



HACETTEPE ÜNİVERSİTESİ

İstatiksel

Analiz



MEHMET ŞENEL
2210329077

İSTATİKSEL ANALİZ ÖDEVİ

İçindekiler

1.0 Kullanılan Paketler	2
1.1 Dbplyr	2
1.2 Ggiraph	2
1.3 E1071	2
1.4 Shiny	2
2.0 Veri Hakkında	3
2.1 Verinin Rstudioya aktarımı	3
2.2 Verinin görünümü	4
2.3 Veri Hakkında	4
3.0 Örneklem Oluşturma	5
3.1 Örneklem Oluşturma	5
3.2 Örneklem Çıktısı	5
4.0 Verinin Analizi	7
4.1 Temel Analizler	7
4.2 Creating One Scatter Plot for Each Group	8
4.2.1 Kullanılan Paketler	8
4.2.2 Grafiğin Oluşturulması	8
4.3 Creating One Boxplot for Each Factor Level	10
4.3.1 Kullanılan Paketler	10
4.3.2 Grafiğin Oluşturulması	11
4.3.3 Grafik Çıktıları ve Yorumlama	12
4.4 Creating a Histogram	13
4.4.1 Kullanılan Paketler	13
4.4.2 Grafiklerin Oluşturulması	13
4.4.3 Grafik Çıktıları ve Yorumlama	14
4.4.4 Adding a Density	14
.....	14
5.0 Changing the Type, Width, or Color of a Line	17
5.1 Line Type	17
5.2 Change width and color	18
6.0 Writing Your Plot to a File	19
6.1 Uygulamadan Yazdırma	19
6.2 Kod ile Yazdırma	20
Kaynakça	20

1.0 Kullanılan Paketler

```
11 library(shiny)
12 library(ggiraph)
13 library(e1071)
14 library(dbplyr)
```

Yapıcak olduğumuz ödevimizde sunumuzun tamamı R Studio programı ile hazırlanacak olup kullanılan temel paketler ve kütüphaneler görüldüğü gibidir.

1.1 Dbplyr

Dbplyr'i, sözlük anlamıyla "veri işlemenin temel prensipleri" veya "veri işleme grameri" olarak çevirebiliriz. Hadley Wickham tarafından geliştirilen R'da sıklıkla kullanılan paketlerin koleksiyonundan oluşan [tidyverse](https://www.tidyverse.org/) evreninin bir parçası.

Dbplyr'a gramer denmesinin sebebi, veri setlerinin düzenlenmesi, filtrelenmesi, sıralanması ve belirli değişkenlerin hesaplanmasını mümkün kılan 6 ayrı fiile (fonksiyona) sahip olması. Bunlar: select (), mutate (), filter (), arrange (), summarize () ve group_by (). R programlama dilini öğrenmek bu fiilleri kullanmaktan geçiyor.

1.2 Ggiraph

Ggiraph, dinamik ggplot grafikleri oluşturmanıza olanak tanıyan bir araçtır. Bu, grafiklere araç ipuçları, vurgulu efektler ve JavaScript eylemleri eklemenizi sağlar. Paket ayrıca parlak uygulamalarda kullanıldığında grafik öğelerin seçilmesine de olanak tanır.

1.3 E1071

SVM'lerin hızlı ve kolay uygulanmasını sağlar. Doğrusal, polinom, RBF ve sigmoid dahil olmak üzere en yaygın çekirdekleri sağlar. Tahminler için karar ve olasılık değerleri için hesaplama gücü sunar. Ayrıca sınıflandırma modunda ve çapraz doğrulamada sınıfların ağırlıklandırılmasını sağlar.

1.4 Shiny

Shiny son yıllarda kullanımı oldukça yaygınlaşan açık kaynaklı bir R paketidir. Shiny ile R programlama dili kullanarak web tabanlı ara yüzlerin oluşturulması sağlanır. Bu sayede HTML, Java Script gibi programlama dillerini bilmeseniz bile analizlerinizi interaktif bir web uygulaması üzerinden gerçekleştirebilirsiniz. İlk versiyonu 28 Şubat 2011 yılında çıkan R shiny Winston Chang önderliğinde bir grup tarafından geliştirilmiştir. Winston'un yanı sıra Joe Cheng, JJ Allaire, Yihui Xie ve Jonathan McPherson R shiny geliştirme sürecinde önemli katkılar sunmuşlardır. Paketin güncelleme ve yeni versiyonlarının geliştirilmesi sürecinde ise pek çok kişinin katkısı vardır ve halen devam etmektedir.

Import Dataset

Name

train

Encoding

Automatic

Heading

☒ Yes ☐ No

Row names

Automatic

Separator

Comma

Decimal

Period

Quote

Double (")

Comment

None

na.strings

NA

☐ Strings as factors

Input File


id,age,height(cm),weight(kg),waist(cm),eyesight(left),eyesight(right)
0,55,165,60,81.0,0.5,0.6,1,1,135,87,94,172,300,40,75,16.5,1.2
1,70,165,65,89.0,0.6,0.7,2,2,146,83,147,194,55,57,126,16.2
2,20,170,75,81.0,0.4,0.5,1,1,118,75,79,178,197,45,93,17.4
3,35,180,95,105.0,1.5,1.2,1,1,131,88,91,180,203,38,102,15.1
4,30,165,60,80.5,1.5,1.0,1,1,121,76,91,155,87,44,93,15.4
5,50,170,55,51.0,1.2,1.2,1,1,146,95,101,199,343,31,99,15.9
6,45,160,55,69.0,1.5,1.2,1,1,150,88,84,222,153,69,122,13.0
7,55,155,60,84.5,0.7,0.9,1,1,137,91,100,282,165,51,198,14.1
8,40,165,70,89.0,0.7,1.0,1,1,130,80,104,243,163,59,150,15.1
9,40,155,50,73.0,1.5,1.5,1,1,105,70,64,183,27,55,122,13.2
10,40,155,50,68.0,0.8,0.6,1,1,107,60,71,165,76,48,101,12.9
11,50,160,45,64.0,1.2,1.5,1,1,105,73,85,217,50,81,126,13.6
12,40,155,55,80.0,0.8,0.8,1,1,95,65,84,163,94,89,56,12.6
13,40,170,65,81.0,1.2,1.0,1,1,120,76,106,215,84,54,145,15.1
14,50,175,90,103.0,0.8,0.5,1,1,113,72,90,220,145,60,131,16
15,45,170,65,81.8,0.7,0.7,1,1,111,84,102,204,180,52,117,18

Data Frame

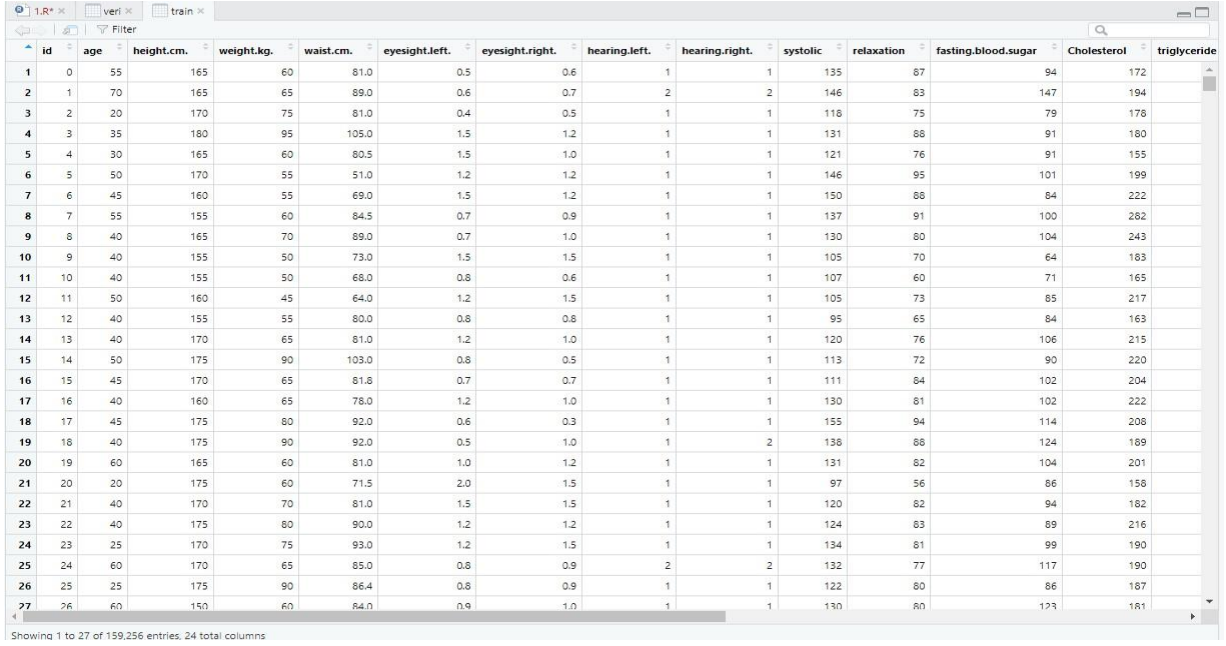
id	age	height.cm.	weight.kg.	waist.cm.	eyesight.le
0	55	165	60	81.0	0.5
1	70	165	65	89.0	0.6
2	20	170	75	81.0	0.4
3	35	180	95	105.0	1.5
4	30	165	60	80.5	1.5
5	50	170	55	51.0	1.2
6	45	160	55	69.0	1.5
7	55	155	60	84.5	0.7
8	40	165	70	89.0	0.7
9	40	155	50	73.0	1.5
10	40	155	50	68.0	0.8
11	50	160	45	64.0	1.2
12	40	155	55	80.0	0.8
13	40	170	65	81.0	1.2
14	50	175	90	103.0	0.8
15	45	170	65	81.8	0.7

