

# פרויקט סימולציה

הכנות המשלחת הישראלית בהתעמלות קרקע לאולימפיאדה



סמסטר א' תשפ"ב

**קבוצה 23**

318356995

207133109

318359726

## תוכן עניינים

3.	תקציר	3
4.	מבוא	3
4.1.	תיאור המערכת הנחקרת	3
4.2.	מטרות	4
5.	מודל סימולציה	4
5.1.	תיאור המודל	4
5.2.	הנחות	6
6.	ניתוח המצב הקיים והצעת חלופות לשיפור	7
6.1.	בחירת מדדים לניתוח	7
6.2.	סוג המערכת	8
6.3.	ניתוח מצב קיים ביחס למדדים ולמטרות	8
6.4.	תיאור החלופות	9
7.	מסקנות	11
8.	נספחים	12
4.1.1.	משאבים וזמני שירות	12
4.1.2.	חישוב רמת עייפות	14
5.1.2.	תכונות הישויות	23
5.1.3.	המסלולים הראשיים	24
5.1.4.	מסלולי האימון	25
5.1.5.	מסלולי המכשירים	25
5.1.6.	מסלולי הוידאו	26
5.1.7.	מסלולי התזונאים	27
5.1.8.	מסלול הפיזיותרפיסט	27
6.1.6.	חישוב מדד אחוז מסיימים	28
6.1.2.	חישוב מדד זמן שהייה ממוצע	28
6.2.1.	חישוב הדיוק היחסי	29
6.3.1.	תוצאות מדדי המצב הקיים עבור 15 ההרצות	29
6.4.1.	תוצאות מדדי החלופות	30
6.5.1.	טיב התאמה להתפלגות נורמלית עבור תוצאות מדד זמן שהייה	32
7.	קוד הסימולציה:	35

### 3. תקציר

בפרויקט זה התבקשנו לחקור את פעילות מרכז הספורט הלאומי המכונה גם "הבניין האולימפי" בהכנתו לקראת שליחת הנבחרת הישראלית בהתעמלות קרקע לאולימפיאדה. לרשותנו תקציב על סך 50,000 ₪ למטרת שיפור המצב הקיים והגדלת הסיכוי לזכות במדליה בענף זה באולימפיאדות הבאות.

לאחר מידול המצב הקיים, בחרנו לשפר את מדדי אחוז מסיימי האימון ביום וזמן השהייה ממוצע באולם האימונים. בחקר המצב הקיים נמצא כי אחוז מסיימי האימון היומי הוא 41.15% בממוצע וזמן השהייה הממוצע הוא 423.541 דקות, כלומר 7.059 שעות.

בשלב זה, בחרנו להציע שתי חלופות אשר לדעתנו יוכלו לסייע לשפר מדדים אלה. האחת, הוספת עמדת ריענון בתאי ההלבשה והוספת שני מכשירי התעמלות קרקע. השנייה, ביקור יו"ר הוועד האולימפי באולם, הוספת עמדת ריענון בתאי ההלבשה והוספת מכשירי התעמלות קרקע. בין החלופות השונו במבחן t מזווג בשיטת CRN.

מתוצאות ההשוואה גילינו ששתי החלופות אכן משפרות את המצב הקיים, ובין שתיהן קיימת אדישות בשני המדדים. לכן, הצענו לוועד האולימפי לקחת את החלופה הראשונה מאחר והיא זולה יותר ומניבה ומבחינה סטטיסטית טובה כמו החלופה השנייה.

### 4. מבוא

#### 4.1. תיאור המערכת הנחקרת

בפרויקט זה ננתח את הפעילות היומית של הבניין האולימפי.

- אולם האימונים פועל על בסיס יומי בין השעות 06:00-20:00, ואילו מגיעים המתעמלים והמתעמלות בין השעות 06:00-15:00.
- באולם קיימים שני מתחמי הלבשה ומקלחות, מכשירי אימון, חמישה חדרים לעשרת בוחני הוידאו, שני תזונאים וחמישה פיזיותרפיסטים )
- [נספח 4.1.1 – משאבים וזמני שירות](#).
- במהלך יום האימון, המתעמלים נדרשים להתאמן על כלל מכשירי האימון המתאימים לפי מינם, אלא אם כן צברו במהלך האימון רמת עייפות גבוהה מידי.
- בכל יום בשעה 13:00, כלל אנשי המקצוע יוצאים להפסקה.

מתעמל המגיע אל אולם האימון נכנס אל תאי ההלבשה ומתארגן לקראת תחילת יום האימון. בסיום ההתארגנות, הולך אל מכשירי האימון המתאימים לפי מינו ומתחיל את יום האימונים. בסיום כל תרגיל פונה אל חדר הוידאו הקבוע שלו, שם הוא צופה יחד עם בוחן הוידאו בתרגיל שביצע לצורך הפקת לקחים ושיפור. במידה והמתעמל מסיים אימון במכשיר ובוחן הוידאו טרם הגיע, הוא ממשיך להתאמן ועם הגיעו צופה בכל הסרטונים ברציפות. לאחר סיום האימון במכשירים, חלק מהמתעמלים מעוניינים להשתתף בהרצאה תזונה המתקיימת בכל שעה עגולה או עם הגיעם של 10 מתעמלים, ובסוף היום כלל המתעמלים הולכים לטיפול אצל פיזיותרפיסט. מתעמלים אשר צברו רמת עייפות גבוהה במהלך האימון היומי שלהם מקבלים עדיפות בכניסה אל הפיזיותרפיסט (נספח 4.1.2 – חישוב רמת עייפות). בתום האימון היומי, המתעמלים ניגשים אל המקלחות ועוזבים את אולם האימונים.

## 4.2. מטרות

מטרת הועד האולימפי הינה להגדיל את סיכויי המשלחת הישראלית לזכות במדליה בענף התעמלות הקרקע באולימפיאדות הבאות. על כן, מטרתנו בפרויקט זה הינה לייעל את המצב הקיים ובכך לאפשר למתעמלים להתאמן בצורה הטובה ביותר אשר תביא אותם אל ההישגים הגבוהים ביותר. לצורך כך, נרצה שאחוז המתעמלים אשר מצליחים להשלים את האימון היומי שלהם יהיה גדול ככל שניתן, ושזמן השחייה שלהם באולם האימונים יהיה נמוך ככל הניתן וזאת כדי לאפשר למתעמלים רבים להתאמן.

מטרת ביצוע הסימולציה היא לזהות את הנקודות בהן נוכל לשפר מדדים אלו, וזאת באמצעות בחינת מספר חלופות העומדות לרשותנו תחת התקציב שקיבלנו.

## 5. מודל סימולציה

### 5.1. תיאור המודל

לצורך בניית הסימולציה השתמשנו בחבילת Simmer בתוכנת RStudio. על מנת למדל כראוי את אולם האימונים, יצרנו ישויות, משאבים ומסלולים.

עם תחילת הסימולציה, מתחילות להיווצר ישויות המתעמלים והמתעמלות בהתאם לקצב ההגעה שלהם )

נספח 5.1.1 – טיב התאמה לקצב הגעת הישויות). מיד עם היווצרותן, המתעמלים והמתעמלות נשלחים אל מסלול האימון היומי המתאים לפי מינם. מסלולי האימון זהים ברובם אך בהתאם למין המתעמל קיימות הפניות אל מכשירי אימון ומלתחות שונים. לצורך הפשטות, נתאר את סדר המסלולים ותוכנם עבור נשים וגברים כאחד.

#### המסלולים הראשיים (womenTraj, menTraj), (נספח 5.1.3 – המסלולים הראשיים):

המסלול המרכזי אליו מגיעים המתעמלים עם הגעתם לסימולציה. במסלול זה מוגדרות ומאותחלות התכונות הראשוניות שלהם (נספח 5.1.2 – תכונות הישויות) 0 והם ניגשים אל המלתחות. לאחר מכן, מתחילים ביום האימון ונשלחים ממסלול זה אל מסלולי האימון במכשירים. בכל פעם, ישלחו המתעמלים למכשיר אחר שטרם היו בו. בסיום האימון, הם חוזרים אל מסלול זה ומנסים לתפוס את הדלת לחדרי הוידאו. אם הצליחו – עוברים למסלול הוידאו. במידה ולא הצליחו – נשלחים שוב ממסלול זה להתאמן ומנסים לתפוס את הדלת בשנית, עד אשר מסיימים להתאמן או שרמת העייפות שלהם מגיעה לרמה המרבית. בסיום האימון, חלקם נשלחים להרצאת התזזואי, וכלל המתעמלים נשלחים לטיפול פיזיותרפיסט ולמקלחות.

#### מסלולי האימון (womenTraining, menTraining), (נספח 5.1.4 – מסלולי האימון):

המסלול אליו נשלחים המתעמלים על מנת להתאמן. ממסלול זה הם מתפצלים למסלולים הרלוונטיים עבור כל אחד מהמכשירים, ואם רמת העייפות שלהם הגיעה לרמה המרבית, הם מקבלים עדיפות אצל הפיזיותרפיסט.

#### מסלולי המכשירים (נספח 5.1.5 – מסלולי המכשירים):

למסלולים אלו מגיעים המתעמלים על מנת להתאמן במכשירים. לאחר האימון במכשיר, תכונת המתעלמים "האם התאמנו במכשיר" משתנה ל-1 (כלומר חיווי שאכן התאמנו במכשיר), ותכונות ה-tiredness, numOfMachines, count- שלהם מתעדכנות בהתאמה. מסלולי המכשירים הקיימים: ground, steppedParallels, beam1, beam2, jumps1, jumps2, parallels, rings, tension, adjacentHorse.

מסלולי הוידאו (videoTrajWomen, videoTrajMen),

נספח 5.1.6 – מסלולי הוידאו):

המסלולים אליהם נשלחים המתעמלים אשר הצליחו לתפוס את הדלת של חדרי הוידאו. במסלול זה הם נכנסים לחדר הוידאו אליו הם משויכים וצופים ברצף בסרטוני הוידאו אשר עליהם לראות. משם, כל עוד הם לא סיימו להתאמן ורמת העייפות שלהם לא הגיעה לרמה המרבית, הם נשלחים שוב להתאמן. בסיום האימון, חלקם נשלחים להרצאת התזונאי, וכלל המתעמלים נשלחים לטיפול פיזיותרפיסט ולמקלחות.

מסלולי התזונאים (nutritionTrajWomen, nutritionTrajMen), נספח 5.1.7 – מסלולי התזונאים):

המסלולים אליהם מגיעים המתעמלים על מנת לצפות בהרצאת התזונאי. במסלול זה, הם מתאגדים בקבוצה וצופים בהרצאה אשר מתקיימת בכל שעה עגולה, או ברגע שמגיעים 10 מתעמלים.

מסלול הפיזיותרפיסטים (physiotherapyTraj), נספח 5.1.8 – מסלול הפיזיותרפיסט):

המסלול אליו מגיעים המתעמלים והמתעמלות על מנת לקבל טיפול מהפיזיותרפיסטים. במסלול זה הם מקבלים שירות לפי העדיפות שלהם (רמת העייפות).

## 5.2. הנחות

1. מתעמלים אשר סיימו את האימון היומי שלהם טרם הגעת בוחני הוידאו, עוזבים את אולם האימונים מבלי לצפות בתרגילים שביצעו.
2. אולם האימונים גדול מאוד ועל כן אורך התור של כלל המשאבים אינו מוגבל וקיים מקום עבור כל המתעמלים.
3. יחידת זמן הסימולציה שנקבעה היא דקה אחת.
4. זמן השהייה של מתעמלים באולם האימונים יימדד החל משעת כניסתם לסימולציה ועד עזיבתם. מתעמלים אשר לא הספיקו לסיים את הסימולציה לא יכללו בחישוב זמני השהייה מאחר והם לא נחשבים כמתעמלים אשר סיימו את יום האימון בצורה תקינה ולכן עלולים לגרום לתוצאת המדד להיות שונה ממה שהיא בפועל.
5. כלל זמני השירות הנתונים אשר התפלגותם נורמלית חושבו כהתפלגות נורמלית קטומה מאחר ולא יתכנו זמני שירות שליליים.
6. בעת שעת ההפסקה, אנשי מקצוע אשר נמצאים במהלך טיפול במתעמלים ישלימו אותו, ורק לאחר מכן יצאו להפסקה. זמן ההפסקה היומי נקבע בתחילת כל יום ובסיום ההפסקה על כל אנשי המקצוע לשוב אל עבודתם, ועל כן לאנשי מקצוע אשר יצאו להפסקה בשעה מאוחרת יותר, יתקצר בהתאם זמן ההפסקה שלהם.
7. אנשי המקצוע אשר מסיימים את המשמרת שלהם בשעה אחרת משעת סיום יום העבודה, יסיימו לטפל במתעמל אשר נמצא אצלם ורק לאחר מכן יסיימו את עבודתם.

