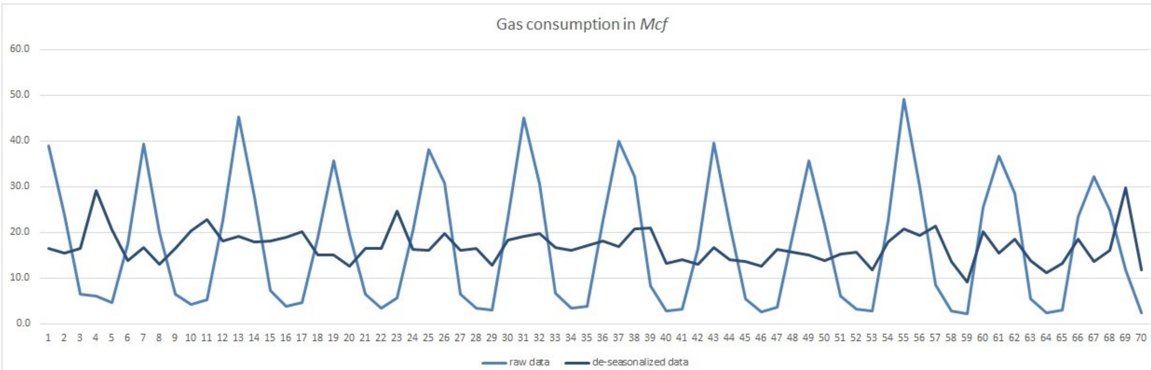
IE 2082 HW 2

Yichen Xue, Yingze Hou

September 2020

1. Plotting the data. Compute the seasonality indices.

The blue line is the original data. The dark blue line is the de-seasonalized data. The 1st label on x-axis is for the data in 2009 Jan. - Feb., and the second label is 2009 Mar. - Apr. The last 70th label is for 2020 July - Aug. The y-axis is D_t.



The seasonal period is 6 ($j = 6$). To compute the seasonality index (S_j^*), we need centered-moving-average (CMA_t):

$$CMA_t = (\frac{1}{2}D_{t-3} + D_{t-2} + D_{t-1} + D_t + D_{t+1} + D_{t+2} + \frac{1}{2}D_{t+3})\frac{1}{6}, \ \forall t = 4, \cdots, 67.$$

Then, we compute the un-adjusted seasonality indices (S_t) for $D_4 - D_{67}$:

$$S_t = D_t / CMA_t, \ \forall t = 4, \cdots, 67.$$

At last, we compute the seasonality (S_j^*) index by averaging all (S_t) in the same period and normalization:

$$S'_1 = (S_7 + S_{13} + \cdots + S_{67})/11,\ S'_2 = (S_2 + S_8 + \cdots + S_{62})/10,$$
$$S'_3 = (S_3 + S_9 + \cdots + S_{63})/10,\ \ S'_4 = (S_4 + S_{10} + \cdots + S_{64})/11,$$
$$S'_5 = (S_5 + S_{11} + \cdots + S_{65})/11,\ S'_6 = (S_6 + S_{12} + \cdots + S_{66})/11,$$
$$S^*_j = S'_j \cdot 6\ /\ \frac{1}{6}(S'_1 + S'_2 + S'_3 + S'_4 + S'_5 + S'_6),\ \forall j = 1, \cdots, 6.$$

The de-seasonalized data (D^*_t) is dividing D_t by its corresponding S^*_j:

$$D^*_t = D_t/S^*_j,\ \forall j = 1, \cdots, 6,\ \forall t = 1, \cdots, 70,\ t = j + 6k\ (k \in \mathbb{Z}).$$

The results are: $S^*_1 = 2.362$, $S^*_2 = 1.549$, $S^*_3 = 0.396$, $S^*_4 = 0.205$, $S^*_5 = 0.228$, $S^*_6 = 1.260$.

2. (1) Single exponential smoothing to update the mean level. (2) Locally constant linear trend model with double exponential smoothing. (3) Justify the choice of initialization and parameters. (4) Assess the model forecasting performance over the last 6 years (31th – 70th data).

