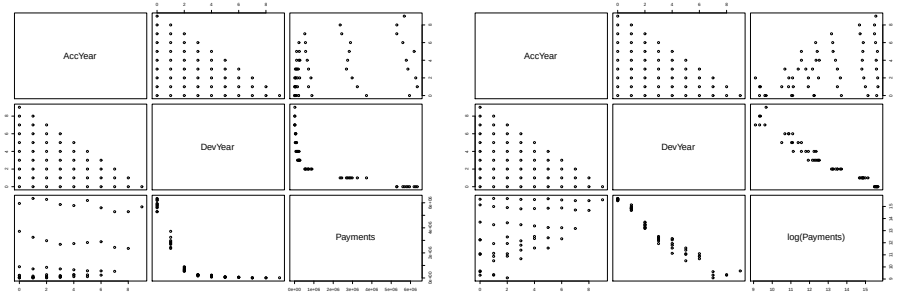
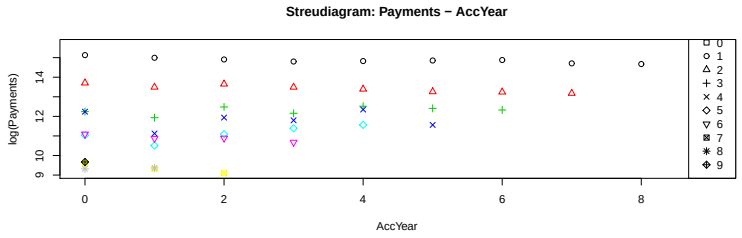
Angewandte Regression — Musterlösungen zur Serie 9 (Aufgabe 2)

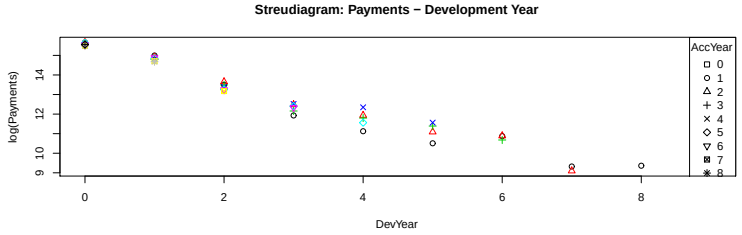
a) Daten einlesen und erste Plots:

```
d.runoff <- read.table("http://stat.ethz.ch/Teaching/Datasets/WBL/RunOff.dat",sep=",",header=TRUE)
plot(d.runoff)
plot(~AccYear+DevYear+log(Payments),data=d.runoff)
```



Log-Transformationen für `Payments` scheint sehr sinnvoll zu sein. Eine Transformation für `DevYear` und `AccYear` sind nicht nötig, was der obige, rechte Plot bestätigt:





b) Für die folgenden Zusammenfassungen der Ergebnisse benützen wir die Abkürzungen F=Faktor, N=Numerisch

Modell 1:

- Ergebnisse der Regressionen:

