Inferență statistică în ML

Cap 9. Ajustări.

May 21, 2019

Adjustment

- prin **adjustment** se înțelege procesul de a reprezenta răspunsurile într-o regresie liniară (în funcție de predictor) pentru a studia impactul unei o a treia variabilă asupra relației dintre primele două
- considerăm exemplul cu capacitatea pulmonară, influențată de doi factori: fumatul și consumul de bomboane mentolate
- găsim o corelație între capacitatea pulmonară scăzută și bomboanele mentolate
- pentru a vedea dacă doar bomboanele mentolate sunt corelate cu volumul respirator, vedem ce influență are o a treia variabilă - fumatul
- ipoteza este că cele două variabile s-ar putea să fie confounded - o variabilă are efect mascat asupra alteia (după ce scoatem efectul fumatului, corelația cu bomboanele dispare)

Adjustment (2)

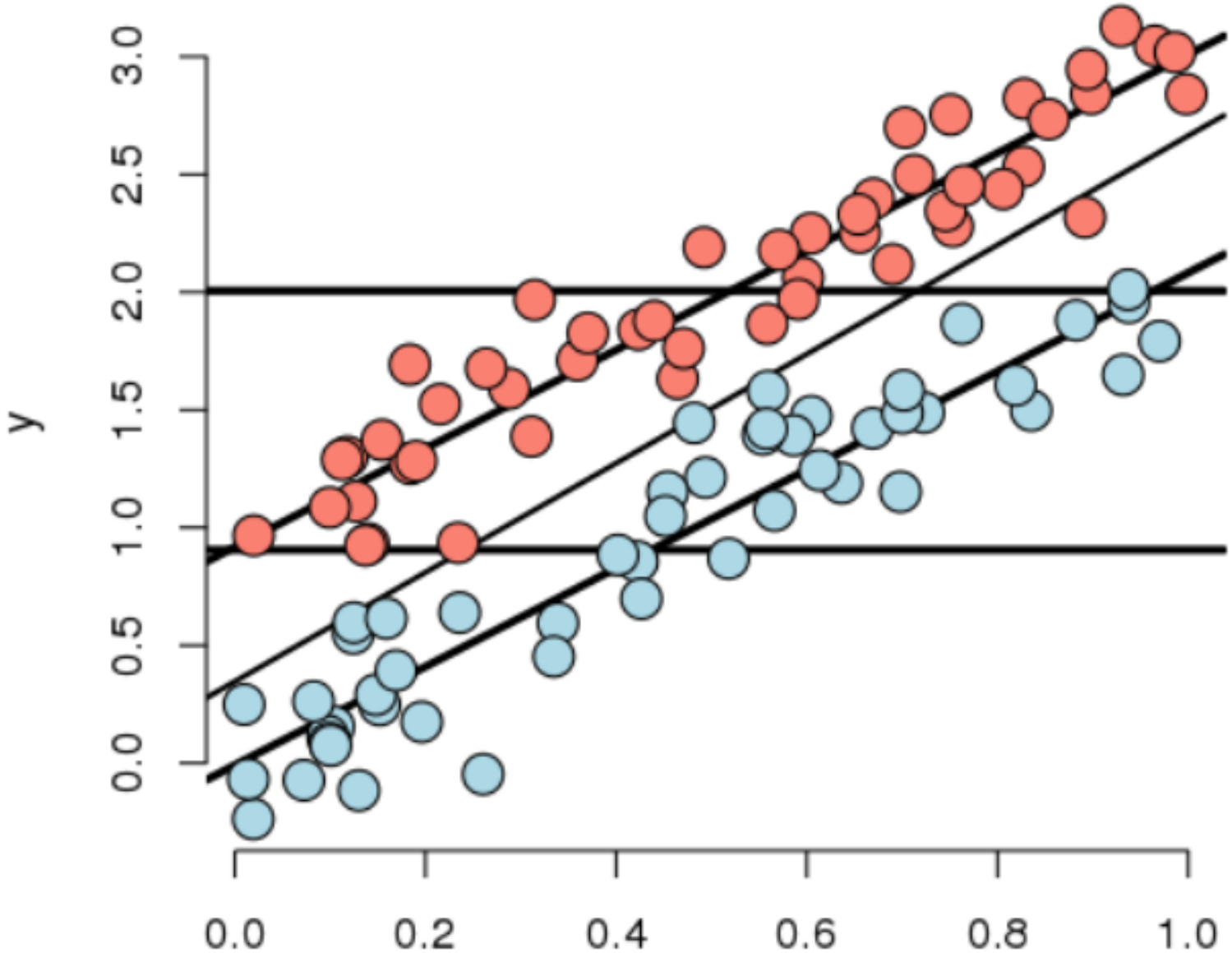
- considerăm modelul:

```
n, sigma = 50, .2
beta0, beta1 = 0, 2
x = np.r_[np.random.rand(n), + np.random.rand(n)]
t = np.array([0]*n + [1]*n)
y = beta0 + t + x * beta1
          np.random.randn(2*n)*sigma +
```

$$Y_i = \beta_0 + \beta_1 X + T + \epsilon_i$$

- ne interesează relația dintre variabila binară (fumatul, sau tratamentul) și Y
- ideea este că s-ar putea ca relația să depindă de variabila continuă X

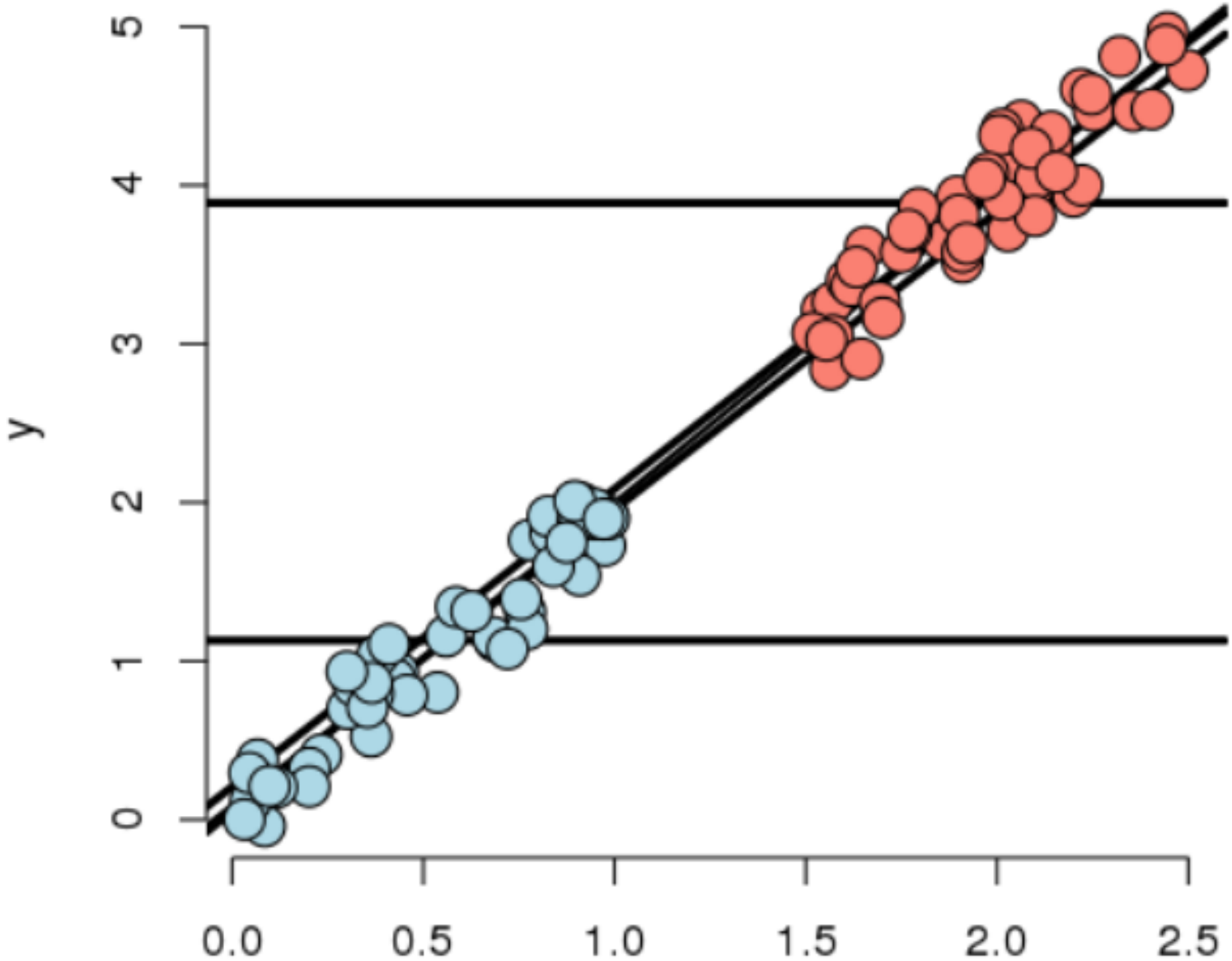
Cazul 1



Cazul 1 (1)

- variabila X nu este corelată cu tratamentul (culoarea)
- Y depinde liniar de X, iar intercept-ul depinde de tipul grupului
- Y depinde de asemenea și de tratament (de grup, roșu/albastru), se poate observa din intersecția liniilor orizontale (mediile) cu axa y
- relația dintre grup și Y este constantă și depinde de X
- modelul vine să confirme relația aparentă între medii
- relația dintre grup (T) și răspunsul Y nu se schimbă mult, indiferent dacă introducem X sau nu; se vede din comparația distanței dintre liniile orizontale și cea dintre intercept-uri
- distanța dintre medii este efectul marginal al variabilei grup
- efectul lui X este echilibrat asupra grupurilor, X poate fi inclus sau nu, se obține același răspuns
- situația este una ideală

