

STK1000 - oblig 1

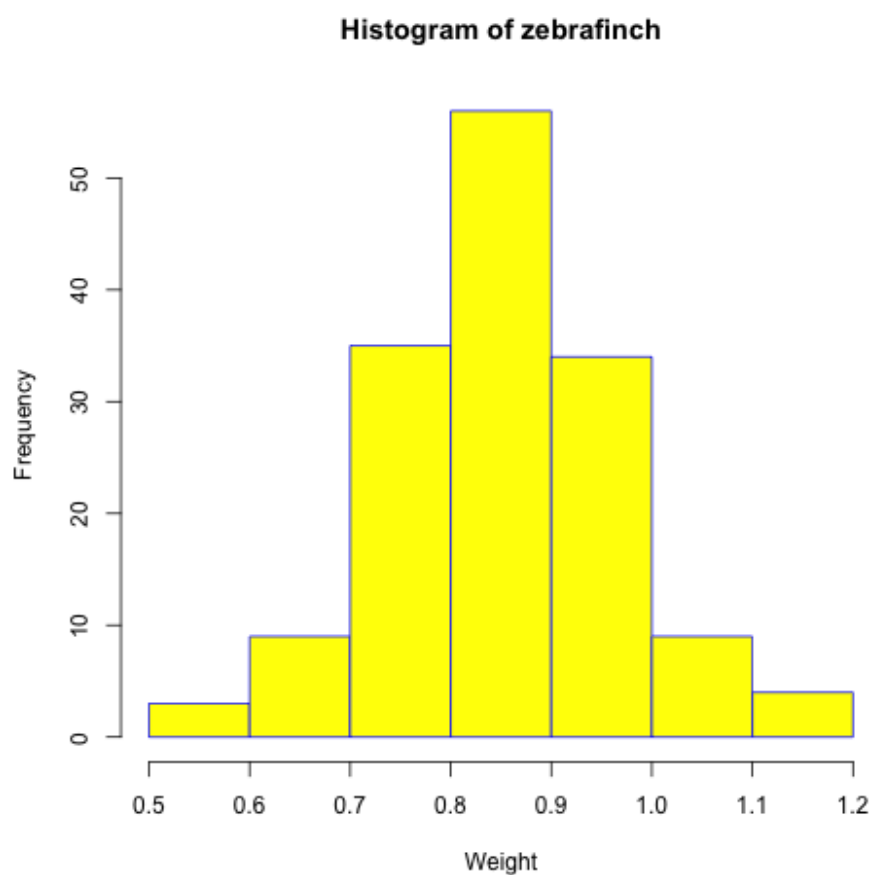
Mikael B. Kiste

19. september 2018

1

a

Her er et histogram av BMR verdiene til sebrafinkene. Histogrammet ser normalfordelt ut, med en god del sebrafinker som har en BMR verdi nær gjennomsnittet og et kraftig avtagende antall finker jo lenger man går vekk fra gjennomsnittet.



Figur 1: Histogram av BMR-verdiene til 150 zebrafisker

b

Ved å bruke kommandoen `mean` i R får jeg gjennomsnitt og `median` gir meg median.

$$\begin{aligned}\bar{x} &= 0.8485003 \text{mL} \frac{\text{O}_2}{\text{m}} \\ \text{mean} &= 0.8397846 \text{mL} \frac{\text{O}_2}{\text{m}}\end{aligned}$$

Gjennomsnittet tar summen av måleverdien til alle datapunktene før det deler på antall datapunkter mens median er verdien på det midterste elementet i en ordnet (numerisk økende) liste av datapunktene.

