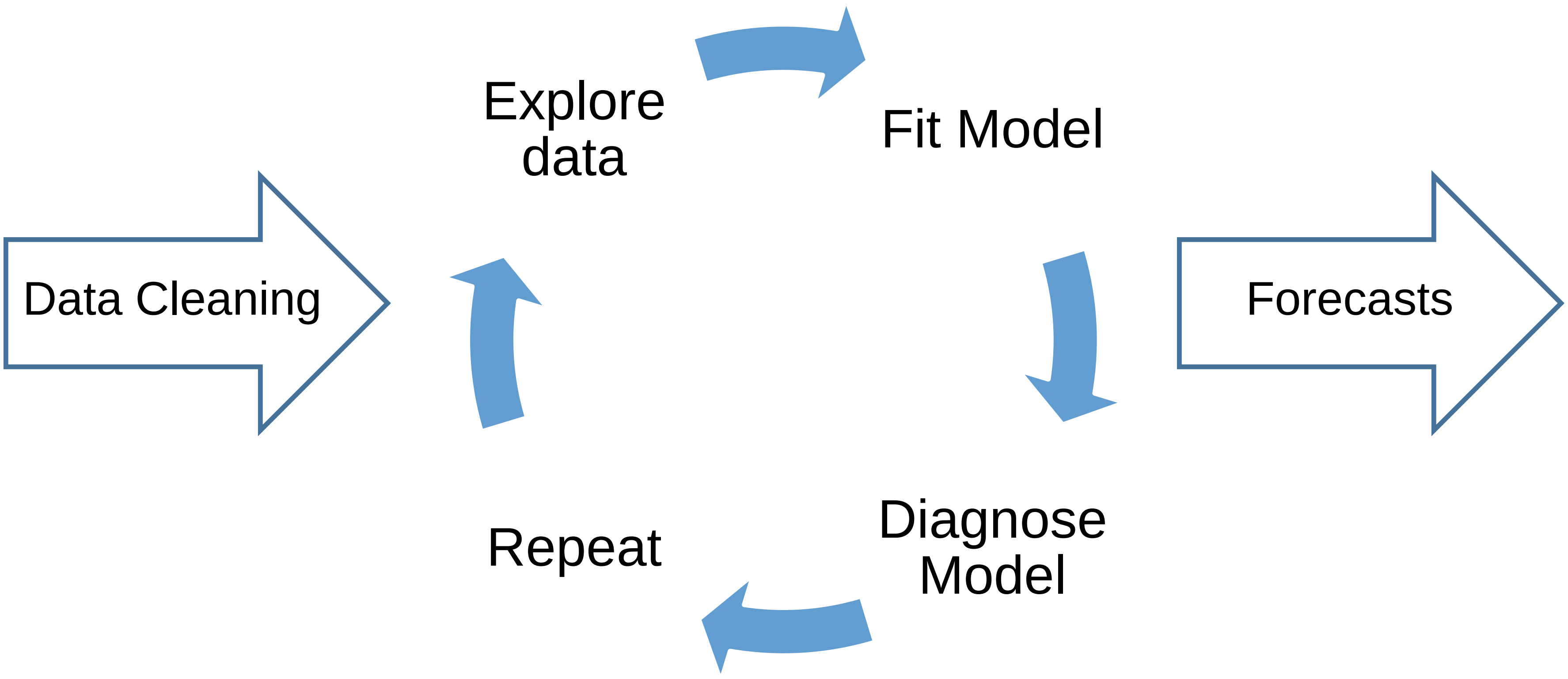
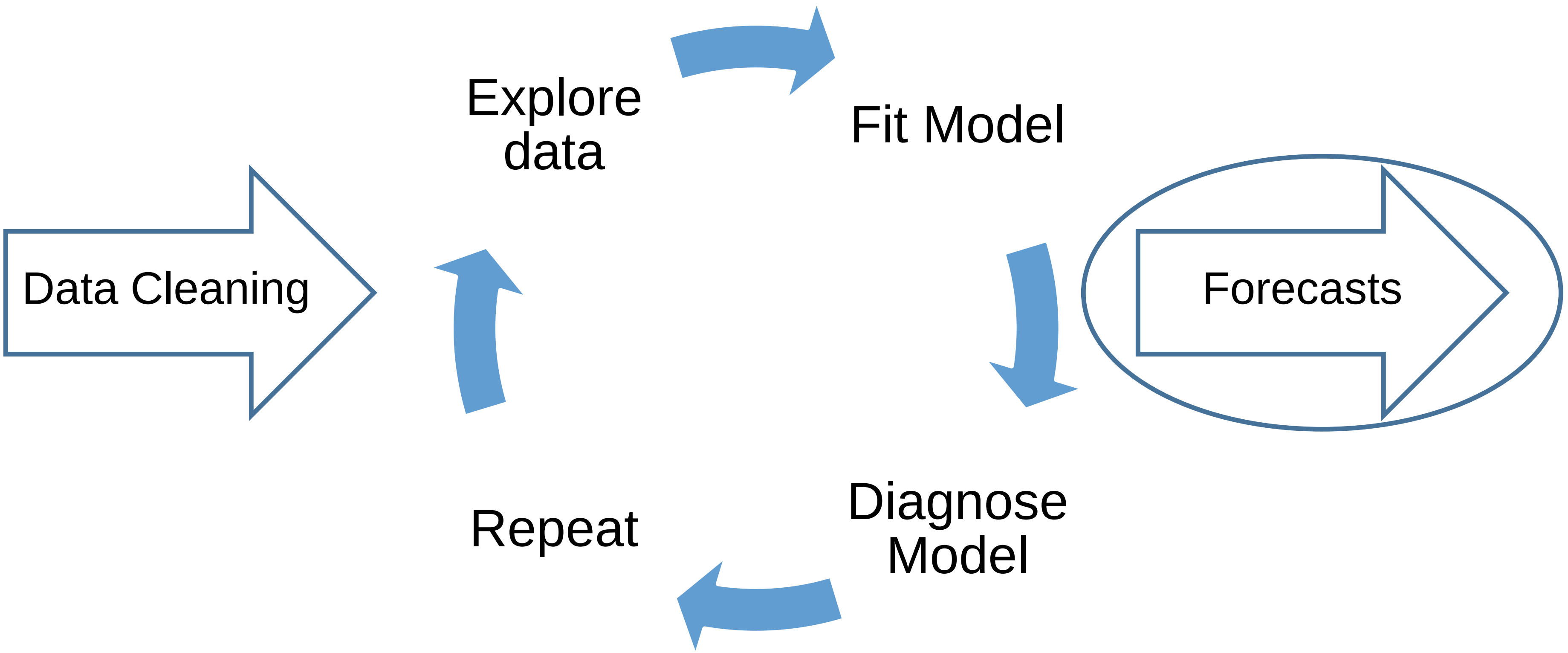
Source: xkcd.com/2586

Modeling/Forecasting Process



Modeling/Forecasting Process



EVALUATING FORECASTS

Forecasting Strategy

- Accuracy of forecasts depends on your definition of accuracy.
  - Different across different fields of industry.

- Good forecasts should have the following characteristics:
  - Be highly correlated with actual series values
  - Exhibit small forecast errors
  - Capture the important features of the original time series.

Judgment Forecasting

- When using data, forecasts are found using quantitative (or modeling) approaches. However, there are instances where models are not available (or potentially past data is not available) and a qualitative or judgement forecast is used.
- Occasionally a qualitative and quantitative approach are merged together.

Accuracy vs. Goodness-of-Fit

- A diagnostic statistic calculated using the same sample that was used to build the model is a ***goodness-of-fit*** statistic (example: AIC, BIC, but can also use MAPE, MAE, etc…).
- A diagnostic statistic calculated using a hold out sample that was not used in the building of the model is an ***accuracy*** statistic.

