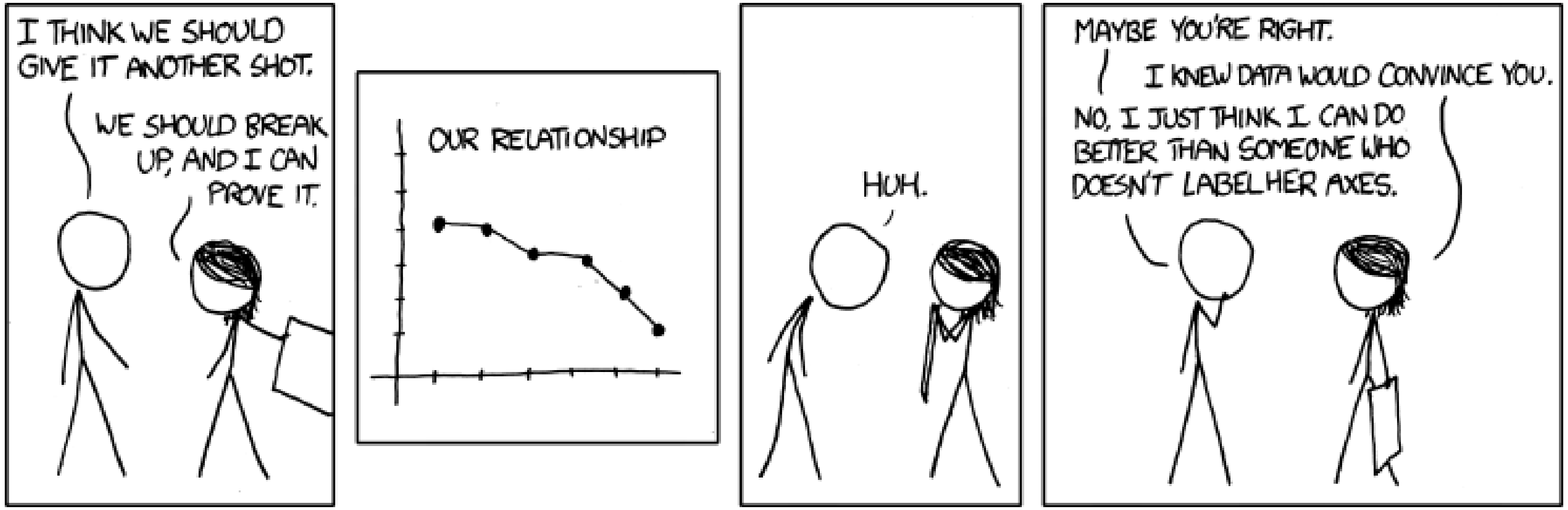
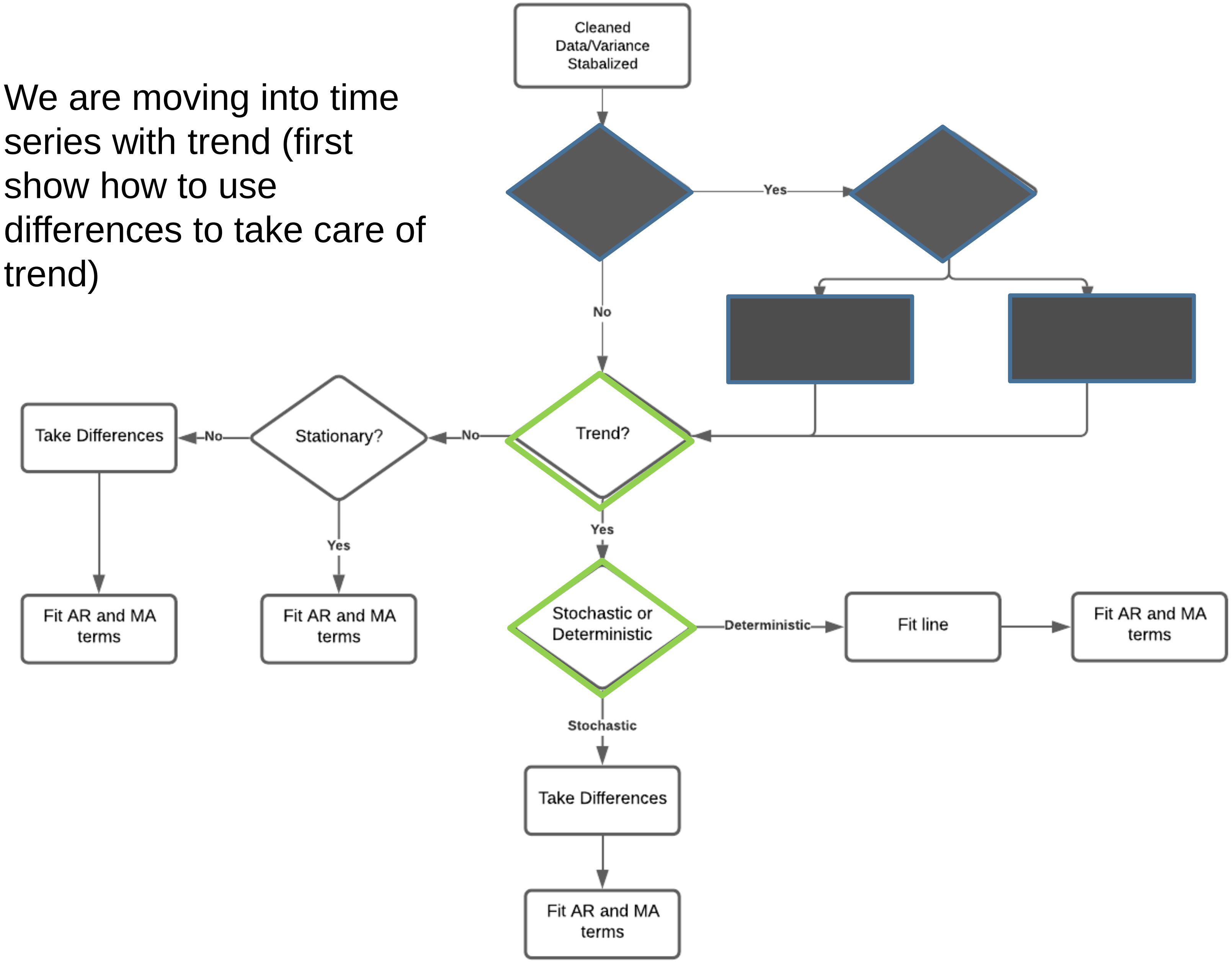


Source: xkcd comics and statistical thinking

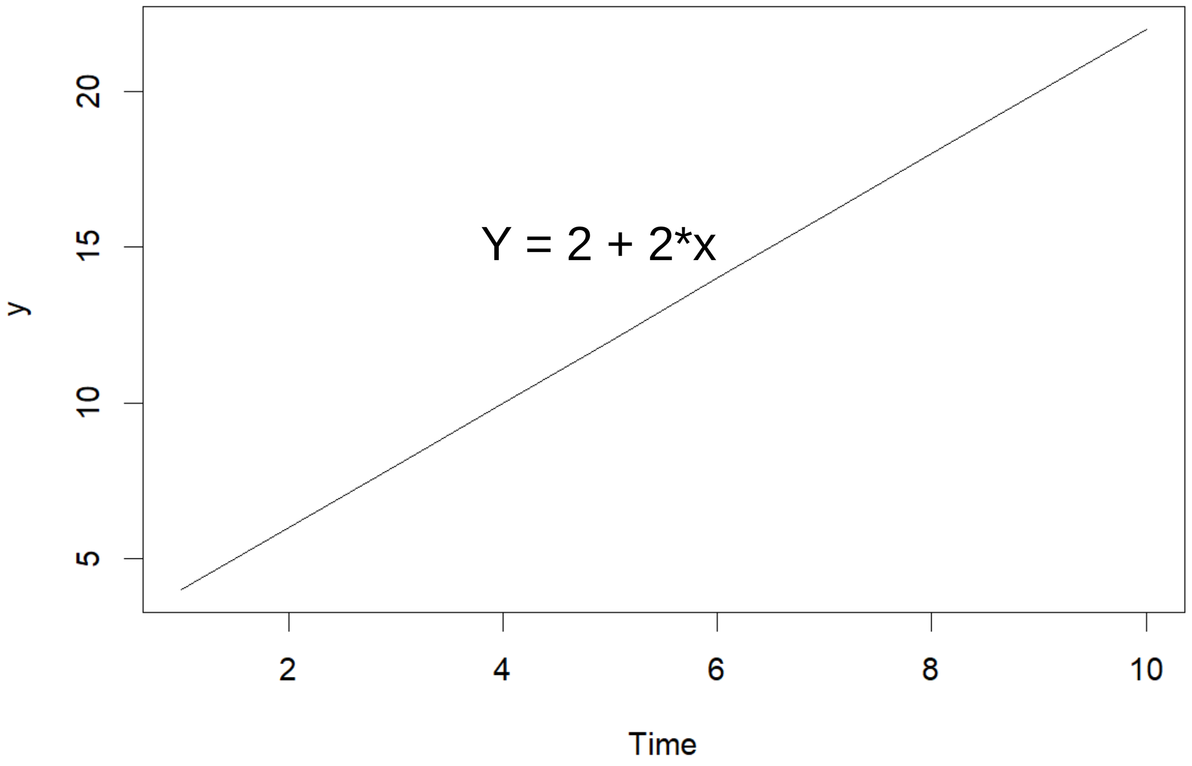
TRENDING TIME SERIES

We are moving into time series with trend (first show how to use differences to take care of trend)



