Coding and Results

McKenzie

2023-07-18

##CLEANING

*Loading Packages*

* The shelter data is being pulled from the city of Toronto’s [Daily Shelter & Overnight Service Occuapncy & Capacity](https://open.toronto.ca/dataset/daily-shelter-overnight-service-occupancy-capacity/) dataset on Toronto’s open portal website.
* The dataset is split into 3 files,1 for each year. For the 2023 year file, the data source is updated daily.

*Dealing with Missing Values*

## Index Variable\_Name Variable\_Type Sample\_n Missing\_Count  
## 1 1 X\_id integer 126588 0  
## 2 2 OCCUPANCY\_DATE Date 126588 0  
## 3 3 ORGANIZATION\_ID integer 126588 0  
## 4 4 ORGANIZATION\_NAME character 126588 0  
## 5 5 SHELTER\_ID integer 126588 0  
## 6 6 SHELTER\_GROUP character 126372 216  
## 7 7 LOCATION\_ID integer 126142 446  
## 8 8 LOCATION\_NAME character 125626 962  
## 9 9 LOCATION\_ADDRESS character 122919 3669  
## 10 10 LOCATION\_POSTAL\_CODE character 122919 3669  
## 11 11 LOCATION\_CITY character 122905 3683  
## 12 12 LOCATION\_PROVINCE character 122905 3683  
## 13 13 PROGRAM\_ID integer 126588 0  
## 14 14 PROGRAM\_NAME character 126553 35  
## 15 15 SECTOR character 126588 0  
## 16 16 PROGRAM\_MODEL character 126586 2  
## 17 17 OVERNIGHT\_SERVICE\_TYPE character 126586 2  
## 18 18 PROGRAM\_AREA character 126586 2  
## 19 19 SERVICE\_USER\_COUNT integer 126588 0  
## 20 20 CAPACITY\_TYPE character 126588 0  
## 21 21 OVER\_OCCUPIED numeric 126588 0  
## 22 22 ADDRESS character 126588 0  
## 23 23 OCCUPANCY\_RATE numeric 126588 0  
## Per\_of\_Missing No\_of\_distinct\_values  
## 1 0.000 50944  
## 2 0.000 924  
## 3 0.000 35  
## 4 0.000 35  
## 5 0.000 69  
## 6 0.002 69  
## 7 0.004 130  
## 8 0.008 130  
## 9 0.029 124  
## 10 0.029 119  
## 11 0.029 7  
## 12 0.029 2  
## 13 0.000 200  
## 14 0.000 205  
## 15 0.000 5  
## 16 0.000 3  
## 17 0.000 8  
## 18 0.000 6  
## 19 0.000 545  
## 20 0.000 2  
## 21 0.000 2  
## 22 0.000 126  
## 23 0.000 1354

* The proportion of missing values is very small. Most of the missing values are for fields related to location. Since the fields that contain missing values are non-numerical I am going to convert all missing values to “unknown”.

## [1] FALSE

## [1] FALSE

*Additional geographic information*

* Part of my analysis will involve understanding shelter related metrics across different neighborhood.
* The following code brings in location related characteristics. The [census tract boundary file](https://www12.statcan.gc.ca/census-recensement/alternative_alternatif.cfm?l=eng&dispext=zip&teng=lct_000b21a_e.zip&k=%20%20%20%2013089&loc=//www12.statcan.gc.ca/census-recensement/2021/geo/sip-pis/boundary-limites/files-fichiers/lct_000b21a_e.zip) comes from statistics Canada and the city of Toronto’s [Neighbourhood file.](https://open.toronto.ca/dataset/neighbourhoods/)

## Passing 125 addresses to the Nominatim single address geocoder

## Query completed in: 127.1 seconds

**Creating Measures of Interest**

*Creating measures*

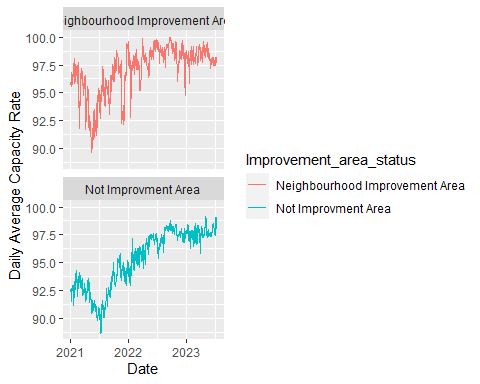
* Note all of these measures were used in the analysis.

##Neighbourhood Improvement Area

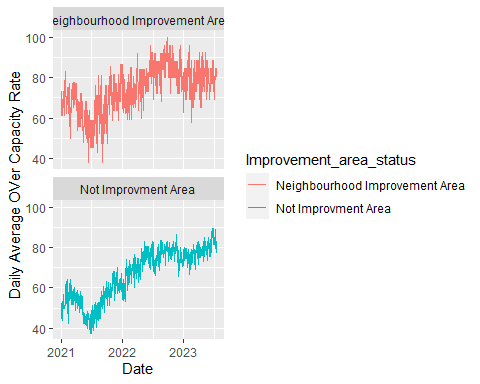
## [1] "Neighbourhood Improvement Area"   
## [2] "Not an NIA or Emerging Neighbourhood"  
## [3] "Emerging Neighbourhood"   
## [4] NA

