STAT 4360, Fall 2024
Benny Frisella

Mini Project V

Section 1

Using 'Hitters' data from the ISLR2 library in R. This data includes observations from 263 baseball players with data from 1986 and salary data from 1987. There were 3 categorical variables of two classes in which dummies were used.

1. Considering an unsupervised problem with only data on the predictors.

a. Do you think standardizing the variables before performing the analysis would be a good idea?

Yes, we have many different ranges with many predictors. Standardization may make interpretability a bit harder, but it will aid in the precision of identifying the relationships between variables. Moreover, standardization aids Principal Component Analysis(PCA) because it ensures that each variable contributes equally to the analysis without one predictor dominating due to its larger scale.

b. Standardize the variables and perform PCA on the data. Summarize the results using appropriate tables and graphs. How many PCs would you recommend?

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
Standard deviation     2.6981 2.0371 1.4249 1.24763 0.99933 0.90855 0.83027 0.7163
Proportion of Variance 0.3831 0.2184 0.1069 0.08193 0.05256 0.04345 0.03628 0.0270
Cumulative Proportion  0.3831 0.6016 0.7084 0.79034 0.84290 0.88635 0.92263 0.9496
                           PC9    PC10    PC11    PC12    PC13    PC14    PC15
Standard deviation     0.5007 0.42990 0.37047 0.35704 0.30917 0.24706 0.22798
Proportion of Variance 0.0132 0.00973 0.00722 0.00671 0.00503 0.00321 0.00274
Cumulative Proportion  0.9628 0.97255 0.97978 0.98649 0.99152 0.99473 0.99747
                          PC16    PC17    PC18    PC19
Standard deviation     0.16735 0.11871 0.06973 0.03446
Proportion of Variance 0.00147 0.00074 0.00026 0.00006
Cumulative Proportion  0.99894 0.99968 0.99994 1.00000
```

Figure 1: Explains Variance in Principle Components

Briefly, PCA is a dimensionality reduction technique that identifies the most important directions of variance in data.

After running PCA on the predictors, it can be seen in *Figure 1* above that the **first two** Principal Components account for **60.16%** of the variance. The Cumulative Proportion of Variance of the first 5 Principal Components(PC) accounts for **84.29%** with diminishing marginal Proportion of Variance being added after PC5.

Furthermore, after visualizing a Scree Plot of the Principal Components, it can easily be seen in *Figure 2* below that the change in variance to an extra PC slows down considerably after PC5.

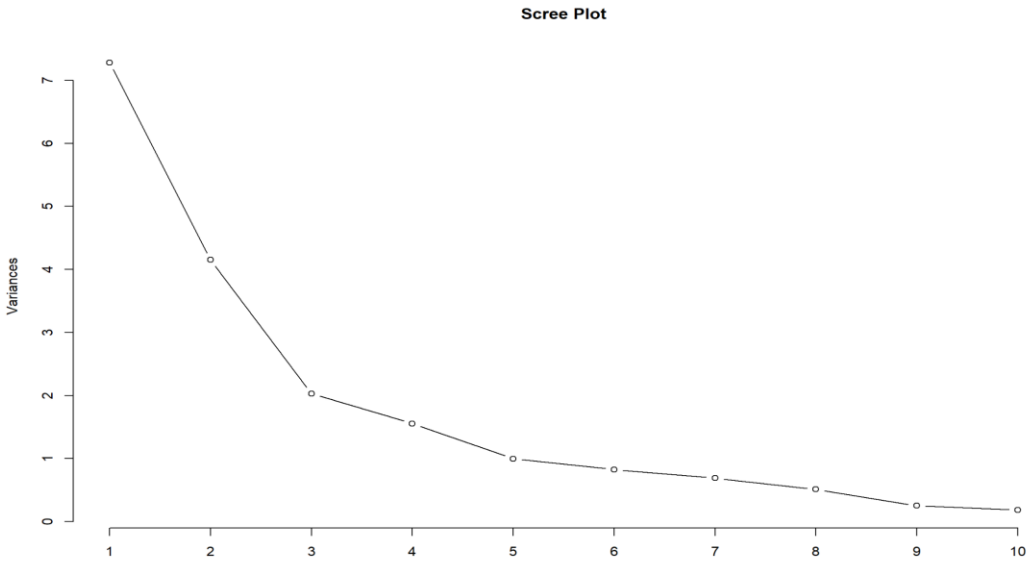

Figure 2: Shows how variance decreases by each successive Principal Component

