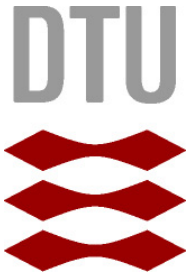
Technical University of Denmark

02424 - Advanced Data Analysis and Statistical Modeling

Project 2

Anders Olsen (s154043)
Patrick Klarskov Jensen (s136392)
Michael Aagaard-Hansen (s961654)



April 3rd 2020

1 Introduction

This report summarizes the second assignment of the course 02424 Advanced Data Analysis and Statistical Modeling. The assignment is divided into two parts. In the first part, we attempt to model ozone concentrations in Los Angeles using a combination of 8 different variables. The second part entails investigating the amount of times office workers change clothing level during a work day as a function of 7 potential variables.

2 Ozone model

In this part, model building using generalized linear models (GLM) is investigated using the *ozone* data set originally discussed in Breiman and Friedman, 1985. The data is comprised of 330 observations, each from a separate day of the year. Along with the observations are 8 different measures that may be of importance in describing the ozone level in the city. These are temperature ($^\circ F$), inversion base height (ft), Daggett pressure gradient (mmHg), visibility (miles), Vandenburg 500 millibar height (m), humidity (pct), inversion base temperature ($^\circ F$), and wind speed (mph).

2.1 Presentation of data

The histogram of realized observations $\mathbf{y}$ is shown in Figure 1 along with the Poisson, Gamma, inverse Gaussian, and log-normal distributions fit to the data. All four distributions were fit using the likelihood approach. Poisson, inverse Gaussian, and log-normal have closed-form solutions that maximizes the likelihood. For the Gamma distribution, a grid search was used to find the α and β values that optimize the log-likelihood. It is readily seen that the Poisson model will not fit the observations well, whereas the Gamma, inverse Gaussian and log-normal distributions might provide good fits to the data. From looking at the histogram, the observations do not seem to stem from a Gaussian distribution, but perhaps a log-Gaussian one. If the fit is poor to the log-Gaussian, generalized linear models will be needed to explain the data properly, potential families being the Gamma and inverse Gaussian.

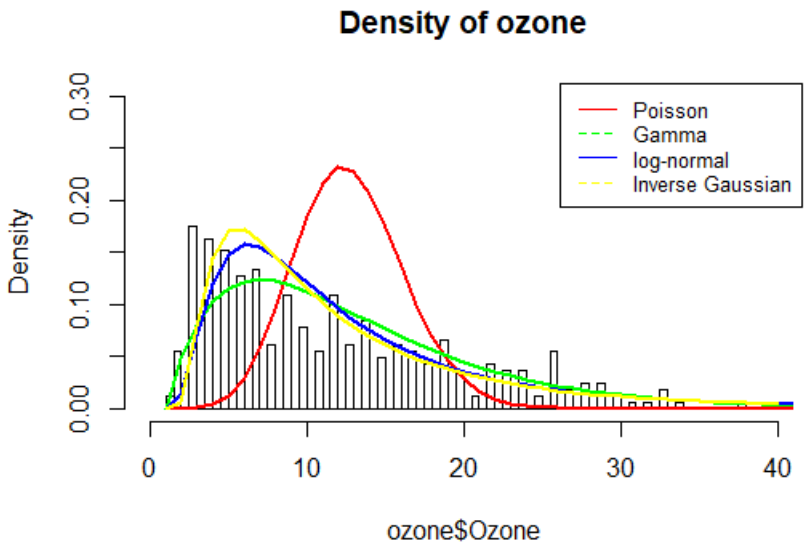


Figure 1: The density of observations $\mathbf{y}$ along with potential distributions of Poisson, Gamma, Inverse Gaussion, and log-normal.

Figure 2 shows the observations $\mathbf{y}$ against each of the 8 predicting variables. Some strong correlations are seen, e.g. ozone level and temperature with a correlation coefficient of $\rho = 0.78$, ozone level and Vandenburg height with a correlation coefficient of $\rho = 0.6$ and ozone level and inverse temperature ($\rho = 0.75$). Some of the variables also have high intercorrelation, e.g. temperature and inverse temperature ($\rho = 0.87$) and Vandenburg height and temperature ($\rho = 0.81$). These high correlations will need to be taken into account when building models to explain the data.

Figure 2 also gives reason to investigate some of the predictors as polynomials. For example, temperature, inverse temperature, and Vandenburg height could be investigated to the 2nd degree. Pressure and wind speed may even be investigated to the 3rd degree.

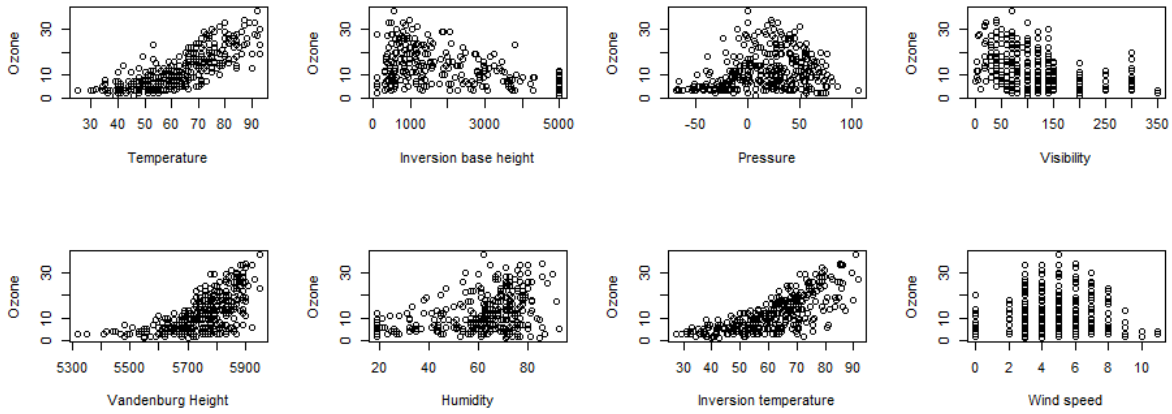


Figure 2: The observations shown against all 8 variables

2.2 General linear model

To provide a measure of comparison when building generalized linear models, and to provide evidence that the normal distribution probably is a poor assumption for the data, we first fit a general linear model. The starting point is all variables to the first degree, but with no interactions or higher-order terms. Doing backward type III selection, we end up with a model that includes temperature, inverse height, and humidity in the following relation:

$$Ozone = -10.5 + 0.33 \cdot Temp - 0.0010 \cdot InvHt + 0.077Hum$$

This relation shows that Temperature and humidity have a high positive impact on ozone level, which is also needed to counteract the large negative intercept, as the response variable is purely nonnegative. Figure 3(left) shows the fitted values versus residuals of the model, and it is clearly seen that the variance of the residuals increases with the mean value. The residual plot is funnel-shaped, perhaps even helicon-shaped. This indicates that assumptions of the general linear model are violated. These assumptions include that the error variances are normally distributed, and constant and independent of the mean (homoscedasticity). Figure 3(middle) also indicates that as the (mean) value of the true observations goes up, so does the variance of the fitted values. This plot also hints that some second-order terms may be missing from the model. Residuals plotted against the variables of the model also indicates that all three predictors could be included as a 2nd order polynomial (not shown). The QQ-plot shows that the residuals are normally distributed to a high degree. Thus, we conclude that the assumption of homoscedasticity is violated.

Model comparison will in this part of the report be carried out mostly using the root-mean-square error (RMSE) in the untransformed domain and diagnostic plots. The RMSE is given in Equation 1.

The AIC will mostly not be used since transformations of the data set also change the appearance of the residuals. For the general linear model, we get $RMSE = 81.69$.

$$RMSE = \sqrt{\sum_{i=1}^{n}(y_i - \mu_i)^2} \tag{1}$$

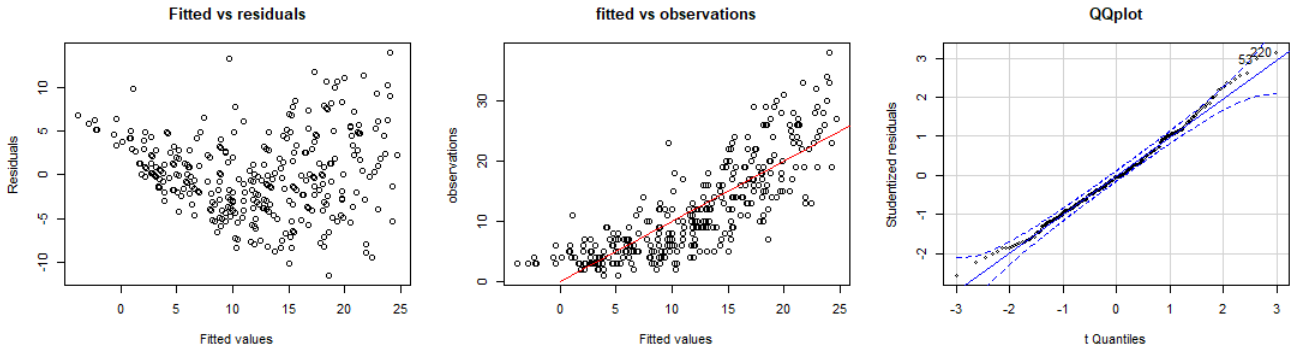


Figure 3: Model residuals from a general linear model. Left: Fitted vs. residuals show error variances increasing with the mean. Middle: Fitted vs observations also show error variance increasing with the mean. Right: QQ-plot showing that error variances are normally distributed to a high degree.