(1) Single exponential smoothing

estimate in period t: $\hat{L}_t = (1-\alpha)\hat{L}^*_{t-1} + \alpha D^*_t,\ \hat{L}_{29} = D^*_{29},\ \forall t = 30, \cdots, 70,$

de-seasonalized forecast for period $t+1$: $F^*_{t,t+1} = \hat{L}_t,\ \forall t = 30, \cdots, 70,$

forecast for period $t+1$: $F_{t,t+1} = F^*_{t,t+1} \cdot S^*_j$, j is the corresponding period,

forecast error for period $t+1$: $e_{t+1} = D_{t+1} - F_{t,t+1}$.

Choose the initial estimate $\hat{L}_t = D^*_{29}$. Since our forecast starts at the 31st data ($F_{30,31}$), we need de-seasonalized estimate $\hat{L}_{30}$, which requires $\hat{L}_{29}$. It is reasonable to set the estimate $\hat{L}_{29} = D^*_{29}$.

(2) Double exponential smoothing

mean estimate in period t: $\hat{L}_t = (1-\alpha)\hat{L}_{t-1} + \alpha D_t^*, \ \hat{L}_{29} = 16.808, \ \forall t = 30, \cdots, 70,$

trend estimate in period t: $\hat{T}_t = (1-\beta)\hat{T}_{t-1} + \beta(\hat{L}_t - \hat{L}_{t-1}), \ \hat{T}_{29} = -0.064, \ \forall t = 30, \cdots, 70,$

de-seasonalized forecast for period $t+1$: $F_{t,t+1}^* = \hat{L}_t + \hat{T}_t, \ \forall t = 30, \cdots, 70,$

forecast for period $t+1$: $F_{t,t+1} = F_{t,t+1}^* \cdot S_j^*, \ j$ is the corresponding period,

forecast error for period $t+1$: $e_{t+1} = D_{t+1} - F_{t,t+1}$.

Choose the initial estimate $\hat{L}_t = 16.808$ and $\hat{T}_t = -0.064$.
Since our forecast starts at the 31st data ($F_{30,31}$), we need de-seasonalized estimate $\hat{L}_{30}$ and $\hat{T}_{30}$.
To calculate $\hat{L}_{30}$ and $\hat{T}_{30}$, we need to set values for $\hat{L}_{29}$ and $\hat{T}_{29}$.
To choose the value for $\hat{L}_{29}$ and $\hat{T}_{29}$, we did a linear regression on the first 29 de-seasonalized data points, giving us slope = -0.064 and intercept = 18.652. Thus, $\hat{T}_{29}$ = slope = -0.064. $\hat{L}_{29} = 29 \cdot \text{slope} + \text{intercept} = 29 \cdot (-0.064) + 18.652 = 16.808$.

(3) Forecasting performance assessment
We choose different values for α and β. To assess accuracy, we compute the mean error, mean absolute error, mean squared error, and mean absolute percentage error.

mean error: $ME = \sum_t e_t / n,$

mean absolute error: $MAE = \sum_t |e_t| / n,$

mean squared error: $MSE = \sum_t e_t^2 / n,$

mean absolute percentage error: $MAPE = 100 \cdot \sum_t |e_t / D_t| / n.$

An accurate model should have small values for all of these indices.
To assess bias, we compute the smoothed mean error, smoothed mean absolute deviation, and

tracking signal.

smoothed mean error: $E_t = \omega e_t + (1-\omega)E_{t-1},\ E_{37} = \sum_{t=31}^{36} E_t/6,\ t = 38, \cdots, 70,$

smoothed mean absolute deviation: $MAD_t = \omega|e_t| + (1-\omega)MAD_{t-1},\ MAD_{37} = \sum_{t=31}^{36} MAD_t/6,$

tracking signal: $TS_t = E_t/MAD_t$.

An unbiased model should have normally distributed error, and 95% of the time the errors should be within 2σ, where σ is the standard deviation of the error ($\sigma = MAD/\sqrt{\pi/2} \approx 1.25MAD$). We choose the average of MAD for this evaluation.

(4) Model comparison and selection

We compute the forecasts and the error based on the above process with different values of α and β. The R code is attached at the end of this file.

We find that all models have similar $ME, MAE, MSE, MAPE$ and E_t, MAD_t. But β is better to be 0.01, since larger values will cause an increase in these accuracy-measuring indices. Figure 1 shows an increasing β value will increase all these indices, expect ME.

