Read Data

11/23/2021

```
library(foreign)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(e1071)
library(tree)
```

```
## Registered S3 method overwritten by 'tree':
##   method     from
##   print.tree cli
```

```
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(caret)
```

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

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

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(ggplot2)
library(dplyr)
library(tidyr)
library(tidyverse)
library(patchwork)
sesame <- read.dta("sesame.dta")
sesame <- sesame %>%
  mutate(site=factor(site)) %>%
  mutate(bodyDiff = postbody - prebody,
         letDiff = postlet - prelet,
         formDiff = postform - preform,
         numbDiff = postnumb - prenumb,
         relatDiff = postrelat - prerelat,
         clasfDiff = postclasf - preclasf)
sesame.sd <- sesame%>%
  mutate(sd_pBod = scale(prebody, center = TRUE, scale = TRUE),
         sd_plet = scale(prelet, center = TRUE, scale = TRUE),
         sd_pform = scale(preform, center = TRUE, scale = TRUE),
         sd_pnumb = scale(prenumb, center = TRUE, scale = TRUE),
         sd_prelat = scale(prerelat, center = TRUE, scale = TRUE),
         sd_pclasf = scale(preclasf, center = TRUE, scale = TRUE),
         sd_peabody = scale(peabody, center = TRUE, scale = TRUE),
         sd_age = scale(age, center =TRUE, scale = TRUE),
         male=if_else(sex==1, 1, 0),
         female=if_else(sex==2, 1, 0))
```

Exploratory Data Analysis

```r
head(sesame)
```

```
##   rownames id site sex age viewcat setting viewenc prebody prelet preform
## 1        1  1    1   1  66       1       2       1      16     23      12
## 2        2  2    1   2  67       3       2       1      30     26       9
## 3        3  3    1   1  56       3       2       2      22     14       9
## 4        4  4    1   1  49       1       2       2      23     11      10
## 5        5  5    1   1  69       4       2       2      32     47      15
## 6        6  6    1   2  54       3       2       2      29     26      10
##   prenumb prerelat preclasf postbody postlet postform postnumb postrelat
## 1      40       14       20       18      30       14       44        14
## 2      39       16       22       30      37       17       39        14
## 3       9        9        8       21      46       15       40         9
## 4      14        9       13       21      14       13       19         8
## 5      51       17       22       32      63       18       54        14
## 6      33       14       14       27      36       14       39        16
##   postclasf peabody agecat encour _Isite_2 _Isite_3 _Isite_4 _Isite_5 regular
## 1        23      62      1      1        0        0        0        0       0
## 2        22       8      1      1        0        0        0        0       1
## 3        19      32      1      0        0        0        0        0       1
## 4        15      27      0      0        0        0        0        0       0
## 5        21      71      1      0        0        0        0        0       1
## 6        24      32      1      0        0        0        0        0       1
##   bodyDiff letDiff formDiff numbDiff relatDiff clasfDiff
## 1        2       7        2        4         0         3
```

```
## 2         0        11          8          0         -2          0
## 3        -1        32          6         31          0         11
## 4        -2         3          3          5         -1          2
## 5         0        16          3          3         -3         -1
## 6        -2        10          4          6          2         10
```

Variables:

The ID refers to a subject's identification number. The site refers to the age and background information of the child. A site value of 1 indicates a 3-5 year old disadvantaged child from the inner city. A site value of 2 represents a 4 year old advantaged child from the suburbs. A value of 3 represents an advantaged rural child. A site value of 4 indicates a disadvantaged rural child. Lastly, a value of 5 represents a disadvantaged Spanish speaking child. For the sex, a value of 1 indicates male, and a value of 2 indicates female. The age category is the child's age in months. The viewcat column is the frequency of viewing Sesame Street (1 = rarely, 2 = once/twice per week, 3 = 3-5 times a week, 4 = more than 5 times per week). The setting is where Sesame Street was viewed; a value of 1 indicates home and a value of 2 indicates school. The viewenc column refers to if the child was encouraged to watch or not (1 = child not encouraged, 2 = child encouraged). Encour is the same variable but with values 0 and 1, respectively. Regular is an indicator variable representing if a child is a regular viewer (0 = rarely watched, 1 = watched once per week or greater).

The prebody, prelet, preform, prenumb, prerelat, and preclasf columns all decribe pretest scores on varying types of assessments (body parts, letters, forms, numbers, relational terms, and classification skills, respectively). The columns labelled postbody, postlet, postform, postnumb, postrelat, and postclasf are the children's respective posttest scores. Above, we created the following variables - bodydiff, letDiff, formDiff, numbDiff, relatDiff, clasfDiff - to represent the difference in posttest scores and pretest scores for each child. Lastly, peabody represents a score of "mental age" for vocabulary maturity from the Peabody Picture Vocabulary Test.

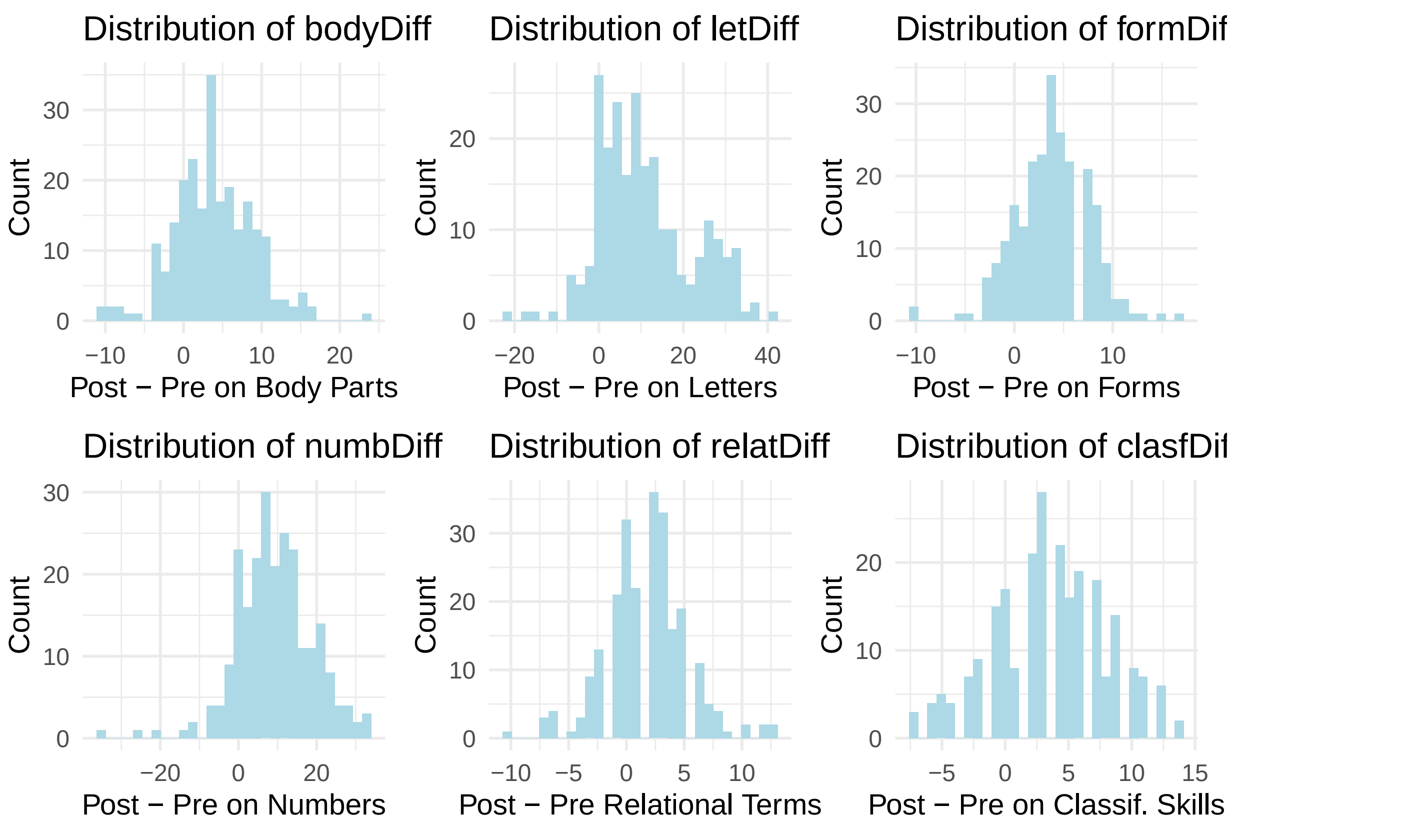
Our main focus will be on the new variables we created (bodyDiff, letDiff, formDiff, numbDiff, relatDiff, clasfDiff) and variables related to how often the children watch Sesame Street (namely, viewcat and regular). Lastly, we will look into the backgrounds of the children, including site, sex, and age.

Distributions:

For the purposes of our analysis, we will first look at the distributions of bodyDiff, letDiff, formDiff, numbDiff, relatDiff, and clasfDiff.

```r
#want to visualize distributions of bodyDiff, letDiff, formDiff, numbDiff, relatDiff, clasfDiff

bodyDiffplot <- ggplot(sesame, aes(x = bodyDiff)) +
  geom_histogram(fill = "lightblue") +
  labs(title = "Distribution of bodyDiff", x = "Post - Pre on Body Parts", y = "Count") +
  theme_minimal()

letDiffplot <- ggplot(sesame, aes(x = letDiff)) +
  geom_histogram(fill = "lightblue") +
  labs(title = "Distribution of letDiff", x = "Post - Pre on Letters", y = "Count") +
  theme_minimal()

formDiffplot <- ggplot(sesame, aes(x = formDiff)) +
  geom_histogram(fill = "lightblue") +
  labs(title = "Distribution of formDiff", x = "Post - Pre on Forms", y = "Count") +
  theme_minimal()

numbDiffplot <- ggplot(sesame, aes(x = numbDiff)) +
  geom_histogram(fill = "lightblue") +
```

