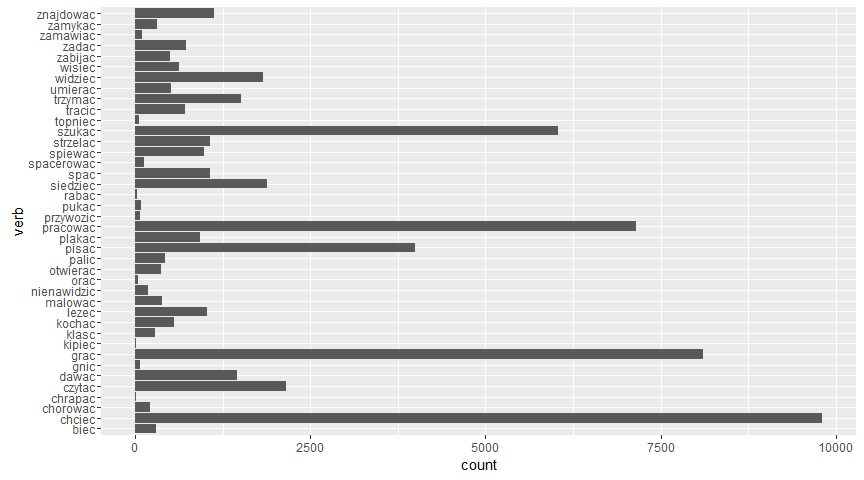
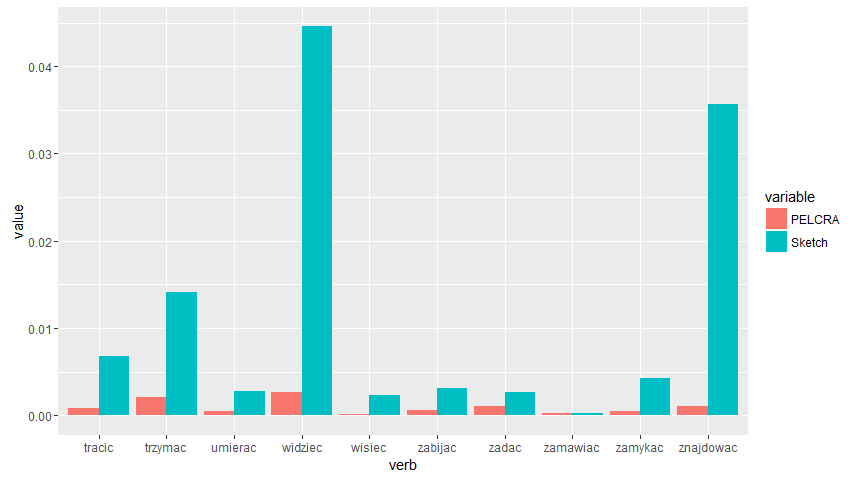
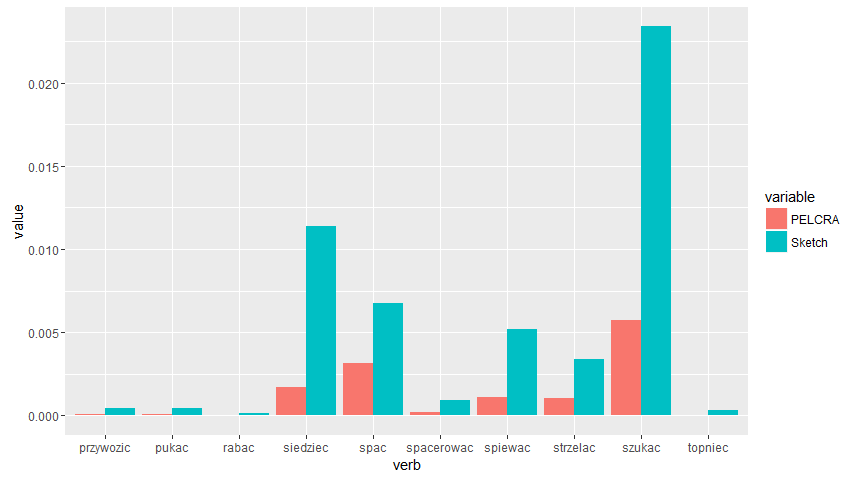
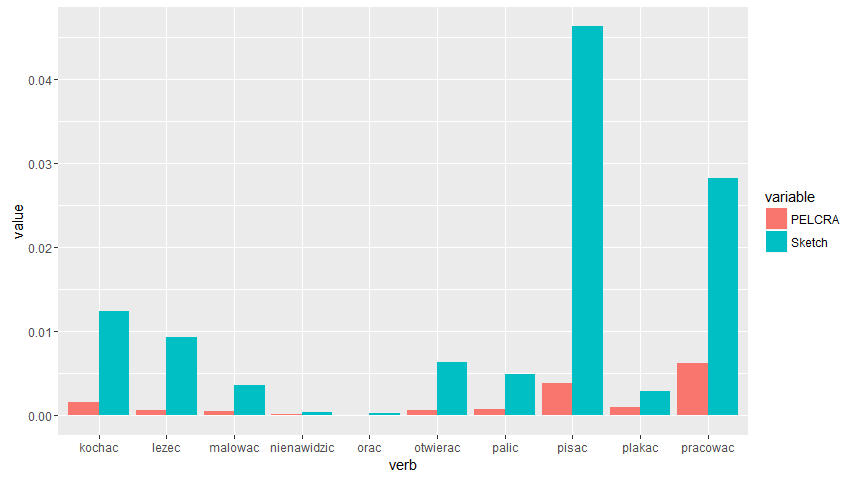
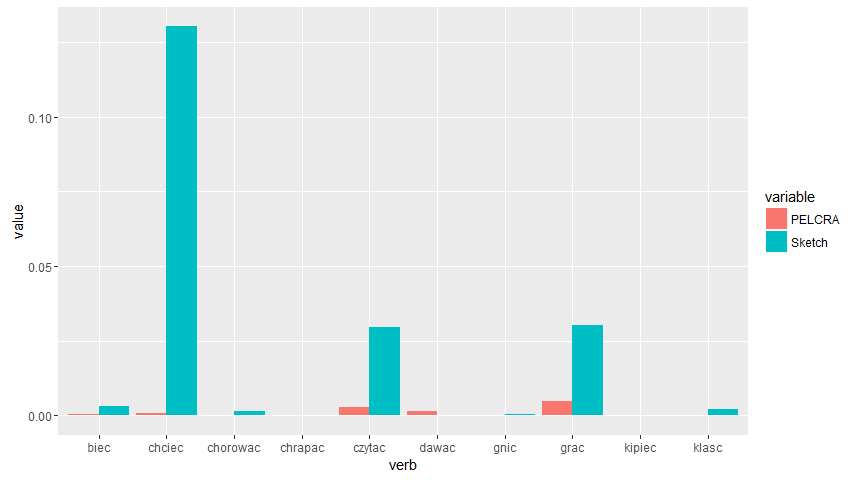
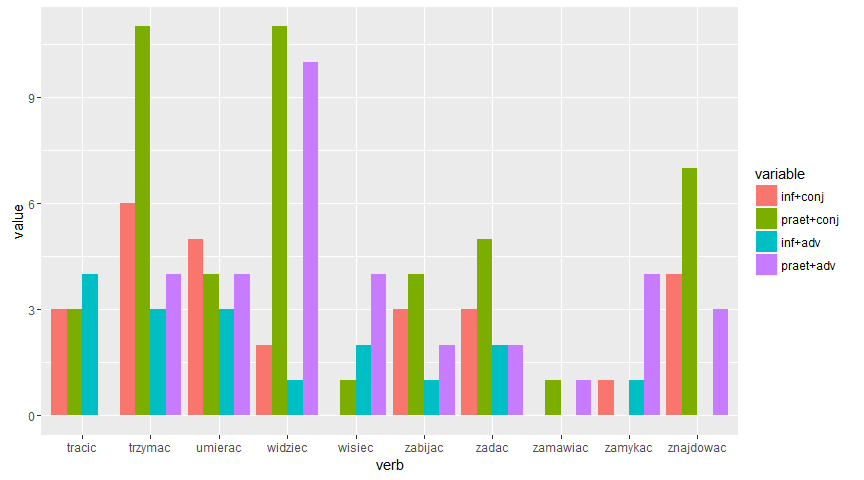
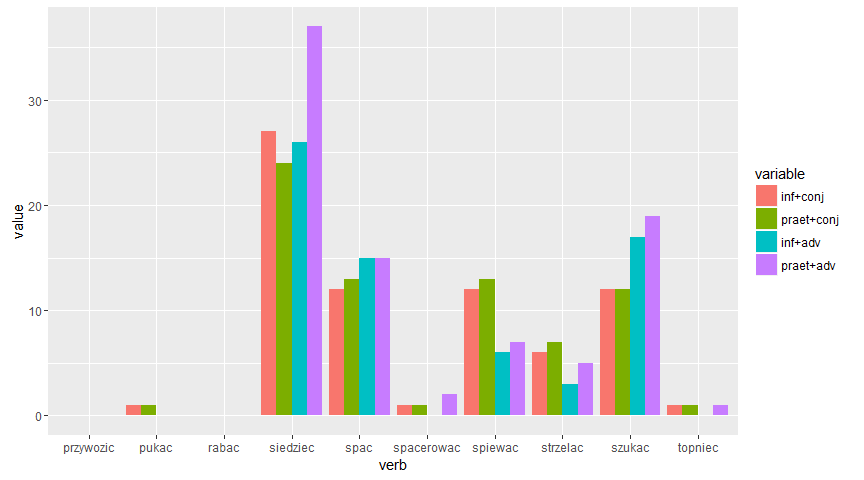
