

Project in regression - practical part

Contents

Exploratory data analysis	1
Missing data	4
Analysis using SOI phase	7
Estimating k with a linear regression model	7
Fitting a Tweedie model to the data	11
Estimating probability of zero rain in July	12
Determining k by minimizing AIC	13
Re-estimating with new value of k	14
Model diagnostics	15
Analysis using SOI directly	18
Should we include additional predictors?	20
Explore possible inclusion of nonlinear effects	21
Model Diagnostics	22
Reporting a final model and interpretation	24
20 35.57	24
Conclusion	26

In this project we seek to predict the rainfall at Eromanga in Queensland, Australia, during the month of July. We consider a data set that contains total rainfall in July at Eromanga in the period 1905 to 2024 with the exception of a few years. In addition we have measurements of the southern oscillation index (SOI, the standardized difference between the air pressures at Darwin and Haiti, related to el niño) for the same years. The hypothesis is that the SOI is related to the rainfall in Eromanga. We will throughout the project investigate this hypothesis and ultimately use the SOI index to predict the rainfall in Eromanga.

Exploratory data analysis

First, we load the data set

```
#Rain.data = read.table("~/Desktop/Studie/Master/First year/Block 1/Regression/Project/RaindataEromanga.txt", header = TRUE, colClasses = c("integer", "numeric", "integer"))  
Rain.data = read.table("RaindataEromanga.txt", header = TRUE, colClasses = c("integer", "numeric", "integer"))
```

We specify the column classes and print the `head()` and `summary()` of the data set to make sure that the data is read in correctly.

```
head(Rain.data)  
summary(Rain.data)
```

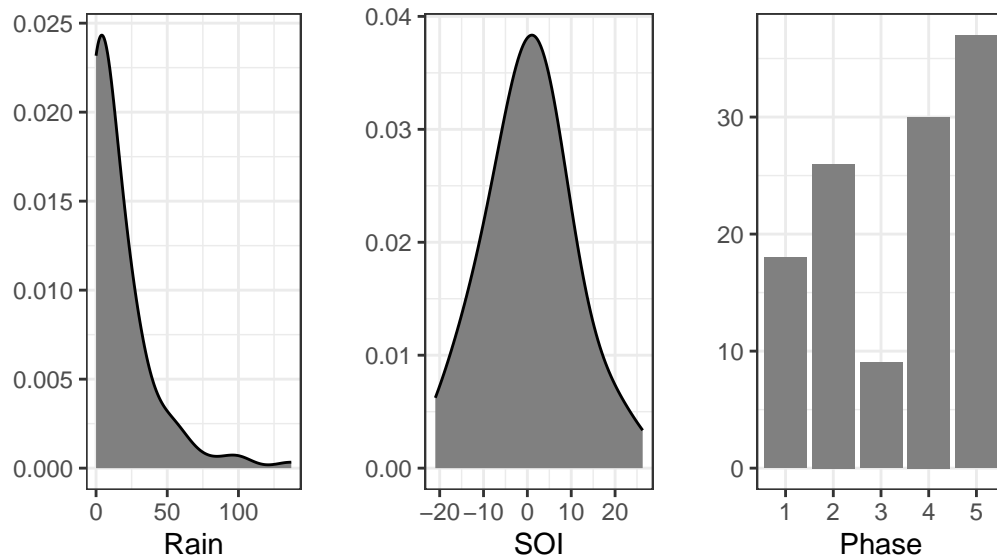
```
##   Year Rain Month   SOI Phase  
## 1 1905  0.0     7 -19.8     1  
## 2 1906 20.8     7   6.3     4  
## 3 1907 12.0     7  -5.1     5  
## 4 1908  NA     7  -3.2     5  
## 5 1909  NA     7   9.9     2
```

```
## 6 1910 NA 7 19.0 2
##      Year      Rain      Month      SOI      Phase
## Min.   :1905 Min.   : 0.00 Min.   :7 Min.   : -21.0000 1:18
## 1st Qu.:1935 1st Qu.: 0.00 1st Qu.:7 1st Qu.: -5.3575 2:26
## Median :1964 Median : 4.60 Median :7 Median : 0.8000 3: 9
## Mean   :1964 Mean   :15.13 Mean   :7 Mean   : 0.4817 4:30
## 3rd Qu.:1994 3rd Qu.:20.70 3rd Qu.:7 3rd Qu.: 5.8500 5:37
## Max.   :2024 Max.   :137.10 Max.   :7 Max.   : 26.3000
##      NA's   :9
```

The data set contains five variables: **Year**, **Rain**, **SOI**, **Phase** and **Month**. The **Month** variable is constant and will not be analyzed, but keep in mind, that the analysis carried out is for the month of July. The response variable **Rain** represents the total rainfall in July at Eromanga and contains nine missing values, which will be addressed in a subsequent part of the EDA. The **SOI** variable is the southern oscillation index, while **Phase** is a categorical variable indicating the SOI phase on five different levels:

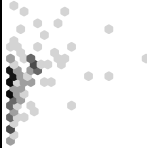
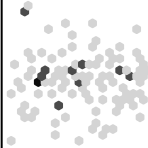
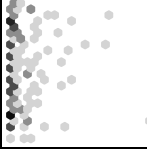
- Phase 1: Consistently negative
- Phase 2: Consistently positive
- Phase 3: Rapidly falling
- Phase 4: Rapidly rising
- Phase 5: Consistently near zero

We plot the marginal distributions of the variables to visually explore the data.



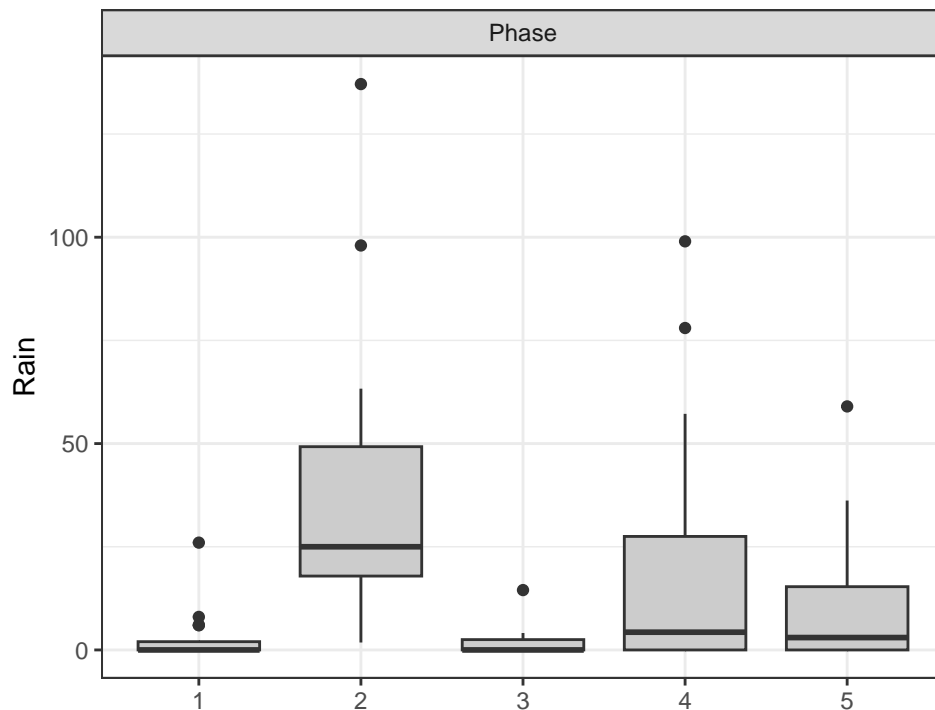
We observe that the response variable **Rain** is right-skewed, which should be considered in later analysis. Additionally, Category 3 in **Phase** has relatively few observations, which may also require consideration in subsequent analysis.

We proceed to investigate possible co-linearity between the variables in the data set. We assess correlation between the numerical variables **Rainfall**, **SOI** and **Year** with a correlation plot:

		SOI
	Year	-0.04
Rain	0.04	0.39

Year is almost uncorrelated with both **Rain** and **SOI**. There is a weak positive correlation between **Rain** and **SOI**.

To investigate the relation between **Rain** and **Phase** we consider the distribution of **Rain** stratified by **Phase**:



The boxplot suggests that the rainfall is larger in phase 2 and 4, while it appears particularly low in phase 1 and 3.

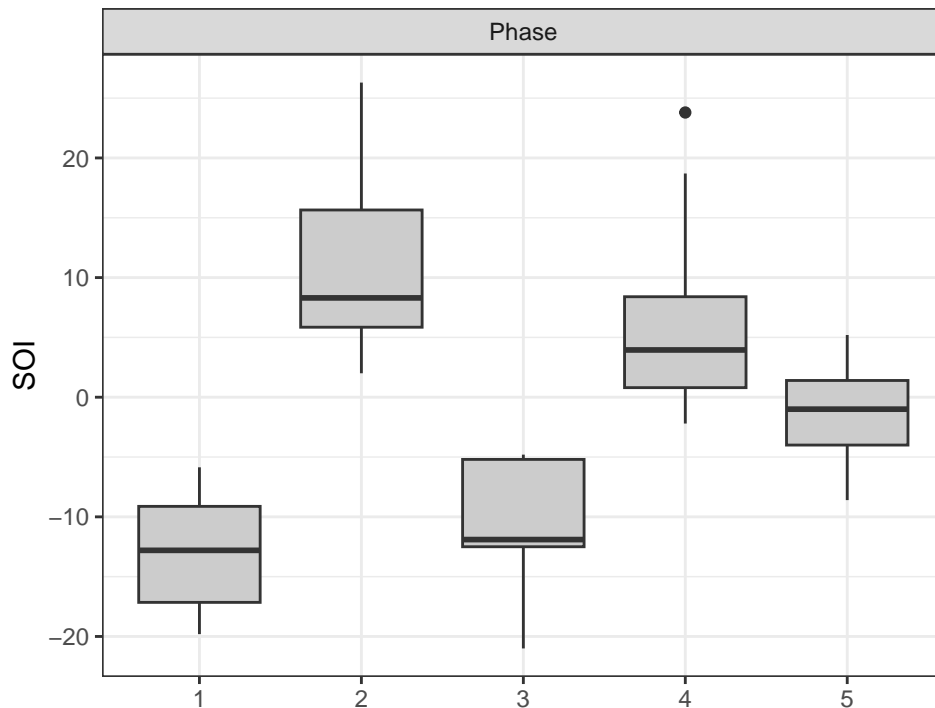
We finally investigate the relation between **SOI** and **Phase**. Since both variables describe aspects of **SOI** we expect some correlation between them.

```
summary(lm(SOI ~ Phase, Rain.data))
```

```
##
## Call:
## lm(formula = SOI ~ Phase, data = Rain.data)
##
## Residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -10.233  -3.880  -1.111   3.159  18.423
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -12.605      1.282  -9.832 < 2e-16 ***
## Phase2        23.014      1.668  13.799 < 2e-16 ***
## Phase3         1.838      2.221   0.828  0.409
## Phase4        17.982      1.622  11.088 < 2e-16 ***
## Phase5        11.244      1.563   7.193 6.9e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.439 on 115 degrees of freedom
## Multiple R-squared:  0.6915, Adjusted R-squared:  0.6808
## F-statistic: 64.46 on 4 and 115 DF,  p-value: < 2.2e-16
```

The simple linear model uncovers a high adjusted R^2 implying a high correlation between the two variables. All coefficients except **Phase3** are significant indicating that the **Phase** variable is a good predictor of **SOI**. This discovery can further be supported by the boxplot below.



