HACETTEPE
ÜNİVERSİTESİ

İSTATİSTİK BÖLÜMÜ

Sentetik Veri Üretimi ve Otokorelasyonun İstatistiksel Analizi: Python ve R Uygulamalarıyla Karşılaştırmalı Bir Yaklaşım

İST 493 SEMİNER

Doç. Dr. Melike Bahçecitapar

BÜŞRA SARI

İçindekiler

1. Giriş

Veriye dayalı karar verme süreçlerinin temelinde doğru, güvenilir ve analiz edilebilir veri kaynakları yer alır. Ancak özellikle etik kaygıların yüksek olduğu sağlık, finans ve güvenlik gibi alanlarda gerçek veriye ulaşmak her zaman mümkün değildir. Bu noktada, sentetik veri üretimi hem bilimsel araştırmalarda hem de yapay zekâ modellerinin eğitimi gibi uygulamalarda kritik rol oynamaya başlamıştır [1].

Sentetik veri, gerçek dünyada gözlemlenmiş verilere benzer özellikler taşıyan, ancak tamamen yapay olarak üretilmiş veri setleridir. Bu veriler, gizliliği ihlal etmeksizin veri paylaşımına olanak tanıdığı için mahremiyetin korunması, test senaryolarının çeşitlendirilmesi ve model doğrulama gibi amaçlarla sıklıkla kullanılmaktadır.

Bu çalışma kapsamında, Python ve R programlama dillerinde Normal, Poisson ve Gamma dağılımlarına uygun olarak sentetik veriler üretilmiş; hem univarite hem de longitudinal yapıdaki bu veriler üzerinde otokorelasyon analizleri gerçekleştirilmiştir. Ayrıca, sağlık alanına yönelik bir senaryoyla (hasta tansiyon takibi) uygulama örneği oluşturularak veri üretim sürecinin gerçek hayattaki yansımaları da modellenmiştir.

Çalışma aşağıdaki sorulara cevap aramaktadır:

- Farklı dağılımlara göre sentetik veriler nasıl oluşturulur?
- Longitudinal verilerde zaman bağımlılığı nasıl simüle edilir?
- Otokorelasyon istatistiksel olarak nasıl test edilir?
- Bu süreçler Python ve R ortamında nasıl uygulanabilir?

Bu bağlamda sentetik veriler, hem istatistiksel modellemeler hem de etik kısıtların bulunduğu alanlarda kullanılabilirliği açısından büyük potansiyel taşımaktadır [1][2].

2. Teorik Çerçeve

2.1 Sentetik Veri Nedir?

Sentetik veri, gerçek dünyada gözlemlenmiş veri yapılarının istatistiksel özelliklerini taşıyan, ancak yapay olarak üretilmiş veri setleridir. Bu veriler, özellikle mahremiyetin korunması gereken alanlarda (örneğin sağlık ve finans) araştırma yapmayı mümkün kılar. Gerçek verilere benzer dağılımlarla oluşturulan sentetik veriler, model eğitimi, test senaryoları ve algoritma doğrulama gibi pek çok amaçla kullanılabilir [1].

Sentetik veri üretimi; istatistiksel modelleme, olasılık dağılımları ve makine öğrenmesi algoritmaları kullanılarak gerçekleştirilebilir. Üretilen veri seti, yapısı itibariyle gerçek veriyle aynı biçimde analiz edilebilir, ancak kişisel bilgi içermez. Bu özelliği sayesinde etik sorunları ortadan kaldırır ve veri paylaşımını kolaylaştırır [1].

Aşağıdaki örnek kod, Python'da ortalaması 100, standart sapması 15 olan 1000 adet Normal dağılımlı sentetik veri üretmektedir:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

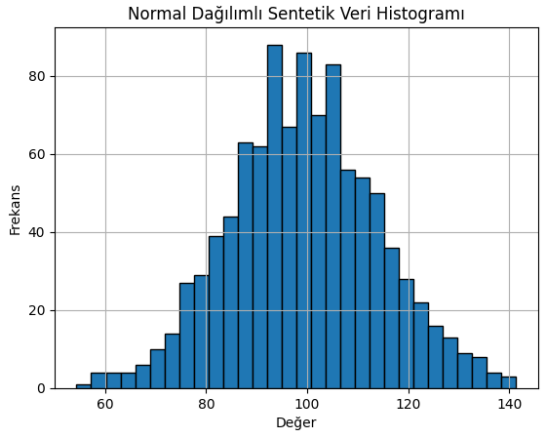
```
np.random.seed(0)
veri = np.random.normal(loc=100, scale=15, size=1000)
df = pd.DataFrame({'SentetikVeri': veri})
print(df.head())
```

Bu kod, temel düzeyde sentetik veri oluşturmayı sağlar. np.random.normal() fonksiyonu sayesinde veri, belirli bir dağılım (Normal) altında oluşturulur. Bu işlem, araştırma sürecinde test edilecek analiz modellerinin önce yapay veriler üzerinde denenmesini sağlar.