Hold-out Sample

- A hold out sample in time series analysis is different than cross-sectional analysis.
- The hold-out sample is always at the end of the time series, and doesn’t typically go beyond 25% of the data.
- IF YOU HAVE A SEASONAL TIME SERIES - Ideally, an entire season should be captured in a hold-out sample.

Hold-out Sample

1. Divide the time series into two or three segments – training and validation and/or test.
2. Derive a set of candidate models.
3. Calculate the chosen *accuracy* statistic by forecasting the validation data set.
4. Pick the model with the best accuracy statistic.
5. Provide the accuracy of the model on the *test* data set (recommend combining train and validation to get the most updated parameters…at this point, you will NOT change the model!).

Model Diagnostic Statistics

1. Mean Absolute Percent Error:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right|$$

2. Mean Absolute Error:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|Y_t - \hat{Y}_t\right|$$

Model Diagnostic Statistics

1. Mean Absolute Percent Error:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right| \longrightarrow$$

Problems:
- Overweight of Over-predictions
- Actual of 0

2. Mean Absolute Error:

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}\left|Y_t - \hat{Y}_t\right| \longrightarrow$$

Problems:
- Not scale invariant

Model Diagnostic Statistics

3. Square Root of Mean Square Error:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2}$$

4. Symmetric Mean Absolute Percent Error:

$$\mathrm{sMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|Y_t - \hat{Y}_t\right|}{\left(|Y_t| + |\hat{Y}_t|\right)/2}$$

Model Diagnostic Statistics

3. Square Root of Mean Square Error:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2} \longrightarrow$$

Problems:
- Overweight of larger errors
- Not scale invariant

4. Symmetric Mean Absolute Percent Error:

$$\text{sMAPE} = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|Y_t - \hat{Y}_t\right|}{\left(\left|Y_t\right| + \left|\hat{Y}_t\right|\right)/2} \longrightarrow$$

Problems:
- Divide by 0
- Still asymmetric

Comparison Across Diagnostics

	$Y_t = 1$, $\widehat{Y}_t = 3$	$Y_t = 2$, $\widehat{Y}_t = 3$	$Y_t = 3$, $\widehat{Y}_t = 3$	$Y_t = 4$, $\widehat{Y}_t = 3$	$Y_t = 15$, $\widehat{Y}_t = 3$	MEAN
APE	200%	50%	0%	25%	80%	71%
AE	2	1	0	1	12	3.2
SE	4	1	0	1	144	30
Sym. APE	100%	40%	0%	28.6%	133.3%	60.4%

Comparison Across Diagnostics

	$Y_t = 0$, $\hat{Y}_t = 3$	$Y_t = 2$, $\hat{Y}_t = 3$	$Y_t = 3$, $\hat{Y}_t = 3$	$Y_t = 4$, $\hat{Y}_t = 3$	$Y_t = 15$, $\hat{Y}_t = 3$	MEAN
APE	∞	50%	0%	25%	80%	…?
AE	3	1	0	1	12	3.4
SE	9	1	0	1	144	31
Sym. APE	200%	40%	0%	28.6%	133.3%	80.4%

Model Diagnostic Statistics

5. Akaike's Information Criterion

$$\mathrm{AIC} = -2\log(L) + 2k$$

$$\mathrm{AIC} = n\log\left(\frac{SSE}{n}\right) + 2k$$

Likelihood Based

6. Schwarz's Bayesian Information Criterion:

$$\mathrm{SBC} = -2\log(L) + k\log(n)$$

$$\mathrm{SBC} = n\log\left(\frac{SSE}{n}\right) + k\log(n)$$

Model Diagnostic Statistics

5. Akaike's Information Criterion

$$\text{AIC} = -2\log(L) + 2k$$

$$\text{AIC} = n\log\left(\frac{SSE}{n}\right) + 2k$$

Error Based

6. Schwarz's Bayesian Information Criterion:

$$\text{SBC} = -2\log(L) + k\log(n)$$

$$\text{SBC} = n\log\left(\frac{SSE}{n}\right) + k\log(n)$$

EXPONENTIAL SMOOTHING MODELS (ESM)

Time Dependencies

- Time series data relies on the assumption that the observations at a certain time point depend on previous observations in time.

Time Dependencies

- Time series data relies on the assumption that the observations at a certain time point depend on previous observations in time.

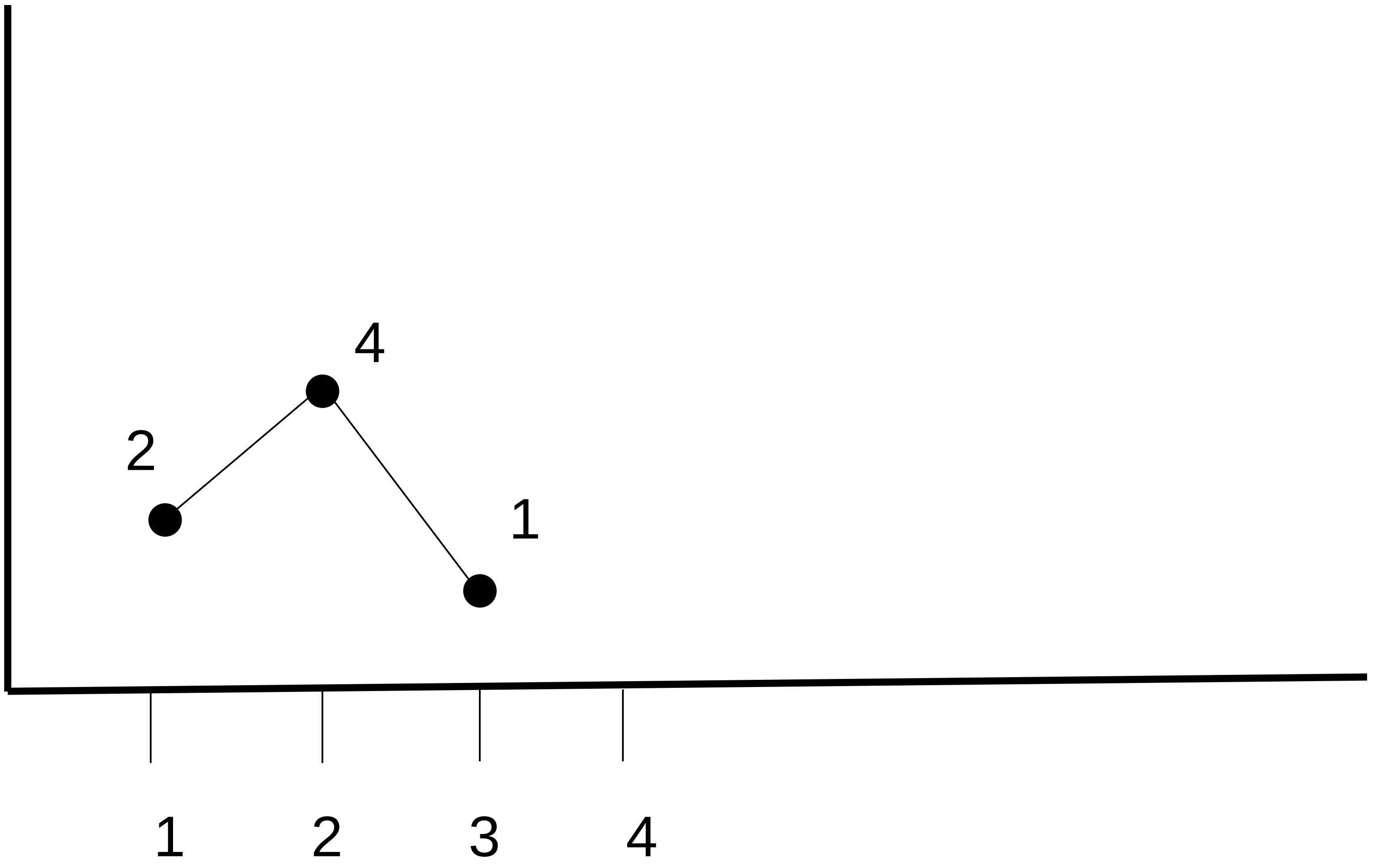
Naïve Model:

