Final

Chance Kang / A13605546 / csk025@ucsd.edu
Jayden Kim / A16271107 / s0k003@ucsd.edu
Sia Sheth / A16357789 / snsheth@ucsd.edu
Math 189
Spring 2023

```
library(ISLR2)
library(glmnet)
```

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

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

```
library(MASS)
```

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

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

```
library(randomForest)
```

```
## randomForest 4.7-1.1
```

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

```
library(tree)
library(caret)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

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

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

```
library(e1071)
library(car)
```

```
## Loading required package: carData
```

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

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

```
## The following object is masked from 'package:lattice':
##
##     melanoma
```

Application Question

1.

```
data(Carseats)
car <- Carseats
```

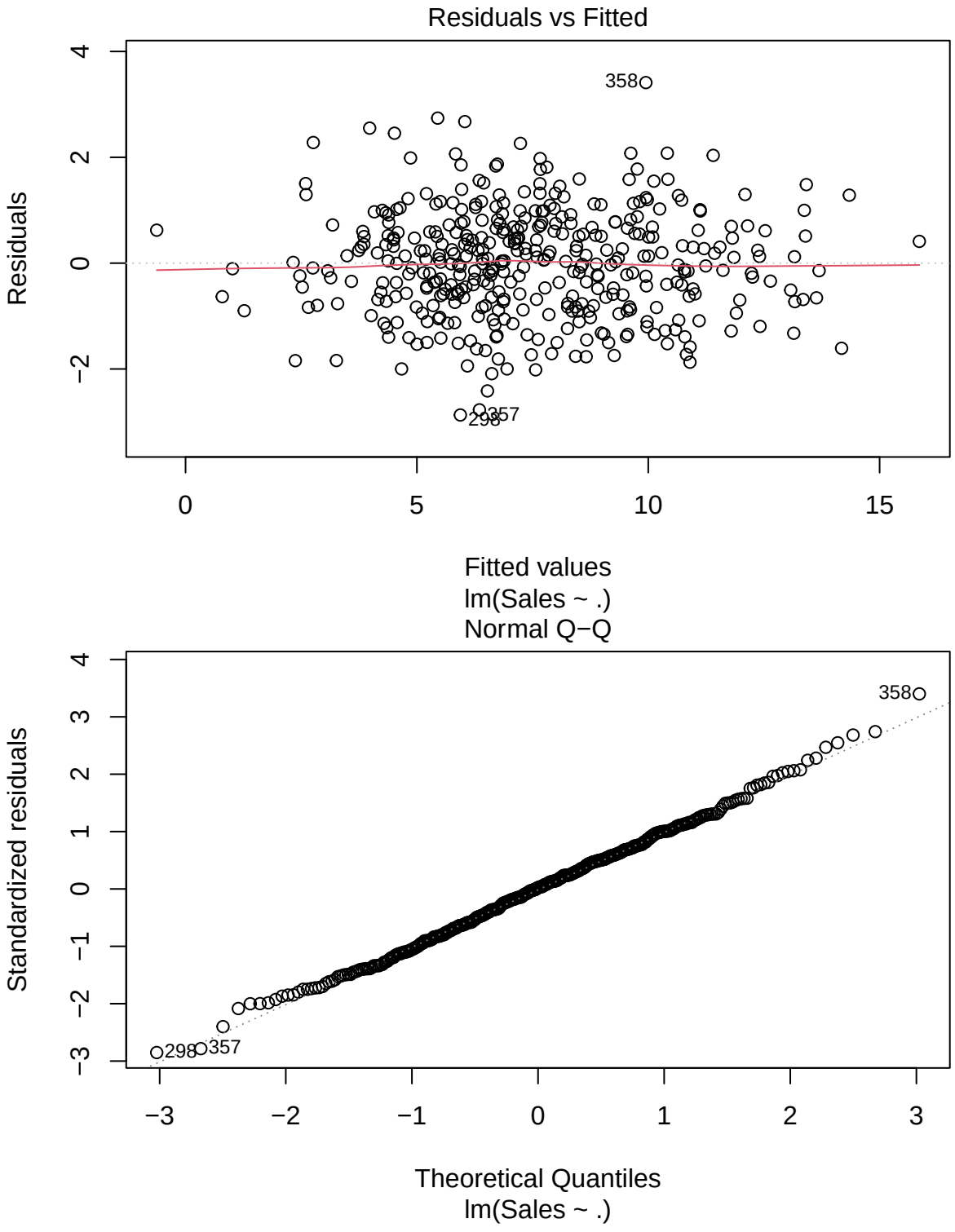
a.

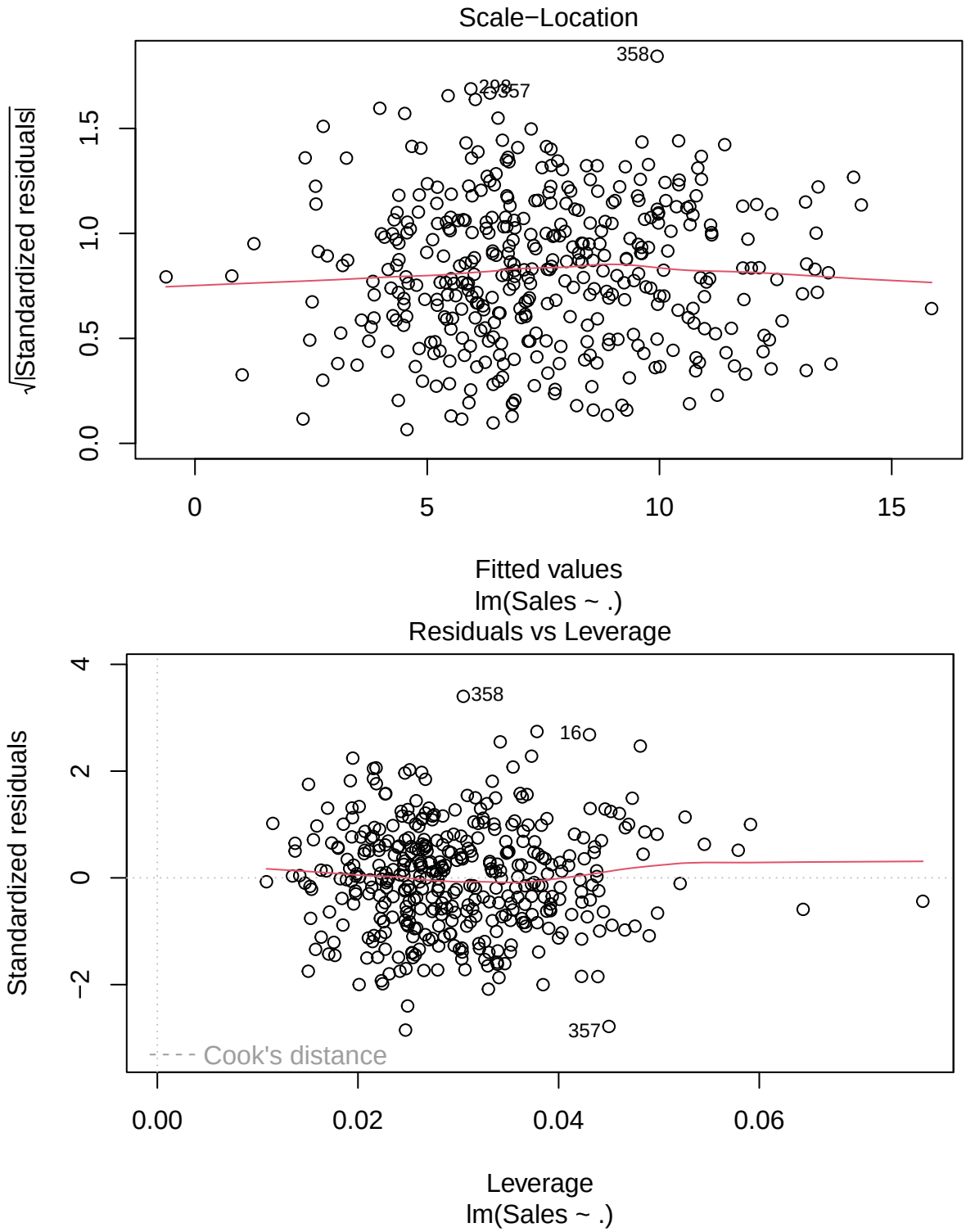
```
lm_cs <- lm(Sales ~ ., data = car)
summary(lm_cs)$coefficients
```

