LEPlmer2018: 2. Vizuelizacija podataka

Dušica Filipović Đurđević

16 November 2018

## Prvo učitamo jedan mali i čist data frame

* prilagoditi putanji na svom hard disku:

dat = read.table("c:/Sam/Dokumenti/Obuke/LEPlmer2018/cas2.podaci.txt",sep="\t", header=TRUE)

* proverimo dimenzije (broj redova i broj kolona) data frame-a

dim(dat)

## [1] 100 6

* vidimo da postoji 100 redova i 6 kolona (to možemo da vidimo i u Environment)
* da vidimo naslove kolona

colnames(dat)

## [1] "Ispitanik" "Manipulacija" "Rukost" "Nacitanost"   
## [5] "IQ" "RT"

## Napravimo mali uvid u prvih nekoliko redova

head(dat)

## Ispitanik Manipulacija Rukost Nacitanost IQ RT  
## 1 s1 kontrolna grupa levoruk 370 92 1648  
## 2 s2 kontrolna grupa levoruk 80 94 1370  
## 3 s3 kontrolna grupa levoruk 390 96 1663  
## 4 s4 kontrolna grupa levoruk 90 96 1396  
## 5 s5 kontrolna grupa levoruk 220 97 1521  
## 6 s6 kontrolna grupa levoruk 110 97 1409

* vidimo da se u prvoj koloni nalaze šifre ispitanika, u drugoj podatak o pripadnosti nivou grupe u okviru varijable Manipulacija, u trećoj isto to za varijablu Rukost, u četvrtoj Nacitanost, u petoj IQ, u šestoj RT

## Možemo da proverimo tip podataka u kolonama

class(dat$Ispitanik)

## [1] "factor"

class(dat$Manipulacija)

## [1] "factor"

class(dat$Rukost)

## [1] "factor"

class(dat$Nacitanost)

## [1] "integer"

class(dat$IQ)

## [1] "integer"

class(dat$RT)

## [1] "integer"

## Da vidimo koji nivoi varijabli postoje

levels(dat$Ispitanik)

## [1] "s1" "s10" "s100" "s11" "s12" "s13" "s14" "s15" "s16" "s17"   
## [11] "s18" "s19" "s2" "s20" "s21" "s22" "s23" "s24" "s25" "s26"   
## [21] "s27" "s28" "s29" "s3" "s30" "s31" "s32" "s33" "s34" "s35"   
## [31] "s36" "s37" "s38" "s39" "s4" "s40" "s41" "s42" "s43" "s44"   
## [41] "s45" "s46" "s47" "s48" "s49" "s5" "s50" "s51" "s52" "s53"   
## [51] "s54" "s55" "s56" "s57" "s58" "s59" "s6" "s60" "s61" "s62"   
## [61] "s63" "s64" "s65" "s66" "s67" "s68" "s69" "s7" "s70" "s71"   
## [71] "s72" "s73" "s74" "s75" "s76" "s77" "s78" "s79" "s8" "s80"   
## [81] "s81" "s82" "s83" "s84" "s85" "s86" "s87" "s88" "s89" "s9"   
## [91] "s90" "s91" "s92" "s93" "s94" "s95" "s96" "s97" "s98" "s99"

levels(dat$Manipulacija)

## [1] "eksperimentalna grupa" "kontrolna grupa"

levels(dat$Rukost)

## [1] "desnoruk" "levoruk"

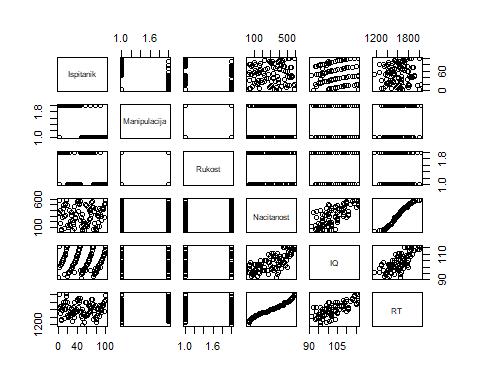
## A možemo i sve odjednom:

str(dat)

## 'data.frame': 100 obs. of 6 variables:  
## $ Ispitanik : Factor w/ 100 levels "s1","s10","s100",..: 1 13 24 35 46 57 68 79 90 2 ...  
## $ Manipulacija: Factor w/ 2 levels "eksperimentalna grupa",..: 2 2 2 2 2 2 2 2 2 2 ...  
## $ Rukost : Factor w/ 2 levels "desnoruk","levoruk": 2 2 2 2 2 2 2 2 2 2 ...  
## $ Nacitanost : int 370 80 390 90 220 110 260 430 320 340 ...  
## $ IQ : int 92 94 96 96 97 97 97 99 100 100 ...  
## $ RT : int 1648 1370 1663 1396 1521 1409 1540 1706 1609 1627 ...

## A možemo i vizuelnim putem:

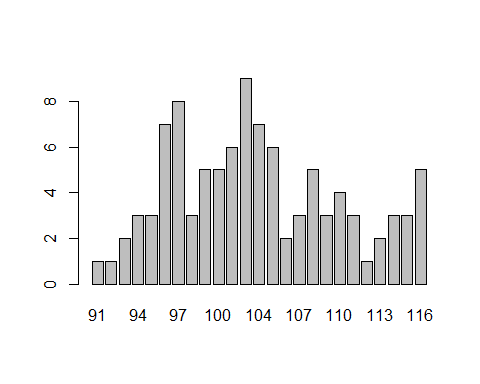
plot(dat)



## Da vizualizujemo jednu varijablu

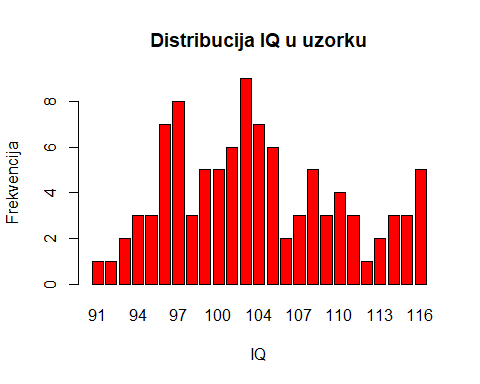
* napravimo prvo frekvencijsku tabelu pomoću table(), a onda tu tabelu prikažemo grafički

barplot(table(dat$IQ))



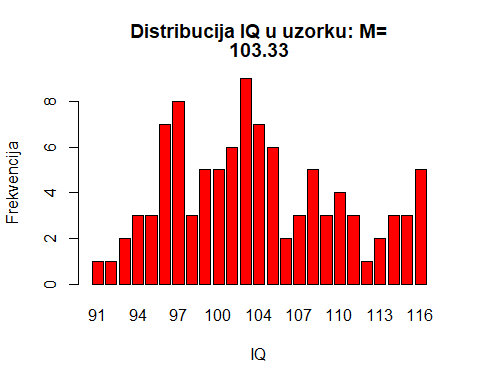
## Dodamo nazive osa, naslov i promenimo boju

barplot(table(dat$IQ),   
 xlab="IQ", # naziv x ose  
 ylab="Frekvencija", # naziv y ose  
 main="Distribucija IQ u uzorku", # naslov na vrhu grafikona  
 col="red") # boja stubića



## Dodamo deskriptivnu statistiku u naslov

barplot(table(dat$IQ),   
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Distribucija IQ u uzorku: M=", mean(dat$IQ)), #kombinujemo tekst i brojeve  
 col="red")



## Grafikon koji smo napravili možemo sačuvati u fajl

png("Grafikon1.png",   
 width=400, height=400)  
  
plot(density(dat$IQ))  
  
dev.off()

## png   
## 2

## Grafički parametri

* Grafikon kao objekat sa kojim R operiše ima niz karakteristika
* Možemo sami upravljati izgledom grafikona tako što definišemo vrednosti grafičkih parametara
* To činimo komandom par(), a unutar zagrada navedemo ono što želimo da definišemo
* Da biste saznali koji sve parametri postoje i kako se menjaju kucajte ovu komandu ili pogledajte Help:

par()

## $xlog  
## [1] FALSE  
##   
## $ylog  
## [1] FALSE  
##   
## $adj  
## [1] 0.5  
##   
## $ann  
## [1] TRUE  
##   
## $ask  
## [1] FALSE  
##   
## $bg  
## [1] "white"  
##   
## $bty  
## [1] "o"  
##   
## $cex  
## [1] 1  
##   
## $cex.axis  
## [1] 1  
##   
## $cex.lab  
## [1] 1  
##   
## $cex.main  
## [1] 1.2  
##   
## $cex.sub  
## [1] 1  
##   
## $cin  
## [1] 0.15 0.20  
##   
## $col  
## [1] "black"  
##   
## $col.axis  
## [1] "black"  
##   
## $col.lab  
## [1] "black"  
##   
## $col.main  
## [1] "black"  
##   
## $col.sub  
## [1] "black"  
##   
## $cra  
## [1] 14.4 19.2  
##   
## $crt  
## [1] 0  
##   
## $csi  
## [1] 0.2  
##   
## $cxy  
## [1] 0.03989363 0.09259264  
##   
## $din  
## [1] 4.999999 3.999999  
##   
## $err  
## [1] 0  
##   
## $family  
## [1] ""  
##   
## $fg  
## [1] "black"  
##   
## $fig  
## [1] 0 1 0 1  
##   
## $fin  
## [1] 4.999999 3.999999  
##   
## $font  
## [1] 1  
##   
## $font.axis  
## [1] 1  
##   
## $font.lab  
## [1] 1  
##   
## $font.main  
## [1] 2  
##   
## $font.sub  
## [1] 1  
##   
## $lab  
## [1] 5 5 7  
##   
## $las  
## [1] 0  
##   
## $lend  
## [1] "round"  
##   
## $lheight  
## [1] 1  
##   
## $ljoin  
## [1] "round"  
##   
## $lmitre  
## [1] 10  
##   
## $lty  
## [1] "solid"  
##   
## $lwd  
## [1] 1  
##   
## $mai  
## [1] 1.02 0.82 0.82 0.42  
##   
## $mar  
## [1] 5.1 4.1 4.1 2.1  
##   
## $mex  
## [1] 1  
##   
## $mfcol  
## [1] 1 1  
##   
## $mfg  
## [1] 1 1 1 1  
##   
## $mfrow  
## [1] 1 1  
##   
## $mgp  
## [1] 3 1 0  
##   
## $mkh  
## [1] 0.001  
##   
## $new  
## [1] FALSE  
##   
## $oma  
## [1] 0 0 0 0  
##   
## $omd  
## [1] 0 1 0 1  
##   
## $omi  
## [1] 0 0 0 0  
##   
## $page  
## [1] TRUE  
##   
## $pch  
## [1] 1  
##   
## $pin  
## [1] 3.759999 2.159999  
##   
## $plt  
## [1] 0.1640000 0.9160000 0.2550001 0.7949999  
##   
## $ps  
## [1] 12  
##   
## $pty  
## [1] "m"  
##   
## $smo  
## [1] 1  
##   
## $srt  
## [1] 0  
##   
## $tck  
## [1] NA  
##   
## $tcl  
## [1] -0.5  
##   
## $usr  
## [1] 0 1 0 1  
##   
## $xaxp  
## [1] 0 1 5  
##   
## $xaxs  
## [1] "r"  
##   
## $xaxt  
## [1] "s"  
##   
## $xpd  
## [1] FALSE  
##   
## $yaxp  
## [1] 0 1 5  
##   
## $yaxs  
## [1] "r"  
##   
## $yaxt  
## [1] "s"  
##   
## $ylbias  
## [1] 0.2

