Veri Analizi Ödev

Barışcan Bozkurt - 20181101031

01 05 2021

Veri setinin ilk 6 gözlemi listelenmiştir.

head(blockbusters)

## # A tibble: 6 x 13  
## release\_year rank\_in\_year imdb\_rating mpaa\_rating film\_title film\_budget  
## <dbl> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 2019 1 8.5 PG-13 Avengers:~ 356000000  
## 2 2019 2 7 PG The Lion ~ 260000000  
## 3 2019 3 7.2 PG Frozen II 150000000  
## 4 2019 4 7.6 PG-13 Spider-Ma~ 160000000  
## 5 2019 5 6.9 PG-13 Captain M~ 175000000  
## 6 2019 6 7.9 G Toy Story~ 200000000  
## # ... with 7 more variables: length\_in\_min <dbl>, domestic\_distributor <chr>,  
## # worldwide\_gross <dbl>, domestic\_gross <dbl>, genre\_1 <chr>, genre\_2 <chr>,  
## # genre\_3 <chr>

Verimizin özet bilgisine bakalım.

summary(blockbusters)

## release\_year rank\_in\_year imdb\_rating mpaa\_rating   
## Min. :1977 Min. : 1.0 Min. :3.700 Length:430   
## 1st Qu.:1987 1st Qu.: 3.0 1st Qu.:6.600 Class :character   
## Median :1998 Median : 5.5 Median :7.000 Mode :character   
## Mean :1998 Mean : 5.5 Mean :7.077   
## 3rd Qu.:2009 3rd Qu.: 8.0 3rd Qu.:7.675   
## Max. :2019 Max. :10.0 Max. :9.000   
## film\_title film\_budget length\_in\_min domestic\_distributor  
## Length:430 Min. : 325000 Min. : 60.0 Length:430   
## Class :character 1st Qu.: 22000000 1st Qu.:104.0 Class :character   
## Mode :character Median : 70750000 Median :119.0 Mode :character   
## Mean : 88928663 Mean :120.6   
## 3rd Qu.:150000000 3rd Qu.:134.0   
## Max. :356000000 Max. :201.0   
## worldwide\_gross domestic\_gross genre\_1 genre\_2   
## Min. :3.719e+07 Min. : 2721100 Length:430 Length:430   
## 1st Qu.:2.155e+08 1st Qu.:105684362 Class :character Class :character   
## Median :3.801e+08 Median :173697808 Mode :character Mode :character   
## Mean :4.994e+08 Mean :202099606   
## 3rd Qu.:7.317e+08 3rd Qu.:261092025   
## Max. :2.798e+09 Max. :936662225   
## genre\_3   
## Length:430   
## Class :character   
## Mode :character   
##   
##   
##

Satırda kayıp gözlemler:

rowSums(is.na(blockbusters))

## [1] 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0  
## [75] 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0  
## [112] 1 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0  
## [149] 0 1 0 0 0 0 1 0 1 1 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1  
## [186] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 1 0  
## [223] 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1  
## [260] 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 1 1 1 0 0 0 0 0 1 1 1 1 0 1 1 0 0 0  
## [297] 1 1 0 1 1 1 0 1 0 0 0 0 0 1 0 0 1 0 1 1 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 0 1  
## [334] 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 1 0 1 0 0 1 0  
## [371] 1 0 1 1 1 1 1 0 0 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1 1 1 1 0 1 0 0 1 1 0 1 1 0  
## [408] 0 1 1 1 0 0 1 1 0 1 0 1 1 0 1 1 1 0 0 0 1 1 0

Sütunda kayıp gözlemler:

colSums(is.na(blockbusters))

## release\_year rank\_in\_year imdb\_rating   
## 0 0 0   
## mpaa\_rating film\_title film\_budget   
## 0 0 0   
## length\_in\_min domestic\_distributor worldwide\_gross   
## 0 0 0   
## domestic\_gross genre\_1 genre\_2   
## 0 0 0   
## genre\_3   
## 105

Genre 3’de 100’den fazla NA var, bu yüzden datamızdan çıkartacağım.

blockbusters\_yeni<-blockbusters[,-13]

Elimizdeki veriden NA oluşturmak için önce yedekledim.

blockbusters\_miss<-blockbusters\_yeni

Length\_in\_min yani film uzunluklarından rastgele NA veri oluşturmak için:

aa<-sample(1:nrow(blockbusters\_miss),floor(nrow(blockbusters\_miss)\*0.03))  
blockbusters\_miss$length\_in\_min[aa]<-NA

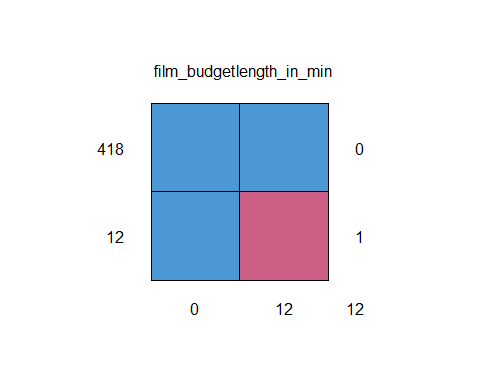
Kaç tane kayıp verimiz oluştuğuna bakmak için:

colSums(is.na(blockbusters\_miss))

