kx2224_hw4

Kangyu Xu (kx2224)

2024-11-18

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(readxl)
library(knitr)
library(ggplot2)
library(dplyr)
```

Problem 1

(a)

```r
# Data
data = c(125, 123, 117, 123, 115, 112, 128, 118, 124, 111, 116, 109, 125, 120,
          113, 123, 112, 118, 121, 118, 122, 115, 105, 118, 131)

# Hypothetical median
median_hypothesis = 120

# Compare data with the hypothetical median
signs = data - median_hypothesis

# Count positive and negative signs
positive_count = sum(signs > 0) # Count of values greater than 120
negative_count = sum(signs < 0) # Count of values less than 120

# Test statistic for the sign test
test_statistic = min(positive_count, negative_count)
```

```r
# Total number of non-zero signs
n = positive_count + negative_count

# Calculate the p-value for a one-tailed test (median < 120)
p_value = pbinom(test_statistic, size = n, prob = 0.5, lower.tail = TRUE)

# Output results
cat("Sign Test Results:\n")
```

```
## Sign Test Results:
```

```r
cat("Test Statistic (minimum count of signs):", test_statistic, "\n")
```

```
## Test Statistic (minimum count of signs): 10
```

```r
cat("p-value:", p_value, "\n")
```

```
## p-value: 0.2706281
```

As p-value is 0.2706281 > 0.05. Therefore, we fail to reject the null hypothesis. There is no significant evidence that the median is less than 120.

(b)

```r
wilcox_test_result = wilcox.test(data, mu = 120, alternative = "less")
```

```
## Warning in wilcox.test.default(data, mu = 120, alternative = "less"): cannot
## compute exact p-value with ties
```

```
## Warning in wilcox.test.default(data, mu = 120, alternative = "less"): cannot
## compute exact p-value with zeroes
```

```r
print("Wilcoxon Signed-Rank Test Result:")
```

```
## [1] "Wilcoxon Signed-Rank Test Result:"
```

```r
print(wilcox_test_result)
```

```
## 
##  Wilcoxon signed rank test with continuity correction
## 
## data:  data
## V = 112.5, p-value = 0.1447
## alternative hypothesis: true location is less than 120
```

Similarly, the p-value is greater than 0.05, indicating no significant evidence to conclude that the median blood sugar reading is less than 120 at the 0.05 level.

Problem 2

(a)

```
file_path = "Brain.xlsx"
brain_data = read_excel(file_path)|>
  janitor::clean_names()
nonhuman_data = brain_data[brain_data$species != "Homo sapiens", ]|>
  janitor::clean_names()
head(nonhuman_data)
```

```
## # A tibble: 6 x 4
##   species            brain_mass_g       ln_brain_mass glia_neuron_ratio
##   <chr>              <chr>                      <dbl>             <dbl>
## 1 Pan troglodytes    336.2                       5.82              1.2 
## 2 Gorilla gorilla    509.2                       6.23              1.21
## 3 Pongo pygmaeus     342.7                       5.84              0.98
## 4 Hylobates muelleri 101.8                       4.62              1.22
## 5 Papio anubis       155.80000000000001          5.05              0.97
## 6 Mandrillus sphinx  159.19999999999999          5.07              1.02
```

```
model = lm(`glia_neuron_ratio` ~ `ln_brain_mass`, data = nonhuman_data)
summary(model)
```

```
## 
## Call:
## lm(formula = glia_neuron_ratio ~ ln_brain_mass, data = nonhuman_data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.24150 -0.12030 -0.01787  0.15940  0.25563 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    0.16370    0.15987   1.024 0.322093    
## ln_brain_mass  0.18113    0.03604   5.026 0.000151 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.1699 on 15 degrees of freedom
## Multiple R-squared:  0.6274, Adjusted R-squared:  0.6025 
## F-statistic: 25.26 on 1 and 15 DF,  p-value: 0.0001507
```

Finish fitting a model.

(b)

```
# Extract human ln(brain mass)
human_ln_brain_mass = brain_data[brain_data$species == "Homo sapiens", ]$ln_brain_mass
human_ln_brain_mass
```

