עופרי פוקס ת.ז 313528341  
עידו נחמני ת.ז 204677108  
ריטה ברודסקי ת.ז 318194115

‏19/6/2022

רכבת ישראל – פרויקט סימולציה

תמונה שמכילה עץ, חוץ, דשא, אנשים

התיאור נוצר באופן אוטומטי

תוכן עניינים

[תקציר 3](#_Toc106476838)

[מבוא 3](#_Toc106476839)

[תיאור המערכת הנחקרת 3](#_Toc106476840)

[מטרות 4](#_Toc106476841)

[מודל סימולציה 5](#_Toc106476842)

[תיאור המודל 5](#_Toc106476843)

[הנחות 6](#_Toc106476844)

[ניתוח מצב קיים והצעת חלופות לשיפור מצב קיים 7](#_Toc106476845)

[בחירת מדדים לניתוח 7](#_Toc106476846)

[סוג המערכת 7](#_Toc106476847)

[ניתוח מצב קיים ביחס למדדים ולמטרות 7](#_Toc106476848)

[תיאור החלופות 9](#_Toc106476849)

[השוואה סטטיסטית בין המצב הקיים לחלופות המוצעות 10](#_Toc106476850)

[מסקנות 11](#_Toc106476851)

[נספחים 12](#_Toc106476852)

[נספח 1- משאבי התחנה 12](#_Toc106476853)

[נספח 1.1- משאבי התחנה 12](#_Toc106476854)

[נספח 1.2- הסתברויות לבחירת קווי עליה וירידה 12](#_Toc106476855)

[נספח 2- זמני שירות 13](#_Toc106476856)

[נספח 2.1- זמני שירות עבור עמדת אבטחה 13](#_Toc106476857)

[נספח 2.2- זמני שירות עבור קניית כרטיסים 13](#_Toc106476858)

[נספח 2.3- זמני שירות עבור שירותים נלווים ועליה לרציף 13](#_Toc106476859)

[נספח 2.4- זמני שירות תיקון הרציף 13](#_Toc106476860)

[נספח 2.5- זמני יציאה בסבסבות. 13](#_Toc106476861)

[נספח 3- מבחני טיב התאמה זמני ההגעה של הנוסעים 14](#_Toc106476862)

[נספח 4- הגדרת משאבים וישויות 18](#_Toc106476863)

[נספח 4.1- אתחול הישויות 18](#_Toc106476864)

[נספח 4.2- הגדרת המשאבים 18](#_Toc106476865)

[נספח 4.3- הגדרת הישויות 19](#_Toc106476866)

[נספח 5- חישוב זמן הנסיעה ברכבת 20](#_Toc106476867)

[נספח 6- בדיקות נכונות המודל 22](#_Toc106476868)

[נספח 7- בדיקת דיוק יחסי 24](#_Toc106476869)

[נספח 7.1 24](#_Toc106476870)

[נספח 7.2 25](#_Toc106476871)

[נספח 8- בדיקת משאבים לבחינת חלופות 28](#_Toc106476872)

[נספח 8.1 – בדיקת זמן השירות 28](#_Toc106476873)

[נספח 8.2- בדיקת היציאות העמוסות 36](#_Toc106476874)

[נספח 8.3 – חלוקת זמן במשאבי הרכבת 37](#_Toc106476875)

[נספח 9- השוואת מדדים 37](#_Toc106476876)

[נספח 10 – פונקציות 39](#_Toc106476877)

# תקציר

הפרוייקט עוסק בפעילות רכבת ישראל בארץ, החל מהכניסה לתחנת הרכבת עד יציאת הנוסעים במחוז חפצם. חקרנו פעילות של קו אחד מבין שלושה קווים עיקריים של רכבת ישראל, שעובר בתחנות מרכז הארץ: חדרה, הרצליה, תל אביב, לוד ורחובות, לשני הכיוונים. מטרתנו הייתה לבחון את האפשרויות הקיימות לייעול ושיפור קו הרכבת על מנת לשפר את התדמית של הנוסעים על רכבת ישראל על מנת למקסם את שיעור הנוסעים שבוחרים לנסוע ברכבת. נעזרנו בתוכנת Studio – R על מנת לבצע את חקר המצב ודימינו בעזרתו סימולציה של אחד הקווים בשירותי הרכבת. בעזרת הסימולציה עקבנו אחר מדדים שהגדרנו כחשובים במיוחד כמו: הזמן הממוצע שהנוסע מבלה בתוך התחנה עד שעובר לחכות לרכבת ברציף, הזמן הממוצע שלוקח לנוסע להגיע לתחנת היעד שלו בשלום, ופרופורציית הנוסעים שנוסעים באוטובוס מתוך כלל האנשים שהגיעו לקחת רכבת.

לאחר בניית מודל הסימולציה, ביצענו התאמות ושינויים על מנת לבחון אלטרנטיבות שונות (במגבלות התקציב) היכולות להביא לשיפור החוויה של הנוסע ובכך להגדיל את מספר הנוסעים שבוחרים לנסוע ברכבת ישראל. לבסוף החלטנו להוסיף פסנתרן שינגן בשעת העומס בתחנת תל אביב לשיפור מצב הרוח של העובדים והגברת יעילותם, הוספת סבסבות יציאה בתחנות חדרה ורחובות והגדלת הרציף בלוד על מנת לאשר ליותר נוסעים לנסוע ברכבת. בסופו של דבר, הצלחנו להביא לשיפור במדדים הנ"ל: שיפור הזמן הממוצע שהנוסע מבלה בתחנה לפני העלייה לרציף (12.62 במצב קיים לעומת 12.58 בחלופה 1), שיפור זמן הנסיעה הכולל (244.15 במצב קיים לעומת 242.27 בחלופה 1) והורדה משמעותית של פרופורציית הנוסעים באוטובוס (2.7% במצב קיים לעומת 0.4% בחלופה 1). אנו מאמינים ששיפורים אלה בהחלט יאפשרו לנוסעי רכבת ישראל להשתמש יותר בשירותי הרכבת ולסמוך עליה כמערך הסעות אמין ויעיל.

# מבוא

## תיאור המערכת הנחקרת

המערכת אותה אנו חוקרים היא הפעילות של רכבת ישראל אשר פעילה בין השעות 6:00-23:00 ואנו בחרנו להתמקד בקו הכחול אשר עובר בין התחנות חדרה, הרצליה, תל אביב, לוד ורחובות בין השעות 7:30-13:00 ו15:00-22:00 כל 45 דקות, קו זה הינו קו מתוך שלושה קווים, החוצים את הארץ. חווית הנוסע מתחילה החל ממעמד כניסתו של הלקוח בתחנת המוצא עד ליציאתו מתחנת היעד. התחנות מחולקים ל-2 סוגים, תחנות גדולות ותחנות קטנות, אשר נבדלות בכמות נותני השירות, מקומות העמידה ברציף והמתקנים שיש בכל תחנה [(נספח 1.1).](#_נספח_1.1-_משאבי) התחנות הגדולות בקו הכחול הן תחנת תל אביב ותחנת הרצליה, והתחנות הקטנות הן רחובות, לוד וחדרה. ישנם מספר שלבים אשר כל נוסע בקו הרכבת צריך לעבור:

1. בחירת תחנת עלייה וירידה . כלל הנוסעים המגיעים לרכבת בוחרים את תחנת העלייה והירידה שלו בהסתברויות שונות [(נספח 1.2)](#_נספח_1.2-_הסתברויות).
2. הגעה לתחנה- לתחנת הרכבת מגיעים נוסעים מכל רחבי הארץ על מנת להגיע ליעדם בבטחה ובמהירות. הנוסעים מגיעים לרכבת משעת פתיחת התחנה עד כשעה לפני שעת הסגירה. בהגעה לתחנה על הנוסעים לעבור מספר עמדות :
3. עמדת בידוק ביטחוני – אשר בה כל אדם שמגיע לתחנה עליו לעבור, כאשר חיילים מגיעים לעמדת האבטחה הינם עוקפים את התור במידה וקיים. זמן השירות מפורט בנספח 2.1 [בנספח 2.1.](#_נספח_4.1-_זמני)
4. עמדת קופות – הנוסעים מגיעים לקופות על מנת לרכוש את הכרטיס המתאים עבור נסיעתם . זמן השירות מפורט [בנספח 2.2.](#_נספח_4.2-_זמני)

לאחר קבלת השירות בקופות כ-50% מן הנוסעים ניגשים לקנות קפה ומאפה, 30% מן הנוסעים ניגשים לשירותים ושאר הנוסעים ממשיכים לעבר הרציף. לאחר כל הסידורים כלל הנוסעים מגיעים אל הרציף. זמני השירות והמעבר מפורטים [בנספח 2.3.](#_נספח_4.3-_זמני)

1. הגעה לרציף ונסיעה ברכבת- הנוסעים מגיעים לרציף ומחכים ברציף עד הגעת הרכבת המתאימה עבורם, נוסעים אשר אין להם מקום ברציף יוותרו על הנסיעה ברכבת ויעברו לנסיעה באוטובוס. כאשר הרכבת מגיעה לתחנה היא ממתינה כ5 דקות עד שכלל הנוסעים יעלו. כל רכבת מכילה עד כ80 נוסעים. זמן הנסיעה ברכבת משתנה בין כל נסיעה [(נספח 5)](#_נספח_5-_חישוב). נוסע אשר צריך לעבור מספר תחנות עד ההגעה לתחנת היעד ירד בכל תחנה שהוא עובר ויעלה על הרכבת הבאה לכיוון תחנת היעד שלו. כאשר הנוסע מגיע אל תחנת היעד הוא יוצא מן התחנה דרך הסבסבות. בתחנת תל אביב החיילים הינם יכולים לעקוף את התור לסבסבות זמן היציאה מפורט [בנספח 2.5](#_נספח_4.5-_זמני).
2. תקלות ברכבת – בכל שעה מתקיימות בסיכוי של כ30% תקלות בתחנה אחרת בקו. כאשר ישנה תקלה ברציף על הנוסעים להמתין מחוץ לרציף עד לסיום תיקון התקלה ורק אז יוכלו לעלות על הרציף. זמן השירות מפורט [בנספח 2.4.](#_נספח_7)

## מטרות

* **מטרת הארגון-** היא שיפור תדמית הנוסע על שירותי רכבת ישראל באמצעות שיפור וייעול מערך הרכבות והשירות המקצועי במתקנים הפנימיים בתחנות. הארגון ישאף להיות מערך ההסעות המוביל בישראל מבחינת יעילות ומקסום רווחים.
* **מטרת הפרויקט-** היא בחינה של אלטרנטיבות לייעול הקו הכחול של הרכבת, וזאת ע"י מידול של אלטרנטיבות שונות בתקציב מוגבל ומעקב אחר מדדים רלוונטיים. לבסוף, מתן המלצה על אלטרנטיבה מובילה מבחינת התועלת שלה.
* **מטרות הסימולציה-** הן שיפור המדדים שהגדרנו בהמשך: מדד חווית לקוח, מדד יעילות הרכבת ומדד פספוס .