```
##                     Estimate   Std. Error     t value      Pr(>|t|)
## (Intercept)      5.6606230631 0.6034486581   9.3804551  5.596251e-19
## CompPrice        0.0928153421 0.0041476529  22.3777990  7.935340e-72
## Income           0.0158028363 0.0018451176   8.5646772  2.579912e-16
## Advertising      0.1230950886 0.0111236855  11.0660346  6.353734e-25
## Population       0.0002078771 0.0003704559   0.5611385  5.750270e-01
## Price           -0.0953579188 0.0026710774 -35.7001707 1.175168e-124
## ShelveLocGood    4.8501827110 0.1531099670  31.6777725 1.192737e-109
## ShelveLocMedium  1.9567148062 0.1261056428  15.5164730  1.383807e-42
## Age             -0.0460451630 0.0031817142 -14.4718098  2.924395e-38
## Education       -0.0211018389 0.0197204930  -1.0700462  2.852637e-01
## UrbanYes         0.1228863965 0.1129760904   1.0877204  2.773938e-01
## USYes           -0.1840928246 0.1498422926  -1.2285772  2.199750e-01
```

b.

```
plot(lm_cs)
```


Residuals vs Fitted
Residuals
Fitted values
lm(Sales ~ .)
Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Sales ~ .)
358
357
298



Residuals vs Fitted shows linearity and seems to show homoscedasticity. The points are well on the fitted line of the QQ plot so normality holds as well.

c.

$\beta_1 =$ `CompPrice` $\beta_2 =$ `Income`

$$\mathcal{H}_0 : \beta_1 = \beta_2 = 0$$

$$\mathcal{H}_1 : \beta_i \neq 0, \text{ for at least one value of i} = 1, 2$$

$\mathcal{H}_0$ states there is no relationship between (`CompPrice`, `Income`), and `Sales`

```
lm_cs_sub <- lm(Sales ~ CompPrice + Income, data = car)
summary(lm_cs_sub)
```

```
##
## Call:
## lm(formula = Sales ~ CompPrice + Income, data = car)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.2679 -1.9705 -0.0498  1.7491  8.6970
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.632191   1.226865   3.776 0.000184 ***
## CompPrice   0.014150   0.009138   1.548 0.122299
## Income      0.015959   0.005007   3.187 0.001550 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.79 on 397 degrees of freedom
## Multiple R-squared:  0.02895,    Adjusted R-squared:  0.02406
## F-statistic: 5.919 on 2 and 397 DF,  p-value: 0.002931
```

```
linearHypothesis(lm_cs_sub, c("CompPrice = 0", "Income = 0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## CompPrice = 0
## Income = 0
##
## Model 1: restricted model
## Model 2: Sales ~ CompPrice + Income
##
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    399 3182.3
## 2    397 3090.1  2     92.14 5.9188 0.002931 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Choosing $\alpha = 0.05$ and test static as f-statistic, hence we reject null hypothesis if f-statistic value of `linearHypothesis` matches model's reported f-statistic value, and corresponding p-value is < 0.05. Above rejects null hypothesis. Therefore, at least one of `CompPrice` and `Income` are statistically significant in refitted model when predicting `Sales`.