```
## [1] 7.22
```

```
# Predict the glia-neuron ratio for humans
predicted_human_ratio = predict(model, newdata = data.frame(ln_brain_mass = human_ln_brain_mass))

# Print the predicted ratio
predicted_human_ratio
```

```
##        1
## 1.471458
```

(c)

The most reasonable range for the prediction is an interval for the predicted mean glia-neuron ratio, as our focus is on estimating the population average rather than individual cases.

(d)

```
# 95% confidence interval for the predicted mean glia-neuron ratio
mean_interval = predict(model, newdata = data.frame(ln_brain_mass = human_ln_brain_mass), interval = "c

# Print intervals
mean_interval
```

```
##        fit      lwr      upr
## 1 1.471458 1.229558 1.713358
```

The 95% confidence interval is [1.230, 1.713]. The observed glia-neuron ratio for humans is 1.65, which falls within this interval. Therefore, we cannot reject the null hypothesis, indicating that the human brain does not have an excessive glia-neuron ratio for its size compared to other primates. This suggests that the ratio is consistent with what would be expected based on brain mass, similar to other primates.

(e)

The human data point lies far beyond the range of brain masses observed in non-human primates, introducing greater uncertainty into the linear regression model. This raises the possibility that the relationship between brain mass and the glia-neuron ratio may not remain linear across all primates, including humans. As a result, predictions for humans may be less reliable, and caution is needed when interpreting these results. To improve prediction accuracy and reduce uncertainty, it would be beneficial to include more data points from primates with brain masses closer to that of humans.

Problem 3

(a)

```
heart_disease_data = read_csv("HeartDisease.csv")
```

```
## Rows: 788 Columns: 10
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (10): id, totalcost, age, gender, interventions, drugs, ERvisits, compli...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
summary(heart_disease_data)
```

```
##        id          totalcost            age            gender
##  Min.   :  1.0   Min.   :    0.0   Min.   :24.00   Min.   :0.0000
##  1st Qu.:197.8   1st Qu.:  161.1   1st Qu.:55.00   1st Qu.:0.0000
##  Median :394.5   Median :  507.2   Median :60.00   Median :0.0000
##  Mean   :394.5   Mean   : 2800.0   Mean   :58.72   Mean   :0.2284
##  3rd Qu.:591.2   3rd Qu.: 1905.5   3rd Qu.:64.00   3rd Qu.:0.0000
##  Max.   :788.0   Max.   :52664.9   Max.   :70.00   Max.   :1.0000
##  interventions        drugs           ERvisits      complications
##  Min.   : 0.000   Min.   :0.0000   Min.   : 0.000   Min.   :0.00000
##  1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.:0.00000
##  Median : 3.000   Median :0.0000   Median : 3.000   Median :0.00000
##  Mean   : 4.707   Mean   :0.4467   Mean   : 3.425   Mean   :0.05711
##  3rd Qu.: 6.000   3rd Qu.:0.0000   3rd Qu.: 5.000   3rd Qu.:0.00000
##  Max.   :47.000   Max.   :9.0000   Max.   :20.000   Max.   :3.00000
##  comorbidities       duration
##  Min.   : 0.000   Min.   :  0.00
##  1st Qu.: 0.000   1st Qu.: 41.75
##  Median : 1.000   Median :165.50
##  Mean   : 3.767   Mean   :164.03
##  3rd Qu.: 5.000   3rd Qu.:281.00
##  Max.   :60.000   Max.   :372.00
```

```
# Separate continuous and categorical variables
continuous_vars = heart_disease_data |> select(totalcost, ERvisits, age, duration)
categorical_vars = heart_disease_data |> select(gender, complications)
# Descriptive statistics for continuous variables
summary(continuous_vars)
```

