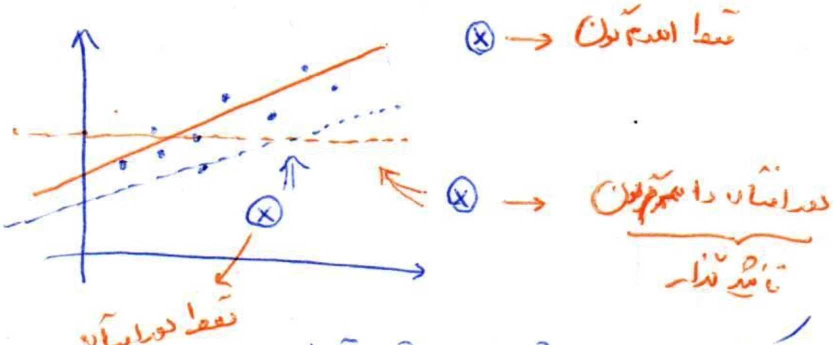
داده تأثیرگذار:

داده تأثیرگذار، داده‌ای است که بودن یا نبودن آن در مدل، ویژگی‌های کلیدی مدل را تحت تأثیر قرار داده.



نقاطی که فقط اهرم‌گون هستند، مشکلی ایجاد نمی‌کنند. فقط برای پیش‌بینی برای آن نقاط باید با احتیاط عمل کرد. ممکن است الگوی بین X و Y در آن محدوده تغییر کرده باشد و چون مشاهداتی نداشته‌ایم این مسأله مشخص نشده باشد.

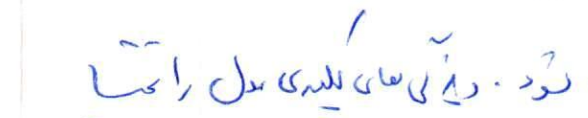
(عرض از مبدأ را با احتیاط تفسیر کرد.)

نقاطی که فقط دورافتاده هستند نیز فقط MSE را افزایش می‌دهند و ویژگی‌های کلیدی مدل را خیلی تحت تأثیر قرار نمی‌دهند. لازم است با این داده‌ها برخورد شود ولی به اندازه داده‌های تأثیرگذار، مشکل‌ساز نیستند.

داده‌ای که هم اهرم‌گون و هم دورافتاده باشد، داده تأثیرگذار خواهد بود. ویژگی‌های کلیدی مدل را تحت کنترل خودش درآورده. درحالی‌که ما می‌خواهیم مدل براساس تمامی مشاهدات ساخته شود و نه فقط براساس چند مشاهده محدود، تشکیل شود.

ممکن است داده تأثیرگذاری داشته باشیم که دورافتاده یا اهرم‌گون نباشد. پس برای تشخیص همه آنها لازم است معیارهایی نظیر آماره کوک و ... که در ادامه معرفی می‌شوند را نیز محاسبه کنیم.

نمودار داده‌های اهرم‌گون در مقابل دورافتاده:



۱۴۲۰

```
> Highway1 = read.table(" . . . . . ", header = T)

> mymodel = lm(rate ~ shld, Highway1)

> h = influence(mymodel) $ hat

> library(MASS)

> r = studres(mymodel)

> plot(h, r) ; abline(v = (2*2/39), col = 2) ; abline(h = c(-1, 1))

> text(h, r, label = rownames(Highway1))

> Highway2 = Highway1[-32, ]

> mymodel2 = lm(rate ~ shld, Highway2)
⋮
```

دادهٔ 32ام تأثیرگذار بوده.

تکرار مراحل قبل

تمرین: تا کی پیش بروید که مدل بدون نفوذ را بگیرید؟ داده‌های تأثیرگذار را تا جایی که لازم است حذف کنید مدل نهایی را گزارش دهید.

* می‌بینید که در مدل با دادهٔ 32ام و بدون آن هم تغییری نکرده‌اند. در واقع این داده روی ضرایب کلیدی از مدل بیشتر تأثیرگذار است.

