Untitled

2023-05-17

## Hypothesis testing and Chi-square connections

# This sets up the data  
status <- c("authentic", "forgery", "status unconfirmed",  
"authentic", "forgery", "status unconfirmed")  
source <- c("gallery", "gallery", "gallery",  
"private collection", "private collection", "private collection")  
count <- c(61, 10, 23, 37, 24, 15)  
# Data frame format  
forgery\_counts <- data.frame(status, source, count)  
# Table format  
forge.tbl <- xtabs(count ~ status + source, data = forgery\_counts)

# 1. Perform chi-square test

# Chi-square test  
chi\_square <- chisq.test(forge.tbl)  
  
# Print the test results  
print(chi\_square)

##   
## Pearson's Chi-squared test  
##   
## data: forge.tbl  
## X-squared = 11.55, df = 2, p-value = 0.003104

Null hypothesis (H0): The status and source variables are independent.  
Alternative hypothesis (Ha): The status and source variables are dependent.

# 2. Interpretation of the p-value

The p-value obtained from the Chi-square test represents the probability of observing the data or more extreme results under the assumption that the null hypothesis is true. If the p-value is small (usually less than the significance level, commonly 0.05), we reject the null hypothesis in favor of the alternative hypothesis. In this case, a small p-value would indicate evidence that the status and source variables are dependent.

Regarding the type of error, if we reject or fail to reject the p-value based on a 1% threshold, it would correspond to a Type I error. Type I errors occur when we incorrectly reject the null hypothesis, assuming there is a significant relationship when there isn’t one.

# Fitting two Poisson models representing the null and alternative hypotheses

# Fit Poisson models  
null\_model <- glm(count ~ status + source, data = forgery\_counts, family = "poisson")  
alternative\_model <- glm(count ~ status \* source, data = forgery\_counts, family = "poisson")  
  
# Compare models using anova  
anova\_result <- anova(null\_model, alternative\_model, test = "Rao")  
  
# Print the test results  
print(anova\_result)

## Analysis of Deviance Table  
##   
## Model 1: count ~ status + source  
## Model 2: count ~ status \* source  
## Resid. Df Resid. Dev Df Deviance Rao Pr(>Chi)   
## 1 2 11.665   
## 2 0 0.000 2 11.665 11.55 0.003103 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# Forging ahead

# Load libraries  
library(ggplot2) # For data visualization  
library(mgcv) # For GAM modeling

## Loading required package: nlme

## This is mgcv 1.8-42. For overview type 'help("mgcv-package")'.

library(car) # For diagnostic plots

## Loading required package: carData

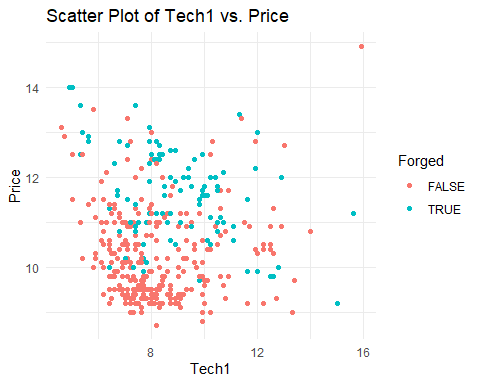
# B. Load the data  
art <- read.csv("C:/Users/User/Downloads/reallynotforplanningacrime.csv")  
head(art)

## title forged tech1 tech2  
## 1 22 FALSE 6.8 0.99498  
## 2 7/11 FALSE 9.8 0.99726  
## 3 A Bar at the Folies-Bergère TRUE 6.6 0.99520  
## 4 A New Zealand River TRUE 10.2 0.99680  
## 5 A Sunday Afternoon on the Island of La Grande Jatte TRUE 7.4 0.99736  
## 6 A Sunday on La Grande Jatte TRUE 15.6 1.00320  
## tech3 tech4 price  
## 1 3.35 0.62 10.4  
## 2 3.15 0.58 9.8  
## 3 3.52 0.56 12.3  
## 4 3.16 0.78 12.5  
## 5 3.58 0.69 10.8  
## 6 2.95 0.68 11.2

