COVID-19 Spatial Dynamics in the US: Spatiotemporal Analysis of Predicting Covid Cases on Correlation Length and Local Correlation with Time-invariant Controls

Aarushi Somani, Eleanor Kim, Lauren Murai, Meichen Chen

2024-04-16

```r
#load packages
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

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

```r
library(tibble)
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```r
library(car)
```

```
## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

Read in 9 Datasets

All county data + Urban county data with Wave dummies

```r
# Weekly Spatial Metrics with Time Wave Columns
all_weekly_spatial_metrics_waves = read.csv("weekly_spatial_metrics_waves.csv")

# same data set but calculations from urban counties only (pop >250k)
urban_weekly_spatial_metrics_waves = read.csv("urban_weekly_spatial_metrics_waves.csv")
```

All county data + Urban county data with Region + Wave dummies

```r
# Weekly Spatial Metrics with Region Columns
all_regional_weekly_spatial_metrics = read.csv("regional_weekly_spatial_metrics.csv")

# same data set but calculations from urban counties only (pop >250k)
urban_regional_weekly_spatial_metrics = read.csv("urban_regional_weekly_spatial_metrics.csv")
```

All county data + Urban county data with MHHINC + Wave dummies

```r
# Weekly Spatial Metrics with Median Household Income Columns
all_mhhinc_weekly_spatial_metrics = read.csv("mhhinc_weekly_spatial_metrics.csv")

# same data set but calculations from urban counties only (pop >250k)
urban_mhhinc_weekly_spatial_metrics = read.csv("urban_mhhinc_weekly_spatial_metrics.csv")
```

All county data + Urban county data with Region + MHHINC + Wave dummies

```r
# Local Corelation with Region + MHHINC Columns
all_region_mhhinc_weekly_localcor = read.csv("region_mhhinc_weekly_localcor.csv")

# same data set but calculations from urban counties only (pop >250k)
urban_region_mhhinc_weekly_localcor = read.csv("urban_region_mhhinc_weekly_localcor.csv")
```

All county data with Mask Usage + Wave dummies

```
# Weekly Spatial Metrics with Mask Usaege Columns
all_mask_weekly_spatial_metrics = read.csv("all_mask_weekly_spatial_metrics.csv")
```

Simple Models

Prepare Model Combinations

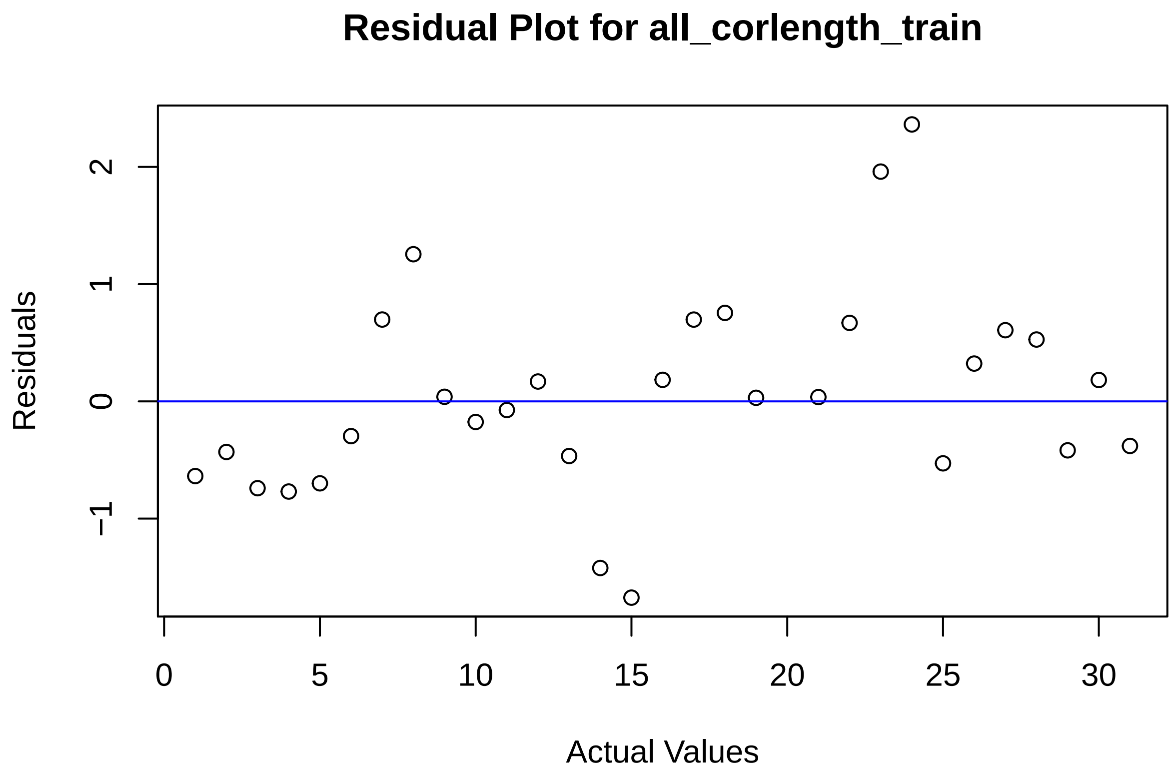
```
# Data sets: (1) all counties & (2) urban counties
datasets = list(all_weekly_spatial_metrics_waves, urban_weekly_spatial_metrics_waves)
df_names = c("all","urban")

# Predictor: (1) correlation length & (2) log(local correlation)
predictor = c("cor_lengths", "r_0_50")
pred_names = c("corlength","localcor")
```

Generate Models

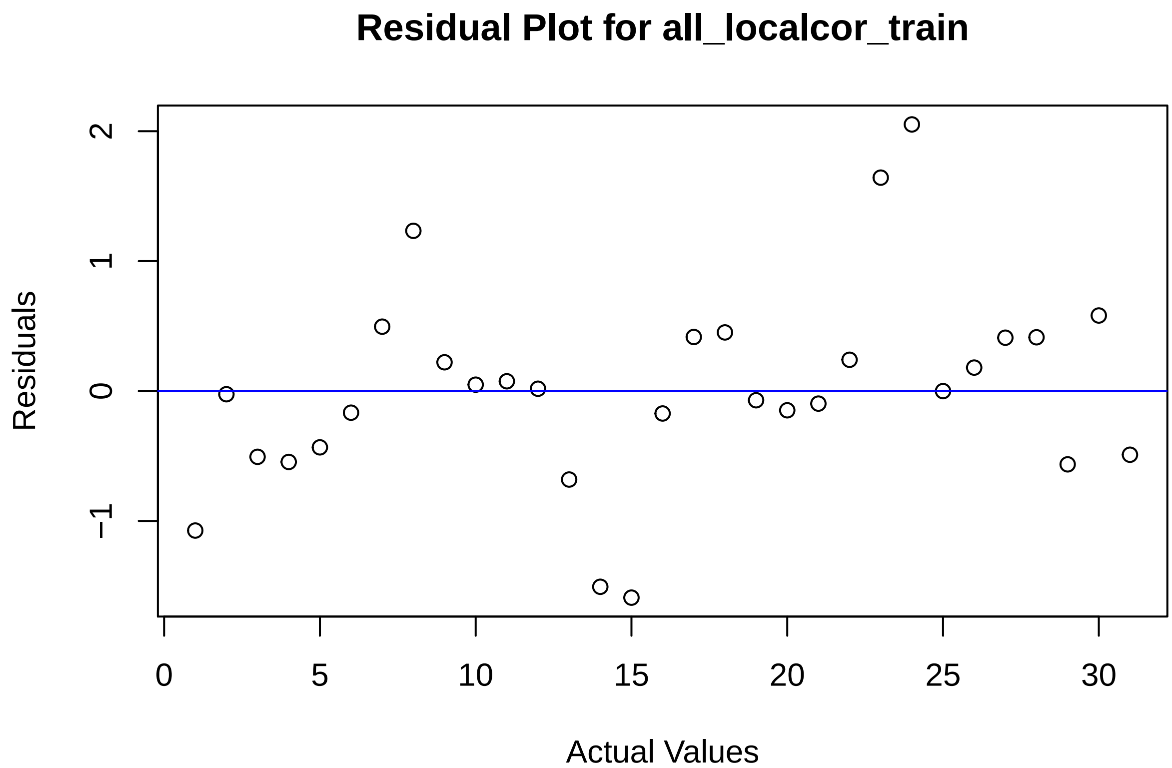
Training/Testing Models

```
# Set seed for reproducibility
set.seed(1)

# Create an empty list to store the models
simple_lm_models_train <- list()

# Define the train-test split ratio
train_ratio <- 0.8
test_ratio <- 1 - train_ratio

# Apply the trained model to the test data and make predictions
for (d in 1:length(datasets)) {
  df <- datasets[d][[1]]
  n_obs <- nrow(df)

  # Create indices for train-test split
  train_indices <- sample(1:n_obs, size = round(train_ratio * n_obs), replace = FALSE)
  test_indices <- setdiff(1:n_obs, train_indices)

  # Split the data into train and test sets
  train_df <- df[train_indices, ]
  test_df <- df[test_indices, ]

  for (p in 1:length(predictor)) {

    # Create the formula
    lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", predictor[p]))
```

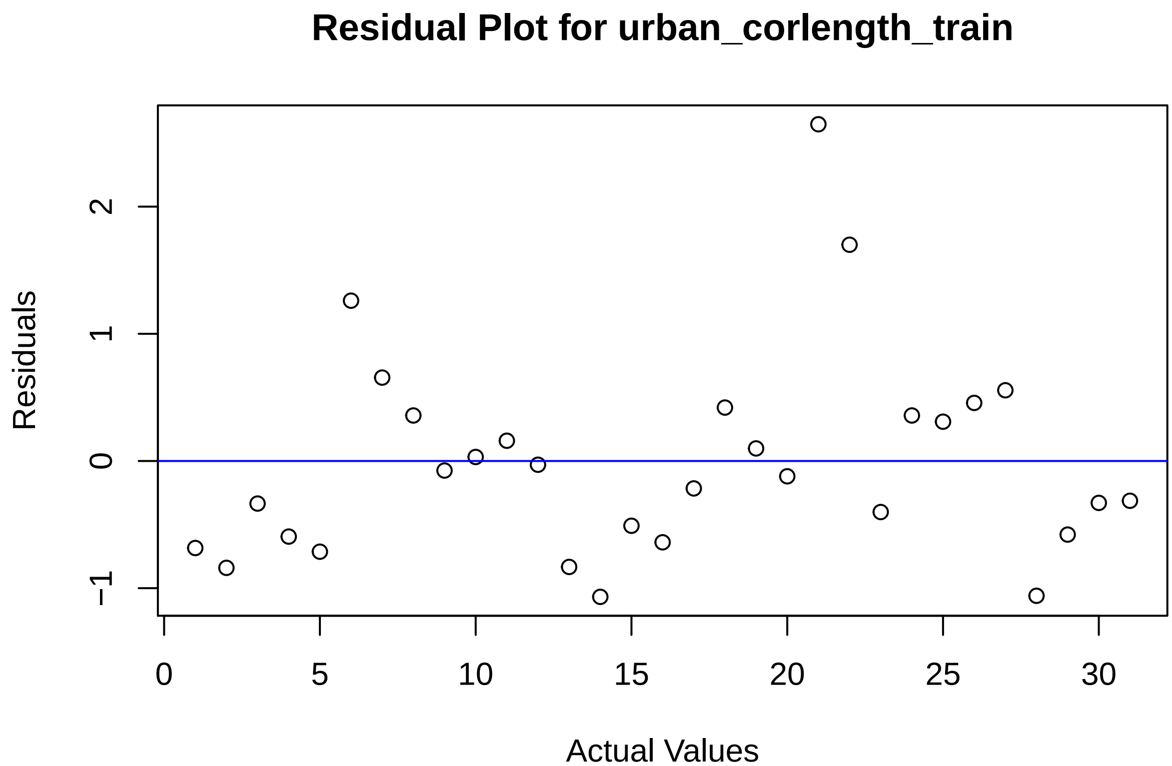
```
    # Fit the linear model on the train set
    lm_model_train <- lm(lm_formula, data = train_df)

    # Generate a name for the model object
    model_name_train <- paste(df_names[d], "_", pred_names[p], "_train", sep = "")

    # Save the train model to the list
    simple_lm_models_train[[model_name_train]] <- lm_model_train

    # Apply the trained model to the test data and make predictions
    test_predictions <- predict(lm_model_train, newdata = test_df)

    # Calculate residuals using predictions from the test dataset
    residuals <- log(test_df$next_week_marginal_cases)-test_predictions

    # Create residual plot
    plot(1:length(residuals),residuals,
         main = paste("Residual Plot for", model_name_train),
         xlab = "Actual Values",
         ylab = "Residuals")
    abline(h = 0, col = "blue")

    #is the mean of residuals less than standard deviation of data?
    print(mean(abs(residuals), na.rm=TRUE) < sd(log(test_df$next_week_marginal_cases), na.rm=TRUE))


  }
}
```


Residual Plot for all_corlength_train
Residuals
Actual Values
2
1
0
−1
0
5
10
15
20
25
30

```
## [1] TRUE
```

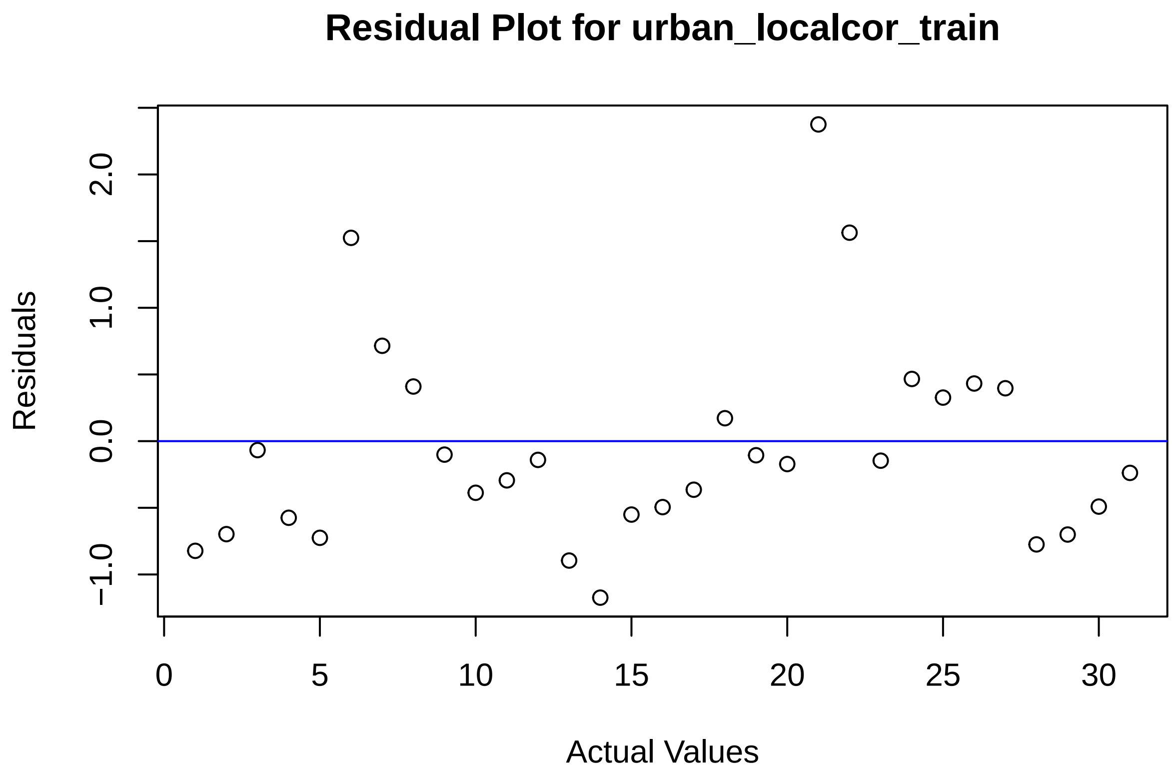

Residual Plot for all_localcor_train
Residuals
Actual Values
2
1
0
−1
0
5
10
15
20
25
30

```
## [1] TRUE
```



Residual Plot for urban_corlength_train
Residuals
2
1
0
−1
0
5
10
15
20
25
30
Actual Values


```
## [1] TRUE
```




```
## [1] TRUE
```

```r
names(simple_lm_models_train)
```

```
## [1] "all_corlength_train"   "all_localcor_train"    "urban_corlength_train"
## [4] "urban_localcor_train"
```

View Model Summaries

```r
for (m in 1:length(simple_lm_models_train)){
  print(summary(simple_lm_models_train[[m]]))
}
```

```
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4739 -0.5177 -0.1375  0.4399  2.7850
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.426e+01  1.985e-01  71.813  < 2e-16 ***
## cor_lengths 9.781e-04  2.815e-04   3.475 0.000722 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7357 on 115 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.09501,    Adjusted R-squared:  0.08714
## F-statistic: 12.07 on 1 and 115 DF,  p-value: 0.000722
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.53340 -0.36854 -0.05406  0.38658  2.44181
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   13.848      0.176  78.689  < 2e-16 ***
## r_0_50         2.089      0.324   6.448 2.38e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6517 on 122 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.2542, Adjusted R-squared:  0.248
## F-statistic: 41.57 on 1 and 122 DF,  p-value: 2.382e-09
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.98320 -0.40177 -0.07564  0.33458  2.59276
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.405e+01  1.402e-01 100.211  < 2e-16 ***
## cor_lengths 9.618e-04  2.724e-04   3.532 0.000584 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7457 on 122 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.09275,    Adjusted R-squared:  0.08531
## F-statistic: 12.47 on 1 and 122 DF,  p-value: 0.0005839
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6977 -0.4661 -0.1244  0.3781  2.5549
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  13.1490     0.3320  39.607  < 2e-16 ***
## r_0_50        1.8538     0.4517   4.104 7.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7338 on 122 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.1213, Adjusted R-squared:  0.1141
## F-statistic: 16.85 on 1 and 122 DF,  p-value: 7.359e-05
```

```
# stargazer table for all_corlengths_region and urban_corlengths_region
# (1) is all
# (2) is urban
stargazer(
  c(simple_lm_models_train[1], simple_lm_models_train[3], simple_lm_models_train[2], simple_lm_models_t
  type = 'text',
  title = "Regression of Covid Cases on Correlation Length"
)
```

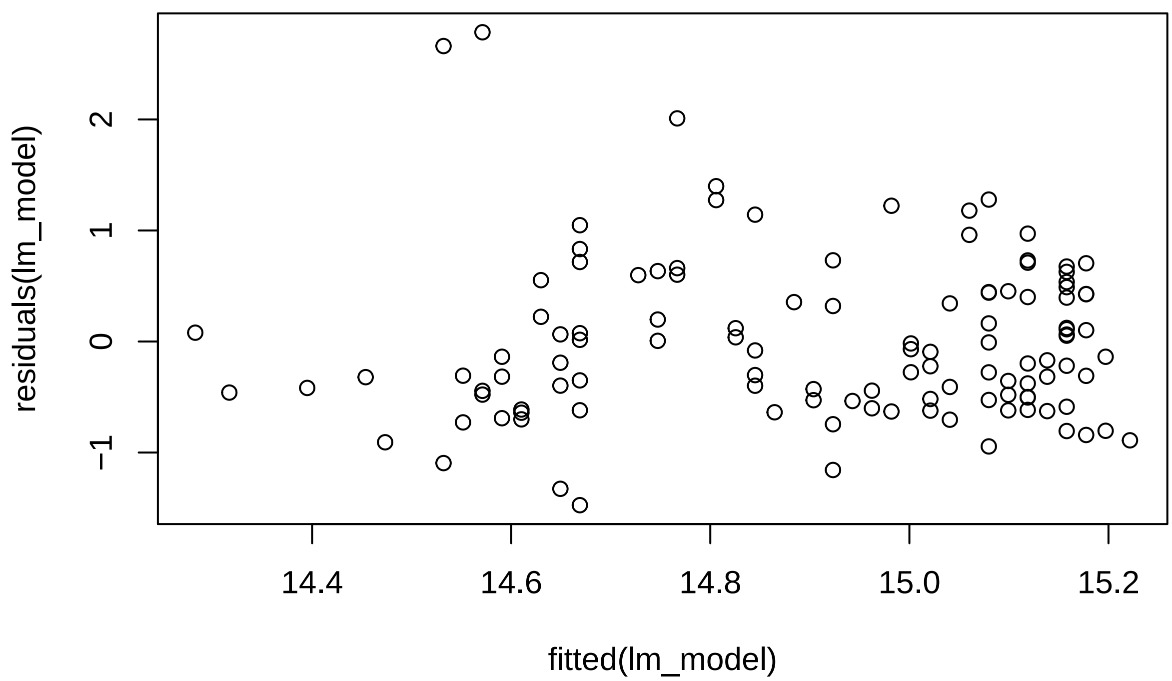
```
##
## Regression of Covid Cases on Correlation Length
## ==================================================================================================
##                                                          Dependent variable:
##                     ------------------------------------------------------------------------------
##                                                  log(next_week_marginal_cases)
##                               (1)                    (2)                    (3)
## --------------------------------------------------------------------------------------------------
## cor_lengths                0.001***               0.001***
##                            (0.0003)               (0.0003)
##
## r_0_50                                                                   2.089***               1
##                                                                           (0.324)
##
## Constant                   14.258***              14.050***              13.848***              1
##                             (0.199)                (0.140)                (0.176)
##
## --------------------------------------------------------------------------------------------------
## Observations                 117                    124                    124
## R2                          0.095                  0.093                  0.254
## Adjusted R2                 0.087                  0.085                  0.248
## Residual Std. Error    0.736 (df = 115)       0.746 (df = 122)       0.652 (df = 122)       0.734
## F Statistic         12.074*** (df = 1; 115) 12.472*** (df = 1; 122) 41.574*** (df = 1; 122) 16.846**
## ==================================================================================================
## Note:                                                                                 *p<0.1; **p<
```

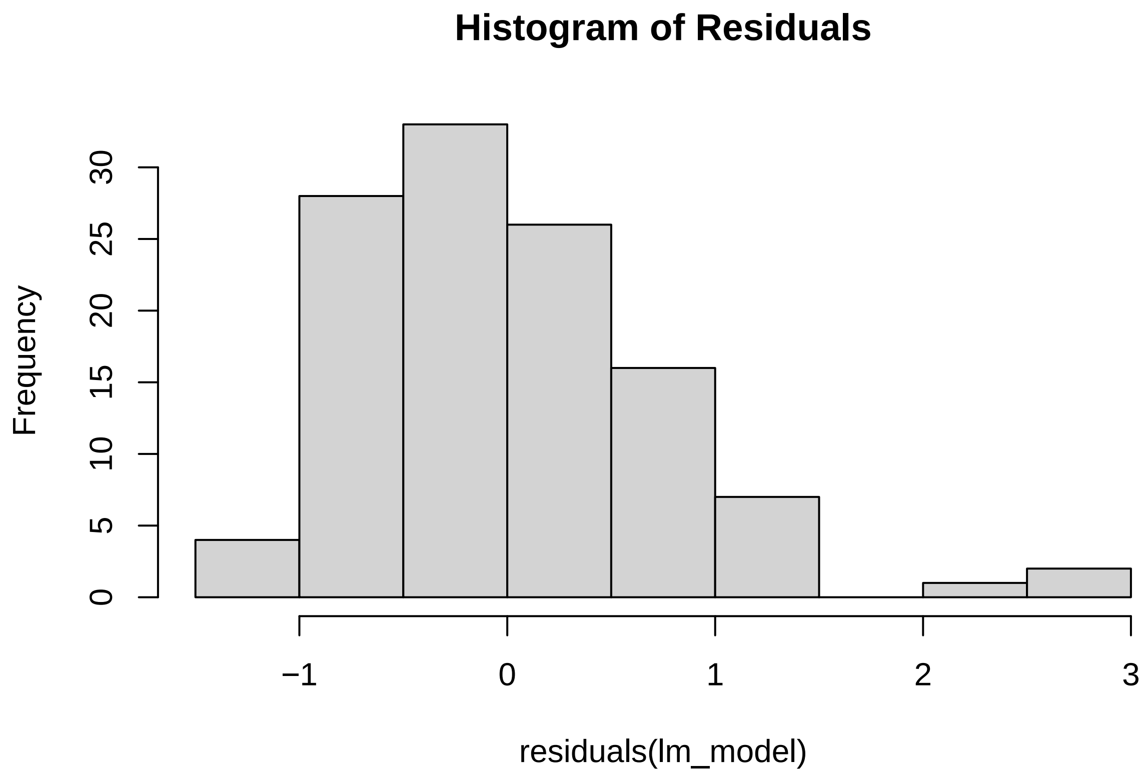
```r
# all_localcor_wave_region and urban_localcor_wave_region
# (1) is all
# (2) is urban
stargazer(
  c(simple_lm_models_train[2], simple_lm_models_train[4]),
  type = 'text',
  title = "Regression of Covid Cases on Local Correlation"
)
```

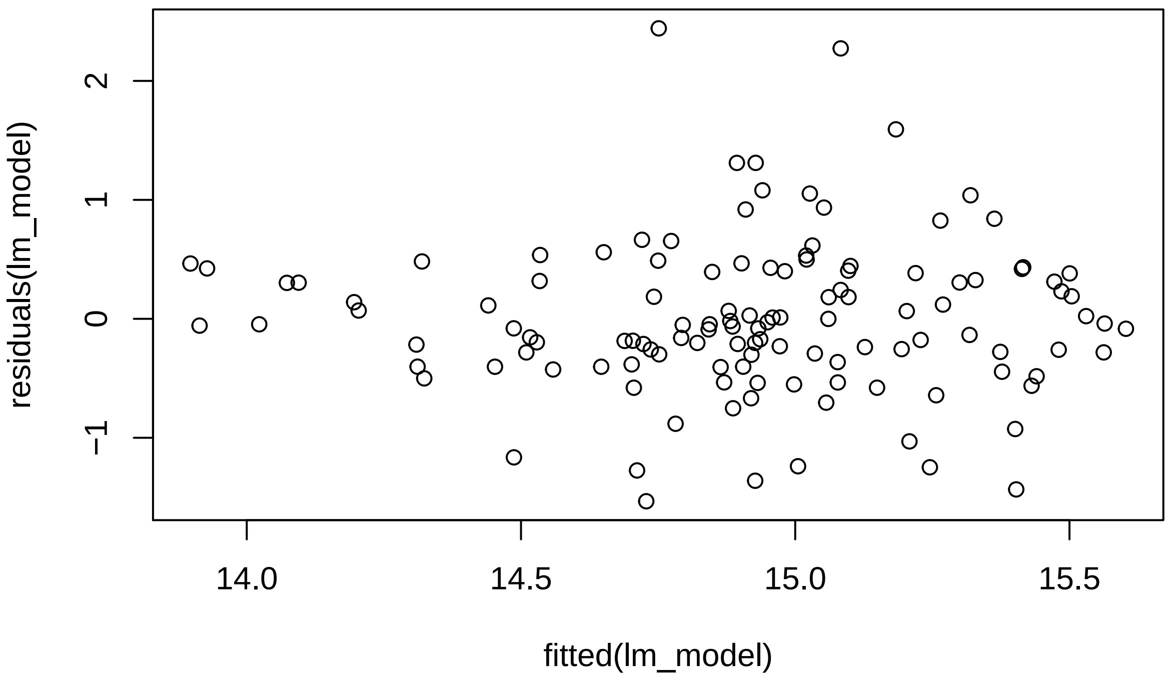
```
##
## Regression of Covid Cases on Local Correlation
## ==============================================================
##                                    Dependent variable:
##                               --------------------------------
##                               log(next_week_marginal_cases)
##                                    (1)             (2)
## --------------------------------------------------------------
## r_0_50                          2.089***        1.854***
##                                  (0.324)         (0.452)
##
## Constant                        13.848***       13.149***
##                                  (0.176)         (0.332)
##
## --------------------------------------------------------------
## Observations                      124             124
## R2                               0.254           0.121
## Adjusted R2                      0.248           0.114
## Residual Std. Error (df = 122)   0.652           0.734
## F Statistic (df = 1; 122)       41.574***       16.846***
## ==============================================================
## Note:                           *p<0.1; **p<0.05; ***p<0.01
```

Check conditions of Normality of Residuals, Homoskedasticity, Linearity, Independence of Residuals

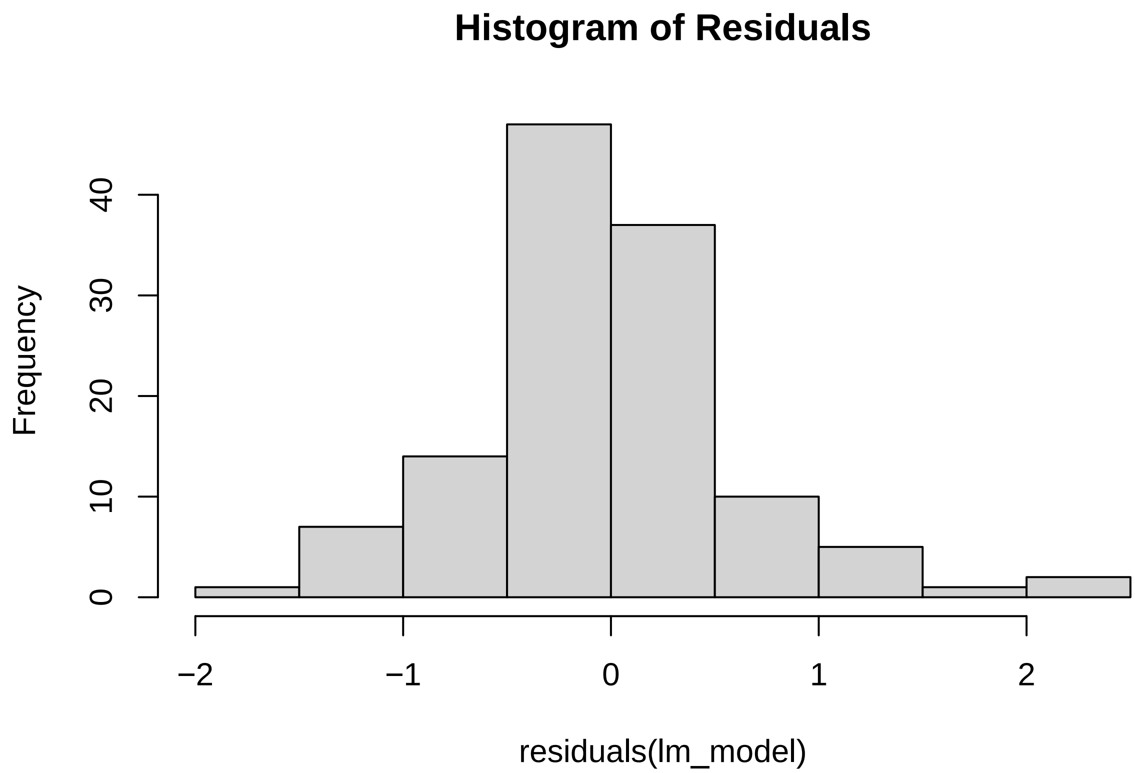
```r
for (i in 1:length(simple_lm_models_train)) {
lm_model = simple_lm_models_train[[i]]

# 1. Linearity Check (Visual Inspection)
plot(residuals(lm_model) ~ fitted(lm_model))

# 2. Independence Check# Durbin-Watson test
durbinWatsonTest(lm_model)

# 3. Homoscedasticity Check (Visual Inspection)
# Breusch-Pagan test for Heteroscedasticity
bptest(lm_model)

# 4. Normality of Residuals Check
# Histogram of residuals
hist(residuals(lm_model), main = "Histogram of Residuals")
}
```

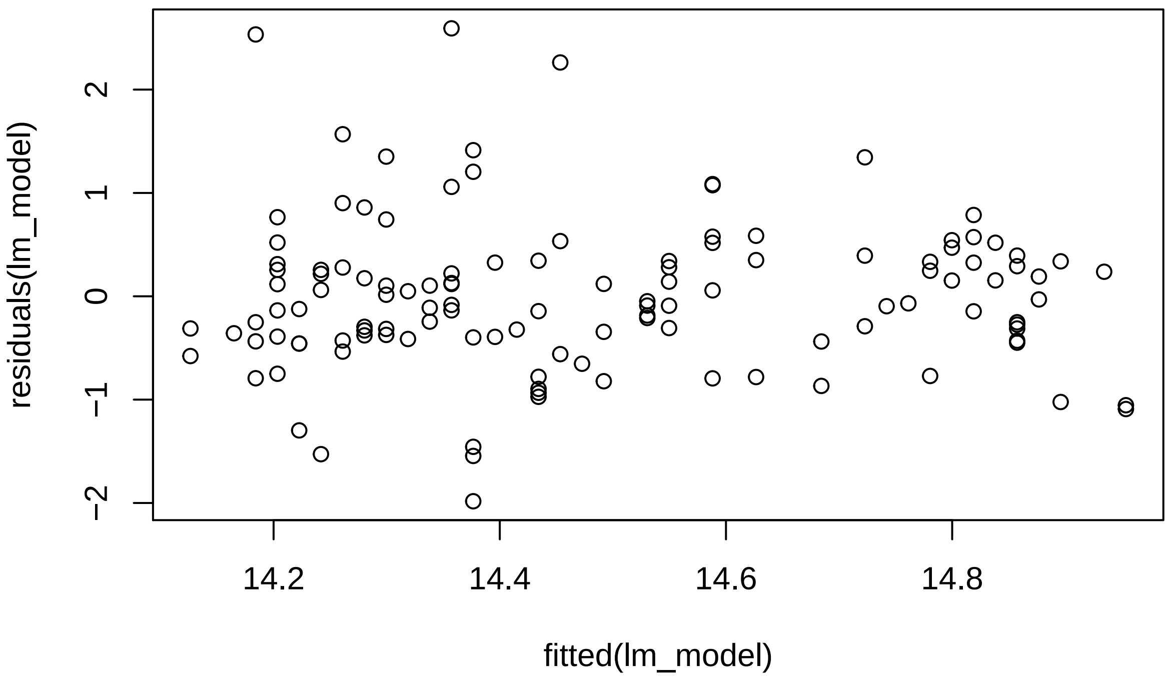

residuals(lm_model)
2
1
0
−1
14.4
14.6
14.8
15.0
15.2
fitted(lm_model)

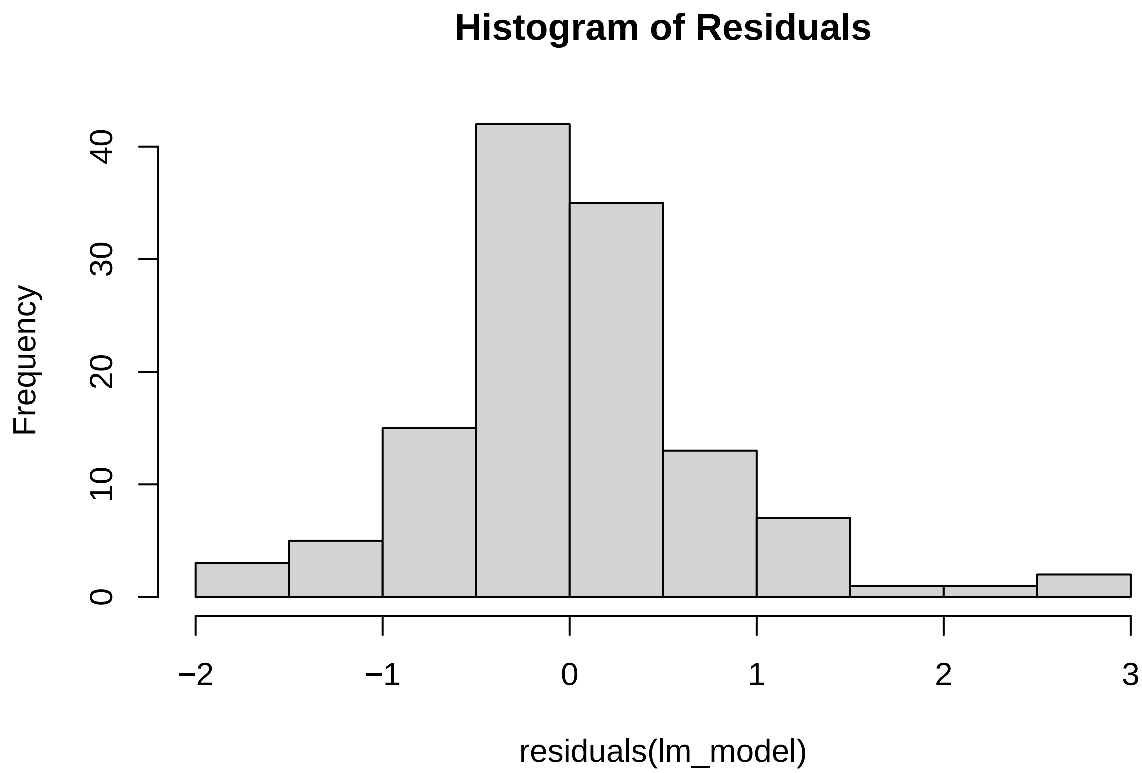

Histogram of Residuals
Frequency
0
5
10
15
20
25
30
−1
0
1
2
3
residuals(lm_model)

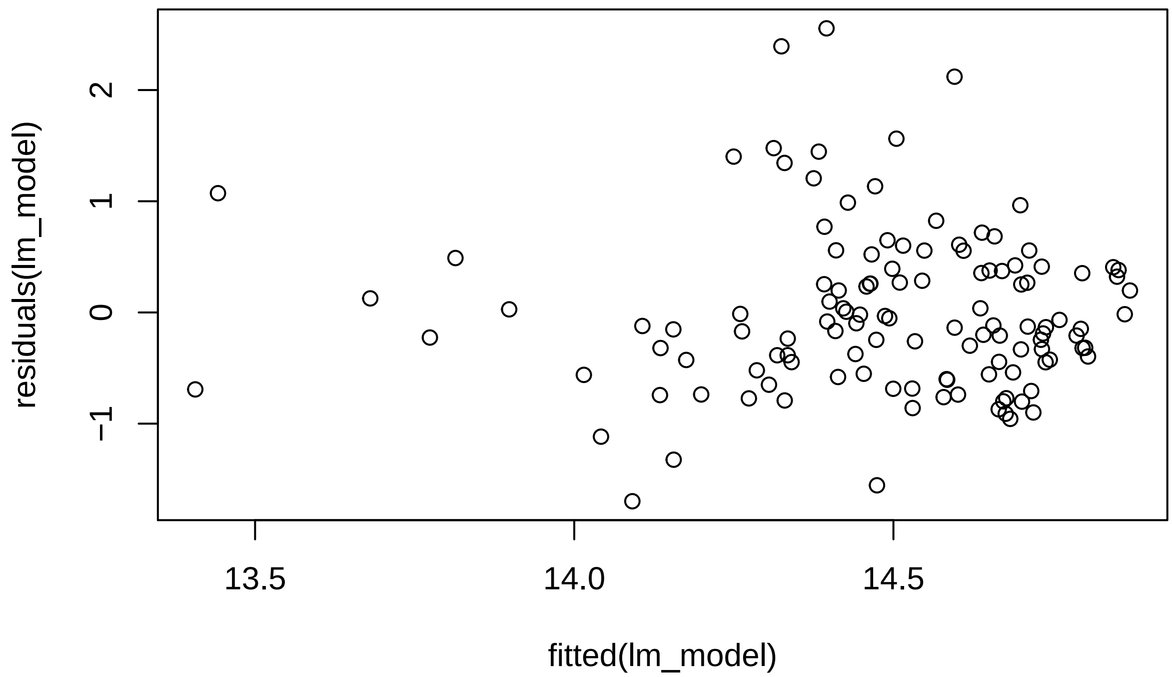


residuals(lm_model)
2
1
0
−1
14.0
14.5
15.0
15.5
fitted(lm_model)

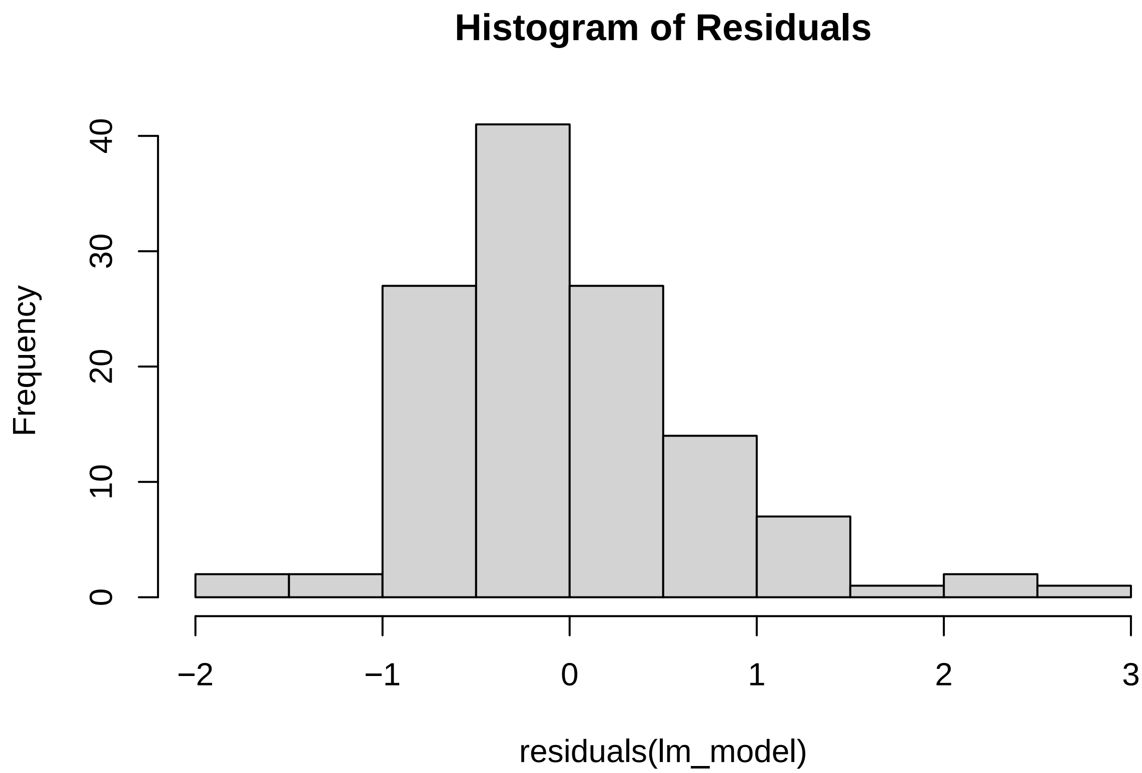

Histogram of Residuals
Frequency
0
10
20
30
40
−2
−1
0
1
2
residuals(lm_model)


residuals(lm_model)
fitted(lm_model)
−2
−1
0
1
2
14.2
14.4
14.6
14.8


Histogram of Residuals
Frequency
0
10
20
30
40
−2
−1
0
1
2
3
residuals(lm_model)



residuals(lm_model)
fitted(lm_model)
13.5
14.0
14.5
−1
0
1
2




```
# Heteroskedasticity/Non-linearity in Urban Local Correlation Model
```

Interpretations

```
units = c(round(sd(all_weekly_spatial_metrics_waves$cor_lengths, na.rm = TRUE)), round( sd(all_weekly_sp
var = c("correlation length","local correlation")
for (i in 1:4) {
  adj_i = 2-i%%2
  print(paste("a 1 sd increase in ", var[adj_i]," (",units[adj_i], ") corresponds to a ", round(simple_l
```

```
## [1] "a 1 sd increase in correlation length (244) corresponds to a 23.87% change in cases in the foll
## [1] "a 1 sd increase in local correlation (0.18) corresponds to a 37.6% change in cases in the follow
## [1] "a 1 sd increase in correlation length (244) corresponds to a 23.47% change in cases in the foll
## [1] "a 1 sd increase in local correlation (0.18) corresponds to a 33.37% change in cases in the foll
```

Effects are less strong when comparing across only the urban counties. Local correlation and correlation length seem to both hold a strong effect in indicating/predicting a rise in cases in the next week.

