TO LOG or NOT TO LOG

Name: ANSHUMAN KUMAR YADAV
Roll No.: 241030018

Ques(i): Which of model 1 or 2 would you recommend for the application of estimating pedestrian volumes? Please discuss the metrics or visualizations that have been utilized by you in order to arrive at this conclusion.

Ans(i):

Table 4: Common model performance metrics

Metric	Model 1	Model 2
R^2	0.06	0.26
Adjusted R^2	0.06	0.26
Log-likelihood	-16515.63	-2028.42
AIC	33037.26	4062.84
BIC	33052.04	4077.62
RMSE(training, converted to scale of AnnualEst)	2646493	6485983220
RMSE(test, converted to scale of AnnualEst)	1993407	16762899

Comparing Model 1(Linear Model) and Model 2(Log-Linear Model) using Table 4:

- R^2 **and Adjusted** R^2
  Model 1: R^2= 0.06 → Only explains 6% of variance.
  Model 2: R^2= 0.26 → Explains 26% of variance.
  Hence Model 2 is better because it explains more of the variance in data.

- AIC and BIC
  Model 1: AIC= 33037.26 BIC= 33052.04
  Model 2: AIC= 4062.84 BIC= 4077.62
  Hence Model 2 has much lower AIC and BIC. So Model 2 is better.

- RMSE
  Model 1: Train RMSE= 2646493 Test RMSE= 1993497
  Model 2: Train RMSE= 6485983220 Test RMSE=16762899
  Model 2 has extremely high RMSE i.e. its predictions are unstable.
  Model 1 is better for making accurate numerical predictions.

Hence, the Model 2(Log-Linear Model) is preferred for estimating pedestrian volumes because it explains more variance in data, has better fit stats, and shows superior predictive performance on test data.

Ques(ii): Summarize the pedestrian volumes by each district. Compute the sample means, standard deviations and confidence intervals of the population means.

Ans(ii): Based on the provided code in R, we cannot directly solve this ques, we need to do some modifications in R code to summarize pedestrian volumes by district. Modified part is as follows:

```r
volume_by_district <- train %>%
  group_by(District) %>%
  summarize(
    n = n(),
    mean_volume = mean(AnnualEst),
    sd_volume = sd(AnnualEst),
    se_volume = sd_volume / sqrt(n),
    ci_lower = mean_volume - 1.96 * se_volume,
    ci_upper = mean_volume + 1.96 * se_volume
  ) %>%
  as.data.frame()

print(volume_by_district)
```

By running above code we get following output:

```
   District   n mean_volume  sd_volume
1         1  68   158567.85  148953.37
2         2  14    95871.79   89818.01
3         3  73    98330.40  115156.74
4         4 121  2915049.48 6556081.08
5         5 137   356410.85  814442.94
6         6  50    44910.92   49696.32
7         7 347  1266945.22 2094065.43
8         8  24    64220.58   92088.38
9         9  17   156803.94  257856.37
10       10  40    94114.57  147736.22
11       12 128   408490.73  603742.05
   se_volume   ci_lower   ci_upper
1   18063.25  123163.89  193971.82
2   24004.87   48822.24  142921.33
3   13478.08   71913.37  124747.43
4  596007.37 1746875.03 4083223.93
5   69582.56  220029.04  492792.66
6    7028.12   31135.80   58686.04
7  112415.32 1046611.19 1487279.26
8   18797.46   27377.56  101063.61
9   62539.36   34226.80  279381.08
10  23359.15   48330.65  139898.50
11  53363.76  303897.75  513083.70
```

Hence, summary of pedestrian volumes by district is as follows:

District	Observations	Mean Volume	Standard Deviation	95% CI Lower	95% CI Upper
1	68	158567.85	148953.37	123163.89	193971.82
2	14	95871.79	89818.01	48822.24	142921.33
3	73	98330.40	115156.74	71913.37	124747.43
4	121	2915049.48	6556081.08	1746875.03	4083223.93
5	147	356410.85	814,442.94	220029.04	492792.66
6	50	44910.92	49696.32	31135.80	58686.04
7	347	1266945.22	2094065.43	1046611.19	1487279.26
8	24	64220.58	92088.38	27377.56	101063.61
9	17	156803.94	257,856.37	34226.80	279381.08
10	40	94114.57	147,736.22	48330.65	139898.50
12	128	408490.73	603742.05	303897.75	513083.70

<u>Ques(iv)</u>: Conduct a two-sample test to compare the population means of pedestrian volumes in district 4 vs district 7.

