

Assignment 1

Group 47: Luca Cavellini, Paul Hosek, Robert Satzger

2/20/2023

Exercise 1. Trees

a)

```
trees = read.table("data/treeVolume.txt", header = T)
trees$type <- as.factor(trees$type)
trees_lm1 <- lm(volume ~ type, data = trees)
summary(trees_lm1)
```

```
##
## Call:
## lm(formula = volume ~ type, data = trees)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.97  -9.96  -2.77   5.94  46.83
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    30.17      2.54    11.88  <2e-16 ***
## typeoak         5.08      3.69     1.38    0.17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.1 on 57 degrees of freedom
## Multiple R-squared:  0.0322, Adjusted R-squared:  0.0153
## F-statistic:  1.9 on 1 and 57 DF,  p-value: 0.174
```

```
trees_aov <- anova(trees_lm1)
print(trees_aov)
```

```
## Analysis of Variance Table
##
## Response: volume
##              Df Sum Sq Mean Sq F value Pr(>F)
## type           1    380      380    1.9   0.17
## Residuals     57  11395      200
```

One-way Anova only makes sense with three or more levels. With two levels, p-value is the same as from t-test for difference of means between the two levels. Based on summary of lm, beech is 30.17 and oak is 5.08.

b)

```
trees_lm2 <- lm(volume ~ type * diameter + height, data = trees)
summary(trees_lm2)
```

```
##
## Call:
## lm(formula = volume ~ type * diameter + height, data = trees)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.350 -2.194 -0.141  1.701  8.176
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -63.873     5.539  -11.53 3.5e-16 ***
## typeoak         -4.963     5.149   -0.96  0.34
## diameter         4.608     0.207   22.26 < 2e-16 ***
## height          0.434     0.079    5.49 1.1e-06 ***
## typeoak:diameter 0.259     0.359    0.72  0.47
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.26 on 54 degrees of freedom
## Multiple R-squared:  0.951, Adjusted R-squared:  0.948
## F-statistic: 264 on 4 and 54 DF, p-value: <2e-16
```

```
anova(trees_lm2)
```

```
## Analysis of Variance Table
##
## Response: volume
##              Df Sum Sq Mean Sq F value    Pr(>F)
## type           1    380     380    35.77 1.8e-07 ***
## diameter       1  10492   10492   989.02 < 2e-16 ***
## height         1    324     324    30.56 9.6e-07 ***
## type:diameter  1      6       6     0.52  0.47
## Residuals     54    573      11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
trees_lm3 <- lm(volume ~ type * height + diameter, data = trees)
summary(trees_lm3)
```

```
##
## Call:
```

```
## lm(formula = volume ~ type * height + diameter, data = trees)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.230 -2.113 -0.161  1.801  8.165
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -57.551      7.111   -8.09   7e-11 ***
## typeoak       -17.471     11.826   -1.48   0.1454
## height         0.321      0.102    3.14   0.0027 **
## diameter       4.779      0.173   27.55  <2e-16 ***
## typeoak:height  0.212      0.154    1.37   0.1761
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.22 on 54 degrees of freedom
## Multiple R-squared:  0.953, Adjusted R-squared:  0.949
## F-statistic: 271 on 4 and 54 DF, p-value: <2e-16
anova(trees_lm3)
```

```
## Analysis of Variance Table
##
## Response: volume
##              Df Sum Sq Mean Sq F value    Pr(>F)
## type           1    380     380    36.67 1.4e-07 ***
## height          1   2239     2239   216.34 < 2e-16 ***
## diameter        1   8577     8577   828.64 < 2e-16 ***
## type:height     1     19       19     1.88   0.18
## Residuals      54    559       10
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Neither interaction effect seems to be present. Instead, use purely additive model.

```
trees_lm4 <- lm(volume ~ type + diameter + height, data = trees)
summary(trees_lm4)
```

```
##
## Call:
## lm(formula = volume ~ type + diameter + height, data = trees)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -7.186 -2.140 -0.087  1.721  7.701
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -63.7814      5.5129  -11.57  2.3e-16 ***
```

```
## typeoak      -1.3046      0.8779     -1.49      0.14
## diameter     4.6981      0.1645     28.56 < 2e-16 ***
## height       0.4172      0.0752      5.55 8.4e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.24 on 55 degrees of freedom
## Multiple R-squared:  0.951, Adjusted R-squared:  0.948
## F-statistic: 355 on 3 and 55 DF, p-value: <2e-16
```