## `summarise()` has grouped output by 'OCCUPANCY\_DATE'. You can override using  
## the `.groups` argument.

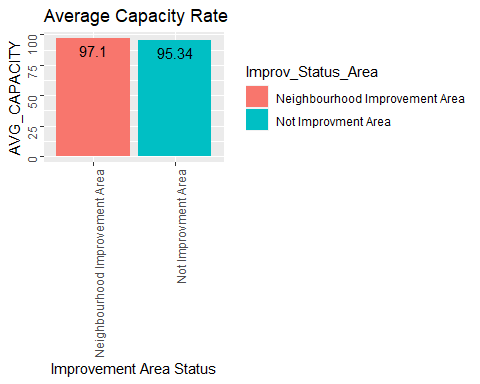
##Creating line graph for average daily capacity rates by neighbourhood improvement area  
ggplot(ImprAvgCap,aes(x=as.Date(OCCUPANCY\_DATE), y=AVG\_CAPACITY,color=Improv\_Status\_Area ))+  
 geom\_line()+  
 labs(x='Date', y='Daily Average Capacity Rate', color='Improvement\_area\_status')+  
 facet\_wrap(~Improv\_Status\_Area,ncol = 1);



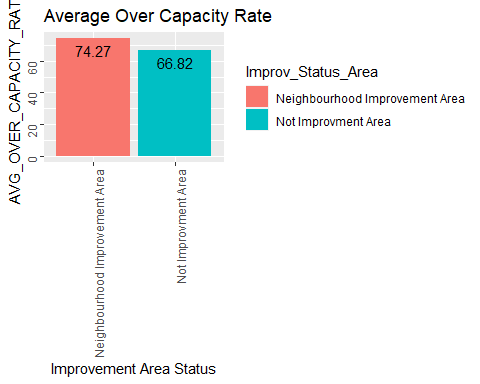
##Creating a line graph for the proportion of shelters operating at or over capacity  
ggplot(ImprAvgCap,aes(x=as.Date(OCCUPANCY\_DATE), y=OverCapacityRate,color=Improv\_Status\_Area ))+  
 geom\_line()+  
 labs(x='Date', y='Daily Average OVer Capacity Rate', color='Improvement\_area\_status')+  
 facet\_wrap(~Improv\_Status\_Area,ncol = 1)



##Aggregating to get the average of the average capacity rates for neighbourhood improvement areas and the average proporiton of shelters operating at or over capacity  
ImprOverallAvgCap<-ImprAvgCap %>%  
 group\_by(Improv\_Status\_Area) %>%  
 summarise(AVG\_CAPACITY = round(mean(AVG\_CAPACITY),digits=2),  
 AVG\_OVER\_CAPACITY\_RATE = round(mean(OverCapacityRate),digits = 2))  
  
##Creating a barchart to compare averages  
ggplot(ImprOverallAvgCap,aes(x=Improv\_Status\_Area, y=AVG\_CAPACITY,fill=Improv\_Status\_Area ))+  
 geom\_bar(stat='identity')+  
 theme(axis.text = element\_text(angle=90, hjust=1))+  
 geom\_text(aes(label = AVG\_CAPACITY),vjust=1.5,color='black')+  
 labs(title = 'Average Capacity Rate',x='Improvement Area Status',color = 'Improvement Area Status');



ggplot(ImprOverallAvgCap,aes(x=Improv\_Status\_Area, y=AVG\_OVER\_CAPACITY\_RATE,fill=Improv\_Status\_Area ))+  
 geom\_bar(stat='identity')+  
 theme(axis.text = element\_text(angle=90, hjust=1))+  
 geom\_text(aes(label = AVG\_OVER\_CAPACITY\_RATE),vjust=1.5,color='black')+  
 labs(title = 'Average Over Capacity Rate',x='Improvement Area Status',color = 'Improvement Area Status')



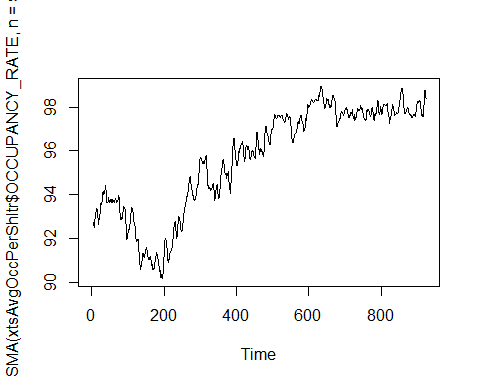
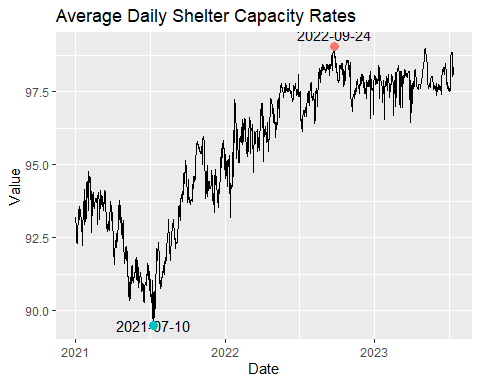
##Kruskal Wallis test   
###Test for comparing average capacity rates  
kruskal.test(AVG\_CAPACITY ~ Improv\_Status\_Area,data=ImprAvgCap)