Import

Cancel

```
1:1 (Top Level) ⚙  
Console Terminal × Background Jobs ×  
R 4.3.2 · ~/   
> train <- read.csv("C:/Users/pc/Desktop/analiz ödev/train.csv")  
> view(train)  
> |
```

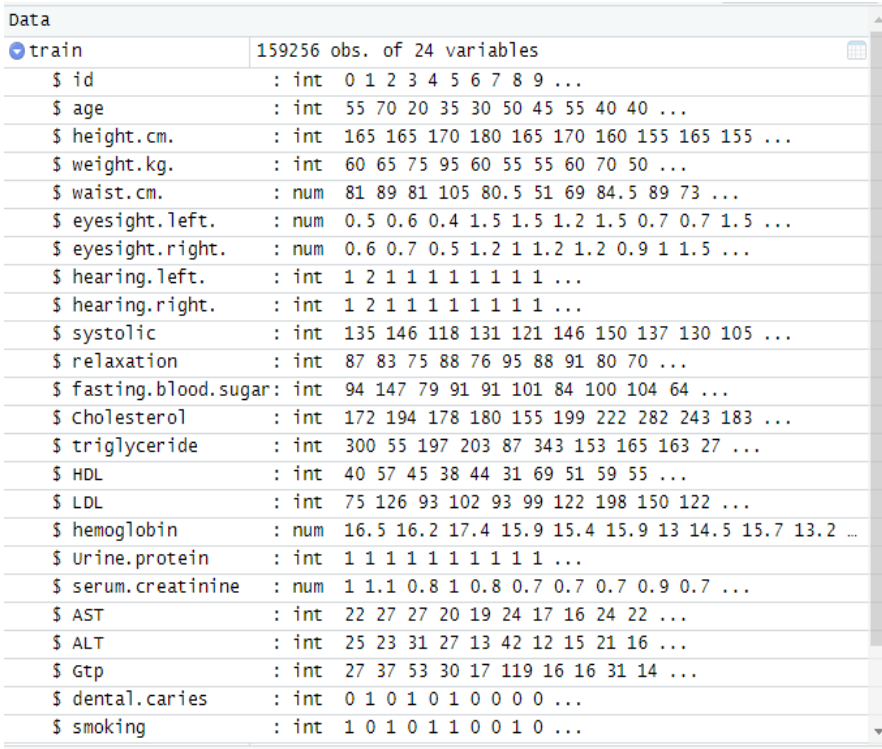
2.2 Verinin görünümü



	id	age	height.cm	weight.kg	waist.cm	eyesight.left	eyesight.right	hearing.left	hearing.right	systolic	relaxation	fasting.blood.sugar	Cholesterol	triglyceride
1	0	55	165	60	81.0	0.5	0.6	1	1	135	87	94	172	
2	1	70	165	65	89.0	0.6	0.7	2	2	146	83	147	194	
3	2	20	170	75	81.0	0.4	0.5	1	1	118	75	79	178	
4	3	35	180	95	105.0	1.5	1.2	1	1	131	88	91	180	
5	4	30	165	60	80.5	1.5	1.0	1	1	121	76	91	155	
6	5	50	170	55	51.0	1.2	1.2	1	1	146	95	101	199	
7	6	45	160	55	69.0	1.5	1.2	1	1	150	88	84	222	
8	7	55	155	60	84.5	0.7	0.9	1	1	137	91	100	282	
9	8	40	165	70	89.0	0.7	1.0	1	1	130	80	104	243	
10	9	40	155	50	73.0	1.5	1.5	1	1	105	70	64	183	
11	10	40	155	50	68.0	0.8	0.6	1	1	107	60	71	165	
12	11	50	160	45	64.0	1.2	1.5	1	1	105	73	85	217	
13	12	40	155	55	80.0	0.8	0.8	1	1	95	65	84	163	
14	13	40	170	65	81.0	1.2	1.0	1	1	120	76	106	215	
15	14	50	175	90	103.0	0.8	0.5	1	1	113	72	90	220	
16	15	45	170	65	81.8	0.7	0.7	1	1	111	84	102	204	
17	16	40	160	65	78.0	1.2	1.0	1	1	130	81	102	222	
18	17	45	175	80	92.0	0.6	0.3	1	1	155	94	114	208	
19	18	40	175	90	92.0	0.5	1.0	1	2	138	88	124	189	
20	19	60	165	60	81.0	1.0	1.2	1	1	131	82	104	201	
21	20	20	175	60	71.5	2.0	1.5	1	1	97	56	86	158	
22	21	40	170	70	81.0	1.5	1.5	1	1	120	82	94	182	
23	22	40	175	80	90.0	1.2	1.2	1	1	124	83	89	216	
24	23	25	170	75	93.0	1.2	1.5	1	1	134	81	99	190	
25	24	60	170	65	85.0	0.8	0.9	2	2	132	77	117	190	
26	25	25	175	90	86.4	0.8	0.9	1	1	122	80	86	187	
27	26	60	150	60	84.0	0.9	1.0	1	1	130	80	123	181	

Veri yukarıda belirtildiği şekilde uygulamaya aktarıldıktan sonra çıktısı yukarıdaki gibi olmaktadır.

2.3 Veri Hakkında



Data	
train	159256 obs. of 24 variables
\$ id	: int 0 1 2 3 4 5 6 7 8 9 ...
\$ age	: int 55 70 20 35 30 50 45 55 40 40 ...
\$ height.cm.	: int 165 165 170 180 165 170 160 155 165 155 ...
\$ weight.kg.	: int 60 65 75 95 60 55 55 60 70 50 ...
\$ waist.cm.	: num 81 89 81 105 80.5 51 69 84.5 89 73 ...
\$ eyesight.left.	: num 0.5 0.6 0.4 1.5 1.5 1.2 1.5 0.7 0.7 1.5 ...
\$ eyesight.right.	: num 0.6 0.7 0.5 1.2 1.2 1.2 1.2 0.9 1 1.5 ...
\$ hearing.left.	: int 1 2 1 1 1 1 1 1 1 1 ...
\$ hearing.right.	: int 1 2 1 1 1 1 1 1 1 1 ...
\$ systolic	: int 135 146 118 131 121 146 150 137 130 105 ...
\$ relaxation	: int 87 83 75 88 76 95 88 91 80 70 ...
\$ fasting.blood.sugar	: int 94 147 79 91 91 101 84 100 104 64 ...
\$ cholesterol	: int 172 194 178 180 155 199 222 282 243 183 ...
\$ triglyceride	: int 300 55 197 203 87 343 153 165 163 27 ...
\$ HDL	: int 40 57 45 38 44 31 69 51 59 55 ...
\$ LDL	: int 75 126 93 102 93 99 122 198 150 122 ...
\$ hemoglobin	: num 16.5 16.2 17.4 15.9 15.4 15.9 13 14.5 15.7 13.2 ...
\$ Urine.protein	: int 1 1 1 1 1 1 1 1 1 1 ...
\$ serum.creatinine	: num 1 1.1 0.8 1 0.8 0.7 0.7 0.7 0.9 0.7 ...
\$ AST	: int 22 27 27 20 19 24 17 16 24 22 ...
\$ ALT	: int 25 23 31 27 13 42 12 15 21 16 ...
\$ Gtp	: int 27 37 53 30 17 119 16 16 31 14 ...
\$ dental.caries	: int 0 1 0 1 0 1 0 0 0 0 ...
\$ smoking	: int 1 0 1 0 1 1 0 0 1 0 ...

değerinin 0 veya 1 olması gibi)

Rstudio uygulamanızın sağ tarafında yer alan kısımdan direkt olarak veri hakkında genel bilgi sahibi olabiliriz.

- Verimiz 24 değişkenden oluşmaktadır.
- Verimizde toplamda 159256 veri yer almaktadır.
- Verimizde yer alan değerler genel anlamda int ve num olarak yer almakta olup bu değerler sayısal ya da sınıflandırılmış verilerdir. (Örneğin smoking

3.0 Örneklem Oluşturma

3.1 Örneklem Oluşturma

```
4 orneklem <- sample(1:nrow(train),2000)
5 veri <- train[orneklem,]
6 write.csv(veri,"C:/Users/pc/Desktop/analiz ödev/veri.csv", row.names = FALSE)
7
```

Yukarı görüldüğü üzere “Sample” fonksiyonu aracılığı ile örneklem oluşturma işlemi gerçekleştirilmekte olup, genel kullanımı;

Sample (baş: bitiş (veri kümesi), adet sayısı)

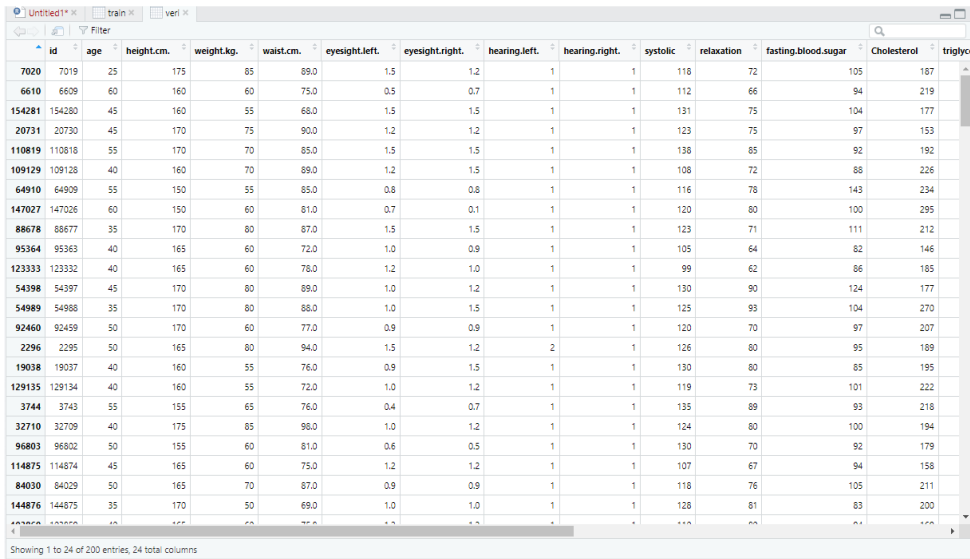
Şeklindeir.

