

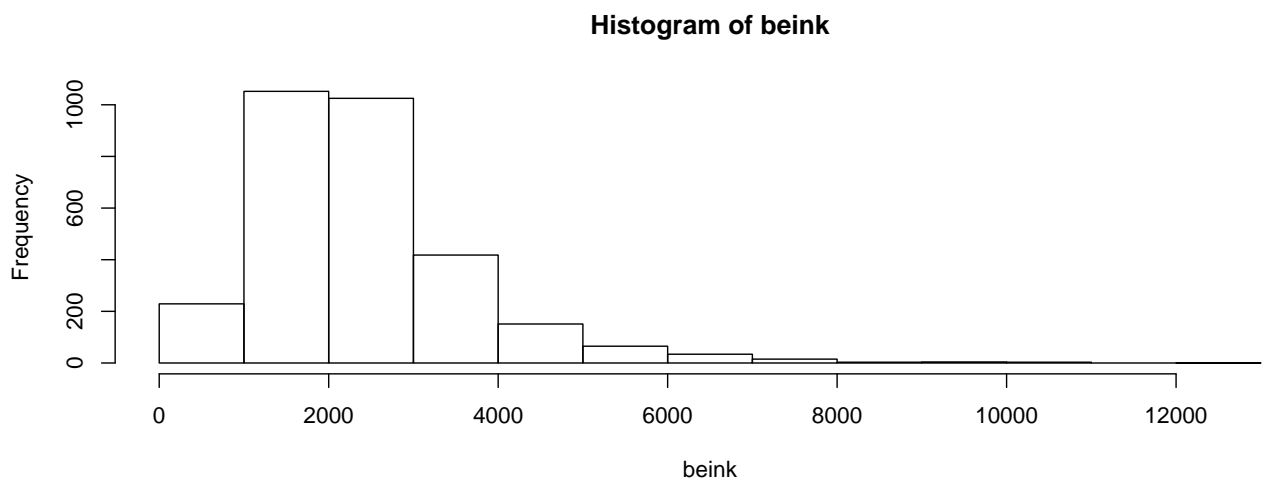
### 3 Generalisierte lineare Modelle

#### Lösung zu Aufgabe 1

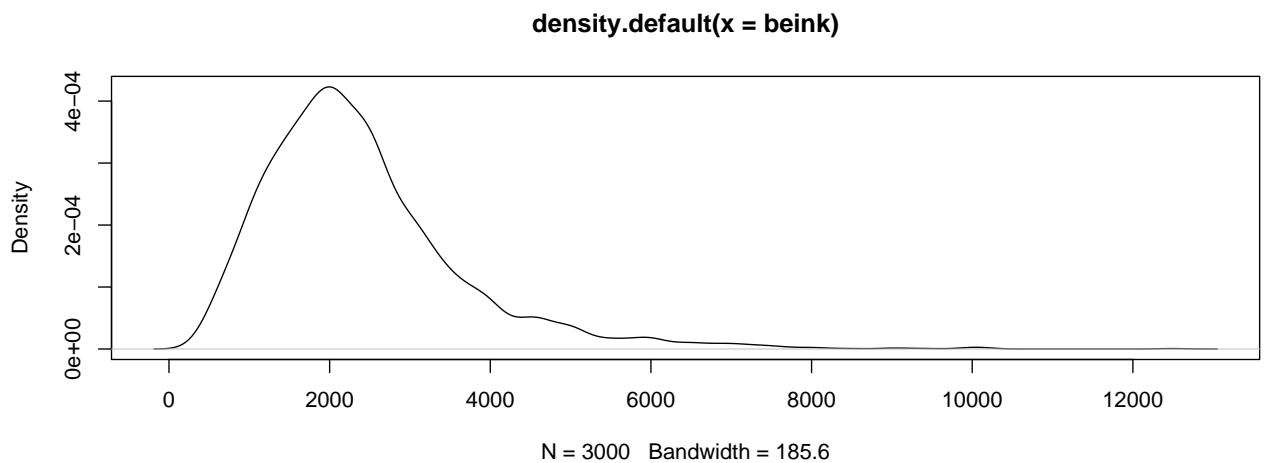
```
# Einlesen der Daten  
fakesoep <- read.table("fakesoep.dat",header=T)  
attach(fakesoep)
```