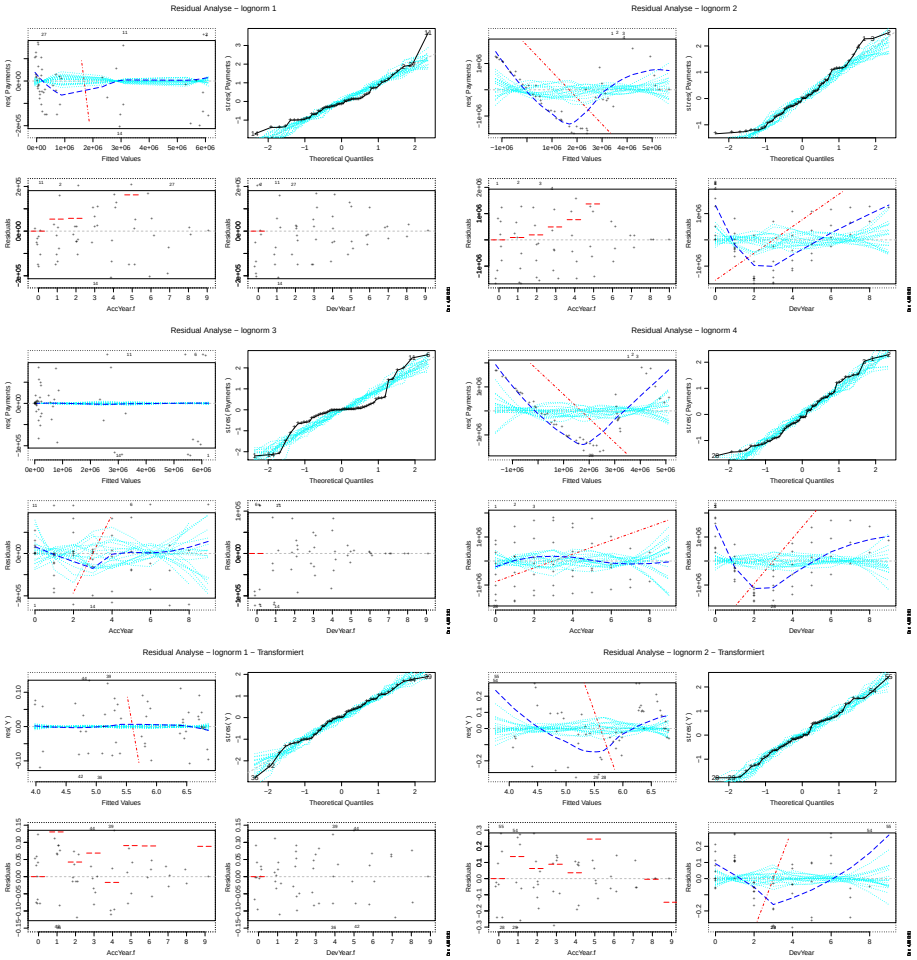
```
d.runoff$AccYear.f <- as.factor(d.runoff$AccYear)
d.runoff$DevYear.f <- as.factor(d.runoff$DevYear)

r.lognorm1 <- regr(Payments~AccYear.f+DevYear.f,data=d.runoff)
r.lognorm2 <- regr(Payments~AccYear.f*DevYear,data=d.runoff)
r.lognorm3 <- regr(Payments~AccYear*DevYear.f,data=d.runoff)
r.lognorm4 <- regr(Payments~AccYear*DevYear,data=d.runoff)

r.lognorm1.l <- regr(log10(Payments)~AccYear.f+DevYear.f,data=d.runoff)
r.lognorm2.l <- regr(log10(Payments)~AccYear.f*DevYear,data=d.runoff)
r.lognorm3.l <- regr(log10(Payments)~AccYear*DevYear.f,data=d.runoff)
r.lognorm4.l <- regr(log10(Payments)~AccYear*DevYear,data=d.runoff)
```

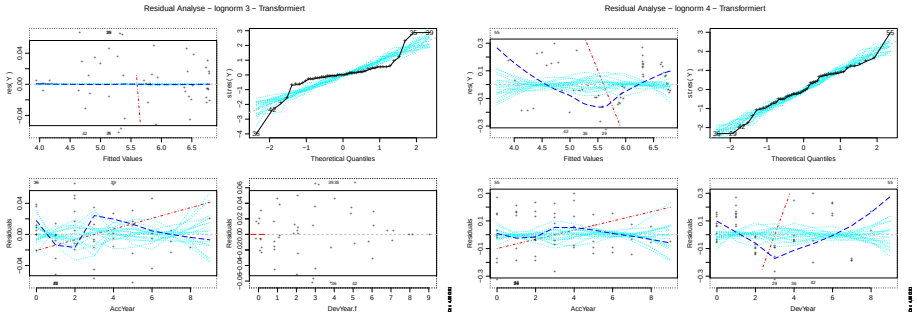
		`AccYear`		`DevYear`		Interaktion
log-Transf	Regression	F / N	Signf	F / N	Signf	
N	Lognormal1	F	***	F	***	nicht moeglich
N	Lognormal2	F		N	**	
N	Lognormal3	N	***	F	***	**
N	Lognormal4	N	***	N	.	***
J	Lognormal1.l	F		F	***	nicht moeglich
J	Lognormal2.l	F		N	***	
J	Lognormal3.l	N		F	***	*
J	Lognormal4.l	N		N	***	**

Die Eingangsvariable `DevYear` erscheint immer als signifikant.