Since the two predictors are collinear to some extent we will continue with care, when we consider models with both covariates. It is worth noting though, that while **SOI** is a numeric variable containing more exact values of **SOI**, **Phase** also contains information on the ‘direction’ of the **SOI**, i.e. whether it is increasing, decreasing or constant. So although the two variables are correlated they both contain information that the other does not so it could still be valuable to include both variables in a model. We will go into further details on this matter in subsequent sections.

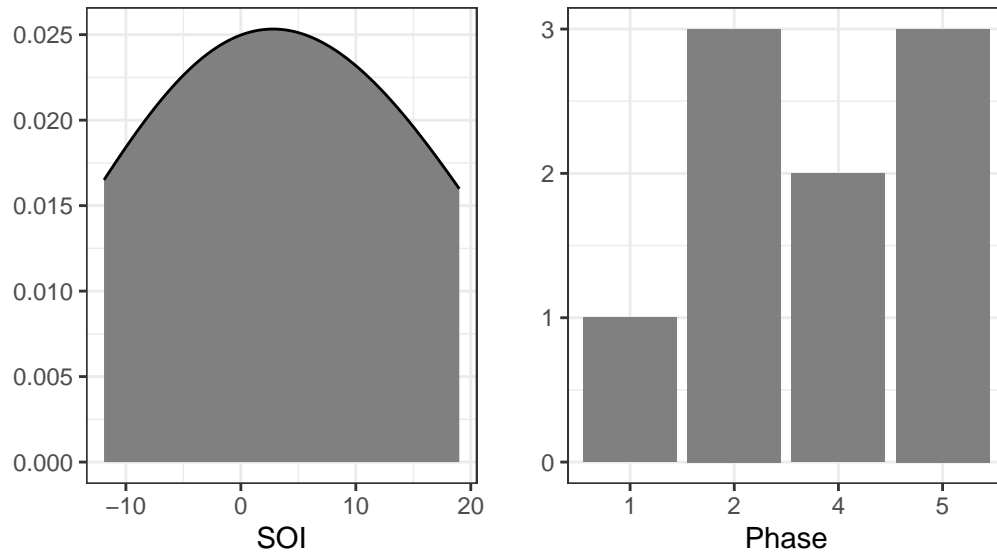
Missing data

With a better understanding of the data we proceed to investigate and handle the missing data. As noted previously there are only missing data (7.5 pct. missing) in the response variable **Rain**.

```
Rain.data %>% dplyr::filter(is.na(Rain))
```

##	Year	Rain	Month	SOI	Phase
## 1	1908	NA	7	-3.20	5
## 2	1909	NA	7	9.90	2
## 3	1910	NA	7	19.00	2
## 4	1911	NA	7	-11.90	1
## 5	1933	NA	7	3.30	4
## 6	2021	NA	7	16.26	4
## 7	2022	NA	7	7.63	2
## 8	2023	NA	7	-3.32	5
## 9	2024	NA	7	-5.83	5

We plot the marginal distribution of the variables with missing data to obtain a visual understanding of the missing data:



Given the small number of missing observations, it is difficult to identify any clear trends in the missingness. We note that the missingness occurs in consecutive years (except 1933), but there are no distinct pattern beyond that. Without metadata and additional knowledge we are unable to determine if the data is missing completely at random (MCAR), at random (MAR) or not at random (MNAR). It seems unlikely that observations of large rainfall were selectively deleted or that someone was too lazy to record rainfall during heavy rainfalls. A more plausible explanation, given the consecutive years, is that the equipment may be broken for consecutive years or there was a lack of funding these years. The latter explanations would imply that the data is MCAR which implies MAR and we therefore choose to adopt this assumption.

Assuming MAR we can use multiple imputation techniques to impute the missing data. If the MAR assumption holds the imputed values are unbiased and the variation in the data set is preserved. We use the `mice()` function (multiple imputations using chained equations) from the package `mice` to perform the multiple imputations. This procedure use random draws from the conditional distribution of the target variable given the other variables. That is, we take a bootstrap sample from our data and fit a regression model to this sample to predict the missing values. We use *predictive mean matching* (PMM) to replace the missing values in each imputed data set. This method uses the value of a donor observations to fill in the missing values. The donors are identified by matching the predicted value of the target to the donor value. PMM does not require any distributional assumptions and is therefore a fairly robust method (F. E. Harell, Jr., 2015).

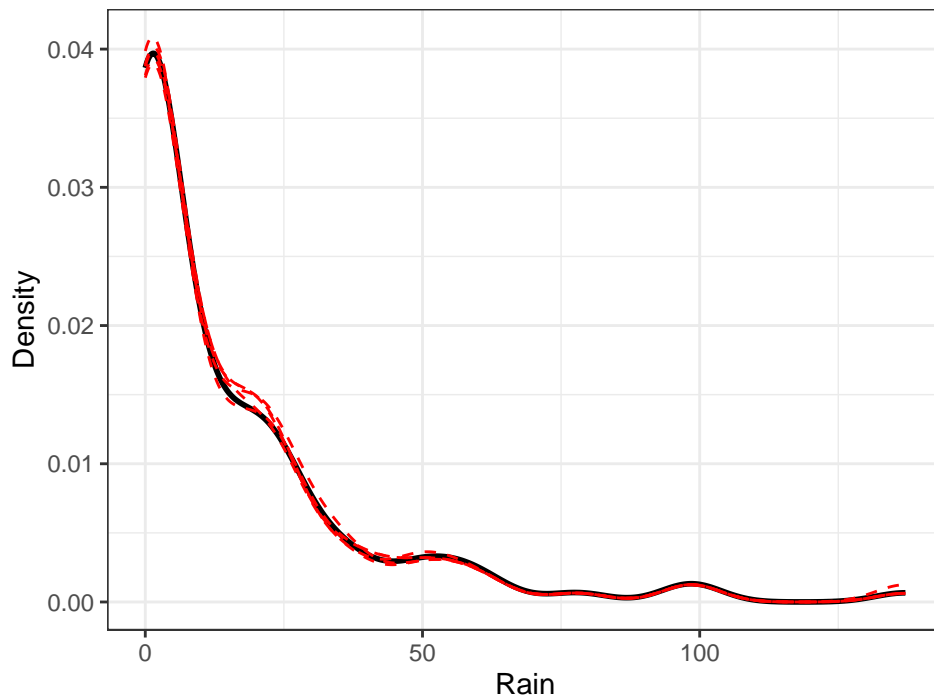
We run the imputations and display the first five rows of the first two imputed data sets:

```
Rain.data.impute <- mice(Rain.data, method = "pmm",
                        m = 5, seed = 10102024,
                        printFlag = FALSE)
head(complete(Rain.data.impute,1))
head(complete(Rain.data.impute,2))
```

```
##   Year Rain Month   SOI Phase
## 1 1905  0.0     7 -19.8     1
## 2 1906 20.8     7   6.3     4
## 3 1907 12.0     7  -5.1     5
## 4 1908  0.0     7  -3.2     5
## 5 1909 19.8     7   9.9     2
## 6 1910  2.8     7  19.0     2
##   Year Rain Month   SOI Phase
## 1 1905  0.0     7 -19.8     1
## 2 1906 20.8     7   6.3     4
## 3 1907 12.0     7  -5.1     5
## 4 1908  1.2     7  -3.2     5
## 5 1909 19.0     7   9.9     2
## 6 1910 19.0     7  19.0     2
```

We plot the density of the five imputed data sets (red dotted) along with the density of the original data set (black solid) to check if the imputed data sets resemble each other and the original data set.

Density plot of original data and imputed data sets



We note that the densities for all the imputed data sets look very similar to the original data set. We therefore proceed with the imputed data sets.

Analysis using SOI phase

In the following part of the project we seek to fit the Tweedie exponential dispersion model to the data, and predict rainfall as a function of the SOI phase. In order to fit a Tweedie model to the data we need to estimate the nuisance parameter k which we assume to be between 1 and 2.

Estimating k with a linear regression model

We initially try to estimate k using the linear relation $VY = \psi\mathcal{V}(\mu) = \psi\mu^k$. This implies that $\log(VY) = \log(\psi) + k\log(\mu)$. We can therefore estimate k by a linear regression of $\log(VY)$ on $\log(\mu)$. We estimate the variance and mean of the response variable within each SOI phase for each imputed data set using the empirical mean and variance:

```
grouped_imputatations <- list()
for (i in 1:5) {
  grouped_imputatations[[i]] <- complete(Rain.data.impute, i) %>% group_by(Phase) %>%
    summarise(meanY = mean(Rain), varY = var(Rain))
}
pander(grouped_imputatations)
```

Phase	meanY	varY
1	2.733	40.17
2	32.7	915.6
3	2.344	22.98
4	18.03	640.5
5	8.359	161

•

Phase	meanY	varY
1	2.733	40.17
2	33.06	893.2
3	2.344	22.98
4	18.25	634.8
5	8.716	158.8

•

Phase	meanY	varY
1	3.878	57.13
2	38.48	1300
3	2.344	22.98
4	16.98	622.6
5	8.792	157.8

•

Phase	meanY	varY
1	2.733	40.17
2	33.18	895.8

Phase	meanY	varY
3	2.344	22.98
4	17.71	632.1
5	9.538	163.5

•

Phase	meanY	varY
1	2.733	40.17
2	32.33	921.7
3	2.344	22.98
4	16.72	627.5
5	9.657	161.3

•

With estimates of the mean and variance of Y we can proceed to fit an additive linear regression to estimate k and the dispersion parameter for each imputed data set:

```
lm.fit.imputed <- list()
for (i in 1:5) {
  lm.fit.imputed[[i]] <- lm(log(varY) ~ log(meanY), data = grouped_imputatations[[i]])
}
pander(lm.fit.imputed)
```

