Christina De Cesaris University of California, Davis

Sta 223 Project 1

Analysis of Characteristics Influencing Mushroom Edibility

Abstract

Mushrooms are diverse, complex structures with a multitude of overlapping and contrasting characteristics between species. To better understand how mushroom characteristics affect edibility, a previously unanalyzed dataset containing 173 detailed mushroom profiles was processed and fit with a logistic regression model using edibility status as a binary outcome (Wagner et al., 2021). Model inference revealed mushrooms with white stems, winter growing season, and brown caps have a lower probability of being inedible, while mushrooms with bell shaped caps, and green caps have a higher chance of being inedible. The overall model fit was sufficient, but future studies would benefit from datasets with increased sample sizes across all predictors.

Introduction

Recent genomic research estimates there are between three to five million different species of fungi, making one of the most diverse organisms on earth (Blackwell , 2011). However, despite the high estimates, mycologists have recorded only 144,000 in literature, and roughly 9.7% of these recorded fungi species form fruiting bodies, the spore producing organs commonly known as mushrooms (Casadevall et al., 2008). Most mushrooms are indigestible by humans and a small percent are poisonous. Expanding research in potential medicinal benefits of mushrooms has led to an increase in foraging, and since mushrooms are complex structures often with shared characteristics, they can be difficult to identify (Wasser, 2017). Using a dataset containing 173 descriptive profiles of mushrooms, this project seeks to investigating the traits of mushrooms which influence edibility to humans.

The mushrooms dataset contains 77 edible and 96 inedible mushroom profiles, initially labeled as poisonous in the dataset but described as mostly inedible from the documentation (Wagner et al., 2021). The profiles were parsed from the reference guide *Mushrooms and Toadstools* and this primary dataset was used to generate a secondary theoretical dataset for prediction models (Wagner et al., 2021). Using the unanalyzed primary dataset, this project seeks to answer which features of a mushroom, if any, influence whether it is edible to humans?

The raw data consists of seventeen categorical variables describing the color, texture, and shape of the parts of the mushroom and three numerical variables containing the ranges for stem width, stem height, and cap diameter. Categorical variables are represented as strings of the character codes and can vary in length between samples (Table A1). Two additional variables provided information on the family and common name of the mushroom. A complete list of all variables and descriptions can be found in

Table A2, and better understand the terminology described by the variables a diagram of a mushroom is provided in Figure A1.

Methods

Data Preprocessing

Due to missingness, not all variables were suitable for analysis, and variables with missing values over 15% were dropped from consideration (Table A3). The brackets surrounding each entry were removed along with excess white space before variables were processed. The numerical variables, containing the range of min and max values, were separated into two variables, and all measurements were converted to centimeters.

Categorical variables were divided into binary representing the presence of every possible code. Codes with all zero entries were dropped. The outcome was converted to binary where 1 represented inedibility status, and the associated class variables of *family* and species *name* were removed. An example of data encoding for the *season* and *cap.diameter* variables can be found in Table 1.

The processed dataset contained 70 predictors in which 6 were numeric measurements. Numeric values were plotted and compared by *class* (Figures A2 & A3). Charts for key categorical variables were grouped by *class* to better understand the proportion of data each attribute. It is important to recognize that pie charts can often appear misleading, so it is crucial to pay attention to the number of samples belonging to each variable (Figures A4, A5, A6 & A7).

cap.diameter	season	cap.diameter_max	cap.diameter_min	season_a	season_s	season_u	season_w
[10, 20]	[u, a, w]	20	10	1	0	1	1
[5, 10]	[u, a]	10	5	1	0	1	0
[10, 15]	[u, a]	15	10	1	0	1	0

Table 1: Preprocessing example. Raw columns are unhighlighted, red columns represent the encoded categorical variables, blue columns represent the split numeric variables

Model Methods

The binary outcome *class* makes this analysis suitable for a logistic regression model with a logit link function, in order to better understand which predictors influence edibility status. The model can be defined as (Faraway, 2016):

$$\text{Link Function: } \eta = log\left(\frac{p(y)}{1 - p(y)}\right)$$

$$Logistic\ \text{Regression}: \eta = \beta_1 X_1 + \cdots + \beta_p X_p\ for\ p\ predictors$$

Model selection was performed by first fitting the full model with the seventy predictors. Backwards, stepwise selection was performed on the full model using AIC and BIC criteria, and the resulting models were the same. The model contained the 6 categorical predictors in the equation which follows.

$$\eta_{BIC} = \beta_0 + \beta_1 season_{winter} + \beta_2 cap.shape_{bell} + \beta_3 cap.color_{brown} + \beta_4 cap.color_{green} + \beta_5 stem.color_{white} + \beta_6 ring.type_{zone}$$

Assessment of Fit

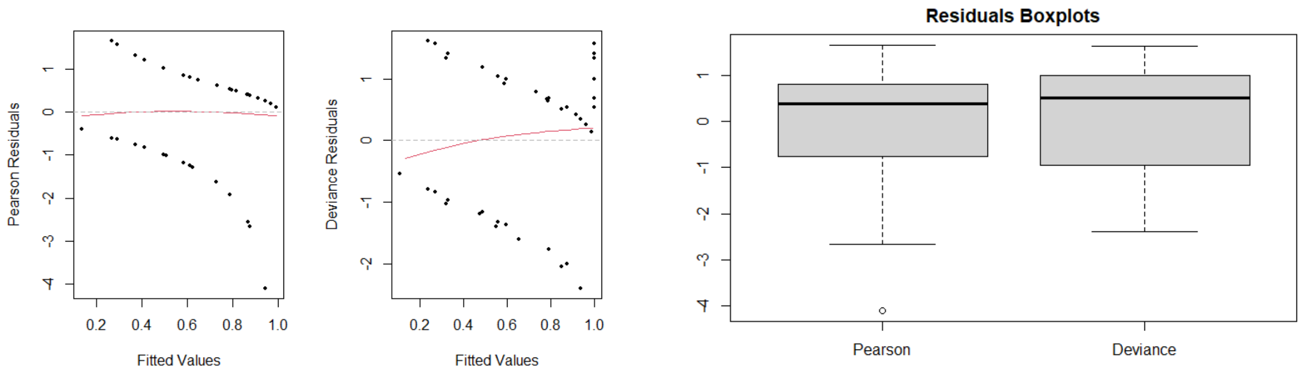

Figure 2: Deviance and Pearson residuals vs fitted values and boxplot residuals

The Deviance and Pearson residuals for the BIC model follow approximately the same distribution (Figure 2). When plotted against the fitted values, the Pearson and Deviance residuals display no clear pattern, but the Deviance residuals appear to follow a weak pattern, and therefore do not indicate a lack of fit.

Results

	Estimate	**Std. Error**	**P Value**	**CI 5%**	**CI 95%**
Intercept	-1.3543	0.4062	0.000857 ***	-2.0492121	-0.7076126
$\mathbf{season_{winter}}$	1.1118	0.4199	0.008102 **	0.4348772	1.8227207
$\mathbf{cap.shape_{bell}}$	-1.3548	0.6118	0.026802 *	-2.4295820	-0.3955981
$\mathbf{cap.color_{brown}}$	0.9534	0.3970	0.016322 *	0.3148581	1.6255934
$\mathbf{cap.color_{green}}$	-2.0067	0.8498	0.018203 *	-3.6208862	-0.7421159
$\mathbf{stem.color_{white}}$	1.3908	0.3828	0.000280 ***	0.7760159	2.0398872
$\mathbf{ring.type_{zone}}$	-16.6113	869.5533	0.984759	NA	47.6069152
Null deviance: 237.74 on 172 df			**Residual deviance:** 195.24 on 166 df		

AIC: 209.24			

Table 2: AIC/BIC Backwards selected model summary

For the BIC model, all estimates except $\mathbf{ring.type_{zone}}$ are statistically significant at the level $\alpha = 0.05$ (Table 2). The $\mathbf{ring.type_{zone}}$ estimator has a large standard error resulting in an uninterpretable the lower 95% confidence interval. Analysis of the $\mathbf{ring.type_{zone}}$ variable reveals its inclusion in the model is influenced by the small sample size of mushrooms with $\mathbf{ring.type_{zone}}$ which are all classified as edible (Table 3). The grouped pie charts for each predictor likewise reveal this imbalance in $\mathbf{ring.type_{zone}}$ (Figure 3). Further supported by the large p_{value}, $\mathbf{ring.type_{zone}}$ was deemed insignificant dropped as a predictor in the model resulting in the final model.

	Zone Ring Type	**No Zone Ring Type**
Edible	6	77
Inedible	0	90

Table 3: Frequency table for $ring.type_{zone}$

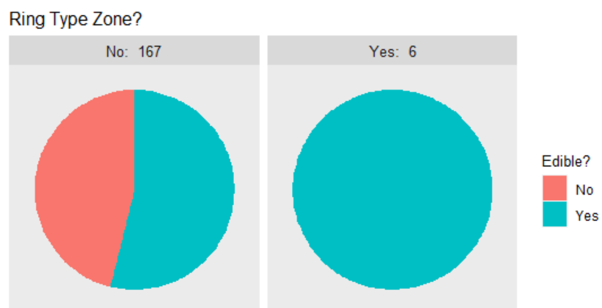

Figure 3: Proportion charts for $ring.type_{zone}$

Final Model

$$\eta_{final} = 1.3156 - 0.9822season_{winter} + 1.5043cap.shape_{bell} - 0.8634cap.color_{brown}$$
$$+1.8991\,cap.color_{green} + 1.3404\,stem.color_{white}$$

	Estimate	**Std. Error**	**P Value**	**CI 5%**	**CI 95%**
Intercept	1.3156	0.3915	0.000778 ***	0.6913500	1.9840120
$\mathbf{season_{winter}}$	-0.9822	0.4011	0.014345 *	-1.6573755	-0.3323078
$\mathbf{cap.shape_{bell}}$	1.5043	0.6010	0.012317 *	0.5676204	2.5649473
$\mathbf{cap.color_{brown}}$	-0.8634	0.3827	0.024068 *	-1.5098363	-0.2465492
$\mathbf{cap.color_{green}}$	1.8991	0.8395	0.023693 *	0.6493048	3.4973006
$\mathbf{stem.color_{white}}$	-1.3404	0.3688	0.000279 ***	-1.9639632	-0.7466615
Null deviance: 237.74 on 172 df			**Residual deviance:** 203.69 on 167 df		
AIC: **215.69**					

Table 4: Final model summary

Outlier Removal

Points 20 and 30 were identified as influential points with large cook distance (Figure A9). These points were removed and the model refit. There was no significant improvement in the lack of fit or estimated values (Figure A10).