## 6. ניתוח המצב הקיים והצעת חלופות לשיפור

### 6.1. בחירת מדדים לניתוח

בהתאם למטרות הפרויקט שתיארנו לעיל, נבחרו מדדי הניתוח הבאים:

1. אחוז מסיימי האימון היומי – לצורך הגעה להישגים ספורטיביים גבוהים, על המתעמלים להשלים כמה שיותר אימונים אשר יעזרו להם להשתפר ובכך להגדיל את סיכוייהם לזכות במדליה. מתוך כלל המתעמלים המגיעים אל אולם האימונים בכל יום, נרצה שאחוז גבוה ככל הניתן ישלים את האימון. ככל שאחוז זה יהיה גבוה יותר, יותר מתעמלים יגיעו לרמת מיומנות גבוהה שתוכל להקנות להם את המדליה

האולימפית. מסיימי האימון הוגדרו להיות כלל המתעמלים אשר השלימו את כמות המכשירים שנדרשו בהתאם למינם, ובנוסף עברו טיפול פיזיותרפיסטי החשוב על מנת ששריריהם לא יפגעו. לכן, ערכו של אחוז מסיימי האימון היומי הינו מספר מסיימי האימון מתוך כלל המתעמלים שהגיעו לאולם האימונים. (נספח 6.1.6 – חישוב מדד אחוז מסיימים)

זמן השהייה הממוצע באולם האימונים – על מנת להגיע לרמת התמדה גבוהה אצל המתעמלים, נרצה שחויית האימון שלהם תהיה טובה ככל שניתן. זמן שהייה ממוצע יכול להעיד על זמני ההמתנה הכוללים למשאבים וכאשר מדד זה נמוך, ניתן להסיק יעילות בכל הנוגע לטיפול במתעמלים. בנוסף, זמן שהייה נמוך באולם האימונים מפנה זמן גם לשינה ושחרור השרירים החיוניים מאוד למתעמלים האולימפיים. ערכו של מדד זה חושב כממוצע ההפרשים בין זמני ההתחלה לזמני הסיום של המתעמלים בסימולציה. מתעמלים אשר לא סיימו את האימון היומי לא נלקחו בחשבון בחישוב ערכו של מדד זה )

2. נספח 6.1.2 – חישוב מדד זמן שהייה ממוצע).

## 6.2. סוג המערכת

אולם האימונים של הוועד האולימפי הינו מערכת מסתיימת, וזאת מאחר שהוגדר זמן סיום היום, ומתעמלים אשר לא הספיקו להשלים את האימון שלהם עד שעה זו נשלחים לביתם ועוזבים את המערכת. מאחר וזוהי מערכת מסתיימת, אין צורך בזמן חימום. מספר הריצות הראשוני נקבע להיות  $n_0 = 15$ , ומשך כל ריצה הוא 840 דקות (14 שעות). עבור כל אחד מהמדדים שבחרנו, חישבנו את ממוצע, סטיית התקן וחצי רווח הסמך עבור  $n_0$  ההרצות הראשונות.

בכדי לבדוק מהו מספר החזרות הנדרש עבור המצב הקיים והחלופות שבחרנו, בדקנו עבור

$$\frac{\delta(n, \alpha)}{\bar{x}} \leq \frac{\gamma}{1+\gamma}$$

$$\frac{\gamma}{1+\gamma} = 0.0654 \text{ בחרנו ברמת דיוק יחסי } \gamma = 0.07, \text{ אשר עבורה מתקבל החסם}$$

רמת המובהקות שנבחרה היא  $\alpha = 0.1$ , ולכן רמת המובהקות עבור כל אחד מהמדדים היא

$$\alpha_i = \frac{\alpha}{\text{Num of measurments}} = 0.05 \text{ על כן, רמת הבטחון היא } 1 - \alpha_i = 0.95$$

כפי שניתן לראות בתוצאות הדיוק היחסי שהתקבלו (נספח 6.2.1 – חישוב הדיוק היחסי), אין צורך בהרצת חזרות נוספות מאחר ואנו עומדות ברמת הדיוק שהגדרנו ולא נמצאו חריגות.



במידה ולא היינו עומדות ברמת הדיוק הנדרשת, היינו מחשבות את מספר החזרות הנדרש

$$n = n_0 \left( \frac{\delta_0}{\frac{\gamma}{\bar{x}^*_{1+\gamma}}} \right)^2 \text{ באמצעות דיוק גס לפי הנוסחה:}$$

### 6.3. ניתוח מצב קיים ביחס למדדים ולמטרות

תוצאות המדדים עבור המצב הקיים (נספח 6.3.1 – תוצאות מדדי המצב הקיים עבור 15 ההרצות):

זמן שהייה	אחוז מסיימים	ממוצע
423.541	0.4115	ממוצע
36.896	0.0286	סטיית תקן

ניתן לראות שאחוז מסיימי האימון היומי עומד על 0.4115, כלומר רוב המתעמלים שמגיעים לאולם האימונים לא מצליחים להשלים את האימון. לדעתנו מדובר בנתון נמוך יחסית והיינו רוצות שמספר זה יהיה גבוה יותר על מנת למקסם את סיכויי ההצלחה של המתעמלים באולימפיאדה. לצורך כך, נרצה לבחון חלופות אשר יוכלו להגדיל את ערכו של מדד זה. ככל שאחוז המסיימים יהיה גבוה יותר, כך אנו משערות שסיכויי ההצלחה של המתעמלים יהיו גבוהים יותר מאחר והם יבואו לתחרות יותר מוכנים.

בנוסף, ממוצע זמן השהייה במצב הקיים עומד על 423.541 דקות, כלומר 7.059 שעות. נתון זה נשמע גבוה וייתכן שהוא נובע מעומסים בתור אצל חלק מהמשאבים. אנו סבורות כי זהו נתון גבוה יחסית ליום האימון ונרצה לבחון חלופות שיוכלו למזער נתון זה. ככל שזמן השהייה של המתעמלים יהיה נמוך יותר, אנו מאמינות שיעילות הטיפול במתעמלים תהיה טובה יותר ויתפנה להם זמן רב שנחוץ למנוחה החיונית לגוף לצורך הגעה להישגים אולימפיים.

### 6.4. תיאור החלופות

חלופה 1:

שיפור	תיאור	עלות
עמדת קפה וריענון בתאי ההלבשה	הורדת תוחלת האימון במכשירים ב- 25%	14,000 ₪
הוספת שני מכשירי התעמלות	הוספת שני מכשירי התעמלות קרקע	32,000 ₪
סה"כ		46,000 ₪

## חלופה 2:

שיפור	תיאור	עלות
איפוס רמת העייפות של המתעמלים	ביקור יו"ר הועד האולימפי באולם	20,000 ₪
עמדת קפה וריענון בתאי ההלבשה	הורדת תוחלת האימון במכשירים ב- 25%	14,000 ₪
הוספת מכשיר התעמלות	הוספת מכשיר התעמלות קרקע	16,000 ₪
סה"כ		50,000 ₪

להלן תוצאות המדדים לאחר מימוש השיפורים בשתי החלופות (נספח 6.4.1 – תוצאות מדדי  
החלופות):

אחוז מסיימים			ממוצע זמן שהייה			
	מצב קיים	חלופה 1	חלופה 2	מצב קיים	חלופה 1	חלופה 2
ממוצע	0.4115	0.7103	0.6884	423.541	385.3043	389.1026
סטיית תקן	0.0286	0.0585	0.07202	36.896	25.3707	21.1624

## 6.5. השוואה סטטיסטית בין המצב הקיים לחלופות המוצעות

על מנת לבחון את המצב הקיים ואת שתי החלופות שהצענו ביחס לשני הממדים שבחרנו, נבצע מבחן  $t$  מזווג עבור 6 ההשוואות הקיימות, כל אחת מהן ברמת מובהקות

$$\alpha_i = \frac{\alpha}{\text{Num of comparisons}} = \frac{0.1}{6} = 0.01666$$

ריצות מקבילות עבור כל אחת מהחלופות מאחר וה-*seed*. בכל זוג ריצות מקבילות זהה.

על מנת לבצע את ההשוואה יש להניח תלות בין ריצות מקבילות בין הסדרות, חוסר תלות בין ריצות בכל חלופה בפני עצמה, שוויון שונויות ושהתוצאות מתפלגות נורמלית (נספח 6.5.1 – טיב התאמה להתפלגות נורמלית עבור תוצאות מדד זמן שהייה).

#### תוצאות ההשוואה הסטטיסטית – מבחן $t$ מזוג:

ממד	השוואה	גבול עליון רווח סמך	גבול תחתון רווח סמך	תוצאות המבחן	סיכום עבור המדד
מדד אחוז מסיימים	מצב קיים וחלופה 1	-0.25139	-0.34605	חלופה 1	חלופה 1/
	מצב קיים וחלופה 2	-0.22381	-0.32982	חלופה 2	חלופה 2
	חלופה 1 וחלופה 2	0.092272	-0.04846	אדישות	
מדד זמן שהייה	מצב קיים וחלופה 1	70.687427	5.786439	חלופה 1	חלופה 1/
	מצב קיים וחלופה 2	65.90431	2.97296	חלופה 2	חלופה 2
	חלופה 1 וחלופה 2	19.97084	-27.5674	אדישות	

מהטבלה ניתן לראות כי עבור שני המדדים שבחנו, שתי החלופות שהצענו עדיפות על המצב הקיים, אך בין השתיים קיימת אדישות. שתי החלופות אכן שיפרו רבות את המדדים, ומידת השיפור בשתייה הייתה דומה, לכן התקבלה אדישות בהשוואה ביניהן וניתן להסיק ששתי החלופות טובות באותה המידה עבור מדדים אלה ברמת המובהקות של האלפא שצינו.

## 7. מסקנות

מטרתנו בפרויקט זה הייתה למצוא חלופות אשר יוכלו לייעל את התנהלות הוועד האולימפי הישראלי ולשפר את סיכויי המשלחת הישראלית לזכות במדליות בענף התעלמות הקרקע. התוצאות שקיבלנו השוואה הסטטיסטית מראות כי שתי החלופות שהצגנו בפני הוועד האולימפי אכן משפרות את שני המדדים שהגדרנו בצורה ניכרת, ושתי החלופות הללו עדיפות על המצב הקיים בשני המדדים. בין שתי החלופות קיימת אדישות, ניתן להסיק שאדישות זו מתקיימת מאחר ושתי החלופות משפרות את המצב בצורה ניכרת, ואין הבדל ביניהן ברמת מובהקות 1.666%. מאחר ועלינו להמליץ לוועד האולימפי באילו חלופה לנקוט, נציע לו להחיל את מדיניות החלופה הראשונה – הוספת עמדת ריענון בתאי ההלבשה והוספת שני מכשירי התעמלות קרקע. הסיבה לבחירה בחלופה זו הוא עלותה הנמוכה יחסית לחלופה השנייה, ותועלתה הדומה לה.

לסיכום, ניתן לומר כי אכן הצלחנו לשפר את סיכויי המשלחת לזכות במדליה תוך הוצאת עלות נדרשת העומדת במסגרת התקציב שניתן לנו, ולכן עמדנו במטרתנו לסייע בכך לוועד האולימפי.

## 8. נספחים

### נספח 4.1.1 – משאבים וזמני שירות

- **Door** – דלת לחדרי הוידאו אשר מגבילה את כניסת המתאמנים לתורים של בוחני הוידאו טרם הגעתם בשעה 8:00. עד הגעת הבוחנים, אורך התור לדלת הוא 0 והקיבולת 0, ולאחר הגעתם אורך התור הוא אינסופי והקיבולת 10 – מספר האנשים המקסימלי שיכול להיות בכל חדרי הוידאו בו זמנית.
- **womenDressingRoom** – חדר ההלבשה של הנשים. לאורך כל הסימולציה הקיבולת 20 ואורך התור אינסופי.
- **menDressingRoom** – חדר ההלבשה של הגברים. לאורך כל הסימולציה הקיבולת 20 ואורך התור אינסופי.
- **womenShower** – מקלחות הנשים. לאורך כל הסימולציה הקיבולת היא 5 ואורך התור אינסופי.

