Homework 1

LATE

Ross B. Alexander (rbalexan@stanford.edu)

Problem 1 (C2E2)

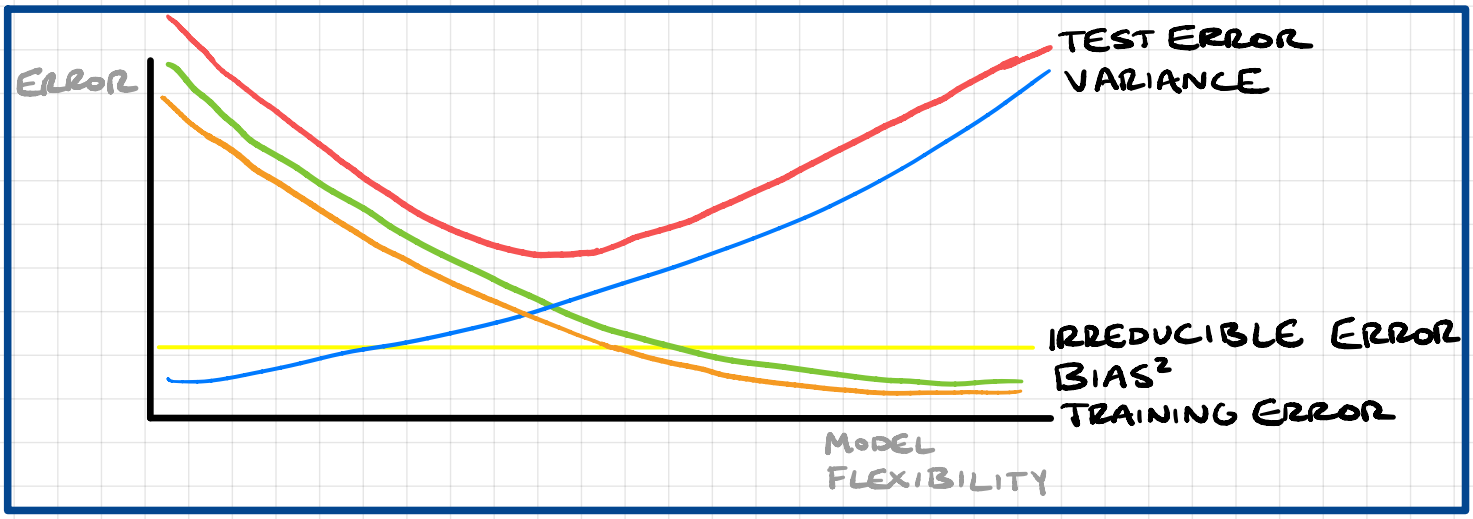
(a) The scenario is a regression problem.
We are interested in inference.
$n = 500$, $p = 3$

(b) The scenario is a classification problem.
We are interested in prediction.
$n = 20$, $p = 13$

(c) The scenario is a regression problem.
We are interested in prediction.
$n = 52$, $p = 3$

Problem 2 (CZE3)

(a)



(b)

Irreducible error: constant (due to underlying noise in data)
Bias^2: decreases since increased model flexibility means the model can fit the underlying trend more tightly
Variance: increases since increased model flexibility means the model parameters are more sensitive to the dataset and any inherent noise
Train error: decreases with bias^2 since increased model flexibility allows training data to be fit with increasing accuracy
Test error: decreases, then increases as model flexibility begins to allow overfitting and test error increases

Problem 3 (C2E7)

(a) The Euclidean distance from $X = (0, 0, 0)$ is:

Obs.	Dist.
1	3
2	2
3	$\sqrt{10}$
4	$\sqrt{5}$
5	$\sqrt{2}$
6	$\sqrt{3}$

(b) With $K=1$, we use the vote from observation 5 – green. This is the result since it is the majority vote from the 1-nearest neighbors.

(c) With $K=3$, we use the majority vote from observations 5, 6, and 2; green, red, and red, which is red. This is the result since it is the majority vote from the 3-nearest neighbors.

(d) If the Bayes decision boundary is highly nonlinear, we would expect K to be small, which allows the decision boundary to be highly flexible. A large K-value would be highly-smoothing and result in a closely linear decision boundary.

Problem 4 (C10E1)

(a) Prove that

$$\frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2$$

where $\bar{x}_{kj} = \frac{1}{|C_k|}\sum_{i \in C_k} x_{ij}$

- we can add and subtract an $\bar{x}_{kj}$ in the squared term to expose the relationship

$$\begin{aligned}
\frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 &= \frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j}+\bar{x}_{kj}-\bar{x}_{kj})^2 \\
&= \frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}\left[(x_{ij}-\bar{x}_{kj})-(x_{i'j}-\bar{x}_{kj})\right]^2 \\
&= \frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}\left[(x_{ij}-\bar{x}_{kj})^2+(x_{i'j}-\bar{x}_{kj})^2-2(x_{ij}-\bar{x}_{kj})(x_{i'j}-\bar{x}_{kj})\right] \\
&= \underbrace{\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2}_{\text{same summation}} + \underbrace{\sum_{i' \in C_k}\sum_{j=1}^{p}(x_{i'j}-\bar{x}_{kj})^2}_{\text{same summation}} - \frac{2}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})(x_{i'j}-\bar{x}_{kj}) \\
&= 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2 - \frac{2}{|C_k|}\sum_{j=1}^{p}\sum_{i \in C_k}(x_{ij}-\bar{x}_{kj})\sum_{i' \in C_k}(x_{i'j}-\bar{x}_{kj}) \\
&= 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2 - \frac{2}{|C_k|}\sum_{j=1}^{p}\sum_{i \in C_k}(x_{ij}-\bar{x}_{kj})\underbrace{(|C_k|\bar{x}_{kj}-|C_k|\bar{x}_{kj})}_{\text{definition of } \bar{x}_{kj}} \to 0 \\
&= 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2
\end{aligned}$$

- therefore, we have shown that

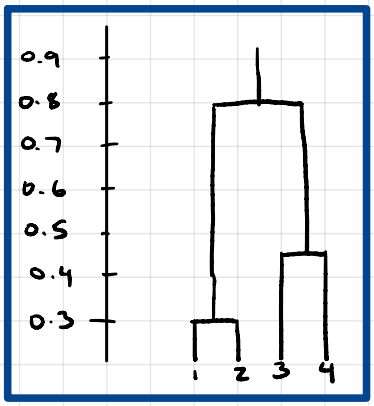
$$\frac{1}{|C_k|}\sum_{i,i' \in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i \in C_k}\sum_{j=1}^{p}(x_{ij}-\bar{x}_{kj})^2 \quad \blacksquare$$

(b) Using the above identity, argue that the k-means clustering algorithm decreases the objective at each iteration.

We see that our objective of minimizing the within-cluster variation is the same as minimizing the distances of points in cluster k to the centroid of cluster k. Since k-means clustering iteratively moves the cluster center to the mean of the points in the cluster, we know that the objective must necessarily decrease. If after reassignment, a point changes clusters, it must also decrease the objective, since the distance to the new cluster is guaranteed to be smaller.