Region + Median Household Income Models

Prepare Model Combinations

```
# Data sets: (1) all counties & (2) urban counties
datasets = list(all_region_mhhinc_weekly_localcor, urban_region_mhhinc_weekly_localcor)
df_names = c("all_","urban_")

# Predictor: (1) correlation length & (2) local correlation
predictor = c("r_0_50")
pred_names = c("localcor")

# Covariates (1) region + mhhinc & (2) region + mhhinc + wave
covariates = list(c("*factor(region)*same_tier","*factor(region)*tier_diff_1","*factor(region)*tier_dif
var_names = c("region_mhhinc")
```

Generate Models

```
# Create an empty list to store the models
region_tiers_lm_models <- list()

# Iterate over every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df = datasets[d][[1]]
for (p in 1:length(predictor)) {
  for (c in 1:length(covariates)) {
    interactions = c()
    for (i in 1:length(covariates[[c]])) {
      interactions[i] <- paste(predictor[p], covariates[[c]][i]) }

    # Create the formula & fit linear model
    lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), colla
    lm_model <- lm(lm_formula, data = df)

    # Name model and save to list
    model_name <- paste(df_names[d], pred_names[p], "_", var_names[c], sep = "")
    region_tiers_lm_models[[model_name]] <- lm_model}}}
```

```
## Warning in log(next_week_marginal_cases): NaNs produced

## Warning in log(next_week_marginal_cases): NaNs produced
```

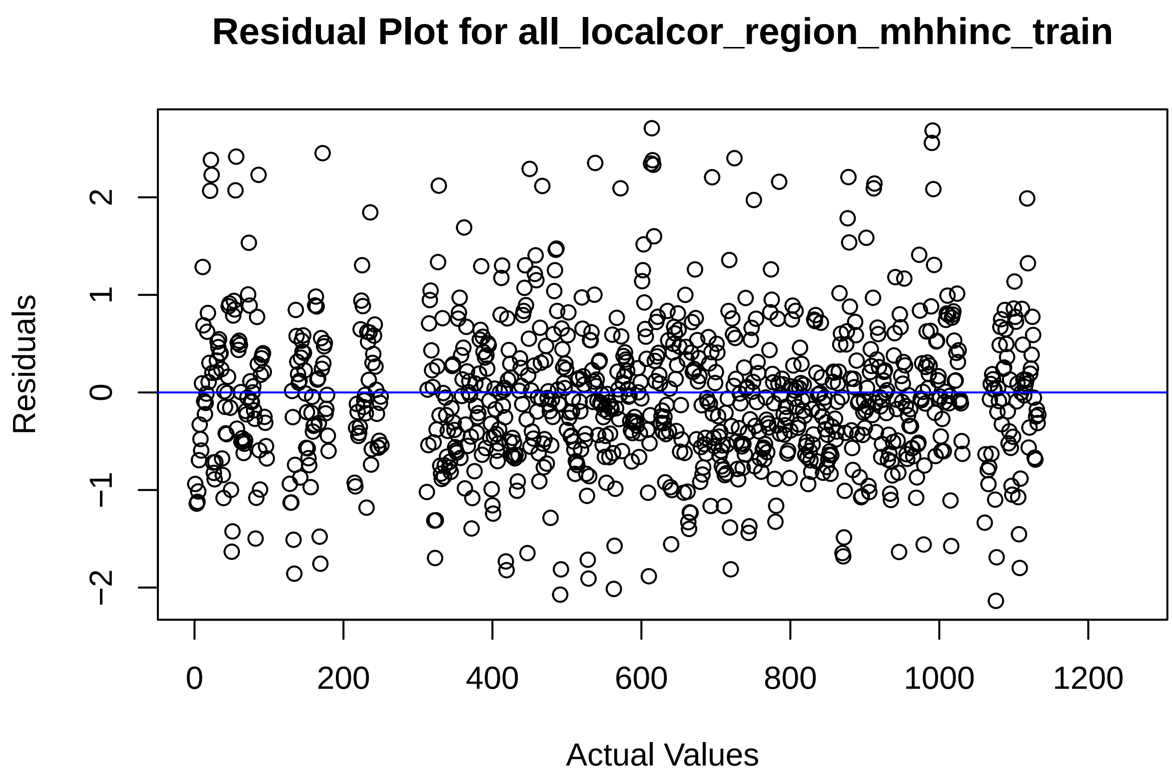
```
names(region_tiers_lm_models)
```

```
## [1] "all_localcor_region_mhhinc"   "urban_localcor_region_mhhinc"
```

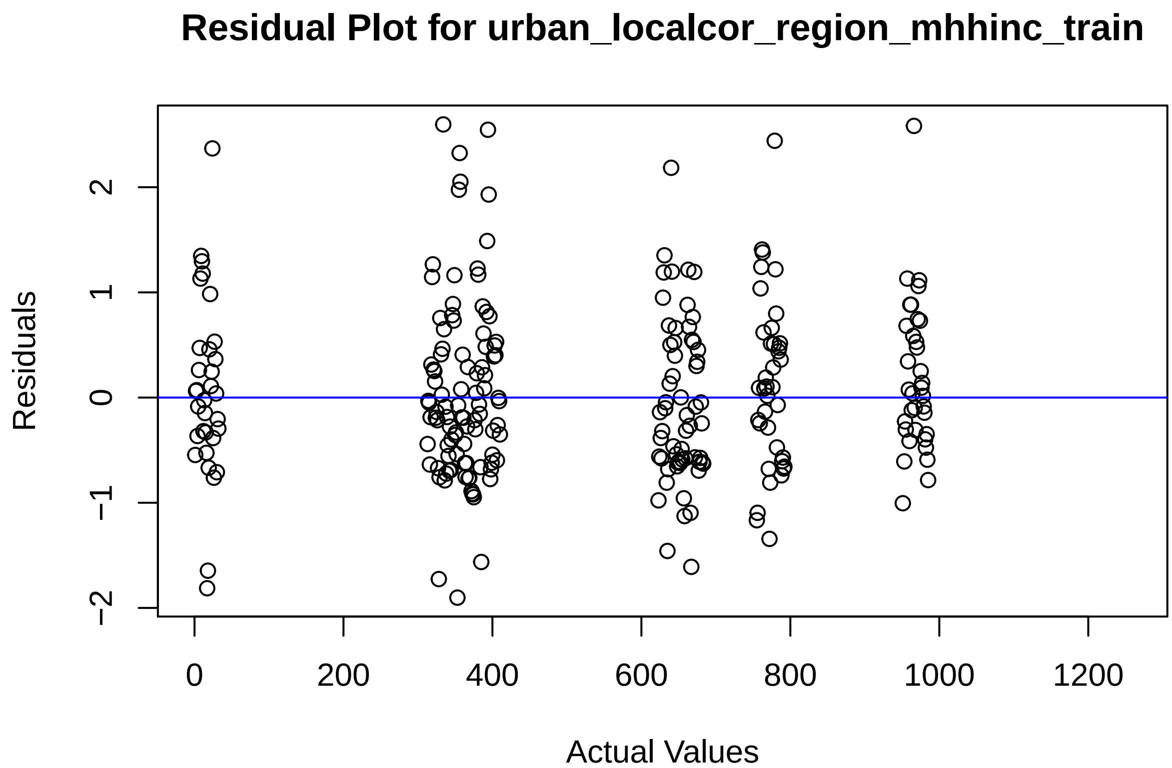
Training/Testing Models

```r
# Set seed for reproducibility
set.seed(1)

# Create an empty list to store the models
region_mhhinc_lm_models_train <- list()

# Iterate over every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df <- datasets[d][[1]]

  # Create indices for train-test split
  n_obs <- nrow(df)
  train_indices <- sample(1:n_obs, size = round(train_ratio * n_obs), replace = FALSE)
  test_indices <- setdiff(1:n_obs, train_indices)

  # Split the data into train and test sets
  train_df <- df[train_indices, ]
  test_df <- df[test_indices, ]

  for (p in 1:length(predictor)) {
    for (c in 1:length(covariates)) {
      interactions <- c()
      for (i in 1:length(covariates[[c]])) {
        interactions[i] <- paste(predictor[p], covariates[[c]][i])
      }

      # Create the formula & fit linear model on the training data
      lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), col
      lm_model_train <- lm(lm_formula, data = train_df)

      # Name model and save to list
      model_name_train <- paste(df_names[d], pred_names[p], "_", var_names[c], "_train", sep = "")
      region_mhhinc_lm_models_train[[model_name_train]] <- lm_model_train

      # Apply the trained model to the test data and make predictions
      test_predictions <- predict(lm_model_train, newdata = test_df)

      # Calculate residuals using predictions from the test dataset
      residuals <- log(test_df$next_week_marginal_cases) - test_predictions

      # Create residual plot
      plot(1:length(residuals), residuals,
           main = paste("Residual Plot for", model_name_train),
           xlab = "Actual Values",
           ylab = "Residuals")
      abline(h = 0, col = "blue")

      # Check if the mean of residuals is less than the standard deviation of the data
      print(mean(abs(residuals), na.rm = TRUE) < sd(log(test_df$next_week_marginal_cases), na.rm = TRUE
    }
  }
}
```

```
## Warning in log(next_week_marginal_cases): NaNs produced

## Warning in predict.lm(lm_model_train, newdata = test_df): prediction from a
## rank-deficient fit may be misleading

## Warning in log(test_df$next_week_marginal_cases): NaNs produced

## Warning in log(test_df$next_week_marginal_cases): NaNs produced

## [1] TRUE

## Warning in log(next_week_marginal_cases): NaNs produced

## Warning in predict.lm(lm_model_train, newdata = test_df): prediction from a
## rank-deficient fit may be misleading

## Warning in log(test_df$next_week_marginal_cases): NaNs produced
```




```
## Warning in log(test_df$next_week_marginal_cases): NaNs produced
```




```
## [1] TRUE
```

```r
names(region_mhhinc_lm_models_train)
```

```
## [1] "all_localcor_region_mhhinc_train"   "urban_localcor_region_mhhinc_train"
```

Urban fit is suspicious

View Model Summaries

```r
for (m in 1:length(region_mhhinc_lm_models_train)){
  print(summary(region_mhhinc_lm_models_train[[m]]))
}
```

```
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.15303 -0.49470 -0.05571  0.44359  2.79140
##
```

```
## Coefficients: (4 not defined because of singularities)
##                                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                14.79590    0.13223 111.895  < 2e-16
## r_0_50                                      0.31229    0.21319   1.465 0.143036
## factor(region)Northeast                     0.29805    0.17963   1.659 0.097143
## factor(region)South                        -0.23734    0.19692  -1.205 0.228179
## factor(region)West                          0.25550    0.15160   1.685 0.091999
## same_tier                                  -0.43368    0.15472  -2.803 0.005088
## tier_diff_1                                -0.54935    0.16876  -3.255 0.001143
## tier_diff_2                                -0.35046    0.17655  -1.985 0.047204
## r_0_50:factor(region)Northeast             -0.57198    0.36047  -1.587 0.112649
## r_0_50:factor(region)South                  0.59155    0.38449   1.539 0.124003
## r_0_50:factor(region)West                  -0.60022    0.26587  -2.258 0.024028
## r_0_50:same_tier                            0.85931    0.26816   3.204 0.001365
## factor(region)Northeast:same_tier           0.20728    0.20887   0.992 0.321055
## factor(region)South:same_tier               0.10897    0.22818   0.478 0.633002
## factor(region)West:same_tier                0.06642    0.19138   0.347 0.728562
## r_0_50:tier_diff_1                          1.05966    0.29689   3.569 0.000362
## factor(region)Northeast:tier_diff_1         0.14705    0.23602   0.623 0.533291
## factor(region)South:tier_diff_1             0.28924    0.23965   1.207 0.227526
## factor(region)West:tier_diff_1              0.12776    0.20878   0.612 0.540616
## r_0_50:tier_diff_2                          0.62422    0.31005   2.013 0.044152
## factor(region)Northeast:tier_diff_2              NA         NA      NA       NA
## factor(region)South:tier_diff_2            -0.02633    0.25489  -0.103 0.917731
## factor(region)West:tier_diff_2                   NA         NA      NA       NA
## r_0_50:factor(region)Northeast:same_tier   -0.51726    0.42620  -1.214 0.224950
## r_0_50:factor(region)South:same_tier       -0.16212    0.45482  -0.356 0.721520
## r_0_50:factor(region)West:same_tier        -0.00392    0.34881  -0.011 0.991034
## r_0_50:factor(region)Northeast:tier_diff_1 -0.37835    0.46721  -0.810 0.418101
## r_0_50:factor(region)South:tier_diff_1     -0.41563    0.47780  -0.870 0.384424
## r_0_50:factor(region)West:tier_diff_1      -0.31640    0.37416  -0.846 0.397818
## r_0_50:factor(region)Northeast:tier_diff_2       NA         NA      NA       NA
## r_0_50:factor(region)South:tier_diff_2      0.21517    0.50613   0.425 0.670774
## r_0_50:factor(region)West:tier_diff_2            NA         NA      NA       NA
##
## (Intercept)                                ***
## r_0_50
## factor(region)Northeast                    .
## factor(region)South
## factor(region)West                         .
## same_tier                                  **
## tier_diff_1                                **
## tier_diff_2                                *
## r_0_50:factor(region)Northeast
## r_0_50:factor(region)South
## r_0_50:factor(region)West                  *
## r_0_50:same_tier                           **
## factor(region)Northeast:same_tier
## factor(region)South:same_tier
## factor(region)West:same_tier
## r_0_50:tier_diff_1                         ***
## factor(region)Northeast:tier_diff_1
## factor(region)South:tier_diff_1
## factor(region)West:tier_diff_1
```

```
## r_0_50:tier_diff_2                        *
## factor(region)Northeast:tier_diff_2
## factor(region)South:tier_diff_2
## factor(region)West:tier_diff_2
## r_0_50:factor(region)Northeast:same_tier
## r_0_50:factor(region)South:same_tier
## r_0_50:factor(region)West:same_tier
## r_0_50:factor(region)Northeast:tier_diff_1
## r_0_50:factor(region)South:tier_diff_1
## r_0_50:factor(region)West:tier_diff_1
## r_0_50:factor(region)Northeast:tier_diff_2
## r_0_50:factor(region)South:tier_diff_2
## r_0_50:factor(region)West:tier_diff_2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7596 on 3757 degrees of freedom
##   (1239 observations deleted due to missingness)
## Multiple R-squared:  0.08599,    Adjusted R-squared:  0.07943
## F-statistic: 13.09 on 27 and 3757 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90875 -0.51914 -0.05337  0.45924  2.68111
##
## Coefficients: (18 not defined because of singularities)
##                                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                                 14.7205     0.1179 124.878   <2e-16
## r_0_50                                       0.2833     0.1790   1.583   0.1137
## factor(region)Northeast                      0.1864     0.2089   0.892   0.3725
## factor(region)South                          0.5396     0.2170   2.487   0.0131
## factor(region)West                          -0.3037     0.2310  -1.315   0.1888
## same_tier                                   -0.1805     0.1907  -0.947   0.3441
## tier_diff_1                                 -0.4639     0.2002  -2.317   0.0207
## tier_diff_2                                      NA         NA      NA       NA
## r_0_50:factor(region)Northeast              -0.2889     0.3757  -0.769   0.4421
## r_0_50:factor(region)South                  -0.6659     0.2745  -2.426   0.0155
## r_0_50:factor(region)West                    0.3243     0.3472   0.934   0.3504
## r_0_50:same_tier                             0.3703     0.2987   1.240   0.2154
## factor(region)Northeast:same_tier                NA         NA      NA       NA
## factor(region)South:same_tier               -0.3470     0.2832  -1.225   0.2207
## factor(region)West:same_tier                     NA         NA      NA       NA
## r_0_50:tier_diff_1                           0.5553     0.2660   2.088   0.0371
## factor(region)Northeast:tier_diff_1              NA         NA      NA       NA
## factor(region)South:tier_diff_1                  NA         NA      NA       NA
## factor(region)West:tier_diff_1                   NA         NA      NA       NA
## r_0_50:tier_diff_2                               NA         NA      NA       NA
## factor(region)Northeast:tier_diff_2              NA         NA      NA       NA
## factor(region)South:tier_diff_2                  NA         NA      NA       NA
## factor(region)West:tier_diff_2                   NA         NA      NA       NA
```

```
## r_0_50:factor(region)Northeast:same_tier        NA         NA       NA        NA
## r_0_50:factor(region)South:same_tier        0.2821     0.3999    0.705    0.4808
## r_0_50:factor(region)West:same_tier             NA         NA       NA        NA
## r_0_50:factor(region)Northeast:tier_diff_1      NA         NA       NA        NA
## r_0_50:factor(region)South:tier_diff_1          NA         NA       NA        NA
## r_0_50:factor(region)West:tier_diff_1           NA         NA       NA        NA
## r_0_50:factor(region)Northeast:tier_diff_2      NA         NA       NA        NA
## r_0_50:factor(region)South:tier_diff_2          NA         NA       NA        NA
## r_0_50:factor(region)West:tier_diff_2           NA         NA       NA        NA
##
## (Intercept)                                ***
## r_0_50
## factor(region)Northeast
## factor(region)South                        *
## factor(region)West
## same_tier
## tier_diff_1                                *
## tier_diff_2
## r_0_50:factor(region)Northeast
## r_0_50:factor(region)South                 *
## r_0_50:factor(region)West
## r_0_50:same_tier
## factor(region)Northeast:same_tier
## factor(region)South:same_tier
## factor(region)West:same_tier
## r_0_50:tier_diff_1                         *
## factor(region)Northeast:tier_diff_1
## factor(region)South:tier_diff_1
## factor(region)West:tier_diff_1
## r_0_50:tier_diff_2
## factor(region)Northeast:tier_diff_2
## factor(region)South:tier_diff_2
## factor(region)West:tier_diff_2
## r_0_50:factor(region)Northeast:same_tier
## r_0_50:factor(region)South:same_tier
## r_0_50:factor(region)West:same_tier
## r_0_50:factor(region)Northeast:tier_diff_1
## r_0_50:factor(region)South:tier_diff_1
## r_0_50:factor(region)West:tier_diff_1
## r_0_50:factor(region)Northeast:tier_diff_2
## r_0_50:factor(region)South:tier_diff_2
## r_0_50:factor(region)West:tier_diff_2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7713 on 950 degrees of freedom
##   (4060 observations deleted due to missingness)
## Multiple R-squared:  0.04875,    Adjusted R-squared:  0.03574
## F-statistic: 3.745 on 13 and 950 DF,  p-value: 7.198e-06
```

```r
# Correlation Length Models
stargazer(
  region_mhhinc_lm_models_train,
  type = 'text'
```

```
)
```

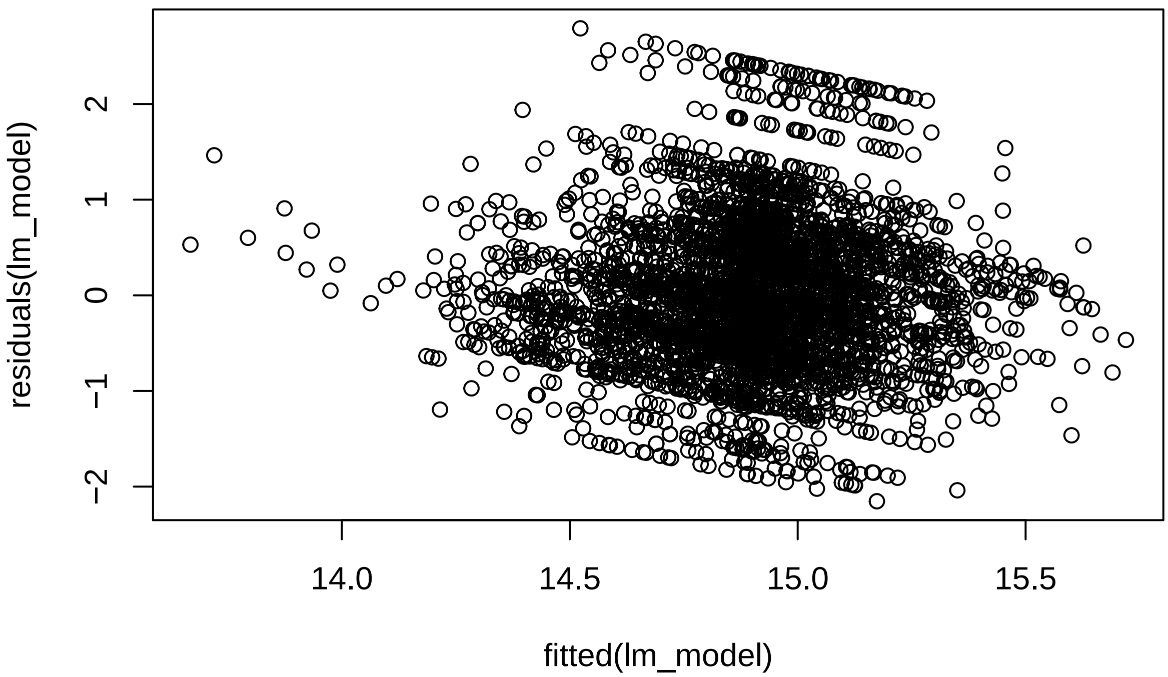
```
##
## ===============================================================================================
##                                                           Dependent variable:
##                                              -------------------------------------------------
##                                                    log(next_week_marginal_cases)
##                                                     (1)                       (2)
## -----------------------------------------------------------------------------------------------
## r_0_50                                             0.312                     0.283
##                                                   (0.213)                   (0.179)
##
## factor(region)Northeast                           0.298*                     0.186
##                                                   (0.180)                   (0.209)
##
## factor(region)South                               -0.237                    0.540**
##                                                   (0.197)                   (0.217)
##
## factor(region)West                                0.256*                    -0.304
##                                                   (0.152)                   (0.231)
##
## same_tier                                        -0.434***                  -0.181
##                                                   (0.155)                   (0.191)
##
## tier_diff_1                                      -0.549***                 -0.464**
##                                                   (0.169)                   (0.200)
##
## tier_diff_2                                      -0.350**
##                                                   (0.177)
##
## r_0_50:factor(region)Northeast                    -0.572                    -0.289
##                                                   (0.360)                   (0.376)
##
## r_0_50:factor(region)South                         0.592                   -0.666**
##                                                   (0.384)                   (0.275)
##
## r_0_50:factor(region)West                        -0.600**                    0.324
##                                                   (0.266)                   (0.347)
##
## r_0_50:same_tier                                 0.859***                    0.370
##                                                   (0.268)                   (0.299)
##
## factor(region)Northeast:same_tier                  0.207
##                                                   (0.209)
##
## factor(region)South:same_tier                      0.109                    -0.347
##                                                   (0.228)                   (0.283)
##
## factor(region)West:same_tier                       0.066
##                                                   (0.191)
##
## r_0_50:tier_diff_1                               1.060***                   0.555**
##                                                   (0.297)                   (0.266)
```

```
##
## factor(region)Northeast:tier_diff_1                  0.147
##                                                     (0.236)
##
## factor(region)South:tier_diff_1                      0.289
##                                                     (0.240)
##
## factor(region)West:tier_diff_1                       0.128
##                                                     (0.209)
##
## r_0_50:tier_diff_2                                   0.624**
##                                                     (0.310)
##
## factor(region)Northeast:tier_diff_2
##
##
## factor(region)South:tier_diff_2                     -0.026
##                                                     (0.255)
##
## factor(region)West:tier_diff_2
##
##
## r_0_50:factor(region)Northeast:same_tier            -0.517
##                                                     (0.426)
##
## r_0_50:factor(region)South:same_tier                -0.162                   0.282
##                                                     (0.455)                 (0.400)
##
## r_0_50:factor(region)West:same_tier                 -0.004
##                                                     (0.349)
##
## r_0_50:factor(region)Northeast:tier_diff_1          -0.378
##                                                     (0.467)
##
## r_0_50:factor(region)South:tier_diff_1              -0.416
##                                                     (0.478)
##
## r_0_50:factor(region)West:tier_diff_1               -0.316
##                                                     (0.374)
##
## r_0_50:factor(region)Northeast:tier_diff_2
##
##
## r_0_50:factor(region)South:tier_diff_2               0.215
##                                                     (0.506)
##
## r_0_50:factor(region)West:tier_diff_2
##
##
## Constant                                           14.796***               14.721***
##                                                     (0.132)                 (0.118)
##
## ----------------------------------------------------------------------------------------
## Observations                                         3,785                    964
```

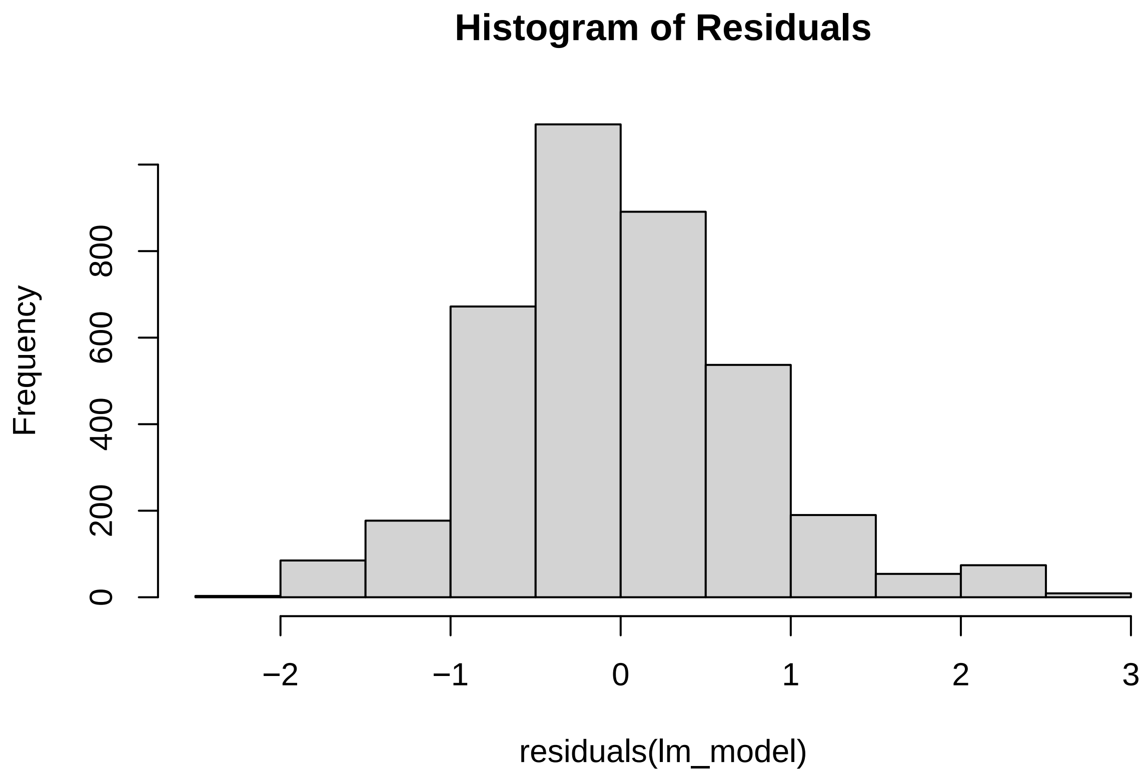
```
## R2                                                   0.086                   0.049
## Adjusted R2                                          0.079                   0.036
## Residual Std. Error                            0.760 (df = 3757)       0.771 (df = 950)
## F Statistic                           13.092*** (df = 27; 3757) 3.745*** (df = 13; 950)
## ========================================================================================
## Note:                                                        *p<0.1; **p<0.05; ***p<0.01
```

Check conditions of Normality of Residuals, Homoskedasticity,Linearity , No Perfect Multicollinearity, Independence of Residuals

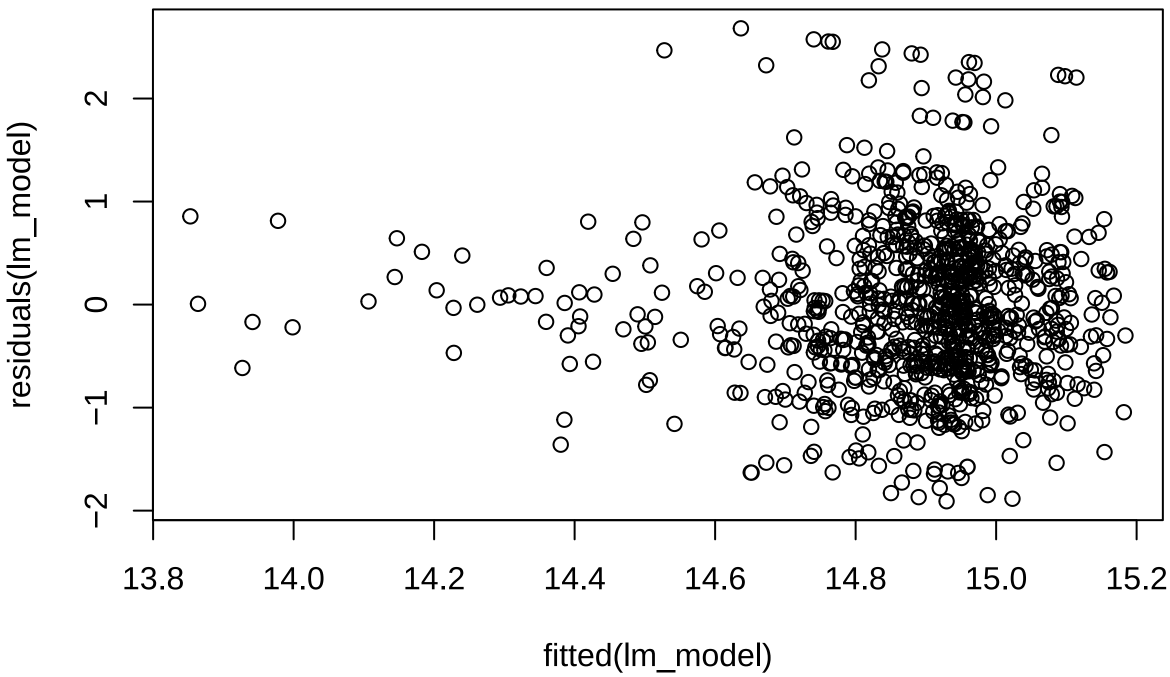
```r
for (i in 1:length(region_mhhinc_lm_models_train)) {
lm_model = region_mhhinc_lm_models_train[[i]]

# 1. Linearity Check (Visual Inspection)
plot(residuals(lm_model) ~ fitted(lm_model))

# 2. Independence Check
# Durbin-Watson test
print(durbinWatsonTest(lm_model)[3])


# 3. Homoscedasticity Check
# Breusch-Pagan test for Heteroscedasticity
print(bptest(lm_model)$p.value)

# 4. Normality of Residuals Check (Visual Inspection)
hist(residuals(lm_model), main = "Histogram of Residuals")
}
```



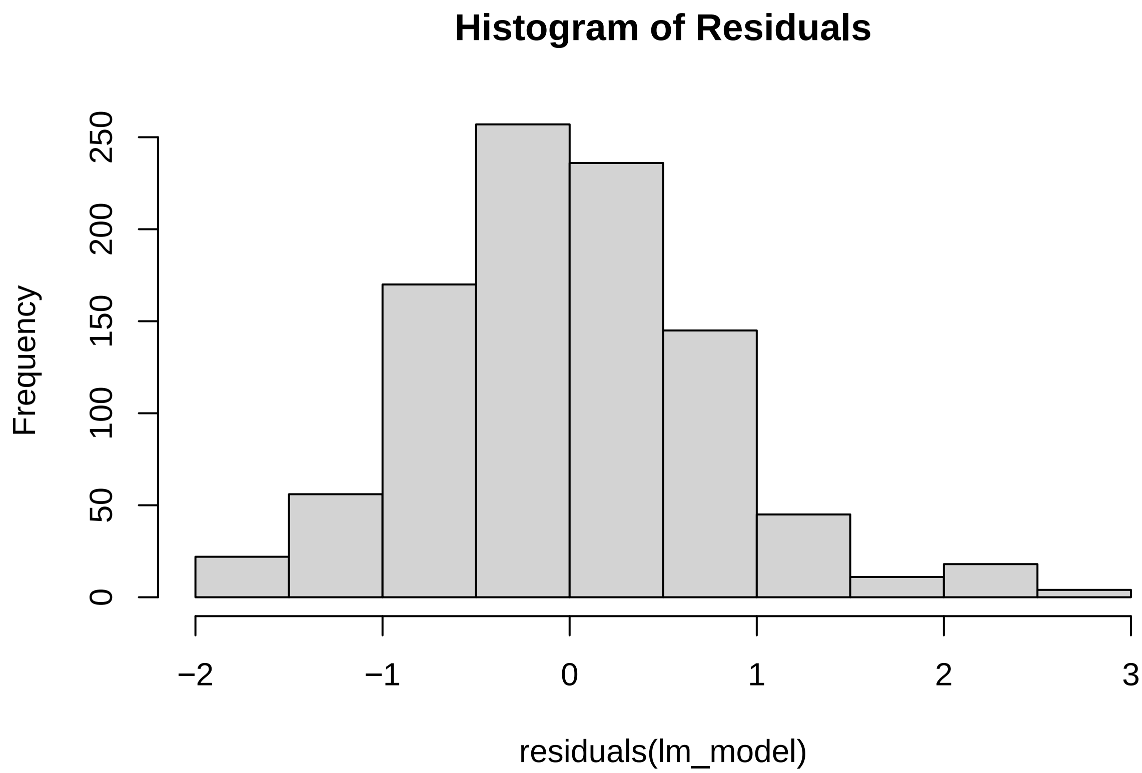

```
## $p
## [1] 0.17
##
##           BP
## 2.128747e-06
```



Histogram of Residuals
Frequency
0
200
400
600
800
−2
−1
0
1
2
3
residuals(lm_model)




```
## $p
## [1] 0.184
##
##            BP
## 5.430107e-06
```




```r
# 5. No Perfect Multicollinearity Check
# Variance Inflation Factors (VIF)
vif(lm(log(next_week_marginal_cases) ~  r_0_50 + factor(region) + factor(mhhinc_tiers) , data = all_reg
```

```
## Warning in log(next_week_marginal_cases): NaNs produced
```

```
##                          GVIF Df GVIF^(1/(2*Df))
## r_0_50               1.050392  1        1.024886
## factor(region)       1.310908  3        1.046153
## factor(mhhinc_tiers) 1.331176  9        1.016019
```

```r
# Heteroskedasticity in the urban model --> use robust standard errors
```

Functions to print significant results

```r
# Create function that prints interpretation of coefficients

print_interpretation <- function(coef, wave_val = "every wave",region_val = "whole US", tier_descr_val =
  if (!is.na(coef) && isTRUE(significant)) {

    # Prep the data frame to calculate appropriate sd
    if (wave_val !="every wave") {
```

```r
    df = filter(df, wave == as.numeric(gsub("\\D", "", wave_val))) }
    if (region_val !="whole US") {
    df = filter(df, region == region_val)}
    if (tier_descr_val != "are disregarded") {
    df = filter(df, tier_descr == tier_descr_val)}

    # Calculate sd units
    if (pred == "cor_lengths") {
      # factor_value = round(sd(df$cor_lengths, na.rm = TRUE),2)
      # constant sd units
      factor_value = round(sd(all_weekly_spatial_metrics_waves$cor_lengths, na.rm = TRUE))
      pred_type = paste("correlation length (",factor_value ," km)",sep="")
    } else {
      # factor_value = round(sd(df$r_0_50, na.rm = TRUE),2)
      factor_value = round(sd(all_weekly_spatial_metrics_waves$r_0_50, na.rm = TRUE), 2)
      pred_type = paste("local correlation (",factor_value,")",sep="")
    }

    # To Print
    cat("- For", county,"across", wave_val, "in the", region_val, "where the median household income ti
    cat(" and where mask usage in both counties", usage_descr_val, "a 1 sd increase", pred_type,"\n")
    cat(" corresponds to a ", round(coef * factor_value * 100, 2), "% increase in cases in the followin
  }
}

# Create function that returns coefficient
save_coefficient <- function(interaction) {
  if (!is.na(interaction)) {
    coef <- interaction + coefficients[2]
    return(coef)
  } else {
    return(NA)
  }
}

# Create function to calculate significance of coefficient based on t-statistic
is_significant <- function(coef, se, threshold = 2.57) {
  if (!is.na(coef) && !is.na(se)) {
    # Calculate t-statistic
    t_stat <- coef / se
    # Check if absolute value of t-statistic exceeds threshold for significance
    return(t_stat > threshold) # for 99% confidence
  } else {
    return(FALSE)  # Treat missing values as not significant
  }
}
```

Interpretations

```r
# Iterate over Regions and Tier diff dummies
for (x in 1:length(region_mhhinc_lm_models_train)) {
  lm_model <- region_mhhinc_lm_models_train[[x]]
```

```
if (x == 1) {
  county <- "all counties"
  df = all_region_mhhinc_weekly_localcor
} else {
  county <- "urban counties"
  df = urban_region_mhhinc_weekly_localcor
}

# Get stats
coefficients <- coef(lm_model)
coef_names <- names(coef(lm_model))
se <- sqrt(diag(vcovHC(lm_model)))
se_named <- rep(NA, length(coef_names))
se_named[names(se)] <- se
st_errors <- se_named[coef_names]

# Initialize values
regions <- c("Northeast", "South", "West", "Midwest")
tier_diff <- c("same_tier", "tier_diff_1", "tier_diff_2", "tier_diff_3")
tier_descr <- c("are the same", "differ by 1", "differ by 2", "differ by 3")
full_coef <- c()
significant <- c()
i <- 1

# Iterate over Regions and Tier diff dummies
for (r in 1:length(regions)) {
  region <- regions[r]
  for (m in 1:length(tier_diff)) {
    tierdiff <- tier_diff[m]
    if (region == "Midwest") {
      partial_interaction_r <- 0
      se_partial_r <- 0
    } else {
      partial_interaction_r <- coefficients[(8 + r)]
      se_partial_r <- st_errors[(8 + r)]
    }
    if (tierdiff == "tier_diff_3") {
      partial_interaction_m <- 0
      se_partial_m <- 0
    } else {
      partial_interaction_m <- coefficients[(4*(m+2))]
      se_partial_m <- st_errors[(4*(m+2))]
    }
    if (tierdiff != "tier_diff_3" & region != "Midwest") {
      full_interaction_rm <- coefficients[paste0("r_0_50:factor(region)", region, ":", tierdiff)]
      se_full_rm <- st_errors[paste0("r_0_50:factor(region)", region, ":", tierdiff)]
    } else {
      full_interaction_rm <- 0
      se_full_rm <- 0
    }

    # Calculate total standard error
    se_total <- sqrt(se_partial_r^2 + se_partial_m^2 + se_full_rm^2 + st_errors[2]^2)
```

```
      # Store coefficient information
      interaction <- partial_interaction_r + partial_interaction_m + full_interaction_rm
      full_coef[i] <- save_coefficient(interaction)

      # Print interpretation and significance
      significant[i] <- is_significant(as.numeric(full_coef[i]), as.numeric((se_total)))
      print_interpretation(full_coef[i], wave= "every wave",region, tier_descr[m], significant[i], pred
      i <- i + 1
    }
  }
}
```

```
## - For all counties across every wave in the Midwest where the median household income tiers are the s
##   and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##   corresponds to a 21.09% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the Midwest where the median household income tiers differ by
##   and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##   corresponds to a 24.7% increase in cases in the following week. Significant? TRUE
## - For urban counties across every wave in the Midwest where the median household income tiers differ
##   and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##   corresponds to a 15.1% increase in cases in the following week. Significant? TRUE
```

Regional and equality level effects are significant in the South and Midwest across every wave. Slightly stronger effects in the South than the midwest among counties of similar median household income tier. Effects are not as discernible among Urban counties. We don't have enough data to calculate correlation length but local correlation is an effective metric here.

Wave Models

Prepare Model Combinations

```
# Data sets: (1) all counties & (2) urban counties
datasets = list(all_weekly_spatial_metrics_waves, urban_weekly_spatial_metrics_waves)
df_names = c("all","urban")

# Predictor: (1) correlation length & (2) log(local correlation)
predictor = c("cor_lengths", "r_0_50")
pred_names = c("corlength","localcor")
```

Generate Models

```
wave_lm_models <- list()

for (d in 1:length(datasets)) {
  df = datasets[[d]]
  for (p in 1:length(predictor)) {
    # Create the formula by including wave_1 and wave_2 as covariates
```

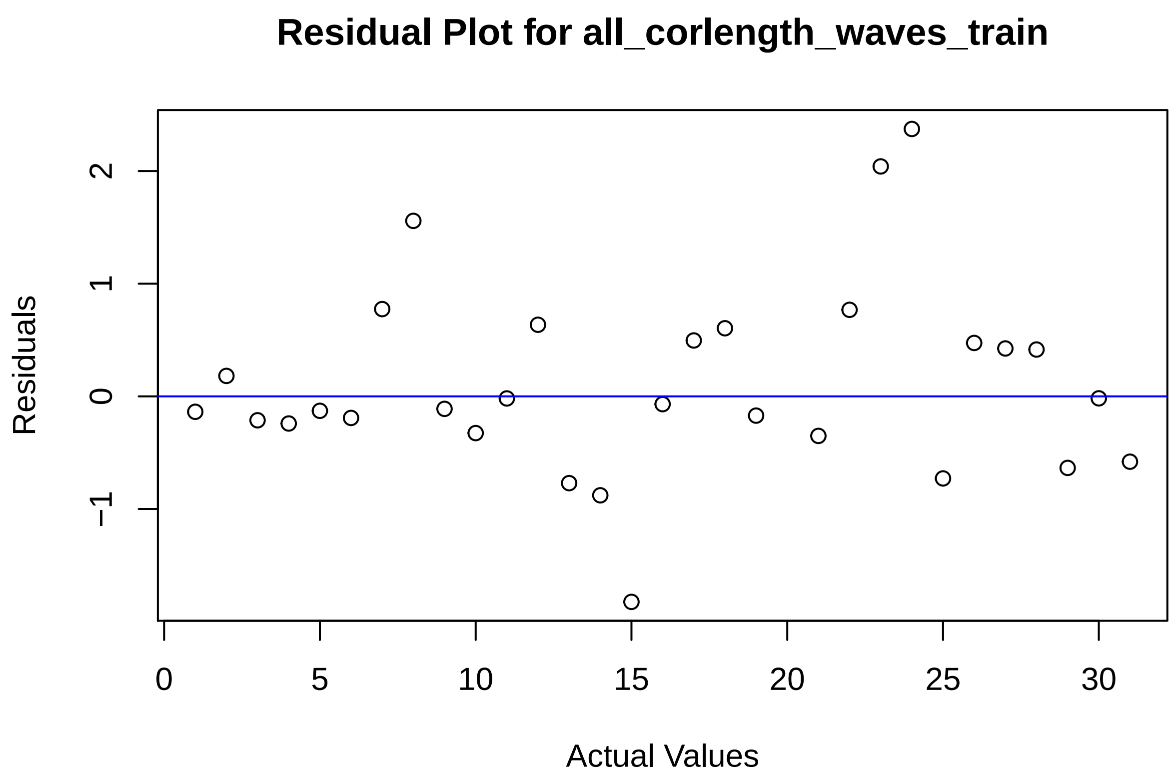
```r
    lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(predictor[p],"*factor(wave)

    # Fit the linear model
    lm_model <- lm(lm_formula, data = df)

    # Generate a name for the model object
    model_name <- paste(df_names[d], "_", pred_names[p], "_waves", sep = "")

    # Save the model to the list
    wave_lm_models[[model_name]] <- lm_model
  }
}
names(wave_lm_models)
```

```
## [1] "all_corlength_waves"   "all_localcor_waves"    "urban_corlength_waves"
## [4] "urban_localcor_waves"
```

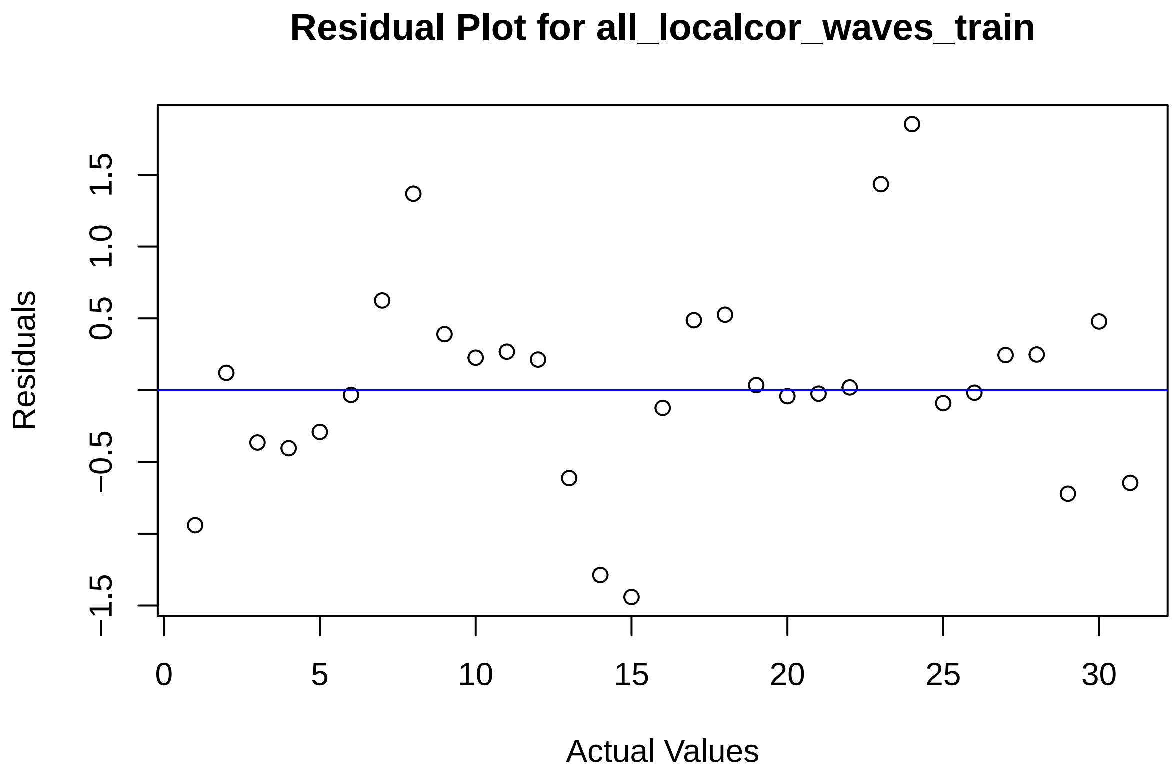
Training/Testing Models

```r
# Set seed for reproducibility
set.seed(1)

# Create an empty list to store the models
wave_lm_models_train <- list()

# Iterate over every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df <- datasets[[d]]

  # Create indices for train-test split
  n_obs <- nrow(df)
  train_indices <- sample(1:n_obs, size = round(train_ratio * n_obs), replace = FALSE)
  test_indices <- setdiff(1:n_obs, train_indices)

  # Split the data into train and test sets
  train_df <- df[train_indices, ]
  test_df <- df[test_indices, ]

  for (p in 1:length(predictor)) {
    # Create the formula
    lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(predictor[p], "*factor(wave

    # Fit linear model on the training data
    lm_model_train <- lm(lm_formula, data = train_df)

    # Generate a name for the model object
    model_name_train <- paste(df_names[d], "_", pred_names[p], "_waves_train", sep = "")

    # Save the train model to the list
    wave_lm_models_train[[model_name_train]] <- lm_model_train
```

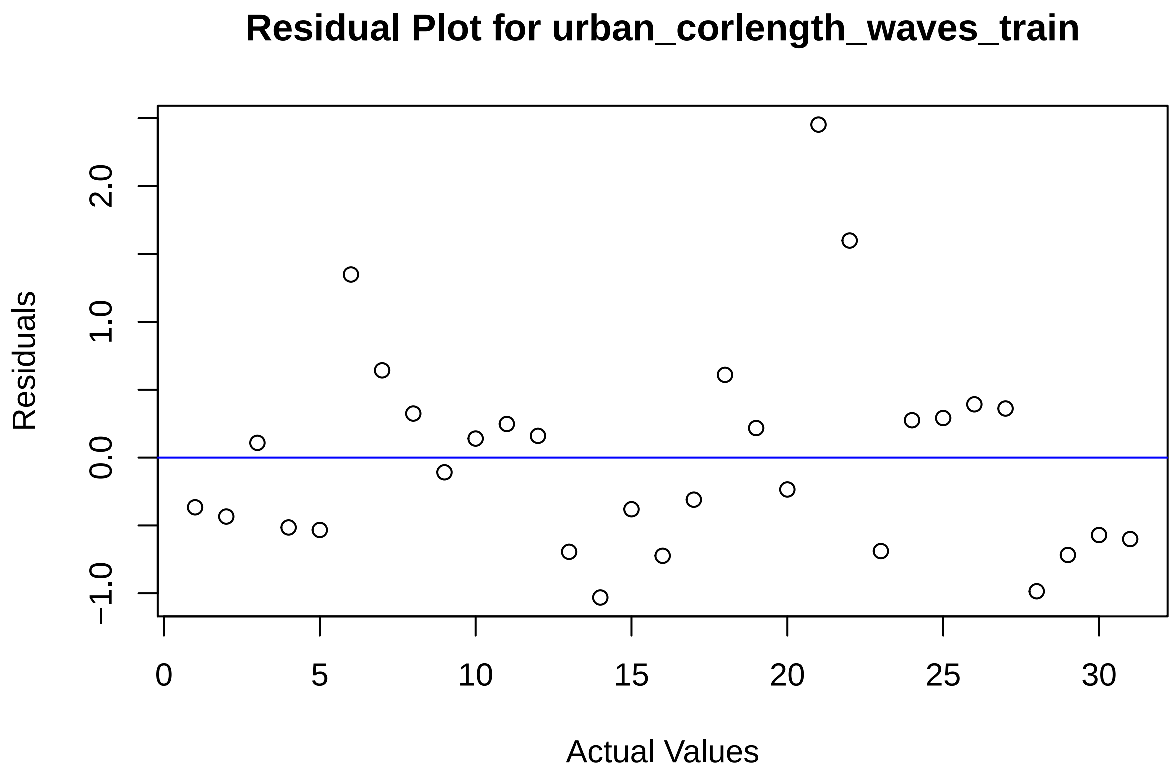
```
    # Apply the trained model to the test data and make predictions
    test_predictions <- predict(lm_model_train, newdata = test_df)

    # Calculate residuals using predictions from the test dataset
    residuals <- log(test_df$next_week_marginal_cases) - test_predictions

    # Create residual plot
    plot(1:length(residuals), residuals,
         main = paste("Residual Plot for", model_name_train),
         xlab = "Actual Values",
         ylab = "Residuals")
    abline(h = 0, col = "blue")

    # Check if the mean of residuals is less than the standard deviation of the data
    print(mean(abs(residuals), na.rm = TRUE) < sd(log(test_df$next_week_marginal_cases), na.rm = TRUE))
  }
}
```



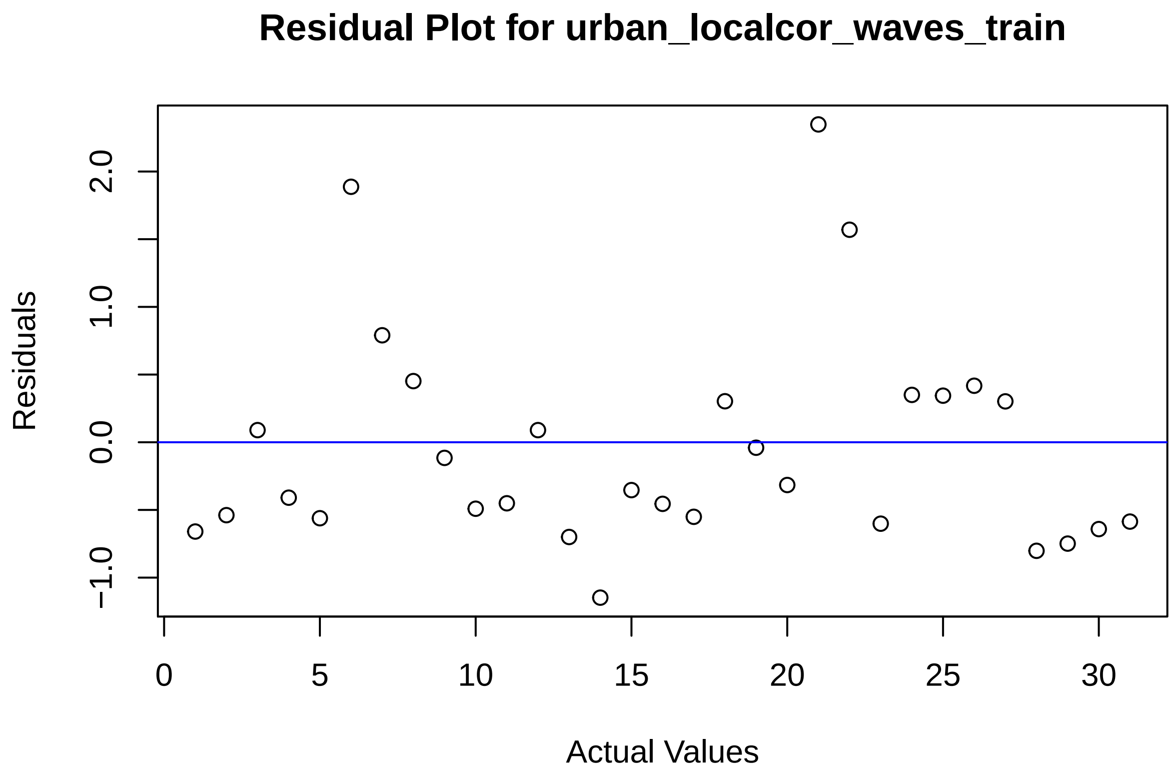

```
## [1] TRUE
```


Residual Plot for all_localcor_waves_train
Residuals
Actual Values
1.5
1.0
0.5
−0.5
−1.5
0
5
10
15
20
25
30

```
## [1] TRUE
```


