Final_Project

Jason Dai, Andrew Zhao, Yunfan Long
All teammates contributed equally in this homework
Math 189

2023-04-23

Application Questions

Question 1

```
library(ISLR2)
data("Carseats")
```

(a)

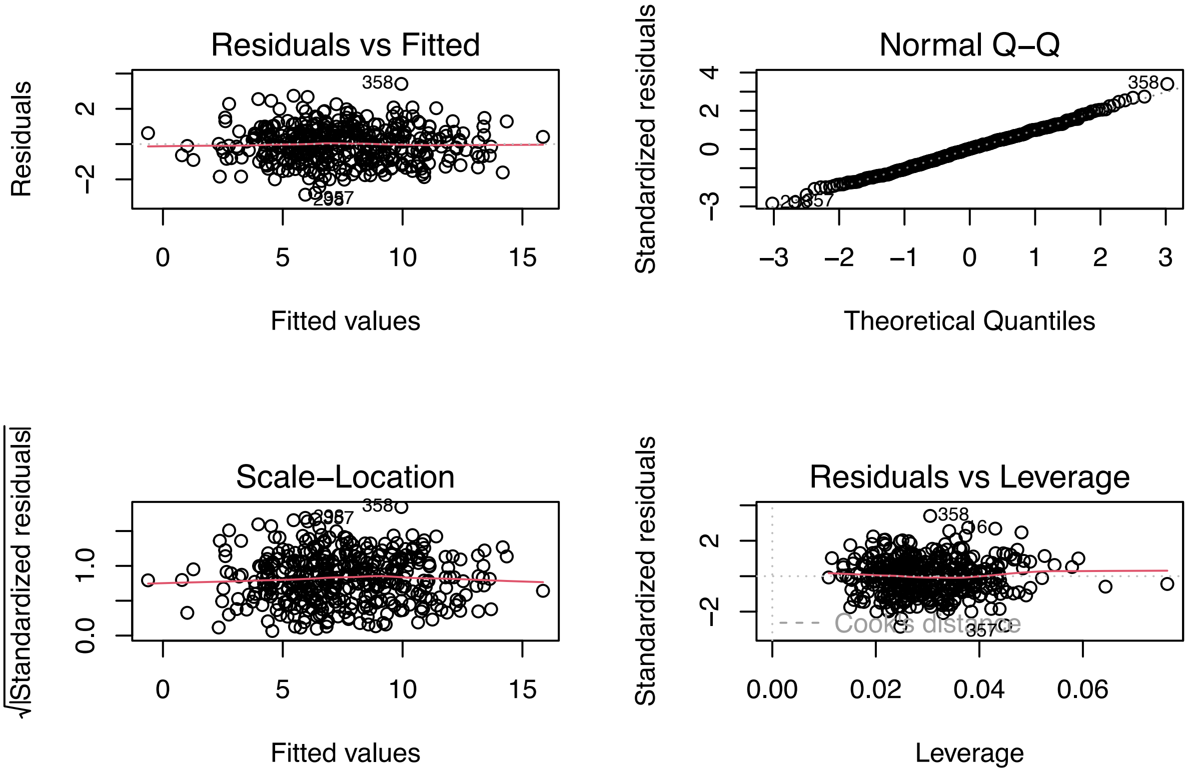
```
response <- Carseats$Sales
Carseats_new <- Carseats[, -which(names(Carseats) == "Sales")]
lm.fit <- lm(response ~., data = Carseats_new)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = response ~ ., data = Carseats_new)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8692 -0.6908  0.0211  0.6636  3.4115
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.6606231  0.6034487   9.380  < 2e-16 ***
## CompPrice        0.0928153  0.0041477  22.378  < 2e-16 ***
## Income           0.0158028  0.0018451   8.565 2.58e-16 ***
## Advertising      0.1230951  0.0111237  11.066  < 2e-16 ***
## Population       0.0002079  0.0003705   0.561    0.575
## Price           -0.0953579  0.0026711 -35.700  < 2e-16 ***
## ShelveLocGood    4.8501827  0.1531100  31.678  < 2e-16 ***
## ShelveLocMedium  1.9567148  0.1261056  15.516  < 2e-16 ***
## Age             -0.0460452  0.0031817 -14.472  < 2e-16 ***
## Education       -0.0211018  0.0197205  -1.070    0.285
## UrbanYes         0.1228864  0.1129761   1.088    0.277
## USYes           -0.1840928  0.1498423  -1.229    0.220
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.019 on 388 degrees of freedom
## Multiple R-squared:  0.8734, Adjusted R-squared:  0.8698
## F-statistic: 243.4 on 11 and 388 DF,  p-value: < 2.2e-16
```