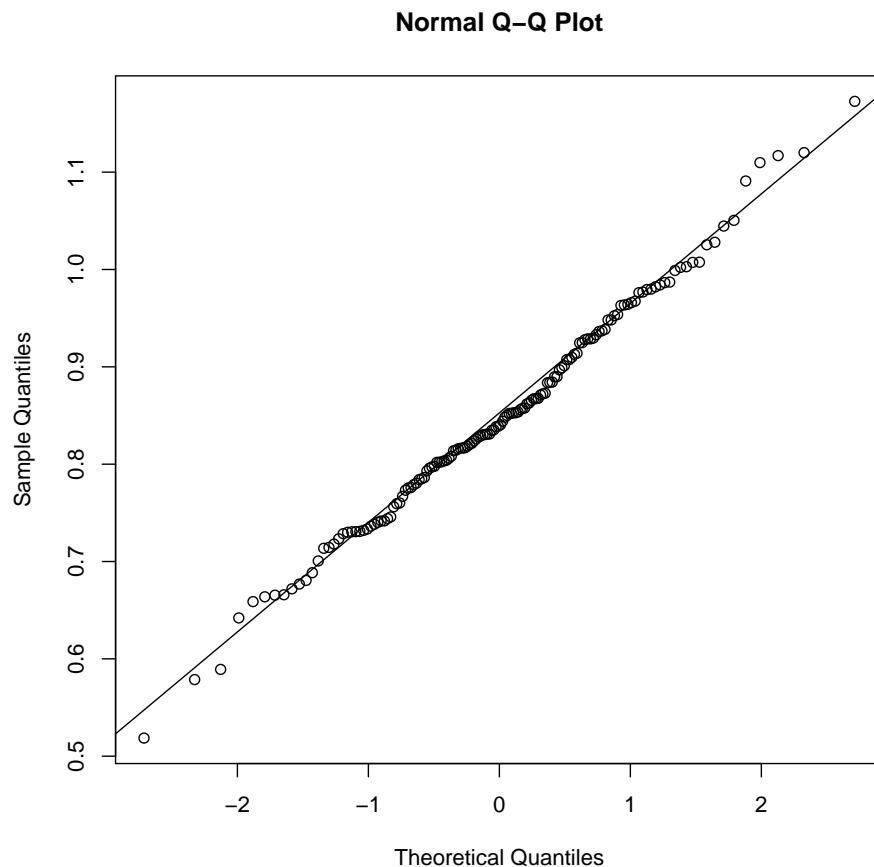
c

Ved å bruke kommandoen `IQR` i R får jeg inter-kvartil og `sd` gir meg standardavvik.

$$\begin{aligned}\text{IQR} &= 0.1517731 \text{mL} \frac{\text{O}_2}{\text{m}} \\ \sigma &= 0.1134001 \text{mL} \frac{\text{O}_2}{\text{m}}\end{aligned}$$

d

Ved å bruke kommandoene `qqnorm()` i R får jeg inter-kvartil og `sd()` gir meg standardavvik.



Figur 2: Figuren viser målte verdier mot standardavviket vi fikk tidligere i oppgaven. Ved normalfordelingen burde en forvente en rett strek. Det er så og si det vi ser. Og en burde forvente at det er flere datapunkter ved $x = 0$.

Plottet indikerer hvordan de målte verdiene har fordelt seg i forhold til standardavviket. Igjen ser man at de fleste datapunktene legger seg rundt gjennomsnittet, men i tillegg får man et inntrykk av hvordan målverdiene avviker fra den teoretiske perfekte normalfordelingen (over eller under den lineære funksjonen).

e

Standardiserte verdien til BMR:

$$z = \frac{x - \mu}{\sigma}$$

For $\text{BMR} = 0.8 \text{ mL O}_2/\text{min}$ gir det:

$$z = \frac{x - \mu}{\sigma} = \frac{0.8 - 0.8485003}{0.1134001} \approx -0.43$$

Den standardiserte verdien til x , sier noe om hvor mange standardavvik x ligger unna forventningsverdien. For 0.8 er det -0.43 standardavvik.

f & g

Her står det ikke at jeg trenger å bruke R, så jeg bruker python! Vi definerer gaussian funksjonen og bruker numpy sin innebygde funksjon integrate til å definere integralet! Dette gir:

$$\begin{aligned} p(x < 0.6) &= 0.013949646902606319 \\ p(x > 1.0) &= 0.09004947601825308 \end{aligned}$$

2

a

Kategoriske variabler er variabler som sier noe om gruppering objektet en måler på.