```
anova(trees_lm4)
```

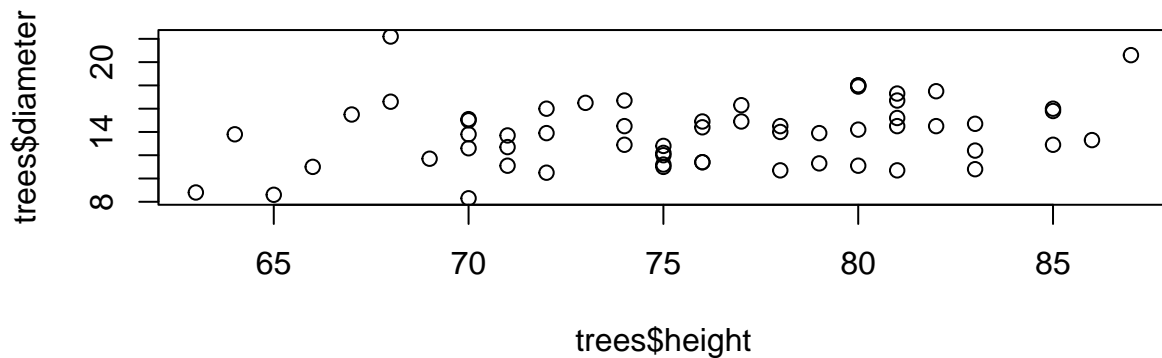
```
## Analysis of Variance Table
##
## Response: volume
##          Df Sum Sq Mean Sq F value    Pr(>F)
## type       1    380      380    36.1 1.6e-07 ***
## diameter   1  10492   10492   997.7 < 2e-16 ***
## height     1    324      324    30.8 8.4e-07 ***
## Residuals 55     578        11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

d) A cylinder's volume is given by $V = \pi * (d/2)^2 * h$. So most logical explanatory factor would be to calculate the volume using this formula.

```
trees$transform <- pi * (trees$diameter/2)^2 * trees$height
trees_lm5 <- lm(volume ~ transform, data = trees)
summary(trees_lm5)
```

```
##
## Call:
## lm(formula = volume ~ transform, data = trees)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.846 -1.343 -0.245  1.533  5.532
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.79e-01   7.63e-01   -0.5    0.62
## transform    2.73e-03   5.82e-05   46.9   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.28 on 57 degrees of freedom
## Multiple R-squared:  0.975, Adjusted R-squared:  0.974
## F-statistic: 2.2e+03 on 1 and 57 DF, p-value: <2e-16
```

```
plot(trees$height, trees$diameter)
```



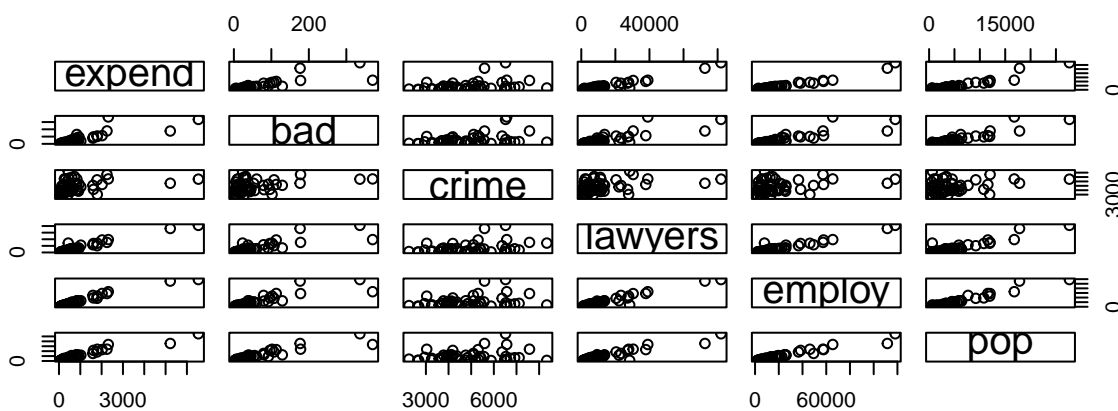
R^2 shows better than the other models.

Exercise 2. Expenditure on criminal activities

```
data <- read.table("data/expensescrime.txt", header = TRUE);
```

a)

```
# graphical summary?
pairs(data[,2:7])
```

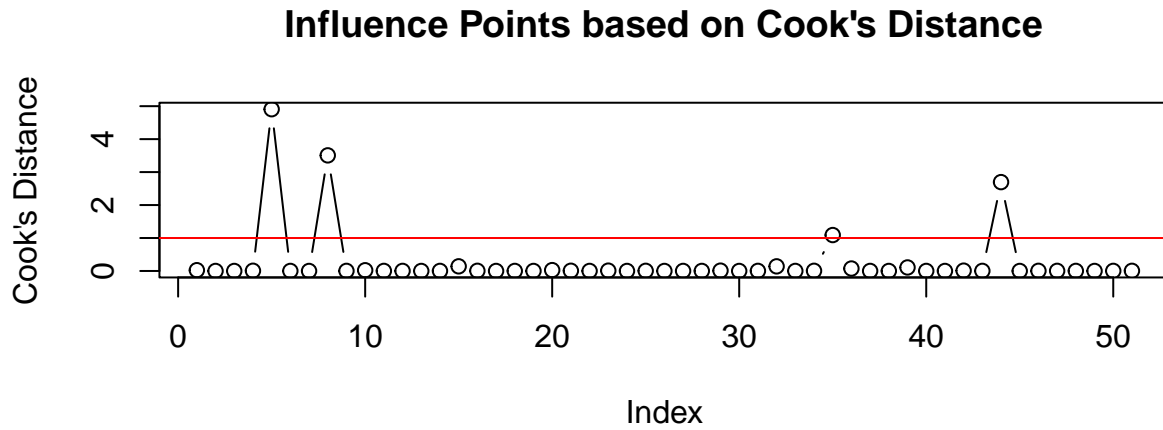


```
# Influence points
full_model <- lm(expend ~ bad + crime + lawyers + employ + pop, data = data)
cdist <- cooks.distance(full_model)
```