# מודל סימולציה

## תיאור המודל

על מנת ליצור את הסימולציה של הרכבות השתמשנו בתוכנת R-Studio והשתמשנו בחבילת simmer אשר מאפשר למדל סימולציות במצבים מורכבים.

הסימולציה מתחילה בזמן "0" אשר ממדל לנו את השעה 6:00 ומסתיים בזמן "1020" אשר ממדל לנו את השעה 23:00. את קצב ההגעה של הישויות בדקנו בעזרת מדגם של קצב הגעה של ישויות מסוג אזרחים ומסוג חיילים ביום מסויים. ביצענו על מדגם זה מבחני טיב התאמה [(נספח 3 )](#_נספח_3-_מבחני)  על מנת לבדוק לאיזה התפלגות הינם מתאימים ויצא כי ניתן למדל את התפלגויות ההגעה שלהם על פי התפלגות מעריכית כאשר קצת ההגעה של החיילים הינו עם ושל האזרחים קצב ההגעה הינו . לאחר ניתוח זמני ההגעה יצרנו 2 מחוללי ישויות [(נספח 4.1)](#_נספח_4.2-_אתחול) :

1. חיילים-עם , ומתחילים לייצר את הישויות בזמן "0" הממדל את השעה 6:00 עד "960" אשר ממדל את השעה 22:00.
2. אזרחים-עם , ומתחילים לייצר את הישויות בזמן "0" הממדל את השעה 6:00 עד "960" אשר ממדל את השעה 22:00.

יצרנו משאבים שכוללים את כל המתקנים והשירותים שיש בכל תחנה בנפרד- אבטחה, כרטיסים, עמדת קפה, שירותים, רציף וסבסבות יציאה [(נספח 4.2(.](#_נספח_4.2-_הגדרת) יצרנו 2 סוגי ישויות- אזרחים וחיילים, כל נוסע שהוא אזרח הולך למסלול הראשי שלו בשם start\_citizen וכל חייל הולך למסלול הראשי בשם start\_soldier. במסלולים הראשיים בעצם התכונות של הישות מתאתחלות והיא נשלחת למסלול הרלוונטי לפי תחנת העלייה שהוגרלה לה [(נספח 4.3).](#_נספח_4.3-_הגדרת)

היישות עוברת למסלול של תחנת ההתחלה שלהם מהמסלול הראשי שלה, בכל מסלול היישות מתחילה בעמדת הבידוק הבטחוני ועוברת לעמדת הכרטיסים. לאחר מכן הישות עוברת ל branch אשר בו היא מחליטה לפי ההסתברויות הנתונות האם להמשיך לעבר הרציף במסלול הראשי או לעבור למסלולים של קפה או שירותים ולאחריהם לחזור לנסות לתפוס את הרציף. בתחנות הגדולות איפה שיש בחירה בין קופות הכרטיסים, הישויות יבחרו את התור הקצר ביותר. הרציף הוא משותף גם לקו שנוסע דרומה וגם לקו שנוסע צפונה, ולכן נבדוק את כיוון הנסיעה של הישות (באמצעות תכונה שהיישות קיבלה בתחילת הסימולציה בשם direction) ובאמצעות ניתוב נשלח אותה למסלול עם הכיוון הרלוונטי. הרציף מוגבל במקומות ולכן אם אין מקום ברציף נפנה את היישות למסלול נוסף באמצעות פעולת reject ששם הישות נוסעת באוטובוס.

לאלו שהיה להם מקום ברציף, נשתמש בפונקציית עזר בשם time\_out\_platform שמחשבת את הזמן שכל נוסע מחכה לרכבת שתגיע. (כל יישות מחכה זמן שונה, בהתאם לזמן הגעתה לרציף- זאת נבדוק באמצעות פונקציית now). הסבר על כלל הפונקציות מפורט [בנספח 10.](#_נספח_10_–)

בזמן ההמתנה, ישנו סיכוי של 30% שתופיע תקלה ברציף ועל הנוסעים יהיה להמתין מחוץ לרציף בזמן התיקון. את הניתוב הזה נבצע בעזרת פונקציית עזר בשם malfunction\_in\_the\_train שמגרילה שעת תקלה, תחנת התקלה, והאם קרתה תקלה או לא. אם אכן קרתה תקלה, הנוסעים יחכו את משך זמן תיקון הטכנאי ואז יחזרו לרציף וימשיכו בהמתנתם, ואם לא פשוט יחכו לרכבת שעתידה להגיע.

במסלול עם הכיוון הרלוונטי על מנת לעלות לרכבת נשתמש באיחוד ישויות לישות מאוחדת, עד 80 אנשים, באמצעות שם משותף לכל הנוסעים שיצטרפו לאותה הרכבת. הרכבת תמתין 5 דקות לפני שתצא לדרכה, וחישוב זמן הנסיעה שלה מופיע [בנספח 5.](#_נספח_5-_חישוב)

כלל הנוסעים יורדים אחרי תחנה אחת; תכונתם המקומית בשם "מיקום נוכחי" תתעדכן ותקבל את ערך התחנה בה עצרנו, נפצל את איחוד הישויות ובאמצעות ניתוב נבדוק אם זאת תחנת היעד שלהם – ימשיכו לסבסבות היציאה (נבדוק זאת על ידי שליפת התכונה שהוגרלה בתחילת הסימולציה בשם end ובאמצעות פונקציית עזר check\_if\_last\_station). אם זאת לא תחנת היעד שלהם – הישויות יעברו לרציף של התחנה אליה הגיעו ולאחר מכן יעלו שוב לרכבת באמצעות איחוד חדש של ישויות לישות מאוחדת, עם חוק שהוגדר באמצעות פונקציית עזר was\_on\_train שמעלה את הנוסעים שכבר היו באמצע המסלול שלהם קודם, ולאחר מכן נוסעים חדשים.

## הנחות

1. זמן יציאה בסבסבות - רק בתחנת תל אביב החיילים עוקפים ביציאה מהתחנה, בכל מקום אחר משך זמן היציאה של כל הישויות מתפלג באופן זהה. בנוסף, בכל עקיפה אין עצירת מתן שירות למי שכרגע תופס את המשאב, אלא עקיפה של התור בלבד.
2. בהגרלת תחנת עלייה וירידה, נמשיך להגריל עד שיצאו תחנות שונות. לפי ההיגיון תחנת המוצא ותחנת היעד אמורות להיות תחנות שונות אחרת אין טעם בשירות.
3. הרציף הוא אותו רציף מצפון לדרום ומדרום לצפון .
4. כל נוסע אשר מגיע לרכבת הינו מעדיף נסיעה ברכבת מאשר נסיעה באוטובוס .
5. כאשר ישות נמצאת במשאב רציף (הצליחה לתפוס אותו) ולא הצליחה לתפוס את הרכבת היא איננה תעזוב את הסימולציה עד אשר היא תצליח לתפוס את הרכבת הבאה.
6. כל הנוסעים יורדים אחרי תחנה אחת. נוסע שזאת לא תחנת היעד שלו עולה שוב פעם לרכבת אחרת במסלול אחר לכיוון של תחנת היעד. עשינו פונקציה שתוודא שיהיה לו מקום ברציף וכלל שקובע שמי שהיה באמצע הדרך יעלה לרכבת קודם.
7. בשביל להשתמש במבחן t-מזווג הנחנו את ההנחות הבאות:
8. יש תלות בין ריצות מקבילות בשלושת החלופות - ההנחה מתקיימת בזכות שה- seed זהה לכל חלופה.
9. אין תלות בין ריצות בכל חלופה בפני עצמה - כל ריצה מקבלת seed שונה.
10. ניתן להניח שוויון שונויות בין הסדרות - ה- seed קבוע בתחילת הסימולציה ועולה באותו הקצב ולכן ניתן להניח שוויון שונויות.
11. המדדים מתפלגים נורמלית – כפי שלמדנו ניתן להניח כי מדדים מסוג ממוצע ופרופורציה

# ניתוח מצב קיים והצעת חלופות לשיפור מצב קיים

## בחירת מדדים לניתוח

תחילה ביצענו בדיקות נכונות המודל שנמצא תקין. [נספח 6](#_נספח_6-_בדיקות)

**בחירת מדדים**

מדד חווית לקוח – בהתאם למטרת הארגון, שמנו דגש על שיפור תדמית הנוסע על איכות השירות ברכבת שנמדדת בין היתר גם במשך זמן השירות.

נרצה להעצים את חווייתו של הנוסע על ידי מזעור משך הזמן שבו הנוסע ממתין לקבלת שירות במתקני הרכבת. למדידת מדד זה נבצע חישוב **של ממוצע משך הזמן בתוך התחנה**, כלומר:

ככל שהמדד יהיה נמוך יותר > איכות השירות תהיה גבוהה יותר, שכן הנוסע יודע שהוא לא צריך להיערך ולהקדים יותר מדיי זמן לפני שעת הגעת הרכבת כי השירות הינו יעיל וזריז.

מדד יעילות הרכבת – נוסע יבחר שוב ושוב להשתמש בשירותי הרכבת רק במידה והרכבת תתעלה על האלטרנטיבות האחרות (נסיעה באוטובוס, מונית, רכב). נרצה לחשב **מה אורך הזמן הממוצע שלוקח לכל נוסע** להגיע לתחנת היעד שלו ולוודא שהממוצע לא גבוה מדיי, שאין תורים עמוסים וצווארי בקבוק במערכת.