##   
## Kruskal-Wallis rank sum test  
##   
## data: AVG\_CAPACITY by Improv\_Status\_Area  
## Kruskal-Wallis chi-squared = 263.95, df = 1, p-value < 2.2e-16

##Test for comparing the average of proportion of shelters operating at or over capacity.   
kruskal.test(OverCapacityRate ~ Improv\_Status\_Area ,data=ImprAvgCap)

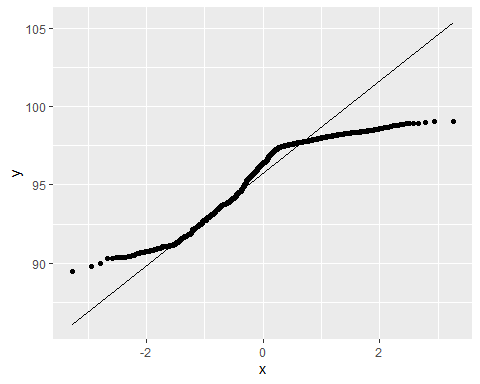
##   
## Kruskal-Wallis rank sum test  
##   
## data: OverCapacityRate by Improv\_Status\_Area  
## Kruskal-Wallis chi-squared = 144.29, df = 1, p-value < 2.2e-16

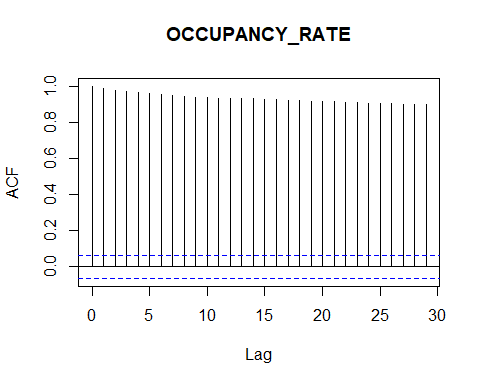
## Average Capacity Rate Trend\*\*



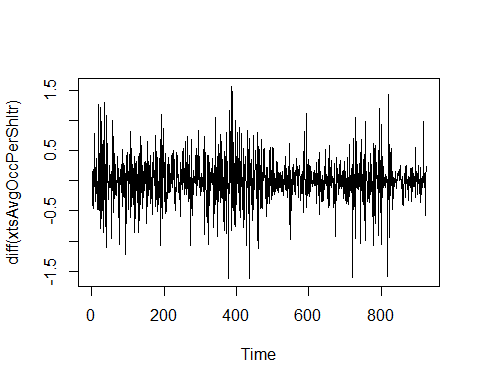
| Metric | Daily Capacity Rate |
| --- | --- |
| Mean | 95.65 |
| Median | 96.37 |
| Standard Deviation | 2.43 |
| Minimum | 89.5 |
| Maximum | 99.04 |
| Range | 89.5 - 99.04 |

**Capacity Rate Distribution**

 **ACF PLOT for Average Daily Capacity Rates**



**Differenced plot**



**Average Daily Capacity Rate: Stationairy**

## Warning in kpss.test(ts(xtsAvgOccPerShltr)): p-value smaller than printed  
## p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ts(xtsAvgOccPerShltr)  
## KPSS Level = 11.293, Truncation lag parameter = 6, p-value = 0.01

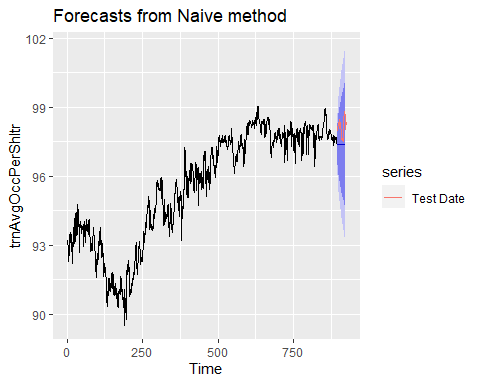
##   
## Augmented Dickey-Fuller Test  
##   
## data: ts(xtsAvgOccPerShltr)  
## Dickey-Fuller = -2.5543, Lag order = 9, p-value = 0.3437  
## alternative hypothesis: stationary

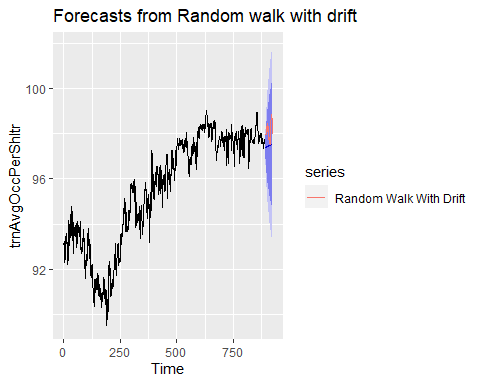
**Average Capacity RAte: Trend Assessment**

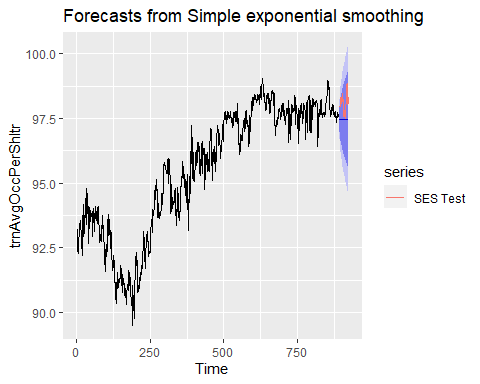
##   
## Mann-Kendall trend test  
##   
## data: ts(xtsAvgOccPerShltr$OCCUPANCY\_RATE)  
## z = 30.816, n = 924, p-value < 2.2e-16  
## alternative hypothesis: true S is not equal to 0  
## sample estimates:  
## S varS tau   
## 2.887410e+05 8.779529e+07 6.777061e-01

