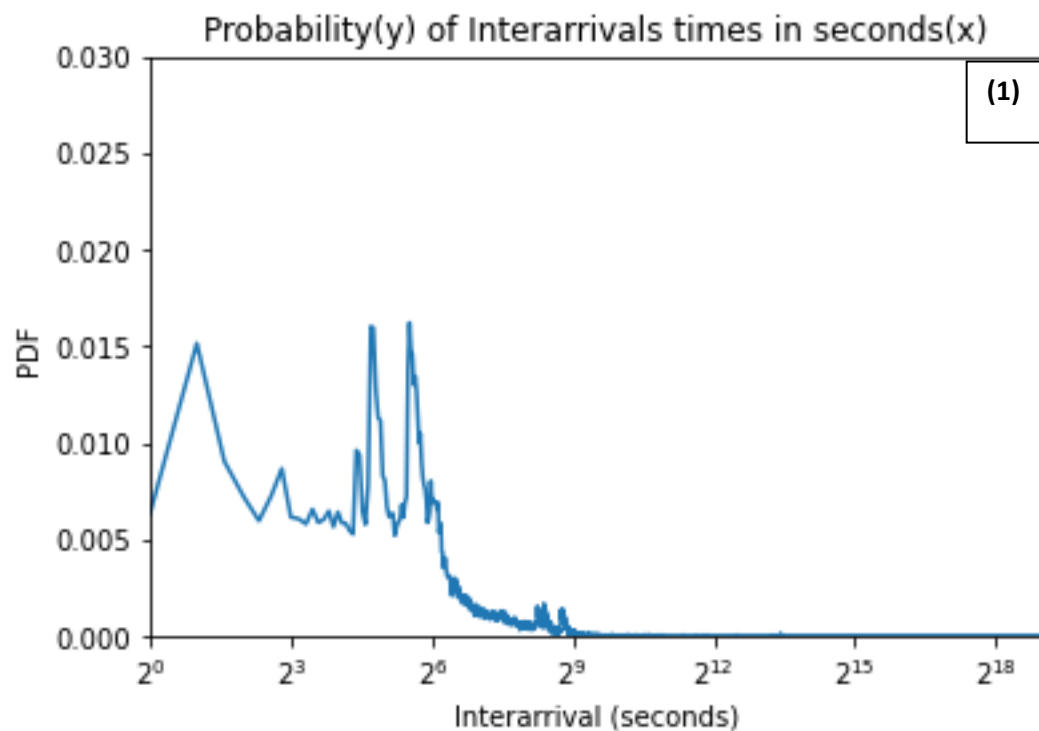


December 30

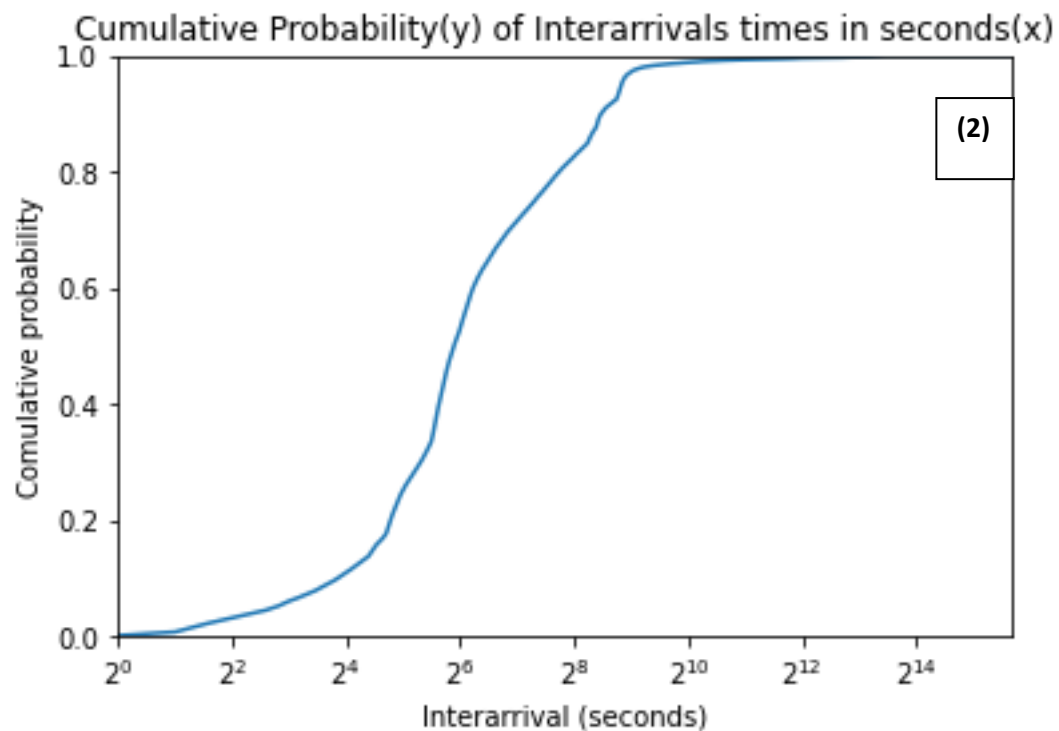


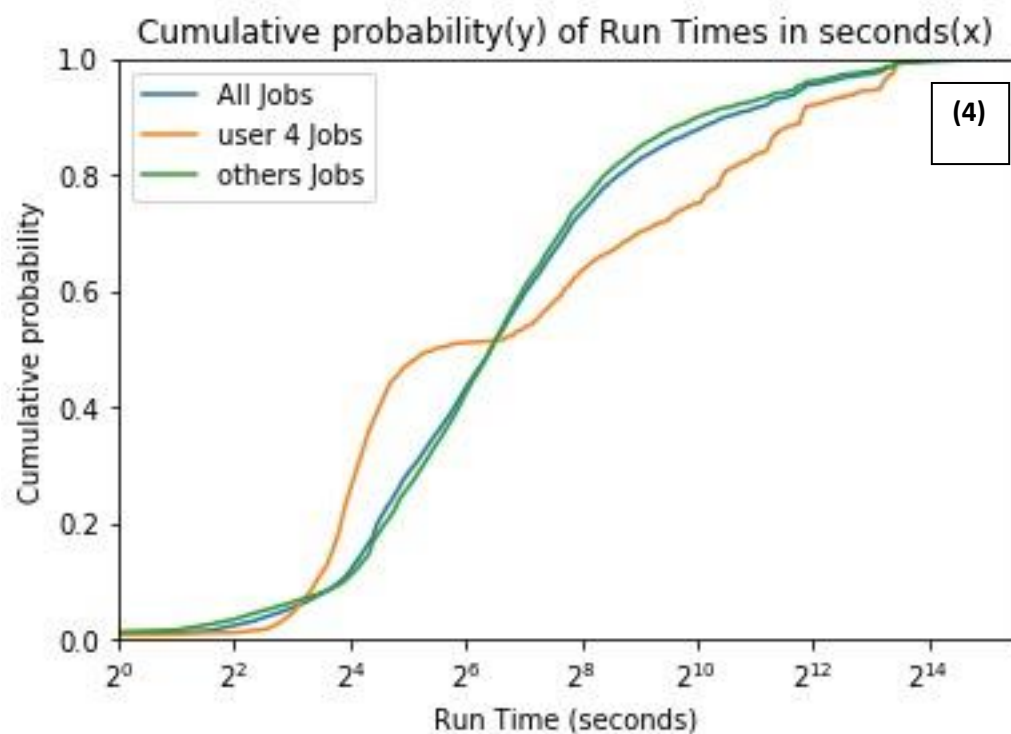
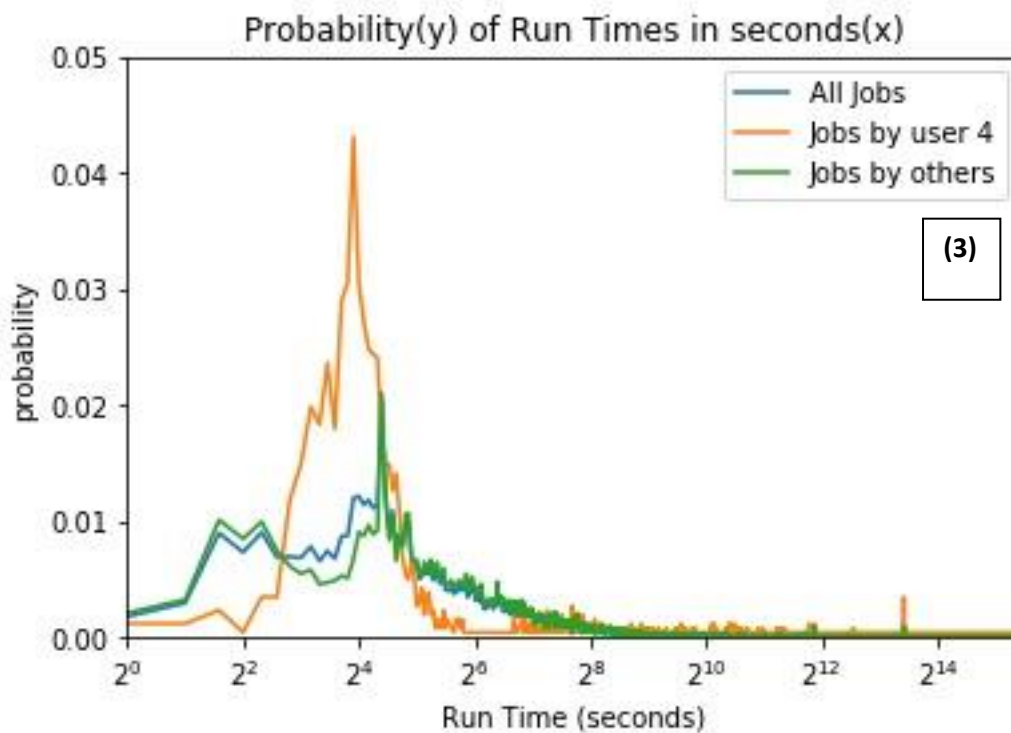
University of Haifa
Workload for Computer Systems Performance Evaluation

NASA-Log File Graphs:



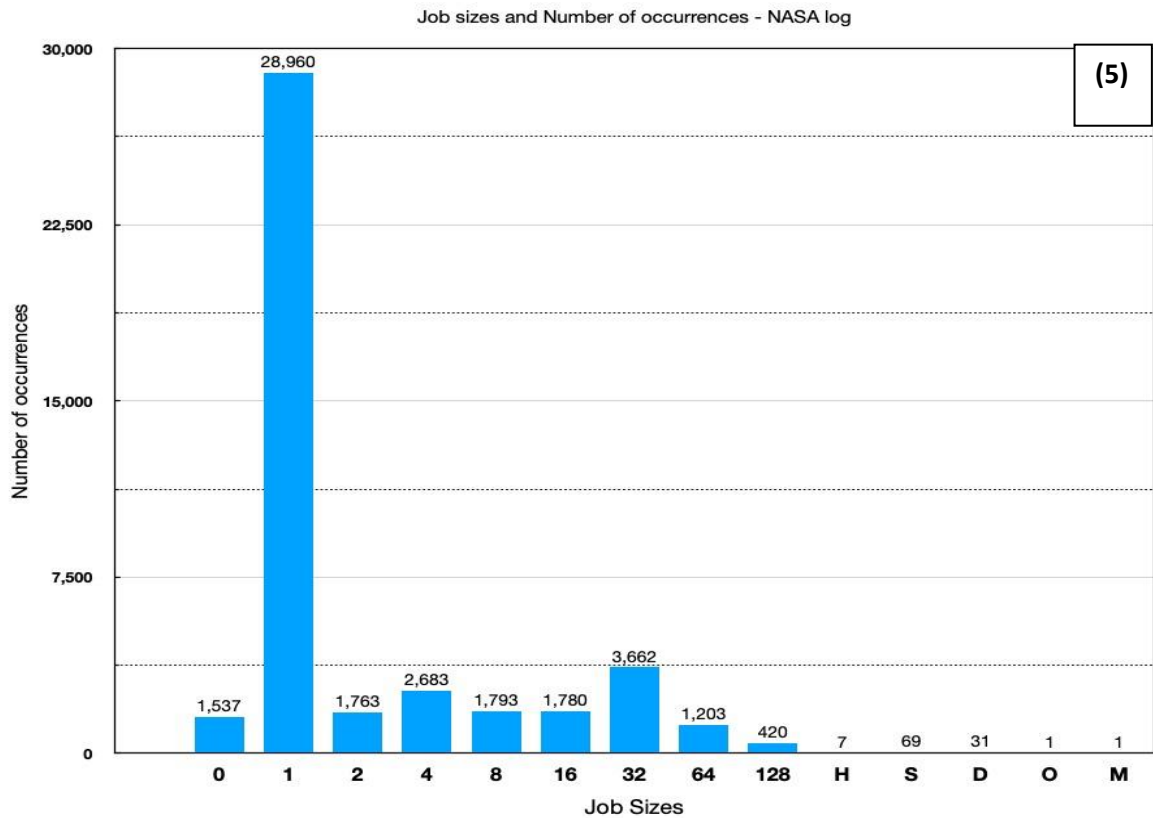
אנחנו יכולים לראות שהגרף מיוצג בצורה הלוגריתמית בציר ה-x והסיבה היא שהיו ערכים ממש גדולים שאי אפשר לייצג עם הערכים הרגילים, בנוסף שמנו לב שהיה משתמש בעל שם משתמש "special" שהיו לו JOBS ארוכים במיוחד ואחד מהם נמשך כמעט ל 6 ימים. זה בעצם מסביר את ה Heavy tail שאפשר לראות בגרף.



