```
#Midterm-1

> install.packages("readr")
trying URL 'https://cran.rstudio.com/bin/macosx/contrib/4.1/readr_2.1.3.tgz'
Content type 'application/x-gzip' length 1856611 bytes (1.8 MB)
==================================================
downloaded 1.8 MB


The downloaded binary packages are in
	/var/folders/ln/6zltcwbj7cs9vmkqjl9z2llr0000gn/T//Rtmp6wPBwY/
downloaded_packages
> library(readr)
Warning message:
package ‘readr’ was built under R version 4.1.2
> library(fpp)
Loading required package: forecast
Registered S3 method overwritten by 'quantmod':
  method            from
  as.zoo.data.frame zoo
Loading required package: fma
Loading required package: expsmooth
Loading required package: lmtest
Loading required package: zoo

Attaching package: ‘zoo’

The following objects are masked from ‘package:base’:

    as.Date, as.Date.numeric

Loading required package: tseries

    ‘tseries’ version: 0.10-51

    ‘tseries’ is a package for time series analysis and computational finance.

    See ‘library(help="tseries")’ for details.

Warning messages:
1: package ‘forecast’ was built under R version 4.1.2
2: package ‘lmtest’ was built under R version 4.1.2
3: package ‘zoo’ was built under R version 4.1.2
4: package ‘tseries’ was built under R version 4.1.2
> library(fpp2)
── Attaching packages
──────────────────────────────────────────────────────────────
──────────────────────────────────────────────────────────────
────────────────────────────────────── fpp2 2.4 ──
✔ ggplot2 3.3.5
```

```
Attaching package: ‘fpp2’

The following objects are masked from ‘package:fpp’:

    ausair, ausbeer, austa, austourists, debitcards, departures, elecequip, euretail,
guinearice, oil,
    sunspotarea, usmelec

> NJ_MedianListingPrice_AllHomes <- read_csv("/Users/sarthakmehta/Desktop/
NJ_MedianListingPrice_AllHomes.csv")
Rows: 257 Columns: 2
── Column specification
────────────────────────────────────────────────────────────────────────
────────────────────────────────────────────────────────────────────────
────────────────────────────────────────────────────────
Delimiter: ","
chr (1): YYYY-MM
dbl (1): Value

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
> NJ_Home_Raw <- NJ_MedianListingPrice_AllHomes$Value
> NJ_Home_TS <- ts(NJ_Home_Raw,frequency = 12, start = c(1996,4))
>
> #Plot and Inference
> NJ_Home_TS
       Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
1996                      144200 143700 143200 143100 143200 143100 143000 142900
143000
1997 143100 143500 144000 144400 144600 144900 144900 144900 145000
145400 145800 146500
1998 147400 148500 149400 149900 150100 150300 150400 150400 150600
151100 151700 152300
1999 153200 154400 155400 156100 157000 158000 159100 160500 161900
162800 163900 165200
2000 166600 167800 169300 170600 172000 173600 175300 176700 178000
179500 181000 182700
2001 184400 186100 187900 189800 191800 193700 195600 197800 200300
202800 205300 207600
2002 209800 212300 215000 217900 220800 223800 227300 231000 234400
237600 240800 244000
2003 247200 250200 253100 256000 258900 262000 265200 268100 270900
273800 276700 279600
2004 282400 285100 287900 291100 294800 298600 302300 306100 310100
314200 318500 322700
2005 326300 330100 334000 337900 341800 345700 349300 352500 355500
358700 361700 364100
2006 366200 368100 369900 371600 372700 373100 373000 373100 373000
```

```
372300 371100 369800
2007 369000 368700 368300 367700 367200 366300 364700 362900 361200
359600 358200 356800
2008 355200 353100 350800 348300 345500 342300 339300 336500 333700
330900 327700 325100
2009 323800 322500 320200 317700 315200 312700 310600 309200 308100
306900 306100 305100
2010 303500 301900 300800 299600 298200 297200 296100 294500 292600
291100 290000 288600
2011 286800 285200 283500 281600 280100 278900 277500 275800 273700
271900 270300 268600
2012 267400 266700 266100 265700 265600 265800 266100 266700 267200
267500 267800 268700
2013 269300 269600 270200 270900 271500 272100 273100 274300 275700
276600 277200 277800
2014 278400 278700 279200 280200 281500 282400 282800 283200 283400
283300 283200 283400
2015 283900 284200 284300 285000 286000 287000 287800 288400 289000
289800 290500 291000
2016 291600 292100 293200 294700 295700 296100 296900 298000 299600
301500 303400 304700
2017 305500 306800 307900 307800 307700 308800 310400 311600
> plot(NJ_Home_TS) #plotting the TS plot
>               #Observation: from 1996 to ~2007-08 there was a sharp increase in the
in the median listing of price and then
>               #there was a sudden drop till the year 2012, and was in a constant
decline(thanks to the 2008 financial
>               #crisis). hence I would like to choose the data from the start of 2012 so
that my forecast is not tampered due
>               #to these fluctuations.