Residual Plot for urban_corlength_waves_train
Residuals
−1.0
0.0
1.0
2.0
0
5
10
15
20
25
30
Actual Values

```
## [1] TRUE
```




```
## [1] TRUE
```

```r
# Print the names of the models
names(wave_lm_models_train)
```

```
## [1] "all_corlength_waves_train"   "all_localcor_waves_train"
## [3] "urban_corlength_waves_train" "urban_localcor_waves_train"
```

View Model Summaries

```r
for (m in 1:length(wave_lm_models_train)) {
  print(summary(wave_lm_models_train[[m]]))
}
```

```
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1534 -0.4779 -0.1766  0.3237  2.4271
##
## Coefficients:
```

```
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)               13.5039978  0.3554354  37.993  < 2e-16 ***
## cor_lengths                0.0016838  0.0004696   3.586 0.000501 ***
## factor(wave)2             -0.1228400  0.5839381  -0.210 0.833769
## factor(wave)3              1.3932172  0.4415257   3.155 0.002063 **
## cor_lengths:factor(wave)2  0.0005782  0.0007737   0.747 0.456498
## cor_lengths:factor(wave)3 -0.0015842  0.0006219  -2.547 0.012225 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6973 on 111 degrees of freedom
##   (9 observations deleted due to missingness)
## Multiple R-squared:  0.2152, Adjusted R-squared:  0.1799
## F-statistic: 6.088 on 5 and 111 DF,  p-value: 5.087e-05
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33926 -0.36152 -0.07113  0.36071  2.29755
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)            13.7003     0.3053  44.879  < 2e-16 ***
## r_0_50                  2.1136     0.5989   3.529 0.000595 ***
## factor(wave)2          -0.2178     0.5871  -0.371 0.711357
## factor(wave)3           0.1995     0.3940   0.506 0.613542
## r_0_50:factor(wave)2    0.3812     0.9970   0.382 0.702880
## r_0_50:factor(wave)3    0.1886     0.7803   0.242 0.809395
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6462 on 118 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.2907, Adjusted R-squared:  0.2607
## F-statistic: 9.673 on 5 and 118 DF,  p-value: 9.245e-08
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90519 -0.45514 -0.00565  0.30482  2.39813
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)               13.4807249  0.2831411  47.611  < 2e-16 ***
## cor_lengths                0.0015891  0.0004973   3.195  0.00179 **
## factor(wave)2              0.3184360  0.3933956   0.809  0.41988
## factor(wave)3              0.9123309  0.3429313   2.660  0.00889 **
## cor_lengths:factor(wave)2 -0.0001203  0.0007115  -0.169  0.86597
```

```
## cor_lengths:factor(wave)3 -0.0010925  0.0006500  -1.681  0.09545 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7269 on 118 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.1662, Adjusted R-squared:  0.1308
## F-statistic: 4.702 on 5 and 118 DF,  p-value: 0.0005949
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4674 -0.4787 -0.1331  0.3335  2.3881
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           13.0004     0.6858  18.955   <2e-16 ***
## r_0_50                 1.8351     0.9578   1.916   0.0578 .
## factor(wave)2         -0.8971     0.8939  -1.004   0.3176
## factor(wave)3          1.0560     0.8463   1.248   0.2146
## r_0_50:factor(wave)2   1.3599     1.2139   1.120   0.2649
## r_0_50:factor(wave)3  -1.0830     1.1771  -0.920   0.3594
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7184 on 118 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.1855, Adjusted R-squared:  0.151
## F-statistic: 5.376 on 5 and 118 DF,  p-value: 0.0001728
```

```r
# stargazer table for all_corlengths_region and urban_corlengths_region
# (1) is all
# (2) is urban
stargazer(
  c(wave_lm_models_train[1], wave_lm_models_train[3]),
  type = "text",
  title = "Regression of Covid Cases on Correlation Length with Waves"
)
```

```
##
## Regression of Covid Cases on Correlation Length with Waves
## =======================================================================
##                                         Dependent variable:
##                           ---------------------------------------------
##                                   log(next_week_marginal_cases)
##                                    (1)                    (2)
## -----------------------------------------------------------------------
## cor_lengths                      0.002***               0.002***
##                                  (0.0005)               (0.0005)
##
## factor(wave)2                     -0.123                  0.318
```

```
##                              (0.584)              (0.393)
##
## factor(wave)3                1.393***             0.912***
##                              (0.442)              (0.343)
##
## cor_lengths:factor(wave)2     0.001               -0.0001
##                              (0.001)              (0.001)
##
## cor_lengths:factor(wave)3    -0.002**             -0.001*
##                              (0.001)              (0.001)
##
## Constant                    13.504***            13.481***
##                              (0.355)              (0.283)
##
## ------------------------------------------------------------------------
## Observations                   117                  124
## R2                            0.215                0.166
## Adjusted R2                   0.180                0.131
## Residual Std. Error      0.697 (df = 111)       0.727 (df = 118)
## F Statistic           6.088*** (df = 5; 111) 4.702*** (df = 5; 118)
## ========================================================================
## Note:                                        *p<0.1; **p<0.05; ***p<0.01
```

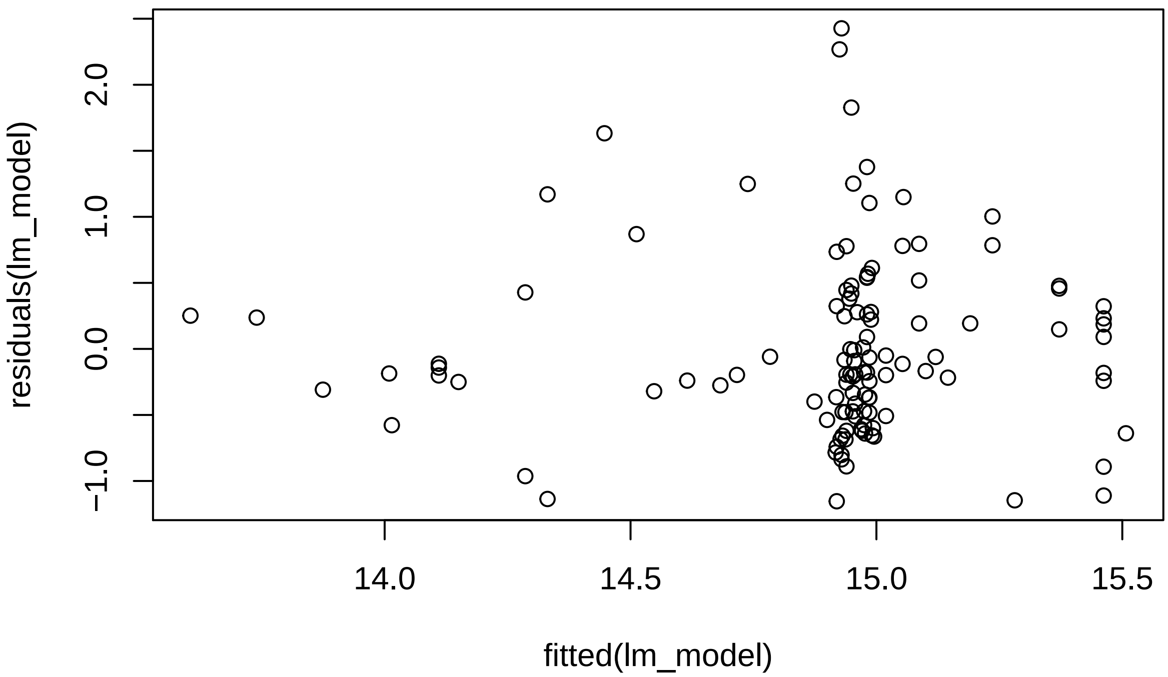
```r
# all_localcor_wave_region and urban_localcor_wave_region
# (1) is all
# (2) is urban
stargazer(
  c(wave_lm_models_train[2], wave_lm_models_train[4]),
  type = 'text',
  title = "Regression of Covid Cases on Local Correlation with Waves"
)
```

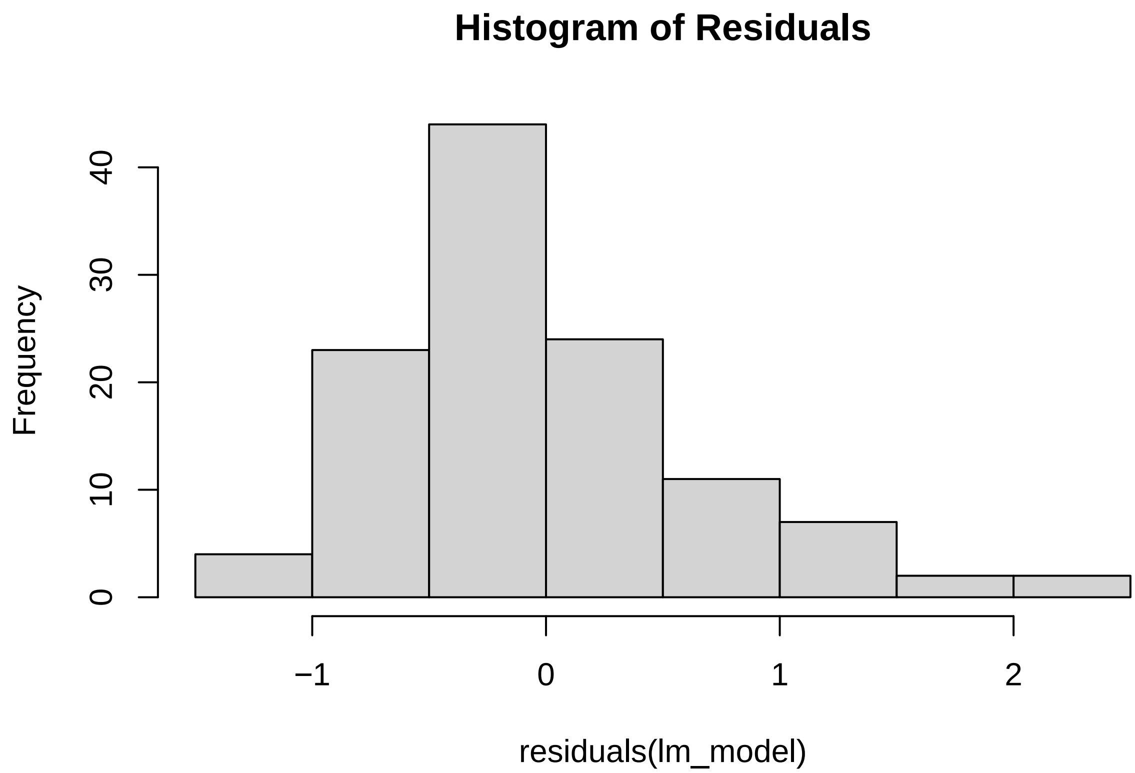
```
##
## Regression of Covid Cases on Local Correlation with Waves
## ==============================================================
##                                    Dependent variable:
##                               ------------------------------
##                               log(next_week_marginal_cases)
##                                    (1)            (2)
## --------------------------------------------------------------
## r_0_50                           2.114***        1.835*
##                                   (0.599)        (0.958)
##
## factor(wave)2                    -0.218          -0.897
##                                   (0.587)        (0.894)
##
## factor(wave)3                     0.200           1.056
##                                   (0.394)        (0.846)
##
## r_0_50:factor(wave)2              0.381           1.360
##                                   (0.997)        (1.214)
##
## r_0_50:factor(wave)3              0.189          -1.083
##                                   (0.780)        (1.177)
```

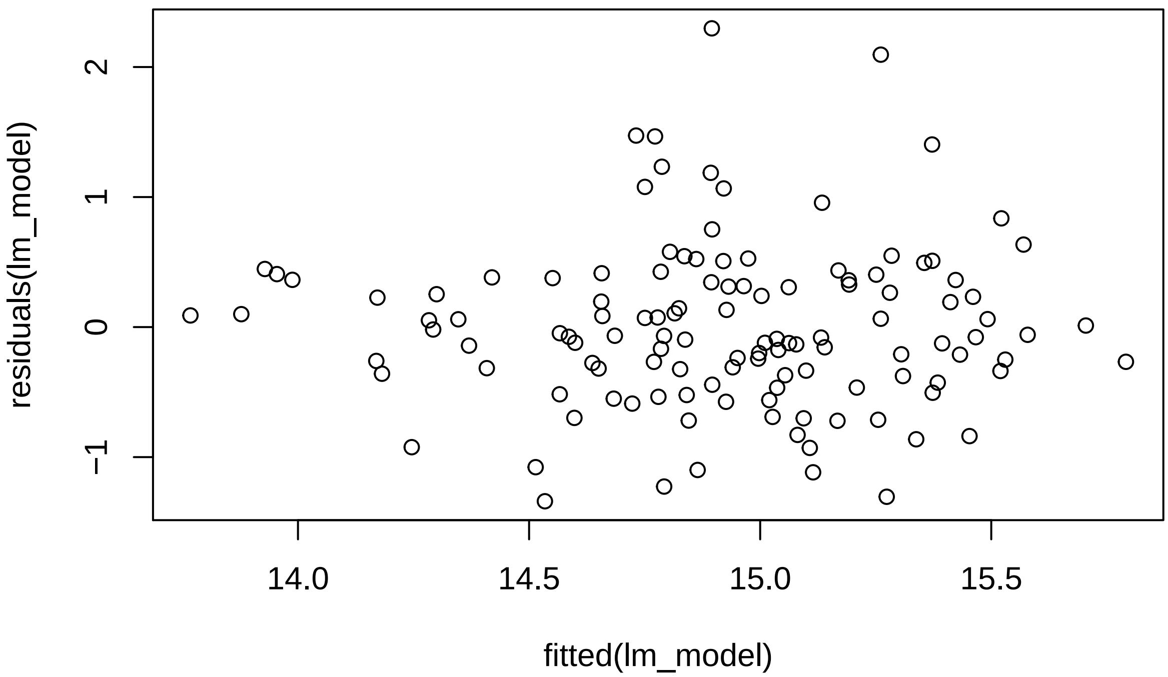
```
##
## Constant                              13.700***      13.000***
##                                        (0.305)        (0.686)
##
## ---------------------------------------------------------------
## Observations                             124            124
## R2                                      0.291          0.186
## Adjusted R2                             0.261          0.151
## Residual Std. Error (df = 118)          0.646          0.718
## F Statistic (df = 5; 118)             9.673***       5.376***
## ===============================================================
## Note:                                 *p<0.1; **p<0.05; ***p<0.01
```

Check conditions of Normality of Residuals, Homoskedasticity,Linearity , No Perfect Multicollinearity, Independence of Residuals

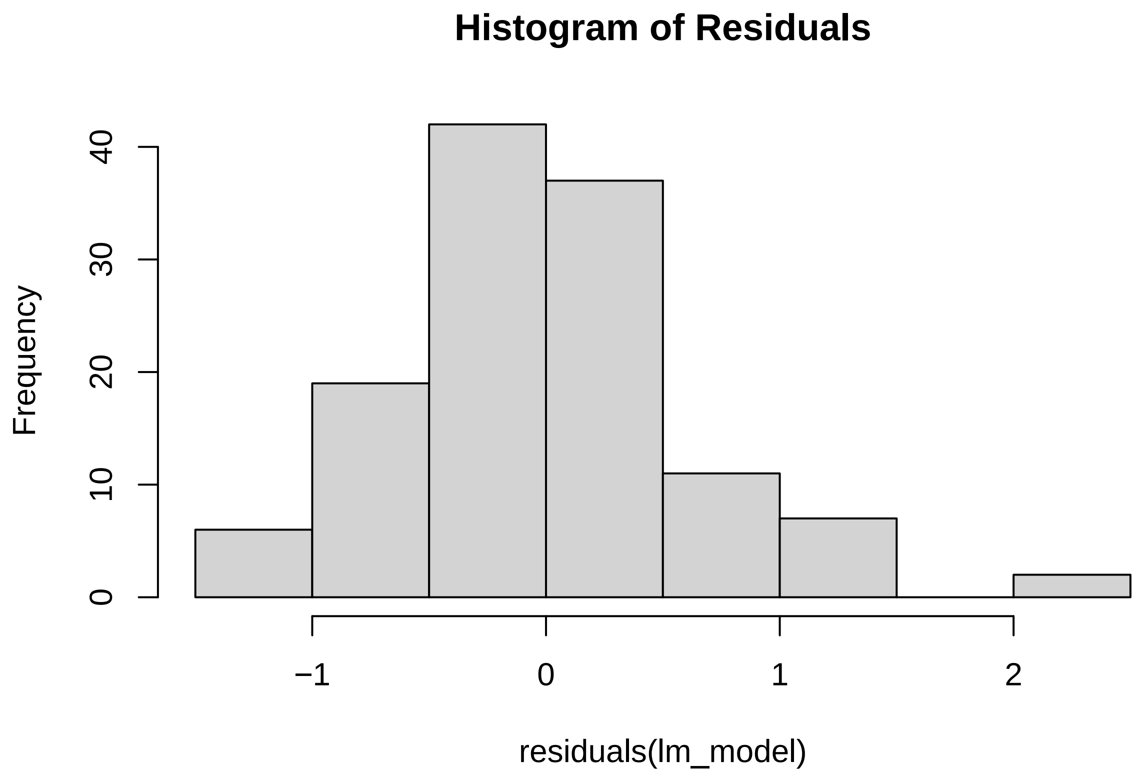
```r
for (i in 1:length(wave_lm_models_train)) {
  lm_model = wave_lm_models_train[[i]]

  # Linearity Check (Visual Inspection)
  plot(residuals(lm_model) ~ fitted(lm_model))

  # 2. Independence Check
  # Durbin-Watson test
  durbinWatsonTest(lm_model)

  # 3. Homoscedasticity Check (Visual Inspection)
  # Breusch-Pagan test for Heteroscedasticity
  bptest(lm_model)

  # 4. Normality of Residuals Check
  # Histogram of residuals
  hist(residuals(lm_model), main = "Histogram of Residuals")
}
```

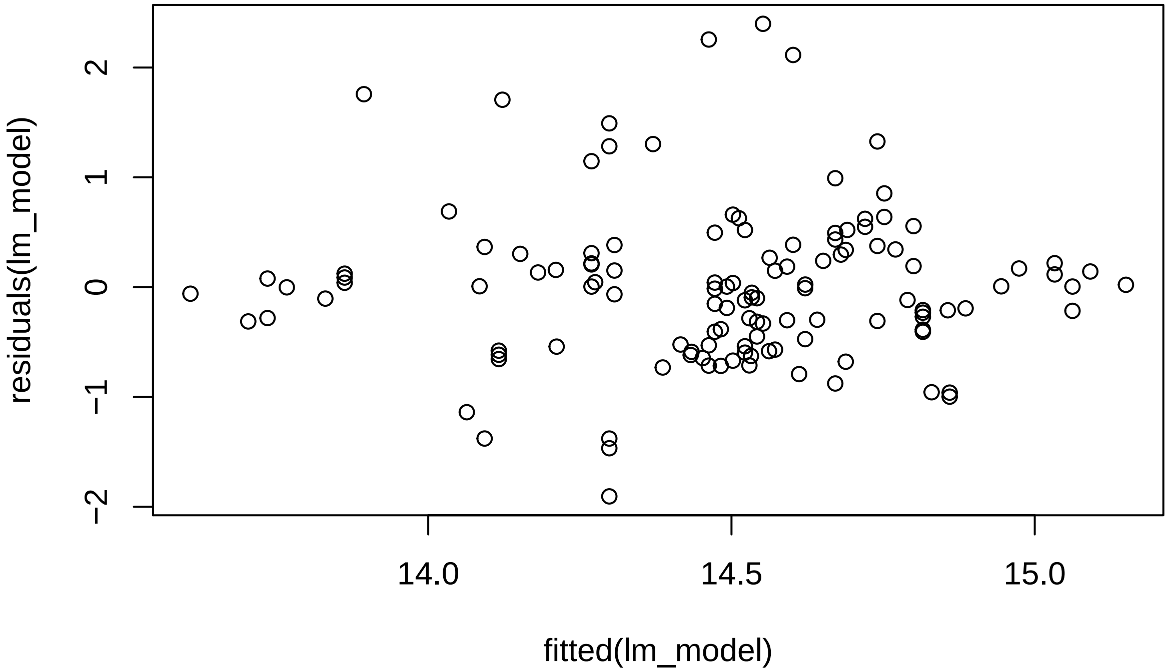

residuals(lm_model)
2.0
1.0
0.0
−1.0
14.0
14.5
15.0
15.5
fitted(lm_model)

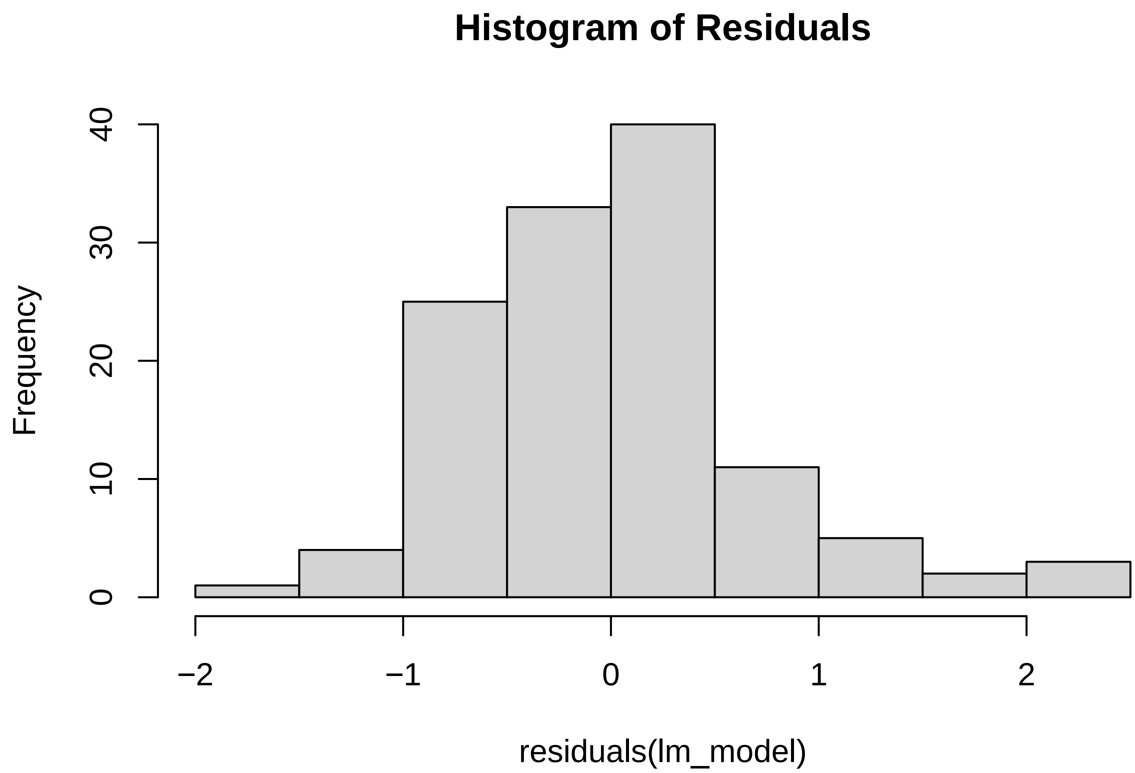

Histogram of Residuals
Frequency
0
10
20
30
40
−1
0
1
2
residuals(lm_model)

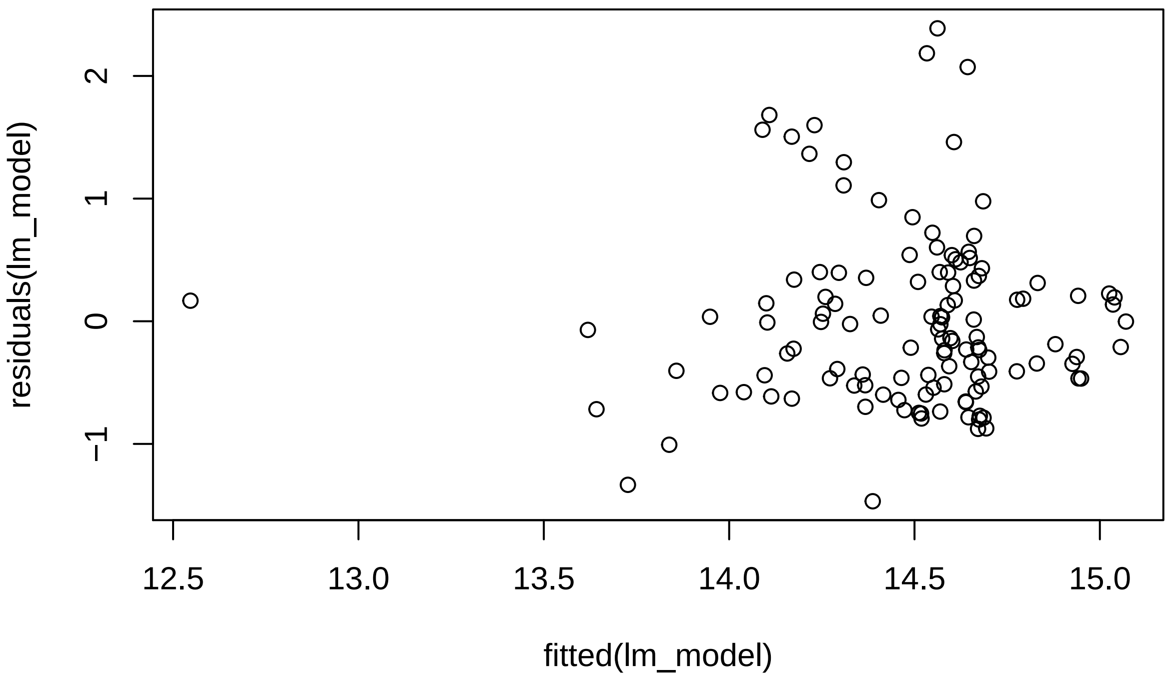


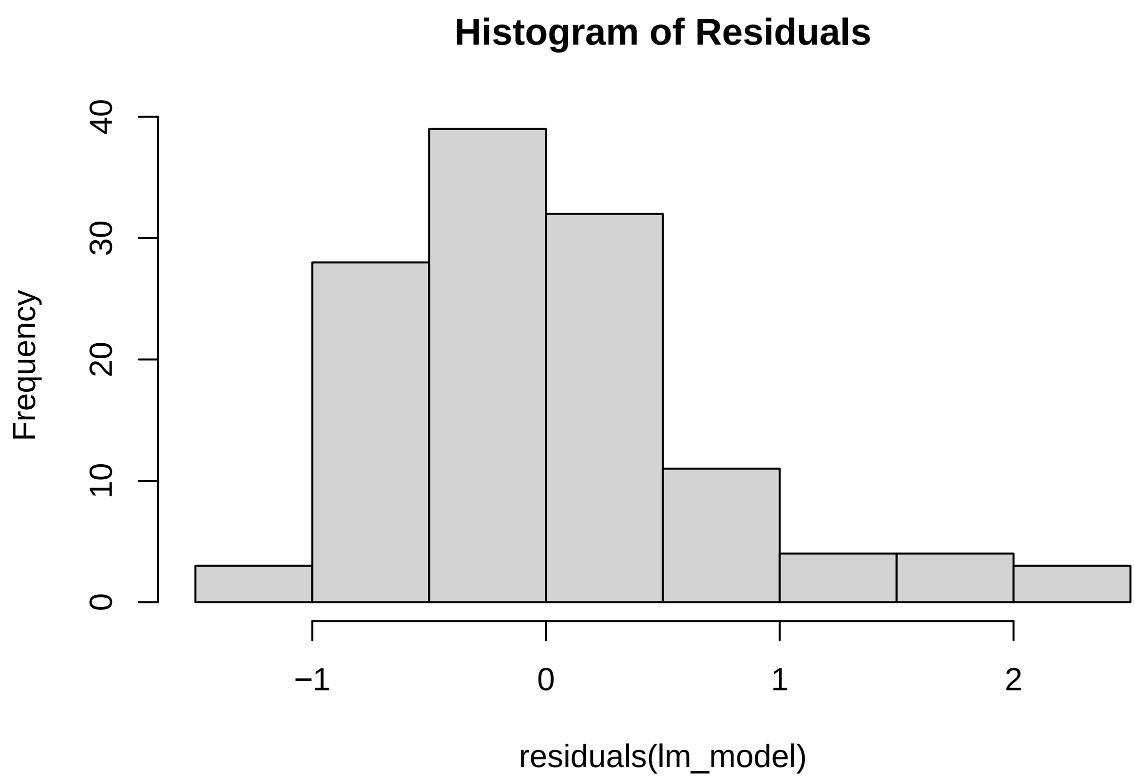
residuals(lm_model)
2
1
0
−1
14.0
14.5
15.0
15.5
fitted(lm_model)


Histogram of Residuals
Frequency
0
10
20
30
40
−1
0
1
2
residuals(lm_model)


residuals(lm_model)
2
1
0
−1
−2
14.0
14.5
15.0
fitted(lm_model)


Histogram of Residuals
Frequency
0
10
20
30
40
−2
−1
0
1
2
residuals(lm_model)


residuals(lm_model)
2
1
0
−1
12.5
13.0
13.5
14.0
14.5
15.0
fitted(lm_model)




```
# Non-linearity  in all_corlength_waves and urban_localcor_waves
```

Interpretations

Print Significant Results

```
# Lauren's

print_significant_wave_interpretations <- function(models) {
  for (i in 1:length(models)) {
    model <- models[[i]]
    coefficients <- summary(model)$coefficients

    # Print significant wave coefficients at 0.05 significance level
    for (wave in 2:3) {
      coef_name <- paste("factor(wave)", wave, sep="")
      if (coef_name %in% rownames(coefficients)) {
        coef_val <- coefficients[coef_name, "Estimate"]
        p_value <- coefficients[coef_name, "Pr(>|t|)"]

        if (p_value < 0.05) {
          percent_change <- round(coef_val * 100, 2)

          cat("- Among all", ifelse(i <= 2, "all", "urban"), "counties in wave", wave, "\n")
```

```r
          cat("a 1 unit increase in wave ", wave, " corresponds to a ",
              percent_change, "% change in cases in the following week. Significant? TRUE\n\n", sep = "
        }
      }
    }
  }
}
print_significant_wave_interpretations(wave_lm_models)
```

```
## - Among all all counties in wave 3
## a 1 unit increase in wave 3 corresponds to a 124.92% change in cases in the following week. Significa
##
## - Among all urban counties in wave 3
## a 1 unit increase in wave 3 corresponds to a 95.24% change in cases in the following week. Significan
```

```r
# Initialize values
predictor <- c("cor_lengths","r_0_50")
waves <- c("wave 1","wave 2", "wave 3")
full_coef = c()
significant = c()
i=1

# Iterate over Regions and Wave dummies
for (model in 1:length(wave_lm_models_train)) {

  # Get model stats
  lm_model = wave_lm_models_train[[model]]
  coefficients <- coef(lm_model)
  coef_names <- names(coef(lm_model))

  # Get stats
  # se <- summary(lm_model)$coefficients[, "Std. Error"] use robust SE's
  se <- sqrt(diag(vcovHC(lm_model)))
  se_named <- rep(NA, length(coef_names))
  se_named[names(se)] <- se
  st_errors <- se_named[coef_names]

  # Based on model, decide if urban or all county df
  if (model <=2) {
    county = "all counties"
    df = all_weekly_spatial_metrics_waves}
 else {
    county = "urban counties"
    df = urban_weekly_spatial_metrics_waves}

  # Based on model, choose cor_length or local_cor as predictor
  if (model %in% c(1,3)) {pred = predictor[1]}
  else {pred = predictor[2]}

  # Iterate over each wave model
  for (w in 1:length(waves)) {
    wave <- waves[w]
    if (wave != "wave 1") {
```

```
      full_interaction_w <- coefficients[paste0(pred,":factor(wave)", w)]
      se_full_w <- st_errors[paste0(pred,":factor(wave)", w)]
    } else {
      full_interaction_w <- 0
      se_full_w <- 0}
    # Calculate total standard error
    se_total <- sqrt(se_full_w^2+ st_errors[2]^2)
    # Store coefficient information
    interaction <- full_interaction_w

    # Store coefficient
    full_coef[i] <- save_coefficient(interaction)

    # Print interpretation and significance
    significant[i] <- is_significant(as.numeric(full_coef[i]), as.numeric(se_total))
    print_interpretation(full_coef[i], wave, "whole US", "are disregarded", significant[i], pred, count
    i <- i + 1
    }}
```

```
## - For all counties across wave 1 in the whole US where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 41.09% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 2 in the whole US where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 55.19% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 1 in the whole US where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 38.04% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 3 in the whole US where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 41.44% increase in cases in the following week. Significant? TRUE
## - For urban counties across wave 1 in the whole US where the median household income tiers are disre
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 38.77% increase in cases in the following week. Significant? TRUE
## - For urban counties across wave 1 in the whole US where the median household income tiers are disre
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 33.03% increase in cases in the following week. Significant? TRUE
## - For urban counties across wave 2 in the whole US where the median household income tiers are disre
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 57.51% increase in cases in the following week. Significant? TRUE
```

Region + Wave Models

Prepare Model Combinations

```
# Data sets: (1) all counties & (2) urban counties
datasets = list(all_regional_weekly_spatial_metrics, urban_regional_weekly_spatial_metrics)
df_names = c("all","urban")

# Predictor: (1) correlation length & (2) local correlation
```

```r
predictor = c("cor_lengths", "r_0_50") #unlog local cor
predictor_names = c("corlengths","localcor")

# Covariates (1) mhhinc & (2) mhhinc + wave
covariates = list(c("*factor(wave)*northeast","*factor(wave)*midwest","*factor(wave)*south"),
                  c("*northeast","*midwest","*south"))
var_names = c("wave_region","region")
```

Generate Models

```r
# Create an empty list to store the models
region_lm_models <- list()

# Iterate of every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df = datasets[d][[1]]
  for (p in 1:length(predictor)) {
    for (c in 1:length(covariates)) {
      interactions = c()
      for (i in 1:length(covariates[[c]])) {
        interactions[i] <- paste(predictor[p], covariates[[c]][i])}

      # Create the formula
      lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), col

      # Fit the linear model
      lm_model <- lm(lm_formula, data = df)

      # Generate a name for the model object
      model_name <- paste(df_names[d],"_", predictor_names[p], "_", var_names[c], sep = "")

      # Save the model to the list
      region_lm_models[[model_name]] <- lm_model
    }}}
```

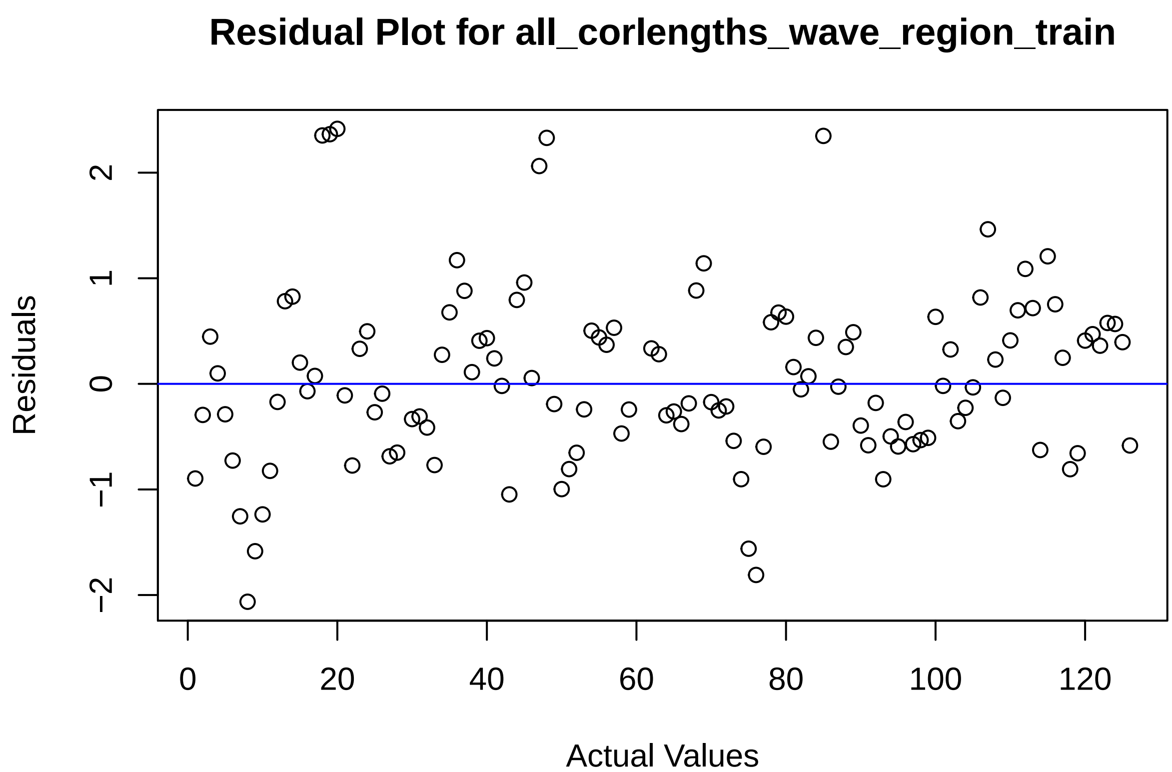
```r
print(names(region_lm_models))
```

```
## [1] "all_corlengths_wave_region"   "all_corlengths_region"
## [3] "all_localcor_wave_region"     "all_localcor_region"
## [5] "urban_corlengths_wave_region" "urban_corlengths_region"
## [7] "urban_localcor_wave_region"   "urban_localcor_region"
```

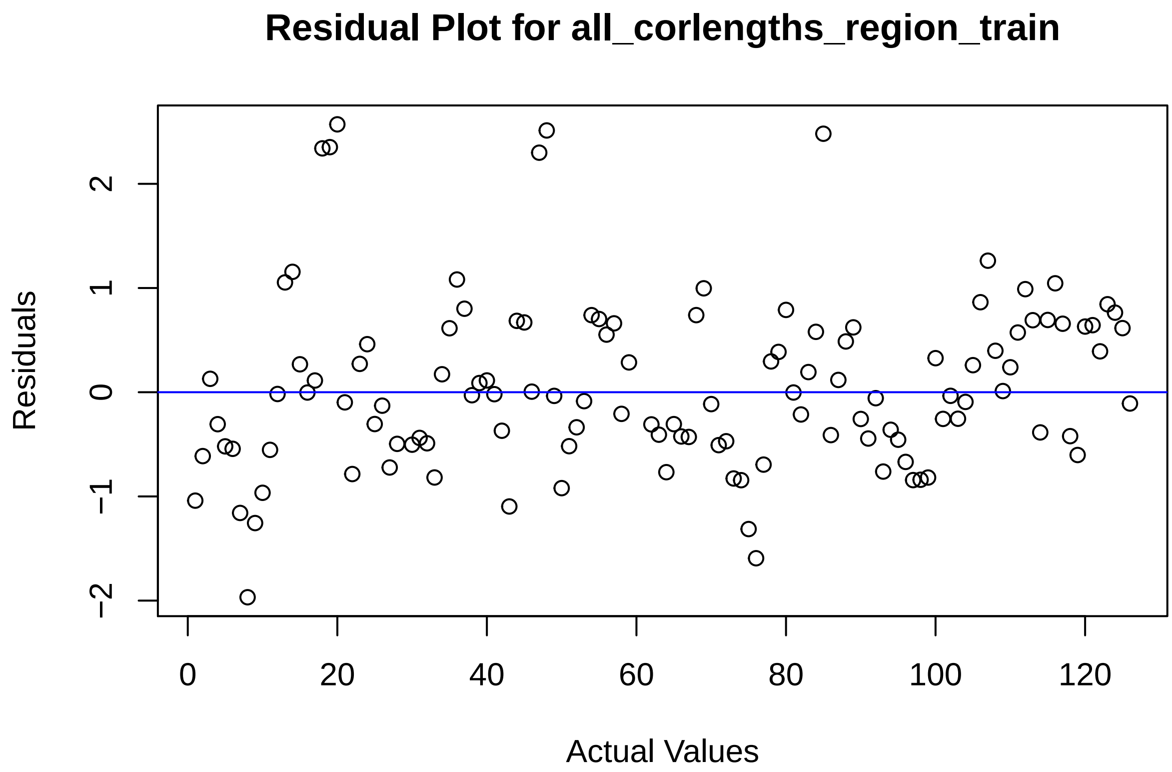
Training/Testing Models

```r
# Set seed for reproducibility
set.seed(1)

# Create an empty list to store the models
region_lm_models_train <- list()
```

```r
# Iterate over every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df <- datasets[[d]]

  # Create indices for train-test split
  n_obs <- nrow(df)
  train_indices <- sample(1:n_obs, size = round(train_ratio * n_obs), replace = FALSE)
  test_indices <- setdiff(1:n_obs, train_indices)

  # Split the data into train and test sets
  train_df <- df[train_indices, ]
  test_df <- df[test_indices, ]

  for (p in 1:length(predictor)) {
    for (c in 1:length(covariates)) {
      interactions <- c()
      for (i in 1:length(covariates[[c]])) {
        interactions[i] <- paste(predictor[p], covariates[[c]][i])
      }

      # Create the formula & fit linear model on the training data
      lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), col
      lm_model_train <- lm(lm_formula, data = train_df)

      # Name model and save to list
      model_name_train <- paste(df_names[d], "_", predictor_names[p], "_", var_names[c], "_train", sep =
      region_lm_models_train[[model_name_train]] <- lm_model_train

      # Apply the trained model to the test data and make predictions
      test_predictions <- predict(lm_model_train, newdata = test_df)

      # Calculate residuals using predictions from the test dataset
      residuals <- log(test_df$next_week_marginal_cases) - test_predictions

      # Create residual plot
      plot(1:length(residuals), residuals,
           main = paste("Residual Plot for", model_name_train),
           xlab = "Actual Values",
           ylab = "Residuals")
      abline(h = 0, col = "blue")

      # Check if the mean of residuals is less than the standard deviation of the data
      print(mean(abs(residuals), na.rm = TRUE) < sd(log(test_df$next_week_marginal_cases), na.rm = TRUE
    }
  }
}
```

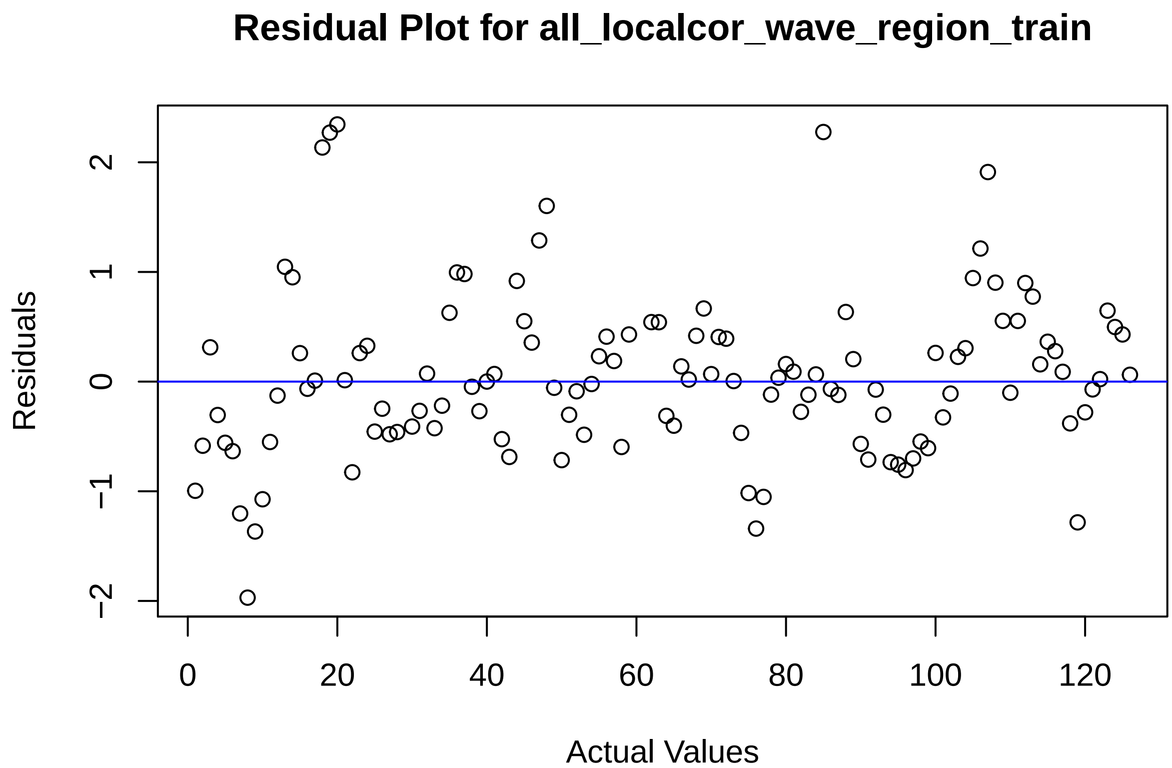

Residual Plot for all_corlengths_wave_region_train
Residuals
−2
−1
0
1
2
0
20
40
60
80
100
120
Actual Values

```
## [1] TRUE
```

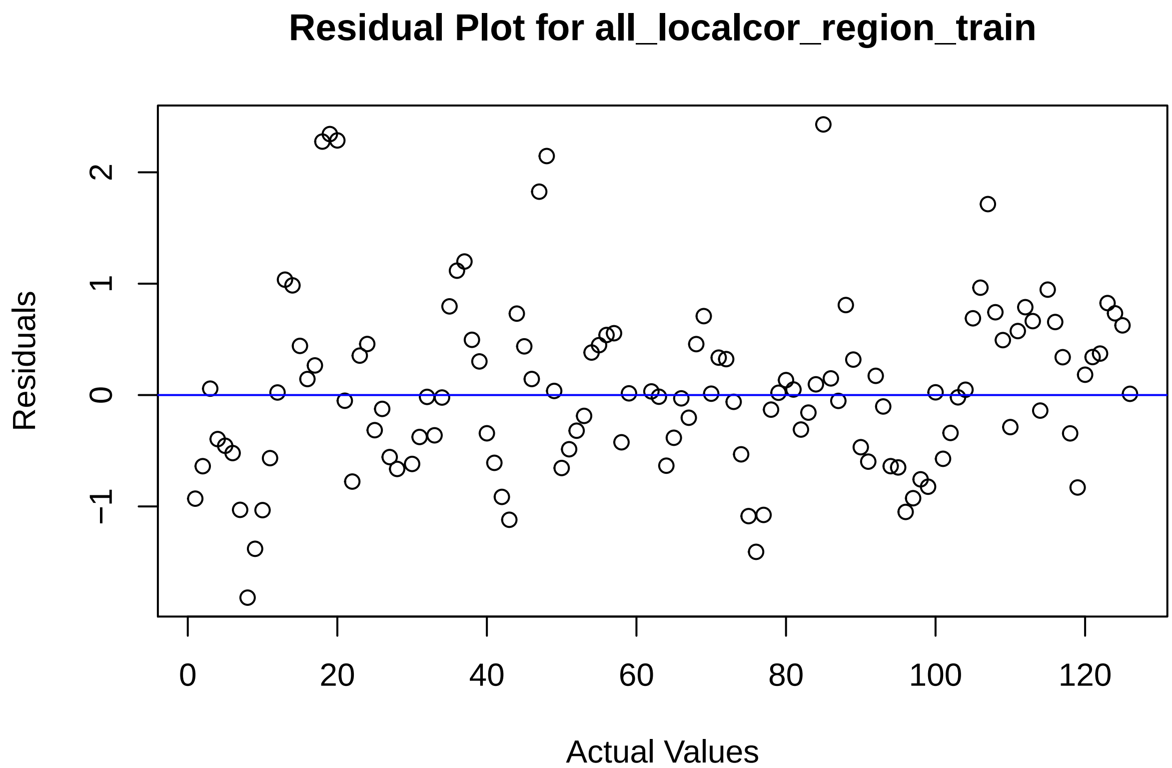


Residual Plot for all_corlengths_region_train
Residuals
Actual Values
2
1
0
−1
−2
0
20
40
60
80
100
120

