תכנות מתקדם ב-R – תרגיל מסכם

תאריך הגשה: 31.7.2016

מבוא

מטרת תרגיל זה היא התנסות בתכנות רחב היקף המשלב כמה מהנושאים שנלמדו בקורס, ולמידה עצמאית של חבילת R חדשה. **התרגיל צפוי לדרוש עשרות שעות עבודה**, ולכן מומלץ להתחיל לעבוד עליו בהקדם.

התרגיל עוסק בנתוני זכייה של מדליות אולימפיות, הלקוחים מהאתר www.databaseolympics.com. הוא מורכב מ-5 חלקים:

1. הורדת הנתונים הדרושים מהאתר למחשב, ושמירתם כמסגרות נתונים ב-R.
2. עבודה עם SQL: יצירת טבלאות וכתיבת שאילתות.
3. עיבוד נתונים והצגה גרפית עם ggplot.
4. חיזוי של זכייה במדליות.
5. יצירת אפליקציית "Shiny" – עמוד אינטרנט אינטראקטיבי המציג נתוני זכייה במדליות.

בנוסף לניקוד על נכונות הפתרון בכל חלק (כמצוין בהמשך), יוענקו 10 נקודות על "תכנות נאות": תיעוד הקוד בהערות, בחירת שמות אינפורמטיביים למשתנים ולפונקציות, חלוקת עבודה סבירה בין פונקציות, הזחות (אינדנטציות) מסודרות בתנאים ולולאות, וכו'.

את התרגיל יש להכין ולהגיש **ביחידים**, ולא בזוגות או בקבוצות אחרות. בבדיקת העבודות יושם דגש על יושר והגינות.

אנא קראו היטב את ההוראות. בכל חלק מצוין בדיוק איזה קבצים יש להגיש, מה כל אחד מהם צריך להכיל, מה בדיוק הקוד צריך לעשות, איך נראים הגרפים, מהם שמות העמודות במסגרות הנתונים, וכו'. פתרונות שלא יקפידו על מילוי ההוראות יקבלו ניקוד חלקי.

לפני תחילת העבודה, מומלץ לשוטט באתר ממנו יילקחו הנתונים ולהתרשם מהמבנה שלו. האתר לא מעודכן בתוצאות האוליפיאדה האחרונה (לונדון 2012), וגם הנתונים הקיימים בו כוללים טעויות. למשל, בעמוד המדליות של ענף הספורט "קרב חמש" (pentathlon) באולימפיאדת 1976 (ראו כאן), מופיע בטעות ספורטאי נטול מדליה בשם Daniele Masala בנוסף לשלושת המדליסטים הנכונים.

התרגיל עוסק רק באוליפיאדות הקיץ, ולכן ניתן להתעלם מכל הנתונים הקשורים לאולימפיאדות חורף.

הבהרה בנוגע לטרמינולוגיה: באולימפיאדות מתקיימות תחרויות (events) בכמה ענפים (sports). למשל, "קפיצה לגובה גברים" (High Jump Men) ו"100 מטר משוכות נשים" (100m Hurdles Women) הם תחרויות בענף "אתלטיקה קלה" (Track & Field), ואילו "200 מטר פרפר גברים" (200m Butterfly Men) ו"100 מטר חזה נשים" (100m Breaststroke Women) הם תחרויות בענף "שחייה" (Swimming).

הקוד שאתם מגישים צריך לרוץ בלי הודעות שגיאה או אזהרה (פרט להודעות אזהרה בדבר גירסת ה-R הדרושה לחבילה שאתם טוענים). בכל השאלות אפשר ורצוי להשתמש בפונקציות שכתבתם עבור שאלות קודמות.

המשתנים הגלובליים היחידים שמותר להשתמש בהם הם מסגרות הנתונים שתכינו בחלק 1. כל המשתנים האחרים הדרושים לפונקציות שאתם כותבים צריכים או להיות מועברים כארגומנטים, או להיות מוגדרים בתוך הפונקציה.

