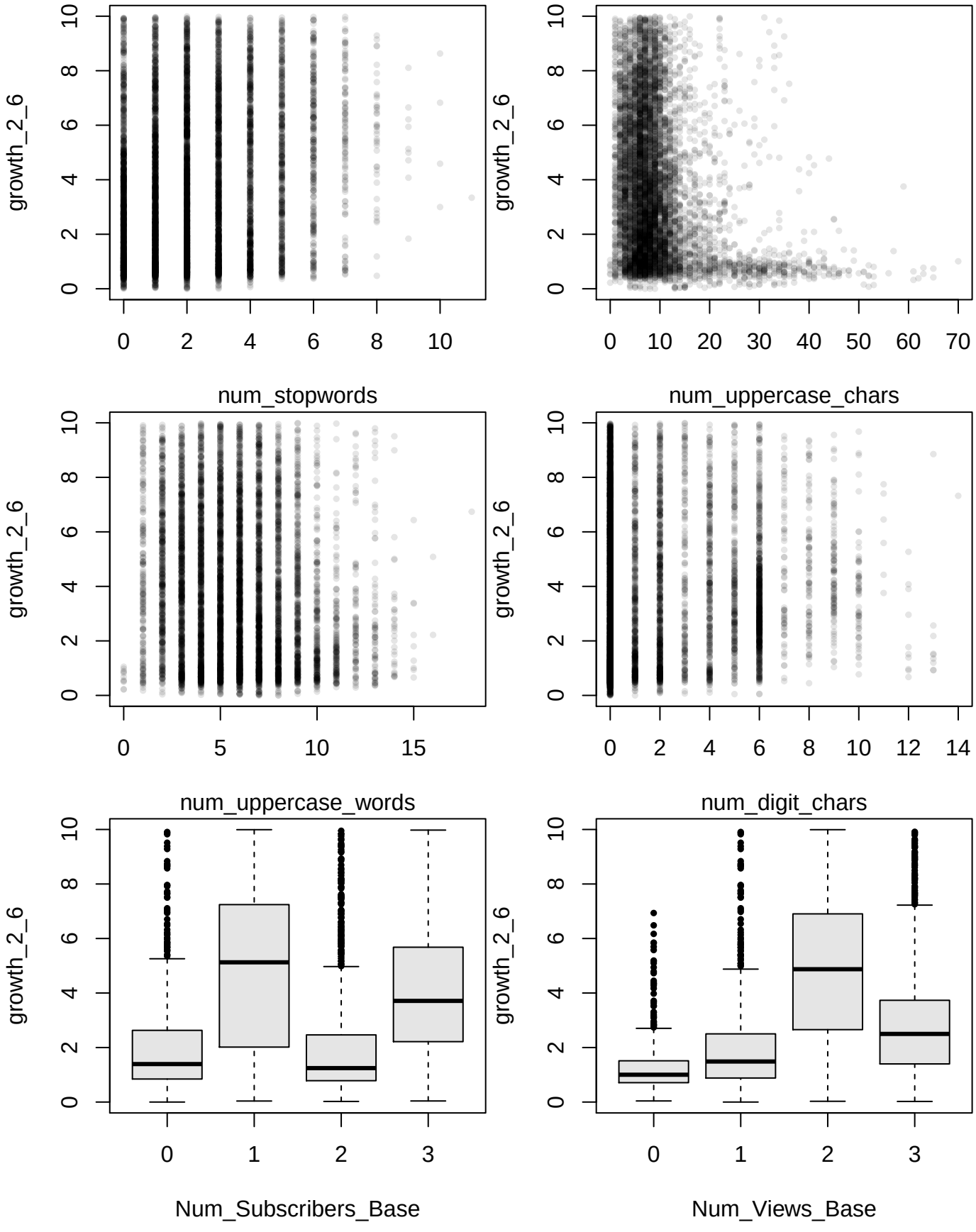

growth_2_6
punc_num_..5
punc_num_..6
punc_num_..7
punc_num_..8
punc_num_..9
punc_num_..10