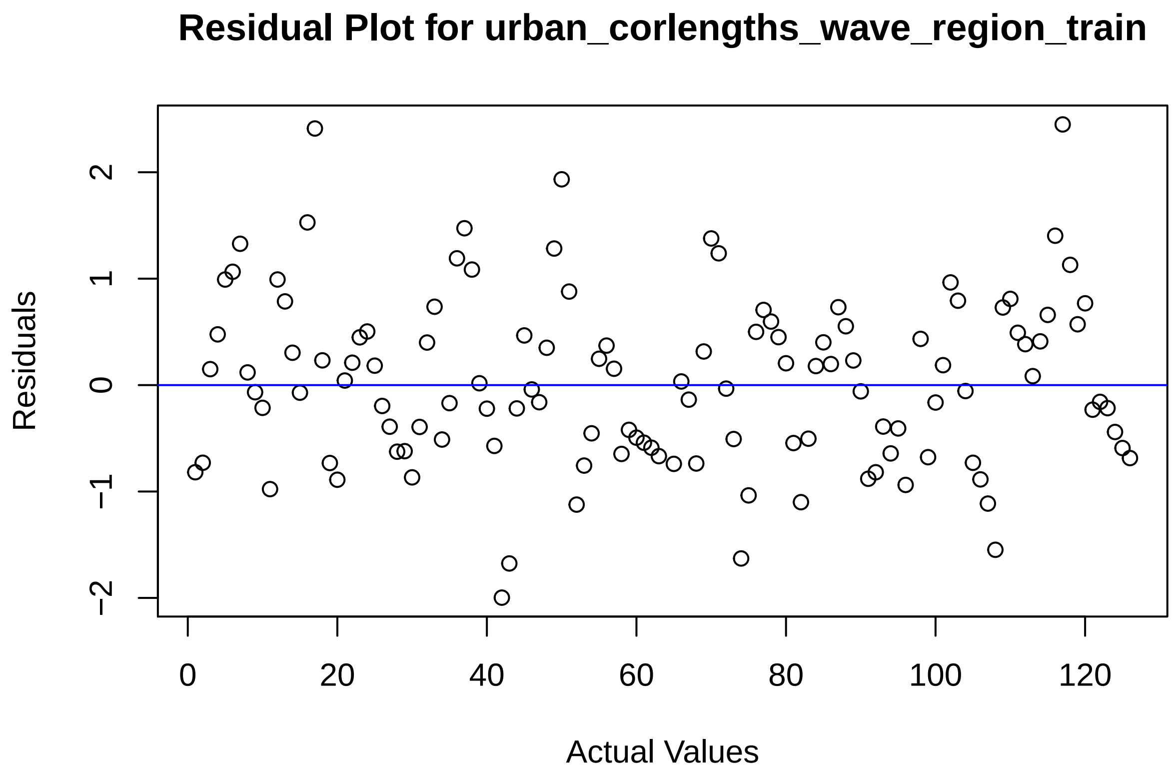

```
## [1] TRUE
```


Residual Plot for all_localcor_wave_region_train
Residuals
Actual Values
2
1
0
−1
−2
0
20
40
60
80
100
120

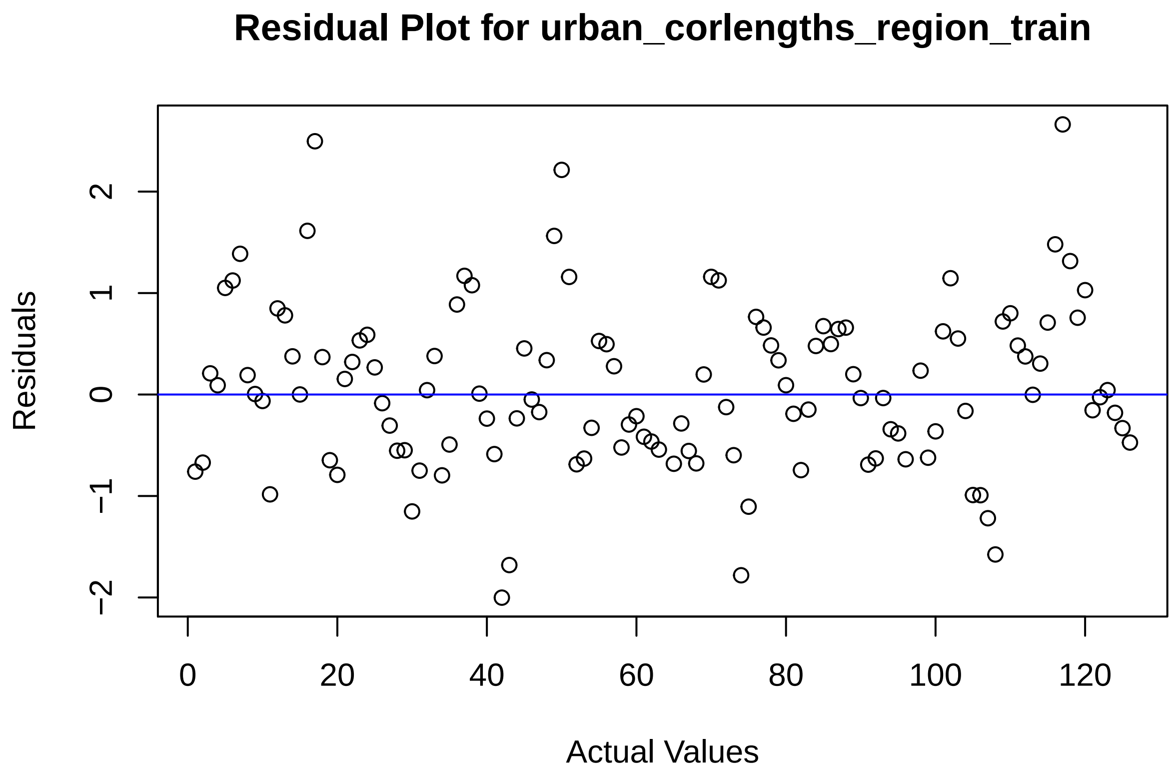
```
## [1] TRUE
```


Residual Plot for all_localcor_region_train
Residuals
2
1
0
−1
0
20
40
60
80
100
120
Actual Values

```
## [1] TRUE
```

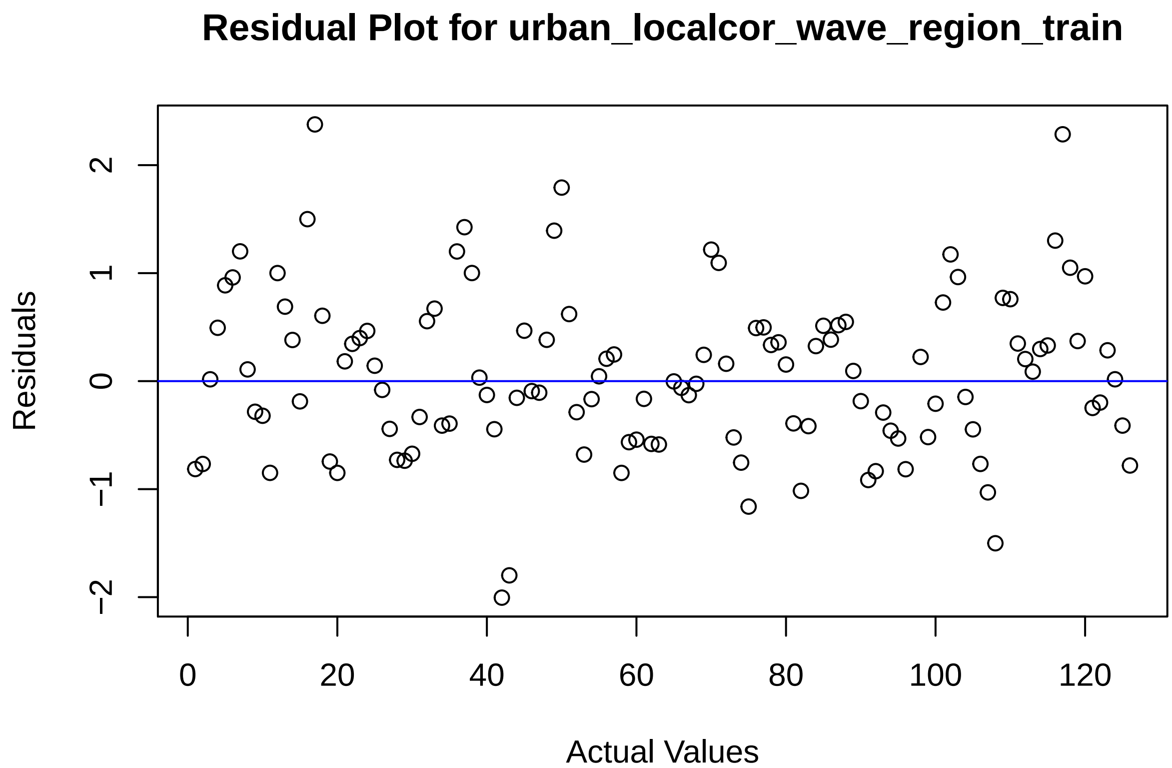

Residual Plot for urban_corlengths_wave_region_train
Residuals
Actual Values
2
1
0
−1
−2
0
20
40
60
80
100
120

```
## [1] TRUE
```

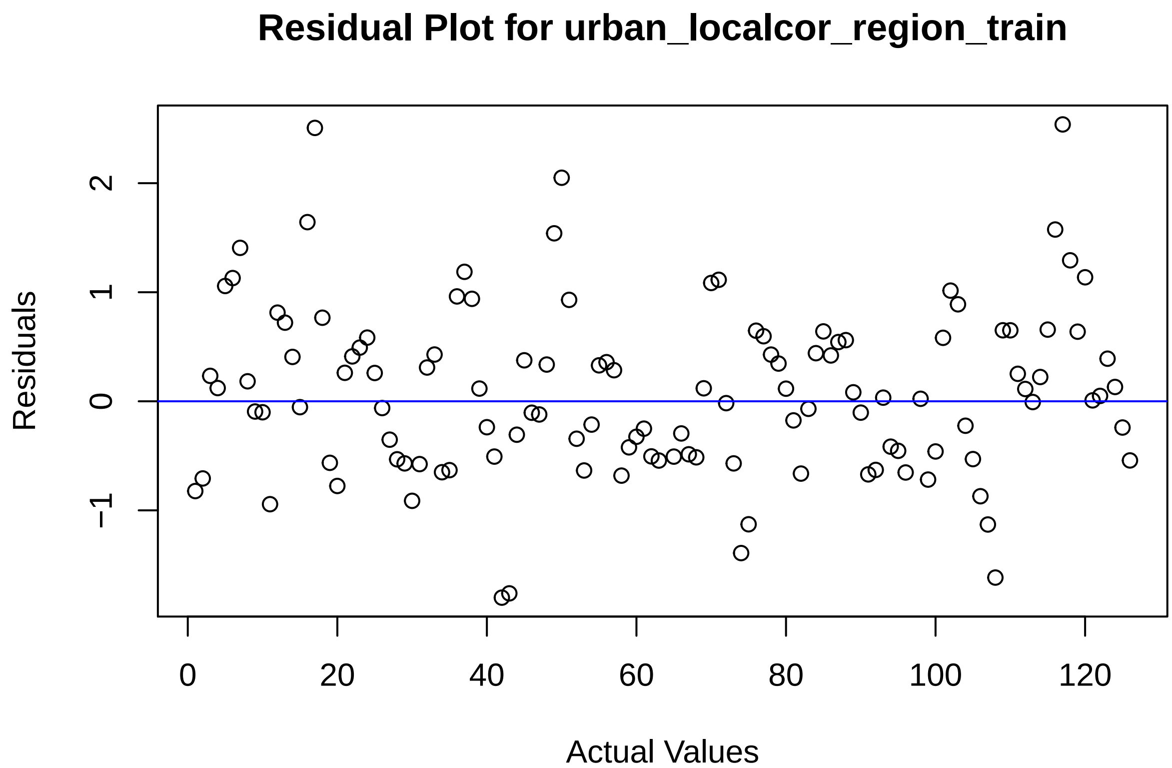


Residual Plot for urban_corlengths_region_train
Residuals
Actual Values
2
1
0
−1
−2
0
20
40
60
80
100
120


```
## [1] TRUE
```


Residual Plot for urban_localcor_wave_region_train
Residuals
2
1
0
−1
−2
0
20
40
60
80
100
120
Actual Values

```
## [1] TRUE
```




```
## [1] TRUE
```

```r
# Print the names of the models
names(region_lm_models_train)
```

```
## [1] "all_corlengths_wave_region_train"   "all_corlengths_region_train"
## [3] "all_localcor_wave_region_train"     "all_localcor_region_train"
## [5] "urban_corlengths_wave_region_train" "urban_corlengths_region_train"
## [7] "urban_localcor_wave_region_train"   "urban_localcor_region_train"
```

View Model Summaries

```r
for (m in 1:length(region_lm_models_train)){
  print(summary(region_lm_models_train[[m]]))
}
```

```
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -1.88751 -0.47825 -0.06911  0.38750  2.39876
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         13.9592886  0.2190174  63.736  < 2e-16 ***
## cor_lengths                          0.0023446  0.0006328   3.705 0.000236 ***
## factor(wave)2                        0.1917868  0.2928022   0.655 0.512784
## factor(wave)3                        0.9539634  0.2795121   3.413 0.000698 ***
## northeast                           -0.5725131  0.5005887  -1.144 0.253336
## midwest                             -0.2294504  0.3522594  -0.651 0.515126
## south                               -0.4620469  0.3931436  -1.175 0.240482
## cor_lengths:factor(wave)2           -0.0002764  0.0007761  -0.356 0.721876
## cor_lengths:factor(wave)3           -0.0020690  0.0007846  -2.637 0.008638 **
## cor_lengths:northeast                0.0047484  0.0024167   1.965 0.050017 .
## factor(wave)2:northeast              1.2714418  0.6861497   1.853 0.064505 .
## factor(wave)3:northeast              0.3149526  0.6080253   0.518 0.604706
## cor_lengths:midwest                  0.0007566  0.0010670   0.709 0.478599
## factor(wave)2:midwest               -0.7459588  0.5880014  -1.269 0.205197
## factor(wave)3:midwest               -0.1282193  0.5404653  -0.237 0.812575
## cor_lengths:south                    0.0008116  0.0010409   0.780 0.435925
## factor(wave)2:south                  1.2170878  0.5883983   2.068 0.039139 *
## factor(wave)3:south                 -0.1138728  0.5026573  -0.227 0.820878
## cor_lengths:factor(wave)2:northeast -0.0055479  0.0032528  -1.706 0.088745 .
## cor_lengths:factor(wave)3:northeast -0.0034958  0.0028435  -1.229 0.219537
## cor_lengths:factor(wave)2:midwest    0.0014449  0.0015307   0.944 0.345669
## cor_lengths:factor(wave)3:midwest    0.0004029  0.0016205   0.249 0.803775
## cor_lengths:factor(wave)2:south     -0.0025507  0.0014405  -1.771 0.077254 .
## cor_lengths:factor(wave)3:south      0.0008644  0.0013220   0.654 0.513514
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.706 on 472 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.2135, Adjusted R-squared:  0.1752
## F-statistic: 5.572 on 23 and 472 DF,  p-value: 2.32e-14
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.80298 -0.46142 -0.07188  0.37444  2.64387
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           14.4082275  0.1142494 126.112  < 2e-16 ***
## cor_lengths            0.0014612  0.0002931   4.986  8.6e-07 ***
## northeast             -0.0042796  0.2474771  -0.017   0.9862
## midwest               -0.4616378  0.2227174  -2.073   0.0387 *
## south                 -0.1913932  0.2175468  -0.880   0.3794
## cor_lengths:northeast  0.0012667  0.0011234   1.128   0.2601
## cor_lengths:midwest    0.0013057  0.0006040   2.162   0.0311 *
## cor_lengths:south      0.0004442  0.0005412   0.821   0.4122
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7277 on 488 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.1361, Adjusted R-squared:  0.1237
## F-statistic: 10.98 on 7 and 488 DF,  p-value: 6.522e-13
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.90696 -0.40419 -0.05519  0.32707  2.05223
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  13.83024    0.37793  36.594  < 2e-16 ***
## r_0_50                        1.50646    0.69599   2.165  0.03093 *
## factor(wave)2                 0.55815    0.63626   0.877  0.38080
## factor(wave)3                -0.06903    0.46223  -0.149  0.88135
## northeast                     1.71850    0.54502   3.153  0.00172 **
## midwest                      -0.03463    0.46973  -0.074  0.94126
## south                        -0.54553    0.51923  -1.051  0.29395
## r_0_50:factor(wave)2         -0.70258    1.05568  -0.666  0.50604
## r_0_50:factor(wave)3          1.05778    0.86615   1.221  0.22260
## r_0_50:northeast             -3.26127    1.04445  -3.122  0.00190 **
## factor(wave)2:northeast      -1.56530    0.86457  -1.810  0.07085 .
## factor(wave)3:northeast      -1.20302    0.69632  -1.728  0.08470 .
## r_0_50:midwest                0.94135    1.02035   0.923  0.35670
## factor(wave)2:midwest         0.92164    0.84468   1.091  0.27578
## factor(wave)3:midwest        -0.20391    0.64638  -0.315  0.75255
## r_0_50:south                  2.17406    1.13924   1.908  0.05695 .
## factor(wave)2:south          -0.09398    0.82110  -0.114  0.90893
## factor(wave)3:south           0.91395    0.62073   1.472  0.14158
## r_0_50:factor(wave)2:northeast  3.48495    1.52547   2.285  0.02278 *
## r_0_50:factor(wave)3:northeast  1.83951    1.28859   1.428  0.15408
## r_0_50:factor(wave)2:midwest   -2.29190    1.50228  -1.526  0.12777
## r_0_50:factor(wave)3:midwest   -0.26190    1.36518  -0.192  0.84795
## r_0_50:factor(wave)2:south     -0.22367    1.58497  -0.141  0.88784
## r_0_50:factor(wave)3:south     -2.46619    1.33604  -1.846  0.06553 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6919 on 473 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.2432, Adjusted R-squared:  0.2064
## F-statistic:  6.61 on 23 and 473 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9647 -0.4200 -0.0466  0.3827  2.3063
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        14.0014     0.2019  69.360  < 2e-16 ***
## r_0_50              1.6195     0.3544   4.570  6.2e-06 ***
## northeast           0.6501     0.2998   2.168   0.0306 *
## midwest             0.1768     0.2735   0.646   0.5184
## south              -0.1515     0.2621  -0.578   0.5634
## r_0_50:northeast  -1.1085     0.5248  -2.112   0.0352 *
## r_0_50:midwest    -0.1284     0.5075  -0.253   0.8004
## r_0_50:south       1.0166     0.5130   1.982   0.0481 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7184 on 489 degrees of freedom
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.1566, Adjusted R-squared:  0.1445
## F-statistic: 12.97 on 7 and 489 DF,  p-value: 2.439e-15
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.10751 -0.48267 -0.06032  0.38398  2.47157
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        13.402105   0.237340  56.468  < 2e-16 ***
## cor_lengths                         0.009336   0.002429   3.844 0.000137 ***
## factor(wave)2                       0.774666   0.354154   2.187 0.029205 *
## factor(wave)3                       1.000842   0.299389   3.343 0.000895 ***
## northeast                           0.970933   0.351303   2.764 0.005937 **
## midwest                             0.284338   0.373891   0.760 0.447345
## south                               1.279053   0.344613   3.712 0.000231 ***
## cor_lengths:factor(wave)2          -0.007311   0.002893  -2.527 0.011823 *
## cor_lengths:factor(wave)3          -0.007704   0.002933  -2.626 0.008909 **
## cor_lengths:northeast              -0.013138   0.004617  -2.845 0.004628 **
## factor(wave)2:northeast            -1.276688   0.584659  -2.184 0.029480 *
## factor(wave)3:northeast            -1.228283   0.515515  -2.383 0.017584 *
## cor_lengths:midwest                -0.002914   0.003946  -0.739 0.460538
## factor(wave)2:midwest              -0.387271   0.618111  -0.627 0.531264
## factor(wave)3:midwest              -0.960863   0.643038  -1.494 0.135777
## cor_lengths:south                  -0.014435   0.003373  -4.279 2.27e-05 ***
## factor(wave)2:south                -1.138414   0.470248  -2.421 0.015859 *
## factor(wave)3:south                -0.746843   0.432828  -1.725 0.085093 .
## cor_lengths:factor(wave)2:northeast  0.019304   0.007105   2.717 0.006835 **
## cor_lengths:factor(wave)3:northeast  0.016425   0.006024   2.727 0.006636 **
## cor_lengths:factor(wave)2:midwest    0.006165   0.006324   0.975 0.330110
## cor_lengths:factor(wave)3:midwest    0.014507   0.008204   1.768 0.077666 .
```

```
## cor_lengths:factor(wave)2:south      0.013451   0.003839   3.504 0.000503 ***
## cor_lengths:factor(wave)3:south      0.011353   0.003855   2.945 0.003390 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7419 on 472 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.1256, Adjusted R-squared:  0.08296
## F-statistic: 2.947 on 23 and 472 DF,  p-value: 7.718e-06
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.19409 -0.53631 -0.05999  0.45798  2.65702
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           14.1086840  0.1243116 113.494  < 2e-16 ***
## cor_lengths            0.0029921  0.0009741   3.072  0.00225 **
## northeast              0.0035447  0.2167662   0.016  0.98696
## midwest               -0.0416714  0.2350403  -0.177  0.85935
## south                  0.3886589  0.1673154   2.323  0.02059 *
## cor_lengths:northeast  0.0012824  0.0023871   0.537  0.59136
## cor_lengths:midwest    0.0024725  0.0026045   0.949  0.34291
## cor_lengths:south     -0.0030628  0.0011494  -2.665  0.00796 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7654 on 488 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.03771,    Adjusted R-squared:  0.0239
## F-statistic: 2.732 on 7 and 488 DF,  p-value: 0.008664
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.01057 -0.48208 -0.07242  0.39414  2.40310
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    13.2896     0.3888  34.177  < 2e-16 ***
## r_0_50                          1.3901     0.5785   2.403  0.01666 *
## factor(wave)2                   0.3164     0.6189   0.511  0.60943
## factor(wave)3                   0.6225     0.4465   1.394  0.16395
## northeast                       1.0590     0.4370   2.423  0.01575 *
## midwest                         0.3357     0.6137   0.547  0.58458
## south                           0.2269     0.5918   0.383  0.70161
## r_0_50:factor(wave)2           -0.2786     0.8294  -0.336  0.73709
```

```
## r_0_50:factor(wave)3           -0.1692     0.6871  -0.246  0.80556
## r_0_50:northeast               -2.1144     0.8098  -2.611  0.00932 **
## factor(wave)2:northeast        -0.6433     0.7004  -0.918  0.35884
## factor(wave)3:northeast        -0.4610     0.5797  -0.795  0.42688
## r_0_50:midwest                 -0.4169     0.9480  -0.440  0.66031
## factor(wave)2:midwest           0.3251     0.8563   0.380  0.70437
## factor(wave)3:midwest          -0.7446     0.8072  -0.922  0.35677
## r_0_50:south                   -0.3057     0.8796  -0.347  0.72838
## factor(wave)2:south            -0.1114     0.9027  -0.123  0.90186
## factor(wave)3:south             0.8404     0.7255   1.158  0.24732
## r_0_50:factor(wave)2:northeast   2.0503     1.1518   1.780  0.07570 .
## r_0_50:factor(wave)3:northeast   1.0333     1.0406   0.993  0.32123
## r_0_50:factor(wave)2:midwest    -0.3032     1.2289  -0.247  0.80520
## r_0_50:factor(wave)3:midwest     0.9127     1.2394   0.736  0.46186
## r_0_50:factor(wave)2:south       0.3373     1.2788   0.264  0.79207
## r_0_50:factor(wave)3:south      -1.3584     1.0796  -1.258  0.20891
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7427 on 472 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.1236, Adjusted R-squared:  0.08089
## F-statistic: 2.894 on 23 and 472 DF,  p-value: 1.121e-05
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.11687 -0.52222 -0.08028  0.44831  2.71995
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      13.81298    0.17657  78.229  < 2e-16 ***
## r_0_50            0.97123    0.25752   3.771 0.000182 ***
## northeast         0.42160    0.22632   1.863 0.063089 .
## midwest           0.05856    0.30326   0.193 0.846967
## south             0.38846    0.30290   1.282 0.200287
## r_0_50:northeast -0.50866    0.38680  -1.315 0.189114
## r_0_50:midwest   -0.01007    0.44901  -0.022 0.982114
## r_0_50:south     -0.53969    0.44023  -1.226 0.220822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7604 on 488 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.05021,    Adjusted R-squared:  0.03659
## F-statistic: 3.685 on 7 and 488 DF,  p-value: 0.0006802
```

all_localcor_region and urban_localcor_region do not have significant interaction terms, so we will not make a stargazer table for this model. Will make 3 stargazer tables, for these pairs of models: (1) all_corlengths_wave_region and urban_corlengths_wave_region (2) all_corlengths_region and urban_corlengths_region (3) all_localcor_wave_region and urban_localcor_wave_region

```r
# stargazer table for all_corlengths_wave_region and urban_corlengths_wave_region
# (1) is all
# (2) is urban
stargazer(
  c(region_lm_models_train[1], region_lm_models_train[5]),
  type = 'text',
  title = "Regression of Covid Cases on Correlation Length, Wave and Region"
)
```

```
##
## Regression of Covid Cases on Correlation Length, Wave and Region
## ==================================================================
##                                         Dependent variable:
##                                    -----------------------------
##                                    log(next_week_marginal_cases)
##                                          (1)            (2)
## ------------------------------------------------------------------
## cor_lengths                            0.002***       0.009***
##                                        (0.001)        (0.002)
##
## factor(wave)2                           0.192         0.775**
##                                        (0.293)        (0.354)
##
## factor(wave)3                          0.954***       1.001***
##                                        (0.280)        (0.299)
##
## northeast                               -0.573        0.971***
##                                        (0.501)        (0.351)
##
## midwest                                 -0.229         0.284
##                                        (0.352)        (0.374)
##
## south                                   -0.462        1.279***
##                                        (0.393)        (0.345)
##
## cor_lengths:factor(wave)2              -0.0003        -0.007**
##                                        (0.001)        (0.003)
##
## cor_lengths:factor(wave)3             -0.002***      -0.008***
##                                        (0.001)        (0.003)
##
## cor_lengths:northeast                   0.005*       -0.013***
##                                        (0.002)        (0.005)
##
## factor(wave)2:northeast                 1.271*        -1.277**
##                                        (0.686)        (0.585)
##
## factor(wave)3:northeast                 0.315         -1.228**
##                                        (0.608)        (0.516)
##
## cor_lengths:midwest                     0.001          -0.003
##                                        (0.001)        (0.004)
##
```

```
## factor(wave)2:midwest                    -0.746          -0.387
##                                          (0.588)        (0.618)
##
## factor(wave)3:midwest                    -0.128          -0.961
##                                          (0.540)        (0.643)
##
## cor_lengths:south                         0.001         -0.014***
##                                          (0.001)        (0.003)
##
## factor(wave)2:south                       1.217**        -1.138**
##                                          (0.588)        (0.470)
##
## factor(wave)3:south                      -0.114          -0.747*
##                                          (0.503)        (0.433)
##
## cor_lengths:factor(wave)2:northeast      -0.006*         0.019***
##                                          (0.003)        (0.007)
##
## cor_lengths:factor(wave)3:northeast      -0.003          0.016***
##                                          (0.003)        (0.006)
##
## cor_lengths:factor(wave)2:midwest         0.001           0.006
##                                          (0.002)        (0.006)
##
## cor_lengths:factor(wave)3:midwest        0.0004           0.015*
##                                          (0.002)        (0.008)
##
## cor_lengths:factor(wave)2:south          -0.003*         0.013***
##                                          (0.001)        (0.004)
##
## cor_lengths:factor(wave)3:south           0.001          0.011***
##                                          (0.001)        (0.004)
##
## Constant                                13.959***      13.402***
##                                          (0.219)        (0.237)
##
## -------------------------------------------------------------------
## Observations                               496            496
## R2                                        0.214          0.126
## Adjusted R2                               0.175          0.083
## Residual Std. Error (df = 472)            0.706          0.742
## F Statistic (df = 23; 472)              5.572***       2.947***
## ===================================================================
## Note:                                   *p<0.1; **p<0.05; ***p<0.01
```

```r
# stargazer table for all_corlengths_region and urban_corlengths_region
# (1) is all
# (2) is urban
stargazer(
  c(region_lm_models_train[2], region_lm_models_train[6]),
  type = 'text',
  title = "Regression of Covid Cases on Correlation Length and Region"
)
```

```
##
## Regression of Covid Cases on Correlation Length and Region
## ==============================================================
##                                    Dependent variable:
##                                ------------------------------
##                                log(next_week_marginal_cases)
##                                     (1)            (2)
## --------------------------------------------------------------
## cor_lengths                       0.001***       0.003***
##                                  (0.0003)        (0.001)
##
## northeast                         -0.004          0.004
##                                   (0.247)        (0.217)
##
## midwest                          -0.462**        -0.042
##                                   (0.223)        (0.235)
##
## south                             -0.191         0.389**
##                                   (0.218)        (0.167)
##
## cor_lengths:northeast              0.001          0.001
##                                   (0.001)        (0.002)
##
## cor_lengths:midwest               0.001**         0.002
##                                   (0.001)        (0.003)
##
## cor_lengths:south                 0.0004        -0.003***
##                                   (0.001)        (0.001)
##
## Constant                         14.408***      14.109***
##                                   (0.114)        (0.124)
##
## --------------------------------------------------------------
## Observations                        496            496
## R2                                 0.136          0.038
## Adjusted R2                        0.124          0.024
## Residual Std. Error (df = 488)     0.728          0.765
## F Statistic (df = 7; 488)        10.983***      2.732***
## ==============================================================
## Note:                            *p<0.1; **p<0.05; ***p<0.01
```

```r
# all_localcor_wave_region and urban_localcor_wave_region
# (1) is all
# (2) is urban
stargazer(
  c(region_lm_models_train[3], region_lm_models_train[7]),
  type = 'text',
  title = "Regression of Covid Cases on Local Correlation, Wave and Region"
)
```

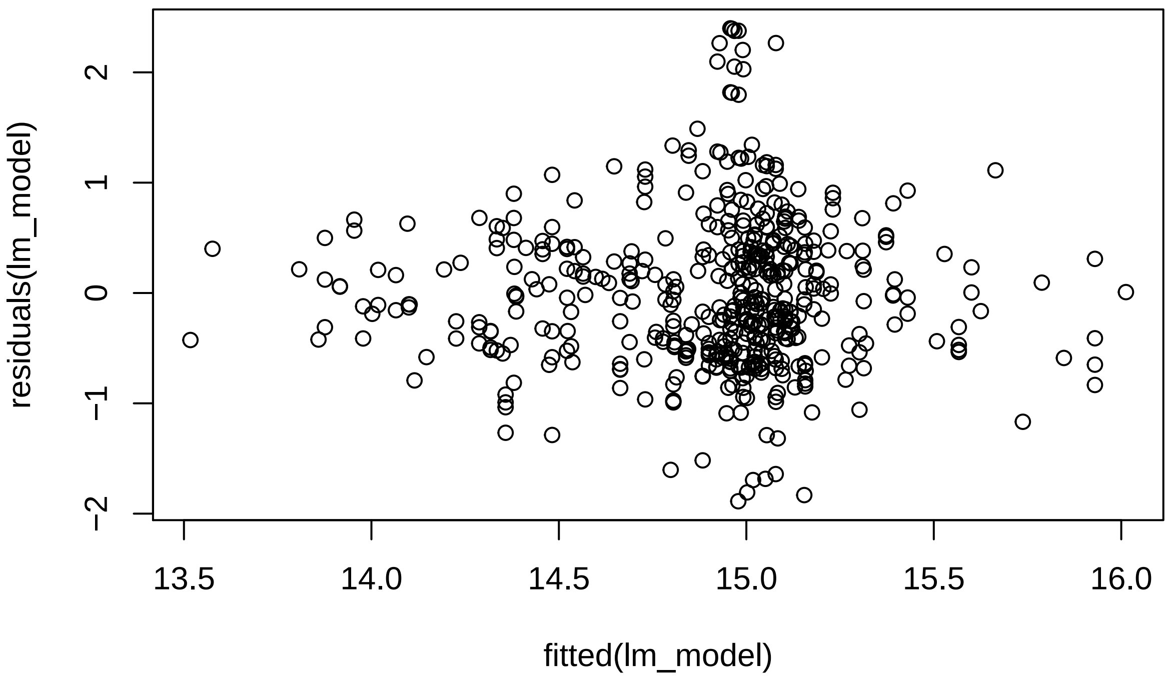
```
##
## Regression of Covid Cases on Local Correlation, Wave and Region
## ===============================================================================
##                                                  Dependent variable:
```

```
##                                ----------------------------------------------
##                                          log(next_week_marginal_cases)
##                                           (1)                     (2)
## ----------------------------------------------------------------------------
## r_0_50                                  1.506**                 1.390**
##                                         (0.696)                 (0.579)
##
## factor(wave)2                            0.558                   0.316
##                                         (0.636)                 (0.619)
##
## factor(wave)3                           -0.069                   0.622
##                                         (0.462)                 (0.447)
##
## northeast                              1.719***                 1.059**
##                                         (0.545)                 (0.437)
##
## midwest                                 -0.035                   0.336
##                                         (0.470)                 (0.614)
##
## south                                   -0.546                   0.227
##                                         (0.519)                 (0.592)
##
## r_0_50:factor(wave)2                    -0.703                  -0.279
##                                         (1.056)                 (0.829)
##
## r_0_50:factor(wave)3                     1.058                  -0.169
##                                         (0.866)                 (0.687)
##
## r_0_50:northeast                       -3.261***               -2.114***
##                                         (1.044)                 (0.810)
##
## factor(wave)2:northeast                 -1.565*                 -0.643
##                                         (0.865)                 (0.700)
##
## factor(wave)3:northeast                 -1.203*                 -0.461
##                                         (0.696)                 (0.580)
##
## r_0_50:midwest                           0.941                  -0.417
##                                         (1.020)                 (0.948)
##
## factor(wave)2:midwest                    0.922                   0.325
##                                         (0.845)                 (0.856)
##
## factor(wave)3:midwest                   -0.204                  -0.745
##                                         (0.646)                 (0.807)
##
## r_0_50:south                            2.174*                  -0.306
##                                         (1.139)                 (0.880)
##
## factor(wave)2:south                     -0.094                  -0.111
##                                         (0.821)                 (0.903)
##
## factor(wave)3:south                      0.914                   0.840
##                                         (0.621)                 (0.726)
```

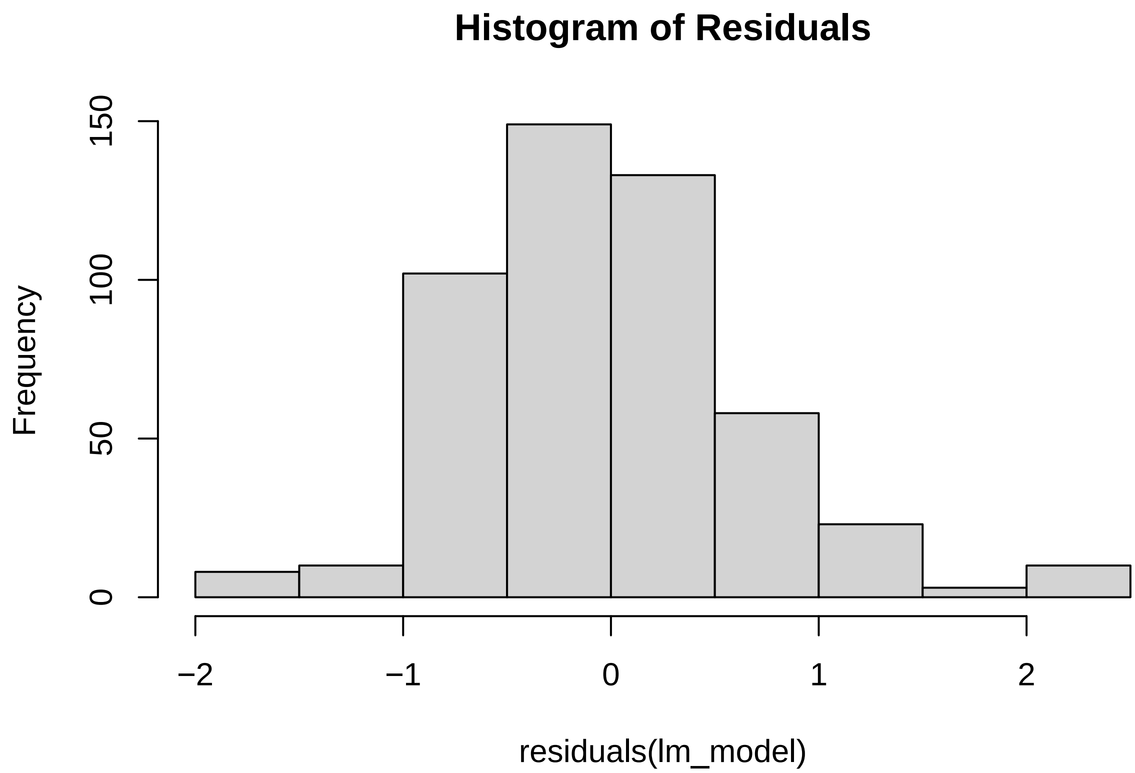
```
##
## r_0_50:factor(wave)2:northeast         3.485**                 2.050*
##                                        (1.525)                 (1.152)
##
## r_0_50:factor(wave)3:northeast          1.840                   1.033
##                                        (1.289)                 (1.041)
##
## r_0_50:factor(wave)2:midwest           -2.292                  -0.303
##                                        (1.502)                 (1.229)
##
## r_0_50:factor(wave)3:midwest           -0.262                   0.913
##                                        (1.365)                 (1.239)
##
## r_0_50:factor(wave)2:south             -0.224                   0.337
##                                        (1.585)                 (1.279)
##
## r_0_50:factor(wave)3:south             -2.466*                 -1.358
##                                        (1.336)                 (1.080)
##
## Constant                              13.830***               13.290***
##                                        (0.378)                 (0.389)
##
## ----------------------------------------------------------------------------
## Observations                             497                     496
## R2                                      0.243                   0.124
## Adjusted R2                             0.206                   0.081
## Residual Std. Error                0.692 (df = 473)        0.743 (df = 472)
## F Statistic                     6.610*** (df = 23; 473) 2.894*** (df = 23; 472)
## ============================================================================
## Note:                                            *p<0.1; **p<0.05; ***p<0.01
```

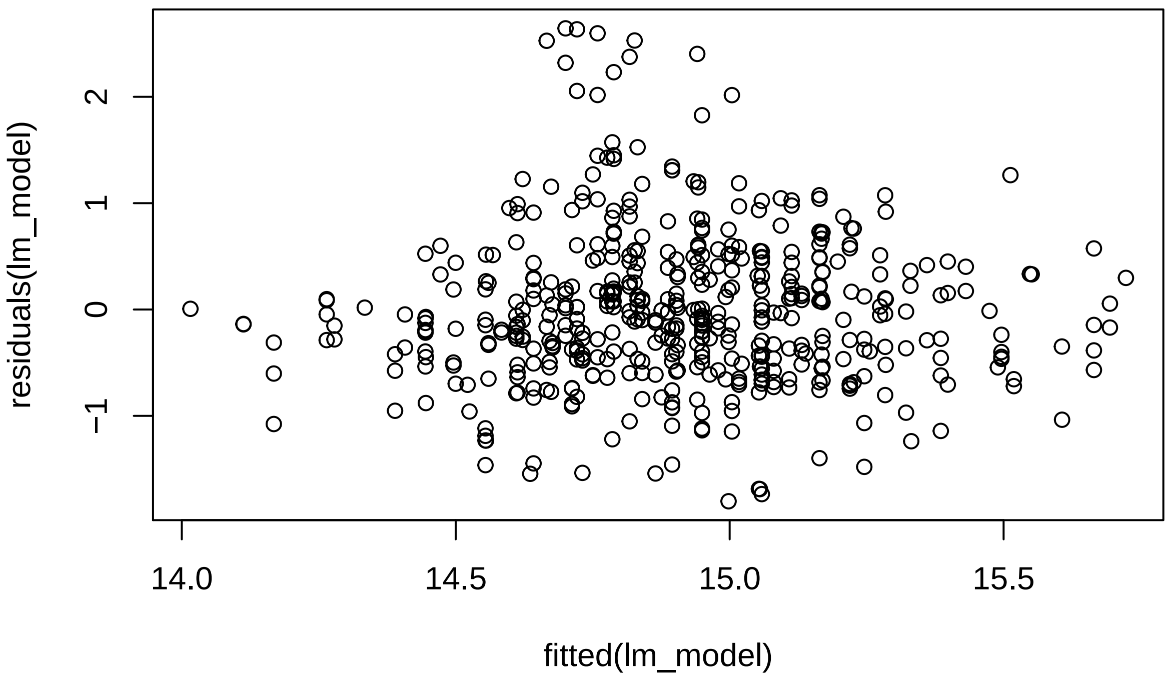
Check conditions of Normality of Residuals, Homoskedasticity,Linearity , No Perfect Multicollinearity, Independence of Residuals

```r
for (lm_model in region_lm_models_train) {

# 1. Linearity Check (Visual Inspection)
plot(residuals(lm_model) ~ fitted(lm_model))

# 2. Independence Check
# Durbin-Watson test
print(durbinWatsonTest(lm_model)[3])

# 3. Homoscedasticity Check
# Breusch-Pagan test for Heteroscedasticity
print(bptest(lm_model)$p.value)

# 4. Normality of Residuals Check (Visual Inspection)
hist(residuals(lm_model), main = "Histogram of Residuals")
}
```

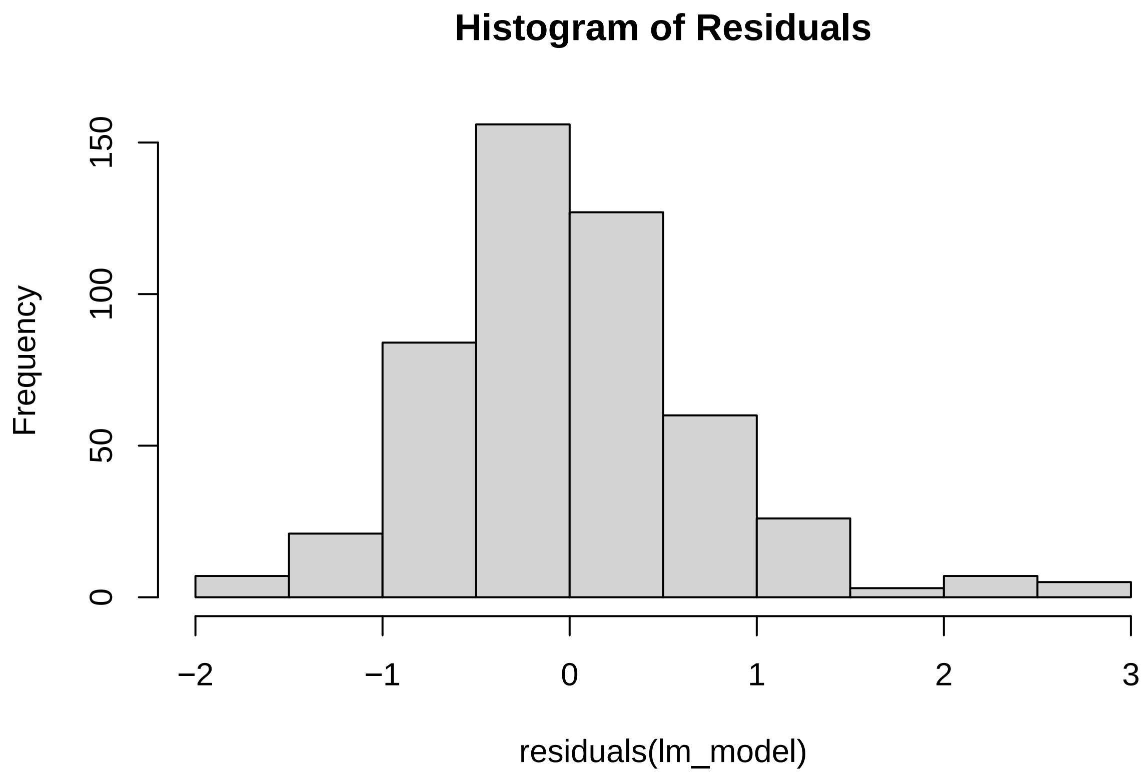


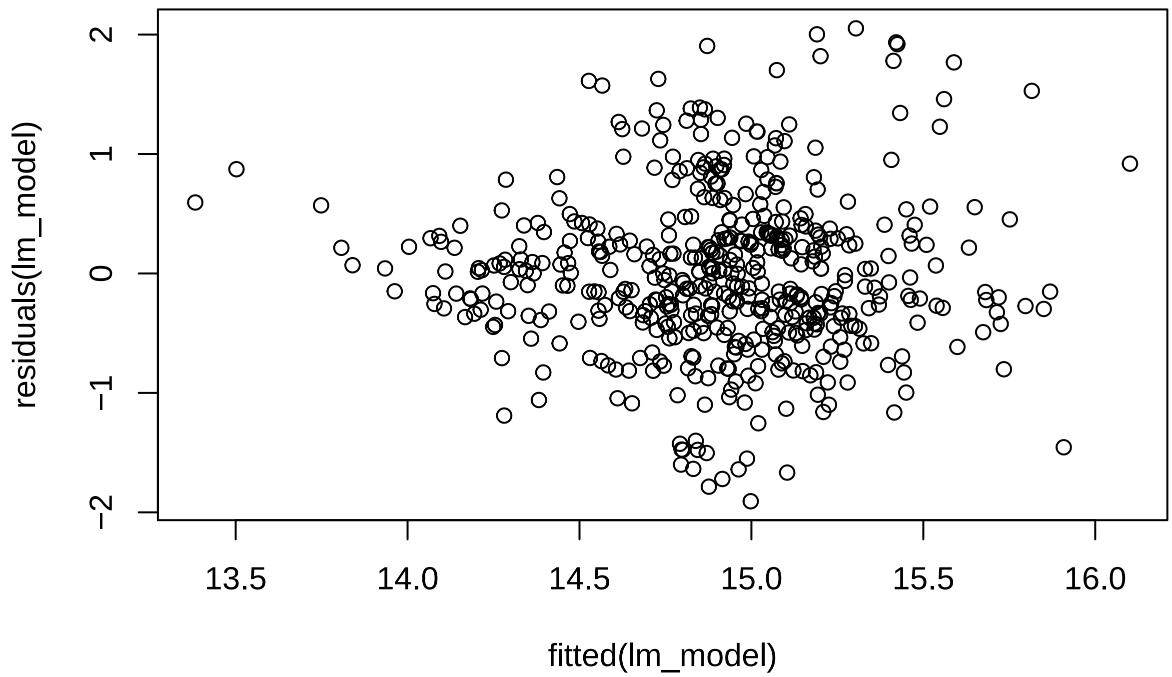

```
## $p
## [1] 0.202
##
##        BP
## 0.4839808
```


Histogram of Residuals
Frequency
0
50
100
150
−2
−1
0
1
2
residuals(lm_model)

