Bu çok değişkenli zaman serisi kümeleme problemine üç farklı şekilde yaklaşabiliriz.

1. Her bir zaman noktasını ayrı bir değişken olarak alıp normal kümeleme algoritmalarını uygulamak: Bu durumda her bir gün için 3 değişken* 24 zaman noktası, yani toplamda 72 değişkenimiz olacak. Bu uygulamada zaman noktaları arasındaki bağı kopardığımız için aradığımız zamanla ilişkili patternları keşfetmemiz, bunu dikkate alan algoritmalara nazaran daha zor. O yüzden bu yaklaşımı kullanmadım.

2. Çok değişkenli zaman serisi algoritmalarını kullanmak: Bu algoritmalar diğer normal kümeleme algoritmalarına ve tek değişkenli zaman serisi algoritmalarına nazaran daha ender bulunuyor. "dtwclust" paketinde yer alan partitional ve hiererchical kümeleme algoritmalarını inceledim. Bu uygulamalar zaman serisi ilişkili uzaklıklar kullanarak daha anlamlı kümelemeler yapmamıza olanak sağlıyorlar. İncelediğim bir başka metod ise "Permutation Distribution Kümelemesi"ydi. Bu kümelemede doğrulama yöntemi ancak etiketlenmiş training verileriyle mümkün olduğundan bu metodu ilerleyen aşamalarda kullanmadım.

3. Herbir zaman serisinden bir takım zaman serisi ilişkili değişkenler üretip bu değişkenleri kullanarak normal kümeleme algoritmalarını kullanmak: "Tsfeatures" R-paketi, zaman serilerini alıp entropi, stability, crossing points, autocorrelation function vb. özellikleri çıkarıyor. Zaman serilerinin özetini sağlayan bu değişkenleri ve normal kümeleme algoritmalarını kullanarak patternleri bulabiliriz.

ÇOK DEĞİŞKENLİ ZAMAN SERİSİ KÜMELEMESİ

ÖN İŞLEMLER

Veriyi R a yükledikten sonra öncelikle ptf ve smf değişkenlerini nümerik veriye çevirdim. Nisan ayındaki herbir gün saat başı alınan 24 veri noktasından oluşan 3 zaman serisiyle temsil ediliyor: smf, ptf ve netYön. Ptf (15.4, 567) ve smf (1.99, 567) değişkenlerinin açıklıkları birbirine yakınken netYön (-2346.2 , 3880.3) çok daha açık. Ortalamalarıysa birbirlerine yakın: 312.1, 311.43 ve 305.4 . BU değişkenlere scaling uygulayarak aynı açıklığa sahip olmalarını sağladım. Analizlerimiz uzaklık temelli metotlar içerdiğinden bu aşama önemli. Ardından veriyi, dtwclust paketinin gerektirdiği şekile yani 30 adet zaman serisinden ("xts") oluşan bir listeye dönüştürdüm. Zaman serilerimiz kayıp değer içermiyor, o yüzden herhangi bir imputation yapmamıza gerek kalmadı.

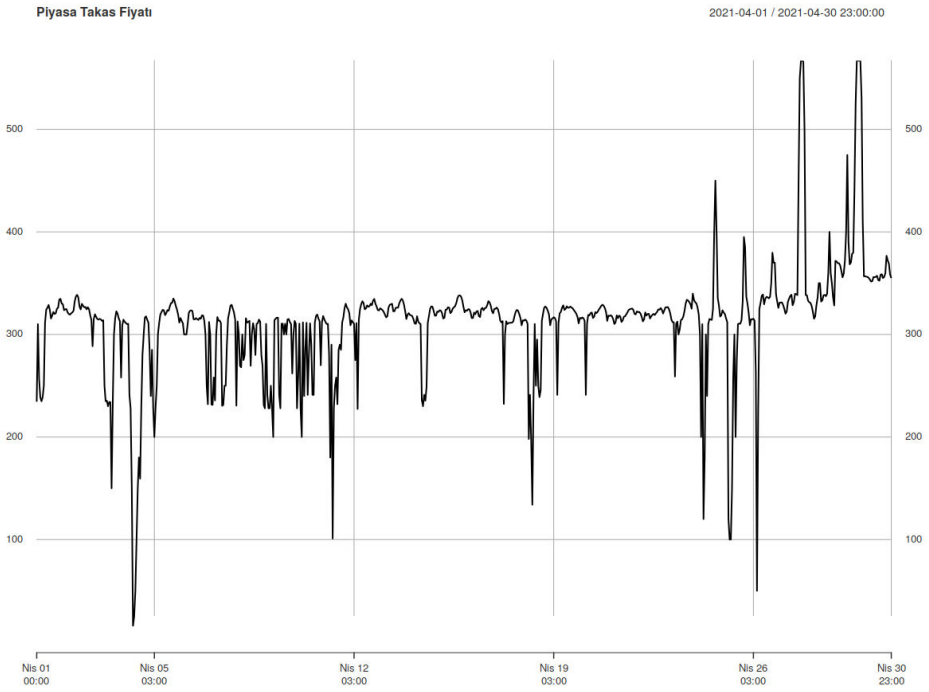
PATTERNLAR VE GÖRSELLEŞTİRME

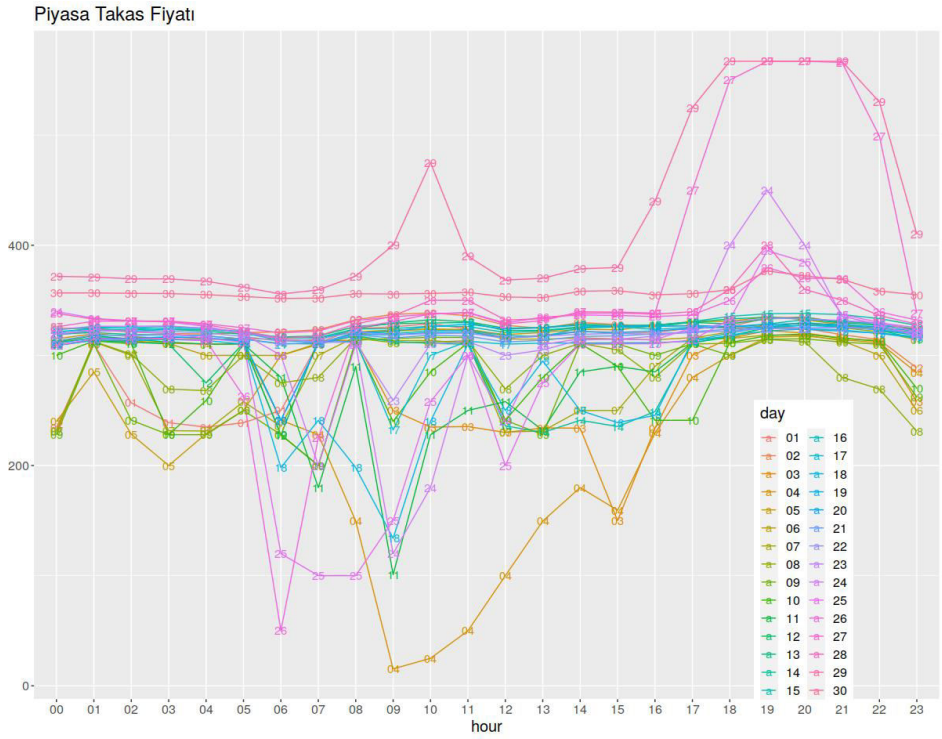
Kümeleme yaptıktan sonra patternları daha iyi bir şekilde yakalamayı bekliyoruz, ancak çıplak gözle görseller üzerinde bir öndeğerlendirme yapabiliriz.