Regression Effect Test

$$H_0: \beta_1 = \beta_2 = \cdots = \beta_p$$

$$H_a: At\ least\ one\ \beta_i \neq 0, \text{where } i \in [1, p]$$

$$p_{value} = 2.329 \text{ x } 10^{-06}$$

The p_{value} is less than the test level $\alpha = 0.05$, allowing us to reject the null hypothesis. Therefore, we can conclude that there is at least one predictor in the model which has a significant effect on the outcome of whether a mushroom is edible.

Discussion

In the final model, the predictors for $\mathbf{season_{winter}}$, $\mathbf{cap.shape_{bell}}$, $\mathbf{cap.color_{brown}}$, and $\mathbf{cap.color_{green}}$ are significant to the level $\alpha = 0.05$. The $\mathbf{stem.color_{white}}$ predictor is most significant at the level of $\alpha = 0.001$. The high significance is possibly explained by the balance of samples which have white stems. From Figure 8A, the ratio of mushrooms which have white stems to those which do not is 74:99, and significantly more balanced than any of the other predictors in the model. Extensions of this study should acquire additional data, if possible, to maximize the number of samples, in order to improve the interpretability of influential mushroom characterizes.

The $\mathbf{stem.color_{white}}$ predictor is fit with a negative estimate indicating that mushrooms that develop white stems are less likely to be inedible compared to those without white stems. The $\mathbf{cap.shape_{bell}}$ predictor has a positive value and therefore implies mushrooms with bell shaped caps have an increased probability of being inedible compared to those without. Likewise mushrooms with green caps have a greater chance of being inedible. Mushrooms which grow during the winter ($\mathbf{season_{winter}}$) have a decreased probability of being inedible in comparison to those which never grow in the winter. Finally, mushrooms with brown caps ($\mathbf{cap.color_{brown}}$) are also less likely to be inedible than those without brown caps. In summary mushrooms with white stems, winter growing season, and brown caps have a lower probability of being inedible, while mushrooms with bell shaped caps, and green caps have a higher chance of being inedible according to the final model.

As mentioned earlier, a major downside in this dataset is the small sample size, especially among less common characteristics. It is possible that the string structure of the raw data codes is partially to blame for the presence of characteristics with small sample sizes. None of the numeric variables were significant in the model, values representing the median or mean numeric measurement would be preferable in future studies.

References

Blackwell M. (2011). The fungi: 1, 2, 3 ... 5.1 million species ?. *American journal of botany*, *98*(3), 426–438. https://doi.org/10.3732/ajb.1000298

Casadevall, A., Heitman, J., & Buckley, M. (2008). The Fungal Kingdom: Diverse and Essential Roles in Earth's Ecosystem.

Faraway, J. (2016). Extending the linear model with R. Second Edition, Chapman and Hall. ISBN 9781498720960

Harding, P., (2013). *Mushroms and Toadstools*. Dorling Kindersley.

Wagner, D., Heider, D., & Hattab, G. (2021). Mushroom data creation, curation, and simulation to support classification tasks. *Scientific reports*, *11*(1), 8134. https://doi.org/10.1038/s41598-021-87602-3

Wasser S. P. (2017). Medicinal Mushrooms in Human Clinical Studies. Part I. Anticancer, Oncoimmunological, and Immunomodulatory Activities: A Review. *International journal of medicinal mushrooms*, *19*(4), 279–317. https://doi.org/10.1615/IntJMedMushrooms.v19.i4.10

Appendix A

Additional Tables and Figures

Variable Name	Description
family	Taxomic Family
name	Species Name
class	Edibility Status
cap.diameter	Diameter of Mushroom Cap (cm)
cap.shape	Mushroom Cap Shape
Cap.surface	Cap Surface Texture
cap.color	Color of Cap
does.bruise.or.bleed	Does Mushroom Bruise or Bleed (t/f)
gill.attachment	Gill Attachment Present
gill.spacing	Gill Spacing Type
gill.color	Gill Color
stem.height	Stem Height (cm)
stem.width	stem.width (mm)
stem.root	Stem Root Type
stem.surface	Stem Surface Type
stem.color	Stem Color
veil.type	Veil Type
veil.color	Veil Color
has.ring	Is a Ring Present (t/f)
ring.type	Type of Ring
Spore.print.color	Color of Spores
habitat	Native Habitat
season	Growing Season

Table A1: List of all raw variables and descriptions

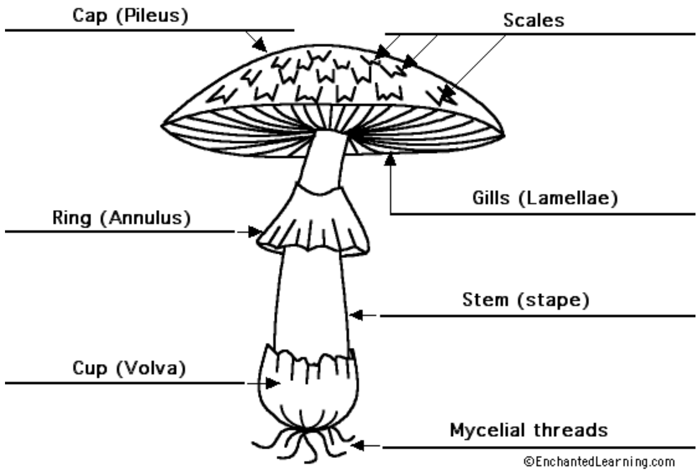

Figure A1: Anatomy of a fruiting body

Variable	Percent Missing
family	0.000000
name	0.000000
class	0.000000
cap.diameter	0.000000
cap.shape	0.000000
Cap.surface	**23.121387**
cap.color	0.000000
does.bruise.or.bleed	0.000000
gill.attachment	**16.184971**
gill.spacing	**41.040462**
gill.color	0.000000
stem.height	0.000000
stem.width	0.000000
stem.root	**84.393064**
stem.surface	**62.427746**
stem.color	0.000000
veil.type	**94.797688**
veil.color	**87.861272**
has.ring	0.000000
ring.type	4.046243
Spore.print.color	**89.595376**
habitat	0.000000
season	0.000000

Table A2: Percent of missing values for each raw variable. Variables highlighted in yellow exceed the 15% threshold and were dropped from the data

Color	Color Code	Ring Type	Ring Code	Habitat Type	Habitat Code	Cap Shape	Cap Code	Season	Season Code
brown	n	cobwebby	c	grasses	g	bell	b	Spring	s
buff	b	evanescent	e	leaves	l	conical	c	Summer	u
gray	g	flaring	r	meadows	m	convex	x	Autumn	a
green	r	grooved	g	paths	p	flat	f	Winter	w
pink	p	large	l	heaths	h	sunken	s		
purple	u	pendant	p	urban	u	spherical	p		
red	e	sheathing	s	waste	w	others	o		
white	w	zone	z	woods	d				
yellow	y	scaly	y						
blue	l	movable	m						
orange	o	none	f						
black	k	unknown	?						

Table A3: Categorical variables retained in the model with their associated code

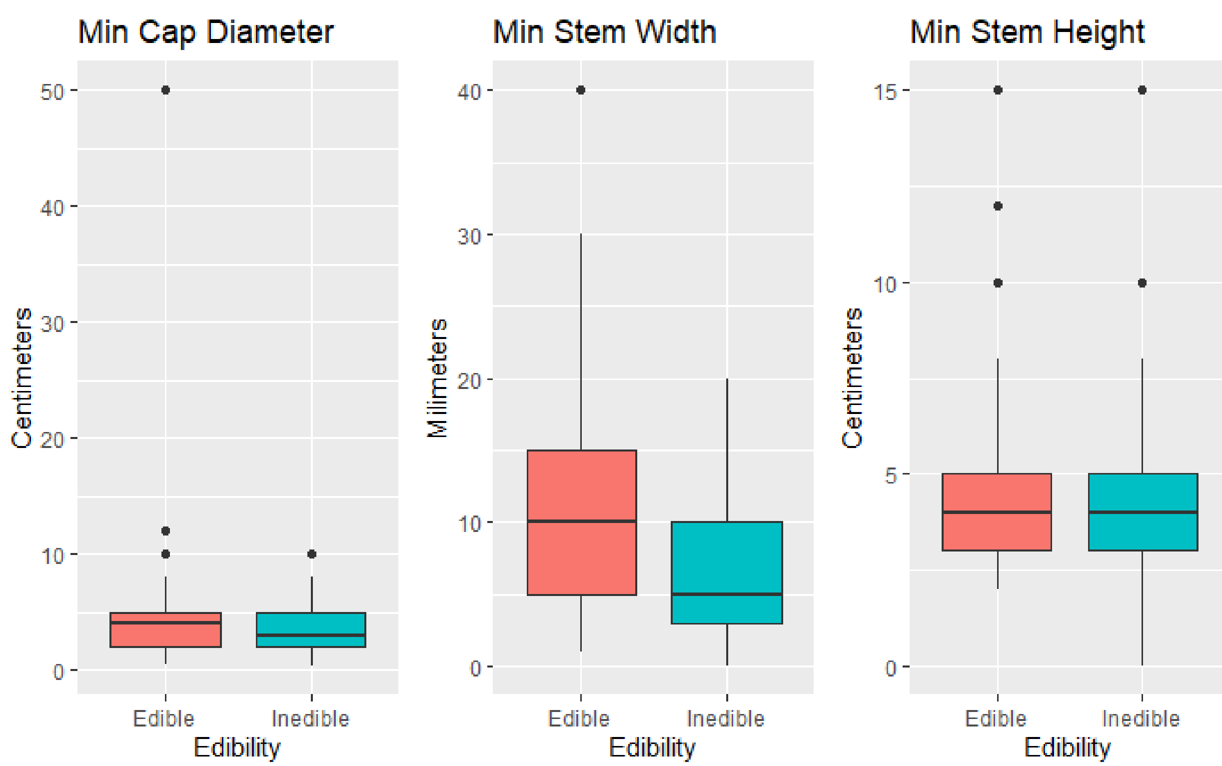

Figure A2: Boxplots of minimum cap diameter, stem width, and stem height by edibility status

