# Homework 2: Generalized Additive Models and Storytelling

Harvard CS 109B, Spring 2017

*Tim Hagmann*

*February 16, 2017*

## Contents



## Problem 1: Heart Disease Diagnosis

In this problem, the task is to build a model that can diagnose heart disease for a patient presented with chest pain. The data set is provided in the files `dataset_1_train.txt` and `dataset_1_test.txt`, and contains 6 predictors for each patient, along with the diagnosis from a medical professional.

### Initialize

In the following code chunk all the necessary setup for the modelling environment is done.

```
## Options
options(scipen = 10)                          # Disable scientific notation
update_package <- FALSE                       # Use old status of packages

## Init files (always execute, eta: 10s)
source("scripts/01_init.R")                   # Helper functions to load packages
source("scripts/02_packages.R")               # Load all necessary packages
source("scripts/03_functions.R")              # Load project specific functions
```

### Load the data

```
## Read data
df_train1 <- read_csv("data/q1/dataset_1_train.txt")
df_test1 <- read_csv("data/q1/dataset_1_test.txt")
```

**Prepare data**

```
# Transform dummys to factor variables
df_train1$Sex <- factor(df_train1$Sex, labels=c("Sex 1", "Sex 2"))
df_train1$ExAng <- factor(df_train1$ExAng, labels=c("No-ExAng", "ExAng"))
df_train1$ChestPain <- factor(df_train1$ChestPain)
df_train1$Thal <- factor(df_train1$Thal)
df_train1$HeartDisease <- factor(df_train1$HeartDisease)

df_test1$Sex <- factor(df_test1$Sex, labels=c("Sex 1", "Sex 2"))
df_test1$ExAng <- factor(df_test1$ExAng, labels=c("No-ExAng", "ExAng"))
df_test1$ChestPain <- factor(df_test1$ChestPain)
df_test1$Thal <- factor(df_test1$Thal)
df_test1$HeartDisease <- factor(df_test1$HeartDisease)
```

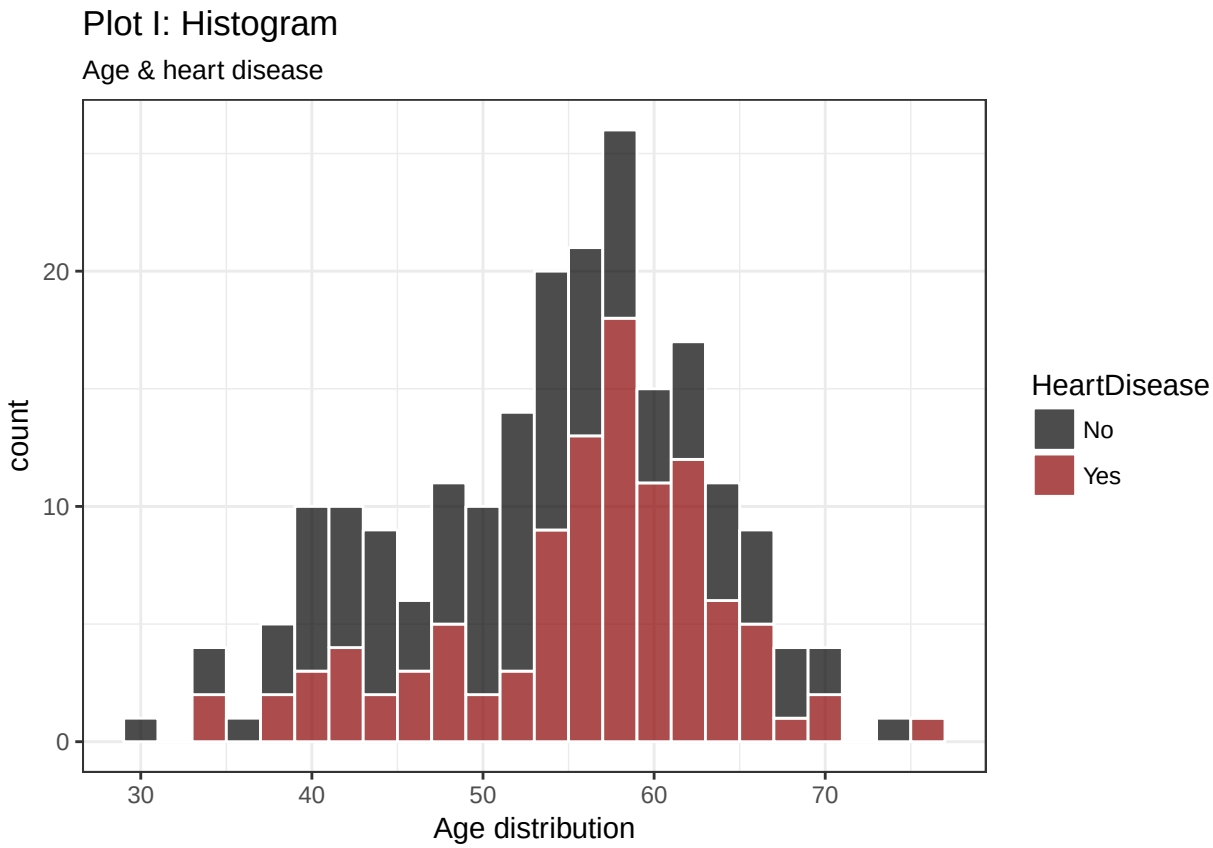
## Visual inspection

By visual inspection, do you find that the predictors are good indicators of heart disease in a patient?

**Visualise the data:**

**Plot and visualize age**

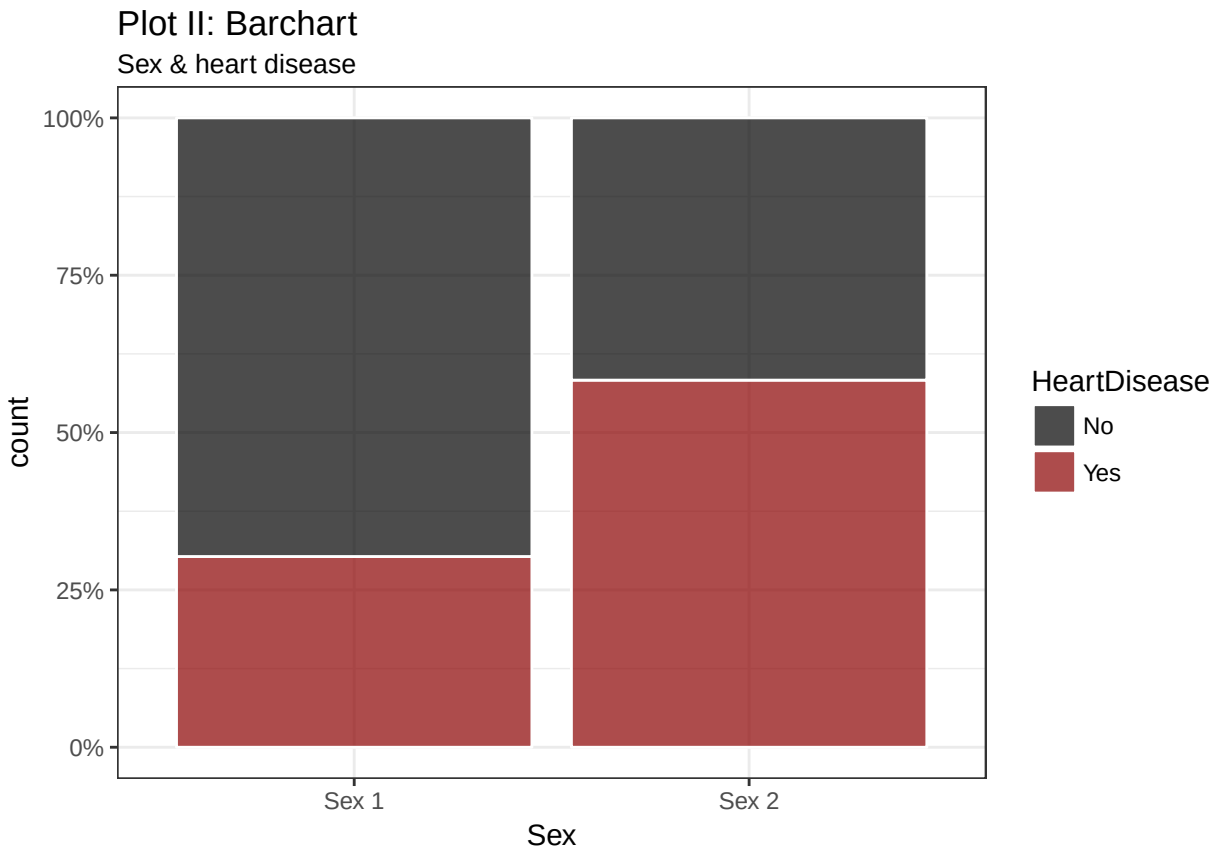
```
ggplot(data=df_train1, mapping=aes(x=Age, fill=HeartDisease)) +
  labs(title="Plot I: Histogram",
       subtitle="Age & heart disease") +
  geom_histogram(binwidth=2, colour="white") +
  scale_fill_manual(values=alpha(c("black", "darkred"), 0.7)) +
  xlab("Age distribution") +
  theme_bw()
```




The distribution appears to be sqewed to the left, i.e., older people appear to have a higher incidents of heart disease, especially above the age of 55.