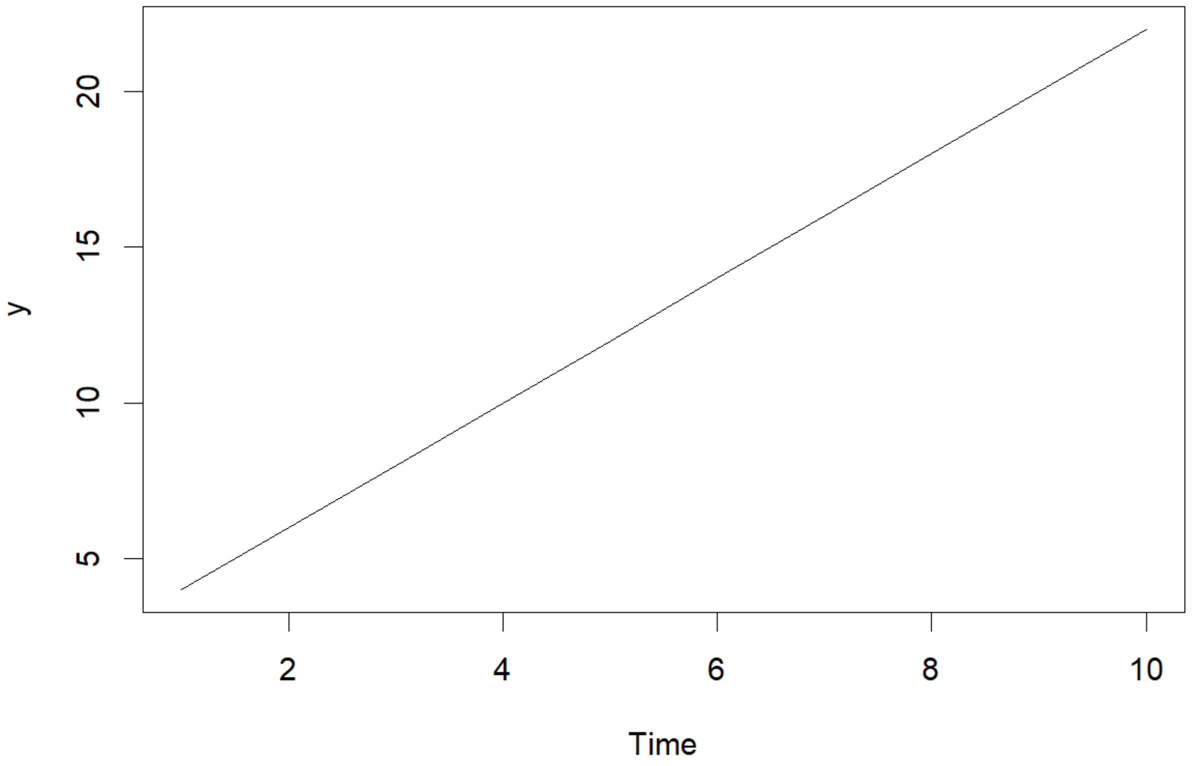
USING DIFFERENCES TO HANDLE TREND

Trends



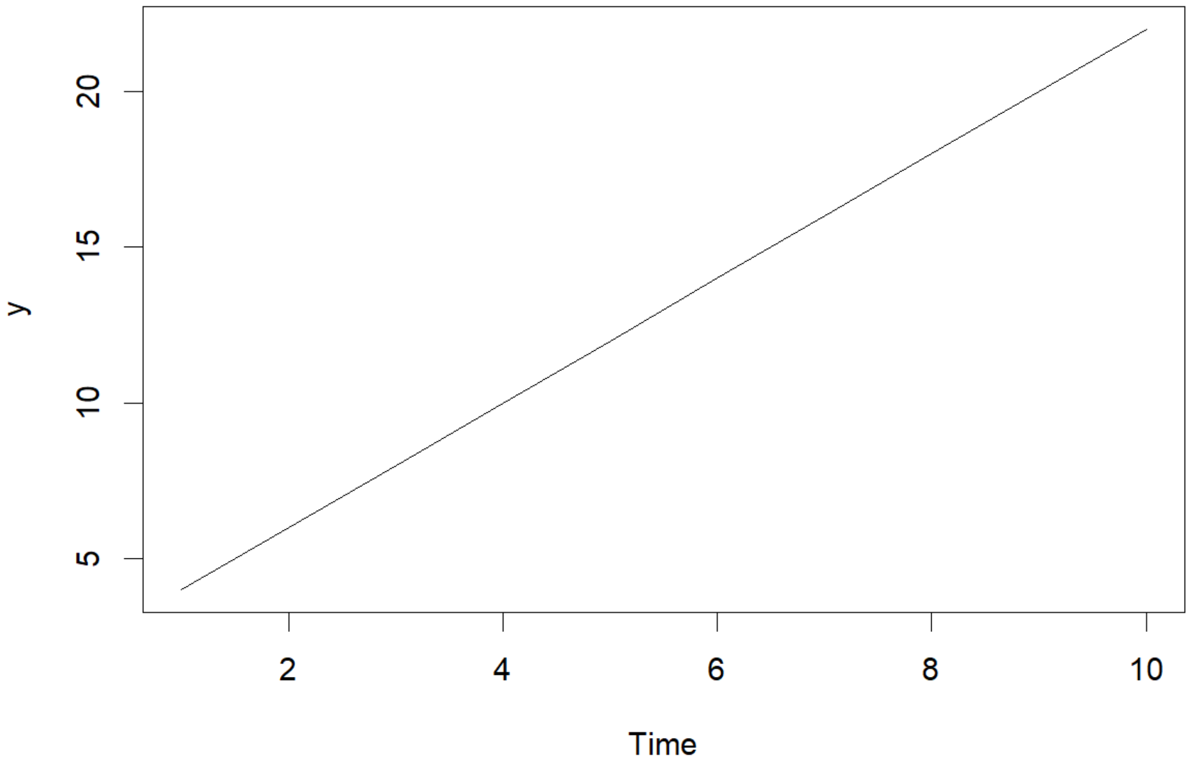
Line with slope = 2 (y-intercept =2)
At x=0, y = 2
At x=1, y=4
At x=2, y=6
At x=3, y=8
Etc….

Trends



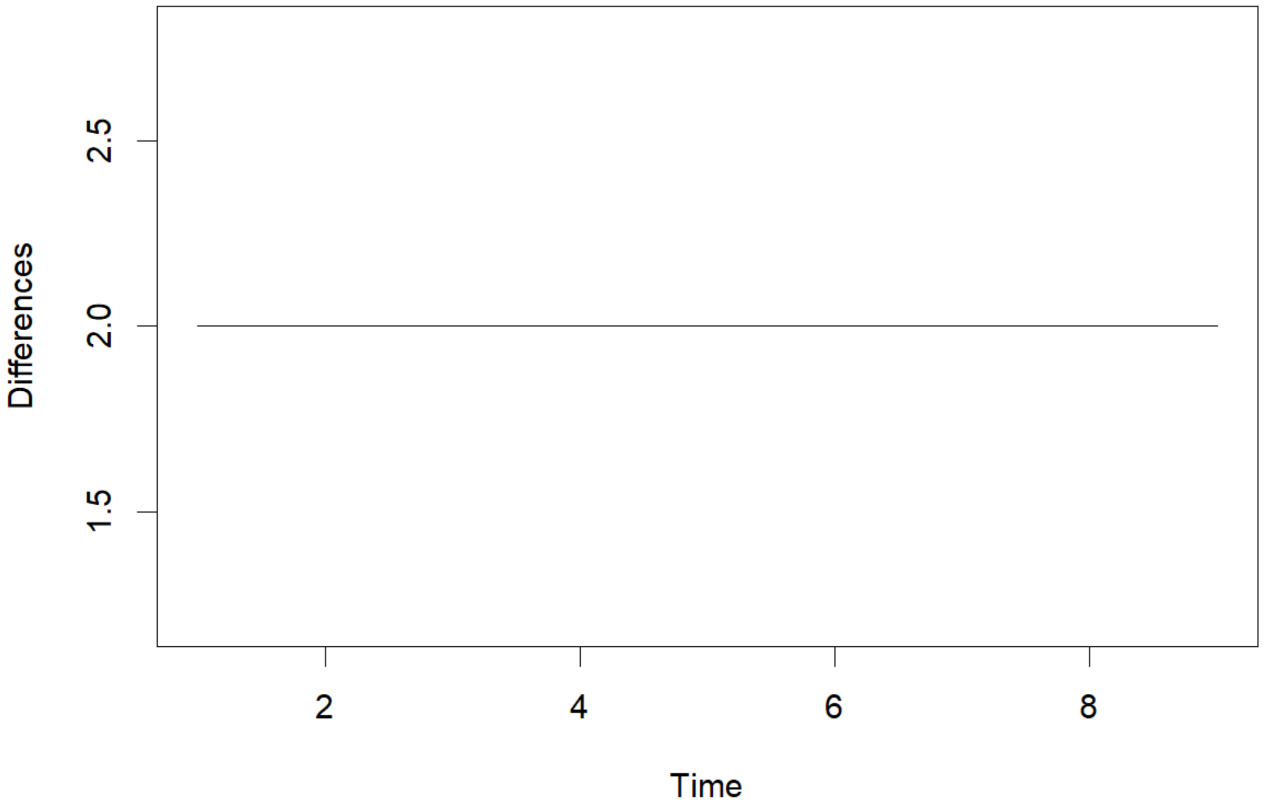
What happens when we difference y?
22-20=2
20-18=2
18-16=2
Etc.

Trends



So, if we think about Y as a time series with a trend, when we take the difference, we remove that trend

Differenced series:



Differencing to remove a trend

- So one way we can deal with a trending time series is take the difference and then try to model the ARMA terms on the differenced series (same thing we did with a Random Walk!!)
- In fact, there is a Random Walk with drift defined as:

$$Y_t = \omega + Y_{t-1} + \varepsilon_t$$

Where ω is the "drift" of the Random Walk (think drifting up or drifting down)

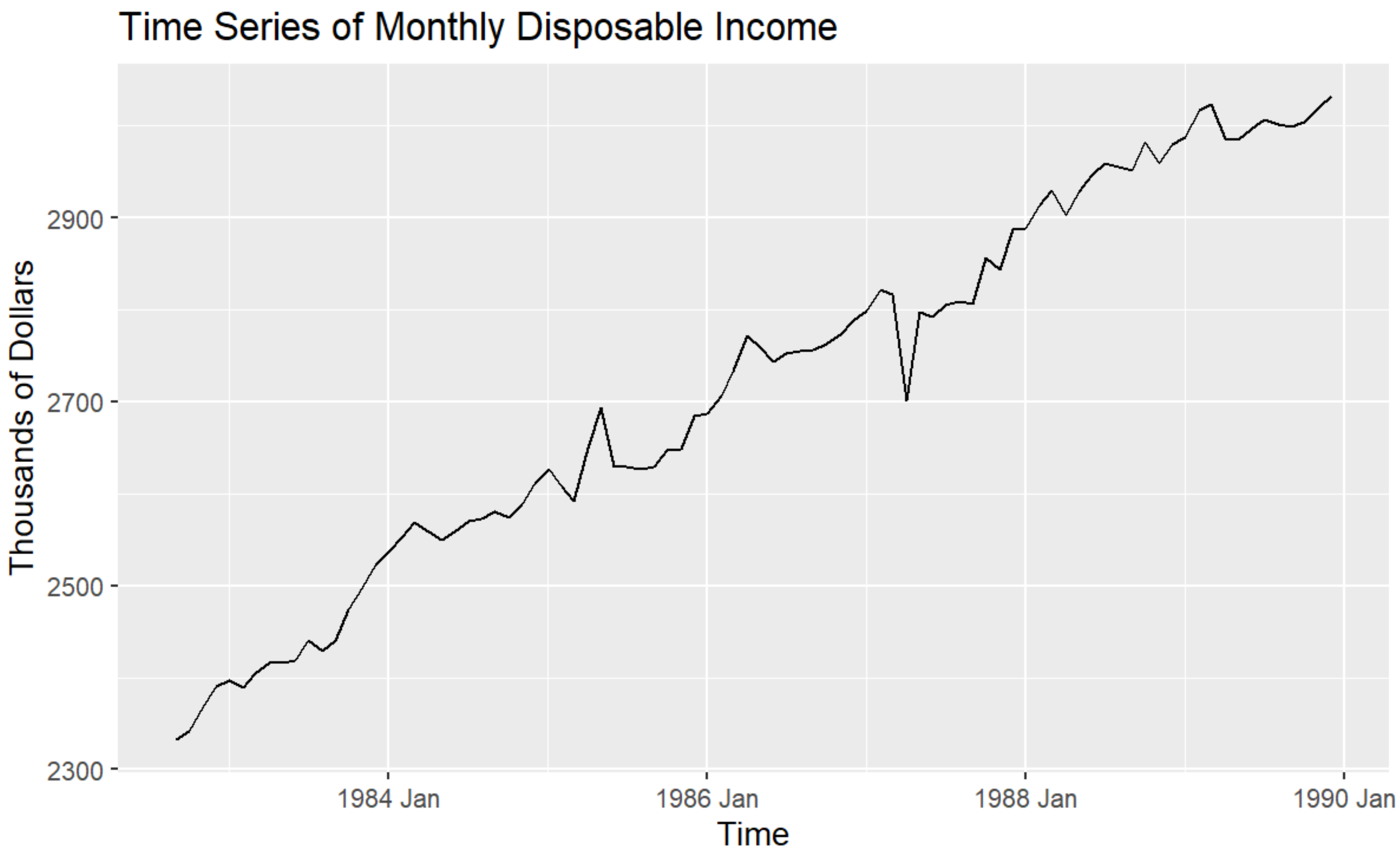
NOTE: IF you use an automated search procedure and the time series has a trend, IT will find that d = 1 or d=2!

Procedure:

- Clean your time series data set
- If you notice that your time series has a trend, it is NOT stationary (you cannot start fitting AR and MA terms yet)
- Take differences of your time series data and look at ACF and PACF plots of your differenced data to try some models
- Try some automated models
- Pick best models and see if you have white noise and can forecast well
- Choose your best model

Example: Consumer spending example

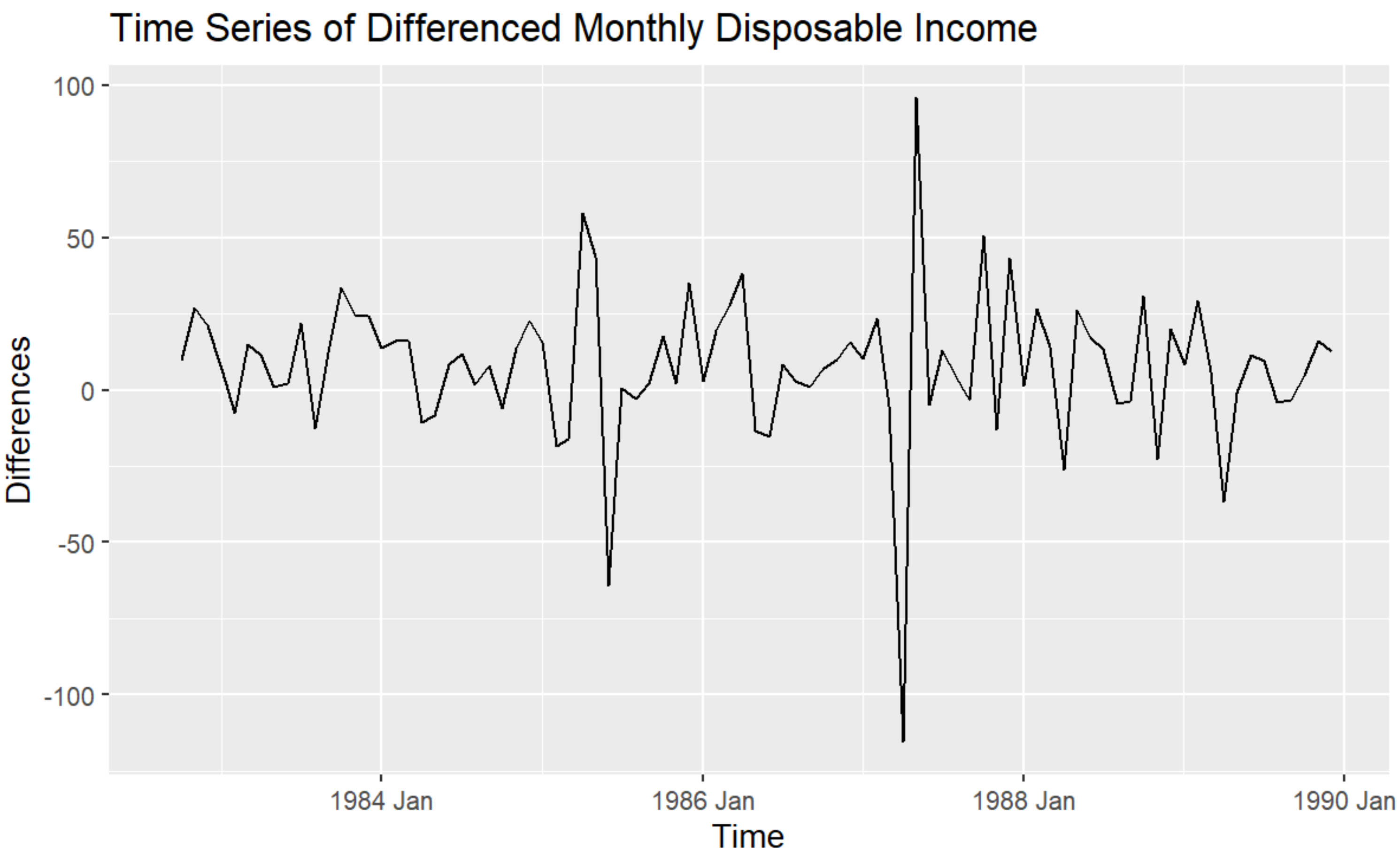
- The consumer data set has disposable consumer spending in thousands of dollars from September 1982 through June 1990



