Çağrı Merkezi Kuyruk Simülasyonu

1- Problemin Tanımı ve Amaçlar

Çağrı merkezlerinde müşteri trafiği dalgalıdır; bazı zaman dilimlerinde yoğunluk birikerek uzun bekleme sürelerine yol açar. Mevcut veri setimizde (560 gözlem) geliş, hizmete başlama ve bitiş zamanları ölçülmüş durumda. **Problemimiz**, çağrı merkezinde müşteri bekleme sürelerini ve kuyruk uzunluklarını azaltarak hizmet kalitesini artırmak ve kaynak kullanımını optimize etmek.

Neden Önemli: Yapacağımız simülasyon Verimlilik & Maliyet ve Karar Destek gibi konularda bize ışık tutacak. Yani elde edilen model genellenerek ölçeklenebilir bir karar-destek aracı oluşturulacak.

Amaç: Müşteri bekleme sürelerini azaltmak, kuyruk uzunluklarını kontrol altına almak ve senaryo analizleri yapmak.

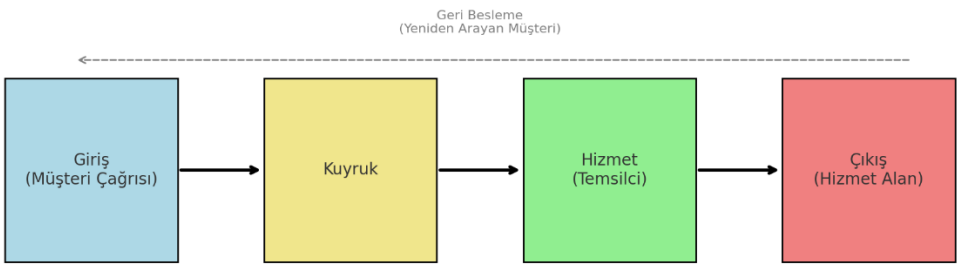
2- Projenin Planı: Verinin kaynağı Kaggle, yapılması gerekenler haftalık;

1.Hafta: Problemi, sistemi ve modeli tanımlama

2.Hafta: Veriyi analiz etme, model geliştirme…

3.Hafta: Doğrulama, çıktıyı çözümleme…

3- Sistemi Tanımlama ve Çözümleme:



Temel Unsurlar: Müşteriler, sunucular(temsilciler), telefonlar, kulaklıklar…

Temel Olaylar: Müşterinin araması(giriş), hizmetin başlaması, görüşmenin bitişi(çıkış)

Değişkenler: arrival_time: Müşteri geliş zamanı

start_time: Hizmetin başlama zamanı

finish_time: Hizmetin bitiş zamanı

wait_time: Bekleme süresi

queue_length: Anlık kuyruk uzunluğu

service_time: (start_time ile finish_time farkı) hesaplanabilir

Varsayımlar: Hizmet tipi aynıdır. Kuyruk kapasitesi sınırsızdır.

Performans Ölçüleri: Bekleme süresinin, kuyruk uzunluğunun olabildiğince az olması.

4- Model Tanımlama:

Modeli çağrı merkezi yöneticileri/vardiya planlayıcıları kullanabilir. Bana çağrı sıklığı, yoğun saatler gibi verileri verip; çağrı merkezinin verimli çalışabileceği bir denge oluşturulmasını, yetersiz temsilci varsa bunu fark etmemi ve bazı saatlerde kuyruk artışını önceden tahmin edebilmemi bekleyebilirler.

Söyleyebileceğimiz bazı varsayımlar: Tüm temsilciler eşit hızda çalışıyor, farklı becerileri yok. Tüm müşteriler aynı önceliğe sahip. Müşteriler sırayı terk etmiyorlar.

Hipotezler: Müşteri çağrıları bağımsız ve rastgele gelir, Hizmet süreleri rastgele ama belirli bir dağılıma uyar.