Therefore, as the focus of PCA is to achieve the lowest number of components that explain majority of the variability, **I would recommend using only the first 5 Principal Components as they explain ~84% of the variance and the change in variance tends to stabilize after PC5 with small improvements to changes in variance thereafter.**

c. *Focus on the first two PCs obtained in (b). Prepare a table showing correlations of the standardized quantitative variables with the two components. Also, display the scores on the two components and the loadings on them using a biplot. Interpret the results.*

Focusing on the first two Principal Components, the correlations between them and the predictors can be seen in *Figure 3* below. The quantitative variables were centered and standardized before running PCA.

	PC1 <dbl>	PC2 <dbl>
AtBat	0.535005797	0.78180908
Hits	0.398990387	1.01791254
HmRun	0.551406348	0.48307059
Runs	0.404033853	1.01912729
RBI	0.634520777	0.64073368
Walks	0.425600010	0.61949859
Years	0.762414828	-0.53454081
CAtBat	0.673188298	-0.52047244
CHits	0.892371793	-0.37258446
CHmRun	0.649795316	-0.34076190
CRuns	0.912516238	-0.35094485
CRBI	0.693314736	-0.45352805
CWalks	0.854763766	-0.39176613
PutOuts	0.158277450	0.42019190
Assists	-0.002270826	0.34356192
Errors	-0.016010376	0.54166892

Figure 3: Correlations of Quantitative Variables with PC1 and PC2

The projected variable vectors(loadings) of all predictors in respect to PC1 and PC2 can be found in *Figure 4* below inside the unit circle. Notice the scores on the axis:

PC1 score = 38.3% & PC2 score = 21.8%

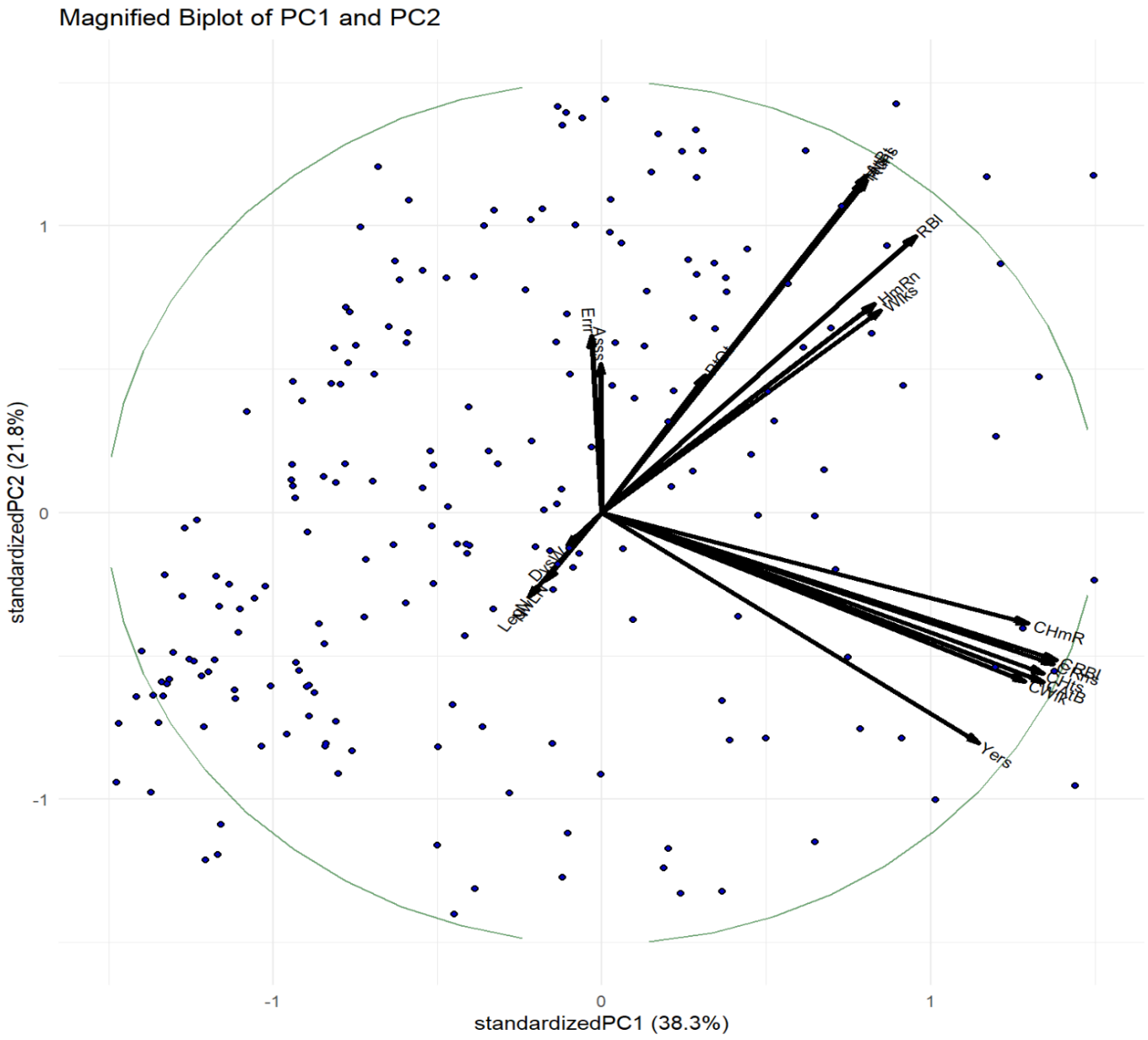

Figure 4: Biplot of PC1 and PC2 Loadings

Per usual it's hard to tell which predictors are where in the Biplot, therefore I separated them out by quadrant. The results are shown below in *Figure 5* on the next page where one can easily distinguish the relationships between PC1/PC2 and the variables. As the angles between the vectors in each quadrant are small, we can acknowledge that,

the variables in their respective quadrants are positively related to each other. Furthermore, notice that only the League/Division(qualitative) variables show negative correlations between both PC1 and PC2. Overall, PC1 and PC2 do a good job of separating the variables.

Quadrant.1 <chr>	Quadrant.2 <chr>	Quadrant.3 <chr>	Quadrant.4 <chr>
AtBat	Assists	LeagueN	Years
Hits	Errors	DivisionW	CAtBat
HmRun	*NA*	NewLeagueN	CHits
Runs	*NA*	*NA*	CHmRun
RBI	*NA*	*NA*	CRuns
Walks	*NA*	*NA*	CRBI
PutOuts	*NA*	*NA*	CWalks

7 rows

Figure 5: Quadrant Separation of Variables in Biplot