```
  labs(title = "Distribution of numbDiff", x = "Post - Pre on Numbers", y = "Count") +
  theme_minimal()

relatDiffplot <- ggplot(sesame, aes(x = relatDiff)) +
  geom_histogram(fill = "lightblue") +
  labs(title = "Distribution of relatDiff", x = "Post - Pre Relational Terms", y = "Count") +
  theme_minimal()

clasfDiffplot <- ggplot(sesame, aes(x = clasfDiff)) +
  geom_histogram(fill = "lightblue") +
  labs(title = "Distribution of clasfDiff", x = "Post - Pre on Classif. Skills", y = "Count") +
  theme_minimal()


bodyDiffplot + letDiffplot + formDiffplot + numbDiffplot + relatDiffplot + clasfDiffplot
```

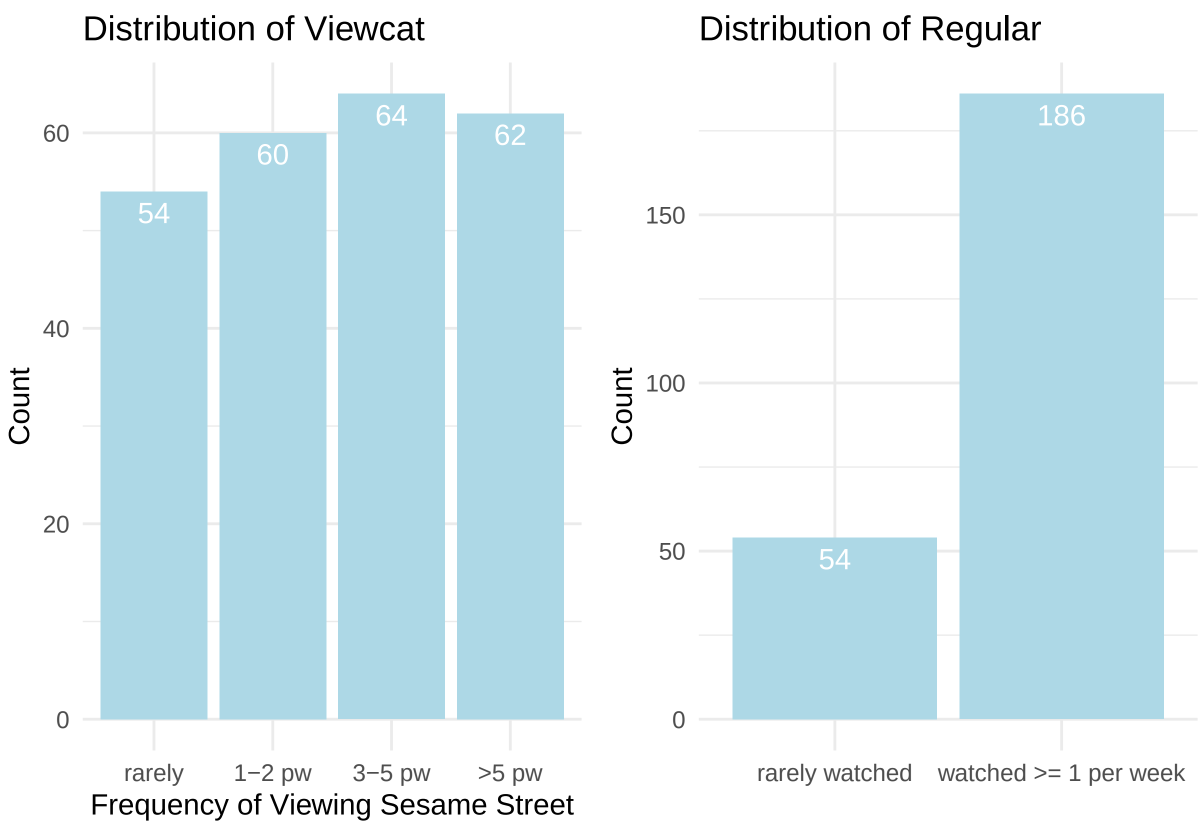


The six variables above were calculated by subtracting pre-test scores from post-test scores, so they are all numerical. The distributions of these six variables (bodyDiff, letDiff, formDiff, numbDiff, relatDiff, and clasfDiff) all appear to be roughly normal and unimodal. BodyDiff, letDiff, formDiff, relatDiff, and classDiff do not appear to have any obvious extreme outliers. Numbdiff, however, seems to be slightly left-skewed with outliers to the left -20. All of the six variables appear to have centers between 2 and 4.

We will now examine the distrubtions of the variables related to how often children watch Sesame Street (namely, viewcat and regular).

```
# want to visualize distributions of viewcat and regular

viewcatplot <- ggplot(sesame, aes(x = factor(viewcat))) +
  geom_bar(fill = "lightblue") +
```

```
  labs(title = "Distribution of Viewcat", x = "Frequency of viewing Sesame Street", y = "Count") +
  scale_x_discrete("Frequency of Viewing Sesame Street", labels=c("rarely", "1-2 pw", "3-5 pw", ">5 pw"
  theme_minimal() +
  geom_text(aes(label = ..count..), stat = "count", vjust = 1.5, colour = "white")


regularplot <- ggplot(sesame, aes(x = factor(regular))) +
  geom_bar(fill = "lightblue") +
  labs(title = "Distribution of Regular", y = "Count") +
  scale_x_discrete(labels=c("rarely watched", "watched >= 1 per week")) +
  theme_minimal() +
  theme(axis.title.x = element_blank()) +
  geom_text(aes(label = ..count..), stat = "count", vjust = 1.5, colour = "white")

viewcatplot + regularplot
```

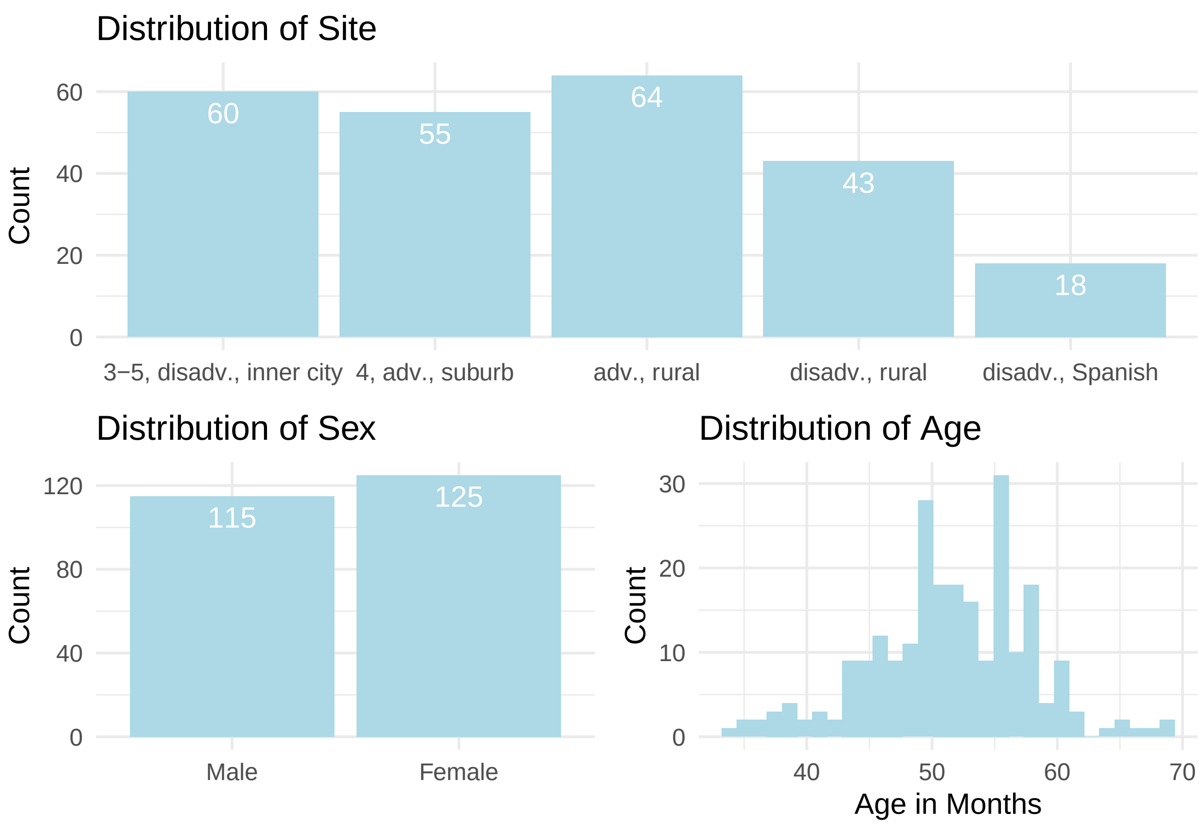


Both of these variables are categorical. On the left, viewcat appears to have a roughly uniform distribution, with "rarely" having the least amount of children and 3-5 times per week having the most (the range is only 10 children, so all of the bars are relatively close in height). For the variable regular, the category "watched once per week or greater" has far more observations than "rarely watched." The former category has more than triple the amount of the latter. We will be aware of this disparity in our analysis and continue with caution towards potential bias.

Lastly, we want to examine the distributions of site, sex, and age, all variables that relate to a child's background.

```
# want to visualize distributions of site, sex, and age

siteplot <- ggplot(sesame, aes(x = factor(site))) +
```

```
  geom_bar(fill = "lightblue") +
  labs(title = "Distribution of Site", y = "Count") +
  scale_x_discrete(labels=c("3-5, disadv., inner city", "4, adv., suburb", "adv., rural", "disadv., rura
  theme_minimal() +
  theme(axis.title.x = element_blank()) +
  geom_text(aes(label = ..count..), stat = "count", vjust = 1.5, colour = "white")

sexplot <- ggplot(sesame, aes(x = factor(sex))) +
  geom_bar(fill = "lightblue") +
  labs(title = "Distribution of Sex", y = "Count") +
  scale_x_discrete(labels=c("Male", "Female")) +
  theme_minimal() +
  theme(axis.title.x = element_blank()) +
  geom_text(aes(label = ..count..), stat = "count", vjust = 1.5, colour = "white")

ageplot <- ggplot(sesame, aes(x = age)) +
  geom_histogram(fill = "lightblue") +
  labs(title = "Distribution of Age", x = "Age in Months", y = "Count") +
  theme_minimal()

siteplot / (sexplot + ageplot)
```



Distribution of Site and Sex are both categorical variables. Distribution of Site has four categories with roughly the same amount of children (ranging from 43 to 64), but one category with far fewer observations (disadvantaged Spanish-speaking). This cateogry has less than half of the observations as the next smallest category, which is a relatively large disparity. We will continue our analysis with caution towards this bias in the data. The distribution of sex is very even - the male category has 115 observations, while the female category has 125 observations. Age is a numerical variable that appears to be normal and bimodal, with two peaks around 50 and 56. There do not appear to be any extreme outliers in the distribution of age.

Q.1 Prediction Question: Can we use linear regression to predict the change in a child's test scores that occur after watching Sesame street (or in some instances, not watching Seasame street)?

Here, I am fitting 7 linear regression models. The first 6 models predict a different difference in test score while the last model predicts the total difference in test scores.