Kodun çıktısında aşağıdaki gibi ilk beş gözlem yer alabilir:

	SentetikVeri
0	126.460785
1	103.002358
2	114.681070
3	138.613398
4	127.105985

Bu çıktılar üzerinden veri setinin ortalama etrafında simetrik dağıldığı görülmektedir. Verinin dağılımını daha iyi anlamak için histogram grafiği çizilebilir:



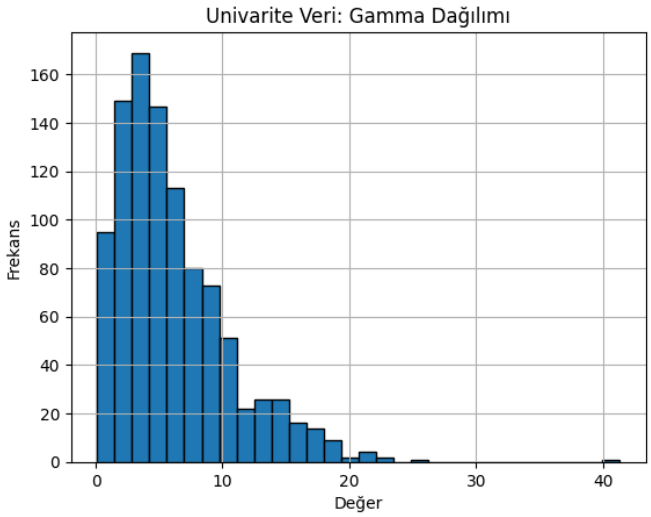
Bu grafik, veri değerlerinin büyük çoğunluğunun 85-115 arasında yoğunlaştığını ve dağılımın çan eğrisi şeklinde olduğunu gösterecektir.

Bu temel örnek, daha karmaşık senaryolarla birleştirildiğinde (örneğin zaman serisi, longitudinal yapı) sentetik verilerin ne kadar güçlü bir araç olabileceğini gösterir [1].

2.2 Veri Türleri: Univarite ve Longitudinal

Veri türlerinin doğru anlaşılması, sentetik veri üretiminin hem yapısını hem de analiz stratejilerini doğrudan etkiler. Bu bağlamda, bu çalışmada iki temel veri yapısı olan univarite ve longitudinal veri türleri üzerinde durulmuştur.

Univarite Veri: Sadece tek bir değişkenin değerlendirildiği veri türüdür. Genellikle dağılım analizi, merkezi eğilim ölçüleri ve varyans gibi temel istatistiksel tanımlayıcılarla incelenir. Aşağıda Python ile 1000 gözlemden oluşan bir univarite veri seti örneği verilmiştir:



Histogramda görüldüğü üzere veri sağa çarpık ve sürekli yapıya sahiptir. Bu tip verilerde medyan, ortalamadan daha küçüktür. Gamma dağılımı pozitif sürekli veriler için uygundur [3].

Longitudinal Veri: Aynı bireyden zaman içinde tekrarlı olarak toplanan veri türüdür. Longitudinal veri, zamana bağlı değişimi ve birey içi farklılıkları anlamada oldukça etkilidir. Aşağıda 5 bireyin 10 günlük gözlemlerinin oluşturulması örneği yer almaktadır:

```
import pandas as pd

np.random.seed(2)
subjects = 5
days = 10
values = np.random.normal(loc=100, scale=5, size=subjects * days)
df_long = pd.DataFrame({
  "Birey": np.repeat(np.arange(1, subjects + 1), days),
  "Gün": np.tile(np.arange(1, days + 1), subjects),
  "Ölçüm": values
})
print(df_long.head())
```

```
   Birey  Gün       Ölçüm
0      1    1   97.916211
1      1    2   99.718666
2      1    3   89.319020
3      1    4  108.201354
4      1    5   91.032822
```

Her birey için 10 gözlem yer almakta olup bu yapı, otokorelasyon ve varyans analizlerinde kullanıma uygundur. Longitudinal veriler zaman içi bağımlılık içerdiğinden

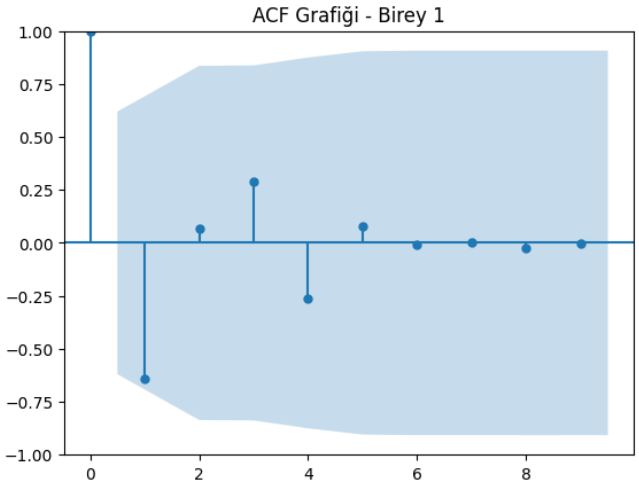
klasik bağımsızlık varsayımları ihlal edilebilir. Bu nedenle özel modelleme yöntemleri (örneğin karışık modeller) tercih edilir [4].

