DSC 40A - Homework 03

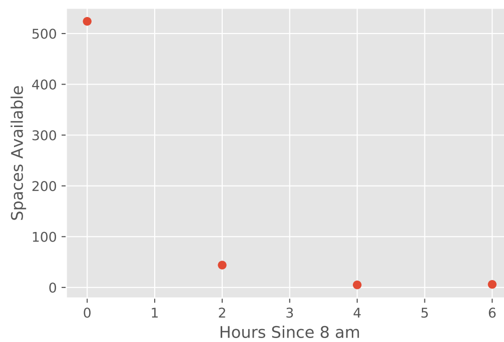
Due: Friday, January 31, 2020

Write your solutions to the following problems by either typing them up or handwriting them on another piece of paper. Unless otherwise noted by the problem's instructions, show your work or provide some justification for your answer. Homeworks are due via Gradescope on Friday afternoon at 5:00 p.m.

Problem 1.

The table below and the accompanying plot show the total number of "S" parking spaces available on the UCSD campus at various times on a typical Tuesday:[1]

Hours Since 8 am	Spaces Available
0	524
2	44
4	5
6	6



a) Use least squares regression to find a prediction rule of the form $H(x) = w_1 x + w_0$ for the number of spaces available. The variable x represents the number of hours since 8 am. *Hint*: you can check whether your answer is reasonable by plotting.

Solution: We'll use the familiar formulas for the slope and intercept of the linear least squares regression line:

$$w_1 = \frac{\sum\limits_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum\limits_{i=1}(x_i - \bar{x})^2}$$

$$w_0 = \bar{y} - b_1 \bar{x}$$

In this case, the x_i are the hours since 8 am, and the y_i are the number of spaces available. Therefore: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = 3$ and $\bar{y} = \sum_{i=1}^{n} y_i = 144.75$.

$x_i - \bar{x}$	$y_i - \bar{y}$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
-3.0	379.25	-1137.75	9.0
-1.0	-100.75	100.75	1.0
1.0	-139.75	-139.75	1.0
3.0	-138.75	-416.25	9.0
	sum:	-1593.00	20.0

[1] Parking availability for January 14, 2020 was scraped from the UCSD transportation office website. Lot P701 was excluded.

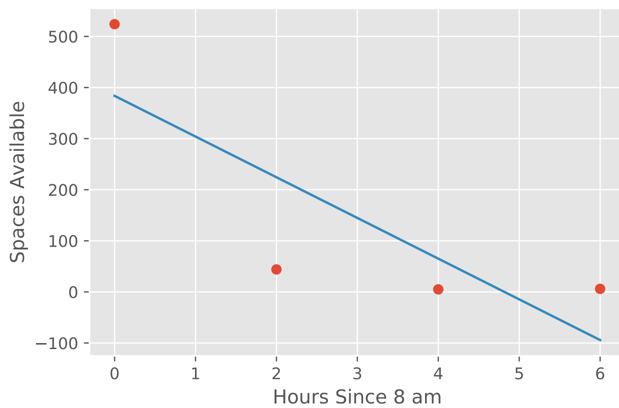
This gives $w_1 = -1593/20 = -79.65$. Plugging this into the formula for w_0, we find:

$$w_1 = \bar{y} - w_1\bar{x} = 383.7$$

So our prediction rule is:

$$H(x) = -79.65x + 383.7$$

The plot below shows our linear prediction rule:



You might like to know that `numpy` has a function which performs least squares regression. It is called `np.polyfit`. Here's a demo:

```
>>> x = [0, 2, 4, 6]
>>> y = [524, 44, 5, 6]
>>> import numpy as np
>>> np.polyfit(x, y, deg=1)
array([-79.65, 383.7 ])
```

b) Use your prediction rule from above to predict the number of parking spots available at 9 am. Use it again to predict the number of parking spaces available at 4 pm. Do you believe that your prediction rule makes good predictions? Why or why not?