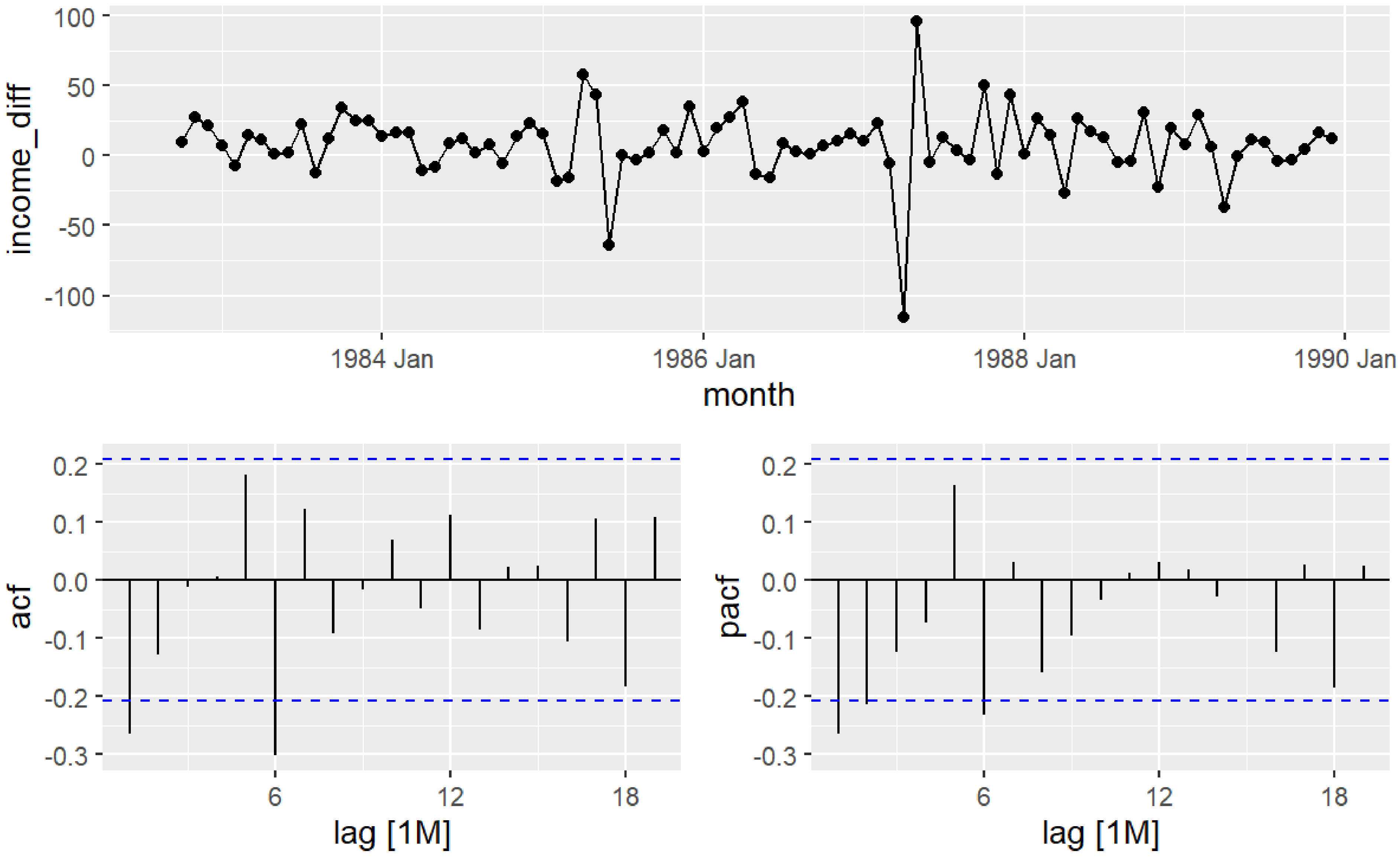
```
ndiffs(consume_train$Disposable_income)
```

```
[1] 1
```

```
consume_train<-consume.ts %>% filter(year(date2)<1990)
consume_train<- consume_train %>%   mutate(income_diff =
difference(Disposable_income))
autoplot(consume_train,income_diff)+labs(title="Time Series of Differenced Monthly
Disposable Income", x="Time", y="Differences")
```



```
consume_train %>% gg_tsdisplay(income_diff,plot_type = 'partial')
```



```
consume_model <-consume_train %>%
  model(ar1 = ARIMA(Disposable_income ~ pdq(1,1,0) + PDQ(0,0,0)),
  ma1 = ARIMA(Disposable_income ~ pdq(0,1,1) + PDQ(0,0,0)),
  ar6 = ARIMA(Disposable_income ~ pdq(6,1,0) + PDQ(0,0,0)),
  ma6 = ARIMA(Disposable_income ~ pdq(0,1,6) + PDQ(0,0,0)),
  search1 = ARIMA(Disposable_income),
  search2 = ARIMA(Disposable_income,stepwise = F))
  consume_model2<-as.data.frame(consume_model)
t(consume_model2)
```

```
        [,1]
ar1     ARIMA(1,1,0) w/ drift
ma1     ARIMA(0,1,1) w/ drift
ar6     ARIMA(6,1,0) w/ drift
ma6     ARIMA(0,1,6)
search1 ARIMA(0,1,1) w/ drift
search2 ARIMA(0,1,1) w/ drift
```

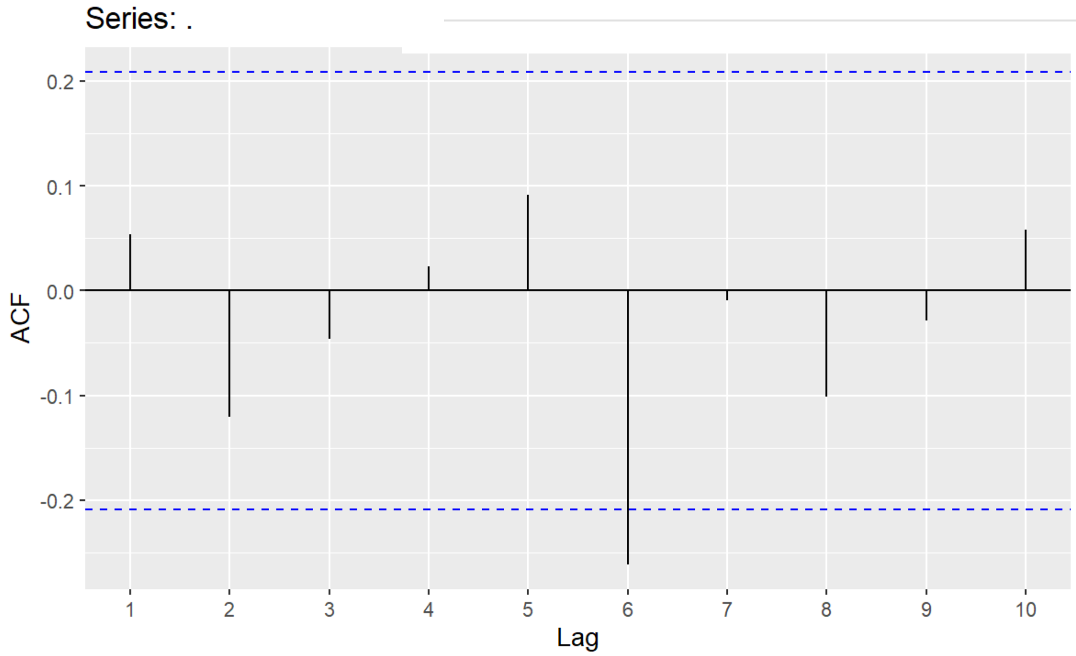
```
glance(consume_model) %>% arrange(AICc) %>% select(.model:BIC)
```

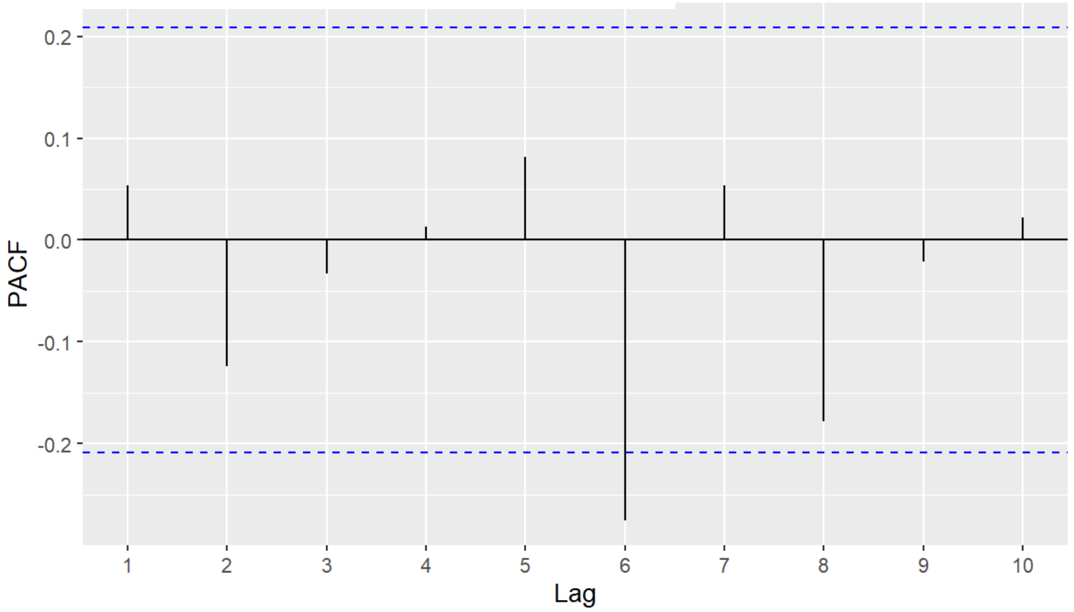
.model <chr>	sigma2 <dbl>	log_lik <dbl>	AIC <dbl>	AICc <dbl>	BIC <dbl>
ma1	545.6705	-396.6613	799.3225	799.6117	806.7202
search1	545.6705	-396.6613	799.3225	799.6117	806.7202
search2	545.6705	-396.6613	799.3225	799.6117	806.7202
ar6	520.1097	-392.1708	800.3416	802.1877	820.0688
ar1	570.8393	-398.5659	803.1318	803.4209	810.5295
ma6	610.4192	-399.7644	813.5287	814.9464	830.7901

MA(1)

```
consume_model %>% select(ma1) %>% residuals() %>% ggAcf(lag.max = 10)
consume_model %>% select(ma1) %>% residuals() %>% ggPacf(lag.max = 10)
augment(consume_model) %>% filter(.model=='ma1') %>% features(.innov,ljung_box, lag=10, dof = 1)
```