2.3 General linear model with transformation

The residual analysis in the previous section suggests a transformation of the data is necessary. The most straightforward and simple transformation when dealing with error variances that increase with the mean is the log-transformation. Furthermore, as seen in Figure 1, the log-normal distribution may be a valid assumption for the observations. However, this transformation works best when the fitted-vs-residuals plot is funnel-shaped. In our case, it looks more complicated, almost resembling a helicon. Regardless, we fit a new model in the same way as in Section 2.2 and end up with significant parameter values for exactly the same variables: temperature, inverse height, and humidity as in the following relation:

$$log(Ozone) = 0.31 + 0.030 \cdot Temp - 1.2 \cdot 10^{-4} \cdot InvHt + 6.4 \cdot 10^{-3} \cdot Hum$$

The relation is different to the model without response transformation in that the intercept is now positive and that coefficients now play a smaller role. The residual plot can be seen in Figure 4. In the fitted vs. residuals plot, we clearly see that error variances do not seem to be randomly distributed. Instead, we get an inverse-funnel shaped residual plot, indicating that the log-transformation overcompensates the lack of homoscedasticity. The fitted values shown are in the untransformed domain. Also, as seen in the middle plot, there still seems to be some dependence between error variances and the mean. The QQplot is shown with residuals in the log-domain, meaning that the tails are bent downwards. We conclude that a simple log-transformation of the observations does not suffice in fulfilling the assumptions of normality of the error variance and homoscedasticity.

The RMSE for the model using a log-transformed response variable is $RMSE = 76.93$, i.e. a better fit to the data.

2.4 Generalized linear models

In this section, we turn to build generalized linear models in the hopes of achieving a model that better explains the data. In generalized linear models, we build models describing a transformation of the mean value function, as opposed to the actual observations as seen in the previous section. As seen in

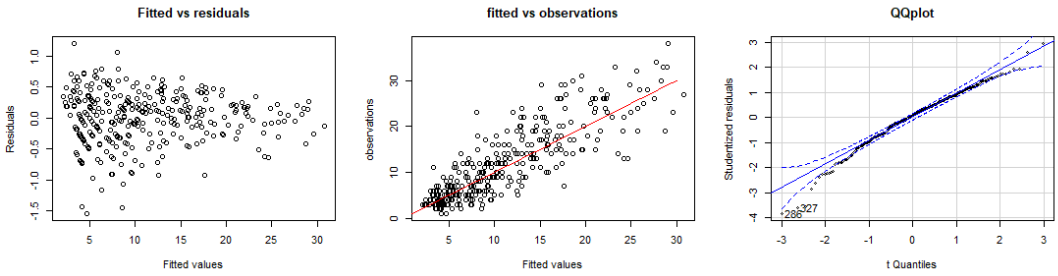


Figure 4: Model residuals for a general linear model using a transformed response variable. Left: fitted vs residuals, middle: fitted vs observations, right: QQplot.

Figure 1, possible distributions for the observations are the inverse Gaussian and the generalized Gamma distribution. On the other hand, the Poisson distribution will probably not fit the data well, and the Binomial distribution does not make sense in this scenario. In *R*, we fit GLMs using these two distributions and their corresponding link functions. Both families have three link functions each: identity, log, and reciprocal, all of which were tried out using a model formula that includes all variables, all second and third-order degree terms and all interactions. The scale of the model is used to compare the families and link functions as well as possible. To compare families and link functions directly, unreduced models are compared using the RMSE and AIC scores, which here is accepted since the models are built using the same data. Both families include a dispersion parameter, meaning that we cannot do a test for goodness of fit. The results are included in Table 1 according to family and link function $g(\mu)$. We see that the Gamma distribution with the canonical link function (the reciprocal) provides a lower RMSE than all of the other models, but not the lowest AIC. Since RMSE is the main method of model comparison here, we opt to move forward with the Gamma distribution with canonical link function.

If the models had been tested using only first-order variables, as in Sections 2.2 and 2.3, the Gamma family with the *log* link function would have had the lowest RMSE. Since the model that should be used when comparing GLM families and link functions should be complex, we chose to present the results from using the largest possible model.

Table 1: Results from the GLM model comparison between the families inverse Gaussian (IG) and Gamma, as well as link functions.

Family	IG	IG	IG	Gamma	Gamma	Gamma
Link	$log(\mu)$	μ	$1/\mu$	$log(\mu)$	μ	$1/\mu$
RMSE	68.2	52.8	61.5	40.1	41.7	35.7
AIC	1457	1346	1743	1614	1553	1728

2.5 Model comparison

As well as performing better than its GLM peers in Section 2.4, the RMSE of the Gamma distribution is also lower than that of the general linear model for both untransformed and transformed response variable. However, this is before any model reduction has taken place. Thus, we proceed with building the Gamma model in the same way as in Sections 2.2 and 2.3. By doing backward type III selection, we end up with a larger model than in the previous sections. Here, variables included are again the same as for the general linear models: temperature, inverse height and humidity in the following relation:

$$Ozone = 0.24 + -1.8 \cdot 10^{-3} \cdot Temp + 1.3 \cdot 10^{-5} \cdot InvHt - 8.5 \cdot 10^{-4} \cdot Hum$$

The resulting RMSE is somewhat higher than before: 88.99. Figure 5 shows residuals in the untransformed domain. The plots still show that the variance of residuals has some dependency on the mean

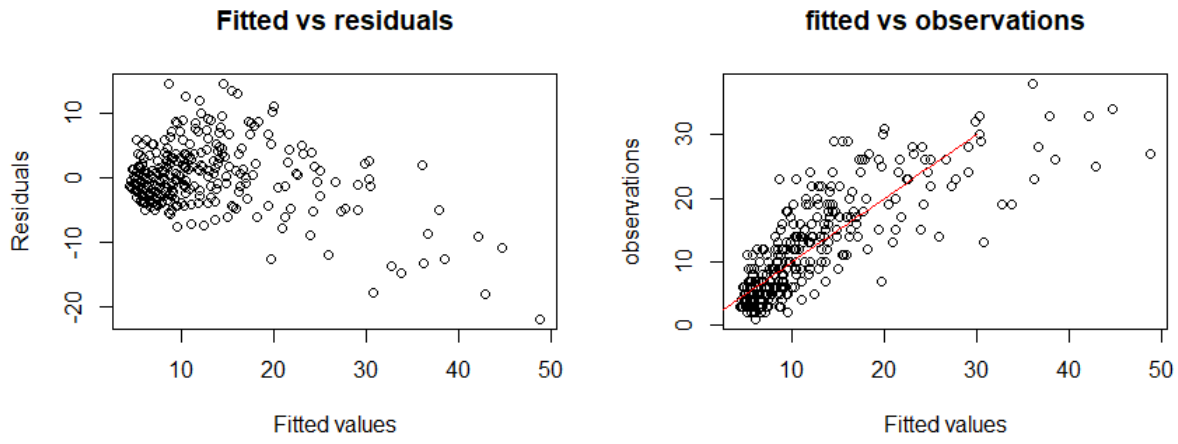


Figure 5: Residuals of the model in the untransformed domain.

value. Especially for high fitted values, residuals are very large, which may also explain the higher RMSE. To remedy this problem, more variables will need to be included in the model.

2.6 Weight matrix

In this section, we estimate the weight matrix $\mathbf{W}$ of the Gamma distribution with the canonical link function. As per theorem 4.2 in Madsen (2011) the weight matrix is given as

$$\mathbf{W}(\boldsymbol{\beta}) = diag\Big[\frac{w_i}{[g'(\mu_i)]^2 V(\mu_i)}\Big] \tag{2}$$

With the reciprocal link function $g(\mu_i) = 1/\mu_i$ we get $g'(\mu_i) = -1/\mu_i^2$. Further, the Gamma distribution has variance function $V(\mu_i) = \mu_i^2$. This amounts to

$$\mathbf{W}(\boldsymbol{\beta}) = diag\Big[\frac{w_i}{1/\mu_i^2}\Big] = diag\big[w_i\mu_i^2\big] \tag{3}$$

This result is also given by Remark 4.18. The weight matrix can be used to construct the parameter covariance matrix using the fact that, asymptotically, $\frac{\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}}{\sqrt{\sigma^2}} \in N_k(\mathbf{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{\Sigma} = [\mathbf{X}^T W(\boldsymbol{\beta})\mathbf{X}]^{-1}$. Using the inverse of the estimated dispersion of the model as the precision, i.e. $w_i = 1/0.1272$ for all i, we get almost exactly the same as the scaled covariance matrix, as seen in Table 2. The differences amount only to machine precision from matrix inversion, as very small values are truncated to zero.

2.7 The Final Model

In this final section of part A, backward selection is performed on a large GLM including all 8 variables to the first, second and third degree, and second-order interactions. The significant variables and interactions can be seen in Figure 6 along with the 95% confidence interval for their parameters. The values shown are in the predictor domain. The intercept, which is 0.93, is not shown in the plot. The final RMSE of the model is 69.39, i.e. a small decrease from previous models.