2.

a. Train 80: Validation 20

```
set.seed(123)
train <- sample(1:nrow(car), nrow(car)*0.8)
```

```
val <- (-train)
car.train <- car[train,]
car.val <- car[val,]
x<-model.matrix(Sales~., data = car)[,-1]
y<-car$Sales
```

b. `glmnet` defaults `nfolds = 10`, K = 10.

```
ridge.mod<-glmnet(x[train,], y[train], alpha = 0)
set.seed(123)
cv.out <- cv.glmnet(x[train,], y[train], alpha=0)
bestlam<-cv.out$lambda.min
print(bestlam)
```

```
## [1] 0.1442638
```

```
coef(cv.out)
```

```
## 12 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)      6.7979688729
## CompPrice        0.0756187432
## Income           0.0138252304
## Advertising      0.1061651330
## Population       0.0001263579
## Price           -0.0814120891
## ShelveLocGood    4.2415861929
## ShelveLocMedium  1.5457528547
## Age             -0.0449834939
## Education       -0.0294476950
## UrbanYes         0.0409576824
## USYes           -0.0155305942
```

c.

```
ridge.pred <- predict(ridge.mod, s = bestlam ,newx = x[val,])
sqrt(mean((ridge.pred- y[val])^2))
```

```
## [1] 1.065929
```

```
sd(car$Sales)
```

```
## [1] 2.824115
```

RMSE being far below the standard deviation of `Sales` suggests ridge regression performed well.

d.

```
set.seed(123)
rf <- randomForest(Sales ~ ., data = car, subset = train, mtry = 9, importance = TRUE)
rf.pred <- predict(rf, newdata = car.val)
sqrt(mean((rf.pred - car.val$Sales)^2))
```

```
## [1] 1.442598
```

e. A marketing team which advertises for this specific dataset may prefer ridge regression over random forest for its RMSE was lower, suggesting better performance. Ridge regression is also relatively simpler model than random forest, hence easier to interpret and faster to train. A marketing team of TikTok, for example, may prefer to use random forest, for the data regarding user data may form a complex relationship and the company can handle the cost of training large data.

3.

a.

```
set.seed(123)
x <- rt(200,15)
```

b.

```
set.seed(123)
e <- rt(200,5)
```

c.