Model	alpha	beta	ME	MAE	MSE	MAPE
Double Exp	0.2	0.01	1.1280	2.5643	14.3552	0.2096
Double Exp	0.2	0.04	1.1123	2.5899	14.6349	0.2126
Double Exp	0.2	0.07	1.1128	2.6208	14.9712	0.2157
Double Exp	0.2	0.1	1.1148	2.6515	15.2746	0.2189
Double Exp	0.2	0.13	1.1140	2.6786	15.5008	0.2220
Double Exp	0.2	0.16	1.1115	2.7003	15.6322	0.2249
Double Exp	0.2	0.2	1.1105	2.7226	15.6602	0.2284

Figure 1: Selected double exp. models with increasing β

Figure 2 gives the model with satisfying accuracy (choices with relatively small error indices after comparing multiple models).

Model	alpha	beta	ME	MAE	MSE	MAPE
Single Exp	0.1	---	1.0477	2.5070	13.9964	0.1975
Double Exp	0.05	0.01	1.1113	2.4975	14.1791	0.1898
Double Exp	0.1	0.01	1.0836	2.5003	14.4231	0.1974
Double Exp	0.15	0.01	1.1007	2.5257	14.4538	0.2041
Double Exp	0.2	0.01	1.1280	2.5643	14.3552	0.2096
Double Exp	0.25	0.01	1.1561	2.5900	14.2692	0.2135
Double Exp	0.3	0.01	1.1800	2.6092	14.2733	0.2165

Figure 2: Selected models with forecast error indices

Single exponential smoothing model ($\alpha = 0.1$), with lowest ME and MSE, seems to be preferred. Double exponential smoothing model ($\alpha = 0.05, \beta = 0.01$ and $\alpha = 0.15, \beta = 0.01$) seem to be the second preferred choice.

The next thing is checking unbiasedness. We plot the histogram for the three selected model.