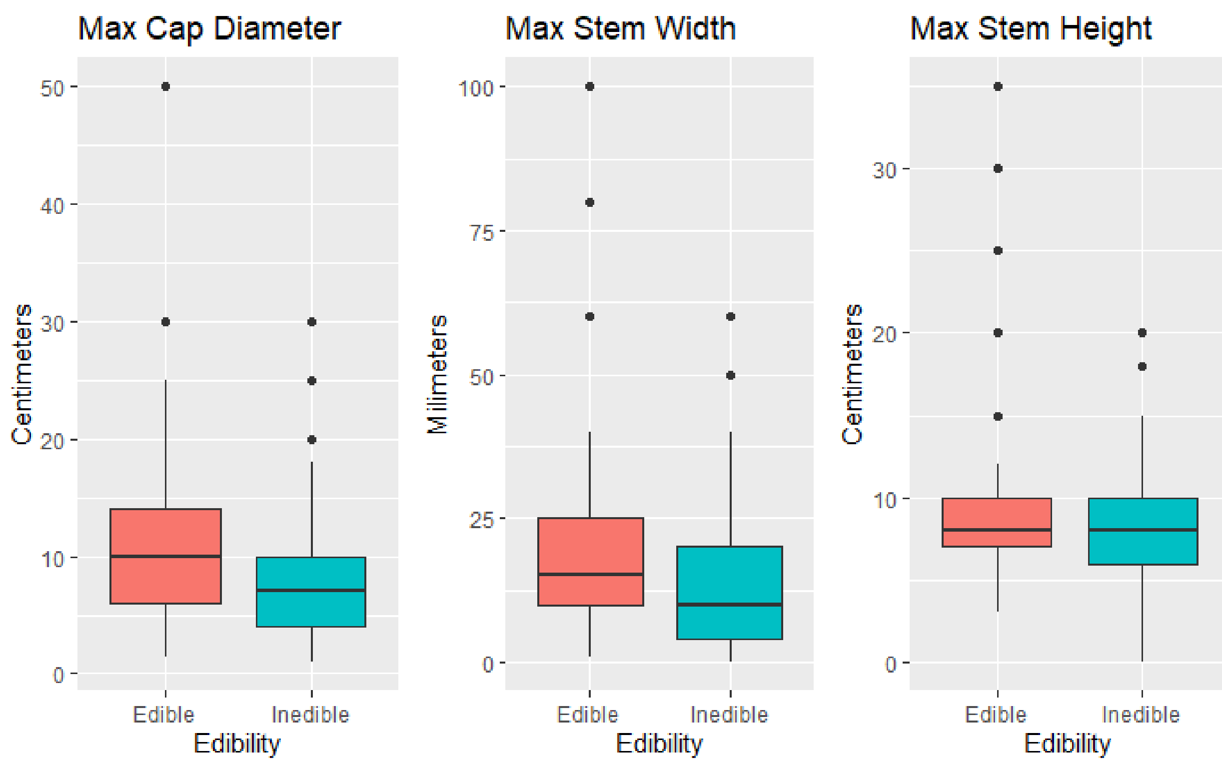

Figure A3: Boxplots of maximum cap diameter, stem width, and stem height by edibility status

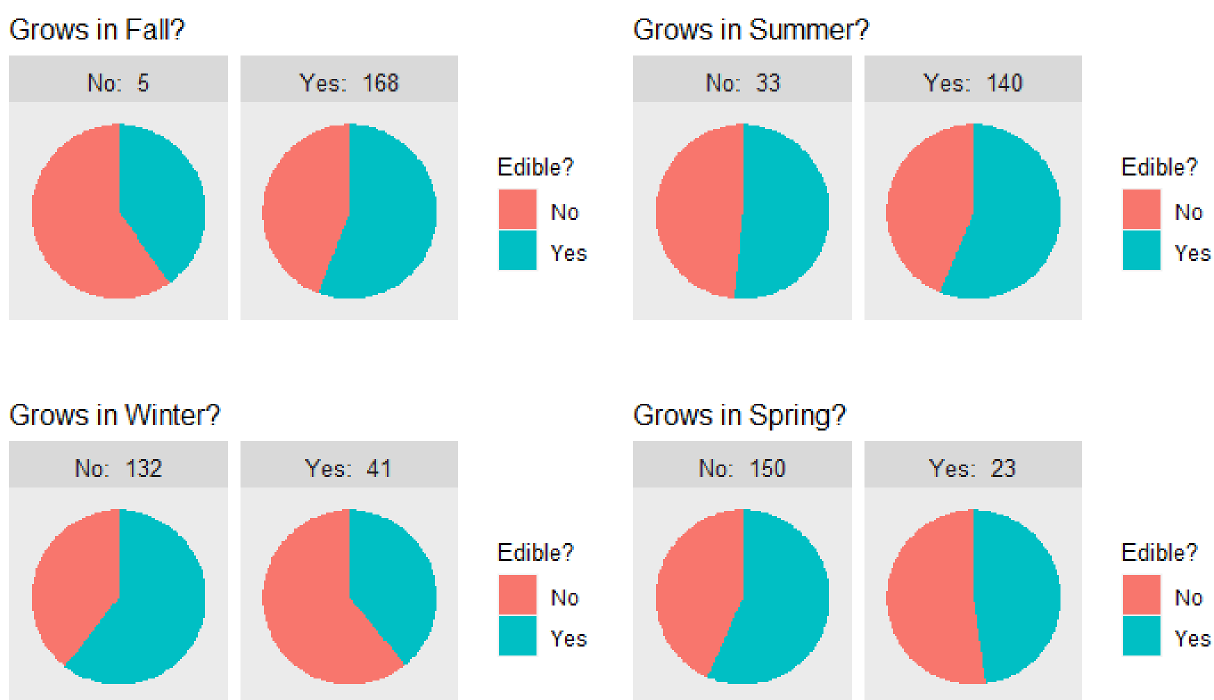

Figure A4: Season binary variables plotted by count and split by edibility status

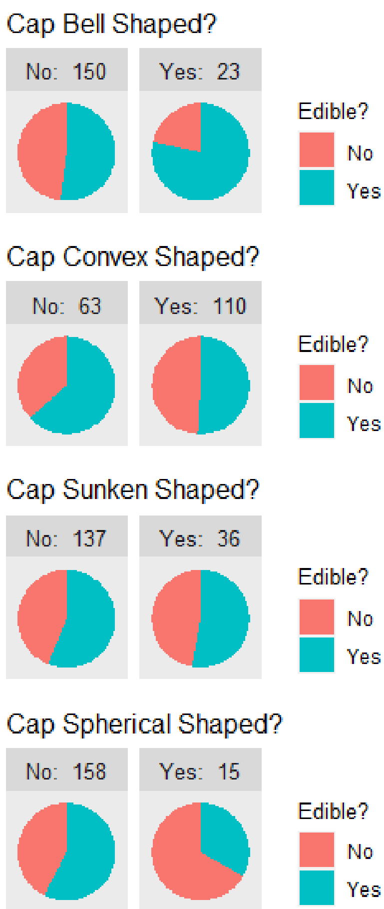

Figure A5: Cap Shape binary variables plotted by count and split by edibility status

Figure A6: Stem color binary variables plotted by count and split by edibility status. Splits with low sample sizes (<5) should not be considered in the model