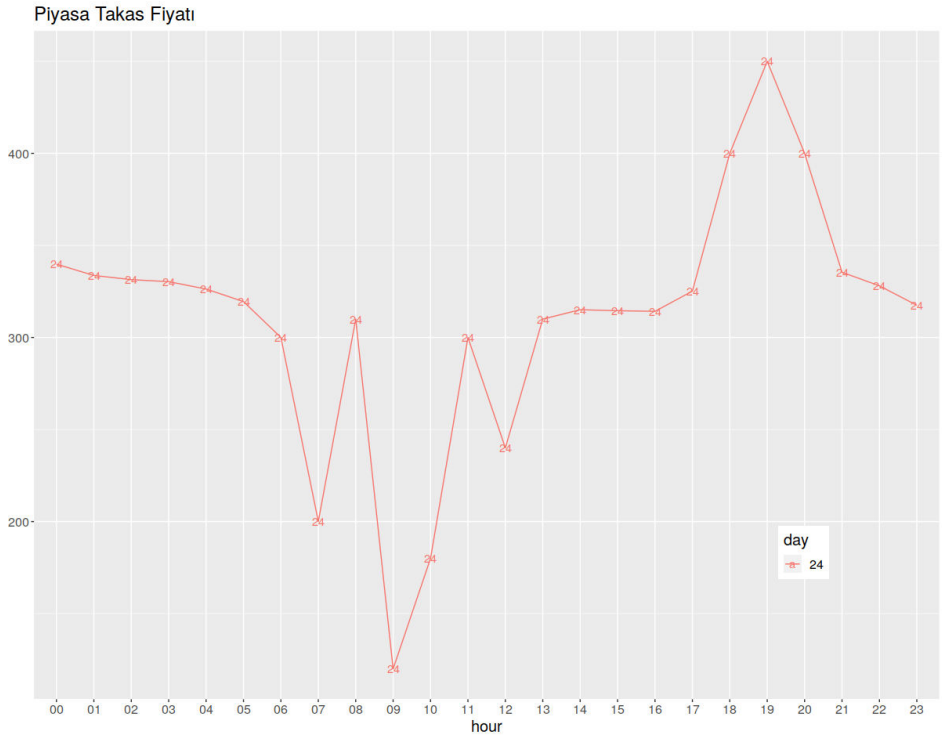
PTF DEĞİŞKENİ

Gördüğümüz üzere piyasa takas fiyatı genelde 300 civarında seyrederken özellikle ayın son haftasında bazı günlerde peak yapmış, bazı günlerde ise yüksek düşüşler yaşanmış. Seasonplot'a bakarsak gün gün bu düşüş ve yükselişlerin hangi saatler civarında olduğuna da bakabiliriz. Aynı zamanda daha yakından bakmak için tek tek günleri ayrıştırarak da görselleştirebiliriz. Mesela 29. günde hem saat 10 civarı hem de akşam saatlerinde güçlü bir

yükseliş görüyoruz. 27. günde ise bir defa saat 4 ten sonra yükselip gece yarısına doğru tekrar düştüğünü görüyoruz. 24. gün ise çok daha farklı bir patterns sahip, 7, 9 ve 12 saatlerinde düşüp 19 civarında ortalamanın üstüne çıkıp peak yapıyor. 4. günde ise 5 ve 17 saatleri arasında azalarak düşüp sonra artışa geçerek tekrar ortalamaya dönüyor.