$$\hat{Y}_{t+h} = Y_t$$

Average Model:

$$\hat{Y}_{t+h} = \frac{1}{T}\sum_{t=1}^{T} Y_t$$

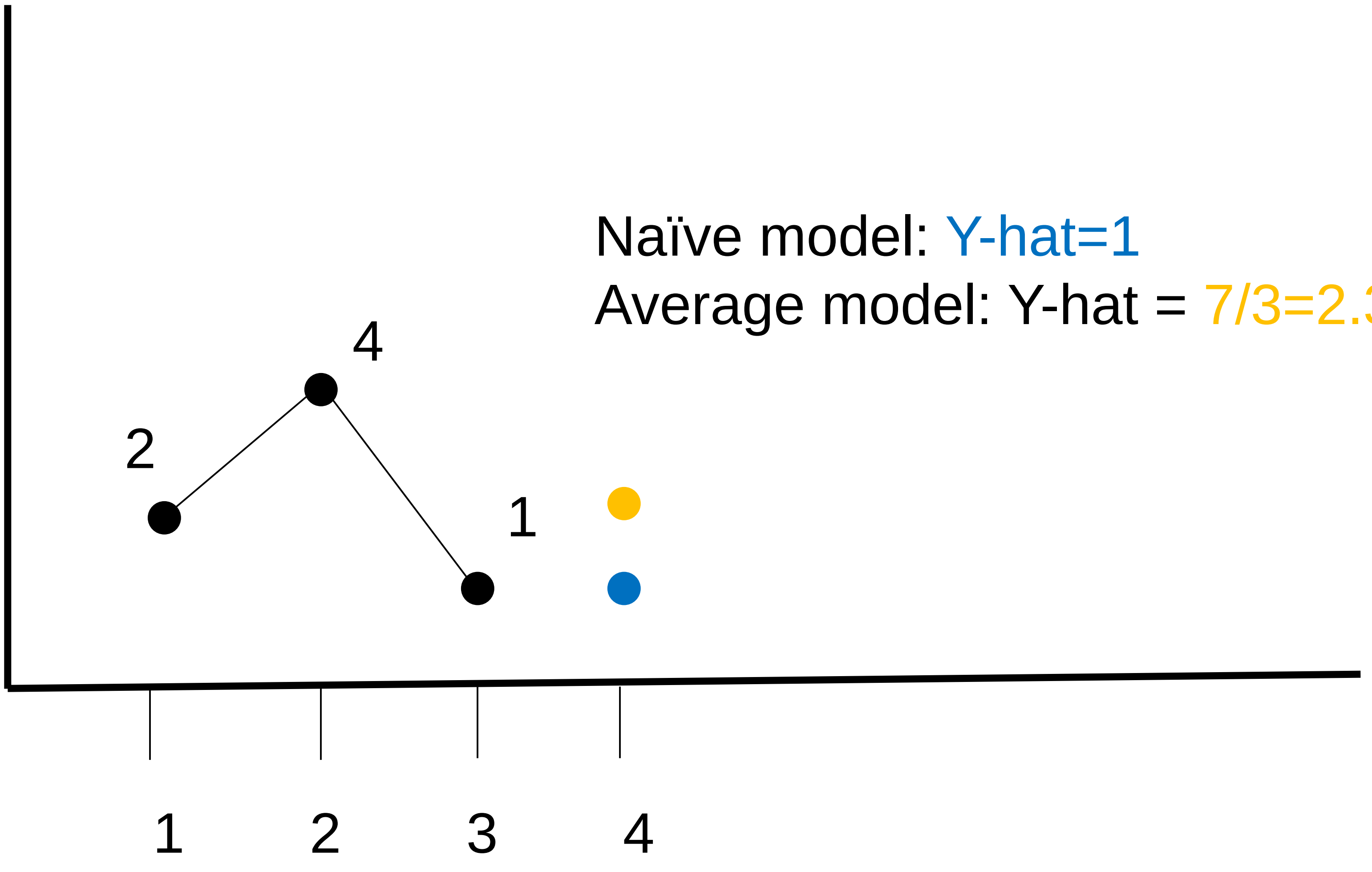
Naïve model versus Average model



Naïve model versus Average model

Naïve model: Y-hat=1

Average model: Y-hat = 7/3=2.333



Exponential Smoothing

- This is what exponential smoothing does (however, it is a WEIGHTED average, not a simple average)
- Models only require a few parameters.
- Equations are simple and easy to implement.

Exponential Smoothing

- There are many different types of exponential smoothing models.
- We will discuss traditional Exponential Smoothing Models:
  - Single (or Simple)
  - Linear / Holt (incorporates trend)
  - Holt-Winters (incorporates trend and seasonality)
- And newer algorithms (ETS)
- These models are great for “one-step ahead” forecasting

SINGLE EXPONENTIAL SMOOTHING

Single Exponential Smoothing

- The Single Exponential Smoothing model equates the predictions at time *t* equal to the weighted values of the previous time period along with the previous time period's prediction:

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)\hat{Y}_t$$

Where $\hat{Y}_t$ is the estimate of Y_t (weighted average of previous observations)

Single Exponential Smoothing

- We can apply a weighting scheme that decreases exponentially the further back in time we go.

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)\hat{Y}_t$$

Single Exponential Smoothing

- We can apply a weighting scheme that decreases exponentially the further back in time we go.

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)\hat{Y}_t$$

Single Exponential Smoothing

- We can apply a weighting scheme that decreases exponentially the further back in time we go.

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)\hat{Y}_t$$

$$\hat{Y}_{t+1} = \alpha Y_t + (1 - \alpha)[\alpha Y_{t-1} + (1 - \alpha)\hat{Y}_{t-1}]$$

Single Exponential Smoothing

- We can apply a weighting scheme that decreases exponentially the further back in time we go.

$$\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)\hat{Y}_t$$

$$\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)[\alpha Y_{t-1} + (1-\alpha)\hat{Y}_{t-1}]$$

Single Exponential Smoothing

- We can apply a weighting scheme that decreases exponentially the further back in time we go.

$$\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)\hat{Y}_t$$

$$\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)[\alpha Y_{t-1} + (1-\alpha)\hat{Y}_{t-1}]$$

$$\hat{Y}_{t+1} = \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + (1-\alpha)^2\hat{Y}_{t-1}$$

$$\hat{Y}_{t+1} = \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2}$$
$$+(1-\alpha)^3\hat{Y}_{t-2}$$

$$\vdots$$

$$\hat{Y}_{t+1} = \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} + \cdots$$