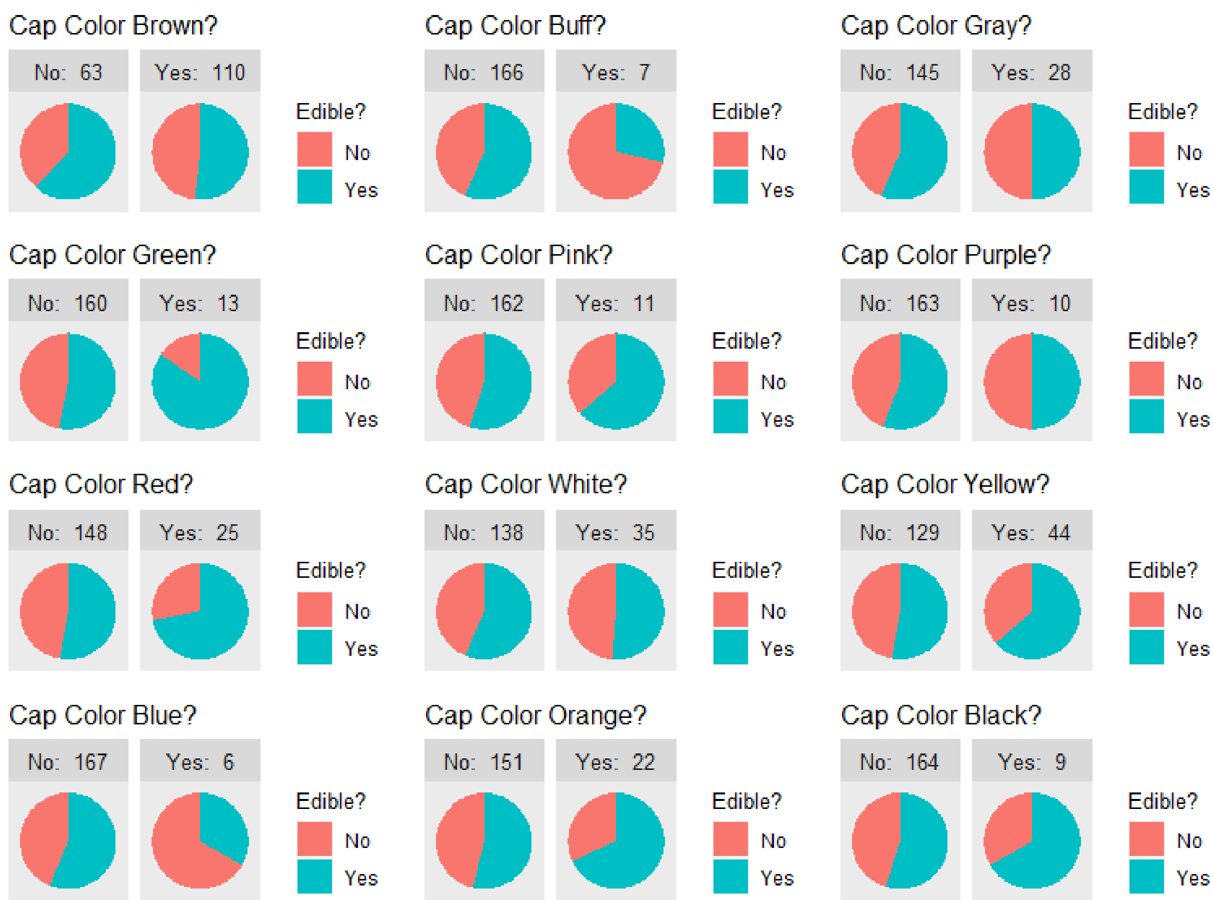

Figure A7: Cap color binary variables plotted with count and split by edibility status. Splits with low sample sizes (<5) should not be considered in the model

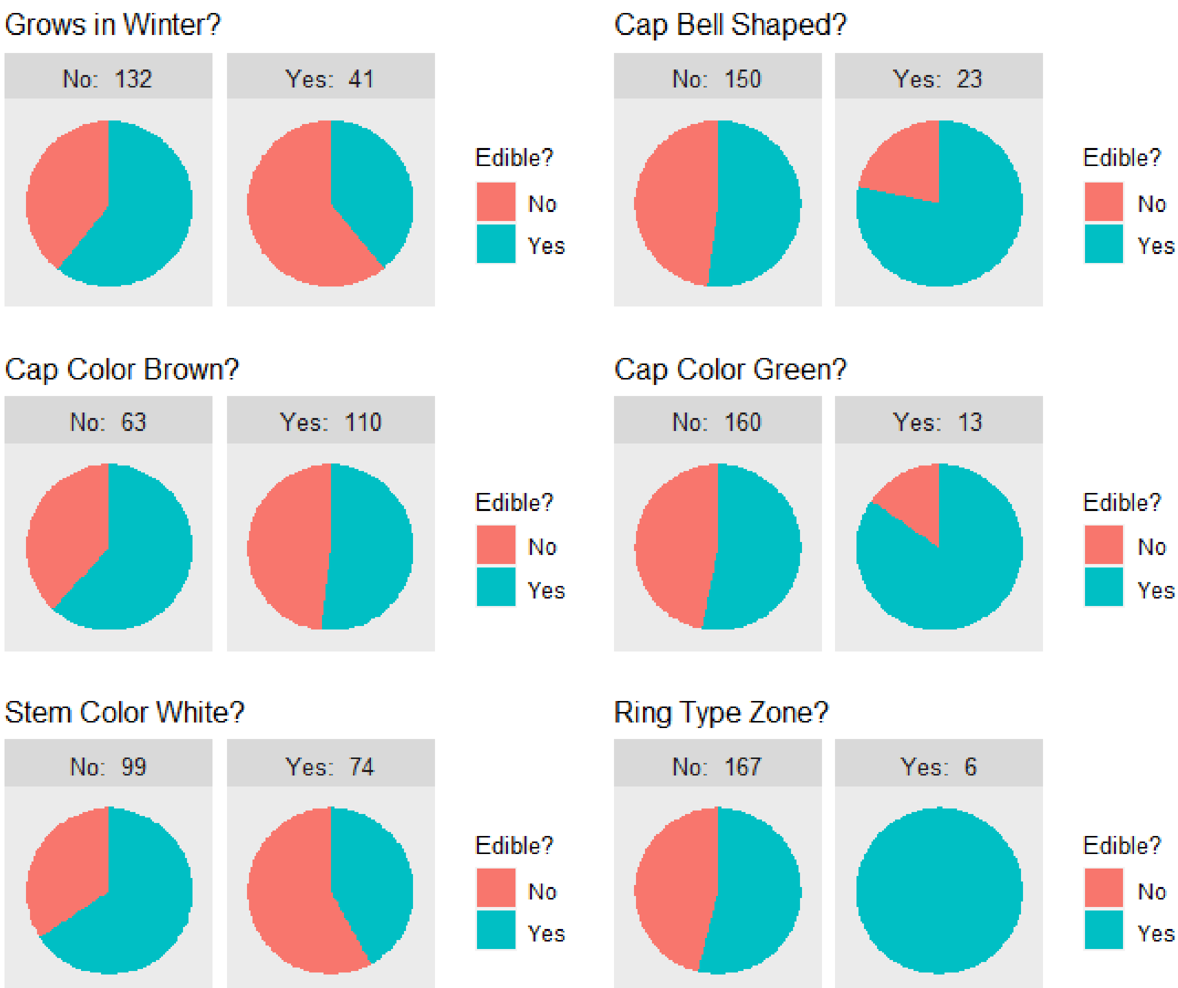

Figure A8: BIC Model predictor variables plotted with count and split by edibility status. Ring Type Zone shows poor balance, and has no samples with a zone type that are inedible.

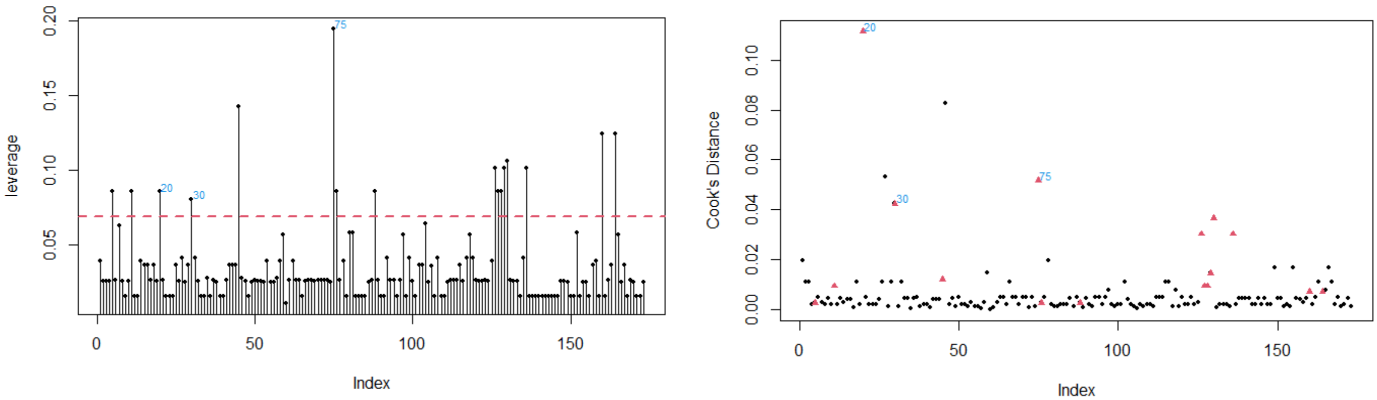

Figure A9: Leverage and Cooks distance plots for the final model. Points 20 and 75 were removed.