آماره کوک

برآورد β بدون iامین نقطه داده..

$$D_i = \frac{(\hat{\underset{\sim}{\beta}} - \hat{\underset{\sim}{\beta}}_{(i)})'(X'X)(\hat{\underset{\sim}{\beta}} - \hat{\underset{\sim}{\beta}}_{(i)})}{p\,MSE}$$

اگر $D_i > 1$ باشد می‌گوییم مشاهده iام بر روی بردار ضرایب $\hat{\underset{\sim}{\beta}}$ تأثیرگذار است.

```
> mymodel = lm(rate ~ acpt + len + slim + shld, Highway1)
> inf = influence.measures(mymodel)
> D = inf$infmat[, 8]
```

می‌گوید که آیا مشاهده پرت بوده یا خیر

آماره DFBeta :

$$DFBeta_{j,i} = \frac{\hat{\beta}_j - \hat{\beta}_{j(-i)}}{\sqrt{S^2_{(i)} C_{jj}}}$$

اگر $|DFBeta_{j,i}| > \frac{2}{\sqrt{n}}$ یعنی داده iام بر برآورد ضریب β_j تأثیرگذار است.

```
> dfbeta_int = inf$infmat[, 1]
> dfbeta_acpt = inf$infmat[, 2]
> dfbeta_len = inf$infmat[, 3]
> dfbeta_slim = inf$infmat[, 4]
> dfbeta_shld = inf$infmat[, 5]
```

حال باید بررسی کنیم که کدام یک از $|DFBeta|$ها بیشتر از $\frac{2}{\sqrt{n}}$ هستند.

```
> which(abs(dfbeta_int) > (2/sqrt(39)))
> plot(1:39, abs(dfbeta_int))
> abline(h = (2/sqrt(39)))
> text(1:39, abs(dfbeta_int), label = rownames(Highway1))
```

یک بار هم نمودار

آماره Covratio :

$$Covratio_i = \frac{|(X'_{(i)} X_{(i)})^{-1} S^2_{(i)}|}{|(X'X)^{-1} MSE|}$$

اگر $|Covratio_i - 1| > \frac{3p}{n}$ آنگاه مشاهده iام به دسته [illegible]

ماتریس ($\hat{\beta}$ ها) تأثیرگذار است.

آماره DFFits :

$$DFFits_i = \frac{\hat{Y}_i - \hat{Y}_{(i)}}{\sqrt{S^2_{(i)} h_{ii}}}$$

اگر $|DFFits| > 2\sqrt{\frac{p}{n}}$ آن گاه مشاهده iام روی پیش‌بینی y_i یعنی $\hat{y}_i$ تأثیرگذار است.

```
> dffits = inf $ influence [ , ]
```

شماره ستون مربوط

انتخاب متغیر:

اگر متغیری، آزمون صفر بودن ضریب رگرسیونی آن رد نشد، حذف آن در مدل باعث افزایش MSE و کاهش دقت نمی‌شود. با این حال اگر چند متغیر همزمان صفر بودن ضریبشان رد نشد، نمی‌توان یکدفعه همه را با هم حذف کرد. برای این منظور دو روش می‌توان در پیش گرفت:

الف: آزمون کرد که آیا همزمان این متغیرها می‌توانند از مدل حذف شوند. (توضیحات اول یکجا با هم را بالا ملاحظه کنید.)

ب: تک تک هم به ترتیب از متغیری که p-value بزرگتری دارد یعنی صفر بودن ضریبش قوی‌تر رد نمی‌شود، حذف کنیم و دوباره مدل بزنیم.

* لازم است در هر مرحله برازش مدل، تمامی فرضیات مدل، داده‌های تأثیرگذار، دورافتاده و هم‌خطی چندگانه چک شود.

* الگوریتم‌های آماری زیادی وجود دارد که با انتخاب متغیر، مدل سازی می‌کنند ولی فرضیات مدل را در هر مرحله چک نمی‌کنند. از جمله این الگوریتم‌ها، Backward، Forward و Stepwise است.

توضیحات مربوط به این الگوریتم‌ها در درس رگرسیون داده شده است. شیء‌ای که در بسته R این الگوریتم‌ها را اجرا می‌کند براساس آماره‌های F آزمون‌ها نیست بلکه براساس مقدار AIC است.

$$AIC = Deviance + 2p$$

در هر مرحله مدلی را انتخاب می‌کند که کمترین AIC را دارد.