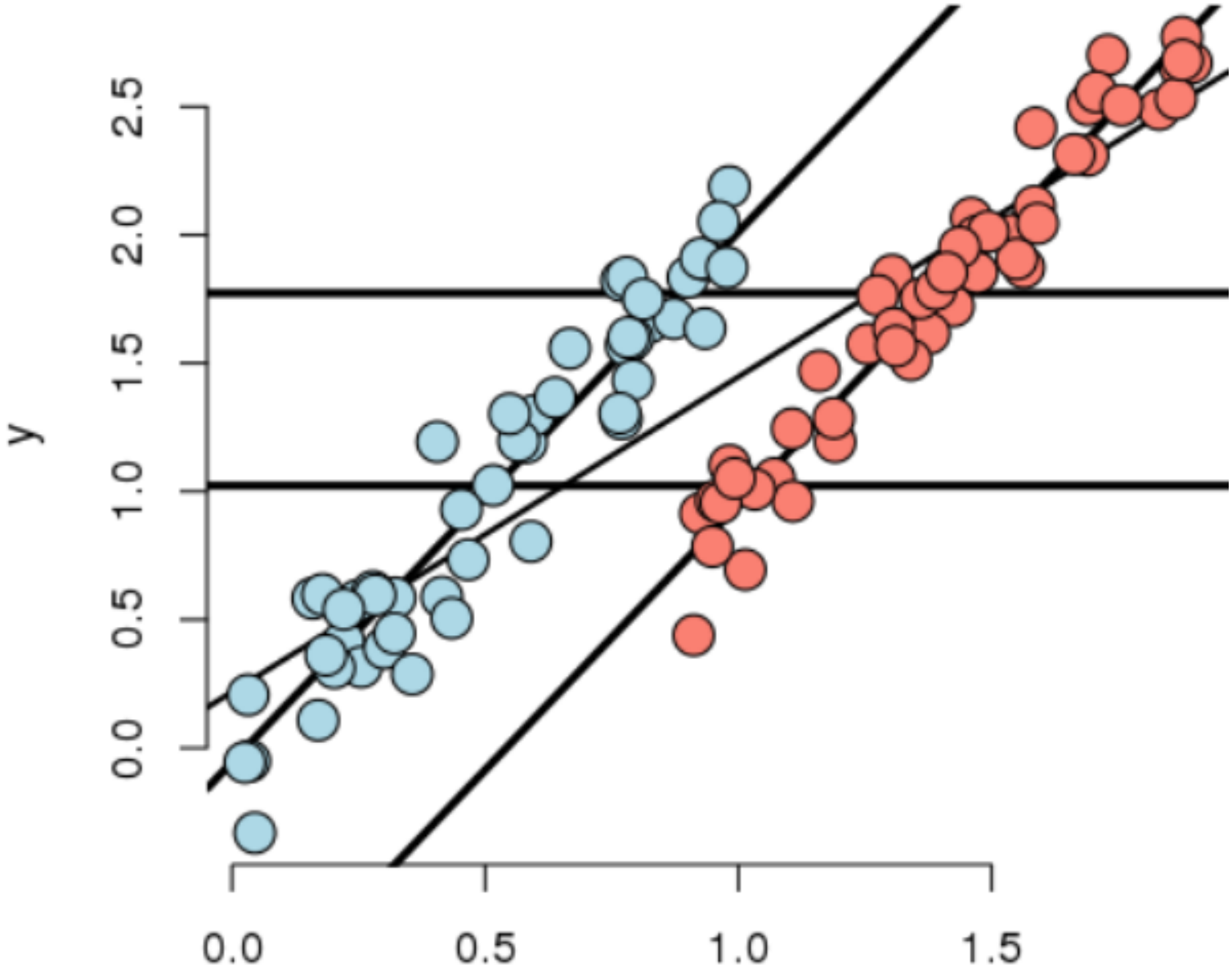
Cazul 2



Cazul 2 (1)

- variabila X este puternic legată de grup (variabila T)
- dacă știm X, putem determina din ce grup face parte punctul; există o margine bine delimitată
- dacă nu considerăm X, pare că Y depinde puternic de grup; dacă introducem X, nu avem corelație cu X
- efectul aparent al grupului asupra lui Y este explicat de X
- putem verifica că Y nu depinde de grup dacă calculăm reziduurile considerând X; valorile reziduurilor nu mai sunt corelate cu grupul
- diferența dintre medii indică o puternică dependență de grup; însă diferența dintre intercept-uri e minoră, dacă considerăm X, deci nu gruparea explică răspunsul, ci X

Cazul 3

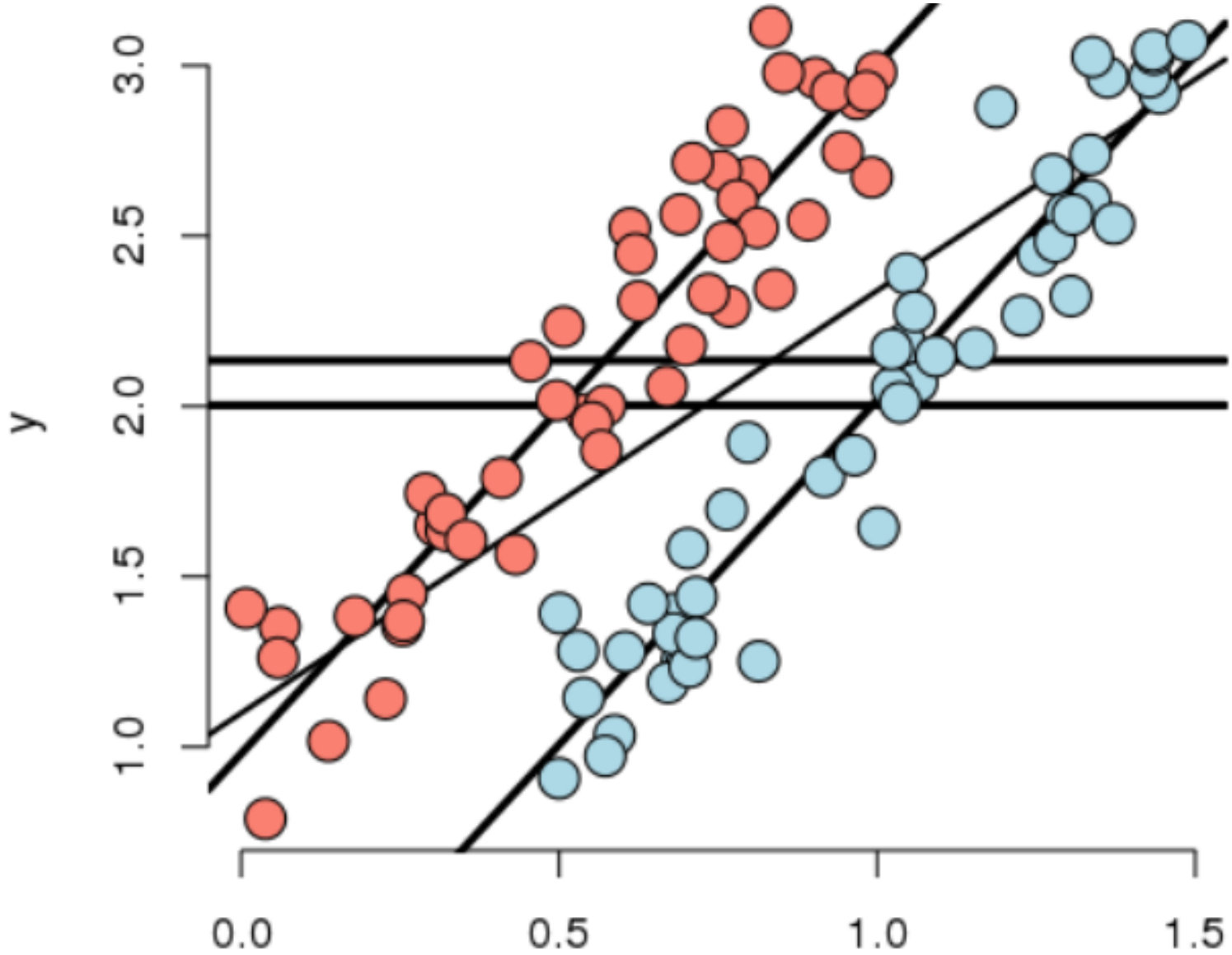


Cazul 3 (1)

- există suprapunere între grupuri
- caz dificil: diferența între medii indică grupul roșu mai mare decât cel albastru, dar modelul indică, prin diferența dintre intercept-uri, că grupul albastru e mai mare decât cel roșu
- adică răspusul ajustat cu X este semnificativ și exact opusul răspunsului neajustat (doar Y funcție de grupul T) - acesta este paradoxul Simpson[1]
-

[1] https://en.wikipedia.org/wiki/Simpson%27s_paradox)