- **menShower** – מקלחות הגברים. לאורך כל הסימולציה הקיבולת היא 5 ואורך התור אינסופי.
- **video(1-5)** – חדרי הוידאו הממוספרים בהתאמה. עבור כל אחד מהחדרים, עד השעה 8:00 הקיבולת 0 מאחר והבוחנים טרם הגיעו. החל מהשעה 8:00 הקיבולת של כל אחד מהחדרים היא 2. אורך התור עד להגעת הבוחנים הוא 0 ולאחר ההגעה אינסופי.
- **nutrition(1-2)** – שני התזונאים. עבור כל אחד מהם, עד הגעת התזונאים בשעה 8:00 הקיבולת היא 0 ולאחריה 1 (קבוצה אחת). אורך התור לאורך כל הסימולציה הוא אינסופי.
- **physiotherapist** – הפיזיותרפיסטים. עד השעה 8:00 הקיבולת היא 0 מאחר והפיזיותרפיסטים טרם הגיעו. ב-8:00 הקיבולת עולה ל-2 ובשעה 12:00 עולה ל-5. בשעה 16:00, הקיבולת יורדת ל-3. אורך התור לאורך כל הסימולציה אינסופי.
- **parallels, rings, tension, adjacentHorse, ground, jumps1, jumps2, steppedParallels, beam1, beam2** – מכשירי האימון. לאורך כל הסימולציה הקיבולת של כל אחד מהמשאבים הללו היא 2 ואורך התור אינסופי.
- עבור בוחני הוידאו, התזונאים והפיזיותרפיסטים – במהלך ההפסקה היומית, הקיבולת של המשאבים הללו מתאפסת.

#### התפלגויות זמני שירות של המשאבים:

התפלגות	פעולה
$U(3,5)$	התארגנות בתאי הלבשה (לשני המגדרים)
$U(8,14)$	זמן מקלחת (לשני המגדרים)
$Norm(5,1.7)$	זמן אימון במכשיר
$Norm(3,0.75)$	משך צפייה בווידיאו
$U(30,40)$	משך הרצאת תזונה
$Triangle(25,40,33)$	משך טיפול פיזיותרפיסט
$Norm(6,5/6)$	התפלגות הפסקה של בעלי המקצוע

נספח 4.1.2 – חישוב רמת עייפות

נפתח את אלגוריתם הדגימה של רמת העייפות באמצעות שיטת הטרנספורם ההופכי:

התפלגות רמת העייפות בסיום כל מכשיר:

$$f(x) = \begin{cases} 8x^2, & 0 \leq x \leq \frac{1}{2} \\ 2, & \frac{1}{2} \leq x \leq \frac{2}{3} \\ 6 - 6x, & \frac{2}{3} \leq x \leq 1 \end{cases}$$

הפונקציה המצטברת:

$$F(x) = \begin{cases} 0, & x \leq 0 \\ \frac{8x^3}{3}, & 0 \leq x \leq \frac{1}{2} \\ 2x - \frac{2}{3}, & \frac{1}{2} \leq x \leq \frac{2}{3} \\ -3x^2 + 6x - 2, & \frac{2}{3} \leq x \leq 1 \\ 1, & x \geq 1 \end{cases}$$

חישוב הטנספורם ההופכי:

$$\frac{8x^3}{3} = u \Rightarrow x = \sqrt[3]{\frac{3u}{8}}$$

$$2x - \frac{2}{3} = u \Rightarrow x = \frac{u + \frac{2}{3}}{2} = \frac{3x + 2}{6}$$

$$-3x^2 + 6x - 2 = u \Rightarrow x = \frac{-6 \pm \sqrt{12(1-u)}}{-6} \Rightarrow x = \frac{-6 + \sqrt{12(1-u)}}{-6}$$

בדיקות:

$$u = 0: \quad x = \sqrt[3]{\frac{3u}{8}} = 0 \quad \checkmark, \quad u = \frac{1}{3}: \quad x = \sqrt[3]{\frac{3u}{8}} = \frac{1}{2} \quad \checkmark$$

$$u = \frac{1}{3}: \quad x = \frac{u + \frac{2}{3}}{2} = \frac{1}{2} \quad \checkmark, \quad u = \frac{2}{3}: \quad x = \frac{u + \frac{2}{3}}{2} = \frac{2}{3} \quad \checkmark$$

$$u = \frac{2}{3}: \quad x = \frac{-6 + \sqrt{12(1-u)}}{-6} = \frac{2}{3} \quad \checkmark, \quad x = \frac{-6 - \sqrt{12(1-u)}}{-6} = \frac{4}{3} \quad \times$$

$$u = 1: \quad x = \frac{-6 + \sqrt{12(1-u)}}{-6} = 1 \quad \checkmark, \quad x = \frac{-6 - \sqrt{12(1-u)}}{-6} = 1 \quad \checkmark$$

## אלגוריתם הדגימה עבור פונקציית העייפות:

1. דגום  $u \sim U(0,1)$

2. אם  $u \leq \frac{1}{2}$  : החזר  $x = \sqrt[3]{\frac{3u}{8}}$

3. אם  $\frac{1}{2} \leq u \leq \frac{2}{3}$  : החזר  $x = \frac{u + \frac{2}{3}}{2}$

4. אחרת: החזר  $x = \frac{-6 + \sqrt{12(1-u)}}{-6}$

## פונקציית חישוב רמת העייפות ב-R:

```
#returns the accumulative tiredness
getTiredness<-function(){
  u<-runif(1,0,1)
  if (u<=0.5){
    return ((3*u/8)^(1/3)+get_attribute(mySimulation, "tiredness"))
  }
  if (u> 0.5 && u<= 2/3){
    return (((u+2/3)/2)+get_attribute(mySimulation, "tiredness"))
  }
  if (u>2/3){
    return (((-6+sqrt(12*(1-u)))/-6)+get_attribute(mySimulation, "tiredness"))
  }
}
```

נספח 5.1.1 – טיב התאמה לקצב הגעת הישויות

## arrival distribution from excel

318356995 , 207133109 , 318359726

12/17/2021

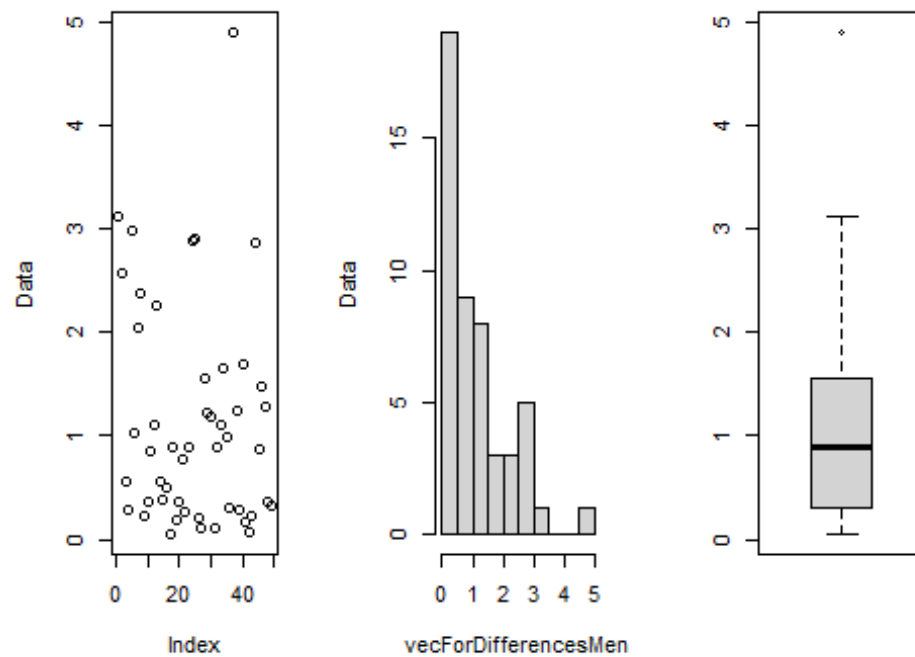
## 2. First Impressions

Issue some charts in order to estimate Which distributions we should check later on.

```
par(mfrow=c(1,3)) # defines 3 plots on the same window
plot(vecForDifferencesMen,ylab="Data") # plot (left graph)
hist(vecForDifferencesMen, ylab="Data") # histogram (right graph)
boxplot(vecForDifferencesMen)
```

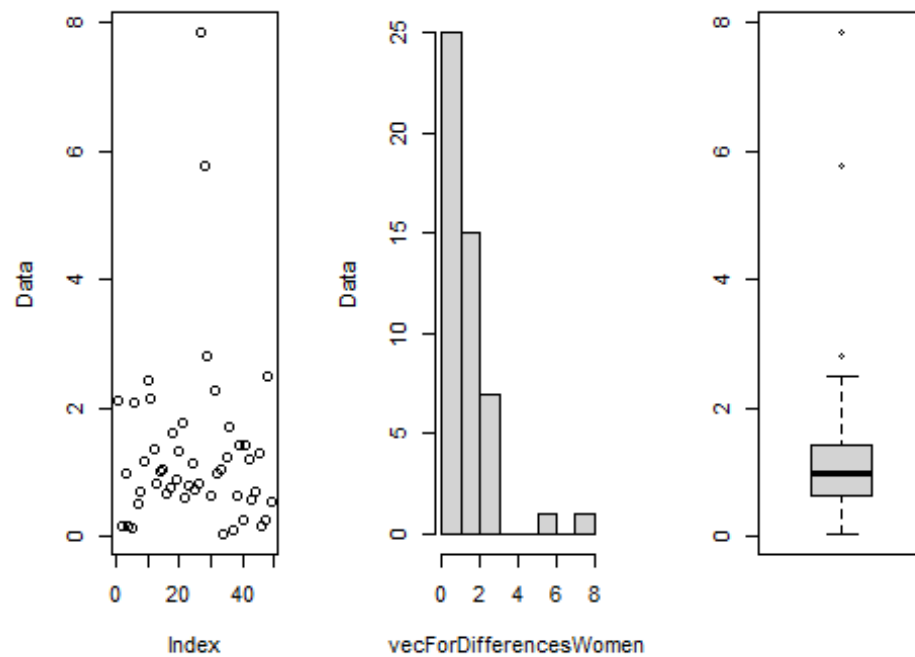


logram of vecForDifference



```
par(mfrow=c(1,3)) # defines 3 plots on the same window
plot(vecForDifferencesWomen,ylab="Data") # plot (left graph)
hist(vecForDifferencesWomen, ylab="Data") # histogram (right graph)
boxplot(vecForDifferencesWomen)
```

