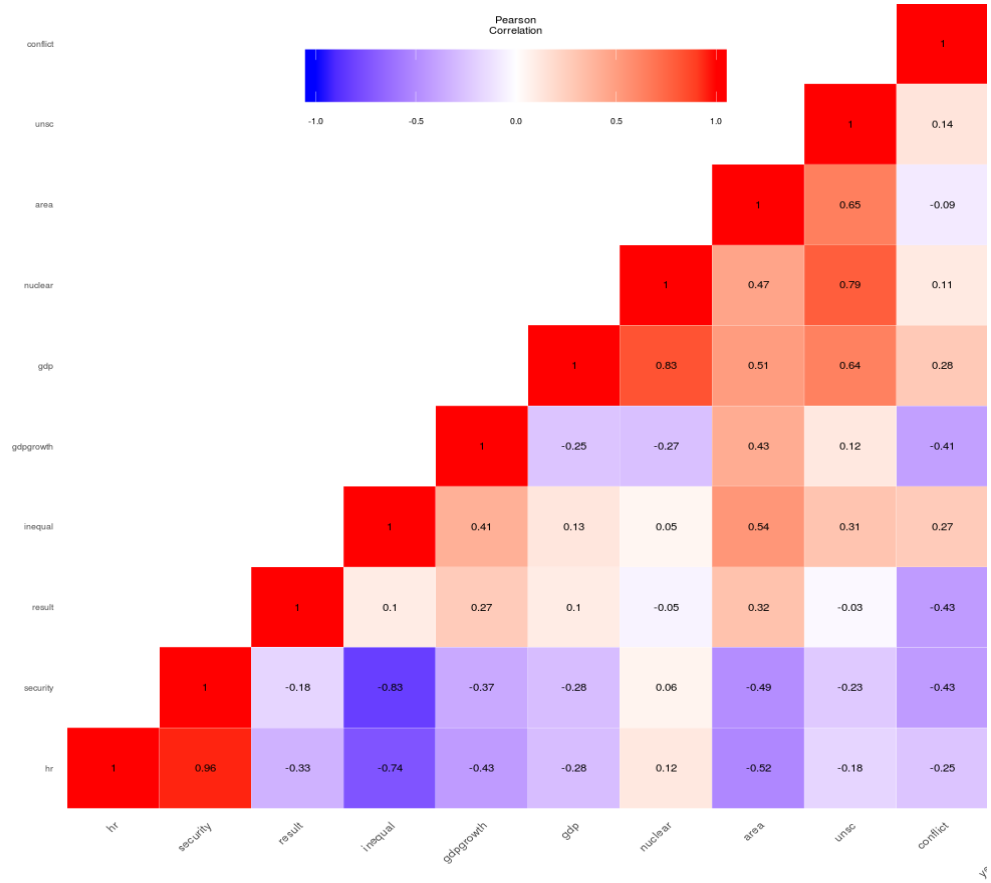
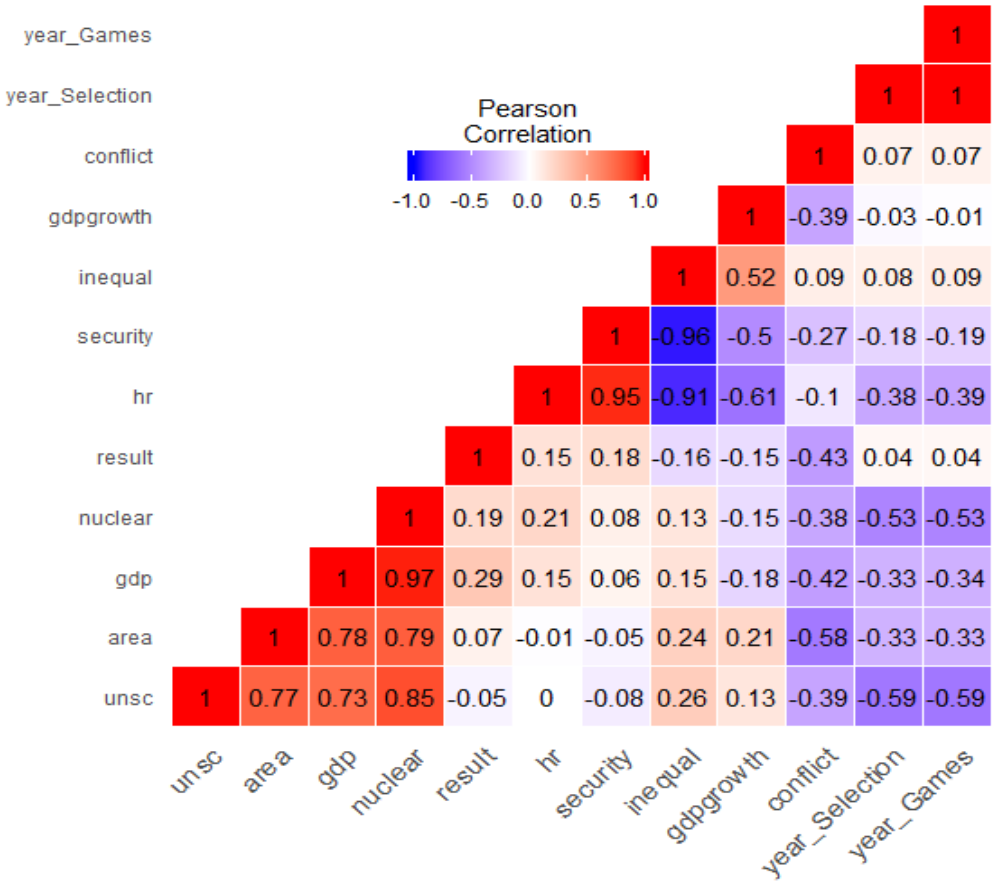
List of Appendices

Appendices

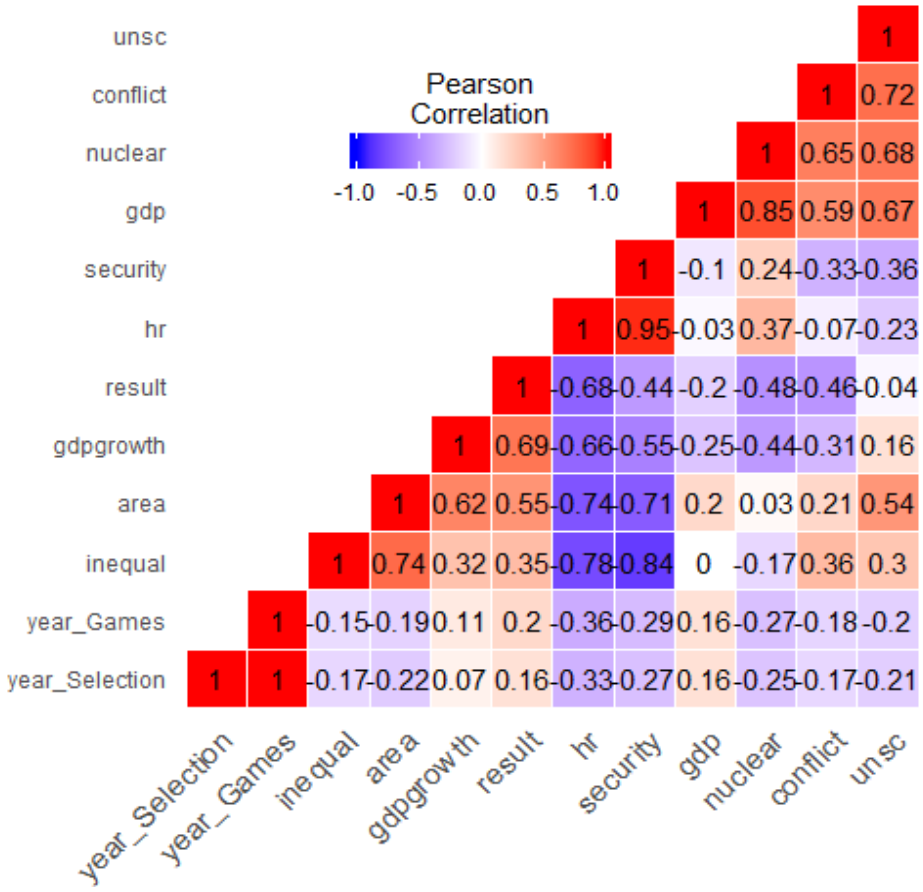
Appendix 1- Pearson correlation heatmap- model 1992-2022



Appendix 2- Pearson correlation heatmap- model 1992-2006



Appendix 3- Pearson correlation heatmap- model 2008-2022



Appendix 4 – R transcript

Model 1992-2022

```
# Adresar pro nacteni dat
dir='C:/Users/user/Desktop/regrese'
setwd(dir)
# Soubor pro nacteni
#file = 'dataset_1992-2022.csv'
#file = 'dataset_1992-2006.csv'
#file = 'dataset_2008-2022.csv'
file = 'dataset_1992-2022.csv'
data = read.table(file, header = TRUE ,sep = ";", dec=",")
# Zobrazeni dat
#show(data)
#show(data$candidate)
#summary(data)
# ------------------------------------------------------------------ #
# odmitnuti/neodmitnuti nulove hypotezy
shapr_conflict = shapiro.test(data$conflict)
shapr_hr = shapiro.test(data$hr)
shapr_inq = shapiro.test(data$inequal)
shapr_area = shapiro.test(data$area)
shapr_gdp = shapiro.test(data$gdp)
shapr_security = shapiro.test(data$security)
```

```
shapr_unsc = shapiro.test(data$unsc)
shapr_gdpg = shapiro.test(data$gdpgrowth)
shapr_nuclear = shapiro.test(data$nuclear)
# conflict
shapr_conflict
# hr
shapr_hr
#inq
shapr_inq
# area
shapr_area
# gdp
shapr_gdp
# security
shapr_security
# unsc
shapr_unsc
# gdpg
shapr_gdpg
# nuclear
shapr_nuclear
# ---------------------------------------------------------------------- #
# normalizace dat
normalize = function(x) {return ((x - min(x)) / (max(x) - min(x)))}
dataN = as.data.frame(lapply(data, normalize))              # celeho datove vstupu
# One could also use sequence such as df[1:2]
# dfNorm <- as.data.frame(lapply(data[4], normalize))          # pouze urciteho sloupce
# ---------------------------------------------------------------------- #
# pro urceni struktury vstupnich dat
# graf Q-Q norm
png(filename="C:/Users/user/Desktop/regrese/QQ_conflict.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$conflict, pch = 1, frame = FALSE)
qqline(data$conflict, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
