Modeling

Contents

1 Data gathering

1.1 Read climate anomaly data

This will be the dependent variable

```
df_temp_anomaly <- read.csv("data/NASA/global_temperature_anomaly.csv",
                            sep = ",", header = TRUE, skip = 2)

# Set the year as rownames and delete from first colum
year_names <- df_temp_anomaly[, 1]
df_temp_anomaly[, 1] <- NULL
rownames(df_temp_anomaly) <- year_names
```

1.2 Read possible drivers of climate change, i.e. independent variabels

Considered as independent variables different drivers such as greenhouse gas emissions, energy, transport, industrial processes, and waste

```
df_climate_raw <- read.csv("data/Eurostat/df_climate.csv", sep = ",", header = TRUE)

# Set the year as rownames and delete from first colum
year_names <- df_climate_raw [, 1]
df_climate_raw[, 1] <- NULL
rownames(df_climate_raw) <- year_names
```

2 Preparation of Dataset for the Model

Add climate anomaly data to drivers data, i.e. in one dataset both dependent and independent variables

```
df_climate_raw <- merge(df_climate_raw, df_temp_anomaly, by="row.names", all=TRUE)
names(df_climate_raw)[names(df_climate_raw) == 'Row.names'] <- 'Year'

# Do not consider lowess smothing (just for visualization purposes)
df_climate_raw$Lowess.5. <- NULL
```

2.1 Keep interesting variable

Realize that for some variable there are a lot of missing value. Only keep value from 1995 onwards. Data since 1995 has been considered because most of the dataset from Eurostat has 1995 as the first data collection date. Note that not all variables are starting from that date, as an example some variables start from 2000

```
# Keep value from 1995 onwards
mask <- df_climate_raw$Year > 1994
df_climate <- df_climate_raw[mask, ]

# Set year as rownames
# Set the year as rownames and delete from first colum
year_names <- df_climate[, 1]
df_climate[, 1] <- NULL
rownames(df_climate) <- year_names

# Look at two columns and see that they have different starting date
kable(df_climate[2:9, 1:2])
```

	sts_copr_a_PROD_F_CA_I10_EU28	sts_copr_a_PROD_F_CC1_CA_I10_EU28
1996	96.5	NA
1997	95.4	NA
1998	95.7	NA
1999	98.7	NA
2000	101.8	103.7
2001	102.9	104.4
2002	103.4	104.8
2003	105.2	107.0

2.2 Inptutation of time series missing data

In the presence of missing data, most statistical packages use listwise deletion, which removes any row that contains a missing value from the analysis.

```
# Imputation by Kalman Smoothing and State Space Models
df_inp <- data.frame(sapply(df_climate, function(x) na_kalman(x)))
```

2.3 Correlation analysis

Correlations analysis among the different variables was conducted. All the variables with correlations higher than 0.9 were rejected in order to avoid multicollinearity problems

```
df_cor <- cor(df_inp)
```

2.3.1 Eliminate highly correlated variable, i.e correlation higher than 0.9

```
# Set the upper triangle equal to zero
df_cor[upper.tri(df_cor)] <- 0
diag(df_cor) <- 0

data_no_corr <- df_inp[, !apply(df_cor, 2, function(x) any(abs(x) > 0.90, na.rm = TRUE))]
```

3 Modeling

3.1 Base model

```
lm_0 <- lm(No_Smoothing ~ ., data = data_no_corr)
#summary(lm_0)
#texreg(lm_0)
```

Dependent variable	Average annual yearly anomaly temperature *No_smoothing*
Number of observation	26
Type	OLS linear regression
Residual standard error:	0.0449 on 5 degrees of freedom
Multiple R^2	0.989
Adjusted R^2	0.9452
F-statistic	22.56 on 20 and 5 DF
F-statistic p-value	0.001329