## release\_year rank\_in\_year imdb\_rating   
## 0 0 0   
## mpaa\_rating film\_title film\_budget   
## 0 0 0   
## length\_in\_min domestic\_distributor worldwide\_gross   
## 12 0 0   
## domestic\_gross genre\_1 genre\_2   
## 0 0 0

Analizde fazla değişken olduğu için sadece NA olan sütun ile olmayan bir sütunu görüntüledim:

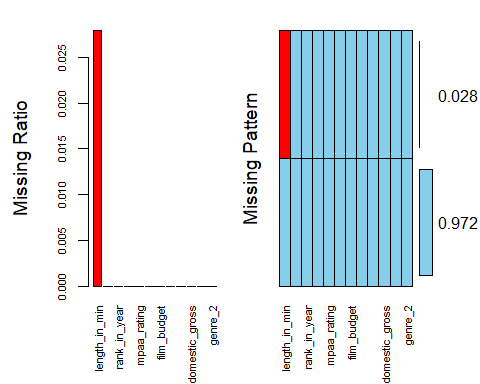
md.pattern(blockbusters\_miss[,c("film\_budget","length\_in\_min")])



## film\_budget length\_in\_min   
## 418 1 1 0  
## 12 1 0 1  
## 0 12 12

NA verilere bir başka paket ile bakalım:

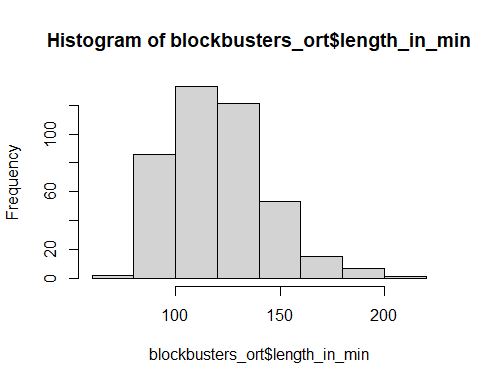
aggr(blockbusters\_miss,numbers=TRUE, sortVars=TRUE, labels=names(blockbusters\_yeni),cex.axis=.7,gap=3,ylab=c("Missing Ratio","Missing Pattern"))



##   
## Variables sorted by number of missings:   
## Variable Count  
## length\_in\_min 0.02790698  
## release\_year 0.00000000  
## rank\_in\_year 0.00000000  
## imdb\_rating 0.00000000  
## mpaa\_rating 0.00000000  
## film\_title 0.00000000  
## film\_budget 0.00000000  
## domestic\_distributor 0.00000000  
## worldwide\_gross 0.00000000  
## domestic\_gross 0.00000000  
## genre\_1 0.00000000  
## genre\_2 0.00000000

Basit doldurma yöntemi olan ortalama ile doldurma yapılabilir. Önce histogram grafiğine bakalım:

blockbusters\_ort<-blockbusters\_miss  
hist(blockbusters\_ort$length\_in\_min)



Bu kod ile de verimizi ortalaması olan 120 ile doldurduğunu göreceğiz:

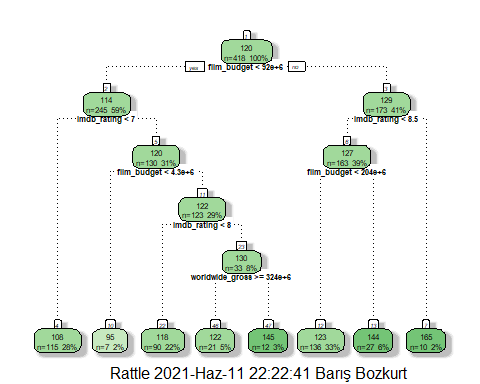
blockbusters\_ort$length\_in\_min[is.na(blockbusters\_ort$length\_in\_min)]<-mean(blockbusters\_ort$length\_in\_min,na.rm=TRUE)  
blockbusters\_ort$length\_in\_min<-as.integer(blockbusters\_ort$length\_in\_min)

Bir başka doldurma yöntemi olan karar ağacı dolduralım:

library(rpart)

## Warning: package 'rpart' was built under R version 4.0.5

data\_dt<-blockbusters\_miss  
rtree <- rpart(length\_in\_min ~ imdb\_rating + film\_budget+ worldwide\_gross, data\_dt, method="anova")  
library(rattle)  
fancyRpartPlot(rtree,cex=0.5)

 Boş yerleri elde ettiğimiz verilerle doldurmak için:

data\_dt$length\_in\_min <- ifelse(is.na(data\_dt$length\_in\_min), predict(rtree,data\_dt,type="vector"),data\_dt$length\_in\_min)  
data\_dt$length\_in\_min<-as.integer(data\_dt$length\_in\_min)

Hem sayıları NA verilere atamış oldum hem de integer haline çevirdim. Ancak bu NA verileri rastgele kendim oluşturmuştum. Bu yüzden ödevin devamında verinin orjinaline kullanmaya devam edeceğim.

