

# MA5701 COVID-19 FINAL PROJECT

Ezequiel Carrillo, Evan Neibauer, Samantha Hair

11/13/2021

```
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

library(ggplot2)

covid_data <- read.csv('covid_states_data.csv') %>%
  filter(!state %in% c("MP", "DC", "PR"))

set.seed(1234)

rand_sample <- sample(1:50, 20, replace = FALSE)

sample_covid <- covid_data[rand_sample, ]

sample_covid_clean <- sample_covid %>%
  select(-c(fips, country, county, level, lat, locationId, long, unused1, unused2, unused3, unused4, unused5,
            unused6, unused7, unused8, url))

attach(sample_covid_clean)

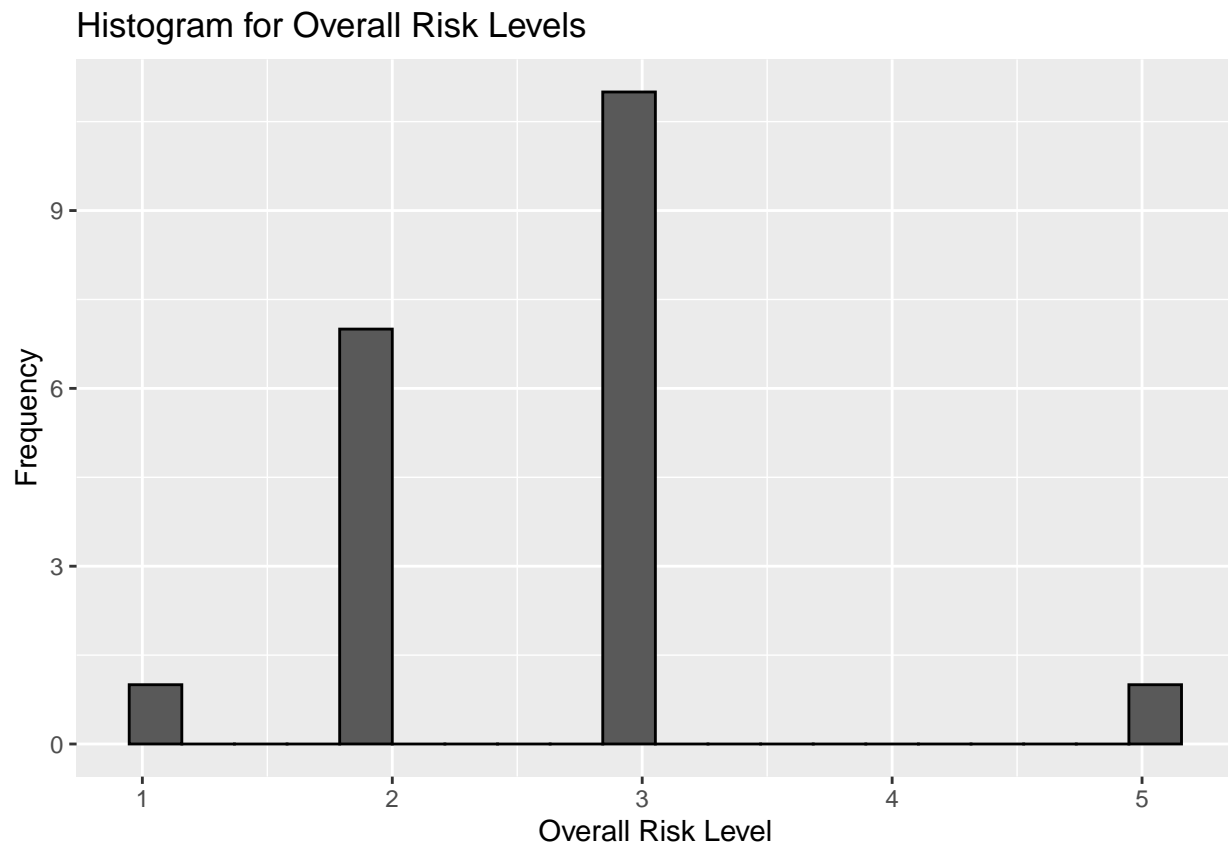
write.csv(sample_covid_clean, 'sample_covid_data.csv')

str(sample_covid_clean)

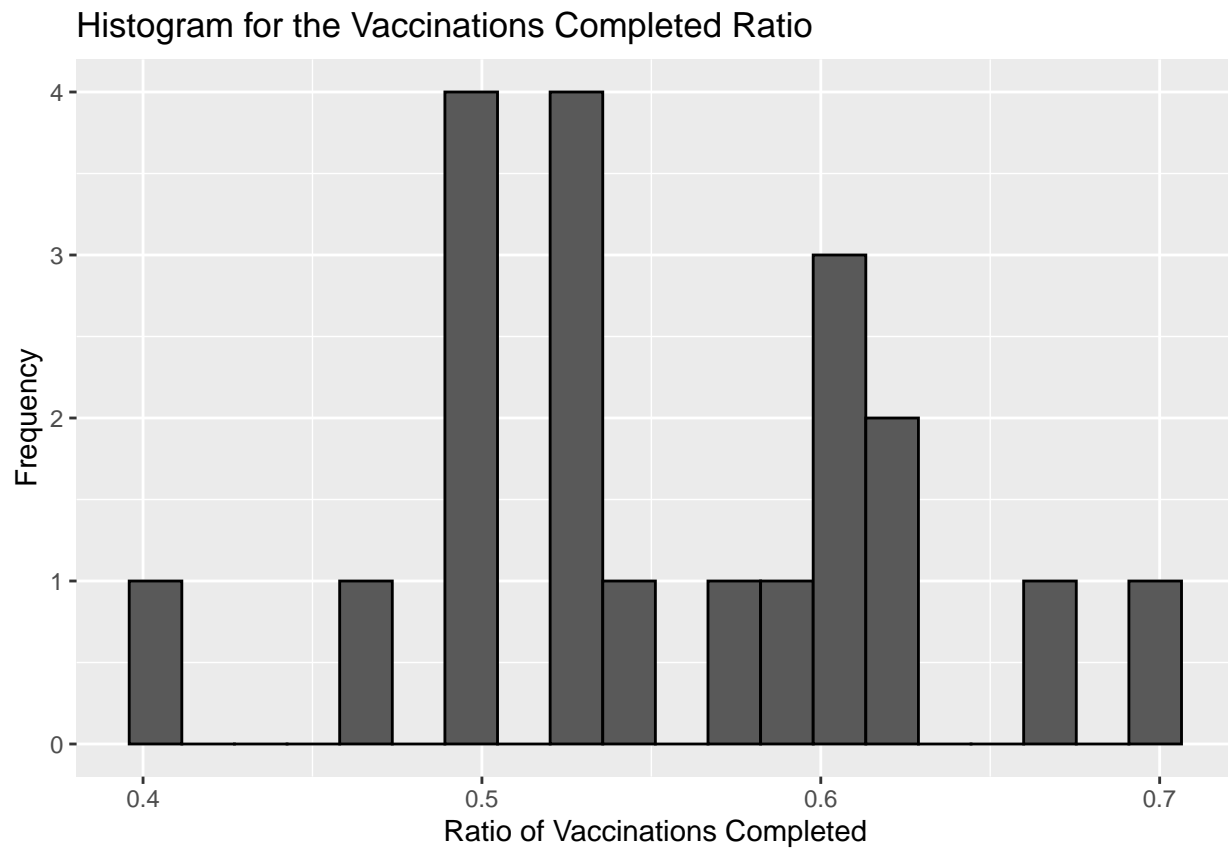
## 'data.frame':   20 obs. of  36 variables:
## $ state                : chr  "ND" "KS" "MI" "OR" ...
## $ population            : int  762062 2913314 9986857 4217737 3205958 21477737 3...
## $ metrics.testPositivityRatio : num  0.091 0.084 0.105 0.068 0.117 0.038 0.021 0.085 0...
## $ metrics.testPositivityRatioDetails.source: chr  "other" "other" "other" "other" ...
## $ metrics.caseDensity    : num  62.3 27.7 37.9 26.1 46.1 8.7 14 30.6 49.9 35.3 ..
## $ metrics.contactTracerCapacityRatio : num  0.21 0.09 0.05 0.12 0.14 0.28 0.38 0.08 0.06 0.03...
## $ metrics.infectionRate  : num  0.9 0.96 0.98 0.95 1.03 0.78 0.9 0.93 0.89 0.97 .
## $ metrics.infectionRateCI90 : num  0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 ...
## $ metrics.icuCapacityRatio : num  0.72 0.75 0.82 0.69 0.68 0.73 0.75 0.83 0.77 0.63...
## $ riskLevels.overall     : int  3 3 3 3 3 1 2 3 3 3 ...
## $ riskLevels.testPositivityRatio : int  1 1 2 1 2 1 0 1 2 2 ...
```

```
## $ riskLevels.caseDensity : int 3 3 3 3 3 1 2 3 3 3 ...
## $ riskLevels.contactTracerCapacityRatio : int 1 2 2 1 1 1 1 2 2 2 ...
## $ riskLevels.infectionRate : int 0 1 1 1 1 0 0 1 0 1 ...
## $ riskLevels.icuCapacityRatio : int 1 1 2 0 0 1 1 2 1 0 ...
## $ actuals.cases : int 145038 431258 1253227 359733 542639 3635410 48691...
## $ actuals.deaths : int 1753 6268 23342 4295 3149 58933 71639 30976 4292...
## $ actuals.positiveTests : int 120219 394797 1337786 319838 622531 4290991 48474...
## $ actuals.negativeTests : int 1784431 3685974 16213483 4920467 4722714 37126660...
## $ actuals.contactTracers : int 500 359 1000 656 1200 2600 10600 1508 248 396 ...
## $ actuals.hospitalBeds.capacity : int 2060 7949 22116 6869 5571 53036 61790 29039 5961 ...
## $ actuals.hospitalBeds.currentUsageTotal : int 1376 4838 17510 5361 3373 40223 47332 23890 4488 ...
## $ actuals.hospitalBeds.currentUsageCovid : int 162 423 2184 611 538 2276 3992 3136 751 1657 ...
## $ actuals.icuBeds.capacity : int 83 787 2514 795 616 6264 6943 3462 669 2150 ...
## $ actuals.icuBeds.currentUsageTotal : int 60 594 2053 552 420 4577 5230 2859 518 1348 ...
## $ actuals.icuBeds.currentUsageCovid : int 55 125 590 123 187 563 961 667 203 401 ...
## $ actuals.newCases : int 255 1498 8557 2207 3637 4154 913 2425 877 158 ...
## $ actuals.vaccinesDistributed : int 997460 4148765 14303410 7108555 4278230 34664565 ...
## $ actuals.vaccinationsInitiated : int 403191 1804886 5781965 2870229 1945560 14757995 2...
## $ actuals.vaccinationsCompleted : int 349427 1540988 5322469 2638467 1700674 12756457 2...
## $ lastUpdatedDate : chr "#####" "#####" "#####" "##### " ...
## $ metrics.vaccinationsInitiatedRatio : num 0.529 0.62 0.579 0.681 0.607 0.687 0.74 0.765 0.4...
## $ metrics.vaccinationsCompletedRatio : num 0.459 0.529 0.533 0.626 0.53 0.594 0.609 0.601 0.4...
## $ actuals.newDeaths : int 0 30 56 11 21 130 111 4 29 11 ...
## $ actuals.vaccinesAdministered : int 799769 3277118 11268319 5534888 3641273 28141859 ...
## $ cdcTransmissionLevel : int 3 3 3 3 3 2 2 3 3 3 ...
```