- Ergebnisse der Residuenanalyse:

			`AccYear`	`DevYear`	
log-Transf	Regression	TA	QQ	Part.Res	Part. Resid
N	Lognormal1	nicht so io	nicht so io		
N	Lognormal2	trichterförmig	nicht so io		schlecht
N	Lognormal3	scheint io	nicht io	akzeptabel	
N	Lognormal4	trichterförmig	io	io	schlecht
J	Lognormal1.l	io	io		
J	Lognormal2.l	io	io		schlecht
J	Lognormal3.l	io	schlecht	akzeptabel	
J	Lognormal4.l	schlecht	io	io	schlecht





- Wir benützen `predict()` für die Vorhersagen der Zahlungen der restlichen Developement Years für das Accident Year = 9:

```
predict <- NULL
predict <- cbind(c(1:9),predict.lm(r.lognorm1.l, newdata=data.frame
           (AccYear.f="9",DevYear.f=as.factor(c(1:9))), interval="none"))
predict <- cbind(predict,predict.lm(r.lognorm2.l, newdata=data.frame
           (AccYear.f="9",DevYear=c(1:9)), interval="none"))
predict <- cbind(predict,predict.lm(r.lognorm3.l, newdata=data.frame
           (AccYear=9,DevYear.f=as.factor(c(1:9))), interval="none"))
predict <- cbind(predict,predict.lm(r.lognorm4.l, newdata=data.frame
           (AccYear=9,DevYear=c(1:9)), interval="none"))
predict <- data.frame(predict)
names(predict) <- c("DevYear","(F,F)","(F,N)","(N,F)","(N,N)")
predict
```

Die berechnet Vorhersagen sind in einer Tabelle zusammengefasst:

	untransformiert				log-transformiert			
DevYear	(F,F)	(F,N)	(N,F)	(N,N)	(F,F)	(F,N)	(N,F)	(N,N)
1	2711495	5164502	2219126	3252763	2776560	2755866	2279826	2081542
2	456091	4653435	408987	1380465	647506	1338157	450438	682785
3	-70924	4142369	275992	-491832	203780	649764	285278	223966
4	-171727	3631302	141615	-2364130	127010	315504	141552	73465
5	-260205	3120236	168103	-4236428	60929	153198	260873	24098
6	-300555	2609169	3865	-6108725	49956	74388	21034	7905
7	-380694	2098103	-12051	-7981023	10624	36120	1487	2593
8	-391089	1587036	15774	-9853320	10794	17539	16735	851
9	-413559	1075970	-809029	-11725618	12902	8516	13711	279

Zusammenfassung: Wir haben aus der Residuenanalyse gesehen, dass das additive Modell (nicht transformiete) Modell nicht zu gebrauchen ist. Zum gleichen Ergebnis kommen wir mit dem Prediction - negative Zahlungen machen einfach keinen Sinn. Beim multiplikativen (log transformiertem) Modell kommen die Residuenanalyse und die Vorhersagen zum gleichen Ergebnis: das beste Modell ist das mit den Faktoren.

Modell 2: Gamma-Modell

- Da bei `regr()` die Linkfunktion nicht funktioniert, machen wir die GLM's mit `glm`:

```
r.gamma1.c <- glm(Payments~AccYear.f+DevYear.f,data=d.runoff, family=Gamma)
...
r.gamma1.c.l <- glm(log10(Payments)~AccYear.f+DevYear.f,data=d.runoff,
family=Gamma)
...
r.gamma1.l <- glm(Payments~AccYear.f+DevYear.f,data=d.runoff,
```

```
family=Gamma(link=log))
...
r.gamma1.l.l <- glm(log10(Payments)~AccYear.f+DevYear.f,data=d.runoff,
family=Gamma(link=log))
...
r.gamma1.i <- glm(Payments~AccYear.f+DevYear.f,data=d.runoff, family=Gamma(link=i...
r.gamma1.i.l <- glm(log10(Payments)~AccYear.f+DevYear.f,data=d.runoff,
family=Gamma(link=identity))
...
```

und benützen `drop1()` zum Untersuchen der Signifikanz:

```
drop1(r.gamma1.c,scope=c("AccYear.f","DevYear.f"),test="F")
drop1(r.gamma2.c,scope=c("AccYear.f","DevYear","AccYear.f:DevYear"),test="F")
drop1(r.gamma3.c,scope=c("AccYear","DevYear.f","AccYear:DevYear.f"),test="F")
...
```

Wir tragen die Ergebnisse der GLM's in einer Tabelle zusammen:

			AccYear		DevYear		Interaktion
Linkf	log-Transf	Modell	F / N	Signif	F / N	Signif	
kanon.	N	r.gamma1.c	F		F	***	keine
kanon.	N	r.gamma2.c	F		N	***	
kanon.	N	r.gamma3.c	N		F	***	*
kanon.	N	r.gamma4.c	N		N	***	*
kanon.	J	r.gamma1.c.l	F		F	***	keine
kanon.	J	r.gamma2.c.l	F		N	***	
kanon.	J	r.gamma3.c.l	N		F	***	*
kanon.	J	r.gamma4.c.l	N		N	***	
log	N	r.gamma1.l	F		F	***	keine
log	N	r.gamma2.l	F		N	***	
log	N	r.gamma3.l	N		F	***	*
log	N	r.gamma4.l	N		N	***	**
log	J	r.gamma1.l.l	F		F	***	keine
log	J	r.gamma2.l.l	F		N	***	
log	J	r.gamma3.l.l	N		F	***	
log	J	r.gamma4.l.l	N		N	***	*
ident.	N	r.gamma1.i	F	.	F	***	keine
ident.	N	r.gamma3.i	N		F	***	*
ident.	J	r.gamma1.i.l	F	.	F	***	keine
ident.	J	r.gamma3.i.l	N		F	***	*
ident.	J	r.gamma4.i.l	N	.	N	***	**

Bemerkung: bei der `glm()`-Procedure für das Modell r.gamma4.i erhalten wir eine Fehlermeldung; beim Ausführen von `drop1()` erhalten wir Fehlermeldungen bei den Modellen r.gamma2.i und r.gamma2.i.l - deshalb keine Eintr|"ag in der Tabelle.

Zusammenfassung: man sieht, dass im Gamma-Modell für verschiedene Links nur das Developement Year signifikant ist.

- Für die Residuenanalyse machen wir die TA und die Partial Residual-Plots:

```
par(mfrow=c(4,3))
TA.plot(r.gamma1.c, main="Gamma1 - canonical Link")
termplot(r.gamma1.c,main="Gamma1 - canonical Link", terms=c("AccYear.f","DevYear.f"), partial=T)
TA.plot(r.gamma2.c,main="Gamma2 - canonical Link")
termplot(r.gamma2.c,main="Gamma2 - canonical Link", terms=c("AccYear.f","DevYear"), partial=T)
TA.plot(r.gamma3.c,main="Gamma3 - canonical Link")
termplot(r.gamma3.c,main="Gamma3 - canonical Link", terms=c("AccYear","DevYear.f"),partial=T)
```



```
TA.plot(r.gamma4.c,main="Gamma4 - canonical Link")
termplot(r.gamma4.c,main="Gamma4 - canonical Link", terms=c("AccYear","DevYear"),partial=T)
```

Für die Plots der anderen GLM's analog.

Die Ergebnisse der Residuenanalyse fassen wir in einer Tabelle zusammen:

Modell	TA	AccYear Residuals	DevYear Residuals
r.gamma1.c	io	io	io
r.gamma2.c	nicht io	io	nicht io
r.gamma3.c	io	io	io
r.gamma4.c	nicht io	io	nicht io
r.gamma1.c.l	io	io	io
r.gamma2.c.l	nicht so io	io	io
r.gamma3.c.l	io	io	io
r.gamma4.c.l	io	io	io
r.gamma1.l	io	io	io
r.gamma2.l	nicht so io	io	io
r.gamma3.l	io	io	io
r.gamma4.l	nicht so io	io	io
r.gamma1.l	io	io	io
r.gamma2.l.l	nicht so io	io	io
r.gamma3.l.l	io	io	io
r.gamma4.l.l	nicht so io	io	io
r.gamma1.i	io	io	io
r.gamma2.i	nicht io	io	io
r.gamma3.i	io	io	nicht io
r.gamma1.i.l	io	io	io
r.gamma2.i.l	nicht io	io	io
r.gamma3.i.l	io	io	io
r.gamma4.i.l	nicht io	io	io

Bemerkung: aus der Residuenanalyse sehen wir, dass `DevYear` ein Faktor sein muss.

- Für die Berechnung der Vorhersagen gehen wir gleich wie oben vor, wobei man auf die Link-Funktion und die Log-Transformation achten muss.

```
predict <- NULL
predict <- cbind(... ,1/predict.glm(r.gamma1.c,
           newdata=data.frame(AccYear.f="9",DevYear.f=as.factor(c(1:9)))))
...
predict <- cbind(... ,10^(1/predict.glm(r.gamma1.c.l,
           newdata=data.frame(AccYear.f="9",DevYear.f=as.factor(c(1:9))))))
...
predict <- cbind(... ,exp(predict.glm(r.gamma1.l,
           newdata=data.frame(AccYear.f="9",DevYear.f=as.factor(c(1:9))))))
...
predict <- cbind(... ,10^exp(predict.glm(r.gamma1.l.l,
           newdata=data.frame(AccYear.f="9",DevYear.f=as.factor(c(1:9))))))
...
predict <- cbind(... ,predict.glm(r.gamma1.i,
           newdata=data.frame(AccYear.f="9",DevYear.f=as.factor(c(1:9)))))
...
predict <- cbind(... , 10^predict.glm(r.gamma1.i.l,
           newdata=data.frame(AccYear.f="9",DevYear.f=as.factor(c(1:9)))))
...
```