Table 1: VIF Values by term dropped from full model and Variable

	vif.bad	vif.crime	vif.lawyers	vif.employ	vif.pop
Population	4.46	1.30	16.8	21.1	NA
Employment	8.24	1.49	10.1	NA	20.7
Lawyers	8.07	1.31	NA	20.0	32.5
Crime Rate	NA	1.23	16.4	33.1	17.6

```
plot(1:length(cdist), cdist, xlab = "Index", ylab = "Cook's Distance",
     main = "Influence Points based on Cook's Distance", type="b")
abline(h=1, col="red")
```



```
# Collinearity
print(vif(full_model))
```

```
##      bad      crime lawyers  employ      pop
##  8.36    1.49   16.97   33.59   32.94
```

Our graphical summary of the data shows that the crime rate, lawyers, employment and population are most notably related with expenditure. “Crime” shows random variation with all other variables.

Analyses of Influence points based on Cook’s distance indicates that 4 observations with disproportionate influence on the model, with one marginally outside the cutoff of one. All four observations must undergo further inspection and potentially dropped from the model.

Concerning Colliniarity, our rule of thumb dicatates that a $VIF > 5$ is concerning, indicating $R^2 > .8$. In our dataset, all variables besides “crime” were found with $VIF > 5$, further “Pop” and “Employ” show extreme VIF values with $VIF > 30$. This indicates that at least one variable is a linear combination of the others. This corroborates the inpection of the scatter-matrix from before, indicating close relationships between the variables. We can try to drop a single term from the model, but may need to drop more.

Dropping the term with the highest vif did not reduce all VIFs to under five, as such we should use the step-down or step-up method to select the correct model.

b) Fit a linear regression model to the data using the step-up method.

This method can be summarized in 4 steps:

1. Build model with constant predictor (background model).
2. Find term that, if added to the model would maximise R^2 .
3. If the variable is significant, add it to the model.
4. Go to 2 until 3 does not occur.

```
# find the next best term that significantly improves model fit
find_best_term <- function(formula, data) {
  best_fit <- summary(lm(formula, data))$adj.r.squared
  best_term <- 1
  # maximize r^2
  for (term in names(data[, -c(1:2)])) {
    if (!term %in% attr(formula, "term.labels")) {
      cur_formula <- paste(formula, term, sep="+")
      cur_model <- lm(cur_formula, data = data)
      cur_fit <- summary(cur_model)$r.squared
      if (cur_fit > best_fit) {
        best_fit <- cur_fit
        best_term <- term
        best_coef_pval <- summary(cur_model)$coefficients[
          nrow(summary(cur_model)$coefficients), "Pr(>|t|)"]
      }
    }
  }
  # test if increase is significant and return new formula
  if (best_coef_pval < 0.05) {
    return(best_term)
  } else {
    return(1)
  }
}

# recursively add terms
step_up <- function(formula, data) {
  new_term <- find_best_term(formula, data)
  if (new_term == 1) {
    return(formula)
  }
  step_up(paste(formula, new_term, sep="+"), data)
}

best_formula <- step_up("expend~1", data)
best_model <- lm(best_formula, data=data)
print(summary(best_model))
```

##

```
## Call:
## lm(formula = best_formula, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -599.5   -94.4    36.0    92.0   936.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.11e+02   4.26e+01  -2.60   0.0124 *
## employ       2.97e-02   5.11e-03   5.81  4.9e-07 ***
## lawyers      2.69e-02   7.76e-03   3.46  0.0011 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 233 on 48 degrees of freedom
## Multiple R-squared:  0.963, Adjusted R-squared:  0.962
## F-statistic: 628 on 2 and 48 DF,  p-value: <2e-16

print(vif(best_model))

## employ lawyers
##      14.8      14.8
```

The model found includes only “employ” and “lawyers”. It explains a high proportion of the variance ($R^2 = 0.96$). Analysis of collinearity found all VIF values larger than five. This model may not be the most suitable model as factors linearly depend on another.

c)

```
hypo_state= data.frame(bad = 50, crime = 5000, lawyers = 5000, employ = 5000, pop = 5000)

pred_interv_b <- predict(best_model,hypo_state,interval="prediction",level=0.95)
```

Given this model, we predict $p = 172.21$ for “expend”. 95% and lower bounds are: $[-302.93, 647.35]$. We could improve this prediction by trying out different models (e.g., add more terms) and examine if the prediction interval becomes smaller. For example, we could try the full model from earlier:

```
pred_interv_f <- predict(full_model,hypo_state,interval="prediction",level=0.95)
```

The prediction interval of the full model is larger (1003.18) than using the model found with the step-up method (950.28). Alternatively, assuming “improving” the interval means making it smaller, we could lower the confidence level.