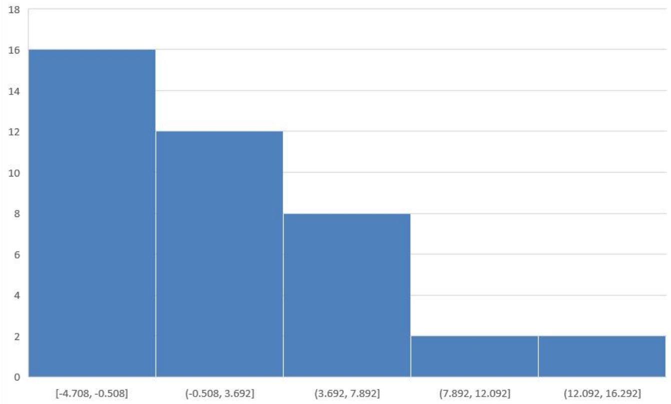


Figure 3: Histogram of e_t from single exp. model $\alpha = 0.1$

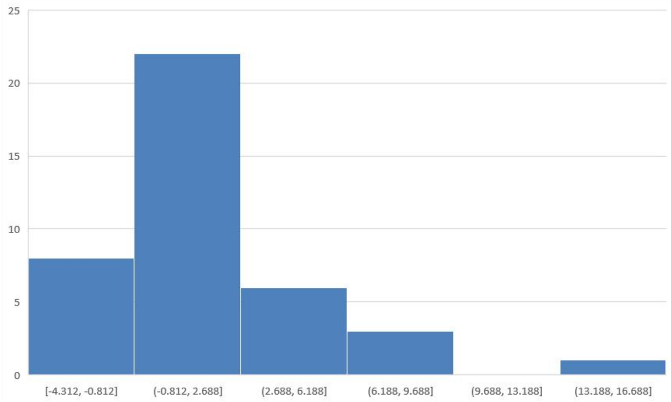


Figure 4: Histogram of e_t from double exp. model $\alpha = 0.05, \beta = 0.01$

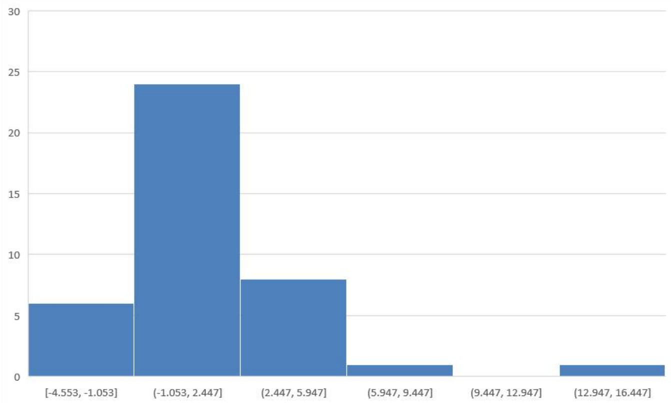


Figure 5: Histogram of e_t from double exp. model $\alpha = 0.15, \beta = 0.01$

Figure 2 shows an unbalanced distribution of e_t. So single exp. ($\alpha = 0.1$) tends to be biased in forecast, and should not be recommended. Figure 3 and 4 show a nearly normal distribution, with most e_t lie in $[-2\sigma, 2\sigma]$. But figure 4 is more preferred, since it is more balanced than firgure 3. Therefore, we recommend double exponential smoothing model ($\alpha = 0.15, \beta = 0.01$) for forecasting.

The forecast $F_{t,t+1}$ is provided at the end of this file.

3. Forecast D_t for the last two period in 2020 and all six period in 2021 (forecast for $t = 71, \cdots, 78$).

(1) Single exponential smoothing

$$\text{de-seasonalized forecast for period } t\text{: } F^*_{70,70+k} = \hat{L}_{70},\ k = 1, \cdots, 8,$$

$$\text{forecast for period } t\text{: } F_{70,70+k} = F^*_{70,70+k} \cdot S^*_j,\ j \text{ is the corresponding period.}$$

(2) Double exponential smoothing

$$\text{de-seasonalized forecast for period } t\text{: } F^*_{70,70+k} = \hat{L}_{70} + k \cdot \hat{T}_{70},\ k = 1, \cdots, 8,$$

$$\text{forecast for period } t\text{: } F_{70,70+k} = F^*_{70,70+k} \cdot S^*_j,\ j \text{ is the corresponding period.}$$

The choice of $\hat{L}_{70}$ and $\hat{T}_{70}$ are determined by the "best" model (with high accuracy and being unbiased) from part (2).

(3) Forecasting

We recommend double exponential smoothing model ($\alpha = 0.15, \beta = 0.01$) for forecasting. Therefore, $\hat{L}_{70} = 16.366$ and $\hat{T}_{70} = -0.044$. The forecasts are:

$$F_{70,71} = F^*_{70,71} \cdot S^*_5 = (\hat{L}_{70} + 1 \cdot \hat{T}_{70}) \cdot 0.228 = 3.717\ ,$$
$$F_{70,72} = F^*_{70,72} \cdot S^*_6 = (\hat{L}_{70} + 2 \cdot \hat{T}_{70}) \cdot 1.260 = 20.514\ ,$$
$$F_{70,73} = F^*_{70,73} \cdot S^*_1 = (\hat{L}_{70} + 3 \cdot \hat{T}_{70}) \cdot 2.362 = 38.348\ ,$$
$$F_{70,72} = F^*_{70,74} \cdot S^*_2 = (\hat{L}_{70} + 4 \cdot \hat{T}_{70}) \cdot 1.549 = 25.077\ ,$$
$$F_{70,72} = F^*_{70,75} \cdot S^*_3 = (\hat{L}_{70} + 5 \cdot \hat{T}_{70}) \cdot 0.396 = 6.394\ ,$$
$$F_{70,72} = F^*_{70,76} \cdot S^*_4 = (\hat{L}_{70} + 6 \cdot \hat{T}_{70}) \cdot 0.205 = 3.307\ ,$$
$$F_{70,72} = F^*_{70,77} \cdot S^*_5 = (\hat{L}_{70} + 7 \cdot \hat{T}_{70}) \cdot 0.228 = 3.657\ ,$$
$$F_{70,72} = F^*_{70,78} \cdot S^*_6 = (\hat{L}_{70} + 8 \cdot \hat{T}_{70}) \cdot 1.260 = 20.183.$$