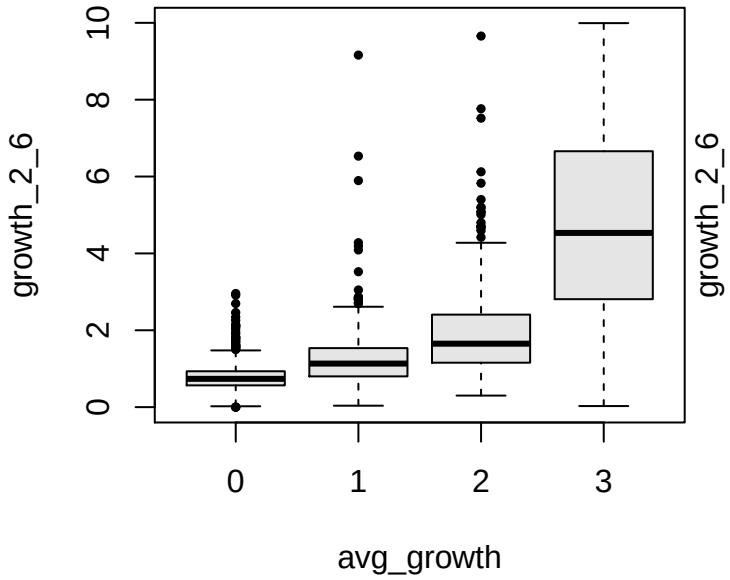

growth_2_6
punc_num_..11
punc_num_..12
punc_num_..13
punc_num_.
punc_num_..14
punc_num_..15

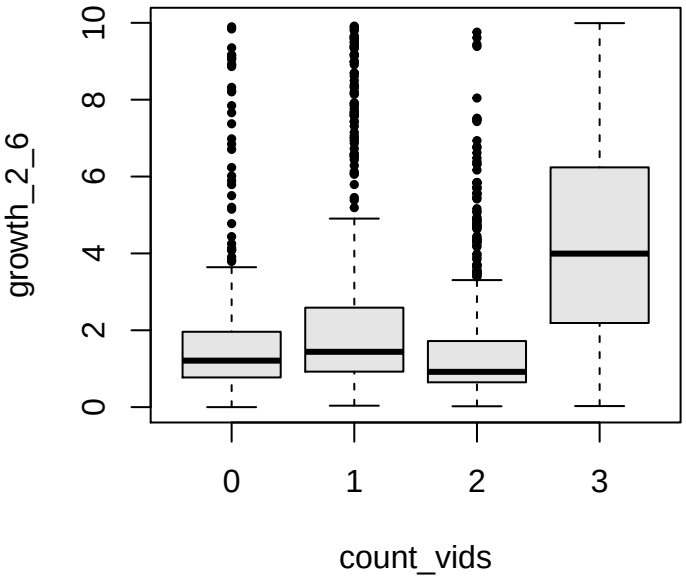

growth_2_6
punc_num_..16
punc_num_..17
punc_num_..18
punc_num_..19
punc_num_..20
punc_num_..21


growth_2_6
punc_num_..22
punc_num_..23
punc_num_..24
punc_num_..25
punc_num__
punc_num_..26


growth_2_6
0
2
4
6
8
10
0.0
0.2
0.4
0.6
0.8
1.0
0
1
2
3
4
5
punc_num_..27
punc_num_..28
−1.0
−0.5
0.0
0.5
1.0
punc_num_..29
punc_num_..30
5
10
15
20
num_words
20
40
60
80
100
num_chars


growth_2_6
num_stopwords
num_uppercase_chars
num_uppercase_words
num_digit_chars
Num_Subscribers_Base
Num_Views_Base


growth_2_6
0
2
4
6
8
10
0
1
2
3
avg_growth


growth_2_6
0
2
4
6
8
10
0
1
2
3
count_vids