Checking model assumptions

```
## Loading data.
income<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/COMPLETE_income50_75k.csv", header = TRUE)
poverty<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/COMPLETE_povertytable.csv", header = TRUE)
homeown<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/COMPLETE_homeowner_occupied.csv", header = TRUE)
foodstamps<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/COMPLETE_percentfoodstamps.csv", header = TRUE)
mobility<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/mobility.csv", header = TRUE)
insurance<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/healthinsurance.csv", header = TRUE)
crimes<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/crimestable.csv", header = TRUE)
artcount<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/Indyarts_censuscount.csv", header = TRUE)
walk<- read.csv("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/walk.csv", header = TRUE)

# Aggregate by GEO_ID and calculate the mean of walkability
walk <- aggregate(walkability ~ GEO_ID, data = walk, mean)

load("C:/Users/ureka/OneDrive/Documents/Stata 2022/Danicia_stats/ICPSR_38586-V1/ICPSR_38586/DS0001/38586-0001-Data.rda")
parks<- da38586.0001
parks$TRACT_FIPS10 <- paste0("1400000US", as.character(parks$TRACT_FIPS10))
parks$GEO_ID <- parks$TRACT_FIPS10
```

```
## Merge each table one at a time. Each iteration only keeps observations that exist in BOTH tab
les,
## while excluding those that only exist in one.
merged_table <- merge(income, poverty, by = "GEO_ID")
merged_table <- merge(merged_table, homeown, by = "GEO_ID")
merged_table <- merge(merged_table, foodstamps, by = "GEO_ID")
merged_table <- merge(merged_table, mobility, by = "GEO_ID")