Kvantitativ variabler er variabler som representerer variabler med en diskret tallverdi.

kvalitativ variabler er variabler som har kontinuerlige verdier.

sex og population er kvantitativ.

cpmg og tpmg kvalitativ.

b

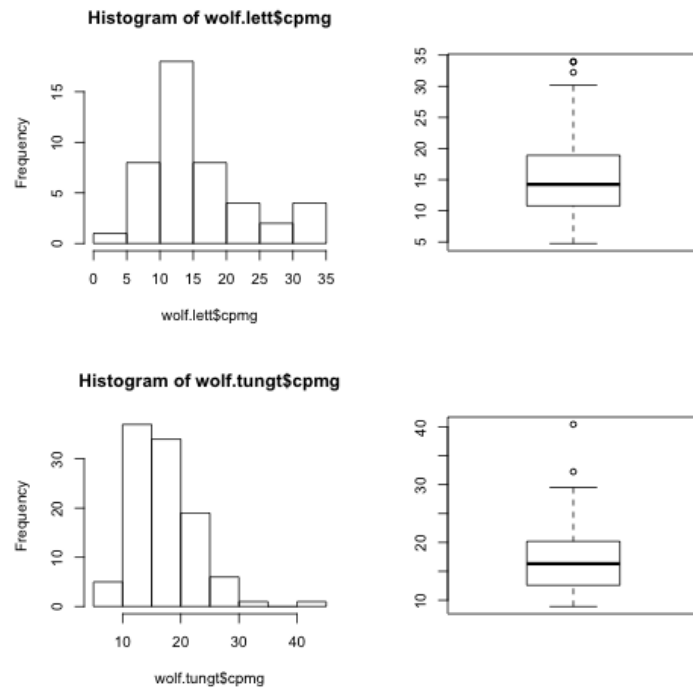


Figur 3: Sirka halvparten er menn. Sirka 2 tredjedeler er fra en tungt jaktet populasjon. Det er jo kult?

c

ok... se kode... Misforsto jeg denne deloppgaven?

d



Figur 4: a) Formen på histogrammet minner om normalfordeling, men med litt for mange verdier høyt oppe. b) Formen minner mindre på en normalfordeling, men vi legger merke til at stolpene er tykkere enn for a) og at det ikke er like mange høye verdier. Forskjellen i verdier er høyere for lett enn tungt.

e

Tabell 1: Vi ser det jeg sa i forrige oppgave. SD er mindre for tungt enn for lett. Gjennomsnittet er lavere for lett, men dette gjør at du for et høyere standardavvik (pga de høye verdiene vi så i figur 4 a))

	Mean	Median	Standard deviation
Lett	15.56222	14.24	7.298785
Tungt	17.07495	16.32	5.543389

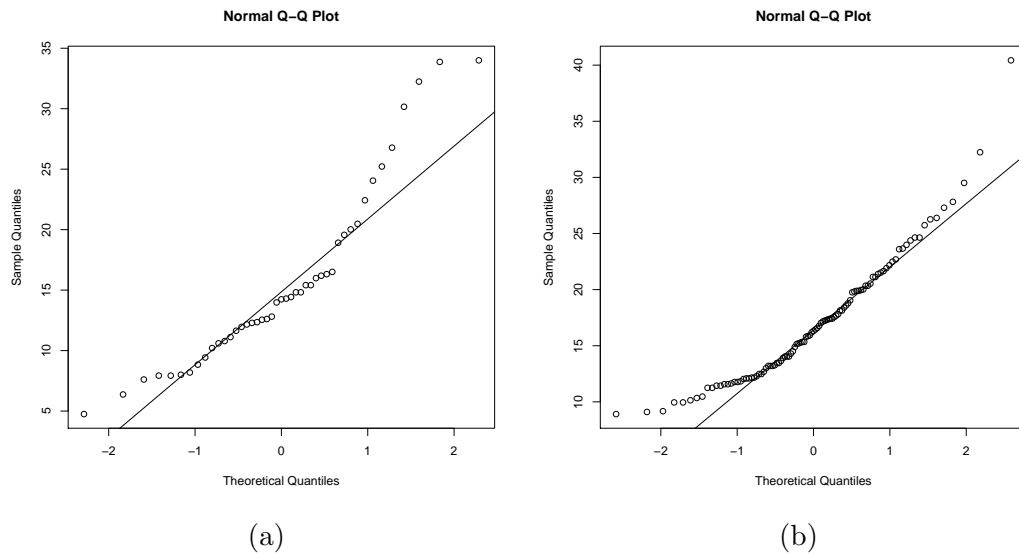
f

FEMTALLS: min, Q_1 , μ , Q_3 , max

Gjennomsnitt og standardavvik: $\mu \wedge \sigma$

Vel, er ikke dette dumt å spørre om før vi har gjort g? Hvis dataen ligner på en normalfordeling burde en bruke gjennomsnitt og standardavvik. Hvis det er en jevn fordeling, så er det bedre med femtallsoppsummering.