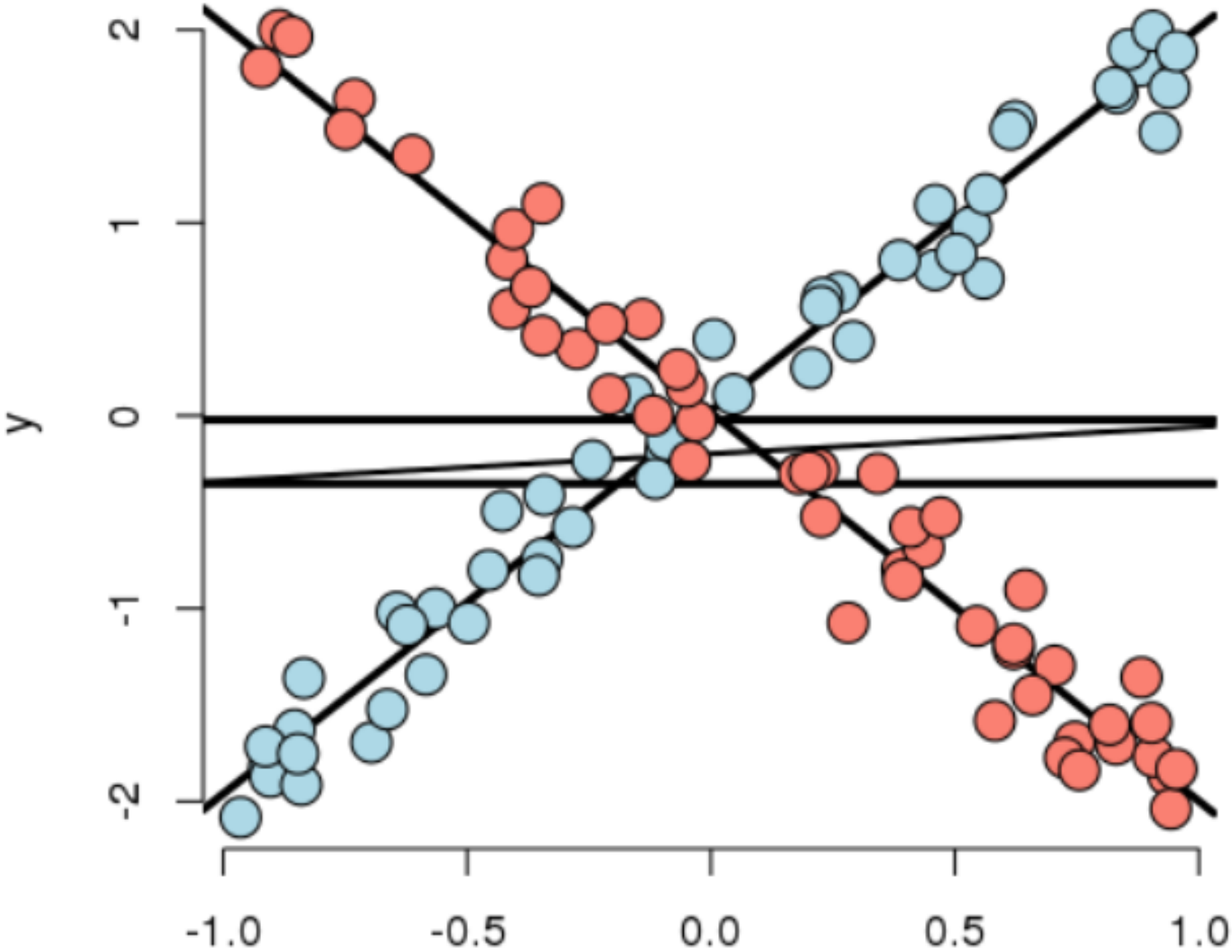
Cazul 4



Cazul 4 (1)

- aproape nu există efect marginal al grupului (diferența între medii)
- un efect major când ajustăm (includem) X (diferența între intercept-uri)
- prin adăugarea lui X la model, efectul grupului devine semnificativ

Cazul 5



Cazul 5 (1)

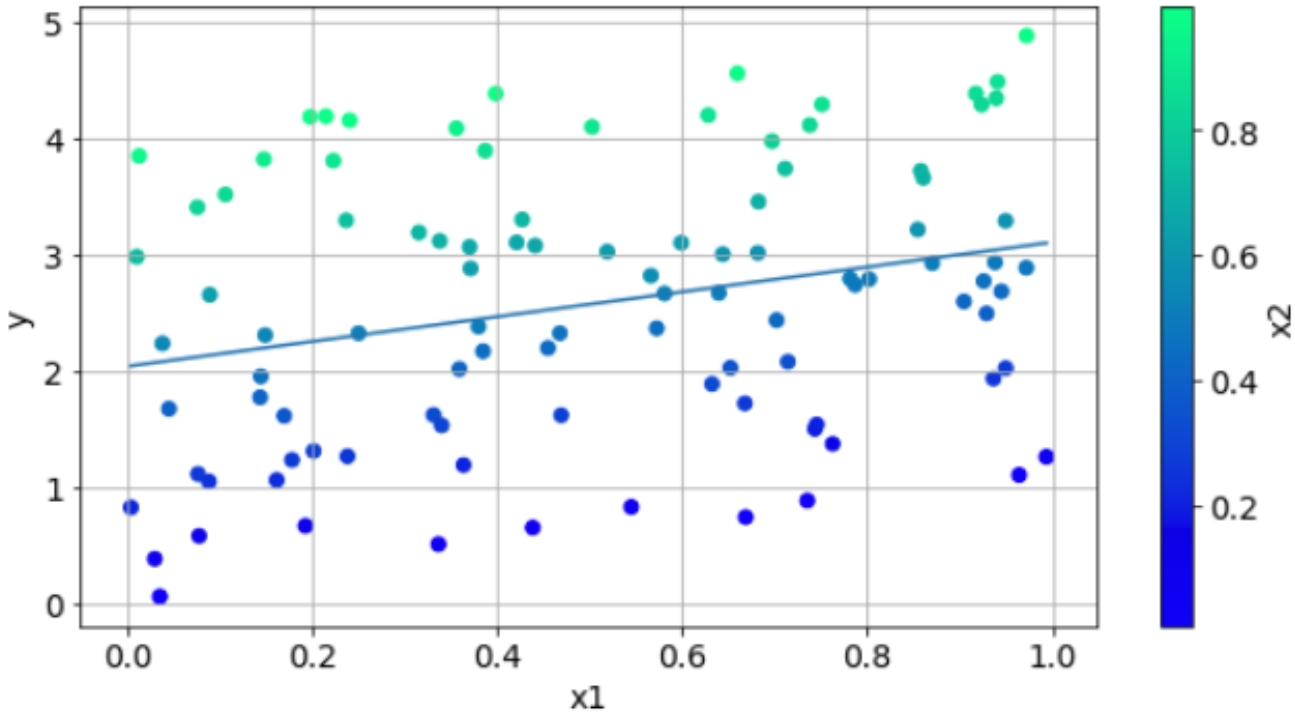
- exemplu în care am produce rezultate greșite dacă am presupune că pantele celor două drepte de regresie ar fi aceleași
- modelul corect ar fi:

$$Y_i = \beta_0 + \beta_1 T + \beta_2 X_i + \beta_3 T X_i + \epsilon_i$$

- pentru acest set, nu există un efect al variabilei T (tratament)
- comparând mediile, nu există acest efect
- comparând pentru X mic, există un efect pregnant în favoarea grupului roșu
- comparând pentru X mare există un efect pregnant în favoarea grupului albastru
- rezultatul este că β_1 nu poate fi interpretat ca efect al tratamentului, pentru că nu există acest lucru; există un termen de interacțiune între cele două variabile X și T în model
- efectul tratamentului depinde de nivelul lui X

Variabile continue

```
p, n, x2 = 1, 100, np.random.rand(n)
x1 = p * np.random.rand(n) - (1-p) * x2
beta0, beta1, tau, sigma = 0, 1, 4, .01
y = beta0 + x1*beta1 + tau*x2 + np.random.randn(n)*sigma
```



- răspunsul Y pare că nu depinde de X_1, ci de X_2

Ajustări (2)

- alegerea uneia sau a alteia dintre variante depinde de domain knowledge
- exemplu de variabile corelate: presiunea sistolică și presiunea diastolică; ambele în model - compensând pentru una, cealaltă dispare, pentru că cealaltă nu mai e corelată cu răspunsul (logic, ambele exprimă același lucru, sunt puternic corelate)
- se fac simulări repetate cu diverse variante de model, pentru a studia influența tuturor variabilelor (variabilitatea cînd includem / excludem aceeași variabilă)
- construirea automată a modelului nu oferă interpretabilitate coeficienților, în schimb e folosit pentru construcția unor funcții de predicție (cu asta se ocupă machine learning)
- toate ajustările realizate trebuie analizate și găsite justificări valide pentru ele