```
sesame$site <- as.factor(sesame$site)
sesame$sex <- as.factor(sesame$sex)
sesame$viewcat <- as.factor(sesame$viewcat)
sesame$setting <- as.factor(sesame$setting)
sesame$viewenc <- as.factor(sesame$viewenc)
```

```
sesame <- sesame %>%
  mutate(totDiff = bodyDiff + letDiff + formDiff + numbDiff + relatDiff + clasfDiff)
```

```
lin.mod1 <- lm(bodyDiff ~ site + sex + age + viewcat + setting + viewenc, data = sesame)
summary(lin.mod1)
```

```
##
## Call:
## lm(formula = bodyDiff ~ site + sex + age + viewcat + setting +
##     viewenc, data = sesame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8472  -2.9931  -0.1636   2.9872  15.7957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.37538    2.91287   2.875  0.00442 **
## site2       -0.40192    0.94226  -0.427  0.67011
## site3        1.79273    0.93470   1.918  0.05636 .
## site4        2.78127    1.04971   2.650  0.00862 **
## site5        2.98238    1.43311   2.081  0.03855 *
## sex2        -0.27325    0.64579  -0.423  0.67260
## age         -0.13087    0.05431  -2.410  0.01676 *
## viewcat2     1.14604    1.04116   1.101  0.27217
## viewcat3     1.80580    1.05592   1.710  0.08860 .
## viewcat4     2.55334    1.06156   2.405  0.01696 *
## setting2    -0.51477    0.74974  -0.687  0.49303
## viewenc2     0.11079    0.77336   0.143  0.88621
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.941 on 228 degrees of freedom
## Multiple R-squared:  0.09493,    Adjusted R-squared:  0.05126
## F-statistic: 2.174 on 11 and 228 DF,  p-value: 0.01665
```

```
# training MSE
mean(summary(lin.mod1)$residuals^2)
```

```
## [1] 23.18944
```

```
# cross validation metrics
m1.ctrl <- trainControl(method = "cv", number = 5)
m1.cv <- train(bodyDiff ~ site + sex + age + viewcat + setting + viewenc,
                     data = sesame,
                     trControl = m1.ctrl,
                     method = "lm",
                     na.action = na.pass)
m1.cv
```

```
## Linear Regression
##
## 240 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 192, 192, 192, 191, 193
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   4.964177  0.05429436  3.865897
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
AIC(lin.mod1)
```

```
## [1] 1461.578
```

```
lin.mod1.full <- lm(bodyDiff ~ (site + sex + age + viewcat + setting + viewenc)^2, data = sesame)
AIC(lin.mod1.full)
```

```
## [1] 1497.39
```

```
# Split data between x and y
x <- model.matrix(bodyDiff ~ site + sex + age + viewcat + setting + viewenc,sesame)[,-1]
y <- sesame$bodyDiff

# set seed
set.seed(1)

# cross validation for lambda
# I couldn't run cv.glmnet, so I commented it out, make sure to uncomment these calls.
# cv.out <- cv.glmnet(x, y, alpha = 1) # setting alpha = 1 indicates lasso regression

# optimal lambda value
#best.lam <- cv.out$lambda.min

# lasso regression model with optimal lambda
#las.mod <- glmnet(x, y, alpha = 1, lambda = best.lam)
```

```
lin.mod2 <- lm(letDiff ~ site + sex + age + viewcat + setting + viewenc, data = sesame)
summary(lin.mod2)
```

```
##
## Call:
## lm(formula = letDiff ~ site + sex + age + viewcat + setting +
##     viewenc, data = sesame)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -33.791  -5.959  -0.300   5.645  23.831
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.53227    5.36242  -0.286   0.7753
## site2        7.46862    1.73463   4.306 2.47e-05 ***
## site3       -4.07540    1.72072  -2.368   0.0187 *
## site4        0.23354    1.93244   0.121   0.9039
## site5        0.91891    2.63827   0.348   0.7279
## sex2         0.85713    1.18886   0.721   0.4717
## age          0.10529    0.09998   1.053   0.2934
## viewcat2     4.38759    1.91671   2.289   0.0230 *
## viewcat3    10.56203    1.94389   5.433 1.42e-07 ***
## viewcat4    10.36156    1.95427   5.302 2.70e-07 ***
## setting2     0.04554    1.38022   0.033   0.9737
## viewenc2    -2.39825    1.42370  -1.685   0.0935 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.095 on 228 degrees of freedom
## Multiple R-squared:  0.3673, Adjusted R-squared:  0.3367
## F-statistic: 12.03 on 11 and 228 DF,  p-value: < 2.2e-16
```

```
# training MSE
mean(summary(lin.mod2)$residuals^2)
```

```
## [1] 78.59028
```

```
# cross validation metrics
m2.ctrl <- trainControl(method = "cv", number = 5)
m2.cv <- train(letDiff ~ site + sex + age + viewcat + setting + viewenc,
                     data = sesame,
                     trControl = m2.ctrl,
                     method = "lm",
                     na.action = na.pass)
m2.cv
```

```
## Linear Regression
##
## 240 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 192, 192, 192, 192, 192
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   9.395665  0.3040119  7.316402
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
AIC(lin.mod2)
```

```
## [1] 1754.51
```

```r
lin.mod2.full <- lm(letDiff ~ (site + sex + age + viewcat + setting + viewenc)^2, data = sesame)
AIC(lin.mod2.full)
```

```
## [1] 1780.128
```

```r
lin.mod3 <- lm(formDiff ~ site + sex + age + viewcat + setting + viewenc, data = sesame)
summary(lin.mod3)
```

```
##
## Call:
## lm(formula = formDiff ~ site + sex + age + viewcat + setting +
##     viewenc, data = sesame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0591  -2.2388   0.0013   2.2777  12.6717
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.76068    2.19535   1.713   0.0881 .
## site2        0.75430    0.71015   1.062   0.2893
## site3        0.92224    0.70446   1.309   0.1918
## site4        0.27788    0.79113   0.351   0.7257
## site5        1.77471    1.08009   1.643   0.1017
## sex2         0.02527    0.48671   0.052   0.9586
## age         -0.02875    0.04093  -0.702   0.4832
## viewcat2     0.76459    0.78469   0.974   0.3309
## viewcat3     1.11501    0.79582   1.401   0.1625
## viewcat4     1.91846    0.80007   2.398   0.0173 *
## setting2    -0.14435    0.56506  -0.255   0.7986
## viewenc2    -0.02050    0.58286  -0.035   0.9720
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.724 on 228 degrees of freedom
## Multiple R-squared:  0.05761,    Adjusted R-squared:  0.01214
## F-statistic: 1.267 on 11 and 228 DF,  p-value: 0.2447
```

```r
# training MSE
mean(summary(lin.mod3)$residuals^2)
```

```
## [1] 13.17211
```

```r
# cross validation metrics
m3.ctrl <- trainControl(method = "cv", number = 5)
m3.cv <- train(formDiff ~ site + sex + age + viewcat + setting + viewenc,
                     data = sesame,
                     trControl = m3.ctrl,
                     method = "lm",
                     na.action = na.pass)
m3.cv
```

```
## Linear Regression
```

```
##
## 240 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 192, 192, 192, 192, 192
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   3.738912  0.01203227  2.848853
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
AIC(lin.mod3)
```

```
## [1] 1325.835
```

```
lin.mod3.full <- lm(formDiff ~ (site + sex + age + viewcat + setting + viewenc)^2, data = sesame)
AIC(lin.mod3.full)
```

```
## [1] 1354.506
```

```
lin.mod4 <- lm(numbDiff ~ site + sex + age + viewcat + setting + viewenc, data = sesame)
summary(lin.mod4)
```

```
##
## Call:
## lm(formula = numbDiff ~ site + sex + age + viewcat + setting +
##     viewenc, data = sesame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.409  -5.360  -0.175   5.716  22.755
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.28859    5.50981   1.141 0.254924
## site2        3.97388    1.78231   2.230 0.026748 *
## site3        1.88588    1.76802   1.067 0.287253
## site4        1.44408    1.98556   0.727 0.467796
## site5        4.82105    2.71079   1.778 0.076659 .
## sex2         0.44345    1.22154   0.363 0.716924
## age         -0.09483    0.10272  -0.923 0.356921
## viewcat2     4.14991    1.96939   2.107 0.036193 *
## viewcat3     6.72915    1.99732   3.369 0.000885 ***
## viewcat4     7.18587    2.00798   3.579 0.000422 ***
## setting2     1.94076    1.41816   1.369 0.172501
## viewenc2     0.03824    1.46284   0.026 0.979169
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.345 on 228 degrees of freedom
## Multiple R-squared:  0.1136, Adjusted R-squared:  0.07085
## F-statistic: 2.657 on 11 and 228 DF,  p-value: 0.0032
```

```r
# training MSE
mean(summary(lin.mod4)$residuals^2)
```

```
## [1] 82.97012
```

```r
# cross validation metrics
m4.ctrl <- trainControl(method = "cv", number = 5)
m4.cv <- train(numbDiff ~ site + sex + age + viewcat + setting + viewenc,
                     data = sesame,
                     trControl = m4.ctrl,
                     method = "lm",
                     na.action = na.pass)
m4.cv
```

```
## Linear Regression
##
## 240 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 191, 192, 192, 192, 193
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   9.736313  0.03522287  7.480826
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
AIC(lin.mod4)
```

```
## [1] 1767.526
```

```r
lin.mod4.full <- lm(numbDiff ~ (site + sex + age + viewcat + setting + viewenc)^2, data = sesame)
AIC(lin.mod4.full)
```

```
## [1] 1817.319
```

```r
lin.mod5 <- lm(relatDiff ~ site + sex + age + viewcat + setting + viewenc, data = sesame)
summary(lin.mod5)
```

```
##
## Call:
## lm(formula = relatDiff ~ site + sex + age + viewcat + setting +
##     viewenc, data = sesame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4245  -1.9159   0.1349   1.7887  11.5949
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.416806   1.967433   3.262  0.00128 **
## site2       -0.034343   0.636425  -0.054  0.95701
## site3        1.366734   0.631320   2.165  0.03143 *
## site4        1.912144   0.709000   2.697  0.00752 **
## site5       -0.465298   0.967961  -0.481  0.63119
```

```
## sex2        -0.005095   0.436183  -0.012  0.99069
## age         -0.116681   0.036681  -3.181  0.00167 **
## viewcat2     0.515579   0.703226   0.733  0.46421
## viewcat3     0.774911   0.713199   1.087  0.27839
## viewcat4     1.986853   0.717006   2.771  0.00605 **
## setting2    -0.181790   0.506393  -0.359  0.71994
## viewenc2    -0.273053   0.522347  -0.523  0.60166
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.337 on 228 degrees of freedom
## Multiple R-squared:  0.1208, Adjusted R-squared:  0.07843
## F-statistic: 2.849 on 11 and 228 DF,  p-value: 0.001621
```