.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
ma1	10.68682	0.2977881





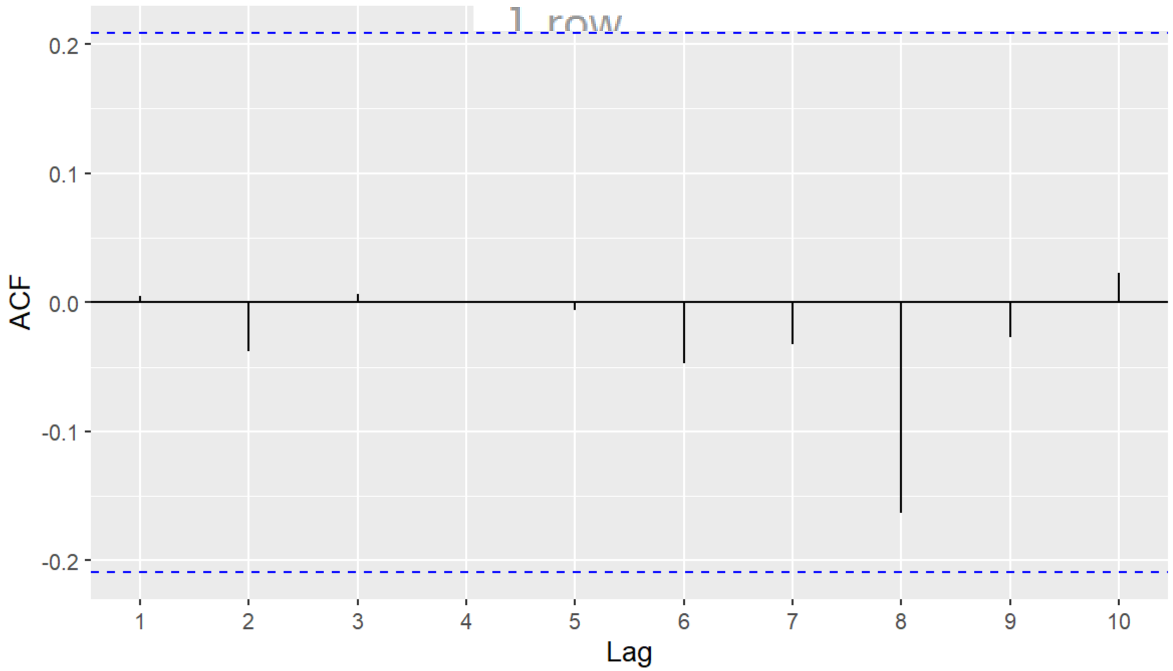
AR(6)

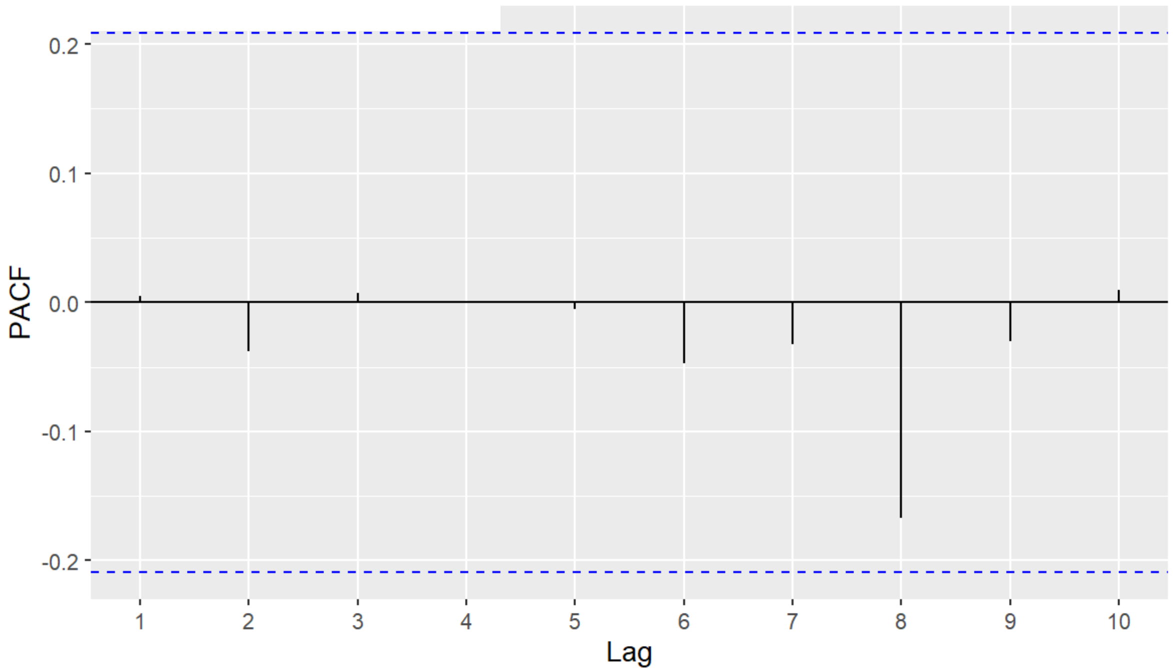
```
consume_model %>% select(ar6) %>% residuals() %>% ggAcf(lag.max = 10)
consume_model %>% select(ar6) %>% residuals() %>% ggPacf(lag.max = 10)
augment(consume_model) %>% filter(.model=='ar6') %>% features(.innov,ljung_box, lag=10, dof = 6)
```

.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
ar6	3.214432	0.5226031

1 row

Series: .

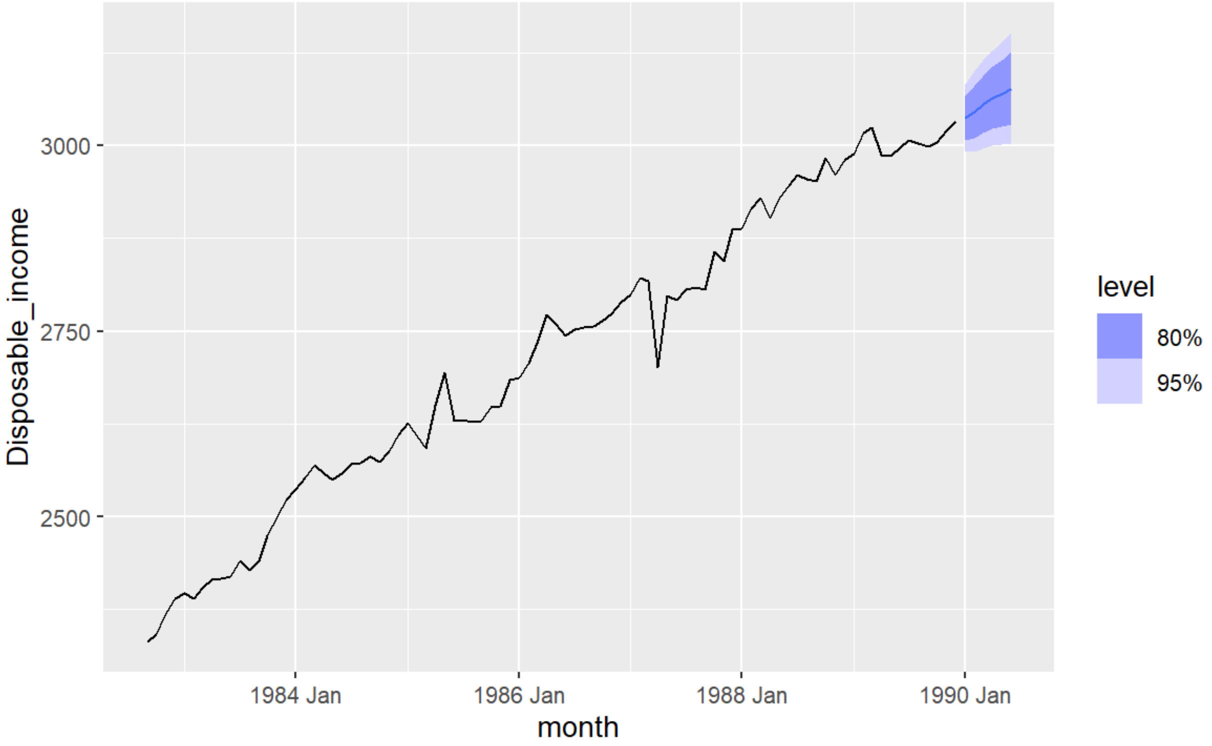




```
pred_ar6 <- consume_model %>% select(ar6) %>%    fabletools::forecast(h=6)
error_ar6 <- consume$Disposable_income[89:94] - pred_ar6$.mean
MAPE_ar6 <-mean(abs(error_ar6/consume$Disposable_income[89:94]))
MAE_ar6 <- mean(abs(error_ar6))
```

```
> MAPE_ar6
[1] 0.006767987
> MAE_ar6
[1] 20.56257
```

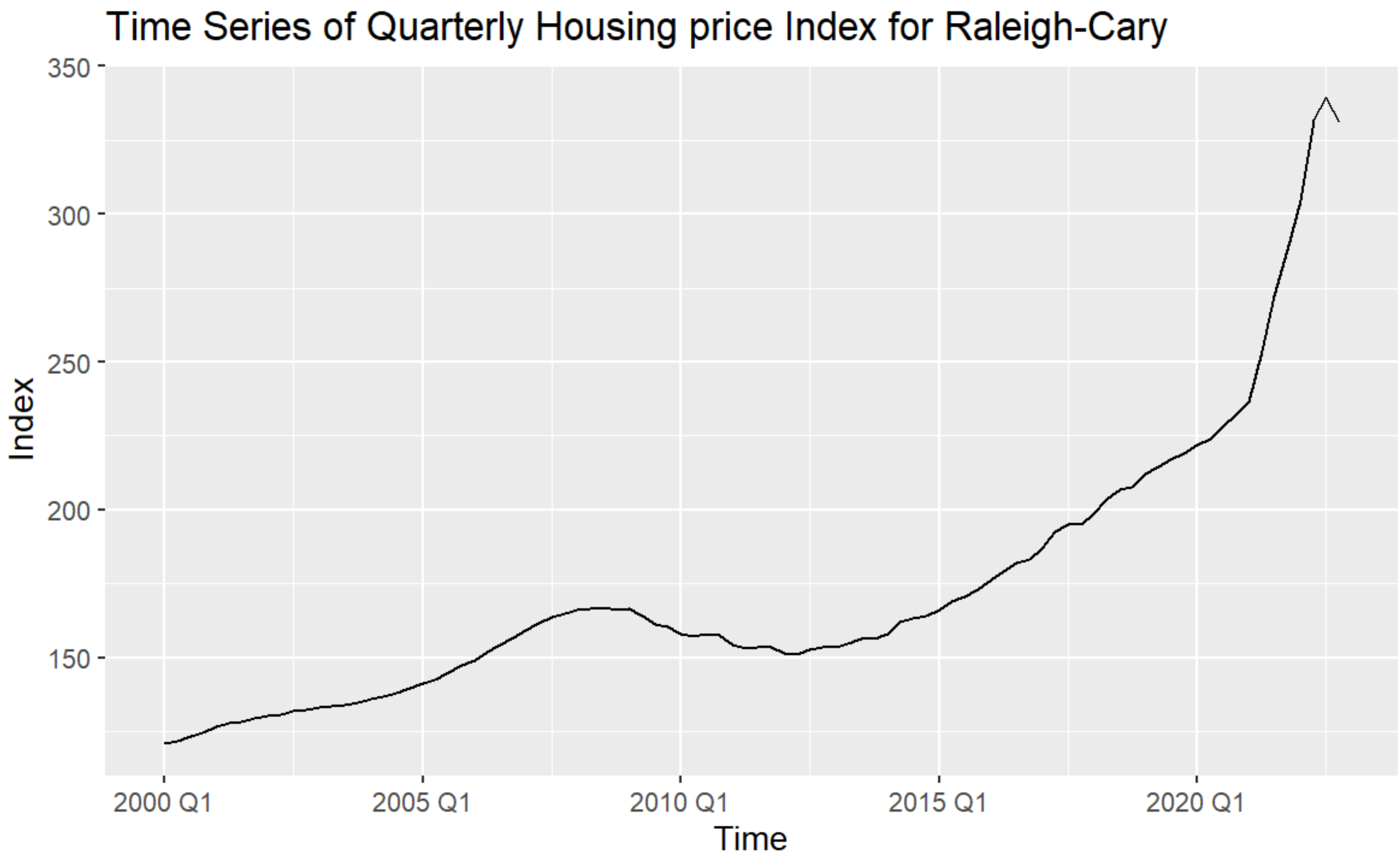
```
> MAPE_ma1
[1] 0.006698975
> MAE_ma1
[1] 20.3521
```