As for inverse relationships, Assists and Errors were positively related to PC2 and, although minimally, negatively related to PC1. Meanwhile: Years, At Bats, Hits, Home Runs, Runs, Runs Batted In, and Walks were all extremely positively related to PC1 while also noticeably inversely related to PC2.

2. *Considering a supervised problem with **log(Salary)** as the response(due to skewness) and using **all 19 other predictors** with **all data as training data**, let us test different models.*

a. *Fit a linear regression model. Compute the test MSE of the model.*
b. *Fit a PCR model with M chosen optimally via LOOCV. Compute the test MSE of the model.*
c. *Fit a PLS model with M chosen optimally via LOOCV. Compute the test MSE of the model.*
d. *Fit a ridge regression with penalty parameter chosen optimally via LOOCV. Compute the test MSE of the model.*

e. *Compare the four models. Which model(s) would you recommend? Justify.*

The table below in *Figure 6* shows the test MSE of each model. *The M was chosen optimally via Leave-One-Out-Cross-Validation(LOOCV)* for both Principal Component Regression(PCR) and Partial Least Squares Regression(PLS), while the penalty parameter for Ridge Regression was also chosen via LOOCV.

	Model <chr>	Test_MSE <dbl>
1	Linear Regression	0.3477046
4	Ridge Regression	0.3607908
2	PCR	0.3609446
3	PLS	0.3626667

Figure 6: Comparison of Test MSE for Different Models

According to Figure 6, <u>Linear Regression is my recommendation for predicting log(Salary)</u> as it's the easiest to interpret and has the lowest test MSE - meaning it outperforms the others in prediction accuracy. While PCR and PLS are designed to reduce dimensionality and handle collinearity, they may not capture the relationships as effectively as a simple linear regression model in this case, possibly due to the limited amount of variance explained by the additional components."

3. Using the same data as 2.

a. Fit a linear model to the dataset. Which predictor do you think is the most important?

The linear regression model using all 19 predictors against a response of log(Salary) resulted in Figure 7 below.

Paying attention to the asterisks(**), it's inferred that **Hits and Walks have the smallest p-values at less than 0.05** and thus a higher probability of predicting increases in log(Salary). Ignore the (Intercept), as that's only important when all else are zero. This is interesting because both Hits and Walks contribute to getting players on base, which is generally associated with ***offensive success***

Although **Walks has the lowest p-value (0.00229**), I believe **Hits are the most important variable** with a **p-value of 0.00503**, as they appear to have a **slightly stronger relationship with the outcome**.