קובץ זה צפוי להתעדכן בתקופת הכנת התרגיל, כדי להבהיר נקודות מסוימות או לתקן טעויות. עקבו אחרי תאריך העדכון האחרון המופיע בראש עמוד זה משמאל.

חלק 1 – Web Scraping (25 נקודות)

בחלק זה עליכם להוריד נתונים מהאתר www.databaseolympics.com ולשמור אותם כמסגרות נתונים.
הקפידו ששמות העמודות יהיו בדיוק כפי שמופיע להלן.

השמות של חלק מהספורטאים ושל חלק מהתחרויות כוללים תווים לא-סטנדרטיים (למשל המתעמל הנורבגי Øistein Schirmer, או תחרות הסיף שנקראת Épée). אין צורך להתמודד עם הקידוד של תווים כאלה, ואפשר להשאיר אותם כפי שהם מתקבלים אצלכם ב-R.

כדי לפתור חלק זה, ניתן להשתמש בביטויים רגולריים, בחבילה XML, או בשילוב של שני הכלים. להלן שלושה טיפים עבור המשתמשים ב-XML:

- אפשר להוריד טבלת HTML שלמה, בפורמט של מסגרת נתונים, ע"י הפקודה readHTMLTable. למשל, הטבלה המרכזית בעמוד הזה היא טבלה מספר 3, ולכן:

```
> library("XML")
> URL <- "http://www.databaseolympics.com/games/gamessport.htm?g=3&sp=BOX"
> tab <- readHTMLTable(URL, which=3)
> head(tab)
          Event         Athlete Country Result  Medal
1     Flyweight George Finnegan     USA          GOLD
2                  Miles Burke     USA        SILVER
3  Bantamweight     Oliver Kirk     USA          GOLD
4               George Finnegan     USA        SILVER
5 Featherweight     Oliver Kirk     USA          GOLD
6                 Frank Haller     USA        SILVER
```

- כדי להוריד מקוד HTML את כל הקישורים שמתחילים במחרוזת מסוימת, למשל המחרוזת "/games", אפשר להשתמש בפונקציה xpathSApply באופן הבא:

```
> library("RCurl")
> library("XML")
> URL <- "http://www.databaseolympics.com/games/gamesyear.htm?g=19"
> parsed.page <- htmlParse(getURL(URL))
> URL.vec <- xpathSApply(parsed.page, "//a[starts-with(@href, '/games/')]",
                         xmlGetAttr, 'href')
> head(URL.vec)
[1] "/games/gameslist.htm"
[2] "/games/gamessport.htm?g=19&sp=ARC"
[3] "/games/gamessport.htm?g=19&sp=ATH"
[4] "/games/gamessport.htm?g=19&sp=BAS"
[5] "/games/gamessport.htm?g=19&sp=BOX"
[6] "/games/gamessport.htm?g=19&sp=CAN"
```

- כדי לקבל במקרה האחרון את עוגני הקישורים (הטקסט שמופיע על המסך), ולא את הכתובות, אפשר לכתוב:

```
> val.vec <- xpathSApply(parsed.page, "//a[starts-with(@href, '/games/')]",
                         xmlValue)
> head(val.vec)
[1] "Olympic Games" "Archery"       "Track & Field"
[4] "Basketball"    "Boxing"        "Canoeing"
```

1.1. צרו מסגרת נתונים בשם games.df המכילה את השנים, המדינות, והערים שבהן נערכו המשחקים האולימפיים.

```
> head(games.df)
  year        country      city
1 1896         Greece    Athens
2 1900         France     Paris
3 1904  United States St. Louis
4 1906         Greece    Athens
5 1908 United Kingdom    London
6 1912         Sweden Stockholm
```