##   
## Sen's slope  
##   
## data: ts(xtsAvgOccPerShltr$OCCUPANCY\_RATE)  
## z = 30.816, n = 924, p-value < 2.2e-16  
## alternative hypothesis: true z is not equal to 0  
## 95 percent confidence interval:  
## 0.007096774 0.007741935  
## sample estimates:  
## Sen's slope   
## 0.007418182

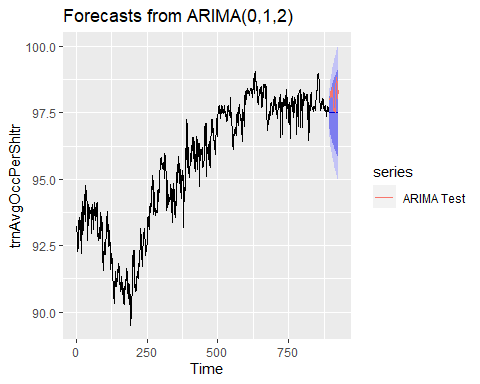
### Capacity Rate Forcasting



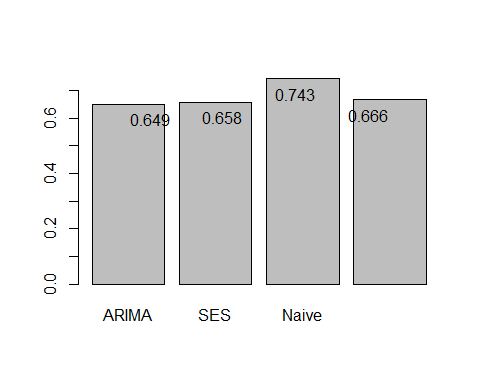


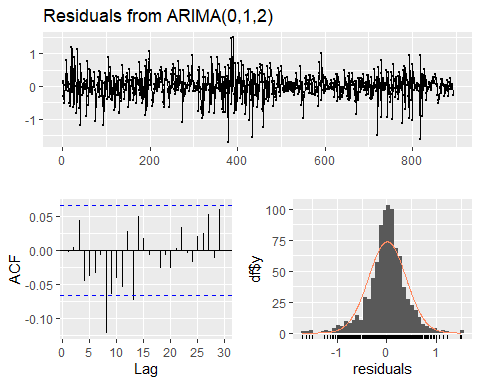


## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 894 97.48332 96.99592 97.97072 96.73791 98.22873  
## 895 97.48332 96.89580 98.07084 96.58478 98.38186  
## 896 97.48332 96.81041 98.15623 96.45420 98.51244  
## 897 97.48332 96.73470 98.23194 96.33841 98.62823  
## 898 97.48332 96.66598 98.30066 96.23330 98.73334  
## 899 97.48332 96.60260 98.36404 96.13637 98.83027  
## 900 97.48332 96.54348 98.42316 96.04596 98.92068  
## 901 97.48332 96.48787 98.47877 95.96091 99.00573  
## 902 97.48332 96.43521 98.53143 95.88037 99.08627  
## 903 97.48332 96.38507 98.58157 95.80369 99.16296  
## 904 97.48332 96.33712 98.62953 95.73035 99.23629  
## 905 97.48332 96.29109 98.67555 95.65997 99.30668  
## 906 97.48332 96.24678 98.71986 95.59220 99.37444  
## 907 97.48332 96.20400 98.76264 95.52677 99.43987  
## 908 97.48332 96.16261 98.80403 95.46347 99.50317  
## 909 97.48332 96.12248 98.84417 95.40209 99.56455  
## 910 97.48332 96.08349 98.88315 95.34247 99.62417  
## 911 97.48332 96.04556 98.92108 95.28446 99.68218  
## 912 97.48332 96.00861 98.95803 95.22795 99.73869  
## 913 97.48332 95.97256 98.99408 95.17282 99.79382  
## 914 97.48332 95.93735 99.02929 95.11897 99.84767  
## 915 97.48332 95.90293 99.06371 95.06632 99.90032  
## 916 97.48332 95.86924 99.09740 95.01480 99.95184  
## 917 97.48332 95.83624 99.13040 94.96433 100.00231  
## 918 97.48332 95.80389 99.16276 94.91485 100.05179  
## 919 97.48332 95.77214 99.19450 94.86630 100.10034  
## 920 97.48332 95.74098 99.22566 94.81864 100.14800  
## 921 97.48332 95.71037 99.25628 94.77182 100.19482  
## 922 97.48332 95.68027 99.28637 94.72579 100.24085  
## 923 97.48332 95.65067 99.31597 94.68052 100.28612