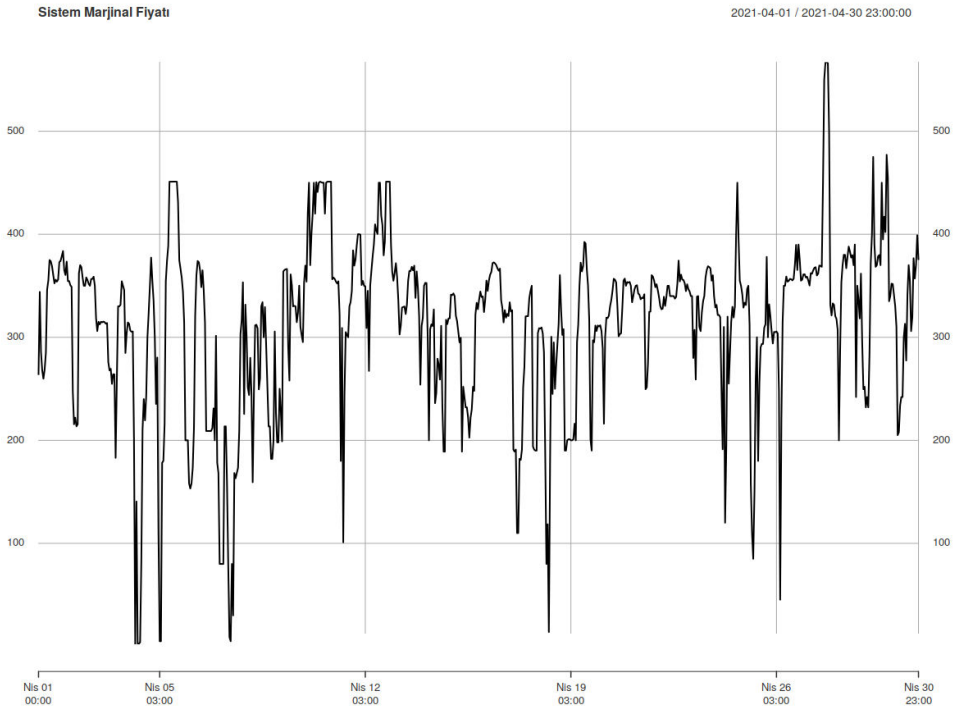


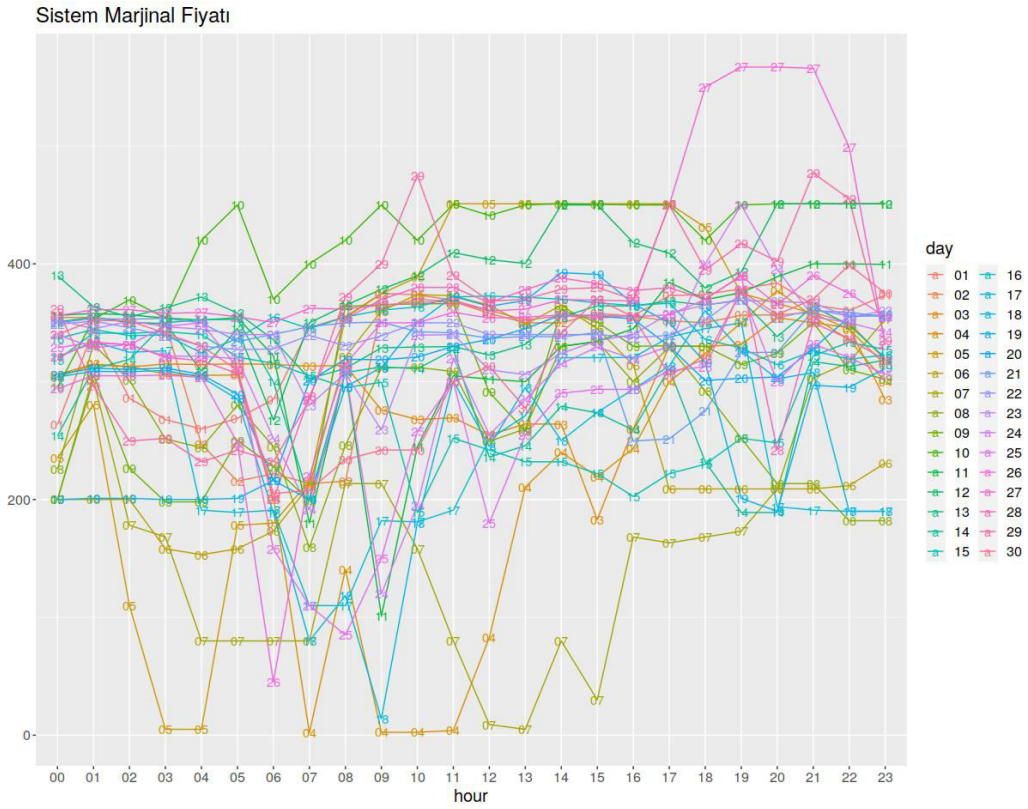




SMF DEĞİŞKENİ

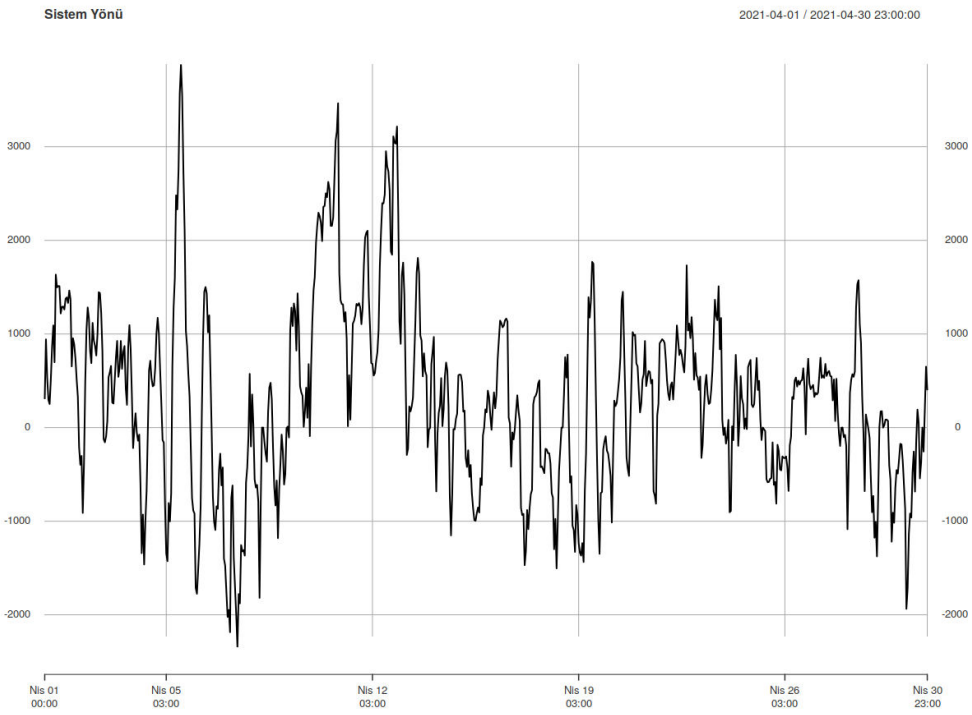
Görüldüğü üzere smf değişkenin ptf değişkenine göre varyansı daha yüksek, ptf kadar stabil değil. Gerçek zaman kaosunun sisteme anlık etkileri düşünüldüğünde bu beklenen bir durum.

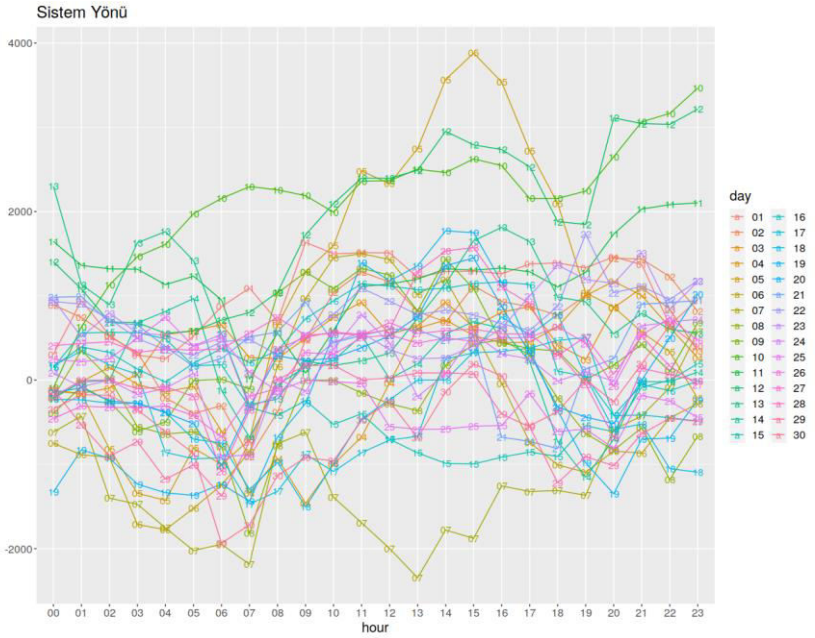


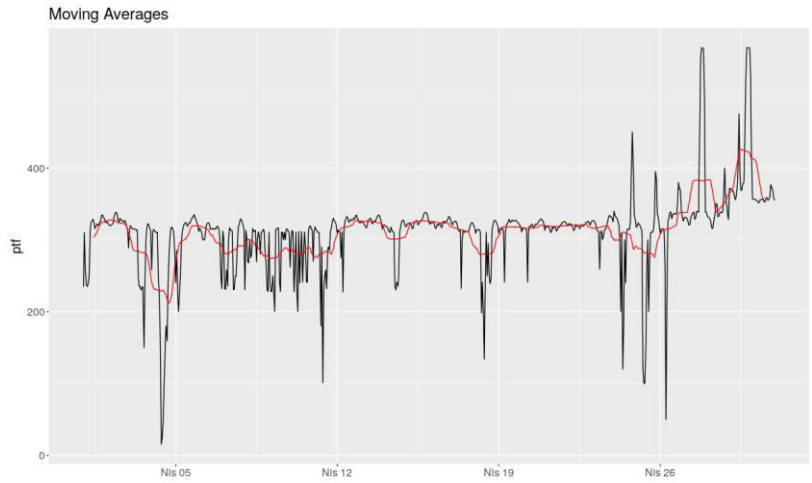


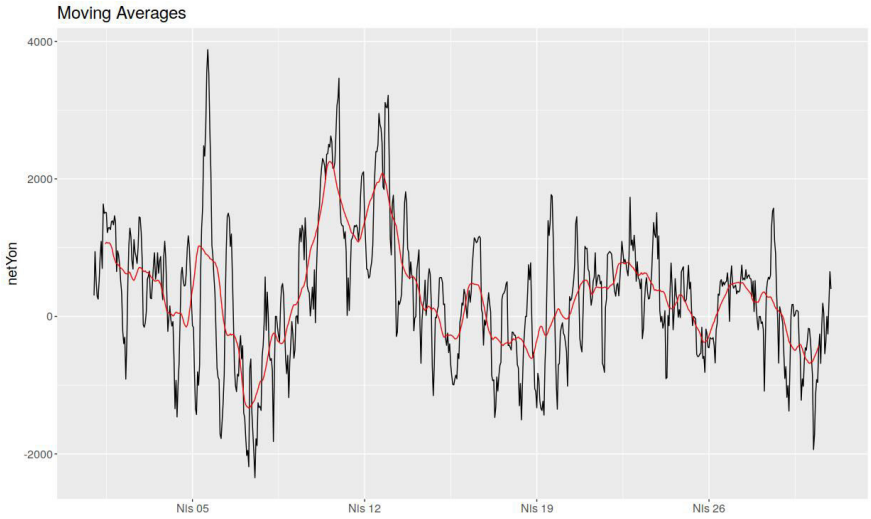
NETYON DEĞİŞKENİ

Gördüğümüz üzere, netyön değişkeninin açıklığı diğerlerine göre daha geniş. 24 saatlik moving average ına baktığımızda diğerlerine nazaran daha değişken olduğunu görüyoruz. Smf değişkeninde iniş çıkışlar olsa da günlük ortalama değer nisbten sabit kalsa da netyön değişkeni için bu geçerli değil. Varyansı da diğerlerine nazaran daha fazla gözüküyor.