```
# training MSE
mean(summary(lin.mod5)$residuals^2)
```

```
## [1] 10.57907
```

```
# cross validation metrics
m5.ctrl <- trainControl(method = "cv", number = 5)
m5.cv <- train(relatDiff ~ site + sex + age + viewcat + setting + viewenc,
                     data = sesame,
                     trControl = m5.ctrl,
                     method = "lm",
                     na.action = na.pass)
m5.cv
```

```
## Linear Regression
##
## 240 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 194, 191, 192, 192, 191
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   3.455163  0.06023434  2.60737
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
AIC(lin.mod5)
```

```
## [1] 1273.221
```

```
lin.mod5.full <- lm(relatDiff ~ (site + sex + age + viewcat + setting + viewenc)^2, data = sesame)
AIC(lin.mod5.full)
```

```
## [1] 1276.381
```

```
lin.mod6 <- lm(clasfDiff ~ site + sex + age + viewcat + setting + viewenc, data = sesame)
summary(lin.mod6)
```

```
##
## Call:
## lm(formula = clasfDiff ~ site + sex + age + viewcat + setting +
```

```
##     viewenc, data = sesame)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9046  -3.0679   0.4323   2.7764  10.5317
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.17711    2.61617   1.597   0.1117
## site2        0.94736    0.84628   1.119   0.2641
## site3        0.15156    0.83949   0.181   0.8569
## site4       -0.12651    0.94279  -0.134   0.8934
## site5        1.37810    1.28714   1.071   0.2854
## sex2         0.61877    0.58001   1.067   0.2872
## age         -0.04389    0.04878  -0.900   0.3692
## viewcat2     0.81364    0.93511   0.870   0.3852
## viewcat3     1.43903    0.94837   1.517   0.1306
## viewcat4     1.93911    0.95343   2.034   0.0431 *
## setting2     0.20761    0.67337   0.308   0.7581
## viewenc2    -0.66613    0.69459  -0.959   0.3386
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.437 on 228 degrees of freedom
## Multiple R-squared:  0.06353,    Adjusted R-squared:  0.01835
## F-statistic: 1.406 on 11 and 228 DF,  p-value: 0.171
```

```r
# training MSE
mean(summary(lin.mod6)$residuals^2)
```

```
## [1] 18.70599
```

```r
# cross validation metrics
m6.ctrl <- trainControl(method = "cv", number = 5)
m6.cv <- train(clasfDiff ~ site + sex + age + viewcat + setting + viewenc,
                     data = sesame,
                     trControl = m6.ctrl,
                     method = "lm",
                     na.action = na.pass)
m6.cv
```

```
## Linear Regression
##
## 240 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 191, 192, 192, 193, 192
## Resampling results:
##
##   RMSE      Rsquared    MAE
##   4.459824  0.03489856  3.610567
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```r
AIC(lin.mod6)
```

```
## [1] 1410.013
```

```r
lin.mod6.full <- lm(clasfDiff ~ (site + sex + age + viewcat + setting + viewenc)^2, data = sesame)
AIC(lin.mod6.full)
```

```
## [1] 1438.226
```

```r
lin.mod7 <- lm(totDiff ~ site + sex + age + viewcat + setting + viewenc, data = sesame)
summary(lin.mod7)
```

```
##
## Call:
## lm(formula = totDiff ~ site + sex + age + viewcat + setting +
##     viewenc, data = sesame)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -96.527 -13.115  -0.129  15.194  65.715
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.4863    14.2697   1.926  0.05532 .
## site2        12.7079     4.6160   2.753  0.00638 **
## site3         2.0437     4.5789   0.446  0.65578
## site4         6.5224     5.1423   1.268  0.20596
## site5        11.4099     7.0206   1.625  0.10550
## sex2          1.6663     3.1636   0.527  0.59892
## age          -0.3097     0.2660  -1.164  0.24558
## viewcat2     11.7773     5.1005   2.309  0.02183 *
## viewcat3     22.4259     5.1728   4.335 2.18e-05 ***
## viewcat4     25.9452     5.2004   4.989 1.20e-06 ***
## setting2      1.3530     3.6728   0.368  0.71293
## viewenc2     -3.2089     3.7885  -0.847  0.39788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.2 on 228 degrees of freedom
## Multiple R-squared:  0.2037, Adjusted R-squared:  0.1653
## F-statistic: 5.303 on 11 and 228 DF,  p-value: 1.842e-07
```

```r
# training MSE
mean(summary(lin.mod7)$residuals^2)
```

```
## [1] 556.5132
```

```r
# cross validation metrics
m7.ctrl <- trainControl(method = "cv", number = 5)
m7.cv <- train(totDiff ~ site + sex + age + viewcat + setting + viewenc,
                     data = sesame,
                     trControl = m7.ctrl,
                     method = "lm",
                     na.action = na.pass)
m7.cv
```

```
## Linear Regression
```

```
##
## 240 samples
##   6 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold)
## Summary of sample sizes: 193, 191, 193, 191, 192
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   24.65473  0.1412037  18.78817
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
AIC(lin.mod7)
```

```
## [1] 2224.296
```

```
lin.mod7.full <- lm(totDiff ~ (site + sex + age + viewcat + setting + viewenc)^2, data = sesame)
AIC(lin.mod7.full)
```

```
## [1] 2257.56
```

```
# df1 <- data.frame(Response = c("bodyDiff", "letDiff", "formDiff", "numbDiff", "relatDiff", "clasfDiff
# table <- kable(df1, caption = "MSE", booktabs=T)
# kable_styling(table, bootstrap_options = "striped", full_width = F, latex_options = "HOLD_position")
```

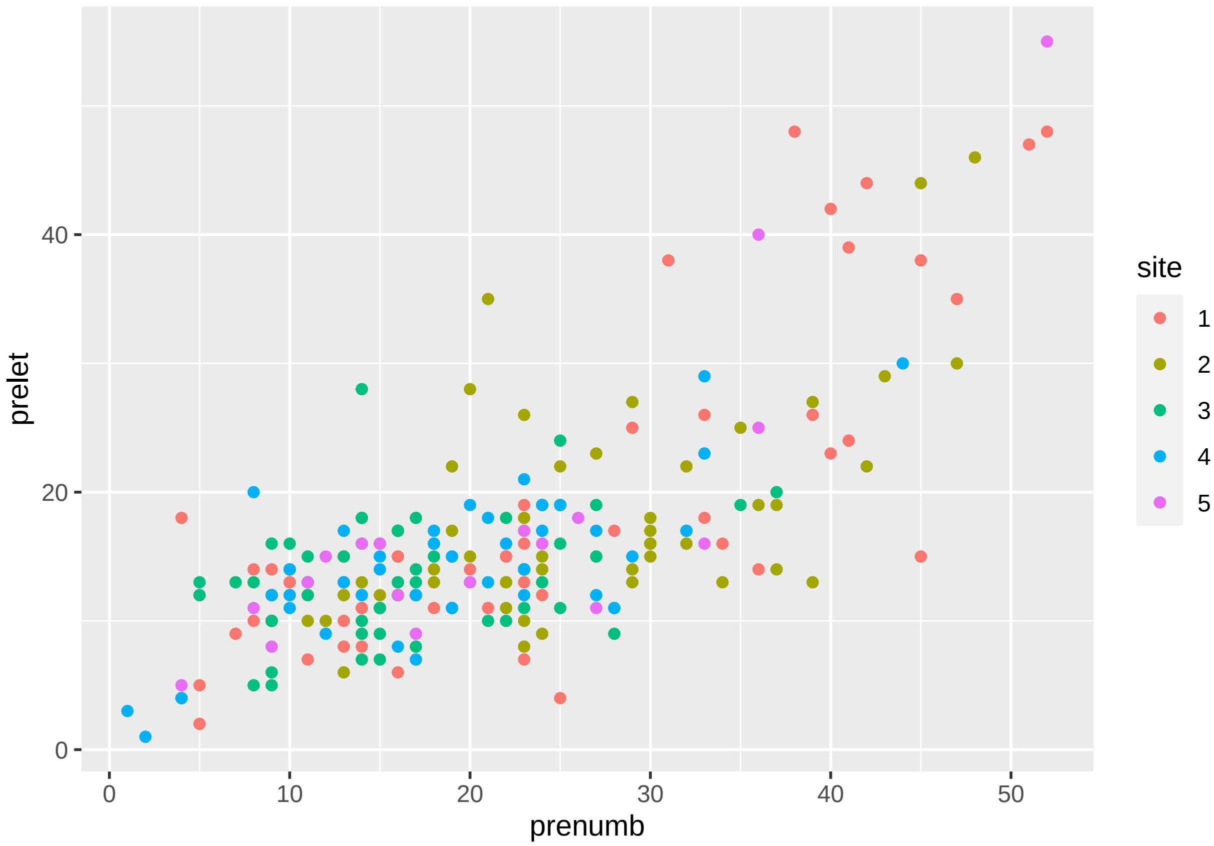
Should I look for interactions? Is LASSO necessary here?

Q.2 Classification Question: Can we use the pre-test scores and other demographic variables to predict which region the children came from?

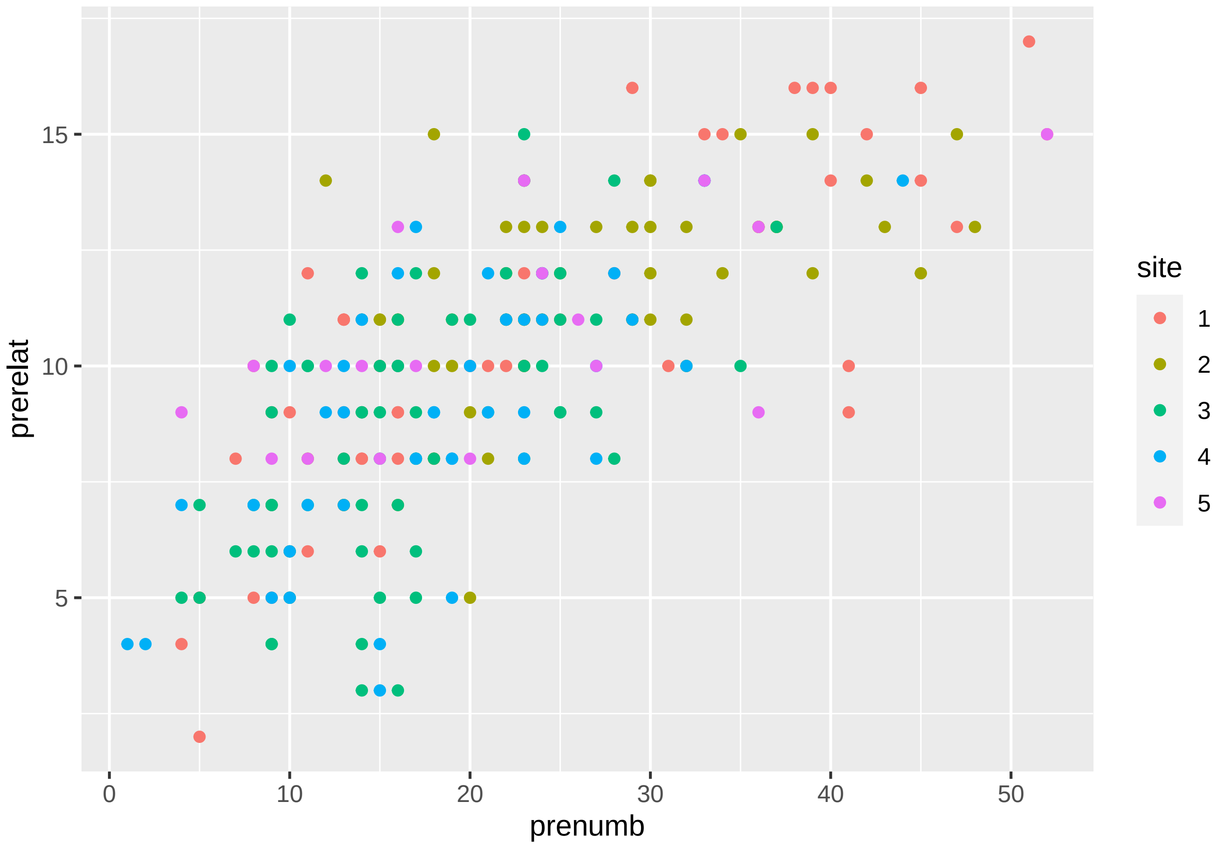
```
# "viewcat", "setting", "viewenc",
pca.features <- c("age", "prebody", "prelet","preform", "prenumb", "prerelat", "preclasf", "peabody")

