



Inspire...Educate...Transform.

Statistics and Probability in Decision Modeling

Regularization, Time Series Forecasting,

Dr. Venkatesh Sunkad

Jan 12, 2019

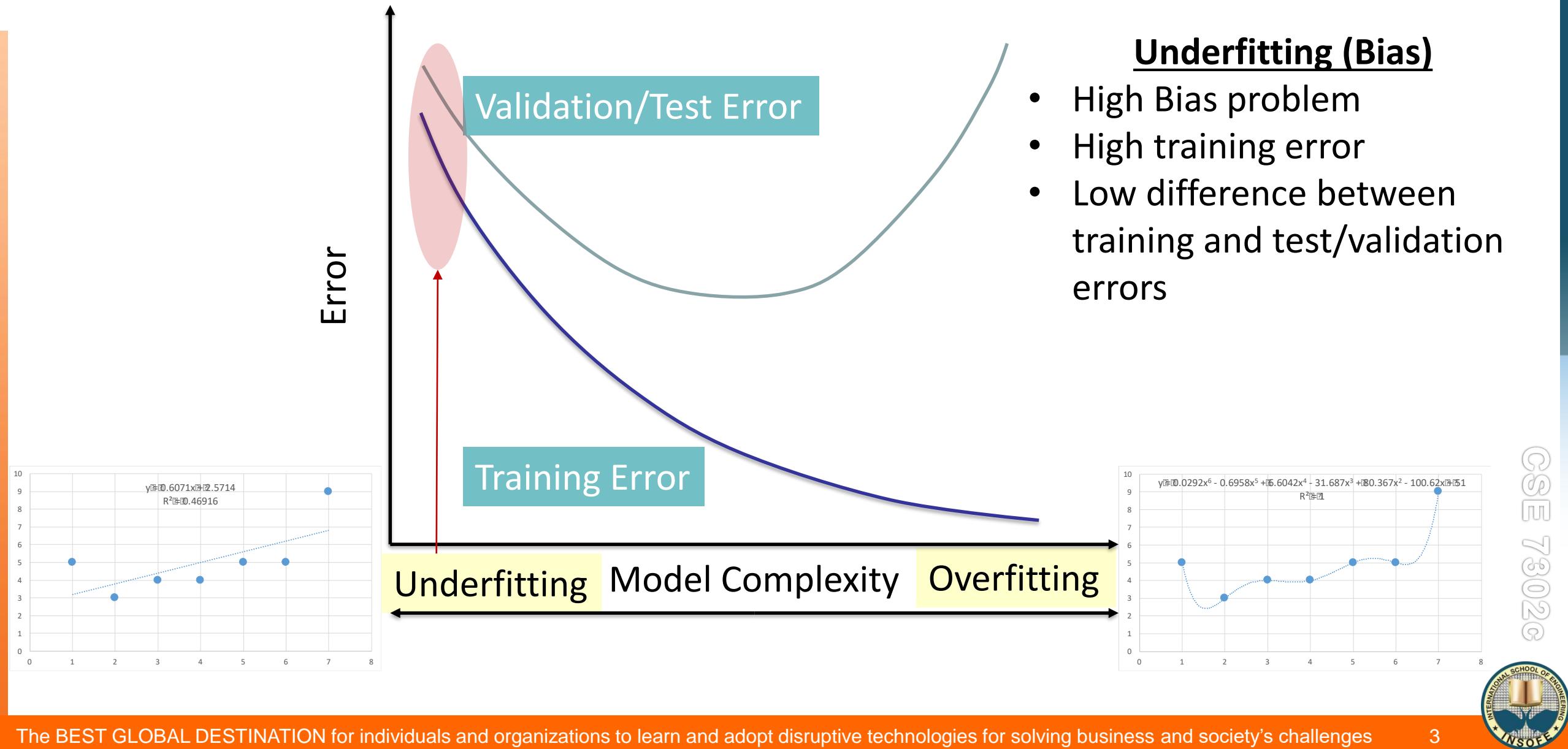
Regularization and Naïve Bayes slides courtesy Dr. Anand Jayaraman



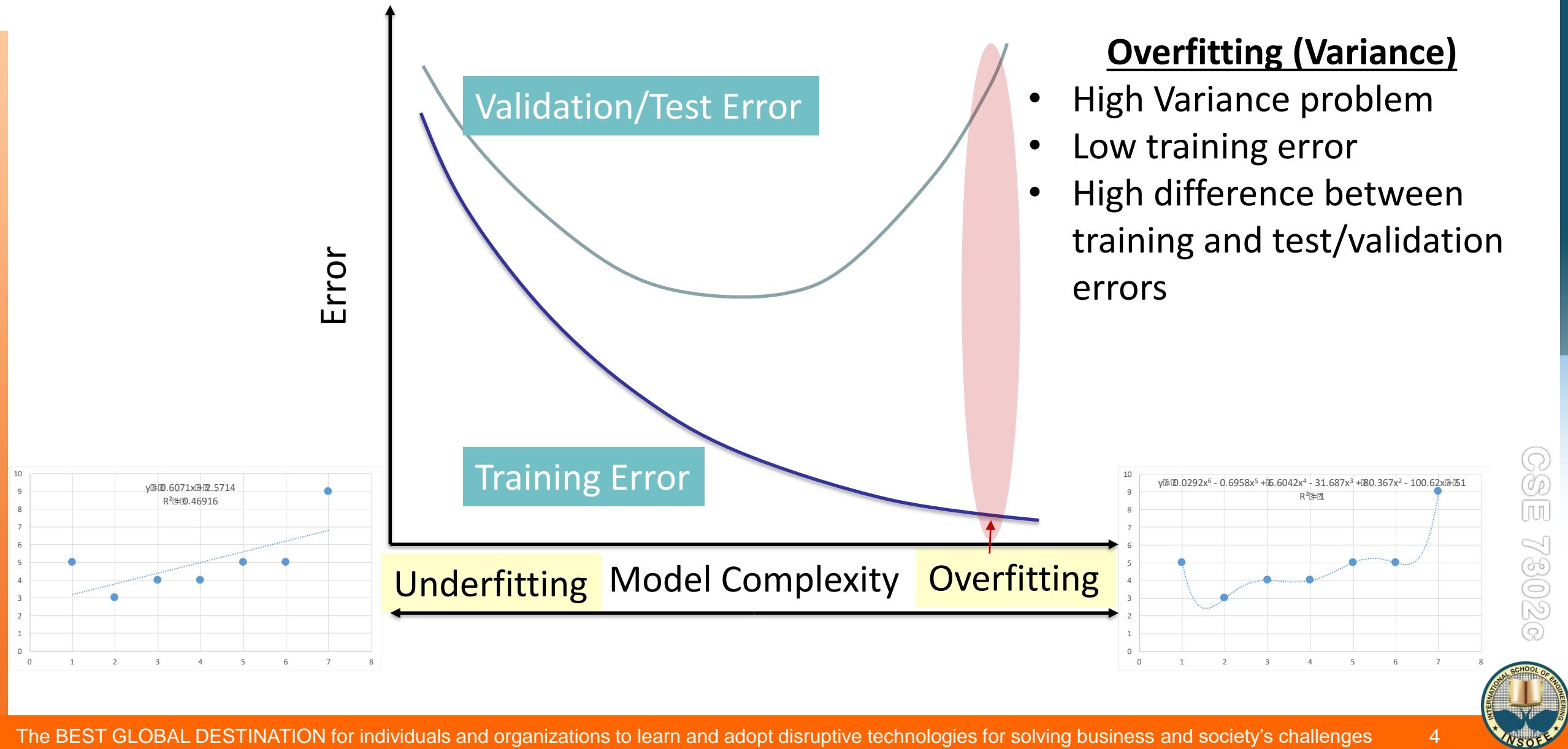
Advanced Regression Methods: Ridge Regression, LASSO Regression, Elastic Nets

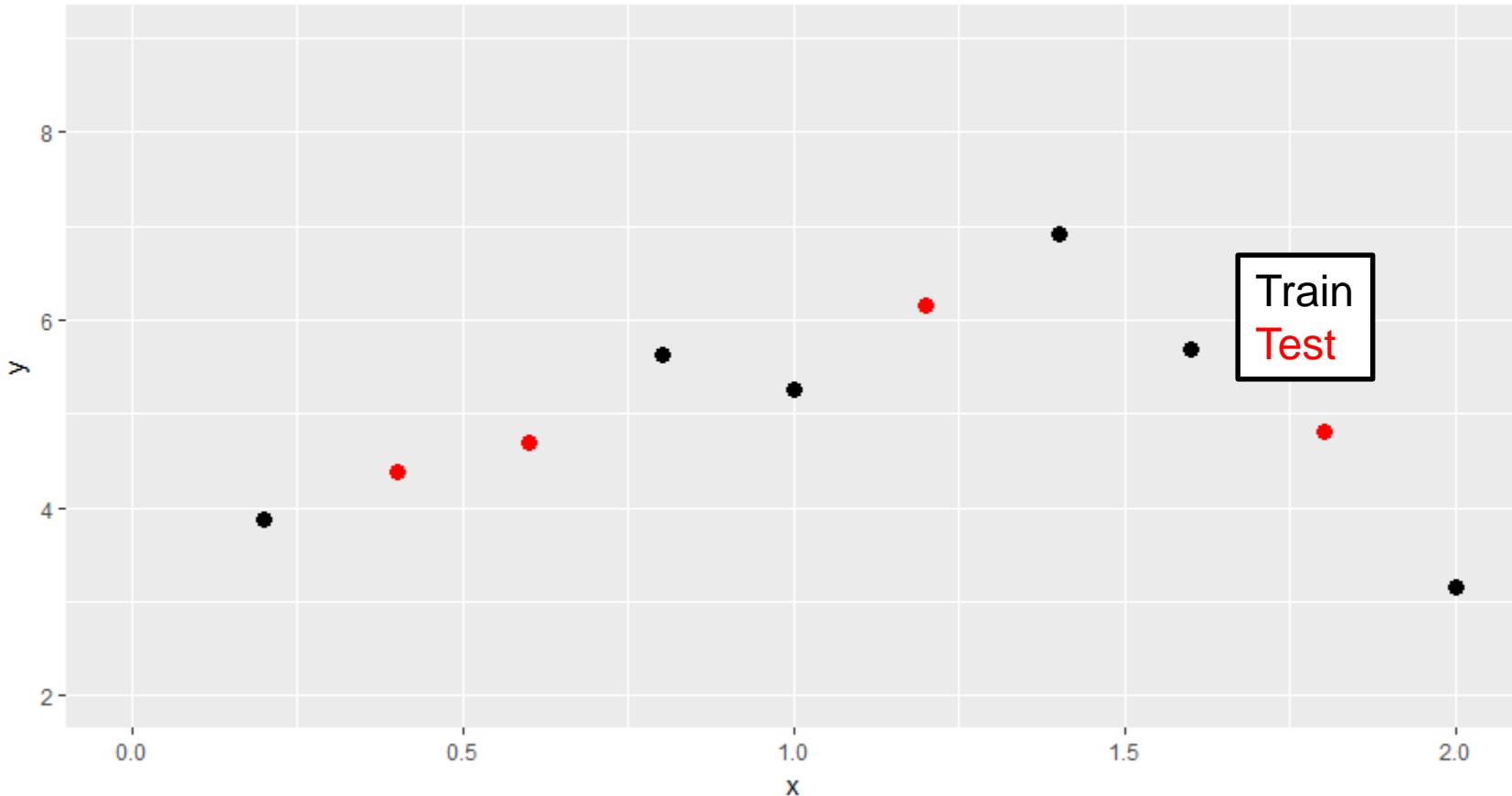
REGULARIZATION

Diagnosing Bias and Variance



Diagnosing Bias and Variance

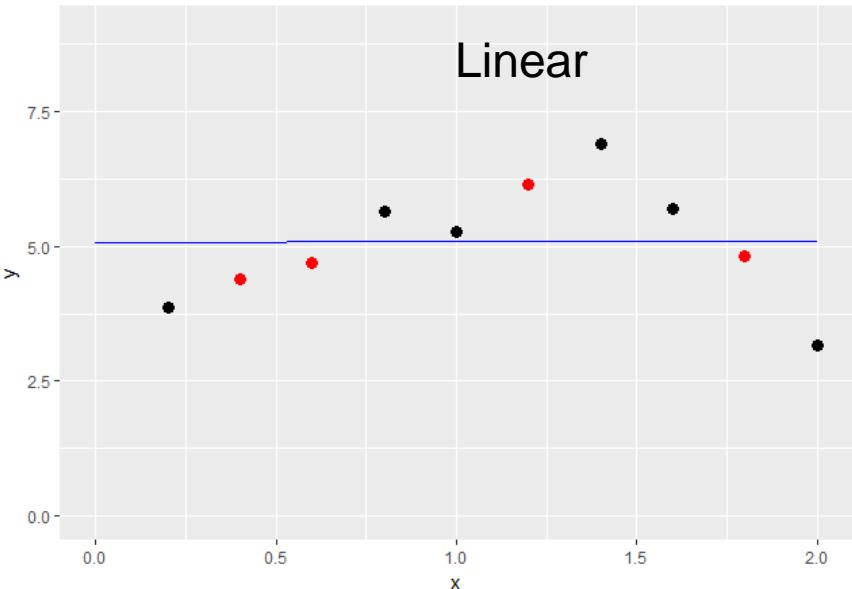




Decision point: What order polynomial should we use for the fit?

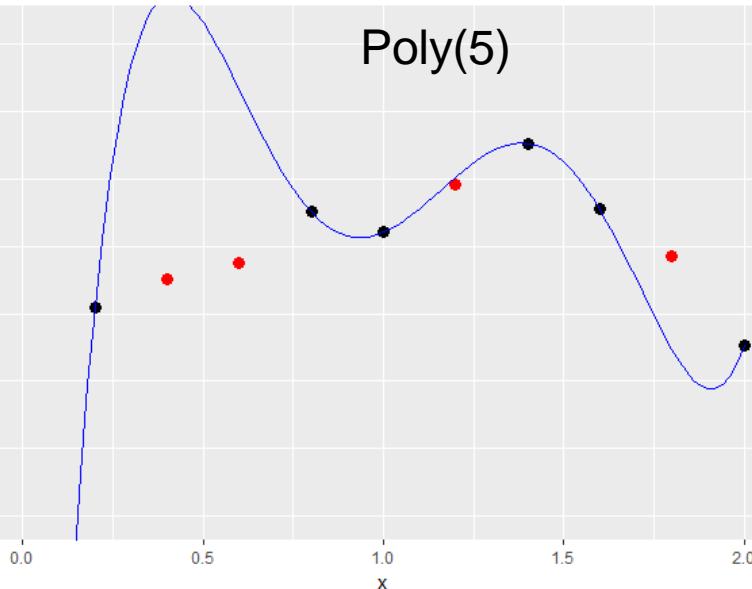
Problem with Overfitting

Linear



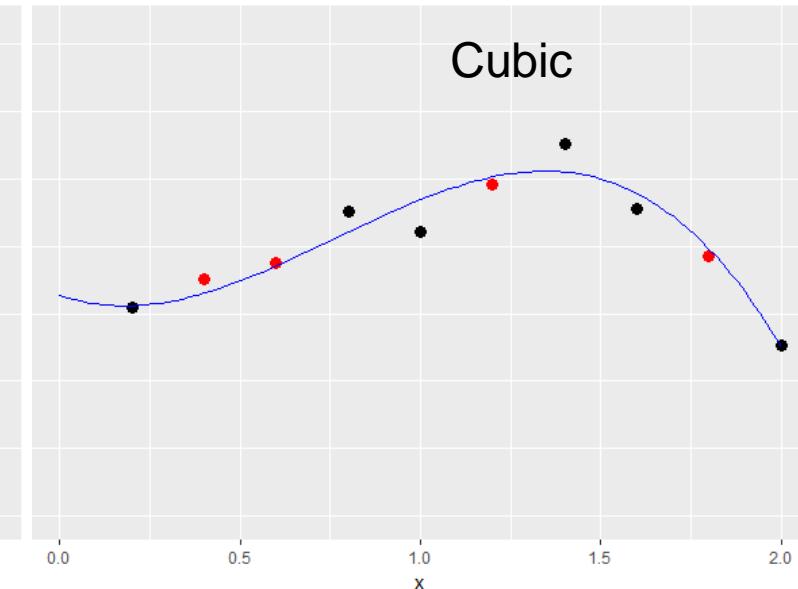
Underfit
TrainErr=1.23
TestErr=0.7

Poly(5)



Overfit
TrainErr=0
TestErr=3.2

Cubic



Right fit
TrainErr=0.38
TestErr=0.17

Overfitting Problem

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \beta_4 x^4 + \dots$$

How many β_i do we need for a reasonable fit, while at the same time minimizing the risk of overfitting?

In general, given a regression problem with several features, is there a way to determine how many β_i do we need for a reasonable fit, while at the same time minimizing the risk of overfitting?

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \dots$$



Regularization

- Standard Least Squares Regression involves minimizing SSE

$$\text{SSE} = \sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T \cdot (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

\mathbf{y} = vector of all training responses y_j

\mathbf{X} = matrix of all training samples \mathbf{x}_j

- We can reduce overfitting by penalizing large coefficients (**must standardize** in order to be able to compare/add β s)

Ridge Regression

$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^d \beta_i^2 \right)$$

- Penalization of large coefficients through L2 norm (another name for Least Squares)
- Regularization parameter λ controls the extent of penalization
- $\lambda = 0$ corresponds to Least Squares Regression

Ridge Regression

- Occam's Razor

$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^d \beta_i^2 \right)$$

- Least Squares Regression

- Fails when the number of features d is larger than number of data points N , e.g., in genetics studies. For example, there are infinite solutions for $5 = \beta_1 * 3 + \beta_2 * 2$ (two unknowns and one equation)



Ridge Regression

- Least Squares Regression
 - Fails when we have perfect multicollinearity

$y = x_1 + x_2$ or $y = 0.5x_1 + 1.5x_2$ or $y = 2x_1$ and so on

$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^d \beta_i^2 \right)$$

y	x1	x2
0	0	0
2	1	1
4	2	2
6	3	3

- Ridge Regression allows for a unique solution even in above situation. Which equation will you choose if you add the constraint to minimize $\sum_{i=1}^d b_i^2$ as well?

Ridge Regularization

The penalty term “regularizes” the problem when there is perfect multicollinearity

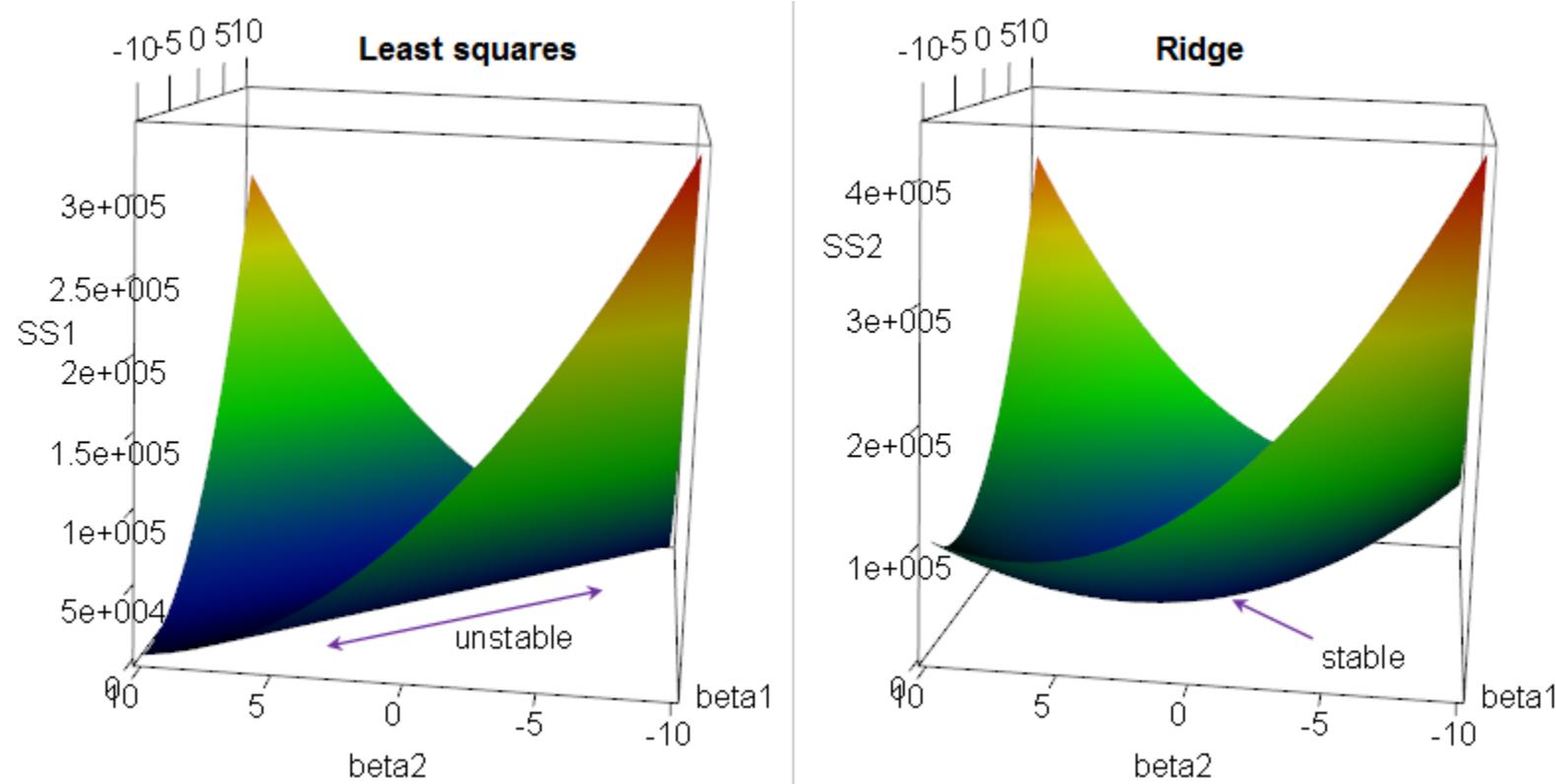
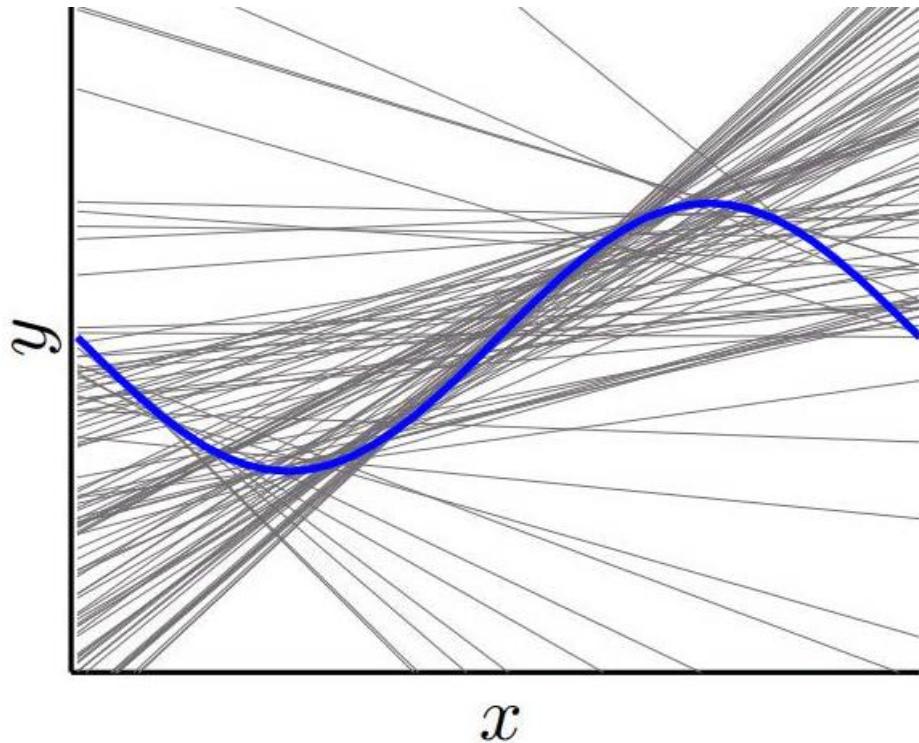
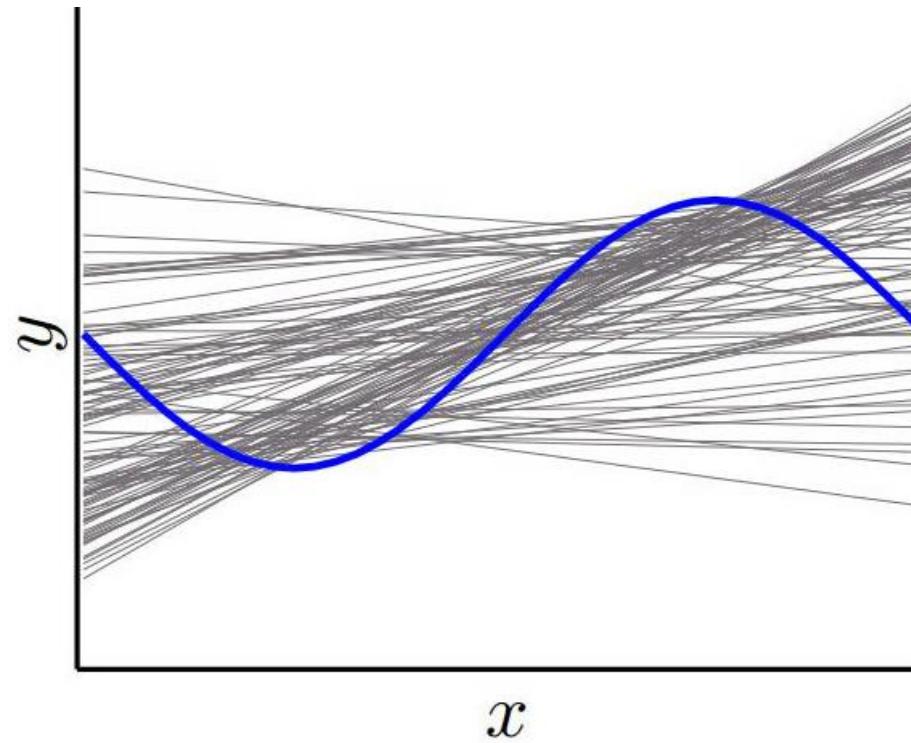


Image Source: <https://stats.stackexchange.com/questions/151304/why-is-ridge-regression-called-ridge-why-is-it-needed-and-what-happens-when>

Effect of Regularization



without regularization

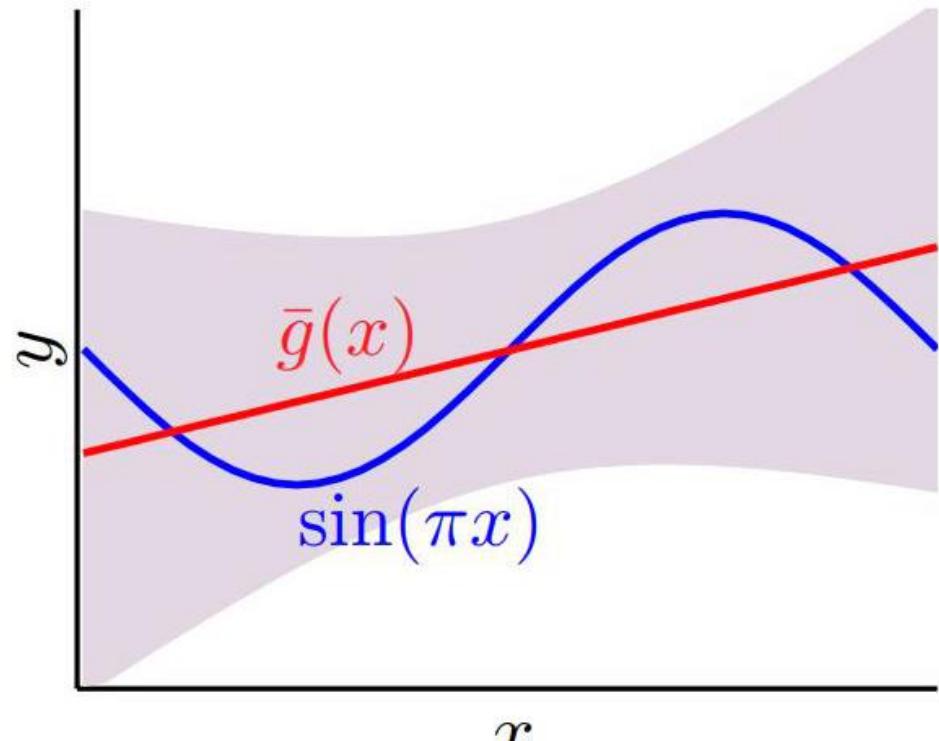


with regularization

Source: <http://work.caltech.edu/slides/slides12.pdf>

Effect of Regularization

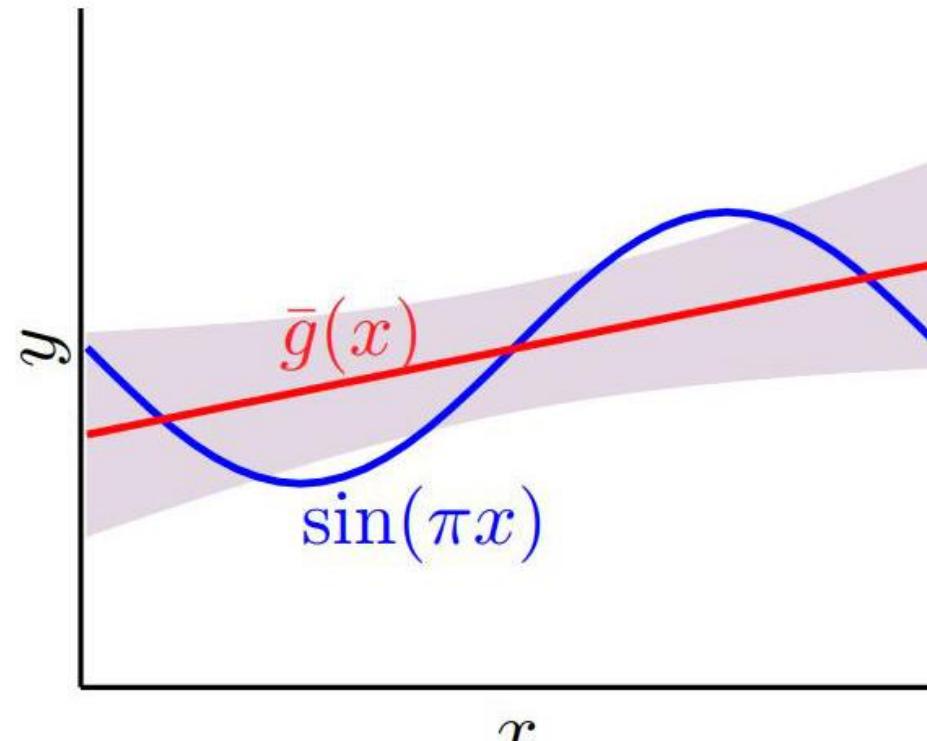
without regularization



bias = **0.21**

var = **1.69**

with regularization



bias = **0.23**

var = **0.33**

Source: <http://work.caltech.edu/slides/slides12.pdf>

LASSO Regularization

$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^d |\beta_i| \right)$$

- Penalization of large coefficients through L1 norm (another name for Least Absolute Error)
- Least Absolute Shrinkage and Selection Operator (LASSO)



LASSO Regularization

$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^d |\beta_i| \right)$$

Not useful when there is strong multicollinearity.