**Plot and visualize sex**

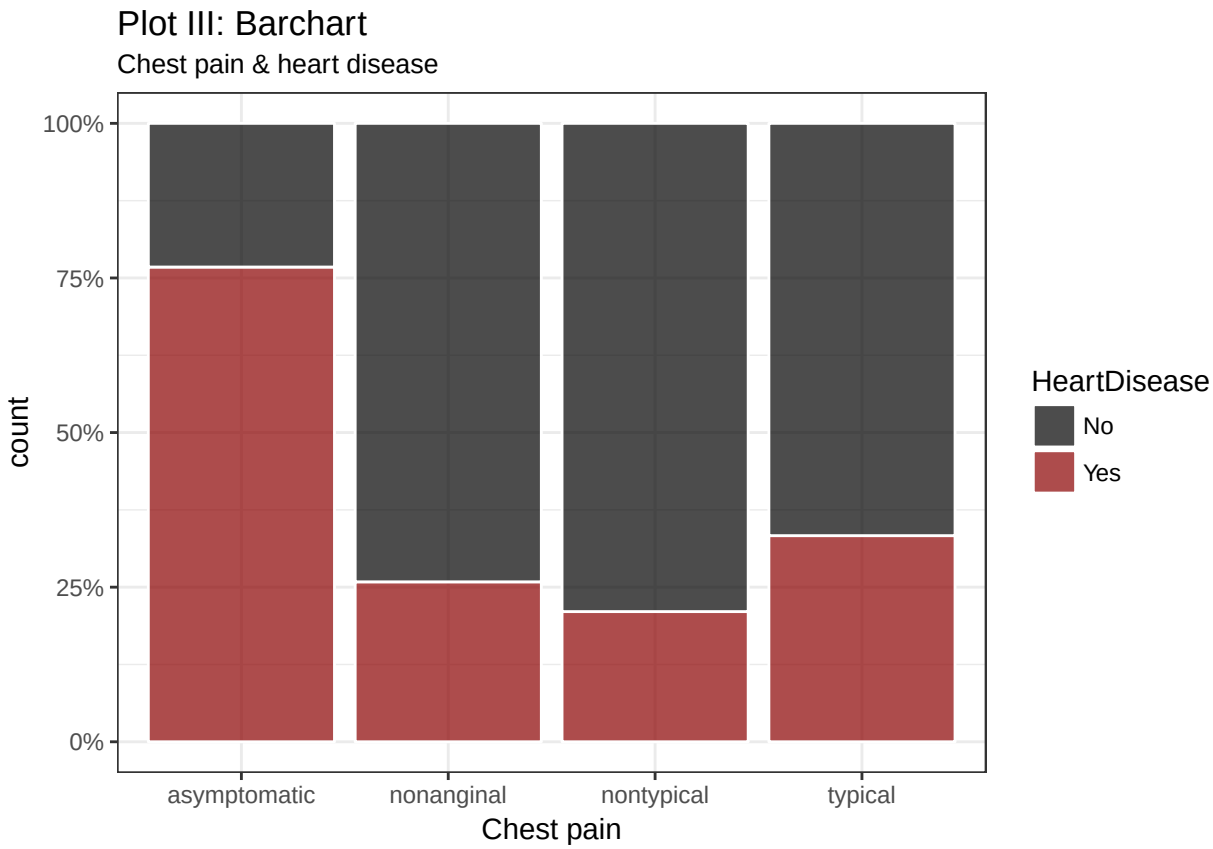
```
ggplot(data=df_train1, mapping=aes(x=Sex, fill=HeartDisease)) +
  labs(title="Plot II: Barchart",
       subtitle="Sex & heart disease") +
  geom_bar(position="fill", color="white") +
  scale_fill_manual(values=alpha(c("black", "darkred"), 0.7)) +
  scale_y_continuous(labels=percent) +
  xlab("Sex") +
  theme_bw()
```




The above plot shows There appears to be a much higher incidence of heart disease in sex 2 compared to sex 1.

**Plot and visualize Chest Pain**

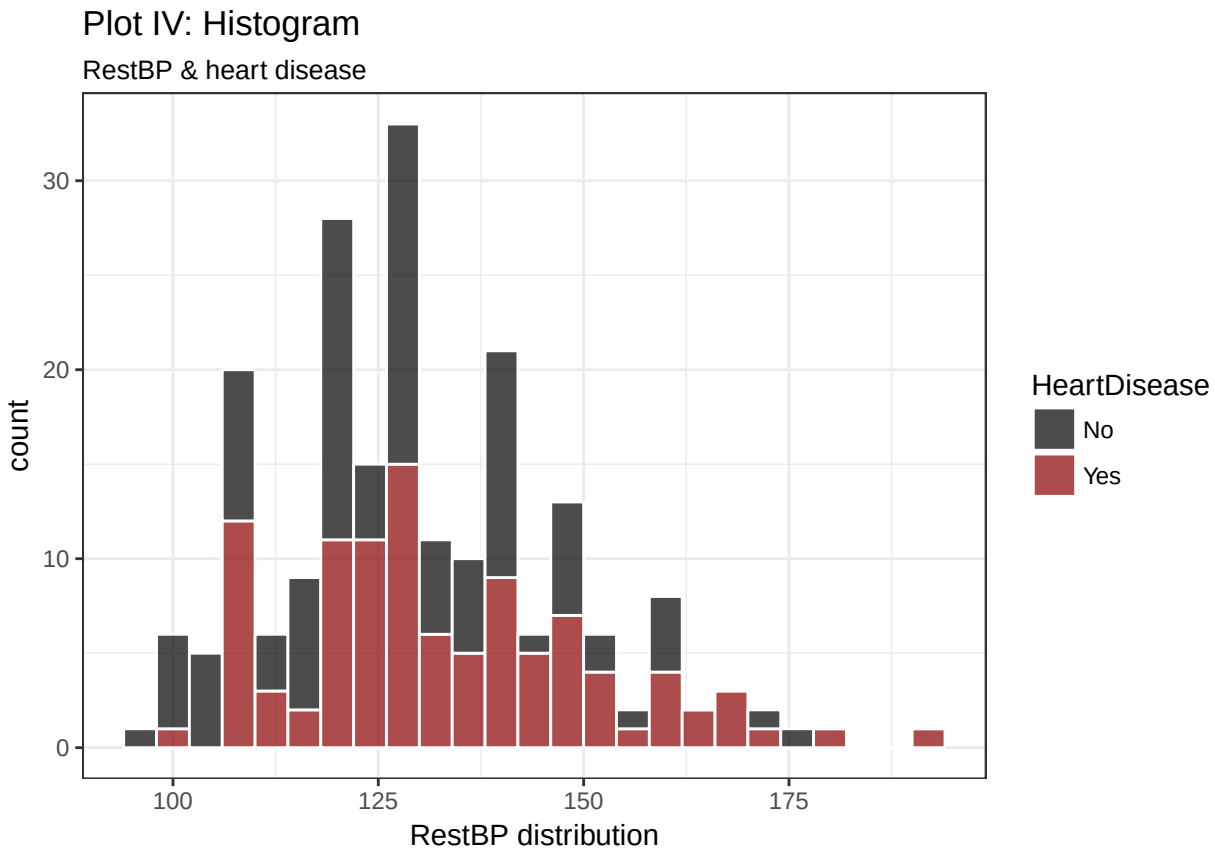
```
ggplot(data=df_train1, mapping=aes(x=ChestPain, fill=HeartDisease)) +
  labs(title="Plot III: Barchart",
       subtitle="Chest pain & heart disease") +
  geom_bar(position="fill", color="white") +
  scale_fill_manual(values=alpha(c("black", "darkred"), 0.7)) +
  scale_y_continuous(labels=percent) +
  xlab("Chest pain") +
  theme_bw()
```




Barchart III above shows, thatindividuals with asymptomatic chest pain have a much higher rate of heart disease than the indiviuals with other pain types.