pca.data <- sesame[, pca.features]

round(cor(pca.data),2)
```

```
##           age prebody prelet preform prenumb prerelat preclasf peabody
## age      1.00    0.33   0.33    0.28    0.43     0.44     0.36    0.25
## prebody  0.33    1.00   0.45    0.68    0.70     0.62     0.61    0.54
## prelet   0.33    0.45   1.00    0.51    0.72     0.47     0.49    0.37
## preform  0.28    0.68   0.51    1.00    0.67     0.60     0.71    0.53
## prenumb  0.43    0.70   0.72    0.67    1.00     0.72     0.71    0.58
## prerelat 0.44    0.62   0.47    0.60    0.72     1.00     0.64    0.52
## preclasf 0.36    0.61   0.49    0.71    0.71     0.64     1.00    0.45
## peabody  0.25    0.54   0.37    0.53    0.58     0.52     0.45    1.00
```

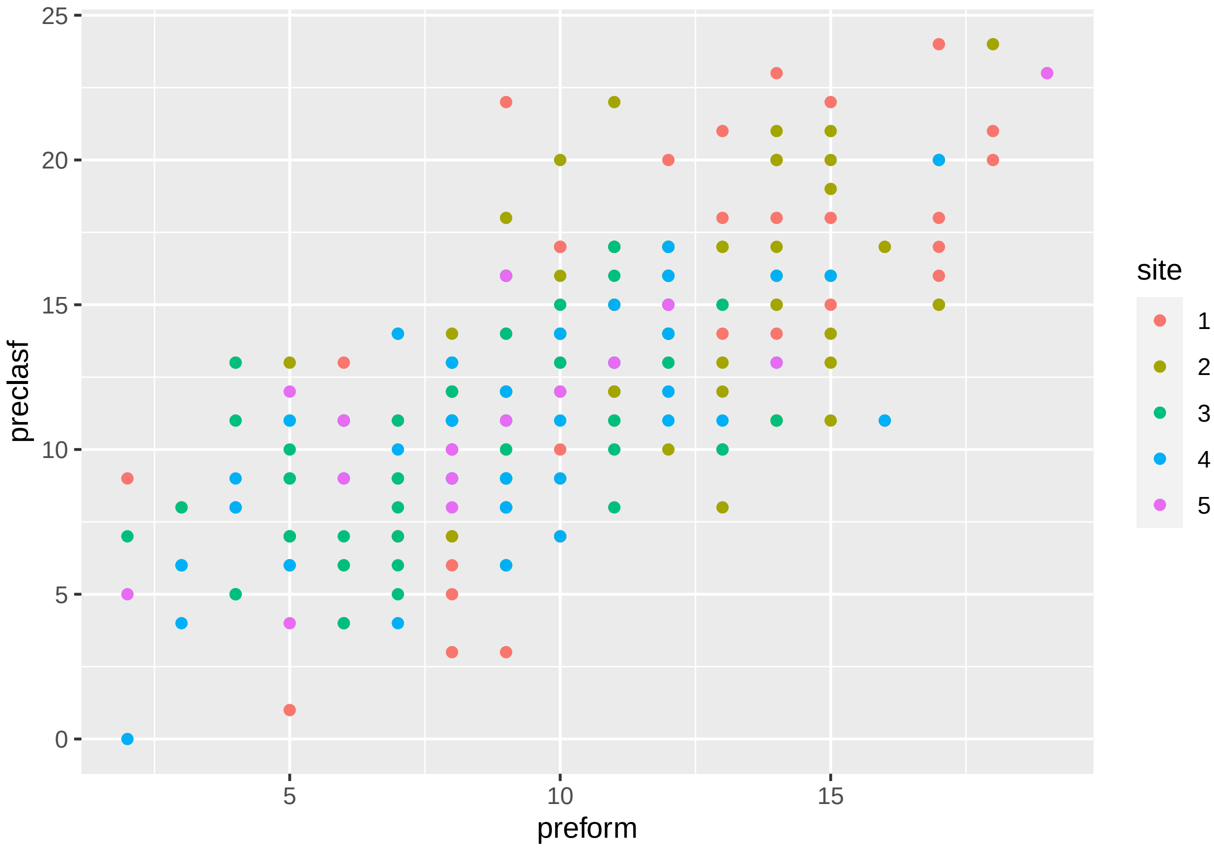
```
ggplot(data=sesame, aes(x=prenumb, y=prelet, color=site))+
  geom_point()
```



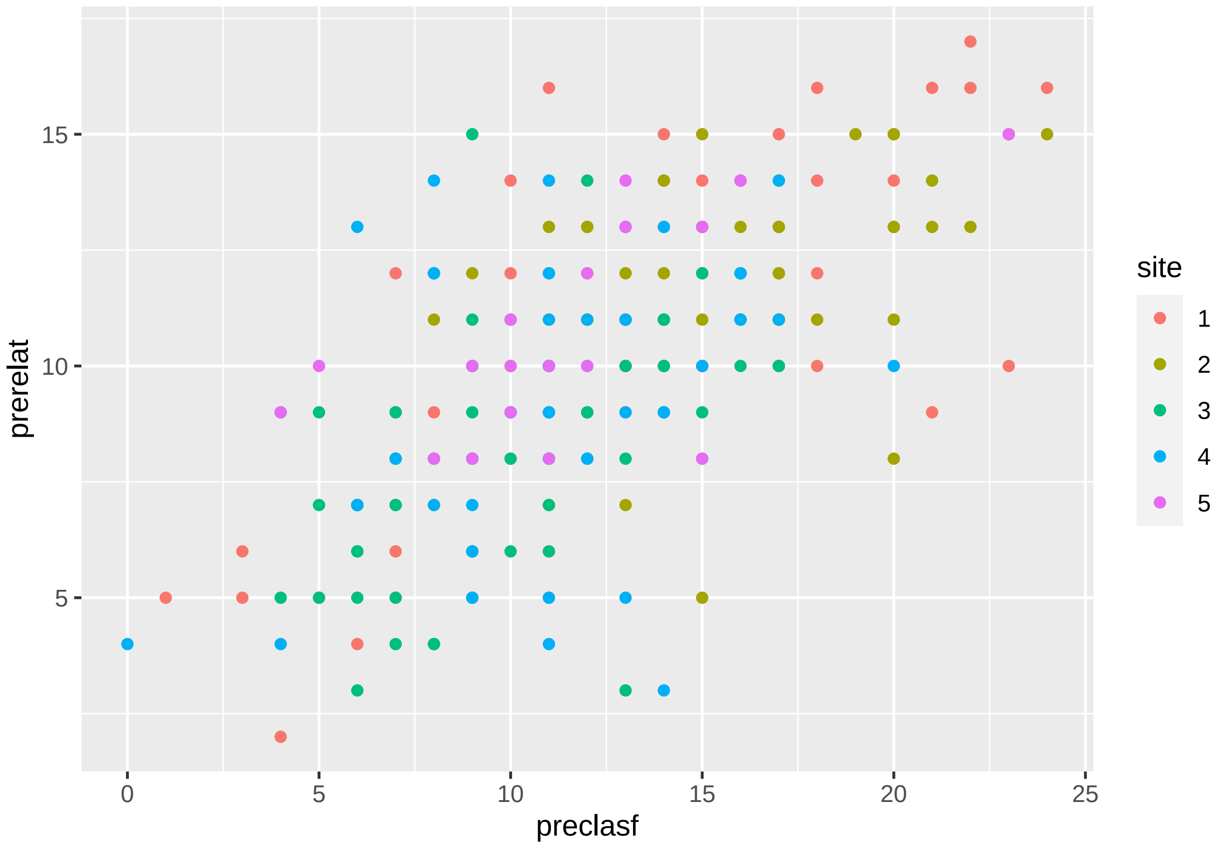

```
ggplot(data=sesame, aes(x=prenumb, y=prerelat, color=site))+
  geom_point()
```




```
ggplot(data=sesame, aes(x=preform, y=preclasf, color=site))+
  geom_point()
```




```
ggplot(data=sesame, aes(x=preclasf, y=prerelat, color=site))+
  geom_point()
```



SVM

```
set.seed(3241)

n <- nrow(sesame)
train.index <- sample(1:n, size = floor(0.7*n), replace=FALSE)
train.data <- sesame.sd[train.index,]
test.data <- sesame.sd[-train.index,]

train.data %>%
  count(site)
```

```
##   site  n
## 1    1 40
## 2    2 42
## 3    3 48
## 4    4 25
## 5    5 13
```

```
# Response: site (categorical)
set.seed(315)
costs <- c(0.001, 0.01, 0.1, 1, 5, 10, 100)
gammas <- c(0.1, 0.5, 1, 2, 3, 4)

linear.tune <- tune(svm, site~female+ male + sd_age+sd_pBod+sd_plet+sd_pform + sd_pnumb+sd_prelat+sd_pcl
                    data=train.data, kernel="linear",
                    ranges=list(cost=costs))
```

