تمرین تحویلی (تاریخ تحویل: ۲۲ تیر)

- دانشجویان گرامی، تمرین مربوط به همین جلسه با عنوان «روش دوم برآورد پارامترهای مدل: بیشینه‌سازی تابع درست‌نمایی» را در مخزنی در GitHub که توسط دستیار آموزشی درس، آقای سلمانی، ایجاد و به اشتراک گذاشته شده است، تحویل دهید.
- توجه کنید که به دلیل اهمیت این سری تمرین، نمره آن، وزن ۳ خواهد داشت.

الهام نبریزی - دانشگاه خوارزمی

تمارین تحویلی (تاریخ تحویل: ۲۲ تیر)
روش دوم برآورد پارامترهای مدل: بیشینه‌سازی تابع درستنمایی

در این فعالیت می‌خواهیم برای مدل رگرسیون پواسون (پیش‌بینی تعداد ماهواره‌های خرچنگ نعل‌اسبی)، پارامترهای مدل را به صورت دستی و با استفاده از روش بیشینه‌سازی درست‌نمایی (MLE) تخمین بزنیم.

مدل:

$$\log(\lambda_i) = \beta_0 + \beta_1 C2_i + \beta_2 C3_i + \beta_3 C4_i + \beta_4 S2_i + \beta_5 S3_i + \beta_6 W_i + \beta_7 Wt_i$$

که در آن λ_i میانگین توزیع پواسون برای مشاهده i ام است.

گام ۱: آماده‌سازی داده‌ها و کدگذاری متغیرهای کیفی (Dummy Coding)

در رگرسیون پواسون (و به طور کلی در مدل‌های خطی)، ما نمی‌توانیم نام دسته‌ها یا کدهای عددی ساده (مثل ۱، ۲، ۳) را برای متغیرهای کیفی وارد فرمول کنیم؛ زیرا مدل تصور می‌کند که بین دسته‌ی ۱ و ۲، یک واحد فاصله وجود دارد. برای حل این مشکل، باید از **متغیرهای مجازی یا دامی** استفاده کنیم.

همان‌طور که در جلسات قبل برای «متغیر تحصیلات» بحث کردیم، قانون کلی این است:

> اگر یک متغیر کیفی دارای k سطح (دسته‌بندی) باشد، ما باید $k-1$ متغیر مجازی برای آن بسازیم. یک سطح همواره به عنوان **سطح مرجع (Reference Level)** در نظر گرفته می‌شود و ضرایب سایر سطوح نسبت به آن تفسیر می‌گردد.

ادامه تمارین تحویلی (تاریخ تحویل: ۲۲ تیر)
روش دوم برآورد پارامترهای مدل: بیشینه‌سازی تابع درستنمایی

در فایل داده‌های خرچنگ، دو متغیر کیفی داریم:

۱. **متغیر رنگ** ©: این متغیر دارای **۴ سطح** است. بنابراین طبق فرمول $k-1$ ، باید **۳ متغیر مجازی** بسازیم. از آنجایی که می‌خواهیم خروجی تابع خودمان را با تابع `glm` مقایسه کنیم، سطح ۱ را مرجع می‌گیریم و سه ستون جدید تعریف می‌کنیم:

- **C2**: اگر رنگ برابر با ۲ بود، مقدار ۱ و در غیر این صورت ۰ می‌گیرد.
- **C3**: اگر رنگ برابر با ۳ بود، مقدار ۱ و در غیر این صورت ۰ می‌گیرد.
- **C4**: اگر رنگ برابر با ۴ بود، مقدار ۱ و در غیر این صورت ۰ می‌گیرد.