5- Veri Toplama ve Analiz:

arrival_time: Nicel Sürekli

start_time: Nicel Sürekli

finish_time: Nicel Sürekli

wait_time: Nicel Sürekli

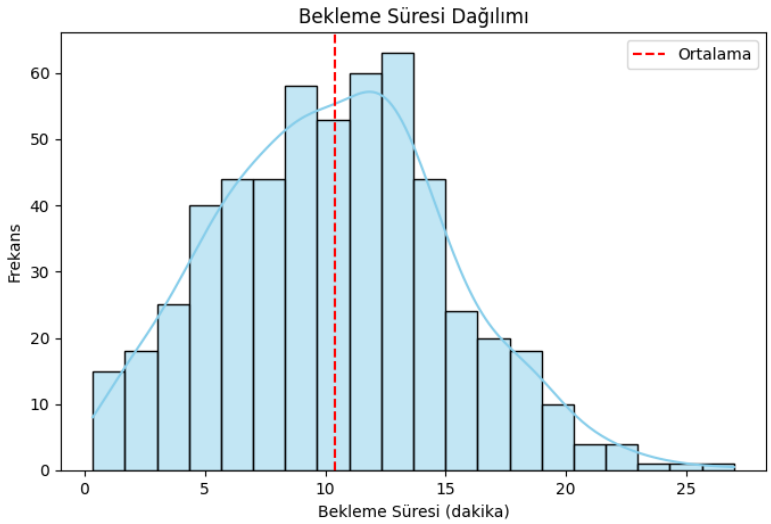
queue_length: Nicel Kesikli

Verimi, Python'da hata denetlemesine soktuktan sonra aldığım bazı sonuçlar:

Hata Denetimi Sonuçları:
- Negatif bekleme süresi olan kayıt sayısı: 13
- Geliş saatinden önce başlamış hizmetler: 0
- Hizmetten önce bitmiş çağrılar: 0

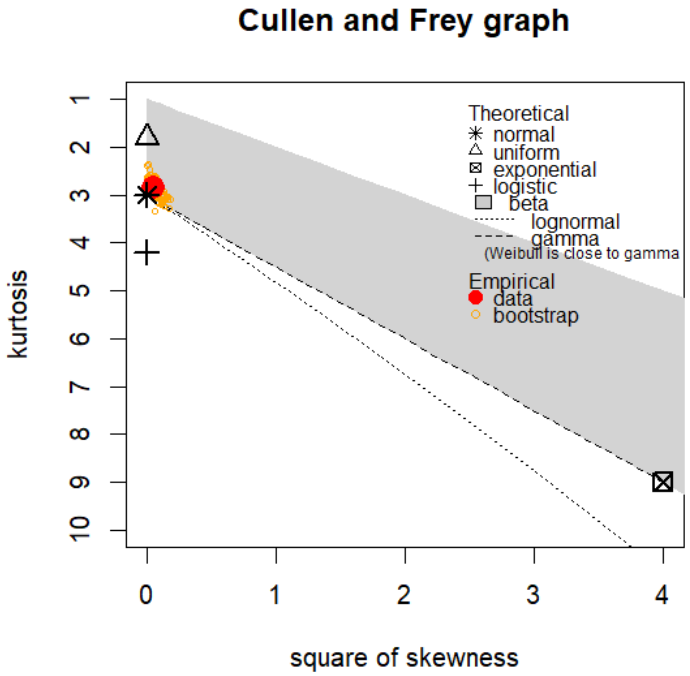
✅ Temizlenmiş veri kümesinde kalan satır sayısı: 547

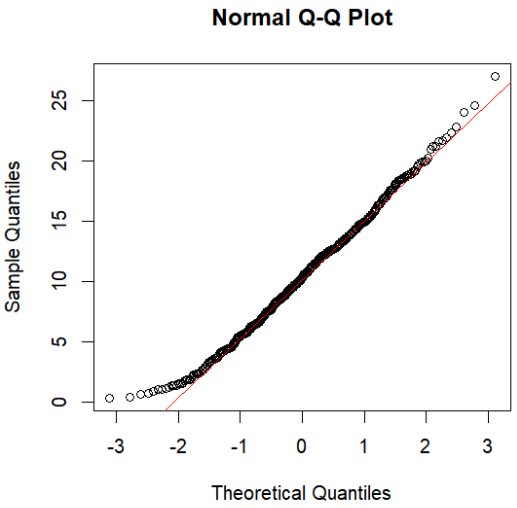
Bizim için çok önemli olan bekleme süresi:

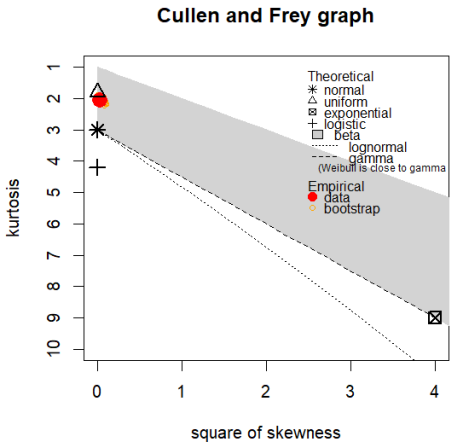


	wait_time
count	547.000000
mean	10.377806
std	4.771044
min	0.340000
25%	6.845000
50%	10.360000
75%	13.420000
max	26.980000

wait_time değişkeni, küçük sapmalara rağmen en iyi şekilde normal dağılıma uymaktadır.
Alternatif dağılımların hiçbiri hem istatistiksel hem görsel olarak daha iyi sonuç vermiyor.







6- Model Geliştirme:

Sistemimizi/modelimizi aşağıdaki değişkenlerle tanımlayabiliriz:

A_i : i. müşterinin geliş zamanı

S_i : i. müşterinin hizmete başlama zamanı

F_i : i. müşterinin hizmeti bitirme zamanı

$W_i = S_i - A_i$: i. müşterinin bekleme süresi

L_i : i. müşteri geldiğinde kuyrukta bulunan kişi sayısı (kuyruk uzunluğu)

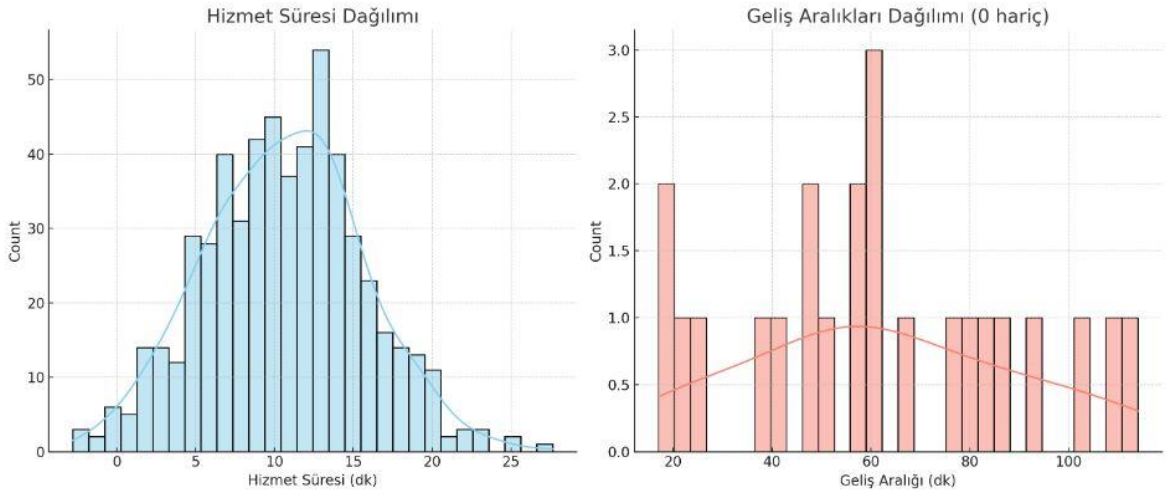
Ortalama bekleme süresi:

$\bar{W} = (1 / n) \times \sum W_i$ (i = 1'den n'e)

Ortalama kuyruk uzunluğu:

$\bar{L} = (1 / n) \times \sum L_i$

Geliş aralıkları üstel, hizmet süresi de bekleme süresi gibi normal dağılıma veya üstel uyuyor gibi gözükmekte **Histograma göre**.



