İLETİŞİM VE İLETİŞİM ARAŞTIRMALARINDA

RStudio® İLE

VERİ GÖRSELLEŞTİRME

Dr. Öğr. Üyesi İnanç ALİKILIÇ

İKSAD
Publishing House

İLETİŞİM VE İLETİŞİM ARAŞTIRMALARINDA

R STUDIO İLE

VERİ GÖRSELLEŞTİRME

Dr. Öğr. Üyesi İnanç ALİKILIÇ[1]

İKSAD Publishing House

[1] Malatya Turgut Özal Üniversitesi, Sanat Tasarım ve Mimarlık Fakültesi
Görsel İletişim Tasarımı Bölümü, Orcid id: https://orcid.org/ 0000-0001-9079-8420
Email: inanc.alikilic@ozal.edu.tr

Copyright © 2021 by iksad publishing house
All rights reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by
any means, including photocopying, recording or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of
brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. Institution of Economic Development and Social
Researches Publications®
(The Licence Number of Publicator: 2014/31220)
TURKEY TR: +90 342 606 06 75
USA: +1 631 685 0 853
E mail: iksadyayinevi@gmail.com
www.iksadyayinevi.com

It is responsibility of the author to abide by the publishing ethics rules.
Iksad Publications – 2021©

ISBN: 978-625-8007-70-1
Cover Design: İbrahim KAYA
November / 2021
Ankara / Turkey
Size = 16 x 24 cm

ÖN SÖZ

Günümüzde veri, veri analizi ve veri görselleştirme hayatın her anının içinde yer almaktadır. İnternet ortamında verilerin görselleştirilmediği bir alan hemen hemen hiç yoktur. İletişimin önemine inanan, görsel materyallerin gücünden yararlanmak isteyen tüm kişiler için kaynak bir kitap olmasını ümit ettiğim bu kitabım, değerli görüşlerinize ve eleştirilerinize sunulmuştur.

Bu kitabın yazılmasına dolaylı olarak vesile olan çok değerli sevgili eşim Prof. Dr. Özlem ALİKILIÇ'a, rahmetli babam İbrahim ALİKILIÇ ve annem Muazzez KILIÇ'a beni yetiştirdikleri için çok teşekkür ederim.

Kızım Ela ALİKILIÇ, bu kitabı sana hediye ediyorum. Umarım ben bu diyardan gidince baban ile gurur duymana vesile olur bu kitap.
Sevgiyle kalın.

İÇİNDEKİLER

ÖNSÖZ ..... **i**
GİRİŞ **2**
1. VERİ, ENFORMASYON, BİLGİ VE BİLGELİK **5**
1.1. Veri 6
1.2. Enformasyon 7
1.3. Bilgi 8
1.4. Bilgelik 9
2. VERİ GÖRSELLEŞTİRME **10**
2.1. Veri Görselleştirme Nedenleri ve Amaçları 11
2.2. Veri Görselleştirmenin Tarihçesi 13
2.3. Veri görselleştirme araçları 25
3. R DİLİ **27**
3.1. Neden R ile Veri Görselleştirme? 28
4. GÖRSELLEŞTİRME ÇEŞİTLERİ **29**
4.1. Histogram Grafikler 29
4.2. Çubuk Grafikler (Bar Chart) 37
4.3. Nüfus Piramit Grafikleri (Population Pyramid) 45
4.4. Kutu- Bıyık Grafikleri (Box Plot) 46
4.5. Viyolin (Viyolonsel) Grafikleri (Violin Plot) 48
4.6. Hedef Grafiği (Bullet Graph) 49
4.7. Mum Grafik (Candle stick Chart) 51
4.8. Zaman Çizelgesi- Gantt Grafiği (Gantt Chart) 53
4.9. Isı Haritası (Heatmap) 56
4.10. Saçılım Grafikleri (Scatter Plot) 57
4.11. Çizgi Grafikleri (Line Plot) 61
4.12. Alan Grafikleri (Line Plot) 68
4.13. Ridge Grafiği (Ridge Graph) 71

4.14. Lolipop Grafiği (Lollipop Chart) 72
4.15. Radar Grafiği (Spider- Radar Chart) 74
4.16. Radyal Çubuk Grafik (Radial Bar Plot) 76
4.17. Sarmak Çizim Grafiği (Spiral Plot) 78
4.18. Güneş Yanığı Grafiği (Sunburst Graph) 79
4.19. Dairesel Grafik, Gül Diyagramı (Nightingale Rose Chart) . 82
4.20. Pasta Grafiği (Pie Plot) 85
4.21. Halka Grafiği (Donut Chart) 86
4.22. Ağaç Haritası Grafiği (Tree Map) 88
4.23. Ağaç Diyagram- Karar Ağacı (Tree Diagram- Decision Tree) 93
4.24. Akış Grafiği (Flow Chart) 96
4.25. Sankey Diyagramı (Sankey Diagram) 99
4.26. Akor Diyagramı – (Chord Diagram) 105
4.27. Ark Diyagramı (Arc Diagram) 106
4.28. Ağ Diyagramı (Network Diagram) 107
4.29. Paralel Setler, Paralel Koordinat Grafiği (Parallel Sets) ... 109
4.30. Kelime Bulutu (Word Cloud) 110
4.31. Koroplet Harita (Choropleth Map) 113
4.32. Bağlantı Haritası (Connection Map) 115
4.33. Nokta Haritası (Dot Map) 116
4.34. Dairesel Grafik (Circle Packing) 118
4.35. Piktogram Grafiği (Pictogram Chart) 120
5. R ile Veri Görselleştirmede Animasyon 125
6. R ile Veri Görselleştirmede 3D uygulamalar 132
7. SONUÇ 137
KAYNAKÇA 139

BU KİTAP KİMLER İÇİN

İletişim ve İletişim Araştırmalarında R Studio ile Veri Görselleştirme; iletişime gönül vermiş, verilen mesajın sadece sözle değil yazı ve görsel ile de verilmesini önemseyen, bilginin görsel olarak sunulmasının avantajlarına inanan ve sayıların gücünü önemseyip bu sayıların görsel olarak estetik, düzenli ve ilgi çekici olarak sunulmasını destekleyen kişilere yazılmıştır.

Kitabımızda veri görselleştirme uygulamaları R dili kullanılarak örneklendirilmiştir. Kitabın hiçbir yerinde R dili anlatımı yapılmamıştır. Bu nedenle temel ve orta seviye R dili bilgisi bu kitaptan yarar sağlamak isteyenler için ön şart olarak karşımıza çıkmaktadır.

Örneklerde kullanılan veri setleri ya örnek için varsayımsal olarak yaratılmış, ya da R paketleri içinden her kullanıcının rahatça erişim sağlayabilmesi düşünülerek alıntılanmıştır. Yer verilen her görselleştirme örneği içinde kullanılan paket bilgisine yer verilmiştir. Bu sayede okuyucular hangi paketi kullanmaları gerektiğini örnek içinde anlayabileceklerdir.

Kitapta geçen örneklerde kullanılan R paketleri ile ilgili paket açıklamaları atıflar içinde bulunmaktadır. R paketlerinin nasıl kullanılacağı örnek içinde gösterilmiş, lakin ayrıntılı, her paket ile ilgi açıklamalara kitap içinde yer verilmemiştir. Bir paketin içeriğinde yüzlerce ihtiyaca yönelik fonksiyon olabilmektedir. Bu nedenle paket ile ilgili daha detaylı bilgiye atıflardan ulaşılabilecektir.

GİRİŞ

İletişim en temel tanımla mesajlar aracılığı ile gerçekleştirilen toplumsal bir etkileşim sürecidir (Ertan ve Sansarcı, 2016, s. 85). İletişim en az iki taraf arasında bir eylem olarak karşımıza çıkmaktadır. Türkçe dilinde iletişim, ileti ve iletme sözcüklerinden bu nedenle türetilmiştir. İleti "dilsel bildirişim eyleminde konuşucunun belli bir dizgeye uygun olarak oluşturulup dinleyiciye yönelttiği göstergesel bir bütündür." (Vardar ve Güz, 1988, s.46).

Becer, iletişim sürecinin temelde beş unsurdan oluştuğunu vurgular. Bunlar sırası ile gönderici, iletilen mesaj, iletişim aracı yani kanal, alıcı ve dönüttür (Becer, 1997, s.14).

İletişim yazılı ve sözlü olabileceği gibi görsel göstergelerin de karşı tarafa iletilmesi ile yapılabilmektedir. Göstergelerin iletişim öğesi olarak kullanılmasıyla ortaya çıkan iletişim türü görsel iletişimdir. Görsel göstergeler; rakamlar, resimler, çizimler ve fotoğraf olabilir. Görsel öğelerin taşıdığı anlam ve bu anlamı taşıyan görseller görsel iletişimin inceleme konusu içinde yer almaktadır.

Görsel iletişim içinde büyük önem ve anlam kazanan *gösterge*; kendisinden başka bir şeyi anlatan, temsil eden anlam yüklü öğelerdir. "Kendisi o şey olmadığı halde, o şeyi çağrıştırarak iletişim sağlayan ve bir başka şeyi temsil eden her şey bir göstergedir." (Günay ve Parsa, 2012, s.9). Saussure'ye göre; gösterge, gösteren ve gösterilenden oluşan anlamlı bir fiziksel nesnedir. Gösterilen verilmek istenen kavram, gösteren ise bu kavramı aktaran imge olarak nitelendirildiği

görülmektedir. *Gönderge* ise gerçek nesne olarak karşımıza çıkmaktadır. Buradan yol çıkarak tüm kültürel ve coğrafi ögeleri iletişim etkinliği içeresinde tartışan, irdeleyen ve bu öğe ve oluşan dizgilerin okunabilmesine ve anlamdırılması üzerine çalışmalar yapan bilime *Göstergebilim* denmektedir. Bir sözcüğün, sesin, görüntünün veya bir imgenin ifade ettiği anlamın keşfedilmesini göstergebilimin konusunu oluşturmaktadır (Ertan ve Sansarcı, 2016, s.21).

Görsel göstergebilim (*visual semiotics*) ise grafik temsillerinin hem bilgiyi hem de bununla ilgili belirsizliği gösterebilecek mekanizmaları kavramsal bir temele oturtmak için bir teorik çerçeveyi sunmaktadır. Haritacılık, bilgi görselleştirme, semboller ve ikonlar bu temel içinde irdelenebilmektedir. Görüldüğü gibi veri görselleştirme temelde iletişim ve görsel iletişim içinde, göstergebilim ve görsel göstergebilim çatısı altında yer alabilmektedir.

İnsan beyni görsel materyalleri metinlerden çok daha hızlı işler. Bu nedenle bir mesajın verilmesinde verilerin görsel olarak temsili gayet mantıklıdır. Veri görselleştirme, insanların büyük ve karmaşık fikirleri basit yollarla ifade etmelerine olanak tanıyan bir hikâye anlatım biçimidir. Son 20 yılda veri görselleştirme verileri keşfetmek, analiz edip anlamlı sonuçlar çıkarmak, farklı branşlardaki kullanıcılara sunmak için çeşitli disiplinler tarafından benimsenmiştir. Yurtdışında birçok üniversitede veri görselleştirme ve iletişim adı altında dersler bulunmaktadır (*Data Visualization and Communication*). Özellikle geniş kitlelerle iletişim için görselleştirme tasarımının temel ilkelerin daha iyi ve yaygın bir şekilde anlaşılması önem kazanmıştır.

Günümüzde her alandaki buna iletişim ve iletişim araştırmaları da dahil, büyük veri kümeleri ile analizler yapılmaya başlanmıştır. Büyük veri setleri ile uğraşanlar insanların çıkan analiz sonuçlarını daha iyi anlayabileceği biçimde görselleştirerek kişilerin karar verme aşamasında daha çabuk geçmelerini sağlamayı amaçlamaktadırlar. Bu nedenle veri görselleşmenin önemi ve kullanım alanları her geçen gün çok daha fazla artacaktır.

Sosyal bilimlerin her alanında, iletişim ve iletişim araştırmalarında raporlama gereksinimlerinde elde edilen bilgilerin net ve basit şekilde iletilmesinde veri görselleştirme araçlarından yararlanılmaktadır. Medya analitiği raporlanmasında ve sunulmasında sıkça veri görselleştirme tekniklerinden yararlanılmaktadır. Sosyal medya analizlerinde ve raporlamasında veri görselleştirme teknikleri kullanılmaktadır. Elde edilen verilerin coğrafik olarak raporlanmasında son yıllarda veri görselleştirme uygulamalarına yer verilmektedir. Büyük miktarda verinin içerik analizi ile analiz edilmesinde ve elde edilen etiketlerin görselleştirilmesinde yine veri görselleştirme çalışmalarına yer verilmektedir. Bunun yanından her türlü nicel ve nitel araştırma sonuçlarının raporlanmasında veri görselleştirme ön planda yer almaktadır. Son yıllarda sıkça kullanılan metin madenciliğinde ve sosyal ağ analizlerinde yine farklı veri görselleştirme teknikleri kullanılmaktadır.

Sonuçta mesajın olduğu ve bu mesajın geniş kitlelere basitçe anlatılmasında en etkin araçlardan birisi olarak veri görselleştirme öne çıkmaktadır. Bu kitapta okuyucuların verileri analiz etmelerine

yardımcı olmak ve veri sonuçlarını etkili bir şekilde sunmaları için gereken yöntemler ve araçlar aktarılmıştır. Veri tanımından, veri tarihine ve oradan da R ile veri görselleştirme uygulamalarına bir gezintiye bu kitap ile çıkacağız. Bu kitabın ikinci baskısında çok önem verdiğim ama bu kitap için yetişmeyen *Bilgi Görselleştirme* (*Infographic*) konusunu da ekleyeceğim.

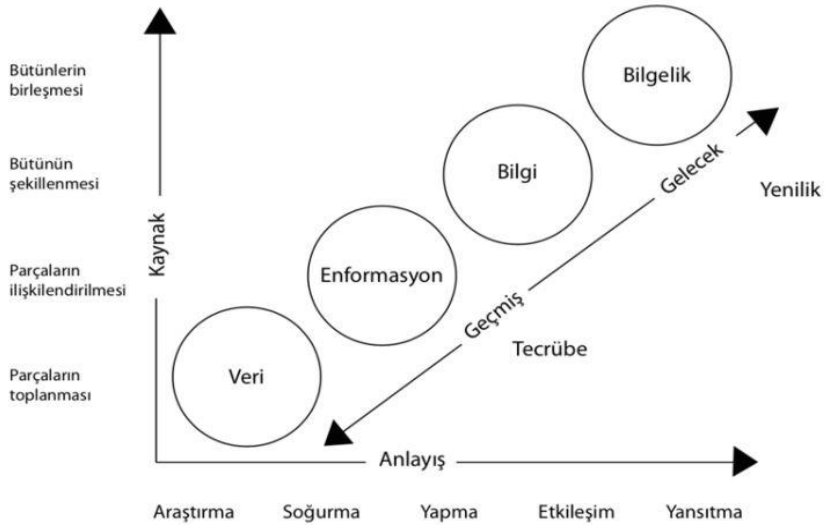
1. VERİ, ENFORMASYON, BİLGİ VE BİLGELİK

Veri ile başlayan ve sonrasında verinin enformasyona dönüşmesi ile şekillenen ve sonrasında elde edilen enformasyonun bilgiye dönüşmesi ve nihai olarak bilgiden de bilgelik mertebesine ulaşılmasını tasfir eden hiyerarlişi; bilişim sistemini kavramada bize ışık tutan en anlaşılır yollardan birisidir. Zeleny(1987) ve Ackoff(1989) tarafından geliştirilen hiyerarşisi, veriden bilgeliği giden yolu net bir şekilde ortaya koymaktadır.

Bu kavramlar topluca ilk kez Harlan Cleveland'ın *Information as Resource* adlı makalesinde kullanılmıştır (Cleveland, 1982). VEBB hiyerarşisine halen kavramsal eklemeler devam etmektedir. Örneğin, Russell Ackoff Ackoff (1989), VEBB hiyerarşisi olarak bilinen bu piramite *kavrayış* kademisini de dahil etmiştir.

VEBB hiyerarşisine göre sınırları çok net olmasa da bir hiyerarşi olan piramit düşünülmektedir. Piramittin en altında en geniş alanda veri; yani "gerçek" bulunmaktadır. Veri sonrasında ise "enformasyon" kavramı yer almaktadır. Enformasyon kavramından kasıt "ne olduğunu bileme" dir. Enformasyona ulaşıldıktan sonra "bilgi" yanı "nasılı

bilme" evresine geçilir. Piramidin en tepesinde ise nihai hedef olan "bilgelik" yani "nedenini bilme" olgusu karşımıza çıkar.



Grafik 1: VEBB Hiyerarşi Şeması (Öğüt 2012, s. 58)

1.1. Veri

Kurtaran ve Çubukçu, veriyi "işlenmemiş bilgi" olarak basitçe tanımlar (Kurtaran ve Çubukçu ,1991, s. 177). Barutçugil ise Bilgi Yönetimi adlı kitabında, veriyi "özümlenmemiş ve yorumlanmamış gözlemler, işlenmemiş gerçekler" olarak betimlemiştir (Barutçugil, 2002, s. 57).
Öğüt ise bu tanımı genişleterek veriyi "bilgi işleme sürecinin temel hammaddesi olarak ve çeşitli sembol, harf, rakam ve işaretlerle temsil edilen, ham, işlenmemiş gerçekler ya da izlenimler" olarak tanımlamıştır (Öğüt, 2012, s. 11). Işıklı, veri tanımını bir adım öteye taşımış ve veriyi "kaydedilen, yeniden düzenlenebilen ve analiz edilebilen bilgi (İngilizce: *information*) birimi" olarak tanımlamaktadır (Işıklı, 2014, s. 92).

Veri herhangi bir biçimde; kullanılabilir ya da gizli, örtük veya işe yaramaz şekilde bulunabilir. Bu durumda verinin kendi başına anlamı olmamaktadır. Fricke, bu neden ile veri "nesnelerin özelliklerini, olayları ve ilişkili çevreleri tanımlayan semboller" olarak tanımlamıştır (Fricke, 2009). Bu tanımdan yol çıkarak veriyi gözlemler ile elde edilen, yapılan gözleme dair kayıtlar olarak da betimlemek mümkün olabilecektir.

Yukarıdaki tanımları değerlendirdiğimizde elde edilen verinin işlenmemiş, saf ve ham olması ve içinde bilgi taşıyor olması gerektiği sonucuna ulaşılabilir. Bu verinin işlenmesi, analiz edilmesi ve bu yolla anlamlı hale gelmesi ile bilgiye yani enformasyona ulaşılabilecektir. Toplanan verinin anlamlı bilgiye dönüştürülme, işlenme süreci yani Enformasyon çok eskilere, antik duvar resimlerine dayanmaktadır. M.Ö. 20000- M.Ö.18000 Paleolitik Dönem'e ait iki adet kemik İşango ve Lebombo Bölgesinde bulunmuştur. Bu kemikleri özel yapan ise üzerinde sıralı çentiklerin bulunmuş olmasıdır. Çentikli kemiklerin, o dönemki toplulukların gıda stoklarını takip etmek, ticari aktiviteleri izlemek adına yapıldığı, yani ham verinin işlenmesi için kullanıldığı düşünülmektedir. Her ne amaçla çizikler atılmış olursa olsun kemiklerin gözlemler sonucu elde edilen verilerin düzenlenmesi, analiz edilmesi için kullanıldığı söylenebilir (Morriss, 2009).

1.2. Enformasyon

Gündelik konuşma dilinde enformasyon kelimesi çok kullanılmamakta, bunun yerine bilgi veya veri kelimesi daha çok tercih edilmektedir.

Enformasyon ya da malumat; ham, işlenmemiş verinin ilişkili bağlantılar sonucunda anlamdırılması yani anlam kazandırılmış biçimidir (Aydın et al., 2015). Yararsız, ham şekilde duran verinin yararlı hale getirilmesi sonucu elde edilen çıktıya enformasyon denmektedir (Bellinger, Castro ve Mills, 2004). Kim, ne, nerede ve ne zaman gibi sorulara cevap verme yetisi olan enformasyon, veriler gibi olay ve nesnelerin özelliklerini temsil ederler ama veriler gibi dağınık değil, daha kullanışlıdır (Ackoff, 1989, ss. 3-9).

Çağıltay ve Tokdemir'e göre ise enformasyon, organize edilmiş verilerinden ulaşılan gerçeklik durumudur. Yazarlara göre enformasyon verilerin bir araya getirilmesi ve anlamlı bir hal verilmesinden sonra analiz edilerek işlenmiş bir hal almasıdır (Tokdemir ve Cagiltay, 2010).

1.3. Bilgi

Cairo, bilgiyi insanlar tarafından algılanabilen ve yöntemler kullanılarak ölçülebilen her türlü fenomen olarak tanımlamıştır (Cairo, 2012). Gözlemlerden elde edilen veri işlenerek, kişiler arasında farklı şekilde deneyimleyerek ve farklı özellikleri tespit edilerek bilgiye ulaşılabilmektedir (Shedroff ve diğ., 1999). Bu neden ile bilgi, bilen tarafından anlam kazandırıldığı, yaşamdan elde edilen tecrübe ve deneyimler ile yapılandığı için kişisel, dinamik ve esnektir (Durmuş, 2019).

Bilgi açık ve örtülü bilgi olmak üzere iki grupta incelenebilir. Kolaylıkla semboller ile paylaşabildiğimiz, kodlanabilen, söz ile

aktarılabilen bilgiye açık bilgi denmektedir. Bu tarz bilgilere hemen hemen her yerde erişebiliriz. İnternette, kitaplarda, görsellerde ve sesler yoluyla ulaştığımız bütün bilgiler bu bilgi kapsamında değerlendirilmektedir. Örtülü bilgi ise kişisel olup, biçimlendirilmesi zordur. Örüntülü bilgi, kişiye, deneyime ve içeriği bağımlı olarak oluşur (Aydın et al., 2015).

1.4. Bilgelik

Veriyi işleyip enformasyona dönüştüren ve sonra bu enformasyondan bilgi edinme sürecinin sonrasında ulaştığımız anlayış kazanma durumu bilgelik kavramı ile açıklanmaktadır. Bilgiyi nasıl kullanılacağını kavrama, elde edilen bilgiler ile sağlıklı analizler yapma ve durum karşısında onu kullanıp rasyonel kararları alma süreci bilgelik kavramı ile betimlenmiştir. Sürekli yeni verileri işleyen ve buradan çıkarımlar yapan insanın bilgisi artar ve gelişir. Veri, enformasyon ve bilginin toplanmasının nihai amacı ise bilge kararlar alabilmektir (Ahsan ve Shah, 2006).

Bilgelik diğer işlemsel süreçlerden farklı olarak geleceğe yöneliktir. Elde edilen bilginin rasyonel kararlarda kullanılmasını temsil eder bu yönü ile de akıl olarak da isimlendirilebilir (Durmuş, 2019).

2. VERİ GÖRSELLEŞTİRME

Veri Görselleştirme en basit tanımı ile elde edilen verilerin görsel temsil bilimidir (Friendly, 2008). Doğada, çevremizde gizli, dağınık halde bulunan veriler genellikler karmaşıktır. Verinin anlam kazanma serüveninde bu verilerin sadeleştirilmesi zihnimizde şekillenmesi için bazı yöntemler ihtiyaç bulunmaktadır. Verilerin görsel şekiller ve formlar olarak kodlanması hem hatırlamayı kolaylaştırmakta hem de beynin veri ile ilişki kurmasına aracılık etmektedir. Cleveland, "veriler görsel formlara kodlanırsa, insan beyni ilişkileri ve örüntüleri daha iyi tanımlayabilir ve kavrayabilir. Grafikler, verileri keşfetmemize ve başka hiçbir yaklaşımın başaramayacağı modelleri gözlemlememize olanak tanır" cümlesi ile durumu çok güzel özetlemektedir (Cleveland, 1982, s.5).

Veri görselleştirme ise bilginin daha iyi anlaşılması, analiz edilmesi ve kavranması adına elde edilen verinin grafiksel olarak ortaya konulmasıdır. Burada elde olan bilginin kullanıcılar tarafından anlamlandırılması ve kullanıcılar ile iletişim kurmak amaçlanmaktadır (Stephen, 2013).

Veri görselleştirme, "bazı şemalarda soyutlanmış olan 'verilerin' görsel temsil bilimidir. Veri görselleştirmenin temel amacı, grafiksel araçlar kullanarak bilgiyi daha net ve etkili bir şekilde iletmektir" (Gürler, Yılmaz ve Tekerek, 2018).

2.1. Veri Görselleştirme Nedenleri ve Amaçları

İnsan öğrenmesi üzerine yapılan çalışmalarda, metin yerine görselin daha hızlı ve etkili algılanması tespit edilmiştir. Okuma işlemi metin analiz hızını sınırlandırırken, resim ve imaj yorumlama hızı ise insanın algısal sistemi ile doğru orantıda yol almaktadır (Ward, Grinstein ve Keim, 2010).

Veri görselleştirme temelde belirli bir konuyla ilgili yoğun ve karmaşık bilgileri daha anlaşılır şekilde görsel olarak göreceli basit olarak sunma, bu sayede bilginin veya konunun daha anlaşılır ve akılda kalıcı olmasını sağlama görevi görür. Bu temel amaç ile konu veya bilginin etkisinin arttırılması insanları bu yolla daha kolay ikna edilmesi, yönlendirilmesi ve belki de harekete geçirilmesi, konu hakkında bağlantıların farklı şekilde kurulabilmesi, ilk başta pek de görünür olmayan yeni gizli örüntülerin ortaya çıkarılması için veri görselleştirme iyi bir araç olarak karşımıza çıkmaktadır (Kim ve DiSalvo, 2010; Dur, 2014).

Bir başka görüşe göre de veri görselleştirmenin iki amacı vardır (Faloutsos, 2012; Keim, 1997):

- Konunun fikirlerin, kavram veya kuralların bu sayede daha iyi anlaşılması,
- Veri görselleştirme ile yeni sonuçlara ulaşmak, yeni ilişkiler kurmak, yeni fikirlere ulaşmak, hipotezleri test etmek, yeni gizlik örüntüler keşfetmek.

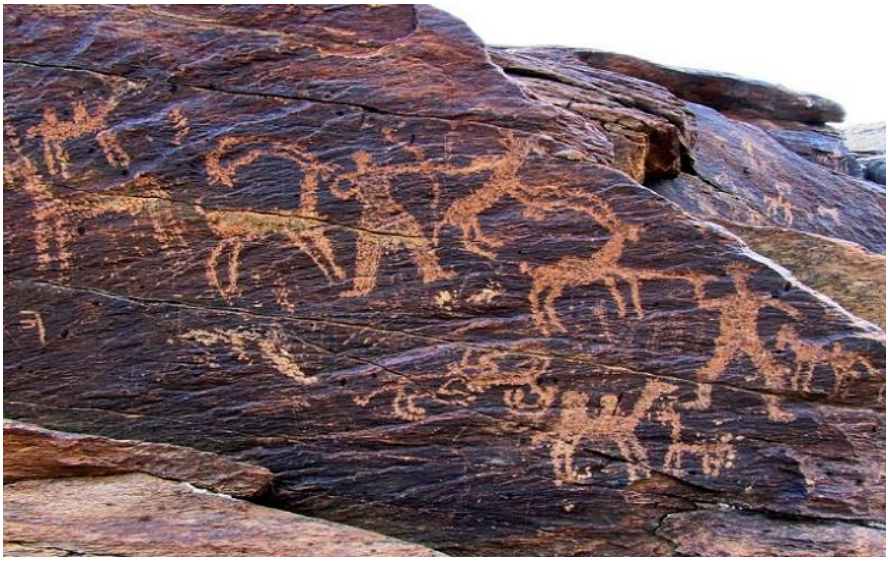
NeoMam Studios'un hazırladığı "İnfografiklerin Neden Tercih Edildiğini Açıklayan 13 Bilimsel Neden" başlıklı infografik, okuyuculara veri görselleştirmenin nedenlerini çok güzel açıklamaktadır.

"Beynin neredeyse %50'si görsel işlemleri kapsamaktadır. Tüm duyu alıcılarının %70'i gözlerdedir. Görsel bir sahneyi anlamak saniyenin 1/10'nu alır. Bir sembolün işlenmesi sadece 150 milisaniye, bir anlama eklemek ise 100 milisaniye sürmektedir. Etrafımızdaki bilgi sayısı ve büyüklüğü her geçen gün artmaktadır. Sadece iş dışında, ortalama bir günde 34 gigabayt veri tüketilmektedir. Bir web sayfasındaki kelimelerin ortalama bir ziyarette sadece %28'i okunmaktadır. Yapılan araştırmada renkli görsellerin okunma istekliliğini %80 oranında artırdığı tespit edilmiştir. Yapılan araştırmada ilaç etiketlerinin metin ve resim birlikteliği ile daha fazla okunduğu tespit edilmiştir. İnsanlar, yol tariflerini resim ve metin birleşimi ile daha iyi yapmaktadır. " (Durmuş, 2019). Wharton İşletme Okulunda yapılan bir araştırmada, hatırlama üzerinde görmenin ve tecrübe etmenin etkisi tespit edilmiştir.

Bilginin daha kolay anlaşılması ve akılda kalıcılığının artırılmasında görsel yollar kullanılarak yapılan sunum etkili olmaktadır. Bu sayede bilgi çok daha dikkat çekici olup, hatırlanmasının önü açılmaktadır. Birçok mesaja maruz kalan insanın mesajları seçmesinde algı önplana çıkarken görsel yolla sunulan bilgi de verinin bizim tarafımızdan algılanmasına önayak olmaktadır.

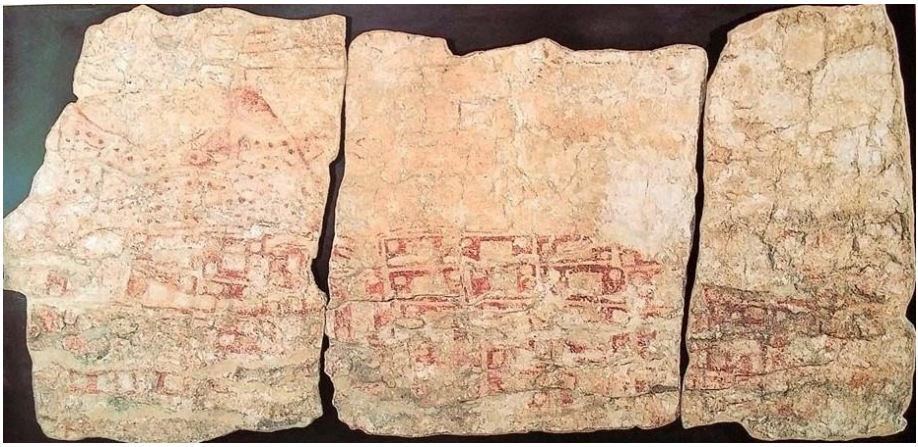
2.2. Veri Görselleştirmenin Tarihçesi

Veri görselleşmenin tarihsel gelişimi incelendiğinde sözsüz iletişimin en ilkel biçimi olan petrogliflere Antarktika kıtası hariç tüm kıtalarda rastlanmaktadır. Düşüncelerin ve verilerin taşlara oyulmak sureti ile bir sonraki nesillere aktarılması amaçlanmıştır.



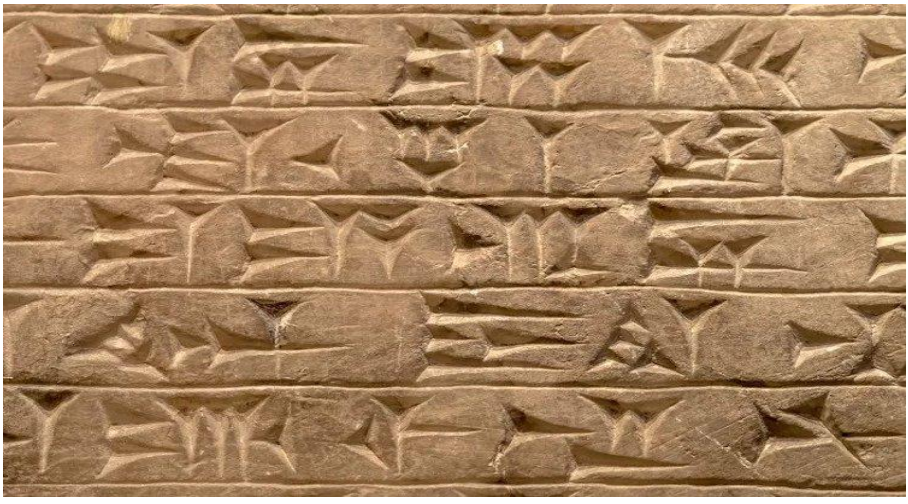
Grafik 2: İran Teimareh Bölgesinde Kaya Sanatı (Wikipedia 2012)

Çatalhöyük, M.Ö. 6800- M.Ö. 5700 yıllarında yaklaşık 10.000 insanın yaşamış olduğu büyük bir Cilalı Taş Çağı yerleşim alanı olup, Konya yakınlarında bulunmaktadır. 1963 yılında bir harita bulunmuştur. Bu haritanın uzunluğu 3 metre, genişliği ise yaklaşık 1 metredir. Bu haritada da veri görselleştirmenin ilk örneklerinden birisi sayılabilir.