As is evident in Figure 6, the predictors temperature, pressure, humidity, and inverse temperature all contribute negatively to the prediction. The temperature may be explained by the fact that the colder the weather, the more energy is used for heating buildings, and the more people use polluting transportation measures instead of walking or using a bicycle. Further, with high temperature follows large amounts of sunlight with rays that break apart molecular compounds thus releasing ozone destroying components, in turn decreasing the ozone level. Humidity, which has the highest impact on prediction, may possibly

Cov	intercept	Temp	InvHt	Hum
Intercept	$1.91{\cdot}10^{-4}$	$-1.61{\cdot}10^{-6}$	$-1.36{\cdot}10^{-8}$	$-6.78{\cdot}10^{-7}$
Temp	$-1.61{\cdot}10^{-6}$	$2.32\cdot10^{-8}$	$1.29{\cdot}10^{-10}$	$-5.03{\cdot}10^{-9}$
InvHt	$-1.36{\cdot}10^{-8}$	$1.29{\cdot}10^{-10}$	$2.78{\cdot}10^{-12}$	$1.61{\cdot}10^{-12}$
Hum	$-6.78{\cdot}10^{-7}$	$-5.03{\cdot}10^{-9}$	$1.61{\cdot}10^{-12}$	$1.54{\cdot}10^{-8}$
$\hat{\Sigma}$	intercept	Temp	InvHt	Hum
Intercept	$1.91{\cdot}10^{-4}$	$-1.61{\cdot}10^{-6}$	$-1.00{\cdot}10^{-8}$	$-6.78{\cdot}10^{-7}$
Temp	$-1.61{\cdot}10^{-6}$	$2.00\cdot10^{-8}$	0	$-1.00{\cdot}10^{-8}$
InvHt	$-1.00{\cdot}10^{-8}$	0	0	0
Hum	$-6.80{\cdot}10^{-7}$	$-1{\cdot}10^{-8}$	0	$2.00{\cdot}10^{-8}$

Table 2: Top: The unscaled covariance matrix estimated from the estimated parameter coefficients. Bottom: The ***Sigma*** as calculated above.

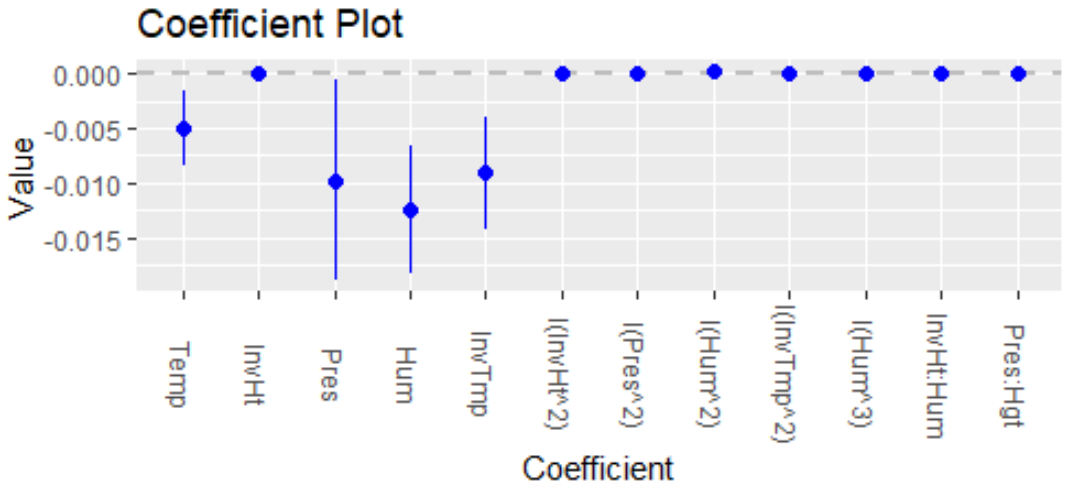


Figure 6: Coefficient plot for the final model.

be explained by ozone being water-soluble to a moderate degree, and therefore not measured as free particles at high levels of humidity, thereby decreasing ozone levels. Also, ozone production in nature decreases when there is less sunlight, and with high humidity often comes overcast weather, and since the consumption is unaffected we detect lower levels of ozone.

Only humidity is in the model at both first, second and third degrees, which can be explained mostly by its importance in the prediction of ozone levels. All interactions and higher-degree dependencies have very low parameter coefficients, yet still significant. It is therefore difficult to determine the actual importance of these terms in the model. If a fit using only temperature, pressure, humidity, and inverse temperature was carried out, RMSE would increase to 76.5, which is only marginally smaller than the general linear model with log-transformation.

Figure 7 shows model diagnostics in the linear domain, i.e. the transformed domain. The residuals versus predicted plot shows a very fine random distribution of residuals. The QQplot shows a good fit in the transformed domain, and the scale-location and leverage plots show that a few observations may be deemed as outliers, but nothing worth investigating.

Finally, Figure 8 shows predictions in the untransformed domain versus the residuals $(y_i - \mu_i)$ and versus actual observations. We see that the variance of residuals still has some dependency on the mean that remains unresolved.

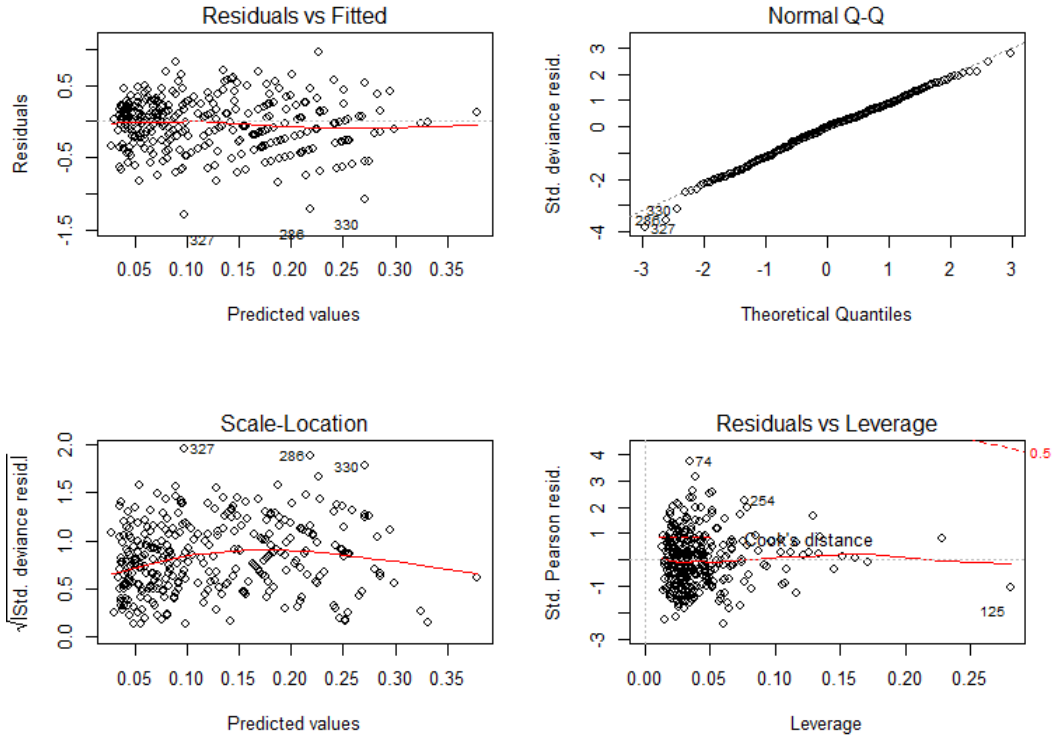


Figure 7: Residuals of the final model in the linear domain.

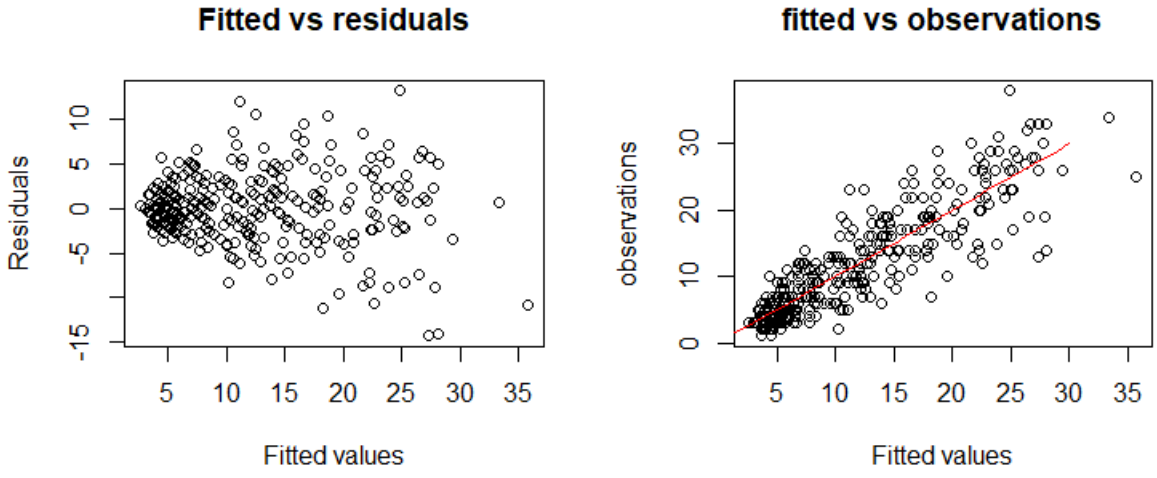


Figure 8: Fitted vs residuals and fitted vs observations in the original, untransformed domain

3 Clothing

In this second part of the assignment, we return to the clothing insulation level data from the last assignment. Here, the response variable has been changed to the number of times during the workday that the individual subject has changed the level of clothing, e.g. taken a sweater on/off. Furthermore, an extra variable representing the number of hours the subject spent at work the day of observation (e.g. 7 hours) and the number of observations during the day. We model the response variable using first the binomial distribution and then the Poisson distribution.