$y = x_1 + x_2$ or $y = 0.5x_1 + 1.5x_2$ or $y = 2x_1$ and so on

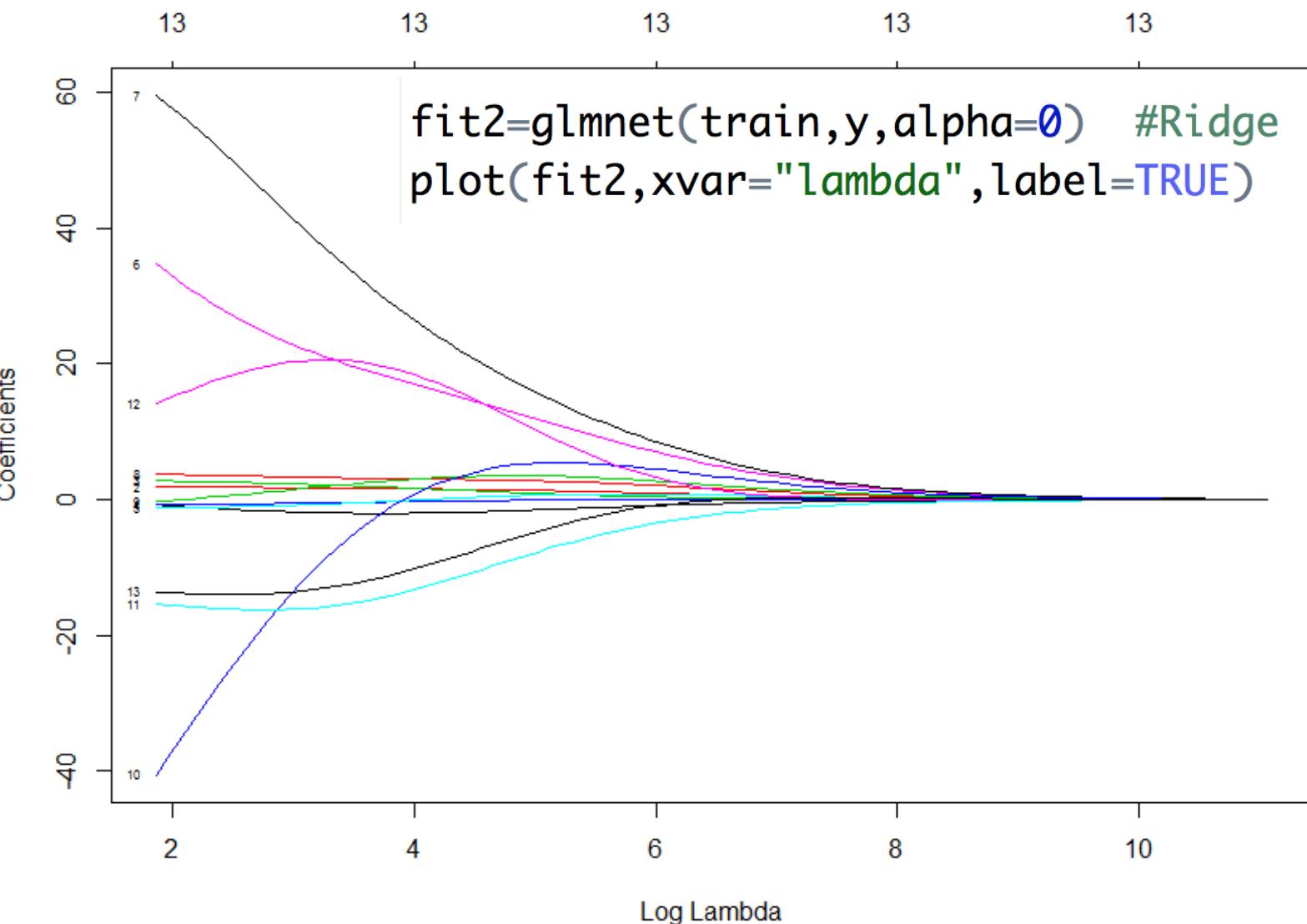
y	x1	x2
0	0	0
2	1	1
4	2	2
6	3	3

Regularization – Toy Purchase Data



CustomerID	City	NoOfChildren	MinAgeOfCh	MaxAgeOfCh	Tenure	FrqnacyOfPu	NoOfUnitsPu	FrequencyOF	NoOfGames	NoOfGames	FavoriteChar	FavoriteGam	TotalRevenueGenerated	
1001		1	2	3	8	210	11	11	2344	108	10	Uniform	Uniform	107.51
1002		1	2	3	6	442	20	20	245	22	7	Favorite	Uniform	382.4
1003		1	4	3	5	424	18	18	1059	130	18	Favorite	Uniform	135.01
1004		1	1	6	6	261	11	9	365	34	11	Favorite	Uniform	125
1005		1	3	6	9	422	44	31	1066	102	44	Uniform	Uniform	335.05
1006		1	2	3	4	378	16	16	228	12	16	Favorite	Favorite	150
1007		1	3	8	12	369	25	15	75	2	25	Favorite	Favorite	127.5
1008		1	2	6	8	404	13	12	1488	118	13	Favorite	Uniform	122.5
1009		1	4	6	9	420	20	16	2743	163	16	Uniform	Uniform	164.96
1010	2	3	5	6	333	15	15	1967	56	15	Favorite	Uniform	112.62	
1011	1	1	5	5	323	9	9	218	1	3	Favorite	Favorite	211.92	
1012	1	3	4	9	167	17	17	1961	124	17	Uniform	Uniform	125.04	
1013	1	1	4	4	381	15	15	267	51	15	Favorite	Uniform	167.5	
1014	1	4	5	7	399	36	24	6411	355	36	Favorite	Uniform	202.5	
1015	1	2	4	7	444	14	14	2804	133	14	Uniform	Uniform	105.5	
1016	1	2	3	5	371	6	6	1746	61	6	Favorite	Uniform	117.5	
1017	1	1	6	6	355	11	11	1367	49	11	Favorite	Uniform	175	
1018	1	1	7	7	346	18	16	1029	106	14	Uniform	Uniform	181.71	
1019	1	2	6	7	211	7	7	428	26	0	Favorite	Uniform	144.93	
1020	1	2	4	7	278	12	12	1080	49	8	Uniform	Uniform	182.36	
1021	1	3	4	10	448	6	6	1016	56	6	Favorite	Uniform	100	
1022	1	2	3	6	447	14	14	1617	58	11	Favorite	Uniform	199.93	
1023	1	1	8	8	471	18	18	2131	67	18	Favorite	Uniform	172.5	
1024	1	1	4	4	189	22	18	267	33	12	Favorite	Uniform	348.09	
1025	1	2	5	7	427	16	16	3781	288	16	Favorite	Uniform	185.02	
1026	1	2	5	8	351	5	5	182	26	3	Favorite	Uniform	139.99	
1027	1	2	6	6	453	12	10	4036	146	12	Uniform	Uniform	135	
1028	1	4	3	7	418	51	41	2947	241	51	Uniform	Uniform	388.01	
1029	1	3	3	13	424	17	17	1606	132	17	Favorite	Uniform	152.51	
1030	1	5	2	7	444	26	26	7766	583	26	Favorite	Uniform	205.01	
1031	1	1	7	7	126	8	8	515	21	8	Favorite	Uniform	100	
1032	1	1	7	7	290	9	9	1233	82	8	Favorite	Uniform	144.99	
1033	1	4	3	7	448	20	10	4788	370	20	Uniform	Uniform	117.5	
1034	1	2	4	6	181	14	14	221	19	14	Favorite	Uniform	125	

Ridge: Coefficients



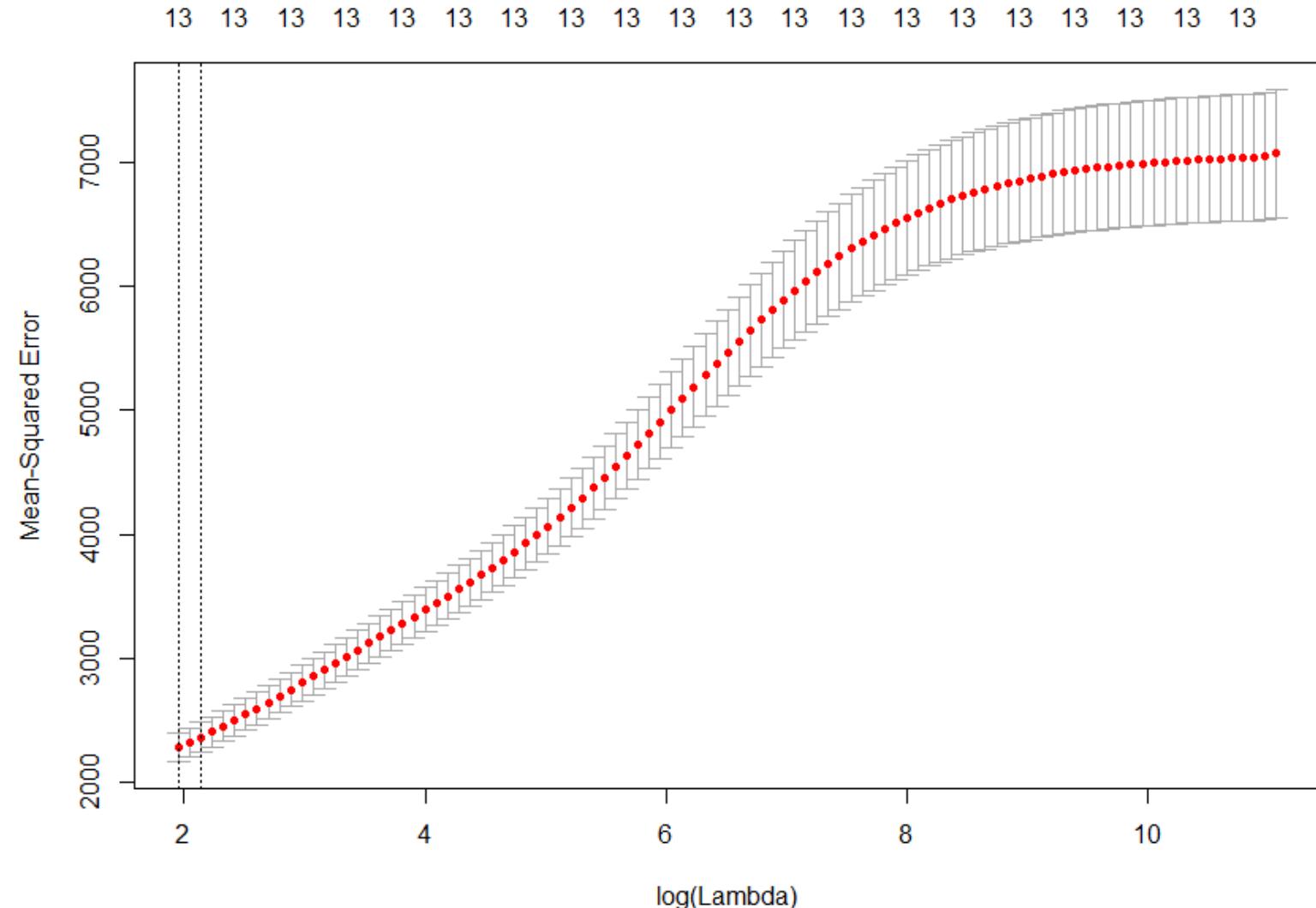
$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^d \beta_i^2 \right)$$

How to choose λ

- Choose a value for λ .
- Run k -fold cross validation and calculate mean “squared” error on the **test dataset**.
- Repeat the above 2 steps for various λ values.
- Pick λ that gives least error. R recommends the range from min λ to 1 stdev away.
- Pick λ by looking at the coefficients corresponding to these λ values, in addition to the minimum error.

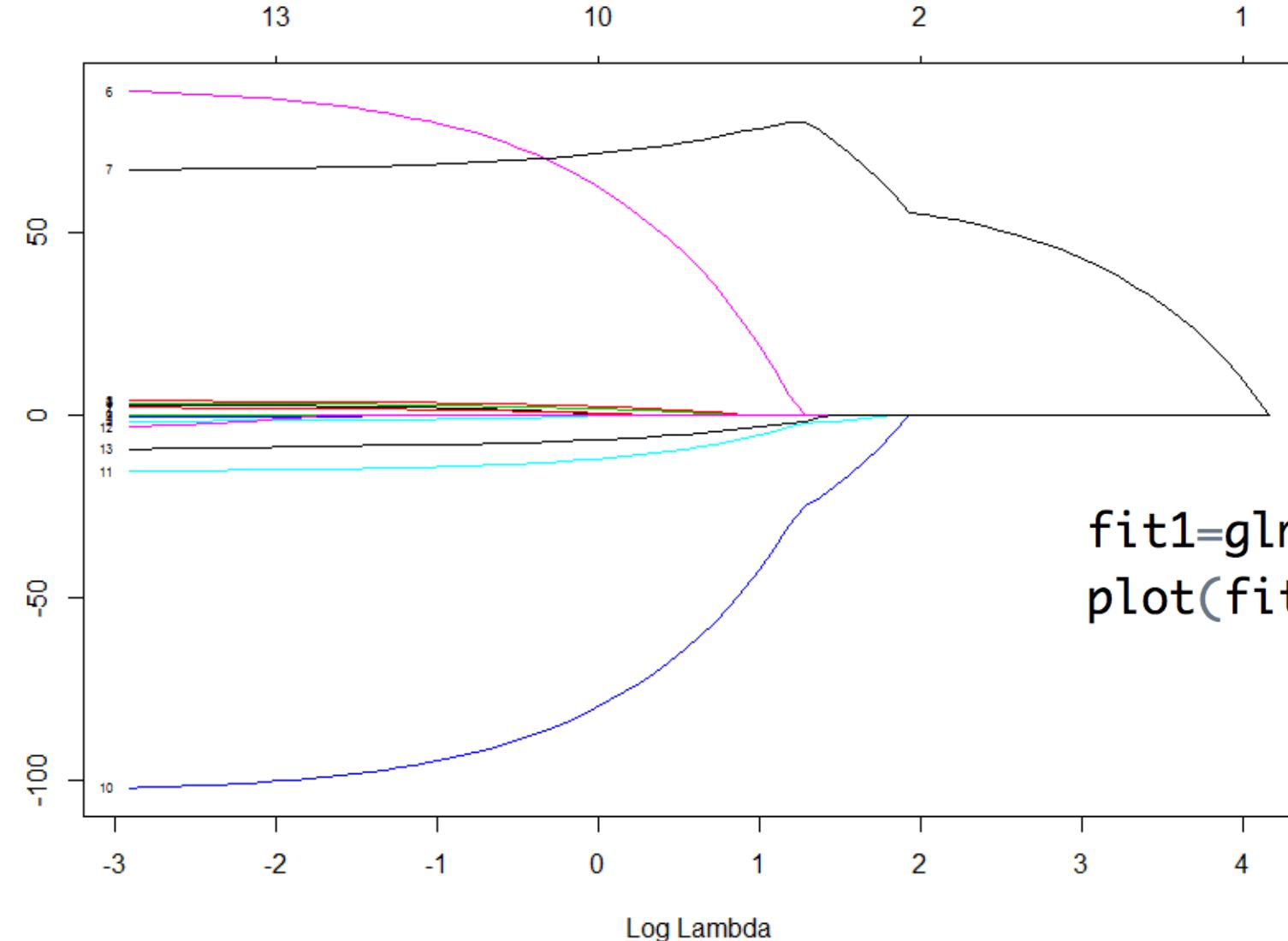


Ridge: MSE

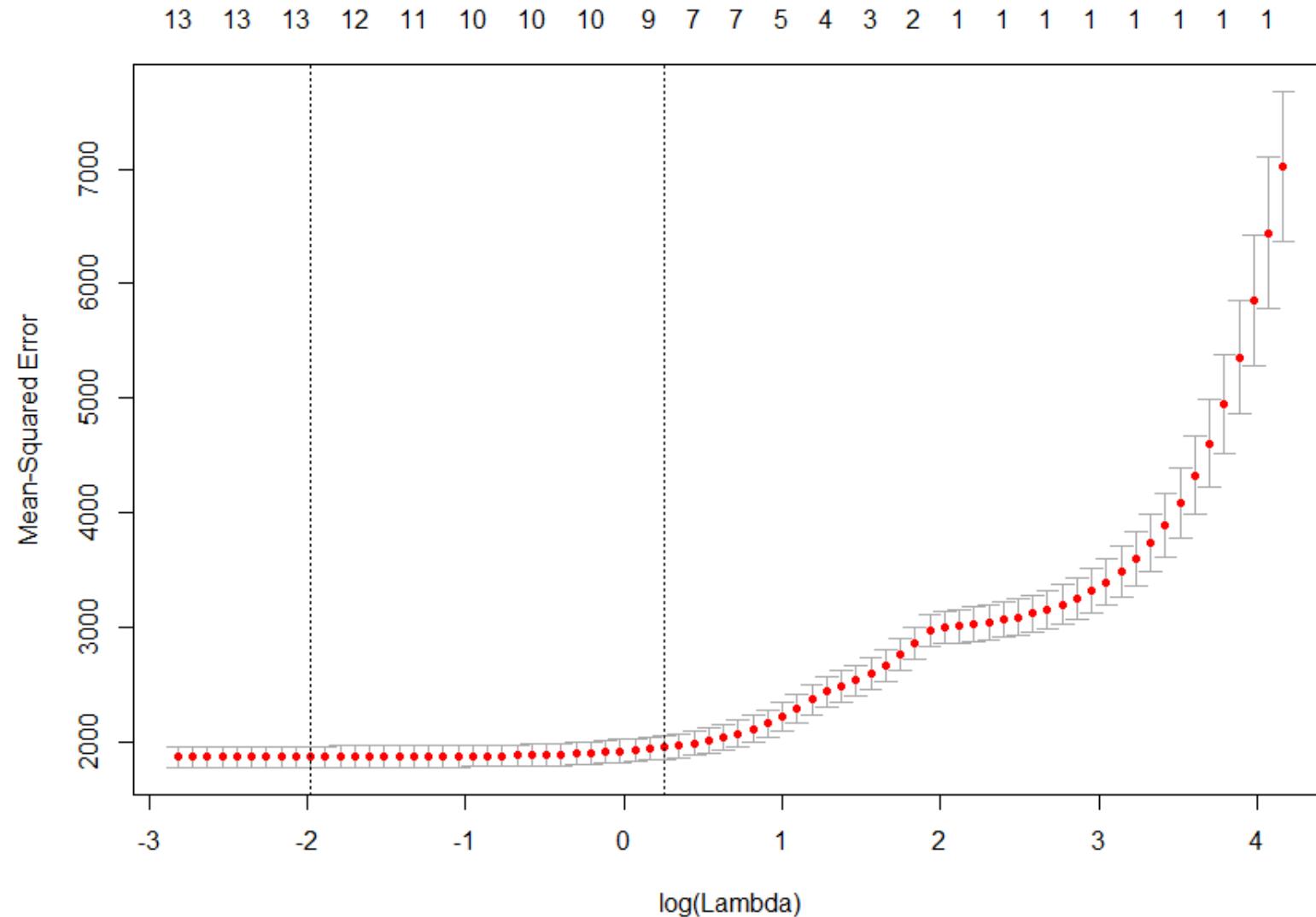


LASSO: Coefficients

$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \sum_{i=1}^d |\beta_i| \right)$$



LASSO: MSE



Ridge vs Lasso Regression

Ridge

- Reduces the magnitude of coefficients (does not make them zero)
- Solution is unique – however, doesn't do feature selection since it keeps all the terms
- Stable even if $n \ll p$
- Very stable even in multicollinear case

LASSO

- Values of irrelevant coefficients set to zero
- Good for feature selection
- Doesn't do well when $n \ll p$
- Out of sample performance can be impacted for multicollinear case

CSE 7302C



Elastic-net

- A mix-between Ridge and LASSO regression

$$\min_{\beta} \left(\sum_{j=1}^N (y_j - \sum_{i=0}^d \beta_i \cdot x_i)^2 + \lambda \left(\sum_{i=1}^d (\alpha |\beta_i| + (1-\alpha) \beta_i^2) \right) \right)$$

- $\alpha=0$ is Ridge
- $\alpha=1$ is LASSO
- $\alpha=0.5$ is an equal mix of the two

- Keep in mind that Ridge regression can't zero out coefficients; thus, you either end up including all the coefficients in the model, or none of them.
- In contrast, the LASSO does both parameter shrinkage and variable selection automatically.
- If some of your covariates are highly correlated, you may want to look at the Elastic Net instead of the LASSO.





TIME SERIES FORECASTING

Why Time Series

Causal independent variables are

- Unknown to us
- Not available
- Might not fit the data well
- Difficult to forecast



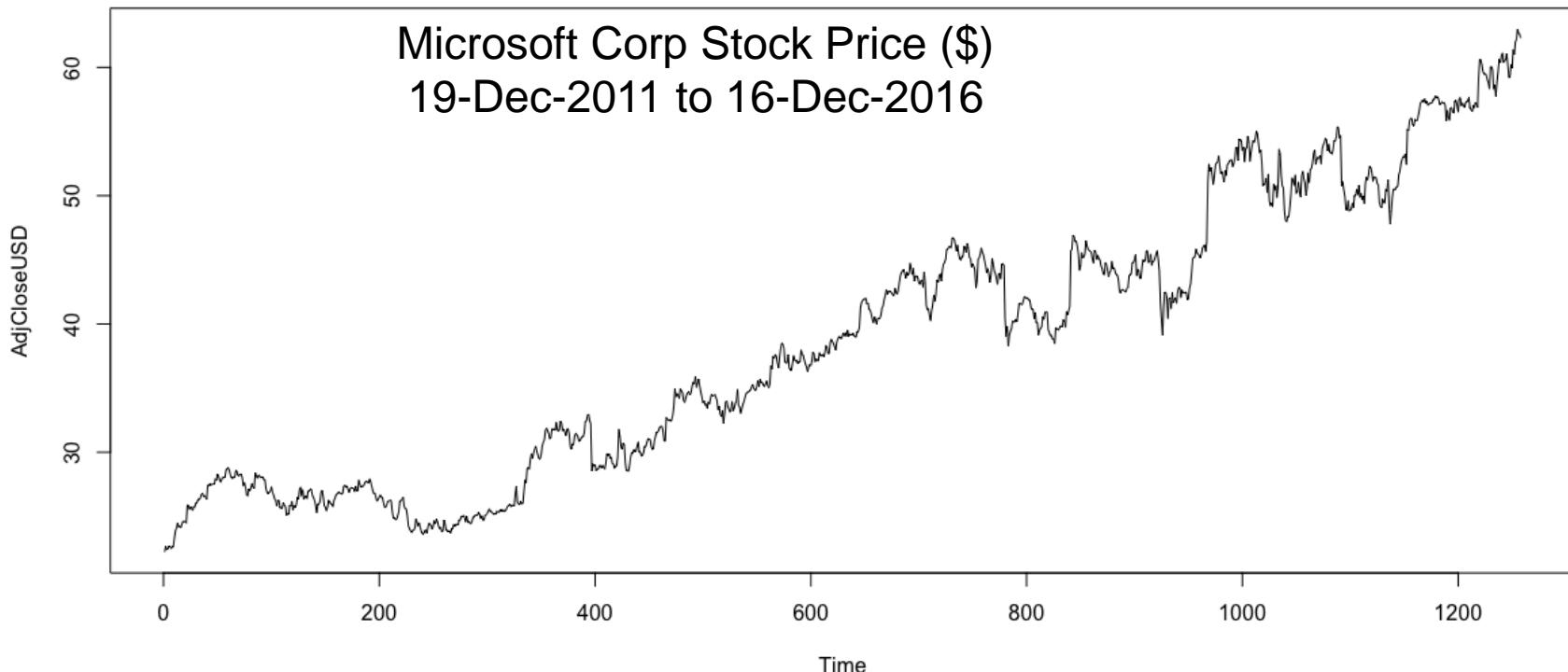
Typical Time Series

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, y_{t-2} \dots) + f(x_1, x_2, x_3 \dots)$$

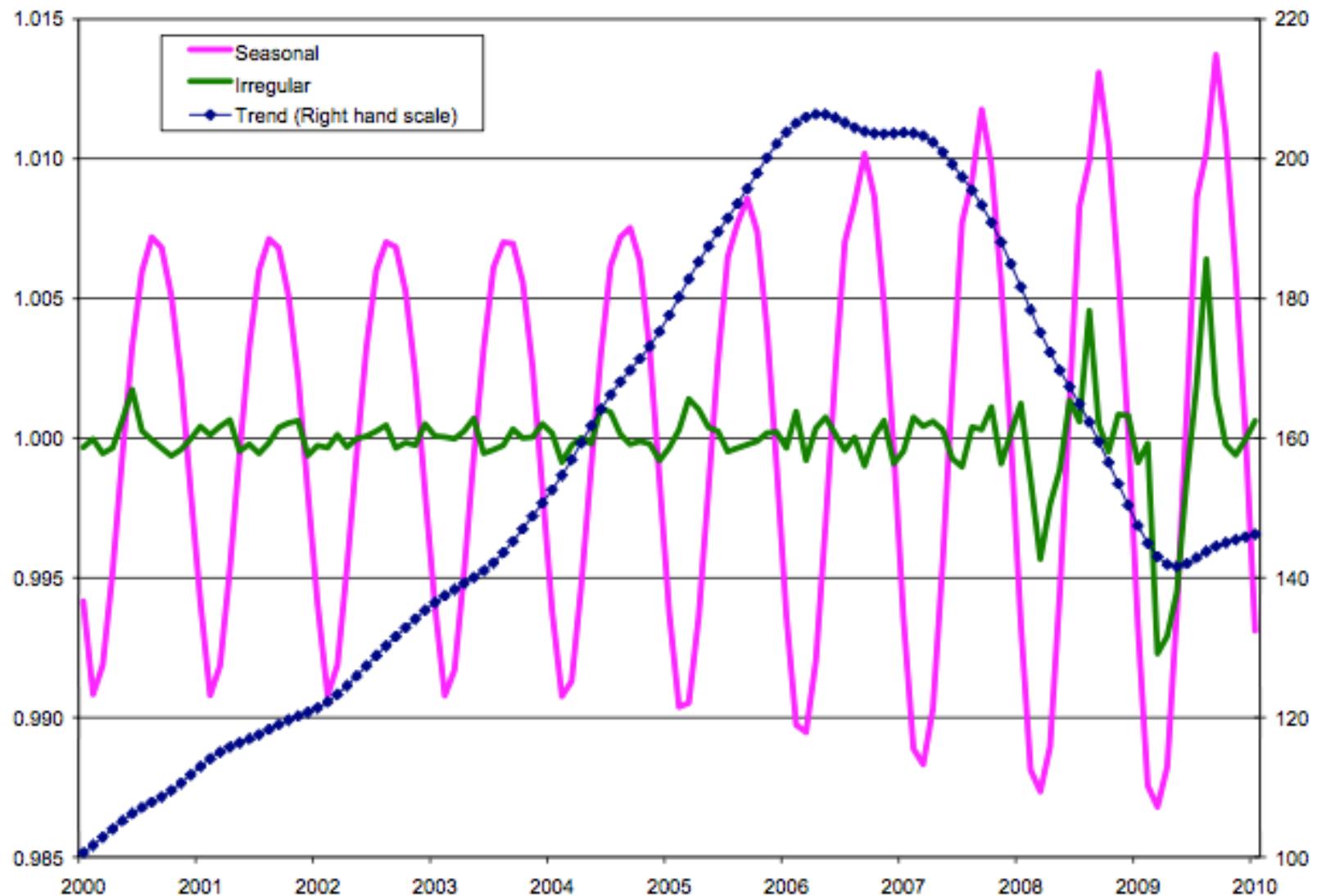
f can be linear or nonlinear

Components of Time Series

- Trend
- Seasonality
- Random component



Trend, Seasonality and Randomness



US Air Carrier Traffic – Revenue Passenger Miles ('000) RPM

Month	U.S. Air Carrier Traffic Statistics - Revenue Passenger Miles
1996-01	41972194
1996-02	42054796
1996-03	50443045
1996-04	47112397
1996-05	49118248
1996-06	52880510
1996-07	55664750
1996-08	57723208
1996-09	47035464
1996-10	49263120
1996-11	43937074
1996-12	48539606
1997-01	45850623
1997-02	42838949
1997-03	53620994
1997-04	49282817
1997-05	51191842
1997-06	54707221

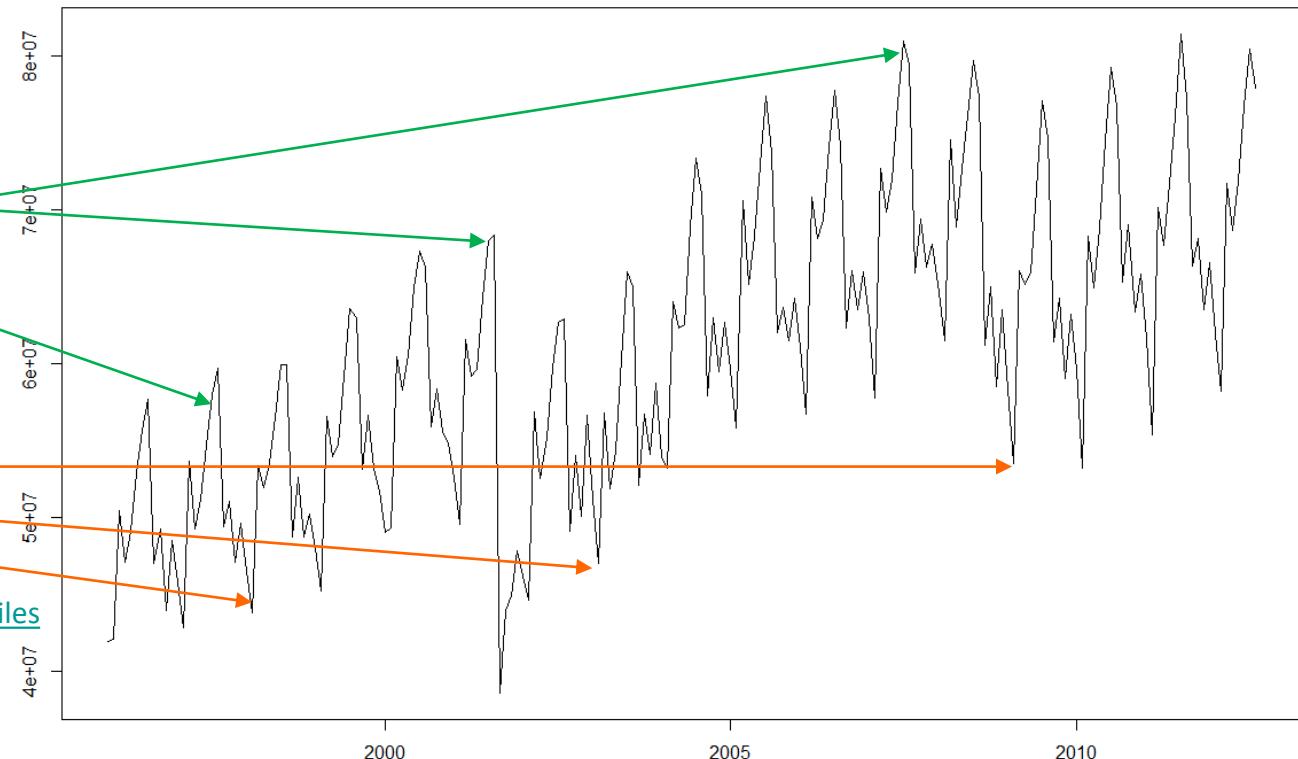
20 years of data
from Jan 1996

Data sources: http://www.bts.gov/xml/air_traffic/src/index.xml and
<https://datamarket.com/data/set/281x/us-air-carrier-traffic-statistics-revenue-passenger-miles>

US Air Carrier Traffic – Revenue Passenger Miles ('000) RPM

R code: `milestimeseries <- ts(miles, frequency = 12, start = c(1996,1))`

	Jan	Feb	Mar	Apr	May	Jun	Jul
1996	41972194	42054796	50443045	47112397	49118248	52880510	55664750
1997	45850623	42838949	53620994	49282817	51191842	54707221	57995025
1998	46514139	43769273	53361926	51968480	53515798	56460422	59939170
1999	47988560	45241211	56555731	53920855	54674958	59213000	63572248
2000	49045412	49306303	60443541	58286680	60533783	64903295	67346377
2001	52634354	49532578	61575055	59151645	59662416	64353323	67965298
2002	46224031	44615129	56897729	52542164	55116060	59745343	62664511
2003	51197175	47040806	56766580	51857453	54335598	60272900	65962215
2004	53979786	53179693	64035864	62340117	62530704	68866398	73335888
2005	59629608	55795165	70595861	65145552	68268899	72952959	77432998
2006	61035027	56729212	70799794	68120559	69352606	74099239	7798621
2007	63016013	57793832	72700241	69836156	71933109	76926452	80988340
2008	64667106	61504426	74575531	68906882	72725750	76162105	79707545
2009	58373786	53506580	66027341	65166300	65868254	71350227	77136799
2010	59651061	53240066	68307090	64953250	68850904	74474550	79304441
2011	61630362	55391206	70158268	67683558	71711448	76057910	81423215
2012	61940180	58243763	71696039	68669228	71887523	76760759	80499353
	Aug	Sep	Oct	Nov	Dec		
1996	57723208	47035464	49263120	43937074	48539606		
1997	59715433	49418190	51058879	47056048	49654209		



Data sources: http://www.bts.gov/xml/air_traffic/src/index.xml and

<https://datamarket.com/data/set/281x/us-air-carrier-traffic-statistics-revenue-passenger-miles>

Last accessed: 31-Mar-2016

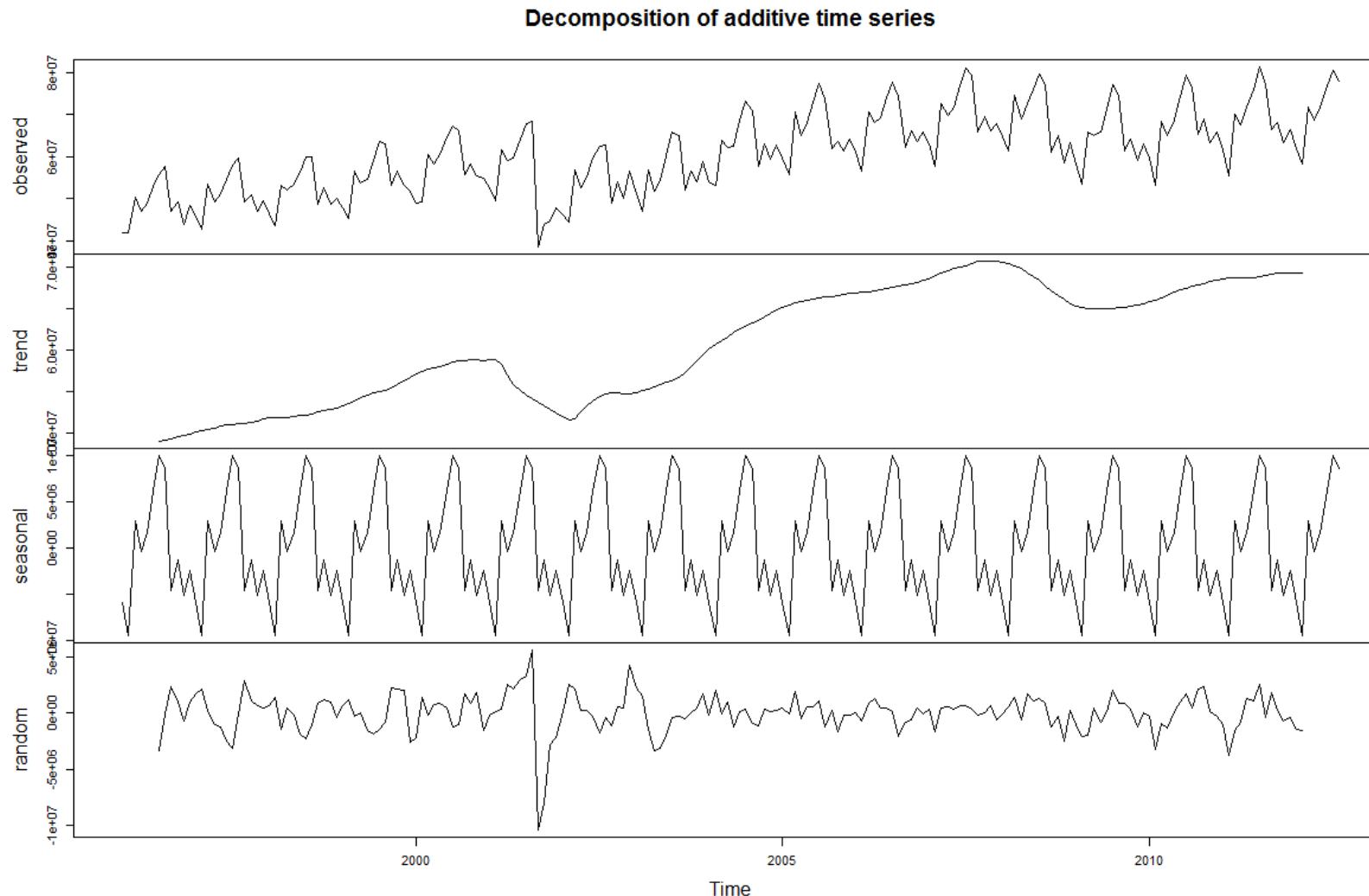
	Aug	Sep	Oct	Nov	Dec
1996	57723208	47035464	49263120	43937074	48539606
1997	59715433	49418190	51058879	47056048	49654209
1998	59927214	48751280	52578217	48734375	50208641
1999	63003663	53131972	56653901	53215500	51746821
2000	66256804	55900504	58373996	55590325	54822970
2001	68377080	38601868	43964788	44915764	47836501
2002	62944816	49096035	54019748	50106814	56656594
2003	64989766	52121480	56724551	54128776	58739845
2004	70961522	57881042	63021142	59453943	62680310