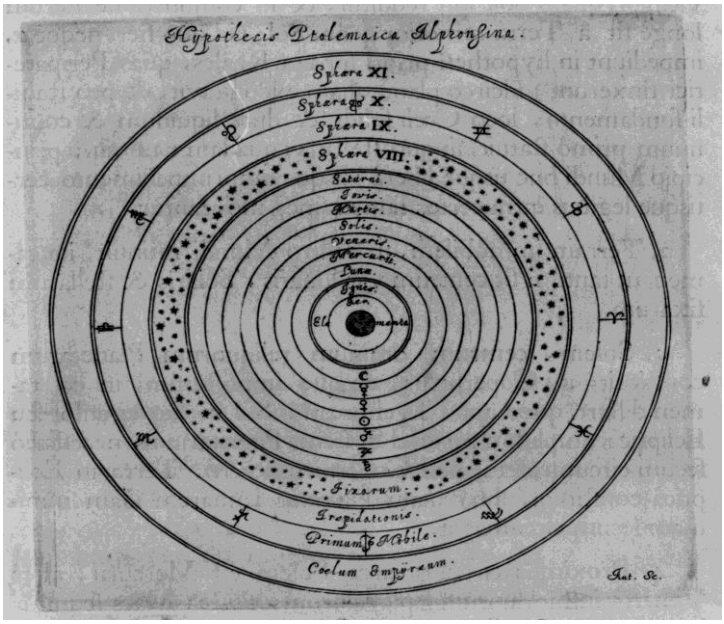
Grafik 3: Çatalhöyük Haritası (Harita Müzesi, 2015)

Petroglifler zamanla evrilerek Sümer Çivi Yazısına (M.Ö. 3300) ve Mısır Hiyerogliflerine dönüşmüşler, sonrasında ise günümüzde kullandığımız dilsel yazıların zeminini hazırlamışlardır.



Grafik 4: Sümer Çivi Yazısı Örneği (Arkeofili, 2019)

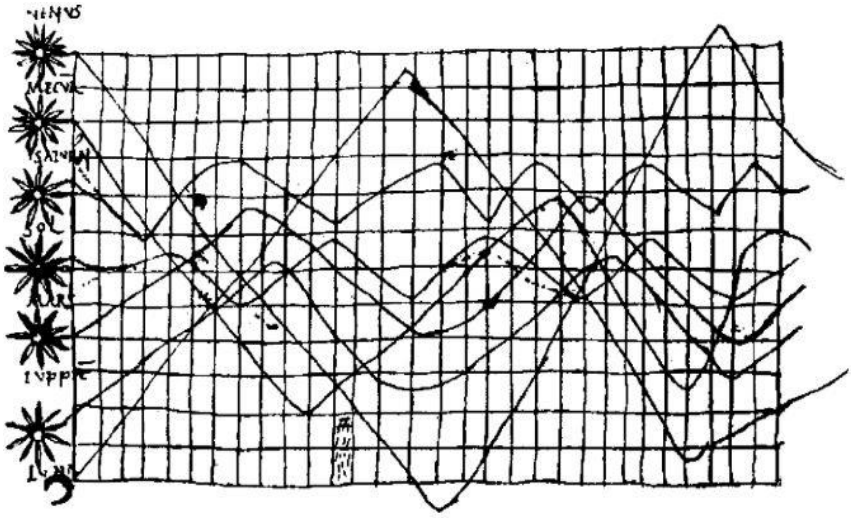
Friendly'nin aktarımlarına göre görselleştirmenin ilk örneklerinden birisi de 100- 170 yılları arasında yaşamış olduğu kabul edilen matematikçi, coğrafyacı ve astronom olan Klaudyos Batlamyus (Ptolemy) yapmış olduğu çalışmalardır. Önerdiği *dünya* merkezli bir güneş sistemi ve Yunanlıların tarafından bilinen dünyanın büyüklüğü, haritalandırması ve bugünkü enlem ve boylama benzer ilkel ölçeklendirmesi ile veri görselleştirmeye katkı sağlamıştır (Friendly, 2008).



Grafik 5: Batlamyus Güneş Sistemi (Budge 2015)

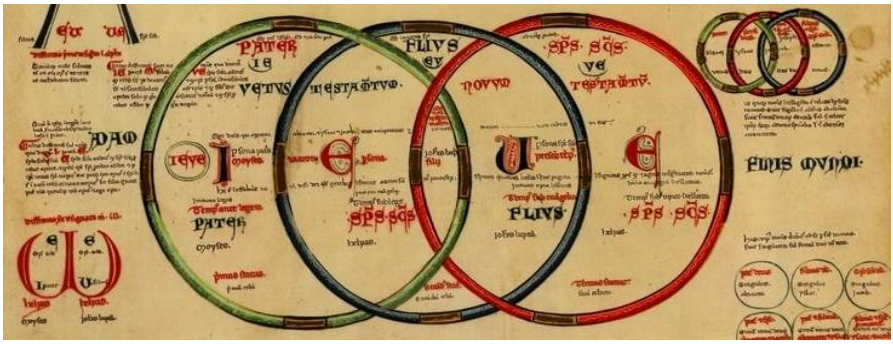
Uzay-zaman boylamında en önemli yedi gök cisminin tasvirinin yapıldığı ve yapanın belli olmadığı anonim grafiksel çalışmanın onuncu yüzyılda yapıldığı düşünülmektedir. Bu veri görselleştirmede dikey eksende gezegenlerin yörüngelerini, yatay eksende 30 eşit aralığa

bölünmüş zaman veri görselleştirme ile betimlenmiştir (Friendly, 2008).



Grafik 6: 10.ncu Yüzyıla Ait Anonim Grafik Funkhouser'den (1936, s.261) Aktaran (Friendly 2008)

Joachim'li Fiore (1135-1202)'nin Daire Üçlemesinde Fiore, dini inancına göre baba, oğul ve kutsal ruhu diyagram çizerek görselleştirmiştir.

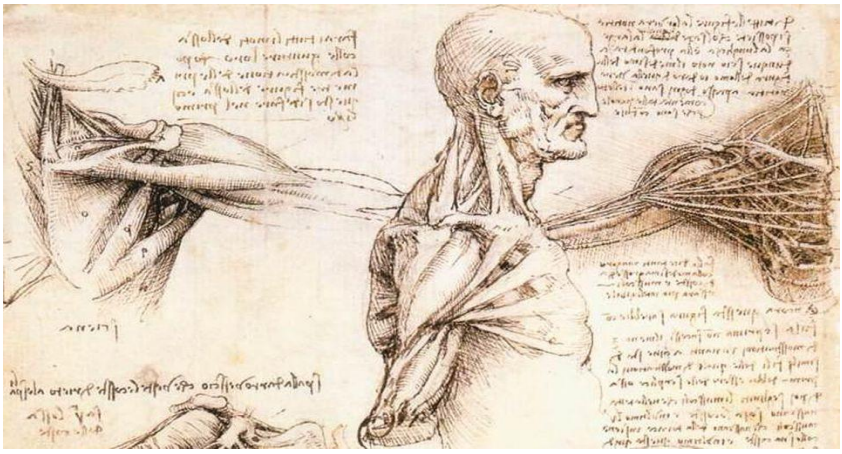


Grafik 7: Fiore'nin Daire Üçlemesi

1350 yılında, Orta çağ filozofu ve rahip olan Nicole d'Orseme hareket eden bir objenin nasıl görselleştirmesi konusunda çalışmalar yapmıştır, ayrıca değerlerin birbirleri arasındaki ilişkileri açıklamak veya zaman içinde değerlerin nasıl değiştiğini ve büyüklüklerini tasvir etmek için grafikler kullanmıştır (Tufte, 2006; Gürler, Yılmaz ve Tekerek, 2018).

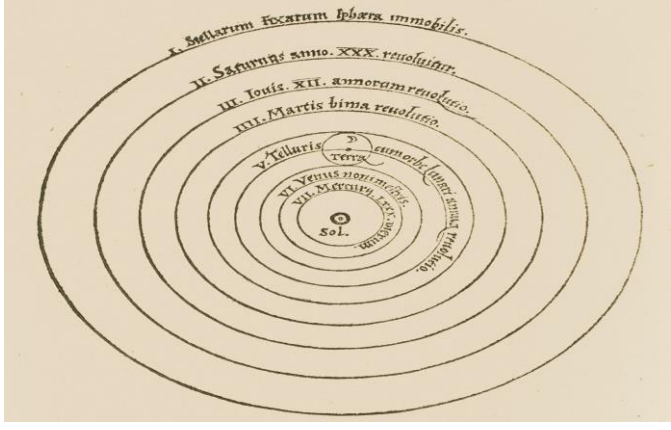
Grafik 8: Oresme'nin Orantılar ile Hareketin Gösterimi (Eğitimde Bilim Tarihi, 2020)

Leonardo da Vinci (1452-1519), günümüzdeki X-Ray ve diğer görüntüleme tekniklerinden yoksun insan anatomisi üzerine kapsamlı resimli bilgi grafikleri oluşturmuş, anatomik çizimleri metin ile destekleyerek okuyucuları bilgilendirmiştir. Onun çalışmaları da veri görselleştirme örnekleri olarak değerlendirilmektedir (Smiciklas, 2012).



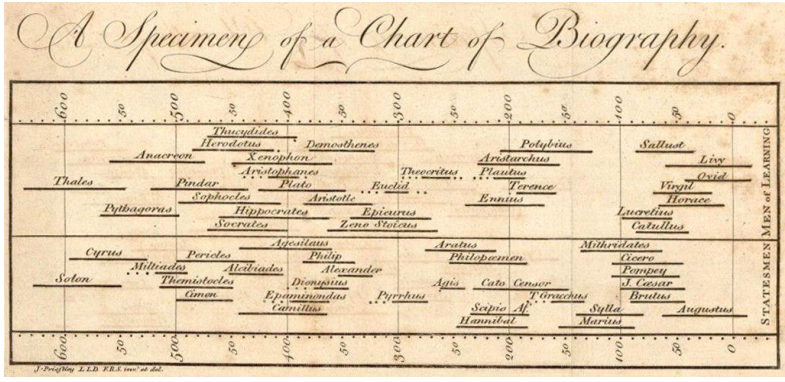
Grafik 9: Leonardo da Vinci Veri Görselleştirme Örneği (Forbes, 2021)

Nicolas Copernicus'un *De Revolutionibus Orbium Coelestium* eserindeki Gezegenler Diyagramı (1543), binlerce yıldır gözü kapalı inanılan dünya merkezli dünya anlayışını yıkmıştır. Güneşi merkeze alan evren anlayışını kitabında görselleştiren Copernicus bu sayede sabit bir düşünce sisteminin yıkılmasına önayak olmuştur (Gürler, Yılmaz ve Tekerek, 2018).

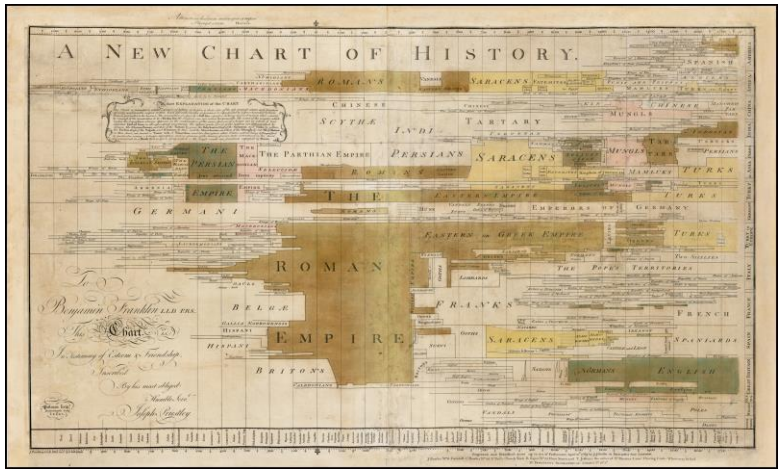


Grafik 10: Copernicus Gezegenler Diyagramı

1765'te Joseph Priestley, *Chart of Biography* isimli çalışmasında birçok meslek grubundan (devlet adamları, savaşçılar, metafizikçiler, doktorlar, matematikçiler v.b.) 2000 kişinin (M.Ö. 1200- 1800) yaşam aralıklarını görselleştirmiştir. Ayrıca aynı yazarın *New Chart of History* isimli bilgi görselleştirmesi ise yine erken dönem veri görselleştirme çalışmaları olarak değerlendirilmektedir.



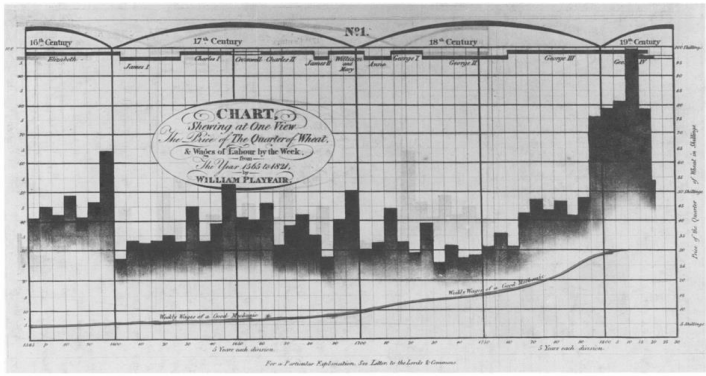
Grafik 11: Priestley'in Zaman Çizelgesi (1765) (Marshall 2016)



Grafik 12: Priestley'in New Chart of History Çalışması (1769) (Marshall 2016)

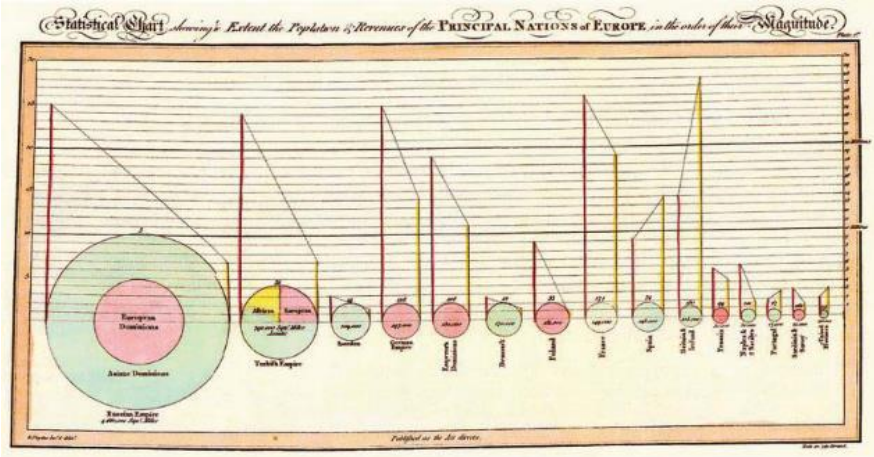
Bildiğimiz anlamda elde edilen matematiksel verilerin ve istatiksel ifadelerin görselleştirmesi ise mühendis ve siyasal ekonomist olan İskoç mucit *William Playfair* (1759-1823)'in çalışmaları ile başlamıştır. Yazara kadar olan dönemde mevcut ekonomik durumlar, tablolar aracılığı ile okuyucuya sunulmaktaydı. Yazar rakamların daha iyi anlaşılması adına ekonomik değişkenleri açıklamak için grafiksel gösterimleri kullanan ilk kişi olmuştur. İlk olarak çizgi ve çubuk grafiklerine eserlerinde yer vermiş, sonrasında ise pasta ve çember grafikleri literatüre kazandırmıştır.

Yazarın *Commercial and Political Atlas* (Ticari ve Politik Atlas) adlı yapıtında 44 çizelge bulunmaktadır. Üç zaman serisi analizin tek grafikte gösterildiği çalışmasında 1565-1820 yılları arasında 250 yıllık dönemde buğday fiyatları, haftalık işçi ücretleri ve kraliyetin hükümdarlık süresi bir arada görselleştirmiştir. Bu görsel ile işçilerin yaşam standartlarının son yıllarda daha da iyileştiğini kanıtlamaya çalışmıştır.



Grafik 13: Üç Zaman Serisi Analizin Tek Grafikte Gösterildiği Çalışma (Tufte and Graves-Morris 1983)

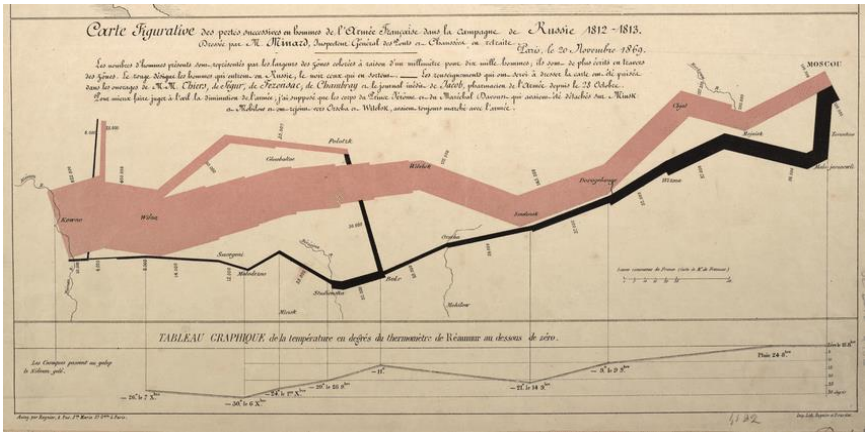
Yazarın *Statistical Breviary* isimlerindeki yapıtlarında yazar; pasta grafikleri, çizgi grafikler, çubuk grafikleri ilk kez birleştirip kullanmıştır. 1801 yılındaki Avrupa ülkelerini karşılaştıran grafikte büyük yuvarlak ülkelerin yüz ölçümünü, soldaki dikine çizgi nüfusu, sağdaki dikine çizgi ise toplanan kişi başına vergileri göstermektedir. Yazar bu çalışmasında nüfusa oranla toplanan vergilerdeki ülkelerin durumunu kıyaslamış ve İngiltere'de toplanan kişi başı vergi oranının çok yüksek olduğunu grafiksel olarak ortaya koymuştur (Chen, Härdle, ve Unwin, 2007; Playfair, 1801).



Grafik 14: Pasta, Çubuk ve Çizgi Grafiklerinin Beraber Kullanımı (Playfair, 1801)

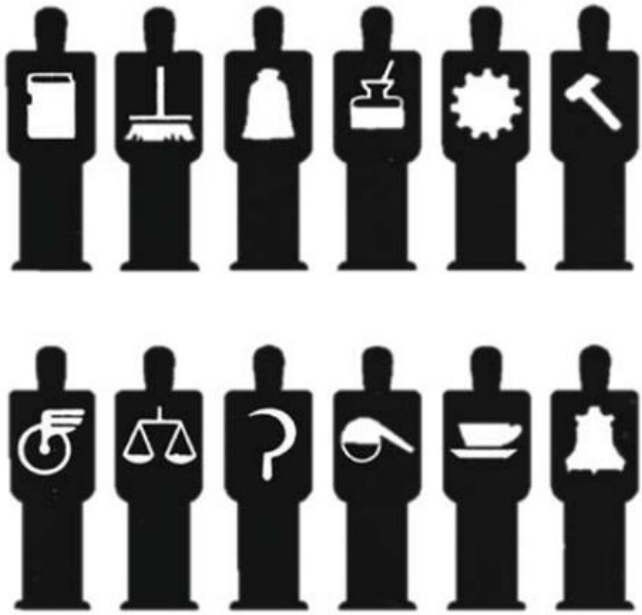
Charles Joseph Minard (1781-1870) ise aynı yıllarda yaptığı bir çalışmada belli bir zamanda Napolyon ordusunun Moskova'ya gidişindeki asker kayıplarını işlemiştir. Veri görselleştirmenin hikayeleştirildiği bu çalışma zaman içinde 422000 askerden Kovno'ya geri dönülürken 10000 askerin kaldığı dramatik olarak gözler önüne

serilmiştir. Enlem, boylam, hareket yönü, ısı ve ordu büyüklüğü çok değişkenli olarak okuyuculara verilmiştir (Dur, 2014).



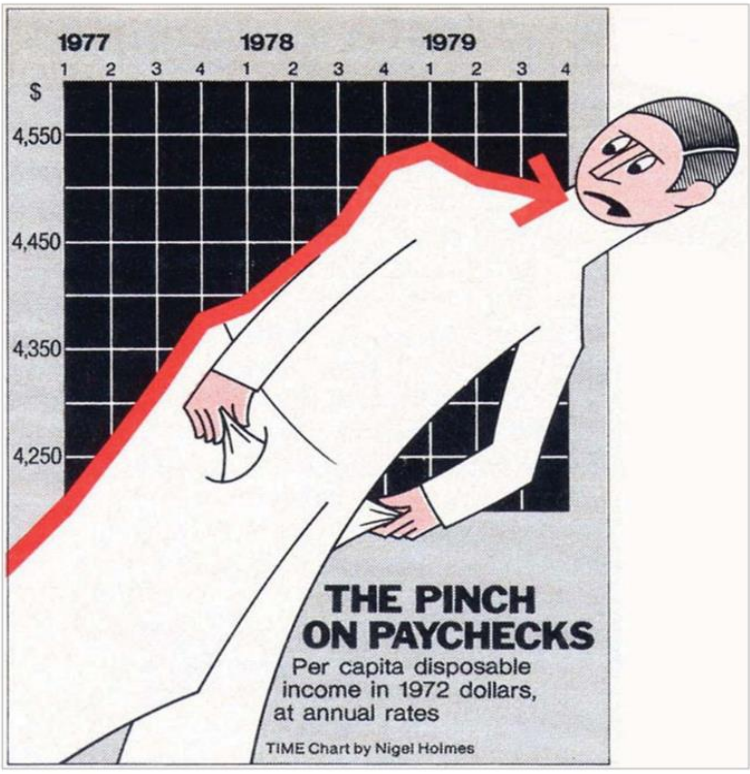
Grafik 15: Charles Joseph Minard'ın İnfografik Çalışması (http://www.edwardtufte.com/tufte/posters#cyclogram)

Otto Neurath tarafından geliştirilen *Isotype (International System of Typographic Picture Education)* teknolojik, tarihsel, biyolojik ve sosyal ilişkileri resimsel biçimde betimlemek üzere geliştirilmiştir. Neuath'ın çalışmaları rakamların ve bilginin kolayca görselleştirmesinden ziyade iletişim geliştirmek amacı ile tasarlanmıştır (Omran, 2019). Neurath'un "Uluslararası Resim Dili" kitabında farklı insan oraklarının sembolize edildiği bir resim aşağıda verilmiştir.

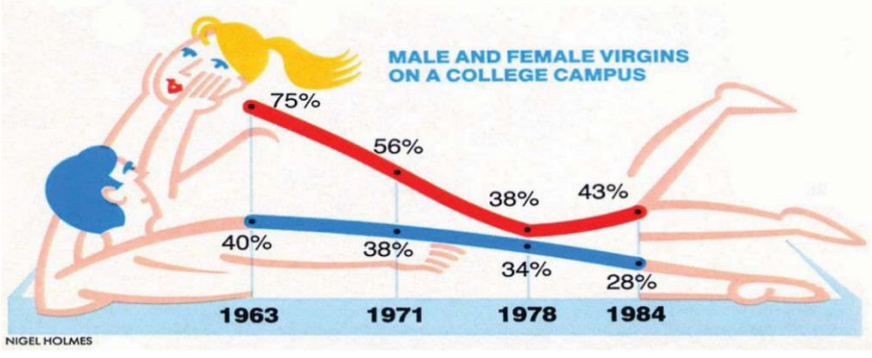


Grafik 16: Otto Neurath'ın Infografik Örneği

Neurath'ın görselleştirme bayrağını, 1978'den 1994'e kadar Time Dergisinde çalışan ve istatisksel bilgileri görsel olarak sunmak adına "açıklama grafikleri" kavramını yaratan Nigel Holmes taşımıştır. Açıklayıcı grafiklerinde duygu ve mizahı kullanarak verileri okuyucularına anlatmak isteyen Holmes, bu sayede okuyucuların ilgi ve dikkatini çekmeyi ve görsel kullanımı ile onların zihninde daha fazla yer etmeyi amaçlamıştır (Durmuş, 2019).



Grafik 17: Nigel Holmes'un Infografik Çalışmaları



Grafik 18: Nigel Holmes'un Infografik Çalışmaları

Grafik 19: Nigel Holmes'un Infografik Çalışmaları

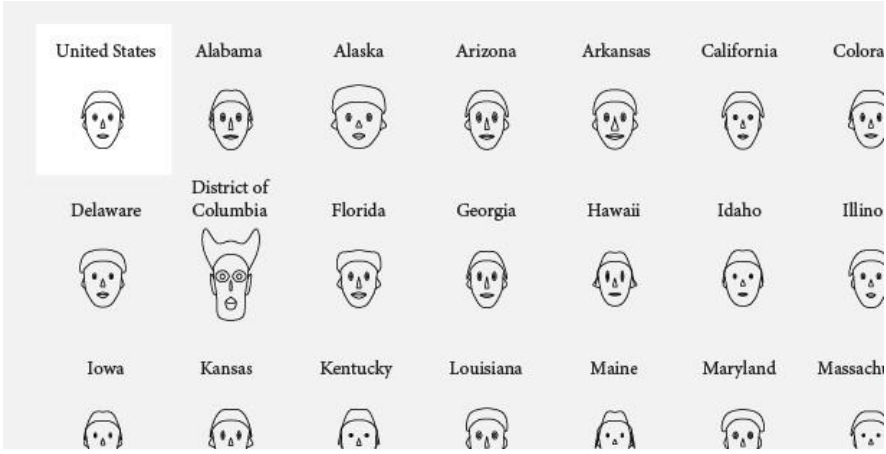
2.3. Veri görselleştirme araçları

Veri görselleştirmenin araçları üzerinden çalışmalar yapan ve bu yönde birçok araştırmaları ile tanınan Kriegel ve Keim bu araçları 6 sınıfa ayırmıştır. Aşağıda sınıflar ayrıntılı işlenmiştir (Berchtold, Keim ve Kriegel, 1996).

- **Geometrik iz düşüm tabanlı teknikler:** İki boyutlu bir veri setinde x ve y değerlerini x ve y ekseni boyunca kutupsal koordinat sistemi üzerinde işaretlendiği *Saçılım Grafikleri* bu tekniklere en güzel örnek olarak karşımıza çıkmaktadır. Bu tekniğin avantajları dezavantajları da bulunmaktadır. Çok fazla değer içeren büyük veri setlerinde saçılım grafiklerinde üst üste

binen çok fazla nokta olabilmektedir. Bu da veri setinin anlaşılabilir olmasının zorlaştırabilmektedir (Inselberg ve Dimsdale, 1990).

- **İkon tabanlı teknikler:** İkon tabanlı teknikler, veri görselleştirmede çok boyutlu bilginin ikonlar ile görselleştirildiği bir teknik olarak karşımıza çıkmaktadır. Bu tekniğin gelişimine ön ayak olan çalışma *Chernoff* yüzleri tekniğidir. İnsan suratında bulunan dudakların, gözlerin, burun boyutunun ve ağzı gibi bölümlerin veri setindeki bir değere göre değişmesi ile yüzün tamamen değiştiği bir teknik olarak karşımıza çıkmaktadır. Bu teknik güzel bir teknik olsa da insanın yüzündeki belli noktaların diğer noktalara göre göreceli daha fazla dikkat çekmesi, okuyucuları o noktalara yönlendirebilmektedir (Keim, 1997).



Grafik 20: Chernoff Yüzleri

- **Piksel tabanlı teknikler:** Çok büyük ve çok boyutlu veri setlerlerinin görselleştirmesinde kullanılan bu teknik her bir veri

parçasının değerinin farklı bir renk ile pikselleştirilerek görselleştirilmesi ile oluşmaktadır (Pickett ve Grinstein, 1988).

- **Hiyerarşik teknikler:** Hiyerarşik teknikler farklı boyuttaki uzayı farklı sayıdaki alt uzaylara ayırarak verinin hiyerarşik olarak görüntülemeyi sağlarlar (Berchtold, Keim ve Kriege, 1996).
- **Grafik tabanlı teknikler:** Grafik tabanlı teknikler, sorgulama dilleri ve bazı soyutlama teknikleri kullanılarak, algoritmalar kullanılarak ortaya etkili grafikler oluşturulmasıdır (Card, Mackinlay ve Shneiderman, 1999)
- **Karma teknikler:** Karma tekniklerde görselleştirmenin daha etkili ve akılda kalıcı olması için birçok veri görselleştirme tekniği bir arada kullanılır. Birçok pencere içinde farklı veri görselleştirme örnekleri kullanılması, okuyucuya daha fazla bilgi aktarılmasına zemin hazırlarken, okuyucunun verilen bilgiyi kavraması ve hatırlaması kolaylaştırılır (Cristina, Oliveira ve Levkowitz, 2003).

3. R DİLİ

1976'da John. M. Chambers ve arkadaşları tarafından verileri görselleştirme ve analiz etme amacı ile AT&T ve Bell laboratuvarlarında geliştirilen S dili ve sonradan lisanslanan S- Plus R dilini öncelemiştir.

*R*oss Ihaka ve *R*obert Gentleman tarafından geliştirilen, S dilinin açık kaynak ve GNU versiyonu öne çıkan R dili istatistiksel hesaplama ve grafik işlemleri için geliştirilen, diğer katılımcılarında katkı

sağlayabildiği, paketler üretebildiği ve geliştirebildiği açık kaynak kodlu biri programlama dilidir (Ihaka ve Gentleman 1996).

R dilinin açık kaynak kodları sayesinde kullanılan tüm fonksiyonlar ve algoritmalar kullanıcılar tarafından analiz edilebilir ve değiştirilebilir. R dilinin gelişmesine olanak sağlayan büyük bir topluluk desteği vardır.

R; makine öğrenmesi, veri bilimi, istatistik, matematik, ekonometri, tıp ve akademik diğer disiplinlerde sıkça kullanılmakta farklı alanlarda oluşan her ihtiyaç için ise paket veya kütüphane geliştirilmiştir. R dilinin bu esnek yapısı diğer programlama dilleri ile de sorunsuz konuşabilmesine olanak tanımaktadır. R esnek olduğu kadar interaktif bir dildir. Soru ve cevap şeklinde ilerleyerek derleyici bir dil olmaktan çıkıp, yorumlayıcı bir dil olarak tanımlanmaktadır.

3.1. Neden R ile Veri Görselleştirme?

R dilinin esnek, açık kaynak kodlu ve ücretsiz olması dilin farklı disiplinlere ve uzmanlık alanlarına paketleri ile çözümler bulması diğer dillere göre tercih edilmesinin nedenleri arasındadır. R dili ile ileri derecede veri işleme, veri analizi ve veri manipülasyonu yapılabilmesi, iletişim araştırmalarında da R dilinin kullanılmasının önünü açmıştır. Bu kitapta R dili ile yapılabilecek veri görselleştirme örneklerine yer verilmiştir. Grafiklerin neden yapıldığı, arkasındaki kuramın okuyucular tarafından bilindiği varsayılmıştır. Ayrıca bu kitapta ilgili paketlerin kullanılmasının öğretilmesi de amaçlanmamıştır. Kitapta amaçlanan okuyucunun bildiği konuda veri görselleştirmesi yapmasına olanak tanımaktır. Aşağıda verilen kodların için kullanılan paketler ve

açıklama gereken yerlerde # işareti sonrasında yapılan açıklama yer alacaktır.