# C. Calculate proportion of forged paintings  
proportion\_forged <- mean(art$forged)  
proportion\_forged\_comment <- paste0("Approximately ", round(proportion\_forged \* 100, 2), "% of the paintings were forged.")  
proportion\_forged\_comment

## [1] "Approximately 34.94% of the paintings were forged."

# D. Exploratory data analysis (EDA)  
# Example: Scatter plot of tech1 vs. price  
ggplot(art, aes(x = tech1, y = price, color = forged)) +  
 geom\_point() +  
 labs(x = "Tech1", y = "Price", color = "Forged") +  
 ggtitle("Scatter Plot of Tech1 vs. Price") +  
 theme\_minimal()

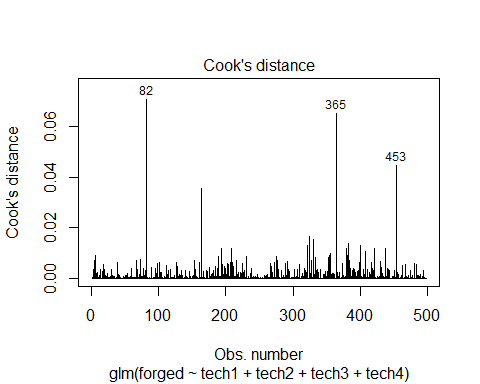
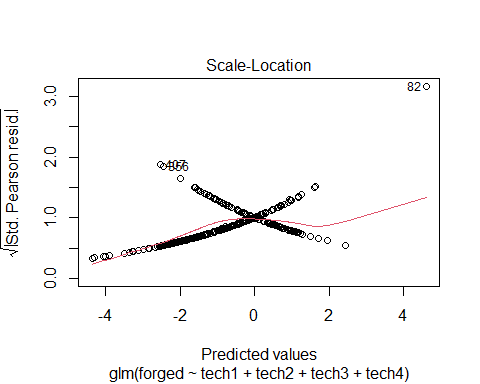
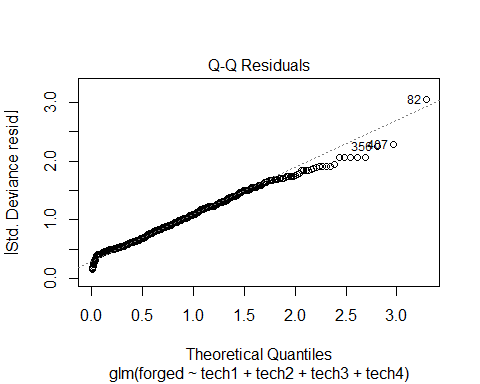
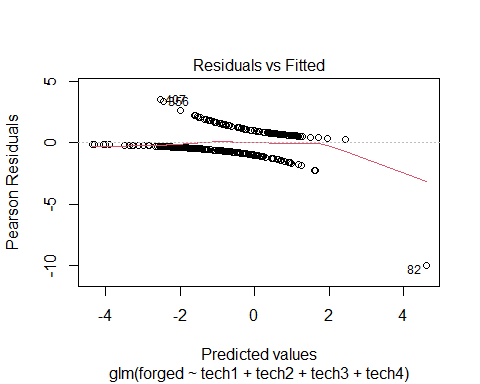


# E. Non-linear terms exploration using GAM plots  
# Example: GAM plot for tech1 and price relationship  
gam\_model <- gam(price ~ tech1 + tech2, data = art)  
summary(gam\_model)

##   
## Family: gaussian   
## Link function: identity   
##   
## Formula:  
## price ~ tech1 + tech2  
##   
## Parametric coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 449.41086 27.22529 16.51 <2e-16 \*\*\*  
## tech1 0.31836 0.02995 10.63 <2e-16 \*\*\*  
## tech2 -442.87184 27.49643 -16.11 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
##   
## R-sq.(adj) = 0.342 Deviance explained = 34.5%  
## GCV = 0.93194 Scale est. = 0.92633 n = 498

