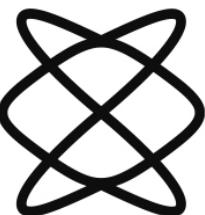


הפקולטה למדעים מדויקים
אוניברסיטת תל אביב
ע"ש רייןmond וברלי סאקלר

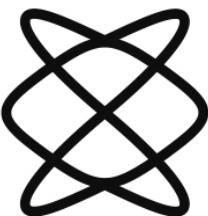


זיכרון
מלחתת חרבויות ברזיל

מבוא למדעי הנתונים (0300) (גרסה ארוכה)

מרצה: גיורא שמחוני
תשפ"ד, סמסטר ב' (2024)

מסכם: רועי מעין



The Raymond and
Beverly Sackler Faculty
of Exact Sciences
Tel Aviv University



פרק 1 - מבוא

3	מבוא
10	ענון הסתבותות
18	חקר נתונים.
26	PCA

פרק 2 – שיטות הסתבותיות

35	הסקה סטטיסטית – חלק א'
45	הסקה סטטיסטית – חלק ב'

פרק 3 – מודלים של למידת מכונה

58	רגסיה
72	שכנים ועצים החלטה
78	שיטות אנסמבל

פרק 4 – שיטות מודרניות

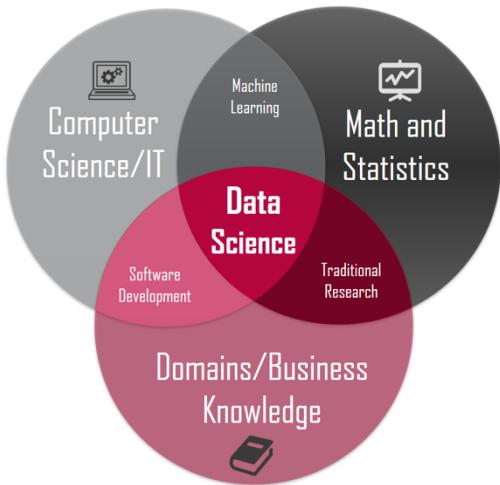
85	מבוא לרשתות נוירוניים
97	רשתות קונבולוציה
103	התיאוריה של מידול לחיזוי



1 – מבוא

מבוא

מבוא



מה זה DS: פירוגמה שsspאלת רעיון ממספר תחומיים, ושם את הדאטה במרכז. כולל את האפן שבו אנחנו אוסף נתונים, מנהחים אותם, מפיקים מהם תובנות, מודלים אותם, והכל במטרה להביא תועלות לביעות פרקטיות בעולם האמיתי.

לא תמיד ברורים הגבולות בין DS לבעיה סטטיסטית או מתמטית. ניתן להסתכל על תחום זה כאשר הוא משלב 3 תחומיים:

1. Domain – בעיה DS לרובה נוגעת בתחום ידע ספציפי, ובעיה אמיתית שמעסיקה אנשים או חברה.
2. Math and Statistics – הדריך לטפל בעיה זאת היא לנסוט להחיל עליה כלים מתמטיים וסטטיסטיים, بما שהוא במחקר מואז ומתמיד.
3. CS – אנחנו משלבים מושבר לה כלים של הנדסת תוכנה ולמידת מכונה.

מרכיבים של פרויקט DS מול ל�� אפשר:

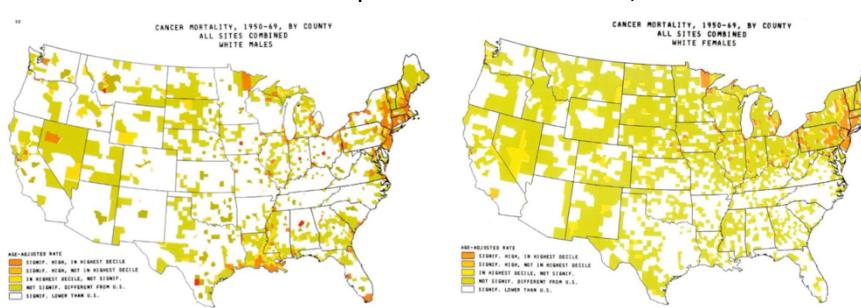
1. הגדרת הבעיה, והבנת אילו נתונים צריכים כדי לפתור אותה.
2. איסוף ויצירת הנתונים – לפעמים אפשר פשטוט להויריד מטור DB מסודר של חברה או ממשלה, ולפעמים איסוף הנתונים הוא דבר הרובה יותר אקטיבי כמו עיבת סקרים. זה יכול להיות בתיבת קוד שיוריד נתונים מאלפי דפים באינטרנט.
3. ארגון, הבנה וייצוג הנתונים – מה עושים עם כל הדאטה הזאת? במגוון עצומות של משתנים. איך ניגשים נתונים אלה, מציגים אותם, ומבינים אותם טוב יותר.
4. ניתוח הנתונים – יכול לבוא בצורה של בדיקת השערות סטטיסטית, או מיזול וחיזוי על dataset שהמודל לא ראה, מtower מטריה שהמודל ירוץ בסביבת prod ויתן חזוי שאנו יכולים לכמה מראש מהטיב שלו – בכמה מהדוגמאות שהוא רואה המודל צודק.
5. תוצאות ומסקנות – לפעמים התוצאה הוא תשובה לשאלת (אם קיים קשר בין משתנה אחד לאחר), או מודל/אלגוריתם שאפשר להריץ עליו נתונים חדשים ולקבל חזוי.

דוגמאות:

1) **Higgs boson search:** בעיה מתחום פיזיקת החלקיקים. במשך שנים היה קיומו של החלקיק בגדר השערה בלבד. ב-2012 הודיעה קבוצה של מדענים שעבדה במאיצ' החלקיקים בcern (שויז'), כי הם משוכנעים במידה רבה שהחלקיק שאות קיומו ניבאו שנים לפני כן, סוף סוף נמצא. אין נמסגר את הסיפור כפרויקט DS:

- בעיה: מציאת החלקיק.
- הדאטה: מדידות של מסות של חלקיקים מניסויים מהמאיצ'.
- ארגון והבנה: המאיצ' מייצר بمיליארד אינטראקציות בין חלקיקים בשנייה. כאן נוצר סיוור מוחות בין פיזיקאים ומדען מחשב.
- ניתוח: בדיקת השערות – האם peak ישאנו רואים בגרף כלשהו הוא אכן עדות לקיומו של חלקיק Higgs boson.
- תוצאות: פרס נובל לפיזיקה (2013).

2) **חיזוי סרטן:** מתחום הרפואה ומערכות הבריאות. לפניו מפה של שיעור התמותה מסרטן לפי מחוזות בארה"ב בשנות ה-50 וה-60 של המאה ה-20. בכל שבעה מחוז חם יותר, כך חריג יותר שיעור התמותה מסרטן במחוז זה. מצד שמאל שיעורי התמותה של גברים, מצד ימין נשים. ההבדל הבולט ביותר בין גברים לנשים בתמותה מסרטן, הוא שיעור התמותה החיריג של גברים לאורך מדינות החוף הדרומיות (פלורידה, לואיזיאנה, וטקסס). הסברת הרוחות ביום היא שבאזורים אלה היו ריכוזים גדולים של נמלים ומספנות. באופן טבעי בשנים אלו, עבדו בנמלים יותר גברים, והם חחשפו לחומר מסרטן במיזוח – אזבסטן.



אפשר לחשב על פרויקט יותר רחב – מה הם גורמי הסיכון לסרטן ולמחלות אחרות, מתוך רשומות רפואיות, בסקלאלת ענק:

- בעיה: זיהוי גורמי סיכון למחלות.
- הדאטה: רשומות בתי חולים, בדיקות מעבדה, סקרי בריאות, ביקורי חופה, תמונות וסקירות שמגינות מרנטגן/MRI.
- ניקוי נתונים: קשה לעבד עם הדאטה המקורי – מדובר בטקסט חופשי, פורמטים שונים, תרומות שונות.
- ארגון והבנה: התחום של Exploratory data analysis (EDA) מחפש את הוויזואליזציה המתאימה להסתכלות על הנתונים, לפי מה לצבעו את המוחוזות (מין, גזע, דת).
- ניתוח: יש לנו השערה של גורמי הסיכון, ונctrיך לבדוק סטטיסטית מול השערות אחרות.
- תוצאות: נקווה שיביא לשינוי חיובי במדדיות בריאות הציבור, והתנהגות הציבור.

פרויקט WIKIART:



Claude Monet, Water Lilies, 1919



Vladimir Makovsky, Brew Seller, 1879

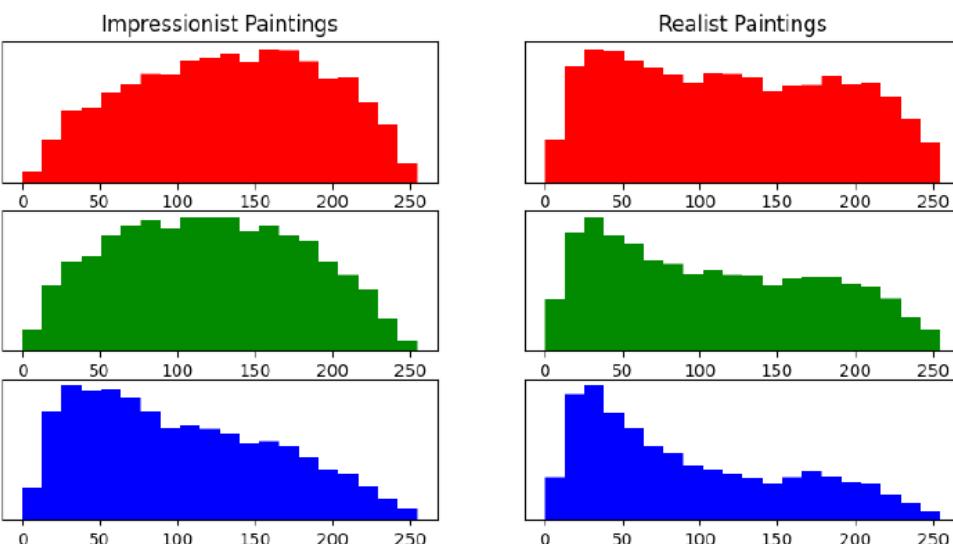
לאורך ההיסטוריה ציירים ציירו בסוגנותות רבים. האם נוכל לבנות מודל שיבדל בין סוגנות ציור? לצורך האתגר, לקחנו סוגנות ציור שאינם מאוד רחוקים זה מהה, ויש לנו לפחות דוגמאות מכל אחד, ריאליזם ואמפרסיוניזם:

- הסגנון הריאליסטי הגיע קודם באמצע המאה ה-19, בשיאו של המהפקה התעשייתית. הציירים הריאליסטים ביקשו ליצור אנשי עבודה רגילים, מכל המעמדות, בסיטואציות יומיומיות מהרחוב בלי לעשות להם אידאליזציה ובלי לipyות את פניו הדברים.
- התנועה האימפרסיונית נחשה לתנועה שקמה נגד התנועה הריאליסטיבית, בסוף המאה ה-19. הציירים לא שאפו לתאר במדויק ובטכניקה מושלמת את אשר עיניהם רואות. הם נטו ליצור בצבעים בהירים יותר, תוך התמקדות באפקטים של אור, ומשיכות מכחול חופשיות. צייר בצבעוניות נופים וצמחיים, אך גם סיטואציות יומיומיות עליהן כמו פיקניקים וגידול ילדים.

איך נשיג לפחות ציורים כאלה? אתר שמאחסן למעלה מ-100,000 ציורים מתקופות שונות בראשון פתוח, הוא WIKIART. אי אפשר סתם להורד ממנו קובץ של X תמונות לפי בקשה. הן מסודרות לפי תקופה, אמנים וסוגנות. אין מגיינים מדף HTML לאוסף תמונות על המחשב שלנו שהן אינם אלא אוסף של מספרים, שעלינו נוכל להפעיל את המודלים שנלמד.

ארגון וניקוי הנתונים: אם נוריד סתם את התמונות מהאתר, נקבל קבצי זבל כמו `slogos` למיניהם. איך נאחסן את כל התמונות? אם היינו רצים 100,000 תמונות של (1000 X 1000) פיקסלים עם 3 שכבות צבע, היינו מגיעים לגודל של 300GB.

הבנת הנתונים: נctrיך להבין מה הנתונים בדיק אומרים. מה הכוונה ב-3 שכבות צבע? כל תמונה יכולה להיחשב כמערך תלת-ממדי של מספרים בין 0 ל-255. כל שכבה צבע מייצגת עד כמה הצבע זהה מתבצע בפייל נטען. 3 הצבעים הם: אדום, ירוק, וכחול = RGB. ניתן ליצור היסטוגרמות של במות האדום, הירוק והכחול על פני מוגדים של ציורים ריאלייטיים, ואימפרסיוניסטיים. בראלייטיים יש יותר פיקים באזורי הנמרך בכל שלושת הצבעים – נצפה שהם יהוו מעט קודרים יותר.



בנייה מודל חיזוי: בגיןה הקלאסית, נקדים חלק מהצירם להיות מדגם למידה (train set), אך המדי לטיב המודל יהיה צירום שהמודל לא ראה (test set). בגיןה המסורתית בתחום הסטטיסטייה ולמידת המכונה – נגידו מראש את המשתנים שמעוניינים אותנו, ומניחים מראש שיש מספר פרמטרים שצורך להעיר (כמו ברגסיה). בשיטות מודרניות יותר (boosting, DL) נרצה לתת למודל ליצור את הפרמטרים שהוא לומד עליהם, שיטות אלו מתאימות במיוחד לתמונות.

דוגמה למודל: כאן יש לנו מודל רגסיה לוגיסטי, שאומן על כמות האדום, הכחול והירוק בסט של 2000 צירום אימפרסיוניסטיים ו- 2000 ריאליסטיים. הוא מביא לנוסחה פשוטה. הוא נותן דיק של 56%. כמה גובה אחוז הדיק שאנחנו שואפים אליו? 75-80%.

$$\text{Predict 'realist' if: } 1.28 - 0.012 \cdot \text{red} - 0.001 \cdot \text{green} + 0.004 \cdot \text{blue} > 0$$

קווי 1

שאלה 1: מהו כל הדאטא סיננס העיקרי בשלב ניתוח הנתונים במחקר של חלקיק בזון היגס?

- בדיקת השערות סטטיסטיות.
- למידת מכונה.
- ויזואליזציה וקומון סנס.
- ניתוח אשכבות.

שאלה 2: לפניך הציור "אישה עם מחרוזת פנינים בתא הצפייה" של מרקי קאסאט. סביר לומר שהזה ציור אימפרסיוניסטי? נכון.

שאלה 3: בשקף 19 מוצג מודל ליניארי פשוט לחיזוי האם ציור הוא מסגן ריאליסטי או אימפרסיוניסטי. לפי מודל זה אם לצייר יש רמות אדום, ירוק וכחול ממוצעות של 100 כל אחת, נרצה שזה ציור אימפרסיוניסטי? לא נכון, נחשב ונקבל:

$$1.28 - 0.012 \cdot 100 - 0.001 \cdot 100 + 0.004 \cdot 100 = 1.28 - 1.2 - 0.1 + 0.4 = 0.38 > 0$$

ולכן נרצה ריאליסטי.

תרגול 1 – Python וספריות חשובות

רענון פיתוח:

ביסיס: השמה, הדפסה, type, חלוקה (/ לשברים לעומת // לשלים)

מחרוזות:

- לשים לב שהן immutable, לא ניתן לשנות תווים במחרוזת
- id – הכתובת בזיכרון שאליה מצביע המשתנה

מבנה נתונים בסיסיים:

- רשימה – לשים לב שהוספה באמצעות + מוסיפה כל איבר בתוך אובייקט שהוא Iterable. אם מדובר במחרוזת, כלתו בה מתווסף בפרט לרישימה.
- מילון – אם רצים להסתמך על סדר האיברים במילון, ניעזר ב-OrderedDict.
- סט – מאפשר הרבה פעולות מוכנות על קבוצות.

ולואות ותנאי בקרה:

- תנאים – if, else, elif
- לולאות – for, while, break, continue
- שימוש ב-range ו-enumerate – List comprehension

פונקציות:

- לשים לב ל-scope, אם פונקציה לא מזהה משתנה global ב-local scope היא יכולה לקחת אותו מה-scope וזה יכול לגרום לבאגים.

Class	Description	Immutable?
<code>bool</code>	Boolean value	✓
<code>int</code>	integer (arbitrary magnitude)	✓
<code>float</code>	floating-point number	✓
<code>list</code>	mutable sequence of objects	
<code>tuple</code>	immutable sequence of objects	✓
<code>str</code>	character string	✓
<code>set</code>	unordered set of distinct objects	
<code>frozenset</code>	immutable form of set class	✓
<code>dict</code>	associative mapping (aka dictionary)	

רענון numpy:

בסיס:

- נוצר מערך מטיפוס `ndarray` מתוך רשיימה באמצעות `np.array()`.
- לוקטור זהה שיצרנו יש את השדה `shape` שומרה את המימדים שלו.
- על שני מערבי קה אפשר לבצע חיבור, מכפלת סקלרית, מכפלת `element-wise` (בעזרת `*`) או מכפלת פנימית (בעזרת `@`). אפשר לקרוא לפונקציות ישירות באמצעות קריאה `l-pd.dot()` או `l-pd.add()` וכו'.
- לא ניתן לבצע `add` בין שני וקטורים ממימדים שונים.

מטריצות:

- נוצר מטריצה לפי שורות, ונראה כי `shape` מראה לנו את מימדי השורות ועוד את מימדי העמודות.
- כאשר נבצע `slicing` המימד הראשון יהיה השורות, והשני יהיה העמודות. בשים לב כי אנחנו מקיבלים `view` אל אותם הנתונים, ואם נשנה אותם מה"עתק" שיצרנו (`slice` בשם `B`), נשפייע גם על המטריצה המקורי `A`. אפשר לפתור זאת באמצעות `copy` (של `Q` או של פיתון באופן בליל שמבצע `copy` deep).
- פלטור/תנאים: אפשר להדפיס תוצאה של שאלתה בוליאנית, נניח `(A>8)` והוא מדפיס את כל ערכי המטריצה, ומחליף כל אחד שעונה על התנאי ב-`True` וכל אחד אחר ב-`False`. אפשר גם להדפיס את הערכים המקוריים כאשר נסתכל על `[A>8]` בລומר המטריצה `A` בכל המיקומים שבהם הערכים גדולים מ-8.
- פעולות על מטריצות: ניתן לקבל את המטריצה ההפכית, לבצע `transpose`, מכפלת וכו'. **סכום על סכום על פניו בעמודות (לכל שורה) בוצע על ידי `sum(axis=0)`** בעוד שוכם על פניו בעמודות (לכל שורה) בוצע על ידי `sum(axis=1)`. אותו דבר אפשר לעשות על `mean` (שורות ועמודות).

יצירת מערך דוח:

- אפשר להשתמש ב-`range` שמאוד דומה ל-`range`.
- `linspace` מאפשר ליצור `X` מספרים שמספריהם בקרה לינארית את הקצאות המבוקשיות.
- `zeros` יוצר מערך שכלו אפסים, `ones` עבד אחדות.
- `eye` יוצר מטריצת יחידה, `diag` עבר מטריצה אלכסונית בכללית.
- אפשר ליצור באמצעות `random` ערכים אקראיים מהתפליגויות כמו `normal`, `uniform` וכו'.

הערות נוספת:

- ישנו מבחן עצום של פונקציות מתמטיות בספריה של `scipy`. אם יש צורך בדברים נוספים אפשר להסתכל על הצצה קלה למה שאפשר לעשות עם `matplotlib` תוך שימוש ב-`linspace` שראינו קודם.

axis 1

	col-0	col-1	col-2	col-3
row-0				
row-1				
row-2				

```

[ ] A = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
print(A)
print(A.shape)
print('A has %d rows and %d columns' % A.shape)

[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
A has 3 rows and 4 columns

Again, be sure you know how to subset:

[ ] A[0, 0]
1

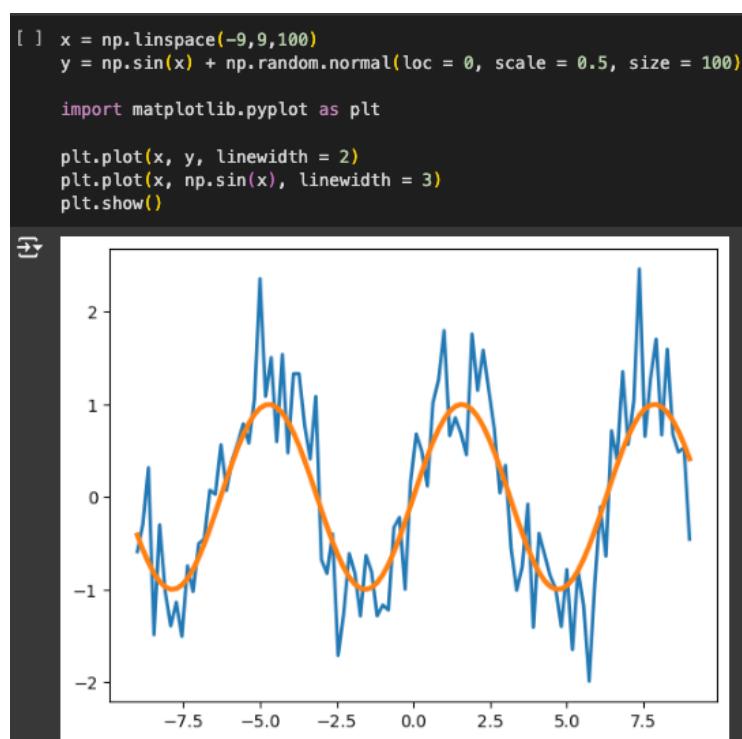
[ ] B = A[:2]
print(B)

[[ 1  2  3  4]
 [ 5  6  7  8]]

[ ] B = A[:2, 1:3]
print(B)

[[ 2  3]
 [ 6  7]]

```



```
[18] # create with a dict
d = {'Chicago': 1000, 'New York': 1300,
      'Portland': 900, 'San Francisco': 1100,
      'Austin': 450, 'Boston': None}
cities = pd.Series(d)
cities
```

	0
Chicago	1000.0
New York	1300.0
Portland	900.0
San Francisco	1100.0
Austin	450.0
Boston	NaN

dtype: float64

:pandas**:Series**

מייצג לנו וקטורי של נתונים. אפשר ליצור מטור רשימה של ערכים כלשהם. בשנדייפס מיד בצד שמאל נראה את האינדקס של כל ערך (באופן אוטומטי). לכל series יש את השדות index, shape, dtype.

- אפשר להעביר index ברצוננו בתור פרגמטר ביצירת ה-series.

- אפשר גם ליצור מ-dict, אז keys יהיו-hikeys.

- כדי לגשת לערכים עצם, באמצעות השדה values מגיע ל-np array גען של הערכים.

- גישה:** ניתן לגשת (באיילו כמו במילון) לערך של index ספציפי, או בכמה indices (ואז מקבל בחזרה series בהתאם). מקובל יותר להעתמש במתודה loc שמקבלת סוגרים מרובעים [], ומחזירה תת-series בדומהה.

- ניתן לסמן באופן בולאיין מי NaN ומילוי לא, על ידי המתודה isnull(). זה יכול להיות שימושי להצבת ערכים חסרים (כל מה ש-null נציג במקומו X).

- פונקציה שימושית היא value_counts (במה פעם מופיע בערך).

- הוא mutable Series, ניתן לשנות את הערכים בו והמקומ בזיכרן לא משתנה, מדובר באותו האובייקט.

- אפשר לבצע איטרציה על ערכים בס-series אבל זה לא יעיל. עדיף להשתמש בפורמט של [v>s].

:Data Frame

ניתן ליצור בצורה ידנית באמצעות dict שמאיפה key (עמודה) לכל הערכים שלו series (בלשנו). מתבצעת השלמה לכמות השורות המתאימה עבור כל עמודה, בולמר foo-1 ישובלו עבור כל השורות.

- index, columns, values הם שדות שימושיים.

- אפשר גם ליצור ממשתנה dict קיים באמצעות .read_csv(), או מטור קובץ from_dict()

- על df חדש שנוצר נבצע head() כדי לראות את 5 השורות הראשונות, () עבור תיאור קצר של הדאטא עם סטטיסטיות שונות, -info()

- ניתן למיין עם sort_values() כאשר נשים לב שהאינדקסים נשמרים ב-df.

:Subsetting

- כדי לגשת לעמודה ספציפית משתמש-[X] ואפשר גם לגשת בתור שדה ישירות ל-X. מה שחוור עבור עמודה אחד הוא series, אם נבחר מספר עמודות נקבל df.

- באמצעות loc המימד הראשון הוא לפי שורות, והפער הטוויה הוא כולל (עד אינדקס 3 כולל), ובמידה שנייה יוכל לבחור את העמודות הספציפיות.

- אפשר גם לבצע שאלות יותר מורכבות על ידי ('m') str.startswith('m') columnstr.startwith('m') למשל, או ff['month'].isin()。

- באמצעות loc או אנחנו בוחרים לפי מיקום (לא כולל), הוא עובד לפי integers, בעוד loc בעוד לבי האינדקס עצמו.

- יש את המתודה query() שמקבלת גם תנאי לפטלור.

:הוספת עמודות חדשות:

- נפתח שם חדש בתוך [] ונשים בו ערכים כלשהם. אפשר להתבסס על ערכים מעמודה אחרת + [X][fff].
- אפשר להיעזר בפונקציית lambda שמודגרת on-the-fly על הפלטורה x (שהוא שורה ב-df שלנו), ניגש לשדה temp ווחזיר ערך חדש. ממיררים מצלויים לפורמייט, ואת כל זה נשים בתוך assign().

- אפשר לשירות ערכים לפי מילון מוגדר מראש, ולהעתמש ב-map() של series. כך נῆפה בין 1 ל-1 וכו'.

```
def march_fridays(row):
    return row['month'] == 'mar' and row['day'] == 'fri'
forestfires['is_march_friday'] = forestfires.apply(march_fridays, axis=1)
```

- המתודה apply מתקבל פונקציה של ממש, אם נעביר axis=1 הוא תעביר שורה-שורה. את התוצאה שלה נשמר בעמודה חדשה.

:Group By

לאחר ביצוע הפעולה groupby נקבל אובייקט DataFrameGroupBy או SeriesGroupBy. ניתן לבצע עליו מספר

```
# mean salary per department (notice the specification of ['salary'])
df_chicago.groupby('department')[['salary']].mean().head()
```

פעולות, כאשר השימוש בויתר הוא agg.

מבצע() על העמודה הרצויה, ואז נוכל לבצע

size (במota הרשותות הכלולית) לקבלת series. במאtuות count נקבל כמה יש בכל עמודה, ללא NaN. כאן נקבל כבר df.

```
# groupby department, each column with the number of people and the average salary
df_chicago_avg_salary = df_chicago.groupby('department').agg({'salary': [np.size, np.mean]})
```

אפשר לשער כל שדה לפעולה agg ולשיך ב- size (במota הרשותות הכלולית) למשל גם ממוצע וטוחנו רוחים לבצע עלי' (cotubits של הפעולה, כמו 'mean'). אם נרצה יותר מפעולה אחת (למשל גם ממוצע וגם

במota כללה) נצטרך להשתמש בשם המתוודה mismatch (np.mean). נשים לב שאוטומטיות department נכנס לנו כ- index ובנוסף, העמודות הן בעט מטיפוס MultiIndex שכך יש לנו גם את salary וגם את mean(index.size, mean(index)). אפשר לפתור את זה על ידי JOIN בין השמות וערכות הערך של columns. אפשר לפתור את עניין ה-index באמצעות הפעלת הפטט as_index=False.

```
df_chicago.groupby(['department', 'title'], as_index=False) \
    .agg({'salary': np.mean}) \
    .rename({'salary': 'mean_salary'}, axis=1)
```

אפשר מבון גם בגישת pipeline כמו שראינו קודם.

-

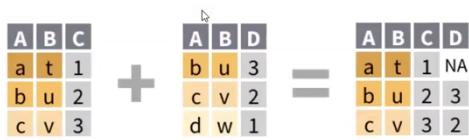
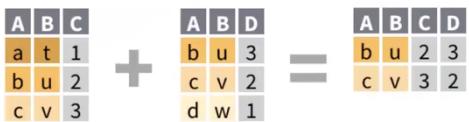
-

-

-

-

-

:Join Concatenate

inner_join()

left_join()

right_join()

הרבba פעמים נרצה שלב df שmaguiim מכמה מקורות לאחד, בין אם לחבר אותם (concatenate) או לשלב אותם לפי מפתחות משותפים (join).

חיבור פשוט באמצעות הפונקציה concat עם הפטט axis=0 לפי שורות או axis=1 לפי עמודות.

ליצוע join יש כמה אפשרויות:

- Inner – חיבור, שורות משותפות בלבד.

שבצד שמאל כפי שהוא, רק העמודה S (ימין) מצטרפה. ערכים שאין עוברים חיבור יקבלו NA.

- Left – לשבל את שמאל עם צד ימין כפי שהוא.

הפונקציה merge מקבלת הפטט how את אופי התווך.

הפטט how מגדיר את המשטנה שהוא המשותף לשני ה-df-ים (יגלה בעצמו אם לא נזכיר אותו).

```
# standard inner join
pd.merge(hospital, hospital2, how='inner', on=['month', 'day'])
```

הJOIN הסטנדרטי שנרצה לעשות הוא inner,

לעתות חיתוך על העמודות המשותפות, כמו

.month, day נקבל עמודות חדשות בשם

x_num_patients ו-y_num_patients.

לשימן לב! אם לשלב ביצוע merge של hospital עם forestfires נקבע plot_one_together=True, בזאת שהשדרה plot_one_together=True מילא את month ו-oct וmonth נתקה את שורות המתאימות מ-ff_fill.

-

-

-

-

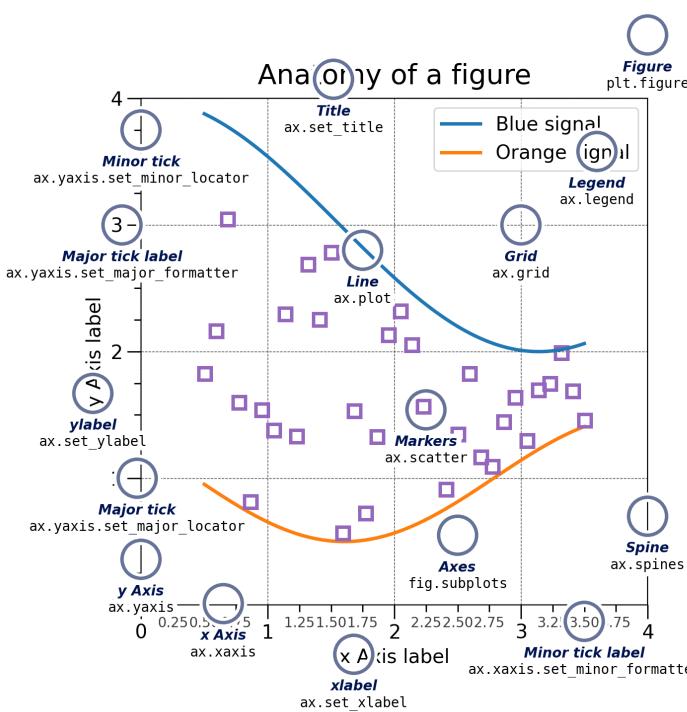
-

-

-

-

-

:matplotlib

המודול העיקרי בספריה זו הוא plt.plot ואנחנו מיבאים אותו בתווך plt. האובייקטים שיש לנו:

הΚΚΒΕΣ עצמו הוא figure (מייצג מערכת ציריהם).

plot אחד ויחיד. הוא יכול להכיל אחד או יותר Axis.

לכל Axis (בנich ציר-x) יש גבולות, חולצה וכו'.

האובייקט של הגרף עצמו נקרא artist. יכול להיות line лишь.

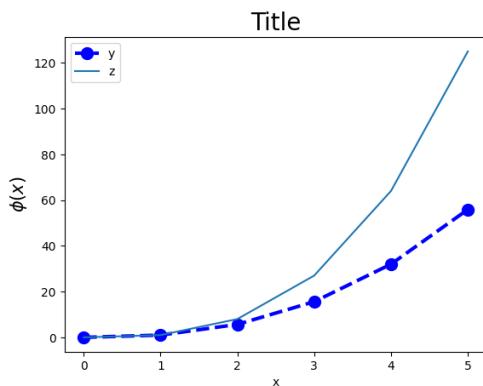
יש שני מושקים עיקריים לעבודה עם הספריה:

1. Implicit – נועד ליצור plot-ים on-the-fly בצוורה ישירה.

2. Explicit – מייצרים את האובייקטים אחד אחרי השני

(סיטיל OOP), ובונים את התרשים שכבה, שכבה.

```
plt.plot(x, y, 'o', markersize=10, color='blue', linewidth=3,
         linestyle='--', label = 'y')
plt.plot(x, z, label = 'z')
plt.xlabel('x')
plt.ylabel('$\phi(x)$', fontsize=15)
plt.title('Title', fontsize = 20)
plt.legend(loc='upper left')
plt.show()
```



הממשק implicit

- בגדייר array דח עברו א', נגידר את y ונשים את שמיים בפונקציה `plot()`. לאחר מכן נבצע `show()`. אפשר לקרוא שוב ל-`plot` כדי להוסיף עוד גרפים. אפשר להוסיף `label` לכל גרפ, לציין מה ה-`xlabel` וה-`ylabel` מקרה (legend).

תכונות מתקדמות יותר כוללות שינוי סוג ה-marker (נקודות במקום קו), ולקבוע את גודלו. כמו'ל עבור השרטוט: `marker`-`color`-`linestyle`-`linewidth`.

שרטוטים סטטיסטיים נפוצים כוללים `scatter`, `histogram`, `bar` ועוד.

אפשר ליצור מספר subplotים באמצעות `subplots`, באשר נקבע מספר תתי-שרטוטים שונים תחת אותו plot (בל אחד במערכת צירים משלה, הוא subplot).

הממשק explicit

הדרך המומלצת היא קרייה `l = plt.subplots(2,2)` לקבלת אובייקטי `Figure` ו-`Axes`.

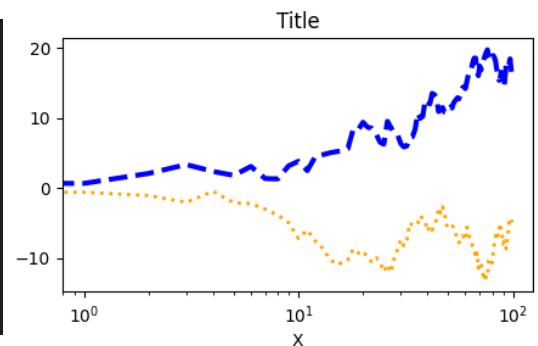
אם נבקש `subplots(2,2)` נקבל בפועל 4 axes שונים לעבד איתם, ובכל אחד מהם יוכל לשרטט גרף נפרד. יש פרמטר `figsize` שנ钦ן להעביר.

אפשר לתת כותרת לכל השרטוטיםividually. `fig.suptitle()`.

לכל ה-`axes` יש setters `set_title()`, `set_xlabel()` ו-`set_ylabel()`.

```
data1, data2 = np.random.randn(2, 100)

fig, ax = plt.subplots(figsize=(5, 2.7))
x = np.arange(len(data1))
ax.plot(x, np.cumsum(data1), color='blue', linewidth=3, linestyle='--')
ax.set_title('Title')
ax.set_xlabel('X')
ax.set_xscale('log')
l, = ax.plot(x, np.cumsum(data2), color='orange', linewidth=2)
l.set_linestyle(':')
plt.show()
```



ספירת seaborn

יש שתי דרכים להכניס נתונים פנימה: המעדפת היא הפרמטר `data`, תוך העברת שמות העמודות בפרמטרים `y`, `x`. אפשר גם להשתמש ישרות בהעبرا של `series` לתוך `u`.

וככל שהרטט `scatterplot()` או `FacetGrid` נוכל לבקש לא>Create את קו הרגרסיה באמצעות `fit_reg=False`. ביוון שהוא מת ממושך היבט עם `matplotlib`, אפשר לשחק עם `kind` אחריו `dots` ולהגדיר דברים נוספים (לעתך את ה-`title` למשל).

היסטוגרמה – באמצעות `distplot(kind="hist")`, או עם הפונקציה `hist` (histplot).

יש גם `violinplot`, `swarmplot`, `boxplot`, `pairplot` ועוד.

הסבר של ggplot: נרצה לדעת כיצד Attack משתנה לפי Stage. באמצעות `FacetGrid` קיבל את זה ישרות.

תרשים `pairplot` מציג איך כל משתנה מתנהג עם כל משתנה אחר, ומספר דרך טובה להתרשםות כלילית מה данא.

אפשר גם לראות heatmap: אחרי שביבצע `drop` של עמודות שלא מעניינות אותנו, נקרא `corr()` על ה-`df`. את זה נספק `heatmap()`.



רענון הסתברות

הסתברות בדידה

חשיבותה הסתברותית היא אבן יסוד בעובודה עם נתונים. גם בשנубוד עם מודלים שנראים כמו אלגוריתם טכני, שאין אחריו הנחות כלשהן על התפלגות, נראה שחשיבותה הסתברותית עוזרת לנו להבין הרבה יותר טוב מה האלגוריתם עובד, ואיך אפשר לשפר אותו.

מושגים בסיסיים:

- מרחב המדגם (sample space): קבוצה סופית או בת-מניה של כל המאורעות שיכולים לקרות. מסומן ב- Ω .
- פונקציית ההסתברות/התפלגות (probability distribution): מוגדרת לכל מאורע במרחב המדגם מספר שהוא $\sum_{\omega \in \Omega} F(\omega)$.
- הסתברות $[0,1] \rightarrow F: \Omega$, כך שסכום ההסתברויות הוא אחת: $1 = \sum_{\omega \in \Omega} F(\omega)$.
- לכל תת-קבוצה של מאורעות $\Omega \subseteq A$, ההסתברות שליה היא סכום המאורעות: $F(A) = \sum_{\omega \in A} F(\omega)$.
- משתנה מקרי (RV, random variable): פונקציה ממרחב המדגם לשער הממשי, $F \sim X$, המ"מ מתפלג F , אם $P[X \in A] = F(A)$.

נדגים על ניסוי של 2 הטלות מטבע: $\{\{H, H\}, \{H, T\}, \{T, H\}, \{T, T\}\} = \Omega$. אם המטבע הוגן (מרחב מדגם סימטרי), פונקציית ההסתברות היא רבע לכל אחד מהמאורעות הללו: $F(\omega) = \frac{1}{4} \in \Omega$.

$F(A) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$, היא מרכיבת משני המאורעות הבאים: $\{T, H\}, \{H, T\}$. ההסתברות היא $\frac{1}{2}$. נגידור מ"מ X להיות מספר הפעמים שהתקבל H – הוא מקבל את הערכים 0, 1, 2 בסיכוי רבע, חצי, ורבע בהתאם.

תוחלת (expectation): עבור מ"מ ממשי $\mathbb{E}[X]$ אם $F \sim X$ אז התוחלת של X היא ממוצע משוקלל לפי F :

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} \omega \cdot F(\omega) = 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{2} + 2 \cdot \frac{1}{4} = 1$$

תכונות חשובות של התוחלת:

- אדייטיביות: $\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$
- לינאריות: $\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$
- אינווריאנטיות לא מובטחת לכל פונקציה f : $\mathbb{E}\left[\frac{1}{x}\right] \neq \frac{1}{\mathbb{E}[x]}$

שונות (variance): אם תוחלת מוגדת מיקום (מרכז המסה של המשתנה המקרי), שונות מוגדת פיזור של המשתנה המקרי סביב התוחלת שלו. בהגדרת השונות אנחנו מוגדים את התוחלת של הסטייה הריבועית של X מהתוחלת שלו, וגם השונות היא ממוצע משוקלל של סטיות ריבועיות:

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \sum_{\omega \in \Omega} (\omega - \mathbb{E}[X])^2 \cdot F(\omega)$$

סטיית תקן (standard deviation): מחזירה אותנו לסקלה המקורית של X על ידי לקיחת שורש:

$$SD(X) = \sqrt{Var(X)}$$

תכונות חשובות של השונות:

- אם נפתח את הריבוע של הביטוי ונשתמש באדייטיביות התוחלת, נוכל לבטא: $Var(X) = \mathbb{E}[X^2] - [\mathbb{E}[X]]^2$
- השונות אינה אדייטיבית, אלא אם כן זוג משתנים הם ב"ת": $Var(X + Y) \neq Var(X) + Var(Y)$
- סוג של לינאריות: $Var(aX + b) = a^2 Var(X)$

התפלגות בדידות:

התפלגות ברוכול: מדברים שני תוצאות אפשריות – הצלחה/כשלון, נסמן $\{0, 1\} = \Omega$. נסמן (p) באשר $X \sim Ber(p)$ הוא מטבע ב-1. לדוגמה: הטלת מטבע באשר $X \sim Ber(0.5)$.

- פונקציית צפיפות (PDF): $P[X = 1] = p$
- תוחלת: $\mathbb{E}[X] = p \cdot 1 + (1 - p) \cdot 0 = p$
- שונות: $Var(X) = \mathbb{E}[X^2] = [\mathbb{E}[X]]^2 = p - p^2 = p(1 - p)$
- אנחנו רואים כי התוחלת של X היא גם p .

התפלגות בינימית: אם יש לנו אוסף של n משתני ברנולי ב"ת עם התפלגות זהה p , אז הסכום שלהם מתפלג בינימית. נסמן $(\mathbf{p}, \mathbf{n}) \sim Bin(n, p) \sim Y = \sum_{i=1}^n X_i \sim Ber(p)$ כאשר X_1, \dots, X_n משתנה זה סופר את מספר ההצלחות שנקל מתחזק n ניסיונות דיים. הערכים האפשריים הם $\{0, 1, \dots, n\}$. לדוגמה: $Y \sim Bin(10, 0.5)$.

- פונקציית צפיפות (PDF): $P[Y = k] = \binom{n}{k} p^k (1-p)^{n-k}$.
- תוחלת: בשתחלות של סכום משתני הברנולי שהוא סכום התוחלות: $E[Y] = E[\sum X_i] = \sum E[X_i] = np$.
- שונות: המשתנים הם ב"ת ולכן זה סכום השונות: $Var(Y) = np(1-p)$.

התפלגות פואסונית: מרחב המדגם הוא אינסופי וכלול את כל השלמים האו-שליליים: $\{0, 1, 2, \dots\}$. נסמן $(X \sim Pois(\lambda))$

$$\begin{aligned} P[X = k] &= e^{-\lambda} \cdot \frac{\lambda^k}{k!} \\ E[X] &= \sum_{k=0}^{\infty} k \cdot e^{-\lambda} \cdot \frac{\lambda^k}{k!} = \sum_{k=1}^{\infty} k \cdot e^{-\lambda} \cdot \frac{\lambda^k}{k!} = \lambda e^{-\lambda} \left[\sum_{k=1}^{\infty} \frac{\lambda^{k-1}}{(k-1)!} \right] \\ &= e^{\lambda} \\ Var(X) &= E[X^2] - [E[X]]^2 = \lambda^2 + \lambda - \lambda^2 = \lambda \end{aligned}$$

התפלגות זו חשובה כיון שהיא מתעדת בקרוב טוב לתהליכי ספירה שקרו ביוםום ובקשר: מספר הלקחות שmag'uisים לtower כלשהו, מספר החלקיים הרדיואקטיביים הנפלטים לאורך זמן, או מספר המוטציות שקרו בגנטום של יצור מסוים לאורך הרבה דורות.

- המנגנון שעומד מאחורי כל התהליכים הללו הוא **תכונת חוסר הזיכרון** – כל עוד זמן ההמתנה למאורע הבא T_t בלתי תלוי.
- בזמן שעבר מהקודם T_{t-1} , מספר המאורעות בפרק הזמן הזה מתפלג פואסוני.
- אידיביות של משתנים ב"ת – **הסכום שלהם מתפלג גם פואסוני** עם סכום הקצבים: $(X + Y \sim Pois(\lambda_1 + \lambda_2))$.

התפלגות אמפירית

התפלגות אמפירית: פונקציית ההתפלגות האמפירית חשובה לנו כל כך, כי היא הדרך שלנו לעבור נתונים (מספרים) להסתברות. יש לנו מודגם בגודל n שבו אנו מסמנים כל תצפית C_i , לא ראיינו ערכים אחרים – אנחנו מניחים שהוא שראינו הם כל מה שיכול לקרות. אם און חזות במדגם, אז פונקציית ההסתברות הוא $F(x_i) = \frac{|\{j: x_j = x_i\}|}{n}$

- התוחלת תהיה ממוצע התצפויות חלק i .
- המשמעות של $F(\omega_1) = F(\omega_2) \in \omega_1, \omega_2$ היא שהן מופיעות אותן במספר פעמים במדגם.

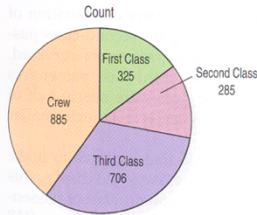
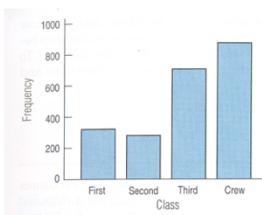
דוגמא – **נוסעים הטיטanic**: כל שורה בטבלה היא נסוע שיש לגבי 4 נתונים/משתנים: האם שוד, גיל, מין, מחלוקת בה נסע. נתמך בשני משתנים Y, X כאשר הערכים האפשריים הם x_K, \dots, x_1 (K – המחלוקת, x_1, \dots, x_K – האם שוד, $L = 2$ – גיל, y_1, \dots, y_L – מין). נראה את ההתפלגות האמפיריות ומהן נשאל שאלות מעניינות.

התפלגות שלות (marginal): במקרה ש- X היא למשל ההתפלגות של נתון המחלוקת שבה נמצא הנוסע:

Class	Count
First	325
Second	285
Third	706
Crew	885

Class	%
First	14.766
Second	12.949
Third	32.076
Crew	40.209

התפלגות משותפת (joint): מה ההסתברות שבו-זמןית מתקבלים ערכים מסוימים עבור Y, X , בך נקבל כי מרחב המדגם הוא מעין מכפלה של מרחבי המדגמים של Y, X :



- מתקיים $\forall k. P[X = x_k] \geq 0$
 - בנוסף $P[X = x_1] + \dots + P[X = x_K] = 1$
- התפלגות משותפת (joint):** מה ההסתברות שבו-זמןית מתקבלים ערכים מסוימים עבור Y, X , בך נקבל כי מרחב המדגם הוא מעין מכפלה של מרחבי המדגמים של Y, X :
- מתקיים $\forall k, l. P[X = x_k, Y = y_l] \geq 0$
 - בנוסף $\sum_{k=1, \dots, K, l=1, \dots, L} P[X = x_k, Y = y_l] = 1$
 - אם נקבע את $y_l = Y$ ונסכם את כל ההסתברויות על X נקבל את **התפלגות השולית הכללית** $P[Y = y_l]$:
 - $P[Y = y_l] = P[X = x_1, Y = y_l] + \dots + P[X = x_K, Y = y_l]$
 - $P[X = x_k, Y = y_l] = P[Y = y_l | X = x_k] P[X = x_k]$
 - למשל, ההסתברות שנסוע מצוות הספינה שוד + נסוע מצוות הספינה טבע = ההסתברות למצוא בכל נסוע מצוות הספינה.



Survival

	Class				
	First	Second	Third	Crew	Total
Alive	202	118	178	212	710
Dead	123	167	528	673	1491
Total	325	285	706	885	2201

איך ראוי להציג התפלגות משותפת של שני משתנים בדים עם מעט נתונים? **contingency table**.

- בשולטים נמצאים הסכומים שמייצגים את ההתפלגות השולית.
- כדי להציג ההתפלגות המשותפת צריך לחלק את המספרים בפניהם הטבלה, בסך המדגם של הנוסעים (2201). לדוגמה ההסתברות שנושע מצוחה הספינה שוד: $\frac{212}{2201} = 9.6\%$.

התפלגות מותנית (conditional): מקביעים משתנה אחד X להיות בקבוצת ערכים מסוימים, ושולאים בהינתן ש- X נמצא בקבוצת הערכים הזאת, מהי ההתפלגות של Y : $P[Y|X]$.
הנוסחה היא $P(Y|X) = \frac{P(Y \cap X)}{P(X)}$. זו פונקציית הסתברות רגילה:

- מתקיים $0 \leq P[Y = y_k | X = x_l] \leq 1$.
- בנוסף $1 = P[Y = y_1 | X = x_k] + P[Y = y_2 | X = x_k] + \dots + P[Y = y_L | X = x_k]$.

במקרה של הטיטניק, נשאל מה הסיכוי שנושע שוד או טבע, בהינתן שהוא במחלקה הראשונה/שנייה/שלישית/צוות הספינה. ההתפלגות של מצב הנושע לאחר הטבעה בהינתן המחלקה שלו, שונה לחולויין. פה טמון העניין האמתי בננתוני הטיטניק.

שתי נוסחאות מפתח:

$$1. \text{ נסחת ביחס: } \text{מקשרת בין ההסתברות המותנית } P[Y|X] \text{ לבין } P[X|Y]: \\ P[Y = y|X = x] = \frac{P[X = x, Y = y]}{P[X = x]} = \frac{P[X = x|Y = y] \cdot P[Y = y]}{P[X = x]}$$

$$2. \text{ נסחת ההסתברות השלמה: } \text{סובכת על פני כל האפשרויות השונות להנתנות:} \\ P[Y = y] = P[Y = y|X = x_1] \cdot P[X = x_1] + P[Y = y|X = x_2] \cdot P[X = x_2] + \dots + P[Y = y|X = x_k] \cdot P[X = x_k]$$

		Class				
		First	Second	Third	Crew	Total
Alive	Count	202	118	178	212	710
	% of Row	28.5%	16.6%	25.1%	29.9%	100%
	% of Column	62.2%	41.4%	25.2%	24.0%	32.3%
	% of Table	9.18%	5.36%	8.09%	9.63%	32.3%
Dead	Count	123	167	528	673	1491
	% of Row	8.25%	11.2%	35.4%	45.1%	100%
	% of Column	37.8%	58.6%	74.8%	76.0%	67.7%
	% of Table	5.59%	7.59%	24.0%	30.6%	67.7%
Total	Count	325	285	706	885	2201
	% of Row	14.8%	12.9%	32.1%	40.2%	100%
	% of Column	100%	100%	100%	100%	100%
	% of Table	14.8%	12.9%	32.1%	40.2%	100%

הציג שמסכמת את כל ההתפלגיות:

- מהי ההתפלגות השולית של X (מחלקה)?
בשורות ה-Total האחרונה. עבור $P[X = First]$ נראה שמדובר ב-14.8%.
- מהי ההתפלגות המשותפת של Y, X ?
בשורות הטבלה $P[X = First, Y = Alive]$ עבור $P[X = First, Y = Alive] = \frac{202}{2201} = 9.18\%$.
- מהי ההתפלגות המותנית $X|Y$?
בשורות הטבלה $P[Y = Alive|X = First]$ עבור $P[Y = Alive|X = First] = \frac{202}{325} = 62.2\%$. הוא שומרה את הבדיל המעניין. נראה שגם נסחת ביחס מתקיימת:
$$0.622 = \frac{0.285 \cdot 0.323}{0.148}$$

תלות בין משתנים

משתנים בלתי תלויים: משתנים בלתי תלויים אם ידוע על משתנה אחד לא מוסיף לידע על ההתפלגות של המשתנה האחר ולהיפך. למשל בדוגמה של הטיטניק, המחלקה והשרידות של נושע תלויות! נושע במחלקה הראשונה בעל סיכוי הישרדות גבוהים יותר מאשר במחלקה השלישית. אם אנחנו ידעים שנושע שוד – הסבירות שהוא הגיע מהמחלקה הראשונה גדולה פי כמה מאשר המצביע בו אנחנו ידעים שהנושע טבע.

- באופן פורמלי נאמר כי משתנים ב"ת, אם ההתפלגות המותנית שווה להתפלגות השולית: $P[Y|X] = P[Y]$.
- באופן שקול: $P[X|Y] = P[X]$.
- עבור ההתפלגות המשותפת מתקיים: $P[X, Y] = P[X] \cdot P[Y]$.
- עבור תוחלת מתקיים: $E[X \cdot Y] = E[X] \cdot E[Y]$.



שונות משותפת: מודד להשתנות המשותפת של זוג משתנים X, Y – עד כמה הם משתנים באופן דומה. ההגדרה הפורמלית:

$$\text{Cov}(X, Y) = \mathbb{E}_{XY}[(X - \mathbb{E}[X]) \cdot (Y - \mathbb{E}[Y])] = \mathbb{E}[X \cdot Y] - \mathbb{E}[X] \cdot \mathbb{E}[Y]$$

נשים לב שההתחלות היא על ההתפלגות המשותפת. לא נכון לומר ש- Cov מודד אי-תלות, אלא הוא מודד מתאם, קשר.

תכונות חשובות:

- כאשר Y, X ב"ת מתקיים $0 = \text{Cov}(X, Y)$, אבל זה לא נכון בכיוון הפוך! למשל – משתנה X שמקבל את הערכים 1,1 – בהסתברות 0.5 כל אחד. $0 = \text{Cov}(X, X^2)$ אבל המשתנים תלויים,ידע על X קבוע בבדיקה מה יהיה X^2 ?!
- נאמר שהם **בלתי-מתואמים**.
- אם באשר X מקבל ערכים "גבויים" גם Y : $0 > \text{Cov}(X, Y)$ כי הם נתונים להשתנות בצורה מתואמת.
- אם באשר X מקבל ערכים "גבויים" Y נתה לקבל ערכים "ণומיים": $0 < \text{Cov}(X, Y)$
- עבור אותו המשתנה: $\text{Cov}(X, X) = \mathbb{E}[(X - \mathbb{E}[X]) \cdot (X - \mathbb{E}[X])] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \text{Var}(X)$
- "לינאריות": $\text{Cov}(aX + b, cY + d) = ac\text{Cov}(X, Y) + ad + bc\text{Cov}(Y, X) + bd$
- שונות סכום: $\text{Var}(X \pm Y) = \text{Var}(X) + \text{Var}(Y) \pm 2\text{Cov}(X, Y)$

קורלציה (מתאם): אם השונות המשותפת שווה ל-0, אנחנו יודעים שהמשתנים בלתי מתואמים. אם הוא חיובי או שלילי? קשהeparsh מה אומר הגודל שלו. מקובל לנרגמל באופן הבא:

$$\rho_{XY} = \text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\text{SD}(X) \cdot \text{SD}(Y)}$$

כאשר $1 \leq \text{Corr}(X, Y) \leq -1$. מותאם של +1 אומר שהמשתנים מתואמים זה עם זה באופן מושלם (X עולה, Y עולה). מותאם של -1 אומר שהמשתנים בן מתואמים אבל בכיוונים מנוגדים (X עולה, Y יורדת).

אם יש מודגם עם n זוגות (x_i, y_i) ממשיים, ניתן לחשב עליהם את המתאם האמפירי – מקדם המתאים של פירסום:

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

אפשר להראות שהיא שווה לנו כאן הוא המרחק הקוסינוסי בין שני הווקטורים x, y אחרי שמרכנו אותם (חישרנו מהם את הממוצע). אם הזווית ישירה – אין ביניהם שום תיאום ונקבל 0. אם הווקטורים נעים בדיק לאותו כיוון נקבל 1:

$$r_{xy} = \cos\theta = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

QUIZ 2

שאלה 1: ידוע ששכר עובדי האוניברסיטה מתפלג לפי התפלגות נורמלית עם תוחלת 10 אלף שקלים. מכאן בהכרח נובע שתוחלת $"x"$ חלקו השכר" היא 1 חלקו 10 אלף. לא נכון, למדנו כי אינטגרנטיות לא מובטחת לכל פונקציה f : $\frac{1}{x} \neq \frac{1}{\mathbb{E}[x]}$. הפונקציה $\frac{1}{x+y}$ אינה טרנספורמציה לינארית (למשל $(y+f)(x) = f(x) + \frac{1}{x+y} \neq \frac{1}{x} + \frac{1}{y} = f(x+y)$).

שאלה 2: מיקל ג'ordon זורק שתי קליעות עונשין. הוא קולע בהסתברות 0.8 אחד בכל ניסיון (כלומר אין תלות בין הניסיונות). התפלגות המתאימה למדל את מספר הניסיונות שיקלע הוא:

- **בינומית** – התפלגות המתאימה למידול מספר ההצלחות (קליעות) מתוך מספר קבוע של ניסיונות, כאשר לפחות בכל ניסיון יש הסתברות הצלחה קבועה והניסיונות בלתי תלויים, היא התפלגות **בינומית**.
- ברנולי – במה ניסיונות יש למיקל? (2)
- פואסון – מיקל יbole לקלוע גם אינספור ניסיונות?
- גיאומטרית – מה יקרה אם מיקל יקלע בניסיון הראשון?



שאלה 3: דנית ערכות סקר, לכל שאלת צריכה ציר לסטטן מספר בין 1 ל-5. היא חשושת מmagיבים ש"סתם" יملאו את הסקר, ובפרט magיבים שימלאו את אותו מספר (למשל 1) לכל השאלות. איזה מהמדוברים הבאים **שישוחב לכל magיב יכול לעזור לה לגנות magיבים אדיישים באלה בצורה הפשואה והמידית ביותר?**

- **סטיית תקן** – כאשר magיב מלא את אותו מספר עבור כל השאלות, סטיית התקן של התשובות שלו תהיה אפס, משום שאין שונות בין התשובות. לעומת זאת, עבור magיבים שנוטנים תשובות מגוונות, סטיית התקן תהיה גדולה מאפס.
- **ממוצע** – התשובה נconaה אם יצא ממוצע 1.0 או 5.0. אבל נניח שיצא למגיב ממוצע 2.0. מה זה אומר?
- **מתאים** – מתאים בין מה למה? בין זוג שאלות? בין תשובות המגיב לממוצע האחרים?
- **התפלגות אמפירית** – התפלגות אמפירית לכל magיב יכול לסייע אבל יש תשובה הרבה יותר מידיית.

שאלה 4: נמצא ניצול מהטייטאניק בלבד ים! על פי הנתונים במצגת, סביר להניח (הסתברות יותר מחצי) שהוא מצוות הספינה או מחלוקתה שלישית. **נכוו**, ההסתברות שניצול יהיה מצוות הספינה היא 29.9%, ומחלוקתה שלישית 25.1% מה"ב נקבע יותר מ-50%.

תרגול 2 – הסתברות

ראשית, נשים לב שבממשק של `umpy` ו-`scipy` תמייד מבדילים בין PDF (פונקציית צפיפות) לבין CDF (הסתברות מצטברת). נציין כי אנחנו מבדילים בין משתנים בדיםיהם שלהם יש פונקציית הסתברות, בין משתנים רציפים שלהם יש פונקציית צפיפות, אשר לבשעצתה לא אומרת הסתברות.

התפלגות בינומית: סופרים מספר ההצלחות ב- n ניסויים ב"ת עם סיכוי קבוע p להצלחה.

דוגמא: לברון זורק 50 פעמים במשחק (כל זריקה שווה 2 נקודות), ומצילח 51% מהזמן. הזריקות והשחקים ב"ת.

א) מה ההסתברות שבמשחק מסויים הוא יקבל פחות מ-60 נקודות? X זה מספר ההצלחות שמתפלג $Bin(50, 0.51)$. נסמן $2X = Y$ מספר הנקודות. מדרשו $P[X \leq 29] = P[Y < 60] = P[X < 30]$ כי אין משמעות להגעה ל-29.999, אפשר לחשב בפייתון.

- בעזרת `umpy` נקבל 0.87.

```
shots = np.arange(30)
def prob_i(i, n=50, p=0.51):
    return math.comb(n, i) * (p ** i) * ((1-p) ** (n - i))
print(np.sum([prob_i(i) for i in shots]))
```

0.8712563051615556

- בעזרת `.scipy.stats`

```
import scipy.stats as stats
stats.binom.cdf(k=29, n=50, p=0.51)
```

- בעזרת סמלץ בוללה וחישוב ממוצע על פני 1000 סימולציות.

```
def lebron_game(throws=50, points=2):
    return np.sum([points for _ in range(throws) if np.random.rand() <= 0.51])
n_games = 1000
points_vec = np.array([lebron_game() for _ in range(n_games)])
np.mean(points_vec) < 60
```

0.863

ב) מהי התוחלת של מספר הנקודות למשחק? מהי סטיית התקן?

- מילינאריות התוחלת: $E[Y] = E[2X] = 2E[X] = 2 \cdot 50 \cdot 0.51 = 50$
- סטיית התקן: $SD(Y) = \sqrt{Var(Y)} = \sqrt{Var(2X)} = \sqrt{4Var(X)} = 2\sqrt{Var(X)} = 2\sqrt{50 \cdot 0.51 \cdot (1 - 0.51)} = 2\sqrt{12.5} = 5\sqrt{2}$

את שני החישובים האלה ניתן להשיג באמצעות `stats.binom` עם **mean** ו-**std**.

ג) אם לברון זורק מעל 150 נקודות ב-3 המשחקים הבאים הוא יקבל תואר GOAT. מהי ההסתברות שזה יקרה?

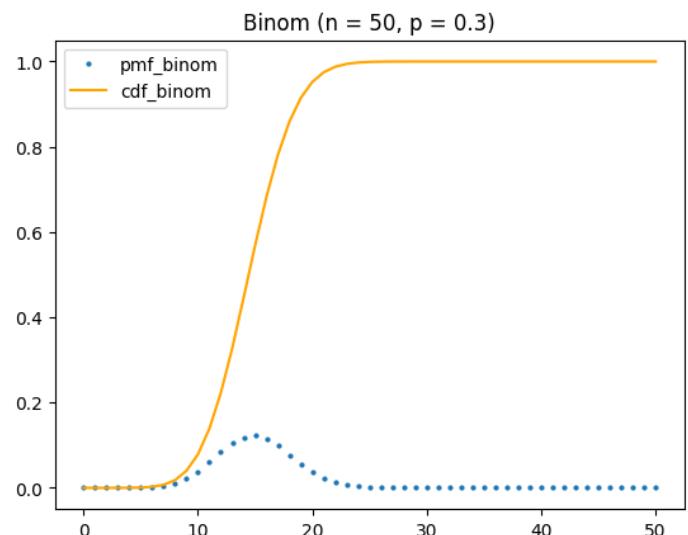
- כל אחד מהמשחקים (X_1, X_2, X_3) הוא $Bin(50, 0.51)$ ומגדירים $Z = X_1 + X_2 + X_3 \sim Bin(150, 0.51)$ ואנו רוצים את מספר הנקודות הסופי שהוא מוגדר $Y = 2Z$.
- נחשב $P[Y > 150] = 1 - P[Y \leq 150] = 1 - P[Z \leq 75] = \sum_{i=0}^{75} \binom{150}{i} 0.51^i (1 - 0.51)^{150-i}$.

לסיכום:

```

n = 50
p = 0.3
z = np.arange(0, n + 1) # note n is an option!
pmf_binom = stats.binom.pmf(z, n, p) # f(k) = P[X=k]
cdf_binom = stats.binom.cdf(z, n, p) # F(k) = P[X<=k]
plt.plot(z, pmf_binom, 'o', ms=2, label='pmf_binom')
plt.plot(z, cdf_binom, label='cdf_binom', color = 'orange')
plt.title(f'Binom (n = {n}, p = {p})')
plt.legend()
plt.show()

```



התפלגות פואסוני: כאשר אנחנו מבצעים מבחן על אירועים שקרים.

דוגמה: בMOTE של המילויים שטודנטים שולחים כל יום הוא תהליכי פואסוני. בממוצע, 2 ביום.

א) מה ההסתברות שביום שלישי קיבל לפחות מיל אחד?

- נסמן ב- X את מספר המילאים: $X \sim Pois(2)$

$$P[X \geq 1] = 1 - P[X = 0] = 1 - e^{-2} \cdot \frac{2^0}{0!} = 1 - e^{-2} - e^{-2} = 1 - 2e^{-2}$$

```

def one_day_email(lamb = 2):
    return stats.poisson.rvs(mu = lamb)
n_days = 10000
# easier: stats.poisson.rvs(lamb, size=n_days)
emails_vec = np.array([one_day_email() for _ in range(n_days)])
np.mean(emails_vec) >= 1

```

0.8725

ב) מה ההסתברות שבימים שלישי ושני קיבל בדיקות מיל אחד (אחד בכל יום), אם נתנו שביום ראשון שטודנטים שלחו 4?

- נזכור בתמונה החשובה של תהליכי פואסוני – קצב האירועים בין שני ייחידות זמן הוא בלתי תלוי. נסמן X_i כמספר המילאים ביום i עברו [7].

$$P[X_2 = 1 \cap X_3 = 1] = P[X_2 = 1] \cdot P[X_3 = 1] = (2e^{-2})^2 = 4e^{-4}$$

ג) מה ההסתברות שבימים שלישי ושני קיבל בסך הכל 2?

$$P[X_2 + X_3 = 2] = 8e^{-4}$$
 וכן נחשב $Pois(4)$

ד) מה התוחלת של מספר המילאים בשבוע בווד? מהי סטיית התקן?

- תוחלת הסכום היא סכום התוחלות ונקבל $.7 \cdot \mathbb{E}[X_i] = 14$.
- סטיית התקן: $SD(\sum X_i) = \sqrt{Var(\sum X_i)} = \sqrt{\sum Var(X_i)} = \sqrt{14}$. הם ב"ת וכן שונות הסכום היא סכום השונות.

ה) אולי לא מדובר ב-2 בממוצע. שבוע שעבר קיבלי ב-7 ימים [3, 2, 1, 4, 5, 2, 6] מילאים. מה יותר סביר, הממוצע הוא ?? או ?? או ??

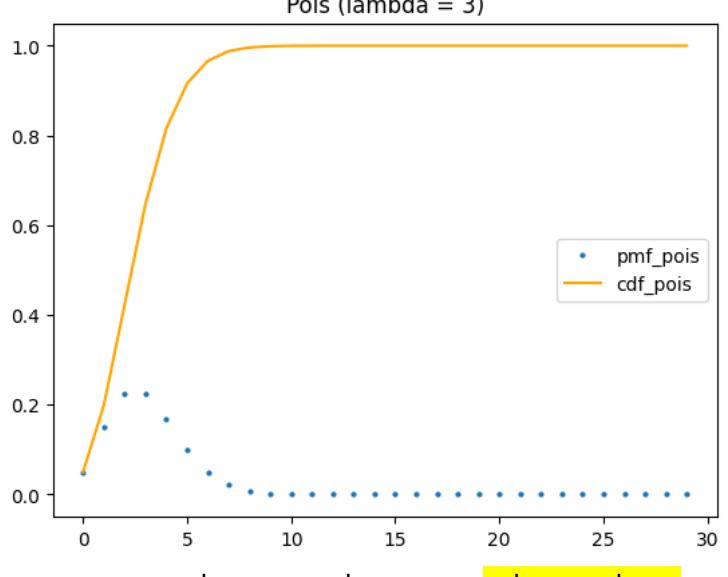
- נחשב את הסיכוי לקבל את התוצאה תחת $\lambda = 2$ ותחת $\lambda = 7$. נבצע מכפלה באמצעות `np.prod`.
- יותר סביר שהפרמטר הוא 4 מילאים ליום.