**Plot and visualize RestBP**

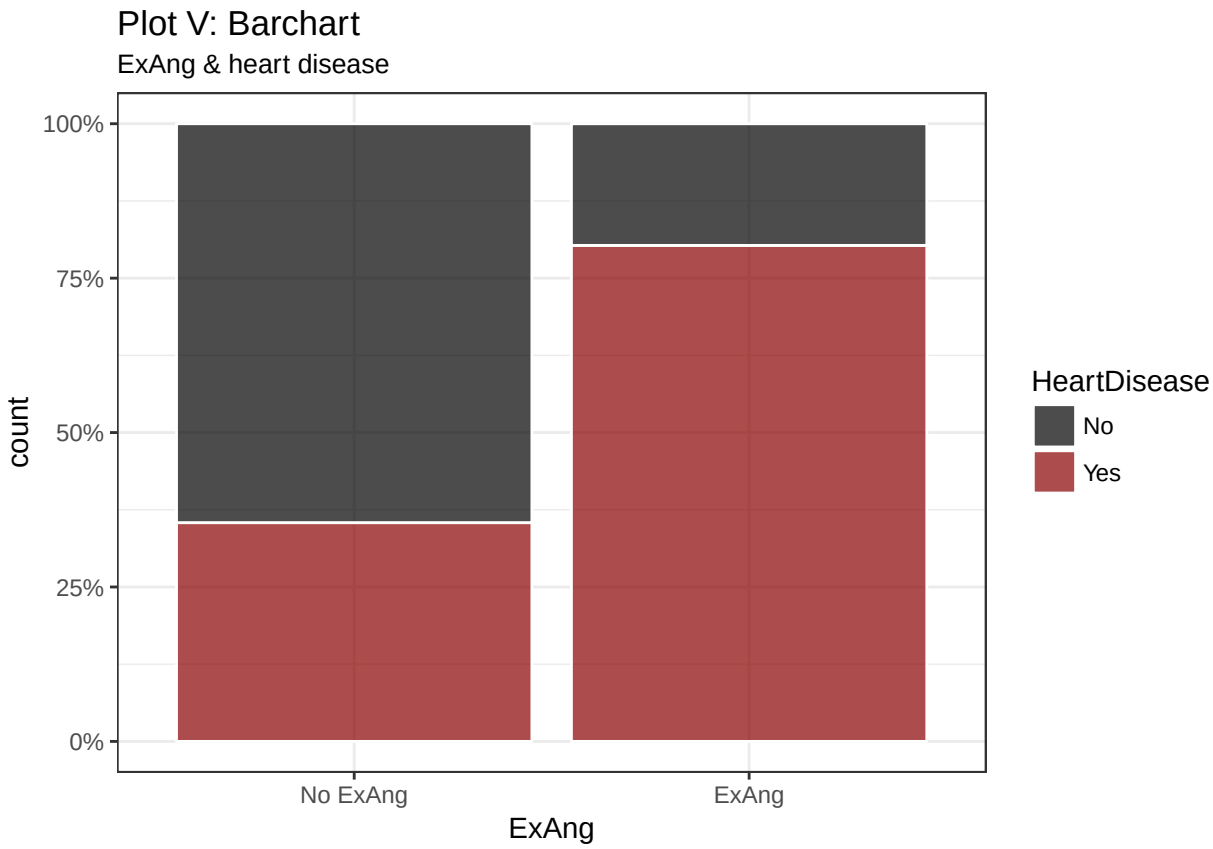
```
ggplot(data=df_train1, mapping=aes(x=RestBP, fill=HeartDisease)) +
  labs(title="Plot IV: Histogram",
       subtitle="RestBP & heart disease") +
  geom_histogram(binwidth=4, colour="white") +
  scale_fill_manual(values=alpha(c("black", "darkred"), 0.7)) +
  xlab("RestBP distribution") +
  theme_bw()
```




Plot IV above shows, that there appears to be a association between higher resting blood pressure values with heart disease.

**Plot and visualize ExAng**

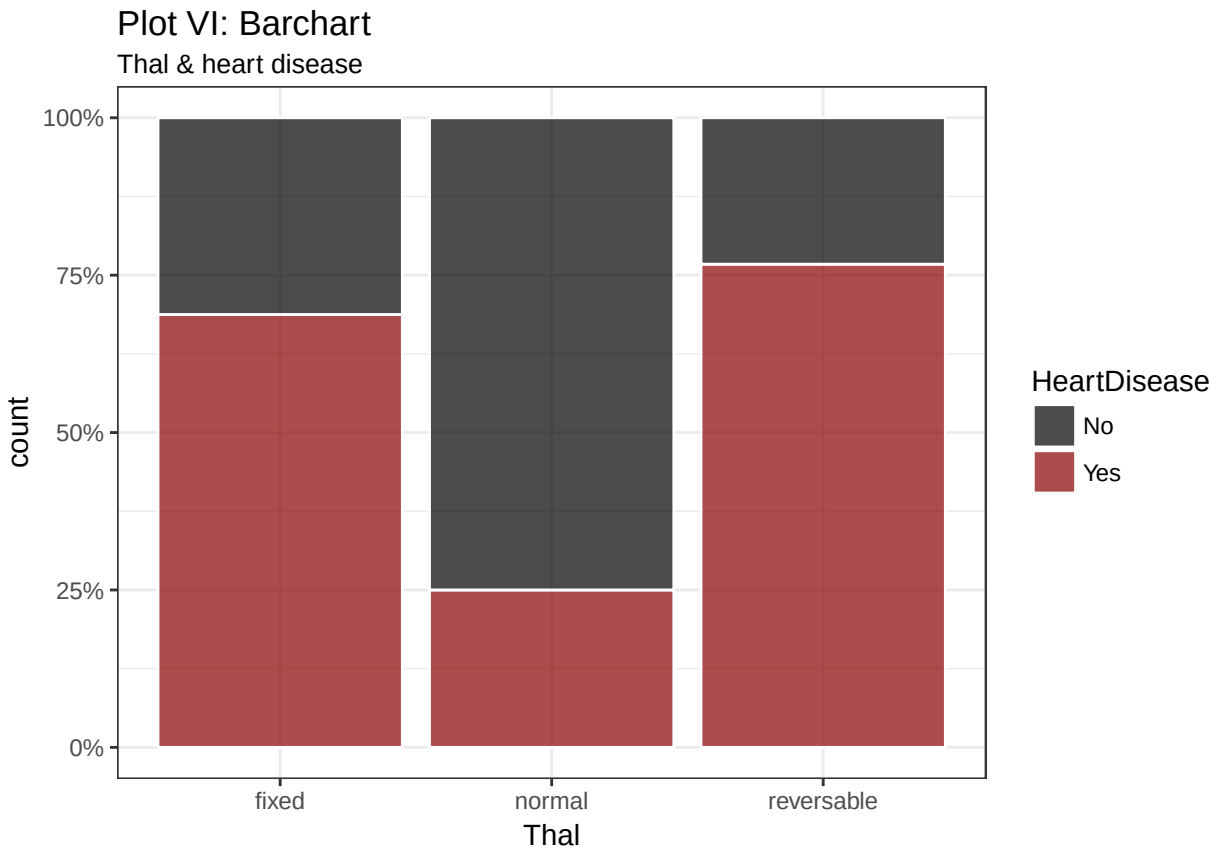
```
ggplot(data=df_train1, mapping=aes(x=factor(ExAng, labels=c("No ExAng", "ExAng")),
                                   fill=HeartDisease)) +
  labs(title="Plot V: Barchart",
       subtitle="ExAng & heart disease") +
  geom_bar(position="fill", color="white") +
  scale_fill_manual(values=alpha(c("black", "darkred"), 0.7)) +
  scale_y_continuous(labels=percent) +
  xlab("ExAng") +
  theme_bw()
```




Plot V shows, that people with ExAng have a much higher incidence of heart disease.

**Plot and visualize Thal**

```
ggplot(data=df_train1, mapping=aes(x=Thal, fill=HeartDisease)) +
  labs(title="Plot VI: Barchart",
       subtitle="Thal & heart disease") +
  geom_bar(position="fill", color="white") +
  scale_fill_manual(values=alpha(c("black", "darkred"), 0.7)) +
  scale_y_continuous(labels=percent) +
  xlab("Thal") +
  theme_bw()
```




Plot VI shows, that people with a normal Thal have a lower heart disease rate than people with a fixed or reversable thal.

## Applying a GAM (generalized additive model)

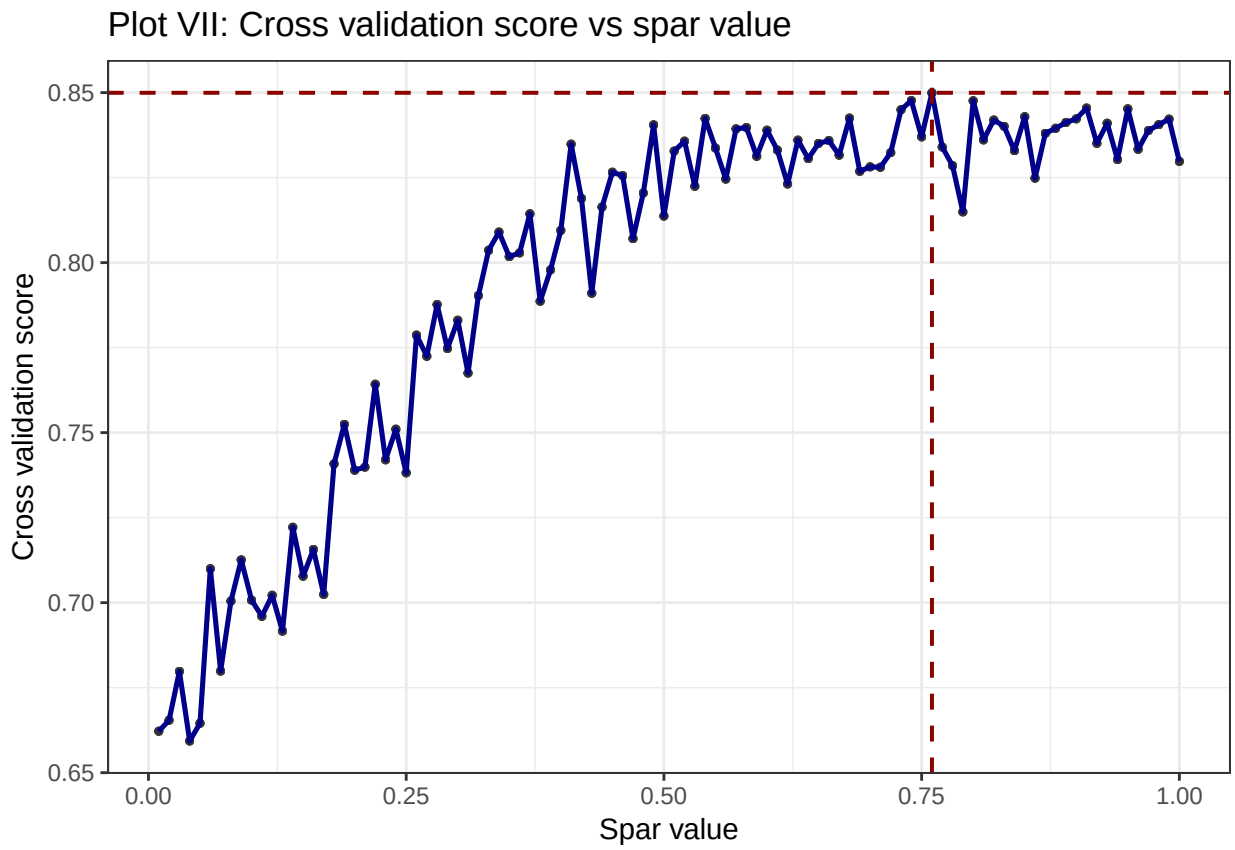
Apply the generalized additive model (GAM) method to fit a binary classification model to the training set and report its classification accuracy on the test set. You may use a smoothing spline basis function wherever relevant, with the smoothing parameter tuned using cross-validation on the training set.