Single Exponential Smoothing

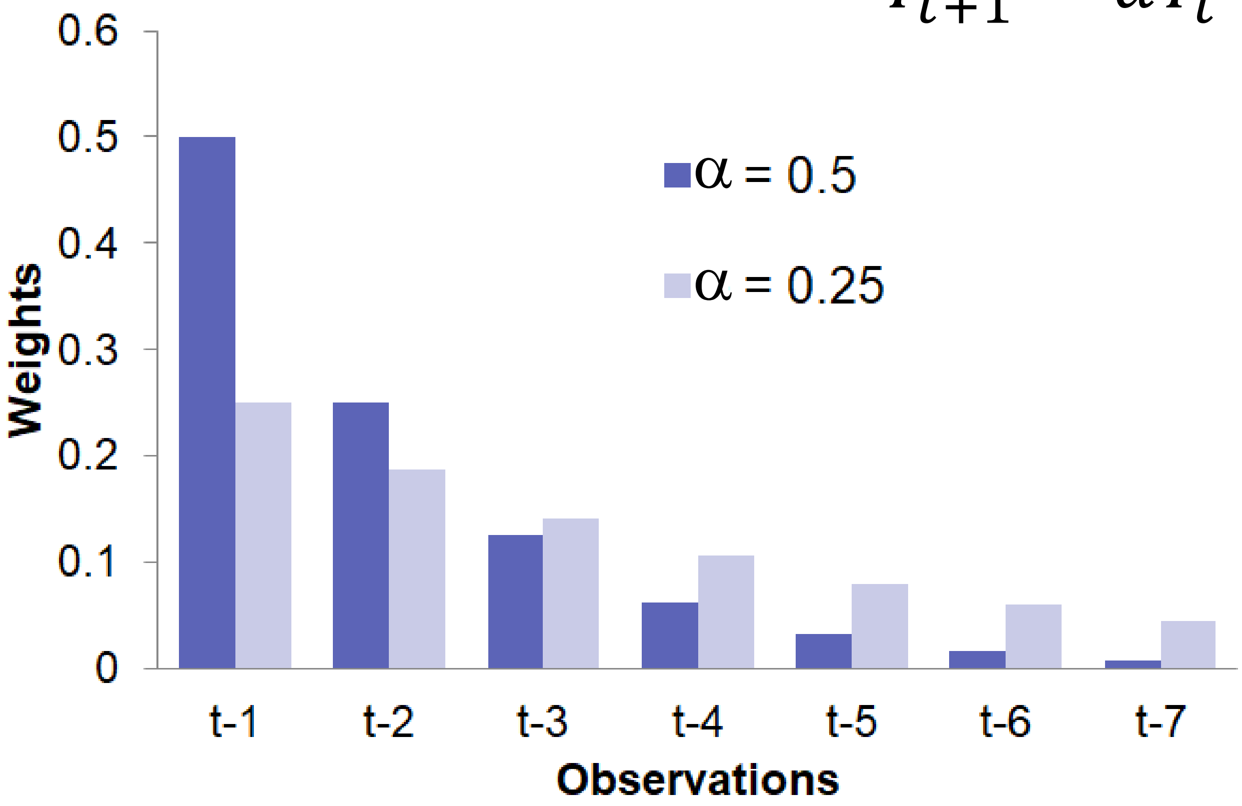
- As you can see, as we go further back in time, the weights decrease exponentially (more weight is put on the most recent observations).

$$\hat{Y}_{t+1} = \alpha Y_t + \alpha(1-\alpha)Y_{t-1} + \alpha(1-\alpha)^2 Y_{t-2} + \alpha(1-\alpha)^3 Y_{t-3} + \alpha(1-\alpha)^4 Y_{t-4} + \cdots$$

$$0 \leq \alpha \leq 1$$

Single Exponential Smoothing

$$\hat{Y}_{t+1} = \alpha Y_t + (1-\alpha)\hat{Y}_t$$



Component Form

- ESM can be written in component form…think back to time series decomposition
  - Seasonal
  - Trend
  - Error
- We will need to specify each of these when it comes to writing the code (Hyndman’s way of doing it)
- For the Single Exponential Smoothing Model (SES), we have NO season, NO trend and we will start with an additive error (error is allowed to be additive or multiplicative….most older models assume additive error)

Component Form

- The Single ESM written in **component form**:

Forecast Equation: $\hat{Y}_{t+1} = L_t$

Level Equation: $L_t = \alpha Y_t + (1 - \alpha) L_{t-1}$

There is NO seasonal component and NO trend component (we do NOT write out the error since this is the "Signal" part of the equation…..NOT stochastic)

Parameter Estimation

$$\hat{Y}_t = \alpha Y_{t-1} + (1-\alpha)\hat{Y}_{t-1}$$

- The typical method for calculating the optimal value of α in the Exponential Smoothing model is through one-step ahead forecasts.
- The value of α that minimizes the one-step ahead forecast errors is considered the optimal value.

$$SSE = \sum_{t=1}^{T} \left(Y_t - \hat{Y}_t\right)^2$$

Parameter Estimation

$$\hat{Y}_t = \alpha Y_{t-1} + (1-\alpha)\hat{Y}_{t-1}$$

- Estimates that are not statistically significant should not be disqualified (in fact, significance test usually test if α = 0….which simplifies down to the average model).
- Models were originally derived without statistical distribution consideration (estimates are fine even without normality!).
- HOWEVER, normality is needed if trying to construct a confidence interval.

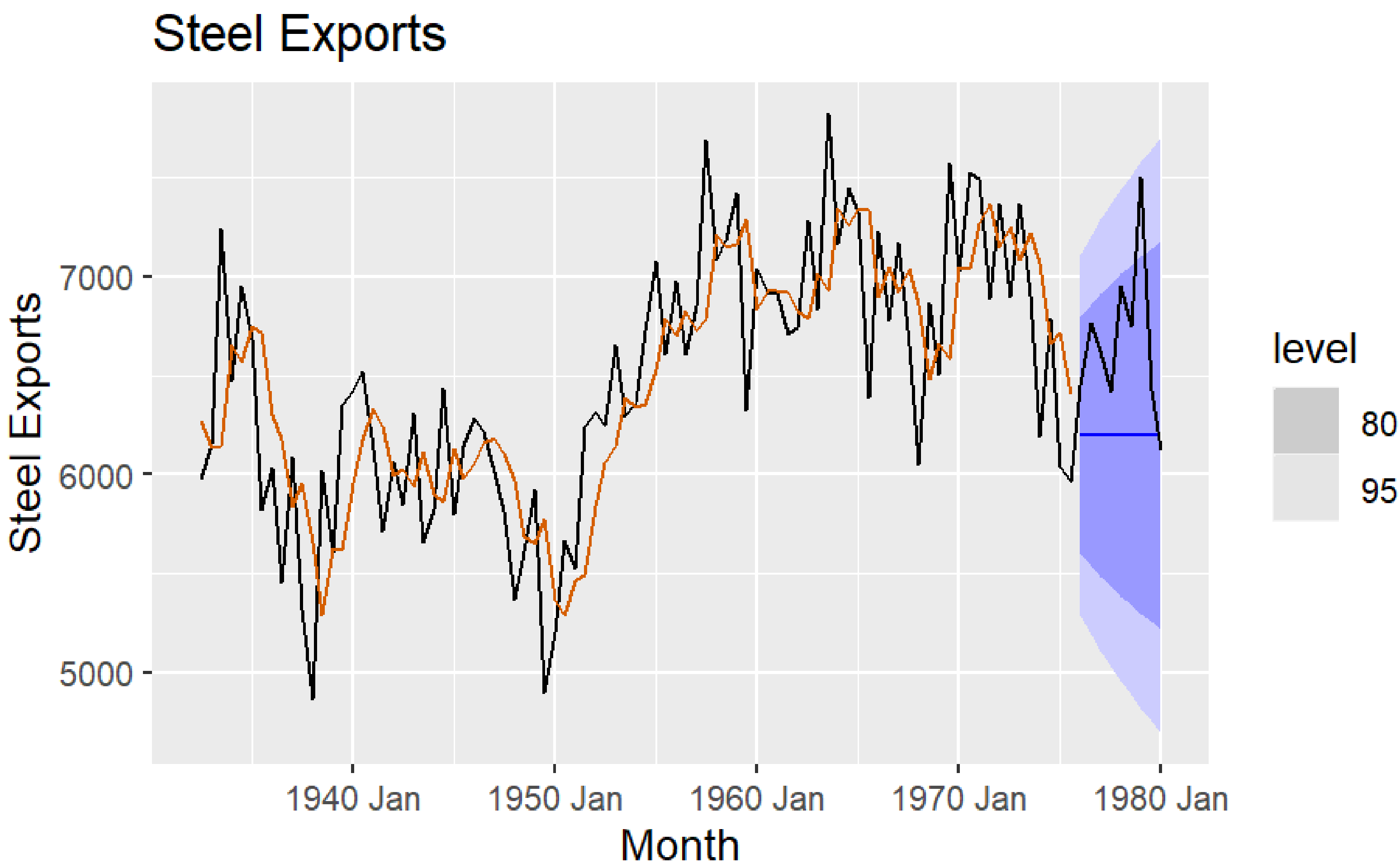
SES Function

```
steel_train <-steel_ts |> filter(year(date) <= 1975)
SES.Steel <- steel_train |>  model(ETS(steelshp ~ error("A")
+ trend("N") + season("N")))

report(SES.Steel)
```

