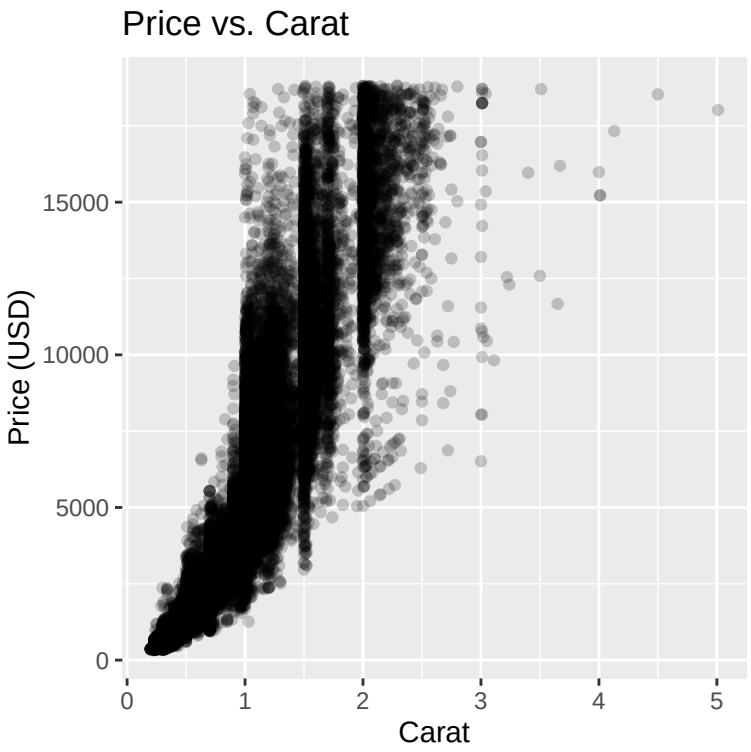
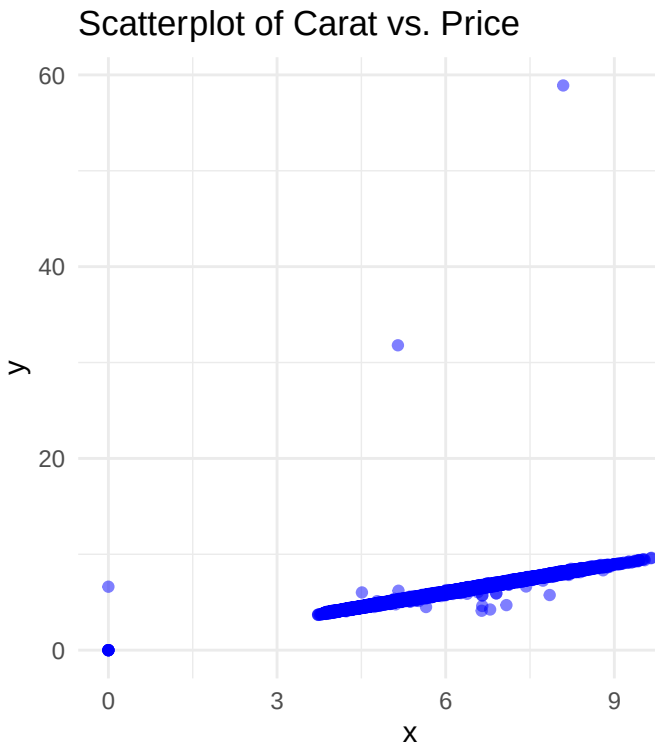
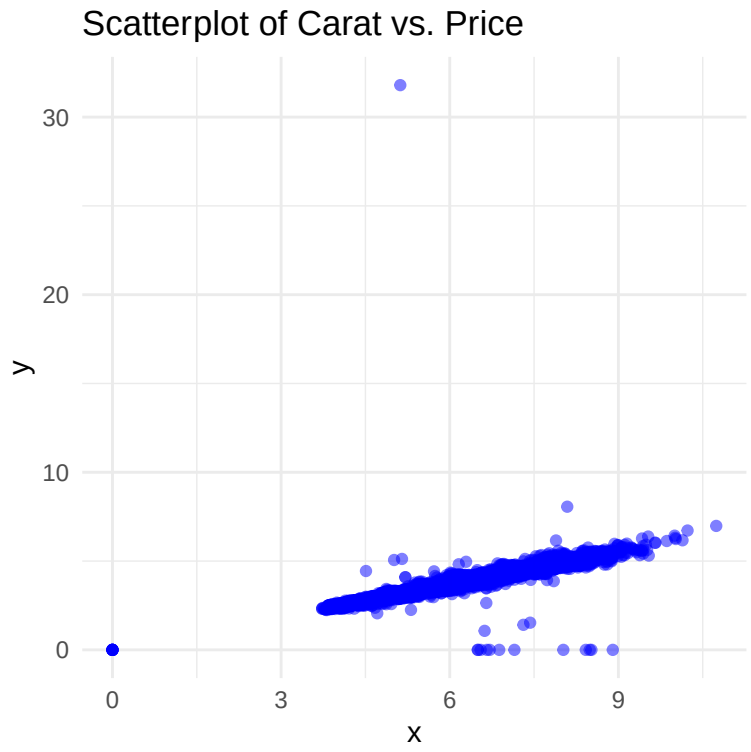
HW1