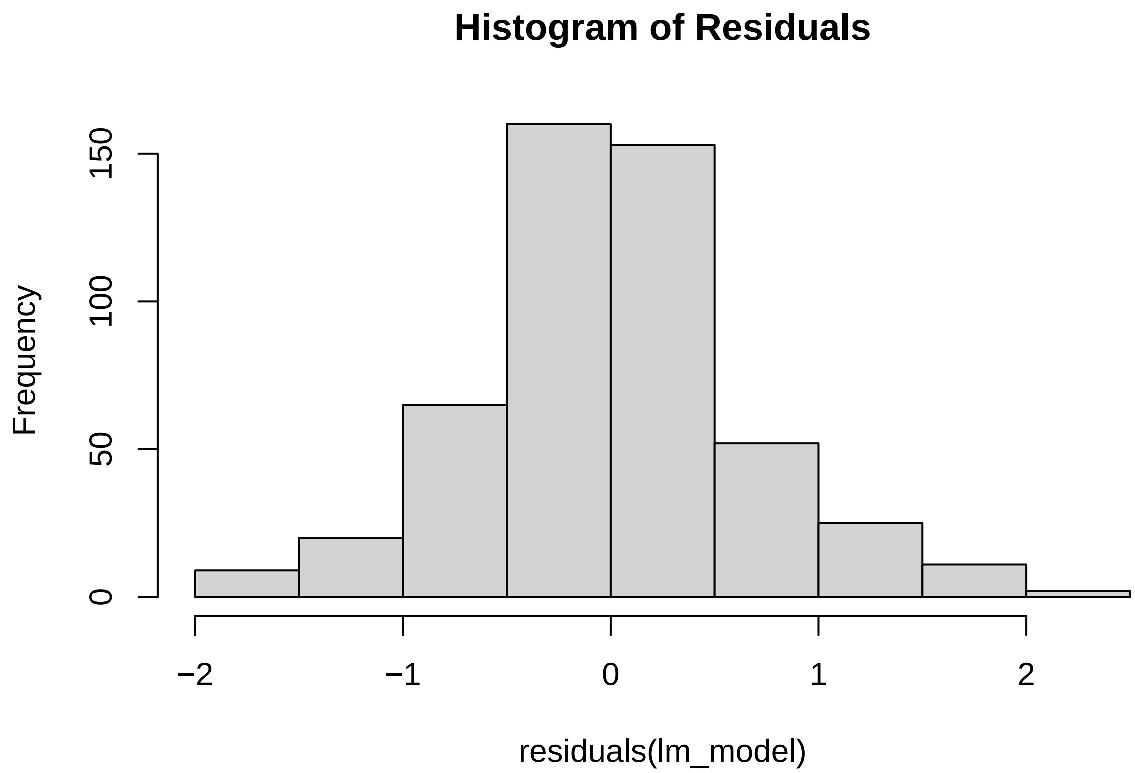


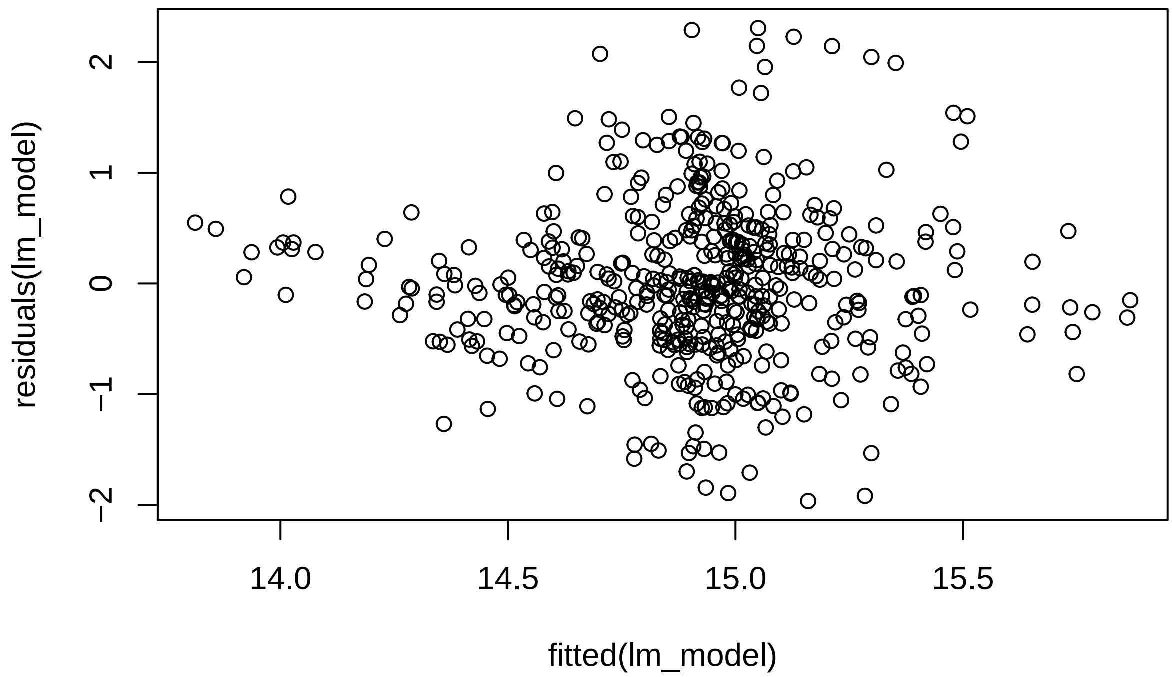

```
## $p
## [1] 0.246
##
##        BP
## 0.8431523
```


Histogram of Residuals
Frequency
0
50
100
150
−2
−1
0
1
2
3
residuals(lm_model)

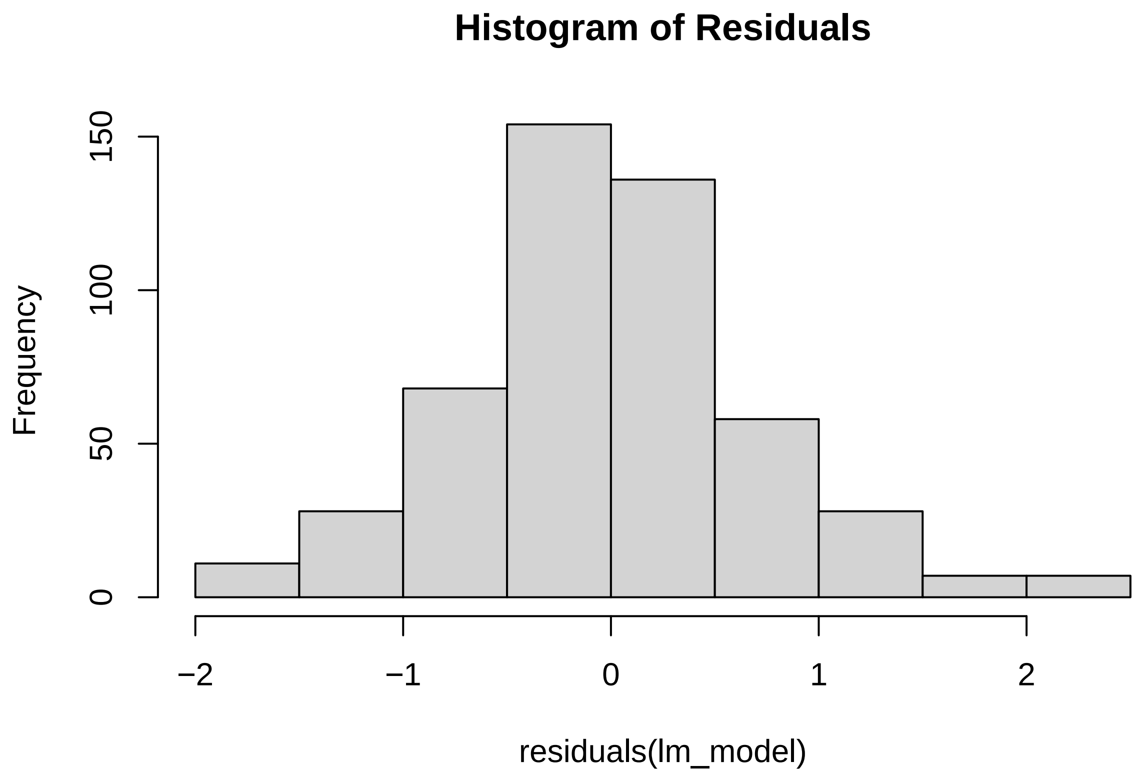


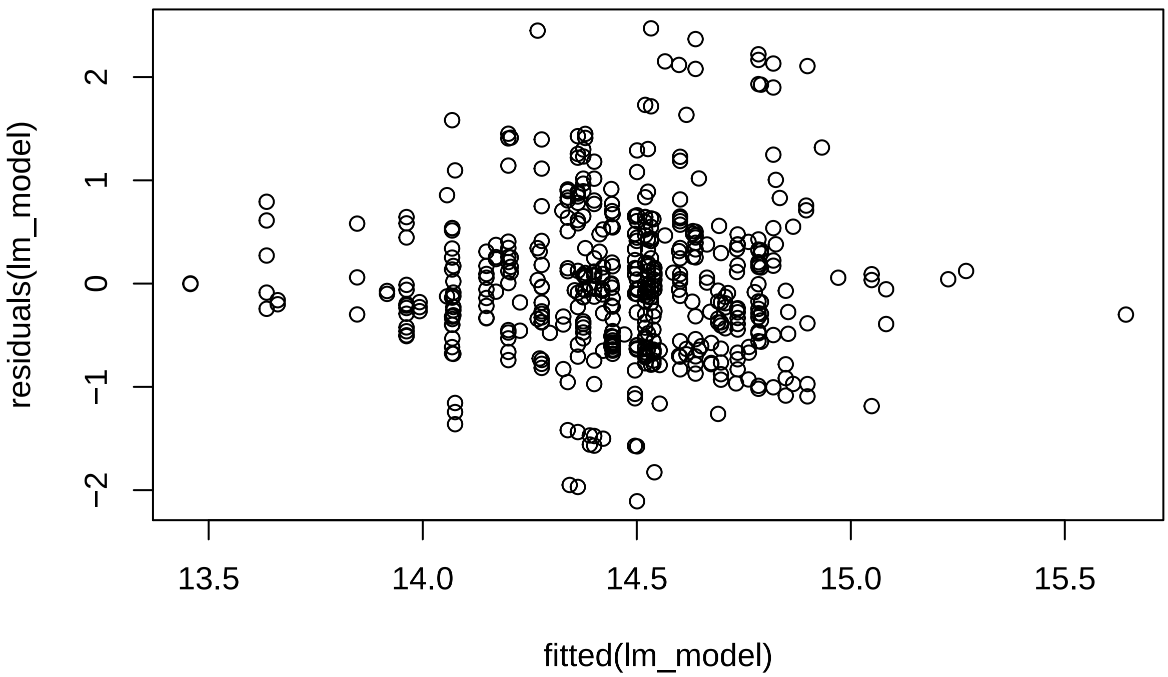

```
## $p
## [1] 0.322
##
##          BP
## 0.00089838
```


Histogram of Residuals
Frequency
0
50
100
150
−2
−1
0
1
2
residuals(lm_model)

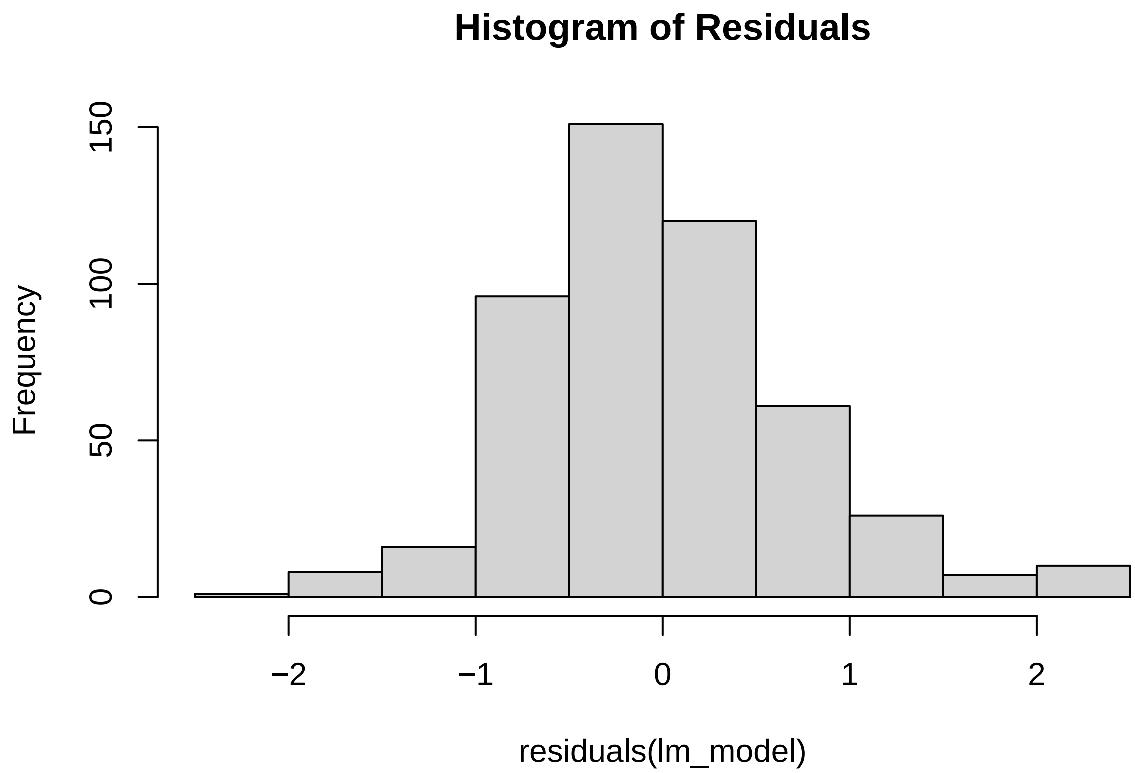


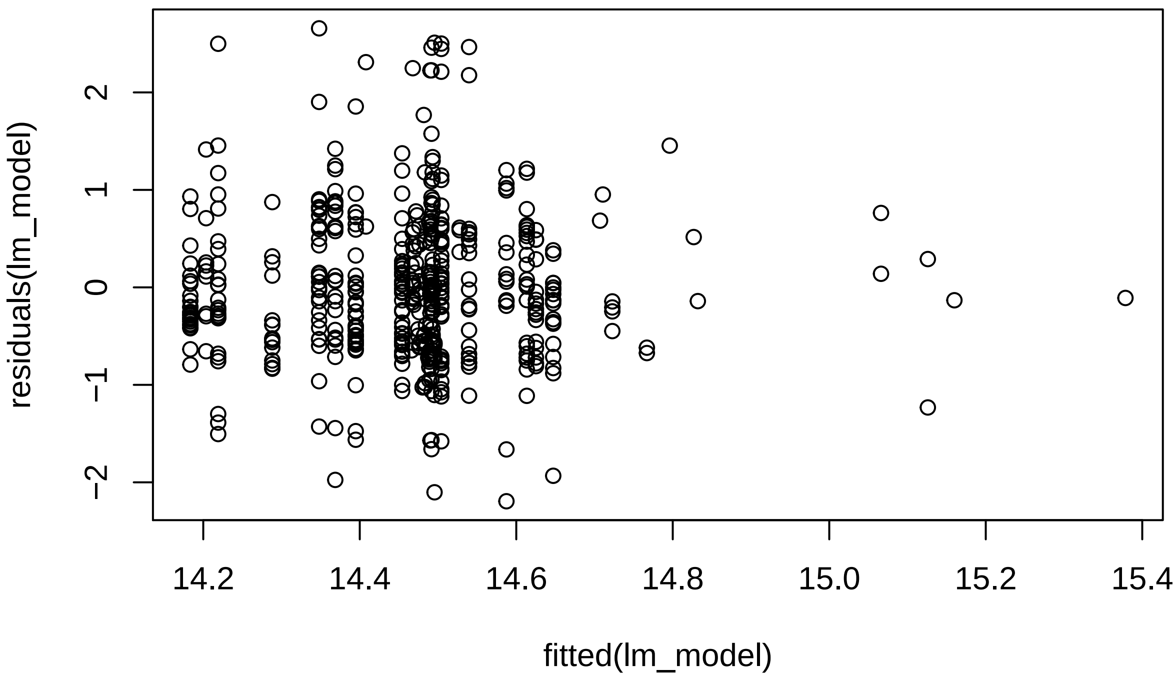

```
## $p
## [1] 0.136
##
##         BP
## 0.0447421
```


Histogram of Residuals
Frequency
0
50
100
150
−2
−1
0
1
2
residuals(lm_model)

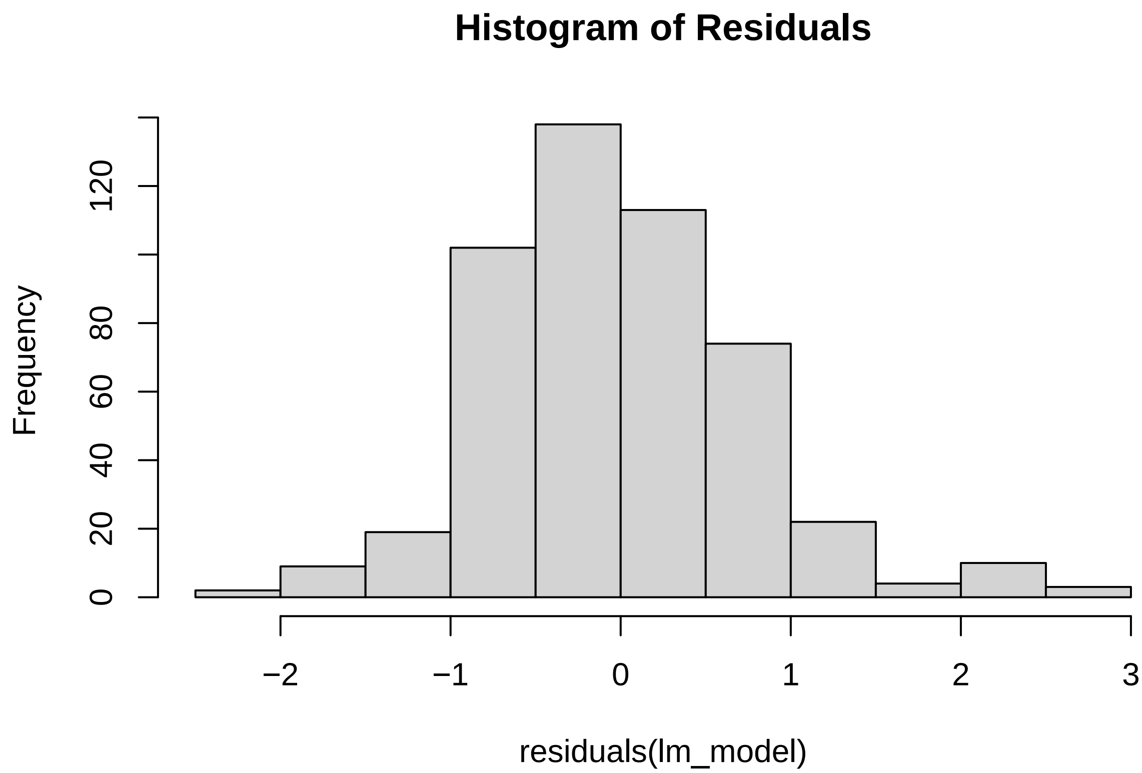


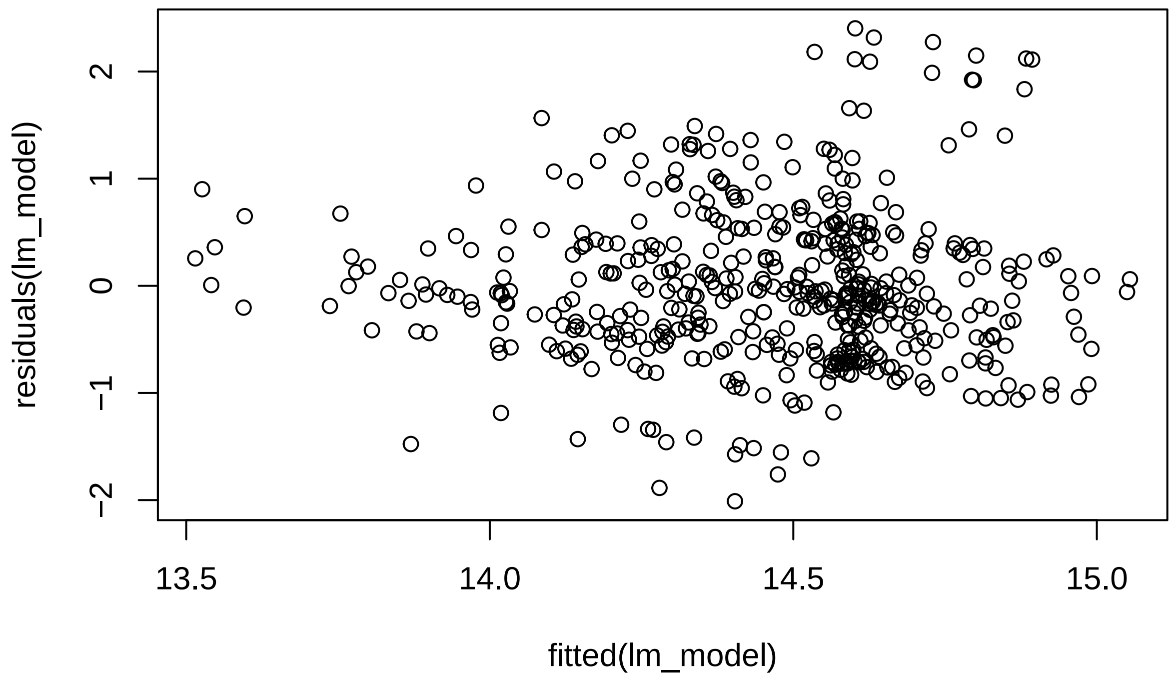

```
## $p
## [1] 0.474
##
##          BP
## 0.02973841
```


Histogram of Residuals
Frequency
0
50
100
150
−2
−1
0
1
2
residuals(lm_model)

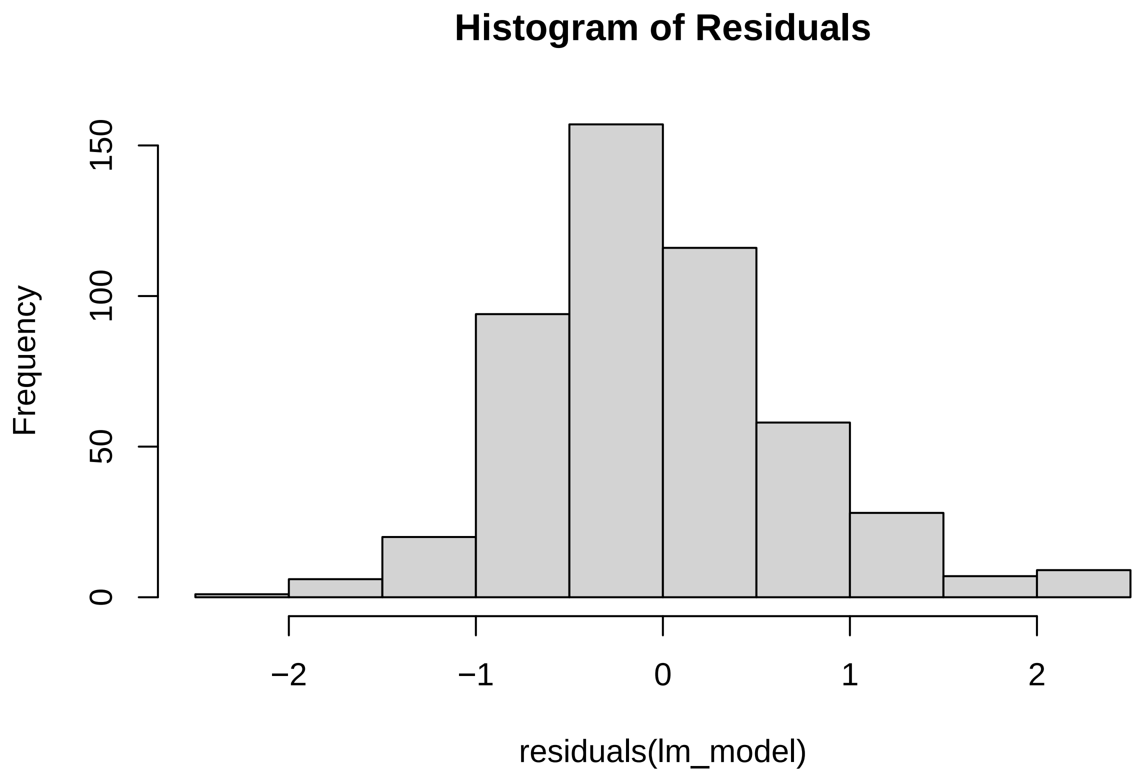


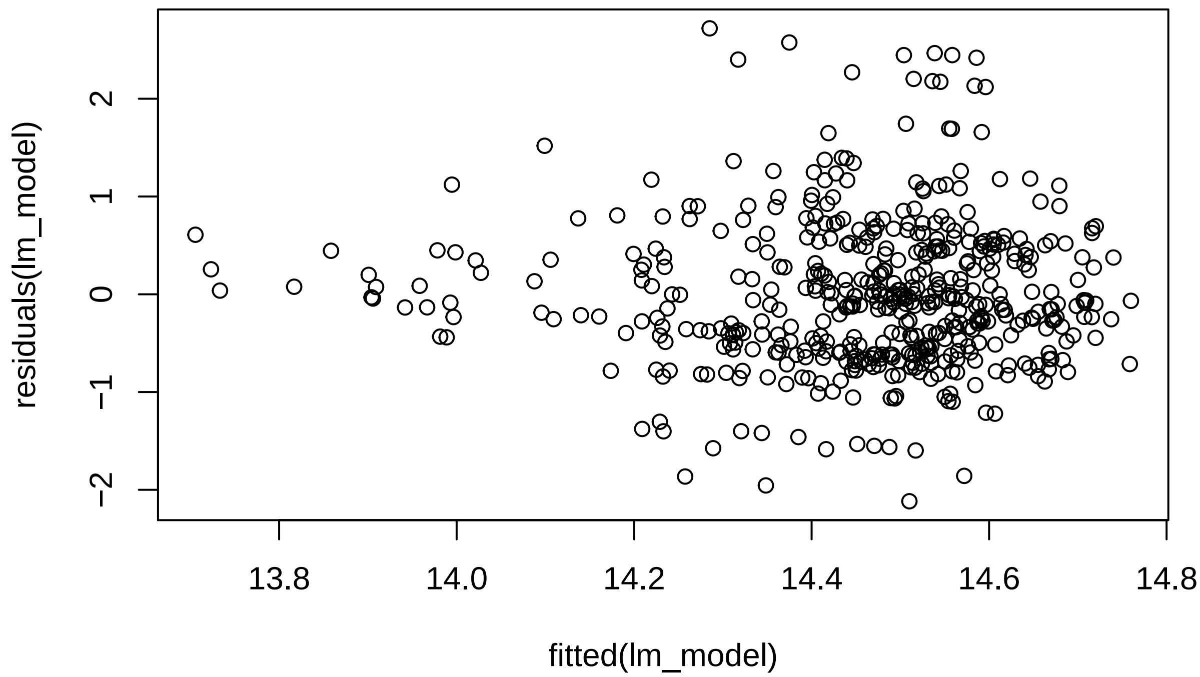

```
## $p
## [1] 0.864
##
##        BP
## 0.2998349
```


Histogram of Residuals
Frequency
0
20
40
60
80
120
−2
−1
0
1
2
3
residuals(lm_model)

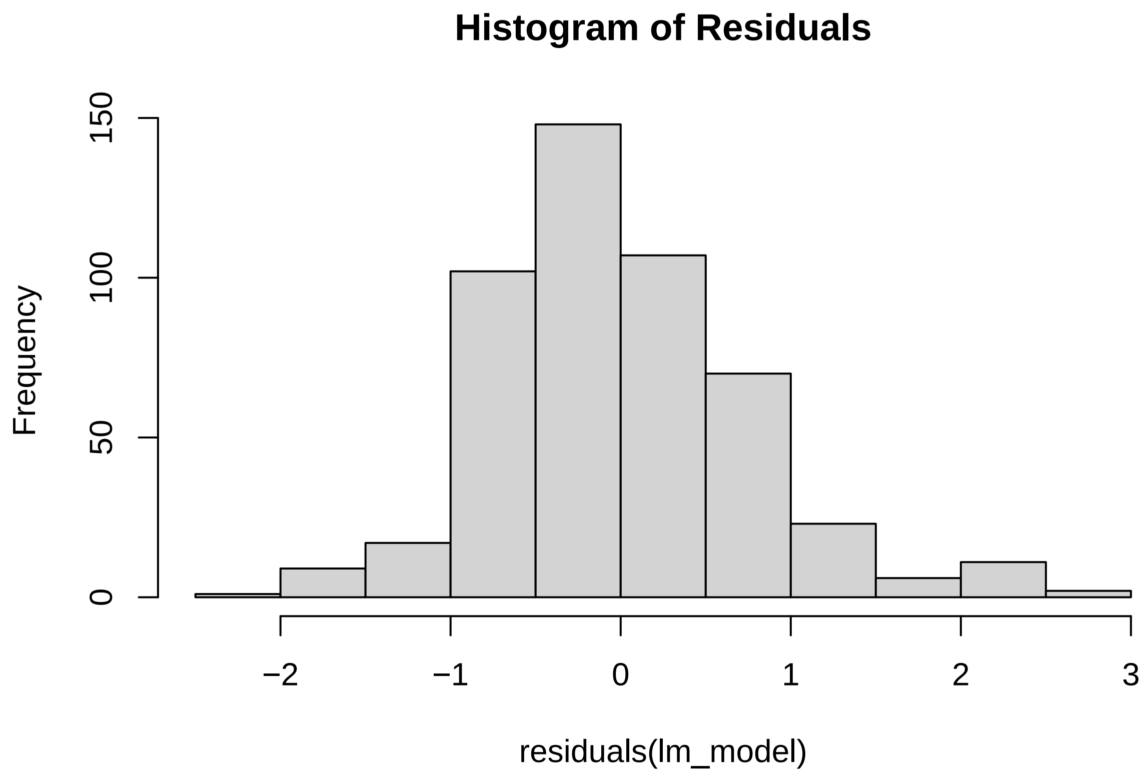



```
## $p
## [1] 0.874
##
##           BP
## 0.004687493
```


Histogram of Residuals
Frequency
0
50
100
150
−2
−1
0
1
2
residuals(lm_model)




```
## $p
## [1] 0.87
##
##         BP
## 0.05684475
```




```r
# 5. No Perfect Multicollinearity Check
# Variance Inflation Factors (VIF)
vif(lm(log(next_week_marginal_cases) ~  cor_lengths + factor(wave) + factor(region) , data = all_regiona
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## cor_lengths     1.271433  1        1.127578
## factor(wave)    1.039231  2        1.009667
## factor(region) 1.232046  3        1.035391
```

```r
vif(lm(log(next_week_marginal_cases) ~  cor_lengths + factor(wave) + factor(region) , data = urban_regi
```

```
##                     GVIF Df GVIF^(1/(2*Df))
## cor_lengths     1.205064  1        1.097754
## factor(wave)    1.044123  2        1.010853
## factor(region) 1.160942  3        1.025184
```

```r
names(region_lm_models)
```

```
## [1] "all_corlengths_wave_region"   "all_corlengths_region"
## [3] "all_localcor_wave_region"     "all_localcor_region"
## [5] "urban_corlengths_wave_region" "urban_corlengths_region"
## [7] "urban_localcor_wave_region"   "urban_localcor_region"
```

```
# Conditions not passed:
## Heteroskedastic in all_corlengths_wave_region, all_corlengths_region, urban_corlengths_region, urban_
## Non-linearity in urban_localcor_region
## solutions --> use robust standard errors, reject a 99% level for strong results
```

Interpretations

```
# Initialize values
predictor <- c("cor_lengths","r_0_50")
waves <- c("wave 1","wave 2", "wave 3")
regions <- c("Northeast", "Midwest", "South", "West")
full_coef = c()
full_coef1 = c()
significant = c()
significant1 = c()
i=1
j=1

# Iterate over Regions and Wave dummies
for (model in 1:length(region_lm_models_train)) {

  # Get model stats
  lm_model = region_lm_models_train[[model]]
  coefficients <- coef(lm_model)
  coef_names <- names(coef(lm_model))

   # Get stats
  # se <- summary(lm_model)$coefficients[, "Std. Error"] use robust SE's
  se <- sqrt(diag(vcovHC(lm_model)))
  se_named <- rep(NA, length(coef_names))
  se_named[names(se)] <- se
  st_errors <- se_named[coef_names]

  # Based on model, decide if urban or all county df
  if (model <=4) {
    county = "all counties"
    df = all_regional_weekly_spatial_metrics}
  else {
    county = "urban counties"
    df = urban_regional_weekly_spatial_metrics}

  # Based on model, choose cor_length or local_cor as predictor
  if (model %in% c(1,2,5,6)) {pred = predictor[1]}
  else {pred = predictor[2]}

  # Wave dummies not included for even models
  if (model %% 2 == 0) {
    wave = "every wave"
    for (r in 1:length(regions)) {
      region <- regions[r]
      if (r==4) {interaction = 0
```

```r
    se_total = st_errors[2]}
    else {
    interaction = coefficients[5+r]
    se_total = sqrt(sum((st_errors[5+r])^2 + (st_errors[2])^2))
    }
    # Store coefficient
    full_coef1[j] = save_coefficient(interaction)

    # Print interpretation and significance
    significant1[j] <- is_significant(as.numeric(full_coef1[j]), as.numeric((se_total)))
    print_interpretation(full_coef1[j], wave, region, "are disregarded", significant1[j], pred, county

# Wave dummies with mhhinc tier dummies
else {
# Iterate over each wave
for (w in 1:length(waves)) {
  wave <- waves[w]
  # Iterate of each tier difference
  for (r in 1:length(regions)) {
    region <- regions[r]
    if (wave == "wave 1") {
      partial_interaction_w <- 0
      se_partial_w <- 0
    } else {
      partial_interaction_w <- coefficients[(6 + w)]
      se_partial_w <- st_errors[(6 + w)]
    }
    if (region == "West") {
      partial_interaction_r <- 0
      se_partial_r <- 0
    } else {
      partial_interaction_r <- coefficients[(3*r+7)]
      se_partial_r <- st_errors[(3*r+7)]
    }
    if (region != "West" & wave != "wave 1") {
      full_interaction_wr <- coefficients[paste0(pred,":factor(wave)", w, ":", tolower(region))]
      se_full_wr <- st_errors[paste0(pred,":factor(wave)", w, ":", region)]
    } else {
      full_interaction_wr <- 0
      se_full_wr <- 0
    }
  # Calculate total standard error
  se_total <- sqrt(se_partial_w^2 + se_partial_r^2 + se_full_wr^2 + st_errors[2]^2)
  # Store coefficient information
  interaction <- partial_interaction_w + partial_interaction_r + full_interaction_wr

  # Store coefficient
  full_coef[i] = save_coefficient(interaction)

  # Print interpretation and significance
  significant[i] <- is_significant(as.numeric(full_coef[i]), as.numeric(se_total))
  print_interpretation(full_coef[i], wave, region, "are disregarded", significant[i], pred, county, di
  i <- i + 1
```

```
    }}}
}
```

```
## - For all counties across wave 1 in the Northeast where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 173.07% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 1 in the Midwest where the median household income tiers are disregard
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 75.67% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 1 in the South where the median household income tiers are disregarded
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 77.01% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 1 in the West where the median household income tiers are disregarded
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 57.21% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 2 in the West where the median household income tiers are disregarded
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 50.46% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the Northeast where the median household income tiers are di
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 66.56% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the Midwest where the median household income tiers are disr
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 67.51% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the South where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 46.49% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the West where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 35.65% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 1 in the Midwest where the median household income tiers are disregard
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 44.06% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 1 in the South where the median household income tiers are disregarded
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 66.25% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 1 in the West where the median household income tiers are disregarded
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 27.12% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 3 in the West where the median household income tiers are disregarded
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 46.16% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the Midwest where the median household income tiers are disr
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 26.84% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the South where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 47.45% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the West where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 29.15% increase in cases in the following week. Significant? TRUE
## - For urban counties across wave 1 in the Midwest where the median household income tiers are disrega
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
```

```
##  corresponds to a 156.69% increase in cases in the following week. Significant? TRUE
## - For urban counties across wave 1 in the West where the median household income tiers are disregard
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 227.8% increase in cases in the following week. Significant? TRUE
## - For urban counties across every wave in the Midwest where the median household income tiers are di
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 133.34% increase in cases in the following week. Significant? TRUE
## - For urban counties across every wave in the West where the median household income tiers are disre
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 73.01% increase in cases in the following week. Significant? TRUE
## - For urban counties across wave 1 in the West where the median household income tiers are disregard
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 25.02% increase in cases in the following week. Significant? TRUE
## - For urban counties across every wave in the West where the median household income tiers are disre
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 17.48% increase in cases in the following week. Significant? TRUE
```

MHHINC + Wave Models

Prepare Model Combinations

```r
# Data sets: (1) all counties & (2) urban counties
datasets = list(all_mhhinc_weekly_spatial_metrics, urban_mhhinc_weekly_spatial_metrics)
df_names = c("all","urban")

# Predictor: (1) correlation length & (2) local correlation
predictor = c("cor_lengths", "r_0_50") #unlog local cor
predictor_names = c("corlengths","localcor")

# Covariates (1) mhhinc & (2) mhhinc + wave
covariates = list(c("*factor(wave)*same_tier","*factor(wave)*tier_diff_1","*factor(wave)*tier_diff_2"),
                  c("*same_tier","*tier_diff_1","*tier_diff_2"))
var_names = c("wave_mhhinc","mhhinc")
```

Generate Models

```r
# Create an empty list to store the models
mhhinc_lm_models <- list()

# Iterate of every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df = datasets[d][[1]]
  for (p in 1:length(predictor)) {
    for (c in 1:length(covariates)) {
      interactions = c()
      for (i in 1:length(covariates[[c]])) {
        interactions[i] <- paste(predictor[p], covariates[[c]][i])}

      # Create the formula
```

```r
      lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), col

      # Fit the linear model
      lm_model <- lm(lm_formula, data = df)

      # Generate a name for the model object
      model_name <- paste(df_names[d],"_", predictor_names[p], "_", var_names[c], sep = "")

      # Save the model to the list
      mhhinc_lm_models[[model_name]] <- lm_model
    }}}
```

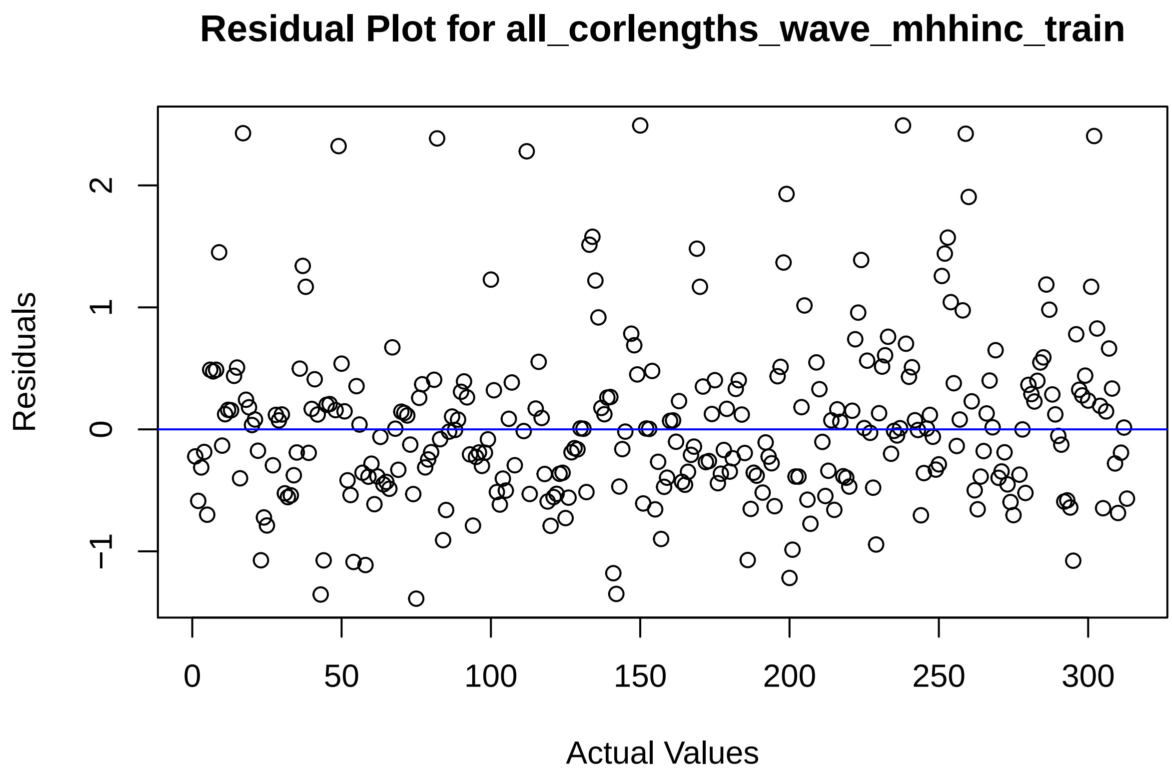
```r
print(names(mhhinc_lm_models))
```

```
## [1] "all_corlengths_wave_mhhinc"   "all_corlengths_mhhinc"
## [3] "all_localcor_wave_mhhinc"     "all_localcor_mhhinc"
## [5] "urban_corlengths_wave_mhhinc" "urban_corlengths_mhhinc"
## [7] "urban_localcor_wave_mhhinc"   "urban_localcor_mhhinc"
```

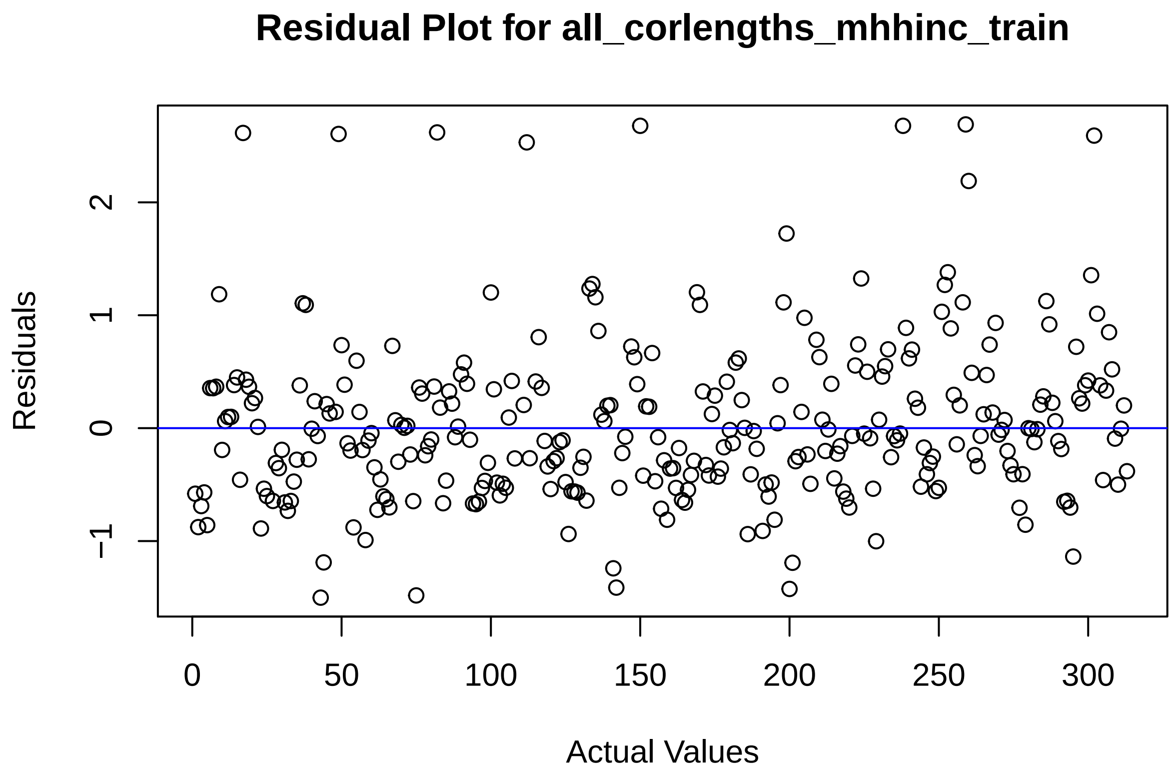
Training/Testing Models

```r
# Set seed for reproducibility
set.seed(1)

# Create an empty list to store the models
mhhinc_lm_models_train <- list()

# Iterate over every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df <- datasets[[d]]

  # Create indices for train-test split
  n_obs <- nrow(df)
  train_indices <- sample(1:n_obs, size = round(train_ratio * n_obs), replace = FALSE)
  test_indices <- setdiff(1:n_obs, train_indices)

  # Split the data into train and test sets
  train_df <- df[train_indices, ]
  test_df <- df[test_indices, ]

  for (p in 1:length(predictor)) {
    for (c in 1:length(covariates)) {
      interactions <- c()
      for (i in 1:length(covariates[[c]])) {
        interactions[i] <- paste(predictor[p], covariates[[c]][i])
      }

      # Create the formula & fit linear model on the training data
      lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), col
      lm_model_train <- lm(lm_formula, data = train_df)
```

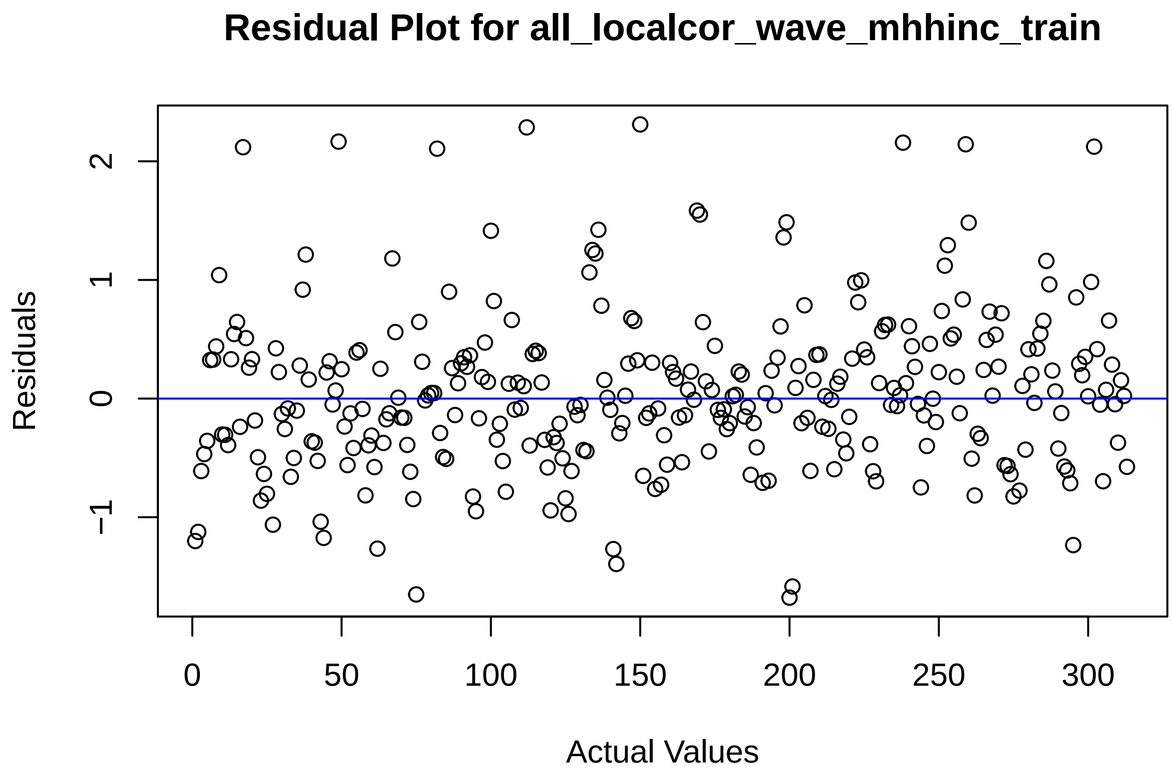
```
      # Name model and save to list
      model_name_train <- paste(df_names[d], "_", predictor_names[p], "_", var_names[c], "_train", sep =
      mhhinc_lm_models_train[[model_name_train]] <- lm_model_train

      # Apply the trained model to the test data and make predictions
      test_predictions <- predict(lm_model_train, newdata = test_df)

      # Calculate residuals using predictions from the test dataset
      residuals <- log(test_df$next_week_marginal_cases) - test_predictions

      # Create residual plot
      plot(1:length(residuals), residuals,
           main = paste("Residual Plot for", model_name_train),
           xlab = "Actual Values",
           ylab = "Residuals")
      abline(h = 0, col = "blue")

      # Check if the mean of residuals is less than the standard deviation of the data
      print(mean(abs(residuals), na.rm = TRUE) < sd(log(test_df$next_week_marginal_cases), na.rm = TRUE
    }
  }
}
```




```
## [1] TRUE
```

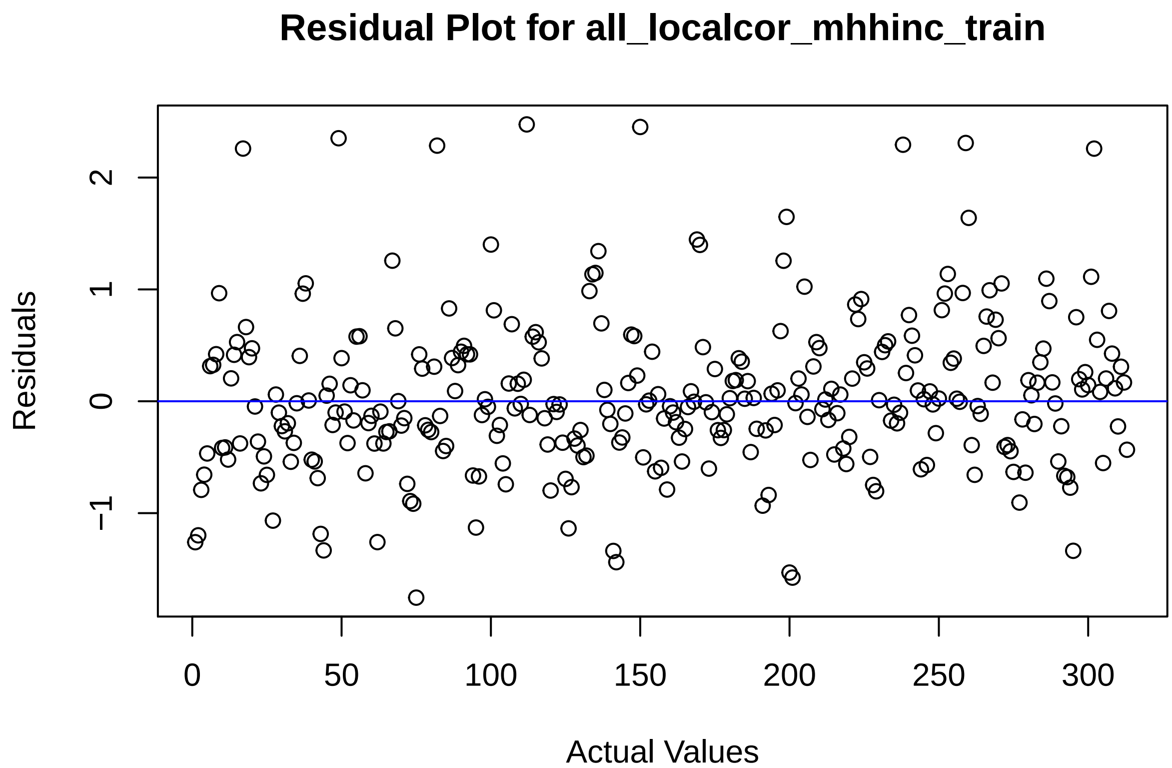

Residual Plot for all_corlengths_mhhinc_train
Residuals
Actual Values
−1
0
1
2
0
50
100
150
200
250
300

```
## [1] TRUE
```