d) Lasso method

```
x <- as.matrix(data[,-2:-1]) # remove expend and state
y <- as.double(as.matrix(data[,2])) # expend is response
```



```

train=sample(1:nrow(x),0.67*nrow(x))
x.train=x[train,]; y.train=y[train]
x.test=x[-train,]; y.test=y[-train]

lasso.mod=glmnet(x.train,y.train,alpha=1)
lasso.cv=cv.glmnet(x.train,y.train,alpha=1,type.measure="mse")
par(mfrow=c(1,2))

lambda.min=lasso.cv$lambda.min; lambda.1se=lasso.cv$lambda.1se
coef(lasso.mod,lasso.cv$lambda.min) #beta's for the best lambda

```

```

## 6 x 1 sparse Matrix of class "dgCMatrix"
##              s1
## (Intercept) -294.8115
## bad         -1.8987
## crime        0.0632
## lawyers      0.0045
## employ       0.0203
## pop          0.0858

```

```

y.pred=predict(lasso.mod,s=lambda.min,newx=x.test) #predict for test
mse.lasso=mean((y.test-y.pred)^2) #mse for the predicted test rows

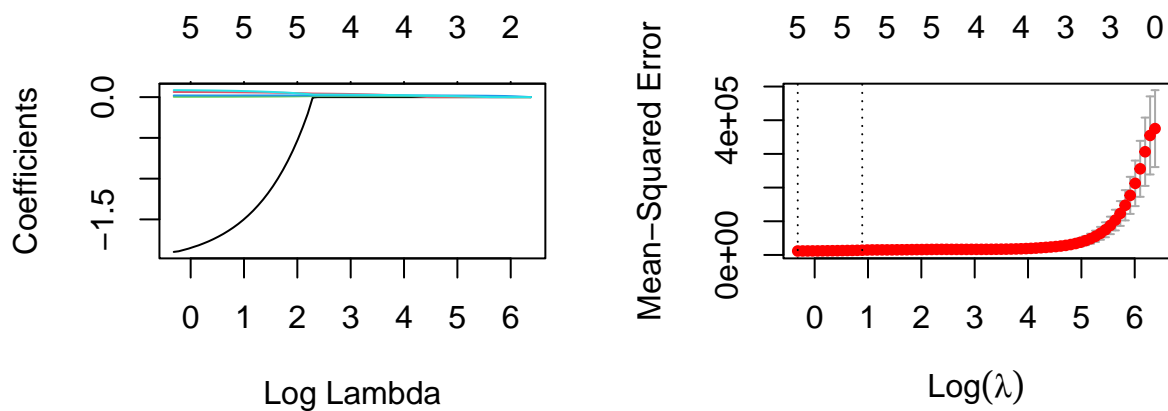
```

```

plot(lasso.mod,label=T,xvar="lambda") #have a look at the lasso path
plot(lasso.cv) # the best lambda by cross-validation

```

idk if we need



```

# plot(lasso.cv$glmnet.fit,xvar="lambda",label=T) # same as before
print(mse.lasso)

```

```
## [1] 336962
```

```

lass_model = lm("expend~bad", data=data)
summary(lass_model)

```

```
##
## Call:
## lm(formula = "expend~bad", data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2744.8  -130.0   -69.0    91.8  2739.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   126.72     114.86     1.1    0.28
## bad           13.32       1.26    10.6 2.8e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 661 on 49 degrees of freedom
## Multiple R-squared:  0.696, Adjusted R-squared:  0.69
## F-statistic: 112 on 1 and 49 DF, p-value: 2.8e-14
```

The model chosen by the LASSO method only includes “bad” (i.e., the crime rate) as a predictor. This model is much more parsimonious than the model chosen in b, as it only includes a single predictor (Occam’s Razor). However, this model explains $R^2 = 0.70$ of the variance, which is lower than the variance explained by our previous model ($R^2 = 0.96$).