	Estimate	Standard Error	Pr(>|t|)
(Intercept)	22.20	8.185	0.04214 *
sts_copr_a_PROD_F_CC1_CA_I10_EU28	$-5.543e-02$	$2.313e-02$	0.06187 .
sts_copr_a_PROD_F_CC2_CA_I10_EU28	$4.587e-02$	$2.812e-02$	0.16379
sts_inpr_a_PROD_C_CA_I10_EU28	$6.138e-02$	$2.190e-02$	0.03789 *
sts_inpr_a_PROD_D_CA_I10_EU28	$-6.904e-02$	$1.782e-02$	0.01170 *
env_wasmun_GEN_KG_HAB_EU28	$-6.972e-03$	$1.014e-02$	0.52215
env_wasmun_TRT_KG_HAB_EU28	$-3.643e-03$	$1.186e-02$	0.77109
tai08_CO2_PC_CRF3_EU28	$1.796e+00$	$1.118e+00$	0.16901
tran_hv_psmod_PC_BUS_TOT_EU28	$-6.441e-02$	$2.843e-01$	0.82975
tran_hv_psmod_PC_TRN_BUS_TOT_AVD_EU28	$-3.288e-01$	$2.370e-01$	0.22404
tran_hv_pstra_I10_EU28	$-3.241e-02$	$8.067e-02$	0.70451
ttr00005_TOT_LOADED_THS_T_EU28	$-5.819e-07$	$2.572e-07$	0.07314 .
t2020_rk200_TOTAL_KTOE_FC_OTH_HH_E_EU28	$2.564e-06$	$2.545e-06$	0.35983
ten00123_FC_E_C0000X0350.0370_KTOE_EU28	$-1.362e-04$	$6.230e-05$	0.08042 .
ten00123_FC_E_C0350.0370_KTOE_EU28	$6.108e-04$	$2.054e-04$	0.03102 *
ten00123_FC_E_E7000_KTOE_EU28	$-1.197e-04$	$4.119e-05$	0.03356 *
ten00123_FC_E_H8000_KTOE_EU28	$2.348e-04$	$5.517e-05$	0.00804 **
ten00123_FC_E_O4000XBIO_KTOE_EU28	$4.972e-05$	$2.732e-05$	0.12841
ten00123_FC_E_S2000_KTOE_EU28	$2.711e-03$	$4.253e-03$	0.55196
ten00123_FC_E_TOTAL_KTOE_EU28	$-1.374e-06$	$4.934e-06$	0.79180
ten00123_FC_E_W6100_6220_KTOE_EU28	$2.168e-04$	$1.997e-04$	0.32715

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$; $^{\cdot}p < 0.1$

3.1.1 Base model info and fit

3.1.2 Base model coefficients

3.2 Diagnostics of base model

3.2.1 Multicollinearity (vif should be <10)

```
kable(vif(lm_0), col.names = c("VIF"))
```

	VIF
sts_copr_a_PROD_F_CC1_CA_I10_EU28	294.69597
sts_copr_a_PROD_F_CC2_CA_I10_EU28	175.23350
sts_inpr_a_PROD_C_CA_I10_EU28	405.76854
sts_inpr_a_PROD_D_CA_I10_EU28	117.06705

	VIF
env__wasmun__GEN__KG__HAB__EU28	343.06186
env__wasmun__TRT__KG__HAB__EU28	319.40429
tai08__CO2__PC__CRF3__EU28	23.80831
tran__hv__psmod__PC__BUS__TOT__EU28	377.59264
tran__hv__psmod__PC__TRN__BUS__TOT__AVD__EU28	126.15657
tran__hv__pstra__I10__EU28	2093.51451
ttr00005__TOT__LOADED__THS__T__EU28	186.04308
t2020__rk200__TOTAL__KTOE__FC__OTH__HH__E__EU28	12.39700
ten00123__FC__E__C0000X0350.0370__KTOE__EU28	222.41453
ten00123__FC__E__C0350.0370__KTOE__EU28	55.56415
ten00123__FC__E__E7000__KTOE__EU28	182.27139
ten00123__FC__E__H8000__KTOE__EU28	60.47841
ten00123__FC__E__O4000XBIO__KTOE__EU28	2830.04005
ten00123__FC__E__S2000__KTOE__EU28	36.18083
ten00123__FC__E__TOTAL__KTOE__EU28	71.89523
ten00123__FC__E__W6100__6220__KTOE__EU28	427.26112

Since each value is higher than 10 there is a multicollinearity issue, meaning that significant test for coefficient would be off.