>
> CNJ_Home_TS <- window(NJ_Home_TS, start = 2012)
> plot(CNJ_Home_TS)
>
> #mCentral Tendency
>
> min(CNJ_Home_TS)
[1] 265600
> # [1] 265600
>
> max(CNJ_Home_TS)
[1] 311600
> # [1] 311600
>
> mean(CNJ_Home_TS)
[1] 284414.7
> # [1] 284414.7
>
> median(CNJ_Home_TS)
[1] 283350
```

```
> # [1] 283350
>
> quantile(CNJ_Home_TS)
    0%    25%    50%    75%   100%
265600 271950 283350 293575 311600
> #   25%    75%
> # 271950  293575
>
> boxplot((CNJ_Home_TS))
>
> #Summary: Well the summaries show us that the amongst the median values calculated min median value of a house from the
> #period of 2012 to 2017 was 265,600$ and the max median value was 311,600$. Furthermore, the mean value was 284,414.7$ and
> #the median value was 283,350$. The 1st quartile was 271,950$ and the 3rd quartile was 293,575$. The box plot tells us that
> #that the there are no outliers, and the minimum value is a bit more closer to the 1st quartile than the maximum value is to
> #the 3rd quatile,and the median is approximate in between the 1st and 3rd quartile. But when we look closely at the mean and
> #median we can see that as mean is a bit greater than median the median (2nd quartile line) would be positively skewed.
>
>
> #Decomposition
>
> stl_decomp <- stl(CNJ_Home_TS,s.window =12)
> stl_decomp
 Call:
 stl(x = CNJ_Home_TS, s.window = 12)

Components
          seasonal    trend    remainder
Jan 2012  324.62834 265597.1  1478.295828
Feb 2012  -47.30725 265828.2   919.145715
Mar 2012 -269.31656 266059.2   310.069331
Apr 2012 -279.87608 266290.3  -310.456850
May 2012 -249.32178 266551.4  -702.080809
Jun 2012 -211.69453 266812.5  -800.777729
Jul 2012 -105.77328 267073.5  -867.768649
Aug 2012   62.48700 267354.7  -717.139926
Sep 2012  157.89533 267635.8  -593.659256
Oct 2012  224.54246 267916.9  -641.417385
Nov 2012  177.61669 268384.1  -761.744469
Dec 2012  200.96663 268851.4  -352.347282
Jan 2013  303.70438 269318.6  -322.337897
Feb 2013  -45.86394 269982.2  -336.328334
Mar 2013 -247.72831 270645.8  -198.022732
Apr 2013 -258.93741 271309.3  -150.372403
May 2013 -242.57710 272060.9  -318.336404
```

Jun 2013	-210.14920	272812.5	-502.367990
Jul 2013	-99.48285	273564.1	-364.638032
Aug 2013	61.29595	274339.7	-101.001389
Sep 2013	131.76266	275115.3	452.947333
Oct 2013	210.94947	275890.9	498.175963
Nov 2013	186.19995	276681.6	332.208172
Dec 2013	207.32412	277472.3	120.366689
Jan 2014	280.32805	278263.0	-143.354551
Feb 2014	-46.93814	278948.0	-201.104902
Mar 2014	-228.72268	279633.1	-204.336906
Apr 2014	-240.89063	280318.1	122.814502
May 2014	-239.03357	280837.3	901.693404
Jun 2014	-212.42689	281356.6	1255.822669
Jul 2014	-97.63728	281875.9	1021.769020
Aug 2014	54.70219	282303.3	842.035275
Sep 2014	99.26945	282730.7	570.073740
Oct 2014	189.86957	283158.1	-47.920658
Nov 2014	186.16995	283560.8	-546.926888
Dec 2014	204.00585	283963.5	-767.468647
Jan 2015	256.25162	284366.2	-722.420276
Feb 2015	-33.37867	284842.2	-608.840524
Mar 2015	-185.48239	285318.3	-832.787355
Apr 2015	-208.99363	285794.3	-585.326652
May 2015	-231.49390	286395.5	-163.992699
Jun 2015	-205.32463	286996.7	208.671720
Jul 2015	-77.92606	287597.8	280.106838
Aug 2015	65.76487	288305.7	28.574752
Sep 2015	59.29997	289013.5	-72.801505
Oct 2015	159.97782	289721.3	-81.320509
Nov 2015	176.53974	290497.3	-173.887980
Dec 2015	191.60763	291273.4	-464.961421
Jan 2016	227.32856	292049.4	-676.687894