# ili ?par

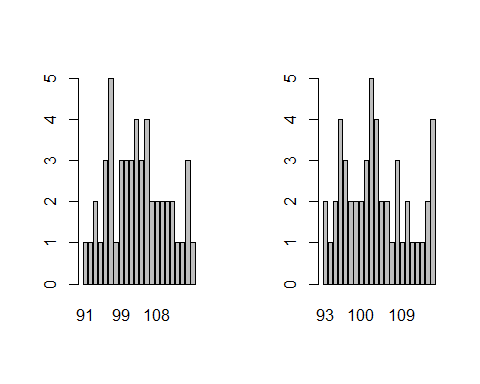
## svoj omiljen seting možete sačuvati i u neku varijablu, pa ga kasnije poziveti

* npr:

# moj\_stil <- par(bg="transparent", col = "red", cex = 0.7)  
# par(moj\_stil)

## Prikažemo raspodelu IQ odvojeno za levoruke i desnoruke

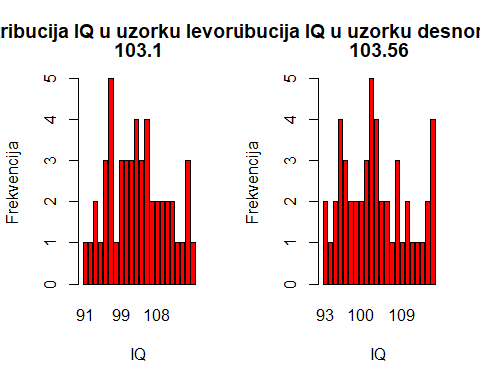
par(mfrow=c(1, 2)) # podesimo grafički parametar na rešetku od 1 reda i 2 kolone  
  
barplot(table(dat[dat$Rukost=="levoruk",]$IQ)) # tražimo IQ samo za levoruke  
   
barplot(table(dat[dat$Rukost=="desnoruk",]$IQ)) # tražimo IQ samo za desnoruke



par(mfrow=c(1,1)) # resetujemo grafički parametar na grid od 1 reda i 1 kolone

* isto to sa oznakama

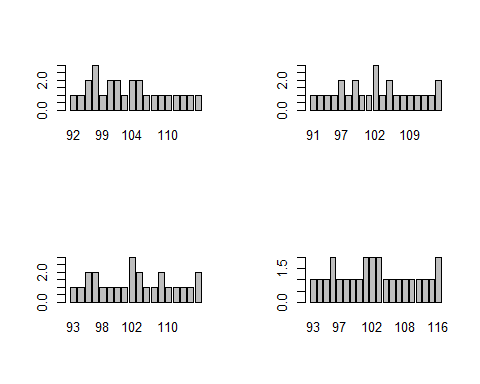
par(mfrow=c(1, 2)) # podesimo grafički parametar na rešetku od 1 reda i 2 kolone  
  
barplot(table(dat[dat$Rukost=="levoruk",]$IQ), # tražimo IQ samo za levoruke  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Distribucija IQ u uzorku levorukih: M=", mean(dat[dat$Rukost=="levoruk",]$IQ)),   
 col="red")  
  
barplot(table(dat[dat$Rukost=="desnoruk",]$IQ), # tražimo IQ samo za desnoruke  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Distribucija IQ u uzorku desnorukih: M=", mean(dat[dat$Rukost=="desnoruk",]$IQ)),   
 col="red")



par(mfrow=c(1,1)) # resetujemo grafički parametar na grid od 1 reda i 1 kolone

## Prikažemo raspodelu IQ odvojeno za levoruke i desnoruke, a unutar svakih odvojeno za kontrolnu i eksperimentalnu grupu

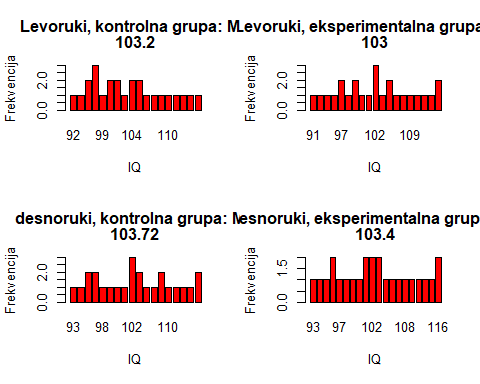
par(mfrow=c(2, 2)) # naprvimo rešetku od dva reda i dve kolone (svaka ćelija za jedan grafikon)  
# grafikoni se crtaju tako da se prvo pupuni prvi red s leva na desno, a potom drugi red  
# ako želite da ih rasporedite tako da se prvo popuni prva kolona s vrha ka dnu,  
# onda kucate par(mfcol=c(2, 2))  
  
barplot(table(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ))  
  
barplot(table(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ))  
  
barplot(table(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ))  
  
barplot(table(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ))



par(mfrow=c(1,1))

## isto to sa oznakama

par(mfrow=c(2, 2))  
barplot(table(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Levoruki, kontrolna grupa: M=", mean(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ)),   
 col="red")  
  
barplot(table(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Levoruki, eksperimentalna grupa: M=", mean(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ)),   
 col="red")  
  
barplot(table(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("desnoruki, kontrolna grupa: M=", mean(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ)),   
 col="red")  
  
barplot(table(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("desnoruki, eksperimentalna grupa: M=", mean(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ)),   
 col="red")



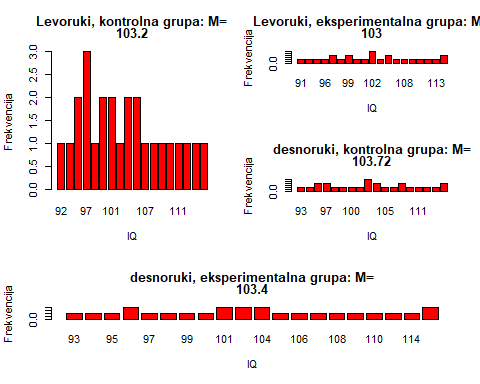
par(mfrow=c(1,1))

## postoji i alternativni način, koji daje malo više slobode

resetka <- matrix(c(1, 2, 1, 3, 4, 4), nrow = 3, ncol = 2, byrow=TRUE)   
# napravimo matricu sa onoliko ćelija  
# tj. redova i kolona koliko želimo da ima grafikona i  
# rasporedimo brojeve u matricu na način na koji želimo da budu raspoređeni grafikoni,  
# pri čemu redosled nizanja grafikona odgovara redosledu kojim ih zadajemo  
  
resetka # da vidimo kako izgleda matrica za raspored

## [,1] [,2]  
## [1,] 1 2  
## [2,] 1 3  
## [3,] 4 4

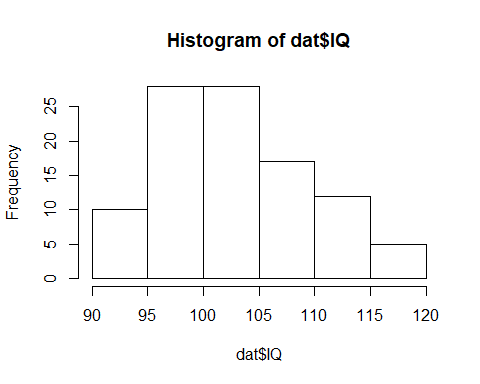
layout(resetka) # tražimo da raspored prati našu matricu  
  
barplot(table(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Levoruki, kontrolna grupa: M=", mean(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ)),   
 col="red")  
  
barplot(table(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Levoruki, eksperimentalna grupa: M=", mean(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ)),   
 col="red")  
  
barplot(table(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("desnoruki, kontrolna grupa: M=", mean(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ)),   
 col="red")  
  
barplot(table(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ),  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("desnoruki, eksperimentalna grupa: M=", mean(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ)),   
 col="red")



layout(1) # resetujemo rešetku za crtanje grafikona

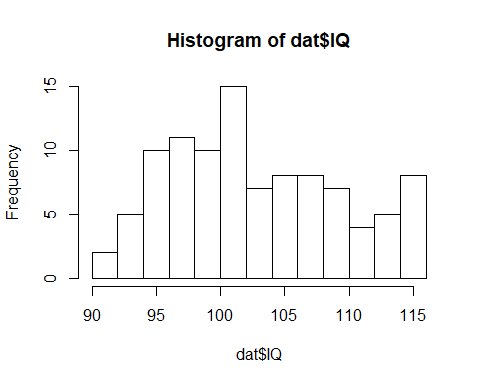
## Napravimo histogram za raspodelu IQ

hist(dat$IQ)



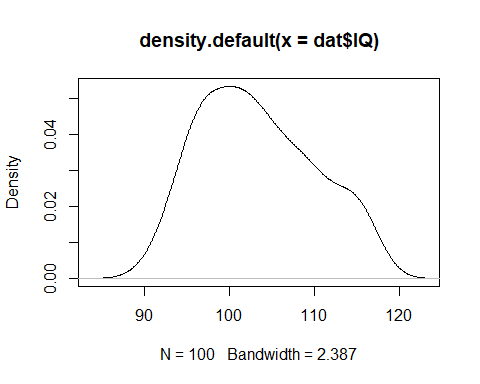
## isto to, ali sami odredimo broj kategorija

hist(dat$IQ, breaks=10)



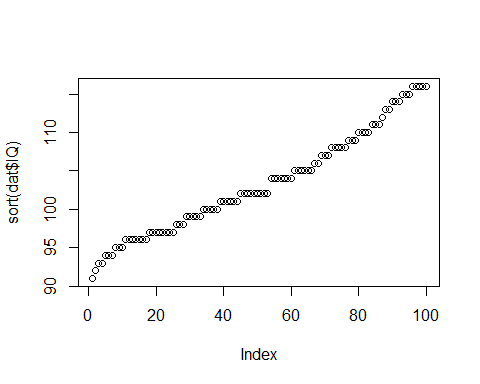
## Prikažemo gustinu raspodele varijable IQ

plot(density(dat$IQ))



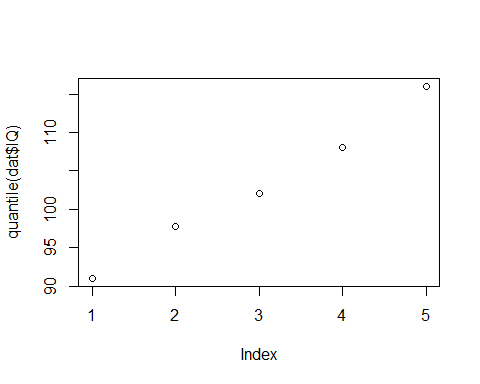
## kumulativno

plot(sort(dat$IQ))



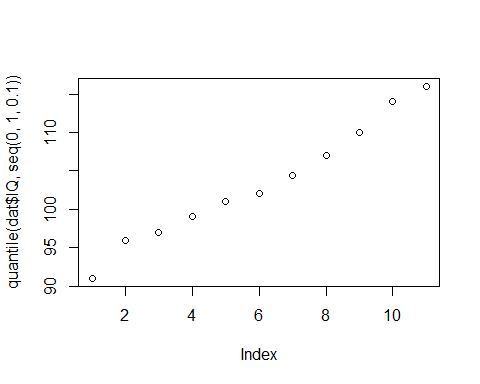
## kvantili

plot(quantile(dat$IQ))



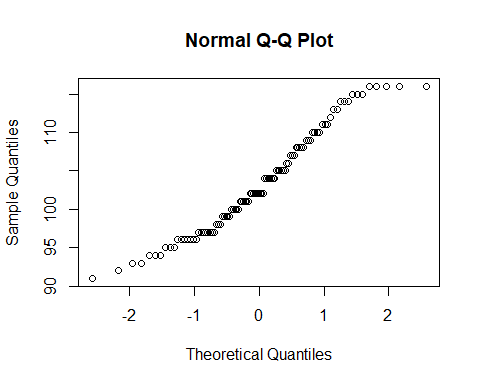
## decili

plot(quantile(dat$IQ, seq(0,1, 0.1)))