Verimi %80 eğitim, %20’si test olacak şekilde böldüm.

set.seed(1234567)  
trainIndex <- sample(1:nrow(blockbusters\_yeni), size = round(0.8\*nrow(blockbusters\_yeni)), replace=FALSE)  
train<- blockbusters\_yeni[trainIndex ,]  
test <- blockbusters\_yeni[-trainIndex ,]

library("openxlsx")

## Warning: package 'openxlsx' was built under R version 4.0.5

write.xlsx(train, 'train.xlsx')  
write.xlsx(test, 'test.xlsx')

Yıl sayısı çok fazla olduğu için yılları kategorik hale getirmeye karar verdim.

train$yil\_kategorik[train$release\_year >= 1900 & train$release\_year <= 1989] <- "1\_1990\_oncesi"

## Warning: Unknown or uninitialised column: `yil\_kategorik`.

train$yil\_kategorik[train$release\_year >= 1990 & train$release\_year <= 2004] <- "2\_1990\_2005\_arasi"  
train$yil\_kategorik[train$release\_year >= 2005] <- "3\_2005\_sonrasi"

Kategorik veriyi oluşturduk fakat R’da henüz factor olarak algılanmıyor. Kategorik verilerimizin factor olarak algılanması için:

train<-as.data.frame(train)  
train$mpaa\_rating<-as.factor(train$mpaa\_rating)  
train$yil\_kategorik<-as.factor(train$yil\_kategorik)  
summary(train)

## release\_year rank\_in\_year imdb\_rating mpaa\_rating film\_title   
## Min. :1977 Min. : 1.000 Min. :3.700 G : 19 Length:344   
## 1st Qu.:1987 1st Qu.: 3.000 1st Qu.:6.600 PG :112 Class :character   
## Median :1997 Median : 6.000 Median :7.000 PG-13:138 Mode :character   
## Mean :1997 Mean : 5.642 Mean :7.082 R : 75   
## 3rd Qu.:2008 3rd Qu.: 8.000 3rd Qu.:7.700   
## Max. :2019 Max. :10.000 Max. :9.000   
## film\_budget length\_in\_min domestic\_distributor worldwide\_gross   
## Min. : 325000 Min. : 60.0 Length:344 Min. :3.719e+07   
## 1st Qu.: 20000000 1st Qu.:104.0 Class :character 1st Qu.:2.003e+08   
## Median : 70000000 Median :118.0 Mode :character Median :3.729e+08   
## Mean : 86628561 Mean :120.3 Mean :4.845e+08   
## 3rd Qu.:145000000 3rd Qu.:134.0 3rd Qu.:7.260e+08   
## Max. :356000000 Max. :201.0 Max. :2.798e+09   
## domestic\_gross genre\_1 genre\_2   
## Min. : 31755742 Length:344 Length:344   
## 1st Qu.:100438485 Class :character Class :character   
## Median :167613460 Mode :character Mode :character   
## Mean :196022744   
## 3rd Qu.:261577940   
## Max. :936662225   
## yil\_kategorik  
## 1\_1990\_oncesi :109   
## 2\_1990\_2005\_arasi:120   
## 3\_2005\_sonrasi :115   
##   
##   
##

Göründüğü üzere hem MPAA rating (Amerikan Sinema Filmleri Derneği Yaş Derecelendirmesi) hem de Yıl değişkenimiz artık kategorik olarak var.

library(funModeling)

## Warning: package 'funModeling' was built under R version 4.0.5

## Loading required package: Hmisc

## Warning: package 'Hmisc' was built under R version 4.0.5

## Loading required package: lattice

## Warning: package 'lattice' was built under R version 4.0.5

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##   
## Attaching package: 'Hmisc'

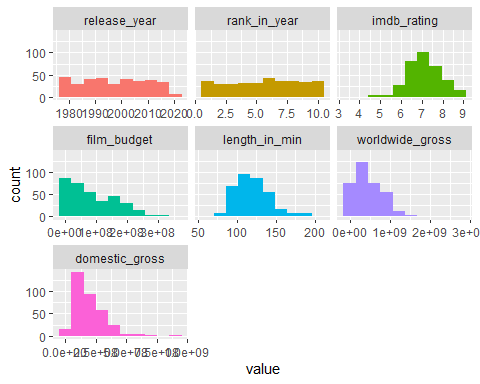
## The following objects are masked from 'package:base':  
##   
## format.pval, units

## funModeling v.1.9.4 :)  
## Examples and tutorials at livebook.datascienceheroes.com  
## / Now in Spanish: librovivodecienciadedatos.ai

profiling\_num(train)