Solution: Our prediction rule was $H(x) = -79.65x + 383.7$, where x is the number of hours past 8am. Our prediction for 9 am is found by calculating $H(1)$:

$$H(1) = -79.65 + 383.7 = 304.05$$

Since 4 pm is 8 hours after 8 am, our prediction for 4 pm is:

$$H(8) = -79.65 \cdot 8 + 383.7 = -253.5$$

It seems that our linear prediction rule does not make good predictions. For one, it is predicting a negative number of parking spaces at 4 pm, which is not possible. Second, even when its predictions are positive, they don't match the data particularly well.

c) Use least squares regression to find a prediction rule of the form $H(x) = \frac{w_1}{x+1} + w_0$.

Solution: We wish to fit a function of the form

$$H(x) = \frac{w_1}{x+1} + w_0.$$

If we define $z(x) = 1/(x+1)$, then our prediction rule becomes:

$$H(x) = w_1 z(x) + w_0.$$

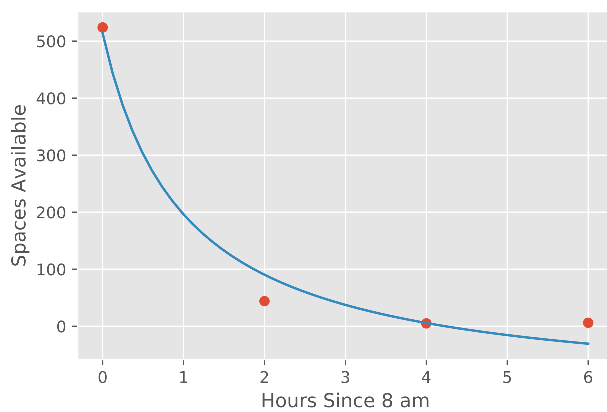
This is a linear function of w_1 and w_0, so our formulas apply. We start by making a table of the various quantities involved:

z_i	$z_i - \bar{z}$	$y_i - \bar{y}$	$(z_i - \bar{z})(y_i - \bar{y})$	$(z_i - \bar{z})^2$
1.00	0.58	379.25	220.33	0.34
0.33	-0.09	-100.75	8.64	0.01
0.20	-0.22	-139.75	30.61	0.05
0.14	-0.28	-138.75	38.32	0.08
		sum:	297.90	0.47

Here, $z_i = z(x_i) = 1/(x_i+1)$. Using our formulas: $w_1 = 297.90/0.47 = 635.02$ and $w_0 = -121.35$. So our prediction rule is:

$$H(x) = \frac{635.02}{x+1} - 121.35$$

Here is our prediction rule in action:



It certainly looks better than the linear prediction rule, but it still predicts negative parking spaces.

d) It looks like the number of parking spaces decreases exponentially as the day goes on. A better prediction rule might be $H_{\exp}(x) = 524 \cdot e^{-wx}$, where w is a parameter that we want to learn from data. Write down the general formula for computing the mean squared error of this prediction rule as a function of w, using x_i for the hours since 8 a.m. and y_i for the number of parking spaces, and n for the number of data points.

Hint: the formula for the mean squared error of a linear prediction rule, $H(x) = w_1 x + w_0$, is: $R(w_1, w_0) = \frac{1}{n} \sum_{i=1}^{n} \left((w_1 x_i + w_0) - y_i\right)^2$.

Solution: In general, the mean squared error of a prediction rule $H(x)$ is:

$$R(H) = \frac{1}{n}\sum_{i=1}^{n}\left(H(x_i) - y_i\right)^2$$

in this case, $H(x) = 524 \cdot e^{-wx}$. Therefore, the MSE of this prediction rule is:

$$R(w) = \frac{1}{n}\sum_{i=1}^{n}\left(524 \cdot e^{-wx_i} - y_i\right)^2$$

Bonus (+3 points) Unlike the case of linear prediction rules, there's no formula for the minimizer of R_{exp}. Instead, we have to minimize the mean squared error numerically by using gradient descent or some other method. The Python package `scipy` has a function called `scipy.optimize.minimize` which numerically minimizes a function. Use it to find the value of w which minimizes the mean squared error of the exponential prediction rule given the data in the table above.