7- Doğrulama: Bu FIFO (ilk gelen ilk hizmet alır) tek sunuculu (M/M/1) kuyruk modelidir. Sonuçlar simülasyon çıktısıdır, geliş ve hizmet süreleri üstel dağılımdan rastgele çekilmektedir.

arrival_time	start_time	finish_time	interarrival_time	service_time	wait_time
0	1.41	1.41	1.51	1.41	0.10	0.00
1	10.44	10.44	27.96	9.03	17.52	0.00
2	14.39	27.96	36.89	3.95	8.93	13.57
3	17.13	36.89	38.08	2.74	1.19	19.76

4	17.64	38.08	39.08	0.51	1.00	20.45
5	18.14	39.08	40.10	0.51	1.01	20.94
6	18.32	40.10	41.91	0.18	1.81	21.77
7	24.36	41.91	45.63	6.03	3.72	17.55
8	27.12	45.63	48.46	2.76	2.83	18.52
9	30.81	48.46	50.18	3.69	1.72	17.65

Hipotezler:

H1: Geliş oranı hizmet oranından düşükse ($\lambda < \mu$), kuyruk oluşmaz.

H2: Geliş oranı hizmet oranına eşit veya büyükse ($\lambda \geq \mu$), bekleme süresi hızla artar.

H3: Simülasyon tekrar sayısı arttıkça sonuçlar daha kararlı hale gelir.

Tekrar Sayısı:

- Her senaryo için 30 tekrar yapılacak.
- Ortalama bekleme süresi ve kuyruk uzunluğu analiz edilecek.

Senaryo	**Geliş Oranı λ**	**Hizmet Oranı μ**	**Beklenen Yoğunluk ρ**	**Beklenen Durum**
S1	0.1	0.2	0.5	Sistem rahattır
S2	0.15	0.15	1.0	Sistem tam kapasite çalışır
S3	0.2	0.1	2.0	Kuyruklar büyür

Sonuç: Pythonda(kodunu da paylaştım) yaptığım simülasyon

Senaryo	**Geliş Oranı (λ)**	**Hizmet Oranı (μ)**	**Tekrar Sayısı**	**Her Tekrarda Ne Yapıldı?**
S1	0.10	0.20	30	100 müşterilik simülasyon
S2	0.15	0.15	30	100 müşterilik simülasyon
S3	0.20	0.10	30	100 müşterilik simülasyon

Senaryo	λ	μ	ρ	Ortalama Bekleme	Ortalama Kuyrukta Kalma Oranı
0 S1	0.10	0.20	0.5	4.35	0.48
1 S2	0.15	0.15	1.0	43.59	0.90

2	S3	0.20	0.10	2.0	252.53	0.98

PS C:\vscode python>

Senaryo karşılaştırma sonuçlarına göre:

- S1 senaryosunda (λ=0.1, μ=0.2), sistem rahat çalışmakta, bekleme süresi düşük ve istikrarlıdır.

- S2 senaryosunda (λ=0.15, μ=0.15), sistem tam kapasitede çalışmakta ve bekleme süresi dalgalı seyretmektedir.

- S3 senaryosunda (λ=0.2, μ=0.1), sistem aşırı yük altında olup, kuyruklar ve bekleme süreleri dramatik şekilde artmaktadır.

Bu analiz, sistemin performansını doğrudan etkileyen en önemli faktörün trafik yoğunluğu (ρ = λ / μ) olduğunu göstermektedir.

Optimum çalışan sayısını bulalım;

ÇALIŞAN SAYISI ANALİZİ (M/M/c)

Simülasyon Parametreleri:

- Toplam müşteri sayısı: 100

- Geliş hızı (λ): 0.8

- Hizmet hızı (μ): 0.5

- Denenen sunucu sayıları (c): 1 – 5

Sonuçlar:

Sunucu Sayısı (c)	Ortalama Bekleme Süresi
1	39.682893
2	3.394125
3	0.707059
4	0.138184
5	0.013632

Yorum:

Sunucu sayısı arttıkça sistemdeki ortalama bekleme süresi ciddi oranda düşmektedir.

Tek sunuculu sistem (c=1), yüksek trafik yoğunluğu nedeniyle darboğaza girmekte ve ortalama 39.68 birim bekleme süresi oluşmaktadır.

İki sunucuda bu süre 3.39 birime düşerken, üç sunucu ve üzeri durumda sistemdeki bekleme neredeyse sıfırlanmaktadır.

Bu durum, c=3 sunucunun sistemin dengede çalışması için yeterli olduğunu göstermektedir.

Bu sayının üzerindeki sunucular, maliyeti artırmak dışında anlamlı bir performans artışı sağlamamaktadır.

Sonuç olarak, bu sistem için optimum sunucu sayısı **3** olarak önerilmektedir.