**Tune smoothing parameter with 5-fold cross validation**

```
set.seed(123)
spar_coefs <- c(1:100)/100
gam_cv_scores <- cv_accuracy(df_train1, spar_coefs, 5)
spar_max_coef <- spar_coefs[which.max(gam_cv_scores)]
```

```
ggplot(data.frame(spar_coefs, gam_cv_scores), aes(x = spar_coefs, y = gam_cv_scores)) +
  geom_point(stroke=0, alpha=0.8) +
  geom_line(colour="darkblue", size=0.9) +
  geom_hline(yintercept=max(gam_cv_scores), linetype="dashed",
                color="darkred", size=0.7) +
  geom_vline(xintercept=spar_max_coef, linetype="dashed",
                color="darkred", size=0.7) +
  labs(title="Plot VII: Cross validation score vs spar value") +
  ylab(label="Cross validation score") +
```

```
  xlab("Spar value") +
  theme_bw()
```




The optimal spar parameter appears to be *0.76* with a cross-validation score of *0.85*

**Apply GAM with optimal spar**

```
# Formula
gam_formula <- as.formula(sprintf("HeartDisease ~ s(Age, spar=%1$f) + Sex +
                                  s(RestBP,spar=%1$f) + ExAng + ChestPain + Thal",
                                  spar_max_coef))

# Model
fit_gam <- gam(gam_formula, family = binomial(link="logit"), data=df_train1)

# Prediction
pred_gam <- predict(fit_gam, newdata=df_test1, type="response")

# Accuracy
cm_gam <- as.matrix(table(Actual=df_test1$HeartDisease, Predicted=pred_gam > 0.5))
acc_gam <- sum(diag(cm_gam))/ sum(cm_gam)
```

The GAM model classification accuracy is: 0.8022

### Smoothing splines to categorical predictors

*Would you be able to apply the smoothing spline basis to categorical predictors?*

No, the smoothing spline basis should not be applied to categorical predictors. This is because the weighted average would either produce the predictors actual value or some meaningless value between predictors. If we take the gender as an example where 0 represents female and 1 male, and a weighted average would produces 0.7, we couldn't use this result as it doesn't correspond with either male nor female.

### Difference between `R` and `Python` for dummy handling

*Is there a difference in the way you would handle categorical attributes in `R` compared to `sklearn` in `Python`?*

R has the ability to handle categorical variables as factors. When reading a dataset with the *read.csv* function, R transforms categorical variables automaticly into factor variables. This is done unless the parameter *stringasfactors* is set to *false* or using the *readr* packages. The advantage of R modeling packages is, that most models are able to do one-hot encoding internally. On the other hand, sklearn in Python needs one-hot encoding in order to get dummy variables.

## Plot the smooth

Plot the smooth of each predictor for the fitted GAM. By visual inspection, do you find any benefit in modeling the numerical predictors using smoothing splines?

### Predictor Smoothing

```
p1 <- preplot(fit_gam, terms=sprintf("s(Age, spar = %.2f)", spar_max_coef))[[1]]
df1 <- data.frame(x=p1$x, y=p1$y, se=p1$se.y)
g1 <- ggplot(df1, aes(x=x, y=y)) +
      geom_line(size=0.9) +
      geom_ribbon(aes(ymin=df1$y - df1$se, ymax=df1$y + df1$se),
                  alpha=0.2, fill="black") +
      scale_y_continuous(limits = c(min(df1$y*4), max(df1$y*4))) +
      labs(title="Plot VIII: Age") +
      ylab(label=p1$ylab) +
      xlab(label=p1$xlab) +
      theme_bw()

p2 <- preplot(fit_gam, terms=sprintf("s(RestBP, spar = %.2f)", spar_max_coef))[[1]]
df2 <- data.frame(x=p2$x, y=p2$y, se=p2$se.y)
g2 <- ggplot(df2, aes(x=x, y=y)) +
      geom_line(size=0.9) +
      geom_ribbon(aes(ymin=df2$y - df2$se, ymax=df2$y + df2$se),
                  alpha=0.2, fill="black") +
      scale_y_continuous(limits = c(min(df2$y*4), max(df2$y*4))) +
      labs(title="Plot IX: Rest blood pressure") +
      ylab(label=p2$ylab) +
      xlab(label=p2$xlab) +
      theme_bw()

grid.arrange(g1, g2, nrow=1, ncol=2)
```


**Table 1: Anova for Parametric Effects**

```
pander(summary(fit_gam)[[4]],
       add.significance.stars=TRUE, style='rmarkdown', caption="",
       split.table=Inf, digits=2, justify='center')
```

| | Df | Sum Sq | Mean Sq | F value | Pr(>F) |
|---|---|---|---|---|---|
| **s(Age, spar = 0.76)** | 1 | 2.4 | 2.4 | 2.2 | |
| **Sex** | 1 | 5.9 | 5.9 | 5.5 | * |
| **s(RestBP, spar = 0.76)** | 1 | 0.97 | 0.97 | 0.89 | |
| **ExAng** | 1 | 14 | 14 | 13 | * * * |
| **ChestPain** | 3 | 21 | 7 | 6.5 | * * * |
| **Thal** | 2 | 15 | 7.4 | 6.8 | * * |
| **Residuals** | 194 | 211 | 1.1 | NA | NA |

Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ‘’ 1

It appears that there is no additional benefit trought smoothing splines. This can be seen in the plots above trought there large standard errors for non-zero coefficient values. Furthermore, the summary statistics also show, that the smoothing parameters are not significant.

## Using the likelihood ratio test

Using a likelihood ratio test, compare the fitted GAM with the following models: i) a GAM with only the intercept term ii) a GAM with only categorical predictors iii) a GAM with all predictors entered linearly.

### (i) GAM with only the intercept

```
fit_gam_intercept <- gam(HeartDisease ~ 1, family=binomial(link = "logit"),
                         data=df_train1)
anova_tbl <- anova(fit_gam, fit_gam_intercept, test="Chi")
```

**Table 2: Anova intercept vs. full model**

```
pander(anova_tbl,
       add.significance.stars=TRUE, style='rmarkdown', caption="",
       split.table=Inf, digits=2, justify='center')
```

| Resid. Df | Resid. Dev | Df | Deviance | Pr(>Chi) |
|:---:|:---:|:---:|:---:|:---:|
| 194 | 174 | NA | NA | NA |
| 209 | 291 | -15 | -117 | * * * |

Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ‘’ 1

Table 2 above shows, that including all predictors with smoothing performs significantly better (0.001) than a gam model with only the intercept.

### (ii) GAM with categorical variables

```
fit_gam_cat <- gam(HeartDisease ~ Sex + ChestPain + ExAng + Thal,
                   family=binomial(link="logit"), data=df_train1)
anova_tbl <- anova(fit_gam, fit_gam_cat, test="Chi")
```

**Table 3: Anova categorical variables vs. full model**

```
pander(anova_tbl,
       add.significance.stars=TRUE, style='rmarkdown', caption="",
       split.table=Inf, digits=2, justify='center')
```

| Resid. Df | Resid. Dev | Df | Deviance | Pr(>Chi) |
|:---:|:---:|:---:|:---:|:---:|
| 194 | 174 | NA | NA | NA |
| 202 | 190 | -7.8 | -16 | * |

Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ‘’ 1

Table 3 above shows, that the full model performs significantly better (0.05) than the gam model with only the categorical predictors.

**(iii) GAM with linear predictors**

```
fit_gam_lin <- gam(HeartDisease ~ Age + Sex + ChestPain + RestBP + ExAng + Thal,
                   family=binomial(link="logit"), data=df_train1)
anova_tbl <- anova(fit_gam, fit_gam_lin, test="Chi")
```

**Table 4: Anova linear model vs. full model**

```
pander(anova_tbl,
       add.significance.stars=TRUE, style='rmarkdown', caption="",
       split.table=Inf, digits=2, justify='center')
```

| Resid. Df | Resid. Dev | Df | Deviance | Pr(>Chi) |
|---|---|---|---|---|
| 194 | 174 | NA | NA | NA |
| 200 | 182 | -5.8 | -7.7 | |

Signif. codes: 0 ‘’ ***0.001*** ’’ *0.01* ’’ 0.05 ‘.’ 0.1 ‘’ 1

Table 4 above shows, that the full model doesn't perform significantly better than the gam model with only linear precdictors. The linear model is therefore to be prefered to the full model.