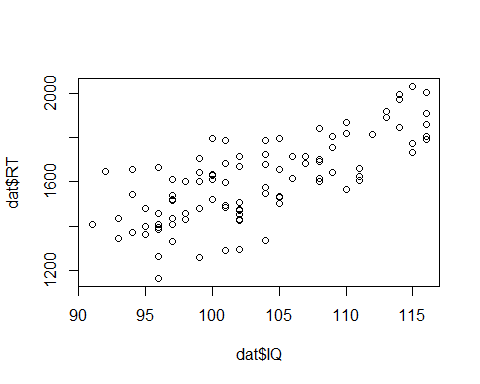
## qqplot

qqnorm(dat$IQ)



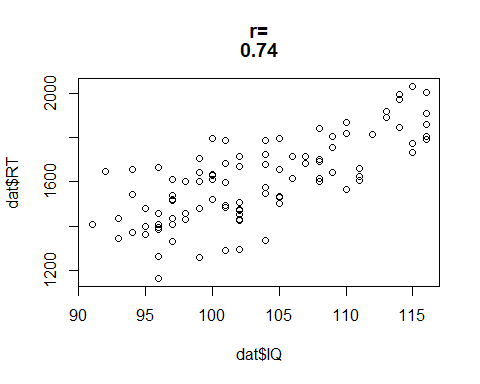
## Scatterplot

plot(dat$IQ, dat$RT) # prvi argument: šta se mapira na x osu, drugi argument: šta se mapira na y osu



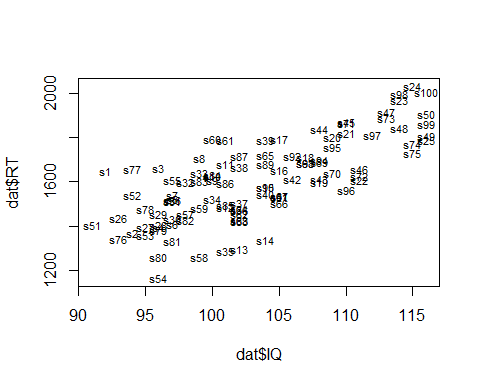
## Dodamo koeficijent korelacije u naslov

x=cor(dat$IQ, dat$RT) # izračunamo koeficijent korelacije između dve varijable  
x= round(x,2) # zaokružimo na dve decimale  
  
plot(dat$IQ, dat$RT,   
 main=c("r=", x)) # iskombinujemo tekst i broj (vrednost r) u naslovu



## Umesto tačaka prikažemo oznake ispitanika

plot(dat$IQ, dat$RT,  
 type="n") # podesimo tip tačaka na "none" -- "ne dodaji ništa"  
  
text(dat$IQ, dat$RT, # komanda za dodavanje reči umesto tačaka na grafikon; prvo damo koordinate  
 as.character(dat$Ispitanik), # onda kažemo šta treba prikazati; konvertujemo u karaktere,   
 #jer ova komanda zahteva vektor stringova (nizova karaktera)  
 cex=0.7) # smanjimo font da bi bilo čitljivije

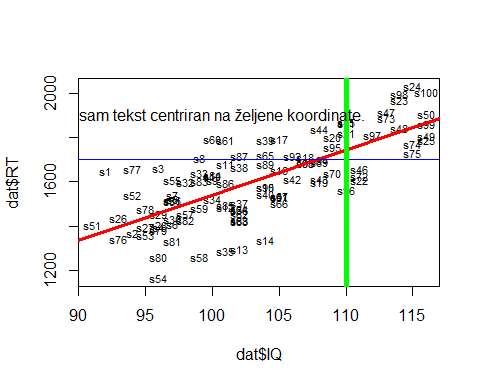


## dodamo regresionu pravu

plot(dat$IQ, dat$RT,  
 type="n") # podesimo tip tačaka na "none" -- "ne dodaji ništa"  
  
text(dat$IQ, dat$RT, # komanda za dodavanje reči umesto tačaka na grafikon; prvo damo koordinate  
 as.character(dat$Ispitanik), # onda kažemo šta treba prikazati; konvertujemo u karaktere,   
 #jer ova komanda zahteva vektor stringova (nizova karaktera)  
 cex=0.7) # smanjimo font da bi bilo čitljivije  
  
  
lm\_dat <- lm(RT ~ IQ, data=dat) # prvo napravimo linearni model  
# dobijemo objekat koji sadrži regresione koeficijente  
coef(lm\_dat) # da vidimo koeficijente

## (Intercept) IQ   
## -497.52579 20.35726

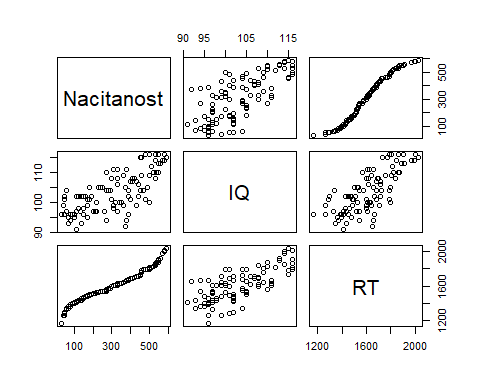
abline(a=coef(lm\_dat)[1], b=coef(lm\_dat)[2])   
# na osnovu tih koeficijenata nacrtamo regresionu liniju  
  
abline(coef(lm\_dat)) # ovo je isto kao komanda u gornjem redu, samo kraće  
  
abline(coef(lm\_dat), lwd = 3) # podebljamo liniju  
  
abline(coef(lm\_dat), lwd = 3, col="red") # linija da bude crvena  
  
abline(h=1700, lwd=1.5, col="blue") # dodamo horizontalnu liniju  
  
abline(v=110, lwd=5, col="green") # dodamo vertikalnu liniju  
  
text(100, 1900, "Ja sam tekst centriran na željene koordinate.")



## Za brzi uvid u odnose među svim numeričkim varijablama

* sada nije impresivno, imamo samo tri numeričke varijable
* kada budemo radili analizu sa velikim brojem numeričkih prediktora, biće vrlo korisno kao prvi uvid u multikolinearnost

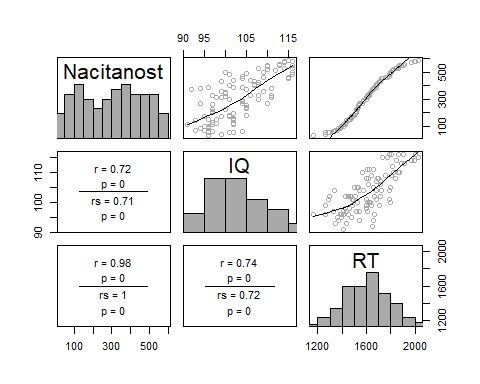
pairs(dat[,c(4,5,6)]) # primeniti komandu na sve redove, na 4. i 5. kolonu data frame-a



## Isto to sa koeficijentima korelacije

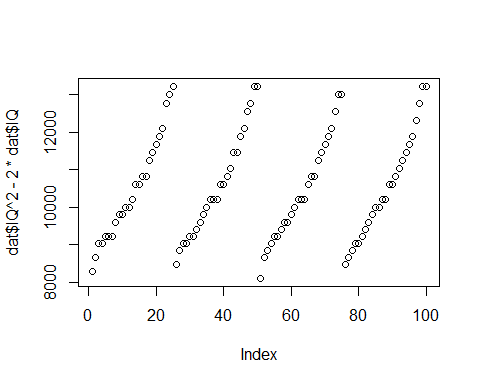
* pozivamo funkviju koja je deo paketa languageR

library(languageR)  
pairscor.fnc(dat[,c("Nacitanost", "IQ","RT" )], hist=TRUE, smooth=TRUE, cex.point=1, col.points="darkgrey")



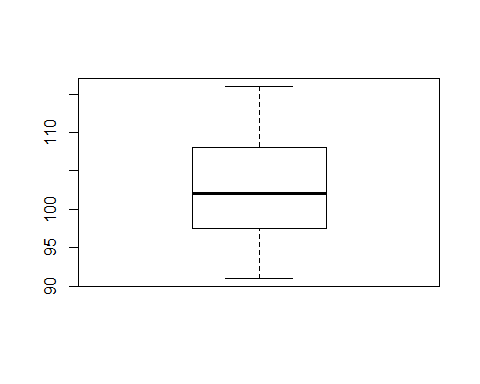
## plotovanje funkcije

plot(dat$IQ^2-2\*dat$IQ)



## Box and whiskers plot

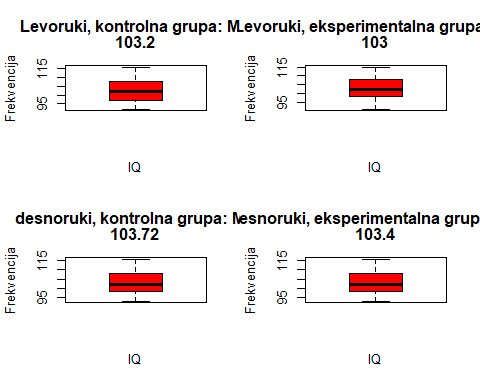
boxplot(dat$IQ)



## Da prikažemo Box and whiskers plot po grupama

* nemojte ovo kucati sada (samo gledamo)

par(mfrow=c(2, 2))  
boxplot(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ,  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Levoruki, kontrolna grupa: M=", mean(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ)),   
 col="red")  
  
boxplot(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ,  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("Levoruki, eksperimentalna grupa: M=", mean(dat[dat$Rukost=="levoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ)),   
 col="red")  
  
boxplot(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ,  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("desnoruki, kontrolna grupa: M=", mean(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="kontrolna grupa",]$IQ)),   
 col="red")  
  
boxplot(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ,  
 xlab="IQ",   
 ylab="Frekvencija",   
 main=c("desnoruki, eksperimentalna grupa: M=", mean(dat[dat$Rukost=="desnoruk" & dat$Manipulacija=="eksperimentalna grupa",]$IQ)),   
 col="red")

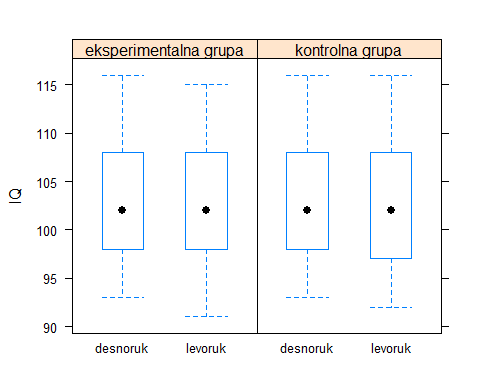


par(mfrow=c(1,1))

## Isto to, sa mnogo manje kucanja - lattice

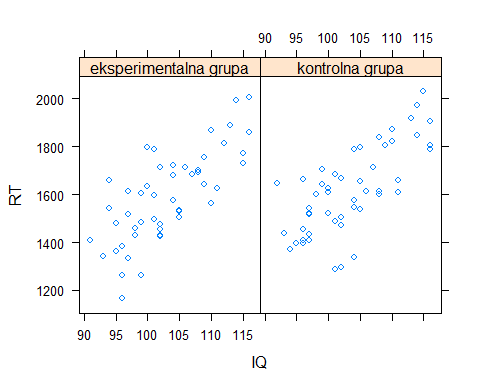
* Formula se čita kao: IQ u zavisnosti od nivoa fakotra Rukost i grupisano po nivoima faktora Manipulacija
* Kada budemo radili sa pravim podacima, vratićemo se na paket lattice (trellis grafiku), da vidimo njene pune moći

library(lattice) # učitamo paket lattice  
bwplot(IQ ~ Rukost | Manipulacija, # prvi argument je formula koja definiše šta prikazati  
 data=dat) # drugi argument je data frame