מדד פספוס – לעיתים, בעקבות הגבלת מקום ברציף או ברכבת ישנם נוסעים שעוברים לאוטובוס לאחר שעברו את כל המסלול בתחנת המוצא אך לא הצליחו לתפוס רכבת. דבר כזה יכול לפגוע אנושות בתדמית וברווחיות של שירות הרכבות, שכן אנשים לא ירצו לחזור ולבזבז זמן בהמתנה רק כדי לגלות שבסוף יצטרכו לנסוע באוטובוס. לכן, נרצה לחשב מדד שיבדוק מהי כמות האנשים שנאלצים לקחת אוטובוס ונרצה למזערו ככל שניתן. אחת ההנחות שהסתמכנו עליה היא שכל לקוח שלוקח אוטובוס כאשר בא לקחת רכבת- אינו מרוצה. את המדד נחשב כך:

## סוג המערכת

סוג המערכת בסימולציה הינה היא מערכת מסתיימת בגלל שהמערכת מוגבלת בזמן בכל יום פעילות בין השעות 6:00 ל23:00, ולכן אין זמן חימום. זמן הריצה של הסימולציה הוא 1020 דקות (משך זמן פעילות תחנת הרכבת ביום).

## ניתוח מצב קיים ביחס למדדים ולמטרות

מספר הריצות הראשוניות של המודל היה עם נתונים של ו- . כדי שנוכל לבחון שלושה מדדים, השתמשנו באי שוויון בונפורני כדי לחשב אלפא יחסית ומצאנו שיש להשתמש ב

: = 0.03. מכאן שרמת הביטחון היא 0.97 .

על מנת לבדוק מהו מספר הריצות הנדרש למצב הקיים ולשתי החלופות, ע"פ כל מדד, בדקנו האם עומד בחסם של הדיוק היחסי ע"פ הנוסחה: . החסם העליון הוא .

להלן תוצאות הדיוק היחסי:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **מדד חווית לקוח** | **מדד יעילות הרכבת** | **מדד פספוס** |
| **מצב קיים** | 0.0007 | 0.0380 | 0.00438 |
| **חלופה 1** | 0.0008 | 0.0254 | 0.00209 |
| **חלופה 2** | 0.0007 | 0.0363 | 0.00436 |

כפי שניתן לראות מהטבלה, עבור מדד יעילות הרכבת נמצאה חריגה הן עבור המצב הקיים והן עבור חלופה 2. על מנת לדעת את מספר הריצות הכולל הדרוש הצבנו את הנתונים בנוסחה הבאה:

עבור מצב קיים התקבל 25.6 𝑛 = ואילו עבור חלופה 2 התקבל 23.3 𝑛 = לכן נבחר את הגבוה מבין השניים ונאמר כי דרושות בסך הכל 26 ריצות[.(נספח 7.1)](#_נספח_7.1)

לאחר הרצת הריצות הנוספות שנדרשו, נמצא כי מספר ההרצות החדש שביצענו עדיין לא מספיק עבור רמת המובהקות ורמת הדיוק היחסי שהגדרנו קודם לכן, ולאחר הרצנו פעמיים נוספות, פעם עם n=35 ולאחר מכן עם n=36. את הפירוט המלא של החישובים ניתן לראות [נספח 7.2.](#_נספח_7.2)

|  |  |  |
| --- | --- | --- |
|  | **ממוצע** | **סטיית תקן** |
| **מדד חווית לקוח** | 12.625809 | 0.122639 |
| **מדד יעילות רכבת** | 244.1519 | 4.17938 |
| **מדד פספוס** | 0.0272477 | 0.0133291 |

להלן תוצאות המצב הקיים לאחר 36 ריצות:

**מדד חווית הלקוח** שמתאר את ממוצע הזמן שנוסע מבלה בתוך התחנה עומד על 12.62 דקות. בזמן הזה הנוסע עובר בעמדת האבטחה, קונה כרטיס בעמדה ומשתמש בשירותים/ קונה קפה. משמעות הדבר שנוסע צריך לקחת בחשבון שעליו להקדים לרכבת לפחות ברבע שעה לפני שעת יציאת הרכבת על מנת להספיק את כל הסידורים שלפני. נרצה למזער את המדד הזה כל הניתן כדי שתדמית הנוסע על השימוש ברכבת תשתפר והרכבת תיתפס כאלטרנטיבה העדיפה עליו מפאת יעילותה ומזורזת.

**מדד יעילות הרכבת** שמתאר את משך הזמן הממוצע שלוקח לנוסע להגיע מתחנת המוצא עד ליציאה מתחנת היעד עומד על 244.15 דקות. בפרויקט זה לא נעסוק בייעול מערך הקווים והתכיפות בהם הרכבת עוברת ולכן אין לנו השפעה על זמן הנסיעה ברכבת וההמתנה ברציפים. נרצה למזער את המדד הזה באמצעות שיפורים שנציע בהמשך במסגרת המשאבים שכן ברי השפעה על ידנו.

**מדד פספוס** שמתאר את פרופורציית מספר האנשים שלוקחים אוטובוס לאחר שהגיעו לקחת רכבת מתוך כל האנשים שהגיעו לקחת רכבת עומד על 2.72%. את המדד הזה נרצה להקטין עד כדי 0% כי בחרנו להתייחס לכל נוסע כזה כנוסע מפוספס שלא יחזור שנית לשירותי רכבת ישראל לאחר שהתאכזב והמתין לשווא. כל אחוז גדול מ-0 פוגע ברווחיות ובתדמית הרכבת.

## תיאור החלופות

**חלופה 1:**

|  |  |  |
| --- | --- | --- |
| **שיפור** | **תוכן** | **עלות** |
| נגינה על פסנתר בתחנה ספציפית למשך שעה אחת במהלך היום. | נגינה על פסנתר בתחנת תל אביב למשך שעה אחת בין השעות 13:30-14:30 | 16,000 |
| הוספת סבסבות יציאה | הוספת 2 סבסבות יציאה בתחנות חדרה וברחובות | 20,000 |
| הוספת מקומות ישיבה להמתנה ברציף | הוספת 20 מקומות ישיבה ברציף בתחנת לוד | 12,000 |
| סך הכל |  | 48,000 |

בחלופה זו ניסינו לשפר את כלל המדדים. נסביר:

בחרנו בנגינה על פסנתר דווקא בתחנת תל אביב בשעת העומס לאחר שבדקנו מהי שעת העומס ובאיזה תחנה ([נספח 8.1),](#_נספח_8.1_–) וכך נשפר את מדד חווית הלקוח (מצב הרוח של עובדי התחנה משתפר מה שצפוי לצמצם את זמן השהות בתחנה ב-20%). החלטנו להוסיף 2 סבסבות ביציאה כדי לשפר את מדד יעילות הרכבת (הנסיעה ברכבת לא מאלצת אותו להישאר מעבר לזמן שצריך בתחנה לאחר שהגיע). הוספת מקומות ישיבה להמתנה ברציף צפוי למזער את מדד הפספוס של הלקוחות הפוטנציאלים שלא יחזרו בעתיד כי ליותר לקוחות צפוי להיות מקום והם לא יאלצו לקחת אוטובוס במקום לנסוע ברכבת, וכך יישארו מרוצים.

**תוצאות המדדים לאחר מימוש השיפורים המוצעים בחלופה 1 הן:**

|  |  |  |
| --- | --- | --- |
|  | **ממוצע** | **סטיית תקן** |
| **מדד חווית לקוח** | 12.584446 | 0.10427641 |
| **מדד יעילות רכבת** | 242.27029 | 4.233129 |
| **מדד פספוס** | 0.0044566 | 0.00370211 |

**חלופה 2:**

|  |  |  |
| --- | --- | --- |
| **שיפור** | **תוכן** | **עלות** |
| הוספת עובד לתחנות בקו | הוספת בריסטה לעמדת הקפה בתחנת רחובות | 13,000 |
| הוספת סבסבות יציאה | הוספת 2 סבסבות יציאה בתחנות חדרה ובהרצליה | 20,000 |
| קיצור הפסקת הקו | קיצור ההפסקה של הרכבות בשעה אחת | 15,000 |
| סך הכל |  | 48,000 |

בחלופה זו ניסינו לשים דגש על מדד יעילות הרכבת. השתמשנו בשיפור של קיצור הפסקת הקו (יותר רכבות, משרת יותר נוסעים) והוספת סבסבות [(נספח 8.2)](#_נספח_8.2-_בדיקת) מה שצפוי לקצר את הזמן של הנסיעה הכוללת ברכבת, ובכך לגרום לתדמית של נסיעה ברכבת בעיניי הנוסע להיות חיובית יותר כי החוויה קצרה יותר. קיצור הקו יתרום גם למדד הפספוס כאשר פחות נוסעים ילכו לאוטובוס. בנוסף, הוספת עובד לתחנות הקפה תשפר את מדד חווית הלקוח בכך שמשך הזמן באחד התורים בתוך התחנה יתקצר, מה שיכול גם לגרום לעלייה ברווחים של רכבת ישראל (אחת ממטרות הארגון) מפני שלרוב תורים מרתיעים לקוחות שמתלבטים האם לרכוש או לא [(נספח 8.3)](#_נספח_8.3_–)

**תוצאות המדדים לאחר מימוש השיפורים המוצעים בחלופה 12 הן:**

|  |  |  |
| --- | --- | --- |
|  | **ממוצע** | **סטיית תקן** |
| **מדד חווית לקוח** | 12.6256044 | 0.122886 |
| **מדד יעילות רכבת** | 243.067203 | 4.056635 |
| **מדד פספוס** | 0.02707528 | 0.013244 |

ניתן לראות שיפור בכל המדדים, אמנם במדד חווית הלקוח ומדד הפספוס השינויים לא גבוהים כך שניתן לומר שבמסגרת השיפורים שהצענו המדד לא השתפר ולא הורע.

## השוואה סטטיסטית בין המצב הקיים לחלופות המוצעות

בתום בחינת המצב הקיים ושתי החלופות ע"פ שלושת המדדים, כעת יש לקבוע איזו חלופה היא הטובה ביותר. לצורך כך בצענו מבחן t מזווג ל-9 השוואות שונות ברמת מובהקות 0.01 =

ההשוואות הן על פי החלופות ועבור כל מדד[.(נספח 9)](#_נספח_9-_השוואת)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **סיכום עבור כל מדד** | **תוצאת המבחן** | **גבול תחתון רווח סמך** | **גבול עליון רווח סמך** | **השוואה** | **מדד** |
| **אדישות** | אדישות | -0.006123054 | 0.88812137 | מצב קיים לחלופה 1 | **מדד חווית הלקוח** |
| אדישות | -0.000187615 | 0.000597889 | מצב קיים לחלופה 2 |
| אדישות | -0.08857441 | 0.006295601 | חלופה 1 לחלופה 2 |
| **חלופה 1/ חלופה 2** | חלופה 1 | 0.466119849 | 3.297165208 | מצב קיים לחלופה 1 | **מדד יעילות הרכבת** |
| חלופה 2 | 0.935986966 | 1.233483598 | מצב קיים לחלופה 2 |
| אדישות | -2.236425753 | 0.64261126 | חלופה 1 לחלופה 2 |
| **חלופה 1** | חלופה 1 | 0.018703708 | 0.031334748 | מצב קיים לחלופה 1 | **מדד הפספוס** |
| חלופה 2 | 0.000134014 | 0.000210528 | מצב קיים לחלופה 2 |
| חלופה 1 | -0.031125405 | -0.018568508 | חלופה 1 לחלופה 2 |

בטבלה זו ניתן לראות כי עבור מדד חווית הלקוח לא ניתן לקבוע באופן חד משמעי איזו חלופה עדיפה. במדד יעילות הרכבת גם כן לא נוכל לקבוע כי אין חלופה שעדיפה על רוב המדדים. במדד הפספוס נעדיף את חלופה 1 בבירור.