Hint: There is a short demo notebook on using SciPy for Numerical Optimization Demo. It will show you how to use `scipy.optimize.minimize`, and you can perform your analysis in the notebook itself.

Solution: Here is part of the documentation of the `scipy.optimize.minimize` function:

scipy.optimize.**minimize**(*fun*, *x0*, *args=()*, *method=None*, *jac=None*, *hess=None*, *hessp=None*, *bounds=None*, *constraints=()*, *tol=None*, *callback=None*, *options=None*) [source]

Minimization of scalar function of one or more variables.

Parameters:

fun : *callable*

The objective function to be minimized.

```
fun(x, *args) -> float
```

where x is an 1-D array with shape (n,) and *args* is a tuple of the fixed parameters needed to completely specify the function.

x0 : *ndarray, shape (n,)*

Initial guess. Array of real elements of size (n,), where 'n' is the number of independent variables.

To use it, we must define a function and specify a starting location. Here is a Python function that computes the mean squared error of the exponential prediction rule:

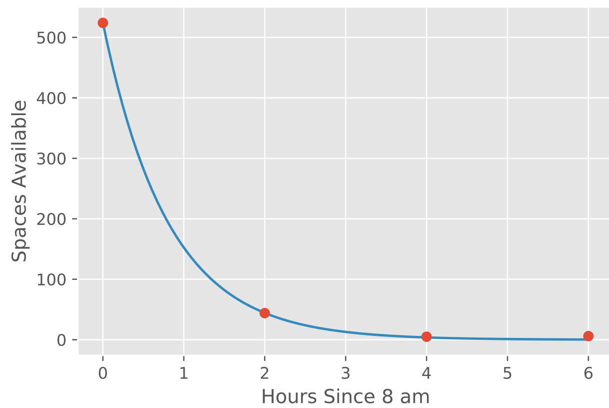
```python
x = np.array([0, 2, 4, 6])
y = np.array([524, 44, 5, 6])

def mse(w):
    return ((524 * np.exp(-w*x) - y)**2).mean()
```

Running `scipy.optimize.minimize(mse, 1)` finds an optimizer of $w \approx 1.235$. Our prediction rule is therefore:

$$H(x) = 524 \cdot e^{-1.235\,x}$$

The prediction rule is plotted below.



Looks pretty good!

One advantage of this prediction rule is that it never predicts a negative number of available spaces. As $x \to \infty$, our prediction will go to zero. Of course, we do expect the number of parking spaces to increase once again as people go home for the night, so this model only works for the day time. If we had data for before 8 am and in the evening, we might expect to see two sigmoid functions.

Problem 2.

In lecture, we derived the least squares solutions for linear prediction rules $H(x) = w_1 x + w_0$. They were:

$$\begin{aligned} w_1 &= \frac{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})^2} \\ w_0 &= \bar{y} - b_1 \bar{x} \end{aligned}$$

Where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

You may see these solutions written in various equivalent forms. In this problem, we'll derive another form that you may find useful in solving other problems.

a) Show that $\sum_{i=1}^{n}(x_i - \bar{x}) = 0$.

Solution: We begin by breaking the sum apart:

$$\sum_{i=1}^{n}(x_i - \bar{x}) = \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \bar{x}$$

Since $\bar{x}$ does not depend upon i, we can pull it out in front of its summation:

$$\begin{aligned} &= \sum_{i=1}^{n} x_i - \bar{x}\sum_{i=1}^{n} 1 \\ &= \sum_{i=1}^{n} x_i - n\bar{x} \end{aligned}$$

Using the definition of $\bar{x} = \frac{1}{n}\sum x_i$:

$$\begin{aligned} &= \sum_{i=1}^{n} x_i - n \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} x_i \\ &= \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i \\ &= 0 \end{aligned}$$

b) Use the result of the previous part to show that

$$w_1 = \frac{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})y_i}{\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

is equivalent to the formula for w_1 that was given in lecture.