## variable mean std\_dev variation\_coef p\_01  
## 1 release\_year 1.997480e+03 1.252634e+01 0.006271073 1977.00  
## 2 rank\_in\_year 5.642442e+00 2.866831e+00 0.508083338 1.00  
## 3 imdb\_rating 7.081977e+00 8.272478e-01 0.116810297 4.90  
## 4 film\_budget 8.662856e+07 7.514807e+07 0.867474559 3000000.00  
## 5 length\_in\_min 1.202820e+02 2.298483e+01 0.191091181 82.86  
## 6 worldwide\_gross 4.845283e+08 3.853553e+08 0.795320654 44289010.56  
## 7 domestic\_gross 1.960227e+08 1.293257e+08 0.659748475 39332809.86  
## p\_05 p\_25 p\_50 p\_75 p\_95 p\_99  
## 1 1978.000 1987.0 1997 2008.25 2017.0 2.019000e+03  
## 2 1.000 3.0 6 8.00 10.0 1.000000e+01  
## 3 5.815 6.6 7 7.70 8.5 8.800000e+00  
## 4 5745000.000 20000000.0 70000000 145000000.00 225000000.0 2.828000e+08  
## 5 89.150 104.0 118 134.00 162.7 1.881400e+02  
## 6 67251772.900 200285280.0 372852959 726031181.50 1121509119.1 1.767098e+09  
## 7 52853048.850 100438485.2 167613460 261577939.50 414638659.2 6.219920e+08  
## skewness kurtosis iqr range\_98  
## 1 0.03756584 1.789241 21.25 [1977, 2019]  
## 2 -0.09211809 1.823948 5.00 [1, 10]  
## 3 -0.24262638 3.519903 1.10 [4.9, 8.8]  
## 4 0.85770456 2.973908 125000000.00 [3e+06, 282800000]  
## 5 0.74110990 3.717121 30.00 [82.86, 188.14]  
## 6 1.54190166 7.461962 525745901.50 [44289010.56, 1767098023.52]  
## 7 1.72724325 8.092830 161139454.25 [39332809.86, 621992002.96]  
## range\_80  
## 1 [1980, 2015]  
## 2 [1.3, 10]  
## 3 [6.1, 8.1]  
## 4 [9300000, 2e+08]  
## 5 [93, 147.7]  
## 6 [79642748, 969501383.5]  
## 7 [69745350.3, 354608614.1]

plot\_num(train)



library(psych)

## Warning: package 'psych' was built under R version 4.0.5

##   
## Attaching package: 'psych'

## The following object is masked from 'package:Hmisc':  
##   
## describe

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:Hmisc':  
##   
## src, summarize

## The following objects are masked from 'package:stats':  
##   
## filter, lag

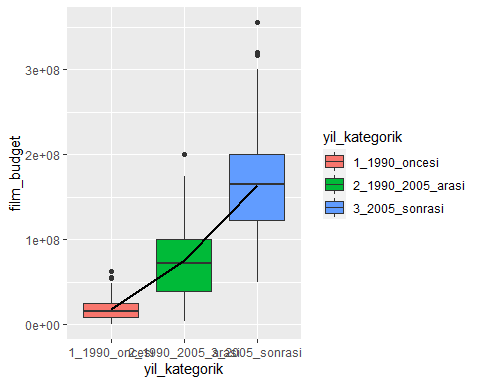
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

df<-select(train,film\_budget,yil\_kategorik)  
describeBy(df, df$yil\_kategorik)

##   
## Descriptive statistics by group   
## group: 1\_1990\_oncesi  
## vars n mean sd median trimmed mad min  
## film\_budget 1 109 17347018 12509234 1.5e+07 15782022 10378200 325000  
## yil\_kategorik\* 2 109 1 0 1.0e+00 1 0 1  
## max range skew kurtosis se  
## film\_budget 6.3e+07 62675000 1.27 1.45 1198167  
## yil\_kategorik\* 1.0e+00 0 NaN NaN 0  
## ------------------------------------------------------------   
## group: 2\_1990\_2005\_arasi  
## vars n mean sd median trimmed mad min  
## film\_budget 1 120 74970000 44077496 71750000 72510417 47072550 3500000  
## yil\_kategorik\* 2 120 2 0 2 2 0 2  
## max range skew kurtosis se  
## film\_budget 2e+08 196500000 0.5 -0.37 4023706  
## yil\_kategorik\* 2e+00 0 NaN NaN 0  
## ------------------------------------------------------------   
## group: 3\_2005\_sonrasi  
## vars n mean sd median trimmed mad min  
## film\_budget 1 115 164460870 63019530 1.65e+08 162935484 51891000 5e+07  
## yil\_kategorik\* 2 115 3 0 3.00e+00 3 0 3e+00  
## max range skew kurtosis se  
## film\_budget 3.56e+08 3.06e+08 0.26 -0.03 5876601  
## yil\_kategorik\* 3.00e+00 0.00e+00 NaN NaN 0

Aslında burada bariz bir şekilde beklediğimiz sonucu alıyoruz. Sonuçlara baktığımızda 1990 öncesindeki film bütçelerinin ortalama 17 milyon olduğu, 1990 ile 2005 arasındaki film bütçelerinin ortalama 74 milyon olduğu, 2005’den sonra da bu ortalama bütçenin 164 milyon gibi bir rakama uçtuğunu görüyoruz. Bunun en büyük sebebi sinemaya olan ilginin artmasıyla daha yüksek risklere girilmesi olarak açıklanabilir.