Note: training data set has 87 observations and test data has 9 observations

```
Model: ETS(A,N,N)
  Smoothing parameters:
    alpha = 0.466543

  Initial states:
     l[0]
 6269.498

  sigma^2:  214894.3

     AIC     AICc      BIC
1460.688 1460.977 1468.086
```

```
Steel.for |>  autoplot(steel_ts) +  geom_line(aes(y = .fitted),
col="#D55E00",    data = augment(SES.Steel)) +  labs(y="Steel
Exports", title="Steel Exports") +  guides(colour = "none")
```



To get fitted values:

Fitted values of training data:

```
Steel_fitted <-fitted(SES.Steel)$.fitted
```

Fitted values of test data:

```
Steel.for <- SES.Steel |>  fabletools::forecast(h = 9)
Steel_test <- Steel.for$.mean
```

```
fabletools::accuracy(Steel.for, steel_ts)
```

A tibble: 1 × 10

.type <chr>	ME <dbl>	RMSE <dbl>	MAE <dbl>	MPE <dbl>
Test	468.4698	599.4653	486.2914	6.745431

1 row | 2-6 of 10 columns

LINEAR TREND FOR EXPONENTIAL SMOOTHING

Trending Exponential Smoothing

- The Single Exponential Smoothing model are better used for short-term forecasts.
- The SES model **cannot** adequately handle data that is trending up or down.
- There are different ways to incorporate a trend in the Exponential Smoothing Model.
  - Linear / Holt Exponential Smoothing
  - Damped Trend Exponential Smoothing

Linear / Holt Exponential Smoothing

- The Linear Exponential Smoothing model has two components.
- The second component incorporates trending into the model.

$$\hat{Y}_{t+h} = L_t + hT_t$$

$$L_t = \alpha Y_t + (1-\alpha)(L_{t-1} + T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$

Linear / Holt Exponential Smoothing

- The Linear Exponential Smoothing model has two components.
- The second component incorporates trending into the model.

$$\hat{Y}_{t+h} = L_t + hT_t$$

$$L_t = \alpha Y_t + (1 - \alpha)(L_{t-1} + T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

There are only two parameters to estimate here (both smoothing or “weight” parameters)

Damped Trend Exponential Smoothing

- The Linear Exponential Smoothing model has two components.
- The second component incorporates trending into the model.

$$\hat{Y}_{t+h} = L_t + \sum_{i=1} \phi^i \, T_t$$

$$L_t = \alpha Y_t + (1-\alpha)(L_{t-1} + \phi T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)\phi T_{t-1}$$

Damped Trend Exponential Smoothing

- The Linear Exponential Smoothing model has two components.
- The second component incorporates trending into the model.

Between 0 and 1

$$\hat{Y}_{t+h} = L_t + \sum_{i=1}^{h} \phi^i T_t$$

$$L_t = \alpha Y_t + (1-\alpha)(L_{t-1} + \phi T_{t-1})$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)\phi T_{t-1}$$

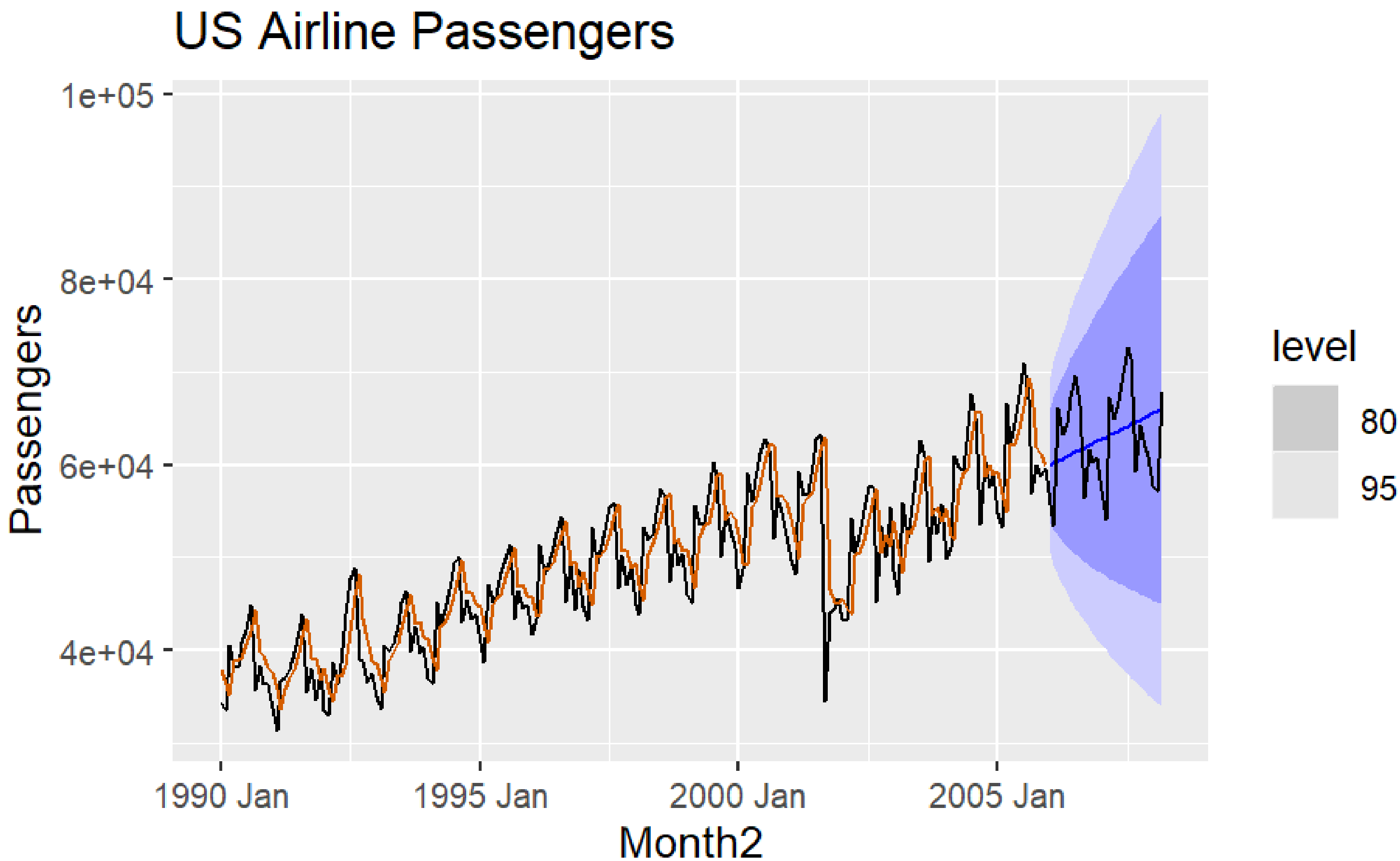
HOLT Function – R

```
USAirlines_ts <- USAirlines |>
mutate(date=myd(paste(Month, Year, "1"))) |>
mutate(Month2=yearmonth(date)) |>
as_tsibble(index=Month2)

air_train <-USAirlines_ts |> filter(year(date) <= 2005)

LES.air <- air_train |>  model(ETS(Passengers ~ error("A") +
trend("A") + season("N")))

air.for <- LES.air |>  fabletools::forecast(h = 27)
report(LES.air)
```