3.2.2 Normality of residuals

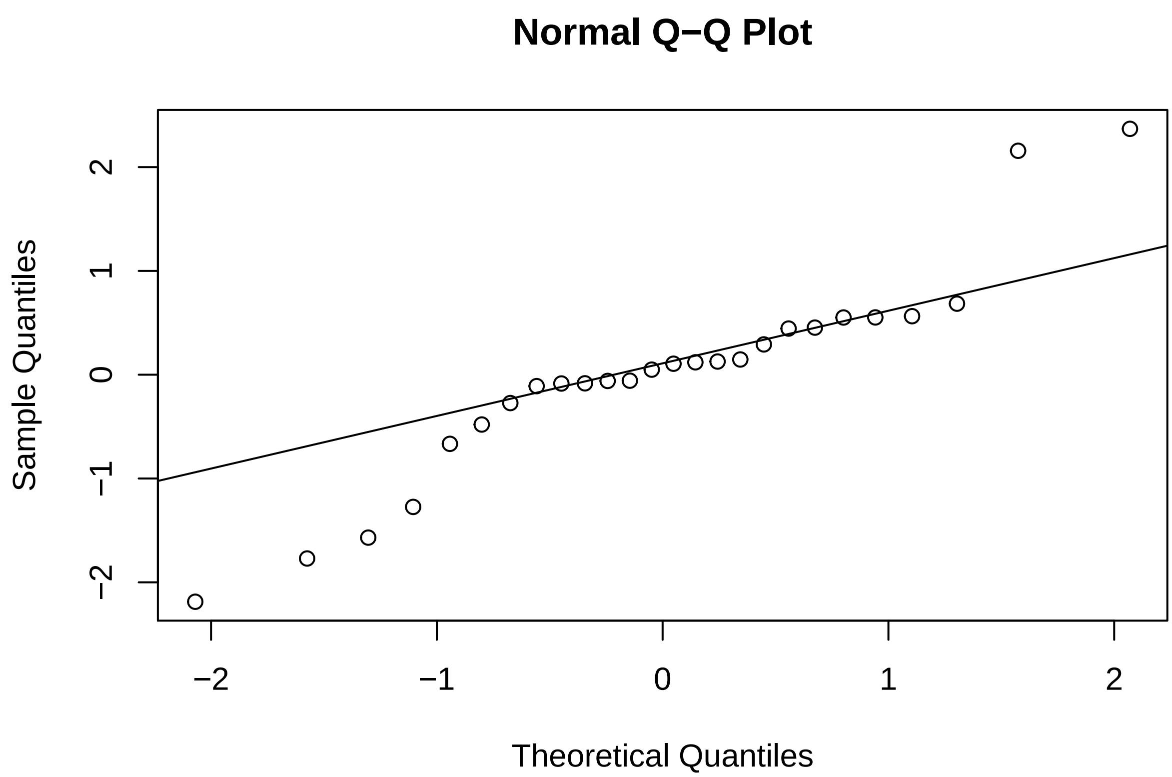
```
#shapiro.test(lm_0$residuals)
```

Shapiro-Wilk Normality Test
data: base model residuals

W	0.91189
p-value	0.02913

The residuals are not normally distributed since p-value is lower than 0.05

```
qqnorm(scale(lm_0$residuals))
qqline(scale(lm_0$residuals))
```



As it is possible to see from the graph theoretical quanties and sample quanties do not match.

3.2.3 Autocorrelation

```
#durbinWatsonTest(lm_0)
```

Durbin-Watson Test
Alternative hypothesis: rho != 0

log	Autocorrelation	D-W Statistic	p-value
1	−0.4822647	2.937878	0.482

No autocorrelation since p-value > 0.05, significant test would not be impacted as we suspect the variance in error term will be lower

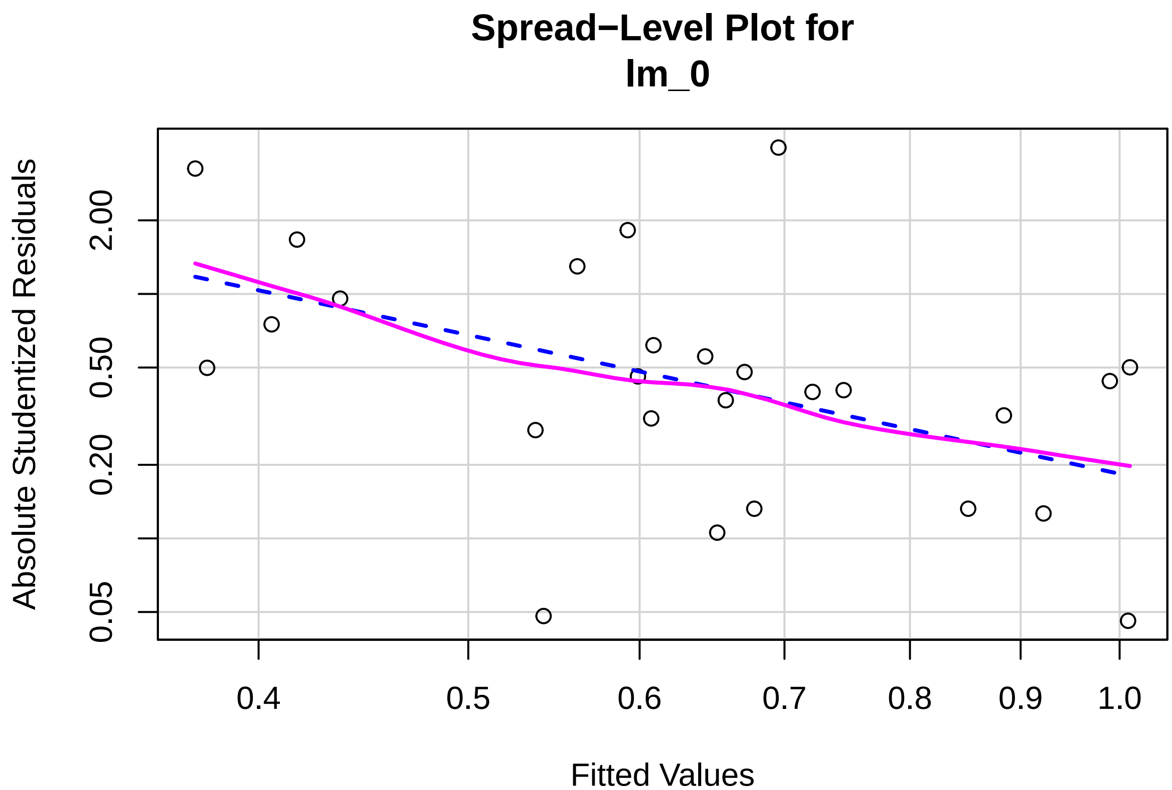
3.2.4 Heteroskedasticity - Homoskedasticity - Or in Lay Statistician's Terms: Non-Constant Variance

```
#ncvTest(lm_0)
```