png(filename="C:/Users/user/Desktop/regrese/QQ_hr.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$hr, pch = 1, frame = FALSE)
qqline(data$hr, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
png(filename="C:/Users/user/Desktop/regrese/QQ_inequal.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$inequal, pch = 1, frame = FALSE)
qqline(data$inequal, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
png(filename="C:/Users/user/Desktop/regrese/QQ_area.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$area, pch = 1, frame = FALSE)
qqline(data$area, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
png(filename="C:/Users/user/Desktop/regrese/QQ_gdp.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$gdp, pch = 1, frame = FALSE)
qqline(data$gdp, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
```

```
png(filename="C:/Users/user/Desktop/regrese/QQ_security.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$security, pch = 1, frame = FALSE)
qqline(data$security, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
png(filename="C:/Users/user/Desktop/regrese/QQ_nuclear.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$nuclear, pch = 1, frame = FALSE)
qqline(data$nuclear, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
png(filename="C:/Users/user/Desktop/regrese/QQ_gdpgrowth.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$gdpgrowth, pch = 1, frame = FALSE)
qqline(data$gdpgrowth, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem
png(filename="C:/Users/user/Desktop/regrese/QQ_unsc.png", width = 1920, height = 1200,
pointsize=30)
qqnorm(data$unsc, pch = 1, frame = FALSE)
qqline(data$unsc, col = "steelblue", lwd = 2)
dev.off() # zavreni souboru s grafem

#COL = [candidate conflict hr inequal area gdp securityunsc gdpgrowth unsc
          result  year_Selection  year_Games]
# ------------------------------------------------------------------- #
# Pearsonovu korelacni koeficient
#a = cor.test(data$gdp,data$hr,method="pearson")
#a$estimate
# ------------------------------------------------------------------- #
correl_mat = cor(data, method = "pearson")