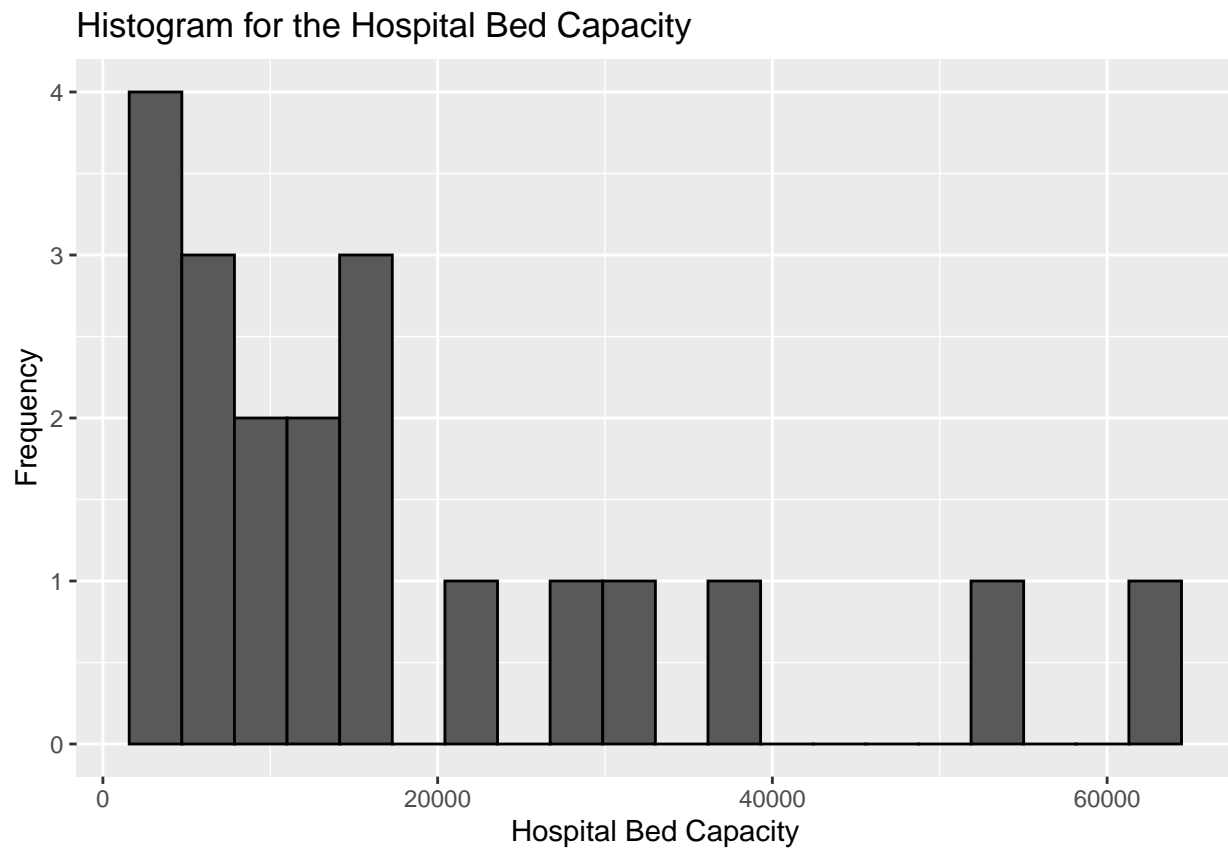
```
sample_covid_clean %>%
  ggplot(aes(riskLevels.overall))+
  geom_histogram(bins = 20,col="Black") +
  xlab("Overall Risk Level")+
  ylab("Frequency")+
  labs(title = "Histogram for Overall Risk Levels")
```



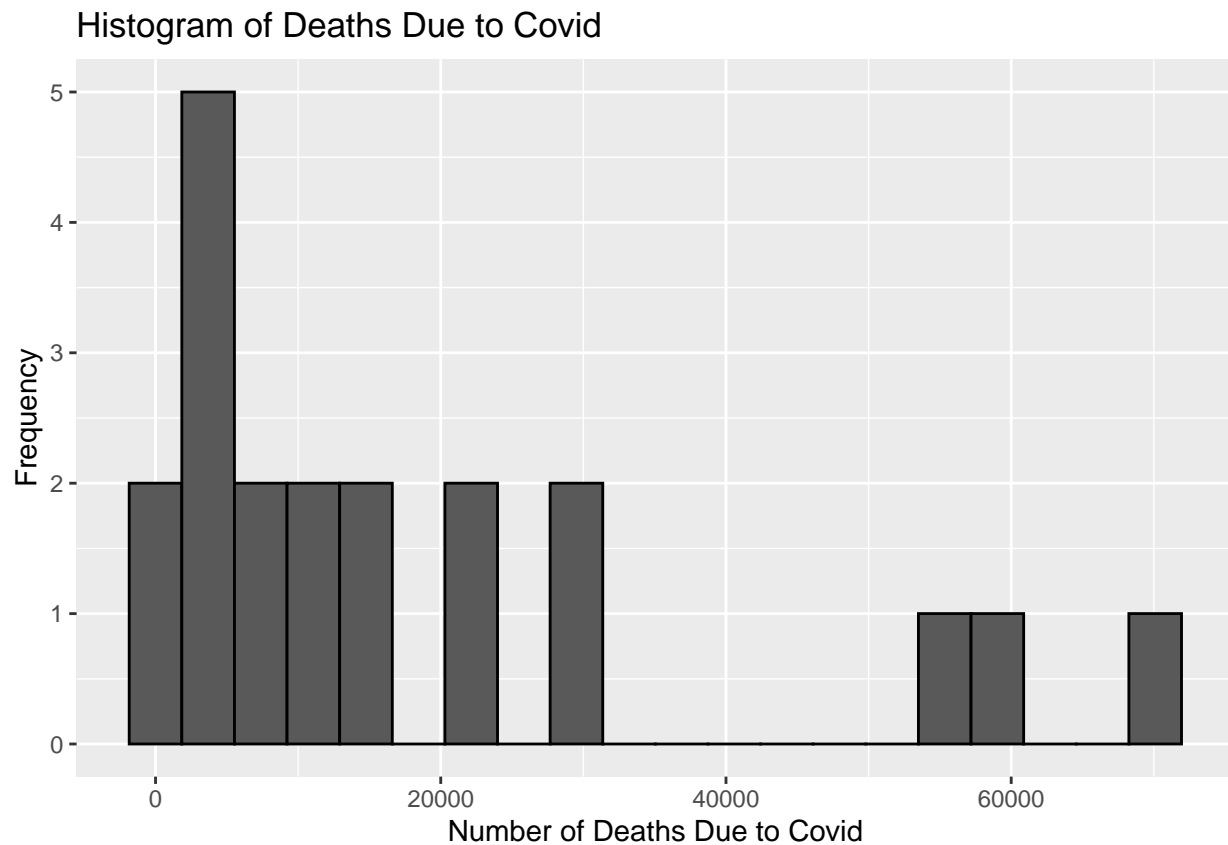
```
sample_covid_clean %>%  
  ggplot(aes(metrics.vaccinationsCompletedRatio))+  
  geom_histogram(bins = 20,col="Black") +  
  xlab("Ratio of Vaccinations Completed")+  
  ylab("Frequency")+  
  labs(title = "Histogram for the Vaccinations Completed Ratio")
```



```
sample_covid_clean %>%  
  ggplot(aes(actuals.hospitalBeds.capacity))+  
  geom_histogram(bins = 20, col="Black") +  
  xlab("Hospital Bed Capacity")+  
  ylab("Frequency")+  
  labs(title = "Histogram for the Hospital Bed Capacity")
```