Decomposing Time Series into the 3 Components –

Revenue Passenger Miles (RPM)

R code: `decompose(milestimeseries, type = "additive")`





Time Series

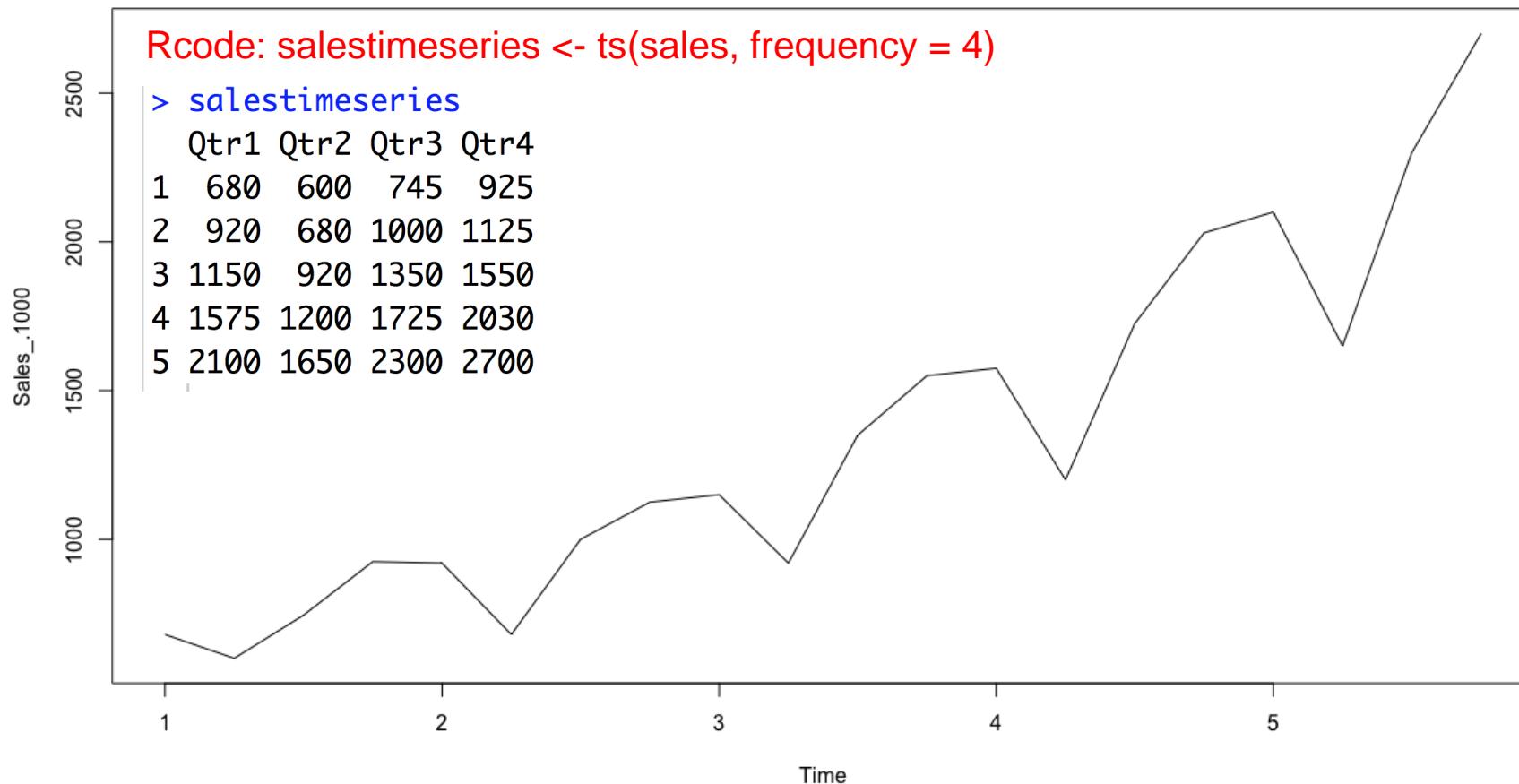
CURVE FITTING / REGRESSION ON TIME METHODS

Regression on Time

Use when trend is the most pronounced

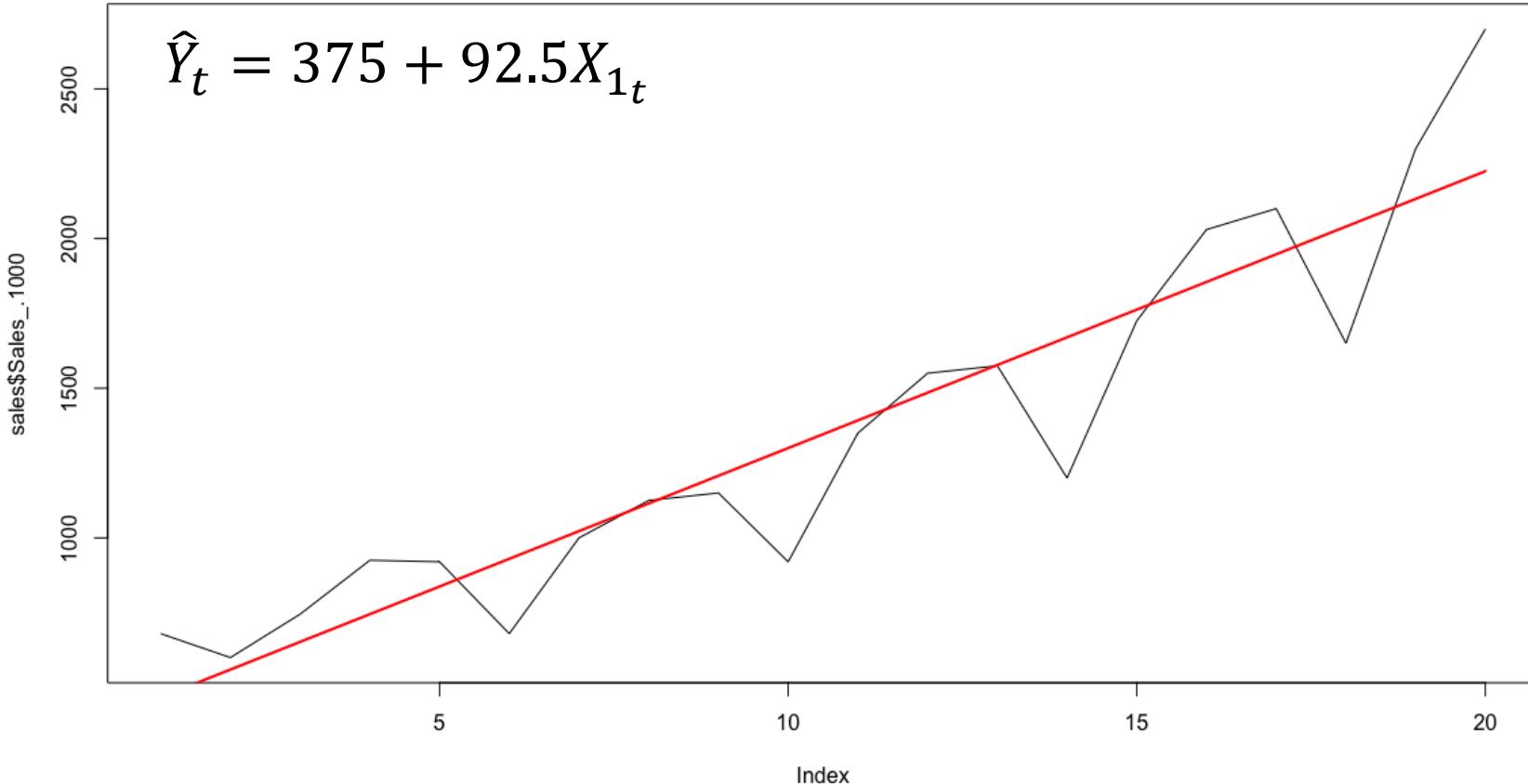


Quarter	Sales_\$1000
1	680
2	600
3	745
4	925
5	920
6	680
7	1000
8	1125
9	1150
10	920
11	1350
12	1550
13	1575
14	1200
15	1725
16	2030
17	2100
18	1650
19	2300
20	2700



Regression Analysis

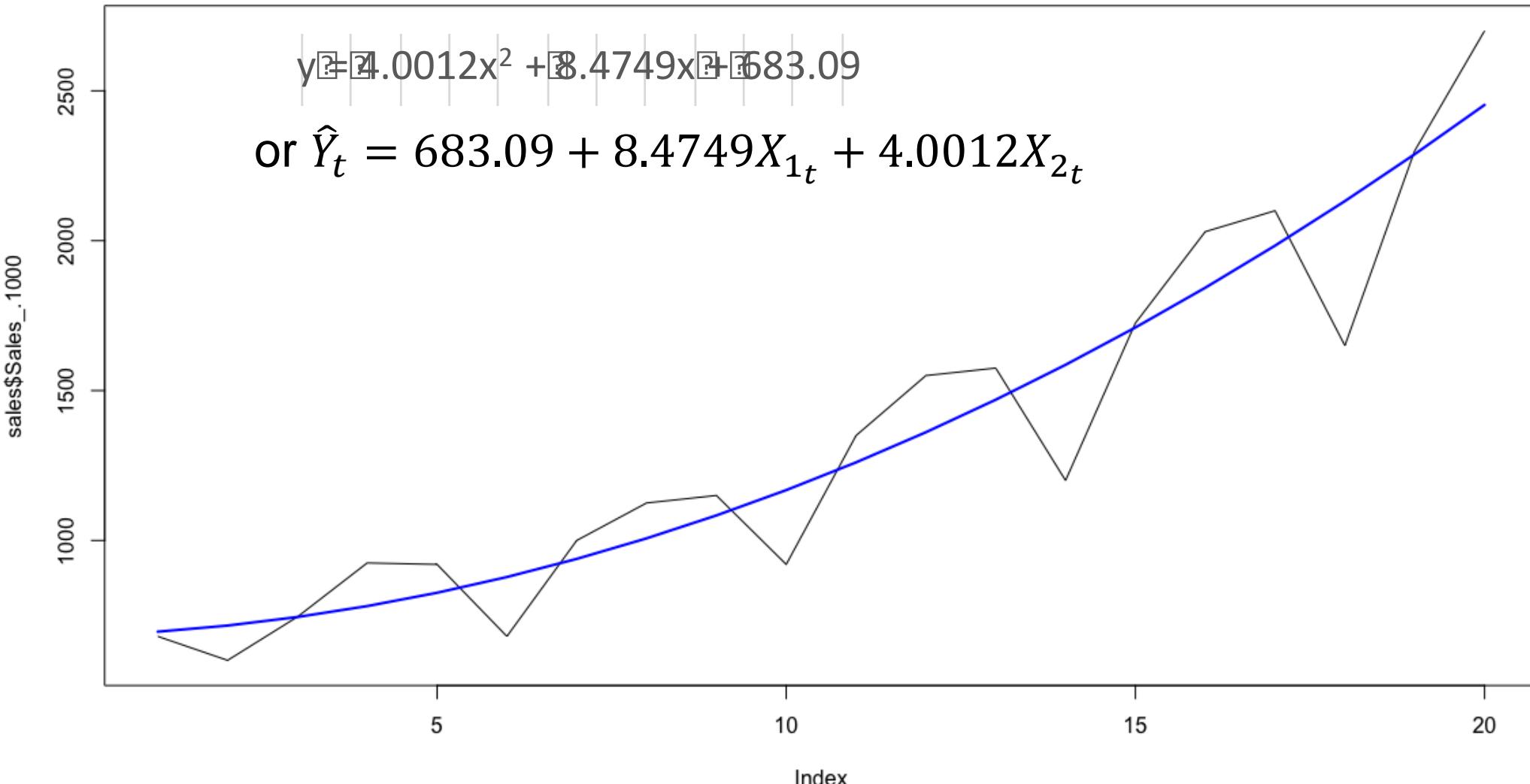
```
Rcode: sales <- data.frame(sales)
sales$time <- seq(1:20)
saleslm1 <- lm(sales$Sales_.1000 ~ sales$time)
```



	Sales_.1000	time
1	680	1
2	600	2
3	745	3
4	925	4
5	920	5
6	680	6
7	1000	7
8	1125	8
9	1150	9
10	920	10
11	1350	11
12	1550	12
13	1575	13
14	1200	14
15	1725	15
16	2030	16
17	2100	17
18	1650	18
19	2300	19
20	2700	20

Quadratic Trend

Rcode: saleslm2 <- lm(sales\$Sales_.1000 ~ poly(sales\$time, 2, raw=TRUE))



Incorporating Seasonality – Dummy Variable Approach

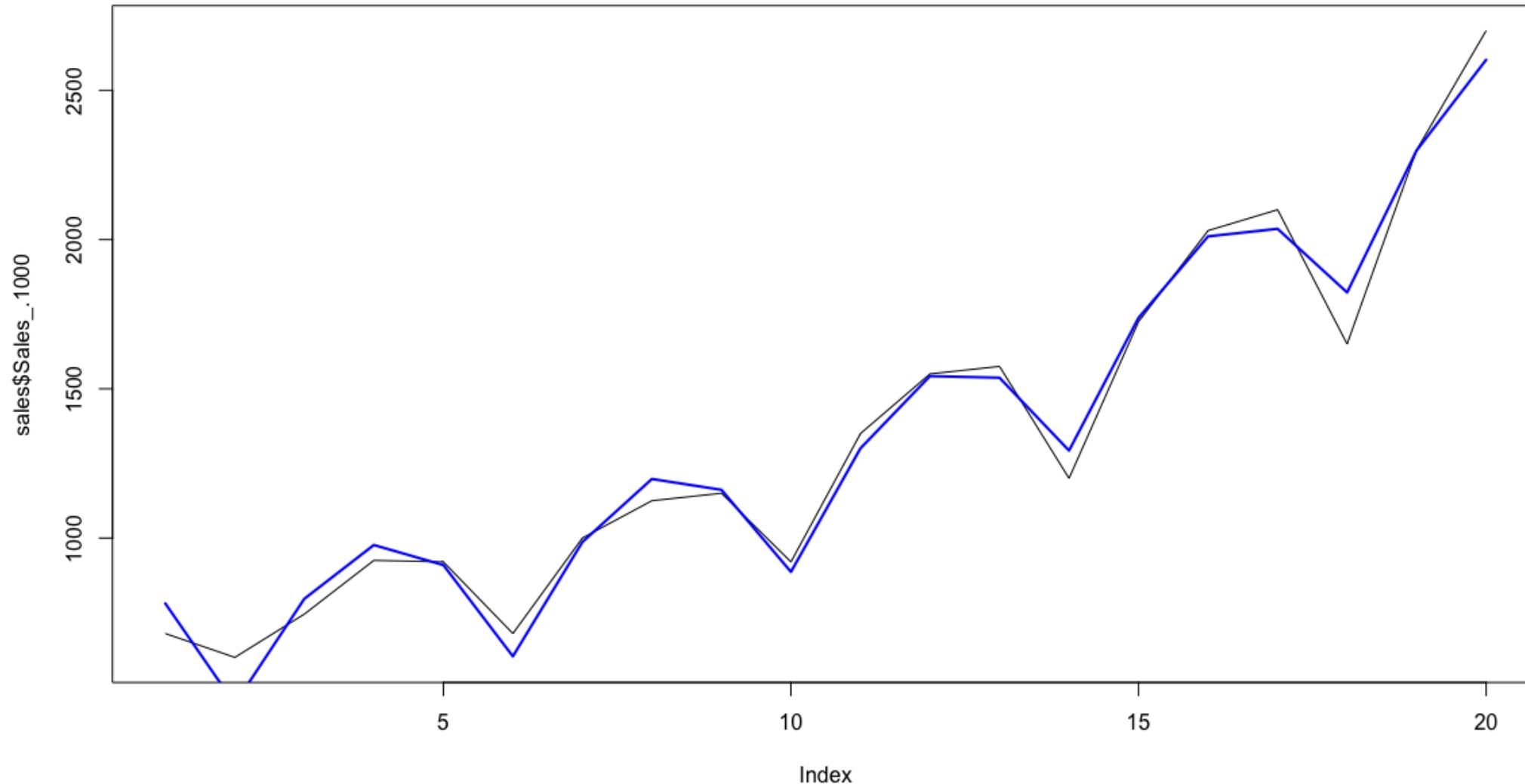
Quarter	Value of		
	X_{3t}	X_{4t}	X_{5t}
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	0

$$\hat{Y}_t = \beta_0 + \beta_1 X_{1t} + \beta_2 X_{2t} + \beta_3 X_{3t} + \beta_4 X_{4t} + \beta_5 X_{5t} + \varepsilon_t$$

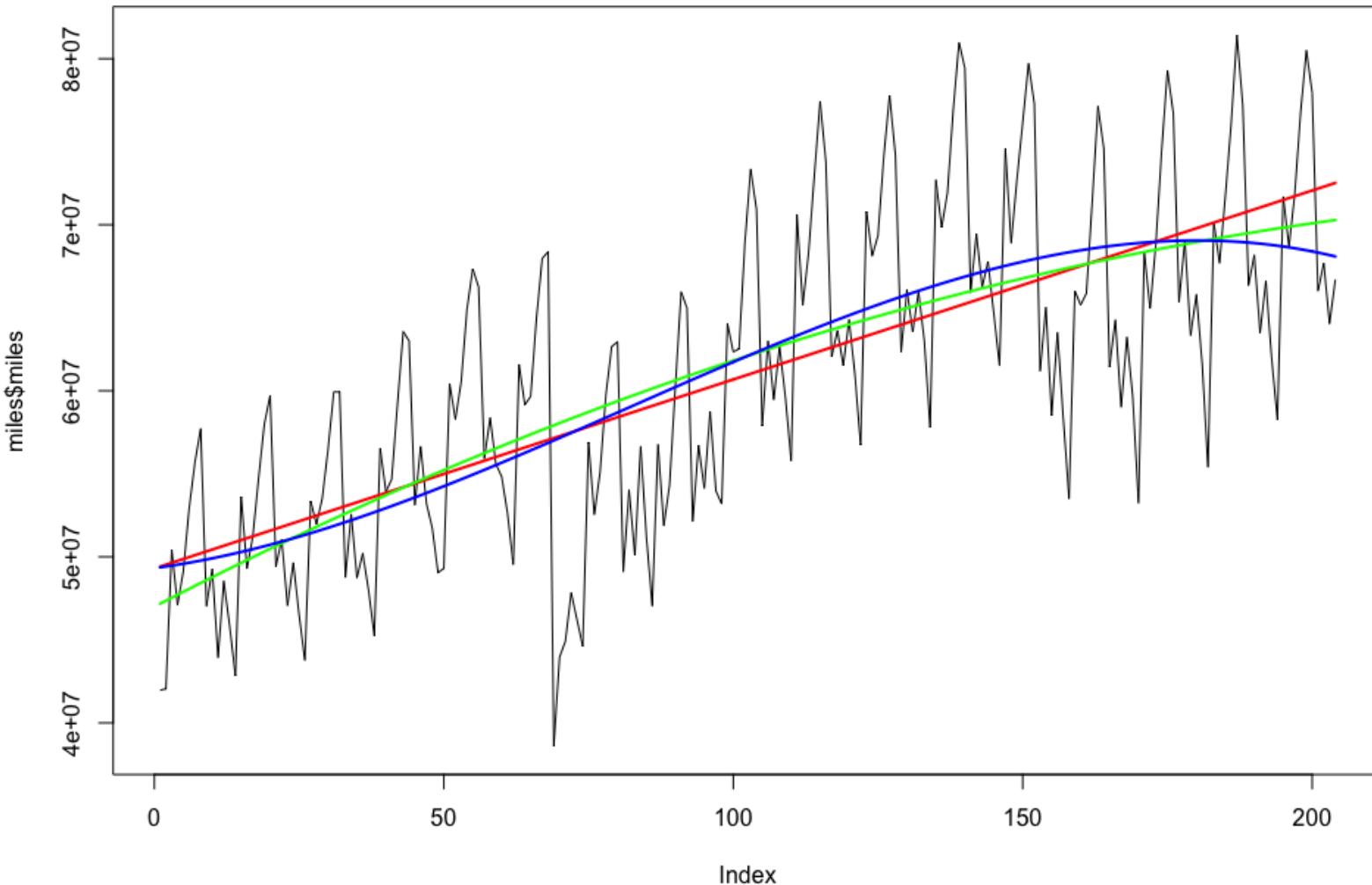
```
Rcode: sales$seasonal <- as.factor(rep(c(1:4),5))
saleslm2s <- lm(sales$Sales_.1000 ~ poly(time, 2, raw=TRUE)+seasonal,
                 data=sales)
```

> sales	Sales_.1000	time	seasonal
1	680	1	1
2	600	2	2
3	745	3	3
4	925	4	4
5	920	5	1
6	680	6	2
7	1000	7	3
8	1125	8	4
9	1150	9	1
10	920	10	2
11	1350	11	3
12	1550	12	4
13	1575	13	1
14	1200	14	2
15	1725	15	3
16	2030	16	4
17	2100	17	1
18	1650	18	2
19	2300	19	3
20	2700	20	4

Seasonal Regression Models

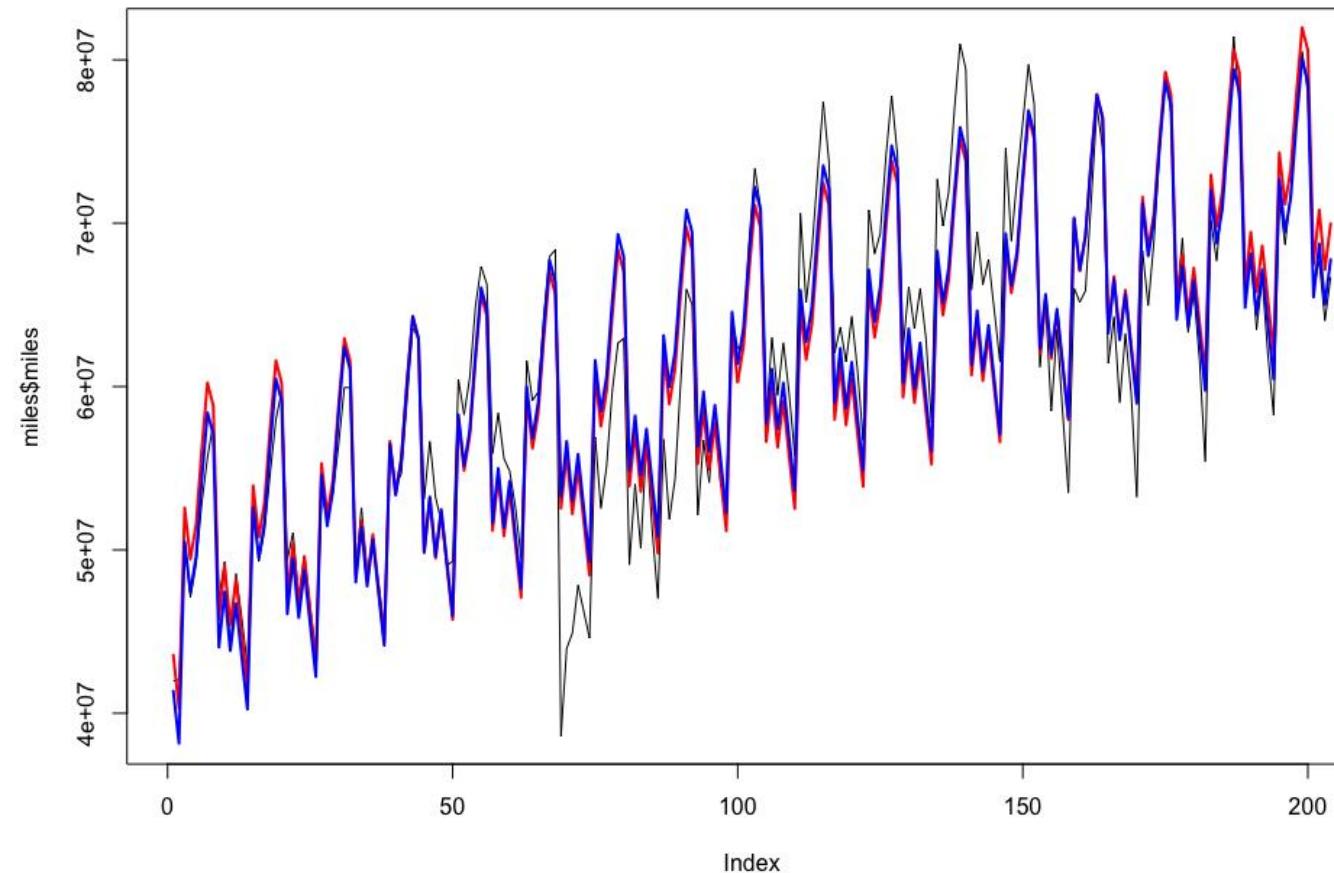


Seasonal Regression Models - RPM



	miles	time	var3
1	41972194	1	
2	42054796	2	
3	50443045	3	
4	47112397	4	
5	49118248	5	
6	52880510	6	
7	55664750	7	
8	57723208	8	
9	47035464	9	
10	49263120	10	
11	43937074	11	
12	48539606	12	
13	45850623	13	
14	42838949	14	
15	53620994	15	

Seasonal Regression Models - RPM



	miles	time	seasonal
1	41972194	1	1
2	42054796	2	2
3	50443045	3	3
4	47112397	4	4
5	49118248	5	5
6	52880510	6	6
7	55664750	7	7
8	57723208	8	8
9	47035464	9	9
10	49263120	10	10
11	43937074	11	11
12	48539606	12	12
13	45850623	13	1
14	42838949	14	2
15	53620994	15	3

Example

Year	Quarter	Time variable (this is created)	Revenues (in \$M)
2016	I	1	10.2
	II	2	12.4
	III	3	14.8
	IV	4	15
2017	I	5	11.2
	II	6	14.3
	III	7	18.4
	IV	8	18

```
Call:  
lm(formula = y ~ x)
```

```
Residuals:
```

Min	1Q	Median	3Q	Max
-3.5595	-0.9384	0.4405	1.3265	1.9286

```
Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	10.0393	1.5531	6.464	0.00065 ***
x	0.9440	0.3076	3.069	0.02196 *

```
---
```

```
Signif. codes: 0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.993 on 6 degrees of freedom
```