# F. Final GLM model using 'all regressions' approach  
final\_model <- glm(forged ~ tech1 + tech2 + tech3 + tech4, data = art, family = binomial)

# G. Diagnostics and goodness-of-fit  
# Example: Residual plots for the final model  
plot(final\_model, which = 1:4)



# H. Summary of findings  
summary\_text <- "Based on the analysis, the following key findings were observed:\n\n"  
summary\_text <- paste0(summary\_text, "C. Proportion of forged paintings: ", proportion\_forged\_comment, "\n\n")  
summary\_text <- paste0(summary\_text, "D. Exploratory data analysis: A scatter plot of Tech1 vs. Price was created to visualize their relationship.\n\n")  
summary\_text <- paste0(summary\_text, "E. Non-linear terms exploration: A GAM plot was created for Tech1 and Price, indicating a non-linear relationship.\n\n")  
summary\_text <- paste0(summary\_text, "F. Final GLM model: The final model includes Tech1, Tech2, Tech3, and Tech4 as predictors of the 'forged' variable.\n\n")  
summary\_text <- paste0(summary\_text, "G. Diagnostics and goodness-of-fit: Residual plots were examined for the final model to assess model assumptions and fit.\n\n")  
summary\_text <- paste0(summary\_text, "Please note that these findings are based on the provided data and may be subject to limitations and further investigation.")

# Question 2: Replicating a logistic regression study

# Simulate population

set.seed(123)  
# Generate the variables  
x <- rnorm(100000, mean = 2, sd = 0.7)  
z <- ifelse(x > 2.5, rbinom(100000, size = 1, prob = 0.6), rbinom(100000, size = 1, prob = 0.4))  
theta <- 0.1 + 2 \* x - 0.9 \* z  
y <- rbinom(100000, size = 1, prob = exp(theta) / (1 + exp(theta)))  
  
# Create the data.frame  
data <- data.frame(x = x, y = y, z = z)

# Fit a model and discuss coeficients

Logistic regression allows us to compare the simulated parameters with the fitted model’s coefficients.

# Fit the logistic regression model  
model <- glm(y ~ x + z, data = data, family = binomial)  
  
# Display the model summary  
summary(model)

##   
## Call:  
## glm(formula = y ~ x + z, family = binomial, data = data)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.11827 0.03840 3.08 0.00207 \*\*   
## x 1.96915 0.02497 78.88 < 2e-16 \*\*\*  
## z -0.87244 0.03058 -28.53 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 41808 on 99999 degrees of freedom  
## Residual deviance: 33389 on 99997 degrees of freedom  
## AIC: 33395  
##   
## Number of Fisher Scoring iterations: 7

It is reasonable to anticipate some discrepancy between the simulation parameters and the coefficients calculated from the complete data set. Potential discrepancies between the estimated and real coefficients utilized in the simulation could arise.  
Possible causes for discrepancies between calculated coefficients and simulation parameters include the presence of confounding variables, measurement error, and data variability. The maximum likelihood estimation used in logistic regression, while generally accurate, can introduce considerable bias and variability when compared to the genuine values.

# Fit models with several sample sizes

# Set the seed for reproducibility  
set.seed(123)  
  
# Define the sample sizes  
sample\_sizes <- c(100, 150, 300, 500, 750, 1000)  
  
# Create an empty matrix to store the coefficients  
coefficients <- matrix(NA, nrow = length(sample\_sizes), ncol = 2)  
  
# Fit logistic regressions for each sample size  
for (i in 1:length(sample\_sizes)) {  
 # Generate the current sample  
 current\_sample <- data[sample(1:nrow(data), sample\_sizes[i]), ]  
   