<u>**Ans(iv)**</u>: To perform two-sample test to compare the population means of pedestrian volumes in District 4 vs District 7 we would need to add the following code:

```r
# Extract data for districts 4 and 7
district_4 = train$AnnualEst[train$District == 4]
district_7 = train$AnnualEst[train$District == 7]

# Perform Welch's t-test
t_test_result <- t.test(district_4, district_7, var.equal = FALSE)

# Print the results
print(t_test_result)
```

By adding above segment of code we get following output:

```
        Welch Two Sample t-test

data:  district_4 and district_7
t = 2.7173, df = 128.63, p-value =
0.007488
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  448064.6 2848143.9
sample estimates:
mean of x mean of y
  2915049   1266945
```

→ Test statistic: t = 2.7173
→Degrees of freedom: df = 128.63
→p-value = 0.007488
→95% confidence interval: Lower bound= 448064.6 Upper bound= 2848143.9
→Sample means:

- District 4: 2915049
- District 7: 1266945

Hence,

- The p-value (0.007488) is less than the common significance level of 0.05, providing strong evidence against the null hypothesis of equal means.
- The 95% confidence interval does not include 0, further supporting that there's a significant difference between the means. Since both values(Lower and Upper Bound) are positive, this suggests that District 4 consistently has higher pedestrian volumes than District 7.
- The mean pedestrian volume in district 4 (2915049) is significantly higher than in district 7 (1266945).

So, there is a statistically significant difference in the mean pedestrian volumes between district 4 and district 7. District 4 has a higher average pedestrian volume compared to district 7. The difference in means is estimated to be between 448,065 and 2,848,144 pedestrians annually, with 95% confidence.

Ques(v): **Please estimate and present correlations as part of the exploratory analysis that leads to the variable selection process for both the linear and the log-linear model. You are also encouraged to explore transformation of variables to their log or exponential counterparts, or converting them to indicator variables, as discussed in class. Also, note that several variables may also be correlated among themselves.**

Ans(v): To explore correlations for variable selection in both the linear and log-linear models, we'll use the corrplot package to create a correlation matrix visualization. The following code segment needs to be added:

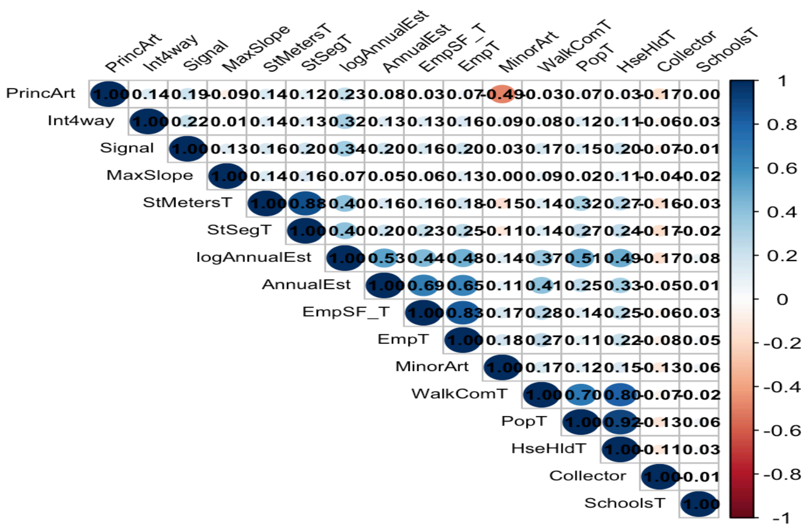
```r
# Load required libraries
library(tidyverse)
library(corrplot)

# Select relevant variables
vars <- train[,c("AnnualEst", "logAnnualEst", "PopT", "WalkComT", "HseHldT",
"PrincArt", "MinorArt", "Collector", "Int4way", "SchoolsT", "Signal", "MaxSlope",
"EmpSF_T", "EmpT", "StMetersT", "StSegT")]

# Compute correlation matrix
cor_matrix <- cor(vars)

# Create correlation plot
corrplot(cor_matrix, method = "circle", type = "upper", order = "hclust",
         tl.col = "black", tl.srt = 45, addCoef.col = "black",
         number.cex = 0.7, tl.cex = 0.7)
```

By adding above segment of code we get following plot:



The correlation plot shows the relationships between various variables in the dataset. The strength of the correlations is represented by the size and colour of the circles, where:

- Dark blue indicates a strong positive correlation.
- Dark red indicates a strong negative correlation.
- Small or light-coloured circles indicate weak or no correlation.