# מסקנות

מטרת הפרויקט בראש ובראשונה שיפור תדמית הנוסע על שירותי רכבת ישראל באמצעות שיפור וייעול מערך הרכבות והשירות המקצועי במתקנים הפנימיים בתחנות. התוצאות שהתקבלו במחקר מראות כי עבור מדד חווית הלקוח קיימת אדישות בין החלופות למצב הקיים לעומת מדד יעילות הרכבת ששם ניתן לומר בוודאות שהחלופות עדיפות על המצב הקיים אבל לא עדיפות אחת על השנייה, ולפי מדד הפספוס נראה כי חלופה 1 עדיפה מול כל ההשוואות. נציין כי עלות החלופות זהה. אי לכך ובהתאם לזאת, נמליץ לרכבת ישראל לממש את חלופה 1 מפני שחלופה זו משפיעה לחיוב **על 2 מדדים, גם על מדד יעילות הרכבת וגם על מדד הפספוס, לעומת חלופה 2 שעשויה להשפיע לחיוב רק על מדד יעילות הרכבת. יישום נכון של ההמלצה שלנו יכלול הוספת בריסטה בתחנת רחובות, הוספת 2 סבסבות יציאה בתחנת חדרה וברחובות והגדלת רציף לוד.** במידה ויהיה ניתן להגדיל את תקציב, נמליץ להוסיף בריסטה לכל שעות היום בעמדות הקפה ברחובות ובכך גם למקסם רווחים. לבסוף, מטרת הפרוייקט הייתה מתן המלצה על אלטרנטיבה מובילה מבחינת התועלת שלה -והיא הושגה **בהצלחה.**

# נספחים

## נספח 1- משאבי התחנה

### נספח 1.1- משאבי התחנה

|  |  |  |
| --- | --- | --- |
|  | תחנה גדולה | תחנה קטנה |
| מספר קופות | **2** | **1** |
| מספר מוכרים | **8** | **6** |
| מספר תאי שירותים | **15** | **10** |
| מספר סבסבות יציאה | **3** | **2** |
| מספר עמדות אבטחה | 3 | 3 |
| עובדים בבית קפה | 4 | 4 |
| תגבור בריסטות בשעות  לחץ | **2** | **1** |

ההבדלים בין התחנות הגדולות והקטנות מפורטים בטבלה:

### נספח 1.2- הסתברויות לבחירת קווי עליה וירידה

תחנות העלייה והירידה מוגרלות בהסתברויות המפורטות מטה. במידה ומוגרלת אותה התחנה כתחנת עלייה וירידה, מתבצעת הגרלה נוספת עד שיהיו תחנות שונות.

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטיקו צפון > דרום קו צפון > דרום

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | תחנת עלייה | תחנת ירידה |
| חדרה | 24% | 25% |
| הרצליה | 10% | 33% |
| תל אביב | 37% | 12% |
| לוד | 9% | 10% |
| רחובות | 20% | 20% |

## נספח 2- זמני שירות

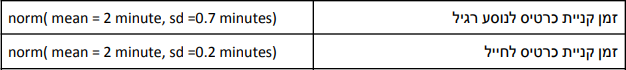
### נספח 2.1- זמני שירות עבור עמדת אבטחה

החיילים משתמשים מדי יום בשירותי הרכבת, על מנת להקל על התורים מספיקה הצגת חוגר שמקנה להם אפשרות לעקוף את כל הממתינים ולהיכנס מיידית לתחנה. זמני המעבר מתפלגים אחיד רציף עם פרמרטרים שונים כפי שמפורט בטבלה.



### נספח 2.2- זמני שירות עבור קניית כרטיסים

זמני רכישת הכרטיס מתפלגים נורמלית עם פרמטרים שונים, כפי שמפורט בטבלה להלן.



### נספח 2.3- זמני שירות עבור שירותים נלווים ועליה לרציף

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטיזמני השהייה בכל אחד מהמתקנים בתחנה מתפלגים נורמלית וזמן העלייה לרציף מפולג אחיד רציף. יש שידלגו על המתקנים ויש כאלה שזמן שהותם בתחנה יכלול גם שימוש במתקנים (שירותים/ קפה) וגם זמן עלייה לרציף.

### נספח 2.4- זמני שירות תיקון הרציף

הנוסעים יחכו מחוץ לרציף משך משך זמן המתפלג נורמלית עד שיוכלו לחזור חזרה לרציף ולחכות לרכבת, זאת במידה ואכן הייתה תקלה בתחנה שבה נמצאו.



### נספח 2.5- זמני יציאה בסבסבות.

חיילים אשר יורדים בתחנת **תל אביב**, עוקפים את כל התור ויוצאים ראשונים.

בכל שאר התחנות וסוגי הנוסעים, משך זמן המעבר ביציאה מהתחנה מתפלג בהתפלגות משולשית כמפורט מטה.



## נספח 3- מבחני טיב התאמה זמני ההגעה של הנוסעים

החיילים מגיעים בזמן המתפלג מעריכית בקצב של דקות.

האזרחים מגיעים בזמן המתפלג מעריכית בקצב של דקות. ניתן לראות את החישוב כאן:

Rmarkdown Arrival Distribution Analysis

library(fitdistrplus)

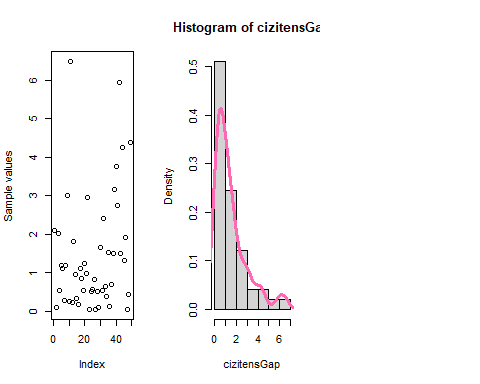
library(magrittr)R Markdown

First we will load the data:

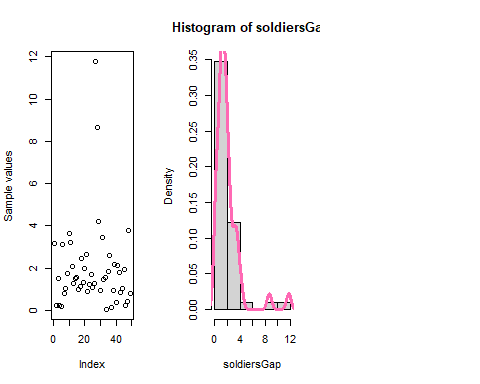
filePath=choose.files()  
table <-read.csv(filePath,header=T)  
  
arrivalTimes <- subset(table[2:50,3:4]) #gets the data from the excel  
  
cizitensGap <- arrivalTimes[,1] #The gaps between the arrival times of citizens  
soldiersGap <- arrivalTimes[,2] #The gaps between the arrival times of soldiers

At first we will look at the data and understand - the data range, continous, are there any outliers? For that we can use the following plots:

par(mfrow=c(1,3)) # plot of citizen   
plot(cizitensGap,ylab="Sample values") # plot ( left graph )  
hist(cizitensGap, prob=TRUE)  
lines(density(cizitensGap), col = "hotpink", lwd = 3)  
  
  
par(mfrow=c(1,3)) # plot of soldiers



plot(soldiersGap,ylab="Sample values") # plot ( left graph )  
hist(soldiersGap, prob=TRUE)  
lines(density(soldiersGap), col = "hotpink", lwd = 3)



for citizen

citizensExpFit <- fitdist(cizitensGap, distr = "exp")  
 citizensunifFit <- fitdist(cizitensGap, distr = "unif")  
  
 citiznesNormFit <- fitdist(cizitensGap, distr = "norm")  
 summary(citizensExpFit)

## Fitting of the distribution ' exp ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## rate 0.691058 0.09872237  
## Loglikelihood: -67.10704 AIC: 136.2141 BIC: 138.1059

summary(citiznesNormFit)

## Fitting of the distribution ' norm ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## mean 1.447057 0.2112383  
## sd 1.478668 0.1493677  
## Loglikelihood: -88.69393 AIC: 181.3879 BIC: 185.1715   
## Correlation matrix:  
## mean sd  
## mean 1 0  
## sd 0 1

for soldiers

soldiersExpFit <- fitdist(soldiersGap, distr = "exp")  
 soldiersNormFit <- fitdist(soldiersGap, distr = "norm")  
 summary(soldiersExpFit)

## Fitting of the distribution ' exp ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## rate 0.5132449 0.07332042  
## Loglikelihood: -81.68311 AIC: 165.3662 BIC: 167.258

summary(soldiersNormFit)

## Fitting of the distribution ' norm ' by maximum likelihood   
## Parameters :   
## estimate Std. Error  
## mean 1.948388 0.2869591  
## sd 2.008713 0.2029105  
## Loglikelihood: -103.7052 AIC: 211.4104 BIC: 215.1941   
## Correlation matrix:  
## mean sd  
## mean 1 0  
## sd 0 1

Comparing the maximum likelihood of the normal and exponential distribution, we can see that both the aic and bic are lower in the exponential threrefore we will say that the exponential distribution are more fittted to our data. we can see that the maximum likelihood Parameter is -67.10704 for the citizen and -81.68311 for the soldiers.

citiznesExpGof<- gofstat(citizensExpFit)%>%print()

## Goodness-of-fit statistics  
## 1-mle-exp  
## Kolmogorov-Smirnov statistic 0.06111912  
## Cramer-von Mises statistic 0.02117294  
## Anderson-Darling statistic 0.16361993  
##   
## Goodness-of-fit criteria  
## 1-mle-exp  
## Akaike's Information Criterion 136.2141  
## Bayesian Information Criterion 138.1059

citiznesExpGof$kstest%>%print()

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

soldiersExpGof<- gofstat(soldiersExpFit)%>%print()

## Goodness-of-fit statistics  
## 1-mle-exp  
## Kolmogorov-Smirnov statistic 0.1713391  
## Cramer-von Mises statistic 0.2302899  
## Anderson-Darling statistic 1.2815863  
##   
## Goodness-of-fit criteria  
## 1-mle-exp  
## Akaike's Information Criterion 165.3662  
## Bayesian Information Criterion 167.2580

soldiersExpGof$kstest%>%print()

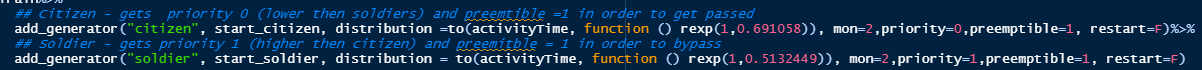
## 1-mle-exp   
## "not rejected"

we executed KS test and we can see that we will not reject H0.