Ergebnisse der Vorhersagen:

```
 Gamma - kanon. Link  - untransformiert     Gamma - kanon. Link - log-transformiert
 DevYear   (F,F)  (F,N)   (N,F)   (N,N)      (F,F)   (F,N)   (N,F)   (N,N)
       1 2843658 790309 2313799 2575218     2783428 1816931 2285406 1807554
       2  685118 424725  478854 1677343      654747  679490  454951  648607
```



```
       3  221842 290394  290681 1243710      208799  288140  286772  266682
       4  148937 220618  141961  988230      131067  135447  141560  122485
       5   71471 177877 -234373  819824       63643   69355  312169   61630
       6   53809 149009   27564  700457       51681   38155   22595   33461
       7   11650 128203    3730  611432       11115   22307    2134   19370
       8   11385 112495   18512  542485       11205   13737   17014   11841
       9   15806 100216   15807  487512       14187    8847   14972    7585

Gamma - log-Link - untransformiert          Gamma - log-Link - log-transformiert
DevYear   (F,F)   (F,N)   (N,F)   (N,N)      (F,F)   (F,N)   (N,F)   (N,N)
      1 2778441 2792158 2282363 2061303    2778771 2304712 2283262 1966305
      2  649449 1373632  449814  673960     650625  986064  452647  658429
      3  204403  675773  284535  220357     206092  443159  286190  239463
      4  131185  332454  141995   72048     128617  208614  141549   94001
      5   62754  163554  240751   23557      62175  102586  279007   39599
      6   49801   80462   21053    7702      51145   52565   21845   17807
      7   10670   39584    1487    2518      10981   27999    1828    8505
      8   10724   19474   16735     823      11106   15468   16870    4295
      9   12822    9580   13717     269      13588    8844   14479    2284

Gamma - identity-Link - untransformiert     Gamma - identity-Link - log-transformiert
DevYear   (F,F)   (F,N)   (N,F)              (F,F)   (F,N)   (N,F)   (N,N)
      1 2710913 5521115 2245954            2774798 2873422 2281098 2214571
      2  463678 5366662  412505             646445 1454754  450300  722585
      3  -27679 5212209  279146             203024  736512  285615  235770
      4 -108979 5057757  141990             125793  372881  141538   76929
      5 -187278 4903304  146763              60378  188782  253701   25101
      6 -194824 4748851    4407              50402   95576   21039    8190
      7 -236461 4594398  -11162              10782   48388    1493    2672
      8 -237239 4439945   15774              10929   24498   16735     872
      9 -234534 4285492 -812446              12963   12403   13710     285
```

Zusammenfassung: Wieder bemerken wir, dass aus den Residuenanalysen und Vorhersagen, das log-Link Modell resp die log-transformierte Zielvariable sinnvoll sind.