Non-constant Variance Score Test
Variance formula: fitted values

Chisquare	7.343481
Df	1
p-value	0.02913

```
spreadLevelPlot(lm_0)
```




```
##
## Suggested power transformation:  2.885878
```

The model is characterized by heteroskedasticity, meaning it suffers from non constant variance and that the model is more reliable for certain values of estimated values (where variance is smaller) and less reliable for other values.

4 Model improvement

4.1 More imputation of missing data

```r
# Keep value from 1985 onwards
mask <- df_climate_raw$Year > 1984
df_climate <- df_climate_raw[mask, ]

# Set year as rownames
# Set the year as rownames and delete from first colum
year_names <- df_climate[, 1]
df_climate[, 1] <- NULL
rownames(df_climate) <- year_names
```

4.1.1 Inptutation of time series missing data

In the presence of missing data, most statistical packages use listwise deletion, which removes any row that contains a missing value from the analysis.

```r
# Imputation by Kalman Smoothing and State Space Models
df_inp <- data.frame(sapply(df_climate, function(x) na_kalman(x)))
```

4.1.2 Check correlation

```r
df_cor <- cor(df_inp)
```

4.1.3 Eliminate highly correlated variable, i.e correlation higher than 0.9

```r
# Set the upper triangle equal to zero
df_cor[upper.tri(df_cor)] <- 0
diag(df_cor) <- 0

data_no_corr <- df_inp[, !apply(df_cor, 2, function(x) any(abs(x) > 0.90, na.rm = TRUE))]
```

4.1.4 Run linear regression with more data

```r
lm_1 <- lm(No_Smoothing ~ ., data = data_no_corr)
#summary(lm_1)
#texreg(lm_1)
```

4.1.5 More sample size model info and fit

4.1.6 Diagnostic

```r
vif(lm_1)
```

4.1.6.1 Multicollinearity (vif should be <10)

Dependent variable	Average annual yearly anomaly temperature $No\_smoothing$
Number of observation	36
Type	OLS linear regression (More sample size)
Residual standard error:	0.08998 on 20 degrees of freedom
Multiple R^2	0.9196
Adjusted R^2	0.8593
F-statistic	15.25 on 15 and 20 DF
F-statistic p-value	$8.686e-08$