לסיכום:

```

lamb = 3 # why not "lambda"
z = np.arange(30) # note this could go on to infinity!
pmf_pois = stats.poisson.pmf(z, lamb) # f(k)
cdf_pois = stats.poisson.cdf(z, lamb) # F(k)
plt.plot(z, pmf_pois, 'o', ms=2, label='pmf_pois')
plt.plot(z, cdf_pois, label='cdf_pois', color = 'orange')
plt.title(f'Pois (lambda = {lamb})')
plt.legend()
plt.show()

```



התפלגות נורמלית: חסובה בשל משפט הגבול המركזי.

דוגמיה: בכל לילה ישנה הופעה, באשר מספר האנשים שבאים לקנות ברטיסים מהתפלג נורמלית עם תוחלת 10K וסטיית תקן 2K.

א) יש 12K מושבים באולם, מהי ההסתברות שביליה מסויים לא ישארו ברטיסים?

- ראשית נגדיר $N \sim X$. אנו רוצים לחשב את הסיכוי שמספר האנשים שהגיעו גדול מ-12K:

$$P[X > 12] = 1 - P[X < 12] = 1 - \Phi\left(\frac{12 - 10}{2}\right)$$

ב) אמם, אם פחות מ-7K אנשים באים לקנות ברטיסים, הזרמת משלמת קנס. אם אנחנו יודעים שאתמול האולם לא היה מלא, מהי ההסתברות שהזמרת שילמה קנס?

$$P[\text{fine}|\text{not full}] = P[X < 7|X < 12] = \frac{P[X < 7 \cap X < 12]}{P[X < 12]} = \frac{P[X < 7]}{P[X < 12]} = \frac{\Phi(-1.5)}{\Phi(1)}$$

ג) אם ידוע כי אתמול האולם לא היה מלא אבל הזרמת לא שילמה קנס – מהי ההסתברות שהוא יותר מ-10K 12K אנשים בקהל?

$$P[X > 10|X < 12, X < 7] = \frac{P[10 < X < 12]}{P[7 < X < 12]} = \frac{P[X < 12] - P[X < 10]}{P[X < 12] - P[X < 7]} = \frac{\Phi(1) - \Phi(0)}{\Phi(1) - \Phi(-1.5)}$$

ד) מהו האחוזון ה-90 של אנשים שבאים לקנות ברטיסים?

- האחוזון ה-90 של $X = \text{הערך שהסיכוי להיות מעליו הוא } 10\%$. נרצה למצוא את X שעבורו $\Phi\left(\frac{x-\mu}{\sigma}\right) = 0.9$.
- ניתן להשתמש ב-`stats.norm.ppf`.
- קיבלנו 12.5 כולם 12,500 אנשים.

ה) נניח כי אין הגבלת מקום באולם. ברטיס עולה 100 דולר, והוצאות לכל מופע הן 200K דולר. מהי תוחלת הרוח (במאות הברטיסים * מספר האנשים – ההוצאות) עבורהليلיה אחד, ומהי סטיית התקן?

נחשב: $100K \cdot X - 200K = R$ שוכן בערך של X הוא אלף איש.

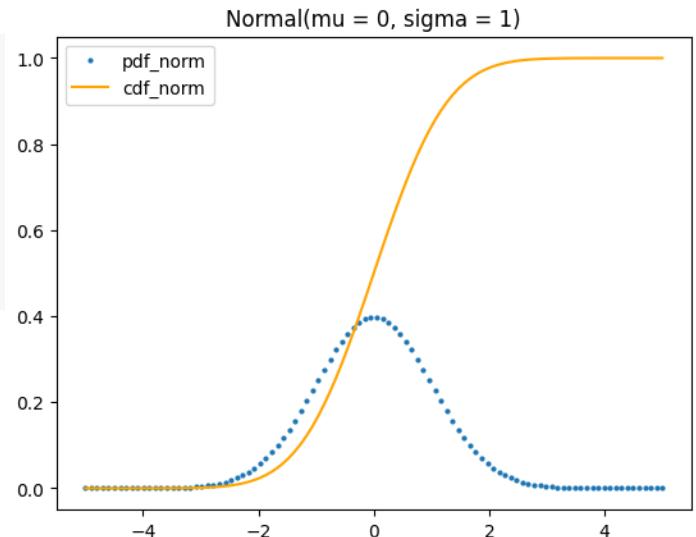
- תוחלת: $\mathbb{E}[R] = \mathbb{E}[100K \cdot X - 200K] = 100K \cdot \mathbb{E}[X] - 200K = 100K \cdot 10 - 200K = 800K$
- סטיית התקן: $SD(R) = \sqrt{V(100K \cdot X - 200K)} = \sqrt{V(100K \cdot X)} = 100K\sqrt{V(X)} = 100K \cdot 2 = 200K$
- בפועל קיבלנו כי $R \sim (800K, 200K^2)$

לסיכום:

```

mu = 0
sigma = 1
z = np.linspace(-5, 5, 100) # note can be from - to + infinity!
pdf_norm = stats.norm.pdf(z, mu, sigma) # f(x)
cdf_norm = stats.norm.cdf(z, mu, sigma) # F(x)
plt.plot(z, pdf_norm, 'o', ms=2, label='pdf_norm')
plt.plot(z, cdf_norm, label='cdf_norm', color = 'orange')
plt.title(f'Normal(mu = {mu}, sigma = {sigma})')
plt.legend()
plt.show()

```



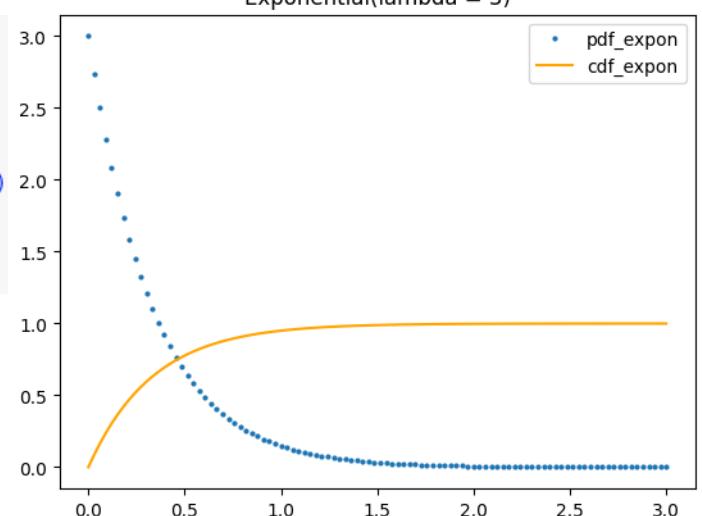
הערה חשובה: צפיפות אינה הסתברות!

התפלגות מעריכית: הזמן בין שני אירועים פאסוניים מתפלג מעריכית.

```

lamb = 3
z = np.linspace(0, 3, 100) # note can be infinity!
pdf_expon = stats.expon.pdf(z, scale = 1 / lamb) # f(x)
cdf_expon = stats.expon.cdf(z, scale = 1 / lamb) # F(x)
plt.plot(z, pdf_expon, 'o', ms=2, label='pdf_expon')
plt.plot(z, cdf_expon, label='cdf_expon', color = 'orange')
plt.title(f'Exponential(lambda = {lamb})')
plt.legend()
plt.show()

```



```

# Notice in scipy.stats.expon the scale parameter is the standard deviation,
# therefore for our notation you should input 1/lamb
print(f'lambda is: {lamb}')
print(f'E(X) is: {stats.expon.mean(scale = 1/lamb):.5f} as expected.')

```

הערה חשובה:

```

lambda is: 3
E(X) is: 0.33333 as expected.

```

סיכום כל סוג ההתפלגות:

התפלגות	סימון	CDF/F(k)	צפיפות: f(k)	תוחלת שונות
בינומית	$Bin(n, p)$ $p \in [0,1]$	$\sum_{i=0}^{ k } \binom{n}{i} p^i (1-p)^{n-i}$	$\binom{n}{k} p^k (1-p)^{n-k}$	$np(1-p)$
פואסונית	$Pois(\lambda)$ $\lambda > 0$	$e^{-\lambda} \sum_{i=0}^{ k } \frac{\lambda^i}{i!}$	$\frac{\lambda^k e^{-\lambda}}{k!}$	λ
נורמלית	$N(\mu, \sigma^2)$	$\Phi\left(\frac{x-\mu}{\sigma}\right)$	$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$	μ
מעריכית	$Exp(\lambda)$	$1 - e^{-\lambda x}$	$\lambda e^{-\lambda x}$	$\frac{1}{\lambda^2}$



חקר נתונים

איך מושגים נתונים? מה כדי לעשות דבר ראשון בשם קובץ נתונים? מה יכול להיות כל כך לא "נכוי" בהם צריך לנகוט אותו? וראשת נסובל על מה עקרונות בסיס נתונים, שיהו חומר הגלם עבורנו ב-DS.

איסוף וארגון של נתונים

פורמטים נפוצים נתונים:

1. **CSV (Comma Separated Values):** זה פורטmez מצוין לDATA בצורת טבלה. למשל, DATA שבKOHOOT תיאר את החוויות שלהם מתרופות שונות, וירוגו אותן. ניתן לקרוא בKOHOOT את הקובץ באמצעות pandas והפונקציה `read_csv()`.
2. **JSON (JavaScript Object Notation):** פורטmez נוסף שהו גמיש הרבה יותר, שבו מגעים רבים פעמים נתונים כתוצאה מקראית ל-API של שירותים באינטרנט. הוא מתאים למידע מוקן (nested), כמו MILION (`key-value`). במקרה של פונקציית `load()`, מטר מפקד אוכלוסין על אדם מסוים, המפתחות ה-JSON שוכנים על אותו אדם, והערכים יוכלים להיות מטיפוסים שונים (מחוזת, רשימה, אפילו MILION). ניתן לקרוא את DATA באמצעות `json.load()`, שקורא את הנתונים ישירות לערך `dict` של פירטו.
3. **Plain Text:** אפשר לקבל קובץ נתונים פשוט, `raw`, ואפשר לקרוא אותו עם `readlines`, כל שורה טקסט היא מחוזת.
4. **HTML:** אפשר להציג נתונים כמעט מכל אתר אינטרנט, והמידע בפורטmez HTML בני מעצים של תוכנות שמתפרשים לתתי-ענפים. אפשר להשתמש בספרייה BeautifulSoup ולקראת המידע לערך אובייקט `BeautifulSoup`. ניתן לקרוא ל-`prettify()`.

```

1 id,drugName,condition,review,rating
2 206461,Valsartan,Left Ventricular Dysfunction,"""It has n
3 95260,Guanfacine,ADHD,"""My son is halfway through his fo
4 We have tried many different medications and so far this
5 92703,Lybrele,Birth Control,"""I used to take another oral
6 The positive side is that I didn't have any other si
7 138000,Ortho Evra,Birth Control,"""This is my first time
8 35696,Buprenorphine / naloxone,Opiate Dependence,"""Subox
9

```

```

1   {
2     "firstname": "John",
3     "lastname": "Smith",
4     "age": "27",
5     "address": {
6       "streetAddress": "21 2nd Street",
7       "city": "New York",
8       "state": "NY",
9       "postalCode": "10021-3100"
10      },
11      "children": [],
12      "spouse": null
13    }

```

איסוף נתונים:

איך מושגים אלינו נתונים? בעבר נתונים הגיעו כ-MSH, מישוה כתוב אותם שורה אחריו שורה. היום בעידן הביג DATA, רוב הנתונים נאספים בצורה אוטומטית. נראה כמה דוגמאות:

- גם היום יש שפע דוגמאות של הוצאות נתונים ידנית, כמו סימון הגובה של ילד על הקיר.
- בrama הלאומית, נתונים תמיד הגיעו מממשלות שפיטו תיעדו את הקורה בתוך המדינה לצורך ניתוח כלכלי. למשל, כמו צריכת האלכוהול השנתית של אזרח אמריקאי.

כיום, כל הנתונים נאספים בצורה אוטומטית. כל פעם שנכנים לארון אינטרנט, אפילו אם רק מסתכלים על מחירים של מוצרים – אנחנו מייצרים DATA, עצם פתיחת הדפסן מייצרת שורה במאגר נתונים (מי אנחנו, גיל, מקום, מתי פתחנו את הדפסן), ואפילו באתרים מסוימים יכולה להיות שורה בסיס נתונים (מתי, מה, מה הפעולה הבאה).

ניתן לאסוף נתונים מהאינטרנט באמצעות **web scraping**:

1. **Public APIs:** דוגמה ל-API זה הוא Google Trends, המאפשרת לקבל את דפוס החיפוש אחר מונח כלשהו לאורך תקופה, ואפילו להשוו למונחים אחרים. אפשר ליצא את הנתונים בתור CSV.

```

1 albums = dict()
2 id = 0
3 albums[id] = dict()
4 tables = soup.find_all('table')
5 for table in tables:
6     caption = table.find('caption')
7     if caption is not None:
8         header = caption.get_text()
9         if re.match(re.compile('^List of(.+?)albums'), header):
10             rows = table.find_all('tr')
11             for row in rows:
12                 title_col = row.find('th')
13                 if title_col is not None and 'scope' in title_col.attrs and\
14                     title_col.attrs['scope'] == 'row':
15                     title_cell = title_col.find('a')
16                     if title_cell is not None and title_cell.attrs is not None and\
17                         'title' in title_cell.attrs:
18                         albums[id]['name'] = title_cell.attrs['title']
19                         release_col = row.find('td')
20                         release_date, release_label = get_release_details(release_c

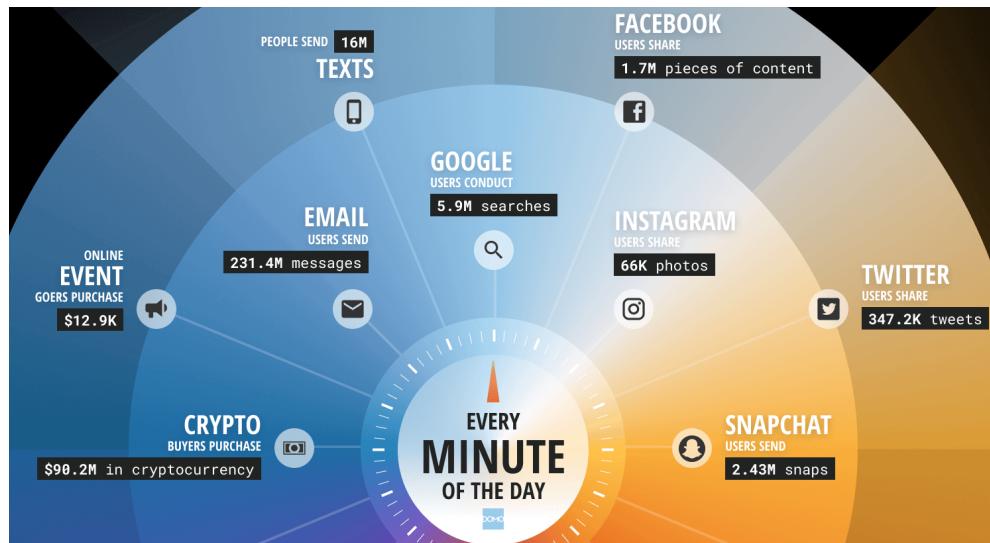
```

2. **Beautiful Soup:** פועלה אקטיבית מול דף שהוא אונליין, שאין לו מדיניות נוקשה נגד סקרים טרורדים DATA. נסובל למשל על עמוד הויקיפדיה של תקליטי ה-beatles. יש בעמוד טבלה, וכך לעוד איתה בצורה נוחה לקרוא את התוכן של העמוד לערך אובייקט `BeautifulSoup`, נועור על DATA שקיבלו ונחפש את הטבלה עם הכוורת המבוקשת. ניתן לעטוף MILION זהה DataFrame, ומשם לעבד עם טבלה pandas. כפי שאנחנו מכירים עם pandas.

מספר הערות על המונחים Big data ו-Small data. יש שיטענו שדאטה הוא קטן או גדול לפי **הגודל שהוא צריך פיזית על הדיסק**. למשל דאטא קטן נכנס בקובץ אקסל, ודאטא גדול לא נכנס בזיכרון של מחשב בלבד. הגדרות אחרות חושבות על האופן בו אנחנו מגדירים את הדאטא: דאטא קטן הוא דאטא שאנשים יכולים להבין, וגדול הוא在乎 שיטות מסורתיות לעיבוד נתונים כבר לא יכולות לפעול עליו. אפשר לשלב את ההגדרות:

- נניח שיש לנו דאטא שנכנס באקסל – מיליון שורות ו16 אלף עמודות. ננסה להזין מטריצה זאת בזיכרון בפייטון, לכפול אותה, להפוך את התוצאה... לא בטוח שנצליח. למחרת שהנתונים נכנסים באקסל, הם לא גדולים מדי?
- לחילופין לפי מקור מסוים פייסבוק מייצרים כמה petabytes של דאטא כל יום (מיליון GB). עם זאת, אם נציג על מדענית נתונים בפייסבוק, ספק אם היא משתמשת ביוםיום בהםם מעל מה שהמחשב שלהם יכול להכיל, רוב הזמן נפעיל על תתקבוצה קטנה של נתונים.

אפשר להסביר בפשטות כי הנתונים שהאינטרנט מייצר הם נתונים ענק.



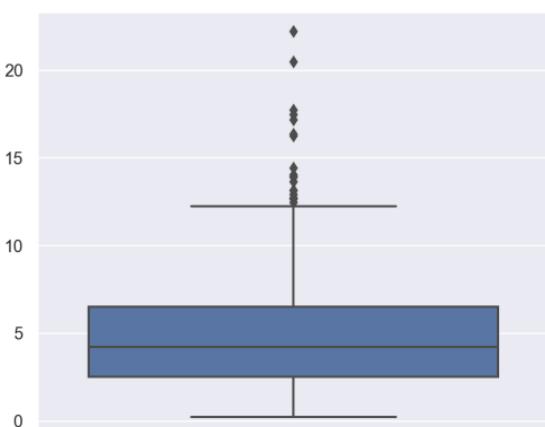
חקר נתונים - תרשימים

אחד הדברים הראשונים שנרצה לעשות עם נתונים (קטנים או גדולים), הוא למצות מהם כמה תרשימים. תרשימים ייחודיים אחד הוא בעל פוטנציאלי להיות הרובה דברים על הנתונים שלנו, אולי גם בעיות.

Boxplot: השתמש בספריית `seaborn` שמתממשקת היטב עם `pandas`. נציג תרשימים קופסה של משתנה אקראי X , ונרצה שה קופסה תהיה עומדת (בניגוד לשוכבת), וכן נפרט $X = u$. תרשימים זה מרכיב מ-5 קווים בלבד:

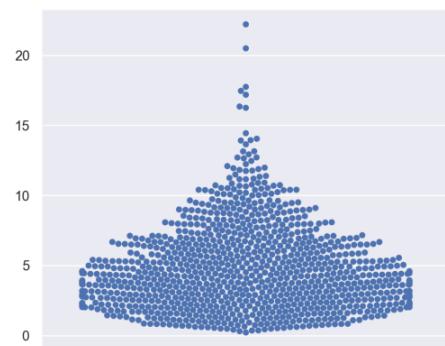
- הקו במרכז הקופסה הוא החיצון של הנתונים – הערך שכני מההתפלגות מעליון, וחצי מתחתיו.

הגבול התיכון והגבול העליון של הקופסה, הם הרבעון התיכון (אחוודן 25) והרביעון העליון (אחוודן 75) בהתאם. רוחב הקופסה הוא אחוודן הרבעון העליון פחות הרבעון התיכון, ערך זה נקרא **inter-quartile range (IQR)**.



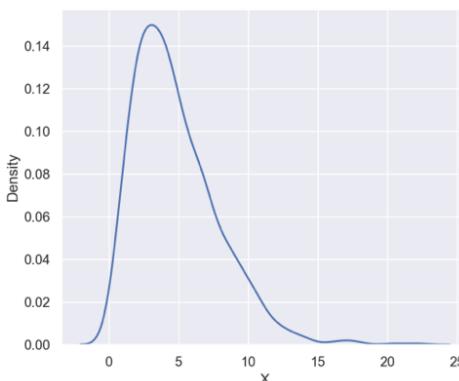
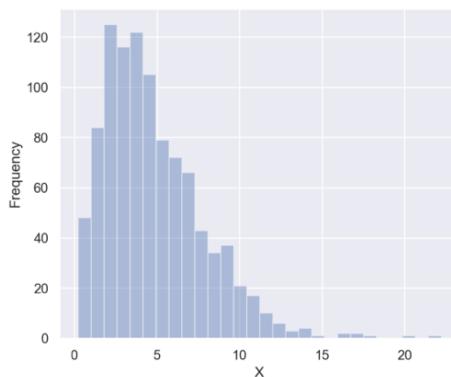
כדי לקבל את השפם העליון, נוסיף לגבול העליון של הקופסה את $1.5 \cdot IQR$

אבל לא נמתח עד שם את השפם, אלא עד הערך שכני קרוב אליו מלמטה. עבור השפם התיכון נעשה אותו דבר בלבד, נמתח את השפם בערך הכי קרוב אליו מלמטה, לרבעון תיכון פחות $1.5 \cdot IQR$.



נסמן את התציפות שנפללו מעבר לגבול, מעבר לשני השפמים – אלא תציפות ייחסית חריגות.

Swarmplot: מגע מהAMILה באנגלית לנחיל (כמו נחיל של דוברים). כל תציפות מיוצגת על ידי נקודה, ואפשר לראות את צורת ההתפלגות במדויק יותר. הוא מתאים יותר לדאטא קטן מאשר נשרטט עליו `boxplot` התמונה תהיה אולי קצת מטעה.



Histogram: תרשימים נפוץ לראות התפלגות. בrix עם הפקודה `sns.distplot` (הגנרטיב יותר). ההיסטוגרמה מחלקת את הטווח הרלונטי למשתנה, למקטעים שווים שנקראים `bins`. היא סופרת כמה ערכים נמצאים בכל מזב. בוצרה כזו קל לראות אם ההתפלגות סימטרית או לא, אולי יש לה זנב ימינה לכיוון ערכים גדולים, או זנב שמאלה לכיוון ערכים קטנים.

-
-
Density: `hist=False` בקריאה `distplot`. זה ייצור לנו תרשימים צפיפות, שנוצר עם מה שקרי kernel density estimation (KDE), קונבולוציה על הנתונים. נגידו פונקציית גרעין $\mathbb{R}^+ \rightarrow \mathbb{R}$: w שנדרש ממנה:

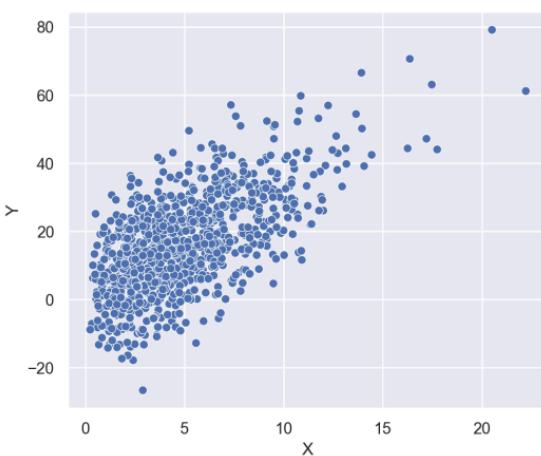
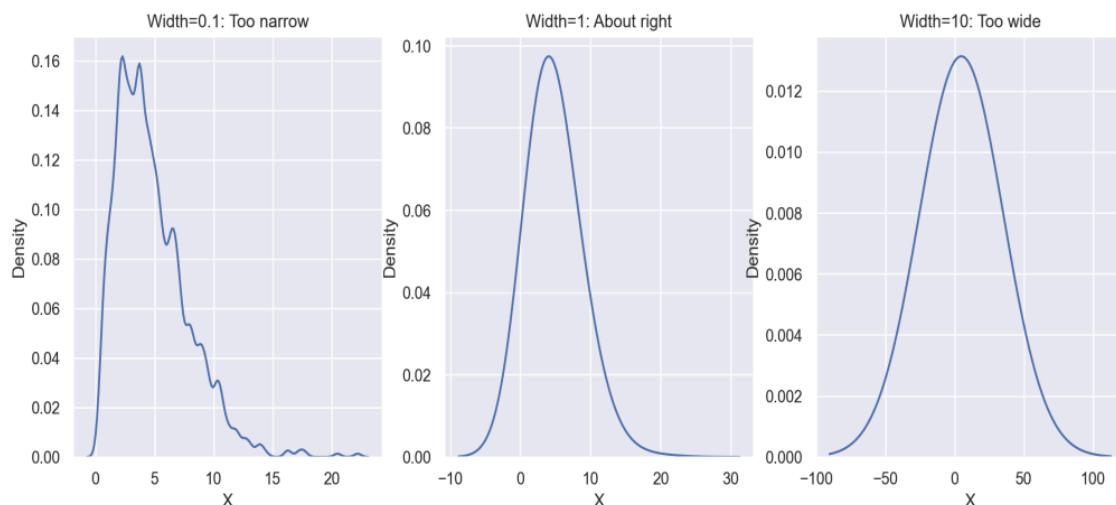
1. סימטרית (או-שלילית): $(x-w)w = w(x)$.
2. האינטגרל עליה הוא 1: $\int_{\mathbb{R}} w(x)dx = 1$

בפועל אנחנו מוחפשים חלון קטן. נציג אותו לאורך טווח הערכים שהמשתנה X מקבל ונפעיל אותו על המשתנה X – זו תהיה הצפיפות:

$$J(x) = \frac{1}{n} \cdot \sum_{i=1}^n w(x_i - x)$$

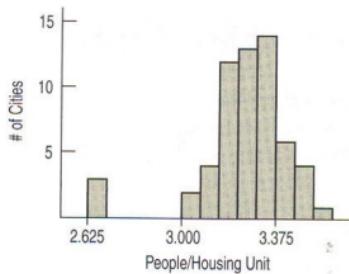
השטח תחת תרשימים הצפיפות שמתקיים: $1 = \int_{\mathbb{R}} J(x)dx$. למעשה קיבלנו מעין "היסטוגרמה מוחלקת".

בחירה bandwidth (גודל החלון): אם נבחר חלון רחב מדי, התוצאה עשויה להיות חלקה מאוד אבל לא לייצג את המבנה האמתי של ההתפלגות (התקבלת ההתפלגות פעמן מושלמת וסימטרית, שלא מייצגת את הנתונים). אם נבחר חלון קטן מדי, תרשימים הצפיפות יהיה ספציפי מדי ולא מספיק חלק (הרבה קפיצות).

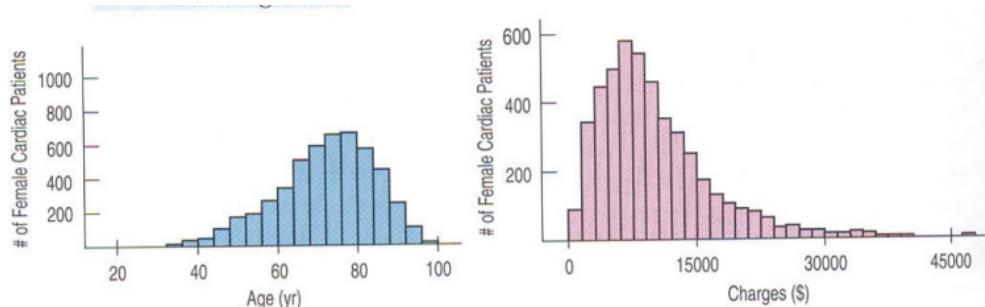


Scatterplot: תרשימים חשובים בין זוג משתנים רציפים הוא תרשימים פיזור. נדמה משתנה Y על סמך משתנה X , נוסף לו קצת רעש, ונשתמש ב-`scatterplot` ליצור תרשימים פיזור ביניהם. כל זוג ערכים של המשתנים מיוצג על ידי נקודה, ומאפשר לראות את הקשר בין המשתנים, ובפרט מה שיונין אותם – האם יש לו דפוס מסוים? כמו **עליה או ירידה**.

מתרשימים פשוטים אפשר ללמוד לא מעט:



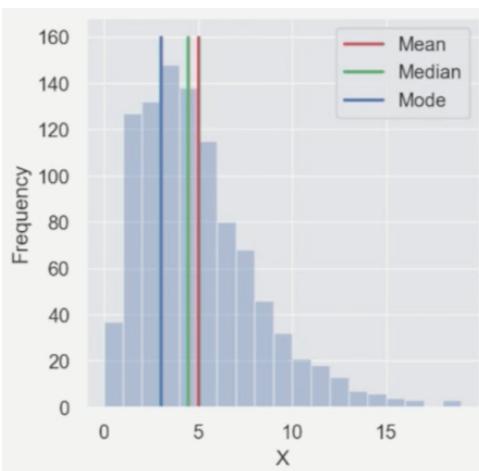
- האם יש תצפויות חריגות, כמו בהיסטוגרמה הבאה: יש בבירור כמה תצפויות חריגות של כמה ערים שבהן המספר הוא קטן במיוחד.
- מענין אותנו לדעת מה צורת ההתפלגות ואיזה זנב יש לה במידה שהיא לא סימטרית. באנו רואים זנב שמאלי בהיסטוגרמה שמתארת גיל של נשים שלקן בהתפק לב, כי התופעהנדירה יותר בקרב צעירות. לעומת זאת, במקרים מסוימים בסוף (התפלגות החזאות של נשים שטופלו במחלקות לב), נראה זנב ימני, חריגות של ערכים גבוהים. ברוב המקרים בסוף זה דבר שכיח רק לגדול, וכן במקרים פיננסיים נראה הרבה פעמים זנב ימני.



חקר נתונים - אומדן

מקום (Location): היכן מרכז המסיה של ההתפלגות.

- ממוצע (mean)** – מסמן לרוב בתו \bar{X} . מוגדר כך: $\text{Mean}(X) = \bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$.
- חציון (median)** – הערך שמעלוי חצי מהערכים בהתפלגות, ו מתחתיו החצי השני. הוא גם האחוון ה-50. מוגדר כך: $\text{Med}(X) = m \text{ s.t. } P(X \leq m) = P(X \geq m) = 0.5$.
- שכיח (mode)** – הערך הכי נפוץ במדגם. מסמן $(Mode(X))$.



בצד נבחר בינהם ומתי להשתמש בהם? למשל בהתפלגות עם זנב ימני, נחשב את שלוש האומדןים שלם. **הממוצע** הוא וודאי הסטטיסטי המוכר מכלם, אבל נשים לב כמה הוא מושפע ממערכות קיצוניים בהתפלגות. הוא לא בהכרח מעיד מה שהיינו רוצים שהוא יעד – סביר אולי ערך ממוצע התפלגות.

החציון לעומת זאת, אינו מושפע ממערכות קיצוניים, גם אם נוסיף להתפלגות ערך מקסימלי גדול פי 1000 מהערך הנוכחי – החציון עצמו לא יוזד. לצורך אוחזון הנוכחות, השכיח אויל הוא התשובה הטובה ביותר לגבי מיקום, אבל הוא יכול גם לצרכים אחרים, השכיח אויל הוא התשובה הטובה ביותר לגבי מיקום, אבל הוא יכול גם להטעות. קל לחשב על התפלגותות בהן השכיח הוא בעייתי מאוד במדד למיקום.

פיזור (Dispersion): אפשר לשאול על ערכים במיקומים ספציפיים כמו אוחזונים.

- אוחזון-Q:** אוחזון-Q הוא הערך שמתוחתי Q מההתפלגות האמפירית של המדגם, ומעליו $Q - 1$. מסמן באופן הבא: $Q(X, q) = n \text{ s.t. } P(X \leq q) = 1 - P(X \geq q) = n$.
- למשל האוחזון ה-75 הוא הערך שמתוחתי 75% מההמדגם ומעליו 25%, ונסמן $(Q(X, 0.75))$.
- טווות:** מוגדר בתו הפרש בין מוקסומים ההתפלגות למכנים שלה: $\text{Range}(X) = \text{Max}(X) - \text{Min}(X)$.
- IQR (IQR):** הטווח שבו נמצא 50% הערכים המרכזיים של ההתפלגות, ההפרש בין אוחזון 75 לאותו אוחזון 25: $IQR(X) = Q(X, 0.75) - Q(X, 0.25)$.

```
print(f'90th percentile: {np.percentile(X, 90) :.2f}')
print(f'Range: {np.max(X) - np.min(X) :.2f}')
print(f'IQR: {np.percentile(X, 75) - np.percentile(X, 25) :.2f}')
print(f'Variance: {np.var(X) :.2f}')
print(f'Standard Deviation: {np.std(X) :.2f}')

90th percentile: 8.95
Range: 22.00
IQR: 3.97
Variance: 9.12
Standard Deviation: 3.02
```

המדדים הנפוצים ביותר למדידת פיזור הם כמפורט:

- שונות: $\text{Var}(X) = \frac{1}{N} \sum_{i=1}^N (X_i - \mathbb{E}[X])^2$.
- סטיית תקן: $STD(X) = \sqrt{\text{Var}(X)}$.

לכל אחת הממדדים שראינו יש פונקציה פשוטה מספרית עוקבנת שבדאי להזכיר.

צורה (Shape): בשונות אנחנו שואלים מהו ממוצע הסטיות הריבועיות של X מהממוצע שלו. מסתבר, שאם ניקח את ממוצע הסטיות בחזקת 3, נקבל כמות שיכולה למדוד עד כמה ההתפלגות שלנו רוחקה מסימטריה – קוראים לה **skewness**. אם ההתפלגות סימטרית למדי נצפה לראות ערך 0 בקירוב. אם יש לה צד ימני, הסטיות הימניות מהממוצע גדולות יותר ונצפה לראות ערך חיובי. ולהיפך, צב שמאלית נצפה בערך שלילי.

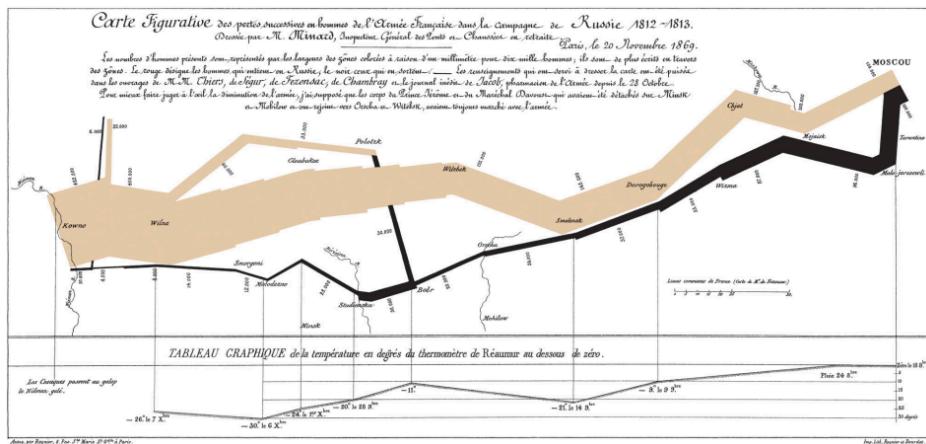
```
from scipy import stats
print(f'Skewness: {stats.skew(X) : .2f}')
Skewness: 1.22
```

$$Skew(X) = \frac{1}{N} \frac{\sum_{i=1}^N (X_i - \mathbb{E}[X])^3}{STD(X)^3}$$

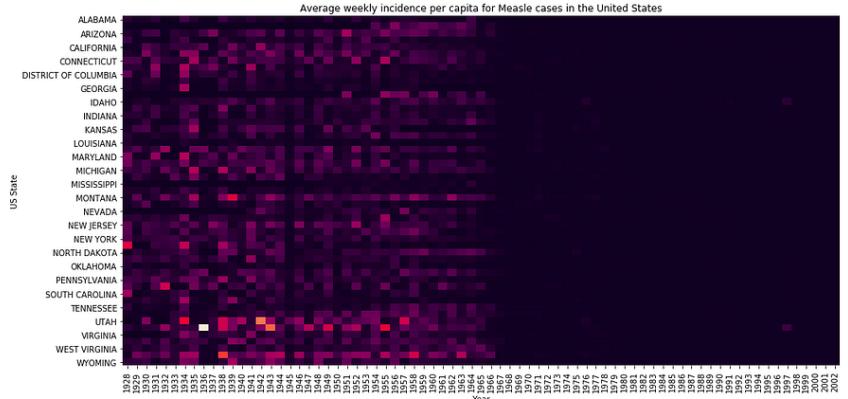
ניקוי והבנה של נתונים

תרשיים מתקדמים:

התחום של visualization/storytelling תופס תאוצה בשנים האחרונות, עד כדי כך שיש מדענים נתונים שהקיירה שלהם מוקדשת לדבר. דרכים שונות ו מגוונות להציג נתונים צצות כל יום. נסתכל על כמה מהן.



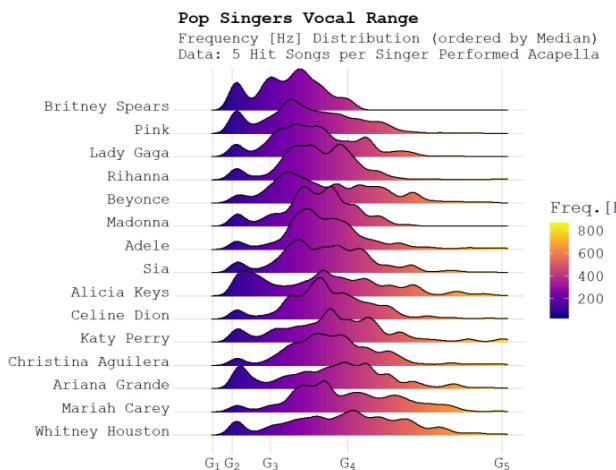
נפוליאון: הדוגמה הקלסית היא דזוקא ישנה (מהמאה ה-19) של מהנדס וגיאוגרפ צרפת. כאן הוא תיאר את הצבא של נפוליאון במסע שלו מפולין אל מוסקבה במטרה לבוש את רוסיה. עובי הרצועה מתאר את גודל הצבא, שכפי שניתן לראות התensus בצורה אכזרית במסע זהה. בין הרצועה והאורך שלה מתארים את הכוון והמרקח בהם הצבא נעה.



מפות חום: גם **תרשיים** זה מפורסם מאד. הוא מתאר את שיעור מקרי החצתה במדינות שונות בארצות הברית, החל משנת 1928 ועד המאה ה-21. המיחוד בתרשימים, הוא מבונן היחסות הבלתי מוחלטת של חצתה לאורך כל אריה"ב החל משנת 1963, השנה שבה פותח החיסון לחצתה.

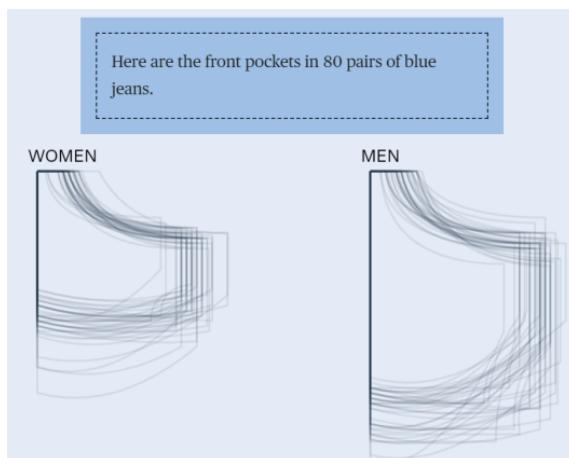
ספוטופי: דוגמה **לייזאליזציה** מ-2017 שמדגימה מספר דברים יפים. הבין אותה מדענו נתונים מסווני, והוא מראה לאורך ליקוי חמה מאד בולט שהוא על אדמה אריה"ב, את מספר ההשמעות של השיר "Total Eclipse of the Heart". ניתן לראות איך הpeak עוקב אחרי ליקוי החמה בצד אחד. בעוד המחשב, לייזאליזציה לא חייבת להיות סטטיסטית.

פרצופים: **תרשיים** על גבול המגוחך, פה מביעים נתונים באמצעות זומסום. אכן מדובר על אוכלוסייה במדינות שונות באלה"ב, כאשר כל מדינה מייצגת על ידי פרצוף אחד. הפרמטרים הם גודל הסנטר (אחוז האנשים שסובלים מהshanma במדינה), גוון הפרצוף (אחוז התושבים שאין להם ביתוח, בכל שהוא בהה יותר בר גודל האחוז), והאורינטציה של הפה (עצב ועד צחוק מבטא את אחוז התושבים מתחת ליקו השוני).



רכסים: תרשימים Ridge/Joy. היפוי בתרשימים זהה, זה שהוא מורכב מהרבה תתי-תרשיים שאנו חנו מכירם (של צפיפות) שמסודרים אחד מעל השני לפי החזון של התפלגות. במקרה של פנינו השתמשו בו כדי להשוות את המגע הקולי של זמרות פופ מפורסמות, מבריטני ספירס ועד ויטני יוסטן שהייתה לה מגע גם רחב מאד, וגם חזון התווים שהוא שרה היה הגבוה ביותר מבין כל הזמרות האלה.

A day in the life of americans: מתחא מה אנשים עושים במהלך כל שעה ביום, לפי סקר שנערך בארצות הברית. ניתן לראות שבבחוץ רוב האנשים מתארים שהם ישנים. רוב האנשים מתארים כיצד הם מתעוררים והולכים לעבודה וחזר חיללה.

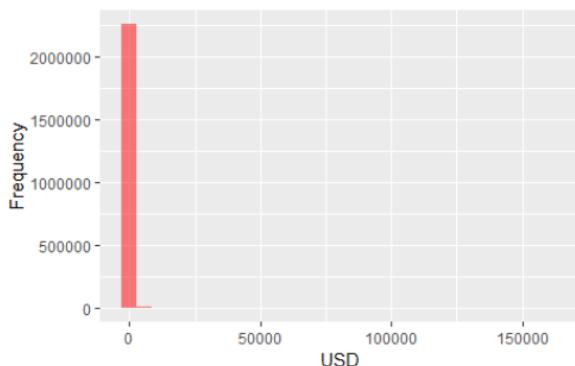


השערת מס' מדעיות הנתונים מאחריו **השערת ג'ינס**, דגמו 80 זוגות של ג'ינסים וشرطו את מדדי הביס. ניתן לתהות מדוע בסיסים במכנסי ג'ינס שמיועדים לנשים הם קטנים כל כך, עד כדי שהם לא פרקטיים בכלל. התרשים הוא הוכחה ניצחת לתופעה.

סיפור: אפידמיולוג שוודי יצר את **התרשימים** הבא. על ציר ה- x יש הכנסה של תושבים במדינה מסוימת, ועל ציר ה- y את תוחלת החיים שלהם. כל עיגול מייצג מדינה, והעיגולים גדולים בגודל האוכלוסייה במדינה (למשל הוודו וסין היכן שהם גדולים). ניתן להפעיל את התרשימים ולראות את גודל הכנסה, גודל האוכלוסייה וביצד הם משתנים לאורק זמן, לצד תוחלת החיים. אפשר להסתכל גם על פרמטרים אחרים במקום גודל הכנסה כמו במוות היולדת.

תיקוי נתונים:

קיים של DATA מהו חלק גדול משגרת יומו של מדען נתונים, והוא מושך מחקר גדול,ណזן בו כאן על קצה המזלג. מה יכול להיות מולכלה נתונים שצורך לנוקות אותם? אפשר לדבר גם על הנתונים עצם, וגם על המבנה שלהם.



שאנו חנו מדברים על מושגים כמו outliers, DATA לא נקי יכול להיות DATA עם תציפות כל ברחריגות שנקרה להן outliers. אלא לא חייב להיות תציפות לא חוקיות או טעויות הקלדה (למרות שגם מכאה צריך להציגן). באן למשל, יש לנו היסטוגרמה של כמה שורת אלפי רכישות בדולרים באתר ebay במשך כמה שבועות בשנת 2013. הסיבה שאנו חנו וואים באן רק מיל סיבי ה-0, היא כיacht הרכישות האלה הייתה של מכונית יוקרה בכ-160,000 דולר. המכונית הזאת היא כל ברחריגת ביחס לסכומי הרכישות האחרים, שהיא מעוותת את התפלגות לחלוין ולא מאפשרת לראות את צורתה – איפה נמצאת המשא של הרכישות.

aaaaaaaaaaaaaaaaaaaaargh, lolzi, jfjgfjhgjhgfjf,
dunno

נתונים אחרים רועשים מאד, מהאינטרנט בפרט, יכולים להיות נתונים טקסט. באן, מופיעות מילים אמריתות מתוך פוסטיםם מהאתר blogger.com. אם נרצה לקחת את הטקסט זהה ולאמן בעזרתו מודל שפה, אנחנו עלולים להיות בבעיה.



נתונים חסרים: כל מיינ מנגנונים יכולים לעמוד מתחייב נתונים חסרים, ובהתאם למוגנון נטפל בנתונים. בקורס זה לא עוסוק בנושא בהרחבה, אבל בהתעסקות עם נתונים אמיתיים יתתקל בעיה זאת, שכן מודלים רבים מלאה שנלמד לא יודעים מה לעשות עם תצפית שבמשתנה מסוימת, הנתן שלח חסר. דוגמה מפורסמת מהבחירות של 2015, בה המדגמים טעו לגורמי לגבי מבנה הקואלייטה המשטמן. הסוקרים טענו מאוחר יותר, שכנראה אנשים שמתכוונים להצביע בדרך מסוימת לא ענו, או פשוט שיקרו.

Australian Bureau of Statistics												
1800 Australian Marriage Law Postal Survey, 2017												
Released on 15 November 2017												
Table 5 Participation by Federal Electoral Division(s), Males and Age												
Gender apartheid												
Yeah NA												
18-19 years 20-24 years 25-29 years 30-34 years 35-39 years 40-44 years 45-49 years 50-54 years 55-59 years 60-64 years												
2.2. Lingfield	Total participants	292	1,059	1,406	1,653	1,515	1,516	1,710	1,730	1,753	2,574	
2.3. Lingfield	Eligible participants	572	2,910	3,789	3,995	3,607	3,506	3,645	3,331	2,960	2,456	
2.4. Lingfield	Participation rate (%)	51.0	36.4	41.4	42.0	43.2	46.9	51.9	59.2	64.1		
Primary keynotes												
Merged cells												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												
Solomon												



קוויז 3

שאלה 1: התAIMו את הנתונים המתוארים לפורמט המתאים ביותר לשמר אותם.

- חוקרי AI רוצים לבנות בסגנון ChatGPT בעל ידע של ימי הביניים בלבד. לצורך כך נסרקו דיגיטלי בכל כתבי ימי הביניים הידועים לאדם. **פורמט Plain Text**.
- נתונים מניסוי בפסיכולוגיה עוסקים בתפיסה מדידת של פרצופים. לכל נבדק הוצגו בסדר שונה אותם עשרים פרצופים של גברים ונשים (10 גברים, 10 נשים), פעם עם שיער ופעם ללא, ונרשמו זמן התגובה וההתשובה של הנבדק לשאלת מהו מין הפרצוף, כמו גם מין הנבדק, גילו ומשך ביצוע הניסוי. **פורמט CSV**.
- נתונים מסקר רב-שלבי של הלשכה המרכזית לסטטיסטיקה. דוגמאות לשאלות: 1) "האם יש לך ילדים" ובמידה והנਸרת עונתה בן היא הופנתה לפרט גיל ומין של כל ילך. 2) "allowも מפלגות את שוקלת להצביע לבחירות הבאות?" והנתקרת יכולה לסמן מתוך רשימת המפלגות המתמודדות בין אפס למספר המפלגות האפשריות. **פורמט JSON**.

שאלה 2: חוקרת העבירה שאלונים בתום השיעור בין הסטודנטים עם שתי שאלות רבות-ברירה: 1) האם אתם מעוניינים ללמוד את יתר הקורס בקורס מקוונת לחולון (זום)? א. כן ב. לא ג. לא יודעים. 2) מהו המין שלכם? א. זכר ב. נקבה ג. מעדים לא יודע.

- תרשימים בוקספלוט
- תרשימים פיזור
- תרשימים צפיפות
- **אף אחד מלאה** – במקרה זה, ניתן להיעזר ב-**mosaic plots** עבור נתונים count של מספר שדות שונים (כאשר המשתנים הם categorical/qualitative).

שאלה 3: טרנספורמציה לוג מעוותת את הנתונים, עלולה לגרום למסקנות מוטעות ואין להשתמש בה. **לא נכון**, ראיינו כי היא אכן יכולה לעזור לנו להעיף תוצאות חריגה מאוד מהנתונים.

שאלה 4: אורניתולוג עוקב אחרי זמני נדידה של עופות שונים מאיזור גאוגרפי X לאיזור גאוגרפי Y, בימים. ידוע שבממוצע לעומת הנמדדים לוקח שבועיים להשלים את המensus. אבל ברגע של היסח דעתן הzin החקור עבר את הציפורים "999". כיצד סביר שתיראה ההתפלגות של זמן הנדידה?

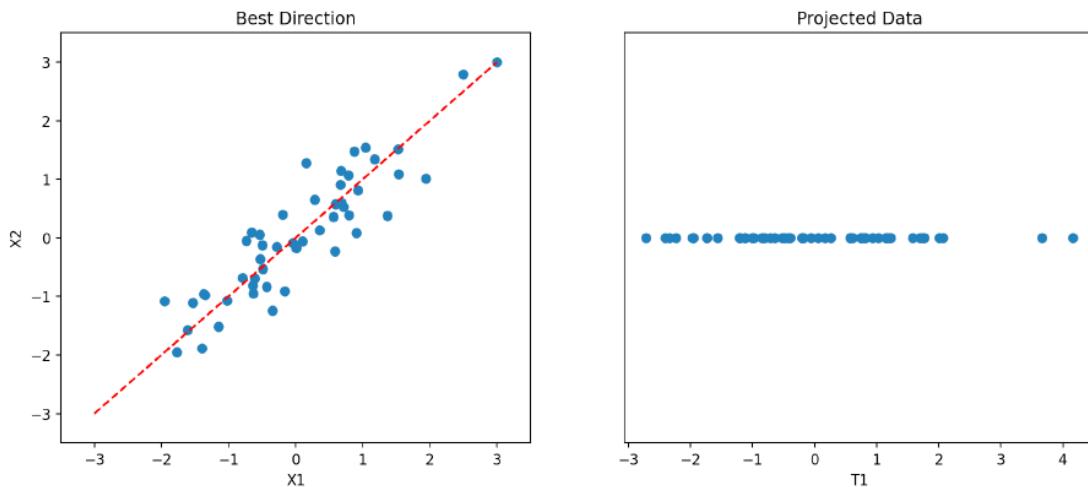
- **זמן שמאלי** עם חציו גדול מהממוצע
- **זמן שמאלי** עם ממוצע גדול מהחציון
- **זמן ימני** עם חציו גדול מהממוצע
- **זמן ימני עם ממוצע גדול מהחציון** – התכפיות 999 חריגה מאוד מהממוצע 14 לכיוון ימין, لكن נקבל זמן ימני, הממוצע מושפע ממערכות קיצוניים וכן יהיה גדול יותר מהחציון שפחות מושפע מכך

PCA

בעית ה-PCA

ניתוח גורמים ראשיים (principal components analysis) היא שיטה מתقدמת לחקר של נתונים, אך אפשר גם לראות בה הרבה מעבר. PCA היא הבסיס להרבה מאוד אלגוריתמים מתתקדים בחזית המחקר, אלגוריתמים שעוסקים בלמידת ייצוגים ממינימל נמור לדאנא מ민ימל עצום, על מנת שנוכל ללמידה מהדאנא זהה.

אינטואיציה: יש לנו נתונים עם n תצפיות ב- $p = k$ ממדים (משתנים). נרצה ליזג כל תצפית עם מספר אחד ולא 2, זאת תוך איבוד כמה שפחות מידע. בלומר, אנחנו רוצחים להוריד את המינימל מ- p ל-1. אם נשרט את הנתונים, ברור לנו אינטואיטיבית מה הכיוון בהם, מה הווקטור ששומר את מקסימום המידע. PCA יעשה בדיקות אלה – ימצא את הכיוון/ווקטור שלארכו נשמר מירב המידע בתנאים, ובאשר נמצא את הווקטור הזה נוכל להטיל את הנתונים עליו. כל תצפית תקבל את הערך שלה על הקו החשוב הזה וכן נקבל את מה שרצינו.



PCA: באופן כללי יש לנו n תצפיות עם k ממדים. נסתכל על הדאנאסט של נטפליקס: $14 = n$. נרצה להוריד את המינימל של הנתונים להיות $k \ll q$, לרוב 3 or $2 = q$. בדרך זו נוכל:

- להזות **כיוונים חשובים** בדאנא שמתמצחים אותו היפט.
- נוכל לעשות **תישים** לדאנא (דו/תלת-מימדי).
- נוכל גם אולי להזות **מבנה** בתנאים שלא הינו רואים אחרת, מתחקלים לאשכולות (clusters).

איך מבצעים PCA? הדרך הנאבית היא לבצע בחירה של q משתנים מתוך k . למשל, כמו שבחרנו 2 סרטים בדאנא ועשינו תרשימים פיזור של סרט אחד מול סרט אחר, אבל איבדנו כביה הרבה מידע. במקום זאת, נחפש **טולות/צירופים לינאריים** של המשתנים שיהוו **כיוונים מעניינים** בדאנא.

תנאי היעיה:

- יש לנו תצפיות: $x_1, \dots, x_n \in \mathbb{R}^p$.
- אפשר לסדר אותן מתחת לשני במטריצה $X_{n \times p}$.
- נניח לרוגע שבל עמודה (מימד) ממורצת – הסכום שלה/הממוצע שלו הוא אפס: $\sum_i x_{ij} = 0$. ניתן לראות בתנאים מען "ען" סביר הריאטי במרחב k -מימדי (נחשב על 2,3 = k כדי לתפוס את העניין).
- דרך לתאר כמה מידע יש במטריצה הזאת, שנרצה לשمر, הוא **הפיזור הריבועי של התצפיות מהמרכז שלhn**. אם הנחנו שהמרכז שלhn ב-0, אז הכמות הזאת תהיה פשוט הווקטורים בריבוע (ה-trace של מטריצת covariance של המדגם הזהה):

$$\sum_{ij} x_{ij}^2 = \|x_1\|_2^2 + \dots + \|x_n\|_2^2 = \text{tr}(X^T X)$$

- אנחנו רוצחים למצוא q כיוונים, שאם נטיל את המטריצה שלhn עליהם נשמר כמה שייתר מהכמות הזאת, הפיזור/השונות של הדאנא. כיוון ממשמעו וקטור $\mathbb{R}^p \in U$ מנורמה 1, וקטור יחידה/מנורמל: $1 = \frac{1}{\sqrt{p}} \mathbf{1}$.



אם כן, המטריה שלנו היא למצוא וקטור הטלה ששמור על הכיוון הרבה פייזר:

- אם $(0, \dots, 1) = u$ הוא הווקטור הטרויאלי. בשניטיל את המטריצה עליו, דבר זה יהיה שקול לבחירת המשטנה הראשוני.
- המספר שיציג כל תצפית יהיה הערך שלה בעמודה הראשונה.
- אם $\left(\frac{1}{\sqrt{p}}, \dots, \frac{1}{\sqrt{p}}\right)^T = v$ אנחנו נתונים משקלות שווה לכל קוודינטה (שומרים על כך שהוקטור יהיה מנורמל). הטלה של המטריצה תהיה שקול להליקת הממציע של כל המשטנים עבור כל תצפית.

הפייזר של הטלה uX : הוא הנורמה של הווקטור כך שמתקיים $u^T X^T X u = \|uX\|^2$. **לסיכום, נרצה למקסם את הפיזר של הטלה כך שהנורמה של הווקטור תהיה 1.** כשנמצא את הווקטור v_1 נקרא לו **ה-PCA הראשוני**, זה הכוון הטוב ביותר להטייל עליון.

$$v_1 = \arg \max_{v: \|v\|^2=1} \|Xu\|^2$$

principal components: אנחנו רוצים q כיוונים. איך נמצא את הכוונים הבאים? הכוון v_2 יהיה הכוון שמקסם את פיזור הטלה אחרי שהתחשבנו כבר בכוון v_1 . אפשר לראות שהדבר שקול לדרישת אורתוגונליות:

$$v_2 = \arg \max_{v: \|v\|^2=1, v^T v_1 = 0} \|Xu\|^2$$

הוקטור שאנו ממחפשים מקסם את פיזור הטלה (הנורמה שלה), כך שהוא מנורמל וגם אורתוגונלי לווקטור הראשוני. אפשר להמשיך הלאה עד שימושים q מנורמליים אורתוגונליים זה לזה. בשנשיג q וקטורים בלבד, כל אחד באורך k , וכך כל ה张יב אוטם זה לצד זה ולקבל מטריצה שנסמן כ- $W_{p \times q}$: **loadings matrix**.

נניח למשל כי X והוקטור שמצאנו הוא $(0, \frac{1}{\sqrt{3}}, \frac{2}{\sqrt{3}})$. זה אומר שהמשטנה הראשוני קיבל משקלות $\frac{1}{\sqrt{3}}$ והשני $\frac{2}{\sqrt{3}}$ והשלישי 0 הוא לא חשוב לבירור שמצאנו. הטלה עצמה תמיד תהיה חשובה לנו. אם נכפול הדטא שולמו במטריצה הטלה W עם q הכוונים הראשוניים, נקבל: $T_{n \times q} = X_{n \times p} W_{p \times q}$, כלומר דאטא ממוקד נמוך יותר!

Netflix Dataset על PCA

נזכיר בDATA של התחרות של Netflix, שambil דירוגים 1-5 על סרטים: X תהיה המטריצה ממימד $14 \times 10,000$ המכילה את 14 הסרטים הראשונים שדורגו על ידי 10,000 משתמשים.

ביצוע PCA:

- נתען את הנתונים באמצעות `pd.read_csv` ונשייר למטריצה X שאיתה נעבד.
- נזכור שניתוח PCA אנחנו עושים על מטריצה נתונים אחרי **הבראה מרכזית** (**centering**), ברגע היא לא ממורצת אז הממוצע של כל עמודה שונה מ-0. כדי למרכז הנתונים, נחסר מכל עמודה את הממוצע שלה.

```
# currently..
X.mean(axis=0)

array([4.1463, 4.1073, 3.7045, 4.3482, 4.0748, 4.5143, 4.4563, 3.7287,
       3.7546, 3.6749, 3.7316, 4.0183, 4.0168, 3.706 ])

# centering X: subtracting the mean from each column
X_centered = X - X.mean(axis=0)

print(X_centered.mean(axis=0))

[-9.66338121e-17 -3.95061761e-16  1.38555833e-16 -2.87059265e-16
 2.44426701e-16 -4.23483471e-16  2.94164693e-16  9.66338121e-17
 6.25277607e-17 -5.68434189e-17  1.43174361e-16  1.70530257e-17
 7.38964445e-17  7.36299910e-17]
```

ביצוע PCA על הנתונים: ניעזר בספריית **sklearn** ובחלקה **A**. כאשר נאתחל את האובייקט נוכל להוביל פרמטר שמתאר את ה- q הרצוי ($2 = n_components$ למשל). פועלות ה-PCA עצמה קורית בשימושה במתודה **fit** על המטריצה הממורצת שלנו.

```
W = pca.components_.T
print(W.shape)
```

קיבלנו את המטריצה W תחת השדה `components_`. נשים לב שמאחר שלא פירטנו מהו q , קיבלנו את מקסימום הכוונים האפשריים שזהו k , וקיבלנו בפועל $W_{p \times p}$. sklearn מחזיר את המטריצה כפי שישימוןו עם `transpose`. לרוב מעוניינים אותנו רק 2 או 3 הכוונים הראשונים.

(14, 14)

ניתוח התוצאות:

	mean_rating	PC1	PC2
Independence Day	4.15	-0.25	-0.26
The Patriot	4.11	-0.26	-0.06
The Day After Tomorrow	3.70	-0.32	-0.15
Pirates of the Caribbean	4.35	-0.15	-0.03
Pretty Woman	4.07	-0.23	0.45
Forrest Gump	4.51	-0.11	0.01
The Green Mile	4.46	-0.16	0.00
Con Air	3.73	-0.30	-0.32
Twister	3.75	-0.30	-0.14
Sweet Home Alabama	3.67	-0.30	0.59

נסתכל על עשרה הסרטים הראשונים, ממוצע הדירוג שלהם, והמשקלות שלהם בקטוים 1, 2, ביוני ה-PCA הראשונים, שתי העמודות הראשונות במטריצה W. ננסה למצאו פרשנות למשקלות של PC1 ושל PC2. המשקלות ב-PC1 הן בולן לאווטו בין. הגודל שלהם קשור מאוד לפופולריות הכלילית של הסרט. ככל שהדרוג הממוצע של הסרט גדול יותר, כך קטנות המשקלות בערך מוחלט. בלומר נראה שהכיוון בתוצאות שומר על היבנה, אם היינו צריכים לחתם מספר אחד לכל תצפית היינו משתמשים **כמה הצופה הזאת מסכימה עם הממוצע**.

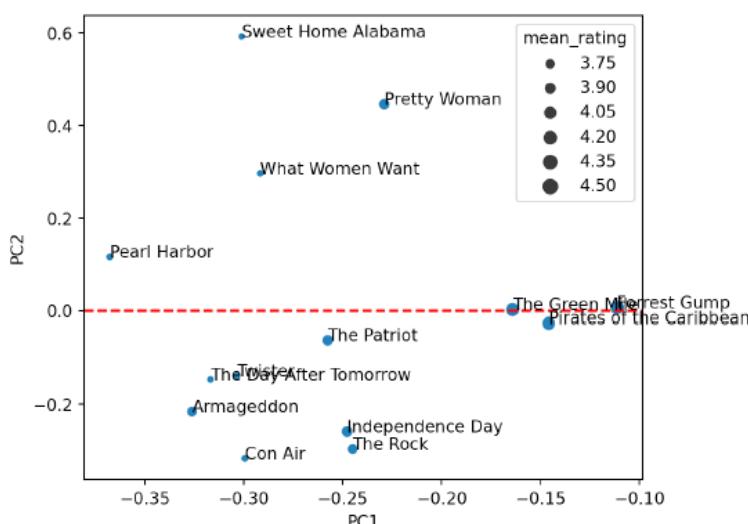
- לגבי הביוון השני צריך לבחיר קצת הסרטים כדי לראות את זה. **בתרשים הפיזור**, כל סרט הוא נקודה שהגדול שלה נקבע על ידי **כמה הסרט פופולרי**. ריבוב ה- x של הנקודה הוא המשקל שלה ב- PC_1 , ורכיב ה- y הוא המשקל ב-

PC2. את מה שראינו בטבלה אפשר לראות גם כאן – ככל שהסרט פופולרי יותר, כך הנקודה שלו מבדיל ב- PC_1 , וככל שהסרט רומנטי, כך הנקודה שלו מבדיל ב- PC_2 . מה מדובר?

- **מבדיל בין סרטים רומנים** (Pretty Woman) לבין סרטים אקסן (Con Air). אחרי שהתחשבנו בביון הראשון שסבב-zA ממנה צופים שאוהבים הסרטים רומנים, לבין צופים שאוהבים הסרטים אקסן.

חשוב להציג בבר unicso – זה פרשנות. כאן היא ממש קופצת מול העיניים, אבל הרבה פעמים זה לא כה פשוט.

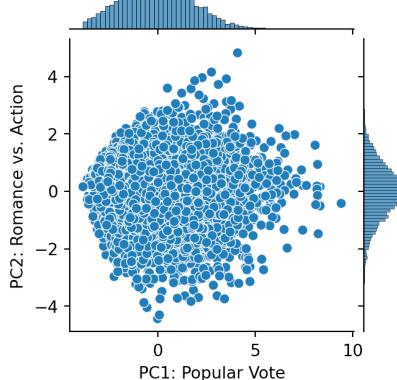
```
ax = sns.scatterplot(x='PC1', y='PC2', size='mean_rating', data=first_2_PCs)
for i, point in first_2_PCs.iterrows():
    ax.text(point['PC1'], point['PC2'], str(i))
plt.axhline(y=0, color='r', linestyle='--')
plt.show()
```



גם הדאטא של ההטלה מעניין אותנו. אחרי שנכפול את המטריצות קיבל $WX = T$. קיבלנו שוב מטריצת נתונים מסדר גודל 10,000 על 14, אבל העמודות שלה הן אורותונגוליות זו זו, והיא מחלקת את הפיזור/השונות המקוריים בצורה שונה למגרמי. העמודה הראשונה תהיה עם הפיזור היבי גדול, לאחר מכן השניה וכו'. בדרך כלל נתענין בהורדת מידע, בהכפלה של X רק בעמודות הראשונות של W . כדי לקבל מטריצת T עם 2 עמודות בלבד. עכשווי יוכל לבצע ויזואלייזציה לנواتים שלנו:

- על ציר ה- x ה zweן של כל צופה ב- PC_1 (עד כמה הוא דומה בדירוגים שלו לדעה הפופולרית, הסקאלה נעה מדיירגים גבוהים (אהבת סרטים) לנמוכים (שנאת סרטים)).
- על ציר ה- y ה zweן של כל צופה ב- PC_2 (עד כמה הוא מעדיף סרטים רומנים על פני סרטים אקסן, הסקאלה נעה מהעדפת אקסן לרומנטי).

בשלב זה מתגאים אשבולות מעניינים בדאטא, כאן קשה לראות. מעניין לציין את המגדר של הצופים, ולראות האם אכן נוכנה הסטיגמה שנשים אהבות יותר סרטים רומנים וגברים מעדיפים סרטים אקסן.

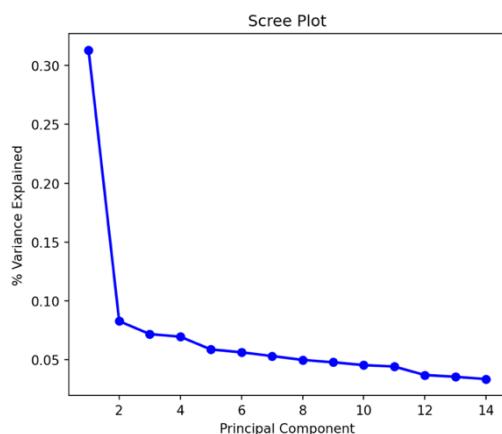


דוגמה – הסרט "אייזו מין שוטרת" (Miss Congeniality) – סרט יחסית רומנטית:

- PC1:
 - אילץ צוינים נצפה לראות לצופים שבוגרים ב-PC1 (שונאים את הסרטים)? ניקח את 100 הצלפים עם הציון הכי גבוהה, נראה שאכן נתנו ציון נמוך גם לסרט זה (בעיקר דירוג 1).
 - ומה עם הצלפים עם הציון הכי נמוך ב-PC1 (אהובים הכל)? אכן אהבו גם את הסרט הזה (בעיקר דירוג 5).