- a) Man untersucht die Gestalt der Verteilung des Bruttoeinkommens empirisch, z.B. mit einem Histogramm, Kerndichteschätzer oder Boxplot.

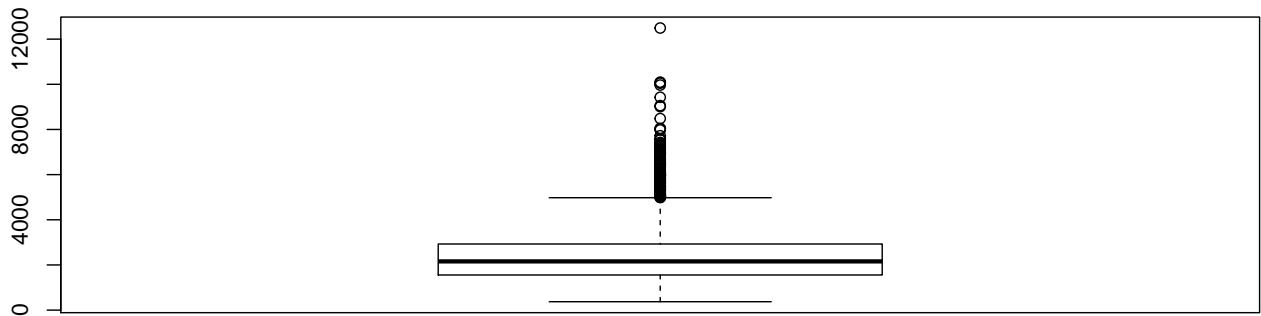
```
hist(beink)
```



```
plot(density(beink))
```



```
boxplot(beink)
```



```
summary(beink) # Median < Mean
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      371   1556   2157   2395   2926   12490
```

→ Die Variable beink nimmt positive Werte auf metrischem Skalenniveau an, die (etwas) linkssteil bzw. rechtsschief verteilt sind. Dafür spricht auch die 5-Zahlen-Zusammenfassung aus

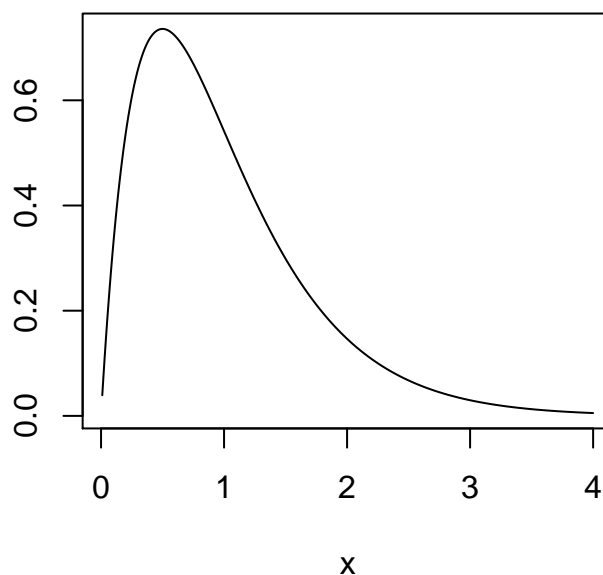
b) siehe Übungsmitschrift

c) siehe Übungsmitschrift

d) `xseq <- seq(0.01,4,by=0.01)`

```
# ein Beispiel
a <- 2
plot(xseq,dgamma(xseq,shape=a,scale=1/a),ylab="",xlab="x",
     main=paste("shape =",a),type="l")
```