2.3 Otokorelasyon Nedir?

Otokorelasyon, zaman serisi ya da longitudinal veri analizlerinde sıklıkla karşılaşılan bir bağımlılık yapısıdır. Bir gözlemin, geçmişteki bir veya daha fazla gözlemle olan ilişkisinin derecesini ifade eder. Pozitif otokorelasyonda, bir dönemdeki yüksek değerin takip eden dönemde de yüksek olma ihtimali yüksektir. Negatif otokorelasyonda ise, bir gözlem yüksekse takip eden gözlemin düşük olması beklenir [3].

Otokorelasyonun varlığı, özellikle regresyon modellerinde artıkların bağımsızlığı varsayımını ihlal ederek modelin geçerliliğini zayıflatabilir. Bu nedenle otokorelasyonun tespiti ve analizi oldukça önemlidir.

Python'da bir bireyin zaman serisindeki otokorelasyonu analiz etmek için ACF (Autocorrelation Function) grafiği aşağıdaki gibi oluşturulabilir:

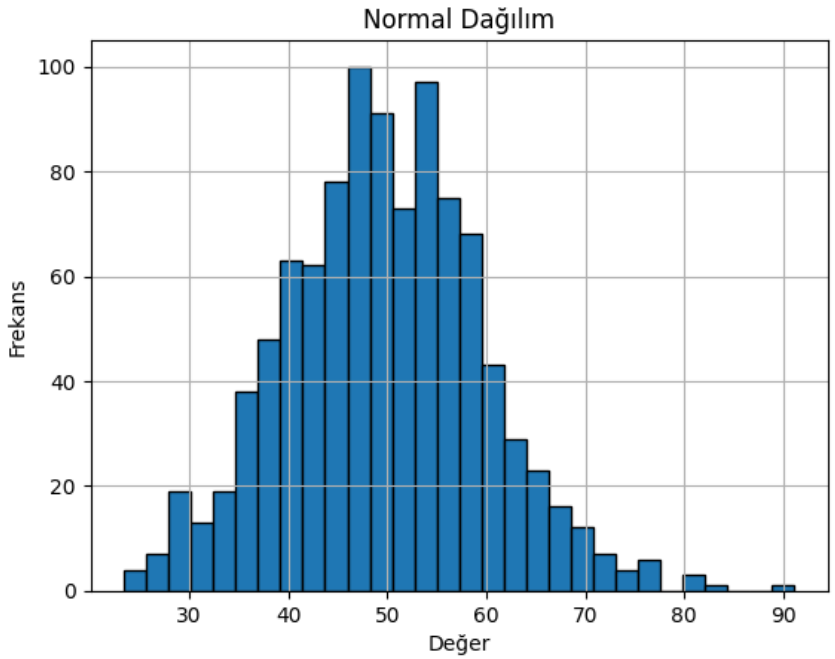


ACF grafiği, belirli gecikmelerde gözlemlenen otokorelasyon değerlerini gösterir. İlk birkaç gecikmede değerler 0'a yakınsa, zaman içi bağımlılığın düşük olduğu anlaşılır. Eğer anlamlı gecikmelerde otokorelasyon varsa, daha karmaşık zaman serisi modelleri (ARIMA, GEE vb.) önerilir [3].