- PC2:
 - צופים שבוגרים ב-PC2 נתונים צוינים גבוהים יחסית לסרט.
 - צופים שנמנוכים ב-PC2 נתונים צוינים נמוכים יחסית לסרט.

מאוחר יותר נרצה לחזות כיצד ידרג משתמש שלא ראיינו בDATA את סרט כלשהו, על סמך שאר הדיווגים שלו. הכוונים שהאליה שגילנו יוכלים לעזור לנו מאוד, אפשר לומר שמצאנו משתנים חדשים בDATA – שאנו ידועים לשמורם הרבה מהשונות, וכן לבקש על הצלפהזה שאלה פשוטה שאולי לא הינו חושבים עליה בכלל – האם יש לך העדפה לסרטים רומנים על פני סרטים אחרים?



תרשים מעניין נוסף הוא **Scree Plot** שימושו בשדה_explained_variance_ratio של כל PC. לחוב נראה דפוס יוד בחנות, ה-PC הראשון מסבירים אחוז השונות מהשונות, והוא שבאים אחריהם מסבירים מעט מאוד. לעיתים נערמים בתרשימים זהה לדעת מה המימד הרואי לננתונים. למשל באנו, ככל להגיד שאם השתמש רק בשני ה-PC הראשונים למרות שהורכנו מימד מ-14 ל-2 שמרנו על כ-40% מהשונות במצטבר של הDATA.

אחוז השונות המוסברת על ידי PC1 גבוה פי כמה מאשר השונות המוסברת על ידי PC2, זו ירידת מחד הדעה. המשמעות היא שהתקונה שעומדת לאחר PC1 היא דומיננטית ממש בדירוג הסרטים, הרבה יותר מאשר אחרות.

שימוש ב-SVD

נסכם שוב את הבעיה של PCA. עבור מטריצה $X_{n \times p}$ אנחנו מוחפשים את הוקטור המנורמל שմביא לפיקסימים פיזור הטללה של X עליו. בשנשיג אותו, נרצה את הוקטור הבא שייה מנורמל, אורותוגונלי לקודם ומקסם את הפיזור של הטללה של X עליו. בכיה עד \hat{X} וקטוריים הבאים.

$$\begin{aligned} v_1 &= \arg \max_{v: \|v\|^2=1} \|Xv\|^2 \\ v_2 &= \arg \max_{v: \|v\|^2=1, v^T v_1 = 0} \|Xv\|^2 \end{aligned}$$

פירוק SVD: המפתח להשגת הוקטוריים האלה הוא פירוק SVD (singular value decomposition). אם מטריצת הנתונים שלנו ממשית, ונניח שמתקיים $p > n$, ניתן לפרק: $UDV^T = X$ כאשר:

- $U_{n \times p}$ עם עמודות אורותונורמליות, בלומר מותקיים $I_{p \times p} = U^T U$.
- $D_{p \times p}$ אלכסונית של האלבසון שלה נמצאים ערכים אי-שליליים (ערכים סינגולריים).
- $V_{p \times p}$ אורותוגונלית, בלומר מתקיים $I = VV^T = VV^T V = VV^T V = I$ (העמודות שלה הן בסיס אורותונורמלי של \mathbb{R}^p).

כדי להגדיר את הפירוק באופן ייחודי, נחליט שגם הערכים הסינגולריים על האלבסון של D **מסודרים מגדל לקטן**, והם **שניים** זה מזה: $d_p > d_{p-1} > \dots > d_1$. נסמן את העמודות של V כך: $[v_1, \dots, v_p] = V$ ושל U באופן דומה: $[u_1, \dots, u_p] = U$.

נשים לב מה קורה להטללה של X על אחד מוקטורי הבסיס של V :

$$Xv_j = UDV^T v_j = UDe_j = u_j d_j$$

העמודה v_j יוצרת 0 במכפלה עם כל העמודות ב- V כי V אורותוגונלית, ואילו עם העמודה v_j היא יוצרת 1 כי הוא וקטור ייחידה (הנורמה של 1). זה אומר שבמכפלה התוצאה היא $d_j e_j$. בלומר נשאר לנו רק הריבוב ה- j של המכפלה UD .

הקשר ל-PCA: אנחנו מעוניינים למצוא $\mathbb{R}^p \in v$. נביע אותו בבסיס של V המטריצה, בצל"ל של העמודות:

$$v = a_1 v_1 + a_2 v_2 + \dots + a_p v_p \text{ s.t. } a_1^2 + a_2^2 + \dots + a_p^2 = 1$$

אם כך, מהי ההטלה של X על אותו v ?

$$Xv = a_1 d_1 u_1 + \cdots + a_p d_p u_p \Rightarrow \|Xv\|^2 = a_1^2 d_1^2 + \cdots + a_p^2 d_p^2 \leq d_1^2$$

הערך הבci גדול ותקבל כאשר $1 = a_1$ ואז הפיזור שיתקבל הבci גבוה. המשמעות היא $v = u$ כלומר העמודה הראשונה של המטריצה V . גילינו ש- u הוא הוקטור PC1 שיביא למסימום את פיזור ההטלה, ובבהכרח מתקיים $d_1^2 = \|u\|^2$.

שימוש בפירוק ערכים עצמיים

נזכר בבעית ע"ע: נרצה למצוא ומטריצה ריבועית $A_{p \times p}$ אם מתקיים: $u = Av$, כאשר g הוא הע"ע. מבחינה גאומטרית כל מה שהמטריצה עשתה לוקטור הוא להאריך אותו או לכווץ אותו. למציאת הוקטור זהה שימושים רבים. פירוק ערכים עצמיים של A הוא המכפלה הבא:⁻¹ $A = V\Lambda V^T$ באשר:

- V מטריצה ריבועית שהעמודות שלה הן הוקטוריים העצמיים.
- Λ מטריצה אלכסונית שעל האלכסון שלה נמצאים הע"ע.

הערות:

- אם A מטריצה ממשית וסימטרית – אז V גם אורתוגונלית וההופכי שלה הוא transpose שלה ונוכל לרשום $V^T = A$, והע"ע שלה הם ממשיים.
- אם A היא חיובית למחצית (PSD) אז $0 \geq \lambda_j$, הע"ע אי-שליליים.

הקשר ל-PCA: אפשר לבטא את הבנייה שלנו בבעית אופטימיזציה אם נשתמש בכפול**י** לגראנץ. אנחנו רוצים למקסם את הפיזור עם אילוץ על הנורמה של הוקטור:

$$\max_{v_1} v_1^T X^T X v_1 + \lambda_1(1 - v_1^T v_1)$$

אם נגזר את הבמות הזאת לפי הריבבים ב- v נקבל את הביטוי:

$$2X^T X v_1 - 2\lambda_1 v_1 = 0 \Leftrightarrow X^T X v_1 = \lambda_1 v_1$$

זאת ב**דיק שגואה שגדירה ע"ע של המטריצה $X^T X$ – מטריצת covariance של מדגם הנתונים**. לאחר שככל מטריצה כזו היא ממשית סימטרית וחובית – $0 \geq \lambda_1$.

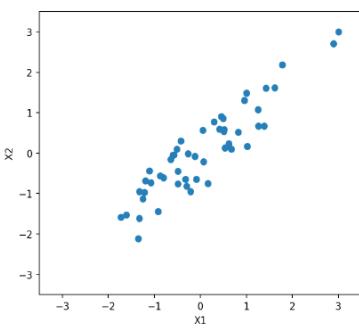
כפול את הביטוי ב- v_1^T בשני צדי המשוואה משמאלי, נקבל **שהפיזור עצמו שווה לע"ע**. אנחנו רוצים פיזור כמה שיוצר גדול, אך ניקח את ה"ע" שמתאים לע"ע הגודל ביותר.

נסכם:

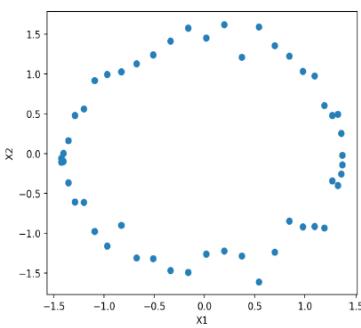
- מוחשים את המטריצה $W_{p \times p}$ שעמודותיה הם ה"ע" של המטריצה $X^T X$ והע"ע שלה מסודרים מגודל לקטן.
- **הע"ע עצם שווים לפיזור או לנורמה של ה"ע – כיווני-ה-PC.**
- **הקשר ל-SVD:** במקרה הזה $V D^2 V^T = X^T X$, כאשר D אלכסונית שעל האלכסון שלה נמצאים הערכים הסינגולריים בירובע, שהוא בדיק פירוק ערכים עצמיים שראינו. ככלומר המטריצה SVD היא בדיק המטריצה V מפירוק הע"ע והמטריצה שאנו מוחשים. הע"ע הם הערכים הסינגולריים בירובע.

PCA לא לינארי

Previously...



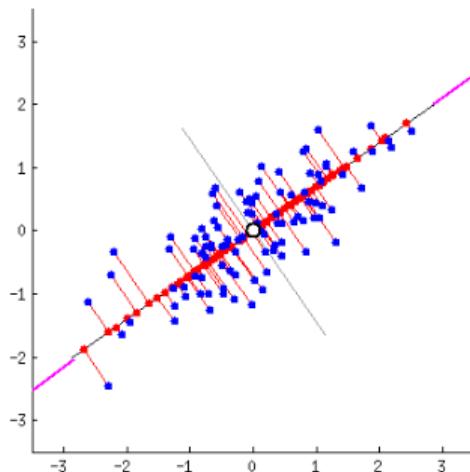
Now:



PCA לא לינארי: PCA מבוסני מסויים היא שיטה לינארית. מה קורה אם הנתונים לא מסתדרים בצורה זאת. קודם קודם היה כיוון, וקטור ברור בנתונים. מה קורה אם הנתונים יוצרים מעגל? הוא ימצא כיוונים שכולם טובים כמו אחר, אך ברור שיש כיוון שלאורכו הנתונים משתנים, אך הוא לא ניתן לצ"ל של x_2 , x_1 , לא ניתן לביטוי בוקטור במערכת הצירים הזאת. הנתונים נוצחו באמצעות סיבוב. יש שיטות למצב זה: AutoEncoders ו שימוש ב-Kernel PCA.

תרגול 3 – PCA

מוטיבציה:



בשים לנו דאטא שמשמעותו לרוב במטריצה X ממימד $k \times n$ (ה שורות (משתנים) ו- k عمودות), נרצה לקבל ויזואליזציה ותובנות ברורות גם באשר k גדול מאוד. ככלומר, נרצה לבצע הוכחת ממידים ולהציג למטריצה $q \times n$ אשר $k \ll q$. המימד הנמוך הופך את הדאטא לשימושי יותר – עבור clustering, עבור אימון של רשותות וכו'.

אינטואיציה: דמיין שדה טרס שהוא בדו-מימד, אך אנחנו יכולים לקלוט רק בחד-מימד. אנחנו רוצים למצאו את הכוון שבו הנתונים משתנים הכוי הרבה, כאשר הקו השחור מגיע למטריה, נשמר על הפיזור הרב ביותר בנתונים. ביצענו הטלה מדו-מימד אחד-מימד, כל נקודה קיבלה מספר חד-מימי אחד. אנחנו ממערים את reconstruction error בין הנקודה הכחולה לאדומה.

הסבר אלגברי: יש לנו את המטריצה $X_{n \times p}$ ונרצה לבספל אותה בוקטור $x_1 \in \mathbb{R}^p$ שהוינה משקלות לכל אחד מהמשתנים, כך שהפיזור יהיה מקסימלי, באילו ערך שהונרמה של הוקטור היא $1: max \rightarrow \|u\|^2 = \|uX\|^2$ כאשר u הוא פיזור כיוון הגדירה של שונות של וקטור, אם אנחנו יודעים שהממוצע שלו הוא 0 (אם לא, נחסר מכל עמודה את הממוצע שלה, ונקבל X ממורכז).

לאחר שימוש בכופלי לגראנד, קיבל בעיה של ערכים עצמאיים: $u = X^T X$ כאשר המטריצה $X^T X$ שנקראת מטריצת covariance היא סימטרית, ממשית, חיובית. נרצה למצוא את הערכים שמקיימים את המשוואה. ביוון שהיא חיובית $0 \leq \lambda$. הוקטורים שאנו ממחפשים הם u , וה- g הם $u^T u$.

לבסוף נוכל לבצע את הטלה: $T_{n \times q} = W_{p \times q} X_{n \times p}$ המטריצה W מכונה **loadings matrix**, וניתן למצוא שם לעיתים מבנים מעניינים. בנוסף, ניתן להסתכל על **אחד השונות המוסברת**:

- נניח כי לקחנו את $X_{n \times p}$ ועשינו PCA עם $p = q$ לקבלת $T_{n \times p}$.
- במטריצה X העמודות לא בהכרח אורותוגונליות, אך העמודות ב- T הן **אורותוגונליות**.
- נסתכל על מטריצת השונות $S = \begin{bmatrix} Var(x_1) & \dots & Cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ Cov(x_n, x_1) & \dots & Var(x_n) \end{bmatrix}$. לאחר ה-PCA יש לנו את המטריצה הבאה: $S = \begin{bmatrix} Var(t_1) & \dots & Cov(t_1, t_n) \\ \vdots & \ddots & \vdots \\ Cov(t_n, t_1) & \dots & Var(t_n) \end{bmatrix}$.

עבדיו נוכל להגיד שהשונות של t_1 (ה-PC הראשון) היא **הבי גודלה**. אפשר להסתכל על השונות הזאת יחסית לשונות.

הדגמה ללא sklearn:

נתבון בדאטא על פיצות. ניפטר מהשורות של id ו-brand כי נרצה לעבוד רק עם דאטא כמותי. יש לנו 300 פיצות ו-7 שדות. ככלומר בבעיה שלנו נקבל $7 = p = 300 = n$. נעזר ב-StandardScaler על מנת לקבל מטריצה X_s מתוקנת: נחסר את התוחלת ונחלק בסטיית התקן, עבור כל עמודה j : $x_j^* = \frac{x_j - \bar{x}_j}{s_j}$. נקבל כי הממוצע של כל עמודה הוא 0, וסטיית התקן של כל עמודה היא 1.

```
eigvals, eigvecs = np.linalg.eig(X_s.T @ X_s)
print(eigvals)
print(eigvecs.shape)

[1.25153457e+03 6.87137216e+02 1.24368703e+02 2.85522682e+01
 8.30310542e+00 1.01282806e-01 2.85563410e-03]
(7, 7)
```

```
W = eigvecs
v1 = W[:, 0]
print(v1.shape)
v1 = v1[:, np.newaxis]
print(v1)
print(v1.shape)

(7, )
[[ 0.06470937]
 [ 0.3787609]
 [ 0.44666592]
 [ 0.47188953]
 [ 0.43570289]
 [-0.42491371]
 [ 0.2444873 ]]
(7, 1)
```

מבצע PCA ידנית:

- באמצעות `linalg.eig` ביצע פירוק לפי ערכים עצמיים למטריצה $X^T X$ ונתקבל את מטריצת $W_{p \times p}$.

- ניקח את העמודה הראשונה של המטריצה ונוסיף לה מימד כדי שהיא תהיה וקטור עמודה. אלו המשקלות של המשתנים שלנו.

```
PC1 = X_s @ v1
print(PC1.shape)
print(PC1[:5])
(300, 1)
[[5.01034284]
[5.02375538]
[4.8054393 ]
[4.4695434 ]
[4.47189256]]
```

- נבע הטלה של ה-PC הראשון, וככפי λ_1 ש- X_s . קיבלנו וקטור באורך 300, כל פיצה קיבלה מספר.
- אחוז השונות המוסברת: ניקח את $\frac{\lambda_1}{\sum \lambda_i}$, או באופן שקול ניקח את השונות של PC1 חלקי סכום השונות לאורך האלכסון של מטריצת covariance. קיבל כמעט 60%.

```
# pct of variance explained is proportional to the 1st eigenvalue
eigvals[0] / eigvals.sum()
0.5959688423344788

np.var(PC1) / np.diagonal(np.cov(X_s.T)).sum()
0.593982279526697
```

- אפשר לראות שהנורמה של PC1 היא הערך של הע"ע הראשון λ_1 .

```
# this also means the first PC's norm is the 1st eigenvalue
(PC1**2).sum()
1251.5345689024055

eigvals[0]
1251.5345689024073
```

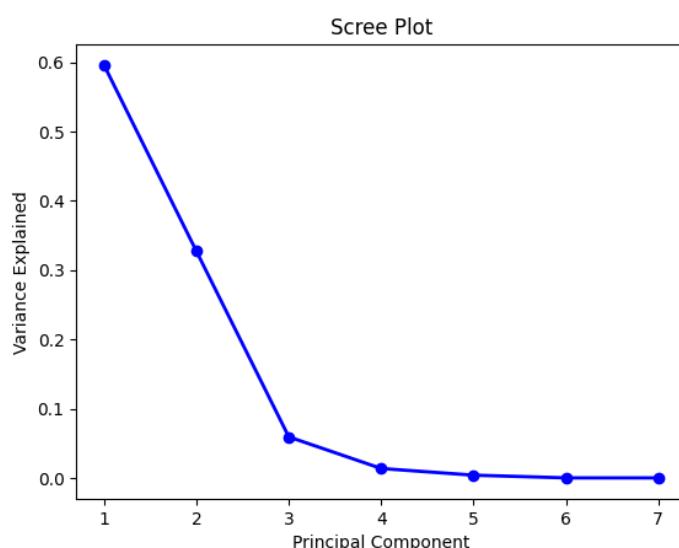
עבודה עם sklearn

```
from sklearn.decomposition import PCA
pca = PCA(n_components=7)
_=pca.fit(X_s)
# get that loadings matrix W
W = pca.components_.T
print(W.shape)
# compare this to our v1
print(W[:, 0])
# compare to our eigenvalue
print(pca.explained_variance_ratio_[0])
(7, 7)
[ 0.06470937  0.3787609   0.44666592   0.47188953   0.43570289 -0.42491371
 0.2444873 ]
0.5959688423344786
```

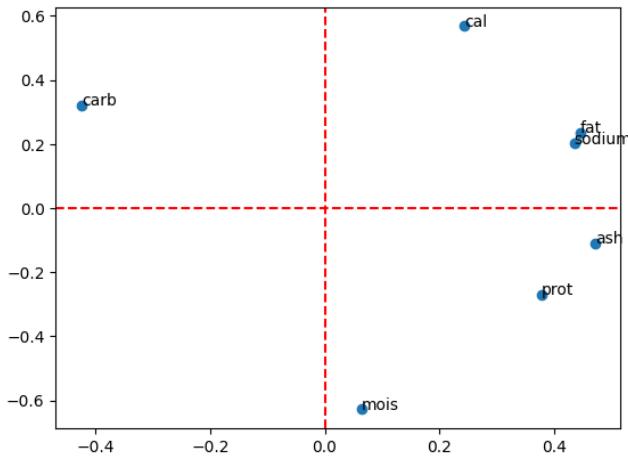
- ניתן במחלקה PCA עם מספר ה-components שנרצה (הפרמטר q). כדי להריץ אותו נקרא fit על X_s המנורמל.
- כדי לקבל את loadings matrix או בצע transpose את components השדה .components השדה .explained_variance_ratio את אחוז השונות המוסברת ניקח את .explained_variance_ratio_.explained_variance_ratio_[0]

גרפים לדוגמה:

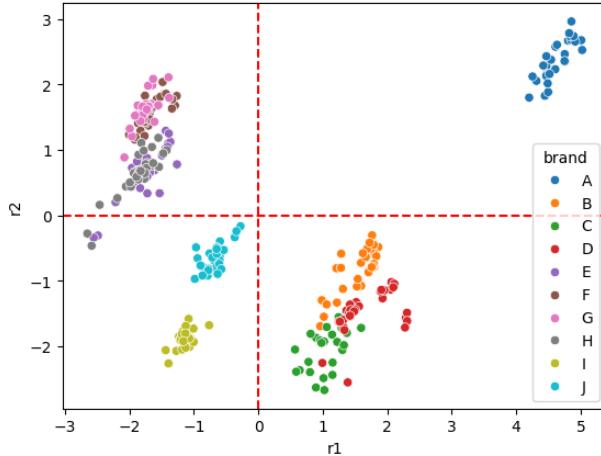
```
# might want to draw a scree plot, showing explained variance ratio per PC:
import matplotlib.pyplot as plt
plt.plot(np.arange(1, 8), pca.explained_variance_ratio_, 'o-', linewidth=2, color='blue')
plt.title('Scree Plot')
plt.xlabel('Principal Component')
plt.ylabel('Variance Explained')
plt.show()
```



```
# visualizing first 2 loadings W with original features
v1 = W[:, 0]
v2 = W[:, 1]
plt.scatter(v1, v2)
plt.axhline(0, color='r', linestyle='--')
plt.axvline(0, color='r', linestyle='--')
for i, txt in enumerate(pizza.columns[2:]):
    plt.annotate(txt, (v1[i], v2[i]))
plt.show()
```



```
# visualizing first 2 projections with observations,
# notice the transform method which takes care of multiplication for you
XW = pca.transform(X_s)
r1 = XW[:, 0]
r2 = XW[:, 1]
import seaborn as sns
pca_df = pd.DataFrame({‘r1’: r1, ‘r2’: r2, ‘brand’: pizza[‘brand’]})
sns.scatterplot(x=‘r1’, y=‘r2’, hue=‘brand’, data=pca_df)
plt.axhline(0, color=‘r’, linestyle=‘--’)
plt.axvline(0, color=‘r’, linestyle=‘--’)
plt.show()
```

שאלות:

1. למה דרישנו $1 = \|\mathbf{v}\|$ ב-PCA לגבי ההטלה?
 a. אפשר היה לדרש גם $2 = \|\mathbf{v}\|$. יש לנו את הקריטריון $\max_{\mathbf{v}} \mathbf{v}^T \mathbf{X} \mathbf{v}$ $\rightarrow \max_{\mathbf{v}} \|\mathbf{v}\|^2 = \mathbf{v}^T \mathbf{X}^T \mathbf{X} \mathbf{v}$. חיבים אילוץ במשהו על בעיית המקסימיזציה שלנו. ככל שהונומה גדלה גם הערך יגדל.
2. דני קיבל תוצאות מבן של 100 סטודנטים שבו היו 5 שאלות בנושא הבאים: PCA, Prob, NN, Clustering, Boost. דני ביצעה PCA על התוצאות, ומראה את שני הכיוונים הראשוניים שתופסים 95% מהשונות. נתונה המטריצה \mathbf{W} שהיא

$$\mathbf{W} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & -1 \\ 0 & 0 \end{pmatrix} \quad \text{(loading matrix)}$$

- a. הממוצע של תשובה לשאלות על NN, PCA, Prob, NN, Clustering, Boost הוא הכיוון בדעתו שנותן את הבי הרבה פיזור, שומר על שונות מksamילית. ביוון שהמשקלות שוות, אז הממוצע שלהן הוא הדבר הכי חשוב בדעתו, מבחינת הצורה שבה הסטודנטים ענו על השאלות.

- b. סטודנטים שקיבלו תוצאות גבוהות ב-PC2 ענו נכון על PCA ולא נכון על PC2: ניתן לראות ב-Clustering.

- c. אם כל PC מייצגconstruct איזה משקלות, אז מושג **boosting** לא מושג על ה-construct הראשון: המשקלות היא 0, וכך לא משנה איזה צוין קיבל בשאלת boosting זה לא מושג.

- d. על אף ש-boosting מקבל 0 גם ב-PC1 וגם ב-PC2, זה לא אומר שאין **אף** שונות באיך שהסטודנטים ענו על השאלה זו. יכול להיות שהוא מופיע ב-PC הבא...

```
n = 1000
x1 = np.random.normal(size=n)[:, np.newaxis]
x2 = np.random.normal(size=n)[:, np.newaxis]
x1 = x1 - x1.mean(axis=0)
x2 = x2 - x2.mean(axis=0)
x1 = x1 / np.linalg.norm(x1)
x2 = x2 / np.linalg.norm(x2)
z1 = (x1+x2)/2
z2 = (x1-x2)/2
Z = np.concatenate([z1, z2], axis=1)

pca1 = PCA(n_components=2)
pca1.fit(Z)
pca1.explained_variance_ratio_
array([0.50514366, 0.49485634])

pca1.components_
array([[ 0., -1.],
       [ 1., -0.]])

np.linalg.eig(np.cov(Z.T)) # X.T @ X / X.shape[0]
EigResult(eigenvalues=array([0.00050182, 0.00049919])
```

3. נתונים שני וקטורים $\mathbf{x}_1, \mathbf{x}_2$ באורך n עם נורמה 1 וממוצע 0. נגיד $\mathbf{z}_1 = \frac{\mathbf{x}_1 + \mathbf{x}_2}{2}, \mathbf{z}_2 = \frac{\mathbf{x}_1 - \mathbf{x}_2}{2}$. תהיו $Z_{n \times 2} = [\mathbf{z}_1, \mathbf{z}_2]$ מטריצה. מוצעים PCA על Z . מה ניתן להגיד המפתח הוא $\mathbf{z}_1 - \mathbf{z}_2$, \mathbf{z}_1 כל תורם משהו אחר (המ לא צ"ל אחד של השני), וכשנעשה PCA כל אחד מה-PC יתרום 50 אחוז מהשונות.

- a. לבן: אם $(1,0) = PC1 = (0,1) \vee (0,-1) = w_1$ אז $w_2 = (0,1)$
 b. בנוסף ניתן לומר כי ה"ע" הראשון של $Cov(Z)$, חלק סכום כל שאר ה"ע" הוא **0.5**.

קוויז 5

שאלה 1: בשיעור למדנו שלא ניתן לבצע PCA על דatasets שבו יש יותר משתנים (k) מתחפויות (n). לא נכון כלל, הטעמך לנו במקרה הנפוץ יותר של יותר תחפויות מאשר משתנים ($k > n$) אבל PCA הוא בעצם פירוק SVD שאפשר לעשות לכל מטריצת DATA, במקרה זה כמות הרכיבים האפשריים שאפשר למצוא היא $1 - n$ כי הדרגה של מטריצת covariance הוא לפחות $1 - n$.

שאלה 2: נניח שבדאטה יש $4 = k$ משתנים, והתקבלו משקלות עבור ה-PC הראשון (לפני נורמול לאורך 1):

- משתנה ראשון: 0.5
- משתנה שני: 0.25
- משתנה שלישי: 0.25
- משתנה רביעי: 0

איזה מההיבטים הבאים נכון?

- הצירוף הליניארי שומר על הכיו פחות שונות בDATA הוא לחת ממוצע משוקלל של 50% מהמשנה הראשון, 25% מהמשנה השני, 25% מהמשנה השלישי ולהתעלם מהמשנה הרביעי.
- הצירוף הליניארי שומר על הכיו הרובה שנות בDATA הוא לחת ממוצע משוקלל של 50% מהמשנה הראשון, 25% מהמשנה השני, 25% מהמשנה השלישי ולהתעלם מהמשנה הרביעי – PC1 תמיד נבחר כך שהוא מקסם את השונות בנתונים.
- לא יתכן לקבל וקטור משקלות זהה גם לאחר נורמול לאורך 1.
- המשנה השלישי בלתי תלוי בשלוש המשנים האחרים.

שאלה 3: סרט חדש יצא לאקרנים, "לרכוד עם עקרות בית". הסרט הוא קומדי רומנטית והוא מקבל דירוגים נמוכים באופן כללי. כיצדסביר שתיה המשקלות היחסית של הסרט ב-PC הראשון והשני בהתאם, לפי ניתוח PCA של הנתונים של נטפליקס?

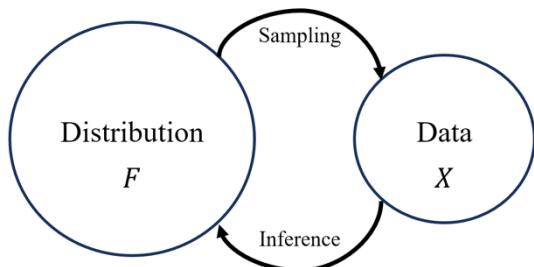
- נמוכה, נמוכה
- נמוכה, גבוהה – PC1 הוא מייצג את הדעה הפופולרית (נמוך לגבהה), ו-PC2 אקסן לרומנטי. לכן נצפה למשקלות נמוכה בראשוון (כי הסרט מקבל דירוגים נמוכים), ומשקלות גבוהה בשני (כי הוא סרט רומנטי).
- גבוהה, נמוכה
- גבוהה, גבוהה

שאלה 4: דניאל, סטודנטית בחוג ל孔ונו, מביטה בנתונים וקובעת: בסופה של דבר מה שהכי מבדיל בין צופים זה עד כמה הם מסכימים עם הממוצע הכללי. הרבה פחות מזה העדפה שלהם לסרטים רומנים או סרטי אקסן. נכון, ניתן לראות ב-scree plot PC1 מסביר לבודו 30 אחוזים מהשונות המוסברת, פי 4 מ-PC2.

2 – הסקה סטטיסטית

הסקה סטטיסטית – חלק א'

התמונה הגדולה



הסקה סטטיסטית היא נושא מורכב, ונסה לתת "מבט עלי" בנושא, ולהשתמש במה שכתוב בתוכנות כדי להבהיר את המושגים השונים. התושים של פננו מתאים בצורה פשוטה את מטרת ההסקה הסטטיסטית. אנחנו מניחים שיש בעולם מציאות מסוימת, אובלוסיה, שבה המדד שמעוניין אותנו מתפלג בצורה מסוימת (התפלגות F). העולם כל כך גדול שלא ניתן לראות את כלו, יש לנו מגבלות (קibilitות/תקציב) לדוגם רק מוגן סופי מהאובלוסיה הזאת (**Sampling**). לעומת זאת המוגן רוחים להסיק בחזרה אל ההתפלגות הגדולה (**Inference**).

אילו דוגמאות יש לעולם/לאובלוסיה עליהן היינו רוחים למדוד? למשל, בזמן לבחירות לראשות הממשלה בין שני מועמדים, היינו רוחים לדעת למי מהם יש רוב באובלוסיה. בדוגמה הבחירה שלנו, היינו רוחים לטעות אם לצירום אימפרסיוניסטיים יש רמה גבוהה יותר בממוצע של צבע אדום. בשני המקרים האלה – כמעט בלתי אפשרי לאוסף את כל האובלוסיה, יש צורך לאסוף מוגן ממנה. במקרה של לבחירות זה יהיה סקר לבחירות. במקרה של הבחירה, נניח שיש לנו תקציב לבדוק את רמת האדום רק ב-30 צירום מכל סוג.

בדיקה השערות (Hypothesis testing): הפרדיגמה שלנו נקראת בדיקת השערות.

- יש לנו עולם שנכנה אותו "עולם האפס", הנחה בסיסית שכולם יודעים. ננסה להגיד ממשו חדש ומשמעותי ולהתרחק מהעולם הזה. בדוגמה של הבחירה לראשות הממשלה, ברירת המחדל היא שלמוצעדים יש תמייבה זהה. בדוגמה של הבחירה, ברירת המחדל היא שיש לצירום משני הסוגים אותה רמה של צבע אדום.
- בחוקרים, נרצה להגיד ממשו חדש. המוגן יגיד את המידע שלו ויתקדם מעולם ה-*H₀*: האם המוגן משכנע מספיק שלמוצעדים מסוים יש יותר תמייבה? או שלצירום מסוים יש יותר צבע אדום?
- נכמת את מידת ההפתעה הזאת באמצעות **value-k**: שמוודע עד כמה המוגן קונסיסטנטי עם השערת האפס. ככל שהוא יהיה קטן יותר כך נהיה מופתעים יותר, תחת ההנחהות הידועות שלנו – ואולי אף בדחה אותן.

דוגמה מעולם המשפט: בבית משפט חשוד לא מושרע אלא אם כן הוכח אשםתו מעלה אייזהו סף ספק סביר.

- אפשר לראות את ברירת המחדל בעולם זה, שהחושד חף מפשע, בהשערת האפס.
- המודגם/הנתונים הם הראיות.
- השאיפה של התובע היא להוכיח את אשמת החשוד מעבר לספק סביר – זה הניסיון שלו להראות שהראיות לא תואמות את הנחת החפות.

הבדל הוא שברוב המשפטים, הדבר שסביר אם הדאטה מרוחיק אותו מהשערת ה-*H₀* הוא הניסיון והאינטואיציה של השופט. בעולם מדעי הנתונים, אנחנו שופטים. ובדי לקבוע האם לדוחות את השערת האפס נראת שנחשה הסתבירות.

עקרון מפתח: **שתי השערות – האפס והאלטרנטטיבית – אין סימטריות**. אין בין הכרעה בין שני מצבים שככל יכול להחליף את השני. לרוב השערת האפס היא ברירת מחדל שצריך לעבוד קשה כדי לצאת ממנה אל השערת אלטרנטטיבית. לכן גם מפקדים על המינוח של **לדוחות/לא לדוחות את השערת האפס**.

בעולם של גילויים מדעיים:

- השערת האפס: לא גילינו שום דבר חדש (חליקן חדש, השפעה גנטית בלהשי).
- דוגמאות: *Higgs boson*, מחקרים למציאת גנים שגורמים למחלות.
- value-k**: חזק העדות שכן יש כאן ממשו חדש.

דוגמה הבחירה: ראיינו שציירים אימפרסיוניסטיים (IM) נטו להשתמש ביוטר צבע כדי ליצור סיטואציות עליזות יותר, לעומת העולם הריאליסטי (RE) הקודר של המהפכה התעשייתית במאה ה-19. נרצה לבדוק האם יש באמת יותר אדום (עליז) ב-IM? במדגם הלמידה שלנו (wikiart) יש 8 צירום IM ו-8 צירום RE. ניתן לחקות רק מוגן ולשאול האם הוא מהווע עדות חזקה מספיק כדי להגיד שציירים IM הם אדומים יותר במשמעות, נעשה זאת עם הפרדיגמה של בדיקת השערות.

```

real_sample = get_images_matrix(folder + 'realism_train.csv', fold)
impr_sample = get_images_matrix(folder + 'impressionism_train.csv')

real_red = real_sample[:, :, :, 0].mean(axis = (1, 2))
impr_red = impr_sample[:, :, :, 0].mean(axis = (1, 2))

print(real_red[:10])
print(impr_red[:10])

[161.3162 147.0798 140.2261 122.5448 191.0334 52.9117 96.4099 110.3566
 171.9048 78.3864]
[ 51.7256 99.2127 95.8073 105.3173 81.8901 125.3137 128.208 78.7658
 81.1359 201.9115]

```

נלקח מדגם של 30 ציורים IM ו-30 ציורים RE. ניקח רק את השכבה הראשונה שהוא הצבע האדום, ונתקבל רמה ממוצעת של אדום לכל ציור וצייר במדגם. רמת כל צבע היא מספר בין 0 ל-255, כך שהמספר גבוה יותר כך יש שקיבלנו הגוינו, ובכל שהמספר גבוה יותר כך יש יותר מאותו צבע בציור. נחשב את ממוצע המדגם ה-IM וה-RE ומגליים שאכן בהתאם להשערה האלטרנטיבית שלנו יש יותר אדום ב-IM מאשר ב-RE! בכמה יותר? בערך 15.

```

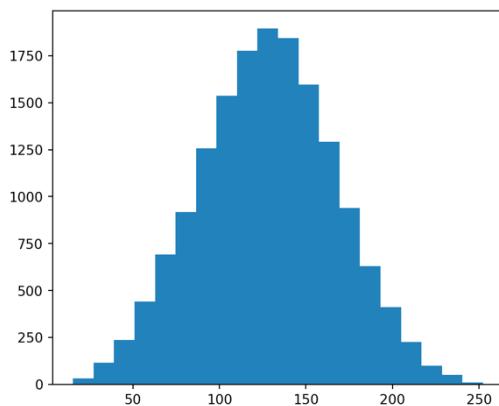
print(f'Realist paintings mean red value: {real_red.mean():.2f}')
print(f'Impressionist paintings mean red value: {impr_red.mean():.2f}')
print(f'Means difference: {impr_red.mean() - real_red.mean():.2f}')

Realist paintings mean red value: 115.47
Impressionist paintings mean red value: 130.75
Means difference: 15.28

```

אם נדגום שוב, נקבל בטוח תוצאה אחרת, ואכן כך. נדמיין שהדוגמה יקרה כמו שקרה פעמים רבות במציאות (במחקר ובתעשייה). יש לנו תקציב ל-60 ציורים בלבד. פעם אחת בלבד ניתן לדגום אותם. איך נדע שההבדל שאנו חונם רואים במדגם אחד הוא חשוב ומובהך? כמובן מעד על כך שבירות המחדל שלושני סגנוןות הציור יש אותה רמת אדום בהם – היא לא נכונה.

התפלגות האפס



איך נבנה את השערת האפס באמצעות סימולציה: תחת השערת האפס אין הבדלים בין ציורים IM ו-RE מבחינת כמות האדום בהם. ניקח את אלף התמונות שלנו ונأخذ אותן לטור **עולם מלאכותי**, אחד ויחיד, של K 16K תמונות. בעולם זהה, כל פעם שנציגים 2 קבוצות של 30 ציורים שנקרו להם IM ו-30 ציורים שנקרו להם RE – **אנחנו יודעים** שרמת האדם זהה בין שתי הקבוצות.

נחשב את רמת האדום הממוצעת של כל K 8 ציורים מאותו סוג, ונأخذ ל-K 16 של רמת אדום – `population`. אם נציג היסטוגרמה של האוכלוסייה שלנו, נראה את רמתת האדום הממוצעת שהיא די סימטרית (כך יצא במקורה), עם ערכים בין 0 ל-255. בצעת, נדגום שני מדגמים של 30 ציורים IM לבארה, ו-30 ציורים RE לבארה, ונסתכל על ההבדל ברמתת האדום שלהם (לבארה – כי הדוגמה נעשית מתוך אוכלוסייה אחת גדולה של ציורים כלשהם). נקבל מספר שונה מאפס. אנחנו יודעים שהמספר שונה מ-0 (בגלל אקראיות). אם נחזיר על הדוגמה שוב ושוב, נקבל מספרים אחרים (גם שליליים).

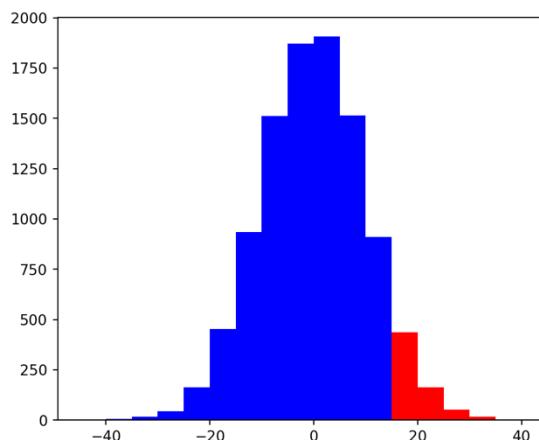
הຍינו רצים לעת כמה הערך המקורי שקיבלנו של הפרש 15 הוא חריג ביחס למדגמים באלה של אוכלוסיית האפס – שבה לא אמרו להיות הבדל ביניהם. לכן, נדגום הרבה זוגות של מדגמים מהאוכלוסייה הגדולה, נחשב את הפרש האדום ביניהם, **נסתכלו על התפלגות של הפרשים האלה, וו תהיה התפלגות האפס שלנו – התפלגות הייחוס שלנו**. נקבל 10,000 הפרשי ממוצעים, והרשימה `_mean_diffs` היא התפלגות האפס שלנו. אפילו בתפלגות זו יש הפרשים גדולים הרבה יותר מ-15 נקודות.

```

1 def sample_null_mean_diff(n = 30):
2     real_red_null = np.random.choice(population, n, replace=False)
3     impr_red_null = np.random.choice(population, n, replace=False)
4     return impr_red_null.mean() - real_red_null.mean()
5
6 null_mean_diffs = np.array([sample_null_mean_diff() for i in range(10000)])
7
8 print(f'Max null mean diff: {max(null_mean_diffs): .2f}')
9 print(f'Min null mean diff: {min(null_mean_diffs): .2f}')

```

Max null mean diff: 36.21
Min null mean diff: -37.46



p-value: האם ההפרש שלנו של 15 שתהתקבל מזוג מדגמים אמיתי של צירום MI ו-RE הוא כל כך חריג? באופן מפורש יותר, נביט בכל ההתפלגות של ההפרשים, עם היסטוגרמה, ונסמן את הערך שאנו קיבלנו של 15 נקודות הפרש. נראה ש-**15 נקודות זה לא ערך כל כך קיצוני תחת התפלגות האפס.** בשביל זה יש לנו את מדד **value-k** – הוא הסיכוי לקבל ערך קיצוני כפי שקיבלנו או קיצוני יותר תחת התפלגות האפס.

בתרשים שראינו, אפשר לחשב עלי'ו בשטח האודם בהיסטוגרמה מעל הערך שקיבלנו (15). נחשב את השטח באופן הבא: מה יחס המספרים שהם מעל 15? **סוג זה הוא one sided p-value (חד-צדדי) כי אנחנו ממעוניינים רק בתוצאות שהן קיצניות לביון שבו צירום MI הם אדומים יותר מאשר RE.** כאן הוא יוצא 0.07 בלבד 7%.

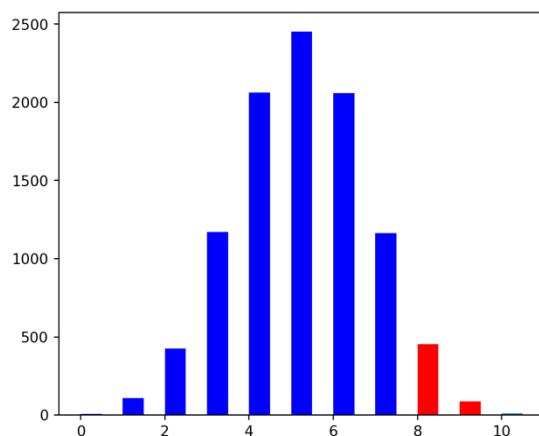
גם תחת השערת האפס שבה אין הבדל בין סגנונות הצירום ברמת האודם, אפשר בסיכוי לא רע לקבל הפרש של 15 נקודות אודם או יותר מכך בין שני מדגמים אקראים. האם זה משכנע אותנו שיש הבדל בין צירום MI ו-RE באוכולוסיה? שהם באים ממשי התפלגותות שונות? מתקבל להשוות **value-k** למספר של 1% או 5%, ותחת המבחן הזה הערך שקיבלנו לא מושגים מספיק, ולא היינו דוחים את השערת האפס. [ב庫וד אנחנו עושים ממוצע של **null_mean_diffs** רק עבור הערכים שהם גודלים מ-15, ואשיות זה הופך למערךבוליאני של **True/False** ועוד באמצעות סכימה וחולקה באורך הכלול – ממוצע – נקבל את החלק היחסני]

```
one_sided_p_value = np.mean(null_mean_diffs >= 15)
print(f'P(mean_diff >= 15 | H0) = {one_sided_p_value: .2f}')
P(mean_diff >= 15 | H0) = 0.07
```

```
two_sided_p_value = np.mean(np.abs(null_mean_diffs) > 15)
print(f'P(|mean_diff| >= 15 | H0) = {two_sided_p_value: .2f}')
P(|mean_diff| >= 15 | H0) = 0.13
```

סימטרית כמו שראינו, זה אומר להכפיל בקיוח את הערך שקיבלנו קודם פי 2, ולקבל 14% עבר **value-k**. גם כאן לא היינו דוחים את השערת האפס.

```
null_res = np.random.binomial(10, 0.5, size=10000)
pd.value_counts(null_res, normalize=True).sort_index()
```



התפלגות בינומית: נניח שנוטנים לנו מטבע ואנחנו חצאים לבודק האם הוא הוגן. נטיל את המטבע 10 פעמים, ונקבל 8 פעמים H. אפשר לסמן את השערת האפס שלנו:

$\frac{1}{2} = P[\text{head}] = P[H_0]$ ולשאול האם המדגם שלנו הוא עדות מסיפה כדי לדוחות את H_0 . באשר תהיה לנו התפלגות האפס בשמטבע הוגן, נוכל לראות באיזה אחוז מהמקרים אנחנו מקבלים תוצאה קיצונית כמו 8 פעמים עצ או יותר.

דרך ראשונה - נסמלץ! משתנה ביןומי עם 10 הטלות, בסיכוי שווה חצי, 10,000 פעמים. התוצאה 8 התקבלה למשל ב-4% מהמקרים. אפילו תוצאה כמו 10 פעמים H התקבלה האחוז מסוים של פעמים. נזכיר שוב היסטוגרמה של התפלגות. במקרה שלנו יש רק תוצאה אפשרית אחת מ-0 עד 10. התוצאה 8 לא נראית כל כך קיצונית. אם אכן נחיש **value-k**, השכיחות של תוצאות קיצניות כמו 8 או יותר, נקבל גם במקרה החוד-צדדי וגם במקרה הדו-צדדי ערך שלא עומד ב מבחן הסף של 5%. זה פשוט משווה שיקפה אחת ל-10 פעמים גם בשמטבע הוגן.

דרך שנייה - אם כבר מניחים שההתפלגות של תוצאות הניסוי שלנו תחת השערת האפס היא ביןומית - אפשר להשתמש בחוקי הסתברות הבינומית כדי לחשב התפלגות מדויקת. X יהיה מספר תוצאות H ב-10 הטלות, ככלומר $\binom{10}{2} \cdot P[X=8] = 0.055$. הנוסחה שמחשבת הסתברות מצטברת של משתנה ביןומי.

$$\text{במקרה החוד-צדדי נבדק } P[X \geq 8] + P[X \leq 2] = 2 \cdot P[X \geq 8] = 0.11$$

לקח חישוב: סימולציה טובה, אמורה לתת תוצאות בהתאם לנition מתמטי. אחרת, יש טעות באחד מהם.

```
import scipy.stats as stats
res = 1 - stats.binom.cdf(k=7, n=10, p=0.5)
np.round(res, 3)
```

0.055



טעויות מסדר ראשון ושני

בשיטות בדיקת השערות אפשר לחשב על שתי סוגים טעויות:

- .1 Type 1: השערת האפס היא נכונה, וכן נדחה אותה.
- .2 Type 2: הינו צירcisム לדוחות את השערת האפס, ולא נדחה אותה.

ונסה להקטין את הסיכוי לעשות אותן. נכתב גם את ההשערה האלטרנטיבית בצורה מעט יותר פורמלית. בדוגמה בית המשפט: $H_0 = \text{innocent}$ השערת האפס. $H_1 = \text{guilty}$ ההשערה האלטרנטיבית. הסיכוי ל- H_1 גבוה יותר. בדוגמה הציורים: $H_0 = \mu = 0.8$ השערת האפס היא שההפרש בצלב אדום בין ציורים MI ו-RE הוא 0, והשערה אלטרנטיבית תהיה $\mu_1 = 0.2$.

גדר פורמלית את סוגים טעויות:

- .1 Type 1: לדוחות את H_0 כנכון, זו טעות מסוג ראשון ומסומנת [$P[\text{reject } H_0 | H_0 \text{ true}] < \alpha$].
- .2 Type 2: לא לדוחות את H_0 בשczריך, זו טעות מסוג שני ומסומנת [$P[\text{not reject } H_0 | H_1 \text{ true}] < \beta$].

ניתן לסכם זאת בטבלה הבאה:

Reality\Decision	Not Reject H_0	Reject H_0
H_0	Confidence: $1 - \alpha$	Type I Error: α
H_1	Type II Error: β	Power: $1 - \beta$

עוצמת המבחן (power) היא כמות חשובה מאוד, והינו רצים שתהייה כמה שיותר גדולה. היא מסומנת לרוב בתור π והוא הסיכוי לדוחות את H_0 כשי- H_1 נכון – כשבולם – באמת מתקיימת תופעה חדשה (חליקן חדש, הנאשם אשם בפשע).

גישה לבדיקת השערות:

1. חישוב $\text{value}-k$ והשווואה למספר מסוים α (רמת מובהקות)

a. אם $\alpha \leq k$ אז נדחה את H_0

b. אם $\alpha > k$ אז לא נדחה את H_0

2. **אזור דחיה (rejection area):** נקבע את ערך α , הסיכוי לטעות מהסוג הראשון, ואם אנחנו יודעים מראש איך מתפלג יזשאו מודד סטטיסטי של המדגם שלנו (X) למשל המוצע, נוכל לחץ ממנו מה הערך הקרייטי המקסימלי שהוא יכול להיות כדי לשמר על α , יזשאו ערך C. בעת ביצוע את הניסוי ונשווה:

a. אם $C \geq T(X)$ אז נדחה את H_0

b. אם $C < T(X)$ אז לא נדחה את H_0

נחזיר לדוגמת המבחן: עשינו בדיקת השערות באמצעות סימולציה ובאמצעות חישוב $\text{value}-k$. בעת נקבע אזור דחיה. מודל סביר לו- X הוא התפלגות בינומית ולכן (p, n). תחת H_0 נאמר כי $0.5 = p$ ככלומר המוצע הוגן. הסטטיסטי היבי נוח להסתכל עליו הוא X עצמו.

נקבע את הסיכוי לטעות מסוג ראשון $0.01 = \alpha$: ככלומר אנחנו מוכנים שאם נחזיר על הניסוי הרבה פעמים, גם אם המוצע הוגן, אם נבצע את המבחן שלו יש סיכוי של 1-100 שנטענה ונדחה את השערת האפס. מתוך העבודה שקבענו את α על 1%, נראה מהו הערך הקרייטי תחת השערת האפס של המוצע הוגן – שמספר ה- H שהתקבלו ב-10 בהטלות – שאם נקבל אותו או יותר נאמר שנפלנו באזור הדחיה (ולכן נדחה את H_0). אז C שלנו הוא:

$$\alpha = 0.01 \approx P[X \geq C] \Rightarrow C = 9$$

ככלומר אם נקבל בניסוי 9 פעמים H ומעלה – נדחה את H_0 , ואם פחות – לא נדחה. לא השתמשנו בכך בהשערה אלטרנטיבית.

דוגמת הציורים (לפי סימולציה): את הגישה של אזור דחיה אפשר להחיל גם על הסימולציה של הציורים. לא הייתה לנו התפלגות מדויקת אבל התפלגות מסוימת, באובייקט `mean_diffs_llawt`, שנוצר בשdagmeno הרובה הפרשי ממוצעים תחת השערת האפס.

המשמעות של קביעה $0.01 = \alpha$ וממנה לחץ את אזור הדחיה – היא לשאל מהו האחוזון-99 של התפלגות זו (מהו הערך הכל-בר גובה שמעליו) אפילו אם השערת האפס נכון, הסיכוי לראות הבדל כל כך גדול בין הציורים השניים המדגים שלנו הוא לפחות 1%. כאן הערך זהה יוצא $C = 23$, ונחנו קיבלנו 15. אז לא נדחה את השערת האפס.



מתי נתיחס לשערת האלטרנטיבית?

כאשר אנחנו רצינו לעשות חישובי עוצמה (power) כדי שהסיכוי שלנו לדוחות את H_0 אם באמת צריך יהיה כמה שיוצר גדול. בኒוסי המetu, נניח שמשיחנו אומר לנו שהסיכוי ל- H הוא 0.8, אז נסמן $p = H_1$. בעת אנחנו יכולים לחשב את העוצמה הסטטיסטי:

$$\text{Power} = P[\text{reject } H_0 \mid H_1 \text{ true}] = P_{H_1}[X \geq 9] = 0.376$$

בולם חישבנו את הסיכוי לקבל לפחות 9 או יותר H (אזרך הדחיה שמצאננו קודם, $C = 9$) תחת התפלגות ביןומית עם סיכוי 0.8: ($n, 0.8 \sim H_1$). קיבלנו סיכוי של 38% בערך. המשמעות היא שאם נעורק את המבחן זהה הרבה פעמים, גם אם החשד שלנו נכון והetu איננו הוגן (הסיכוי לקבל H הוא 0.8), בפחות מ-40% מה מבחנים נגיע למסקנה הנכונה – שהetu איננו הוגן. וזה לא עוצמה גדולה בכלל.

QUIZ 7

שאלה 1: עלית מפרסמת שחופיסת שוקולד פרה שוקלת 100 גרם בממוצע, בהתפלגות ידועה. אלון חושב שהיא שוקלת 90 גרם בממוצע, באותה התפלגות. הוא דוגם באקראי 100 חփיסות ומציע מבחן סטטיסטי לפי כל ההנחה הנדרשת, מחשב P -value חד-צדדי כמו צריך, ה- e-value - k קטן מ-0.8 ו-אלון דוחה את השערת האפס.

אילו מהמספרים הבאים היה משתנה אם היה מתברר בעצם שאלון חושב שחופיסה שוקלת 80 גרם (כל נתון אחר זהה)?

- **הסיכוי לטעות מסווג שני** – כאשר אלון משנה את ההשערה החלופית מ-90 ל-80, הסיכוי לטעות מסווג שני משתנה. אם ההבדל בין ההשערה החלופית החדשה (80) לבין השערת האפס (100) גדול יותר – הסיכוי לאטאר את ההבדל הזה גדול, והסיכוי לטעות מסווג שני קטן.
- **מבחן ביחס לשערת האפס** ולא להשערה החלופית, שכן הוא לא ישתנה.
- **הסיכוי לטעות מסווג ראשון** – שינוי ההשערה החלופית לא משפיע על הסיכוי הזה, כי הוא מוגדר מראש ונקבע על ידי הספק **של ה- e-value - k** , שכן הוא לא ישתנה.
- **איזור הדחיה** – נקבע על פי הספק של הסיכוי לטעות מסווג ראשון, ועל פי התפלגות הנתונים תחת השערת האפס. אין קשר שובי להשערה החלופית, שכן הוא לא ישתנה.

שאלה 2: בשיעור הוצגו שתי גישות של בדיקת השערות: באמצעות סימולציה ובאמצעות הנחתת מודל והתפלגות. אם ניתן לבחור תמיד עדיף להניח מודל והתפלגות ולא להיעזר בסימולציה.
לא נכון, הנחה היא הנחה והוא עלולה להיות לא נכונה, ניתן לחשב על מקרים רבים שambilן באמצעות סימולציה יהיה בעל עוצמה גבוהה יותר ממבחן בעל הנחות לא נכונות, ולכן עדיף.

שאלה 3: בשיעור הוצגו שתי דרכים לבדיקת השערת האפס. חישוב P -value- k ושוואתו לרמת מובהקות אלפא, וחישוב ערך קרייטי והשוואת סטטיסטי המדגם אליו ("איזור הדחיה"). בר או בר, לא יכול להיות שבדרכ אחות נחליט לדוחות את השערת האפס ובדרך השניה נחליט לא לדוחות את השערת האפס.

נכון, שתי הדרכים שקולות, בהנחה שהambilן מתבצע כפי שצריך ניתן לומר ש- P -value- k יותר אינפורטטיבי כי הוא לא החלטה בינהית, אבל לא יכול להיות מצב שיביא להחלטות שונות, שימושו לב שערך הקרייטי נגזר מרמת המובהקות.



תרגול 4 – הסקה סטטיסטית א'

טעויות מסדר ראשון ושני – שאלות (חישוב β , α , מציאת C)

שאלה 1: קרא טוענת כי "גברים לא יכולים להבדיל בין קולה וופפסי". המנכ"ל של קולה: "אני הולך לתת ל-100 אנשים שתי כוסות של קולה או פפסי, מה ישכנן אותו מהם יכולים?". קרא: "אפיו 60 אנשים שייזהו נכון ישכננו אותו". נניח שהגברים מחליפים בזרה ב"ת ואכן קיימים סיבויים לכך את המשקה. מהי שגיאת Type-1 של המבחן של קרא? (בלומר α)?

- נשים לב כי משאלו קבוע לנו מה הערך הקרייטי (C)我们知道計算的統計量和判斷標準。 α .
- נגדיר את X להיות מספר האנשים שצדוקים בנסיבות זהה, אז $X \sim Binom(100, p = 0.5)$ ונגדיר $H_0: p = 0.5$ וההשערה האלטרנטיבית (חד-צדדי) היא $H_1: p > 0.5$. נגדיר את הסטטיסטי להיות $X = T(X)$.
- אנחנו דוחים את H_0 אם $60 > X$. נחשב:

$$\begin{aligned}\alpha &= P[\text{reject } H_0 | H_0 \text{ true}] = P_{H_0}[X \geq 60] \\ &= 1 - P_{H_0}[X \leq 59] \\ &= 1 - \sum_{k=0}^{59} P[X = k]\end{aligned}$$

אפשר לחשב ידנית, או להשתמש במודול `scipy.stats`. אפשר גם לבצע **סימולציה** של

ניסויים: כל פעם דוגמים 10,000 אנשים. נבדוק מתי התוצאה גדולה שווה 60. קיבלנו כי $\alpha = 0.029 = 2.9\%$.

שאלה 2: מהי טעות Type-2 של קרא אם אנשים יכולים אכן לה辨別 בין קולה לפפסי עם הסתברות $p = 0.8$?

- נגדיר את X הפעם תחת $H_1: X \sim Binom(100, 0.8)$.
- אנחנו לא דוחים את H_0 אם $60 < X$ למרות שהסיכוי הוא 0.8. נחשב:

$$\beta = P[\text{not reject } H_0 | H_1 \text{ true}] = P_{H_1}[X < 60] = \sum_{k=0}^{59} P[X = k]$$

נקל ב- $\beta = \alpha$, וזה אומר שהמבחן של קרא טוב. α קטן מ-3% והטעות מסדר שני פרקטית 0.

- מה היה הופך את המבחן הזה ללא טוב? להקטין α מואוד את גודל המדגם למשל. אבל מה שמאוד נכון פה זה הנતון של 0.8. אם זה היה אחרת היינו מקבלים תוצאות שונות.

שאלה 3: מניחים כי שזמנן המסתנה לקו 25 מתפלג $\left(\frac{1}{20}\right) Exp$ – כלומר מחכים בממוצע 20 דקות לקו הבא. דני ואני לא תפסו את האוטובוס, ודני מתעכבר: "נכחבה לאוטובוס הבא, **ואם זה יהיה מעל 30 דקות** תהיה לי הוכחה שחברת האוטובוסים משקרת". נפרמל זאת בבדיקה השערות ונחשב את הטעות Type-1 (α – רמת המובהקות).

- זה הוא זמן המסתנה לאוטובוס והוא מתפלג (λ). $H_1: \lambda < \frac{1}{20}$ ו- $H_0: \lambda = \frac{1}{20}$. נגדיר $Y \sim Exp(\lambda)$. הסטטיסטי שלנו יהיה שוב המשטנה עצמו – **מבחן עבור אוטובוס אחד**: Y_1 . **הערך הקרייטי** נקבע $C = 30$.
- נזכור כי $\lambda = \frac{1}{\mu}$ היא התוחלת של השערת האפס.
- אנחנו דוחים את H_0 אם $Y_1 > C$. נחשב:

$$\alpha = P[\text{reject } H_0 | H_0 \text{ true}] = P_{H_0}[Y_1 > 30] = 1 - P_{H_0}[Y_1 < 30] = e^{-\frac{1}{20} \cdot 30} = 0.22$$

- קיבliśmy 22% סיכוי לטעות. יש סיכוי של 1-5 שזה יקרה, תחת התפלגות אקספוננציאלית לקבל 30 דקות ומעלה בשזמנן הממוצע הוא 20 דקות ומעלה, זה סביר שזה יקרה תחת המודל.

```
from scipy.stats import expon
lamb = 1 / 20
1 - expon.cdf(30, scale=1 / lamb)
```

0.22313016014842982

שאלה 4: מהי עצמת המבחן של דני?

- אין לנו את H_1 !
לא הוספנו כאן שום מידע. מהו הזמן באמת? 21 דקות או חצי שעה? צריך לדעת מהו העולם האמיתי זהה.
ולכן אי אפשר לחשב את β .

שאלה 5: מגיעה האמת – בממוצע ממכבים 23 דקות לקו האוטובוס עם אותה התפלגות אקספוננציאלית. מהי עצמת המבחן?

- בעת נוכן לחשב:

$$\pi = P[\text{reject } H_0 | H_1 \text{ true}] = P_{H_1}[Y_1 > 30] = e^{-\frac{1}{23} \cdot 30} = 0.27$$

- בלומר 27% היא עצמת המבחן. זה מספר יחסית קטן, למרות שדני צודק בגadol. האוטובוסים באים בממוצע 23 דקות ולא 20 דקות וחברת האוטובוסים משקרת. הסיכוי שנגלה את זה לפי אוטובוס אחד, רק ב-27% מההmarker הניסוי באמת יראה את זה.

שאלה 6: כדי שדני תקבל לפחות 80% עצמה, מה צריך להיות הזמן הממוצע האמיתי?

- די גדול – כדי שב-80% מההmarkerים שאנו מכבים מסתכלים על אוטובוס אחד הניסוי שלנו יראה את זה.
נמצא את ה- λ שעונה על תנאי הבעה:

```
1 / (-np.log(0.8) / 30)
134.44260353173652
1 - expon.cdf(30, scale=134.45)
0.8000098206579453
```

$$\begin{aligned}\pi &= P[\text{reject } H_0 | H_1 \text{ true}] = P_{H_1}[Y_1 > 30] = e^{-\frac{1}{\lambda} \cdot 30} \geq 0.80 \\ \Leftrightarrow \lambda &= \frac{1}{-\log(0.8)/30} = 135\end{aligned}$$

ניתן לבצע power analysis: אנחנו רואים בשרטוט את $e^{-\frac{30}{\lambda}}$.

```
import matplotlib.pyplot as plt
def power(inv_lamb, critical_value=30):
    return 1 - expon.cdf(critical_value, scale=inv_lamb)
inv_lambdas = np.arange(1, 200)
powers = [power(inv_lamb) for inv_lamb in inv_lambdas]
plt.plot(inv_lambdas, powers)
plt.axline((0, 0.8), slope=0, color='r', linestyle='--')
plt.axvline(x=135, color='r', linestyle='--')
plt.show()
```

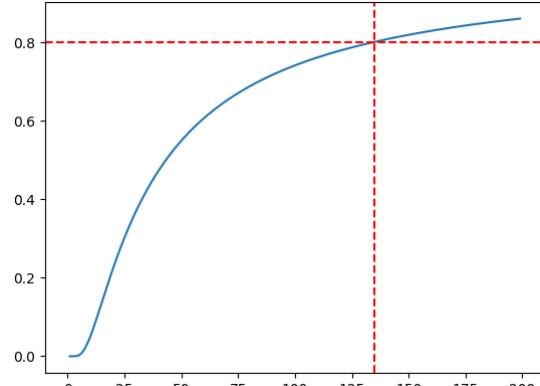
You should ace these:

$$Y \sim N(\mu, \sigma^2)$$

- $P(Y \leq y) = ?$
- $P(Y > y) = ?$
- $P(y_1 < Y < y_2) = ?$
- $P(|Y| > y) = ?$
- $P(Y = y) = ?$

Through $\Phi(a) = P(Z \leq a) = 1 - P(Z > a)$, where $Z \sim N(0, 1)$:

$$P(Y \leq y) = P\left(\frac{Y-\mu}{\sigma} \leq \frac{y-\mu}{\sigma}\right) = P\left(Z \leq \frac{y-\mu}{\sigma}\right) = \Phi\left(\frac{y-\mu}{\sigma}\right)$$



מכוררת קטנה – חישוב הסתברות של התפלגות נורמלית:

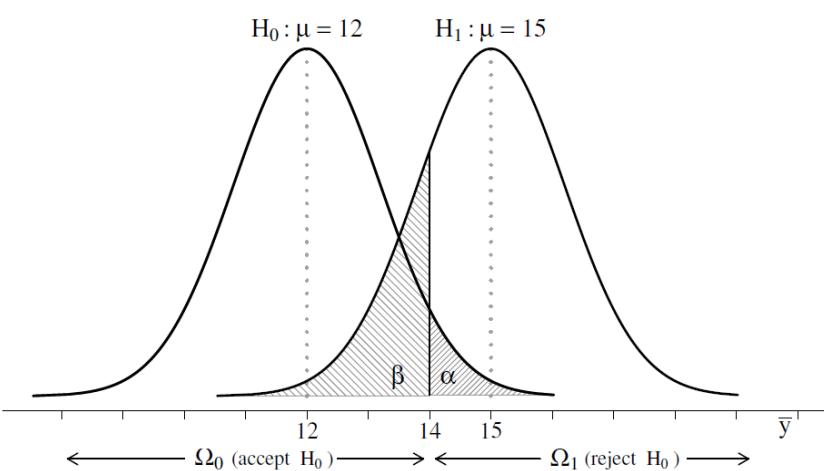
```
from scipy.stats import norm
print(f'\Phi(1.960) = {norm.cdf(1.96, loc=0, scale=1) :.3f}')
print(f'Z_97.5 = {norm.ppf(0.975, loc=0, scale=1) :.3f}')
print(f'P(Y < 2) if Y ~ N(0, 2^2) = {norm.cdf(2, loc=0, scale=2) :.3f}')
print(f'P(|Y| < 2) if Y ~ N(0, 2^2) = '\
      f'{norm.cdf(2, loc=0, scale=2) - norm.cdf(-2, loc=0, scale=2) :.3f}')
print(f'P(|Y| < 3) if Y ~ N(0, 3^2) = '\
      f'{norm.cdf(3, loc=0, scale=3) - norm.cdf(-3, loc=0, scale=3) :.3f}, coincidence?)'
```

```
\Phi(1.960) = 0.975
Z_97.5 = 1.960
P(Y < 2) if Y ~ N(0, 2^2) = 0.841
P(|Y| < 2) if Y ~ N(0, 2^2) = 0.683
P(|Y| < 3) if Y ~ N(0, 3^2) = 0.683, coincidence?
```

שאלה 7: חברת מכוניות מציגה מודל חדש של מכונית ומספרת שהיא יותר חסכונית וצורכת פחוט דלק מכל מכונית אחרת אותה. בפרט, החברה טוענת שהמכונית החדשה בעלת ביצועים של 15 קילומטר לליטר על ביש מהיר, בעוד שהמתחרים רק 12. נרצה לבדוק את אמינות הטענה באמצעות **מבחן רנדומלי של 5 מכוניות חדשות של החברה.** **נניח התפלגות נורמלית עם סטיית תקן ידועה 2.** יהי Y_1, \dots, Y_5 צירויות הדלק התואמות. **מהו הערך הקרייטי של \bar{Y} (המוצג) שמעליו יש לדחות את H_0 עם רמת מובהקות של $\alpha = 0.01$?**

- נניח כי $(Y_i, \mu = 12, \sigma^2 = 2^2)$ נורמלית. הסטטיסטי שלנו יהיה המוצע: $\bar{Y} = \frac{1}{5} \sum Y_i$.
- אנחנו נדחה את H_0 אם $C > \bar{Y}$ בלבד. מתקיים כי $\bar{Y} \sim N\left(\mu = 12, \sigma^2 = \frac{4}{5}\right)$ [ממוצע של משתנים נורמליים מתפלג עם השונות המקורית חלקי N – כמות המשתנים].
- נקבע את $0.01 = \alpha$. מהו הערך הקרייטי C כדי לדחות את השערת האפס? נחשב ונחלץ את C (באמצעות חישוב אחודן):

$$\begin{aligned} \alpha &= P[\text{reject } H_0 \mid H_0 \text{ true}] = P_{\bar{Y} \sim H_0}(\bar{Y} \geq C) = P_{\mu=12}(\bar{Y} \geq C) = 1 - \Phi\left(\frac{C - 12}{\sqrt{0.8}}\right) = 0.01 \\ &\Leftrightarrow \Phi\left(\frac{C - 12}{\sqrt{0.8}}\right) = 0.99 \Leftrightarrow Z_{99} = \frac{C - 12}{\sqrt{0.8}} \Leftrightarrow C = 12 + Z_{99}\sqrt{0.8} = 14.08 \end{aligned}$$



- בזרה יותר מפורטת:
 - תחת $\mu = 12$: H_0 , מצאנו כי $C = 14$, כלומר נדחה את השערת האפס אם המוצע יהיה לפחות 14. מה ש踔רי הערך הקרייטי הוא α .
 - תחת $\mu = 15$: H_1 , בגלל זה צריך מעל 14. α . אפשר לחשב את β , הסיכוי לקבל 14 ומטה. משם אפשר לחשב גם את העוצמה.

```
norm.ppf(0.99, loc=12, scale=np.sqrt(0.8))
14.080748794266976
```

שאלות נוספת:

1. **איזה מידע נוסף נדרש כדי לחשב את עוצמת המבחן?**

אנחנו צריכים לדעת את H_1 האמיתית! הנחנו שהאפס $\mu = 12$ במו שהחברה טוענת, אבל בשetail לחשב נניח לרוגע שההיפואזיס $\mu = 14.5$, ככלומר הביצועים הם 14.5 ק"מ לליטר והם שיקו קצר.

