

# UMNW, Mathematik 1, Lösung Serie 3

Thomas Kuster

25. November 2003

## 1

### 1.1

#### 1.1.1

$$f(10) = 15$$

#### 1.1.2

$$f(20) = 25$$

### 1.2

$$f(f(10)) = f(15) = 21$$

### 1.3

für welche  $x$  gilt  $f(x) = x \Rightarrow f(0) = 0$  oder  $f(30) = 30$

### 1.4

$$f(10) - 10 = 15 - 10 = 5$$

### 1.5

$$\max f(x) - x = 21 - 15 = 6 \Rightarrow x = 15$$

## 2

### 2.1

#### 2.1.1

$$q = -0.2, \quad m = 0,9$$

### 2.1.2

$$q = 4, \quad m = -0.8$$

### 2.2

#### 2.2.1

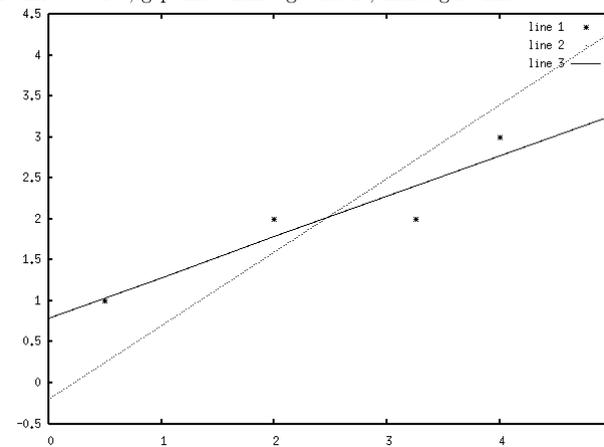
$$m = \frac{\frac{1}{n} \sum_{k=1}^n x_k y_k - \bar{x} \bar{y}}{\frac{1}{n} \sum_{k=1}^n x_k^2 - \bar{x}^2}$$

$$m = 0.4966740576$$

$$q = \bar{y} - m \bar{x}$$

$$q = 0.789356985$$

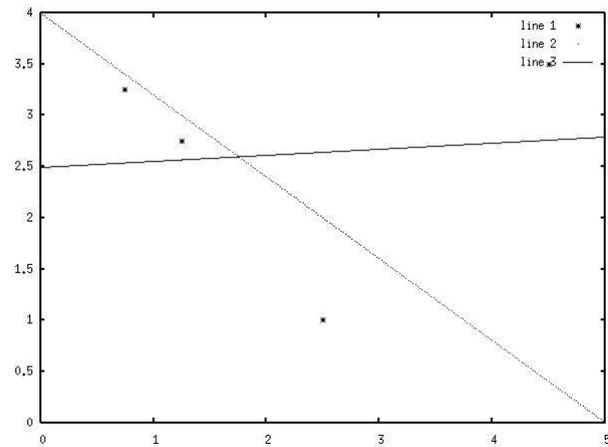
Punkte: Werte, gepunktete Linie: geschätzt, Linie: gerechnet



#### 2.2.2

$$m = 0.05970149254$$

$$q = 2.490671642$$



### 3

#### 3.1

*xm*: Mittelwert der x-Werte (Summe der Einträge durch die Anzahl Einträge) im eigenen Programm Funktion *mittel*

*ym*: Mittelwert der y-Werte

*sigxy*: Korrelationskoeffizient (Ableitung nach x und y)

*sigx2*: Korrelationskoeffizient (2 mal Ableitung nach x)

*sigy2*: Korrelationskoeffizient (2 mal Ableitung nach y)

```
#lineare approximation (kleinste fehlerquadrate)
#
#in: zwei vektoren mit den x- und y-werten
# der gesuchten linearen funktion
#out: vektor mit m: steigung, q: achsenabschnitt und
# c: korrelationskoeffizient
```

```
quadapprox := proc(x,y);
```

```
#linalg-zeug einbinden
with(linalg):
```

```
n := vectdim(x);

#ersatz fuer diese funktion? [x] geht nicht!
mittel := proc(vec);
#da global n nicht funktioniert
n := vectdim(x);
dotprod(vec,[seq(1,i=1..n)])/n;
end;

#sinnvolle eingabe?
if( vectdim(y) <> n)
then print("Fehler: x und y haben andere Dimensionen");
return(0);
fi;

#m berechnen
sigxy := 1/n*dotprod(x,y)-mittel(x)*mittel(y);
sigx2 := 1/n*dotprod(x,x)-mittel(x)^2;
sigy2 := 1/n*dotprod(y,y)-mittel(y)^2;
m := sigxy/sigx2;

#q berechnen
q := mittel(y)-m*mittel(x);

#korrelationskoeffizient
c := sigxy/sqrt(sigx2*sigy2);

#beide werte als vector zurueckgeben
print("[q, m, c]");
return([q, m, c]);
end;
```

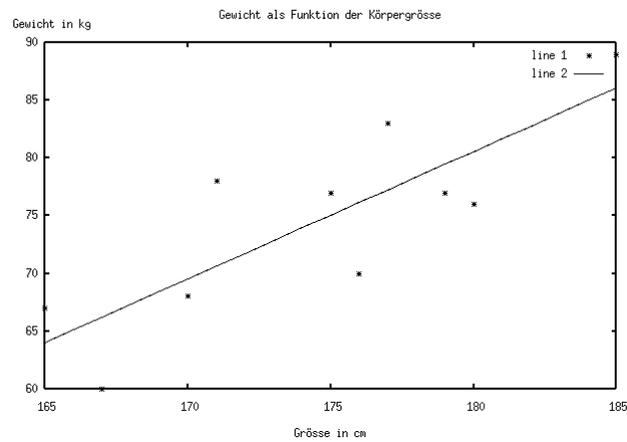
#### 3.2

```
> read quadapprox:
.
.
.
> quadapprox(a1,a2);
      "[q, m, c]"

      [.789356985, .4966740576, .9322980215]

>
```

## 4



```
> evalf(quadapprox(g1,g2));
      "[q, m, c]"
      [-118.0258250, 1.103299857, .8151067735]
```

```
> evalf(quadapprox(g2,g1));
      "[q, m, c]"
      [129.6366484, .6021926390, .8151067735]
```

Gewicht als Funktion der Körpergröße (wie Graph):

$$y = -118.0258250 + 1.103299857x$$

Korrelationskoeffizient:

$$c = 0.8151067735$$

Körpergröße als Funktion des Gewichts:

$$y = 129.6366484 + 0.6021926390x$$

Korrelationskoeffizient:

$$c = 0.8151067735$$