Table 6: Fitting linear model: $\log(\text{varY}) \sim \log(\text{meanY})$

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.129	0.2441	8.721	0.003174
log(meanY)	1.404	0.1055	13.31	0.0009163

•

Table 7: Fitting linear model: $\log(\text{varY}) \sim \log(\text{meanY})$

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.133	0.2505	8.516	0.003402
log(meanY)	1.389	0.1076	12.9	0.001005

•

Table 8: Fitting linear model: $\log(\text{varY}) \sim \log(\text{meanY})$

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.975	0.2331	8.469	0.003456
log(meanY)	1.47	0.09743	15.09	0.0006316

•

Table 9: Fitting linear model: $\log(\text{varY}) \sim \log(\text{meanY})$

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.123	0.2815	7.543	0.004831
log(meanY)	1.388	0.1204	11.53	0.001402

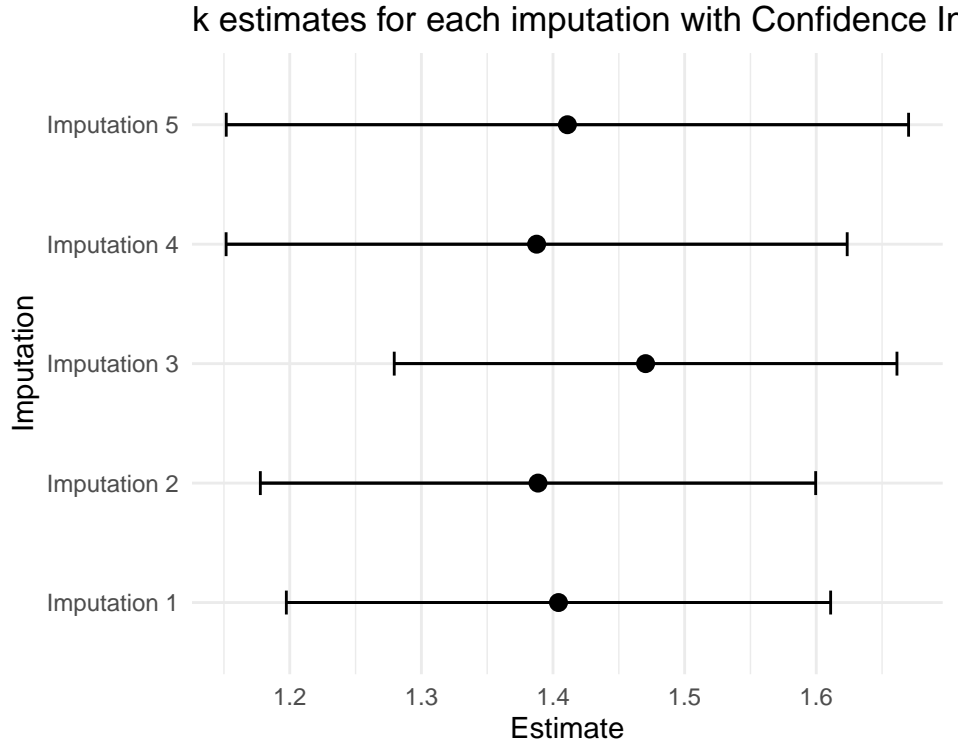
•

Table 10: Fitting linear model: $\log(\text{varY}) \sim \log(\text{meanY})$

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.096	0.3067	6.834	0.006412
log(meanY)	1.411	0.1322	10.67	0.00176

•

We get an initial estimate of k between 1.388 and 1.470 and ψ between $\exp(1.975) = 7.201$ and $\exp(2.133) = 8.440$. Naturally the estimates vary due to the randomness in the imputations. However, like the density plots the predictions are fairly similar. A simple confidence interval for the estimates of each imputation can be plotted:



As all values of k are within a reasonable margin of error, we choose to merge the imputations into a single data set to simplify calculations and communicate results more clearly. We should however keep in mind that this will reduce the variance of the data slightly and for a more thorough analysis we would keep all five imputed data sets.

We use the `merge_imputations` from the `sjmisc` package which merges multiple imputed data frames from `mice::mids()`-objects into a single data frame by computing the mean or selecting the most likely imputed value.

```

Rain.data.comp <- Rain.data %>%
  mutate(Rain = merge_imputations(Rain.data, Rain.data.impute)$Rain)

Rain.data.grp <- Rain.data.comp %>%
  group_by(Phase) %>%
  summarise(meanY = mean(Rain), varY = var(Rain))

lm.fit <- lm(log(varY) ~ log(meanY), data = Rain.data.grp)
pander(lm.fit)

```

Table 11: Fitting linear model: $\log(\text{varY}) \sim \log(\text{meanY})$

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	2.068	0.2647	7.81	0.00437
log(meanY)	1.4	0.1132	12.37	0.001139

We end up with estimates for k and ψ of

```

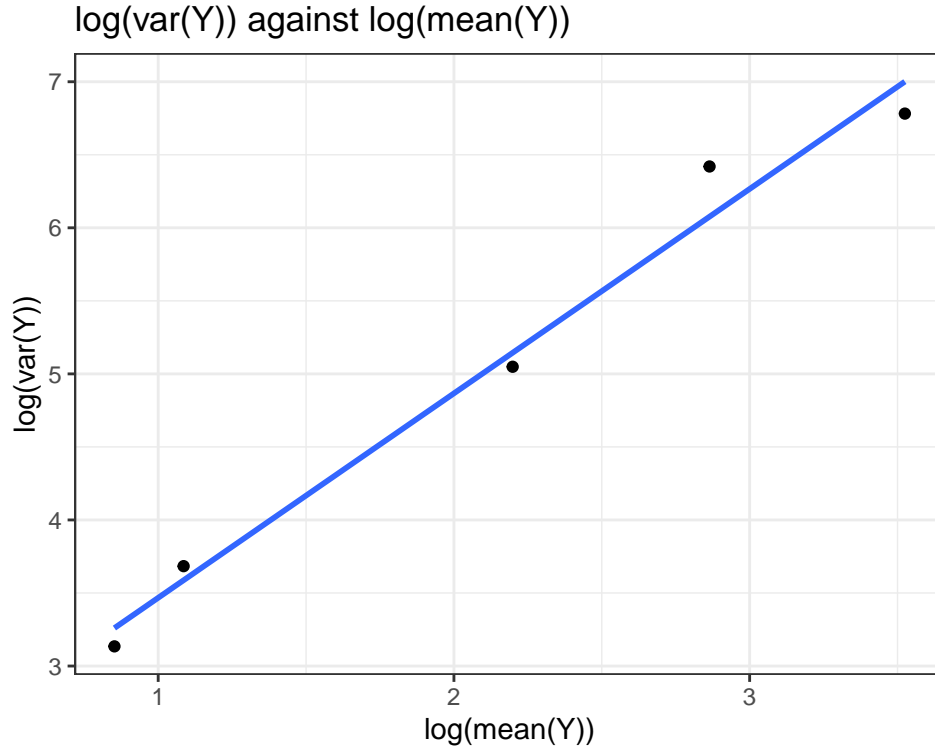
k_hat_lin <- lm.fit$coef[[2]]
psi_hat_lin <- exp(lm.fit$coef[[1]])

pander(c("Estimate of k" = k_hat_lin, "Estimate of psi" = psi_hat_lin))

```

Estimate of k	Estimate of psi
1.4	7.905

We can visually check that the estimated regression coefficients are reasonably estimated by plotting the fitted regression line on top of the data:



We see no warning signs from the plot and is fair to assume, that the value of k and ψ are descent estimates of the true values.

Fitting a Tweedie model to the data

We proceed to fit a Tweedie model to the data. We use the `tweedie()` family specification from the `tweedie` package to fit the model with the estimated value of k and `link.power = 0` for the log-link.

```
tweedie.fit <- glm(Rain ~ Phase, data = Rain.data.comp,
                  family = tweedie(var.power = k_hat_lin, link.power = 0))
pander(summary(tweedie.fit))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.086	0.4875	2.227	0.02786
Phase2	2.439	0.5251	4.645	9.114e-06
Phase3	-0.2339	0.8858	-0.264	0.7922
Phase4	1.778	0.5354	3.321	0.001201
Phase5	1.113	0.5449	2.042	0.04345

(Dispersion parameter for Tweedie family taken to be 8.212105)

Null deviance:	1260 on 119 degrees of freedom
Residual deviance:	910 on 115 degrees of freedom

To interpret the model output we recall the five SOI levels

- Phase 1: Consistently negative
- Phase 2: Consistently positive
- Phase 3: Rapidly falling

- Phase 4: Rapidly rising
- Phase 5: Consistently near zero

Note that Phase 1 is taken to be the reference phase. The model suggests that the rainfall for Phase 1 is significantly different from 0 with a point estimate of the average rainfall of

```
##          1
## 2.962222
```

millimeters in July. The model estimates that rainfall for Phase 2 is significantly different from the rainfall in Phase 1 with a point estimate of average rainfall of

```
##          1
## 33.95
```

millimeters in July when SOI is in this phase. The model estimates that the rainfall for Phase 3 is not significantly different from the rainfall in Phase 1 with a point estimate of rainfall of

```
##          1
## 2.344444
```

millimeters on average in July. The rainfall for Phase 4 is significantly different from the rainfall in Phase 1 according to the model with a point estimate of average rainfall of

```
##          1
## 17.538
```

millimeters in July. The model estimates rainfall for Phase 5 to be borderline significantly different from rainfall in Phase 1 with a point estimate of average rainfall of

```
##          1
## 9.012432
```

millimeters in July. Hence, the model suggests that Phase 2 and Phase 4 leads to significantly more rain on average than Phase 1. Phase 2 and 4 are when the SOI is consistently positive and rapidly rising respectively. A bit less rain seem to fall on average in Phase 5 when the SOI is consistently near zero, while the model predicts least rain on average in SOI phases 1 and 3 which are estimated not to be significantly different.

Note that the model predicts ψ to be

```
psi_hat_tweedie.fit <- summary(tweedie.fit)$dispersion
psi_hat_tweedie.fit
```

```
## [1] 8.212105
```

which is slightly different from the result obtained from the linear regression where ψ was estimated to be

```
psi_hat_lin
```

```
## [1] 7.905133
```

Estimating probability of zero rain in July

We use the estimates obtained in the previous exercises to estimate the probability that it will not rain in July. In the theoretical exercises we derived the probability of zero rain to be

$$\mathbb{P}(Y = 0) = \exp(-\lambda^*) = \exp\left(-\frac{\mu^{2-k}}{\psi(2-k)}\right)$$

With the two estimates of ψ from the previous exercise, we compute two estimates of the the probability that it will not rain in July. We plug in the estimated values of k and the empirical mean of our data:

```
mu_hat <- mean(Rain.data.comp$Rain)

exp(-mu_hat^(2 - k_hat_lin)/(psi_hat_lin*(2 - k_hat_lin)))
exp(-mu_hat^(2 - k_hat_lin)/(psi_hat_tweedie.fit*(2 - k_hat_lin)))
```

```
## [1] 0.3406069
## [1] 0.3545994
```

The two results 34.1 pct. and 35.5 pct. are quite similar and compared to the empirical probability of zero rain in July

```
sum(Rain.data.comp$Rain == 0)/nrow(Rain.data.comp)
```

```
## [1] 0.35
```

we obtain three estimates that are all very similar.

Determining k by minimizing AIC

We now estimate k by minimizing the Akaike Information Criterion (AIC) with a profile likelihood of a model with SOI phase as explanatory variable. That is, we search for the value of $k \in (1, 2)$ that minimizes the AIC. We start by constructing a general profile likelihood function that takes inputs: A formula, a family, a data set and an evaluation metric that we wish to optimize.