2.adım olarak ise tanımlanmış olan örneklemin bir veri kümesi haline getirilmesi ve son olarak da kaybolmaması veya değişmemesi adına kaydedilmesi aşamaları yer alır.

Write.cvs (veri kümesi, “Bilgisayarda kaydedilmesini istediğin alan”)

Şeklindeir.

3.2 Örneklem Çıktısı



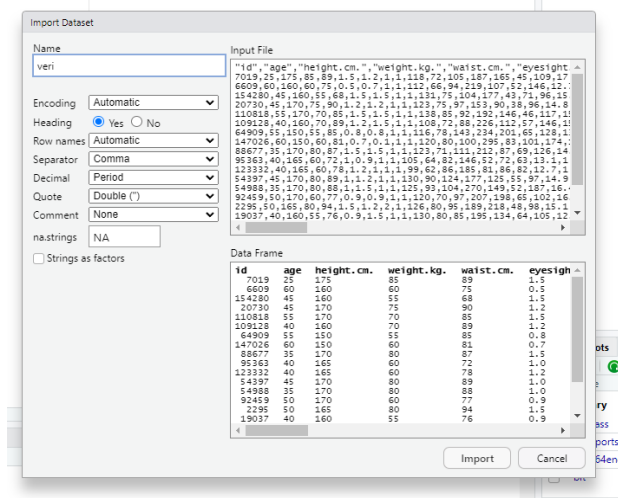
id	age	height.cm	weight.kg	waist.cm	eyesight.left	eyesight.right	hearing.left	hearing.right	systolic	relaxation	fasting.blood.sugar	Cholesterol	triglyce
7020	7019	25	175	85	89.0	1.5	1.2	1	118	72	105	187	
6610	6609	60	160	60	75.0	0.5	0.7	1	112	66	94	219	
154281	154280	45	160	55	68.0	1.5	1.5	1	131	75	104	177	
20731	20730	45	170	75	90.0	1.2	1.2	1	123	75	97	153	
110819	110818	55	170	70	85.0	1.5	1.5	1	138	85	92	192	
109129	109128	40	160	70	89.0	1.2	1.5	1	108	72	88	226	
64910	64909	55	150	55	85.0	0.8	0.8	1	116	78	143	234	
147027	147026	60	150	60	81.0	0.7	0.1	1	120	80	100	295	
88678	88677	35	170	80	87.0	1.5	1.5	1	123	71	111	212	
95364	95363	40	165	60	72.0	1.0	0.9	1	105	64	82	146	
123333	123332	40	165	60	78.0	1.2	1.0	1	99	62	86	185	
54398	54397	45	170	80	89.0	1.0	1.2	1	130	90	124	177	
54989	54988	35	170	80	88.0	1.0	1.5	1	125	93	104	270	
92460	92459	50	170	60	77.0	0.9	0.9	1	120	70	97	207	
2296	2295	50	165	80	94.0	1.5	1.2	2	126	80	95	189	
19038	19037	40	160	55	76.0	0.9	1.5	1	130	80	85	195	
129135	129134	40	160	55	72.0	1.0	1.2	1	119	73	101	222	
3744	3743	55	155	65	76.0	0.4	0.7	1	135	89	93	218	
32710	32709	40	175	85	98.0	1.0	1.2	1	124	80	100	194	
96803	96802	50	155	60	81.0	0.6	0.5	1	130	70	92	179	
114875	114874	45	165	60	75.0	1.2	1.2	1	107	67	94	158	
84030	84029	50	165	70	87.0	0.9	0.9	1	118	76	105	211	
144876	144875	35	170	50	69.0	1.0	1.0	1	128	81	83	200	

Yazılan kodu çalıştırdıktan sonra yan da yer alan şekilde 2000 adet gözlem taşıyan rastgele seçilmiş bir örneklem kümemiz oluşmuş hale gelir.

veri	2000 obs. of 24 variables									
\$ id	: int	101269	150813	111186	74026	3909	126822	51506	427...	
\$ age	: int	40	50	55	60	60	50	40	30	65 40 ...
\$ height.cm.	: int	175	170	170	150	165	165	185	170	165 155 ...
\$ weight.kg.	: int	60	60	85	40	60	65	90	85	55 70 ...
\$ waist.cm.	: num	72	71	103.4	73	80.2	...			
\$ eyesight.left.	: num	1.5	0.8	1	0.8	1	1	1.2	0.7	1.5 ...
\$ eyesight.right.	: num	0.9	1	1.2	1	0.5	0.8	1.2	1.2	0.9 1.5 ...
\$ hearing.left.	: int	1	1	1	1	1	1	1	1	1 ...
\$ hearing.right.	: int	1	1	1	1	1	1	1	1	1 ...
\$ systolic	: int	130	94	106	130	120	125	130	134	134 104 ...
\$ relaxation	: int	70	59	74	82	80	81	80	94	74 62 ...
\$ fasting.blood.sugar	: int	76	88	101	87	110	81	87	101	101 90 ...
\$ cholesterol	: int	149	175	149	238	188	197	273	210	171 186 ...
\$ triglyceride	: int	80	148	91	65	37	85	214	329	147 189 ...
\$ HDL	: int	79	64	50	56	62	72	58	44	63 45 ...
\$ LDL	: int	54	80	82	170	118	108	173	91	80 104 ...
\$ hemoglobin	: num	15.4	14.3	15.3	13.9	15.2	16.7	15.6	16.2	13.9 13...
\$ Urine.protein	: int	1	1	1	1	1	1	1	1	1 ...
\$ serum.creatinine	: num	1.2	0.9	0.9	0.7	0.9	1.1	0.9	1	0.9 0.8 ...
\$ AST	: int	19	28	22	21	27	32	21	32	23 24 ...
\$ ALT	: int	18	17	14	14	22	36	26	60	15 24 ...
\$ Gtp	: int	23	14	29	17	28	77	54	99	44 47 ...
\$ dental.caries	: int	0	0	0	0	0	0	0	1	0 ...
\$ smoking	: int	1	0	1	0	0	1	1	0	0 ...
values										
orneklem	: int [1:2000]	101270	150814	111187	74027	3910	126823	5150...		

Böylece analizin sağlanması istenilen veri kümemizin incelemesi sağlanılacak olan Örneklemi hazır hale getirmiş bulunmaktayız.

Not:



Raporlamanın başında da belirtildiği gibi tekrar vurgulamak gerekirse örneklem kümesi oluşturduktan sonra “csv” olarak kaydetmemizin temel sebebi projemiz anında bitmeyeceği için uygulama üzerinde kapama açma değişiklik gibi durumlarda sürekli yeni örneklem oluşturmadan mevcut yer alanla devam edebilmektir o yüzden yeniden giriş yapılıncaya erişim olmaması durumunda “import dataset” kısmında aynı örneklem yüklenerek devam edilir.

4.0 Verinin Analizi

4.1 Temel Analizler

```
11
12 grouped_cb_smoking <- group_by(veri, smoking)
13 grouped_cb_smoking
14
15 summarise(grouped_cb_smoking, mean(age))
16
```

Dbplyr kütüphanesinden faydalanarak verileri gruplayarak ortalama, toplam vb. birçok temel analizi gerçekleştirebilmekteyiz, örneğin yukarıda sigara içme durumuna göre sınıflandırmanın ardından sigara içen kişilerin ve içmeyen kişilerin yaş ortalamasını elde etmiş bulunmaktayız.

```
R 4.3.2 ~ /
> grouped_cb_smoking <- group_by(veri, smoking)
> grouped_cb_smoking
# A tibble: 200 x 24
# Groups:   smoking [2]
   id    age height.cm. weight.kg. waist.cm. eyesight.left. eyesight.right. hearing.left. hearing.right. systolic relaxation
   <int> <int>   <int>      <int>    <dbl>      <dbl>      <dbl>      <int>      <int>      <int>      <int>
1  7019    25     175       85      89        1.5        1.2        1         1        118       72
2  6609    60     160       60      75        0.5        0.7        1         1        112       66
3 154280   45     160       55      68        1.5        1.5        1         1        131       75
4  20730   45     170       75      90        1.2        1.2        1         1        123       75
5 110818   55     170       70      85        1.5        1.5        1         1        138       85
6 109128   40     160       70      89        1.2        1.5        1         1        108       72
7  64909   55     150       55      85        0.8        0.8        1         1        116       78
8 147026   60     150       60      81        0.7        0.1        1         1        120       80
9  88677   35     170       80      87        1.5        1.5        1         1        123       71
10 95363   40     165       60      72        1         0.9        1         1        105       64
# i 190 more rows
# i 13 more variables: fasting.blood.sugar <int>, Cholesterol <int>, triglyceride <int>, HDL <int>, LDL <int>, hemoglobin <dbl>,
#   urine.protein <int>, serum.creatinine <dbl>, AST <int>, ALT <int>, Gtp <int>, dental.carries <int>, smoking <int>
# i Use 'print(n = ...)' to see more rows
> summarise(grouped_cb_smoking, mean(age))
# A tibble: 2 x 2
  smoking `mean(age)`
   <int>      <dbl>
1     0      44.7
2     1      42.5
> |
```

```
17 summarise_at(vars(id:smoking), mean, na.rm = TRUE)
18
19 no_smoking <- filter(veri, smoking == 1)
20 no_smoking
21
22 library(shiny)
```

Yanda yer
aldığı gibi
filtreleme
yönetimi

aracılığı ile çeşitli verileri sıralayabilir ve kaydedebiliriz. Burada gördüğümüz örnekte “smoking” eşittir 1 olan veri kümesi çekilmiştir.