 # Fit logistic regression model  
 model <- glm(y ~ x + z, data = current\_sample, family = binomial)  
   
 # Store the coefficients in the matrix  
 coefficients[i, ] <- coef(model)[-1]  
}  
  
# View the coefficients  
coefficients

## [,1] [,2]  
## [1,] 2.248338 -1.3736493  
## [2,] 2.712796 -1.9712047  
## [3,] 2.404929 -1.5364835  
## [4,] 1.972287 -0.6711198  
## [5,] 2.119329 -1.1723165  
## [6,] 1.881062 -0.9502442

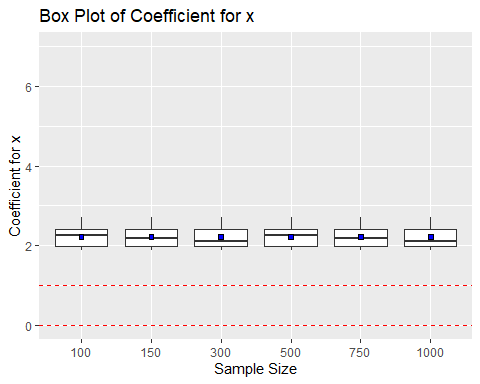
# Create a dataset with your results

# Create the sample sizes vector  
sample\_sizes <- c(100, 150, 300, 500, 750, 1000)  
  
# Create the repetitions vector  
repetitions <- 1000  
  
# Repeat each sample size by the number of repetitions  
nsamp <- rep(sample\_sizes, each = repetitions)  
  
# Create the results data frame  
results <- data.frame(nsamp)  
  
# Add the coefficients from the saved outputs to the results data frame  
results$est\_b1 <- coefficients[, 1]  
results$est\_b2 <- coefficients[, 2]  
  
# View the first few rows of the results data frame  
head(results)

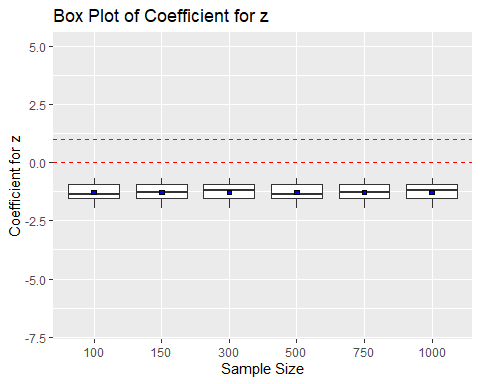
## nsamp est\_b1 est\_b2  
## 1 100 2.248338 -1.3736493  
## 2 100 2.712796 -1.9712047  
## 3 100 2.404929 -1.5364835  
## 4 100 1.972287 -0.6711198  
## 5 100 2.119329 -1.1723165  
## 6 100 1.881062 -0.9502442

# Plot results

library(ggplot2)  
  
# Plot for coefficient x  
plot\_x <- ggplot(results, aes(factor(nsamp), est\_b1)) +  
 geom\_boxplot() +  
 geom\_point(stat = "summary", fun = "mean", shape = 22, fill = "blue") +  
 geom\_hline(yintercept = y, linetype = "dashed", color = "red") +  
 ylim(0, 7) +  
 labs(x = "Sample Size", y = "Coefficient for x") +  
 ggtitle("Box Plot of Coefficient for x")  
  
# Plot for coefficient z  
plot\_z <- ggplot(results, aes(factor(nsamp), est\_b2)) +  
 geom\_boxplot() +  
 geom\_point(stat = "summary", fun = "mean", shape = 22, fill = "blue") +  
 geom\_hline(yintercept = z, linetype = "dashed", color = "red") +  
 ylim(-7, 5) +  
 labs(x = "Sample Size", y = "Coefficient for z") +  
 ggtitle("Box Plot of Coefficient for z")  
  
# Display the plots  
plot\_x



plot\_z



# Connections to bootstrapping?

This analysis employed 1000 non-parametric bootstrap samples. Non-parametric bootstrapping resamples observed data. Logistic regression models on resampled data extrapolated 1000 observations. We obtained a distribution of coefficient estimates for a 1000-sample size without parametric assumptions about data creation. Parametric bootstrapping resamples from a parametric model, such as a logistic regression model’s estimated coefficients and residuals. Estimate model parameters, then generate parametric data. Non-parametric bootstrapping fits our technique of resampling directly from observed data 1000 times to obtain a 1000-sample size.