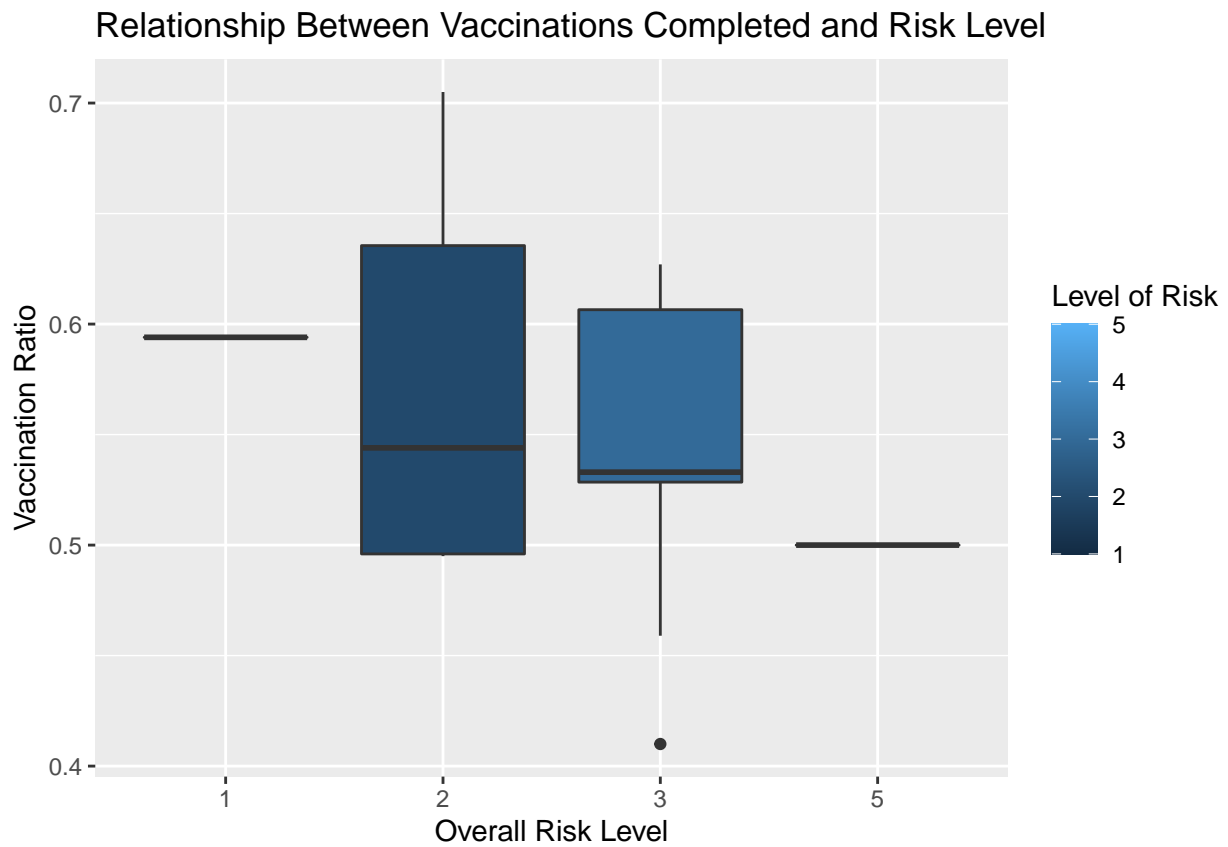


```
sample_covid_clean %>%  
  ggplot(aes(actuals.deaths))+  
  geom_histogram(bins = 20, col="Black") +  
  xlab("Number of Deaths Due to Covid")+  
  ylab("Frequency")+  
  labs(title = "Histogram of Deaths Due to Covid")
```



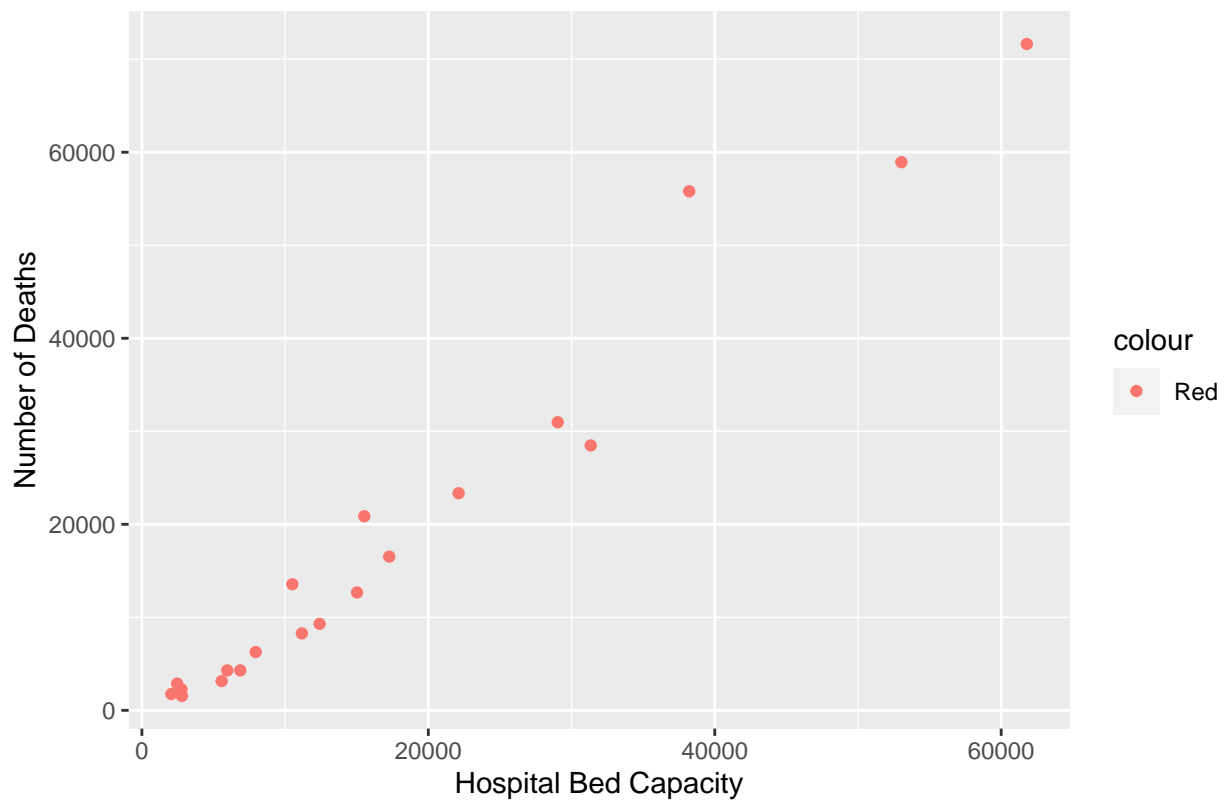
```
riskLevels.overall <- factor(riskLevels.overall, levels=c('1', '2', '3', '4', '5'))

sample_covid_clean %>%
  ggplot()+
  geom_boxplot(aes(x = factor(riskLevels.overall), y = metrics.vaccinationsCompletedRatio, fill=riskLevels.overall))
  xlab("Overall Risk Level")+
  ylab("Vaccination Ratio")+
  labs(title = "Relationship Between Vaccinations Completed and Risk Level",
       fill = "Level of Risk")
```



```
sample_covid_clean %>%
  ggplot(aes(actuals.hospitalBeds.capacity, actuals.deaths, col='Red'))+
  geom_point()+
  xlab("Hospital Bed Capacity")+
  ylab("Number of Deaths")+
  labs(title = "Relationship Between the Hospital Bed Capacity and Number of Deaths")
```