Bu gibi temel veri analizleri içeren Dbplyr kütüphanesi yardımı ile çoğaltabiliriz.

```
Console ~ /R/
> yeni_tweet_verisi2 <- cbs_tweets %>%
+   select(aday, tarih, cihaz, begenme, retweet, takipci_sayısı) %>%
+   mutate(etkilesim_degeri =(begenme+retweet/2)/takipci_sayısı) %>%
+   filter(aday %in% c("aksener", "demirtas", "erdogan", "ince")) %>%
+   arrange(aday)
> identical(yeni_cbs_tweets, yeni_tweet_verisi2)
[1] TRUE
```

Yanda yer aldığı gibi “select, mutate, filter, arrange” bunların yanı sıra summarise, mean, gibi birçok fonksiyon ile temel analiz yapılabilir.

4.2 Creating One Scatter Plot for Each Group

4.2.1 Kullanılan Paketler

```
27  
28 require(ggiraphExtra)  
29 require(ggplot2)  
30 require(ggiraph)  
31 require(plyr)  
32
```

grafikte yer alan kütüphanelerden faydalanacağız.

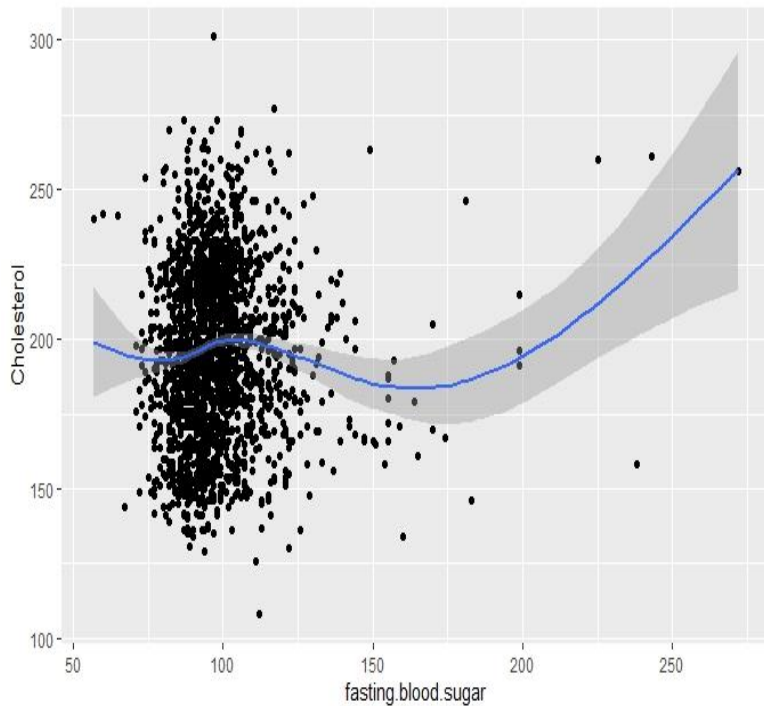
Yan taraftaki resimde yer alan Rstudio kütüphanelerinin temel kullanım amacı 2 ve 3 boyutlu grafiklerde destek ve olanak sağlamak olup oluşturacağımız çoğu

4.2.2 Grafiğin Oluşturulması

Rstudio uygulaması üzerinde one scatter grafiği oluşturma adına “ggplot” ya da “ggpoint” fonksiyonlarından faydalanmamız gerekecek.

```
32  
33 ggplot(veri,aes(fasting.blood.sugar,cholesterol)) + geom_point() + geom_smooth()  
34
```

En temel One Scatter Plot grafiğinden başladığımızda yukarıda yer alan ggplot ya da alternatif olarak ggPoint fonksiyonu kullanılarak yanda yer alan grafiğe ulaşım sağlanır.



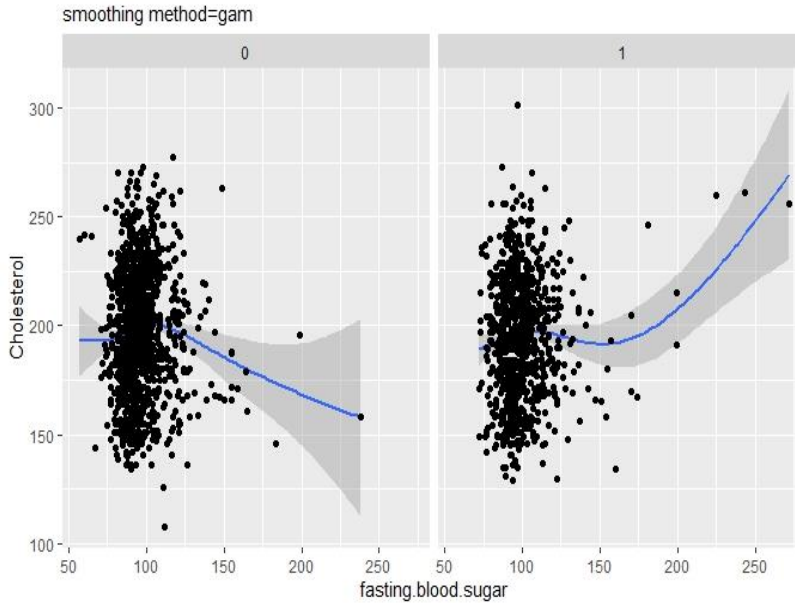
- Grafiği inceleme aşamasına geldiğimizde ise Kolesterol değeri ve açlık şekeri diye de tabir edebileceğimiz iki değeri incelemiş oluyoruz.
- İncele sonucunda görüyoruz ki diğer kriterler bir kenara bırakıldığında her iki değerinde de ekstrem bir değişim olmadığını ve birbirlerini de olumlu ya da olumsuz bir etkisi yer almamaktadır.

```

34
35 ggPoints(veri, aes(fasting.blood.sugar, cholesterol)) +
36   geom_point() +
37   facet_wrap(~ smoking)
38

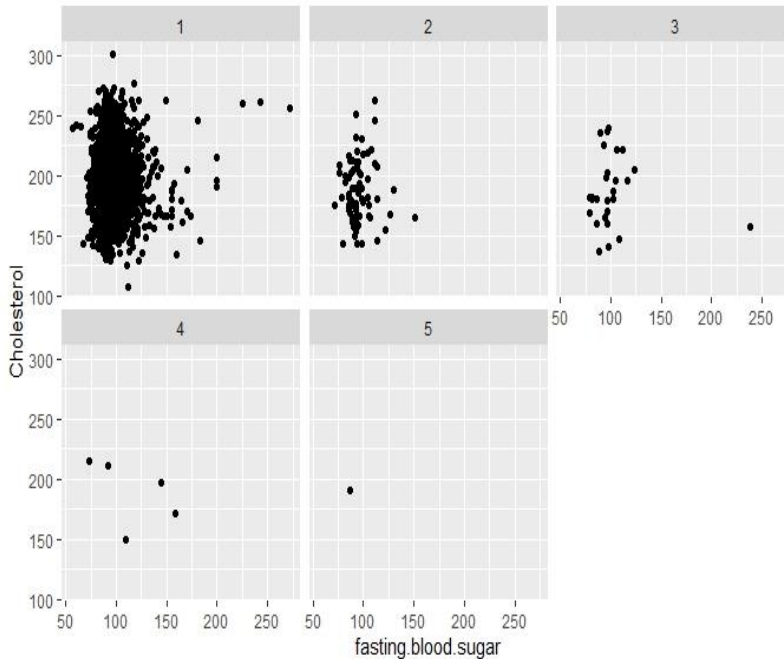
```

Burada yer alan kodlama aracılığı ile temel incelemesi sağlanacak olan numerik bir verinin çeşitli grafikleri üzerinden inceleme sağlanacak olup yine aynı şekilde başta belirtilen kütüphanelerden faydalanılmıştır.



- Öncelikli olarak grafiğimize göz geçirdiğimizde x ekseninde açlık şekeri de diyebileceğimiz verilerimiz, y ekseninde ise Kolesterol verilerimiz yer almakla birlikte buradaki temel inceleme koşulumuz kişinin sigara tüketip tüketmemesi ile ilgilidir.
- Sigara tüketen bireyleri yani “1” olan bireyleri incelediğimizde oluşan grafik daha çok kolesterolün

değişkenlik gösterdiğini belirtir. Sigara tüketmeyen bireylere baktığımızda ise çok fazla bir değişkenlik olmaksızın hem kolesterol değerinin hem de açlık şekerinin belli düzeylerde sabit kaldığını görmekteyiz. Buradan sonuç olarak sigaranın bireylere kolesterol ve açlık şekeri bakımından aynı etkiyi gösterdiğini belirtmek mümkündür.