Your Name

Problem 1


Price vs. Carat
Price (USD)
15000
10000
5000
0
0
1
2
3
4
5
Carat


Scatterplot of Carat vs. Price
y
60
40
20
0
0
3
6
9
x

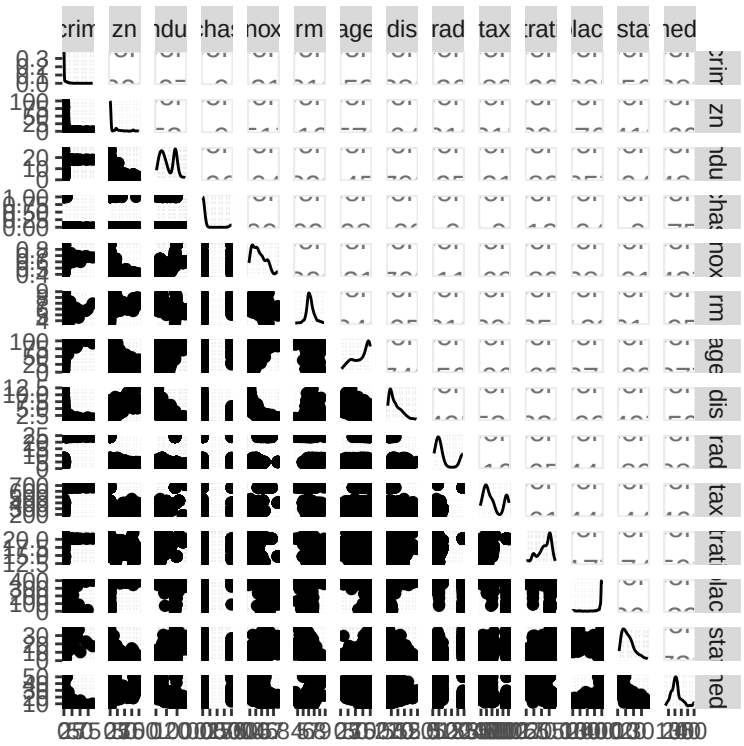

Scatterplot of Carat vs. Price
y
30
20
10
0
0
3
6
9
x

-There is a strange slashing that seems to occur through the plot between the high one karat diamonds, and the 2 carat line. This may indicate that some carats are being rounded up when the are close to 2. -The Minimum X, Y, Z dimensions are 0. this should not be possible and indicates a data entry or rounding error -There are two extreme outliers in X and Y. It seems unrealistic to have a diamond whose x is around8 and y is around 60 -This problem is also evident when you compare X to Z. There is one data ponit with x=5 and z = 35. this does not seem reasonable

Problem 2

Problem 2(a)

Here are the results:




```
## [1] "This is interesting. There are many complex shapes shown by the ggpairs image. additionally, se
```

- There is no clear correlation or colinearity issue.
- The distributions of the features are novel with many different skwed, bimodal, and L shaped distributions. this indicates that some feature engineering may be needed when fitting a linear model

Problem 2(b)

```
##    Predictor  Coefficient      P_Value
```

```
## 1       crim  -0.41519028 1.173987e-19
## 2         zn   0.14213999 5.713584e-17
## 3      indus  -0.64849005 4.900260e-31
## 4       chas   6.34615711 7.390623e-05
## 5        nox -33.91605501 7.065042e-24
## 6         rm   9.10210898 2.487229e-74
## 7        age  -0.12316272 1.569982e-18
## 8        dis   1.09161302 1.206612e-08
## 9        rad  -0.40309540 5.465933e-19
## 10       tax  -0.02556810 5.637734e-29
## 11   ptratio  -2.15717530 1.609509e-34
## 12     black   0.03359306 1.318113e-14
## 13     lstat  -0.95004935 5.081103e-88
## 14      chas   6.34615711 7.390623e-05
```

Problem 2(c)

The following variables are statistically significant in the SLR models:

- It appears that all predictors are statistically signifficant albeit to varyaing levels.