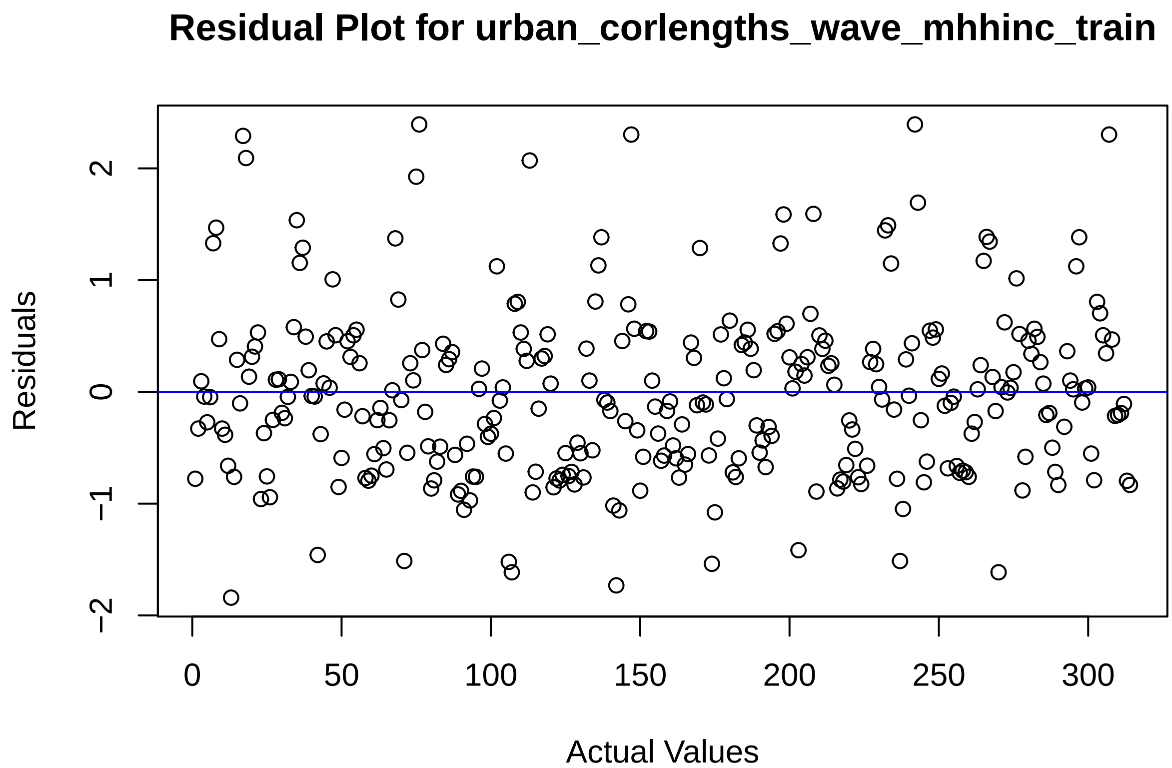


Residual Plot for all_localcor_wave_mhhinc_train
Residuals
Actual Values
2
1
0
−1
0
50
100
150
200
250
300

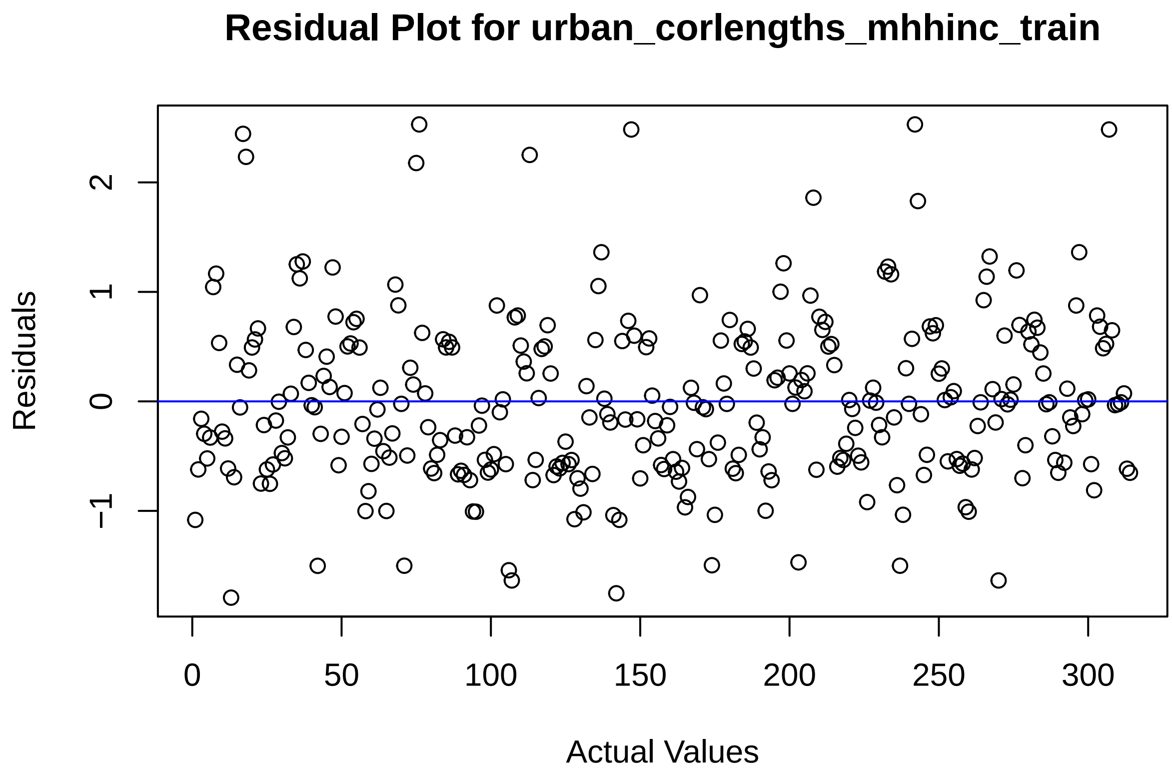

```
## [1] TRUE
```


Residual Plot for all_localcor_mhhinc_train
Residuals
Actual Values
2
1
0
−1
0
50
100
150
200
250
300

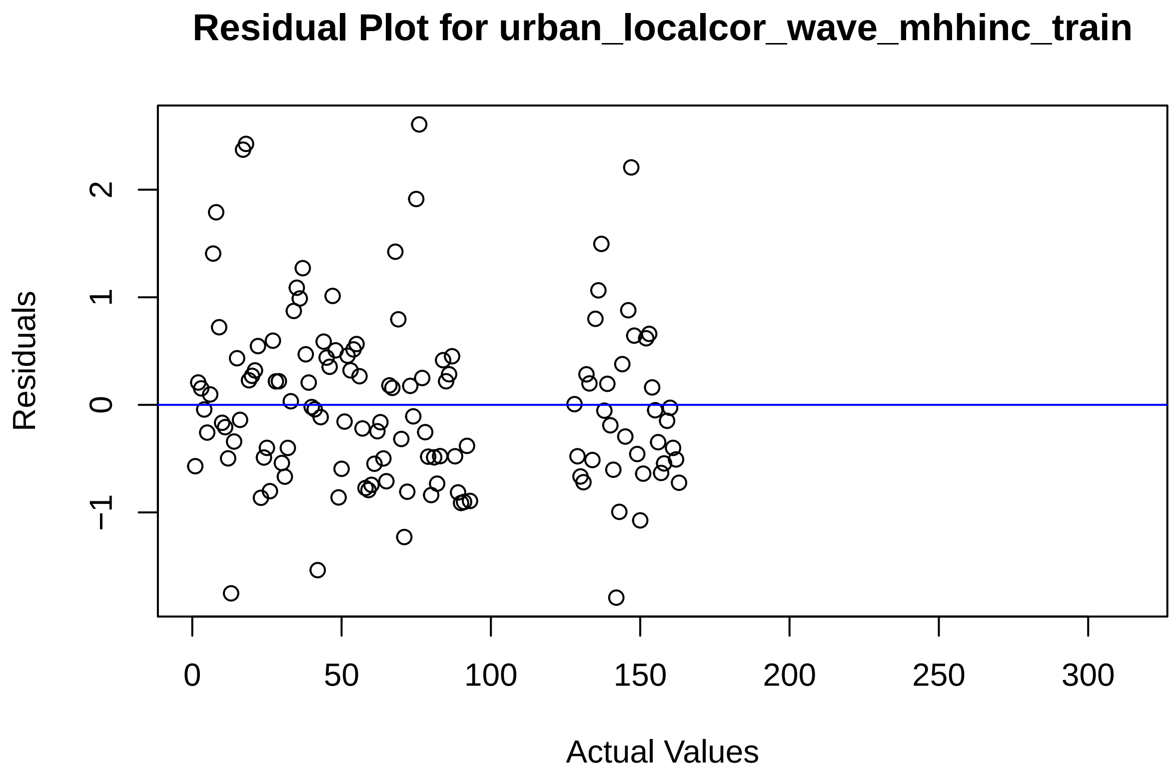
```
## [1] TRUE
```


Residual Plot for urban_corlengths_wave_mhhinc_train
Residuals
Actual Values
2
1
0
−1
−2
0
50
100
150
200
250
300

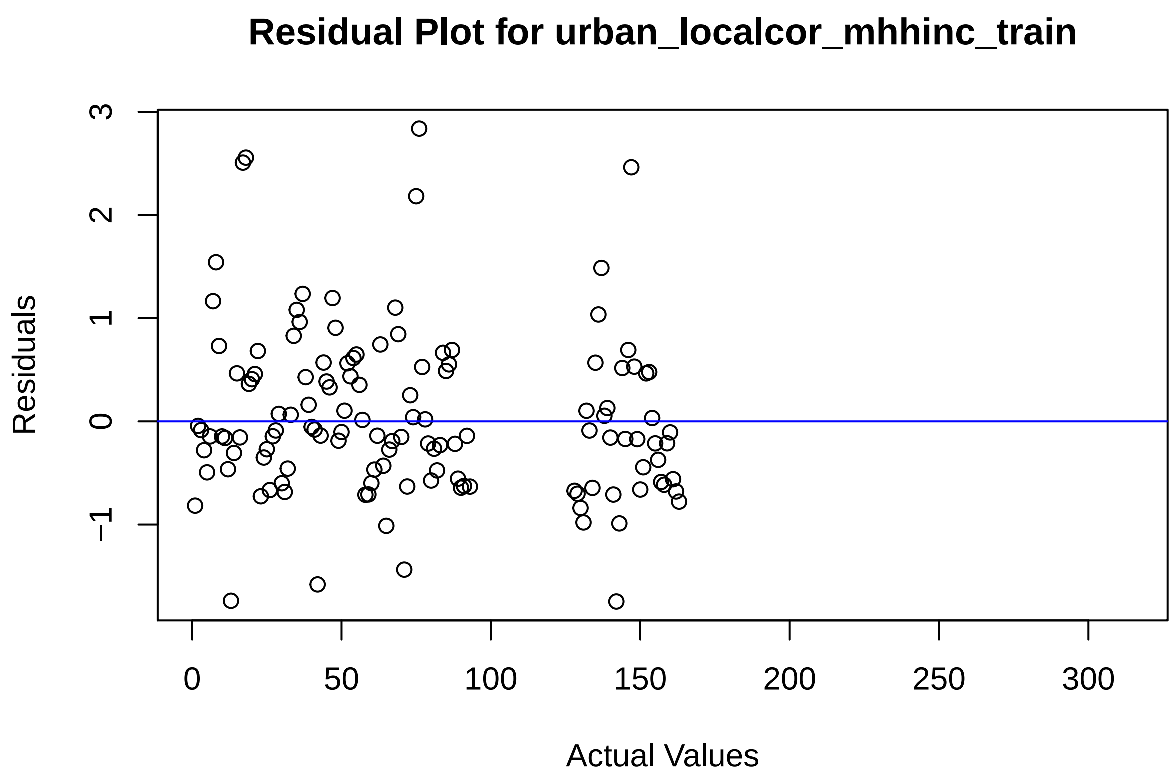
```
## [1] TRUE
```


Residual Plot for urban_corlengths_mhhinc_train
Residuals
Actual Values
0
50
100
150
200
250
300
−1
0
1
2

```
## [1] TRUE
```


Residual Plot for urban_localcor_wave_mhhinc_train
Residuals
Actual Values
2
1
0
−1
0
50
100
150
200
250
300

```
## [1] TRUE
```




```
## [1] TRUE
```

```r
# Print the names of the models
names(mhhinc_lm_models_train)
```

```
## [1] "all_corlengths_wave_mhhinc_train"   "all_corlengths_mhhinc_train"
## [3] "all_localcor_wave_mhhinc_train"     "all_localcor_mhhinc_train"
## [5] "urban_corlengths_wave_mhhinc_train" "urban_corlengths_mhhinc_train"
## [7] "urban_localcor_wave_mhhinc_train"   "urban_localcor_mhhinc_train"
```

Seems like the models are unreliable for the urban + local correlation models

View Model Summaries

```r
for (m in 1:length(mhhinc_lm_models_train)){
  print(summary(mhhinc_lm_models_train[[m]]))
}
```

```
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0002 -0.4526 -0.1204  0.3459  2.4546
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            1.414e+01  1.934e-01  73.131  < 2e-16
## cor_lengths                            1.043e-03  3.964e-04   2.631  0.00862
## factor(wave)2                          3.756e-01  2.723e-01   1.379  0.16809
## factor(wave)3                          7.322e-01  2.446e-01   2.993  0.00282
## same_tier                             -2.018e-01  2.262e-01  -0.892  0.37255
## tier_diff_1                           -2.783e-01  2.376e-01  -1.171  0.24166
## tier_diff_2                           -9.062e-02  2.409e-01  -0.376  0.70689
## cor_lengths:factor(wave)2             -1.212e-04  5.303e-04  -0.228  0.81932
## cor_lengths:factor(wave)3             -3.029e-04  5.709e-04  -0.531  0.59585
## cor_lengths:same_tier                  5.192e-04  4.504e-04   1.153  0.24915
## factor(wave)2:same_tier               -2.467e-02  3.257e-01  -0.076  0.93963
## factor(wave)3:same_tier               -1.299e-01  2.899e-01  -0.448  0.65420
## cor_lengths:tier_diff_1                6.314e-04  4.629e-04   1.364  0.17288
## factor(wave)2:tier_diff_1             -2.130e-01  3.406e-01  -0.625  0.53191
## factor(wave)3:tier_diff_1              5.176e-02  3.052e-01   0.170  0.86534
## cor_lengths:tier_diff_2                3.459e-04  4.695e-04   0.737  0.46147
## factor(wave)2:tier_diff_2             -2.050e-01  3.487e-01  -0.588  0.55665
## factor(wave)3:tier_diff_2              2.269e-02  3.111e-01   0.073  0.94187
## cor_lengths:factor(wave)2:same_tier   -1.828e-04  6.182e-04  -0.296  0.76747
## cor_lengths:factor(wave)3:same_tier   -1.230e-05  6.532e-04  -0.019  0.98498
## cor_lengths:factor(wave)2:tier_diff_1  7.791e-05  6.320e-04   0.123  0.90191
## cor_lengths:factor(wave)3:tier_diff_1 -3.480e-04  6.704e-04  -0.519  0.60378
## cor_lengths:factor(wave)2:tier_diff_2  2.170e-04  6.477e-04   0.335  0.73762
## cor_lengths:factor(wave)3:tier_diff_2 -4.455e-04  6.963e-04  -0.640  0.52242
##
## (Intercept)                           ***
## cor_lengths                           **
## factor(wave)2
## factor(wave)3                         **
## same_tier
## tier_diff_1
## tier_diff_2
## cor_lengths:factor(wave)2
## cor_lengths:factor(wave)3
## cor_lengths:same_tier
## factor(wave)2:same_tier
## factor(wave)3:same_tier
## cor_lengths:tier_diff_1
## factor(wave)2:tier_diff_1
## factor(wave)3:tier_diff_1
## cor_lengths:tier_diff_2
## factor(wave)2:tier_diff_2
## factor(wave)3:tier_diff_2
## cor_lengths:factor(wave)2:same_tier
## cor_lengths:factor(wave)3:same_tier
## cor_lengths:factor(wave)2:tier_diff_1
## cor_lengths:factor(wave)3:tier_diff_1
## cor_lengths:factor(wave)2:tier_diff_2
```

```
## cor_lengths:factor(wave)3:tier_diff_2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7105 on 1197 degrees of freedom
##   (35 observations deleted due to missingness)
## Multiple R-squared:  0.2316, Adjusted R-squared:  0.2169
## F-statistic: 15.69 on 23 and 1197 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0583 -0.4863 -0.1170  0.4029  2.6900
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            14.6076778  0.1031919 141.558  < 2e-16 ***
## cor_lengths             0.0007907  0.0002233   3.541 0.000414 ***
## same_tier              -0.2543245  0.1232816  -2.063 0.039329 *
## tier_diff_1            -0.3092946  0.1296142  -2.386 0.017173 *
## tier_diff_2            -0.1592452  0.1323603  -1.203 0.229165
## cor_lengths:same_tier   0.0004592  0.0002597   1.768 0.077279 .
## cor_lengths:tier_diff_1  0.0005227  0.0002657   1.967 0.049403 *
## cor_lengths:tier_diff_2  0.0003191  0.0002743   1.163 0.245006
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7373 on 1213 degrees of freedom
##   (35 observations deleted due to missingness)
## Multiple R-squared:  0.1614, Adjusted R-squared:  0.1566
## F-statistic: 33.36 on 7 and 1213 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.94513 -0.47805 -0.01981  0.36628  2.54775
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    13.82076    0.31157  44.359  < 2e-16 ***
## r_0_50                          1.65699    0.54872   3.020  0.00258 **
## factor(wave)2                   0.57549    0.41518   1.386  0.16596
## factor(wave)3                   0.91567    0.39523   2.317  0.02068 *
## same_tier                      -0.02292    0.34578  -0.066  0.94717
## tier_diff_1                    -0.35102    0.35587  -0.986  0.32414
## tier_diff_2                    -0.05904    0.37180  -0.159  0.87386
## r_0_50:factor(wave)2           -0.53722    0.76566  -0.702  0.48304
## r_0_50:factor(wave)3           -0.88806    0.76051  -1.168  0.24315
```

```
## r_0_50:same_tier                 0.29962    0.62994   0.476  0.63442
## factor(wave)2:same_tier         -0.29434    0.49116  -0.599  0.54910
## factor(wave)3:same_tier         -0.46754    0.44435  -1.052  0.29293
## r_0_50:tier_diff_1               1.22900    0.66227   1.856  0.06373 .
## factor(wave)2:tier_diff_1       -0.31507    0.49831  -0.632  0.52732
## factor(wave)3:tier_diff_1       -0.23956    0.46162  -0.519  0.60388
## r_0_50:tier_diff_2               0.56962    0.70284   0.810  0.41784
## factor(wave)2:tier_diff_2        0.17599    0.52832   0.333  0.73910
## factor(wave)3:tier_diff_2       -0.60870    0.48437  -1.257  0.20911
## r_0_50:factor(wave)2:same_tier    0.06287    0.89978   0.070  0.94431
## r_0_50:factor(wave)3:same_tier    0.52561    0.86305   0.609  0.54263
## r_0_50:factor(wave)2:tier_diff_1 -0.22802    0.92524  -0.246  0.80538
## r_0_50:factor(wave)3:tier_diff_1 -0.04598    0.90909  -0.051  0.95967
## r_0_50:factor(wave)2:tier_diff_2 -0.84680    0.98012  -0.864  0.38777
## r_0_50:factor(wave)3:tier_diff_2  0.73865    0.96013   0.769  0.44185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7113 on 1216 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.2234, Adjusted R-squared:  0.2087
## F-statistic: 15.21 on 23 and 1216 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05103 -0.45388 -0.06537  0.41372  2.77408
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        14.4573     0.1576  91.758  < 2e-16 ***
## r_0_50              0.9844     0.3095   3.181  0.00151 **
## same_tier          -0.3793     0.1824  -2.080  0.03778 *
## tier_diff_1        -0.5914     0.1888  -3.133  0.00177 **
## tier_diff_2        -0.3491     0.1992  -1.752  0.07996 .
## r_0_50:same_tier    0.6634     0.3538   1.875  0.06101 .
## r_0_50:tier_diff_1  1.1772     0.3676   3.202  0.00140 **
## r_0_50:tier_diff_2  0.7091     0.3870   1.832  0.06720 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7251 on 1232 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.1823, Adjusted R-squared:  0.1776
## F-statistic: 39.23 on 7 and 1232 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.16176 -0.54910 -0.05372  0.40215  2.47788
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            1.407e+01  1.915e-01  73.470   <2e-16
## cor_lengths                            5.867e-03  6.357e-03   0.923   0.3562
## factor(wave)2                          2.498e-01  2.287e-01   1.092   0.2749
## factor(wave)3                          6.089e-01  2.208e-01   2.758   0.0059
## same_tier                              1.508e-02  2.286e-01   0.066   0.9474
## tier_diff_1                           -6.155e-02  2.289e-01  -0.269   0.7880
## tier_diff_2                            3.738e-02  3.067e-01   0.122   0.9030
## cor_lengths:factor(wave)2             -9.373e-04  7.228e-03  -0.130   0.8968
## cor_lengths:factor(wave)3             -7.226e-03  7.009e-03  -1.031   0.3028
## cor_lengths:same_tier                 -5.195e-03  6.371e-03  -0.815   0.4150
## factor(wave)2:same_tier                1.106e-01  2.855e-01   0.387   0.6985
## factor(wave)3:same_tier               -1.907e-01  2.723e-01  -0.700   0.4839
## cor_lengths:tier_diff_1               -3.759e-03  6.433e-03  -0.584   0.5591
## factor(wave)2:tier_diff_1              4.485e-02  2.804e-01   0.160   0.8729
## factor(wave)3:tier_diff_1              3.843e-02  2.720e-01   0.141   0.8877
## cor_lengths:tier_diff_2               -3.825e-03  8.373e-03  -0.457   0.6479
## factor(wave)2:tier_diff_2             -3.922e-02  3.779e-01  -0.104   0.9173
## factor(wave)3:tier_diff_2             -3.287e-01  3.780e-01  -0.870   0.3847
## cor_lengths:factor(wave)2:same_tier    8.601e-04  7.260e-03   0.118   0.9057
## cor_lengths:factor(wave)3:same_tier    7.409e-03  7.044e-03   1.052   0.2931
## cor_lengths:factor(wave)2:tier_diff_1  4.721e-05  7.313e-03   0.006   0.9949
## cor_lengths:factor(wave)3:tier_diff_1  5.091e-03  7.099e-03   0.717   0.4734
## cor_lengths:factor(wave)2:tier_diff_2  3.386e-03  9.882e-03   0.343   0.7319
## cor_lengths:factor(wave)3:tier_diff_2  1.186e-02  1.022e-02   1.161   0.2460
##
## (Intercept)                           ***
## cor_lengths
## factor(wave)2
## factor(wave)3                         **
## same_tier
## tier_diff_1
## tier_diff_2
## cor_lengths:factor(wave)2
## cor_lengths:factor(wave)3
## cor_lengths:same_tier
## factor(wave)2:same_tier
## factor(wave)3:same_tier
## cor_lengths:tier_diff_1
## factor(wave)2:tier_diff_1
## factor(wave)3:tier_diff_1
## cor_lengths:tier_diff_2
## factor(wave)2:tier_diff_2
## factor(wave)3:tier_diff_2
## cor_lengths:factor(wave)2:same_tier
## cor_lengths:factor(wave)3:same_tier
## cor_lengths:factor(wave)2:tier_diff_1
## cor_lengths:factor(wave)3:tier_diff_1
## cor_lengths:factor(wave)2:tier_diff_2
## cor_lengths:factor(wave)3:tier_diff_2
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.769 on 1214 degrees of freedom
##   (18 observations deleted due to missingness)
## Multiple R-squared:  0.06334,    Adjusted R-squared:  0.0456
## F-statistic: 3.569 on 23 and 1214 DF,  p-value: 3.032e-08
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.11332 -0.56496 -0.05827  0.49108  2.61499
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             14.3971963  0.0755445 190.579   <2e-16 ***
## cor_lengths              0.0027949  0.0021443   1.303    0.193
## same_tier                0.0105010  0.0994480   0.106    0.916
## tier_diff_1             -0.0297825  0.0979144  -0.304    0.761
## tier_diff_2             -0.0610437  0.1383042  -0.441    0.659
## cor_lengths:same_tier   -0.0022452  0.0021634  -1.038    0.300
## cor_lengths:tier_diff_1 -0.0018884  0.0021736  -0.869    0.385
## cor_lengths:tier_diff_2  0.0005899  0.0034691   0.170    0.865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7851 on 1230 degrees of freedom
##   (18 observations deleted due to missingness)
## Multiple R-squared:  0.01099,    Adjusted R-squared:  0.005359
## F-statistic: 1.952 on 7 and 1230 DF,  p-value: 0.05843
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05339 -0.50288 -0.06399  0.41810  2.34539
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     13.66800    0.32669  41.838  < 2e-16 ***
## r_0_50                           1.02631    0.52858   1.942  0.05278 .
## factor(wave)2                    0.45072    0.38453   1.172  0.24175
## factor(wave)3                    1.51210    0.47565   3.179  0.00158 **
## same_tier                       -0.39649    0.63367  -0.626  0.53182
## tier_diff_1                     -1.24362    0.67122  -1.853  0.06454 .
## tier_diff_2                     -0.17117    0.64614  -0.265  0.79119
## r_0_50:factor(wave)2            -0.31807    0.60529  -0.525  0.59950
## r_0_50:factor(wave)3            -1.74964    0.69158  -2.530  0.01174 *
## r_0_50:same_tier                 0.31976    0.95017   0.337  0.73663
```

```
## factor(wave)2:same_tier           0.01452    1.02735   0.014  0.98873
## factor(wave)3:same_tier          -1.13315    0.83497  -1.357  0.17540
## r_0_50:tier_diff_1                1.39036    0.95326   1.459  0.14537
## factor(wave)2:tier_diff_1         0.52592    0.77203   0.681  0.49607
## factor(wave)3:tier_diff_1         0.74571    0.86959   0.858  0.39159
## r_0_50:tier_diff_2               -0.03940    0.90570  -0.044  0.96532
## factor(wave)2:tier_diff_2        -0.54346    0.75744  -0.717  0.47343
## factor(wave)3:tier_diff_2        -0.76891    0.76582  -1.004  0.31588
## r_0_50:factor(wave)2:same_tier    0.01727    1.42925   0.012  0.99037
## r_0_50:factor(wave)3:same_tier    1.74705    1.20961   1.444  0.14932
## r_0_50:factor(wave)2:tier_diff_1 -0.79741    1.07337  -0.743  0.45791
## r_0_50:factor(wave)3:tier_diff_1 -0.70750    1.18218  -0.598  0.54982
## r_0_50:factor(wave)2:tier_diff_2  0.86967    1.05695   0.823  0.41103
## r_0_50:factor(wave)3:tier_diff_2  1.52482    1.07887   1.413  0.15822
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7315 on 467 degrees of freedom
##   (765 observations deleted due to missingness)
## Multiple R-squared:  0.1544, Adjusted R-squared:  0.1127
## F-statistic: 3.706 on 23 and 467 DF,  p-value: 3.147e-08
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.03432 -0.52626 -0.08259  0.44869  2.79109
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        14.1966     0.1570  90.422   <2e-16 ***
## r_0_50              0.4810     0.2217   2.170   0.0305 *
## same_tier          -0.6654     0.3485  -1.909   0.0568 .
## tier_diff_1        -0.6068     0.2871  -2.114   0.0350 *
## tier_diff_2        -0.1786     0.2382  -0.750   0.4539
## r_0_50:same_tier    0.8387     0.4870   1.722   0.0857 .
## r_0_50:tier_diff_1  0.6413     0.3761   1.705   0.0888 .
## r_0_50:tier_diff_2  0.2180     0.3409   0.640   0.5227
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7551 on 483 degrees of freedom
##   (765 observations deleted due to missingness)
## Multiple R-squared:  0.0682, Adjusted R-squared:  0.05469
## F-statistic:  5.05 on 7 and 483 DF,  p-value: 1.52e-05
```

all_corlengths_wave_mhhinc and urban_corlengths_wave_mhhinc do not have significant interaction terms, will not make a stargazer table for this model. Will make 3 stargazer tables, for these pairs of models: (1) all_corlengths_mhhinc and urban_corlengths_mhhinc (2) all_localcor_wave_mhhinc and urban_localcor_wave_mhhinc (3) all_localcor_mhhinc and urban_localcor_mhhinc

```r
# stargazer table for all_corlengths_mhhinc and urban_corlengths_mhhinc
# (1) is all
# (2) is urban
stargazer(
  c(mhhinc_lm_models_train[2], mhhinc_lm_models_train[6]),
  type = 'text',
  title = "Regression of Covid Cases on Correlation Length and Mhhinc Tier Difference"
)
```

```
##
## Regression of Covid Cases on Correlation Length and Mhhinc Tier Difference
## ======================================================================
##                                      Dependent variable:
##                            -------------------------------------------
##                                   log(next_week_marginal_cases)
##                                     (1)                    (2)
## ----------------------------------------------------------------------
## cor_lengths                       0.001***                0.003
##                                   (0.0002)               (0.002)
##
## same_tier                        -0.254**                 0.011
##                                   (0.123)                (0.099)
##
## tier_diff_1                      -0.309**                -0.030
##                                   (0.130)                (0.098)
##
## tier_diff_2                       -0.159                 -0.061
##                                   (0.132)                (0.138)
##
## cor_lengths:same_tier             0.0005*                -0.002
##                                   (0.0003)               (0.002)
##
## cor_lengths:tier_diff_1           0.001**                -0.002
##                                   (0.0003)               (0.002)
##
## cor_lengths:tier_diff_2           0.0003                  0.001
##                                   (0.0003)               (0.003)
##
## Constant                         14.608***              14.397***
##                                   (0.103)                (0.076)
##
## ----------------------------------------------------------------------
## Observations                       1,221                  1,238
## R2                                 0.161                  0.011
## Adjusted R2                        0.157                  0.005
## Residual Std. Error          0.737 (df = 1213)      0.785 (df = 1230)
## F Statistic             33.355*** (df = 7; 1213) 1.952* (df = 7; 1230)
## ======================================================================
## Note:                                      *p<0.1; **p<0.05; ***p<0.01
```

```r
# stargazer table for all_localcor_wave_mhhinc and urban_localcor_wave_mhhinc
# (1) is all
# (2) is urban
```

```r
stargazer(
  c(mhhinc_lm_models_train[3], mhhinc_lm_models_train[7]),
  type = 'text',
  title = "Regression of Covid Cases on Local Correlation,Mhhinc Tier Difference, and Wave"
)
```

```
##
## Regression of Covid Cases on Local Correlation,Mhhinc Tier Difference, and Wave
## ==================================================================================
##                                                  Dependent variable:
##                                   -------------------------------------------------
##                                            log(next_week_marginal_cases)
##                                              (1)                        (2)
## ----------------------------------------------------------------------------------
## r_0_50                                     1.657***                   1.026*
##                                            (0.549)                    (0.529)
##
## factor(wave)2                               0.575                      0.451
##                                            (0.415)                    (0.385)
##
## factor(wave)3                              0.916**                    1.512***
##                                            (0.395)                    (0.476)
##
## same_tier                                   -0.023                     -0.396
##                                            (0.346)                    (0.634)
##
## tier_diff_1                                 -0.351                     -1.244*
##                                            (0.356)                    (0.671)
##
## tier_diff_2                                 -0.059                     -0.171
##                                            (0.372)                    (0.646)
##
## r_0_50:factor(wave)2                        -0.537                     -0.318
##                                            (0.766)                    (0.605)
##
## r_0_50:factor(wave)3                        -0.888                    -1.750**
##                                            (0.761)                    (0.692)
##
## r_0_50:same_tier                             0.300                      0.320
##                                            (0.630)                    (0.950)
##
## factor(wave)2:same_tier                     -0.294                      0.015
##                                            (0.491)                    (1.027)
##
## factor(wave)3:same_tier                     -0.468                     -1.133
##                                            (0.444)                    (0.835)
##
## r_0_50:tier_diff_1                          1.229*                     1.390
##                                            (0.662)                    (0.953)
##
## factor(wave)2:tier_diff_1                   -0.315                      0.526
##                                            (0.498)                    (0.772)
##
```

```
## factor(wave)3:tier_diff_1                  -0.240                  0.746
##                                            (0.462)                (0.870)
##
## r_0_50:tier_diff_2                          0.570                 -0.039
##                                            (0.703)                (0.906)
##
## factor(wave)2:tier_diff_2                   0.176                 -0.543
##                                            (0.528)                (0.757)
##
## factor(wave)3:tier_diff_2                  -0.609                 -0.769
##                                            (0.484)                (0.766)
##
## r_0_50:factor(wave)2:same_tier              0.063                  0.017
##                                            (0.900)                (1.429)
##
## r_0_50:factor(wave)3:same_tier              0.526                  1.747
##                                            (0.863)                (1.210)
##
## r_0_50:factor(wave)2:tier_diff_1           -0.228                 -0.797
##                                            (0.925)                (1.073)
##
## r_0_50:factor(wave)3:tier_diff_1           -0.046                 -0.707
##                                            (0.909)                (1.182)
##
## r_0_50:factor(wave)2:tier_diff_2           -0.847                  0.870
##                                            (0.980)                (1.057)
##
## r_0_50:factor(wave)3:tier_diff_2            0.739                  1.525
##                                            (0.960)                (1.079)
##
## Constant                                  13.821***              13.668***
##                                            (0.312)                (0.327)
##
## -----------------------------------------------------------------------------------
## Observations                                1,240                   491
## R2                                          0.223                  0.154
## Adjusted R2                                 0.209                  0.113
## Residual Std. Error                   0.711 (df = 1216)        0.732 (df = 467)
## F Statistic                       15.210*** (df = 23; 1216) 3.706*** (df = 23; 467)
## ===================================================================================
## Note:                                                 *p<0.1; **p<0.05; ***p<0.01
```

```r
# all_localcor_mhhinc and urban_localcor_mhhinc
# (1) is all
# (2) is urban
stargazer(
  c(mhhinc_lm_models[4], mhhinc_lm_models[8]),
  type = 'text',
  title = "Regression of Covid Cases on Local Correlation and Mhhinc Tier Difference"
)
```

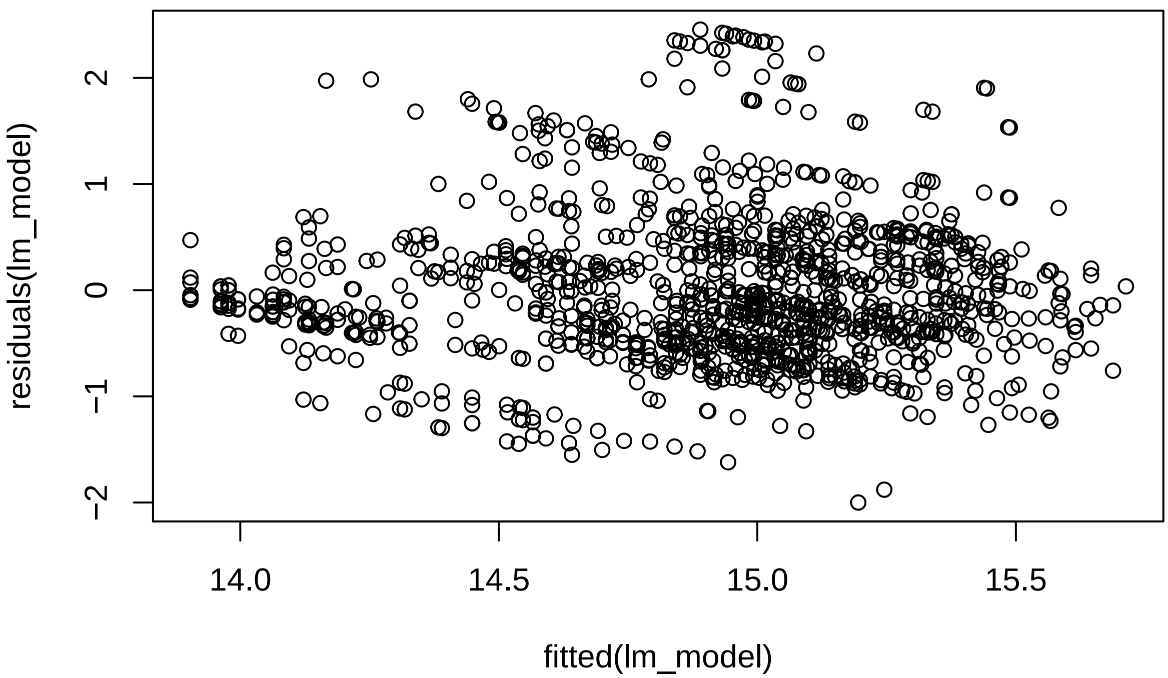
```
##
## Regression of Covid Cases on Local Correlation and Mhhinc Tier Difference
## ==================================================================
```

```
##                                   Dependent variable:
##                     ----------------------------------------------
##                             log(next_week_marginal_cases)
##                              (1)                      (2)
## ------------------------------------------------------------------
## r_0_50                    0.889***                 0.565***
##                           (0.264)                  (0.197)
##
## same_tier                -0.432***                 -0.384
##                           (0.158)                  (0.320)
##
## tier_diff_1              -0.591***                 -0.487*
##                           (0.161)                  (0.265)
##
## tier_diff_2               -0.383**                 -0.018
##                           (0.173)                  (0.217)
##
## r_0_50:same_tier           0.780**                  0.507
##                           (0.304)                  (0.449)
##
## r_0_50:tier_diff_1        1.194***                  0.547
##                           (0.313)                  (0.344)
##
## r_0_50:tier_diff_2         0.757**                  0.033
##                           (0.334)                  (0.310)
##
## Constant                 14.511***                14.113***
##                           (0.136)                  (0.141)
##
## ------------------------------------------------------------------
## Observations               1,550                     620
## R2                         0.186                    0.057
## Adjusted R2                0.182                    0.046
## Residual Std. Error   0.719 (df = 1542)        0.764 (df = 612)
## F Statistic         50.340*** (df = 7; 1542) 5.294*** (df = 7; 612)
## ==================================================================
## Note:                                  *p<0.1; **p<0.05; ***p<0.01
```

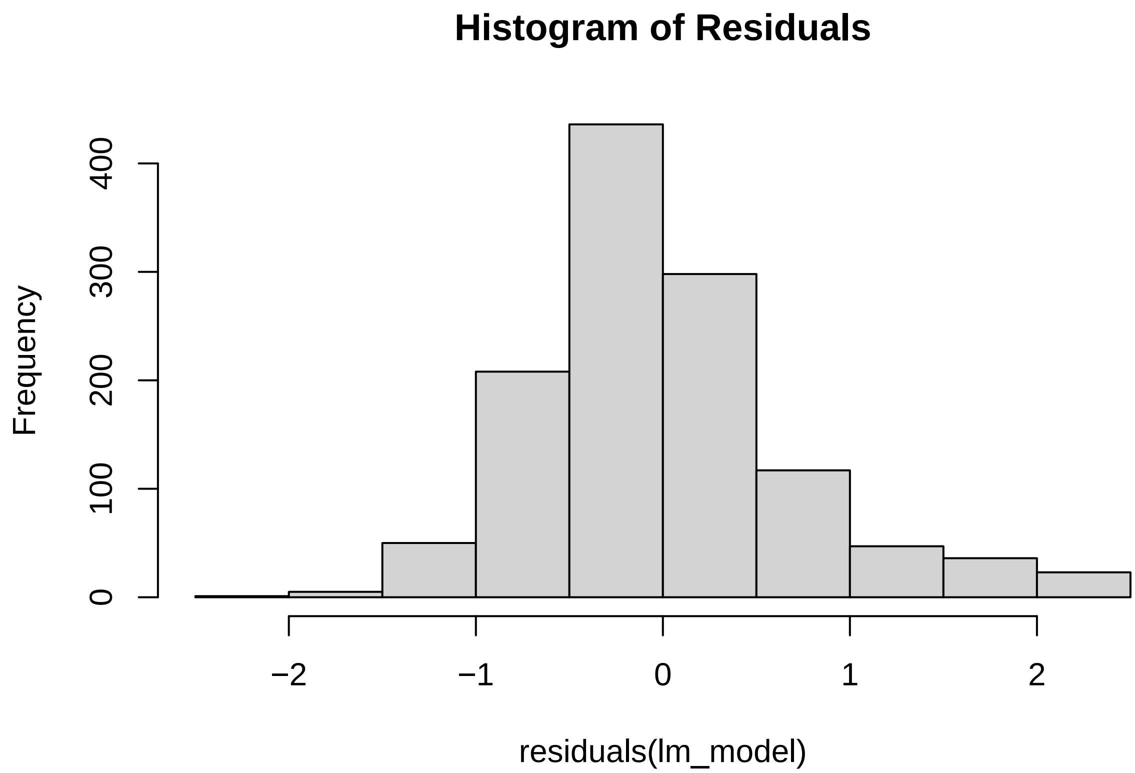
Check conditions of Normality of Residuals, Homoskedasticity,Linearity , No Perfect Multicollinearity, Independence of Residuals

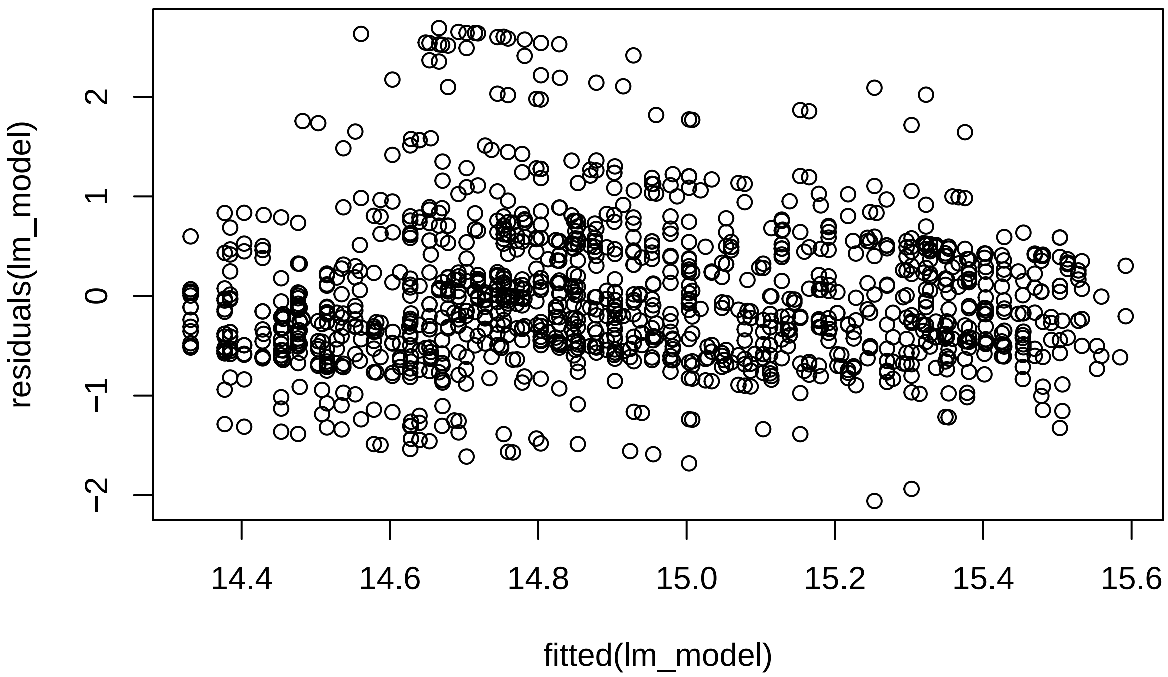
```r
for (lm_model in mhhinc_lm_models_train) {
# 1. Linearity Check (Visual Inspection)
plot(residuals(lm_model) ~ fitted(lm_model))

# 2. Independence Check
# Durbin-Watson test
print(durbinWatsonTest(lm_model)[3])

# 3. Homoscedasticity Check
# Breusch-Pagan test for Heteroscedasticity
print(bptest(lm_model)$p.value)
```

```r
# 4. Normality of Residuals Check (Visual Inspection)
hist(residuals(lm_model), main = "Histogram of Residuals")
}
```

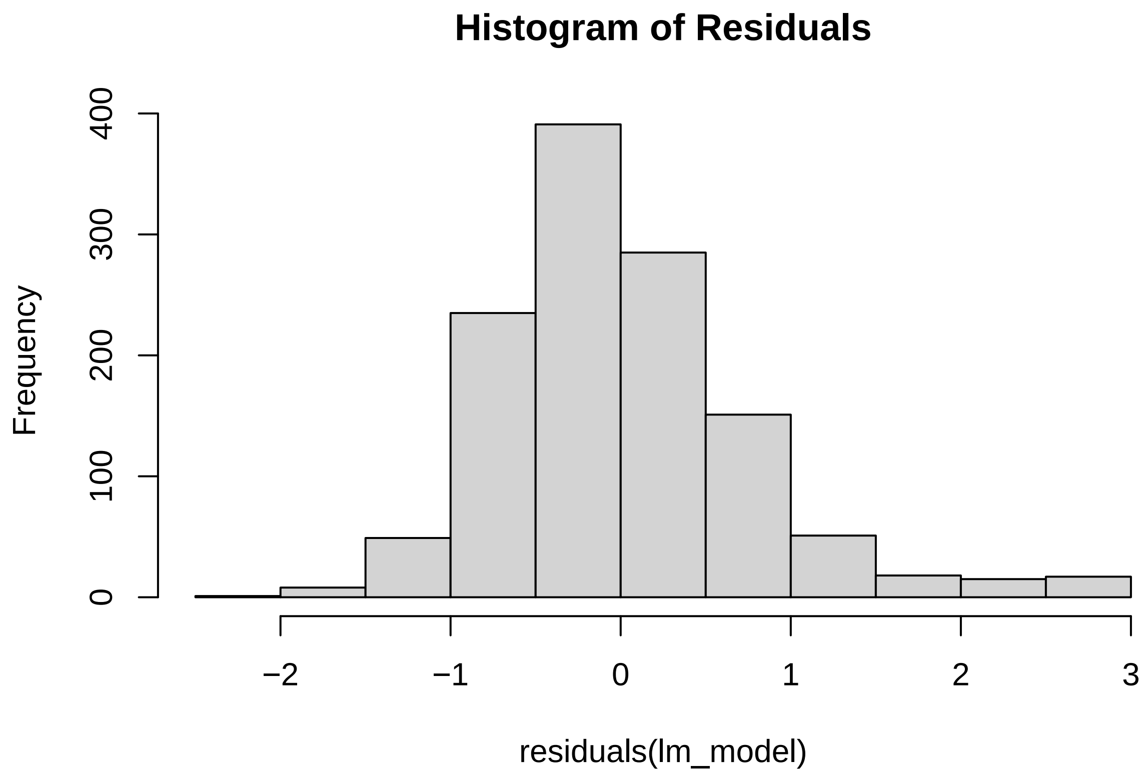


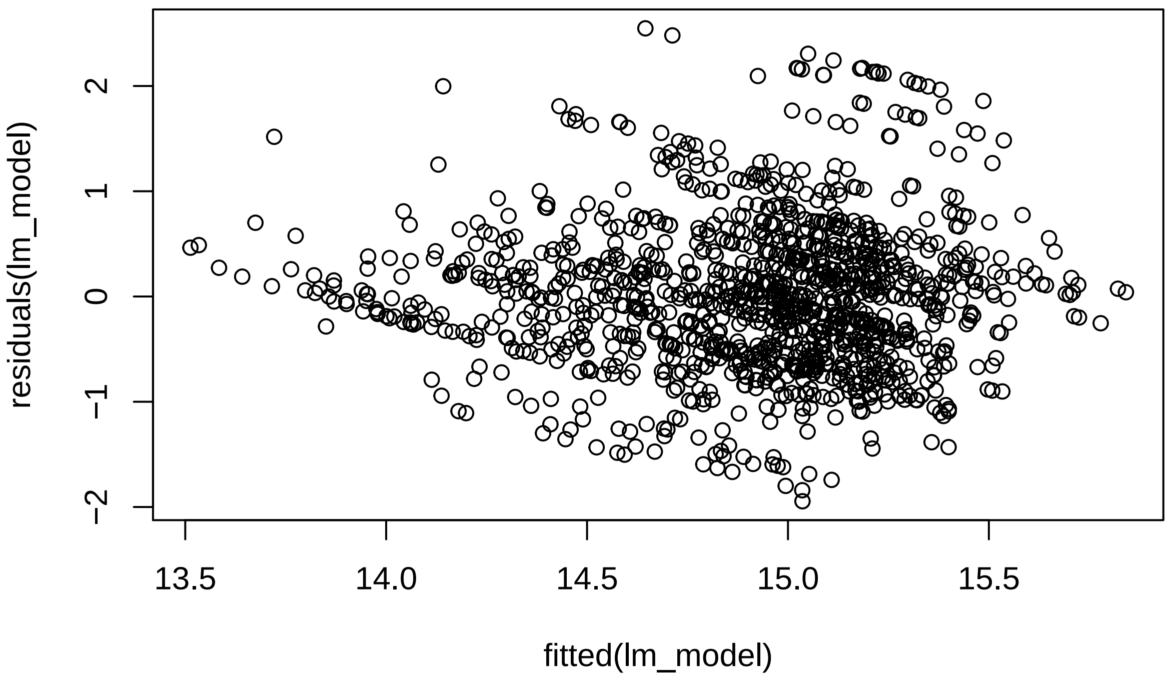

```
## $p
## [1] 0.586
##
##           BP
## 3.008527e-05
```