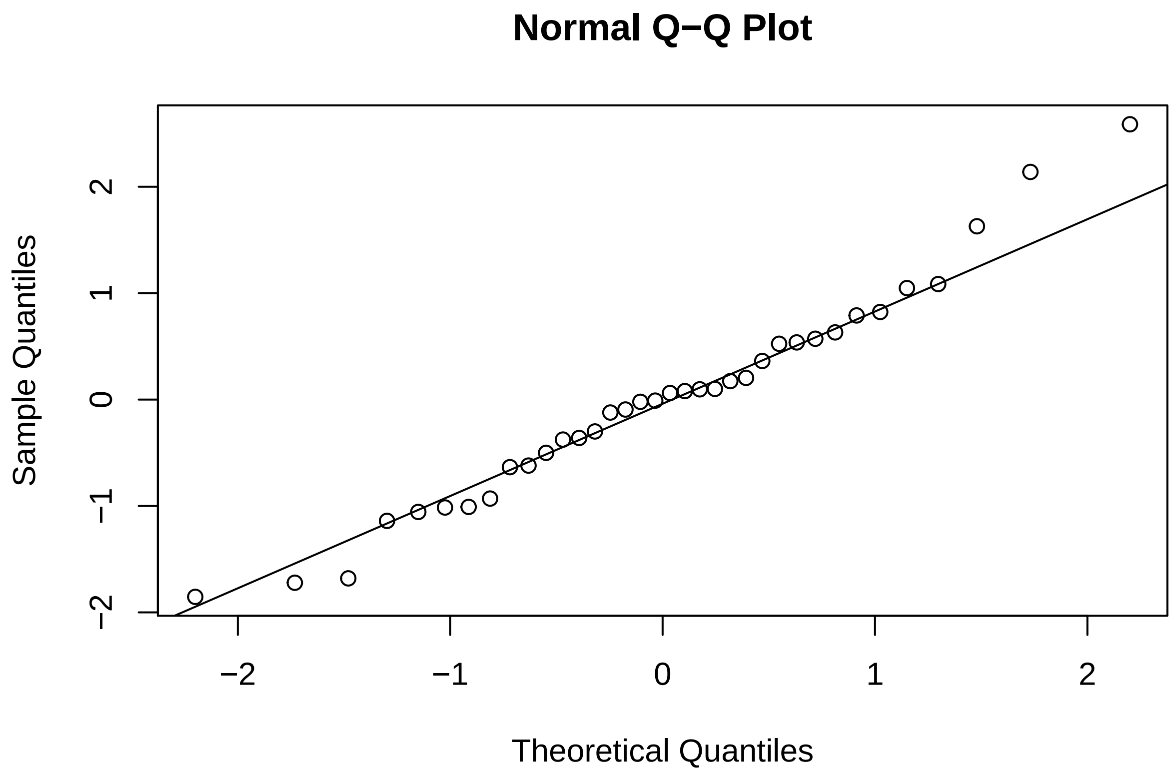
```
##       sts_copr_a_PROD_F_CC1_CA_I10_EU28       sts_copr_a_PROD_F_CC2_CA_I10_EU28
##                               11.220591                               14.293511
##           sts_inpr_a_PROD_D_CA_I10_EU28              env_wasmun_TRT_KG_HAB_EU28
##                               50.449515                                9.918700
##                  tai08_CO2_PC_CRF3_EU28   tran_hv_psmod_PC_TRN_BUS_TOT_AVD_EU28
##                               13.118913                                7.108433
##          ttr00005_TOT_LOADED_THS_T_EU28 t2020_rk200_TOTAL_KTOE_FC_OTH_HH_E_EU28
##                               30.420687                                7.109971
## ten00123_FC_E_C0000X0350.0370_KTOE_EU28      ten00123_FC_E_C0350.0370_KTOE_EU28
##                               32.236755                               25.834684
##           ten00123_FC_E_E7000_KTOE_EU28           ten00123_FC_E_H8000_KTOE_EU28
##                               45.414450                               28.769224
##           ten00123_FC_E_S2000_KTOE_EU28           ten00123_FC_E_TOTAL_KTOE_EU28
##                                4.883624                               44.589318
##      ten00123_FC_E_W6100_6220_KTOE_EU28
##                               51.390183
```

```
shapiro.test(lm_1$residuals)
```

4.1.6.2 Normality of residuals

```
##
##  Shapiro-Wilk normality test
##
## data:  lm_1$residuals
## W = 0.97472, p-value = 0.5676
```

```
qqnorm(scale(lm_1$residuals))
qqline(scale(lm_1$residuals))
```




```
durbinWatsonTest(lm_1)
```