2. **מהי עוצמת המבחן?**

$\pi = P[\text{reject } H_0 \mid H_1 \text{ true}] = P_{\bar{Y} \sim H_1}(\bar{Y} \geq 14.08) = P_{\mu=14.5}(\bar{Y} \geq 14.08) = 1 - \Phi\left(\frac{14.08 - 14.5}{\sqrt{0.8}}\right) = 0.68$
כלומר יש סיכוי של 68% לדחות את השערת האפס.

3. **מהי הטעות מסדר ראשון?**

$\alpha = P[\text{reject } H_0 \mid H_0 \text{ true}] = P_{\bar{Y} \sim H_0}(\bar{Y} \geq 14.08) = 1 - \Phi\left(\frac{14.08 - 12}{\sqrt{0.8}}\right) = 0.01$

זה לא מפתיע כי חישבנו את הערך הקרייטי ($C = 14.08$) בהתבסס על $\alpha = 0.01$.

4. **מהי הטעות מסדר שני?**

$\beta = P[\text{not reject } H_0 \mid H_1 \text{ true}] = P_{\bar{Y} \sim H_1}(\bar{Y} \leq 14.08) = \Phi\left(\frac{14.08 - 15}{\sqrt{0.8}}\right) = 0.15$

כאן נחזור על ההנחה שלנו כי $\mu = 15$: אנחנו לא מחשבים את עוצמת המבחן ולא צריכים לדעת את H_1 האמיתית!

נשים לב במקרה זה כי אפשר לומר לנו $\pi = 1 - \beta = 0.85$. למה זה לא מסתדר עם סעיף 2? כי שם הנחמנו $\mu = 14.5$. אם נחשב

מחדש כאשר $\mu = 15$ נקבל: $1 - \Phi\left(\frac{14.08 - 15}{\sqrt{0.8}}\right) = 0.85$

5. מה גודל המדגם המינימלי N שנדרש כדי להשיג עצמת מבחן של 80%?

$$\pi = P[\text{reject } H_0 | H_1 \text{ true}] = P_{\mu=14.5}[\bar{Y} \geq 14.08] = 1 - \Phi\left(\frac{14.08 - 14.5}{\sqrt{\frac{4}{N}}}\right) = 0.80$$

$$\Leftrightarrow \Phi\left(\frac{14.08 - 14.5}{\sqrt{\frac{4}{N}}}\right) = 0.2 \Leftrightarrow \frac{14.08 - 14.5}{\sqrt{\frac{4}{N}}} = -0.84 \Leftrightarrow N \approx 16.01$$

6. אם הממוצע יתגלה להיות $\bar{Y} = 13$, תחת איזה α נדחה את השערת האפס?

$$\alpha = P[\text{reject } H_0 | H_0 \text{ true}] = P_{\bar{Y} \sim H_0}[\bar{Y} \geq 13] = 1 - \Phi\left(\frac{13 - 12}{\sqrt{0.8}}\right) = 0.131$$

7. מה משפיע על עצמת המבחן?

1. Sample Size (n)

- a. Effect: Increasing the sample size generally increases the power of the test.
- b. Reason: A larger sample size reduces the standard error of the mean, leading to a narrower confidence interval. This makes it easier to detect a true effect, thereby increasing the power.

2. Significance Level (α)

- a. Effect: Increasing the significance level (e.g., from 0.01 to 0.05) increases the power of the test
- b. Reason: A higher α makes it easier to reject the null hypothesis, which increases the likelihood of detecting a true effect. However, this also increases the risk of Type-I error (false positives).

3. Effect Size ($\mu_1 - \mu_0$)

- a. Effect: A larger effect size increases the power of the test.
- b. Reason: The effect size is the difference between the true mean (μ_1) under the alternative hypothesis and the hypothesized mean (μ_0) under the null hypothesis. Larger differences make it easier to distinguish between the null and alternative hypotheses, increasing power.

4. Population Variability (σ)

- a. Effect: Lower population variability increases the power of the test.
- b. Reason: Lower variability (smaller σ) reduces the standard error of the mean, making it easier to detect a true effect. High variability makes it harder to distinguish between the null and alternative hypotheses, thereby decreasing power.

5. Test Type (One-Tailed vs. Two-Tailed)

- a. Effect: A one-tailed test generally has more power than a two-tailed test, assuming the direction of the effect is correctly specified.
- b. Reason: A one-tailed test focuses on detecting an effect in one direction, concentrating the critical region in that direction, which increases the likelihood of rejecting the null hypothesis if the effect exists.

6. True Mean (μ_1)

- a. Effect: The further the true mean (μ_1) is from the null hypothesis mean (μ_0), the higher the power of the test.
- b. Reason: When the true mean is significantly different from the hypothesized mean, the test is more likely to detect this difference, thus increasing power.

Summary: To increase the power of a test: Consider increasing the **sample size**, choosing a higher **significance level** (with caution regarding Type-I error), ensuring that the test is designed to detect a sufficiently large effect size, and **minimizing variability** if possible. Additionally, using a one-tailed test can increase power if you have a strong rationale for the direction of the effect.

שאלה חשובה: נתון $n=10$. האם לוחות את $H_0: \mu = 12, H_1: \mu > 12$, $\bar{Y} = 13$. האם צריך את H_0 ? האם צריך עוד מידע? לא לדוחות את השערת האפס! ההשערה היא חד-צדדית לימין, וקיים משווה מושלם, לא ציר סטטיסטי. זו אחת הביעתיות בשימושים השערת חד-צדדית. אם ממוצע המדגם הוא בניגוד לכיוון ההשערה – אין על מה לדבר יותר.



p-value – שאלות (לא לפי ערך קרייטי C):

הערה: אם דוחים את H_0 , זה לא אומר שההבדל שמצאנו הוא משמעותי! יכול להיות שמצאנו הבדל בזמן תגובה של 0.0001 מיילישיות זהה מובהק, אבל יגיע מזמן פסיבולוגית קוגנטיבית ויטען שהוא לא משמעותי בכלל.

שאלה 1: בניסוי הקולה והפפסי של קרן: מהו ה-p-value של ה- t ? והאם קרן נדחה את H_0 עם רמת מובהקות של 5%?

- אין מספיק נתונים! צריך שמיישחו יעשה את הניסוי, אף אחד לא שאל את 100 האנשים. p-value – זה הסיכוי לקבל תוצאה שהתקבלה, או קיצונית יותר. אבל לא התקבלה שום תוצאה!

שאלה 2: עשו את המבחן: 59 אנשים ידעו להבחין בין קולה לפפסי. גענה שוב על 1.

- נחזיר להגדירה מקודם: ($p = 0.5, H_0: p > 0.5, H_1: p < 0.5$) באשר $X \sim Binom(100, p)$. נגדיר X . נחשב:

$$P - value = P_{H_0}[T(X) > t_{obs}] = P_{H_0}[X \geq 59] = 1 - P_{H_0}[X \leq 58] = 1 - \sum_{k=0}^{58} P[X = k] = 0.04$$

- כלומר קיבלנו פחות מ-5%, ונדחה את H_0 ויחסית לסוף שנקבע בשאלה.

שאלה 3: הניסוי של דני וקוו 25: האוטובוס הגיע אחרי 35 דקות. מה ה-p-value? האם נדחה את H_0 ברמת מובהקות 5%?

- נחזיר להגדירה מקודם: ($\lambda = \frac{1}{20}, H_0: \lambda = \frac{1}{20}, H_1: \lambda < \frac{1}{20}$) באשר $Y \sim Exp(\lambda)$. נגדיר Y . נחשב:

$$P - value = P_{H_0}[T(Y) > t_{obs}] = P_{H_0}[Y > 35] = e^{-\frac{35}{20}} = 0.17$$

- כלומר קיבלנו 17% (שהה ממש גדול כי הניסוי לא טוב), لكن לא נדחה את H_0 .

Reject H_0 if: $\bar{Y} > C = 14$ (does it matter???)

$$\bar{Y} \sim N(\mu, \frac{4}{3})$$

$$P - value = P_{H_0}(T(Y) > t_{obs}) = P_{H_0}(\bar{Y} > 14.5) = 1 - \Phi\left(\frac{14.5 - 12}{\sqrt{0.8}}\right) = 1 - \Phi(2.8)$$

שאלה 4: בניסוי של המבוןיות החדרשות: התקבל 14.5 וויצא משאו שקטן מ-1% (0.0025) לנוכח.

שאלה 5: עבשו נס同胞 על השערת דו-צדדית: בהינתן המידע הקודם – מה הוא p-value והאם נדחה את השערת האפס?

$$\begin{aligned} P - value &= P_{H_0}[|T(Y)| > t_{obs}] \\ &= P_{H_0}[\bar{Y} > 14.5, \bar{Y} < 9.5] \\ &= 2 \cdot \left[1 - \Phi\left(\frac{14.5 - 12}{\sqrt{0.8}}\right)\right] = 0.005 \end{aligned}$$

- בגלל שההתפלגות הנורמלית סימטרית, אפשר פשוט להכפיל ב-2 את הערך הקודם.
- אפילו ש- α הוא 1%, אנחנו בן נדחה את השערת האפס, עם הערך **0.005**.

שאלה 6: גובה של גברים בעולם מתפלג נורמלית עם ממוצע 175 וסטיית תקן 10. רומי ומיכל מסכימות שגברים בהולנד הרבה גבויים יותר. רומי: "חושבת שהמוצע בהולנד הוא 180", מיכל: "חושבת שהמוצע בהולנד הוא 185". הן דוגמאות טרגדיה ומחילתו לדוחות את $\mu = 175$: $H_0: \mu = 175$ אם המוצע \bar{Y} גדול מ- $C = 175$. בלהה. נגדיר את ערכי ה-p-value בהתחמה: M, k, q , נגדיר את עצמות המבחן שלهن: π_M, π_R, π . **עבור ערכי ה-p מתקיים:** $p_R = M$: זאת כי זה מחושב אל מול השערת האפס, שהיא זהה אצל שתיהן. אין חשיבות להשערה החלופית כאן. עבור עצמות המבחן מתקיים: $\pi > M$ כי הפונקציה Φ מונוטונית עולה:

$$\begin{aligned} \pi_M &= P[reject H_0 | H_{1M} \text{ true}] = P_{\mu_M=185}[\bar{Y} \geq 175] = 1 - \Phi\left(\frac{175 - 185}{\frac{2}{\sqrt{n}}}\right) > 1 - \Phi\left(\frac{175 - 180}{\frac{2}{\sqrt{n}}}\right) \\ &\Leftrightarrow \Phi(-5\sqrt{n}) < \Phi\left(-\frac{5\sqrt{n}}{2}\right) \end{aligned}$$

הסקה סטטיסטית – חלק ב'

התפלגות נורמלית

חשיבות: התפלגות זו היא מהותית בנושא בדיקת השערות. נזכור שבמקרים בדים התפלגות היא של הסתברויות (ערבים בדים) ומסומנת PMF. עברו משתנים רציפים לדברים על פונקציית צפיפות, המסומנת PDF.

עבור (σ^2, μ) $X \sim N$ נתאר את פונקציית הצפיפות שלו:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

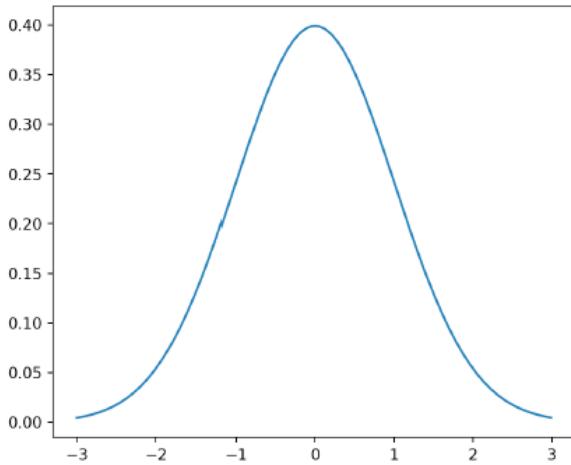
מקרה פרטי של התפלגות נורמלית שמעניין אותנו, הוא כאשר התוחלת $0 = \mu$ והשונות $1 = \sigma$ ואז נקבל את התפלגות הנורמלית הסטנדרטית ($N(0,1)$):

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

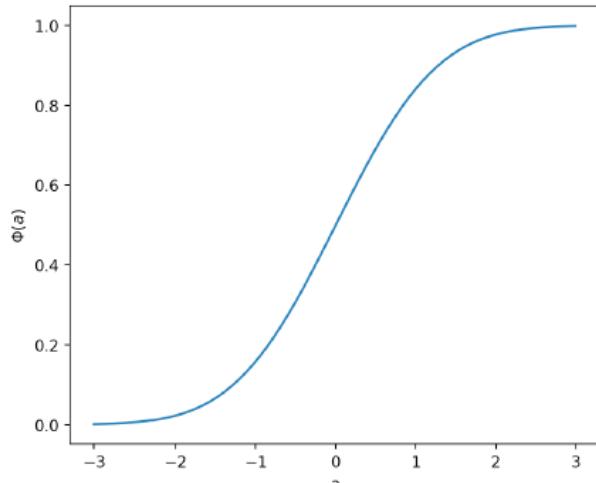
בשכזר את הצפיפות נקבל את עקומת הפעמון המפורסמת. כאן אנחנו משתמשים במודול **stats.norm** ובمتודה **pdf** כדי לצייר את פונקציית הצפיפות. ניתן לראות שהמרכז שלה סביב 0 (התוחלת), ויש לה אסימפטוטה מצד ימין ומצד שמאל של 0. כמו כן, נזכיר כי פונקציית צפיפות היא תמיד ממשית, חיובית, והשיטה בינה לבן ציר ה- x צריך להסתכם ב-1.

נזכיר כי מדובר בצפיפות ולא בהסתברות! כדי לחץ הסתברות צריך לחשב את השטח תחתיה, בלחומר לבצע אינטגרציה. לדוגמה: $\int_{-\infty}^a f(x) dx = P[X \leq a]$. האינטגרל הזה ידוע גם כפונקציית ההסתברות המצטברת (CDF). במקרה של התפלגות נורמלית אין לו פתרון סגור, רק מקובל. לכן נסמן $[a] = P[X \sim N(0,1) \leq a] = \Phi(a)$. כדי לקבל את Φ נשימוש בפונקציה **cdf** של **stats.norm**.

```
import scipy.stats as stats
x = np.arange(-3, 3, 0.01)
plt.plot(x, stats.norm.pdf(x, 0, 1))
plt.show()
```



```
plt.plot(x, stats.norm.cdf(x, 0, 1))
plt.xlabel('a')
plt.ylabel('$\Phi(a)$')
plt.show()
```



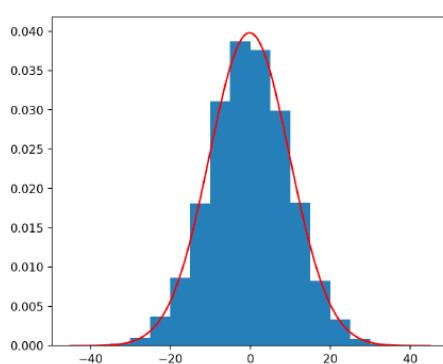
איך מקבלים מכל משתנה נורמלי (σ^2, μ) $X \sim N$ את התפלגות הנורמלית הסטנדרטית? ע"י תקנון (סטנדרטיזציה), מחסרים מ- X את התוחלת שלו, ומחלקים בסטיית התקן. משתנה זה מסומן: $Z = \frac{X-\mu}{\sigma} \sim N(0,1)$.

התוחלת והשונות: שני הפרמטרים האלה מספקים בשילוב להגדר באופן חד משמעית את התפלגות. אפשר לראות את זה בנוסחת פונקציית הצפיפות. יותר מכך, ראיינו שעבור התפלגות נורמלית סטנדרטית, הערכים יהיו כמעט כולם בתחום $[-3, 3]$. אפשר להכליל את הטענה, בכל התפלגות נורמלית, כמעט 100% מהערכים יהיו בתחום $[\mu - 3\sigma, \mu + 3\sigma]$, כ-95% יהיו בתחום של 2 סטיות תקן, וכ-68% בתחום של עד סטיות תקן אחת מהຕוחלת.



```
plt.hist(null_mean_diffs, bins = np.arange(-45, 45, 5), density = True)
x = np.arange(-45, 45, 0.01)

plt.plot(x, stats.norm.pdf(x, mu, sigma), color = 'red')
plt.show()
```



```
print(f' {mu - 2 * sigma: .2f}, {mu + 2 * sigma: .2f} ')
(-20.23, 19.87)
```

לדוגמא, באוכטוסית הפרשי הממוצעים שלנו, אם אנחנו טוענים שהוא מהתנהגת כמו התפלגות נורמלית, אנחנו אמורים להיות מסוגלים "להביש" עליה התפלגות זאת ולראות התאמה לא רעה. נאמוד את התוחלת μ עם ממוצע התפלגות, ואת σ עם סטיית התקן.

אם ההתאמה טובה, ואפשר לקבל את התפלגות ההפרשים המשומצת שלנו באמצעות התפלגות נורמלית, אפשר למשל להגיד אמירה כמו: "95% מהערכים יהיו במרחק 2 סטיות התקן מההתוחלת", לחשב, ולראות שהז' יוצא ערך (20,20) – ולחזר על הפרדיגמה של בדיקת השערות, ולומר שההתוצאה של הפרש 15 היא בהחלט בתחום זה של ± 20 וכן לא נראה חריגה. אז אנחנו מבינים למה הינו רוצה שההתפלגויות ייחוס כללה יתפלגו נורמלית – זו התפלגות שכונה לנו מאוד.

משפט הגבול המركזי

CLT: משפט הגבול המركזי טוען שבעור כל מבחן בגודל n צפיפות בלתי תלויות X_n, \dots, X_1 שבא מהתפלגות עם תוחלת μ ושונות סופיות σ^2 . אם גודל המבחן n גדול מספיק, ממוצע המבחן מתפלג בקרוב נורמלית, עם התוחלת המקורית μ ושונות קטנה פי n .

$$\frac{\sum_i X_i}{n} = \bar{X} \sim N\left(\mu, \frac{\sigma^2}{n}\right)$$

הרבה פעמים השתמש בגרסה המתוקננת של המשפט. אם נחסר ממוצע המבחן את התוחלת ונחלק בסטיית התקן, הבמות הزادת מתפלגת נורמלית סטנדרטית.

$$\frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \sim N(0,1)$$

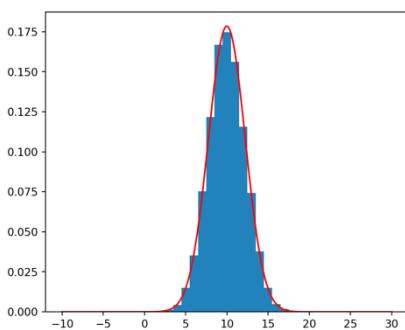
- סטיית התקן של ממוצע המבחן – **טעות התקן** (SE): $\frac{\sigma}{\sqrt{n}}$
- לכמota המתוקננת נקרא **סטטיסטי**: $Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$

חשיבות: המשפט נכון לכל התפלגות מקוריות של X , תהא צורתה אשר תהא, ובלבד שהשונות שלה סופית. הגרסה הראשונה של משפט הגבול המركזי הוכחה כבר במאה ה-18 ע"י דה-מורבר, ואחר כך במאה ה-19 ע"י פלס. המילה "מרכזי" הוכנסה במאה ה-20 כשהבינו עד כמה המשפט מהותי לבדיקת השערות.

דוגמה מהתפלגות ברנולי: יש לנו מבחן $X_n, \dots, X_1 \sim Ber(p = 0.5)$. מתקיים $E[X_i] = p = 0.5$ כאשר $(p = 0.5)$. $Var(X_i) = p(1 - p) = 0.25$. אנחנו יודעים כי סכום התציפות מתפלג בינומית: $\sum_i X_i \sim Bin(n, p = 0.5)$. משפט הגבול המركזי אומר לנו כי:

$$\frac{\sum_i X_i}{n} = \bar{X} \sim N\left(0.5, \frac{0.25}{n}\right) \Leftrightarrow \sum_i X_i = n\bar{X} \sim N(0.5n, 0.25n)$$

אם מספר הניסויים n גדול מספיק אז נקבל: $(n, 0.5) \approx Bin(0.5n, 0.25)$. נדגום את התוצאה של משתנהBINOM ונציר את ההיסטוגרמה כדי לראות את התפלגות, והוא נראה ממש בתפלגות נורמלית.



דוגמה מהתפלגות מעריכית: עבור $X \sim Exp(\lambda)$ זה אומר ש- X יכול לקבל ערכים בטוחה $[0, \infty]$. משפט הגבול המركזי אומר לנו פונקציית הצפיפות של $x \lambda e^{-\lambda x}$ ונוסחה ל-CDF של $x \lambda e^{-\lambda x}$ והשונות $\frac{1}{\lambda^2}$. לדוגמא, נהוג למדל זמן עד שיקרא אירוע מסויים במשתנה מעריכי. אז, הזמן בין שתי רכבות מثال אביב לחיפה, הוא לא קבוע, הוא משתנה ומתפלג מעריכית. בממוצע, מרתינים כ-20 דקות בין שתי רכבות. לכן התוחלת היא $\frac{1}{\lambda}$ (שעה) ולכן $\lambda = 3$. זו התפלגות שראתה בכלל לא סימטרית, לא נורמלית. אמנם, כאשר מסתכלים על התפלגות ממוצע המבחן, היא נורמלית סביב התוחלת המקורית $\frac{1}{3}$.



מחנכים סטטיסטיים

מבחן Z:

הישום המידי של CLT הוא **בדיקה השערות ומבחן Z**. נחזר לדוגמה של הזמן בין שתי רכבות מTEL אביב לחיפה. "אם הוא באמת 20 דקות? לאחרונה נראה יותר". השערת האפס היא תוחלת של 20 דקות: $H_0: \mu = \frac{1}{3}$. ההשערה האלטרנטיבית היא חד-צדדית, יותר מ-20 דקות: $H_1: \mu > \frac{1}{3}$. נציג בקורס אקרטיות 30 זמני המתנה, נחשב ממוצע, ונקבל $26 \approx \frac{4}{9}$ דקות המתנה. אולי זה קרה במקרה? צריך לדעת איך מתפלג ממוצע המדגם (הסטטיסטי שלנו).
לפי CLT: $\bar{X} \sim N(\frac{1}{3}, \frac{1/9}{30})$ עם התוחלת המקורי $\frac{1}{3}$ ושותות קטנה פי 30 מהשותות המקורי ($\text{שהיא } 1/9$). בקורס המתוקנה: $\bar{X} - 1/3 \sim N(0, 1/\sqrt{270})$. נציג ממוצע של $9/4 = \bar{X}$ ונקבל $1.825 = Z$. זה ערך של התפלגות נורמלית סטנדרטית, ואין שום בעיה לחשב לו **p-value**: הסיכוי לקבל ערך קיצוני כמו 1.825 או יותר, תחת ההתפלגות הנורמלית הסטנדרטית. לכל התהילה זה קוראים מבחן Z, ואפשר לעשות אותו אם השונות ידועה.

בפייטון נוכל להשתמש בפונקציה `cdf` (או `1 - cdf`) את ההסתברות להיות גדולים מערך כלשהו). ככל קorra, הכמות שאנו מחפשים תחת השערת האפס היא 3%, זה ה-p-value-ה חד-צדדי, ונראה קטן מאוד. לכן נדחה את השערת האפס (קטן מרמת מובהקות של 5%). בולם, זמני המתנה בתוחלת הם יותר מ-20 דקות.

```
one_sided_p_value = 1 - stats.norm.cdf(4/9, 1/3, np.sqrt(1/270))
#or
one_sided_p_value = 1 - stats.norm.cdf(1.8257, 0, 1)

print(f'P(X_bar >= 4/9 | H0) = {one_sided_p_value: .2f}')
P(X_bar >= 4/9 | H0) = 0.03
```

מה היה קורה אם $2/9 = \bar{X}$ בערך 13 דקות? אפשר היה להמשיך ברגיל ולהчисל p-value-k. אולם, התוצאה הזאת בכלל לא בכיוון **השערת האלטרנטיבית** של זמן המתנה של יותר מ-20 דקות. בולם ההשערה נדחית על הסף.

דוגמת הציורים: נגדיר X רמת אדום בציורי MI, ו-Y רמת אדום בציורי RE, ולא נניח שהם מתפלגים נורמלית. זה בסדר כי יש לנו את CLT. תחת H_0 התוחלות שלן זהות לערך μ , והשוניות גם זהות לערך σ^2 . גודל המדגם הוא 30 בשני המדגמים.

נתון לפי CLT כי ממוצע כל אחד מהדגמים מתקרב בקירוב לנורמלית. בולם $N\left(\mu, \frac{\sigma^2}{n}\right) \sim \bar{X} \sim \bar{Y}$. אנחנו מתעניינים בהפרשי הממוצעים (הפרש משתנים נורמליים מתפלג נורמלית עם הפרש התוחלות, וסכום השוניות):

$$\bar{X} - \bar{Y} \sim N\left(0, \frac{2\sigma^2}{n}\right) \Leftrightarrow \frac{(\bar{X} - \bar{Y}) - 0}{\sqrt{\frac{2\sigma^2}{n}}} \sim N(0, 1)$$

אנו, אנחנו לא יודעים את התוחלת והשוניות. התוחלת נעלמת תחת ההפרש, אבל מה לגבי השוניות? נניח שאנחנו יודעים אותה, אז נוכל לחשב אותה: $Z = \frac{15}{10.15} = 1.48 = np.std(population) = 39.3$. נקבל $np.std(population) = 10.15 = \sqrt{\frac{2 \cdot 39.3^2}{30}}$. נקבל p-value של 7% וזה מאד דומה לערך שקיבלנו בסימולציה – לא מספיק קטן ומרשים כדי לדחות את השערת האפס.

זה לא סביר שנדע את סטיית התקן של ההתפלגות המקורי – ובמקרה זה אנחנו עושים התאמות ובמקום מבחן Z מקבלים את מבחן T.

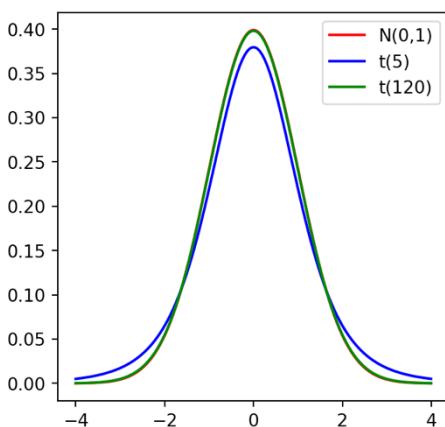
```
one_sided_p_value = 1 - stats.norm.cdf(15, 0, 10.15)
#or
one_sided_p_value = 1 - stats.norm.cdf(1.48, 0, 1)

print(f'P(mean_diff >= 15 | H0) = {one_sided_p_value: .2f}')
P(mean_diff >= 15 | H0) = 0.07
```

מבחן T:

התפלגות T: מה נעשה כשההתפלגות המקורית והשונות שלה לא באמת ידועות? במקרה זה נחליף את σ^2 באמצעות האומד (חסר הטיה) לשונות שהתקבל מהדגם ומסומן $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$, ומתקיים $\sigma^2 = E[S^2]$. במקרה שאנחנו מביצים את ההחלפה, הסטטיסטי Z שלנו כבר לא מתפלג נורמלית סטנדרטית, אלא **מתפלג T** עם פרמטר שנקרא דרגות חופש והוא $1 - n$. עבור המשטנה שהגדכנו קודם $Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}$, נחליף את σ ב-S ונקבל:

$$T = Z \cdot \frac{\sigma}{S} = \frac{\bar{X} - \mu}{\frac{S}{\sqrt{n}}} \sim t_{n-1}$$



לפונקציית הצפיפות של T יש צורה שמצוירה מאוד את ההתפלגות הנורמלית, פשוט רחבה יותר, והזנותה שלה עבים יותר. אפשר לראות את זה בתוצאה של עד איז-וידאות שנכנסה לתהיל' הסתברותי. הרוי אנחנו אומדים בעת את סטיית התקן S והוא לא נתונה לנו במקודם.

נציר את ההתפלגות T עם 5 דרגות חופש בכחול, ולידה 120 דרגות חופש בירוק. קשה לראות את ההתפלגות הנורמלית הסטנדרטית באדום, היא ממש בלתי מובחנת מההתפלגות בירוק. הסיבה היא שבכל שהדגם או דרגות החופש גדולות, כך ההתפלגות T געשית הרבה יותר וושאפת להתפלגות הנורמלית סטנדרטית. עבור מדגמים גדולים מספיק, זה כבר לא משנה כל כך אם משתמשים במבחן Z עם הנחה שסטיית התקן ידועה או משתמשים במבחן T עם אומד לסטיית התקן.

מבחן T: הסטטיסטי T מתקבל כאשר מחסרים מהממוצע את התוחלת (תחת השערת האפס) ומחלקים בטיעות התקן הנאמנת:

$$T = \frac{\bar{X} - \mu_{H_0}}{\sqrt{\frac{S^2}{n}}} \underset{H_0}{\sim} t_{n-1}, \quad S = \sqrt{\frac{1}{n_x - 1} \sum (X_i - \bar{X})^2}$$

אם נרצה לקבל אותו ואת ה-value-p שמקורר אליו, אפשר להשתמש בפונקציה `(x,0).stats.ttest_1samp()`. לדוגמה, אם תחת השערת האפס התוחלת היא 0, והדגם שלנו נמצא במצב אובייקטיבי, כך נבצע מבחן T לראות אם התוחלת שונה מ零.

אם יש לנו שני מדגמים, המבחן הראו הוא **מבחן T למדגמים ב"ת"**. יש לנו מדגם $X_{n_x}, X_1, \dots, X_{n_x}$ (במלה אדום בציורים MI), ומדגם זהה עבור ציורים RE, $Y_{n_y}, Y_1, \dots, Y_{n_y}$. במקרה זה $n_x = n_y = 30$ (זה לא חייב להיות כך). השערת האפס היא $\mu_y = \mu_x$: H_0 . נניח גם שווין שונות: $\sigma^2 = \sigma_X^2 = \sigma_Y^2$. הסטטיסטי המתוקן הוא הפרש הממוצעים, פחות הפרש התוחלות, חלקי טיעות התקן שמקבלת כאן צורה אחרת:

$$\frac{(\bar{X} - \bar{Y}) - (\mu_x - \mu_y)}{\sqrt{\sigma^2 \left(\frac{1}{n_x} + \frac{1}{n_y} \right)}} \sim N(0,1)$$

זה היה מתפלג בקירוב נורמלית סטנדרטית תחת CLT, אבל אנחנו לא ידעים מה הן השונות המשותפת, ומחליפים אותן באמצעות שנקרא S_p^2 שהוא ממוצע משוקלל של האומד לשונות של מדגם X ושל מדגם Y. הסטטיסטי הסופי שלנו, הוא הפרש הממוצעים, פחות הפרש התוחלות, מחולק באומד לטיעות התקן:

$$T = \frac{(\bar{X} - \bar{Y}) - (\mu_x - \mu_y)}{\sqrt{S_p^2 \left(\frac{1}{n_x} + \frac{1}{n_y} \right)}} \underset{H_0}{\sim} t_{n_x + n_y - 2}, \quad S_p^2 = \frac{(n_x - 1)S_x^2 + (n_y - 1)S_y^2}{n_x + n_y - 2}$$

נשים לב שהרבה פעמים השערת האפס היא $\mu_x = \mu_y$ ואז הפרש זהה במנונה יהיה 0.

הערה: מה אם לא נניח שווין שונות? האומד לטיעות התקן שלנו משתנה מעט,哿ר לא מחשבים את S_p^2 , מחשבים ביטוי אחר. אז מקובל לבצע תיקון בחישוב דרגות החופש df . יש נוסחה לא מאד סימפטית לה...>.



דוגמת ה μ : השערת האפס שלנו היא אכן שתחולת רמת האודם בציורים RE ו-MI היא זהה. הפרש התוחלות הוא 0. נסתכל על 10 התוצאות הראשונות מכל מדגם. במצב ידנית (כדי לראות שהנוסחאות עובדות), אפשר לחשב את סטטיסטי T, שיזכר 1.38. באטעןות המתודה stats.t.cdf() נחשב את p -value של hypothesis T. הסיכוי לקבל בהתפלגות T את הערך 1.38 או יותר. את כל החישוב המסורבל הזה ניתן לבצע בשורה אחת עם הפונקציה ttest_ind(). כדי לבצע מבחן חד-צדדי צריך לפרט פרמטר 'alternative='greater''.

```
stats.ttest_ind(impr_red, real_red, alternative='greater')
Ttest_indResult(statistic=1.385446057741497, pvalue=0.08561058232473277)
```

נשים לב שה- p -value של 8.5% גדול יותר מה- p -value של מבחן Z שיצא 7%. זה לא במקורה, ויתרנו על הנחת השונות היודעה, זה בא לידי ביטוי בתיאוריה שלנו שהובילה אותנו להתפלגות Z שיש לה זוגות עבים יותר. כמובן, הסיכוי לקבל תוצאות קיצניות יותר, יותר גובל בהתפלגות החדש. כדי לדוחות את השערת האפס בהתפלגות Z, צריך תוצאות קיצניות יותר.

רווח סמך

רקע: אחת הביעות עם חישובי μ -value או אוזורי דחיה, היא שמדובר במבחן החלטה ביןאי שאיננו אינפורטטיבי. אם מגדים את מרכיב המדגם, גם אפקטים קטנים מאוד יכולים להיות פתאום מובהקים סטטיסטי. זה שאפקט מובהק סטטיסטי, לא אומר שהוא אכן מעוניין מדעית, מה שambil הרבה פעמים מדענים לעשות ודוקציה למחקר המדעי שלהם, למעבר של הערך הזה ותו לא. אולי עדיף לתאר אומד לפרקטי עליון מתבצע המבחן? תוחלת של משתנה או הפרש התוחלות. הביעה היא שהפרש ממוצע המדגם הוא אומד נקודתי שבסבירות גבוהה אין נכו. لكن הינו רצים לעת טווח כלשהו – רוח סמך (confidence interval). האמירה תהיה, שהתוחלת μ תהיה בטוחה מרווח מסוים $[\bar{X} - \epsilon, \bar{X} + \epsilon]$ ברמת ביטחון מסוימת.

בנית CI מבחן Z: נרצה ערכים $[LB(X), UB(X)]$ שאנו נתונים שבהתברות $(\alpha - 1)$ מכיסים פרמטר אמיטי θ שאנו אנו חרים להערך. טווח זה קרא רוח סמך ברמת ביטחון $\alpha - 1 = 100\%$. ורוח הסמך שלו יהיה סימטרי סביב \bar{X} אומד $\hat{\theta}$ שבסבוס על המדגם, למשל ממוצע המדגם \bar{X} . עבור מבחן Z, אנחנו יודעים מה-CLT שמתקיים:

$$P\left(Z_{0.025} < \frac{\bar{X} - \mu}{\frac{\sigma}{\sqrt{n}}} < Z_{0.975}\right) = P\left(\left[\bar{X} + Z_{0.025} \cdot \frac{\sigma}{\sqrt{n}}\right]_L < \mu < \left[\bar{X} + Z_{0.975} \cdot \frac{\sigma}{\sqrt{n}}\right]_U\right) = 0.95$$

נקבל חסם תחתון ועלין שמכסים את הפרמטר μ בהסתברות 0.95. הם מהווים רוח סמך ברמת ביטחון 0.95 עברו התוחלת. מכאן

שאחוזונים של ההתפלגות הנורמלית הסטנדרטית הם סימטריים, נקבל $\frac{\sigma}{\sqrt{n}} \cdot 1.96 \pm \bar{X}$.

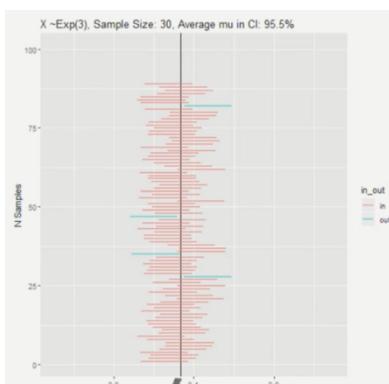
```
LB = 4/9 - stats.norm.ppf(0.975) * (1/3)/np.sqrt(30)
UB = 4/9 + stats.norm.ppf(0.975) * (1/3)/np.sqrt(30)
print(f'[LB: {LB:.3f}, {UB:.3f}]')
```

דוגמת זמני המתנה לריבבת: תחת השערת האפס, זמן המתנה מתפלג $Exp(\lambda)$, והממוצע מתפלג נורמלי בקיורוב CLT. נניח גם שסטטיסטיקת Z צייבת $\frac{1}{3}$. התקבל ממוצע מדגם 30 = $n = \frac{1}{9}$. $\bar{X} = \frac{4}{9}$.

בנוסף רוח הסמך ונתקבל: [0.325, 0.564]. תוחלת הזמן של השערת האפס (20 דקות = 0.333), נמצאת ברוח הסמך. כמובן, אנחנו לא פסלים את האפשרות הזאת.

אם נסמן, דחינו את השערת האפס כשביצענו בדיקת השערות לנواتים אלה במבחן Z, אבל ביצענו בדיקת השערות חד-צדנית. שיערנו מראש זמני המתנה ארוכים יותר ($\frac{1}{3} > \mu$). אם הינו מבצעים בדיקה דו-צדנית ($\frac{1}{3} \neq \mu$) אך לא הינו דוחים את השערת האפס גם

בגישה ה- μ -value. יש קשר הדוק בין בדיקת השערות דו-צדנית להימצאות או אי-הימצאות הפרמטר מהשערת האפס ברוח הסמך.



הערה: טווח נפוצה שעושים בתחום היא להגיד "הסתברות $\mu - \sigma$ בטוחה [0.33, 0.56]" או "95%". מבחינה הסתברותית זו אמירה בעייתית מאוד. התוחלת היא פרמטר, היא או נמצאת ברוח הסמך או לא – היא לא משתנה מקרי! למה לנו מתכוונים ברמת הביטחון? אנחנו אומרים שאם הינוعروכים הרבה מדגמים בצורה דומה, ב-95% מהם, רוח הסמך שאנו בונים, יבסה את הפרמטר האמיטי μ , כך שהאמירה ההסתברותית היא על גבולות רוח הסמך – הם המשתנים המקרים! לא הפרטן!



בנייה CI מבחן T: עבור מדגם בודד יש לנו נוסחה, רק שהפעם סטיית התקן לא ידועה והיא נאמדת באמצעות S , והאחוזונים מגיעים מההתפלגות T עם $1 - n$ דרגות חופש. אותו דבר למוגמים ב"ת, נרצה לבנות רוח סマー להפרש התוצאות. עדיין נקבל נוסחה מהצורה " $\text{סטטיסטי} \pm \text{טוח}$ ".

$$\left[\bar{X} + t_{n-1;0.025} \cdot \frac{S}{\sqrt{n}}, \bar{X} + t_{n-1;0.975} \cdot \frac{S}{\sqrt{n}} \right] \text{ or } \bar{X} \pm t_{n-1,0.975} \cdot \frac{S}{\sqrt{n}}$$

is a 95% CI for μ

$$\left[(\bar{X} - \bar{Y}) + t_{n_x+n_y-2;0.025} \sqrt{\frac{S_p^2}{n_x} + \frac{S_p^2}{n_y}}, (\bar{X} - \bar{Y}) + t_{n_x+n_y-2;0.975} \sqrt{\frac{S_p^2}{n_x} + \frac{S_p^2}{n_y}} \right] \text{ or}$$

$$(\bar{X} - \bar{Y}) \pm t_{n_x+n_y-2,0.975} \sqrt{\frac{S_p^2}{n_x} + \frac{S_p^2}{n_y}}$$

is a 95% CI for $\mu_x - \mu_y$

בוחנת הצירום: נשים לב כי רוח הסマー כולל את האפשרות 0 בתוכו = אין הבדל בرمת האדים בין שני סגנונות הצירום. מכאן לא הינו דוחים השערת אפס דו-צדדית: $H_0: \mu_x = \mu_y, H_1: \mu_x \neq \mu_y$ בرمת מובהקות 5%.

Recall we got: $\bar{X} - \bar{Y} = 15$; $n_x = 30$; $n_y = 30$; $S_p^2 = 1824$

95% CI for $\mu_x - \mu_y$:

```
LB = (np.mean(impr_red) - np.mean(real_red)) - stats.t.ppf(0.975, n_x + n_y - 2) * np.sqrt(S_p**2 / (n_x + n_y - 2))
UB = (np.mean(impr_red) - np.mean(real_red)) + stats.t.ppf(0.975, n_x + n_y - 2) * np.sqrt(S_p**2 / (n_x + n_y - 2))
print(f'[{LB:.1f}, {UB:.1f}]')
[-6.8, 37.4]
```

עוצמת המבחן

רקע: נזכיר מהי עוצמת המבחן באמצעות הדוגמה של הצירום. כאן, כל האוכלוסייה היא בcanf' ידנו, בשורה אחת של קוד נגלה מההבדל בرمת האדים בין סוג הצירום. נמדד את ההפרש בין כל הצירום בדאנא, ויש הבדל של 15.35 בرمת האדים – זו האמת. נניח לרגע שהבדל מעוניין מבחינה מדעית (למרות שבסקאלת אדים של 0 עד 255 לא בטוח שזה מעוניין). זה אומר שהשערת האפס שלנו אוטה לא דוחינו, הייתה שגיה! ובעצם **ביצענו טעות מסוג ראשון**. ומה טעינו? ואם הינו עורכים את הניסוי שוב האם סביר שהיינו טועים שוב? **התשובה נעוצה בעוצמת המבחן – ההסתברות לדחות את השערת האפס כשהו באמת לא נכונה**. כאן, ההסתברות זו הייתה קטנה, ואם הינו אומדים אותה מראש, אולי הינו יכולים לתכנן ניסוי טוב יותר. **עוצמת המבחן מסומנת (reject H_0) | H_1 true = P**

סימולציה: ראשית נראה זאת עם סימולציה. נבצע ניסוי שמעורב לקיחת שני מדגמים בגודל של התקציב שלנו $30 = n$ מכל סוג, שוב ושוב, לביצוע מבחן T למוגמים ב"ת, ולבוחן בכמה מהפעים נדחה בצדק את השערת האפס.

```
def random_sample_t_test(alpha, n):
    real_red_sample = np.random.choice(real_red_all, n, replace=False)
    impr_red_sample = np.random.choice(impr_red_all, n, replace=False)
    t_test = stats.ttest_ind(impr_red_sample, real_red_sample, alternative='greater')
    return (impr_red_sample.mean() > real_red_sample.mean()) and t_test[1] < alpha

n_simulations = 10000
n = 30
alpha = 0.05

rejections = [random_sample_t_test(alpha, n) for i in range(n_simulations)]
print(f'Power = P(reject H0 | H1) = {np.mean(rejections): .2f}')
Power = P(reject H0 | H1) = 0.45
```

הfonקציה מבצעת דוגמה ומבחן T חד-צדדי, ומבחן האם יש דוחיה או לא. נעשה זאת 10,000 פעמים ומחשבים כמה פעמים דוחינו בצדק את השערת האפס. **נקבל סיכוי של 45% עצמה של 45% פירושה, שבערך ב-1 מכל 2 מבחנים לא נדחה את H_0 בשוצריך לדחות אותה** כי הממציאות היא H_1 . אם הינו יודעים חזיו עוצמת המבחן, סביר להניח שהיינו רוצים לשנות את הניסוי כדי להגדיל אותה.

גודל האפקט: בפועל, אין לנו את כל האוכלוסייה, אנחנו לא יכולים לחזור על הניסוי 10,000 פעמים, והבי מאתגר – אנחנו לא באמת יודעים את הפרש התוצאות באוכלוסייה (גם אם נניח שהוא חיובי). גישה נפוצה במקרה זה, היא לחשב את עוצמת המבחן לפי פרמטר שנקרה לו **effect size** ולנסות להתחשב בכמה גדי אפקט אלו. הוא מוגדר כך: $\frac{\mu_H - \mu_L}{\sigma}$ כולל סטנדרטיזציה לסטיית התקן, ומוגדים במונחי סטיית התקן (למשל אפקט של "0.5 סטיית התקן"). אם נניח שהפרש התוצאות הוא 15 תחת ההשערה החלופית -0 תחת השערת האפס, لكن גודל האפקט בלי תקנון הוא 15 נקודות. נשאל מה הסיכוי לדחות את H_0 תחת האפקט זהה, בולם לקבל $\text{value}-\text{k}$ קטע מרמת המובהקות בשחרש התוצאות הוא 15:

$$\pi = P[\text{reject } H_0 | H_1 \text{ true}] = P(p - \text{value} < \alpha | \text{true mean diff is } 15) = \dots$$

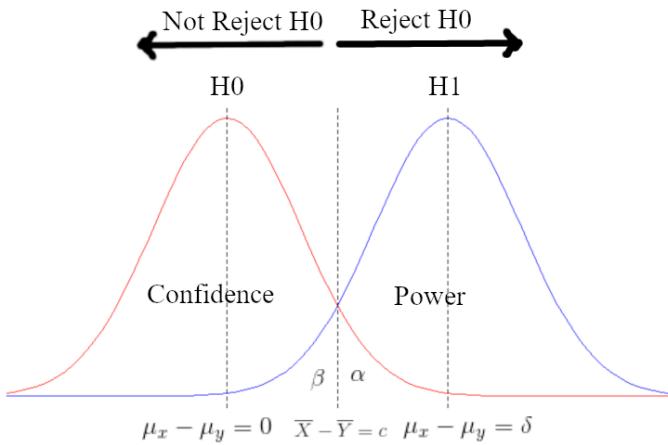
מדובר במציאת הסתברות תחת הnull hypothesis T עם בר ובר דרגות חופש. מציבים את הביטוי בפיטון ומתקבלים שהעוצמה היא 0.38. ככלומר, רק ב-40% מהניסויים היינו דוחים בצדκ את השערת האפס, ב מבחן T למדגמים ב"ת", ברמת מובהקות 5%. הערך החסר הוא העוצמה, ונקבל כי היא 0.35. הערך החסר הוא העוצמה, ונקבל כי היא 0.38.

```
# with Python's built-in method:
from statsmodels.stats.power import TTestIndPower

effect = 15/np.sqrt(s2_p) # true means difference divided by s_p
power = TTestIndPower().power(effect_size = effect, nobs1 = 30, alpha = 0.05, ratio = 1)

print(f'Effect size = {effect : .2f} Power = P(reject H0 | H1) = {power: .2f} ')
Effect size = 0.35 Power = P(reject H0 | H1) = 0.38
```

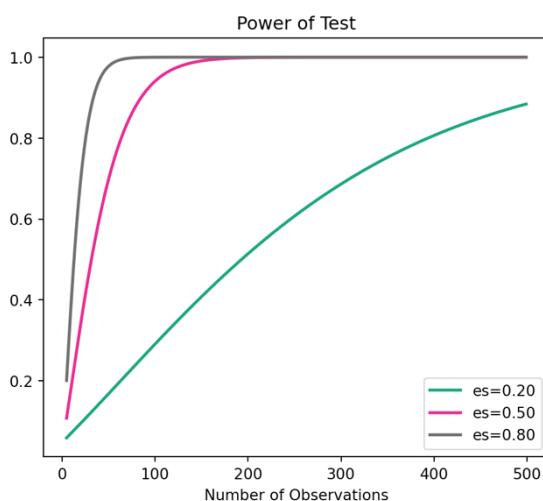
עוצמת המבחן גדלה באשר:



- **גודל האפקט:** אם ההתפליגויות היו רוחוקות יותר זו מזו, ככלומר גודל האפקט גדול בכל הנinstant. התופעה שאנוחנו מודדים גדולה, ברורה ומובחנת יותר.
- **סטיית התקן (ס'):** אם ההתפליגויות היו חותת יותר, ככל שאפשר להקטין את שונות התופעה, בר ייה קל להבחין בהבדלים דקים יותר.
- **גודל המדגם (נ'):** הדרך הקללה ביותר להגדיל את העוצמה, אף על פי שהיא קשורה בקשר ישיר לתקציב הניסוי.
- **רמת המובהקות (α):** אם α הייתה גדולה יותר, בר נקבע את הערך הקритי אך נגדיל את הסיכוי לטעות מסדר ראשון. זה נחשב למנהג פסול – "מוכנים לטעות בניסוי הרבה".

ניתן לראות את זה גם מהביטוי: נרצה שהוא יהיה כמה שייותר גדול, וכן ההסתברות שאנו חשים מחסירים, כמו שייותר קטנה.

$$1 - P \left(\frac{\bar{X} - \bar{Y} - (\mu_x - \mu_y)}{\sqrt{\frac{s_p^2}{n_x} + \frac{s_p^2}{n_y}}} < t_{n_x + n_y - 2; 1-\alpha} - \frac{(\mu_x - \mu_y)}{\sqrt{\frac{s_p^2}{n_x} + \frac{s_p^2}{n_y}}} \mid \mu_x - \mu_y = 15 \right)$$



בהתוח עוצמה: נסתכל גם על השרטוט באמצעות `power_plot`, ונקבל עקומות עוצמה עבור ערכים שונים, ולהבין היכן אנחנו נמצאים. כאן אנחנו בזדקים גדלי מדגם שונים, וגודל אפקט שונה (במנוחים של סטיית התקן). עברו גודל אפקט צבוע (0.2) אם נרצה עוצמה של 80 אחוז לפחות, נדרש גודל מדגם של 400 בערך.

לסבירם: מובהקות סטטיסטית אינה מובהקות מדעית! יש לבדוק האם הממצא שמתארים אכן מעניין מדעית. נניח בדוגמה הצירורים שרמת האדים ב-MI הייתה גדולה מ-RE בחצי נקודה בלבד, ככלומר גודל האפקט של עבר 0.01, מאות סטיית התקן. אם נבצע ניתוח עוצמה, נראה שעבור מדגם גדול מספיק, אם יהיו לנו מעל 150,000 דוגמאות, היינו מקבלים תוצאה מובהקת סטטיסטית בעוצמה של מעל 80%. הלקח: לא לדודף רק אחריו מובהקות סטטיסטית, לדודוח על גודל האפקט, רוחם סמרק לפרמטר, ולא רק p -value.



קוויז 8

שאלה 1: התאמת מושגים לסימנים:

ז' μ	הפרש תוחלות שני מדגמים בלתי-תלויים
$\frac{\sigma}{\sqrt{n}}$	טעות התקן (סטטיסטית התקן של ממוצע המדגם)
σ^2	שונות ההתפלגות הנורמלית
S	סטטיסטית התקן של המדגם
\bar{X}	ממוצע המדגם
φ	פונקציית ההתפלגות המצטברת (CDF) הנורמלית סטנדרטית

שאלה 2: מה הבאים בהכרח נכון לגבי ביצוע מבחן סטטיסטי להשערה אפס דו-צדדית לתחולת מדגם בודד, באשר גודל המדגם קטן, נאמר פחות מ20.

- לא ניתן להשתמש במשפט הגבול המרceği או בהתפלגות הנורמלית באשר גודל המדגם כל כך קטן – **אם ידוע שהמשתנה בעל ההתפלגות נורמלית ממוצע המדגם יתפלג נורמלית בלי קשר** לגודלו המדגם.
- תחת הנחות מתאימות ניתן להשתמש בהתפלגות הנורמלית אבל עצמת המבחן תהיה קטנה – **יש מספר דברים המשפיעים על עצמת המבחן, גודל המדגם הוא רק אחד מהם.**
- טעות התקן תהיה גדולה מה שיגדל את רוח הסמך כל כך שנתקשה לדוחות את השערת האפס – **טעות התקן אינה נקבע רק מגודלו המדגם אלא גם מהשינויים הטבעית של הנסיבות הנ마다.**
- אף תשובה אינה בהכרח נכונה.**

שאלה 3: אליוו עורך ניסוי להשוואה שתי שיטות להרידה. קבוצה של נבדקים בבדי משקל מחולקת באקריא לשתי קבוצות, האחת עוסרת טיפול א' והשנייה עוסרת טיפול ב'. משקל הנבדקים נמדדים בקילוגרם לפניהם ואחרי הטיפול והפרשיים מושווים באמצעות מב奸 T למוגדים בלתי תלויים תוך הקפדה על ההנחה הסטטיסטיות המתאימות. איזה מהתרחישים הבאים יקיין את עצמת המבחן?

- מסתבר שההकצהה לקבוצות לא הייתה אקראית ויש בקבוצת טיפול א' אנשים שמניכים יותר בממוצע מקבוצת טיפול ב' בכ-1 קילוגרם – **רצוי שהההקצתה תהיה אקראית, אבל המבחן הסטטיסטי נערך על הפרשיים במשקלים, לא על המשקלים עצםם**, והבדל של 1 קילו לא בטוח **שישיפיע על הפרשיים במשקלים בין שתי השיטות**. אם ההבדל היה גדול יותר **יתכן** והדבר דואק היה מגדיל את עצמת המבחן (אפקט תקרה, لأنשים בבדי משקל בקבוצת א' יש "יותר להוריד" ונראה הבדל גדול יותר) אבל הבדל בה גודל בביישליין כבר מערער על הולידות של כל המבחן.
- שונות ההפחתה במשקל בקרוב שתי הקבוצות קטנה מאוד, נאמר 1 – **שונות קטנה מגדילה את עצמת המבחן הסטטיסטי, קל יותר לראות את האפקט מעבר לרעש.**
- ההבדל "אמיתי" בין ההפחטה במשקל בין שתי השיטות הוא מודער, פחות ממחצית קילוגרם – גודל אפקט קטן מקשה על מדידה מהימנה.
- אליהו לא מחויב לשום סטנדרטים מחייבים והוא משתמש ברמת מובהקות של 20 אחוז במקום 5 או 1 אחוז בנהוג – **ניפוי הטעות מסוג ראשון תביא דואק להקטנת הטעות מסווג שני ולהגדלת עצמת המבחן.**



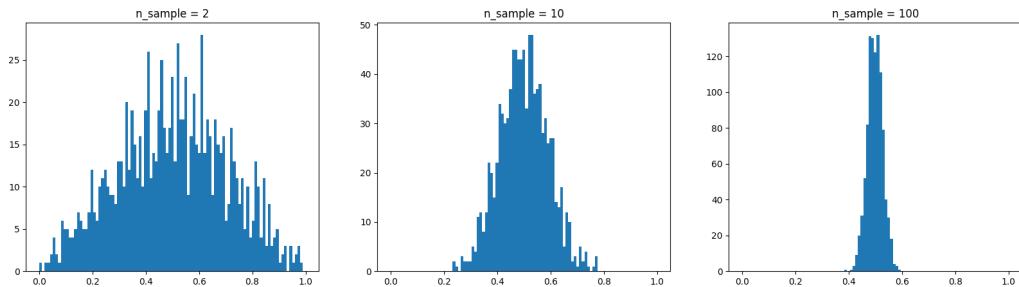
תרגול 5 – הסקה סטטיסטית ב'

משפט הגבול המרבי:

זוכרת: יש לנו משתנים IID המסומנים Y_n, Y_1, \dots, Y_1 , עם תוחלת μ ושונות σ^2 . נגיד $\bar{Y}_n = \frac{1}{n} \sum Y_i$ אז כאשר $n \rightarrow \infty$ מתקיים:

$$\sqrt{n}(\bar{Y}_n - \mu) \rightarrow N(0, \sigma^2)$$

כלומר, עבור n גדול מספיק, ההתפלגות של הממוצע \bar{Y}_n היא בקירוב $N\left(\mu, \frac{\sigma^2}{n}\right)$. נראה דוגמה באמצעות סימולציה: נדגום באופן רנדומלי מהתפלגות אחידה, מדגים בגודל 2, 10, 100, ונצರף את זה ליקטו. בשמגעים ל-100 = n זה כבר ממש פעמוני. השונות קטנה פי n .



דוגמה: נניח שיש לנו התפלגות אחידה: $(X \sim U(0,1))$ אז ממוצע של n משתנים באלו יתפלג נורמלית באופן הבא: $\bar{X}_n \sim N\left(\frac{1}{2}, \frac{1}{12n}\right)$

שאלה 0: נחזור לדני שמחה לאוטובוס, הזמן מתפלג $Exp(\frac{1}{\lambda})$, הערך הקритי הוא 30 דקות. נגיד $\lambda = 1$. הגדרנו $T(Y) = Y_1 + \dots + Y_{20}$. הסטטיסטיק היה זמן ההגעה של האוטובוס הראשון: $H_0: \mu = 20, H_1: \mu > 20$. טעות מסדר ראשון:

$$\alpha = P[\text{reject } H_0 | H_0 \text{ true}] = P_{H_0}[Y_1 > 30] = e^{-\frac{1}{20} \cdot 30} = 0.22$$

שאלה 1: דני מאוכזבת מהניסי. כדי להגדיל את עוצמת הניסוי היא דוגמת 100 זמני הגעה Y_1, \dots, Y_{100} ומסתכלת על הממוצע כדי להחליט האם לדוחות את השערת האפס. מהו הערך הקритי עבור \bar{Y} שמעליו נדחה את השערת האפס ברמת מובהקות 5%?

נגיד $\bar{Y} \sim N\left(\mu, \frac{20^2}{100}\right)$, הקבוע יהיה C (מה שאנו צריכים למצאו), ונזכור שכאשר n גדול מ-CLT נקבל

בדומה לשאלה 7 מתרגול 4 נחשב:

```
from scipy.stats import norm
norm.ppf(0.95, loc=20, scale=2)
alpha = P[reject H0 | H0 true] = P_{\bar{Y} \sim H0}(\bar{Y} \geq C) = P_{\mu=20}(\bar{Y} \geq C) = 1 - \Phi\left(\frac{C-20}{\sqrt{4}}\right) = 0.05 \Leftrightarrow \Phi\left(\frac{C-20}{2}\right) = 0.95 \Leftrightarrow Z_{95} = \frac{C-20}{2} \Leftrightarrow C = 20 + Z_{95} \cdot 2 = 23.29
```

```
def one_sample_mean(scale=20, n=100):
    return np.mean(np.random.exponential(scale=scale, size=n))
n_samples = 100000
y_means = np.array([one_sample_mean() for i in range(n_samples)])
print(f'Critical value above which 5% under H0: {np.quantile(y_means, 0.95):.2f}')
```

Critical value above which 5% under H0: 23.37

שאלה 2: דני קיבל $\bar{Y} = 24$. באיזה רמת מובהקות נדחה את השערת האפס?

הערך הקритי יצא 23.29, ומה שהתקבל נמצא באוזור הדחיה. נחשב את ה-p-value:

$$P - value = P_{H_0}[T(Y) > t_{obs}] = P_{H_0}[\bar{Y} > 24] = 1 - \Phi\left(\frac{24 - 20}{2}\right) = 1 - \Phi(2) = 0.023$$

היא תדחה עבור α של 10% ושל 5%

שאלה 12 מוגדרת: – מחיר כפה וקרואסן בתל אביב. מה הערך הקרייטי של המחיר הממוצע במדגם, שמעליו נדחה את השערת האפס ברמת מובהקות 5%?
 נגיד $(N \sim Y_1 + 2Y_2 = 20 + Y_1 + 2X; X \sim Exp(\frac{1}{5}))$ עברו מחיר בס קפה, $Y_2 = 10 + X$; $Y_1 + 2Y_2 = 20 + Y_1 + 2X = Y$ עברו המחיר הכללי. ראיינו במלטה כי קיבל $101 = \mu$.
 נגיד בעת את ההשערות: $H_0: \mu = 45$; $H_1: \mu > 45$. תחת השערת האפס לפי CLT קיבל $\bar{Y}_{30} \sim (45, \frac{101}{30})$.

```
from scipy.stats import norm
norm.ppf(0.95, loc=45, scale = np.sqrt(101/30))
48.01805615622394
```

$$\alpha = P[\bar{Y} > C] = 1 - \Phi\left(\frac{C - 45}{\sqrt{\frac{101}{30}}}\right) = 0.05$$

$$\Leftrightarrow C = 45 + Z_{95} \cdot \sqrt{\frac{101}{30}} = 48.02$$

רוח סמן:

תווכות: עברו $\alpha = 0.05$ למשל נקבל $[\bar{X} + Z_{2.5} \cdot \frac{\sigma}{\sqrt{n}}, \bar{X} + Z_{97.5} \cdot \frac{\sigma}{\sqrt{n}}]$ או $[\bar{X} - Z_{97.5} \cdot \frac{\sigma}{\sqrt{n}}, \bar{X} + Z_{97.5} \cdot \frac{\sigma}{\sqrt{n}}]$.

```
import numpy as np
from scipy.stats import norm
sample_mean = 97
n = 40
sigma = 10
L = sample_mean - norm.ppf(0.975) * sigma / np.sqrt(n)
U = sample_mean + norm.ppf(0.975) * sigma / np.sqrt(n)
print(f'95% CI for mu: [{L:.02f}, {U:.02f}]')
```

95% CI for mu: [93.90, 100.10]

שאלה 4: ידוע שבגביע קווטיל שוקל 100 = μ גרם עם 10 = σ . דגנית בטוחה שזה לא הממוצע הנכון, אבל השונות זהה. היא דגמה 40 = n גבאים, וקיבלה $97 = \bar{X}$.
 בנו רוח סמן לתוחלת ברמת ביטחון 95%.

נציב בנוסחה ונקבל: $[\bar{X} - Z_{0.975} \cdot \frac{\sigma}{\sqrt{n}}, \bar{X} + Z_{0.975} \cdot \frac{\sigma}{\sqrt{n}}] = [93.90, 100.10]$

```
# 1
val = (103-100) / (10/np.sqrt(40))
2 * (1 - norm.cdf(val))

0.05777957112359711
```

```
# 2
val = (97-100) / (10/np.sqrt(40))
2 * (norm.cdf(val))

0.057779571123597204
```

```
alpha = 0.05
# two sided p-value for mean which is smaller than mu_0
p_val = 2 * norm.cdf(sample_mean, loc=100, scale = sigma/np.sqrt(n))
if p_val < alpha:
    print(f'Reject H0, p-value = {p_val: .3f}')
else:
    print(f'Do not reject H0, p-value = {p_val: .3f}')

Do not reject H0, p-value = 0.0578
```

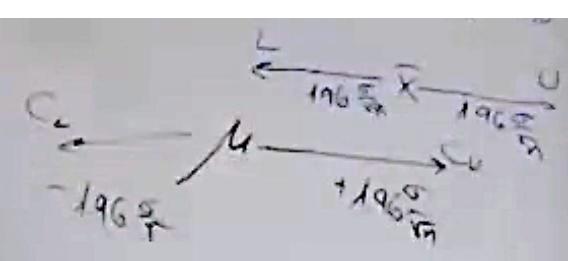
שאלה 5: כאן בבר נבצע בדיקת השערות. האם יש לדחות את השערת האפס $100 = \mu$ ברמת מובהקות 5%?
 נחשב $p\text{-value}$ דו-צדדי! נשים לב כי הערך שהתקבל (97) הוא קטן מהתוחלת המקורית (100) ולכן צריך לבדוק גם את הקצה השני (100+3) ולא רק (100-3).

$$P\text{-value} = P_{H_0}[\bar{X} > 103, \bar{X} < 97]$$

$$= 2 \cdot \left[1 - \Phi\left(\frac{103 - 100}{\sqrt{40}}\right) \right]$$

$$= 2 \cdot \Phi\left(\frac{97 - 100}{\sqrt{40}}\right) = 0.0578$$

מעבר מבדק דו-צדדי: רוח הסמן שלנו (סביב \bar{X}) כולל את התוחלת 100 = μ_0 !
 וכן לא נדחה את השערת האפס!
 זה אומר שאם \bar{X} נמצא בטוחו של μ_0 .



שאלה 6: מה נקבע לגבי הרוחב של רוח הסמן של דגנית?

$$U - L = \bar{X} + Z_{0.975} \cdot \frac{\sigma}{\sqrt{n}} - \left(\bar{X} - Z_{0.975} \cdot \frac{\sigma}{\sqrt{n}} \right) = 2 \cdot Z_{0.975} \cdot \frac{\sigma}{\sqrt{n}}$$

- בש- α גדל (ההתפלגות תעשה יותר וזרה) הרוח קטן (רוח הסמן שמכסה 95% קטן מאוד עד שאנחנו ממש בטוחים).
- בש- σ קטן (השינויים הטבעית של מה שאנחנו מודדים קטנה יותר) הרוח קטן (יש פחות מקום לשפק).
- בש- α קטן איזה סיכון לטעות מסווג ראשון קטנה (איז לא מוכן לטעויות), והרוח גדל (כדי שבאמת לא נטענה).