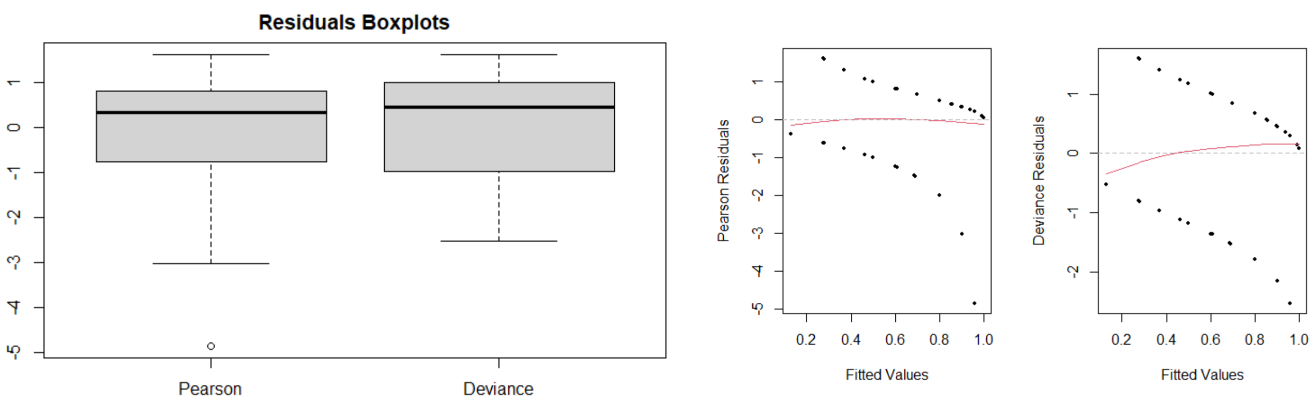

Figure A10: Diagnostic plots for the fit without outlier points 20 and 75

Appendix B

R Code

```
library(MASS)

library(tidyverse)
library(kableExtra)
library(gridExtra)

#download from sourse
mush = read.csv("primary_data.csv",
sep=";",na.strings=c(""," ","NA",character(0)))
head(mush)
percent_na =
data.frame(colMeans(is.na(mush[c(4:23)])))
percent_na =
cbind(rownames(percent_na),percent_na*100)
rownames(percent_na)=NULL

percent_na%>%kable(col.names =
c("Variable","Percent Missing"))%>%
kable_classic_2(full_width =F,lightable_options
=c("striped","bordered") )%>%
  column_spec (1:2,
              border_left = T,
              border_right = T) %>%
  row_spec(c(3,6,7,11,12,14,15,18), bold = T,
color = "black", background = "gold")



#drop NA columns

mush1 = mush[,-c(6,8,9,10,14,15,17,18,21)]
head(mush1,3)%>%kable%>%
kable_classic(full_width =F,lightable_options
=c("striped","bordered") )%>%
  column_spec (1:14,
              border_left = T,
              border_right = T)

colMeans(is.na(mush1))
mush1$class =
as.factor(ifelse(mush1$class=="p",1,0)) #set
inedible to 1

#process text
mush1 = as.data.frame(sapply(mush1,
                    function(x) gsub("\\[|\\]", "",
x)))
mush1 = as.data.frame(sapply(mush1,
                    function(x) gsub(" ","", x)))

#STEM.WIDTH, STEM.HEIGHT,
CAP.DIAMETER, extraction processing

minmax=function(x){
  result=c()
for (ele in x){
  result= rbind(result,
cbind(min(as.numeric(unlist(strsplit(ele,",")))),

max(as.numeric(unlist(strsplit(ele,","))))))

}
  return(result)
}
mush1=mush1 %>%
  mutate(
    cap.diameter_max =
minmax(mush1$cap.diameter)[,2],
    stem.height_max =
minmax(mush1$stem.height)[,2],
    stem.width_max =
minmax(mush1$stem.width)[,2],
    cap.diameter_min =
minmax(mush1$cap.diameter)[,1],
    stem.height_min =
minmax(mush1$stem.height)[,1],
    stem.width_min =
minmax(mush1$stem.width)[,1]
)
#Hot encoding for every possible present string
code
# SEASON

mush1=mush1 %>%
  mutate(
    season_a = str_extract(season, "a"),
    season_s = str_extract(season, "s"),
    season_u = str_extract(season, "u"),
    season_w = str_extract(season, "w")
  )
```

```
mush1$season_a=ifelse(is.na(mush1$season_a),
0,1)
mush1$season_s=ifelse(is.na(mush1$season_s),
0,1)
mush1$season_u=ifelse(is.na(mush1$season_u),
0,1)
mush1$season_w=ifelse(is.na(mush1$season_w
),0,1)

# CAP.SHAPE

mush1=mush1 %>%
  mutate(
    cap.shape_b = str_extract(cap.shape, "b"),
    cap.shape_c = str_extract(cap.shape, "c"),
    cap.shape_x = str_extract(cap.shape, "x"),
    cap.shape_f = str_extract(cap.shape, "f"),
    cap.shape_s = str_extract(cap.shape, "s"),
    cap.shape_p = str_extract(cap.shape, "p"),
    cap.shape_o = str_extract(cap.shape, "o"),
  )


mush1$cap.shape_b =
ifelse(is.na(mush1$cap.shape_b),0,1)
mush1$cap.shape_c =
ifelse(is.na(mush1$cap.shape_c),0,1)
mush1$cap.shape_x =
ifelse(is.na(mush1$cap.shape_x),0,1)
mush1$cap.shape_f =
ifelse(is.na(mush1$cap.shape_f),0,1)
mush1$cap.shape_s =
ifelse(is.na(mush1$cap.shape_s),0,1)
mush1$cap.shape_p =
ifelse(is.na(mush1$cap.shape_p),0,1)
mush1$cap.shape_o =
ifelse(is.na(mush1$cap.shape_o),0,1)

#HAS.RING

mush1$has.ring =
ifelse((mush1$has.ring)=="t",1,0)

#CAP.COLOR

mush1=mush1 %>%
  mutate(
      cap.color_n = str_extract(cap.color, "n"),
      cap.color_b = str_extract(cap.color, "b"),
      cap.color_g = str_extract(cap.color, "g"),
      cap.color_r = str_extract(cap.color, "r"),
      cap.color_p = str_extract(cap.color, "p"),
      cap.color_u = str_extract(cap.color, "u"),
      cap.color_e = str_extract(cap.color, "e"),
      cap.color_w = str_extract(cap.color, "w"),
      cap.color_y = str_extract(cap.color, "y"),
      cap.color_l = str_extract(cap.color, "l"),
      cap.color_o = str_extract(cap.color, "o"),
      cap.color_k = str_extract(cap.color, "k")
)

mush1$cap.color_n =
ifelse(is.na(mush1$cap.color_n),0,1)
mush1$cap.color_b =
ifelse(is.na(mush1$cap.color_b),0,1)
mush1$cap.color_g =
ifelse(is.na(mush1$cap.color_g),0,1)
mush1$cap.color_r =
ifelse(is.na(mush1$cap.color_r),0,1)
mush1$cap.color_p =
ifelse(is.na(mush1$cap.color_p),0,1)
mush1$cap.color_u =
ifelse(is.na(mush1$cap.color_u),0,1)
mush1$cap.color_e =
ifelse(is.na(mush1$cap.color_e),0,1)
mush1$cap.color_w =
ifelse(is.na(mush1$cap.color_w),0,1)
mush1$cap.color_y =
ifelse(is.na(mush1$cap.color_y),0,1)
mush1$cap.color_l =
ifelse(is.na(mush1$cap.color_l),0,1)
mush1$cap.color_o =
ifelse(is.na(mush1$cap.color_o),0,1)
mush1$cap.color_k =
ifelse(is.na(mush1$cap.color_k),0,1)



# HABITAT


mush1=mush1 %>%
  mutate(
      habitat_g = str_extract(habitat, "g"),
      habitat_l = str_extract(habitat, "l"),
      habitat_m = str_extract(habitat, "m"),
      habitat_p = str_extract(habitat, "p"),
      habitat_h = str_extract(habitat, "h"),
      habitat_u = str_extract(habitat, "u"),
      habitat_w = str_extract(habitat, "w"),
      habitat_d = str_extract(habitat, "d")
```

```
)

mush1$habitat_g =
ifelse(is.na(mush1$habitat_g),0,1)
mush1$habitat_l =
ifelse(is.na(mush1$habitat_l),0,1)
mush1$habitat_m =
ifelse(is.na(mush1$habitat_m),0,1)
mush1$habitat_p =
ifelse(is.na(mush1$habitat_p),0,1)
mush1$habitat_h =
ifelse(is.na(mush1$habitat_h),0,1)
mush1$habitat_u =
ifelse(is.na(mush1$habitat_u),0,1)
mush1$habitat_w =
ifelse(is.na(mush1$habitat_w),0,1)
mush1$habitat_d =
ifelse(is.na(mush1$habitat_d),0,1)


# STEM.COLOR
mush1=mush1 %>%
  mutate(
    stem.color_n = str_extract(stem.color, "n"),
    stem.color_b = str_extract(stem.color, "b"),
    stem.color_g = str_extract(stem.color, "g"),
    stem.color_r = str_extract(stem.color, "r"),
    stem.color_p = str_extract(stem.color, "p"),
    stem.color_u = str_extract(stem.color, "u"),
    stem.color_e = str_extract(stem.color, "e"),
    stem.color_w = str_extract(stem.color, "w"),
    stem.color_y = str_extract(stem.color, "y"),
    stem.color_l = str_extract(stem.color, "l"),
    stem.color_o = str_extract(stem.color, "o"),
    stem.color_k = str_extract(stem.color, "k")
)

mush1$stem.color_n =
ifelse(is.na(mush1$stem.color_n),0,1)
mush1$stem.color_b =
ifelse(is.na(mush1$stem.color_b),0,1)
mush1$stem.color_g =
ifelse(is.na(mush1$stem.color_g),0,1)
mush1$stem.color_r =
ifelse(is.na(mush1$stem.color_r),0,1)
mush1$stem.color_p =
ifelse(is.na(mush1$stem.color_p),0,1)
mush1$stem.color_u =
ifelse(is.na(mush1$stem.color_u),0,1)
mush1$stem.color_e =
ifelse(is.na(mush1$stem.color_e),0,1)
mush1$stem.color_w =
ifelse(is.na(mush1$stem.color_w),0,1)
mush1$stem.color_y =
ifelse(is.na(mush1$stem.color_y),0,1)
mush1$stem.color_l =
ifelse(is.na(mush1$stem.color_l),0,1)
mush1$stem.color_o =
ifelse(is.na(mush1$stem.color_o),0,1)
mush1$stem.color_k =
ifelse(is.na(mush1$stem.color_k),0,1)


# GILL.COLOR

mush1=mush1 %>%
  mutate(
    gill.color_n = str_extract(gill.color, "n"),
    gill.color_b = str_extract(gill.color, "b"),
    gill.color_g = str_extract(gill.color, "g"),
    gill.color_r = str_extract(gill.color, "r"),
    gill.color_p = str_extract(gill.color, "p"),
    gill.color_u = str_extract(gill.color, "u"),
    gill.color_e = str_extract(gill.color, "e"),
    gill.color_w = str_extract(gill.color, "w"),
    gill.color_y = str_extract(gill.color, "y"),
    gill.color_l = str_extract(gill.color, "l"),
    gill.color_o = str_extract(gill.color, "o"),
    gill.color_k = str_extract(gill.color, "k")
)

mush1$gill.color_n =
ifelse(is.na(mush1$gill.color_n),0,1)
mush1$gill.color_b =
ifelse(is.na(mush1$gill.color_b),0,1)
mush1$gill.color_g =
ifelse(is.na(mush1$gill.color_g),0,1)
mush1$gill.color_r =
ifelse(is.na(mush1$gill.color_r),0,1)
mush1$gill.color_p =
ifelse(is.na(mush1$gill.color_p),0,1)
mush1$gill.color_u =
ifelse(is.na(mush1$gill.color_u),0,1)
mush1$gill.color_e =
ifelse(is.na(mush1$gill.color_e),0,1)
mush1$gill.color_w =
ifelse(is.na(mush1$gill.color_w),0,1)
mush1$gill.color_y =
ifelse(is.na(mush1$gill.color_y),0,1)
mush1$gill.color_l =
ifelse(is.na(mush1$gill.color_l),0,1)
```

```
mush1$gill.color_o =
ifelse(is.na(mush1$gill.color_o),0,1)
mush1$gill.color_k =
ifelse(is.na(mush1$gill.color_k),0,1)

# RING.TYPE
mush1=mush1 %>%
  mutate(
      ring.type_c = str_extract(ring.type, "c"),
      ring.type_e = str_extract(ring.type, "e"),
      ring.type_r = str_extract(ring.type, "r"),
      ring.type_g = str_extract(ring.type, "g"),
      ring.type_l = str_extract(ring.type, "l"),
      ring.type_p = str_extract(ring.type, "p"),
      ring.type_s = str_extract(ring.type, "s"),
      ring.type_z = str_extract(ring.type, "z"),
      ring.type_y = str_extract(ring.type, "y"),
      ring.type_m = str_extract(ring.type, "m"),