```
profile_likelihood <- function(form, family, data, eval) {
  model <- glm(form,
               family = family,
               data = data)
  eval_val <- eval(model)
  return(eval_val)
}
```

We define the specific profile likelihood that minimizes the AIC for different values of k of a Tweedie exponential model with rainfall as response and SOI phase as covariate. We specify `form = Rain ~ Phase`, `family = tweedie(var.power = k, link.power = 0)`, `data = Rain.data.comp` and `eval = AICtweedie`:

```
tweedie.AIC_profile_likelihood <- function(k) {
  profile_likelihood(form = Rain ~ Phase,
                    family = tweedie(var.power = k, link.power = 0),
                    data = Rain.data.comp,
                    eval = AICtweedie)
}
```

We use the `optimize` function to minimize the AIC and find the optimal value of k , where we search for k in the range (1.05, 1.95).

```
k_hat_AIC <- optimize(tweedie.AIC_profile_likelihood, lower = 1.05, upper = 1.95)$minimum
k_hat_AIC
```

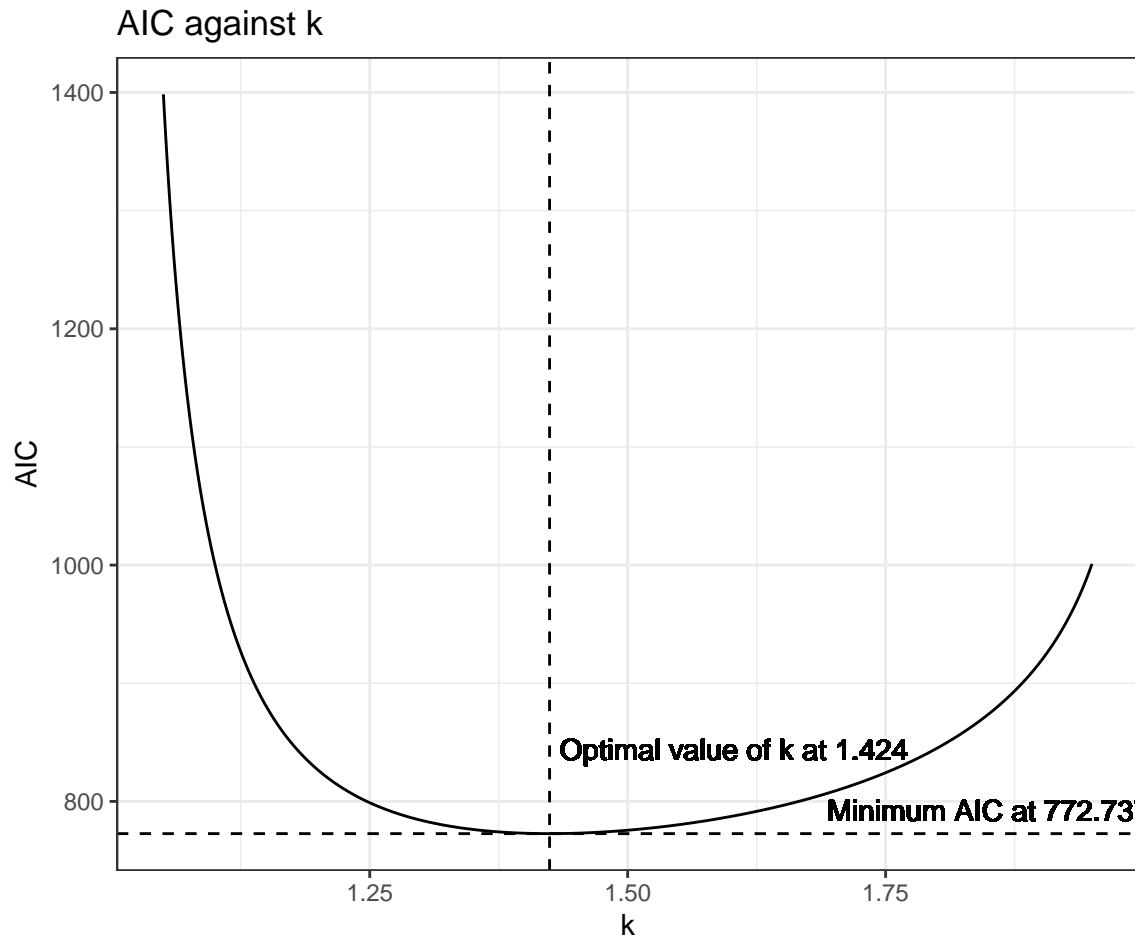
```
## [1] 1.42432
```

We note that the optimal value of k using the profile likelihood method is fairly close to the value of k estimated by the linear model

```
k_hat_lin
```

```
## [1] 1.399512
```

To ensure, that we have in fact found a minimum for $1 < k < 2$ we plot the AIC against k and add a vertical and a horizontal line at the optimal value of k .



The plot confirms, that `optimize()` has found the global minimum of the AIC for $1 < k < 2$. We repeat the calculations from the previous exercises using the optimal value of k found by minimizing the profile likelihood.

Re-estimating with new value of k

As before we initially fit a Tweedie model to the data using the optimal value of k found by minimizing the AIC.

```
tweedie.fit.AIC <- glm(Rain ~ Phase, data = Rain.data.comp,
                      family = tweedie(var.power = k_hat_AIC, link.power = 0))
pander(summary(tweedie.fit.AIC))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.086	0.4797	2.264	0.02547
Phase2	2.439	0.5189	4.7	7.267e-06
Phase3	-0.2339	0.8699	-0.2689	0.7885
Phase4	1.778	0.5289	3.363	0.001049
Phase5	1.113	0.5377	2.069	0.04076

(Dispersion parameter for Tweedie family taken to be 7.739979)

Null deviance: 1203.2 on 119 degrees of freedom

Residual deviance:	874.5 on 115 degrees of freedom
--------------------	---------------------------------

We note a slight decrease in the p-value for all the coefficients but otherwise the results are very similar to the previous model and the interpretation is the same. This is not surprising since the estimated value of k from the AIC profile likelihood is close to the value of k estimated by the linear model. The estimated dispersion parameter $\hat{\psi}_{k_{AIC}} = 7.740$ is slightly different from the previous Tweedie model estimate $\hat{\psi}_{k_{lin}} = 8.212$ and closer to the linear model estimate $\hat{\psi} = 7.905$.

We recalculate the estimated probability that it will not rain in July. We plug in the estimated values of k and ψ and the empirical mean of our data we get the following estimate of the probability that it will not rain:

```
psi_hat_model_AIC <- summary(tweedie.fit.AIC)$dispersion
exp(-mu_hat^(2 - k_hat_AIC)/(psi_hat_lin*(2 - k_hat_AIC)))
exp(-mu_hat^(2 - k_hat_AIC)/(psi_hat_model_AIC*(2 - k_hat_AIC)))

## [1] 0.3498665
## [1] 0.3421135
```

Again the estimates are similar to the estimates obtained from the previous estimate of k .

Model diagnostics

We check the model assumptions for the two Tweedie models fitted in the previous exercise. First we construct a data frame with the relevant diagnostic information for both Tweedie models. We extract the fitted values, Pearson residuals and deviance residuals for both models and add the SOI phase as a variable to the data frame.

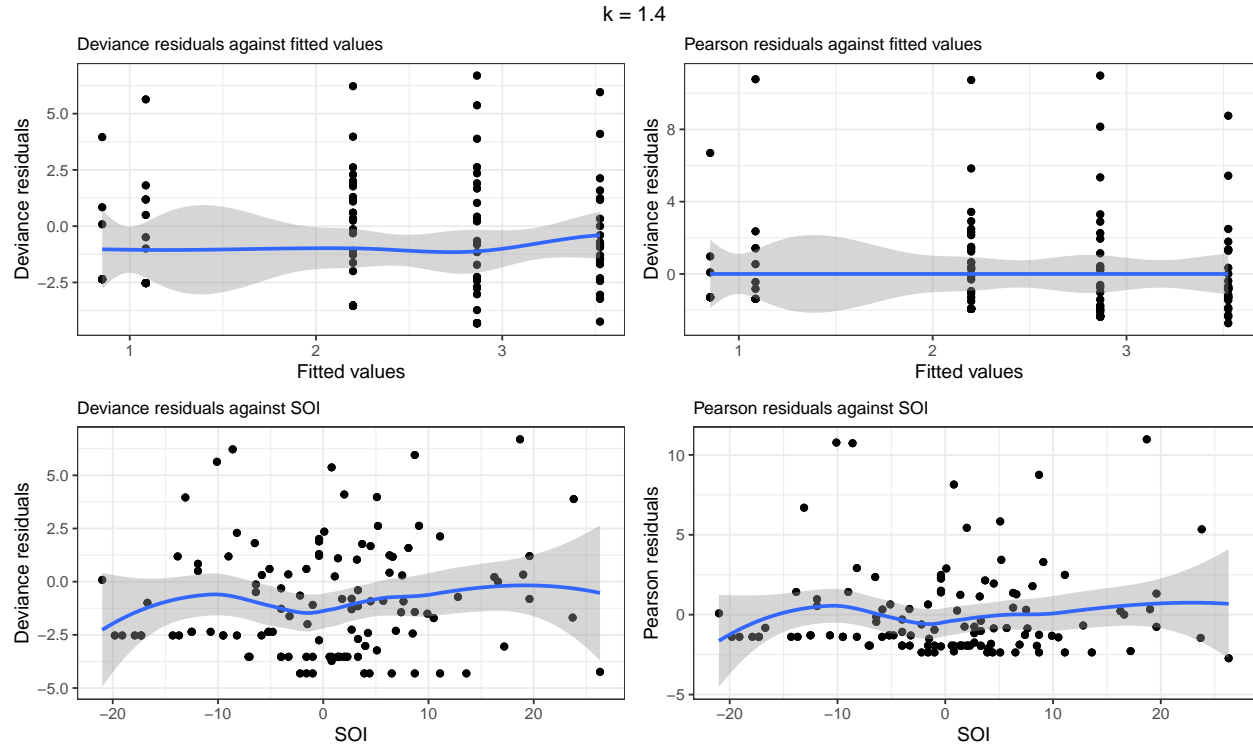
Table 17: Table continues below

linmod.fitted	linmod.pearson	linmod.deviance	AIC.fitted
Min. :0.852	Min. :-2.729	Min. :-4.3128	Min. :0.852
1st Qu.:2.199	1st Qu.: -1.935	1st Qu.: -2.8228	1st Qu.:2.199
Median :2.199	Median :-1.209	Median :-1.3611	Median :2.199
Mean :2.385	Mean : 0.000	Mean :-0.9051	Mean :2.385
3rd Qu.:2.864	3rd Qu.: 1.010	3rd Qu.: 0.8893	3rd Qu.:2.864
Max. :3.525	Max. :10.977	Max. : 6.6923	Max. :3.525