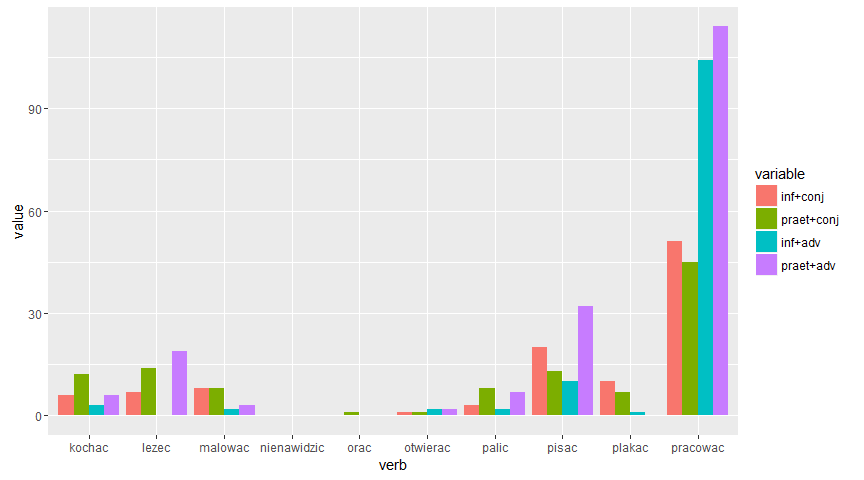
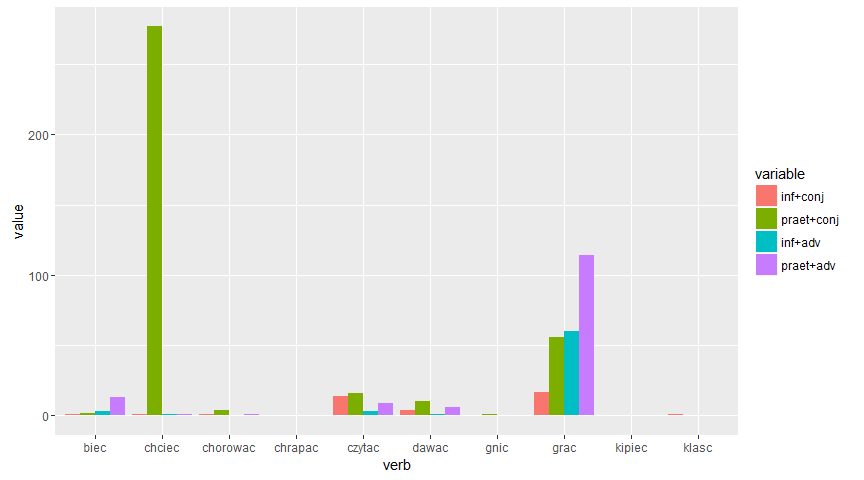
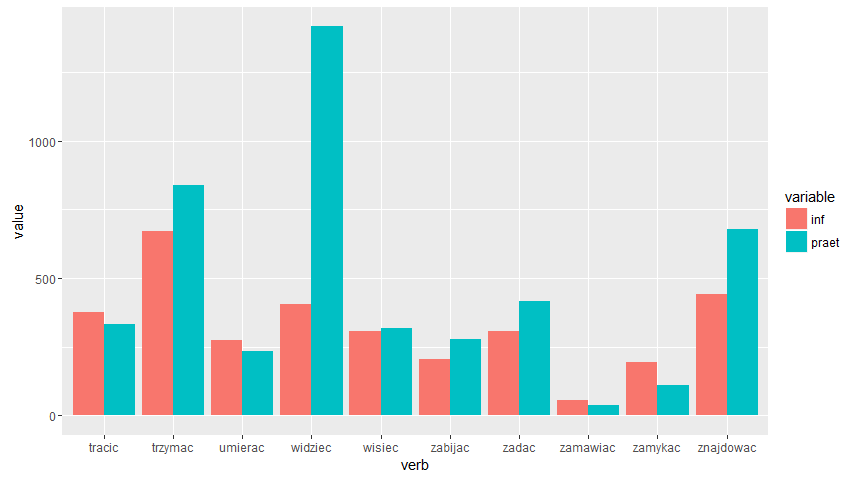
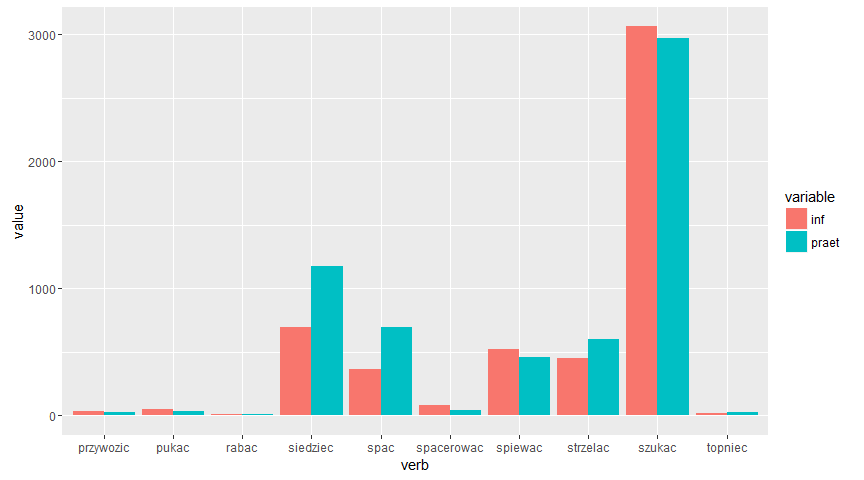
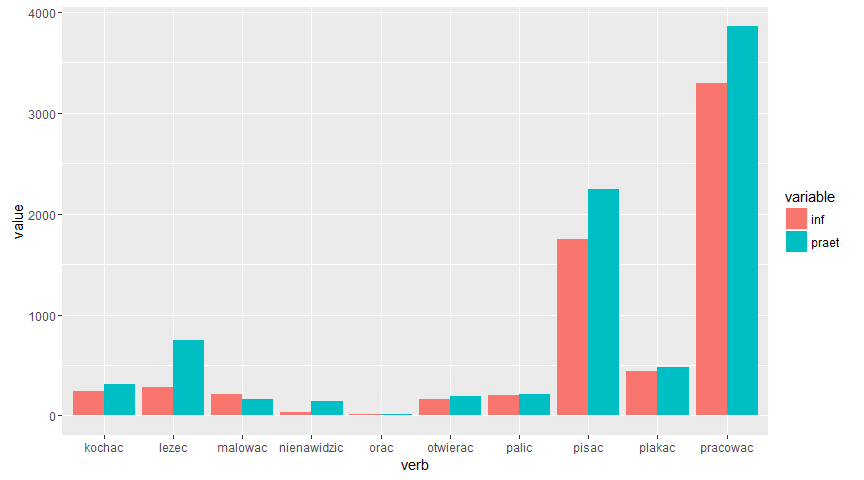
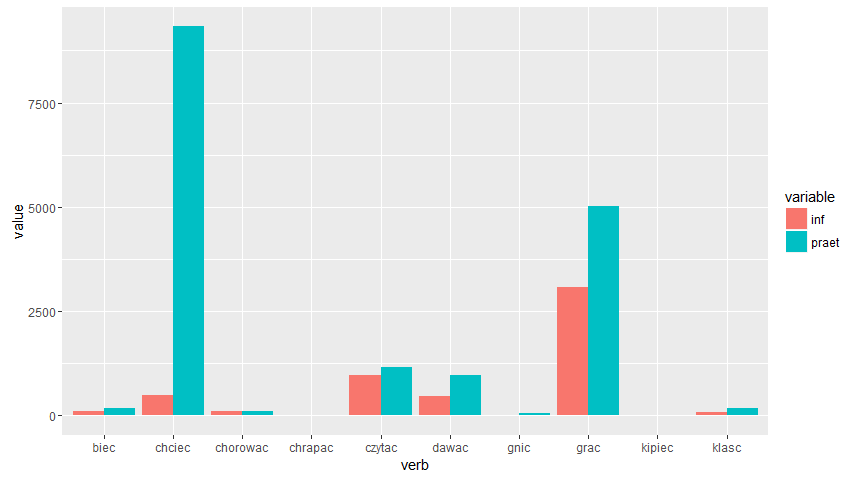
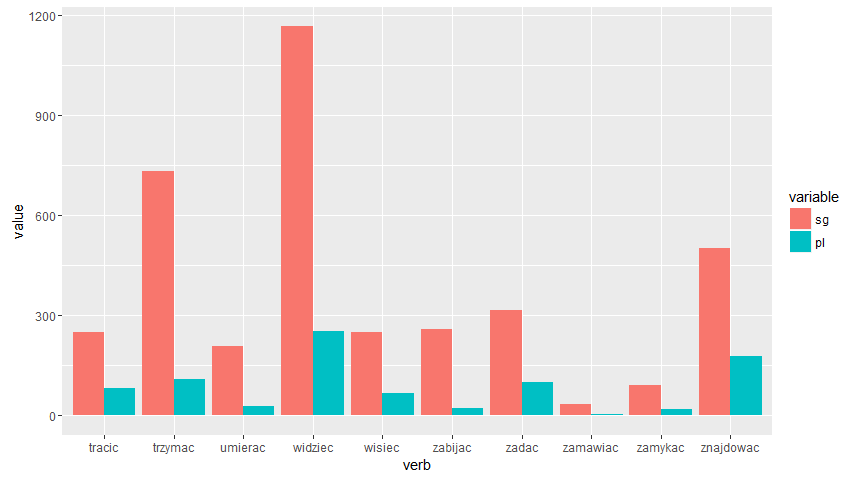