## U istom paketu: skaterplot po kategorijama

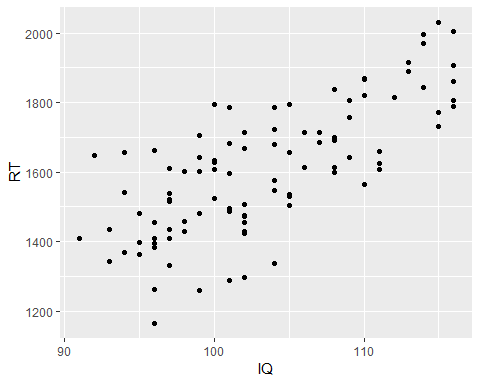
xyplot(RT ~ IQ | Manipulacija,   
 data=dat)



## ggplot2

* Paket napisan na osnovu knjige GRAMMAR OF GRAPHICS (Wilkinson, 1999)
* Trenutno najbolji i najzastupljeniji paket za grafičko prikazivanje
* Svaki grafikon ima barem tri osnovne komponente:
  1. podatke
  2. mapiranje varijabli na koordinate na osama, boju i veličinu: aes
  3. način na koji treba prikazati svaku opseravciju (tačke, linije, stubići…): geom

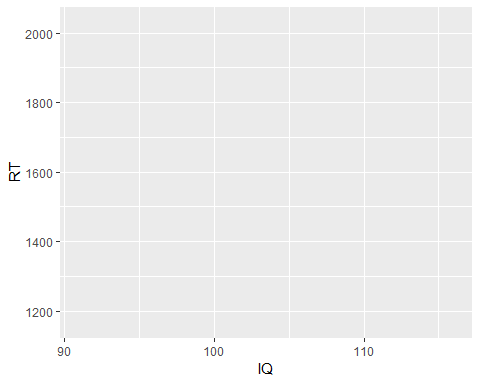
library (ggplot2) # učitamo paket  
  
ggplot(dat, # definišemo koji data frame treba koristiti  
 aes(x = IQ, y = RT)) + # mapiramo varijable na ose  
geom\_point() # definišemo kako treba prikazati vrednosti: ovde kao tačke



## ggplot layers

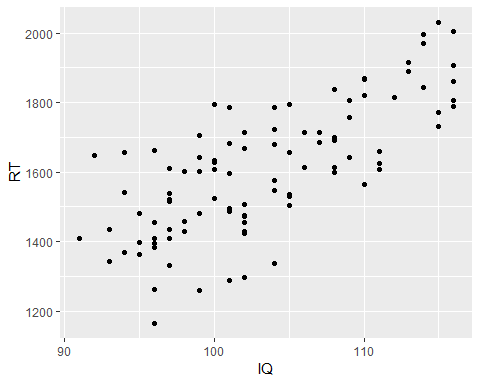
* grafikon možemo da gradimo dodajući sloj po sloj
* koristeći + da dodajemo slojeve

g0 = ggplot(dat, aes(x=IQ, y=RT)) # osnovni grafikon sačuvamo u neku varijablu  
g0 # vidimo da su tu samo ose



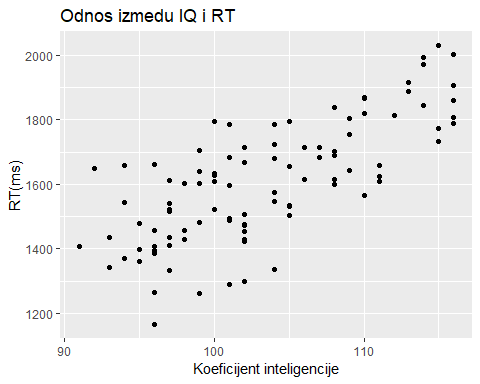
## Dodamo tačke

g1 = g0 + # nove slojeve dodajemo pomoću znaka +  
 geom\_point() # da prikažemo tačke (skatterplot)  
g1



## Dodamo naslov i nazive osa

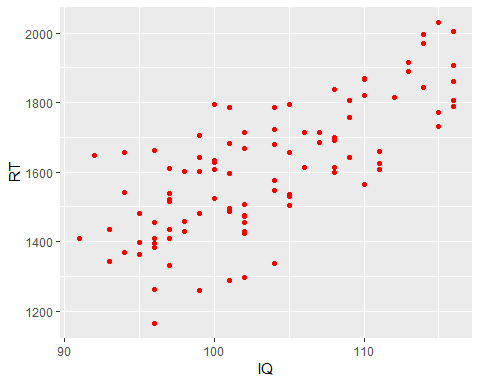
g2 = g1 +   
 ggtitle("Odnos između IQ i RT") + # damo naslov grafikonu  
 xlab("Koeficijent inteligencije") + # damo naslov x osi  
 ylab("RT(ms)") # damo naslov y osi  
  
g2



## geom\_point: boja tačaka

* Da promenimo boju svim tačkama

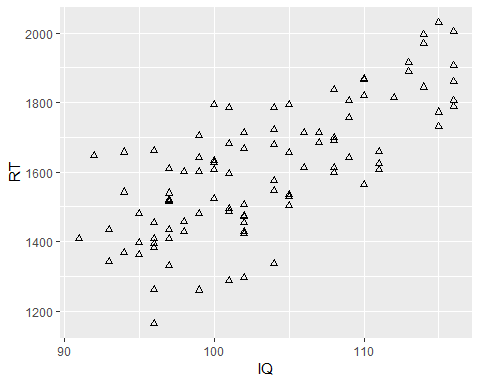
g2 = g0 + geom\_point(colour = "red")   
g2



## geom\_point: oblik tačaka

* Da promenim oblik svim tačkama (postoji “šifrarnik”)

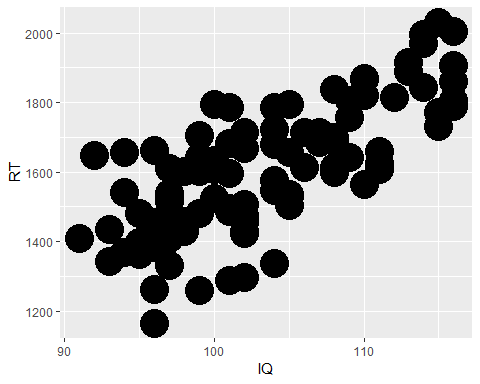
g3 = g0 + geom\_point(shape = 24)  
g3



## geom\_point: veličina tačaka

* Da promenimo veličinu svim tačkama

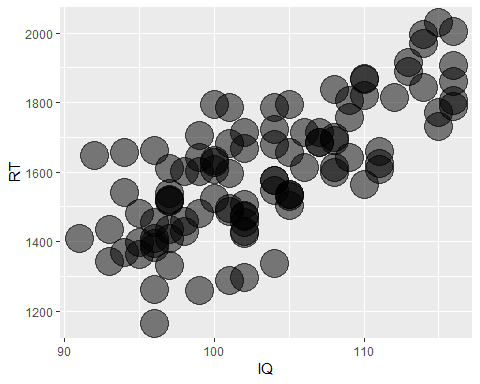
g4 = g0 + geom\_point(size = 10)  
g4



## geom\_point: transparentnost tačaka

* Pogodno za big data, za preopterećene grafikone

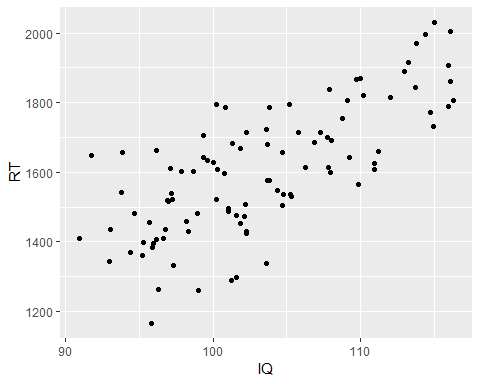
g0 + geom\_point(size = 10, alpha = 0.5)



## geom\_point: da se tačke “razmaknu”

* još jedno rešenje za preopterećene grafikone

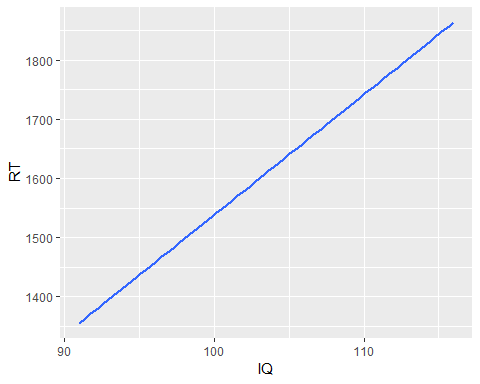
g0 + geom\_jitter()



## Regresiona prava

* samo linija

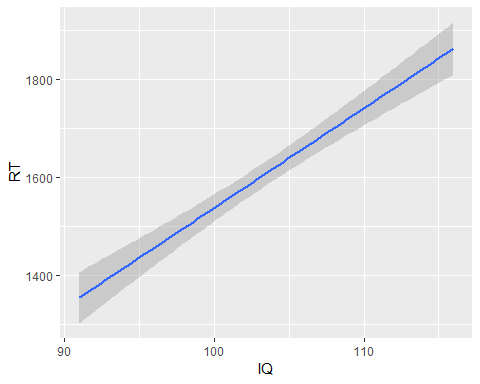
g9 = g0 + geom\_smooth(method = "lm", se = FALSE)  
g9



## Regresiona prava

* linija i 95% CI

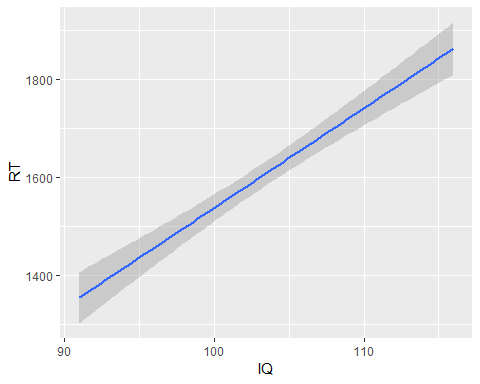
g10 = g0 + geom\_smooth(method = "lm", se = TRUE)  
g10



# Predikcija van opsega zabeleženih vrednosti

* ggplot automatski crta liniju samo unutar zabeleženog raspone
* ako želimo ekstrapolaaciju, moramo to da tražimo:

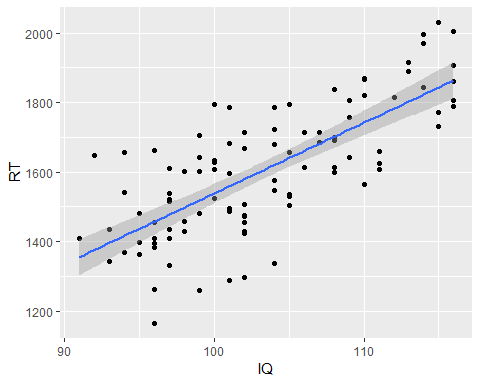
g0 + geom\_smooth(method = "lm", se = TRUE, fullrange=TRUE)