1.2. צרו מסגרת נתונים בשם medals.df המכילה את השנה, הענף (sport), התחרות (event), קוד הספורטאי (athlete_id), וסוג המדליה, עבור כל אחת מהזכיות.

```
> head(medals.df)
  year         sport    event athlete_id  medal
1 1896 Track & Field 100m Men BURKETOM01   GOLD
2 1896 Track & Field 100m Men HOFMAFRI01 SILVER
3 1896 Track & Field 100m Men  LANEFRA01 BRONZE
4 1896 Track & Field 100m Men SZOKOALA01 BRONZE
5 1896 Track & Field 400m Men BURKETOM01   GOLD
6 1896 Track & Field 400m Men JAMISHER01 SILVER
```

את קוד הספורטאי ניתן לחלץ מתוך הקישור לעמוד הספורטאי. למשל, בעמוד הזה, הקישור לעמוד של Michael Phelps הוא http://www.databaseolympics.com/players/playerpage.htm?ilkid=PHELPMIC01. התווים האחרונים בקישור הם PHELPMIC01, וזה הקוד שלו. עליכם לשמור את כל קודי הספורטאים באותיות גדולות (באתר חלקם כתובים באותיות גדולות וחלקם בקטנות).

השמיטו ממסגרת הנתונים את השורות של ספורטאים שמופיעים בה בטעות, מבלי שזכו באף מדליה, כמתואר במבוא (יש 3 שורות כאלה). העמודה medal צריכה בסופו של דבר להיות פקטור עם 3 רמות: GOLD, SILVER, BRONZE.

אין צורך לשמור במסגרת הנתונים את העמודות "מדינה" (Country) ו"תוצאה" (Result) שב-HTML.

1.3. צרו מסגרת נתונים בשם athletes.df המכילה עבור כל ספורטאי את קוד הספורטאי שלו, השם הפרטי, שם המשפחה, וקוד המדינה שלו:

```
> head(athletes.df)
  athlete_id  first_name     last_name country_id
1 AABYEEDG01       Edgar         Aabye        DEN
2 AALTOARV01        Arvo      Aaltonen        FIN
3 AALTOPAA01       Paavo      Aaltonen        FIN
4 AALVIKAR01        Kari Aalvik Grimsb        NOR
5 AAMODKJE01 Kjetil Andr        Aamodt        NOR
6 AAMODRAG01    Ragnhild        Aamodt        NOR
```

1.4. צרו מסגרת נתונים בשם countries.df המכילה את קוד המדינה ושם המדינה של כל המדינות שזכו במדליות באולימפיאדה:

```
> head(countries.df)
  country_id   country_name
1        AFG    Afghanistan
2        ALB        Albania
3        ALG        Algeria
4        ASA American Samoa
5        AND        Andorra
6        ANG         Angola
```

מה להגיש

קובץ בשם project-1.R שמכיל את הסקריפטים שכתבתם. הקובץ צריך להיראות בדיוק כך:

```
############# question 1.1 #############

1.1 הקוד עבור שאלה

############# question 1.2 #############

1.2 הקוד עבור שאלה

############# question 1.3 #############

1.3 הקוד עבור שאלה

############# question 1.4 #############

1.4 הקוד עבור שאלה
```

חלק 2 – SQL (15 נקודות)

השאלות בחלקים 2-5 של התרגיל מתבססות על מסגרות הנתונים שהתבקשתם ליצור בחלק 1. על מנת לאפשר לכם להתקדם גם אם לא הצלחתם ליצור חלק ממסגרות נתונים אלה, ניתן להוריד אותן מאתר הקורס ולהשתמש בהן. כדי להוריד את מסגרת הנתונים games.df, למשל, הורידו את הקובץ בשם זה מהאתר, והריצו את הפקודה load(games.df). מסגרת הנתונים תיטען אז לזיכרון.

2.1. צרו בסיס נתונים בשם olympics.db המכיל ארבע טבלאות, המתאימות לארבע מסגרות הנתונים שהכנתם בחלק 1: Games, Medals, Athletes, Countries (שימו לב ששמות הטבלאות מתחילים באותיות גדולות).