4.1.6.3 Autocorrelation

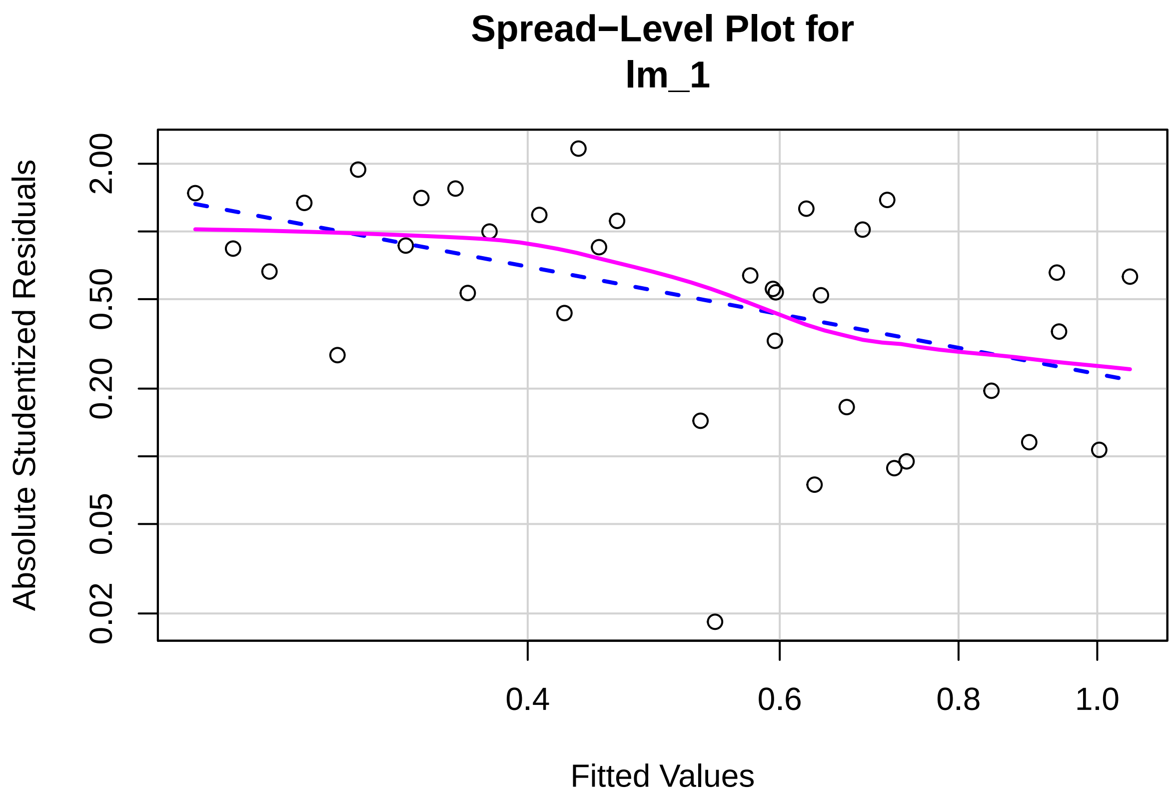
```
##  lag Autocorrelation D-W Statistic p-value
##    1      0.03742465      1.837309   0.068
##  Alternative hypothesis: rho != 0
```

```
ncvTest(lm_1)
```

4.1.6.4 Heteroskedasticity - Homoskedasticity - Or in Lay Statistician's Terms: Non-Constant Variance

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 10.19554, Df = 1, p = 0.0014078
```

```
spreadLevelPlot(lm_1)
```




```
##
## Suggested power transformation:  2.199082
```

4.2 Standardization

Apply standardization of data to see if could lead to improvement of model performance. Since lm_1, better than lm_0. Use dataset from lm_1 and standardize. Check correlation after standardization

```
# Standardization, df_inp is the dataframe after the inputation of missing data
df_norm <- data.frame(scale(df_inp))

## Check correlation
df_cor <- cor(df_inp)
```

4.2.1 Eliminate highly correlated variable, i.e correlation higher than 0.9

```
# Set the upper triangle equal to zero
df_cor[upper.tri(df_cor)] <- 0
diag(df_cor) <- 0

data_no_corr <- df_norm[, !apply(df_cor, 2, function(x) any(abs(x) > 0.90, na.rm = TRUE))]
```

```r
# Run the model
lm_2 <- lm(No_Smoothing ~ ., data = data_no_corr)
#texreg(lm_2)
#summary(lm_2)
```

4.2.2 Standardize model info and fit

Dependent variable	Average annual yearly anomaly temperature $No\_smoothing$
Number of observation	36
Type	OLS linear regression (Standardize)
Residual standard error:	0.3751 on 20 degrees of freedom
Multiple R^2	0.9196
Adjusted R^2	0.8593
F-statistic	15.25 on 15 and 20 DF
F-statistic p-value	$8.686e-08$

4.3 Stepwise regression

```r
# Define intercept-only mode, df_inp is the dataframe after the inputation of missing data
intercept_only <- lm(No_Smoothing ~ 1, data = df_inp)

# Define model with all predictors
all <- lm(No_Smoothing ~ ., data = df_inp)

# Perform forward stepwise regression
lm_3 <- step(intercept_only, direction='forward', scope=formula(all), trace=0)

# View results of forward stepwise regression
#lm_3$anova

# View final model
#lm_3$coefficients

# View model result
#texreg(lm_3)
#summary(lm_3)
```

4.3.1 Stepwise model info and fit

4.3.2 Stepwise regression model coefficients

4.4 Diagnostic of stepwise regression

4.4.1 Multicollinearity (vif should be <10)

```r
#vif(lm_3)
```

Dependent variable	Average annual yearly anomaly temperature *No_smoothing*
Number of observation	36
Type	OLS linear regression (Stepwise regression)
Residual standard error:	0.08111 on 33 degrees of freedom
Multiple R^2	0.8922
Adjusted R^2	0.8857
F-statistic	136.6 on 2 and 33 DF
F-statistic p-value	$2.2e-16$

	Estimate	Standard Error	Pr(>|t|)
(Intercept)	-0.727173	0.145886	$1.93e-05$ ***
sts_inpr_a_PROD_C10_CA_I10_EU28	0.016324	0.001253	$1.46e-14$ ***
ten00123_FC_E_S2000_KTOE_EU28	-0.005231	0.001384	0.000628 ***

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$; $^{\cdot}p < 0.1$

The model is not experiencing multicollinearity issues, since the VIF value of the variables is significantly lower than 10. This means there is no sizable correlations between multiple variables within the model.

4.4.2 Normality of residuals