```
Multiple R-squared: 0.6109, Adjusted R-squared: 0.5461
```

```
F-statistic: 9.422 on 1 and 6 DF, p-value: 0.02196
```

What is the Regression equation?

$$y = 10.0393 + 0.9440x$$

Incorporating Seasonality – Another Approach

- Take the trend prediction and actual value.
- Depending on additive or multiplicative model, compute the deviation and map it as seasonality effect for each prediction.
- Take averages of the seasonality value. Use this to make future predictions.



Seasonality: Multiplicative

Time	Observed values TSI* (assuming no impact of cyclicality)	Predicted values (per the regression) T*	SI* = TSI/T
1	10.2	10.983	0.929
2	12.4	11.927	1.040
3	14.8	12.871	1.150
4	15.0	13.815	1.086
5	11.2	14.759	0.759
6	14.3	15.703	0.911
7	18.4	16.647	1.105
8	18.0	17.591	1.023

* T: Trend; S: Seasonal; I: Irregular

Quarterly Seasonality

Time	Observed values TSI* (assuming no impact of cyclicalty)	Predicted values (per the regression) T*	SI* = TSI/T	Time	Average seasonality factor
				Q1	$0.844 \left(= \frac{0.929+0.759}{2}\right)$
1	10.2	10.983	0.929		
2	12.4	11.927	1.040		
3	14.8	12.871	1.150		
4	15.0	13.815	1.086		
5	11.2	14.759	0.759	Q3	1.127
6	14.3	15.703	0.911		
7	18.4	16.647	1.105		
8	18.0	17.591	1.023	Q4	1.054

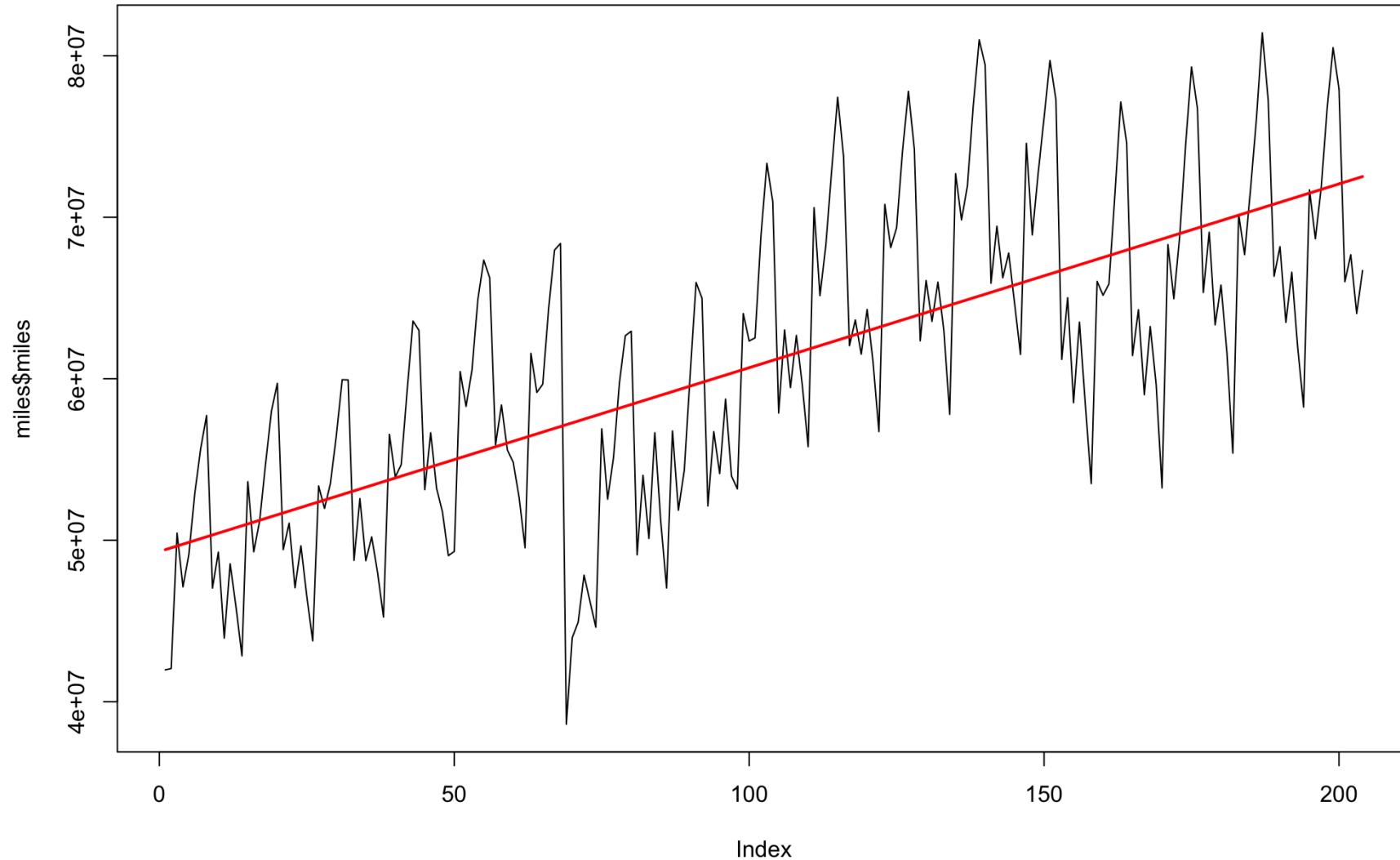
Computations

$$y = 10.0393 + 0.9440x$$

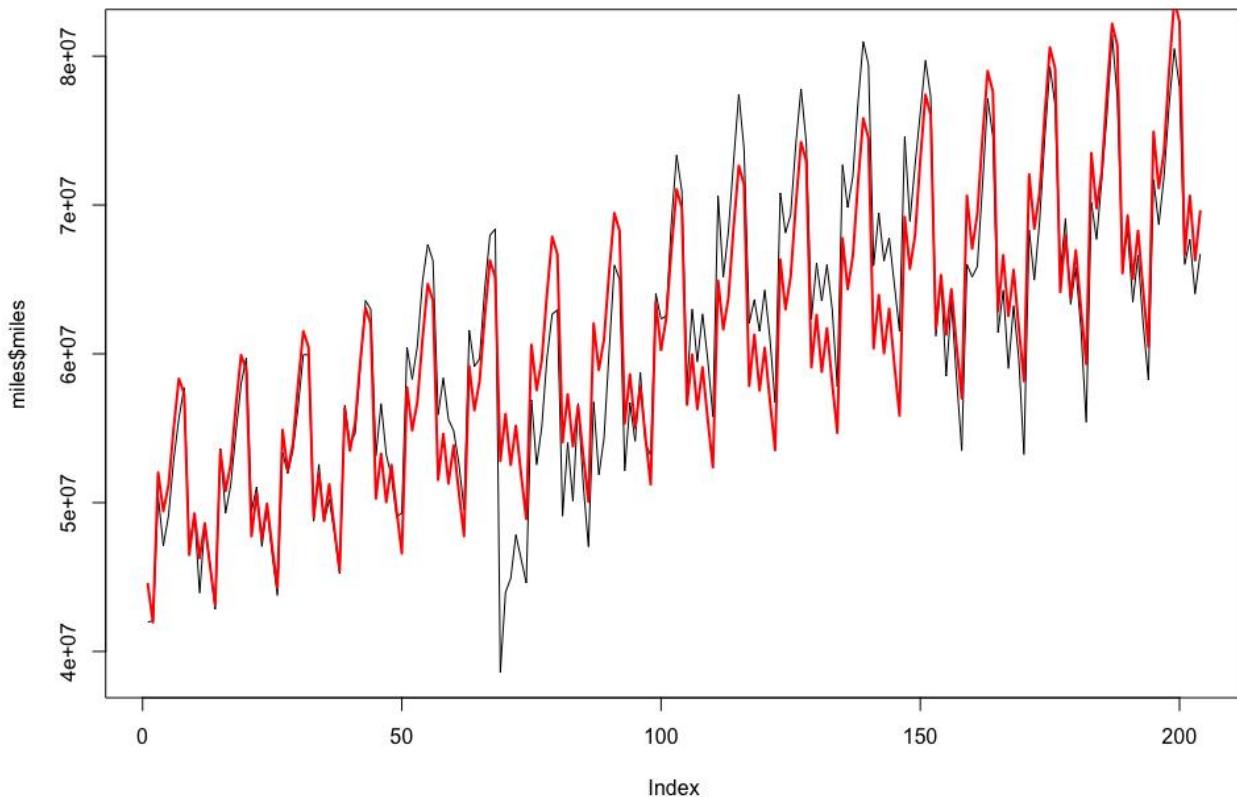
- Trend $Y_9 = 10.039 + 0.944(9) = 18.535$
- Corrected for seasonality and randomness: $18.535 * 0.844 = 15.643$



Regression on Time - RPM

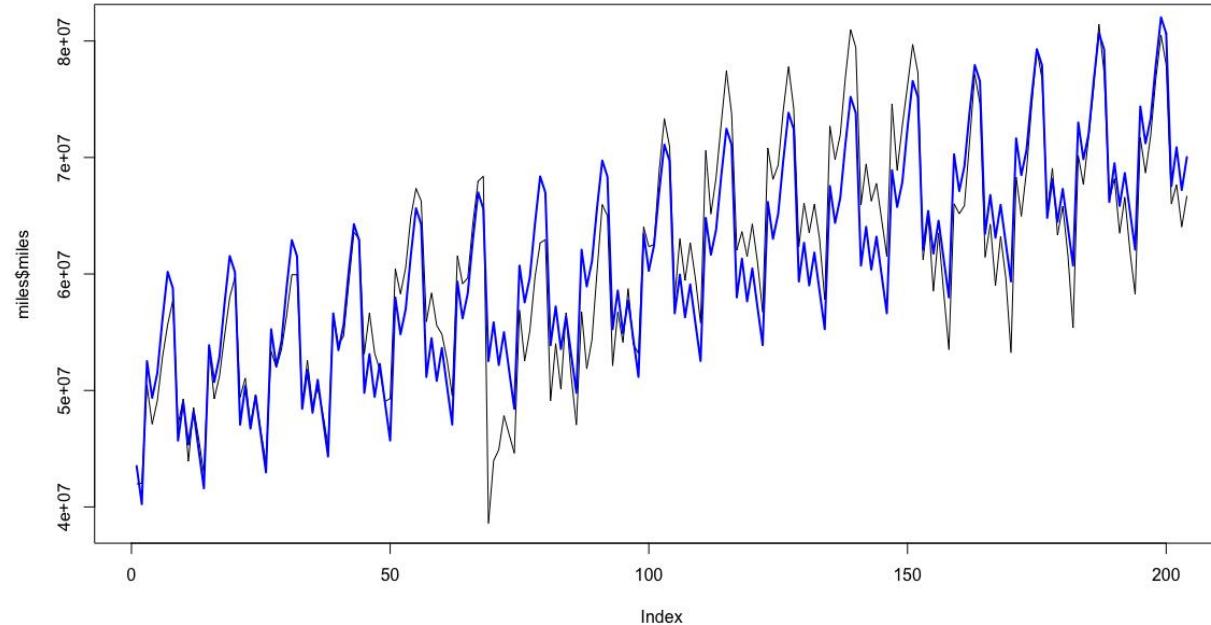


Seasonality: Multiplicative



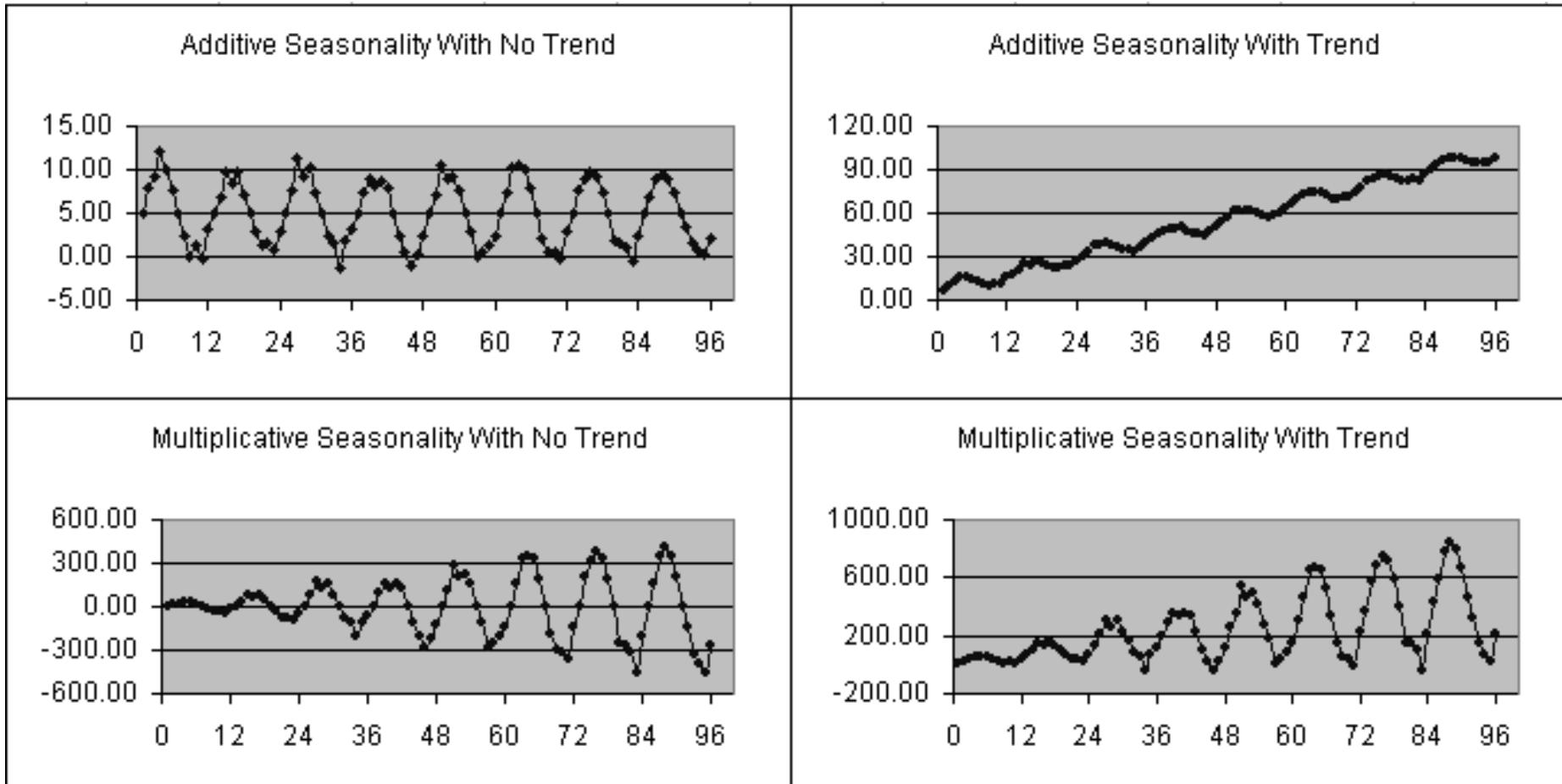
	miles	time	seasonal	mae
1	41972194	1	1	0.849386
2	42054796	2	2	0.8491019
3	50443045	3	3	1.016129
4	47112397	4	4	0.946865
5	49118248	5	5	0.9849257
6	52880510	6	6	1.057953
7	55664750	7	7	1.111125
8	57723208	8	8	1.149602
9	47035464	9	9	0.9346292
10	49263120	10	10	0.9766855
11	43937074	11	11	0.8691307
12	48539606	12	12	0.9580177
13	45850623	13	1	0.9029174
14	42838949	14	2	0.8417232
15	53620994	15	3	1.051224

Seasonality: Additive



	miles	time	seasonal	mae
1	41972194	1	1	-7442550
2	42054796	2	2	-7473763
3	50443045	3	3	800670.5
4	47112397	4	4	-2643793
5	49118248	5	5	-751757.1
6	52880510	6	6	2896690
7	55664750	7	7	5567114
8	57723208	8	8	7511757
9	47035464	9	9	-3289802
10	49263120	10	10	-1175962
11	43937074	11	11	-6615823
12	48539606	12	12	-2127106
13	45850623	13	1	-4929905
14	42838949	14	2	-8055394
15	53620994	15	3	2612836
.

Additive or Multiplicative



Issues with Regressing on Time

If there is no trend or if seasonality and fluctuations are more important than trend, then the coefficients behave weirdly





Advanced Time Series Methods

HOLT-WINTERS



Holt-Winters – Important Concepts

SMOOTHING

Simple Moving Averages

	Number of products sold	SMA (K=2)	SMA (K=3)
1	15		
2	20	17.5	
3	16	18	17
4	13	14.5	16.333333
5	18	15.5	15.666667
6	14	16	15
7	15	14.5	15.666667
8	17	16	15.333333
9	20	18.5	17.333333
10	20	20	19

Weighted Moving Averages

	Number of products sold	WMA (K=2)	WMA (K=3)		Number of products sold	WMA (K=2)	WMA (K=3)
1					1		
2	15				2	15	
3	20	$=(A3*2+A2*1)/3$			3	20	18.3333333
4	16	$=(A4*2+A3*1)/3$	$=(A4*3+A3*2+A2*1)/6$		4	16	17.3333333
5	13	$=(A5*2+A4*1)/3$	$=(A5*3+A4*2+A3*1)/6$		5	13	14
6	18	$=(A6*2+A5*1)/3$	$=(A6*3+A5*2+A4*1)/6$		6	18	16.3333333
7	14	$=(A7*2+A6*1)/3$	$=(A7*3+A6*2+A5*1)/6$		7	14	15.3333333
8	15	$=(A8*2+A7*1)/3$	$=(A8*3+A7*2+A6*1)/6$		8	15	14.6666667
9	17	$=(A9*2+A8*1)/3$	$=(A9*3+A8*2+A7*1)/6$		9	17	16.3333333
10	20	$=(A10*2+A9*1)/3$	$=(A10*3+A9*2+A8*1)/6$		10	20	19
11	20	$=(A11*2+A10*1)/3$	$=(A11*3+A10*2+A9*1)/6$		11	20	20
12	18	$=(A12*2+A11*1)/3$	$=(A12*3+A11*2+A10*1)/6$		12	18	18.6666667
13	20	$=(A13*2+A12*1)/3$	$=(A13*3+A12*2+A11*1)/6$		13	20	19.3333333

Exponential Weighted Moving Averages or Exponential Smoothing

Averaging over long periods dampens fluctuations, removing not only the noise but also trend and seasonality.

Moving averages over short recent periods maintain trend and seasonality but determining an optimum number for periods is tricky, even when using metrics like MAE. If averaged over too few periods, irregularities continue to remain and if averaged over long periods, dampening again becomes a problem.

Exponential smoothing **retains all older periods** while giving a greater weight to more recent periods (hence not a MOVING average).

Caution: It doesn't make any one method superior for all situations.



Stationary Model: Exponential Weighted Moving Averages or Exponential Smoothing

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

Above equation indicates that the predicted value for time period $t+1$ (\hat{Y}_{t+1}) is equal to the predicted value for the previous period (\hat{Y}_t) plus an adjustment for the error made in predicting the previous period's value ($\alpha(Y_t - \hat{Y}_t)$).

The parameter α can assume any value between 0 and 1 ($0 \leq \alpha \leq 1$).

Exponential Smoothing in Other Ways

$\hat{Y}_{t+1} = \hat{Y}_t + \alpha(Y_t - \hat{Y}_t)$ can be rewritten variously as

$$\begin{aligned}\boxed{\hat{Y}_{t+1}} &= \alpha Y_t + (1 - \alpha) \hat{Y}_t \\ &= Y_t - (1 - \alpha)(Y_t - \hat{Y}_t) \\ \rightarrow &= \alpha Y_t + \alpha(1 - \alpha)Y_{t-1} + \alpha(1 - \alpha)^2Y_{t-2} + \cdots + \alpha(1 - \alpha)^nY_{t-n} + \cdots\end{aligned}$$

Note

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

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

So

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

Then

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

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

Various Ways of Understanding Exponential Smoothing

- Forecast
 - Interpolation between previous *forecast* and previous *observation*
$$= \alpha Y_t + (1 - \alpha) \hat{Y}_t$$
 - Previous *forecast* plus fraction of previous error
$$= \hat{Y}_t + \alpha(Y_t - \hat{Y}_t)$$
 - Previous *observation* minus fraction 1- of previous error
$$= Y_t - (1 - \alpha)(Y_t - \hat{Y}_t)$$
 - *Exponentially weighted (i.e., discounted) moving average*
$$= \alpha Y_t + \alpha(1 - \alpha)Y_{t-1} + \alpha(1 - \alpha)^2 Y_{t-2} + \cdots + \alpha(1 - \alpha)^n Y_{t-n} + \cdots$$





Holt-Winters **MODEL BUILDING**

Holt-Winters Method

- 3 components – Trend, Seasonality and Level.

Additive Seasonality

$$\hat{y}_t = l_{t-1} + b_{t-1} + s_{t-m}$$

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m}$$

l = level

b = Trend

s = Seasonality

Multiplicative Seasonality

$$\hat{y}_t = (l_{t-1} + b_{t-1})s_{t-m}$$

$$\hat{y}_{t+h|t} = (l_t + hb_t)s_{t+h-m}$$



Holt-Winters Method

- 3 weights – smoothing parameters – are used to update components at each period.

Additive Seasonality

$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$

$$s_t = \gamma(y_t - (l_{t-1} + b_{t-1})) + (1 - \gamma)s_{t-m}$$

The 3 smoothing equations are:

Multiplicative Seasonality

$$l_t = \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$

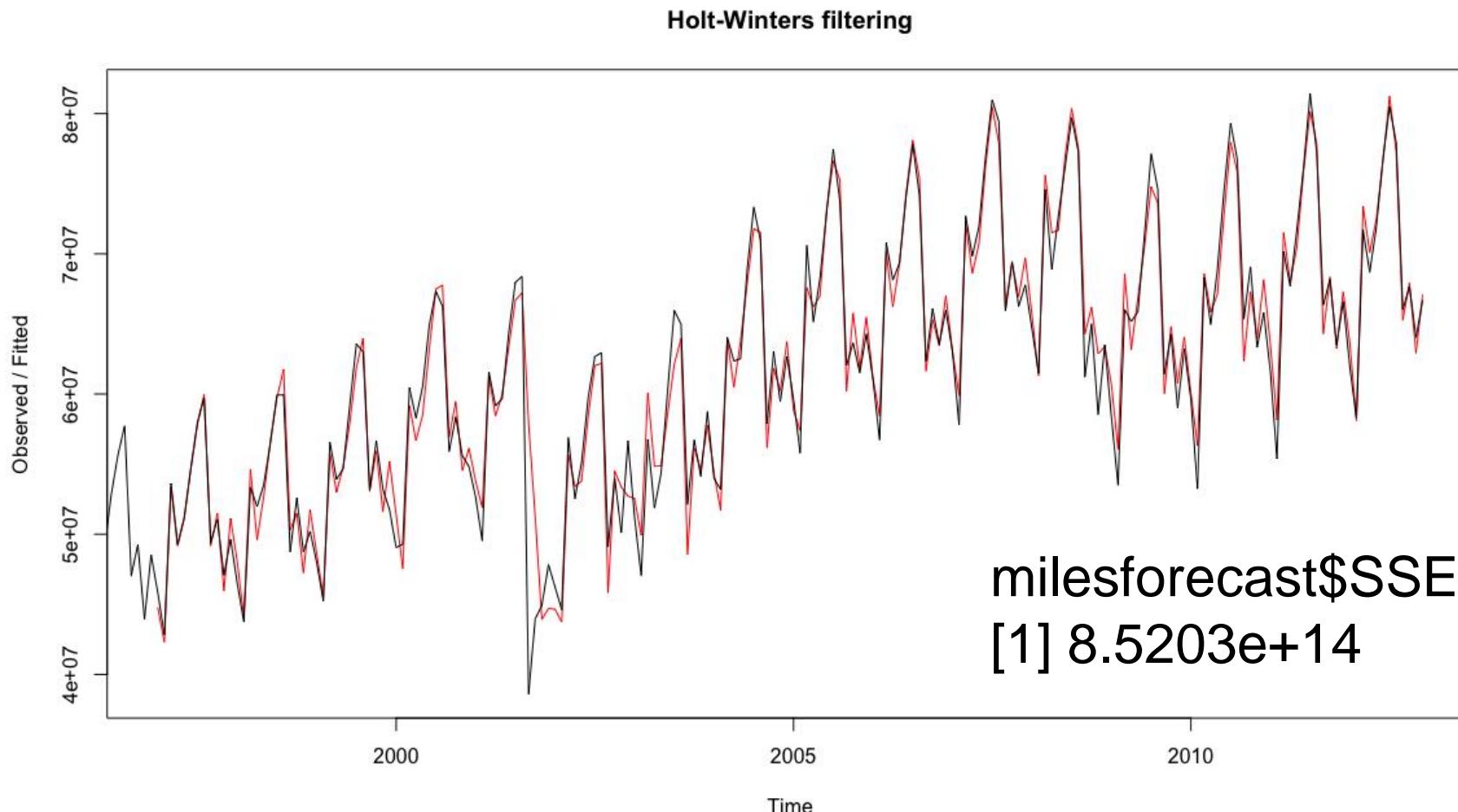
$$s_t = \gamma \frac{y_t}{l_{t-1} + b_{t-1}} + (1 - \gamma)s_{t-m}$$

Recall Exponential Smoothing form: $\hat{Y}_t = \alpha Y_{t-1} + (1 - \alpha)\hat{Y}_{t-1}$



Holt-Winters Method: *RPM*

R code: `HoltWinters(milestimeseries)`



Holt-Winters Method: *RPM*

```
> milesforecast
Holt-Winters exponential smoothing with trend and additive seasonal component.
```

Call:

```
HoltWinters(x = milestimeseries)
```

Smoothing parameters:

```
alpha: 0.5024519
beta : 0
gamma: 0.6698853
```

$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}$$

$$s_t = \gamma(y_t - (l_{t-1} + b_{t-1})) + (1 - \gamma)s_{t-m}$$

Coefficients:

```
[,1]
```

```
a 71153143.7
```

```
b 181508.7
```

```
s1 -8052713.5
```

```
s2 -12314754.8
```

```
s3 2106026.2
```

```
s4 -447843.5
```

```
s5 2758176.7
```

```
s6 7232862.9
```

```
s7 11309888.8
```

```
s8 8022549.6
```

```
s9 -4490660.8
```

```
s10 -2715650.2
```

```
s11 -7315682.9
```

```
s12 -4384446.7
```

```
> milesforecast$fitted
```

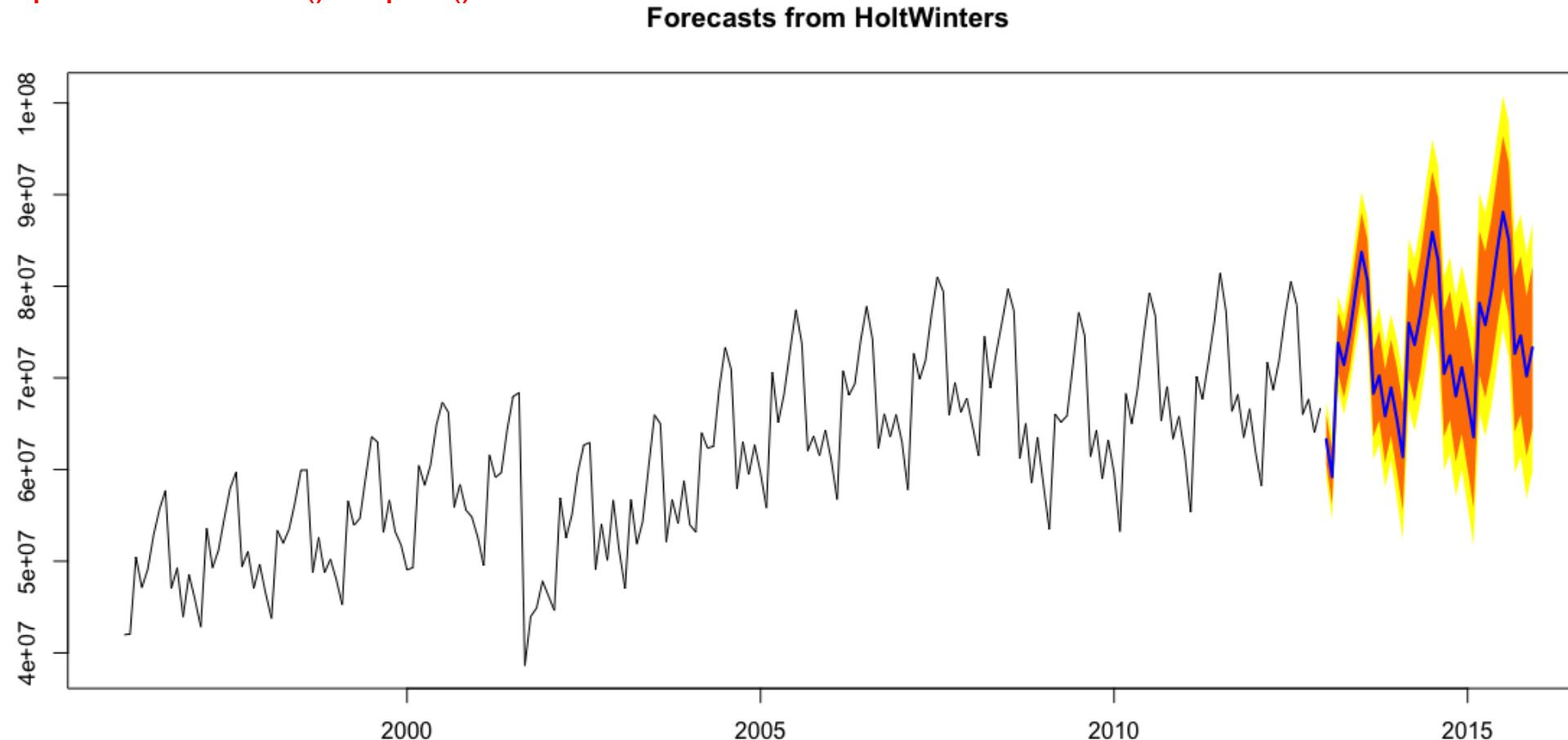
	xhat	level	trend	season
Jan 1997	44780388	48804078	181508.7	-4205198.60
Feb 1997	42307860	49523328	181508.7	-7396976.76

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m}$$

R refers to l as a

Holt-Winters Method: Forecasting

R code: `forecast.HoltWinters(milesforecast, h=36)` or `forecast(milesforecast, h=36)` depending on the version of R being used. Similarly for plot we need to use `plot.holtWinters()` or `plot()`





Advanced Time Series Methods

ARIMA



**ARIMA [AutoRegressive Integrated Moving Average–
Important Concepts]**

ACF-PACF AND STATIONARITY

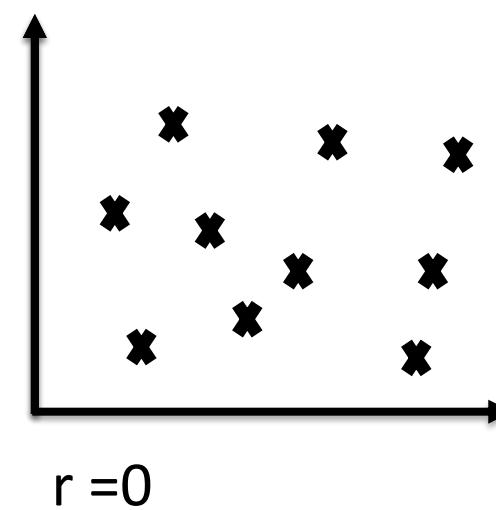
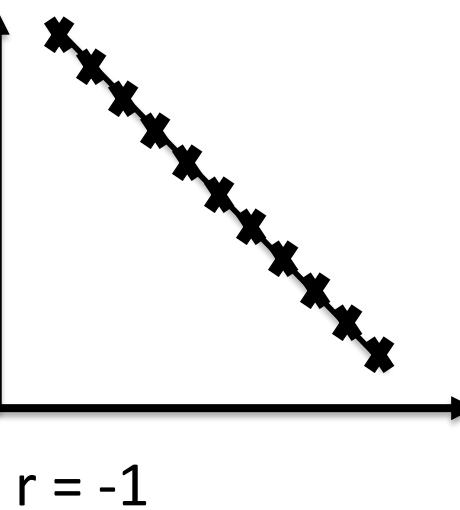
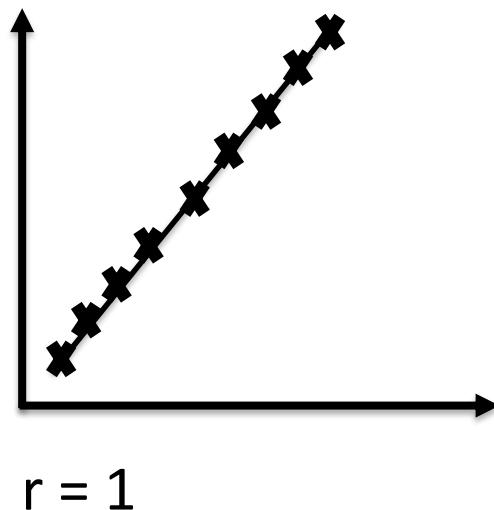
Time Series Descriptive Statistics

- In descriptive statistics covered earlier (central tendencies, measures of variability, distributions, correlations, etc.), the order of observations in the data was of no consequence.
- In time series descriptive statistics, order of observations is of primary importance and so **autocorrelations** play a vital role in identifying the models and their characteristics.



Correlation Coefficient

Correlation coefficient, r , is a number between -1 and 1 and tells us how well a regression line fits the data.



It gives the strength and direction of the relationship between two variables.

Autocorrelation (ACF) and Partial ACF (PACF) [Excel “Correlation Stationarity”]

- ACF: n^{th} lag of ACF is the correlation between a day and n days before that.
- PACF: The same as ACF with all intermediate correlations removed. It is the k_{th} coefficient of the ordinary least squares regression.

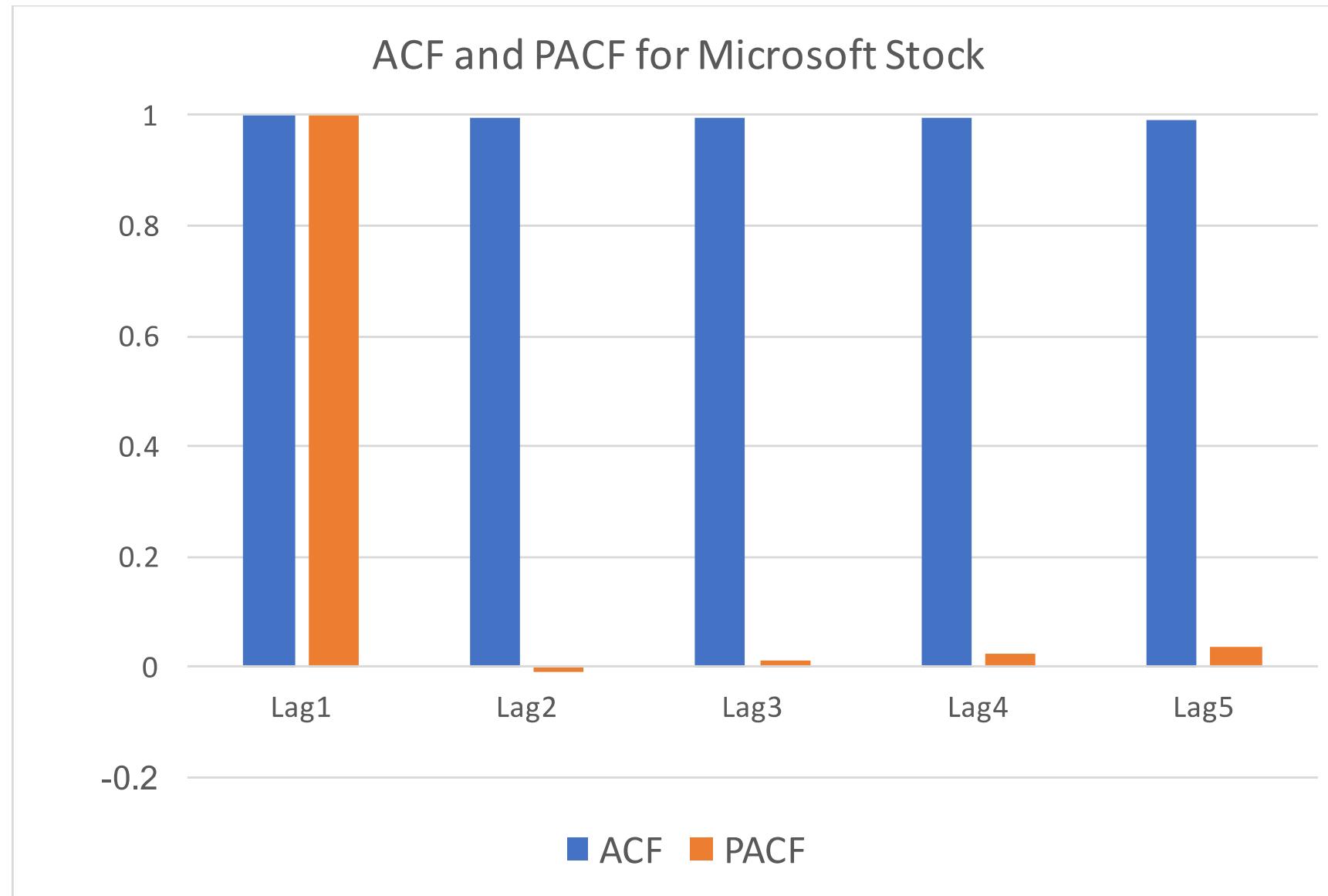
$$[y_t] = \beta_0 + \sum_{i=1}^k \beta_i [y_{t-i}] \text{ where}$$

$[y_t]$ is the input time series, k is the lag order and β_i is the i_{th} coefficient of the linear multiple regression.

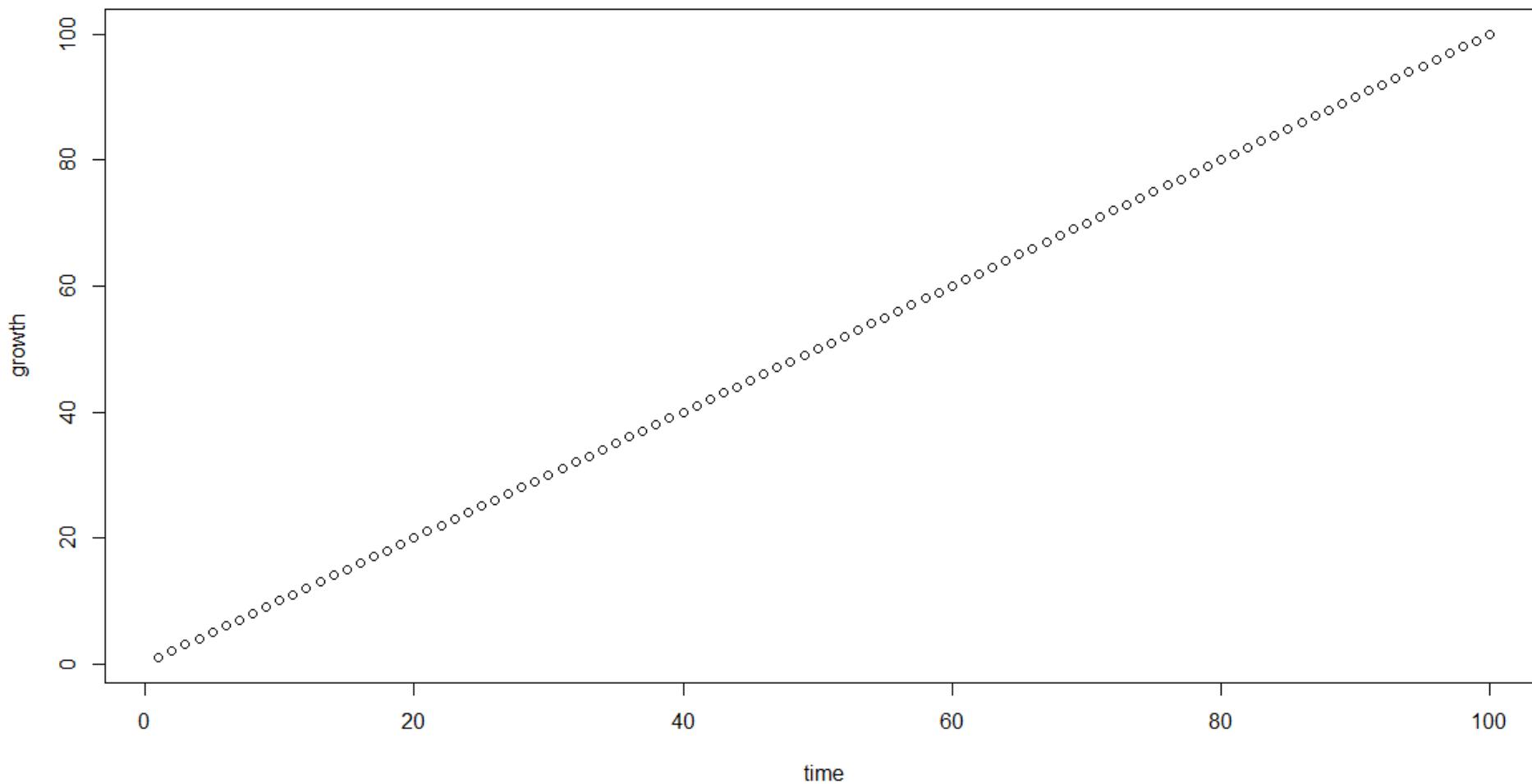
EXCEL ACTIVITY



Autocorrelation (ACF) and Partial ACF (PACF)

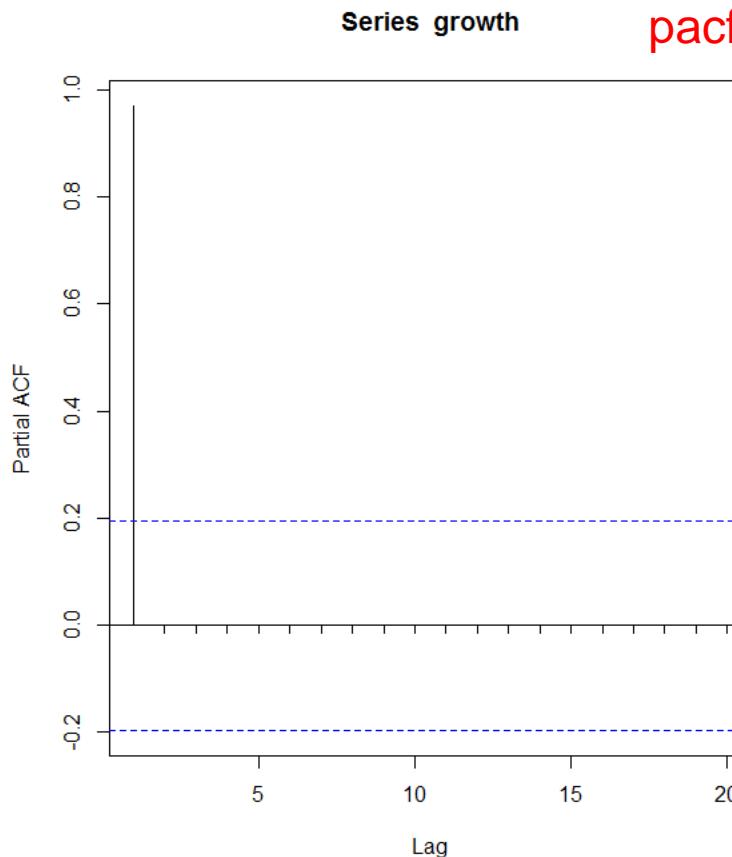
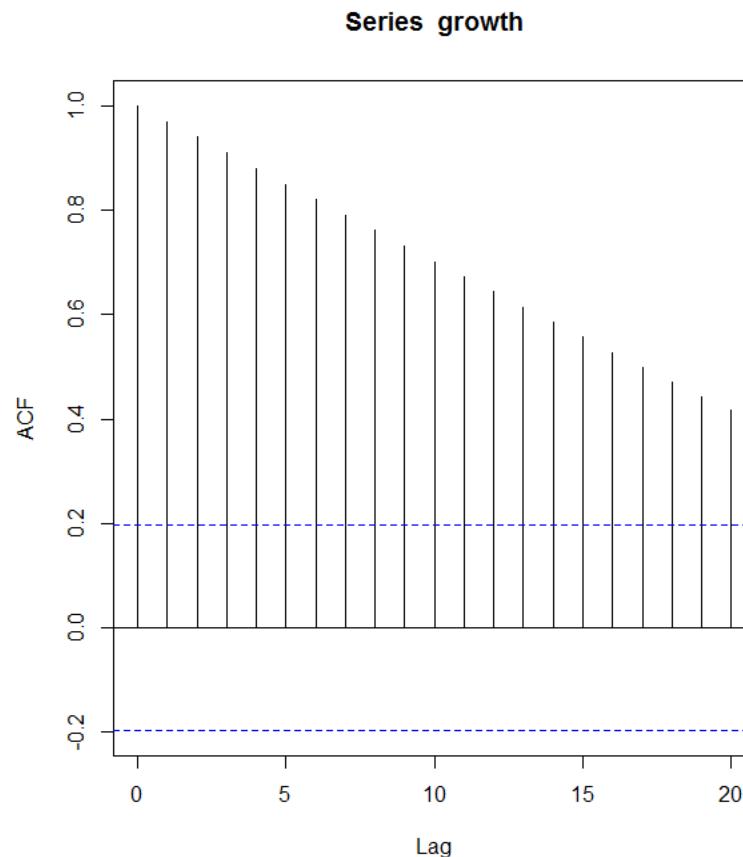


ACF and PACF – Idealized Trend



ACF and PACF – Idealized Trend

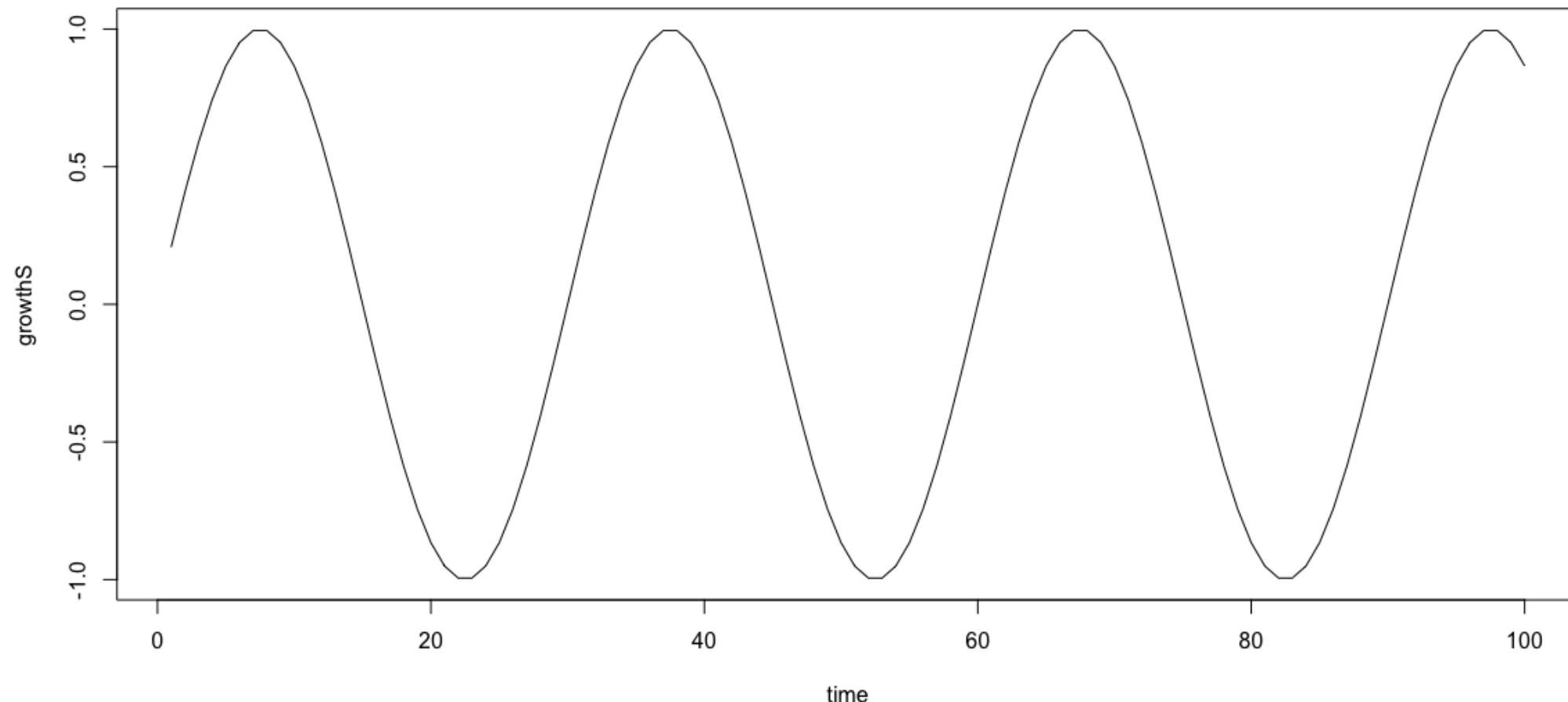
R code: `growthT <- ts(growthT)`
`acf(growthT)`
`pacf(growthT)`



95% CI: $0 \pm \frac{1.96}{\sqrt{n}}$

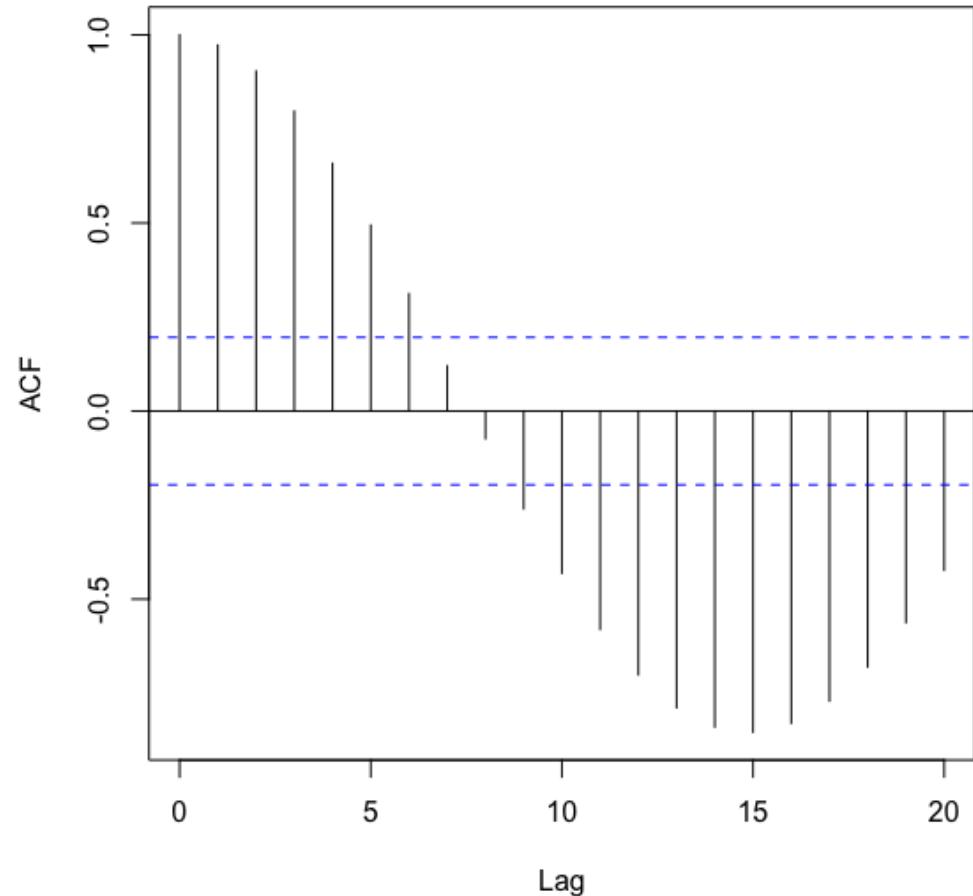
- ACF is a bar chart of correlation coefficients of the time series and its lags.
- PACF is a plot of the partial correlation coefficients of the time series and its lags.

ACF and PACF – Idealized Seasonality

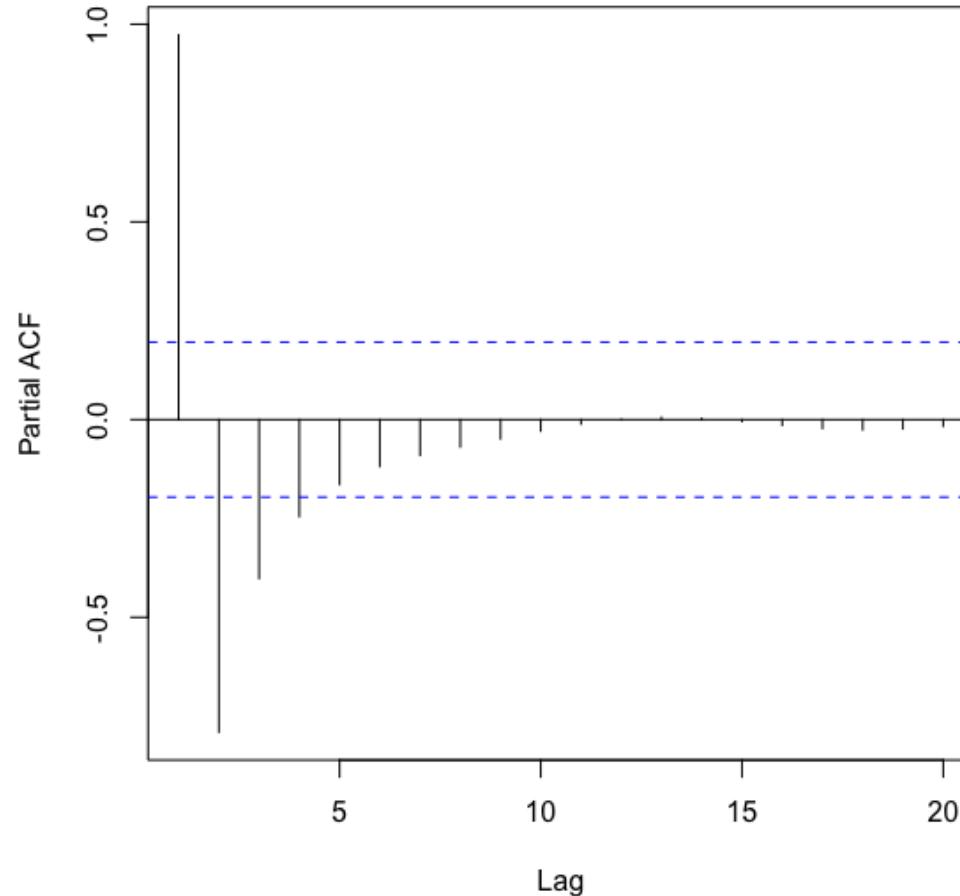


ACF and PACF – Idealized Seasonality

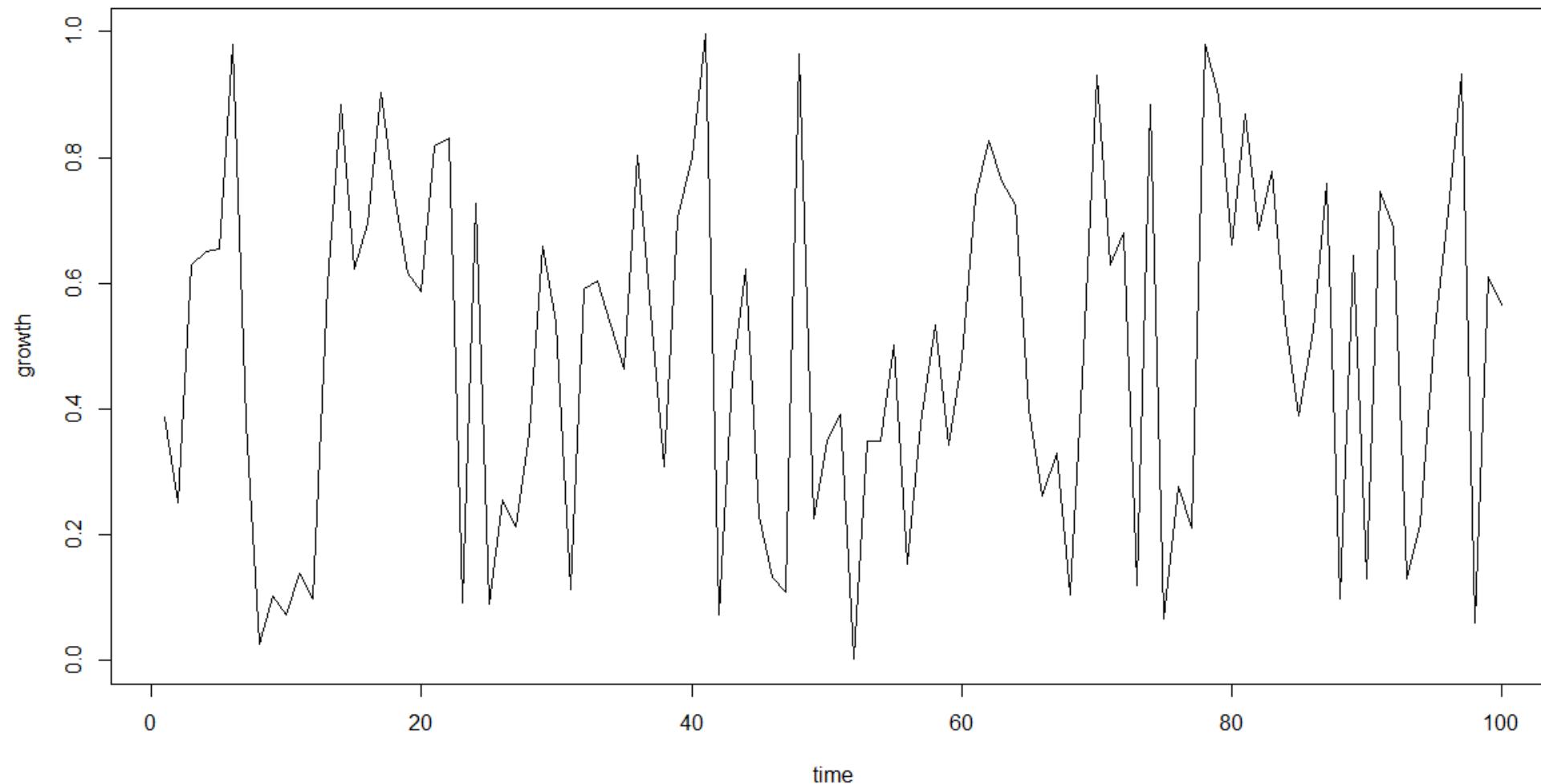
Series growth



Series growth

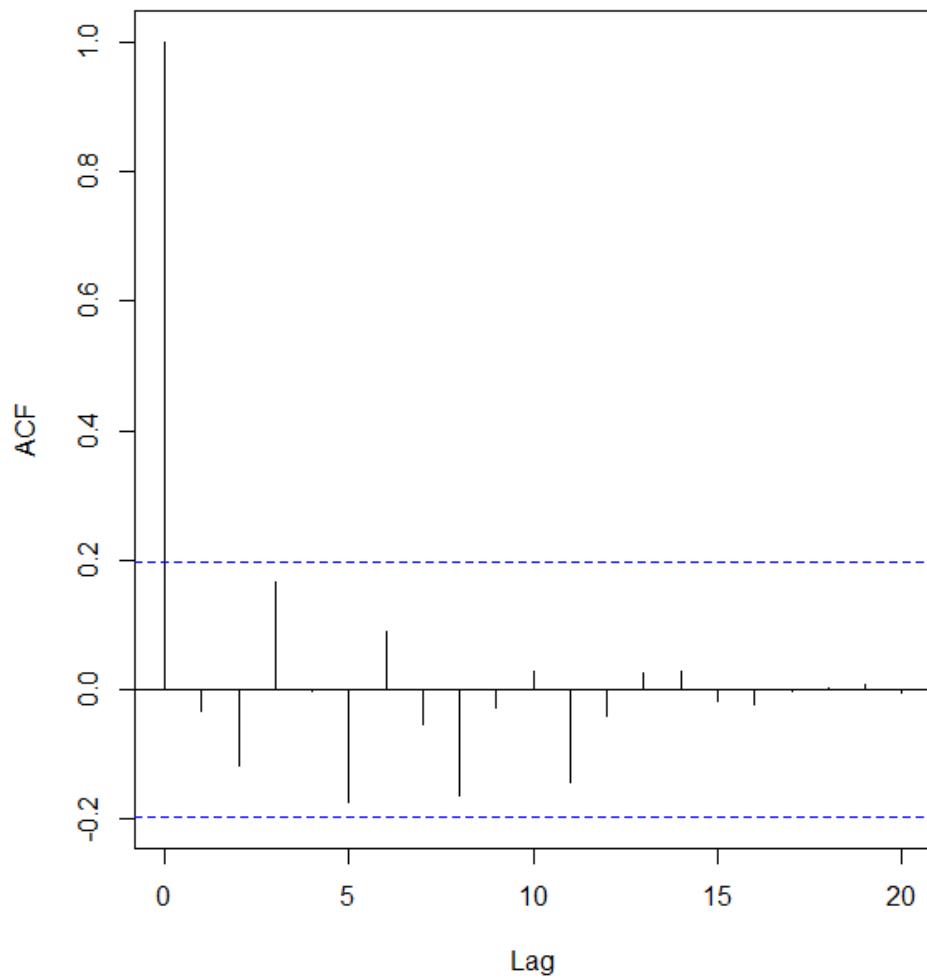


ACF and PACF – Idealized Randomness

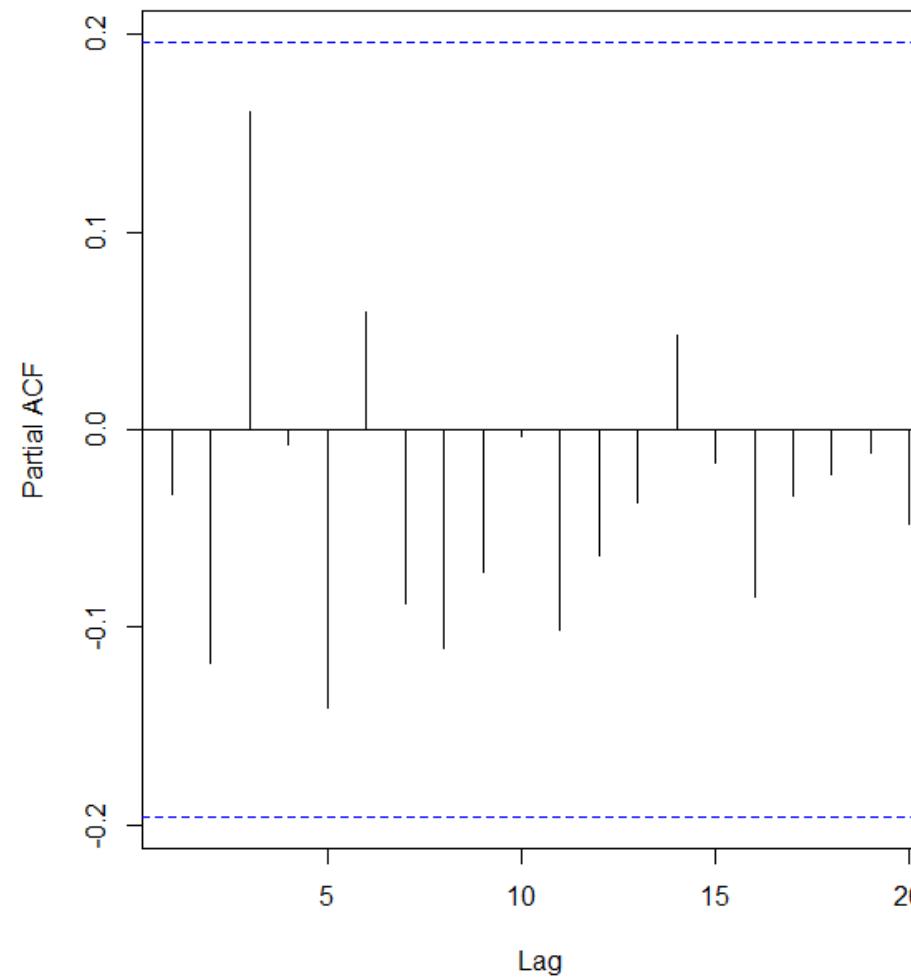


ACF and PACF – Idealized Randomness

Series growth



Series growth

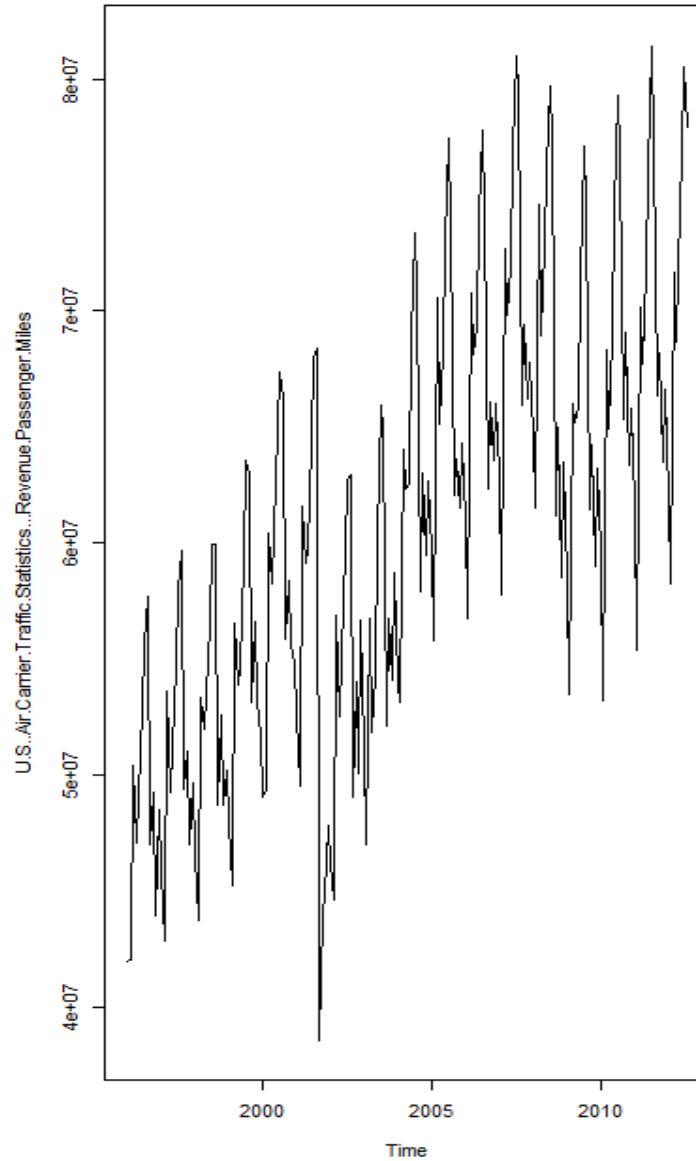


ACF and PACF – Idealized Trend, Seasonality and Randomness

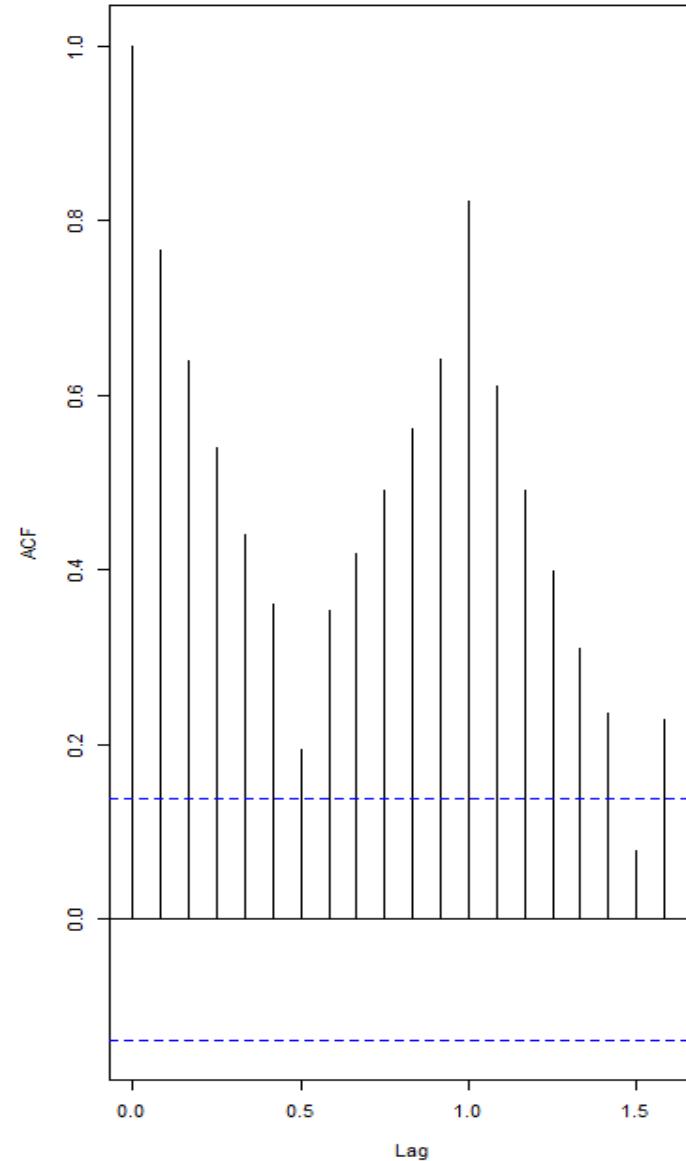
- Ideal Trend: **Decaying ACF** and **Cutoff** after 1 or 2 lags of PACF
- Ideal Seasonality: **Decaying Cyclical**ity in ACF and **Cutoff** after a few lags of PACF with some positive and some negative
- Ideal Random: A spike may or may not be present; even if present, magnitude will be small



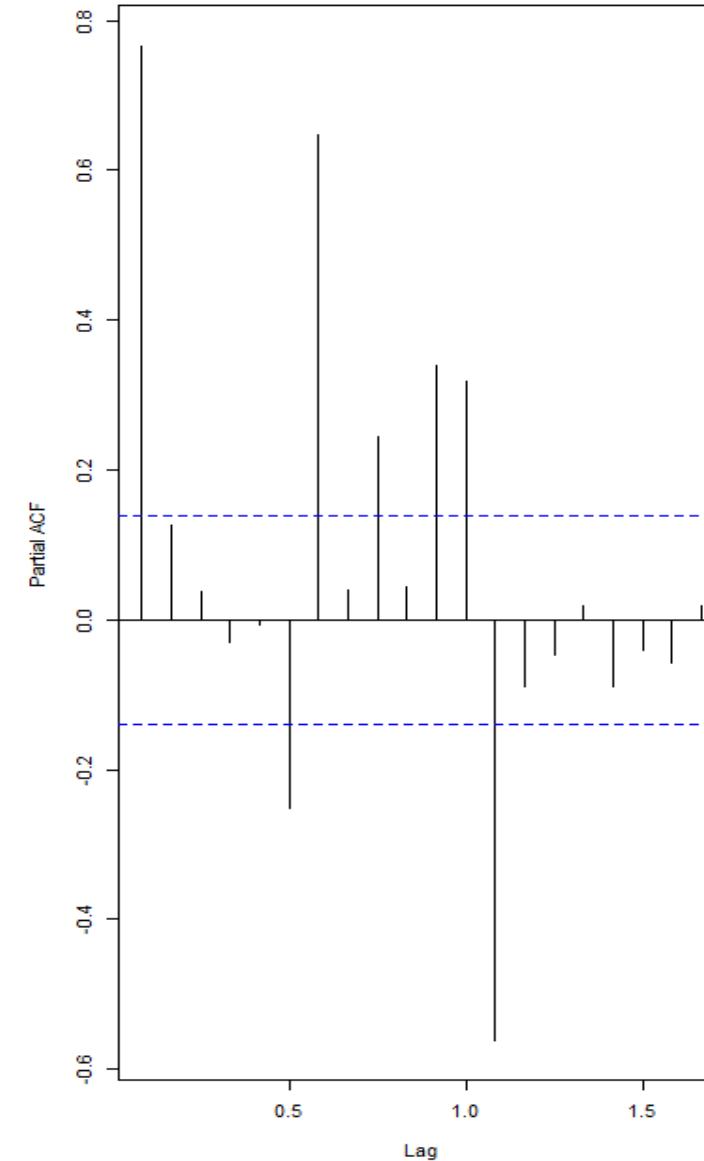
ACF and PACF (Real-world): Decomposing Time Series into the 3 Components – RPM



U.S..Air.Carrier.Traffic.Statistics...Revenue.Passenger.Miles



Series milestimeseries



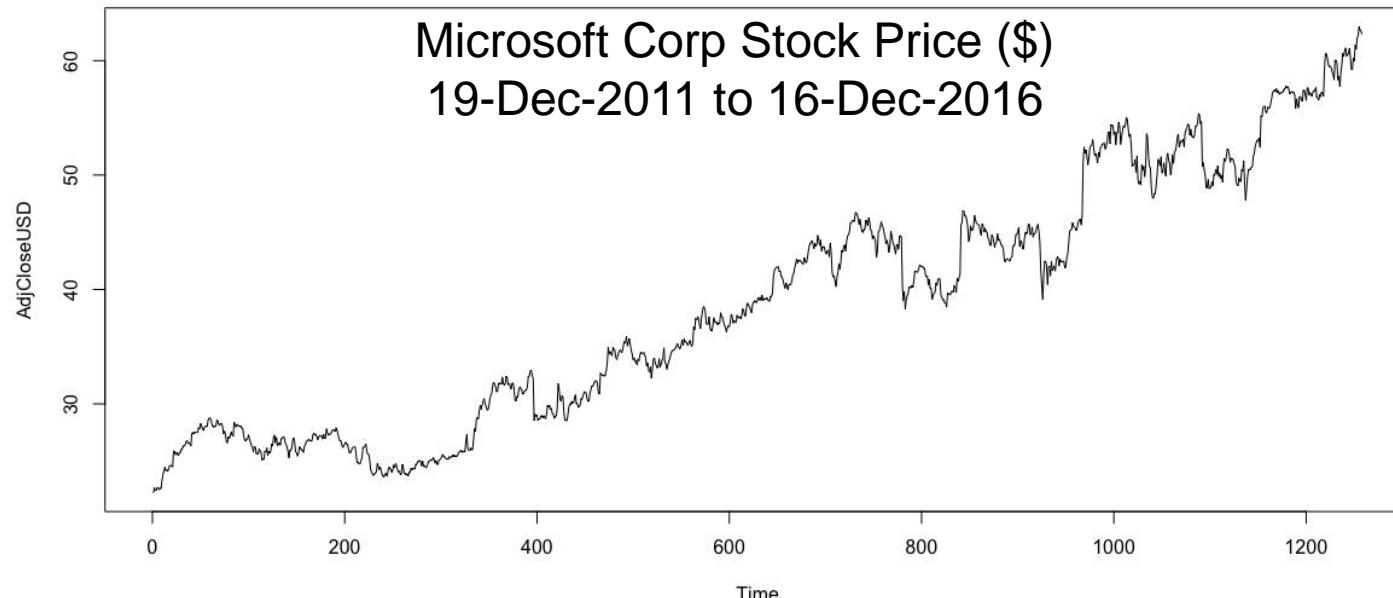
Stationary and Non-Stationary [Excel “Correlation Stationarity”]

- Stationary data has constant statistical properties – mean, variance, autocorrelation, etc. – over time
- If the data is stationary, forecasting is easier!
- Differencing to convert non-stationary to stationary

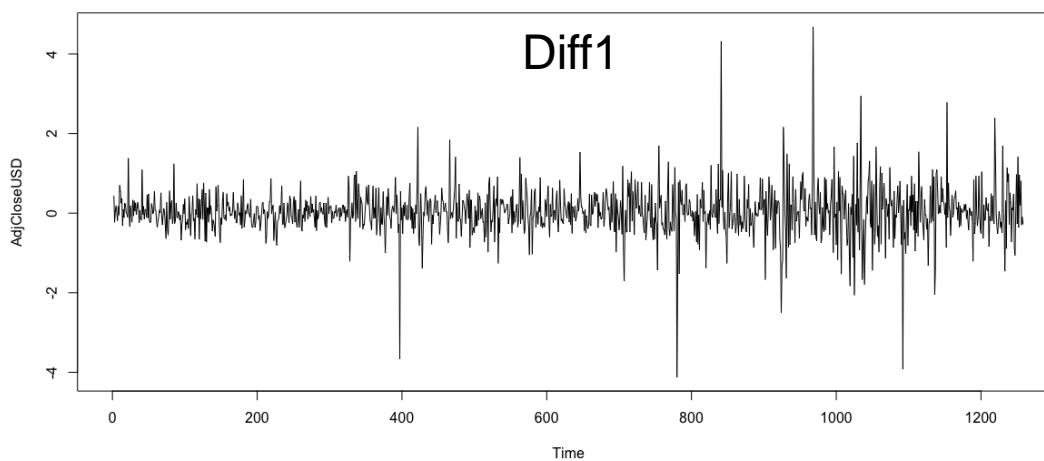


EXCEL ACTIVITY

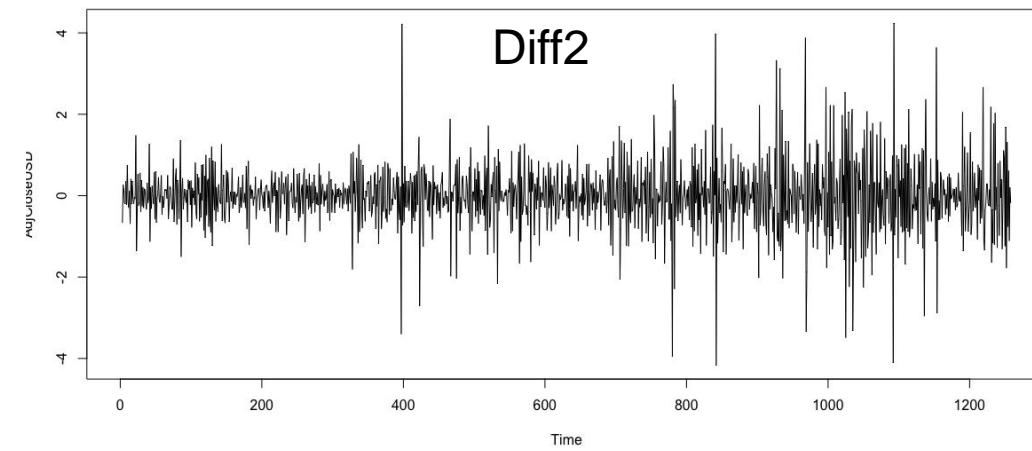
Removing Trend from Data



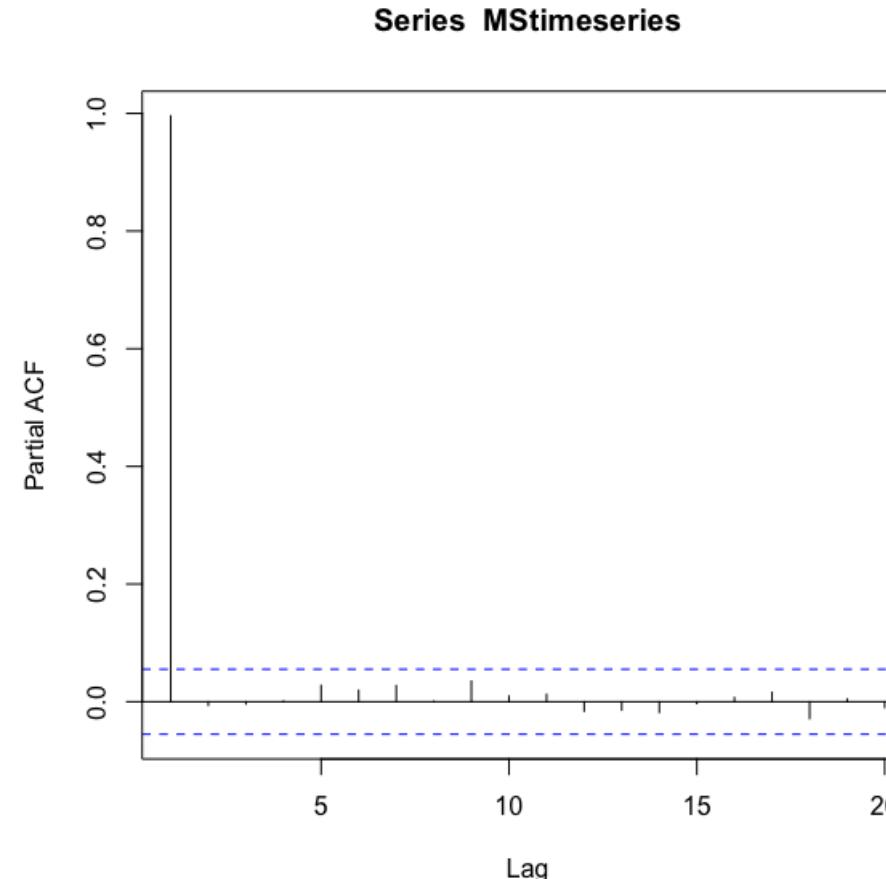
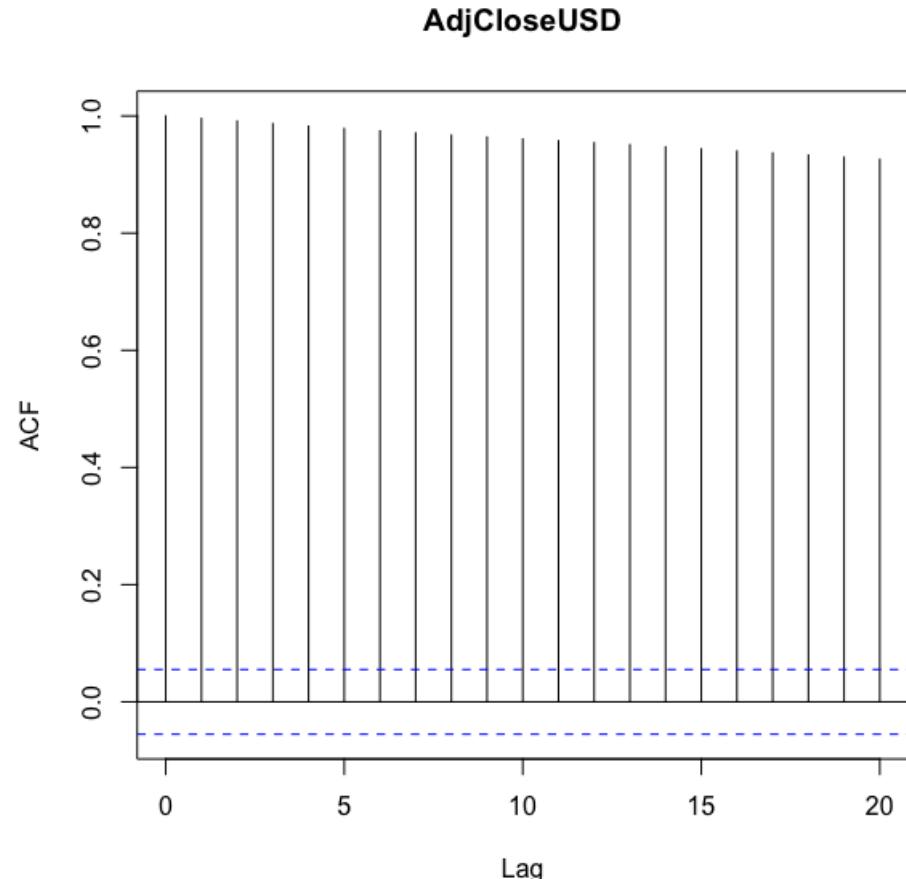
Rcode: MStimeseriesdiff1 <- diff(MStimeseries, differences=1)



Rcode: MStimeseriesdiff2 <- diff(MStimeseries, differences=2)



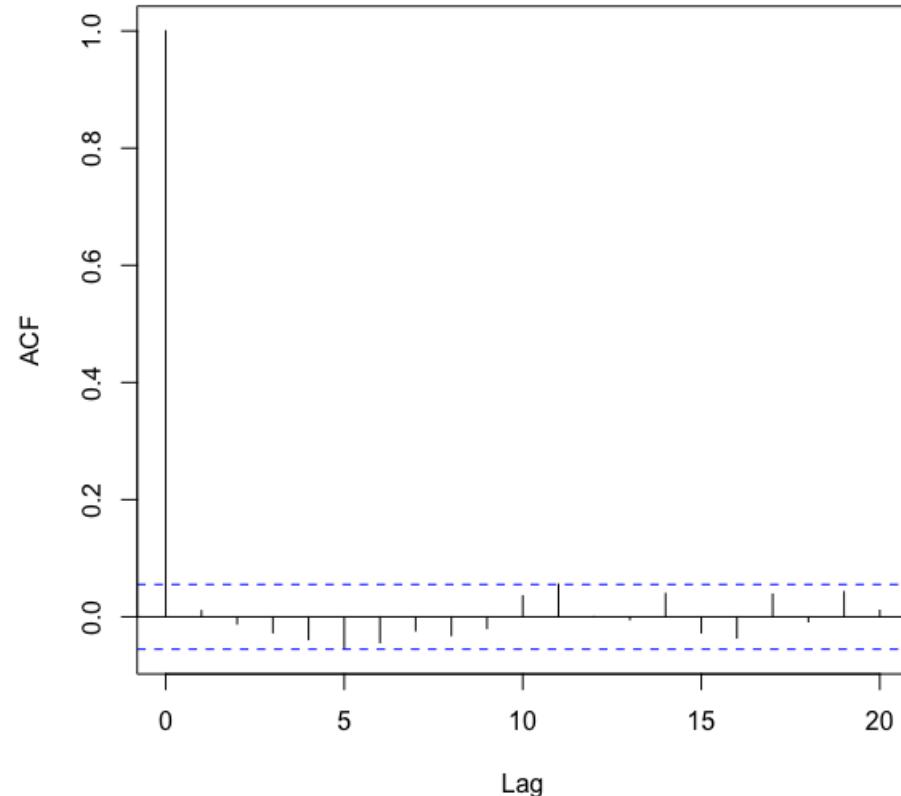
ACF and PACF of Non-Stationary Data



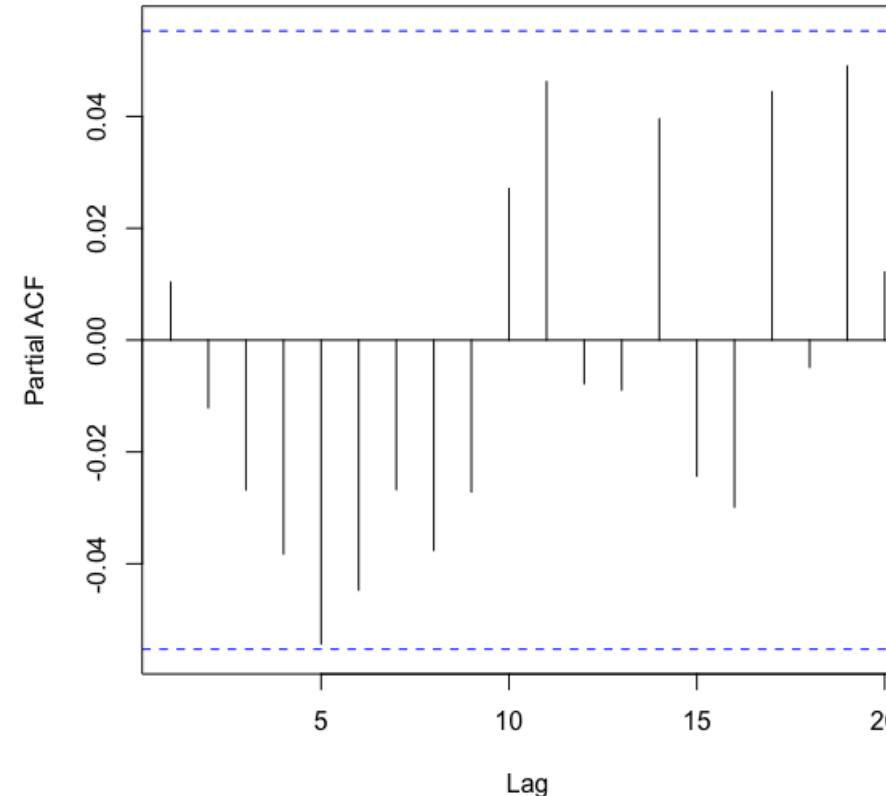
Price of Microsoft stock is highly correlated with the previous day's value.

ACF and PACF of Stationary Data

AdjCloseUSD



Series MStimeseriesdiff1



Daily changes in Microsoft stock price are essentially random.

ACF and PACF of Stationary and Non-Stationary

- Non-stationary series have an ACF that remains significant for half a dozen or more lags, rather than quickly declining to zero.
- You must difference such a series until it is stationary before you can identify the process. *Tip: In practice, never perform more than 2 non-seasonal differences, or one seasonal and one non-seasonal difference; also, never perform 2 seasonal differences.*



ARIMA MODEL BUILDING

Box–Jenkins Methodology

- Model identification and model selection
 - Make sure variables are stationary. Difference as necessary to get a constant mean and transformations to get constant variance.
 - Check for seasonality: Decays and spikes at regular intervals in ACF plots.
- Parameter estimation
 - Compute coefficients that best fit the selected model.
- Model checking
 - Check if residuals are independent of each other and constant in mean and variance over time (white noise).

http://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/NCSS/The_Box-Jenkins_Method.pdf

AutoRegressive Integrated Moving Average - ARIMA(p,d,q) Model

- p is the number of autoregressive [AR(p)] terms (a linear regression of the current value of the series against one or more prior values of the series)
 - Maximum lag beyond which PACF is 0 when PACF shows a **sharp cutoff** and/or lag1 autocorrelation is **positive**, i.e., series is slightly *underdifferenced*.



AutoRegressive Integrated Moving Average - ARIMA(p,d,q) Model

- d is the number of non-seasonal differences (order of the differencing) used to make the time series stationary [$I(d)$]



AutoRegressive Integrated Moving Average - ARIMA(p,d,q) Model

- q is the order of the moving average [MA(q)] model
 - Maximum lag beyond which the ACF is 0 when ACF shows a **sharp cutoff** and/or lag1 autocorrelation is **negative**, i.e., the series is slightly ***overdifferenced***.

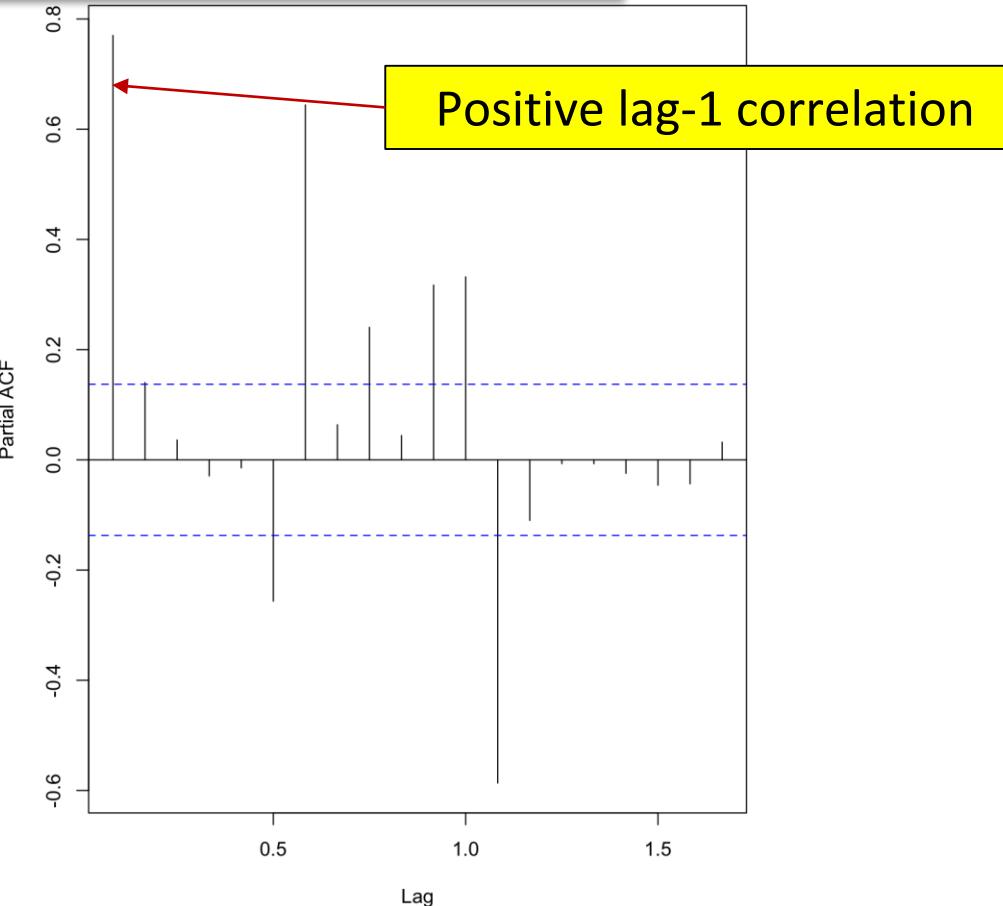
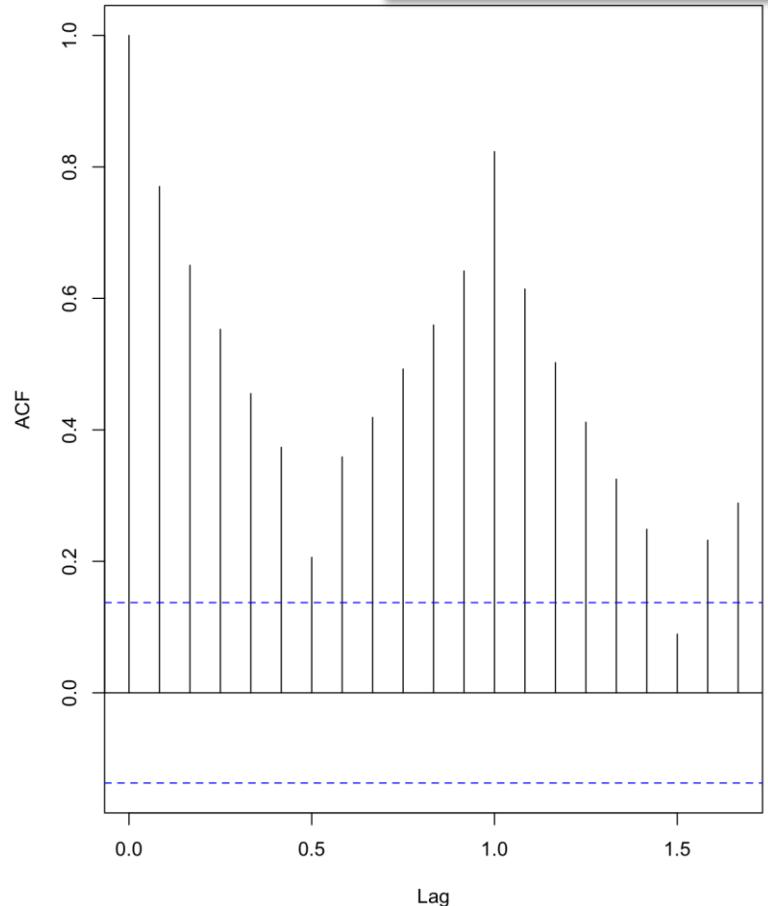
For a comprehensive list of rules for identification of ARIMA models, see

<https://people.duke.edu/~rnau/411arim3.htm> and <https://people.duke.edu/~rnau/arimrule.htm>.



AutoRegressive Integrated Moving Average - ARIMA(p,d,q) Model – Effect of Differencing

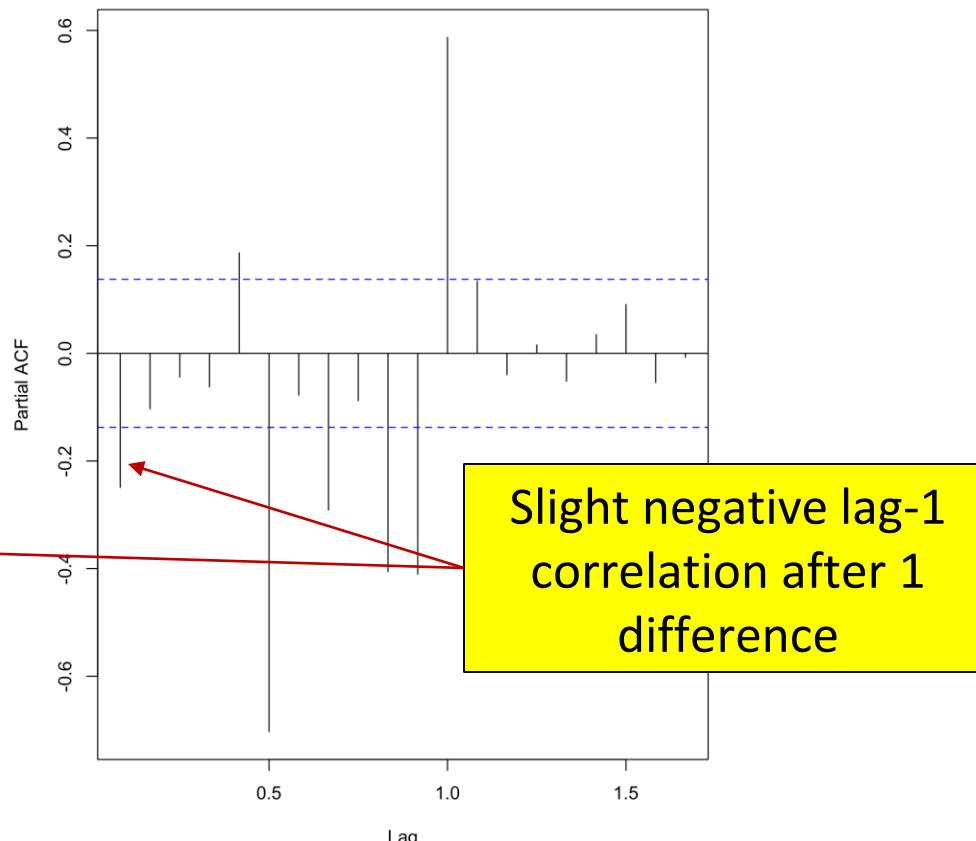
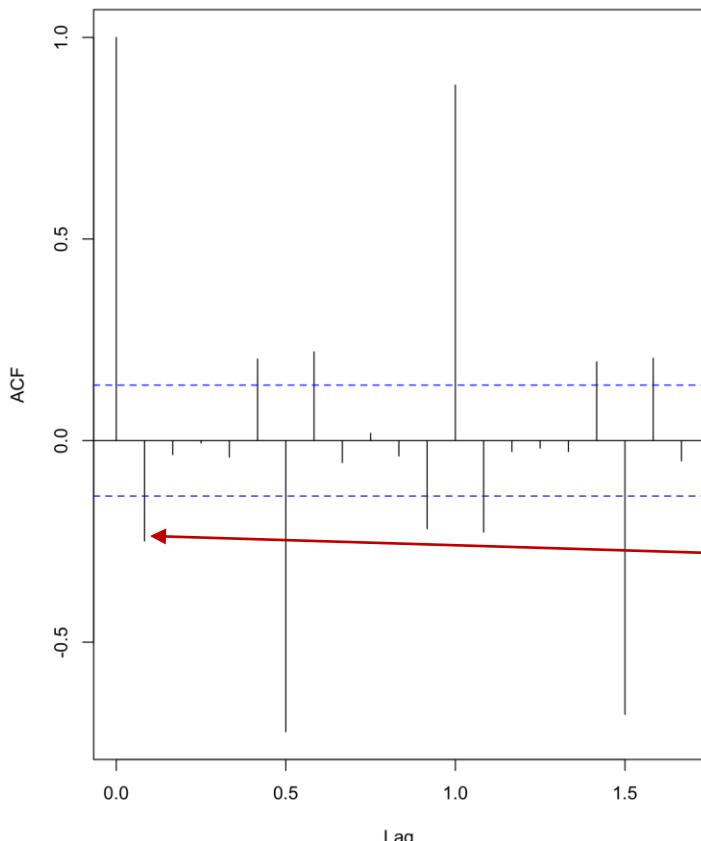
```
acf(milestimeseries, lag.max=20)  
pacf(milestimeseries, lag.max=20)
```



Positive lag-1 correlation

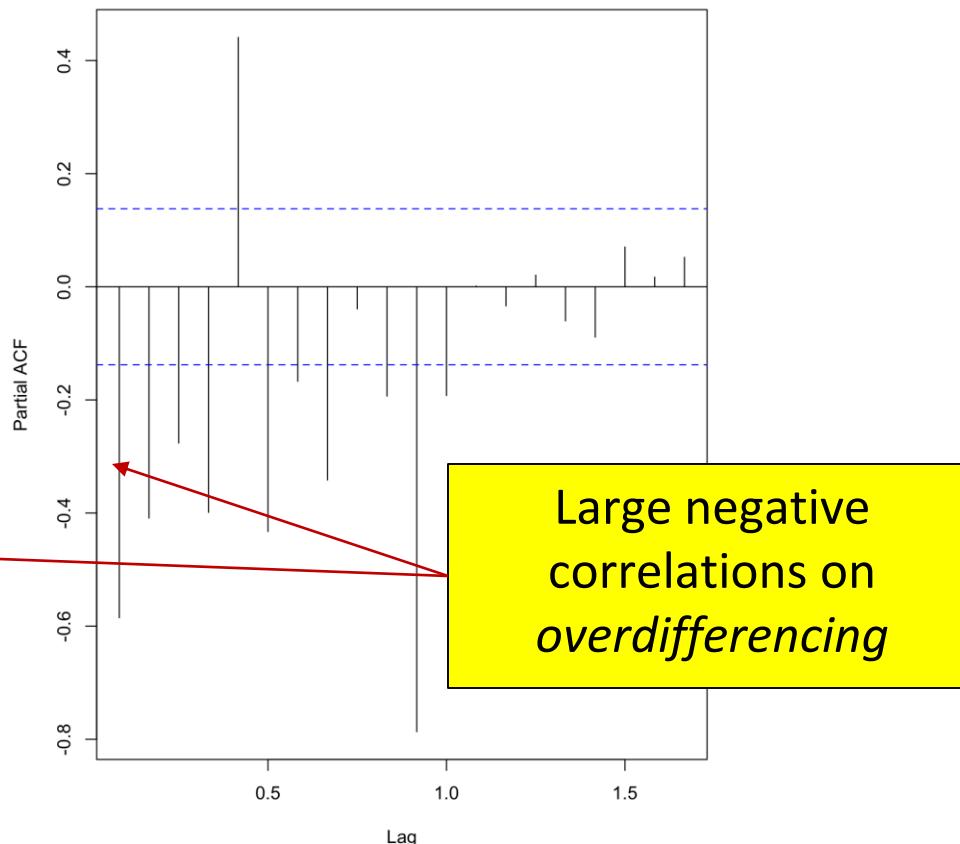
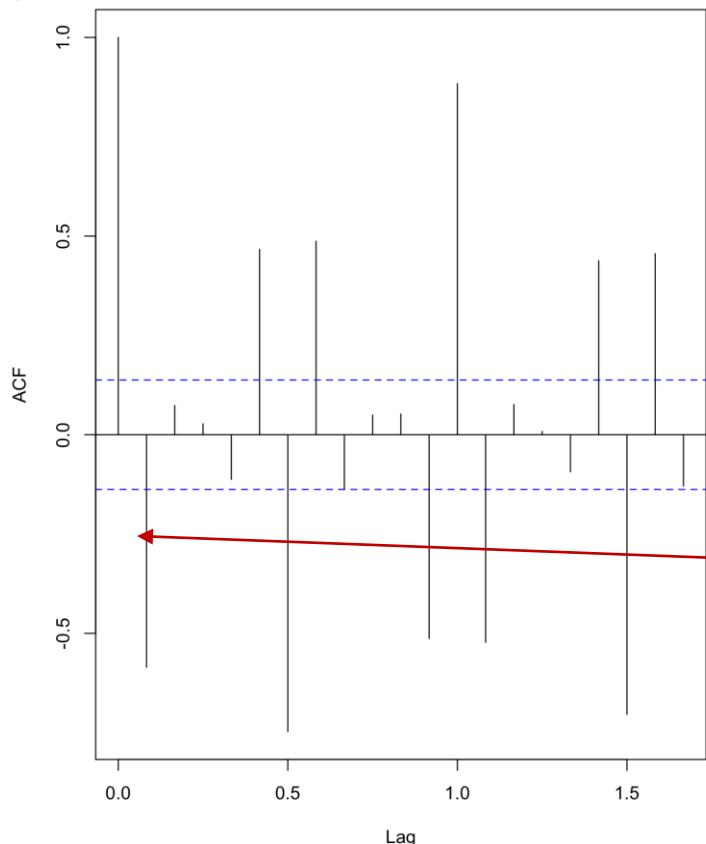
AutoRegressive Integrated Moving Average - ARIMA(p,d,q) Model – Effect of Differencing