Problem 2(d)

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

Problem 2(e)

The majority of features come back as statistically significant. The only ones which are not are: - indus - age

Problem 2(f)

The coeficients are actually larger in the multiple regression scenerio. This model also has a more reasonable Rsquared at .74 than any of the SLR models. This all makes sense as it is able to controll for multiple factors.

Problem 3

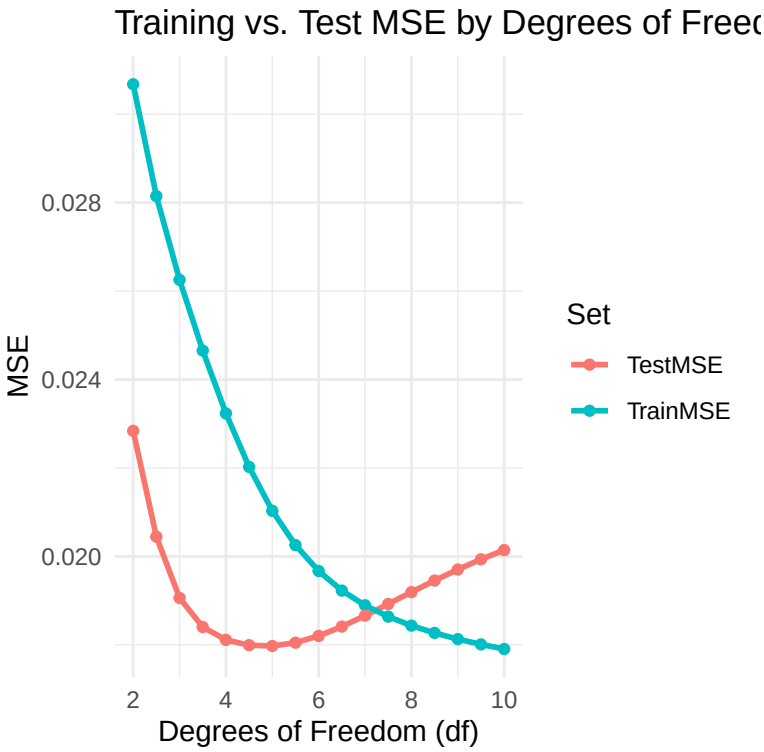
Problem 3(a)

Increasing df makes the spline conform to the data more closely. This can eventually lead to overfitting

Problem 3(b)

Overfitting the data occurs when you are actually fitting your model to the natural error in the data, not jsut the signal. This results in strong train - metric performance, but weakened test metric performance as you are describing the actual sample at hand more than the general population. A great example of this is a decision tree where you can almost always reach 100% accuracy (unless you have duplicate datapoitns with different lables) but as you increase the depth of your tree and you try to get closer to 100% you begin to actuallly fit the charictaristic of the dataset instead of the population.

Problem 3(c)




```
## Minimum MSE: 0.01797442
```

Problem 3(d)

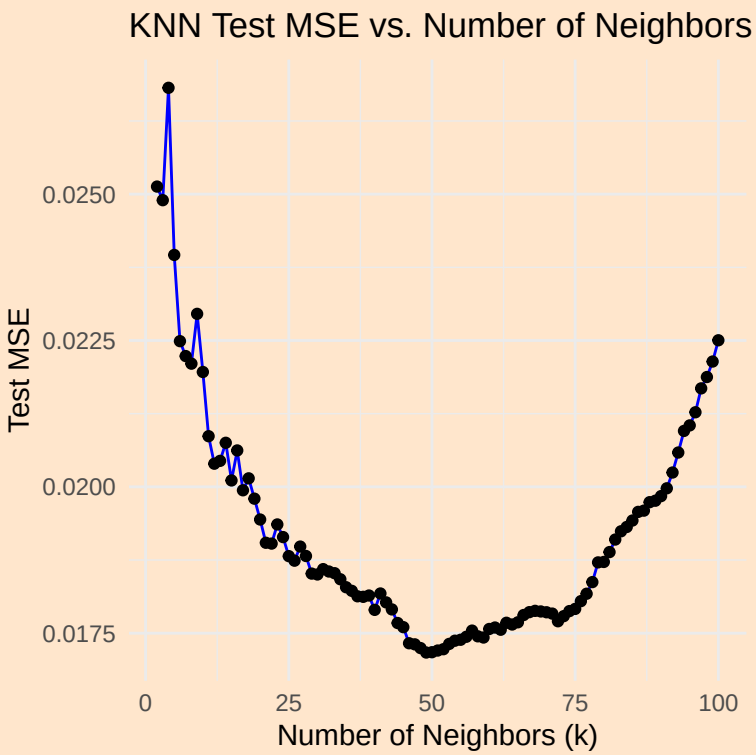
either df = 4.5 or 5 would be ideal in this case as this is where test mse minimizes. Its worth noting that one its rare for your MSE to be lower than train mse like it is here. Id like to see this on a cross fold or something else to confirm.