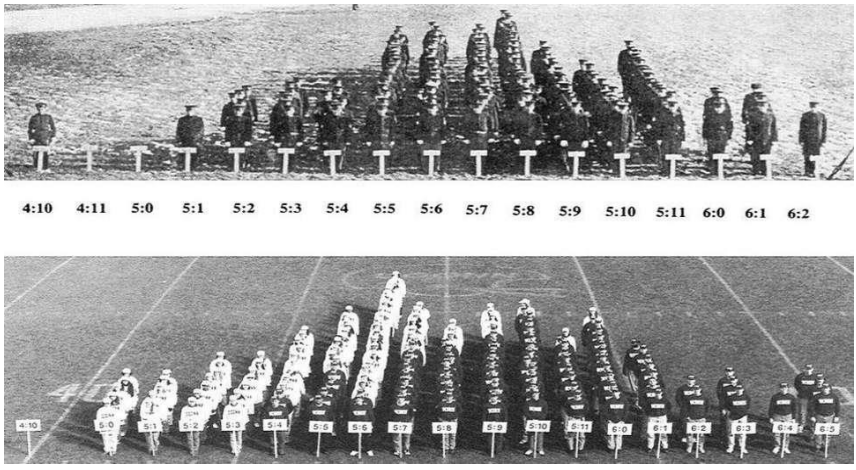
4. GÖRSELLEŞTİRME ÇEŞİTLERİ

Bu kitapta örneklendirilen veri görselleştirme türlerinin birçoğu Severino Ribecca tarafından oluşturulan http://datavizcatalogue.com adresindeki sistematik içinde değerlendirilmiştir. *Veri Görselleştirme Kataloğu* adı altında editörlüğünü Öğr. Gör. Pınar Dağ'ın yaptığı toplam 8 kişilik bir ekip ile Türkçe'ye de çevrilen çalışma kitabımızın ana çerçevesini oluşturmaktadır.

4.1. Histogram Grafikler

Histogram grafiği bir dağılım grafiğidir. Histogram bir dizi sürekli verinin temeldeki frekans dağılımını(şeklini) keşfetmenize ve göstermenize olanak tanıyan bir grafik türüdür. Bir çeşit çubuk grafik olan bu gösterim şekli *Karl Pearson* tarafından literatüre kazandırılmıştır(Pearson, 1895). Histogram verilerin temeldeki dağılımı(ör. Normal dağılım), aykırı değerler, çarpıklık vb. için incelenmesine olanak tanır. Histogram grafiği ile çubuk grafikler birbirleri ile çokça karışabilmektedir. Değişkenlerin dağılımlarını göstermek için histogramlar kullanılırken, değişkenleri karşılaştırmak için çubuk grafikler kullanılır. Çubuk grafiklerde çubuklar birbirlerine ayrı dururken, histogram grafiklerinde çubuklar birbirlerine bitişik halde yer alırlar. X ekseninde ayrılan sınıflar bulunurken y ekseninde bu sınıflara ait frekanslar veya yüzdeler yer almaktadır.

Histogram'ın daha iyi anlaşılabilmesi için Blakeslee'nin 1914 yılında çektiği resim örnek olarak verilebilir. 175 erkek öğrencinin boyalara göre sınıflandırılmasını konu alan resim sonrasında *Linda Strausbaugh* tarafından 80 yıl sonra kız öğrencilerde resime dahil edilerek yeniden canlandırılmıştır.



Grafik 21: Alfred Blakeslee (1914) ve Linda Strausbaugh Çalışması (Inglis-Arkell 2015).

Bir dersin öğrencileri üzerinde histogram grafiği oluşturulması aşağıda sunulmuştur.

```
cinsiyet<- c("Erkek", "Erkek", "Kız", "Erkek", "Kız"
,"Kız","Kız","Kız","Kız", "Kız", "Kız" ,"Kız", "Erkek", "Kız", "Kız",
"Erkek", "Kız","Kız","Erkek","Kız", "Kız",
"Kız","Kız","Erkek","Erkek","Kız","Erkek","Kız","Kız","Kız")


kilo<-
```

```
c(62,73,60,75,57,60,48,48,62,62,53,78,65,40,56,70,40,50,75,55,52,45,
52,80,102,59,58,70,54,54)

boy<-
c(181,181,157,185,155,165,160,164,170,163,160,172,180,145,165,17
5
,160,168,173,160,169,150,160,175,196,162,177,176,160,170)
ayakkabino<-
c(43,44,36,44,37,38,38,38,40,37,38,38,42,35,38,43,36,39,42,38
,38,36,37,43,44,36,43,39,36,39)
#kiloboy isimli bir data.frame (tablo) oluşturulması
kiloboy<- data.frame(cinsiyet, kilo, boy, ayakkabino)
```

Örneğimizdeki veri seti *summary* ve *str* fonksiyonları ile aşağıdaki şekilde detaylı incelenebilir.

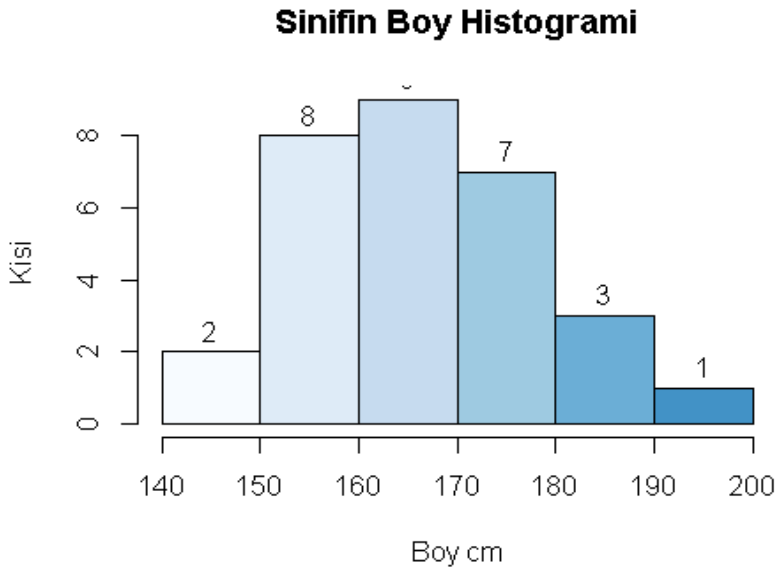
```
summary(kiloboy)
##    cinsiyet            kilo            boy          ayakkabino
##  Length:30          Min.   : 40.00   Min.   :145.0   Min.   :35.00
##  Class :character   1st Qu.: 52.25   1st Qu.:160.0   1st Qu.:37.00
##  Mode  :character   Median : 58.50   Median :166.5   Median :38.00
##                     Mean   : 60.50   Mean   :167.8   Mean   :39.17
##                     3rd Qu.: 68.75   3rd Qu.:175.0   3rd Qu.:42.00
##                     Max.   :102.00   Max.   :196.0   Max.   :44.00
str(kiloboy)
```

```
##    'data.frame':                30 obs. of  4 variables:
##    $ cinsiyet   : chr  "Erkek" "Erkek" "Kiz" "Erkek" ...
##    $ kilo       : num  62 73 60 75 57 60 48 48 62 62 ...
##    $ boy        : num  181 181 157 185 155 165 160 164 170 163 ...
##    $ ayakkabino: num  43 44 36 44 37 38 38 38 40 37 ...
```

Histogramı oluşturmak için R Studio içinde hazır gelen *hist* fonksiyonumuzu kullanacağız. Grafiklerimizi renklendirmek içinde *RColorBrewer* (Neuwirth, 2011) kütüphanesindeki renk seçeneklerini kullacağız. Örneklerde yer alan kütüphanelerin yüklenmesi için *install.packages("kütüphane ismi")* kütüphaneyi çalıştırmak için ise *library(kütüphane ismi)* kullanmamız gerektedir. *Lütfen örnekte olan kütüphaneleri bu şekilde indirip hazır hale getiriniz*.

Yukarıdaki veri setinde öğrencilerin boylarını histogram grafiği yardımı ile 6 aralık olacak şekilde görselleştireceğiz.

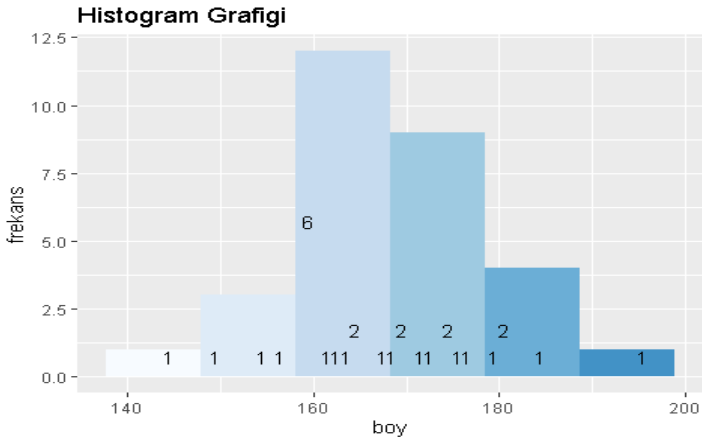
```
library(RColorBrewer)
#renk kütüphanemizi tanımlıyoruz.
renk<- brewer.pal(15,"Blues") #kullanacağımızı mavi renki atarız.

hist(kiloboy$boy, breaks = 6, #breaks ile grup belirlenir
    col=renk,
    labels=TRUE,
    xlab="Boy cm",ylab="Kisi", # x ve y etiketlerini tanımlarız.
    main="Sınıfın Boy Histogramı")# baslık oluştururuz.
```



Grafik 22: Sınıfın Boy Histogramı

Histogram oluşturmanın bir diğer yolu ise *ggplot* (Wickham ve Wickham, 2007) kütüphanesinden yararlanmaktır. Bu kütüphane *tidyverse* (Wickham, 2017) kütüphanesi içinde yüklenmiştir. *ggplot* fonksiyonu kullanırken ilk tanımlamamız gereken veri seti bilgisidir. Veri seti ve veri kolonu aşağıdaki şekilde tanımlanır ve histogram *geom_histogram()* ile çizdirilir. *ggplot* ile yaptığımız histogramın aşağıdaki gibi belirli alanlarını değiştirerek, grafiğin daha güzel görünmesi sağlanabilir. Örneğin *geom_histogram* içinde *color* ile kontür atabileceğimiz gibi, *fill* ile de çubuklarımızın iç rengini değiştirebiliriz. Başlık eklemek istersek *ggtitle* komutunu, x ve y ekseni bilgileri için *labs* komutunu kullanabiliriz. Verimizin ayrıca değerlerini eklemek istersek *geom_text* ile bunu kolayca yapabiliriz.

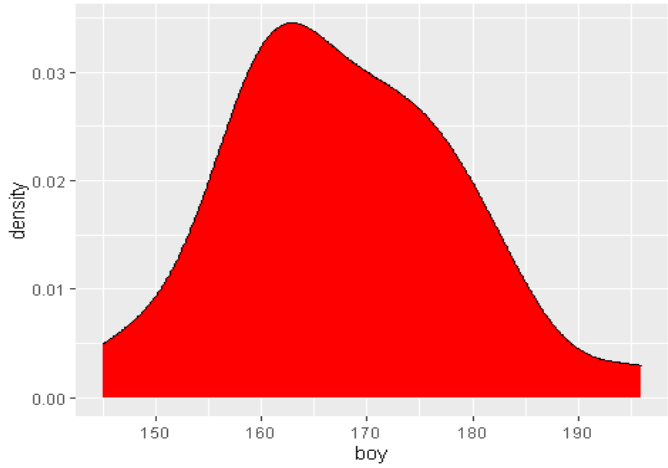
```
library(tidyverse)
ggplot(kiloboy,aes(boy))+
 geom_histogram(fill= renk[1:6], bins = 6)+
 geom_text(stat = "count", aes(label = ..count.., hjust = 1, vjust = 1))+
 ggtitle("Histogram Grafiği")+
 labs(y="frekans", x="boy")
```



Grafik 23: Sınıfın Boy Histogramı

Eğer histogram grafiğimizin yoğunluğunu görmek istersek, aşağıdaki komutlar ile yoğunluk grafiği de çizdirilebilir.

```
ggplot(kiloboy, aes(boy)) +
 geom_density(fill ="Red") #yoğunluk grafiği çizimi için kullanılır.
```

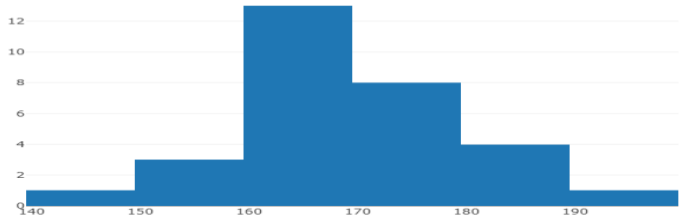


Grafik 24: Sınıfın Boy Histogramı (Yoğunluk)

```
# fill ="Red" ile çizilen yoğunluk çizgisinin altını doldurmuş olduk.
```

Eğer html için interaktif grafikler hazırlamak istersek bunu *plotly* (Sievert, 2020) paketi ile yapabiliriz. Aşağıda aynı örneğin interaktif hali gösterilmiştir.

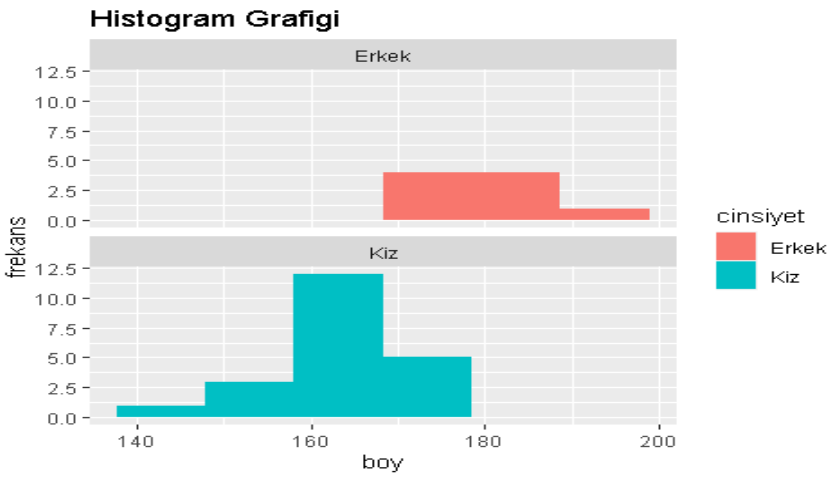
```
library(plotly)
graf <- plot_ly(x= kiloboy$boy, type="histogram")
graf
```



Grafik 25: Sınıfın Boy Histogramı

Histogram grafiğimizi bir kırılıma göre görmek istersek *geom_histogram* içinde *aes* fonksiyonu içinde *fill* değerine istediğimiz kırılımı belirtmemiz yeterlidir. Aşağıdaki örnekte sınıftaki öğrencilerin boylarının cinsiyet kırılımı ile görselleştirilmesi yapılmıştır. *facet_wrap* fonksiyonu ile çizdirilecek grafiğin kolon ve sütün sayısı verilebilmektedir. Aşağıdaki örnekte grafikler 1 kolonda gösterilecektir.

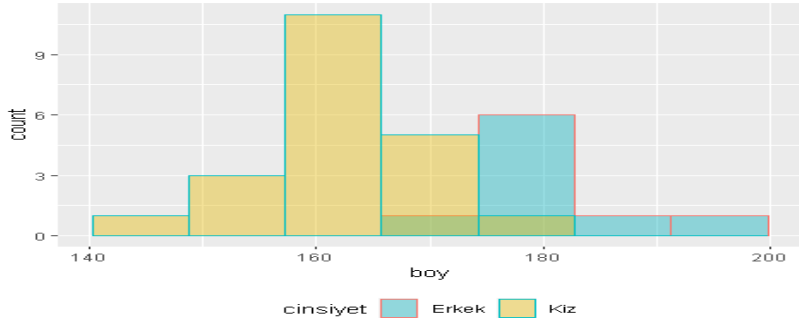
```
ggplot(kiloboy, aes(boy)) +
  geom_histogram(aes(fill= cinsiyet), bins = 6) +
  ggtitle("Histogram Grafiği") +
  labs(y="frekans", x="boy") + facet_wrap( ~ cinsiyet, ncol=1)
```



Grafik 26: Kırılımlı Histogram

Kırılımları aynı grafikte görmek istersek *aes* içindeki *poisition* değerini "identity" yapmamız yeterli olacaktır.

```
ggplot(kiloboy, aes(x = boy)) +
  geom_histogram(aes(color = cinsiyet, fill = cinsiyet),
                 position = "identity", bins = 7, alpha = 0.4) +
  #alpha değeri ile saydamlık verdik.
  scale_fill_manual(values = c("#00AFBB", "#E7B800")) +
  #çubuklarımızın rengini atadık.
  theme(legend.position="bottom") #Açıklama kısmının yerini
değişebiliriz.
```



Grafik 27: İki Histogramın Tek Grafikte Gösterimi

4.2. Çubuk Grafikler (Bar Chart)

Histogram gibi çubuk grafiklerde çubuk şeklindedir. Bu yönü ile birbirlerine çok benzemektedirler. Burada da çubuk uzunluğu temsil ettiği değişkenin değeridir. Değer arttıkça çubuk şeklide orantısal olarak artar. X ekseninde genelde değişkenler yer alır. Y ekseninde ise ölçüm birimleri yer almaktadır. Dolayısı ile x ekseni bir ölçeğe sahip değilken y ekseni bir ölçeğe sahiptir. Birbirinden farklı gruplar için frekans, sayı, yüzde gibi değerleri göstermek, daha da önemlisi gruplar veya kategoriler arasındaki farklılıkları kıyaslamak için çubuk grafikler

kullanılmaktadır. Bu arada değişkenlerin Y ekseninde de gösterildiği durumlar olmaktadır. Kategori sayısının çok olduğu durumlarda bu yöntem tercih edilmektedir.

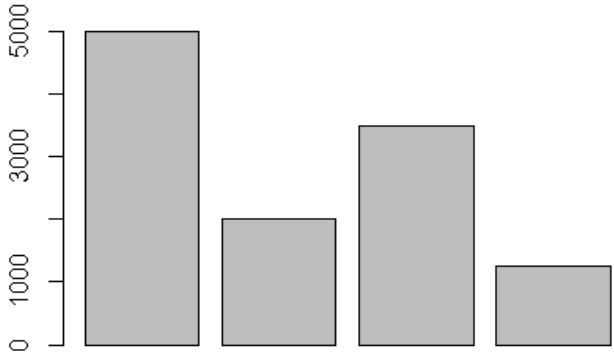
R ortamında en basit şekilde çubuk grafini *barplot()* fonksiyonu ile oluşturmak mümkündür. Aşağıda hem bu fonksiyon ile hem de *ggplot* paketi ile yapılmış örneklere yer verilmiştir. *ggolot* paketi içinde geom_bar ve stat_count fonksiyonları sadece tek bir eksen verisi üzerinden çalışır. X ve Y ekseninde veri setinden iki veri ile çubuk gösterimi yapılmak istenirse, o zaman *geom_col()* fonksiyonun çağırılması gerekmektedir.

İlk önce veri setimizi oluşturalım:

```
sehirler <- c("istanbul", "izmir", "ankara", "bursa")
okul_sayisi <- c(5000, 2000, 3500, 1250)
veri <- data.frame(x=sehirler, y=okul_sayisi)
```

Verimizi oluşturduğumuza göre şimdi çubuk grafiklerimiz oluşturabiliriz. Aşağıdaki örnekte görüldüğü üzere grafiğimiz veri setindeki sıra ile oluşmaktadır.

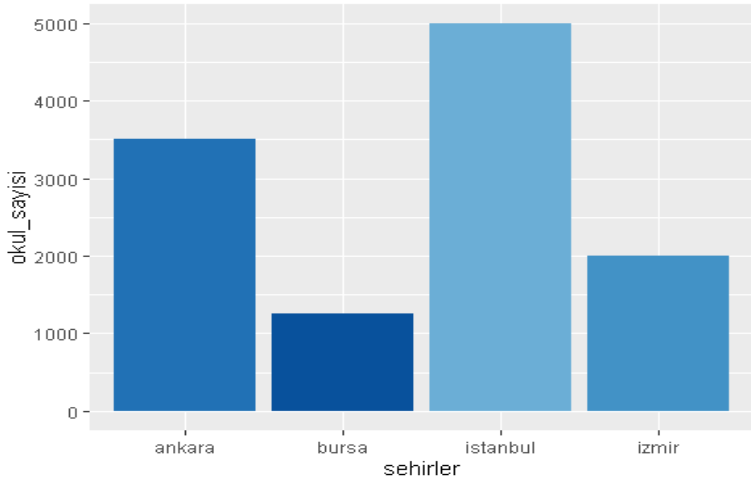
```
barplot(okul_sayisi)
```



Grafik 28: Basit Çubuk Grafik Gösterimi

Aynı grafiği *ggplot* ile çizmek istediğimizde *geom_bar* komutunu kullanmamız gerekmektedir. Aşağıdaki kod çalıştırıldığında çubuk grafiğimiz elde edilmiş olacaktır. Burada dikkat edilmesi gereken oluşan grafik x ekseni sıralaması baz alınarak yapılmaktadır.

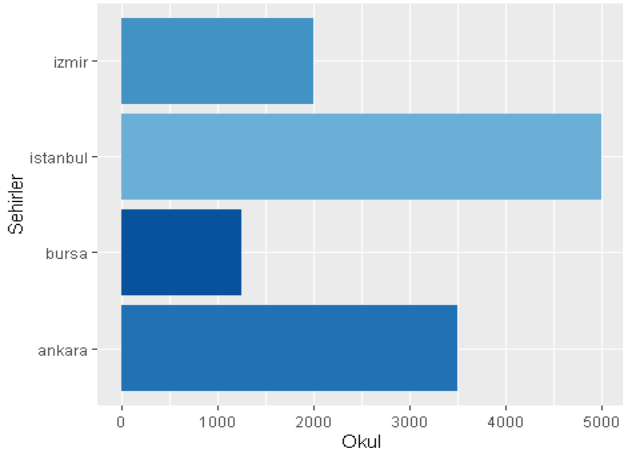
```
grafik<- ggplot(data=veri,aes(x=sehirler,y=okul_sayisi)) +
geom_bar(stat="identity",fill= renk[5:8])
grafik
```



Grafik 29: Ggplot ile Çubuk Grafik Oluşturma

corrd_flip fonksiyonu ile x ve y eksenindeki verilerin yeri değiştirilebilmektedir.

```
Sehirler <- c("istanbul", "izmir", "ankara", "bursa")
Okul <- c(5000, 2000, 3500, 1250)
veri <- data.frame(x=Sehirler, y=Okul)
grafik<- ggplot(data=veri,aes(x=Sehirler,y=Okul)) +
  geom_bar(stat="identity",fill=renk[5:8]) +
  coord_flip()
grafik
```



Grafik 30: Ggplot ile Çubuk Grafik Oluşturma

Yukarıdaki örneklerin yanında konunun daha da iyi anlaşılması için çok bilindik veri seti olan titanik ile birkaç veri görselleştirme çalışması yapılmıştır. *Titanic* veri seti 1912 yılındaki kazada gemi içindeki 1309 yolcuya ait bilgileri içerir. *survived* kolonu kazada kurtulan ve ölenleri, *sex* kolonu cinsiyeti, *age* kolonu yaşı, *passengerClass* kolonu ise oturdukları mevkii göstermektedir.

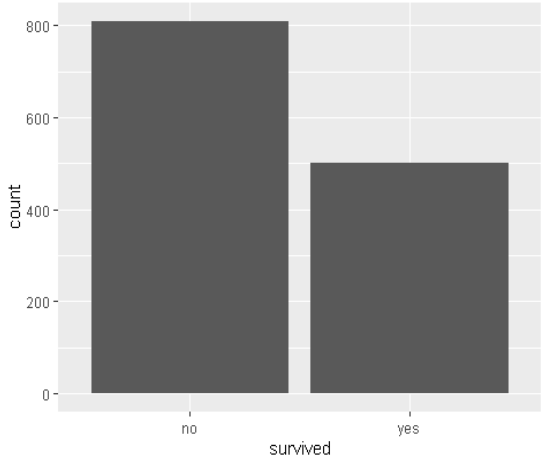
```
library(tidyverse)
library(carData)
Titanic<- as.data.frame(TitanicSurvival)
glimpse(Titanic)
## Rows: 1,309
## Columns: 4
## $ survived <fct> yes, yes, no, no, no, yes, yes, no, yes, no, no, yes,
y~
## $ sex <fct> female, male, female, male, female, male, female,
```

```
male,~
## $ age <dbl> 29.0000, 0.9167, 2.0000, 30.0000, 25.0000, 48.0000,
63.~
## $ passengerClass <fct> 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st, 1st,
1st, ~
```

glimpse komutu ile verimizi inceleyebiliriz. Görüldüğü üzere veri setindeki verilerin hepsi kategori formatındadır. Bu gruplandırma yaparken işimize gelecektir.

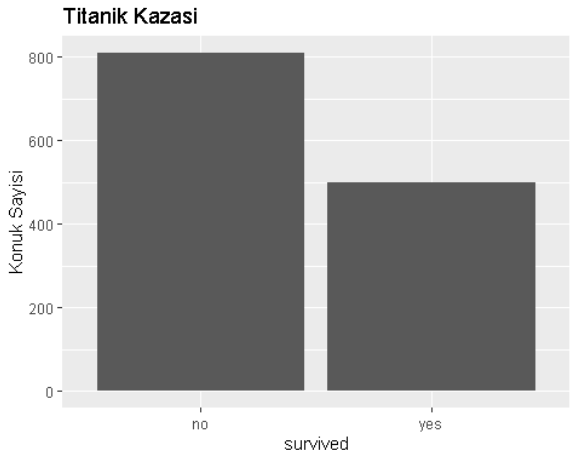
Aşağıdaki kod ile basit bir çubuk grafik çizdirilebilir. Burada kurtulma durumu ile ilgili veri görselleştirilmiştir.

```
ggplot(Titanic, aes(x=survived)) + geom_bar()
```



Grafik 31: Titanik Veri Setinde Kazadan Kurtulanların Çubuk Grafik ile Görselleştirilmesi

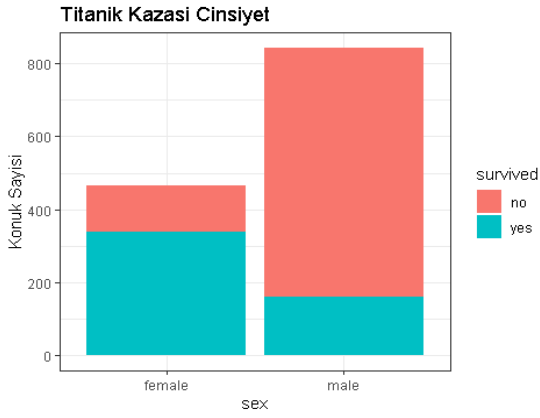
```
ggplot(Titanic, aes(x=survived)) +
  geom_bar()+ #kurtulanların sayısının gösterimi için çubuk grafik seçimi
  labs(y = "Konuk Sayısı", title = "Titanik Kazası")
#labs fonksiyonu ile eksen isimlendirilmesi ve başlık isimlendirilmesi yapılmaktadır.
```



Grafik 32: Grafiğe Eksen Adı ve Başlık Eklenmesi

Veri setimizdeki kurtulanları cinsiyete göre görselleştirmek istersek, x ekseninin *sex* ve kırılımı *fill* survived olarak değiştirebiliriz.

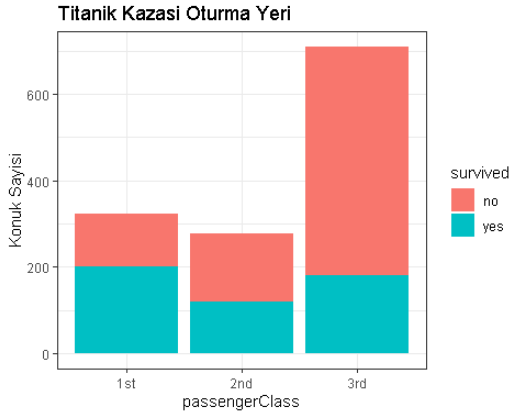
```
ggplot(Titanic, aes(x=sex, fill=survived)) +
  geom_bar() + theme_bw() +
  labs(y = "Konuk Sayısı", title = "Titanik Kazası Cinsiyet")
```



Grafik 33: Cinsiyet Kırılımına Göre Kurtulanların Çubuk Grafik ile Görselleştirilmesi

Oturma yerine göre kurtulanların çubuk grafiği ile görselleştirmesi aşağıdaki kod yardımı ile yapılabilmektedir.

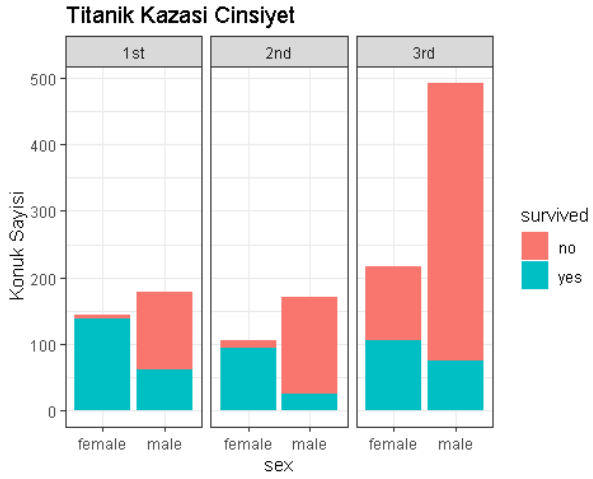
```
ggplot(Titanic, aes(x=passengerClass, fill=survived)) +
  geom_bar() + theme_bw() +
  labs(y = "Konuk Sayısı", title = "Titanik Kazası Oturma Yeri")
```



Grafik 34: Alt Kırılımlı Çubuk Grafik Oluşturulması

Cinsiyete göre kurtulanların oturma yerlerine göre ayrı ayrı görselleştirmesi için facet_wrap() fonksiyonu kullanılmaktadır.

```
ggplot(Titanic, aes(x=sex, fill=survived)) +
 geom_bar() + theme_bw() +
 labs(y = "Konuk Sayısı", title = "Titanik Kazası Cinsiyet") +
 facet_wrap(~passengerClass)
```



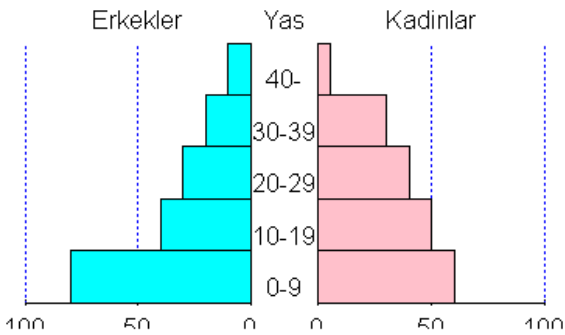
Grafik 34: Facet_Wrap Fonksiyonu Kullanımı

4.3. Nüfus Piramit Grafikleri (Population Pyramid)

Bir toplulukta nüfusun cinsiyete ve yaş aralıkları ile görselleştirilmesi sonucunda nüfus piramitleri oluşmaktadır. Peş peşe temelde 2 histogramdan oluşur. X ekseninde nüfus sayıları, y eksenin tüm yaş grupları bulunmaktadır. Nüfus piramitleri ile güncel nüfusun dağılımı, gelecekteki nüfusun yaşa ve cinsiyete göre durumu ve gelişimi hakkında çıkarımlarda bulunulmaktadır.