```r
radial.tune <- tune(svm, site~female + male + sd_age+sd_pBod+sd_plet+sd_pform + sd_pnumb+sd_prelat+sd_p
                    data=train.data, kernel="radial",
                    ranges=list(cost=costs,
                                gamma=gammas))
#radial.tune <- tune(svm, site~sex+age+prebody+prelet+preform+prenumb+prerelat+preclasf,
#                    data=train.data, kernel="radial",
#                    ranges=list(cost=costs,
#                                gamma=gammas))
```

```r
linear.conMatrix <- table(true=test.data[, "site"],
                          pred=predict(linear.tune$best.model, newdata=test.data))

radial.conMatrix <- table(true=test.data[, "site"],
                          pred=predict(radial.tune$best.model, newdata=test.data))

confusionMatrix(linear.conMatrix)
```

```
## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1  2  5 13  0  0
##    2  0  8  5  0  0
##    3  1  1 14  0  0
##    4  0  4 14  0  0
##    5  0  1  4  0  0
##
## Overall Statistics
##
##                Accuracy : 0.3333
##                  95% CI : (0.2266, 0.4543)
##     No Information Rate : 0.6944
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1523
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.66667   0.4211   0.2800       NA       NA
## Specificity           0.73913   0.9057   0.9091     0.75  0.93056
## Pos Pred Value        0.10000   0.6154   0.8750       NA       NA
## Neg Pred Value        0.98077   0.8136   0.3571       NA       NA
## Prevalence            0.04167   0.2639   0.6944     0.00  0.00000
## Detection Rate        0.02778   0.1111   0.1944     0.00  0.00000
## Detection Prevalence  0.27778   0.1806   0.2222     0.25  0.06944
## Balanced Accuracy     0.70290   0.6634   0.5945       NA       NA
```

```r
confusionMatrix(radial.conMatrix)
```

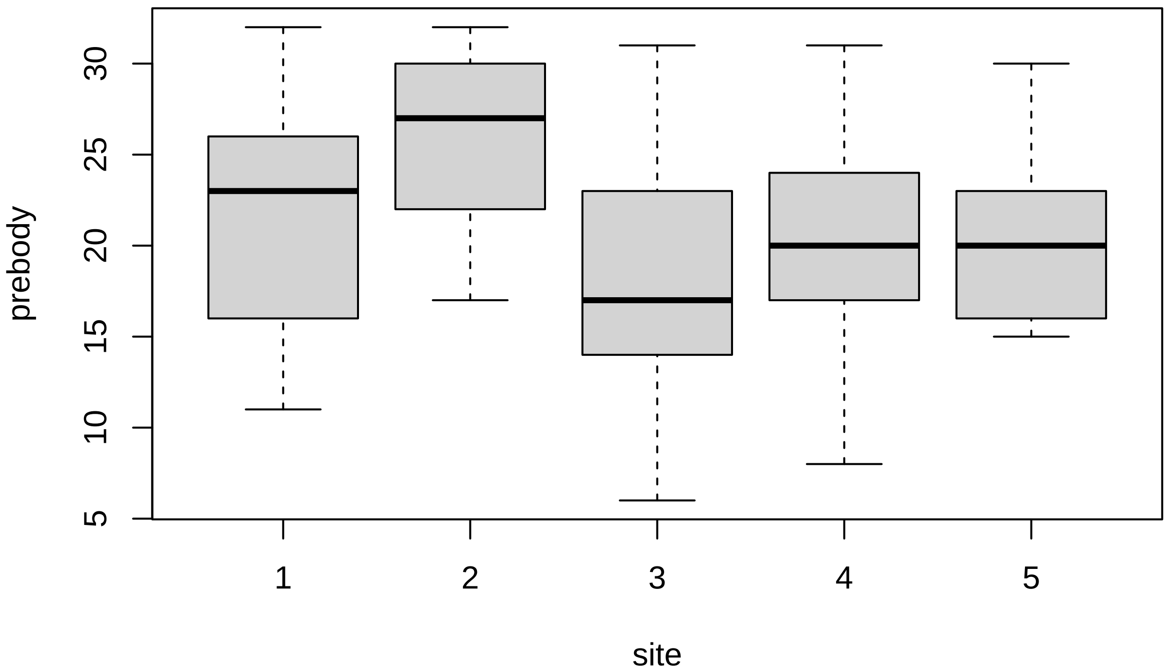
```
## Confusion Matrix and Statistics
##
```

```
##     pred
## true  1  2  3  4  5
##    1  7  3 10  0  0
##    2  1  8  4  0  0
##    3  1  1 14  0  0
##    4  1  3 14  0  0
##    5  0  1  4  0  0
##
## Overall Statistics
##
##                Accuracy : 0.4028
##                  95% CI : (0.2888, 0.525)
##     No Information Rate : 0.6389
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2337
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity           0.70000   0.5000   0.3043       NA       NA
## Specificity           0.79032   0.9107   0.9231     0.75  0.93056
## Pos Pred Value        0.35000   0.6154   0.8750       NA       NA
## Neg Pred Value        0.94231   0.8644   0.4286       NA       NA
## Prevalence            0.13889   0.2222   0.6389     0.00  0.00000
## Detection Rate        0.09722   0.1111   0.1944     0.00  0.00000
## Detection Prevalence  0.27778   0.1806   0.2222     0.25  0.06944
## Balanced Accuracy     0.74516   0.7054   0.6137       NA       NA
```

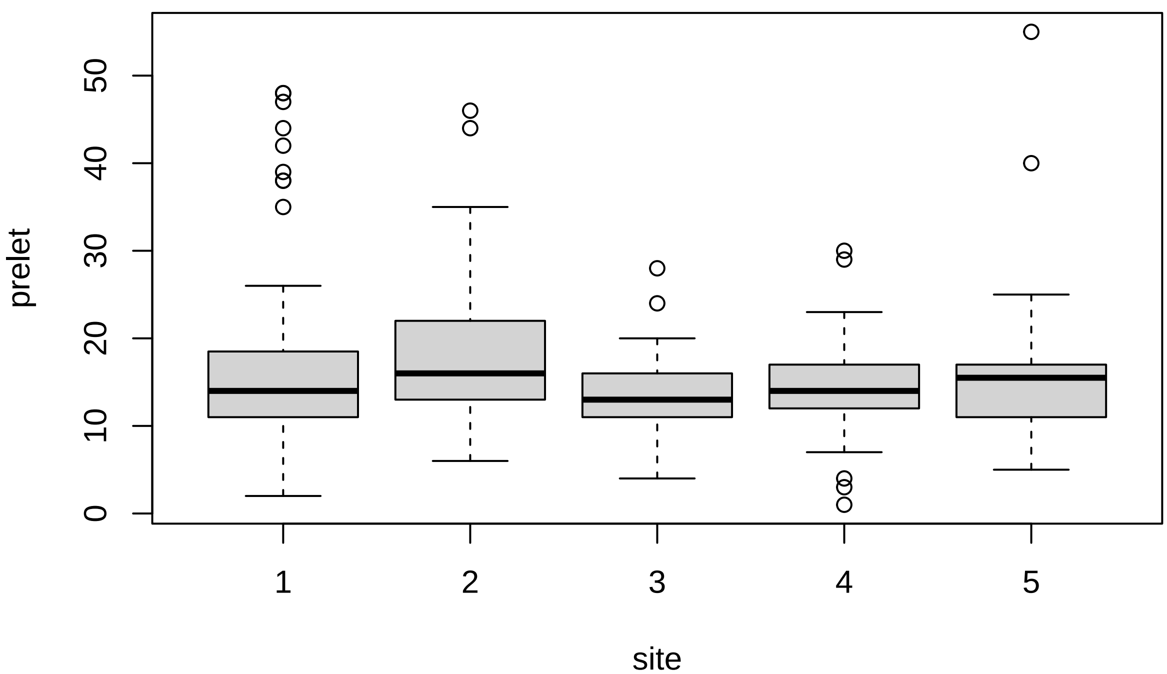
Radial kernel improves prediction on class 1.

RBF slightly improved after standardizing? (it seems slightly more likely to predict on class 1.) thought, simpler models still retain the same performance (arguably better) sd_age+sd_pBod+sd_plet. But we are still not getting any prediction on class 4 & 5.

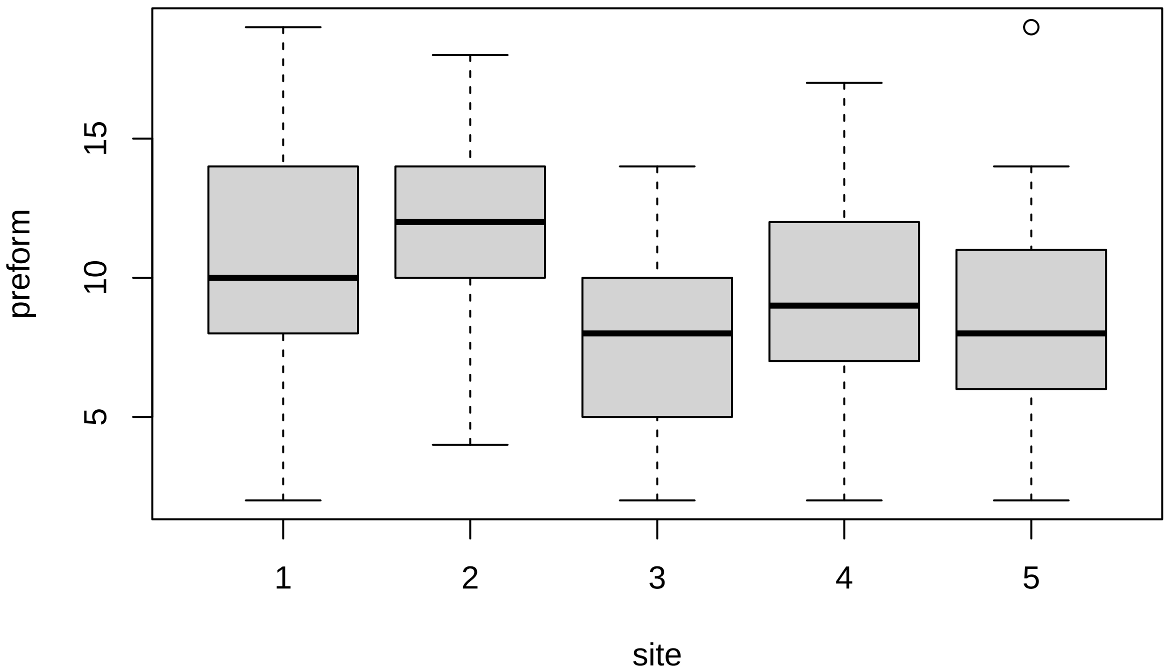
```r
# trying to do more EDA to see if anything explains why the data is not linearly separable
boxplot(prebody~site, data=sesame)
```



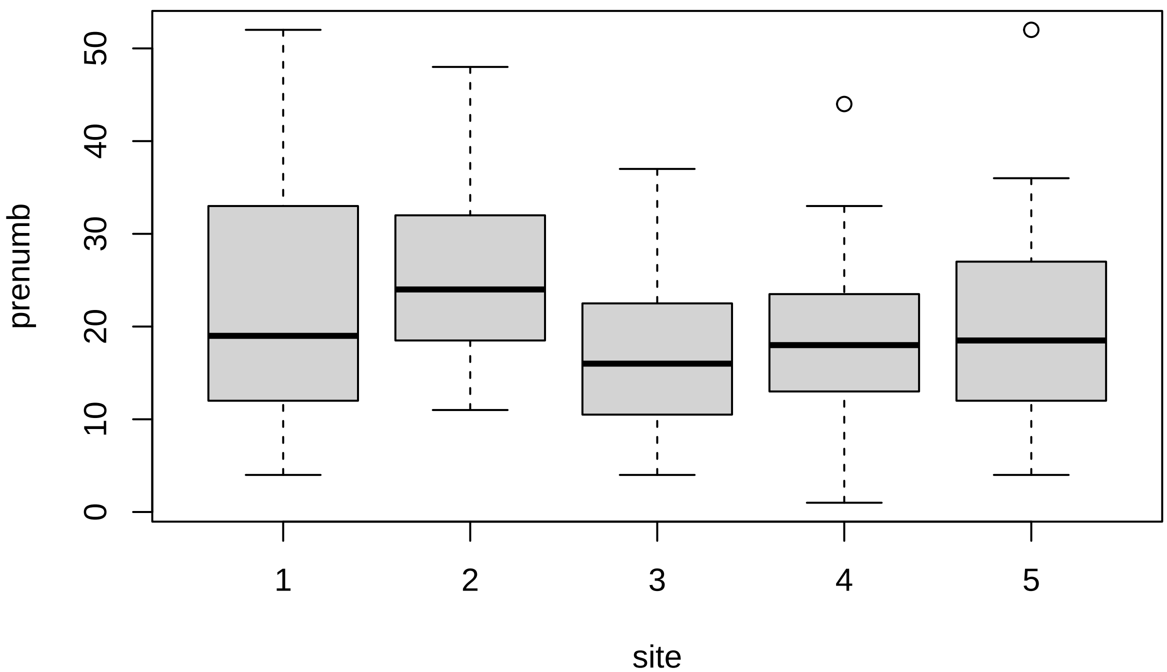