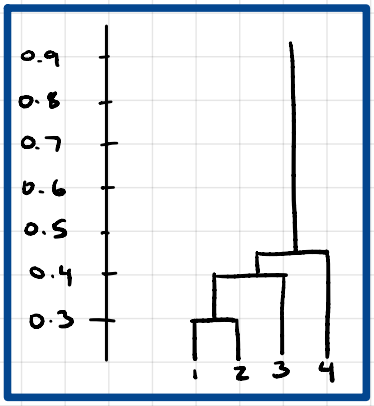
Problem 5 (C10E2)

(a) A sample dendrogram would be:



min over every point 1,2 → 0.3 ✓
(1,2), 3 → 0.5
(1,2), 4 → 0.8
3, 4 → 0.45 ✓
(1,2), (3,4) → 0.8 ✓

(b) A sample dendrogram would be:



min over every point 1,2 → 0.3 ✓
(1,2), 3 → 0.4 ✓
(1,2), 4 → 0.7
3, 4 → 0.45
(1,2,3), 4 → 0.45 ✓

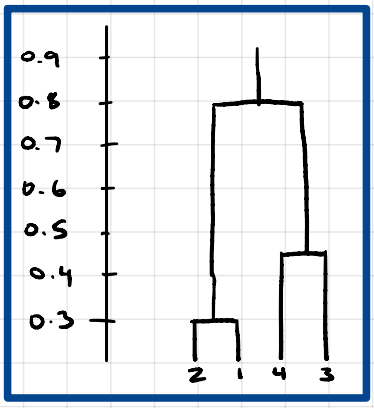
(c) Cutting the dendrogram in (a) for two clusters yields

(1, 2) (3, 4)

(d) Cutting the dendrogram in (b) for two clusters yields

(1, 2, 3) (4)

(e) A modified but equivalent dendrogram of (a) would be



Problem 6 (C10E4)

(a) There is not enough information to tell. If and only if the maximum pairwise dissimilarity between any element in $\{1, 2, 3\}$ and any other element in $\{4, 5\}$ (complete linkage) is equal to the minimum pairwise dissimilarity between the same groups (single linkage) would they fuse at the same level. Otherwise, the complete linkage will fuse above the single linkage. Without these pairwise dissimilarities, we cannot know for certain which will occur.

(b) They will fuse at the same height. Using the above argument, since there is only one point in each cluster, the pairwise dissimilarity between 5 and 6 is only one value and therefore, trivially the maximum and minimum of all pairwise dissimilarities. Since the maximum pairwise dissimilarity is equal to the minimum pairwise dissimilarity, both linkage methods will fuse the points at the same height.

Problem 7 (C10E9)

(a) See attached code.

(b) See attached code.

(c) See attached code.

(d) Standardizing the variables maintains similar clusters with similar sizes, though the states assigned to each cluster shift slightly. Looking at the dataframe, we see that variables `Murder` and `Rape` are on roughly the same scale, but `Assault` and `UrbanPop` are much larger. It is important to measure these features on a similar scale if we have equal weighting of them, since our clustering method relies on a distance metric that will be skewed if features are on different scales.

Problem 8 (C3E4)

(a) We would expect the cubic regression to achieve lower training RSS. The cubic regressor is more flexible than the linear regressor, so we would likely have a better fit to the observed data (including the observation noise) using the cubic regressor.

(b) We would expect the linear regression to achieve lower test RSS. For the same reason as above, the cubic regressor will fit more tightly to the data. However, since the underlying relationship is actually linear, we are likely overfitting with the cubic regressor. Therefore, the linear regressor should achieve lower RSS on the test set.

(c) We would still expect the cubic regressor to have lower training RSS than the linear regressor, even if the underlying relationship is not perfectly linear. More flexible models can always achieve lower training RSS regardless of the underlying trend.

(d) There is not enough information to tell. Without knowing the true underlying relationship, we don't know which, if any, models are overfitting, or by how much. The test RSS actually is meant to provide insight into this, since in reality it is uncommon to know the form of the true underlying relationship.

Problem 9 (C3E9)

- for (e) and (f), you only need to try a few interactions and transformations

(a) See attached code

(b) See attached code

(c) See attached code.

(i) There is a relationship between the predictors and the response (F-statistic prob. is 2.04×10^{-139}.)

(ii) The constant, 'displacement', 'weight', 'year', and 'origin' are all statistically significant ($p < 0.05$) for the 'mpg' response.

(iii) The coefficient for 'year', which is 0.7508 suggests that there is a 0.7508 mile per gallon increase in fuel efficiency per year (on-average).

(d) See attached code.

There are a few problems with the fit. There are of course a few outliers, but the main issues look like quadratic trends in the residuals for 'displacement', 'horsepower', and 'weight'. The 'buick estate wagon (sw)' is also a high leverage point with a studentized residual of around -1.6.

(e) See attached code.

I tried a few combinations and found statistical significance of the 'horsepower' * 'displacement' interaction on the response

(f) See attached code.

I tried a few transformations, but only find statistical significance of 'horsepower'-squared and 'weight'-squared on the response. However, both of these coefficients are near zero, indicating that there is unlikely to be a quadratic trend.

Problem 10 (C3E14)

(a) See attached code.

The model form is $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon, \quad \epsilon \sim N(0,1)$
The regression coefficients are $\beta_0 = 2, \beta_1 = 2, \beta_2 = 0.3$.

(b) See attached code.

$Corr(x_1, x_2) = 0.8351$

(c) See attached code.

We obtain $\hat{\beta}_0 = 2.1305$ (very close to $\beta_0 = 2$)
$\hat{\beta}_1 = 1.4396$ (relatively close to $\beta_1 = 2$)
$\hat{\beta}_2 = 1.0097$ (not that close to $\beta_2 = 0.3$).
The p-value for x_1 is 0.049, thus we can reject $H_0: \beta_1 = 0$.
The p-value for x_2 is 0.375, thus we cannot reject $H_0: \beta_2 = 0$.
Overall, the F-statistic probability is 1.16×10^{-5}, so our model is generally statistically significant.

(d) See attached code.

We obtain $\hat{\beta}_0 = 2.1124$ (very close to $\beta_0 = 2$)
$\hat{\beta}_1 = 1.9754$ (very close to $\beta_1 = 2$).
The p-value for x_1 is less than 1×10^{-3}, so we can reject $H_0: \beta_1 = 0$.

(e) See attached code.

We obtain $\hat{\beta}_0 = 2.3899$ (relatively close to $\beta_0 = 2$)
$\hat{\beta}_2 = 2.8996$ (not close to $\beta_2 = 0.3$)
The p-value for x_2 is less than 1×10^{-3}, so we can reject $H_0: \beta_2 = 0$.

(f) The results in (c) - (e) do contradict each other, but this is to be expected since x_1 and x_2 are highly correlated. The collinearity in the aggregate model leads to rejecting only the H_0 for β_1. However, in the independent models, this collinearity is removed and either variable does describe the relation to the response variable.

(g) See attached code.

On (c) $\hat{\beta}_0$ changes from 2.1305 to 2.2267 ($\beta_0 = 2$) a little worse
$\hat{\beta}_1$ changes from 1.4396 to 0.5394 ($\beta_1 = 2$) much worse
$\hat{\beta}_2$ changes from 1.0097 to 2.5146 ($\beta_2 = 0.3$) much worse
The model gets worse since the new observation is an outlier and it gets worse by a lot since the new observation is also a high leverage point. We also now accept $H_0: \beta_1 = 0$ and reject $H_0: \beta_2 = 0$, a reversal from (c).