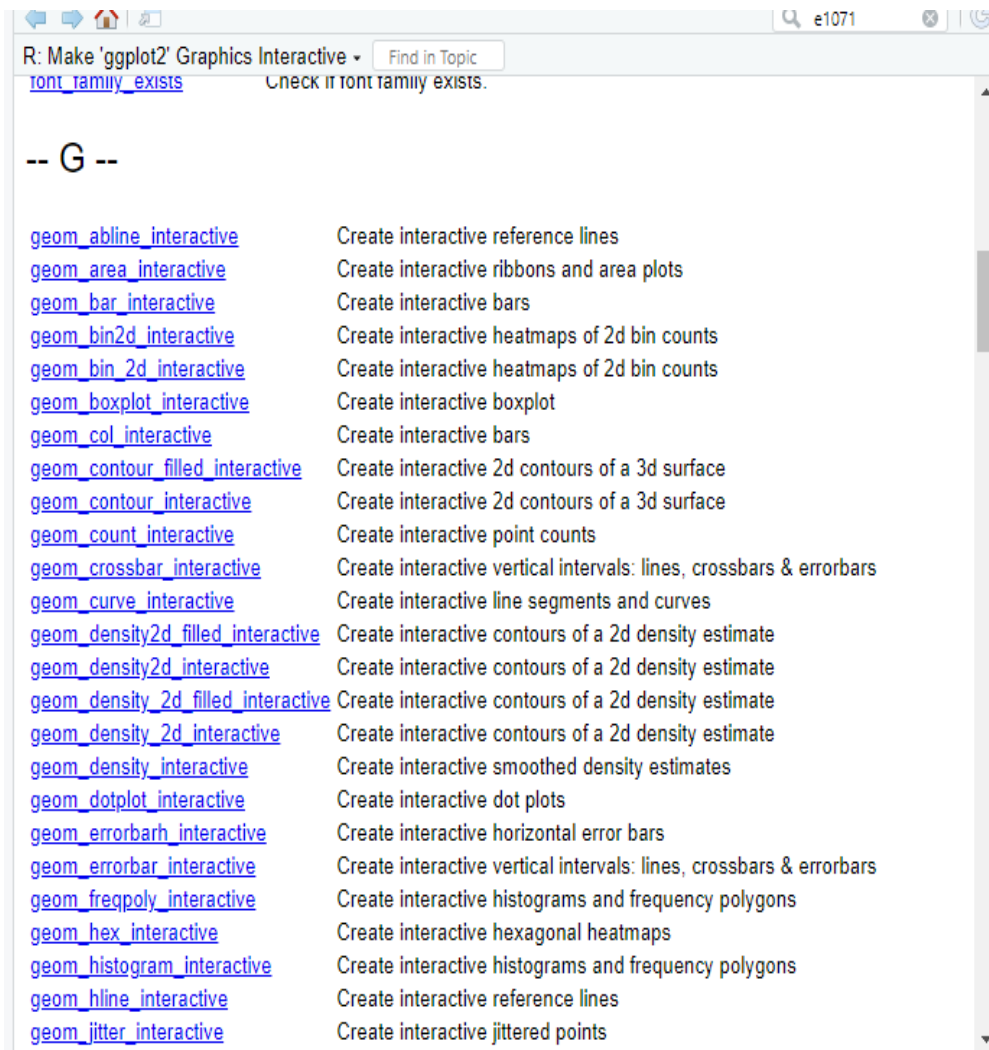
- İncelemesini sağlayacağımız bir diğer grafik ise yine aynı x ve y değerlerine istinaden insan vücudundaki bazı protein çeşitlerine etkisini inceleyeceğiz.
- İnceleme sonucunda direkt olarak ilgili protein çeşitlerinden genel olarak 1 numaralı proteinin insan vücudunda yer alması kaynaklı özellikle kolesterol bakımından farklı değerlere rahatlıkla ulaşabileceği kanısına ulaşırız

4.3 Creating One Boxplot for Each Factor Level

4.3.1 Kullanılan Paketler

```
43  
44 require(ggiraphExtra)  
45 require(ggplot2)  
46 require(ggiraph)  
47
```

- Yukarıdaki görselde de görüldüğü üzere kullanacağımız ve faydalanacağımız 3 adet kütüphane yer almakta olup aslında baktığımızda aynı kütüphanelerden bir önceki grafiğimizde de yararlandığımızı görmekteyiz.



Yukarıda görmüş olduğumuz ilgili ggiraph kütüphanesinin G harfi ile başlayan fonksiyonlarını göstermektedir. Burada bu kütüphanenin gösterilmesinin temel nedeni yapacak olduğumuz grafikte geom_ ile başlayan fonksiyondan faydalanacak olmamızın yanı sıra eğer bir kütüphane içerisinde fonksiyon bulmamız gerekiyor ise nasıl bulunacağı hakkında ufak bir gösterim olmasıdır.

4.3.2 Grafiğin Oluşturulması

```
47  
48  
49 ggplot(veri) +  
50   aes(x = as.factor(urine.protein), y = age) +  
51   geom_boxplot() |  
52
```

- Yukarıda yer alan “ggplot” fonksiyonu aracılığı ile grafiğe ait yer alan ve aktarım işlemlerini tamamlamış olup ardından bir önceki alanda da belirttiğimiz gibi;

Geom_boxplot ()

Fonksiyonu ile ggiraph kütüphanesinden faydalanıp grafiğimizi şekillendirmiş olduk.

```
52   labs(  
53     title = "Urine Protein yaş dağılım",  
54     x = "Urine Protein",  
55     y = "Yaş"  
56   )  
57
```

- Bunun yanı sıra ek olarak yukarıda belirtilen “labs” fonksiyonu aracılığı ile de grafiğimizin başlığı, x eksenin ismi ve y eksenin ismi gibi değişimler yapabilmekteyiz.

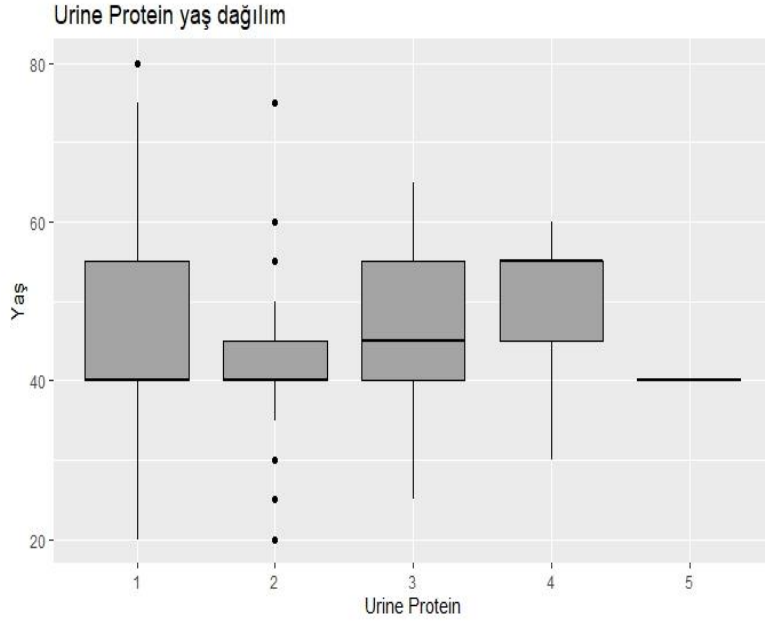
```
48  
49 ggplot(veri) +  
50   aes(x = as.factor(urine.protein), y = age) +  
51   geom_boxplot(fill="#A4A4A4", color="black") +  
52   labs(  
53     title = "Urine Protein yaş dağılım",  
54     x = "Urine Protein",  
55     y = "Yaş"  
56   )  
57
```

- Bir diğer özelliğimizde yine “geom_” fonksiyonu içerisinde yer alan “fill” ve “color” araçlarıdır. Renklendirme kısım ve aşamalarında çeşitli renk kodu ve isimleriyle görsel düzen sağlayabilmemize katkı sağlıyor.

- **Kaynakça**

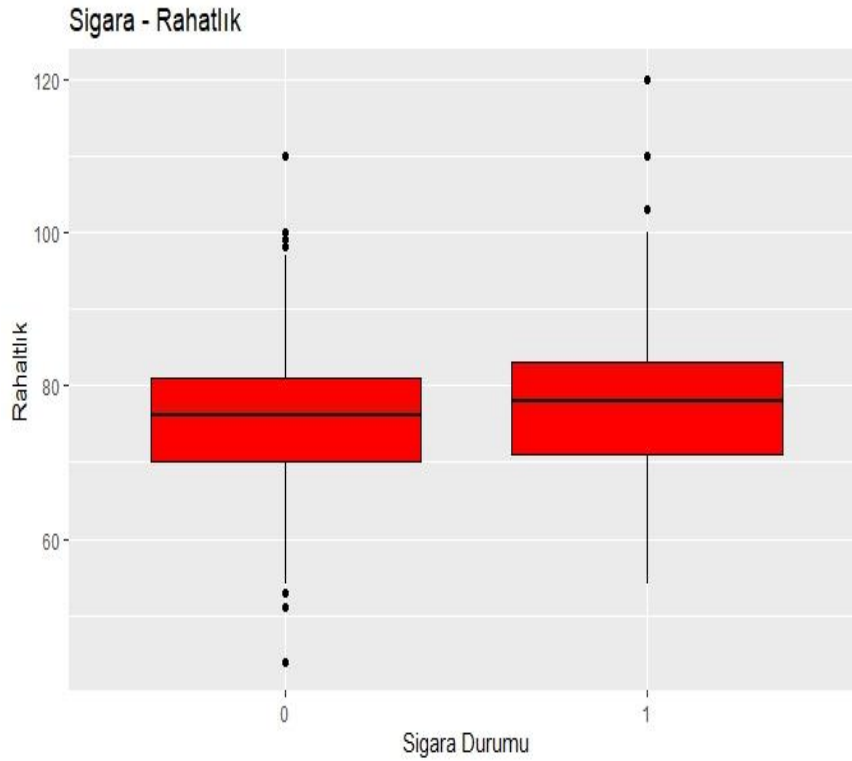
<http://www.sthda.com/english/wiki/ggplot2-box-plot-quick-start-guide-r-software-and-data-visualization>

4.3.3 Grafik Çıktıları ve Yorumlama



- Yukarıda kodlamalar aracılığı ile oluşturduğumuz grafik bu şekilde yer almaktadır.
- Grafiğimize genel anlamda baktığımızda Urine Protein miktarının yaş durumuna göre dağılımını baz almaktadır.
- Her beş protein türü içinde inceleme sağladığımızda geçtiğimiz grafiklerde de karşılaştığımız gibi 5 numaralı protein çeşidi az ve istisnai bir şekilde bulunuyor ve 40 yaş dolaylarında yer almaktadır.