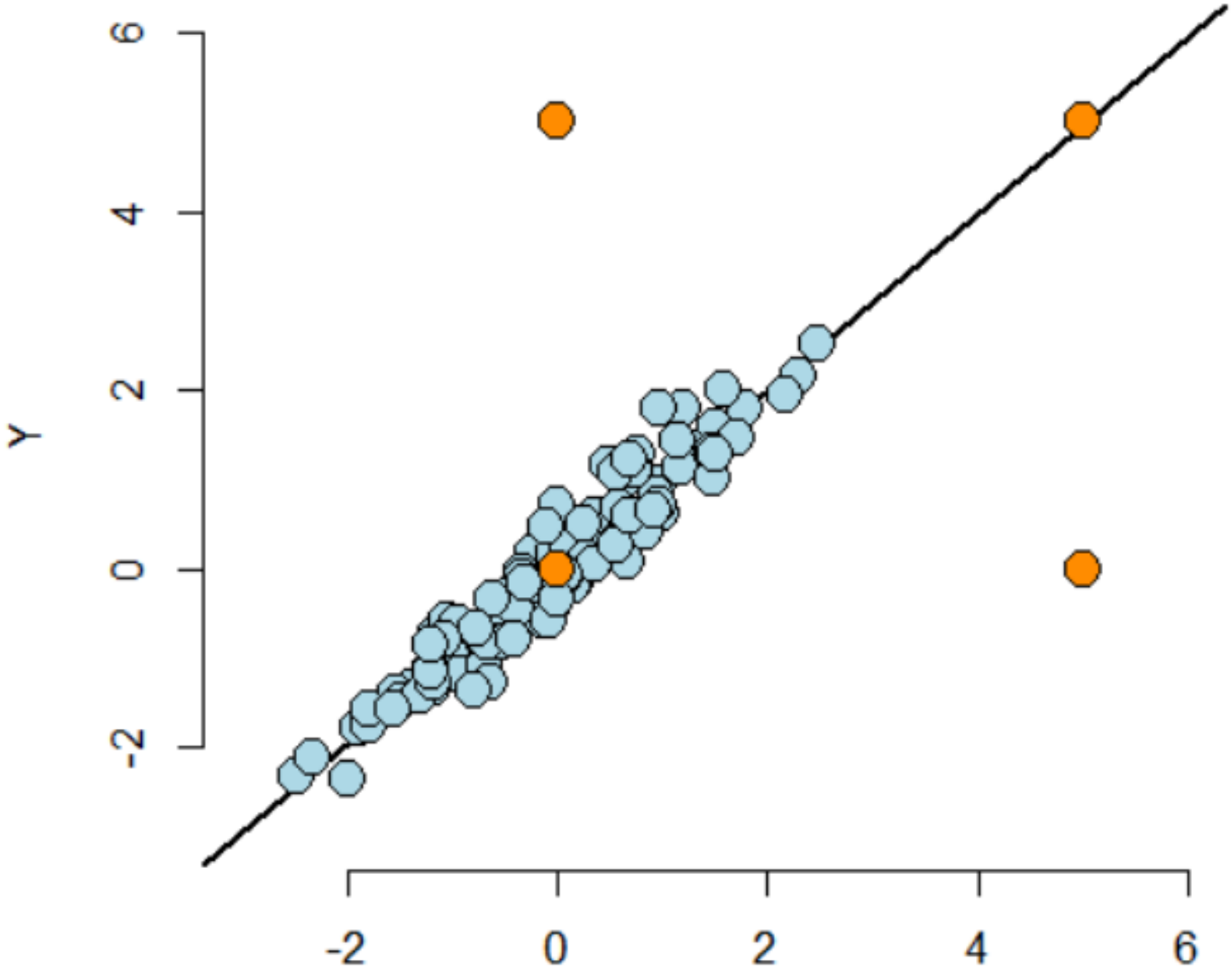
Reziduuri și diagrame de diagnoză

- modelul liniar, $Y_i = \sum_{k=1}^{p} X_{ik}\beta_j + \epsilon_i$
- presupunem $\epsilon_i \sim N(0, \sigma^2)$
- reziduurile sunt definite ca $\epsilon_i = Y_i - \hat{Y}_i = Y_i - \sum_{k=1}^{p} X_{ik}\hat{\beta}_j$
- estimarea dispersiei reziduurilor:

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} e_i^2}{n-p}$$

- diagramele de diagnoză ne ajută să vizualizăm cum modelul 'potrivește' datele și dacă presupunerile făcute de modelul de regresie liniară sunt încălcate
- diagramele verifică:
  - dacă datele pot fi 'fitted' cu o linie;
  - erorile sunt distribuite normal cu medie zero;
  - erorile au dispersie constantă, adică homoscedasticity;
  - nu există puncte leverage (puncte cu x extrem) (vs. outlier, puncte cu y extrem)

Puncte de influență



Puncte de influență (1)

- distincție între puncte de influență și outliers
- stânga-jos, punct din date; nu are influență sau leverage
- dreapta-sus, punct cu leverage major (x mare, poate influența linia de regresie pentru că e izolat de celelalte dpdv. al lui x, dar e aproape de regresie, deci nu influențează)
- dreapta-jos, dacă punctul ar fi inclus în regresie, ar avea o influență majoră - high-leverage și high-influence
- stânga-sus, punct cu x în mijlocul norului, are low leverage, dar nu aderă la relație - outlier, nu influențează regresia

Outliers, leverage și puncte de influență

- punctele outlier sunt rezultate din procesul care suferă, accidental, deviații
- există distincție între outliers, leverage points și puncte de influență
- **outlier**: o observație cu un reziduu mare; valoarea Y este neobișnuit de mare față de cea prezisă de dreapta de regresie; indică fie o ciudățenie, fie o eroare de introducere, fie altceva
- **leverage**: o observație cu o valoare extremă pentru X (variabila predictor); leverage este o măsură a depărtării unei observații față de media acelei variabile predictor; acestea pot avea efect asupra estimării coeficienților regresiei
- **influence**: o observație pentru care, dacă ar fi eliminată, coeficienții regresiei s-ar schimba major; poate fi asimilat produsului între outlier și leverage

Ideea de leverage

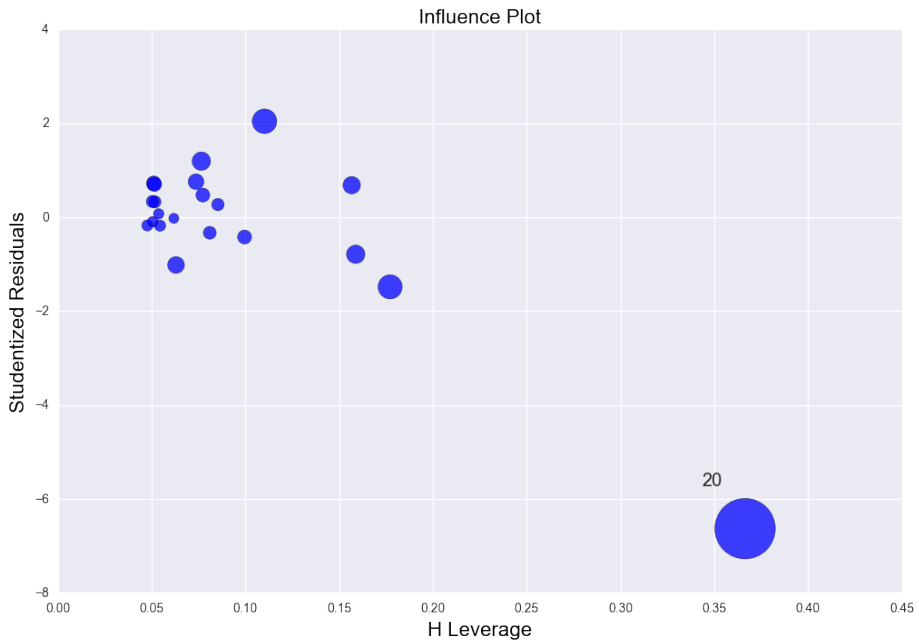
- de reamintit că dreapta de regresie trece prin $(\bar{X}, \bar{Y})$
- dreapta de regresie va 'pivota' în jurul acestui 'mijloc'
- punctele situate departe de $\hat{X}$ vor 'împinge mai tare', vor avea o influență mai mare asupra dreptei de regresie
- rezultat - dreapta e determinată de câteva puncte leverage
- de aceea punctele leverage tind să se apropie de dreaptă (de fapt dreapta tinde la ele..)
- **Cook's distance**: cum se vor deplasa valorile prezise pentru date dacă modelul ar fi calculat fără acest punct de leverage

Influence measures

- residuals au aceeași unitate de măsură ca Y, nu sunt comparabile între diverse configurații
- am vrea să comparăm reziduul cu o valoare și să decretăm dacă este sau nu outlier
- de regulă se standardizează reziduurile $((x - \mu)/stderr)$; pot fi **internally standardized**, pentru care standard error este calculată cu excluderea punctului reziduu, respectiv **externally standardized**, fără excluderea punctului
- **reziduurile standardizate** pot fi considerate ca fiind parte a unei distribuții de tip Student T sau normale

Leverage points

- punctele situate departe de media $\bar{X}$
- depinde dacă punctul își exercită sau nu proprietatea de leverage
- `http://songhuiming.github.io/pages/2016/11/27/linear-regression-in-python-outliers-leverage-detect/`



dffits, dfbetas și hat values

- diferențele în model cu și fără acel punct
- **dffits**: diferența între fitted value pentru acel punct (valoarea de pe dreapta de regresie, $\hat{Y}_i$) considerând punctul, respectiv excluzîndu-l de la construirea modelului
- vom avea tot atâtea dffits câte puncte avem
- **dfbetas**: cât de mult se schimbă panta dacă acel punct este inclus
- dfbetas o matrice de număr coeficienți $\times$ număr de puncte
- **hat values**: măsoară leverage, cât de mult este deplasat față de medie punctul pe x
- **Cook's distance**: sumarizează deplasările predicțiilor (per global), dacă respectivul punct ar fi eliminat

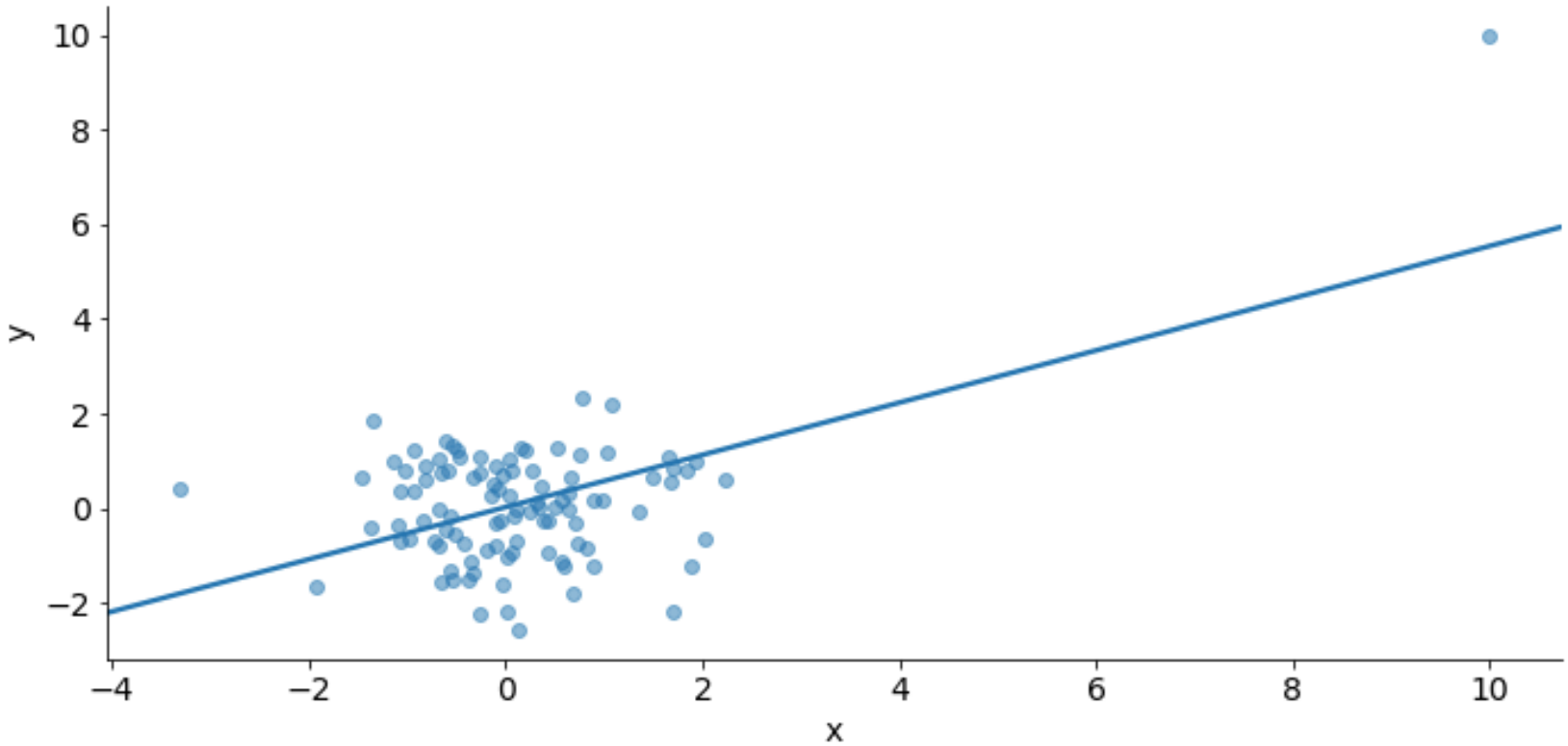
Principii generale

- de regulă evităm să lucrăm cu thresholds absolute pentru a determina puncte outlier
- cea mai comună diagramă: residuals vs. fitted values (dacă există un pattern = poor model fit)
- Q-Q plot: normalitatea, dacă sunt sau nu puncte cu leverage ridicat (erori de introducere?)
- diagnoza influenței (influence plots): care este impactul eliminării unui punct