88



Figur 5: a) Datapunktene ser jevnt fordelt ut over x-verdien. Datapunktene ligger ikke spesielt bra på linjen. b) Datapunktene ser ikke jevnt fordelt ut over x-verdien. Det er en større andel som ligger mot $x = 0$. Majoriteten til datapunktene ligger bra på linjen.

Fra figur 5 ser det ut til at tungt er normalfordelt. Lett ser ikke ut til å være det. Ergo; femtallsoppsummering for lett og gjennomsnitt og standardavvik for tungt.

3

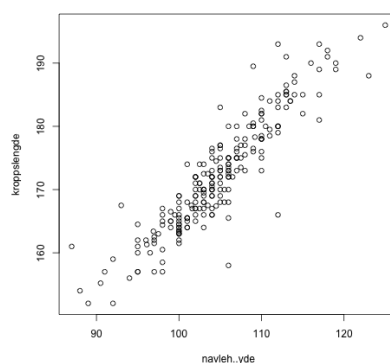
a

150 kvinner og 73 menn deltok i forsøket!

Tabell 2: Tabellen viser en oversikt over femtallsoppsummeringen til de fire målingene i datasettet.

	min	Q_1	μ	Q_3	max
kroppslengde	152.0	166.0	172.3	178.0	196.0
fot.navle	87.0	101.0	104.8	109.0	125.0
navle.isse	52.00	65.00	67.34	70.00	81.00
favn	146.0	165.0	172.4	180.0	202.0

b



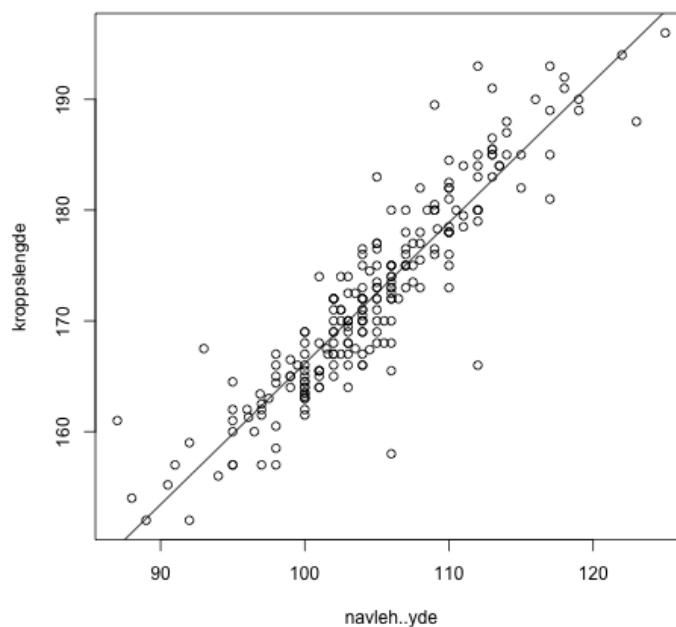
Figur 6: Figuren viser kroppshøyde som funksjon av navlelengde. En skulle nesten tro at det var en høy korrelasjon mellom disse to variablene.

c

Ooooo, ser man det! Det var en høy korrelasjon mellom de. cor funksjonen gir 0.9140397. Det vil si at høyden til et menneske er høyt avhengig av høyden til navlen for det mennesket.

d

Jeg tok meg friheten til å skifte på koden gitt i obligen, slik at programmet mitt kjørte.



Figur 7: Dette ser jo bra ut.

e

Summary funket dårlig. Jeg brukte `print(fit)` for å fastslå koeffisientene til modellen. Hvis navlehigh øker med en cm øker høyden med 1.273 cm.

Tabell 3: Coefficients for the linear model

Coefficients	a	b
	38.897	1.273

f

Kroppshøyden til en person med navlehigh på 121 cm vil med modellen min bli predikert til 192.93 cm.