**shape = 2**



```
## mit Funktion manipulate
## library(manipulate)
##
## dens_gamma <- function(x,a){
##   plot(xseq,dgamma(xseq,shape=a,scale=1/a),ylab="",xlab="x",
##        main=paste("shape =",a),type="l")
## }
```

```
# manipulate(dens_gamma(x=xseq,a),a=slider(0,5,step=0.1))

# funktioniert nur in R-Studio!
```

e) siehe Übungsmitschrift und

```
# Natürlicher Link
gammaNat <- glm(beink ~ groesse + alter + dauer + verh + geschl + deutsch + abitur,
family=Gamma())
summary(gammaNat)

##
## Call:
## glm(formula = beink ~ groesse + alter + dauer + verh + geschl +
## deutsch + abitur, family = Gamma())
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.46133  -0.31502  -0.05706   0.19162   1.91254
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.064e-03  8.659e-05  12.287 < 2e-16 ***
## groesse     -3.287e-06  4.680e-07  -7.024 2.66e-12 ***
## alter       -2.452e-07  3.966e-07  -0.618  0.536
## dauer       -3.839e-06  3.701e-07 -10.372 < 2e-16 ***
## verh        -7.339e-06  7.111e-06  -1.032  0.302
## geschl      1.419e-04  9.549e-06  14.855 < 2e-16 ***
## deutsch    -1.065e-05  1.210e-05  -0.880  0.379
## abitur     -1.318e-04  6.220e-06 -21.196 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1748559)
##
## Null deviance: 757.46  on 2999  degrees of freedom
## Residual deviance: 507.22  on 2992  degrees of freedom
## AIC: 49212
##
## Number of Fisher Scoring iterations: 5

# Log Link
gammaLog <- glm(beink ~ groesse +alter + dauer + verh + geschl + deutsch + abitur,
family=Gamma(link=log))
summary(gammaLog)

##
## Call:
## glm(formula = beink ~ groesse + alter + dauer + verh + geschl +
## deutsch + abitur, family = Gamma(link = log))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47399  -0.31490  -0.05961   0.18374   1.94176
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.2202325  0.2182771  28.497 < 2e-16 ***
## groesse     0.0082530  0.0011930   6.918 5.58e-12 ***
## alter       0.0001786  0.0009345   0.191  0.848
```

```
## dauer      0.0119425  0.0009816  12.166 < 2e-16 ***
## verh      -0.0178813  0.0173453  -1.031  0.303
## geschl    -0.3179608  0.0216168 -14.709 < 2e-16 ***
## deutsch   0.0050755  0.0279452  0.182  0.856
## abitur    0.3466434  0.0167621  20.680 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1747557)
##
## Null deviance: 757.46  on 2999  degrees of freedom
## Residual deviance: 502.50  on 2992  degrees of freedom
## AIC: 49184
##
## Number of Fisher Scoring iterations: 5
```

→ Da man für beide Modelle die gleiche Verteilungsannahme tätigt, kann man die Anpassungsgüte anhand der Devianz,  $Dev = -2\phi \sum_i (l_i(\hat{\mu}_i) - l_i(y_i))$ , vergleichen. Da das Modell mit log-Link mit 502.50 den kleineren Wert aufweist (nat. Link: 507.22), ist dieses zu bevorzugen. Vergleich über AIC führt (hier selbstverständlich) zum selben Ergebnis.

→ Man beachte, dass hier der Dispersionsparameter gemäß der Formel  $\hat{\phi} = 1/(n-p) \sum_i (y_i - \hat{\mu}_i)^2 / \hat{\mu}_i^2$  mitgeschätzt wird.