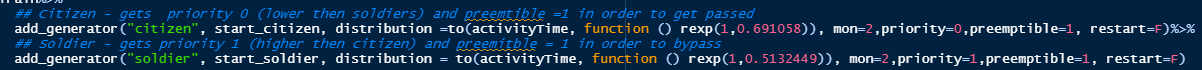
## נספח 4- הגדרת משאבים וישויות

### נספח 4.1- אתחול הישויות

עבור האזרחים :



עבור החיילים:



### נספח 4.2- הגדרת המשאבים

כל משאב מאופיין אחרת בהתאם לגודלה של התחנה.

1. כרטיסים: 2 עמדות בהרצליה ו-2 עמדות בת"א, עם capacity=4 בכל אחת בהתאם למספר הקופאים. בחדרה, לוד ורחובות יש עמדה אחת עם capacity=6.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1. שירותים: הרצליה ות"א - 15 capacity=, חדרה, לוד ורחובות -capacity=10.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1. סבסבות יציאה: הרצליה ות"א- capacity=3, חרדה לוד ורחובות 2 capacity=

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1. עמדות קפה: הרצליה ות"א- capacity= coffe\_big\_schedule (תזמון ערכים- תגבור בריסטה בתחנות הגדולות בשעות העומס בין 08:00-12:00) עולה מ-4 בריסטות ל-6.

חדרה, לוד ורחובות- capacity= coffe\_small\_schedule (תזמון ערכים- תגבור בריסטה בתחנות הקטנות בשעות העומס בין 08:00-12:00) עולה מ-4 בריסטות ל-5.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי



1. עמדות אבטחה: בכל התחנות capacity=3

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1. תמונה שמכילה טקסט

   התיאור נוצר באופן אוטומטירציפים: הרצליה ות"א: capacity=100, חדרה, לוד, רחובות capacity =50

בכל המשאבים הגדרנו תור אינסופי פרט למשאב הרציפים בהם התור הוא 0.

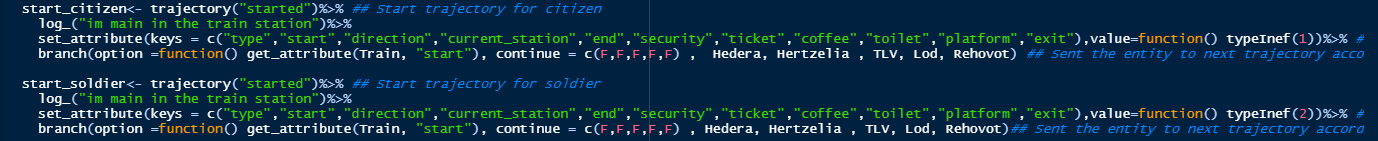
### נספח 4.3- הגדרת הישויות

**תכונות הישויות הינם:**

סוג הישות- על מנת להגדיר את תכונות הנוסע :

1. סוג הנוסע- חייל או אזרח
2. תחנת עלייה – נקבע על ידי חישוב לפי ההסתברויות של הבחירה בכל תחנה .
3. תחנת סיום – נקבע על ידי חישוב לפי ההסתברויות של הבחירה בכל תחנה.
4. כיוון- נקבע על ידי תחנת העלייה ותחנת הירידה- במידה והנוסע נוסע לכיוון צפון אז הוא מקבל את הערך 1 ובמידה והוא נוסע לכיוון דרום הוא מקבל את הערך 2.
5. מיקום נוכחי- נקבע תחילה על ידי תחנת ההתחלה.
6. משך זמן האבטחה- על פי התפלגויות אשר הסברנו בנספח 2.1
7. משך זמן רכישת כרטיס- על פי התפלגויות אשר הסברנו בנספח 2.2
8. משך זמן בעמדת הקפה- על פי התפלגויות אשר הסברנו בנספח 2.3
9. משך זמן בשירותים- על פי התפלגויות אשר הסברנו בנספח 2.3
10. משך זמן עלייה לרציף- על פי התפלגויות אשר הסברנו בנספח 2.3
11. משך זמן בסבסבת היציאה- על פי התפלגויות אשר הסברנו בנספח 2.5

שלחנו את הנוסעים למסלול ראשוני של start\_citizen אשר מקבל את הערך 1 ונשלח לפונקציה typeInf אשר מחזיר את התכונות לפי ההתפלגויות המתאימות ואת הנוסעים מסוג חייל שלחנו למסלול ראשוני של start\_solder אשר מקבל את הערך 2 ונשלח לפונקציה typeInf אשר מחזיר את התכונות לפי ההתפלגויות המתאימות.



מצורף צילום מסך של הפונקציה לפיה היא מחזירה את התכונות : תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

## נספח 5- חישוב זמן הנסיעה ברכבת

**X הינו משתנה מקרי המוגדר:**

תחום 1 :

הסיכוי ליפול בתחום *:*

תחום 2 :

הסיכוי ליפול בתחום *:*

***כעט נגדיר את פונקציה המקורית שלנו כסכום פונקציות לפי שיטת הקומפוזיציה***

*תחום 1- :*

*נבדוק עבור התחומים שלנו :*

*כאשר*

*בתחום :*

*מחוץ לתחום:*

*כאשר*

*בתחום :*

*מחוץ לתחום:*

*מכאן ניתן לראות כי האלגוריתם כאשר התחום שלנו :*

תחום 2- :

*נבדוק עבור התחומים שלנו :*

*כאשר*

*כאשר*

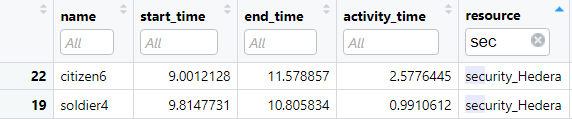
*ניתן לראות כי בשני התחומים של U אנו נמצאים בתחומים של X.*

***האלגוריתם דגימה הינו :***

1. *דגום*
2. *אם החזר אחרת החזר*

## נספח 6- בדיקות נכונות המודל

בדיקה ראשונה – חיילים עוקפים אזרחים בבידוק הבטחוני.



ניתן לראות שהחייל הגיע אחרי האזרח אבל עקף את האזרח.

**בדיקת שניה - ישות שהגיעה לרציף כאשר הוא מלא הוסטה למסלול האוטובוס.**

* ראשית, בדקנו באילו שעות ובאיזה תחנות הרציף הגיע לתפוסה מלאה. החלטנו להתמקד ברציף של תחנת תל אביב בין הזמנים 966.32 ל-971.83, שם הרציף הגיע לתפוסה מלאה.

תמונה שמכילה טקסט, אלקטרוניקה

התיאור נוצר באופן אוטומטי

* לאחר מכן, חקרנו את הנתונים על מנת למצוא ישויות אשר נטשו לאוטובוס בשעות הנ"ל ומצאנו את אזרח 664: ניתן לראות שהישות מקבלת את כל התכונות בזמן שהיא נוצרה ולאחר כ18 דקות - הגיעה לאוטובוס.

תמונה שמכילה טקסט, לוח תוצאות, רהיטים, ארון

התיאור נוצר באופן אוטומטי

* עקבנו אחרי אזרח 644 בטבלת הAttribute בכדי לראות האם אכן ניסה לתפוס את הרציף בזמנים המתוארים למעלה, נכשל ועזב לאוטובוס

תמונה שמכילה טקסט, לוח תוצאות, צילום מסך

התיאור נוצר באופן אוטומטי

כפי שניתן לראות בטבלה השנייה - זמן העלייה לרציף של אזרח 644 הוא 13.975 דקות. היישות סיימה את קניית הכרטיסים בזמן 955.2826 ולאחר מכן הייתה אמורה לעלות לרציף במשיך 13.975 דקות ואז לנסות לתפוס את משאב הרציף בזמן 969.2575.

כפי שניתן לראות, זהו הזמן בו היישות הוסטה למסלול האוטובוס והועברה ליציאה מתחנת תל אביב.

מה שמוכיח שכאשר משאב הרציף מלא, היישות מוסטת לאוטובוס ויוצאת מהתחנה.

בדיקה שלישית- לוודא שישות לא מקבלת תחנת התחלה ותחנת סיום זהות.

שאילתת הSQL הנ"ל מציגה ישויות אשר קיבלו ערך זהה בתחנת ההתחלה ותחנת הסיום.

כפי שניתן לראות, השאילתא מחזירה טבלה ריקה, מה שמעיד על כך שאין ישויות כאלה.