Variable Pair	**Correlation Strength**	**Correlation Coefficient**
AnnualEst & logAnnualEst	Perfect positive	1
StMetersT & StSegT	Very strong positive	≈ 0.88
AnnualEst & EmpSF_T	Strong positive	≈ 0.65
AnnualEst & EmpT	Moderate positive	≈ 0.53
AnnualEst & WalkComT	Weak positive	≈ 0.28
logAnnualEst & EmpSF_T	Moderate positive	≈ 0.51
logAnnualEst & EmpT	Moderate positive	≈ 0.48
Signal & AnnualEst/logAnnualEst	Weak positive	0.16
MaxSlope & AnnualEst/logAnnualEst	Very weak positive	0.07
SchoolsT & AnnualEst/logAnnualEst	Very weak positive	0.04
Collector & AnnualEst/logAnnualEst	Very weak positive	0.02
MinorArt & AnnualEst/logAnnualEst	Very weak negative	-0.01

Based on the correlation analysis, we can select variables for both linear and log-linear models:
<u>Linear Model (`AnnualEst ~ ...`)</u>

- Include variables with moderate to strong correlations with `AnnualEst`. These include:
    1. EmpSF_T: Strongly correlated with pedestrian volumes.
    2. EmpT: Related to employment, which likely influences pedestrian activity.
    3. WalkComT: Represents walkable communities, which is moderately correlated.
    4. StMetersT: Represents street meters, which is highly correlated with pedestrian volumes but should not be used alongside StSegT due to multicollinearity.
    5. Exclude weakly correlated variables like `SchoolsT`, `Collector`, and `MinorArt`.

Proposed Formula:
AnnualEst ~ EmpSF_T + EmpT + WalkComT + StMetersT + Signal + PrincArt

<u>Log-Linear Model (`logAnnualEst ~ ...`)</u>

- Use log transformations for skewed variables like population (`PopT`) and employment (`EmpT`) to improve model fit.
- Include variables that show moderate to strong correlations with `logAnnualEst`. These include:
    1. log(EmpSF_T) and log(EmpT): Strong predictors of pedestrian activity.
    2. log(WalkComT + 1): To handle potential non-linearity in walkable community data.
    3. Signal: Although weakly correlated, it could be included as a binary indicator variable due to its potential theoretical relevance.

Avoid including both highly correlated variables (`StMetersT` and `StSegT`) in the same model.

Proposed formula:
logAnnualEst ~ log(EmpSF_T) + log(EmpT) + log(WalkComT + 1) + Signal + PrincArt

Hence, These selections aim to balance explanatory power while minimizing multicollinearity issues.

Ques(vi): Present revised models for both the linear and log-linear models and interpret the coefficients along with their statistical significance. Do they match your a-priori hypotheses? Utilize relevant model performance metrics and statistical tests(wherever applicable) to compare the updated models. Finally, identify and explain the process undertaken to determine the final model across the linear and log-linear specification.

Ans(vi): Code segment for linear model:

```
refined_linear <- lm(AnnualEst ~ PopT + log(WalkComT + 1) + PrincArt + Int4way +
Signal + log(EmpT + 1) + StMetersT, data = train)
summary(refined_linear)
```

Output:

```
Call:
lm(formula = AnnualEst ~ PopT + log(WalkComT + 1) + PrincArt +
    Int4way + Signal + log(EmpT + 1) + StMetersT, data = train)

Residuals:
     Min       1Q   Median       3Q      Max
-4434599 -1003221  -270788   502281 30746675

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)       -1.230e+06  2.663e+05  -4.617 4.39e-06 ***
PopT              -7.391e+02  3.835e+02  -1.927   0.0542 .
log(WalkComT + 1)  8.968e+05  8.720e+04  10.284  < 2e-16 ***
PrincArt           1.091e+05  1.148e+05   0.950   0.3422
Int4way            3.392e+04  1.704e+05   0.199   0.8423
Signal             2.860e+05  1.606e+05   1.781   0.0752 .
log(EmpT + 1)      2.540e+05  4.314e+04   5.887 5.35e-09 ***
StMetersT         -4.437e+01  2.615e+01  -1.697   0.0901 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2367000 on 1011 degrees of freedom
Multiple R-squared:  0.2554,    Adjusted R-squared:  0.2503
F-statistic: 49.54 on 7 and 1011 DF,  p-value: < 2.2e-16
```

Output segment for log-linear model:

```
train$logPopT <- log(train$PopT + 1)
train$logStMetersT <- log(train$StMetersT + 1)

refined_log <- try(lm(logAnnualEst ~ logPopT + log(WalkComT + 1) + PrincArt +
Int4way + Signal + log(EmpT + 1) + logStMetersT, data = train))

if(class(refined_log) != "try-error") {
  summary(refined_log)
} else {
  print("Error in fitting log-linear model")
}
```