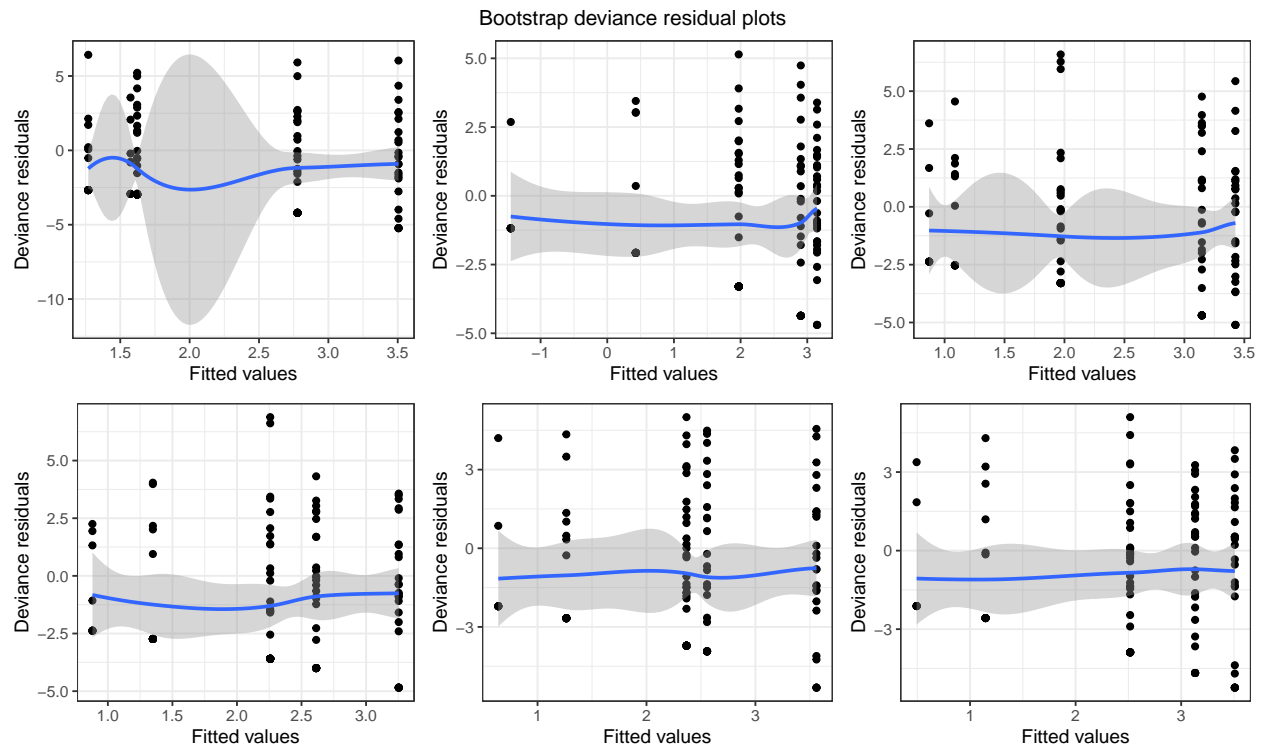
AIC.pearson	AIC.deviance	SOI
Min. :-2.6120	Min. :-4.2510	Min. :-21.0000
1st Qu.: -1.8830	1st Qu.: -2.7432	1st Qu.: -5.3575
Median :-1.1577	Median :-1.3066	Median : 0.8000
Mean : 0.0000	Mean :-0.9101	Mean : 0.4817
3rd Qu.: 0.9928	3rd Qu.: 0.8718	3rd Qu.: 5.8500
Max. :10.6310	Max. : 6.4126	Max. : 26.3000

We then plot the residuals against the fitted values first for the linear model estimate of k and then for the AIC estimate of k .

Linear model estimate of k

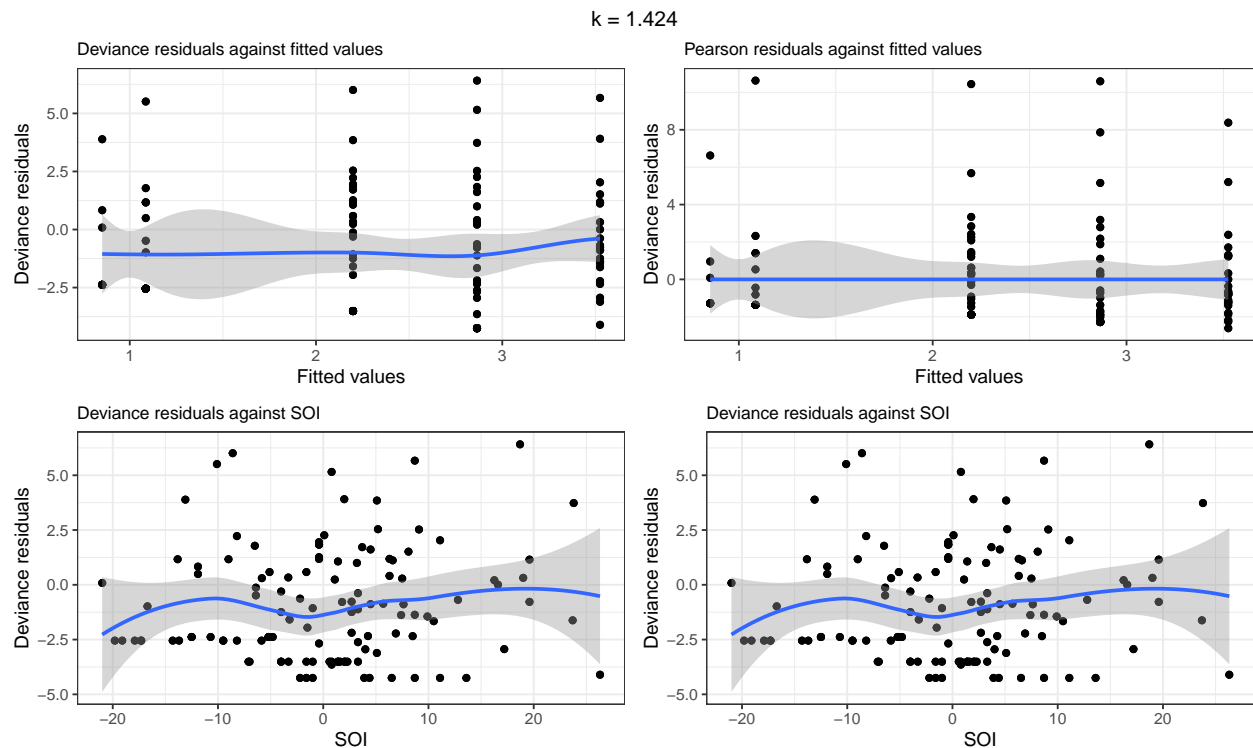


From the plots there is no clear indication that the model is misspecified. The residuals appear to be randomly scattered around zero which indicates that the model captures the mean and variance structure of the data. We further evaluate the plots with bootstrapping. In particular we simulate data from the fitted model. That is, assuming data is from a Tweedie distribution with the estimated mean, dispersion parameter and k we simulate new data and fit a Tweedie model to the new data. We then plot the residuals against the fitted values and the SOI phase for the new data. We hope to see, that the bootstrapped residuals are similar to the residuals from the original data.

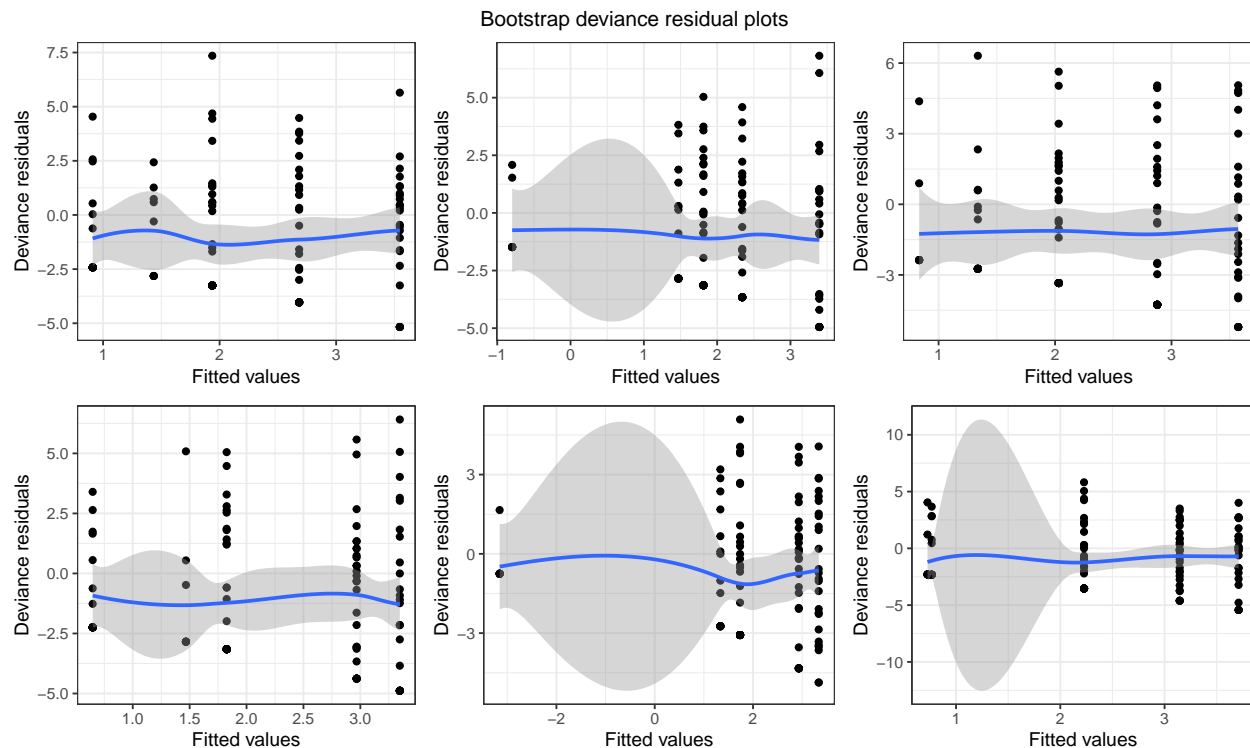


We see that the plots by and large resemble the residual plots of the original model. This supports the initial diagnostic plots. That is, there is no clear evidence that the model assumptions are violated.