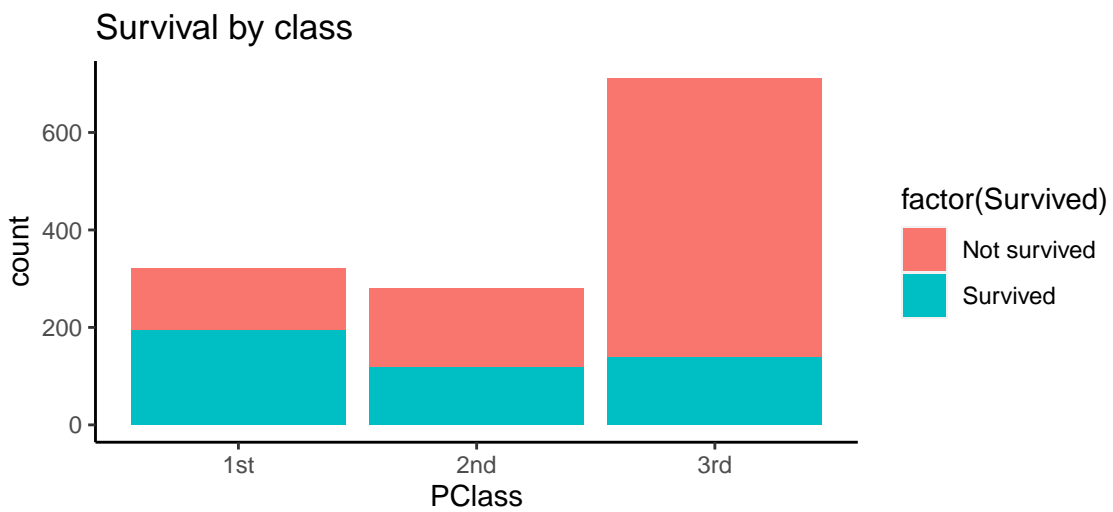
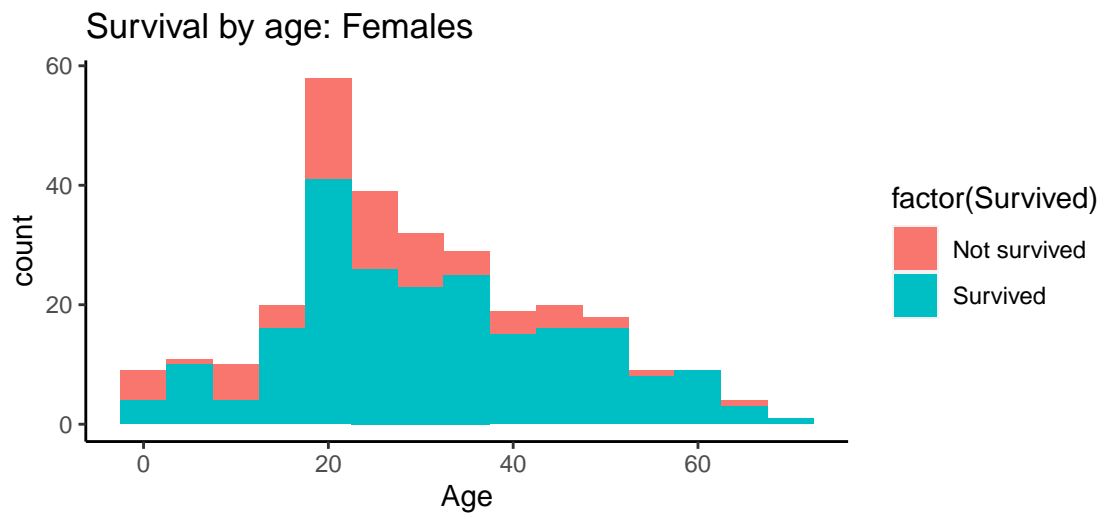
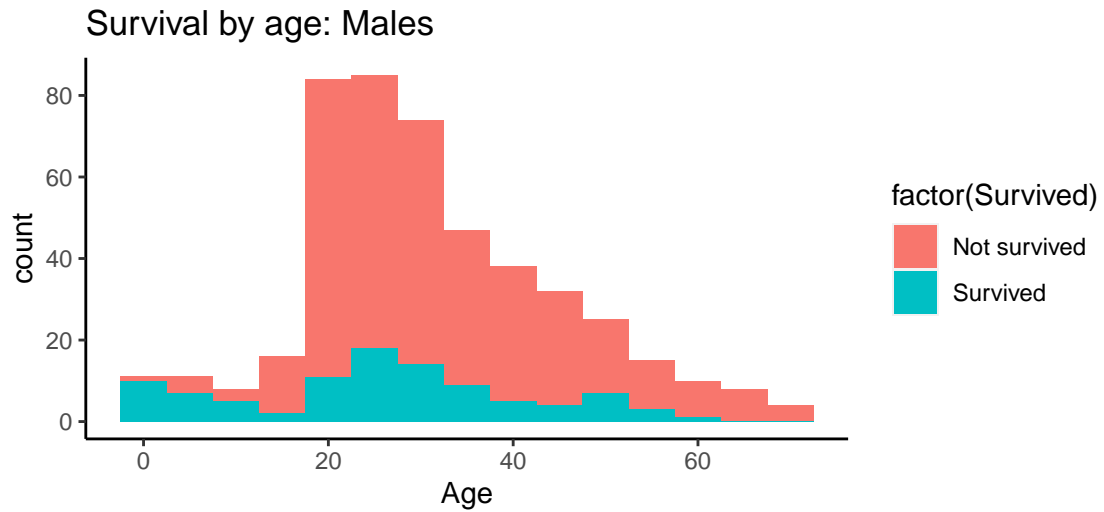
- models will only work with !nan ages

Exercise 3. Titanic

a)

- multiple summaries of data
- fit log regression,, w/o interactions -> survival & predictor Pclass, age, sex

```
data_titanic <- read.table("data/titanic.txt", header=TRUE)
data_titanic$PClass <- as.factor(data_titanic$PClass)
data_titanic$Sex <- as.factor(data_titanic$Sex)
data_titanic$Survived <- as.factor(data_titanic$Survived)
```



```
model_log1 <- glm(Survived ~ PClass + Age + Sex, data = data_titanic, family = binomial())
summary(model_log1)
```

```
##
## Call:
## glm(formula = Survived ~ PClass + Age + Sex, family = binomial(),
##      data = data_titanic)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.723  -0.707  -0.392   0.649   2.529
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.75966    0.39757   9.46 < 2e-16 ***
## PClass2nd    -1.29196    0.26008  -4.97 6.8e-07 ***
## PClass3rd    -2.52142    0.27666  -9.11 < 2e-16 ***
## Age         -0.03918    0.00762  -5.14 2.7e-07 ***
## Sexmale     -2.63136    0.20151 -13.06 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 1025.57  on 755  degrees of freedom
## Residual deviance:  695.14  on 751  degrees of freedom
## (557 observations deleted due to missingness)
## AIC: 705.1
##
## Number of Fisher Scoring iterations: 5
```