Check\_if\_start\_equals\_end <- sqldf("select \* from (select \*

from attributeData as a join attributeData as b on a.name=b.name

where a.key = 'start' and b.key='end') as c

where c.value='value:1'

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי ("

## נספח 7- בדיקת דיוק יחסי

פונקציה אשר בודקת את הדיוק היחסי :

Accuracy <- function(t, n, v, avr){

delta <- t\*(v/(sqrt(n)))

return(delta/avr)

}

### נספח 7.1

חיפוש ראשוני של מספר הריצות על ידי בדיקת דיוק ראשוני כאשר

tableValues <- c(Accuracy(t,n0,(madad\_1\_STDEV)^2, madad\_1\_Mean),Accuracy(t,n0,(madad2\_STDEV)^2, madad2\_Mean),Accuracy(t,n0,(madad\_3\_stdev)^2, madad\_3\_mean),

Accuracy(t,n0,(madad\_1\_STDEV\_a1)^2, madad\_1\_Mean\_a1),Accuracy(t,n0,(madad2\_STDEV\_a1)^2, madad2\_Mean\_a1),Accuracy(t,n0,(madad\_3\_stdev\_a1)^2, madad\_3\_mean\_a1),

Accuracy(t,n0,(madad\_1\_STDEV\_a2)^2, madad\_1\_Mean\_a2),Accuracy(t,n0,(madad2\_STDEV\_a2)^2, madad2\_Mean\_a2),Accuracy(t,n0,(madad\_3\_stdev\_a2)^2, madad\_3\_mean\_a2))

checkAccuracy <- matrix(tableValues,ncol=3,byrow=TRUE)

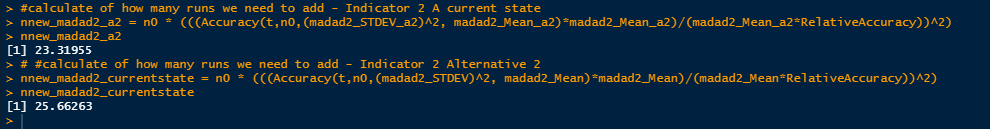
colnames(checkAccuracy) <- c("Madad1", "madad2", "madad3")

rownames(checkAccuracy) <- c("Current state","Alternative1","Alternative2")

checkAccuracy <- as.table(checkAccuracy)

checkAccuracy

מספר הריצות הדרוש כדי לשפר את הדיוק היחסי לאחר 15 ריצות ראשוניות.



#calculate of how many runs we need to add - Indicator 2 A current state

nnew\_madad2\_a2 = n0 \* (((Accuracy(t,n0,(madad2\_STDEV\_a2)^2, madad2\_Mean\_a2)\*madad2\_Mean\_a2)/(madad2\_Mean\_a2\*RelativeAccuracy))^2)

nnew\_madad2\_a2

# #calculate of how many runs we need to add - Indicator 2 Alternative 2

nnew\_madad2\_currentstate = n0 \* (((Accuracy(t,n0,(madad2\_STDEV)^2, madad2\_Mean)\*madad2\_Mean)/(madad2\_Mean\*RelativeAccuracy))^2)

nnew\_madad2\_currentstate

### נספח 7.2

בדיקת מספר הריצות כאשר

n1 <- 26

t1 <- qt(0.985,(n1-1))

tableValues1 <- c(Accuracy(t1,n1,(madad\_1\_STDEV)^2, madad\_1\_Mean),Accuracy(t1,n1,(madad2\_STDEV)^2, madad2\_Mean),Accuracy(t1,n1,(madad\_3\_stdev)^2, madad\_3\_mean),

Accuracy(t1,n1,(madad\_1\_STDEV\_a1)^2, madad\_1\_Mean\_a1),Accuracy(t1,n1,(madad2\_STDEV\_a1)^2, madad2\_Mean\_a1),Accuracy(t1,n1,(madad\_3\_stdev\_a1)^2, madad\_3\_mean\_a1),

Accuracy(t1,n1,(madad\_1\_STDEV\_a2)^2, madad\_1\_Mean\_a2),Accuracy(t1,n1,(madad2\_STDEV\_a2)^2, madad2\_Mean\_a2),Accuracy(t1,n1,(madad\_3\_stdev\_a2)^2, madad\_3\_mean\_a2))

checkAccuracy1 <- matrix(tableValues1,ncol=3,byrow=TRUE)

colnames(checkAccuracy1)<- c("Madad1", "madad2", "madad3")

rownames(checkAccuracy1) <- c("Current state","Alternative1","Alternative2")

checkAccuracy1 <- as.table(checkAccuracy1)

checkAccuracy1

לאחר הוספת הריצות הנוספות עד לn=26 קיבלו את טבלת הדיוק היחסי הבאה:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **מדד חווית לקוח** | **מדד יעילות הרכבת** | **מדד פספוס** |
| **מצב קיים** | 0.0005 | 0.0337 | 0.00251 |
| **חלופה 1** | 0.0004 | 0.0302 | 0.00139 |
| **חלופה 2** | 0.0005 | 0.0320 | 0.00249 |

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיניתן לראות שעדיין מדד 2 לא עומד בדיוק ויש לעשות ריצות נוספות:

# #calculate of how many runs we need to add - Indicator 2 Alternative 2

nnew\_madad2\_a21 = n1 \* (((Accuracy(t1,n1,(madad2\_STDEV\_a2)^2, madad2\_Mean\_a2)\*madad2\_Mean\_a2)/(madad2\_Mean\_a2\*RelativeAccuracy))^2)

nnew\_madad2\_a21

#calculate of how many runs we need to add - Indicator 2 A current state

nnew\_madad2\_currentstate1 = n1 \* (((Accuracy(t1,n1,(madad2\_STDEV)^2, madad2\_Mean)\*madad2\_Mean)/(madad2\_Mean\*RelativeAccuracy))^2)

nnew\_madad2\_currentstate1

# #calculate of how many runs we need to add - Indicator 2 Alternative 1

nnew\_madad2\_currentstate1 = n1 \* (((Accuracy(t1,n1,(madad2\_STDEV)^2, madad2\_Mean)\*madad2\_Mean)/(madad2\_Mean\*RelativeAccuracy))^2)

nnew\_madad2\_currentstate1

עבור מצב קיים יש להריץ n=34.9 ועבור מדד 2 יש להריץ n=31.4 ולכן נבחר את הגבוה מבניהם ונבחר

n2 <- 35

t2 <- qt(0.985,(n2-1))

tableValues2 <- c(Accuracy(t2,n2,(madad\_1\_STDEV)^2, madad\_1\_Mean),Accuracy(t2,n2,(madad2\_STDEV)^2, madad2\_Mean),Accuracy(t2,n2,(madad\_3\_stdev)^2, madad\_3\_mean),

Accuracy(t2,n2,(madad\_1\_STDEV\_a1)^2, madad\_1\_Mean\_a1),Accuracy(t2,n2,(madad2\_STDEV\_a1)^2, madad2\_Mean\_a1),Accuracy(t2,n2,(madad\_3\_stdev\_a1)^2, madad\_3\_mean\_a1),

Accuracy(t2,n2,(madad\_1\_STDEV\_a2)^2, madad\_1\_Mean\_a2),Accuracy(t2,n2,(madad2\_STDEV\_a2)^2, madad2\_Mean\_a2),Accuracy(t2,n2,(madad\_3\_stdev\_a2)^2, madad\_3\_mean\_a2))

checkAccuracy2 <- matrix(tableValues2,ncol=3,byrow=TRUE)

colnames(checkAccuracy2)<- c("Madad1", "madad2", "madad3")

rownames(checkAccuracy2) <- c("Current state","Alternative1","Alternative2")

checkAccuracy2 <- as.table(checkAccuracy2)

checkAccuracy2

נקבל את הדיוק היחסי:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **מדד חווית לקוח** | **מדד יעילות הרכבת** | **מדד פספוס** |
| **מצב קיים** | 0.0004 | 0.0281 | 0.00254 |
| **חלופה 1** | 0.0003 | 0.0291281 | 0.00110 |
| **חלופה 2** | 0.0004 | 0.0266 | 0.00252 |

נזכיר שהחסם העליון שלנו הוא 0.291262, ולכן חלופה 1 במדד 2 עדיין לא עומדת בדיוק היחסי. נבדוק שוב לפי הנוסחא כמה כדאי להריץ

# #calculate of how many runs we need to add - Indicator 2 Alternative 1

nnew\_madad2\_currentstate2 = n2 \* (((Accuracy(t2,n2,(madad2\_STDEV)^2, madad2\_Mean)\*madad2\_Mean)/(madad2\_Mean\*RelativeAccuracy))^2)

nnew\_madad2\_currentstate2

ונקבל n=36 .

נבדוק את הדיוק היחסי כאשר

הדיוק היחסי שיוצא עכשיו הוא:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **מדד חווית לקוח** | **מדד יעילות הרכבת** | **מדד פספוס** |
| **מצב קיים** | 0.0004 | 0.0268 | 0.00245 |
| **חלופה 1** | 0.0003 | 0.0278 | 0.00115 |
| **חלופה 2** | 0.0004 | 0.0255 | 0.00244 |

על כן נבחר כי זמן הריצות שלנו יהיה n=36.

n3 <-36

t2 <- qt(0.985,(n3-1))

tableValues3 <- c(Accuracy(t3,n3,(madad\_1\_STDEV)^2, madad\_1\_Mean),Accuracy(t3,n3,(madad2\_STDEV)^2, madad2\_Mean),Accuracy(t3,n3,(madad\_3\_stdev)^2, madad\_3\_mean),

Accuracy(t3,n3,(madad\_1\_STDEV\_a1)^2, madad\_1\_Mean\_a1),Accuracy(t3,n3,(madad2\_STDEV\_a1)^2, madad2\_Mean\_a1),Accuracy(t3,n3,(madad\_3\_stdev\_a1)^2, madad\_3\_mean\_a1),

Accuracy(t3,n3,(madad\_1\_STDEV\_a2)^2, madad\_1\_Mean\_a2),Accuracy(t3,n3,(madad2\_STDEV\_a2)^2, madad2\_Mean\_a2),Accuracy(t3,n3,(madad\_3\_stdev\_a2)^2, madad\_3\_mean\_a2))

checkAccuracy3 <- matrix(tableValues3,ncol=3,byrow=TRUE)

colnames(checkAccuracy3)<- c("Madad1", "madad2", "madad3")

rownames(checkAccuracy3) <- c("Current state","Alternative1","Alternative2")

checkAccuracy3 <- as.table(checkAccuracy3)

checkAccuracy3

## נספח 8- בדיקת משאבים לבחינת חלופות

### נספח 8.1 – בדיקת זמן השירות

כאן אנחנו בודקים פר כל משאב מהו ממוצע זמן השירות בכל שעה בכל תחנה כדי לראות מהי שעת העומס.

rush\_hour\_coffe\_0\_to\_60 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 0 and 60

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_60\_to\_120 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 60 and 120

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_120\_to\_180 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 120 and 180

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_180\_to\_240 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 180 and 240

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_240\_to\_300 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 240 and 300

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_300\_to\_360 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 300 and 360

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_360\_to\_420 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 360 and 420

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_420\_to\_480 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 420 and 480

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_480\_to\_540 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 480 and 540

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_540\_to\_600 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 540 and 600