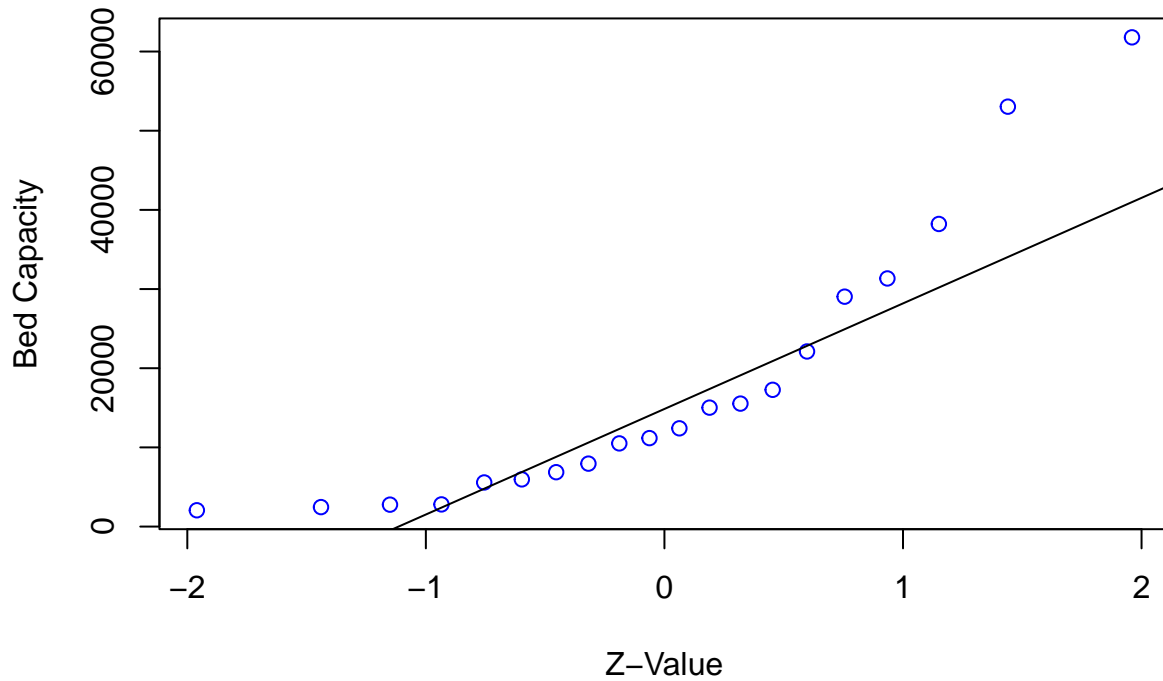
Relationship Between the Hospital Bed Capacity and Number of Deaths



```
print("correlation coefficient between actuals.hospitalBeds.capacity and actuals.deaths:");  
## [1] "correlation coefficient between actuals.hospitalBeds.capacity and actuals.deaths:"  
print(cor(actuals.hospitalBeds.capacity, actuals.deaths))  
## [1] 0.9826945  
qqnorm(actuals.hospitalBeds.capacity, main='Hospital Bed Capacity QQ Plot', xlab='Z-Value', ylab='Bed C  
qqline(actuals.hospitalBeds.capacity)
```

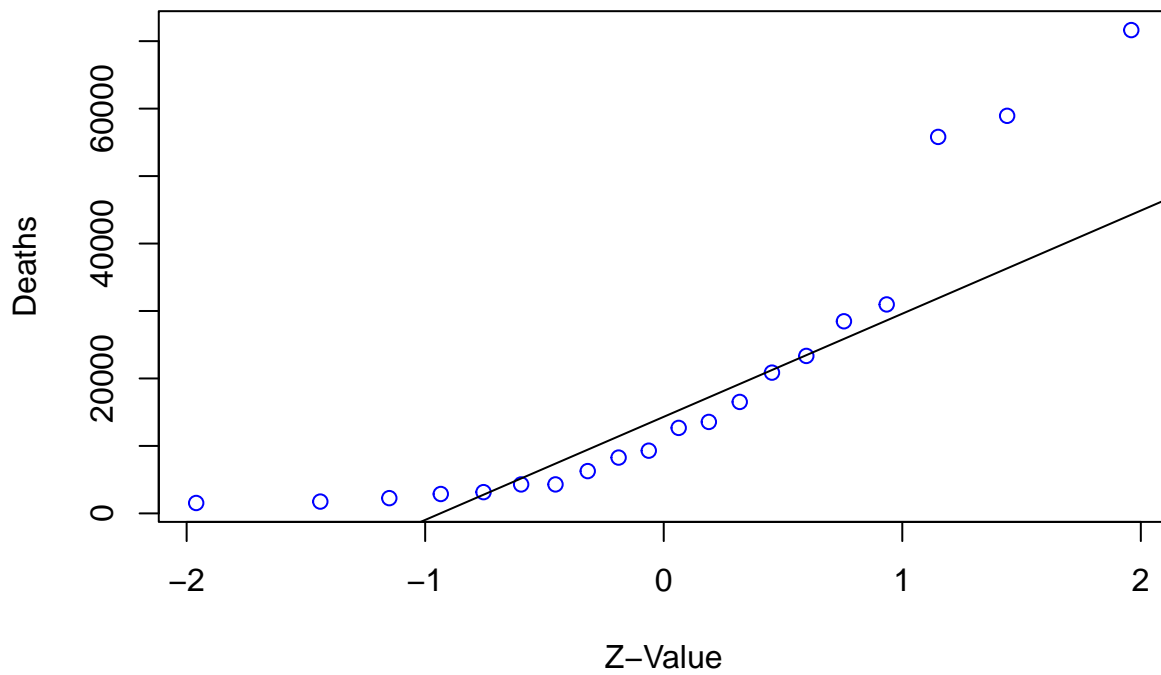


### Hospital Bed Capacity QQ Plot



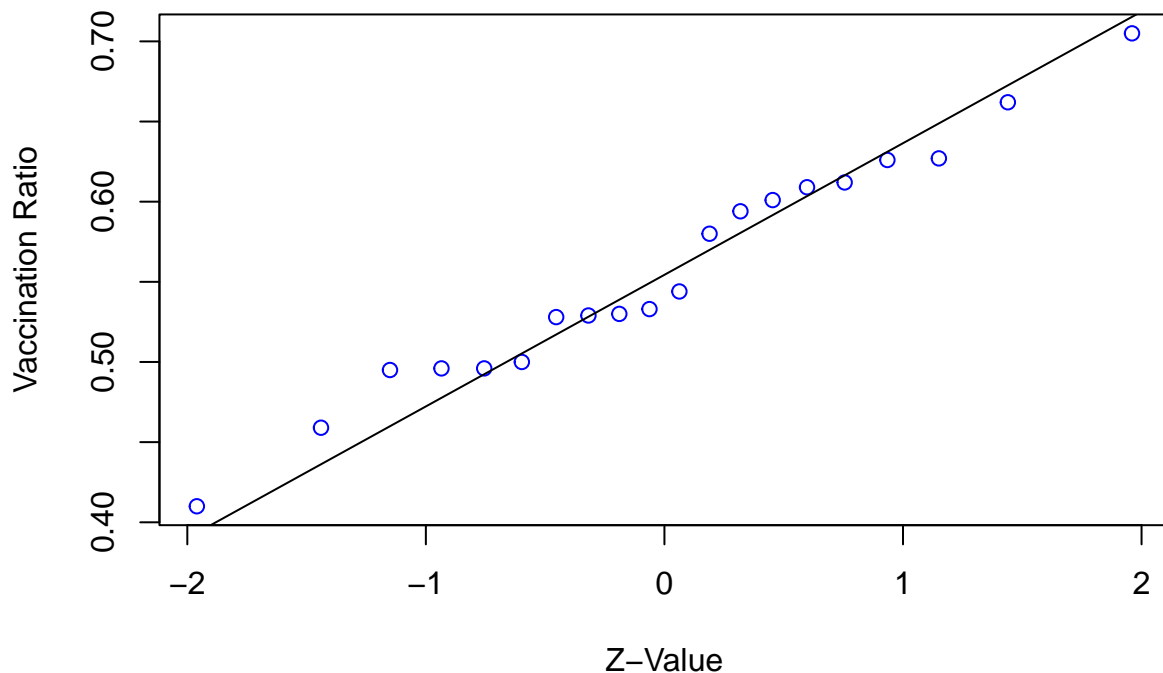
```
qqnorm(actuals.deaths, main='Total Covid Deaths QQ Plot', xlab='Z-Value', ylab='Deaths', col='Blue')  
qqline(actuals.deaths)
```

### Total Covid Deaths QQ Plot



```
qqnorm(metrics.vaccinationsCompletedRatio, main='Vaccination Ratio QQ Plot', xlab='Z-Value', ylab='Vaccination Ratio')  
qqline(metrics.vaccinationsCompletedRatio)
```

## Vaccination Ratio QQ Plot



```
print('The actual.hospitalBeds.capacity variable is non-normally distributed, indicated by the QQ plot :')

## [1] "The actual.hospitalBeds.capacity variable is non-normally distributed, indicated by the QQ plot"
shapiro.test(actuals.hospitalBeds.capacity)

##
##  Shapiro-Wilk normality test
##
## data:  actuals.hospitalBeds.capacity
## W = 0.82762, p-value = 0.0023
#actual.hospitalBeds.capacity p_value < .05, So, we can reject H0;normality and accept HA; non-normal d
writeLines('-----')

## -----

print('The actuals.deaths variable is non-normally distributed, indicated by the QQ plot and the result:')

## [1] "The actuals.deaths variable is non-normally distributed, indicated by the QQ plot and the result"
shapiro.test(actuals.deaths)

##
##  Shapiro-Wilk normality test
##
## data:  actuals.deaths
## W = 0.788, p-value = 0.0005753
#actuals.deaths p_value < .05, So, we can reject H0;normality and accept HA; non-normal distribution
writeLines('-----')

## -----
```

```

print('The metrics.vaccinationsCompletedRatio variable is normally distributed, indicated by the QQ plot')

## [1] "The metrics.vaccinationsCompletedRatio variable is normally distributed, indicated by the QQ plot"
shapiro.test(metrics.vaccinationsCompletedRatio)

##
## Shapiro-Wilk normality test
##
## data: metrics.vaccinationsCompletedRatio
## W = 0.97757, p-value = 0.899
#metrics.vaccinationsCompletedRatio p_value > .05, So we can fail to reject H0; normal distribution

writeLines('riskLevels.overall descriptive statistics:\n')

## riskLevels.overall descriptive statistics:
summary(sample_covid_clean$riskLevels.overall)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   2.00   3.00   2.65   3.00   5.00
writeLines('-----\n')

## -----
writeLines('metrics.vaccinationsCompletedRatio descriptive statistics:\n')

## metrics.vaccinationsCompletedRatio descriptive statistics:
summary(sample_covid_clean$metrics.vaccinationsCompletedRatio)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.4100  0.4990  0.5385  0.5568  0.6098  0.7050
writeLines('-----\n')

## -----
writeLines('actuals.deaths descriptive statistics:\n')

## actuals.deaths descriptive statistics:
summary(sample_covid_clean$actuals.deaths)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1543   4006   10986   18840   24626   71639
writeLines('-----\n')

## -----
writeLines('actuals.hospitalBeds.capacity descriptive statistics:\n')

## actuals.hospitalBeds.capacity descriptive statistics:
summary(sample_covid_clean$actuals.hospitalBeds.capacity)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      2060   5864   11792   17695   23847   61790
install.packages("pastecs")