## Regresiona prava

* linija i CI i pojedinačna merenja

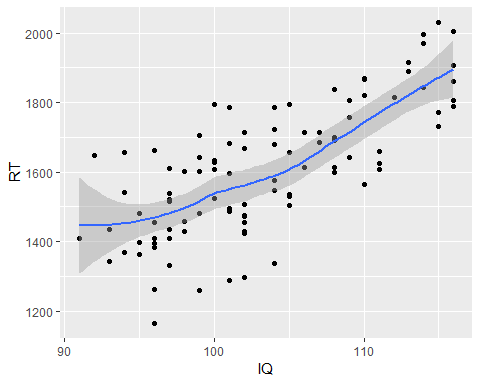
g10 = g1 + geom\_smooth(method = "lm", se = TRUE)  
g10



## Regresiona “kriva”

* za brzi uvid u nelinearnost

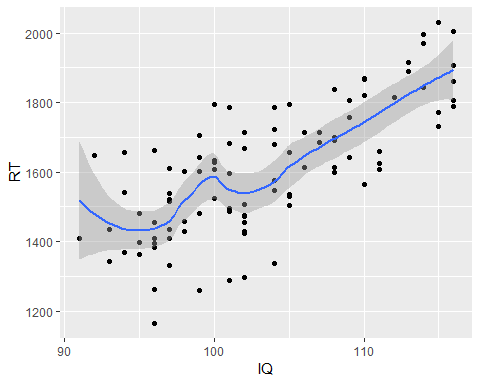
g1 + geom\_smooth(method = "loess", se = TRUE)



## Regresiona “kriva”

* smanjimo veličinu prozora za računanje lokalnih proseka (dozvolimo veću krivudavost)

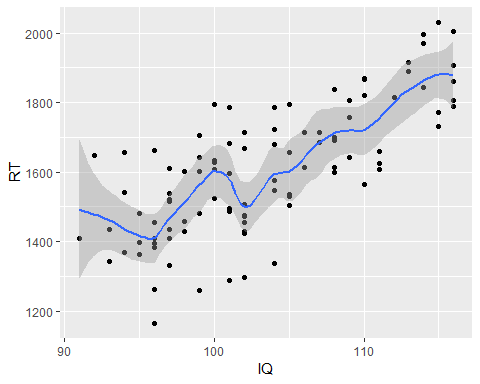
g1 + geom\_smooth(method = "loess", se = TRUE, span = 0.5)



## Regresiona “kriva”

* dozvolimo još veću krivudavost

g1 + geom\_smooth(method = "loess", se = TRUE, span = 0.3)



## Posle prvog upoznavanja sa ggplot2, da se sistematičnije pozabavimo suštinom

* Svaki grafikon ima nekoliko komponenti:
  1. podatke (najbolje čist i uredan data frame); različiti lejeri mogu da prikazuju različite podatke
  2. aes: kako se varijable mapiraju na vizuelne karakteristike; mapiranje varijabli na koordinate na osama, boju i veličinu
  3. geom: naziv geometrijskog objekta koji se koristi za crtanje, način na koji treba prikazati svaku opseravciju (tačke, linije, stubići…)
  4. stat: vrsta statističke transformacije koju treba primeniti (identity, smooth…)
  5. position: način na koji se rešava problem preklapanja, kako organizujemo elemente unutar grafikona (jitternig, stacking, dodging), ali i kako organizujemo višestruke grafikone unutar jedne slike (faceting)
  6. ose: kako ih obeležvamo, orijentišemo i sl.
  7. koordinatni sistem: mi ćemo se baviti dekartovskim, ali postoje i drugi
  8. teme: kakav opšti stil dajemo grafikonu (klasični, minimalni…)

## Layer: data

* podaci koje vizualizujemo
* naš data frame
* obično prvi argument funkcije ggplot()

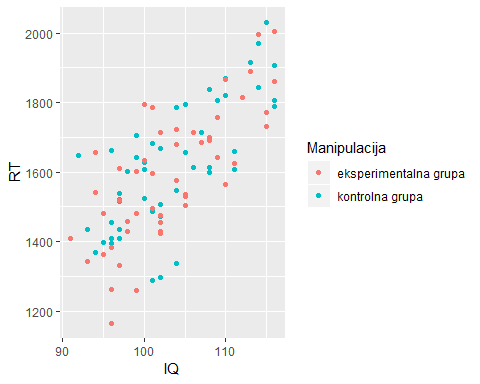
## Layer: aes

* mapiranje varijabli na vizuelne karakteristike
* osnovna: mapiranje varijabli na x i y osu (to smo već videli)

## Layer aes: boja tačaka

* Da stavimo boju tačaka u funkciju neke varijable

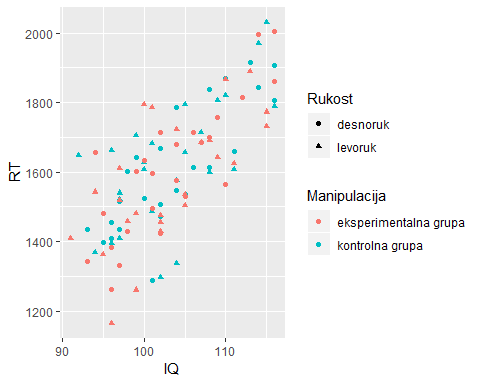
g5 = g1 + aes(colour = Manipulacija)  
g5



## Layer aes: oblik tačaka

* Da stavimo oblik tačaka u funkciju neke varijable

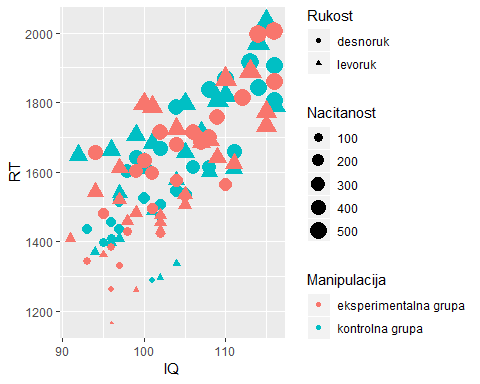
g6 = g5 + aes(shape = Rukost)  
g6



## Layer aes: veličina tačaka

* Da stavimo veličinu tačaka u funkciju neke varijable

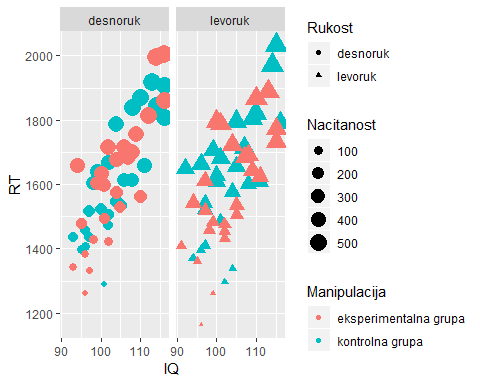
g7 = g6 + aes(size = Nacitanost)  
g7



## faceting

* podela na “podgrafikone” na osnovu nivoa jedne ili više kategoričkih varijabli

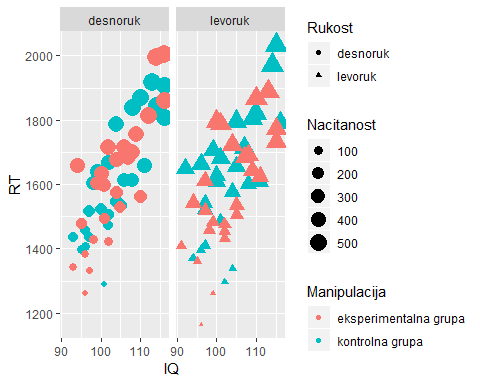
g8 = g7 + facet\_wrap(~Rukost)   
# za svaki nivo varijable Rukost napravi odvojen grafikon,   
# poređa ih u niz i "upakuje" kako je najzgodnije  
g8



## faceting

* slično, ali nivoi isključivo po kolonama grafikona

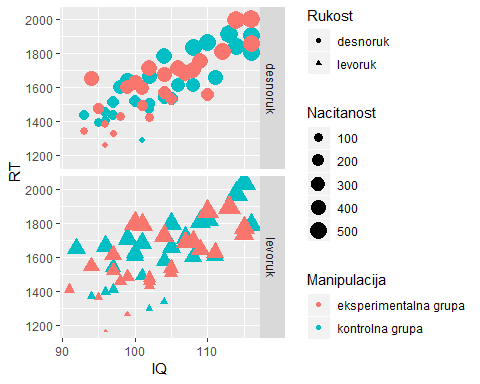
g7 + facet\_grid(.~Rukost)



## faceting

* slično, ali nivoi isključivo po redovima grafikona

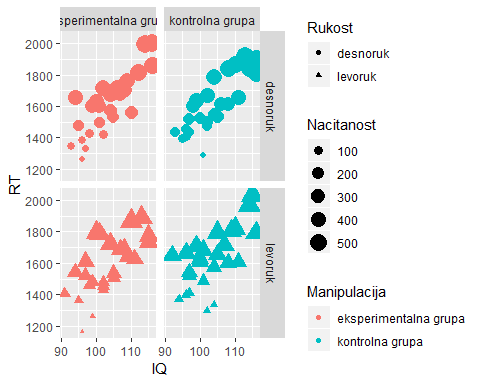
g7 + facet\_grid(Rukost~.)



## faceting

* da napravimo matricu grafikona:

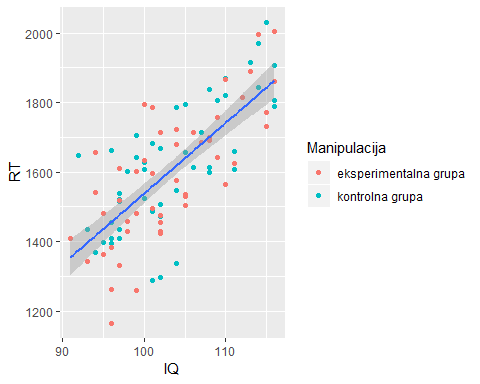
g7 + facet\_grid(Rukost~Manipulacija)



## Malo mudrosti: upotreba boje u zavisnosti od nivoa na koji se odnosi aes()

* samo tačke različitom bojom, linija zajednička

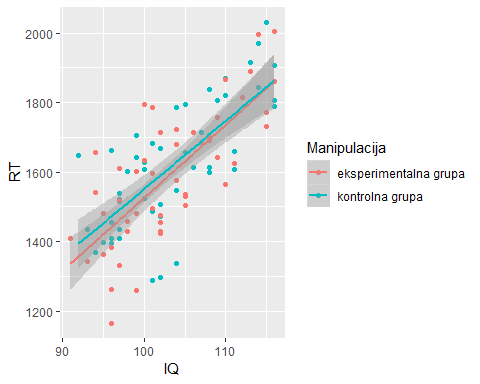
ggplot(dat, aes(x=IQ, y=RT)) +  
 geom\_point(aes(colour = Manipulacija)) + # dodelimo boju   
 # estetici na nivou tačaka  
 geom\_smooth(method = "lm", se = TRUE)



## upotreba boje u zavisnosti od nivoa na koji se odnosi aes()

* tačke različitom bojom i odvojene linije

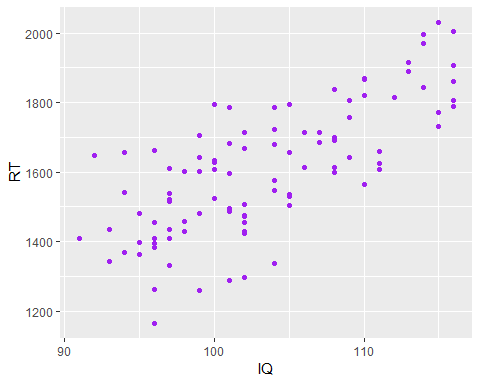
ggplot(dat, aes(x=IQ, y=RT, colour = Manipulacija)) + # dodelimo boju   
 # estetici na globalnom nivou  
 geom\_point() +  
 geom\_smooth(method = "lm", se = TRUE)



