

```

##-----
## 5ab. "Shop" dataset
##-----
Shop = read.table("http://edu.sablab.net/data/txt/shop.txt",
                  header=T, sep="\t")

str(Shop)
head(Shop)
summary(Shop)
fix(Shop)

## show only records for customers with Visa card.
Shop[Shop$Payment == "Visa",]

## write only records for customers with Visa card.
write.table(Shop[Shop$Payment == "Visa",],
            "temp.txt", row.names=F, col.names=T)

##-----
## 7ab. Visualization
##-----
Mice = read.table("http://edu.sablab.net/data/txt/mice.txt",
                  header=T, sep="\t")

str(Mice)

x11()
plot(density(Mice$Ending.weight[Mice$Sex == "f"]), col="#FF4444", lwd=2,
     main="Weight")
lines(density(Mice$Ending.weight[Mice$Sex == "m"]), col="#4444FF", lwd=2)

x11()
boxplot(Mice$Bleeding.time ~ Mice$Strain, las=2)

##-----
## 8ab. Visualization
##-----
Mice = read.table("http://edu.sablab.net/data/txt/mice.txt",
                  header=T, sep="\t")

str(Mice)

## 8a
sum(Mice$Bleeding.time > 120, na.rm = T)

## 8b
x = Mice$Starting.weight
p0 = c(mean(x), median(x), sd(x), mad(x))
x[1] = 1000
p1 = c(mean(x), median(x), sd(x), mad(x))

p0
p1

##-----
## 9a. PCA of ALL
##-----
Data = read.table("http://edu.sablab.net/data/txt/all_data.txt",
                  as.is=T, header=T, sep="\t")
str(Data) ## see here - experiments are in columns!!!

## create classes: 1 - normal, 2 - cancer
classes = integer(ncol(Data[, -c(1,2)]))
idx.norm = grep("normal", names(Data[, -c(1,2)]))
idx.all = grep("ALL", names(Data[, -c(1,2)]));

```

```

classes[idx.norm] = 1
classes[idx.all] = 2

## PCA
PC = prcomp(t(Data[,-c(1,2)]))
str(PC)
x11()
plot(PC$x[,1],PC$x[,2],col = classes, pch=19)

## 3D
plot3d(PC$x[,1],PC$x[,2],PC$x[,3],col = classes, type="s")

##-----
## 11a. Statistical Tests
##-----

Data = read.table("http://edu.sablab.net/data/txt/mice.txt",header=T,sep="\t")
## check the structure
str(Data)

plot(Data$Ending.weight ~ Data$Strain, las=2)
plot(Data$Weight.change ~ Data$Strain, las=2)

t.test(Data$Ending.weight[Data$Strain=="C58/J"],
Data$Ending.weight[Data$Strain=="CAST/EiJ"])
t.test(Data$Weight.change[Data$Strain=="C58/J"],
Data$Weight.change[Data$Strain=="CAST/EiJ"])

wilcox.test(Data$Ending.weight[Data$Strain=="C58/J"],
Data$Ending.weight[Data$Strain=="CAST/EiJ"])
wilcox.test(Data$Weight.change[Data$Strain=="C58/J"],
Data$Weight.change[Data$Strain=="CAST/EiJ"])

var.test(Data$Ending.weight[Data$Strain=="C58/J"],
Data$Ending.weight[Data$Strain=="CAST/EiJ"])
var.test(Data$Weight.change[Data$Strain=="C58/J"],
Data$Weight.change[Data$Strain=="CAST/EiJ"])

##-----
## 12a. Linear Regression
##-----

Data = read.table("http://edu.sablab.net/data/txt/leukemia.txt",
header=T,sep="\t",quote="\"")
str(Data)

x11()
plot(density(Data$WBC))

## log WBC and time
Data$WBC = log10(Data$WBC)

x11()
plot(density(Data$WBC))

## separate to two datasets

```

```

data1 =Data[Data$AG == "Positive",]
data2 =Data[Data$AG == "Negative",]

##-----
## Q1. Build scatter plot.

## build scatter plots
x11()
par(mfcol=c(1,2))
plot(data1$WBC,data1$Survival,pch=19,col=3)
plot(data2$WBC,data2$Survival,pch=19,col=4)

##-----
## Q2. Linear model. Estimations.

## build linear models
lm1 = lm(Survival ~ WBC, data1)
lm2 = lm(Survival ~ WBC, data2)
summary(lm1)
summary(lm2)

## estimate survial for patient with WBC = 20000

wbc = log2(20000)
surv1 = lm1$coefficients[1]+lm1$coefficients[2]*wbc

## build scatter plots
x11()
plot(data1$WBC,data1$Survival,pch=19,col=3, main="Survival")
lines(data1$WBC, predict(lm1,int = "confidence")[,1],col=4,lwd=2)
lines(data1$WBC, predict(lm1,int = "confidence")[,2],col=4)
lines(data1$WBC, predict(lm1,int = "confidence")[,3],col=4)
points(wbc,surv1,col=2,pch=19)

lines(data1$WBC, predict(lm1,int = "pred")[,2],col=2)
lines(data1$WBC, predict(lm1,int = "pred")[,3],col=2)

##-----
## 12b. ANOVA
##-----

summary(Data)

WBC.factor = character(ncol(Data))
WBC.factor[Data$WBC <= median(Data$WBC)] = "Low"
WBC.factor[Data$WBC > median(Data$WBC)] = "High"
WBC.factor = as.factor(WBC.factor)

Data1 = cbind(Data,WBC.factor)

model = aov(Survival ~ WBC.factor + AG + WBC.factor*AG, Data1)
summary(model)

```