```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(pastecs)
```

```
##
```

```
## Attaching package: 'pastecs'
```

```
## The following objects are masked from 'package:dplyr':
```

```
##
```

```
## first, last
```

```
variables_of_interest=sample_covid_clean[1:20,c(10,17,21,33)]
```

```
#variables_of_interest
```

```
summary_stats <- stat.desc(variables_of_interest)
```

```
round(summary_stats,2)
```

```
##          riskLevels.overall actuals.deaths actuals.hospitalBeds.capacity
## nbr.val          20.00          20.00          20.00
## nbr.null          0.00          0.00          0.00
## nbr.na            0.00          0.00          0.00
## min              1.00         1543.00         2060.00
## max              5.00         71639.00        61790.00
## range            4.00         70096.00        59730.00
## sum              53.00        376802.00       353894.00
## median           3.00         10986.00        11791.50
## mean             2.65         18840.10        17694.70
## SE.mean          0.18         4659.06         3804.88
## CI.mean.0.95     0.38         9751.53         7963.70
## var              0.66        434136916.20      289542154.22
## std.dev           0.81         20835.95         17015.94
## coef.var          0.31          1.11          0.96
```

```
##          metrics.vaccinationsCompletedRatio
```

```
## nbr.val          20.00
## nbr.null          0.00
## nbr.na            0.00
## min              0.41
## max              0.70
## range            0.30
## sum              11.14
## median           0.54
## mean             0.56
## SE.mean          0.02
## CI.mean.0.95     0.03
## var              0.01
## std.dev           0.07
## coef.var          0.13
```

```
all_quantitative_variables=sample_covid_clean[1:20,c(2,3,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100)]
```

```
#all_quantitative_variables
```

```
all_stats <- stat.desc(all_quantitative_variables)
```

```
round(all_stats,1)
```

```
##          population metrics.testPositivityRatio metrics.caseDensity
## nbr.val          2.000000e+01          20.0          20.0
## nbr.null          0.000000e+00          0.0          0.0
## nbr.na            0.000000e+00          0.0          0.0
```

```

## min          7.620620e+05          0.0          8.7
## max          3.951222e+07          0.2          80.9
## range        3.875016e+07          0.1          72.2
## sum          1.691615e+08          1.5          639.2
## median       5.790585e+06          0.1          28.4
## mean         8.458075e+06          0.1          32.0
## SE.mean      2.095344e+06          0.0          4.0
## CI.mean.0.95 4.385605e+06          0.0          8.4
## var          8.780931e+13          0.0          319.3
## std.dev      9.370662e+06          0.0          17.9
## coef.var     1.100000e+00          0.5          0.6
##
## metrics.contactTracerCapacityRatio metrics.infectionRate
## nbr.val              19.0              20.0
## nbr.null             0.0              0.0
## nbr.na               1.0              0.0
## min                 0.0              0.8
## max                 0.5              1.0
## range              0.5              0.3
## sum                2.8              18.4
## median             0.1              0.9
## mean              0.1              0.9
## SE.mean           0.0              0.0
## CI.mean.0.95      0.1              0.0
## var              0.0              0.0
## std.dev          0.1              0.1
## coef.var         0.9              0.1
##
## metrics.infectionRateCI90 metrics.icuCapacityRatio
## nbr.val              20.0              19.0
## nbr.null             0.0              0.0
## nbr.na               0.0              1.0
## min                 0.1              0.6
## max                 0.1              0.9
## range              0.0              0.2
## sum                2.0              14.3
## median             0.1              0.8
## mean              0.1              0.8
## SE.mean           0.0              0.0
## CI.mean.0.95      0.0              0.0
## var              0.0              0.0
## std.dev          0.0              0.1
## coef.var         0.0              0.1
##
## riskLevels.overall riskLevels.testPositivityRatio
## nbr.val              20.0              20.0
## nbr.null             0.0              4.0
## nbr.na               0.0              0.0
## min                 1.0              0.0
## max                 5.0              2.0
## range              4.0              2.0
## sum               53.0              21.0
## median             3.0              1.0
## mean              2.6              1.0
## SE.mean           0.2              0.2
## CI.mean.0.95      0.4              0.3
## var              0.7              0.5

```

## std.dev	0.8	0.7	
## coef.var	0.3	0.7	
##	riskLevels.caseDensity	riskLevels.contactTracerCapacityRatio	
## nbr.val	20.0	20.0	
## nbr.null	0.0	0.0	
## nbr.na	0.0	0.0	
## min	1.0	1.0	
## max	5.0	4.0	
## range	4.0	3.0	
## sum	53.0	33.0	
## median	3.0	2.0	
## mean	2.6	1.6	
## SE.mean	0.2	0.2	
## CI.mean.0.95	0.4	0.3	
## var	0.7	0.6	
## std.dev	0.8	0.7	
## coef.var	0.3	0.5	
##	riskLevels.infectionRate	riskLevels.icuCapacityRatio	actuals.cases
## nbr.val	20.0	20.0	2.000000e+01
## nbr.null	8.0	5.0	0.000000e+00
## nbr.na	0.0	0.0	0.000000e+00
## min	0.0	0.0	1.324410e+05
## max	1.0	4.0	4.869133e+06
## range	1.0	4.0	4.736692e+06
## sum	12.0	26.0	2.328107e+07
## median	1.0	1.0	8.687210e+05
## mean	0.6	1.3	1.164053e+06
## SE.mean	0.1	0.3	2.764230e+05
## CI.mean.0.95	0.2	0.5	5.785601e+05
## var	0.3	1.3	1.528194e+12
## std.dev	0.5	1.1	1.236201e+06
## coef.var	0.8	0.9	1.100000e+00
##	actuals.deaths	actuals.positiveTests	actuals.negativeTests
## nbr.val	20.0	2.000000e+01	2.000000e+01
## nbr.null	0.0	0.000000e+00	0.000000e+00
## nbr.na	0.0	0.000000e+00	0.000000e+00
## min	1543.0	1.096130e+05	1.408178e+06
## max	71639.0	4.847452e+06	8.132844e+07
## range	70096.0	4.737839e+06	7.992026e+07
## sum	376802.0	2.446771e+07	3.201007e+08
## median	10986.0	9.219545e+05	8.942334e+06
## mean	18840.1	1.223385e+06	1.600503e+07
## SE.mean	4659.1	2.952803e+05	4.623258e+06
## CI.mean.0.95	9751.5	6.180287e+05	9.676591e+06
## var	434136916.2	1.743809e+12	4.274904e+14
## std.dev	20836.0	1.320534e+06	2.067584e+07
## coef.var	1.1	1.100000e+00	1.300000e+00
##	actuals.contactTracers	actuals.hospitalBeds.capacity	
## nbr.val	20.0	20.0	
## nbr.null	0.0	0.0	
## nbr.na	0.0	0.0	
## min	94.0	2060.0	
## max	10600.0	61790.0	
## range	10506.0	59730.0	