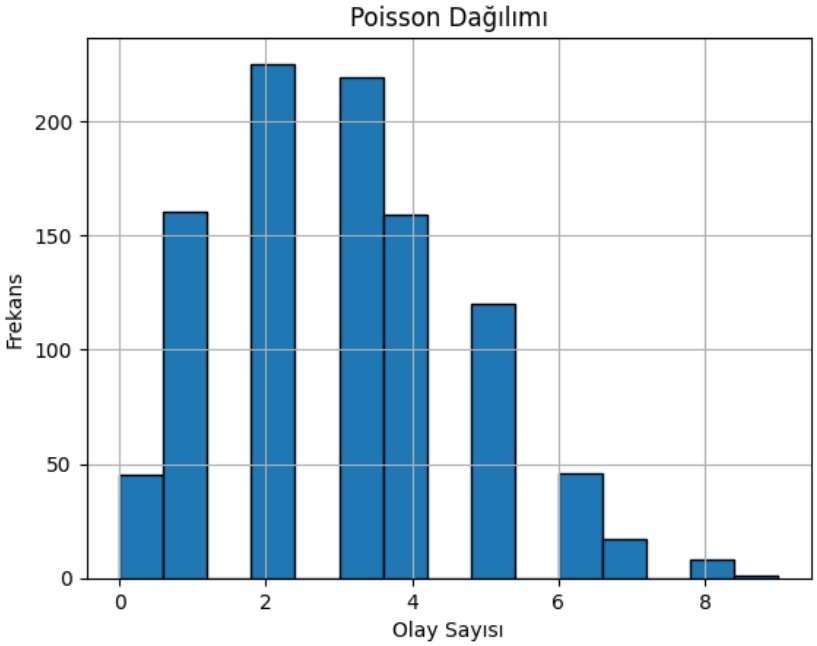
2.4 Kullanılan Dağılımlar

Sentetik veri üretiminde kullanılan dağılımlar, üretilecek verinin türüne ve analitik ihtiyaçlara göre belirlenir. Bu çalışmada üç temel dağılım türü kullanılmıştır: Normal, Poisson ve Gamma. Aşağıda her biri için kısa tanım, örnek üretim kodu ve görsel çıktılarla birlikte açıklama verilmiştir.

Normal Dağılım: Sürekli, simetrik ve çan eğrisi şeklindeki dağılımdır. Ortalama ve standart sapma parametreleri ile tanımlanır.

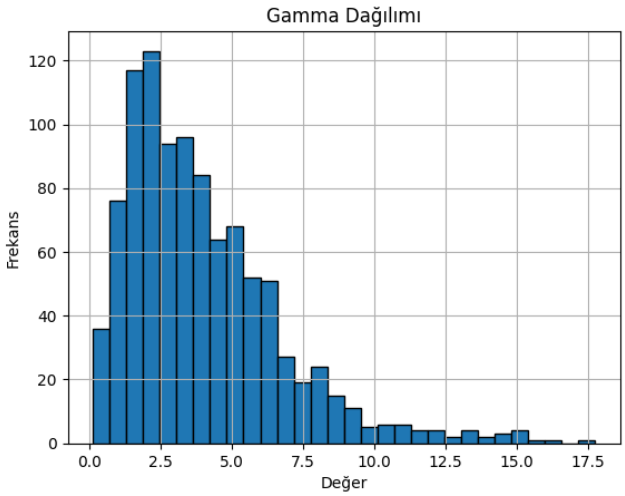


Grafik çan eğrisi şeklindedir, veri ortalama etrafında simetrik dağılır [4].
Poisson Dağılımı: Belirli zaman ya da mekânda nadir olayların sayısını modellemek için kullanılır.



Dağılım sağa çarpıktır ve genellikle düşük değerlerde yoğunlaşır. Sayım verileri için uygundur [4].

Gamma Dağılımı: Pozitif, sürekli ve sağ çarpık veriler için kullanılır.



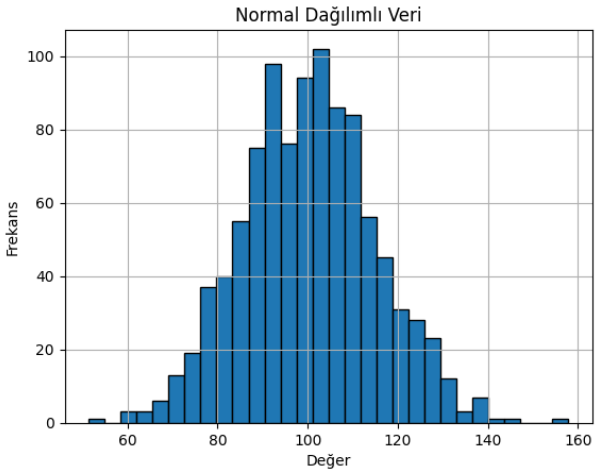
Değerler pozitif ve sağ çarpık yapıdadır. Süre, yoğunluk gibi değişkenler için kullanışlıdır [4]

3. Sentetik Veri Üretimi

3.1 Python ile

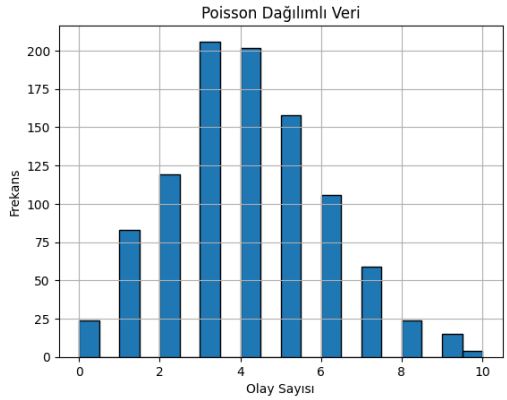
Bu bölümde Python dili ile Normal, Poisson ve Gamma dağılımlı veriler üretilmiş ve univarite yapıdaki örnekler üzerinden değerlendirme yapılmıştır. Her dağılım için örnek kod, grafik ve yorumlara yer verilmiştir.