On (d) $\hat{\beta}_0$ changes from 2.1124 to 2.2569 ($\beta_0 = 2$) a little worse
$\hat{\beta}_1$ changes from 1.9754 to 1.7657 ($\beta_1 = 2$) a little worse
The model gets worse since the new observation is an outlier but it doesn't get much worse since the new observation isn't that high leverage in x_1 alone. We still reject $H_0: \beta_1 = 0$.

On (e) $\hat{\beta}_0$ changes from 2.3899 to 2.3451 ($\beta_0 = 2$) a little better
$\hat{\beta}_2$ changes from 2.8996 to 3.1190 ($\beta_2 = 0.3$) a little worse, still wrong
The model gets worse since the new observation is an outlier but it doesn't get much worse since the new observation isn't that high leverage in x_2 alone. We still reject $H_0: \beta_2 = 0$.

Problem 11

Let $x_1, \dots, x_n$ be a fixed set of input points and $y_i = f(x_i) + \epsilon_i$ where $\epsilon_i \overset{iid}{\sim} P_\epsilon$ with $\mathbb{E}[\epsilon_i] = 0$ and $\text{Var}(\epsilon_i) < \infty$. Prove that the MSE of a regression estimate $\hat{f}$ fit to $(x_1, y_1), \dots, (x_n, y_n)$ for a random test point x_0 or $\mathbb{E}\left[y_0 - \hat{f}(x_0)\right]^2$ decomposes into variance, square bias, and irreducible error components.

- We have

$$\begin{aligned}
\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right] &= \mathbb{E}\left[y_0^2 - 2y_0\hat{f}(x_0) + \hat{f}(x_0)^2\right] \\
&= \mathbb{E}[y_0^2] - 2\mathbb{E}[y_0\hat{f}(x_0)] + \mathbb{E}[\hat{f}(x_0)^2] \\
&= \mathbb{E}\left[(f(x_0) + \epsilon_0)^2\right] - 2\mathbb{E}\left[(f(x_0) + \epsilon_0)\hat{f}(x_0)\right] + \mathbb{E}[\hat{f}(x_0)^2] \\
&= f(x_0)^2 + 2f(x_0)\cancelto{0}{\mathbb{E}[\epsilon_0]} + \mathbb{E}[\epsilon_0^2] - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] - 2\cancelto{0}{\mathbb{E}[\epsilon_0]}\mathbb{E}[\hat{f}(x_0)] + \mathbb{E}[\hat{f}(x_0)^2] \\
&= f(x_0)^2 + \text{Var}(\epsilon_0) - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + \mathbb{E}[\hat{f}(x_0)^2] \\
&= \text{Var}(\epsilon_0) + f(x_0)^2 - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + \mathbb{E}[\hat{f}(x_0)^2] \\
&= \text{Var}(\epsilon_0) + \mathbb{E}\left[(f(x_0) - \hat{f}(x_0))^2\right] \\
&= \text{Var}(\epsilon_0) + \mathbb{E}\left[\left(f(x_0) - \hat{f}(x_0) + \mathbb{E}[\hat{f}(x_0)] - \mathbb{E}[\hat{f}(x_0)]\right)^2\right] \\
&= \text{Var}(\epsilon_0) + \mathbb{E}\left[\left(f(x_0) - \left(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right) - \mathbb{E}[\hat{f}(x_0)]\right)^2\right] \\
&= \text{Var}(\epsilon_0) + \mathbb{E}\Big[f(x_0)^2 - 2f(x_0)\left(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right) - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + \\
&\qquad + \left(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right)^2 + 2\left(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right)\mathbb{E}[\hat{f}(x_0)] + \\
&\qquad + \mathbb{E}[\hat{f}(x_0)]^2\Big] \\
&= \text{Var}(\epsilon_0) + f(x_0)^2 - 2f(x_0)\mathbb{E}\left[\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right] - 2f(x_0)\mathbb{E}\left[\mathbb{E}[\hat{f}(x_0)]\right] + \\
&\quad + \mathbb{E}\left[\left(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right)^2\right] + 2\mathbb{E}\left[\left(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right)\mathbb{E}[\hat{f}(x_0)]\right] + \\
&\quad + \mathbb{E}\left[\mathbb{E}[\hat{f}(x_0)]^2\right] \\
&= \text{Var}(\epsilon_0) + f(x_0)^2 - 2f(x_0)\mathbb{E}\left[\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right] - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + \\
&\quad + \text{Var}(\hat{f}(x_0)) + 2\mathbb{E}[\hat{f}(x_0)\mathbb{E}[\hat{f}(x_0)]] - 2\mathbb{E}[\mathbb{E}[\hat{f}(x_0)]\mathbb{E}[\hat{f}(x_0)]] + \\
&\quad + \mathbb{E}[\hat{f}(x_0)]^2 \\
&= \text{Var}(\epsilon_0) + f(x_0)^2 - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + 2f(x_0)\mathbb{E}[\mathbb{E}[\hat{f}(x_0)]] - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + \\
&\quad + \text{Var}(\hat{f}(x_0)) + \cancel{2\mathbb{E}[\hat{f}(x_0)]^2} - \cancel{2\mathbb{E}[\hat{f}(x_0)]^2} + \mathbb{E}[\hat{f}(x_0)]^2 \\
&= \text{Var}(\epsilon_0) + f(x_0)^2 - \cancel{2f(x_0)\mathbb{E}[\hat{f}(x_0)]} + \cancel{2f(x_0)\mathbb{E}[\hat{f}(x_0)]} - \\
&\quad - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + \text{Var}(\hat{f}(x_0)) + \mathbb{E}[\hat{f}(x_0)]^2 \\
&= \text{Var}(\epsilon_0) + \text{Var}(\hat{f}(x_0)) + f(x_0)^2 - 2f(x_0)\mathbb{E}[\hat{f}(x_0)] + \mathbb{E}[\hat{f}(x_0)]^2 \\
&= \text{Var}(\epsilon_0) + \text{Var}(\hat{f}(x_0)) + \mathbb{E}\left[(f(x_0) - \hat{f}(x_0))^2\right]
\end{aligned}$$

- We have shown that

$$\mathbb{E}\left[(y_0 - \hat{f}(x_0))^2\right] = \underbrace{\mathbb{E}\left[\left(\hat{f}(x_0) - f(x_0)\right)\right]^2}_{\text{bias}^2} + \underbrace{\mathbb{E}\left[\left(\hat{f}(x_0) - \mathbb{E}[\hat{f}(x_0)]\right)^2\right]}_{\text{variance } (\text{Var}(\hat{f}(x_0)))} + \underbrace{\mathbb{E}[\epsilon_0^2]}_{\text{irreducible error } (\text{Var}(\epsilon_0))} \quad \blacksquare$$

Problem 12

Consider the regression through the origin model:

$$y_i = \beta x_i + \epsilon_i$$

(a) The least squares estimate for β is

$$\begin{aligned}\hat{\beta} &= \underset{\beta}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2 \\ &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n}(y_i - \beta x_i)^2\end{aligned}$$