library(ggplot2)  
ggplot(train, aes(x=yil\_kategorik,y=film\_budget, fill=yil\_kategorik))+  
 geom\_boxplot()+  
 stat\_summary(fun = mean, geom="line", group= 1, color= "black", size = 1)



train$begeni<-ifelse(train$imdb\_rating>7,"Begenildi","Begenilmedi")  
dt<-table(train$begeni,train$yil\_kategorik)  
prop.table(dt,2)

##   
## 1\_1990\_oncesi 2\_1990\_2005\_arasi 3\_2005\_sonrasi  
## Begenildi 0.4403670 0.4833333 0.5478261  
## Begenilmedi 0.5596330 0.5166667 0.4521739

Burada beğeni olarak kabul edilen IMDB puanı 7 olarak aldım. 7’nin üstündeki filmler beğenildi, altındakiker ise beğenilmedi olarak sınıflandı. Genelde de 7 puan izleyici açısından kritik bir değerdir. Bu kategorik sınıflandırmada beğeninin arttığını görüyoruz. Grafiksel olarak bakmak istersek:

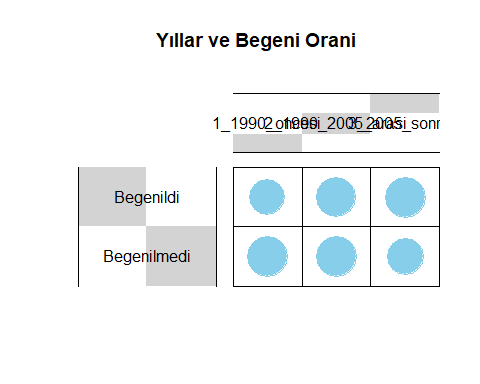
library("gplots")

## Warning: package 'gplots' was built under R version 4.0.5

##   
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':  
##   
## lowess

balloonplot(t(dt), main ="Yıllar ve Begeni Orani ", xlab ="", ylab="",  
 label = FALSE,show.margins = FALSE)

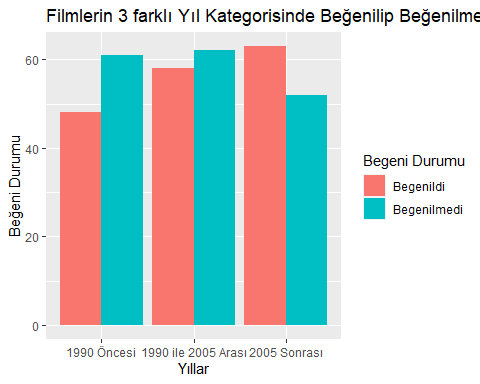


dt\_c<-table(train$begeni,train$yil\_kategorik)  
dtc\_exp <- chisq.test(dt\_c)$expected  
rowcs <- function(i, obs, exp) {  
 sum(((obs[i,] - exp[i,])^2)/exp[i,])  
}  
chi\_dtc<-as.matrix(lapply(seq\_len(nrow(dt\_c)), rowcs, obs = dt\_c, exp = dtc\_exp))  
rownames(chi\_dtc)<-rownames(dt\_c)  
chi\_dtc

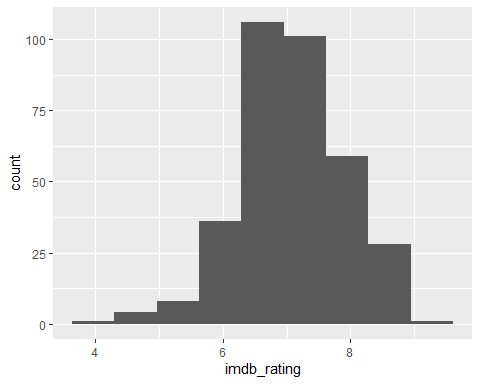
## [,1]   
## Begenildi 1.339012  
## Begenilmedi 1.293103

Farklı bir grafik ile:

ggplot(train,aes(yil\_kategorik, fill=begeni))+  
 geom\_bar(position=position\_dodge())+  
 ggtitle("Filmlerin 3 farklı Yıl Kategorisinde Beğenilip Beğenilmeme Grafiği")+  
 xlab("Yıllar")+  
 ylab("Beğeni Durumu")+  
 scale\_fill\_discrete(name = "Begeni Durumu",  
 labels = c("Begenildi", "Begenilmedi"))+  
 scale\_x\_discrete(labels = c("1\_1990\_oncesi" ="1990 Öncesi","2\_1990\_2005\_arasi"="1990 ile 2005 Arası","3\_2005\_sonrasi"="2005 Sonrası"))

 1990 öncesine baktığımda beğenilmeyen filmlerin beğenilenlerden farkla önde olduğunu görmekteyiz. Bu fark 1990-2005 yılları arasında çıkan filmlerde azalmış. 2005 sonrasında çıkan blockbuster filmlerde ise beğenilenler beğenilmeyenlerden fazla.