Normal Dağılımlı Sentetik Veri



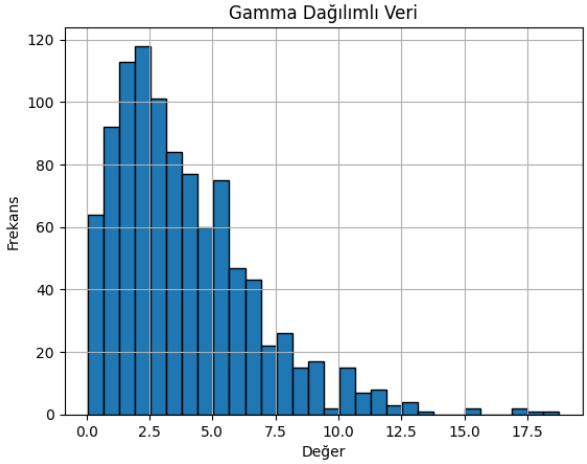
Çan eğrisi şeklindeki simetrik dağılım, doğal olayların modellenmesinde yaygın olarak kullanılmaktadır. Ortalama değerin etrafında yoğunlaşan veri, regresyon ve varyans analizlerinde uygundur [4].

Poisson Dağılımlı Sentetik Veri



Poisson dağılımı özellikle olay sayılarının modellendiği durumlarda (örneğin arıza sayısı, müşteri gelişi) uygundur. Grafikte sağa çarpıklık gözlemlenmektedir [4].

Gamma Dağılımlı Sentetik Veri



Gamma dağılımı pozitif sürekli veriler için uygundur. Özellikle süre, işlem süresi gibi veriler bu dağılımla modellenebilir. Dağılım sağa çarpık yapıdadır [4].

3.2 R ile

rnorm, rpois, rgamma fonksiyonları kullanıldı. expand.grid yardımıyla longitudinal yapı sağlandı [1][2].

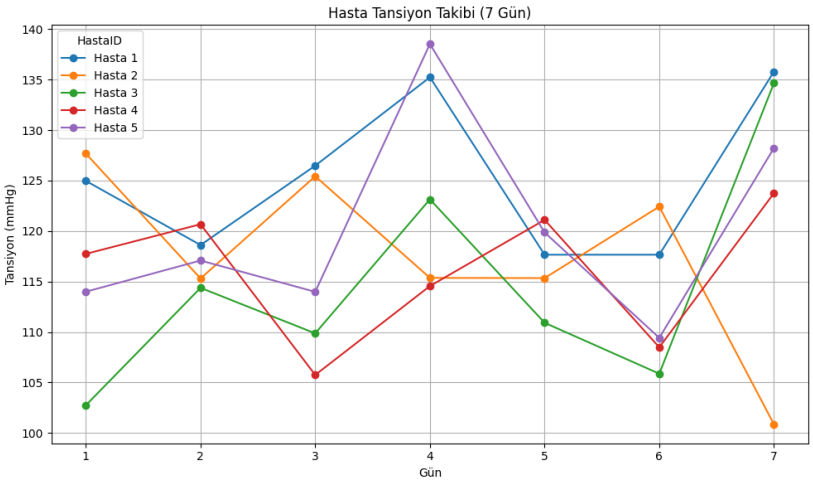
3.3 Senaryo Tabanlı Uygulama: Hasta Takibi Örneği

Bu örnekte, 5 hastanın 7 gün boyunca sabah saatlerinde yapılan tansiyon ölçümleri üzerine kurgulanmış bir sentetik veri yapısı üretilmiştir. Amaç, sağlık alanında longitudinal veri üretimi ve otokorelasyon analizine uygun bir yapı oluşturmaktır. Python kullanılarak her hasta için 7 günlük ölçümler normal dağılım varsayımıyla şu şekilde simüle edilmiştir:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.graphics.tsaplots import plot_acf

np.random.seed(42)
hasta_sayisi = 5
gun_sayisi = 7

data = {
  "HastaID": np.repeat(np.arange(1, hasta_sayisi+1), gun_sayisi),
  "Gun": np.tile(np.arange(1, gun_sayisi+1), hasta_sayisi),
  "Tansiyon": np.random.normal(loc=120, scale=10, size=hasta_sayisi * gun_sayisi)
}