merged_table <- merge(merged_table, insurance, by = "GEO_ID")
merged_table <- merge(merged_table, artcount, by = "GEO_ID")
merged_table <- merge(merged_table, parks, by = "GEO_ID")
merged_table <- merge(merged_table, walk, by = "GEO_ID")
merged_table <- merge(merged_table, crimes, by = "GEO_ID")

wbindex <- merged_table

wbindex$percentPoverty <- as.numeric(wbindex$percentPoverty)
wbindex$income <- as.numeric(wbindex$income)
wbindex$owneroccupied  <- as.numeric(wbindex$owneroccupied)
wbindex$foodstamps <- as.numeric(wbindex$foodstamps)
wbindex$mobility <- as.numeric(wbindex$mobility)
wbindex$healthinsurance <- as.numeric(wbindex$healthinsurance)
```

```
## Creating outcome variable
wbindex$index <- ((wbindex$income) + (wbindex$percentPoverty) + (wbindex$owneroccupied) + (wbind
ex$foodstamps))/ 4

wbindex <- wbindex[, c("GEO_ID", "index", "income", "percentPoverty",
                       "owneroccupied", "foodstamps", "healthinsurance",
                       "artcount", "mobility", "crimes", "walkability", "TOT_PARK_AREA_SQMILE
S")]
colnames(wbindex)
```

```
##  [1] "GEO_ID"                "index"                 "income"               
##  [4] "percentPoverty"        "owneroccupied"         "foodstamps"           
##  [7] "healthinsurance"       "artcount"              "mobility"             
## [10] "crimes"                "walkability"           "TOT_PARK_AREA_SQMILES"
```

```
## Linear regression
model <- lm(index ~ artcount + mobility + crimes + healthinsurance + walkability + TOT_PARK_AREA
_SQMILES, data = wbindex)
summary(model)
```

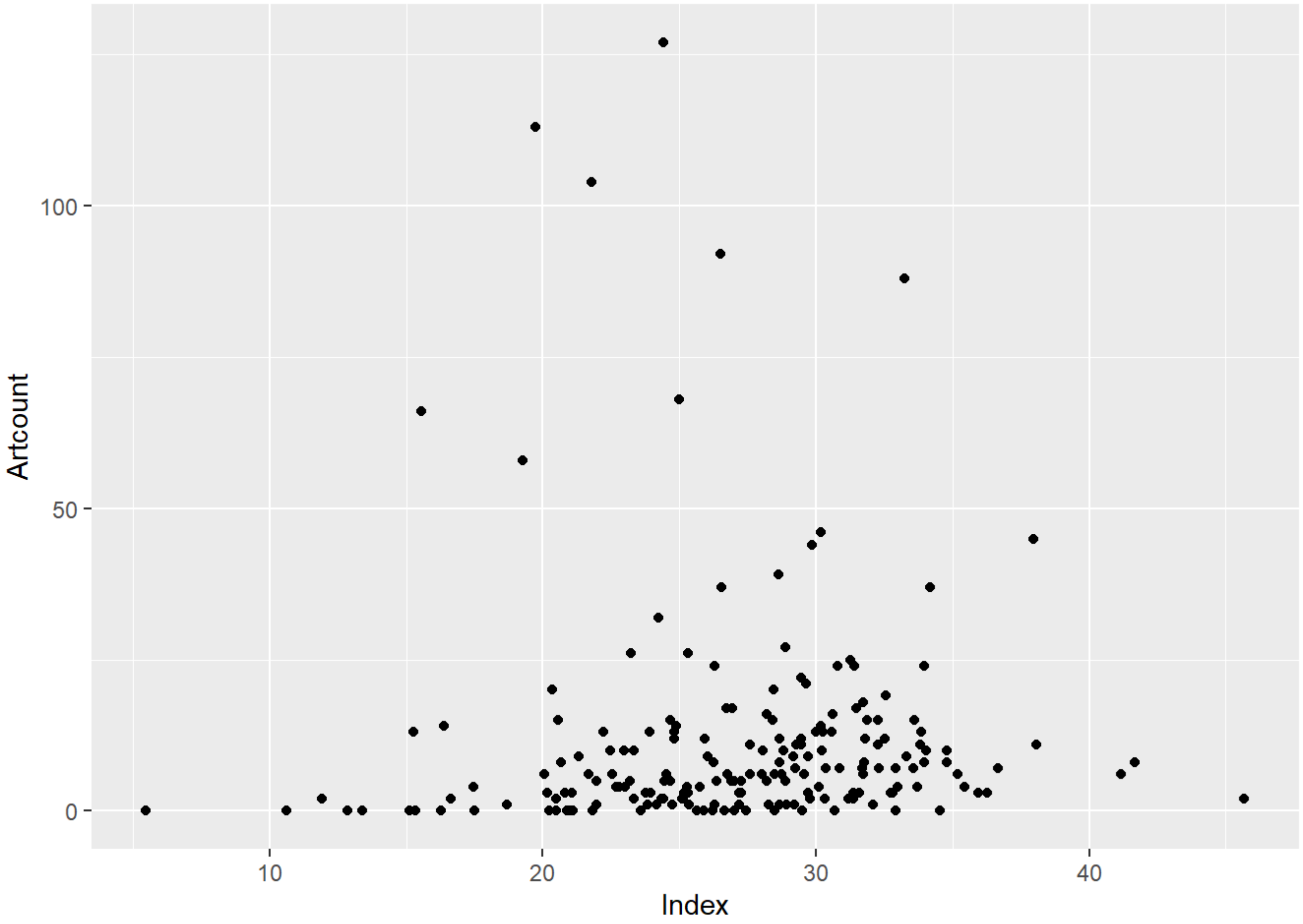
```