→ Ein weiterer Vorteil des log-Links ist, dass die Erwartungswerte aufgrund von  $\mu = \exp(\eta)$  nur positive Werte annehmen können. Das ist beim natürlichen (inversen) Link mit  $\mu = \eta^{-1}$  nicht der Fall, d.h. gegebenenfalls werden Restriktionen an den Parametervektor  $\beta$  erforderlich, um  $\mu > 0$  zu garantieren. Vorsicht bei der Interpretation der Parameter: bei Verwenden des inversen Links gilt  $\mu = \eta^{-1}$ , d.h. negative Werte für Koeffizienten von  $\beta$  indizieren einen steigenden Erwartungswert.

```
cbind(gammaNat$coefficients, gammaLog$coefficients)
```

```
##              [,1]      [,2]
## (Intercept) 1.063881e-03 6.2202325333
## groesse    -3.287389e-06 0.0082529622
## alter      -2.452122e-07 0.0001785662
## dauer      -3.839142e-06 0.0119424909
## verh       -7.338988e-06 -0.0178813399
## geschl     1.418518e-04 -0.3179607922
## deutsch    -1.064851e-05 0.0050755384
## abitur     -1.318256e-04 0.3466434243
```

→ Die Effekte der Kovariablen auf das Bruttoeinkommen weisen in beiden Modellen (in fast allen Fällen) in die gleiche Richtung (positiv: groesse, alter, dauer, deutsch, abitur; negativ: geschl), zu beachten ist, dass der Effekt von verh beim natürlichen Link in die andere Richtung weist als beim log-Link. Der Effekt ist allerdings auch nicht signifikant. Das selbe gilt für alter und deutsche Staatsangehörigkeit. Bzgl. Tests sei aber auf die kommenden Übungen verwiesen.

Für eine tiefergehende inhaltliche Interpretation der Parameter wäre die Einbeziehung von Interaktionstermen (z.B. verh und geschl) wünschenswert. also etwa (Zugabe, keine Aufgabenstellung aber interessant!):

```
gammaLog2 <- update(gammaLog, . ~ . + verh:geschl)
summary(gammaLog2)

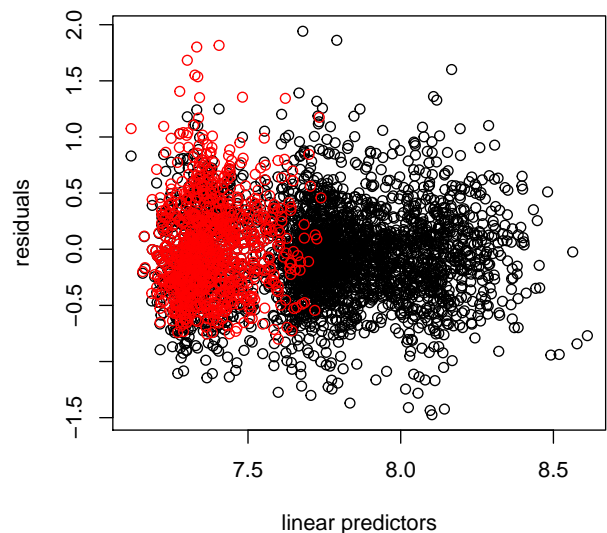
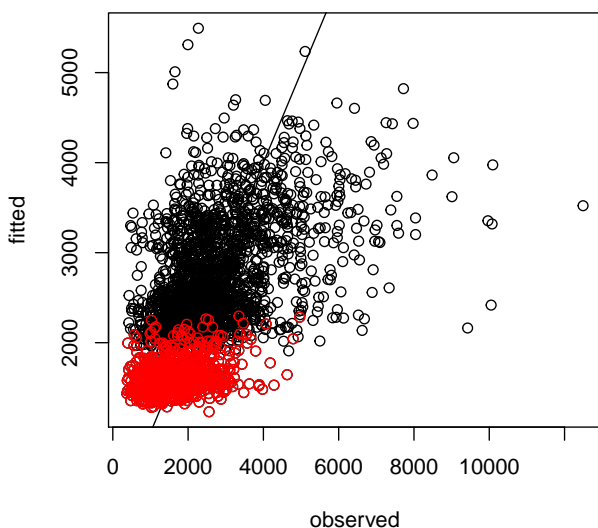
##
## Call:
## glm(formula = beink ~ groesse + alter + dauer + verh + geschl +
##      deutsch + abitur + verh:geschl, family = Gamma(link = log))
```

```
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -1.41195   -0.30952   -0.05603    0.18303    1.86195
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.142e+00  2.150e-01  28.562 < 2e-16 ***
## grosse      8.106e-03  1.174e-03   6.906 6.06e-12 ***
## alter       7.459e-05  9.195e-04   0.081 0.935351
## dauer      1.186e-02  9.658e-04  12.277 < 2e-16 ***
## verh       1.292e-01  2.252e-02   5.737 1.06e-08 ***
## geschl     -1.157e-01  2.986e-02  -3.875 0.000109 ***
## deutsch    1.254e-02  2.751e-02   0.456 0.648615
## abitur     3.415e-01  1.650e-02  20.698 < 2e-16 ***
## verh:geschl -3.142e-01  3.204e-02  -9.805 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1691591)
##
##      Null deviance: 757.46  on 2999  degrees of freedom
## Residual deviance: 486.27  on 2991  degrees of freedom
## AIC: 49084
##
## Number of Fisher Scoring iterations: 5
```