gram of vecForDifference



### 3. Fit a known distribution

After figuring out that the dist might be either Normal or Exp, we compared those distributions by likelihood, AIC and BIC. We found out that the exp dist gets lower AIC&BIC and higher loglikelihood. Therefore we determine that the correct dist would be exp.

```
normFit<-fitdist(vecForDifferencesMen,"norm")
expFit <- fitdist(vecForDifferencesMen,"exp") # fitting an exp distribution
summary(expFit) # summary function displays the results

## Fitting of the distribution ' exp ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## rate 0.8815592  0.1259369
## Loglikelihood: -55.17709    AIC:  112.3542    BIC:  114.246

summary(normFit)

## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## mean 1.134354  0.1496269
## sd   1.047389  0.1058018
## Loglikelihood: -71.79669    AIC:  147.5934    BIC:  151.377
## Correlation matrix:
##      mean sd
## mean    1  0
## sd      0  1

#exp dist suits better because it presents lower AIC&BIC values

normFit2<-fitdist(vecForDifferencesWomen,"norm")
expFit2 <- fitdist(vecForDifferencesWomen,"exp") # fitting an exp distribution
summary(expFit2) # summary function displays the results

## Fitting of the distribution ' exp ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## rate 0.7744995  0.1106426
## Loglikelihood: -61.52138    AIC:  125.0428    BIC:  126.9346

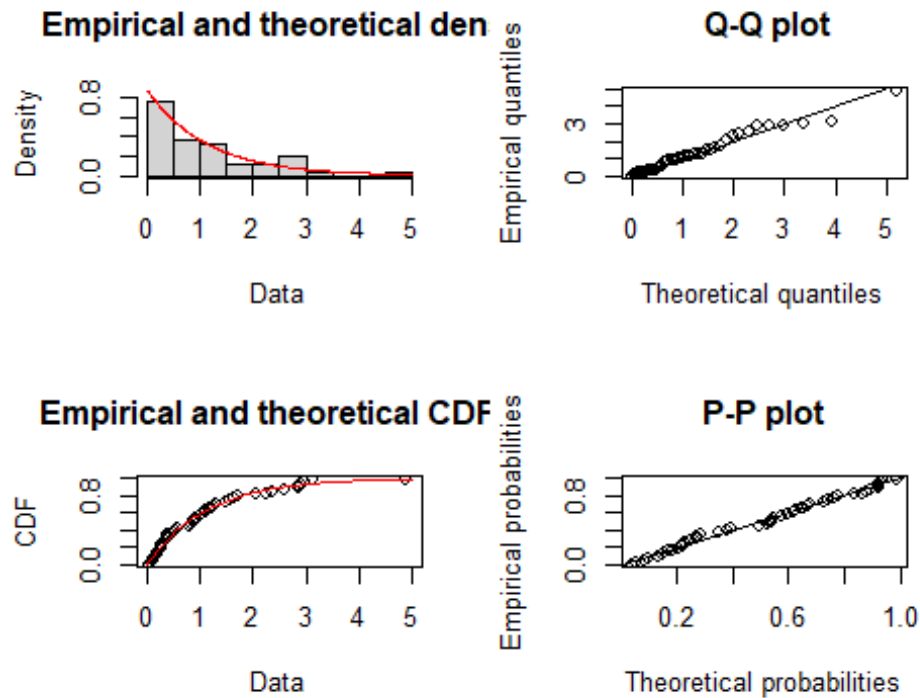
summary(normFit2)

## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## mean 1.291156  0.1912679
## sd   1.338875  0.1352465
## Loglikelihood: -83.82764    AIC:  171.6553    BIC:  175.4389
## Correlation matrix:
##      mean sd
## mean 1.000000e+00 -4.595133e-11
## sd   -4.595133e-11 1.000000e+00
```

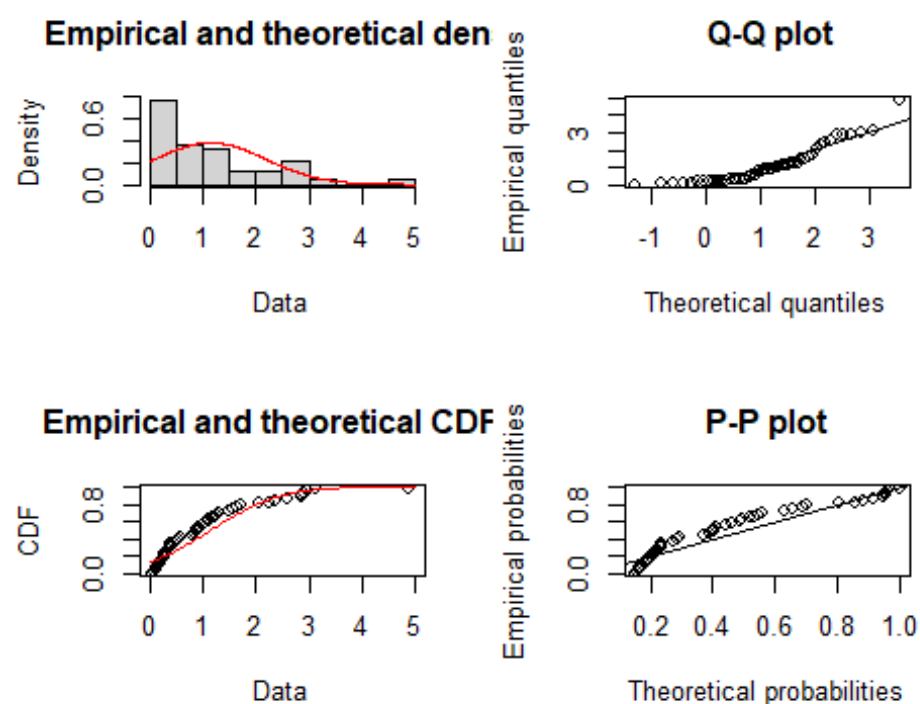
*#exp dist suits better because it presents lower AIC&BIC values*

## 5. View the values on some charts of exp distribution

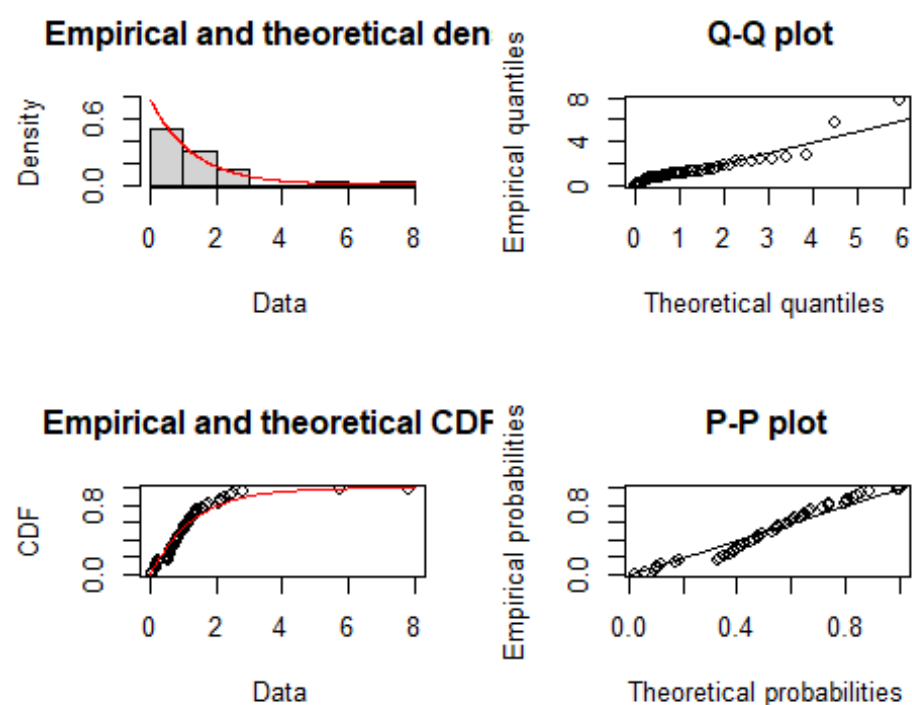
```
plotdist(vecForDifferencesMen,"exp",para=list(rate=expFit$estimate[1]))
```



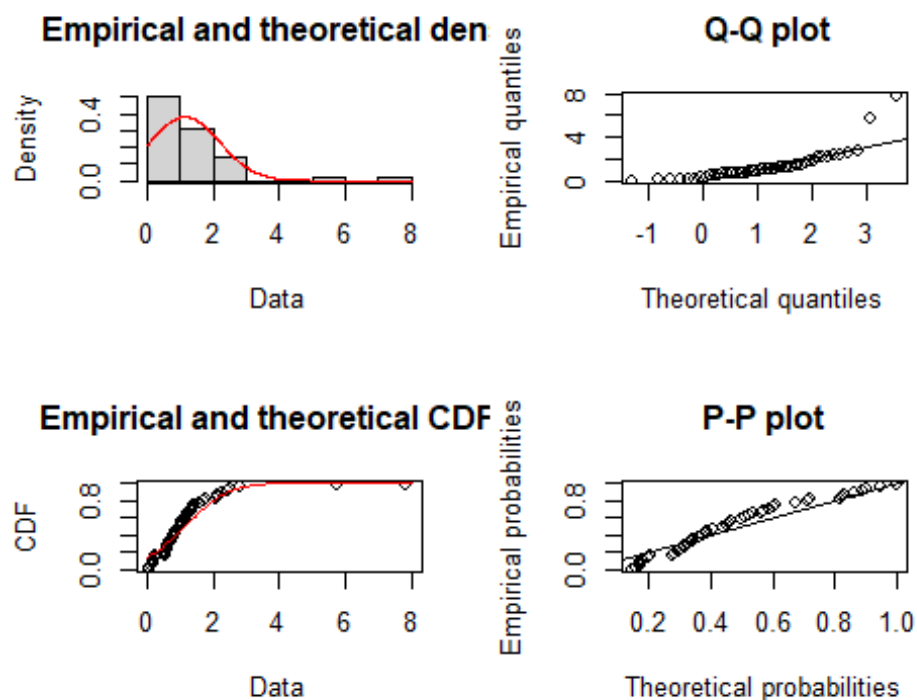
```
plotdist(vecForDifferencesMen,"norm",para=list(mean=normFit$estimate[1],sd=normFit$estimate[2]))
```



```
plotdist(vecForDifferencesWomen,"exp",para=list(rate=expFit2$estimate[1]))
```



```
plotdist(vecForDifferencesWomen,"norm",para=list(mean=normFit$estimate[1],sd=normFit$estimate[2]))
```



*#both seems to be exp*

## 6. Perform Goodness of Fit tests on fitted distribution

for both vectors: the statistical significance 0.05 because that is the default value We can see that we wouldn't reject  $H_0$  by looking at the P-val=the min alpha for rejection

```
expgof<- gofstat(expFit)%>%print()

## Goodness-of-fit statistics
##                               1-mle-exp
## Kolmogorov-Smirnov statistic 0.08059141
## Cramer-von Mises statistic   0.04580558
## Anderson-Darling statistic   0.34756679
##
## Goodness-of-fit criteria
##                               1-mle-exp
## Akaike's Information Criterion 112.3542
## Bayesian Information Criterion 114.2460

expgof$kstest%>%print()

##      1-mle-exp
## "not rejected"

expgof2<- gofstat(expFit2)%>%print()

## Goodness-of-fit statistics
##                               1-mle-exp
## Kolmogorov-Smirnov statistic 0.1665206
```

```

## Cramer-von Mises statistic    0.2230086
## Anderson-Darling statistic   1.2385506
##
## Goodness-of-fit criteria
##                               1-mle-exp
## Akaike's Information Criterion 125.0428
## Bayesian Information Criterion 126.9346

expgof2$ks.test%>%print()

##      1-mle-exp
## "not rejected"

normgof<- gofstat(normFit)%>%print()

## Goodness-of-fit statistics
##                               1-mle-norm
## Kolmogorov-Smirnov statistic  0.1577313
## Cramer-von Mises statistic    0.3838611
## Anderson-Darling statistic    2.2898892
##
## Goodness-of-fit criteria
##                               1-mle-norm
## Akaike's Information Criterion 147.5934
## Bayesian Information Criterion 151.3770

normgof$ks.test%>%print()

##      1-mle-norm
## "not rejected"

normgof2<- gofstat(normFit2)%>%print()

## Goodness-of-fit statistics
##                               1-mle-norm
## Kolmogorov-Smirnov statistic  0.2128173
## Cramer-von Mises statistic    0.6477200
## Anderson-Darling statistic    3.9329606
##
## Goodness-of-fit criteria
##                               1-mle-norm
## Akaike's Information Criterion 171.6553
## Bayesian Information Criterion 175.4389

normgof2$ks.test%>%print()

## 1-mle-norm
## "rejected"

```

## נספח 5.1.2 – תכונות הישויות

עבור הישויות במערכת (מתעמלים ומתעמלות), נשמרו התכונות הבאות:

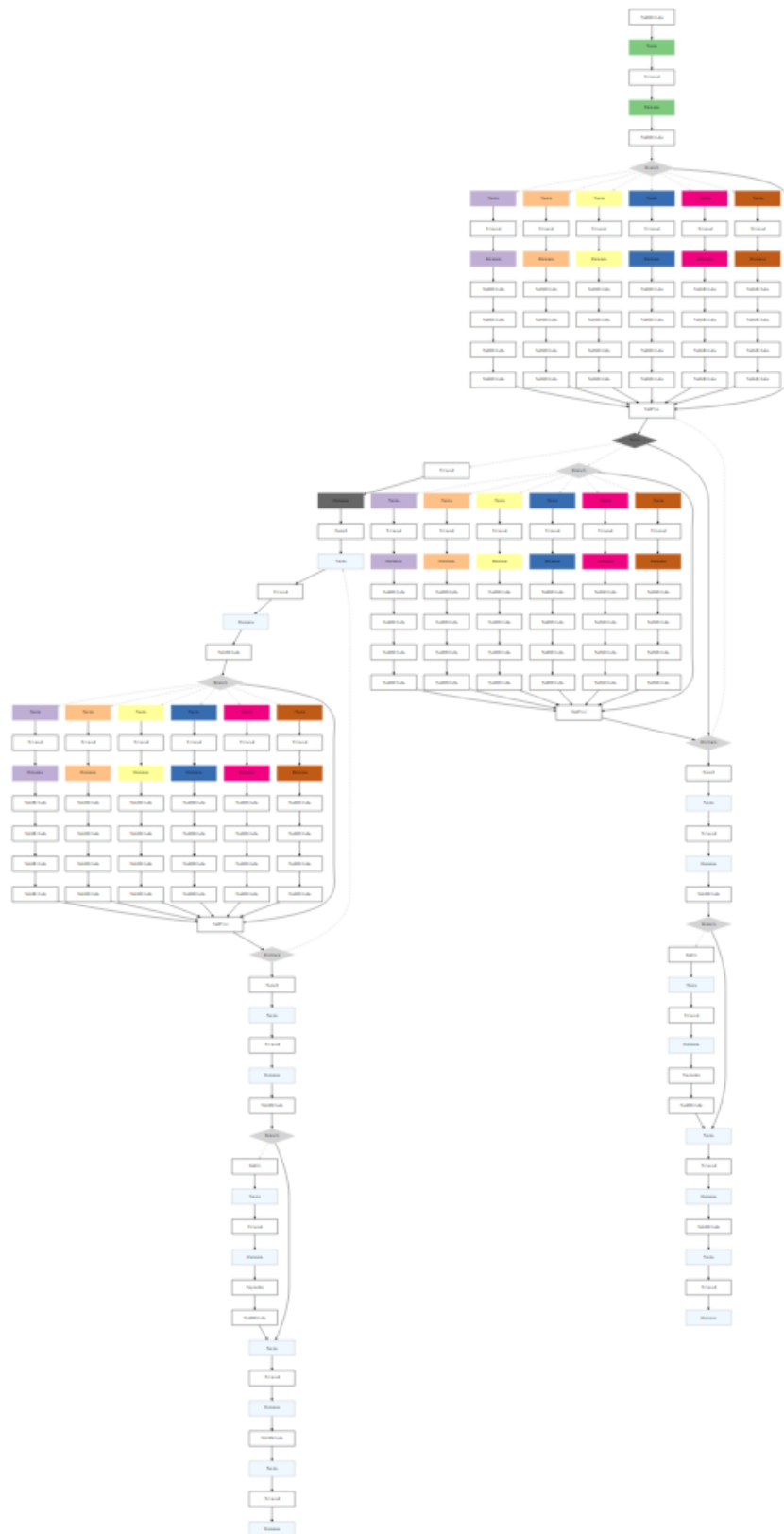
שם התכונה	תיאור התכונה
<b>video</b>	התכונה מתארת את חדר הוידאו הקבוע אליו הישות צריכה ללכת כאשר עליה לצפות בסרטוני הוידאו. התכונה מקבלת את הערכים 1-5 שהם מספרי חדרי הוידאו בהתאמה.
<b>count</b>	התכונה משמשת על מנת לספור את כמות הסרטונים שעל הישות לצפות ברצף כאשר היא מגיעה אל בוחן הוידאו. התכונה משמשת כאשר בוחני הוידאו טרם הגיעו והמתעמלים ממשיכים להתאמן, ובהגיעם עליהם לצפות במספר סרטונים ברצף.
<b>numOfMachines</b>	התכונה משמשת על מנת לספור את כמות המכשירים אשר המתעמלים ביקרו באימון.
<b>tiredness</b>	התכונה משמשת על מנת לשמור לכל מתעמל את רמת העייפות המצטברת שלו.
<b>ifGround</b>	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר הקרקע.
<b>ifSteppedParallels</b>	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר המקבילים המדורגים.

ifBeam1	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר הקורה הראשון.
ifBeam2	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר הקורה השני.
ifJumps1	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר הקפיצות הראשון.
ifJumps2	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר הקפיצות השני.
ifParallels	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר המקבילים.
ifRings	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר הטבעות.
ifTension	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר המתח.
ifAdjacentHorse	תכונה אשר משמשת כחיווי האם המתעמל התאמן במכשיר הסוס הסמוכות.

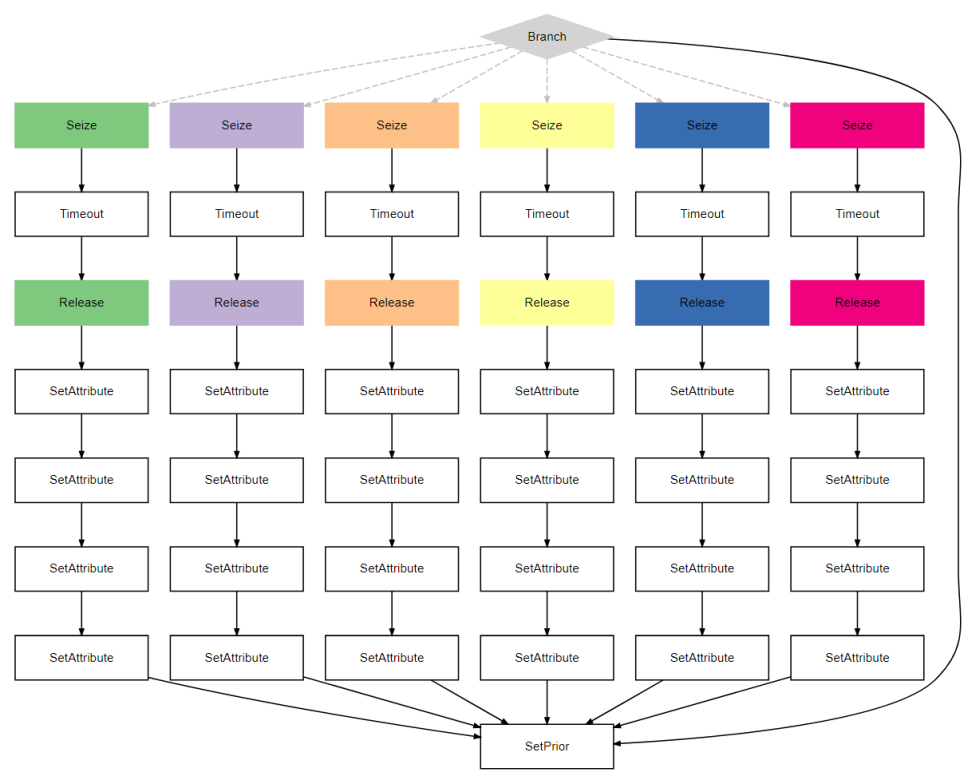
- timeForBatch – תכונה גלובלית שמגדירה את זמן ההרצאה בשעה עגולה.



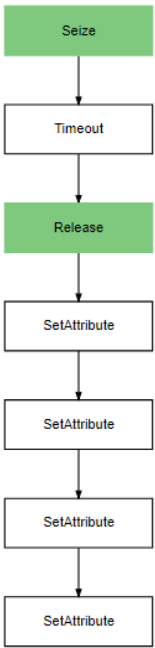
### נספח 5.1.3 – המסלולים הראשיים



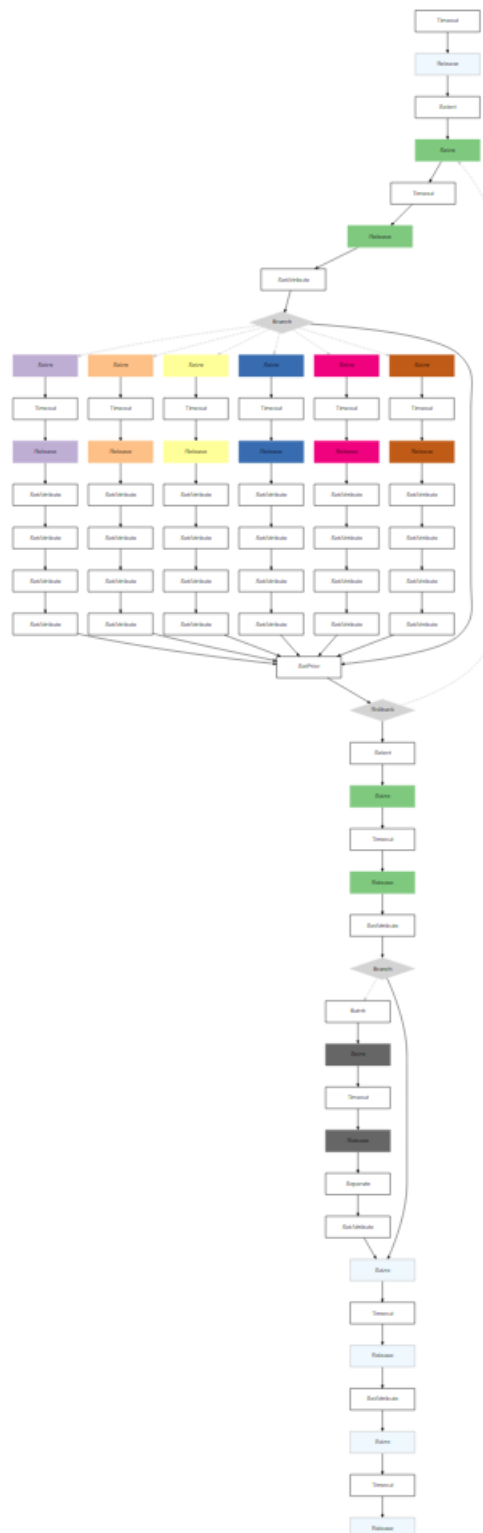
נספח 5.1.4 – מסלולי האימון



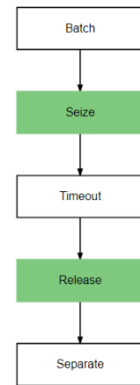
נספח 5.1.5 – מסלולי המכשירים



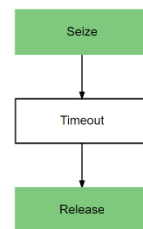
### נספח 5.1.6 – מסלולי הוידאו



### נספח 5.1.7 – מסלולי התזונאים



### נספח 5.1.8 – מסלול הפיזיותרפיסט



## נספח 6.1.6 – חישוב מדד אחוז מסיימים

```
#----- Proportion of gymnasts done
a<-(sqldf("select A.replication, count(*) as count
          from attributeData as A join attributeData as ARR on A.name=ARR.name
          where A.name like 'menGymnast%' and A.key='numOfMachines' and A.value='6' and ARR.key='isPhysyo' and ARR.value='1'
          group by A.replication "))

b<-(sqldf("select A.replication, count(*) as count
          from attributeData as A join attributeData as ARR on A.name=ARR.name
          where A.name like 'womanGymnast%' and A.key='numOfMachines' and A.value='4' and ARR.key='isPhysyo' and ARR.value='1'
          group by A.replication "))

c<-(sqldf("select replication, name
          from attributeData
          where name like 'menGymnast%' or name like 'womanGymnast%'
          group by replication, name"))

num_arrival<-(sqldf("select replication, count(*) as count
                    from c
                    group by replication"))

ans_current<-c()

for (i in 1:15){
  ans_current[0]=i
  ans_current[i]=(a$count[i]+b$count[i])/num_arrival$count[i]
}
```

$$\text{done percentage} = \frac{\text{num of gymnasts done training}}{\text{num of gymnast came}}$$

## נספח 6.1.2 – חישוב מדד זמן שהייה ממוצע

```
#----- Average flow time

FlowMeanData_current<- sqldf("select ARRIV.replication, avg(ARRIV.end_time - ARRIV.start_time) as meanFlow
                              from arrivalData as ARRIV join attributeData as A on ARRIV.name=A.name join
                              attributeData as ATT on ATT.name=A.name
                              where (A.name like 'womanGymnast%' and A.key='numOfMachines' and A.value='4'
                              and ATT.key='isPhysyo' and ATT.value='1')
                              or (A.name like 'menGymnast%' and A.key='numOfMachines'
                              and A.value='6' and ATT.key='isPhysyo' and ATT.value='1')
                              group by ARRIV.replication")
```

$$\text{mean flow} = \frac{\sum(\text{endTime} - \text{startTime})_{\text{of all done gymnasts}}}{\text{num of gymnast done}}$$