##
## Call:
## lm(formula = index ~ artcount + mobility + crimes + healthinsurance +
##     walkability + TOT_PARK_AREA_SQMILES, data = wbindex)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -20.4391  -2.2498   0.4523   3.3850  13.1212
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           33.1073132  2.8641083  11.559  < 2e-16 ***
## artcount               0.0224592  0.0218787   1.027  0.30597
## mobility               0.0064542  0.0010980   5.878 1.89e-08 ***
## crimes                -0.0003304  0.0029667  -0.111  0.91144
## healthinsurance       -0.0009839  0.0003381  -2.910  0.00405 **
## walkability           -0.5641508  0.1950582  -2.892  0.00428 **
## TOT_PARK_AREA_SQMILES  1.1367998  3.1170593   0.365  0.71575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.422 on 186 degrees of freedom
## Multiple R-squared:  0.1744, Adjusted R-squared:  0.1477
## F-statistic: 6.547 on 6 and 186 DF,  p-value: 2.736e-06
```

```
##VIF values greater than 5 or 10 indicate a high degree of multicollinearity.
vif(model)
```

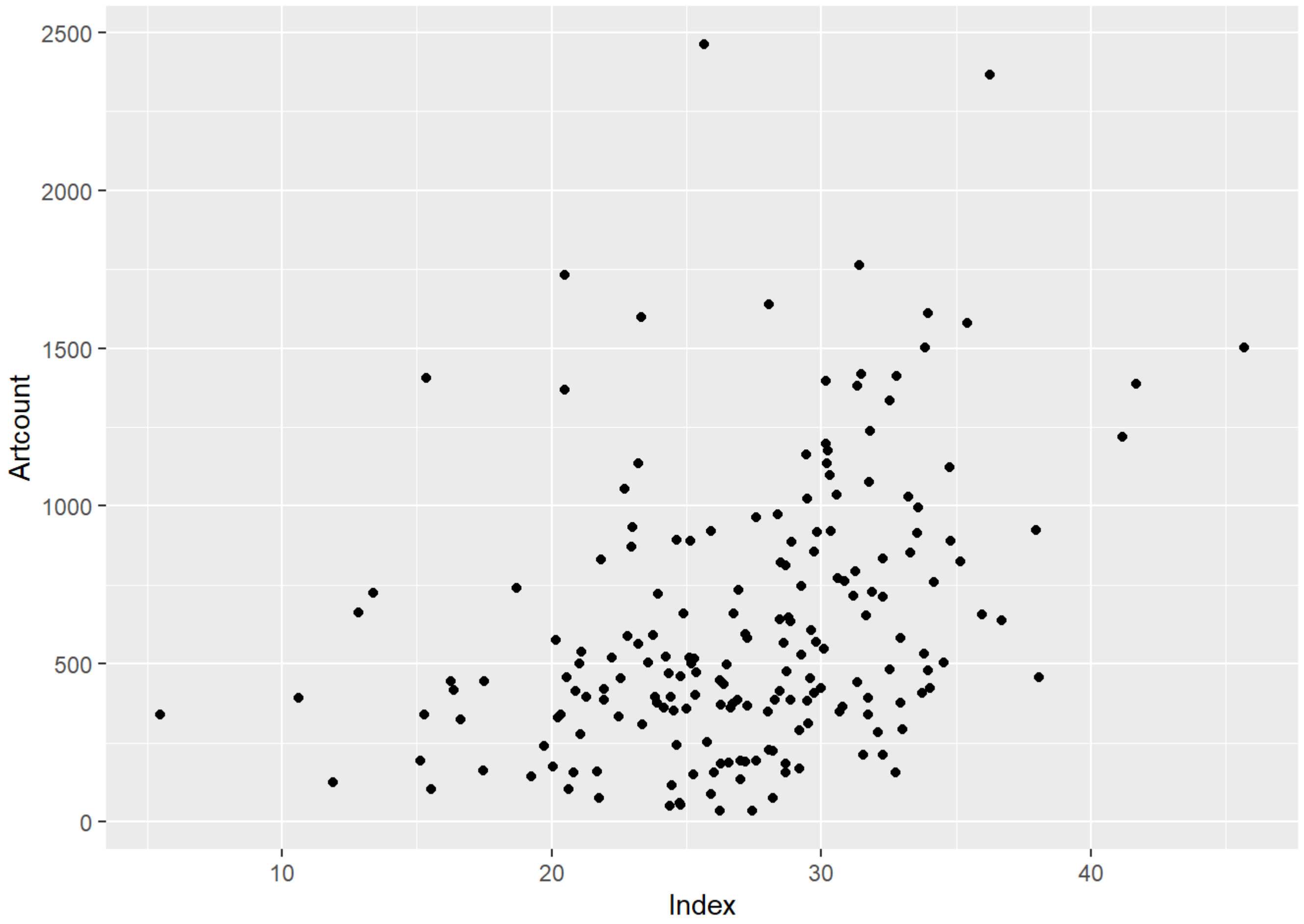
```
##              artcount              mobility                crimes
##              1.135038              1.489126              1.390956
##       healthinsurance           walkability TOT_PARK_AREA_SQMILES
##              1.815871              1.843643              1.069486
```

#correlation plots

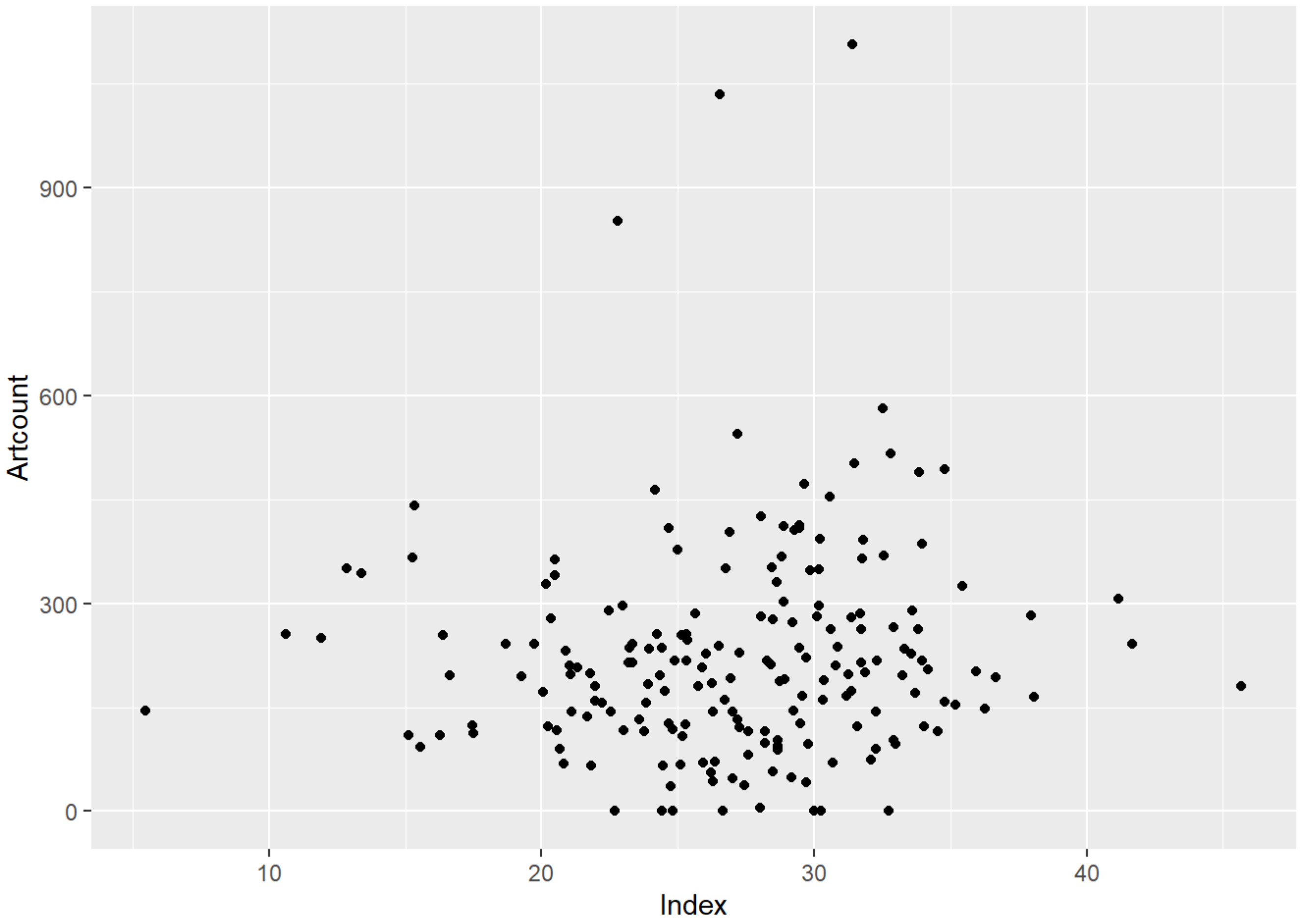
```
ggplot(data = wbindex, aes(x = index, y = artcount)) +
  geom_point() +
  labs(x = "Index", y = "Artcount")
```



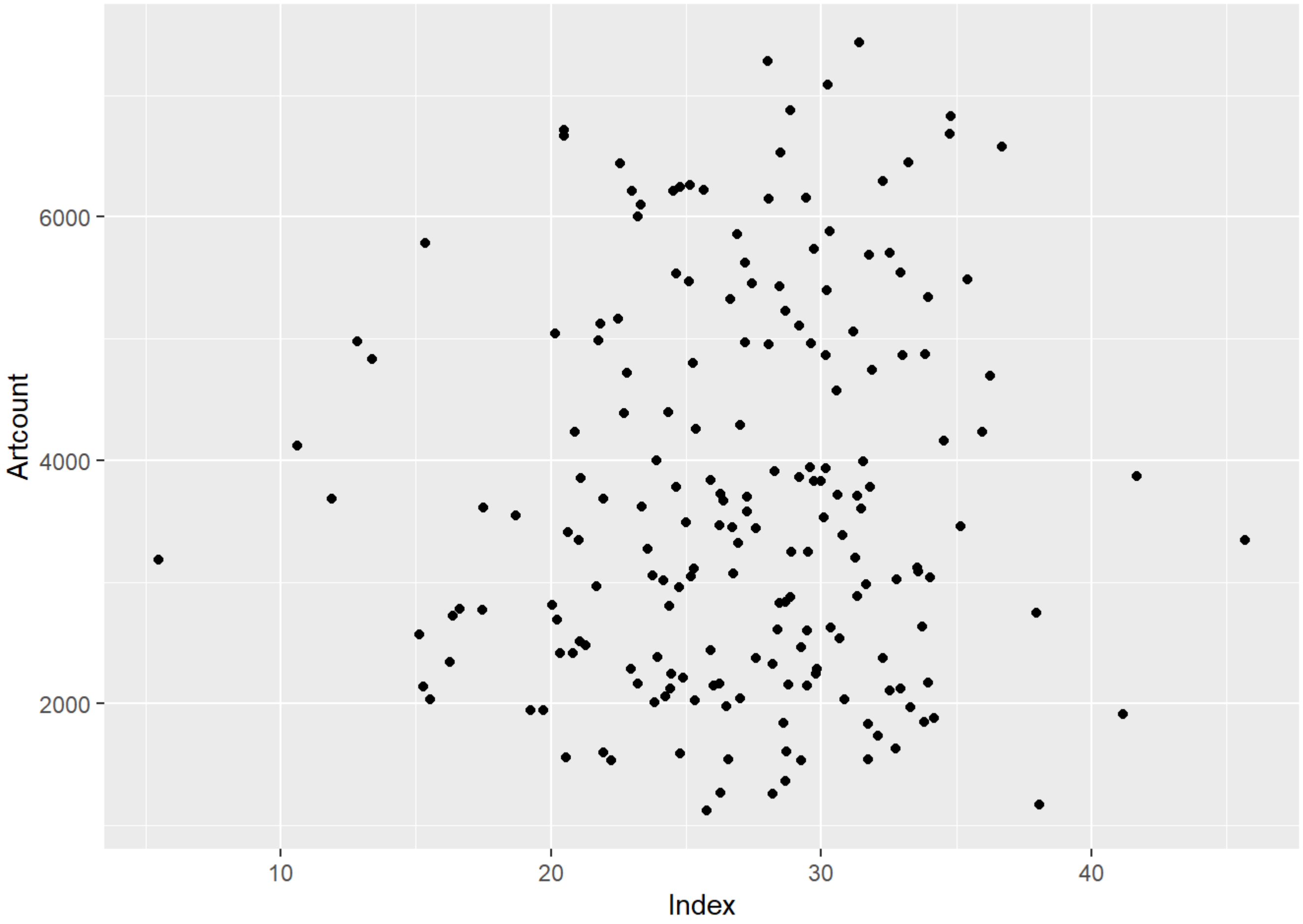
```
ggplot(data = wbindex, aes(x = index, y = mobility)) +
  geom_point() +
  labs(x = "Index", y = "Artcount")
```



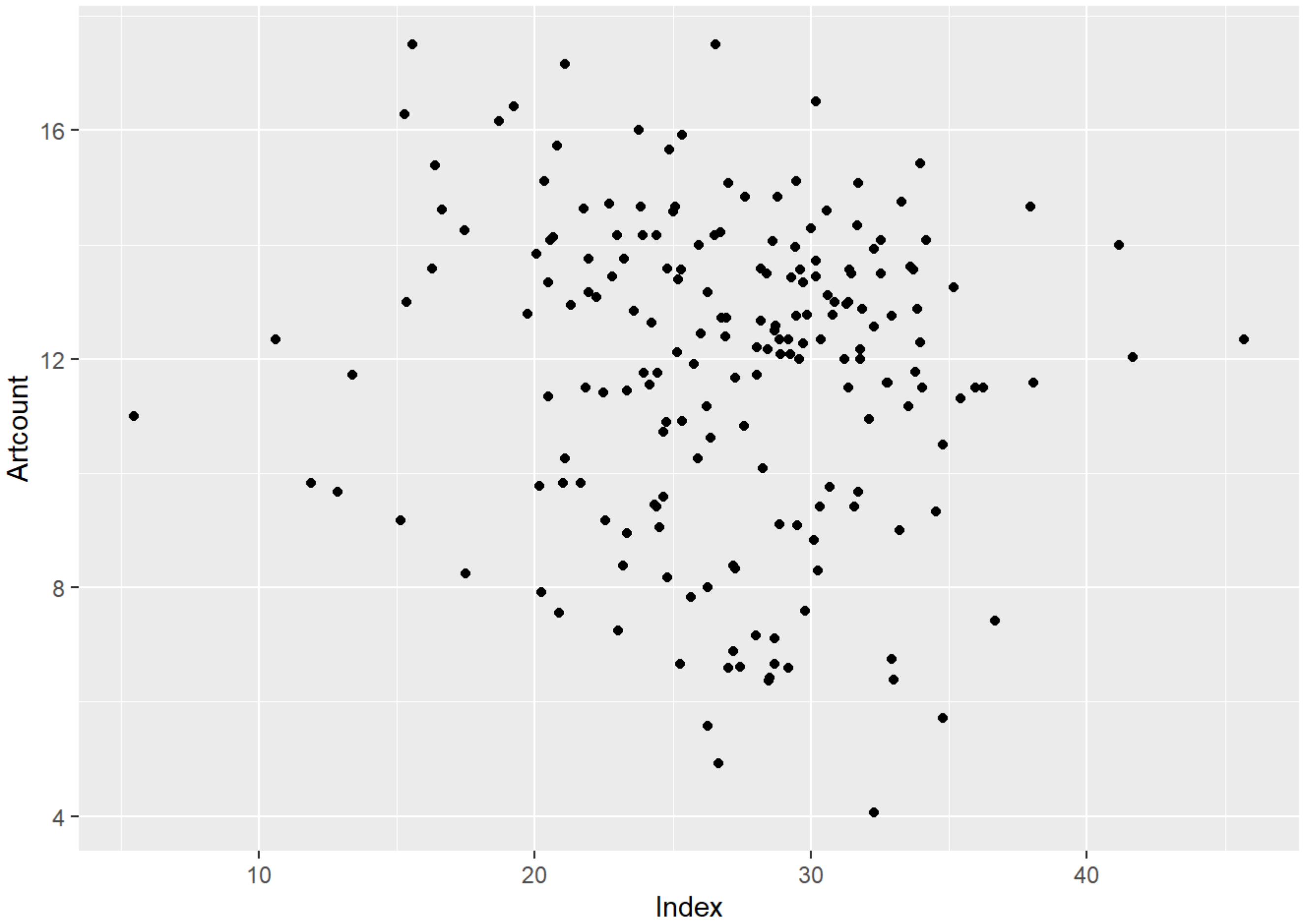
```
ggplot(data = wbindex, aes(x = index, y = crimes)) +