```
##    totalcost          ERvisits           age            duration
##  Min.   :    0.0   Min.   : 0.000   Min.   :24.00   Min.   :  0.00
##  1st Qu.:  161.1   1st Qu.: 2.000   1st Qu.:55.00   1st Qu.: 41.75
##  Median :  507.2   Median : 3.000   Median :60.00   Median :165.50
##  Mean   : 2800.0   Mean   : 3.425   Mean   :58.72   Mean   :164.03
##  3rd Qu.: 1905.5   3rd Qu.: 5.000   3rd Qu.:64.00   3rd Qu.:281.00
##  Max.   :52664.9   Max.   :20.000   Max.   :70.00   Max.   :372.00
```

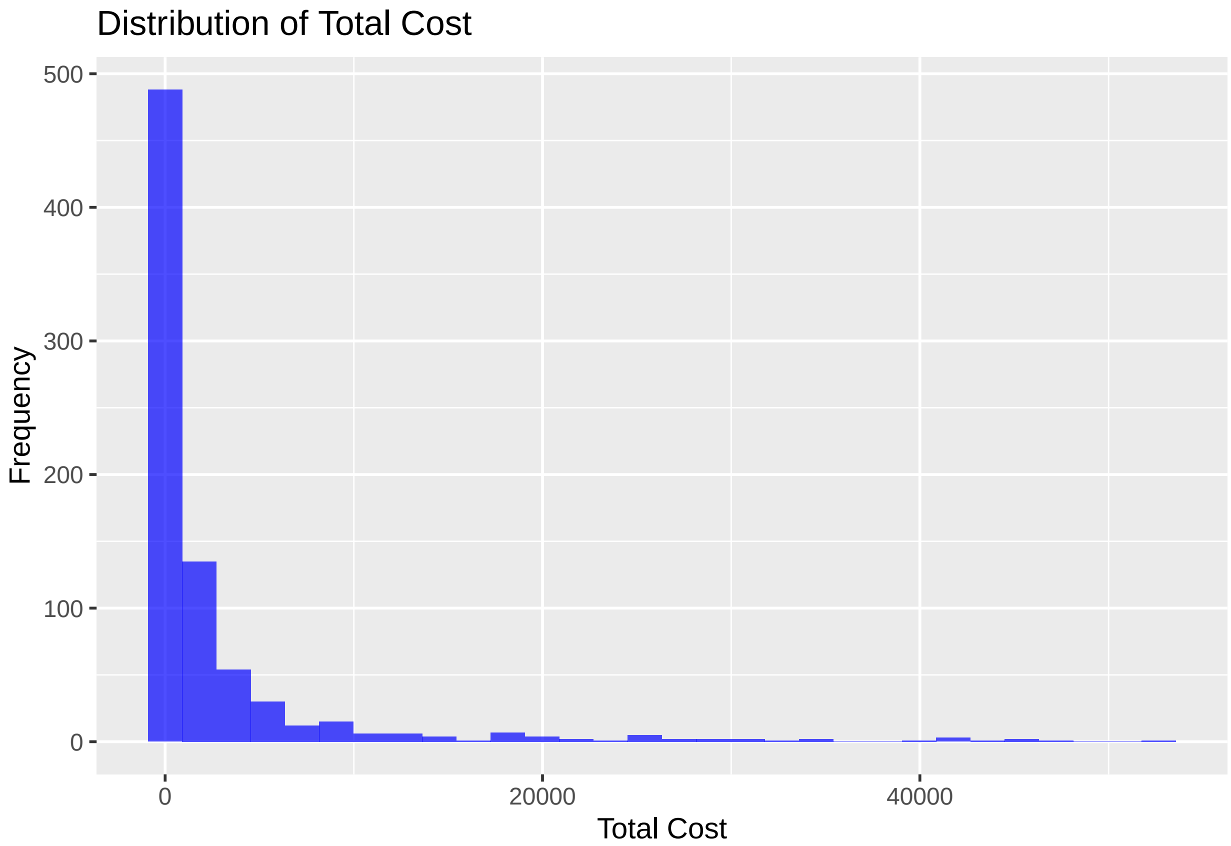
```
# Frequency table for categorical variables
lapply(categorical_vars, table)
```