3.1 Binomial distribution

Modeling using the binomial distribution is very useful for binary data and ratios. In this example, we can define the response variable as the ratio of the clothing variable to the number of observations during the day. In this way, the number of observations is used as a type of offset.

In the binomial family, there are 5 link functions, all of which are tested out using a large model of all 5 variables and all interactions. Model sufficiency is evaluated using the sum of the deviance residuals against a χ^2-distribution of the same degrees of freedom as the model. However, all five models showed too high deviance, as the p-values were below 5%. This indicates some problem with the model, such as outliers, wrong choice of family or link function, or influential observations. All the link functions within the binomial family were tested. The issue is therefore not a wrong link function. To better understand the data the residuals are checked. This is done by looking at both the deviance and Pearson residuals. Both the Pearson residual and deviance residuals show that there are 2 observations which are outliers.

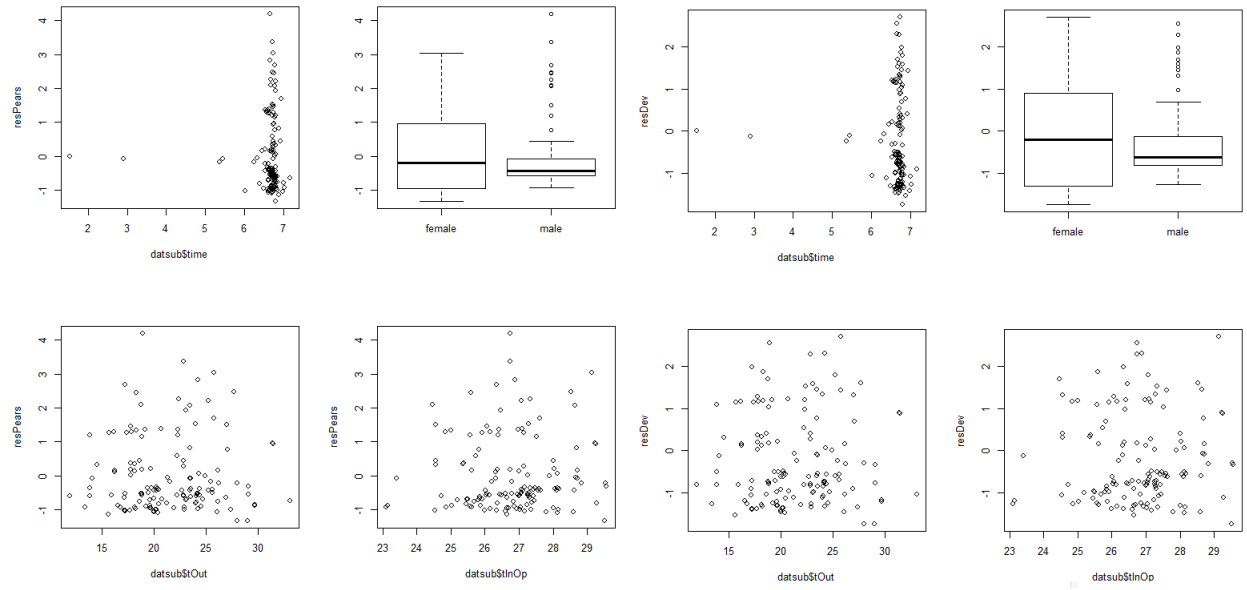


Figure 9: Pearson and deviance residuals of the data

These are observations at 1 and 3 hours. These two observations are therefore removed and the 5 link functions are tested again. Removing the two observations did not have a significant effect as all the 5 link functions were still not significant. The reason neither of the link functions provide a good fit must therefore be overdispersion of the data. Overdispersion is when the data show a greater variance in the response than the mean. It was therefore decided to use overdispersion model on the full data set. The link function "cauchit" provided the best fit to the data. Due to the overdispersion the family of is tested as quasibinomial. The first model includes all the variables and their interactions.

```
> formula = resp ~ time*sex*tOut*tInOp
> summary(fitbin31)

Call:
glm(formula = formula, family = quasibinomial(link = "cauchit"),
    data = datsub)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7132  -1.0987  -0.5746   0.2216   2.6861

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             512.55991 1691.97701   0.303    0.762
time                    -74.83485  250.18219  -0.299    0.765
sexmale                -787.00398 7415.53500  -0.106    0.916
tOut                    -16.63966   90.94051  -0.183    0.855
tInOp                   -20.44554   62.76870  -0.326    0.745
time:sexmale            100.54792 1095.87416   0.092    0.927
time:tOut                 2.33268   13.43466   0.174    0.862
sexmale:tOut            117.30169  434.48249   0.270    0.788
time:tInOp                2.98769    9.28645   0.322    0.748
sexmale:tInOp            19.33941  294.72828   0.066    0.948
tOut:tInOp                0.67899    3.30559   0.205    0.838
time:sexmale:tOut       -16.22290   64.25801  -0.252    0.801
time:sexmale:tInOp       -2.25717   43.54758  -0.052    0.959
time:tOut:tInOp          -0.09613    0.48845  -0.197    0.844
sexmale:tOut:tInOp       -4.12334   16.99987  -0.243    0.809
time:sexmale:tOut:tInOp   0.56614    2.51377   0.225    0.822

(Dispersion parameter for quasibinomial family taken to be 1.455665)

    Null deviance: 193.66  on 135  degrees of freedom
Residual deviance: 159.32  on 120  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 25
```

Figure 10: Summary of the original binomial model

The model is reduced by using backward selection, where the least significant parameters are removed one at a time. Lower-order interactions or parameters are not removed before the higher interactions. Backward selection yields a model where only sex is significant. The confidence interval of the coefficient estimates are also calculated. The summary and the confidence interval of the parameters can be seen in 11. It should be noted that the confidence intervals of coefficient estimates are based on the profiled log-likelihood function for logistic models in R. Based on the coefficients it is seen that the intercept, female clothing, will be -1.6 and the males will have a negative contribution of -3.6.

```
> summary(fitbin31)

Call:
glm(formula = resp ~ sex, family = quasibinomial(link = "cauchit"),
    data = datsub)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3735  -0.7849  -0.7849   0.1631   3.0756

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.6679     0.2782  -5.995 1.78e-08 ***
sexmale      -3.5969     1.4179  -2.537   0.0123 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for quasibinomial family taken to be 1.344408)

    Null deviance: 193.66  on 135  degrees of freedom
Residual deviance: 172.57  on 134  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 7

> confint(fitbin31)
Waiting for profiling to be done...
                2.5 %    97.5 %
(Intercept) -2.323910 -1.201812
sexmale     -7.627035 -1.483080
```

Figure 11: Summary and confidence interval of the final binomial model

3.2 Poisson distribution

The same set of data is investigated with a Poisson distribution. It makes more sense to discuss rates in Poisson. Time is therefore used as an offset while the number of observations are discarded per subject per day (*nobs*).

The Poisson family has 3 different link functions. The link function "log" was significant and is therefore a good fit to the data. This link function is therefore used in the Poisson model, and backward selection was performed. When this is done only sex was left as a significant parameter which is the same as the binomial model. The model summary and the confidence interval can be seen in Figure 12.

```
> summary(fitpois1)

Call:
glm(formula = clo ~ sex + offset(log(time)), family = poisson(link = "log"),
    data = datsub)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.3519  -0.7790  -0.7667   0.2033   2.4521

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.0573     0.1291 -15.935  < 2e-16 ***
sexmale      -1.0632     0.2632  -4.039 5.37e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 168.09  on 135  degrees of freedom
Residual deviance: 149.15  on 134  degrees of freedom
AIC: 270.13

Number of Fisher Scoring iterations: 6

> confint(fitpois1)
Waiting for profiling to be done...
                2.5 %     97.5 %
(Intercept) -2.321423 -1.8144627
sexmale     -1.605472 -0.5673245
```

Figure 12: Summary and confidence interval of the final Poisson model

Model analysis is done by looking at the Pearson and deviance residuals for the model. These can be seen in Figure 13. The figure shows that there is a much larger deviation for the females compared to the males.