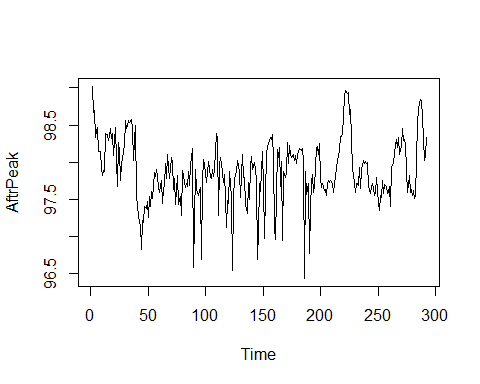
##Evaluating Model Performance: Capacity RAte



**Checking Risduals** 

##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(0,1,2)  
## Q\* = 24.499, df = 8, p-value = 0.001889  
##   
## Model df: 2. Total lags used: 10

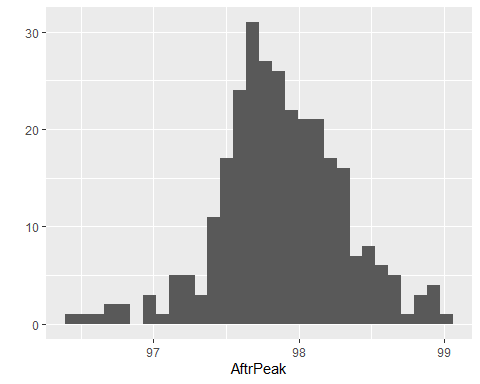
###Trend Segmentation: Average Capacity Rate

 **Distribution of Capacity Rate After PEak**

qplot(AftrPeak)

## Warning: `qplot()` was deprecated in ggplot2 3.4.0.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Don't know how to automatically pick scale for object of type <xts/zoo>.  
## Defaulting to continuous.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



**After Peak: Stationary Test**

##   
## Mann-Kendall trend test  
##   
## data: ts(AftrPeak)  
## z = 0.21291, n = 292, p-value = 0.8314  
## alternative hypothesis: true S is not equal to 0  
## sample estimates:  
## S varS tau   
## 3.560000e+02 2.780035e+06 8.408175e-03

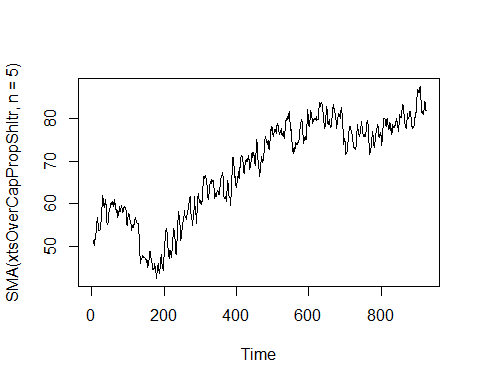
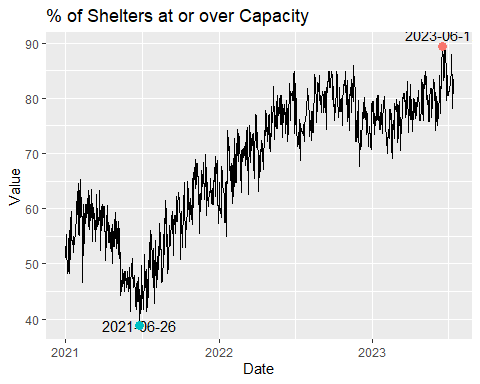
## Warning in kpss.test(ts(AftrPeak)): p-value greater than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ts(AftrPeak)  
## KPSS Level = 0.26059, Truncation lag parameter = 5, p-value = 0.1

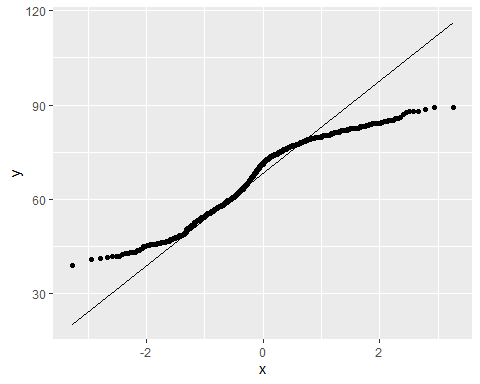
## Warning in adf.test(ts(AftrPeak)): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: ts(AftrPeak)  
## Dickey-Fuller = -4.7548, Lag order = 6, p-value = 0.01  
## alternative hypothesis: stationary

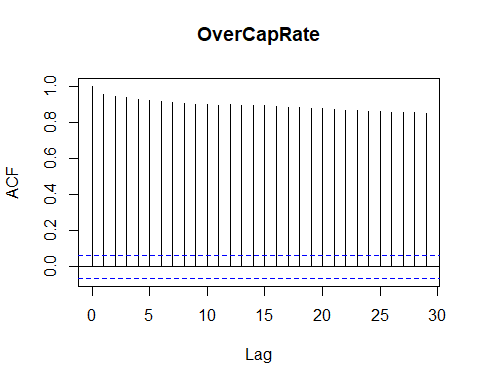
###Proportion of shelters operating at or over capacity



**Distribution: At or Over Capacity**



**ACF PLOT: At or Over Capacity Rate**



**Stationary Assessment: At or Over Capacity**

## Warning in kpss.test(ts(xtsOverCapPropShltr)): p-value smaller than printed  
## p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ts(xtsOverCapPropShltr)  
## KPSS Level = 11.275, Truncation lag parameter = 6, p-value = 0.01