```
boxplot(prelet~site, data=sesame)
```



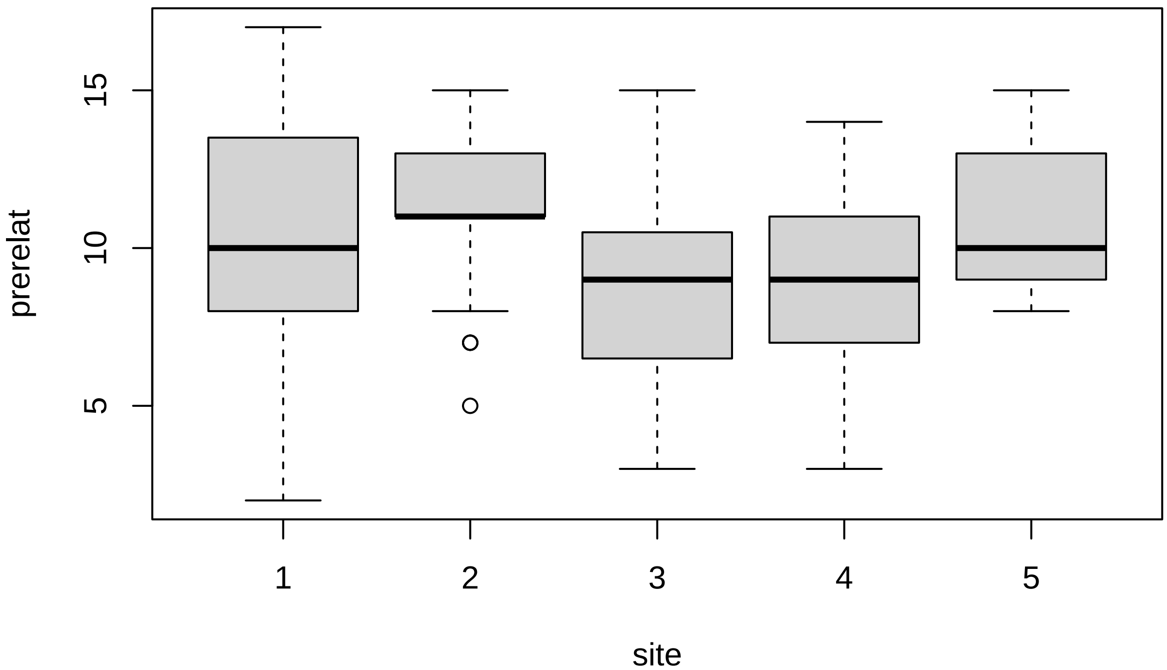

```
boxplot(preform~site, data=sesame)
```



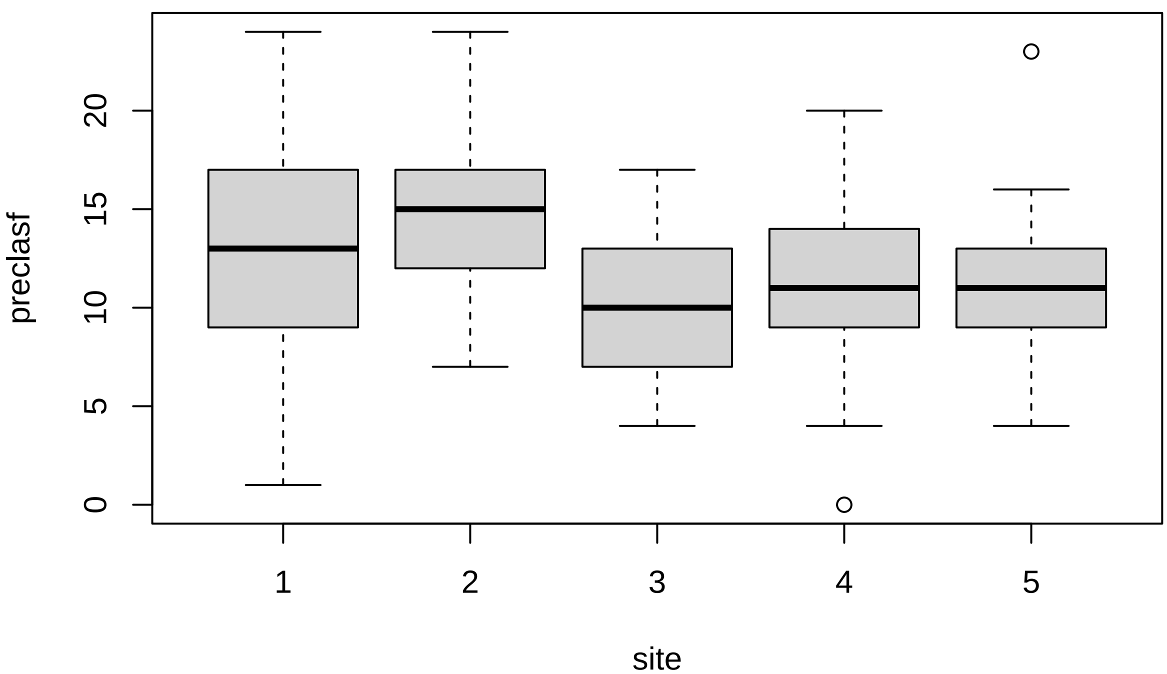

```
boxplot(prenumb~site, data=sesame)
```



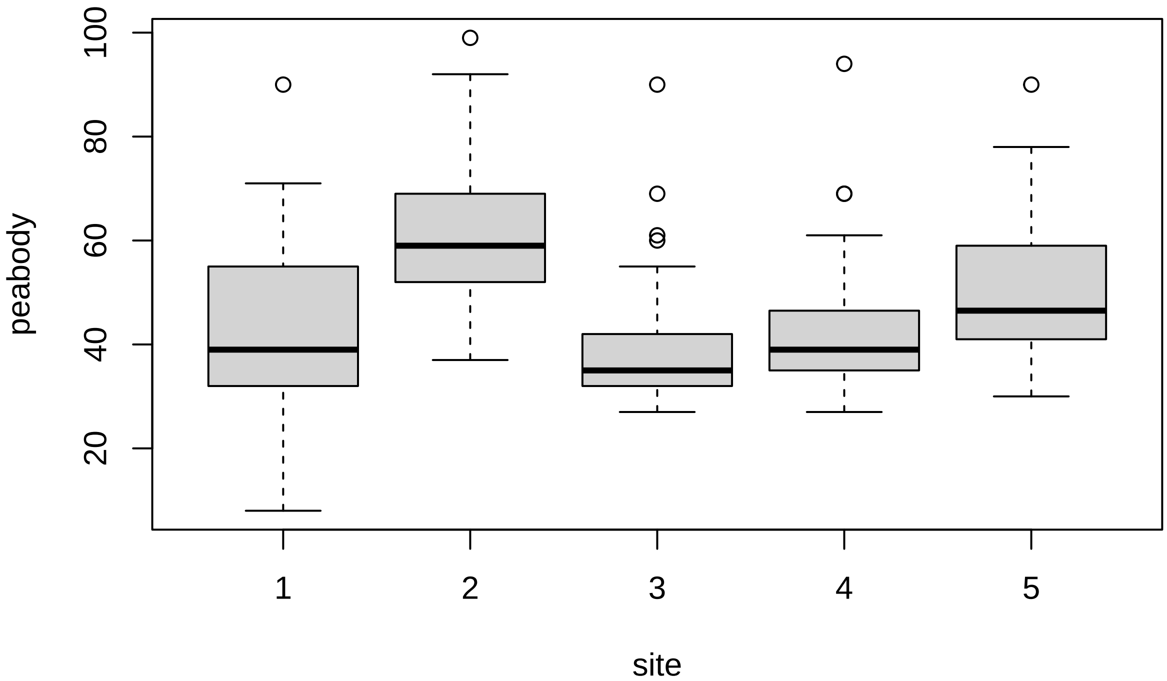

```
boxplot(prerelat~site, data=sesame)
```




```
boxplot(preclasf~site, data=sesame)
```




```
boxplot(peabody~site, data=sesame)
```



Tree

```
set.seed(3215)
#tree.data <- sesame %>%
#  select(site, sex, age, viewcat, setting, viewenc, prebody, prelet, preform,
#         prenumb, prerelat, preclasf)

n <- nrow(sesame)
train.index <- sample(1:n, size = floor(0.7*n), replace=FALSE)
#train.tree <- tree.data[train.index,]
#test.tree <- tree.data[-train.index,]
# "viewcat", "setting", "viewenc",