Holt output from R

```
Model: ETS(A,A,N)
  Smoothing parameters:
    alpha = 0.5860139
    beta  = 0.003740218

  Initial states:
     l[0]          b[0]
 37303.09 526.6595

  sigma^2:  23188824

     AIC     AICc      BIC
4271.560 4271.882 4287.847
```

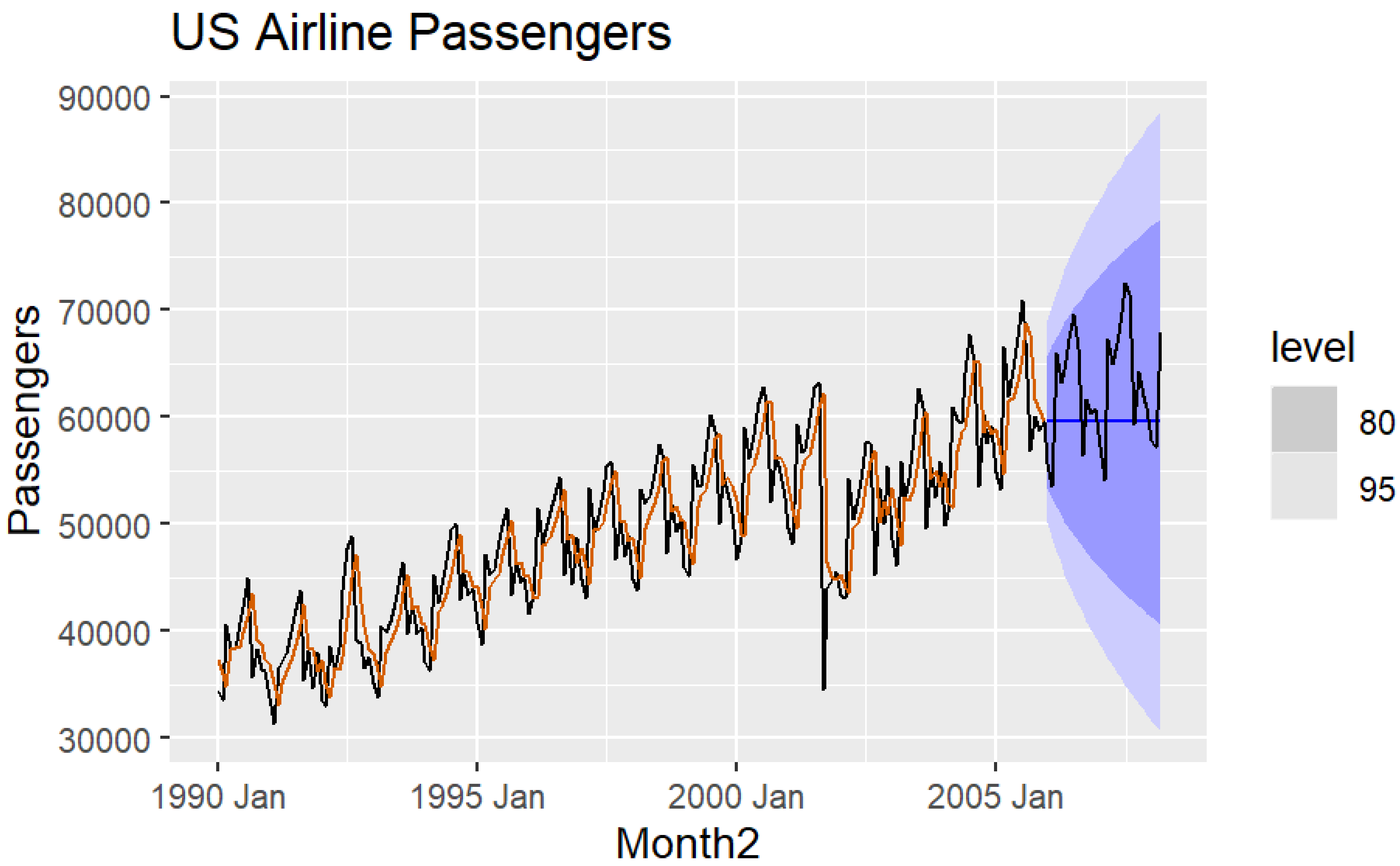
```
air.for |>  autoplot(USAirlines_ts) +  geom_line(aes(y = .fitted),
col="#D55E00", data = augment(LES.air)) +
labs(y="Passengers", title="US Airline Passengers") +
guides(colour = "none")
```



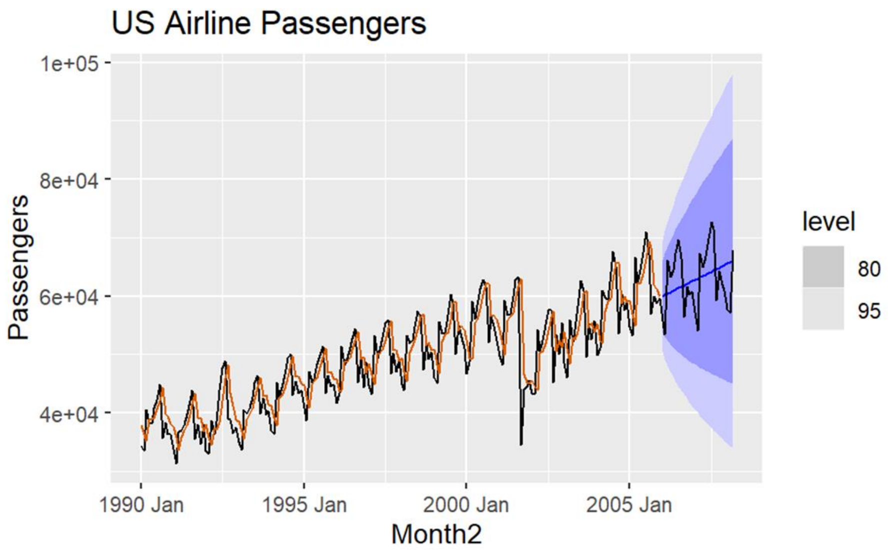
```
LdES.air <- air_train |>  model(ETS(Passengers ~
error("A") + trend("Ad") + season("N")))

air.for <- LdES.air |>  fabletools::forecast(h = 27)

report(LdES.air)
```

Damped Holt output from R (edited)

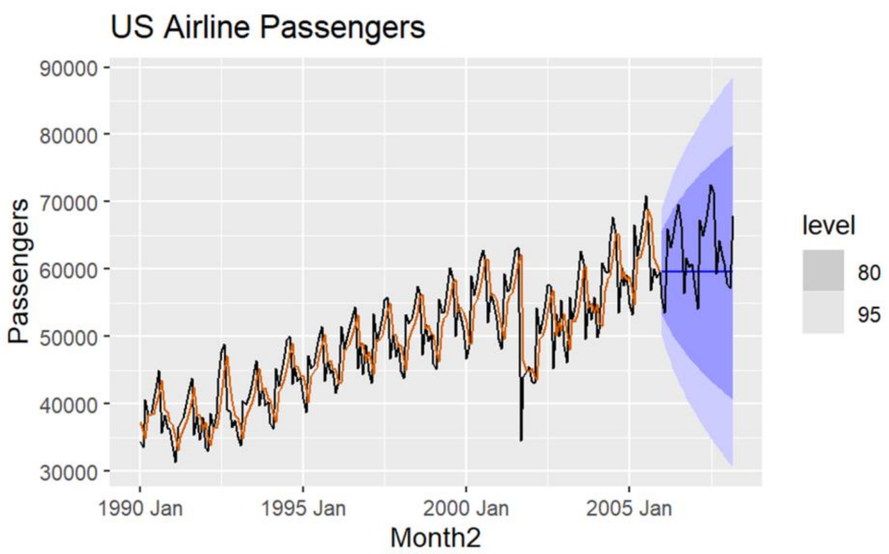
```
Model: ETS(A,Ad,N)
  Smoothing parameters:
    alpha = 0.5705768
    beta  = 0.0001003564
    phi   = 0.8085131

  Initial states:
     l[0]         b[0]
 36885.52 526.5046

  sigma^2:  23007825

     AIC     AICc      BIC
4271.031 4271.485 4290.576
```