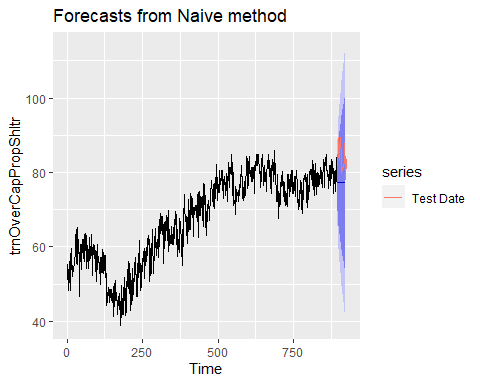
##   
## Augmented Dickey-Fuller Test  
##   
## data: ts(xtsOverCapPropShltr)  
## Dickey-Fuller = -2.8818, Lag order = 9, p-value = 0.205  
## alternative hypothesis: stationary

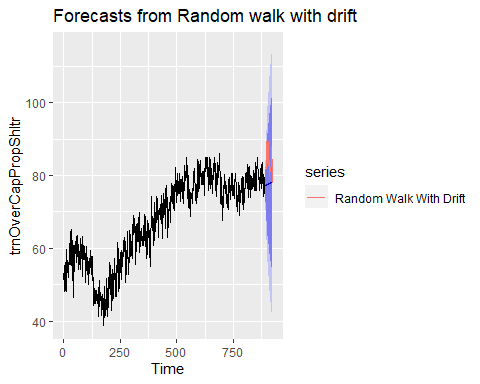
**Trend Assessment: At or Over Capacity**

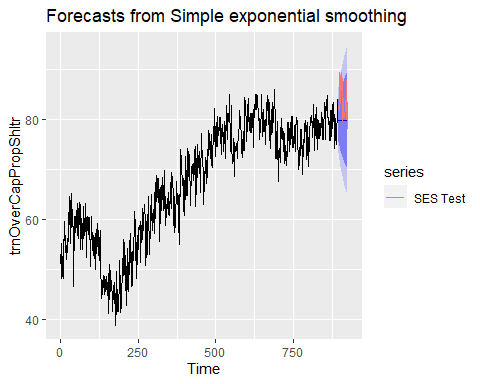
##   
## Mann-Kendall trend test  
##   
## data: ts(xtsOverCapPropShltr)  
## z = 29.986, n = 924, p-value < 2.2e-16  
## alternative hypothesis: true S is not equal to 0  
## sample estimates:  
## S varS tau   
## 2.809630e+05 8.779267e+07 6.600662e-01

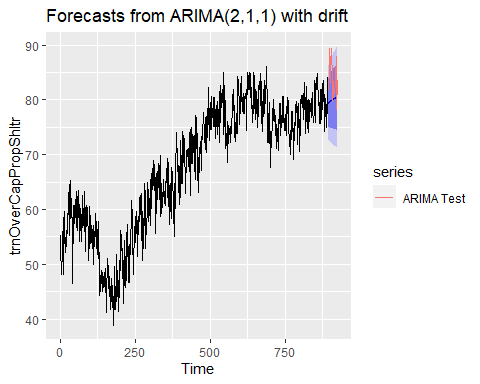
##   
## Sen's slope  
##   
## data: ts(xtsOverCapPropShltr$OverCapRate)  
## z = 29.986, n = 924, p-value < 2.2e-16  
## alternative hypothesis: true z is not equal to 0  
## 95 percent confidence interval:  
## 0.0352381 0.0384728  
## sample estimates:  
## Sen's slope   
## 0.03684211

**Forecasting: At or Over Capacity**

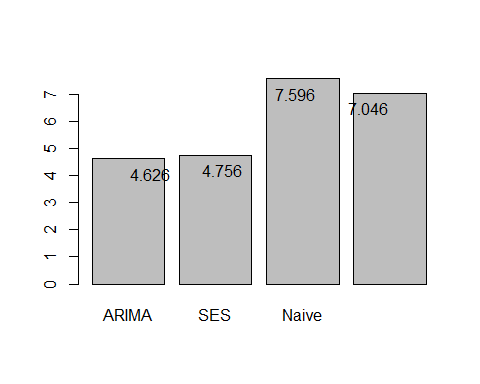




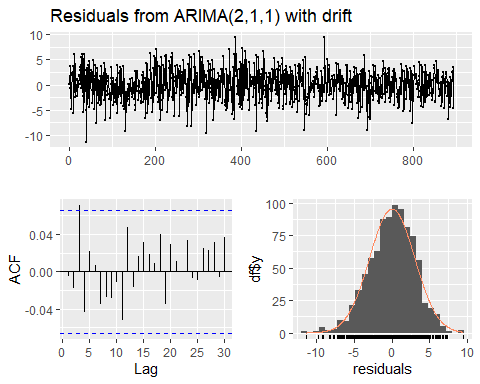




**Evaluating Model Performance: At or Over Capacity**

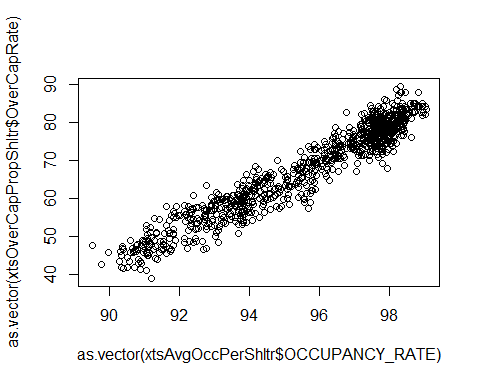


**Residuals: At or Over Capacity**



##   
## Ljung-Box test  
##   
## data: Residuals from ARIMA(2,1,1) with drift  
## Q\* = 9.6229, df = 7, p-value = 0.211  
##   
## Model df: 3. Total lags used: 10