2.2. כתבו שאילתת SQL שתוצאתה היא רשימת הענפים שנכללו באולימפיאדת 1976, ממוינים בסדר אלף-ביתי:

```
               sport
1            Archery
2         Basketball
3             Boxing
4           Canoeing
5            Cycling
6             Diving
...
```

2.3. כתבו שאילתת SQL שתוצאתה היא רשימה של המדינות שאירחו אולימפיאדות, ומספר הפעמים שכל מדינה עשתה זאת. הרשימה צריכה להיות ממוינת ראשית לפי מספר האירוחים (בסדר יורד), ואז לפי א"ב של שם המדינה:

```
          country no_games
1   United States        4
2          Greece        3
3       Australia        2
4          France        2
5         Germany        2
6  United Kingdom        2
7         Belgium        1
8          Canada        1
...
```

2.4. כתבו שאילתת SQL שתוצאתה היא רשימת הספורטאיות שזכו במדליית כסף במרתון נשים, לאורך כל השנים שבהן התקיימה התחרות הזו. הרשימה צריכה להיות ממוינת כרונולוגית:

```
           full_name year
1        Grete Waitz 1984
2        Lisa Martin 1988
3       Yuko Arimori 1992
4 Valentina Yegorova 1996
5        Lidia Simon 2000
6  Catherine Ndereba 2004
7  Catherine Ndereba 2008
```

2.5. כתבו שאילתת SQL שתוצאתה היא רשימת כל הספורטאים המקסיקניים שזכו במדליית זהב אולימפית בין השנים 1920 ו-1980, בצורה הבאה:

```
         full_name         sport year               location
1  Daniel Bautista Track & Field 1976       Montreal, Canada
2  Ricardo Delgado        Boxing 1968    Mexico City, Mexico
3       Felipe Moz      Swimming 1968    Mexico City, Mexico
4   Antonio Roldan        Boxing 1968    Mexico City, Mexico
5  Joaquim Capilla        Diving 1956   Melbourne, Australia
6 Humberto Mariles    Equestrian 1948 London, United Kingdom
7       Rubn Uriza    Equestrian 1948 London, United Kingdom
8    Alberto Valds    Equestrian 1948 London, United Kingdom
```

העמודה location מכילה את העיר והמדינה שבה נערכה האולימפיאדה. שימו לב ש-Humberto Mariles מופיע ברשימה רק פעם אחת, למרות שהוא זכה בשתי מדליות זהב ב-1948.

2.6. כתבו שאילתת SQL המוצאת את הספורטאי שזכה במספר המדליות הכולל הגבוה ביותר:

```
        full_name top_no_medals
1 Larisa Latynina            18
```

(מייקל פלפס שבר את השיא הזה באולימפיאדת 2012, אבל הנתונים שבאתר הם כאמור לא מעודכנים.)

2.7. כתבו שאילתת SQL המוצאת את הספורטאי שזכה במספר מדליות הזהב הכולל הגבוה ביותר, ואת המספר הנ"ל:

```
       full_name no_gold_medals
1 Michael Phelps             14
```

2.8. כתבו שאילתת SQL המוצאת את הספורטאי שזכה במספר מדליות הזהב הכולל הגבוה ביותר בשנה אחת, את השנה שבה הדבר קרה, ואת המספר הנ"ל:

```
       full_name year no_gold_medals
1 Michael Phelps 2008              8
```

מה להגיש

קובץ בשם project-2.R שמכיל את הקוד שכתבתם לכל אחד מהסעיפים, בפורמט שתואר בחלק הקודם.