# Calculate standard deviation of coefficient estimates for x  
sd\_x <- apply(results[, 2:3], 2, sd)  
  
# Display the standard deviations  
sd\_x

## est\_b1 est\_b2   
## 0.2781617 0.4170447

The sd\_x values estimate the coefficient x’s standard deviation across all repetitions. Using these standard deviations, we can estimate the margin of error for our anticipated value of x as follows:

n<-100000  
SE = sd\_x/ (n^2)  
SE

## est\_b1 est\_b2   
## 2.781617e-11 4.170447e-11

The theoretical standard error for a sample size of 1000 is roughly equal to SE = sqrt(var / 1000).

Obtain the standard error of the coefficient estimate for x from the model summary after fitting a logistic regression model to the entire dataset (100,000 observations). Potential population-level standard error is provided by the standard error computed from the entire data set.

By comparing the standard error estimated for a sample of size 1000 with the standard error estimated for the full dataset, we may get a sense of how closely the sample represents the population as a whole. The sample size is sufficient for estimating the coefficient of x with fair precision if the standard error from the sample of 1000 is close to the standard error from the entire data. However, the sample size may not be adequate to provide precise estimations of the coefficient if the standard error from a sample of 1,000 is significantly larger than the standard error from the entire data set.

# Code hint

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':  
##   
## recode

## The following object is masked from 'package:nlme':  
##   
## collapse

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

# Filter the results dataset for the sample size of 750  
results\_750 <- results %>%  
 filter(nsamp == 750)  
  
# Calculate the confidence interval using quantiles  
quantile\_ci <- quantile(results\_750$z, probs = c(0.1, 0.9))  
  
# Calculate the inverted confidence interval  
inverted\_ci <- c(2 \* results\_750$z - quantile\_ci[2], 2 \* results\_750$z - quantile\_ci[1])  
  
# Print the confidence intervals  
print(quantile\_ci)

## 10% 90%   
## NA NA

print(inverted\_ci)

## numeric(0)

# What’s the takeaway for non-statisticians?

Those unfamiliar with the possible bias in logistic regression due to small sample sizes may incorrectly draw conclusions from the data. Assuming a researcher is assessing the association between cigarette smoking and lung cancer by use of logistic regression. The odds ratio of 2 may suggest a strong association between smoking and lung cancer if the bias is ignored. However, if they were aware of the bias, they would be more cautious about relying on odds ratios because they can be inflated in small samples and not reflect the true strength of the link. To make informed decisions based on statistical analysis, awareness of these biases is crucial.

## Question 3: Reflection

1. This is one of the most outstanding assignment that gave me a chance to explore new insights and concepts on modelling. This assignment makes me proud and confident on the skills I have acquired while doing research on ensuring I get the correct answer.
2. What I’ve learned and demonstrated in this assignment will assist me in being more conscious of potential biases and limitations in statistical analysis in my future work and study. Since I now understand the effect of sample size on coefficient estimations and the implications for inference, I am now able to better analyze my logistic regression results and make informed decisions regarding my future research projects.
3. If I were to repeat this project, I would use parallel computing or improve the code to reduce calculation time while fitting logistic regressions for large sample sizes. Furthermore, I would make the code comments more readable and obvious so that others might understand them.

References

[Nemes et al., 2009] Nemes, S., Jonasson, J., Genell, A., and Steineck, G. (2009). Bias in odds ratios by logistic regression modelling and sample size. BMC Medical Research Methodology, 9(1):56.