**Correlation: Capacity Rate ~ Over Capacity**



## [1] 0.961566

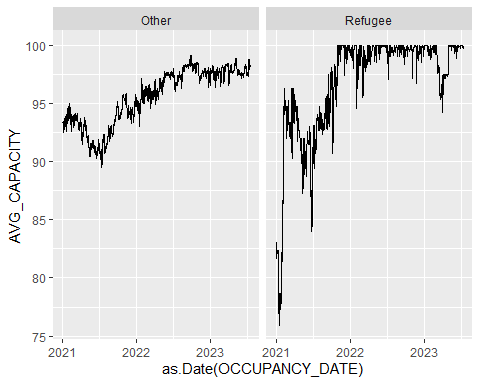
##Refugee Programs: Capacity Rate

**Looking at those more susceptible to homelessness**

# Creating Refugee identification column

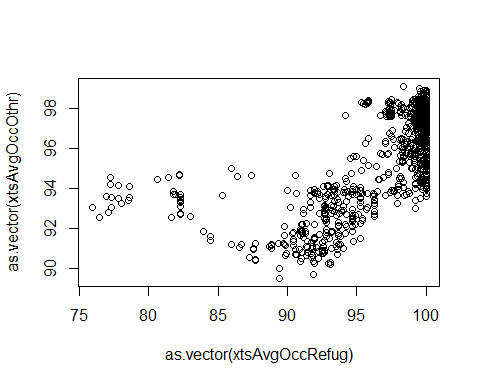
**Trend: Refugee Programs vs non-refugee programs**

## `summarise()` has grouped output by 'OCCUPANCY\_DATE'. You can override using  
## the `.groups` argument.

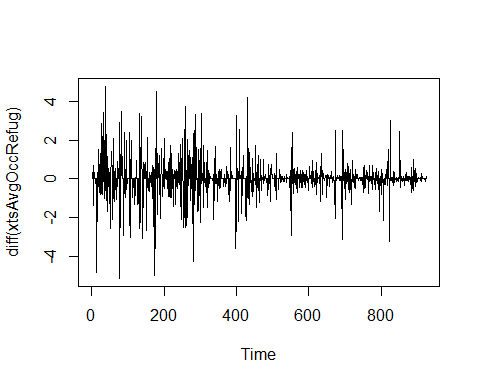
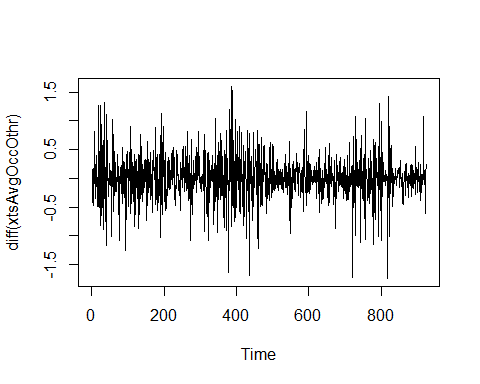


**Testing Averages**

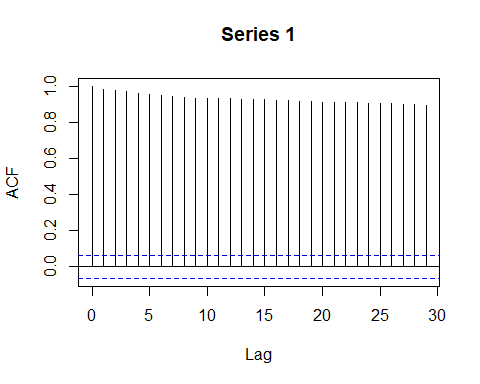
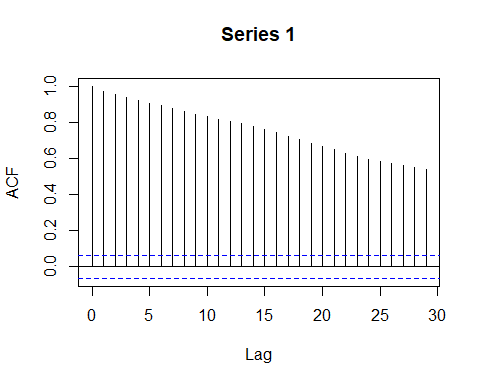
##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: as.vector(AvgOccRefug$AVG\_CAPACITY) and as.vector(AvgOccOthr$AVG\_CAPACITY)  
## W = 607031, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

**Correlation between refugee programs and other programs** 

## [1] 0.6793728

**Differenced plots in refugee capacity rates and other programs** 

**ACF PLOTS: Refugee and non-refugee programs**



**Stationary assessment: Refugee programs**

## Warning in kpss.test(ts(xtsAvgOccRefug)): p-value smaller than printed p-value

##   
## KPSS Test for Level Stationarity  
##   
## data: ts(xtsAvgOccRefug)  
## KPSS Level = 7.1999, Truncation lag parameter = 6, p-value = 0.01