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_600\_to\_660 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 600 and 660

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_660\_to\_720 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 660 and 720

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_720\_to\_780 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 720 and 780

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_780\_to\_840 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 780 and 840

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_840\_to\_900 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 840 and 900

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_900\_to\_960 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 900 and 960

group by resource

order by Avg\_system DESC")

rush\_hour\_coffe\_960\_to\_1020 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'coffee%' and time between 960 and 1020

group by resource

order by Avg\_system DESC")

###Exit

rush\_hour\_exit\_0\_to\_60 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 0 and 60

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_60\_to\_120 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 60 and 120

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_120\_to\_180 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 120 and 180

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_180\_to\_240 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 180 and 240

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_240\_to\_300 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 240 and 300

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_300\_to\_360 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 300 and 360

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_360\_to\_420 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 360 and 420

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_420\_to\_480 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 420 and 480

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_480\_to\_540 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 480 and 540

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_540\_to\_600 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 540 and 600

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_600\_to\_660 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 600 and 660

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_660\_to\_720 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 660 and 720

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_720\_to\_780 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 720 and 780

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_780\_to\_840 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 780 and 840

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_840\_to\_900 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 840 and 900

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_900\_to\_960 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 900 and 960

group by resource

order by Avg\_system DESC")

rush\_hour\_exit\_960\_to\_1020 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'exit%' and time between 960 and 1020

group by resource

order by Avg\_system DESC")

###security

rush\_hour\_security\_0\_to\_60 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 0 and 60

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_60\_to\_120 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 60 and 120

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_120\_to\_180 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 120 and 180

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_180\_to\_240 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 180 and 240

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_240\_to\_300 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 240 and 300

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_300\_to\_360 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 300 and 360

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_360\_to\_420 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 360 and 420

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_420\_to\_480 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 420 and 480

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_480\_to\_540 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 480 and 540

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_540\_to\_600 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 540 and 600

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_600\_to\_660 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 600 and 660

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_660\_to\_720 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 660 and 720

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_720\_to\_780 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 720 and 780

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_780\_to\_840 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 780 and 840

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_840\_to\_900 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 840 and 900

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_900\_to\_960 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 900 and 960

group by resource

order by Avg\_system DESC")

rush\_hour\_security\_960\_to\_1020 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'security%' and time between 960 and 1020

group by resource

order by Avg\_system DESC")

###tickets

rush\_hour\_ticket\_0\_to\_60 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 0 and 60

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_60\_to\_120 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 60 and 120

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_120\_to\_180 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 120 and 180

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_180\_to\_240 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 180 and 240

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_240\_to\_300 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 240 and 300

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_300\_to\_360 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 300 and 360

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_360\_to\_420 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 360 and 420

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_420\_to\_480 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 420 and 480

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_480\_to\_540 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 480 and 540

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_540\_to\_600 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 540 and 600

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_600\_to\_660 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 600 and 660

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_660\_to\_720 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 660 and 720

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_720\_to\_780 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 720 and 780

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_780\_to\_840 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 780 and 840

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_840\_to\_900 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 840 and 900

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_900\_to\_960 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

where resource like 'ticket%' and time between 900 and 960

group by resource

order by Avg\_system DESC")

rush\_hour\_ticket\_960\_to\_1020 <- sqldf("select resource, Avg(server) as Avg\_system from resourceData as a

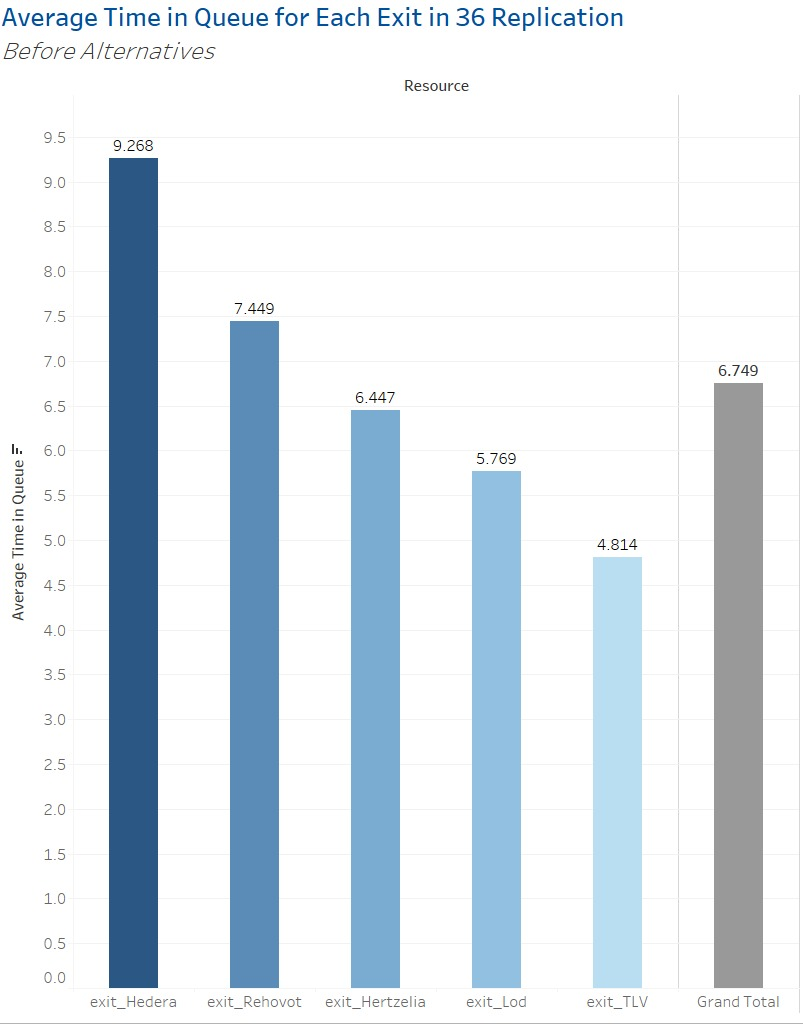
where resource like 'ticket%' and time between 960 and 1020

group by resource

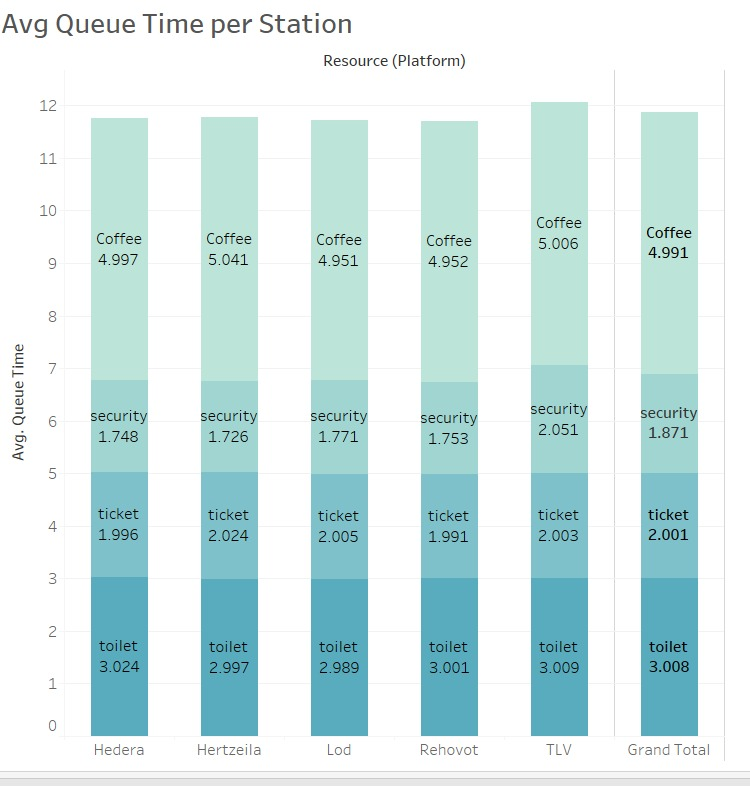
order by Avg\_system DESC")

### נספח 8.2- בדיקת היציאות העמוסות

התרשים מציג את הזמן הממוצע שלוקח לנוסע לעבור בסבסבות היציאה בכל תחנה.

לפי הגרף בחרנו את התחנות העמוסות ביותר להם נרצה להוסיף סבסבת להקלת העומס.

### נספח 8.3 – חלוקת זמן במשאבי הרכבת

בגרף ניתן לראות איפה נוסע ממוצע "מבזבז" הכי הרבה זמן במתקני הרכבת. ניתן להסיק בבירור שהוספת עובד תהיה אפקטיבית ביותר בעמדות הקפה מאשר בעמדות אחרות.

## נספח 9- השוואת מדדים

ביצוע t-test לפי המדדים

############## madad 1 - mean\_time ###############################

#compare current state and alternative1 - madad 1

CS\_alt1\_1\_tTest <- t.test(x = madad1$mean\_time , y = madad1\_a1$mean\_time, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(CS\_alt1\_1\_tTest)

#compare current state and alternative2 - madad 1

CS\_alt2\_1\_tTest <- t.test(x = madad1$mean\_time , y = madad1\_a2$mean\_time, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(CS\_alt2\_1\_tTest)

#compare alternative1 and alternative2 - madad 1

alt1\_alt2\_1\_tTest <- t.test(x = madad1\_a1$mean\_time , y = madad1\_a2$mean\_time, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(alt1\_alt2\_1\_tTest)

t\_test\_madad1<-c(CS\_alt1\_1\_tTest$conf.int[1], CS\_alt1\_1\_tTest$conf.int[2], CS\_alt2\_1\_tTest$conf.int[1], CS\_alt2\_1\_tTest$conf.int[2], alt1\_alt2\_1\_tTest$conf.int[1], alt1\_alt2\_1\_tTest$conf.int[2])

madad1\_t\_test <- matrix(t\_test\_madad1,ncol=2,byrow=TRUE)

colnames(madad1\_t\_test)<- c("confidence interval bottom", "confidence interval upper")

rownames(madad1\_t\_test) <- c("Current state vs Alternative 1","Current state vs Alternative 2","Alternative 1 vs Alternative 2")

madad1\_t\_test <- as.table(madad1\_t\_test)

madad1\_t\_test

############## madad 2 - AVG Simulation time ###############################

#compare current state and alternative1 - madad 2

CS\_alt1\_2\_tTest <- t.test(x = madad2$AVG\_Simulation\_time , y = madad2\_a1$AVG\_Simulation\_time, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(CS\_alt1\_2\_tTest)