Aşağıda basit bir veri seti üzerinde bir örneğe yer verilmiştir. *pyramid* (Nakazawa ve Nakazawa, 2019) paketi grafiği çizmemizde işimizi oldukça kolaylaştırmaktadır.

```
library(pyramid)
Erkek <- c(80,40,30,20,10)
names(Erkek) <- c('0-9','10-19','20-29','30-39','40-')
Kadin <- c(60,50,40,30,5)
pyramids(Left=Erkek, Llab="Erkekler", Right=Kadin,
Rlab="Kadınlar",
Laxis=c(0,50,100), main="Nüfus Piramit Örneği", Clab ="Yaş")
```



Grafik 35: Nüfus Piramit Örneği

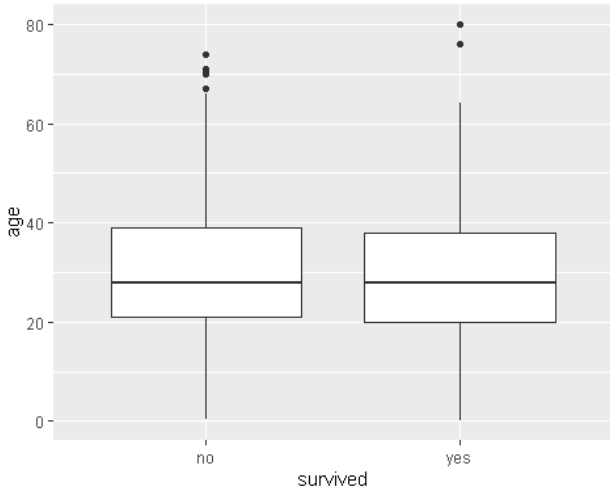
4.4. Kutu- Bıyık Grafikleri (Box Plot)

Kutu-Bıyık grafikleri, elimizdeki verilerin çeyreklik şekillerde (dörtte birinin) dağılımını göstermenin alternatif bir şeklidir. Histogram grafiklerinin bir kutu ile aslında tasviridir. Kutunun üstünde ve altında olan çizgiler *bıyık* olarak adlandırılmakta, aykırı ve uç değerleri temsil

etmektedir. Verinin ortalaması, yüzdelik dilimleri, uç ve aykırı değerlerin tespit edilmesinde sıkça kullanılan grafik, ayrıca az yer kapladığı için birçok araştırmada görselleştirme olarak tercih edilmektedir.

Çok bilindik *Titanic* veri seti ile yapılan kutu grafikleri aşağıda sunulmuştur. Kutu-bıyık grafiği çizilmesi için *ggplot* kütüphanesi içindeki *geom_boxplot* fonksiyonu kullanılmaktadır.

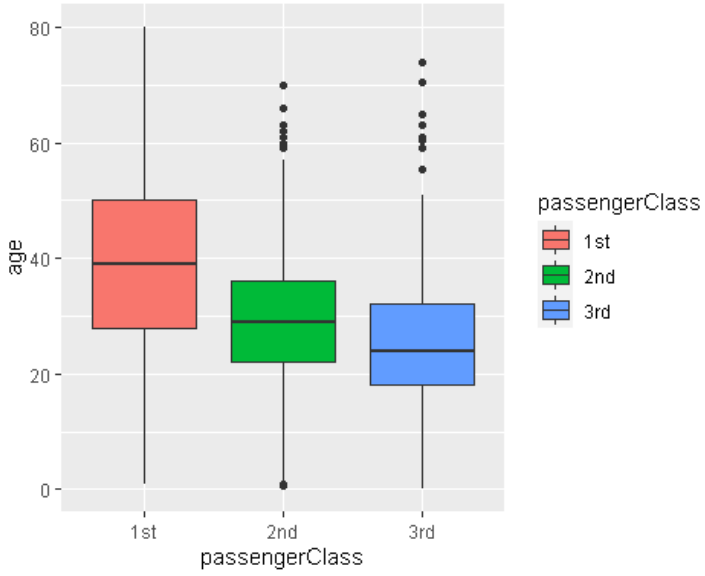
```
ggplot(Titanic, aes(x=survived, y=age))+
 geom_boxplot()
```



Grafik 36: Kutu Grafik Örneği

Aşağıda aynı veri setinde yolcuların yeri ile yaş arasındaki ilişki tespit edilmiştir. Birinci sınıfta seyahat edenlerin yaş ortalamalarının diğer sınıflarda seyahat edenlerden daha yüksek olduğu görülmektedir.

```
ggplot(Titanic, aes(x=passengerClass, y=age, fill=passengerClass)) +
  geom_boxplot()
```



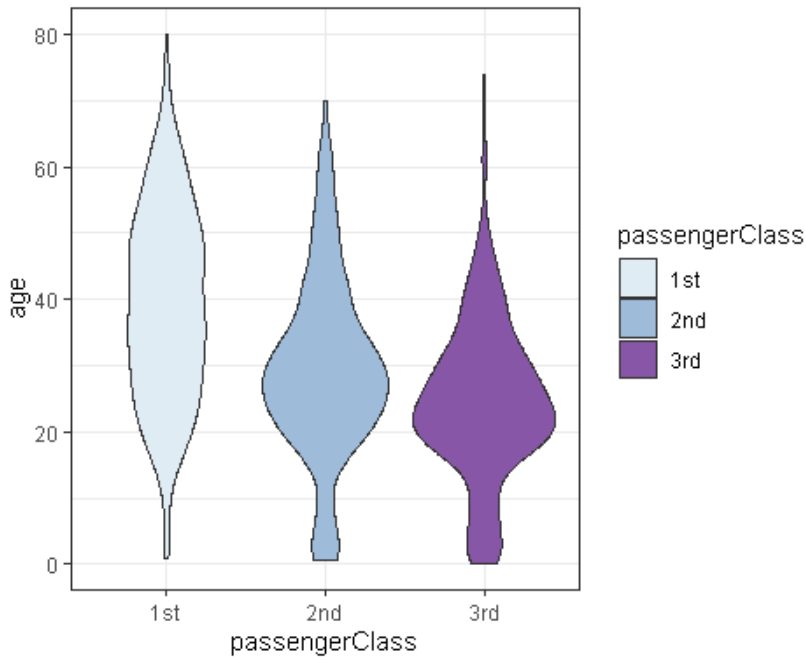
Grafik 37: Kutu Grafik Örneği

4.5. Viyolin (Viyolonsel) Grafikleri (Violin Plot)

Viyolin (Viyolonsel-Keman) grafikleri temelde kutu grafikler ile histogram grafiklerinin bir birleşimi olarak karşımıza çıkar. Verilerin dağılımı ve olasılık yoğunluğu hakkında bizlere fikir vermektedir. Viyolin grafik, kutu grafiklerinden farklı olarak bizlere dağılımın hangi şekilde olduğu konusunda bilgi vermektedir. Ggplot kütüphanesi

içindeki geom_violin() komutu ile grafiğimiz kolayca çizilebilemektedir.

```
ggplot(Titanic, aes(x=passengerClass, y=age, fill=passengerClass)) +
  theme_bw() +
  geom_violin() + scale_fill_brewer(palette="BuPu")
```



Grafik 38: Viyolin Grafik Örneği

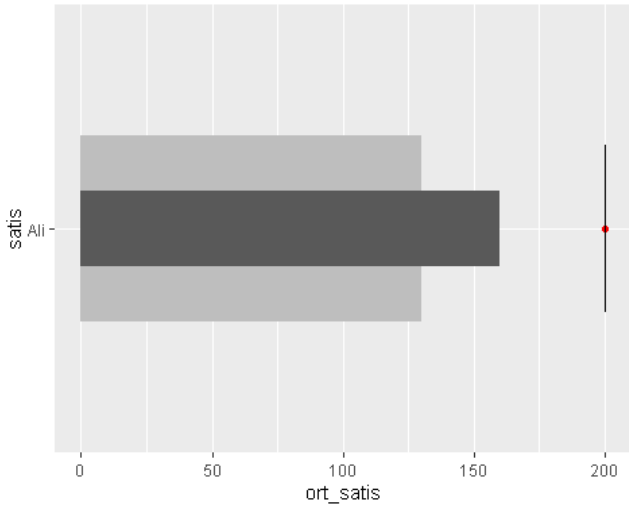
4.6. Hedef Grafiği (Bullet Graph)

Stephen Few tarafında geliştirilen hedef grafiği, iç içe geçmiş çubuk grafikler olarak karşımıza çıkmaktadır (Few, 2006). Genelde KPI (Kilit Performans Göstergeleri) ve performans verilerinin

görselleştirilmesinde, hedefe ve amaçlanana ulaşılıp ulaşılmadığının tespitinde kullanılmaktadır. Birçok ölçütün bir arada görselleştirilmesine olanak tanıyan grafik, kullanılan ölçütlerinin karşılaştırılmasında da kullanılmaktadır (Düşük, orta, zayıf vb.).

Aşağıda Ali isimli satışçının gerçekleşen satış miktarı, firma satış ortalaması ve satış hedefi görselleştirilmiştir.

```
library(ggplot2)
veri <- data.frame(satis= c("Ali"),
                   ort_satis= 130,
                   gerceklesen= 160,
                   hedef= 200)
p <- ggplot(veri, aes(satis, ort_satis) )
p <- p + geom_col(fill="grey", width=0.5)
p <- p + geom_col(aes(satis, gerceklesen), width=0.2)
p <- p + geom_point(aes(satis, hedef), colour="red")
p <- p + geom_errorbar(aes(y = hedef,x = satis, ymin = hedef,ymax
= hedef), width = .45)
p <- p + coord_flip()
p
```



Grafik 39: Hedef Grafiği Örneği

4.7. Mum Grafik (Candle stick Chart)

Bir ürünün belli bir zaman periyodu içindeki ticari faaliyetlerinin görselleştirilmesinde mum grafik kullanılmaktadır. Hisse senedi hareketlerinde veya borsa da işlem gören bir ürünün hareketlerinin görselleştirilmesinde sıklıkla kullanılır. Grafik; ürünün belli bir zamandaki açılış fiyatı, kapanmış fiyatı, en yüksek ve en düşük satış düzeyleri ve hacmi gibi verileri içinde saklar. Mum grafiklerini çizmenin en kolay yolu plotly kütüphanesini kullanmak ve *type=”candlestick”* fonkisyonunu çağırmaktır.

Aşağıda *Apple* firmasının hisse senedi hareketi görselleştirilmiştir.

```
library(plotly)
library(quantmod)
```

```
#quantmod kütüphanesi ile hisse senedi bilgileri çekilir.
getSymbols("AAPL",src='yahoo')
## [1] "AAPL"
df <- data.frame(Date=index(AAPL),coredata(AAPL))
df <- tail(df, 30)
mum <- df %>% plot_ly(x = ~Date, type="candlestick",
        open = ~AAPL.Open, close = ~AAPL.Close,
        high = ~AAPL.High, low = ~AAPL.Low)
mum <- mum %>% layout(title = "Apple Firmasına Ait Hisse Senedi
Bilgileri",
       xaxis = list(rangeslider = list(visible = F)))

mum
```

Grafik 40: Mum Grafiği Örneği

4.8. Zaman Çizelgesi- Gantt Grafiği (Gantt Chart)

Bir projenin ne kadar süreceğinin görselleştirilmesinde, projede çalışanların yapacakları işi hangi süre için yapacağının takibinde, bir etkinliğin hangi süreler arasında gerçekleşmesi gerektiği gibi konularda *Gantt* grafikleri kullanılmaktadır. Grafik, yapılacak iş veya etkinliğin, etkinlikte kullanacak personelin çakışmasını önlemekte kullanışlı bir araçtır.

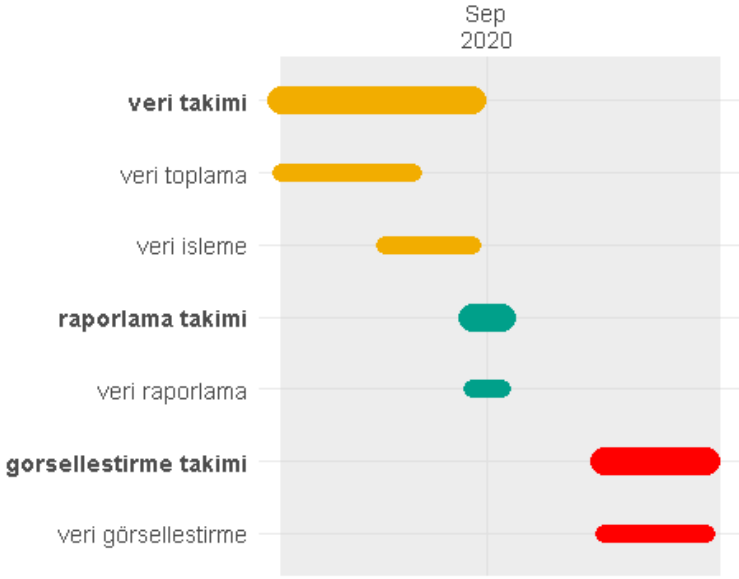
Grafikte, faaliyetler için satırlar kullanılır ve zaman ölçeği olarak sütunlar genelde kullanılmaktadır. Etkinliğin süresi çubukların boyları ile görselleştirilmektedir. R ile bu grafik türünü çizmek için aşağıdaki paketin yüklenmesi gerekmektedir.

```
# install.packages("remotes")
#remotes::install_github("giocomai/ganttrify")
```

ganttrify (GitHub, 2021) paketi ile oluşturulan grafik aşağıda sunulmuştur. Paketin çalışması için yapılan aktivitenin ve başlangıç-bitiş tarihlerinin girilmesi gerekmektedir.

```
library("ganttrify")

wp<- c("veri takımı", "veri takımı", "raporlama takımı",
       "görselleştirme takımı")
activity<- c("veri toplama", "veri işleme", "veri raporlama",
             "veri görselleştirme")
start_date<- c("2020-09-01", "2020-09-08","2020-09-14","2020-09-
23")
```

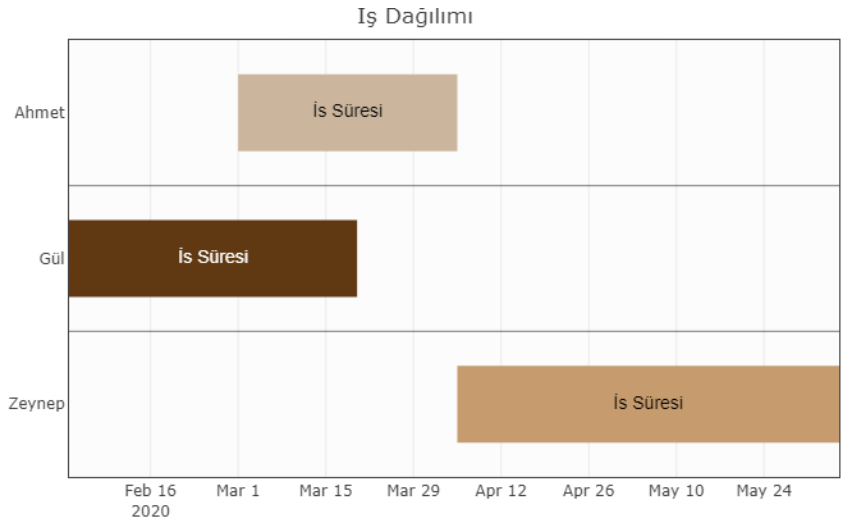
```
end_date <- c("2020-09-10","2020-09-14","2020-09-16","2020-09-30")

takvim<- data.frame(wp, activity, start_date, end_date)

ganttrify(project = takvim,
        #spots = ganttrify::test_spots_date_day,
        by_date = TRUE,
        exact_date = TRUE,
        size_text_relative = 1.2,
        month_number_label = FALSE)
```



Grafik 41: Gantt Grafik Örneği

Aşağıda *vistime* (Raabe, 2020) paketi ile diğer bir örneğe ver verilmiştir.

```
library(vistime)
pres <- data.frame(Position = rep(c("İs Süresi"), each = 3),
                   Name = c("Ahmet", "Gül", "Zeynep"),
                   start = c("2020-03-1", "2020-02-03", "2020-04-05"),
                   end = c("2020-04-05", "2020-03-20", "2020-06-05"),
                   color = c('#cbb69d', '#603913', '#c69c6e'),
                   fontcolor = c("black", "white", "black"))

vistime(pres, col.event = "Position", col.group = "Name", title = "İş Dağılımı")
```

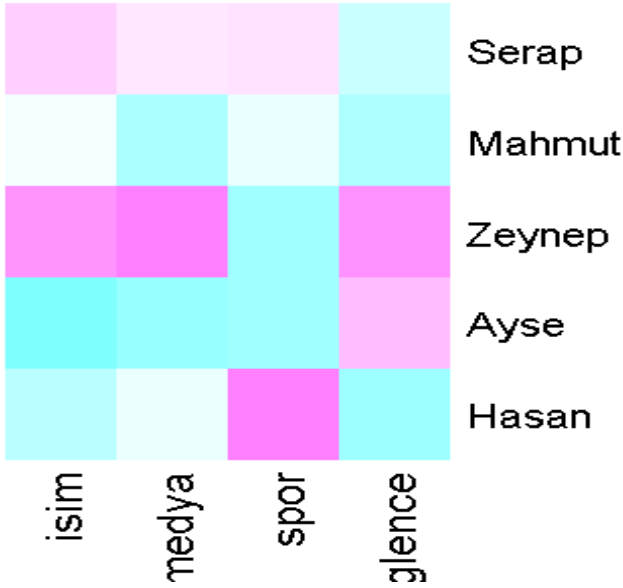


Grafik 42: Gantt Grafik Örneği

4.9. Isı Haritası (Heatmap)

Isı haritaları ile sayıların veya sayı değer aralıklarının renkler ile ifade edilmesi amaçlanmaktadır. Sayıların renklerle ifade edilmesi sonucunda tablo veya sütun özelinde bir örüntünün varlığı veya olası bir korelasyonun varlığı kolayca tespit edilebilmektedir. Excel dahil birçok program ile ısı haritası oluşturulabilir. Aşağıda 5 öğrencinin belli faaliyetlere ayırdıkları saatler görselleştirilmiştir. Bunun için R ile birlikte gelen *stats* kütüphanesi içindeki *heatmap()" fonksiyonu kullanılmıştır.

```
isim<- c("Hasan", "Ayse", "Zeynep", "Mahmut", "Serap")
sosyalmedya <- c(5,1,12,2,7)
spor<- c(12,3,3,6,8)
eglence<- c(1,12,15,2,4)
heatmap<- data.frame(isim,sosyalmedya,spor,eglence)
row.names(heatmap) <- heatmap$isim

heatmapmatrix<- data.matrix(heatmap)
heatmap(heatmapmatrix, Rowv=NA, Colv=NA, col = cm.colors(256),
      scale="column", margins=c(5,10))
```



Grafik 43: Isı Haritası Örneği

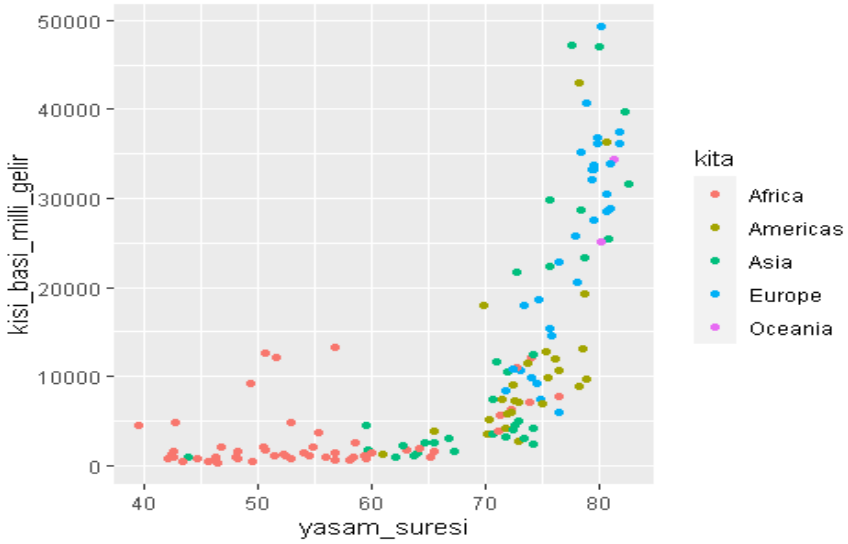
4.10. Saçılım Grafikleri (Scatter Plot)

İki değişken ait sayısal verilerin bir nokta ile temsil edildiği en ilkel grafik türlerinden birisi olan saçılım grafiği, verilerin kendi aralarındaki ilişkiyi görselleştirir. Grafik üzerindeki her bir nokta bir gözlemi temsil etmektedir.

Bu ve bundan sonrasında birçok görselleştirmede *gapminder* (Bryan, 2015) kütüphanesi içindeki *gapminder* veri seti kullanılacaktır. Bu veri setinden ülkeler, ülkelerin kıtaları, yaşam süresi, kişi başına düşen milli gelir ve nüfus bilgileri vardır.

Aşağıda, veri seti yüklenmiş, 2007 yılındaki veriler çekilmiş ve yaşam süresi ve milli gelir, kıtalarına göre renklendirilmiştir.

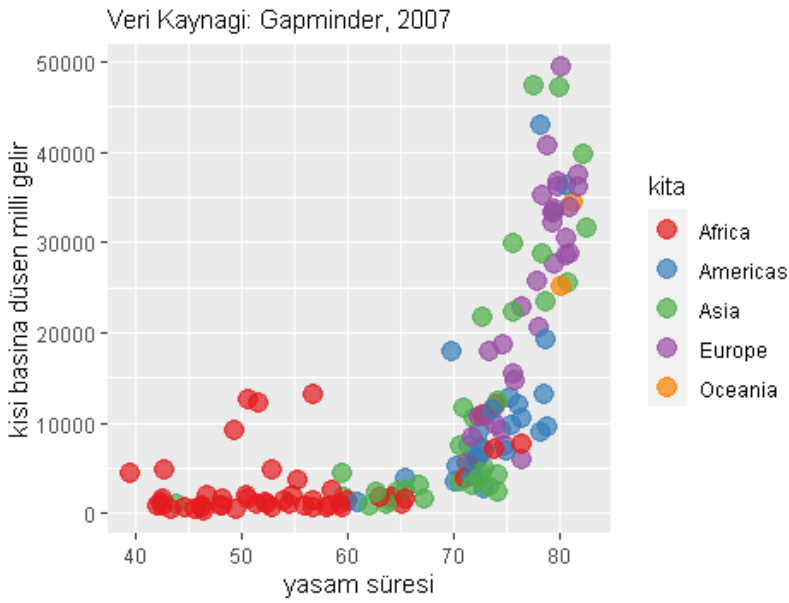
```
library(gapminder)
gapminder<- gapminder
names(gapminder)<- c("ülke", "kita", "yil","yasam_suresi","nufus",
                     "kisi_basi_milli_gelir" )
gapminder_2007 <-filter(gapminder, yil == 2007)
ggplot(gapminder_2007,aes(yasam_suresi, kisi_basi_milli_gelir, color
= kita)) +
geom_point()
```



Grafik 44: Saçılım Grafiği Örneği

```
renk<- c("#e41a1c","#377eb8","#4daf4a","#984ea3","#ff7f00")
gapminder_2007 <-filter(gapminder, yil == 2007)
ggplot(gapminder_2007,aes(yasam_suresi, kisi_basi_milli_gelir, color
```

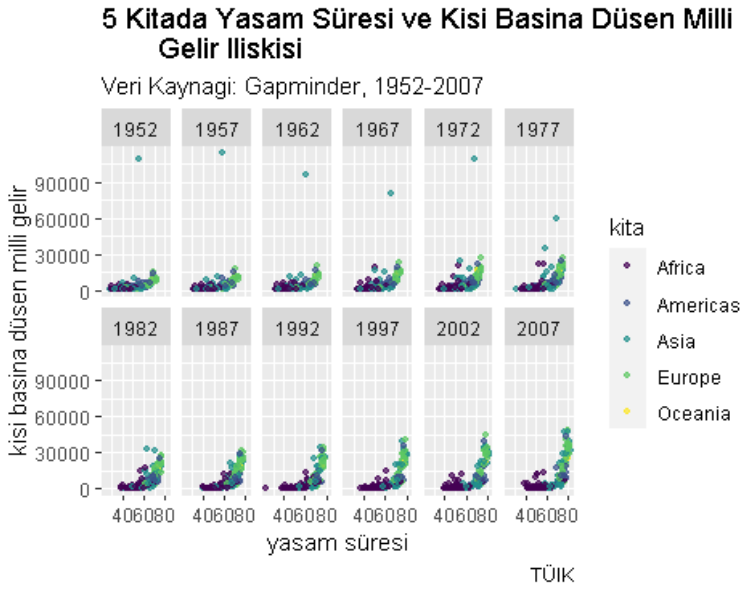
```
= kita))+
  geom_point(size = 4, alpha = 0.7) +
labs(x="yaşam süresi",
     y ="kişi başına düşen milli gelir",
     title = "5 Kıtada Yaşam Süresi ve Kişi Başına Düşen Milli Gelir
İlişkisi",
     subtitle = "Veri Kaynağı: Gapminder, 2007") +
scale_color_manual(values = renk)
```



Grafik 45: Saçılım Grafiği Örneği

Aynı örnek üzerinden 1952-2007 yılları arasında (facet_wrap(~yil, ncol = 6)) kişi başına düşen milli gelirin ve yaşam süresinin ilişkisi kıta renklendirilmesi ile sunulmuştur.

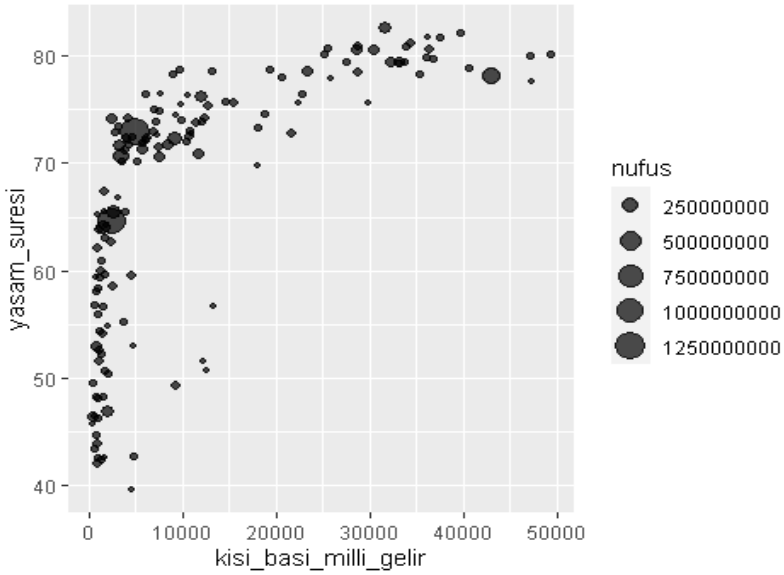
```
library(viridis)
ggplot(gapminder, aes(yasam_suresi, kisi_basi_milli_gelir, color =
kita))+
 geom_point(size = 1, alpha = 0.7) +
 labs(x="yaşam süresi",
     y ="kişi başına düşen milli gelir",
     title = "5 Kıtada Yaşam Süresi ve Kişi Başına Düşen Milli
     Gelir İlişkisi",
     subtitle = "Veri Kaynağı: Gapminder, 1952-2007",
     caption = "TÜİK")+
 scale_color_viridis(discrete=TRUE) +
 #viridis kütüphanesi ile renklendirilmiştir.
 facet_wrap(~yil, ncol = 6)
```



Grafik 46: Saçılım Grafiği Örneği

Noktaların renkleri ve büyüklükleri kırılımlara göre değişebilir. Aşağıda *size* komutu ile noktanın büyüklüğü nüfusa göre şekillenmiştir.

```
ggplot(gapminder_2007, aes(x=kisi_basi_milli_gelir, y=yasam_suresi,
size = nufus)) +
  geom_point(alpha=0.7)
```



Grafik 47: Saçılım Grafiği Örneği

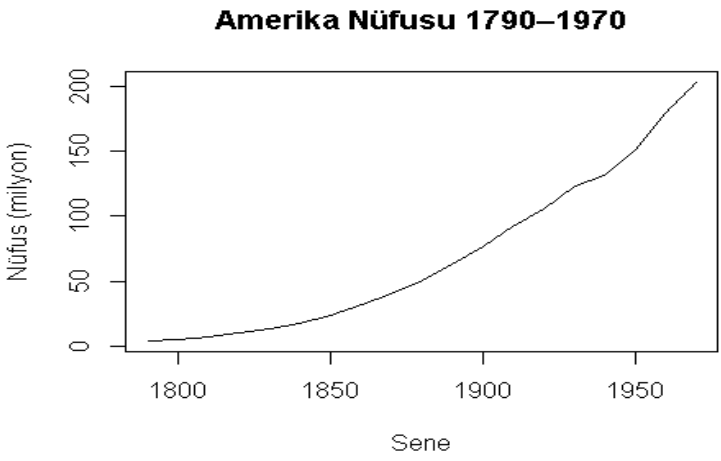
4.11. Çizgi Grafikleri (Line Plot)

Çizgi grafik, genellikle bir değişkenin zaman içindeki gelişimini ve sayısal değişimini gösteren bir grafiktir. Büyük veri setleri içinde bir değişkenin zaman içinde nasıl evrildiğini görsel olarak anlamamıza ve çıkarım yapmamıza olanak veren çizgi grafikleri sayesinde birçok değişkenin zaman içersindeki değişimlerini ve kendi aralarındaki

ilişkilerini karşılatırabilmekteyiz. Çizgi grafiklerinde zaman genellikle x ekseninde yer almakta ve zamana göre değişkenin aldığı değer y ekseninde bulunmakta ve noktalanmaktadır. Sonradan bu noktalar birleştirilerek çizgi grafikleri oluşturulur.

Bu grafik türünü görselleştirmek için *uspop* isimli veri seti kullanılmıştır. Amerika'nın 1790-1970 arası nüfus verileri veri seti içinde yer almaktadır. R Studio içindeki *plot()* fonksiyonu ile basit çizgi grafiği çizilebilir.

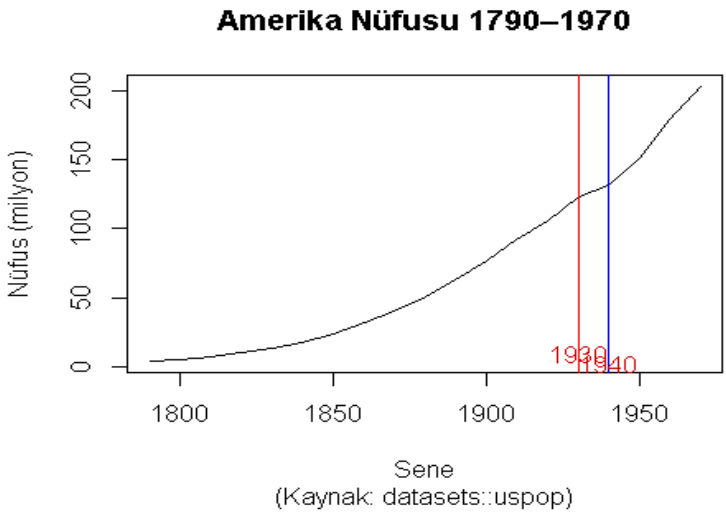
```
nufus<- uspop
nufus
## Time Series:
## Start = 1790
## End = 1970
## Frequency = 0.1
## [1]   3.93   5.31   7.24   9.64  12.90  17.10  23.20  31.40  39.80
50.20
## [11]  62.90  76.00  92.00 105.70 122.80 131.70 151.30 179.30
203.20
nufus<- data.frame(nufus)
plot(nufus,
   main = "Amerika Nüfusu 1790–1970",
   xlab = "Sene",
   ylab = "Nüfus (milyon)")
```



Grafik 48: Çizgi Grafiği Örneği

Çizdiğimiz temel grafiği dikine çizgiler çizerek, renklendirerek ve başlıklarını düzenleyerek daha da güzelleştirme imkânımız bulunmaktadır.

```
nufus %>%
  plot(
    main = "Amerika Nüfusu 1790–1970",
    sub  = "(Kaynak: datasets::uspop)",
    xlab = "Sene",
    ylab = "Nüfus (milyon)",
  )
abline(v = 1930, col = "red")
text(1930, 10, "1930", col = "red3")
abline(v = 1940, col = "blue")
text(1940, 2, "1940", col = "red3")
```



Grafik 49: Çizgi Grafiği Örneği

Çoklu çizgi grafiği oluşturmak için *orange* veri setini kullanacağız. Bu veri setinde portakal ağaçları(tree), ağaçların yaşları/gün (age) ve gövde genişlikleri (circumference) bulunmaktadır. Farklı ağaçların gövde büyüklüklerini aşağıdaki gibi görselleştirebiliriz.

