

```

#-----
# Abgabe: Unit 6
# Verfasser: Jonas Benischek
# Beschreibung: Vorhersagefehler eines linearen Modells
#-----

# MANDATORY: Definieren des Stammordners Ändern Sie diese Zeile NICHT
rootDIR = "C:/Datenanalyse/UsingR/Unit6/"

#####
# Erholung vs. Siedlung neu betrachtet
# Link: https://geomor.github.io/mor-mpg-data-analysis/unit06/unit06-03\_assignment.html
#
#####
# Bitte schreiben Sie ein R-Markdown-Skript, Dies
# hilft Ihnen, die Leistung eines einfachen linearen
# Modells, das die prozentuale Erholungsfläche
# aus dem jeweiligen Siedlungsgebiet vorhersagt,
# statistisch auszuwerten. Wählen Sie daher bitte ein
# geeignetes Kreuzvalidierungsverfahren.
# Das Verfahren sollte als Funktion implementiert werden. Fügen Sie am Ende ein Diagramm !
#
# Bitte schreiben Sie genau einen Satz als
# Zusammenfassung der Zuverlässigkeit des Modells.
#####
|
#####

#####
# Datensatz Laden

df <- readRDS("lu_clean.rds")

str(df) # Ansicht in der Console

#####
# Leave-many-out-Kreuzvalidierung

range <- nrow(anscombe)
nbr <- nrow(anscombe) * 0.8

cv_sample <- lapply(seq(100), function(i){
  set.seed(i)
  smp1 <- sample(range, nbr)
  train <- anscombe[smp1,]
  test <- anscombe[-smp1,]
  df <- lm(y1 ~ x1, data = train)
  pred <- predict(lmod, newdata = test)
  obsv <- test$y1
  resid <- obsv - pred
  ss_obsrv <- sum((obsv - mean(obsv))**2)
})

```

```

ss_model <- sum((pred - mean(obsv))**2)
ss_resid <- sum((obsv - pred)**2)
mss_obsrv <- ss_obsrv / (length(obsv) - 1)
mss_model <- ss_model / 1
mss_resid <- ss_resid / (length(obsv) - 2)
data.frame(pred = pred,
            obsv = obsv,
            resid = resid,
            ss_obsrv = ss_obsrv,
            ss_model = ss_model,
            ss_resid = ss_resid,
            mss_obsrv = mss_obsrv,
            mss_model = mss_model,
            mss_resid = mss_resid,
            r_squared = ss_model / ss_obsrv
)
})
cv_sample <- do.call("rbind", cv_sample)

ss_obsrv <- sum((cv_sample$obsv - mean(cv_sample$obsv))**2)
ss_model <- sum((cv_sample$pred - mean(cv_sample$obsv))**2)
ss_resid <- sum((cv_sample$obsv - cv_sample$pred)**2)

mss_obsrv <- ss_obsrv / (length(cv_sample$obsv) - 1)

mss_resid <- ss_resid / (length(cv_sample$obsv) - 2)
#####

data.frame(NAME = c("cross validation F value",
                    "linear model F value",
                    "cross validation r squared",
                    "linear model r squared"),
            VALUE = c(round(mss_model / mss_resid, 2),
                      round(anova(lmod)$'F value'[1], 2),
                      round(1 - ss_resid / ss_obsrv, 2),
                      round(summary(lmod)$r.squared, 2)))

#####

se <- function(x) sd(x, na.rm = TRUE)/sqrt(length(na.exclude(x)))

me <- round(mean(cv_sample$pred - cv_sample$obs, na.rm = TRUE), 2)
me_sd <- round(se(cv_sample$pred - cv_sample$obs), 2)
mae <- round(mean(abs(cv_sample$pred - cv_sample$obs), na.rm = TRUE), 2)
mae_sd <- round(se(abs(cv_sample$pred - cv_sample$obs)), 2)
rmse <- round(sqrt(mean((cv_sample$pred - cv_sample$obs)^2, na.rm = TRUE)), 2)
rmse_sd <- round(se((cv_sample$pred - cv_sample$obs)^2), 2)

data.frame(NAME = c("Mean error (ME)", "Std. error of ME",
                    "Mean absolute error (MAE)", "Std. error of MAE",
                    "Root mean square error (RMSE)", "Std. error of RMSE"),
            VALUE = c(me, me_sd,

```

```
mae, mae_sd,
rmse, rmse_sd))

#####
#####
# NAME VALUE
#           Mean error (ME) -0.02
#           Std. error of ME  0.06
#       Mean absolute error (MAE)  0.84
#           Std. error of MAE  0.04
# Root mean square error (RMSE)  1.11
#           Std. error of RMSE  0.08
#####

print(cv_sample)

summary(cv_sample)
```