Influence measures (1)

```
n = 100
x, y = np.r_[10, np.random.randn(n)], np.r_[10, np.random.randn(n)]
df = pd.DataFrame({'x': x, 'y': y})

sns.lmplot(x='x', y='y', data=df, aspect=2, ci=None,  # ci='95'
           scatter_kws={'lw': 1, 'alpha': 0.5})
plt.show()
```



Influence measures (2)

```
model = smf.ols(formula='y ~ x', data=df).fit()
print(OLSInfluence(model).dfbetas[:10, 1])
```

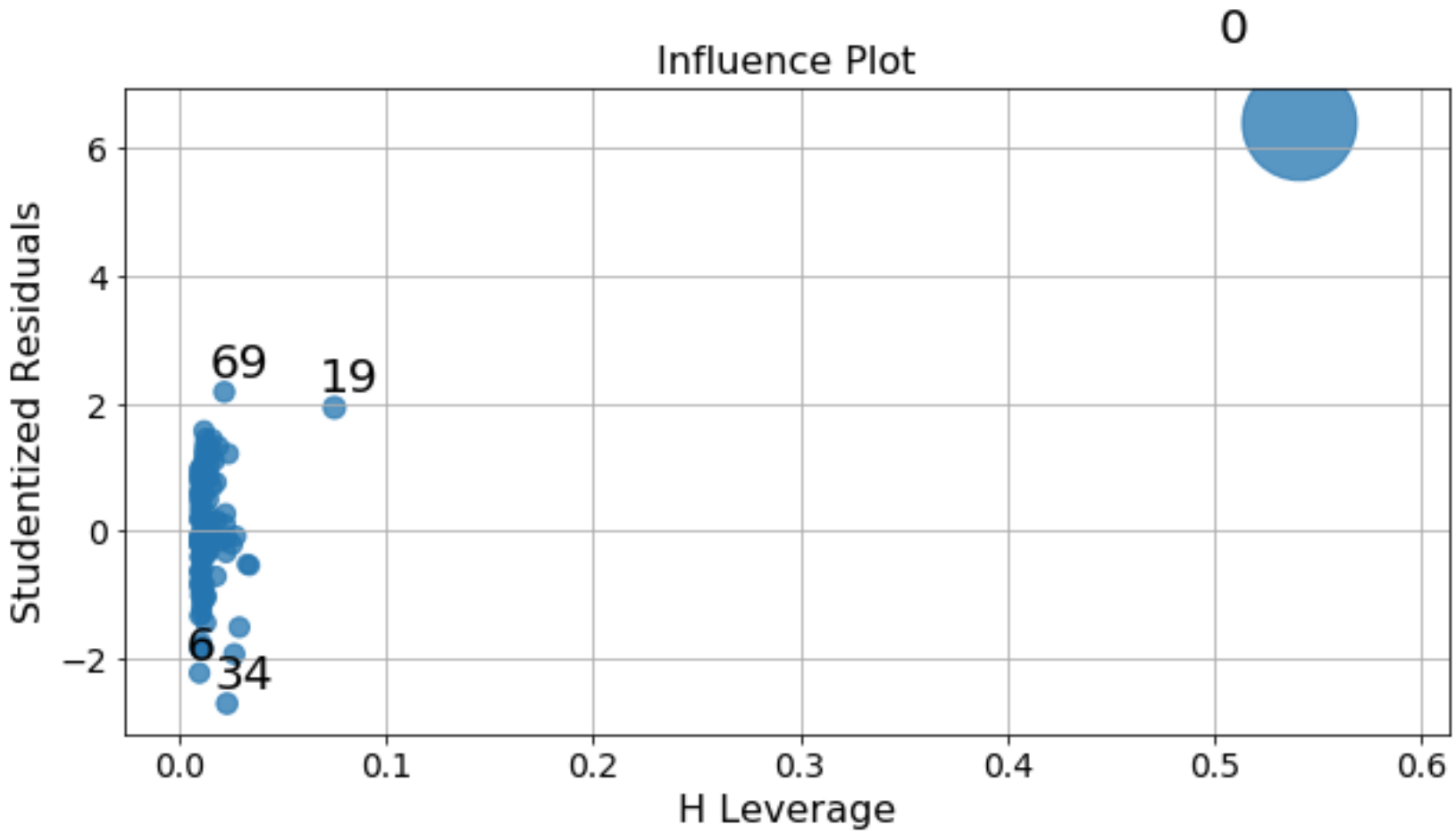
```
[ 6.86557911e+00 -5.95792974e-03  1.07834189e-02 -1.39593485e-02
 -1.65152973e-03  1.64359008e-02  3.07835727e-03  7.99790432e-02
  7.34552393e-02  7.88461847e-03]
```

```
print(OLSInfluence(model).summary_frame().hat_diag.values[:10])
```

```
[0.54106316 0.01058114 0.01218019 0.01023342 0.01120034 0.01049163
 0.00990289 0.03296386 0.01206669 0.0099835 ]
```

Influence measures (3)

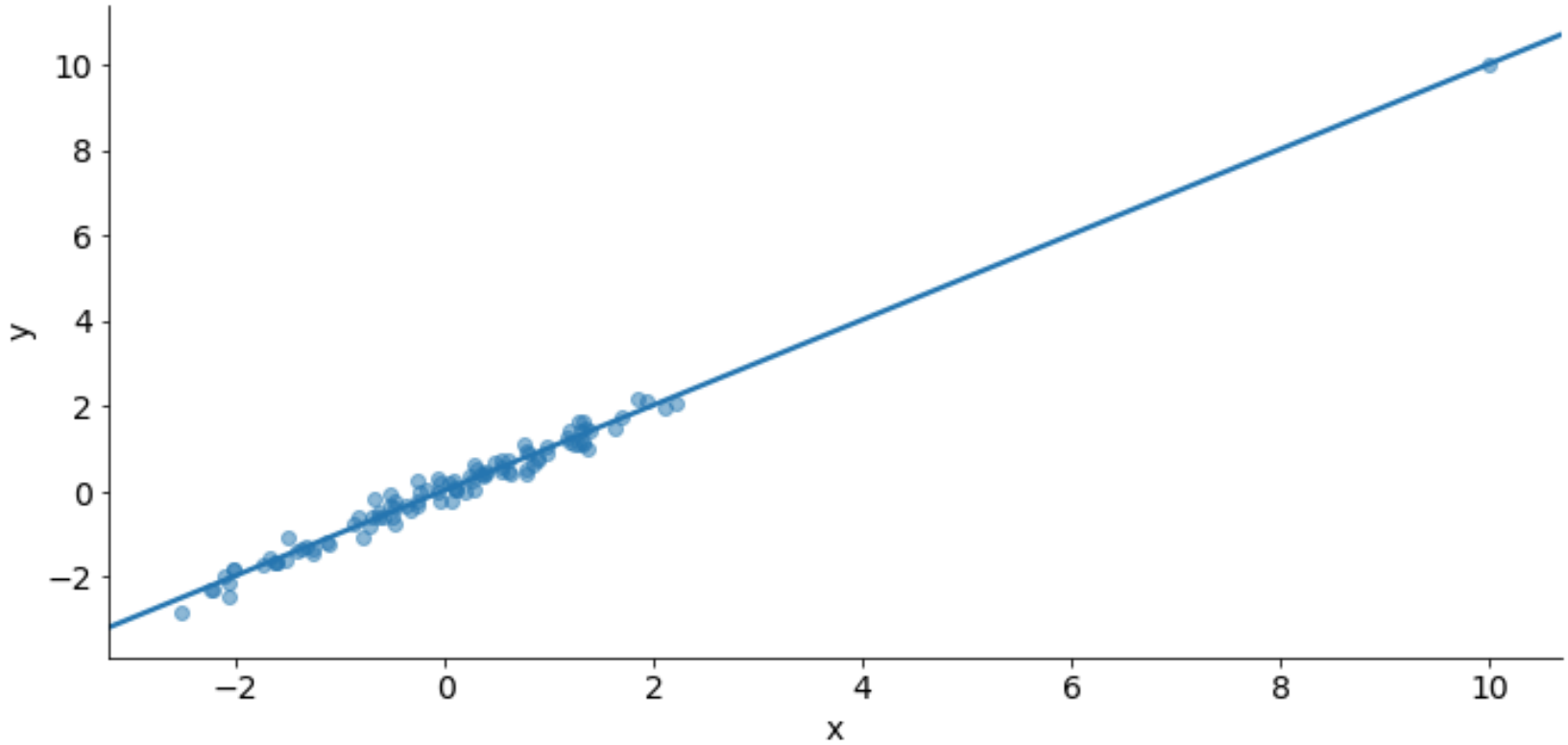
```
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
sm.graphics.influence_plot(model, criterion='cooks', ax=ax)
plt.grid() ; plt.show()
```