Excluding interaction effects, we find that being a female or a first class passengers or young increases your odds of survival. However, we cannot know how a combination of these will impact the odds. From the main effects we can conclude: Males are 13.89 more likely to die compared to females. 2nd-class passengers are 3.64 and third-class passengers are 12.45 as likely to die than passengers in other classes (calculated as $1/\exp(\text{coefficient_of_interest})$). Further, for each year a person is older, odds decrease by a factor of 0.96: younger passengers are more likely to survive (calculated as $\exp(\text{age})$). All these main effects are statistically significantly associated with survival.

b)

```
model_log2 <- glm(Survived ~ PClass + Age + Sex + PClass:Age + Age:Sex, data = data_titanic, family = binomial())
summary(model_log2)
```

```
##
## Call:
## glm(formula = Survived ~ PClass + Age + Sex + PClass:Age + Age:Sex,
##      family = binomial, data = data_titanic)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

Table 2: Survival probability for 55 year olds.

PClass	Sex	Age	Survival_Probability
1st	female	55	0.967
2nd	female	55	0.667
3rd	female	55	0.619
1st	male	55	0.179
2nd	male	55	0.015
3rd	male	55	0.012

```
## -2.686 -0.646 -0.339 0.675 2.727
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.0299    0.6598   3.08  0.0021 **
## PClass2nd    -0.2115    0.7101  -0.30  0.7658
## PClass3rd    -2.0811    0.6658  -3.13  0.0018 **
## Age          0.0246    0.0198   1.25  0.2131
## Sexmale     -0.3889    0.4803  -0.81  0.4180
## PClass2nd:Age -0.0451    0.0220  -2.05  0.0401 *
## PClass3rd:Age -0.0148    0.0211  -0.70  0.4834
## Age:Sexmale  -0.0821    0.0171  -4.81  1.5e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##    Null deviance: 1025.57  on 755  degrees of freedom
## Residual deviance:  662.45  on 748  degrees of freedom
## (557 observations deleted due to missingness)
## AIC: 678.4
##
## Number of Fisher Scoring iterations: 5
all_comb_55 <- expand.grid(PClass = levels(data_titanic$PClass), Sex = levels(data_titanic$Sex),
all_comb_55$Survival_Probability <- predict(model_log2, all_comb_55, type = "response") # response
kable(all_comb_55, format = "latex", caption = "Survival probability for 55 year olds.")
```

We observe that being female has the largest influence on survival. Independent of gender, more expensive classes have larger survival probability. We observe that females in the first class have a extremely high survival probability of 0.97.

c)

- method to predict
 - logistic regression
 - MCMC
 - machine learning

- idk bro lecture is not out yet
- measure of quality
 - validation set
 - area under ROC

We can use the estimated logistic regression model to predict the probability of survival for a new observation, and then apply a threshold to classify the observation as either a survivor (1) or a non-survivor (0).

Here, we first fit the logistic regression model to the observed data to estimate $\hat{\theta}$ in $P(Y_k = 1) = \frac{1}{1+e^{-x_k^T \hat{\theta}}}$, $k = 1, \dots, N$. Then, we use this estimate to predict the probability of survival for a new passenger with predictor values X_{new} . We then apply a threshold p_0 to classify the new passenger into survivor (1) or a non-survivor (0). Specifically, whether the predicted probability \hat{P}_{new} is above or below the threshold is used to classify a new passenger.

The threshold p_0 determines the trade-off between sensitivity and specificity of our model. We may choose a validation set and some quality measure (e.g., accuracy: predictions correct) to maximize on this data set. Note, however, that this quality measure should be chosen depending on what our goal is, if we want high sensitivity or high specificity, maximizing these may also guide the threshold.

d)

```
ct_class <- xtabs(~ PClass + Survived, data = data_titanic)
ct_class
```

```
##      Survived
## PClass    0    1
##   1st 129 193
##   2nd 161 119
##   3rd 573 138
```

```
xtest_class <- chisq.test(ct_class)
xtest_class
```

```
##
## Pearson's Chi-squared test
##
## data:  ct_class
## X-squared = 172, df = 2, p-value <2e-16
```

```
ct_sex <- xtabs(~ Sex + Survived, data = data_titanic)
ct_sex
```

```
##      Survived
## Sex        0    1
## female 154 308
## male   709 142
```

```
# chisq.test(ct_sex)
xtest_sex <- fisher.test(ct_sex)
```

```
xtest_sex
```

```
##
## Fisher's Exact Test for Count Data
##
## data:  ct_sex
## p-value <2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.0762 0.1316
## sample estimates:
## odds ratio
##          0.1
```

Here, we find that both class and sex have significant p-values. This indicates, that survival odds are not independent of neither sex nor class. The tables of residuals below show that the higher the class, relatively more people survive. The same is true for females compared to males.