  geom_point() +
  labs(x = "Index", y = "Artcount")
```



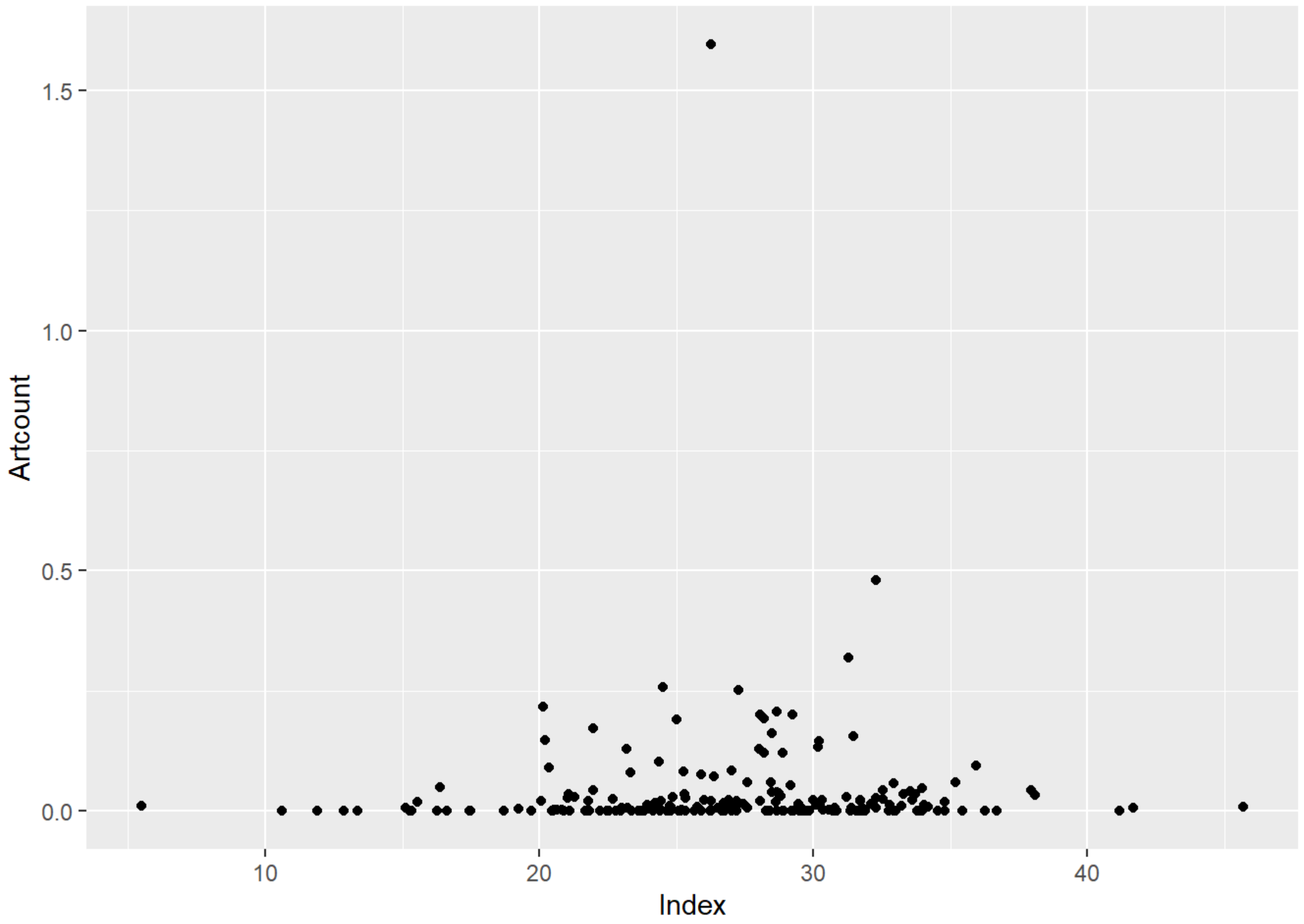
```
ggplot(data = wbindex, aes(x = index, y = healthinsurance)) +
  geom_point() +
  labs(x = "Index", y = "Artcount")
```



```
ggplot(data = wbindex, aes(x = index, y = walkability)) +
  geom_point() +
  labs(x = "Index", y = "Artcount")
```



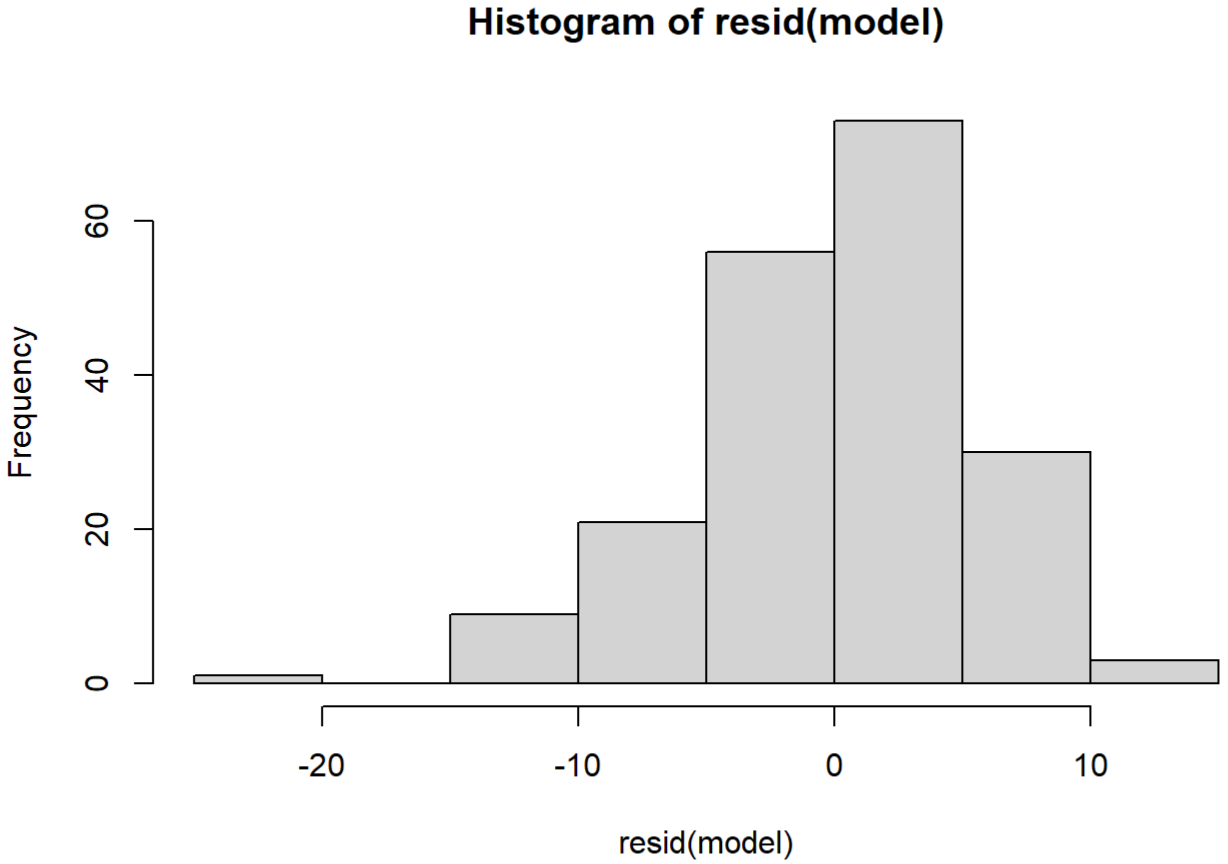
```
ggplot(data = wbindex, aes(x = index, y = TOT_PARK_AREA_SQMILES)) +
  geom_point() +
  labs(x = "Index", y = "Artcount")
```



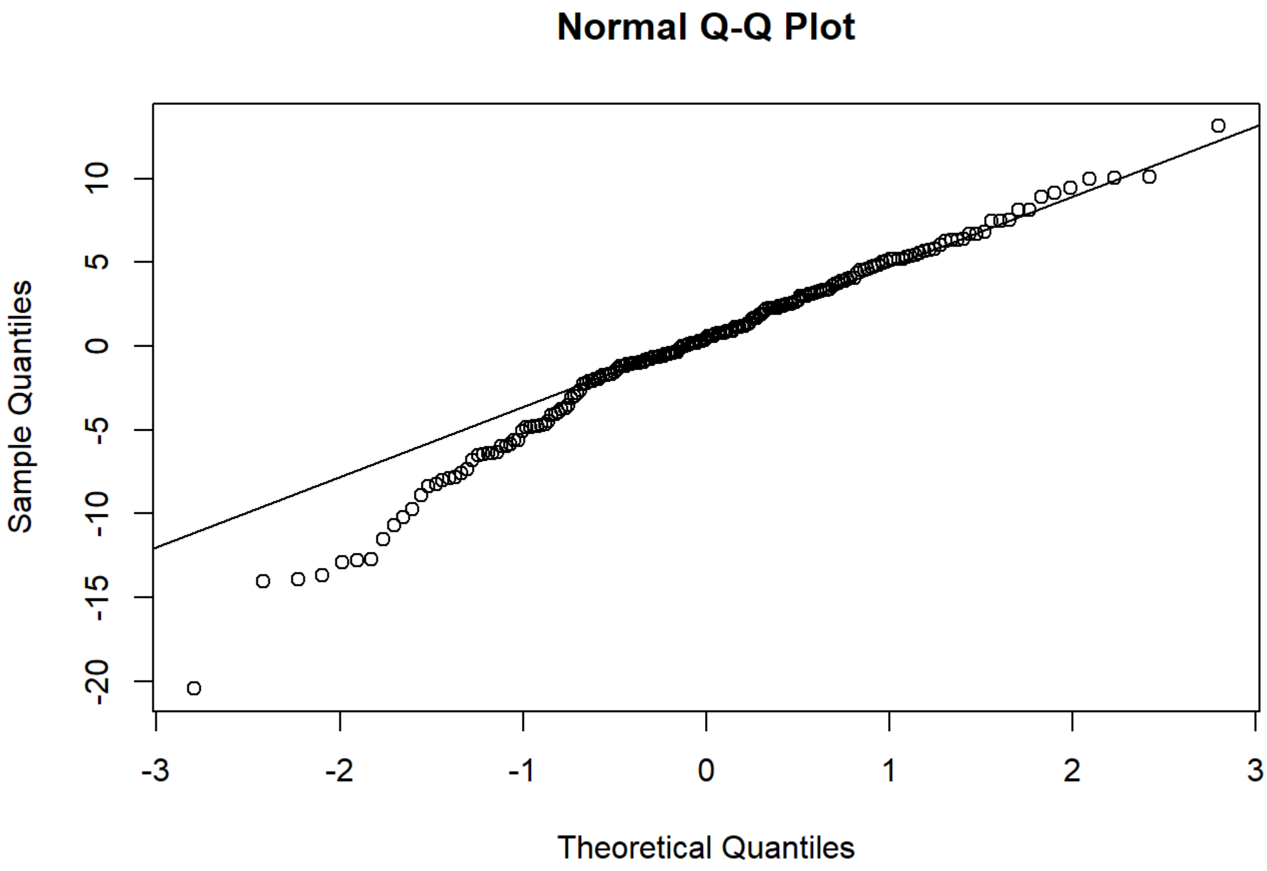
#Check for normality using a histogram and QQ plot

```
#If the histogram appears to be approximately normally distributed and the points on the QQ plot follow the diagonal line closely, then the normality assumption is met.

# Histogram of residuals
hist(resid(model))
```