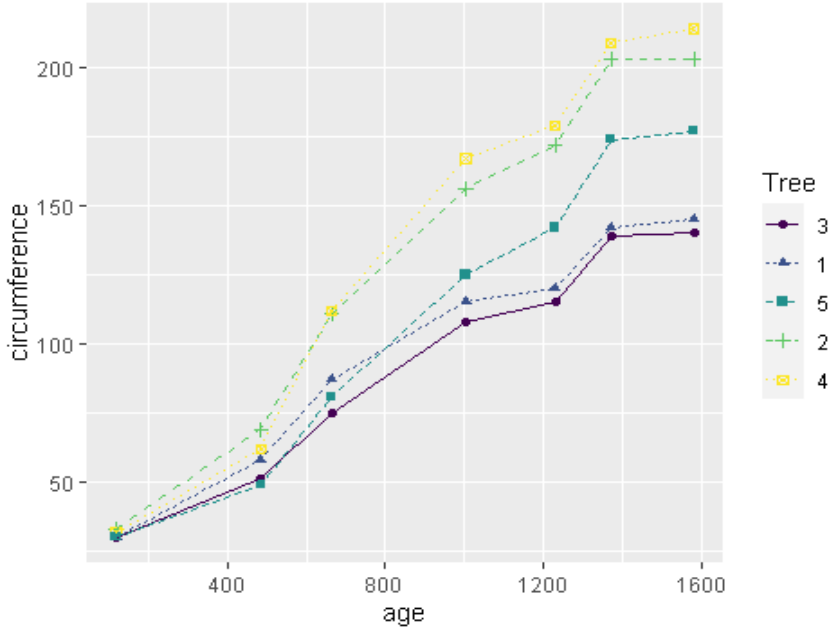
Öncelikle veri setimizi bir inceleyelim:

```
agac<- Orange
agac$Tree<- as.factor(agac$Tree)
head(agac)
##   Tree  age circumference
## 1    1  118            30
## 2    1  484            58
## 3    1  664            87
## 4    1 1004           115
```

```
## 5    1 1231        120
## 6    1 1372        142
```

X eksenini yaş (age) ve y eksenini gövde genişliği (circumference) olarak tanımlayıp, çizgi rengini ve şeklini agaç (tree) olarak atayıp aşağıdaki grafiği çizdirebiliriz.

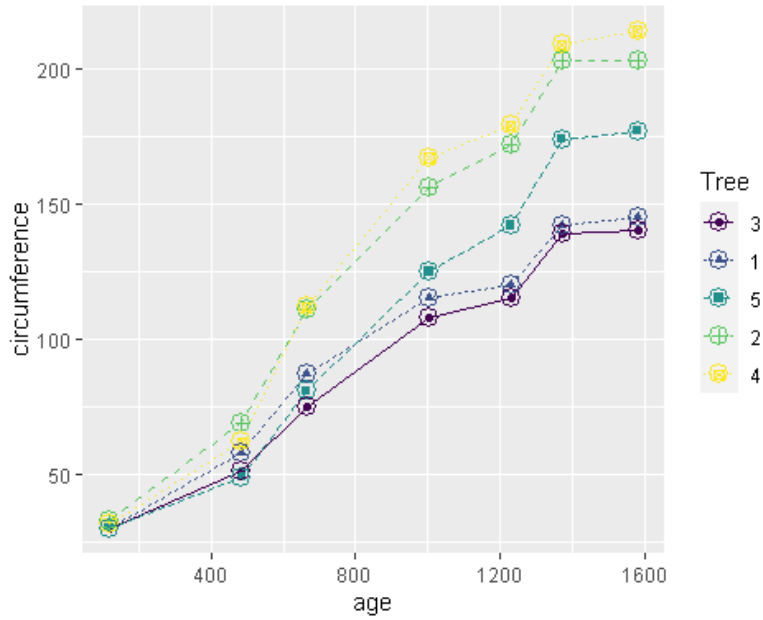
```
ggplot(data=agac, aes(x = age, y = circumference, color =Tree,
linetype = Tree, shape=Tree)) +
  geom_point() +
  geom_line()
```



Grafik 50: Çoklu Çizgi Grafiği Örneği

İstenilirse noktaların rengi veya şekli *geom_point(size = 4, shape = 21)* yazılarak değiştirilebilir.

```
ggplot(data=agac, aes(x = age, y = circumference, color =Tree,
linetype = Tree, shape=Tree)) +
  geom_point() +
  geom_line() +
  geom_point(size = 4, shape = 21)
```

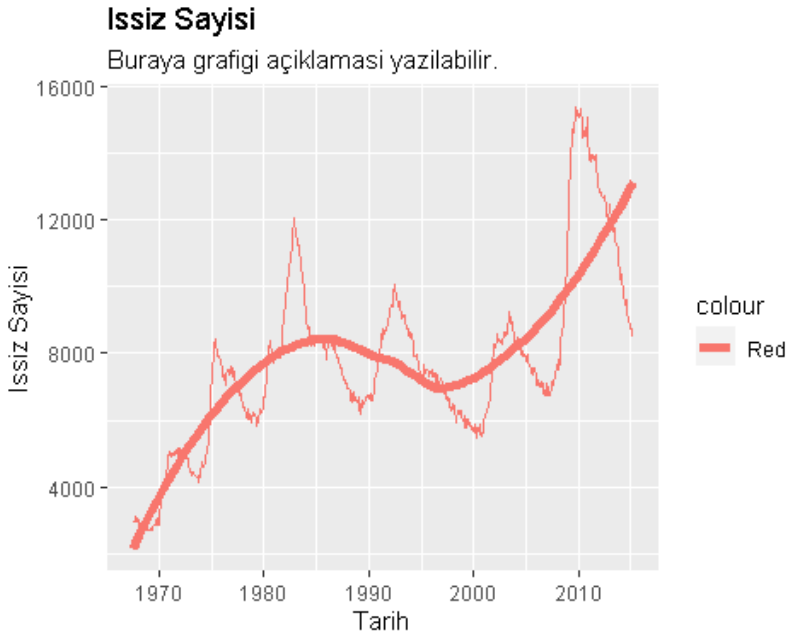


Grafik 51: Çoklu Çizgi Grafiği Örneği

Çizgi grafiklere son bir örnekte *economics* veri seti ile verilebilir. Bu veri seti Amerika için ekonomik göstergeleri içermektedir. Veri setinde tarih (date), tüketim harcaması (pce), nüfus (pop), tasarruf oranı (psavert), issiz sayısı (unemploy) gibi veriler bulunmaktadır. Buradaki veriler ile çizgi grafiği aşağıda örnek olarak oluşturulmuştur.

```
veri<- economics
baslik<- c("tarih", "pce", "nufus", "tasarruf", "issizlik_oranı", "issiz")
```

```
colnames(veri) <- baslik


ggplot(veri, aes(x = tarih, y = issiz, color= "Red")) +
geom_line() +
geom_smooth(lwd=2, se= FALSE) +
 labs(title = 'İssiz Sayısı'
      ,y = 'İssiz Sayısı'
      ,x = 'Tarih'
      ,subtitle = str_c("Buraya grafiği açıklaması yazılabilir.")
      )
```

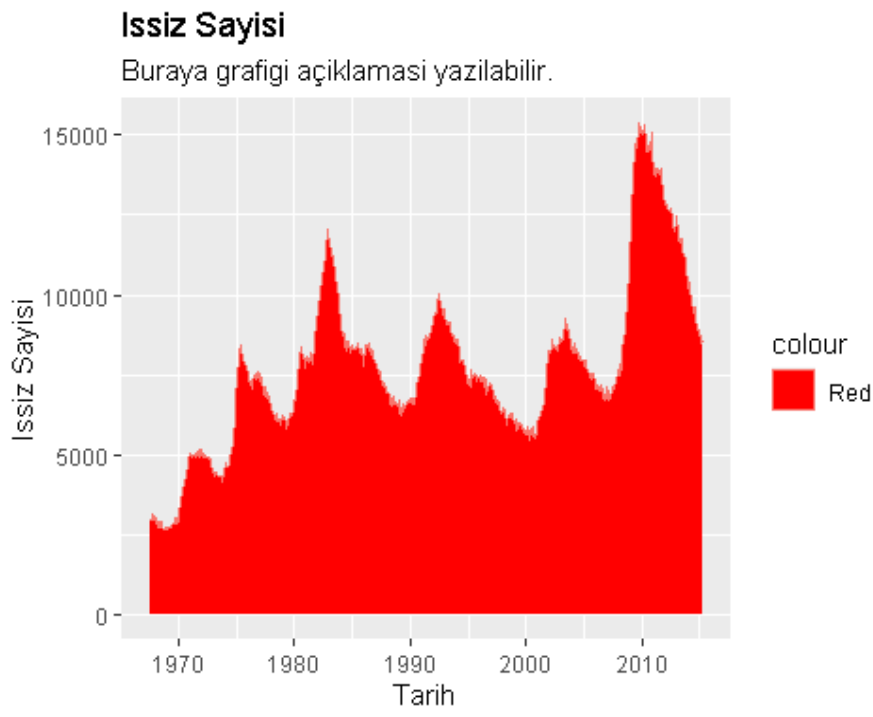


Grafik 52: Çizgi Grafiği Örneği

4.12. Alan Grafikleri (Line Plot)

Alan Grafiği, çizgi grafikleri ile oluşan çizginin altının belli bir renk veya doku ile doldurulduğu bir çizgi grafik türüdür. Bu grafik türü için daha önce kullandığımız *economics* veri seti ile bir alan grafiği oluşturulmuştur. Alanı yaratmak için *geom_area* fonkisyonu kullanılmıştır.

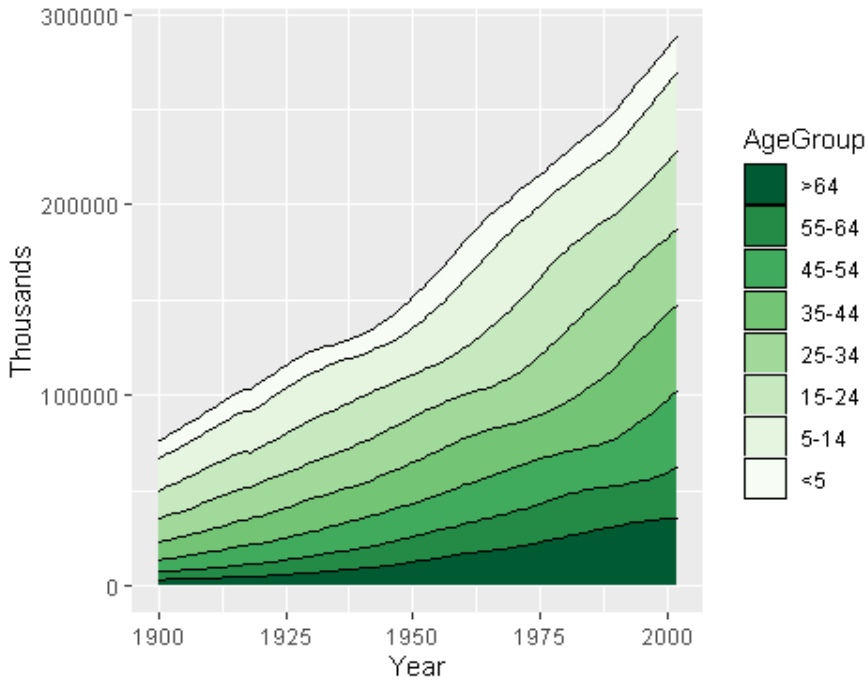
```
veri<- economics
baslik<- c("tarih", "pce", "nufus", "tasarruf", "issizlik_oranı", "issiz")
colnames(veri) <- baslik

ggplot(veri, aes(x = tarih, y = issiz, color= "Red")) +
geom_area(fill="Red") +

 labs(title = 'İssiz Sayısı'
      ,y = 'İssiz Sayısı'
      ,x = 'Tarih'
      ,subtitle = str_c("Buraya grafiği açıklaması yazılabilir.")
      )
```



Grafik 53: Alan Grafiği Örneği

Konuyla ilgili bir başka örnek ise *gcookbook* paketi içindeki *uspopage* veri seti ile verilebilir. Amerika Birleşik Devletleri'nde nüfusun yaşa göre dağılım 1900-2002 senelerini kapsayacak şekilde veri setinde tanımlanmıştır. Veri seti içinde 3 adet değişken vardır: Sene (Year), Yaş Aralığı (AgeGroup) ve kişi sayısı/bin (Thousand). X ekseninde yıl, y ekseninde kişi sayısının, yaş grubuna göre kırılımı aşağıdaki şekilde oluşmuştur. Aşağıdaki örnek Journaldev.com sitesinden alınmıştır (Journaldev, 2020)

```
#Kütüphaneleri yükleyelim
library(gcookbook)
library(ggplot2)

#Verimizi okutalım
ggplot(uspopage, aes(x=Year, y=Thousands, fill=AgeGroup))+
 geom_area(color='black', size=0.3, alpha=1)+
 scale_fill_brewer(palette = 'blues',breaks=rev(uspopage$AgeGroup))
```



Grafik 54: Alan Grafiği Örneği

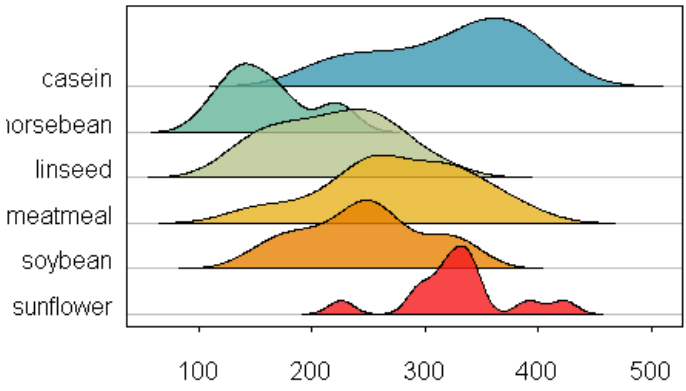
4.13. Ridge Grafiği (Ridge Graph)

Ridge grafiği (bazen Joyplot olarak da isimlendirilir), aynı yatay ölçeğe hizalanmış, birden çok sayısal değişkenin dağılım veya yoğunluk grafiklerinin görselleştirilmesinde kullanılmaktadır.

Görselleştirmede *ggplot* içindeki *geom_density_ridges* fonksiyonu kullanılabileceği gibi *ridgeline* paketi de kullanılabilir. Ayrıca *ggplot2* içindeki *ggridges* paketi ile de görselleştirme yapmak mümkündür.

Aşağıdaki örnekte *chickwts* veri seti kullanılmıştır. 71 gözlemden oluşan veri setinde civcivlerin ağırlıkları ve beslenme türleri vardır. Civciv beslenme türleri ile ağırlıklarının grafik ile görselleştirilmesi aşağıda sunulmuştur.

```
#install.packages("remotes")
#remotes::install_github("R-CoderDotCom/ridgeline@main")

library(ridgeline)
ridgeline(chickwts$weight, chickwts$feed)
```



Grafik 55: Ridge Grafiği Örneği

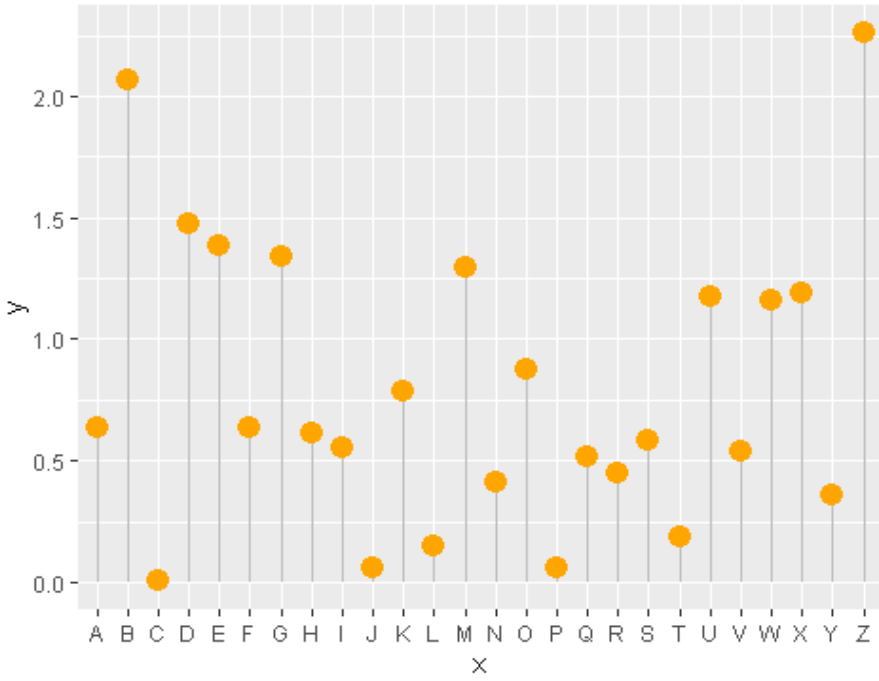
4.14. Lolipop Grafiği (Lollipop Chart)

Çubuk grafiğin farklı bir çeşidi olan lolipop grafiği, farklı grup ve kategorileri karşılaştırmak amacı ile kullanılmaktadır. Çubukların yerini çizgi almakta ve bitiş noktası bir yuvarlak bir simge ile temsil edilmektedir. İlişki ve dağılımların karşılaştırılması amacı ile kullanım uygun değildir.

Aşağıda grafiği oluşturmak için *ggplot* içindeki *geom_segment()* fonksiyonu kullanılmıştır.

```
library(ggplot2)

# Veri oluşturma
data <- data.frame(
```

```
  x=LETTERS[1:26],
  #x harflerden, y numaralardan oluşur.
  y=abs(rnorm(26))
)

# Grafik çizilmesi
ggplot(data, aes(x=x, y=y)) +
  geom_segment( aes(x=x, xend=x, y=0, yend=y), color="grey") +
  geom_point( color="orange", size=4)
```



Grafik 56: Lolipop Grafiği Örneği

4.15. Radar Grafiği (Spider- Radar Chart)

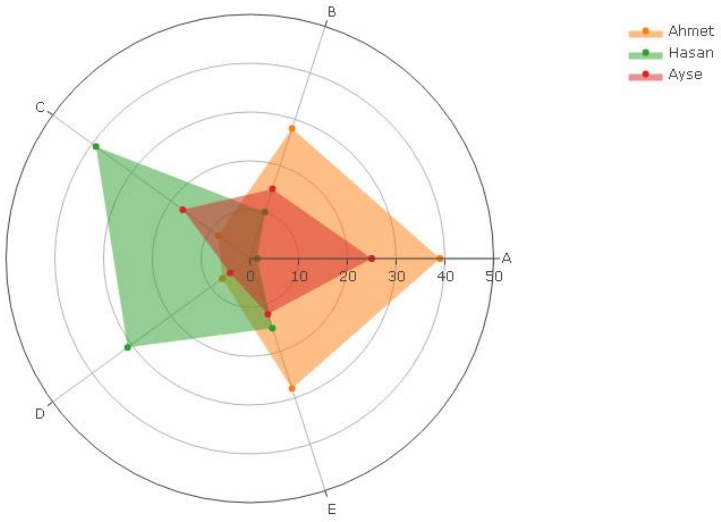
Çoklu nicel değişkenleri karşılaştırmak, aralarındaki ilişkiyi tespit etmek amacı ile radar grafiği kullanılmaktadır. Bu grafik sayesinde düşük ve yüksek değerler kolayca tespit edilmektedir. Radar Grafik ikiden fazla değişkenin bir merkezden çıkan eksenleri üzerinde şekillendirir. Bu eksenin boyutu tüm değişkenler için aynıdır. Tüm eksenler birbirine eşit mesafede bir çember (radyal) oluşturacak şekilde sıralanır. Değişkenin aldığı değerler kendi ekseni boyunca ve diğer değişkenlerin ekseni üzerinde gösterilirken noktalar poligonal olarak birleştirilir. Görüntüyü daha anlaşılır hale getirmek adına da kılavuz denilen çizgilere başvurulur.

Aşağıdaki örnekte *plotly* paketi kullanılmıştır. *type = 'scatterpolar'* komutu ile basit radar grafiği çizilebilir. Örneğimizde Ahmet, Hasan ve Ayşe'nin bir ay boyunca kaç saat belirli sosyal medya araçlarını (A, B, C, D, E) kullandığı grafiğe yansıtılmıştır.

```
library(plotly)

fig <- plot_ly(
   type = 'scatterpolar',
   fill = 'toself'
 )
fig <- fig %>%
 add_trace(
```

```
    r = c(39, 28, 8, 7, 28, 39),
    theta = c('A','B','C', 'D', 'E', 'A'),
    name = 'Ahmet'
  )
fig <- fig %>%
  add_trace(
    r = c(1.5, 10, 39, 31, 15, 1.5),
    theta = c('A','B','C', 'D', 'E', 'A'),
    name = 'Hasan'
  )
fig <- fig %>%
  add_trace(
    r = c(25, 15, 17, 5, 12, 25),
    theta = c('A','B','C', 'D', 'E', 'A'),
    name = 'Ayşe'
  )
fig <- fig %>%
  layout(
    polar = list(
      radialaxis = list(
        visible = T,
        range = c(0,50)
      )
    )
  )
```

```
fig
```



Grafik 57: Radar Grafiği Örneği

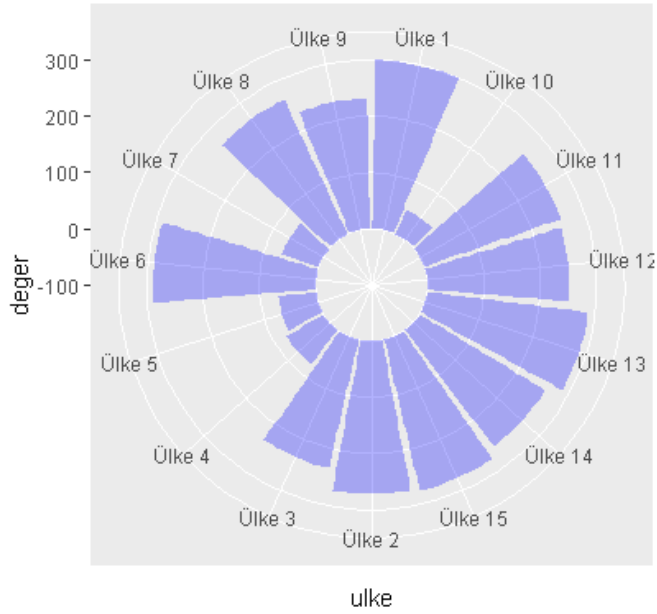
4.16. Radyal Çubuk Grafik (Radial Bar Plot)

Kutuplu bir koordinat sisteminde çizilen basit çubuk grafikler radyal çubuk grafik diye adlandırılmaktadır. Bu tarz grafikler çoklu veri setlerinde yer kazanmak veya estetik olarak farklılık yaratmak amacı ile tercih edilir. Lakin bu grafiklerin yorumlanma zorluğu olduğu da unutulmamalıdır (r-graph-gallery.com, 2020). Aşağıdaki örnekte data isimli veri seti üretilip kullanılmıştır. 15 farklı ülke ve bu ülkelerin aldığı 10-100 arasındaki değerler ile veri görselleştirme yapılmıştır.

```
#veri setinin oluşturulması
data <- data.frame(
  id=seq(1,60),
  ulke=paste( "Ülke ", seq(1,15), sep=""),
  deger=sample( seq(10,100), 15, replace=T)
)
 head(data)
##   id   ulke deger
## 1  1 Ülke 1   100
## 2  2 Ülke 2    68
## 3  3 Ülke 3    78
## 4  4 Ülke 4    19
## 5  5 Ülke 5    17
## 6  6 Ülke 6    96
```

Oluşturulan veri *ggplot* kütüphanesi ile görselleştirilmiştir. Önce veriden çubuk grafik üretilmiş, sonra *coord_polar* fonksiyonu ile çember şeklini almıştır.

```
ggplot(data, aes(x = ulke, y = deger)) +
  geom_bar(stat="identity", fill=alpha("blue", 0.3)) +

  #ylim(iç boşluk, dış sınırı)
  ylim(-100,300) +
  coord_polar(start = 0)
```



Grafik 58: Radyal Grafik Örneği

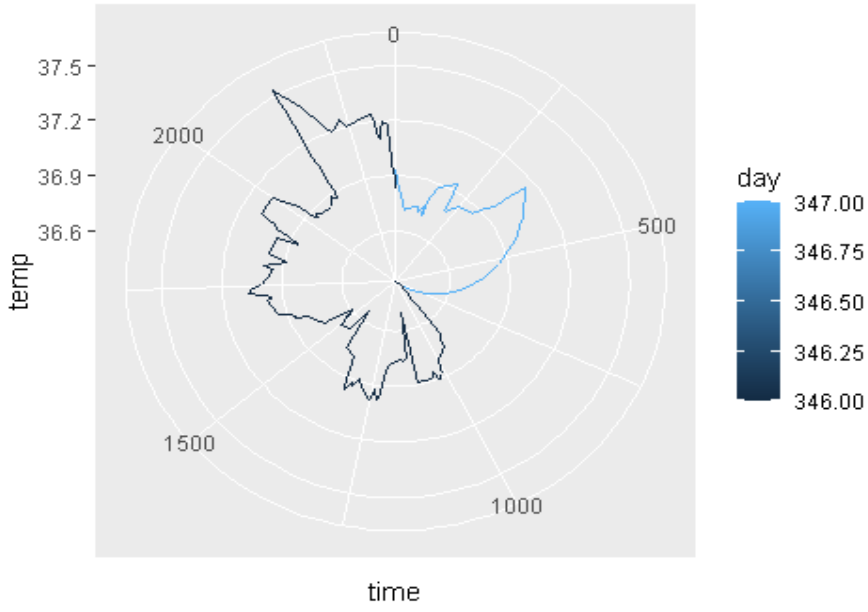
4.17. Sarmak Çizim Grafiği (Spiral Plot)

Zamana dayalı verilerin spiral şeklinde görselleştirilmesinde sarmal çizim grafiği kullanılmaktadır. Sarmak grafik sarmalın merkezinden başlar ve dışarı doğru açılarak ilerler. Zamanın mevsimselliğini, periyot içinde değişiklikleri anlamak açısından verimli bir grafik türüdür. Ayrıca büyük veri setlerinin görselleştirilmesinde kullanılmaktadır.

Aşağıda *beaver* veri seti ile basit bir görselleştirme yapılmıştır. Bu veri seti Wisconsin de konduzlara ait verileri içeren bir veri setidir. Veri seti gün, zaman, sıcaklık ve aktivite kolonlarından oluşmaktadır.

```
library(ggplot2)
data(beavers)
```

```
kunduz <- ggplot(beaver1, aes(time, temp, colour = day)) +
  geom_line() +
  coord_polar(theta = "x")
kunduz
```



Grafik 59: Sarmal Grafik Örneği

4.18. Güneş Yanığı Grafiği (Sunburst Graph)

Büyük veri setlerinden belli bir hiyerarşik yapıları görüntülemek için güneş yanığı grafiği kullanılabilir. İçten dışa doğru hiyerarşik yapıların daire ve halka üzerinde temsili yapılır. Temelde yığılmış çubuk grafiğinin dairesel gösterimi gibi düşünülebilir.

Aşağıda *sunburstR* paketi ile yapılan güneş yanığı grafiği oluşturulmuştur (Bostock ve diğ., 2020). Veri seti olarak kıtalar,

bölgeler ve ülkeler hiyerarşik yapısı içinde ülkelerin nüfusları görselleştirilmiştir. Veri seti;

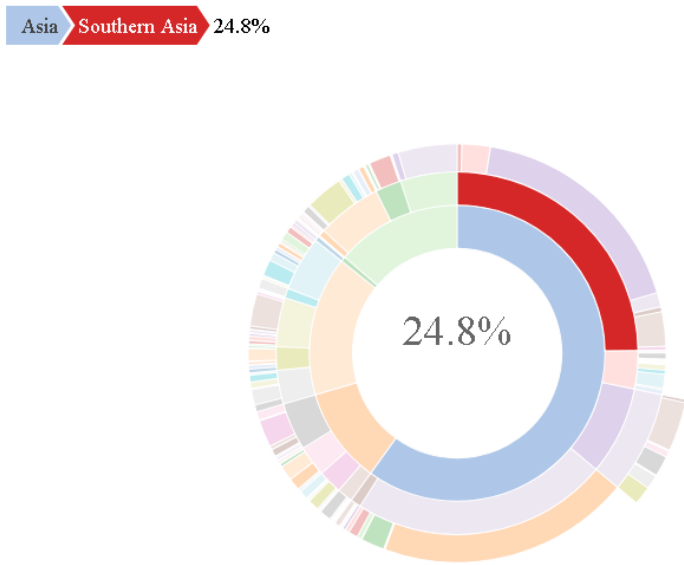
https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/11_SevCatOneNumNestedOneObsPerGroup.csv adresinden yüklenebilir.

```
library(tidyverse)
library(treemap)
library(sunburstR)
data <-
read.table("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/11_SevCatOneNumNestedOneObsPerGroup.csv", header=T, sep=";")
data[ which(data$value==-1),"value"] <- 1
colnames(data) <- c("Continent", "Region", "Country", "Pop")
head(data)
##   Continent         Region       Country      Pop
## 1      Asia   Southern Asia    Afghanistan 25500100
## 2    Europe Northern Europe Ã…land Islands    28502
## 3    Europe Southern Europe        Albania  2821977
## 4    Africa Northern Africa        Algeria 37900000
## 5   Oceania      Polynesia American Samoa    55519
## 6    Europe Southern Europe        Andorra    76246
data <- data %>%
 filter(Continent != "") %>%
 mutate(path = paste(Continent, Region, Country, sep="-")) %>%
```

```
 dplyr::select(path, Pop)
head(data)
##                                path      Pop
## 1        Asia-Southern Asia-Afghanistan 25500100
## 2 Europe-Northern Europe-Ã…land Islands    28502
## 3        Europe-Southern Europe-Albania  2821977
## 4        Africa-Northern Africa-Algeria 37900000
## 5     Oceania-Polynesia-American Samoa    55519
## 6        Europe-Southern Europe-Andorra    76246
```

Görselleştirme aşağıdaki şekilde interaktif olarak yapılabilir.

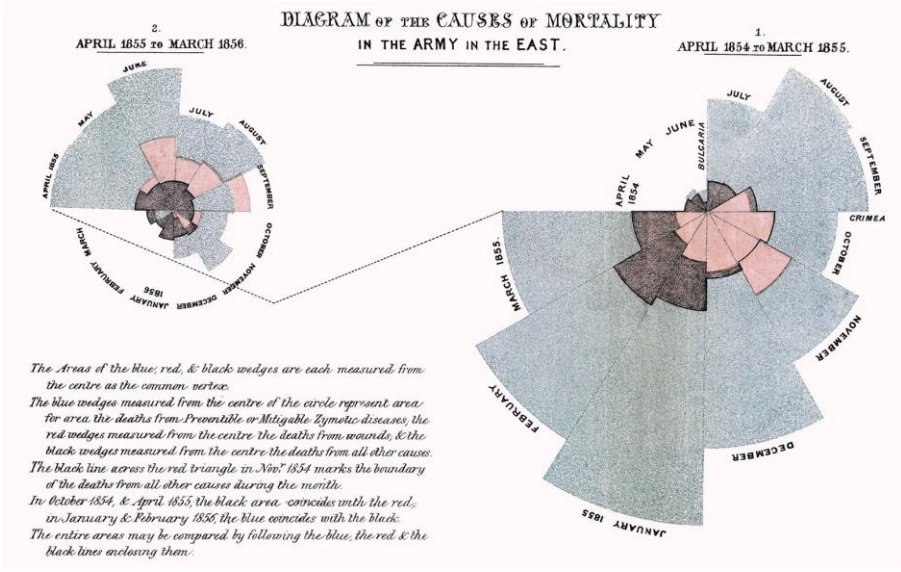
```
p <- sunburst(data, legend=FALSE)
p
```



Grafik 60: Güneş Yanığı Grafik Örneği

4.19. Dairesel Grafik, Gül Diyagramı (Nightingale Rose Chart)

Dairesel Grafik, bazı kaynaklarda Gül Diyagramı olarakta anılır; ilk olarak Kırım Savaşı esnasında ölen İngiliz askerlerinin ölüm nedenlerinin analizinde Florence Nigtingale tarafından tercih edilmiştir. Hemşire olmasının yanında iyi bir istatistikçi olan Nigtingale, Kırım'da ölen İngiliz askerlerinin birçoğunun aldıkları yaralardan veya başka nedenlerden çok hastalıklarından öldüklerini çarpıcı bir şekilde gözler önüne serilmiştir.