f) `par(mfrow=c(1,2))`

```
# Plot: Beobachtete Werte gegen Geschätzte Werte
plot(beink,gammaLog$fitted,xlab="observed",ylab="fitted")
abline(0,1)
# Zusatz
points(beink[geschl==1 & abitur==0],
       gammaLog$fitted[geschl==1 & abitur==0],col=2)

# Plot: Geschätzte lineare Prädiktoren gegen Residuen
plot(gammaLog$linear.predictors,residuals(gammaLog),xlab="linear predictors",ylab="residuals")
# Zusatz
points(gammaLog$linear.predictors[geschl==1 & abitur==0],
       gammaLog$residuals[geschl==1 & abitur==0],col=2)
```



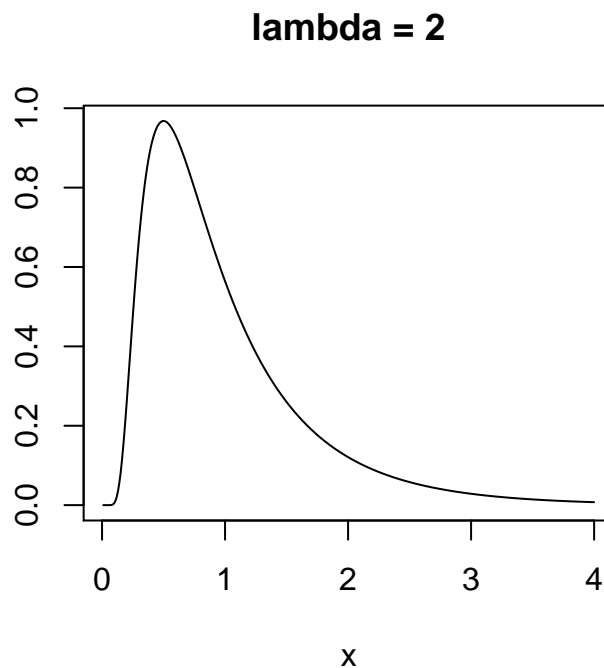
→ In beiden Darstellungen sieht man ein paar wenige Ausreißer. Ansonsten sind keine Auffälligkeiten zu erkennen.