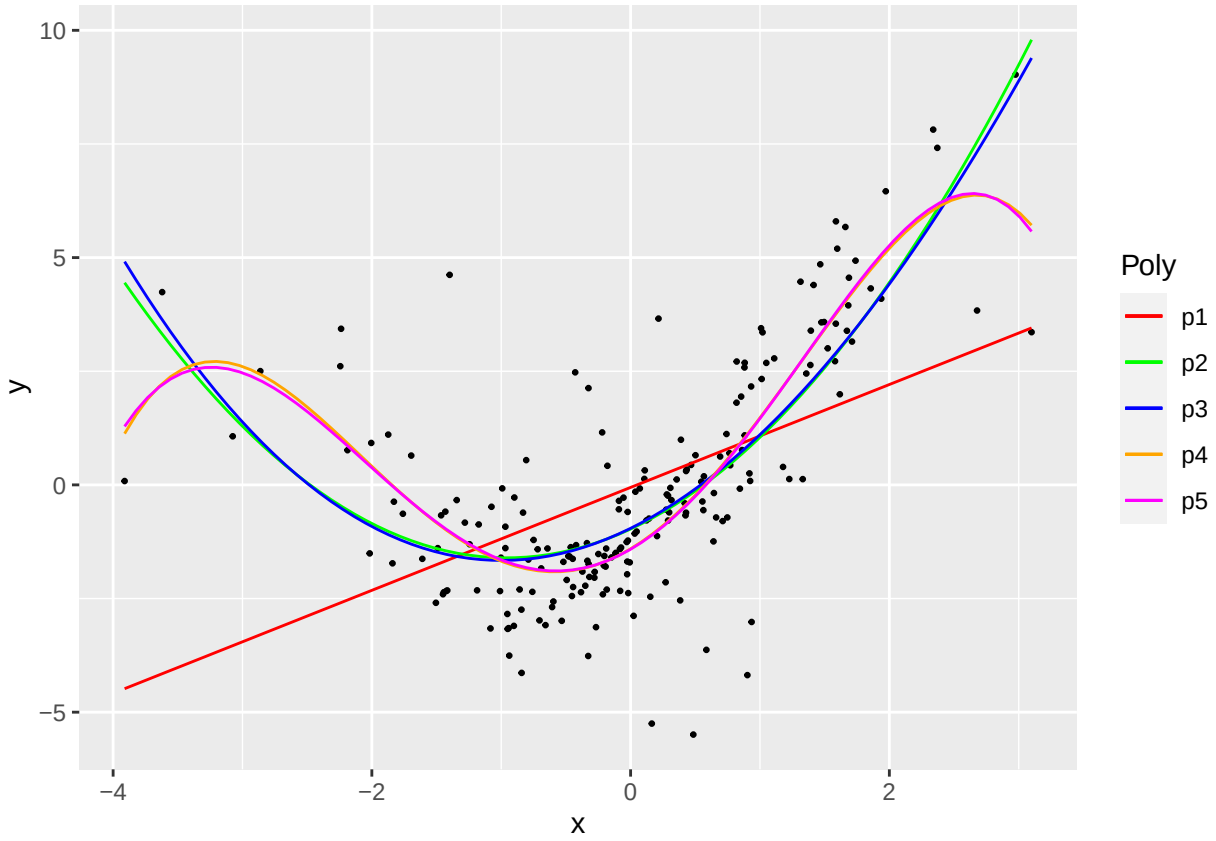
```
y <- 5 + (2*sin(x)) - (7 * (exp(2*cos(x))/(1+exp(2*cos(x))))) + e
```

d.

```
data <- data.frame(y,x)

ggplot(data = data, aes(x = x, y = y)) + geom_point(size = 0.5) + geom_smooth(method = 'lm', se = FALSE
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



e.

```
lm_p1 <- lm(y ~ x, data = data)
lm_p2 <- lm(y ~ poly(x,2), data = data)
```

```
lm_p3 <- lm(y ~ poly(x,3), data = data)
lm_p4 <- lm(y ~ poly(x,4), data = data)
lm_p5 <- lm(y ~ poly(x,5), data = data)

sqrt(mean(lm_p1$residuals^2))
```

```
## [1] 2.178776
```

```
sqrt(mean(lm_p2$residuals^2))
```

```
## [1] 1.611642
```

```
sqrt(mean(lm_p3$residuals^2))
```

```
## [1] 1.610003
```

```
sqrt(mean(lm_p4$residuals^2))
```

```
## [1] 1.454665
```

```
sqrt(mean(lm_p5$residuals^2))
```

```
## [1] 1.454324
```

Aside from `lm_p1`, the model fits the observed data points relatively well. The mean squared of residuals suggests that `lm_p5` performed the best amongs the models, hence we will choose `lm_p5`.

f.

```
lsm <- predict(lm_p2,newdata=data.frame(x=1.00), interval = "confidence", level = 0.9)
lsm
```

```
##        fit       lwr      upr
## 1 1.046632 0.7943725 1.298892
```

The confidence interval of the polynomial regression model with order 2 is [0.7553318, 1.222551] and implies that we can be 90% confident that the true response value at X = 1 falls between 0.7553318 and 1.222551. This interval accounts for the uncertainty in the estimated coefficients and provides a measure of precision for predicting the response value at X = 1 based on the given model.

g.