3. עיבוד נתונים (15 נקודות)

3.1. כתבו פונקציה בשם list.medals המקבלת כארגומנט ראשון קוד של מדינה, כארגומנט שני שנת התחלה וכארגומנט שלישי שנת סיום, ומחזירה מסגרת נתונים של מספר המדליות מכל סוג שהמדינה זכתה בין שנת ההתחלה לשנת הסיום (כולל):

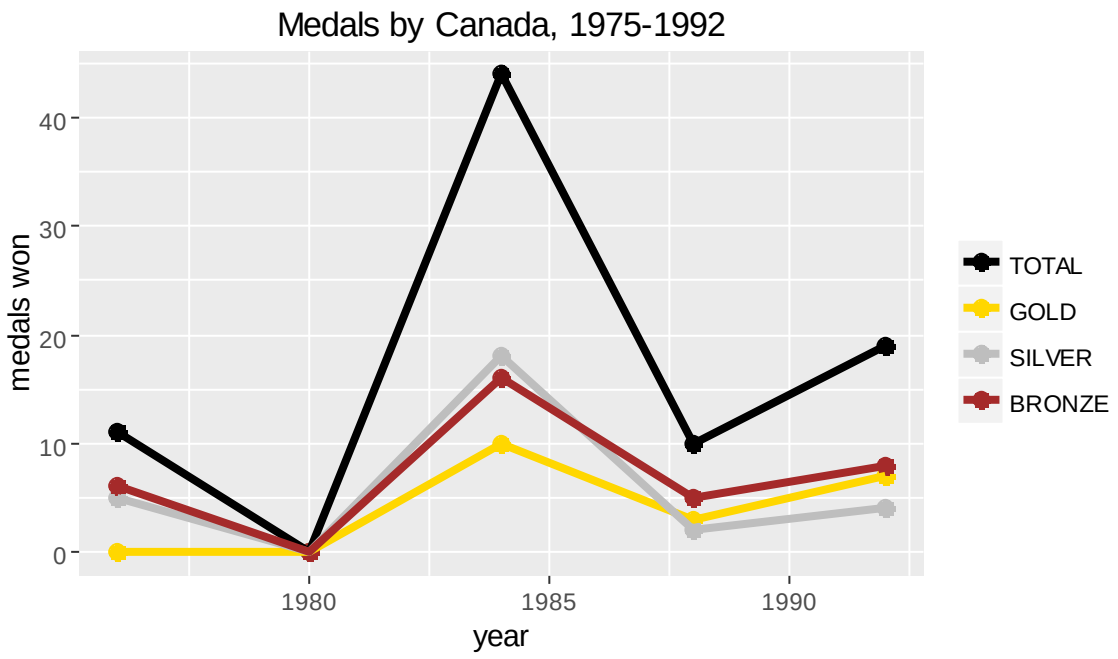
```
> list.medals("CAN", 1975, 1992)
  year GOLD SILVER BRONZE
1 1976    0      5      6
2 1980    0      0      0
3 1984   10     18     16
4 1988    3      2      5
5 1992    7      4      8

> list.medals("ARG", 2008, 2008)
  year GOLD SILVER BRONZE
1 2008    2      0      4
```

שימו לב ששנת ההתחלה / סיום לא חייבת להיות שנה שבה התקיימה אולימפיאדה. בתחרויות קבוצתיות (כגון כדורסל או מירוץ שליחים) צריך לספור רק מדליה אחת למדינה על כל זכייה, ולא מדליה נפרדת לכל ספורטאי שהשתתף בקבוצה.