Feb 2016	-23.27121	292930.3	-807.072236
Mar 2016	-144.29946	293811.3	-467.028089
Apr 2016	-177.71485	294692.3	185.403203
May 2016	-223.13329	295762.7	160.396999
Jun 2016	-195.90001	296833.2	-537.260922
Jul 2016	-54.39104	297903.6	-949.194529
Aug 2016	82.21365	299054.7	-1136.878131
Sep 2016	26.27889	300205.7	-632.022282
Oct 2016	138.60759	301356.8	4.570106
Nov 2016	177.00419	302423.5	799.494005
Dec 2016	190.71416	303490.2	1019.104532
Jan 2017	202.59981	304556.9	740.539383
Feb 2017	-28.28100	305626.4	1201.892657
Mar 2017	-132.64957	306695.9	1336.733693
Apr 2017	-165.91856	307765.4	200.475156
May 2017	-221.27629	308818.4	-897.131128
Jun 2017	-196.41318	309871.4	-874.958245
Jul 2017	-50.51428	310924.3	-473.821148

```
Aug 2017   79.23076 311959.0  -438.229425
> plot(stl_decomp) #Decomposition Plot
> #Yes the time series is seasonal, and it is additive seasonal not multiplicative as
the magnitude is constant. For the month of
> #January, September, October and December the values for the seasonal
component are pretty high, i believe the reason for it is that
> #around the time of new-years people like to list their houses. And in the months of
March, April, May and June the values are
> #really negative, the reason could be Summer season being around the corner
and summer vacations.
>
> seasadj(stl_decomp)
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
Nov      Dec
2012 267075.4 266747.3 266369.3 265979.9 265849.3 266011.7 266205.8
266637.5 267042.1 267275.5 267622.4 268499.0
2013 268996.3 269645.9 270447.7 271158.9 271742.6 272310.1 273199.5
274238.7 275568.2 276389.1 277013.8 277592.7
2014 278119.7 278746.9 279428.7 280440.9 281739.0 282612.4 282897.6
283145.3 283300.7 283110.1 283013.8 283196.0
2015 283643.7 284233.4 284485.5 285209.0 286231.5 287205.3 287877.9
288334.2 288940.7 289640.0 290323.5 290808.4
2016 291372.7 292123.3 293344.3 294877.7 295923.1 296295.9 296954.4
297917.8 299573.7 301361.4 303223.0 304509.3
2017 305297.4 306828.3 308032.6 307965.9 307921.3 308996.4 310450.5
311520.8
>
> plot(CNJ_Home_TS)
> lines(seasadj(stl_decomp), col="Orange")
> #even after taking out the seasonal component the graph still looks the same this
means that all the change that is happening
> #in the time series is happening due to fundamental changes and are not affected
by seasonality.
>
> #Naive
>
> naive_forecast <- naive(CNJ_Home_TS,12) #give me the forecast for the next 12
months using Naive forecast method
> plot(naive_forecast)
>
> attributes(naive_forecast)
$names
 [1] "method"    "model"     "lambda"    "x"         "fitted"    "residuals" "series"    "mean"
"level"     "lower"
[11] "upper"

$class
[1] "forecast"

> #$names
```

```
> #[1] "method"    "model"     "lambda"    "x"        "fitted"    "residuals" "series"
"mean"      "level"     "lower"
> #[11] "upper"
>
> #$class
> #[1] "forecast"
>
> plot(naive_forecast$residuals) #the residual plot shows an anomaly around the
end of year 2016 and another sharp decline
>                         #just at the start of year 2017.
>
> hist(naive_forecast$residuals) #shows a pretty decent normal distribution
>
> Acf(naive_forecast$residuals) #shows all values are within thresholds apart from
the 1st and 2nd value they are dependent on
>                         #the previous values (correaltion present) apart from them no
values are dependent on each other.