נספח 6.2.1 – חישוב הדיוק היחסי

ממוצע זמן שהייה	אחוז מסיימים	
$\frac{\delta}{\bar{x}} = \frac{t_{15-1,1-\frac{0.05}{2}} * \frac{36.896}{\sqrt{15}}}{423.541} = 0.0482$	$\frac{\delta}{\bar{x}} = \frac{t_{15-1,1-\frac{0.05}{2}} * \frac{0.0286}{\sqrt{15}}}{0.4115} = 0.0384$	מצב קיים
$\frac{\delta}{\bar{x}} = \frac{t_{15-1,1-\frac{0.05}{2}} * \frac{25.3707}{\sqrt{15}}}{385.3043} = 0.0364$	$\frac{\delta}{\bar{x}} = \frac{t_{15-1,1-\frac{0.05}{2}} * \frac{0.0585}{\sqrt{15}}}{0.7103} = 0.0456$	חלופה 1
$\frac{\delta}{\bar{x}} = \frac{t_{15-1,1-\frac{0.05}{2}} * \frac{21.1624}{\sqrt{15}}}{389.1026} = 0.0301$	$\frac{\delta}{\bar{x}} = \frac{t_{15-1,1-\frac{0.05}{2}} * \frac{0.07202}{\sqrt{15}}}{0.6884} = 0.0579$	חלופה 2

נספח 6.3.1 – תוצאות מדדי המצב הקיים עבור 15 ההרצות

זמן שהייה ממוצע	אחוז מסיימים	מספר ריצה
442.3335	0.4405830	1
433.4329	0.4244186	2
388.8837	0.3905192	3
452.8499	0.3627232	4
398.6368	0.4689655	5
372.9534	0.4043956	6
441.7322	0.3878453	7
516.1523	0.4040632	8
402.3628	0.4438265	9
377.2389	0.3971239	10
451.4239	0.4340240	11
402.7645	0.4190586	12
431.8805	0.4282632	13
437.0796	0.3897216	14
403.3940	0.3784378	15

## חלופה 1:

אחוז מסיימים:

	Percentage of gymnasts done
1	0.8076923
2	0.7095391
3	0.6902748
4	0.6836735
5	0.5873526
6	0.6792669
7	0.8206331
8	0.6598930
9	0.7431092
10	0.6607143
11	0.7545455
12	0.6966414
13	0.7193182
14	0.7396648
15	0.7025463

מדד זמן שהייה ממוצע:

	replication	meanFlow
1	1	401.3008
2	2	364.4947
3	3	388.2058
4	4	411.9479
5	5	380.2627
6	6	389.1301
7	7	420.4630
8	8	342.2057
9	9	402.0098
10	10	370.1458
11	11	422.9616
12	12	358.0708
13	13	402.1027
14	14	346.7535
15	15	379.5099

## חלופה 2:

אחוז מסיימים:

	Percentage of gymnasts done
1	0.5715812
2	0.6324022
3	0.7585825
4	0.6192053
5	0.7366629
6	0.6926503
7	0.6363636
8	0.6240180
9	0.8061927
10	0.6630553
11	0.7546296
12	0.6511879
13	0.6376963
14	0.7567887
15	0.7852835

מדד זמן שהייה ממוצע:

	replication	meanFlow
1	1	383.7166
2	2	367.1257
3	3	381.2577
4	4	413.3632
5	5	389.4314
6	6	401.8093
7	7	394.2224
8	8	376.1423
9	9	356.6459
10	10	398.1942
11	11	352.9243
12	12	401.4546
13	13	432.8194
14	14	403.7350
15	15	383.6973



## Fitting With R

### Libraries

we will need the following libraries: fitdistrplus, magrittr

```
library(fitdistrplus)

## Loading required package: MASS

## Loading required package: survival

library(magrittr)
```

### 1. Empirical Data

Empirical data - observations. We will load the flow mean data of the current state

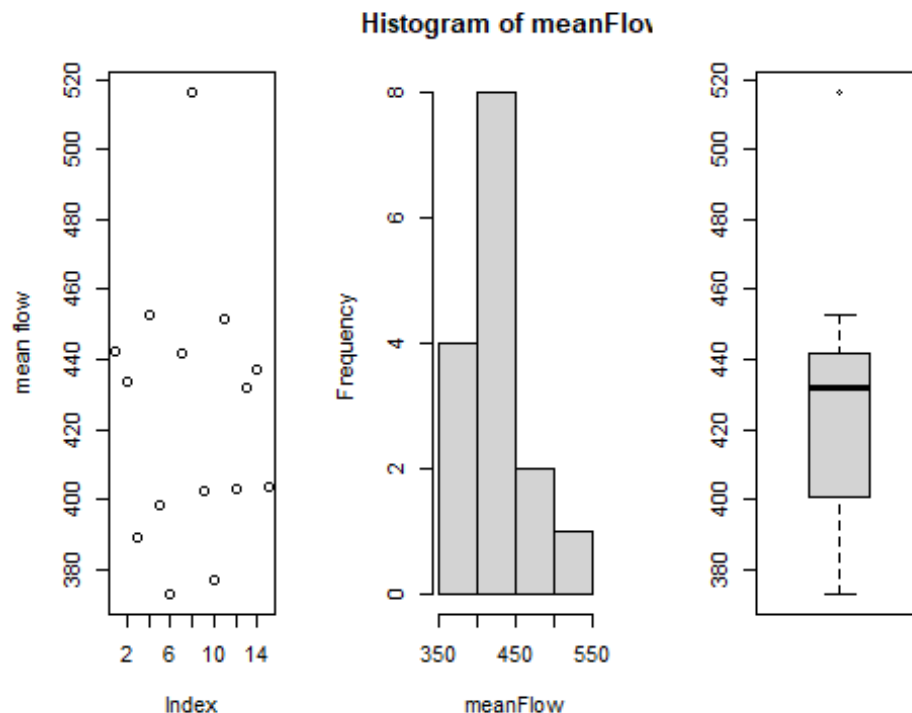
```
meanFlow<-c(442.3335,433.4329,388.8837,452.8499,398.6368,372.9534,441.7322,516.1523,402.3628,377.2389,451.4239,402.7645,431.8805,437.0796,403.3940)
print(meanFlow)

## [1] 442.3335 433.4329 388.8837 452.8499 398.6368 372.9534 441.7322 516.1523
## [9] 402.3628 377.2389 451.4239 402.7645 431.8805 437.0796 403.3940
```

### 2. First Impressions

At first always look at the data and understand - the data range, is it discrete or continuous, are there any outliers? For that we can use the following plots:

```
par(mfrow=c(1,3)) # defines 2 plots on the same window
plot(meanFlow,ylab="mean flow") # plot ( left graph )
hist(meanFlow, ylab="Frequency ") # histogram (right graph)
boxplot(meanFlow)
```



Now we can look further, and plot the histogram not in original values but as density (empirical  $f(x)$ ) and cumulative probability ( $F(x)$ ), using `plotdist` from the `fitdistrplus` library.

### 3. Fit a known distribution

Case A: samples from  $N(\text{mean}=0, \text{sd}=1)$

Now we can use R to estimate the parameters of known distribution to our data

```
normFit<-fitdist(meanFlow,"norm") # fitting a normal distribution
summary(normFit) # summary function displays the results

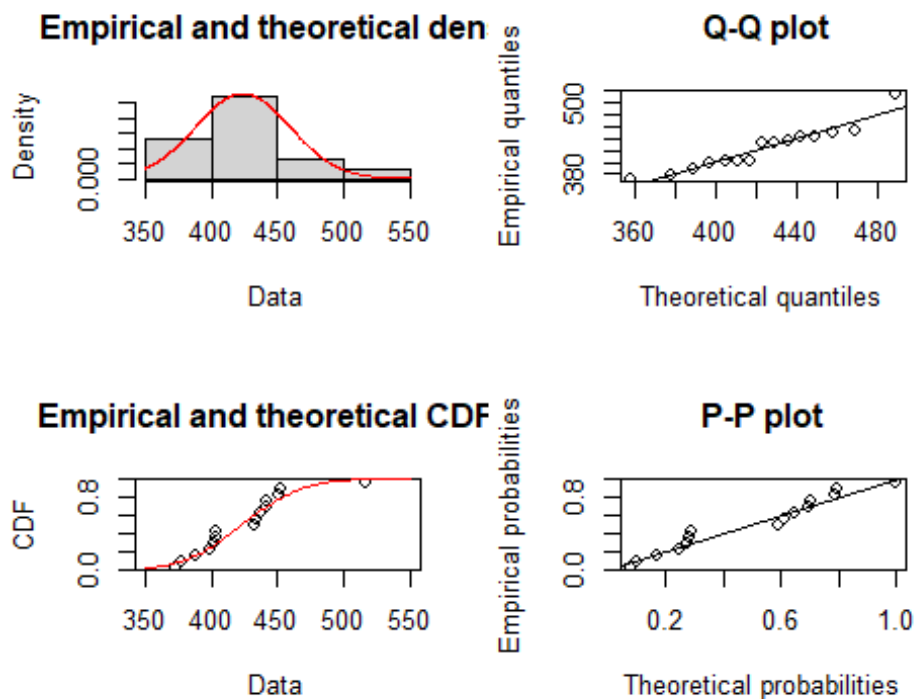
## Fitting of the distribution ' norm ' by maximum likelihood
## Parameters :
##      estimate Std. Error
## mean 423.54126    9.203504
## sd   35.64502    6.507859
## Loglikelihood: -74.88822    AIC: 153.7764    BIC: 155.1925
## Correlation matrix:
##      mean      sd
## mean 1.000000e+00 -1.063951e-07
## sd   -1.063951e-07 1.000000e+00
```

The estimates are R estimates of the parameters of a normal distribution - **If it was a normal distribution**, mean and sd. We can reach the parameters directly from code using the following:

```
normFit$estimate[1] # is the first parameter- mean
normFit$estimate[2] # is the second parameter- sd
```

Now we will use `plotdist` to plot our data versus the theoretical distribution

```
plotdist(meanFlow, "norm", para=list(mean=normFit$estimate[1], sd=normFit$estimate[2]))
```



```
normgof<- gofstat(normFit)%>%print()

## Goodness-of-fit statistics
##                               1-mle-norm
## Kolmogorov-Smirnov statistic 0.18070445
## Cramer-von Mises statistic   0.06684127
## Anderson-Darling statistic   0.46270200
##
## Goodness-of-fit criteria
##                               1-mle-norm
## Akaike's Information Criterion 153.7764
## Bayesian Information Criterion 155.1925

normgof$kstest%>%print()

##      1-mle-norm
## "not computed"
```

## 7. קוד הסימולציה:

```
#.libPaths("D:/soft/r/4.1")#this row only for labs
```

```
library(rlang)
```

```
library(MASS)
```

```
library(fitdistrplus)
```

```
library(magrittr)
```

```
library(dplyr)
```

```
library(lazyeval)
```

```
library(parallel)
```

```
library(e1071)
```

```
library(plotly)
```

```
library(ggplot2)
```

```
library(triangle)
```

```
library(sqldf)
```

```
#library(readxl)
```

```
# library(knitr)
```

```
# library(rmarkdown)
```

```
library(simmer)
```

```
library(simmer.plot)
```

```
##----- 1. all functions -----
```

```
## add service
```

```
addService<- function (trajectory,sname,timeDist){
```

```
  updatedPath <- seize(trajectory, resource = sname, amount = 1)%>%
```

```
    timeout(timeDist) %>%
```

```
    release(resource = sname, amount = 1)
```

```
  return(updatedPath)
```

```
}
```

#add service with reneging

```
addServiceRenge<- function (trajectory,sname,attName,renegDist, continueTo){
```

```
  updatedPath <- renege_in(trajectory,renegDist,out= continueTo)%>%
```

```
    seize(resource = sname, amount = 1)%>%
```

```
    renege_abort()%>%
```

```
    timeout_from_attribute(attName) %>%
```

```
    release(resource = sname, amount = 1)
```

```
  return(updatedPath)
```

```
}
```