Grafik 61: Nightingale Orijinal Gül Diyagram

Dairesel Grafik/Gül Diyagramı, kutupsal bir koordinat sistemine içerisinde yer almaktadır. Her bir veri veya değişklen eşit radyal dairesel bölümler ile temsil edilmektedir. Verinin değerine göre her bir radyol bölüm merkezden uzaklaşmaktadır. Her bir veri kendi değerine

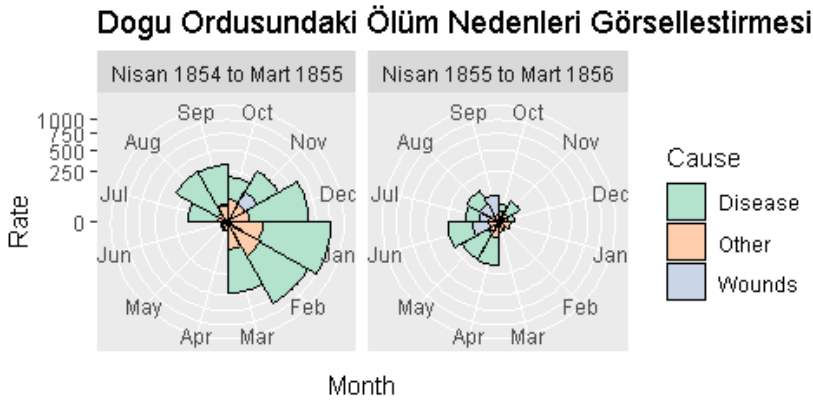
göre merkeze oranlanır. Merkezden çıkan dilim ne kadar büyükse ilgili veri de o kadar büyüktür.

Aşağıda Nightingale tarafından çizilen dairesel grafik görselleştirilmiştir. Bu grafiğin çizilmesi normal R kullanıcıları için zorlayıcı olabilir.

Önce *HistData* kütüphanesi içinden Nightingale veri setine ulaşılabilir.

```
library(HistData)
library(tidyverse)
library(magrittr)
library(grid)
library(gridExtra)
head(Nightingale)
##         Date Month Year  Army Disease Wounds Other Disease.rate
Wounds.rate
## 1 1854-04-01   Apr 1854  8571       1      0     5         1.4         0.0
## 2 1854-05-01   May 1854 23333      12      0     9         6.2         0.0
## 3 1854-06-01   Jun 1854 28333      11      0     6         4.7         0.0
## 4 1854-07-01   Jul 1854 28722     359      0    23       150.0         0.0
## 5 1854-08-01   Aug 1854 30246     828      1    30       328.5
0.4
## 6 1854-09-01   Sep 1854 30290     788     81    70       312.2
32.1
##   Other.rate
## 1        7.0
## 2        4.6
```

```
## 3       2.5
## 4       9.6
## 5      11.9
## 6      27.7


Nightingale %>%
  select(Date, Month, Year, contains("rate")) %>%
  pivot_longer(cols = 4:6, names_to = "Cause", values_to = "Rate")
%>%
  mutate(Cause = gsub(".rate", "", Cause),
         period = ifelse(Date <= as.Date("1855-03-01"),
                         "Nisan 1854 to Mart 1855", "Nisan 1855 to Mart
1856"),
         Month = fct_relevel(Month, "Jul", "Aug", "Sep", "Oct", "Nov",
"Dec", "Jan", "Feb", "Mar", "Apr", "May",
                             "Jun")) %>%
  ggplot(aes(Month, Rate)) +
  geom_col(aes(fill = Cause), width = 1, position = "identity",
           color = "black") +
  coord_polar(start = 3 * pi / 2)+
  facet_wrap(~period) +
  scale_fill_brewer(type = "qual", palette = "Pastel2") +
  scale_y_sqrt() +
  ggtitle("Doğu Ordusundaki Ölüm Nedenleri Görselleştirmesi")
```



Grafik 62: Nightingale Gül Diyagram

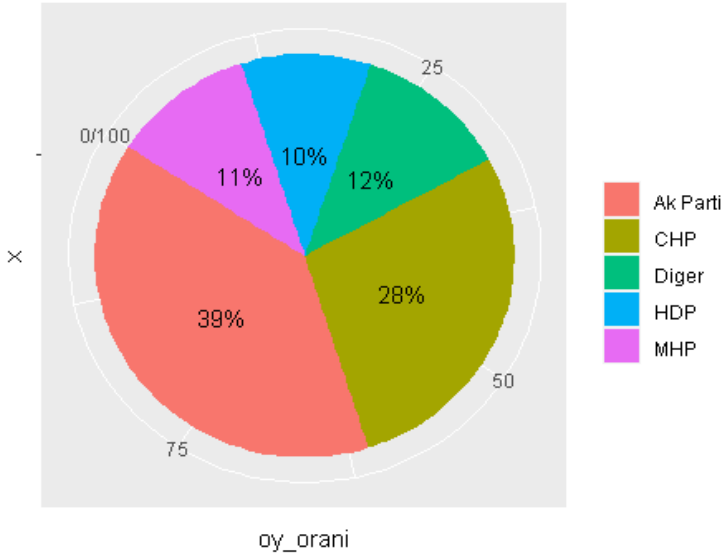
4.20. Pasta Grafiği (Pie Plot)

Değişkenler arasındaki ilişkiyi oranlar veya yüzdeler ile bir daireyi o orana göre bölümler bölerek gösteren grafiğe pasta grafiği adı verilmektedir. Tam daire %100'e eşittir. Daire içindeki her bir yayın uzunluğu bir değişkeni ifade eder. Değişkenlerin toplam değeri %100 kabul edilerek değişkenlerin toplam içinde oranları görselleştirilir. Verilerin oranlarını toplam içinde hızlıca değerlendirmek için uygun bir yöntem olan pasta grafiği, değişken sayısı arttıkça kullanışsız hale gelmektedir.

Aşağıda partilerin oy dağılımları varsayımsal olarak görselleştirilmiştir.

```
parti <- c("Ak Parti", "CHP", "MHP", "HDP", "Diğer")
oy_orani <- c(39, 28, 11, 10, 12)
data<- data.frame(x=parti, y=oy_orani)
```

```
ggplot(data=data,aes(x="",y=oy_orani, fill=parti)) +
  geom_bar(stat="identity",width=1) +
  coord_polar("y",start = -1) +
  geom_text(aes(label = paste0(oy_orani, "%")),
            position = position_stack(vjust = 0.5)) +
  labs(fill="", title = "Partilerin Oy Dağılımı")
```

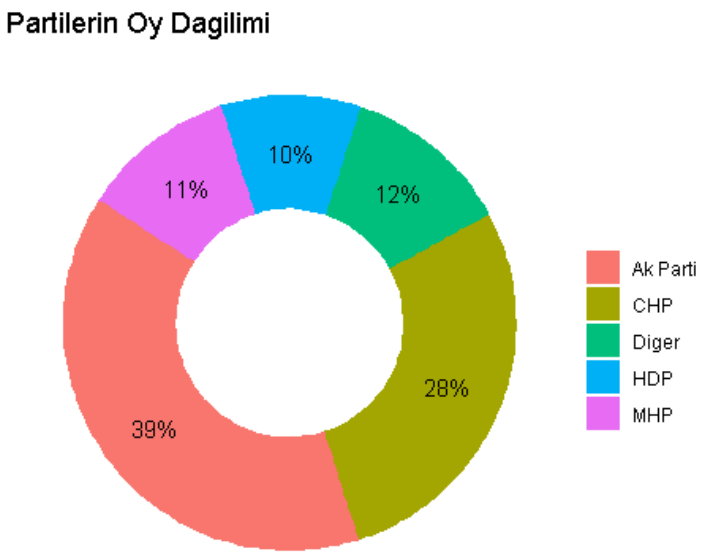


Grafik 63: Pasta Grafiği Örneği

4.21. Halka Grafiği (Donut Chart)

Halka grafik grafiği, içinde bir delik bulunan basit bir pasta grafiğidir. Pasta grafiği arasındaki tek fark, pasta grafiğin içinde delik oluşturmak için: x = 2 ve xlim = c (0.5, 2.5) ayarlamamızdır. Ayrıca *geom_bar ()* fonksiyonundaki argüman genişliği artık gerekli değildir.

```
parti <- c("Ak Parti", "CHP", "MHP", "HDP", "Diğer")
oy_orani <- c(39, 28, 11, 10, 12)
data<- data.frame(x=parti, y=oy_orani)

ggplot(data=data,aes(x=2,y=oy_orani, fill=parti)) +
  geom_bar(stat="identity",width=1) +
  coord_polar("y",start = -1) +
  geom_text(aes(label = paste0(oy_orani, "%")),
    position = position_stack(vjust = 0.5)) +
  labs(fill="", title = "Partilerin Oy Dağılımı") +
  theme_void()+
  xlim(0.5, 2.5)
```



Grafik 64: Halka Grafiği Örneği

4.22. Ağaç Haritası Grafiği (Tree Map)

Ağaç haritası grafiği, ağaç benzeri bir yapıdaki hiyerarşik verileri temsil etmek için iç içe yerleşmiş dikdörtgenleri kullanılır. Dallar ve alt dallar olarak düzenlenen veriler, boyutları ve çizim renkleri her dikdörtgenle ilişkili nicel değişkenler kullanılarak hesaplanan dikdörtgenler kullanılarak temsil edilir. Veri içindeki hiyerarşik yapının temsilinde her dikdörtgen iki nicel veriyi temsil eder. Teorik olarak bu yöntem ile sınırsız sayıda dikdörtgene ulaşılabilir.

Bu görselleştirme için *GNI2014* veri seti kullanılacaktır. Bu veri seti dolar cinsinden, 2014 yılında ülkelerin safi milli gelir rakamlarını vermektedir. Ayrıca veri seti içinde ülkenin bulunduğu kıta ve ülkelerin nüfusu bulunmaktadır. Analiz için öncelikle aşağıdaki iki paketin yüklenmesi gerekmektedir. Sonrasında aşağıdaki gibi veri seti tanıtılır. *head()* komutu ile ilk beş veri incelenir (Treemap, 2020).

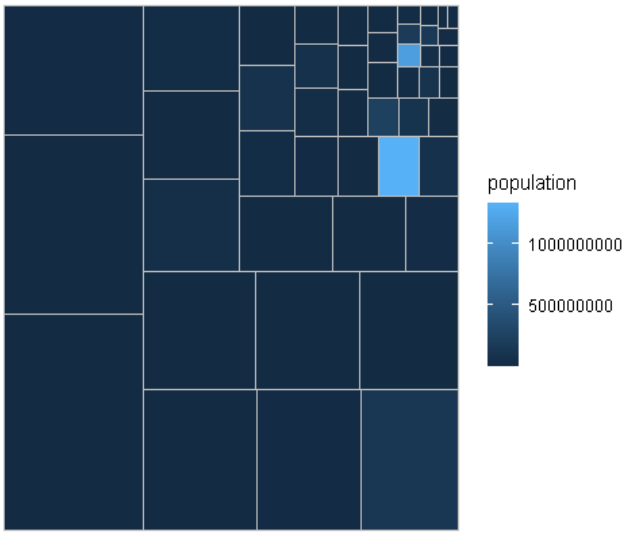
```
library(treemapify)
library(treemap)
```

```
data("GNI2014")
data <- GNI2014
head(data)
##    iso3 country continent population GNI
## 3   BMU Bermuda North America 67837 106140
## 4   NOR Norway Europe 4676305 103630
## 5   QAT Qatar Asia 833285 92200
```

```
## 6   CHE        Switzerland          Europe     7604467   88120
## 7   MAC  Macao  SAR,  China          Asia       559846   76270
## 8  LUX      Luxembourg      Europe    491775  75990
```

Asya kıtası içindeki kişi başı safi milli geliri büyüklük olarak görmek istediğimizi ve arka plan rengin nüfusu temsil etmesini istediğimizi düşünelim. Bu durumda aşağıdaki kod sayesinde bunu yapabiliriz. *ggplot* kütüphanesi içinde *geom_treemap()* fonksiyonu ile görselleştirme yapılır.

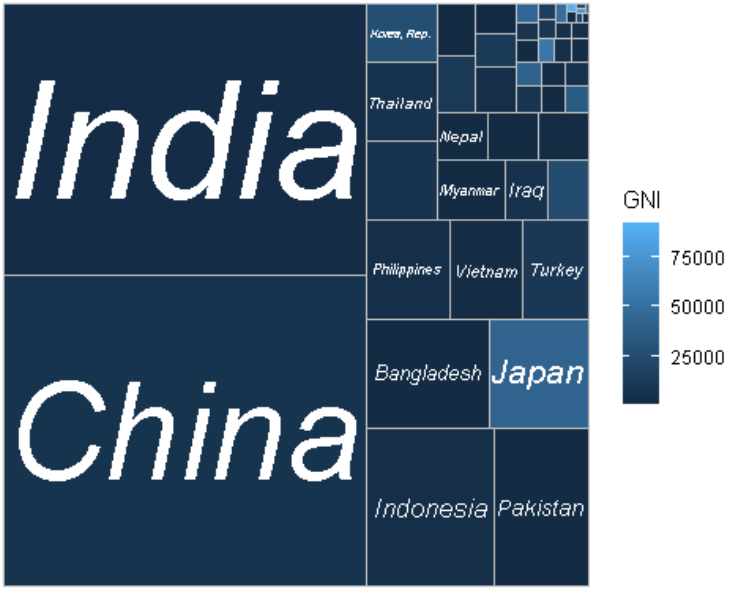
```
#sadece Asya kıtasındaki ülkeleri filtreyelim
data%>%filter(continent=="Asia")->Asia

ggplot(Asia, aes(area = GNI, fill = population)) +
  geom_treemap()
```



Grafik 65: Ağaç Haritası Grafiği Örneği

Örneğimize *label* fonkisyonu ile etiket ekleyebilir, etiketin özelliklerini ise geom_treemap_text () ile ayarlayabiliriz.

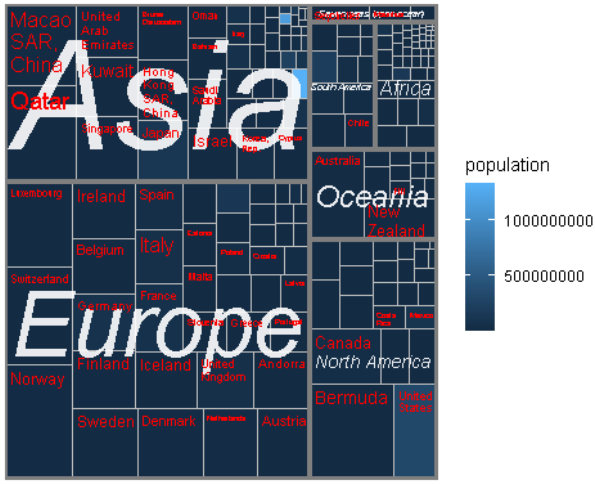
```
#sadece Asya kıtasındaki ülkeleri filtreyelim
data%>%filter(continent=="Asia")->Asia

ggplot(Asia, aes(area = population, fill = GNI, label = country)) +
  geom_treemap() +
  geom_treemap_text(fontface = "italic", colour = "white", place = "centre", grow = TRUE)
```



Grafik 66 Ağaç Haritası Grafiği Örneği

Kıtaların içinde ülke kırımlı bir görselleştirme istediğimizi düşünelim bu sefer elimizde iki gruplu bir veri olacaktır. Bunun için ana grubu yani kıtayı *subgroup* fonksiyonu ile tanımlamamız gerekecektir.

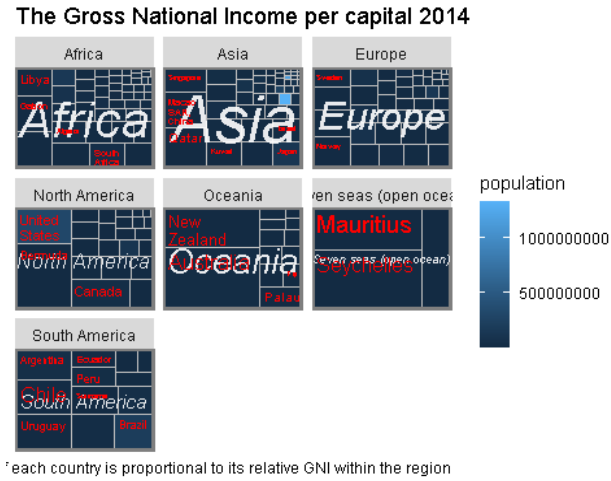
```
ggplot(GNI2014, aes(area = GNI, fill = population, label = country,
                    subgroup = continent)) +
  geom_treemap() +
  geom_treemap_subgroup_border() +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha =
.9, colour =
                               "White", fontface = "italic", min.size = 0) +
  geom_treemap_text(colour = "Red", place = "topleft", reflow = T)
```



Grafik 67: Ağaç Haritası Grafiği Örneği

Yukarıdaki örnekteki bilgileri kıtalara göre ayrı ayrı görselleştirmek istersek aşağıdaki kodu kullanabiliriz.

```
ggplot(GNI2014, aes(area = GNI, fill = population, label =
country,subgroup = continent)) +
  geom_treemap() +
  geom_treemap_subgroup_border() +
  geom_treemap_subgroup_text(place = "centre", grow = T, alpha =
.9, colour ="White", fontface = "italic", min.size = 0) +
  geom_treemap_text(colour = "Red", place = "topleft", reflow = T) +

  #facet_wrap fonkisyonu ile alt kırılımlara göre görselleştirme
yapabiliyorduk.
  facet_wrap(~continent)+
  labs(title = "The Gross National Income per capital 2014 ",
    caption = "The area of each country is proportional to its relative
GNI within the region",
    fill = "population")
```



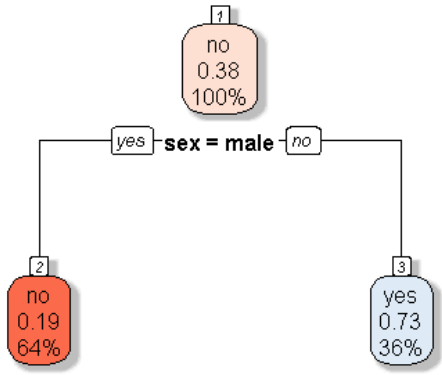
Grafik 68: Ağaç Haritası Grafiği Örneği

4.23. Ağaç Diyagram- Karar Ağacı (Tree Diagram- Decision Tree)

Ağaç diyagram ağaç benzeri gövdesi ve gövdeden çıkan dallar ile verinin görselleştirmesini sağlayan grafiklerdir. Hiyerarşik yapının temsilinde sıkça kullanılmaktadır. Şirketlerin organizasyon yapılarının görselleştirilmesinde, aile ilişkilerinin temsilinde, sınıflandırma işlemlerinde sıklıkla kullanılmaktadır. Makine öğrenmesinde karar ağacının oluşturulmasında da kullanılmaktadır.

Titanic veri setimizi ile bir görselleştirme örneği aşağıda sunulmuştur. Bu örnekte *rpart* ve *rpart.plot* kütüphanelerinden yararlanılmıştır. *rpart* fonksiyonu içindeki *cp* değeri ağaç dallarının ne kadar kompleks olacağını ne kadar dallanacağını belirler. Rakam küçüldükçe oluşan dal sayısı artmaktadır.

```
library(rpart)
library(rpart.plot)

agac<- rpart(survived ~., data=Titanic, cp=.02)

rpart.plot(agac, box.palette = "RdBu", shadow.col = "gray",
nn=TRUE)
```

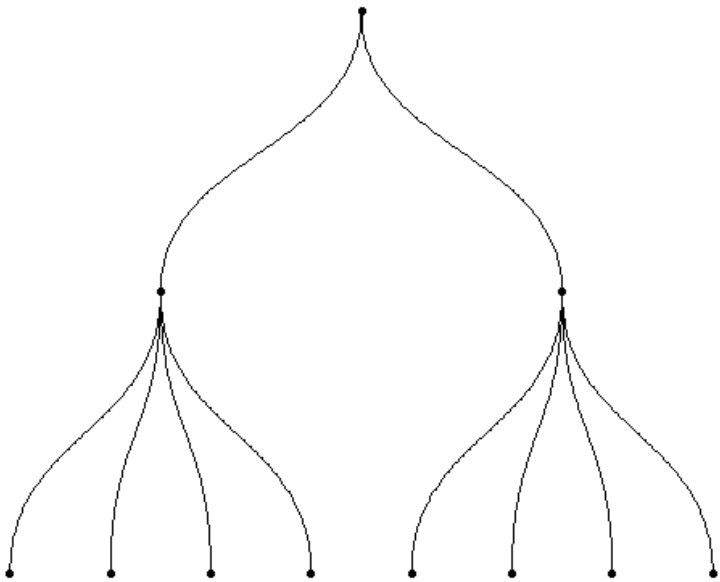


Grafik 69: Karar Ağacı Örneği

Konuyla ilgili bir başka örnek de aşağıda alıntılanmıştır. Bu örnekte *ggraph* ve *igraph* paketleri kullanılmıştır (R-graph, 2021).

```
# Gerekli kütüpanelerin yüklenmesi
library(ggraph)
library(igraph)
library(tidyverse)


# Verisetinin tanımlanması
data <- data.frame(
  level1="CEO",
  level2=c( rep("boss1",4), rep("boss2",4)),
  level3=paste0("mister_", letters[1:8])
)
# Düğümlerin tanımlanması
```

```
edges_level1_2 <- data %>% select(level1, level2) %>% unique %>%
rename(from=level1, to=level2)
edges_level2_3 <- data %>% select(level2, level3) %>% unique %>%
rename(from=level2, to=level3)
edge_list=rbind(edges_level1_2, edges_level2_3)

# Grafigin çizdirilmesi
mygraph <- graph_from_data_frame( edge_list )
ggraph(mygraph, layout = 'dendrogram', circular = FALSE) +
 geom_edge_diagonal() +
 geom_node_point() +
 theme_void()
```



Grafik 70: Karar Ağacı Grafiği Örneği

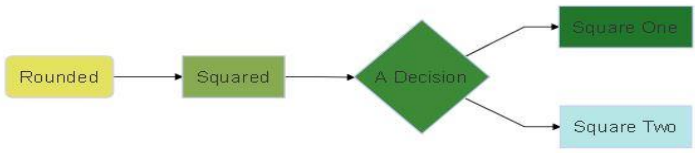
4.24. Akış Grafiği (Flow Chart)

Bir işlemin başlangıcı ve sonrasında yapılması gereken adımların gösterildiği veya bağlantılı olayların tasviri ve değişkenler arası ilişkilerin gösterildiği bir başka grafik türü de akış grafiğidir. Bu grafik modeli; işlem haritası, işlem grafiği, diyagram vb. isimlerde anılmaktadır. Akışın simgeler ile anlatılması ile kullanıcılar iş akışı hakkında görsel mesaja maruz kalırlar ve akışın anlaşılması kolaylaşmaktadır. Süreçlerin anlaşılmasında, planlanma ve iyileştirilmesinde grafik kullanıcılara yardımcı olur. Süreçler belli bir sembolün içine yazılır, çizgi ve oklar ile süreç adımlarının yönü ve akışı belirlenmektedir.

Aşağıdaki örnekte akış grafiği oluşturmak için çok kullanılan *DiagrammeR* paketi kullanılmıştır (Iannone, 2020).

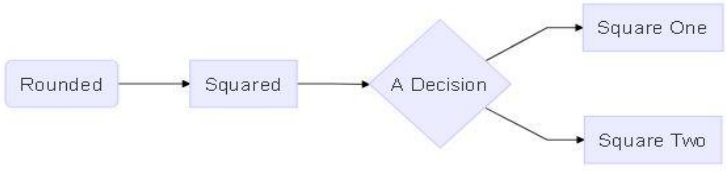
```
library(DiagrammeR)

DiagrammeR("graph LR;
A(Rounded)-->B[Squared];
B-->C{A Decision};
C-->D[Square One];
C-->E[Square Two];

style A fill:#E5E25F;
style B fill:#87AB51;
style C fill:#3C8937;
```

```
style D fill:#23772C;
style E fill:#B6E6E6;"
)
```



Grafik 71: Akış Grafiği Örneği

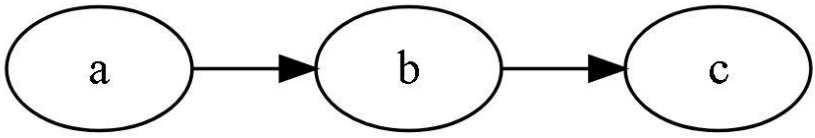
```
library(DiagrammeR)
DiagrammeR("graph LR;A(Rounded)-->B[Squared];B-->C{A
Decision};
 C-->D[Square One];C-->E[Square Two];
 style A fill:#E5E25F;  style B fill:#87AB51; style C fill:#3C8937;
 style D fill:#23772C;  style E fill:#B6E6E6;"
)
```



Grafik 72: Akış Grafiği Örneği

Aşağıdaki örnekte aynı paket kullanılmıştır. *layout* değişkeni ile şekil *rankdir* değişkeni ile akış yönü tanımlanmıştır.

```
DiagrammeR::grViz("digraph {

graph[layout = dot, rankdir = LR]

a
b
c
a -> b -> c
}")
```

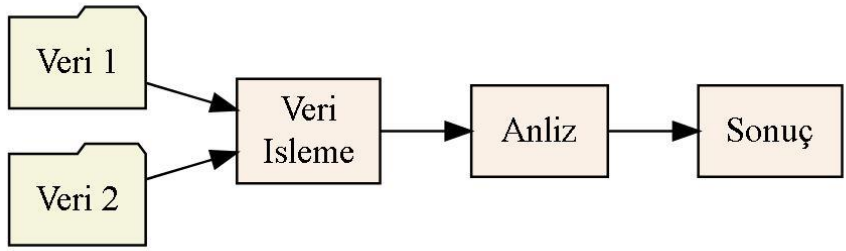


Grafik 73: Akış Grafiği Örneği

Basit bir iş akışı örneği de aşağıda sunulmuştur.

```
DiagrammeR::grViz("digraph {

graph [layout = dot, rankdir = LR]

# Node tanımlaması yapılmıştır.
```

```
node [shape = rectangle, style = filled, fillcolor = Linen]

data1 [label = 'Veri 1', shape = folder, fillcolor = Beige]
data2 [label = 'Veri 2', shape = folder, fillcolor = Beige]
process [label =  'Veri \n İşleme']
statistical [label = 'Anliz']
results [label= 'Sonuç']

# edge definitions with the node IDs
{data1 data2}  -> process -> statistical -> results
}")
```



Grafik 74: Akış Grafiği Örneği

4.25. Sankey Diyagramı (Sankey Diagram)

Sankey diyagramları temelinde akış diyagramıdır. Verilerin bir yerden diğer bir yere akışlarını ve çizgiler yardımı ile verinin yer değiştirirken aldığı değeri yani büyüklüğü görselleştirmektedir. Çizgi ve okların büyüklüğü verinin de büyük olduğuna işarettir. Diyagramda farklı birçok kategori olabilmekte, bu kategoriler arasında alt kategoriler ve

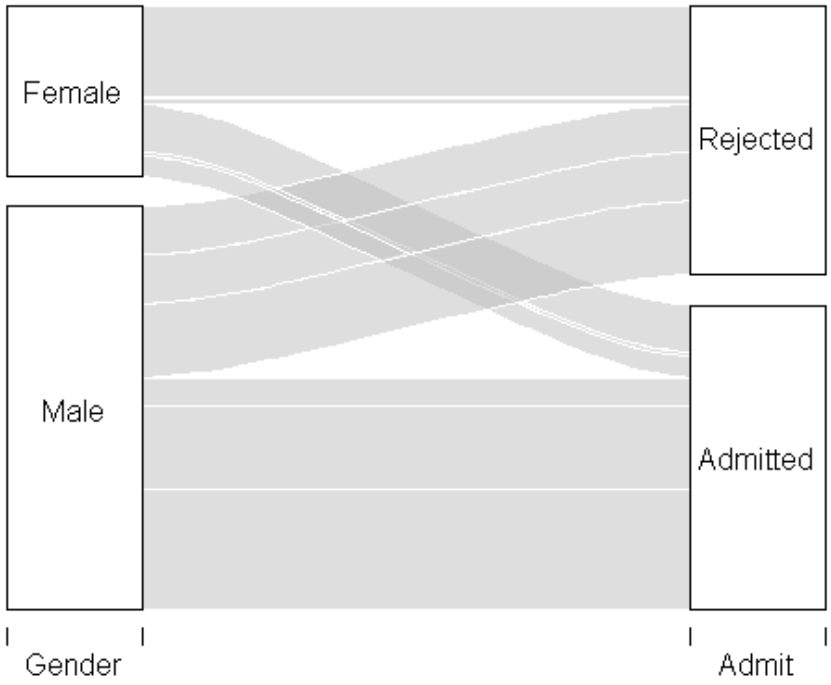
bu kategorilere de geçişler mümkündür. Süreç içinde akış çizgileri birleşebilir veya ayrılabilmektedir.

Alüvyon grafiği ise Sankey grafiğinin farklı bir çeşididir. Zaman içindeki frekans dağılımlarını görselleştirir.

UC Berkley Üniversitesine yapılan başvuruların cinsiyeti, bölümleri ve kabul durumundan oluşan data (*UCBAdmissions*) ile aşağıda basit bir görselleştirme yapılmıştır. Bu görselleştirme de *alluvial* kütüphanesi kullanılmıştır. Sankey diyagramı görselleştirmesi için *networkD3* kütüphanesinde sankey diyagramı çizmek için kullanılabilir. Ayrıca *ggplot* içindeki *geom_alluvium()* fonksiyonu ile de diyagramı çizdirmek mümkündür.