מבחן Z ומבחן T:

מבחן T: באשר השונות לא ידועה.

$$H_0 : \mu = \mu_0$$

$$S_X^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

$$T = \frac{\bar{X} - \mu_0}{S_X / \sqrt{n}} \sim t_{n-1} \text{ ("t with } n-1 \text{ degrees of freedom")}$$

$$\text{Cl: } \bar{X} \pm t_{n-1; 100(1-\alpha/2)} \frac{S_X}{\sqrt{n}}$$

$$H_0 : \mu = \mu_0$$

$$Z = \frac{\bar{X} - \mu_0}{\sigma / \sqrt{n}} \sim N(0, 1)$$

This is the test stemming from CLT.

$$\text{Cl: } \bar{X} \pm Z_{100(1-\alpha/2)} \sigma / \sqrt{n}$$

מבחן Z: כאשר השונות ידועה.

אפשר לבנות מבחן T עבור מדגים מזוגיים (תציפות שבאות בזוגות – זוג תאומים, ניסוי קליני של גבר מול אישה, לפני ואחריו), ואז בצע את המבחן על תוחלת ההפרשים. אפשר גם עבור מדגים לא מזוגיים.
(לא ניתן על המקרה שבו השונות לא שווה במבחן!)

- two independent samples:

$$H_0 : \mu_X = \mu_Y$$

- equal variances assumed

$$S_p^2 = \frac{(n_x-1)S_x^2 + (n_y-1)S_y^2}{n_x+n_y-2}$$

$$T = \frac{\bar{X} - \bar{Y} - (\mu_x - \mu_y)}{S_p \sqrt{\frac{1}{n_x} + \frac{1}{n_y}}} \sim t_{n_x+n_y-2}$$

$$\text{Cl: } (\bar{X} - \bar{Y}) \pm t_{n_x+n_y-2; 100(1-\alpha/2)} \sqrt{\frac{S_p^2}{n_x} + \frac{S_p^2}{n_y}}$$

- equal variances not assumed

$$T = \frac{\bar{X} - \bar{Y} - (\mu_x - \mu_y)}{\sqrt{\frac{S_x^2}{n_x} + \frac{S_y^2}{n_y}}} \sim t_{df'}$$

$$df' = \frac{\left(\frac{S_x^2}{n_x} + \frac{S_y^2}{n_y}\right)^2}{\frac{(S_x^2/n_x)^2}{n_x-1} + \frac{(S_y^2/n_y)^2}{n_y-1}}$$

- paired sampled:

$$d_i = X_i - Y_i$$

$$H_0 : \mu = \mu_d$$

$$\bar{d} = \frac{1}{n} \sum_{i=1}^n d_i$$

$$S_d = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (d_i - \bar{d})^2}$$

$$T = \frac{\bar{d} - \mu_d}{S_d / \sqrt{n}} \sim t_{n-1}$$

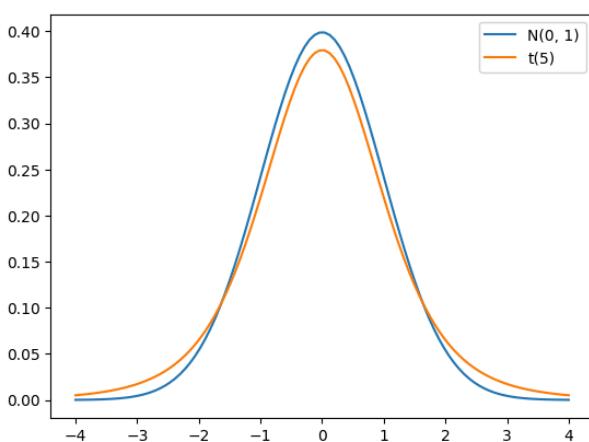
שאלה 7: בחרה לדגנית, נניח שהיא לא מכירה שהשונות ידועה, והיא מחשבת את האומדן: $S_x = 10$. עבשו בניית רווח של 95%.

```
from scipy.stats import t
S_x = sigma # Dganit computed this, make sure you know how
L = sample_mean - t.ppf(0.975, n - 1) * S_x / np.sqrt(n)
U = sample_mean + t.ppf(0.975, n - 1) * S_x / np.sqrt(n)
print(f'95% CI for mu: [{L:.02f}, {U:.02f}]')
```

95% CI for mu: [93.80, 100.20]

```
import matplotlib.pyplot as plt
x = np.linspace(-4, 4, 100)
plt.plot(x, norm.pdf(x), label='N(0, 1)')
plt.plot(x, t.pdf(x, 5), label='t(5)')
plt.legend(loc='best')
plt.show()
```

האם זה מפתיע שרווח הסטן יותר רחב? לא, זה נובע מהתוכנה של התפלגות T. התפלגות T עם 5 דרגות חופש עם זנבות יותר ורחבים.





שאלה 8: מכון פיסיקומטרי דוגם בצורה רנדומלית 50 סטודנטים, כל אחד לוקח את הבחינה לפני הקורס, ולאחריו הקורס. האם לkiemת הקורס מגדילה את הציונים ברמת מובהקות ?1%

$$T = \frac{\bar{d} - \mu_d}{S_d / \sqrt{n}} = \frac{5.08 - 0}{11.64 / \sqrt{50}} \sim t_{50-1}$$

```
d = x_after - x_before
d_bar = d.mean()
S_d = d.std(ddof=1)
print(f'd_bar = {d_bar:.2f}')
print(f'S_d = {S_d:.2f}')

d_bar = 5.08
S_d = 11.64
```

יש כאן שני מקרים מוגדים לכל סטודנט יש ציון לפני ואחריו.

אפשר לחשב הפרש תחת השערת האפס (שם התוחלתו היא 0), ותהייה לנו השערה חד-צדנית (ההפרש גדול יותר).

נגידו: $d_i = X_{\text{after},i} - X_{\text{before},i}$. כאשר $\mu_d = 0; H_0: \mu_d = 0; H_1: \mu_d > 0$.

מבחן בודד על ההפרשים. קודם כל בצורה ידנית. (הפרמטר dof זה מה שמשמעותו מ- t -חדר שנקבל $1 - n$ דרגות חופש).

```
[38] t_obs = d_bar / (S_d / np.sqrt(50))
    print(f't_obs = {t_obs:.2f}'')
```

⤵ t_obs = 3.08

```
[39] p_val = 1 - t.cdf(t_obs, 50-1)
    print(f'P-value = {p_val:.3f}'')
```

⤵ P-value = 0.002

אפשר גם **לчисל בצורה אוטומטית** עם stats:

```
[40] from scipy.stats import ttest_rel # related
    ttest_rel(x_after, x_before, alternative='greater')

⤵ TtestResult(statistic=3.0849093118785116, pvalue=0.0016715225565595937, df=49)
```

Which is the same as:

```
[41] from scipy.stats import ttest_1samp
    ttest_1samp(d, 0, alternative='greater')

⤵ TtestResult(statistic=3.0849093118785116, pvalue=0.0016715225565595937, df=49)
```

שאלה 9: שני מתרגלים מתוכוחים מי מהם טוב יותר. הם בוחרים באופן רנדומלי 40 תלמידים מכל אחת מכיתות התרגול שלהם, ובודקים את תוצאות התלמידים במבחן בסוף הסמסטר. צריך לבדוק האם הממוצעים שונים ברמת מובהקות 5%?

יש כאן מקרים בליינו! נשים לב שהגודל הקבוצות כאן שונה, אפשר להניח שונות זהה. בולם: $H_0: \mu_A = \mu_B; H_1: \mu_A \neq \mu_B$

```
X_A_bar = X_A.mean()
S_A = X_A.std(ddof=1)
n_A = len(X_A)
X_B_bar = X_B.mean()
S_B = X_B.std(ddof=1)
n_B = len(X_B)
print(f'X_A_bar = {X_A_bar:.2f}')
print(f'S_A = {S_A:.2f}')
print(f'n_A = {n_A}')
print()
print(f'X_B_bar = {X_B_bar:.2f}')
print(f'S_B = {S_B:.2f}')
print(f'n_B = {n_B}'')
```

X_A_bar = 75.91
S_A = 11.50
n_A = 23

X_B_bar = 79.06
S_B = 8.88
n_B = 17

$$S_p^2 = \frac{(n_x-1)S_x^2 + (n_y-1)S_y^2}{n_x+n_y-2} = \frac{(23-1)11.5^2 + (17-1)8.9^2}{23+17-2}$$

✓ 0s ⏵ S2p = ((n_A - 1) * (S_A**2) + (n_B - 1) * (S_B **2)) / (n_A + n_B - 2)
S_p = np.sqrt(S2p)
print(f'S_p = {S_p:.2f}'')

⤵ S_p = 10.48

$$T = \frac{\bar{X} - \bar{Y} - (\mu_x - \mu_y)}{S_p \sqrt{\frac{1}{n_x} + \frac{1}{n_y}}} = \frac{75.9 - 79.1 - 0}{10.5 \sqrt{\frac{1}{23} + \frac{1}{17}}} \stackrel{H_0}{\sim} t_{n_x+n_y-2}$$

✓ 0s [45] t_obs = ((X_A_bar - X_B_bar) - 0) / (S_p * np.sqrt(1 / n_A + 1 / n_B))
print(f't_obs = {t_obs:.2f}'')

⤵ t_obs = -0.94

✓ 0s [46] # two-sided p-value of t_obs lower than 0
p_val = 2 * t.cdf(t_obs, n_A + n_B - 2)
print(f'P-value = {p_val:.3f}'')

⤵ P-value = 0.354

```
from scipy.stats import ttest_ind
ttest_ind(X_A, X_B, equal_var=True, alternative='two-sided')
```

TtestResult(statistic=-0.9385695366223592, pvalue=0.35387953336474676, df=38.0)



עוצמת המבחן:

עוצמת המבחן תלויה ב-3 פרמטרים: $\alpha, n, \text{effect size}$. בשיש לנו 3 מתוך ה-4 תמיד אפשר לחשב את החסר. היחיד שלא נחשיר אף פעם הוא α – לא מקובל להגדיל אותו (להיות מוכנים לטעות יתר) כדי להגדיל את עוצמת המבחן.

Once we have 3 of these 4 quantities ($\alpha, n, \text{effect size}, \text{power}$), we can calculate for the missing one. Examples:

- What is the minimum sample size n to achieve a power of 80% with effect size 0.8 at 5% significance level?
- What is the power for a statistical test with 100 random subjects, statistical significance 5% trying to detect an effect size of about 0.6?
- What is the maximum effect size we can detect with 5% significance level, 100 subjects and power 90%?

שאלה 10: נתון $\alpha = 0.05$. דמי רוצה לוודא שהוא תמצא אפלו שינוי קטן מאוד בממוצע זמן המתנה לאוטובוס. מה ה- n המינימלי שנדרש כדי להשיג לפחות 90% עוצמה, אם אכן האוטובוס מגיע אחרי 22 דקות בממוצע במקום 20 דקות?

עבור n גדול מ-CLT נקבל $22 \sim N\left(\mu, \frac{20^2}{n}\right)$, $H_0: \bar{Y} \sim N\left(\mu, \frac{22^2}{n}\right)$. **נשים לב כי לפי CLT הממוצע מתפלג נורמלית עם אותה השונות** (עבור משתנה מעריצי במקור, השונות היא ריבוע התוחלת). קודם צרכן להשיג את C :

$$\begin{aligned} \alpha &= P[\text{reject } H_0 \mid H_0 \text{ true}] = P_{\bar{Y} \sim H_0}(\bar{Y} \geq C) = P_{\mu=20}(\bar{Y} \geq C) = 1 - \Phi\left(\frac{C - 20}{\sqrt{\frac{20}{n}}}\right) = 0.05 \\ \Leftrightarrow C &= 20 + Z_{0.95} \cdot \frac{20}{\sqrt{n}} = 20 + \frac{32.9}{\sqrt{n}} \end{aligned}$$

עבשו נחלץ את n מתוך חישוב עוצמת המבחן תוך הצבת C :

$$\begin{aligned} \pi &= P[\text{reject } H_0 \mid H_1 \text{ true}] = P_{\mu=22}\left[\bar{Y} \geq 20 + \frac{32.9}{\sqrt{n}}\right] = 1 - \Phi\left(\frac{20 + \frac{32.9}{\sqrt{n}} - 22}{\sqrt{\frac{22}{n}}}\right) \geq 0.9 \\ \Leftrightarrow \Phi &\left(\frac{20 + \frac{32.9}{\sqrt{n}} - 22}{\sqrt{\frac{22}{n}}}\right) \leq 0.1 \Leftrightarrow Z_{10} \geq \frac{20 + \frac{32.9}{\sqrt{n}} - 22}{\sqrt{\frac{22}{n}}} \Leftrightarrow n \geq 30.545^2 = 932.99 \end{aligned}$$

```
n_samples = 100000
n = 933
y_means_h0 = np.array([one_sample_mean(n=n) for i in range(n_samples)])
C = np.quantile(y_means_h0, 0.95)
print(f'Critical value above which 5% under H0 with n = {n}: {C:.2f}')
y_means_h1 = np.array([one_sample_mean(scale=22, n=n) for i in range(n_samples)])
print(f'Power under H1 with n = {n}: {np.mean(y_means_h1 > C):.2f}'')
```

Critical value above which 5% under H0 with n = 933: 21.08
Power under H1 with n = 933: 0.90

יש גם את שאלה 11 ושאלת 12 במחברת אבל לא נראה שהן קרייטיות.

3 – מודלים לחיזוי

גרסיה

גרסיה לינארית

מבוא למודלים לחיזוי

מודלים לחיזוי (Predictive Modeling): יש לנו תצפית לחיזוי $\hat{y} = \mathcal{X} \in x$ (בעל k משתנים), וסקול ע' שאנו חשים מהו פונקציה של x . המטרה היא לבנות מודל שיעשה קירוב ליחס זה: $y \approx f(x)$.

- x :特性ות שונות ל- x : predictors, regressors, features, exogenous variable
- y : באופן דומה ע': response, dependent variable, endogenous variable

אם נצליח לבנות מודל איקוטי, זה ימלא שתי מטרות:

1. חיזוי: אם תגיע לתצפית חדשה x תוכל לחזות לה את ערך ה- y המתאים לה, אף על פי שהמודל לא ראה אותה: $f(x) = \hat{y}$.
2. הסקה/הבנה: מודל טוב יאפשר לנו להבין את הקשר בין x , y . אילו משתנים ב- x חשובים כדי להסביר או לחזות את y ? מה אופי התלות של y ב- x (לינארי או מורכב יותר?).

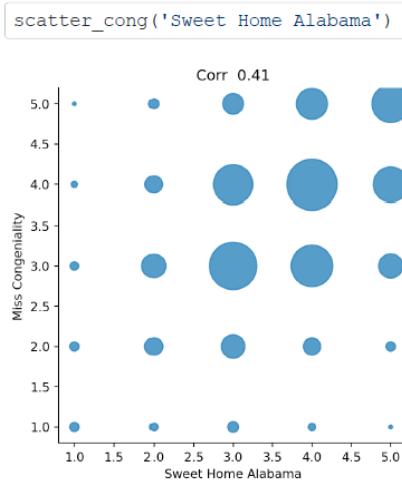
דבר על שני סוגי מודלים לחיזוי:

גרסיה: כאשר $\mathcal{X} \in x$ כמו גובה של תינוק שעומד להיוולד.

- בדוגמה של Netflix, ציוני הסרטים יהיו בוקטו $\mathcal{X} \in x$, כאשר בוקטו נפרד יהי הסרט "אייזו מין שוטרת" ע'. הערכבים של x במקורה הזה לא באמת נמצאים על כל הישר המשני, אלא בסולם 1 עד 5 (יש בהם גם ערכים חסרים). גם y לא מקבל כל ערך, אלא רק מספרים בין 1 ל-5. מודל טוב, קיבל את הציונים הקיימים של משתמש, כולל הסרטים שהוא לא ראה, ויתן חיזוי לדירוג של הסרט "אייזו מין שוטרת", הבci קרוב לדירוג y של אותו משתמש.
- פרסום אינטרנטני: גולשת מגעה לאתר, וצריך להחליט האם להראות לה פרסום, ואיזו. ע' היא השורה התחתונה, במא בסך תוצאות על המוצר פרסום. x יכול להיות מאוד מגוון: היסטוריית גישה, המיקום שלה, מותי היא גולשת ואיפיל Cookies.

סינוג: כאשר $\mathcal{G} \in y$ שיר לסת של קטגוריות, רוחצים לסוג לפי קטגוריה.

- דוגמת הציורים: $\mathcal{X} \times K \times K \times 3 \in x$ התמונה עצמה, והתיוג הוא $\{RE, MI\} \in y$. מודל טוב יספק ערך שקרוב ל-1 עבור ציורים MI, וערך קרוב ל-0 עבור ציורים RE. למשל, נבדוק מה הרמה הממוצעת של פיקסל אדום בתמונה, ומסתכל על סף כלשהו – עד אילו יחזא 0, ומעלהו יחזא 1. סביר להניח שהמודול לא טוב, ולא יפריד היטב בין סוגי הציורים.
- GWAS (genome-wide association studies): מנסים לבדוק גנים שאחראים על מחלות.
- $\{sick, healthy\} \in y$ והתצפית שלהם היא $\{0, 1, 2\} \in x$ (הגנים שלהם). זה יכול להגיעה למיליאון מקומות על ברומוזום, שבהם יכול להיות לאדם 2-0-0 עותקים של מוטציות בגין.
- מעכין אותנו חיזוי על אדם חדש בהתאם לגנים שלו – האט הוא בסיכון למחלת.
- מעכין אותנו גם לראות אילו גנים אחראים למחלת, אילו משתנים ב- x משפיעים על y .
- חיזוי ספאם: בהינתן מייל x (השולח, התוכן וכו'), מסווג $\{OK, spam\} \in y$. המטרה היא לחזות האם מייל הוא ספאם או שער להעבר אותו לתיקית הספאם או לא.



בנייה מודל בסיסי: נחזור לדוגמה של Netflix ובבנה מודל פשוט. נחזה את הדירוג של צופה ב-"אייזו מין שוטרת" באמצעות סרט שדומה לו. למשל ראיינו שהסרט של "Sweet Home Alabama" דומים מאד לסרט המקורי מחפשים, שניהם באוטו סגןון. ניתן לראות את גרפף הפיזור שמחиш את הקשר חזק יחסית בין שני הסרטים. אנשים נתונים לדרג גבוה את שני הסרטים, המתאים חיווי ורוחוק יחסית מאפס.

מודל פשוט אחר שדיברנו עליו בקשרי PCA, הוא $h(x)$ של צופה ב-PC1 (משמעותו הרכבה שwonot), הוא מנבא טוב לצוין של הסרט המבוקש. הצוין של כל צופה גבוה יותר בכל שהוא פחות מסכום עם הדירוג הממוצע של הסרטים. הדירוגים בממוצע הם די גבוהים, ולכן נצפה שככל שצווין של צופה גבוה יותר ב-PC1, כך הוא שונא יותר סרטים וייתן צוין נמוך לסרט המבוקש. זה אכן מה שמתקיים: יחס יודד, ומתאים שלילי לא מבוטל.



אימון והערכת השגיאה: בכל בעיה של מודל לחיזוי, יהיה לנו **training set** שנסמך $T_r = \{(x_i, y_i)\} = (X_{n \times p}, Y_{n \times 1})$ וכן IID זוגות התצפיות בלתי תלויים, נציגים בצורה בלתי תליה מהתפלגות משותפת $P_{X,Y}$. בסופו של דבר, הפלט שלנו יהיה מודל לחיזוי \hat{f} שمبוסס על מודם הלמידה. המטרה היא שכשתגיע לתצפית חדשה נחזה $\hat{y}_0 = \hat{f}(x_0)$.

$$L(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{if } y \neq \hat{y} \end{cases}$$

איך נמדד את הביצועים? נגיד פונקציית הפסד $L(y, \hat{y})$, ותוחלת הכמות הזאת תחת התצפיות שהמודל לא ראה, היא הכמות שהיינו רצים לעשות לה מינימיציה.

- More complex approach: penalize different types of error differently

$$L(y, \hat{y}) = \begin{cases} 0 & \text{if } y = \hat{y} \\ 1 & \text{if } y = 0, \hat{y} = 1 \\ 10 & \text{if } y = 1, \hat{y} = 0 \end{cases}$$

הינו רצים שפונקציה זו תنبא כמה אנחנו מפסידים בשהתצפיות האמיתית היא y ואנחנו חוזים בעצם \hat{y} . יש מספר דרכים להגדיר פונקציית הפסד, לתת מחיר של 1 על טעות, ו- 0 על דיווק, או לחת מרוחה יותר גדול של קטגוריות.

- Simple example for regression: *squared error loss*

בביעות וגרסיה מודד מקובל הוא הטעות הריבועית (MSE).

$$L(y, \hat{y}) = (y - \hat{y})^2.$$

או אפשר באמצעות חישוב את תוחלת הפסד תחת התפלגות התצפיות שהמודל לא ראה, בלי הערכה טוביה של התפלגות הדאטא. לכן, נהוג לחשב את הפסד האמפירי על סט נתונים נפרד שהוא **test set**: $Te = \{(x_{n+1}, y_{n+1}), (x_{n+m}, y_{n+m}), \dots\}$. אז הטעות שנרצה ל愍ור היא **ממוצע הפסד על פנוי ה-test set** זהה שהמודל לא ראה: $Err = \frac{1}{m} \sum_{i=n+1}^{n+m} L(y_i, \hat{f}(x_i))$. נהוג לחשב את שורש הטעות הריבועית:

$$RMSE = \sqrt{Err} = \sqrt{\frac{1}{m} \sum_{i=n+1}^{n+m} (y_i - \hat{f}(x_i))^2}$$

לרוב נקבל סט אחד של נתונים, בשאייפה עם מספיק תצפיות כדי לחלק אותם בצורה אקראית למודם train ומודם test, לרובה 80 אחוז ו-20 אחוז.

Let's divide our Netflix data 80-20:

```

1 X = ratings.values
2 Y = miss_cong.values[:, 0]
3 n = X.shape[0]
4 tr_size = int(0.8 * n)
5 te_size = n - tr_size
6 tr_ind = np.random.choice(range(n), tr_size, replace=False)
7 Xtr = X[tr_ind, :]
8 Xte = np.delete(X, tr_ind, axis=0)
9 Ytr = Y[tr_ind]
10 Yte = np.delete(Y, tr_ind)
11
12 print(f"No. of train rows: {Xtr.shape[0]}, no. train of cols: {Xtr.shape[1]}")
13 print(f"No. of test rows: {Xte.shape[0]}, no. test of cols: {Xte.shape[1]}")
14 print(f"no. of obs in train y: {Ytr.shape[0]}")
15 print(f"no. of obs in test y: {Yte.shape[0]}")

No. of train rows: 8000, no. train of cols: 99
No. of test rows: 2000, no. test of cols: 99
no. of obs in train y: 8000
no. of obs in test y: 2000

```



רגסיה לינארית:

בහינתן וקטור צפיפות $\hat{y} \in \mathbb{R}$ לסקלר $x \in \mathbb{R}$, נבנה מודל לינארי: $\hat{y}(x) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \dots + \hat{\beta}_p x_p$. ננסה למצוא מקדים שambilאים את המודל הכו קרוב ל- y האמייתי על בסיס ה- X -Tr שלנו: $\hat{y}(x_i) \approx y_i, i \in [n]$. למשל, נרצה למינימיזציה את סכום השגיאות הריבועיות בין התחזית של המודל לבין ה- y האמייתי, נקרא לכך RSS (residual sum of squares):

$$RSS(\beta_0, \dots, \beta_p) = \sum_{i=1}^n \left(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \dots + \hat{\beta}_p x_{ip}) \right)^2 = \|Y - X_{n \times (p+1)}\beta\|^2$$

ניתן גם לבתוב מטריציונית, מה שיכל להאיץ את המימוש – הנורמה הריבועית של הווקטור \hat{y} פחות מטריצה X מוכפלת בווקטור המקדים שלנו. נשים לב שנוספה עמודה למטריצה הדטא X – זאת על מנת לאפשר את החותר β שיש לנו במודל. זהו וקטור שכלו $(1, \dots, 1)$. המודל הזה נקרא OLS (ordinary least squares), מודל הרוגסיה הלינארית הקלאסי.

The `statsmodels` approach:

```
1 sweet_home_idx = 9
2 X_sweet_tr = Xtr[:, [sweet_home_idx]]
3
4 import statsmodels.api as sm
5
6 X_sweet_tr1 = sm.add_constant(X_sweet_tr)
7 model = sm.OLS(Ytr, X_sweet_tr1)
8 model = model.fit()
9 print(f'y = {model.params[0]:.2f} + {model.params[1]:.2f}*x1')
y = 2.13 + 0.40*x1
```

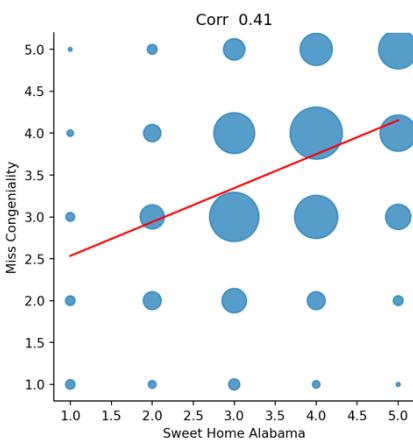
The `SKlearn` approach:

```
1 from sklearn.linear_model import LinearRegression
2
3 model = LinearRegression()
4 model.fit(X_sweet_tr, Ytr)
5 print(f'y = {model.intercept_:.2f} + {model.coef_[0]:.2f}*x1')
y = 2.13 + 0.40*x1
```

ניקח רק את המשתנה היחיד x_1 שהוא ציון הסרט "Sweet Home Alabama"

באמצעות הספרייה `statsmodels`: נשתמש בمدגם הלמידה Z , נוסיף את עמודת החותר באמצעות `add_constant()`. נאתחל מחלוקת בשם OLS, והרגסיה קוית במתודה `fit()`. בעת, באובייקט יש את `params` ובתוכו β_i .

באמצעות הספרייה `sklearn` (בלתי אוירונטיקה סטטיסטיות):
באו המחלוקת היא `LinearRegression`, ואז נחלץ את המקדים באמצעות `intercept_-coef`. בירית המandal היא להוסיף את החותר למודל, אין צורך לעשות את זה.
ידנית כמו ב-`statsmodels`.



המודל שקיבלנו הוא משווה את κ ישר פשוט. נבנה X מלאכותי עם הדירוגים 1 עד 5, ונבקש מהמודל לחזות את Y . נציג את תרשימים הפיזור ונוסיף את הקוו ה ישיר שחזזה המודל. הקוו נראה די מתאים לנ נתונים, גם אם הוא פשוט. הדירוגים מתחילה ב-1 ולבן לא נוכל לראות מה קורה בנקודה 0 (החותר של 2.13).

בעת נבחן את המודל על ה-`test set` ומדד ה-`RMSE`. נחלץ את `sweet home alabama` מה-`test set`, ונראה לא-`predict` כדי לקבל את החיזוי ל- y .

- נגידו את `test_RMSE` שיתקבל אם פשוט נחזה את הממוצע של y שראינו במדגם הלמידה, זה יתנו לנו `baseline`, האם המודל שלנו שיפר במשהו. במודל הכו פשוט נקבל **0.97**, כמעט 1.
- נגידו את `test_RMSE_imovie` שהוא `RMSE` של המודל שלנו, סרט אחד בלבד. כאן נקבל ה-**0.88**.

בעת נשתמש בכל 14 הסרטים שלגביהם אין לנו תציפות חסרות. קרייה ל-`summary` תחתן לנו פلت יותר מפורט על הרוגסיה. נתמקד בשדה של `coef`. אפשר לראות שעבור החותר המקדם הוא **0.4** ("ציון הבסיס ל'איזו מין שוטרת"), ועוד לכל סרט ניתן לראות את ההשפעה שלו. למשל: `forest_gump` מקבל מקדם שלילי.

נרצה לבדוק את מדד `RMSE` על מדגם ה-`test`, ונקבל עוד הफחתה ממשמעותית, עם 14 סרטים אנחנו בבר-**0.81**. בשחמודל היה סרט אחד, היה ברור שהוא קי-ישר. איך נראה המודל שלנו עבשו? אם נכלול לשני משתנים – המודל יהיה בעצם מישור. ביוטר מושני משתנים נראה לו מישור ק-מידדי.

לצורך התרגיל במקום ערכיהם חסרים נציג 0 – אולי מי שלא דירג סרט מסוים, ממש לא אהב אותו (בחר שזה לא הדבר הכי חכם שכן לעשות בכך, ועל טיפול בערכים חסרים אפשר להעמיך בזמן אחר). בכל זאת, אולי זה יועל לנו כפי שמתבטה מה-`RMSE`? מסתבר שכן, בדוגמה הדעת – אם נשתמש בכל 99 הסרטים בהצתת 0 במקום ערכים חסרים, הוא יורד ל-**0.78**.

גרסיה לינארית – התאמת המודל:

פרשנות אלגברית: במקרה הפשטוט $1 = d$, יש לנו את הסרט sweet home alabama ועוד חותך. אנחנו רוצים לבצע מינימיזציה ל-

$$\min_{\beta_0, \beta_1} \sum_{i=1}^n \left(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i) \right)^2.$$

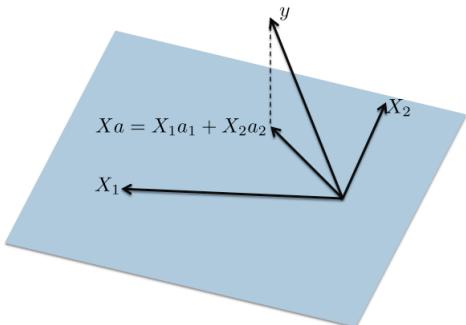
אפשר לגזר לפיקטורי β_1 , להשוות לאפס, ולמצאו את הפטרון, ואז אותו דבר עבור β_0 . לבסוף, כדי לוודא שגם אכן נקודת מינימום, צריך להביט על מטריצת הנגזרות השנייה ולוודא שהיא חיובית (PD). האומד $\hat{\beta}_1$ נראה מוכר, דומה למתרם המדגם של פירסון.

אם נרצה להכליל: $\|Y - X\beta\|^2 = \min_{\beta} \|Y - X\beta\|^2$ נכתוב בכתב וקטוורי. אם נגזר ונשווה לאפס נקבל מה שמכונה המשוואות הנורמליות: $0 = \nabla_{\beta} \text{RSS}(\beta) = \min_{\beta} \|Y - X\beta\|^2$. אפשר להראות שבסופו של דבר מדובר בספט של משוואות בצורה ה затה:

$$\frac{\partial \text{RSS}(\beta)}{\partial \beta_j} = \sum_{i=1}^n x_{ij} \left(y_i - (\beta_0 + \sum_{k=1}^p x_{ik} \beta_k) \right) = 0, \quad j = 0, \dots, p$$

גע לפתרון הריבועים הפחותים (מטריצת הנגזרות השנייה היא PD ולכן מינימום):

$$X^T X \beta = X^T Y \Rightarrow \hat{\beta} = (X^T X)^{-1} X^T Y.$$



פרשנות גיאומטרית: ברגע במרחב הלמידה, העמודות של X פורשות מרחב ו- y הוא גם וקטור נוסף. אנחנו רוצים, למצוא צ"ל של העמודות האלה של X , וזה המשמעות של הביטוי $X\beta$. בשאנחנו עושים מינימיזציה ל-RSS, אנחנו מחפשים את הצ"ל של עמודות X שהוא הקירוב הטוב ביותר ל- y .

הפתרון שלו הוא הווקטור הכי קרוב במרחב הנפרש לוקטור y (שכנראה לא נמצא במרחב זהה). הוקטור \hat{y} הוא ההטלה האורתוגונלית אל המרחב הנפרש על ידי עמודות X .

מבט סטטיסטי: עד כה, הייתה לנו רק אלגברת לינארית וחשבון דיפרנציאלי. מאיפה באים ערכי t-ה-value-k שראינו בפלט הגרסיה? עד עכשיו לא נהננו שום מודל שהוא הקשר האמיוטי בין x ל- y או התפלגות, לא היו תנאים על LSQ. בעת, כן נניח שיש מודל (הנחה די מלחירה), שמתקיים $\epsilon \sim N(0, \sigma^2)$; $y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon$. התייחסות זו לא תשנה את פתרון הריבועים הפחותים, בלבד $\hat{y} = (X^T X)^{-1} X^T Y$ כמו קודם. מה התווסף לנו?

1. $x^T \beta = E[y|x]$ היא פונקציה לינארית של x .
2. $(E[y|x] - y)$ הטעות היא גם משתנה מקורי שמתפלג נורמלי: $\epsilon \sim N(0, \sigma^2)$.

תחת הנחת המודל הסטטיסטי, גם $\hat{\beta}$ יהיה וקטור של משתנים מקרים עם התפלגות. אפשר לא רק להגיד מהו, אלא בכלל לבצע הסקה סטטיסטית על $\hat{\beta}$: נוכל להגיד האם הוא שונה בצורה מובהקת מ一封ס, האם הוא באמת חשוב במידול של y או לא.

נרצה למצוא את התפלגות של האומד $\hat{\beta}$, שהוא צ"ל של משתנים נורמליים, ולכן מתפלג נורמלית. ידוע לנו עד כה כי: $E[Y] = X\beta$; $Cov(Y) = \sigma^2 I_n$; $\hat{\beta} = (X^T X)^{-1} X^T Y$. נחשב ונקבל:

- נחשב את התוחלת ונקבל $E[\hat{\beta}] = \beta$, כלומר מדבר באומד חסר הטיה ל- β .
- נחשב את השונות ונקבל $Cov(\hat{\beta}) = \sigma^2 (X^T X)^{-1}$.

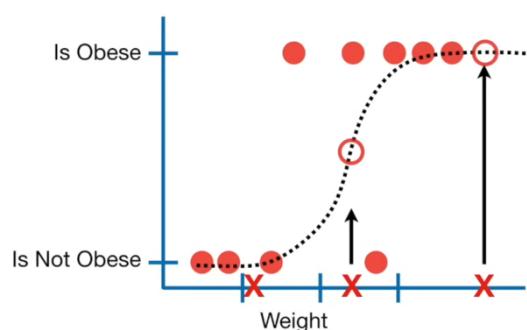
$$E(\hat{\beta}) \stackrel{(c)}{=} (X^T X)^{-1} X^T E(Y) \stackrel{(a)}{=} (X^T X)^{-1} X^T X \beta = \beta.$$

$$Cov(\hat{\beta}) \stackrel{(c)}{=} (X^T X)^{-1} X^T Cov(Y) X (X^T X)^{-1} \stackrel{(b)}{=} \sigma^2 (X^T X)^{-1} (X^T X) (X^T X)^{-1} = \sigma^2 (X^T X)^{-1}.$$

ביצוע הסקה סטטיסטית: באופן שלו ניתן לראות, שככל $\beta_j \sim N_{jj} \left(\beta_j, \sigma^2 (X^T X)^{-1} \right)$. זה אומר שאפשר לבצע בדיקת השערות על כל אחד בזהה. השערת האפס: $0 = \beta_j - H_0$, וההשערה החלופית: $0 \neq \beta_j$; והוא חשוב במודל. תחת השערת האפס, ההסתפלגות היא נורמלית עם אותן שונות אבל עם תוחלת אפס. אם הימנו יודעים את σ^2 הימנו יכולים לגשת למבחן Z, אבל אנחנו לא יודעים אותה, ולכן נגיע למבחן T.

הערות סיכום:

- כדי לקבל את אומד הריבועים הפחותים אנחנו עושים מינימיזציה של RSS, וזה יביא לנו את הצ"ל הטוב ביותר של $\hat{\beta}$ כדי להתקרב לוקטור β . הפתרון אלגברי לחילוין, והפירוש הגאומטרי היא הטללה על המרחב הנפרש על ידי X .
- המודל הסטטיסטי מאפשר לנו לבצע הסקה סטטיסטית על המקדים, ולתת אמירות כמו "המשתנה חשוב" כי הוא "שונה מאפס" ברמת מובהקות מסוימת. זה הכליל אליו החשוב ביותר מבחינה מסורתית שיש למடע נתונים.
- תחת המודל הסטטיסטי – האומד $\hat{\beta}$ הוא אומד חסר הטיה. זה אומר שגם \hat{y} בהינתן x הן אומד חסר הטיה למודל הלינארי: $[x|y] = E[\hat{y}] = x^T \beta$.
- ההירוך את מה שאנו יודעים על y בהינתן שראינו את כל המשתנים ב- x . ננסה לחזות גם במודלים אחרים את $[x|y]$, אבל לא נהייה מוחייבים לגבי הילינארי הנקשה הזה.
- אפשר להגע לפתרון בעיית הריבועים הפחותים גם בדרך אחרת, תחת אותה הנחה, y הוא משתנה מקרי נורמלי, ויש לו פונקציית צפיפות מוכרת. אפשר לנסות למצסם במתוות אחרות מה-RSS, שנראית נראות (likelihood) והיא מכפלה הצפיפות. זה היה מביא אותנו לפתרון ריבועים פחותים ל- $\hat{\beta}$.



邏輯יסטייה: התחלנו ברגסיה בש- y הוא ממשי, נבעור לסיווג כאשר עטגוריאלי. נתמך בבעיה הפושא של **סיווג ביןארי**. דוגמאות יכולות להיות צויר MI מול RE אדם חוליה או בריא, لكنות או לא לכנים וכו'. כמו קודם, יש לנו $(Y, X) = Tr(X) = Te - Tm$. האם ניתן להשתמש בכל התוצאות שלנו מרגסיה לינארית עבור עשוואה קטגוריאלי עם 2 קטגוריות? התשובה היא שכן, פשוט צריך להפוך אותן לנומרו. נגידו ש-MI זה "1", RE זה "0" והנה יש לנו ע"מ ממשי וניתן לבצע OLS.

איפה השגיאה? ה-mdom לאطبعי לבעה!

- זכור שרגסיה לינארית אנחנו מודלים את התוחלת המותנית $[x|y] = \text{מה שלמדנו על } y \text{ במשמעות למידה על משתנים אחרים } (x)$. מהי התוחלת של משתנה שיש לו 2 תוצאות (משתנה ברנולי)? זו ההסתברות שהוא יקבל 1 (נניח צויר MI) בהינתן שראינו את הפיקסלם בתמונה. זה מצין, אבל הסתברות היא בטווח בין 0 ל-1, ואין שום אילוץ ברגסיה לינארית לקבל תוצאות בין 0 ל-1! נוכל לקבל תוצאות קטנות מ-0, גדולות מ-1...
- האם סביר להניח ש- y הוא צ"ל של משתנים ועוד רעש נורמלי **בלתי תלוי**? לא, כי y מקבל ערכים בין 0 ל-1.

רגסיה לוגיסטיבית: צריכים גישה אחרת – לא מודלים את y עצמה במודל לינארי, אלא את הכמות הבאה:

$$\log \frac{P[y=1|x]}{P[y=0|x]} = \log \frac{P[y=1|x]}{1-P[y=1|x]} = \text{logit}(P[y=1|x]) = x^T \beta$$

במוקם למدل את התוחלת $[x|y] = \text{ancocko מודלים פונקציה של } \beta$. הכמות שאנו מודלים היא בטוחה ($\infty, -\infty$) בغالל ההסתפלגות הנורמלית, ולכן כל חיזוי שלנו יהיה בתחום לוגיטי, כי זה גורר $P[y=1|x] \leq P[y=0|x] \leq 0$. אם יש לנו את המקדים ונרצה לקבל בחזזה את ההסתברות החזויה (הכמות בין 0 ל-1), הפונקציה ההופכית נראה כז:

$$P[y=1|x] = \frac{e^{x^T \beta}}{1 + e^{x^T \beta}}, \quad P[y=0|x] = 1 - P[y=1|x] = \frac{1}{1 + e^{x^T \beta}}$$

air מוצאים את המקדים ברגסיה לוגיסטיבית? נבצע אמידת נראות מקסימלית.

$$L(\beta|X, y) = \prod_{i=1}^n P(y_i|x_i; \beta) = \prod_{i=1}^n P(y_i = 1|x_i; \beta)^{y_i} P(y_i = 0|x_i; \beta)^{1-y_i}$$

פונקציית הנראות שלנו היא פונקציה של β בהינתן הדטאנו-trainning, ומסומנת לרוב ב- L . במקורה הבודד כמו לפנינו, y הוא בעצם משתנה ברנולי, מדובר במקרה הסתברויות במדגם תחת המודל. מאחר ש- y מקבל ערכים 0 או 1, ניתן לרשום את הביטוי כך. נציב ונגיע לביטוי מפורש.

$$\max_{\beta} \prod_{i=1}^n \left(\frac{\exp(x_i^T \beta)}{1 + \exp(x_i^T \beta)} \right)^{y_i} \left(\frac{1}{1 + \exp(x_i^T \beta)} \right)^{1-y_i}$$



בשtag'gi צפיפות חדשה מודגמת test , נוכל לחזות את ההסתברות שהיא 1 באמצעות הצבה בנוסחה של הפונקציה ההופכית. אם נרצה חיזוי סופי, האם $y = 1$ או 0 , אפשר להשוות את ההסתברות הנחוצה לסוף מסום (0.5).

$$P(\widehat{y=1}|x) = \frac{\exp(x^T \hat{\beta})}{1 + \exp(x^T \hat{\beta})} \Rightarrow \hat{y} = \begin{cases} 1 & \text{if } P(\widehat{y=1}|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

נשים לב כי הפונקציה הלוגיסטיבית נכתבת כך באופן שקווי: $\sigma(t) = \frac{e^t}{e^{t+1}} = \frac{1}{1+e^{-t}}$.

הערות:

- למה אין ביטוי מפורש ל- $\hat{\beta}$? הסיבה היא שאין כזה! פונקציית הנראות אמנים מפורשת וקמורה, אבל אין לה פתרון סגור. לכן משתמשים בשיטות אופטימיזציה כמו ניוטון-רפסון כדי למצוא את המקבדים בזורה איטרטיבית.
- יחס הסיכויים שאנו חנו מודלים הוא בעצם ה- $\log(\text{odds}) = \log\left(\frac{p}{1-p}\right)$. הפרשנות של מקדמי β - β ברגסיה לוגיסטיבית הרבה פחות פשוטה מאשר ברגסיה לינארית. המשמעות של עלייה של 1 ב- β פירושה עלייה של p ב- odds .

:SAHeart

```

1 saheart_X=pd.get_dummies(saheart.iloc[:, :9]).iloc[:, :9]
2 saheart_y=saheart.iloc[:, 9]
3
4 from sklearn.model_selection import train_test_split
5
6 Xtr, Xte, Ytr, Yte = train_test_split(saheart_X, saheart_y, test_size=0.2, random_s
7
8 print(f'No. of train rows: {Xtr.shape[0]}, no. train of cols: {Xtr.shape[1]}')
9 print(f'No. of test rows: {Xte.shape[0]}, no. test of cols: {Xte.shape[1]}')
10 print(f'no. of obs in train y: {Ytr.shape[0]}' )
11 print(f'no. of obs in test y: {Yte.shape[0]}' )

No. of train rows: 369, no. train of cols: 9
No. of test rows: 93, no. test of cols: 9
no. of obs in train y: 369
no. of obs in test y: 93

```

בנתונים שלפנינו יש 462 גברים מדרום אפריקה. יש לנו מידע רפואי עליהם, והמשתנה שעניינו אותנו הוא chd (האם لكו במחלת לב או לא). נגדיר את מטריצת ה-X שלנו ואת ה-y. יש לנו משתנה קטגוריאלי שמצויר טיפול מיוחד עם הפונקציה `pd.get_dummies()`. נחלק את הדאטאסט ל-set `train`, `set` באמצעות הפונקציה המקובלת `.train_test_split`

באמציאות `statsmodels`, ביעזר במחלקה `Logit`. בפלט נקבל תוצאות של מבחני Z. לכל מקדם יש טעות תקן שנאה, וכן נסתבל על ערך ה-Z. משתמשים עם ערכיהם הם `age` למשל – לכל שנת חיים נוספת, המודל מוסיף בmoths ל- \log odds. משתנה עם מקדם שלילי גדול הוא `famhist_Absent` – האם אין היסטורי של מחלת לב. אם אין, יורדת כמות של \log odds 0.8.

באמציאות `sklearn`, ניעזר במחלקה `LogisticRegression` – ברגיל, יותר מוכoon ML ופחות סטטיסטיקה וכן הפלט שלו מוגבל.

```

import statsmodels.api as sm

model = sm.Logit(Ytr, sm.add_constant(Xtr))
model = model.fit()

print(model.summary())

```

```

1 from sklearn.metrics import confusion_matrix
2
3 p_hat_te = model.predict_proba(Xte)[:, 1]
4 y_hat_te = p_hat_te > 0.5
5 conf = confusion_matrix(Yte, y_hat_te)
6
7 pd.DataFrame(
8     confusion_matrix(Yte, y_hat_te),
9     index=['true:no', 'true:yes'],
10    columns=['pred:no', 'pred:yes']
11 )

```

	pred:no	pred:yes
true:no	52	7
true:yes	15	19

```

acc = np.mean(Yte == y_hat_te)
err = np.mean(Yte != y_hat_te)
print(f'Accuracy: {acc:.2f}, Misclassification loss: {err:.2f}')

```

Accuracy: 0.76, Misclassification loss: 0.24

```

from sklearn.linear_model import LogisticRegression
model = LogisticRegression(solver='lbfgs', max_iter=10000)
model.fit(Xtr, Ytr)

print('intercept:', model.intercept_)
print('coef:', model.coef_)

intercept: [-5.37683503]
coef: [[ 0.0080158   0.0557117   0.16737956   0.0271416   0.0410251  -0.08127558
         0.00199583   0.04753859  -0.76405442]]

```

אין מכמתים את הביצועים של המודול על מודגמ ה- test ? על זה לא דיברנו. אפשר לבדוק את הנראות שאותה מקסמונה, אבל זה לא יגיד הרבה ברגע. מקובל יותר למשל לחזות את ההסתברויות \hat{P} באמציאות הנוסחה שראינו. את החיזוי הזה נקבעים לתוך `confusion matrix` שומרה לנו מתוך הפייצנטים שהם לא חולים – כמה המודול חזה שהם כן חולים, וכמה לא. כנ"ל עבור החלומים.

אפשר לחשב על מודדים אינטואיטיביים של אחוז דיווק, ואחוז שגיאה. מדובר על אחוז התוצאות מה-set test שהמודול צדק בהן, והאחוז שהוא טעה בהן.

המודדים האלה יכולים להיות בעייתיים! נניח שש-99% מהפייצנטים היו בראים, ורק 1% חולמים. מה אם קיבל מודל שחזזה שככל הפיצנטים הם בראים? אחוז ההצלחה שלו יהיה 99%! חyb להיות מודד טוב יותר שיעיד על כך שזה לא מודל נבדר, ואפילו די גרוע.



אבלואציה של מודלים לסיווג:

אנחנו זקוקים למדדים טובים יותר. לטעויות שונות יכול להיות משקל שונה ולהגיד לו שהוא בריא – זו ממשמעות אחרת לגמרי מה להגיד לאדם שהוא חולה. נסמן את הב�ויות בטבלת **confusion matrix** בצורה זו:

Pred			
Real	Pos	Neg	Total
Pos	TP	FN	P
Neg	FP	TN	N
Total	\hat{P}	\hat{N}	m

- Real: מספר הדוגמאות החיוביות ("חולים") באופן שוי נסמן ב-P.
- מספר הדוגמאות השילוליות ("בריאים") נסמן ב-N.
- Pred: מספר החזויים החיוביים נסמן ב- \hat{P} , ואת השילולים ב- \hat{N} .
- אם הממציאות היא "חולה":
 - אם המודל חזה "חולה" TP זה נכון. זה נכון.
 - אם המודל חזה "בריא" FN זה טעות. זו טעות.
 - אם הממציאות היא "בריא":
 - אם המודל חזה "חולה" FP זה טעות. זו טעות.
 - אם המודל חזה "בריא" TN זה נכון. זה נכון.

בעת נוכל להסתכל על מדדים הרבה יותר ספציפיים:

- $P(Correct) = \frac{TN+TP}{P+N} = \frac{TN+TP}{m}$: Accuracy (דיוק)
- $P(\\text{Error}) = \frac{FN+FP}{m}$: Prediction error (טעות ניבוי)
- $P(True + | Pred +) = \frac{TP}{TP+FP} = \frac{TP}{\\hat{P}}$: (positive predictive value) Precision+ (נשאף שווייה קרוב ל-1) (נשאף שווייה קרוב ל-1) (precission+)
 - בהינתן שחייתי "חולה", מה הסיכוי שאכן האדם "חולה" במציאות = כמה מהחיזויים שלו נכונים?
 - יכולה להיות לקוחה שאין לה בעיה לפפס חולים, אבל בשמודל מסמן חולים, הוא חייבות להיות צודקת. הוא לא יכולה להעניק טיפול קשה זהה לבריאים. בגין נרצה למקסם את ה-precision+.
- $P(Pred + | True +) = \frac{TP}{TP+FN} = \frac{TP}{P}$: (sensitivity, true positive rate) Recall+ (נשאף שווייה קרוב ל-1) (Recall+)
 - בהינתן שאדם "חולה" במציאות, מה הסיכוי שהמודל חזה "חולה" = כמה מהדוגמאות הנכונות המודל מבסה?
 - יכול להיות לקוח שפיטוט לא מסוגל לפפס אף חולה, וידרש recall בכמה שייתר גבוהה.
- $P(Pred + | True -) = \frac{FP}{FP+TN} = \frac{FP}{N}$: False positive rate (נשאף שווייה קטן) (FPR)
 - (ניסיון לאזן בין precision ו-recall) $F_1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}$: Harmonic mean of precision and recall (precision/recall)

כדי לקבל את המדדים הללו, אפשר לבקש מ-`sklearn.confusion_matrix()`. נחשב כמה מהם על הדאטה הבא:

```
from sklearn.metrics import classification_report
print(classification_report(Yte, y_hat_te))
precision    recall   f1-score   support
0            0.78     0.88     0.83      59
1            0.73     0.56     0.63      34
accuracy                           0.76      93
macro avg       0.75     0.72     0.73      93
weighted avg    0.76     0.76     0.76      93
```

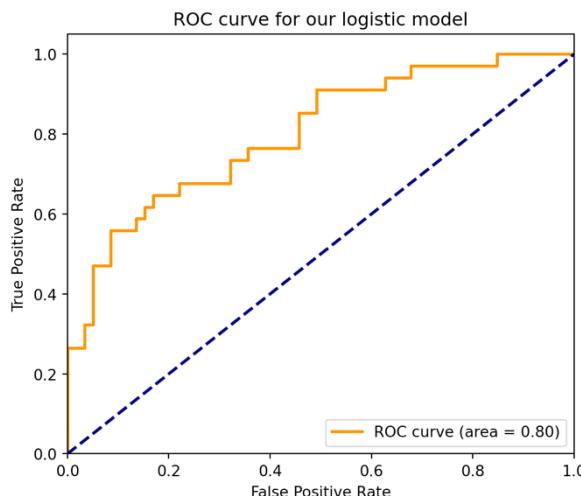
$Precision = \frac{TP}{\hat{P}} = \frac{19}{26} = 73\%$	•
$Recall = \frac{TP}{P} = \frac{19}{34} = 56\%$	•
<hr/>	
pred: no	pred: yes
<hr/>	
true: no	52
	7
<hr/>	
true: yes	15
	19

All is still based on that cutoff, 0.5!

באופן כללי, יכולות להיות לנו מטרות שונות במודל סיוג (המטרה משפיעה על המדריך לטיב המודל):

1. פשוט לחזות נכון: לכן פשוט מסתכלים על precision/recall.
2. לדאוג שההסתברויות הנחותו יהיו מוכילות כמה שייתר עם ההסתברויות התאורטיות: לכן נדוח על loss שהשתמשנו בו.
3. דירוג התצפיות מבחן הסיכוי שהן 1: הפ齊ינט חולה. מתייחסים להסתברויות כ-score כלל שידרג נכון את התצפיות שלו: איך כדאי להסתכל על ביצועי המודל? עקומת ROC (receiver operating characteristic).

עקומת ROC: לא נסתפק ב-cutoff cutoff דיפולטיבי של 0.5 כמו שהוא, אולי ה-score שהוצאים אייננו בדיק הסתברות. נשנה אותו בצדדים קבועים, ובכל צעד נמדד את ה- TPR (true positive rate) ואת ה- FPR (false positive rate). **עקומה זו מצירת את ה-TPR מול FPR לכל cutoff אפשרי.** מודל מושלם, יימצא cutoff שעבורו ה- FPR קרוב ל-0 וה- TPR קרוב ל-1. גם אם ההסתברות הנחותה מגיעה ממודול שיכל לא אמר לו להוציא הסתברויות והיא לא מכילה, או לא הסתברות בכלל – הדירוג עצמו עדין יכול להיות מצוין! וזה היתרון הגדול של גישה זאת.



כർ נראית עוקמת ה-ROC על מבחן-h-test שלנו, עבר מודל רגסיה לוגיסטי שמנסה לחזות האם פציינט יחלה במחלת לב או לא. אנחנו רואים עוקמה יפה של TPR מול FPR. אפשר לראות את trade-off הנקודה האופטימלית שומרת על TPR כמה שיטור גבוה, ו-N FPR כמה שיטור נמוך, אז זו תהיה הנקודה הביקי קרובה לכך השמאלי העליון של הגраф. במודל מושלם, ימצא סף שmagiu עד נקודה זו.

במודל מושלם השטח תחת העוקמה יהיה השטח של כל הריבוע, ובאן השטח הוא רק 0.8 כולם. השטח תחת עוקמת ה-ROC מוגדר **AUC** (area under curve): הוא מגד מקובל ומוציא, כי הוא לא תלוי ב-*cut-off* ספציפי כי הוא מתחשב בכלם, מען ממוצע או אינטגרציה של המודל על פני כלם. מודל אקראי לחוטין, שככל שאנו מועלם את הסף כך יורד ה-TPR ועליה ה-*FPR* באופן שווה, העוקמה תהיה בעצם קו אלכסוני בתוך הריבוע, והשטח מתחתיה יהיה 0.5. מודל מושלם יגיע לערך 1.

Very nice interpretation of AUC: Assume the test set has m_1 ones ($y = 1$) and m_0 zeros, then AUC is the % of correctly ranked pairs with different response:

$$AUC = \frac{\#\{(i, j) : y_i = 0, y_j = 1 \text{ and } \hat{p}_i < \hat{p}_j\}}{m_1 \times m_0}$$

שמדוברים כמובן, במודל אקראי היה 50% ! אחד הזוגות שמדורגים נכון במודל מושלם, היה 100% ! בcontra פורמלית, אפשר לסמן זאת כי. המגד לא "סובל" אם יש הרבה יותר תצפיות חיוביות מאשר משליליות במודם (כמו מגד ה-*accuracy*), זו פרופורציה שבכל מקרה נרצה שתהיה כמו שיטור קרובה ל-1.

הערה: לא הזכרנו כלל ברגסיה את דוגמת הצירום שלנו, לא השתמשנו ברגסיה לוגיסטי כדי לבנות מודל שיחזה אם ציור הוא IM או לא, אם לבנה מודל כזה – נגלה שהוא לא יהיה מאד איבוטי.

פרשנות ל-AUC: מתוך כל זוגות התצפיות האפשריות, כר שאות חיובית ואחת שלילית, כמה מדורגות נכון? אם ה-score הנמדד לתצפית החביבה בזוג, גדול מה-score לתצפית השלילית. אחוז הזוגות

נרצה שתהיה כמו שיטור קרובה ל-1.

קווי 9

שאלה 1: מהי ההצדקה התיאורטית לנוסחה לחישוב ה-MSE או RMSE?

ה-MSE מגד את מרחק ההטלה של וקטור התצפיות \hat{y} על המרחב הנפרש באמצעות עמודות המטריצה X .

הגדרכנו שאנו ממעוניינים בתוכלת ההפסד על תצפיות שלא נראה, ה-MSE על מבחן הטסט הוא אומד בלתי מוטה לתוכלת IT.

אין הצדקה תיאורטית, זה פשוט מגד "נוח" (גדי למשל)

הגדרכנו שאנו ממעוניינים בתוכלת ההפסד על תצפיות שלא נראה, ה-MSE על מבחן הטסט הוא אומד בלתי מוטה לתוכלת זו, יכולנו להשתמש גם בחzinן השגיאות הריבועיות.

שאלה 2: בלי הנחת הנורמלויות לא ניתן לבצע רגסיה ליניארית. לא נכון, בלי הנחת הנורמלויות לא ניתן לבצע הסקה סטטיסטייה כפי שהוזגה בשיעור, אבל בהחלט ניתן לבצע רגסיה ליניארית שהיא פעולה אלגברית. במו-בן יש שיטות רגסיה גם עבור הנחות התפלגיות אחרות של הטעות.

שאלה 3: נידה מנסה לשפר מודל שולץ בחישוב את כל הנתונים ששמורים בקובת החולים על يولדי ומונבא בחודש השנהví להריון האם התינוק שייפול והיה בתת-משקל (כן או לא, על-פי הגדירות מקובלות). היא מתרגשת כי הצלחה למצוא מודל שמקtin את FPR על מבחן הטסט, בהשוואה למודל הקיטם. פירוז, המנהלת שלה, ממהרת לצפן את ההתלהות: מודל שמקtin את FPR על להיות מודל גרווע מאד. הטענה של פירוז נכוןה? נכון, האם FPR נמוך מבטיח TPR גבוה? נכון, בכל עוקמת ROC סבירה.

אך מושג שלמדוּן	מתוך המציגים אוחז אלה שהמודל ניבא שלא יציטין, ועוד מתוך הלא-מציגים אוחז אלה שהמודל ניבא שכן יציטין.
Recall	מתוך הסטודנטים המציגים, אוחז אלה שהמודל אכן ניבא יציטין
FPR	מתוך הסטודנטים הלא-מציגים אוחז אלה שהמודל ניבא שיציטין
Precision	מתוך הסטודנטים שהמודל ניבא שלא יהו מציגים, אוחז אלה שכן לא יציטין

שאלה 4: אছיה פיתח מודל שמנבא ערבע התואר הראשון האם סטודנט יס"ם את התואר בהציניות (positive) או לא (negative). התאיםו בין המושג לתיאור הדבר אותו הוא מודד.

תרגול 6 – רגרסיה לינארית

רגרסיה לינארית בסיסית:

נסתכל על דאטה של pokemons, כאשר יש כל מיני שות ומאפיינים לכל פוקימון. האם סביר להთאים פה מודל לינאריו? נבצע תרשימים פיזור בין שני משתנים בלחם, נניח Attack vs HP, נחפש קשר לינאריו בלחמה.

נדיר $x = \text{Attack}$; $y = \text{HP}$. ראיינו את הנוסחאות לחישוב $\hat{\beta}_1, \hat{\beta}_0$, נוכל לחשב בפייטון באופן ישיר. קיבלנו את המשוואה הבאה:

$$\text{HP} = 40.29 + 0.33 \cdot \text{Attack}$$

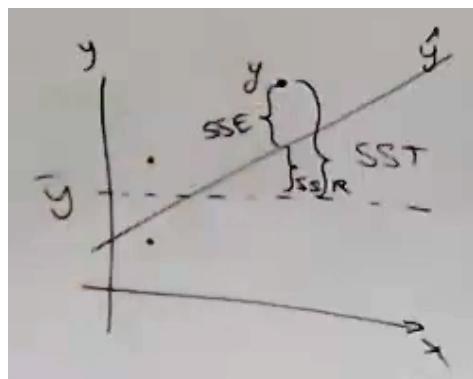
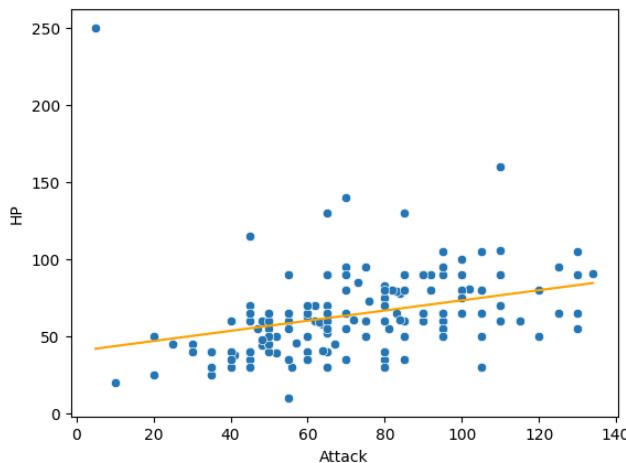
```
X = df['Attack']
y = df['HP']
b1 = np.sum((X - X.mean()) * (y - y.mean())) / np.sum((X - X.mean())**2)
b0 = y.mean() - b1 * X.mean()
print(f'HP = {b0:.02f} + {b1:.02f}*Attack')
```

$\text{HP} = 40.29 + 0.33 \cdot \text{Attack}$

```
# new import from sklearn!
from sklearn.linear_model import LinearRegression
# need X of shape (n, 1) not (n,),0
# can also use df[['Attack']].values[:, np.newaxis]
X = df[['Attack']]
y = df['HP']
lm_1 = LinearRegression()
lm_1.fit(X,y)
b0 = lm_1.intercept_
b1 = lm_1.coef_[0]
print(f'HP = {b0:.02f} + {b1:.02f}*Attack')
```

$\text{HP} = 40.29 + 0.33 \cdot \text{Attack}$

```
pred = lm_1.predict(X)
df['pred_1'] = pred
sns.scatterplot(x='Attack', y='HP', data=df)
sns.lineplot(x='Attack', y='pred_1', data=df, color='orange')
plt.show()
```



חישוב R^2 :

- SST: סכום המרחקים של y מה ממוצע (פרופורציונלי לשונות של הדאטה)
- SSR: המרחק של הקוו שנתחאים מהממוצע.
- SSE: סכום השאריות בריבוע, מהקוו לתצפית האמיתית.
- אפשר להוכיח שמתיקים: $SST = SSR + SSE$. מודד טוב הוא כמה ה-SST מצליך לפחות מתוך ה-SST, וכך מוגדר:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$



קצת חישובים:

```
# In general:
np.corrcoef(y_hat, y) # y_hat is a linear transformation of X!
array([[1.          , 0.50542014],
       [0.50542014, 1.          ]])

from sklearn.metrics import r2_score
r2 = r2_score(y, y_hat)
print(f'R^2: {r2:.03f}')

R^2: 0.255
```

```
X2 = sm.add_constant(X)
lm_4 = sm.OLS(y, X2)
lm_4 = lm_4.fit()
r2 = lm_4.rsquared
print(f'R^2: {r2:.03f}')
```

R^2: 0.254

גרסיה לינארית מרובה:

```
X = df_mod[['Attack', 'Defense']]
y = df_mod['HP']
lm_6 = LinearRegression()
_ = lm_6.fit(X,y)
b = np.concatenate(([lm_6.intercept_], lm_6.coef_])
print(f'HP = {b[0]:.02f} + {b[1]:.02f}*Attack + {b[2]:.02f}*Defense')

HP = 27.53 + 0.45*Attack + 0.04*Defense

# Could come in handy, if you have sklearn > 1.0
lm_6.feature_names_in_
array(['Attack', 'Defense'], dtype=object)
```

כאן יש לנו וקטור של $\beta \in \mathbb{R}^{p+1}$. אחרי אלגברה,
מגיעים לנCONDת המינימום $\hat{\beta} = (X^T X)^{-1} X^T Y$.
نبצע את זה עם sklearn

```
X = df_mod[['Attack', 'Defense', 'Sp_Atk', 'Sp_Def', 'Speed']]
y = df_mod['HP']
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=42)

lm_9 = LinearRegression()
lm_9.fit(X_train, y_train)
print(f'intercept: {lm_9.intercept_: .02f}')
print(f'b: {[np.round(c, 2) for c in lm_9.coef_]}')

intercept: 21.90
b: [0.39, -0.05, 0.21, 0.34, -0.3]
```

```
from sklearn.metrics import mean_squared_error
pred_train = lm_9.predict(X_train)
train_rmse = mean_squared_error(y_train, pred_train, squared=False)
pred_test = lm_9.predict(X_test)
test_rmse = mean_squared_error(y_test, pred_test, squared=False)
print(f'train rmse: {train_rmse: .02f}')
print(f'test rmse: {test_rmse: .02f}')

train rmse: 17.41
test rmse: 20.02
```

אם נוריד את Defence נראה ש-RMSE טיפונה עליה. קשה לדעת אם
זה מובהק. ואז יש מתודה של cross-validation: שומרים
בצד, ואת-h-test מחלקים ל-k folds, כל פעם מאמנים על $1 - k$
באלה ובזדים אותו על זה שלא אימנו עליו. כבה מחליפים כל פעם.

:train-test

למה R^2 זה לא המודד הכי טוב כמדד להערכתה להכניס עוד משתנים? R^2 יכול רק לגודל, ויגרום לו-overfitting (הטעות על ה-test הולכת וקטנה, ואין הכללה להתחממות עם ה-test). לכן נשתמש במתודולוגיות של **train-test**. הרבה יותר נכון לשפטו האם צריך להוסיף משתנים או לא, על פni מבחן ה-test.

**שאלות מבחנים:**

שאלה 1: קארדי מכינה סטודנטים ל מבחון, לאחרונה הינו תהיות לגבי האפקט של המבחן שלו (אל מול הציון בתיכון ובפסיכומטר). היא דגמה 32 סטודנטים: SAT, GPA, PSI. חצי מהם בקורס של הקורס (1 = PSI), וחצי מקרים אחר למגרי (0 = PSI). היא מרצה גרסיה ליניארית של GRADE (הציון בקורס) אל מול המשתנים האחרים, כדי לבדוק האם השתתפות בקורס שלו משפיעה על הציון הסופי (מעבר לאפקט של SAT ו-GPA). היא מגיעה למסקנה שכן – קורס שלו יש השפעה חיובית מעבר ל-SAT ו-GPA.

#	coef	std err	t	P> t	[0.025	0.975]
# -----						
# const	-1.4980	0.524	-2.859	0.008	-2.571	-0.425
# GPA	0.4639	0.162	2.864	0.008	0.132	0.796
# PSI	0.3786	0.139	2.720	0.011	0.093	0.664
# SAT	0.0105	0.019	0.539	0.594	-0.029	0.050
# -----						

1. Yes
2. Not necessarily, we would know if Cardi entered PSI last in the regression
3. Not necessarily, because the size of the β coefficient on its own does not mean anything
4. Not necessarily, because Cardi should have performed hypothesis testing, not linear regression

התשובה המהירה – 1. כי אם ל-PSI יש אפקט? כן, כי יצא value-k קטן מ-5%! עוד נימוק, רוח הסמן לא כולל את אפס (ולכן נדחה את השערת האפס).

תשובה 2: אפשר לשים אותו איפה שרצוים ברגסיה.
תשובה 3: זה נכון, צריך לנקח את ה-beta ולעשות אליו מבחן סטטיסטי, לחלק אותו בטעות התקן, לקבל סטטיסטי, לראות שה-k העומד מובהק, אבל עשו את זה!