# ,"prebody", "prelet","preform", "prenumb", "prerelat", "preclasf", "postbody", "postlet", "postform",



tree.features <- c("site", "age", "viewcat", "setting", "viewenc")

tree.data <- sesame[, tree.features]
train.data <- tree.data[train.index,]
test.data <- tree.data[-train.index,]
```

```
rf.tree<- randomForest(site~., data=tree.data, subset=train.index,
                       mtry=4, importance=TRUE)

importance(rf.tree)
```

```
##                 1          2         3         4          5
## age      15.361234 10.9448858  2.925774  9.065429 -14.246704
## viewcat   1.280978  8.9523401  6.751433 28.181155  -7.719696
## setting   9.091424  0.4941378 15.005652 21.214488   6.474409
```

```
## viewenc  2.694681 -1.2473386  9.354781 -1.810026  -1.360029
##         MeanDecreaseAccuracy MeanDecreaseGini
## age                14.775615        69.088432
## viewcat            20.389357        24.047981
## setting            23.037944        10.510319
## viewenc             4.870229         9.706542
```

```
rf.pred <- predict(rf.tree, newdata=test.data)

tree.conMatrix <- table(true=test.data[,"site"],
                        pred=rf.pred)
confusionMatrix(tree.conMatrix)
```

```
## Confusion Matrix and Statistics
##
##     pred
## true  1  2  3  4  5
##    1 12  2  2  2  0
##    2  4 10  2  1  1
##    3  3  3 10  3  0
##    4  0  3  4  5  0
##    5  0  2  2  0  1
##
## Overall Statistics
##
##                Accuracy : 0.5278
##                  95% CI : (0.4065, 0.6467)
##     No Information Rate : 0.2778
##     P-Value [Acc > NIR] : 6.693e-06
##
##                   Kappa : 0.3818
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: 1 Class: 2 Class: 3 Class: 4 Class: 5
## Sensitivity            0.6316   0.5000   0.5000  0.45455  0.50000
## Specificity            0.8868   0.8462   0.8269  0.88525  0.94286
## Pos Pred Value         0.6667   0.5556   0.5263  0.41667  0.20000
## Neg Pred Value         0.8704   0.8148   0.8113  0.90000  0.98507
## Prevalence             0.2639   0.2778   0.2778  0.15278  0.02778
## Detection Rate         0.1667   0.1389   0.1389  0.06944  0.01389
## Detection Prevalence   0.2500   0.2500   0.2639  0.16667  0.06944
## Balanced Accuracy      0.7592   0.6731   0.6635  0.66990  0.72143
```

0.42 – 0.5139 (but not including the test scores.) around 0.45-0.48, when including the pretest scores.

```
set.seed(231)

# ,"prebody", "prelet","preform", "prenumb", "prerelat", "preclasf", "postbody", "postlet", "postform",



features <- c("site", "age", "viewcat", "setting", "viewenc","prebody", "prelet","preform", "prenumb",
```

```r
tree.2 <- sesame[, features]
train.2 <- tree.2[train.index,]
test.2 <- tree.2[-train.index,]

boost.tree <- gbm(site ~., data=train.2,
                  distribution="multinomial", n.trees=5000,
                  interaction.depth=1)
```

```
## Warning: Setting `distribution = "multinomial"` is ill-advised as it is
## currently broken. It exists only for backwards compatibility. Use at your own
## risk.
```

```r
#y.boost <- table(true=test.2[,"site"],
#                  pred=predict(boost.tree, newdata=test.2))

predict(boost.tree, newdata=test.2)
```

```
## Using 5000 trees...
```

```
## , , 5000
##
##                1           2          3          4            5
##  [1,]  -8.1952329 -29.66492200 -30.823109  -6.159639 -28.519908906
##  [2,]   5.9117380 -43.30486839 -44.515974 -18.781318 -63.907880072
##  [3,] -13.8116433 -19.62157783 -50.394689 -15.431370 -68.130271635
##  [4,]  -5.2692418  -9.38578847 -40.339478 -20.017594 -65.504158602
##  [5,] -31.5295008 -19.25544326 -14.178134 -11.186330 -34.925814153
##  [6,] -37.7719707 -19.88532985   2.191577 -16.118636 -26.860418877
##  [7,] -27.7560719 -25.75826109  -7.383736 -36.320656 -30.526055144
##  [8,] -13.4398988 -26.17883094  18.592292 -37.767866  -4.761008339
##  [9,] -11.4004289 -26.09495321 -13.436709 -26.986970 -45.302516798
## [10,]  -4.8909642 -31.00049704 -14.359102  -2.722996 -19.917290796
## [11,] -13.7469583 -31.87528328  -8.096223 -26.373236 -32.937352830
## [12,] -30.7907310 -65.72964409 -11.994372  -3.391687  -8.945469341
## [13,]  -2.7097144 -15.14430119 -34.704778 -16.901320 -31.972187904
## [14,] -13.8839416 -53.49139985 -18.502627 -40.317287 -31.425893038
## [15,]  33.0017562   5.00712390 -36.842539 -21.477056   6.840913772
## [16,] -13.0382686 -41.87790085  -6.002752 -38.412195 -61.997841279
## [17,] -34.2456569 -12.61017993 -27.193645 -29.348113 -77.112080265
## [18,]   7.0287873 -75.42928786 -32.532306  -7.162624 -53.057008348
## [19,] -15.3888925  11.39225919 -42.567271 -25.921845 -32.525327562
## [20,]   6.6655743  11.23618216 -29.034291 -20.473846  -2.545188189
## [21,]   2.6690615  26.76602803 -37.829753 -31.079086 -46.123932330
## [22,]   0.5123770  16.69425460 -30.451113  -8.156147 -19.022962327
## [23,]  25.0086461  14.07902985 -28.977962 -24.317620 -18.586801292
## [24,] -37.3456440 -10.88154842 -19.448303 -39.556008 -51.247962553
## [25,]  19.5249769  16.17962400 -24.507002 -35.461170   9.232720125
## [26,] -20.2607386  14.59589373 -25.099760 -32.881096 -17.748949753
## [27,]   8.9326274   3.81728993 -29.093353 -10.628209  -6.184490974
## [28,] -31.6316898 -20.96563023 -29.102346 -18.888210 -54.619424281
## [29,]  -0.9490962  30.09730245 -37.357209 -18.609416 -25.947130662
## [30,]  13.4728978  30.93625680 -36.142968  -8.637653  -2.506784206
## [31,] -11.6096409  13.57094279 -36.421557 -34.749302 -35.623192583
## [32,]  -7.0429867  13.40399682  -8.553608 -23.094038   4.837052864
## [33,] -30.3529503  -0.08402483 -37.307595 -25.950072 -36.523823393
```

```
## [34,] -23.0654228 -15.74761758 -16.773517 -37.521742 -60.372329925
## [35,] -12.0834836 -37.61682411 -39.078561 -28.971401 -55.434000385
## [36,] -43.2144144 -33.33813054 -13.408052  11.274825  18.935822256
## [37,] -23.3296594 -54.11322784  -8.807964 -22.033211 -26.803377483
## [38,]  16.7990195 -32.98040152 -28.706776 -32.055610 -30.652983324
## [39,] -10.1843878 -25.23886428 -17.290074  -6.964105 -45.415534092
## [40,]  -5.3531675 -37.54280471 -30.029063 -21.593225 -53.848375096
## [41,] -24.0843625 -63.17673579   3.881288 -29.564692 -29.939718758
## [42,]   0.3458327 -36.60306676  25.451814 -22.894842 -32.894013490
## [43,] -10.2455432 -34.79791723   3.231962   2.359530  -9.157799356
## [44,] -23.8825701 -57.03346196  10.283467  -1.547150 -43.588742447
## [45,] -30.8076731 -55.84598116  -8.054185 -19.829615 -11.851121878
## [46,]   0.2757117 -64.28437932  12.895311 -31.161451 -39.157327775
## [47,] -34.4626003 -18.65252390 -25.157912 -22.713331 -22.736448821
## [48,] -38.4117434 -28.37880909 -33.090735 -14.318930 -43.916161718
## [49,] -36.6597165 -40.13572423 -16.785609 -23.247200 -28.686587117
## [50,] -20.3642456 -38.30757648 -16.398607 -22.350194 -54.835414184
## [51,] -18.1488304 -17.71129748 -26.711381 -52.621258 -45.362875346
## [52,] -13.0218032 -55.07315025  -6.261209 -15.410716 -28.088687023
## [53,] -17.9703561 -46.22153970  -4.655640 -29.329822 -52.921416135
## [54,]  -8.4617626 -37.56214140  13.875190 -23.629669   2.850809140
## [55,]  -1.2992544 -35.83777670 -33.363554 -27.595703 -60.413253915
## [56,] -14.5648309 -51.47224111 -21.279513  -6.309059 -49.702597359
## [57,] -34.7565336 -36.28500814 -30.676484 -21.938083 -74.269197250
## [58,] -20.7392149 -41.79720870 -15.963637 -32.844227 -54.122278746
## [59,] -17.7155502 -32.83029908   7.572634 -11.266941 -27.472331214
## [60,] -35.1789919 -11.87669007  -1.443067  -8.414123 -44.858247307
## [61,] -34.7629353 -30.70456430 -15.454565  -9.820691 -32.872453359
## [62,] -27.4516200 -44.14670273  17.590038 -25.172281 -38.090110011
## [63,] -35.6123424 -10.84866549   1.397036  -4.549489  26.604847143
## [64,] -28.6266460 -21.28830761 -16.166194 -21.692056 -44.297384926
## [65,] -14.6605232 -26.50090006 -25.574690 -27.680894 -49.951529966
## [66,] -29.5437958 -27.59546902  -2.192588 -31.551892 -25.136284071
## [67,] -17.5393922  -7.66670156 -11.527736 -29.676825   7.443543574
## [68,] -46.6187671 -30.41301201  -9.526684 -12.107706 -46.632202876
## [69,] -26.1537917 -45.23154780 -12.175565 -20.789468  -0.007776294
## [70,] -41.9255244   3.83640079 -36.124446 -11.830554 -60.599948174
## [71,] -40.1790751 -14.22947548 -14.648835 -47.035491 -45.310830570
## [72,]   0.7001622 -35.59599617 -17.984758 -12.967685   0.229916822
```

Questions for OH:

anything else needed in EDA?

Both linear and radial kernels never output predictions for 4 & 5?

polynomial kernel? Which variables to give polynomial terms

use PCA to perform feature selection?

feature selections for SVM in general?

how to interpret the confusion matrix tables for SVM & Trees

How to interpret the imporatnce variance for multiclass classification

interpretations about the dataset, using the bad performance of the classifiers