- taking the derivative with respect to β

$$\begin{aligned}\frac{d}{d\beta}(\cdot) &= \sum_{i=1}^{n} 2(y_i - \beta x_i)(-x_i) \\ &= -2\sum_{i=1}^{n}(y_i x_i - \beta x_i^2)\end{aligned}$$

- setting the derivative to zero gives

$$\begin{aligned}0 &= -2\sum_{i=1}^{n}(y_i x_i - \hat{\beta} x_i^2) \\ \sum_{i=1}^{n}\hat{\beta} x_i^2 &= \sum_{i=1}^{n} y_i x_i \\ \hat{\beta}\sum_{i=1}^{n} x_i^2 &= \sum_{i=1}^{n} y_i x_i\end{aligned}$$

$$\boxed{\hat{\beta} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}} \quad \blacksquare$$

(b) Assume $\epsilon_i \overset{iid}{\sim} P_\epsilon$ such that $\mathbb{E}[\epsilon_i] = 0$ and $\operatorname{Var}(\epsilon_i) = \sigma^2 < \infty$. Find the standard error of the estimate.

$$\begin{aligned}SE(\hat{\beta})^2 &= \frac{\operatorname{Var}(\hat{\beta})}{n} \\ &= \frac{1}{n}\mathbb{E}\left[\left(\frac{\sum x_i y_i}{\sum x_i^2} - \mathbb{E}\left[\frac{\sum x_i y_i}{\sum x_i^2}\right]\right)^2\right] \\ &= \frac{1}{n}\frac{1}{(\sum x_i^2)^2}\mathbb{E}\left[(\sum x_i y_i)^2 - 2(\sum x_i y_i)\mathbb{E}[\sum x_i y_i] + \mathbb{E}[\sum x_i y_i]^2\right] \\ &= \frac{1}{n}\frac{1}{(\sum x_i^2)^2}\left(\mathbb{E}\left[(\sum x_i(\beta x_i + \epsilon_i))^2\right] - 2\mathbb{E}\left[\sum x_i(\beta x_i + \epsilon_i)\right]^2 + \mathbb{E}\left[\sum x_i(\beta x_i + \epsilon_i)\right]^2\right) \\ &= \frac{1}{n(\sum x_i^2)^2}\left(\mathbb{E}\left[(\beta\sum x_i^2 + \sum x_i\epsilon_i)^2\right] - \mathbb{E}\left[\beta\sum x_i^2 + \sum x_i\epsilon_i\right]^2\right) \\ &= \frac{1}{n(\sum x_i^2)^2}\left(\mathbb{E}\left[\beta^2(\sum x_i^2)^2 + 2\beta(\sum x_i^2)(\sum x_i\epsilon_i) + (\sum x_i\epsilon_i)^2\right] - \left(\beta\sum x_i^2 + \mathbb{E}[\sum x_i\epsilon_i]\right)^2\right) \\ &= \frac{1}{n(\sum x_i^2)^2}\Big(\cancel{\beta^2(\sum x_i^2)^2} + \cancel{2\beta(\sum x_i^2)\mathbb{E}[\sum x_i\epsilon_i]} + \mathbb{E}\left[(\sum x_i\epsilon_i)^2\right] - \cancel{\beta^2(\sum x_i^2)^2} - \\ &\qquad - \cancel{2\beta(\sum x_i^2)\mathbb{E}[\sum x_i\epsilon_i]} - \mathbb{E}[\sum x_i\epsilon_i]^2\Big) \\ &= \frac{1}{n(\sum x_i^2)^2}\left(\mathbb{E}\left[(\sum x_i\epsilon_i)^2\right] - (\sum x_i \cancel{\mathbb{E}[\epsilon_i]}^{0})^2\right)\end{aligned}$$

$$= \frac{1}{n\left(\sum_{i=1}^{n} x_i^2\right)^2} \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \epsilon_i \epsilon_j\right] \qquad (x_i\epsilon_i)^2 \quad (x_i\epsilon_i x_2\epsilon_3)$$

$$= \frac{1}{n\left(\sum_{i=1}^{n} x_i^2\right)^2} \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j \mathbb{E}[\epsilon_i\epsilon_j]$$

$$= \frac{1}{n\left(\sum_{i=1}^{n} x_i^2\right)^2} \sum_{i=1}^{n} x_i \sum_{j=1}^{n} x_j \, \mathrm{Var}(\epsilon)$$

$$= \frac{\sigma^2\left(\sum_{i=1}^{n} x_i\right)^2}{n\left(\sum_{i=1}^{n} x_i^2\right)^2}$$

- therefore, the standard error of $\hat{\beta}$ is

$$\widehat{SE}(\hat{\beta}) = \frac{\sigma}{\sqrt{n}} \frac{\sum_{i=1}^{n} x_i}{\sum_{i=1}^{n} x_i^2} \quad \blacksquare$$

(c) Find conditions that guarantee the estimator is consistent.

- our estimated $\hat{\beta}(\hat{\beta}_n)$ is

$$\hat{\beta}_n = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$$

- given our y-values follow

$$y_i = \beta x_i + \epsilon_i$$

- we have

$$\hat{\beta}_n = \frac{\sum_{i=1}^{n} x_i(\beta x_i + \epsilon_i)}{\sum_{i=1}^{n} x_i^2}$$

$$= \frac{\sum_{i=1}^{n} \beta x_i^2 + \sum_{i=1}^{n} x_i \epsilon_i}{\sum_{i=1}^{n} x_i^2}$$

$$= \frac{\beta\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} x_i\epsilon_i}{\sum_{i=1}^{n} x_i^2}$$

$$= \beta + \frac{\sum_{i=1}^{n} x_i \epsilon_i}{\sum_{i=1}^{n} x_i^2}$$

- our estimator converges in expectation if the right term is zero

$$\mathbb{E}[\hat{\beta}_n] = \beta + \frac{\sum_{i=1}^{n} x_i \mathbb{E}[\epsilon_i]}{\sum_{i=1}^{n} x_i^2} \xrightarrow[\mathrm{Var}(\epsilon_i) < \infty]{\mathbb{E}[\epsilon_i] = 0} \beta + \frac{\sum_{i=1}^{n} x_i (0)}{\sum_{i=1}^{n} x_i^2} = \beta$$

Thus, our estimator is consistent if $y_i = \beta x_i + \epsilon_i$, $\mathbb{E}[\epsilon_i] = 0$, and $\mathrm{Var}(\epsilon_i) < \infty$. $\blacksquare$

Problem 7 (Chapter 10, Exercise 9)

```
import pandas as pd
from sklearn.cluster import AgglomerativeClustering


us_arrests_df = pd.read_csv("USArrests.csv", index_col=0)
us_arrests_df;
```

Problem 7(a)

```
hc_nonstd = AgglomerativeClustering(n_clusters=3, affinity="Euclidean", linkage="com
hc_nonstd.fit(us_arrests_df);
```