- Diğer dört protein çeşidimizi incelediğimizde 1 ve 3 numaralı protein genelde 40-60 yaş aralığında görülmekte, 2 numaralı protein çeşidimiz ise daha çok 40 yaş arasındaki genç kesimde görülmektedir.



- İkinci bir örnek olarak sigara içme durumunun rahatlık seviyesini etkileme durumunu incelemek adına oluşturulmuştur.

- Grafiğimizi incelediğimizde sigara durumu olarak 0 = içmiyor, 1 = içiyor anlamında bir kullanım mevcuttur.

- Grafiğimize aynı zamanda bir bütün olarak da göz gezdirdiğimizde direkt olarak aşırı

derecede bir fark yaratacak durum göze çarpmaz. Daha detaylıca kontrol edildiğinde ise sigara içmeyenlerin dağılımında birkaç ufak aykırı değer görmekle beraber daha ortalamaya yakın ve düzgün dağılmış olduğunu görmekteyiz.

- Sonuç olarak inceleme doğrultusunda sigara içip içmemenin rahatlığa çok büyük bir etken yaratmayacağını açığa kavuşturabiliriz

4.4 Creating a Histogram

4.4.1 Kullanılan Paketler

Kullanılan paketlerde herhangi bir değişim yoktur bir önceki 4.3.1 de ki paketler tamamen aynıdır.

4.4.2 Grafiklerin Oluşturulması

```
67  
68  
69 ggplot(veri, aes(x = age)) +  
70   geom_histogram() |  
71  
72
```

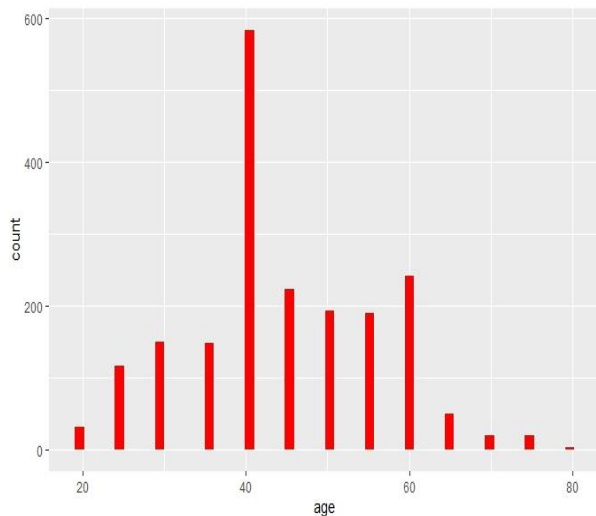
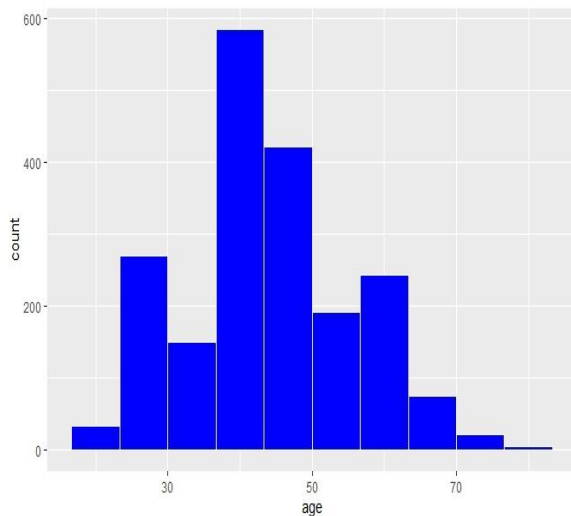
Diğer grafikler de olduğu gibi yine kullandığımız kütüphane kaynaklı “ggplot” ve “geom” fonksiyonlarını kullanmış bulunmaktayız.

```
67  
68  
69 ggplot(veri, aes(x = age)) +  
70   geom_histogram(bins = 20) |  
71  
72
```

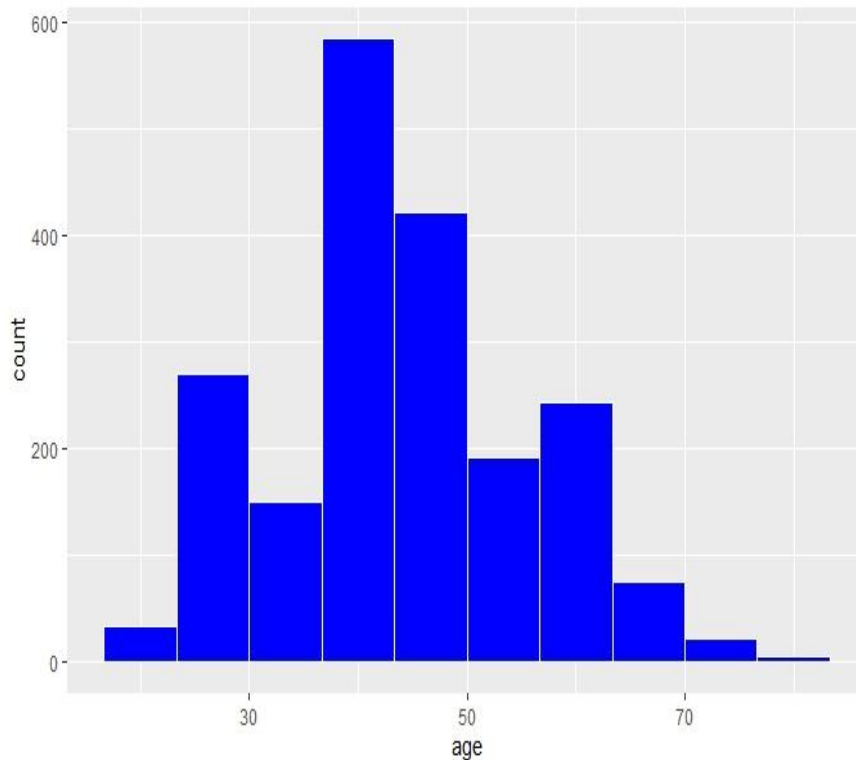
“Bins = “fonksiyonu aracılığı ile x ekseninin sayısal değerlerine düzenleme sağlanır.

```
72  
73 ggplot(data = veri, aes(x = age)) +  
74   geom_histogram(bins=10,color = "white", fill = "blue") |  
75  
76
```

“Geom” fonksiyonu içerisinde color, fill gibi yapılarla oluşturduğumuz grafiğin renklendirilebilmesine olanak sağlarız.



4.4.3 Grafik Çıktıları ve Yorumlama



- Yanda yer alan grafiğimizde histogram üzerinde yer alan örneklemimizdeki kişilerin yaşlarının dağılımı yer almaktadır.
- Grafiğimizin dağılımını kontrol ettiğimizde daha çok orta yaş ağırlıklı dağılım olduğunu görmekteyiz.

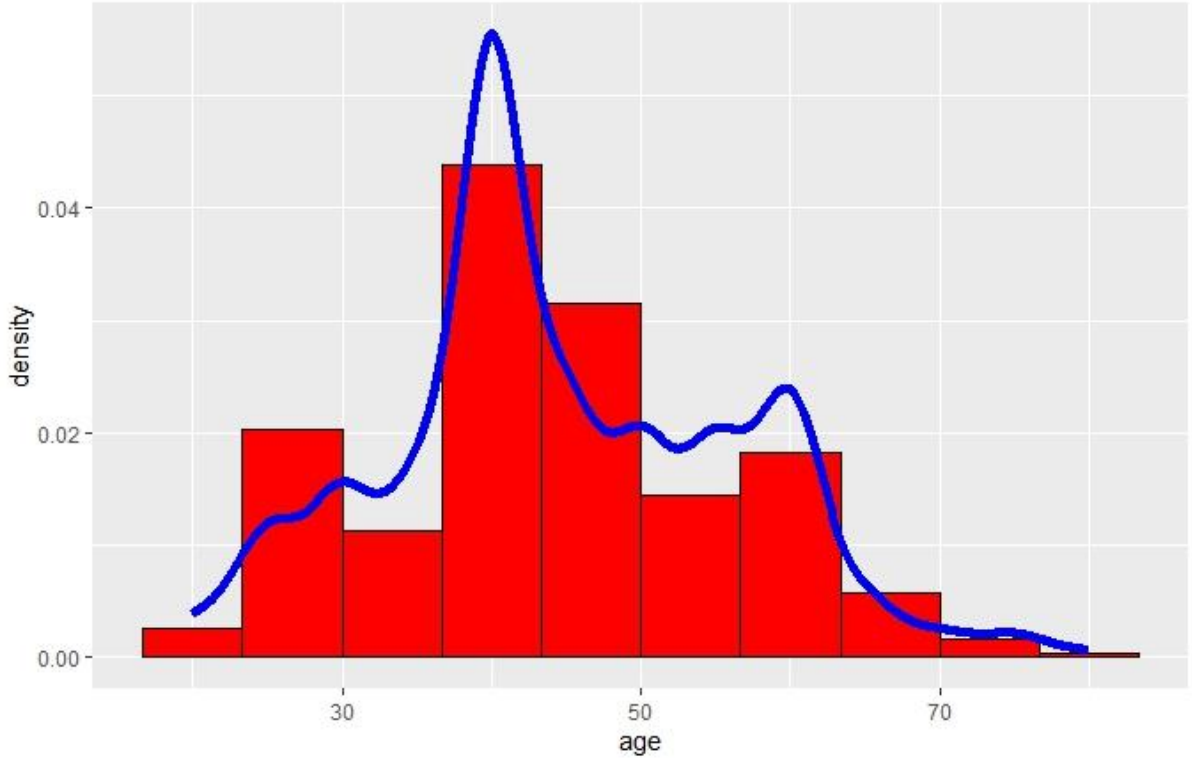
```
> mean(age)
[1] 43.625
> mode(age)
[1] "numeric"
> mod(age)
[1] 25 60 45 45 45 55 40 55 60 35 40 40 45 35 50 50 40 40 55 40 50 45 50 35 40 50 35 30 30 35 45 25 55 40 55 40 55 50 20 40 70 20 40 60 60 40 65
[47] 20 45 45 40 20 45 30 50 50 35 40 30 35 55 40 45 40 20 40 55 75 65 55 30 50 75 40 40 40 45 40 25 30 20 45 50 40 60 40 70 60 40 35 50 30 30
[93] 50 50 45 40 45 40 45 60 25 25 60 55 50 60 60 25 60 40 40 25 40 55 45 55 40 60 50 45 40 45 55 50 35 55 25 40 60 55 55 30 60 40 40 20 60 25
[139] 40 30 45 40 45 40 40 60 40 40 50 55 45 65 40 45 40 60 25 55 40 40 40 60 45 40 25 55 60 40 40 55 40 35 45 20 60 25 45 45 40 30 50 40 40 40
[185] 40 40 40 25 55 35 40 40 45 30 40 40 45 60 40 40
```