hasta_df = pd.DataFrame(data)
print(hasta_df.head())
```

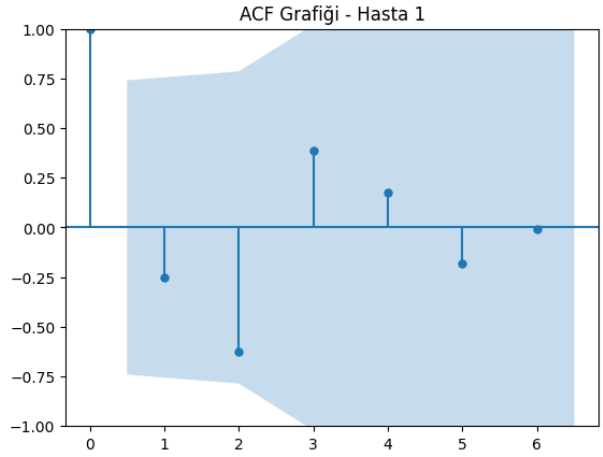
```
   HastaID  Gun    Tansiyon
0        1    1  124.967142
1        1    2  118.617357
2        1    3  126.476885
3        1    4  135.230299
4        1    5  117.658466
```

Bu yapı üzerinden, her hastanın zaman içindeki tansiyon değişimi analiz edilebilir. Verinin normal dağıldığı varsayılarak üretildiği için, ileri analizlerde varyans homojenliği ve otokorelasyon testi uygulanabilir.



Her eğri bir hastayı temsil eder. Tansiyon değerleri doğal varyasyonları gösterecek şekilde üretilmiştir. Bu yapı, longitudinal analizler ve otokorelasyon incelemeleri için uygundur.

ACF Grafiği (Birey 1 için):

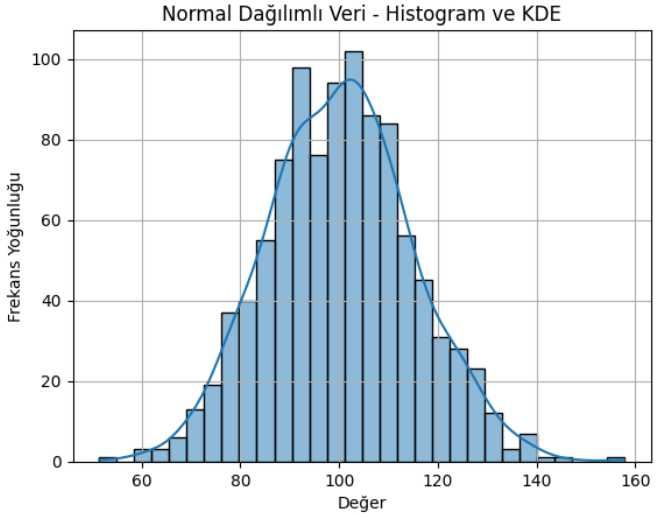


ACF grafiği, ardışık gözlemler arasında korelasyon olup olmadığını gösterir. Eğer otokorelasyon yüksekse, zaman bağımlı modeller (ör. ARIMA, LMM) kullanılabilir. Bu analiz, sentetik verinin gerçekçi davranış üretip üretmediğini anlamak için de kullanılabilir [1][2].

4. Veri Görselleştirme ve Yorumlama

4.1 Univarite Dağılım

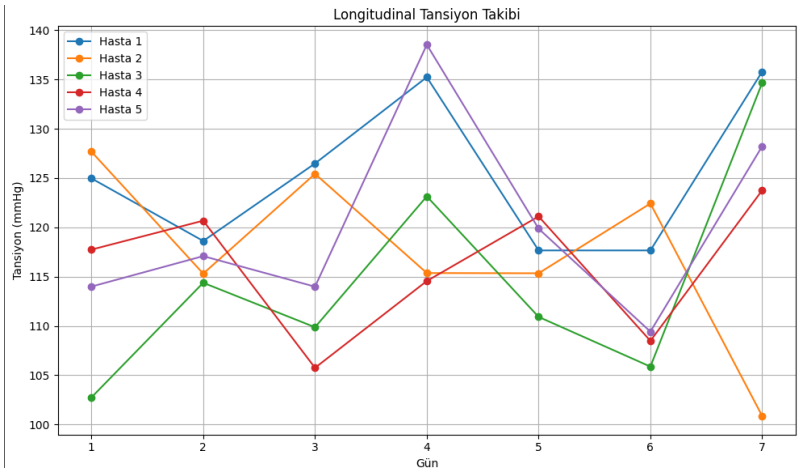
Univarite veri için üretilen Normal, Poisson ve Gamma dağılımlı veriler histogram grafikleri aracılığıyla görselleştirilmiştir. Örneğin:



KDE (Kernel Density Estimation) çizgisi dağılımın şeklini daha net görmeyi sağlar. Normal dağılımda eğri simetriktir; Poisson ve Gamma dağılımlarında çarpıklık gözlemlenir.

4.2 Longitudinal Zaman Serisi

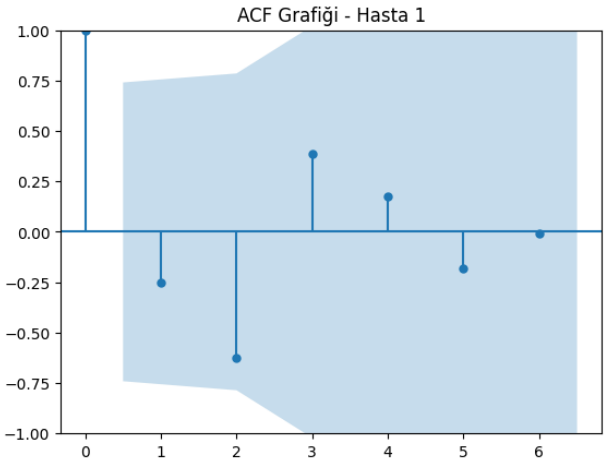
Her birey için zamana bağlı olarak gözlemlenen tansiyon değerleri çizgi grafiğiyle görselleştirilmiştir:



Bu grafik, her bireyin zaman içindeki değişimini ortaya koyar. Trend analizi ve bireyler arası varyasyon gözlenebilir.

4.3 Otokorelasyon Gösterimi ACF (Autocorrelation Function)

grafiği bireysel zaman serisindeki ardışık gözlemler arası ilişkiyi ortaya koyar



İlk gecikmelerdeki yüksek korelasyon, otokorelasyonun varlığını gösterir. Modelleme aşamasında dikkate alınması gereken bir durumdur.

5. Otokorelasyon Analizi ve Regresyon

5.1 Python ile Otokorelasyon Testleri ve Regresyon Uygulaması

Python ortamında regresyon modeli kurularak artıkların otokorelasyon içerip içermediği test edilmiştir. Bunun için Durbin-Watson ve Ljung-Box testleri uygulanmıştır.