```
set.seed(123)
b_up <- rep(NA, 1000)
b_lw <- rep(NA, 1000)
for(i in 1:1000) {
  ind <- sample(1:nrow(data),nrow(data), replace = TRUE)
  t_pred<- predict(lm(y ~ poly(x,2), data = data[ind,]), newdata = data.frame(x=1.00), interval = "confi
  b_lw[i] <- t_pred[2]
  b_up[i] <- t_pred[3]
}
c(mean(b_lw), mean(b_up))
```

```
## [1] 0.8035222 1.3035238
```

The mean confidence interval of the polynomial regression model with order 2 is [0.8035222, 1.3035238] through bootstrapping 1000 iterations, and implies that we can be 90% confident that the true response value at X = 1 falls between 0.8035222 and 1.3035238. The reported confidence interval come close to the confidence interval from using least squares theory, hence safe to assume both methods agree on the confidence interval derived.

4.

```
data(College)
```

a. Train 80: Validation 20

```
set.seed(123)
train <- sample(1:nrow(College), nrow(College)*0.8)
val <- (-train)
col.train <- College[train,]
col.val <- College[val,]
```

b.

```
logr <- glm(Private ~ ., data = col.train, family = 'binomial')
summary(logr)
```

```
##
## Call:
## glm(formula = Private ~ ., family = "binomial", data = col.train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8185  -0.0092   0.0526   0.1655   2.9897
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  8.614e-01  2.122e+00   0.406  0.68484
## Apps        -6.653e-04  3.358e-04  -1.981  0.04761 *
## Accept       6.745e-04  6.583e-04   1.025  0.30555
## Enroll       7.093e-04  1.148e-03   0.618  0.53666
## Top10perc    6.631e-03  3.542e-02   0.187  0.85151
## Top25perc    1.153e-02  2.370e-02   0.487  0.62650
## F.Undergrad -6.689e-04  2.595e-04  -2.578  0.00995 **
## P.Undergrad -4.018e-05  2.738e-04  -0.147  0.88334
## Outstate     6.709e-04  1.353e-04   4.958 7.11e-07 ***
## Room.Board   4.440e-05  3.141e-04   0.141  0.88759
## Books        1.576e-03  1.490e-03   1.058  0.29010
## Personal    -1.750e-04  3.275e-04  -0.534  0.59314
## PhD         -5.052e-02  2.972e-02  -1.700  0.08921 .
## Terminal    -4.184e-02  3.026e-02  -1.383  0.16672
## S.F.Ratio   -9.126e-02  6.671e-02  -1.368  0.17131
## perc.alumni  3.232e-02  2.495e-02   1.296  0.19508
## Expend       2.010e-04  1.368e-04   1.469  0.14171
## Grad.Rate    2.130e-02  1.334e-02   1.597  0.11020
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 732.87  on 620  degrees of freedom
## Residual deviance: 172.39  on 603  degrees of freedom
## AIC: 208.39
##
## Number of Fisher Scoring iterations: 8
```

`Top10perc` is relatively insignificant in contributing to whether a school is private or not.

c. Using accuracy,

```
logpred <- predict(logr,newdata=col.val,type='response')
class_preds <- logpred > 0.5
pred_table <- table(col.val$Private, class_preds)
1-mean(sum(diag(pred_table))/sum(pred_table))
```

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

d.

```
lda <- lda(Private ~ ., data = College, subset = train)
lda_pred <- predict(lda, col.val)
1-mean(lda_pred$class == col.val$Private)
```

```
## [1] 0.07051282
```

e.

```
qda <- qda(Private ~ ., data = College, subset = train)
qda_pred <- predict(qda, col.val)
1-mean(qda_pred$class == col.val$Private)
```

```
## [1] 0.08333333
```

f.

```
tunesvm <- tune(e1071::svm, Private ~., data = col.train, kernel = 'linear', ranges = list(cost = c(0.0
bestmod <- tunesvm$best.model
svmpred <- predict(bestmod, col.val)
1-mean(svmpred == col.val$Private)
```

```
## [1] 0.05128205
```

g.

The best model other than logistic seems to be the SVM model because it has the lowest test error. This makes sense because all of the other models rely on underlying assumptions about the probability distribution, whereas the SVM model does not. The SVM model works by constructing a hyperplane that maximizes class separation, so it makes sense that this use case would produce accurate results using the SVM model.

5.