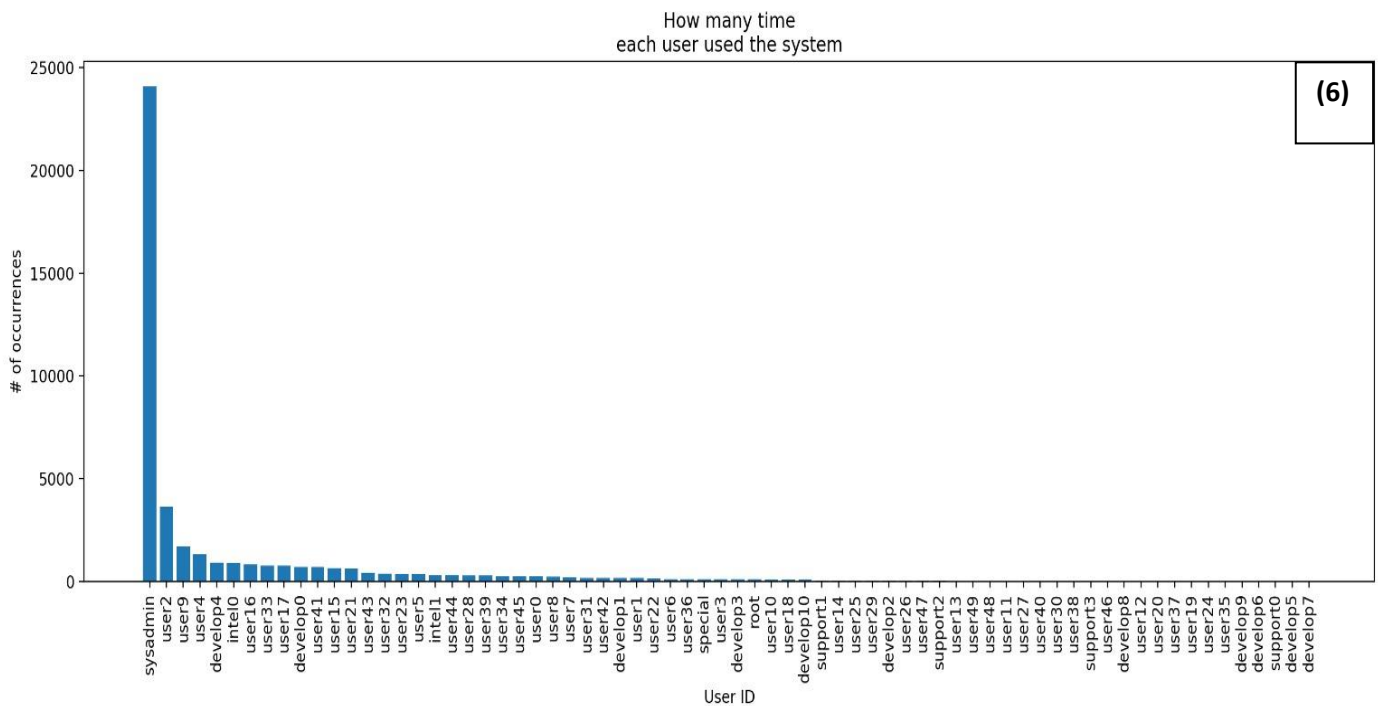


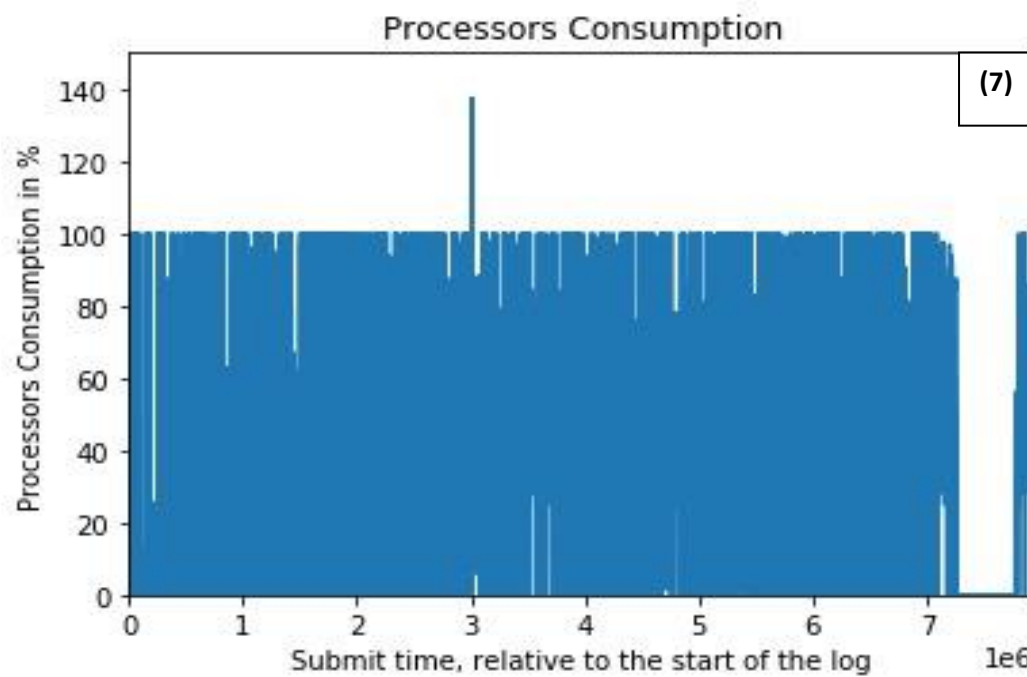
גרפים שמראים התנהגות שונה של אחד הusers משאר הusers שזמני הריצה של jobs שלו יותר גדולים משל השאר.

אפשר לראות גם שבזנב של הקו של משתמש "special" שאפילו כאשר $X > 2^{14}$ את ה Y עדיין לא שווה ל 1 וזה יוצא מהסיבה שיש לא JOB שנמשך לכמעט 6 ימים.



היסטוגרמה שמראה את מספר המופעים של גדלי job שונים. אפשר לראות שהגודל הכי נפוץ הוא 1 וזה נובע מכך שיש המון פעולות של sysadmin (אפשר לראות אותם בגרף הבא) שהם פעולות מערכת, אז בדרך כלל יהיו פעולות קצרות כמו למשל pid שבודקת פעילות המערכת.

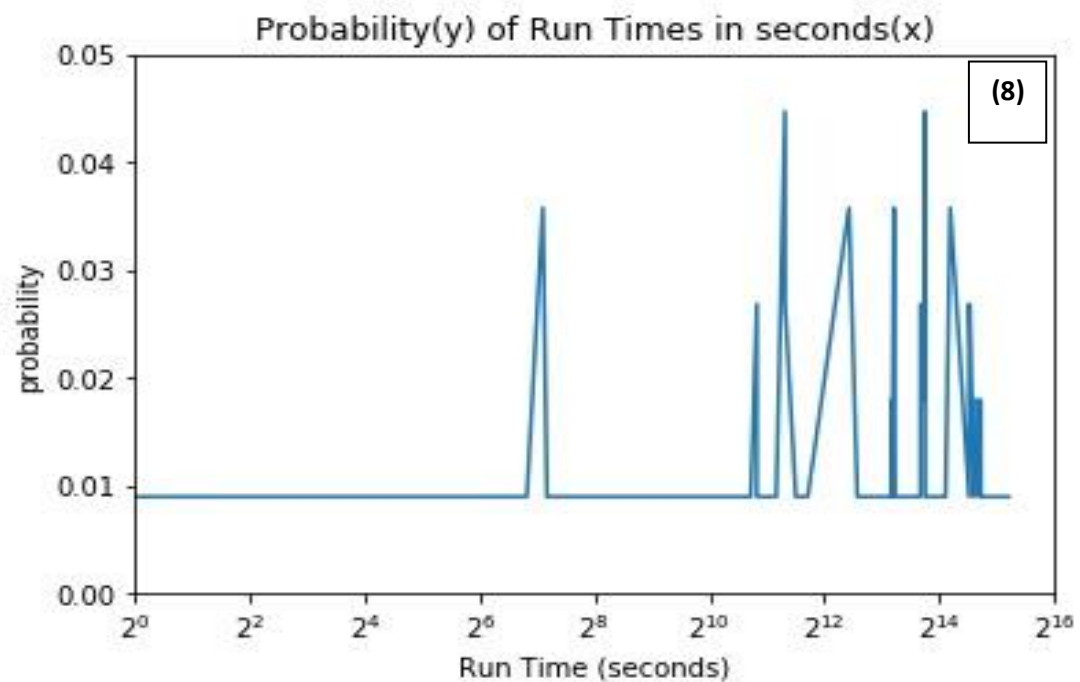




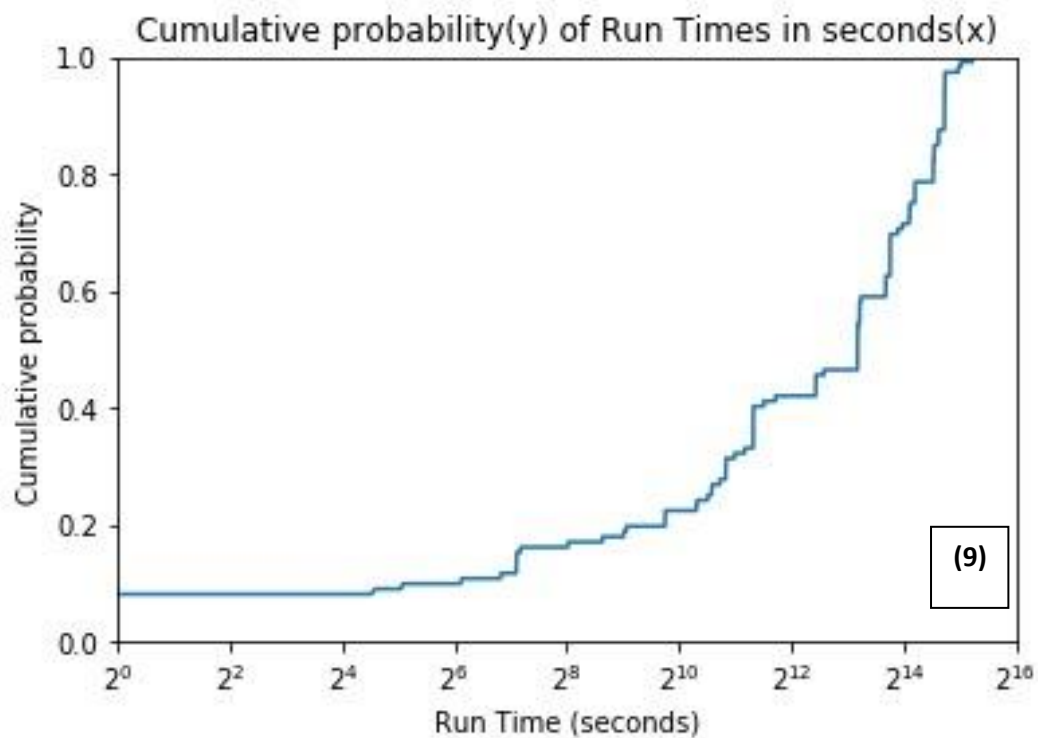
הגרף הזה מראה את אחוזי המעבדים התפוסים בכל שניה מהתחלת ה Log ועד סיומו.

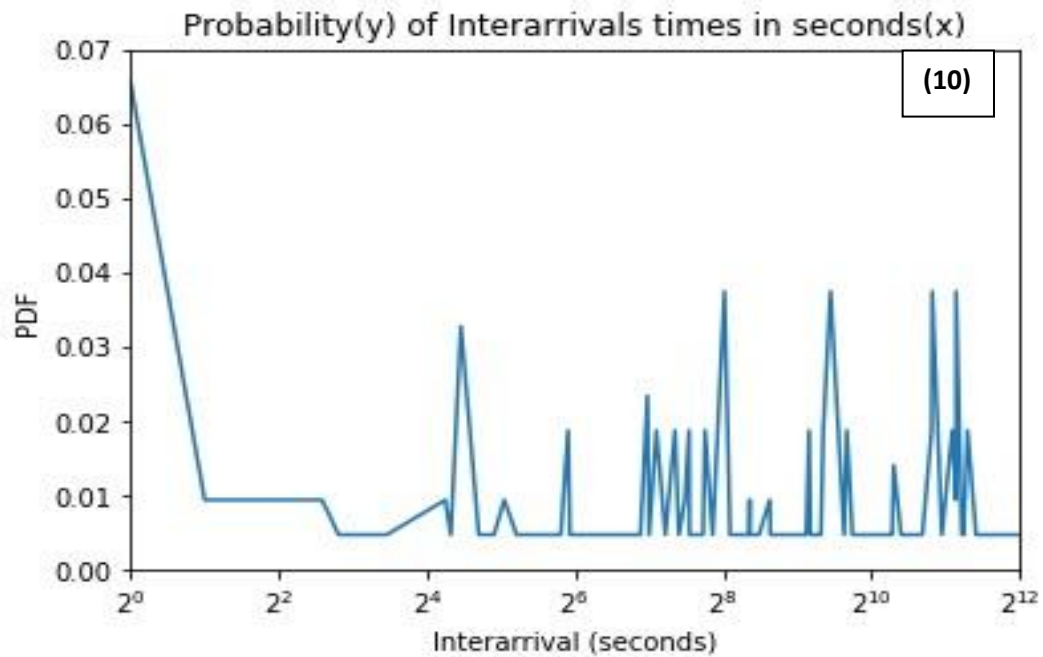
(המספרים בציר ה X הם כפולות של 10^6)

MATLAB-Log File Graphs:

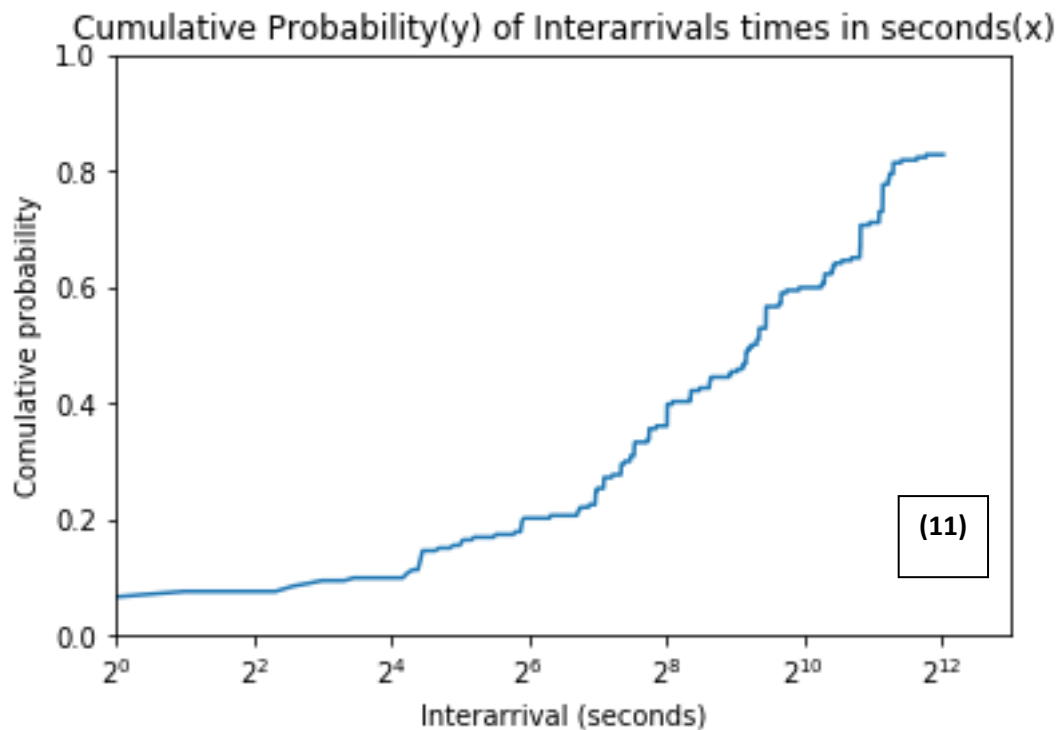


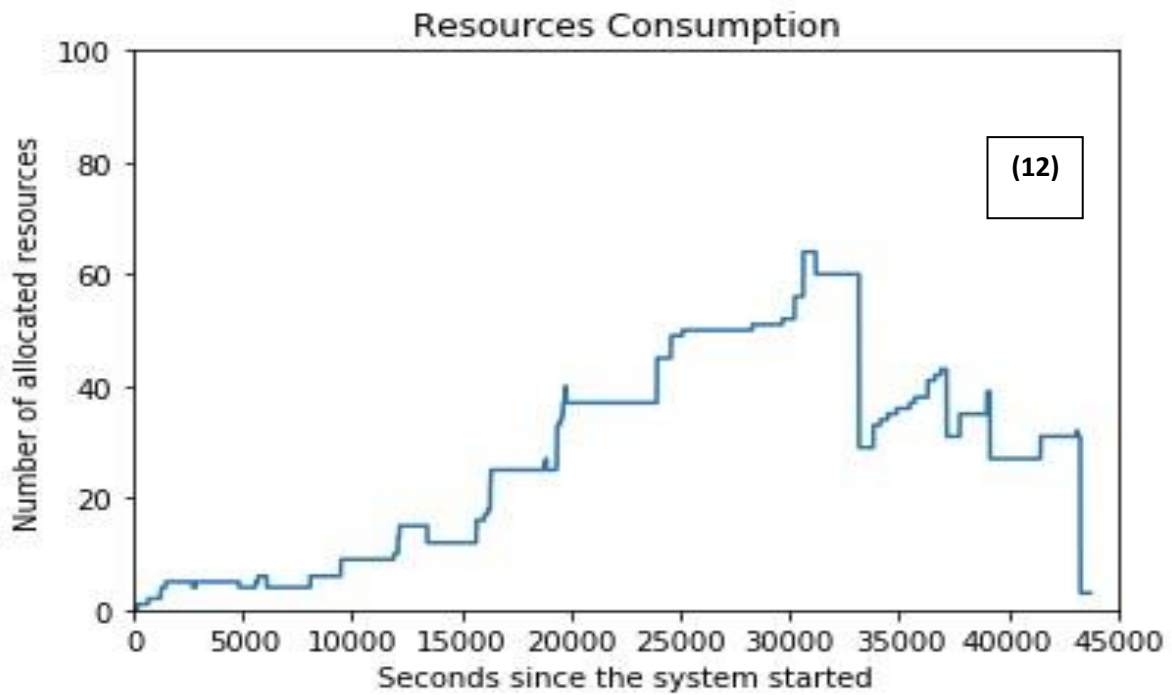
אפשר לראות בגרף ה PDF כל מיני bursts ו flurries שבצמם מראות שיש כמה RunTimes נפוצות.





אפשר לראות בגרף ה PDF כל מיני bursts ו flurries שבעצם מראות שיש כמה Interarrival Times נפוצות. אפשר גם לראות שכאשר $1=X$ יש לנו הסתברות גדולה יחסית לשאר ה-Interarrivals שזה בעצם אומר לנו שיש כל מנה jobs שבאים אחד אחרי השני בשניה אחת.





(12)

ניתן להסיק מהגרף ששעות העומס על המערכת הן בין 17:40-20:05 וזה הגיוני שהעומס ירד באופן משמעותי אחרי השעה 20:00 כי בדרך כלל חלק גדול מהמשתמשים (עובדי היי-טק וסטודנטים) מסיימים עבודה בסביבות השעה הזו.

או ייתכן שהשרת של MATLAB שחרר כל מיני משאבים ביחד באותו זמן.

Stage 2 – Questions:

שאלה 2 :

נקח למשל את הגרף של ה Processors Consumption של הלוג של NASA- ניתן לראות כי יש לנו פיק של 140% של Processors שזאת תופעת Abnormal Behaviour. בגלל הסיבה הפשוטה שבכל רגע נתון, הגרף הנ"ל מייצג את מספר המעבדים ה"תפוסים" ע"י הגיבויים השונים, לא ניתן ל"שים את האצבע" על הבעיה והגיבוי הספיציפי שגורם לפיק הזה. (בספר בעמוד מספר 50- פיסקה 2).

שאלה 3 :

בטרייס של NASA, קיבלנו אותו נקי ומוכן לשימוש, לכן לא הצטרכנו לעשות הרבה בנדון, אלא כבר ליישם את הנדרש (SWF, גרפים וכו').

בטרייס של MATLAB שקיבלנו, אכן היה לנו כל מיני שורות שהיינו צריכים לנקות – כגון :

```
Matlab_Log - Notepad
File Edit Format View Help
10:55:26 (lmgrd) -----
10:55:26 (lmgrd) Please Note:
10:55:26 (lmgrd)
10:55:26 (lmgrd) This log is intended for debug purposes only.
10:55:26 (lmgrd) In order to capture accurate license
10:55:26 (lmgrd) usage data into an organized repository,
10:55:26 (lmgrd) please enable report logging. Use Flexera's
10:55:26 (lmgrd) software license administration solution,
10:55:26 (lmgrd) FlexNet Manager, to readily gain visibility
10:55:26 (lmgrd) into license usage data and to create
10:55:26 (lmgrd) insightful reports on critical information like
10:55:26 (lmgrd) license availability and usage. FlexNet Manager
10:55:26 (lmgrd) can be fully automated to run these reports on
10:55:26 (lmgrd) schedule and can be used to track license
10:55:26 (lmgrd) servers and usage across a heterogeneous
10:55:26 (lmgrd) network of servers including Windows NT, Linux
10:55:26 (lmgrd) and UNIX.
10:55:26 (lmgrd) -----
10:55:26 (lmgrd)
10:55:26 (lmgrd) Server's System Date and Time: Tue Sep 29 2020 10:55:26 Jerusalem Daylight Time
10:55:26 (lmgrd) pid 11508
10:55:26 (lmgrd) SLOG: Summary LOG statistics is enabled.
10:55:26 (lmgrd) Done rereading
10:55:26 (lmgrd) FlexNet Licensing (v11.16.2.1 build 245043 x64_n6) started on license_server_name (IBM PC) (9/29/2020)
10:55:26 (lmgrd) Copyright (c) 1988-2018 Flexera. All Rights Reserved.
10:55:26 (lmgrd) World Wide Web: http://www.flexerasoftware.com
10:55:26 (lmgrd) License file(s): C:\Matlab license path\network.lic
10:55:27 (MLM) Database_Toolbox Datafeed_Toolbox Neural_Network_Toolbox
10:55:27 (MLM) Econometrics_Toolbox RTW_Embedded_Coder Filter_Design_HDL_Coder
10:55:27 (MLM) Fin_Instruments_Toolbox Financial_Toolbox Fixed_Point_Toolbox
10:55:27 (MLM) Fuzzy_Toolbox GPU_Coder GADS_Toolbox
10:55:27 (MLM) Simulink_HDL_Coder EDA_Simulator_Link Image_Acquisition_Toolbox
10:55:27 (MLM) Image_Toolbox Instr_Control_Toolbox LTE_Toolbox
10:55:27 (MLM) MATLAB_Coder MATLAB_Builder_for_Java Compiler
10:55:27 (MLM) MATLAB_Report_Gen MAP_Toolbox Mixed_Signal_Blockset
10:55:27 (MLM) MPC_Toolbox MBC_Toolbox Motor_Control_Blockset
10:55:27 (MLM) Navigation_Toolbox OPC_Toolbox Optimization_Toolbox
10:55:27 (MLM) Distrib_Computing_Toolbox PDE_Toolbox Phased_Array_System_Toolbox
10:55:27 (MLM) Powertrain_Blockset Pred_Maintenance_Toolbox RF_Blockset
10:55:27 (MLM) RF_Toolbox ROS_Toolbox Reinforcement_Learn_Toolbox
10:55:27 (MLM) Risk_Management_Toolbox Robotics_System_Toolbox Robust_Toolbox
10:55:27 (MLM) Sensor_Fusion_and_Tracking SerDes_Toolbox Signal_Toolbox
10:55:27 (MLM) SimBiology SimEvents SimDriveline
10:55:27 (MLM) Power_System_Blocks SimHydraulics SimMechanics
10:55:27 (MLM) Simscape Virtual_Reality_Toolbox SL_Verification_Validation
10:55:27 (MLM) Simulink_Code_Inspector Real-Time_Workshop Simulink_Compiler
10:55:27 (MLM) Simulink_Control_Design Simulink_Coverage Simulink_Design_Optim
10:55:27 (MLM) Simulink_Design_Verifier Real-Time_Win_Target Simulink_PLC_Coder
10:55:27 (MLM) XPC_Target SIMULINK_Report_Gen Simulink_Requirements
10:55:27 (MLM) Simulink_Test SoC_Blockset Excel_Link
10:55:27 (MLM) Stateflow Statistics_Toolbox Symbolic_Toolbox
10:55:27 (MLM) System_Composer Identification_Toolbox Text_Analytics_Toolbox
10:55:27 (MLM) Trading_Toolbox Vehicle_Dynamics_Blockset Vehicle_Network_Toolbox
10:55:27 (MLM) Vision_HDL_Toolbox WLAN_System_Toolbox Wavelet_Toolbox
```

תיעוד של השרת (התחלה) ושל הTOOLBOXES הזמינים שנטענו.

```
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:50:03 Jerusalem Daylight Time,#4,(3744K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:46:31 Jerusalem Daylight Time,#4,(3748K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:40:03 Jerusalem Daylight Time,#4,(3748K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:36:31 Jerusalem Daylight Time,#5,(3744K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:33:38 Jerusalem Daylight Time,#5,(3740K),(user7,computer7,696)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:30:03 Jerusalem Daylight Time,#5,(3740K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:26:31 Jerusalem Daylight Time,#5,(3744K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:23:38 Jerusalem Daylight Time,#5,(3744K),(user7,computer7,696)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:20:03 Jerusalem Daylight Time,#5,(3744K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000000 ms at Tue Sep 29 2020 14:18:52 Jerusalem Daylight Time,#5,(3688K),(user7,computer7,696)
14:55:28 (MLM) (@MLM-SLOG@)
14:55:28 (MLM) (@MLM-SLOG@) === Top 10 Peak Client Requests Processing Time (in ms) ===
14:55:28 (MLM) (@MLM-SLOG@) Time: Tue Sep 29 2020 14:55:28 Jerusalem Daylight Time
14:55:28 (MLM) (@MLM-SLOG@) Request processing time, when, #concurrent clients, (private bytes (in KB)), client info (user, node, FD)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:46 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:55:29 Jerusalem Daylight Time,#3,(3900K),(user8,license_server_name,640)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:45 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:45 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:48 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 10:58:51 Jerusalem Daylight Time,#3,(3976K),(user1,computer1,644)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:17:44 Jerusalem Daylight Time,#5,(3744K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:18:18 Jerusalem Daylight Time,#5,(3688K),(computer3,computer3,376)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:20:27 Jerusalem Daylight Time,#6,(3688K),(user4,computer4,696)
14:55:28 (MLM) (@MLM-SLOG@) 00000016 ms at Tue Sep 29 2020 11:20:32 Jerusalem Daylight Time,#6,(3688K),(user4,computer4,696)
14:55:28 (MLM) (@MLM-SLOG@)
14:55:28 (MLM) (@MLM-SLOG@) === Top 10 Peak In-house Operations time (in ms) ===
14:55:28 (MLM) (@MLM-SLOG@) NOTE: Peak times greater than 4 seconds get recorded.
14:55:28 (MLM) (@MLM-SLOG@) Time: Tue Sep 29 2020 14:55:28 Jerusalem Daylight Time
14:55:28 (MLM) (@MLM-SLOG@) In-house operation time, when, #concurrent clients
14:55:28 (MLM) (@MLM-SLOG@)
14:55:28 (MLM) (@MLM-SLOG@) === Active Connections Info ===
14:55:28 (MLM) (@MLM-SLOG@) Peak active connections #6 attempted at Tue Sep 29 2020 11:20:01 Jerusalem Daylight Time
14:55:28 (MLM) (@MLM-SLOG@)
14:55:28 (MLM) (@MLM-SLOG@) === Memory Usage Info ===
14:55:28 (MLM) (@MLM-SLOG@) Peak private bytes 3984K attempted at Tue Sep 29 2020 10:58:33 Jerusalem Daylight Time
```

סיכום מעת לעת של המשתמשים ושל החיבורים הפעילים במערכת כרגע.

כמובן שלאחר הניקוי, היה לנו מידע נטו שנוכל להשתמש בו, לצורך בניית ה-SWF, כי כל שורה שהיא מהצורה של OUT/IN השארנו, כך שנוכל לקבל את מלוא המידע הדרוש במשימה הזאת ולמשימות הבאות.