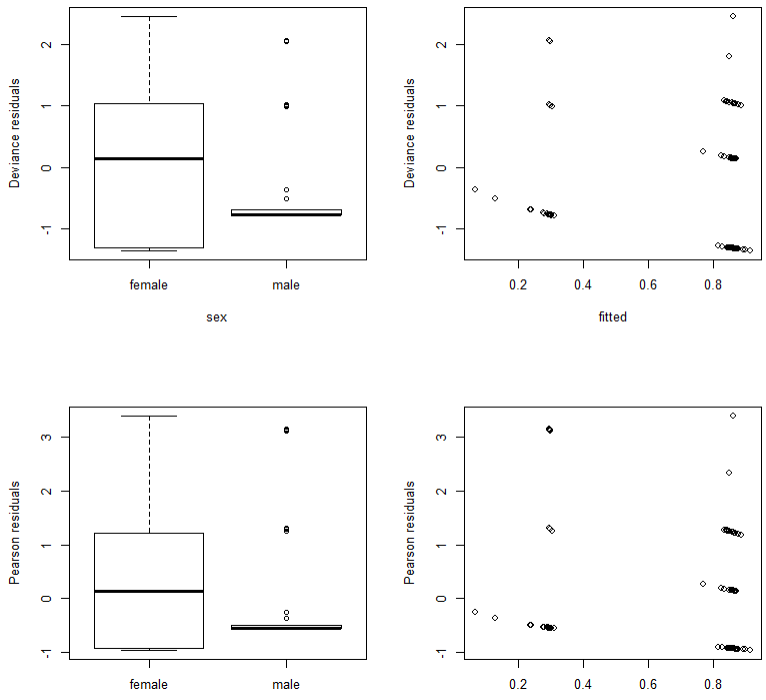


Figure 13: Summary and confidence interval of the final poisson model

3.3 Interpretation of models

The final model has the same significant parameter for both families as the clothing is only dependent on the sex of the observations. Due to overdispersion of the data the binomial model had to be carried out as quasibinomal. This shows that both methods can be used to describe the data, but the Poisson model has lower residuals. The Poisson model is therefore a better fit to the data compared to the binomial model.

3.4 Fit models using subjID instead of Sex

3.4.1 Binomial

In part 3.1 none of the link function were a good fit to the data. All 5 link functions are therefore retested with subjId. These showed a much better fit to the data with the link function "probit" being the best fit. This model did not need to account for overdispersion. The model was fitted in same method as in 3.1. Backward selection was used to reduce the model. The original model, seen in figure 14 showed that subjId was now significant while the rest were not.

```
Model:
resp ~ time + subjId + tOut + tInOp
       Df Deviance    AIC F value    Pr(>F)
<none>      73.873 271.40
time    1   73.915 269.44  0.0495    0.8245
subjId 46  192.260 297.79  2.9961 5.575e-06 ***
tOut    1   73.899 269.43  0.0305    0.8617
tInOp   1   75.297 270.83  1.6576    0.2014
```

Figure 14: Model analysis of the original model

The model can be reduced down to only containing the subjId. The confidence interval of the subjId's shows that the different subjId's dress completely different. This can be seen in figure 15 where it can be seen that some subjects do not dress significantly different than subject 11, these will have zero in their confidence interval, while other dress quite significantly different than subject 11.

```
> confint(fitbin31)
                  2.5 %    97.5 %
(Intercept)          NA  56.84790
subjId17    -28.850445 379.51102
subjId19     -8.940981 396.34551
subjId29    -15.509173  16.42412
subjId35    -27.626184 388.69443
subjId43     23.795257 398.06869
subjId47    -27.652719 387.83309
subjId49    -16.642922  16.89327
subjId51    -28.069434 383.82302
subjId55     67.544979 399.59655
subjId57    -27.265069 386.02507
subjId59    -26.689936 392.75887
subjId61    -15.069612  14.44506
subjId63     38.462375 398.76835
subjId65    -29.339795 378.48357
subjId69    -27.447197 387.23735
subjId71    -17.819158  17.97641
subjId75    -16.001389  16.14751
subjId85    -16.107875  15.91230
subjId87    -27.497669 386.73720
subjId89    -16.282065  16.46121
subjId91    -14.451849  14.69932
subjId93    -15.639911  15.40997
subjId99    -16.886222  17.10351
subjId105   -16.022595  15.90402
subjId107   -13.510004  13.97950
subjId111   -19.648719 395.40826
subjId113   -28.231524 381.69182
subjId119   -15.815963  15.56621
subjId123   -14.817189  15.22140
subjId125   -29.815971 372.07546
subjId127   -16.532240  16.37823
subjId129   -16.260317  16.40093
subjId137   -27.482967 385.80719
subjId141   -27.209006 384.68293
subjId145   -27.408980 385.88118
subjId149   -26.634712 391.67950
subjId153   -15.337372  15.41835
subjId157   -26.682971 389.22317
subjId167   -28.172612 383.23876
subjId171    -9.618539 395.66796
subjId173   -26.877455 388.18285
subjId183   -16.624336  15.82639
subjId187   -26.997352 385.84734
subjId189   -28.791286 380.61997
subjId193    46.303236 399.33790
subjId199   -29.365925 378.45744
```

Figure 15: Confidence interval of the subject id

The residuals plot confirm that there is a large deviance in the different observations. The deviance residual describes deviance contribution from each observation. Residuals should be evenly spread, however, it appears that the residuals of subjIds are grouped together in small groups.

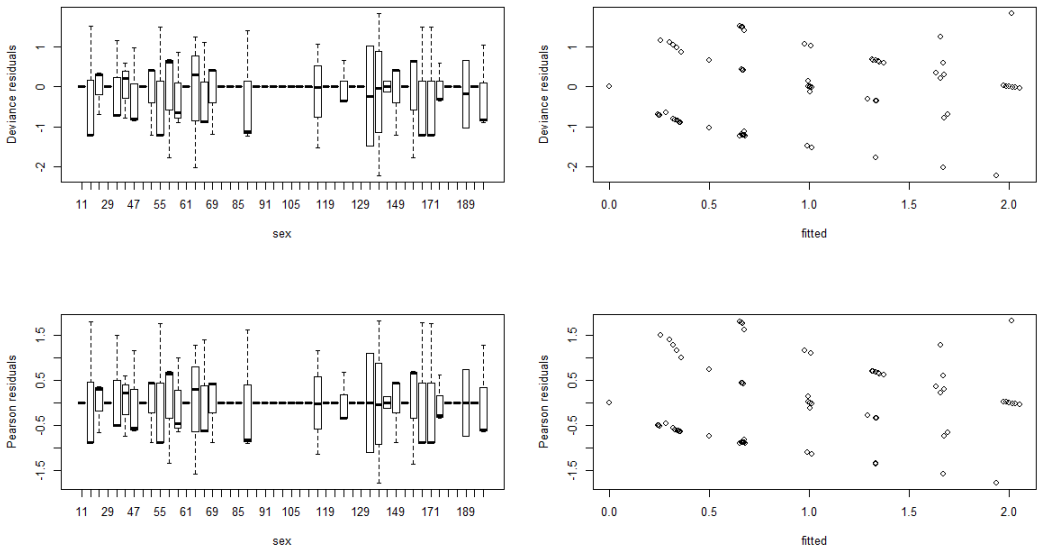


Figure 16: Residual plot of the subject id

3.4.2 Poisson

In part 3.2 the link function "log" for the Poisson family showed to be a viable fit. This link function is therefore used again in this section. Just as with the binomial distribution sex is replaced by the subjId. Backward selection was used to reduce the model and the result was the same as in the binomial, where the final model was only dependant on the subjId. As it was seen in the binomial model both the coefficient estimates and the standard deviation had a large variation. This can be seen in Figure 17.

```
Call:
glm(formula = clo ~ subjId + offset(log(time)), family = poisson(link = "log"),
    data = datsub)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.96821  -0.30262  -0.00006   0.00855   1.33395

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.220e+01  8.973e+03  -0.002    0.998
subjId17     1.989e+01  8.973e+03   0.002    0.998
subjId19     2.081e+01  8.973e+03   0.002    0.998
subjId29     2.521e-03  1.269e+04   0.000    1.000
subjId35     1.890e+01  8.973e+03   0.002    0.998
subjId43     2.080e+01  8.973e+03   0.002    0.998
subjId47     1.920e+01  8.973e+03   0.002    0.998
subjId49    -1.655e-02  1.419e+04   0.000    1.000
subjId51     1.989e+01  8.973e+03   0.002    0.998
subjId55     1.988e+01  8.973e+03   0.002    0.998
subjId57     2.059e+01  8.973e+03   0.002    0.998
subjId59     1.927e+01  8.973e+03   0.002    0.998
subjId61     1.993e-02  1.269e+04   0.000    1.000
subjId63     2.081e+01  8.973e+03   0.002    0.998
subjId65     1.921e+01  8.973e+03   0.002    0.998
subjId69     1.989e+01  8.973e+03   0.002    0.998
subjId71     2.069e-03  1.419e+04   0.000    1.000
subjId75     1.016e-01  1.417e+04   0.000    1.000
subjId85    -9.972e-03  1.269e+04   0.000    1.000
subjId87     1.990e+01  8.973e+03   0.002    0.998
subjId89    -1.384e-02  1.269e+04   0.000    1.000
subjId91    -4.932e-03  1.269e+04   0.000    1.000
subjId93    -7.467e-03  1.269e+04   0.000    1.000
subjId99    -8.293e-03  1.269e+04   0.000    1.000
subjId105   -2.510e-03  1.269e+04   0.000    1.000
subjId107    5.707e-01  1.146e+04   0.000    1.000
subjId111    2.030e+01  8.973e+03   0.002    0.998
subjId113    2.030e+01  8.973e+03   0.002    0.998
subjId119    8.488e-04  1.269e+04   0.000    1.000
subjId123   -1.077e-02  1.269e+04   0.000    1.000
subjId125    2.058e+01  8.973e+03   0.002    0.998
subjId127   -1.322e-02  1.269e+04   0.000    1.000
subjId129   -2.527e-03  1.269e+04   0.000    1.000
subjId137    2.030e+01  8.973e+03   0.002    0.998
subjId141    2.099e+01  8.973e+03   0.002    0.998
subjId145    2.030e+01  8.973e+03   0.002    0.998
subjId149    1.989e+01  8.973e+03   0.002    0.998
subjId153    4.987e-03  1.269e+04   0.000    1.000
subjId157    2.059e+01  8.973e+03   0.002    0.998
subjId167    1.989e+01  8.973e+03   0.002    0.998
subjId171    1.990e+01  8.973e+03   0.002    0.998
subjId173    2.058e+01  8.973e+03   0.002    0.998
subjId183    1.005e-02  1.269e+04   0.000    1.000
subjId187    2.099e+01  8.973e+03   0.002    0.998
subjId189    1.961e+01  8.973e+03   0.002    0.998
subjId193    2.099e+01  8.973e+03   0.002    0.998
subjId199    1.919e+01  8.973e+03   0.002    0.998

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 168.094  on 135  degrees of freedom
Residual deviance:  63.475  on  89  degrees of freedom
AIC: 274.46
```

Figure 17: Summary of the model

The residuals shows the same pattern as seen in the binomial model where is was seen that the observations has a large deviance in their contribution to the residuals. This can be seen in Figure 18.