Sistem Yönü
2021-04-01 / 2021-04-30 23:00:00
3000
2000
1000
0
-1000
-2000
Nis 01
00:00
Nis 05
03:00
Nis 12
03:00
Nis 19
03:00
Nis 26
03:00
Nis 30
23:00


Sistem Yönü
4000
2000
0
-2000
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23
hour
day
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
16 17 18 19 20 21 22 23 24 25 26 27 28 29 30





ALGORİTMA SEÇİMİ

Kümeleme algoritmalarında çok değişkenli zaman serileri için partitional ve hierarchical kümeleme yöntemlerini inceledim. Burada kullanılabilecek 3 uzaklık vardı: DTW(dynamic time warping distance), SOFTDTW(another version of DTW) ve GAK (Fast global alignment kernels). DTW zaman serileri arasındaki uzaklıkları incelerken eğme ve yamultma uyguluyor, bu nedenle euclidean uzaklıklara göre daha esnet ve time shift ve scaling etkilerini azaltma konusunda daha iyi. SOFTDTW'de benzer olmasına karşın bir uzaklık fonksiyonundan beklediğimiz negatif olmama, üçgen eşitsizliğini sağlama vb. bazı önemli özellikleri sağlamıyor, bu yüzden tercih etmedim. GAK ise DTW'nin bir çeşit genelleştirmesi gibi çalışıyor ve bazı benzerlikleri yakalamada DTW den daha iyi olduğu iddia ediliyor. Bu yüzden uzaklık olarak **DTW ve GAK**'ı seçtim.

Uzaklığı seçtikten sonra centroid seçimi yapmamız gerek. Centroid olarak ortalamayı kullanmak algoritmaları aykırı değerlere karşı korumasız hale getiriyor. Medyan daha iyi bir seçenek olmasına rağmen burada da zaman serisinin değerlerinden biri olmadığı için yeterince uygun bir seçenek değil. Centroidin zaman serisininin bir elemanı olmasını tercih ettiğim için **PAM**(partition around medoid) kullanmayı tercih ettim. Aynı zamanda DTW ile

uyum içinde çalışan **DBA**(dtw Barycenter averaging) centroidini de kullandım. Ancak bu centroid sadece DTW ile ve partitional kullanımına uygun.
Sonuç olarak denediğim modeller şunlar:

method	**uzaklık**	**centroid**
partitional	dtw	pam
partitional	dtw	dba
partitional	gak	pam
hierarchical	dtw	pam
hierarchical	gak	pam

Farklı uzaklıklar ve centroidler veriye farklı açılardan bakmamızı sağladığı için bir uzlaşma yöntemi tercih ederek yukarıdaki modellerin hepsini inceledim. Elimizde etiketlenmiş bir veri olmadığı için bu modelleri birbiriyle karşılaştırıp hangisinin bizim durumumuza daha uygun olduğuna karar vermek mümkün değil. Ancak internal kümeleme validasyon değerlerini kullanarak optimum modelleri araştırabiliriz.

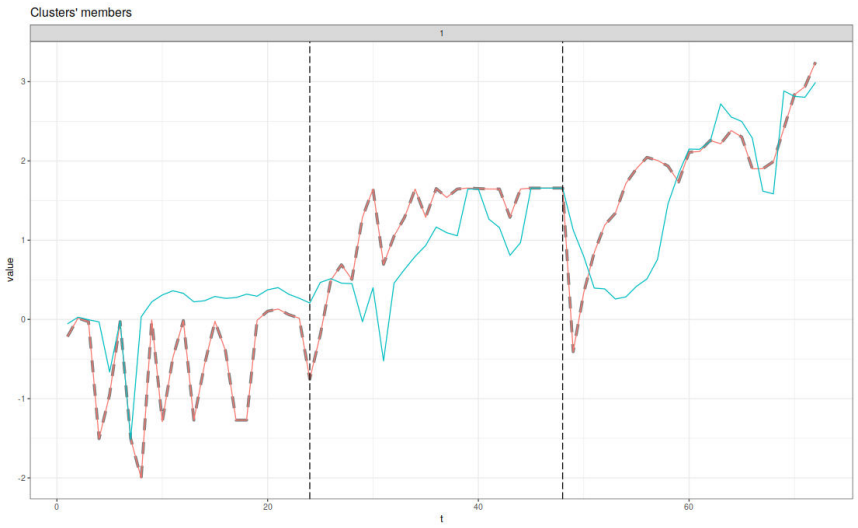
OPTİMUM MODELİN BELİRLENMESİ

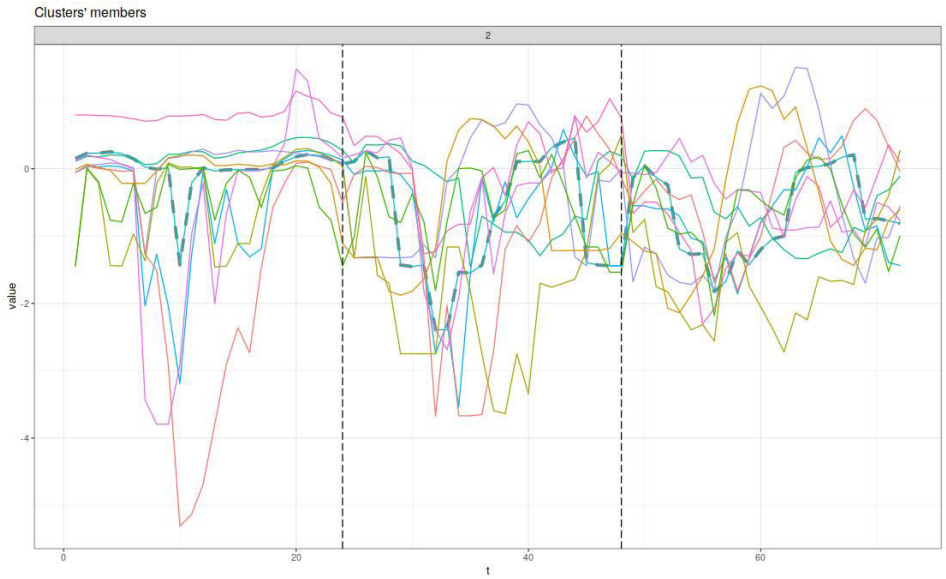
Seçtiğimiz herhangi bir kümeleme algoritması için optimum küme sayısını belirlemek için kümeleme validasyon değerlerini kullanabiliriz. Elimizdeki veri etiketlenmiş olmadığından bunu external değerler değil sadece internal değerlerle yapabiliriz. Burada tek bir değeri kullanmak yerine farklı değerlere baklıp, çoğunluğa uymak ya da veriyle ilgili bir öngörümüz ayrışma için kullanabileceğimiz bir bilgi varsa buna uygun bir validasyon seçeneği seçebiliriz. Aşağıdaki seçenkleri inceledim. Silhouette index (maximize edilecek) , Dunn index(maximize edilecek) , COP index (minimize edilecek), Davies- Bouldi index(minimize edilecek), Modified Davies- Bouldin index(minimize edilecek), Calinski- Harabasz index(maximize edilecek) ve Score Function(maximize edilecek). Önceden kaç kümeye ayırılması gerektiğini bilmiyoruz. Bu yüzden 2 ve 6 arasındaki değerleri inceleyeceğim. Örnek olarak partitional, GAK, PAM durumuna bakalım. Bu durumda Silhoutte, Dunn, DB ve DBSTAR 3 kümeyi tavsiye ederken, CH ve sf 2, COP ise 5 i tavsiye ediyor. Bu durumda 3 kümeyi tercih edebiliriz. Ancak bu kesin bir seçim değil, eğer önceden kümelerin başka bir sayıda olması gerektiğini düşünmemize sebep olacak bir bilgiye sahipsek bu bilgiyi de kullanabiliriz. Özellikle validasyon değerleri birbirine yakın olduğunda bu kümelerin benzer miktarlarda ayrık ve kompact olduğunu gösterir, ayrım bulanıklaşır.