```
#shapiro.test(lm_3$residuals)
```

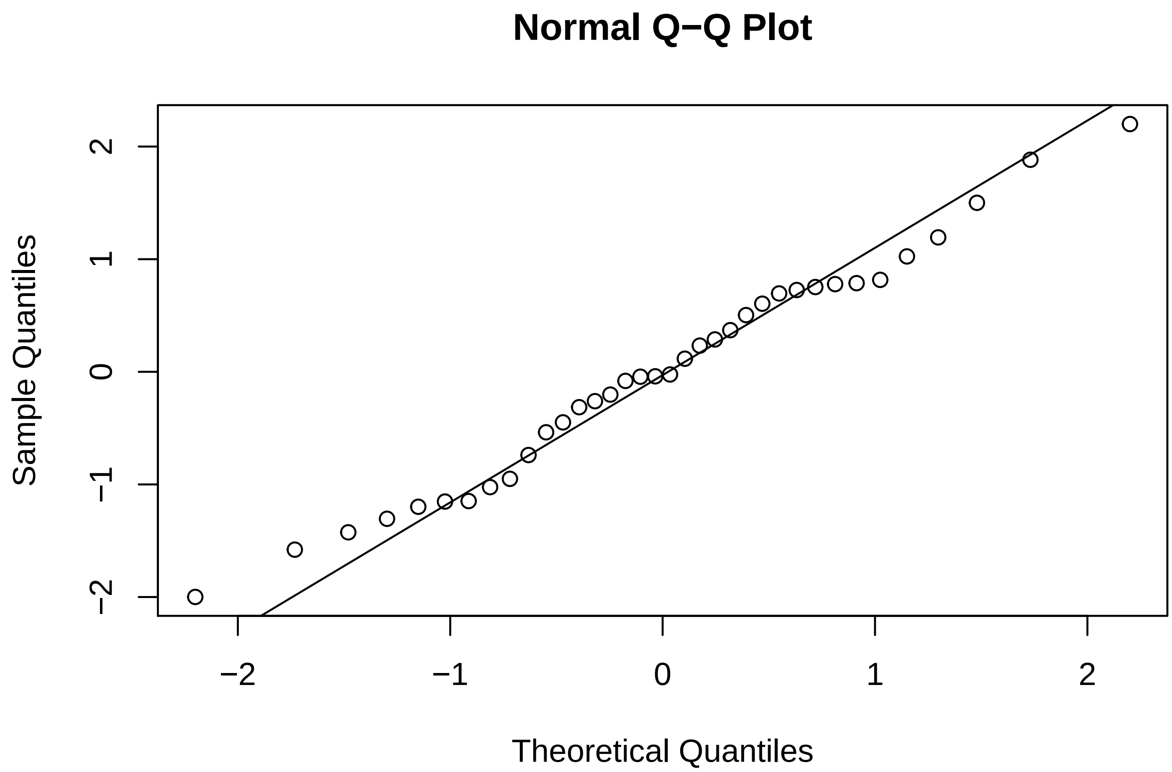
Given that the p-value=0.87 is considerably high (higher than 0.05) the residuals are normally distributed.

```
qqnorm(scale(lm_3$residuals))
qqline(scale(lm_3$residuals))
```

	VIF
sts_inpr_a_PROD_C10_CA_I10_EU28	1.212378
ten00123_FC_E_S2000_KTOE_EU28	1.212378

Shapiro-Wilk Normality Test
data: Stepwise regression residuals

W	0.98429
p-value	0.8779



4.4.3 Autocorrelation

```
#durbinWatsonTest(lm_3)
```

Durbin-Watson Test
Alternative hypothesis: rho != 0

log	Autocorrelation	D-W Statistic	p-value
1	0.1164049	1.639169	0.118

Since the p-value is higher than 0.05 there is no suspect of autocorrelation.

4.4.4 Heteroskedasticity - Homoskedasticity - Or in Lay Statistician's Terms: Non-Constant Variance