##   
## Augmented Dickey-Fuller Test  
##   
## data: ts(xtsAvgOccRefug)  
## Dickey-Fuller = -3.7185, Lag order = 9, p-value = 0.02307  
## alternative hypothesis: stationary

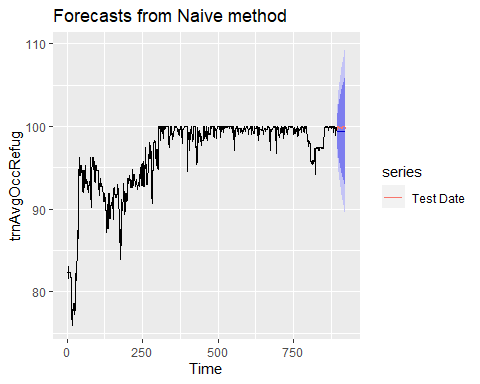
**Trend Assessment: Refugee Programs**

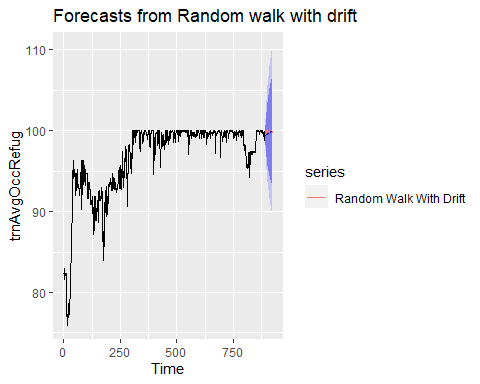
##   
## Mann-Kendall trend test  
##   
## data: ts(xtsAvgOccRefug)  
## z = 21.005, n = 924, p-value < 2.2e-16  
## alternative hypothesis: true S is not equal to 0  
## sample estimates:  
## S varS tau   
## 1.961350e+05 8.719257e+07 4.690059e-01

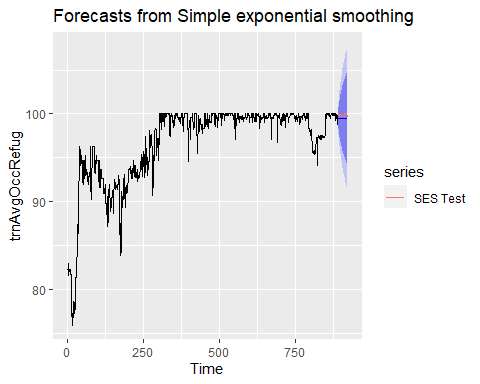
##   
## Sen's slope  
##   
## data: ts(xtsAvgOccRefug)  
## z = 21.005, n = 924, p-value < 2.2e-16  
## alternative hypothesis: true z is not equal to 0  
## 95 percent confidence interval:  
## 0.006572337 0.008403704  
## sample estimates:  
## Sen's slope   
## 0.007547428

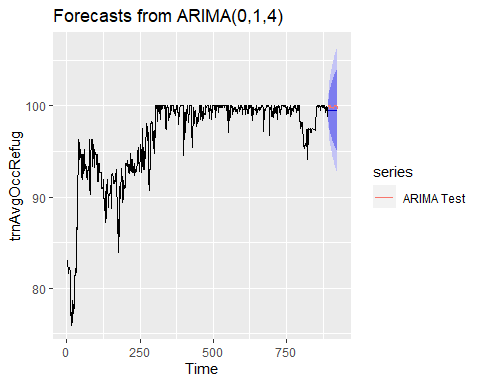
**Forecasting: Refugee Programs**

#Naive Test  
fcNaiAvgOccRefug<-naive(trnAvgOccRefug,h=30)  
autoplot(fcNaiAvgOccRefug)+  
 autolayer(tstAvgOccRefug,series='Test Date')

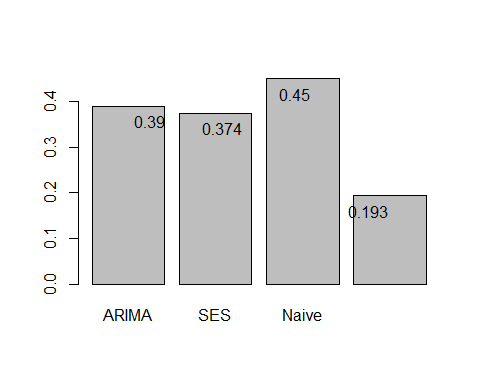


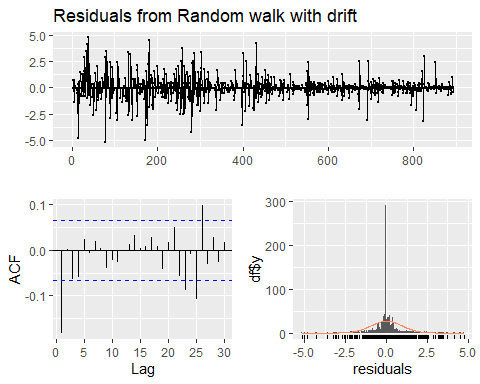






**Evaluating Model Performance: Refugee Capacity Rates**





##   
## Ljung-Box test  
##   
## data: Residuals from Random walk with drift  
## Q\* = 39.113, df = 10, p-value = 2.425e-05  
##   
## Model df: 0. Total lags used: 10