```
residuals(xtest_class)
```

```
##          Survived
## PClass      0      1
## 1st -5.68  7.87
## 2nd -1.70  2.35
## 3rd  4.89 -6.77
```

```
residuals(chisq.test(ct_sex))
```

```
##          Survived
## Sex      0      1
## female -8.59 11.89
## male   6.33 -8.76
```

e)

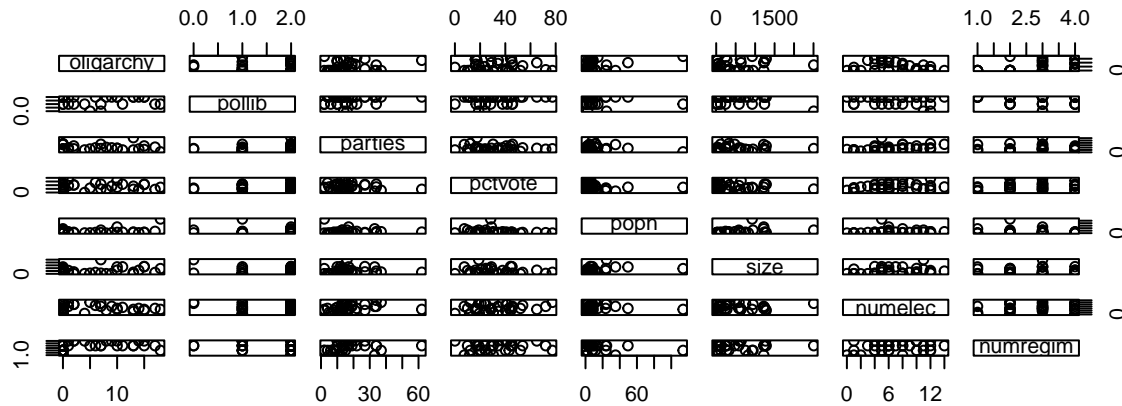
The approach in d) is not wrong, but has limited interpretability. Specifically, the contingency table is limited by its simplicity. It does not account for confounding variables (e.g., age in this case), leading to potentially false conclusions. This is related to the fact that no continuous predictors can be added to this model. Further, this approach does not provide us with a strength of the association between predictor and outcome. Logistic regression on the other hand, is able to account for multiple predictors simultaneously and estimate the magnitude and direction of predictor-outcome relationships. This may be further built upon to predict new data using a machine learning based on some quality criterion. However, logistic regression is more complex and less intuitive than a contingency table, so to guide hypothesis, it may be better to use a contingency table.

Exercise 4. Military Coups

a)

First, we test for collinearity between the explanatory variables in the dataset.

```
coups <- read.table("data/coups.txt", header = T)
pairs(coups[,-1])
```



There is no clear relationship between any of the explanatory variables. Thus, we can assume there is no issue with collinearity and we include all explanatory variables in the Poisson regression.

For simplicity, we treat all discrete numerical variables as continuous. However, we convert the categorical variable pollib to a factor.

```
coups$pollib <- as.factor(coups$pollib)
coupsglm <- glm(miltcoup~., family = poisson, data = coups)
drop1(coupsglm, test = "Chisq")
```

```
## Single term deletions
##
## Model:
## miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##      numelec + numregim
##      Df Deviance AIC  LRT Pr(>Chi)
## <none>          28.2 113
## oligarchy  1      32.4 115 4.10   0.0428 *
## pollib     2      35.6 116 7.33   0.0256 *
## parties    1      35.3 118 7.06   0.0079 **
## pctvote    1      30.6 113 2.32   0.1275
## popn       1      30.6 113 2.35   0.1252
## size       1      29.2 112 0.99   0.3202
## numelec    1      28.4 111 0.18   0.6705
## numregim   1      29.1 112 0.81   0.3681
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The poisson regression suggests that only the explanatory variables of oligarchy, pollib, and parties significantly influence the odds of a successful military coup. As shown by the positive signs of the coefficients below, oligarchy and parties are both estimated to increase the odds of successful

military coups. Compared to no civil rights (“pollib0”), having limited (“pollib1”) or full (“pollib2”) civil rights decreases the odds of coups.

```
summary(coupsglm)$coefficient[c("(Intercept)", "oligarchy", "pollib1", "pollib2", "parties"), "Est."]
## (Intercept)      oligarchy      pollib1      pollib2      parties
##      -0.2334       0.0726      -1.1032      -1.6903       0.0312
```

b)