אין כל כך מה להשוות בין התוצאות שהיינו מקבלים בלי הניקוי לעיל, כי כל מה שניקינו – יכולים לקבל מהמידע שנשאר (חיבורים פעילים, שמות משתמשים, שעת התחלה סיום וכו'...).

לכן, התוצאה הסופית של הלוג הנקי הייתה רק שורות של OUT IN כך שנוכל לדעת עבור כל משתמש וכל משאב מי, מתי וכמה השתמשו בו למשל.

Stage 3:

אפשר לראות שהגרפים ממוספרים מ 1 ל 12 נתחיל לתאם כל גרף להתפלגות מסוימת.

:Multiple Erlang distributions(1)

ניתן לראות שגרף זה "מורכב" משתי התפלגויות Erlang, כאשר הפרמטר k יהיה $k = 2$.

: CDF of Erlang distribution (2)

באותו אופן לגרף מספר 1, נראה כי גרף זה מייצג לנו CDF של התפלגות Erlang כאשר הפרמטר k יהיה $k = 2$.

: More than one Hyper-Erlang with tail of exponential distribution (3)

ניתן לראות שגרף זה הינו שילוב של שתי התפלגויות – התפלגות Erlang של user רגיל והתפלגות Erlang של special user. לכן, שילוב של שתיהן ייתן לנו Hyper-Erlang distribution, כאשר הפרמטרים יהיו

$$k = 1, \quad p = P(\text{user}), \quad 1 - p = P(\text{special user})$$

: CDF of Hyper-Erlang(4)

באותו אופן לגרף מספר 3, נראה כי גרף זה מייצג לנו CDF של התפלגות Hyper-Erlang כאשר ערכי הפרמטרים זהים לגרף ה-PDF.

(5) אין התפלגות שמתאימה להיסטוגרמות

(6) אין התפלגות שמתאימה להיסטוגרמות

(7) אין התפלגות שמתאימה להיסטוגרמות

(8) lognormal מעורב:

ניתן לראות שגרף זה מתפלג בצורה "יחסית" נורמלית, אך גרף זה נמצא במרחב לוגריתמי (ציר ה- x הינו סקאלה לוגריתמית), ולכן גרף זה מתאים להתפלגות לוג-נורמלית. בנוסף נשים לב שה-mean וה-standard deviation גם נמצאים במרחב לוגריתמי.

----- (9)

לפי המידע על התפלגות הלוג-נורמלית, אין צורה חד-משמעית לגבי ה-CDF.

----- (10)

גרף זה מטעה קלות – ראשית, נשים לב שהוא נמצא במרחב לוגריתמי. אם נסתכל על ה-interval שנמצא מ- 2^4 ל- 2^{12} , ניתן לחשוב שמדובר בהתפלגות לוג-נורמלית (מרחב לוגריתמי, התפלגות "יחסית" נורמלית). מצד שני, ה-interval שנמצא בתחילת הגרף עד 2^4 לא ברור לגמרי.

----- (11)

:Normal distribution (12)

ניתן לזהות בגרף זה "עקומת פעמון" אשר מאפיינת את העומס על המערכת (הסקנו שהעומס נובע ישירות מפזור שעות העבודה בכל יום – עלייה בבקרים וירידה בערבים).

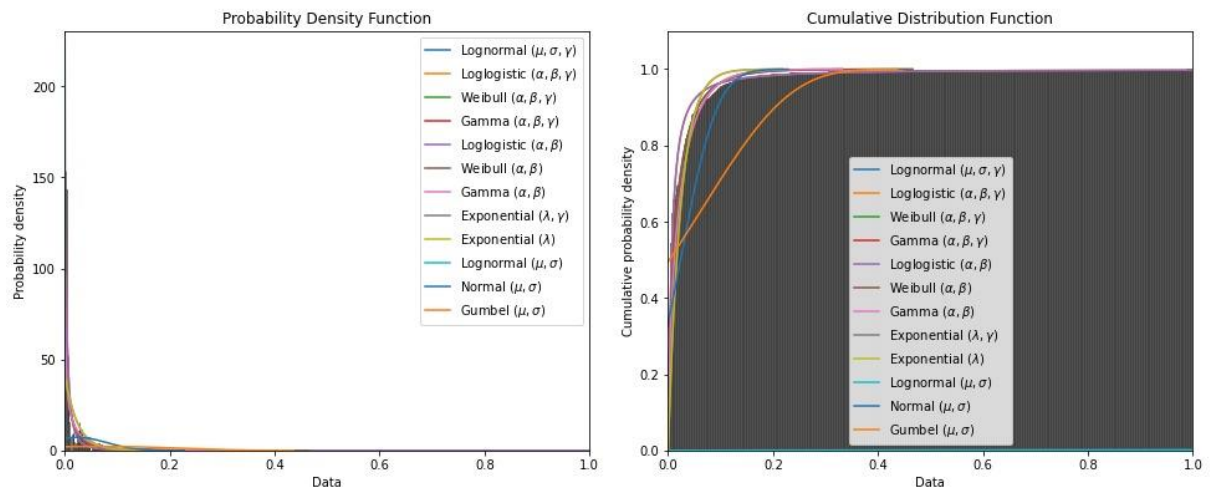
Stage 4:

NASA-Log File Graphs

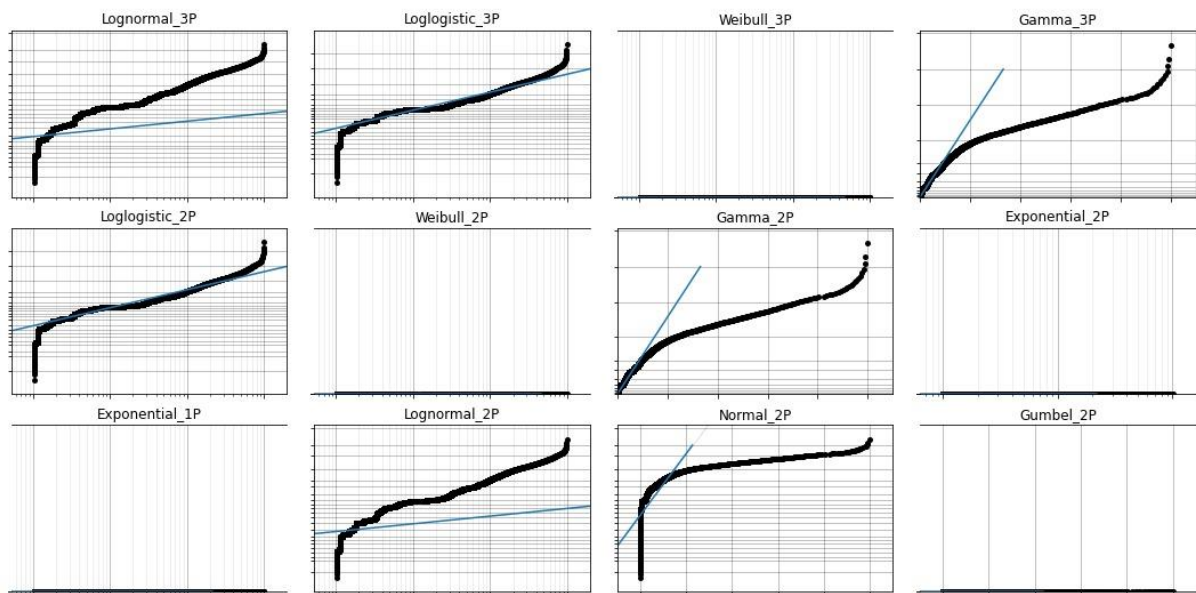
CDF of Runtimes time:

All Users

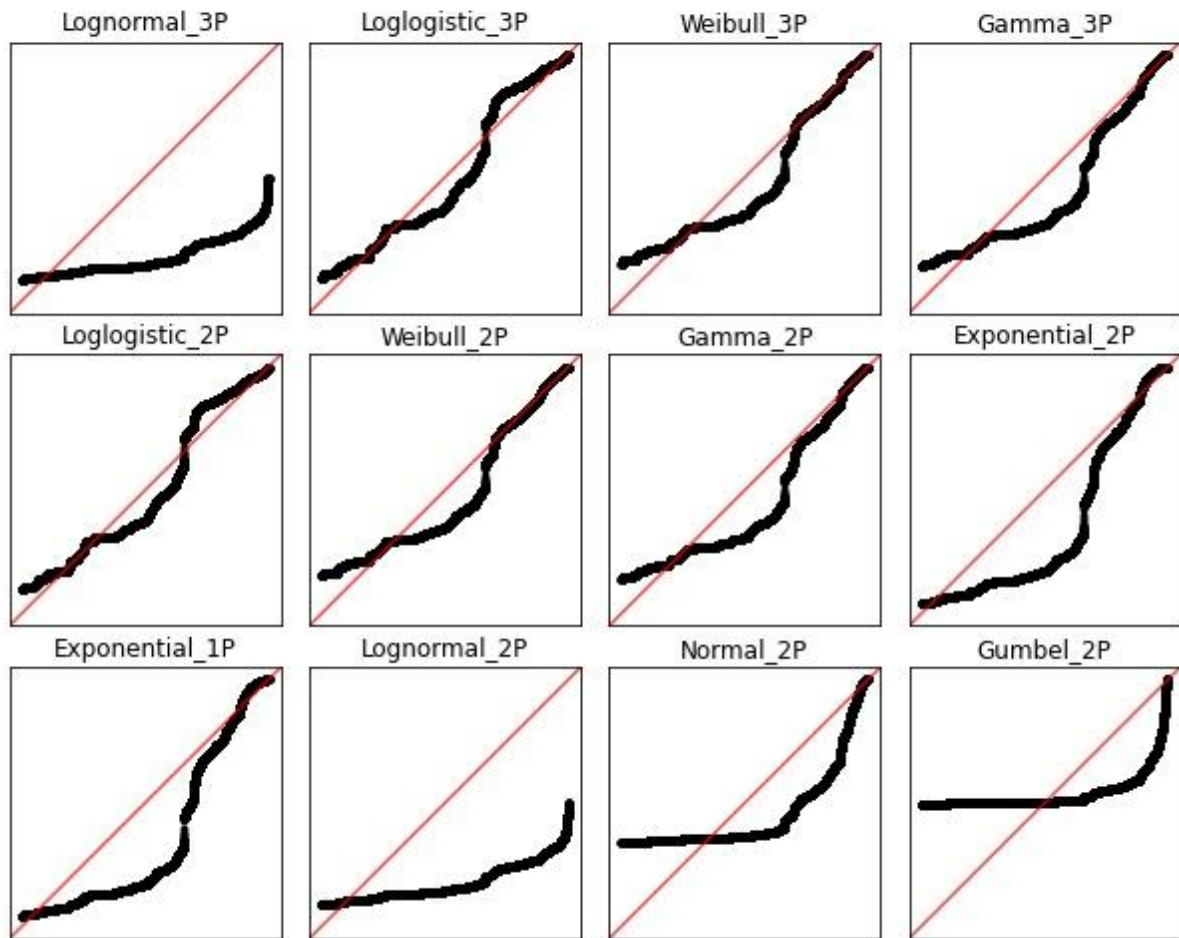
Histogram plot of each fitted distribution



Probability plots of each fitted distribution

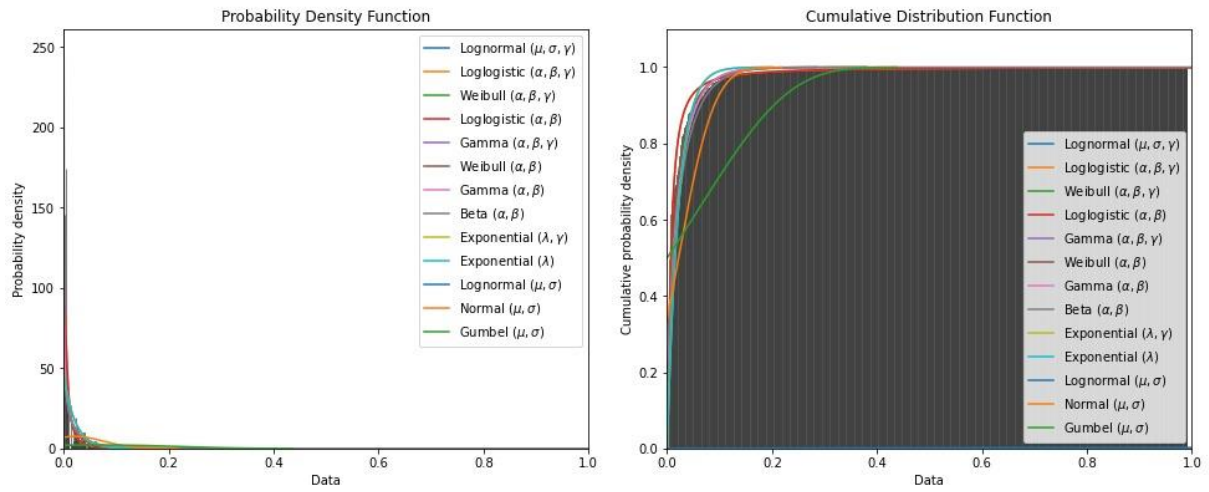


Semi-parametric Probability-Probability plots of each fitted distribution
 Parametric (x-axis) vs Non-Parametric (y-axis)