US Airline Passengers
90000
80000
70000
60000
50000
40000
30000
Passengers
1990 Jan
1995 Jan
2000 Jan
2005 Jan
Month2
level
80
95

Holt Linear Model



Damped Linear Model



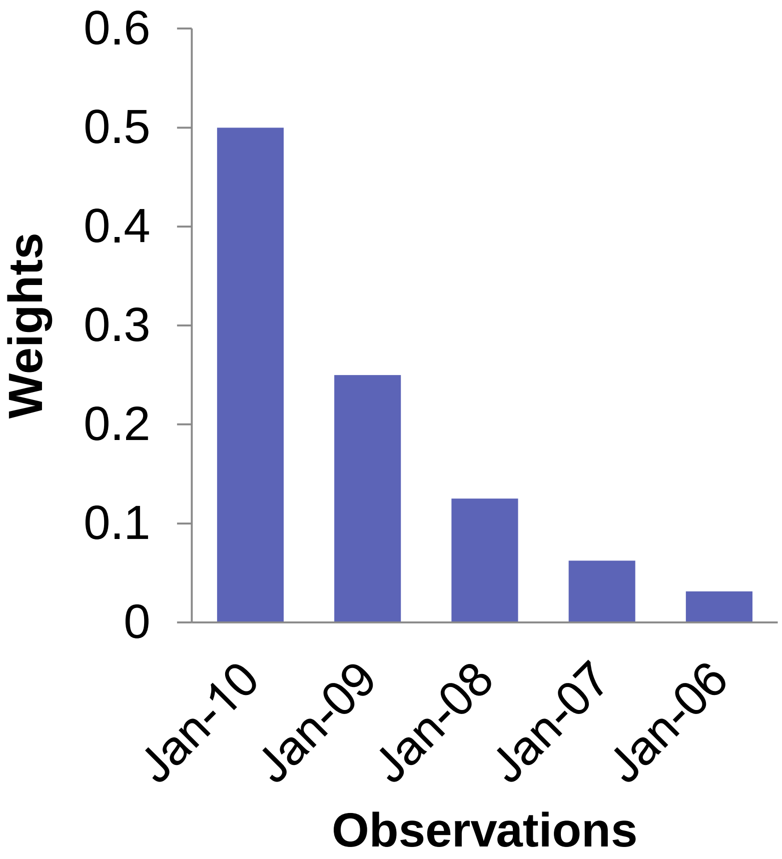
SEASONAL EXPONENTIAL SMOOTHING

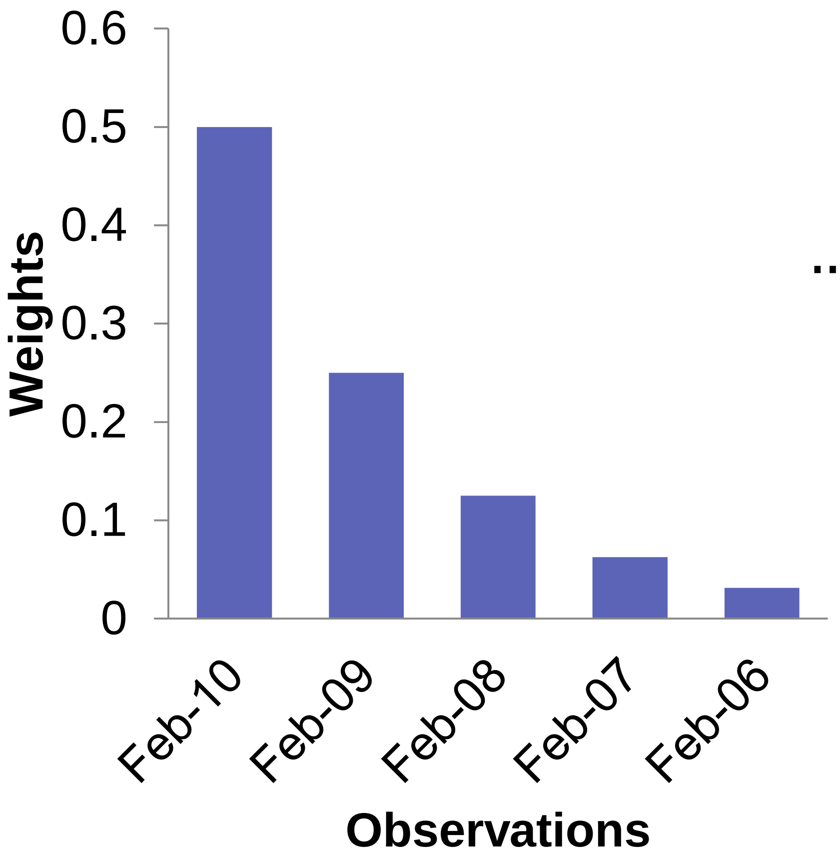
Seasonal Exponential Smoothing

- Exponential Smoothing models can also be adapted to account for seasonal factors.
- Seasonal models can be additive or multiplicative in the seasonal effect in the Exponential Smoothing Model.
  - Holt Winters Additive Exponential Smoothing (includes trend)
  - Holt Winters Multiplicative Exponential Smoothing (includes trend)

Seasonal Exponential Smoothing

- In seasonal exponential smoothing, weights decay with respect to the seasonal factor.





...

Winters / Triple Exponential Smoothing (Additive)

- The Linear Exponential Smoothing model has three components.
  - Level, Trend and Seasonal

$$\hat{Y}_{t+h} = L_t + hT_t + S_{t-p+h}$$

$$L_t = a\left(Y_t - S_{t-p}\right) + (1 - a)(L_{t-1} + T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$
$$S_t = \gamma(Y_t - L_{t-1} - T_{t-1}) + (1 - \gamma)S_{t-p}$$

Winters / Triple Exponential Smoothing (Multiplicative)

- The Linear Exponential Smoothing model has three components.

$$\hat{Y}_{t+h} = (L_t + hT_t)S_{t-p+h}$$

$$L_t = \alpha\left(Y_t/S_{t-p}\right) + (1-\alpha)(L_{t-1} + T_{t-1})$$
$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1}$$
$$S_t = \gamma(Y_t/(L_{t-1} + T_{t-1})) + (1-\gamma)S_{t-p}$$

HW Function

```
HWadd.air <- air_train |>  model(ETS(Passengers ~ error("A")
+ trend("A") + season("A")))

air.for <- HWadd.air |>  fabletools::forecast(h = 27)

report(HWadd.air)

###################################
HWmult.air <- air_train |>  model(ETS(Passengers ~
error(“M") + trend("A") + season("M")))

air.for <- HWmult.air |>  fabletools::forecast(h = 27)

report(HWmult.air)
```

```
Model: ETS(A,A,A)
  Smoothing parameters:
    alpha = 0.5913618
    beta  = 0.0001002723
    gamma = 0.0001000455


  sigma^2:  4099578

     AIC     AICc      BIC
3950.201 3953.718 4005.578

* Note: excluded initial states
```