```
milestimeseriesdiff1 <- diff(milestimeseries, differences=1)  
milestimeseriesdiff1  
acf(milestimeseriesdiff1, lag.max=20)  
pacf(milestimeseriesdiff1, lag.max=20)
```



AutoRegressive Integrated Moving Average - ARIMA(p,d,q) Model – Effect of Differencing

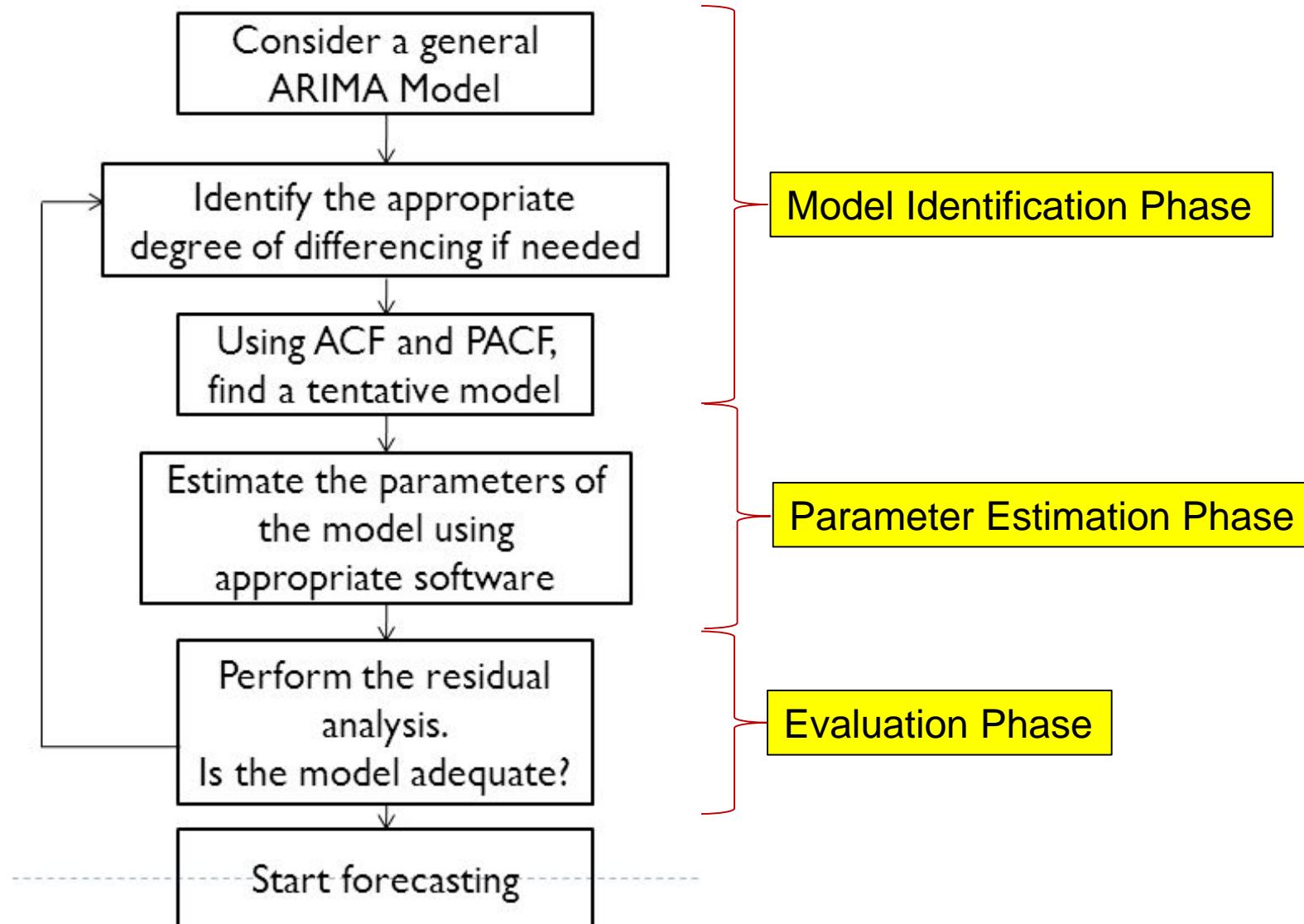
```
milestimeseriesdiff2 <- diff(milestimeseries, differences=2)  
milestimeseriesdiff2  
acf(milestimeseriesdiff2, lag.max=20)  
pacf(milestimeseriesdiff2, lag.max=20)
```



- Non-seasonal ARIMA models are denoted ARIMA(p,d,q)
- Seasonal ARIMA (SARIMA) models are denoted ARIMA(p,d,q)(P,D,Q)_m, where m refers to the number of periods in each season and (P,D,Q) refer to the autoregressive, differencing and moving average terms of the seasonal part of the ARIMA model.



Time Series Model Building Using ARIMA

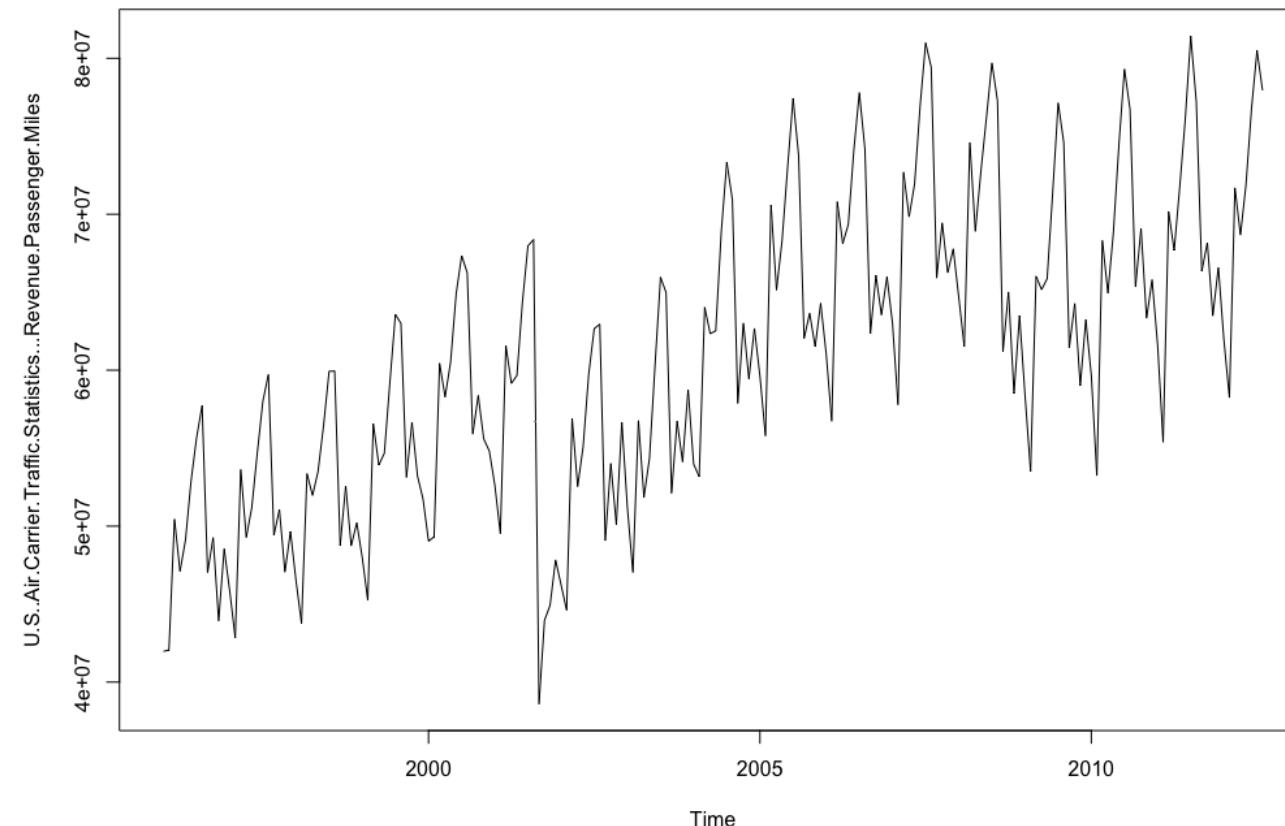


Time Series Model Building Using ARIMA

Identification Phase

Step 1: Plot the data (transform data to stabilize variance, if required)

R code: `plot(milestimeseries)`

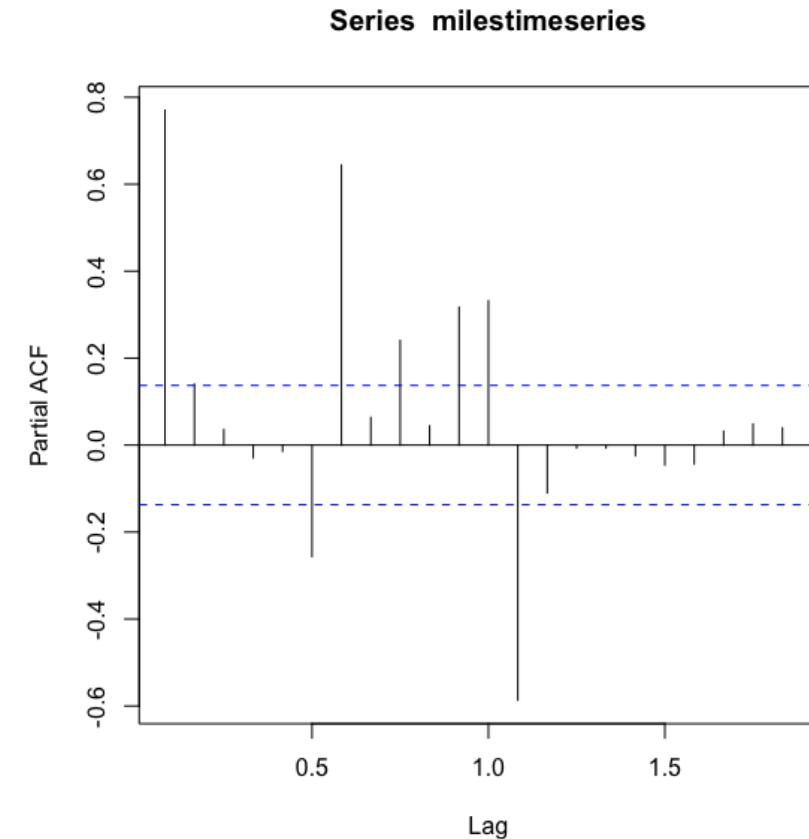
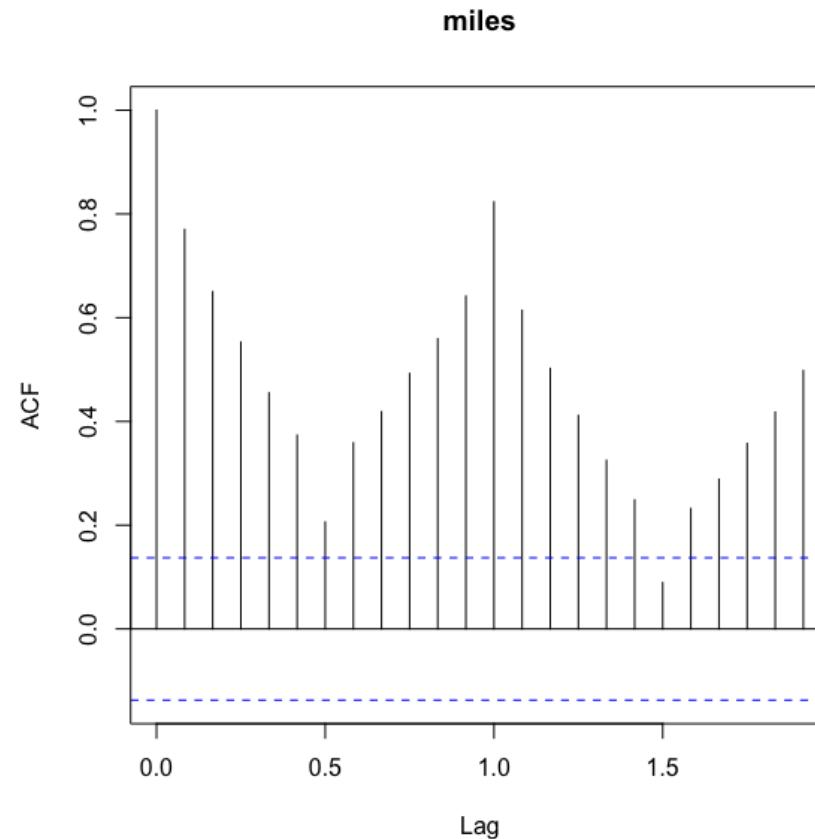


Time Series Model Building Using ARIMA

Identification Phase

Step 2: Plot ACF and PACF to get preliminary understanding of the processes involved.

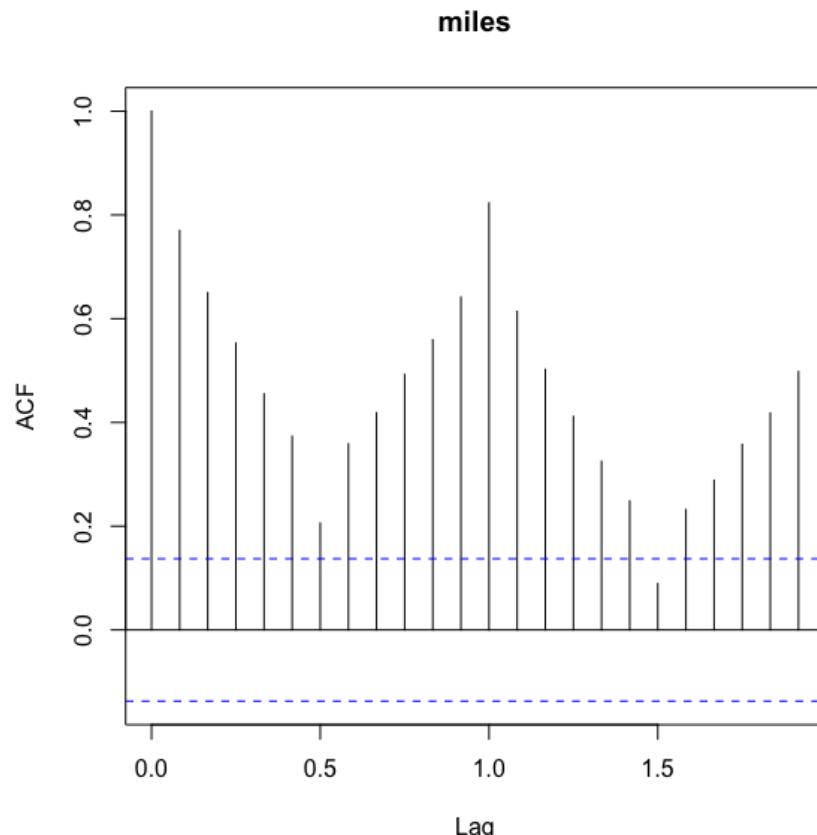
R code: `acf(milestimeseries)` `pacf(milestimeseries)`



Time Series Model Building Using ARIMA

Identification Phase

Step 2: Plot ACF and PACF to get preliminary understanding of the processes involved.

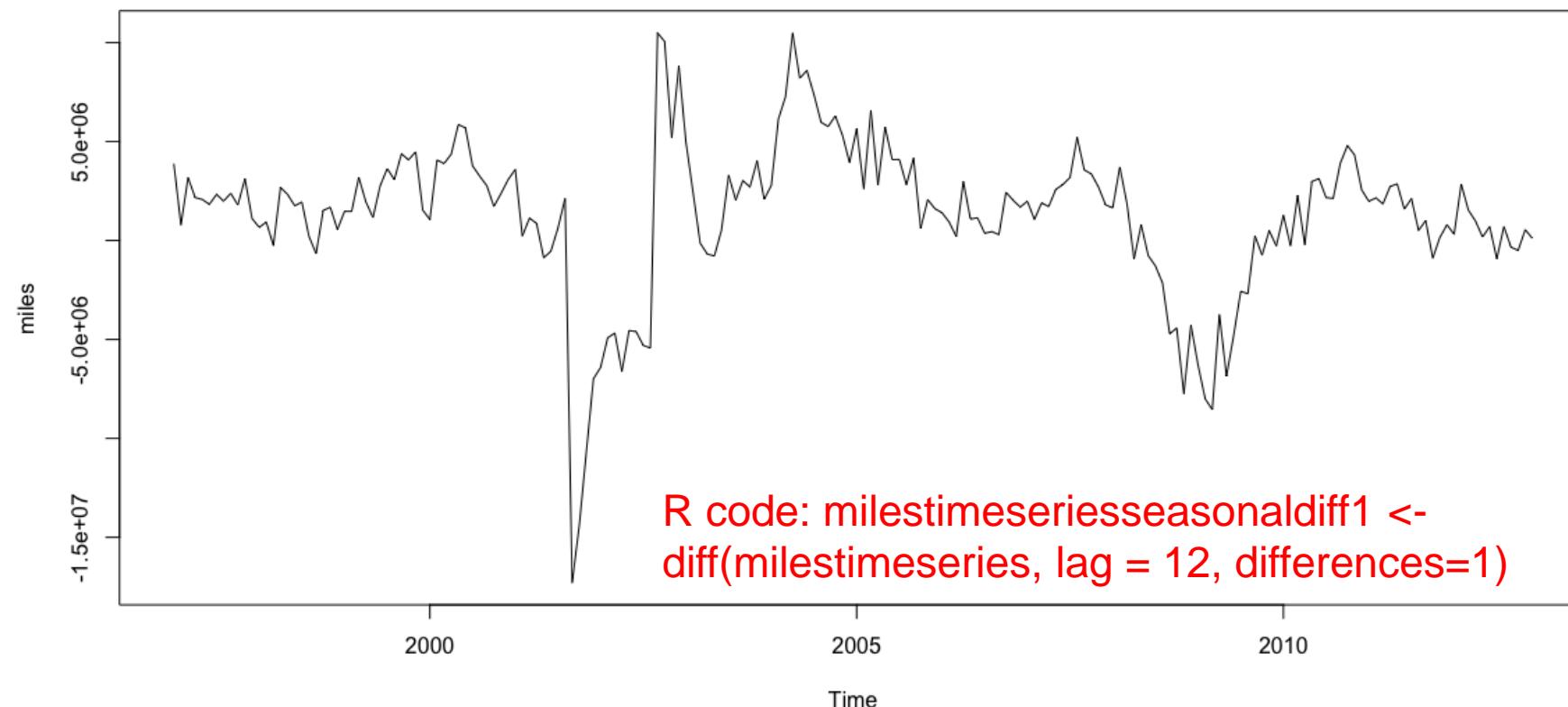


The suspension bridge pattern in ACF (also, positive and negative spikes in PACF) suggests non-stationarity and strong seasonality.

Time Series Model Building Using ARIMA

Identification Phase

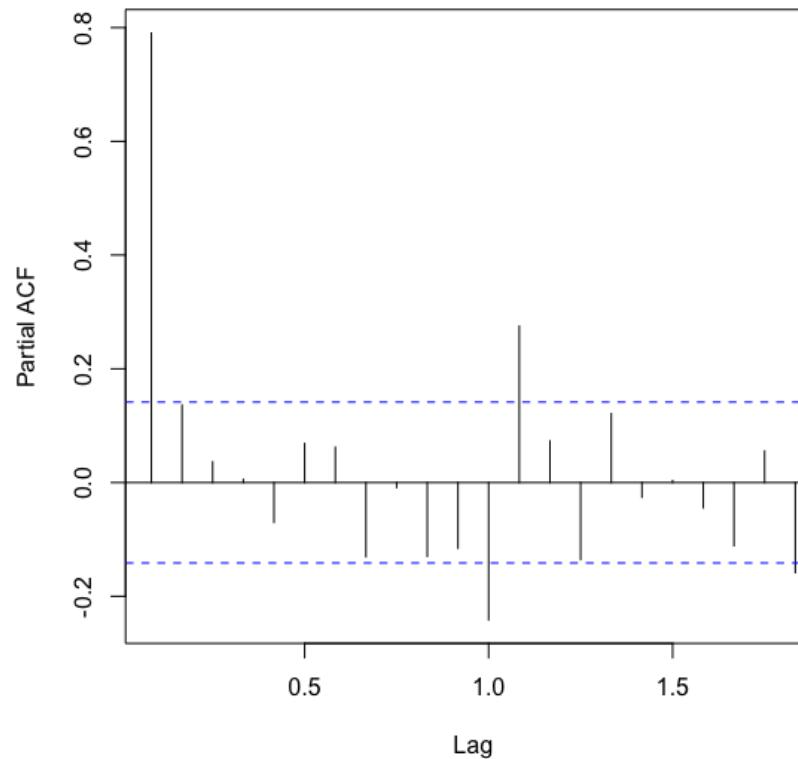
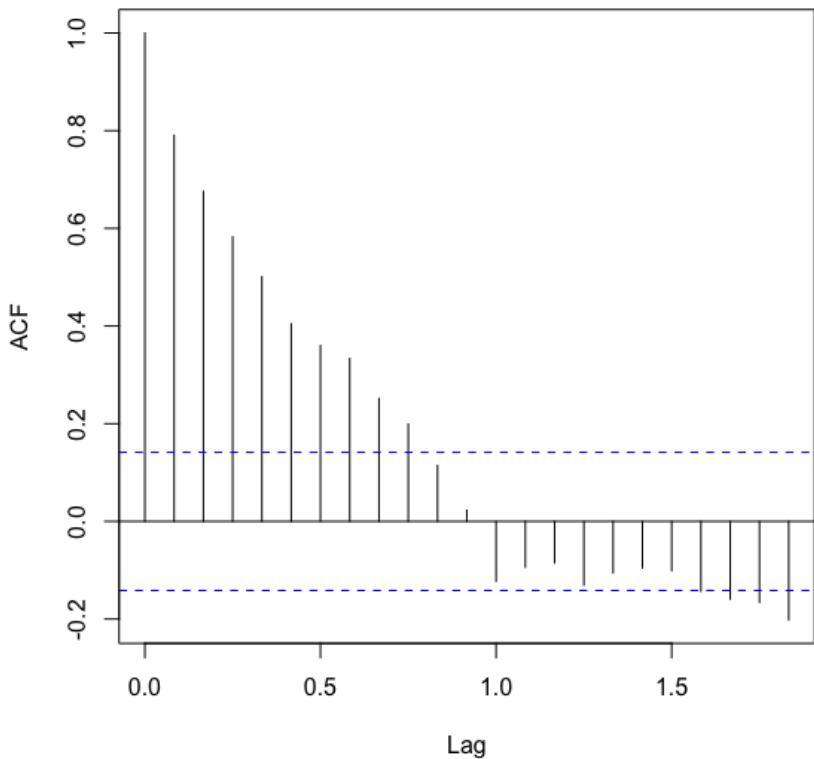
Step 3: Perform seasonal differencing (t_0-t_{12}, t_1-t_{13} , etc.) on the original time series to get seasonal stationarity. This is the same as an ARIMA(0,0,0)(0,1,0)₁₂ model.



Time Series Model Building Using ARIMA

Identification Phase

Step 4: Check ACF and PACF of seasonally differenced data to explore remaining dependencies and identify model(s).



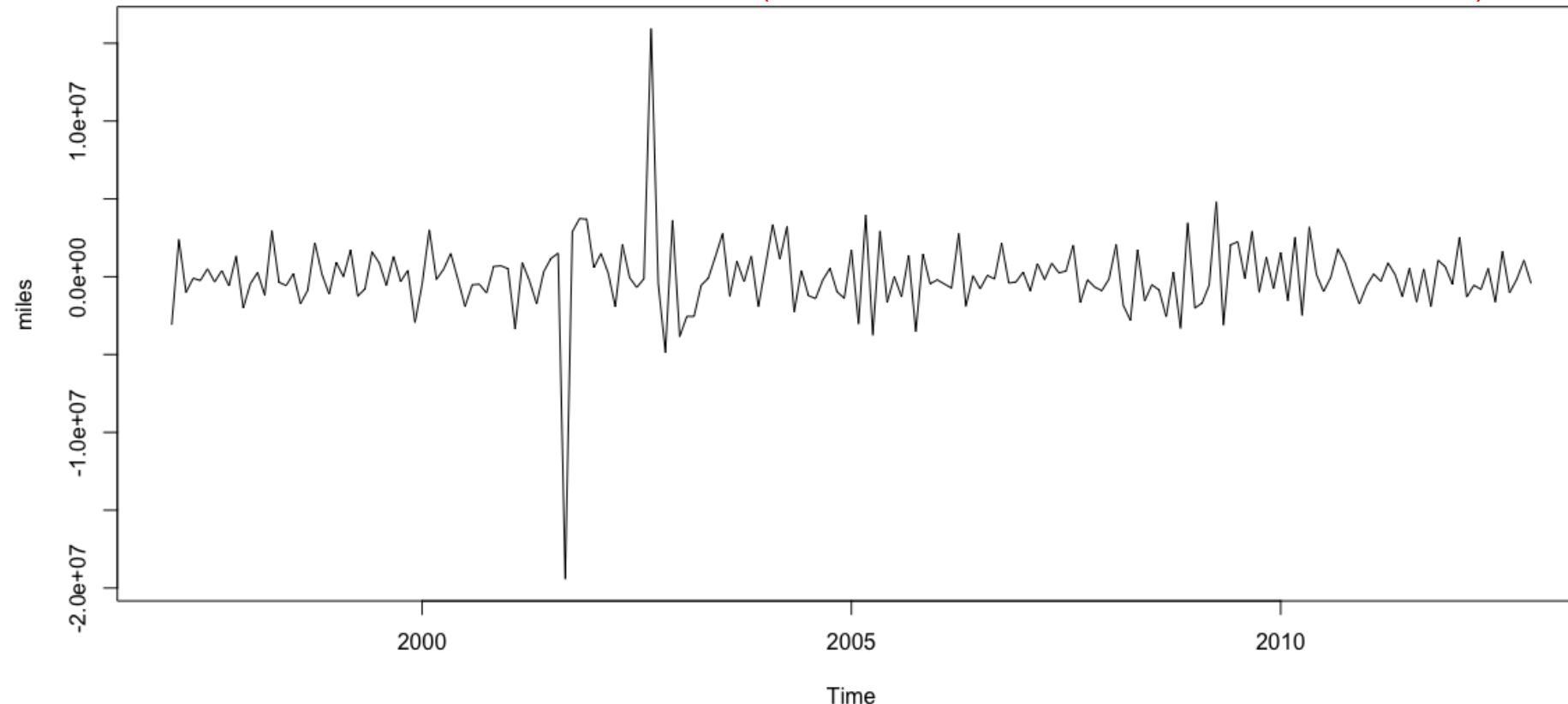
Positive autocorrelation indicates *underdifferencing* requiring either an AR term or a non-seasonal differencing to fix it.

Time Series Model Building Using ARIMA

Identification Phase

Step 5: Perform a non-seasonal differencing on seasonally differenced data. This is like an ARIMA(0,1,0)(0,1,0)₁₂ model.

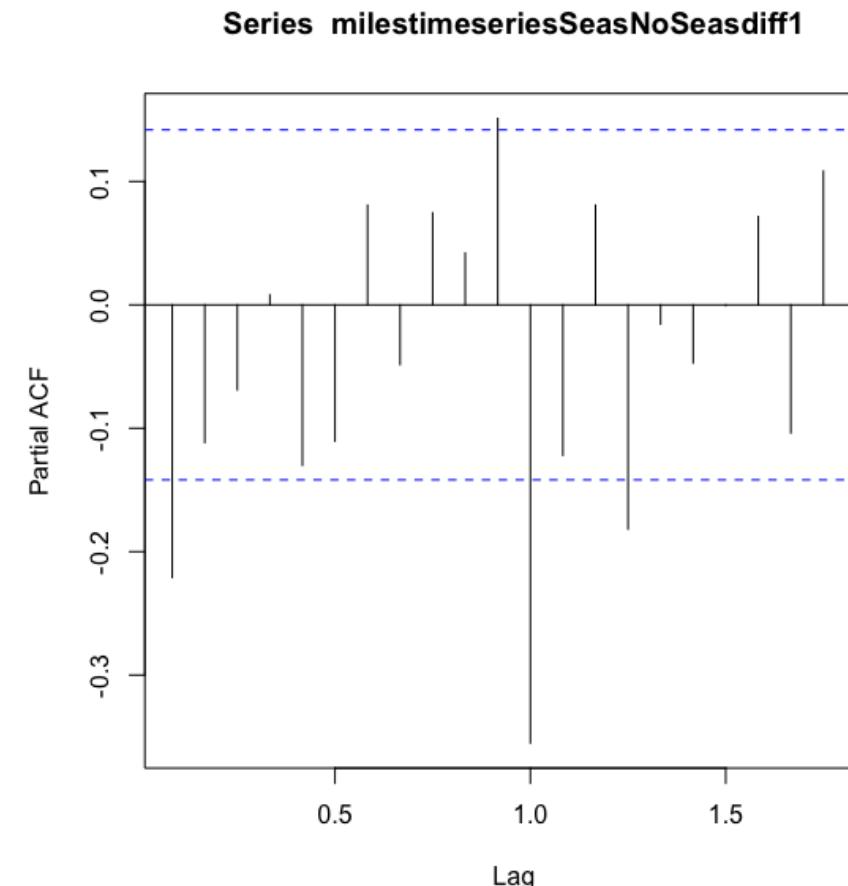
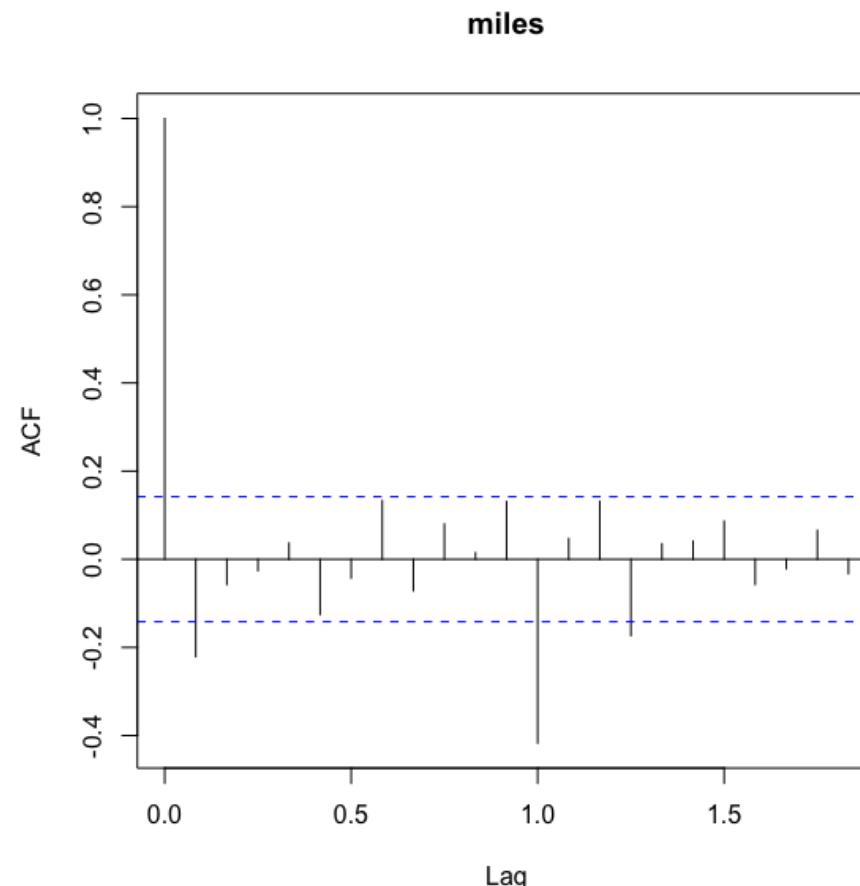
Rcode: `milestimeseriesSeasNoSeasdiff1 <- diff(milestimeseriesseasonaldiff1, differences=1)`



Time Series Model Building Using ARIMA

Identification Phase

Step 6: Check ACF and PACF to explore remaining dependencies.

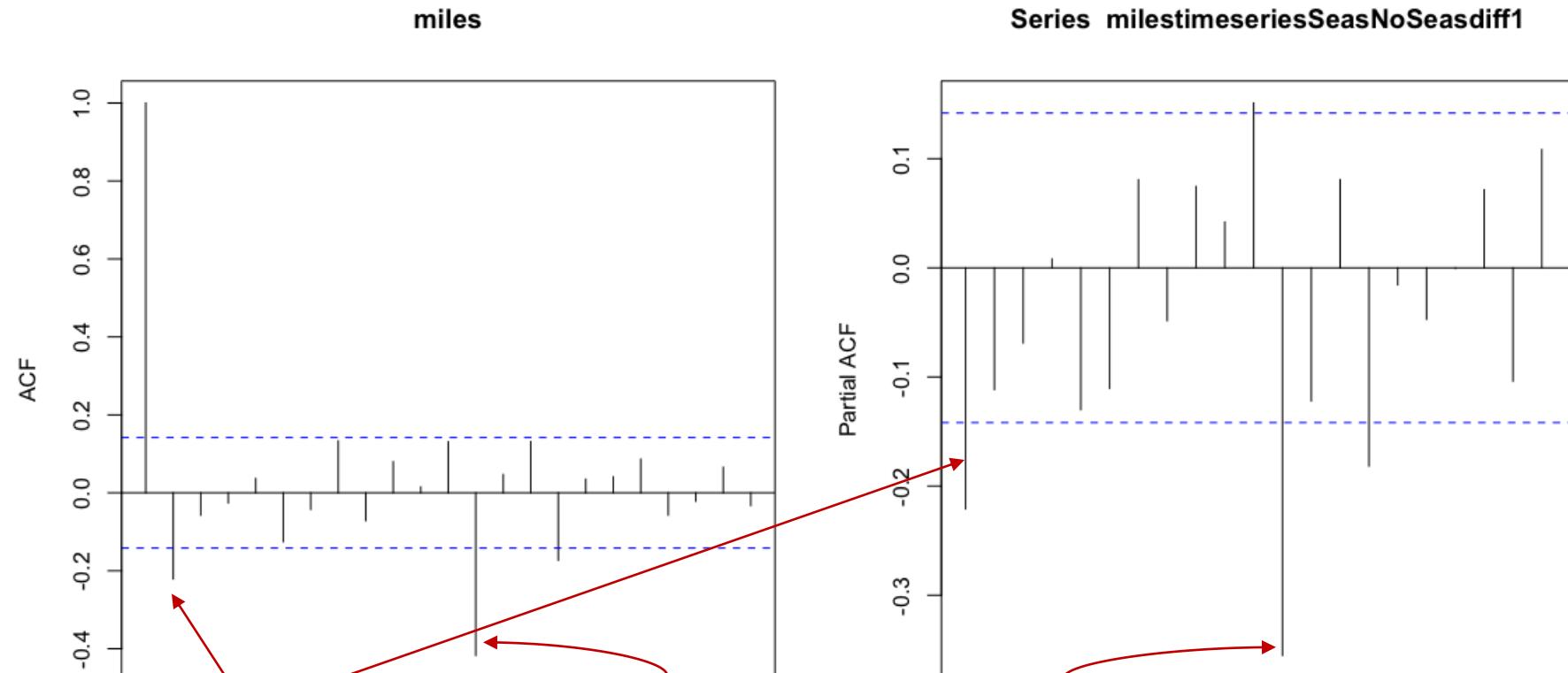


Time Series Model Building Using ARIMA

Identification Phase

Step 7: This indicates an ARIMA(0,1,1)(0,1,1)₁₂ model.

R code: Arima(milestimeseries, order = c(0,1,1), seasonal = c(0,1,1), include.drift = TRUE)



As lag1 and the significant lag at seasonal period are **negative**, include a MA(1) and SMA(1) terms. Negative autocorrelation indicates *overdifferencing* requiring MA term to fix it.

Time Series Model Building Using ARIMA

Parameter Estimation Phase

Step 7: Calculate parameters using the identified model(s). Use AIC to pick the best model.