```
protein <- read.csv("./protein.csv")
proteinOG <-protein
protein <- protein[,-1:-2]
```

a.

```
pca <- prcomp(protein, scale = TRUE)
summary(pca)$importance[2,] #p of variance
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9
## 0.44516 0.18167 0.12532 0.10607 0.05154 0.03613 0.03018 0.01292 0.01101
```

```
summary(pca)$importance[3,] #cum p of variance
```

```
##     PC1     PC2     PC3     PC4     PC5     PC6     PC7     PC8     PC9
## 0.44516 0.62683 0.75215 0.85822 0.90976 0.94589 0.97607 0.98899 1.00000
```

b.

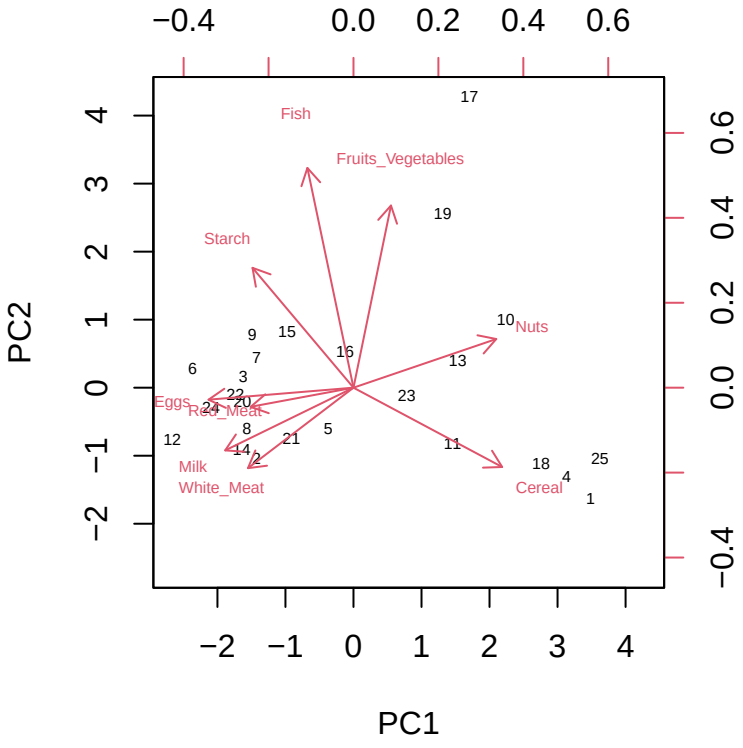
```
pca$rotation
```

```
##                          PC1         PC2         PC3         PC4         PC5
## Red_Meat          -0.3026094 -0.05625165 -0.29757957 -0.646476536  0.32216008
## White_Meat        -0.3105562 -0.23685334  0.62389724  0.036992271 -0.30016494
## Eggs              -0.4266785 -0.03533576  0.18152828 -0.313163873  0.07911048
## Milk              -0.3777273 -0.18458877 -0.38565773  0.003318279 -0.20041361
## Fish              -0.1356499  0.64681970 -0.32127431  0.215955001 -0.29003065
## Cereal             0.4377434 -0.23348508  0.09591750  0.006204117  0.23816783
## Starch            -0.2972477  0.35282564  0.24297503  0.336684733  0.73597332
## Nuts               0.4203344  0.14331056 -0.05438778 -0.330287545  0.15053689
## Fruits_Vegetables  0.1104199  0.53619004  0.40755612 -0.462055746 -0.23351666
##                          PC6         PC7         PC8         PC9
## Red_Meat          -0.45986989  0.15033385 -0.01985770  0.2459995
## White_Meat        -0.12100707 -0.01966356 -0.02787648  0.5923966
## Eggs               0.36124872 -0.44327151 -0.49120023 -0.3333861
## Milk               0.61843780  0.46209500  0.08142193  0.1780841
## Fish              -0.13679059 -0.10639350 -0.44873197  0.3128262
## Cereal             0.08075842  0.40496408 -0.70299504  0.1522596
## Starch             0.14766670  0.15275311  0.11453956  0.1218582
## Nuts               0.44701001 -0.40726235  0.18379989  0.5182749
## Fruits_Vegetables  0.11854972  0.44997782  0.09196337 -0.2029503
```

PC1 is highly correlated with Cereal and Nuts while negatively correlated with Red_Meat, White_Meat, Eggs, and Milk. PC2 is shows high correlation with Fish and Fruits_Vegetables and negative correlation with White_Meat and Cereal. Such suggests PC1 is likely to be negatively correlated with land animal livestock products. From part a, PC1 explains 44.51% of variance of the data and PC2 explains 18.16%.

c.

