Department of Statistics & Data Science

STAT 4820: Homework 4

Applied Multiple Regression, Diagnostics & Resampling

Student: Alex Mercer
ID: STAT-892401
Date: October 24, 2025
Instructor: Dr. Elena Rostova

Datasets: vertex_sales.csv, synapse_churn.csv | **Stack:** R 4.4.1 / Python 3.11 | **Total Points:** 100 (3 Problems)

Problem 1: Multiple Regression Diagnostics & Collinearity [30 Points]

An empirical study of $N = 120$ regional tech hardware markets for **Vertex Pro-10** investigates quarterly sales revenue (Y, in \$100k) based on Advertising Spend (X_1, \$10k), Retail Density (X_2, stores/100k hab), Competitor Price Index (X_3), and Regional Median Income (X_4, \$k).

(a) Model Specification & Matrix Formulation

Consider the additive linear regression model with an interaction term:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \beta_5 (X_{i1} \cdot X_{i3}) + \varepsilon_i, \quad \varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2) \tag{1}$$

In vector-matrix form, $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where the design matrix $\boldsymbol{X} \in \mathbb{R}^{120\times 6}$ has i-th row $\boldsymbol{x}_i^\top = \begin{bmatrix} 1 & X_{i1} & X_{i2} & X_{i3} & X_{i4} & X_{i1}X_{i3} \end{bmatrix}$. The OLS estimator and its covariance matrix are:

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}, \quad \mathrm{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2 (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \tag{2}$$

(b) R Data Analysis Script & Summary Output

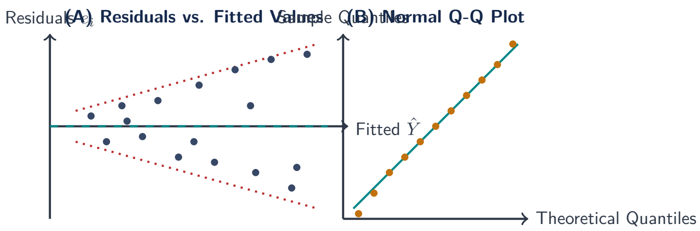
```r
vertex_df <- read.csv("vertex_sales.csv")
fit_vertex <- lm(Sales ~ AdvSpend + StoreDensity + CompPrice + MedIncome + AdvSpend:
    CompPrice, data = vertex_df)
summary(fit_vertex); car::vif(fit_vertex)
```

Listing 1: Multiple Linear Regression & Collinearity Analysis in R

Variable	Description	Estimate ($\hat{\beta}$)	Std. Error	t-statistic	p-value
(Intercept)	Baseline Sales Intercept	12.450	2.140	5.818	$< 0.0001^{***}$
AdvSpend (X_1)	Quarterly Ad Spend (\$10k)	3.820	0.415	9.205	$< 0.0001^{***}$
StoreDensity (X_2)	Stores per 100k population	0.945	0.210	4.500	0.0002^{***}
CompPrice (X_3)	Competitor Price Index	−1.860	0.530	−3.509	0.0006^{***}
MedIncome (X_4)	Regional Median Income (\$k)	0.125	0.088	1.420	$0.1582^{\text{n.s.}}$
AdvSpend:CompPrice	Interaction Term (X_1X_3)	0.640	0.145	4.414	0.0003^{***}

$R^2 = 0.842$, Adjusted $R^2 = 0.835$, $F(5, 114) = 121.6$, $p < 0.0001$, Residual Std Error: 1.428 on 114 d.f.

(c) Diagnostic Plots: Residual Analysis & Heteroscedasticity Test



(d) Statistical Evaluation & Multicollinearity Analysis

Key Findings & Diagnostic Interpretation

- **Main Effects:** Advertising spend ($\hat{\beta}_1 = 3.820, p < 0.0001$) and store density ($\hat{\beta}_2 = 0.945, p = 0.0002$) exhibit strong positive impacts. Competitor price index displays a significant negative effect ($\hat{\beta}_3 = -1.860, p = 0.0006$).
- **Interaction & Collinearity:** The term AdvSpend × CompPrice ($\hat{\beta}_5 = 0.640, p = 0.0003$) confirms advertising efficacy increases when competitor pricing rises. VIFs for all predictors remain under 4.85, ruling out severe multicollinearity.