Histogram of Residuals
Frequency
0
100
200
300
400
−2
−1
0
1
2
residuals(lm_model)

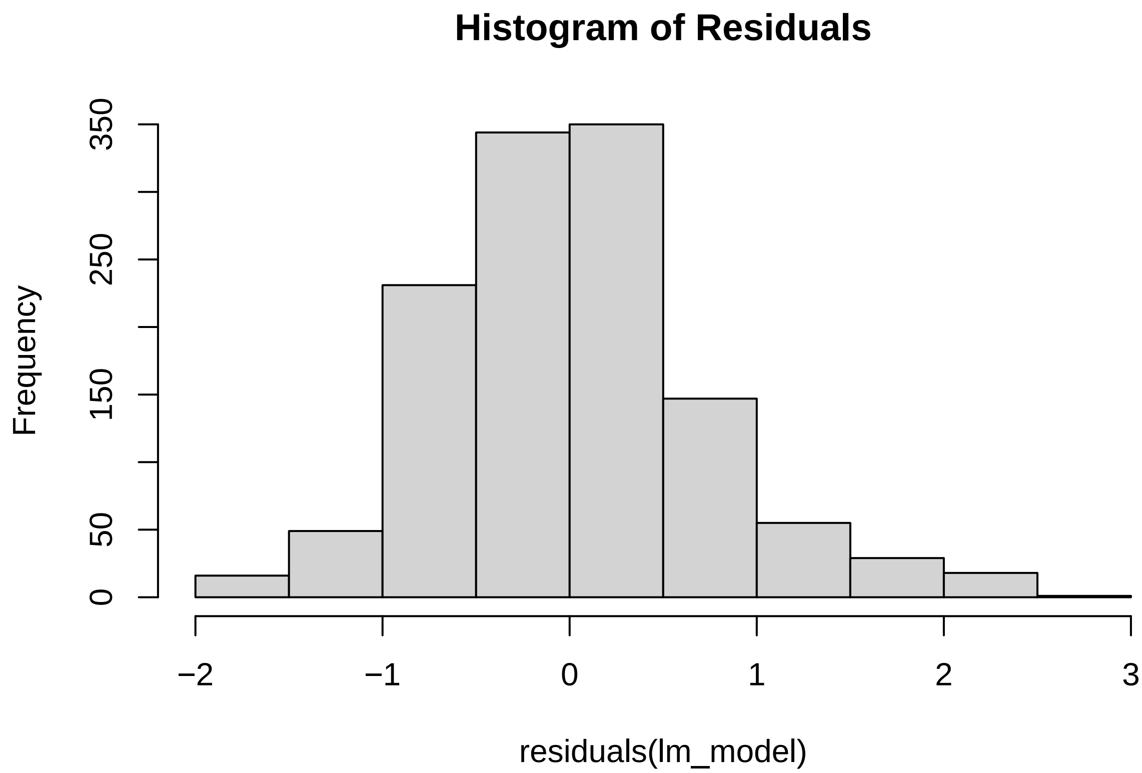


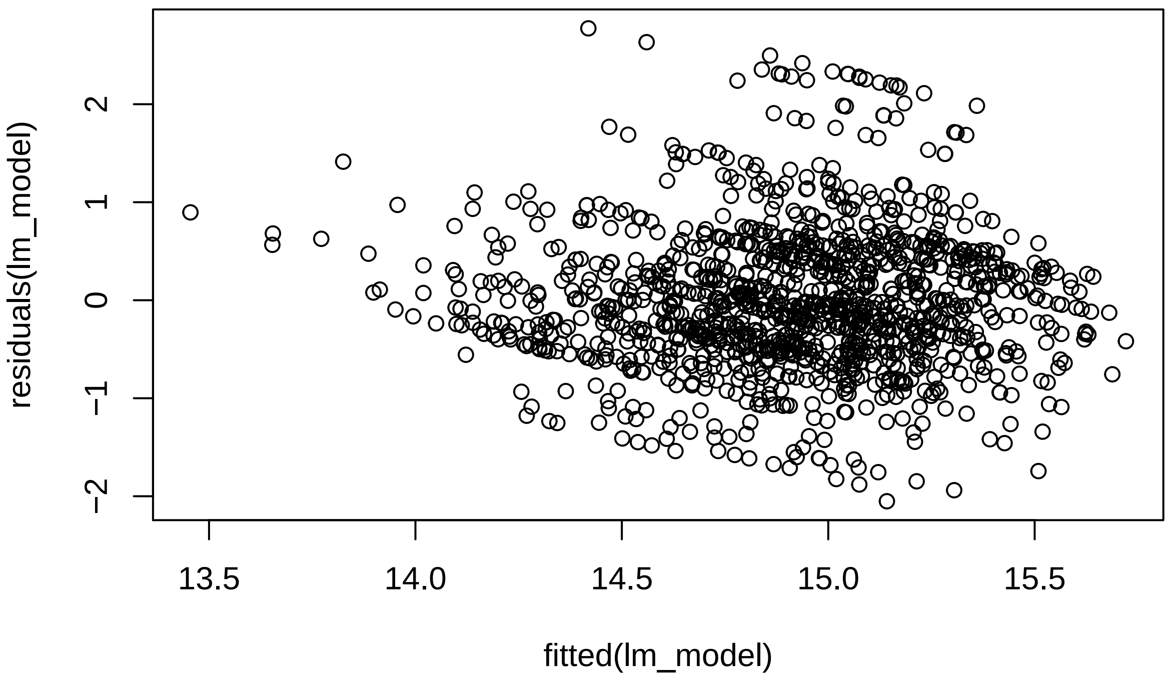

```
## $p
## [1] 0.578
##
##         BP
## 0.02118914
```


Histogram of Residuals
Frequency
0
100
200
300
400
−2
−1
0
1
2
3
residuals(lm_model)

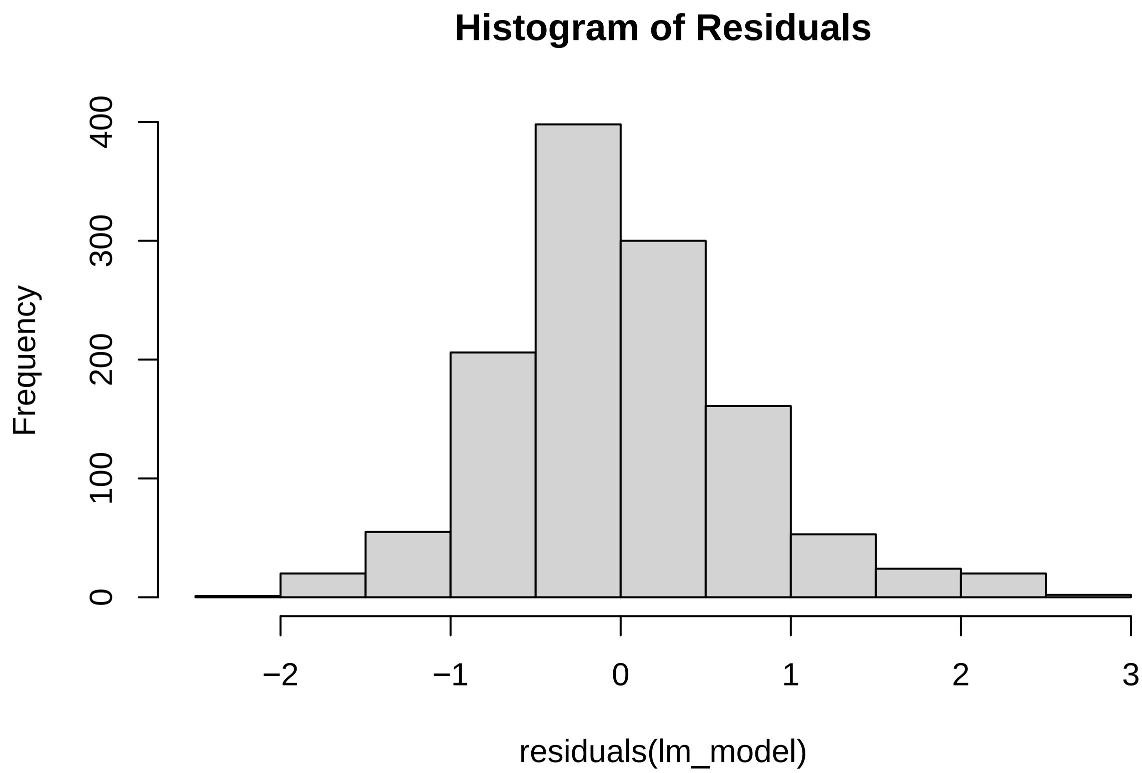


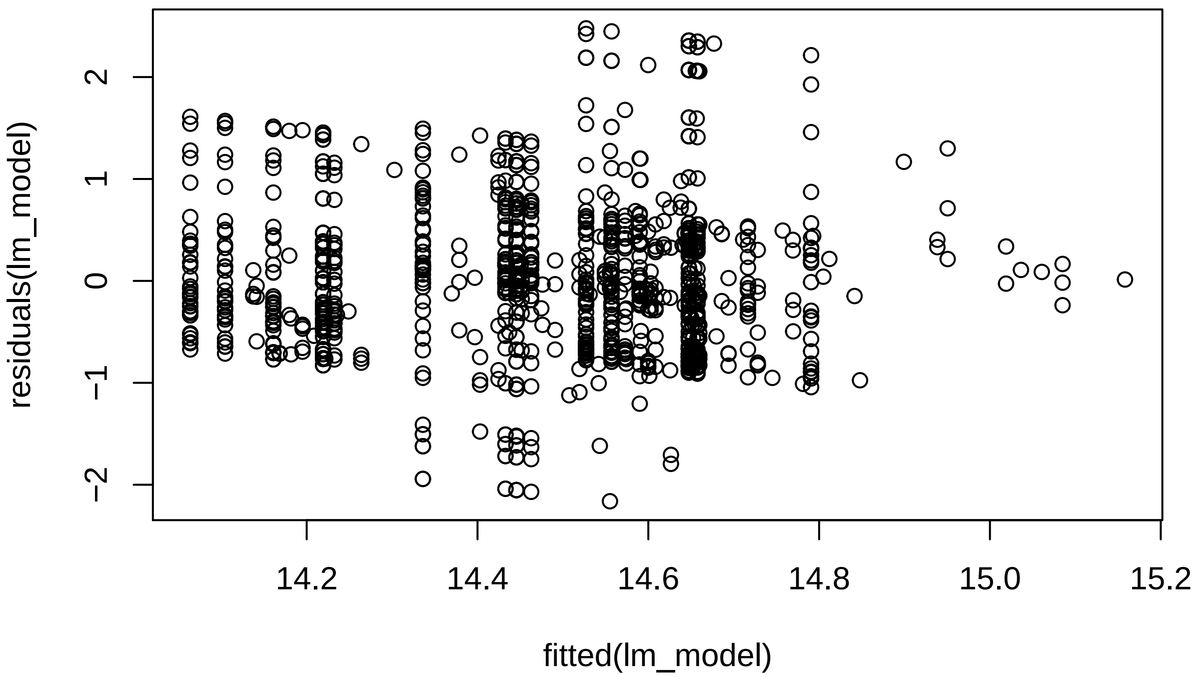

```
## $p
## [1] 0.3
##
##           BP
## 5.11232e-09
```


Histogram of Residuals
Frequency
0
50
150
250
350
−2
−1
0
1
2
3
residuals(lm_model)

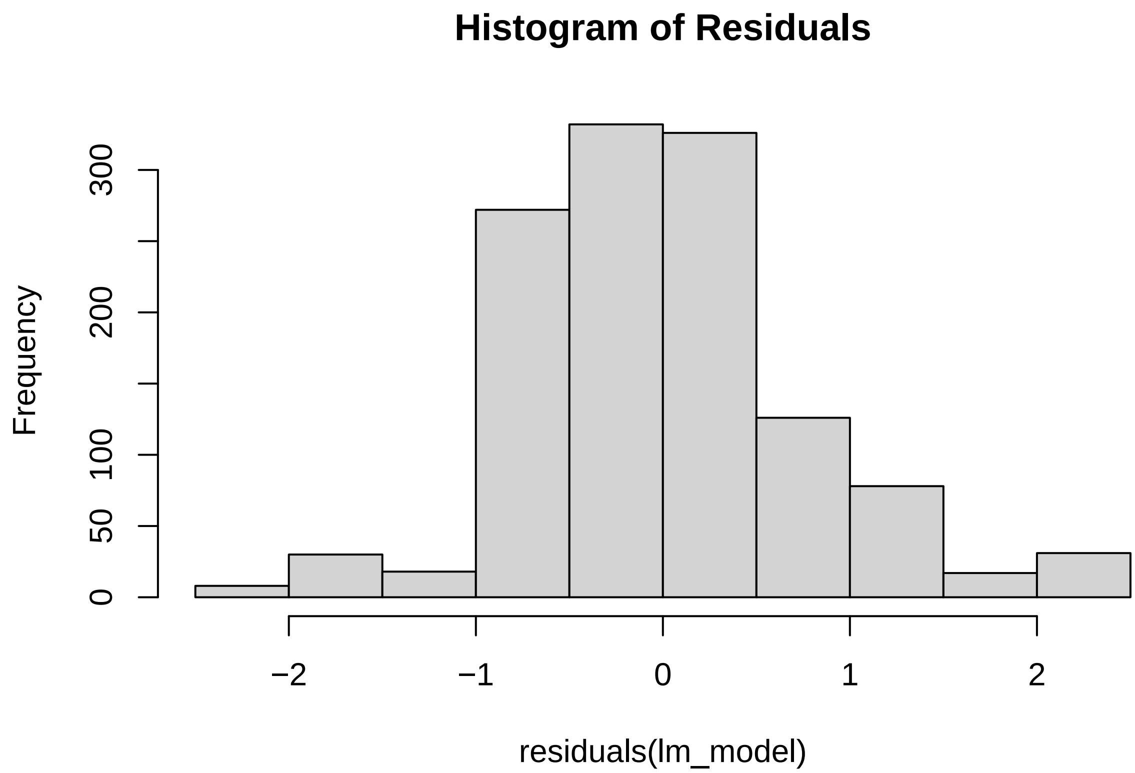


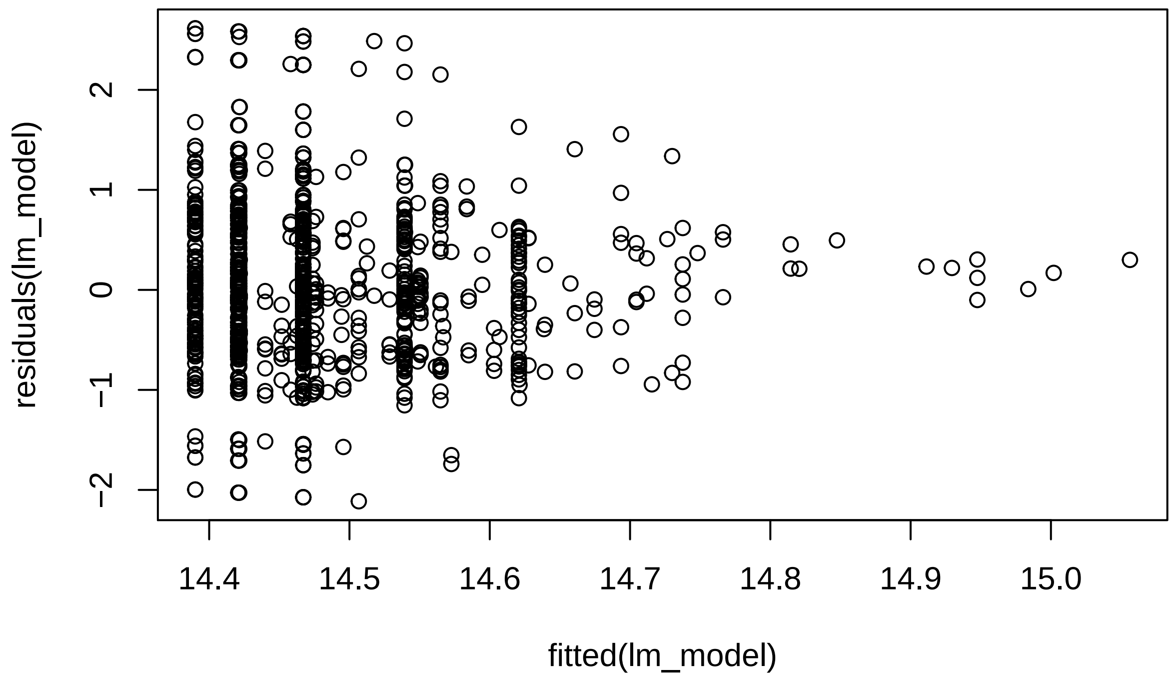

```
## $p
## [1] 0.454
##
##        BP
## 0.5452983
```


Histogram of Residuals
Frequency
0
100
200
300
400
−2
−1
0
1
2
3
residuals(lm_model)

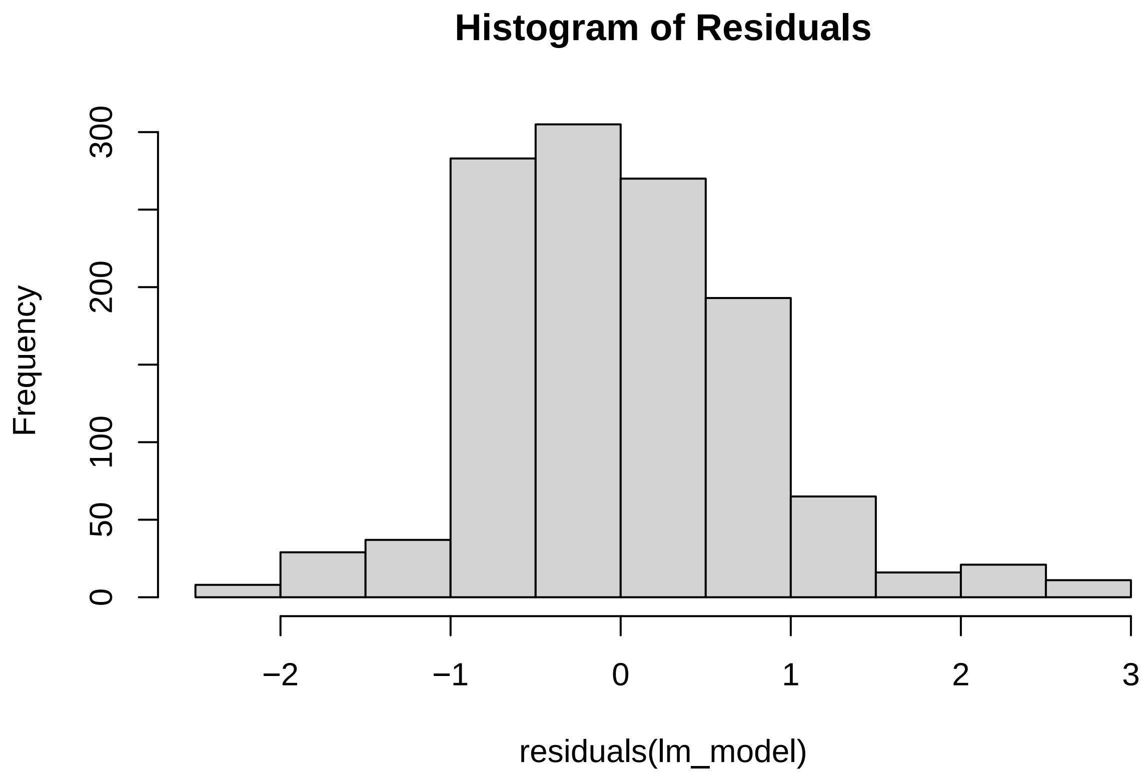


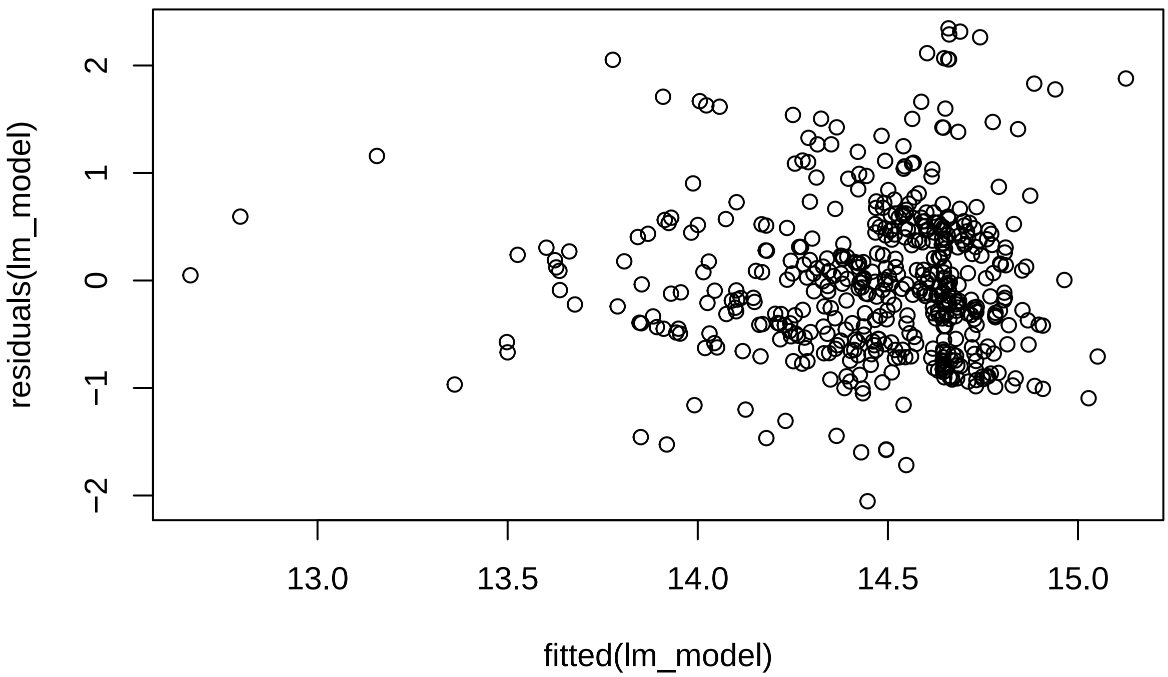

```
## $p
## [1] 0.588
##
##           BP
## 0.008632414
```


Histogram of Residuals
Frequency
0
50
100
200
300
−2
−1
0
1
2
residuals(lm_model)

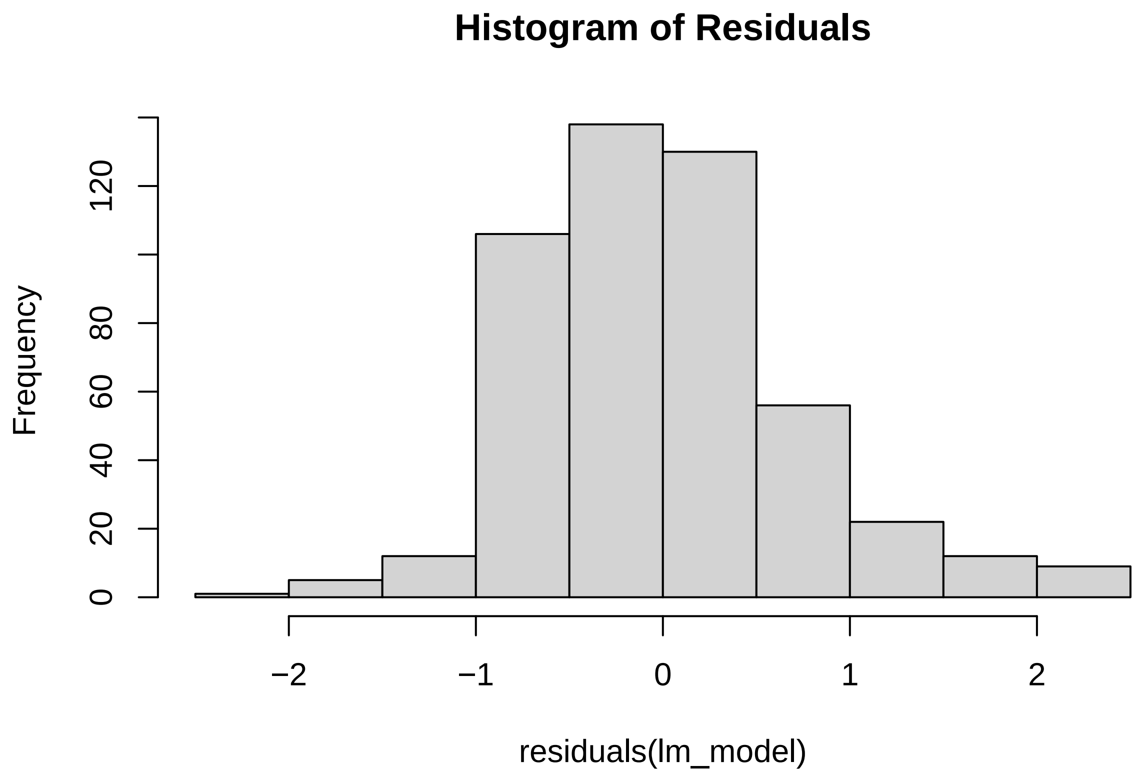



```
## $p
## [1] 0.858
##
##        BP
## 0.0517411
```

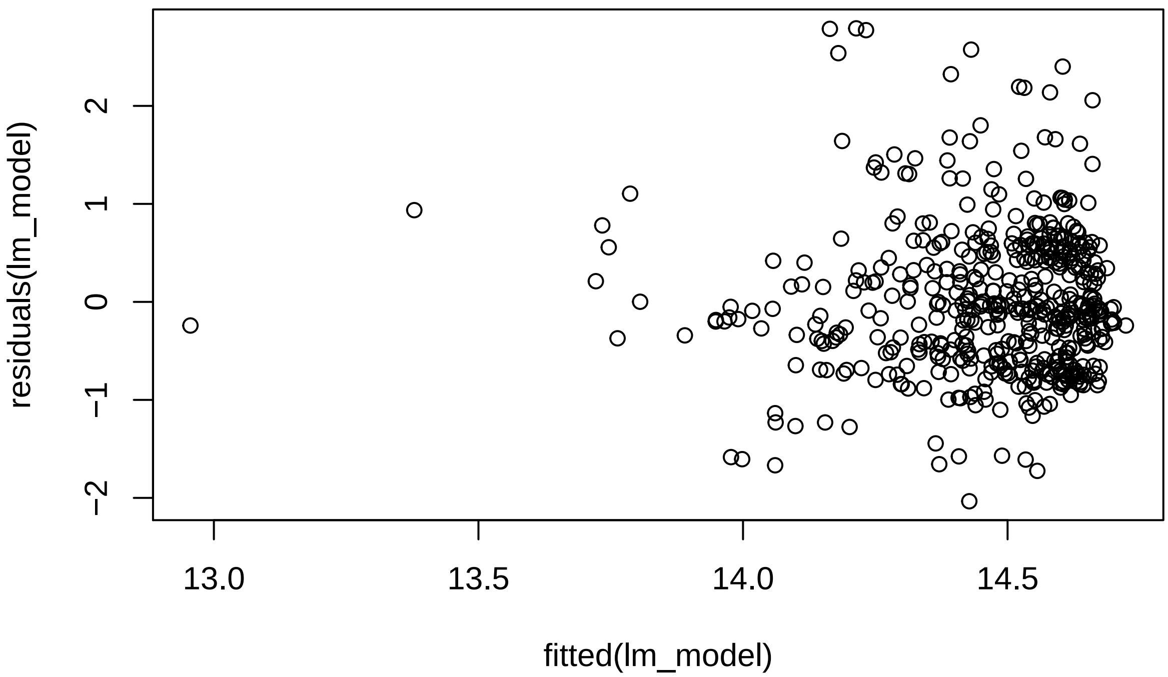

Histogram of Residuals
Frequency
0
50
100
200
300
−2
−1
0
1
2
3
residuals(lm_model)



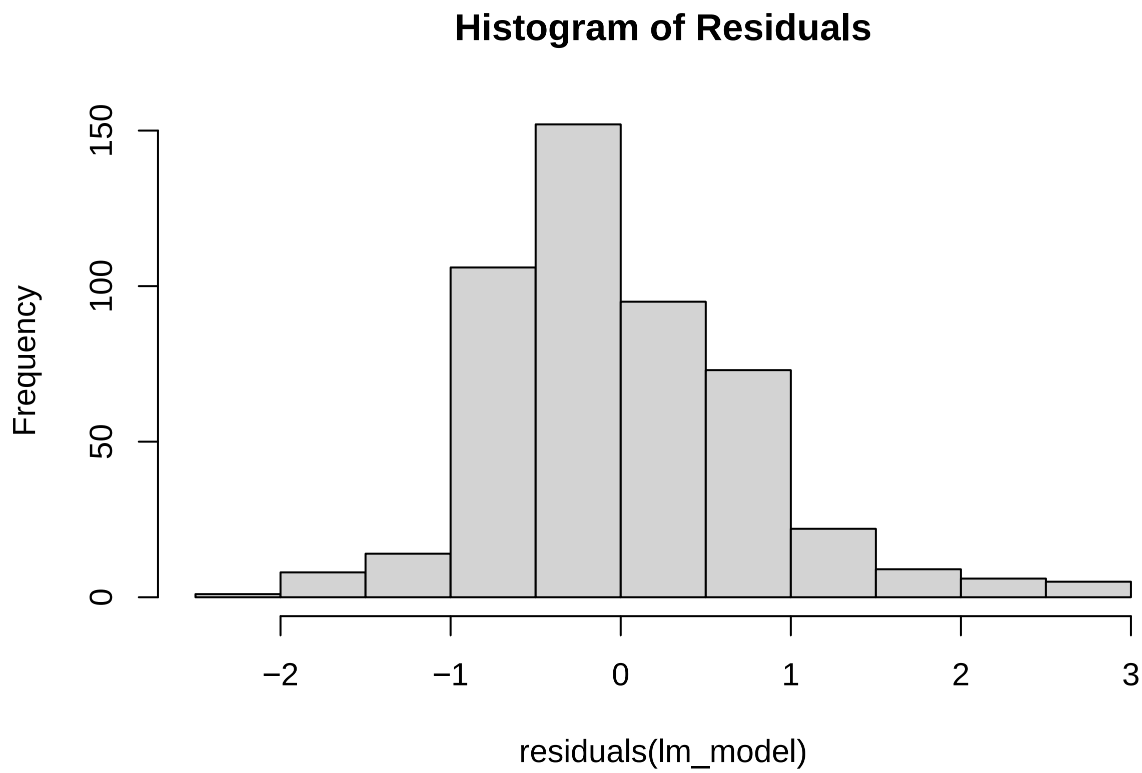

```
## $p
## [1] 0.01
##
##            BP
## 3.617043e-06
```



Histogram of Residuals
Frequency
0
20
40
60
80
120
−2
−1
0
1
2
residuals(lm_model)




```
## $p
## [1] 0.074
##
##        BP
## 0.1017213
```




```r
# 5. No Perfect Multicollinearity Check
# Variance Inflation Factors (VIF)
vif(lm(log(next_week_marginal_cases) ~  cor_lengths + factor(wave) + factor(mhhinc_tier) , data = all_ml
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## cor_lengths         1.149220  1        1.072017
## factor(wave)        1.069424  2        1.016922
## factor(mhhinc_tier) 1.080817  9        1.004327
```

```r
vif(lm(log(next_week_marginal_cases) ~  cor_lengths + factor(wave) + factor(mhhinc_tier) , data = urban_
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## cor_lengths         2.105017  1        1.450868
## factor(wave)        1.004114  2        1.001027
## factor(mhhinc_tier) 2.100903  9        1.042105
```

```r
# Conditions not passed:
## Heteroskedasticity in all_corlengths_wave_mhhinc_train, all_localcor_wave_mhhinc_train, all_localcor_
## Non-linearity in urban_localcor_wave_mhhinc, urban_localcor_mhhinc
names(mhhinc_lm_models_train)
```

```
## [1] "all_corlengths_wave_mhhinc_train"   "all_corlengths_mhhinc_train"
## [3] "all_localcor_wave_mhhinc_train"     "all_localcor_mhhinc_train"
## [5] "urban_corlengths_wave_mhhinc_train" "urban_corlengths_mhhinc_train"
## [7] "urban_localcor_wave_mhhinc_train"   "urban_localcor_mhhinc_train"
```

Interpretations

```
# Initialize values
predictor <- c("cor_lengths","r_0_50")
waves <- c("wave 1","wave 2", "wave 3")
tier_diff <- c("same_tier", "tier_diff_1", "tier_diff_2", "tier_diff_3")
tier_descr <- c("are the same", "differ by 1", "differ by 2", "differ by 3")
full_coef = c()
full_coef1 = c()
significant = c()
significant1 = c()
i=1
j=1

# Iterate over Wave and Tier diff dummies
for (model in 1:length(mhhinc_lm_models_train)) {

  # Get model stats
  lm_model = mhhinc_lm_models_train[[model]]
  coefficients <- coef(lm_model)
  coef_names <- names(coef(lm_model))
  # se <- summary(lm_model)$coefficients[, "Std. Error"] use robust SE's
  se <- sqrt(diag(vcovHC(lm_model)))
  se_named <- rep(NA, length(coef_names))
  se_named[names(se)] <- se
  st_errors <- se_named[coef_names]

  # Based on model, decide if urban or all county df
  if (model <=4) {
    county = "all counties"
    df = all_mhhinc_weekly_spatial_metrics}
  else {
    county = "urban counties"
    df = urban_mhhinc_weekly_spatial_metrics}

  # Based on model, choose cor_length or local_cor as predictor
  if (model %in% c(1,2,5,6)) {pred = predictor[1]}
  else {pred = predictor[2]}

  # Wave dummies not included for even models
  if (model %% 2 == 0) {
    wave = "every wave"
    for (m in 1:length(tier_diff)) {
      tierdiff <- tier_diff[m]
      if (m==4) {interaction = 0
      se_total = st_errors[2]}
      else {
      interaction = coefficients[5+m]
      se_total = sqrt(sum((st_errors[5+m])^2 + (st_errors[2])^2))
      }
      # Store coefficient
      full_coef1[j] = save_coefficient(interaction)
```

```r
      # Print interpretation and significance
      significant1[j] <- is_significant(as.numeric(full_coef1[j]), as.numeric((se_total)))
      print_interpretation(full_coef1[j], wave, region="whole US", tier_descr[m], significant1[j], pred

  # Wave dummies with mhhinc tier dummies
  else {
  # Iterate over each wave
  for (w in 1:length(waves)) {
    wave <- waves[w]
    # Iterate of each tier difference
    for (m in 1:length(tier_diff)) {
      tierdiff <- tier_diff[m]
      if (wave == "Wave 1") {
        partial_interaction_w <- 0
        se_partial_w <- 0
      } else {
        partial_interaction_w <- coefficients[(7 + w)]
        se_partial_w <- st_errors[(7 + w)]
      }
      if (tierdiff == "tier_diff_3") {
        partial_interaction_m <- 0
        se_partial_m <- 0
      } else {
        partial_interaction_m <- coefficients[(3*m+7)]
        se_partial_m <- st_errors[(3*m+7)]
      }
      if (tierdiff != "tier_diff_3" & wave != "Wave 1") {
        full_interaction_wm <- coefficients[paste0(pred,":factor(wave)", w, ":", tierdiff)]
        se_full_wm <- st_errors[paste0(pred,":factor(wave)", w, ":", tierdiff)]
      } else {
        full_interaction_wm <- 0
        se_full_wm <- 0
      }
    # Calculate total standard error
    se_total <- sqrt(se_partial_w^2 + se_partial_m^2 + se_full_wm^2 + st_errors[2]^2)
    # Store coefficient information
    interaction <- partial_interaction_w + partial_interaction_m + full_interaction_wm

    # Store coefficient
    full_coef[i] = save_coefficient(interaction)

    # Print interpretation and significance
    significant[i] <- is_significant(as.numeric(full_coef[i]), as.numeric((se_total)))
    print_interpretation(full_coef[i], wave, region="whole US", tier_descr[m], significant[i], pred, co
    i <- i + 1
    }}}
}
```

```
## - For all counties across wave 3 in the whole US where the median household income tiers differ by 3
##   and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##   corresponds to a 38.12% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are the
##   and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
```

```
##  corresponds to a 30.5% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers differ b
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 32.05% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers differ b
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 27.08% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers differ b
##  and where mask usage in both counties are disregarded a 1 sd increase correlation length (244 km)
##  corresponds to a 19.29% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are the
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 29.66% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers differ b
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 38.91% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers differ b
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 30.48% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers differ b
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 17.72% increase in cases in the following week. Significant? TRUE
## - For urban counties across every wave in the whole US where the median household income tiers are th
##  and where mask usage in both counties are disregarded a 1 sd increase local correlation (0.18)
##  corresponds to a 23.75% increase in cases in the following week. Significant? TRUE
```

Significant differential effects among "equality level." Spread is greater among counties of similar socioeconomic leve. Strong results across every wave but especially in wave 3. Effects are not as discernible among Urban counties. Both correlation length and local correlation are effective metrics for predicting covid to discern socioeconimic effects.

Mask + Wave Models

Prepare Model Combinations

```r
# Data sets: (1) all counties & (2) urban counties
datasets = list(all_mask_weekly_spatial_metrics)
df_names = c("all")

# Predictor: (1) correlation length & (2) local correlation
predictor = c("cor_lengths", "r_0_50")
predictor_names = c("corlengths","localcor")

# Covariates (1) mhhinc & (2) mhhinc + wave
covariates = list(c("*factor(wave)*mask_11","*factor(wave)*mask_22","*factor(wave)*mask_33","*factor(wav
                  c("*mask_11","*mask_22","*mask_33", "*mask_44"))
var_names = c("wave_mask","mask")
```

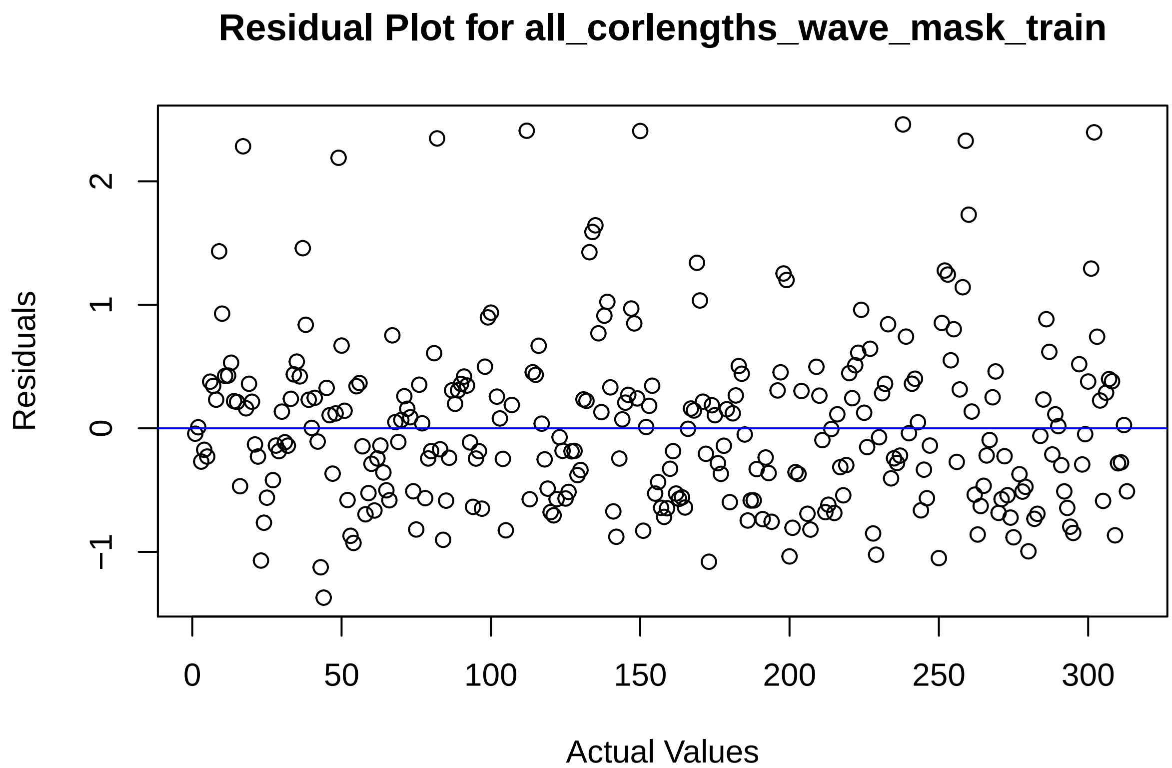
Generate Models

```
# Create an empty list to store the models
mask_lm_models <- list()

# Iterate of every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df = datasets[d][[1]]
  for (p in 1:length(predictor)) {
    for (c in 1:length(covariates)) {
      interactions = c()
      for (i in 1:length(covariates[[c]])) {
        interactions[i] <- paste(predictor[p], covariates[[c]][i])}

      # Create the formula
      lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), col

      # Fit the linear model
      lm_model <- lm(lm_formula, data = df)

      # Generate a name for the model object
      model_name <- paste(df_names[d],"_", predictor_names[p], "_", var_names[c], sep = "")

      # Save the model to the list
     mask_lm_models[[model_name]] <- lm_model
    }}}
```

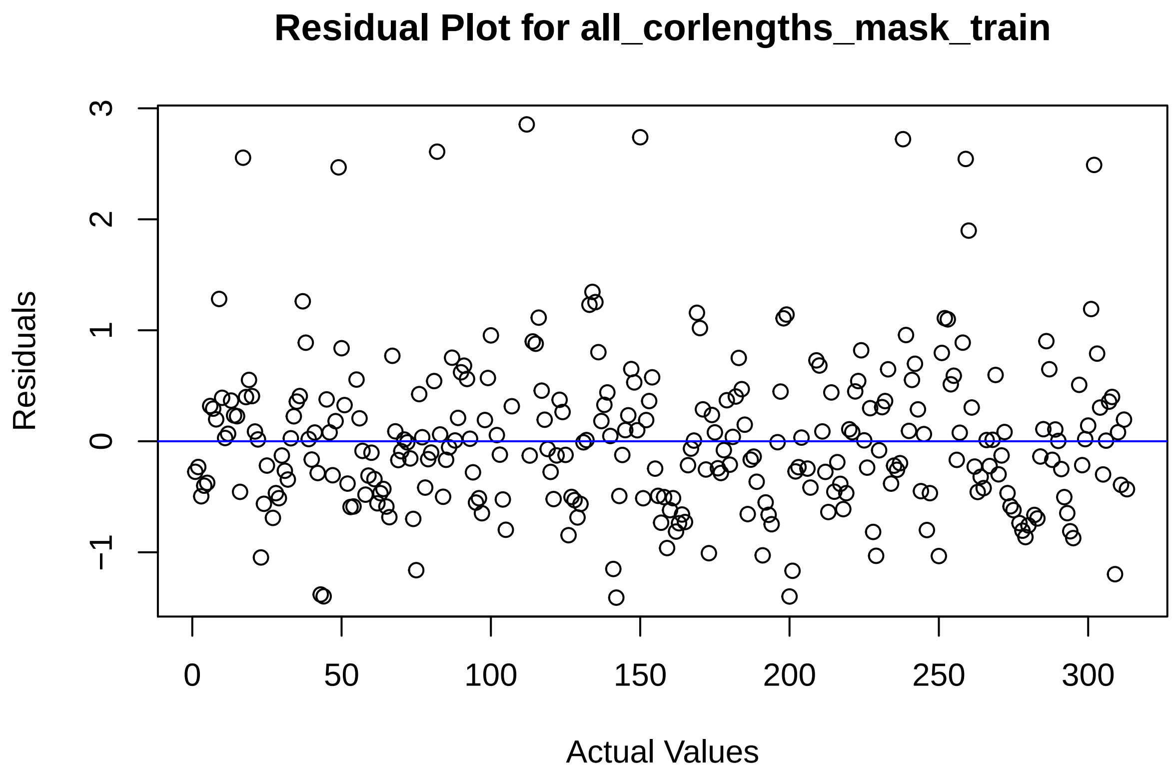
Training/Testing Models

```
# Set seed for reproducibility
set.seed(1)

# Create an empty list to store the models
mask_lm_models_train <- list()

# Iterate over every combination of data, predictor, and covariates
for (d in 1:length(datasets)) {
  df <- datasets[[d]]

  # Create indices for train-test split
  n_obs <- nrow(df)
  train_indices <- sample(1:n_obs, size = round(train_ratio * n_obs), replace = FALSE)
  test_indices <- setdiff(1:n_obs, train_indices)

  # Split the data into train and test sets
  train_df <- df[train_indices, ]
  test_df <- df[test_indices, ]

  for (p in 1:length(predictor)) {
    for (c in 1:length(covariates)) {
      interactions <- c()
```

```
      for (i in 1:length(covariates[[c]])) {
        interactions[i] <- paste(predictor[p], covariates[[c]][i])
      }

      # Create the formula & fit linear model on the training data
      lm_formula <- as.formula(paste("log(next_week_marginal_cases) ~", paste(unlist(interactions), col
      lm_model_train <- lm(lm_formula, data = train_df)

      # Name model and save to list
      model_name_train <- paste(df_names[d], "_", predictor_names[p], "_", var_names[c], "_train", sep =
      mask_lm_models_train[[model_name_train]] <- lm_model_train

      # Apply the trained model to the test data and make predictions
      test_predictions <- predict(lm_model_train, newdata = test_df)

      # Calculate residuals using predictions from the test dataset
      residuals <- log(test_df$next_week_marginal_cases) - test_predictions

      # Create residual plot
      plot(1:length(residuals), residuals,
           main = paste("Residual Plot for", model_name_train),
           xlab = "Actual Values",
           ylab = "Residuals")
      abline(h = 0, col = "blue")

      # Check if the mean of residuals is less than the standard deviation of the data
      print(mean(abs(residuals), na.rm = TRUE) < sd(log(test_df$next_week_marginal_cases), na.rm = TRUE
    }
  }
}
```