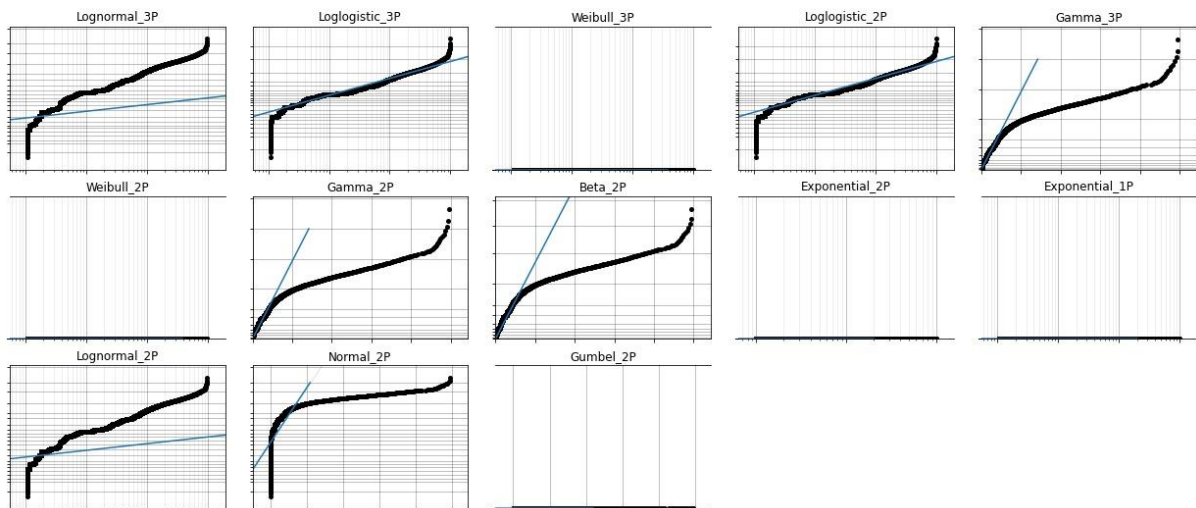


Other Users

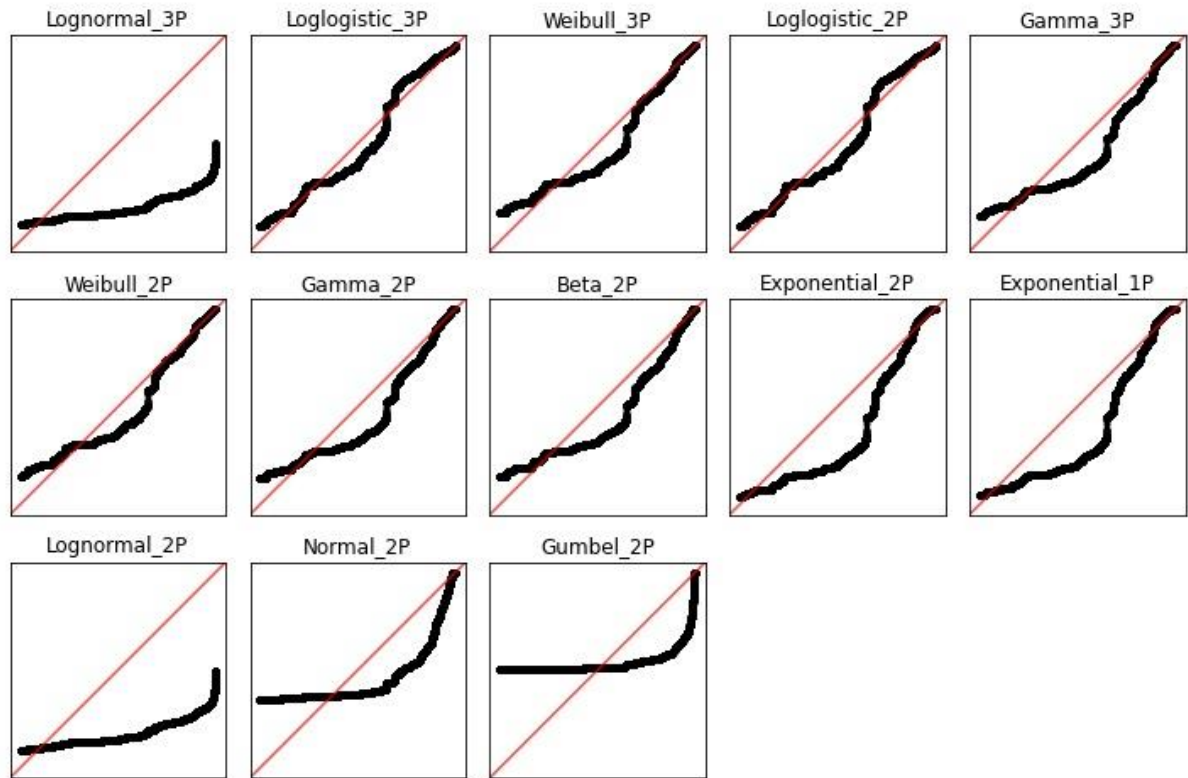
Histogram plot of each fitted distribution



Probability plots of each fitted distribution

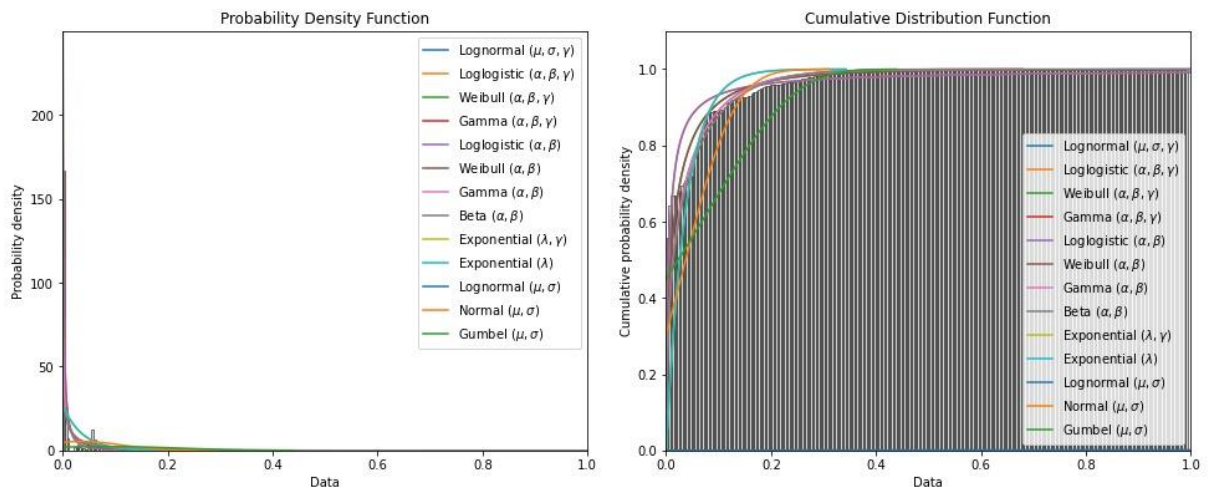


Semi-parametric Probability-Probability plots of each fitted distribution
 Parametric (x-axis) vs Non-Parametric (y-axis)

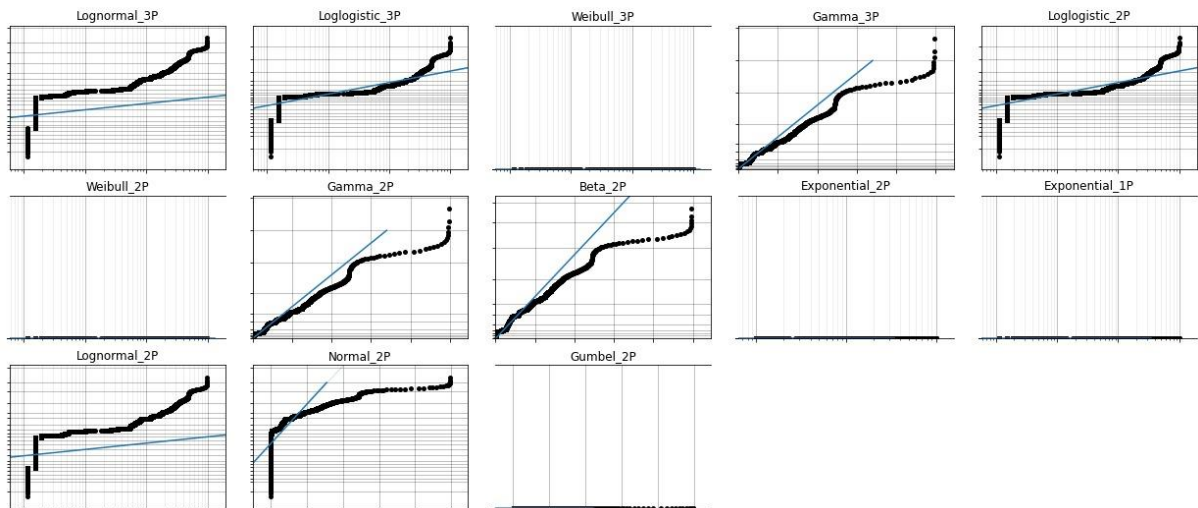


Special User

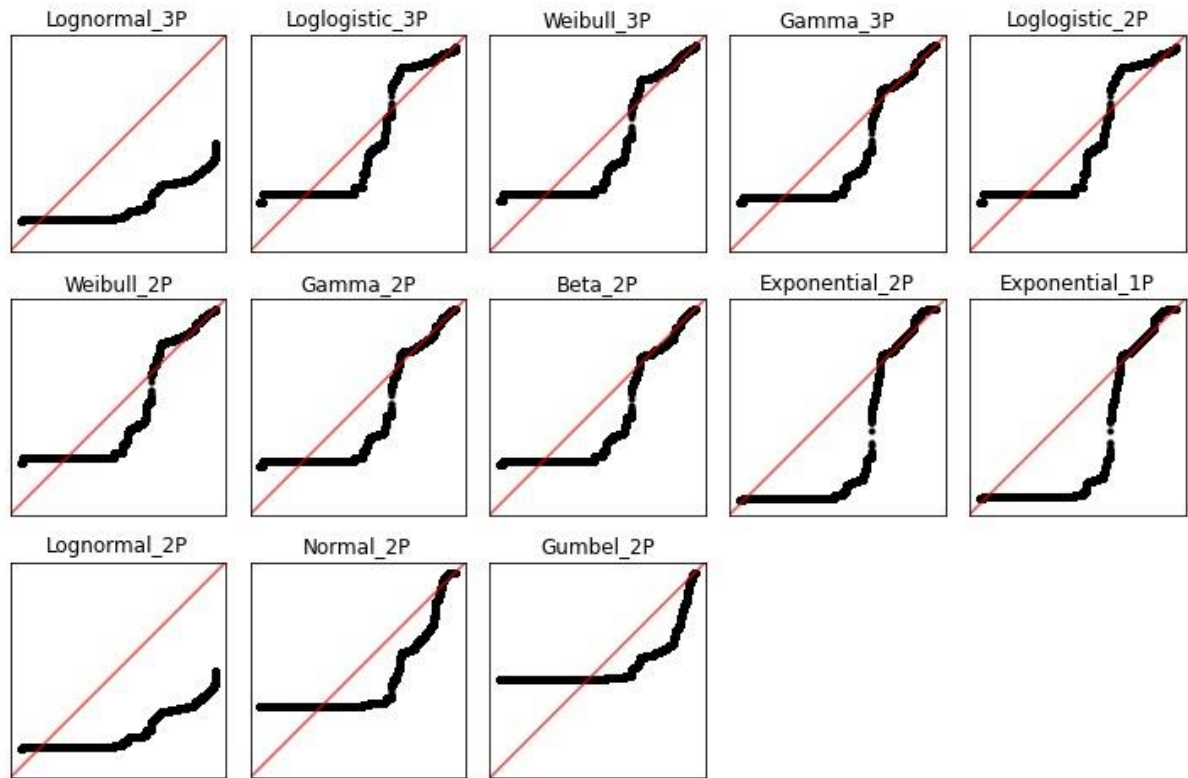
Histogram plot of each fitted distribution



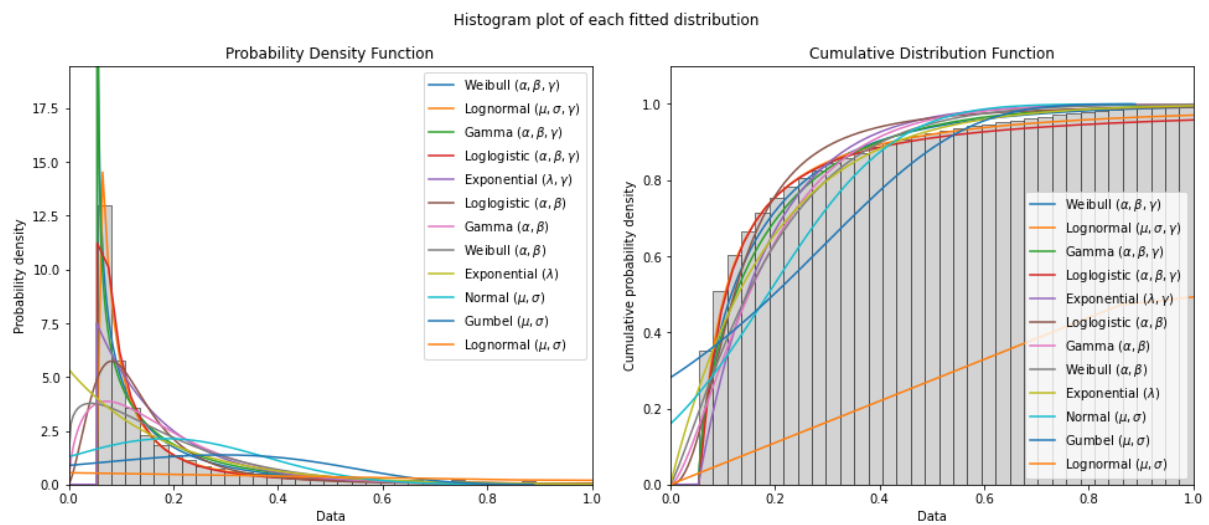
Probability plots of each fitted distribution