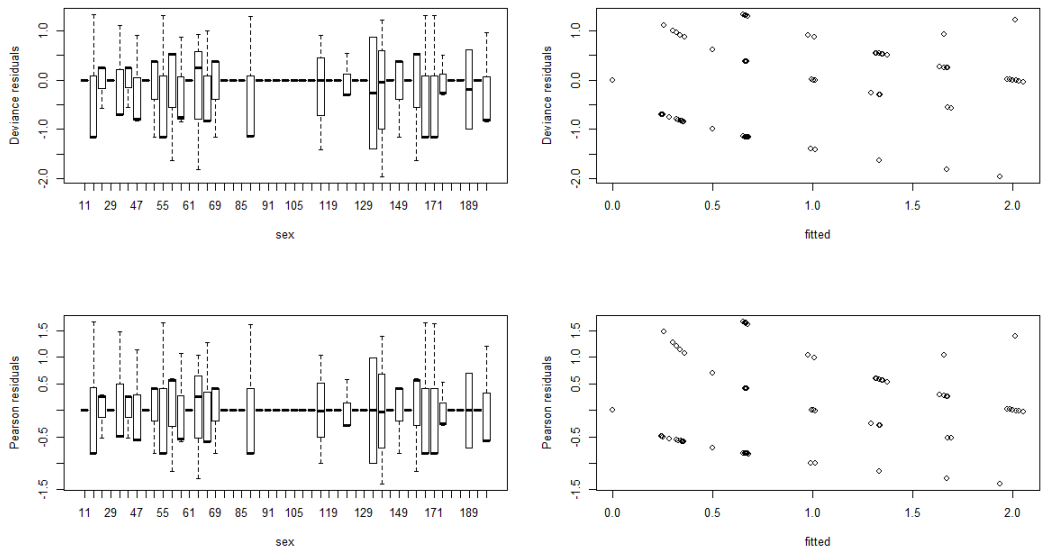


Figure 18: Residual plot of the poisson model

3.5 Conclusion

GLM was used to model the difference in clothing changes between sexes and the individual subject. It was seen that when looking at the two sexes, male and female, females had a higher variance in the level of clothing they were wearing whereas males dressed with a significantly lower variation. The models also showed that males were wearing a lower level of insulation compared to the females. When sexes was replaced by subject ID, it was seen that it was no longer possible to generalize as the subjects dress significantly different.