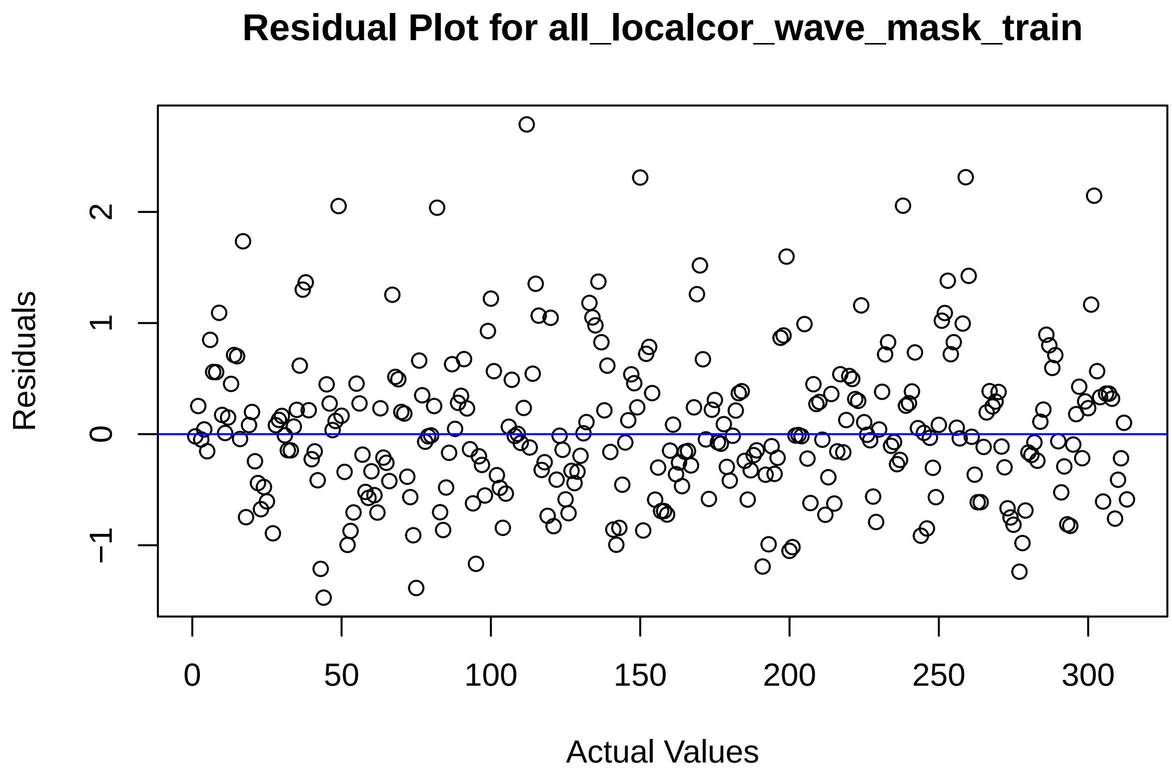

Residual Plot for all_corlengths_wave_mask_train
Residuals
Actual Values
2
1
0
−1
0
50
100
150
200
250
300

```
## [1] TRUE
```

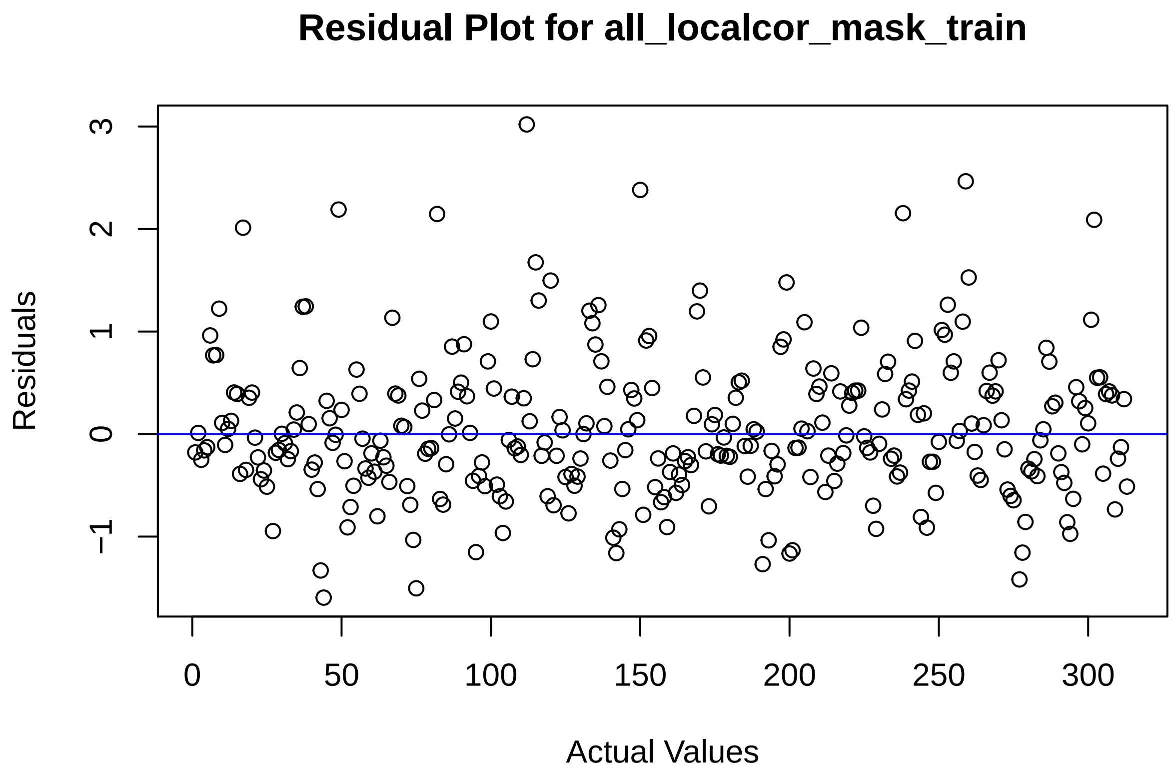


Residual Plot for all_corlengths_mask_train
Residuals
Actual Values
3
2
1
0
−1
0
50
100
150
200
250
300


```
## [1] TRUE
```


Residual Plot for all_localcor_wave_mask_train
Residuals
Actual Values
0
50
100
150
200
250
300
-1
0
1
2

```
## [1] TRUE
```




```
## [1] TRUE
```

```r
# Print the names of the models
names(mask_lm_models_train)
```

```
## [1] "all_corlengths_wave_mask_train" "all_corlengths_mask_train"
## [3] "all_localcor_wave_mask_train"   "all_localcor_mask_train"
```

View Model Summaries

```r
for (m in 1:length(mask_lm_models_train)){
  print(summary(mask_lm_models_train[[m]]))
}
```

```
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q  Median       3Q      Max
## -1.7159 -0.5015 -0.1307  0.3712  2.4743
##
## Coefficients:
```

```
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        1.413e+01  8.132e-02 173.690  < 2e-16 ***
## cor_lengths                        1.463e-03  1.683e-04   8.691  < 2e-16 ***
## factor(wave)2                     -3.820e-02  1.249e-01  -0.306    0.760
## factor(wave)3                      8.022e-01  1.093e-01   7.339 4.05e-13 ***
## mask_11                           -2.089e-01  2.461e-01  -0.849    0.396
## mask_22                           -2.308e-01  2.427e-01  -0.951    0.342
## mask_33                           -2.454e-01  2.306e-01  -1.064    0.287
## mask_44                           -5.029e-01  3.265e-01  -1.540    0.124
## cor_lengths:factor(wave)2          1.770e-04  2.323e-04   0.762    0.446
## cor_lengths:factor(wave)3         -1.149e-03  2.410e-04  -4.770 2.08e-06 ***
## cor_lengths:mask_11                6.168e-04  6.118e-04   1.008    0.314
## factor(wave)2:mask_11             -3.763e-01  3.808e-01  -0.988    0.323
## factor(wave)3:mask_11             -1.689e-01  3.242e-01  -0.521    0.603
## cor_lengths:mask_22                3.989e-04  4.720e-04   0.845    0.398
## factor(wave)2:mask_22             -2.623e-01  4.170e-01  -0.629    0.530
## factor(wave)3:mask_22              1.045e-01  3.482e-01   0.300    0.764
## cor_lengths:mask_33                4.865e-04  5.061e-04   0.961    0.337
## factor(wave)2:mask_33             -2.467e-02  3.640e-01  -0.068    0.946
## factor(wave)3:mask_33             -4.004e-02  3.077e-01  -0.130    0.896
## cor_lengths:mask_44                9.451e-04  6.442e-04   1.467    0.143
## factor(wave)2:mask_44              6.967e-01  4.648e-01   1.499    0.134
## factor(wave)3:mask_44              3.349e-01  4.031e-01   0.831    0.406
## cor_lengths:factor(wave)2:mask_11  4.575e-04  7.942e-04   0.576    0.565
## cor_lengths:factor(wave)3:mask_11  4.542e-04  8.109e-04   0.560    0.576
## cor_lengths:factor(wave)2:mask_22  3.058e-04  7.234e-04   0.423    0.673
## cor_lengths:factor(wave)3:mask_22 -9.487e-05  7.544e-04  -0.126    0.900
## cor_lengths:factor(wave)2:mask_33 -2.149e-05  7.089e-04  -0.030    0.976
## cor_lengths:factor(wave)3:mask_33  1.766e-04  6.953e-04   0.254    0.800
## cor_lengths:factor(wave)2:mask_44 -1.200e-03  8.805e-04  -1.363    0.173
## cor_lengths:factor(wave)3:mask_44 -8.094e-04  8.057e-04  -1.004    0.315
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7072 on 1156 degrees of freedom
##   (70 observations deleted due to missingness)
## Multiple R-squared:  0.2534, Adjusted R-squared:  0.2347
## F-statistic: 13.53 on 29 and 1156 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7155 -0.4782 -0.1106  0.3650  2.8666
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.446e+01  4.840e-02 298.780   <2e-16 ***
## cor_lengths          1.095e-03  9.721e-05  11.262   <2e-16 ***
## mask_11             -2.993e-01  1.412e-01  -2.120   0.0342 *
## mask_22             -1.966e-01  1.531e-01  -1.284   0.1993
## mask_33             -2.340e-01  1.378e-01  -1.698   0.0897 .
```

```
## mask_44             -1.122e-01  1.723e-01  -0.651   0.5152
## cor_lengths:mask_11  7.325e-04  3.041e-04   2.409   0.0162 *
## cor_lengths:mask_22  3.683e-04  3.005e-04   1.226   0.2205
## cor_lengths:mask_33  4.669e-04  2.876e-04   1.624   0.1047
## cor_lengths:mask_44  1.064e-04  3.358e-04   0.317   0.7513
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7392 on 1176 degrees of freedom
##   (70 observations deleted due to missingness)
## Multiple R-squared:  0.1702, Adjusted R-squared:  0.1639
## F-statistic: 26.81 on 9 and 1176 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.85432 -0.47758 -0.06701  0.37475  2.69734
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  13.820630   0.109822 125.845  < 2e-16 ***
## r_0_50                        2.173524   0.245104   8.868  < 2e-16 ***
## factor(wave)2                 0.176816   0.178713   0.989  0.32267
## factor(wave)3                 0.552997   0.159624   3.464  0.00055 ***
## mask_11                      -0.307252   0.310845  -0.988  0.32313
## mask_22                      -0.036781   0.280405  -0.131  0.89566
## mask_33                      -0.151985   0.291795  -0.521  0.60256
## mask_44                       0.124307   0.393291   0.316  0.75201
## r_0_50:factor(wave)2         -0.508015   0.338589  -1.500  0.13378
## r_0_50:factor(wave)3         -0.729772   0.337691  -2.161  0.03089 *
## r_0_50:mask_11                0.629858   0.716027   0.880  0.37922
## factor(wave)2:mask_11         0.378853   0.689182   0.550  0.58262
## factor(wave)3:mask_11        -0.442274   0.457747  -0.966  0.33414
## r_0_50:mask_22                0.128833   0.617505   0.209  0.83477
## factor(wave)2:mask_22        -0.244112   0.539700  -0.452  0.65113
## factor(wave)3:mask_22        -0.204464   0.438381  -0.466  0.64101
## r_0_50:mask_33                0.390216   0.706304   0.552  0.58072
## factor(wave)2:mask_33         0.008262   0.460980   0.018  0.98570
## factor(wave)3:mask_33        -0.062158   0.376730  -0.165  0.86898
## r_0_50:mask_44               -0.586819   0.754657  -0.778  0.43696
## factor(wave)2:mask_44        -0.792502   0.576336  -1.375  0.16936
## factor(wave)3:mask_44         0.060545   0.490469   0.123  0.90178
## r_0_50:factor(wave)2:mask_11 -1.004401   1.161953  -0.864  0.38754
## r_0_50:factor(wave)3:mask_11  0.629819   0.949979   0.663  0.50747
## r_0_50:factor(wave)2:mask_22  0.366049   0.992415   0.369  0.71231
## r_0_50:factor(wave)3:mask_22  0.155214   0.874609   0.177  0.85917
## r_0_50:factor(wave)2:mask_33 -0.088720   0.946846  -0.094  0.92536
## r_0_50:factor(wave)3:mask_33  0.011059   0.847173   0.013  0.98959
## r_0_50:factor(wave)2:mask_44  1.851888   1.031995   1.794  0.07299 .
## r_0_50:factor(wave)3:mask_44  0.040741   0.951185   0.043  0.96584
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7074 on 1210 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.2356, Adjusted R-squared:  0.2173
## F-statistic: 12.86 on 29 and 1210 DF,  p-value: < 2.2e-16
##
##
## Call:
## lm(formula = lm_formula, data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.05080 -0.43912 -0.07781  0.40216  2.91359
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    14.11078    0.06844 206.179   <2e-16 ***
## r_0_50          1.67964    0.13060  12.861   <2e-16 ***
## mask_11        -0.29641    0.20187  -1.468    0.142
## mask_22        -0.20489    0.19513  -1.050    0.294
## mask_33        -0.15726    0.16183  -0.972    0.331
## mask_44        -0.00461    0.20381  -0.023    0.982
## r_0_50:mask_11  0.33403    0.35807   0.933    0.351
## r_0_50:mask_22  0.34026    0.36165   0.941    0.347
## r_0_50:mask_33  0.33929    0.31917   1.063    0.288
## r_0_50:mask_44 -0.06965    0.37753  -0.185    0.854
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7178 on 1230 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.1999, Adjusted R-squared:  0.194
## F-statistic: 34.14 on 9 and 1230 DF,  p-value: < 2.2e-16
```

```r
names(mask_lm_models_train)
```

```
## [1] "all_corlengths_wave_mask_train" "all_corlengths_mask_train"
## [3] "all_localcor_wave_mask_train"   "all_localcor_mask_train"
```

```r
# stargazer table for all_corlengths_wave_mask_train and all_corlengths_mask_train
# (1) is all
# (2) is urban
stargazer(
  c(mask_lm_models_train[1], mask_lm_models_train[2]),
  type = 'text',
  title = "Regression of Covid Cases on Correlation Length and Mask Usage"
)
```

```
##
## Regression of Covid Cases on Correlation Length and Mask Usage
## ================================================================================
##                                                 Dependent variable:
```

```
##                                      ------------------------------------------------
##                                                 log(next_week_marginal_cases)
##                                                   (1)                      (2)
## ---------------------------------------------------------------------------------
## cor_lengths                                     0.001***                 0.001***
##                                                 (0.0002)                 (0.0001)
##
## factor(wave)2                                   -0.038
##                                                  (0.125)
##
## factor(wave)3                                   0.802***
##                                                  (0.109)
##
## mask_11                                         -0.209                   -0.299**
##                                                  (0.246)                 (0.141)
##
## mask_22                                         -0.231                    -0.197
##                                                  (0.243)                 (0.153)
##
## mask_33                                         -0.245                   -0.234*
##                                                  (0.231)                 (0.138)
##
## mask_44                                         -0.503                    -0.112
##                                                  (0.327)                 (0.172)
##
## cor_lengths:factor(wave)2                        0.0002
##                                                 (0.0002)
##
## cor_lengths:factor(wave)3                       -0.001***
##                                                 (0.0002)
##
## cor_lengths:mask_11                               0.001                  0.001**
##                                                  (0.001)                 (0.0003)
##
## factor(wave)2:mask_11                           -0.376
##                                                  (0.381)
##
## factor(wave)3:mask_11                           -0.169
##                                                  (0.324)
##
## cor_lengths:mask_22                              0.0004                   0.0004
##                                                 (0.0005)                 (0.0003)
##
## factor(wave)2:mask_22                           -0.262
##                                                  (0.417)
##
## factor(wave)3:mask_22                             0.104
##                                                  (0.348)
##
## cor_lengths:mask_33                              0.0005                   0.0005
##                                                  (0.001)                 (0.0003)
##
## factor(wave)2:mask_33                           -0.025
##                                                  (0.364)
```

```
##
## factor(wave)3:mask_33                     -0.040
##                                           (0.308)
##
## cor_lengths:mask_44                        0.001                   0.0001
##                                           (0.001)                 (0.0003)
##
## factor(wave)2:mask_44                      0.697
##                                           (0.465)
##
## factor(wave)3:mask_44                      0.335
##                                           (0.403)
##
## cor_lengths:factor(wave)2:mask_11         0.0005
##                                           (0.001)
##
## cor_lengths:factor(wave)3:mask_11         0.0005
##                                           (0.001)
##
## cor_lengths:factor(wave)2:mask_22         0.0003
##                                           (0.001)
##
## cor_lengths:factor(wave)3:mask_22         -0.0001
##                                           (0.001)
##
## cor_lengths:factor(wave)2:mask_33        -0.00002
##                                           (0.001)
##
## cor_lengths:factor(wave)3:mask_33         0.0002
##                                           (0.001)
##
## cor_lengths:factor(wave)2:mask_44         -0.001
##                                           (0.001)
##
## cor_lengths:factor(wave)3:mask_44         -0.001
##                                           (0.001)
##
## Constant                                 14.125***               14.462***
##                                           (0.081)                 (0.048)
##
## ----------------------------------------------------------------------------------
## Observations                               1,186                   1,186
## R2                                         0.253                   0.170
## Adjusted R2                                0.235                   0.164
## Residual Std. Error                 0.707 (df = 1156)        0.739 (df = 1176)
## F Statistic                     13.530*** (df = 29; 1156) 26.807*** (df = 9; 1176)
## ==================================================================================
## Note:                                                  *p<0.1; **p<0.05; ***p<0.01
```

```r
# stargazer table for all_localcor_wave_mask_train and all_localcor_mask_train
# (1) is all
# (2) is urban
stargazer(
  c(mhhinc_lm_models_train[3], mhhinc_lm_models_train[4]),
```

```r
  type = 'text',
  title = "Regression of Covid Cases on Local Correlation and Mask Usage"
)
```

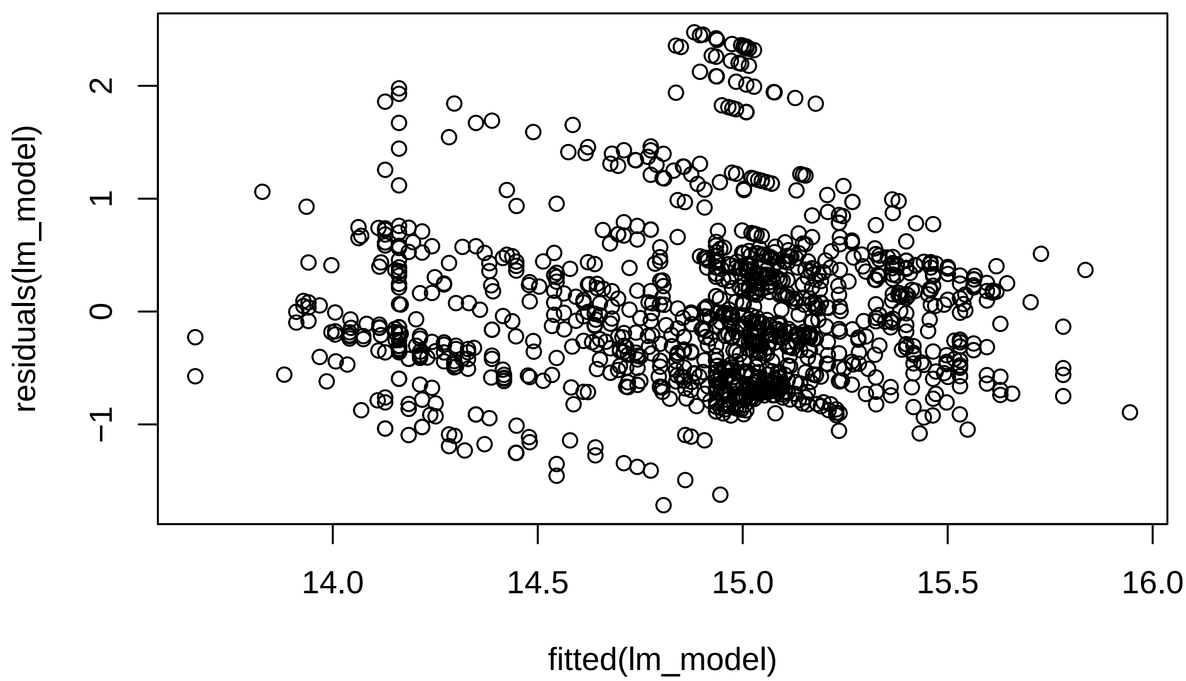
```
##
## Regression of Covid Cases on Local Correlation and Mask Usage
## ==================================================================================
##                                                    Dependent variable:
##                                   ------------------------------------------------
##                                             log(next_week_marginal_cases)
##                                              (1)                      (2)
## ----------------------------------------------------------------------------------
## r_0_50                                     1.657***                0.984***
##                                            (0.549)                 (0.310)
##
## factor(wave)2                               0.575
##                                            (0.415)
##
## factor(wave)3                              0.916**
##                                            (0.395)
##
## same_tier                                  -0.023                  -0.379**
##                                            (0.346)                 (0.182)
##
## tier_diff_1                                -0.351                 -0.591***
##                                            (0.356)                 (0.189)
##
## tier_diff_2                                -0.059                  -0.349*
##                                            (0.372)                 (0.199)
##
## r_0_50:factor(wave)2                       -0.537
##                                            (0.766)
##
## r_0_50:factor(wave)3                       -0.888
##                                            (0.761)
##
## r_0_50:same_tier                            0.300                   0.663*
##                                            (0.630)                 (0.354)
##
## factor(wave)2:same_tier                    -0.294
##                                            (0.491)
##
## factor(wave)3:same_tier                    -0.468
##                                            (0.444)
##
## r_0_50:tier_diff_1                         1.229*                  1.177***
##                                            (0.662)                 (0.368)
##
## factor(wave)2:tier_diff_1                  -0.315
##                                            (0.498)
##
## factor(wave)3:tier_diff_1                  -0.240
##                                            (0.462)
```

```
##
## r_0_50:tier_diff_2                          0.570                   0.709*
##                                            (0.703)                  (0.387)
##
## factor(wave)2:tier_diff_2                   0.176
##                                            (0.528)
##
## factor(wave)3:tier_diff_2                  -0.609
##                                            (0.484)
##
## r_0_50:factor(wave)2:same_tier              0.063
##                                            (0.900)
##
## r_0_50:factor(wave)3:same_tier              0.526
##                                            (0.863)
##
## r_0_50:factor(wave)2:tier_diff_1           -0.228
##                                            (0.925)
##
## r_0_50:factor(wave)3:tier_diff_1           -0.046
##                                            (0.909)
##
## r_0_50:factor(wave)2:tier_diff_2           -0.847
##                                            (0.980)
##
## r_0_50:factor(wave)3:tier_diff_2            0.739
##                                            (0.960)
##
## Constant                                  13.821***               14.457***
##                                            (0.312)                  (0.158)
##
## ----------------------------------------------------------------------------------
## Observations                                1,240                    1,240
## R2                                          0.223                    0.182
## Adjusted R2                                 0.209                    0.178
## Residual Std. Error                   0.711 (df = 1216)        0.725 (df = 1232)
## F Statistic                     15.210*** (df = 23; 1216) 39.234*** (df = 7; 1232)
## ==================================================================================
## Note:                                                *p<0.1; **p<0.05; ***p<0.01
```

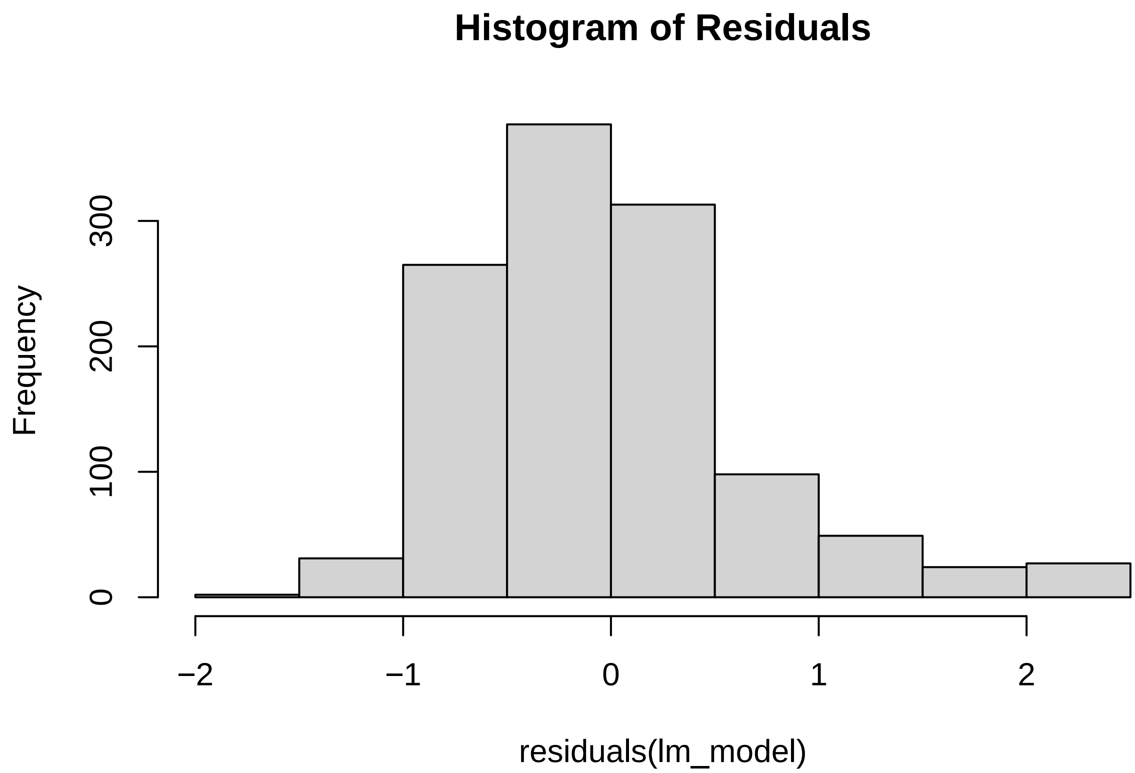
Check conditions of Normality of Residuals, Homoskedasticity,Linearity , No Perfect Multicollinearity, Independence of Residuals

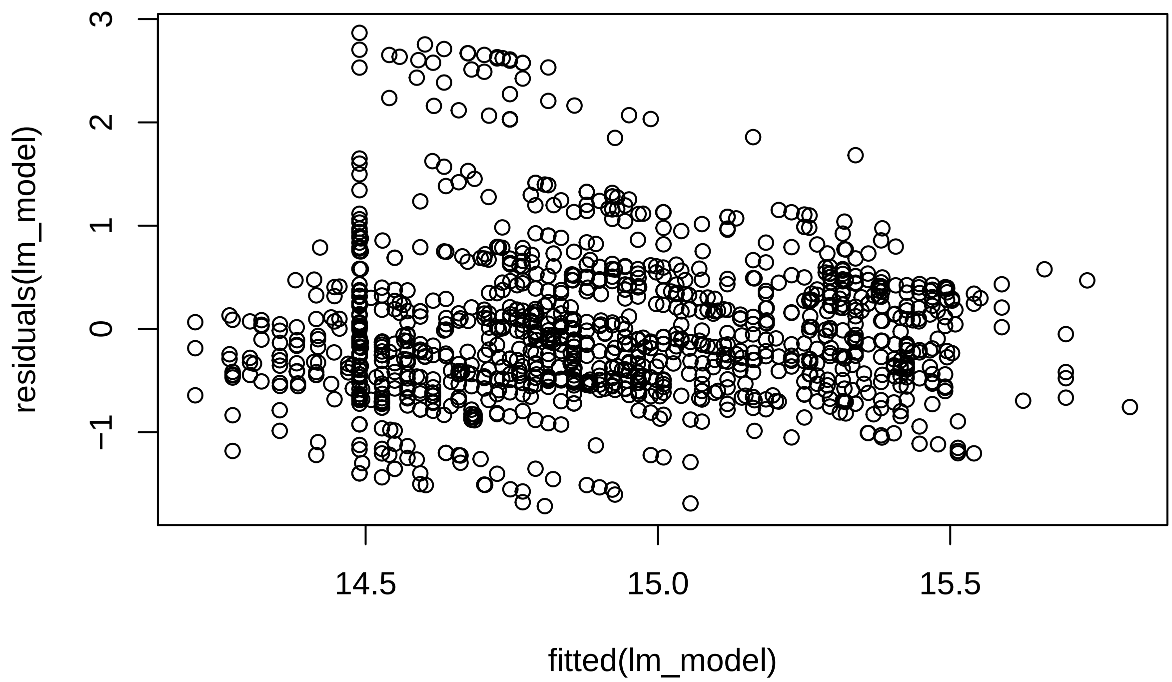
```r
for (lm_model in mask_lm_models_train) {
# 1. Linearity Check (Visual Inspection)
plot(residuals(lm_model) ~ fitted(lm_model))

# 2. Independence Check
# Durbin-Watson test
print(durbinWatsonTest(lm_model)[3])

# 3. Homoscedasticity Check
```

```r
# Breusch-Pagan test for Heteroscedasticity
print(bptest(lm_model)$p.value)

# 4. Normality of Residuals Check (Visual Inspection)
hist(residuals(lm_model), main = "Histogram of Residuals")
}
```

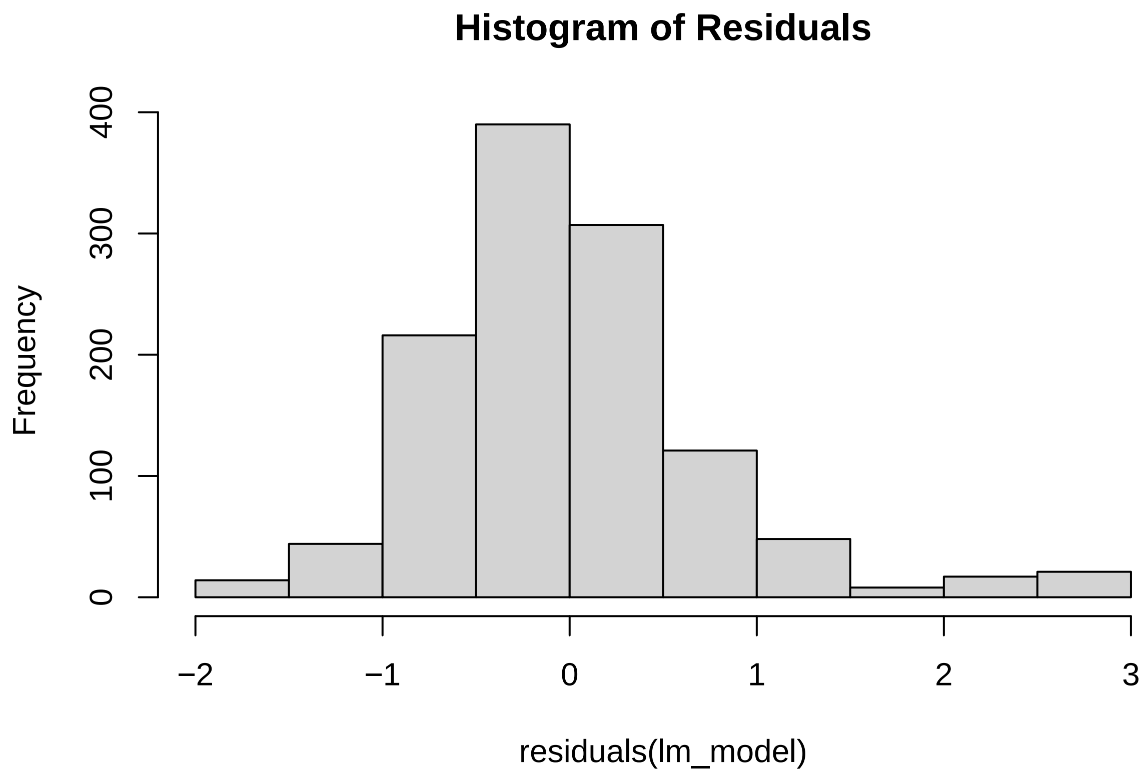



```
## $p
## [1] 0.948
##
##           BP
## 6.851723e-06
```

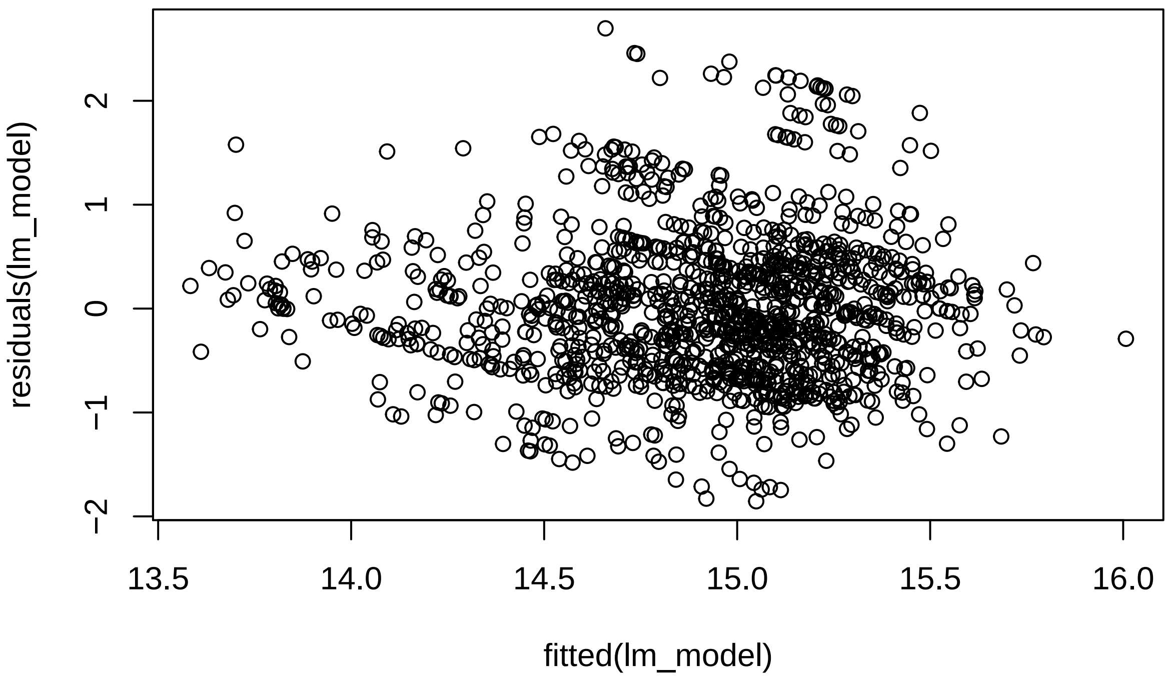

Histogram of Residuals
Frequency
0
100
200
300
−2
−1
0
1
2
residuals(lm_model)



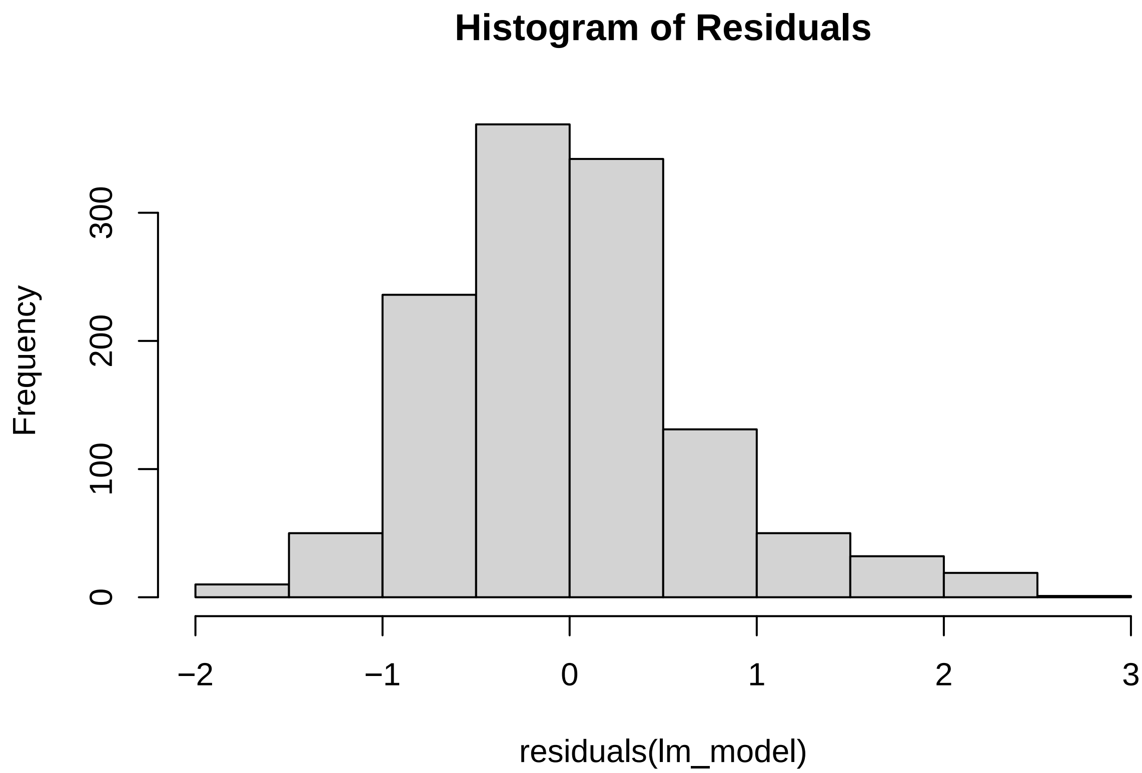

```
## $p
## [1] 0.826
##
##           BP
## 0.0003432578
```

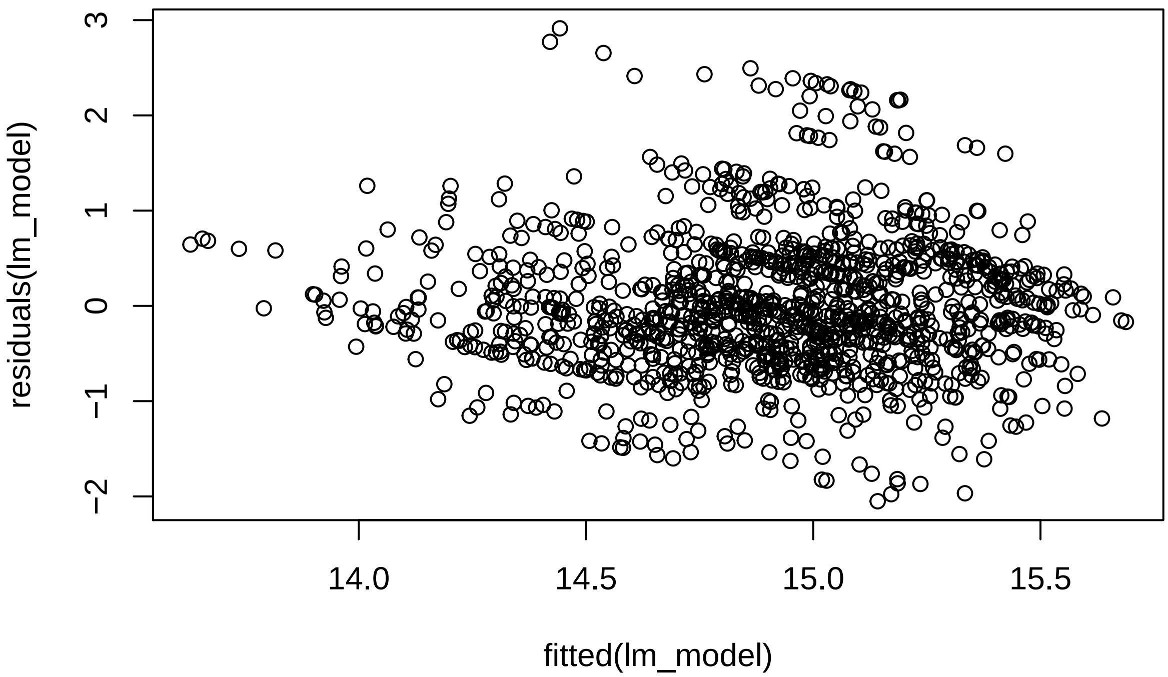


Histogram of Residuals
Frequency
0
100
200
300
400
−2
−1
0
1
2
3
residuals(lm_model)

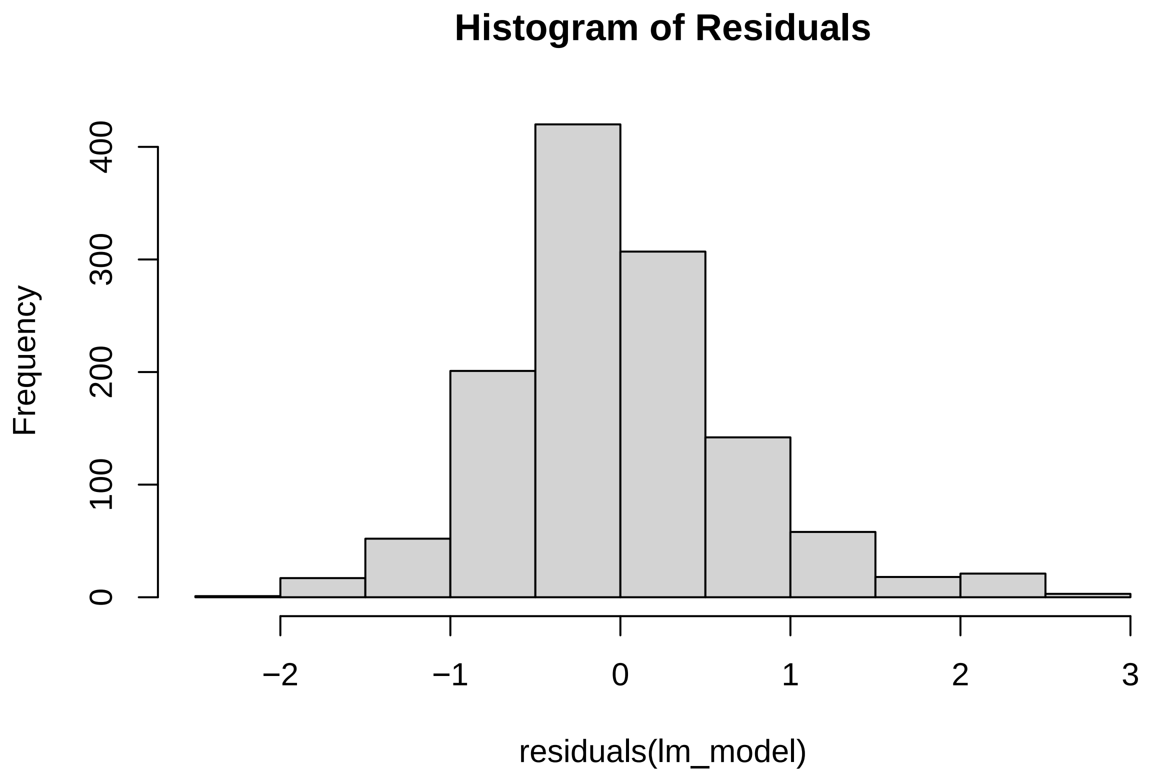



```
## $p
## [1] 0.966
##
##           BP
## 0.0002166926
```


Histogram of Residuals
Frequency
0
100
200
300
−2
−1
0
1
2
3
residuals(lm_model)




```
## $p
## [1] 0.682
##
##        BP
## 0.2130556
```




```
# 5. No Perfect Multicollinearity Check
# Variance Inflation Factors (VIF)
vif(lm(log(next_week_marginal_cases) ~  cor_lengths + factor(wave) + factor(mask_usage) , data = all_ma
```

```
##                         GVIF Df GVIF^(1/(2*Df))
## cor_lengths         1.164340  1        1.079046
## factor(wave)        1.087388  2        1.021165
## factor(mask_usage) 1.077052  9        1.004132
```

```
# Conditions not passed:
## Heteroskedasticity in all_localcor_mask_train
```

Interpretations

```
# Initialize values
predictor <- c("cor_lengths","r_0_50")
waves <- c("wave 1","wave 2", "wave 3")
usage_same <- c("mask_11", "mask_22", "mask_33", "mask_44")
usage_descr <- c("is less than 40%", "is between 40% and 50%", "is between 50% and 60%", "is greater tha
full_coef = c()
full_coef1 = c()
significant = c()
significant1 = c()
```

```
i=1
j=1

# Iterate over Wave and Tier diff dummies
for (model in 1:length(mask_lm_models_train)) {

  # Get model stats
  lm_model = mask_lm_models_train[[model]]
  coefficients <- coef(lm_model)
  coef_names <- names(coef(lm_model))
  # se <- summary(lm_model)$coefficients[, "Std. Error"] use robust SE's
  se <- sqrt(diag(vcovHC(lm_model)))
  se_named <- rep(NA, length(coef_names))
  se_named[names(se)] <- se
  st_errors <- se_named[coef_names]

  # Based on model, decide if urban or all county df
  if (model <=4) {
    county = "all counties"
    df = all_mask_weekly_spatial_metrics}
  else {
    county = "urban counties"
    df = urban_mask_weekly_spatial_metrics}

  # Based on model, choose cor_length or local_cor as predictor
  if (model %in% c(1,2)) {pred = predictor[1]}
  else {pred = predictor[2]}

  # Wave dummies not included for even models
  if (model %% 2 == 0) {
    wave = "every wave"
    for (m in 1:length(usage_same)) {
      usage_comb <- usage_same[m]
      interaction = coefficients[5+m]
      se_total = sqrt(sum((st_errors[5+m])^2 + (st_errors[2])^2))
      # Store coefficient
      full_coef1[j] = save_coefficient(interaction)

      # Print interpretation and significance
      significant1[j] <- is_significant(as.numeric(full_coef1[j]), as.numeric((se_total)))
      print_interpretation(full_coef1[j], wave, region="whole US", tier_descr = "are disregarded", signi

  # Wave dummies with mhhinc tier dummies
  else {
  # Iterate over each wave
  for (w in 1:length(waves)) {
    wave <- waves[w]
    # Iterate of each tier difference
    for (m in 1:length(usage_same)) {
      usage_comb <- usage_same[m]
      if (wave == "Wave 1") {
        partial_interaction_w <- 0
        se_partial_w <- 0
```

```r
      } else {
        partial_interaction_w <- coefficients[(7 + w)]
        se_partial_w <- st_errors[(7 + w)]
      }

        partial_interaction_m <- coefficients[(3*m+8)]
        se_partial_m <- st_errors[(3*m+8)]

      if (wave != "Wave 1") {
        full_interaction_wm <- coefficients[paste0(pred,":factor(wave)", w, ":", usage_comb)]
        se_full_wm <- st_errors[paste0(pred,":factor(wave)", w, ":", usage_comb)]
      } else {
        full_interaction_wm <- 0
        se_full_wm <- 0
      }
    # Calculate total standard error
    se_total <- sqrt(se_partial_w^2 + se_partial_m^2 + se_full_wm^2 + st_errors[2]^2)
    # Store coefficient information
    interaction <- partial_interaction_w + partial_interaction_m + full_interaction_wm

    # Store coefficient
    full_coef[i] = save_coefficient(interaction)

    # Print interpretation and significance
    significant[i] <- is_significant(as.numeric(full_coef[i]), as.numeric((se_total)))
    print_interpretation(full_coef[i], wave, region="whole US", tier_descr = "are disregarded", signific
    i <- i + 1
    }}}
}
```

```
## - For all counties across wave 2 in the whole US where the median household income tiers are disrega
##   and where mask usage in both counties is less than 40% a 1 sd increase correlation length (244 km)
##   corresponds to a 66.23% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 2 in the whole US where the median household income tiers are disrega
##   and where mask usage in both counties is between 40% and 50% a 1 sd increase correlation length (24
##   corresponds to a 57.21% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 2 in the whole US where the median household income tiers are disrega
##   and where mask usage in both counties is between 50% and 60% a 1 sd increase correlation length (24
##   corresponds to a 51.36% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are dis
##   and where mask usage in both counties is between 40% and 50% a 1 sd increase correlation length (24
##   corresponds to a 44.59% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are dis
##   and where mask usage in both counties is between 50% and 60% a 1 sd increase correlation length (24
##   corresponds to a 35.7% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are dis
##   and where mask usage in both counties is greater than 60% a 1 sd increase correlation length (244 k
##   corresponds to a 38.11% increase in cases in the following week. Significant? TRUE
## - For all counties across wave 3 in the whole US where the median household income tiers are disrega
##   and where mask usage in both counties is less than 40% a 1 sd increase local correlation (0.18)
##   corresponds to a 48.66% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are dis
##   and where mask usage in both counties is less than 40% a 1 sd increase local correlation (0.18)
```

```
##  corresponds to a 30.15% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are dis
##  and where mask usage in both counties is between 40% and 50% a 1 sd increase local correlation (0.1
##  corresponds to a 36.25% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are dis
##  and where mask usage in both counties is between 50% and 60% a 1 sd increase local correlation (0.1
##  corresponds to a 36.36% increase in cases in the following week. Significant? TRUE
## - For all counties across every wave in the whole US where the median household income tiers are dis
##  and where mask usage in both counties is greater than 60% a 1 sd increase local correlation (0.18)
##  corresponds to a 36.34% increase in cases in the following week. Significant? TRUE
```