Influence measures (4)

```
n = 100
x = np.r_[10, np.random.randn(n)]
y = np.r_[10, x[1:] + .2 * np.random.randn(n)]
df = pd.DataFrame({'x': x, 'y': y})

sns.lmplot(x='x', y='y', data=df, aspect=2, ci=None, # ci='95'
           scatter_kws={'lw': 1, 'alpha': 0.5})
plt.show()
```



Influence measures (5)

```
model = smf.ols(formula='y ~ x', data=df).fit()
print(OLSInfluence(model).dfbetas[:10, 1])
```

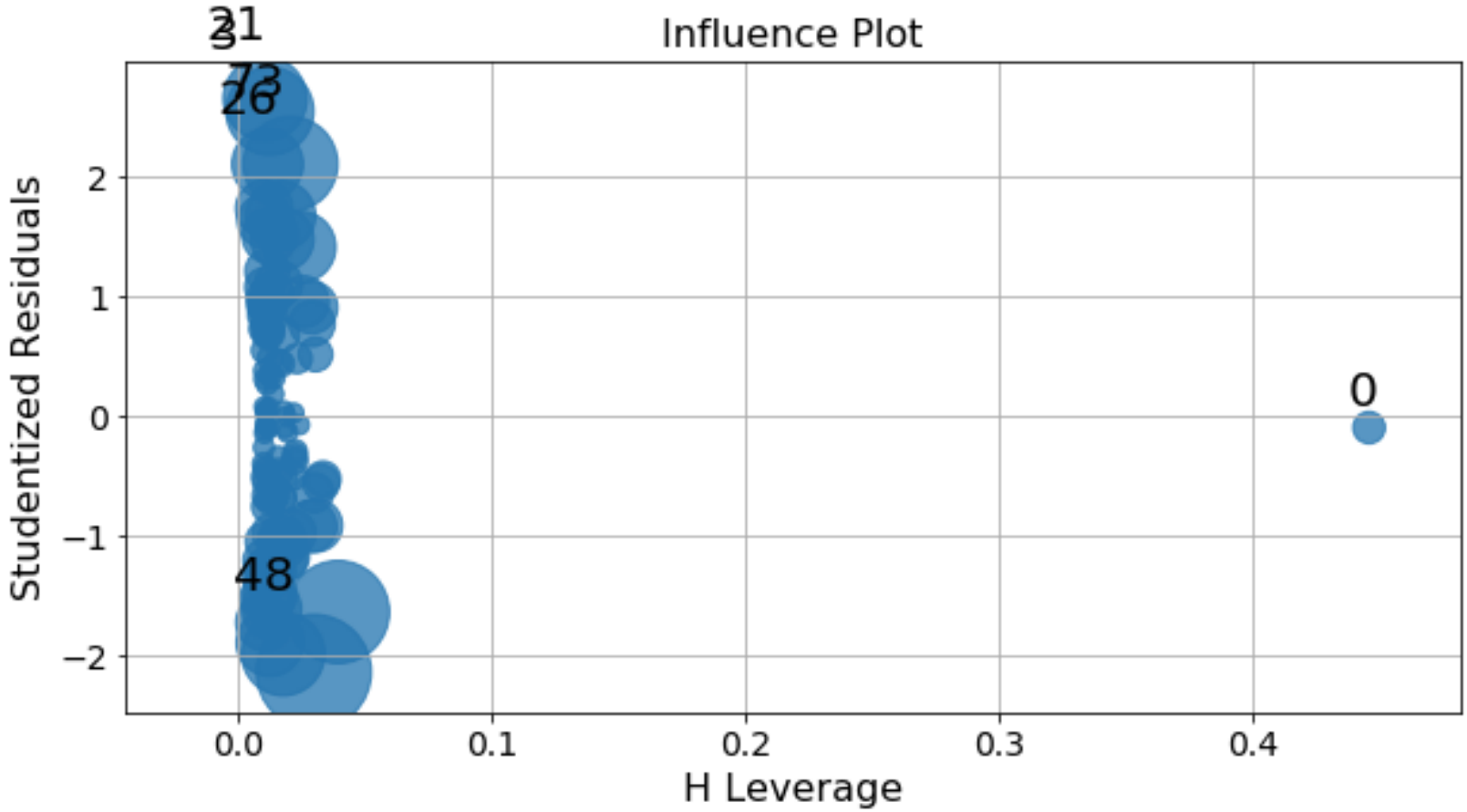
```
[-0.08152765 -0.0014148  -0.00023578 -0.12394666 -0.0598618   0.00408451
 -0.04272144  0.08444678 -0.06929651  0.08503876]
```

```
print(OLSInfluence(model).summary_frame().hat_diag.values[:10])
```

```
[0.44555222 0.01037511 0.01870822 0.01223407 0.01321815 0.01175323
 0.01112607 0.0325016  0.01269735 0.01568622]
```

Influence measures (6)

```
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
sm.graphics.influence_plot(model, criterion='cooks', ax=ax)
plt.grid() ; plt.show()
```



Stefansky (1)

```
st = pd.read_csv('stefanski.txt')
st = np.array([float(y) for x in st.values for y in x[0].split()]).reshape(-1, 5)
st = pd.DataFrame(st)
st.columns = ['v1', 'v2', 'v3', 'v4', 'v5']
st.head()
```

	v1	v2	v3	v4	v5
0	-0.75052	-0.282230	0.228190	-0.084136	-0.24748
1	-0.39380	-0.074787	-0.013689	0.072776	-0.36026
2	-0.15599	0.358390	-0.118070	0.013815	-0.65672
3	-0.68392	-0.059086	-0.060048	-0.231480	-0.03806
4	-0.59474	0.148360	-0.097664	0.667820	-1.05450

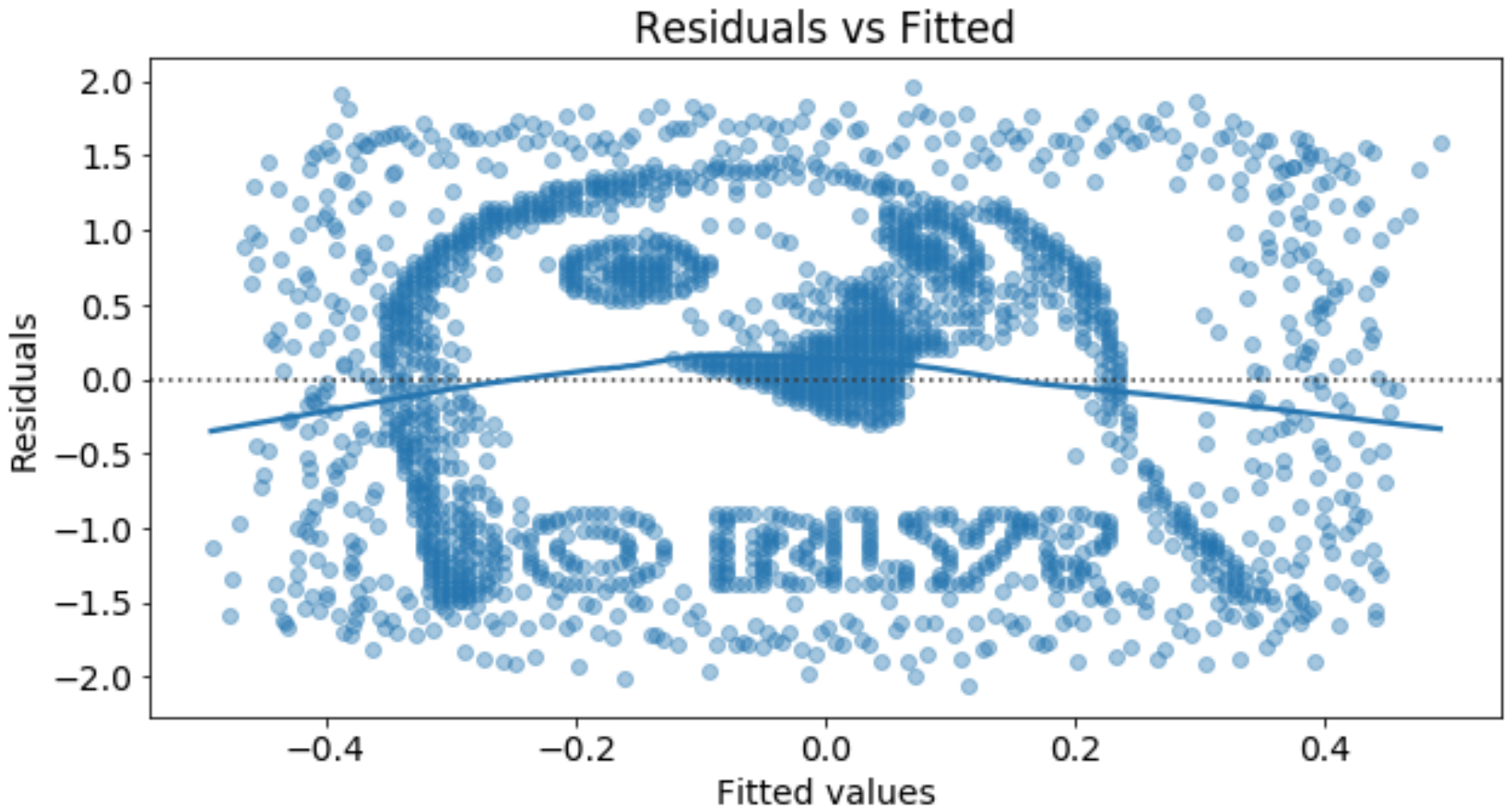
```
model = smf.ols(formula='v1 ~ v2 + v3 + v4 + v5 - 1', data=st).fit()
model.summary()
```

Stefansky (2)