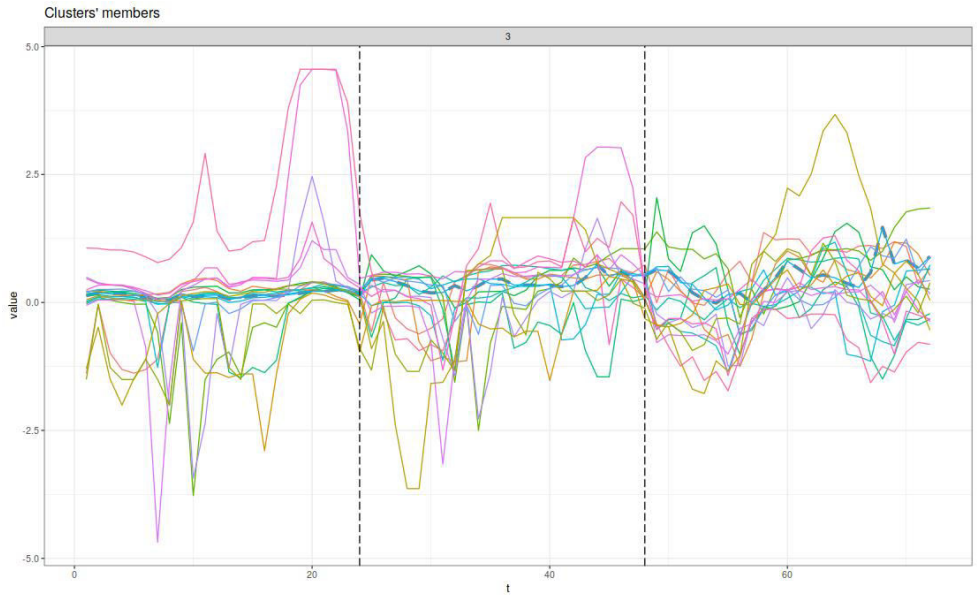
SONUÇLAR

Burada öncelikle partitional, GAK ve PAM kullanarak yapılmış, optimum 3 küme sayısıyla oluşturulmuş bir kümelemeyi görselleştirdim. İlk kümede 10 ve 12. günler, ikinci kümede 4,6,7,8,15,17,18,19,25,30. günler ve üçüncü lümede de diğer günler var. Parçalı çizgiyle ifade edilen kısım küme centroidi. Centroidlere bakarak üçüncü kümenin ortalaması üç değişkende de pek değişmeyen günlerden oluştuğunu söyleyebiliriz. İlk grup iniş ve çıkışlar bakımından

birbirine yakınlar, smf değerleri yükselerek ortalamanın üzerine çıkıyor. Ptf değerleri genelde ortalamanın altında , smf değerleri de ortalamanın üstünde. Netyön değerleri ise ortalamanın üstünde ve akşam saatleri civarı ufak bir düşüş hariç yükselişte. İkinci grupta ise ptf değerleri öğleden biraz önce bir düşüş yaşıyor, ancak genel olarak ortalama civarında. Smf değerleri önce düşüp sonra yükseliyor, ancak hep ortalamanın altında kalıyor, ve netyön değerleri de aynı şekilde.