תשובה 4: יש כאן למשתנה את המבחן שלו, וזה אולי כל אחד בפני עצמו. פלט סטנדרטי של רגסיה סטנדרטית זה בדיק מה שהוא אומר, האם למשתנה יש השפעה מעבר למשתנים האחרים שנכנסו.

הערות:

- ה- H_1 של קארדי היה שהקורס שלו עוזר. המקדם הוא חיובי בתוכאות כי הקורס עוזר. אם המקדם היה שלילי זה לא היה נכון – זהה לא מראה מה שהוא התבוננה.
- היא עשו השערת חד-צדדית, אבל יש כאן פלט של השערה דו-צדדית! אבל.. אם ה-value-k עובר בדו-צדדי, הוא יותר מחייב. אז בודאי נוכל להסיק מזה שהיינו טוערים גם מבחן חד-צדדי.

שאלה 2: לדוגמה יש>Data של 32 סטודנטים: GRADE, GPA, SAT, PSI. היא מרצה גרסיה ליניארית עם המודל הבא:

$$y_{GRADE} = \beta_0 + \beta_{GPA}GPA + \beta_{SAT}SAT + \beta_{PSI}PSI + \varepsilon$$

היא מקבלת את אותו הפלט מקודם. נסתכל על ערך ה-value-k בעמודה $|t| > P$ בשורה השלישי (PSI). איזו היפותזה זה בוחן?

זו בדיקת השערות דו-צדדיות, ולכן $H_1: \beta_{PSI} \neq 0$ (אין דרישת לגודל או קטן), כאשר $H_0: \beta_{PSI} = 0$. זה נכון גם לכל שורה אחרת, אותה בדיקת השערות דו-צדדיות.

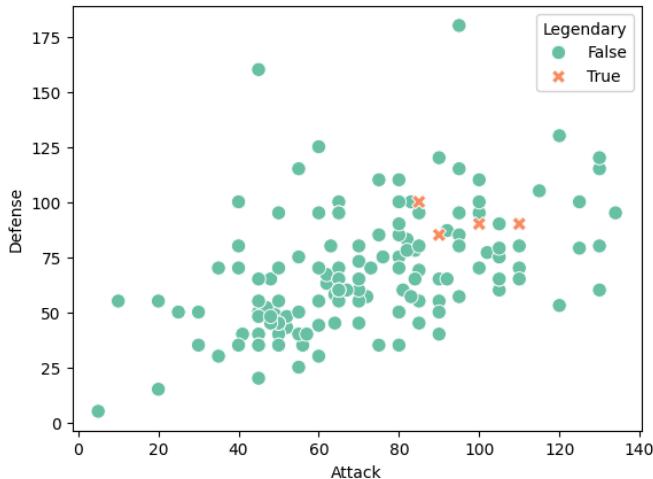
תרגול 7 – רgression לוגיסטיבית

```

fig, ax = plt.subplots(ncols=3, figsize=(20,7))
sns.boxplot(x='Legendary',y='Attack', data=df, ax = ax[0])
sns.boxplot(x='Legendary',y='Speed', data=df, ax = ax[1])
sns.boxplot(x='Legendary',y='Defense', data=df, ax = ax[2])
plt.show()

sns.scatterplot(x='Attack', y='Defense', hue='Legendary',
                 style='Legendary', s=80, palette="Set2", data=df)
plt.show()

```



זיכורות: נזכיר שוב לאטא של הפוקימונים. יש לנו את המשתנה Legendary שמקבל True/False עבשו ה-у בינהari. קודם שרטט ונראה אם יש מה להזות, בצע boxplots מול Attack/Speed/Defense ונראה שיש הבדל ביןיהם מביניהם. לעומת זאת, אם נשרטט scatterplot של Legendary Defence וככבר את ערך ה-True זה נראה שהולך להיות קשה, הדאטה הוא מאד לאamazon, יש 4 פוקימונים מטעם 150 שהםLegendary. הסיכוי שמודול יצילח לא מדהים. איזה סיטואציה מהחיים זה מזכיר? מחלות נדירות באוכלוסייה.

צריך להזכיר היבר את האזעקה classification matrix בטעושים סיווג .Accuracy, Precision, Recall בינהרי: ערכיהם של

זיכורת לרgression לוגיסטיבית:

Remember Y is binary, therefore we model it with Bernoulli distribution:

$$\begin{aligned} Y_i|X_i \sim \text{Bernoulli}(p_i) &\rightarrow P(Y_i = y_i|X_i) = p_i^{y_i}(1 - p_i)^{1-y_i} \\ &\rightarrow E(Y_i|X_i) = P(Y_i = 1|X_i) = p_i \end{aligned}$$

It is **this $E(Y|X)$** that we model, through some link function g , in our case the logit function, so:

$$g(E(Y_i|X_i)) = g(p_i) = \text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X_{1i} + \dots + \beta_p X_{pi}$$

Once we get our estimate $\hat{\beta}$:

- We could "explain" \hat{Y}_i , the size and direction of each component of $\hat{\beta}$ indicating the contribution of that predictor to the *log-odds* of \hat{Y}_i being 1
- We could estimate the probability a of new observation X_i to have $\hat{Y}_i = 1$ by fitting a probability $\hat{p}_i = \frac{1}{1+e^{-X_i\hat{\beta}}}$, where typically if $\hat{p}_i > 0.5$, or $X_i\hat{\beta} > 0$, we predict $\hat{Y}_i = 1$
- Then calculate accuracy, precision, recall, AUC, to evaluate the model

$$\text{נשים לב כי אם } 0 = x \text{ זה יגרור לפי החישוב כי } p_1 = \frac{1}{2}.$$

чисובים: ניעזר ב-sklearn, יש לנו שני משתנים ב- x , וה- y הוא המשתנה של intercept. נקבל שדה של intercept ושל coef ברגיל, וקיבלנו כי $\text{logit}(P(\text{Legendary})) = -11.32 + 0.04 \cdot \text{Attack} + 0.05 \cdot \text{Speed}$. מה זה -11.32 ? אם יש 0 בשני הפרמטרים אחרים, אז ה-log odds הם כמו באלה.

```

from sklearn.linear_model import LogisticRegression
X = df[['Attack', 'Speed']]
y = df['Legendary']
# amazingly the default penalty is 'l2', for older sklearn use 'none'
lr = LogisticRegression(penalty=None)
_ = lr.fit(X, y)
b = np.concatenate([lr.intercept_, lr.coef_[0]])
print(f'logit(P(Legendary)) = {b[0]:.02f} + {b[1]:.02f}*Attack + {b[2]:.02f}*Speed')

logit(P(Legendary)) = -11.32 + 0.04*Attack + 0.05*Speed

```



נקרא `-predict_proba()`, מה הסיכוי להיות מקהלאס הראשון ומה הסיכוי להיות מקהלאס השני. כלם מקבלים הסתברות. נקרא גבוזות לא להיות Legendary, וזה חומר מה הבעיה לחזות קלאס סופי עם סף של 0.5. אם נקרא `-predict()` כלום False.

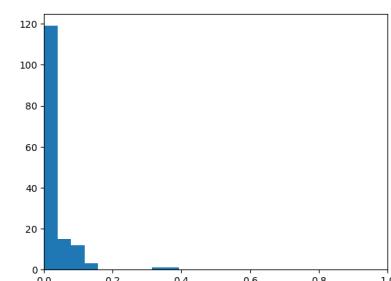
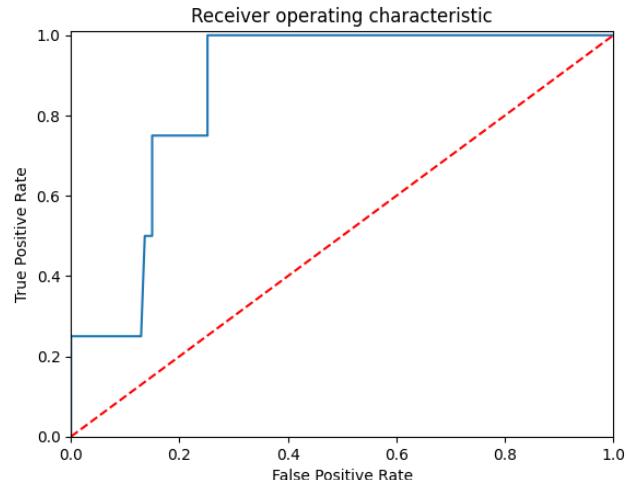
נסתכל על ה-`confusion_matrix`. נחשב `accuracy` ונראה שהוא $\frac{147}{151}$ שזה נשמע הרבה מצד אחד, אבל זה לא מודל מדהים כי הוא לא חוצה אף פעם True! זה לא חוכמה להגיע לדיקן 97% בש-97% מהדאטא אכן לא Legendary ותמיד אומרם False.

		Pred_not_legend	Pred_legend
Not_legend	147	0	
legend	4	0	

	precision	recall	f1-score	support
False_Legend	0.97	1.00	0.99	147
True_Legend	0.00	0.00	0.00	4
accuracy			0.97	151
macro avg	0.49	0.50	0.49	151
weighted avg	0.95	0.97	0.96	151

נشرط את עקומת ROC: עבור כל סף שאפשר לחשב עליו מ-0 עד 1 נشرط את הנקודה שהגענו אליה מבחןת TPR ו-FPR. אינטואטיבית נרצה FPR נמוך ו-TPR גבוהה. מודל מושלם עבור סף מסוים מגע ל-100%-recall ו-0% FPR. העקומה נראית לא רע בסה"כ.

```
from sklearn.metrics import roc_curve, roc_auc_score
y_pred_prob = lr.predict_proba(X)[:, 1]
auc = roc_auc_score(y, y_pred_prob)
fpr, tpr, thresholds = roc_curve(y, y_pred_prob)
plt.figure()
plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1], 'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.01])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.show()
print(f'AUC: {auc:.02f}')
```



לפי ההיסטוגרמה, אם נשים סף של 0.5 זה אומר אף פעם לא לחזות Legendary. בבעיות שבנה הדאטא לא מאד מואזן, עדיף ללבת על השכיחות המקורית באוכלוסייה שהיא $\frac{4}{151}$. בעצם נראה מודל קצת יותר בריא, שכן ראה שייך לנקודת השניה שראינו בעקבות ROC.

		Pred_not_legend	Pred_legend
Not_legend	111	36	
legend	1	3	

	precision	recall	f1-score	support
False_Legend	0.99	0.76	0.86	147
True_Legend	0.08	0.75	0.14	4
accuracy			0.75	151
macro avg	0.53	0.75	0.50	151
weighted avg	0.97	0.75	0.84	151

שאלה 1:

In a logistic regression model predicting the probability of a customer making a purchase based on their age, income, and gender, the coefficient for the variable age is estimated to be -0.05. Which of the following interpretations is correct?

1. The coefficient for age cannot be interpreted without knowledge of the coefficients for income and gender
2. For a one-unit increase in age, the predicted probability of making a purchase decreases by 0.05
3. Age has no significant effect on the probability of making a purchase
4. For a one-unit increase in age, the predicted log odds of making a purchase decreases by 0.05

התשובה היא 4. כולם נקבע $\beta = 0.05$ עבור age. זה בדוק כמו רגסיה לינארית, בשיש מקדם מסוים זה אומר שם כל המשתנים הם אותו דבר, השתנות פי beta היא עליה ביחס ב- α . כאן זה משפיע על log odds. תשובה 2 זה על log odds. אבל אנחנו בלוגיסטיות והשינוי משפיע ישירות על odds. כדי להגיע להסתברות צריך לקחת את המקדם ולהציב בחישוב של e .

 שאלה 2:

A news website is trying to target cat-lovers among its readers, to present them with cat food banners inside articles.

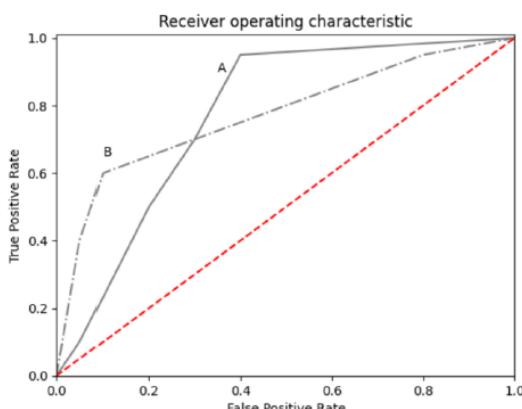
The cat food company pays per exposure, and the price is high, so the CEO says she cares about precision, not recall. Or in her words: "I want to maximize the rate of clicks on banners, i.e. if you're targeting a reader as a cat-lover I want to be sure about it so it's worth my money. I care less about losing some true cat-lovers along the way and not showing them the banners." A data scientist from the news company gathers data on cat-lovers and non-cat-lovers readers.

She splits it to train and test.

She compares two classification algorithms for classifying a reader as cat-lover (positive) or noncat-lover (negative), by looking at the ROC curves on her test set:

Assuming the data is balanced, i.e. there is roughly the same number of cat-lovers and non-catlovers in the data and in production, which of the algorithms will adhere to the cat food company CEO's wishes?

1. Whichever model has the higher AUC
2. B (dot-dashed line)
3. There is no way of knowing from looking at these ROC curves which model is more likely to adhere to the cats food company CEO's wishes
4. A (solid line)



התשובה היא 2.

נזכיר כי ציר ה- y באן, ה-TPR הוא בעצם Recall. בברור נראה שמודל A שמציל יתגעה ל-recall גבוהה לא מעניין אותה. בהינתן שהדעתה מדוונת, וגדיל הקבוצות יחסית דומות, יש קובלץיה מאוד גבוהה בין FPR לבין precision��. ולכן מודל A מוביל לrecall גבוהה, בעוד Moudel B מוביל לprecision גבוהה.

graf B הוא הנכון. FPR נמוך בא עם precision גבוהה.

שכנים ועצי החלטה

KNN

בקע: ראיינו עד בה **מודלי גרגסיה**, שהם מודלים גלובליים – יש לשניהם סט של פרמטרים β , תוצר הרגרסיה הוא הסט הנameda $\hat{\beta}$, ועל נקודה במרחב אונחנו מחלים את נסחתת הרגרסיה, שהיא משווה אחת גלובלית. דוגמה, בשמידלנו את הסיכוי לפצייניטים להחלות במחלת לב, לא התאמנו נסחה שונה לפצייניטים עם היסטוריה משפחתיות וחולים בלי היסטוריה משפחתיות של המחלת. שם נוסף למודלים באלה, הוא **מודלים פרמטריים** – והם באים עם לא מעט הנחות, כמו קשר לנארו, ואם רוצים לבעה הסקה גם הנחות סטטיסטיות. **צורת מחשבה** שונה **לחלווטין** היא **מידול לוקאל**. פציינט שאין לו היסטוריה משפחתיות של מחלות לב, שהוא בן 50 ומעשן, יחפש במדגם הלמידה פצייניטים אחרים שדומים לו – ונראה אם להם יש מחלת לב או לא (או אחד מהם בעלי מחלת לב).

מודל NN: הדוגמה הפשוטה ביותר למודל nearest neighbor הוא NN. הוא מורכב מהשלבים הבאים:

1. מטריקת מרחק: הבחירה הפשוטה ביותר היא מרחק אוקלידי, עברו $\mathbb{R}^d \in x$:

$$d(x, u) = \|x - u\|^2$$

2. בשמגיעה תצפית חדשה $T \in \mathbb{R}^d$, אין "מודל". אנחנו מחשבים מי השכן הקרוב ביותר שלא במדגם הלמידה Z . חשוב לשים לב שאין כאן שום התחשבות ב- y , המטריקה מחושבת רק על וקטורי ה- x :

$$i_0 = \arg \min_i d(x_0, x_i)$$

3. נזהה את התצפית \hat{y}_0 של השכן הכי קרוב i_0 .

ההכללה המתבקשת היא NN. למה להסתבל רק אצל השכן הכי קרוב לתצפית? אולי נסתבל אצל 10 השכנים וنمצע אותם? ברגסיה, החיזוי של תצפית יהיה ממוצע ה- y של שני השכנים: $\hat{y}_0 = \frac{1}{k} \sum_{j=1}^k y_{i_0,j}$.

בסיוג לשתי מחלוקת החיזוי של תצפית יהיה אם הרוב שייר למחלקה אחת: $\begin{cases} 1 & \text{if } \frac{1}{k} \sum_{j=1}^k y_{i_0,j} > \frac{1}{2} \\ 0 & \text{otherwise} \end{cases} = \hat{y}_0$. כמובן, נזהה 1 אם למעלה מ-5 שכנים מתוך 10 הם גם 1. אפשר גם לחזות את ההסתברות להיות 1 בקרוב השכנים, לצורך חישוב עקומת ROC וחישוב AUC. באיזה K משתמש?

```
from sklearn.neighbors import KNeighborsClassifier

ntr = SA_Xtr.shape[0]
nte = SA_Xte.shape[0]
tr_err = []
te_err = []
kvals = [1, 3, 5, 10, 50, 100, 200]

for k in kvals:
    knn = KNeighborsClassifier(n_neighbors = k)
    knn.fit(SA_Xtr, SA_Ytr)
    yhat_tr = knn.predict(SA_Xtr) > 0.5
    yhat = knn.predict(SA_Xte) > 0.5
    tr_err.append(np.sum(yhat_tr != SA_Ytr) / ntr)
    te_err.append(np.sum(yhat != SA_Yte) / nte)
```

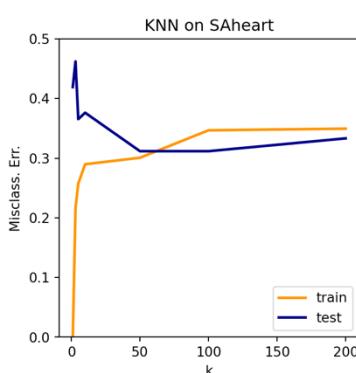
סיווג דאטא SAHeart: אנחנו מעוניינים לחזות האם פצייניטים מודром אפריקה יחלו במחלת לב – כן או לא. נעשו סימולציה של KNN עם K שונים. נאתחל וקוטרו טעות ל-train/test, ולכל K שבבדוק נאתחל את המחלוקת ונתאים על מדגם הלמידה. אם לא ציינו אחרת, המטריקה בשימוש היא מרחק אוקלידי.

עיקר העבודה ב-KNN תהיה בשביל החיזוי: בשמגיעה תצפית חדשה, אנחנו צריכים לחשב את המרחק בין כל התציפות כדי לקבל את K השכנים הקרובים ביותר.

החיזוי הסופי הוא האם ממוצע התציפות גדול מחצי. נחשב טעות misclassification וכןיף לרשיונות שלנו.

האם מרחק אוקלידי ראוי לעיה שלנו? לא בטוח, מרחק אוקלידי כמו שנות מושפע מאוד מערכיים קיצוניים. במקרה שלנו, למשתנים שונים יש סקלות שונות. המרחק עצמו עלול להיות מושפע יותר ממשתנים מסוימים ולא אחרים. לכן, מומלץ לעשות סטנדרטיזציה לפני הצורך, לפני שמරיצים KNN.

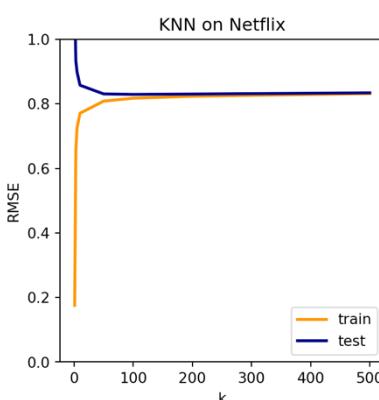
אנחנו מושרטים את טעות החיזוי עבור train/test בפונקציה של K . דבר ראשון ראוי לשים לב אליו, היא טעות החיזוי עבור $train$ במקורה של $1 = K$. במקרה זה נזהה עבור התצפית את הערך שלה (מקרה טריוויאלי). הדרך היחידה שיכולה להיות תחרות היא אם יש תיקו, תצפית אחרת (לא המקורית) שזהה מבחינת המרחק, וזה ערך ה- y לא יהיה של התצפית המקורית. באופן כללי, הקו הכתום הוא פחוסות מעניין. הקו הבהיר הוא המעניין, כי הוא מיציג את השגיאה עבור נתונים שהמודול לא ראה. עבור K קטן שגיאת test גדולה, מגיעים למינימום טעות באמצעות (סביב 50 שכנים), ואז הטעות מתחילה לעלות.





```
from sklearn.neighbors import KNeighborsRegressor
ntr = NE_Xtr.shape[0]
nte = NE_Xte.shape[0]
tr_err = []
te_err = []
kvals = [1, 3, 5, 10, 50, 100, 200, 500]

for k in kvals:
    knn = KNeighborsRegressor(n_neighbors = k)
    knn.fit(NE_Xtr, NE_Ytr)
    yhat_tr = knn.predict(NE_Xtr)
    yhat = knn.predict(NE_Xte)
    tr_err.append(np.sqrt(np.sum((yhat_tr - NE_Ytr)**2) / ntr))
    te_err.append(np.sqrt(np.sum((yhat - NE_Yte)**2) / nte))
```



אינטואטיבית, להתחשב רק במקרים יקרים או החלטה מאוד חושת – סביר להניח שמעט מאוד שכנים יכולים לטענות מאוד בחיזוי. להתחשב בהרבה שכנים זה גם לא טוב – ככל שנגדיל את הרדיוס שאומר מה זו "שכונה", נקבל שכנים רחוקים מהם בבר לא רלוונטיים ולא צריך להתחשב בהם. על כן, ה-K האופטימלי נמצא במקומם בൾשוו אבעמצע (sweet spot).

גראסיה על הדאטא של נטפליקס: במדגם הלמידה יש 8000 צופים שדרוגו מ-1 עד 5 בקטגוריות של 14 סרטים. במדגם הטסט יש 2000 צופים. הפעם ה-ע הוא כמותי ולא קטגוריאלי, אנחנו

ברוגסיה. כאן אפשר לשער שמרקח אוקלידי יהיה מותאים יותר, כי כל הסרטים מדורגים באותה סקללה. התהילה' במעט זהה, רק שימושה ב-KNeighborsRegressor. נקבל במעט ממוצעת על פני השכנים, וטעות החיזוי שלהם היא ה-RMSE.

גם כאן קיבלנו דפוס דומה – שגיאת החיזוי (RMSE) מתקרבת ל-0 אבל לא זהותית 0 במדגם ה-`train` כי יש ברירה ties (משתמשים שדרוגו אותו דבר על כל 14 סרטים). היקו הבחן (test) מראה שבסבירות 100 שכנים מוגעים למיניהם שגיאה ומעבר להabar לא משנה.

היתרון ה-ע גדול של המודול – אין הנחות. אילו בעיות בכל זאת יש עם המודול הזה, שבפועל לא משיג חיזויים טובים על נתונים מסוימים כמו מודלים אחרים שנלמד?

1. **בחירה המרחק** – כאן דזוקא בן מסתורנות הנחות. איך נראים נתונים שמתאימים למרחק אוקלידי או אחר?

2. **דאטה דיליל (sparse) באוצרים מסוימים** – אם נקודה רוחקה מאוד מהשכןibi קרוב שלא, NKN לא מביא את זה לידי ביטוי למרות שהשכן בבר לא רלוונטי. כשהמייד של הבעה (מספר המשתנים) אנחנו יכולים להיות בטוחים שתופעה זו תקרה, ונקרא לה curse of dimensionality.

מודל אדפטיבי: פתרון אחר הוא להגדיר מה "שכונה" של תצפית, לא באמצעות מטריקת מרחק, אלא באמצעות הסתכלות על הנתונים עצם בצורה אדפטיבית. חלקים שבתוכם עמשתנה כמה שפות, ההתפלגות של כמה שיוטר אחדיה. בעת שתגיע תצפית חדשה, נסוג אותה לשכונה שמתאימה לה (שאנו יודעים מה ה-ע שלה פחות או יותר).

עצי החלטה

עץ החלטה: עץ החלטה הם דוגמה מצוינת **לביצירת מודול בצורה אדפטיבית**, על ידי הצעה אל עץ תוך כדי בניית המודול. נרצה לחלק את מרחב הנתונים (X) לשכונות, בצורה רקורסיבית/אדפטיבית. כל פעם נגיע לאזור נתון ונחלק אותו לשניים, בצורה שתחלק את עלי אוזרים שונים זה מזה כמה שיוטר. אם מחלקים לשני אוזרים – מדובר בעץ ביןארי (יש גרסאות עם חלוקה ליותר ענפים).

לדוגמה על הדאטא של טיטאניק: השאלה הרואה ששאלתו שאלת ה-`is sex male?`. עבור נשים – 36% מהמדגם, הן שרכזו בסיכוי 0.73. עבור גברים – יש לשאל עוד שאלות. השאלה הבאה היא "האם הגיל מעל 9.5?". אם כן – 61% מהמדגם שייכים לשכונה זו והם שרכזו בסיכוי 0.17. אחרת – "מספר אחים?", ונגיע לסוף העץ.

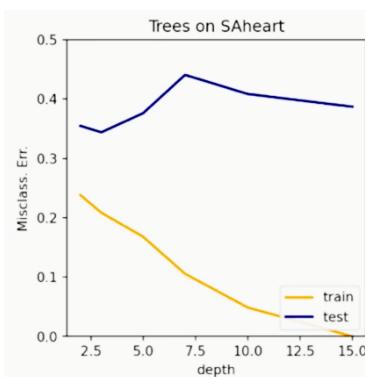
נזהה את מה שראינו בעלים הסופיים של העץ (כמו ב-KNN). אם נרצה הסתברויות חיזיות, נזהה את ההסתברות לכל עלה, שהוא אחוז השורדים בשכונה הזאת. אם נרצה מחילה סופית, נבדוק האם הסיכוי לשרוד בעל גודל מחייב או לא. למשל עבור נשים נזהה שהן ישרדו (כי קיבלו 0.73).

עומק העץ המקסימלי כאן הוא 3. כל תצפית תנגע לסוף העץ ומקבל חיזוי עם מקסימום 3 שאלות.

על SAHeart: נגידר לכל היוטר 2 שאלות בשבייל להגעה לחיזוי. העץ מחלק את המדגם לפצינטיטים בני למעלה מ-50 ו מתחת ל-50. שאלת נספת היא מידת השימוש בטבוק. עבור פצינטיטים צעירים יותר, שאלת נספת היא עבור אישיות מס' A.

```
from sklearn.tree import DecisionTreeClassifier, plot_tree

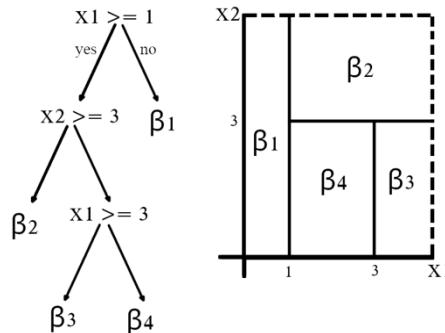
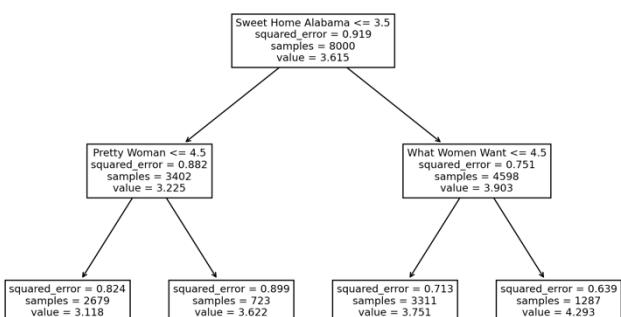
tree = DecisionTreeClassifier(max_depth = 2)
tree.fit(SA_Xtr, SA_Ytr)
plot_tree(tree, feature_names=SA_Xtr.columns)
plt.show()
```



יתרונות: העץ קל לפירוש, לא צריך להיות מודע נתונים כדי להבין את הלוגיקה שבאלגוריתם. מוחבר באלגוריתם שמאוד קל למשח.

אי בוחרם את depth_max: נעשה תרגיל דומה עבור ערכים שונים של A , ומתקיים דפוס זהה – עבור מדגם הלמידה השגיאה קטנה, בסוף נהיה מאוד מאוד ספציפיים. עבור מדגם הטסט – בעומק 3 שאלות מתתקבלת התוצאה הטובה ביותר.

גرسיה על הדאטה של נטפליקס: שאלת ריאונה שנשאל כדי לחזות את הציון של "איזו מין שטרת" הוא על הסרט sweet home alabama. נעשה את אותה בדיקה על train/test וקיים RMSE הבי נמוך של 0.8.



באמת נראה שהמודל חילק את המרחב 4-שכונות די דורות, והזה ציון שונה לחולוטן לכל אחת. **החיזוי לכל קבוצה אחד ייחיד – לכל מי שהגיע – לעלה השמאלי נחזה –** **לכן מודל של עץ הוא אבל זאת לא מאד גמיש.** אפשר לראות את הנוקשות הזאת בחולקה של המודל לשכונות בצורת "מלבנים" בפועל בשמשרטנים לאורך הציר. המודל יתקשה למצאו שכונה מצורת "יעגול".

בנייה עץ לרגרסיה:

יש כמה אספקטים מרכדיים בעיצוב של אלגוריתם עץ החלטה לרגרסיה או סיוג:

1. אין תבצע חלוקה ל-2 בכל צומת?
2. מתי מפסיקים את החלוקה? (הגדemo עומק מקסימלי, אבל יש גם דרכי אחרות)
3. מה יהיה הערך החיזוי עבור התצפית שmagua לעלה? (יכולות להיות תשובות חממות יותר מהתשובה לעלה)

בעיצוע החלוקה: בעץ לרגרסיה, **CART**, **הקריטריון לחולקה יהיה RSS**. הגענו לצומת מסוים בעץ ויש לנו קבוצה מסוימת של תציפות שאנחנו רוצים לחלק. הצעת חלוקה מסוימת אומרת להסתבל על משתנה j ועל הערך שלו s . נשים בשכונה שמאלית את כל התציפות שקטנות מהמשתנה הזה, ובשכונה ימונית את התציפות שגדלות ממנו. מציצים ב- u -ובזקיקים מה המוצע שלו בשכונה השמאלית והימנית. נרצה שכל שכונה תהיה הומוגנית במה שייותר (קרובה למוצע שלה). لكن הקרייטריון להביא למינימום הוא סכום RSS של שתי השכונות.

$$L(j, s) = \{i \leq r : x_{ij} \leq s\}, \quad R(j, s) = \{i \leq r : x_{ij} > s\}$$

$$\bar{y}_L = \frac{\sum_{i \in L(j, s)} y_i}{|L(j, s)|}, \quad \bar{y}_R = \frac{\sum_{i \in R(j, s)} y_i}{|R(j, s)|}$$

$$RSS(j, s) = \sum_{i \in L(j, s)} (y_i - \bar{y}_L)^2 + \sum_{i \in R(j, s)} (y_i - \bar{y}_R)^2$$

כדי למצוא את החלוקה הטובה ביותר ביוותר, נדרש לבצע על כל המשתנים האפשריים, ובכל משתנה על כל הערכים האפשריים. **זמן אימון האלגוריתם עוזה די הרבה,** עבור על כל הזוגות s, j . נעשה את הפיצול ונמשיך וקורסיבית באותו אופן.

אחרי החלוקה, התציפות בכל קבוצה כבר לא משפיעו יותר על החלוקה בקבוצה אחרת – לא נחזור אחריה בעץ הקלאסי לראות אולי הייתה יכולה להיות חלוקה טובה יותר. מדובר באלגוריתם greedy.

תנאי העצירה: **למה להפסיק לגדל את העץ ולא למת לא גדול ולגדול?** **לרוב העץ יבצע overfitting** למדגם הלמידה והחיזוי שלו יהיה באיכות נמוכה על נתונים שהוא לא ראה. סיבה אחרת היא חישובית – האימון והחיזוי ייקחו יותר מדי זמן ככל שנאפשר עצים עמוקים יותר. דרך אחרת להחליט היא פרמטר גלובלי של **max_depth**: נקבל עצים שהם balanced. דרך אחרת יכולה להיות **התוללת מוקומית** – מספר מינימלי של תציפות בעלה, שיפור RSS מעל סף בלחשו. ב-CART יש רעיון נוסף, לגדל את העץ כמה שייותר ובסוף לבצע לו pruning (קטימה).

הערך החיזוי: ראיינו שבעץ לרגרסיה נחזא את המוצע. אבל זו לא החלטה שרירותית, גם ברגרסיה לינארית בעצם מודלים את התוללת המותנית [א]ע[ג]. אם עמשתנה כמה שפחות כי העלה שלנו מאוד הומוגני, התוללת היא קבוצה קטנה או יותר, ואנחנו יודעים שככל עלה ועלה, האומד הבי טוב תחת הפסד ריבועי כמו שלנו, הוא ממוצע המדגם שהגיע לעלה.



יתרונות וחסכנות של עצי החלטה:

משתנים קטגוריאליים: עצים טובים מאוד עם משתנים כategorical, בוגרים לשיטות שלמדו לנו ורגסיה או **NNN** שזה פחות טבעי להם. אפשר בעצים לעשות רק פיצול ביןארי, לא לפי סף, אבל אפשר לחלק אותו לשתי קבוצות של קטגוריות β_1, β_2 .

דאטה חסר: טבעי להזין לעץ דאטא עם תצפיות חסרות. יש אסטרטגיית של עצים לטפל במקרים עם נתונים חסרים. CART בכל פיצול לא מחשב רק את הפיצול הטוב ביותר, אלא גם את הבאים אחריו (פיצולים שנקראים surrogate splits).

יתרונות מרכזים:

1. **Interpretability:** התוצר מודע אינטואיטיבי. מדובר בתרשימים זרימה – אוסף של שאלות בסופן מתקבלת תשובה. כאן יש לנו built-in feature selection.
2. **גמיש מודע (עם bias נמוך):** משתנים קטגוריאליים, נתונים חסרים, ורגסיה או קלסיפיקציה.
3. **לא לינארי:** מדובר במודל שיכל למודל יחסים לא לינאריים אם הוא עמוק מספיק. האם העץ הוא עדין "שיטה לינארית"? הסוד הוא שעדין אנחנו מסמנים β_1, β_2 כי את העץ אפשר לבנות במודל לינארי, אבל לא על הפיצרים המוקורים, אלא משתנים שהם כמו אינדיקטור.

חסכנות מרכזים:

1. **אזור החלטה מלכניים:** מכיר את classifier לעבוד בצורה של מלכניים (rectangular decision boundaries). זה תמיד מתקבל לצירם.
 2. **ኖקות על חלוקת הדאטא:** מבנה העץ אינטואיטיבי אבל גם סובל מנוקשות גבוהה. אם עושים חלוקות קצרה אחרת של הדאטא, עלולים להתקבל עצים שונים לחלווטין.
 3. **עץ הוא פשוט לא מודל חיזוי טוב בפועל!** שגיאת החיזוי על test היא חיסרון גדול שלו. הסיבה נעוצה בכך שלמרות מה שאמרנו, העץ אחד מגיע ממוקב עצום של עצים, שאנו מחשפים בו בצורה חמדנית, וכך מקבלים מודל אחד בנוואה לא אופטימלי, עם שונות גבוהה מודע: תגיע תצפית חדשה, אין לה חיזוי טוב בעץ יחיד.
- נכזה לשמר על היתרונות של העץ ולהשתמש בו ב-model-sub לצד כלים נוספים / קומבינציה של מספר עצים.

קווייז 10

שאלה 1: מודיע נכללים בקורס NNN ועצים החלטה באותו שיעור?

- אלה שיטות "לוקאליות", כלומר לא מניחות מודל בלבד בשחו על כלל הנתונים ובונות איזורי ההחלטה גמישים למורכבים מ"שכנים".
- אין סיבה מהותית, אלה שתי שיטות פשוטות שאיחדנו לשיעור אחד.
- אלה שתי שיטות שבאופן כללי הביצועים שלהם אינם גבוהים אבל הן מהוות בסיס לשיטות חזקות יותר שנלמד בהמשך.
- אלה שתי שיטות שדרושות בזוויתם של פרמטרים (K מספר השכנים בNNN, ועומק העץ למשל בעץ ההחלטה).

שאלה 2: כיצד KNN יתמודד עם משתנה קטגוריאלי? לדוגמא, KNN צריך להתמודד עם משתנה כמו מקצוע, שיש לו 1000 קטגוריות: מורה בבית הספר, מרצה באוניברסיטה, בנאי וכוכי.

- משתנה קטגוריאלי עם Q קטגוריות מומלץ להפוך לQ משתני-דמה, כל משתנה-דמה β_i יכול 1 אם הקטגוריה של התצפית היא קטgorיה β_i ו-0 אחרת (או one-hot encoding) – קשה להגיד שזה "מומלץ" תמיד, לעשות דבר כזה על משתנה עם $Q=1000$ רמות עלול לגרום ל-overfitting ולבעיות אחרות.
- משתנים קטגוריאליים אכן מהווים אתגר עבור KNN ולא דיברנו על זה בשיעור.
- משתנה קטגוריאלי עם Q קטגוריות מומלץ להפוך לQ מספרים שבמילים את הקשר של כל קטגוריה למשתנה Y הנמדד, לדוגמא ערך ה-T המתkeletal בביטוי מבחן T לדוגמאות בלתי-תלויות על Y בין תצפיות מקטגוריה זו לכל שאר התצפיות – תתפלאו, הצעה זאתinskala ברכינותו.
- KNN לא יכול לעבד על משתנים קטגוריאליים – פחות טבעי להם, אבל אפשר.

שאלה 3: להגדלת העומק של העץ ההחלטה יש השפעה דומה על המודל כמו להגדלת מספר השכנים K בNNN. לא נכון, להיפך! הגדלת עומק העץ מאפשר שכבונות יותר ויותר ספציפיות ומפורלות, איזוריים גמישים יותר וייתר אולי גמישים מדי, יש סבנה לא-overfitting. מה יקרה עם גידול את מספר השכנים K בNNN? המודל יכול יותר והסיכוי לבצע overfitting קטן.

תרגול 8 – שכנים ועצים החלטה

:KNN

רעיון: נסתכל שוב על הדאטה של פוקימון. **נרצה לסוג ליותר ממחילה אחת, אלא K מחילות.** למשל $3 = K$ עם המשתנה Stage. בולם במתאר זה יש $\{1, 2, 3\}$ -X יכול את כל שאר המשתנים.

ראשית, צריך לראות אם יש משהו בכלל – האם מודלים לינאריים יעבדופה? נشرط **boxplot** על משתנים שונים מול **Stage**. תרשימים פיזור – האם אפשר על ידי קווים לאוצר אזור החלוקת בין המחלקות? לא נראה מאד לוגاري, לבו

הדאלא עדין לא מאד מאוזן בין Stage-ים שונים.

NKN: כאשר מגיעה תכנית חדשה, נסתכל על A השכנים היכי קרובים שלו ונקבע לפי השכונה איך לסייע את הנקודה (אם זה היה גרסה היינו מבצעים מומוצע של השכנים).

בדי לקבל את ההסתברויות החזיות נבצע `predict_proba()`, בל透fect, שורה, וכל עמודה מתאימה לאחד מ-K המחלקות שאנו חווים. עבור התצפוי הראשונה למשל, 3 השכנים הכי קרובים אליו חשבו שהוא ב-1 Stage. בתצפית השנייה, שכן אחד (שליש) חשב שהוא 1 Stage ושלושה שכנים (שליש) חשבו שהוא 2 Stage לכה בחלוקת עובדה.

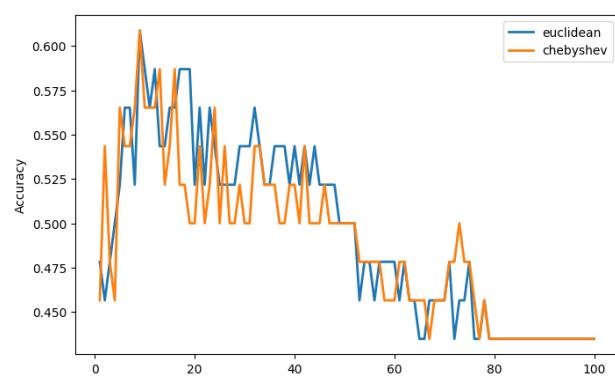
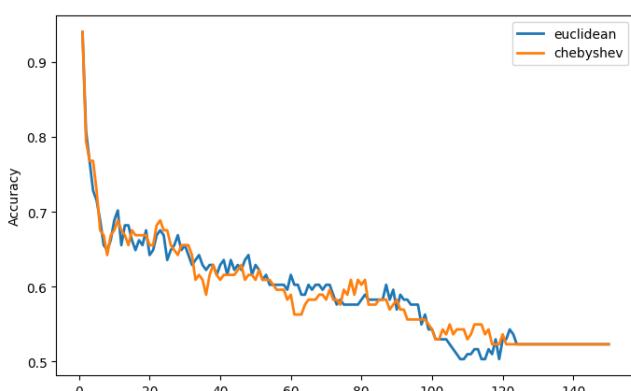
[מה שקרה עם NKN, הוא לוחק את training-data ובל תצפית משוכית כבר ל-stage מסוים, כמשמעותו תצפית חדשה אנחנו קודם מוצאים את ה-K בכדי קרובים אליה, ואך על כל אחד רואים (כ) זה נתנו לנו בחלוקת מה-training (data) לאיזה stage היא שייכת. ניקח את stage שהכי הרובה תצפויות מהשכונה שמצאנו שייכות אליו, ואך נשיר גם את החדשה אליו]

בבצע predict() על הדאטא שאימנו עליי את הנתונים (בדיקה ד' בעייתית, נרחב על זה בהמשך), וניצור confusion matrix.

בחירה K: נקרא **hyperparameter**. אידיאלית צריך לבצע cross-validation כדי לבחור אותו. נסתכל על K-ים שונים, בשוויה מರחק אוקלידי ומרחיק צ'יבש. נראה בבירוק שהypi טוב מבחינת דיוון על ה-**training data** והוא $K = 1$.

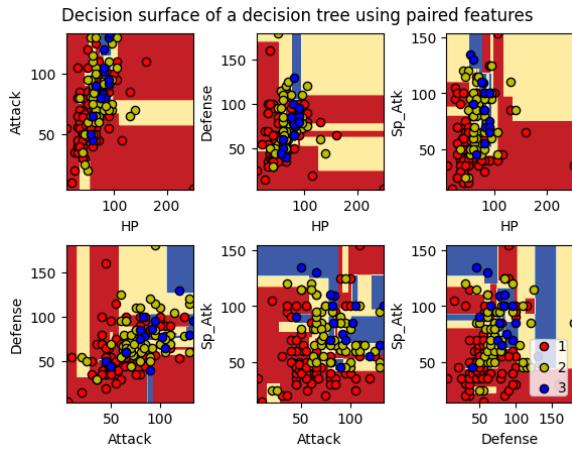
למה לא הגיעו ל-100% דיווק ורק במעט? ברגע שיש חפיפה בין תוצאות הוא לוחץ את הראשון שהוא רואה, בשים ties איז לא תלמיד נבחר את עצמינו! ואז יכול להיות שלא הגיע ל-100% על המדגם train כי ניקח ע' אחר.

אבל... מה יקרה בשמודל יתקל בדעתך אחר? Test/validation וכו'? מכך שני K גדול מידי לא גמיש מספיק, צרי spot sweet spot בפועל. בשנבען את הבדיקה של הדיק על נתונים שהמודל לא ראה (test) באמצעות מתќבל הדפוס שאנו חנו מוצפים לו. **ביצענו**





האם יש עניין של סיבוכיות/סקלאබיליות בשימוש במודל זה? בשניהם את המודל הזה ב-production, צריך לספק את כל ה-training set בivid איתו, לשיט הרבה DATA בענן. ה-inference והחיזוי הוא לא סתם מכפלה של מטריצות וקטורים. ועודין, בשאקריםים ל-fit בגודל המודל לא עושה הרבה, אין אימון מורכב.



עצי החלטה – סיווג:

הדוגמה: כל פעם נבחר שני משתנים אחרים מהDataset של פוקימון, ונשרטטו את המרחב שיצא. התוצאות צבועות לפי 3 האופציות של המשתנה Stage. אפשר להשווות זאת לרוגסיה multinomial וראות כמה היא מוגבלת (יכולת רק לעבור שני קווים).

אחד היתרונות הוא interpretability: יש לנו את המתודה plot_tree(). שמאפשרת לנו לצייר את העץ ולעקוב אחר החלטות שמתיקלות.

גם כאן הפרמטרים של max_depth,min_samples_split,Dورשים מהלך .hyperparameter tuning

עצי החלטה – רוגסיה לינארית:

נשתמש בחלוקת אחרת, DecisionTreeRegressor ומדוד את MSE. מה שנעשה בסוף, בעלה של העץ – **זה הממוצע עליה**, מספר אחד ייחיד שמתבסס על מה שראינו בתוצאות שהגיעו במדגם הלמידה עד לעלה זהה.



שיטת אנסמבל

שיטת אנסמבל עשוות קומבינציה של מודלים חלשים רבים, למודל חזק במילוי. ראיינו בעצמי החלטה, שהם מודלים מוגבלים עם שונות גבוהה, ושאלנו את עצמנו למה עז אחד? למה לא יער? נראה שתי שיטות מבוססות על עצים. הראושנה מבוססת על מיצוע של עצים, והשנייה כוללת בנייתו של עז אחר עז בצורה אדפטיבית.

rndom forest

שיטת Random Forest: במקומות להתאים עז אחד לנ נתונים, נתאים הרבה. אבל לא מתאים אותם לנ נתונים – כל פעם על דאטה קצר אחר, שעבור רנדומיזציה בשתי דרכים שונות. לבסוף – נמצאו את העצים. החיזוי לכל תצפית יהיה ממוצע של פני הרבה עצים. בר נטפל באופן ישיר בעיות של העז היחיד.

ערך המיצוע: בשידירנו על CLT ראיינו כי אם $F \sim z_i \sigma^2 + \mu$ וניקח m תציפות ב"ת באלו, אז השונות של ממוצע המודגם המקורי קטנה פי m . בלומר, **כל ש-m גדול בר הפיזור סביר הממוצע קטן והוא מתקרב לתוחלת האמיתית**:

$$Var(\bar{z}) = \frac{\sigma^2}{m} \xrightarrow{\text{for large } m} \bar{z} \approx \mu$$

נניח שההתציפות תלויות לחלווטין – הן אוטה תצפית בדיק שחוות על עצמה יותר פעמים. מה יהיה הממוצע? התציפות עצמה. האם הקטנו את השונות של ההתפלגות המקורית? בכלל לא. נישאר עם השונות המקורית σ^2 . בלומר יש טווח מתציפות ב"ת ועד תציפות תלויות לחלווטין, והקטנה של σ^2 בהתאם.

נסתכל על מצב בינויים בטוח זהה – בו המתאים בין זוג תציפות הוא $1 < r$ בולם $Cov(z_i, z_j) = r \sigma^2$. אפשר לראות שבعة שונות ממוצע המודגם תלויות ב- r . אם $1 = r$ יש תלות מושלמת ואנחנו נשארים עם σ^2 המקורי. אם $0 = r$ שזה אומר תציפות ב"ת, נקבל את $\frac{\sigma^2}{m}$ שונות המודגם המקורי המובerta לנו.

$$Var(\bar{z}) \approx \rho \sigma^2 + (1 - \rho) \cdot \frac{\sigma^2}{m}$$

זו האינטואיטיבית שמסבירה למה רנדומ פורסט עובד. אם נצליח לחת עז ועוד דגימות עם כמה שפחות תלות (במקרה שלנו עוד ועוד עצים) נקטין את השונות המקורי של אחת מהן עד פי m . אם הדגימות תלויות חזק אחת בשניה, הרוחה מוגבל מפערת המיצוע. נרצה אם בכיה לייצר עצים שהיו שוניים כמו שייתר אחד מהשני, בר שנויה ממהמוצע שלהם.

אלגוריתם: נדריך רנדומיזציה לתהלי.

1. כל עז יראה דאטה אחר, נהוג לחת רק חלק מהנתונים (sample) או מוגבר bootstrap בגודל m המקורי עם החזרה.
2. תוך כדי בנית העצים, בכל צומת נגריל מספר מסוים של משתנים שיווו המועדים לפיזול. אם בעז המקורי בכל צומת הוא מתחשב בכל המשתנים האפשריים – העצים שלנו עושים לראות בכל צומת משתנים אחרים לחלווטין.

בשתייגע תצפית חדשה \hat{x}_0 לחיזוי – נרץ אותה בכל העצים, והחיזוי הסופי שלה יהיה הממוצע שלהם. **העצים הם כמו תציפות מודגם**, הם לא יכולים להיות לגמרי ב"ת (mbossumים על אותו הדאטה פחות או יותר), אבלណא שוויי כמה שפחות תלויות אחד בשני, ובר נרווח ממהמוצע שלהם: $(\hat{x}_0 | \hat{y}) \sim P(z_n, \dots, z_1)$.

נגדל עצים עמוקים או שטוחים? **עמוקים, שטוחים לטא יחסים מורכבים כמה שנייתן**. לעצים כאלה תהיה **שונות גבוהה, שנקטין עם המיצוע**. אם נבחר בעצים שטוחים יותר, נתחיל אולי בעוטות פחות גבוהה, אבל גם לא נרווח מספיק ממהמוצע.

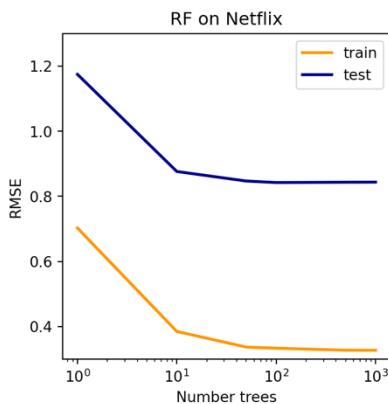
```
from sklearn.ensemble import RandomForestRegressor
```

```
ntr = NE_Xtr.shape[0]
nte = NE_Xte.shape[0]
tr_err = []
te_err = []
ntrees = [1, 10, 50, 100, 500, 1000]

for ntree in ntrees:
    RF = RandomForestRegressor(n_estimators=ntree, min_samples_split=2,
        min_samples_leaf=1, max_features=0.33, bootstrap=True)
    RF = RF.fit(NE_Xtr, NE_Ytr)
    yhat_tr = RF.predict(NE_Xtr)
    yhat = RF.predict(NE_Xte)
    tr_err.append(np.sqrt(np.sum((yhat_tr - NE_Ytr)**2) / ntr))
    te_err.append(np.sqrt(np.sum((yhat - NE_Yte)**2) / nte))
```

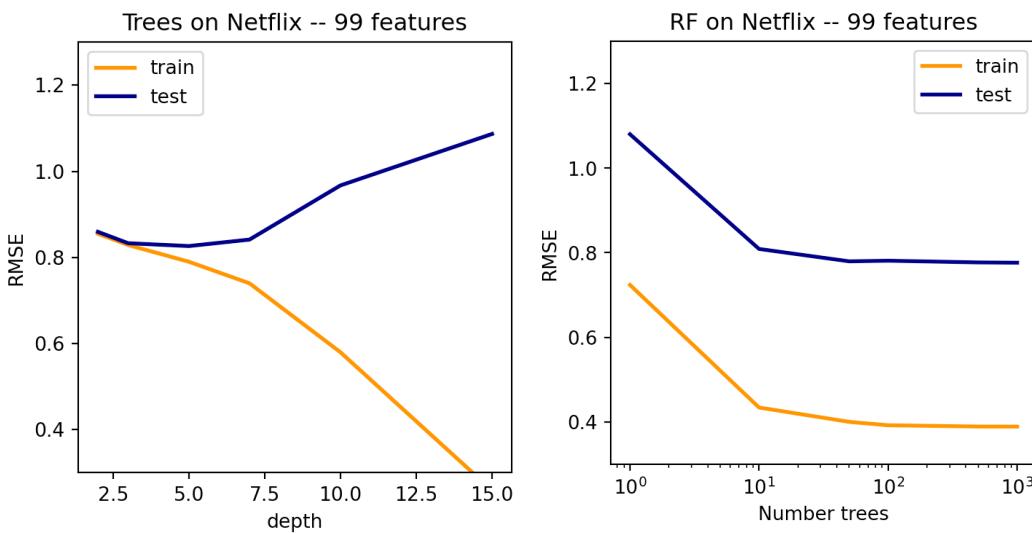
נגדיל את מספר העצים מ-1 ועד 1000. בכל איטרציה נגדל $ntree$ עצים מאוד עמוקים (באמצעות `min_samples_split=2`). פרמטר נוסף הוא `max_features=0.33`, בלומר מוגבר מוגבל בשלוש מהמשתנים (ברירת המחדל היא לחת שורש).

דוגמה על נטפליקס:



לבסוף, נזהה על מוגן המ-*train* וה-*test* ונצברות לתוכה רשיימה את ה-RMSE. ניתן לאות שמעט עצים עמוקים מגיעים לשגיאה די גבוהה על הסטסט (מעל 1), 1000 עצים מגיעים לעומת זאת לאחור-0.8. **שגיאת החיזוי לא עולה שוב! לא מיזדרת – כי השנות יכלה רק לקטון**, בולמר אין התלבטות לגבי מספר העצים. אנחנו מוגבלים רק על ידי בוח החישוב שלנו וגודל המודל הסופי על הדיסק. **בכל שנאאפשר יותר עצים, נזכה לשגיאה קטנה יותר.** ברור שיש מספר עצים כלשהו שממנו לא בטוח שיש لأن לזרת.

ננסה לעבוד עם כל הנתונים שיש בידינו, כל 99 הסרטים כך שתצפית חסירה היא בעצם דיווג 0. נראה מהנו קודם עז יחיד, ופושט נשנה את *the-h-*max_depth** של RMSE של 0.82. בשעושים רנדום פורסט עם כל 99 הסרטים, השגיאה של RMSE יורדת בבר לאחור-0.78. אם נסתכל על עז בודד عمוק, השגיאה שלו גבוהה מאד, זה רק המיצוע של עצים כאלה, שambilו אונטו לתוצאה איכותית.



נקודות לשיכום:

- שיטת רנדום פורסט משמרת את הגמישות של עצים, תוך שהיא מנסה להפחית את החיסרון הבci גדול שלהם – הנקשות, והשנות הגדולה שלהם.
- אנחנו עושים את העצים **במה שיותר שונים זה מזה**, על ידי מוגני *bootstrap* ובחרית משתנים שונים במשמעותם לכל פיצול. מלבד זה, **ודאגים שהעצים יהיו עמוקים ככל האפשר** כדי שנרוויח בכל היותר מאנפקט המיצוע: מעז בודד עם איקות חיזוי גורעה, להרבה עם איקות חיזוי טובה.
- עקרונית, ככל שנבנה יותר עצים איקות החיזוי על ה-*test* יכולת רק לקטון – יתרון ממשמעותי לשיטה. **בפועל, אנחנו מוגבלים על ידי בוח חישוב וגודל על הדיסק.** כל עז יכול להיות אובייקט די גדול, ולשמור 1000 עצים על שרתיים זה בבר לא סימפטי. יתרון נוספים. קל למקבל רנדום פורסט על פני מספר מחשבים. כל עז יכול לגודל באופן ב"ת באחרים. לכן, אם הנתונים גדולים ועומדת לרשותנו סביבת עבודה מבוזרת, ניתן הגיעו לאימון מהיר מאוד של האלגוריתם.
- בכל שיטה יש **hyperparameters** (מספר השכנים ב-KNN, או מטריקת המחק), אך בrndom פורסט אין פרמטרים שיש עליהם סימן שאלה, וזה הופך אותו לאלגוריתם מאד שימושי ופולרי. **בלי כוונון אפשר להגיע מהר לתוצאה מצוינת.**



תרגול 8 – רנדום פורסט

rndom_forest – אנסמבל של עצים:

ניקח את החסרונות של העצים וננסה לטפל בהם. במקום להסתכל על עץ אחד, נסתכל על N עצים ונמצע אותם. אי אפשר סתם לגדל את העץ שוב פעם על אותו DATA. אנחנו רוצים שהעצים יהיו כמו שייתור שונים אחד מהשני, אז יש שני אלמנטים:

1. כל עץ רואה **דאטא קצרה** – מדגם מקרי עם החזרה מאותו DATA (bootstrap).
2. בכל node בעץ, בוחרים **subset** אחר של **פיצ'רים** מתוך כולם נבחר בכל מועדים לפיזול.

זה עובד כי ככל שהעץ יהיה עמוק יותר לנו s_{bias} מאוד קטן, אבל את השונות של כל עץ נמצע וככה נוריד את השונות. בשайפה, אם יש שונות σ^2 לכל עץ, אפשר להגיד לשונות $\frac{\sigma^2}{n}$, מאותה סיבה שאנו לא יכולים תוחלת על ידי תצפית אחת אלא על ידי ממוצע.

```
df_train, df_test = train_test_split(df, test_size=0.2, random_state=4)
mtry = 3
n_samp = 120
B = 100
results = []
for i in range(B):
    BS_DF = df_train.sample(n_samp, replace = True)
    BS_X = BS_DF.drop(['Name', 'Type_1', 'Type_2', 'Total', 'Stage'], axis=1)
    BS_X = BS_X.sample(mtry, axis=1)
    print(BS_X.columns)
    BS_y = BS_DF['Stage']
    cart = DecisionTreeClassifier(max_depth=3)
    cart = cart.fit(BS_X, BS_y)
    results.append(pd.DataFrame(cart.predict(df_test.loc[:,BS_X.columns])))
results_df = pd.concat(results, axis=1)

Index(['Legendary', 'Sp_Atk', 'Sp_Def'], dtype='object')
Index(['Legendary', 'Attack', 'Sp_Def'], dtype='object')
Index(['Legendary', 'Speed', 'Defense'], dtype='object')
```

מימוש ידני: נחלק את DATA ל-*train/test*, נקרא לכמה ה-*mtry* = 3 ו-*B* = 100 = מספר העצים. כל פעם נציג 3 משתנים אחרים.

- הדגימה של כל הדאטא מראש עם החזרה – זה אכן רנדום פורסט.
- מה שאנו עושים בדגימה של 3 עמדות, זה לא מה שאמרנו **שקרה**. מתי נציגות 3 העמדות ברנדום פורסט? בכל פיזול! בلومר במימוש הידני זה לא בדוק אותן דבר.

```
print(df_test.shape)
print(df_test.head(2))
print(results_df.shape)
print(results_df.head(2))
```

```
(31, 12)
      Name Type_1 Type_2 Total  HP  Attack  Defense  Sp_Atk  Sp_Def \
118 Seaking   Water    NaN  450   80      92       65      65      80
18 Rattata  Normal    NaN  253   30      56       35      25      35

      Speed  Stage Legendary
118      68     2    False
18      72     1    False
(31, 100)
      0  0  0  0  0  0  0  0  ...  0  0  0  0  0  0  0  0  0  0  0
0  3  2  2  2  2  2  2  3  3  2  ...  2  2  2  2  2  2  3  2  2  2
1  1  1  1  1  1  1  1  1  1  1  ...  1  1  1  1  1  1  1  1  1  1  1
```

נקבל לבב תצפית 100 חיזויים (כל עץ מספק לנו חיזוי אחר). כדי לעשות חיזוי סופי, נסתכל מה ה-majority (mode). למשל, ניקח את השביב (mode).

```
preds = results_df.mode(axis=1)
preds.head()
```

0	2
0	2
1	1
2	2
3	1
4	1

נבע סיווג באמצעות רנדום פורסט (המחלקה Classifier, גם כאן יש את המחלקה Regressor בהתאם).

```
from sklearn.ensemble import RandomForestClassifier
X = df.drop(['Name', 'Type_1', 'Type_2', 'Total', 'Stage'], axis=1)
y = df['Stage']
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.2,
                                                    random_state=4)
rf = RandomForestClassifier(n_estimators=8, max_features=mtry, random_state=4)
rf.fit(X_train, y_train)
y_pred = rf.predict(X_test)
print(classification_report(y_test, y_pred, target_names=target_names))
print('Test Accuracy: {:.3f}'.format(rf.score(X_test,y_test)))

      precision    recall  f1-score   support

Stage_1      0.75     0.86     0.80      14
Stage_2      0.77     0.77     0.77      13
Stage_3      0.50     0.25     0.33       4

   accuracy          0.74      31
  macro avg      0.67     0.63     0.63      31
weighted avg      0.73     0.74     0.73      31

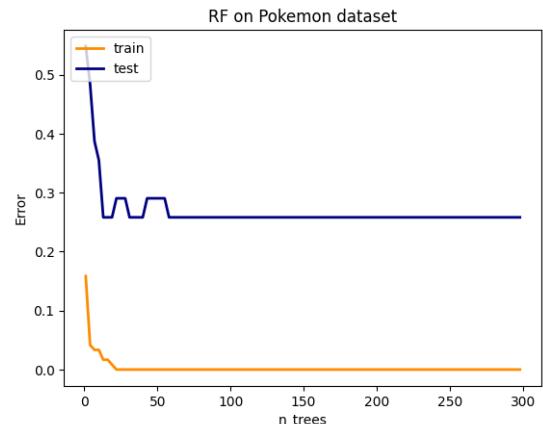
Test Accuracy: 0.742
```



נבדוק את הדיק על במגוון עצים שונים (B). ברגעום פורסט אנחנו מצפים ואכן מקבלים, במשמעות, השונות תקן – כל העצים חסרי הטיה והם נוראים עמוקים. ככל שנעשה יותר ויתר עצים נקבל מודל טוב. בפועל, זה מתכנס לאנשא – יש גבול.

```
n_trees = np.arange(1,300,3)
te_err, tr_err = [], []
for B in n_trees:
    rf = RandomForestClassifier(n_estimators=B, max_features=mtry, random_state=4)
    rf.fit(X_train, y_train)
    tr_err.append(1 - rf.score(X_train,y_train))
    te_err.append(1 - rf.score(X_test,y_test))

plt.figure()
plt.plot(n_trees, tr_err, color='darkorange', lw=2, label='train' )
plt.plot(n_trees, te_err, color='navy', lw=2, label='test' )
plt.xlabel('n_trees')
plt.ylabel('Error')
plt.title('RF on Pokemon dataset')
plt.legend(loc='upper left')
plt.show()
```



בעיה ברגעום פורסט – המונ זיכרנו! כל עץ כזה במיומש אפילו יחסית רזה, שוקל הרבה (יותר מ-1MB), ועוד 1000 עצים זה בגד.

בostoning

בostoning: בostoning הוא גם אנסמבל של מודלים, אבל שונה בחולוין בכך שבו אנחנו מגדלים את תת המודלים וממצעים אותם. הוא לא חייב להיות מבוסס על עצים אבל בפועל זה המימוש הפופולרי ביותר.

איןטואיטיבית, בניית מודל די מורכב אבל בהדרגה – בניית מודל פשוט, ואז נשפר אותו באמצעות מודל פשוט נוסף, ואז נשפר את הביצועים של השניים עם מודל שלישי וכך הלאה. בר קובל מודל שהוא קומבינציה של הרבה מודלים פשוטים (weak learners), והמודל שמתקבל הוא כבר לא בר קובל פשוט.

מה הופך עץ למודל weak learner? אם נעשו אותו יחסית שטוח. בעומק 1 – שאלת אחת על התצפית, בעומק 2 – שתי שאלות. לא יכול להיווצר מודה מודל מורכב. איך זה שונה מרגעום פורסט? הסתכלנו על עצים חזקים ועמוסים, לא חלשים. כל עץ היה ב"ת משאר העצים ואפשר לגдел אותם במקביל. כן אי אפשר לעשות את זה. העץ השני חייב לדעת מה הביצועים של העץ הראשון כדי לשפר אותו. איך נדאג שהמודל הפשוט הביא לשפר את מה שקדם לו?

אלגוריתם:

1. נתחל בחיזוי בסיסי זהה לכל התציפות $0 = F^{(0)}(x)$.
2. בשלב $t \geq 1$:
 - a. נגדיר את הוקטור $\hat{Y} = (y_1^{(t)}, \dots, y_n^{(t)})$ שתווסף את מה שהמודל עד כה $F^{(t-1)}$ לא הצליח לתפוס.
 - b. נתאים weak learner חדש $\hat{f}^{(t)}$ על מדגם הלמידה שלנו $(X, Y^{(t)}) = T^{(t)}$. בשלב ה- t העץ הפשוט ינסה להתאים את עצמו בדומה לשוויתו $\hat{f}^{(t)}(\hat{Y})$ ולא \hat{Y} המקורי.
 - c. נעדכן את המודל החדש $F^{(t-1)} + \varepsilon \hat{f}^{(t)}$. ברגורסיה, הכוונה בעצם לקחת את החיזוי של עץ לכל תצפית עד כה ולהוסיף את החיזוי החדש כפול משקלות קטנה.

באיזה מודל פשוט להשתמש? איך לקבוע את \hat{Y} ? ואיך נחליט מהו וקטור \hat{Y} שאמור לבטא מה שהמודל עד כה לא הצליח לפחות. ברגורסיה למשל יש לנו עז מקורי ו- \hat{Y} שהמודל הצליח לחזות עד כה. איך נקבע את מה שהמודל לא הצליח לחזות? השארית $y - \hat{Y}$. אכן, **نمدل את השארית:** $(y_i - F^{(t-1)})(x_i)$ העץ שלנו יהיה בדרך כלל בעומק 3-2. ה- ε יהיה כמה שיותר קטן: אפשר לראות בו קצב הלמידה שלנו – נוסיף את המודל שלמדו עם משקלות קטנה (ニיח אוטו בערבון מוגבל).

$$\text{החיזוי הסופי יהיה: } \hat{Y} = F^{(0)}(x) + \sum_{t=1}^T \varepsilon \hat{f}^{(t)}(x)$$

```

ntr = NE_Xtr_noNAN.shape[0]
nte = NE_Xte_noNAN.shape[0]
tr_err = []
te_err = []

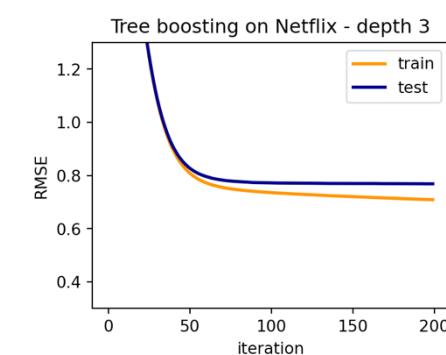
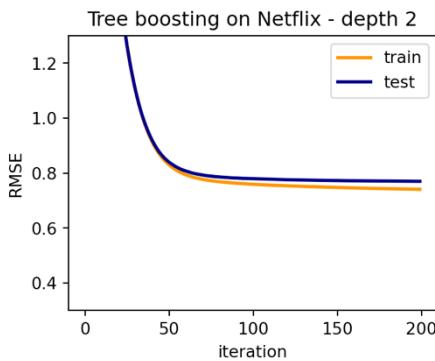
Ytr_now = NE_Ytr
yhat_tr = np.zeros(ntr)
yhat_te = np.zeros(nte)
eps = 0.05

```

```

for iter_num in range(200):
    tree = DecisionTreeRegressor(max_depth = 2)
    tree.fit(NE_Xtr_noNAN, Ytr_now)
    yhat_tr_now = tree.predict(NE_Xtr_noNAN)
    yhat_te_now = tree.predict(NE_Xte_noNAN)
    yhat_tr += eps * yhat_tr_now
    yhat_te += eps * yhat_te_now
    tr_err.append(np.sqrt(np.sum((yhat_tr - NE_Ytr)**2) / ntr))
    te_err.append(np.sqrt(np.sum((yhat_te - NE_Yte)**2) / nte))
    Ytr_now = NE_Ytr - yhat_tr

```



מימוש ידני:

נקבע את y_{train} להיות הווקטור הנוכחי, ובשאר הווקטורים נאחל \hat{y} . נקבע $\epsilon = 0.05$. בונה 200 עצים, כל פעם עז שטוח בעمق 2, ונמלל לא את ה- y המקורי אלא את y_{now} . y_{train} על $train$ - $test$, ולחיזיו שלנו מוסיף את $y_{train} = \hat{y} + \epsilon$. לבסוף נבצע את העדכון $\hat{y} = y - y_{train}$.