Example 2: Raleigh housing price index

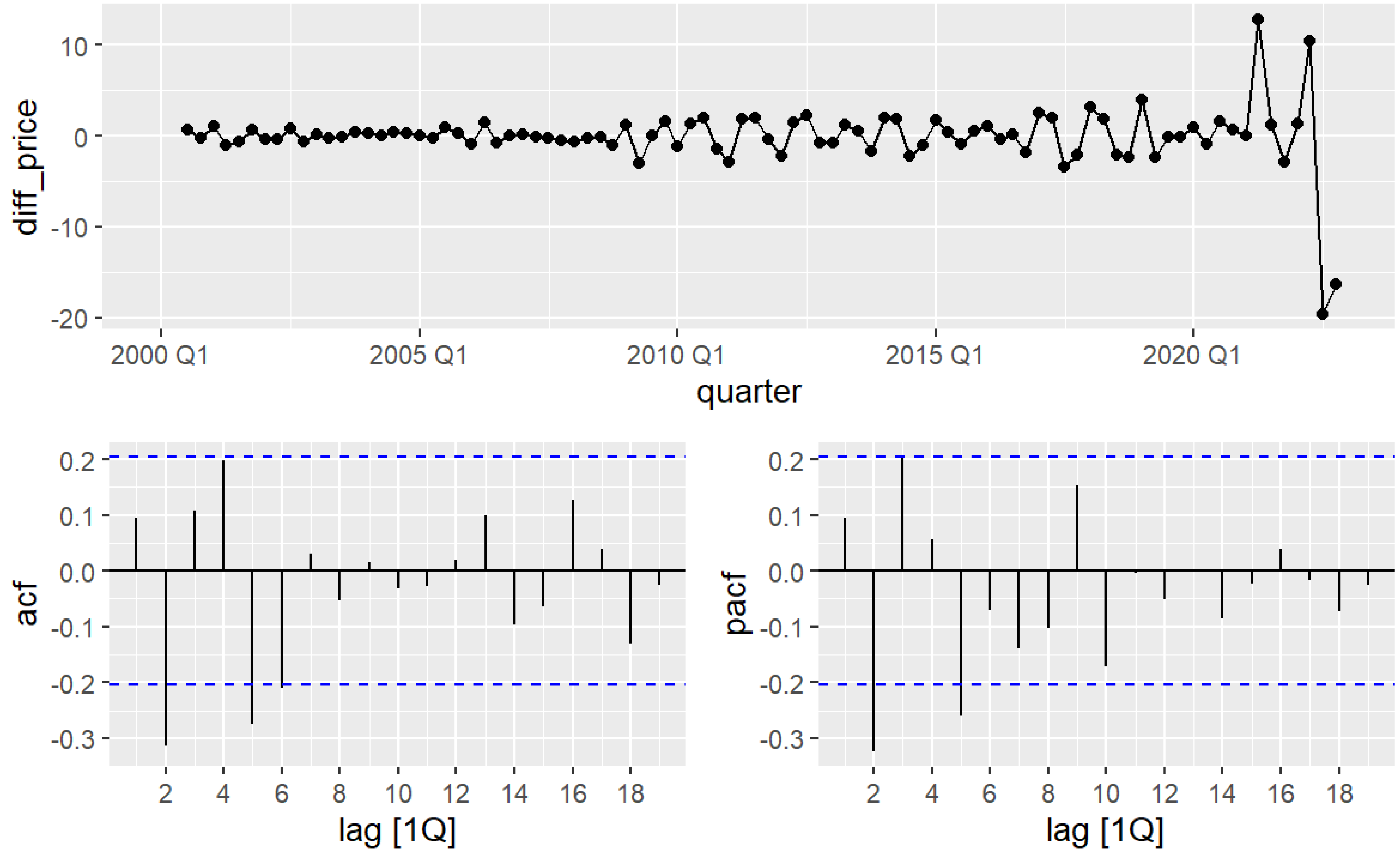
- All-transaction housing price index for Raleigh-Cary

```
Raleigh_train %>%
    features(price_index,unitroot_ndiffs)
```



ndiffs <int>
2

1 row



diff_price
10
0
-10
-20
2000 Q1
2005 Q1
2010 Q1
2015 Q1
2020 Q1
quarter
acf
0.2
0.1
0.0
-0.1
-0.2
-0.3
2
4
6
8
10
12
14
16
18
lag [1Q]
pacf
0.2
0.1
0.0
-0.1
-0.2
-0.3
2
4
6
8
10
12
14
16
18
lag [1Q]

Model search

```
Raleigh_model <-Raleigh_train %>%
  model(ma5 = ARIMA(price_index ~ pdq(0,2,5)+ PDQ(0,0,0)+0),
  ar2 = ARIMA(price_index ~ pdq(2,2,0)+ PDQ(0,0,0)+0),
  ma2 = ARIMA(price_index ~ pdq(0,2,2)+ PDQ(0,0,0)+0),
  search1 = ARIMA(price_index~PDQ(0,0,0)),
  search2 = ARIMA(price_index,stepwise = FALSE) )
Raleigh_model2<-as.data.frame(Raleigh_model)
t(Raleigh_model2)
```

```
        [,1]
ma5     ARIMA(0,2,5)
ar2     ARIMA(2,2,0)
ma2     ARIMA(0,2,2)
search1 ARIMA(1,2,3)
search2 ARIMA(0,2,5)
```

```
glance(Raleigh_model) %>% arrange(AICc) %>% select(.model:BIC)
```

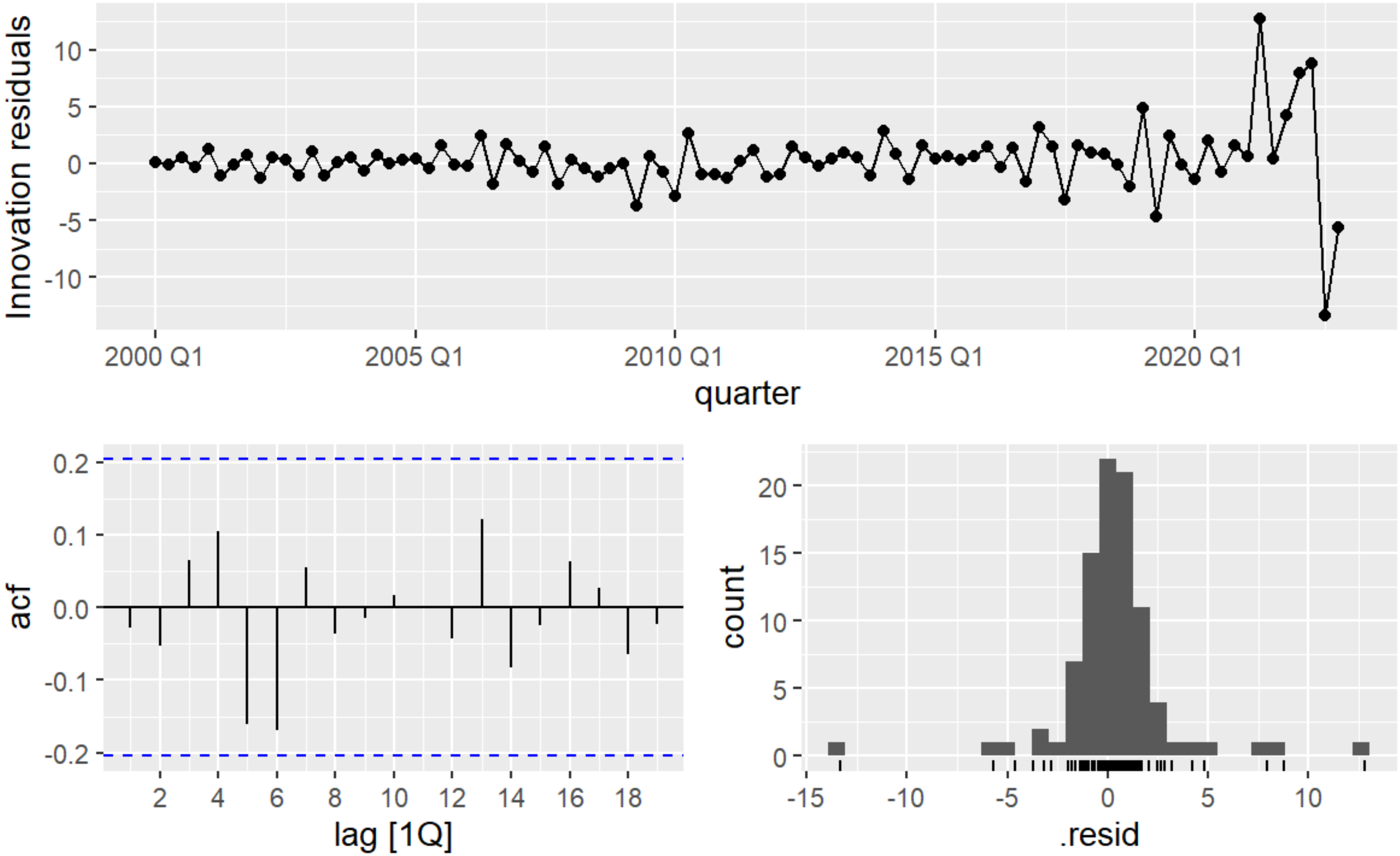
.model <chr>	sigma2 <dbl>	log_lik <dbl>	AIC <dbl>	AICc <dbl>	BIC <dbl>
ma5	7.646308	-219.4104	450.8209	451.8329	465.8197
search2	7.646308	-219.4104	450.8209	451.8329	465.8197
search1	8.355466	-222.5903	455.1805	455.8948	467.6795
ar2	9.501725	-228.7792	463.5585	463.8375	471.0579
ma2	10.083476	-231.2456	468.4911	468.7702	475.9906

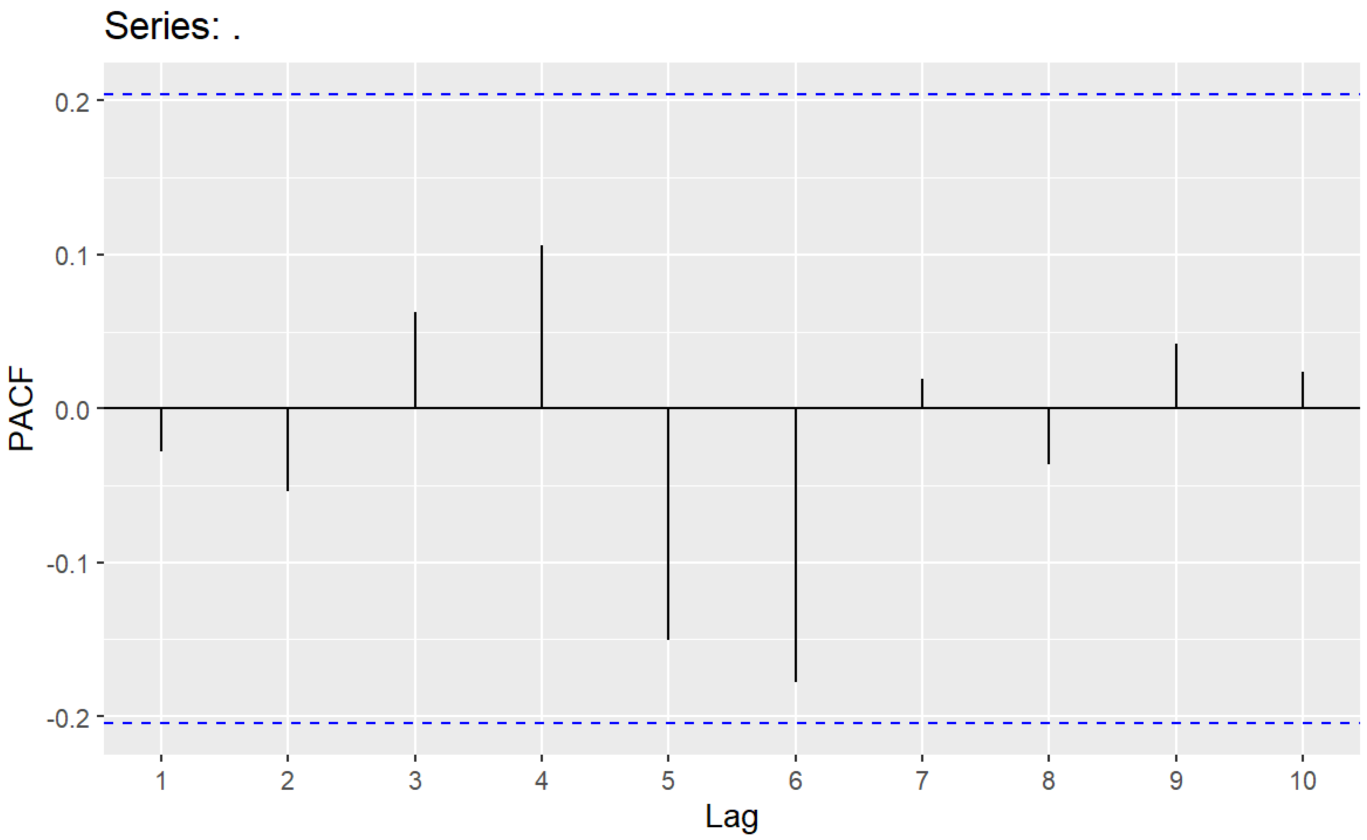
Notes:

- Some of the original AR and ARIMA models were not stationary, so I removed them from the list
- Search1 wanted to do a seasonal ARIMA, so I had to tell it NO (PDQ(0,0,0))
- The ARIMA(0,2,5) model did not have white noise (some spikes beyond the confidence intervals and Ljung-Box p-value was less than 0.02).
- Therefore, showing results for Search1 (ARIMA(1,2,3))

```
Raleigh_model %>% select(search1) %>% residuals() %>% ggAcf(lag.max = 10)
Raleigh_model %>% select(search1) %>% residuals() %>% ggPacf(lag.max = 10)
Raleigh_model %>% select(search1) %>% gg_tsresiduals()
```

```
augment(Raleigh_model) %>% filter(.model=='search1') %>% features(.innov,ljung_box, lag=10, dof = 4)
```

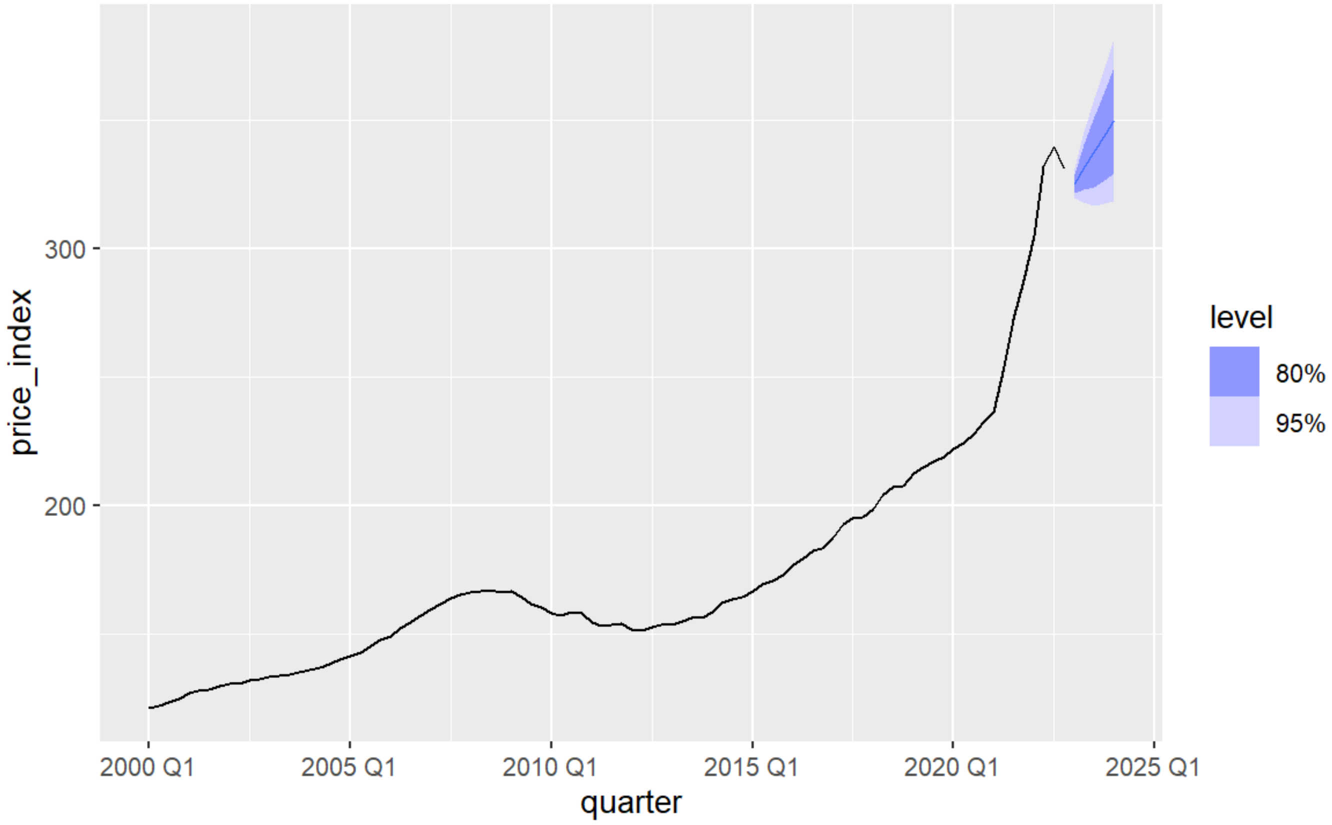


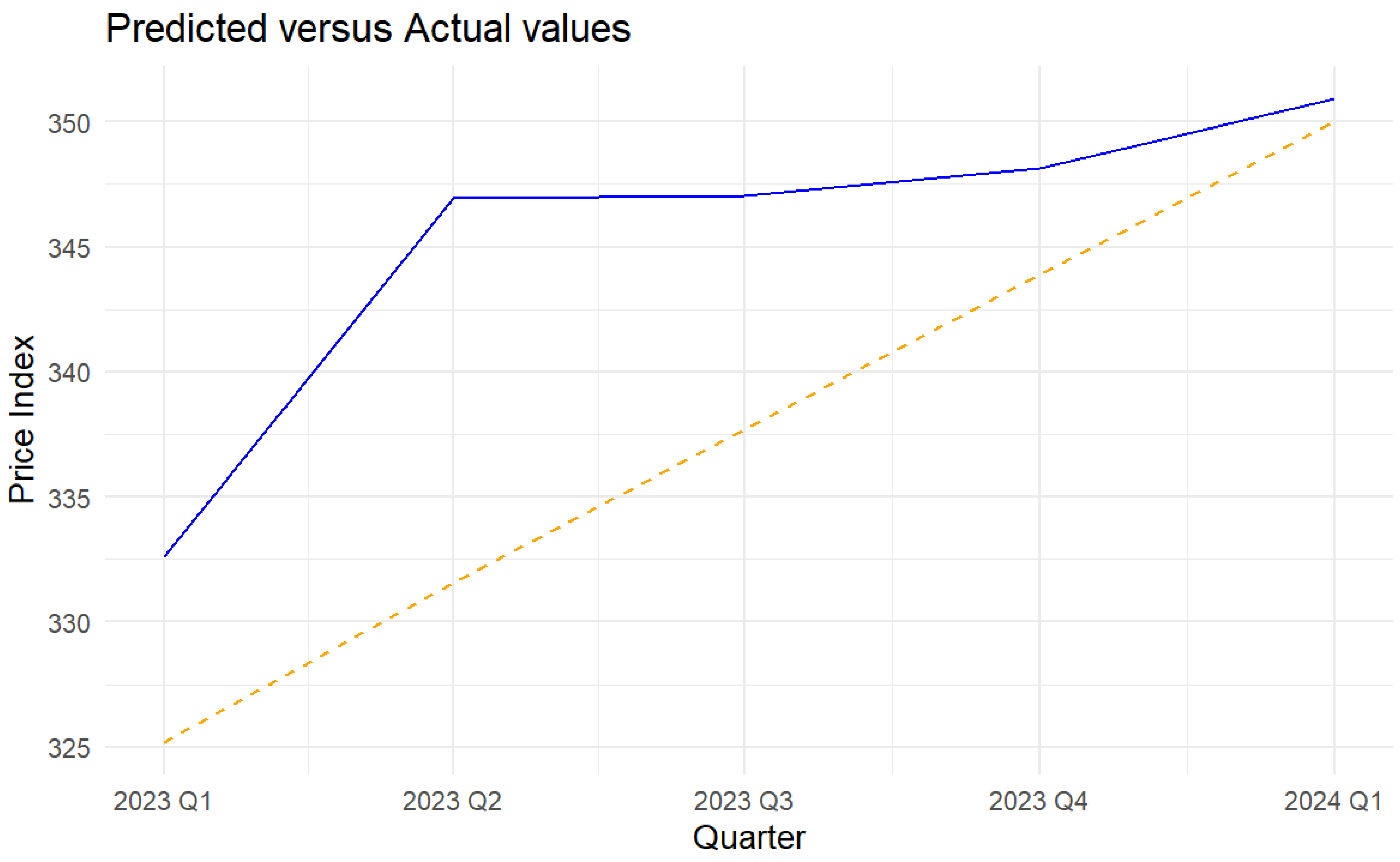


.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
search1	7.755857	0.2565411

```
pred_arima123 <- Raleigh_model %>% select(search1) %>%  fabletools::forecast(h=5)
error_arima123 <- Raleigh.ts$price_index[93:97] - pred_arima123$.mean
MAPE_arima123 <-mean(abs(error_arima123/Raleigh.ts$price_index[93:97]))
MAE_arima123 <- mean(abs(error_arima123))
```

```
> MAPE
[1] 0.06067555
> MAE
[1] 5.986266
```




Predicted versus Actual values
Price Index
350
345
340
335
330
325
2023 Q1
2023 Q2
2023 Q3
2023 Q4
2024 Q1
Quarter

ARIMAX FOR TREND

Trending time series

- If a time series is trending, there is another way of modeling the trend (besides differencing)
- We could fit a linear regression to the time series data set (x = Time, y = value)
- When we fit a linear regression model to time series data, this is referred to as ARIMAX (ARIMA with an “X” variable….in this case X=time)
- In doing an ARIMAX for trend, we are fitting a linear model with time and then fitting an ARMA to the residuals:

$$Y_t = \beta_0 + \beta_1 t + \eta_t$$
$$\eta_t = ARMA(p, 0, q)$$

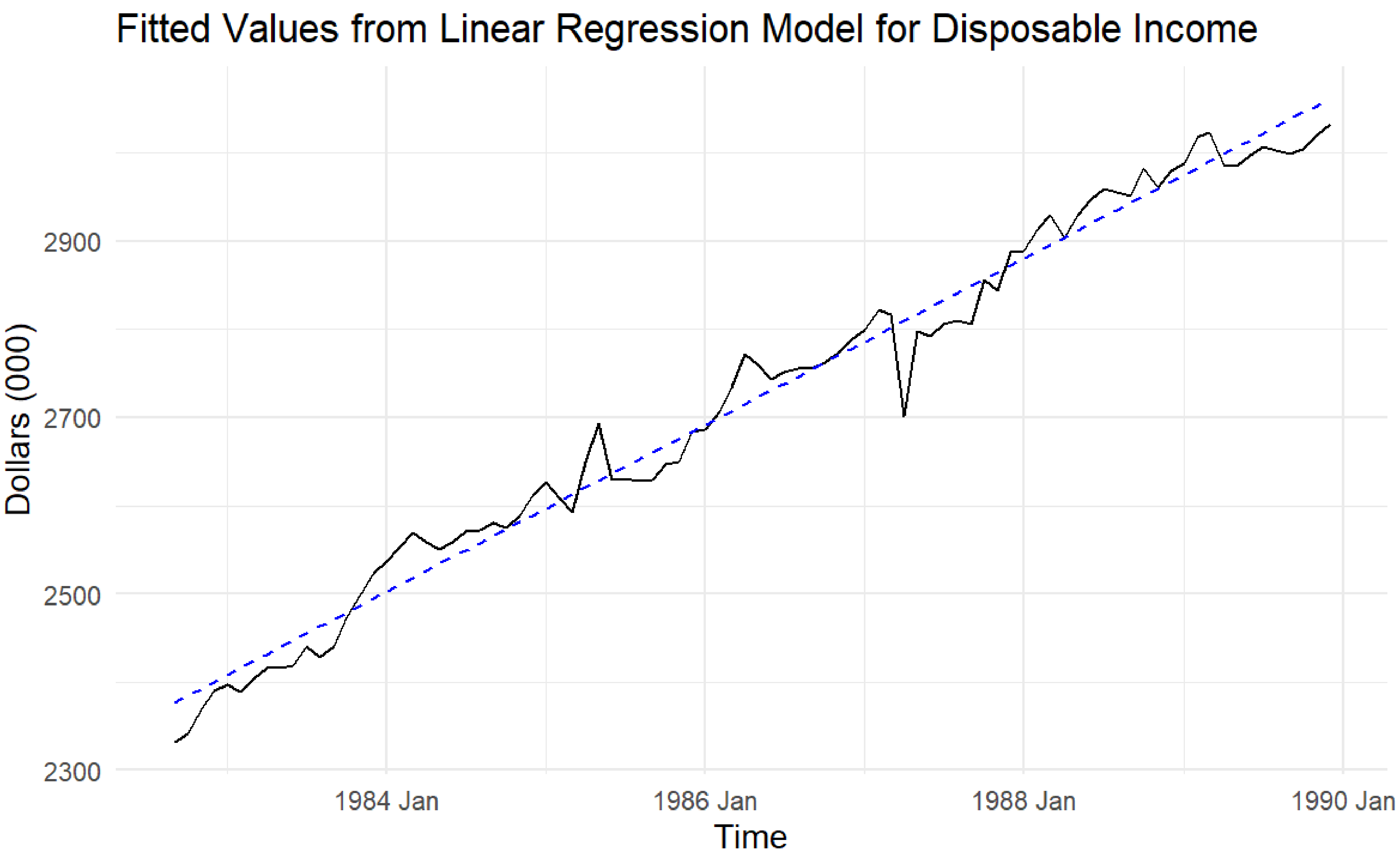
- After fitting the ARMA to the residuals, what is left should be white noise

Consumer spending:

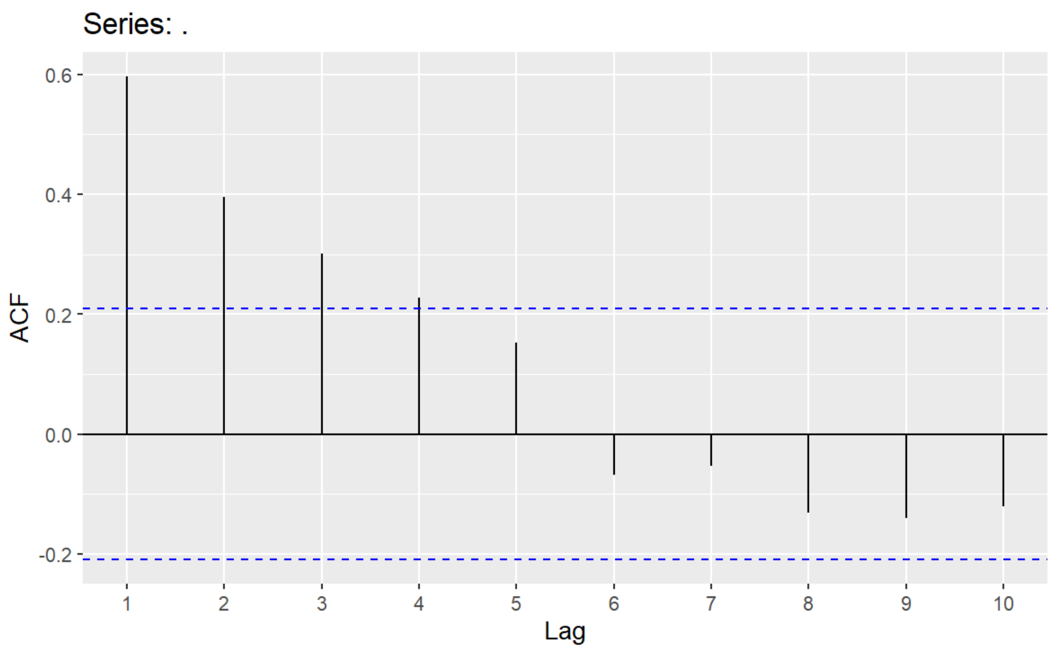
```
consume_linear <-consume_train %>%  model(trend1 = ARIMA(Disposable_income~ trend() + pdq(0,0,0) + PDQ(0,0,0)+1) )
report(consume_linear)
```

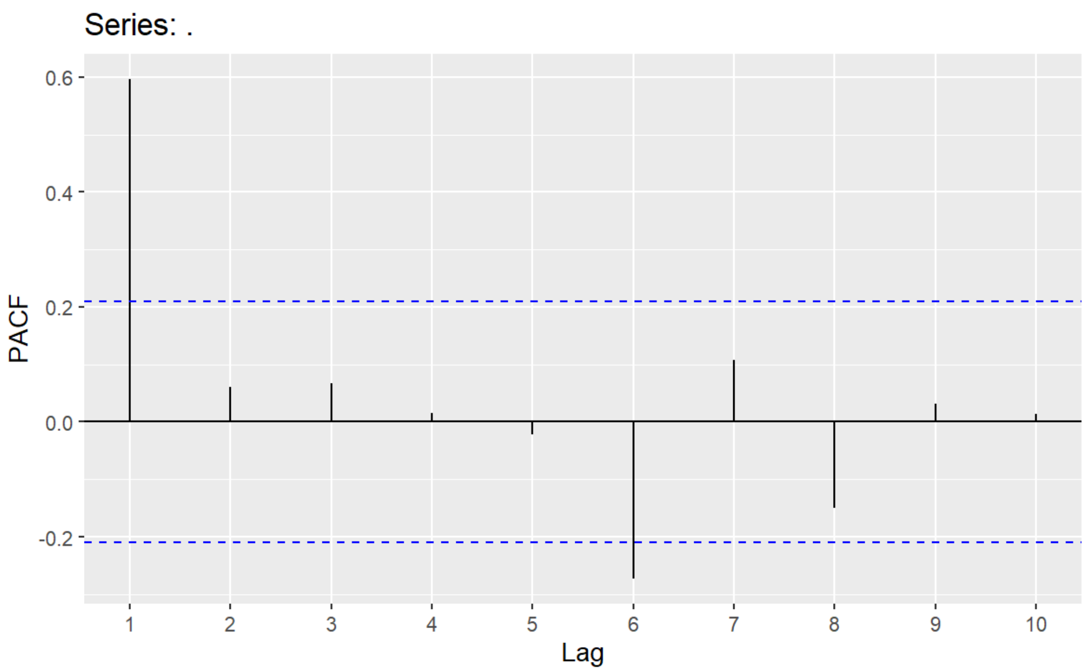
```
Series: Disposable_income
Model: LM w/ ARIMA(0,0,0) errors

Coefficients:
      trend()  intercept
       7.8743  2368.4594
s.e.   0.1168     5.9841
```


Fitted Values from Linear Regression Model for Disposable Income
Dollars (000)
2900
2700
2500
2300
1984 Jan
1986 Jan
1988 Jan
1990 Jan
Time

```
consume_linear %>% residuals() %>% ggAcf(lag.max = 10)
consume_linear %>% residuals() %>% ggPacf(lag.max = 10)
```





```
consume_linear <-consume_train %>%  model(
  trend1 = ARIMA(Disposable_income~ trend() + pdq(6,0,0) + PDQ(0,0,0)+1),
  trend2 = ARIMA(Disposable_income ~ trend() + PDQ(0,0,0) +1))
consume_linear2<-as.data.frame(consume_linear)
t(consume_linear2)
```

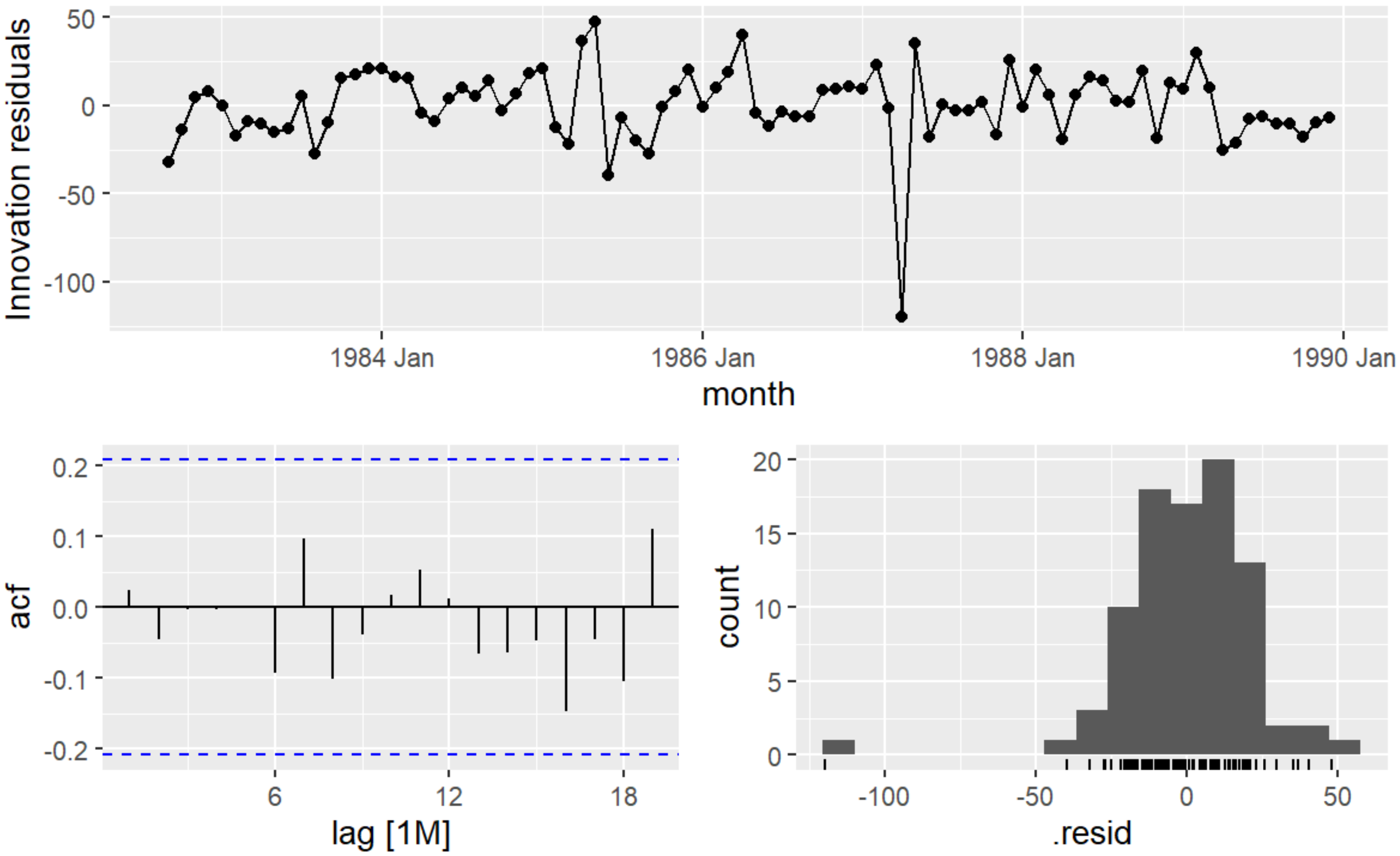
```
       [,1]
trend1 LM w/ ARIMA(6,0,0) errors
trend2 LM w/ ARIMA(1,0,0) errors
```

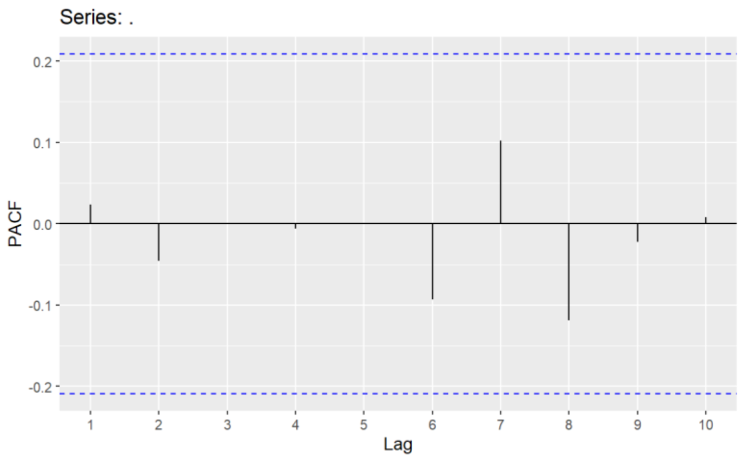
```
glance(consume_linear) %>% arrange(AICc) %>% select(.model:BIC)
```

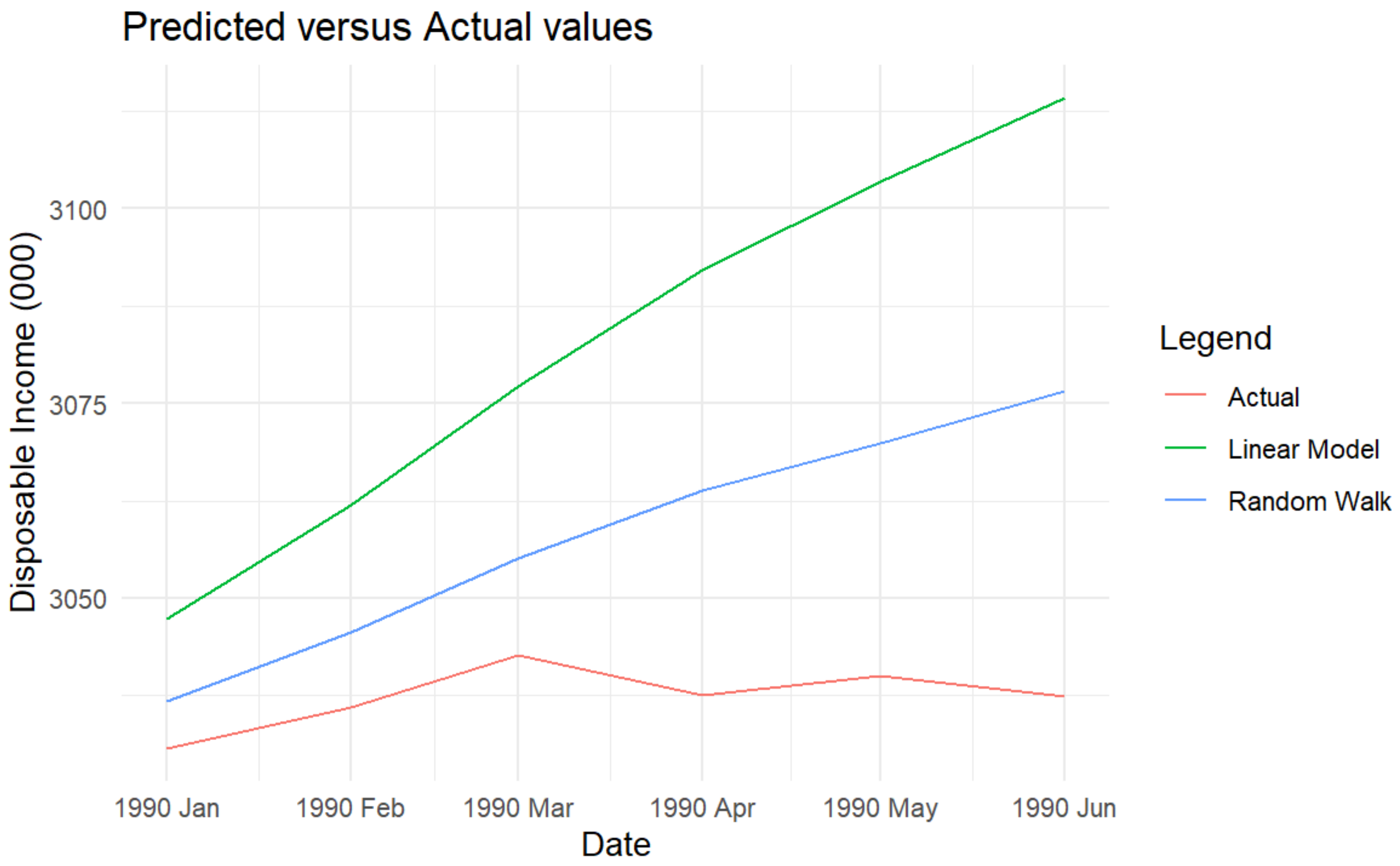
.model <chr>	sigma2 <dbl>	log_lik <dbl>	AIC <dbl>	AICc <dbl>	BIC <dbl>
trend2	503.8598	-397.3595	802.7190	803.201	812.6284
trend1	486.4234	-393.3921	804.7843	807.092	827.0803