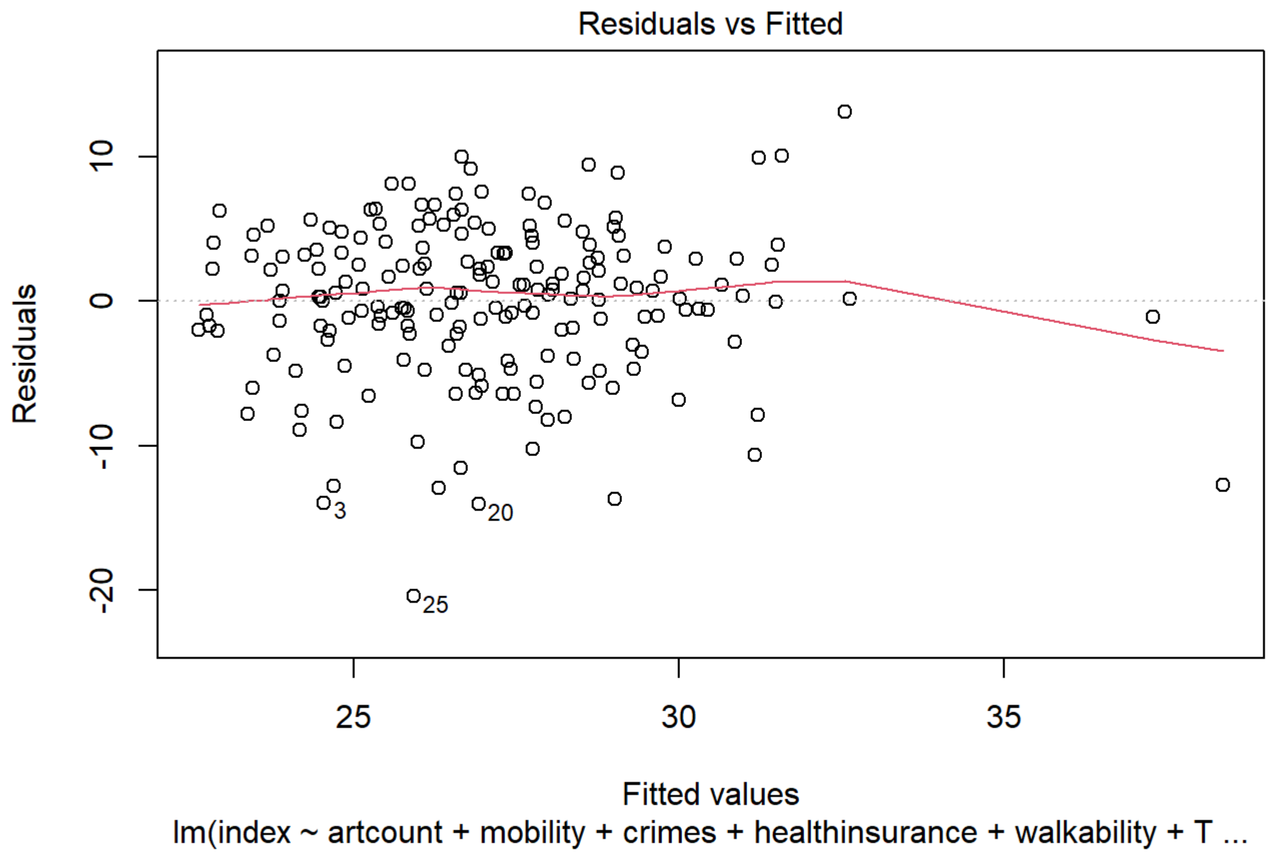


```
# QQ plot of residuals
qqnorm(resid(model))
qqline(resid(model))
```



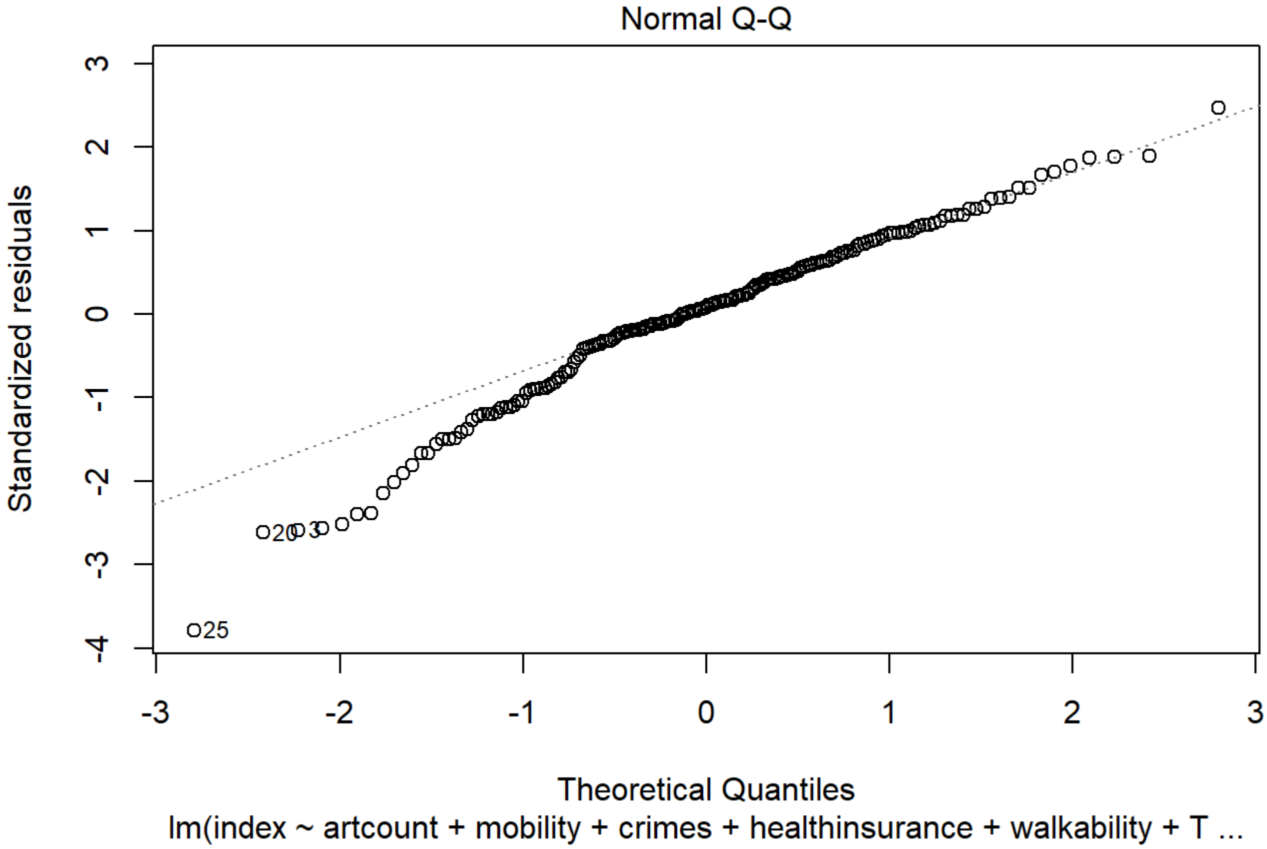
Check for homoscedasticity using a plot of residuals vs. fitted values

```
#If the points on the plot are randomly scattered around the horizontal line with no obvious pattern, then the homoscedasticity assumption is met.
plot(model, which = 1)
```



Check for independence using a plot of residuals vs. order of observations

```
#If the points on the plot are randomly scattered around the horizontal line with no obvious pattern, then the independence assumption is met.
plot(model, which = 2)
```


Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
-3
-4
25
20
3
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(index ~ artcount + mobility + crimes + healthinsurance + walkability + T ...