בשאנחנו מציררים את טעות החיזוי (RMSE), אנחנו רואים צפויו ירידה בכל שימושים עוד עצים, עד אסימפטוטה בלשיה. אחרי העז-200 כבר נראה RMSE של 0.77 (עבור הנתונים האלה זו תוצאה איבוטית מאוד).

יש רק הבדל קטן בין החיזוי של $train$ וה- $test$, בין **שאנחנו בונים מראש עצים שטוחים שלא עושים overfitting לנ נתונים**.

הרבבה חושבים שהעובדת ש- ϵ זהה לכל העצים אומרת שאנחנו משלק זהה לכלם: יש גרסאות לבוסטיינג בהן משתמשים ב- ϵ שונה לבכל t . אפילו אם הוא זהה, לא ממש מדובר במשקלות שווה. העז הראשון ייחזה ב很漂亮 יחסית גדולה (ע' המקורי), וכל עז שבאו אחריו ייחזה שארית שתלך ותיעשה קטנה יותר. **אולי כל עז מוכפל פי ϵ , אבל ההשפעה שלהם במודל הגדל שונה לगמרי. העצים הראשונים ישפיעו הרבה יותר על החיזוי מהעז ה-200.**

עומק העז: מתבקש לנסתה להעמיק בכך את העז, הבדל היחיד הוא שאנחנו מניטים עמוק מקסימלי של 3. עדין עצים שטוחים יחסית. **זה נראה בהתחלה שאין הבדל בין עומק 2 לעומק 3, אבל אם נשווה, נראה שעקבות ה- $train$ - $test$ נמוכה יותר מה- $test$ עבו רום 3, יש יותר overfitting.** ניתן לראות לפי סימולציה שבוסטיינג עם learners שאינם לבאורה (עומק 15), יש overfitting קיצוני למדגם הלמידה.

מבנה מתמטי:

אם רנו כי $y^{(t)}$ צריך לייצג את מה שלא הסבכנו עד עכשיו, וברגרסיה אינטואיטיבית לקחנו את השארית. מה לגבי סיווג? ננסה לא לחשב על העצים בעומק 2 במודלים, אלא כפיצ'רים. בדומה לרגרסיה לינארית/לוגיסטיבית נניח שיש q פיצ'רים בכלל:

$$h_1(x), \dots, h_q(x)$$

המודל הסופי מאד מזכיר רגרסיה: **צ"ל של הפיצ'רים האלה, כל אחד מקבל משקלות $\hat{\beta}_k$:**

$$\hat{f}(x) = \sum_{k=1}^q \hat{\beta}_k h_k(x)$$

אמנם, העצים האלה לא באמת נתונים, אנחנו לא במצב של רגרסיה לינארית. אי אפשר למצאו יישורות את הווקטור $\hat{\beta} \in \mathbb{R}^q$ כי $\hat{\beta}$ הוא עצום וכך הכמות של העצים שניתן לבנות.

היעון הוא למצאו אותו בצורה אדפיטיבית, חמדנית – לחפש כל פעם את המשתנה הבא h_{k_t} , ולהוציא אותו יחד עם משקלות $\hat{\beta}$. במקרה שלנו $\{k\} = k_t = \#\{k : \epsilon = \hat{\beta}\}$. אחרי T איטרציות, עבר העז הספציפי h יהיה ϵ כפול מספר הפעמים שהוא נבחר. אם יש מיליארד עצים אפשריים,Robots לא יבחרו משקלות 0, חלקם יקבלו משקלות ϵ ואולי כמה בודדים יקבלו 2 כי נבחרו פעמיים. ברור שזה לא וקטור ה- $\hat{\beta}$ שנותן את הצ"ל הבci טוב שניתן להたちים במרחב העצים, אלא זהה שנבנה בצורה אדטטיבית.



מהו העץ בכלל שלב שנבחר לצרף לצ'ל? זה העץ שמשפר את המודל הכי הרבה: מודל שכשנוסף $F^{(t-1)}$ כפול החיזוי שלו, זה יקטין את פונקציית הפסד שלו על המודל הנוכחי $F^{(t-1)}$, למשל RSS. אנחנו רוצים להקטין פונקציה כמו שיטור, וזה מושגים לרוב על ידי ירידת צעד קטן במודול הנגזרת (הградיאנט) של הפונקציה שלנו.

$$\frac{\partial \text{RSS}(F^{(t-1)})}{\partial \hat{y}_i} \Big|_{\hat{y}_i=F^{(t-1)}(x_i)} = -2(y_i - F^{(t-1)}(x_i))$$

$$\text{RSS} = (y_i - \hat{y}_i)^2$$

בRiboux: כל פעם שאנו מוסיפים כפול המודל שחוזה את השאריות הכי טוב, אנחנו הולכים עד קטע במודול הגרדיאנט – מקטינים את loss. לכן אנחנו קוראים ל- ϵ צעד למידה, חלק מהתהילן האופטימיזציה שהמבצע כאן.

הכללה: ניקח את loss ובכל איטרציה נקטין אותו, על ידי בחירת ה-weak learner שימדיל את הגרדיאנט השילי של הכי טוב. נסיף כפול החיזוי הזה – נלך צעד ϵ בכיוון שבו loss יורד הכי מהר (כיוון הגרדיאנט).

דוגמה על נטפליקס:

نبחר הפסד squared_error, נגדיר מספר עצים $n_{estimators}=200$, עומק $max_depth=3$. נקבל שגיאת חיזוי דומה מאוד לשימולציה שלנו, באופן לא מפתיע.

```
from sklearn.ensemble import GradientBoostingRegressor

GBR = GradientBoostingRegressor(loss='squared_error', learning_rate=0.05,
                                 n_estimators=200, max_depth=3)
GBR.fit(NE_Xtr_noNAN, NE_Ytr)

yhat_tr = GBR.predict(NE_Xtr_noNAN)
yhat_te = GBR.predict(NE_Xte_noNAN)

RMSE_tr = np.sqrt(np.sum((yhat_tr - NE_Ytr)**2) / ntr)
RMSE_te = np.sqrt(np.sum((yhat_te - NE_Yte)**2) / nte)
print(f'200 trees, depth 3: train RMSE: {RMSE_tr:.2f}, test RMSE: {RMSE_te:.2f}')
200 trees, depth 3: train RMSE: 0.71, test RMSE: 0.77
```

סיכום מודלי אנסטמל:

- RF שואף לבנות בובת הרבה עצים שונים זה מזה בכל שניתן. אנחנו רוצים עצים גדולים, עמוקים, שתהייה להם שונות בגובהה, ובכל שנאמן יותר עצים אנחנו יכולים רק להקטין את שגיאת החיזוי על נתוני ה-test.
- ב-Boosting העצים נבנים אדפטיבית, והם צריכים להיות לא עמוקים. נעדיף למידה איטית בכל האפשר, לשמר ϵ נמוך.
- בפועל אנחנו רואים שעוד ועוד עצים משפרים את טעות החיזוי.
- שתי השיטות לוקחות מודלים מוגבלים (כמו עצים) ועשויות להם קומביינציה למודל טוב ומורכב. שתיהן מאוד פופולריות, כי לא צריך לזכור יותר מדי כדי להשג ביצועים טובים.
- משאבים: מבחינת מקבול – Boosting מאתגר יותר כי כל עץ צריך להתחשב באלה שקדמו לו. מבחינת גודל על הדיסק – Boosting יביא למודלים קטנים יותר שאפשר לשימושם על כמה שרטטים בו זמנית, כי הוא משתמש בעצים שטוחים יותר. סיבה זו טובה להעדיף אותו על RF.



תרגול 9 – בוסטיניג

אם ברכdom פורסט אנחנו מאמנים הרבה עצים ובסופ ממצאים אותם, בbossting אנחנו עושים את זה בצורה אדפטיבית. אנחנו מתמקדים ברגסיה לינארית (הרטזון בסיג'ג די דומה). זה נקרא Gradient Boosting כי אנחנו הולכים כל פעם צעד בביון הגראדיאנט של RSS תוך כדי האלגוריתם.

```
# new imports from sklearn
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV

gbr = GradientBoostingRegressor()
params = {
    'max_depth': range(1, 11),
    'learning_rate': [0.0001, 0.001, 0.01, 0.1],
    'n_estimators': [100, 200, 500, 1000],
    'subsample': [0.2, 0.4, 0.6, 0.8, 1.0]
}
print(f'Total search space... {np.prod([len(v) for v in params.values()])} combinations.')
# this is a silly little dataset, we could easily do 100 iterations on 5 folds
rscv = RandomizedSearchCV(gbr, params, n_iter = 100, cv = 5, scoring = 'neg_mean_squared_error', random_state=0)
search = rscv.fit(X_train, y_train)

# see best params
search.best_params_

Total search space... 800 combinations.
{'subsample': 0.2, 'n_estimators': 200, 'max_depth': 4, 'learning_rate': 0.01}
```

search.best_estimator_

```
GradientBoostingRegressor
GradientBoostingRegressor(learning_rate=0.01, max_depth=4, n_estimators=200,
subsample=0.2)
```

נסתכל על הדאטא של פוקימון, וננסה לחזות את ערך ה-HP. נחקק ל-`train/test`. נעשה tuning לפרמטר `max_depth`. ניעזר בכך בחלוקת. ניעזר ב-`GridSearchCV`, `RandomizedSearchCV` השגיאה על ה-`train-test`, שהוא ה-`MSE`. באופן צפוי, שגיאת הלמידה יורדת לחלוון, ועל הטסט יש בקרובו U, יש sweet spot עבור `max_depth=6`. שיטה נוספת לבצע tuning של יותר פרמטרים היא באמצעות `CV`.

QUIZ 11

שאלה 1: מהו סדר הצעדים הנכון באלגוריתם של בוסטיניג לרוגסיה, כפי שנלמד בשיעור?

- | | |
|----------------------------|---|
| 1 | לחיזות לכל התצפיות 0 |
| לא שיין לאלגוריתם בוסטיניג | למצע את החיזוי של כל העצים |
| 3 | להתחאים מודל "חלש" לחיזוי השאריות הנוכחיות באמצעות X |
| לא שיין לאלגוריתם בוסטיניג | להגריל מספר משתנים לחיזוי המודל הנוכחי |
| 4 | לעדק את החיזוי הנוכחי באמצעות הוספה קבוע אפסילון כפול החיזוי החדש |
| 2 | לחשב את השאריות בין החיזוי עד כה לתצפיות האמיתיות |
| 5 | אם לא הגיעו למספר המודלים "חלשים" הנתון, לחזור לשלב 2 |

שאלה 2: מה מבנים רנדומיזציה לרכיבים פורסט?

- רנדום פורסט בוחר באקראי תת-קובוצה של מודלים "חלשים" מהת קבוצה גדולה, וממצע אותם
- **כל עץ ב"ער"** משתמש בתת-קובוצה אקראיית של משתנים בכל פיזול ותת-מדדgm שונה
- כל עץ מסתכל על שארית החיזוי עד בה וمعدן את המודל בהתאם
- **כל עץ ב"ער"** משתמש בתת-קובוצה אקראיית של משתנים

שאלה 3: ככל נצפה שככל שרכיבים פורסט ישתמש ביוטר עצים שגיאת החיזוי על מדגם הטסט תקטן. אנחנו מוגבלים בעיקר על-ידי הגודל הפיסי של המודל, זמן האימון והחיזוי. **נכון**, טעות החיזוי לא בהכרה תרד לאפס אבל היא לא-עלוה בכלל, אבל לאחסן בסביבת פרודקشن אפילו 1000 עצים עמוקים על נתונים גדולים עלול להיות פשוט גדול מדי לחיזוי מהיר ויעיל בזמן.



4 – רשותת נירוניים

מבוא לרשותת נירוניים

חזקה על רגרסיה לוגיסטיות

חזקה על רגרסיה לוגיסטיות: מזכיר במודל הרגרסיה הלוגיסטי. אנו צופים ב- x זוגות של y , x כאשר $\{x_i \in \mathbb{R}^q, y_i \in \{0,1\}\}$ ביןاري, ו-

$$Y_i | X_i \sim Ber(p_i); \mathbb{E}[Y_i | X_i = x_i] = P[Y_i = 1 | X_i = x_i] = p_i$$

אחר שהתחולת של משתנה זהה היא p עצמה, זה מה שנחנו בסופו של דבר מודלים. הבעה היא שהסתברות היא כמות בין 0 ל-1. נבחר פונקציית link לתוחלת, במקורה שלם logit $\log(1/(1-p))$ יחס הסיכויים, ואת הפונקציה הזאת נמדל באמצעות מודל לינארי רגיל, β הוא וקטור המקבדים ואותו אנחנו רוצים למצאו:

$$g(\mathbb{E}[Y_i | X_i = x_i]) = \text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = x_i' \beta$$

איך מוקטור המקבדים נחזיר לקבל הסתברות? באמצעות הפונקציה ההופכית: $g^{-1}(x_i \beta) = \hat{p} = \frac{1}{1+e^{-x_i \beta}}$. כאשר נשיג אומדן לוקטור המקבדים שלנו, נוכל להסביר את β , את הסיכוי ש- Y_i יהיה 1. הכוון והגודל של כל ריבוב בוקטור המקבדים, יחדו לנו מהו התורמה של המשתנה המתאים, log-odds (log-odds) y_i ו��ל לחזות עבור תצפית חדשה x , את ההסתברות לקבל עבורה $Y_i = 1$. אם נרצה חיזוי סופי נctrar להשוו את ההסתברות הנחוצה למספר מסוים (cutoff) למשל 0.5.

נראות מקסימלית: איך נמצא אומדן לוקטור המקבדים β ? הגישה הסטנדרטית היא למקסם פונקציה בשם **הנראות** המסומנת כרך:

$$L(\beta | X, y) = \prod_{i=1}^n P[Y_i | X; \beta] = \prod_{i=1}^n P[Y_i = 1 | X; \beta]^y_i \cdot P[Y_i = 0 | X; \beta]^{1-y_i}$$

בනחת או תלות בין התציפות, מדובר במכפלת ההסתברויות לקבל כל תצפית ותצפית. אם $y_i = 1$ נכתוב את ההסתברות בחזקת y_i , ואחרת נכתוב בחזקת $1 - y_i$. מהו אותה הסתברות לפיה המודל? הפונקציה ההופכית של g .

$$L(\beta | X, y) = \prod_{i=1}^n (g^{-1}(x_i \beta))^y_i \cdot (1 - g^{-1}(x_i \beta))^{1-y_i}$$

הוקטור $\hat{\beta}$ הוא הוקטור שimaxם את הנראות, וכך נקרא אומדן נראות מקסימלי – **MLE** (maximum likelihood estimator). פונקציית הנראות כפי שהוא מנוסחת ברגע, היא קשה למקסום. נהוג לעבוד עם log הנראות, מה שambilנו את סכום הלוגריתמים של ההסתברויות שרשמננו. נפשט את הפונקציה אפילו יותר כדי לקבל פונקציה שקל לגזר לפיה.

$$\begin{aligned} \ell(\beta | X, y) &= \log(L(\beta | X, y)) = \sum_{i=1}^n \ln(P[Y_i | X; \beta]) = \sum_{i=1}^n y_i \ln(g^{-1}(x_i \beta)) + (1 - y_i) \ln(1 - g^{-1}(x_i \beta)) = \\ &\sum_{i=1}^n \ln(1 - g^{-1}(x_i \beta)) + y_i \ln\left(\frac{g^{-1}(x_i \beta)}{1 - g^{-1}(x_i \beta)}\right) = \sum_{i=1}^n -\ln(1 + e^{-x_i \beta}) + y_i x_i \beta \end{aligned}$$

Differentiate:

$$\frac{\partial \ell(\beta | X, y)}{\partial \beta_j} = \sum_{i=1}^n -\frac{1}{1+e^{x_i \beta}} e^{x_i \beta} x_{ij} + y_i x_{ij} = \sum_{i=1}^n x_{ij} (y_i - g^{-1}(x_i \beta))$$

Or in vector notation:

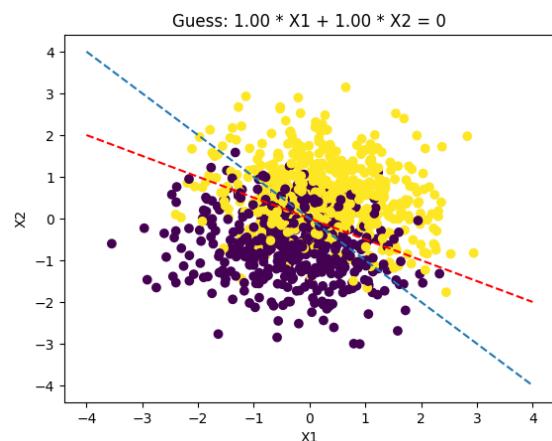
$$\frac{\partial \ell(\beta | X, y)}{\partial \beta} = X'(y - g^{-1}(X\beta)), \text{ where } X \text{ is the } n \times q \text{ data matrix.}$$

נגזר את הפונקציה לפיה אלמנט אחד β_j , ונכתוב בכתיב וקטורי. את הביטוי הזה היינו חיצים להשווות לוקטור האפס אבל לא קיבל פתרון סגור. בשלב זה ניבור לשיטות של אונלייזה כמו נווטון-ראפסון. פונקציה זו היא קמורה ואניינה נחשבת קשה. אנחנו נסתכל על פתרון של מודד הגרדיאנט, וביצע אופטימיזציה.

אלגוריתם GD: במקום למסום את $\log(\text{הנראות})$, נעשה מינימיזציה לוג הנראות השילית, מה שיקל על האופטימיזציה:
 $(\beta - \ell - \alpha) / q$. נתחיל בInitialization וASHOT עבור $\hat{\beta}$. הגרדיאנט מצביע לכיוון שיפוע הירידה התוליה בנקודת $\hat{\beta}$, וכן צעד קטן בגודל α בכיוון זה (כਮון אנחנו הולכים **נגד כיוון הגרדיאנט** כי אנחנו רוצים ללבת בכיוון המינימום ולא המקסימום):

$$\hat{\beta}_{t+1} = \hat{\beta}_t - \alpha_t \nabla f(\hat{\beta}_t)$$

```
n = 1000
q = 2
X = np.random.normal(size = n * q).reshape((n, q))
beta = [1.0, 2.0]
p = 1 / (1 + np.exp(-np.dot(X, beta)))
y = np.random.binomial(1, p, size = n)
```



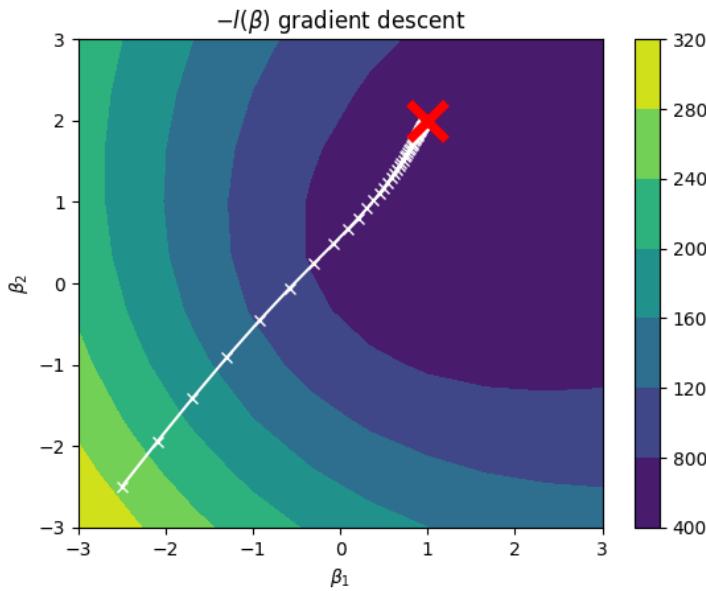
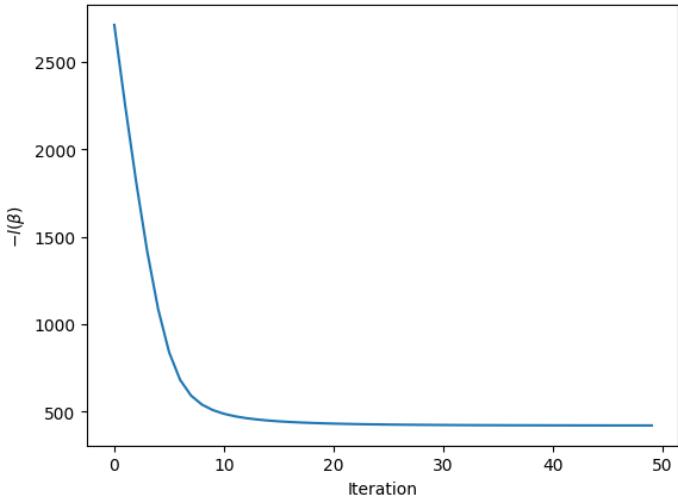
```
# let's see NLL -l(beta) through iterations
alpha = 0.001
beta_hat = np.array([-2.5, -2.5])
betas = [beta_hat]
ls = []
for i in range(50):
    p_hat = 1 / (1 + np.exp(-np.dot(X, beta_hat)))
    nll = -np.sum(y * np.log(p_hat) + (1 - y) * np.log(1 - p_hat))
    ls.append(nll)
    grad = -np.dot(X.T, (y - p_hat))
    beta_hat = beta_hat - alpha * grad
    betas.append(beta_hat)
plt.plot(range(50), ls)
plt.xlabel("Iteration")
plt.ylabel(r"$-\mathcal{L}(\beta)$")
plt.show()
```

נעsha זאת במשך מספר איטרציות עד להתקנות למינימום מקומי. במקרים מסוימים, אפשר להיות די בטוחים שמדובר בנקודת מינימום גלובלית. נראות לכך זה עובד בפייטון: אנחנו מגדים מצב אידיאלי לרגרסיה לוגיסטיות עם 2 משתנים ולא חווון. יש לנו $1000 = n$ תצפיות ו-2 = q משתנים. וקוטו המקדים האמיתי הוא (1,2), וקטור ההסתברויות מחושב לפי פונקציית g הפוכה.

בשאנו מושרטים בתחילת הסימולציה, אנחנו רואים את הקו האידיאלי שב- $\beta_2 = 1, \beta_1 = 1$. זהו קו ההחלטה האידיאלי שmaresיד בין תצפיות ה-0 לבין תצפיות ה-1.

נתחיל בInitialization התחורי (1,1). הקו הכהול מייצג את הקו הנוכחי, והאדום את הקו האידיאלי. לאחר איטרציה אחת של GD נקבל עדכון ל- $\hat{\beta}$ והקו הכהול מתרחק לאדם. נעשה עוד 10 איטרציות, ובו koko הכהול כבר מתחזק עם האדם, והחיזוק מתקרב מאוד ליקטור אמיתי.

וככל גם לראות את $\log(\text{הנראות})$ לאורך האיטרציות, ואכן הפסד יורד.



לבסוף, ממש נציר את הנראות השילית, ואיך שהוא יורד במרחב הפרמטרים β_2, β_1 . אנחנו מתחילה מנקודת לא סבירה. הנראות השילית מיוצגת בgraf kontorim, ובכל צעד אנחנו מתקרדים לעבר נקודת המינימום שהוא אכן באזור (1,2).

מරגרסיה לוגיסטיות לרשת נוירונים

נראה כי וגרסיה לוגיסטיות היא רשת נוירונים פשוטה למדי.

1. Call our $-l(\beta)$ “Cross Entropy”
2. Call $g^{-1}(X\beta)$ the “Sigmoid Function”
3. Call computing \hat{p}_i and $-l(\hat{\beta})$ a “Forward Propagation” or “Feed Forward” step
4. Call the differentiation of $-l(\hat{\beta})$ a “Backward Propagation” step
5. Call our β vector $W_{(q+1) \times 1}$, a weight matrix (add intercept, call it “bias”)
6. Add *stochastic* gradient descent (SGD)
7. Draw a diagram with circles and arrows, call these “neurons”

Cross Entropy: אפשר לראות בכך מכך למרחק בין שתי התפלגיות P, Q . במקרה של פונקציית הסתברות לוגית, כפול לוג ההסתברות של ההתפלגות האחרת. אפשרים, וכן CE מקבלת צורה של סכום פשוט של הסתברות לפי ההתפלגות אחת, כפול לוג ההסתברות של ההתפלגות האחרת.

$$H(P, Q) = -E_P[\log(Q)] = -\sum_{x \in \mathcal{X}} P(x) \log[Q(x)]$$

אם נחשב על שתי ההתפלגיות שלנו כפונקציית ההסתברות של γ הבינארי שלנו, ו- γ עצמו, יוכל להוכיח שה-CE הוא בדיק פונקציית לוג הנראות השילנית שלהם (NLL).

In case X has two categories, and $p_1 = P(X = x_1), p_2 = 1 - p_1$ and same for q_1, q_2 :

$$H(P, Q) = -[p_1 \log(q_1) + (1 - p_1) \log(1 - q_1)]$$

If we let $p_1 = y_i$ and $q_1 = \hat{p}_i = g^{-1}(x_i \hat{\beta})$ we get:

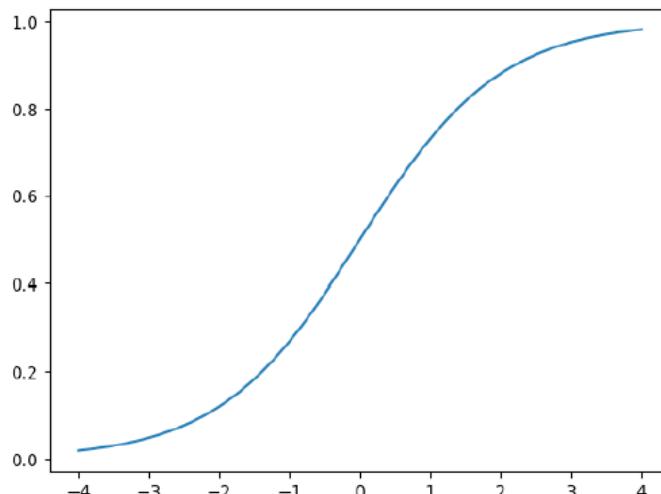
$$\begin{aligned} H(y_i, \hat{p}_i) &= -[y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)] = \\ &= -[y_i \ln[g^{-1}(x_i \hat{\beta})] + (1 - y_i) \ln[1 - g^{-1}(x_i \hat{\beta})]] \end{aligned}$$

Which is exactly the contribution of the i th observation to the NLL $-l(\hat{\beta})$.

Sigmoid function: נשנה את שם הפונקציה ההופכית של g לפונקציית sigmoid אשר מקבלת כל קלט בטווח $(-\infty, \infty)$ וממפה אותו להיות בטווח $(-1, 1)$.

$$g(p) = \text{logit}(p) = \log\left(\frac{p}{1-p}\right); \quad g^{-1}(z) = \sigma(z) = \frac{1}{1+e^{-z}}$$

$$\text{So: } g^{-1}(g(p)) = \sigma(\text{logit}(p)) = p$$



אפשר לומר שכל איטרציה ב-GD כולה: [Forward/backward propagation](#)

```
# forward step
p_hat = 1 / (1 + np.exp(-np.dot(X, beta_hat)))
nll = -np.sum(y * np.log(p_hat) + (1 - y) * np.log(1 - p_hat))
# backward step
grad = -np.dot(X.T, (y - p_hat))
# descent
beta_hat = beta_hat - alpha * grad
```

1. שלב forward (חישוב פונקציית הלוג נראות והחיזוי של \hat{p}).
2. שלב backward (חישוב הגרדיינט).
3. הליכה בכיוון הגרדיינט.

במקרה שלנו גזרת פונקציית הלוג נראות השילית לפיקטור המקדים β הייתה יחסית פשוטה – הענו לפתרון פשוט גם אם לא פתרון סגור. ככל שהאריכת טוירה של רשת הנוירונים תלך ותיעשה מורכבת יותר, נדרש להכפיל את הגזירה הזאת לצעדים נוספים. כאן ניעזר בכל השרשרת. יש בכך מכפלה לאחרו של נגזרות.

Recall that according to the Chain Rule, if $y = y(x) = f(g(h(x)))$ then:

$$y'(x) = f'(g(h(x))) \cdot g'(h(x)) \cdot h'(x)$$

Or if you prefer, if $z = z(x)$; $u = u(z)$; $y = y(u)$ then: $\frac{dy}{dx} = \frac{dy}{du} \cdot \frac{du}{dz} \cdot \frac{dz}{dx}$

כדי שנוכל להשתמש בכל השרשרת, נרשום את ה-LLN בתוור הריבבה של פונקציות. רכיב אחד שלו הוא $\ell_i(p_i(z_i(\beta)))$ כאשר:

Each of these is simpler to calculate:

$$\begin{aligned}\frac{\partial l_i}{\partial p_i} &= \frac{y_i - p_i}{p_i(1-p_i)} \\ \frac{\partial p_i}{\partial z_i} &= p_i(1 - p_i) \\ \frac{\partial z_i}{\partial \beta_j} &= x_{ij}\end{aligned}$$

And so:

$$-\frac{\partial l(\beta)}{\partial \beta_j} = -\sum_i \frac{y_i - p_i}{p_i(1-p_i)} \cdot p_i(1 - p_i) \cdot x_{ij}$$

And we could differentiate using the chain rule like so:

$$-\frac{\partial l(\beta)}{\partial \beta_j} = -\sum_i \frac{\partial l_i}{\partial p_i} \cdot \frac{\partial p_i}{\partial z_i} \cdot \frac{\partial z_i}{\partial \beta_j}$$

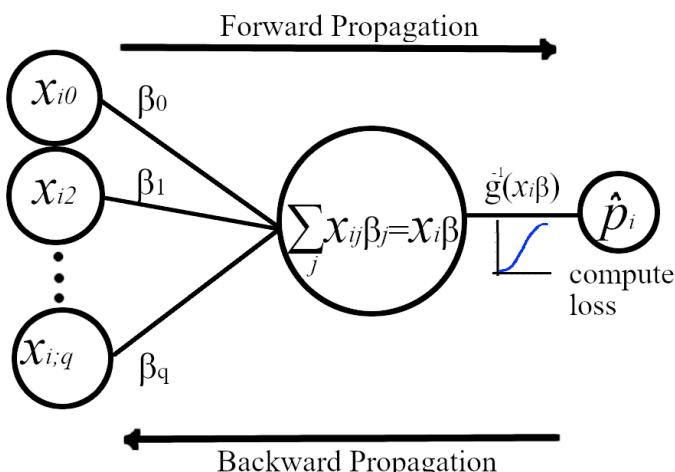
$$\begin{aligned}z_i &= z(\beta) = x_i \beta \\ p_i &= g^{-1}(z_i) = \frac{1}{1+e^{-z_i}} \\ \ell_i &= \ell(p_i) = y_i \ln(p_i) + (1 - y_i) \ln(1 - p_i)\end{aligned}$$

סה"כ נקבל:

$$NLL = -\ell(\beta) = -\sum_{i=1}^n \ell_i(p_i(z_i(\beta)))$$

במחברת הפיתון שלנו: בעת נමמש את הרגרסיה הלוגיסטי באמצעות `forward()` מחשבת שלב אחר שלב את כמה פונקציות פשוטות: `backward()` מחשבת בשני צעדים הסתברויות החזיות ואת ה-LLN. (`backward()` מקבל את הנגזרות ותכפיל בינהן. בעטוף את ביצוע ה-GD עם עוד שתי פונקציות נוספת. לבסוף הפונקציה `compute_loss()` מקבלת את הדאטה Y ואת במודות האיטרציות `epochs`.

SGD: רשותת נוירונים יודעת לזרע על קבועים נתונים גדולים מאוד. אחד מהאופןם שבהן הן עושות את זה, הוא על ידי כך שהגרדיינט לא מוחש בכלי צעד על מסד הנתונים. ברגע שהרשת מורכבת יותר, חישוב הגרדיינט בכל איטרציה יגוזל זמן וכוח חישוב. לכן, נהוג לאמן על `epoch` נחלק את הדאטה למספר תת-مدגים אקראיים בגודל `batch_size` ונдин אותם אחד אחרי השני לרשת. האקריאיות הופכת את ה-GD להיות SGD.



רשת נוירונים: נסמן את הקלט של הרשת באמצעות צמתים – q פיצרים ועוד חותך. הצומת המרכזי נקרא נוירון, ובו קשת מסמלת פרמטרים β , הנוירון מבצע מכפלת פנימית בין שני הוקטוריהם. זה עבר אקטיבציה (עם sigmoid) כדי לחת חיזוי סופי לתצפית \hat{p} . כל זה מכונה עד forward אחד.

backward(). נחשב את הנגזרות בצד loss(). לבסוף נלק צעד α בכיוון הגרדיינט ונמשיך.



שימוש בפייתון: ניעזר בספרייה tensorflow.keras של tensorflow. נאתח מודל sequential.dense, נוסף שכבה dense וקטיבית sigmoid (נבקש מהשכבה לא להשתמש ב-bias, החותך). נكمפל את המודל, שם נפרט את פונקציית ההפסד שהיא CE, ומספק את אובייקט SGD. לבסוף נריץ את המודל על הנתונים באמצעות fit.

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import SGD

model = Sequential()
model.add(Dense(1, input_shape=(X.shape[1], ), activation='sigmoid', use_bias=False))
sgd = SGD(learning_rate=0.1)
model.compile(loss='binary_crossentropy', optimizer=sgd)
model.fit(X, y, batch_size=100, epochs=50)
```

בשורט predict() קיבל את הוקטור $\hat{\beta} \approx (1,2)$ כמו שרצינו! קיבל את וקטור ההסתברויות החזיות באמצעות get_weights().sigmiod()

```
# See that it makes sense:
beta_hat = model.get_weights() # Note Keras gives a list of weights!
beta_hat

[array([[1.0786701],
       [2.1123998]], dtype=float32)]
```

```
pred = model.predict(X, verbose=0)
pred[:3]
```

```
array([[0.8194575 ],
       [0.22995582],
       [0.25202554]], dtype=float32)
```

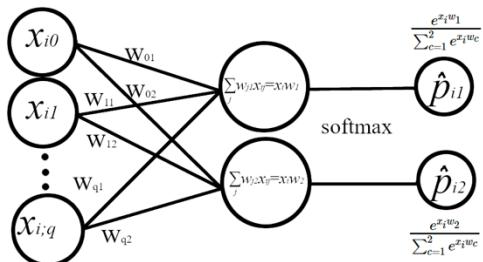
```
pred_manual = 1/(1+np.exp(-np.dot(X, beta_hat[0])))
pred_manual[:3]
```

```
array([[0.81945753],
       [0.22995581],
       [0.25202557]])
```

שכול הרשות

חווי יותר מאשר קלאסים: נכלול את הרשות ל- C קלאסים שהם 2 או יותר. נתאים וקטור מקדים לבל אחד מהקלאסים שלנו. נקרא לוקטור כזה בבר W – מטריצה של משקלות ממימד $C \times (q+1)$. נוסף בשכבה הפלט C נוירונים ל- C קלאסים, כאשר המכפלה הפנימית בכל נוירון כזה תעביר אקטיביציות softmax, שמייצרת C הסתברויות שיסתכמו ב-1.

$$\hat{p}_{i;c} = \text{softmax}(c, W_{(q+1) \times C}, x_i) = \frac{e^{x_i w_c}}{\sum_{c=1}^C e^{x_i w_c}}$$



הפונקציה לוקחת את הפלט של כל נוירון, מעלה אותו באקספוננט, ומחלקת בסך האקספוננטים בכל C הנוירונים. מודל זה ניתן לראות במודל שקול לרגרסיה מולטיאומית, שמכיל גרסיה לוגיסטית ל- C קלאסים. בעת נוסיף לדיאגרמה שני נוירונים עבור שני קלאסים, שמייצרים שתי הסתברויות שמסתכמות ב-1.

```
from tensorflow.keras.utils import to_categorical
y_categorical = to_categorical(y)
model = Sequential()
model.add(Dense(2, input_shape=(X.shape[1], ), activation='softmax', use_bias=False))
sgd = SGD(learning_rate=0.1)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
model.fit(X, y_categorical, batch_size=100, epochs=50)
```

בספרית keras נעשה מספר שינויים קטן, ה- y כבר לא יכול להיות וקטור, אלא מטריצה $C \times n$, בכל שורה יש 1 ב-label המתאים לתכפיות (המחלקה החזויה). האקטיבציה תהיה softmax ולא sigmoid, ופונקציית ההפסד תהיה binary categorical CE במקום

שוב נבדוק ידנית את תוצאות הרשות.

```
# See that it makes sense:
W = model.get_weights()
W
array([[-1.5730529 , -0.68782824],
       [-1.25236   ,  0.65500903]], dtype=float32)

pred = model.predict(X, verbose=0)
pred[:3]
array([[0.9868651 ,  0.01313489],
       [0.52004814,  0.47995186],
       [0.12084129,  0.8791588 ]], dtype=float32)

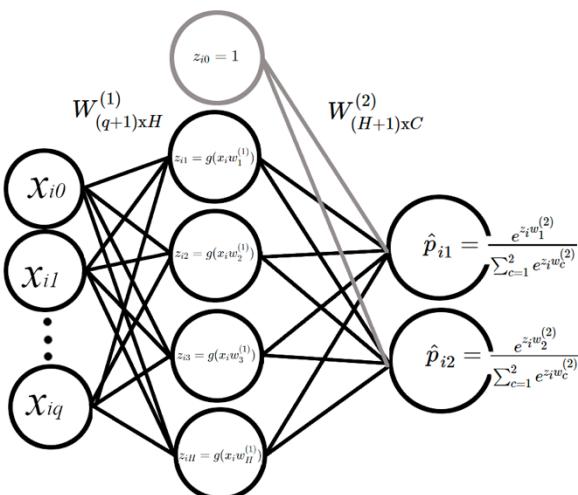
Z = np.dot(X, W[0])
Z_exp = np.exp(Z)
Z_exp_sum = Z_exp.sum(axis=1)[:, None]
pred_manual = Z_exp / Z_exp_sum
pred_manual[:3]
array([[0.98686511,  0.01313489],
       [0.52004815,  0.47995185],
       [0.12084127,  0.87915873]])
```

הוספה שכבות חבויות: הדיאגרמה נראית כבר כמו רשות נוירונים מורכבת. נוספים שכבות ביןים (hidden layer), בה הנוירונים מבצעים מכפלה פנימית ומתבצעת אקטיבציה לא לינארית g . כל קשת ברשת היא פרמטר, אלמנט במטריצת המשקלות W . בשכבה הסופית מתבצעת שוב מכפלה פנימית עם המשקלות W ולבסוף softmax לקבלת הסתברות הסופית.

כבר התרחקנו מרגרסיה לוגיסטית מרחק די רב. הרשות יכולה כבר למדל יחסים לא לינאריים מורכבים ביותר בין x ל-y. מבחינות keras כל מה שנוסף כאן הוא עד קריית-L(`model.add()`). המודל זהה נקרא MLP (perceptron).

Even now, the forward step is a simple formula:

$$\hat{p}_{n \times C} = \text{softmax}\{[1:\sigma([1:X]W^{(1)})]W^{(2)}\}$$



```
model = Sequential()
model.add(Dense(4, input_shape=(X.shape[1], ), activation='sigmoid'))
model.add(Dense(2, activation='softmax'))
sgd = SGD(learning_rate=0.1)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
```

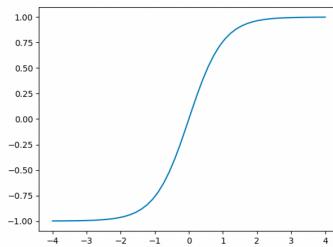
בפלט `summary` נראה את מספר הפרמטרים שהתקבל בכל שכבה. יש לנו 2 קליטים, אנחנו מוסיפים להם חותך, וכל זה כופלים ב-4 נוירונים בשכבה 1: $12 = 4 \cdot 3$. אחר כך יש 4 נוירונים עם חותך, והם מוכפלים ב-2 נוירונים בשכבה הסופית: $10 = 2 \cdot 5$. סה"כ 22 פרמטרים.

פונקציות אקטיבציה: ישנן פונקציה אקטיבציה נוספת לא לנאריות נपוצות יותר:

- \tanh – לוקחת קלט ממשי בלבדו, וודוחת אותו להיות בין $(-1, 1)$
- ReLU – היא פונקציית ציר, המאפשרת קלט שלילי, ומשאירה קלט חיובי כפי שהוא.

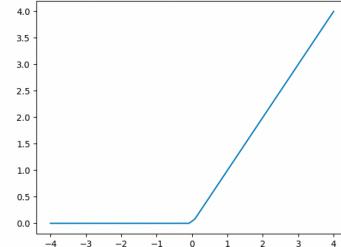
$$g(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

[▶ Code](#)



$$g(z) = \text{ReLU}(z) = \max(z, 0)$$

[▶ Code](#)



הגזרה שלhn קלה. הערה: אם משתמש בפונקציית אקטיבציה לנארית, נישאר במודל לנארוי.

```
# See that it makes sense:
pred = model.predict(X, verbose=0)
pred[:3]

array([[0.9364932 ,  0.0635068 ],
       [0.48637113,  0.5136288 ],
       [0.11851962,  0.88148046]], dtype=float32)

W1, b1, W2, b2 = model.get_weights()
W1.shape # (2, 4)
b1.shape # (4,)
W2.shape # (4, 2)
b2.shape # (2,)
W1 = np.vstack([b1, W1])
W2 = np.vstack([b2, W2])
W1.shape # (3, 4)
W2.shape # (5, 2)
# Get X ready with an intercept column
Xb = np.hstack((np.ones(n).reshape((n, 1)), X))
Xb.shape # (1000, 3)

(1000, 3)
```

```
Z = 1/(1 + np.exp(-np.dot(Xb, W1)))
Zb = np.hstack((np.ones(n).reshape((n, 1)), Z))
Z2_exp = np.exp(np.dot(Zb, W2))
Z2_exp_sum = Z2_exp.sum(axis=1)[:, None]
pred_manual = Z2_exp / Z2_exp_sum

pred_manual[:3]

array([[0.9364932 ,  0.0635068 ],
       [0.48637118,  0.51362882],
       [0.11851959,  0.88148041]])
```

בדוק את הפלט שוב ידנית:

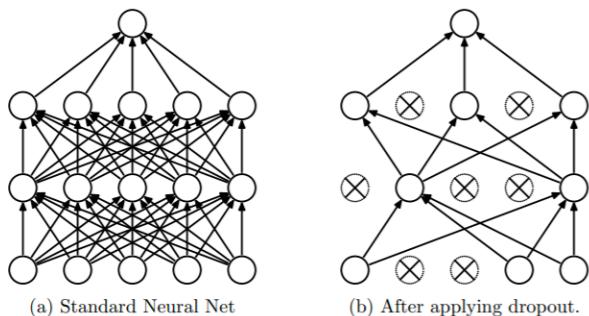
- $P_{L_2}(W) = \lambda \sum_{ijk} (W_{ij}^{(k)})^2$
- $P_{L_1}(W) = \lambda \sum_{ijk} |W_{ij}^{(k)}|$
- or both (a.k.a Elastic Net, but not quite):

$$P_{L1L2}(W) = \lambda_1 \sum_{ijk} (W_{ij}^{(k)})^2 + \lambda_2 \sum_{ijk} |W_{ij}^{(k)}|$$

גולםיזציה: רשתות נוירונים ידועות לשמירה ביכולת שלhn לבצע overfitting לננוירונים, ולכן נדרש גולםיזציה על מרחב הפרמטרים. אפשר להוסיף פונקציית עונש (penalty) שתתוסף לפונקציית הפסד. למשל עונש על מטריצת 2×2 של המקדמים. ככל ש- λ גדול יותר, העונש על פרמטרים גדולים מידי יהיה גדול מידי, והמודול יהיה "צטנע" יותר. אפשר להגדיר גם על נורמת 1 או על שילוב.

ב-keras אפשר להוסיף גולםיזציה על המשקלות בכל שכבה ושבבה.

```
model.add(Dense(4, input_shape=(X.shape[1], ), activation='relu',
                kernel_regularizer=regularizers.l1(0.01),
                bias_regularizer=regularizers.l2(0.01)))
model.add(Dense(2, activation='softmax',
                kernel_regularizer=regularizers.l1_l2(l1=0.01, l2=0.01)))
```



Dropout: רגולרייזציה יכולה להיות גם לא קשורה לשירות לפונקציית הփסיד, אלא אלגוריתמית – כמו `dropout` ב-`d.` אונחנו מכבים נוירונים בשכבה באופן אקראי בכל epoch. בסיסי מסיים ק אונחנו פשטוט מאפסים את הפלט שלהם. בזמן חיזוי על נתונים חדש, כל הנוירונים פעילים, והפלט שלהם מוכפל ב- k .

למה זהה יעבוד? `Dropout` הוא כמו אימון על אנסמבל של רשותות. בכל איטרציה המודל רואה רשות בארכיטקטורה אחרת, והצורה הסופית היא מעין ממוצע של הרובה רשותות כ אלה. היעילות מוסברת גם בעזרת מונחים מתוחם האנוליה הנומורית – בכל איטרציה שוגרים בויאן לעבר פרTRAN שכן בaczora אקראית, וכך גודל הסיכוי שנובסה את מרחב הפרמטרים ונגיע לנקודת אופטימום טוביה יותר.

עצירה מוקדמת (early stopping): במהלך האימון, נשמר בצד עוד סט קטן אקראי של נתונים, עליו המודל לא יתאמן, אלא רק ידוק האם אין ישירדה בפונקציית הփסיד על נתונים שלא ראה. אם אין ירידה במשך M צעדים, נפסיק את האימון כי אונחנו חושדים שהוא פשוט מבע overfitting למדגם הלמידה. ב-`keras` נמשצח זאת עם callback `validation_loss` ו-`validation`.
טור 5 צעדים אין שייפור בפונקציית הփסיד על תת-מדגם זה – הוא יעצור. בקריאה `l-fit` נחלק את מדגם הלמידה ל- 20% training ו- 80% validation

```
callbacks = [EarlyStopping(monitor='val_loss', patience=5)]
model.compile(loss='categorical_crossentropy', optimizer=sgd)

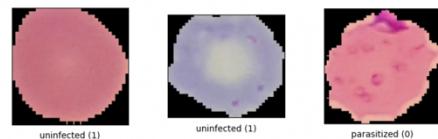
model.fit(X, y_categorical, batch_size=100, epochs=50,
           validation_split=0.2, callbacks=callbacks)
```

מיומש עם Keras: נרחיב על היכולות של Keras. הוא בעצמו API שטוח לתוכנות פופולריות ל-DL כמו TensorFlow ו-`PyTorch`.
היא מאפשרת להגדיר רשותות מורכבות מבלי להיכנס לבחרלים של תהליך האופטימיזציה, וגם רוצים יותר גמישות אפשר לבתו לעבור בספריות מאחוריו הקליעים. היא מתכתבת באופן ישיר עם numpy ו-`pandas`.

נסתכל על מסד נתונים מעניין של Malaria. במסד נתונים זה יש 27K תמונות של תאים שיכולים להיות נגעים במלריה, ונציג בעיתת סיוג של שני קלטים – האם יש מלליה כ/לא. הדאטה הוא מאוזן – חצי מהתאים נגעים וחצי לא. הוא נלקח ממאג' עצום של נתונים שנקרו tensorflow_dataset. בآن משתמש רק ב- 10% של הנתונים – 2500 תמונות, ונأخذ אותו לגודל של (100,100) פיקסלים (RGB).

```
import tensorflow_datasets as tfds
from skimage.transform import resize

malaria, info = tfds.load('malaria', split='train', with_info=True)
fig = tfds.show_examples(malaria, info)


```

נסתכל קצת על הדאטה – חלק מהתאים הם parasitized וחלק uninfected. אונחנו הופכים כל תמונה להיות מערך קטן, נחלק ל- 80% מדגם למידה ו- 20% מדגם טסט. 2000 תמונות לאימון, 500 לטסט, וביצע flatten לפיקסלים כך שכל תמונה עם 30,000.

נסתכל על התוצאות של מודל `LogisticRegression` של sklearn כפי שראינו בעבר. הדיק המתתקבל הוא 0.65 , די מזבב. נאמן בעת רשות عمוקה עם 3 שכבות ביןים. עם אקטיביציות ReLU, בשכבה הפלט sigmoid (הסתברות שהתא נגע במלריה).

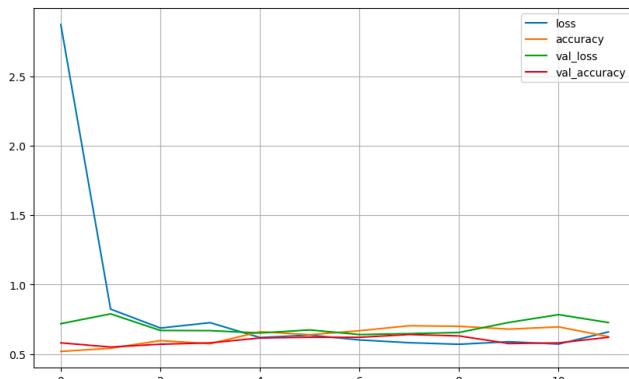
הפרמטרים במודל: יש לנו 30,000 פיקסלים ועוד חותך, שנכנסים אל 300 נוירונים בשכבה הראשונה: $M = 9$ פרמטרים.
לאחר מכך $100 \cdot 100 \cdot 100 + 1 = 30100$, בשכבה הבאה $5050 = 5 \cdot (100 + 1) = 51$. ולבסוף $1 \cdot 51 = 51$.

```
from tensorflow.keras.callbacks import EarlyStopping

callbacks = [EarlyStopping(monitor='val_loss', patience=5,
                           restore_best_weights=True)]

history = model.fit(X_train, y_train,
                      batch_size=100, epochs=50,
                      validation_split=0.1, callbacks=callbacks, verbose=0)
```

ניתן לראות בבר עכשו את המשקלות ההתחלתיות בראשת לפני שביבינו את האימון. **נקمل פל עם binary_crossentropy** ונטהמש באופטימיזציה מעט שונה שנקרה adam. במציאות metrics נבקש מהרשת לדוח לנו את accuracy לצד loss loss הרגל. נירץ את המודול עם early stopping על שמנט 10% מדגם validation.



לאחר הriceה, את history של המודל נטעף ב-DataFrame כדי לראות את האימון ולזהות אномליות. אפשר לראות שה-loss יורד על מוגם הלמידה ומוגם ה-validation ובאשר אין שיפור במשך 5 הצעדים האחרונים, מופסקת הלמידה. הדיק עליה עד שלב מסויים ובכל מקרה לא מראה מידי.

ניתן להריך על הטסט באמצעות evaluate, אחזק הדיק לא הרבה יותר טוב מרגסיה לוגיסטיבית (0.62).

בונן רשותה: מומלץ להשתמש בשיטות קיימות כדי לבצע tuning לפרמטרים, באמצעות ספריית scikeras למשל. נרצה לבנות את מספר השכבות, מספר נוירונים בהנחתה שהיא זהה בכל שכבה, ואת קצב הלמידה. בעת נתון לעטוף את האובייקט בклאס אחר לביצוע חישוב רנדומלי במרחב הפרמטרים שקדמו – RandomizedSearchCV. הוא מקבל מילון עם אפשרותים לכל פרמטרים. נגדיר איך אנחנו רצאים שיוציאו כל המודלים האלה, באופן הגדרנו 10 איטרציות שבהן נגידר פרמטרים מהמרחב שהגדכנו. נרים כל מודל על folds של DATA בפרוצדורה של cross-validation.

```
from tensorflow.keras.layers import InputLayer
from tensorflow.keras.optimizers import SGD
from scikeras.wrappers import KerasClassifier

def malaria_model(n_hidden, n_neurons, lrt):
    model = Sequential()
    model.add(InputLayer(input_shape=(30000, )))
    for layer in range(n_hidden):
        model.add(Dense(n_neurons, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss="binary_crossentropy",
                  optimizer=SGD(learning_rate=lrt),
                  metrics=["accuracy"])
    return model

keras_clf = KerasClassifier(model=malaria_model, n_hidden=1, n_neurons=30,
```

נשתמש בסט הנבחר בריצה על כל מוגם הנתונים, לפני חיזוי סופי על הטסט ט. גם עבורי, לא קיבלנו דיק טוב בהרבה מקודם. לעוד פתרונות מומלץ להסתבל על KerasTuner על מנת לסייע בבחירה。

```
from scipy.stats import reciprocal
from sklearn.model_selection import RandomizedSearchCV

params = {
    'n_hidden': [0, 1, 2, 3],
    'n_neurons': np.arange(1, 100),
    'lrt': reciprocal(3e-4, 3e-2)
}

rnd_search_cv = RandomizedSearchCV(keras_clf, params, cv=5,
                                    n_iter=10)
rnd_search_cv.fit(X_train, y_train, epochs=50,
                   validation_split=0.1, callbacks=callbacks)
```

מודלים של רשותות נוירונים יכולים לקחת זמן לא מעט לאימון, והם יכולים להיות די כבדים בסופו. מודלים פשוטים באים עם מתודה נוחה לשמירה על הדיסק ולהעלתו – save/load.



קוויז 12

שאלה 1: בכל שיטות הלמידה שלמנו דיברנו על כוונן היופרפורט או "כפתרו", שאפשר לשובב כדי להפוך את המודל למורכב וغمיש יותר או פחות. לדוגמה מספר השכבים K בNNKאו עמוק העז בעצם החלטה. איזה מה הבאים הוא לא היופרפורט או כפתר שכח, שאפשר להשתמש בו להפוך רשות ניירונים למורכבות וגמישה יותר או פחות?

- מספר epochs לחוכת לפני early stopping (פרמטר patience)

מספר שבבות

מספר ניירונים בכל שכבה

גודל lambda עונש מסוג L2 על המשקלות של הרשות

שאלה 2: כל השיעור עוסק ברשותות לאלטיפיקציה, כי הן מעת מורכבות יותר. מה עם גרסיה? נניח שאנחנו מנסים ללמידה י' רציף. אילו שינויים יש לעשות כדי שהרשות תהפוך לרשות למידול י'?

- הכל נשאר אותו דבר.

הכל נשאר אותו דבר, רק צריך לשנות את כל האקטיבציות linear, לומר לא אקטיבציה – מה עם פונקציית הפסד?

- הכל נשאר אותו דבר, רק צריך לשנות את כל האקטיבציות linear, לומר לא אקטיבציה, וכך לשנות פונקציית הפסד

להיות למשל MSE – איזו גרסיה תיווצר במקרה זה? וגרסיה לינארית פשוטה, הרשות יכולה מתנהגת באופן לינארי.

- השכבות הנוסתרות עם פונקציות אקטיבציה לא לינאריות – לא ניגע בהן – הן חינויות ביכולת של רשות ניירונים

ללמידה וליצג קשיים מורכבים בתוצאות.

- הכל נשאר אותו דבר, רק צריך לשנות את האקטיבציה בשכבה האחורונה linear, לומר לא אקטיבציה, וכך לשנות

פונקציית הפסד להיות למשל MSE

○ **אקטיבציה בשכבה האחורונה:** בגרסה, המטרה היא לחוץ ערכים רציפים ولكن בשכבה האחורונה אין צורך

באקטיבציה לא לינארית (כמו sigmoid או softmax) ש מגבילה את הערכים לטווח מסוים (למשל 0-1).

אקטיבציה לינארית פשוט מעבירה את הערך כפי שהוא.

-

גרסיה לוגיסטיבית, לעומת זאת, משתמשת בפונקציית אקטיבציה לוגיסטיבית (logistic) בשכבה האחורונה כדי לחוץ הסתברות, ולא ערך רציף.

-

פונקציית הפסד: במקומות פונקציית הפסד ש商量סת על קלאסיפיקציה כמו CE, בגרסה משתמשים בדרך כלל בפונקציית הפסד שמדוודת את הסטייה בין הערכים החזויים לאמתיים, כמו MSE.

שאלה 3: "רשת ניירונים" היא למעשה אלגברי, מודל פרמטרי. אנחנו משתמשים בסכימה כדי לתאר אותה כי ככל שנוסף שכבות ואקטיבציות וקשיים מעוניינים זה יעשה מרכיב יותר לתיאור מתמטי, אבל לו היו מתקשים הינו מצלחים. נכון, אפילו רשותות מודרניות כמו טרנספורמרים שעומדים בבסיס ChatGPT ניתנות לתיאור במודל פרמטרי, נסחה, זה פשוט מאוד מיאש.

שאלה 4: לבסוף קולע סדרה של קליעות עונשין. נניח שם יקלע (1) או לא (0), זו התפלגות ברנולי עם סיכוי k לקליעה. מתוך סדרה של 10 קליעות, זה וקטו התוצאות שהתקבלו: 1,1,1,0,1,1,0,1. מהי פונקציית הנראות של הפרמטר k?

- $p^8(1 - p)^2$ – הסיכוי לקליעה הוא p.
- $p^8(1 - p)^2$

• 0.8 – זהו האומדן נראות מקסימלית שיביא למקסימום את פונקציית הנראות המקסימלית, אבל זו לא פונקציית הנראות המקסימלית.

• $\frac{1}{8}(p^8(1 - p)^2)^{10}$ – תשובה זו תביא לאומדן נראות מקסימלית נכון אבל היא מעברת את ההתפלגות הבינומית שלא לצורך.



תרגול 9 – רשתות נירוניים

בשיעור שלנו עם מטריצת נתונים X רנדומלית, באן נתבוסס על הדאטה של פוקימון, וננסה לחזות את ערך ה-binary-y (כן/לא).
binary_crossentropy. ברגע אנחנו עושים סיווג ביןארי, ולכן נשתמש ב-.keras

```
X.head()

   Attack Defense Sp_Atk Sp_Def Speed Legendary
0      49       49     65     65     45    False
1      62       63     80     80     60    False

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import SGD
# instantiate a NN through Sequential API (there are others)
model = Sequential()
# add a single Dense layer with 1 neuron and sigmoid activation
model.add(Dense(1, input_shape=(X.shape[1], ), activation='sigmoid'))
# define optimizer
sgd = SGD(learning_rate=0.001)
# compile model, specify loss, optimizer, metrics
model.compile(loss='binary_crossentropy', optimizer=sgd)
# train the model, specify no. of epochs, batch-size, callbacks..
history = model.fit(X, y, batch_size=10, epochs=10)
```

כדי לבצע רgression ורילה (Regressor) – הפלט הוא ע"מומי ולא ההסתברות לקלאס בין 0 ל-1) נדרש לבצע שני שינויים:

- פונקציית הփסד תהיה MSE.
- נסיר את פונקציית האקטיבציה של sigmoid (הדיפולט הוא linear).

כדי להכפיל למספר קלאסים נרחיב את במודות הנוירונים בשכבות הביניים. ואז נקבל בקצת מספר בין 0 ל-1 בפלט הראשון \hat{y}_1 , ובפלט השני בדיק $\hat{y}_1 - 1 - \hat{y}_2 = 1 - \hat{y}_2$. את הוקטור צריך להפוך למטריצה $2 \times n$, ואת זה אפשר להכפיל לbumות קלאסים כללית $k \times n$. נשנה גם את האקטיבציה ל-softmax ואת הפסד_categorical-softmax.

```
from tensorflow.keras.utils import to_categorical
y_categorical = to_categorical(y)
model = Sequential()
model.add(Dense(2, input_shape=(X.shape[1], ), activation='softmax'))
sgd = SGD(learning_rate=0.001)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
history = model.fit(X, y_categorical, batch_size=10, epochs=10)
```

bumות הפרמטרים היא $(6 + 1) \cdot 2 = 14$ (6 + 1) · 2 = 14 נוירונים.

```
model.summary()

Model: "sequential_6"
-----
```

Layer (type)	Output Shape	Param #
dense_11 (Dense)	(None, 2)	14

```
Total params: 16 (68.00 B)
Trainable params: 14 (56.00 B)
Non-trainable params: 2 (0.00 B)
Optimizer params: 2 (12.00 B)
```



אנחנו רוצים מודלים לא לינאריים, למdal קשיים יותר מורכבים. וכך נוסיף עוד שכבות חבויות! הסוד הוא שפונקציית האקטיבציה
בשכבות החבויות אינה לינארית (כמו sigmoid, ReLU וכו').
אם האקטיבציה תהיה $Z = \sigma(z)$ היזהות? נקבל וגרסיה פשוטה! מתקבלים מודל לינארי, אין יותר קשרים לא לינאריים.