```
biplot(pca$x[,1:2], pca$rotation[,1:2], cex = 0.5)
```



Most correlated: White_Meat

Unrelated: Fish

Negatively correlated: Nuts

d.

```
rownames(subset(proteinOG, Region == "Center"))
```

```
##  [1] "2"  "3"  "5"  "7"  "9"  "11" "12" "14" "16" "21" "22" "23" "24"
```

```
rownames(subset(proteinOG, Region == "North"))
```

```
## [1] "6"  "8"  "15" "20"
```

The above is printed row index for countries in region Center and North. Corresponding numbers are also plotted in biplot.

The subset of data with Region reported as Center have relatively low PC2 score (around 0) regardless of its PC1 score aside from entry 9. Such suggests the countries in Center region lacks intake from white meat and cereal. Countries in the North region have relatively low PC1 score with it's PC2 score being around 0. Such suggests that countries in North region lacks intake from land animal livestock products.

Conceptual Question

6. The Bootstrap is a resampling technique meant to provide uncertainty estimates of model parameters. In random forests, we are able to use bootstrapping for bagging (bootstrap aggregation), to reduce the variance of a statistical learning method, this splitting each decision tree in a random forest and training them in individual bootstrap training sets to find the average of the resulting predictions. The goal of linear regression is to apply all of our data to determine a relationship between our variables. This method applies a linear relationship between the codependent variables, which is why we use a "best fit line" of our data, and to use bootstrap in linear regression to separate models and average the predictions will not provide a predictive performance of a model in comparison to bootstrapping for random forests.
7. A scenario where we would test multiple hypotheses but would not want to correct for FWER (Family-Wise Error Rate) or FDR (False Discovery Rate) would be when we want to conduct an experiment to study human genes and find any genes that are linked to a specific, understudied disease. Since we do not have enough knowledge about this diseases, we would not have a specific gene, or genetic sequences linked to the disease. In this scenario, we would rather focus on controlling and decreasing false negatives (Type II errors) rather than controlling the risk of false positives (Type I errors), because we would rather accept a higher false positive rate since our goal is to identify the gene and/or sequences associated with this disease that can provided results for the experiment. Thus, our priority with this scenario where we do not want to correct for FWER or FDR is to identify variables before drawing conclusions.
8. It is necessary to be aware of and check a model's assumptions, for the models are built based on their assumptions. Since there is not a single model that fit all situations, assumptions ensure model performs the way it is designed to. For example, if the assumptions of linearity or independence are violated in a linear regression model, the model's predictions may be unreliable. Hence, checking for model not only results in accurate prediction and analysis but also enable model users to take necessary steps to ensure that the model assumptions are met.

Contribution

Our group homework process goes as the following:
1. Each member attempts to complete the homework

2. Compare and discuss the answers
3. Complete a finalized version to submit

All members have contributed about the equal amount to complete this homework.