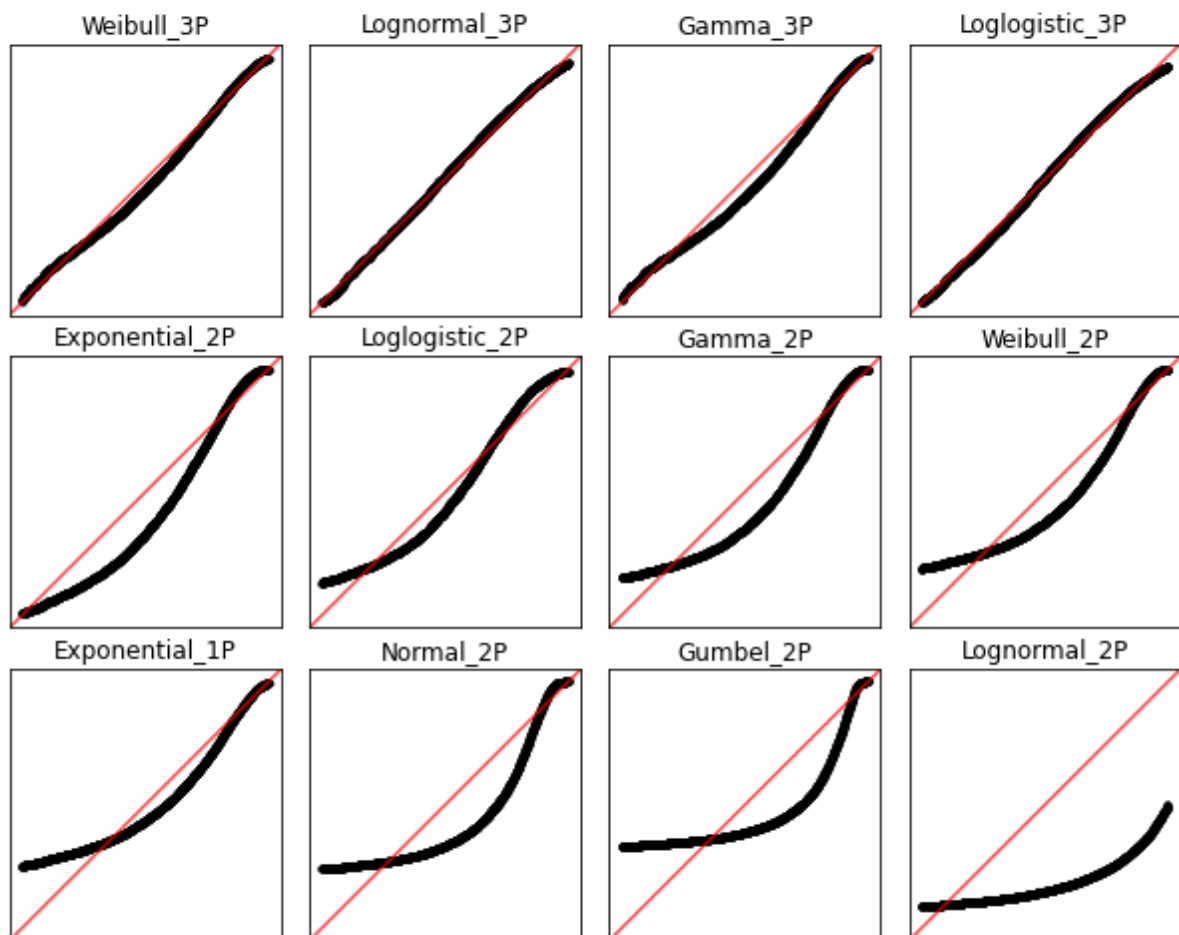
Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)



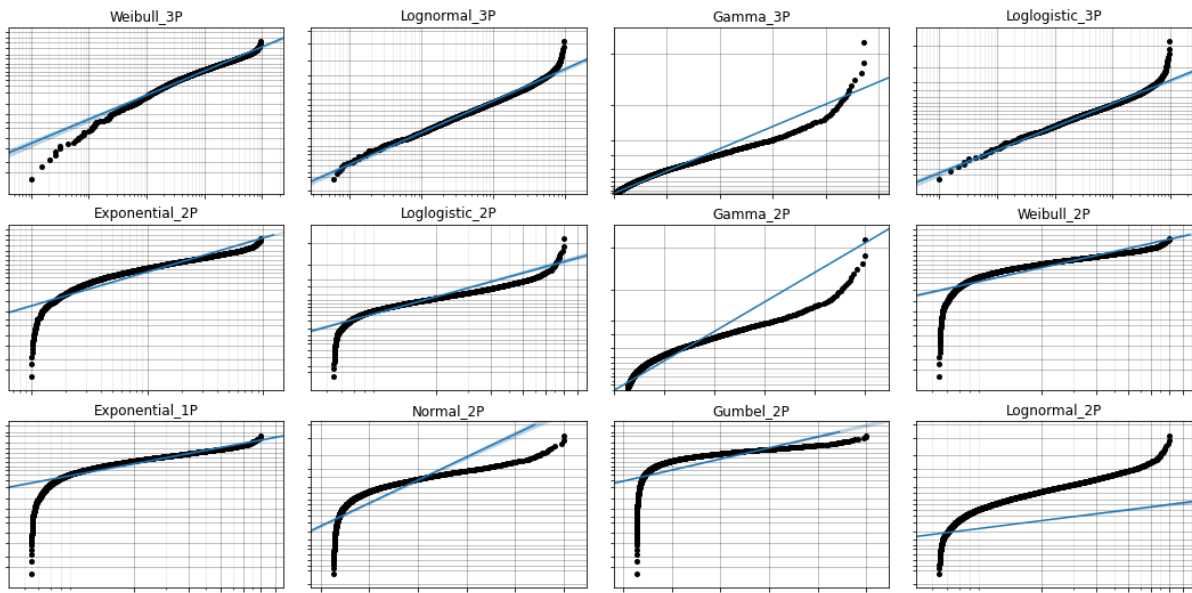
CDF of Interarrival time:



Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)



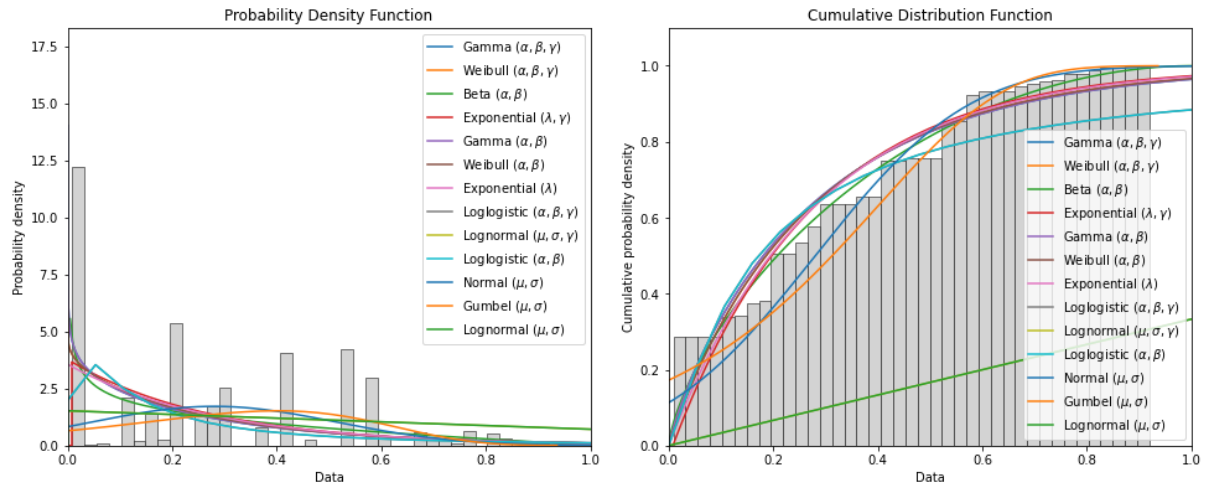
Probability plots of each fitted distribution



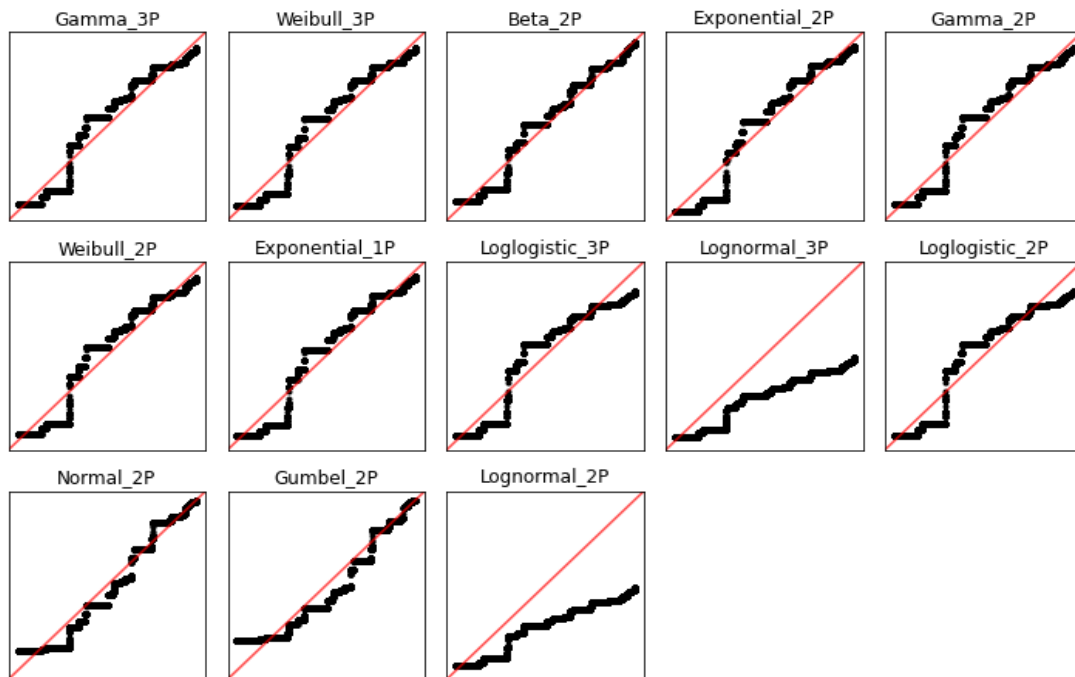
MATLAB-Log File Graphs

CDF of Runtimes:

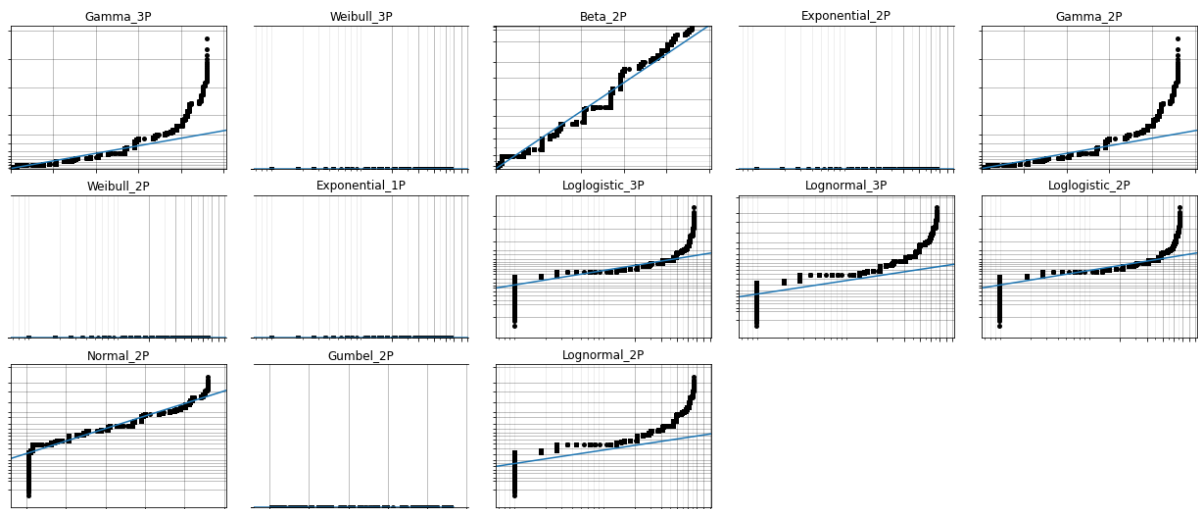
Histogram plot of each fitted distribution



Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)

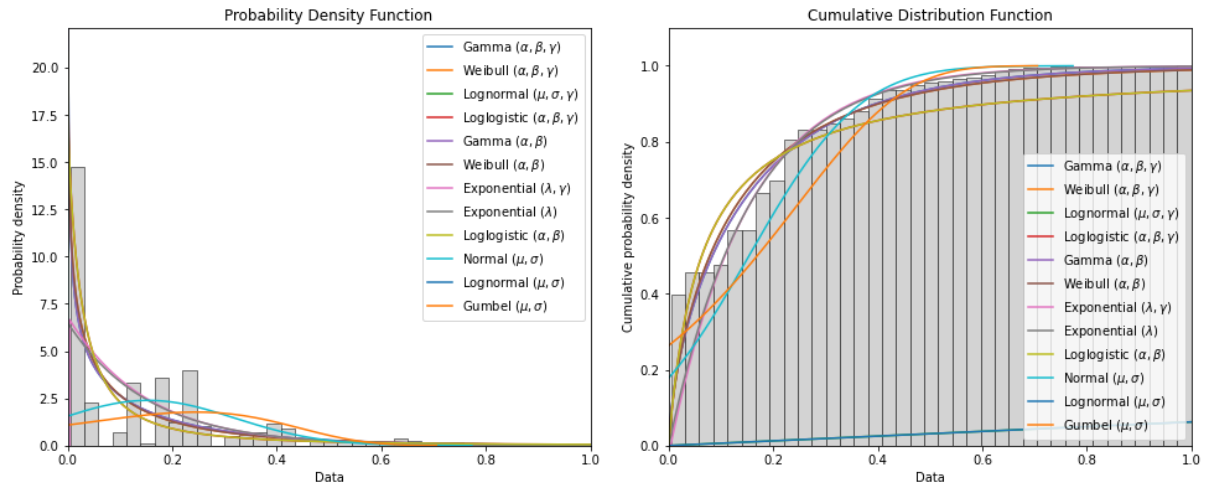


Probability plots of each fitted distribution

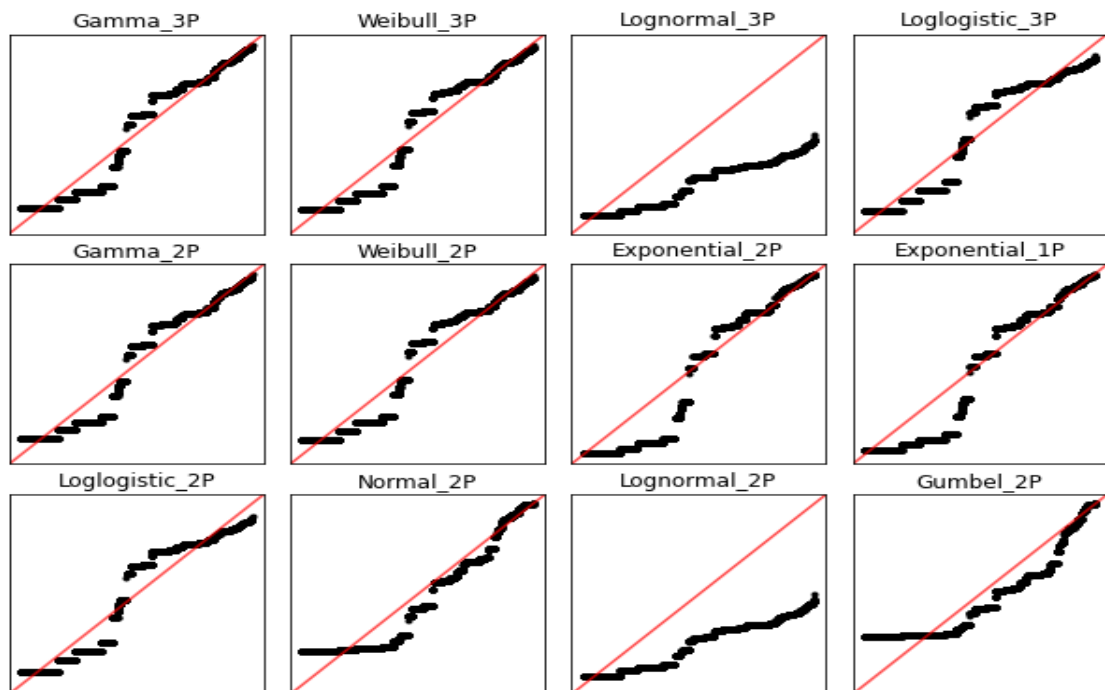


CDF of Interarrival time:

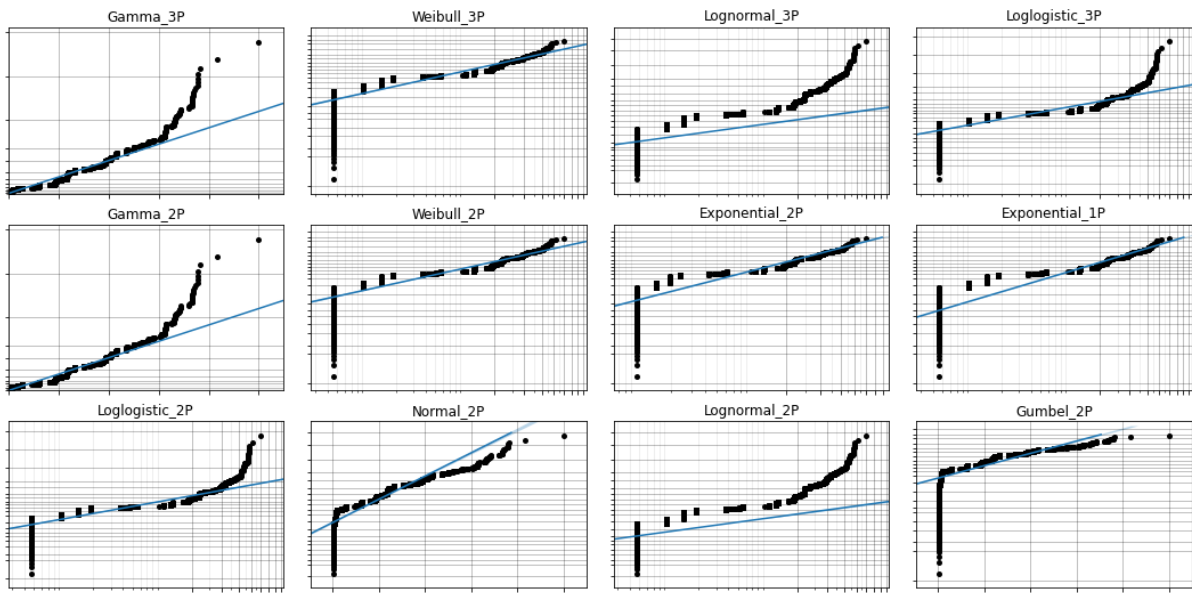
Histogram plot of each fitted distribution



Semi-parametric Probability-Probability plots of each fitted distribution
Parametric (x-axis) vs Non-Parametric (y-axis)



Probability plots of each fitted distribution

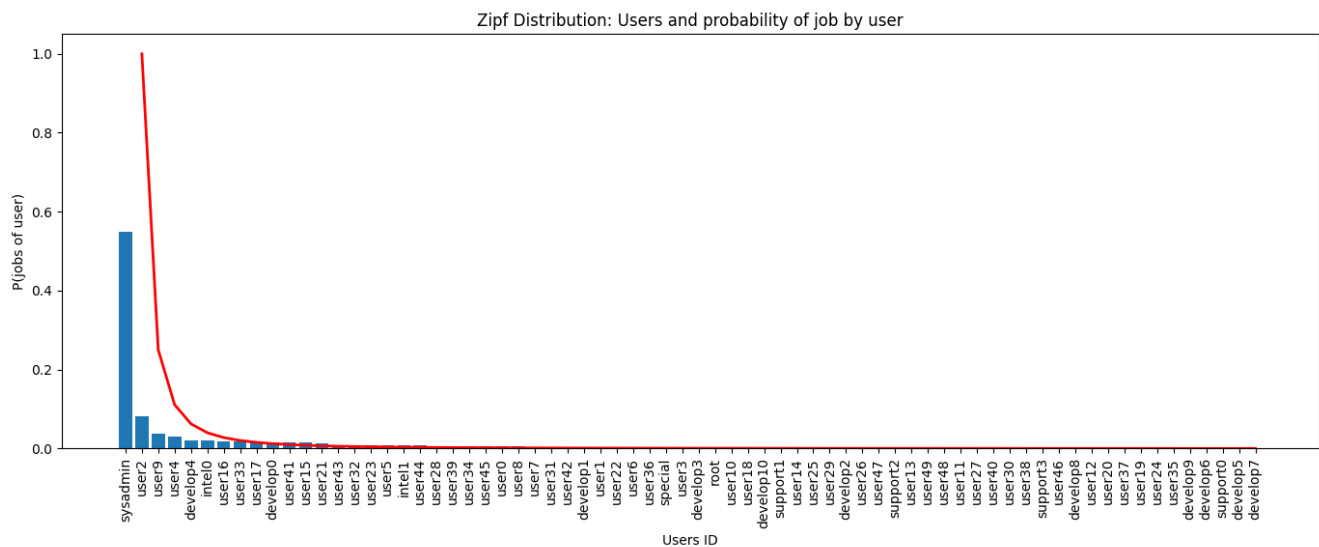


Stage 5:

נראה שגרף מספר 6 (כמות jobs לכל user במערכת) יכול להיות מתואר לפי התפלגות Zipf.

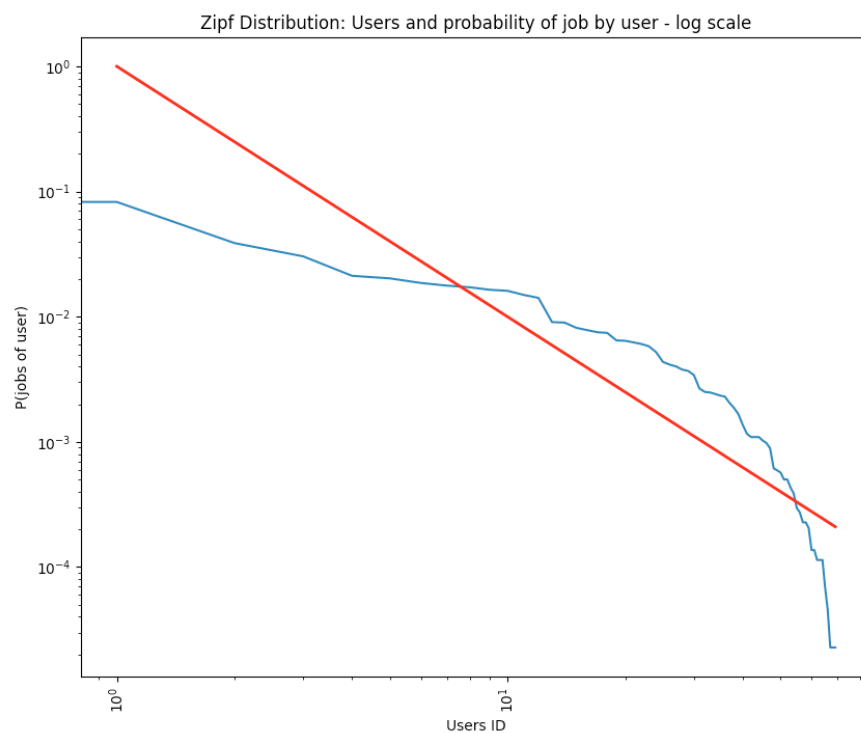
בדומה לגרף Zipf של ההסתברות הופעות מילים בקטע טקסט, נייצר גרף לפי אותו אופן. באמצעות גרף מספר 6 אשר מתאר לנו את כמות ה-jobs לכל user, נוכל למצוא את ההסתברות להופעת job של user ספציפי.

נציג זאת בגרף:



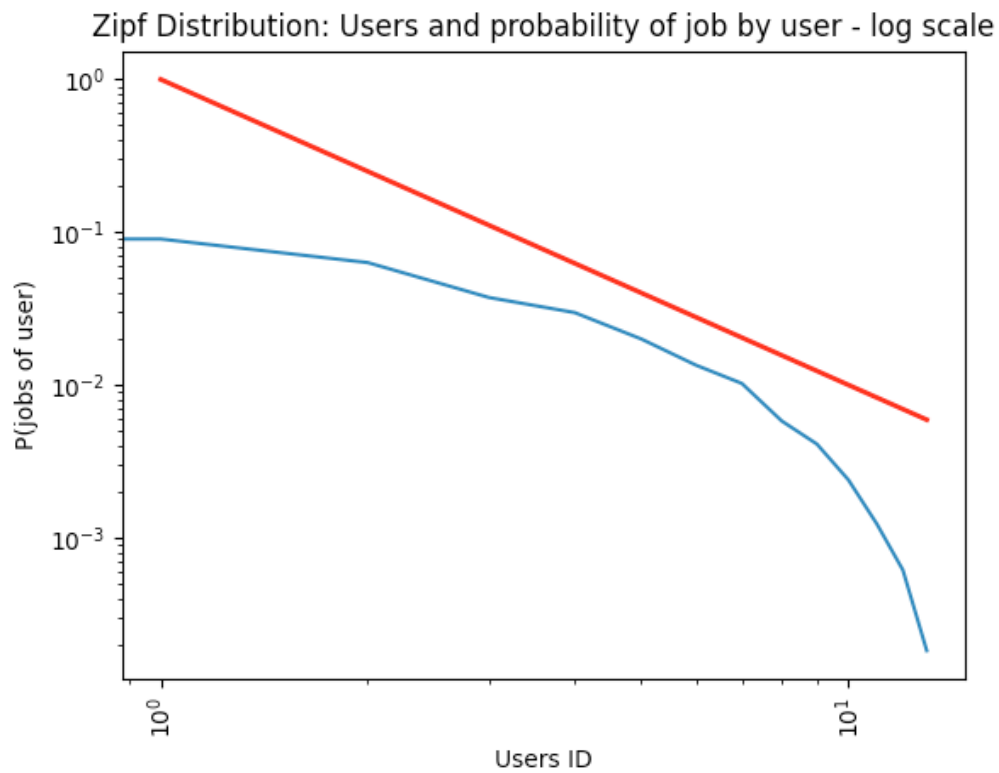
ציר ה-x יהיה המספר הסידורי של ה-users, וציר ה-y יהיה ההסתברות להופעת job מאותו user. הקו האדום מתאר את הפונקציה שנוצרת משימוש בהתפלגות Zipf על הערכים הנ"ל.

בשביל לראות יותר בנוחות את הגרף, נהפוך את העמודות לקו רציף ונציג את צירים בסקאלה לוגריתמית:

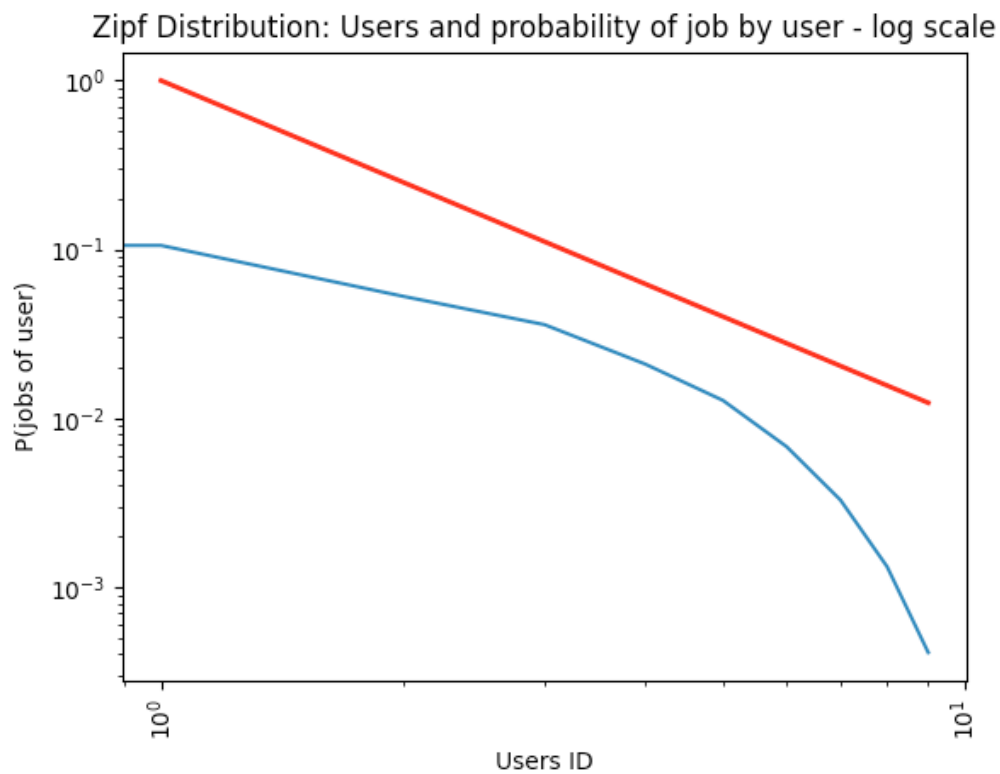


ניתן לראות שההתאמה לא טובה כפי שציפינו. ננסה לקבץ מספר דגימות יחדיו בשביל התאמה טובה יותר.