```
model = Sequential()
model.add(Dense(4, input_shape=(X.shape[1], ), activation='sigmoid'))
model.add(Dense(2, activation='softmax'))
sgd = SGD(learning_rate=0.001)
model.compile(loss='categorical_crossentropy', optimizer=sgd)
history = model.fit(X, y_categorical, batch_size=10, epochs=10)

model.summary()

Model: "sequential_7"
+-----+-----+-----+
| Layer (type) | Output Shape | Param # |
+-----+-----+-----+
| dense_12 (Dense) | (None, 4) | 28 |
| dense_13 (Dense) | (None, 2) | 10 |
+-----+-----+-----+
Total params: 40 (164.00 B)
Trainable params: 38 (152.00 B)
Non-trainable params: 0 (0.00 B)
Optimizer params: 2 (12.00 B)
```

לסיקום נראה את הספריה בפעולה על נתונים של בלוגים:

- 50,000 בלוגים עם 280 פיצ'רים (מספר המילים בבלוג, מתי העלו את הבלוג...).
- ה-ע שלנו הוא מספר התגובהות ב-24 שעות הבאות.
- נתחילה מ-`LinearRegression` קלاسي של `sklearn` והגענו ל-MSE של 0.70.
- עבשוו בניית רשת נירונית כולל `EarlyStopping`, `Dropout`, והמודל יצא מכוון לחלווטין כי צריך לנורמל את הדאטה. הגיענו ל-MSE של 1. לאחר הנורמל, הגיעobar לשגיאה של 0.54.
- **נשווה את זה ל-Gradient Boosting ושם נקבל 0.45!** מדובר בדאטה טבלאי, ונראה שיש **כאן הצלחה טובה יותר מרשת הנוירונים**. אחרי עוד tuning נקבל אפילו 0.42 (ນציגן כי לא ביצענו כאן cross validation אז זה לא הכי טוב).



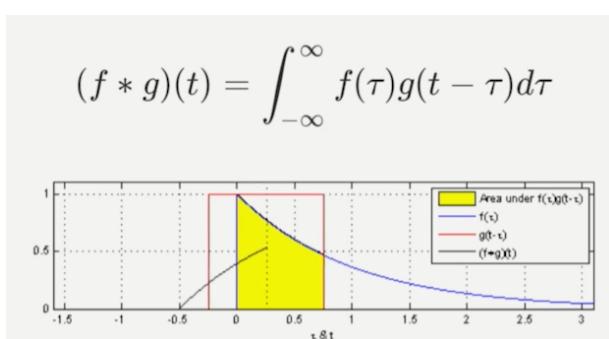
רשתות קונבולוציה

נלמד על רשתות קונבולוציה שמתאימות במיוחד לבניית מודלים לחיזוי על תמונות, והן בnejrah המודול המודרני ביותר בקורס זה. הן הבסיס להבנה של מודלים מתחום הראייה הממוחשבת, שמאפשרים למוכנות לנסוע ללא נהג, וללקוחות לצאת מסופר מוביל לעבר בKOפה.

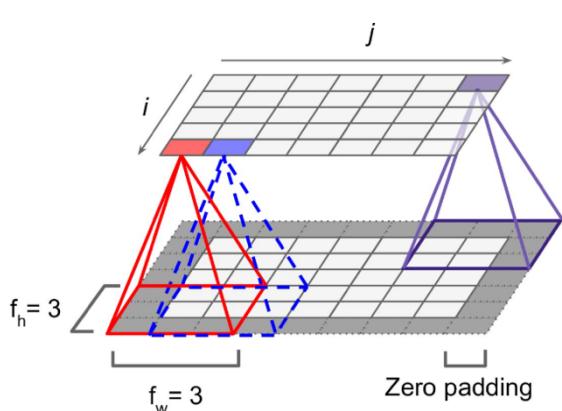
רשת Konvoluzija

רשות Konvoluzija לא צמחו באקום. נזכר בייצוג של תמונה במחשב – למשל מערכ (1400,1400,3), כאשר ישן 3 שכבות צבע (RGB). אם ניקח את השכבה הראשונה ונבקש ולראות ריבוע קטן במרכז התמונה, נראה פשוט מספרים. התמונה היא מערך תלת מימדי של מספרים שלמים (בין 0 ל-255, 256 אפשרויות לכל פיקסל בכל שכבה צבע).סה"כ כממעט 6MB. אם נתיחס לכל פיצ'ר/משתנה, נקבל 6 מיליון!

רשתות Konvoluzija החלו בערך שנות 98, בשחצגה LeNet-5 ליזוי אוטומטי של ספרות. ב-2012 פורסמה הרשות AlexNet שהביאה ליכולות חיזוי שטרם נראו על מאגר תמונות שנקרה ImageNet. עד אז, התיחסו לתמונה בעוד עם מימד מאד גבוה, והתמקדו ברענוןות שונות לפיצ'רים על סמך הבנת הפיזיקה של העצם הזה שנקרה תמונה – למשל היסטוגרמה של במות האדום בתמונה ומתארים את הממוצע, החיצון וה-skewness שלו, ועל זה מפעלים את המודלים שלמדו לנו boosting למשל.



Konvoluzija: Konvoluzija היא מultiplication של שתי פונקציות, ולקיחת אינטגרל או סיכון של ערך המכפלה על פני טווח מסוים. אנחנו מזדים את g במו חלון נב, ובכל נקודה בודקים מהו סך שטח החיפוי בין f (אינטגרל) ומבטאים את השטח באמצעות פונקציה חדשה. במקרים מסוימים, הפונקציה החדשה מאוד מזכירה קורלציה בין הפונקציות, פונקציית מתאם שבודקת עד כמה f דומה ל- g . אם f הייתה זהה ל- g ב点多 נקודה, הן היו חופפות לחולון והאינטגרל היה 1 (שטח הריבוע). כאן זה לא קורה. ראים בבר Konvoluzija כבדיברנו על KDE (kernel density estimation) ואיך יוצרים תרשימים צפיפות של התפלגות.agem קראנו לחולון הנע זהה (x) w .



שכבה Konvoluzija: השכבה הראשונה מורכבת מ- $k \times k$ נירונים, שלכל אחד יש שדה רצפטיבי (receptive). הוא לא מסתכל על כל הפיקסלים בתמונה שמתוחתי, רק על ריבוע של 2×2 או 3×3 . בשכבה הבאה, כל נירון מסתכל בשכבה שמתוחתי שאפשר לראות בה תמונה חדשה (גם על שדה רצפטיבי). יותר במדוק, אפשר לחשב על פרמטרים s_w, f_h שמנדרים את גודל השדה זהה. נירון יסתכל על המלון שנוצר. אפשר לראות את הריבוע שהנירון האדום מסתכל עליי בשכבה שמתוחתי. נצעד צעד ימינה, ונראה את הריבוע של הנירון הכלול. באופן זהה, נסרקת כל השכבה במעין חלון נב שמתפרק כל פעם על אחר. אם לא יהיה שום שינוי, שכבת הנירונים מעלה תמונה בבר לא תוכל להישאר באותו גודל. בڪנות, התמונה הבאה תהיה חסירה פיקסל אחד מכל צד. כדי זהה לא יקרה, נהוג לעשות גם zero padding כדי לעבות את התמונה.

כל הנוירונים בשכבה, לומדים filter/kernel/feature באמצעות הרצפטיב שלהם. אם הוא 3×3 הם ילמדו פילטר W בגודל זהה. למשל, عمודה של אחותות באמצעות אפסים משני צדדי. ה- W הוא בעצם משקלות הרשת, והוא סט הפרמטרים שהוא לומדת וمعدכנת באמצעות GD לאור פונקציית loss כלשהי. הנוירונים בשכבה Konvoluzija יכפלו כל batch שהם מסתכלים עליו עם הפילטר הזה ויסכמו במספר אחד – **כלומר יבצעו Konvoluzija!**

```
X = np.array(
[
    [0, 1, 0, 1, 0],
    [0, 1, 0, 1, 0],
    [0, 0, 0, 0, 0],
    [1, 0, 0, 0, 1],
    [0, 1, 1, 1, 0]
])
```

```
W = np.array(
[
    [0, 1, 0],
    [0, 1, 0],
    [0, 1, 0]
])
```

```
Z = np.array(
[
    [0, 2, 0, 2, 0],
    [0, 2, 0, 2, 0],
    [1, 1, 0, 1, 1],
    [1, 1, 1, 1, 1],
    [1, 1, 1, 1, 1]
])
```

כאן יש לנו תמונה בגודל 5×5 , מעין סמייל. בשכבה הבא Z כל נירון יהיה שווה להכפלה וסיכון של ה- W על 3 שהוא מסתכל עליו עם הפילטר W . חוץ מהפילטר שככל הנירונים בשכבה משתמשים בו, נלמד גם חותך (bias). נניח שבכצעו לתמונה zero padding. איך תראה השכבה Z ? למשל הנירון בקצה השמאלי העליון, הוא המכפלה של W ב- $batch$ השמאלי העליון ונקבל 0. בכיה הפילטר שלנו סורק את כל התמונה ומאחסן את המסקנה שלו בשכבה החדשה.

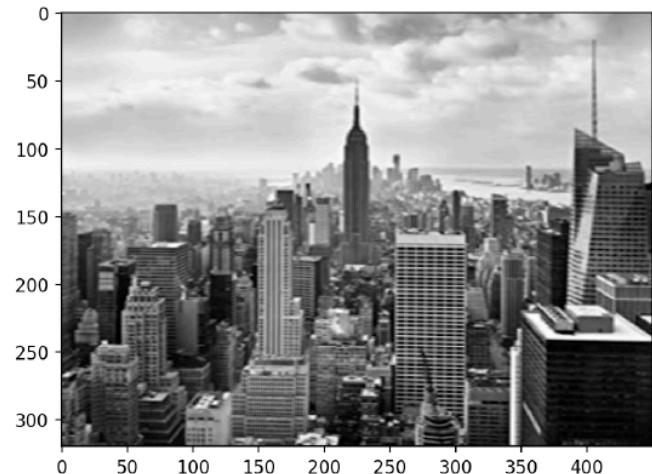
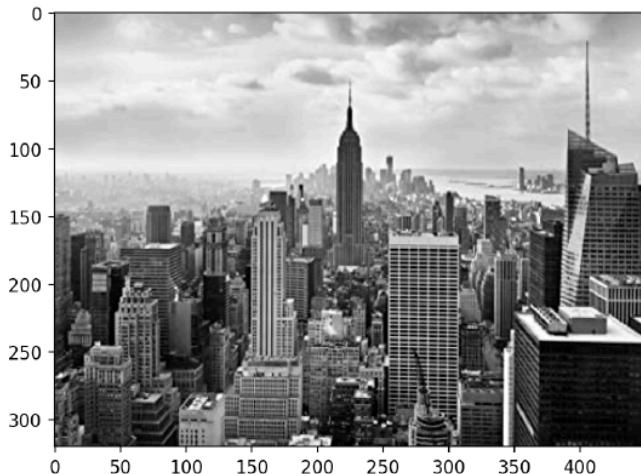
$$Z_{i,j} = b + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} X_{i+u,j+v} W_{u,v}$$

אם נסתכל על Z , מתי היא היכי חיובית? מה היכי "הדלק" את הנירון? הערכים היכי גבוהים הן באזורי "העינום" של השמאלי. הפילטר שלנו שיש בו בעצם קו אנכי באמצע, מגיב היכי הרבה לקוים אנכיים. הוא סורק את התמונה ומחזיר ערכים גבוהים איפה שהוא מוצא מותאם עם הפרט המאוד עדין זהה.

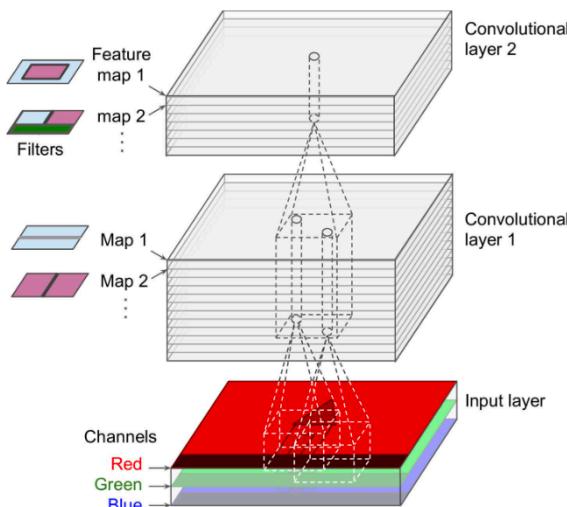
ניתן בפונקציה (`d2vconv`) בין תמונה ניו יורק ל-W שלנו. הפילטר שלנו מדגיש את הקווים האופקיים של הבניין, בעוד שהוא מושטש את הקווים האנכיים.

```
plt.imshow(
    ny,
    cmap = 'gray')
plt.show()
```

```
plt.imshow(
    ny_convolved[0, :, :, 0],
    cmap = 'gray')
plt.show()
```



בנייה רשת קובולוציה: אחת האפשרויות שעומדות ברשותנו בכל שכבת קובולוציה, היא לשנות את פרמטר `strides`. לעתים זה מאד מועליל לעשות קובולוציה על תמונה לא צעד-צעד או פיקסל-פיקסל, אלא לבקש `step size` של 2, ואז השבבה שמעל תיעשה קתנה יותר.



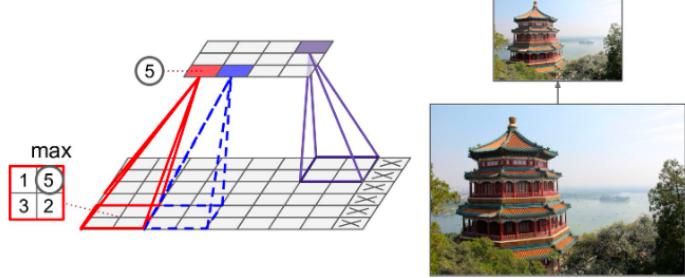
תמונה סטנדרטית מורכבת מ-3 שכבות – ושבבת קובולוציה מסתכלת על כל 3 השכבות של התמונה – RGB. היא לא לומדת פילטר של 3×3 , אלא קוביה $3 \times 3 \times 3$ פרמטרים שונים (לא כולל bias).

זה מכונה **feature map stack** אחד, והשכבה היא **kernel** של כמה **feature maps** באליה, שככל אחד לומדת את **הkernel** שלו שימושות כל הנירונים**-by-map**. ניתן לראות איך הראשון לומד קווים אופקיים על רקע הכלול. השני לומד להציג דזוקא לקווים אנכיים על רקע אדום. השכבה הבאה תסתכל על השכבה הקודמת, והיא לא תלמוד קוביה אלא מעין תיבת ארכובה בזאת – נירונים שמייצגים פיצ'רים מאוד low-level כמו קו אופקי או אנכי, ומשלבת ביניהם. כאן למשל הראשון יgive לריבוע אדום על רקע הכלול. ככל שנלך במULA הרשת אנחנו לומדים פיצ'רים יותר ויותר מורכבים. זה מאפשר ללמידה עין או אוזן שלقلب וכו'. גם בשחשוףנו מימד מדויב בפעולה פשוטה, של הכפלת וסכום. לקרnell שלו נוסף מימד f_n ואנחנו סובמים גם עליו.

למידה ברשת: יש לנו פונקציית הפסד כלשהו, והרשת מבצעת forward/backward כדי ללמדנו עם SGD על הפרמטרים של الكرנלים השונים, שוב ושוב כדי להוריד את loss.

אחסון: יש סיבה שהרשת הזאת לא הייתה בשימוש עד לפני 15-10 שנה. עברו תמונה אחת בגודל סביר של (100,100,3), נבנה בשכבה הראשונה 100 feature maps עם פילטר (3,3,3) של קובייה. קיבל 2800 = $100 \times 3 \times 3 + 1$ (3 פרמטרים עבור שכבה אחת. אבל, בכל feature map צריך לאחסן בשכבה אחת 100 תמונות שונות. כל תמונה היא (100,100,3) אז בשכיב כל התמונות נצטרך לאחסן $100^3 = 1000^3 = 1000000$ מספרים, שככל אחד נניח שוקל 4B, אז מקבל $4MB$ לתמונה אחת בשכבה אחת. כל מסטר הוא סכום ממושקל של 27 מספרים, כלומר $27 \cdot 4 = 108MB$.

בלי מימוש חכם שנוגע לחישוב ואחסון רשותות אליו נשארו בגדיר רעיון בלבד, או נחלתם הבלתי עדין של חוקרים עם מחשבי על.



Pooling: כדי להקל חישובית ולעוזר לרשת ללמידה טוב יותר ניעזר ב- **pooling layer**. שכבה זו לוקחת מקסימום או ממוצע של שדה רצפטיבי כלשהו. באנו, הכוiron האדם לוקח רק את המקסימום, אין פרמטרים. היא לא משתתפת במשחק GD ולא מעיקה חישובית.

אם נחזיר לסמייל שלנו, נעשה max pooling max וגיעו לתמונה קטנה הרבה יותר. שכבה Z2 מבצעת מקסימום, ואם לא ביצנו zero padding לא נcosa את קצוות התמונה. אנחנו מאבדים מידע! בשכבה max pooling max בגודל 2×2 עם stride 2, נקטין את השכבה הקודמת ב-75%.

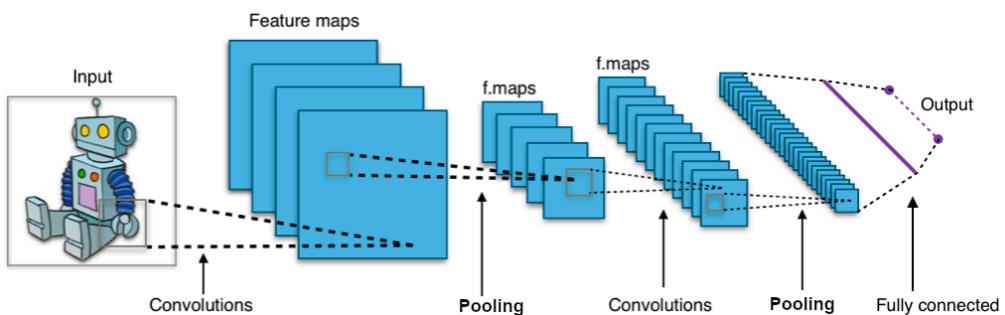
Usually with strides $s_w = s_h = f_w = f_h$ and no padding (a.k.a **VALID**):

<code>X = np.array([[0, 1, 0, 1, 0], [0, 1, 0, 1, 0], [0, 0, 0, 0, 0], [1, 0, 0, 0, 1], [0, 1, 1, 1, 0]])</code>	<code>W = np.array([[0, 0, 0], [0, 0, 0], [1, 1, 1]])</code>	<code>Z = np.array([[1, 1, 2, 1, 1], [0, 0, 0, 0, 0], [1, 1, 0, 1, 1], [1, 2, 3, 2, 1], [0, 0, 0, 0, 0]])</code>	<code>Z2 = np.array([[1, 2], [2, 3]])</code>
---	---	---	---

אולי זה טוב להתעלם מכמה נוירונים? אנחנו יודעים כי מיצוע עוזר לחיזוי ומקטין overfitting. **לקיחת מקסימום מאפשרת לדעת האם פיצ'ר מסיים** קיים או לא בשכבה מתחת, בלי להתחשב ולהיות רגשים למקומם של הפיצ'ר הזה, מעד להזות בכל מקום. זהוי תכמה של אינטואיטיביות.

האם שכבת הקונבולוציה פועלה **ליינארית**? היא יכולה להיות – הכפלת, סיכום, ועוד pooling ע"י ממוצע. עם פרמטריזציה נכונה אפשר לנசח את כל הפעולות האלה כפועלה ליינארית. בפועל, כמעט תמיד נוסיף אקטיבציה לא ליינארית (למשל ReLU). **ニישאר עם פועלה לא ליינארית.**

רשת טיפוסית: תמונה כניסה ונכנסת וועברת שכבת קונבולוציה ראשונה עם מספר feature-maps, כל אחת לומדת פיצ'ר low-level אחר. לאחר מכן נבצע pooling (מקסימום). עוד שכבת קונבולוציה ועוד pooling. בסופו של דבר נעשה שיטוח לנוירונים שנשארו אותם, ונחבר אותם בשכבה Dense. **נוירון בודד עם פונקציית אקטיבציה sigmoid בשביל סיגוג, או ליינארית בשביל רגסיה.**



Augmentation: כמשמעותה בנתוני טבלה (אנשים, מוצרים), קשה לעבות את הטבלה בעוד נתונים, ולעשות לה תכונות זה דבר שונה, זה לא משנה אם כלב נמצא בצד ימין או שמאל בשמהטרה היא להזות בלבד. יש מספר טרנספורמציות שאפשר לעשות על תמונות, ובכך להגדיל את גודל המדגם ולגון אותו באופן שמאפשר לרשת קונבולוציה להחליט טוב יותר: העתקה (להזיז את התמונה ממוקם למוקם), סיבוב, הגדלה והקטנה של המימדים, היפוך שלה, מתייחה שלה (גם בכיוונים אלכסוניים).

צריך להיזהר לא תמיד הינו רוצה לבצע augmentation image. למשל ביישומים רפואיים, כמו רשותות שיקראו צילומי רנטגן. לא הינו רוצה גידול ממוקם למקום, וגם הרשות תבחן רק על צילומי רנטגן, מיטשרים, ממכונת רנטגן. אפשרות אחרות שצריך להיזהר אליה בדוגמה הצורום – למתוח אותו? אולי זה מעוות אותו שלא לצורך ומידק לרשות. **ImageDataGenerator** ישימוש נוח לעשות את זה בצורה אוטומטית **on-the-fly** תוך כדי אימון, באמצעות



נסכם:

- אנחנו לא מתייחסים לפיקסלים כב"ת, לוקחים בחשבון את התלות המרחכית ביניהם.
- הארכיטקטורה של הרשת מאפשרת לשכבות הראשונות ללמידה **פיצ'רים מואוד בסיסיים** כמו קו אנכי, ולאט לאט לבנות מהם **פיצ'רים מורכבים יותר ווותר**.
- **feature weights** – בـ shared weights לומדת סט של משקלות אחד, כל הנוירונים חולקים אותו. זה לא הרבה פחות פרמטרים מרשת רגילה, זה גם אפשר ללמוד פיצ'ר בכל מקום בתמונה.
- **Pooling** מוגע מאיינו **לעשות overfitting** לכל פיקסל ופיקסל, ומאפשר אינוריאנטיות במידה עם מקסימום.
- **Augmentation** מאפשר **הגדיל את מסד הנתונים שלו**, ולרשת להכליל מול קצת רעש, איפה שזה דבר נכון ביסודו, מוגדים חזקים יותר כמו GPU, TensorFlow ו PyTorch. מהווים חלק חשוב במעבר של CNN לבסוף, מהאקדמיה לתעשייה, מה שמאפשר להם ללמידה ולהזות בצורה מהירה כל כך.

שימוש ב-CNN

הגדלת הרשת: נראה את הביצועים של CNN בפועל על תאי malaria. בשחצנו אותם לרשת רגילה, שיטחנו את התמונות ל-30,000 פיקסלים. במקורה החדש עם CNN, נישאר עם תמונות תלת-מימדיות. בניית שכבת קונבולוציה עם 32 פילטרים (-maps), כל קרנל יהיה $(3,3)$, ונגידו zero padding. על הפלט של כל נוירון תהיה אקטיביצית ReLU. הערכה: נשים לב כי העומק של הקרナル (המימד הנוסף) הוא אוטומטי משתווה לעומק של התמונה עצמה (המימד 3). נוסיף עוד שכבות באלו, נבצע flatten ובקצת sigmoid, המשימה של הרשת היא סיווג – אנחנו רוצים score בין 0 ל-1 שיבטא עד כמה הרשת בטוחה שהטהה נגוע במלריה.

model.summary()		
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 100, 100, 32)	896
max_pooling2d (MaxPooling2D)	(None, 50, 50, 32)	0
conv2d_1 (Conv2D)	(None, 48, 48, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 64)	0
flatten (Flatten)	(None, 36864)	0
dropout (Dropout)	(None, 36864)	0
dense (Dense)	(None, 1)	36865
<hr/>		
Total params: 56,257		

$$\begin{aligned} 32(3 \times 3 \times 3 + 1) &= 896 \\ 64(3 \times 3 \times 32 + 1) &= 18496 \\ 36864 + 1 &= 36865 \\ \hline 56K & \end{aligned}$$

כמות הפרמטרים: נשים לב בבר שברשת השיטה ביחידה הקודמת הגענו למעלת 9M פרמטרים. כאן יש רק 56K, שמייד גודל פחות. בשכבה הראשונה יש 32 FM, כל אחת למדת $(3, 3)$ שכבות צבע, ועוד: $32(3^3 + 1) = 896$. בשכבה הבאה של הקונבולוציה: $32 \cdot 32 + 1 = 18496$. לא ביקשנו zero padding ולכן איבדנו שני פיקסלים אופקיים ואנכיים והגענו מ-50 ל-48 pooling מה-convolution השנייה.

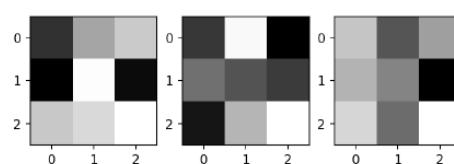
בעת נאמן את הרשת עם EarlyStopping. עם רשת פשוטה אחרי הרבה tuning הגיעו ל-69% בפעם הקודמת. יזכיר שהרשת מוציאה פלט בין 0 ל-1, כדי לקבל את החיזוי הסופי נוצרה להשוות ל-0.5. בבדיקה דיקן קיבל 76%. שט_mbוקשים גם את confusion_matrix וראים שהוא אכן ברורה ורוב התוצאות נמצאות על האלבון. זה רק עם 10% מההתמונות. נעשה אימון על כל 27K התמונות והגיעו ל-94%!

יזואלייזציה: כדי לגרום ללקחות להשתמש במקרים CNN, ולהסתמך על התוצאות שלהם, יש לוודא כי ניתן להסביר מה המודול שלנו עשו (explainability). ננסה לעשות ויזואלייזציה של מה שהמודול שלנו ראה. בבחינת פרמטרים, אנחנו יודעים להשיג את המשקלות בכל שכבה. אנחנו יכולים ממש לציר את הפילטר הראשון עברו כל שכבה צבע. השכבה הראשונה באופן כללי לומדת **פיצ'רים מואודים** low-level.

```
W, b = model.get_layer('conv2d').get_weights()
W.shape
(3, 3, 3, 32)
W[:, :, 0, 0]
array([[-0.05327014,  0.05076471,  0.0883847 ],
       [-0.0933094 ,  0.14134812, -0.08010183],
       [ 0.08545554,  0.10305361,  0.14380825]], dtype=float32)
```

First filter of 32:

```
plt.subplot(1, 3, 1)
plt.imshow(W[:, :, 0, 0], cmap='gray')
plt.subplot(1, 3, 2)
plt.imshow(W[:, :, 1, 0], cmap='gray')
plt.subplot(1, 3, 3)
plt.imshow(W[:, :, 2, 0], cmap='gray')
plt.show()
```





```
from tensorflow.keras import Model
model_1layer = Model(inputs=model.inputs,
                      outputs=model.layers[0].output)
feature_maps = model_1layer.predict(cell)
feature_maps.shape
(1, 100, 100, 32)
```

בדי בכל זאת להבין, אפשר לנקוט תמורה נתונה, התא הראשון במודם הטעט, ולהרייך אותה דרך השכבה הראשונה של הקונבולוציה. נראה מה קרה לתא שבער דרך 32 FM שונים.

נגידור לשם כך מודל עוזר, שהקלט יהיה כמו הקלט של המודם המקורי, והפלט יהיה הפלט של השכבה הראשונה. בשנדין למודל זהה את התא הספציפי שלנו נקבע 32 FM בגודל (100,100), ואפשר להראות אותו כדי להבין מה הדליק כל FM בשכבה הקונבולוציה הראשונה. ככל שהתמונה בהירה יותר, האזרע חשוב יותר ל-FM. ניתן לראות FM שמאוד מתמקדים בגבולים של התא – הם משאים הכוול חוץ מגבולים. בצורה זהירה ננסה להבין יותר ויתר אין הרשות שלנו מזהה תא שנגוע במלליה. הרבה פעמים נראה כמה יצירתיות הרשות עצמה למדה.

ארכיטקטורה: יש ארכיטקטורות שונות על בסיס הרעיון שראינו. בתחום ImageNet אוחז הטיעות בסיווג תמונות לקטגוריות באמצעות 5-top קטגוריות שהמודל מציע מיותר 1000, יורד בכל שנה. קפיצה מרשימה הייתה ב-2012, וב-2015 (עם ארכיטקטורת ResNet-5). כוים מוגעים מתחת ל-1% ב-5-top).

הארכיטקטורה של LeNet-5 ב-1998 פשוטה וניתנת לימוש ב-keras. LeNet כולל לפחות ענפים של קונבולוציה אשר "mdbrim" אחד עם השני עד לחיזי סופי של 1000 נוירונים. ResNet בניית מבוקים שבכל אחד 2 שכבות קונבולוציה. ככל שהמודם יהיה עמוק יותר, כך הוא יהיה מורכב יותר ומסוגל ללמידה דפוסים מורכבים יותר ולהגיע לביצועים טובים יותר.

חוקרים תמיד נזהרו לא לבנות מודלים עמוקים מדי בגלל בעיית vanishing gradient אקטיבציה כמו ReLU מפעסת כל קלט מתחת ל-0, כי זה מה שמוסיף אי-ילינאריות לרשות שלנו. אחרי הרבה שכבות, מה הסיכוי של סיגנל לעפעף מהתמונה עד לказחה הרשת? כך בעצם "געם" הגרדיאנט ולא מתבצעת למידה. באמצעות פרדיגמה של residual learning אנחנו מסיעים לסיגナル להתקדם, יוצרים מעקף ביליאגעת בו וסוכמים אותו רגע לפני אקטיביצית ReLU סופית. הבלוק לא לומד פונקציה של הסיגナル (X) H אלא את השארית $X - H(X)$.

שימוש במודלים מאומנים: הרשות שנלמדה היא אוסף של משקלות. ניתן להוריד אותן מהאלינטראנט עבור מודלים מפורטים, ולהשתמש בהם. נשתמש ב-ResNet50, נtakes את הגודל של התמונה שאנו חיצים ל- S (תמונה של הכלב יון), לגודל שהרשות מצפה לקבל. בצע `predict()` וכן בミニום באמצעות מודל קיים כדי לחזות תמונה מהאוסף האישי שלנו.

QUIZ 13

שאלה 1: אורק/רחב השדה הרצפטיבי של הקונבולוציה (פרמטרים w_h, f_h בשיעור) עשוי לשמש כדוגמא ל:

- גולריזציה ברשותות: ככל שהשדה הרצפטיבי קטן יותר ביחס לפוטרים קענים יותר והמודם יהיה מורכב וגמיש יותר, או יותר מדי.

- שיטוף פרמטרים ברשותות: ככל שהשדה הרצפטיבי קטן יותר פחות פרמטרים משותפים ומספרם הכללי ברשות גדול – אפיו אם השדה הרצפטיבי היה בגודל 1 על 1 הפרמטר זהה היה משווה לכל השכבה.
- אובייפטיניג למדגם הלמידה: ככל שהשדה הרצפטיבי קטן יותר נתקשה להכפיל למודם הטעט לתמונות חדשות.
- אי-ילינאריות ברשותות: ככל שהשדה הרצפטיבי קטן יותר במודל "מתרחך" מלינאריות – גם אם השדה הרצפטיבי היה בגודל כל התמונה המודם יכול היה להיות לא-ילינארי.

שאלה 2: לפניך שכבת קונבולוציה סטנדרטית ראשונה (כולומר מתבצעת ישירות על תמונה וגילתה).

```
Conv2D(filters=20, input_shape=(100, 100, 3), kernel_size=(3, 3), padding='valid', activation='relu',
use_bias=False)
```

כמה פרמטרים יש לשכבה זו? 540 : 20 = 540 · 3³. אין חותך אז לא מוסיפים 1.

שאלה 3: בשיעור השתמשנו ברשות ResNet שבער אומנה על 1000 הקלאסים של imangenet וחידנו באמצעותה את הגזע של כל בתמונה שלא ראתה. אם ננסה לחזות באמצעות הרשות קלאס שהוא לא מכירה (ראו באן), למשל תמונה שמצויה בבירור מיקרוסkop, תוחזר טעות או תשובה "לא ידוע". לא נכון, עדין נקבל הסתברות לכל אחד מ-1000 הקלאסים הקיימים וזה אכן בעיה.

שאלה 4: איזה מה הבאים הוא התיאור הטוב ביותר של מהו קונבולוציה?

- **שילוב שני פונקציות יצירתיות פונקציה לשישית שימושית מתאימים במסים בין שתי הפונקציות**
- פועליה מתמטית שמחשבת נגזרת של פונקציה
- טכניקה להבחין בסיגナル מסוים בתמונה ולהעיצם אותו
- שיטה להווידת המידע של תמונה באמצעות הטעות ממידע לא נכון

תרגול 10 – רשותת קונבולוציה

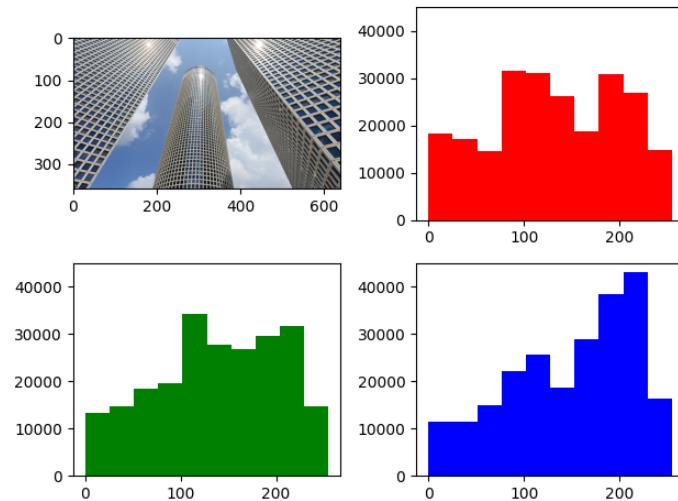
תמונה 10.1: נתחל מהוסף, ונראה איך רשותת קונבולוציה מבצעות על הדאטא של `wikiart`. השתמש בספרייה `keras-tuner`. נעשה סטנדרטיזציה לתמונות באמצעות `ImageDataGenerator`. נגידר גנרטור באמצעות `flow_from_directory` על תיקיות התמונות שלנו מחלוקת לפי `.train/test/validation set`. המתוודה (`tuning`) מגדילה מודל של רשות נירונים ומחזירה אותו. נגידר אותו עם הפרמטרים שנרצה לעשות להם (באמצעות האובייקט `hp`). נסיף שכבות קונבולוציה: `Conv2D + MaxPooling2D` בollowה. נגידר `tuner` באמצעות (באמצעות RandomSearch). עם ה-CNN אנחנו מגיעים לשיפור מהותי ב-`accuracy` וב-`AUC` לעומת מודלים קודמים. ראיים גם דוגמה שמבצעת `Transfer Learning`.

```
# how many numbers?
np.prod(img.shape)
691200

# so how much does 1 360X640 rgb image weigh on disk
print(img.dtype, img.size * img.itemsize)
uint8 691200
```

תמונה 10.2: נקרא את התמונה, נקבל מטריצה ב-3 מימדים: (360,640,3) עם שלושה ערוצי צבע. נכפול את המימדים זה בהזה ונקבל כי יש לנו 691200 מספרים בזיכרון עבור התמונה. נראה היסטוגרמה של הצבעים השונים (RGB).

```
fig, ax = plt.subplots(nrows=2, ncols=2)
red = img[:, :, 0]
green = img[:, :, 1]
blue = img[:, :, 2]
ax[0, 0].imshow(img)
ax[0, 1].hist(red.flatten(), color = 'r')
ax[0, 1].set_ylim(0, 45000)
ax[1, 0].hist(green.flatten(), color = 'g')
ax[1, 0].set_ylim(0, 45000)
ax[1, 1].hist(blue.flatten(), color = 'b')
ax[1, 1].set_ylim(0, 45000)
fig.tight_layout()
fig.show()
```



התיאוריה של מודל לחיזוי

כאן לא נלמד אף מודל חיזוי חדש, אלא עוסוק בתיאוריה של מודלים לחיזוי, מעט יותר עמוק. הרצינו מanalyze הבחירה ללמידה תיאורית לאחר מעשה, הוא שכרגע יש לנו הרבה דוגמאות להמחשת התיאוריה.

מודלים לחיזוי

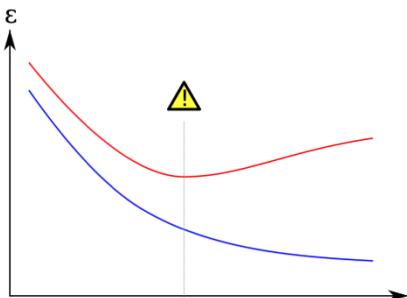
רקע:

עד כה דיברנו על ה-**sets** הבסיסי שבו א' הוא וקטור משתנים ו- y סקלר שנחנו מנסים למינן ולהזות. אם ע' ממשי – ורגסיה, ואם ע' קטגוריאלי – קלסיפיקציה. הפרדיגמה הכללית שלנו היא לשימוש בדאטא נפרד ללמידה (T_r) ודאטא אחר לחיזוי (T_e). הרבה פעמים זה אומר בפועל לקחת את הדאטא שיש לנו ולחולק אותו בעצמו ל-**Train**-**Test**. ראיינו אינטואיטיבית למה שנעשה בדבר זהה. מודגם הלמידה לרוב יכול להציג ל-0% טעות (overfitting) ואיכות המודל נשפטת על פי מודגם הטסט. אם יש לנו set test אחד ועל פי החלטתי לקחת לדוגמה עז בעומק 5 במקום עומק 4, אם נרצה לש Kapoor מה הביצועים הסופיים של המודל, האם אפשר לדוח על הביצועים של הטסט-טסט שבו השתמשנו כדי לבחור בין מודלים?

השיטות שראינו:

- המודלים הפרמטריים שראינו היו ורגסיה OLS ורגסיה לוגיסטיבית.
- מודלים א-פרמטריים שבבסיסם על הגדרת שכנות (KNN) ועצים (DT).
- שיטת אנסמבל מבוססת עצים כמו RF ו-Boosting.
- רשותות נוירוניים וספציית CNN.

ננסה לחשב על כל השיטות שלמדו בקורס אחורזה, באמצעות חשיבות רוחבית על כמה תכונות של מודלים לחיזוי. תכונה אחת מעניינת היא המסוכבות/**complexity** של השיטות (לא סיבוכיות). לכל שיטה יש פתרון **שניתן** לסוגב כדי לנوع בין **מודל פשוט** מאוד **למודל מורכב מאוד** (אולי מורכב מדי). ברגסיה – מספר המשתנים שכונסים למודל (k), בשכנים יש את גודל השכונה (k), בעצים יש את עומק העץ (max_depth), וברשותות נוירוניים יש הרבה היפר-פרמטרים כמו מספר הנוירונים בראשת. ככל שנוסף את הכפטור נקבל מודל מורכב ועשיר יותר לתיאור היחס בין המשתנים המסבירים, למשתנה הבלתי.

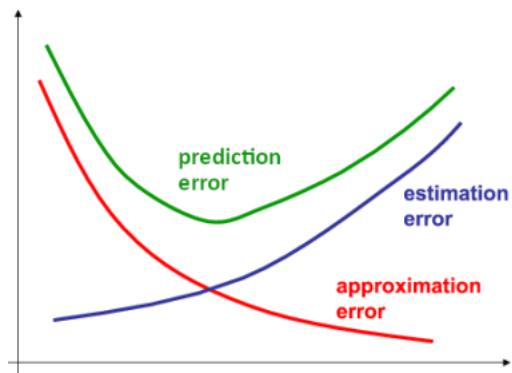


הדיםות הדומה לכל השיטות שראינו: על ציר ה- y טעות החיזוי (למשל RMSE), ועל ציר ה- x רמת המורכבות של המודל. ככל שהמודל מורכב וגייש יותר, שגיאה הלמידה יורדת (בכホל). ואילו, שגיאת הטסט (באdom) היא בעלת אופטימום מסוים. לא הוכחנו תאורטית למה זה קורה, נתנו אינטואיטיבית – המודל עם complexity האופטימלי, גמיש מספיק כדי לבטא יחסים מורכבים, אבל לא גמיש מדי כדי להתחייב את עצמו לכל רעש קטן במודגם הלמידה ומוביל ל-overfitting.

סוגי טיעויות:

- **הטעות שمبיאת עד כמה המודל שלנו מסוגל באמת לתפקידו את הקשר האמתי בין x ל- y .**
 - למשל, בדוגמה עם הסיבוי לחלות במחלה – אם הוא מושפע מ-3 משתנים (גיל, היסטוריה משפחיתית, משקל), אבל ננכис למודל רק 2 משתנים (גיל ומשקל), הטעות של המודל תגען מזה שהוא בכלל לא רואה את המשתנה השלישי, ולא מדובר בשפה הנכונה לקווט את התלות האמיתית – חסרים לו משתנים. הוא לא מושך מספיק.
 - מודל עם שגיאה גבוהה כזו, הוא מודל עם complexity נמוך, מעט מדי משתנים (ע"ז לא עמוק מספיק).
- **עד כמה חסירה לנו אינפורמציה ממודגם הלמידה כדי לחזות על נתונים חדשים, ולהתאים את המודל.**
 - בדוגמה של מחלות לב, נניח שאנחנו כוללים במודל לבדוק את 3 המשתנים שאנחנו צריכים. המודל שלנו עשיר בבדיקה כדי לבטא את התלות, ולא יהיה לו AE גבוהה.
 - אבל אם נלמד משתי תוצאות בלבד? אין שום יכולת לאמן את המודל. הטעות תיגרם לא מפני שהמודל לא נכון, אלא כי אין מספיק תוצאות כדי לקבל הערכה של הפרמטרים (כמו ברגסיה). אם משתמש בדאטא אחר, נגיע לפרמטרים אחרים. ההחלטה ברגע רועשת.
 - אולי מעט תוצאות כן יספיקו, כדי להתאים מודל פשוט יותר – כמו ורגסיה עם משתנה אחד – אין צורך במיילון תוצאות. ה-**zero-error**, estimation, כמו אנחנו מדיקים בשערון הפרמטרים שלנו, יהיה קטן. אם משתמש בדאטא אחר, סביר שנגיעה לאותם פרמטרים במקרה של קו ישר.

אנו מגעים ל-trade-off



- מודל פשוט מדי, עם complexity נמוך מדבר בשפה "דלתה", יהיה לו AE גובה אבל EE נמוך, כי יהיו לו מעט פרמטרים שנשערכ הטוב.
- מודל מורכב ועמוק, יהיה עם AE נמוך כי הוא יכול לבטא יחסים מסובכים אבל EE גבוהה, כי לצורך המון תוצאות כדי לדיק בשערורם הפרמטרים שלנו.

אנו מעניינת Test Error, שגיאת החיזוי על נתונים שהמודל לא ראה. הציור מפרק אותה לשני מרכיבים (ה-AE וה-EE). אנו רואים את צורת ה-U המוכרת לנו, ומ-binנים איך היא נוצרת.

בכל שהמודל מורכב יותר, AE יותר, אבל EE עולה. אנו מוחפשים במרחב גדול יותר של מודלים, וצריכים הרבה יותר נתונים כדי לשערר אותה. בחיבור של הטיעויות הללו יש נקודה אופטימלית של מורכבות – וכן קיבל את צורת ה-U.

אם נגדיל עוד ועוד את הדאטא שיש לנו לאימון, על איזו מסוג הטיעויות נשפיע? EE! ככל שיש יותר נתונים, אפשר ללמוד מודלים מרכיבים יותר, באופן מדויק יותר. אין לה השפעה על AE, הרי שגם אם 2 תוצאות בלבד ניתן לתאר מודל מורכב עם המון משתנים, הביעה תהיה ב-*estimation* שלו.

Bias-Variance tradeoff

נתמקד ברגרסיה, שם אנו חנו מנחים $\sigma^2 \sim N(0, \epsilon) = f(x) + \epsilon$. הערך הנמדד שלו הוא לא בדוק הערך הצפוי שלו, אלא התוחלת ווד רעש כלשהו. בעת אנו חנו לוקחים נתונים Tr וממנו לומדים את $E[y|f(x)] = \hat{f}(x)$. אנו יכולים לבדוק את הביצועים של המודל על מבחן הטעט: $E[\hat{f}(x_0)] = \hat{y}_0$.

נחשב לרגע על כל התהילה של דוגמת נתונים, בנית מודל, למידה שלו מהנתונים – כתהיליך אקרואן. נסתכל על התוחלת של ההפסד. לדוגמה, ברגרסיה MSE על פניו ביצוע התהילה זהה הרבה פעמים, אם היה training data קצר שונה כל פעם. גם המודל $\hat{f}(x)$ הוא משטנה מקרי, שמנוסס על מקורות מוגם הלמידה. **תגיאת תוצאות חדשה, ונסתכל על תוחלת ההפסד שלה,** התנהגות האופיינית על פני הרבה מוגם למידה:

$$\mathbb{E}(y_0 - \hat{f}(x_0))^2 = \mathbb{E} \left(\underbrace{(y_0 - f(x_0))}_A + \underbrace{(f(x_0) - \mathbb{E}(\hat{f}(x_0)))}_B + \underbrace{(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0))}_C \right)^2,$$

אנו מוסיפים ומחסרים את $\hat{f}(x_0)$ שהוא התלות האמיתית של y ב- x , וגם נוסיף ונחסר את $\hat{f}(x_0)$.

אלו מהගורמים תלויים באקרואיות מוגם הלמידה?

$$A = y_0 - f(x_0)$$

- A לא תלוי כלל, הוא מבטאאמת, רעש טבעי שקיים ולא נובל להקטין, ϵ . זהה טעות שאין הרבה מה לעשות לגיביה.

$$B = f(x_0) - \mathbb{E}(\hat{f}(x_0))$$

- B גם לא תלוי שוכן התוחלת היא קבוע. הינו מצפים שמספר זה יהיה 0, המודל עשיר מספיק כדי שבתוחלת הוא קירוב טוב לחסית $\hat{f}(x_0)$. זה AE.

$$C = \mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)$$

- C תלוי בדוגמ הלמידה, המרחק של $\hat{f}(x_0)$ מהתוחלת, אם נעללה בריבוע וניקח תוחלת = קיבלנו את שונות $\hat{f}(x_0)$. זה EE, האם יש מספיק נתונים כדי שם נחזור על התהילה עם נתונים נוספים.

$$\begin{aligned} \mathbb{E} \left(\underbrace{(y_0 - f(x_0))}_A + \underbrace{(f(x_0) - \mathbb{E}(\hat{f}(x_0)))}_B + \underbrace{(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0))}_C \right)^2 \\ = \mathbb{E}A^2 + B^2 + \mathbb{E}C^2 + 2B \cdot \mathbb{E}A + 2B \cdot \mathbb{E}C \end{aligned}$$

כעת, כשבנעה את סכום שלושת הגורמים בሪבוע, נקבל כל אחד מהם בריבוע ועוד 2 כפול מכפלת של בלוז. בשנייה תוחלת נישאר עם הביטוי $\mathbb{E}A^2 + B^2 + \mathbb{E}C^2$:

- Irreducible Error**: השונות של ϵ , הרעש הקיים בטבע שגים תחת מודל מדויק לא נצליח להפחית. זה לא תלוי בDATA שלנו, ובഗדרה אין DATA לחזות אותן.

- Squared Bias**: זה מודל עם "התיה" הוא מוטה לelowם מוסיפה (נכיה עץ החלטה) אל מול מה שהוא מנסה לחזות.

- (קו יגاري). מודל עם "התיה" הוא מוטה לelowם מוסיפה של היפותזות, באשר ההיפותזה האמיתית לאו דווקא נמצאת שם.

- Variance**: השונות של החיזוי, וזה מודל EE. ככל שמדובר הלמידה גדול, השונות הזאת תקינה.

- בשאר הביטויים של המכפלות הן 0.

חשיבות השונות של המודל: מודל עם שונות נמוכה, גם אם ניקח דאטא קצר אחר, נקבל חיזוי מאד דומה – האמידה תהיה יציבה. מודל עם שונות גבוהה, אם ניקח דאטא קצר אחר, נקבל חיזוי שונה, האמידה לא יציבה, ועודת. מתכוון לטיפול בעיות בזאת יכול להיות למשל להגדיל את מדגם הלמידה.

ראינו את זה מתמטית בשדריבנו על ממוצע המדגם המקורי, שהשונות שלו היא $\frac{\sigma^2}{n}$ ובכל ש- n גדול יותר, כבה היא תקנן.

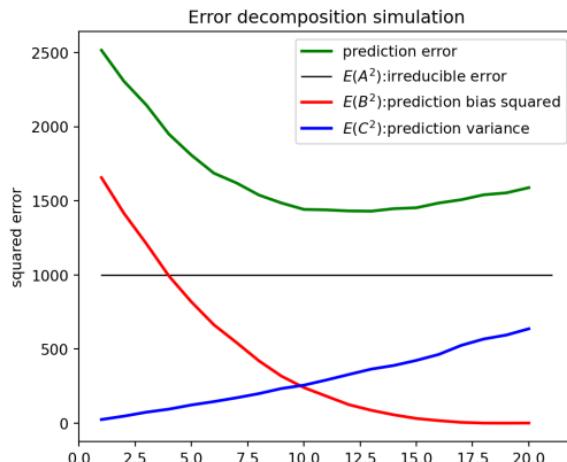
שגיאת החיזוי שלנו היא סכום הגורמים הריבועיים: שגיאת רוש טبعי שאינה תלולה במודל + שגיאת הטיה ריבועית שתלויה במודל אבל לא בדאטא + שגיאת שונות המודל שתלויה גם במודל וגם בדאטא.

אם נחזור לתמונה שציירנו, הקו הירוק צריך להיות סכום של 3 קווים: האdom + הכחול + ה-*r*-error.

$$\mathbb{E}(y_0 - \hat{f}(x_0))^2 = \text{irreducible error} + \text{squared bias} + \text{variance}$$

שימוש בפרק ל-bias-variance

אפשר ממש לחשב את הטיעויות הראשית והרואה שיכל שמודל מסוים מרכיב יותר, ה-*AE* יותר, לעומת *EE* שעולה. לדוגמה ברגסיה לינארית, ככל שנוסף עוד משתנים, המודל יהיה מרכיב ומדויק יותר: ההטיה קטנה ותדר. מצד שני השונות גדל ותגדל, האומדים שלנו יוכלו ליותר ולא מדויקים ונדרקן לעוד דאטא כדי שזה לא ידריך. באופן דומה נחושב על KNN השופך – ככל שיש פחות שכנים המודל נראה מרכיב ומדויק יותר, ה-*AE* יותר אבל גם ה-*EE* עולה, אנחנו תולמים את מבטחנו במעט מאוד שכנים וזו החלטה מאוד רועשת.



נראה סימולציה שמחישה את הדפוס הזה עם מודל לינאר. יש לנו וקטור משתנים מסוירים $\mathbb{R}^2 \in x$. המקדים שלו מסודרים מהכי חשובים משמשים x_0 עד x_N שהוא אחד בכולם x_1 עד x_{N-1} . אם נחשב אותו הוא אמור להיות ϵ + 43.6 כאשר $N \sim 1000$. יש לנו 50 תצפיות במדגם. ניציר הרבה מוגדים עם היחס האמייני הזה, ובכל פעם נבנה מודל f אחר עם מספר משתנים הולך וגדל, מרכיב יותר ויתר. בצד ה-*test*, נוכל לאמוד ממש אמפירית את 3 הגדים שהישבו, ולראות שהם מסתובבים בשגיאת החיזוי.

סיכום:

- *straight line* (מודל עם complexity נמוך בא-עם):
- *AE* גבוהה (גובה – לא טובפס את ה-*train*)
- *EE* נמוך (שינויים נמוכה – הביצועים שלו על ה-*Tr* ועל ה-*Te* **עקביים** – אפילו אם הם לא מאוד טובים).
- *squiggly line* (מודל עם complexity גבוהה בא-עם):
- *AE* נמוך (גובה – מושלם על ה-*train*, מושלם על ה-*test*)
- *EE* גבוהה (גובה – ביצועים מאוד **שוניים** בין *Tr* ל-*Te* – מתאים את עצמו ל-*train* אבל לא מכליל מספיק בשבי תצפית חדשה מה-*test*).

Method	Complexity param(s)	Low Complexity	High Complexity
Linear/logistic regression	Number of variables	Few variables	Many variables
k-NN	Number of neighbors	Many neighbors	Few neighbors
Tree	Depth	Shallow	Deep
Neural Nets	Number of hidden nodes and layers	Few	Many



שיטת העבודה עם דatasets

קיטוֹס-וְלִידָצִיה:

עד כה חילקו את הדatasets ל-80% למידה ו-20% טסט. אם יש כפטור של model complexity, לא סביר להשתמש במודם הטעט, גם לבוחר אותו, וגם לדוח על שגיאות החיזוי הצפויות מהמודל. נIRONן יותר סטטיסטיות לחלק את הדatasets לשולש חלקים: 60% למידה, 20% ולידציה, 20% טסט. יש דרך ייעילה יותר להשתמש בנתונים.

:10-fold cross validation

1. נחלק את הדatasets ל-10 חלקים שווים באופן אקראי.
 2. נ徇ור 10 פעמים:
 - a. נאמת את המודל על 90% מהdatasets.
 - b. נעשה אבלואציה על 10% מהdatasets (the-fold) שלא השתמשנו בו לאימון.
 3. כל fold משמש בתורתו במודם holdout בשא-9 האחרים משמשים במודם הלמידה. כדי לדוח על הביצועים של המודל, נסתכל על ממוצע טעות החיזוי על 100% מהdatasets על פני כל folds.
- בר לא אימנו על 60% או 80%, כל פעם על 90% ומיצענו את עוקמת טעות החיזוי על פני מספר עותקים של הדatasets, כל פעם במודם אחר. כמו בכל מיצוע שראינו עד כה, גם החלק הזה של בחירת complexity עצמו יהיה רושע פחות, השונות של התהיליך תקטן. אם הדגש של המודל הוא על חיזוי (למשל בסביבת production), יש לאמן אותו פעם אחת אחרונה על 100% מהחיזויות, בר שמודל סופי שנאנחנו באמצעות עוזרים לו deployment יאומן על כל הדatasets שהוא ברשותנו ולפיכך אמרור רק להשתפה.

:LOOCV

אפשר גם לחלק ל- n חלקים בשזה גודל הדatasets, ואז נאמת אותו על 99.9% מהנתונים, בשנאמן את המודל על 1 – n תוצאות, ונבדק על התוצאות האחרונות שנותרה.

$L_{n-fold} = \sum_{i=1}^n L\left(y_i, \hat{f}^{(-i)}(x_i)\right)$ זהה שיטה בשם leave-one-out cross validation. טעות החיזוי הכללית שלמן תהיה סכום ההפסדים על פני n מודלים. אם נרצה לבצע בפטור n , נctrיך להוסף גם איבדק司 complexity, ולעשות לכל בחירה של k .

:יתרונות של CV

- שימוש יעיל יותר בנתונים, אפשר לאמן מודל על 90% מהdatasets או 1 – n במקום 80%.
- כל הדatasets משתמשים ב-set, test set, כל פעםfold אחר לבן האבלואציה עצמה עם שונות קטנה יותר, וכוונון הcpfטור שלמן יהיה יציב יותר ומוסבל יותר.

:הסרונות של CV

- חסרון חישובי – אם מדובר במודל שלוקח זמן לאמן, נctrיך הרבה יותר חישוב, במיוחד כדי לבון פרמטרים. אפשר לחשב על מימוש חכם שימקבל את האימון.
- המודל יבחן פעמי אחדות, והטעות הממוצעת על פני folds אולי לא משקפת את המבחן האמיתי של המודל. גם כאן, אם הדatasets גדול מאוד אפשר לחשב על שילוב הגישות. בכל זאת להשאר קצת datasets בצד ל-test set שלא ניגע בו, והוא דומה לנו את הביצועים בסביבת prod.

:רגוליזציה:

ניקח למשל את המודל של רגרסיה ליניארית. כדי להפוך את המודל ליותר מורכב, דיברנו על הוספה משתנים נוספים. הרגוליזציה מאפשרת לשנות complexity של המודל בגישה יותר מhocמת, בלי לשנות את מספר המשתנים במודל. בשנאננו משתנים את מספר המשתנים, אנחנו מרחיבים או מצמצמים את מרחב החיפוש של הפרמטרים שלמן: אם יש לנו $2 = k$ משתנים המרחב נורא קטן, ואם $100 = k$ המרחב עצום.

רגוליזציה שולטת בגודל מרחב החיפוש בדרך אחרת, על ידי הגבלת הנורמה של וקטור המPARAMeters. במקום לרוחב ב- k , נישאר ב- k אבל לא נרצה לוקטור β להיות ב- \mathbb{R}^k , נשמר אותו בטוחה של כדור מסיים כדי שלא ישתול.

:ראשית יש לבחור נורמה, המקבילות הן:

- נורמת L2: סכום בריבוע. הfrac{z}{\sqrt{\sum_{j=1}^p \beta_j^2}}. Ridge regression.
- נורמת L1: סכום בערך מוחלט. נקרא $|\beta|_1 = \sum_{j=1}^p |\beta_j|$. Lasso regression.

במקום להביא למינימום את RSS בלי שום אילוץ, להביא אותו למינימום תחת האילוץ שהנורמה לא גדולה מדי, קטנה מקבועה כלשהו. זה ממש מיצאת מינימום בתחום הבודור שסביר הראשית, במרחב הפרמטרים ה- k -dimensional זהה:



$$\min_{\|\beta\|^2 \leq c} RSS(\beta)$$

בעה שcola היא לחת את הקритריון למינימום, ולהוסיף לאילוץ קритריון כופל לגראמ' ג', תוך ענישה של וקטור המקדים. אם נשים ג' כבד מאוד, המקדים ייאלצו להיות קטנים ולהתקרב לראשית. לחילופין, נעניש באופן קל, ונאפשר למקדים להיות גדולים יותר:

$$\min_{\beta} RSS(\beta) + \lambda \|\beta\|^2$$

בדרכ' זו אנחנו מקטינים את השונות של וקטור המקדים.

פתרון סגור ל-RSS (Ridge Regression): יש פתרון סגור ל-Ridge Regression:

We want to minimize penalized RSS: $PRSS(\beta) = \|Y - X\beta\|^2 + \lambda \|\beta\|^2$.

Differentiating relative to β and equating to 0 :

$$\nabla_{\beta} PRSS(\beta) = -2X^T Y + 2X^T X\beta + 2\lambda\beta = 0.$$

Solution: $\hat{\beta}(\lambda) = (X^T X + \lambda I_p)^{-1} X^T Y$.

בשונה מפתרון OLS, כאן נוסף עוד קבוע קטן לאילכון של המטריצה $X^T X$, עיבינו אותו קצת. אפשר לחשב שהוא נראה עבשו בולט יותר כמו רכס (ridge). מלבד זאת, אפשר להראות שאנו מקטינים את השונות.

QUIZ 14

שאלה 1: בכלל, סכנת overfitting למדגם הלמידה גדולה יותר בשיטות לוקאליות כמו KNN ועצי החלטה מאשר שיטות גlobליות, מודלים פרמטריים כמו רגסיה ליניארית ולוגיסטי. **נכון**, מודל פרמטרי כמו רגסיה יכול להיעשות מורכב וगמיש כל כך שעשו אובייפטיניג לממדגם הלמידה אבל זה יצריך מהחוקר לפרט את המשתנים ואת כל הטרנספורמציות עליהם, מה שבכלל לא קורה, מודלים פרמטריים בדרך כלל מכירחים אותנו לחושם פשוטות. וזאת לעומת KNN ועצי החלטה שבאמצעות פרמטר יחיד יכולים להיות מורכבים למדי.

שאלה 2: איזה מה הבאים נכון לגבי λ העונש על נורמת L2 של מקדמי הרגסיה β ?

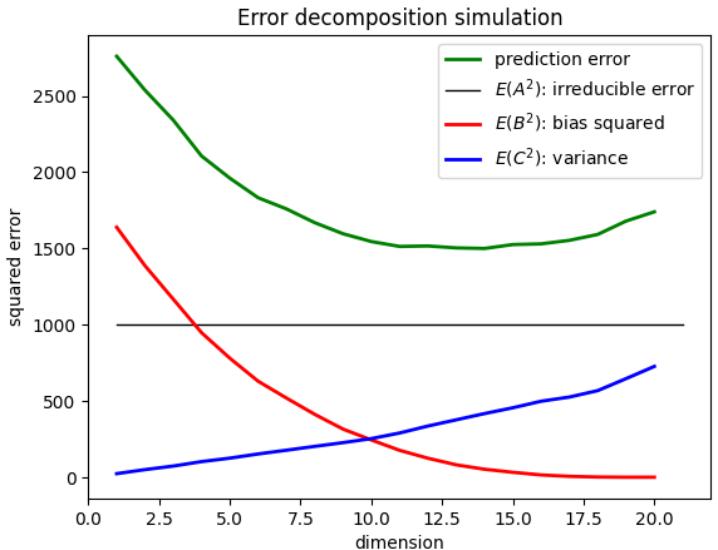
- ככל שיקטן בר בעניש פחות את המקדים, נקטין את שונות החיזוי, ה- $bias$ עלול לגודל אבל טעות החיזוי על נתונים חדשים עשויה לקטן – ככל שיקטן בר נקבל אומדיים גדולים יותר (הטיה קטנה) אבל רועשים יותר (שונות גדולות).
- ככל שיגדל בר בעניש יותר את המקדים, נקטין את $bias$, השונות עלולה לגודל אבל טעות החיזוי על נתונים חדשים עשויה לקטן – ככל שיגדל בר נקבל אומדיים קטנים יותר (שונות קטנה) אבל מושעים יותר (הטיה גדולה).
- ככל שיגדל בר בעניש יותר את המקדים, נקטין את שונות החיזוי, ה- $bias$ עלול לגודל אבל טעות החיזוי על נתונים חדשים עשויה לקטן.
- ככל שיקטן בר בעניש יותר את המקדים, נקטין את שונות החיזוי, ה- $bias$ עלול לגודל אבל טעות החיזוי על נתונים חדשים עשויה לקטן – ככל שיקטן בר בעניש פחות את המקדים.

שאלה 3: גודל המדגם שלנו הוא ג'. נבחר בקריטריון $VOCV$ לבחירת מספר הנוירונים בכל שכבה בראשת נוירונים, כשהאפשרויות הן 10, 20, 30, 40 או 50 נוירונים, וכל השאר קבוע. כמה פעמים נדרש להריץ את הרשת?

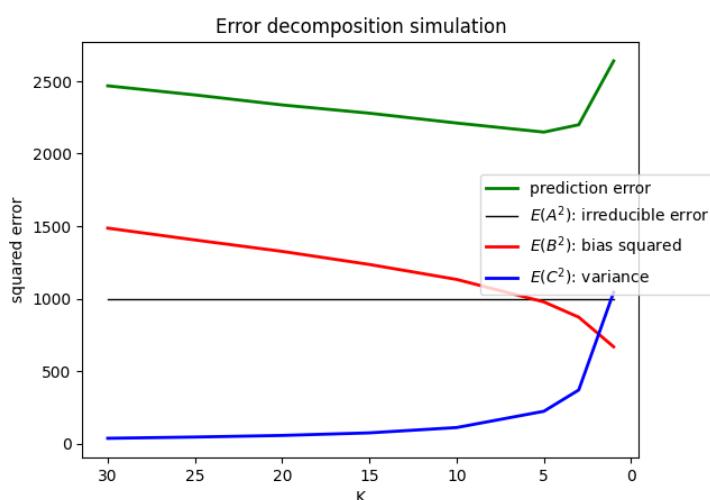
- תלוי במספר הפלדים
- 5 ועוד 1 על כל המדגם אחריו שבחרנו את מספר הנוירונים על מדגם הטעט
- **5 ועוד 1 על כל המדגם אחריו שבחרנו את מספר הנוירונים**
- מ ועוד 1 על כל המדגם אחריו שבחרנו את מספר הנוירונים – איך תבחרו בכמה נוירונים להשתמש?

תרגול 11 – התיאוריה של מידול לחיזוי

Bias-Variance Tradeoff

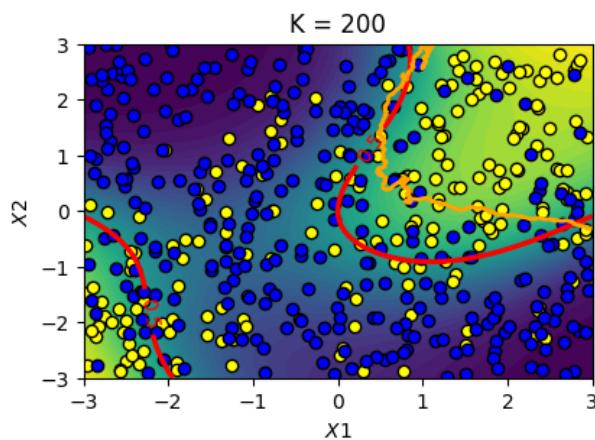
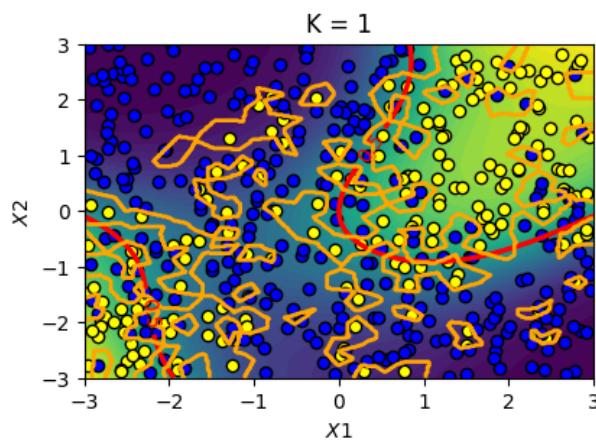


נסתכל על מודלים שונים, ונפרק את השגיאות שהם משיגים.
רגסיה לינארית: ה- AE קטן עד שmagiu ל-0 (המודל הוא לינארי, אנחנו מחפשים בעולם לינארי, ולכן המודל הוא חסר הטיה, אין EE). ה- EE גדול כי השונות הולכת וגדלה.



רגסיה עם KNN: כאן אבחנו כל פעם עולים במספר השכנים. בשרטוט K-1 יוד, בכוונה כדי לראות את אותו הדפוס של bias variance וחסר הטיה לכל תופעה, ובכל שיש פחות שכנים יותר ויתר הטיה והוא פחות מתוחכם (זה אומר הרבה פחות שונות) – ה- bias -error עולה אף variance-_error יורד. $K=1$ נקבע לבדוק את ה-irreducible error.

סיווג עם NNN: יש כאן מפה של הסתברויות. אפשר לראות מאריך $K=1$ הוא יותר מדי ספציפי, גמיש מדי.



שאלה:

I have a predictive model (built using OLS, logistic regression, or any other method).

Now I add more irrelevant covariates (that is, explanatory variables that are not informative about y) to my data and refit the predictive model.

What is expected to happen to my approximation error (bias), estimation error (variance), and overall prediction (test) error, as I keep adding those covariates?

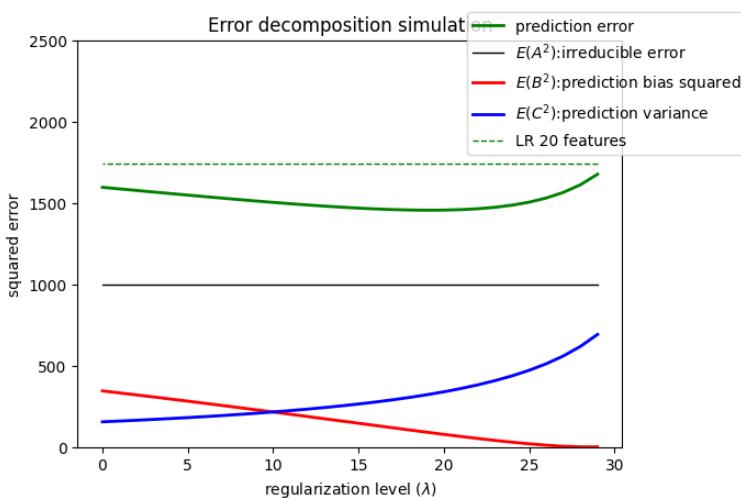
1. Decrease, unchanged, decrease, respectively
2. Decrease, increase, unknown, respectively
3. Unchanged, increase, increase, respectively
4. Decrease, increase, decrease, respectively

התשובה היא **3**. קצת קשה להגיד שה-bias לא משתנה כי אנחנו בן נראה ירידה. אם נמשיך להוסיף ולהוסיף משתנים לא קשורים, מתישנו נצליח להתאים על נתונים הלמידה באופן מושלם ונגיע ל-EAE של 0%. מתיישנו הואobar לא ישתנה. לגבי ה-variance בוודאות יגדל, המודול יותר ויתר ספציפי, מושלם על-h-train אבל לא מצליח מספיק בשבייל תצפית חדשה ב-test. לבסוף, ה-error prediction הכללי אם כן יעלה בסופו של דבר.

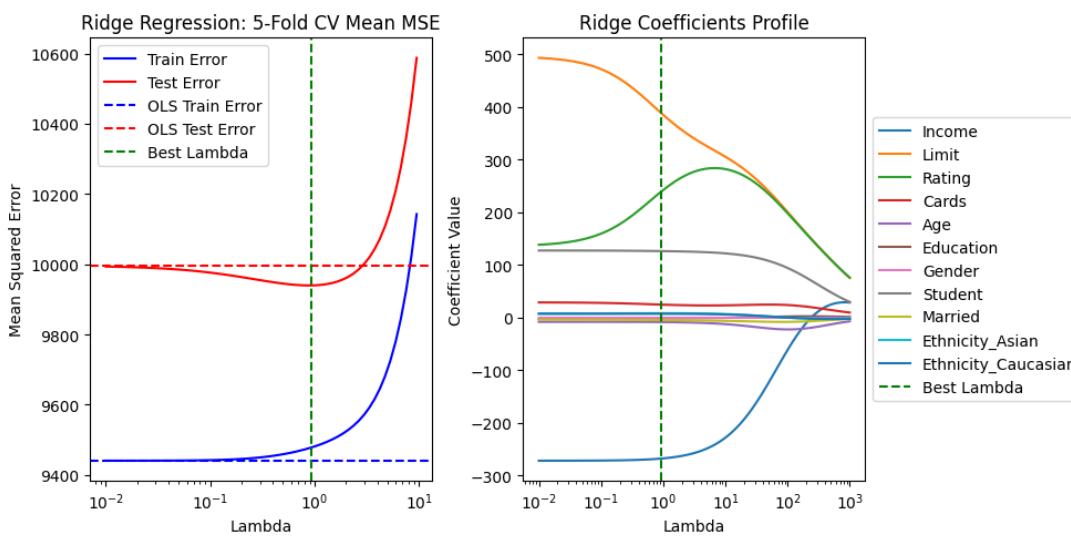
Best subset selection

1. M_0 model: predict $\hat{y} = \beta_0 = \bar{y}$
2. For $k = 1, \dots, p$:
 - i. Fit all $\binom{p}{k}$ models containing k features
 - ii. Pick the best M_k with $\min RSS$
3. Select the best model from M_0, \dots, M_p with the C_p/AIC criterion or CV

הגסית Ridge מבצעת הגבלה על הנורמה של וקטורי המקבדים. נספר את ה-prediction (מודל שהוא קצת שונה מהמודל הלינארי, מכnis קצת bias אבל נקטין מאוד variance). וגם את ה-interpretability (כמה קל לראות את המודל ולהבין מה הוא עשו).



אנחנו מסתכלים על השגיאות בכל שעהונש (λ) הולך ונעשה בבד יותר. זה עד hyper-parameter שצורך לבחור (לרוב CV עם).



נסתכל על נתונים של לקוחות בנק (credit). אנחנו רוצים לחזות את ה יתרה (balance). כדי לבחור את ה- λ נבצע 5-fold CV. את אותו הדבר נעשה גם עם גרסיה וגיליה. אפשר לראות את ההשוואה. 1 = 10^0 קיבלנו את האופטימום.