Solution: The only difference between this new formula and the familiar one is that $\sum(x_i - \bar{x})(y_i - \bar{y})$ is replaced by $\sum(x_i - \bar{x})y_i$, so we'll show that these two are equal.

We'll start with $\sum(x_i - \bar{x})(y_i - \bar{y})$. We want to use the result of the previous part, which requires us to get $\sum(x_i - \bar{x})$ by itself. To do so, we'll try expanding the product in the summand:

$$\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) = \sum_{i=1}^{n}\left[(x_i - \bar{x})y_i - (x_i - \bar{x})\bar{y}\right]$$

We recognize the first term in the summand, $(x_i - \bar{x})y_i$, as the one we want to be left with; can we get rid of the second term somehow? We'll split the summand:

$$= \sum_{i=1}^{n}(x_i - \bar{x})y_i - \sum_{i=1}^{n}(x_i - \bar{x})\bar{y}$$

Now, $\bar{y}$ is a constant as far as the summation is concerned, so we can move it in front:

$$= \sum_{i=1}^{n}(x_i - \bar{x})y_i - \bar{y}\sum_{i=1}^{n}(x_i - \bar{x})$$

And now we've isolated $\sum(x_i - \bar{x})$ as we wanted. We can get rid of the entire second term, since it is zero:

$$= \sum_{i=1}^{n}(x_i - \bar{x})y_i$$

Since $\sum(x_i - \bar{x})(y_i - \bar{y}) = \sum(x_i - \bar{x})y_i$, we have our result:

$$\frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}(x_i - \bar{x})^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})y_i}{\sum_{i=1}(x_i - \bar{x})^2}$$

Problem 3.

A *Boolean feature* is one that is either true or false. For example, when predicting the price of a car, a useful feature might be whether or not the car has an automatic transmission. We can perform least squares regression with Boolean features by "encoding" true and false as numbers: a common choice is to encode true as 1 and false as 0.

In this problem, suppose we have a data set $(x_1, y_1), \ldots, (x_n, y_n)$ of n cars, where the feature x_i is either 1 or 0 (has automatic transmission, or does not) and where y_i is the price of the car. Furthermore, suppose that n_1 of the cars have automatic transmissions, while n_0 do not. Assume for simplicity that the data are sorted so that the first n_0 cars do not have automatic transmissions while the rest do, so that $x_1, \ldots, x_{n_0} = 0$ and $x_{n_0+1}, \ldots, x_n = 1$.

a) Show that $\bar{x} = \frac{n_1}{n}$.

Solution: We know that $\bar{x}$ is the average of the x_i's

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

We know that the first n_0 of the x_i's are zero, and last n_1 are one. So let's break the sum into two sums: one over the first n_0 terms, and the second over the remaining:

$$= \frac{1}{n}\left(\sum_{i=1}^{n_0} x_i + \sum_{i=n_0+1}^{n} x_i\right)$$

Each term in the first sum is zero, and so the sum is zero. Each term in the second sum is one:

$$\begin{aligned}
&= \frac{1}{n}\left(\sum_{i=1}^{n_0} 0 + \sum_{i=n_0+1}^{n} 1\right) \\
&= \frac{1}{n}(0 + n_1) \\
&= \frac{n_1}{n}
\end{aligned}$$

b) Show that $\displaystyle\sum_{i=1}^{n}(x_i - \bar{x})y_i = \frac{n_0}{n}\sum_{i=n_0+1}^{n} y_i - \frac{n_1}{n}\sum_{i=1}^{n_0} y_i$