Problem 4(a)

Problem 4(b)

```
## [1] "Test MSE for k = 10: 0.0219621929120729"
```

Problem 4(c)



Problem 4(d)

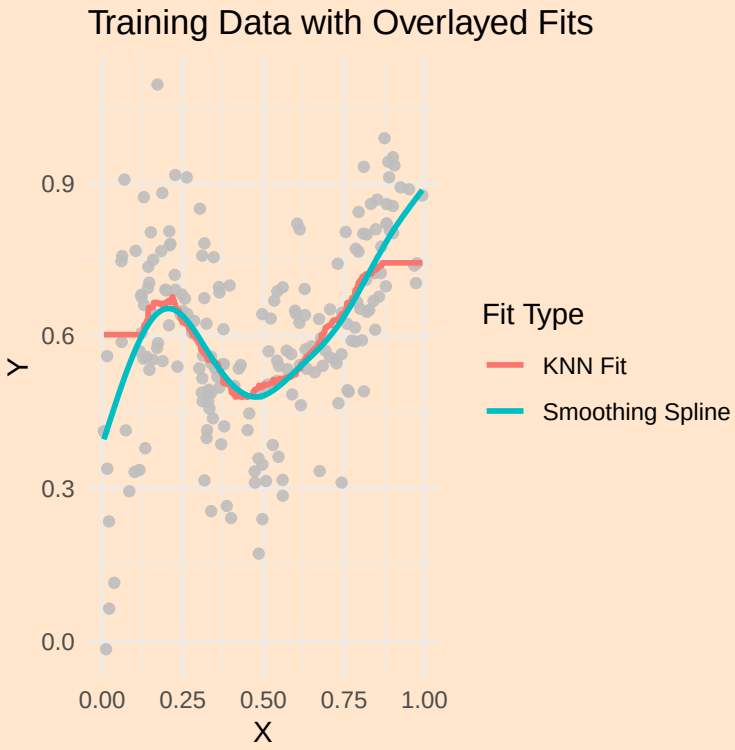
```
## [1] "Best k: 49"

## [1] "KNN Minimum Test MSE: 0.0171694398769078"

## [1] "Splines Test MSE: 0.0179744214783609"

## [1] "KNN provides a lower test MSE"
```

Problem 4(e)