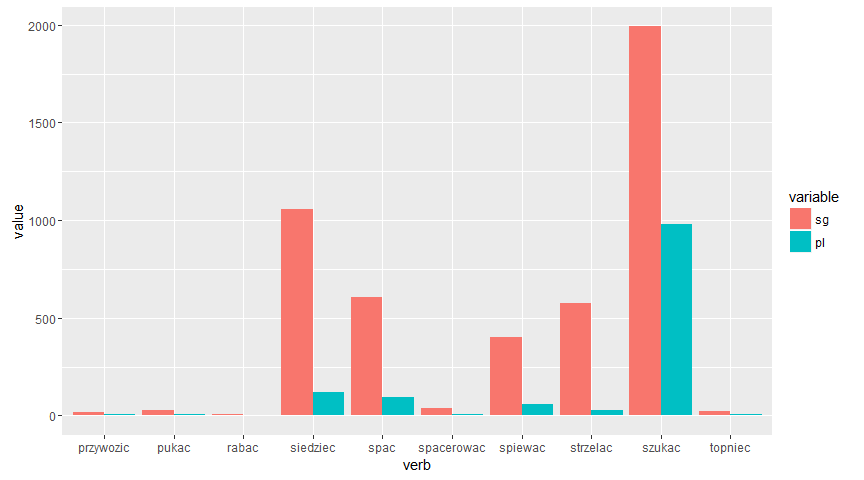
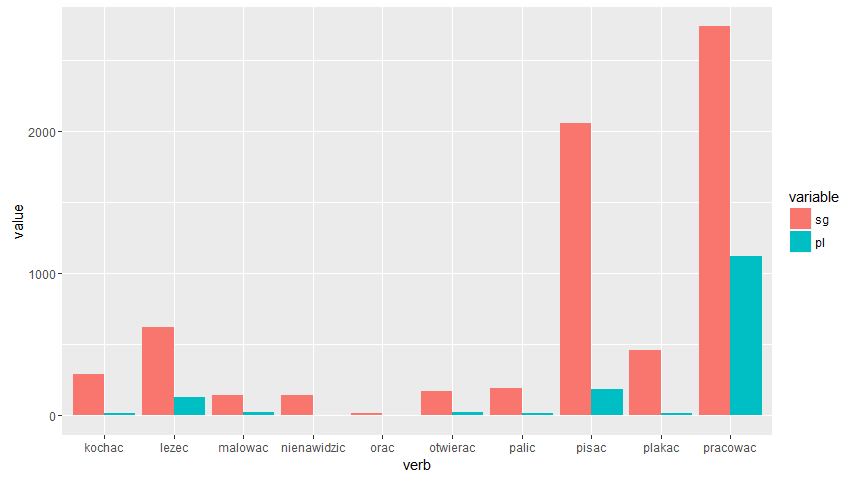
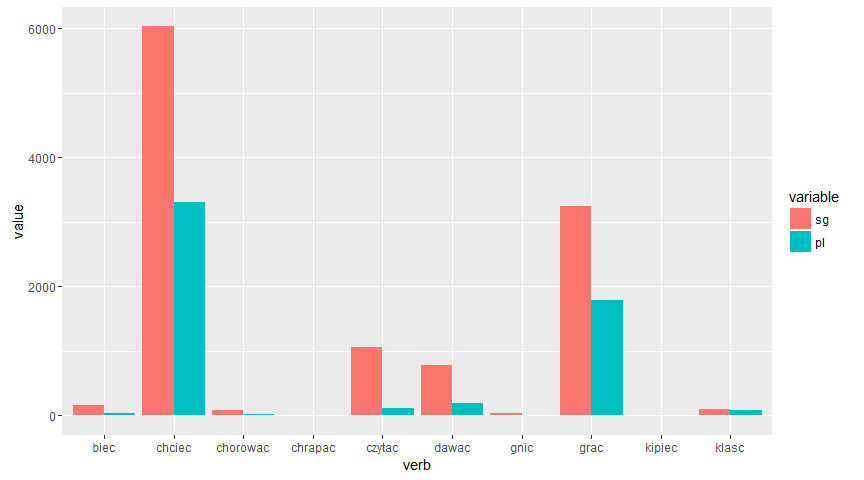
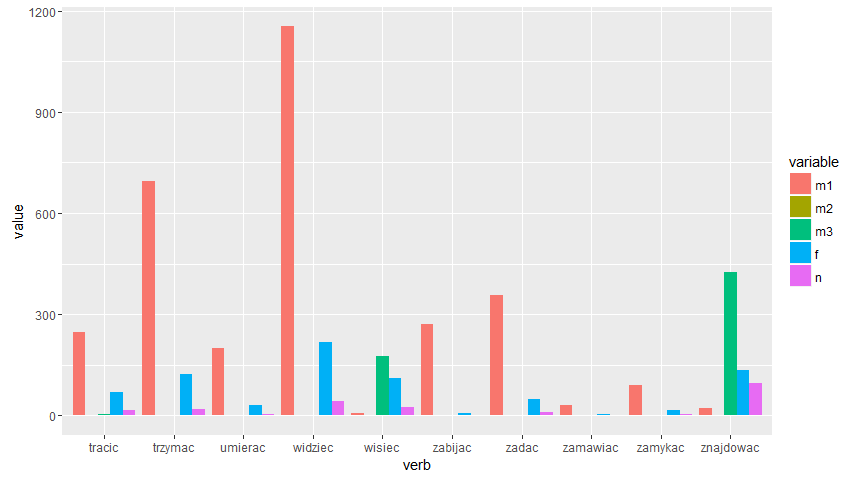
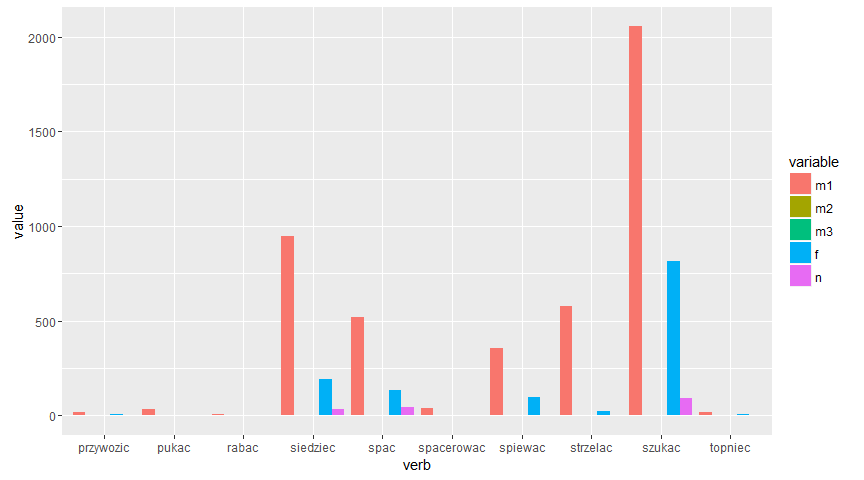
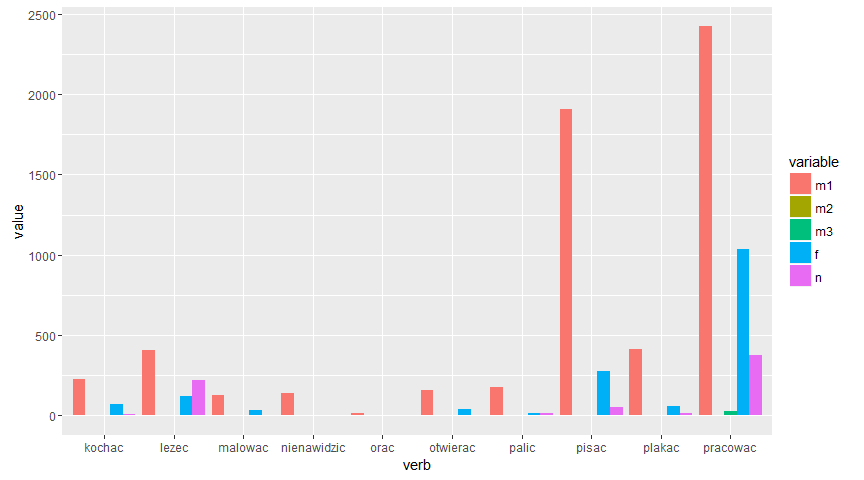
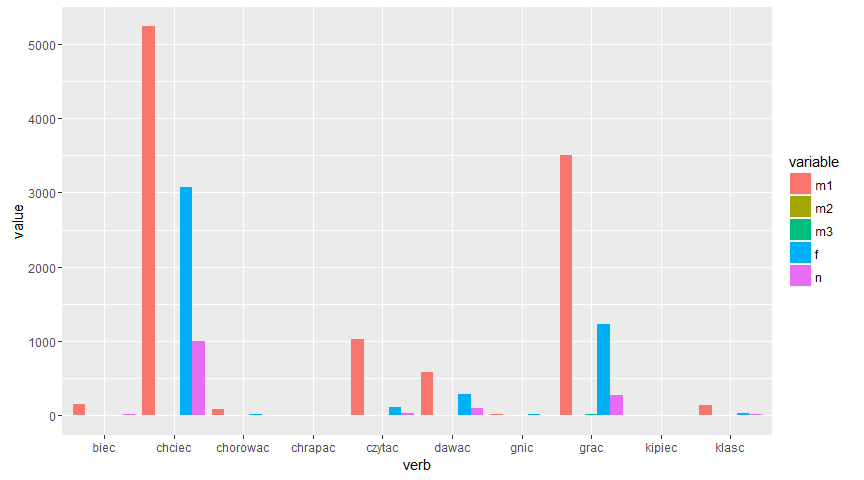
Polish