      ring.type_f = str_extract(ring.type, "f")
      #ring.type_u = str_extract(ring.type, "?")
)

mush1$ring.type_c =
ifelse(is.na(mush1$ring.type_c),0,1)
mush1$ring.type_e =
ifelse(is.na(mush1$ring.type_e),0,1)
mush1$ring.type_r =
ifelse(is.na(mush1$ring.type_r),0,1)
mush1$ring.type_g =
ifelse(is.na(mush1$ring.type_g),0,1)
mush1$ring.type_l =
ifelse(is.na(mush1$ring.type_l),0,1)
mush1$ring.type_p =
ifelse(is.na(mush1$ring.type_p),0,1)
mush1$ring.type_s =
ifelse(is.na(mush1$ring.type_s),0,1)
mush1$ring.type_z =
ifelse(is.na(mush1$ring.type_z),0,1)
mush1$ring.type_y =
ifelse(is.na(mush1$ring.type_y),0,1)
mush1$ring.type_m =
ifelse(is.na(mush1$ring.type_m),0,1)
mush1$ring.type_f =
ifelse(is.na(mush1$ring.type_f),0,1)
#mush1$ring.type_u =
ifelse(is.na(mush1$ring.type_u),0,1) #do not
include ring type '?', this is #unknown, and
therefore NA

mush2 = mush1[c(3,11,15:86)] #subset relevant
variables

mush3=mush2 #new dataset for graphs
mush3$classtype = ifelse(mush2$class==1,
"Inedible","Edible")

#Numerical Graphs
p1 =
mush3%>%ggplot(aes(x=factor(classtype),y=ca
p.diameter_max, fill=factor(classtype)))+
  geom_boxplot(position="dodge")+
  labs(title="Max Cap Diameter", x="Edibility",
y="Centimeters", fill="Edibility")+
    theme(legend.position = "none")

p2 =
mush3%>%ggplot(aes(x=factor(classtype),y=ca
p.diameter_min, fill=factor(classtype)))+
  geom_boxplot(position="dodge")+
  labs(title="Min Cap Diameter", x="Edibility",
y="Centimeters", fill="Edibility")+
    theme(legend.position = "none")


p3 =
mush3%>%ggplot(aes(x=factor(classtype),y=ste
m.width_max, fill=factor(classtype)))+
  geom_boxplot(position="dodge")+
  labs(title="Max Stem Width", x="Edibility",
y="Milimeters", fill="Edibility")+
    theme(legend.position = "none")

p4 =
mush3%>%ggplot(aes(x=factor(classtype),y=ste
m.width_min, fill=factor(classtype)))+
  geom_boxplot(position="dodge")+
  labs(title="Min Stem Width", x="Edibility",
y="Milimeters", fill="Edibility")+
    theme(legend.position = "none")

p5 =
mush3%>%ggplot(aes(x=factor(classtype),y=ste
m.height_max, fill=factor(classtype)))+
  geom_boxplot(position="dodge")+
  labs(title="Max Stem Height", x="Edibility",
y="Centimeters", fill="Edibility")+
    theme(legend.position = "none")
```

```
p6 =
mush3%>%ggplot(aes(x=factor(classtype),y=stem.height_min, fill=factor(classtype)))+
  geom_boxplot(position="dodge")+
  labs(title="Min Stem Height", x="Edibility",
y="Centimeters", fill="Edibility")+
    theme(legend.position = "none")

grid.arrange(p1,p3, p5,nrow = 1)
grid.arrange(p2,p4,p6, nrow = 1)


#Vectors for table assembly

 var_og= c("family",
 "name" ,
 "class",
 "cap.diameter" ,
 "cap.shape",
 "Cap.surface"  ,
 "cap.color",
 "does.bruise.or.bleed",
 "gill.attachment",
 "gill.spacing" ,
 "gill.color",
 "stem.height"  ,
 "stem.width",
 "stem.root"  ,
 "stem.surface",
 "stem.color" ,
 "veil.type",
 "veil.color" ,
 "has.ring",
 "ring.type"  ,
 "Spore.print.color",
 "habitat",
 "season")

  description_og= c("Taxomic Family",
 "Species Name" ,
 "Edibility Status",
 "Diameter of Mushroom Cap (cm)" ,
 "Mushroom Cap Shape",
 "Cap Surface Texture"  ,
 "Color of Cap",
 "Does Mushroom Bruise or Bleed (t/f)",
 "Gill Attachment Present",
 "Gill Spacing Type" ,
 "Gill Color",
 "Stem Height (cm)"  ,
 "stem.width (mm)",
 "Stem Root Type"  ,
 "Stem Surface Type",
 "Stem Color" ,
 "Veil Type",
 "Veil Color" ,
 "Is a Ring Present (t/f)",
 "Type of Ring"  ,
 "Color of Spores",
 "Native Habitat",
 "Growing Season")



colors = color_codes=c("n", "b","g",
"r","p","u","e", "w", "y","l","o","k")
color_names=c("brown","buff","gray","green","pink","purple","red","white","yellow","blue","orange","black")

habitat =
c("grasses","leaves","meadows","paths","heaths","urban","waste","woods","","","","")

habitat_code=c("g","l","m","p","h","u","w","d",
"","","","")

seasons =
c("Spring","Summer","Autumn","Winter","","",
"","","","","","")
seasons_code =
c("s","u","a","w","","","","","","","","")

shape = c("bell", "conical", "convex",
"flat","sunken", "spherical",
"others","","","","","")


shape_code =
c("b","c","x","f","s","p","o","","","","","")

ring =
c("cobwebby","evanescent","flaring","grooved",
"large","pendant","sheathing","zone","scaly","movable","none","unknown")

ring_code =
c("c","e","r","g","l","p","s","z","y","m","f","?")

#combine all
oy=do.call("cbind",list(color_names,
color_codes,ring, ring_code,habitat,
```

```
habitat_code,shape,shape_code,
seasons,seasons_code))



#write tables
data.frame(cbind(var_og,description_og)) %>%
  kable(caption = '',
        col.names = c("Variable
Name","Description"))%>%
kable_classic_2(full_width = F,
html_font = "Ariel",lightable_options
=c("striped","bordered")  )%>%
  column_spec (1:2,
              border_left = T,
              border_right = T)
  data.frame(ring,ring_code) %>%
  kable(caption = '',
        col.names = c("Ring Type","Code"))%>%
kable_classic(full_width = F,
html_font = "Ariel",lightable_options
="striped"  )%>%
  column_spec (1:2,
              border_left = T,
              border_right = T)

data.frame(habitat,habitat_code) %>%
  kable(caption = '',
        col.names = c("Habitat","Code"))%>%
kable_classic(full_width = F,
html_font = "Ariel",lightable_options
="striped"  )%>%
  column_spec (1:2,
              border_left = T,
              border_right = T)

data.frame(shape,shape_code) %>%
  kable(caption = '',
        col.names = c("Cap Shape","Code"))%>%
kable_classic(full_width = F,
html_font = "Ariel",lightable_options
="striped"  )%>%
  column_spec (1:2,
              border_left = T,
              border_right = T)

data.frame(oy) %>%
  kable(caption = '',
        col.names = c("Color","Color Code","Ring
Type","Ring Code","Habitat Type","Habitat
Code","Cap Shape","Cap
Code","Season","Season Code"))%>%
kable_classic_2(full_width =F,lightable_options
=c("striped","bordered") )%>%
  column_spec (1:10,
              border_left = T,
              border_right = T)


#preprocessing example table
cbind(head(mush[c(4,23)],3),head( mush2[c(3,6,
9:12)],3))%>%kable()%>%
kable_classic(full_width =F,lightable_options
=c("striped","bordered") )%>%
  column_spec (1:8,
              border_left = T,
              border_right = T) %>%
  column_spec(c(3:4), bold = T, color = "black",
background = "lightblue") %>%
  column_spec(c(5:8), bold = T, color = "black",
background = "pink")
mush2 = mush1[c(3,11,15:86)]




#Obtain counts for each plotted categorical pie
chart

mush3$cap.color_n=ifelse(mush3$cap.color_n=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_n))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_n))))

mush3$cap.color_b=ifelse(mush3$cap.color_b=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_b))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_b))))

mush3$cap.color_g=ifelse(mush3$cap.color_g=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_g))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_g))))

mush3$cap.color_r=ifelse(mush3$cap.color_r==
"1", paste("Yes:
",as.character(sum(mush2$cap.color_r))),paste("
```

```
No: ",as.character(173-
sum(mush2$cap.color_r))))

mush3$cap.color_p=ifelse(mush3$cap.color_p=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_p))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_p))))

mush3$cap.color_u=ifelse(mush3$cap.color_u=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_u))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_u))))

mush3$cap.color_e=ifelse(mush3$cap.color_e=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_e))),paste("
No: ",as.character(173-
sum(mush2$cap.color_e))))

mush3$cap.color_w=ifelse(mush3$cap.color_w
=="1", paste("Yes:
",as.character(sum(mush2$cap.color_w))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_w))))

mush3$cap.color_y=ifelse(mush3$cap.color_y=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_y))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_y))))

mush3$cap.color_l=ifelse(mush3$cap.color_l==
"1", paste("Yes:
",as.character(sum(mush2$cap.color_l))),paste("
No: ",as.character(173-
sum(mush2$cap.color_l))))

mush3$cap.color_o=ifelse(mush3$cap.color_o=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_o))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_o))))

mush3$cap.color_k=ifelse(mush3$cap.color_k=
="1", paste("Yes:
",as.character(sum(mush2$cap.color_k))),paste(
"No: ",as.character(173-
sum(mush2$cap.color_k))))

mush3$stem.color_n=ifelse(mush3$stem.color_
n=="1", paste("Yes:
",as.character(sum(mush2$stem.color_n))),paste
("No: ",as.character(173-
sum(mush2$stem.color_n))))

mush3$stem.color_b=ifelse(mush3$stem.color_