## Lösung zu Aufgabe 2

a) siehe Übungsmitschrift

```
b) library("statmod")
xseq <- seq(0.01,4,by=0.01)

# ein Beispiel
lambda <- 2
plot(xseq,dinvgauss(xseq,mean=1,dispersion=1/lambda),type="l",ylab="",xlab="x",
     main=paste("lambda =",lambda))
```



```
## mit Funktion manipulate
# library(manipulate)
#
# dens_invgauss <- function(x,lambda){
#   plot(xseq,dinvgauss(xseq,mean=1,dispersion=1/lambda),type="l",ylab="",xlab="x",
#        main=paste("lambda =",lambda))
# }
# manipulate(dens_invgauss(x=xseq,lambda),lambda=slider(0,5,step=0.1))

# funktioniert nur in R-Studio!
```

```
c) # Modell mit natürlichem Link:
invgnat <- glm(beink ~ groesse + alter + dauer + verh + geschl + deutsch + abitur,
              family=inverse.gaussian())
## Error: no valid set of coefficients has been found: please supply starting values
```

→ Der Schätzer mit natürlichem Link kann nicht berechnet werden, da R hier keine Startparameter für den Schätzalgorithmus findet, die die durch den natürlichen Link induzierten Restriktionen erfüllen.

```
# Modell mit log-Link:
invLog <- glm(beink ~ groesse + alter + dauer + verh + geschl + deutsch + abitur,
family=inverse.gaussian(link=log))
#Genau wie im Gamma-Modell unterliegt der Parametervektor auch bei log-Link
#keinen Restriktionen
```

```
# Summary
summary(invLog)
```

```
##
## Call:
## glm(formula = beink ~ groesse + alter + dauer + verh + geschl +
## deutsch + abitur, family = inverse.gaussian(link = log))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.040178 -0.007013 -0.001256  0.003663  0.035412
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.3768957  0.2197381  29.020 < 2e-16 ***
## groesse      0.0077052  0.0012067   6.385 1.97e-10 ***
## alter       -0.0006482  0.0009130  -0.710  0.47780
## dauer        0.0136441  0.0010259  13.300 < 2e-16 ***
## verh        -0.0547059  0.0174229  -3.140  0.00171 **
## geschl      -0.3297230  0.0216232 -15.249 < 2e-16 ***
## deutsch     -0.0096808  0.0276394  -0.350  0.72617
## abitur       0.3432797  0.0181570  18.906 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for inverse.gaussian family taken to be 8.03145e-05)
##
## Null deviance: 0.37536 on 2999 degrees of freedom
## Residual deviance: 0.26421 on 2992 degrees of freedom
## AIC: 49412
##
## Number of Fisher Scoring iterations: 6
```

```
# zum Vergleich
summary(gammaLog)
```

```
##
## Call:
## glm(formula = beink ~ groesse + alter + dauer + verh + geschl +
## deutsch + abitur, family = Gamma(link = log))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.47399 -0.31490 -0.05961  0.18374  1.94176
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.2202325  0.2182771  28.497 < 2e-16 ***
## groesse      0.0082530  0.0011930   6.918 5.58e-12 ***
## alter        0.0001786  0.0009345   0.191  0.848
## dauer        0.0119425  0.0009816  12.166 < 2e-16 ***
## verh        -0.0178813  0.0173453  -1.031  0.303
## geschl      -0.3179608  0.0216168 -14.709 < 2e-16 ***
## deutsch      0.0050755  0.0279452   0.182  0.856
## abitur       0.3466434  0.0167621  20.680 < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.1747557)
##
##      Null deviance: 757.46  on 2999  degrees of freedom
## Residual deviance: 502.50  on 2992  degrees of freedom
## AIC: 49184
##
## Number of Fisher Scoring iterations: 5
```

Übergibt man als Startwerte die Parameterschätzer aus dem Gamma-Modell, beginnt R zwar mit der Berechnung, der Algorithmus konvergiert allerdings nicht:

```
invNat <- glm(beink ~ groesse + alter + dauer + verh + geschl + deutsch + abitur,
family=inverse.gaussian(),start=gammaNat$coef)

tail(warnings(),8)

## NULL
```