```

## sum          37396.0          353894.0
## median       715.5           11791.5
## mean         1869.8          17694.7
## SE.mean      663.4           3804.9
## CI.mean.0.95 1388.5          7963.7
## var          8802211.6        289542154.2
## std.dev      2966.9          17015.9
## coef.var     1.6             1.0
##             actuals.hospitalBeds.currentUsageTotal
## nbr.val      20.0
## nbr.null     0.0
## nbr.na       0.0
## min          1376.0
## max          47332.0
## range        45956.0
## sum          266970.0
## median       8237.5
## mean         13348.5
## SE.mean      2960.8
## CI.mean.0.95 6197.0
## var          175327300.6
## std.dev      13241.1
## coef.var     1.0
##             actuals.hospitalBeds.currentUsageCovid actuals.icuBeds.capacity
## nbr.val      20.0          20.0
## nbr.null     0.0          0.0
## nbr.na       0.0          0.0
## min          99.0          83.0
## max          3992.0         6943.0
## range        3893.0         6860.0
## sum          25653.0         40623.0
## median       1083.5         1449.0
## mean         1282.7         2031.2
## SE.mean      230.4          437.3
## CI.mean.0.95 482.2          915.4
## var          1061484.9        3825240.0
## std.dev      1030.3          1955.8
## coef.var     0.8            1.0
##             actuals.icuBeds.currentUsageTotal
## nbr.val      20.0
## nbr.null     0.0
## nbr.na       0.0
## min          60.0
## max          5230.0
## range        5170.0
## sum          30083.0
## median       1055.5
## mean         1504.2
## SE.mean      325.9
## CI.mean.0.95 682.1
## var          2124247.7
## std.dev      1457.5
## coef.var     1.0
##             actuals.icuBeds.currentUsageCovid actuals.newCases

```

## nbr.val	20.0	20.0
## nbr.null	0.0	0.0
## nbr.na	0.0	0.0
## min	13.0	158.0
## max	961.0	8557.0
## range	948.0	8399.0
## sum	6429.0	54837.0
## median	300.5	2316.0
## mean	321.4	2741.8
## SE.mean	53.3	513.0
## CI.mean.0.95	111.6	1073.8
## var	56877.0	5264078.6
## std.dev	238.5	2294.4
## coef.var	0.7	0.8
##	actuals.vaccinesDistributed	actuals.vaccinationsInitiated
## nbr.val	2.000000e+01	2.000000e+01
## nbr.null	0.000000e+00	0.000000e+00
## nbr.na	0.000000e+00	0.000000e+00
## min	9.974600e+05	4.031910e+05
## max	6.148694e+07	2.923977e+07
## range	6.048948e+07	2.883658e+07
## sum	2.582306e+08	1.149908e+08
## median	8.159550e+06	3.531788e+06
## mean	1.291153e+07	5.749538e+06
## SE.mean	3.314499e+06	1.559074e+06
## CI.mean.0.95	6.937325e+06	3.263179e+06
## var	2.197180e+14	4.861423e+13
## std.dev	1.482289e+07	6.972391e+06
## coef.var	1.100000e+00	1.200000e+00
##	actuals.vaccinationsCompleted	metrics.vaccinationsInitiatedRatio
## nbr.val	2.000000e+01	20.0
## nbr.null	0.000000e+00	0.0
## nbr.na	0.000000e+00	0.0
## min	3.494270e+05	0.5
## max	2.404525e+07	0.8
## range	2.369582e+07	0.3
## sum	9.836978e+07	12.8
## median	3.188114e+06	0.6
## mean	4.918489e+06	0.6
## SE.mean	1.295748e+06	0.0
## CI.mean.0.95	2.712031e+06	0.0
## var	3.357925e+13	0.0
## std.dev	5.794760e+06	0.1
## coef.var	1.200000e+00	0.1
##	metrics.vaccinationsCompletedRatio	actuals.newDeaths
## nbr.val	20.0	20.0
## nbr.null	0.0	2.0
## nbr.na	0.0	0.0
## min	0.4	0.0
## max	0.7	130.0
## range	0.3	130.0
## sum	11.1	766.0
## median	0.5	29.5
## mean	0.6	38.3

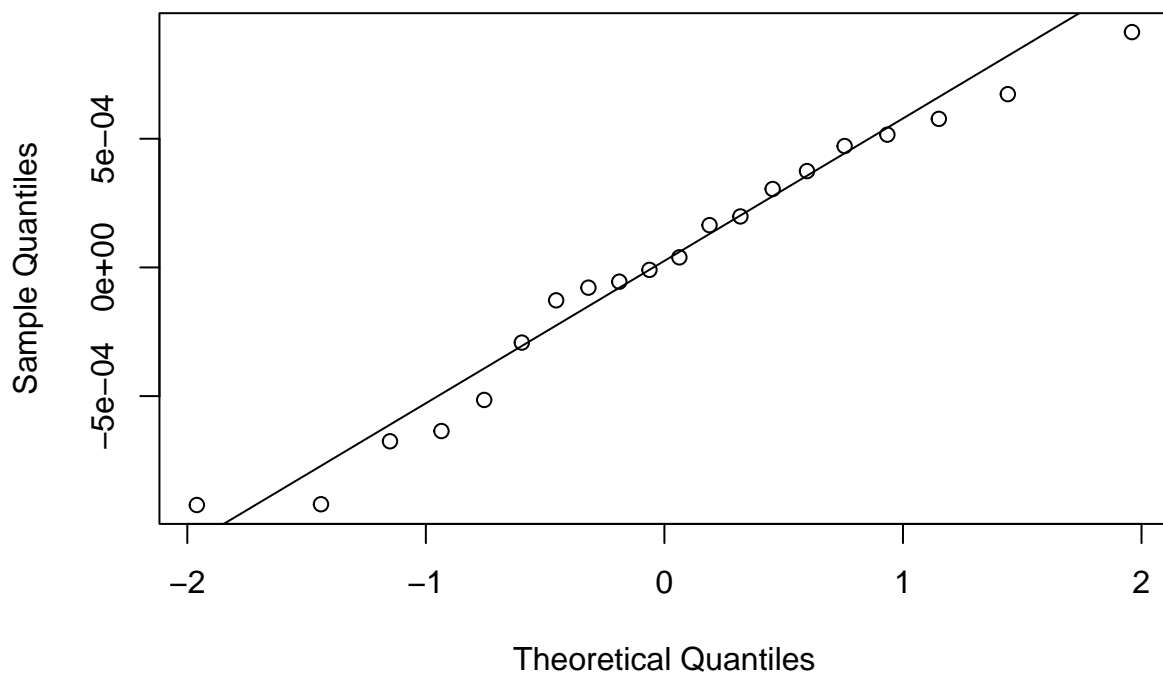


```
## SE.mean                0.0                8.4
## CI.mean.0.95           0.0                17.6
## var                    0.0                1420.9
## std.dev                 0.1                37.7
## coef.var                0.1                1.0
##          actuals.vaccinesAdministered cdcTransmissionLevel
## nbr.val                2.000000e+01                20.0
## nbr.null                0.000000e+00                0.0
## nbr.na                  0.000000e+00                0.0
## min                    7.997690e+05                2.0
## max                    5.328399e+07                3.0
## range                  5.248422e+07                1.0
## sum                    2.151226e+08                58.0
## median                 6.886770e+06                3.0
## mean                   1.075613e+07                2.9
## SE.mean                2.858378e+06                0.1
## CI.mean.0.95           5.982653e+06                0.1
## var                    1.634065e+14                0.1
## std.dev                 1.278305e+07                0.3
## coef.var                1.200000e+00                0.1

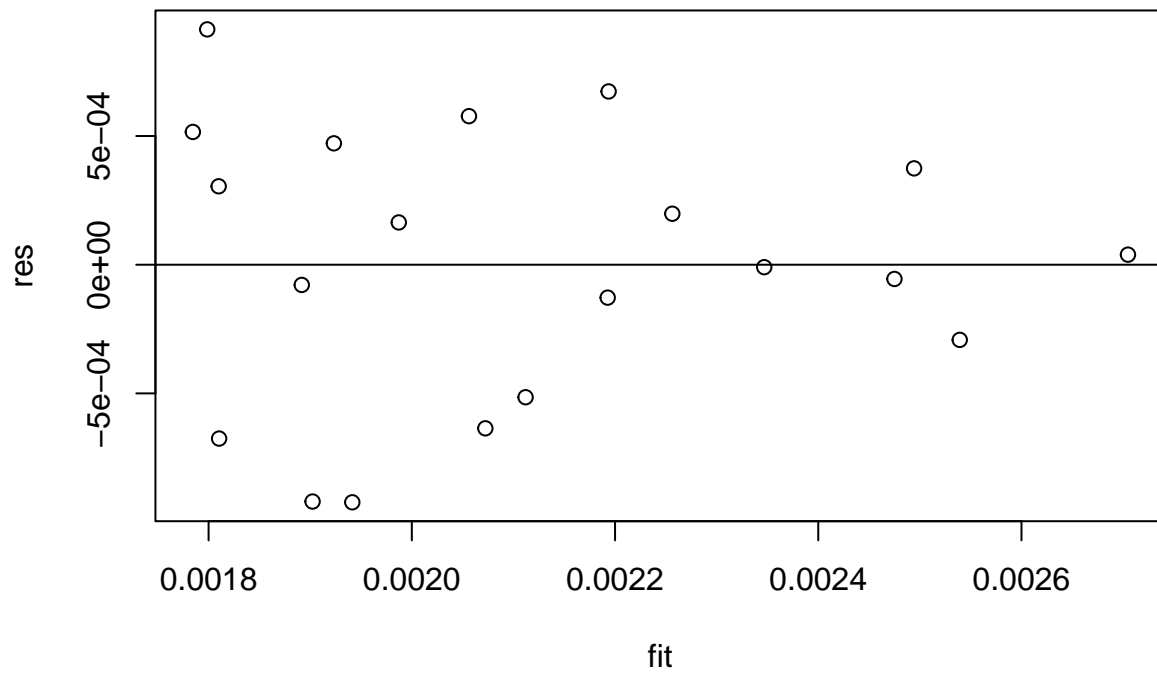
# Simple Linear Regression Model
lm.model <- lm((actuals.deaths/population)~(actuals.hospitalBeds.capacity/population))