Problem 7(b)

```
for i in range(3):
  print("Cluster %d: " % i, end='')
  for state in us_arrests_df.index[hc_nonstd.labels_ == i]:
    print(state + ", ", end='')
  print()
```

```
 Cluster 0: Alabama, Alaska, Arizona, California, Delaware, Florida, Illinois, Lou
 Cluster 1: Connecticut, Hawaii, Idaho, Indiana, Iowa, Kansas, Kentucky, Maine, M
 Cluster 2: Arkansas, Colorado, Georgia, Massachusetts, Missouri, New Jersey, Okl
```

Problem 7(c)

```
from sklearn.preprocessing import StandardScaler

standard_scaler   = StandardScaler()
us_arrests_df_std = standard_scaler.fit_transform(us_arrests_df)

hc_std = AgglomerativeClustering(n_clusters=3, affinity="Euclidean", linkage="comple
hc_std.fit(us_arrests_df_std);
```

Problem 7(d)

```
for i in range(3):
  print("Cluster %d: " % i, end='')
  for state in us_arrests_df.index[hc_std.labels_ == i]:
    print(state + ", ", end='')
  print()
```

```
Cluster 0: Arkansas, Connecticut, Delaware, Hawaii, Idaho, Indiana, Iowa, Kansas
Cluster 1: Alabama, Alaska, Georgia, Louisiana, Mississippi, North Carolina, Sou
Cluster 2: Arizona, California, Colorado, Florida, Illinois, Maryland, Michigan,
```

```
us_arrests_df.describe()
```

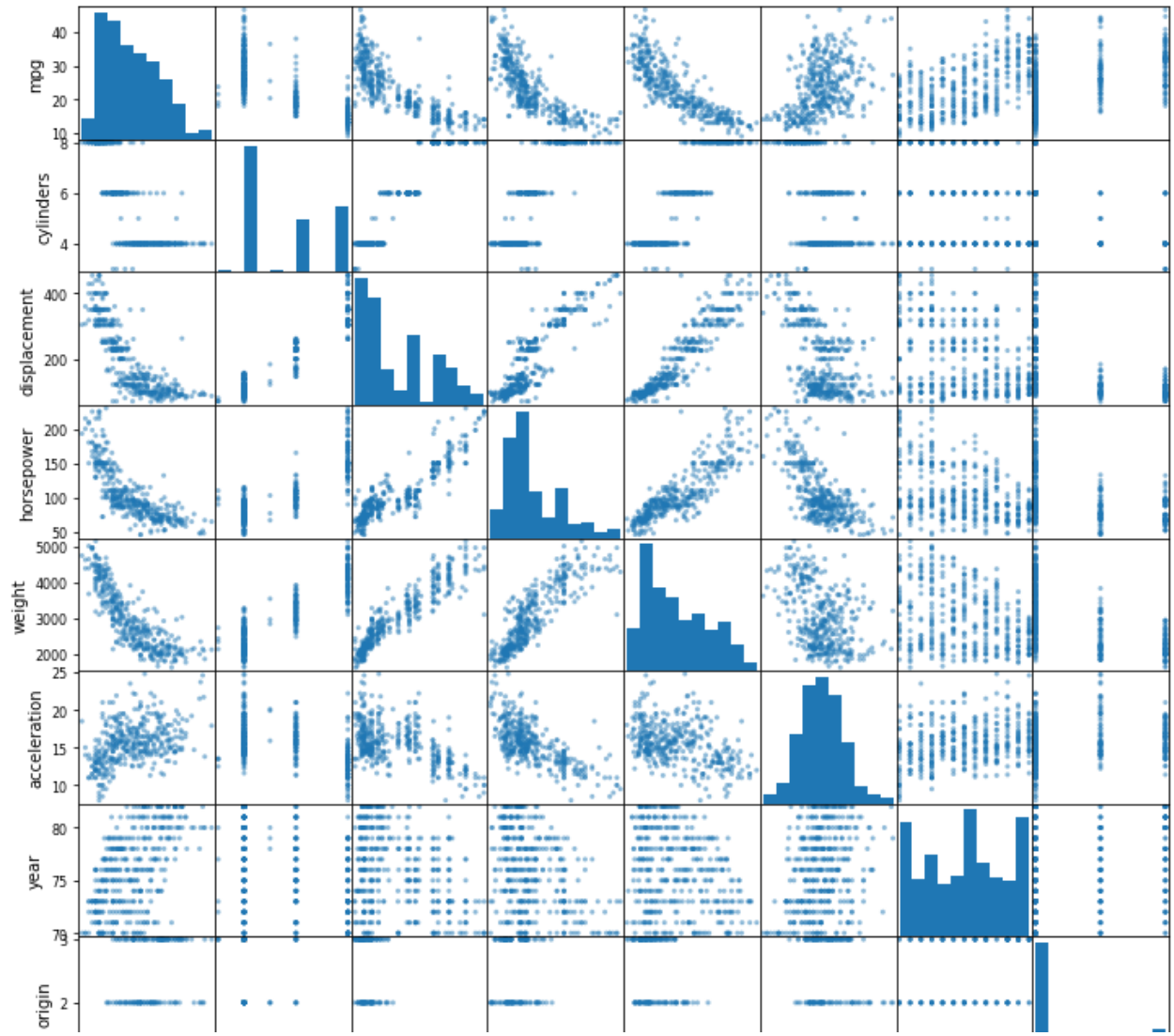
	Murder	Assault	UrbanPop	Rape
count	50.00000	50.000000	50.000000	50.000000
mean	7.78800	170.760000	65.540000	21.232000
std	4.35551	83.337661	14.474763	9.366385
min	0.80000	45.000000	32.000000	7.300000
25%	4.07500	109.000000	54.500000	15.075000
50%	7.25000	159.000000	66.000000	20.100000
75%	11.25000	249.000000	77.750000	26.175000
max	17.40000	337.000000	91.000000	46.000000

Problem 9 (Chapter 3, Exercise 9)

```
auto_df = pd.read_csv("Auto.csv", index_col=-1)
auto_df;
```

Problem 9(a)

```
pd.plotting.scatter_matrix(auto_df, figsize=(12,12));
```



Problem 9(b)

```
auto_df.corr()
```

	mpg	cylinders	displacement	horsepower	weight	acceleration
mpg	1.000000	-0.777618	-0.805127	-0.778427	-0.832244	0.423329
cylinders	-0.777618	1.000000	0.950823	0.842983	0.897527	-0.504683
displacement	-0.805127	0.950823	1.000000	0.897257	0.932994	-0.543800
horsepower	-0.778427	0.842983	0.897257	1.000000	0.864538	-0.689196
weight	-0.832244	0.897527	0.932994	0.864538	1.000000	-0.416839
acceleration	0.423329	-0.504683	-0.543800	-0.689196	-0.416839	1.000000
year	0.580541	-0.345647	-0.369855	-0.416361	-0.309120	0.290316
origin	0.565209	-0.568932	-0.614535	-0.455171	-0.585005	0.212746