```
Series: milestimeseries  
ARIMA(0,1,1)(0,1,1)[12]
```

Coefficients:

	ma1	sma1
-	-0.2757	-0.7277
s.e.	0.0791	0.0653

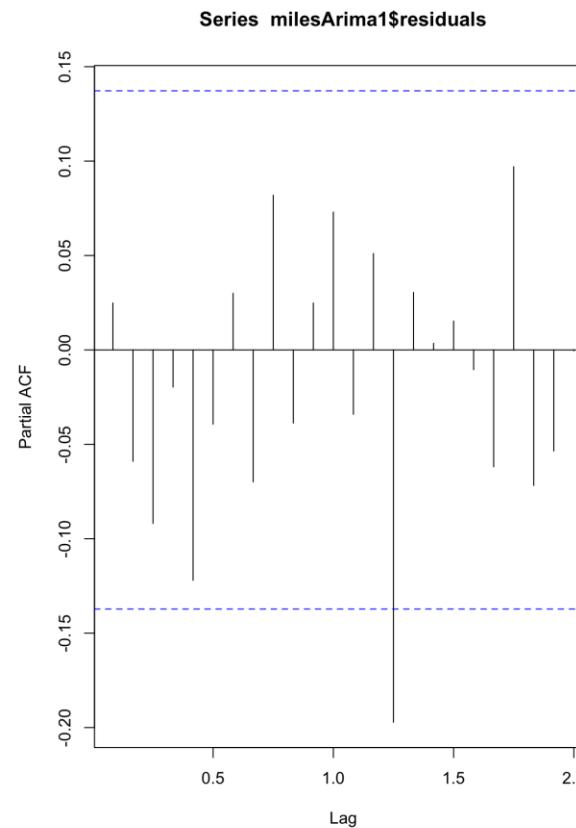
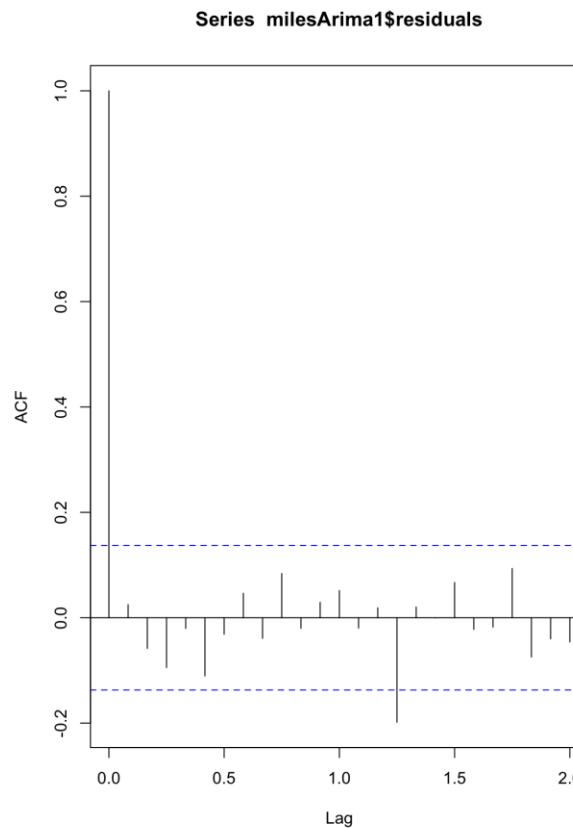
```
sigma^2 estimated as 3.974e+12: log likelihood=-3045.1  
AIC=6096.21    AICc=6096.34    BIC=6105.97
```



Time Series Model Building Using ARIMA

Evaluation Phase

Step 8: Check ACF and PACF of the residuals to evaluate model. The residuals indicate white noise. Indicates a good model that can be used for forecasting.



Rcode: `acf(milesArima1$residuals, lag.max = 24)`
`pacf(milesArima1$residuals, lag.max = 24)`

Time Series Model Building Using ARIMA

Evaluation Phase

Step 8: The residuals indicate white noise. Can be checked using Ljung-Box test.

R code: `Box.test(milesArima1$residuals, lag=24, type="Ljung-Box")`

$$Q^* = n(n + 2) \sum_{k=1}^h \frac{r_k^2}{n - k}$$

h is the maximum lag being considered
 n is the # of observations (length of the time series)
 r_k is the autocorrelation

* For non-seasonal time series, use $h = \min(10, n/5)$

For seasonal time series, use $h = \min(2m, n/5)$, where m is the seasonal period

Box-Ljung test