AIC estimate of k



We notice a similar pattern to the previous model.

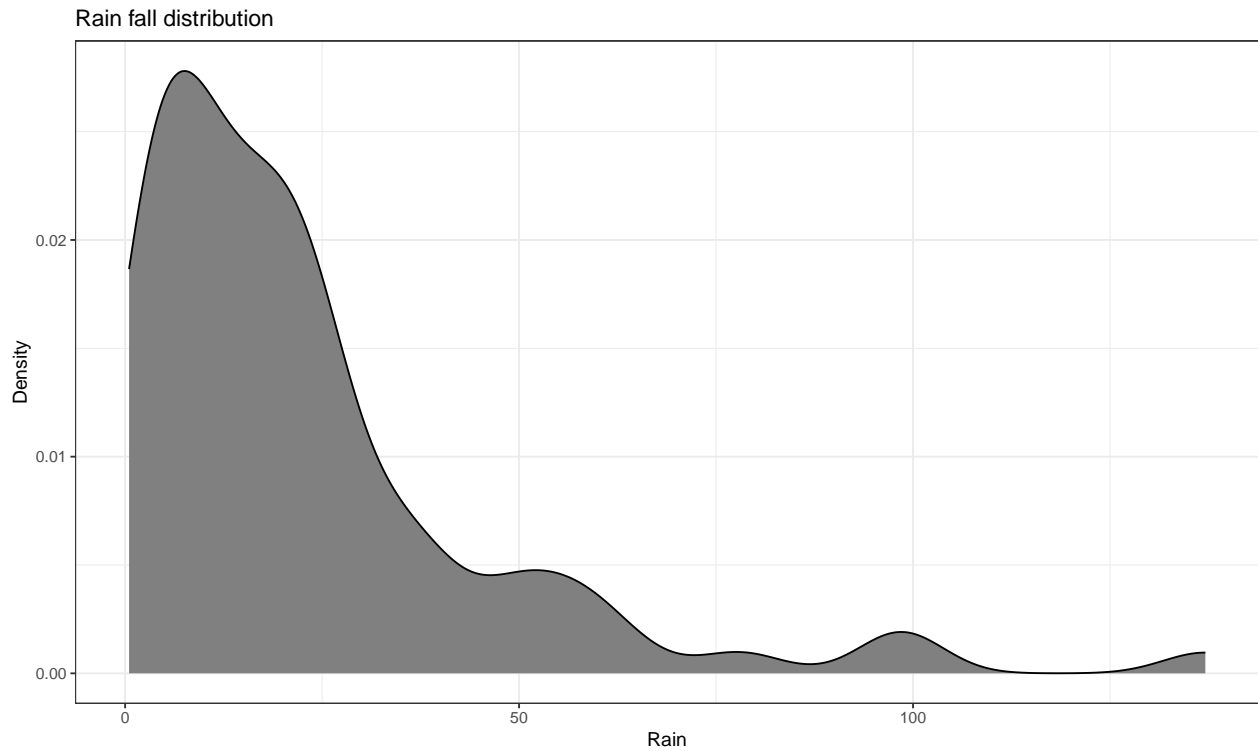


The pattern appears to be similar to the previous model. There may be some signs in the second and fifth plot that the model fitted on the simulated data diverge slightly from the original model. It is difficult to say if this is just noise or if the second estimate of k is worse than the first estimated k .

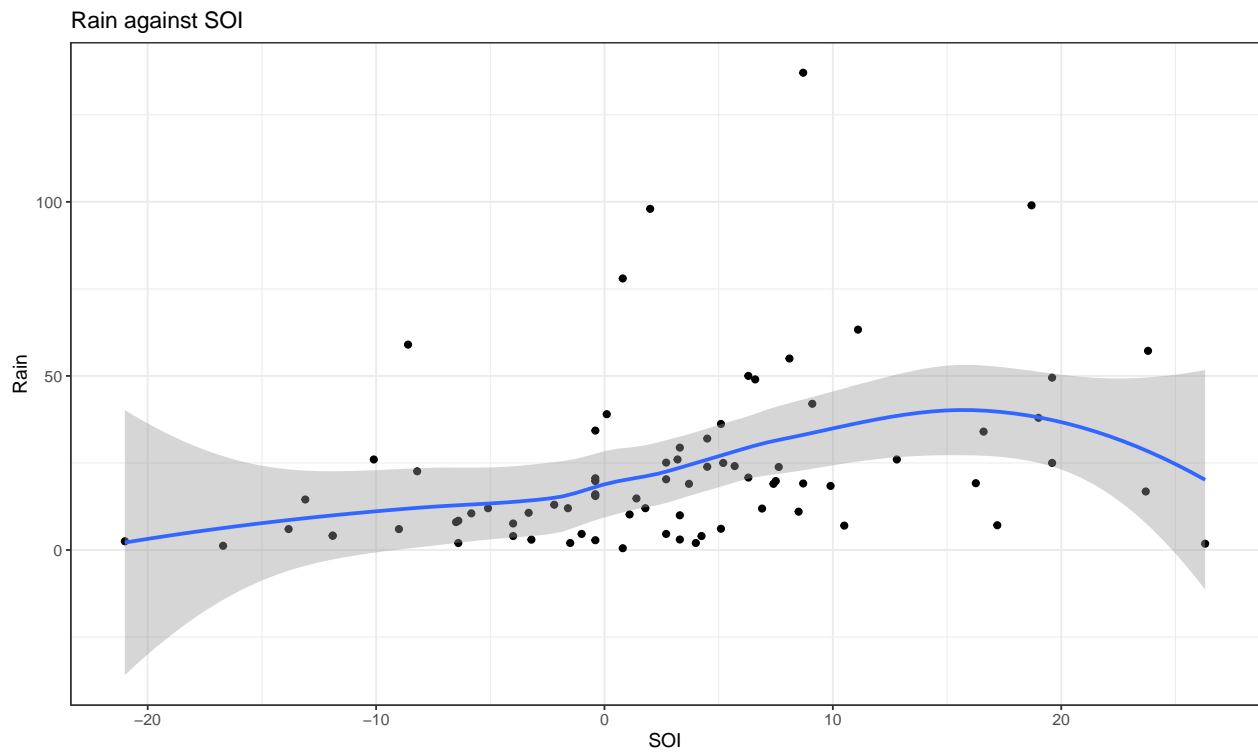
Analysis using SOI directly

We choose to model the rain fall conditionally on having rained. To do this we first filter out the observations where the rain fall is zero.

We look at the distribution of the rain fall.



The distribution is still very right skewed. This motivates the Gamma exponential dispersion model, which is typically used to fit positive continuous right skewed data. Consider now the rain fall as a function of the SOI.



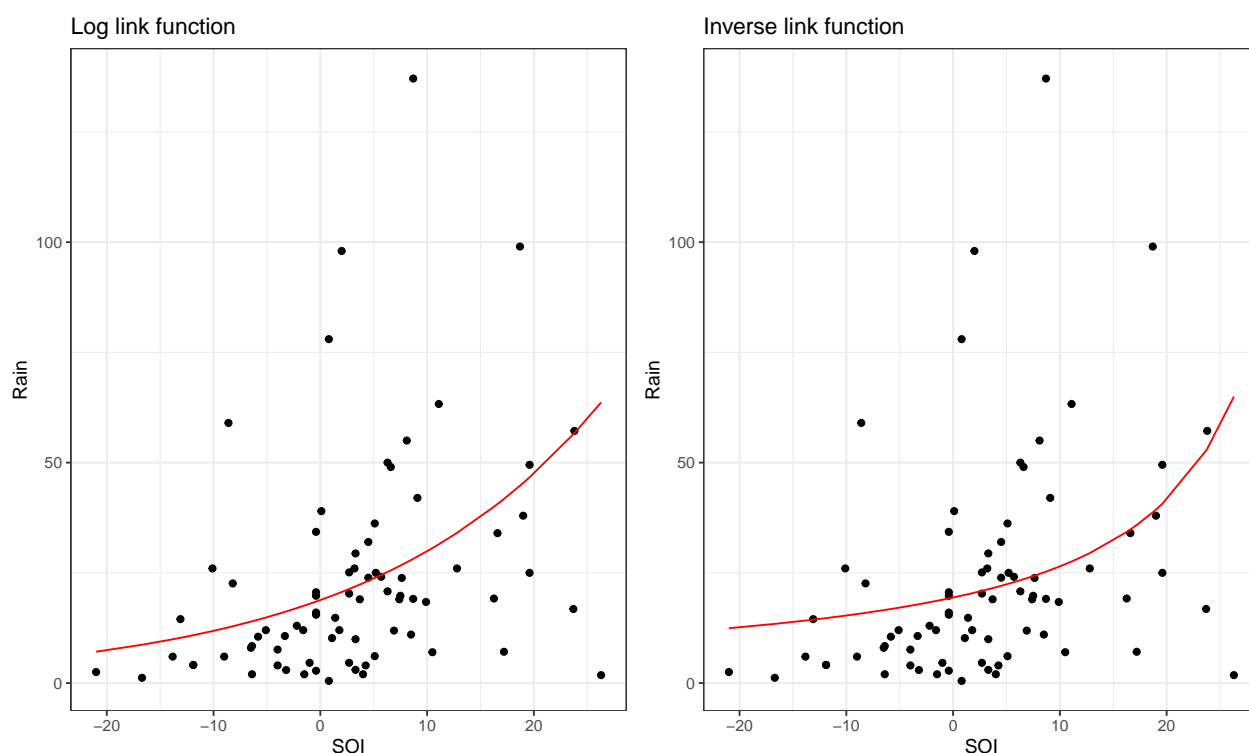
There seems to be a positive relationship between the SOI and the rain fall. The plot also indicates a possibility of non linear trends. The Gamma exponential dispersion model has a quadratic mean variance relationship, which also seems like a good fit from the plot above.