#trimmed norm

```
trimmedNorm<-function(mu,sd){
```

```
  while(TRUE){
```

```
    sample<-rnorm(1,mu,sd)
```

```
    if (sample>0)
```

```
      return (sample)
```

```
  }
```

```
}
```

#setting the initial attributes

```
gymnasticAtt<-function(){
```

```
  p<-runif(1,0,1)
```

```
  if (p<0.2){
```

```
    video=1
```

```
  }
```

```
  if (p>=0.2 && p< 0.4){
```

```
    video=2
```

```
  }
```

```
  if (p>=0.4 && p< 0.6){
```

```

    video=3
  }
  if (p>=0.6 && p< 0.8){
    video=4
  }
  if (p>=0.8){
    video=5
  }
  count=0
  numOfMachines=0
  tiredness=0
  ifDone=0
  isNutrition=0
  isPhysyo=0
  return (c(video, count, numOfMachines, tiredness,ifDone,isNutrition,isPhysyo))
}

```

#counting the number of videos that need to watch in a row

```

trainingAtt<-function(){
  count = get_attribute(mySimulation,"count")+1
  return (count)
}

```

#counting the number of machines that the gymnast has done

```

machinesAtt<-function(){
  numOfMachines = get_attribute(mySimulation,"numOfMachines")+1
  return (numOfMachines)
}

```

#checks if the woman can keep training - if not too tired or hasn't done all her machines

```
canWomenKeepTraining<-function(){  
  if (get_attribute(mySimulation,"numOfMachines")< 4 &&  
get_attribute(mySimulation,"tiredness")<=2.4){  
    return (TRUE)  
  }  
  else{  
    return (FALSE)  
  }  
}
```

#checks if the man can keep training - if not too tired or hasn't done all his machines

```
canMenKeepTraining<-function(){  
  if (get_attribute(mySimulation,"numOfMachines")< 6 &&  
get_attribute(mySimulation,"tiredness")<=2.9){  
    return (TRUE)  
  }  
  else{  
    return (FALSE)  
  }  
}
```

#returns the accomulative tiredness

```
getTiredness<-function(){  
  u<-runif(1,0,1)  
  if (u<=0.5){  
    return ((3*u/8)^(1/3)+get_attribute(mySimulation, "tiredness"))  
  }  
  if (u> 0.5 && u<= 2/3){
```

```

    return (((u+2/3)/2)+get_attribute(mySimulation, "tiredness"))
}
if (u>2/3){
    return (((-6+sqrt(12*(1-u)))/-6)+get_attribute(mySimulation, "tiredness"))
}
}

```

#setting priority according to tiredness

```

prioritySetWomen<- function(){
  if(get_attribute(mySimulation,"tiredness") >2.4){
    v<-c(1,1,F)
    return(v)
  }
  else{
    v<-c(0,0,F)
    return(v)
  }
}

```

#setting priority according to tiredness

```

prioritySetMen<- function(){
  if(get_attribute(mySimulation,"tiredness") >2.9){
    v<-c(1,1,F)
    return(v)
  }
  else{
    v<-c(0,0,F)
    return(v)
  }
}

```



#returns the machines that the woman gymnast hasn't done yet

```
resourceToVisitWomen<-function(){  
  v<-c()  
  if(get_attribute(mySimulation,"ifGround") == 0){  
    v<-append(v,1)  
  }  
  if(get_attribute(mySimulation,"ifSteppedParallels") == 0){  
    v<-append(v,2)  
  }  
  if(get_attribute(mySimulation,"ifBeam1") == 0 && get_attribute(mySimulation,"ifBeam2")  
== 0){  
    p<-runif(1,0,1)  
    if(p<0.5){  
      v<-append(v,3)  
    }  
    if(p>=0.5){  
      v<-append(v,4)  
    }  
  }  
  if(get_attribute(mySimulation,"ifJumps1") == 0){  
    v<-append(v,5)  
  }  
  
  return(sample(v,1))  
}
```

#returns the machines that the man gymnast hasn't done yet

```
resourceToVisitMen<-function(){  
  v<-c()  
  if(get_attribute(mySimulation,"ifParallels") == 0){
```

```

    v<-append(v,1)
  }
  if(get_attribute(mySimulation,"ifRings") == 0){
    v<-append(v,2)
  }
  if(get_attribute(mySimulation,"ifTension") == 0){
    v<-append(v,3)
  }
  if(get_attribute(mySimulation,"ifAdjacentHorse") == 0){
    v<-append(v,4)
  }
  if(get_attribute(mySimulation,"ifGround") == 0){
    v<-append(v,5)
  }
  if(get_attribute(mySimulation,"ifJumps2") == 0){
    v<-append(v,6)
  }

  return(sample(v,1))

}

breakTime<-function (){
  p<-trimmedNorm(6,50/60)
  return (p+420)
}

```

```
##----- 2. all simulation parameters -----  
-----
```

```
simulationTime <- 14*60
```

```
door_schedule<-schedule(timetable = c(0, 120), values = c(0, 10), period = Inf)
```

```
door_queue<-schedule(timetable = c(0, 120), values = c(0, Inf), period = Inf)
```

```
#setting the capacity and queue sizes according to schedule
```

```
nutrition_schedule<-schedule(timetable = c(0,120), values = c(0,1), period = Inf)
```

```
PY_break_schedule<-schedule(timetable = c(0,120,360,600), values = c(0,2,5,3), period = Inf)
```

```
video_schedule<-schedule(timetable = c(0,120), values = c(0,2), period = Inf)
```

```
video_queue<-schedule(timetable = c(0,120), values = c(0,Inf), period = Inf)
```

```
##----- 3. Init Simulation and add all resources -----  
-----
```

```
mySimulation<- simmer("mySimulation")%>%
```

```
  add_resource("Door", capacity = door_schedule, queue_size = door_queue)%>%
```

```
  add_resource("womenDressingRoom",capacity=20,queue_size=Inf)%>%
```

```
  add_resource("menDressingRoom",capacity=20,queue_size=Inf)%>%
```

```
  add_resource("womenShower",capacity=5,queue_size=Inf)%>%
```

```
  add_resource("menShower",capacity=5,queue_size=Inf)%>%
```

```
  add_resource("video1",capacity=video_schedule,queue_size=video_queue)%>%
```

```
  add_resource("video2",capacity=video_schedule,queue_size=video_queue)%>%
```

```
  add_resource("video3",capacity=video_schedule,queue_size=video_queue)%>%
```

```
  add_resource("video4",capacity=video_schedule,queue_size=video_queue)%>%
```

```
  add_resource("video5",capacity=video_schedule,queue_size=video_queue)%>%
```

```
  add_resource("parallels",capacity=1,queue_size=Inf)%>%
```

```
  add_resource("rings",capacity=1,queue_size=Inf)%>%
```

```
  add_resource("tension",capacity=1,queue_size=Inf)%>%
```

```
  add_resource("adjacentHorse",capacity=1,queue_size=Inf)%>%
```

```
  add_resource("ground",capacity=1,queue_size=Inf)%>%
```

```

add_resource("jumps1",capacity=,queue_size=Inf)%>%
add_resource("jumps2",capacity=1,queue_size=Inf)%>%
add_resource("steppedParallels",capacity=1,queue_size=Inf)%>%
add_resource("beam1",capacity=1,queue_size=Inf)%>%
add_resource("beam2",capacity=1,queue_size=Inf)%>%
add_resource("nutrition1",capacity=nutrition_schedule,queue_size=Inf)%>%
add_resource("nutrition2",capacity=nutrition_schedule,queue_size=Inf)%>%
add_resource("physiotherapist",capacity=PY_break_schedule,queue_size=Inf, preemptive
= FALSE)

```

##----- 4. All trajectories, start from main trajectory and add sub-  
trajectories ABOVE IT it . -----

## you can use mySimulation pointer when needed inside the trajectories.

#----- Machines Trajs -----

#for every machine: set that has been done, count to watch vide, count number of  
#machines that has been done and get tiredness.

```

ground<-trajectory("ground")%>%
  addService ("ground",function() trimmedNorm(5,1.7))%>%
  set_attribute(key=c("ifGround"),value=(1))%>%
  set_attribute(key=c("count"),value=function() trainingAtt())%>%
  set_attribute(key=c("tiredness"), value= function() getTiredness())%>%
  set_attribute(key=c("numOfMachines"),value=function() machinesAtt())

```

```

steppedParallels<-trajectory("steppedParallels")%>%
  addService ("steppedParallels",function() trimmedNorm(5,1.7))%>%
  set_attribute(key=c("ifSteppedParallels"),value=(1))%>%
  set_attribute(key=c("count"),value=function() trainingAtt())%>%

```

```
set_attribute(key=c("tiredness"), value= function() getTiredness()))%>%  
set_attribute(key=c("numOfMachines"),value=function() machinesAtt())
```

```
beam1<-trajectory("beam1")%>%  
addService ("beam1",function() trimmedNorm(5,1.7))%>%  
set_attribute(key=c("ifBeam1"),value=(1))%>%  
set_attribute(key=c("count"),value=function() trainingAtt()))%>%  
set_attribute(key=c("tiredness"), value= function() getTiredness()))%>%  
set_attribute(key=c("numOfMachines"),value=function() machinesAtt())
```

```
beam2<-trajectory("beam2")%>%  
addService ("beam2",function() trimmedNorm(5,1.7))%>%  
set_attribute(key=c("ifBeam2"),value=(1))%>%  
set_attribute(key=c("count"),value=function() trainingAtt()))%>%  
set_attribute(key=c("tiredness"), value= function() getTiredness()))%>%  
set_attribute(key=c("numOfMachines"),value=function() machinesAtt())
```

```
jumps1<-trajectory("jumps1")%>%  
addService ("jumps1",function() trimmedNorm(5,1.7))%>%  
set_attribute(key=c("ifJumps1"),value=(1))%>%  
set_attribute(key=c("count"),value=function() trainingAtt()))%>%  
set_attribute(key=c("tiredness"), value= function() getTiredness()))%>%  
set_attribute(key=c("numOfMachines"),value=function() machinesAtt())
```

```
jumps2<-trajectory("jumps2")%>%  
addService ("jumps2",function() trimmedNorm(5,1.7))%>%  
set_attribute(key=c("ifJumps2"),value=(1))%>%  
set_attribute(key=c("count"),value=function() trainingAtt()))%>%  
set_attribute(key=c("tiredness"), value= function() getTiredness()))%>%  
set_attribute(key=c("numOfMachines"),value=function() machinesAtt())
```

```

parallels<-trajectory("parallels")%>%
  addService ("parallels",function() trimmedNorm(5,1.7))%>%
  set_attribute(key=c("ifParallels"),value=(1))%>%
  set_attribute(key=c("count"),value=function() trainingAtt())%>%
  set_attribute(key=c("tiredness"), value= function() getTiredness())%>%
  set_attribute(key=c("numOfMachines"),value=function() machinesAtt())

```

```

rings<-trajectory("rings")%>%
  addService ("rings",function() trimmedNorm(5,1.7))%>%
  set_attribute(key=c("ifRings"),value=(1))%>%
  set_attribute(key=c("count"),value=function() trainingAtt())%>%
  set_attribute(key=c("tiredness"), value= function() getTiredness())%>%
  set_attribute(key=c("numOfMachines"),value=function() machinesAtt())

```

```

tension<-trajectory("tension")%>%
  addService ("tension",function() trimmedNorm(5,1.7))%>%
  set_attribute(key=c("ifTension"),value=(1))%>%
  set_attribute(key=c("count"),value=function() trainingAtt())%>%
  set_attribute(key=c("tiredness"), value= function() getTiredness())%>%
  set_attribute(key=c("numOfMachines"),value=function() machinesAtt())

```

```

adjacentHorse<-trajectory("adjacentHorse")%>%
  addService ("adjacentHorse",function() trimmedNorm(5,1.7))%>%
  set_attribute(key=c("ifAdjacentHorse"),value=(1))%>%
  set_attribute(key=c("count"),value=function() trainingAtt())%>%
  set_attribute(key=c(
    "tiredness"), value= function() getTiredness())%>%
  set_attribute(key=c("numOfMachines"),value=function() machinesAtt())