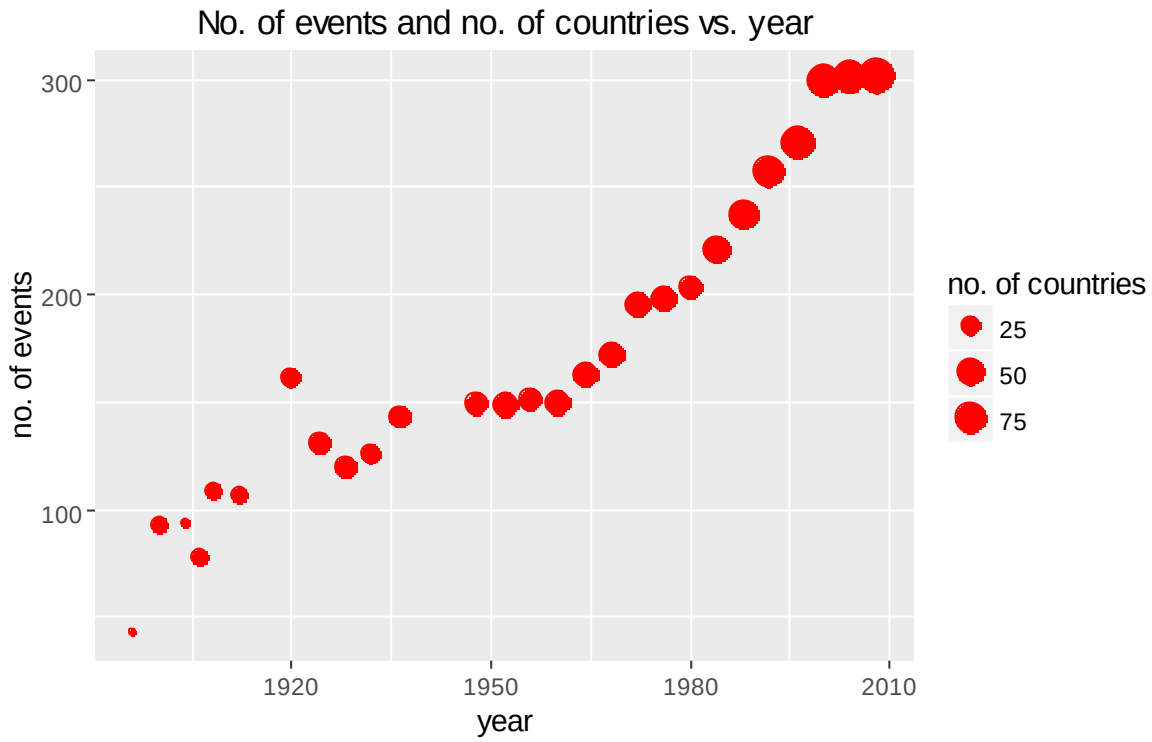
טיפים: כדי לספור רק מדליה אחת למדינה על כל זכייה, ניתן להיעזר בפונקציה duplicate. בשאלה זו (וגם בהמשך) מומלץ להשתמש בפונקציה merge, שהיא המקבילה של R ל-JOIN.

3.2. כתבו פונקציה בשם plot.medals המקבלת את אותם ארגומנטים כמו list.medals, ומייצרת גרף ggplot של נתוני הזכייה, הכולל עקומה נפרדת לכל סוג מדליה (בצבע מתאים) וכן עקומה שחורה של סה"כ מדליות. למשל, הפקודה `plot.medals("CAN", 1975, 1992)` תיצור את הגרף הבא:



שימו לב (כאן ובהמשך) לכל פרטי הגרף: הכותרת הראשית, הכיתוב על הצירים, המקרא בצד ימין, וכו'.

3.3. כתבו סקריפט המייצר גרף ggplot, המתאר את מספר התחרויות שהתקיימו באולימפיאדה כפונקציה של השנה, כאשר הגודל של כל נקודה מתאר את מספר המדינות שהשתתפו באולימפיאדה באותה השנה:



מה להגיש

קובץ בשם project-3.R שמכיל את הקוד שכתבתם (בפורמט שתואר לעיל), קובץ תמונה בשם figure-3.2.png המכיל את הגרף מסעיף 3.2 ושנוצר באמצעות הקוד שכתבתם, וקובץ תמונה דומה נוסף בשם figure-3.3.png עבור הגרף מסעיף 3.3.

4. חיזוי סטטיסטי (15 נקודות)

בחלק זה ננסה לחזות את מספר המדליות הכולל של מדינה באולימפיאדת 2008. נשתמש במודל פשוט שבו המספר הנ״ל הוא פונקציה לינארית של מספר המדליות בהן המדינה זכתה ב-k האולימפיאדות שלפני 2008.