مدل انتخاب پیش‌رو : Forward

```
> fulmodel = lm( rate ~ . -obs , Highway1)
```

مدل کامل

```
> summary(fulmodel)
```

→ سپس کدام مدل متغیر کمترین معناداری است

آن را مدل پایه در نظر می‌گیریم و در اینجا len بوده است.

```
> basemodel = lm(rate ~ len , Highway1)
```

مدل پایه

```
> step( basemodel , scope = list(upper = fulmodel , lower = ~1) ,
        direction = "forward" , trace = F )
```

اگر F باشد تنها مدل نهایی را نشان می‌دهد. اگر T باشد مراحل الگوریتم را نشان می‌دهد.

* تا زمانی الگوریتم مراحل را ادامه می‌دهد که Non بالاتر قرار بگیرد یعنی اضافه کردن متغیر به مدل باعث افزایش AIC شود. در این جا الگوریتم متوقف می‌شود که مدل نهایی گزارش داده می‌شود.

مدل حذف پس‌رو : Backward

```
> step( fulmodel , direction = "backward" , trace = T )
```

در هر مرحله متغیری که بیشترین AIC را ایجاد کرد، حذف می‌کند. تا زمانی که Non ابتدا قرار بگیرد یعنی حذف کردن بیشتر باعث افزایش AIC شود.

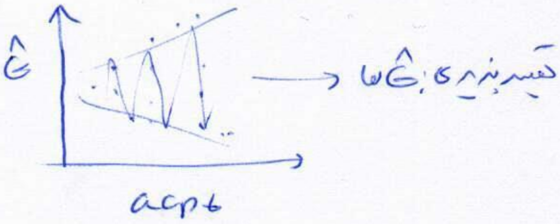
مدل قدم به قدم : Stepwise

```
> step( basemodel , scope = list (upper = fulmodel , lower = ~1) ,
        direction = "both" , trace = F )
```

تمرین مدل :

به منظور چک کردن فرضیه ثبات واریانس نمودارهای باقیمانده‌ها در مقابل پیش‌بینی‌ها و نیز در مقابل هر یک از Xها رسم می‌کنیم، اگر روندی داشت، نقض فرضیه ثبات واریانس برقرار است.

```
> e = resid(mymodel)
> yhat = fitted(mymodel)
> par(mfrow = c(3,2))
> plot(e, yhat)
> plot(e, acpt)
> plot(e, len)
> plot(e, slim)
> plot(e, shld)
```



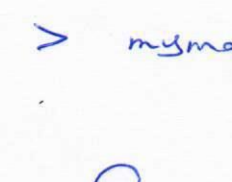
$$Var(e_i) = \sigma^2 acpt_i$$

$$\Rightarrow w_i = 1/acpt_i$$

```
> mymodel_weighted = lm(rate ~ len + slim + shld + acpt,
                        weights = (1/acpt), Highways)
```

می‌توانیم MSE و R^2_{adj} این مدل را با مدل غیروزنی قبلی مقایسه کنیم و همچنین اگر آن بیان شده‌ها را ببینیم آیا وضعیت بهتر شده باشد. در غیر این صورت شاید وزن انتخابی نامناسب بوده، در نتیجه این مشکل را برطرف نکند. همچنین لازم است هر نمودار $\hat{e}_i\sqrt{w_i}$ها در مقابل $\hat{y}_i\sqrt{w_i}$ها رسم شود و ببینیم آیا روند از بین رفته باشد.

```
> w1 = 1/acpt
> plot((sqrt(w1) * yhat), (sqrt(w1) * e)); abline(h=0)
```



وجود

در شکل بالا $w_i = 1/acpt$ در نظر گرفتیم. در این شکل چون روند در داشتن است لازم است تغییر دهیم وزن را $w_i = \frac{1}{\hat{y}_i}$ در نظر بگیریم:

```
> w2 = 1/yhat
> mymodel_weighted = lm( ..... weights = (1/yhat), ... )
```

```
> plot((sqrt(w2)*yhat),(sqrt(w2)*e)); abline(h=0)
```

می بینیم که با این وزن مشکل حل شد. پس وزن $w_i = \frac{1}{acpt_i}$ مناسب است.