Solution: We start by using the fact that $\bar{x} = n_1/n$. So:

$$\sum_{i=1}^{n}(x_i - \bar{x})y_i = \sum_{i=1}^{n}(x_i - n_1/n)y_i$$

I also know that the first n_0 of the x_i are zero, while the rest are one. So we'll once again split the summation into two summations:

$$\begin{aligned}
&= \sum_{i=1}^{n_0}(x_i - n_1/n)y_i + \sum_{i=n_0+1}^{n}(x_i - n_1/n)y_i \\
&= \sum_{i=1}^{n_0}(0 - n_1/n)y_i + \sum_{i=n_0+1}^{n}(1 - n_1/n)y_i \\
&= \sum_{i=1}^{n_0}(-n_1/n)y_i + \sum_{i=n_0+1}^{n}(1 - n_1/n)y_i
\end{aligned}$$

Switching the order of the result to match the target expression:

$$\begin{aligned}
&= \sum_{i=n_0+1}^{n}(1 - n_1/n)y_i + \sum_{i=1}^{n_0}(-n_1/n)y_i \\
&= (1 - n_1/n)\sum_{i=n_0+1}^{n} y_i - \frac{n_1}{n}\sum_{i=1}^{n_0} y_i
\end{aligned}$$

This looks very similar to the target expression, but is $(1 - n_1/n) = n_0/n$? If we rewrite 1 and n/n, we get $(1 - n_1/n) = (n - n_1)/n = n_0/n$, so:

$$= \frac{n_0}{n}\sum_{i=n_0+1}^{n} y_i - \frac{n_1}{n}\sum_{i=1}^{n_0} y_i$$

c) Suppose least squares regression is used to fit a linear prediction rule $H(x) = w_1 x + w_0$ to this data. Show that the prediction $H(0)$ is the mean price of cars without automatic transmissions ($\frac{1}{n_0}\sum_{i=1}^{n_0} y_i$) and the prediction $H(1)$ is the mean price of cars with automatic transmissions ($\frac{1}{n_1}\sum_{i=n_0+1}^{n} y_i$).

Solution: In order to make predictions, we need to first find the slope w_1 and the intercept w_0. Recognize that the expression we found in the last step is the numerator of

$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})y_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

(This is an alternative formula for w_1 that you derived in another problem).

We'll now compute the denominator and simplify to find an expression for w_1. We have:

$$\begin{aligned}
\sum_{i=1}^{n}(x_i - \bar{x})^2 &= \sum_{i=1}^{n_0}(x_i - \bar{x})^2 + \sum_{i=n_0+1}^{n_1}(x_i - \bar{x})^2 \\
&= \sum_{i=1}^{n_0}(0 - \bar{x})^2 + \sum_{i=n_0+1}^{n_1}(1 - \bar{x})^2
\end{aligned}$$

Substituting $\bar{x} = n_1/n$:

$$
\begin{aligned}
&= \sum_{i=1}^{n_0}(0 - n_1/n)^2 + \sum_{i=n_0+1}^{n_1}(1 - n_1/n)^2 \\
&= \sum_{i=1}^{n_0}(n_1/n)^2 + \sum_{i=n_0+1}^{n_1}(1 - n_1/n)^2
\end{aligned}
$$

We can simplify $1 - n_1/n$ by noting that it is $(n - n_1)/n = n_0/n$:

$$
\begin{aligned}
&= \sum_{i=1}^{n_0}(n_1/n)^2 + \sum_{i=n_0+1}^{n_1}(n_0/n)^2 \\
&= n_0(n_1/n)^2 + n_1(n_0/n)^2 \\
&= \frac{n_0 n_1^2}{n^2} + \frac{n_0^2 n_1}{n^2} \\
&= \frac{n_0 n_1(n_1 + n_0)}{n^2} \\
&= \frac{n_0 n_1 n}{n^2} \\
&= \frac{n_0 n_1}{n}
\end{aligned}
$$