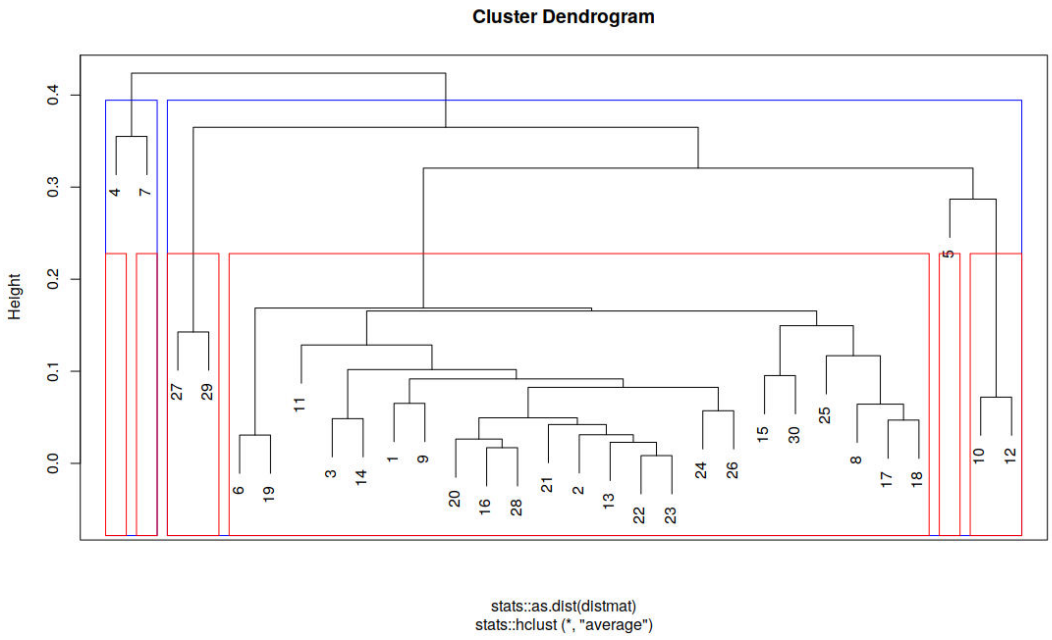


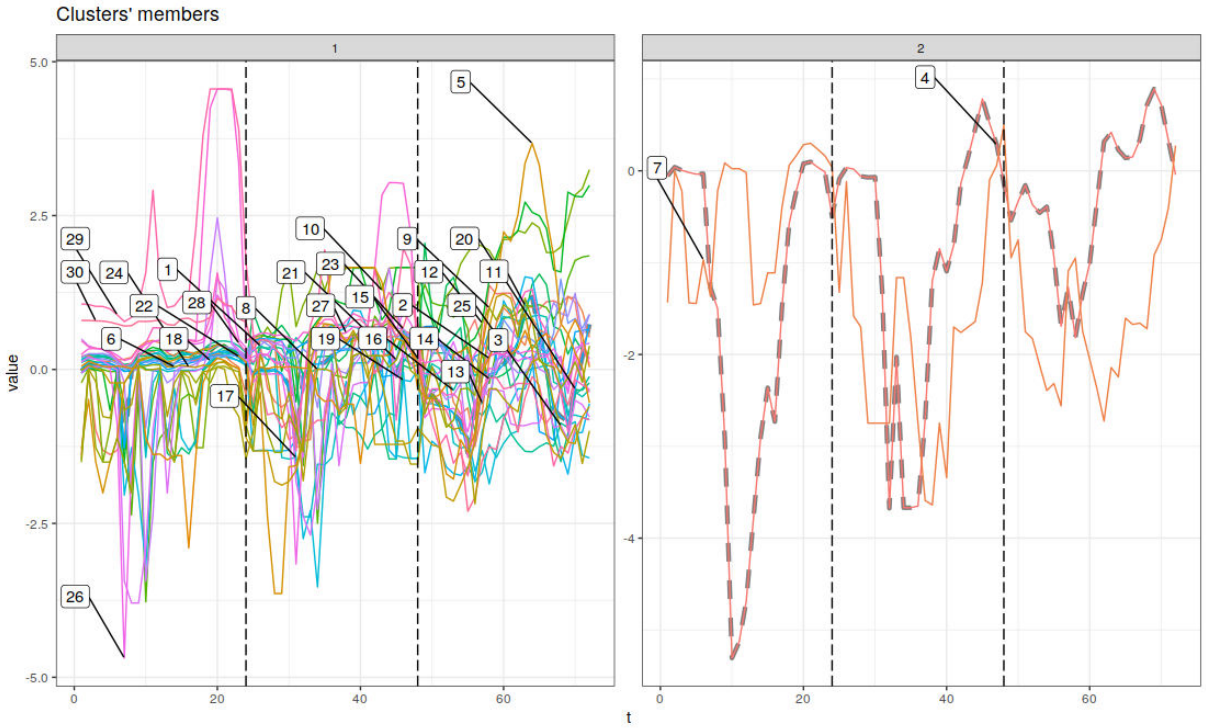




Hierarchical bir yöntemi de göstermek amacıyla bir başka modelin daha görselleştirmesini yapacağım. Bu modelde GAK uzaklığı ve PAM centroidi kullandım. Consensusla tavsiye edilen küme sayısı 2 ya da 6'ydı. Mavi dikdörtgenler 2 kırmızı 6 kümelemeyi gösteriyor. Burada 10 ve 12 değil, 4 ve 7 diğer noktalardan ayrışıyor. Ancak küme sayısını 6ya çıkardığımızda 10 ve 12'nin tekrar beraber başka bir küme oluşturduğunu görüyoruz. Bu farklı sonuç bu algoritmanın iyi çalışmadığını gösterebileceği gibi, farklı bir açıdan bakıldığında 4 ve 7nin de benzerlikleri bakımından bir küme oluşturduğunu göstermiş olabilir. Daha derinlemesine bir inceleme gerekir. Zaman serisi görseline baktığımızda 4 ve 7nin birtakım benzerlikleri olduğunu görebiliyoruz. Ama yine de bu benzerliğin beraber olarak ayrı bir kümeye koyacak kadar fazla oldupunu düşünmüyorum. Mesela ptf değeri ikisinde de ortalamanın altında ama birinde 2 nisbeten küçük düşüş varken birinde tek ve derin bir düşüş var.

Son olarak, diğer yukarıda saydığım farklı modelleri uygulayarak vardığım consensus sonuçlarından bahsetmek isterim. Modellerin çoğu 10 ve 12'yi aynı kümeye koydular, benze biçimde 29 ve 30 da beraber kümelenmişlerdi. Diğer bir grup ise 6,7,8,15,17 19 ve 25 bu günler de neredeyse bütün kümeleme metotlarında beraber kümelendiler. Aynı şekilde 2,3,9,13,16,21,22,23,24,26 ve 27 de. Dolayısıyla bu kümelerin doğruluğu hakkında diğerlerine nazaran daha net olabiliriz. Yeterince zamanım olmadığından ekleyemiyorum ama bu kümeler de görselleştirilip benzerlikleri incelenebilir.