Appendix

```
library(knitr)
# install the tidyverse library (do this once) install.packages('tidyverse')
library(tidyverse)
# set chunk and figure default options
knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE, fig.width = 4,
    fig.height = 4, tidy = TRUE)

## load the data
library(ggplot2)

## take a look head(diamonds)

################################### Continue your analysis here ##


ggplot(diamonds, aes(x = carat, y = price)) + geom_point(alpha = 0.2) + labs(title = "Price vs. Carat",
    x = "Carat", y = "Price (USD)")

print("There is a strange slashing that seems to occur through the plot between the high one karat diam

summary(diamonds[, c("carat", "price", "depth", "table", "x", "y", "z")])
print("The Minimum X, Y, Z dimensions are 0. this should not be possible and indicates a data entry or

ggplot(diamonds, aes(x = x, y = y)) + geom_point(alpha = 0.5, color = "blue") + labs(title = "Scatterpl
```

```
    x = "x", y = "y") + theme_minimal()
print(".")

ggplot(diamonds, aes(x = x, y = z)) + geom_point(alpha = 0.5, color = "blue") + labs(title = "Scatterplo
    x = "x", y = "y") + theme_minimal()
print("This problem is also evident when you compare X to Z. There is one data ponit with x=5 and z = 35
## load the data
library(MASS)

## take a look
head(Boston)
## from the hint
library(GGally)

data("Boston")
ggpairs(Boston)

print("This is interesting. There are many complex shapes shown by the ggpairs image. additionally, seve

# R code for problem 2b
predictors <- c("crim", "zn", "indus", "chas", "nox", "rm", "age", "dis", "rad",
    "tax", "ptratio", "black", "lstat", "chas")

results <- data.frame(Predictor = character(), Coefficient = numeric(), P_Value = numeric())

# Loop through each predictor, fit the model, and extract the p-value
for (predictor in predictors) {
    formula <- as.formula(paste("medv ~", predictor))
    model <- lm(formula, data = Boston)

    p_value <- summary(model)$coefficients[2, 4]
    coefficient <- summary(model)$coefficients[2, 1]  # Coefficient (slope)

    results <- rbind(results, data.frame(Predictor = predictor, Coefficient = coefficient,
        P_Value = p_value))
}

print(results)

lm_full <- lm(medv ~ ., data = Boston)
summary(lm_full)
library(tidyverse)

traindata <- read_csv("training_data.csv")
testdata <- read_csv("test_data.csv")

df_values <- seq(2, 10, by = 0.5)

results <- data.frame(df = df_values, TrainMSE = NA, TestMSE = NA)

for (i in seq_along(df_values)) {
    # Fit spline
    fit_spline <- smooth.spline(traindata$xval, traindata$yval, df = df_values[i])
```

```r
    # Training MSE
    yhat_train <- predict(fit_spline, traindata$xval)$y
    train_err <- mean((traindata$yval - yhat_train)^2)

    # Test MSE
    yhat_test <- predict(fit_spline, testdata$xval)$y
    test_err <- mean((testdata$yval - yhat_test)^2)

    # Store
    results$TrainMSE[i] <- train_err
    results$TestMSE[i] <- test_err
}

results_long <- pivot_longer(results, cols = c("TrainMSE", "TestMSE"), names_to = "Set",
    values_to = "MSE")

ggplot(results_long, aes(x = df, y = MSE, color = Set)) + geom_line(size = 1) + geom_point() +
    labs(title = "Training vs. Test MSE by Degrees of Freedom", x = "Degrees of Freedom (df)",
        y = "MSE") + theme_minimal()

test_mse_smooth = min(results$TestMSE)
cat("Minimum MSE:", test_mse_smooth)

library(caret)

knn_model_k10 <- knnreg(yval ~ xval, data = traindata, k = 10)


yhat_test_k10 <- predict(knn_model_k10, newdata = testdata)

test_mse_k10 <- mean((testdata$yval - yhat_test_k10)^2)
print(paste("Test MSE for k = 10:", test_mse_k10))
# Range of k values
k_values <- 2:100
results_knn <- data.frame(k = k_values, TestMSE = NA)

# Loop over k values
for (k in k_values) {
    knn_model <- knnreg(yval ~ xval, data = traindata, k = k)
    yhat_test <- predict(knn_model, newdata = testdata)
    test_mse <- mean((testdata$yval - yhat_test)^2)

    results_knn$TestMSE[results_knn$k == k] <- test_mse
}

ggplot(results_knn, aes(x = k, y = TestMSE)) + geom_line(color = "blue") + geom_point() +
    labs(title = "KNN Test MSE vs. Number of Neighbors (k)", x = "Number of Neighbors (k)",
        y = "Test MSE") + theme_minimal()

# Optimal k
best_k <- results_knn$k[which.min(results_knn$TestMSE)]
print(paste("Best k:", best_k))
```

```r
best_test_mse <- min(results_knn$TestMSE)
print(paste("KNN Minimum Test MSE:", best_test_mse))

print(paste("Splines Test MSE:", test_mse_smooth))
print(ifelse(best_test_mse < test_mse_smooth, "KNN provides a lower test MSE", "Smoothing splines provi

smooth_spline <- smooth.spline(traindata$xval, traindata$yval, df = 5.5)

x_seq <- seq(min(traindata$xval), max(traindata$xval), length.out = 500)
smooth_fit <- data.frame(xval = x_seq, yval = predict(smooth_spline, x_seq)$y, Fit = "Smoothing Spline"

knn_model_best <- knnreg(yval ~ xval, data = traindata, k = best_k)

knn_fit <- data.frame(xval = x_seq, yval = predict(knn_model_best, newdata = data.frame(xval = x_seq)),
    Fit = "KNN Fit")

fit_data <- rbind(smooth_fit, knn_fit)

ggplot(traindata, aes(x = xval, y = yval)) + geom_point(color = "grey", alpha = 0.9) +
    geom_line(data = fit_data, aes(x = xval, y = yval, color = Fit), size = 1) +
    labs(title = "Training Data with Overlayed Fits", x = "X", y = "Y", color = "Fit Type") +
    theme_minimal()
```