That gives us:

$$
\begin{aligned}
w_1 &= \frac{\sum_{i=1}^{n}(x_i - \bar{x})y_i}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \\
&= \frac{\frac{n_0}{n}\sum_{i=n_0+1}^{n} y_i - \frac{n_1}{n}\sum_{i=1}^{n_0} y_i}{\frac{n_0 n_1}{n}} \\
&= \frac{\dfrac{n_0}{n}\displaystyle\sum_{i=n_0+1}^{n} y_i - \dfrac{n_1}{n}\displaystyle\sum_{i=1}^{n_0} y_i}{\dfrac{n_0 n_1}{n}} \\
&= \left(\frac{n_0}{n}\sum_{i=n_0+1}^{n} y_i - \frac{n_1}{n}\sum_{i=1}^{n_0} y_i\right) \cdot \frac{n}{n_0 n_1} \\
&= \frac{1}{n_1}\sum_{i=n_0+1}^{n} y_i - \frac{1}{n_0}\sum_{i=1}^{n_0} y_i
\end{aligned}
$$

So the slope is just the mean price of cars with automatic transmissions, minus the mean price of cars without automatic transmissions.

Recall that $w_0 = \bar{y} - w_1 \bar{x}$. So

$$\begin{aligned}
w_0 &= \bar{y} - \left(\frac{1}{n_1} \sum_{i=n_0+1}^{n} y_i - \frac{1}{n_0} \sum_{i=1}^{n_0} y_i \right) \bar{x} \\
&= \frac{1}{n} \sum_{i=1}^{n} y_i - \left(\frac{1}{n_1} \sum_{i=n_0+1}^{n} y_i - \frac{1}{n_0} \sum_{i=1}^{n_0} y_i \right) \cdot \frac{n_1}{n} \\
&= \frac{1}{n} \sum_{i=1}^{n} y_i - \left(\frac{1}{n} \sum_{i=n_0+1}^{n} y_i - \frac{n_1}{n \cdot n_0} \sum_{i=1}^{n_0} y_i \right) \\
&= \left(\frac{1}{n} \sum_{i=1}^{n} y_i - \frac{1}{n} \sum_{i=n_0+1}^{n} y_i \right) + \frac{n_1}{n \cdot n_0} \sum_{i=1}^{n_0} y_i \\
&= \frac{1}{n} \sum_{i=1}^{n_0} y_i + \frac{n_1}{n \cdot n_0} \sum_{i=1}^{n_0} y_i \\
&= \left(\frac{1}{n} + \frac{n_1}{n \cdot n_0} \right) \sum_{i=1}^{n_0} y_i \\
&= \left(\frac{n_0}{n \cdot n_0} + \frac{n_1}{n \cdot n_0} \right) \sum_{i=1}^{n_0} y_i \\
&= \frac{n}{n \cdot n_0} \sum_{i=1}^{n_0} y_i \\
&= \frac{1}{n_0} \sum_{i=1}^{n_0} y_i
\end{aligned}$$

Since $H(0) = w_0$, this is our predicted price for cars without automatic transmissions (it is the average price of cars without automatic transmissions).