correl_mat
cormat = cor(data, method = "pearson")
cormat
cormat = round(cor(cormat),2)
cormat
# Get lower triangle of the correlation matrix
get_lower_tri<-function(cormat){
  cormat[upper.tri(cormat)] <- NA
  return(cormat)
}
# Get upper triangle of the correlation matrix
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}
upper_tri <- get_upper_tri(cormat)
upper_tri
# Melt the correlation matrix
library(reshape2)
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Heatmap
library(ggplot2)
ggplot(data = melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
```

```
                    name="Pearson\nCorrelation") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()
reorder_cormat <- function(cormat){
  # Use correlation between variables as distance
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
}
# Reorder the correlation matrix
cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()

# Print the heatmap
print(ggheatmap)


ggheatmap +
  geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
  theme(
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    panel.grid.major = element_blank(),
    panel.border = element_blank(),
    panel.background = element_blank(),
    axis.ticks = element_blank(),
    legend.justification = c(1, 0),
    legend.position = c(0.6, 0.7),
    legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                               title.position = "top", title.hjust = 0.5))

# MODEL regrese
# vsechny parametry
model_1 = glm(result ~ conflict + inequal + gdpgrowth + security + gdp + nuclear ,
family=binomial("logit"),data=dataN)
model_1
summary(model_1)

#kontrolni model
model_c1 = glm(result ~  conflict + gdpgrowth + gdp, family=binomial("logit"),data=dataN)
```

```
model_c1
summary(model_c1)

#likelihood ratio test
library(car)
Anova(model_1, test=c("LR"))
Anova(model_c1, test=c("LR"))
anova(model_1,model_c1,model_1c2,test=c("LR"))
AIC(model_1,model_c1,model_1c2, k = 2)

#log-transformed model
exp(cbind(coef(model_1),confint(model_1)))
library(effects)