נסמן ב-Y_c את מספר המדליות הכולל שמדינה c זכתה באולימפיאדת 2008, וב- $X_{c,i}$ את מספר המדליות שמדינה c זכתה i אולימפיאדות לפני אולימפיאדת 2008 (למשל, $X_{c,2}$ מציין את מספר המדליות שמדינה c זכתה 2 אולימפיאדות לפני 2008, כלומר באולימפיאדת 2000). המודל הוא:

$$Y_c \approx \beta_0 + \beta_1 X_{c,1} + \beta_1 X_{c,2} + \ldots + \beta_k X_{c,k}$$

4.1. כתבו פונקציה בשם estimate.b המקבלת כארגומנט ראשון וקטור של קודי מדינות וכארגומנט שני את k, ומחזירה וקטור של אמדים ל- $\beta_0, \beta_1, \ldots, \beta_k$ (סה״כ $k + 1$ ערכים). הפונקציה צריכה לאמוד את ה-β בשיטת הריבועים הפחותים, באמצעות הפונקציה lm. למשל,

```
> estimate.b(c("AUS", "CHN", "ESP", "FRA", "GBR"), 2)
        b0         b1         b2
-18.056388   3.065643  -1.296149
```

4.2. בהינתן אומדן לפרמטרים $\beta_0, \beta_1, \ldots, \beta_k$, ניתן לחזות את מספר המדליות של מדינה כלשהי באולימפיאדת 2008, בהתבסס על מספר המדליות שבהן היא זכתה ב-k האולימפיאדות שלפני 2008: מציבים את הפרמטרים ואת נתוני הזכיות באגף ימין של המשוואה שלמעלה, ומקבלים את החיזוי ל-2008.

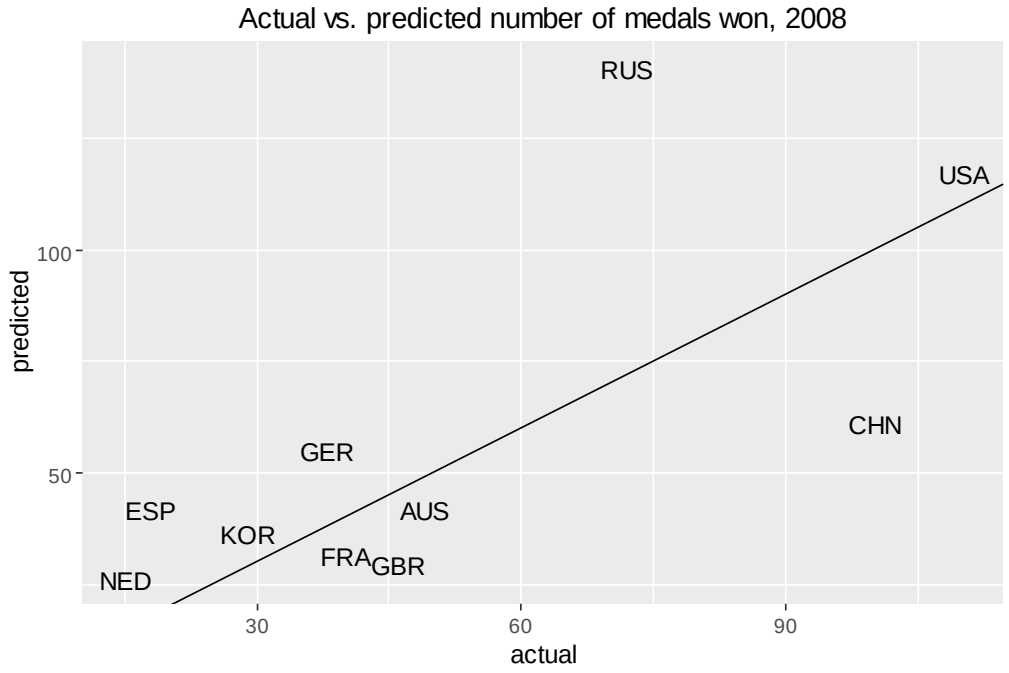
בשאלה זו נעבוד עם נתונים של 10 מדינות בלבד: אוסטרליה, סין, ספרד, צרפת, בריטניה, גרמניה, דרום קוריאה, הולנד, רוסיה וארה״ב. נשתמש בגישה שנקראת leave one out, באופן הבא: בהתחלה נשתמש בנתוני הזכייה