בגרף הבא, נעשה קיבוץ על 5 דגימות. ניתן לראות שישנו שיפור קל.



ננסה לקבץ כמות גדולה יותר של דגימות – הפעם נקבץ 7 דגימות ונראה את הגרף הבא:



נשים לב שה-fitting בגרף עם קיבוץ של 7 דגימות יותר מתאים לפונקציית ה-Zipf.

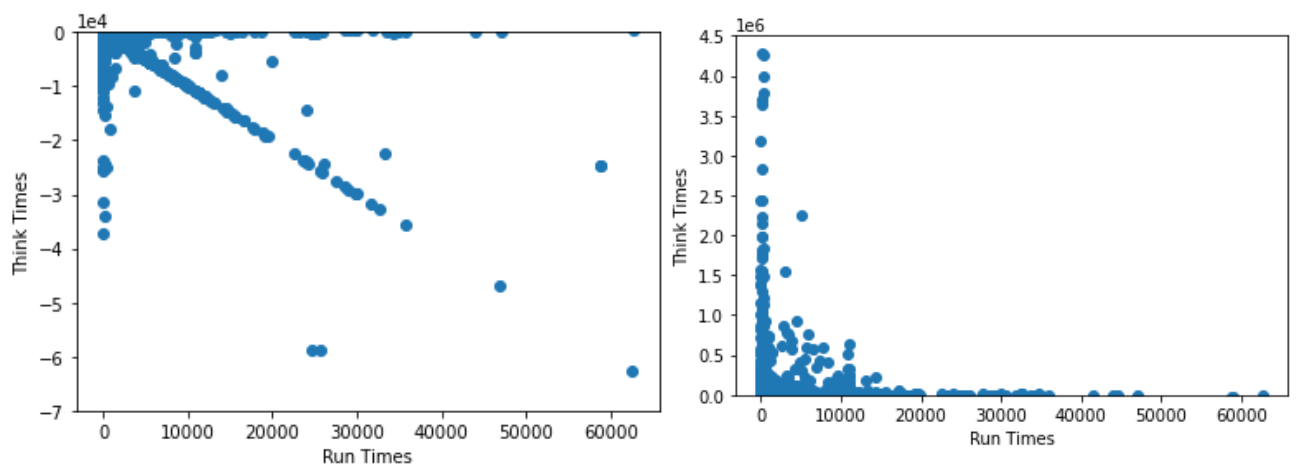
Stage 6:

NASA

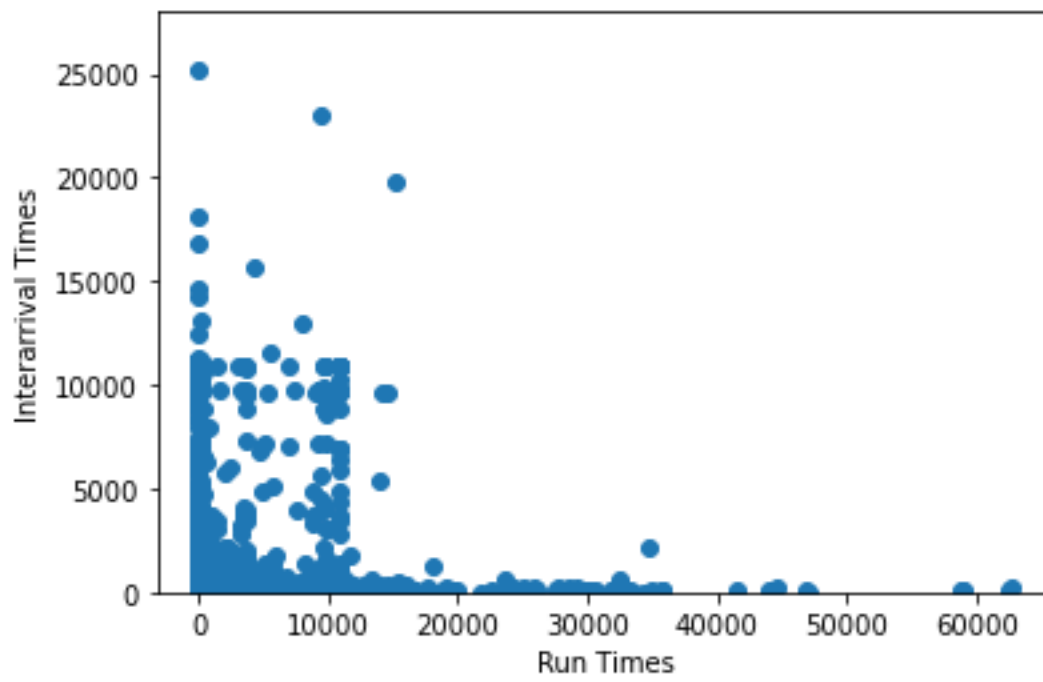
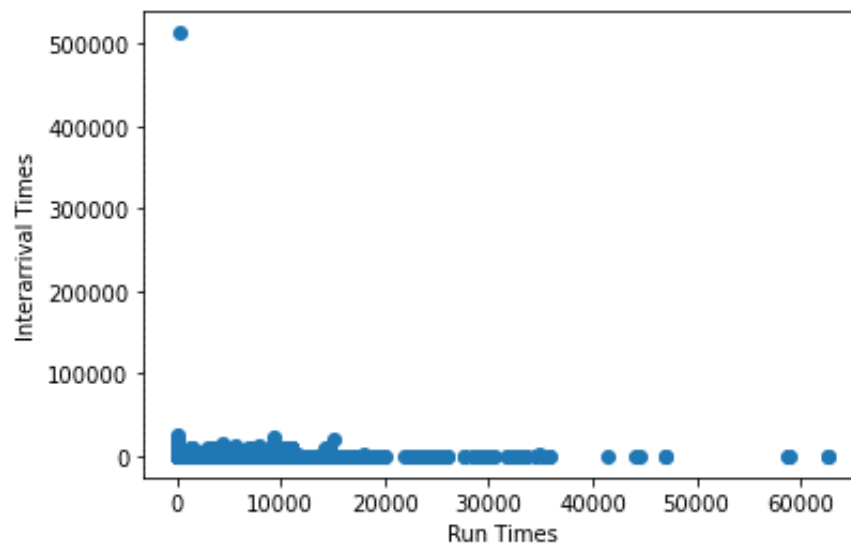
If we look at the CDF graph of the run times, we can see that most of the jobs have run time less than 12,000 (less or more).

On the other hand, users in NASA don't like to wait until the previous job is completed and they start to run in parallel with the previous jobs, as a result, we get that its not a must that longer jobs have longer interarrival time.

This can be also noticed in the CDF graph of interarrival times.

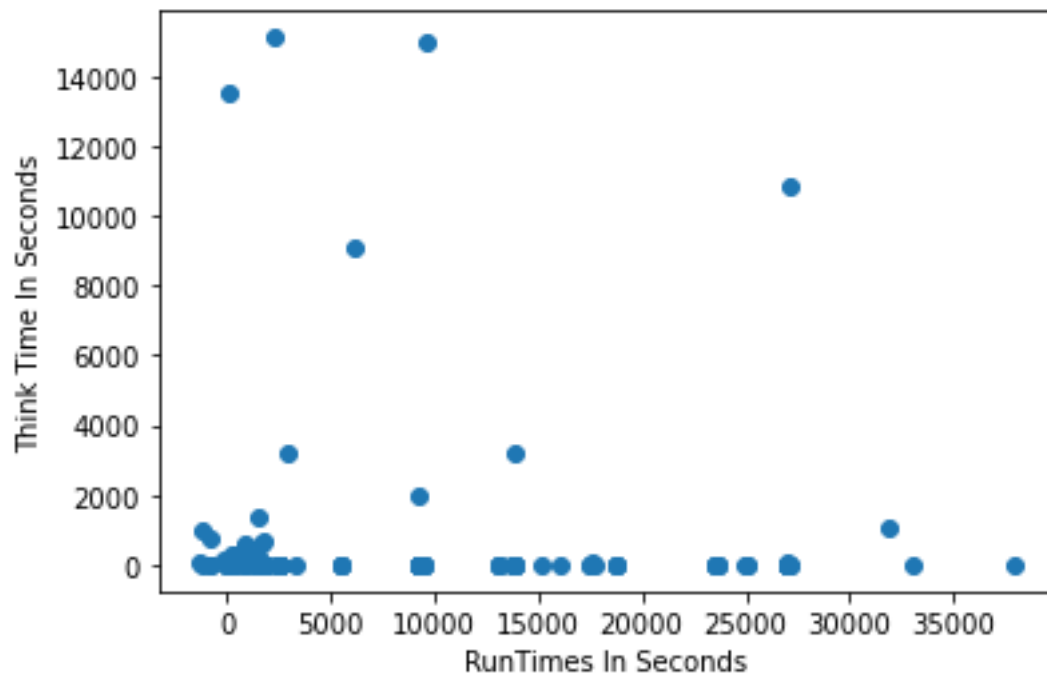


```
In [3]: runfile('C:/Users/elias/OneDrive/Documents/GitHub/SWF-Parser/main_parser.py', wdir='C:/Users/elias/OneDrive/Documents/GitHub/SWF-Parser')
Reloaded modules: RowClass
Correlation Coefficient between run times and think times:
0.0133
Correlation Coefficient between job sizes and run times:
0.1997
Correlation Coefficient between run times and interarrivals times:
0.0211
In [4]: |
```

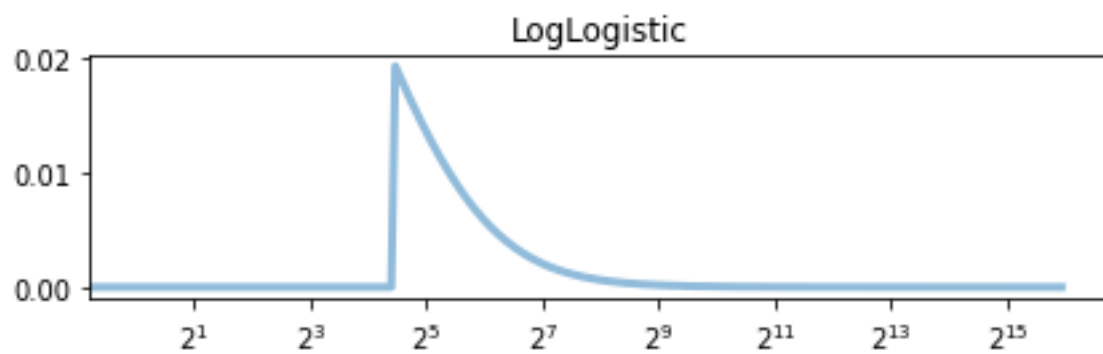
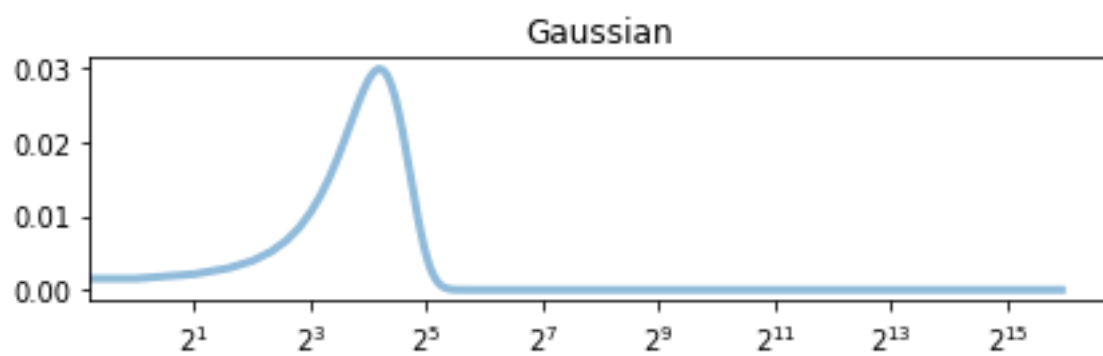
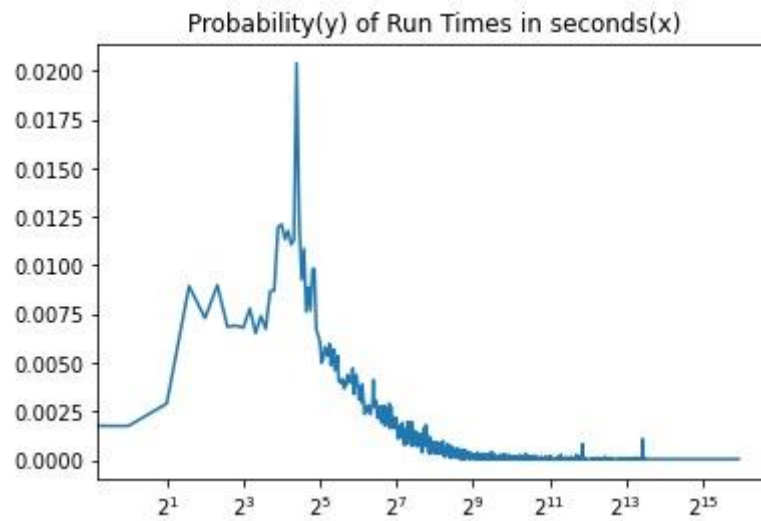


Matlab

Run Times – Think Times Correlation Coefficient: -0.06555923348325185



Stage 7:



The two distributions