```
import statsmodels.api as sm
from statsmodels.stats.stattools import durbin_watson
from statsmodels.stats.diagnostic import acorr_ljungbox

X = sm.add_constant(hasta_df['Gun'])  # Bağımsız değişken (zaman)
y = hasta_df[hasta_df['HastaID'] == 1]['Tansiyon']  # Hasta 1'in verisi
```

```
model = sm.OLS(y, X).fit()
residuals = model.resid

# Durbin-Watson Testi
dw_stat = durbin_watson(residuals)
print(f"Durbin-Watson istatistiği: {dw_stat:.3f}")

# Ljung-Box Testi
lb_test = acorr_ljungbox(residuals, lags=[3], return_df=True)
print(lb_test)
```

Durbin-Watson değeri 2'ye ne kadar yakınsa, artıklar arasında otokorelasyon o kadar azdır. 2'den küçük değerler pozitif otokorelasyonu, büyük değerler negatif otokorelasyonu gösterir. Ljung-Box testi ise artıkların genel anlamda bağımsız olup olmadığını test eder. $p<0.05$ ise otokorelasyon vardır [5].

5.2 R ile Otokorelasyon Testleri ve Regresyon Uygulaması

R dilinde, lm() fonksiyonu ile regresyon modeli kurulmuş ve dwtest() ile Box.test() kullanılarak otokorelasyon testi yapılmıştır.

```
# Veriyi simüle edelim
gun <- 1:7
tansiyon <- c(120, 122, 121, 125, 123, 124, 122)
model <- lm(tansiyon ~ gun)