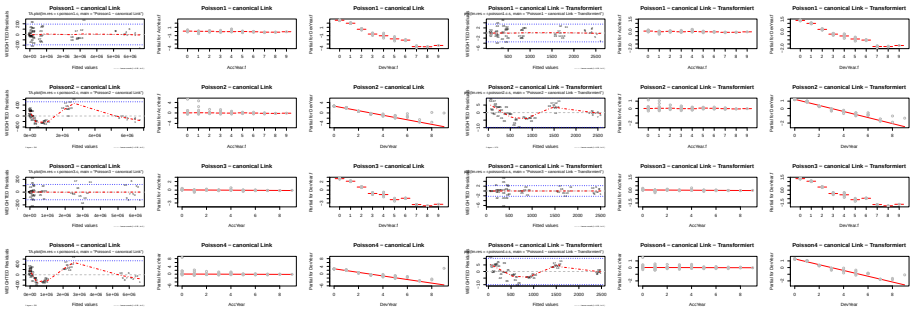
Modell 3: Poisson-Modell

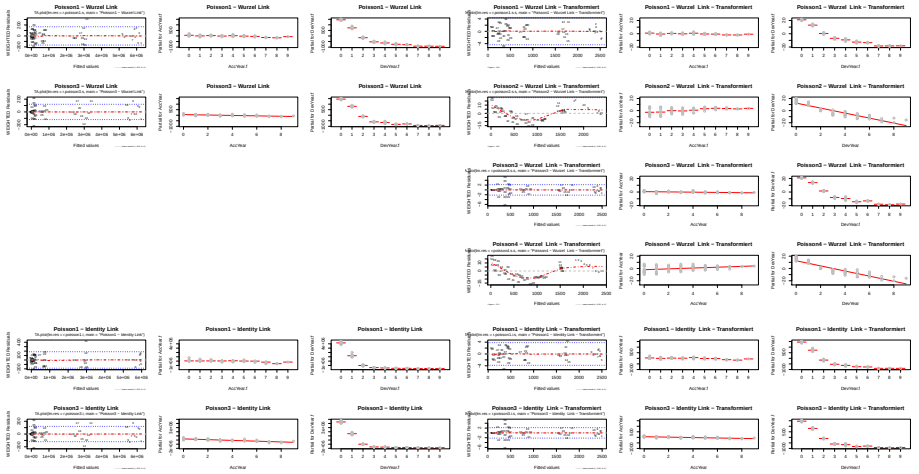
- Als Transformation der Zielvariable nehmen wir die Wurzelfunktion statt den Logarithmus. Wir tragen die Ergebnisse der GLM's in einer Tabelle zusammen:

Linkf	Wurzel-Transf	Modell	AccYear F / N	AccYear Signif	DevYear F / N	DevYear Signif	Interaktion
kanon.	N	r.poisson1.c	F	***	F	***	keine
kanon.	N	r.poisson2.c	F		N	***	***
kanon.	N	r.poisson3.c	N	**	F	***	*
kanon.	N	r.poisson4.c	N	***	N	***	***
kanon.	J	r.poisson1.c.s	F	.	F	***	keine
kanon.	J	r.poisson2.c.s	F		N	***	
kanon.	J	r.poisson3.c.s	N		F	***	*
kanon.	J	r.poisson4.c.s	N	*	N	***	***
sqrt	N	r.poisson1.s	F	**	F	***	keine
sqrt	J	r.poisson1.s.s	F	**	F	***	keine
sqrt	J	r.poisson3.s.s	N		F	***	*
sqrt	J	r.poisson4.s.s	N	***	N	***	***
ident.	N	r.poisson1.i	F	*	F	***	keine
ident.	N	r.poisson3.i	N	***	F	**	***
ident.	J	r.poisson1.i.s	F	*	F	***	keine
ident.	J	r.poisson3.i.s	N		F	***	**

Bemerkung: bei den Modellen r.poisson2.s, r.poisson4.s, r.poisson2.i, r.poisson4.i sind Fehlermeldungen aufgetreten. Bemerkung: bei der `glm()`-Procedure für die Modelle r.poisson2.s, r.poisson4.s, r.poisson2.i, r.poisson4.i erhalten wir Fehlermeldungen; beim Ausführen von `drop1()` erhalten wir Fehlermeldungen bei den Modellen r.poisson3.s und r.poisson2.s.s - deshalb keine Eintr|"ag in der Tabelle.

- Für die Residuenanalyse machen wir die TA und die Partial Residual-Plots:





Die Ergebnisse der Residuenanalyse fassen wir in einer Tabelle zusammen:

Modell	TA	AccYear Residuals	DevYear Residuals
r.poisson1.c	io	io	io
r.poisson2.c	nicht io	io	nicht io
r.poisson3.c	io	io	io
r.poisson4.c	nicht io	io	nicht io
r.poisson1.c.s	io	io	io
r.poisson2.c.s	nicht io	io	nicht so io
r.poisson3.c.s	io	io	io
r.poisson4.c.s	nicht io	io	nicht so io
r.poisson1.l	io	io	io
r.poisson3.s	io	io	io
r.poisson1.s.s	io	io	io
r.poisson2.s.s	nicht so io	io	io
r.poisson3.s.s	io	io	io
r.poisson4.s.s	nicht so io	io	io
r.poisson1.i	io	io	io
r.poisson3.i	io	io	io
r.poisson1.i.s	io	io	io
r.poisson3.i.s	io	io	io

Bemerkung: aus der Residuenanalyse sehen wir, dass für ein gutes Modell die Eingangsvariable`DevYear` ein Faktor sein muss.