4 Appendix

4.1 Part A

```
######### Asignment 2, Ozone

rm(list=ls())
setwd("C:/Users/ander/Documents/10. Semester/Advanced data analysis/Assignment 2")
library("ggplot2")
library("dplyr")
library("GGally")
library("car")
library("MASS")
library("ellipse")
library(gclus)
library("LaplacesDemon")
library(matlib)
data(ozone)
head(ozone)

# log-Likelihood function, probably NOT poisson
pois <- function(lamb){
  sum(dpois(ozone$Ozone,lambda=lamb,log=TRUE))}
optpois <- optimize(pois,c(0,20),maximum=TRUE)
optpois$maximum


poisd=dpois(seq(0,40,by=1),lambda=optpois$maximum)
gammad = dgamma(seq(0,40,by=1),shape=2,rate=1/6)
lnormd = dlnorm(seq(0,40,by=1),mean=mean(log(ozone$Ozone)),sd=sd(log(ozone$Ozone)))
invgaussd = dinvgaussian(seq(0.1,40),mu=mean(ozone$Ozone),lambda=1/(1/330*sum(1/ozone$Ozone-1/mean(ozo


par(mfrow=c(1,1))
a=hist(ozone$Ozone,breaks=100,freq=FALSE, xlim=c(0,40),ylim = c(0, 0.3),main="Density of ozone")
lines(poisd/sum(poisd)*2,col='red',lwd=2)
lines(gammad/sum(gammad)*2,col='green',lwd=2)
lines(lnormd/sum(lnormd)*2,col='blue',lwd=2)
lines(invgaussd/sum(invgaussd)*2,col='yellow',lwd=2)
legend("topright", legend=c("Poisson", "Gamma","log-normal","Inverse Gaussian"),col=c("red", "green","


### Simple presentation of the data
par(mfrow=c(2,4))
plot(ozone$Ozone~ozone$Temp, xlab='Temperature', ylab='Ozone')
plot(ozone$Ozone~ozone$InvHt, xlab='Inversion base height', ylab='Ozone',type="p")
plot(ozone$Ozone~ozone$Pres, xlab='Pressure', ylab='Ozone')
plot(ozone$Ozone~ozone$Vis, xlab='Visibility', ylab='Ozone')
plot(ozone$Ozone~ozone$Hgt, xlab='Vandenburg Height', ylab='Ozone')
plot(ozone$Ozone~ozone$Hum, xlab='Humidity', ylab='Ozone')
plot(ozone$Ozone~ozone$InvTmp, xlab='Inversion temperature', ylab='Ozone')
plot(ozone$Ozone~ozone$Wind, xlab='Wind speed', ylab='Ozone')
```

```
##2.2 Fit a general linear model, without 2nd degree terms
formula1 = Ozone ~ Temp+InvHt+Pres+Vis+Hgt+Hum+InvTmp+Wind
fitlm1 <- lm(formula = formula1, data = ozone)
summary(fitlm1)

drop1(fitlm1,test="F")
fitlm2 <- update(fitlm1,.~. -Wind)

drop1(fitlm2,test="F")
fitlm3 <-update(fitlm2,.~. -Pres)

drop1(fitlm3,test="F")
fitlm4 <- update(fitlm3,.~. -Hgt)

drop1(fitlm4,test="F")
fitlm5 <- update(fitlm4,.~. -InvTmp)

drop1(fitlm5,test="F")
fitlm6 <- update(fitlm5,.~. -Vis)

#RMSE
(sqrt(sum((ozone$Ozone-fitted(fitlm6))^2)))



############### Do some residual plots
par(mfrow=c(3,2))
plot(residuals(fitlm6)) #Residuals versus obs number
plot(fitted(fitlm6),residuals(fitlm6)) #residuals versus fitted values
plot(ozone$Ozone,residuals(fitlm6)) #residuals versus response, this should always be linear!
plot(ozone$Temp,residuals(fitlm6)) #residuals versus Temperature
plot(ozone$InvHt,residuals(fitlm6)) #residuals versus Inverse height
plot(ozone$Hum,residuals(fitlm6)) #residuals versus Humidity

par(mfrow=c(1,3))
plot(fitted(fitlm6),residuals(fitlm6),xlab="Fitted values",ylab="Residuals",main="Fitted vs residuals"
plot(fitted(fitlm6),ozone$Ozone,xlab="Fitted values",ylab="observations",main="fitted vs observations"
lines(0:30,0:30,col="red")
qqPlot(fitlm6,simulate=FALSE,xlab="t Quantiles",ylab="Studentized residuals",main="QQplot",lwd=1)


## 2.3 Fit a general linear model with log
formula22 = log(Ozone) ~ Temp+InvHt+Pres+Vis+Hgt+Hum+InvTmp+Wind
fitlm21 <- lm(formula=formula22,data=ozone)
drop1(fitlm21,test="F")
fitlm21 <- update(fitlm21,.~. -InvTmp)
drop1(fitlm21,test="F")
fitlm21 <- update(fitlm21,.~. -Wind)
drop1(fitlm21,test="F")
fitlm21 <- update(fitlm21,.~. -Hgt)
drop1(fitlm21,test="F")
fitlm21 <- update(fitlm21,.~. -Pres)
```

```
drop1(fitlm21,test="F")
fitlm21 <- update(fitlm21,.~. -Vis)
summary(fitlm21)

### Do some residual plots
par(mfrow=c(1,3))
plot(exp(fitted(fitlm21)),residuals(fitlm21),xlab="Fitted values",ylab="Residuals",main="Fitted vs res
plot(exp(fitted(fitlm21)),ozone$Ozone,xlab="Fitted values",ylab="observations",main="fitted vs observa
lines(0:30,0:30,col="red")
qqPlot(fitlm21,simulate=FALSE,xlab="t Quantiles",ylab="Studentized residuals",main="QQplot",lwd=1)

par(mfrow=c(2,3))
plot(residuals(fitlm21)) #Residuals versus obs number
plot(fitted(fitlm21),residuals(fitlm21)) #residuals versus fitted values
plot(log(ozone$Ozone),residuals(fitlm21)) #residuals versus response, this should always be linear!
plot(ozone$Temp,residuals(fitlm21)) #residuals versus Temperature
plot(ozone$InvHt,residuals(fitlm21)) #residuals versus Inverse Height
plot(ozone$Vis,residuals(fitlm21)) #residuals versus Humidity
par(mfrow=c(1,1))
hist(residuals(fitlm21))

#RMSE
(sqrt(sum((ozone$Ozone-exp(fitted(fitlm21)))^2)))



############ 2.4 GLM part
formula5 = Ozone ~ Temp*InvHt*Pres*Vis*Hgt*Hum*InvTmp*Wind+
  I(Temp^2)+I(InvHt^2)+I(Pres^2)+I(Vis^2)+I(Hgt^2)+I(Hum^2)+I(InvTmp^2)+I(Wind^2)+
  I(Temp^3)+I(InvHt^3)+I(Pres^3)+I(Vis^3)+I(Hgt^3)+I(Hum^3)+I(InvTmp^3)+I(Wind^3)
fitIG1= glm(formula = formula5,family=inverse.gaussian(link="log"),data=ozone)
fitIG2= glm(formula = formula5,family=inverse.gaussian(link="identity"),data=ozone)
fitIG3= glm(formula = formula5,family=inverse.gaussian(link="inverse"),data=ozone)
fitG1 = glm(formula = formula5,family=Gamma(link="log"),data=ozone)
fitG2 = glm(formula = formula5,family=Gamma(link="identity"),data=ozone)
fitG3 = glm(formula = formula5,family=Gamma(link="inverse"),data=ozone)
AIC(fitIG1)
AIC(fitIG2)
AIC(fitIG3)
AIC(fitG1)
AIC(fitG2)
AIC(fitG3)
(sqrt(sum((ozone$Ozone-fitted(fitIG1))^2)))
(sqrt(sum((ozone$Ozone-fitted(fitIG2))^2)))
(sqrt(sum((ozone$Ozone-fitted(fitIG3))^2)))
(sqrt(sum((ozone$Ozone-fitted(fitG1))^2)))
(sqrt(sum((ozone$Ozone-fitted(fitG2))^2)))
(sqrt(sum((ozone$Ozone-fitted(fitG3))^2)))

fitG = glm(formula = formula1,family=Gamma(link="inverse"),data=ozone)
drop1(fitG,test="F")
fitG <- update(fitG,.~. -Hgt)
drop1(fitG,test="F")
```

```r
fitG <- update(fitG,.~. -Wind)
drop1(fitG,test="F")
fitG <- update(fitG,.~. -InvTmp)
drop1(fitG,test="F")
fitG <- update(fitG,.~. -Vis)
drop1(fitG,test="F")
fitG <- update(fitG,.~. -Pres)
summary(fitG)
(sqrt(sum((ozone$Ozone-fitted(fitG))^2)))

res = (ozone$Ozone-fitted(fitG))
par(mfrow=c(1,2))
plot(fitted(fitG),res,xlab="Fitted values",ylab="Residuals",main="Fitted vs residuals")
plot(fitted(fitG),ozone$Ozone,xlab="Fitted values",ylab="observations",main="fitted vs observations")
lines(0:30,0:30,col="red")



############## 2.6
mu <- predict(fitG,type="response")
Vmu <- I(mu^2)
w2 <- 1/summary(fitG)$dispersion
w1 = 1
(W1 <- diag(w1*Vmu))
(W2 <- diag(w2*Vmu))

(Sigma1 = inv(t(model.matrix(fitG))%*%W1%*%model.matrix(fitG)))
(Sigma2 = inv(t(model.matrix(fitG))%*%W2%*%model.matrix(fitG)))

S2 = summary(fitG)$cov.scaled
S1 = summary(fitG)$cov.unscaled


formatC(S2, format = "e", digits = 2)
formatC(Sigma2, format = "e", digits = 2) #These are the same
  formatC(S1, format = "e", digits = 2)
formatC(Sigma1, format = "e", digits = 2) #These are the same


############### 2.7+8
formula4 = Ozone ~ Temp+InvHt+Pres+Vis+Hgt+Hum+InvTmp+Wind+
  I(Temp^2)+I(InvHt^2)+I(Pres^2)+I(Vis^2)+I(Hgt^2)+I(Hum^2)+I(InvTmp^2)+I(Wind^2)+
  I(Temp^3)+I(InvHt^3)+I(Pres^3)+I(Vis^3)+I(Hgt^3)+I(Hum^3)+I(InvTmp^3)+I(Wind^3)+
  Temp:InvHt+Temp:Pres+Temp:Vis+Temp:Hgt+Temp:Hum+Temp:InvTmp+Temp:Wind+
  InvHt:Pres+InvHt:Vis+InvHt:Hgt+InvHt:Hum+InvHt:InvTmp+InvHt:Wind+
  Pres:Vis+Pres:Hgt+Pres:Hum+Pres:InvTmp+Pres:Wind+
  Vis:Hgt+Vis:Hum+Vis:InvTmp+Vis:Wind+
  Hgt:Hum+Hgt:InvTmp+Hgt:Wind+
  Hum:InvTmp+Hum:Wind+InvTmp:Wind

fitG1 = glm(formula = formula4,family=Gamma(link="log"),data=ozone)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -InvTmp:Wind)
```

```
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Pres:Hum)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Temp:Wind)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Hgt:Wind)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -InvHt:InvTmp)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Temp:InvTmp)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Vis:Hum)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Hgt:InvTmp)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Temp:Hgt)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -InvHt:Hgt)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Temp:Pres)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Vis:Hgt)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Temp:Hum)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -InvHt:Wind)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Pres:Vis)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Vis:Wind)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -InvHt:Pres)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -InvHt:Vis)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Temp:Vis)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Vis:InvTmp)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Hum:InvTmp)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Pres:InvTmp)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Temp:InvHt)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Pres:Wind)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Hum:Wind)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Hgt:Hum)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(InvTmp^3))#######Nåede hertil
drop1(fitG1,test="F")
```

```
fitG1 <- update(fitG1,.~. -I(Temp^3))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(Wind^3))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(InvHt^3))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(Pres^3))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(Hgt^3))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(Hgt^2))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Wind)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(Wind^2))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Vis)
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(Vis^3))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -I(Vis^2))
drop1(fitG1,test="F")
fitG1 <- update(fitG1,.~. -Hgt)
summary(fitG1)
par(mfrow=c(2,2))
plot(fitG1)

fitG2 = glm(Ozone~Temp + Pres+Hum+InvTmp,family=Gamma(link="log"),data=ozone)
summary(fitG2)
(sqrt(sum((ozone$Ozone-fitted(fitG1))^2)))
(sqrt(sum((ozone$Ozone-fitted(fitG2))^2)))

x=1:13
c = coef(fitG1)
c = c[2:14]
x2 = names(c)
conf = confint(fitG1)
conf = conf[2:14,]
par(mfrow=c(1,1))
plot(x,c)
arrows(x, conf[,1], x, conf[,2], length=0.05, angle=90, code=3)
axis(1, at=1:13, labels=x2)

library(coefplot)

coefplot(fitG1, horizontal = TRUE, innerCI = 0, pointSize =2,numberAngle = -90,
         coefficients=c("Temp","InvHt","Pres","Hum","InvTmp","(Temp^2)","I(InvHt^2)","I(Pres^2)","I(Hu

ress = (ozone$Ozone-fitted(fitG1))
par(mfrow=c(1,2))
plot(fitted(fitG1),ress,xlab="Fitted values",ylab="Residuals",main="Fitted vs residuals")
plot(fitted(fitG1),ozone$Ozone,xlab="Fitted values",ylab="observations",main="fitted vs observations")
lines(0:30,0:30,col="red")
```