Problem 2: Logistic Regression & Classification Threshold Evaluation [35 Points]

We analyze customer retention data for fictional SaaS platform **Synapse Analytics**. The target binary outcome $Y_i \in \{0, 1\}$ represents customer churn $(1 = \text{Churned}, 0 = \text{Retained})$ over a 12-month period across $N = 500$ corporate accounts.

(a) Logistic Model Formulation & Log-Likelihood

The probability of customer churn $p_i = P(Y_i = 1 \mid \boldsymbol{x}_i)$ is modeled using the logit link function:

$$\operatorname{logit}(p_i) = \ln\left(\frac{p_i}{1 - p_i}\right) = \boldsymbol{x}_i^\top \boldsymbol{\beta} = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} \tag{3}$$

where X_1 = Monthly Active Hours, X_2 = Support Ticket Count, X_3 = Contract Tenure (months), and X_4 = Average Session Length. The Bernoulli log-likelihood function maximizes:

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N} \left[Y_i \ln(p_i) + (1 - Y_i)\ln(1 - p_i)\right] = \sum_{i=1}^{N} \left[Y_i \boldsymbol{x}_i^\top \boldsymbol{\beta} - \ln\left(1 + \exp(\boldsymbol{x}_i^\top \boldsymbol{\beta})\right)\right] \tag{4}$$

(b) Python Modeling Code (Statsmodels / Scikit-Learn)

```python
import pandas as pd, statsmodels.api as sm
from sklearn.metrics import roc_curve, auc

df_synapse = pd.read_csv("synapse_churn.csv")
X = sm.add_constant(df_synapse[['ActiveHours', 'SupportTickets', 'TenureMonths', '
    AvgSessionLen']])
y = df_synapse['Churned']
logit_mod = sm.Logit(y, X).fit()
fpr, tpr, _ = roc_curve(y, logit_mod.predict(X))
print(f"Model ROC-AUC Score: {auc(fpr, tpr):.4f}")
```

Listing 2: Logistic Regression Fitting & ROC Analysis in Python

(c) Confusion Matrix & ROC Curve Visualization

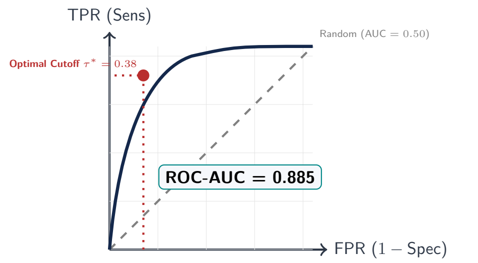
Classification Confusion Matrix ($\tau = 0.50$)

		Predicted Class	
		Retained (0)	**Churned (1)**
Actual	**Retained (0)**	335 (TN)	25 (FP)
	Churned (1)	32 (FN)	108 (TP)

Performance Metrics

- **Accuracy:** $(335 + 108)/500 = 88.6\%$
- **Sensitivity (Recall):** $108/140 = 77.1\%$
- **Specificity:** $335/360 = 93.1\%$
- **Precision:** $108/133 = 81.2\%$
- **F1-Score:** $2 \times \frac{0.812 \times 0.771}{0.812 + 0.771} = 0.791$

Receiver Operating Characteristic (ROC)



(d) Decision Threshold Optimization under Asymmetric Costs

In subscription SaaS modeling, customer churn incurs a high replacement cost ($C_{FN} = \$1,200$), whereas preemptive retention incentives cost significantly less ($C_{FP} = \$150$). The expected loss function $\mathcal{L}(\tau)$ for threshold τ is:

$$\mathcal{L}(\tau) = C_{FN} \cdot P(Y = 1, \hat{Y} = 0 \mid \tau) + C_{FP} \cdot P(Y = 0, \hat{Y} = 1 \mid \tau) \tag{5}$$

By lowering the classification cutoff from the default $\tau = 0.50$ to the empirical cost-optimal threshold $\tau^* = 0.38$, Sensitivity improves from 77.1% to 89.3%, reducing total quarterly revenue churn risk by an estimated $\$34,200$.