```
## $gender
##
##   0   1
## 608 180
##
## $complications
##
##   0   1   3
## 745  42   1
```

The main outcome is the total cost (totalcost), while the key predictor is the number of emergency room visits (ERvisits). Other significant covariates include the subscriber's age (age), gender (gender), total number of interventions or procedures performed (interventions), number of prescribed drugs (drugs), number of complications during heart disease treatment (complications), number of comorbidities experienced during the period (comorbidities), and the duration of the treatment condition in days (duration).

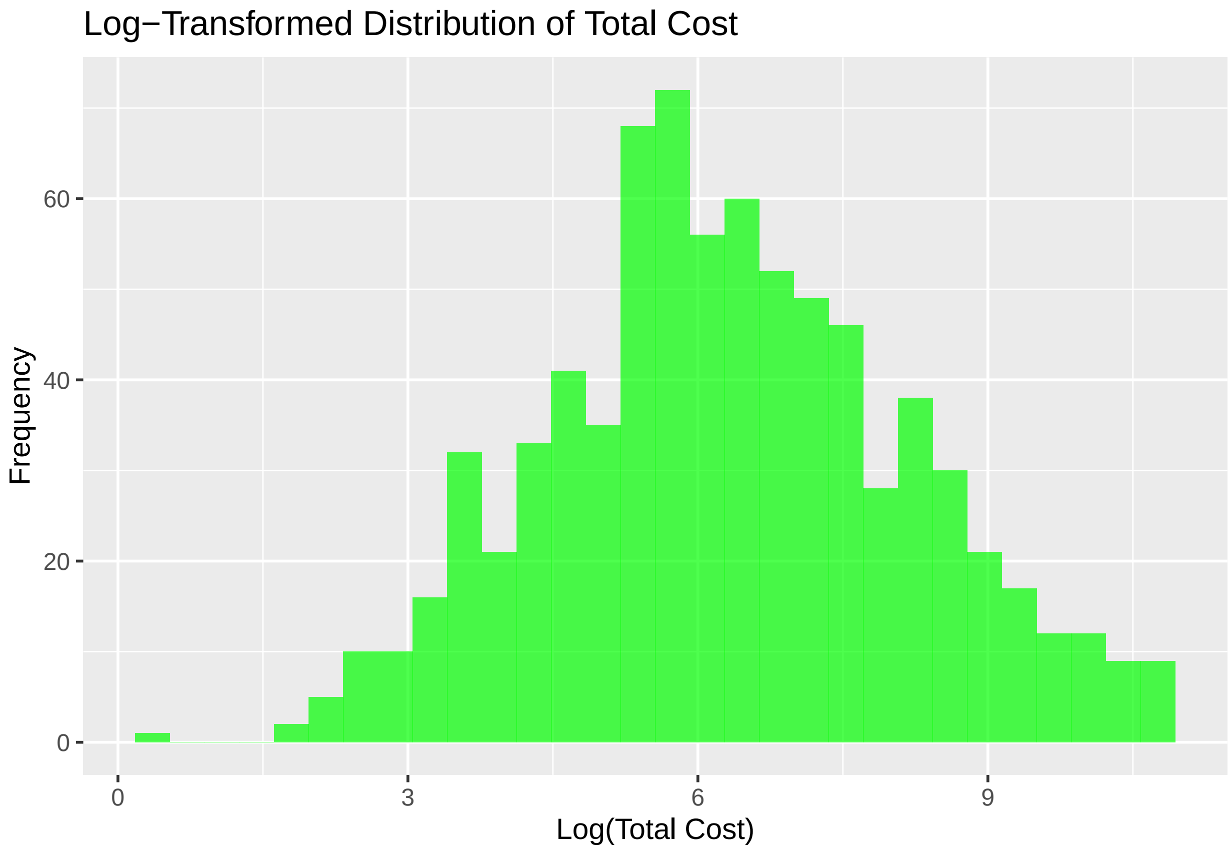
(b)

```r
# Plot histogram of 'totalcost'
ggplot(heart_disease_data, aes(x = totalcost)) +
  geom_histogram(bins = 30, fill = "blue", alpha = 0.7) +
  labs(title = "Distribution of Total Cost", x = "Total Cost", y = "Frequency")
```



```r
# Check if log transformation improves the distribution
ggplot(heart_disease_data, aes(x = log(totalcost))) +
  geom_histogram(bins = 30, fill = "green", alpha = 0.7) +
  labs(title = "Log-Transformed Distribution of Total Cost", x = "Log(Total Cost)", y = "Frequency")
```