b=="1", paste("Yes:
",as.character(sum(mush2$stem.color_b))),paste
("No: ",as.character(173-
sum(mush2$stem.color_b))))

mush3$stem.color_g=ifelse(mush3$stem.color_
g=="1", paste("Yes:
",as.character(sum(mush2$stem.color_g))),paste
("No: ",as.character(173-
sum(mush2$stem.color_g))))

mush3$stem.color_r=ifelse(mush3$stem.color_r
=="1", paste("Yes:
",as.character(sum(mush2$stem.color_r))),paste(
"No: ",as.character(173-
sum(mush2$stem.color_r))))

mush3$stem.color_p=ifelse(mush3$stem.color_
p=="1", paste("Yes:
",as.character(sum(mush2$stem.color_p))),paste
("No: ",as.character(173-
sum(mush2$stem.color_p))))

mush3$stem.color_u=ifelse(mush3$stem.color_
u=="1", paste("Yes:
",as.character(sum(mush2$stem.color_u))),paste
("No: ",as.character(173-
sum(mush2$stem.color_u))))

mush3$stem.color_e=ifelse(mush3$stem.color_
e=="1", paste("Yes:
",as.character(sum(mush2$stem.color_e))),paste
("No: ",as.character(173-
sum(mush2$stem.color_e))))

mush3$stem.color_w=ifelse(mush3$stem.color_
w=="1", paste("Yes:
",as.character(sum(mush2$stem.color_w))),paste
("No: ",as.character(173-
sum(mush2$stem.color_w))))
```

```
mush3$stem.color_y=ifelse(mush3$stem.color_y=="1", paste("Yes: ",as.character(sum(mush2$stem.color_y))),paste("No: ",as.character(173-sum(mush2$stem.color_y))))

mush3$stem.color_l=ifelse(mush3$stem.color_l=="1", paste("Yes: ",as.character(sum(mush2$stem.color_l))),paste("No: ",as.character(173-sum(mush2$stem.color_l))))

mush3$stem.color_o=ifelse(mush3$stem.color_o=="1", paste("Yes: ",as.character(sum(mush2$stem.color_o))),paste("No: ",as.character(173-sum(mush2$stem.color_o))))

mush3$stem.color_k=ifelse(mush3$stem.color_k=="1", paste("Yes: ",as.character(sum(mush2$stem.color_k))),paste("No: ",as.character(173-sum(mush2$stem.color_k))))



mush3$season_a=ifelse(mush3$season_a=="1", paste("Yes: ",as.character(sum(mush2$season_a))),paste("No: ",as.character(173-sum(mush2$season_a))))

mush3$season_u=ifelse(mush3$season_u=="1", paste("Yes: ",as.character(sum(mush2$season_u))),paste("No: ",as.character(173-sum(mush2$season_u))))

mush3$season_s=ifelse(mush3$season_s=="1", paste("Yes: ",as.character(sum(mush2$season_s))),paste("No: ",as.character(173-sum(mush2$season_s))))

mush3$season_w=ifelse(mush3$season_w=="1",  paste("Yes: ",as.character(sum(mush2$season_w))),paste("No: ",as.character(173-sum(mush2$season_w))))


u=mush3%>%ggplot(aes( x=factor(classtype), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(season_u))+
   theme(axis.title = element_blank(),
         axis.text = element_blank(),
         axis.ticks = element_blank(),
         panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         strip.text.x = element_text(size = 10),
         title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Grows in Summer?")


a=mush3%>%ggplot(aes( x=factor(classtype), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(season_a))+
   theme(axis.title = element_blank(),
         axis.text = element_blank(),
         axis.ticks = element_blank(),
         panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         strip.text.x = element_text(size = 10),
         title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Grows in Fall?")

w=mush3%>%ggplot(aes( x=factor(classtype), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(season_w))+
  theme(axis.title = element_blank(),
         axis.text = element_blank(),
         axis.ticks = element_blank(),
         panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         strip.text.x = element_text(size = 10),
         title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Grows in Winter?")

sp=mush3%>%ggplot(aes( x=factor(classtype), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
```

```
  coord_polar(theta = "y")+
  facet_wrap(~factor(season_s))+
   theme(axis.title = element_blank(),
         axis.text = element_blank(),
         axis.ticks = element_blank(),
         panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         strip.text.x = element_text(size = 10),
         title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Grows in Spring?")

#seasons pie
grid.arrange(a, u,w,sp, ncol=2)



mush3$cap.shape_b=ifelse(mush3$cap.shape_b
=="1", paste("Yes:
",as.character(sum(mush2$cap.shape_b))),paste(
"No: ",as.character(173-
sum(mush2$cap.shape_b))))
mush3$cap.shape_c=ifelse(mush3$cap.shape_c
=="1", paste("Yes:
",as.character(sum(mush2$cap.shape_c))),paste(
"No: ",as.character(173-
sum(mush2$cap.shape_c))))

mush3$cap.shape_x=ifelse(mush3$cap.shape_x
=="1", paste("Yes:
",as.character(sum(mush2$cap.shape_x))),paste(
"No: ",as.character(173-
sum(mush2$cap.shape_x))))

mush3$cap.shape_f=ifelse(mush3$cap.shape_f=
="1",  paste("Yes:
",as.character(sum(mush2$cap.shape_f))),paste(
"No: ",as.character(173-
sum(mush2$cap.shape_f))))


mush3$ring.type_z=ifelse(mush3$ring.type_z==
"1", paste("Yes:
",as.character(sum(mush2$ring.type_z))),paste("
No: ",as.character(173-
sum(mush2$ring.type_z))))

mush3$cap.shape_s=ifelse(mush3$cap.shape_s=
="1",  paste("Yes:
",as.character(sum(mush2$cap.shape_s))),paste(
"No: ",as.character(173-
sum(mush2$cap.shape_s))))


mush3$cap.shape_o=ifelse(mush3$cap.shape_o
=="1",  paste("Yes:
",as.character(sum(mush2$cap.shape_o))),paste(
"No: ",as.character(173-
sum(mush2$cap.shape_o))))

mush3$cap.shape_p=ifelse(mush3$cap.shape_p
=="1", paste("Yes:
",as.character(sum(mush2$cap.shape_p))),paste(
"No: ",as.character(173-
sum(mush2$cap.shape_p))))



b=mush3%>%ggplot(aes( x=factor(classtype),
fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.shape_b))+
   theme(axis.title = element_blank(),
         axis.text = element_blank(),
         axis.ticks = element_blank(),
         panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         strip.text.x = element_text(size = 10),
         title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Bell Shaped?")

c=mush3%>%ggplot(aes( x=factor(classtype),
fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.shape_c))+
    theme(axis.title = element_blank(),
         axis.text = element_blank(),
         axis.ticks = element_blank(),
         panel.grid.major = element_blank(),
         panel.grid.minor = element_blank(),
         panel.border = element_blank(),
         strip.text.x = element_text(size = 10),
```

```
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Conical Shaped?")

xp=mush3%>%ggplot(aes( x=factor(classtype),
fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.shape_x))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Convex Shaped?")

f=mush3%>%ggplot(aes( x=factor(classtype),
fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.shape_f))+
   theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Flat Shaped?")


s=mush3%>%ggplot(aes( x=factor(classtype),
fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.shape_s))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Sunken Shaped?")


o=mush3%>%ggplot(aes( x=factor(classtype),
fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.shape_o))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Other Shaped?")


p=mush3%>%ggplot(aes( x=factor(classtype),
fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.shape_p))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Spherical Shaped?")

#cap shape pies
grid.arrange(b,
c,
xp,
f,
s,
o,
p, ncol = 2)
```

```
#stem color pies
stC_n=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_n))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Brown?")


stC_b=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_b))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Buff?")

stC_g=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_g))+
   theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Gray?")

stC_r=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_r))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Green?")

stC_p=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_p))+
   theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Pink?")

stC_u=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_u))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
```

```
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Purple?")

stC_e=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_e))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Red?")

stC_w=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_w))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color White?")

stC_y=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_y))+
   theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
```

