پیشنهاد: با سوال‌های قسمت Alignment شروع کنید!

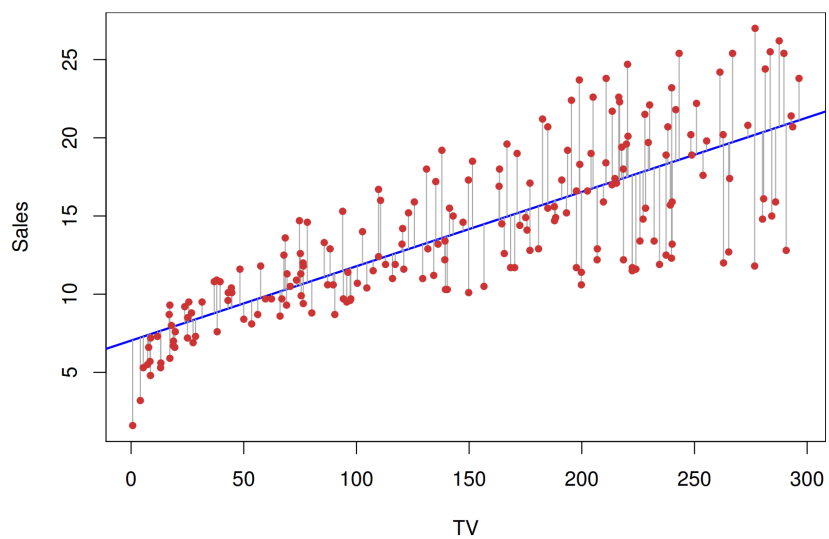
قسمت 1: رگرسیون خطی (با استفاده از Gradient Descent)

The following text is based on the ISLR textbook, please read it thoroughly and answer the questions accordingly.

Simple linear regression lives up to its name: it is a very straightforward approach for predicting a quantitative response Y on the basis of a single predictor variable X. It assumes that there is approximately a linear relationship between X and Y. Mathematically, we can write this linear relationship as

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

در مسئله‌ی رگرسیون خطی، باید با استفاده از داده‌هایی که مشاهده کردیم، بتا صفر و بتا یک را محاسبه کنیم طوری بهترین تخمین ممکن را بدهد. این کار را به طرق مختلف و با تعاریف مختلفی می توان انجام داد. اما در این سوال می خواهیم از روشی که در ادامه توضیح داده می شود استفاده کنیم.



همانطور که در تصویر بالا مشاهده می کنید، داده‌های واقعی (نقاط قرمز) از خط رگرسیون (خطی که توسط معادله رگرسیون رسم می شود) می توانند فاصله داشته باشد. این فاصله را e_i می‌نامیم که برابر است با مقدار حقیقی منهای مقدار پیش‌بینی شده توسط خط رگرسیون:

$e_i = y_i - (\beta_0 + \beta_1 * x_i)$ (اندازه‌ی این مقدار در شکل فوق به رنگ خاکستری نشان داده‌شده‌است)

اکنون Residual Sum of Square (RSS) را به صورت زیر تعریف می کنیم:

$$RSS = e^2_1 + e^2_2 + e^2_3 + ... + e^2_n$$

در نتیجه RSS برابر است با:

$$RSS = (y_1 - (\beta_0 + \beta_1 * x_1))^2 + (y_2 - (\beta_0 + \beta_1 * x_2))^2 + ... + (y_n - (\beta_0 + \beta_1 * x_n))^2$$

⇨ $RSS = \sum_{i=1}^{i=n} (y_i - (\beta_0 + \beta_1 * x_i))^2$

همانطور که در معادله‌ی بالا مشاهده می‌کنید، برای یک سری داده دو متغیر وجود دارد: بتا صفر و بتا یک. و هدف رگرسیون خطی این است که طوری بتا صفر و بتا یک را انتخاب کنیم که مقدار RSS مینیموم شود. حال به سوال‌های زیر پاسخ دهید.