```
data: milesArima1$residuals  
X-squared = 23.114, df = 24, p-value = 0.5131
```

If residuals are white noise (purely random), then Q has a χ^2 distribution with $h-p$ degrees of freedom, where p is the number of parameters estimated in the model

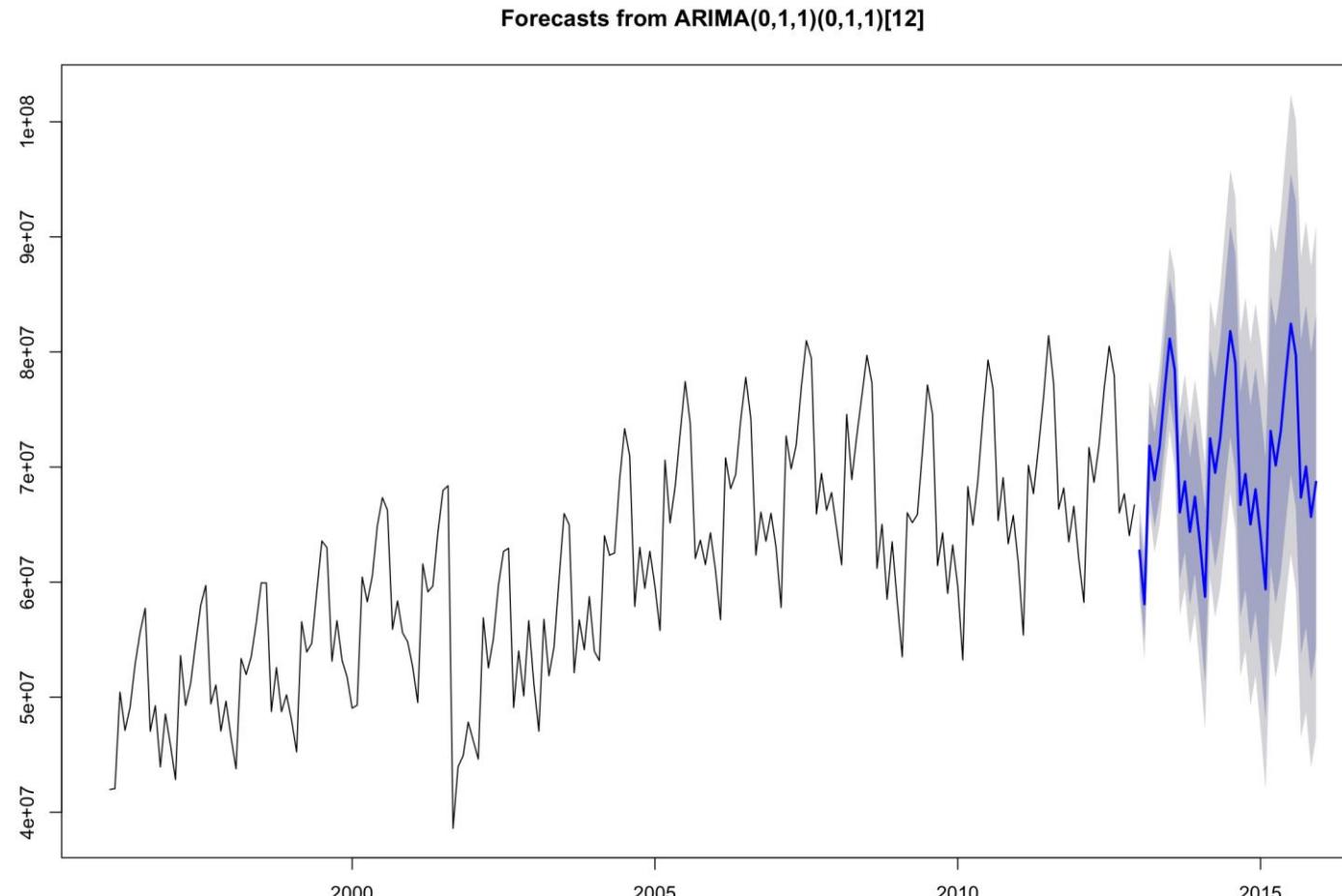
* <http://robjhyndman.com/hyndsight/ljung-box-test/>

Time Series Model Building Using ARIMA

Forecasting Phase

Step 9: Start forecasting.

R code: `forecast.Arima(milesArima1, h=36)` or `forecast(milesArima1, h=36)`



Time Series Model Building Using ARIMA

Alternate Steps 5-9 from previous model

Step 5: Add an AR term to seasonally differenced data. This is like an ARIMA(1,0,0)(0,1,0)₁₂ model.

```
Series: milestimeseries  
ARIMA(1,0,0)(0,1,0)[12] with drift
```

Coefficients:

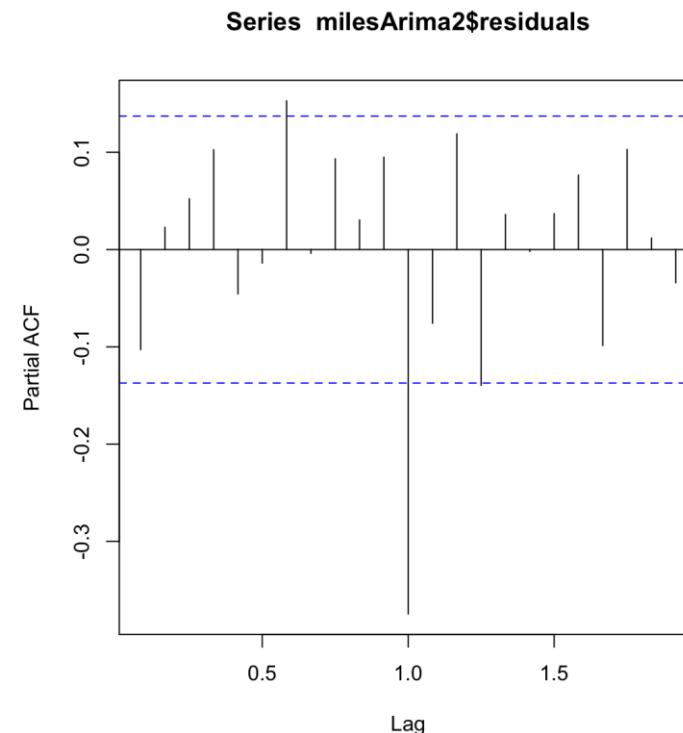
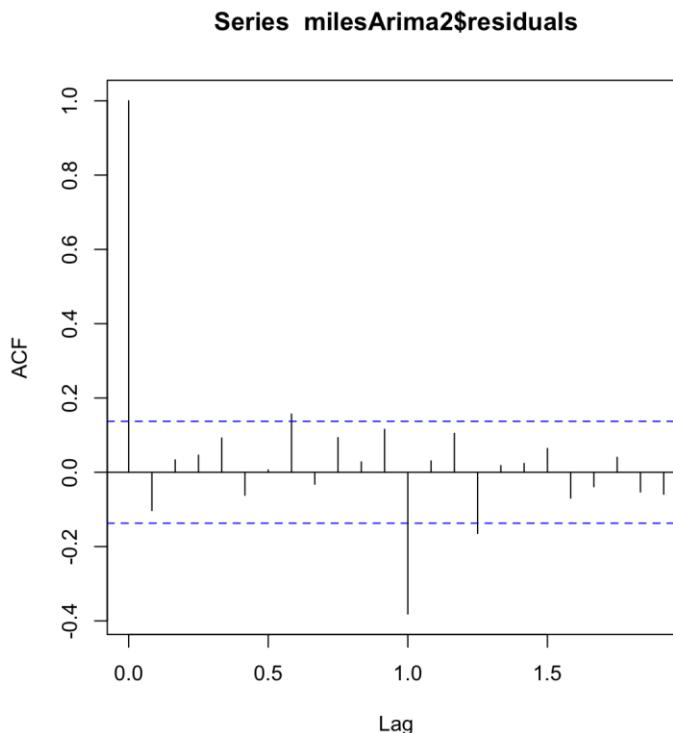
	ar1	drift
	0.7886	109198.14
s.e.	0.0436	65385.94

sigma^2 estimated as 5.541e+12: log likelihood=-3088.87
AIC=6183.73 AICc=6183.86 BIC=6193.5

Time Series Model Building Using ARIMA

Alternate Steps 5-9 from previous model

Step 6: Check ACF and PACF to identify remaining dependencies. Strong negative autocorrelation at the seasonal period indicates need for a seasonal MA term for an ARIMA(1,0,0)(0,1,1)₁₂ model.



Box-Ljung test

```
data: milesArima2$residuals  
X-squared = 61.725, df = 24, p-value = 3.632e-05
```

Time Series Model Building Using ARIMA

Alternate Steps 5-9 from previous model

Step 7: ARIMA(1,0,0)(0,1,1)₁₂ model.

```
Series: milestimeseries
ARIMA(1,0,0)(0,1,1)[12] with drift
```

Coefficients:

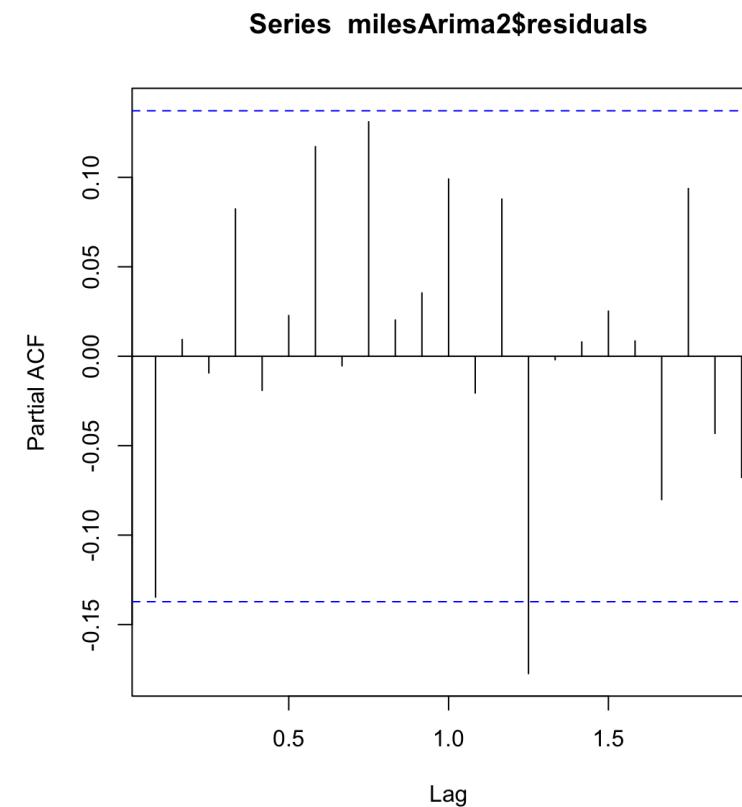
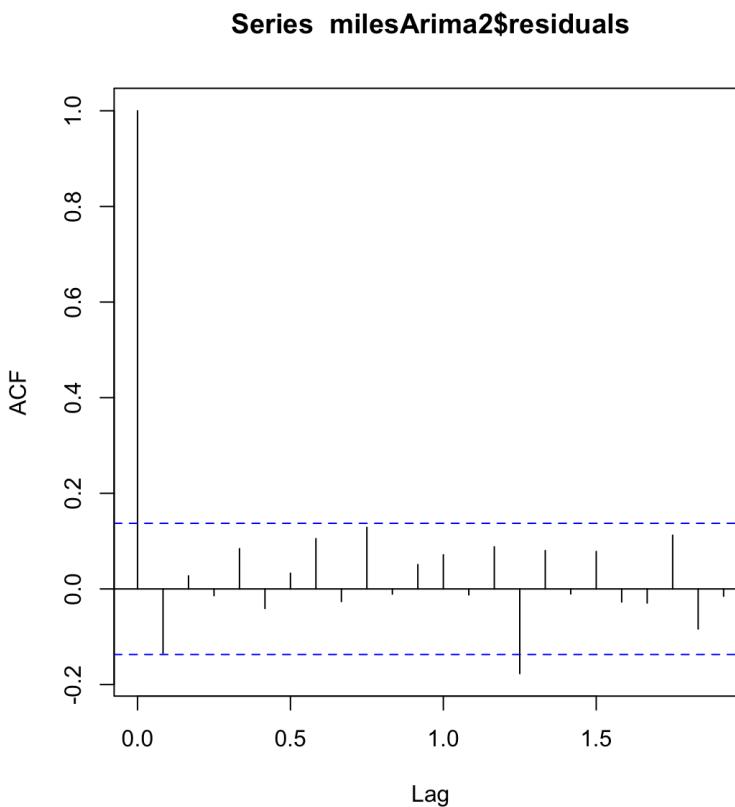
	ar1	sma1	drift
	0.8527	-0.7254	109302.05
s.e.	0.0378	0.0678	25762.84

sigma^2 estimated as 3.923e+12: log likelihood=-3059.73
AIC=6127.46 AICc=6127.67 BIC=6140.49

Time Series Model Building Using ARIMA

Alternate Steps 5-9 from previous model

Step 8: The residuals indicate white noise. Indicates a good model that can be used for forecasting.



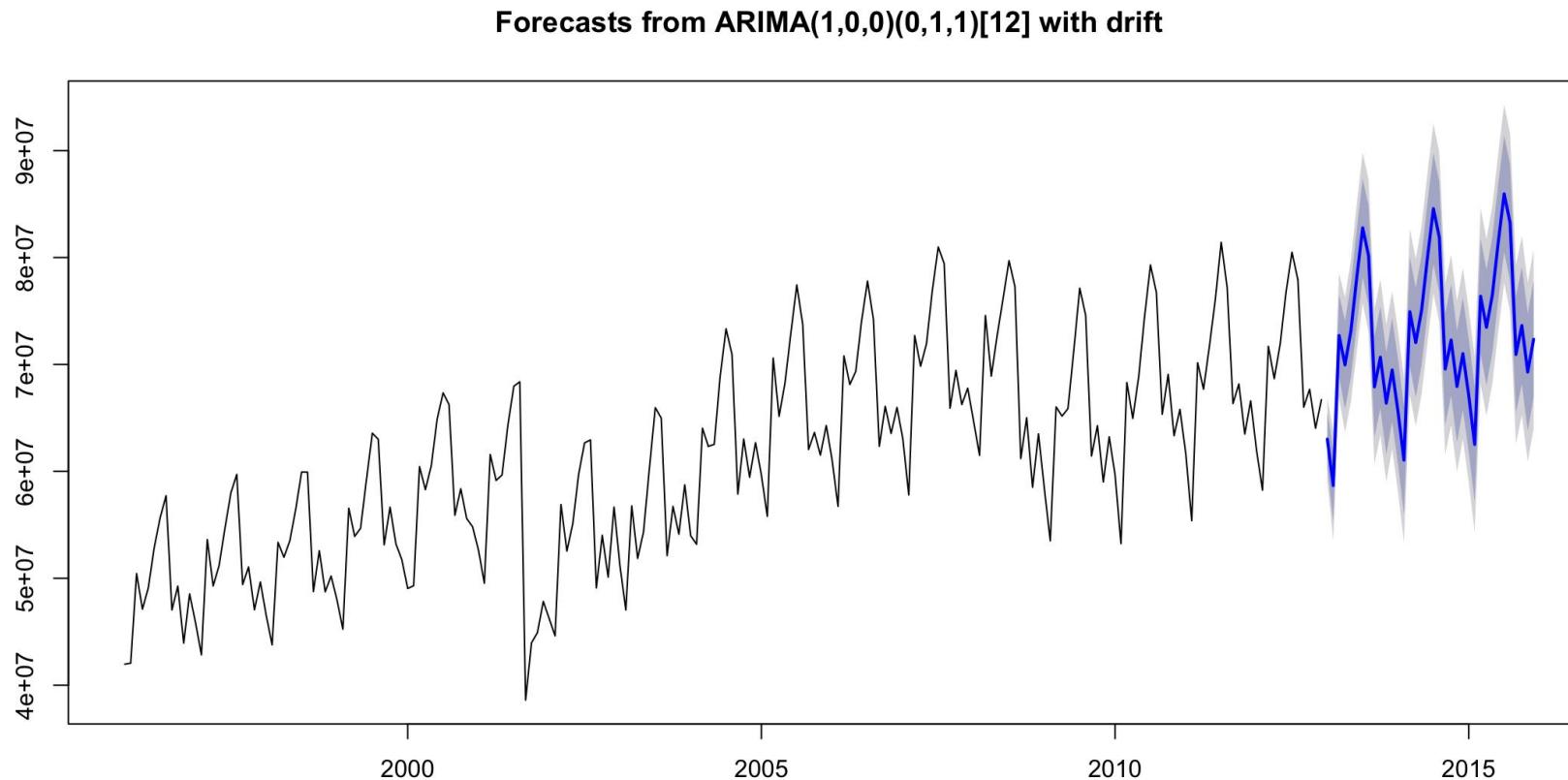
Box-Ljung test

```
data: milesArima2$residuals  
X-squared = 30.776, df = 24, p-value = 0.1603
```

Time Series Model Building Using ARIMA

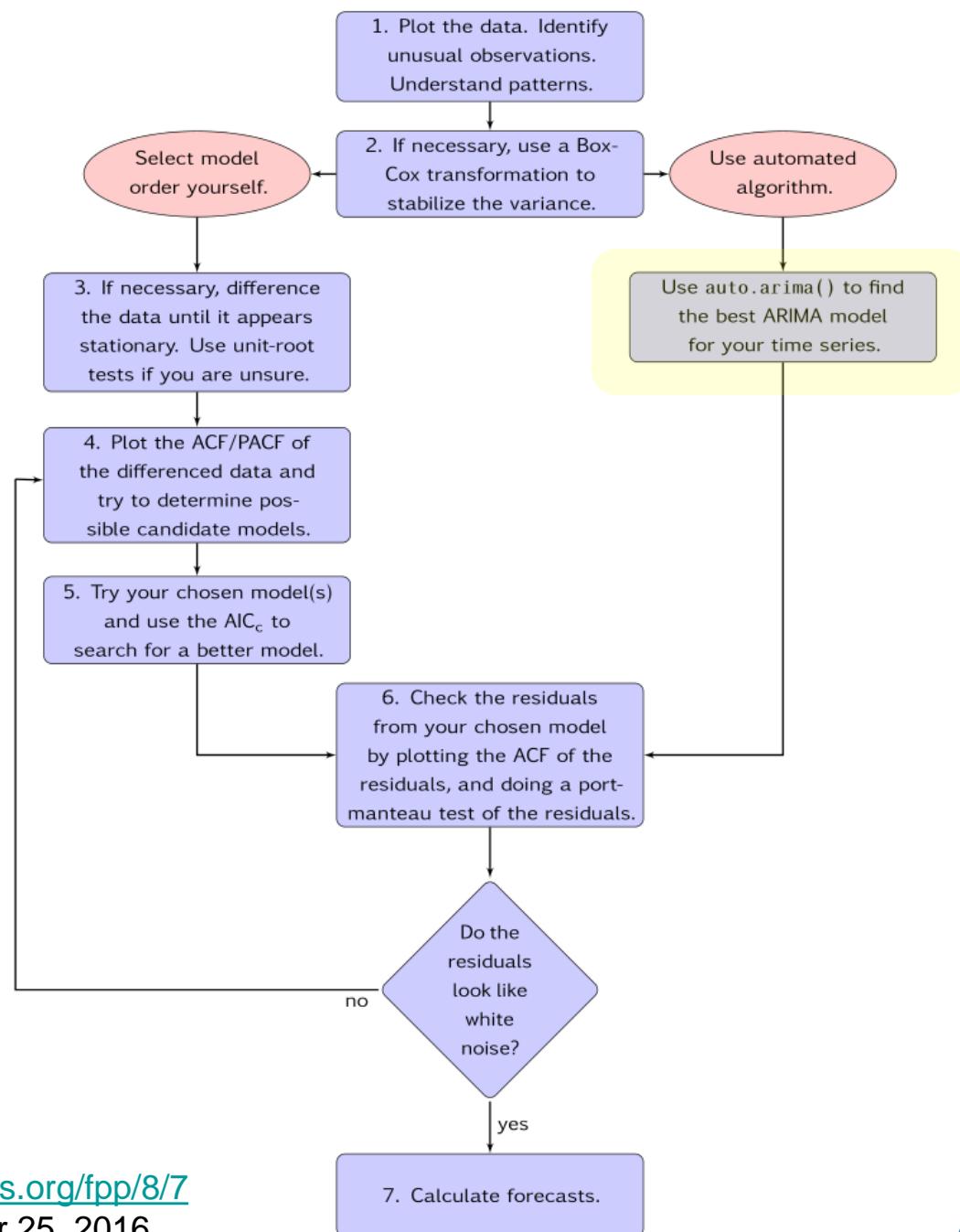
Alternate Steps 5-9 from previous model

Step 9: Start forecasting.



Model Selection in Practice

There are techniques that automate model selection



Source: <https://www.otexts.org/fpp/8/7>
Last accessed: November 25, 2016

Model Selection in Practice

auto.arima() in R using Hyndman-Khandakar algorithm.

1. Find d for stationarity.
2. p and q are selected using a stepwise search to minimize AICc on the differenced data.
 - a. The best model among the following is selected as the current model: ARIMA(2, d ,2), ARIMA(0, d ,0), ARIMA(1, d ,0), ARIMA(0, d ,1). If $d=0$, constant c is included; if $d \geq 1$, c is set to 0.
 - b. p and q are varied by ± 1 .
 - c. c is included/excluded.
 - d. 2(b) and (c) are repeated till AICc doesn't reduce further.



Source: <https://www.otexts.org/fpp/8/7>
Last accessed: November 25, 2016

Forecast using Auto ARIMA

```
Series: milestimeseries
ARIMA(1,0,1)(0,1,1)[12] with drift

Coefficients:
          ar1      ma1     sma1      drift
          0.9092  -0.2128  -0.7257  108456.69
  s.e.  0.0358   0.0873   0.0673   31546.77

sigma^2 estimated as 3.834e+12: log likelihood=-3056.9
AIC=6123.81  AICc=6124.13  BIC=6140.09
```

R code: auto.arima(milestimeseries,ic='aic')

Auto ARIMA

```
Series: milestimeseries
ARIMA(1,0,0)(0,1,1)[12] with drift
```

```
Coefficients:
          ar1      sma1      drift
          0.8527  -0.7254  109302.05
  s.e.  0.0378   0.0678   25762.84
```

```
sigma^2 estimated as 3.923e+12: log likelihood=-3059.73
AIC=6127.46  AICc=6127.67  BIC=6140.49
```

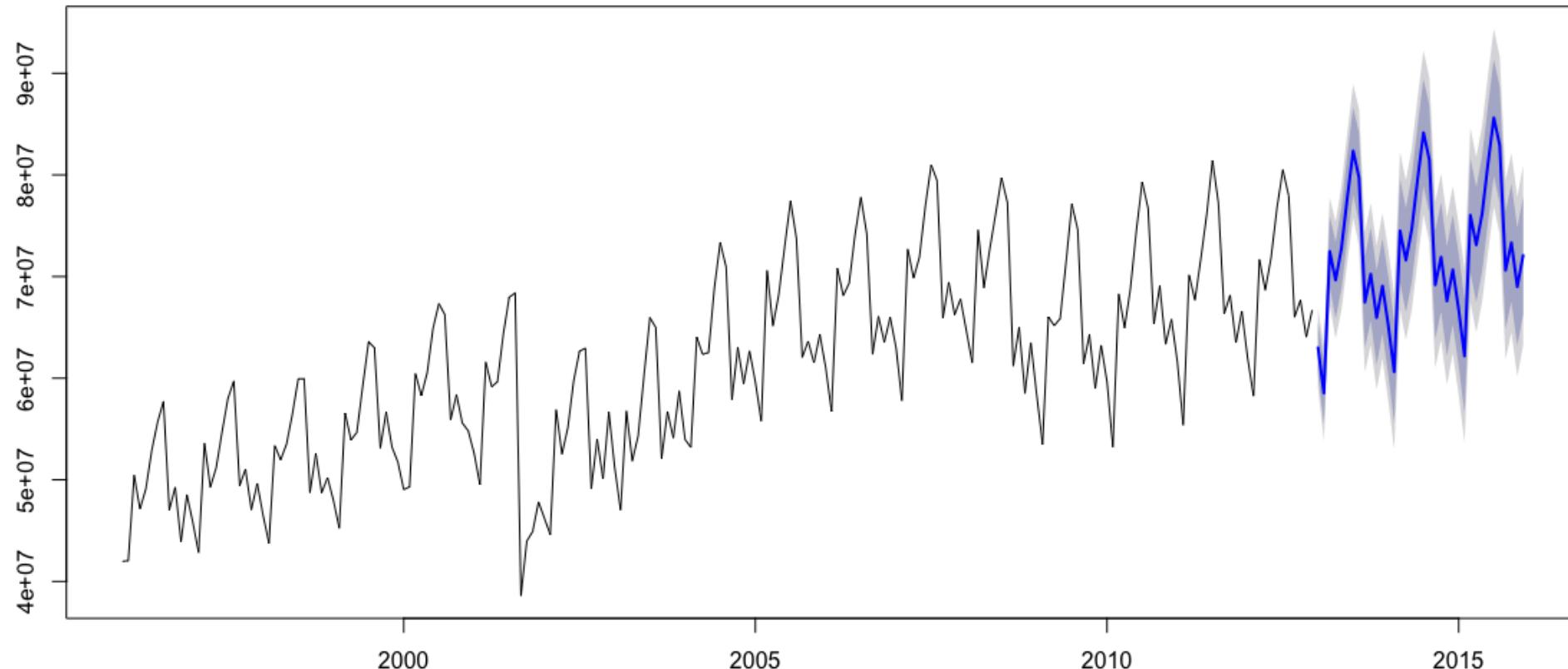
Manual ARIMA2



Forecast using Auto-ARIMA - RPM

R code: `forecast.Arima(milesAutoArima, h=36)`

Forecasts from ARIMA(1,0,1)(0,1,1)[12] with drift



Goodness of Fit

- MAE (Mean absolute error)

$$\frac{\sum |y_i - \hat{y}_i|}{n}$$

- MSE (Mean square error)

$$\frac{\sum (y_i - \hat{y}_i)^2}{n}$$

- RMSE (Root mean square error)

$$\sqrt{\frac{\sum (y_i - \hat{y}_i)^2}{n}}$$

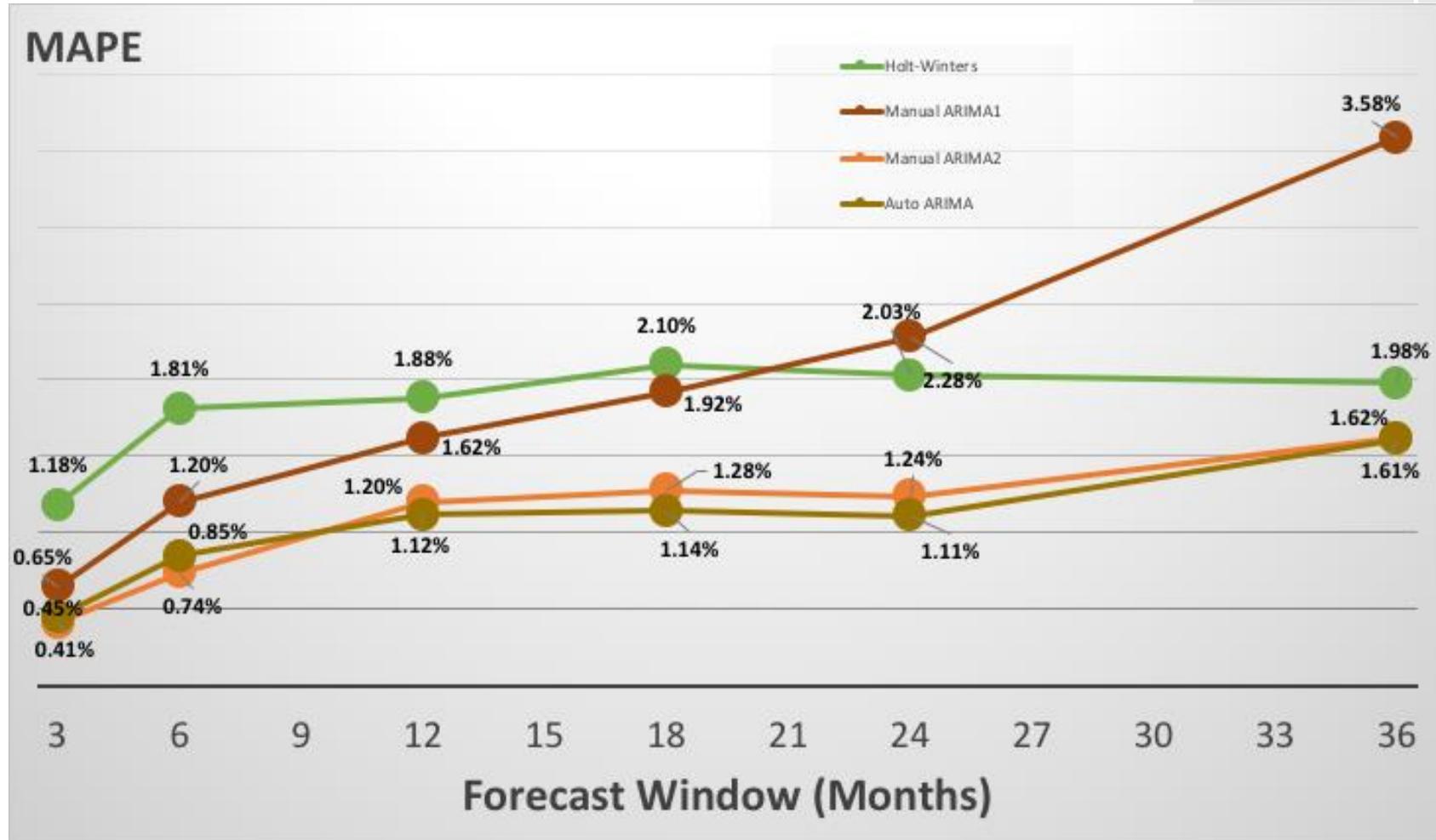
- MAPE (Mean absolute percent error)

$$\frac{1}{n} \left(\frac{\sum |y_i - \hat{y}_i|}{y_i} \right) * 100$$



Comparing Models - MAPE

Forecast window (months)	Holt-Winters	Manual ARIMA1	Manual ARIMA2	Auto ARIMA
3	1.18%	0.65%	0.41%	0.45%
6	1.81%	1.20%	0.74%	0.85%
12	1.88%	1.62%	1.20%	1.12%
18	2.10%	1.92%	1.28%	1.14%
24	2.03%	2.28%	1.24%	1.24%
36	1.98%	3.58%	1.62%	1.61%



Resources

Linear Regression

- <http://people.duke.edu/~rnau/regintro.htm>
- <https://onlinecourses.science.psu.edu/stat501/node/250/>

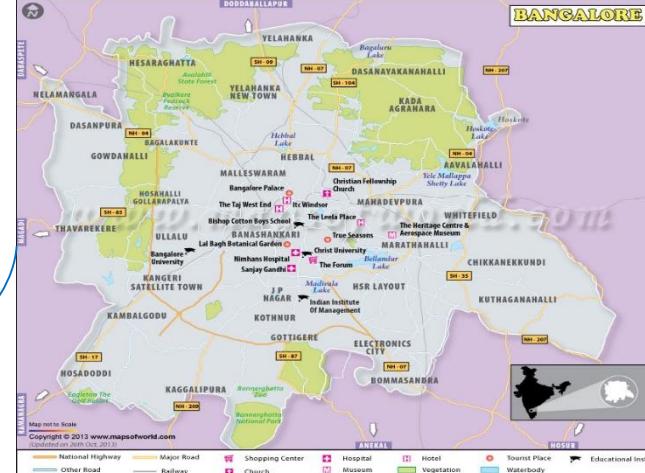
Logistic Regression

- <https://onlinecourses.science.psu.edu/stat504/node/149/>

Time Series Forecasting

- <https://www.otexts.org/fpp2/> Forecasting: Principles and Practice
- <http://people.duke.edu/~rnau/411home.htm>
- <https://onlinecourses.science.psu.edu/stat510/node/41/>





HYDERABAD

2nd Floor, Jyothi Imperial, Vamsiram Builders, Old Mumbai Highway, Gachibowli, Hyderabad - 500 032
 +91-9701685511 (Individuals)
 +91-9618483483 (Corporates)

BENGALURU

Floors 1-3, L77, 15th Cross Road, 3A Main Road, Sector 6, HSR Layout, Bengaluru – 560 102
 +91-9502334561 (Individuals)
 +91-9502799088 (Corporates)

Social Media

- Web: <http://www.insofe.edu.in>
- Facebook: <https://www.facebook.com/insofe>
- Twitter: <https://twitter.com/Insofeedu>
- YouTube: <http://www.youtube.com/InsofeVideos>
- SlideShare: <http://www.slideshare.net/INSOFE>
- LinkedIn: <http://www.linkedin.com/company/international-school-of-engineering>

This presentation may contain references to findings of various reports available in the public domain. INSOFE makes no representation as to their accuracy or that the organization subscribes to those findings.