ADDITIVE

Model: ETS(A,A,M)
  Smoothing parameters:
    alpha = 0.5132388
    beta = 0.006772624
    gamma = 0.0614166

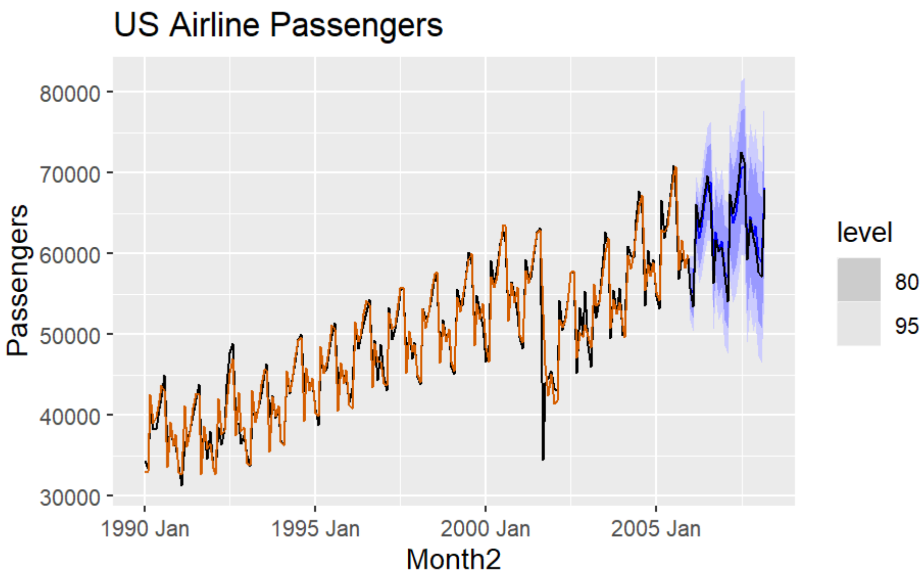
  sigma^2: 0.0015

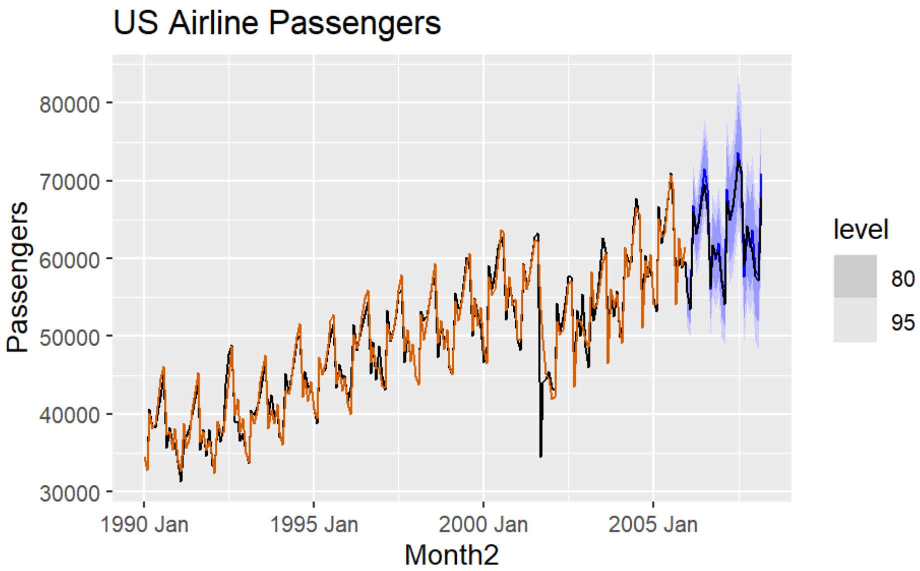
AIC	AICc	BIC
3922.549	3926.066	3977.926

* Note: excluded initial states

MULTIPLICATIVE

Additive Model versus Multiplicative Model





Compare models

```
air_fit <- air_train |>
model(
          SES = ETS(Passengers ~ error("A") + trend("N") + season("N")),
          `Linear` = ETS(Passengers ~ error("A") + trend("A") + season("N")),
          `Damped Linear` = ETS(Passengers ~ error("A") + trend("Ad") +
season("N")),
          HWAdd = ETS(Passengers ~ error("A") + trend("A") + season("A")),
          HWMult = ETS(Passengers ~ error("M") + trend("A") + season("M"))
)
air_fc <- air_fit |>  fabletools::forecast(h = 27)

fabletools::accuracy(air_fc, USAirlines_ts)
```

5 rows

.model <chr>	.type <chr>	ME <dbl>	RMSE <dbl>	MAE <dbl>	MPE <dbl>	MAPE <dbl>	MASE <dbl>	RMSSE <dbl>	ACF1 <dbl>
Damped Linear	Test	3351.23560	6251.053	5238.043	4.6475843	8.057662	1.8441468	1.6879095	0.3819013
HWAdd	Test	-175.40806	1461.985	1258.550	-0.4206807	2.024597	0.4430951	0.3947652	0.4476024
HWMult	Test	466.96181	1319.080	1110.115	0.7046275	1.750540	0.3908357	0.3561781	0.2185923
Linear	Test	-67.26453	5344.911	4736.428	-0.7887208	7.621637	1.6675443	1.4432330	0.4360965
SES	Test	3346.49137	6248.422	5236.041	4.6400058	8.054941	1.8434420	1.6871992	0.3818930

HW Multiplicative has best measures.

MORE ETS

ETS (Error, Trend, Season)

- ETS can also automatically select best model (well, based on certain information criterion)
  - For “Error”, the choices are Additive (A) or Multiplicative (M)
  - For “Trend”, the choices are None (N), Additive (A), Additive damped (Ad)
  - For “Seasonal”, the choices are None (N), Additive (A), Multiplicative (M)
- You can choose which one you want, OR you can let the computer choose
- Default is using the AICc to choose best model

Some notes:

- In the fable package, the damping parameter is restricted to be between $0.8 < \phi < 0.98$.
- You are not restricted to the default of AICc (can also use AIC or BIC).
- Some models may create division by zero and are not included in the search algorithm: they are ETS(A,N,M), ETS(A,A,M), and ETS(A,Ad,M).
- If data contains ANY negative values, multiplicative errors will not be considered

ETS

```
air_auto <- air_train |>  model(ETS(Passengers))

report(air_auto)
```

Note: if you want to use a different search criterion, you can simply change the above code to:

model(ETS(Passengers, ic= “bic”))   [to change to BIC]

R output (edited)

```
Model: ETS(M,Ad,M)
  Smoothing parameters:
    alpha = 0.6388447
    beta  = 0.0001026043
    gamma = 0.0001060611
    phi   = 0.979993

sigma^2:  0.0014

     AIC     AICc      BIC
3910.149 3914.103 3968.784
```

A bit suspicious here…log-based scale?

```
air_fit <- air_train |>
  model(
        HWAdd = ETS(Passengers ~ error("A") + trend("A") + season("A")),
        HWMult = ETS(Passengers ~ error("A") + trend("A") + season("M")),
        AutoETS = ETS(Passengers ~ error("M") + trend("Ad") + season("M"))  )

air_fc <- air_fit |>  fabletools::forecast(h = 27)

fabletools::accuracy(air_fc, USAirlines_ts)
```

.model	.type	ME	RMSE	MAE
<chr>	<chr>	<dbl>	<dbl>	<dbl>
AutoETS	Test	1733.6577	2342.516	2019.5856
HWAdd	Test	-175.4081	1461.985	1258.5500
HWMult	Test	-674.3134	1317.872	979.6035

MPE	MAPE	MASE	RMSSE	ACF1
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
2.7527821	3.188625	0.7110313	0.6325264	0.5962021
-0.4206807	2.024597	0.4430951	0.3947652	0.4476024
-1.0562242	1.559243	0.3448870	0.3558519	0.1433051

Questions??