4.2 Part B

```
## Assignment 2 part B

rm(list=ls())
setwd()
#install.packages("ggplot2")
#install.packages("dplyr")
#install.packages("GGally")
#install.packages("ellipse")
#install.packages("car")
#install.packages("carData")
library("ggplot2")
library("dplyr")
library("GGally")
library("carData")
library("car")
library("MASS")
library("ellipse")

dat = read.csv("dat_count.csv",sep=";")
View(dat)
dat = dat[order(dat$sex),] #sort by sex
datsub = subset(dat,select = -c(day,subjId))

# Short presentation
par(mfrow=c(2,4))
plot(datsub$clo~datsub$time, xlab='time', ylab='clothing')
plot(datsub$clo~datsub$sex, xlab='time', ylab='clothing')
plot(datsub$clo~datsub$tOut, xlab='time', ylab='clothing')
plot(datsub$clo~datsub$tInOp, xlab='time', ylab='clothing')
plot(datsub$clo/datsub$nobs~datsub$time, xlab='time', ylab='response')
plot(datsub$clo/datsub$nobs~datsub$sex, xlab='time', ylab='response')
plot(datsub$clo/datsub$nobs~datsub$tOut, xlab='time', ylab='response')
plot(datsub$clo/datsub$nobs~datsub$tInOp, xlab='time', ylab='response')

################### B.1: Do a GLM based on binomial ####################
datsub$resp<-cbind(dat$clo,datsub$nobs-dat$clo)
formula = resp ~ time*sex*tOut*tInOp

fitbin1 = glm(formula=formula,family=binomial(link="logit"),data=datsub)
summary(fitbin1)#AIC 290
1-pchisq(160.02,120)
fitbin2 = glm(formula=formula,family=binomial(link="probit"),data=datsub)
summary(fitbin2)#AIC 289
1-pchisq(159.82,120)
fitbin3 = glm(formula=formula,family=binomial(link="cauchit"),data=datsub)
summary(fitbin3)#AIC 289
1-pchisq(159.32,120)
fitbin4 = glm(formula=formula,family=binomial(link="log"),data=datsub)
summary(fitbin4)#AIC 290
1-pchisq(160.32,120)
fitbin5 = glm(formula=formula,family=binomial(link="cloglog"),data=datsub)
```

```r
summary(fitbin5)#AIC 290
1-pchisq(160.18,120)

# None of the models are a good fit, since all are below alfa=5%
resDev <- residuals(fitbin2,type="deviance")
par(mfrow=c(2,2))
plot(datsub$time,resDev)
plot(datsub$sex,resDev)
plot(datsub$tOut,resDev)
plot(datsub$tInOp,resDev)

resPears <- residuals(fitbin2,type="pearson")
par(mfrow=c(2,2))
plot(datsub$time,resPears)
plot(datsub$sex,resPears)
plot(datsub$tOut,resPears)
plot(datsub$tInOp,resPears)

# We could argue for removing the two observations with time=1hr and 2hr?
datsub2 = datsub[order(datsub$time),] #sort by time
datsub2 = datsub2[-c(1,2),] #remove first 2

# Compute goodness of fit again

fitbin21 = glm(formula=formula,family=binomial(link="logit"),data=datsub2)
summary(fitbin21)#AIC 288
1-pchisq(158.75,118)
fitbin22 = glm(formula=formula,family=binomial(link="probit"),data=datsub2)
summary(fitbin22)#AIC 288
1-pchisq(158.62,118)
fitbin23 = glm(formula=formula,family=binomial(link="cauchit"),data=datsub2)
summary(fitbin23)#AIC 287
1-pchisq(157.28,118)
fitbin24 = glm(formula=formula,family=binomial(link="log"),data=datsub2)
summary(fitbin24)#AIC 289
1-pchisq(158.99,118)
fitbin25 = glm(formula=formula,family=binomial(link="cloglog"),data=datsub2)
summary(fitbin25)#AIC 288
1-pchisq(158.88,118)

# This did not have an effect!

###### Using overdispersion and the original data set
fitbin31 = glm(formula=formula,family=quasibinomial(link="cauchit"),data=datsub)
summary(fitbin31)

drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time:sex:tOut:tInOp)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time:tOut:tInOp)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time:sex:tOut)
drop1(fitbin31, test = "F")
```

```
fitbin31 = update(fitbin31,.~. -time:sex:tInOp)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -sex:tOut:tInOp)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -tOut:tInOp)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -sex:tOut)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time:tInOp)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time:tOut)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time:sex)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -sex:tInOp)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -tOut)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -tInOp)

# So the final model contains ONLY sex as predictor.
summary(fitbin31)
confint(fitbin31)
#confint.default(fitbin31)

#residual plots
par(mfrow=c(2,2))
plot(fitbin31)

Rd<-residuals(fitbin31,type='deviance')
Rp<-residuals(fitbin31,type='pearson')
par(mfrow=c(2,2))
plot(datsub$sex,Rd, xlab='sex', ylab='Deviance residuals')
plot(fitted(fitbin31),Rd, xlab='fitted', ylab='Deviance residuals')
plot(datsub$sex,Rp, xlab='sex', ylab='Pearson residuals')
plot(fitted(fitbin31),Rp, xlab='fitted', ylab='Pearson residuals')




################### B.2: Do a GLM based on Poisson ####################
formulapois = clo ~ offset(log(time))+sex*tOut*tInOp
datsub = subset(dat,select = -c(day,subjId))

fitpois1 = glm(formula = formulapois,family=poisson(link="log"),data=datsub)
summary(fitpois1) #AIC 277
1-pchisq(143.63,128)
# The poisson model with the "log" link function passed the test!


drop1(fitpois1,test="F")
```

```
fitpois1 = update(fitpois1,.~. -sex:tOut:tInOp)
summary(fitpois1)

drop1(fitpois1,test="F")
fitpois1 = update(fitpois1,.~. -tOut:tInOp)
summary(fitpois1)

drop1(fitpois1,test="F")
fitpois1 = update(fitpois1,.~. -sex:tOut)
summary(fitpois1)

drop1(fitpois1,test="F")
fitpois1 = update(fitpois1,.~. -sex:tInOp)
summary(fitpois1)

drop1(fitpois1,test="F")
fitpois1 = update(fitpois1,.~. -tOut)
summary(fitpois1)

drop1(fitpois1,test="F")
fitpois1 = update(fitpois1,.~. -tInOp)
# We end up with the same model as for the binomial!
summary(fitpois1)
confint(fitpois1)

#residual plots
par(mfrow=c(2,2))
plot(fitpois1)

Rd<-residuals(fitpois1,type='deviance')
Rp<-residuals(fitpois1,type='pearson')
par(mfrow=c(2,2))
plot(datsub$sex,Rd, xlab='sex', ylab='Deviance residuals')
plot(fitted(fitpois1),Rd, xlab='fitted', ylab='Deviance residuals')
plot(datsub$sex,Rp, xlab='sex', ylab='Pearson residuals')
plot(fitted(fitpois1),Rp, xlab='fitted', ylab='Pearson residuals')

################### B.4: Do a GLM based on Poisson ####################

datsub = subset(dat,select = -c(day,sex))
datsub$subjId=factor(datsub$subjId)
datsub$resp<-cbind(dat$clo,datsub$nobs-dat$clo)
formula = resp ~ offset(log(time))+subjId+tOut+tInOp

fitbin1 = glm(formula=formula,family=binomial(link="logit"),data=datsub)
summary(fitbin1)
1-pchisq(73.95,86)
fitbin2 = glm(formula=formula,family=binomial(link="probit"),data=datsub)
summary(fitbin2)
1-pchisq(73.87,86)
fitbin3 = glm(formula=formula,family=binomial(link="cauchit"),data=datsub)
summary(fitbin3)
1-pchisq(72.26,86)
```

```
fitbin4 = glm(formula=formula,family=binomial(link="log"),data=datsub)
summary(fitbin4)
1-pchisq(73.95,86)
fitbin5 = glm(formula=formula,family=binomial(link="cloglog"),data=datsub)
summary(fitbin5)
1-pchisq(74.0,86)

#model fra del 3.1
1-pchisq(159.32,120)


# Beware: the fit is overwritten for each update
#
fitbin31 = glm(formula=formula,family=binomial(link="probit"),data=datsub)
summary(fitbin31)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -tOut)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -time)
drop1(fitbin31, test = "F")
fitbin31 = update(fitbin31,.~. -tInOp)
drop1(fitbin31, test = "F")
summary(fitbin31)

summary(fitbin31)
confint(fitbin31)
#residuals
par(mfrow=c(2,2))
plot(fitbin31)

par(mfrow=c(2,2))
plot(fitpois1)

Rd<-residuals(fitbin31,type='deviance')
Rp<-residuals(fitbin31,type='pearson')
par(mfrow=c(2,2))
plot(datsub$subjId,Rd, xlab='sex', ylab='Deviance residuals')
plot(fitted(fitpois1),Rd, xlab='fitted', ylab='Deviance residuals')
plot(datsub$subjId,Rp, xlab='sex', ylab='Pearson residuals')
plot(fitted(fitpois1),Rp, xlab='fitted', ylab='Pearson residuals')

# Poisson
formulapois = clo ~ offset(log(time))+subjId+tOut+tInOp

datsub = subset(dat,select = -c(day,sex))
datsub$subjId=factor(datsub$subjId)

fitpois1 = glm(formula = formulapois,family=poisson(link="log"),data=datsub)
summary(fitpois1) #AIC 277
1-pchisq(63.475,89)
# The poisson model with the "log" link function passed the test!
drop1(fitpois1,test="F")
fitpois1 = update(fitpois1,.~. -tOut)
```

```r
drop1(fitpois1,test="F")
fitpois1 = update(fitpois1,.~. -tInOp)
drop1(fitpois1,test="F")
summary(fitpois1)
confint(fitpois1)

# We end up with the same model as for the binomial!
#now residual plots
par(mfrow=c(2,2))
plot(fitpois1)

Rd<-residuals(fitpois1,type='deviance')
Rp<-residuals(fitpois1,type='pearson')
par(mfrow=c(2,2))
plot(datsub$subjId,Rd, xlab='sex', ylab='Deviance residuals')
plot(fitted(fitpois1),Rd, xlab='fitted', ylab='Deviance residuals')
plot(datsub$subjId,Rp, xlab='sex', ylab='Pearson residuals')
plot(fitted(fitpois1),Rp, xlab='fitted', ylab='Pearson residuals')
```