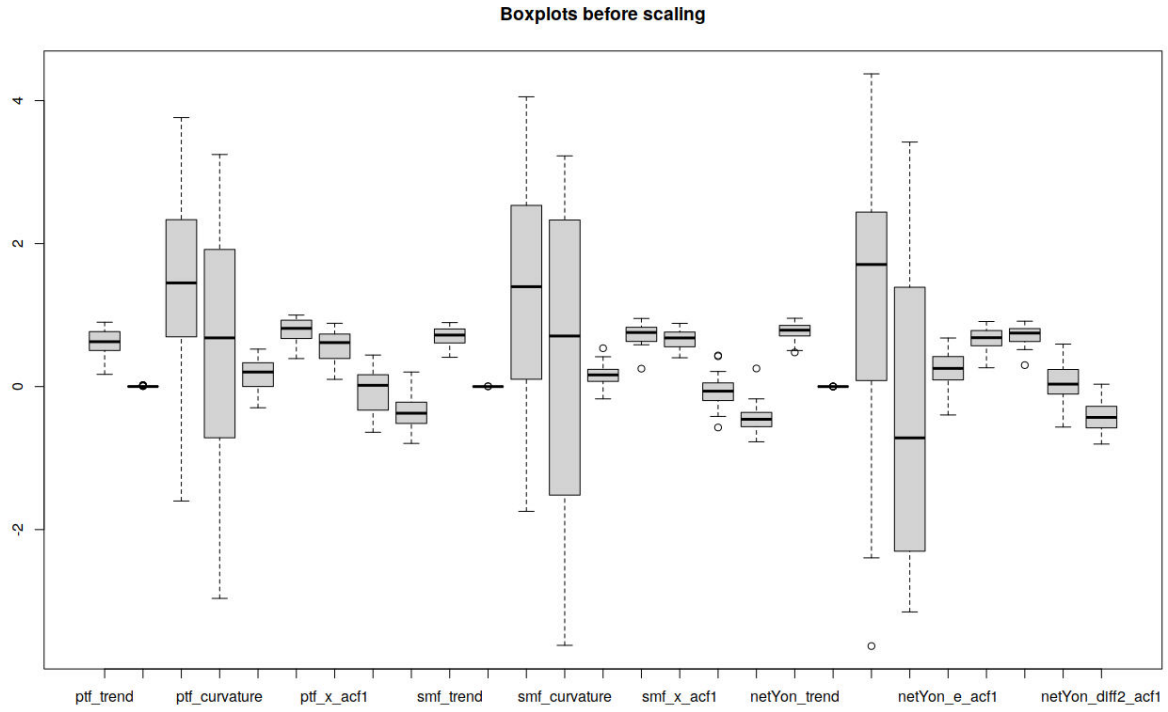


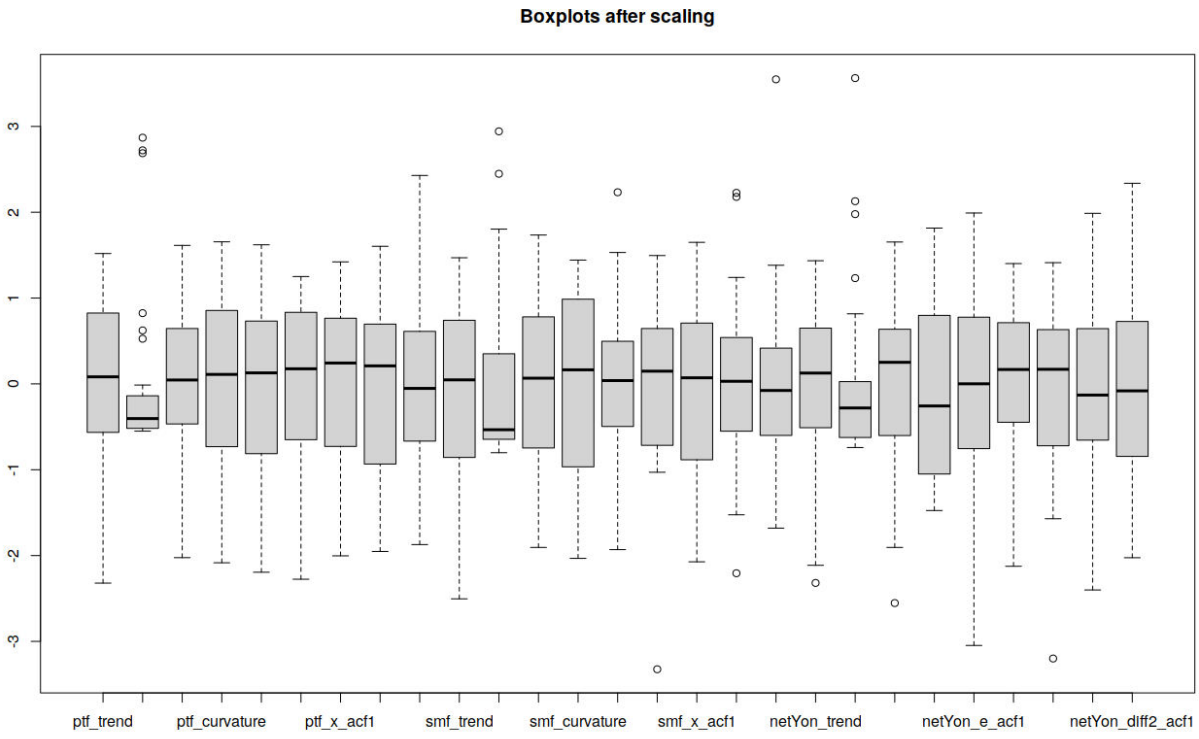


ZAMAN SERİLERİNDEN ELDE EDİLEN DEĞİŞKENLERLE KÜMELEME

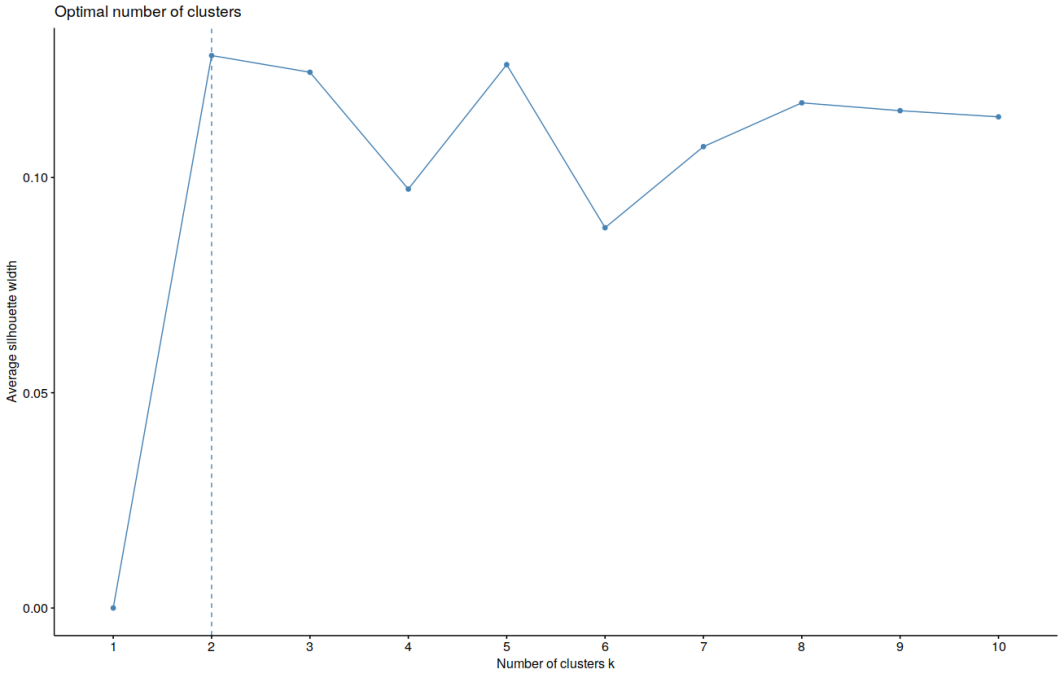
"tsfeatures" R paketini kullanarak ptf, smf ve netyön değişkenlerinden zaman serisi ilişkili değişkenler elde ettim. BU değişkenlerden bazıları bizim durumumuzda kullanılabilir değil. Örneğin günlük değerlere baktığımızdan mevsimsellik değişkenini kullanamıyoruz. Bu şekilde bazı değişkenleri eledikten sonra entropy, trend, curvature gibi 9 değişken elde ettim. Böylece herbir gün için elimizde 27 değişken oldu. Aşağıdaki boxplotta görebileceğiniz üzere bu değişkenlerin ortalama ve açıklıkları birbirinden oldukça farklı. Uzaklık bazlı metotlar kullandığımızdan bu verisetini scale ettim. Bu değişkenlerin bazıları oldukça correlated ancak bu durum kullanacağım model için bir sorun teşkil etmiyor. Ancak boxplotlarda

farkedebileceğiniz üzere bazı aykırı noktalar mevcut, bunun problem oluşturmaması için kmeans algoritması yerine PAM kullanmayı tercih ettim. Uzaklık olarak ise “euclidean” uzaklık yerine aykırı noktaları daha iyi tolere edebilen “manhattan” uzaklığını tercih ettim. Model-based, dağılımları kullanan DBSCAN metodu da kullanılabilirdi. Ancak veri sayımız sadece 30 ve DBSCANin sağlıklı çalışabilmesi için yeterli değil. Ayrıca GMM modelini kullanabilmek için değişkenlerin normal dağılımdan gelmesi gerekiyor, bizim değişkenlerimizin bazılarının densitysi bunu sağlarken bazıları sağlamıyor. PCA kullanıp değişken uzayını küçülttüğümde de yine bu componentların bazılarının normal dağılıma benzemediğini gördüm. Dolayısıyla ilerleyen aşamalarda yalnızca manhattan uzaklığı ve pam kullanarak partitional kümeleme yaptım.

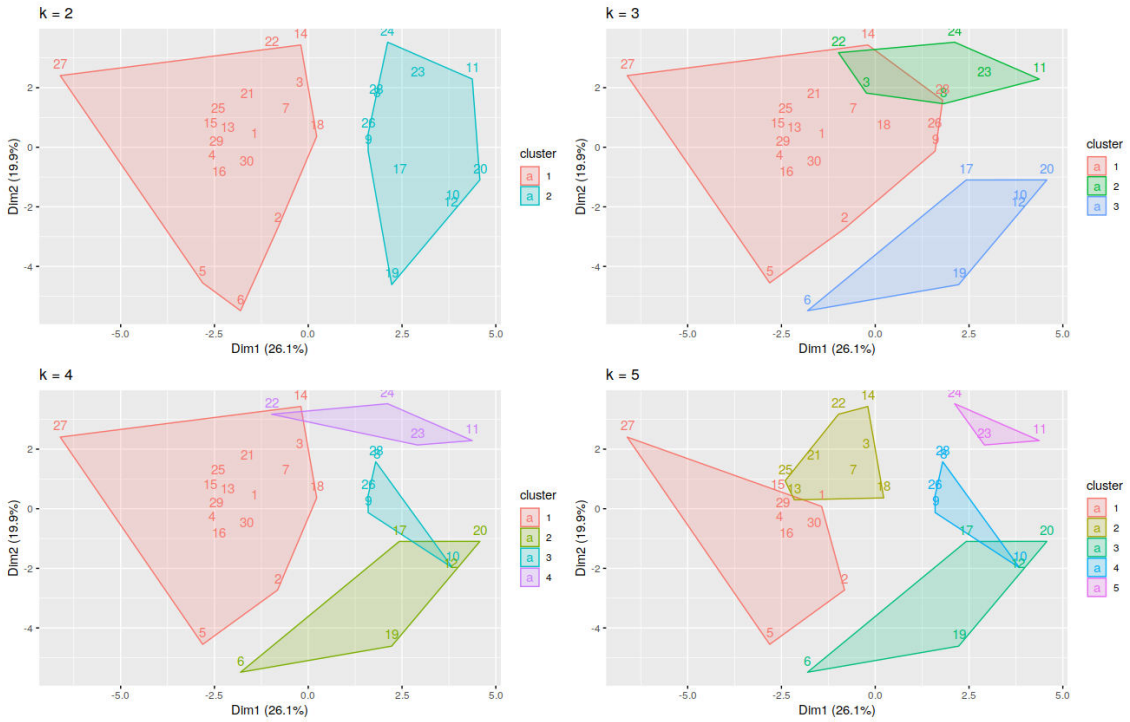




Optimum küme sayısını belirlemek için “average silhouette” plotunu inceledim. Görüldüğü üzere 2,3 ve 5 değerlerinin genişlikleri birbirine oldukça yakın. Üçünü de inceleyebiliriz.