- Für die Berechnung der Vorhersagen gehen wir gleich wie oben vor, wobei man auf die Link-Funktion und die Wurzel-Transformation achten muss.

```
Poisson - canonical-Link - untransformiert    Poisson - canonical-Link - Wurzel-transformiert
DevYear    (F,F)    (F,N)    (N,F)    (N,N)        (F,F)    (F,N)    (N,F)    (N,N)
1        1 2795421  2375007  2271443  1986597     2785839  2472365  2275922  1925781
2        2  658706   993849   451053   716004      653120  1077000   450673   647971
3        3  208825   415888   282179   258060      206466   469158   283751   218024
4        4  138592   174033   141793    93009      132906   204373   140740    73359
5        5   66092    72826   271545    33522       63580    89028   267269    24683
6        6   49107    30475    20999    12082       49001    38782    21020     8305
7        7   10363    12753     1458     4355       10312    16894     1473     2794
8        8   10052     5336    16735     1569       10222     7359    16735      940
9        9   13654     2233    13724      566       13191     3206    13717      316
```

```
Poisson - Wurzel-Link - untransformiert    Poisson - Wurzel-Link - Wurzel-transformiert
 DevYear    (F,F)    (N,F)                      (F,F)    (F,N)    (N,F)    (N,N)
       1 2760000  2250000                    2767771  3988000  2267496  2430000
       2  603000   432000                     631826  2708078   442087   800000
       3  163000   280000                     190699  1764178   282331   171000
       4  940000   142000                     118314  1091600   140752    13500
       5   34300   192000                      53718   632570   217003     2.14
       6   21900    15300                      42096   336240    18547     4880
       7     214     4.85                      7701   158687      660   103000
       8     264    16200                      7915    62911    16447   571000
       9      23     2070                      8523    18841     8032  1890000
```

```
Poisson - Identity-Link - untransformiert    Poisson - Identity-Link - Wurzel-transformiert
 DevYear    (F,F)   (N,F)                        (F,F)    (N,F)
       1 2711072 2234477                      2750539  2260000
       2  463843  410565                       601132   433000
       3  -38005  277618                       161807   281000
       4 -123710  141817                        91578   141000
       5 -205441  156795                        33344   187000
       6 -215417    4223                        26116    15400
       7 -257153  -11585                         1397     1.96
       8 -256238   15774                         1631    16200
       9 -260221 -811079                         1362     2110
```

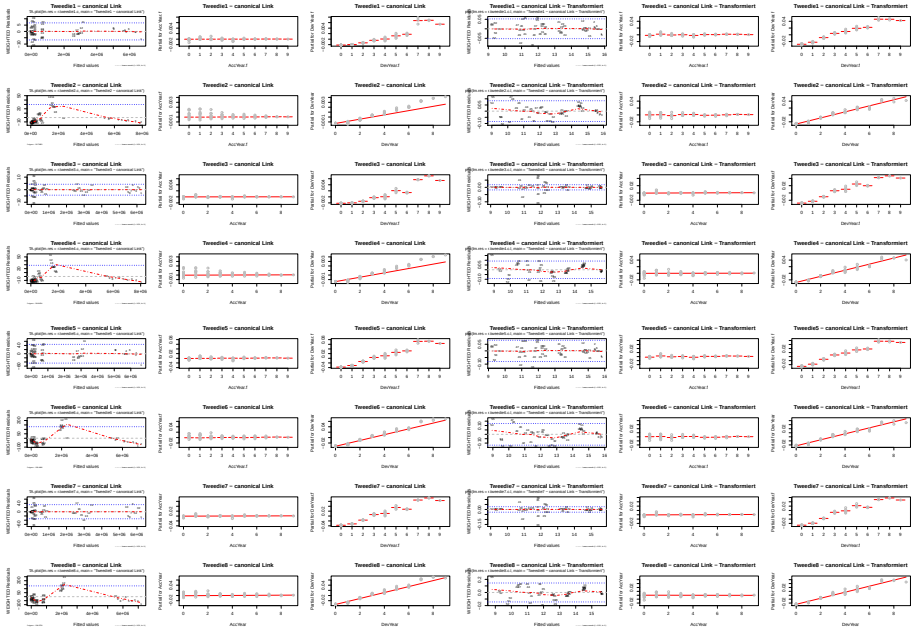
Zusammenfassung: Poisson-Modell mit dem kanonischen Link scheint ein plausibles Modell zu sein. Die restlichen Links hingegen sind nicht vernünftig.