If we choose the log link function, we fit the log of the mean of the response variables as a linear combination of the predictors. Other options include the identity link and the canonical link, which is the inverse function. A problem we have encountered when using the identity link is that for some parameters the model produces negative predictions, which is not possible for the rain fall, and furthermore causes convergence issues. For this reason the identity link is disregarded. We fit two models: one with log link and one using the canonical link.

Below we calculate the training error based on the squared deviance loss function.

Link function	Training error
Log	0.8825
Inverse	0.9217

The log link function seems to be the best fit in terms of training error. We plot the model fits:



The Gamma model with log link function seems to fit data reasonably well. We choose to proceed with the Gamma model rather than the log link function.

Should we include additional predictors?

We perform an LRT test to see if we should include additional predictors. We consider to add the predictors Phase and Year. As mentioned in the EDA, Phase and SOI are very correlated, and it could be problematic to include both in the model.

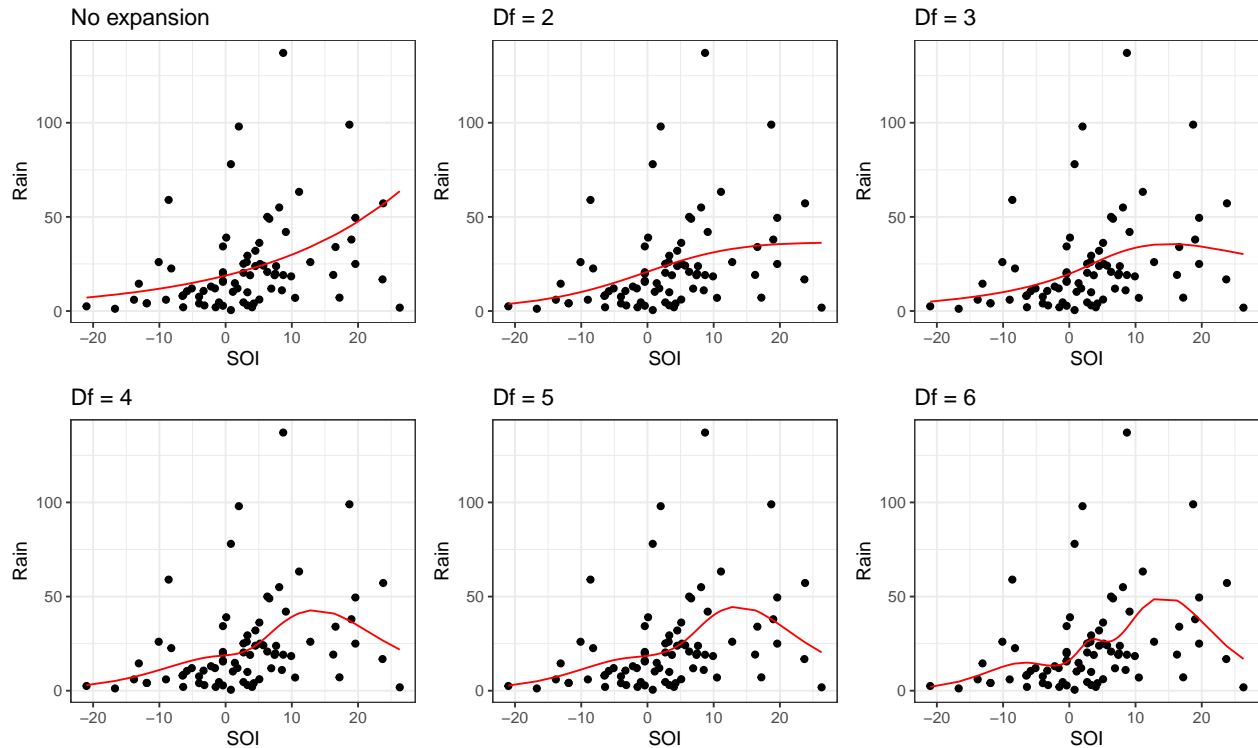
	Df	Deviance	AIC	scaled dev.	Pr(>Chi)
	NA	68.83829	639.2623	NA	NA
Phase	4	63.70767	642.2971	4.9651410	0.2908930
Year	1	68.22435	640.6681	0.5941376	0.4408236

According to the LRT test there is not evidence in data that suggest that the additional predictors should be added to the model.

Explore possible inclusion of nonlinear effects

We consider to include a non-linear effect of SOI. We consider a natural cubic spline with 2,3,4,5 or 6 degrees of freedom:

The model fits:



Adding more degrees of freedom to the natural cubic splines adds flexibility to the model, allowing it to fit data better. This comes at the expense of potentially overfitting. The model fitted with 6 degrees of freedom is quite likely overfitting data. But in order to better assess which model is the best in terms of prediction, we do cross-validation to compare the models. We first define the error function. We use the deviance loss function.

We define the cross validation function.

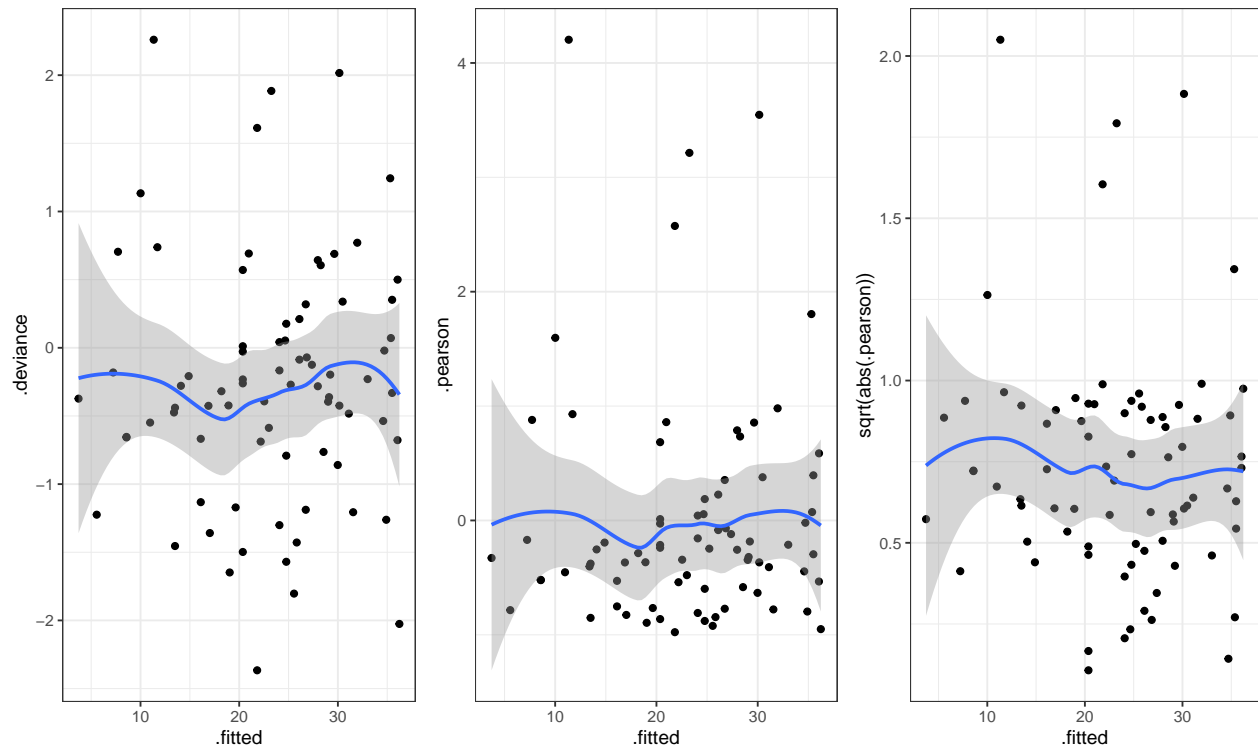
Since the data set is quite small, we perform LOOCV. This is a non random procedure and we therefore set $B = 1$.

```
## [1] 0.939251
## [1] 0.938202
## [1] 0.9728871
## [1] 0.9914927
## [1] 1.019917
## [1] 1.035805
```

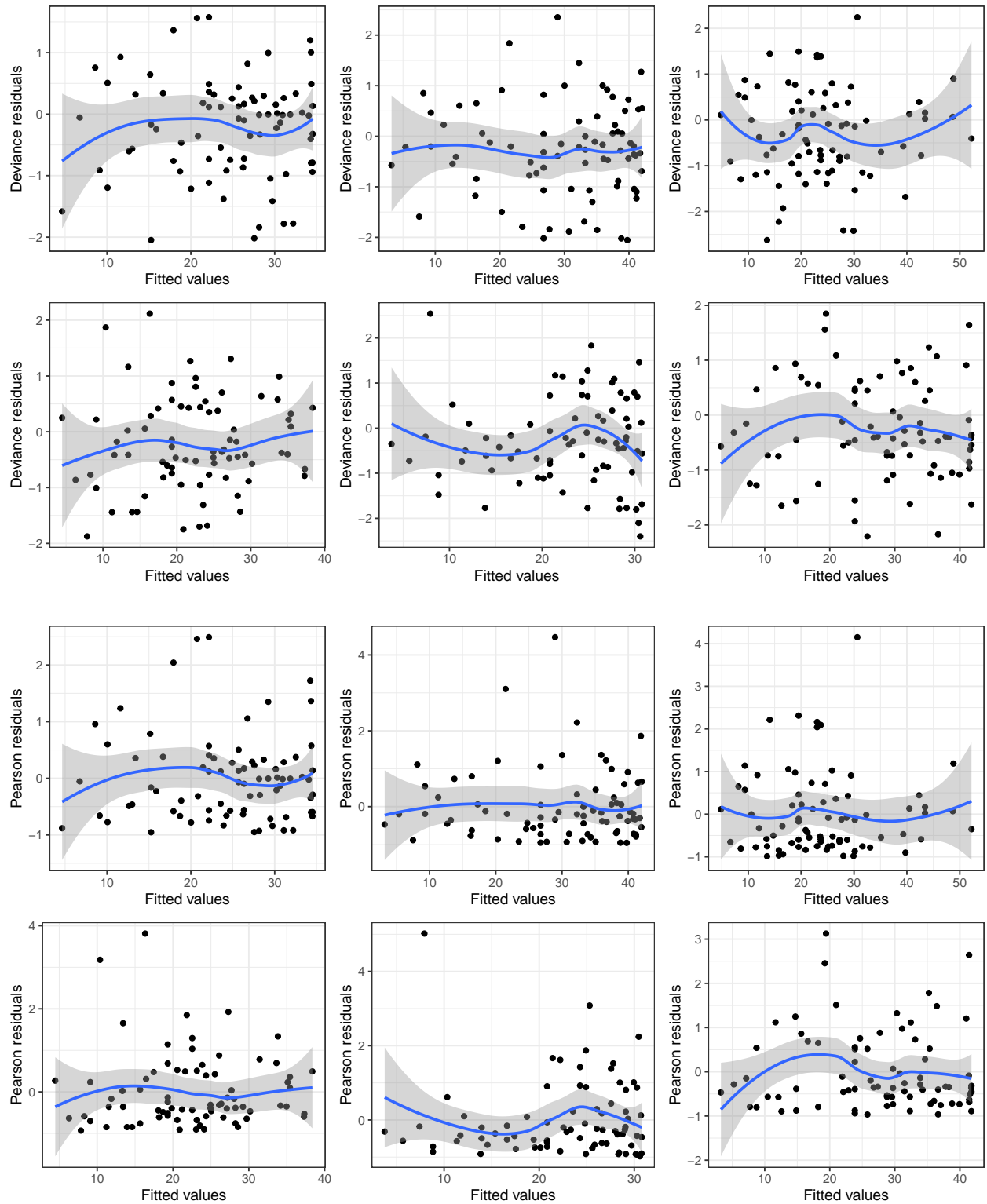
We proceed with the model with 2 degrees of freedom as it has the smallest cross validation error. But note that the model without natural cubic splines is very close to performing just as well in terms of generalization error. Since we are interested in prediction, we choose the model with 2 degrees of freedom. If interpretability was of higher priority we would choose the model without natural cubic splines.