plot(allEffects(model_1), main="1992-2022 subset")

exp(cbind(coef(model_c1),confint(model_c1)))
library(effects)
plot(allEffects(model_c1), main="1992-2022 control model")

# Instala balicku car + nutnosti na strane OS
#install.packages("car")
#install.packages("rlg", dependencies=TRUE, repos='http://cran.rstudio.com/')
#install.packages("clusterSim", dependencies=FALSE, repos='http://cran.rstudio.com/')
#update.packages(lib.loc = "/usr/local/lib/R/site-library")
install.packages("car")
# Nacteni knihovny
library(car)


# Nastaveni grafu
png(filename="C:/Users/user/Desktop/regrese/residualPlots1992-2022.png",
    width = 1920,
    height = 1200,
    pointsize=30
)
# ------------------------------------------------------------------ #
# Residua = jak moc je model dobry
# residua vs promenna -> melo by byt nahodne
# Tisk obrazku do souboru
residualPlots(model_1,fitted=TRUE)

dev.off() # zavreni souboru s grafem

# Zobrazeni koeficientu modelu, format P = coef1*promenna + coef2*promenna ...
#coef(model_1)
#exp(coef(model_1))
#inframe = data.frame(conflict=0,          hr=0, inequal=0, area=1, gdp=20, security=10, unsc=5,
gdpgrowth=0, unsc=0)
#result = predict(model_1,data, type="response")
#result = predict(model_1,inframe, type="response")
pr_result = predict(model_1,dataN, type="response")
pr_result

pr_result = predict(model_c1,dataN, type="response")
pr_result
```

```
#mcfaddenR2index
library(pscl)
head(dataN)
model_1 <- glm(result ~ conflict + inequal + gdpgrowth + security + gdp + nuclear ,
family=binomial("logit"),data=dataN)
summary(model_1)
pR2(model_1)
model_c1 <- glm(result ~  conflict + gdpgrowth + gdp, family=binomial("logit"),data=dataN)
summary(model_c1)
pR2(model_c1)
#likelihoo ratio test for models
library(lmtest)
library(zoo)
lrtest(model_1, model_c1)
AIC(model_1,model_c1,model_1c2, k = 2)

#hosmer-lemeshow test
library(ResourceSelection)
model_1 <- glm(result ~ conflict + inequal + gdpgrowth + security + gdp + nuclear ,
family=binomial("logit"),data=dataN)
hoslem.test(model_1$y, model_1$fitted)
hoslem.test(model_c1$y, model_c1$fitted)
```