>
> accuracy(naive_forecast)
              ME     RMSE      MAE       MPE      MAPE       MASE      ACF1
Training set 659.7015 848.3522 725.3731 0.2278946 0.2521256 0.08562583
0.6964852
> #             ME     RMSE      MAE       MPE      MAPE       MASE      ACF1
> #Training set  659.7015 848.3522 725.3731 0.2278946 0.2521256 0.08562583
0.6964852
>
> ets_forecast <- ets(CNJ_Home_TS)
> plot(CNJ_Home_TS)
>
> forecast_ets <- forecast(ets_forecast, h=12) #forecasting for the next 12 months
> plot(forecast_ets)
> forecast_ets
       Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
Sep 2017       312821.2 312231.6 313410.8 311919.4 313722.9
Oct 2017       314042.3 312753.0 315331.5 312070.5 316014.0
Nov 2017       315263.4 313125.9 317400.9 311994.3 318532.5
Dec 2017       316484.5 313370.3 319598.8 311721.7 321247.4
Jan 2018       317705.6 313500.2 321911.1 311273.9 324137.4
Feb 2018       318926.8 313525.7 324327.8 310666.6 327187.0
Mar 2018       320147.9 313454.6 326841.2 309911.4 330384.4
Apr 2018       321369.0 313293.0 329445.0 309017.9 333720.2
May 2018       322590.1 313045.9 332134.3 307993.6 337186.7
Jun 2018       323811.2 312717.6 334904.9 306844.9 340777.6
Jul 2018       325032.4 312311.4 337753.3 305577.4 344487.4
Aug 2018       326253.5 311830.6 340676.4 304195.6 348311.4
>
> #        Point Forecast  Lo 80    Hi 80    Lo 95    Hi 95
> #Sep 2017       312821.2 312231.6 313410.8 311919.4 313722.9
> #Oct 2017       314042.3 312753.0 315331.5 312070.5 316014.0
> #Nov 2017       315263.4 313125.9 317400.9 311994.3 318532.5
```

```
> #Dec 2017       316484.5 313370.3 319598.8 311721.7 321247.4
> #Jan 2018       317705.6 313500.2 321911.1 311273.9 324137.4
> #Feb 2018       318926.8 313525.7 324327.8 310666.6 327187.0
> #Mar 2018       320147.9 313454.6 326841.2 309911.4 330384.4
> #Apr 2018       321369.0 313293.0 329445.0 309017.9 333720.2
> #May 2018       322590.1 313045.9 332134.3 307993.6 337186.7
> #Jun 2018       323811.2 312717.6 334904.9 306844.9 340777.6
> #Jul 2018       325032.4 312311.4 337753.3 305577.4 344487.4
> #Aug 2018       326253.5 311830.6 340676.4 304195.6 348311.4
>
> ets_forecast$mse
[1] 169276.5
> #[1] 169276.5
> #RSME is 848.3522 on average forecast values were 848.3522 away from the
actual
> #MPE is 22% percenatge of error is 22%
> #MAPE is 25% for this model avg deviation from actual values is 25%
>
>
> #Simple Moving Average
>
> plot(CNJ_Home_TS)
> MA3_forecast <- ma(CNJ_Home_TS,order=3) #taking into account the 5 most
recent values
> MA6_forecast <- ma(CNJ_Home_TS,order=6) #taking into account the 9 most
recent values
> MA9_forecast <- ma(CNJ_Home_TS,order=9) #taking into account the 9 most
recent values
>
>
> rwf_forecast <- rwf(CNJ_Home_TS,12)
> snaive_forecast <- snaive(CNJ_Home_TS, 12)
>
> plot(CNJ_Home_TS)
> lines(rwf_forecast$mean,col="blue") #works best for my time series random walk
> lines(MA3_forecast,col="Red")
> lines(MA6_forecast,col="Blue")
> lines(MA9_forecast,col="Green")
>
> #as the order goes up the line comes up shorter and shorter from the end, and
from the start as well, the sample in getting
> #smaller
>
> #Simple Smoothing
>
> SSE_Simple <- HoltWinters(CNJ_Home_TS)
> attributes(SSE_Simple)
$names
[1] "fitted"       "x"            "alpha"        "beta"         "gamma"        "coefficients"
"seasonal"     "SSE"
```

```
[9] "call"

$class
[1] "HoltWinters"

> #$names
> #[1] "fitted"      "x"          "alpha"       "beta"        "gamma"       "coefficients"
"seasonal"     "SSE"
> #[9] "call"
>
> #$class
> #[1] "HoltWinters"