Code for the approaches by R:

```r
# Read data
# The data is transferred from the origin 6*11 data into a one-list-table csv
    document
datadirectory <- "~/Desktop/supply chain analysis/hw2"
origin_data <- read.csv(file.path(datadirectory,"hw2 origin_data.csv"),header =
     FALSE)
names(origin_data) <- "D_t"
origin_data[,2] <- c(rep(c(1,2,3,4,5,6),11),1,2,3,4)
names(origin_data)[2] <- "Season"
origin_data[,3:6] <- 0
names(origin_data)[3:6] <- c("CMA_j","S*_j","S_j","D*_t")
# De-seasonalize
Sj <- c(0,0,0,0,0,0)
nSj <- c(0,0,0,0,0,0)
for(i in 4:67){
  origin_data[i,3] <- (origin_data[i-2,1]+origin_data[i-1,1]+origin_data[i,1]+
    origin_data[i+1,1]+origin_data[i+2,1]+0.5*origin_data[i-3,1]+0.5*origin_data
    [i+3,1])/6
  origin_data[i,4] <- origin_data[i,1]/origin_data[i,3]
  Sj[origin_data[i,2]] <- Sj[origin_data[i,2]] + origin_data[i,4]
  nSj[origin_data[i,2]] <- nSj[origin_data[i,2]] + 1
}
for(i in 1:6) {
  Sj[i] <- Sj[i] / nSj[i]
}
sumSj <- sum(Sj)
for(i in 1:6) {
  Sj[i] <- Sj[i] * 6 / sumSj
}
for(i in 1:70) {
  origin_data[i,5] <- Sj[origin_data[i,2]]
  origin_data[i,6] <- origin_data[i,1] / origin_data[i,5]
}
# Build the result table
# a_d <- c(0.05,0.1,0.15,0.2,0.25,0.3) for both single and double exponential
# b_d <- c(0.01,0.04,0.07,0.1,0.13,0.16,0.2)
TestResult <- data.frame(approach_type = c(rep("Single Exp",6),rep("Double Exp"
    ,42)))
```

```
TestResult[,2:17] <- 0
names(TestResult)[2:17] <- c("alpha value", "beta value",
                             "2020-S5", "2020-S6", "2021-S1", "2021-S2",
                             "2021-S3", "2021-S4", "2021-S5", "2021-S6",
                             "ME",  "MAE", "MSE", "MAPE",
                             "mean of E_t", "mean of MAD_t")
# Single Exponential Smoothing Approach
s_exp <- data.frame(D_t=origin_data[,1],S_j=origin_data[,5])
s_exp[71,2] <- Sj[5]
s_exp[72,2] <- Sj[6]
for(i in 1:6) {
  s_exp[72+i,2] <- Sj[i]
}
s_exp[1:70,3] <- origin_data[,6]
s_exp[,4:9] <- 0
names(s_exp)[3:9] <- c("D*_t","F*_t,t+1","F_t,t+1","e_t","|e|_t","E_t","MAD_t")
a_s <- c(0.1,0.15,0.2,0.25,0.3,0.35)
# Initialize: the F*_t,t+1 value for No.29 come from D*_29
s_exp[29,4] <- s_exp[29,3]
for(a in 1:length(a_s)) {
  for(i in 30:70){
    s_exp[i,4] <- a_s[a] * s_exp[i,3] + (1 - a_s[a]) * s_exp[i-1,4]
  }
  s_exp[71:77,4] <- s_exp[70,4]
  for(i in 30:77) {
    s_exp[i,5] <- s_exp[i,4] * s_exp[i+1,2]
  }
  for(i in 31:70) {
    s_exp[i,6] <- s_exp[i,1] - s_exp[i-1,5]
    s_exp[i,7] <- abs(s_exp[i,6])
  }
  s_exp[37,8] <- mean(s_exp[31:36,6])
  s_exp[37,9] <- mean(s_exp[31:36,7])
  for(i in 38:70) {
    s_exp[i,8] <- 0.1 * s_exp[i,6] + 0.9 * s_exp[i-1,8]