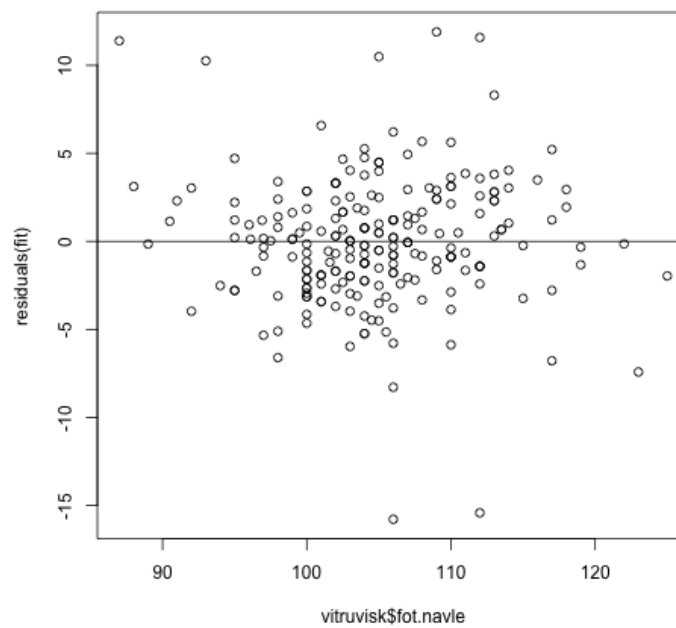
$$f(121) = a + b * x = 192.93$$

g

Tabell 4: ...

	min	Q_1	μ	Q_3	max
kroppslengde	152.0	166.0	172.3	178.0	196.0
fot.navle	87.0	101.0	104.8	109.0	125.0
navle.isse	52.00	65.00	67.34	70.00	81.00
favn	146.0	165.0	172.4	180.0	202.0

h



Figur 8: ...

Appendix

```
#OPPGAVE 1
from matplotlib.pyplot import * #import av shit
from numpy import * #import av shit
import scipy.integrate as integrate #import av shit

infile = open("data.txt","r") #aapne fil
infile.readline() #aapne fil

x = [] #tomme lister
hist_list = [] #tomme lister
for line in infile.readlines(): #leser fil
    words = line.split(" ") #leser fil
    hist_list.append(float(words[1])) #leser fil
    x.append(int(eval(words[0]))) #leser fil

s = std(hist_list) #standard deviation
m = mean(hist_list) #gjennomsnitt

def gaussian(x,s=s,m=m): #gaussian funksjon
    return (1/(sqrt(2*pi)*s))*exp(-0.5*((x-m)/s)**2) #formelen

def integral(fra=0,til=100,func = gaussian): #funksjon
    return integrate.quad(func,fra,til)[0] #numpy sin integrat

if __name__ == '__main__':
    print(integral(0,0.6)) #INTEGRAL
    print(integral(1,100)) #INTEGRAL
    print(integral(0,100)) #INTEGRAL
```

```

#OPPGAVE 1
# data = "http://www.uio.no/studier/emner/matnat/math/STK1000/data/"
# zebrafinch <- read.table(data,header=TRUE)

# Create data for the graph.
zebrafinch <- c(0.7562718,0.7841234,0.8535867,0.82394,0.7804905,0.9

#sum(zebrafinch/150)
mean(zebrafinch)           #AVERAGE = 0.8485003
sd(zebrafinch)             #Standard deviation
IQR(zebrafinch)            #Interquartile
median(zebrafinch)         #MEDIAN

# Standardiserte verdien til 0.8 er lik:
print((0.8-mean(zebrafinch))/sd(zebrafinch))
#-0.4276918 standardavvik til normalfordelingen

# Give the chart file a name.
png(file = "zebrafish.pdf")
# Create the histogram.
hist(zebrafinch,xlab = "Weight",col = "yellow",border = "blue")

# Save the file.
dev.off()

qqnorm(zebrafinch) #lager qq plot
qqline(zebrafinch) #lager qq plot
# pnorm(zebrafinch) #NEI TAKK

```

[illegible]

```

#FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
#EHM tekst...

#GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
qqnorm(wolf.tungt$cpmg)      #QQ plot
qqline(wolf.tungt$cpmg)     #QQ plot
qqnorm(wolf.lett$cpmg)      #QQ plot
qqline(wolf.lett$cpmg)      #QQ plot

```



```
png(file = "3H.pdf")
plot(vitruvisk$fot.navle, residuals(fit))
abline(h=0)
# Save the file.
dev.off()
```