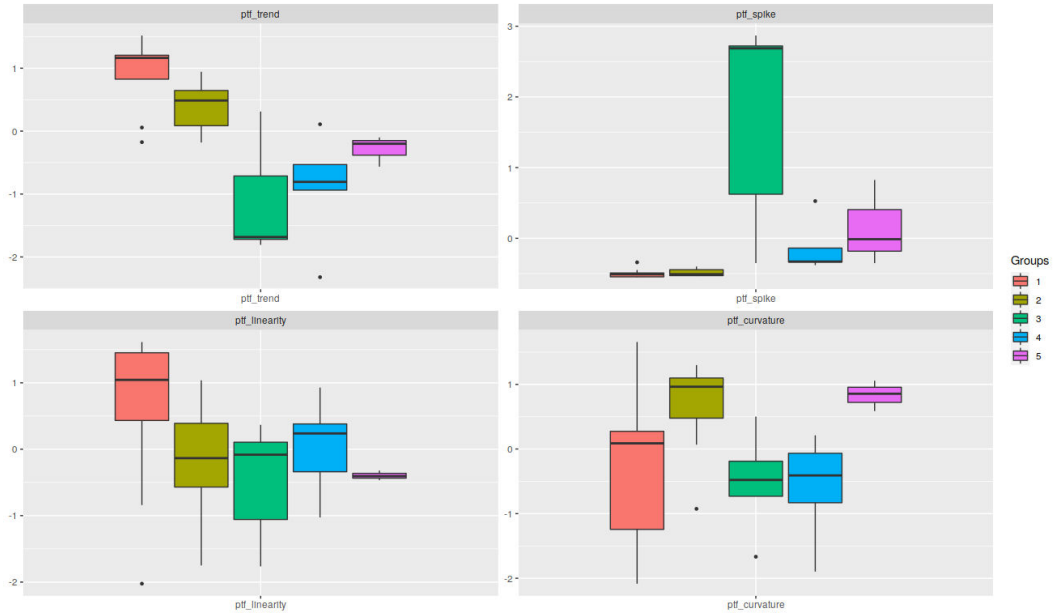


Farklı küme sayılarının kümelemeye nasıl etki ettiğini görmek için aşağıdaki görseli inceleyebiliriz. Burada değişken uzayı birinci ve ikinci pirincipal componenta indirgendiğinde kümelerin nasıl ayrıştığını görebiliriz. Silhouette metodunun da tavsiye ettiği gibi 2 ve 5 küme sayısı kümeleri olabildiğince ayrık olacak şekilde bölmüş gözüküyor.



Değişkenlerin kümelemeyi nasıl etkilediğini, veya kümelerin hangi değişken değerlerini ya da aralıklarını temsil ettiğini incelemek için aşağıdaki boxplotlara bakabiliriz. Örnek olarak ilk 4 değişkeni 5 küme olarak grupladım. Örneğin sağ alt köşedeki ptf_curvature değişkeni 2. ve 5. kümeleri değerlerinden ayırmışa benziyor. Bu kümelerin ptf curvature değeri yüksekken diğerlerininki daha düşük. Sağ üst köşedeki spike değerine bakarsak da 3. kümedeki günlerin spike değerlerinin diğerlerine göre çok daha fazla olduğunu, sonra 5. ve 4. kümedeki günlerin geldiğini görüyoruz. 1. ve 2. kümedeki değerler ise birbirlerine çok yakın ve sıfırın altında. Tek tek group değerlerini incelemek için bir alttaki görsele bakabilirsiniz.

Boxplots for the Clusters




```
 Descriptive statistics by group
group: 1
              vars n  mean   sd median trimmed  mad   min   max range  skew kurtosis   se
ptf_trend        1 9  0.87 0.57   1.16    0.87 0.42 -0.18  1.52  1.69 -0.79    -1.01 0.19
ptf_spike        2 9 -0.50 0.07  -0.51   -0.50 0.05 -0.55 -0.34  0.21  1.26     0.36 0.02
ptf_linearity    3 9  0.57 1.24   1.04    0.57 0.81 -2.02  1.61  3.64 -0.99    -0.54 0.41
ptf_curvature    4 9 -0.24 1.15   0.09   -0.24 0.57 -2.08  1.66  3.74 -0.13    -1.23 0.38
Group            5 9  1.00 0.00   1.00    1.00 0.00  1.00  1.00  0.00   NaN      NaN 0.00
------------------------------------------------------------------------------------------------
group: 2
              vars n  mean   sd median trimmed  mad   min   max range  skew kurtosis   se
ptf_trend        1 8  0.39 0.39   0.49    0.39 0.38 -0.18  0.94  1.12 -0.17    -1.63 0.14
ptf_spike        2 8 -0.48 0.06  -0.51   -0.48 0.04 -0.54 -0.40  0.14  0.52    -1.67 0.02
ptf_linearity    3 8 -0.21 0.95  -0.13   -0.21 0.90 -1.75  1.04  2.79 -0.36    -1.40 0.34
ptf_curvature    4 8  0.65 0.74   0.96    0.65 0.39 -0.93  1.30  2.22 -1.09    -0.32 0.26
Group            5 8  2.00 0.00   2.00    2.00 0.00  2.00  2.00  0.00   NaN      NaN 0.00
------------------------------------------------------------------------------------------------
group: 3
              vars n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
ptf_trend        1 5 -1.12 0.92  -1.68   -1.12 0.19 -1.81 0.31  2.12  0.60    -1.67 0.41
ptf_spike        2 5  1.71 1.48   2.69    1.71 0.27 -0.35 2.87  3.22 -0.40    -2.03 0.66
ptf_linearity    3 5 -0.49 0.89  -0.08   -0.49 0.67 -1.76 0.37  2.13 -0.39    -1.92 0.40
ptf_curvature    4 5 -0.51 0.79  -0.48   -0.51 0.43 -1.67 0.50  2.17 -0.19    -1.56 0.35
Group            5 5  3.00 0.00   3.00    3.00 0.00  3.00 3.00  0.00   NaN      NaN 0.00
------------------------------------------------------------------------------------------------
group: 4
              vars n  mean   sd median trimmed  mad   min  max range  skew kurtosis   se
ptf_trend        1 5 -0.90 0.89  -0.81   -0.90 0.41 -2.32 0.11  2.43 -0.51    -1.37 0.40
ptf_spike        2 5 -0.13 0.38  -0.33   -0.13 0.07 -0.38 0.53  0.91  0.93    -1.12 0.17
ptf_linearity    3 5  0.04 0.75   0.24    0.04 0.86 -1.03 0.93  1.96 -0.24    -1.74 0.33
ptf_curvature    4 5 -0.60 0.82  -0.41   -0.60 0.63 -1.90 0.21  2.11 -0.54    -1.54 0.37
Group            5 5  4.00 0.00   4.00    4.00 0.00  4.00 4.00  0.00   NaN      NaN 0.00
------------------------------------------------------------------------------------------------
group: 5
              vars n  mean   sd median trimmed  mad   min   max range  skew kurtosis   se
ptf_trend        1 3 -0.29 0.24  -0.20   -0.29 0.15 -0.56 -0.10  0.46 -0.31    -2.33 0.14
ptf_spike        2 3  0.15 0.61  -0.01    0.15 0.50 -0.35  0.82  1.18  0.25    -2.33 0.35
ptf_linearity    3 3 -0.40 0.07  -0.40   -0.40 0.09 -0.47 -0.32  0.15  0.10    -2.33 0.04
ptf_curvature    4 3  0.83 0.24   0.86    0.83 0.30  0.59  1.06  0.47 -0.09    -2.33 0.14
Group            5 3  5.00 0.00   5.00    5.00 0.00  5.00  5.00  0.00   NaN      NaN 0.00
```