Problem 9(c)

```
import statsmodels.api as sm

X = auto_df.loc[:, auto_df.columns != "mpg"]
y = auto_df["mpg"]

X = sm.add_constant(X)

linear_regression = sm.OLS(y, X)
linear_regression_results = linear_regression.fit()

print(linear_regression_results.summary())
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureW
  import pandas.util.testing as tm
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.821
Model:                            OLS   Adj. R-squared:                  0.818
Method:                 Least Squares   F-statistic:                     252.4
Date:                Tue, 06 Jul 2021   Prob (F-statistic):          2.04e-139
Time:                        20:33:59   Log-Likelihood:                -1023.5
No. Observations:                 392   AIC:                             2063.
Df Residuals:                     384   BIC:                             2095.
Df Model:                           7
Covariance Type:            nonrobust
================================================================================
                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
const          -17.2184      4.644     -3.707      0.000     -26.350      -8.087
cylinders       -0.4934      0.323     -1.526      0.128      -1.129       0.142
displacement     0.0199      0.008      2.647      0.008       0.005       0.035
horsepower      -0.0170      0.014     -1.230      0.220      -0.044       0.010
weight          -0.0065      0.001     -9.929      0.000      -0.008      -0.005
acceleration     0.0806      0.099      0.815      0.415      -0.114       0.275
year             0.7508      0.051     14.729      0.000       0.651       0.851
origin           1.4261      0.278      5.127      0.000       0.879       1.973
==============================================================================
Omnibus:                       31.906   Durbin-Watson:                   1.309
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               53.100
Skew:                           0.529   Prob(JB):                     2.95e-12
Kurtosis:                       4.460   Cond. No.                     8.59e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
[2] The condition number is large, 8.59e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

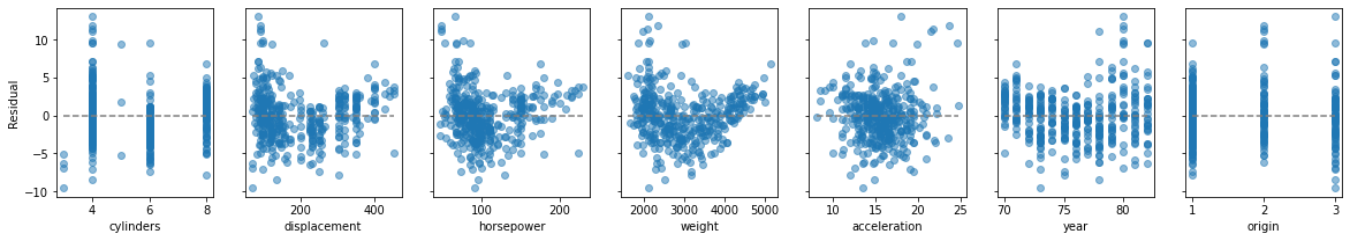
Problem 9(d)

```
auto_df.insert(8, "residuals", linear_regression_results.resid)
```

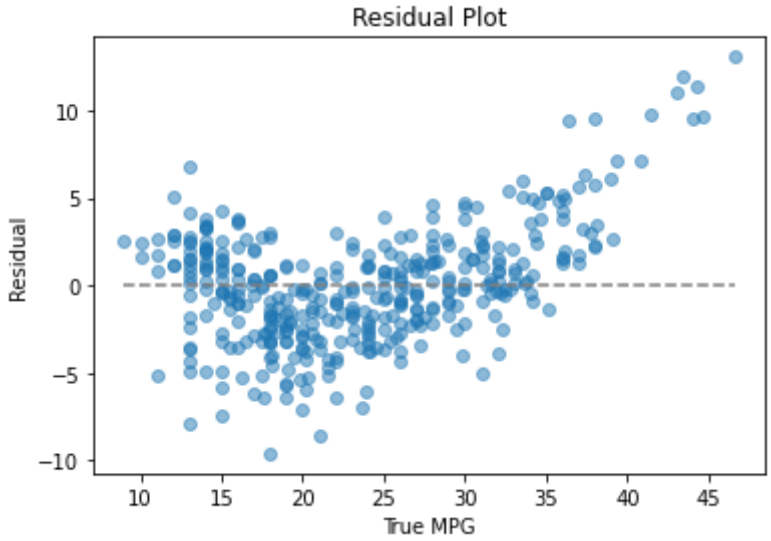
```
import numpy as np
import matplotlib.pyplot as plt

fig, ax = plt.subplots(1, 7, figsize=(20, 3), sharey=True)
ax[0].set_ylabel("Residual")
```

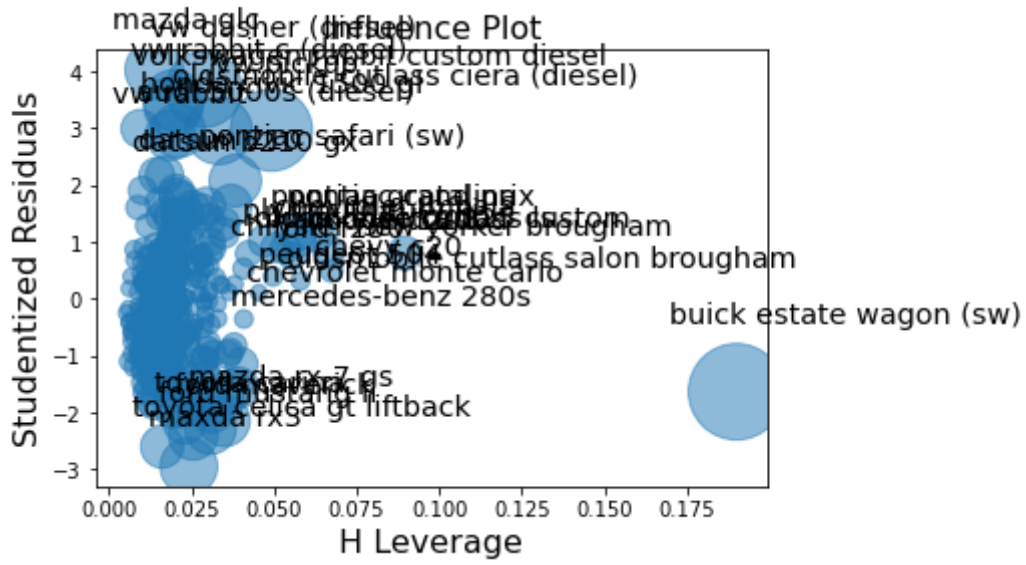
```
for i in np.arange(1,8):
  ax[i-1].plot([auto_df.iloc[:, i].min(), auto_df.iloc[:, i].max()], [0, 0], ls="--'
  ax[i-1].scatter(auto_df.iloc[:, i], auto_df["residuals"], alpha=0.5)
  ax[i-1].set_xlabel(auto_df.columns[i])
```



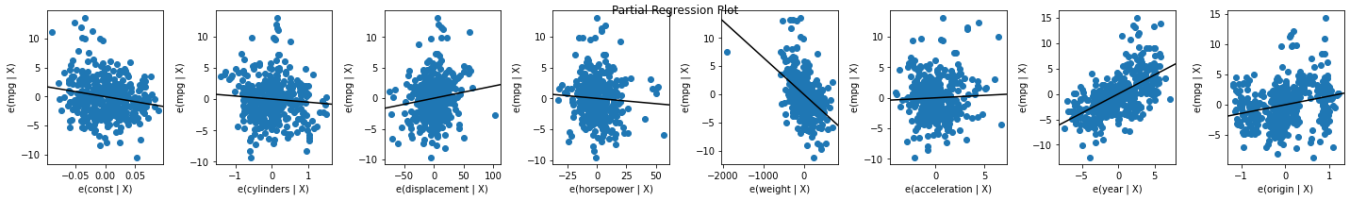