```
Call:
lm(formula = Salary ~ ., data = Hitters)

Residuals:
     Min       1Q   Median       3Q      Max
-2.22870 -0.45350  0.09424  0.40474  2.77223

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.618e+00  1.765e-01  26.171  < 2e-16 ***
AtBat       -2.984e-03  1.232e-03  -2.421  0.01620 *
Hits         1.308e-02  4.622e-03   2.831  0.00503 **
HmRun        1.179e-02  1.205e-02   0.978  0.32889
Runs        -1.419e-03  5.794e-03  -0.245  0.80670
RBI         -1.675e-03  5.056e-03  -0.331  0.74063
Walks        1.096e-02  3.554e-03   3.082  0.00229 **
Years        5.696e-02  2.413e-02   2.361  0.01902 *
CAtBat       1.283e-04  2.629e-04   0.488  0.62596
CHits       -4.414e-04  1.311e-03  -0.337  0.73670
CHmRun      -7.809e-05  3.144e-03  -0.025  0.98020
CRuns        1.513e-03  1.459e-03   1.037  0.30072
CRBI         1.312e-04  1.346e-03   0.097  0.92246
CWalks      -1.466e-03  6.377e-04  -2.298  0.02239 *
LeagueN      2.825e-01  1.541e-01   1.833  0.06797 .
DivisionW   -1.656e-01  7.847e-02  -2.111  0.03580 *
PutOuts      3.389e-04  1.505e-04   2.251  0.02526 *
Assists      6.214e-04  4.300e-04   1.445  0.14970