csvdata = read.csv("verbs\_infinitive.csv", encoding="UTF-8", check.names=FALSE)  
  
genderfields = c("m1", "m2", "m3", "f", "n")  
numberfields = c("sg", "pl")  
formfields = c("inf", "praet")  
syntaxfields = c("inf+conj", "praet+conj", "inf+adv", "praet+adv")  
countfields = c("count")   
percentfields = c("PELCRA", "Sketch")  
  
allfields = list(genderfields, numberfields, formfields, syntaxfields, percentfields)

#Describe data with the help of graphs  
  
ggplot(csvdata, aes(verb, count)) +   
 geom\_bar(position = "dodge", stat="identity") +  
 coord\_flip()



for (fields in allfields){  
 data=csvdata[c("verb", fields)];  
 count = 1;  
 len = 10;  
 while (count<40) {  
 start = count  
 end = start+len-1  
   
 shortdata = data[start:end,]  
 data.m <- melt(shortdata, id.vars="verb")  
 print(  
 ggplot(data.m, aes(verb, value, fill=variable)) +   
 geom\_bar(position = "dodge", stat="identity")  
 )  
   
 count=count+len  
 }  
}



chisq.test(c(csvdata$praet, csvdata$inf))

##   
## Chi-squared test for given probabilities  
##   
## data: c(csvdata$praet, csvdata$inf)  
## X-squared = 208490, df = 79, p-value < 2.2e-16

chisq.test(c(csvdata$m, csvdata$sg))

##   
## Chi-squared test for given probabilities  
##   
## data: c(csvdata$m, csvdata$sg)  
## X-squared = 76540, df = 39, p-value < 2.2e-16

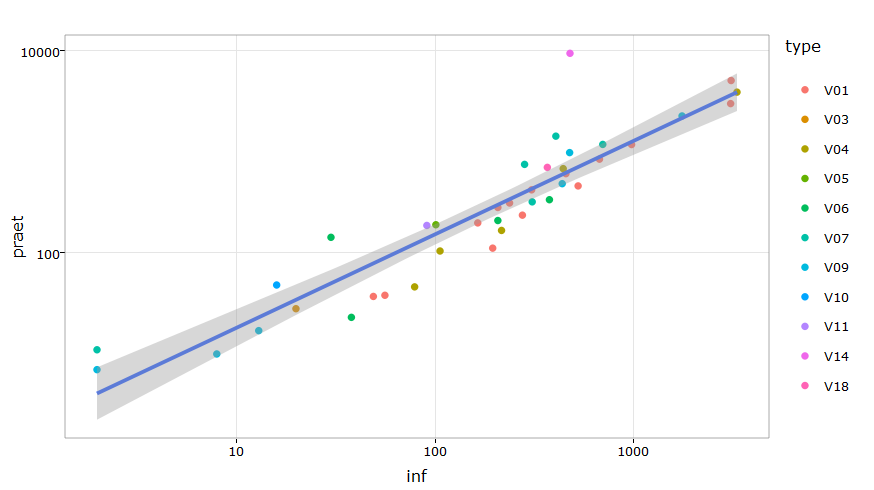
library(plotly)

##   
## Attaching package: 'plotly'

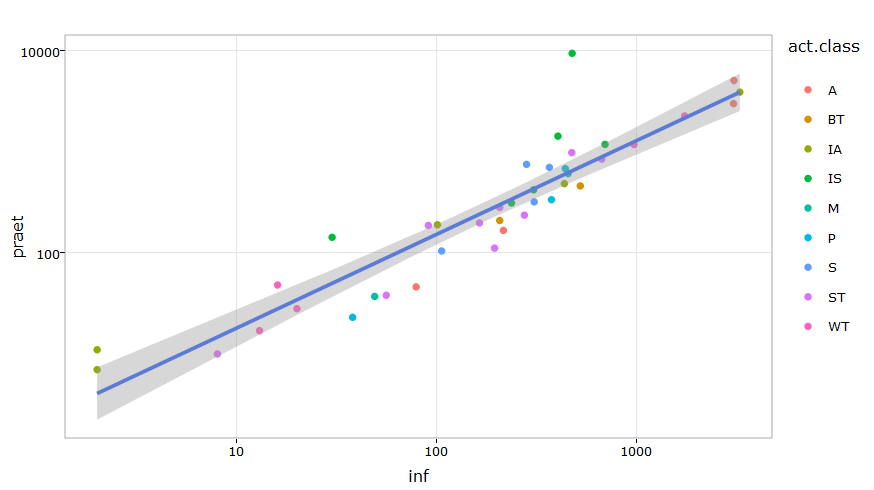
## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:graphics':  
##   
## layout

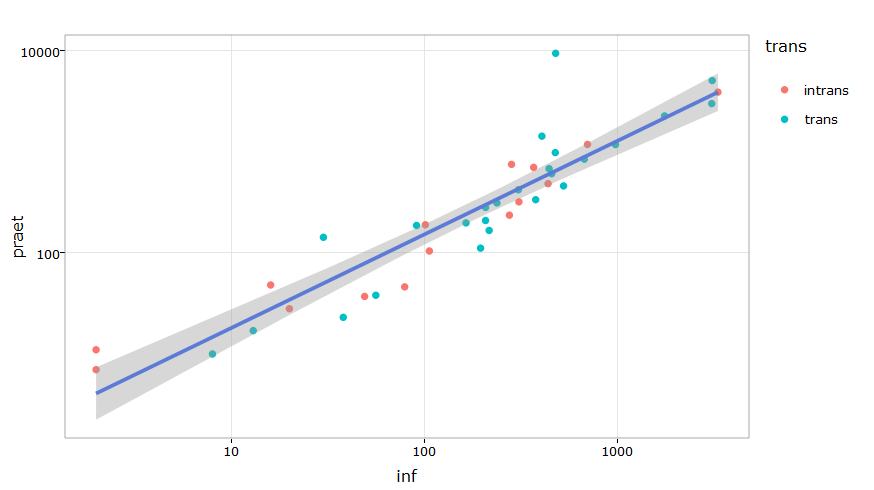
p <- ggplot(csvdata, aes(x = inf, y = praet))+  
 geom\_point(aes(color = type))+  
 scale\_y\_log10()+  
 scale\_x\_log10()+  
 theme\_bw()+  
 geom\_smooth(method = "lm")  
  
  
ggplotly(p, tooltip = c("z"))