```
plt.plot(  [auto_df["mpg"].min(), auto_df["mpg"].max()], [0, 0], ls="--", c="gray")
plt.scatter(auto_df["mpg"], auto_df["residuals"], alpha=0.5)
plt.xlabel("True MPG")
plt.ylabel("Residual")
plt.title("Residual Plot");
```



```
sm.graphics.influence_plot(linear_regression_results, plot_alpha=0.5);
```




```
fig = plt.figure(figsize=(20,3))
sm.graphics.plot_partregress_grid(linear_regression_results, grid=(1,8), fig=fig);
```



Problem 9(e)

```
from statsmodels.formula.api import ols

linear_regression_w_interaction = ols(formula='mpg ~ cylinders + displacement + hors
                                      ' + weight + acceleration + year + origin + '
                                      'cylinders*displacement + cylinders*horsepower
                                      'cylinders*weight + horsepower*displacement',
linear_regression_w_interaction_results = linear_regression_w_interaction.fit()

print(linear_regression_w_interaction_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      mpg   R-squared:                       0.867
Model:                              OLS   Adj. R-squared:                  0.863
Method:                   Least Squares   F-statistic:                     224.9
```

```
Date:                Tue, 06 Jul 2021   Prob (F-statistic):          7.99e-159
Time:                        20:34:03   Log-Likelihood:                -965.98
No. Observations:                 392   AIC:                             1956.
Df Residuals:                     380   BIC:                             2004.
Df Model:                          11
Covariance Type:            nonrobust
================================================================================
                               coef    std err          t      P>|t|      [0.025
--------------------------------------------------------------------------------
Intercept                    7.5536      5.571      1.356      0.176      -3.401
cylinders                   -2.2358      1.147     -1.949      0.052      -4.491
displacement                -0.0178      0.026     -0.673      0.502      -0.070
horsepower                  -0.2092      0.050     -4.204      0.000      -0.307
weight                      -0.0082      0.002     -3.695      0.000      -0.013
acceleration                -0.1597      0.096     -1.668      0.096      -0.348
year                         0.7515      0.045     16.793      0.000       0.664
origin                       0.7382      0.262      2.815      0.005       0.223
cylinders:displacement      -0.0045      0.003     -1.390      0.165      -0.011
cylinders:horsepower         0.0103      0.010      1.022      0.307      -0.010
cylinders:weight             0.0007      0.000      2.311      0.021       0.000
horsepower:displacement      0.0003      0.000      3.047      0.002       0.000
==============================================================================
Omnibus:                       48.397   Durbin-Watson:                   1.543
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               97.726
Skew:                           0.684   Prob(JB):                     6.01e-22
Kurtosis:                       5.028   Cond. No.                     1.46e+06
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
[2] The condition number is large, 1.46e+06. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Problem 9(f)

```
linear_regression_w_transformation = ols(formula='mpg ~ cylinders + displacement + h
                                         ' + weight + acceleration + year + origin + '
                                         'np.power(cylinders, 2) + np.power(displacemer
                                         'np.power(horsepower, 2) + np.power(weight, 2)
                                         data=auto_df)
linear_regression_w_transformation_results = linear_regression_w_transformation.fit(

print(linear_regression_w_transformation_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    mpg   R-squared:                       0.865
Model:                            OLS   Adj. R-squared:                  0.861
Method:                 Least Squares   F-statistic:                     222.0
Date:                Tue, 06 Jul 2021   Prob (F-statistic):          6.80e-158
Time:                        20:34:03   Log-Likelihood:                -968.20
No. Observations:                 392   AIC:                             1960.
Df Residuals:                     380   BIC:                             2008.
```

```
Df Model:                           11
Covariance Type:            nonrobust
==============================================================================
                                 coef    std err          t      P>|t|      [0.02
------------------------------------------------------------------------------
Intercept                     4.3655      6.025      0.725      0.469      -7.48
cylinders                     0.0748      1.470      0.051      0.959      -2.81
displacement                 -0.0329      0.022     -1.487      0.138      -0.07
horsepower                   -0.1941      0.043     -4.564      0.000      -0.27
weight                       -0.0106      0.003     -4.084      0.000      -0.01
acceleration                 -0.1735      0.101     -1.726      0.085      -0.37
year                          0.7683      0.045     16.950      0.000       0.67
origin                        0.5859      0.269      2.180      0.030       0.05
np.power(cylinders, 2)        0.0279      0.119      0.235      0.814      -0.20
np.power(displacement, 2)  5.919e-05   3.87e-05      1.528      0.127    -1.7e-05
np.power(horsepower, 2)       0.0005      0.000      3.733      0.000       0.00
np.power(weight, 2)        1.038e-06   3.51e-07      2.957      0.003    3.48e-07
==============================================================================
Omnibus:                       39.818   Durbin-Watson:                   1.524
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               82.175
Skew:                           0.564   Prob(JB):                     1.43e-18
Kurtosis:                       4.939   Cond. No.                     4.59e+08
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
[2] The condition number is large, 4.59e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Problem 10 (Chapter 3, Exercise 14)

Problem 10(a)

```
data = pd.read_csv("ch3_q14.csv")
data
```

	x1	x2	y
0	0.265509	0.172565	3.032974
1	0.372124	0.124859	2.763146
2	0.572853	0.320539	2.923800
3	0.908208	0.341168	2.989404

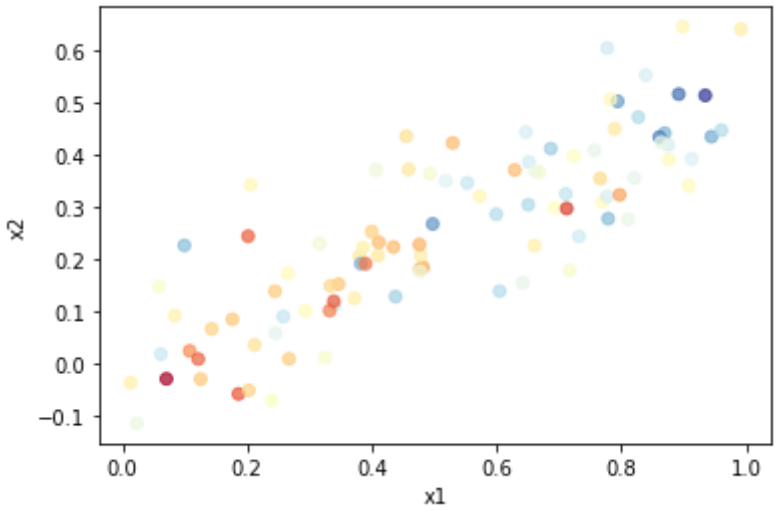
Problem 10(b)

95	0.797209	0.323573	1.691524

```
data.corr()
```

	x1	x2	y
x1	1.000000	0.835121	0.449845
x2	0.835121	1.000000	0.419917
y	0.449845	0.419917	1.000000

```
plt.scatter(data["x1"], data["x2"], c=data["y"], cmap="RdYlBu", alpha=0.7)
plt.xlabel("x1")
plt.ylabel("x2");
```



Problem 10(c)