```
ncvTest(lm_3)
```

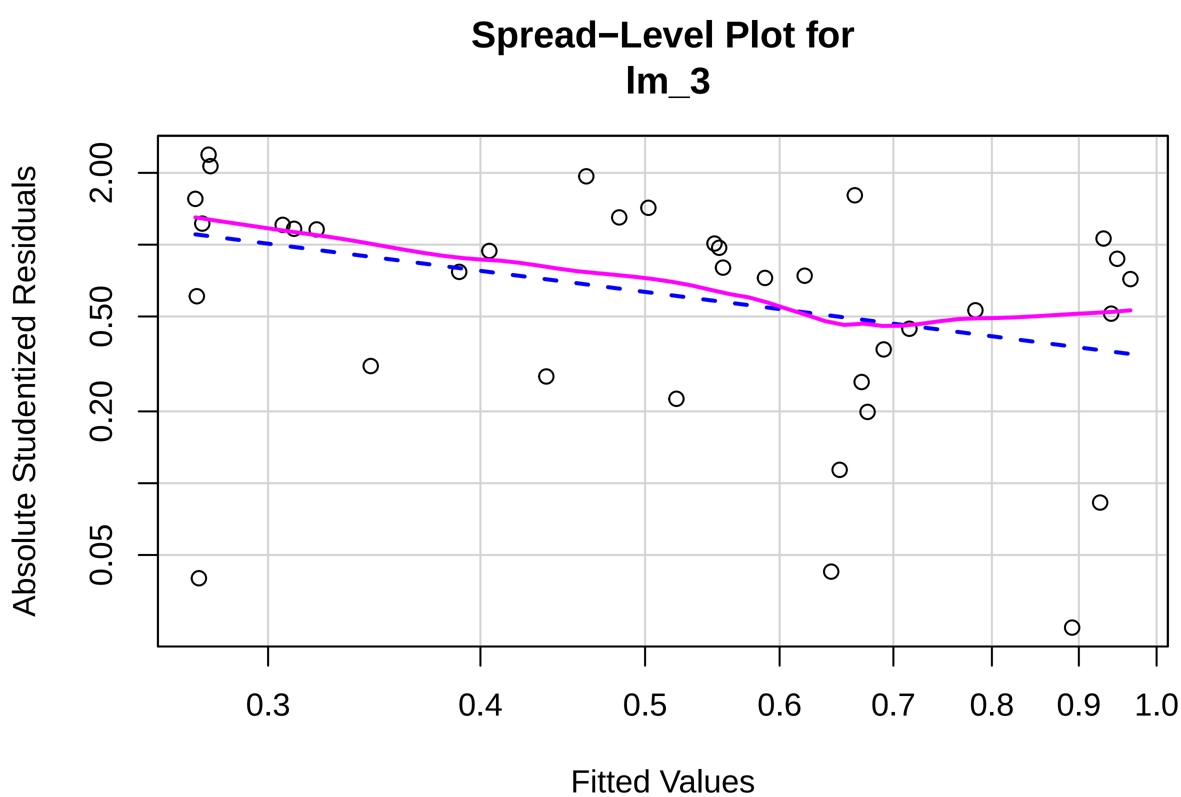
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 5.036168, Df = 1, p = 0.024823
```

Non-constant Variance Score Test
Variance formula: fitted values

Chisquare	5.036168
Df	1
p-value	0.024823

Since the p-value is high (higher than 0.05), the null hypothesis of homoscedasticity is not rejected. This means that the model does not suffer of non-constant variance.

```
spreadLevelPlot(lm_3)
```




```
##
## Suggested power transformation:  1.911458
```

5 Model comparison

Relative model performance metrics, such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), are used to compare time series models. Those metrics are the best approach when dealing with small data and time series. When data is recent and splitting into train, test and validation is not the optimal way to compare models (https://towardsdatascience.com/introduction-to-aic-akaike-information-criterion-9c9ba1c96ced).

5.1 AIC

```
models <- list(lm_3, lm_1, lm_2)
models_names <- c("stepwise_regression", "more_sample_size", "standardize")

AICs <- aictab(cand.set = models, modnames = models_names)
AICs
```

```
##
## Model selection based on AICc:
##
##                      K   AICc Delta_AICc AICcWt Cum.Wt    LL
## stepwise_regression  4 -72.54       0.00      1      1 40.92
## more_sample_size    17 -24.39      48.15      0      1 46.19
## standardize         17  78.40     150.94      0      1 -5.20
```

5.2 Using BIC

```
bic_1 <- BIC(lm_1)
bic_2 <- BIC(lm_2)
bic_3 <- BIC(lm_3)

BICs <- c(bic_3, bic_1, bic_2)
```

5.3 Nice table for model comparison

```
df_model_comparison <- data.frame(models_names, AICs$K, AICs$AICc, BICs)
colnames(df_model_comparison) <- c("Model", "# Parameters", "AIC", "BIC")

kable(df_model_comparison)
```

Model	# Parameters	AIC	BIC
stepwise_regression	4	-72.53989	-67.49614
more_sample_size	17	-24.38827	-31.46844
standardize	17	78.39691	71.31674