- Dağılımın çarpıklığını öğrenebilmek adına mod, medyan ve ortalama incelemesi yaparız
- Ortalama: 43.625
- Mod: 40
- Medyan: 40
- Bu sonuçlar doğrultusunda grafiğimizin sağa çarpık olduğu sonucuna ulaşabiliriz.

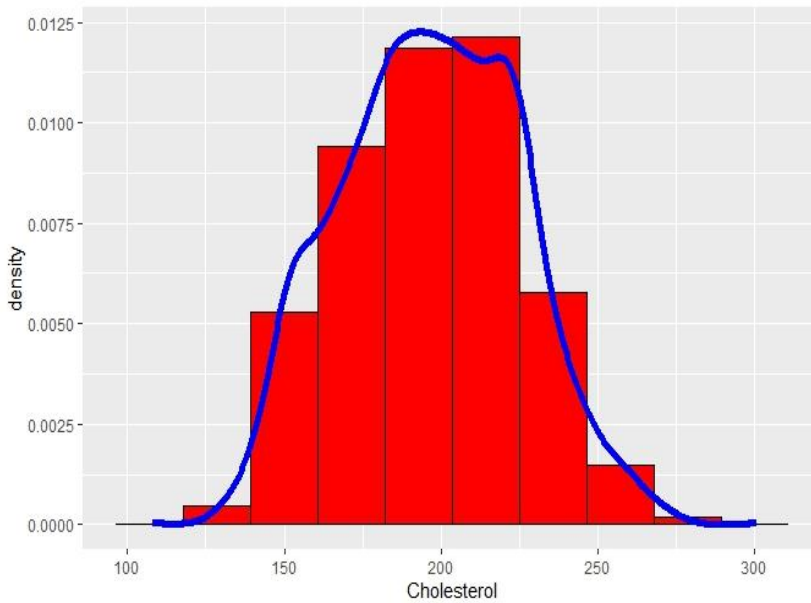
4.4.4 Adding a Density

```
82 ggplot(veri) +
83   aes(x = age) +
84   geom_histogram(aes(y = ..density..), bins = 10, color = "black", fill = "red") +
85   geom_density(color = "blue", linewidth = 2)
86 |
87
```

Yukarıda yer alan kodlara ekstra olarak `geom_density` fonksiyonu eklenerek gerekli eklemeler yapılmakla beraber bu fonksiyonun içerisinde çizgi rengi, çizgi kalınlığı gibi ekstra eklemeler de yapılabilir.



- Yukarıda genel hatları ile verilen histogramın üzerine çizgi grafiği eklenmiştir.
- Bu oluşturulmuş olan çizgi grafiği özellikle dağılım çarpıklığı inceleme açısından daha verimli ve kullanışlı olmakla beraber bir önceki grafiğimizde mod, medyan ve ortalama hesaplamaları ile dağılımın sağa çarpık olduğunu tespit etmiştik. Şimdi oluşturduğumuz çizgi grafiği ile de sağa çarpıklık durumunu netleştirmiş bulunmaktayız.



- Farklı bir örnek de görebilmek açısından kolesterol değerlerini histogram ve çizgi grafiği incelemesine tabi tuttuğumuzda oransal ve eşit bir dağılım göze çarpmaktadır.
- Dağılım değerleri incelendiğinden;
- Ortalama :197
- Medyan: 200
- Olduğundan çok ufak da olsa bir fark ile bu dağılım sağa çarpık diyebiliriz.

NOT

```
93  
94 install.packages("e1071")  
95 |
```

Burada yer alan paket çağırılarak yukarıda yer alan incelemeler için çarpıklık ve basıklık katsayısı

hesaplaması sağlayabiliriz.

```
98  
99 skewness(x, na.rm = FALSE, type = 3)  
100 kurtosis(x, na.rm = FALSE, type = 3)  
101  
102
```

```
# i Use `print(n = ...)` to see more rows  
> skewness(age, na.rm = FALSE, type = 3)  
[1] 0.2538902  
> kurtosis(age, na.rm = FALSE, type = 3)  
[1] -0.2928962  
> |
```

- Yukarıda görmüş olduğumuz hesaplama “age” yani yaş grafiğimize ait olup hesaplamalar doğrultusunda `skewness > 0` olduğundan yukarıda yapmış olduğumuz sağa çarpık sonucumuzu kanıtlamış oluruz.

```
> skewness(Cholesterol, na.rm = FALSE, type = 3)  
[1] 0.04549845  
> kurtosis(Cholesterol, na.rm = FALSE, type = 3)  
[1] -0.4868795  
> |
```

- Aynı işlemi kolesterol grafiğimizi de kanıtlamak adına uygulayabilmekteyiz.
- Yukarıda görüldüğü üzere hesaplamalarımız doğrultusunda kolesterol grafiğimizinde yaş grafiğinde olduğu gibi sağa çarpık olduğunu kanıtlamış ve göstermiş oluruz.
- Sonuç olarak gördüğümüz gibi grafik üzerinden hesaplarken basit yollardan mod ,medyan ve ortalama gibi alanlar değerler kullanırken, aynı zamanda farklı kütüphanelerden daha kısa yollarla hesaplamalarımızda gerçekleştirebilmekteyiz.

5.0 Changing the Type, Width, or Color of a Line

5.1 Line Type

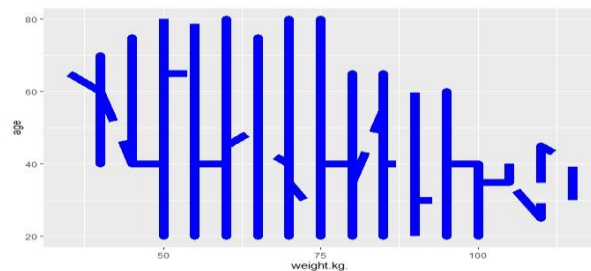
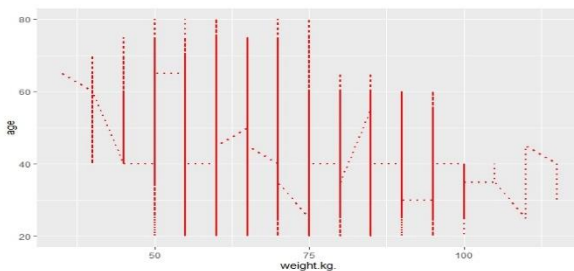
- Kullandığımız veri veya oluşturduğumuz grafikler üzerinde değiştirmek isteyeceğimiz birçok şey bulunabilir. Bunlardan birisi de çizgi şekli veya stili adını verdiğimiz kısımdır.
- Öncelikle olarak direkt uygulama kütüphanesi kaynaklı kullanabileceğimiz örnek stillere yer vereceğiz.

`ggplot` uses the `linetype` parameter for controlling the appearance of lines:

- `linetype="solid"` OR `linetype=1` (default)
- `linetype="dashed"` OR `linetype=2`
- `linetype="dotted"` OR `linetype=3`
- `linetype="dotdash"` OR `linetype=4`
- `linetype="longdash"` OR `linetype=5`
- `linetype="twodash"` OR `linetype=6`
- `linetype="blank"` OR `linetype=0` (inhibits drawing)

```
107  
108 ggplot(veri, aes(weight.kg., age)) +  
109   geom_line(linetype = 5,  
110             size = 4,  
111             col = "blue")  
112
```

- “`geom_line`” fonksiyonu içerisinde yukarıda ki resimde yer alan seçenekler üzerinden seçim yapıp aynı zamanda `size`, `col` gibi fonksiyonlarla da çeşitli boyut ve renklendirme işlemleri sağlayabiliriz.