# Problem 2: The Malaria Report

You work for the Gotham Times media organization and have been tasked to write a short report on the World Health Organisation's (WHO) fight against malaria. The WHO Global Malaria Programme (http://www.who.int/malaria/en/) has been working to eliminate the deadly disease over the past several decades, your job is to discuss their work and spotlight the impact they've had. Your writing and graphics should be easily understood by anyone interested in the topic, and not necessarily just physicians and experts.

## Key Facts and Quotes on Malaria

Here are some informative key facts and quotes about Malaria that you may want to include in your report:

- RISK: About 3.2 billion people – almost half of the world's population – are at risk of malaria.
- CASES: 214 million malaria cases reported worldwide in 2015.
- INCIDENCE: 37% global decrease in malaria incidence between 2000 and 2015.
- MORTALITY: 60% decrease in global malaria mortality rates between 2000 and 2015.
- "Malaria is a life-threatening disease caused by parasites that are transmitted to people through the bites of infected female mosquitoes."
- "Young children, pregnant women and non-immune travelers from malaria-free areas are particularly vulnerable to the disease when they become infected."
- "Malaria is preventable and curable, and increased efforts are dramatically reducing the malaria burden in many places."

Many of these facts were pulled from the WHO website, where you can find many more.

## The malaria data

The datasets consist of country-level information for 2015, estimated malaria cases over time.

**Dataset 1**: data/global-malaria-2015.csv This dataset contains observed and suspected malaria cases as well as other detailed country-level information **for 2015** in 100 countries worldwide. The CSV file consists of the following fields:

- WHO_region, Country, Country Code, UN_population
- At_risk - % of population at risk
- At_high_risk - % of population at high risk
- Suspected_malaria_cases
- Malaria_cases - actual diagnosed cases

**Dataset 2**: data/global-malaria-2000-2013.csv This dataset contains information about suspected number of malaria cases in the same 100 countries for the years 2000, 2005, 2010, 2013.

### Load the data

```
## Read data
rm(list=ls())
df_malaria_15 <- read_csv("data/q2/global-malaria-2015.csv")
df_malaria_years <- read_csv("data/q2/global-malaria-2000-2013.csv")
```

### Clean up the data

```
# Join
df_merge <- merge(df_malaria_years, df_malaria_15[, c("Code", "UN_population",
                                                      "Suspected_malaria_cases",
                                                      "WHO_region")],
                  by='Code')

# Gather
df_malaria <- gather(df_merge, Year, Estimated_Malaria_Counts, Y_2000, Y_2005,
                     Y_2010, Y_2013, Suspected_malaria_cases, factor_key=TRUE)

# Renaming
names(df_malaria)[names(df_malaria) == "UN_population"] <- "UN_population_2015"
levels(df_malaria$Year) <- c('2000', '2005', '2010', '2013', '2015')

# Calculate percentage of suspected malaria cases
df_malaria$Percentage <- round(df_malaria$Estimated_Malaria_Counts /
                                 df_malaria$UN_population_2015, 2)
df_malaria$Percentage[is.na(df_malaria$Percentage)] <- 0
rm(df_merge, df_malaria_years)

df_malaria_15 <- df_malaria[df_malaria$Year == "2015", ]
```

### Table I: Inspect the data

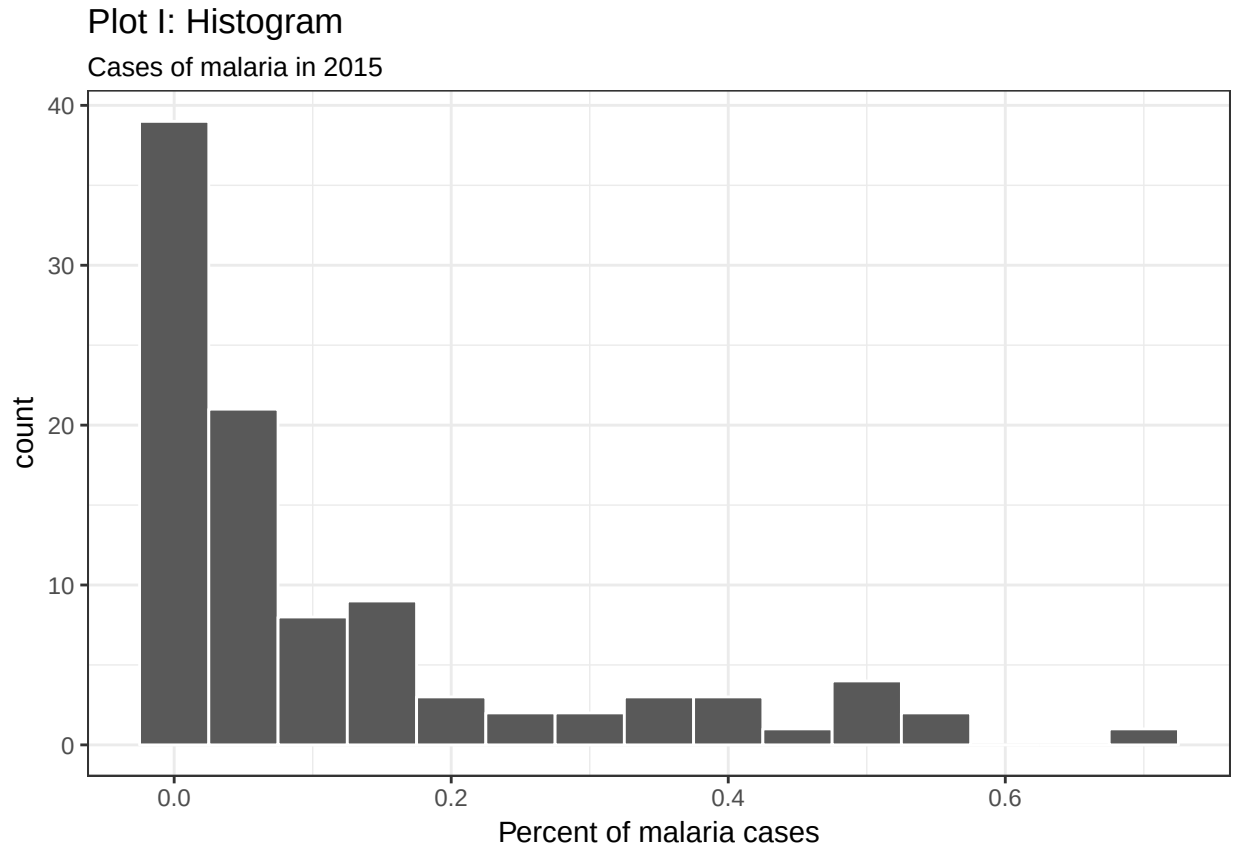
```
pander(table(df_malaria_15$WHO_region[!(df_malaria_15$Estimated_Malaria_Counts < 1 |
                                          is.na(df_malaria_15$Estimated_Malaria_Counts))]),
       add.significance.stars=TRUE, style='rmarkdown', caption="",
       split.table=Inf, digits=2, justify='center')
```

| African | Eastern Mediterranean | European | Region of the Americas | South-East Asia | Western Pacific |
|---|---|---|---|---|---|
| 42 | 6 | 6 | 20 | 10 | 10 |

In 2015, 94 countries and areas had ongoing malaria transmission. Out of those 94 countries 42 are in Africa. This is followed by some regions in America.