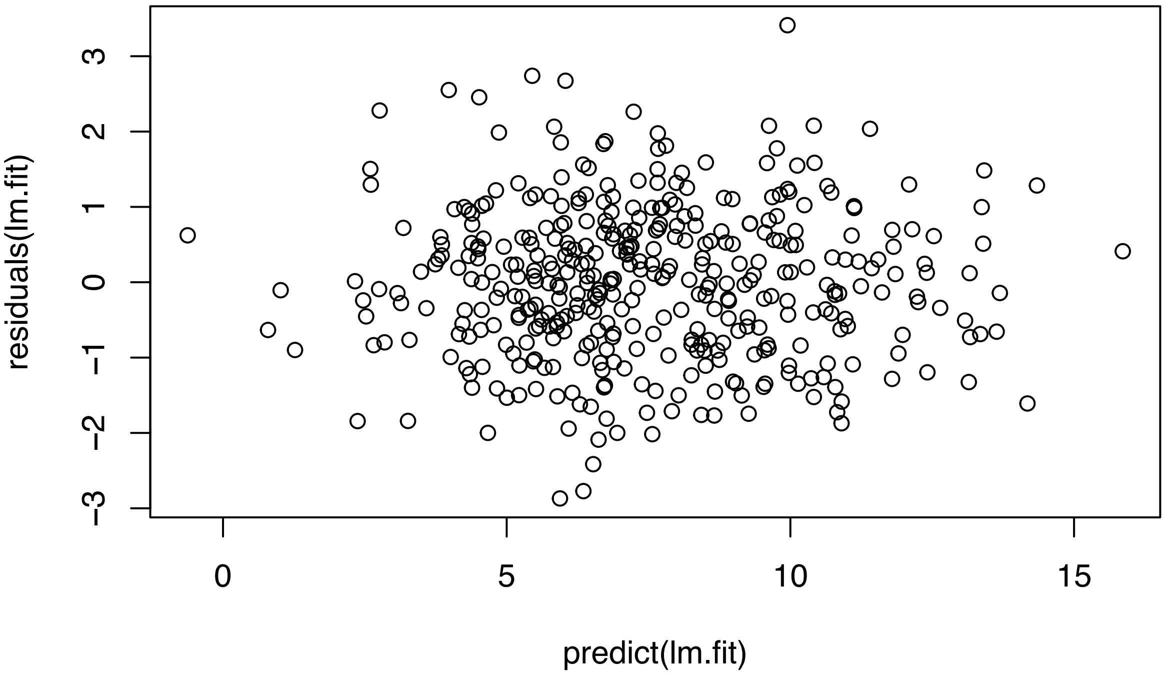
(b)

```
par(mfrow = c(2, 2))
plot(lm.fit)
```



According to the graph we don't have a pattern in residual, we have a constant variance in residual plot which implies homoscedasticity. There is also Uncorrelated error and according to the Q-Q plot normality also holds. Thus we conclude a linear model is appropriate.

```
plot(predict(lm.fit), residuals(lm.fit))
```



According to the graph, our linear model's prediction is close to the actual sales values and there's not an obvious shape in the residual plot so we condlude a linear model is appropriate.

(c)

Null hypothesis: beta1 =0 and beta2 =0 Since the Pr(>|t|) for `CompPrice` and `Income` are lower than 0.05, we conclude that the hypothesis doesn't hold.

Question 2

(a)

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```
set.seed((7))
train_idx <- sample(seq_len(nrow(Carseats)), size = 0.8 * nrow(Carseats))
train_data <- Carseats[train_idx, ]
test_data1 <- Carseats[-train_idx, ]
x_train <- model.matrix(~+ShelveLoc+Urban+US+CompPrice+Income+Advertising+Population+Price+Age+Educa
y_train <- train_data$Sales
test_data <- model.matrix(~Sales+ShelveLoc+Urban+US+CompPrice+Income+Advertising+Population+Price+Ag
head(test_data)
```