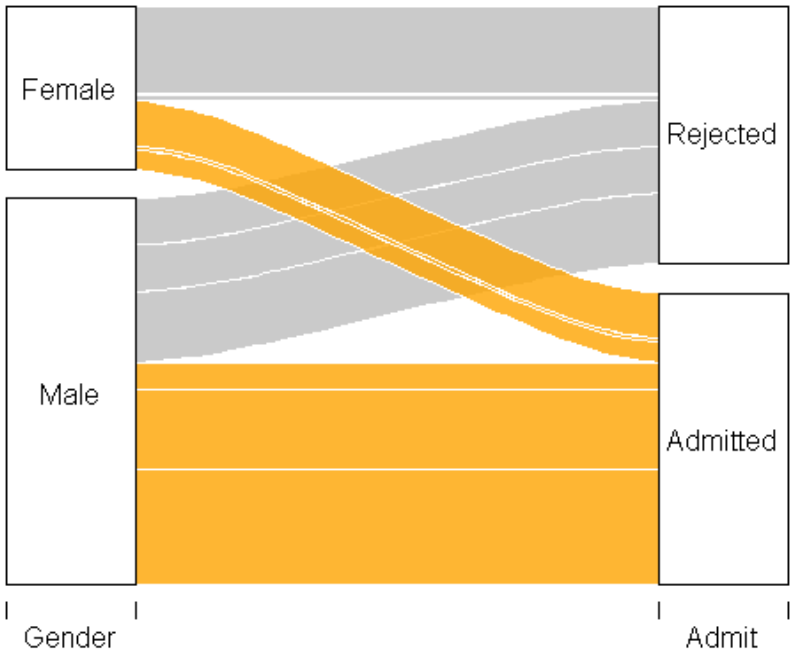
Veri seti içinde kadın ve erkek öğrencilerin üniversiteye kabul edilip edilmemesi durumu aşağıdaki şekilde görselleştirilmiştir.

```
library(alluvial)
kabul<- head(as.data.frame(UCBAdmissions), n = 12)
alluvial(kabul[,c(2,1)], freq=kabul$Freq)
```



Grafik 75: Sankey Grafiği Örneği

```
alluvial(kabul[,c(2,1)], freq=kabul$Freq,
        col = ifelse( kabul$Admit == "Admitted", "orange", "grey"),
        alpha = 0.8,
        layer = kabul$Admit == "Rejected"
        )
```

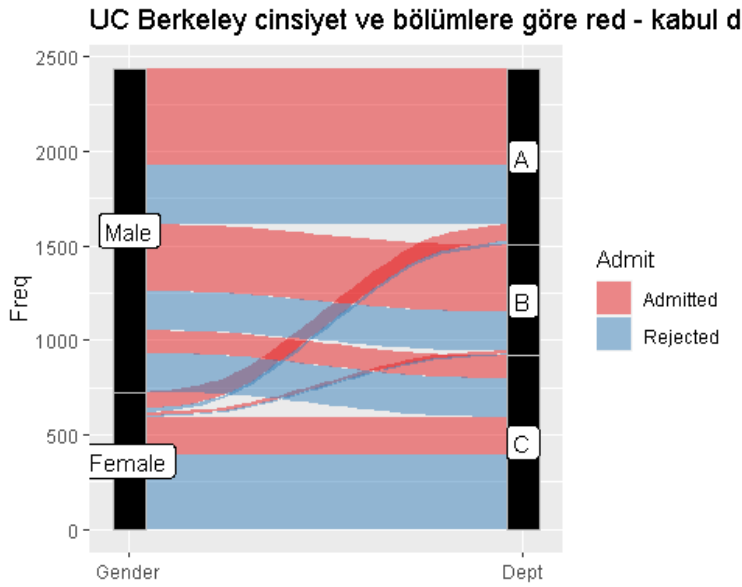


Grafik 76: Sankey Grafiği Örneği

Aynı örneği ggplot kütüphanesi ve *ggalluvial* paketi kullanarak aşağıdaki şekilde görselleştirmemiz mümkündür (Brunson, 2020).

```
library(ggalluvial)
ggplot(kabul,
  aes(y = Freq, axis1 = Gender, axis2 = Dept)) +
  geom_alluvium(aes(fill = Admit), width = 1/12) +
  geom_stratum(width = 1/12, fill = "black", color = "grey") +
  geom_label(stat = "stratum", aes(label = after_stat(stratum))) +
  scale_x_discrete(limits = c("Gender", "Dept"), expand = c(.05, .05))
+
  scale_fill_brewer(type = "qual", palette = "Set1") +
```

```
 ggtitle("UC Berkeley cinsiyet ve bölümlere göre red - kabul durumu")
```

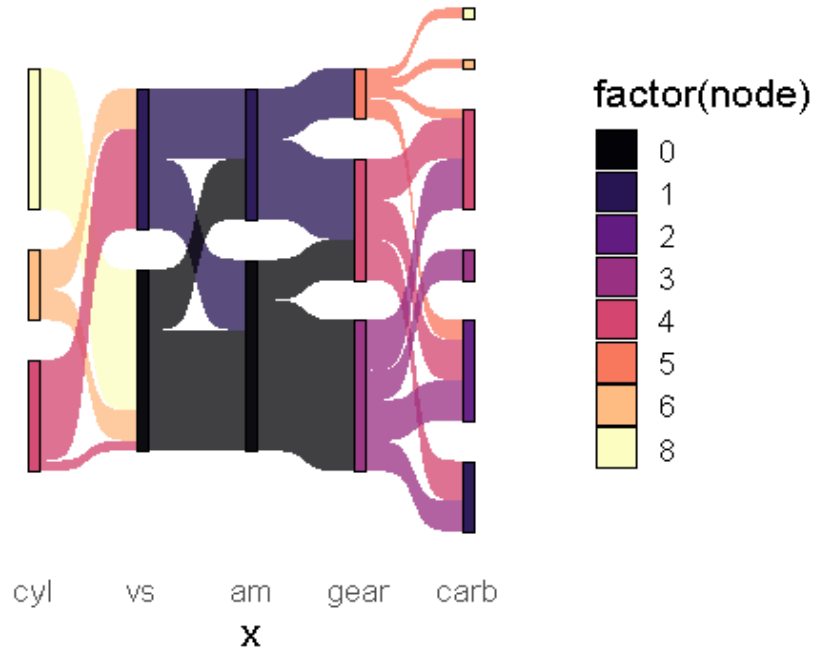


Grafik 77: Sankey Grafiği Örneği

Çok bilindik *mtcars* veri seti ile *gglot2* paketi içinde yer alan *geom_sankey()* fonksiyonu ile de görselleştirme oluşturulabilir . Aşağıdaki örnek Coder'ın çalışmasından alınmıştır (Coder, 2021).

```
#remotes::install_github("davidsjoberg/ggsankey")
library(ggsankey)
# install.packages("ggplot2")
library(ggplot2)
# install.packages("dplyr")
library(dplyr) # Gereklidir.
df <- mtcars %>%
```

```
  make_long(cyl, vs, am, gear, carb)
ggplot(df, aes(x = x,
               next_x = next_x,
               node = node,
               next_node = next_node,
               fill = factor(node))) +
  geom_sankey(flow.alpha = 0.75, node.color = 1) +
  scale_fill_viridis_d(option = "A", alpha = 0.95) +
  theme_sankey(base_size = 16)
```



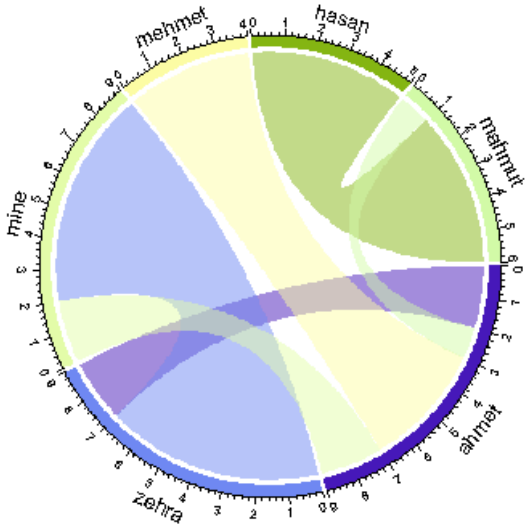
Grafik 78: Sankey Grafiği Örneği

4.26. Akor Diyagramı – (Chord Diagram)

Farklı veri grupları arasındaki benzerlikleri, elemanlar arasındaki ilişkileri görselleştirmek amacı ile Akor diyagramlar kullanılmaktadır. Görselleştirme temelde çember yayı üzerinde düğümlerden oluşur. Bu düğümlerin diğer düğümler ile ilişkisi yaylar veya eğriler ile görselleştirilir. Bu düğümleri birbirlerine bağlayan yayların büyüklüğü ise ilişkinin büyüklüğüne orantılıdır. Akor diyagram ilişki sayısı az ise görsel olarak sorun teşkil etmez, lakin düğüm ve bağlantı sayısı arttıkça görselleştirme giderek karmaşıklaşır ve görsel bağlam kurulmak giderek zorlaşmaktadır.

Aşağıda *circlize* kütüphanesi yardımı ile akor diyagram görselleştirilmiştir (Gu ve GlobalOptions, 2021).

```
library(circlize)
kim<- c("ahmet", "zehra", "mine", "mehmet", "hasan", "mahmut")
kimle<- c("zehra", "mine", "ahmet", "ahmet", "mahmut", "ahmet")
nekadar<- c(2,7,2,4,5,1)

tel<- data.frame(kim,kimle,nekadar)

chordDiagram(x=tel, transparency = 0.5)
```



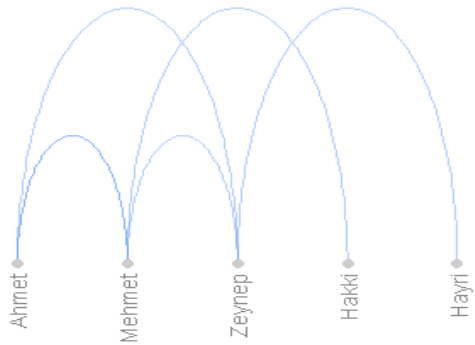
Grafik 79: Akor Grafiği Örneği

4.27. Ark Diyagramı (Arc Diagram)

Akor diyagram çember etrafında ilişkiler görselleştirirken ark diyagram ilişkileri iki boyutlu yatay bir düzlemde görselleştirmektedir. İki boyutlu ağ şemalarının görselleştirilmesinde uygun olup, çok bağlantılı yapılarda diyagram karmaşıklaşmakta ve okunması zorlaşmaktadır. Buradaki çizgiler ilişkinin varlığına işaret etmektedir. Lakin çizgilerin boyutu bir değeri temsil etmez. İki değişken arasında bağlantı sayısı arttıkça çizgi kalınlaşmaktadır.

Aşağıda *arcdiagram* paketi ile yapılan örnek görselleştirmeye yer verilmiştir (Sanchez, 2016).

```
#devtools::install_github('gastonstat/arcdiagram')
library(arcdiagram)
lab = rbind(c("Ahmet", "Mehmet"), c("Ahmet", "Zeynep"),
c("Mehmet", "Hakkı"),
c("Zeynep", "Hayri"), c("Zeynep", "Mehmet"), c("Ahmet",
"Mehmet"))
arcplot(lab)
```



Grafik 80: Ark Grafiği Örneği

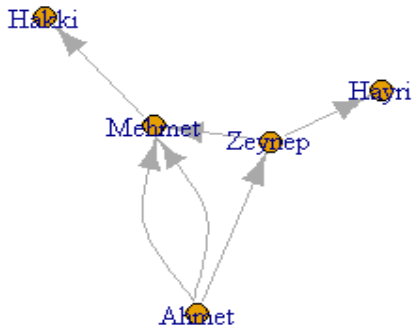
4.28. Ağ Diyagramı (Network Diagram)

Verilerin ve değişkenlerin birbirleri ile nasıl ilişki ve bağlantı içinde olduklarını göstermenin bir diğer yolu da ağ diyagramlar ile görselleştirme yapmaktır. Düğümler, köşeler ve ikisi arasındaki bağlantılar ile ilişkiler görselleştirilir. Düğümlerin büyüklükleri ve bağlantı çizgilerinin kalınlıkları ile ilişkinin gücü, yönü ve etkisi

görselleştirilir. Ağ bağlantılarının veri işlemede sınırlı kapasiteleri vardır ve çok fazla düğüm ve çizgiler görselleştirmeye çok karmaşık hale getirmektedir.

Ağ diyagramı kullanabileceğimiz çok fazla paket mevcuttur. Aşağıdaki örnekte *igraph* paketi ile yukarıda kullandığımız veri seti görselleştirilmiştir.

```
library(igraph)
# Düğümlerin tanıtılması
ag = graph.edgelist(lab, directed = TRUE)
# Grafiğin çizdirilmesi
plot(ag)
```



Grafik 81: Ağ Grafiği Örneği

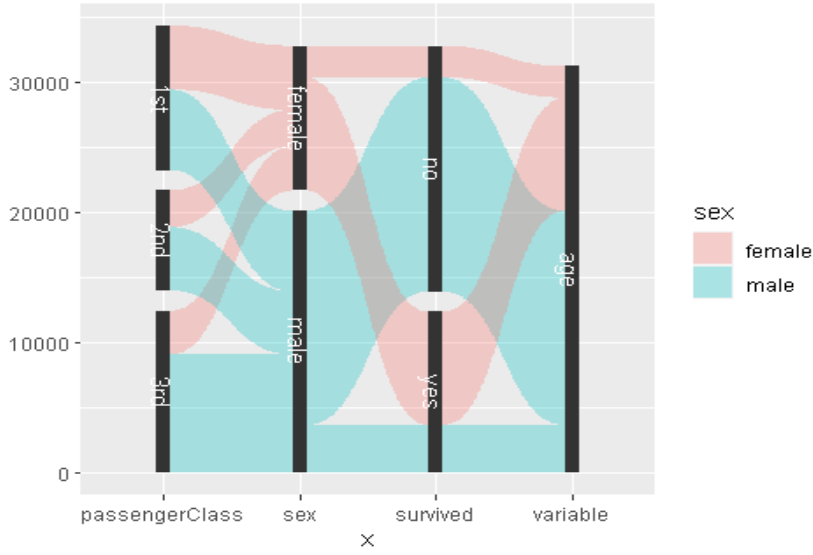
4.29. Paralel Setler, Paralel Koordinat Grafiği (Parallel Sets)

Veri arasında akış ve bağlantının gücünü orantısal olarak gösterilmesinde Sankey grafikleri gibi paralel set grafikleri de kullanılmaktadır.

Paralel set grafikleri akış ve orantıları göstermeleri bağlamında Sankey Diyagramlarına benzer. Öte yandan paralel setler, okları kullanmaz ve akış yolunu görüntülenen her satır kümesinde bölerler. Her satır seti, değer / kategorilerin bu satır setindeki her satırda bölünmesiyle gösterilen bir boyut / veri setine karşılık gelir. Her satır genişliği ve bu hattan kaynaklanan akış yolu, kategori toplamının orantılı kesriyle belirlenir. Her bir akış yolu, farklı kategoriler arasındaki dağılımı göstermek ve karşılaştırmak için renklendirilebilir.

Aşağıda *Titanic* veri setimiz *ggplot* paket içindeki *geom_parallel_sets* fonksiyonu ile görselleştirilmiştir.

```
library(ggforce)
data <- reshape2::melt(Titanic)
data <- gather_set_data(data, 1:4)

ggplot(data, aes(x, id = id, split = y, value = value)) +
  geom_parallel_sets(aes(fill = sex), alpha = 0.3, axis.width = 0.1) +
  geom_parallel_sets_axes(axis.width = 0.1) +
  geom_parallel_sets_labels(colour = 'white')
```



Grafik 82: Paralel Set Grafiği Örneği

4.30. Kelime Bulutu (Word Cloud)

Metin içinde geçen kelimelerin, kelimelerin frekanslarına göre renklendirildiği veya kelimenin büyülüğü ile bu frekans farklarının görselleştirildiği grafiğe kelime bulutu veya kelime etiketi grafiği denir. Metin içindeki tüm kelimeler grafiğin tamamını oluşturmaktadır. Burada istenilirse kelime frekansları filtre olarak kullanılabilir. Bu görselleştirmede uzun kelime öbekleri kısa kelimeleri görsel olarak bastırabilmektedir. Farklı metinlerin karşılaştırılmasında da kullanılan kelime bulutu görsel olarak güzel gözükmesine karşın rakamsan analiz vermekte son derece yetersiz kalmaktadır.

Aşağıda *tm* paketi ile görselleştirilmiş bir kelime bulutu örneği İngilizce dilinde verilmiştir (Feinerer ve Hornik, 2015).

```
#install.packages("tm")
library(tm)
metin<- c("A worker and a returned traveller have tested positive for
coronavirus at Melbourne Airports Holiday Inn, forcing an entire floor
of people
into isolation and prompting a further review of the states hotel
quarantine
system.

The case is the third for the Holiday Inn in less than a week and the
fifth in
less than a fortnight across three Victorian quarantine hotels. Three of
those
are confirmed to be the more infectious British variant of COVID-19.
The new cases did not trigger any tighter community restrictions on
Tuesday
afternoon but contact tracing interviews and genomic sequencing were
still
being carried out on Tuesday night.

The Health Department updated its list of exposure sites late on
Tuesday night
with seven venues in Sunbury, including Cellarbrations and several
stores
in the Sunbury Square Shopping Centre. Anyone at those venues at
the specified
```

```
timesmust get tested and isolate for 14 days.


')
metin2 <- Corpus(VectorSource(metin))
metin2 <- metin2 %>%
  tm_map(removeNumbers) %>%
  tm_map(removePunctuation) %>%
  tm_map(stripWhitespace)
metin2 <- tm_map(metin2, content_transformer(tolower))
dtm <- TermDocumentMatrix(metin2)
matrix <- as.matrix(dtm)
words <- sort(rowSums(matrix),decreasing=TRUE)
df <- data.frame(word = names(words),freq=words)
library(wordcloud)
library(wordcloud2)
library(RColorBrewer)
wordcloud2(data=df, size=1.6, color='random-dark')
```

Grafik 83: Kelime Bulutu Örneği

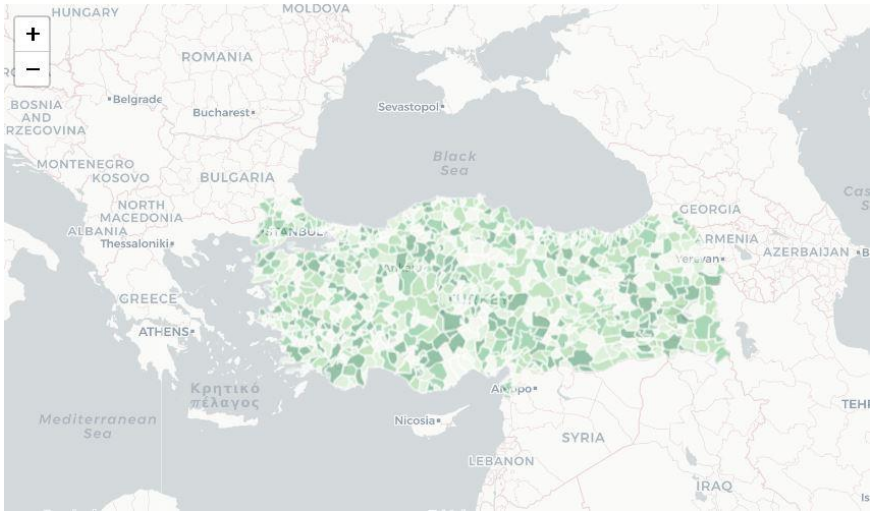
4.31. Koroplet Harita (Choropleth Map)

Koroplet Haritalar ile coğrafi bölgeleri veya alanları ait veriler harita üzerinde gösterilmektedir. Haritanın görselleştirilmesi kullanılan verilere göre renklendirme, noktalama veya diğer şekillerin ilavesi ile mümkündür. Verilerin değişik renk tonları ile harita üzerindeki tasfiri kullanıcılara sınırsız güzellikte görselleştirme imkânı yaratmaktadır. Değişik renk ve gölgelendirme ile kullanıcıların aklında yer edinilmektedir. Verileri açıklamada sıklıkla kullanılan yöntemde tüm sayısal verilerinin harita üzerinde belirtilmesi çok karışık haritaların oluşmasına ve verilerin anlaşılamaması sorunu ortaya çıkarabilmekte, küçük verilerin gözden kaçmasına neden olabilmektedir.

Aşağıda koropet harita örnekleri *leaflet* paketi kullanılarak görselleştirilmiştir (Leaflet, 2015).

```
library(leaflet)
library(raster)

#Türkiye ile ilgili verinin toplanması
usa <- getData("GADM", country="TR", level=2)
usa$randomData <- rnorm(n=nrow(usa), 150, 30)

#Renklerin tanımlanması
pal <- colorQuantile("Greens", NULL, n = 5)
```

```
polygon_popup <- paste0("<strong>Name: </strong>",
usa$NAME_1, "<br>",
                        "<strong>Indicator: </strong>",
round(usa$randomData,2))


#Haritanın yaratılması
 map = leaflet() %>%
    addProviderTiles("CartoDB.Positron") %>%
    setView(39.96, 25.24,
            zoom = 4) %>%
    addPolygons(data = usa,
                fillColor= ~pal(randomData),
                fillOpacity = 0.4,
                weight = 2,
                color = "white",
                popup = polygon_popup)
map
```



Grafik 84: Koroplet Harita Örneği

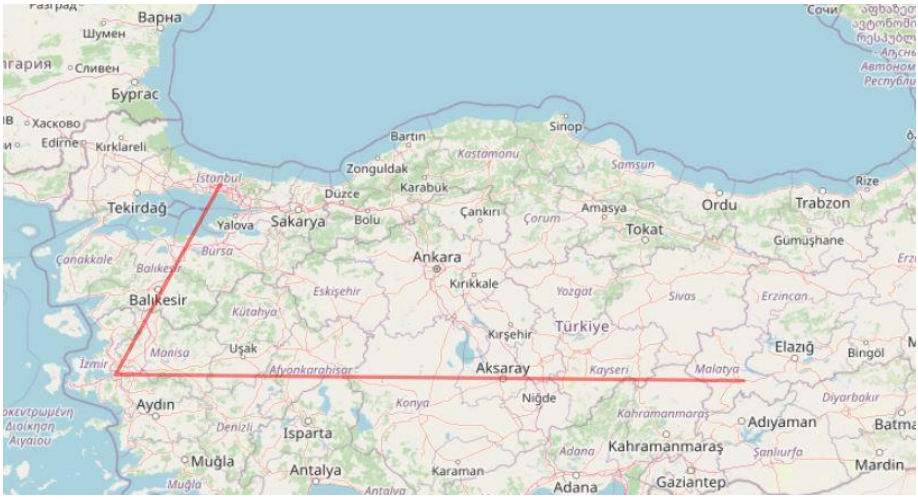
4.32. Bağlantı Haritası (Connection Map)

Bir coğrafi harita üzerinde belirli noktaların düz, eğimli veya kesik çizgiler kullanılarak birleştirilmesi ile bir yol, bağlantı çizilerek oluşturulan harita görselleştirmesine bağlantı haritası ismi verilmektedir. Bir olayda geçen yerlerin rotaların görselleştirilmesinde, yol rotası haritalarında gidilecek yolların bağlantılanmasında sıklıkla kullanılmaktadır.

Aşağıda çok basit bir bağlantı haritası *leaflet* paketi kullanılarak sunulmuştur.

```
library(dplyr)
library(leaflet)
il<- c("istanbul", "izmir", "malatya")
koordinat1 <- c(41.1,38.42,38.35)
```

```
koordinat2 <- c(29.0,27.14,38.33)


baglanti<- data.frame(il,koordinat1, koordinat2)
baglanti%>%
 leaflet() %>%
 addTiles() %>%
 addPolylines(lng = ~koordinat2, lat = ~koordinat1, color="Red",
weight = 3,
              opacity = 0.5)
```



Grafik 85: Bağlantı Haritası Örneği

4.33. Nokta Haritası (Dot Map)

Bir coğrafi alandaki bölgeleri nokta yani küçük bir çember ile hedefleme yapılan görselleştirme türüne nokta haritası (nokta yoğunluk haritası) denilir. Nokta haritaları, verinin coğrafi dağılımını görmek ve

hangi noktalarda çakıştığını gözlemlemek açısından oldukça kullanışlıdır. Nokta büyüklükleri dikkat çekme açısından eşit büyüklükte olabileceği gibi kullanılan veri setine göre veri setindeki bir büyüklüğe göre de oransal olarak şekillendirilebilir.

Aşağıda nokta haritasına örnek verilmiştir. *leaflet* paketi kullanılarak yapılan görselleştirmede *radius* fonksiyonu ile noktaların büyüklüğü bir veriye atanmıştır.

```
iller<- c("istanbul","gaziantep","hatay","sanliurfa", "adana")
suriyelinufus <-c(519,450,434,421,252)
ilnufusunaorani<- c(3.3,21.7,26.6,20.3,11.2)
koordinat1<-c(40.89833,36.94778,36.36278,37.45000,36.98194)
koordinat2<-c(28.8141, 37.4788,36.2822,38.9000,35.2802)

suriyenuf<- data.frame(iller,suriyelinufus,ilnufusunaorani,
                       koordinat1,koordinat2)

library(leaflet)
suriyenuf %>% leaflet() %>%
  addTiles %>%
  addProviderTiles("Stamen.TonerHybrid") %>%
  addCircles(lng = ~koordinat2, lat = ~koordinat1, radius  =
suriyelinufus*100)
```

Grafik 86: Ark Grafiği Örneği

4.34. Dairesel Grafik (Circle Packing)

Önceki bölümlerde açıkladığımız dikdörtgen ağaç haritasının dairesel şekilde olanına, yani iç içe geçen dikdörtgenler yerine büyük bir çemberin içinde diğer çemberler ile hiyerarşik yapıları görselleştirilmesine dairesel grafik denmektedir.

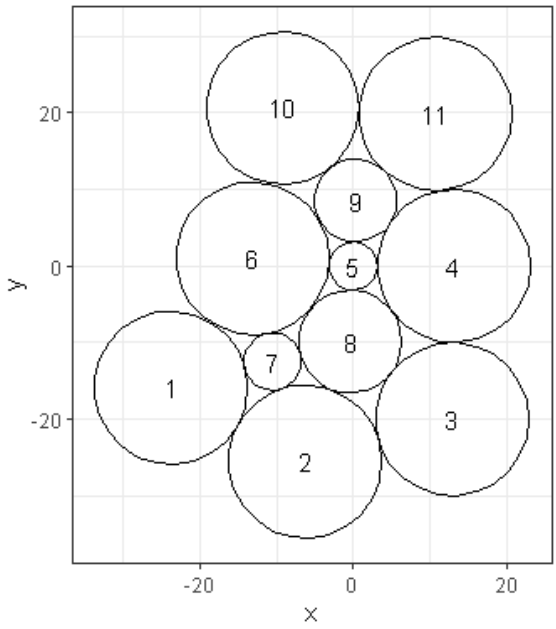
Her bir dairenin kapsama alanı o yapının hiyerarşisini o yapı içindeki diğer küçük daireler ise ağaçların dalları gibi alt hiyerarşik yapısını temsil etmektedir. Daire veya çemberlerin kapladıkları alan yine bir büyüklüğü oransal olarak temsil etmektedir. Büyük çemberin boyutunun sınırlı olması ve içindeki çemberlerin çizimi sonrasında boşlukların fazla olması karmaşık verilerde gözü yormakta ve görselleştirmeyi anlamsız kılmaya itmektedir.

Aşağıda *circlepackeR* ve *packcircles* paketleri ile görselleştirme örnekleri sunulmuştur.

Aşağıdaki örnek *https://www.r-bloggers.com/2015/07/graph-based-circle-packing-2/* adresinden esinlenilerek yaratılmıştır.

```
library(circlepackeR)
# devtools::install_github("jeromefroe/circlepackeR")
library(packcircles)
library(ggplot2)

ic_daireler <- list(
  c(9, 4, 5, 6, 10, 11),
  c(5, 4, 8, 6, 9),
  c(8, 3, 2, 7, 6, 5, 4),
  c(7, 8, 2, 1, 6)
)
## Yukarıdaki örnekte ilk vektör olan 9 ortada olan iç daireyi
simgeler. 4,5,6,10,11 ise 9 dairesinin temas ettiği dairelerdir.
dis_daireler <- data.frame(id = c(1, 2, 3, 4, 6, 10, 11), radius=10.0)
## dıs daireler, grafiği çerçeveleyen daireleri temsil etmektedir.
layout <- circleGraphLayout(ic_daireler, dis_daireler)

daire <- circleLayoutVertices(layout, xysizecols = 2:4, idcol = 1)
ggplot() +
  geom_polygon(data=daire, aes(x, y, group=id),
```

```
            fill=NA, colour="black") +
geom_text(data=layout, aes(x, y, label=id)) +
coord_equal() +
theme_bw()
```



Grafik 87: Dairesel Grafik Örneği

4.35. Piktogram Grafiği (Pictogram Chart)

Küçük verilerin ve bu verilerin anlamlarının daha ilgi çekilmesi amacı için o veriye uygun görsellerin sayısının veri sayısı kadar kullanılarak yapılan görselleş itmelere Piktogram grafikler denir. Örneğin 50 adet arabayı görselleştirmek için 5 adet araba (1 araba 10 arabayı temsil eder) görselleştirilebilir. Bu örneği daha akılda kalıcı yapmak adına 10 adet arabadan ilk 5 adedi farklı bir renkte ayrıştırılabilir.

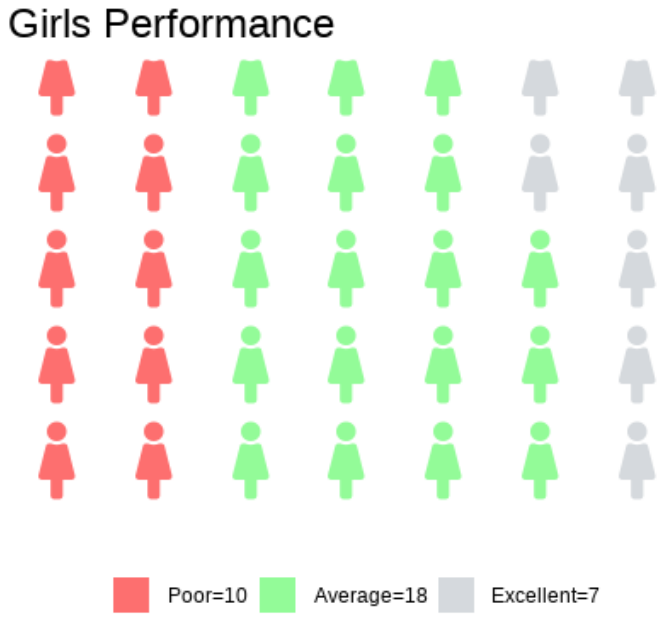
Konu ile ilgili simgelerin kullanılması hem hatırlamayı hem de dilsel ve ulusal yanlış anlamaların önüne geçebilir. Dil, eğitim, kültür gibi farklılıkların yaşanmamasını sağlar. Nüfus insan simgesi olarak, cinsiyet ise kadın ve erkek simgeleri ile görselleştirilebilir. Lakin büyük veri setlerinin görselleştirilmesinde piktogram grafiği tercih edilen bir yöntem değildir.

Aşağıda 2 adet piktogram görselleştirilmesi *waffle* paketi le yapılmış, görselleştirme simgeleri ise *FontAwesome* ikonları kullanılmıştır (Waffle, 2015).

```
library(showtext)
library(extrafont)
#font_import()
font_add(family = "FontAwesome5Free-Solid", regular =
"C:\\Users\\inanc\\AppData\\Local\\Microsoft\\Windows\\Fonts\\fa-
solid-900.ttf")
font_add(family = "FontAwesome5Free-Regular", regular =
"C:\\Users\\inanc\\AppData\\Local\\Microsoft\\Windows\\Fonts\\fa-
regular-400.ttf")
font_add(family = "FontAwesome5Brands-Regular", regular =
"C:\\Users\\inanc\\AppData\\Local\\Microsoft\\Windows\\Fonts\\fa-
brands-400.ttf")
showtext_auto()
```

Yukarıdaki şekilde fontlar bulunup sisteme yüklendikten sonra aşağıdaki şekilde görselleştirme yapılabilmektedir.