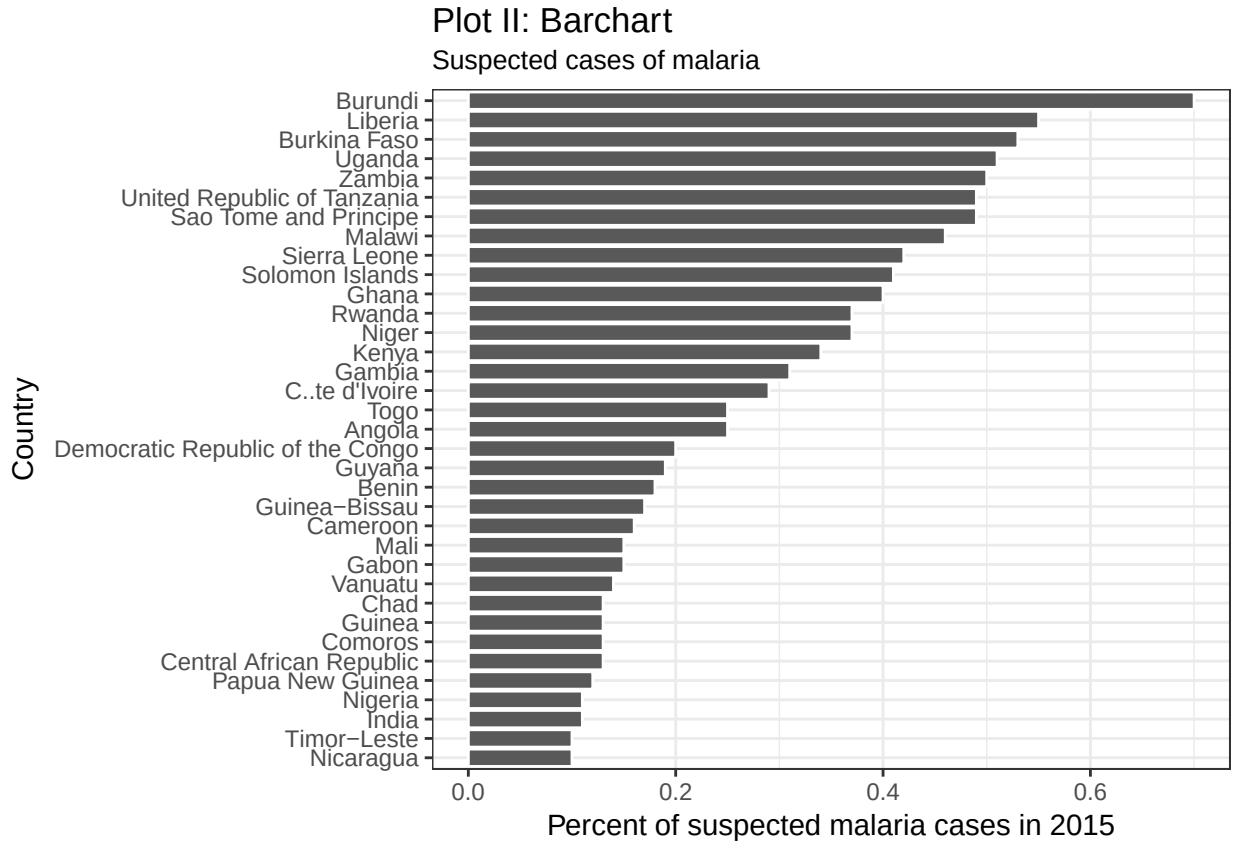
**Visualize malaria data**

```
# Change order
df_malaria_15$Country <- factor(df_malaria_15$Country,
                                levels=df_malaria_15$Country[
                                  order(df_malaria_15$Percentage)])

# Plot
ggplot(data=df_malaria_15, aes(Percentage)) +
  labs(title="Plot I: Histogram",
       subtitle="Cases of malaria in 2015") +
  geom_histogram(binwidth=0.05, colour="white") +
  xlab("Percent of malaria cases") +
  theme_bw()
```




```
# Countries with most suspected malaria cases
ggplot(data=df_malaria_15[df_malaria_15$Percentage >= 0.1, ],
```

```
      mapping=aes(x=Country, y=Percentage)) +
   labs(title="Plot II: Barchart",
      subtitle="Suspected cases of malaria") +
 geom_bar(stat="identity", colour="white") +
 theme_bw() +
 ylab("Percent of suspected malaria cases in 2015") +
 coord_flip()
```




As can be seen from the above plot, most cases of malaria occur in African countries. That is why we're concentrating on this continent for the analyis from this point on.

### Data cleanup

```
# Analysis for Africa
df_malaria <- df_malaria[df_malaria$WHO_region == "African", ]

# Spreading the data
df_malaria_spread <- spread(df_malaria[, c("Code", "Country", "Year", "UN_population_2015",
                                           "WHO_region", "Estimated_Malaria_Counts")],
                            key=Year, value=Estimated_Malaria_Counts)

# Renaming
names(df_malaria_spread) <- c("Code", "Country", "UN_population_2015",
                              "WHO_region", "Y_2000", "Y_2005", "Y_2010",
                              "Y_2013", "Y_2015")
```

```
# Cleanup
df_malaria_spread$Y_2000[df_malaria_spread$Y_2000 == 0] <- NA
df_malaria_spread$Y_2015[df_malaria_spread$Y_2015 == 0] <- NA
df_malaria_spread$perc_change <- round(df_malaria_spread$Y_2015 * 100 /
                                       df_malaria_spread$Y_2000, 2)
df_malaria_spread$perc_change[is.na(df_malaria_spread$perc_change)] <- 0
```

**Table II: Outliers**

```
pander(df_malaria_spread[df_malaria_spread$perc_change > 500, ],
       style='rmarkdown', caption="",
       digits=2, justify='center')
```

Table 6: (continued below)

| | Code | Country | UN_population_2015 | WHO_region |
|---|---|---|---|---|
| **12** | CPV | Cabo Verde | 513906 | African |
| **42** | ZAF | South Africa | 53969054 | African |

Table 7: Table continues below

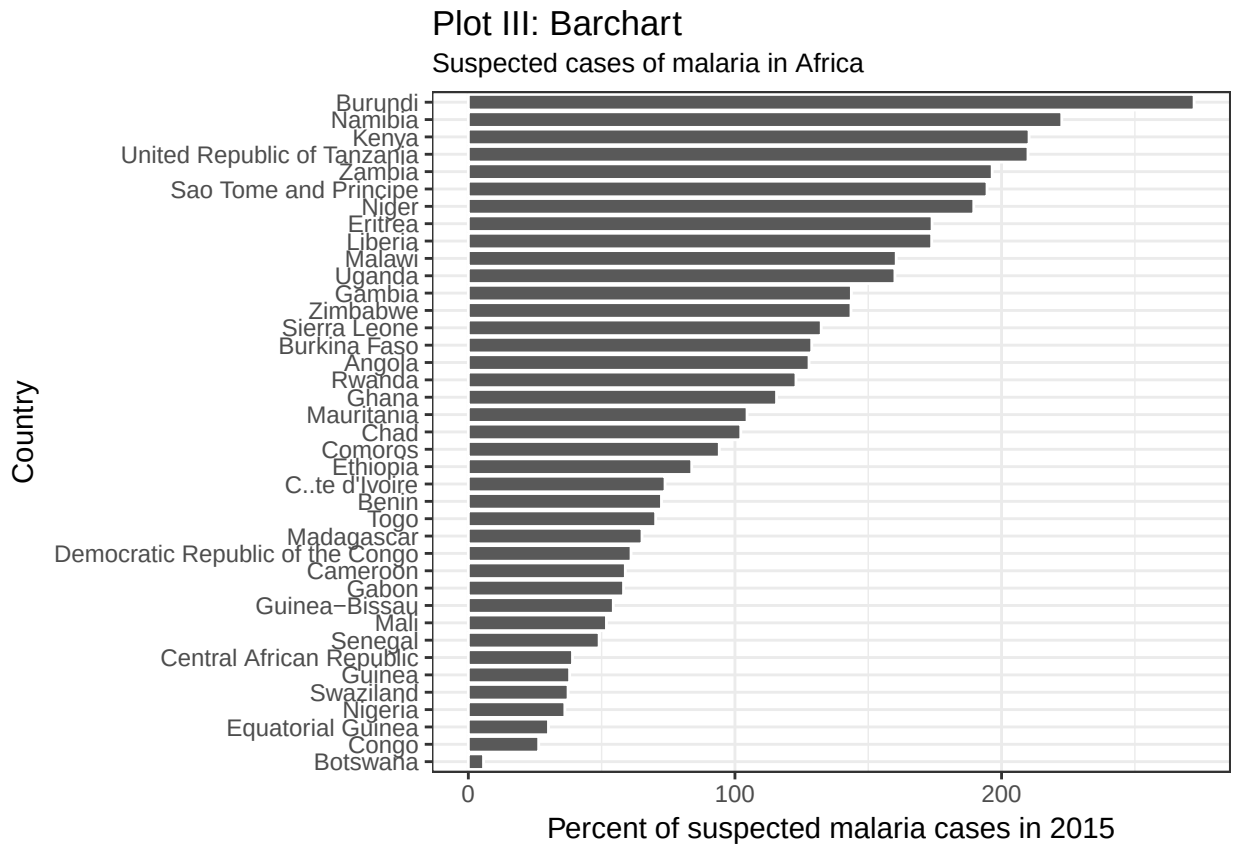
| | Y_2000 | Y_2005 | Y_2010 | Y_2013 | Y_2015 |
|---|---|---|---|---|---|
| **12** | 490 | 220 | 140 | NA | 6894 |
| **42** | 39000 | 17000 | 17000 | 19000 | 543196 |

| | perc_change |
|---|---|
| **12** | 1407 |
| **42** | 1393 |

The data for the two Countries Cabo Verde and South Africa appear to be extremly large. It is very probable that those numbers are not correct (e.g., a mistake in the data). that is why they are excluded from the following plot.

```
# Change order
df_malaria_spread$Country <- factor(df_malaria_spread$Country,
                                    levels = df_malaria_spread$Country[
                                      order(df_malaria_spread$perc_change)])

# Countries with most suspected malaria cases
ggplot(data=df_malaria_spread[df_malaria_spread$perc_change >= 1 &
                                df_malaria_spread$perc_change <= 300, ],
       mapping=aes(x=Country, y=perc_change)) +
  geom_bar(stat="identity", colour="white") +
  labs(title="Plot III: Barchart",
       subtitle="Suspected cases of malaria in Africa") +
  theme_bw() +
  ylab("Percent of suspected malaria cases in 2015") +
```