```
## Warning: Removed 3 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



(c)

```
# Create the new variable
heart_disease_data = heart_disease_data |> mutate(comp_bin = ifelse(complications == 0, 0, 1))

# Check the frequency distribution of the new variable
table(heart_disease_data$comp_bin)
```

```
##
##   0   1
## 745  43
```

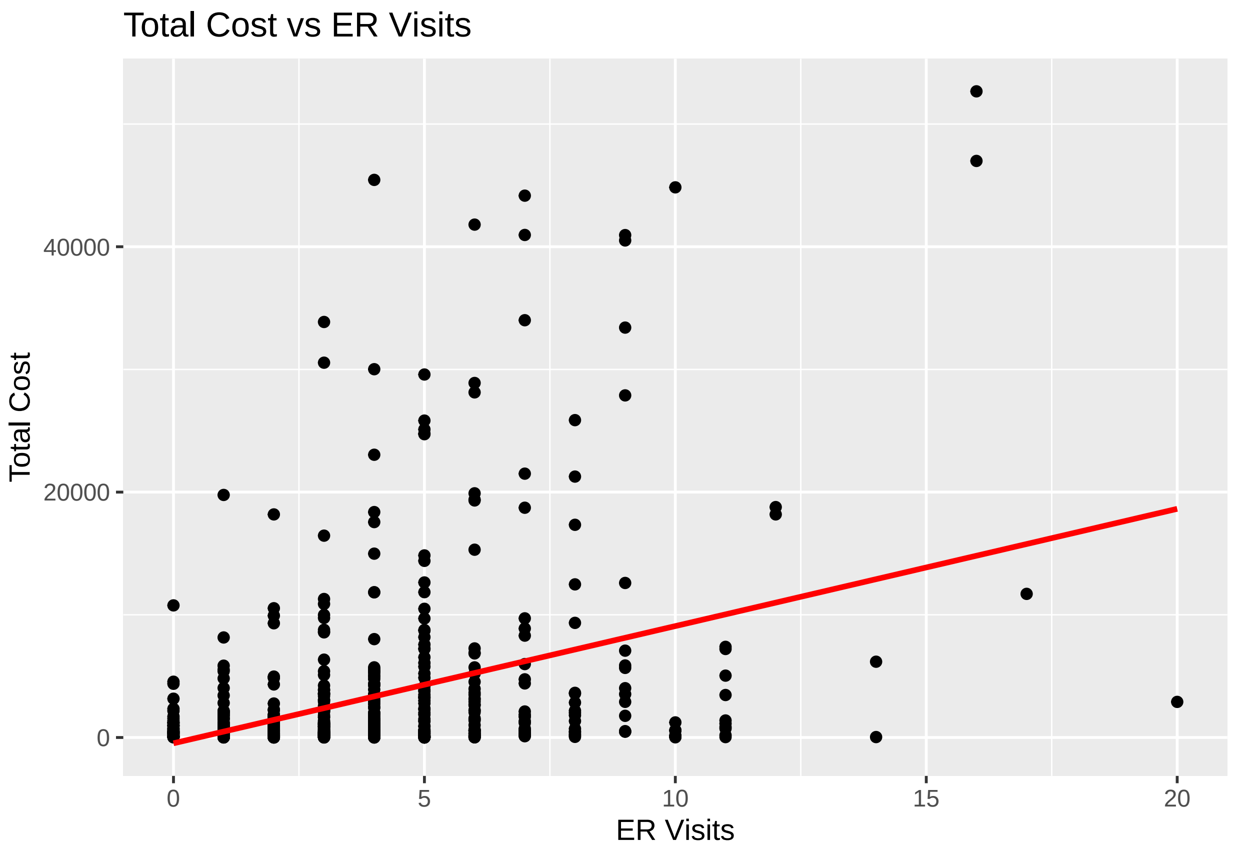
(d)

```
# Fit a simple linear regression model with original 'totalcost'
model_original = lm(totalcost ~ ERvisits, data = heart_disease_data)
summary(model_original)
```

```
##
## Call:
## lm(formula = totalcost ~ ERvisits, data = heart_disease_data)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -15733  -2353  -1062     185   42098
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -472.54     362.24  -1.304    0.192