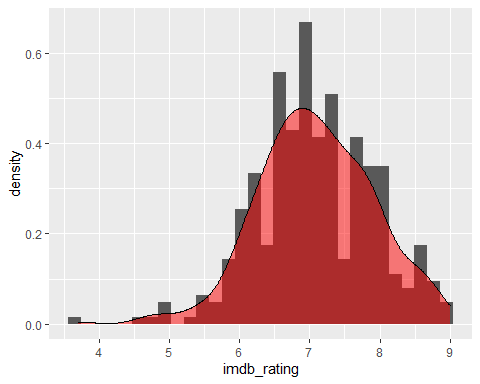
k<-ceiling((log(2\*nrow(train)))+1)   
genislik<-max(train$imdb\_rating)-min(train$imdb\_rating)  
binw<-genislik/k  
  
ggplot(train,aes(imdb\_rating))+  
 geom\_histogram(binwidth=binw)



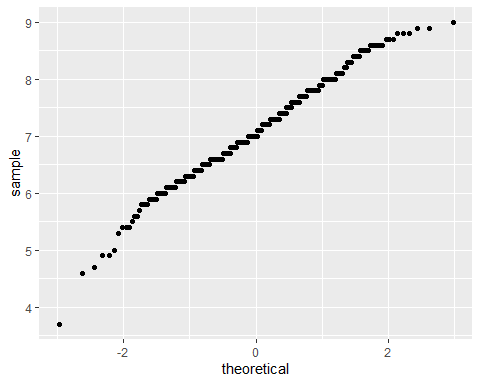
IMDB puanlarının histogramında bir sola çarpıklık var.

ggplot(train,aes(imdb\_rating))+  
 geom\_histogram(aes(y=..density..))+  
 geom\_density(alpha=.5,fill="red")

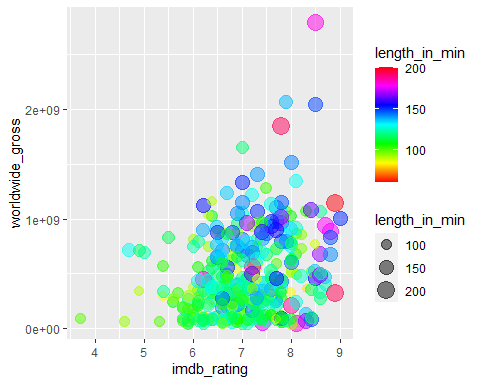
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



ggplot(train, aes(sample=imdb\_rating))+stat\_qq()



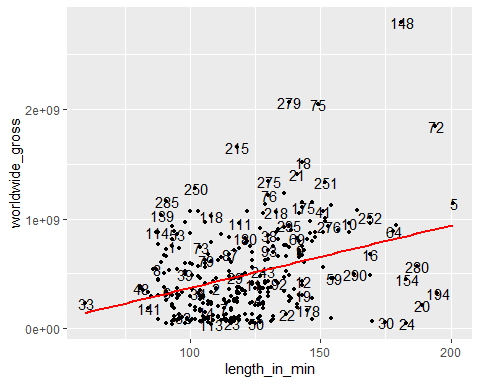
ggplot(train, aes(imdb\_rating, worldwide\_gross, color=length\_in\_min, size=length\_in\_min))+  
 geom\_point(alpha=0.5)+  
 scale\_color\_gradientn(colors =rainbow(unique(train$length\_in\_min))) +  
 theme(legend.position = "right")



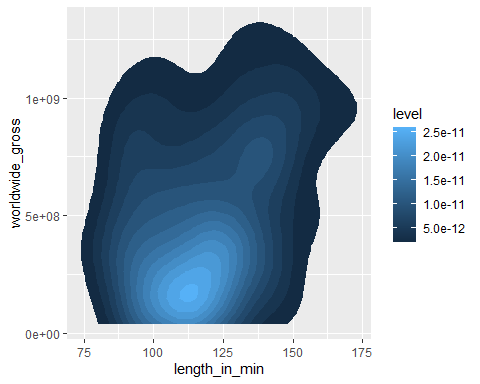
En çok yoğunluğun 7 puan seviyesinde olduğunu ve yine film uzunluklarının da ortalama olarak 120-140 dakika olarak olduğunu görüyoruz.

ggplot(train,aes(x=length\_in\_min,y=worldwide\_gross))+  
 geom\_point(size=1)+  
 geom\_text(label=rownames(train),nudge\_x=0.25,nudge\_y=0.25, check\_overlap=T)+  
 geom\_smooth(method=lm,col="red",se=FALSE)

## `geom\_smooth()` using formula 'y ~ x'



ggplot(train, aes(x=length\_in\_min, y=worldwide\_gross) ) +  
 stat\_density\_2d(aes(fill = ..level..), geom = "polygon")



cor\_train<-train[,c(9,10,3)]  
library(GGally)

## Warning: package 'GGally' was built under R version 4.0.5

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:funModeling':  
##   
## range01

cor(cor\_train)

## worldwide\_gross domestic\_gross imdb\_rating  
## worldwide\_gross 1.0000000 0.9256870 0.2642708  
## domestic\_gross 0.9256870 1.0000000 0.2816848  
## imdb\_rating 0.2642708 0.2816848 1.0000000

library(PerformanceAnalytics)