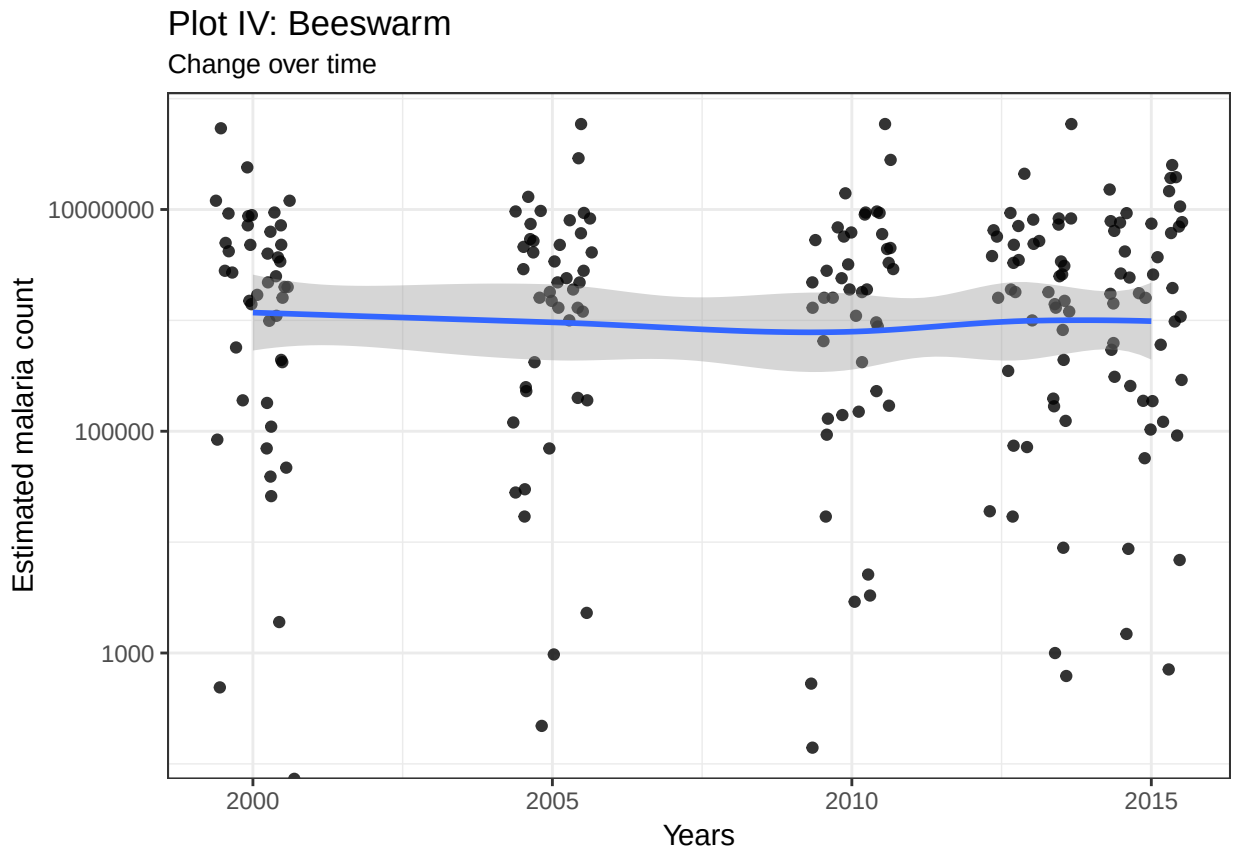
```
  coord_flip()
```




The countries with the hightest percentage increase in suspected Malaria cases are Burundi, Namibia and Kenya, the countries with the lowest increases are Equatorial Guinea, Congo and Botswana.

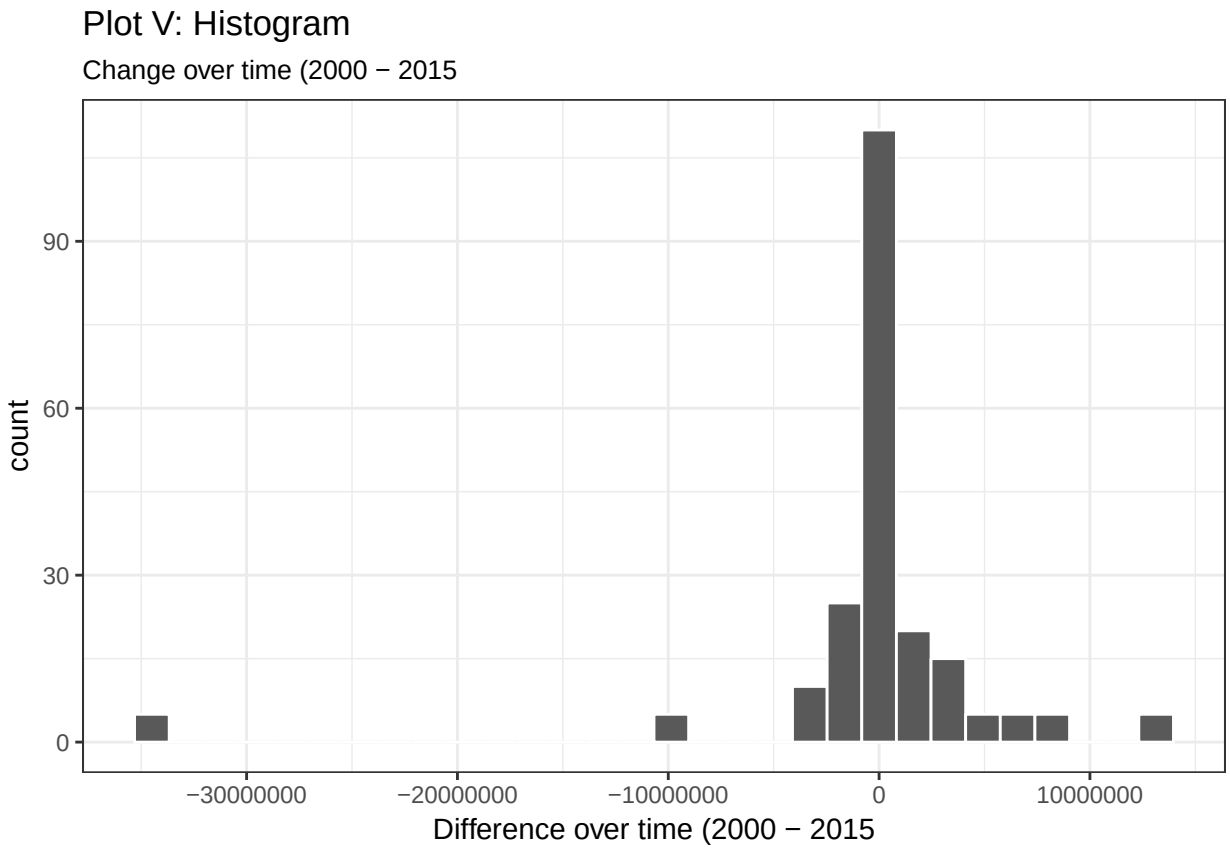
### Change over time

```
## Beeswarm plot
df_malaria$Year <- as.numeric(as.character(df_malaria$Year))
ggplot(df_malaria, aes(Year, Estimated_Malaria_Counts)) +
  labs(title="Plot IV: Beeswarm",
       subtitle="Change over time") +
  geom_jitter(color="black", alpha = 0.8, width = 0.7) +
  geom_smooth(method="loess") +
  xlab("Years") +
  ylab("Estimated malaria count") +
  scale_y_log10() +
  theme_bw()
```




It appears that the estimated malaria count stays stable over the time. We can check if the stability can also be observert over the individual countries trough a histogram of the differences.

```
df_malaria$difference <- df_malaria$Estimated_Malaria_Counts[df_malaria$Year == 2015] -
  df_malaria$Estimated_Malaria_Counts[df_malaria$Year == 2000]

ggplot(data=df_malaria, mapping=aes(x=difference)) +
  labs(title="Plot V: Histogram",
  subtitle="Change over time (2000 - 2015") +
  geom_histogram(bins=30, colour="white") +
  xlab("Difference over time (2000 - 2015") +
  theme_bw()
```




The above plot shows, that even tought that the malaria count is more or less stable over time, the story for some countries is different. A large majority doesn't move by much. However, some countries experience a large change in the overall malaria numbers.

## The funding data

The datasets include the funding values and sources over time.

**Dataset 3**: data/global-funding.csv This dataset contains the total funding for malaria control and elimination (in millions USD) provided by donor governments, multilateral organizations, and domestic sources between 2005 and 2013.