# Check Assumptions
res <- residuals(lm.model) #residuals
fit <- fitted.values(lm.model) #fitted values
qqnorm(res) #check for normality
qqline(res)
```

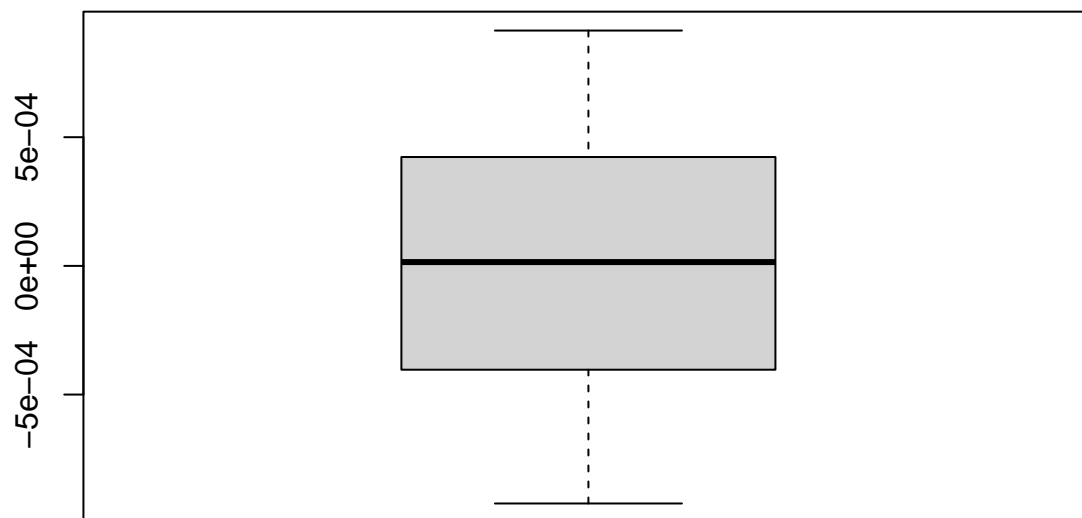
**Normal Q-Q Plot**



```
plot(res~fit) #check for homoscedasticity  
abline(h=0)
```

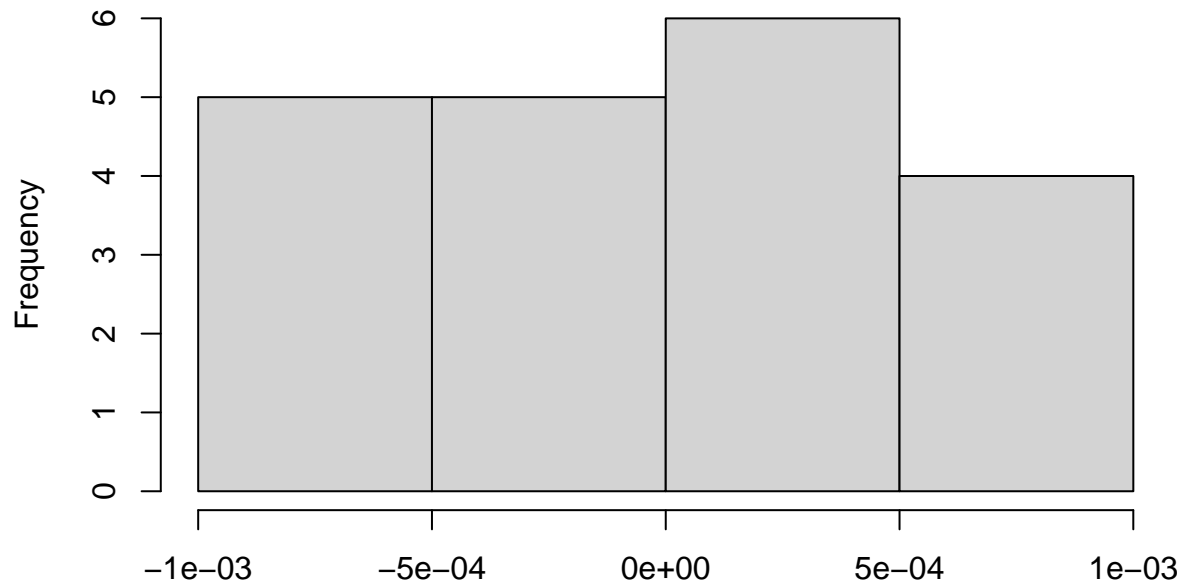


```
boxplot(res)
```



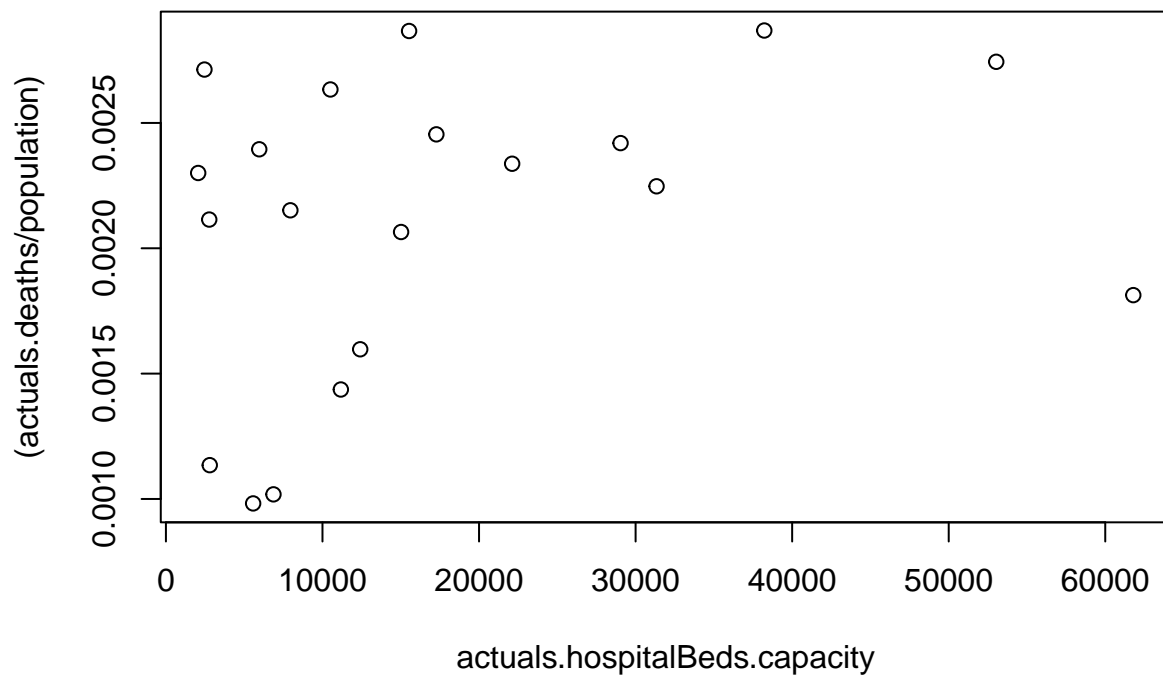
```
hist(res)
```

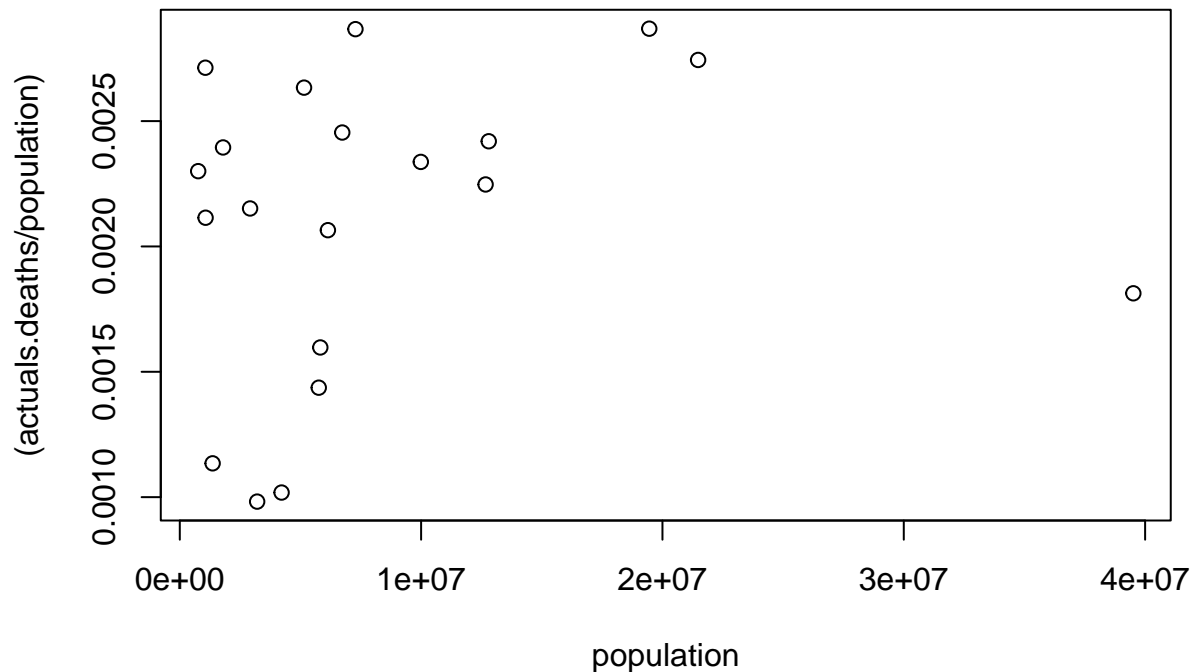
Histogram of res



res