    s_exp[i,9] <- 0.1 * s_exp[i,7] + 0.9 * s_exp[i-1,9]
  }
  # Write in the result table
  for(i in 1:8) {
```

```
    TestResult[a,i+3] <- s_exp[i+69,5]
  }
  TestResult[a,2] <- a_s[a]
  TestResult[a,3] <- "---"
  TestResult[a,12] <- mean(s_exp[38:70,6])
  TestResult[a,13] <- mean(s_exp[38:70,7])
  TestResult[a,14] <- mean(s_exp[38:70,7]^2)
  TestResult[a,15] <- mean(s_exp[38:70,7]/s_exp[38:70,1])
  TestResult[a,16] <- mean(s_exp[38:70,8])
  TestResult[a,17] <- mean(s_exp[38:70,9])
}
#Double Exponential Smoothing Approach
d_exp <- data.frame(D_t=origin_data[,1],S_j=origin_data[,5])
d_exp[71,2] <- Sj[5]
d_exp[72,2] <- Sj[6]
for(i in 1:6) {
  d_exp[72+i,2] <- Sj[i]
}
d_exp[1:70,3] <- origin_data[,6]
d_exp[,4:11] <- 0
names(d_exp)[3:11] <- c("D*_t","F*_t,t+1","L_t","T_t","F_t,t+1","e_t","|e|_t","E_t","MAD_t")
# Initialize: the L_t and T_t value for No.29 come from regression test
d_exp[29,6] <- -0.06359
d_exp[29,5] <- d_exp[29,6] * 29 + 18.652
d_exp[29,4] <- d_exp[29,5] + d_exp[29,6]
a_d <- c(0.05,0.1,0.15,0.2,0.25,0.3)
b_d <- c(0.01,0.04,0.07,0.1,0.13,0.16,0.2)
for(a in 1:length(a_d)) {
  for(b in 1:length(b_d)) {
    for(i in 30:70) {
      d_exp[i,5] <- a_d[a] * d_exp[i,3] + (1 - a_d[a]) * d_exp[i-1,4]
      d_exp[i,6] <- b_d[b] * (d_exp[i,5] - d_exp[i-1,5]) + (1 - b_d[b]) * d_exp[i-1,6]
      d_exp[i,4] <- d_exp[i,5] + d_exp[i,6]
    }
    for(i in 2:8) {
      d_exp[i+69,4] <- d_exp[70,5] + i * d_exp[70,6]
    }
```

```
    for(i in 30:77) {
      d_exp[i,7] <- d_exp[i,4] * d_exp[i+1,2]
    }
    for(i in 31:70) {
      d_exp[i,8] <- d_exp[i,1] - d_exp[i-1,7]
      d_exp[i,9] <- abs(d_exp[i,8])
    }
    d_exp[37,10] <- mean(d_exp[31:36,8])
    d_exp[37,11] <- mean(d_exp[31:36,9])
    for(i in 38:70) {
      d_exp[i,10] <- 0.1 * d_exp[i,8] + 0.9 * d_exp[i-1,10]
      d_exp[i,11] <- 0.1 * d_exp[i,9] + 0.9 * d_exp[i-1,11]
    }
    # Write in the result table
    # The position in the result table should be 6+7*(a-1)+b
    for(i in 1:8) {
      TestResult[6+7*(a-1)+b,i+3] <- d_exp[i+69,7]
    }
    TestResult[6+7*(a-1)+b,2] <- a_d[a]
    TestResult[6+7*(a-1)+b,3] <- b_d[b]
    TestResult[6+7*(a-1)+b,12] <- mean(d_exp[38:70,8])
    TestResult[6+7*(a-1)+b,13] <- mean(d_exp[38:70,9])
    TestResult[6+7*(a-1)+b,14] <- mean(d_exp[38:70,9]^2)
    TestResult[6+7*(a-1)+b,15] <- mean(d_exp[38:70,9]/d_exp[38:70,1])
    TestResult[6+7*(a-1)+b,16] <- mean(d_exp[38:70,10])
    TestResult[6+7*(a-1)+b,17] <- mean(d_exp[38:70,11])
  }
}
TestResult
# Write the result into csv document
write.csv(TestResult,file.path(datadirectory,"hw2 test result.csv"))
```