```
##    Sales ShelveLocGood ShelveLocMedium UrbanYes USYes CompPrice Income
## 4   7.40             0               1        1     1       117    100
## 11  9.01             0               0        0     1       121     78
## 12 11.96             1               0        1     1       117     94
## 13  3.98             0               1        1     0       122     35
## 19 13.91             1               0        0     1       110    110
## 20  8.73             0               1        1     1       129     76
##    Advertising Population Price Age Education
## 4            4        466    97  55         14
## 11           9        150   100  26         10
## 12           4        503    94  50         13
## 13           2        393   136  62         18
## 19           0        408    68  46         17
## 20          16         58   121  69         12
```

(b)

```r
set.seed(7)
cv_model <- cv.glmnet(x_train, y_train, alpha = 0, nfolds = 5)
lambda_optimal <- cv_model$lambda.min
ridge_model <- glmnet(x_train, y_train, alpha = 0, lambda = lambda_optimal)
coefficients <- coef(ridge_model)
print(coefficients)
```

```
## 13 x 1 sparse Matrix of class "dgCMatrix"
##                              s0
## (Intercept)      6.161090e+00
## (Intercept)      .
## ShelveLocGood    4.386789e+00
## ShelveLocMedium  1.783305e+00
## UrbanYes         1.122254e-01
## USYes            2.838470e-02
## CompPrice        8.281132e-02
## Income           1.264387e-02
## Advertising      1.157427e-01
## Population       1.229511e-05
## Price           -8.660647e-02
## Age             -4.460182e-02
## Education       -1.616828e-02
```

(c)

```r
predicted_values <- predict(ridge_model, newx = test_data)
rmse <- sqrt(mean((predicted_values - test_data[, "Sales"])^2))
rmse
```

```
## [1] 1.024566
```

(d)

```r
library(randomForest)
```

```
## randomForest 4.7-1.1

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

```r
rf_model <- randomForest(Sales ~ ., data = train_data, ntree = 50)
predicted_rf_values <- predict(rf_model, newdata = test_data1)
rmse_rf <- sqrt(mean((predicted_rf_values - test_data1$Sales)^2))
rmse_rf
```

```
## [1] 1.420962
```

(e)

As the ridge_model and the random forest model have different role that the market team need to make decisions, the team may prefer to use the random forest if they want to maximize the accuracy of their predictions, also, if the market team want to understand which predictor has what kind of impact on prediction, then they may want to use the ridge regression model.

Question 3

(a)

```
set.seed(1)
n <- 200
X <- rt(n, df=15)
head(X)
```

```
## [1] -0.6266918 -0.6645957  0.4036624 -0.3053975 -0.2458638  0.4572033
```

(b)

```
epsilon <- rt(n, df=5)
head(epsilon)
```

```
## [1] -0.6646696 -0.6458288  1.1965794 -2.1261656 -0.1906402  1.5191501
```

(c)

```
Y <- 5 + 2*sin(X) - 7*exp(2*cos(X))/(1+exp(2*cos(X))) + epsilon
head(Y)
```

```
## [1] -2.6811185 -2.6781131  0.9421316 -3.8226409 -1.7977821  1.3996148
```

(d)