```
plot((actuals.deaths/population)~(actuals.hospitalBeds.capacity/population)) #check if lm is appropriate
```





```
# Check p-value for hypothesis test
summary(lm.model)
```

```
##
## Call:
## lm(formula = (actuals.deaths/population) ~ (actuals.hospitalBeds.capacity/population))
##
## Residuals:
```

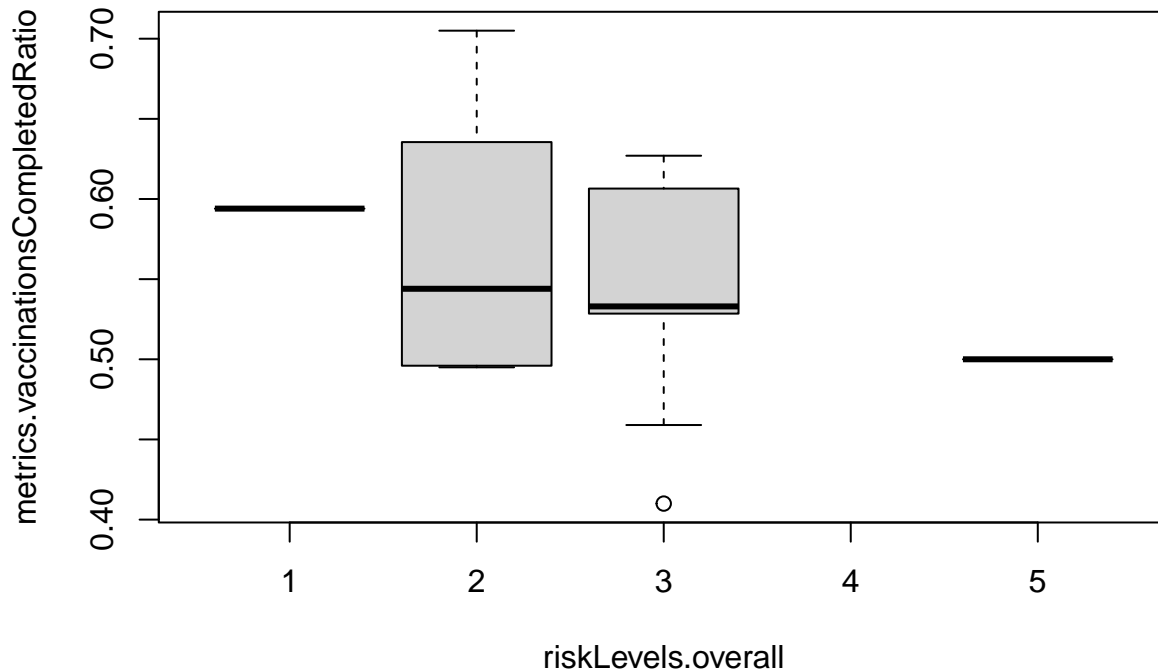
	Min	1Q	Median	3Q	Max
	-9.231e-04	-3.477e-04	1.478e-05	3.987e-04	9.143e-04

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.708e-03	2.269e-04	7.529	8.25e-07
actuals.hospitalBeds.capacity	3.763e-08	1.801e-08	2.090	0.052
actuals.hospitalBeds.capacity:population	-8.773e-16	5.258e-16	-1.669	0.114

```
##
## (Intercept) ***
## actuals.hospitalBeds.capacity .
## actuals.hospitalBeds.capacity:population
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0005608 on 17 degrees of freedom
## Multiple R-squared:  0.2152, Adjusted R-squared:  0.1229
## F-statistic: 2.331 on 2 and 17 DF,  p-value: 0.1275
```

```
boxplot(metrics.vaccinationsCompletedRatio~riskLevels.overall) # once again check box plots for a rough
```



*#Since only levels 2 & 3 have sufficient data points we will not use ANOVA*

`aggregate(metrics.vaccinationsCompletedRatio~riskLevels.overall, FUN=mean)` *#as we get a higher risk lev*

```
## riskLevels.overall metrics.vaccinationsCompletedRatio
## 1 1 0.5940000
## 2 2 0.5724286
## 3 3 0.5486364
## 4 5 0.5000000
```

`aggregate(metrics.vaccinationsCompletedRatio~riskLevels.overall, FUN=mean)[c(2,3),]` *#see avg vaccination*

```
## riskLevels.overall metrics.vaccinationsCompletedRatio
## 2 2 0.5724286
## 3 3 0.5486364
```

`aggregate(metrics.vaccinationsCompletedRatio~riskLevels.overall, FUN=sd)[c(2,3),]`

```
## riskLevels.overall metrics.vaccinationsCompletedRatio
## 2 2 0.08700739
## 3 3 0.06965956
```

*#f-test to test for equal variances*

`sample_count=count(sample_covid_clean)#check count`

`sample_ordered=sample_covid_clean[`

`with(sample_covid_clean, order(riskLevels.overall)),`

`]`

`data = sample_ordered[sample_ordered$riskLevels.overall != c(1,5), ]`

`var.test(data$metrics.vaccinationsCompletedRatio~data$riskLevels.overall, ratio=1, alternative = "two.s`

`##`

`## F test to compare two variances`

`##`

`## data: data$metrics.vaccinationsCompletedRatio by data$riskLevels.overall`

`## F = 1.5601, num df = 6, denom df = 10, p-value = 0.5091`

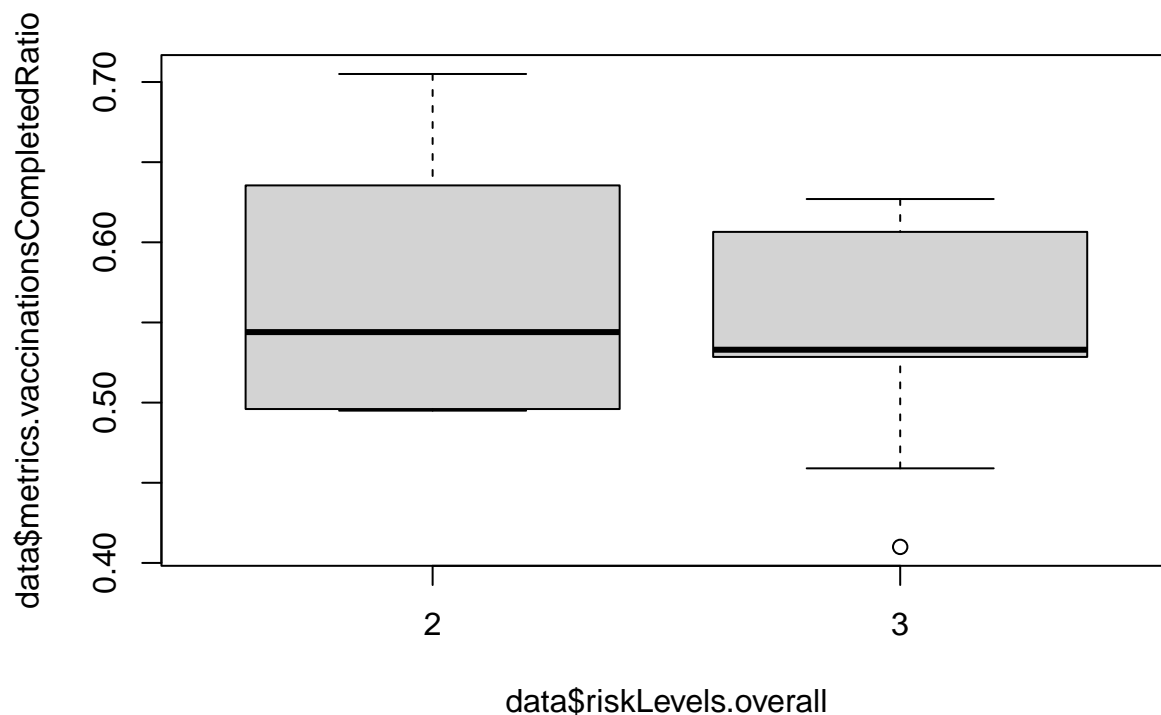
```
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
## 0.3831149 8.5201789
## sample estimates:
## ratio of variances
## 1.560094
```

*#the variances are roughly equal by teh f-test, hence we will use the pooled t-test for testing the dif*

```
aggregate(data$metrics.vaccinationsCompletedRatio~data$riskLevels.overall,FUN=mean)
```

```
## data$riskLevels.overall data$metrics.vaccinationsCompletedRatio
## 1 2 0.5724286
## 2 3 0.5486364
```

```
boxplot(data$metrics.vaccinationsCompletedRatio~data$riskLevels.overall)
```



```
t.test(data$metrics.vaccinationsCompletedRatio~data$riskLevels.overall, alt='t', conf.level=.95, var.equal=TRUE)
```

```
##
## Two Sample t-test
##
## data: data$metrics.vaccinationsCompletedRatio by data$riskLevels.overall
## t = 0.64219, df = 16, p-value = 0.5298
## alternative hypothesis: true difference in means between group 2 and group 3 is not equal to 0
## 95 percent confidence interval:
## -0.05474711 0.10233152
## sample estimates:
## mean in group 2 mean in group 3
## 0.5724286 0.5486364
```

*#By the pooled Sample t-test, we retain H0: mu1=mu2 at the .95 confidence level*