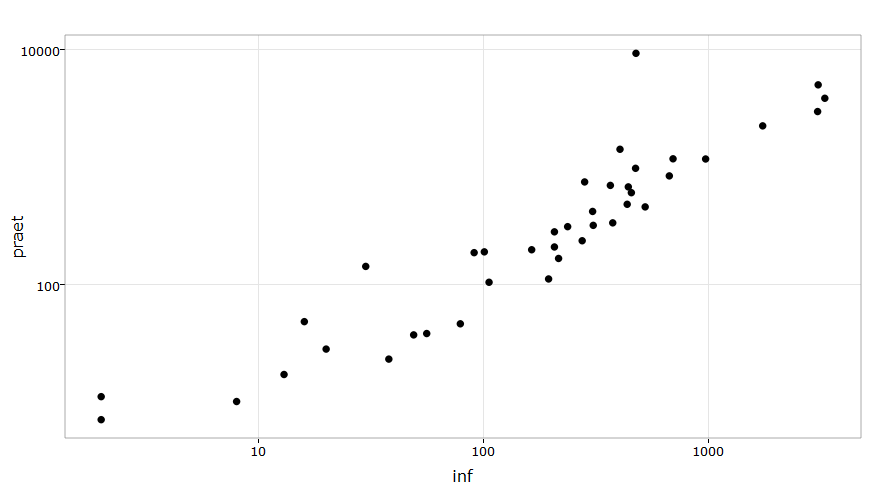
p <- ggplot(csvdata, aes(x = inf, y = praet))+  
 geom\_point(aes(color = act.class))+  
 scale\_y\_log10()+  
 scale\_x\_log10()+  
 theme\_bw()+  
 geom\_smooth(method = "lm")  
  
ggplotly(p, tooltip = c("z"))



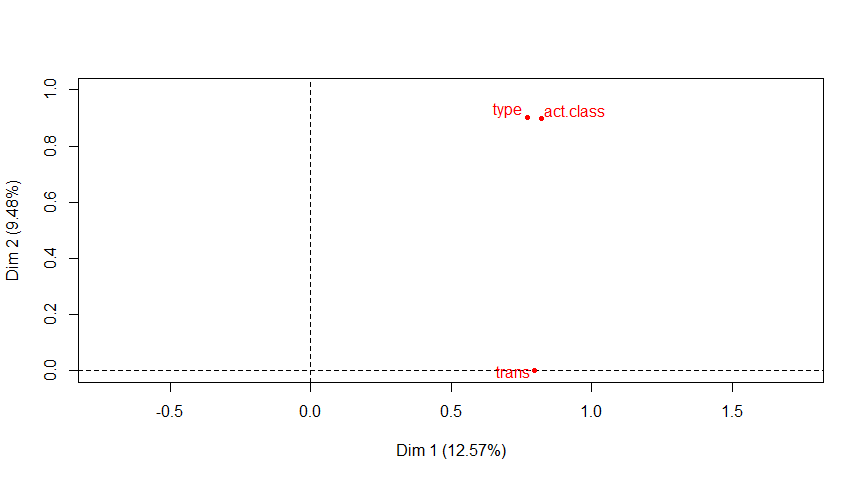
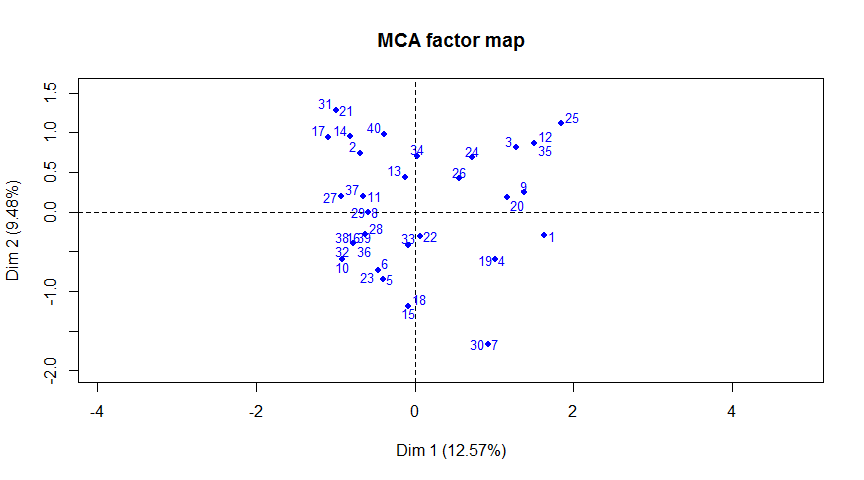
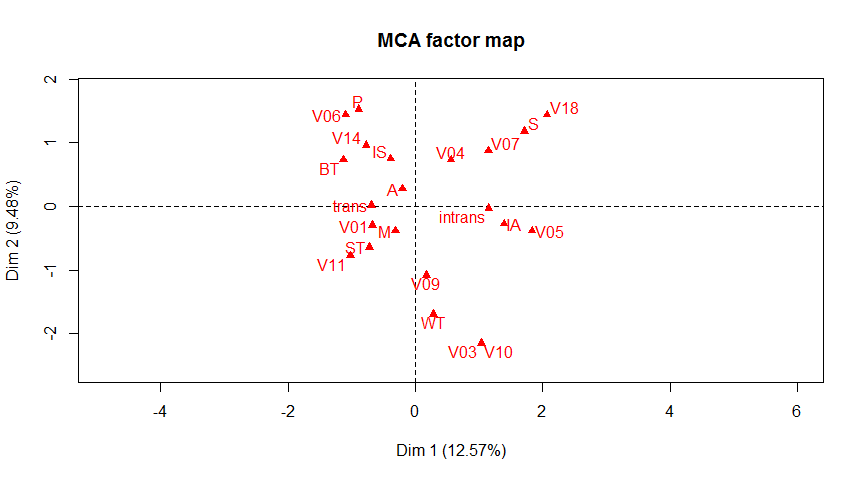
p <- ggplot(csvdata, aes(x = inf, y = praet))+  
 geom\_point(aes(color = trans))+  
 scale\_y\_log10()+  
 scale\_x\_log10()+  
 theme\_bw()+  
 geom\_smooth(method = "lm")  
  
ggplotly(p, tooltip = c("z"))



#Data aggregation  
aggdata <- aggregate(verb ~ inf + praet, data = csvdata,paste, collapse="-")  
p <- ggplot(aggdata, aes(x = inf, y = praet, z = verb))+  
 geom\_point()+  
 scale\_y\_log10()+  
 scale\_x\_log10()+  
 theme\_bw()  
ggplotly(p, tooltip = c("z"))



#MCA  
library(FactoMineR)  
newdata <- csvdata$inf/csvdata$praet  
csvdata$newdata <- newdata  
View(csvdata)  
#MCA doesn't work to work with numerical data  
q <- csvdata[c("trans", "act.class", "type")]  
res.mca <- MCA(q)