```
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Yellow?")

stC_l=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_l))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Blue?")

stC_o=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_o))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Stem Color Orange?")

stC_k=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(stem.color_k))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
```

```
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Stem Color Black?")


grid.arrange(stC_n,
stC_b,
stC_g,
stC_r,
stC_p,
stC_u,
stC_e,
stC_w,
stC_y,
stC_l,
stC_o,
stC_k, ncol = 3)




#cap color pies
cpC_n=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_n))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Brown?")


cpC_b=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_b))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Buff?")

cpC_g=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_g))+
   theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Gray?")

cpC_r=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_r))+
    theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          panel.border = element_blank(),
          strip.text.x = element_text(size = 10),
          title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Green?")

cpC_p=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_p))+
   theme(axis.title = element_blank(),
          axis.text = element_blank(),
          axis.ticks = element_blank(),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
```

```
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Pink?")

cpC_u=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_u))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Purple?")

cpC_e=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_e))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Red?")

cpC_w=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_w))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
```

```
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color White?")

cpC_y=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_y))+
   theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Yellow?")

cpC_l=mush3%>%ggplot(aes( x=factor(classtyp
e), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_l))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                    labels = c("No","Yes"))+
  labs(title="Cap Color Blue?")

cpC_o=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_o))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
```

```
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Color Orange?")

cpC_k=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(cap.color_k))+
    theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size=10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Cap Color Black?")


grid.arrange(cpC_n,
cpC_b,
cpC_g,
cpC_r,
cpC_p,
cpC_u,
cpC_e,
cpC_w,
cpC_y,
cpC_l,
cpC_o,
cpC_k, ncol = 3)


#pie to investigate ring.type_z
ring_z=mush3%>%ggplot(aes( x=factor(classty
pe), fill=factor(classtype)))+
  geom_col(aes(x = 1, y = n), position = "fill")+
  coord_polar(theta = "y")+
  facet_wrap(~factor(ring.type_z))+
   theme(axis.title = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.border = element_blank(),
        strip.text.x = element_text(size = 10),
        title = element_text(size = 10))+
  scale_fill_discrete(name = "Edible?",
                      labels = c("No","Yes"))+
  labs(title="Ring Type Zone?")

# pies for predictors of BIC model
grid.arrange(w,b,
cpC_n,
cpC_r,
stC_w, ncol = 2)
ring_z



#redefine mush 2
mush2 = mush1[c(3,11,15:86)]
mush2$stem.width_max=mush2$stem.width_m
ax/10 #mm->cm
mush2$stem.width_min=mush2$stem.width_mi
n/10


#remove unused variables
mush2[which(sapply(mush2[2:74],sum)==0)]
mush2=mush2[-c( 61,         64       ,   70   ,
72 )]

#check
mush2$class=as.numeric(mush2$class)
(which(lapply(mush2[2:70],sum)==0))

#make all binary variables factors
mush2[c(2,(9:70))] = lapply(mush2[c(2,(9:70))],
as.factor)


#fit intercept
intercept= glm(class~(.),family = binomial(link
= "logit"),mush2)
summary(intercept)

#ge
x_aic=MASS::stepAIC(intercept,trace=F, k = 2)
x=MASS::stepAIC(intercept,trace=F, k =
log(173))
summary(x_aic)
summary(x)  #models are the same


sum=summary(x)
```

```
anova(x,test="Chisq")
confint(x)

#H0: beta_ring.typeZ = 0
anova(glm(formula = class ~ season_w +
cap.shape_b +  cap.color_n+
   cap.color_r + stem.color_w , family =
binomial(link = "logit"), mush2),glm(formula =
class ~ season_w  + cap.shape_b + cap.color_n
+cap.color_r+
    stem.color_w+ ring.type_z,family =
binomial(link = "logit"),mush2),test="Chi")
confint(x2,level=.9,trace=F)

confint(x,level=.65,trace=F)

#drop ring.type_z from model

#final model
x2=glm(formula = class ~ season_w +
cap.shape_b +  cap.color_n+
   cap.color_r + stem.color_w , family =
binomial(link = "logit"), mush2)
summary(x2)
confint(x2)


#test Regression effect for final model
anova(glm(formula = class ~ 1 , family =
binomial(link = "logit"),mush2),glm(formula =
class ~ season_w + cap.shape_b + cap.color_n +
   cap.color_r + stem.color_w , family =
binomial(link = "logit"),mush2), test="Chi")


#residuals for final model
res.P = residuals(x2, type="pearson")
res.D = residuals(x2, type="deviance") #or
residuals(fit), by default
res = cbind(res.P, res.D)
colnames(res)= c("Pearson", "Deviance")


summary(res)

boxplot(res, main="Residuals Boxplots")

par(mfrow=c(1,2))
plot(x2$fitted.values, res.P, pch=16, cex=0.6,
ylab='Pearson Residuals', xlab='Fitted Values')
lines(smooth.spline(x2$fitted.values, res.P,
spar=0.9), col=2)
abline(h=0, lty=2, col='grey')
plot(x2$fitted.values, res.D, pch=16, cex=0.6,
ylab='Deviance Residuals', xlab='Fitted Values')
lines(smooth.spline(x2$fitted.values, res.D,
spar=0.9), col=2)
abline(h=0, lty=2, col='grey')
library(lawstat)
lawstat::runs.test(res.D, F)


#interactions test -> interactions not significant,
do not include in report as there are so many
variables, interactions will cause dispersion
second=glm(formula = class ~ (season_w +
cap.shape_b + cap.color_n +
   cap.color_r + stem.color_w + ring.type_z)^2,
family = binomial(link = "logit"),
   data = mush2)

 stepAIC(second, trace=F, k=log(173))

anova(intercept,second, test="Chi")

confint(x,level = .9,trace=F)


#outlier diagnostics
leverage = hatvalues(x2)

W = diag(x2$weights)
X = cbind(rep(1,nrow(mush2)),
mush2[['season_w']], mush2[['cap.shape_b']],
        mush2[['cap.color_n']],
mush2[['cap.color_r']], mush2[['stem.color_w']])
Hat = sqrt(W) %*% X %*% solve(t(X) %*%
W %*% X) %*% t(X) %*% sqrt(W)
all(abs(leverage - diag(Hat)) < 1e-15)

plot(names(leverage), leverage, xlab="Index",
type="h")
points(names(leverage), leverage, pch=16,
cex=0.6)
```

```
text(susPts, leverage[susPts], susPts, adj=c(-0.1,-
0.1), cex=0.7, col=4)

p <- length(coef(x2))
n <- nrow(mush2)
abline(h=2*p/n,col=2,lwd=2,lty=2)
infPts <- which(leverage>2*p/n)

# ** Cook's Distance ----------------

# high Cook's distance => influential
points/outliers
# leverage points with high Cook's distance =>
suspicious influential points & outliers
#                  may need to be deleted -> check
scatterplots

cooks = cooks.distance(x2)

plot(cooks, ylab="Cook's Distance", pch=16,
cex=0.6)
points(infPts, cooks[infPts], pch=17, cex=0.8,
col=2)
susPts <- as.numeric(names(sort(cooks[infPts],
decreasing=TRUE)[1:3]))
text(susPts, cooks[susPts], susPts, adj=c(-0.1,-
0.1), cex=0.7, col=4)

dispersion <- 1
all(abs(cooks - (res.P/(1 - leverage))^2 *
leverage/(dispersion * p) < 1e-15))


#20, 75
mush2[c(20,75),c("season_w" ,"cap.shape_b" ,
"cap.color_n",    "cap.color_r" ,"stem.color_w")]
mush6=mush2[-c(20,75),] #remove these points
#refit
x_r = glm(formula = class ~ season_w +
cap.shape_b +  cap.color_n+
    cap.color_r + stem.color_w , family =
binomial(link = "logit"), mush6)

summary(x_r)

#check fit, overall no significant change
res.P = residuals(x_r, type="pearson")
res.D = residuals(x_r, type="deviance") #or
residuals(fit), by default
res = cbind(res.P, res.D)
colnames(res)= c("Pearson", "Deviance")

summary(res)

boxplot(res, main="Residuals Boxplots")


par(mfrow=c(1,2))
plot(x_r$fitted.values, res.P, pch=16, cex=0.6,
ylab='Pearson Residuals', xlab='Fitted Values')
lines(smooth.spline(x_r$fitted.values, res.P,
spar=0.9), col=2)
abline(h=0, lty=2, col='grey')
plot(x_r$fitted.values, res.D, pch=16, cex=0.6,
ylab='Deviance Residuals', xlab='Fitted Values')
lines(smooth.spline(x_r$fitted.values, res.D,
spar=0.9), col=2)
abline(h=0, lty=2, col='grey')
```