## ERvisits      955.44      83.81  11.399   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6201 on 786 degrees of freedom
## Multiple R-squared:  0.1419, Adjusted R-squared:  0.1408
## F-statistic: 129.9 on 1 and 786 DF,  p-value: < 2.2e-16
```

```
# Scatterplot with regression line for original 'totalcost'
ggplot(heart_disease_data, aes(x = ERvisits, y = totalcost)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "red") +
  labs(title = "Total Cost vs ER Visits", x = "ER Visits", y = "Total Cost")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



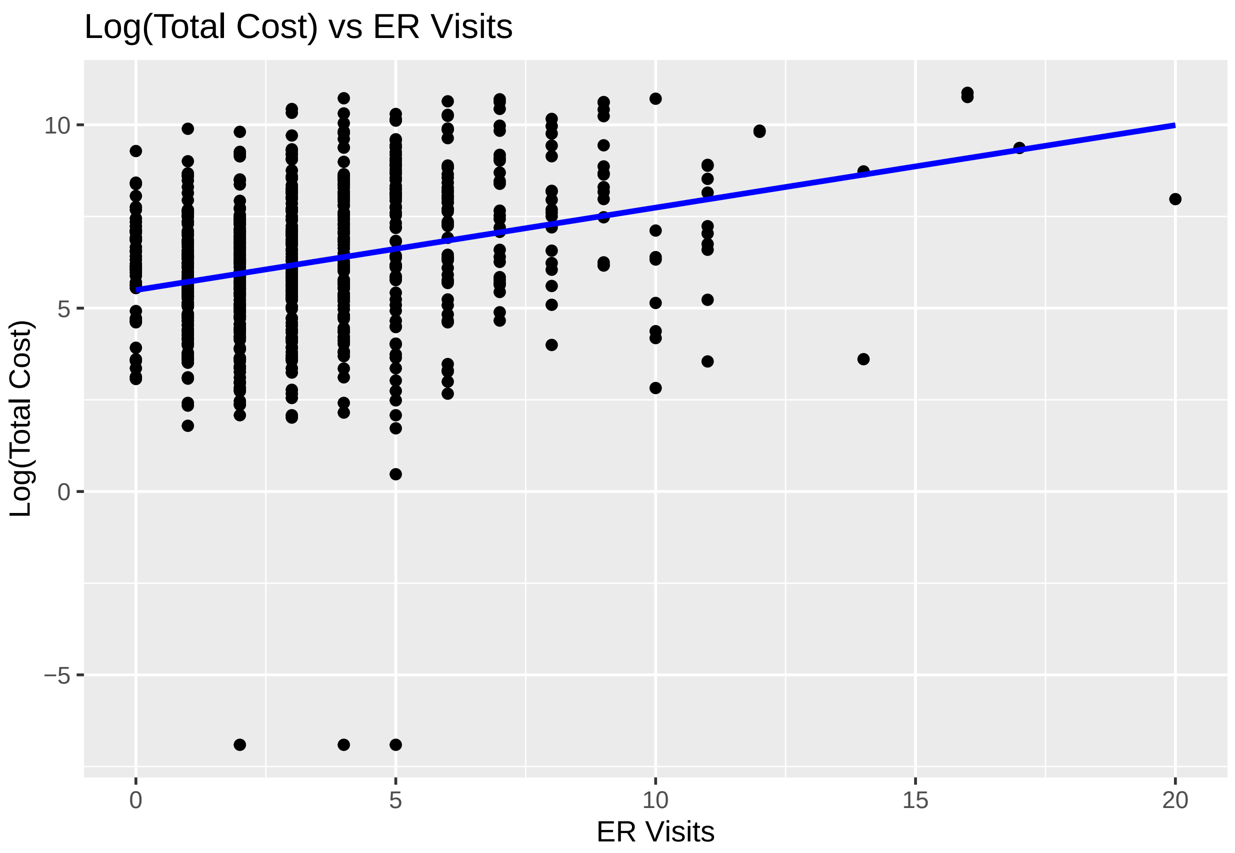
```r
heart_disease_data =heart_disease_data |>
  mutate(log_totalcost = log(totalcost + 0.001)) |>
  select(id, log_totalcost, everything(), -totalcost)