של 9 המדינות **פרט לאוסטרליה** (מספרי המדליות שהמדינות זכו באולימפיאדת 2008 וב-k האולימפיאדות שלפניה) כדי לקבל אומדן לפרמטרים, ואז באמצעות ההצבה שתוארה לעיל, נקבל חיזוי ל-2008 לאוסטרליה. אח״כ נעשה אותו דבר עבור סין, וכו׳. כך נקבל תחזית עבור כל אחת מ-10 המדינות.

כתבו קוד המייצר שני וקטורים: אחד בשם actual.2008, המכיל את מספר המדליות הכולל האמיתי של כל אחת מ-10 המדינות בשנת 2008, ווקטור שני בשם predicted.2008, המכיל את החיזוי שחושב באופן שתואר לעיל, עבור $k = 3$.

```
> actual.2008
AUS CHN ESP FRA GBR GER KOR NED RUS USA
 49 100  18  40  46  38  29  15  72 110
> predicted.2008
      AUS       CHN       ESP       FRA       GBR       GER       KOR
 41.84182  61.22915  42.14818  31.78361  29.90998  55.33730  36.72603
      NED       RUS       USA
 26.42275 141.13368 117.28403
```

4.3. הכינו גרף ggplot המשתמש בשני הווקטורים מהסעיף הקודם, ומראה את מספר המדליות הנחזה מול מספר המדליות האמיתי לכל מדינה, באופן הבא:



הישר המופיע בתמונה הוא של הפונקציה $y = x$, כלומר ישר שעובר דרך ראשית הצירים ושיפועו הוא 1. כשנקודה נמצאת על הקו, החיזוי הוא מושלם, וככל שהיא מרוחקת ממנו יותר, כך החיזוי פחות טוב.

מה להגיש

קובץ בשם project-4.R שמכיל את הקוד שכתבתם (בפורמט שתואר לעיל), וקובץ תמונה בשם figure-4.3.png המכיל את הגרף מסעיף 4.3 ושנוצר באמצעות הקוד שכתבתם.

5. Shiny (20 נקודות)

"shiny" היא חבילת R שמאפשרת ליצור עמודי אינטרנט אינטראקטיביים, המריצים קוד R. קראו את המדריך של החבילה, או צפו בסרטון ההדרכה, בכתובת http://shiny.rstudio.com/tutorial.

בחלק זה עליכם ליצור על המחשב שלכם אפליקציית shiny, זהה לזו שב- yuvaln.shinyapps.io/olympics.

בשביל הפתרון, מומלץ להגדיר ולהשתמש בפונקציה בשם plot.medals.2. פונקציה זו תהיה הרחבה של הפונקציה plot.medals שכתבתם בחלק 3 של העבודה, ויהיו לה ארגומנטים נוספים שיאפשרו לקבוע איזה מבין המדליות יוצגו בגרף שהיא יוצרת.

מה להגיש

קובץ אפליקציה בשם app.R הבנוי בצורה הבאה:

```
library(shiny)
source("helpers.R")

ui <- fluidPage(
  הקוד שלכם
)

server <- function(input, output){
  הקוד שלכם
}

shinyApp(ui = ui, server = server)
```

וקובץ בשם helpers.R המכיל פונקציות עזר, משתנים, וכו' הדרושים לקובץ האפליקציה.