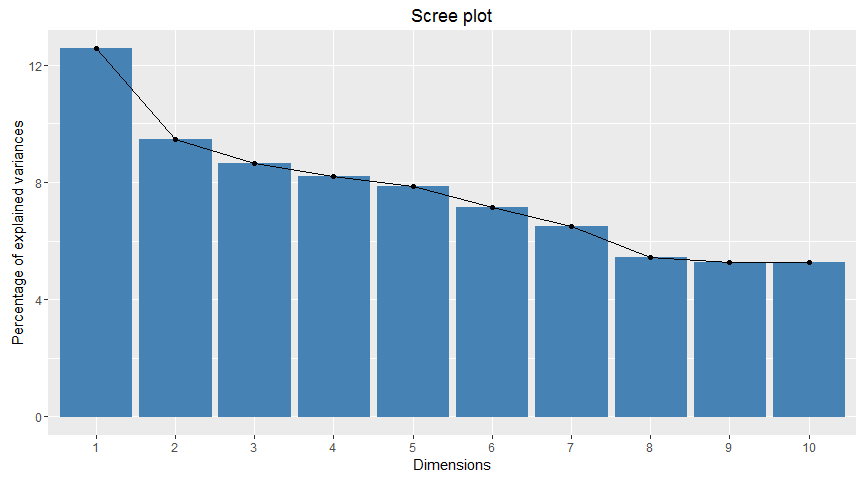
res.mca$eig

## eigenvalue percentage of variance cumulative percentage of variance  
## dim 1 0.79603551 12.5689817 12.56898  
## dim 2 0.60027017 9.4779500 22.04693  
## dim 3 0.54676244 8.6330911 30.68002  
## dim 4 0.51899985 8.1947345 38.87476  
## dim 5 0.49796442 7.8625962 46.73735  
## dim 6 0.45281104 7.1496480 53.88700  
## dim 7 0.41129787 6.4941769 60.38118  
## dim 8 0.34386254 5.4294085 65.81059  
## dim 9 0.33333333 5.2631579 71.07374  
## dim 10 0.33333333 5.2631579 76.33690  
## dim 11 0.33333333 5.2631579 81.60006  
## dim 12 0.27091258 4.2775670 85.87763  
## dim 13 0.22041394 3.4802201 89.35785  
## dim 14 0.18147045 2.8653230 92.22317  
## dim 15 0.15752633 2.4872578 94.71043  
## dim 16 0.12625243 1.9934595 96.70389  
## dim 17 0.09359098 1.4777523 98.18164  
## dim 18 0.06646523 1.0494510 99.23109  
## dim 19 0.04869756 0.7689089 100.00000

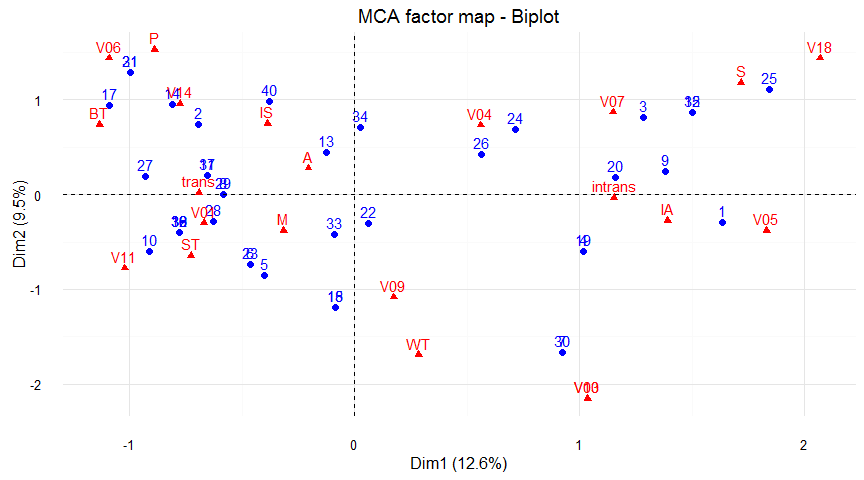
summary(res.mca, nb.dec = 2, ncp = 2)

##   
## Call:  
## MCA(X = q)   
##   
##   
## Eigenvalues  
## Dim.1 Dim.2 Dim.3 Dim.4 Dim.5 Dim.6 Dim.7  
## Variance 0.80 0.60 0.55 0.52 0.50 0.45 0.41  
## % of var. 12.57 9.48 8.63 8.19 7.86 7.15 6.49  
## Cumulative % of var. 12.57 22.05 30.68 38.87 46.74 53.89 60.38  
## Dim.8 Dim.9 Dim.10 Dim.11 Dim.12 Dim.13 Dim.14  
## Variance 0.34 0.33 0.33 0.33 0.27 0.22 0.18  
## % of var. 5.43 5.26 5.26 5.26 4.28 3.48 2.87  
## Cumulative % of var. 65.81 71.07 76.34 81.60 85.88 89.36 92.22  
## Dim.15 Dim.16 Dim.17 Dim.18 Dim.19  
## Variance 0.16 0.13 0.09 0.07 0.05  
## % of var. 2.49 1.99 1.48 1.05 0.77  
## Cumulative % of var. 94.71 96.70 98.18 99.23 100.00  
##   
## Individuals (the 10 first)  
## Dim.1 ctr cos2 Dim.2 ctr cos2   
## 1 | 1.63 8.39 0.17 | -0.30 0.36 0.01 |  
## 2 | -0.69 1.51 0.03 | 0.74 2.29 0.04 |  
## 3 | 1.28 5.16 0.28 | 0.81 2.73 0.11 |  
## 4 | 1.01 3.23 0.22 | -0.60 1.49 0.07 |  
## 5 | -0.40 0.51 0.05 | -0.85 3.00 0.23 |  
## 6 | -0.47 0.68 0.07 | -0.73 2.24 0.17 |  
## 7 | 0.92 2.68 0.05 | -1.67 11.57 0.17 |  
## 8 | -0.59 1.08 0.09 | 0.00 0.00 0.00 |  
## 9 | 1.38 5.97 0.36 | 0.24 0.24 0.01 |  
## 10 | -0.91 2.61 0.06 | -0.60 1.51 0.03 |  
##   
## Categories (the 10 first)  
## Dim.1 ctr cos2 v.test Dim.2 ctr cos2 v.test   
## intrans | 1.15 20.87 0.80 5.58 | -0.03 0.02 0.00 -0.15 |  
## trans | -0.69 12.52 0.80 -5.58 | 0.02 0.01 0.00 0.15 |  
## A | -0.21 0.18 0.00 -0.43 | 0.28 0.43 0.01 0.58 |  
## BT | -1.13 2.69 0.07 -1.62 | 0.73 1.50 0.03 1.05 |  
## IA | 1.39 10.12 0.28 3.28 | -0.27 0.52 0.01 -0.65 |  
## IS | -0.39 0.94 0.03 -1.01 | 0.75 4.67 0.10 1.96 |  
## M | -0.32 0.21 0.01 -0.45 | -0.38 0.40 0.01 -0.55 |  
## P | -0.89 2.48 0.06 -1.58 | 1.53 9.69 0.19 2.71 |  
## S | 1.72 12.35 0.33 3.57 | 1.18 7.75 0.16 2.46 |  
## ST | -0.73 4.97 0.15 -2.44 | -0.64 5.14 0.12 -2.16 |  
##   
## Categorical variables (eta2)  
## Dim.1 Dim.2   
## trans | 0.80 0.00 |  
## act.class | 0.82 0.90 |  
## type | 0.77 0.90 |

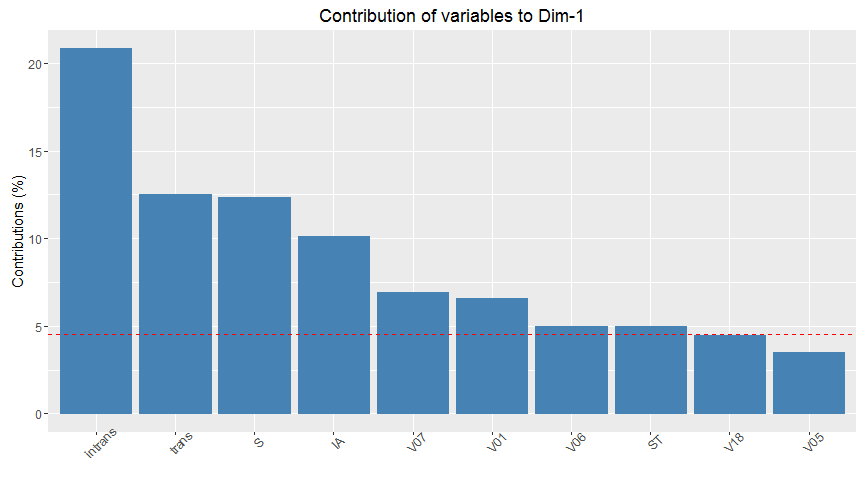
library(factoextra)  
fviz\_screeplot(res.mca)



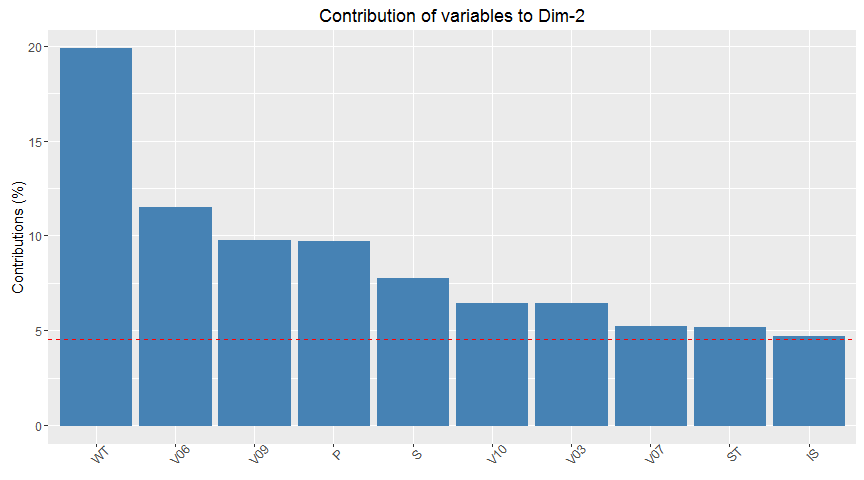
fviz\_mca\_biplot(res.mca) +  
 theme\_minimal()



fviz\_contrib(res.mca, choice = "var", axes = 1, top = 10)