To ensure interpretability of the summary output, we treat the variable pollib as continuous during the step-down procedure. This is possible as pollib is an ordinal variable: Increasing levels of pollib correspond to increasing levels of political liberalization of a country. Once the step-down procedure is completed, we treat pollib categorically again to compare model fit.

```
model_df <- coups # create copy for step-down selection
model_df$pollib <- as.numeric(model_df$pollib)

done <- F
while (!done){
  model <- glm(miltcoup ~ ., family = poisson, data = model_df)
  model_smry <- summary(model)
  print(model_smry$call)
  print(model_smry$coefficients)

  model_smry_coefs <- model_smry$coefficient[-1, "Pr(>|z|)"] # ignore intercept
  is_insignificant <- model_smry_coefs > .05
  if (sum(is_insignificant) == 0) done = T
  else {
    excl_var <- model_smry_coefs[which.max(model_smry_coefs)]
    # if (substr(names(excl_var), 1, 6) == "pollib"){
    #   names(excl_var) <- "pollib"
    #   print("WARNING POLLIB")
    # }
    writeLines(paste("\nExcluding variable:", names(excl_var), "\n"))
    model_df <- model_df[, -which(names(model_df) == names(excl_var))]
  }
}

## glm(formula = miltcoup ~ ., family = poisson, data = model_df)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20271    1.049314   0.193  0.84682
## oligarchy    0.07308    0.034596   2.112  0.03465
## pollib      -0.71298    0.272563  -2.616  0.00890
## parties      0.03077    0.011187   2.751  0.00595
## pctvote      0.01387    0.009753   1.422  0.15491
## popn         0.00934    0.006595   1.417  0.15658
## size        -0.00019    0.000248  -0.765  0.44447
## numelec     -0.01608    0.065484  -0.246  0.80605
## numregim     0.19173    0.229289   0.836  0.40303
```

```
##
## Excluding variable: numelec
##
## glm(formula = miltcoup ~ ., family = poisson, data = model_df)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.069587   0.906928  0.0767  0.93884
## oligarchy    0.078137   0.027766  2.8142  0.00489
## pollib      -0.677390   0.229013 -2.9579  0.00310
## parties      0.029679   0.010289  2.8846  0.00392
## pctvote      0.013129   0.009289  1.4133  0.15756
## popn         0.008931   0.006375  1.4011  0.16120
## size        -0.000202   0.000244 -0.8295  0.40682
## numregim     0.175820   0.221050  0.7954  0.42639
##
## Excluding variable: numregim
##
## glm(formula = miltcoup ~ ., family = poisson, data = model_df)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.576716   0.632427  0.912  0.361817
## oligarchy    0.085962   0.025910  3.318  0.000908
## pollib      -0.689403   0.227857 -3.026  0.002481
## parties      0.029194   0.010195  2.863  0.004190
## pctvote      0.014159   0.009198  1.539  0.123723
## popn         0.006274   0.005399  1.162  0.245272
## size        -0.000195   0.000242 -0.804  0.421378
##
## Excluding variable: size
##
## glm(formula = miltcoup ~ ., family = poisson, data = model_df)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.40836    0.59923  0.681  0.49557
## oligarchy    0.08317    0.02544  3.270  0.00108
## pollib      -0.65283    0.22123 -2.951  0.00317
## parties      0.02980    0.01029  2.895  0.00379
## pctvote      0.01384    0.00928  1.491  0.13591
## popn         0.00559    0.00538  1.039  0.29883
##
## Excluding variable: popn
##
## glm(formula = miltcoup ~ ., family = poisson, data = model_df)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.5730     0.56352  1.02 3.09e-01
## oligarchy    0.0954     0.02242  4.25 2.11e-05
## pollib      -0.6666     0.21756 -3.06 2.18e-03
## parties      0.0256     0.00950  2.70 6.99e-03
## pctvote      0.0121     0.00906  1.34 1.80e-01
##
## Excluding variable: pctvote
```

```
##
## glm(formula = miltcoup ~ ., family = poisson, data = model_df)
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.8255    0.52763   1.56 1.18e-01
## oligarchy     0.0926    0.02178   4.25 2.11e-05
## pollib       -0.5741    0.20438  -2.81 4.97e-03
## parties       0.0221    0.00896   2.46 1.38e-02
```

The step-down procedure corroborates our findings from a. Both approaches suggest that the most sensible additive model would contain only the three variables of oligarchy, pollib, and parties.

We can compare this reduced model to the model full model from a. Here we treat pollib categorically again.

```
coupsglm_reduced <- glm(miltcoup~oligarchy+pollib+parties, family = poisson, data = coups)
comparison <- anova(coupsglm_reduced, coupsglm, test = "Chisq")
comparison
```

```
## Analysis of Deviance Table
##
## Model 1: miltcoup ~ oligarchy + pollib + parties
## Model 2: miltcoup ~ oligarchy + pollib + parties + pctvote + popn + size +
##         numelec + numregim
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         31        32.8
## 2         26        28.2  5     4.57    0.47
```

Excluding five of the eight explanatory variables from the model did not yield a significant drop in model fitness with $p = 0.47$. This shows that we have successfully reduced model complexity without significantly sacrificing fitness.

c)

```
obs <- data.frame(pollib = factor(0:2), oligarchy = mean(coups$oligarchy), parties = mean(coups$parties))
y_hat <- predict(coupsglm_reduced, obs, type="response")
names(y_hat) <- c("0", "1", "2")
print(y_hat)

##      0      1      2
## 2.908 1.772 0.956
```

As could be expected, the number of successful military coups seems to be negatively associated with the level of political liberalization. For a country with an average duration of oligarchy rule and an average number of political parties, the **expected number** of military coups is 2.91, 1.77, and 0.96 for levels 0,1, and 2, of political liberalization, respectively.