# Fit a simple linear regression model with log-transformed 'totalcost'
model_log = lm(log_totalcost ~ ERvisits, data = heart_disease_data)
summary(model_log)
```

```
##
## Call:
## lm(formula = log_totalcost ~ ERvisits, data = heart_disease_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5255  -1.0922   0.0608   1.3147   4.3314
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.49385    0.11387  48.248   <2e-16 ***
## ERvisits     0.22477    0.02635   8.531   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.949 on 786 degrees of freedom
## Multiple R-squared:  0.08475,    Adjusted R-squared:  0.08359
## F-statistic: 72.79 on 1 and 786 DF,  p-value: < 2.2e-16
```

```r
# Scatterplot with regression line for log-transformed 'totalcost'
ggplot(heart_disease_data, aes(x = ERvisits, y = log_totalcost)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Log(Total Cost) vs ER Visits", x = "ER Visits", y = "Log(Total Cost)")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



(e)

```
# Fit a multiple linear regression model using log-transformed 'totalcost'
mlr_log_model = lm(log_totalcost ~ comp_bin + ERvisits, data = heart_disease_data)
summary(mlr_log_model)
```

```
##
## Call:
## lm(formula = log_totalcost ~ comp_bin + ERvisits, data = heart_disease_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3943  -1.0451   0.0252   1.2191   4.4397
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.47694    0.11165  49.054  < 2e-16 ***
## comp_bin     1.74365    0.30321   5.751 1.27e-08 ***
## ERvisits     0.20193    0.02613   7.728 3.33e-14 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.911 on 785 degrees of freedom
## Multiple R-squared:  0.1218, Adjusted R-squared:  0.1195
## F-statistic: 54.41 on 2 and 785 DF,  p-value: < 2.2e-16
```

(e): i

```
# Add interaction term to test if 'comp_bin' is an effect modifier
interaction_log_model = lm(log_totalcost ~ comp_bin * ERvisits, data = heart_disease_data)
summary(interaction_log_model)
```

```
##
## Call:
## lm(formula = log_totalcost ~ comp_bin * ERvisits, data = heart_disease_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.4051  -1.0559   0.0325   1.2269   4.4353
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.45549    0.11406  47.828  < 2e-16 ***
## comp_bin           2.22319    0.60233   3.691 0.000239 ***
## ERvisits           0.20837    0.02705   7.703 4.01e-14 ***
## comp_bin:ERvisits -0.09639    0.10461  -0.921 0.357101
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 1.911 on 784 degrees of freedom
## Multiple R-squared:  0.1227, Adjusted R-squared:  0.1193
## F-statistic: 36.55 on 3 and 784 DF,  p-value: < 2.2e-16
```

```
# Compare models with and without the interaction term
anova(mlr_log_model, interaction_log_model)
```

```
## Analysis of Variance Table
##
## Model 1: log_totalcost ~ comp_bin + ERvisits
## Model 2: log_totalcost ~ comp_bin * ERvisits
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    785 2866.1
## 2    784 2863.0  1    3.1006 0.8491 0.3571
```

The p value for the interaction term is $0.357 > 0.05$, therefore, we fail to reject the null hypothesis, indicating that comp_bin is not an effect modifier.

```
# Fit a model without 'comp_bin'
model_no_comp_log = lm(log_totalcost ~ ERvisits, data = heart_disease_data)

# Compare the models with and without 'comp_bin'
anova(model_no_comp_log, mlr_log_model)
```

(e): ii

```
## Analysis of Variance Table
##
## Model 1: log_totalcost ~ ERvisits
## Model 2: log_totalcost ~ comp_bin + ERvisits
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1    786 2986.8
## 2    785 2866.1  1    120.74 33.07 1.273e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Check adjusted R-squared for both models
summary(model_no_comp_log)$adj.r.squared
```

```
## [1] 0.0835891
```

```r
summary(mlr_log_model)$adj.r.squared
```

```
## [1] 0.1195147
```