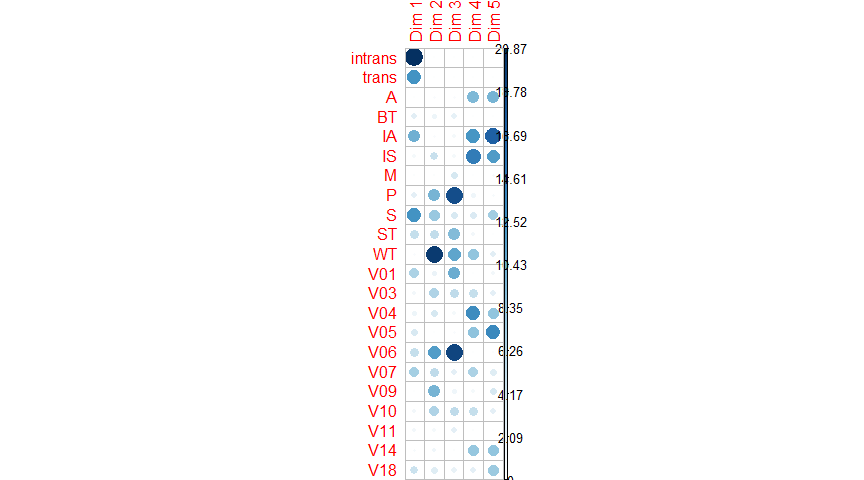
fviz\_contrib(res.mca, choice = "var", axes = 2, top = 10)



var <- get\_mca\_var(res.mca)  
head(round(var$contrib,2))

## Dim 1 Dim 2 Dim 3 Dim 4 Dim 5  
## intrans 20.87 0.02 0.00 0.03 0.01  
## trans 12.52 0.01 0.00 0.02 0.01  
## A 0.18 0.43 0.49 9.32 9.78  
## BT 2.69 1.50 2.14 0.03 0.00  
## IA 10.12 0.52 0.96 12.27 17.04  
## IS 0.94 4.67 0.93 14.54 11.82

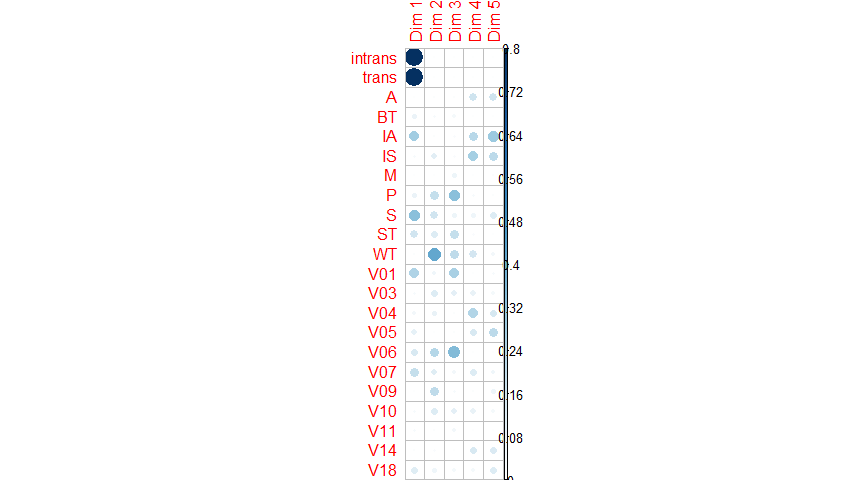
library("corrplot")  
corrplot(var$contrib, is.corr = FALSE)



head(var$cos2)

## Dim 1 Dim 2 Dim 3 Dim 4 Dim 5  
## intrans 0.797341722 0.0005695646 9.736116e-06 0.0008503396 3.077969e-04  
## trans 0.797341722 0.0005695646 9.736116e-06 0.0008503396 3.077969e-04  
## A 0.004711695 0.0085788993 8.910301e-03 0.1611697825 1.623594e-01  
## BT 0.067549489 0.0283619757 3.697386e-02 0.0005391563 1.864322e-06  
## IA 0.276085233 0.0107992892 1.798478e-02 0.2184210196 2.910053e-01  
## IS 0.026322677 0.0989031172 1.792908e-02 0.2664118732 2.077632e-01

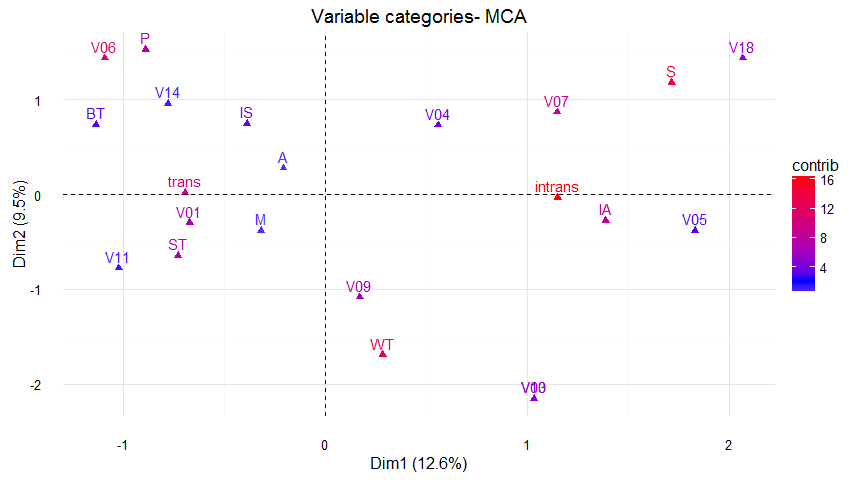
corrplot(var$cos2, is.corr=FALSE)



res.mca$quali.sup

## NULL

fviz\_mca\_var(res.mca, col.var ="contrib")+scale\_color\_gradient2(low="white", mid="blue", high="red", midpoint=2)+theme\_minimal()



res.desc <- dimdesc(res.mca, axes = c(1,2))  
res.desc$'Dim 1'

## $quali  
## R2 p.value  
## trans 0.7973417 9.642117e-15  
## act.class 0.8204011 1.414810e-09  
## type 0.7703637 7.138521e-07  
##   
## $category  
## Estimate p.value  
## intrans 0.8228162 9.642117e-15  
## S 1.5584028 1.135629e-04  
## IA 1.2665205 4.985772e-04  
## V07 0.6767280 5.128288e-03  
## V18 1.4967090 3.690237e-02  
## V06 -1.3222229 2.096246e-02  
## ST -0.6220752 1.248476e-02  
## V01 -0.9457902 1.274993e-03  
## trans -0.8228162 9.642117e-15

res.desc$'Dim 2'

## $quali  
## R2 p.value  
## act.class 0.8998823 2.132183e-13  
## type 0.9003586 7.003694e-12  
##   
## $category  
## Estimate p.value  
## V06 1.2136692 1.748204e-03  
## P 1.0545134 5.110214e-03  
## S 0.7880465 1.194514e-02  
## V07 0.7727478 3.849392e-02  
## IS 0.4528793 4.810991e-02  
## V03 -1.5672919 2.951101e-02  
## V10 -1.5672919 2.951101e-02  
## ST -0.6242381 2.889019e-02  
## V09 -0.7398419 3.177306e-03  
## WT -1.4379513 8.915644e-06