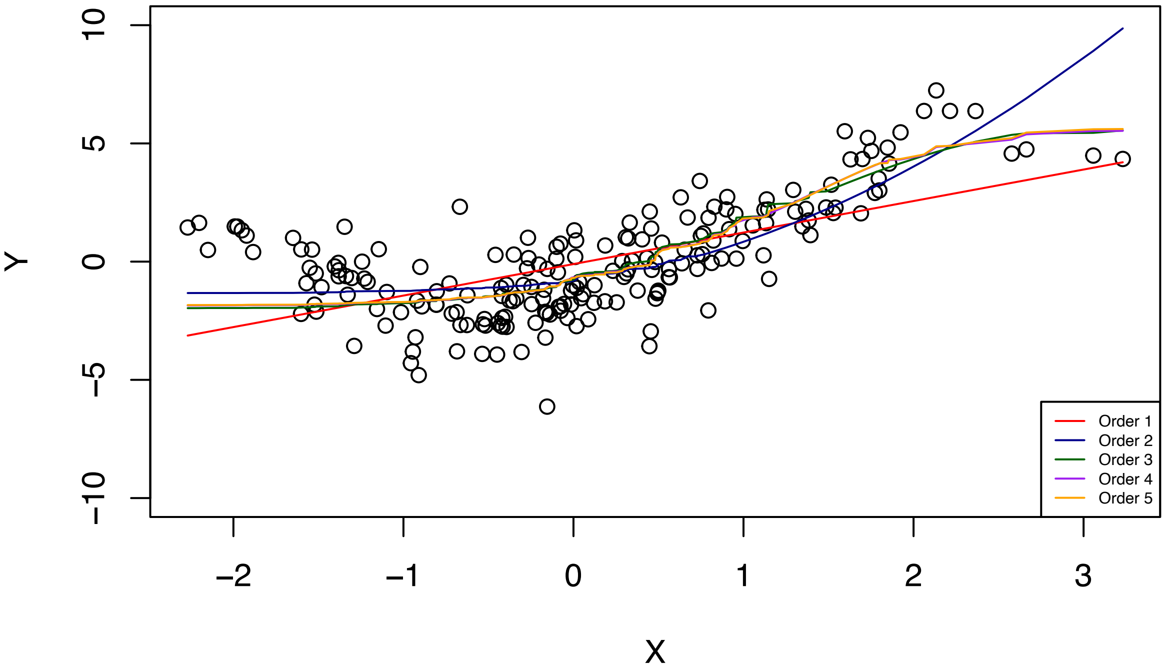
```
df <- data.frame(X, Y)

colors <- c('red', 'darkblue', 'darkgreen', 'purple', 'orange')
legends <- character()

plot(X, Y, ylim = c(-10, 10))

for (i in 1:5) {
    fit <- lm(Y ~ poly(X, i), data=df)
    preds <- predict(fit, df)
    lines(sort(X), sort(preds), col=colors[i])
    legends <- c(legends, paste('Order', i))
}

legend('bottomright', legend=legends, col=colors, lty=1, cex = 0.5)
```



(e)

I would prefer the order 2 model because it captures the general trend of the data without being overfitting to the specific orginal dataset. The order 1 model fails to capture the curvature of the data and thus is too simple for modeling. However, model with order 3, 4, and 5 are overfiting to the original dataset, as we can see some zig-zag parttern, meaning it's overfitting to the trend specific to this dataset. Therefore, I would prefer order 2 model because it fits the shape of the data and is not prone to overfitting.

(f)

```
fit_lsq <- lm(Y ~ poly(X, 2), data=df)
predict(fit_lsq, newdata=data.frame(X=1), interval="confidence", level=0.9)
```

```
##         fit       lwr      upr
## 1 0.8402292 0.6037489 1.076709
```

As shown above, the 90% confidence for prediction using least squares theory is (0.6037489, 1.076709). This means we are 90% confident that the true mean value of the response variable Y would fall into this interval.

(g)

```
library(boot)

boot_func <- function(data, indices) {
  df <- data[indices, ]
  fit <- lm(Y ~ poly(X, 2), df)
  return(predict(fit, newdata=data.frame(X=1)))
}

boot_out <- boot(df, boot_func, 1000)


boot.ci(boot_out, type="basic", conf=0.90)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = boot_out, conf = 0.9, type = "basic")
##
## Intervals :
## Level      Basic
## 90%   ( 0.5882,  1.0672 )
## Calculations and Intervals on Original Scale
```

As shown above, the 90% confidence for prediction using bootstrap is (0.5882, 1.0672). This means we are 90% confident that the true mean value of the response variable Y would fall into this interval. Unlike the least square model, the bootstrap model avoids making assumptions about the data and residuals because the calculation is essentially a process of sampling 1000 times with replacement, fitting model, and calculate prediction value of y at X = 1. Since we let go of some assumptions, the resulting interval is comparatively wider as a trade-off.

Question 4

(a)

```
train_idx <- sample(seq_len(nrow(College)), size = 0.8 * nrow(College))
train_data_college <- College[train_idx, ]
test_data_college <- College[-train_idx, ]
```

(b)

```
logit_model = glm(Private ~ ., data = train_data_college, family = "binomial")
coefficients <- coef(logit_model)
coefficients
```

```
##   (Intercept)          Apps        Accept        Enroll     Top10perc
## -5.235308e-02 -3.537293e-04  4.582441e-04  1.489866e-03 -1.274239e-02
##     Top25perc   F.Undergrad   P.Undergrad      Outstate    Room.Board
##  2.120343e-02 -9.307853e-04  2.454161e-04  7.688134e-04  2.983314e-05
##         Books      Personal           PhD      Terminal     S.F.Ratio
##  1.628969e-03 -1.414257e-04 -5.136300e-02 -3.789128e-02 -7.117294e-02
##   perc.alumni        Expend     Grad.Rate
##  2.857813e-02  1.548533e-04  1.455310e-02
```

The interpretation of the coefficient of `Top10perc` is that there is a `1.907536e-02` log-odds or probability increase that the college is a private college.