`comp_bin` acts as a confounder because its inclusion in the model significantly improves the explanation of variance in log_totalcost. Excluding it would omit critical information that affects the relationship between ERvisits and log_totalcost.

```r
# Evaluate significance of 'comp_bin' in the MLR model
summary(mlr_log_model)$coefficients
```

(e): iii

```
##              Estimate Std. Error   t value      Pr(>|t|)
## (Intercept) 5.4769450 0.11165068 49.054291 2.775161e-241
## comp_bin    1.7436451 0.30320642  5.750687  1.272936e-08
## ERvisits    0.2019283 0.02612836  7.728317  3.333127e-14
```

```r
# Compare adjusted R-squared values of models with and without 'comp_bin'
adj_r_squared_with_comp = summary(mlr_log_model)$adj.r.squared
adj_r_squared_without_comp = summary(model_no_comp_log)$adj.r.squared

adj_r_squared_with_comp
```

```
## [1] 0.1195147
```

```r
adj_r_squared_without_comp
```

```
## [1] 0.0835891
```

`comp_bin` should be retained in the final model to ensure that the results are accurate and account for its confounding effect.

(f)

```r
# Fit a multiple linear regression model with additional covariates
mlr_full_model = lm(log_totalcost ~ comp_bin + ERvisits + age + gender + duration, data = heart_disease_
summary(mlr_full_model)
```

(f): i

```
##
## Call:
## lm(formula = log_totalcost ~ comp_bin + ERvisits + age + gender +
##     duration, data = heart_disease_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1885  -0.9962  -0.0838   1.0099   4.3499
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.8016094  0.5559875  10.435  < 2e-16 ***
## comp_bin     1.5335712  0.2815721   5.446 6.89e-08 ***
## ERvisits     0.1732359  0.0245895   7.045 4.07e-12 ***
## age         -0.0193387  0.0094493  -2.047   0.0410 *
## gender      -0.3234404  0.1510866  -2.141   0.0326 *
## duration     0.0060628  0.0005325  11.386  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.769 on 782 degrees of freedom
## Multiple R-squared:  0.2502, Adjusted R-squared:  0.2454
## F-statistic: 52.18 on 5 and 782 DF,  p-value: < 2.2e-16
```

The F-statistic is 52.18 with a p-value < 2.2e-16, indicating that the model successfully explains a significant portion of the variation in total cost.

The estimated slopes for ERvisits, comp_bin, age, gender, and duration are 0.17, 1.53, -0.02, -0.32, and 0.006, respectively, when controlling for other covariates. This suggests that the number of emergency room visits, comp_bin, and duration have a positive effect on the total cost, while age and gender have a negative effect. Except for gender, all other predictors significantly impact the total cost, while the effect of gender on total cost cannot be determined conclusively based on this model.

The R-squared value is 0.2502, indicating that this model explains a larger proportion of the variance in total cost compared to previous models.

```r
# Compare the adjusted R-squared of the SLR and MLR models
slr_log_model = lm(log_totalcost ~ ERvisits, data = heart_disease_data)

# Adjusted R-squared for the SLR model
slr_adj_r2 = summary(slr_log_model)$adj.r.squared

# Adjusted R-squared for the MLR model
mlr_adj_r2 = summary(mlr_full_model)$adj.r.squared
```

```r
# Print adjusted R-squared values
slr_adj_r2
```

```
## [1] 0.0835891
```

```r
mlr_adj_r2
```

```
## [1] 0.2453885
```

```r
# Compare models using ANOVA
anova(slr_log_model, mlr_full_model)
```

```
## Analysis of Variance Table
##
## Model 1: log_totalcost ~ ERvisits
## Model 2: log_totalcost ~ comp_bin + ERvisits + age + gender + duration
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    786 2986.8
## 2    782 2447.0  4    539.86 43.132 < 2.2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
```

The MLR model is preferred as it achieves a higher Adjusted R-square and a lower Residual Standard Error compared to the SLR model. By adjusting for additional covariates, the MLR model provides a more accurate estimate of the effect of ERvisits on total cost, while the SLR model overestimates this effect.