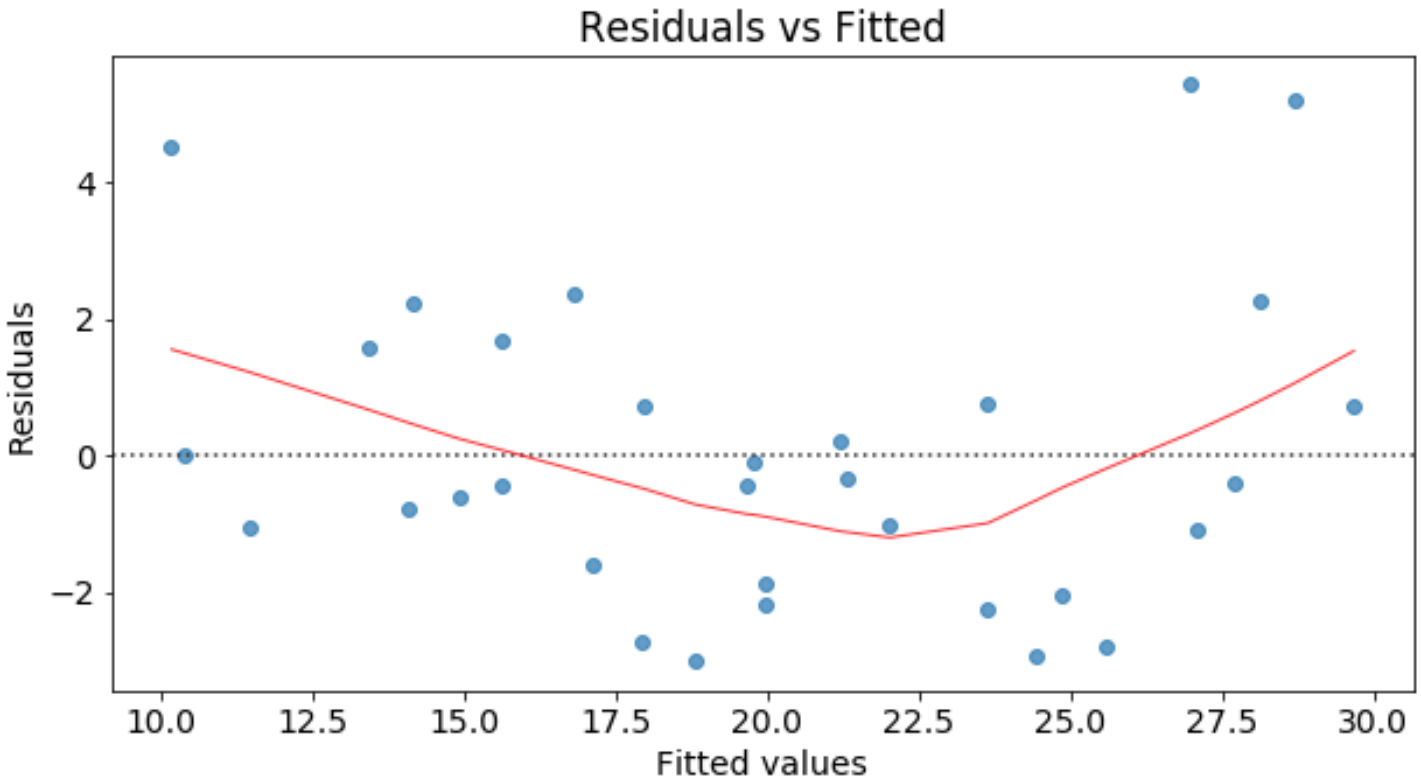
	coef	std err	t	P>|t|	[0.025	0.975]
v2	0.9856	0.128	7.701	0.000	0.735	1.237
v3	0.9715	0.127	7.671	0.000	0.723	1.220
v4	0.8606	0.120	7.197	0.000	0.626	1.095
v5	0.9267	0.083	11.127	0.000	0.763	1.090

Stefansky (3)

```
fig, ax = plt.subplots(1, 1, figsize=(10, 5))
ax = sns.residplot(model.fittedvalues, 'v1', data=st, lowess=True, scatter_kws=
ax.set_title('Residuals vs Fitted') ; ax.set_xlabel('Fitted values')
ax.set_ylabel('Residuals') ; plt.show()
```

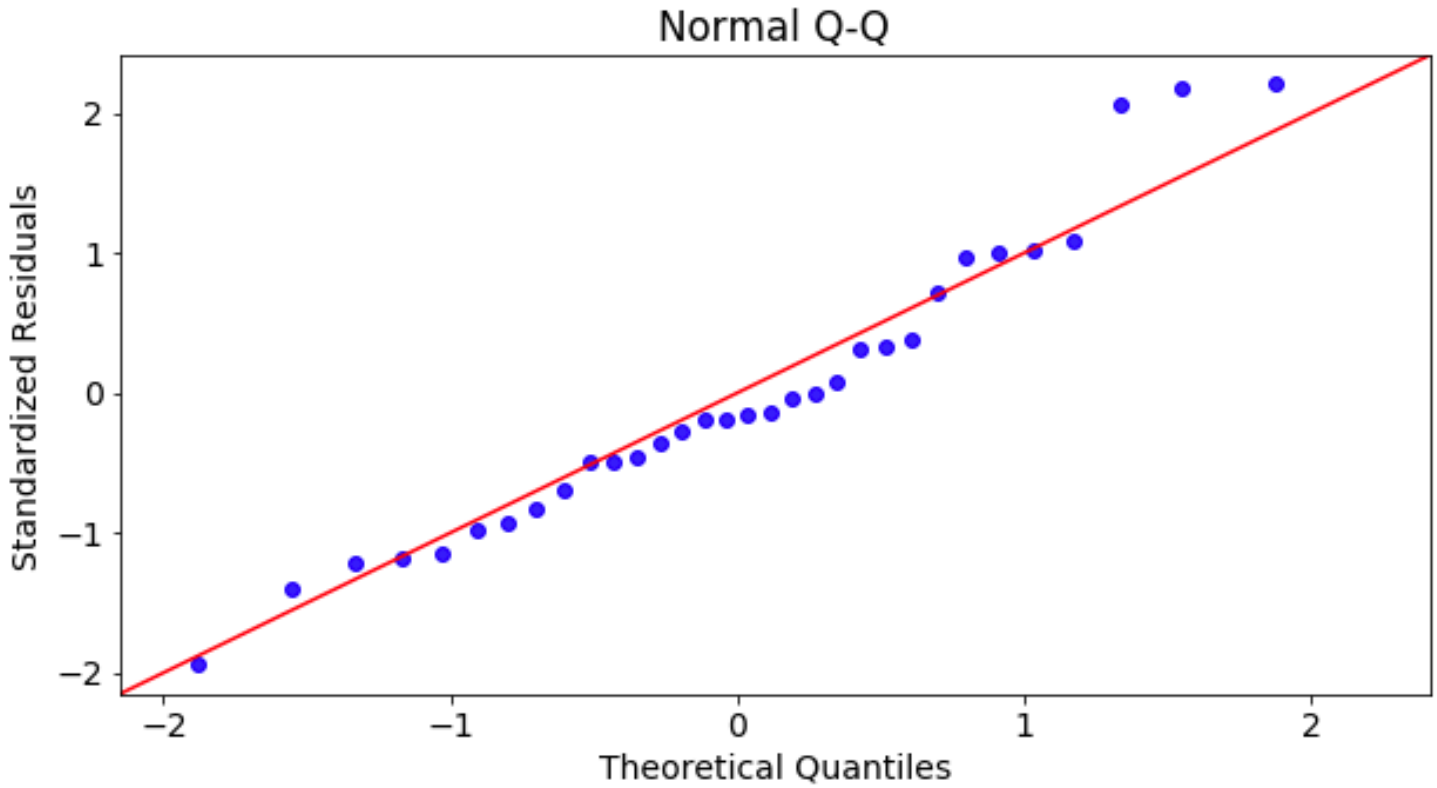


mtcars diagnostic: Residuals vs. Fitted values



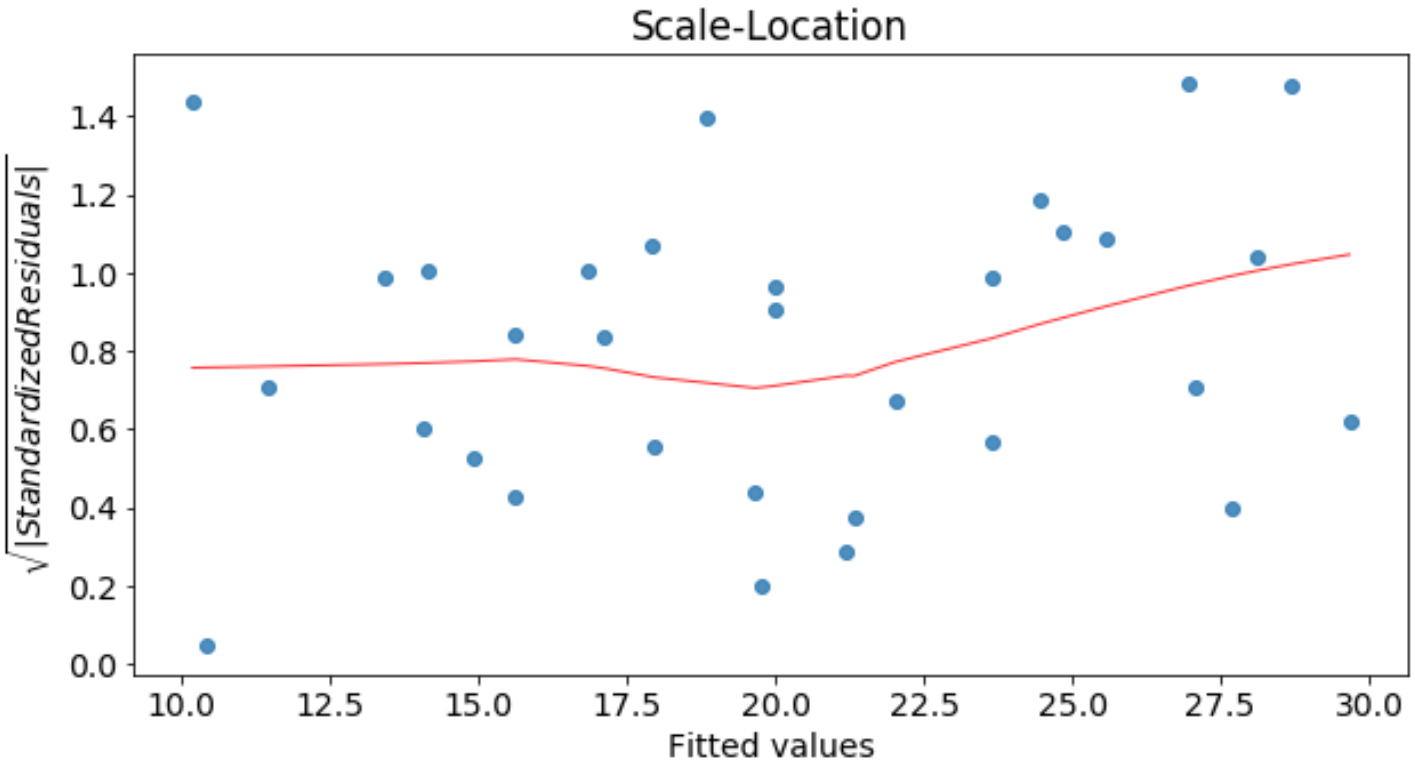
- căutăm pattern-uri, variabilitate neexplicată de regresie
- în general reziduurile au cam aceeași distribuție față de zero
- datele par cam neuniform împrăștiate, linia ar fi trebuit să fie dreaptă
- linia e o parabolă, pare o relație neliniară neexplicată de model