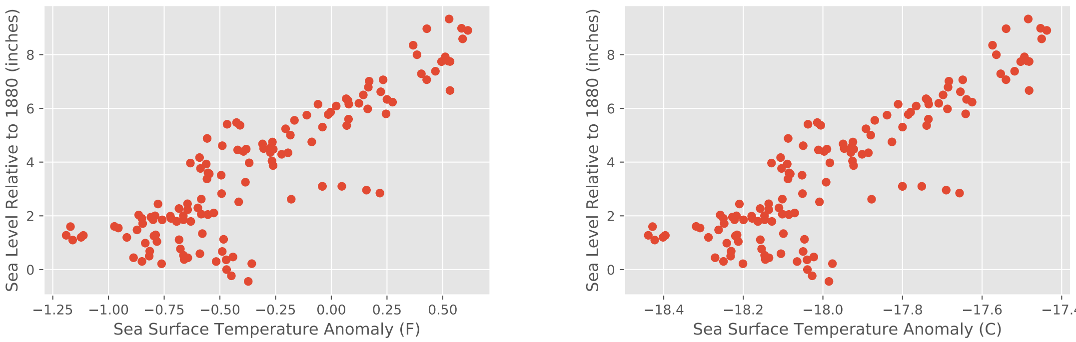
Now we'll compute $H(1)$:

$$\begin{aligned}
H(1) &= w_1 + w_0 \\
&= \left(\frac{1}{n_1} \sum_{i=n_0+1}^{n} y_i - \frac{1}{n_0} \sum_{i=1}^{n_0} y_i \right) + \frac{1}{n_0} \sum_{i=1}^{n_0} y_i \\
&= \frac{1}{n_1} \sum_{i=n_0+1}^{n} y_i
\end{aligned}$$

This is just the average price of cars with automatic transmissions.

Problem 4.

The figures below show the difference in sea level between today and 1880 plotted against the anomaly in sea surface temperature (relative to the 1971-2000 average).[2] Both plots are exactly the same except for the units used to measure temperature; the plot on the left uses Fahrenheit, while the plot on the right uses Celsius.



Suppose a linear prediction $H_1(x) = w_1 x + w_0$ rule is fit to the data on the left using least squares. And suppose that another prediction rule $H_2(x) = b_1 x + b_0$ is fit to the data on the right.

a) Are the slopes of the two lines the same? I.e., is $b_1 = w_1$?

Solution: No, the slopes are not the same. Although the two plots look identical, the scale used for their x-axis is different. Roughly, a line fit to the data in which temperatures are measured in Fahrenheit will increase from around 0 inches at -1.25 degrees to 10 inches at 0.75 degrees. This is a slope of

$$\frac{\Delta y}{\Delta x} = \frac{10}{2} = 5$$

On the other hand, a line fit to the data in which temperatures are in Celsius will increase from around 0 inches at -18.5 degrees to 10 inches at -17.4 degrees. That's a slope of

$$\frac{\Delta y}{\Delta x} = \frac{10}{1.1} \approx 9.09$$

b) Suppose H_1 is used to make a prediction for some temperature anomaly t_F measured in Fahrenheit, and H_2 is used to make a prediction for an anomaly t_C, where t_C is the temperature t_F, but converted to Celsius. Are the two predictions the same? Prove your answer.

Solution: Our goal is to show that $H_2(t_C) = H_1(t_F)$. Recall the formula for converting from Fahrenheit to Celsius:

$$t_C = \frac{5}{9} \cdot (t_F - 32)$$

To calculate $H_2(t_C) = b_1 t_C + b_0$, we'll need b_1 and b_0. Suppose our data set is (x_i, y_i), with x_i is measured in Fahrenheit. We have another data set (z_i, y_i), where $z_i = (5/9)(x_i - 32)$ is the temperature in Celsius. We know from a previous homework that the average of the converted temperatures is just the converted average. That is, $\bar{z} = \frac{5}{9}(\bar{x} - 32)$, so:

[2] The data were scraped from the EPA website.

$$
\begin{aligned}
b_1 &= \frac{\sum_{i=1}^n (z_i - \bar{z})(y_i - \bar{y})}{\sum_{i=1}^n (z_i - \bar{z})^2} \\
&= \frac{\sum_{i=1}^n \left(\frac{5}{9}(x_i - 32) - \frac{5}{9}(\bar{x} - 32)\right)(y_i - \bar{y})}{\sum_{i=1}^n \left(\frac{5}{9}(x_i - 32) - \frac{5}{9}(\bar{x} - 32)\right)^2}
\end{aligned}
$$