>
> plot(SSE_Simple)
> SSE_Simple$SSE
[1] 20477725
> #[1] 20477725
>
> #Holt Winters
>
> HW_forecast <- HoltWinters(CNJ_Home_TS)
> plot(HW_forecast)
> HW_forecast
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = CNJ_Home_TS)

Smoothing parameters:
 alpha: 1
 beta : 0.1539464
 gamma: 0

Coefficients:
             [,1]
a   311614.583333
b      991.011123
s1     193.750000
s2     106.250000
s3     -56.250000
s4     335.416667
s5     381.250000
s6      72.916667
s7       2.083333
s8     -31.250000
s9    -202.083333
s10   -372.916667
s11   -414.583333
s12    -14.583333
>
```

```
> HWForecast <- predict(HW_forecast, 12)
> plot(HWForecast)
> HWForecast
         Jan      Feb      Mar      Apr      May      Jun      Jul      Aug      Sep      Oct
Nov      Dec
2017                                                      312799.3 313702.9 314531.4
315914.0
2018 316950.9 317633.6 318553.7 319511.4 320331.6 321151.8 322101.1
323492.1
> HW_forecast #Alpha = 1 (the last value has maximum weight)
Holt-Winters exponential smoothing with trend and additive seasonal component.

Call:
HoltWinters(x = CNJ_Home_TS)

Smoothing parameters:
 alpha: 1
 beta : 0.1539464
 gamma: 0

Coefficients:
            [,1]
a   311614.583333
b      991.011123
s1     193.750000
s2     106.250000
s3     -56.250000
s4     335.416667
s5     381.250000
s6      72.916667
s7       2.083333
s8     -31.250000
s9    -202.083333
s10   -372.916667
s11   -414.583333
s12    -14.583333
>           #beta : 0.1539464 (this is the coefficient of trend smoothing in HW)
>           #gamma: 0 (non-seasonal model)
>
> ets_forecast #sigma:  0.0015 (standard dev. of residuals)
ETS(M,A,N)

Call:
 ets(y = CNJ_Home_TS)

  Smoothing parameters:
    alpha = 0.9999
    beta  = 0.9427

  Initial states:
```

```
    l = 267559.0133
    b = -1450.1725

  sigma:  0.0015

     AIC     AICc      BIC
1113.551 1114.518 1124.648
>             #Initial states:l = 267559.0133
>                         #b = -1450.1725
>
> #No residuals
>
>
> #accuracy summary
>
>
> accuracy(ets_forecast)
                 ME     RMSE      MAE        MPE      MAPE       MASE      ACF1
Training set 41.67252 411.4322 315.3271 0.01535473 0.1099933 0.03722243
0.1132951
> accuracy(naive_forecast)
                 ME     RMSE      MAE       MPE      MAPE       MASE      ACF1
Training set 659.7015 848.3522 725.3731 0.2278946 0.2521256 0.08562583
0.6964852
> accuracy(rwf_forecast)
                 ME     RMSE      MAE       MPE      MAPE       MASE      ACF1
Training set 659.7015 848.3522 725.3731 0.2278946 0.2521256 0.08562583
0.6964852
> accuracy(snaive_forecast)
                 ME     RMSE      MAE      MPE     MAPE MASE      ACF1
Training set 8471.429 9018.612 8471.429 2.911696 2.911696    1 0.9111869
>
> #I would choose ets_forecast as it has the lowest MAPE. I chose MAPE because
the units are in Percent, it is extremely useful
> #when observations are large numbers and also It can be used to compare same
or different techniques as units is in %.
>
>
> #Best MAPE Forecast
>
> accuracy(ets_forecast)
                 ME     RMSE      MAE        MPE      MAPE       MASE      ACF1
Training set 41.67252 411.4322 315.3271 0.01535473 0.1099933 0.03722243
0.1132951
>
> #worst MAPE Forecast
>
> accuracy(snaive_forecast)
                 ME     RMSE      MAE      MPE     MAPE MASE      ACF1
Training set 8471.429 9018.612 8471.429 2.911696 2.911696    1 0.9111869
```

```
>
> #conclusion
>
> #based on my analysis the value of time series will increase in 1 and 2 both years
```