Double exponential smoothing approach $\alpha = 0.15, \beta = 0.01$											
	D_t	D_t^*	$F_{t,t+1}^*$	$\hat{L}_t$	$\hat{T}_t$	S_j	$F_{t,t+1}$	e_t	$\lvert e_t \rvert$	E_t	MAD_t
29	2.900	12.734	16.744	16.808	-0.064						
30	23.200	18.411	16.933	16.994	-0.061		39.996				
31	45.100	19.094	17.199	17.257	-0.058	2.362	26.638	5.104	5.104		
32	30.500	19.693	17.519	17.573	-0.054	1.549	6.937	3.862	3.862		
33	6.600	16.667	17.336	17.392	-0.055	0.396	3.560	-0.337	0.337		
34	3.300	16.068	17.089	17.146	-0.057	0.205	3.892	-0.260	0.260		
35	3.900	17.125	17.037	17.094	-0.057	0.228	21.469	0.008	0.008		
36	22.700	18.014	17.128	17.184	-0.056	1.260	40.456	1.231	1.231		
37	40.000	16.935	17.043	17.099	-0.056	2.362	26.395	-0.456	0.456	1.601	1.801
38	32.300	20.855	17.564	17.615	-0.050	1.549	6.955	5.905	5.905	2.032	2.211
39	8.300	20.961	18.028	18.074	-0.045	0.396	3.703	1.345	1.345	1.963	2.124
40	2.700	13.146	17.244	17.296	-0.053	0.205	3.927	-1.003	1.003	1.666	2.012
41	3.200	14.052	16.707	16.765	-0.057	0.228	21.053	-0.727	0.727	1.427	1.884
42	16.500	13.094	16.103	16.165	-0.063	1.260	38.035	-4.553	4.553	0.829	2.151
43	39.600	16.765	16.140	16.202	-0.062	2.362	24.998	1.565	1.565	0.903	2.092
44	21.700	14.011	15.756	15.821	-0.065	1.549	6.239	-3.298	3.298	0.483	2.213
45	5.400	13.637	15.370	15.438	-0.068	0.396	3.157	-0.839	0.839	0.350	2.075
46	2.600	12.660	14.891	14.963	-0.072	0.205	3.391	-0.557	0.557	0.260	1.923
47	3.700	16.247	15.024	15.095	-0.070	0.228	18.933	0.309	0.309	0.265	1.762
48	19.700	15.633	15.046	15.116	-0.069	1.260	35.540	0.767	0.767	0.315	1.663
49	35.600	15.072	14.981	15.05	-0.069	2.362	23.202	0.060	0.060	0.289	1.502
50	21.500	13.882	14.745	14.816	-0.071	1.549	5.839	-1.702	1.702	0.090	1.522
51	6.000	15.152	14.736	14.806	-0.070	0.396	3.026	0.161	0.161	0.097	1.386
52	3.200	15.581	14.794	14.863	-0.069	0.205	3.369	0.174	0.174	0.105	1.265
53	2.700	11.856	14.280	14.353	-0.073	0.228	17.994	-0.669	0.669	0.028	1.205
54	22.500	17.855	14.748	14.816	-0.068	1.260	34.835	4.506	4.506	0.475	1.535
55	49.100	20.787	15.595	15.654	-0.059	2.362	24.153	14.265	14.265	1.854	2.808
56	29.900	19.306	16.098	16.152	-0.053	1.549	6.375	5.747	5.747	2.244	3.102
57	8.500	21.466	16.858	16.903	-0.045	0.396	3.462	2.125	2.125	2.232	3.005
58	2.800	13.633	16.324	16.374	-0.050	0.205	3.717	-0.662	0.662	1.942	2.770
59	2.100	9.221	15.198	15.259	-0.061	0.228	19.151	-1.617	1.617	1.586	2.655
60	25.400	20.157	15.888	15.942	-0.053	1.260	37.528	6.249	6.249	2.053	3.014
61	36.700	15.538	15.782	15.836	-0.054	2.362	24.442	-0.828	0.828	1.765	2.796
62	28.600	18.466	16.134	16.184	-0.050	1.549	6.389	4.158	4.158	2.004	2.932
63	5.500	13.890	15.744	15.798	-0.053	0.396	3.234	-0.889	0.889	1.715	2.728
64	2.300	11.199	15.002	15.063	-0.060	0.205	3.417	-0.934	0.934	1.450	2.548
65	3.000	13.173	14.665	14.728	-0.063	0.228	18.480	-0.417	0.417	1.263	2.335
66	23.400	18.570	15.194	15.251	-0.057	1.260	35.888	4.920	4.920	1.629	2.594
67	32.200	13.632	14.900	14.96	-0.059	2.362	23.077	-3.688	3.688	1.097	2.703
68	24.800	16.013	15.010	15.067	-0.058	1.549	5.943	1.723	1.723	1.160	2.605
69	11.800	29.799	17.193	17.228	-0.035	0.396	3.531	5.857	5.857	1.629	2.930
70	2.400	11.686	16.323	16.366	-0.044	0.205	3.717	-1.131	1.131	1.353	2.750
71			16.279			0.228	20.514				
72			16.235			1.260	38.348				
73			16.192			2.362	25.077				
74			16.148			1.549	6.394				
75			16.104			0.396	3.307				
76			16.060			0.205	3.657				
77			16.017			0.228	20.183				
78						1.260					

Mean of E_t	1.168
Mean of MAD_t	2.267