# Durbin-Watson testi
library(lmtest)
dwtest(model)

# Ljung-Box testi
library(stats)
Box.test(residuals(model), lag = 3, type = "Ljung-Box")
```

Yorum: dwtest() ile elde edilen p-değeri anlamlıysa otokorelasyon söz konusudur. Box.test() ise artıkların zaman bağımsızlığı varsayımını değerlendirir. Eğer test sonucunda $p<0.05$ çıkarsa, artıklar arasında otokorelasyon vardır ve modelde bu yapı dikkate alınmalıdır [6].

6. Sonuç ve Değerlendirme

Bu çalışma, sentetik veri üretimi ile otokorelasyon analizlerinin bütünleşik olarak ele alınmasını hedeflemiştir. Univarite ve longitudinal yapıdaki verilerin Python ve R programlama dilleri ile simüle edilmesi, dağılım özelliklerinin farklı veri türlerine göre nasıl değiştiğini ortaya koymuştur. Normal, Poisson ve Gamma dağılımlarına göre

oluşturulan sentetik veriler, hem sayım hem de sürekli değişkenlerin modellenmesine olanak sağlamış ve veri üretiminde esneklik sunmuştur.
Longitudinal veri örneği üzerinden yapılan zaman serisi analizi, birey içi tekrar eden ölçümlerle otokorelasyonun doğal olarak oluştuğunu göstermiştir. ACF grafikleri ve Durbin-Watson testi gibi araçlarla yapılan değerlendirmeler, klasik regresyon varsayımlarının bu tür verilere doğrudan uygulanamayabileceğini ortaya koymuştur. Özellikle sağlık gibi mahremiyet gerektiren alanlarda sentetik veri kullanımı, etik ve pratik açıdan güçlü bir alternatif sunmaktadır.
Ayrıca, bu çalışmada Python ve R ortamlarının sentetik veri üretimi, görselleştirme ve analiz süreçlerindeki işlevselliği açıkça görülmüştür. Kod temelli üretimin şeffaflık ve tekrarlanabilirlik sağlaması, bilimsel araştırmalar açısından önemli bir avantajdır.
Sonuç olarak, sentetik veri üretimi yalnızca veri gizliliğini korumakla kalmamakta; aynı zamanda model test etme, yöntem karşılaştırma ve yapay veri ile gerçek veriye yaklaşma gibi akademik katkılar sunmaktadır. Gelecek çalışmalarda, daha karmaşık veri yapıları ve yapay zekâ destekli üretim teknikleri ile bu analizler genişletilebilir.

7. Kaynakça

1. Goncalves, A. et al. (2020). Generation and Evaluation of Synthetic Patient Data. *npj Digital Medicine*. https://doi.org/10.1038/s41746-020-0273-4
2. Diggle, P. et al. (2002). *Analysis of Longitudinal Data*. Oxford University Press.
3. Box, G.E.P., Jenkins, G.M. (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day.
4. Hilbe, J. M. (2011). *Negative Binomial Regression*. Cambridge University Press.
5. Statsmodels Documentation – Python: https://www.statsmodels.org/
6. R Documentation – lmtest, Box.test: https://cran.r-project.org/web/packages/lmtest/index.html
7. Raghunathan, T. E., Reiter, J. P., & Rubin, D. B. (2003). Multiple Imputation for Statistical Disclosure Limitation. *Journal of Official Statistics*, **19**(1), 1–16. https://www.stat.fi/static/media/uploads/jos/2003/2003_1_raghunathan_etal.pdf
8. Snoke, J., Raab, G. M., Nowok, B., Dibben, C., & Slavkovic, A. (2018). General and Specific Utility Measures for Synthetic Data. *Journal of the Royal Statistical Society: Series A*, **181**(3), 663–688. https://doi.org/10.1111/rssa.12358
9. Drechsler, J. (2011). *Synthetic Datasets for Statistical Disclosure Control: Theory and Implementation*. Springer. https://doi.org/10.1007/978-1-4419-7488-4
10. Akande, O., Li, F., & Reiter, J. P. (2021). Synthetic Data Generation for Longitudinal Data. *Statistical Modelling*, **21**(2), 160–178. https://doi.org/10.1177/1471082X19893662

11. Nowok, B., Raab, G. M., & Dibben, C. (2016). synthpop: Bespoke Creation of Synthetic Data in R. *Journal of Statistical Software*, **74**(11), 1–26. https://doi.org/10.18637/jss.v074.i11
12. Williams, A. R., & Overstall, A. M. (2020). Bayesian Synthesis of Longitudinal Data. *Bayesian Analysis*, **15**(3), 879–904. https://doi.org/10.1214/19-BA1176
13. El Emam, K., Mosquera, L., Bass, J., & Buckeridge, D. (2020). Evaluating Identity Disclosure Risk in Fully Synthetic Health Data: Model Development and Validation. *Journal of Medical Internet Research*, **22**(11), e23139. https://doi.org/10.2196/23139
14. Hu, J., Ng, K., & Hastie, T. (2020). A Generative Model for Synthetic Longitudinal Data. *Proceedings of the National Academy of Sciences*, **117**(48), 30088–30094. https://doi.org/10.1073/pnas.2001984117
15. Day, F. R., Loh, P. R., Scott, R. A., Ong, K. K., & Perry, J. R. B. (2016). A Robust Example of Collapsibility Bias in a Genetic Association Study. *The American Journal of Human Genetics*, **98**(2), 392–393. https://doi.org/10.1016/j.ajhg.2015.12.009
16. Nowok, B., Raab, G. M., & Dibben, C. (2019). *synthpop: Generating synthetic versions of sensitive microdata for statistical disclosure control*. *The R Journal*, **11**(2), 26–37. https://journal.r-project.org/articles/RJ-2019-022/
17. Yüce, M. (2009). *Zaman serisi analizinde ARIMA modelleri ve bir uygulama* (Yüksek Lisans Tezi, Anadolu Üniversitesi). Anadolu Üniversitesi Açık Erişim Sistemi. https://earsiv.anadolu.edu.tr/xmlui/bitstream/handle/11421/5566/470482.pdf
18. Drechsler, J. (2011). *Synthetic longitudinal data for public use*. In *Synthetic Datasets for Statistical Disclosure Control* (pp. 159–180). Springer. https://link.springer.com/chapter/10.1007/978-1-4419-8342-8_7
19. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. *arXiv preprint*. https://arxiv.org/pdf/1305.1656
20. Akar, G. (2023). *Veri çoğaltma tekniklerinden SMOTE algoritması ile dengesiz verilerin sınıflandırılması*. *Uluslararası İktisadi ve İdari İncelemeler Dergisi*, **(37)**, 453–472. https://dergipark.org.tr/tr/download/article-file/2560764