Model Diagnostics

We do model diagnostics for the chosen model.



The model diagnostics look quite good. There is no clear evidence against the model assumptions. We evaluate these plots via bootstrapping. We compare the residuals with simulated residuals under the null hypothesis that our model is correct.



The bootstrapped residual plots resemble the original residual plots sufficiently well. This indicates that the fitted model is correct.

Reporting a final model and interpretation

The final model fit

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.314	0.5208	2.523	0.01374
ns(SOI, df = 2)1	3.775	1.04	3.629	0.0005163
ns(SOI, df = 2)2	1.263	0.5099	2.476	0.01554

(Dispersion parameter for Gamma family taken to be 1.014447)

Null deviance:	80.67 on 77 degrees of freedom
We see that SOI is a	significant predictor of rain fall. Since we have used a natural cubic spline with 2 degrees of freedom to fit our model, the coefficients are difficult to interpret. We instead consider the predictions of the model. Below the model predictions for a few values of SOI are printed

SOI	Rain Fall
-20	4.088

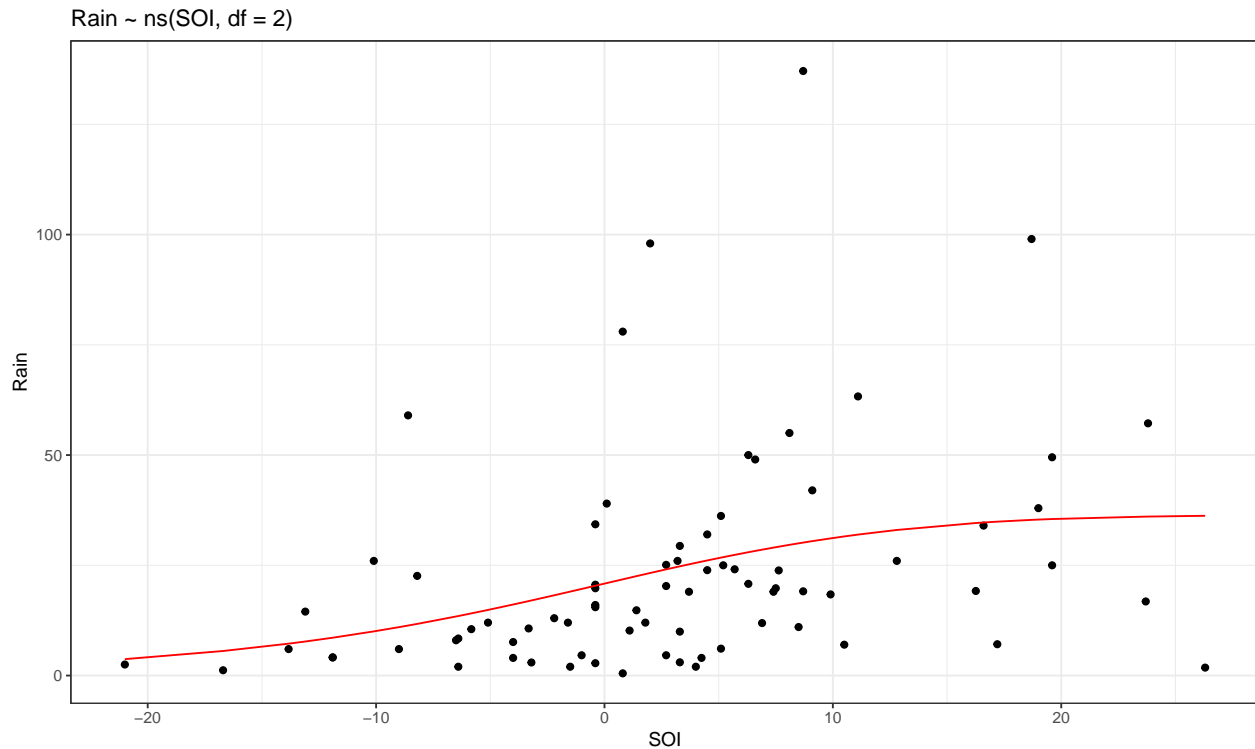
-10 10.1

0 20.85

10 31.18

20 35.57

A plot of the model fit is shown below.



The fitted model predicts that rainfall is increasing as a function of SOI. The slope of the model is largest for values of SOI between -10 and 10 . For SOI values that are larger or smaller than this, the model is more constant.

We will now turn to the construction of confidence intervals for our model. First we will use nonparametric bootstrap to create a combinant based confidence interval for the model predictions as described on page 220.

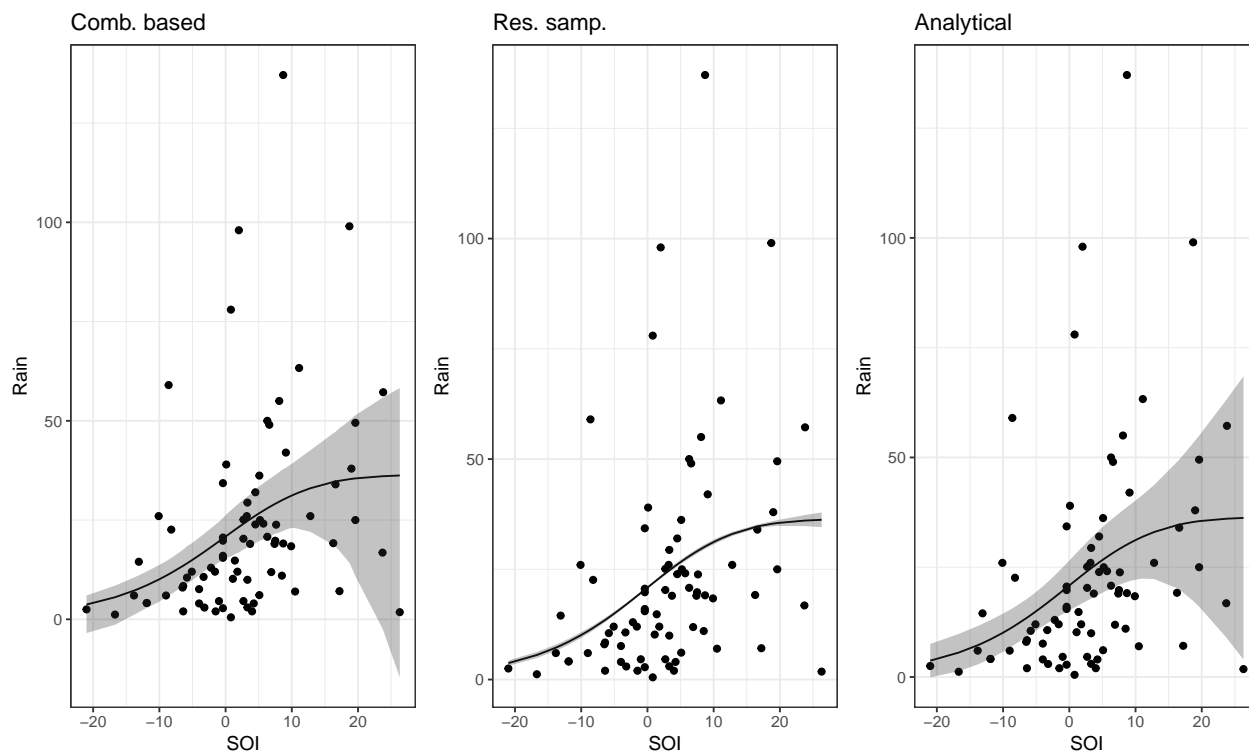
We will further construct confidence intervals of the form

$$f \pm 1.96\hat{se}$$

Note that these intervals will be symmetric around the point estimate. First we will use residual sampling to estimate standard errors of the model predictions and use these to construct the confidence interval.

A last confidence interval we will consider is created in the same way as above, but using analytical standard errors of the model predictions.

We will now compare the three confidence intervals.



The first thing that catches the eye is that the confidence interval based on residual sampling SE estimates is very very narrow, which seems very unlikely. Apart from that we see that the two other confidence interval are narrow for the SOI values where we have many observations and wide for the SOI values for which we have few observations. This is to be expected. A last thing to point out is the asymmetry that is present in the bootstrap combinant based confidence interval, indicating a certain asymmetry of the distribution of the model predictions.

Conclusion

Because of the right skew of data we decided to use the Gamma model to fit the data. We chose the log link as it ensures that the model predictions are kept within the domain of the distribution and since it performed well in terms of training error. We chose not to include further predictors as they were insignificant according to the LRT test.

The model we have fitted is a Gamma model with a log link function and a natural cubic spline with 2 degrees of freedom. This was the model with the smallest generalization error chosen by cross validation, where we considered models fitted on natural cubic splines with degrees of freedom ranging from 1 to 6.

The model is well fitted to the data and the residuals show no evidence against the model assumptions.

The model predicts that rain fall is increasing as a function of SOI, and that SOI is a significant predictor of rain fall. We have constructed confidence intervals for the model predictions, which show that the model is most certain for SOI values where we have many observations, and less certain for SOI values where we have few observations.