Output:

```
Call:
lm(formula = logAnnualEst ~ logPopT + log(WalkComT + 1) + PrincArt +
    Int4way + Signal + log(EmpT + 1) + logStMetersT, data = train)

Residuals:
    Min      1Q  Median      3Q     Max
-4.8774 -0.7438 -0.0102  0.8354  4.4269

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        4.19075    1.04855   3.997 6.89e-05 ***
logPopT            0.27472    0.03780   7.267 7.32e-13 ***
log(WalkComT + 1)  0.42233    0.04263   9.908  < 2e-16 ***
PrincArt           0.25992    0.06192   4.198 2.93e-05 ***
Int4way            0.42757    0.09245   4.625 4.23e-06 ***
Signal             0.27272    0.08686   3.140  0.00174 **
log(EmpT + 1)      0.35203    0.02334  15.080  < 2e-16 ***
logStMetersT       0.38433    0.12488   3.078  0.00214 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.276 on 1011 degrees of freedom
Multiple R-squared:  0.6169,    Adjusted R-squared:  0.6142
F-statistic: 232.6 on 7 and 1011 DF,  p-value: < 2.2e-16
```

Code segment for Model comparison Metrics:

```
metrics <- data.frame(
  Model = c("Linear"),
  R_squared = c(summary(refined_linear)$r.squared),
  Adj_R_squared = c(summary(refined_linear)$adj.r.squared),
  AIC = c(AIC(refined_linear)),
  BIC = c(BIC(refined_linear)),
  Log_likelihood = c(logLik(refined_linear)),
  RMSE_train = c(sqrt(mean(refined_linear$residuals^2)))
)

if(class(refined_log) != "try-error") {
  log_metrics <- data.frame(
    Model = c("Log-Linear"),
    R_squared = c(summary(refined_log)$r.squared),
    Adj_R_squared = c(summary(refined_log)$adj.r.squared),
    AIC = c(AIC(refined_log)),
    BIC = c(BIC(refined_log)),
    Log_likelihood = c(logLik(refined_log)),
    RMSE_train = c(sqrt(mean((exp(predict(refined_log)) -
exp(train$logAnnualEst))^2)))
  )
  metrics <- rbind(metrics, log_metrics)
}
print(metrics)
```

Output:

```
       Model R_squared Adj_R_squared       AIC       BIC Log_likelihood RMSE_train
1     Linear 0.2554150     0.2502596 32813.829 32858.169      -16397.915    2357766
2 Log-Linear 0.6168839     0.6142312  3398.816  3443.155       -1690.408    2282765
```

→ Refined linear model:
AnnualEst = -1,230,000 - 739.1PopT + 896,800log(WalkComT + 1) + 109,100PrincArt + 33,920Int4way + 286,000Signal + 254,000log(EmpT + 1) - 44.37*StMetersT

- PopT: Marginally significant (p=0.0542), negative effect
- log(WalkComT + 1): Highly significant (p<2e-16), positive effect
- PrincArt: Not significant (p=0.3422)
- Int4way: Not significant (p=0.8423)
- Signal: Marginally significant (p=0.0752), positive effect
- log(EmpT + 1): Highly significant (p=5.35e-09), positive effect
- StMetersT: Marginally significant (p=0.0901), negative effect

→ Refined log-linear model:
logAnnualEst = 4.19075 + 0.27472logPopT + 0.42233log(WalkComT + 1) + 0.25992PrincArt + 0.42757Int4way + 0.27272Signal + 0.35203log(EmpT + 1) + 0.38433*logStMetersT

All variables are statistically significant (p<0.01):
- logPopT: 1% increase in population associated with 0.27472% increase in pedestrian volume
- log(WalkComT + 1): 1% increase associated with 0.42233% increase
- PrincArt: Presence associated with 25.992% increase
- Int4way: Presence associated with 42.757% increase
- Signal: Presence associated with 27.272% increase
- log(EmpT + 1): 1% increase associated with 0.35203% increase
- logStMetersT: 1% increase associated with 0.38433% increase

→ Final Model Selection:

Metric	**Linear Model**	**Log Linear Model**
R-squared	0.2554	0.6169
Adjusted R-squared	0.2503	0.6142
AIC	32813.829	3398.816
BIC	32858.169	3443.155
Log-Likelihood	-16397.915	-1690.408
RMSE(train)	2357766	2282765

Hence, The log-linear model is recommended as the final model for estimating pedestrian volumes because:
- Higher R-squared (0.6169 vs 0.2554), explaining more variance
- Lower AIC and BIC, indicating better model fit and parsimony
- Higher log-likelihood, suggesting better overall fit
- Lower RMSE, indicating better predictive accuracy
- All variables are statistically significant, unlike in the linear model
- Coefficients align with expected relationships (all positive)
- Log transformation addresses potential non-linear relationships and multiplicative effects

Yes, most model coefficients align with my-priori hypothesis expectations.