```

```
#-----
```

```
#----- Times And Breakes Trajs -----
```

```
#setting the next nutrition lecture to be every round hour
```

```
batchScheduleTrj<- trajectory("batchScheduleTrj")%>%
```

```
  set_global(key="timeForBatch",value=(function () now(mySimulation)+60))
```

```
#-----
```

```
BreakStart<-trajectory("BreakStart")%>%
```

```
  set_capacity("video1",0)%>%
```

```
  set_capacity("video2",0)%>%
```

```
  set_capacity("video3",0)%>%
```

```
  set_capacity("video4",0)%>%
```

```
  set_capacity("video5",0)%>%
```

```
  set_capacity("nutrition1",0)%>%
```

```
  set_capacity("nutrition2",0)%>%
```

```
  set_capacity("physiotherapist",0)%>%
```

```
  timeout(function() trimmedNorm(6,50/60))%>%
```

```
  log_("break end")%>%
```

```
  set_capacity("video1",2)%>%
```

```
  set_capacity("video2",2)%>%
```

```
  set_capacity("video3",2)%>%
```

```
  set_capacity("video4",2)%>%
```

```
  set_capacity("video5",2)%>%
```

```
  set_capacity("nutrition1",1)%>%
```

```
  set_capacity("nutrition2",1)%>%
```

```
  set_capacity("physiotherapist",5)
```

```
#nutrition - watching lecture in groups of 10 or in a round hour
```

```
nutritionTrajWomen<- trajectory("nutritionTrajWomen")%>%
```

```
batch(n=10,timeout=function() {get_global(mySimulation,"timeForBatch")-  
now(mySimulation)},permanent=FALSE)%>%
```

```
addService("nutrition1", function() runif(1,30,40))%>%
```

```
separate()%>%
```

```
set_attribute(key=c("isNutrition"),value=1)
```

```
#nutrition - watching lecture in groups of 10 or in a round hour
```

```
nutritionTrajMen<- trajectory("nutritionTrajMen")%>%
```

```
batch(n=10,timeout=function() {get_global(mySimulation,"timeForBatch")-  
now(mySimulation)},permanent=FALSE)%>%
```

```
addService("nutrition2", function() runif(1,30,40))%>%
```

```
separate()%>%
```

```
set_attribute(key=c("isNutrition"),value=1)
```

```
#physiotherapy traj
```

```
physiotherapyTraj<- trajectory("physiotherapyTraj")%>%
```

```
addService("physiotherapist",function()rtriangle(1,25,40,33))%>%
```

```
set_attribute(key=c("isPhysyo"),value=1)
```

```
#traj for training - branching to machines
```

```
womenTraining<-trajectory("womenTraining")%>%
```

```
branch(option=function() resourceToVisitWomen() ,continue=  
c(TRUE,TRUE,TRUE,TRUE,TRUE),ground,steppedParallels,beam1,beam2, jumps1)%>%
```

```
#go training
```

```
set_prioritization(function() prioritySetWomen())
```

```
#traj for training - branching to machines
```

```
menTraining<-trajectory("menTraining")%>%
```

```
branch(option= function() resourceToVisitMen() ,continue=  
c(TRUE,TRUE,TRUE,TRUE,TRUE,TRUE),parallels,rings,tension,adjacentHorse,ground,  
jumps2)%>%
```

```
set_prioritization(function() prioritySetMen())
```



```

#traj for women who succeeded entering the video rooms - keep training and watching
videos.

#after done training/ too tired - going to video again and then nutrition & physio/ physio
only and to shower.videoTrajWomen<-trajectory("videoTrajWomen")%>%

videoTrajWomen<-trajectory("videoTrajWomen")%>%

  timeout(0)%>%

  release("Door",1)%>%

  simmer::select(resource= function() paste0("video", get_attribute(mySimulation,
"video")))%>%   #seize video

  seize_selected(amount=1)%>%

  timeout(function() trimmedNorm(3,45/60)*(get_attribute(mySimulation, "count")))%>%
#watch video

  release_selected(amount=1)%>%

  set_attribute(key=c("count"),value=0)%>%      #updating no more video duplication to
watch

  join(womenTraining)%>%

  rollback(amount=6, times=100, check=function() canWomenKeepTraining())%>%   #go
training again----

  branch(option=function() rdiscrete (1, c(0.32,0.68),c(0,1)) ,continue=
c(TRUE),nutritionTrajWomen)%>%

  join(physiotherapyTraj)%>%

  addService("womenShower",function() runif(1,8,14))

```

```

#traj for men who succeeded entering the video rooms - keep training and watching videos.

#after done training/ too tired - going to video again and then nutrition & physio/ physio
only and to shower.

videoTrajMen<-trajectory("videoTrajMen")%>%

  timeout(0)%>%

  release("Door",1)%>%

  simmer::select(resource= function() paste0("video", get_attribute(mySimulation,
"video")))%>%   #seize video

  seize_selected(amount=1)%>%

  timeout(function() trimmedNorm(3,45/60)*(get_attribute(mySimulation, "count")))%>%
#watch video

```

```

release_selected(amount=1)%>%

set_attribute(key=c("count"),value=0)%>%      #updating no more video duplication to
watch

join(menTraining)%>%

rollback(amount=6, times=100, check=function() canMenKeepTraining())%>%  #go training
again----

simmer::select(resource= function())

paste0("video", get_attribute(mySimulation, "video")))%>%  #seize video

seize_selected(amount=1)%>%

timeout(function() trimmedNorm(3,45/60)*(get_attribute(mySimulation, "count")))%>%
#watch video

release_selected(amount=1)%>%

set_attribute(key=c("count"),value=0)%>%      #updating no more video duplication to
watch

branch(option=function() rdiscrete (1, c(0.32,0.68),c(0,1)) ,continue=
c(TRUE),nutritionTrajMen)%>%

join(physiotherapyTraj)%>%

addService("menShower",function() runif(1,8,14))

```

#main traj for women - going to dressing rooms and then training. then trying to watch video.

#women who watched video - moving to video traj and stays there.

#women who can't watch video (didn't arrive yet), going to train again and return to this traj.

#after done training/ too tired - going to video again and then nutrition & physio/ physio only and to shower.

```
womenTraj<- trajectory("womenTraj")%>%
```

```
set_attribute(key=c("video","count","numOfMachines","tiredness","ifDone","isPhysyo","isN
utrition"),value=function() gymnasticAtt())%>%
```

```
addService("womenDressingRoom",function() runif(1,3,5))%>%  #dressing room
```

```
set_attribute(key=c("ifGround","ifSteppedParallels","ifBeam1","ifBeam2","ifJumps1"),value=
c(0,0,0,0,0))%>%
```

```
join(womenTraining)%>%
```

```
seize("Door",amount = 1, continue= c(F,T),
post.seize=videoTrajWomen,reject=womenTraining)%>%      #try to enter the video
room
```

```
rollback(amount=2, times=100, check=function() canWomenKeepTraining())%>%  #go
training again----
```

```
simmer::select(resource= function()
```

```
paste0("video", get_attribute(mySimulation, "video")))%>%  #seize video
```

```
seize_selected(amount=1)%>%
```

```
timeout(function() trimmedNorm(3,45/60)*(get_attribute(mySimulation, "count")))%>%
#watch video
```

```
release_selected(amount=1)%>%
```

```
set_attribute(key=c("count"),value=0)%>%      #updating no more video duplication to
watch
```

```
branch(option=function() rdiscrete (1, c(0.32,0.68),c(0,1)) ,continue=
c(TRUE),nutritionTrajWomen)%>%
```

```
join(physiotherapyTraj)%>%
```

```
addService("womenShower",function() runif(1,8,14))
```

#main traj for men - going to dressing rooms and then training. then trying to watch video.

#men who watched video - moving to video traj and stays there.

#men who can't watch video (didn't arrive yet), going to train again and return to this traj.

#after done training/ too tired - going to video again and then nutrition & physio/ physio  
only and to shower.

```
menTraj <- trajectory("menTraj") %>%
```

```
set_attribute(key=c("video","count","numOfMachines","tiredness","ifDone","isPhysyo","isN
utrition"),value=function() gymnasticAtt())%>%
```

```
addService("menDressingRoom",function() runif(1,3,5))%>%
```

```

set_attribute(key=c("ifParallels","ifRings","ifTension","ifAdjacentHorse","ifGround","ifJumps
2"),value=c(0,0,0,0,0,0))%>%

join(menTraining)%>%

seize("Door",amount = 1, continue= c(F,T), post.seize =videoTrajMen,
reject=menTraining)%>%      #try to enter the video room

rollback(amount=2, times=100, check=function() canMenKeepTraining())%>% #for those
who came before the video guys - go training again

simmer::select(resource= function()

paste0("video", get_attribute(mySimulation, "video")))%>% #seize video

seize_selected(amount=1)%>%

timeout(function() trimmedNorm(3,45/60)*(get_attribute(mySimulation, "count")))%>%
#watch video

release_selected(amount=1)%>%

set_attribute(key=c("count"),value=0)%>%      #updating no more video duplication to
watch

branch(option=function() rdiscrete (1, c(0.32,0.68),c(0,1)) ,continue=
c(TRUE),nutritionTrajMen)%>%

join(physiotherapyTraj)%>%

addService("menShower",function() runif(1,8,14))

```

```

##----- 5. All Generators, ALWAYS LAST. -----
-----

```

```

mySimulation%>%

add_generator("MenGymnast", menTraj, distribution = to(540,function ()
rexp(1,0.8815592)), priority=0, mon=2)%>%

add_generator("WomanGymnast", womenTraj, distribution = to(540,function ()
rexp(1,0.7744995)), priority=0, mon=2)%>%

```

```
add_generator("batchSchedule",batchScheduleTrj,from (0,function ())(60)), mon=2)%>%
add_generator("StartBreak",BreakStart,at (420), mon=2)
```

```
##----- 6. reset, run, plots, outputs -----
-----
```

```
mm2envs <- mclapply(1:15, function(i) {
  set.seed(456+i)
  reset(mySimulation)%>%run(until=840) %>%
  wrap()
})
```

```
arrivalData <- get_mon_arrivals(mm2envs)
resourceData <- get_mon_resources(mm2envs)
attributeData <- get_mon_attributes(mm2envs)
```

```
#----- Proportion of gymnasts done
a<-(sqldf("select A.replication, count(*) as count
          from attributeData as A join attributeData as ARR on A.name=ARR.name
          where A.name like 'menGymnast%' and A.key='numOfMachines'and A.value='6' and
          ARR.key='isPhysyo' and ARR.value='1'
          group by A.replication "))
```

```

b<-(sqldf("select A.replication, count(*) as count
          from attributeData as A join attributeData as ARR on A.name=ARR.name
          where A.name like 'womanGymnast%' and A.key='numOfMachines'and A.value='4'
and ARR.key='isPhysyo' and ARR.value='1'
          group by A.replication  "))

```

```

c<-(sqldf("select replication, name
          from attributeData
          where name like 'menGymnast%' or name like 'WomanGymnast%'
          group by replication, name"))

```

```

num_arrival<-(sqldf("select replication, count(*) as count
          from c
          group by replication"))

```

```

ans_current<-c()

```

```

for (i in 1:15){
  ans_current[0]=i
  ans_current[i]=(a$count[i]+b$count[i])/num_arrival$count[i]
}

```

```

print(ans_current)
print (mean(ans_current))
print(sd(ans_current))

```

```

test<- t.test(x= ans_current,y=NULL, alternative="two.sided",conf.level=0.95)
print(test)

```

#----- Average flow time for every run

```
FlowMeanData_current<- sqldf("select ARRIV.replication, avg(ARRIV.end_time -  
ARRIV.start_time) as meanFlow  
  
    from arrivalData as ARRIV join attributeData as A on ARRIV.name=A.name join  
attributeData as ATT on ATT.name=A.name  
  
    where (A.name like 'womanGymnast%' and A.key='numOfMachines' and  
A.value='4' and ATT.key='isPhysyo' and ATT.value='1') or (A.name like 'menGymnast%' and  
A.key='numOfMachines' and A.value='6' and ATT.key='isPhysyo' and ATT.value='1')  
  
    group by ARRIV.replication")
```

```
print(mean(FlowMeanData_current$meanFlow))
```

```
print(sd(FlowMeanData_current$meanFlow))
```

```
test0<- t.test(x= FlowMeanData_current$meanFlow,y=NULL,  
alternative="two.sided",conf.level=0.95)
```

```
print(test0)
```