mtcars diagnostic: Q-Q plot



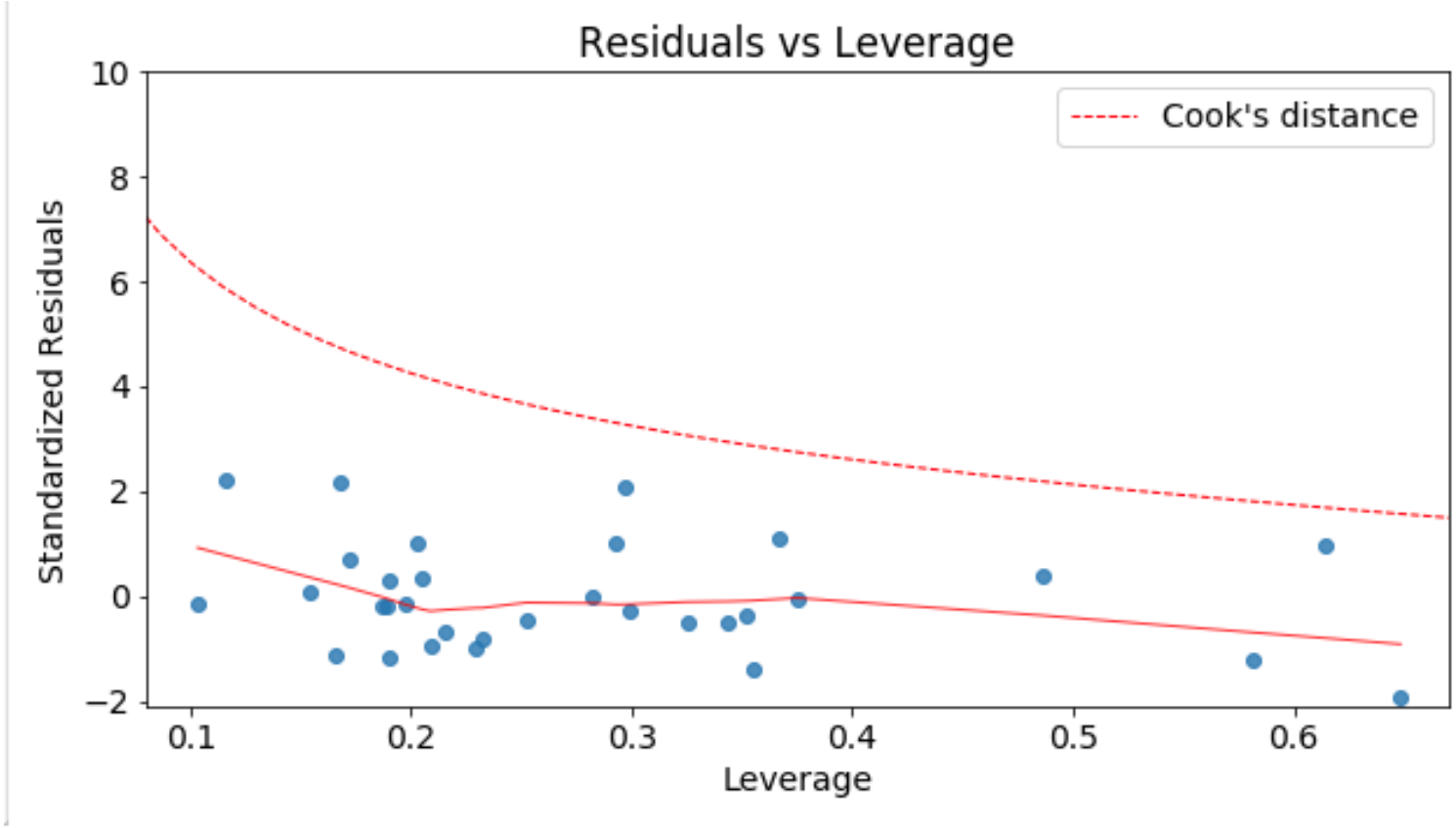
- testează normalitatea reziduurilor
- dacă reziduurile nu sunt distribuite normal, vom observa o abatere puternică de la dreaptă; în acel caz n-ar strica să ridicăm histograma
- `https://data.library.virginia.edu/diagnostic-plots/`

mtcars diagnostic: Scale-Location plot



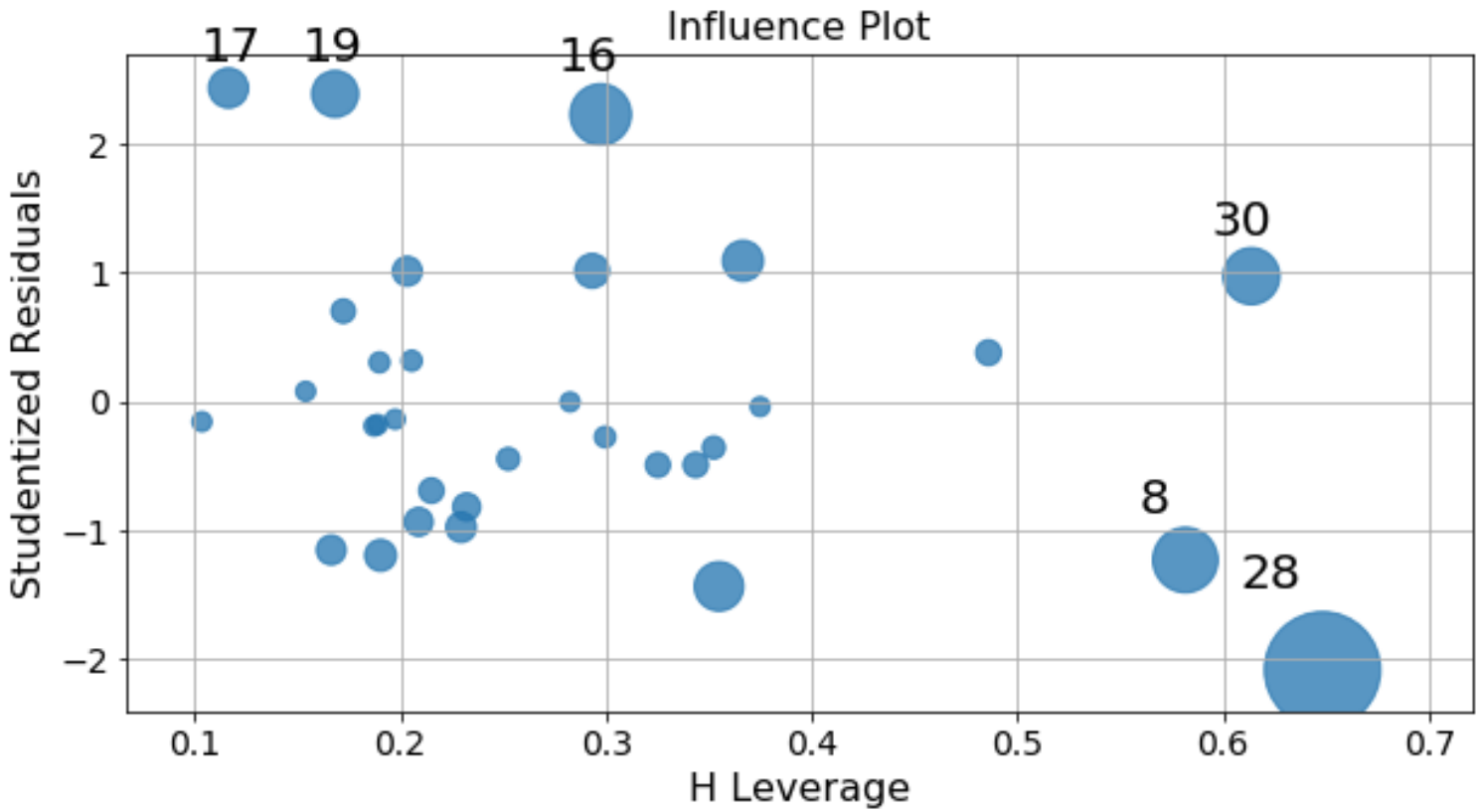
- distribuția in/egală a reziduurilor de-a lungul domeniilor predictorilor
- item ar trebui ca datele să fie 'aruncate' random
- dacă apare heteroscedasticity, linia roșie se înclină

mtcars diagnostic: Residuals vs. Leverage



- căutăm puncte extreme, dreapta-sus vs. stânga-jos, care sunt în afara distanței Cook, pentru că excluderea lor modifică puternic modelul
- `https://data.library.virginia.edu/diagnostic-plots/`

mtcars diagnostic: Residuals vs. Leverage (2)



- căutăm leverage points cât mai departe de zero (dreapta-sus / dreapta-jos)