In an effort to get this to look more like w_0, we'll group x_i and $\bar{x}$:

$$
\begin{aligned}
&= \frac{\sum_{i=1}^n \left(\frac{5}{9}(x_i - \bar{x}) - \frac{5}{9} \cdot 32 + \frac{5}{9} \cdot 32\right)(y_i - \bar{y})}{\sum_{i=1}^n \left(\frac{5}{9}(x_i - \bar{x}) - \frac{5}{9} \cdot 32 + \frac{5}{9} \cdot 32\right)^2} \\
&= \frac{\sum_{i=1}^n \left(\frac{5}{9}(x_i - \bar{x})\right)(y_i - \bar{y})}{\sum_{i=1}^n \left(\frac{5}{9}(x_i - \bar{x})\right)^2} \\
&= \frac{\frac{5}{9}\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left(\frac{5}{9}\right)^2 \sum_{i=1}^n (x_i - \bar{x})^2} \\
&= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left(\frac{5}{9}\right) \sum_{i=1}^n (x_i - \bar{x})^2} \\
&= \frac{9}{5} \cdot w_0
\end{aligned}
$$

Remember that in part (a) we argued that the slope of the line fitting the Celsius data was about 9, while the slope of the Fahrenheit data was around 5. This explains why.

We then have

$$
\begin{aligned}
b_0 &= \bar{y} - b_1 \bar{z} \\
&= \bar{y} - b_1 \left(\frac{5}{9}(\bar{x} - 32)\right) \\
&= \bar{y} - \frac{9}{5} w_1 \cdot \left(\frac{5}{9}(\bar{x} - 32)\right) \\
&= \bar{y} - w_1(\bar{x} - 32) \\
&= \bar{y} - w_1 \bar{x} + 32 w_1 \\
&= w_0 + 32 w_1
\end{aligned}
$$

The prediction for t_C is:

$$
\begin{aligned}
H_2(t_C) &= b_1 t_C + b_0 \\
&= b_1 \left(\frac{5}{9}(t_F - 32)\right) + b_0 \\
&= \frac{9}{5} \cdot w_1 \left(\frac{5}{9}(t_F - 32)\right) + w_0 + 32 w_1 \\
&= w_1 (t_F - 32) + w_0 + 32 w_1 \\
&= w_1 t_F + w_0 - 32 w_1 + 32 w_1 \\
&= w_1 t_F + w_0 \\
&= H_1(t_F)
\end{aligned}
$$

Problem 5.

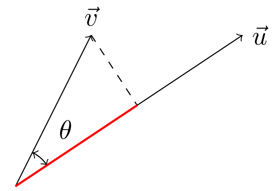
Looking Ahead. We will soon need to remember the key properties of the dot product. This question is meant to help you remember them.

Recall from your class on vector algebra that one way to define the dot product of two vectors, $\vec{u}$ and $\vec{v}$, is:

$$\vec{u} \cdot \vec{v} = \|\vec{u}\| \|\vec{v}\| \cos\theta,$$

where $\|\vec{u}\|$ is the length of the vector $\vec{u}$, $\|\vec{v}\|$ is the length of $\vec{v}$, and θ is the angle between the two vectors.

Two vectors $\vec{u}$ and $\vec{v}$ are shown below.



Argue that the length of the red segment is $(\vec{u} \cdot \vec{v})/\|\vec{u}\|$.

Solution: We see a right triangle whose hypotenuse has length $\|\vec{y}\|$. The length of the adjacent side is given by the length of the hypotenuse by the cosine of θ:

$$\|y\| \cos\theta$$

Since $\vec{u} \cdot \vec{v} = \|\vec{u}\| \|\vec{v}\| \cos\theta$, the length of the red segment must be $(\vec{u} \cdot \vec{v})/\|\vec{u}\|$.