1) در یک فایل تکست یا ورد، وکتور گرادیان (Gradient Vector) را برای تابع RSS بنویسید. یادآوری: از آنجا که این تابع دو متغیری است، وکتور گرادیان آن به صورت مقابل خواهد بود: $\begin{pmatrix} \frac{\partial RSS}{\partial beta0} \\ \frac{\partial RSS}{\partial beta1} \end{pmatrix}$. (1 نمره)

2) یک list پایتون درست کنید که در آن اعداد طبیعی از 1 تا 20 قرار داشته باشند و آن را X بنامید. حال یک لیست دیگر بسازید که مقدار هر عدد درونش برابر با 3 + X * 2 باشد و آن را Y بنامید. سپس به داده‌های Y مقداری noise اضافه کنید. برای اضافه کردن noise می‌توانید از random.random() استفاده کنید. (برای استفاده از random باید در کد خود import random را هم قرار دهید) (1 نمره)

3) حال با استفاده از gradient descent، بر روی داده‌های X و Y، رگرسیون خطی انجام دهید و beta0 و beta1 را به دست آورید. برای حل این سوال، فایل می توانید از کد های فایل gradient_descent.py کمک بگیرید. این فایل دارای کد های مربوط به کد gradient descent چند متغیره است که قبلا در کلاس حل شده است. (در صورت حل با آلفای ثابت 2 نمره، در صورت حل با آلفای متغیر 3 نمره)

نکته 1: در صورت نیاز احتمالی به چک کردن تابع‌های numpy، می‌توانید از CHEETSHEET.pdf استفاده کنید. (هر چند بدون استفاده از numpy هم می‌توان به این سوال جواب داد. همچنین از آنجایی که بدنه کد در اختیارتان قرار داده شده احتمالا نیازی به چک کردن CHEETSHEET نخواهید داشت)

نکته 2: برای یادآوری معادله Gradient Descent، می توانید فایل P1.pdf را باز کنید.

قسمت 2: Alignment

سوال Alignment به نسبت برنامه‌ی ابتدایی **بسیار** ساده‌سازی شده است. در نتیجه سوال‌های 5 و6 با وجود عدم ارتباط با Alignment، جزو آزمون هستند.

4) برنامه‌ای بنویسید که دو توالی Align شده را بخواند و آن‌ها را به صورتی که در مثال زیر آمده‌است پرینت کند(بین نوکلئوتیدهای یکسان خط عمود پرینت کند): (1 نمره)

مثال ورودی:

A C G T T - C A

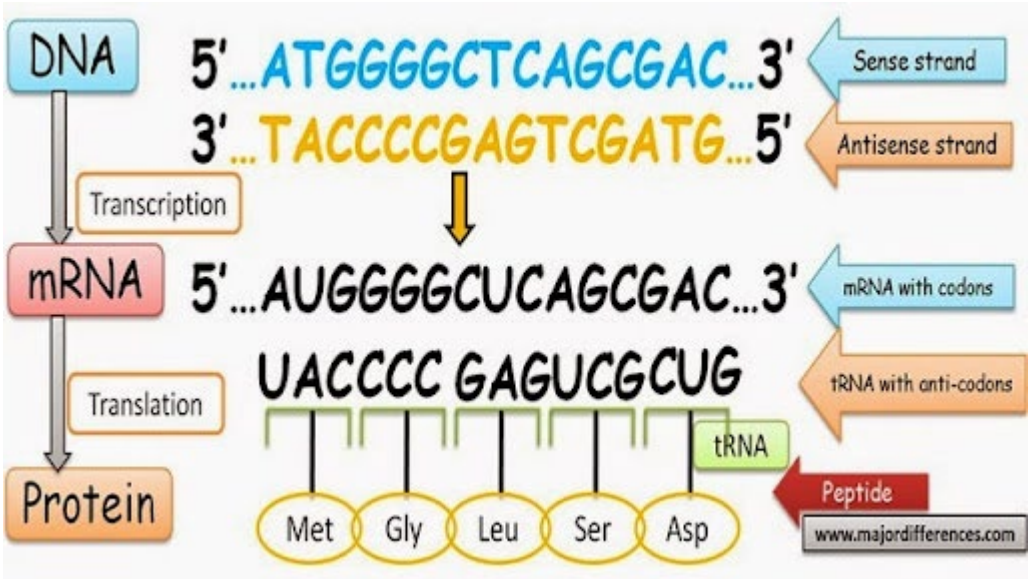
A - G T T C G T

خروجی مد نظری که باید print شود:

A C G T T — C A

| | | | | |

A - G T T C C A



The following sequence is the sequence of alcohol dehydrogenase gene from yeast.

```
"TACTGATAAGGACTGTTCGTCAACCGACGGCAAAAGGTGTGGGTGCCACCAGGGCTCTTGCAGTTCAAGCTCCTTCAGGGGCAGCGGCTCGGGCCA
GTTCTGCTCCAGAACCAATTGTAGTTCATGTGGCCACAGACGGTGTGGCTAAATGTGCGAGAAGTTCCACTGACCGGAGAAGGGCGGTTCTACGGAA
```

```
ACTAGCCACCAGTGCTTCCACGACCACAGCAGCAGTTCCAGCCACGGCCACAGTGAGCAGAATTCTAACCACTGGCACAACCACAGTTCACCTACTT
GAGAAGAACGCCATTGACGCTCATGACATACTTCCGACTCCTCTGGTAGACGGGAGTGTAAGTTGAAAGGCCAATGTGGCAACTGCCATGAAAGGTT
GTGATGACGTAACGGTTACGGTGGGTACGATGGTAGTAGGGGCTCAGGCAAGGGGAGCTCCAACGACGAGGGTAGTACACGCGACCATAGTGAACGA
TAGCACGGAACTTCCTTAGGTTCCAGCCGGGACCACTCACCTAGACGTAAGGGCCACGGCCACCACCAGAACCGGTAGAACGGCAGGTTATGCGGTT
CCGATACCGATACGCACAACAACGGTAACTATGACCACTACTGTTCCGACTCGAGCAGTTCAGGAAACCACGACTCCAGAAGGAACTGAAGTTCTTC
CTTCGGCTGTACTAACTCCGACAGTTCCGACGGTGGTTGCCACCACGGGTGCCATGGAACCAGAATAGGTGGAGGGGGTTCAGAATGCTCGTTCGAC
GACCGAAACGGGCAGGGCCAAGGTGGTACCAGTGACAAAGGTACGGACGGCCACGGTTCGAGCCACGACTATAGAAGACCAACTGGCAATTCTACGA
ATTCTAGACGCCAAGAGTGCAGCCATTGGCATAACTGAGATAGCTCCGAGAACTTATGCAAAGGGCACCAGAGCAGTTCGGAATGATGTTCCAGGTT
GGGAAGAGATGAGAAGGGCTGCAGATGGCAGAGTACGTACTCTTGTTCTAACGGCCGGCATAGCAGAACCTGGAAAGGTTCATT"
```

Copy and paste the sequence and save it in a variable called DNA_String as a string in your code.

Now write a Python script that transcribes the DNA sequence to the corresponding mRNA and print the mRNA string in the console. (2 points)

Hint: You can append a string by using the + operand. For instance, if RNA = "AGG", then you can use RNA = RNA + "A" to turn AGG to AGGA.

6) Below, please find a Python dictionary that contains the corresponding aminoacid for a codon:

```
codon_table = {"UUU":"F", "UUC":"F", "UUA":"L", "UUG :"L",
    "UCU":"S", "UCC":"s", "UCA":"S", "UCG":"S",
    "UAU":"Y", "UAC":"Y", "UAA":"STOP", "UAG":"STOP",
    "UGU":"C", "UGC":"C", "UGA":"STOP", "UGG":"W",
    "CUU":"L", "CUC":"L", "CUA":"L", "CUG":"L",
    "CCU":"P", "CCC":"P", "CCA":"P", "CCG":"P",
    "CAU":"H", "CAC":"H", "CAA":"Q", "CAG":"Q",
    "CGU":"R", "CGC":"R", "CGA":"R", "CGG":"R",
    "AUU":"I", "AUC":"I", "AUA":"I", "AUG":"M",
    "ACU":"T", "ACC":"T", "ACA":"T", "ACG":"T",
    "AAU":"N", "AAC":"N", "AAA":"K", "AAG":"K",
    "AGU":"S", "AGC":"S", "AGA":"R", "AGG":"R",
    "GUU":"V", "GUC":"V", "GUA":"V", "GUG":"V",
    "GCU":"A", "GCC":"A", "GCA":"A", "GCG":"A",
    "GAU":"D", "GAC":"D", "GAA":"E", "GAG":"E",
    "GGU":"G", "GGC":"G", "GGA":"G", "GGG":"G",}
```

You can use this dictionary for translation of the mRNA. For instance if you call codon_table["UUU"], Python would return "F". In this problem, you must write a program that translates your mRNA from the previous question and prints the protein to the console. (2 points)