```
X = data[["x1","x2"]]
y = data["y"]

X = sm.add_constant(X)
```

```
linear_regression_x1_x2 = sm.OLS(y, X)
linear_regression_x1_x2_results = linear_regression_x1_x2.fit()

print(linear_regression_x1_x2_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.209
Model:                            OLS   Adj. R-squared:                  0.193
Method:                 Least Squares   F-statistic:                     12.80
Date:                Tue, 06 Jul 2021   Prob (F-statistic):           1.16e-05
Time:                        20:34:04   Log-Likelihood:                -145.84
No. Observations:                 100   AIC:                             297.7
Df Residuals:                      97   BIC:                             305.5
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.1305      0.232      9.188      0.000       1.670       2.591
x1             1.4396      0.721      1.996      0.049       0.008       2.871
x2             1.0097      1.134      0.891      0.375      -1.240       3.260
==============================================================================
Omnibus:                        0.011   Durbin-Watson:                   2.081
Prob(Omnibus):                  0.995   Jarque-Bera (JB):                0.132
Skew:                          -0.005   Prob(JB):                        0.936
Kurtosis:                       2.823   Cond. No.                         14.3
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

Problem 10(d)

```
X = data["x1"]
y = data["y"]

X = sm.add_constant(X)

linear_regression_x1 = sm.OLS(y, X)
linear_regression_x1_results = linear_regression_x1.fit()

print(linear_regression_x1_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.202
Model:                            OLS   Adj. R-squared:                  0.194
Method:                 Least Squares   F-statistic:                     24.86
Date:                Tue, 06 Jul 2021   Prob (F-statistic):           2.66e-06
Time:                        20:34:04   Log-Likelihood:                -146.24
No. Observations:                 100   AIC:                             296.5
```

```
Df Residuals:                       98   BIC:                             301.7
Df Model:                            1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.1124      0.231      9.155      0.000       1.654       2.570
x1             1.9759      0.396      4.986      0.000       1.190       2.762
==============================================================================
Omnibus:                        0.041   Durbin-Watson:                   2.109
Prob(Omnibus):                  0.980   Jarque-Bera (JB):                0.012
Skew:                           0.003   Prob(JB):                        0.994
Kurtosis:                       2.947   Cond. No.                         4.82
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

Problem 10(e)

```
X = data["x2"]
y = data["y"]

X = sm.add_constant(X)

linear_regression_x2 = sm.OLS(y, X)
linear_regression_x2_results = linear_regression_x2.fit()

print(linear_regression_x2_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.176
Model:                            OLS   Adj. R-squared:                  0.168
Method:                 Least Squares   F-statistic:                     20.98
Date:                Tue, 06 Jul 2021   Prob (F-statistic):           1.37e-05
Time:                        20:34:04   Log-Likelihood:                -147.85
No. Observations:                 100   AIC:                             299.7
Df Residuals:                      98   BIC:                             304.9
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.3899      0.195     12.261      0.000       2.003       2.777
x2             2.8996      0.633      4.580      0.000       1.643       4.156
==============================================================================
Omnibus:                        0.491   Durbin-Watson:                   2.052
Prob(Omnibus):                  0.782   Jarque-Bera (JB):                0.625
Skew:                          -0.024   Prob(JB):                        0.731
Kurtosis:                       2.616   Cond. No.                         6.31
==============================================================================
```

```
Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

Problem 10(g)

```
data_g = data.append(pd.DataFrame([[0.1, 0.8, 6]], columns=["x1", "x2", "y"]), ignor
data_g
```

	x1	x2	y
0	0.265509	0.172565	3.032974
1	0.372124	0.124859	2.763146
2	0.572853	0.320539	2.923800
3	0.908208	0.341168	2.989404
4	0.201682	0.244143	0.989147
...	...	...	...
96	0.455274	0.436354	2.496664
97	0.410084	0.206782	2.626532
98	0.810870	0.276805	3.538661
99	0.604933	0.138406	4.271852
100	0.100000	0.800000	6.000000

101 rows × 3 columns

```
X = data_g[["x1","x2"]]
y = data_g["y"]

X = sm.add_constant(X)

linear_regression_x1_x2 = sm.OLS(y, X)
linear_regression_x1_x2_results = linear_regression_x1_x2.fit()

print(linear_regression_x1_x2_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.219
Model:                            OLS   Adj. R-squared:                  0.203
Method:                 Least Squares   F-statistic:                     13.72
Date:                Tue, 06 Jul 2021   Prob (F-statistic):           5.56e-06
Time:                        20:34:04   Log-Likelihood:                -149.07
No. Observations:                 101   AIC:                             304.1
Df Residuals:                      98   BIC:                             312.0
```

```
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.2267      0.231      9.624      0.000       1.768       2.686
x1             0.5394      0.592      0.911      0.365      -0.636       1.715
x2             2.5146      0.898      2.801      0.006       0.733       4.296
==============================================================================
Omnibus:                        0.608   Durbin-Watson:                   1.992
Prob(Omnibus):                  0.738   Jarque-Bera (JB):                0.708
Skew:                          -0.024   Prob(JB):                        0.702
Kurtosis:                       2.593   Cond. No.                         11.1
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

```
X = data_g["x1"]
y = data_g["y"]

X = sm.add_constant(X)

linear_regression_x1 = sm.OLS(y, X)
linear_regression_x1_results = linear_regression_x1.fit()

print(linear_regression_x1_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.156
Model:                            OLS   Adj. R-squared:                  0.148
Method:                 Least Squares   F-statistic:                     18.33
Date:                Tue, 06 Jul 2021   Prob (F-statistic):           4.29e-05
Time:                        20:34:04   Log-Likelihood:                -152.96
No. Observations:                 101   AIC:                             309.9
Df Residuals:                      99   BIC:                             315.1
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.2569      0.239      9.445      0.000       1.783       2.731
x1             1.7657      0.412      4.282      0.000       0.947       2.584
==============================================================================
Omnibus:                        2.643   Durbin-Watson:                   1.957
Prob(Omnibus):                  0.267   Jarque-Bera (JB):                2.042
Skew:                           0.245   Prob(JB):                        0.360
Kurtosis:                       3.496   Cond. No.                         4.77
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

```
X = data_g["x2"]
```

```
y = data_g["y"]

X = sm.add_constant(X)

linear_regression_x2 = sm.OLS(y, X)
linear_regression_x2_results = linear_regression_x2.fit()

print(linear_regression_x2_results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.212
Model:                            OLS   Adj. R-squared:                  0.204
Method:                 Least Squares   F-statistic:                     26.66
Date:                Tue, 06 Jul 2021   Prob (F-statistic):           1.25e-06
Time:                        20:34:04   Log-Likelihood:                -149.49
No. Observations:                 101   AIC:                             303.0
Df Residuals:                      99   BIC:                             308.2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          2.3451      0.191     12.264      0.000       1.966       2.725
x2             3.1190      0.604      5.164      0.000       1.921       4.318
==============================================================================
Omnibus:                        0.837   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.658   Jarque-Bera (JB):                0.862
Skew:                          -0.044   Prob(JB):                        0.650
Kurtosis:                       2.556   Cond. No.                         6.05
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
```

✓ 0s completed at 3:34 PM