## Warning: package 'PerformanceAnalytics' was built under R version 4.0.5

## Loading required package: xts

## Warning: package 'xts' was built under R version 4.0.5

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.0.5

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

##   
## Attaching package: 'xts'

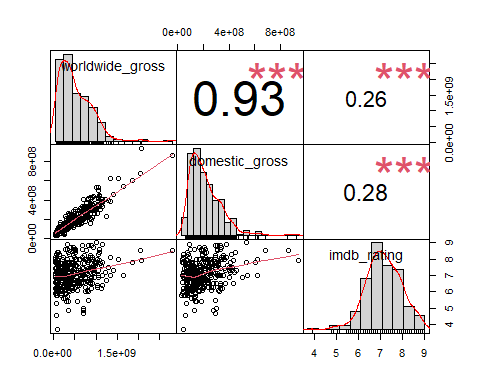
## The following objects are masked from 'package:dplyr':  
##   
## first, last

##   
## Attaching package: 'PerformanceAnalytics'

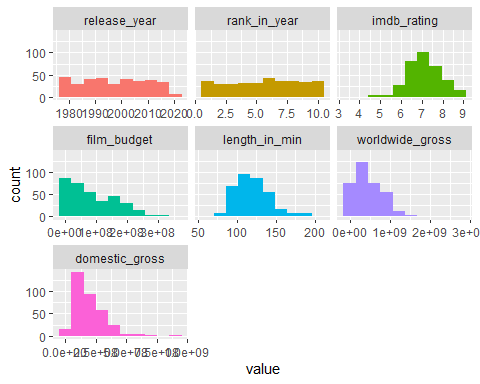
## The following object is masked from 'package:gplots':  
##   
## textplot

## The following object is masked from 'package:graphics':  
##   
## legend

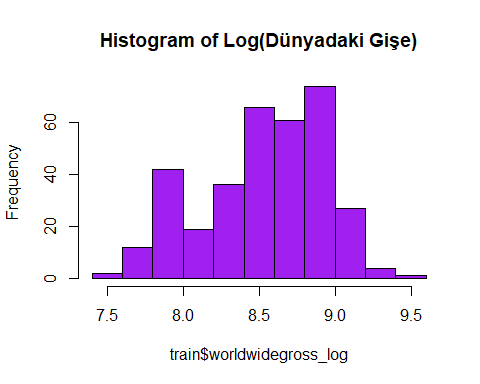
chart.Correlation(cor\_train, histogram=TRUE, pch=19)

 Domestic ile Worldwide arasında güçlü bağlantı gözükmüş olsa bile dağılımlar normal olmadığı için dönüşüm uygulamadan henüz yorumlamayacağım.

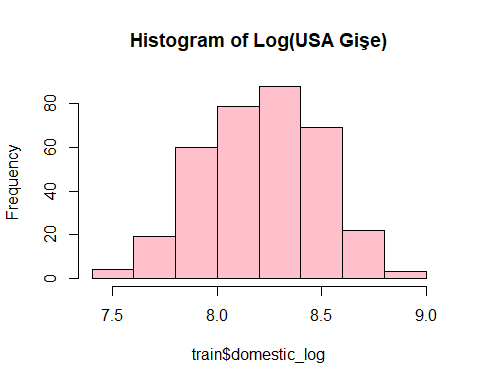
library(funModeling)  
plot\_num(train)

 Dönüşüme ihtiyacı olduklarını görüyorum. Sağa çarpık olanlar için logaritmik dönüşümü deneyeceğim.

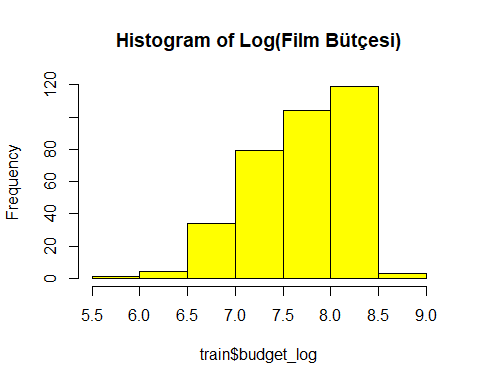
train$worldwidegross\_log<-log10(train$worldwide\_gross)  
hist(train$worldwidegross\_log, col = "purple",main="Histogram of Log(Dünyadaki Gişe)")



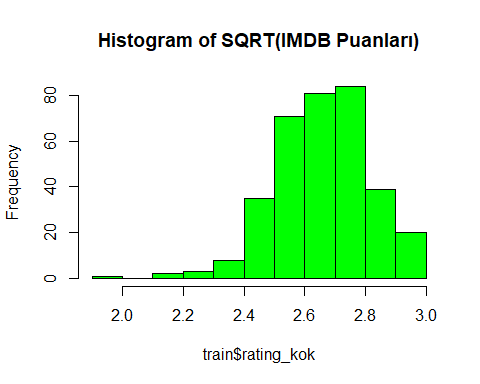
train$domestic\_log<-log10(train$domestic\_gross)  
hist(train$domestic\_log, col = "pink",main="Histogram of Log(USA Gişe)")