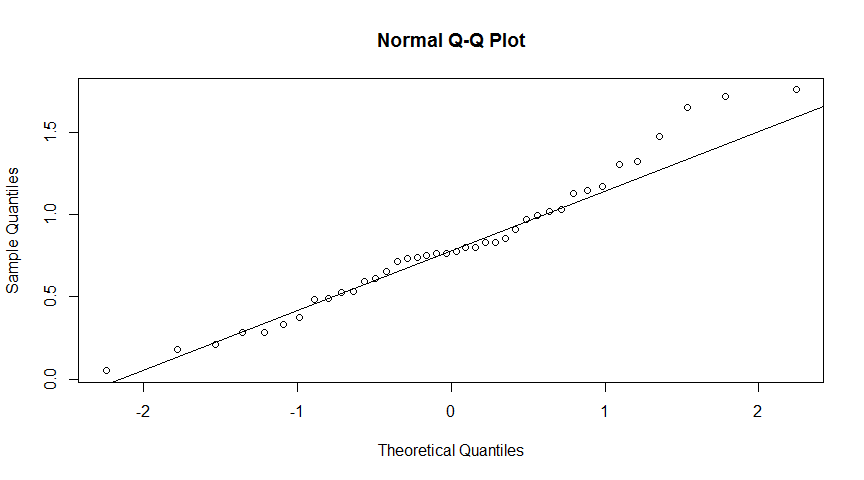
#Regression  
  
ks.test(newdata, "pnorm")

##   
## One-sample Kolmogorov-Smirnov test  
##   
## data: newdata  
## D = 0.54714, p-value = 9.202e-12  
## alternative hypothesis: two-sided

shapiro.test(rnorm(newdata))

##   
## Shapiro-Wilk normality test  
##   
## data: rnorm(newdata)  
## W = 0.9152, p-value = 0.005459

qqnorm(newdata)  
qqline(newdata)



shapiro.test(rnorm(csvdata$type))

##   
## Shapiro-Wilk normality test  
##   
## data: rnorm(csvdata$type)  
## W = 0.98183, p-value = 0.7568

for.regr <- csvdata[,c("newdata", "trans", "act.class", "type", "Sketch", "PELCRA")]  
lr <- lm(newdata ~ ., data = for.regr)  
summary(lr)

##   
## Call:  
## lm(formula = newdata ~ ., data = for.regr)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.4103 -0.1674 0.0000 0.1340 0.7271   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.50662 0.32965 4.570 0.000237 \*\*\*  
## transtrans -0.35448 0.23486 -1.509 0.148567   
## act.classBT 0.01070 0.36838 0.029 0.977140   
## act.classIA -0.67454 0.32956 -2.047 0.055568 .   
## act.classIS -0.43005 0.29450 -1.460 0.161457   
## act.classM -0.27991 0.36849 -0.760 0.457331   
## act.classP 0.17474 0.38983 0.448 0.659333   
## act.classS -0.45963 0.35056 -1.311 0.206300   
## act.classST -0.08961 0.27502 -0.326 0.748293   
## act.classWT 0.03658 0.36410 0.100 0.921094   
## typeV03 -0.82682 0.51033 -1.620 0.122579   
## typeV04 0.04130 0.26281 0.157 0.876886   
## typeV05 -0.27473 0.46266 -0.594 0.560031   
## typeV06 -0.11617 0.30085 -0.386 0.703925   
## typeV07 -0.35427 0.29501 -1.201 0.245365   
## typeV09 -0.29581 0.22222 -1.331 0.199766   
## typeV10 -1.20771 0.51030 -2.367 0.029363 \*   
## typeV11 -0.55795 0.39100 -1.427 0.170708   
## typeV14 0.48555 1.34653 0.361 0.722600   
## typeV18 -0.49789 0.49170 -1.013 0.324679   
## Sketch -8.95580 10.49000 -0.854 0.404465   
## PELCRA 12.00148 78.32771 0.153 0.879928   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3618 on 18 degrees of freedom  
## Multiple R-squared: 0.6471, Adjusted R-squared: 0.2354   
## F-statistic: 1.572 on 21 and 18 DF, p-value: 0.1682

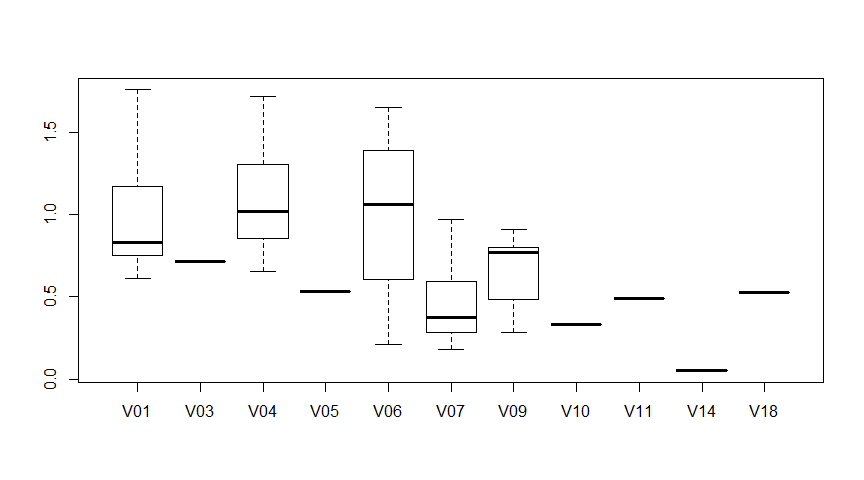
a <- summary(step(lr, direction = "both"))

## Start: AIC=-69.27  
## newdata ~ trans + act.class + type + Sketch + PELCRA  
##   
## Df Sum of Sq RSS AIC  
## - PELCRA 1 0.00307 2.3594 -71.219  
## - type 10 1.38063 3.7369 -70.825  
## - Sketch 1 0.09541 2.4517 -69.684  
## - act.class 8 1.13687 3.4932 -69.523  
## <none> 2.3563 -69.271  
## - trans 1 0.29822 2.6545 -66.505  
##   
## Step: AIC=-71.22  
## newdata ~ trans + act.class + type + Sketch  
##   
## Df Sum of Sq RSS AIC  
## - type 10 1.37912 3.7385 -72.808  
## <none> 2.3594 -71.219  
## - act.class 8 1.16303 3.5224 -71.189  
## - Sketch 1 0.19081 2.5502 -70.109  
## + PELCRA 1 0.00307 2.3563 -69.271  
## - trans 1 0.30002 2.6594 -68.431  
##   
## Step: AIC=-72.81  
## newdata ~ trans + act.class + Sketch  
##   
## Df Sum of Sq RSS AIC  
## - trans 1 0.08739 3.8259 -73.884  
## <none> 3.7385 -72.808  
## - Sketch 1 0.23023 3.9687 -72.417  
## - act.class 8 1.92239 5.6609 -72.212  
## + type 10 1.37912 2.3594 -71.219  
## + PELCRA 1 0.00157 3.7369 -70.825  
##   
## Step: AIC=-73.88  
## newdata ~ act.class + Sketch  
##   
## Df Sum of Sq RSS AIC  
## <none> 3.8259 -73.884  
## + trans 1 0.08739 3.7385 -72.808  
## - act.class 8 2.05453 5.8804 -72.690  
## - Sketch 1 0.33299 4.1589 -72.545  
## + PELCRA 1 0.00079 3.8251 -71.892  
## + type 10 1.16649 2.6594 -68.431

head(predict(lr, interval = "conf"))

## fit lwr upr  
## 1 0.53439153 -0.22574118 1.2945242  
## 2 0.05113636 -0.70899635 0.8112691  
## 3 1.07602735 0.50079093 1.6512638  
## 4 0.53445236 0.02055743 1.0483473  
## 5 0.95800491 0.42554008 1.4904697  
## 6 0.78442288 0.27163388 1.2972119

lrType <- lm(newdata ~ type, data = csvdata)  
boxplot(newdata~type, data=csvdata)



fligner.test(newdata~type, data=csvdata)

##   
## Fligner-Killeen test of homogeneity of variances  
##   
## data: newdata by type  
## Fligner-Killeen:med chi-squared = 10.971, df = 10, p-value =  
## 0.3598

summary(lrType)

##   
## Call:  
## lm(formula = newdata ~ type, data = csvdata)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.7848 -0.2074 0.0000 0.1345 0.7592   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.997597 0.097657 10.215 4.06e-11 \*\*\*  
## typeV03 -0.283311 0.378225 -0.749 0.4599   
## typeV04 0.111115 0.190369 0.584 0.5639   
## typeV05 -0.463205 0.378225 -1.225 0.2306   
## typeV06 -0.001501 0.207162 -0.007 0.9943   
## typeV07 -0.516118 0.190369 -2.711 0.0111 \*   
## typeV09 -0.327318 0.178297 -1.836 0.0767 .   
## typeV10 -0.664263 0.378225 -1.756 0.0896 .   
## typeV11 -0.508349 0.378225 -1.344 0.1894   
## typeV14 -0.946460 0.378225 -2.502 0.0182 \*   
## typeV18 -0.471054 0.378225 -1.245 0.2229   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.3654 on 29 degrees of freedom  
## Multiple R-squared: 0.4201, Adjusted R-squared: 0.2201   
## F-statistic: 2.101 on 10 and 29 DF, p-value: 0.05815