(c)

```
predicted_labels <- predict(logit_model, newdata = test_data_college, type = "response")
class_preds <- predicted_labels > 0.5
table <- table(test_data_college$Private, class_preds)
1 - sum(diag(table)) / sum(table)
```

```
## [1] 0.06410256
```

(d)

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:ISLR2':
##
##     Boston
```

```
model_lda <- lda(Private ~ ., data = train_data_college)
predictions_lda <- predict(model_lda, newdata = test_data_college)$class
table(predictions_lda, test_data_college$Private)
```

```
##
## predictions_lda  No Yes
##             No   37   4
##             Yes   6 109
```

```
1 - mean(predictions_lda == test_data_college$Private)
```

```
## [1] 0.06410256
```

(e)

```
library(MASS)
model_qda <- qda(Private ~ ., data = train_data_college)
predictions_qda <- predict(model_qda, newdata = test_data_college)$class
table(predictions_qda, test_data_college$Private)
```

```
##
## predictions_qda  No Yes
##             No   34   6
##             Yes   9 107
```

```
1 - mean(predictions_qda == test_data_college$Private)
```

```
## [1] 0.09615385
```

(f)

```
library(e1071)
svmfit <- svm(Private ~ ., data = train_data_college , kernel = "linear", cost = 0.1)
ypred <- predict(svmfit, test_data_college)
table(predict=ypred, truth  = test_data_college$Private)
```

```
##        truth
## predict  No Yes
##     No   37   6
##     Yes   6 107
```

```
1 - mean(ypred == test_data_college$Private)
```

```
## [1] 0.07692308
```

(g)

I will choose LDA because it has the best performance on the test set with the lowest test error

Question 5

(a)

```
Sys.setenv("RGL_USE_NULL"="TRUE")
library(MultBiplotR)
```

```
##
## Attaching package: 'MultBiplotR'

## The following object is masked from 'package:MASS':
##
##     ginv

## The following object is masked from 'package:boot':
##
##     logit
```

```
data("Protein")

protein <- Protein[, !(names(Protein) %in% c('Comunist', 'Region'))]

pca <- prcomp(protein, scale = TRUE)

prop_var <- pca$sdev^2 / sum(pca$sdev^2)
print(prop_var[1:5])
```

```
## [1] 0.4451597 0.1816666 0.1253244 0.1060738 0.0515376
```

```
cum_prop_var <- cumsum(prop_var)
print(cum_prop_var[1:5])
```

```
## [1] 0.4451597 0.6268263 0.7521507 0.8582245 0.9097621
```

(b)

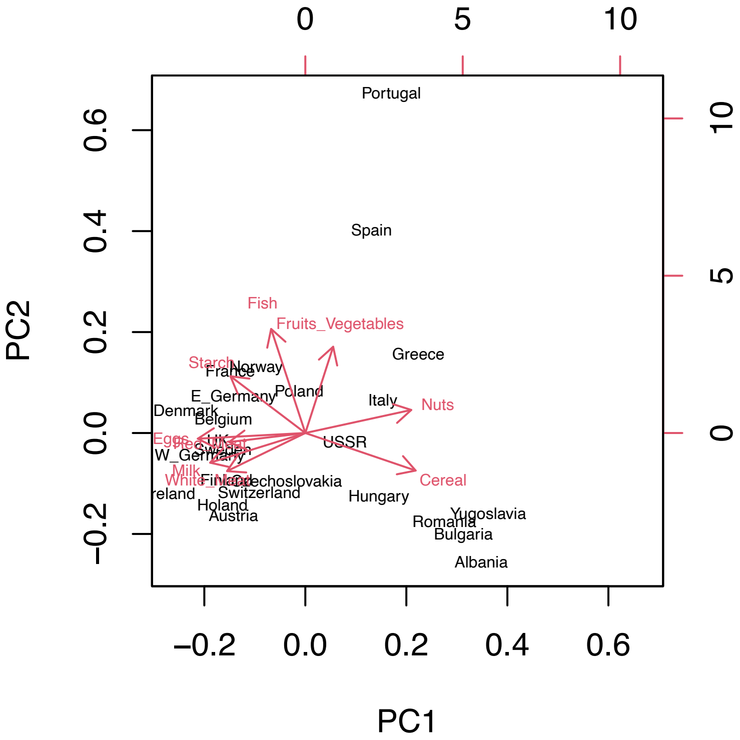
```
pca$rotation[,1:2]
```