Modell 4: Tweedie-Modell

- Wir tragen die Ergebnisse der GLM's in einer Tabelle zusammen:

			AccYear		DevYear		Interaktion
p	Log-Transf	Modell	F / N	Signif	F / N	Signif	
1.5	N	r.tweedie1.c	F		F	***	keine
1.5	N	r.tweedie2.c	F		N	***	
1.5	N	r.tweedie3.c	N	.	F	***	*
1.5	N	r.tweedie4.c	N		N	***	
1.5	J	r.tweedie1.c.l	F		F	***	keine
1.5	J	r.tweedie2.c.l	F		N	***	
1.5	J	r.tweedie3.c.l	N		F	***	*
1.5	J	r.tweedie4.c.l	N		N	***	
1.2	N	r.tweedie5.c	F		F	***	keine
1.2	N	r.tweedie6.c	F		N	***	
1.2	N	r.tweedie7.c	N	*	F	***	*
1.2	N	r.tweedie8.c	N		N	***	
1.2	J	r.tweedie5.c.l	F		F	***	keine
1.2	J	r.tweedie6.c.l	F		N	***	
1.2	J	r.tweedie7.c.l	N		F	***	*
1.2	J	r.tweedie8.c.l	N		N	***	

- Für die Residuenanalyse machen wir die TA und die Partial Residual-Plots:



Die Ergebnisse der Residuenanalyse fassen wir in einer Tabelle zusammen:

Modell	TA	AccYear Residuals	DevYear Residuals
r.tweedie1.c	io	io	io
r.tweedie2.c	nicht io	io	nicht io
r.tweedie3.c	io	io	io
r.tweedie4.c	nicht io	io	nicht io
r.tweedie1.c.l	io	io	io
r.tweedie2.c.l	akzeptabel	io	io
r.tweedie3.c.l	io	io	io
r.tweedie4.c.l	akzeptabel	io	io
r.tweedie5.c	io	io	io
r.tweedie7.c	io	io	io
r.tweedie5.c.l	io	io	io
r.tweedie6.c.l	akzeptabel	io	io
r.tweedie7.c.l	io	io	io
r.tweedie8.c.l	akzeptabel	io	io

- Für die Berechnung der Vorhersagen gehen wir gleich wie oben vor, wobei man auf die Link-Funktion und die Wurzel-Transformation achten muss.

```
Tweedie - p=1.5 - untransformiert            Tweedie - p=1.5 - log-transformiert
 DevYear   (F,F)   (F,N)   (N,F)   (N,N)      (F,F)   (F,N)   (N,F)   (N,N)
       1 2823213 1367200 2293638 1915130    2781678 2036827 2284024 1862869
       2  678450  600265  466151  962049     653063  806274  453847  647185
       3  219609  335602  286056  577010     207703  347801  286391  251491
       4  147401  214005  141878  384168     130200  161816  141558  107653
       5   70889  148255  620475  274024      63084   80489  296740   50119
       6   53351  108734   24787  205257      51348   42480   22225   25106
       7   11586   83141    2772  159467      11031   23634    1983   13410
       8   11319   65626   17462  127451      11138   13783   16941    7579
       9   15679   53114   15697  104191      13921    8384   14753    4503

Tweedie - p=1.2 - untransformiert            Tweedie - p=1.2 - log-transformiert
 DevYear   (F,F)   (F,N)   (N,F)   (N,N)      (F,F)   (F,N)   (N,F)   (N,N)
       1 2807464 1931847 2280590 1904255    2780645 2174951 2283191 1900850
       2  669144  796426  457432  792252     652017  894011  453177  649546
       3  215152  375284  283653  375785     206999  391853  286164  243906
       4  143924  195156  141828  196392     129637  182166  141557   99690
       5   69184  109466  335350  110590      62713   89387  288447   43978
       6   51890   65198   22675   66073      51121   46093   21997   20783
       7   11249   40769    2037   41423      10970   24878    1889   10452
       8   10969   26543   16998   27026      11089   14004   16898    5560
       9   15083   17877   15143   18236      13745    8194   14596    3113
```

Schlussbemerkung:

- Der Log-Link hat zwei Vorteile:
  - ergibt eine multiplikative Struktur für den Erwartungswert - ist leichter zu verstehen
  - bringt immer positive Werte für den Erwartungswert
- Tweedie: man könnte zusätzlich den optimalen Parameter p bestimmen.
- Einige der obigen Modelle sind für das finale Modell nicht geeignet. Die Besten scheinen aber diejenigen mit nur Faktoren zu sein. Welche man jetzt nun als das Beste Modell wählt, hängt nicht nur von den Residuen, Vorhersagen ab, sondern auch von der Wahl der Modellstruktur: möchte man Faktoren oder numerische Werte? Zb, wie steht es mit dem Developement Year 10, 11, resp 12 aus - für diese Fragestellung brauchen wir numerische Werte, da die Jahre 10 bis 12 nicht als Faktoren existieren.