## Podešavanje i mapiranje boje: sličan kod, veoma različito značenje

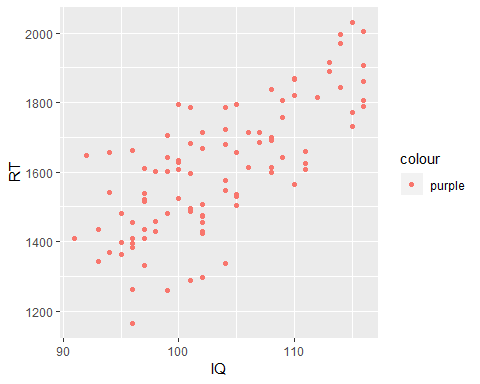
* sada podešavamo boju (dajemo joj sami jednu vrednost)

g1+geom\_point(colour = "purple")



## sada MAPIRAMO boju na vrednost neke varijable (aes se bavi mapiranjem vrednosti varijable na vizuelni domen, ne vizuelnim aspektima samim po sebi)

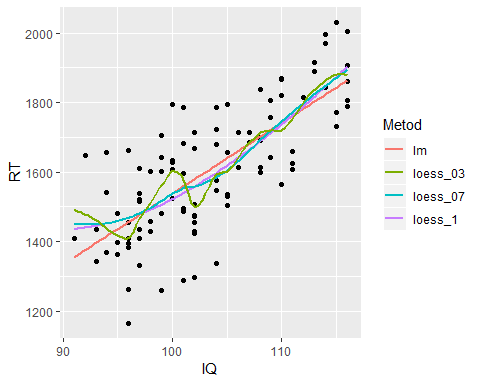
g1+geom\_point(aes(colour = "purple")) # pošto ne postoji varijabla koja se zove "purple", R je napravio takvu varijablu, jer smo mu to tražili



## Ponekad je zgodno iskoristiti boju da uporedimo regresione linije

* Napravimo sami nivoe nove varijable na koje ćemo mapirati boje

g1 +   
 geom\_smooth(aes(colour = "lm"), method = "lm", se = FALSE) +  
 geom\_smooth(aes(colour = "loess\_1"), method = "loess", span = 1, se = FALSE) +  
 geom\_smooth(aes(colour = "loess\_07"), method = "loess", span = 0.7, se = FALSE) +  
 geom\_smooth(aes(colour = "loess\_03"), method = "loess", span = 0.3, se = FALSE) +  
 labs(colour = "Metod")



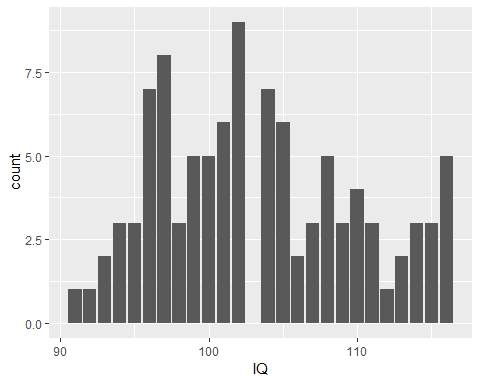
## Layer: geom

* geometrijski elementi koje koristimo za crtanje
* instrukcije kako da se prikažu vrednosti
* po pravilu: geom\_nazivtipaprikaza()

## Layer: geom

* stubići, jedna varijabla

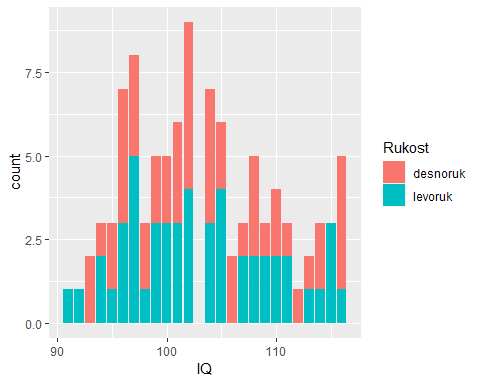
ggplot(dat, aes(x=IQ)) + geom\_bar()



## Layer: geom

* stubići, dve varijable

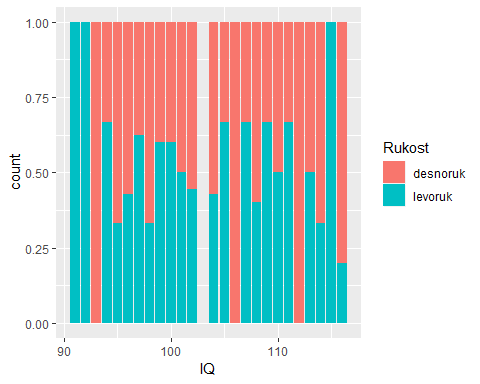
ggplot(dat, aes(x=IQ, fill=Rukost)) + geom\_bar()



## Layer: geom

* stubići, dve varijable, proporcionalno

ggplot(dat, aes(x=IQ, fill=Rukost)) + geom\_bar(position="fill")

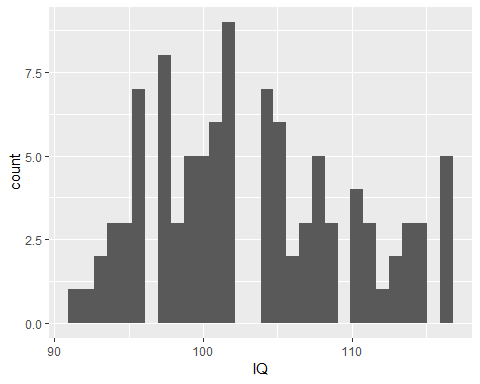


## Layer: geom

* histogram, frekvencije

ggplot(dat, aes(IQ)) + geom\_histogram()

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

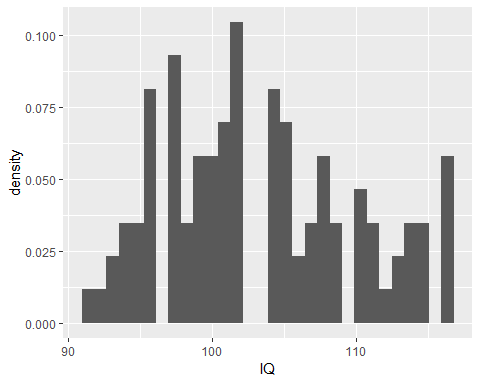


## Layer: geom

* histogram, funkcija gustine verovatnoće

ggplot(dat, aes(IQ)) + geom\_histogram(aes(y = ..density..))

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

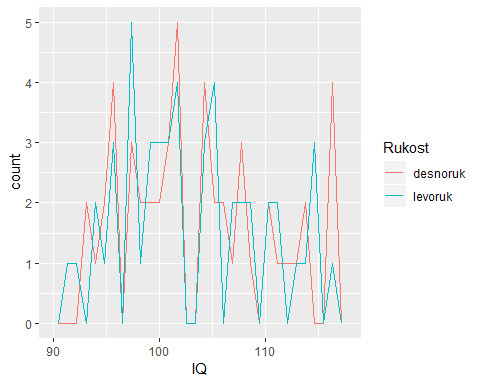


## Layer: geom

* linija, frekvencije

ggplot(dat, aes(IQ, colour = Rukost)) + geom\_freqpoly()

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

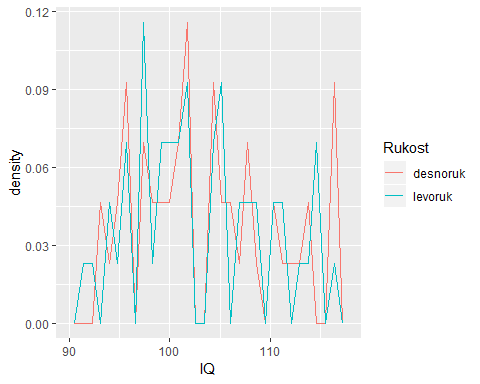


## Layer: geom

* linija,funkcija gustine verovatnoće

ggplot(dat, aes(IQ, colour = Rukost)) + geom\_freqpoly(aes(y = ..density..))

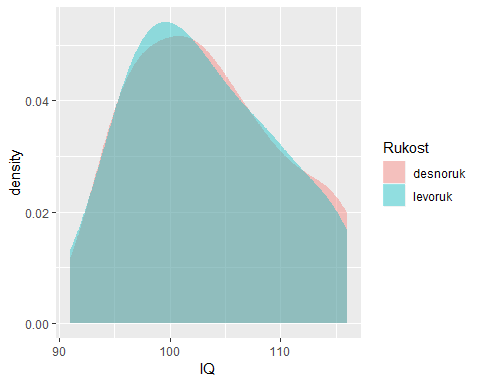
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



## Layer: geom

* funkcije gustine verovatnoće: preklopljene, obojene, delimično transparentne površine

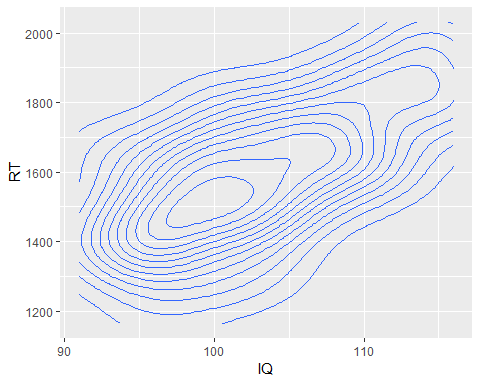
ggplot(dat, aes(IQ, fill = Rukost)) + geom\_density(col = NA, alpha = 0.4)



## Layer: geom

* 2D funkcija gustine raspodele: topografska mapa

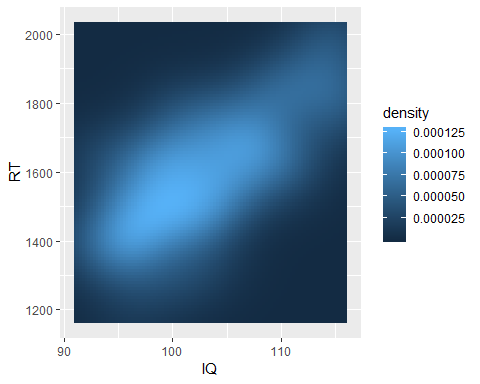
ggplot(dat, aes(x=IQ, y=RT)) + geom\_density\_2d()



## Layer: geom

* 2D funkcija gustine raspodele, boja/senčenje

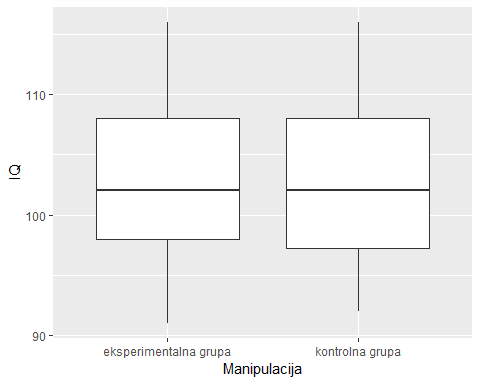
ggplot(dat, aes(x=IQ, y=RT)) + stat\_density\_2d(geom="tile", aes(fill=..density..), contour=FALSE)