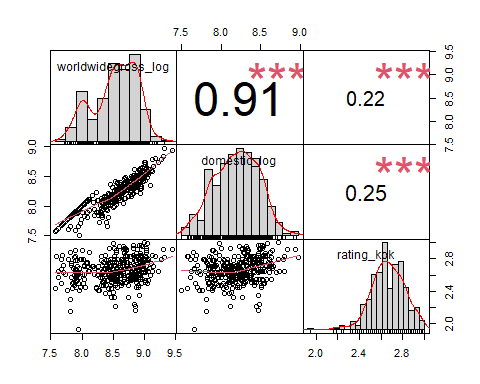
train$budget\_log<-log10(train$film\_budget)  
hist(train$budget\_log, col = "yellow",main="Histogram of Log(Film Bütçesi)")

 Puanlarda sola çarpıklık olduğu için kök dönüşümü deneyeceğim.

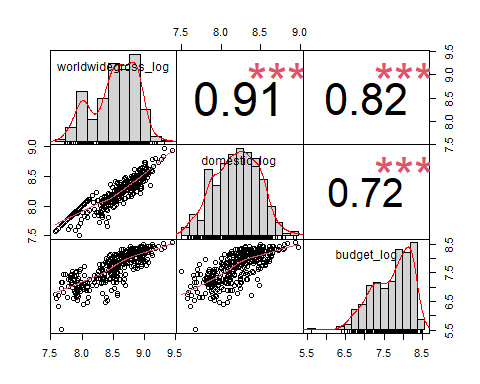
train$rating\_kok<-sqrt(train$imdb\_rating)   
hist(train$rating\_kok, col = "green",main="Histogram of SQRT(IMDB Puanları)")



yeni\_cor\_train<-train[,c(15,16,18)]  
library(GGally)  
library(PerformanceAnalytics)  
chart.Correlation(yeni\_cor\_train, histogram=TRUE, pch=19)

 Dönüşümlü veriler üzerinden korelasyonlara tekrar baktığımızda dünya çapında gişe ile USA gişesinin ciddi şekilde ilişkili olduğunu görüyoruz. Ancak puanın etkilemesi konusunda aynı şeyi söyleyemeyiz.

yeni2\_cor\_train<-train[,c(15,16,17)]  
library(GGally)  
library(PerformanceAnalytics)  
chart.Correlation(yeni2\_cor\_train, histogram=TRUE, pch=19)

 Bütçe ile gişe arasında ilişki var gibi görünüyor. Ancak bütçede dönüşüme rağmen simetrik bir dağılım yakalayamadığımız için kesin bir şey söylemek güç.

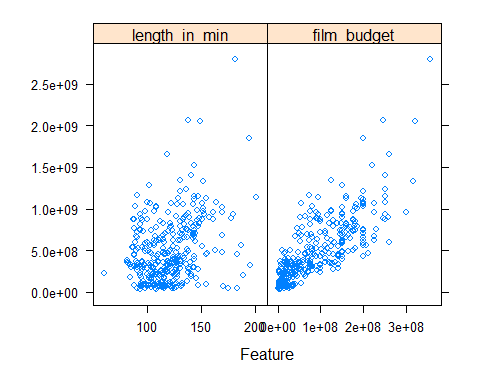
library(caret)

## Warning: package 'caret' was built under R version 4.0.5

##   
## Attaching package: 'caret'

## The following object is masked from 'package:survival':  
##   
## cluster

featurePlot(x=train[,c("length\_in\_min","film\_budget")],y=train$worldwide\_gross)

 Bütçenin filmin uzunluğuna kıyasla gişeyi daha fazla etkiediğini görüyoruz.

## Analizlerin Sonucunda:

Yaptığım analizler sonucu en çok tercih edilen filmlerin genellikle 120 dakika uzunluğuna yakın seviyelerde olduğunu görüyorum. Aslında doğal olarak da bu mantıklı çünkü bunun uzaması durumunda filmin güç içerisinde gösterime girme sayısı azalmakta. Sürenin azalması durumunda da muhtemelen izleyici parasının karşılığını alamayacağını düşündüğü için uzak durmakta. 2 saati bu noktada yapımcılar tarafından amaçlanan süre olarak düşünebiliriz. Onun dışında yine beklediğim şekilde bütçenin de gişeyi etkilediğini görüyorum. Bu da mantıklı çünkü fazla bütçe demek fazla risk demek. Ayrıca verimizde olmasa da büyük bütçeli filmlerde daha büyük reklam kampanyaları yapılmakta. Bu da gişe konusunda geri dönüşünü etkilemektedir. Blockbuster filmler genelde ilk başta Çin’de ya USA’de gösterime girer. Ve buralarda elde ettikleri başarı dünyadaki diğer gişe sayısına da yansır. Yine analizlerim sonucunda bunun doğru olduğunu gördüm. IMDB puanının izleyici açısından çok önemli olmadığını öğrendim. Çok düşük puanlarda elbette gişe sayısında bir düşüklük var ama 6-7 puan seviyesi izleyici için yeterli görünmekte.