این کدگذاری را با دستور `ifelse` در R انجام دهید:

```
C2 = ifelse(C == 2, 1, 0)
C3 = ifelse(C == 3, 1, 0)
C4 = ifelse(C == 4, 1, 0)
```

ادامه تمارین تحویلی (تاریخ تحویل: ۲۲ تیر)
روش دوم برآورد پارامترهای مدل: بیشینه‌سازی تابع درستنمایی

۲. **متغیر وضعیت ستون فقرات (S)**: این متغیر دارای **۳ سطح** است. پس باید **۲ متغیر مجازی** (با در نظر گرفتن سطح ۱ به عنوان مرجع) بسازیم:

- **S2**: اگر وضعیت برابر با ۲ بود، ۱ و در غیر این صورت ۰.
- **S3**: اگر وضعیت برابر با ۳ بود، ۱ و در غیر این صورت ۰.

```
S2 = ifelse(S == 2, 1, 0)
S3 = ifelse(S == 3, 1, 0)
```

ریزی، دانشگاه خوارزمی

چرا این کار را دستی انجام می‌دهیم؟

وقتی در R از تابع `glm(Sa ~ C + S + ...)` استفاده می‌کنید، R خودش به صورت خودکار این ستون‌های ۰ و ۱ را در حافظه می‌سازد. اما وقتی می‌خواهیم تابع درست‌نمایی را «خودمان» بنویسیم، باید این متغیرها را به صورت شفاف تعریف کنیم تا بتوانیم آن‌ها را در «ترکیب خطی» ($\beta_1 C2 + \beta_2 C3 + \ldots$) ضرب کنیم.

گام ۲: نوشتن تابع درست‌نمایی

تابعی به نام `nloglik_loglinear` بنویسید که ورودی آن بردار `theta` (شامل ۸ پارامتر) باشد.

داخل تابع:

1. بردار `theta` را به ۸ پارامتر مدل (β_0 تا β_7) تقسیم کنید.
2. یک حلقه `for` برای محاسبه درست‌نمایی هر مشاهده ایجاد کنید.
3. در هر تکرار حلقه، ابتدا `linear_predictor` را محاسبه کنید:

$$\eta_i = \beta_0 + \beta_1 C2_i + \cdots + \beta_7 Wt_i$$

4. سپس مقدار λ_i را با استفاده از تابع نمایشی به دست آورید:

$$\lambda_i = e^{\eta_i}$$

5. در ادامه، احتمال مشاهده‌شدن مقدار واقعی `Sa[i]` را با فرض توزیع پواسون محاسبه کنید:

$$P(Sa_i) = \frac{e^{-\lambda_i} \lambda_i^{Sa_i}}{Sa_i!}$$

راهنمایی: در R، تابع `dpois(x, lambda)` مستقیماً این مقدار احتمال را برای شما برمی‌گرداند.

6. در پایان، حاصل‌ضرب احتمالاتِ تمامیِ مشاهده‌ها را محاسبه کرده و منفی لگاریتمِ این حاصل‌ضرب را برگردانید.

ادامه تمارین تحویلی (تاریخ تحویل: ۲۲ تیر)
روش دوم برآورد پارامترهای مدل: بیشینه‌سازی تابع درستنمایی

گام ۳: بهینه‌سازی عددی

نقطه‌ی شروع (Initial Values) برای الگوریتم `nlminb` بسیار مهم است.

تمرین: بردار `theta0` را تعریف کنید.

راهنمایی: می‌دانیم که در مدل پواسون، میانگین (λ) برابر با e^{β_0} است (اگر سایر متغیرها صفر باشند). بنابراین بهترین حدس اولیه برای β_0 ، لگاریتمِ میانگینِ کل داده‌ها است:

`log(mean(Sa))`

سایر ضرایب (β_1 تا β_7) را می‌توانید صفر در نظر بگیرید.

حالا با استفاده از تابع `nlminb` ، تابع `nloglik_loglinear` را کمینه کنید (از بازه‌ی `(Inf, Inf-)` برای پارامترها استفاده کنید).

ادامه تمارین تحویلی (تاریخ تحویل: ۲۲ تیر)
روش دوم برآورد پارامترهای مدل: بیشینه‌سازی تابع درستنمایی

گام ۴: مقایسه با `glm`

مدل مشابه را با تابع `glm` برازش دهید:

```r
glm(Sa ~ C + S + W + Wt, data = train_data, family = poisson(link = 'log'))
```

پرسش: ضرایب به دست آمده از `nlminb` را با ضرایب `glm` مقایسه کنید. آیا دقیقاً یکی هستند؟ اگر تفاوت ناچیزی در حد اعشار دارند، به نظر شما علت آن چیست؟