### Load the data

```
## Read data
df_global_funding <- read_csv("data/q2/global-funding.csv")
```

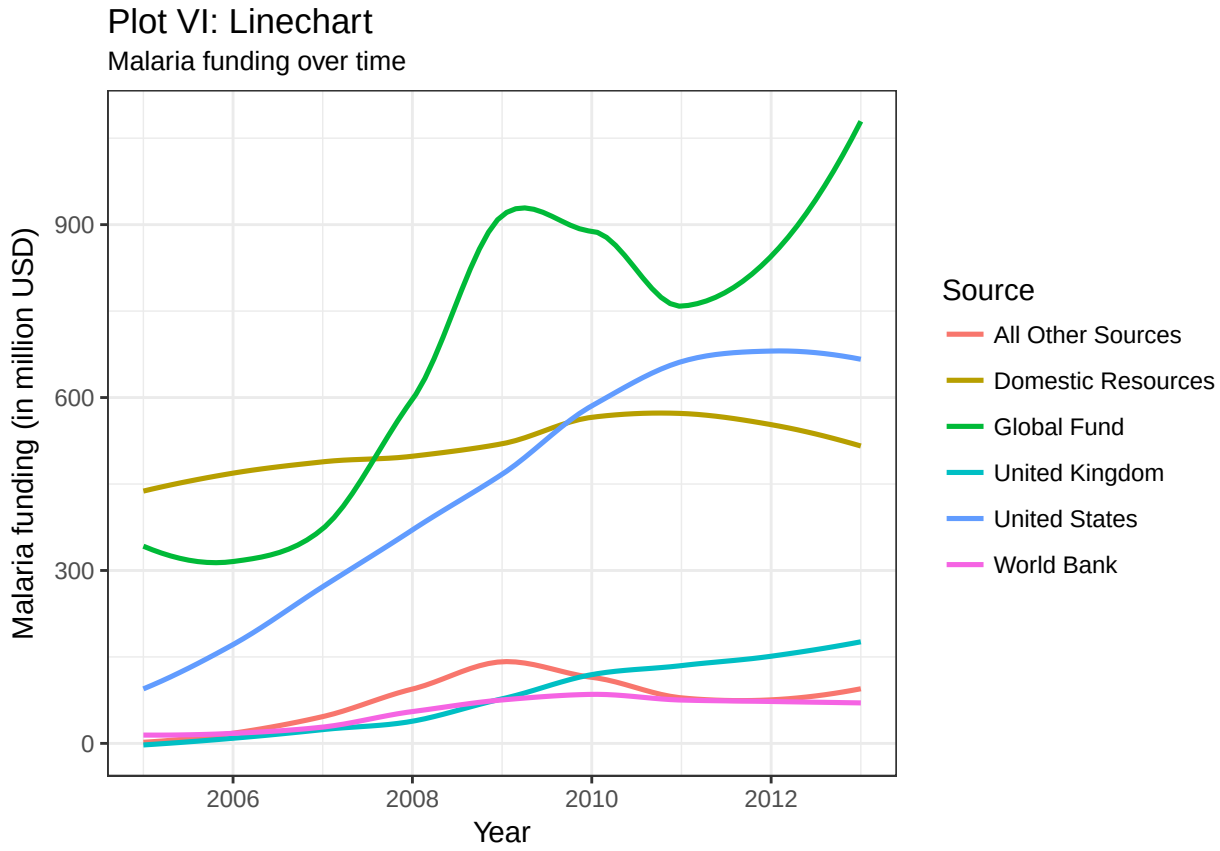
### Clean up the data

```
# Gather
names(df_global_funding) <- c("Source", paste0("X", names(df_global_funding)[2:10]))
df_global_funding <- gather(df_global_funding, Year, Amount, X2005:X2013, factor_key=TRUE)
levels(df_global_funding$Year) <- c('2005', '2006', '2007', '2008', '2009',
```

```
                                    '2010', '2011', '2012', '2013')

# Recode NA's
df_global_funding$Amount[is.na(df_global_funding$Amount)] <- 0
df_global_funding$Source <- factor(df_global_funding$Source)

# Remove total
df_global_funding <- df_global_funding[!df_global_funding$Source == "Total", ]
```

**Visualize the data**

```
# Countries with most suspected malaria cases
df_global_funding$Year <- as.numeric(as.character(df_global_funding$Year))
ggplot(data = df_global_funding, aes(x=Year, y=Amount, group=Source, colour=Source)) +
  geom_smooth(aes(group=Source), method="loess", se=FALSE, size=0.9) +
  theme_bw() +
  labs(title="Plot VI: Linechart",
       subtitle="Malaria funding over time") +
  ylab("Malaria funding (in million USD)") +
  xlab("Year") +
  theme(panel.background=element_rect(fill="transparent"),
        plot.background=element_rect(fill="transparent"))
```




As can be seen , the global fund is contributing the most to followed by the United States. The least amount of money is contributed by the World Bank. Furthermore, there appears to be an upward trend where more

and more money is spend on malaria prevention.

### Geographic data

**Dataset 4**: data/africa.topo.json The TopoJSON file (extension of GeoJSON) contains the data of the boundaries for the African countries.

### Load and clean up the mapping data

```
## Load shape file
shp_africa <- readOGR("data/q2/africa.topo.json")
```

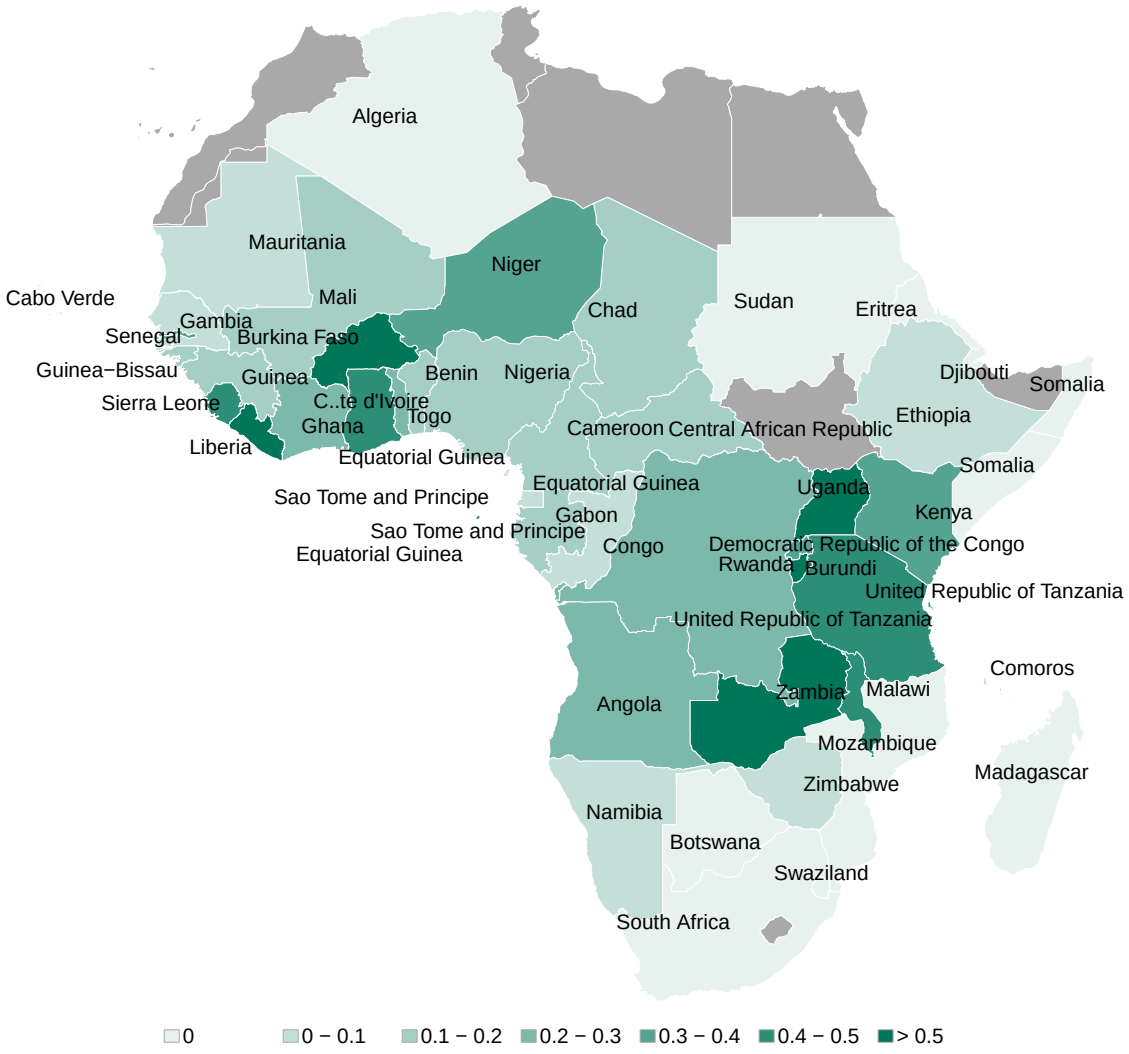
```
## OGR data source with driver: GeoJSON
## Source: "data/q2/africa.topo.json", layer: "collection"
## with 67 features
## It has 64 fields
```

```
## Colors
col_vec <- c(sanCol("green1", alpha = 255),
             sanCol("green1", alpha = 210),
             sanCol("green1", alpha = 170),
             sanCol("green1", alpha = 130),
             sanCol("green1", alpha = 90),
             sanCol("green1", alpha = 60),
             sanCol("green1", alpha = 25))
col_vec <- rev(as.character(unlist(col_vec)))

## Break points (with kmeans)
brks_anzahl <- classIntervals(df_malaria_15$Percentage, n=7, style="kmeans")$brks
brks_anzahl <- c(0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6)
df_malaria_15$color <- col_vec[findInterval(df_malaria_15$Percentage,
                                            brks_anzahl, all.inside=TRUE)]

# Merge
shp_africa <- merge(shp_africa, df_malaria_15, by.x="sov_a3", by.y="Code", all.x=TRUE)

# Missings
shp_africa$color[is.na(shp_africa$color)] <- "darkgrey"

# Legend text
legend_txt <- leglabs(paste0(round(brks_anzahl, 1), ""), under="", over=">")
legend_txt[1] <- "0"
```

### Plotting

```
plot(shp_africa, col=shp_africa$color, border="white", bg="transparent",
     main="Percent of malaria cases in Africa", cex.main=3, lwd=0.5)
pointLabel(coordinates(shp_africa), labels=shp_africa$Country, cex=1.5)
legend("bottom", fill=col_vec, cex=1.5, horiz=TRUE,
       legend=legend_txt,
```

```
      bty="n", x.intersp = 0.2, y.intersp = 0.2,
      border="darkgrey", text.col="black")
```




The above plot shows what we already saw in the barchart. The countries with the hightes suspected malaria incidents are Burundi, Liberia, Burkoina Faso, Uganda an Zambia. It can be seen that the center of Africa has a much higher rate of malaria than the northern and southern bit of africa. Overall, it can be seen that Sub-Saharan Africa carries a disproportionately high share of the global malaria burden. The grey areas mark the teritorys where no data is available.