Val ~ Noise

Luis Nicolás Luarte Rodríguez

```
## fit simple model
glm.mdl.null <- glm(Valence_num ~ 1,
                    data = df,
                    family = binomial(link = "logit"))
glm.mdl.1 <- glm(Valence_num ~ Noise,
                 data = df,
                 family = binomial(link = "logit"))
glm.mdl.2 <- glm(Valence_num ~ Noise + Pedestrians,
                 data = df,
                 family = binomial(link = "logit"))
glm.mdl.3 <- glm(Valence_num ~ Noise + Cars,
                 data = df,
                 family = binomial(link = "logit"))
glm.mdl.4 <- glm(Valence_num ~ Noise + Neighbourhood,
                 data = df,
                 family = binomial(link = "logit"))
glm.mdl.5 <- glm(Valence_num ~ Noise + Socioeconomic,
                 data = df,
                 family = binomial(link = "logit"))
glm.mdl.6 <- glm(Valence_num ~ Noise + Pedestrians + Cars,
                 data = df,
                 family = binomial(link = "logit"))
glm.mdl.7 <- glm(Valence_num ~ Noise + Pedestrians + Neighbourhood,
                 data = df,
                 family = binomial(link = "logit"))
glm.mdl.8 <- glm(Valence_num ~ Noise + Pedestrians + Socioeconomic,
                 data = df,
                 family = binomial(link = "logit"))
glm.list <- list(glm.mdl.1, glm.mdl.2, glm.mdl.3, glm.mdl.4, glm.mdl.5,
                 glm.mdl.6, glm.mdl.7, glm.mdl.8)
```

```
## walds-test
glm.wald <- sapply(glm.list, function(x) Anova(x, test.statistic = c("Wald")),
               simplify = FALSE)
## likelihood ratio test
glm.lr <- sapply(glm.list, function(x) anova(glm.mdl.null, x, test = "LRT"),
                 simplify = FALSE)
## score test
glm.score <- sapply(glm.list, function(x) anova(glm.mdl.null, x, test = "Rao"),
                    simplify = FALSE)
## hosmer & lemeshow, g = covariates + 1
glm.hl <- sapply(glm.list, function(x) hoslem.test(x$y, fitted(x), g= 3))
## AIC for every model
glm.aic <- sapply(glm.list, function(x) AIC(x))
## VIF for all models
glm.vif <- sapply(glm.list[2:8], function(x) vif(x))

## repeated k-fold cross validated models
cntrl <- trainControl(method = "repeatedcv",                        number = 10,
                      repeats = 10,
                      savePredictions = TRUE)
mdl.1.kfold <- train(as.factor(Valence_num) ~ Noise,
                     data = na.omit(df),
                     method = "glm",
                     family = binomial(link = "logit"),
                     trControl = cntrl)
mdl.2.kfold <- train(as.factor(Valence_num) ~ Noise + Pedestrians,
                     data = na.omit(df),
                     method = "glm",
                     family = binomial(link = "logit"),
                     trControl = cntrl)
mdl.3.kfold <- train(as.factor(Valence_num) ~ Noise + Cars,
                     data = na.omit(df),
                     method = "glm",
                     family = binomial(link = "logit"),
                     trControl = cntrl)
mdl.4.kfold <- train(as.factor(Valence_num) ~ Noise + Neighbourhood,
                     data = na.omit(df),
```

```
                    method = "glm",
                    family = binomial(link = "logit"),
                    trControl = cntrl)
mdl.5.kfold <- train(as.factor(Valence_num) ~ Noise + Socioeconomic,
                    data = na.omit(df),
                    method = "glm",
                    family = binomial(link = "logit"),
                    trControl = cntrl)
mdl.6.kfold <- train(as.factor(Valence_num) ~ Noise + Pedestrians + Cars,
                    data = na.omit(df),
                    method = "glm",
                    family = binomial(link = "logit"),
                    trControl = cntrl)
mdl.7.kfold <- train(as.factor(Valence_num) ~ Noise + Pedestrians + Neighbourhood,
                    data = na.omit(df),
                    method = "glm",
                    family = binomial(link = "logit"),
                    trControl = cntrl)
mdl.8.kfold <- train(as.factor(Valence_num) ~ Noise + Pedestrians + Socioeconomic,
                    data = na.omit(df),
                    method = "glm",
                    family = binomial(link = "logit"),
                    trControl = cntrl)


mdl.1.pred <- predict(mdl.1.kfold)
mdl.2.pred <- predict(mdl.2.kfold)
mdl.3.pred <- predict(mdl.3.kfold)
mdl.4.pred <- predict(mdl.4.kfold)
mdl.5.pred <- predict(mdl.4.kfold)
mdl.6.pred <- predict(mdl.4.kfold)
mdl.7.pred <- predict(mdl.4.kfold)
mdl.8.pred <- predict(mdl.4.kfold)

mdl.1.cm <- caret::confusionMatrix(mdl.1.pred, as.factor(df$Valence_num))
mdl.2.cm <- caret::confusionMatrix(mdl.2.pred, as.factor(df$Valence_num))
mdl.3.cm <- caret::confusionMatrix(mdl.3.pred, as.factor(df$Valence_num))
mdl.4.cm <- caret::confusionMatrix(mdl.4.pred, as.factor(df$Valence_num))
```

```
mdl.5.cm <- caret::confusionMatrix(mdl.1.pred, as.factor(df$Valence_num))
mdl.6.cm <- caret::confusionMatrix(mdl.2.pred, as.factor(df$Valence_num))
mdl.7.cm <- caret::confusionMatrix(mdl.3.pred, as.factor(df$Valence_num))
mdl.8.cm <- caret::confusionMatrix(mdl.4.pred, as.factor(df$Valence_num))


par(mfrow = c(4,2))
fourfoldplot(mdl.1.cm$table, main = "Noise")
fourfoldplot(mdl.2.cm$table, main = "Noise + Pedestrians")
fourfoldplot(mdl.3.cm$table, main = "Noise + Cars")
fourfoldplot(mdl.4.cm$table, main = "Noise + Neighborhood")
fourfoldplot(mdl.4.cm$table, main = "Noise + Socioeconomic")
fourfoldplot(mdl.4.cm$table, main = "Noise + Pedestrians + Cars")
fourfoldplot(mdl.4.cm$table, main = "Noise + Pedestrians + Neighborhood")
fourfoldplot(mdl.4.cm$table, main = "Noise + Pedestrians + Socioeconomic")

library(MASS)
odd.ratio <- sapply(glm.list, function(x) exp(cbind(coef(x), confint(x))))

## plot linear odd-ratio
sapply(glm.list, function(x) plot(allEffects(x)))
```