Errors      -1.197e-02  8.537e-03  -1.402  0.16225
NewLeagueN  -1.742e-01  1.536e-01  -1.134  0.25788
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6135 on 243 degrees of freedom
Multiple R-squared:  0.5586,    Adjusted R-squared:  0.524
F-statistic: 16.18 on 19 and 243 DF,  p-value: < 2.2e-16
```

Figure 7: Linear Model Coefficients and p-values

b. Fit a natural cubic spline regression model using the predictor you selected from question (a) to the data. Use LOOCV to determine the number of knots. Summarize the results. Report estimated test MSE for the best model.

Fitting a natural cubic spline regression model using Hits as the lone predictor I used LOOCV to determine the optimal number of knots as shown on the following page in *Figure 8*.

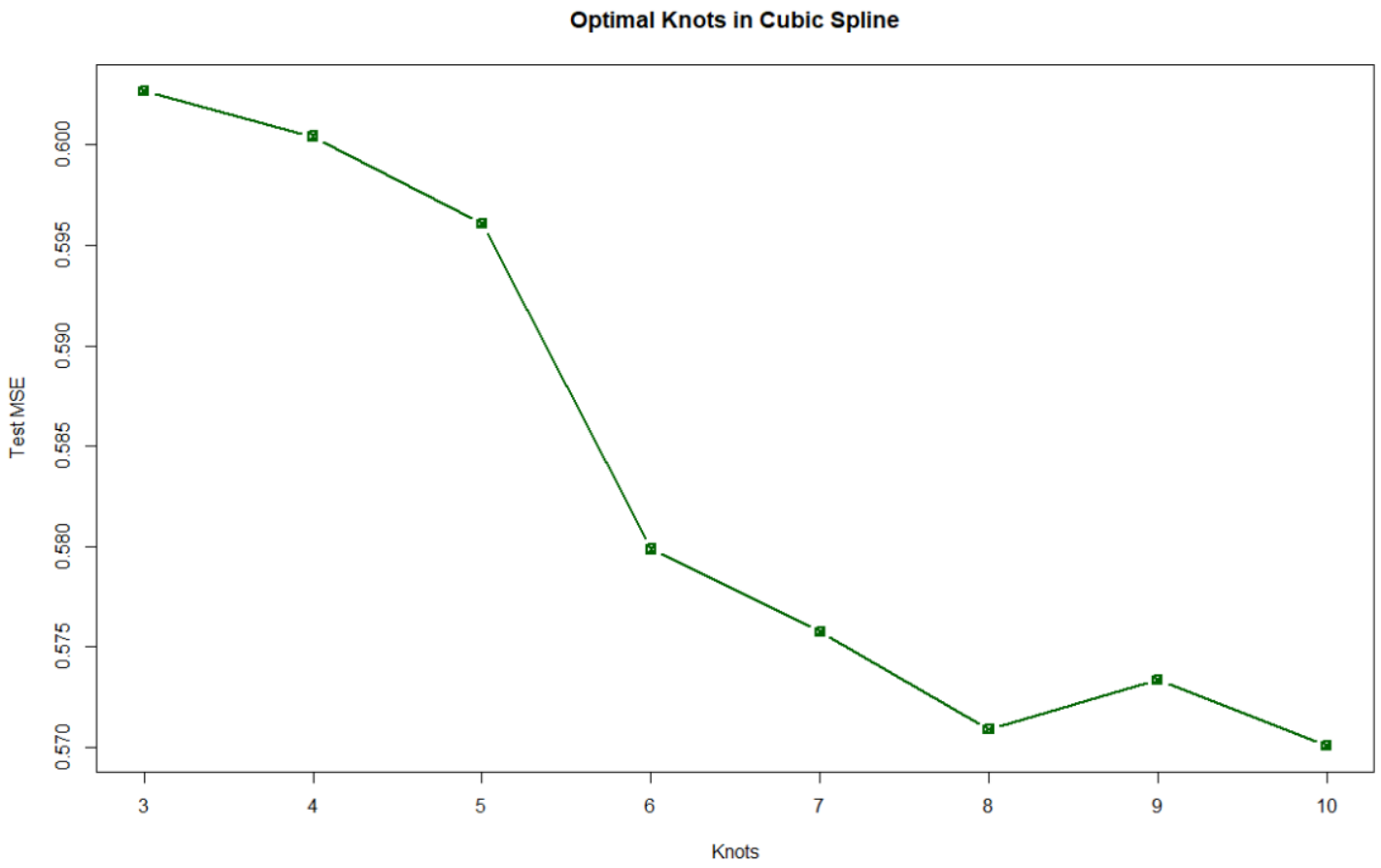

Figure 8: Natural Cubic Spline Model for Hits Predictor

The optimal number of knots was determined to be 10. Using this configuration, the test MSE was 0.5700916. Despite Walks having the lower p-value, the cubic spline model using Walks as the sole predictor resulted in a slightly higher test MSE of **0.5873339**.

Section 2 R Code