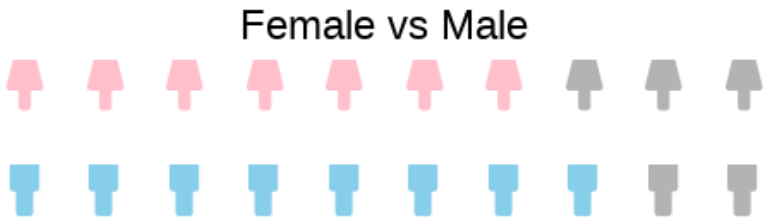
```
library(waffle)
loadfonts(device = "win")
waffle(
 c(`Poor=10` =10, `Average=18` = 18, `Excellent=7` =7), rows = 5,
 colors = c("#FD6F6F", "#93FB98", "#D5D9DD"),
 use_glyph = "female", glyph_size = 12
 ,title = 'Girls Performance', legend_pos="bottom"
)
```



Grafik 88: Piktogram Grafik Örneği

```
iron(
 waffle(
  c('TRUE' = 7, 'FALSE' = 3),
```

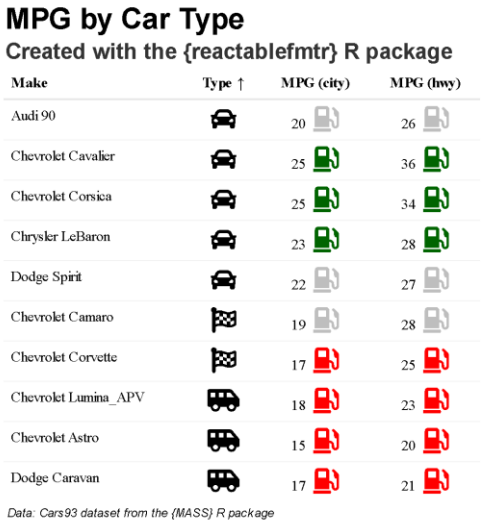
```
  colors = c("pink", "grey70"),
  use_glyph = "female",
  glyph_size = 12,
  title = "Female vs Male",
  rows = 1,
  legend_pos = "none"
) + theme(plot.title = element_text(hjust = 0.5))
,
waffle(
  c('TRUE' = 8, 'FALSE' = 2),
  colors = c("skyblue", "grey70"),
  use_glyph = "male",
  glyph_size = 12,
  rows = 1,
  legend_pos = "none"
)
)
```



Grafik 89: Piktogram Grafik Örneği

Piktogramlar için *reacttablefmtr* kütüphanesi de kullanılabilir. Pakete ait örnek reactablefmtr sayfasından direkt olarak alınmıştır.

```
library(reactablefmtr)
library(tidyverse)

car_data <- MASS::Cars93 %>%
  filter(Type %in% c("Compact", "Sporty", "Van")) %>%
  select(c("Make", "Type", "MPG.city", "MPG.highway")) %>%
  head(10)

reactable(
  car_data,
  defaultColDef = colDef(
    align = "center",
    cell = icon_sets(car_data, icon_size = 28, icons = "gas-pump", color
s = c("red", "grey", "darkgreen"))
  ))
```



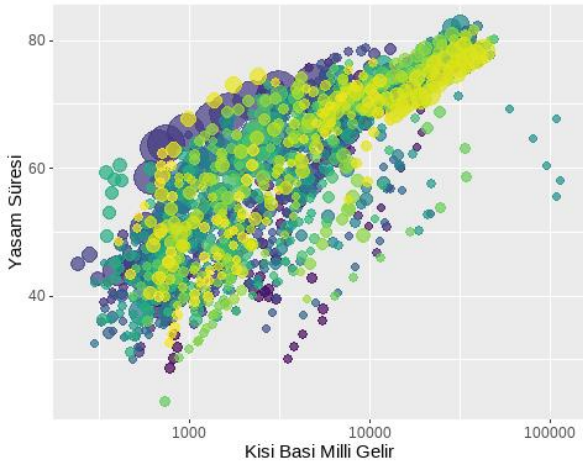
Grafik 90: Piktogram Grafik Örneği

5. R ile Veri Görselleştirmede Animasyon

gganimate paketi Thomas Lin Pedersen ve David Robinson tarafından geliştirilmiş bir pakettir. *gganimate* paketi sayesinde yapılan grafiklerin animasyona ve *gif* formatından kaydedilmesi mümkündür (Pedersen ve Robinson, 2019).

```
grafik<- ggplot(
  gapminder,
  aes(x = kisi_basi_milli_gelir, y=yasam_suresi, size = nufus, colour =
ülke)
  ) +
  geom_point(show.legend = FALSE, alpha = 0.7) +
```

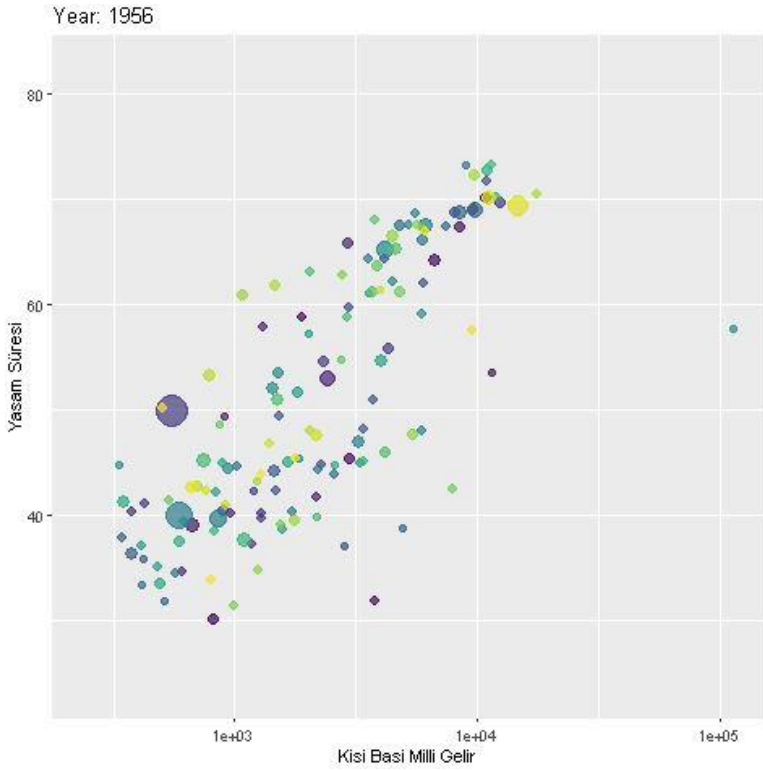
```
  scale_color_viridis_d() +
  scale_size(range = c(2, 12)) +
  scale_x_log10() +
  labs(x = "Kişi Başı Milli Gelir", y = "Yaşam Süresi")
grafik
```



Grafik 91: Saçılım Grafik Örneği

Yukarıdaki grafiğin yıllara göre animasyonu için aşağıdaki *gganimate* paketi kullanılmıştır.

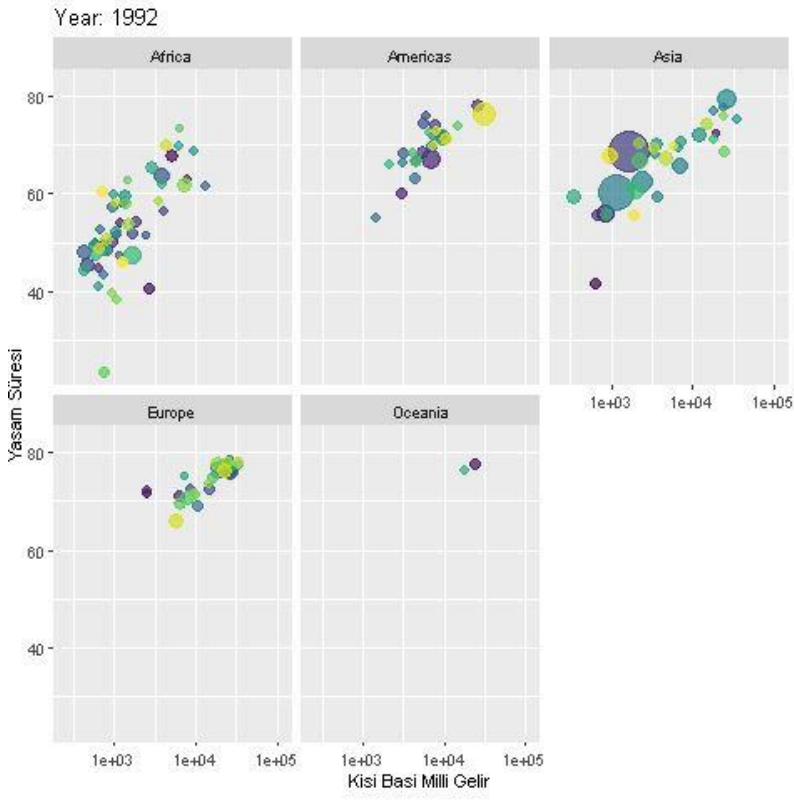
```
library(gganimate)
gr + transition_time(yil) +
  labs(title = "Year: {frame_time}")
```



Grafik 92: Saçılım Grafik Animasyonu

Yukarıdaki grafiğin kıtalara göre kırılımlı şeklinin animasyonu ise aşağıda sunulmuştur.

```
start_pause <- 0
end_pause <- 0
gr + facet_wrap(~kita) +
  transition_time(yil) +
  labs(title = "Year: {frame_time}")
```



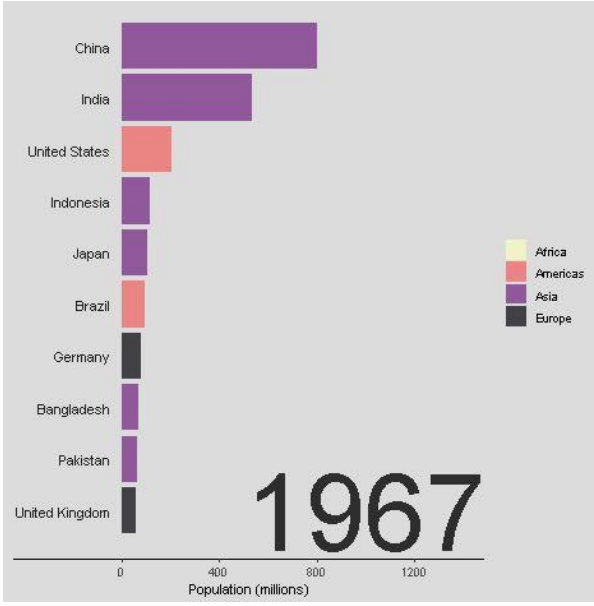
Grafik 93: Saçılım Grafik Animasyonu

Son yıllarda birbirleri ile yarışan çubuk grafik görselleştirmesi oldukça tercih edilen veri görselleştirme örneği olarak karşımıza çıkmaktadır. Yukarıda daha önceden kullandığımız *gapminder* veri setimiz ile bu tarz görselleştirme örneği aşağıda sunulmuştur. Aşağıdaki kodlar "Racing Barchart with gganimate" adlı çalışmadan direkt alıntılarak verilmiştir (Reynolds, 2020).

```
gapminder %>%
  select(ülke, nufus, yil, kita) %>%
  # her sene için bir sıra tanımlanacaktır
```

```
  group_by(yil) %>%
  arrange(yil, -nufus) %>%
  # sıra numarası değerlere göre tanımlanır
  mutate(rank = 1:n()) %>%
  filter(rank <= 10) ->  ranked_by_year

my_theme <- theme_classic(base_family = "Times") +
  theme(axis.text.y = element_blank()) +
  theme(axis.ticks.y = element_blank()) +
  theme(axis.line.y = element_blank()) +
  theme(legend.background = element_rect(fill = "gainsboro")) +
  theme(plot.background = element_rect(fill = "gainsboro")) +
  theme(panel.background = element_rect(fill = "gainsboro"))

ranked_by_year %>%
  ggplot() +
  aes(xmin = 0 ,
      xmax = nufus / 1000000) +
  aes(ymin = rank - .45,
      ymax = rank + .45,
      y = rank) +
  facet_wrap(~ yil) +
  geom_rect(alpha = .7) +
  aes(fill = kita) +
  scale_fill_viridis_d(option = "magma",
                       direction = -1) +
```

```
  scale_x_continuous(
    limits = c(-800, 1400),
    breaks = c(0, 400, 800, 1200)) +
  geom_text(col = "gray13",
            hjust = "right",
            aes(label = ü lke),
            x = -50) +
  scale_y_reverse() +
  labs(fill = NULL) +
  labs(x = 'Nüfus (millions)') +
  labs(y = "") +
  my_theme ->
my_plot

my_plot +
  facet_null() +
  scale_x_continuous(
    limits = c(-355, 1400),
    breaks = c(0, 400, 800, 1200)) +
  geom_text(x = 1000 , y = -10,
            family = "Times",
            aes(label = as.character(yil)),
            size = 30, col = "grey18") +
  aes(group = ülke) +
  gganimate::transition_time(yil)
```



Grafik 94: Birbirleri ile Yarışan Çubuk Grafikler

Aşağıda oluşturulan kod yardımı ile hazırlanan animasyon *gif* veya *mp4* formatında kaydedilebilir.

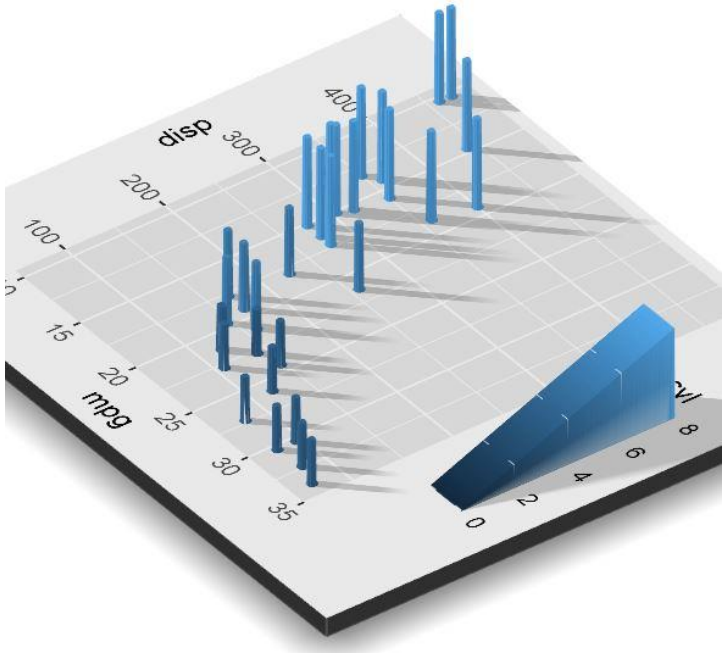
```
animate(animasyon, 200, fps = 20,  width = 1200, height = 1000,
        renderer = gifski_renderer("gganim.gif"))

animate(animasyon, 200, fps = 20,  width = 1200, height = 1000,
        renderer = ffmpeg_renderer()) ->
for_mp4anim_save("animation.mp4",
                 animation = for_mp4 )
```

6. R ile Veri Görselleştirmede 3D uygulamalar

R ile elde edilen grafiklerin üç boyutlu modellemesi ve sunumu mümkündür. R ile elde edilen grafiklerinin 3D gösterimi mümkündür. Tyler-Morgan-Wall tarafından geliştirilen *rayshader* paketi ile verilerimizi 3D olarak görselleştirebiliriz (Morgan-Wall, 2020).

rayshader (Tyler, 2018) ile temel veri görselleştirme örneği için 1973-1974 yıllarına ait 32 araba modelinin özelliklerinin bulunduğu *mtcars* veri seti kullanılmıştır. Veri seti içerisinde 11 özellik bulunmaktadır. Aşağıda verilen örnekte yakıt tüketimi yani mil/galon (mpg), yer değiştirme (disp) ve silindir sayısı (cyl) kolonları kullanılmıştır.

```
names(mtcars)
## [1] "mpg" "cyl" "disp" "hp"  "drat" "wt"   "qsec" "vs"   "am"
"gear"
## [11] "carb"
library(rayshader)
#install.packages("remotes")
#remotes::install_github("thomasp85/ggfx")
library(ggfx)
mtplot = ggplot(mtcars) +
 geom_point(aes(x=mpg,y=disp,color=cyl)) +
 scale_color_continuous(limits=c(0,8))
#plot_gg(mtplot, width=3.5, sunangle=225, preview = TRUE)
plot_gg(mtplot)
```

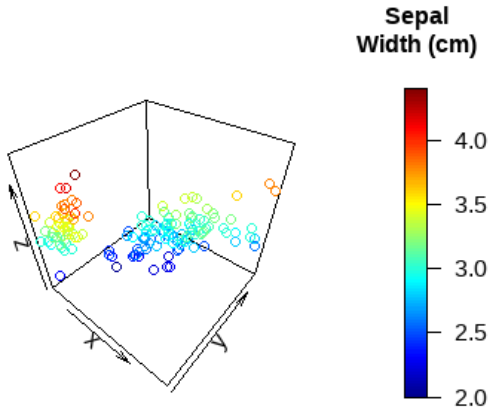


Grafik 95: 3D Grafik Görselleştirme Örneği

Yukarıdaki paketlere ek olarak *rgl* paketi (Adler et al. 2014), *plot3D* (Soetaert, 2021) paketi ve *plotly* paketi de bulunmaktadır. Çok bilindik *İris* veri seti ve *rgl* paketi ile örnek görselleştirmeler aşağıda yer verilmiştir.

```
library(rgl)
library(plot3D)
data(iris)
head(iris)
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
```

```
## 3        4.7         3.2          1.3          0.2  setosa
## 4        4.6         3.1          1.5          0.2  setosa
## 5        5.0         3.6          1.4          0.2  setosa
## 6        5.4         3.9          1.7          0.4  setosa
x <- sep.l <- iris$Sepal.Length
y <- pet.l <- iris$Petal.Length
z <- sep.w <- iris$Sepal.Width
# Plot çizimi
scatter3D(x, y, z, clab = c("Sepal", "Width (cm)"))
```

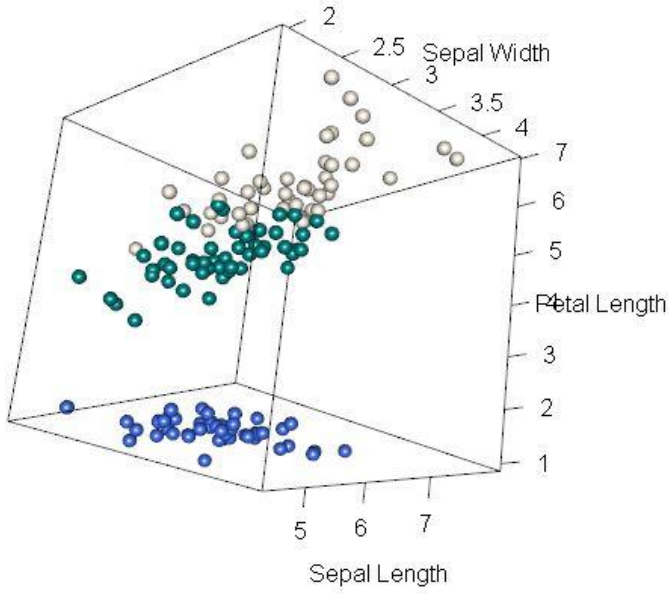


Grafik 96: 3D Grafik Görselleştirme Örneği

Plot3D paketi ile yapılan görselleştirme örneği aşağıda verilmiştir.

```
library(plot3D)
data <- iris
mycolors <- c('royalblue1', 'darkcyan', 'oldlace')
data$color <- mycolors[ as.numeric(data$Species) ]
```

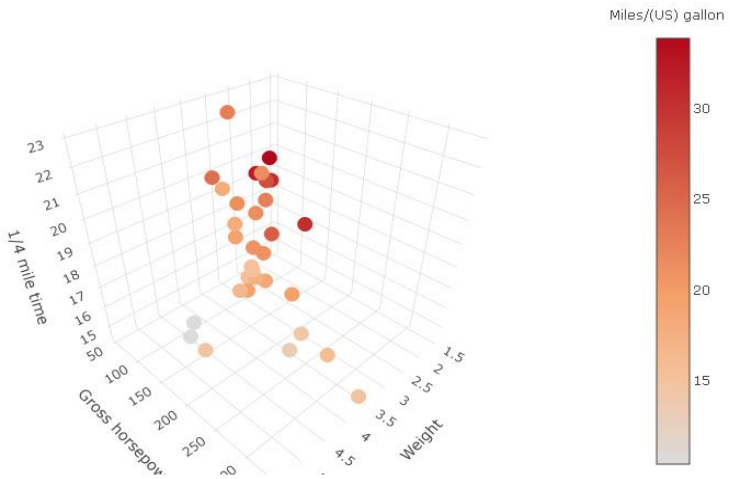
```
# Plot çizimi
plot3d(
  x=data$`Sepal.Length`, y=data$`Sepal.Width`,
z=data$`Petal.Length`,
  col = data$color,
  type = 's',
  radius = .1,
  xlab="Sepal Length", ylab="Sepal Width", zlab="Petal Length")
```



Grafik 97: 3D Grafik Görselleştirme Örneği

Plotly paketi ile yapılan görselleştirme aşağıda sunulmuştur.

```
library(plotly)
fig <- plot_ly(mtcars, x = ~wt, y = ~hp, z = ~qsec,
               marker = list(color = ~mpg, colorscale = c('#FFE1A1',
'#683531'), showscale = TRUE))

fig <- fig %>% add_markers()
fig <- fig %>% layout(scene = list(xaxis = list(title = 'Weight'),
                                   yaxis = list(title = 'Gross horsepower'),
                                   zaxis = list(title = '1/4 mile time')),
                                    annotations = list(x = 1.13,y = 1.05,text =
'Miles/(US) gallon',
                      xref = 'paper',yref = 'paper',showarrow = FALSE))
fig
```



Grafik 98: 3D Grafik Görselleştirme Örneği

7. SONUÇ

Bilginin elde edilmesi, anlamdırılması ve analiz edilmesi kadar elde edilen sonuçların karşı tarafa aktarılması gerekmektedir. Elde edilen sonuçların, yani bir anlamda iletişimin özünü oluşturan *mesajın* basit, anlaşılabilir ve akılda kalıcı olması hedeflenmektedir.

Mesajın anlaşılabilir olmasında, büyük veri setlerinden elde edilen bilgilerin basitleştirilmesinde, varolan değişkenler arasındaki ilişkinin tespitinde "veri görselleştirme" çok önemli bir yer tutmaktadır.

Bu kitap ile veri görselleştirmenin iletişim bağlamında irdelenmesi yapılmış, orta ve ileri seviyede veri görselleştirme türleri tanıtılmış ve her biri örneklendirilmiştir.

Excel programı ile veri görselleştirme bir şekilde temel düzeyde yapılabilse de ileri veri görselleştime yapabilmek için bazı ek yazılımlara veya programlama dillerine ihtiyaç bulunmaktadır. Bu kitapta veri görselleştirme örnekleri R dili ve R Studio aracılığı ile yapılmıştır. Bu kitaptan istenilen verimin alınabilmesi temel R dili bilgisi ile mümkündür.

Bu kitap birçok farklı örneği içinde barındırsa da, örnekler asıl olarak Severino Ribecca tarafından oluşturulan http://datavizcatalogue.com adresindeki sistematik içinde verilmiştir. Veri Görselleştirme Kataloğu adı altında editörlüğünü Öğr. Gör. Pınar Dağ'ın yaptığı toplam 8 kişilik bir ekip ile Türkçe 'ye de çevrilen çalışma kitabımızın ana çerçevesini

oluşturmaktadır. Bunun yanında farklı birçok uygulama ve örnek ile kitap zenginleştirilmiştir.

Kitap içinde bilgi görselleştirme çalışmalarına yer verilmemiştir. Bir sonraki baskıda bu bölümün de kitaba dahil edilmesi planlanmaktadır.

KAYNAKÇA

Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis* 16 (1), 3–9.

Adler, D., Murdoch, M. D., Suggests, M. A. S. S., WebGL, P. L. Y., OBJ, S. ve OpenGL, S. (2014). Package 'rgl'.

Ahsan, S. ve Shah, A. (2006). Data, information, knowledge, wisdom: A doubly linked chain." In *The Proceedings of the 2006 International Conference on Information Knowledge Engineering*, 270–78. Citeseer.

Arkeofili. (2019). Sümerliler tarafından geliştirilen çivi yazısı. https://arkeofili.com/sumerlerin-dunyayi-degistiren-9-icadi/.

Aydın, H., Mutlu, E., Usta, İ., Aydın, S., Keskin, N., Çalışkan, H., Güneş, E. ve Koçdar, S. (2015). *Temel Bilgi Teknolojileri I*. Anadolu Üniversitesi.

Barutçugil, İ. (2002). *Bilgi yönetimi*. Kariyer Yayıncılık.

Becer, E. (1997). İletişim ve grafik tasarım. *Ankara: Dost Kitabevi Yayınları*.

Bellinger, G., Durval C. ve Anthony M. (2004). Data, information, knowledge, and wisdom.

Berchtold, S., Keim, D. A. ve Kriegel, H. P. (1996). The X-tree: An index structure for high-dimensional data. In *Very large data-bases* (pp. 28-39).

Bostock, M., Rodden, K., Warne, K. ve Russell, K. (2020). SunburstR: Sunburst 'Htmlwidget'. *URL: https://CRAN. R-project. org/package= sunburstR. R package version*, *2*(5).

Brunson, J. C. (2020). Ggalluvial: Layered grammar for Alluvial Plots. *Journal of Open Source Software* 5 (49): 2017.

Bryan, J. (2015). Gapminder: Data from Gapminder. *R Package Version 0.2. 0. https://CRAN.R-Project.Org/Packages/ Gapminder*.

Budge, T. (2015). Finding the centre claudius ptolemy and the modern development ındustry ınstitute for human security and social change blog." https://practiceforchange.org/2015/05/27/finding-the-centre-claudius-ptolemy-and-the-modern-development-industry/comment-page-1/.

Cairo, A. (2012). *The Functional Art: An Introduction to information graphics and visualization*. New Riders.

Card, S. K., Mackinlay, J. D. ve Shneiderman, B. (1999). Information visualization. *Readings in information visualization: using vision to think*, 1-34.

Chen, C., Wolfgang K. H. ve Antony U. (2007). *Handbook of data visualization*. Springer Science & Business Media.

Cleveland, H. (1982). Information as a resource. *Futurist* 16 (6): 34–39.

Cristina, M., Oliveira, D.E. ve Levkowitz, H. (2003). From visual data exploration to visual data mining: a survey. *IEEE Transactions on Visualization and Computer Graphics* 9 (3): 378–94.

Coder, R. (2021, September 12). *Sankey diagrams in ggplot2 with ggsankey*. R CHARTS | A Collection of Charts and Graphs Made with the R Programming Language. Retrieved November 14, 2021, from https://r-charts.com/flow/sankey-diagram-ggplot2/

Dur, B. I. U. (2014). Data visualization and infographics in visual communication design education at the age of ınformation. *Journal of Arts and Humanities* 3 (5): 39–50.

Durmuş, D. A. (2019). Etkileşimli infografik: Türkiye bağımsız sinema tarihi örneği. PhD thesis, Hacettepe Üniversitesi, Güzel Sanatlar Enstitüsü.

Eğitimde Bilim Tarihi. (2020). Eğitimde Bilim Tarihi. https://www.egitimdebilimtarihi.org/2018/06/11/grafik-cizimi/.

Ertan, G. ve Sansarcı, E. (2016). *Görsel sanatlarda anlam ve algı*. Hat Baskı Sanatları.

Faloutsos, C. (2012). *Searching multimedia databases by content*. Vol. 3. Springer Science & Business Media.

Feinerer, I. ve Hornik, K. (2015). Package 'Tm'." *Corpus* 10 (1).

Few, S. (2006). Rich data, poor data: Designing dashboards to inform. *Noetix Corporation*.

Frické, M. (2009). The knowledge pyramid: A critique of the DIKW Hierarchy." *Journal of Information Science* 35 (2): 131–42.

Friendly, M. (2008). A brief history of data visualization. In *Handbook of Data Visualization*, 15–56. Springer.

Github. (2021). Github giocomaiganttrify create beautiful gantt charts with ggplot2. https://github.com/giocomai/ganttrify.

Gu, Z. ve Imports GlobalOptions. (2021). Package 'Circlize'.

Günay, V. ve Parsa, A. (2012). *Görsel göstergebilim: Imgenin anlamlandırılması*. Es yayınları.

Gürler, A., Yılmaz, A. S. ve Tekerek, M. (2018). Veri görselleştirme ve infografikler. *Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi* 21 (2): 131–48.

Harita Müzesi. (2015). Çatalhöyük Haritası. https://www.haritamuzesi.com/catal-hoyuk-haritasi/.

Iannone, R. (2020). Package 'DiagrammeR'.

Ihaka, R. ve Gentleman, R. 1996. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5 (3): 299–314.

Inglis-Arkell, E. (2015). This human bar graph shows how we've grown taller over 80 years. io9. https://io9.gizmodo.com/this-human-bar-graph-shows-how-weve-grown-taller-over-8-5899665.

Inselberg, A. ve Dimsdale, B. (1990). Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proceedings of the First IEEE Conference on Visualization: Visualization90*, 361–78. IEEE.

Işıklı, Ş. (2014). Büyük veri, epistemoloji ve etik tartışmalar. *AJIT-e: Bilişim Teknolojileri Online Dergisi* 5 (17): 89–122.

Journaldev.com. (2020, May 10). *How to create an area plot in R using ggplot2*. Hive News. Retrieved November 14, 2021, from https://hive.news/news/how-to-create-an-area-plot-in-r-using-ggplot2?uid=49355.

Keim, D. A. (1997). Visual techniques for exploring databases. In *Knowledge Discovery in Databases (KDD'97)*.

Kim, T. ve DiSalvo, C. (2010). Speculative visualization: A new rhetoric for communicating public concerns.

Kurtaran, Ö. M. ve Çubukçu, F. (1991). *Ansiklopedik Bilgi işlem Terimleri sözlüğü*. Türkmen Kitabevi.

Leaflet. (2015). Leafletpdf. https://cran.rproject.org/web/packages /leaflet/leaflet.pdf.

Forbes. (2021). Leonardo Da Vincis Scientific Studies 500 Years Later. https://www.forbes.com/sites/evaamsen/2019/05/02/leonardo-da-vincis-scientific-studies-500-years-later/?sh=1a942c887f46.

Marshall, C. (2016). Joseph Priestley visualizes history great historical figures with two of the most influential infographics ever 1769 open culture." https://www.openculture.com/2016/07/joseph-priestley-visualizes-history-with-two-of-the-most-influential-infographics-ever-1769.html.

Morgan-Wall, T. (2020). Create maps and visualize data in 2d and 3d rayshader. https://www.rayshader.com/.

Morriss, B. (2009). A brief history of big data everyone should read. http://bobmorris.biz/a-brief-history-of-big-data-everyone-should-read.

Nakazawa, M. (2019). Package 'Pyramid'.

Neuwirth, E. ve Neuwirth, M. E. (2011). *Package 'RColorBrewer'*. CRAN 2011-06-17 08: 34: 00. Apache License 2.0.

Omran, A. (2019). Recommendation of data visualization tools for non-technical people.

Öğüt, A. (2012). *Bilgi çağında yönetim*. Nobel Akademik Yayıncılık.

Pearson, K. (1895). X. Contributions to the mathematical theory of evolution. Skew variation in homogeneous material. *Philosophical Transactions of the Royal Society of London. (A.)*, no. 186: 343–414.

Pedersen, T. L. ve Robinson, D. (2019). A grammar of animated graphics gganimate. https://gganimate.com/index.html.

Pickett, R. M. ve Georges G. G. (1988). Iconographic displays for visualizing multidimensional data. In *Proceedings of the 1988 IEEE Conference on Systems, Man, and Cybernetics*, 514:519.

Playfair, W. (1801). *The statistical breviary*. Wallis.

Raabe, S. (2020). Pretty timelines in r vistime." https://shosaco.github .io/vistime/.

Reynolds, G. A. (2020). Racing Barchart with gganimate. Https://Evamaerey.Github.Io/. Retrieved November 14, 2021, from https://evamaerey.github.io/little_flipbooks_library/racing_bars/racing_barcharts.html#1

R-graph. (2021). Introduction to Tree Diagram with r and Ggraph the r Graph Gallery.https://www.r-graph-gallery.com/334-basic-dendrogram-with-ggraph.html.

R-graph. (2021b). Add Labels to Circular Barplot – the r Graph Gallery."https://www.r-graph-gallery.com/296-add-labels-to-circular-barplot.html.

Sanchez, G. (2016). Introduction to the r package arcdiagram. Citeseer.

Shedroff, N. ve diğ. (1999). Information interaction design: A unified field theory of design. *Information Design*, 267–92.

Sievert, C. (2020). *Interactive web-based data visualization with r, plotly, and shiny*. Chapman; Hall/CRC. https://plotly-r.com.

Smiciklas, M. (2012). *The power of infographics: Using pictures to communicate and connect with your audiences*. Que Publishing.

Soetaert, K. (2014). plot3D: Tools for plotting 3-D and 2-D data. *R package version*, 10-2.

Stephen, F. (2013). Data visualization for human perception. *Interaction Design Foundation, The Encyclopedia of Human-Computer Interaction* 35.

Treemap (GGlot2). (2020). RPubs. Retrieved November 14, 2021, from https://rpubs.com/TX-YXL/653903

Tokdemir, G. ve Nergiz E. C. (2010). A concept map approach for introduction to computer engineering course curriculum. In *IEEE EDUCON 2010 Conference*, 243–50. IEEE.

Tufte, E. R. (2006). *Beautiful evidence*. Vol. 1. Graphics Press Cheshire, CT.

Tufte, E. R. ve Peter R G. (1983). *The visual display of quantitative information*. Vol. 2. 9. Graphics Press Cheshire, CT.

Tyler, M. (2018). GitHub Tyler Morganwall rayshader r package for 2d and 3d mapping and data visualization." https://github.com /tylermorganwall/rayshader.

Vardar, B. ve Nüket G. (1988). *Açıklamalı dilbilim terimleri sözlüğü*. Vol. 1. ABC Kitabevi.

Waffle. (2015). Wafflepdf. https://cran.r project.org/web/packages/ waffle/waffle.pdf.

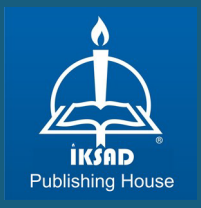
Ward, M. O., Georges G., ve Daniel, K. (2010). *Interactive data visualization: Foundations, techniques, and applications*. CRC press.

Wickham, H.ve Wickham, M. H. (2007). The ggplot package. *URL: https://cran. r-project. org/web/packages/ggplot2/index.*

Wickham, H. (2017). Package 'Tidyverse'. Easily Install and Load the 'Tidyverse.

Wikipedia. (2012). İran Teimareh bölgesinde kaya sanatı petroglif vikipedi." https://tr.wikipedia.org/wiki/Petroglif#/media/Dosya:Golpayegan.petroglyphs0101.jpg.

Zeleny, M. (1987). Management support systems: Towards integrated knowledge Management, *Human Systems Management* 7 (1): 59–70.



ISBN: 978-625-8007-70-1