```
##                          PC1         PC2
## Red_Meat          -0.3026094 -0.05625165
## White_Meat        -0.3105562 -0.23685334
## Eggs              -0.4266785 -0.03533576
## Milk              -0.3777273 -0.18458877
## Fish              -0.1356499  0.64681970
## Cereal             0.4377434 -0.23348508
## Starch            -0.2972477  0.35282564
## Nuts               0.4203344  0.14331056
## Fruits_Vegetables  0.1104199  0.53619004
```

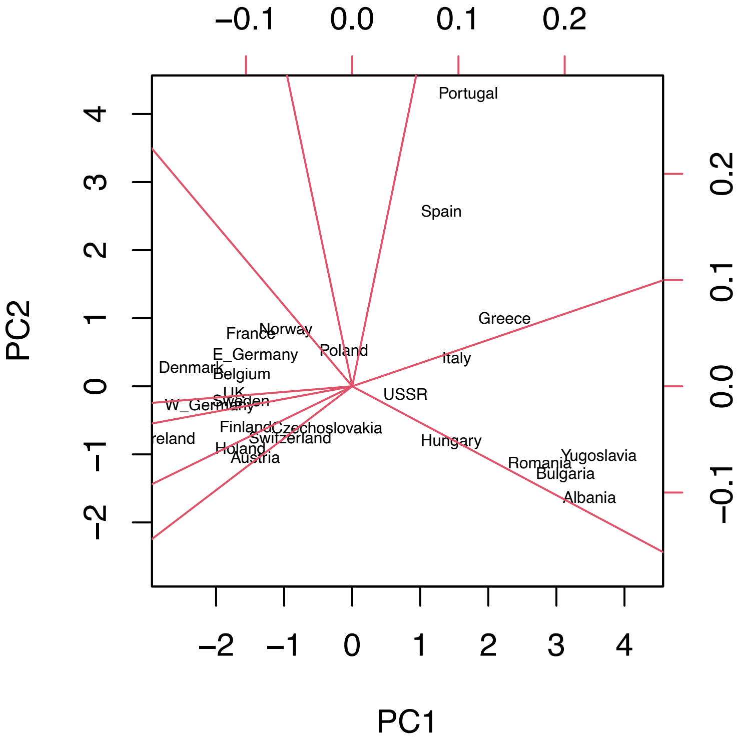
We observe that weights for `Nuts`, `Eggs` and `Cereal` are high in PC1. Therefore, if a country has a higher score for PC1 then it represents that the country prefers those. We can also observe that the weights for `Fish` and `Fruits_Vegetables` are high in PC2. If a country has a higher score of PC2 then it represents that the country prefer those particular items.

(c)

```
biplot(pca, cex = 0.5)
```



```
biplot(pca, cex = 0.5, scale = 0, expand = 2.5)
```



From the biplot above, it seems that milk is: (i) most positively correlated with white_meat (ii) most negatively correlated with nuts (iii) uncorrelated with fish

(d)

It seems like the Central and North both has relatively higher consumption on white meat, milk, eggs and red meat and both have relatively lower consumption of cereal and nuts. Central has relatively higher consumption of starch, fish and fruits then North.

Question 6

It may be more beneficial because bootstrapping creates multiple subsets by random sampling. Since random forest is built by combining multiple decision trees, bootstrapping introduces diversity. The second reason is that the bootstrapping strategy can help random forest handle high-dimensional data by training different subsets. The last reason is that the bootstrapping strategy can make random forest more robust to outliers since the data are splitted into different subsets and the effect if outliers will dramatically decrease.

Question 7

Since `FWER` and `FDR` is used to avoid type I error, and as we know that if we put the type I error to a very low level, the level of type II error will increase, so we would not want to have the type II error value increase a lot when making type I error is that a medicine is incorrectly justified as effective and was used on some patient, and the type II error is the medicine is incorrectly justified as ineffective, while it is actually effective, this would delay treatment time of some patients and that is more severe, so we would not want to correct `FWER` and `FDR` in this case.

Question 8

It is necessary because if the assumptions are not met, there would be no guarantee of the validity of the inferences or predictions we make. This could lead to situations where the results are wrong, misleading, or cannot be interpreted properly.