````
---
title: "Mini Project 5"
author: "Benny Frisella"
date: "2024-11-20"
output: pdf_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

```{r}
library("ISLR2")
library(pls)

##     Question 1     ##

# (b)

Hitters <- na.omit(Hitters)

summary(Hitters)

#?pcr

#unsupervised, without Salary
unsuper_Hitters <- subset(Hitters, select = -19)

#convert 3 columns to categorical
unsuper_Hitters$League <- as.factor(unsuper_Hitters$League)

unsuper_Hitters$Division <- as.factor(unsuper_Hitters$Division)

unsuper_Hitters$NewLeague <- as.factor(unsuper_Hitters$NewLeague)

str(unsuper_Hitters)

#extract numeric first
numeric <- sapply(unsuper_Hitters, is.numeric)
numeric_data <- unsuper_Hitters[, numeric]

#standardize, center data.
numeric_data <- scale(numeric_data)

#extract categorical, create dummys
categorical_data <- unsuper_Hitters[, sapply(unsuper_Hitters, is.factor)]
````
````

```
#exclude intercept column
dummys <- model.matrix(~ ., data = categorical_data)[,-1]

#join numeric and non-numeric
unsuper_Hitters_standard <- cbind(numeric_data, dummys)

str(unsuper_Hitters_standard)

#PCA
PCA <- prcomp(unsuper_Hitters_standard, center = TRUE, scale. = TRUE)

summary(PCA)

#scree
plot(PCA, type = "l", main = "Scree Plot")
```

```{r}

library(ggbiplot)

#?ggbiplot

# (c)
#get loadings/rotation matrix
loadings <- PCA$rotation[, 1:2]

#get standard deviations of the PCs
std_devs <- PCA$sdev[1:2]

#correlations: Loadings scaled by PC standard deviations
correlations <- loadings * std_devs

#display correlations in a table
cor_table <- as.data.frame(correlations)
colnames(cor_table) <- c("PC1", "PC2")
print(cor_table)

#create the biplot and zoom in for presentability
ggbiplot(PCA,
      circle = TRUE,
      labels = NULL,
      varname.adjust = ,
      varname.abbrev = TRUE) +
 theme_minimal() +
 ggtitle("Magnified Biplot of PC1 and PC2") +
 geom_point(aes(color = 'blue'), size = 0.75, alpha = 0.75) +
 scale_color_manual(values = c("blue")) +
```

```
  theme(legend.position = "none") +
  xlim(-1.5, 1.5) +  #zoom x
  ylim(-1.5, 1.5)    #Zoom y
```

```{r}

#(C) continued...

#get the variable loadings (rotation)
var_loadings <- as.data.frame(PCA$rotation)

#create quadrant column
var_loadings$quadrant <- with(var_loadings,
                              ifelse(PC1 > 0 & PC2 > 0, "Quadrant 1",
                              ifelse(PC1 < 0 & PC2 > 0, "Quadrant 2",
                              ifelse(PC1 < 0 & PC2 < 0, "Quadrant 3", "Quadrant 4"))))

#list variables by quadrant
quadrant_list <- split(rownames(var_loadings), var_loadings$quadrant)

#find the maximum length of the lists
max_len <- max(sapply(quadrant_list, length))

#pad shorter lists with NA
quadrant_list_padded <- lapply(quadrant_list, function(x)
{
  length(x) <- max_len  # Pad with NAs
  return(x)
})

#create a table
quadrant_df <- data.frame(
  `Quadrant 1` = quadrant_list_padded$`Quadrant 1`,
  `Quadrant 2` = quadrant_list_padded$`Quadrant 2`,
  `Quadrant 3` = quadrant_list_padded$`Quadrant 3`,
  `Quadrant 4` = quadrant_list_padded$`Quadrant 4`
)

print(quadrant_df)


```

```{r}
##     Question 2     ##
```

#(a) log(Salary) as response, all predictors, linear regression, get MSE

```
Hitters$Salary <- log(Hitters$Salary)

log_Salary_Hitters <- lm(Salary ~ ., data = Hitters)

#predict
predictions <- predict(log_Salary_Hitters, newdata = Hitters)

#grab differences
squared_errors <- (Hitters$Salary - predictions)^2

#get MSE
test.mse.lm <- mean(squared_errors)

print(paste("Test MSE:", test.mse.lm))
```

```{r}
**#(b) log(Salary) as response, all predictors, Principle Component Regression w/ M from LOOCV, get MSE**
#?pcr

set.seed(1)

#PCR using LOOCV to find optimal M
pcr.fit <- pcr(Salary ~ ., data = Hitters, scale = TRUE, validation = "LOO")


#view
summary(pcr.fit)

MSEP(pcr.fit)
sqrt(MSEP(pcr.fit)$val[1, 1,])
which.min(MSEP(pcr.fit)$val[1, 1,])


#plot the cross-validation test MSE estimates
validationplot(pcr.fit, val.type = "MSEP")

#FOUND lowest adjusted CV at M = 14***

#fit PCR model using M
optimal.pcr <- pcr(Salary ~ ., data = Hitters, scale = TRUE, ncomp = 14)

summary(optimal.pcr)

#get new predictions
```

```
predictions.pcr <- predict(optimal.pcr, newdata = Hitters, ncomp = 14)

#get new MSE
test.mse.pcr <- mean((Hitters$Salary - predictions.pcr)^2)
print(paste("Test MSE for PCR:", test.mse.pcr))

