---
title: "Cancerous Cells are Large in Size, and High in Concavity and Compactness"
author: "Alina Ahmed"
format: html
editor: visual

First I downloaded the packages I needed for my project.

```{r}
library(ggplot2)
library(dplyr)
library(janitor)
library(tidyr)
library(titanic)
```

My Dataset Path I used:

```{r}
data = read.csv("/Users/alinaahmed/Documents/DRSP Lung Cancer Research/Alina A. Project 2024/Data Coding for Breast Cancer/breast_cancer_classification_data.csv")
```

Data Cleaning:

```{r}
col_na_count <- colSums(is.na(data))
print(col_na_count)
# Remove the 'X' column

data = clean_names(data)
names(data)

#569 NA values found
nrow(data)

data = na.omit(data)
str(data) #'data.frame':        0 obs. of  33 variables:
nrow(data)
mean(data$perimeter_mean)
mean(data$radius_mean)
mean(data$area_mean)


# everything is fine, no missing values so far

summary(data)
```

In order to remove possible skews/biases in the results, I took 100 samples of benign cases, as well as 100 samples of malignant cases.

```{r}

malignant_data<- filter(breast_cancer_classification_data, diagnosis == "M")
benign_data <- filter(breast_cancer_classification_data, diagnosis == "B")

# Randomly sample 100 rows from each filtered dataset
malignant_data <- sample_n(malignant_data, 100)
benign_data <- sample_n(benign_data, 100)

# Combine the samples into one dataframe
final_sample <- bind_rows(malignant_data, benign_data)
```

```
# Check the dimensions of the final sample
dim(data)
# Check the distribution of diagnoses in the final sample
table(data$diagnosis)
```

Bar Plot used to calculate the range of Tumor Perimeters

```{r}
ggplot(final_sample, aes(x=diagnosis, y=perimeter_mean))+
  geom_bar(stat="summary", fill="lightsalmon", alpha=1)+
  geom_errorbar(stat = "summary", width=0.2,colour="black", size=0.5)+
  theme_minimal() +
  labs(title="Bar Graph of Average Tumor Perimeter Based on Diagnosis",
       x="Diagnosis",
       y="Perimeter Mean") +
  ylim(0, 150)

```

Bar Plot used to calculate the range of Tumor Radius

```{r}
ggplot(final_sample, aes(x=diagnosis, y=radius_mean))+
  geom_bar(stat = "summary", fill ="olivedrab2") +
  geom_errorbar(stat = "summary", width=0.2,colour="black", size=0.5)+
  theme_minimal() +
  labs(title="Bar Graph of Average Tumor Radius Based on Diagnosis",
       x="Diagnosis",
       y="Average Radius") +
  ylim(0, 25)
```

Bar Plot used to calculate the range of Tumor Area

```{r}
ggplot(final_sample, aes(x=diagnosis, y=area_mean))+
  geom_bar(stat = "summary", fill ="plum1") +
  geom_errorbar(stat = "summary", width=0.2,colour="black", size=0.5)+
  theme_minimal() +
  labs(title="Bar Graph of Average Tumor Area Based on Diagnosis",
       x="Diagnosis",
       y="Area Mean") +
  ylim(0, 1000)
```

I wanted to figure out which variables I wanted to include in my heat map based on my findings

```{r}
selected_data <- final_sample[, c("perimeter_mean", "radius_mean",
"area_mean","texture_mean", "concavity_mean", "smoothness_mean","compactness_mean",
"symmetry_mean")]
correlation_matrix <- cor(selected_data)
correlation_melted <- melt(correlation_matrix)
```

```{r}
ggplot(correlation_melted, aes(x=Var2, y=Var1, fill=value)) +
  geom_tile() +
  scale_fill_gradient2(low="oldlace", high="palegreen4", mid="white", midpoint=0,
limit=c(-1,1)) +
  theme_minimal() +
  labs(x="Features", y= "Features", fill="Correlation") +
```

```
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Code for a Chi-Square Test to figure out the exact P-value of my findings

```{r}
print(str(selected_data))

selected_data

# Create a data frame from the main data set.
statistical_data1 = data.frame(selected_data$area_mean,selected_data$concavity_mean)

# Create a contingency table with the needed variables.
statistical_data1 = table(selected_data$area_mean,selected_data$concavity_mean)

print(statistical_data1)

# applying chisq.test() function
print(chisq.test(statistical_data1))
### X-squared = 288198, df = 288368, p-value = 0.588

statistical_data2 =
data.frame(selected_data$concavity_mean,selected_data$compactness_mean)
print(statistical_data2)
print(chisq.test(statistical_data2))

## X-squared = 6.3622, df = 568, p-value = 1
```

Made a training set data and testing set data (splitting).

```{r}
 library(tidyverse)
library(infer)
library(mosaic)
library(Stat2Data)
library(skimr)


view(data)
set.seed(569)
which_train <- sample(1:569, size = 512, replace = FALSE)

training <- data %>%
  slice(which_train)

testing <- data %>%
  slice(-which_train)
```

K-fold cross validation test

```{r}
library(caret)
library(randomForest)
# Load the dataset
data <-  read.csv("/Users/alinaahmed/Documents/DRSP Lung Cancer Research/Alina A. Project
2024/Data Coding for Breast Cancer/breast_cancer_classification_data.csv")

# Drop column X from the dataset
data <- data[, !(names(data) %in% "X")]

# Convert the diagnosis column to a factor
data$diagnosis <- as.factor(data$diagnosis)
```

```
str(data)

# Check for missing values
missing_counts <- sapply(data, function(x) sum(is.na(x)))
print(missing_counts)

# Remove rows with any missing values
data_clean <- na.omit(data)



# Verify that the column has been dropped
print(names(data_clean))

# Verify there are no missing values
print(sum(is.na(data_clean)))

# Set up training control for k-fold cross-validation
k <- 10
train_control <- trainControl(method = "cv", number = k)

# Train the model with cleaned data
model <- train(diagnosis ~ ., data = data_clean, method = "rf", trControl = train_control)

# Print the model results
print(model)
```

Logistical Model Test

```{r}
# Installing the package
install.packages("dplyr")

# Loading package
library(dplyr)

# Summary of dataset in package
summary(data_clean)

# Installing the package

# For Logistic regression
install.packages("caTools")

# For ROC curve to evaluate model
install.packages("ROCR")

# Loading package
library(caTools)
library(ROCR)

# Splitting dataset
split <- sample.split(data_clean, SplitRatio = 0.8)
split

train_reg <- subset(data_clean, split == "TRUE")
test_reg <- subset(data_clean, split == "FALSE")

# Training model

logistic_model <- glm(diagnosis~ concavity_mean + perimeter_mean,
                      data = train_reg,
                      family = "binomial")
```

```
logistic_model

# Summary
summary(logistic_model)
```