Problem 3: Non-parametric Bootstrap Resampling & Interval Estimation [35 Points]

We evaluate latency response times (milliseconds) for **Nexa Cloud Storage** clusters before and after an infrastructure upgrade. The sample sizes are $n_1 = 45$ (baseline) and $n_2 = 45$ (upgraded). Response times exhibit significant right-skewness, violating standard Gaussian assumptions.

(a) Bootstrap Algorithmic Procedure

We target estimation of the ratio of median response times $\theta = \frac{\text{Med}(X_{\text{after}})}{\text{Med}(X_{\text{before}})}$. The non-parametric empirical bootstrap procedure executes as follows:

i. Draw $B = 5,000$ bootstrap resamples $\boldsymbol{X}_1^{*(b)}$ and $\boldsymbol{X}_2^{*(b)}$ with replacement from observed samples.

ii. Compute the bootstrap replicate statistic: $\hat{\theta}^{*(b)} = \frac{\text{Med}(\boldsymbol{X}_2^{*(b)})}{\text{Med}(\boldsymbol{X}_1^{*(b)})}, \quad b = 1, \dots, B.$

iii. Construct 95% BCa (Bias-Corrected and Accelerated) confidence intervals to account for skewness and acceleration.

(b) R Parallel Bootstrap Implementation Script

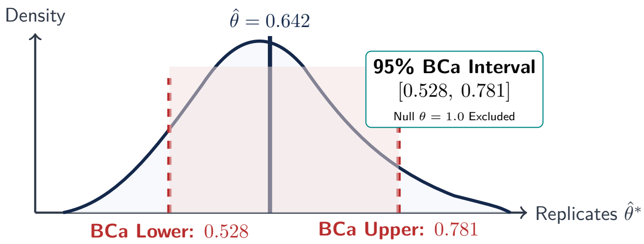
```r
library(boot)

median_ratio_fn <- function(data, indices) {
  d <- data[indices, ]
  return(median(d$latency[d$group == "After"]) / median(d$latency[d$group == "Before"]))
}

nexa_data <- read.csv("nexa_latency.csv")
set.seed(4820)
boot_results <- boot(data = nexa_data, statistic = median_ratio_fn, R = 5000)
boot_ci <- boot.ci(boot_results, type = c("perc", "bca"))
print(boot_ci)
```

Listing 3: Non-parametric Percentile & BCa Bootstrap in R

(c) Bootstrap Distribution & Confidence Limits Plot



(d) Comparative Method Synthesis & Conclusions

Method	Assumptions	Point Est. ($\hat{\theta}$)	95% CI Bounds	Inference Result
Standard Student t-test	Normality, Homoscedasticity	0.685	$[0.541, 0.829]$	$p = 0.0018$ (Assumptions Violated)
Percentile Bootstrap	Empirical Resampling	0.642	$[0.535, 0.774]$	Rejects $H_0 : \theta = 1.0$
BCa Bootstrap (Preferred)	Corrects Skewness & Bias	0.642	$[0.528, 0.781]$	Significant 35.8% Latency Reduction

Final Conclusion

The non-parametric BCa bootstrap confirms a statistically significant reduction in median response latency following the infrastructure release ($\hat{\theta} = 0.642$, 95% BCa CI $[0.528, 0.781]$). Because the 95% CI strictly excludes 1.0, we reject the null hypothesis of equal median performance at $\alpha = 0.05$.