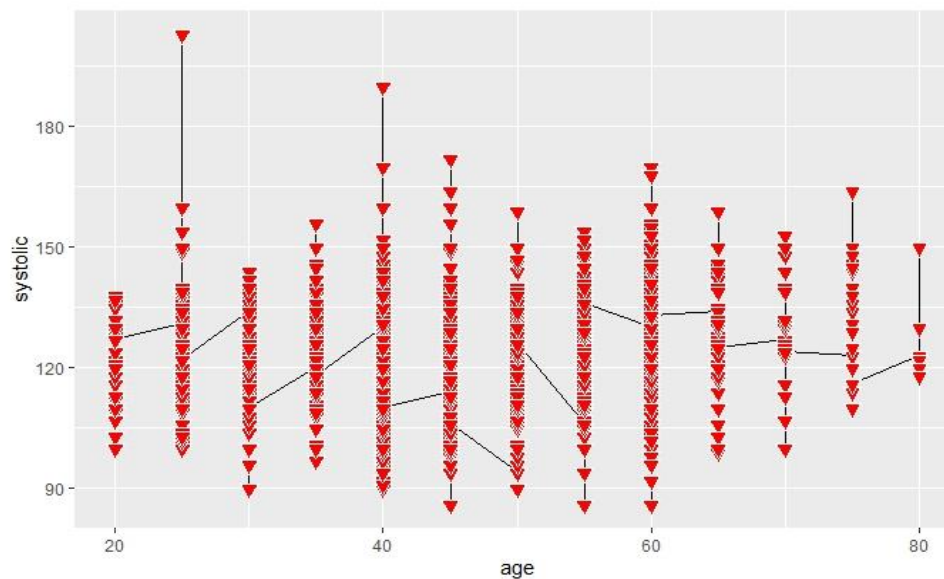
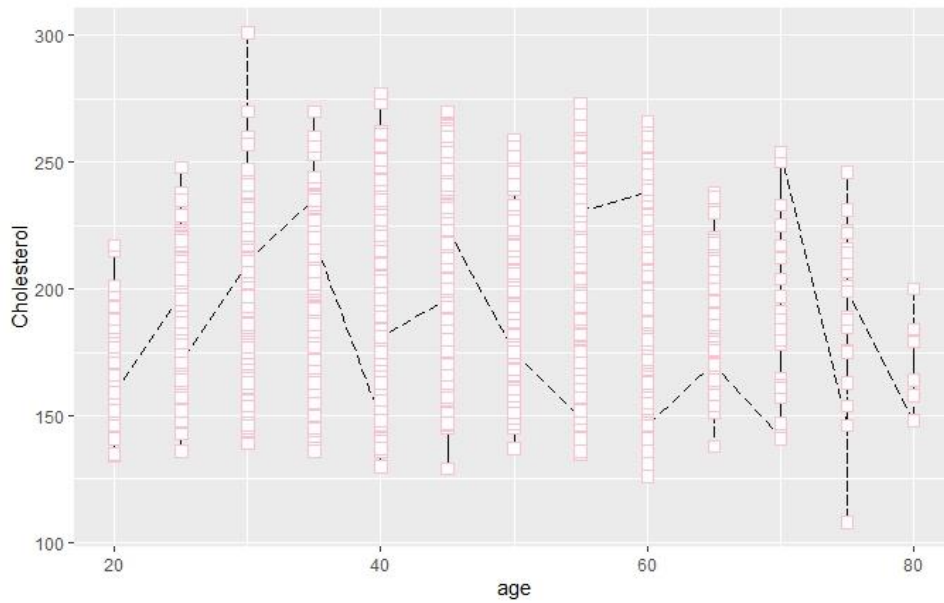


5.2 Change width and color

- Bir önceki 5.1 de gördüğümüz kısımda aynı zamanda çizgimizin boyut ve rengini de değiştirebildiğimizi görmüş olmakla beraber bu madde içerisinde ekstrasdan farklı yollarda boyut ve renk değiştirme işlemleri yapılacaktır.

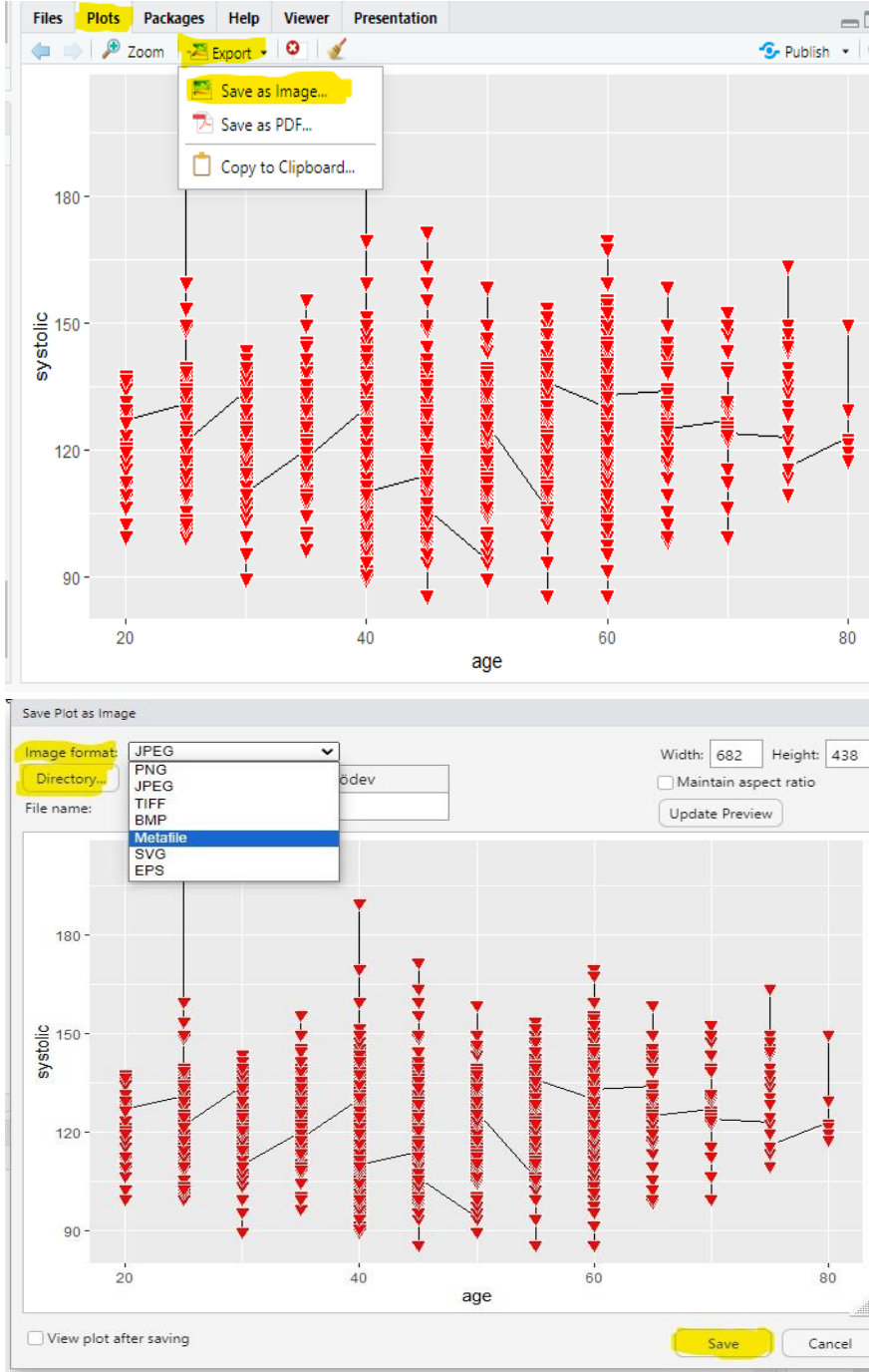
```
113  
114 ggplot(veri,aes(x=age, y = Cholesterol)) +  
115   geom_line(linetype = 5) +  
116   geom_point(shape = 22, size = 3, fill = "white",color = "pink")  
117
```

- Yer alan kodlamada da görüleceği üzeri Dbplyt kütüphanesi ve ggiraph kütüphanesi aracılığı ile geom_line, geom_point fonksiyonları aracılığı ile kolay bir şekilde oluşturulan grafiğimizin şekil, boyut ve renk değişimini gerçekleştirebilmekteyiz.



6.0 Writing Your Plot to a File

6.1 Uygulamadan Yazdırma



- İlk seçeneğimiz direkt olarak Rstudio uygulaması içerisindeki plots konsolu üzerinden ilerlemektir.
- Sırasıyla resimde de belirtildiği şekilde Export ve Save as Image seçenekleri takip etmemiz gerekmektedir.
- Ardından ise gelen ekran üzerinde önemli olan kısımlar özellikle sarı renk içerisine alınmış olup resmin formatı, kaydetmek istediğiniz dosya gibi alanlar seçerek ilerleyip sağ altta yer alan save seçeneği ile kaydetme işlemi başarılı bir şekilde gerçekleştirilir.

6.2 Kod ile Yazdırma

- Kod aracılığı ile yazdırma işleminde birçok şekil ve birçok kütüphane yer alabilir. Bunlardan birkaç örnek gösterimi sağlanılacak olup şu an gösterimi sağlanacak olan en temel kütüphane grafikte de faydalanmış olduğumuz ggiraph ve ggplot, ggplot2 kütüphanesinde yer alan “gg” fonksiyonu olacaktır.

```
119  
120 ggsave("gg.jpg", plot = gg, units = "in", width = 5, height = 4)  
121  
122
```

- Ekran görüntüsünde de gördüğümüz üzere “ggsave” fonksiyonu aracılığı ile boyutu, kayıt türünü vb. bilgiler aracılığı ile grafiğimizi bilgisayarımıza kaydedebilmekteyiz.

Kaynakça

- <https://rc2e.com/> (R Cookbook, 2nd Edition)
- <http://www.sthda.com/english/wiki/ggplot2-box-plot-quick-start-guide-r-software-and-data-visualization>
- <https://cran.r-project.org/web/packages/dplyr/vignettes/dplyr.html>
- <https://medium.com/@darnelbolaos/interactive-line-plot-with-ggplot-and-ggiraph-in-r-606d655a1547>
-