## Layer: geom

* box & whiskers

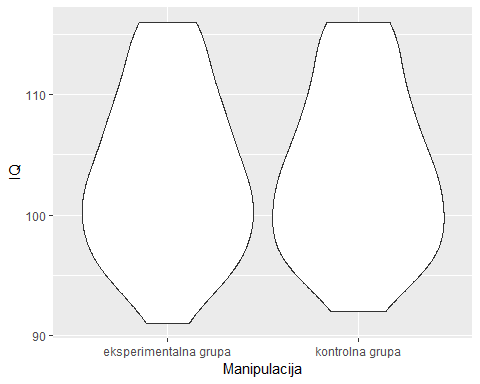
ggplot(dat, aes(x= Manipulacija, y=IQ)) + geom\_boxplot()



## Layer: geom

* violin

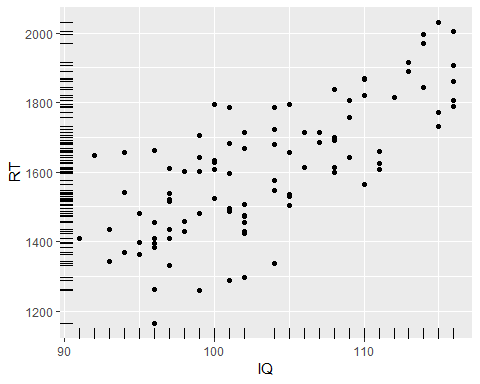
ggplot(dat, aes(x= Manipulacija, y=IQ)) + geom\_violin()



## Layer: geom

* scatterplot smo već sreli, dodamo rug

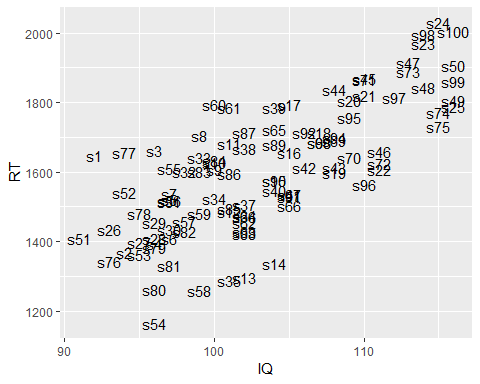
ggplot(dat, aes(x= IQ, y=RT)) + geom\_point() + geom\_rug()



## Layer: geom

* tekst umesto tačaka

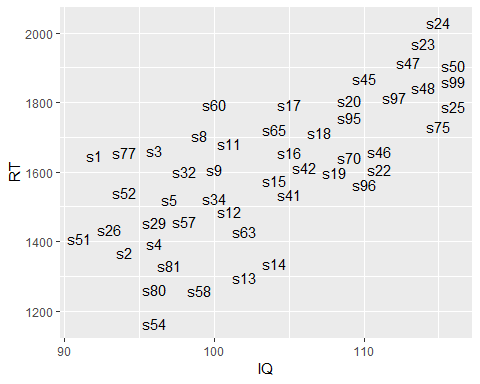
ggplot(dat, aes(x= IQ, y=RT)) + geom\_text(aes(label = Ispitanik))



## Layer: geom

* isto to…
* malo rasteretimo grafikon

ggplot(dat, aes(x= IQ, y=RT)) + geom\_text(aes(label = Ispitanik), check\_overlap = TRUE)

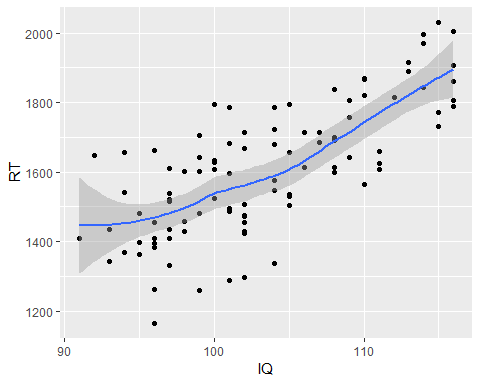


## Layer: geom

* da se vratimo na tačke i dodamo liniju

ggplot(dat, aes(x= IQ, y=RT)) + geom\_point() + geom\_smooth()

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

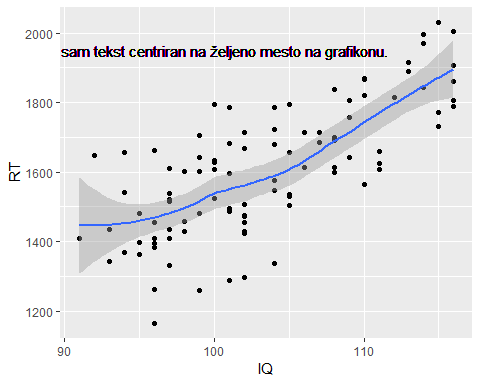


## Layer: geom

* da dodamo tekst negde, po želji

ggplot(dat, aes(x= IQ, y=RT)) + geom\_point() + geom\_smooth() +  
 geom\_text(aes(x=100, y=1950,label = "I ja sam tekst centriran na željeno mesto na grafikonu."))

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

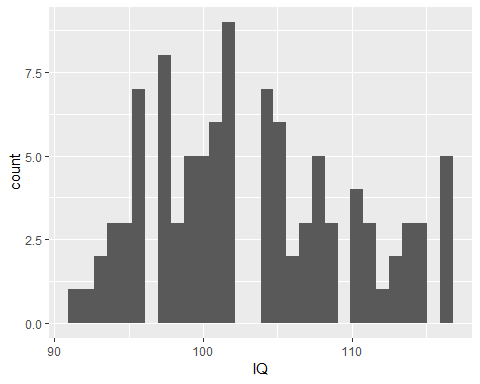


## Layer: stat

* svaki stat je u bliskoj vezi sa geom-om, npr. za:
* stat\_bin() osnovna operacija je brojanje (geom\_bar, geom\_histogram)

ggplot(dat, aes(IQ)) + stat\_bin()

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

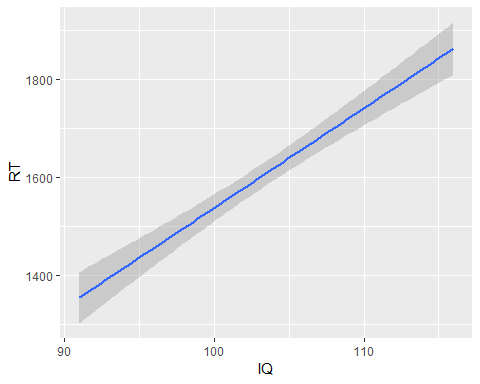


# radi isto što i geom\_bar() i geom\_histogram, jer se ova dva geoma oslanjaju na ovaj stat (brojanje)

## Layer: stat

* stat\_smooth() osnovna operacija je aproksimiranje grupe tačaka (geom\_smooth)

ggplot(dat, aes(IQ,RT)) + stat\_smooth(method="lm")

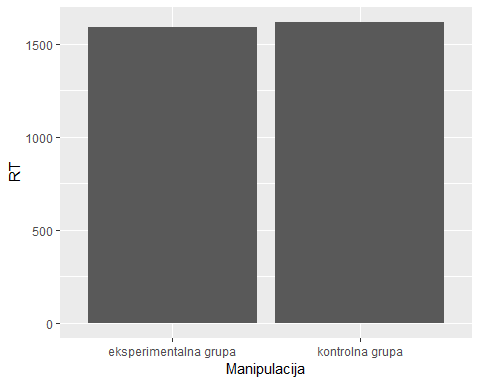


# radi isto što i geom\_smooth(method="lm")

## Layer: stat

* da prikažemo proseke po kategorijama

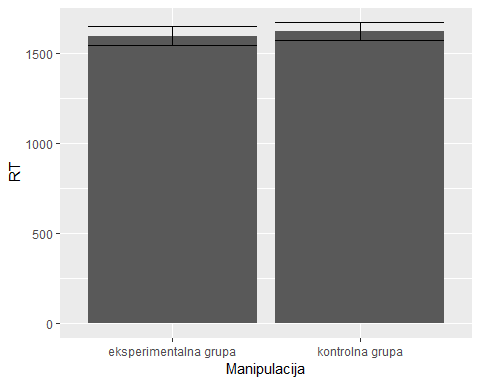
ggplot(dat, aes(x=Manipulacija, y=RT)) +   
 stat\_summary(fun.y = mean, geom = "bar")



## Layer: stat

* da dodamo

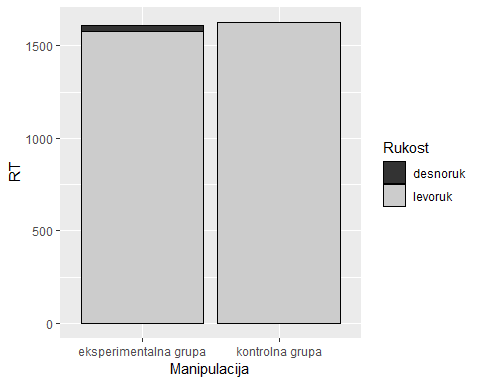
ggplot(dat, aes(x=Manipulacija, y=RT)) +   
 stat\_summary(fun.y = mean, geom = "bar") +   
 stat\_summary(fun.data = mean\_cl\_normal, geom = "errorbar")



## Layer: position

* želim da ukrstim Manipulaciju i Rukost
* ništa se ne vidi, jer ggplot automatski ređa stubiće jedan na drugog (position = “stack”)

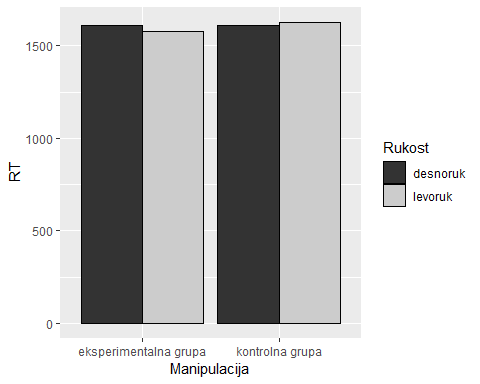
ggplot(dat, aes(x=Manipulacija, y=RT, fill=Rukost)) +   
 stat\_summary(fun.y = mean, geom = "bar", color="black") +   
 scale\_fill\_grey() # da mi stubići budu u nijansama sive, a ne razbih boja



## Layer: position

* da bi se lepo videli, moram da stavim stubiće jedan pored drugog (position = “dodge”)

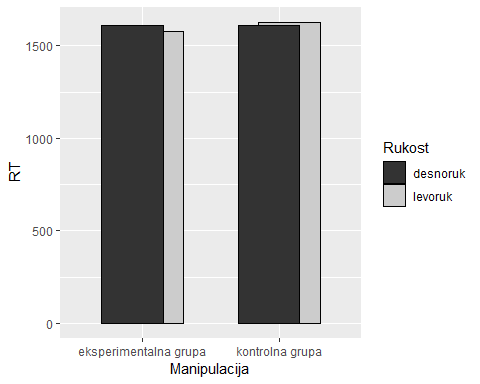
ggplot(dat, aes(x=Manipulacija, y=RT, fill=Rukost)) +   
 stat\_summary(fun.y = mean, geom = "bar", position = "dodge", color="black") +   
 scale\_fill\_grey()



## Layer: position

* ko ne veruje da ih pomeram, da se uveri:

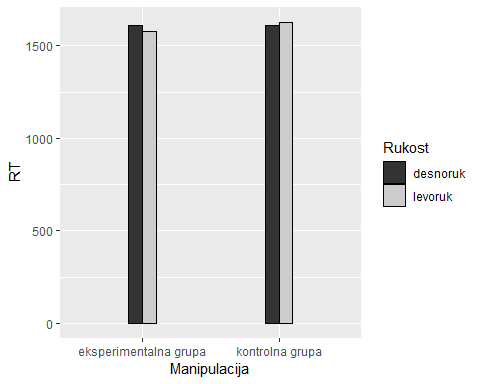
ggplot(dat, aes(x=Manipulacija, y=RT, fill=Rukost)) +   
 stat\_summary(fun.y = mean, geom = "bar", position = position\_dodge(width = 0.30), color="black") +   
 scale\_fill\_grey()