#use all data for predictions


```

```{r}
#(c) log(Salary) as response, all predictors, PLS w/ M from LOOCV, get MSE
set.seed(1)
pls.fit <- plsr(Salary ~ ., data = Hitters, scale = TRUE, validation = "LOO")

summary(pls.fit)

MSEP(pls.fit)
sqrt(MSEP(pls.fit)$val[1, 1,])

optimal_m <- which.min(MSEP(pls.fit)$val[1, 1,])

#plot the cross-validation test MSE estimates
validationplot(pcr.fit, val.type = "MSEP")

#at M = 13,
#%variance increase stabilizes future components see decreasing marginal returns***
final.pls.fit <- plsr(Salary ~., data = Hitters, scale = TRUE, ncomp = 13)

#predict
predictions.pls <- predict(final.pls.fit, newdata = Hitters)

#get MSE
test.mse.pls <- mean((Hitters$Salary - predictions.pls)^2)
print(paste("Test MSE for PLS:", test.mse.pls))

```

```{r}
#(d) log(Salary) as response, all predictors, Ridge Regression w/ pp from LOOCV, get MSE
library(glmnet)

y <- Hitters$Salary

#create design matrix
x <- model.matrix(Salary ~ ., Hitters)[, -1]
```

```
#fit ridge regression using LOOCV to find the optimal lambda
ridge.mod <- cv.glmnet(x, y, alpha = 0, type.measure = "mse", nfolds = nrow(x))

#find optimal lambda
optimal_lambda <- ridge.mod$lambda.min

optimal_lambda

#optimal lambda at 0.009543091***

#fit the ridge regression model using the optimal lambda
final.ridge <- glmnet(x, y, alpha = 0, lambda = optimal_lambda)

#predict
ridge.predictions <- predict(final.ridge, s = optimal_lambda, newx = x)

#compute the test MSE
test.mse.ridge <- mean((ridge.predictions - y)^2)

#print the test MSE
cat("Test MSE for Ridge Regression:", test.mse.ridge)
```

```{r}
#(e) compare models
model_names <- c("Linear Regression", "PCR", "PLS", "Ridge Regression")
test_mse <- c(test.mse.lm, test.mse.pcr, test.mse.pls, test.mse.ridge)

#gather
results_table <- data.frame(Model = model_names, Test_MSE = test_mse)

#test MSE in ascending order
results_table <- results_table[order(results_table$Test_MSE), ]


print(results_table)

#Linear Regression wins


```

```{r}
##     Question 3     ##



#(a) log(Salary) as response, all predictors, which predictor is the most important?
```

#this was done in question 2 (a), reusing...

summary(log_Salary_Hitters)

```

```{r}
**#(b) fit a natural cubie spline using Hits** or Walks**, use LOOCV for optimal knots**,
#summarize and get test MSE

library(splines)

#Hits had the 2nd lowest p-value of 0.00503**, walks was the lowest with 0.00229**
#important_predictor <- "Hits"
important_predictor <- "Walks"

#use LOOCV MSE for natural cubic splines
loocv_spline <- function(k)
{
 #fit a natural spline
 spline_model <- lm(Salary ~ ns(Hitters[[important_predictor]], df = k), data = Hitters)

 # Leave-One-Out Cross-Validation
 loocv_mse <- mean((Hitters$Salary - predict(spline_model, newdata = Hitters))^2)
 return(loocv_mse)
}

#test different values for the degrees of freedom (number of knots)
k_values <- 3:10  #possible knots
mse_values <- sapply(k_values, loocv_spline)

#find optimal number of knots
optimal_k <- k_values[which.min(mse_values)]
cat("Optimal number of knots:", optimal_k, "\n")

#fit a model with the optimal number of knots
best_spline_model <- lm(Salary ~ ns(Hitters[[important_predictor]], df = optimal_k), data = Hitters)

test_mse <- min(mse_values)
cat("Test MSE for the best natural spline model:", test_mse, "\n")

```

```{r}
#visualize
plot(k_values, mse_values, type = "b", pch = 7, col = "dark green",

```
    xlab = "Knots",
    ylab = "Test MSE",
    main = "Optimal Knots in Cubic Spline",
    lwd = 2)

```