#compare current state and alternative2 - madad 2

CS\_alt2\_2\_tTest <- t.test(x = madad2$AVG\_Simulation\_time , y = madad2\_a2$AVG\_Simulation\_time, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(CS\_alt2\_2\_tTest)

#compare alternative1 state and alternative2 - madad 2

alt1\_alt2\_2\_tTest <- t.test(x = madad2\_a1$AVG\_Simulation\_time , y = madad2\_a2$AVG\_Simulation\_time, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(alt1\_alt2\_2\_tTest)

t\_test\_madad2<-c(CS\_alt1\_2\_tTest$conf.int[1], CS\_alt1\_2\_tTest$conf.int[2], CS\_alt2\_2\_tTest$conf.int[1], CS\_alt2\_2\_tTest$conf.int[2], alt1\_alt2\_2\_tTest$conf.int[1], alt1\_alt2\_2\_tTest$conf.int[2])

madad2\_t\_test <- matrix(t\_test\_madad2,ncol=2,byrow=TRUE)

colnames(madad2\_t\_test)<- c("confidence interval bottom", "confidence interval upper")

rownames(madad2\_t\_test) <- c("Current state vs Alternative 1","Current state vs Alternative 2","Alternative 1 vs Alternative 2")

madad2\_t\_test <- as.table(madad2\_t\_test)

madad2\_t\_test

############## madad 3 - bus Proportion ###############################

#compare current state and alternative1 - madad 3

CS\_alt1\_3\_tTest <- t.test(x = madad3$bus\_Proportion , y = madad3\_a1\_t$bus\_Proportion, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(CS\_alt1\_3\_tTest)

#compare current state and alternative2 - madad 3

CS\_alt2\_3\_tTest <- t.test(x = madad3$bus\_Proportion , y = madad3\_a2$bus\_Proportion, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(CS\_alt2\_3\_tTest)

#compare alternative1 state and alternative2 - madad 3

alt1\_alt2\_3\_tTest <- t.test(x = madad3\_a1\_t$bus\_Proportion , y = madad3\_a2$bus\_Proportion, alternative = "two.sided", paired = TRUE, var.equal= TRUE, conf.level = 0.99)

print(alt1\_alt2\_3\_tTest)

t\_test\_madad3<-c(CS\_alt1\_3\_tTest$conf.int[1], CS\_alt1\_3\_tTest$conf.int[2], CS\_alt2\_3\_tTest$conf.int[1], CS\_alt2\_3\_tTest$conf.int[2], alt1\_alt2\_3\_tTest$conf.int[1], alt1\_alt2\_3\_tTest$conf.int[2])

madad3\_t\_test <- matrix(t\_test\_madad3,ncol=2,byrow=TRUE)

colnames(madad3\_t\_test)<- c("confidence interval bottom", "confidence interval upper")

rownames(madad3\_t\_test) <- c("Current state vs Alternative 1","Current state vs Alternative 2","Alternative 1 vs Alternative 2")

madad3\_t\_test <- as.table(madad3\_t\_test)

madad3\_t\_test

## נספח 10 – פונקציות

*##----------------------------------------- 1. all functions --------------------------------------*

* + - 1. *פונקציה אשר תחזיר לנו דגימות חיוביות בלבד מכל התפלגות נורמלית שנשתמש בה. השתמשנו בה כשדגמנו עבור כל ישות זמנים בתוך הסימולציה.*

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

*while(TRUE)*

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

*if (sample>0)*

*return (sample)*

*{{*

* + - 1. *פונקציה שמחזירה דגימות מהתפלגויות הנתונות לתכונות המקומיות של כל ישות לפי סוג הישות.*

*typeInef<-function(type){*

*if (type==1){ #Citizens*

*start<-rdiscrete(1, c(0.24,0.1,0.37,0.09,0.2), c(1,2,3,4,5)) ## "Hedera"=1 ,"Hertzelia"=2,"TLV"=3, "Lod"=4, "Rehovot"=5*

*end <- rdiscrete(1, c(0.25,0.33,0.12,0.1,0.2), c(1,2,3,4,5)) ## "Hedera"=1 ,"Hertzelia"=2,"TLV"=3, "Lod"=4, "Rehovot"=5*

*while (start==end){*

*start<-rdiscrete(1, c(0.24,0.1,0.37,0.09,0.2), c(1,2,3,4,5))*

*end <- rdiscrete(1, c(0.25,0.33,0.12,0.1,0.2), c(1,2,3,4,5))}*

*if(start>end){*

*direction<-1 ##north*

*}*

*else{*

*direction<-2 ##south*

*}*

*current\_station<-start*

*security<-runif(1,2,3)*

*ticket<-trimmedNorm(2,0.7)*

*coffee<-trimmedNorm(5,1.5)*

*toilet<-trimmedNorm(3,45/60)*

*platform<-runif(1,8,14)*

*exit<-rtriangle(1,2,8,4)*

*return (c(type,start,direction,current\_station,end,security,ticket,coffee,toilet,platform,exit))*

*}*

*if (type==2){ #Soldiers*

*start<-rdiscrete(1, c(0.24,0.1,0.37,0.09,0.2), c(1,2,3,4,5)) ## "Hedera"=1 ,"Hertzelia"=2,"TLV"=3, "Lod"=4, "Rehovot"=5*

*end <- rdiscrete(1, c(0.25,0.33,0.12,0.1,0.2), c(1,2,3,4,5)) ## "Hedera"=1 ,"Hertzelia"=2,"TLV"=3, "Lod"=4, "Rehovot"=5*

*while (start==end){*

*start<-rdiscrete(1, c(0.24,0.1,0.37,0.09,0.2), c(1,2,3,4,5))*

*end <- rdiscrete(1, c(0.25,0.33,0.12,0.1,0.2), c(1,2,3,4,5))}*

*if(start>end){*

*direction<-1 ##north*

*}*

*else{*

*direction<-2 ##south*

*}*

*current\_station<-start*

*security<-runif(1,0.5,1)*

*ticket<-trimmedNorm(2,0.2)*

*coffee<-trimmedNorm(5,1.5)*

*toilet<-trimmedNorm(3,45/60)*

*platform<-runif(1,8,14)*

*exit<-rtriangle(1,2,8,4)*

*return (c(type,start,direction,current\_station,end,security,ticket,coffee,toilet,platform,exit))}}*

* + - 1. *פונקציה שמחזירה את זמן הנסיעה ברכבת בין כל שתי תחנות.*

*Drive\_time <-function(){*

*U1<- runif(1,0,1)*

*U2<- runif(1,0,1)*

*if(U1<= 1/5) }*

*return(sqrt(900+2700\*U2))*

*{*

*else}*

*return(1.5+sqrt(49+3200\*U2)/2)}}*

* + - 1. *פונקציה שבודקת אם זאת תחנה אחרונה של הנוסע לפי התכונות שלו.*

*check\_if\_last\_station<-function(current\_station, end){*

*if(current\_station == end){*

*return{(1)*

*} else*

*return{{(2)*

* + - 1. *פונקציה שבודקת האם הישות כבר הייתה על הרכבת, כדי לתת לה עדיפות בעלייה לרכבת לתחנת היעד על פני נוסעים חדשים.*

*was\_on\_train<-function(start,direction,current\_station){*

*if(current\_station == start)}*

*return(FALSE){*

*else{ return(TRUE)}}*

* + - 1. *שינוי העדיפות של החיילים בסבסבות תל אביב. עשינו את זה בפונקציה על מנת שלא יקבלו עדיפות בכל שאר הסימולציה.*

*exit\_TLV<-function(type)}*

*if(type==1) { ##citizen*

*return(c(0,0,F))}*

*if(type==2){*

*return(c(1,1,F))}}*

* + - 1. *פונקציית עזר לתקלות: מחזירה וקטור שמכיל מספר תחנה, שעת תקלה, והאם קרתה תקלה או לא.*

*vector\_of\_cities <- c("1", "2", "3", "4" ,"5")*

*i <- 1*

*try <- c()*

*obstacles <- function (hour)}*

*while(hour != activityTime){*

*where <-vector\_of\_cities[i]*

*obstacle\_or\_not <- rdiscrete(1,c(0.3,0.7),c(TRUE,FALSE))*

*hour <- hour+60*

*i <- i+1*

*if (i>length(vector\_of\_cities)){*

*i <- 1}*

*try <- append(try,c(obstacle\_or\_not,where,hour))}*

*return(try)}*

*8.פונקציה שמחזירה האם יש תקלה או אין.*

*times\_1<-3*

*times\_2<-6*

*problem<-1*

*station\_m<-2*

*malfunction\_in\_the\_train<-function(station, time)}*

*malfunction<-FALSE*

*while(times\_1 <45)}*

*if(time>=malfunction\_vec[times\_1] & time<=malfunction\_vec[times\_2] & station==malfunction\_vec[station\_m])}*

*malfunction<-malfunction\_vec[problem]*

*times\_1<-45*

*else}*

*times\_1<-times\_1+3*

*times\_2<-times\_2+3*

*problem<-problem+3*

*station\_m<-station+3*

*if(malfunction== TRUE){ return(1)}*

*else{ return(0) }*

*9.פוקנציית עזר ל- time\_out\_platform שמחזירה את שם התחנה*

*station\_name<-function(current\_station)}*

*if(current\_station==1)}*

*return("Hedera"){*

*if(current\_station==2){*

*return("Hertzelia")}*

*if(current\_station==3)}*

*return("TLV")}*

*if(current\_station==4){*

*return("Lod")}*

*if(current\_station==5)}*

*return("Rehovot")}}*

*10.פונקציה שמחשבת כמה זמן כל ישות מחכה לרכבת מהרגע שמגיעה לרציף.*

*time\_out\_platform <- function(time,current\_station)}*

*timeout<-train\_schedule-time*

*if(length(timeout[timeout>0])==0)}*

*platform <- paste("platform",station\_name(current\_station),sep = "\_")*

*exit<-paste(station\_name(current\_station),"bus",sep = "\_")*

*trajectory()%>%release(platform)%>%join(exit)*

*return(0) }*

*else{*

*return(min(timeout[timeout>0]))}}*

*11.פונקצייה שמשנה עדיפות למי שהיה על הרכבת כדי שיהיה לו בוודאות מקום על הרציף.*

*check\_prioritization<-function(current\_station,end){*

*if(current\_station!=end){*

*return(c (1,1 ,TRUE)){*

*else}*

*return(c (0,1 ,TRUE))}}*