## Layer: position

* da promenim širinu stubića

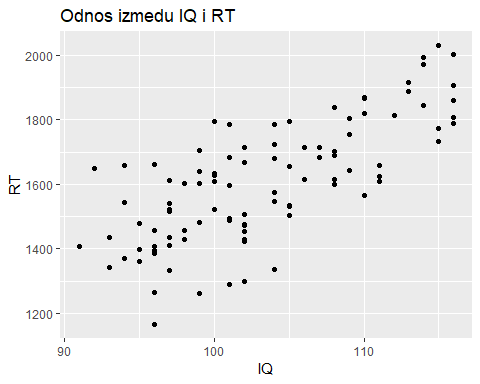
ggplot(dat, aes(x=Manipulacija, y=RT, fill=Rukost)) +   
 stat\_summary(fun.y = mean, geom = "bar", position = "dodge", width = 0.2, color="black") +   
 scale\_fill\_grey()



## Layer: ose

* ovo smo već videli - dajemo naslov grafikonu:

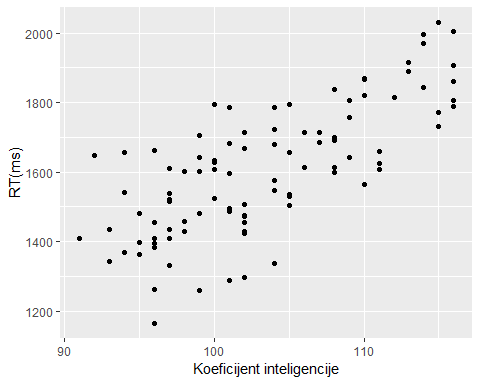
g1 + ggtitle("Odnos između IQ i RT")



## Layer: ose

* ovo smo već videli - dajemo nazive osama:

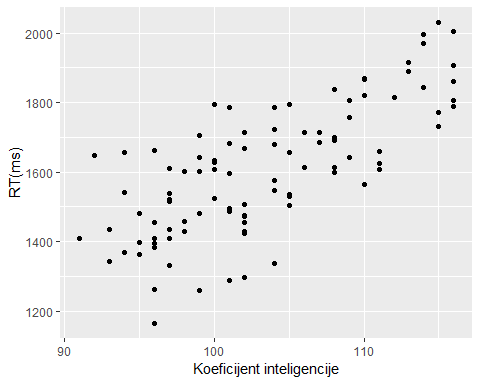
g1 +   
 xlab("Koeficijent inteligencije") +   
 ylab("RT(ms)")



## Layer: ose

* isto to postižemo i sa:

g1 +   
 scale\_x\_continuous("Koeficijent inteligencije") +   
 scale\_y\_continuous("RT(ms)")



## Layer: ose

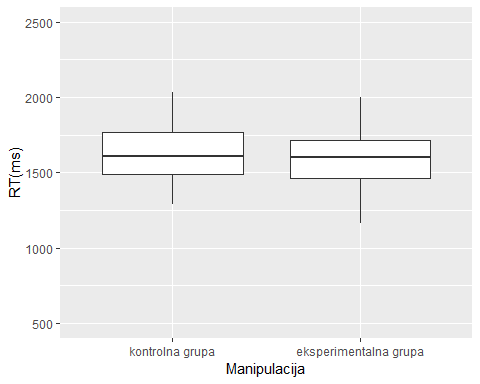
* ako je kategorička varijabla na x osi:

g11 = ggplot(dat, aes(x=Manipulacija, y=RT)) +   
 geom\_boxplot() +  
 scale\_x\_discrete(limits=c("kontrolna grupa", "eksperimentalna grupa")) +  
 # ovo nam dozvoljava da poredjamo nivoe redosledom koji želimo  
 scale\_y\_continuous("RT(ms)")

## Layer: ose

* definisanje raspona osa:

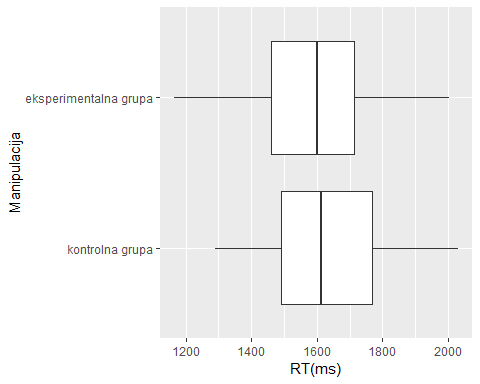
g11 + coord\_cartesian(ylim = c(500, 2500))



## Layer: ose

* rotiranje grafikona

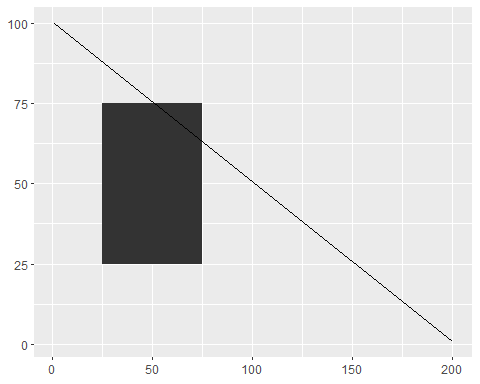
g11 + coord\_flip()



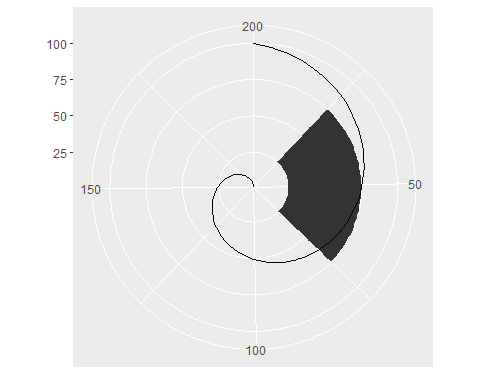
## koordinatni sistemi

* kod koji pokazuje kako prava i pravougaonik izgledaju u polarnom koordinatnom sistemu
* pozajmljen iz Wickham (2015)

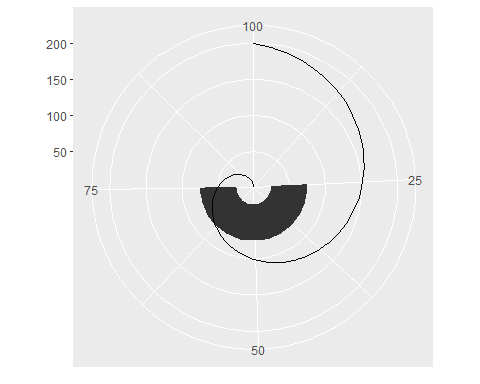
rect <- data.frame(x = 50, y = 50)  
line <- data.frame(x = c(1, 200), y = c(100, 1))  
base <- ggplot(mapping = aes(x, y)) +  
geom\_tile(data = rect, aes(width = 50, height = 50)) +  
geom\_line(data = line) +  
xlab(NULL) + ylab(NULL)  
base



base + coord\_polar("x")



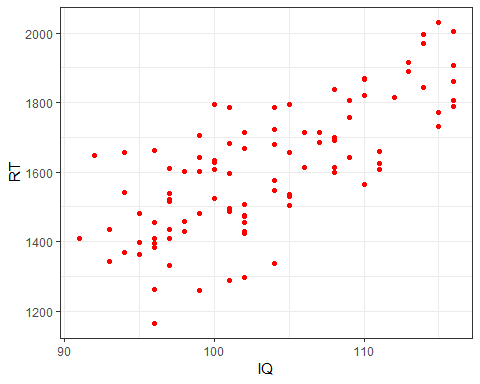
base + coord\_polar("y")



## Layer: teme

* promenimo temu: bw

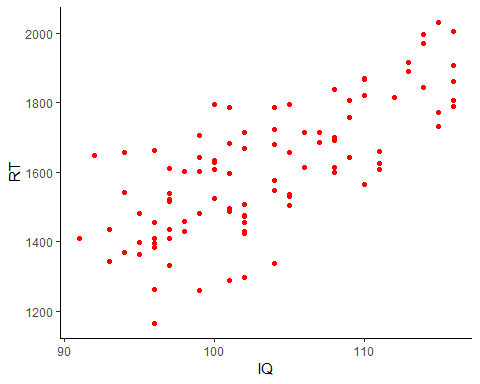
g2 + theme\_bw()



## Layer: teme

* promenimo temu: classic

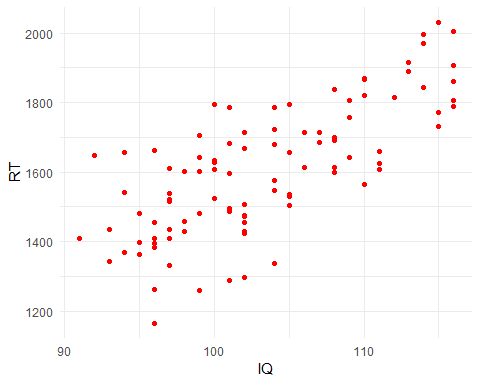
g2 + theme\_classic()



## Layer: teme

* promenimo temu: minimal
* ima ih još
* pored osnovnih tema, postoji i paket ggthemes sa još većim izborom

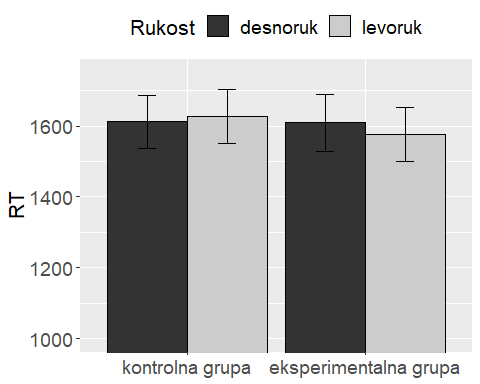
g2 + theme\_minimal()



## Layer: teme

* kako se sve igramo elementima teme,
* za razgledanje kod kuće:

ggplot(data=dat, aes(x=Manipulacija, y=RT, fill=Rukost)) +   
 coord\_cartesian(ylim = c(1000, 1750)) +  
 stat\_summary(fun.y = mean, geom = "bar", position = "dodge", color="black") +   
 scale\_fill\_grey() +  
 scale\_x\_discrete(limits=c("kontrolna grupa", "eksperimentalna grupa")) +  
 theme(axis.title.y = element\_text(size=16), axis.title.x = element\_blank()) +  
 theme(axis.text.x = element\_text(size=14), axis.text.y = element\_text(size=14)) +  
 theme(legend.title = element\_text(size=16)) +  
 theme(legend.text = element\_text(size=14)) +  
 theme(legend.position = "top") +  
 stat\_summary(fun.data = mean\_cl\_normal, geom = "errorbar", position = position\_dodge(width = 0.90), width = 0.2)



## Da sačuvamo grafikon kao sliku:

ggsave("grafikon2.png", width = 5, height = 5)

## Postoji još mnogo mogućnosti

* Razne transformacije podataka
* Dodavanje teksta na željenu lokaciju na grafikonu
* Promena rasporeda elemenata grafikona
* Animirani i interaktivni grafikoni!
* Čak i programiranje u ggplot!

## Za sve što ne znate napamet (tj. za većinu stvari):

* guglajte frazu koja opisuje to što želite da izvedete (obilje blogova)
* <https://www.rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>