Trend with ARMA(6,0) errors

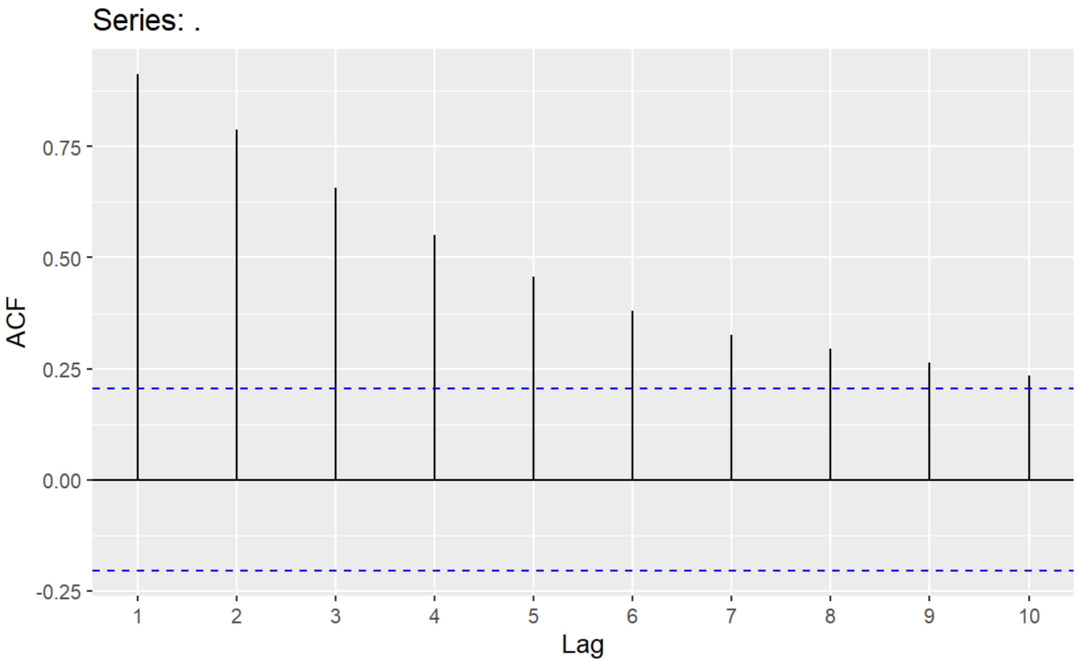
.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
trend1	3.203875	0.5243052

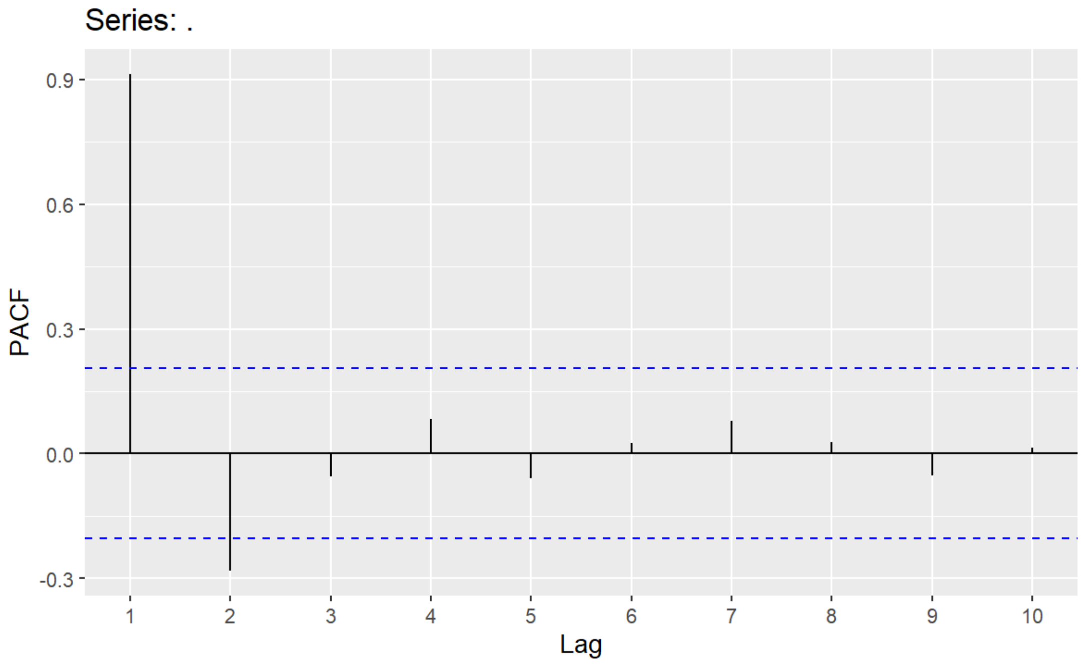





Predicted versus Actual values
Disposable Income (000)
3100
3075
3050
1990 Jan
1990 Feb
1990 Mar
1990 Apr
1990 May
1990 Jun
Date
Legend
Actual
Linear Model
Random Walk

```
Raleigh_linear <-Raleigh_train %>%  model(
   trend1 = ARIMA(price_index~ trend() + pdq(0,0,0) + PDQ(0,0,0)+1))
Raleigh_linear %>% residuals() %>% ggAcf(lag.max = 10)
Raleigh_linear %>% residuals() %>% ggPacf(lag.max = 10)
```





```
Raleigh_linear <-Raleigh_train %>%  model(
   trend1 = ARIMA(price_index~ trend() + pdq(2,0,0) + PDQ(0,0,0)+1),
   trend2 = ARIMA(price_index ~ trend() + PDQ(0,0,0) + 1),
   trend3 = ARIMA(price_index ~ trend() + PDQ(0,0,0) + 1,stepwise = FALSE))
Raleigh_linear2<-as.data.frame(Raleigh_linear)
t(Raleigh_linear2)
```

```
       [,1]
trend1 LM w/ ARIMA(2,0,0) errors
trend2 LM w/ ARIMA(2,0,2) errors
trend3 LM w/ ARIMA(1,0,4) errors
```

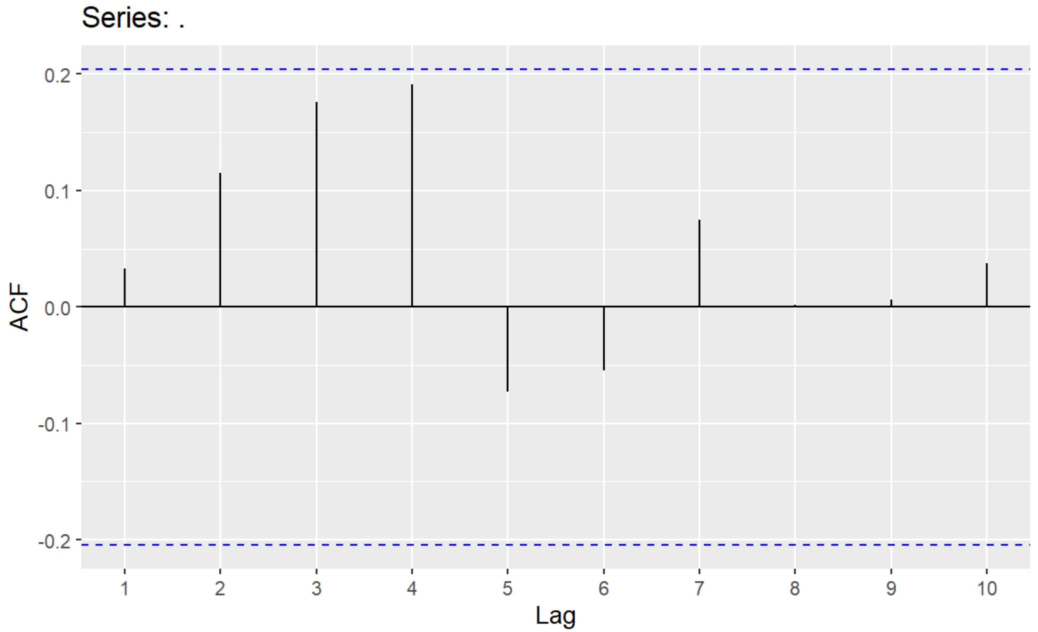
```
glance(Raleigh_linear) %>% arrange(AICc) %>% select(.model:BIC)
```

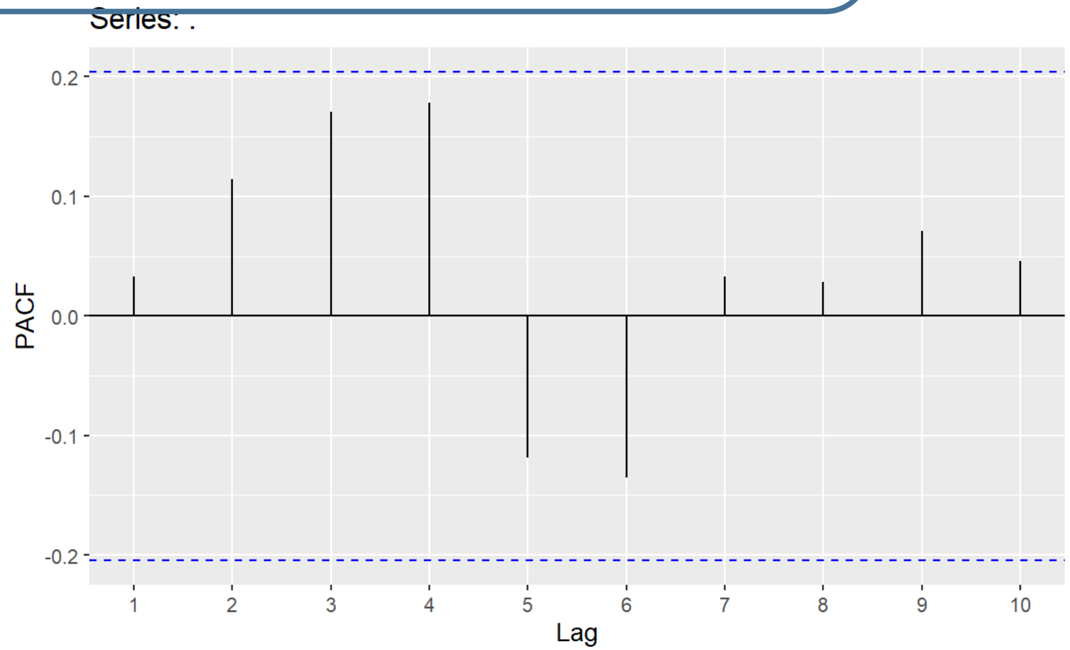
.model <chr>	sigma2 <dbl>	log_lik <dbl>	AIC <dbl>	AICc <dbl>	BIC <dbl>
trend3	7.445263	-224.2592	464.5184	466.2533	484.6927
trend2	8.534616	-229.9226	473.8452	475.1785	491.4977
trend1	10.462400	-239.2474	488.4948	489.1924	501.1037

After looking at all 3 models, the best one for white noise is trend2 (Linear model with ARMA(2,0,2) errors)

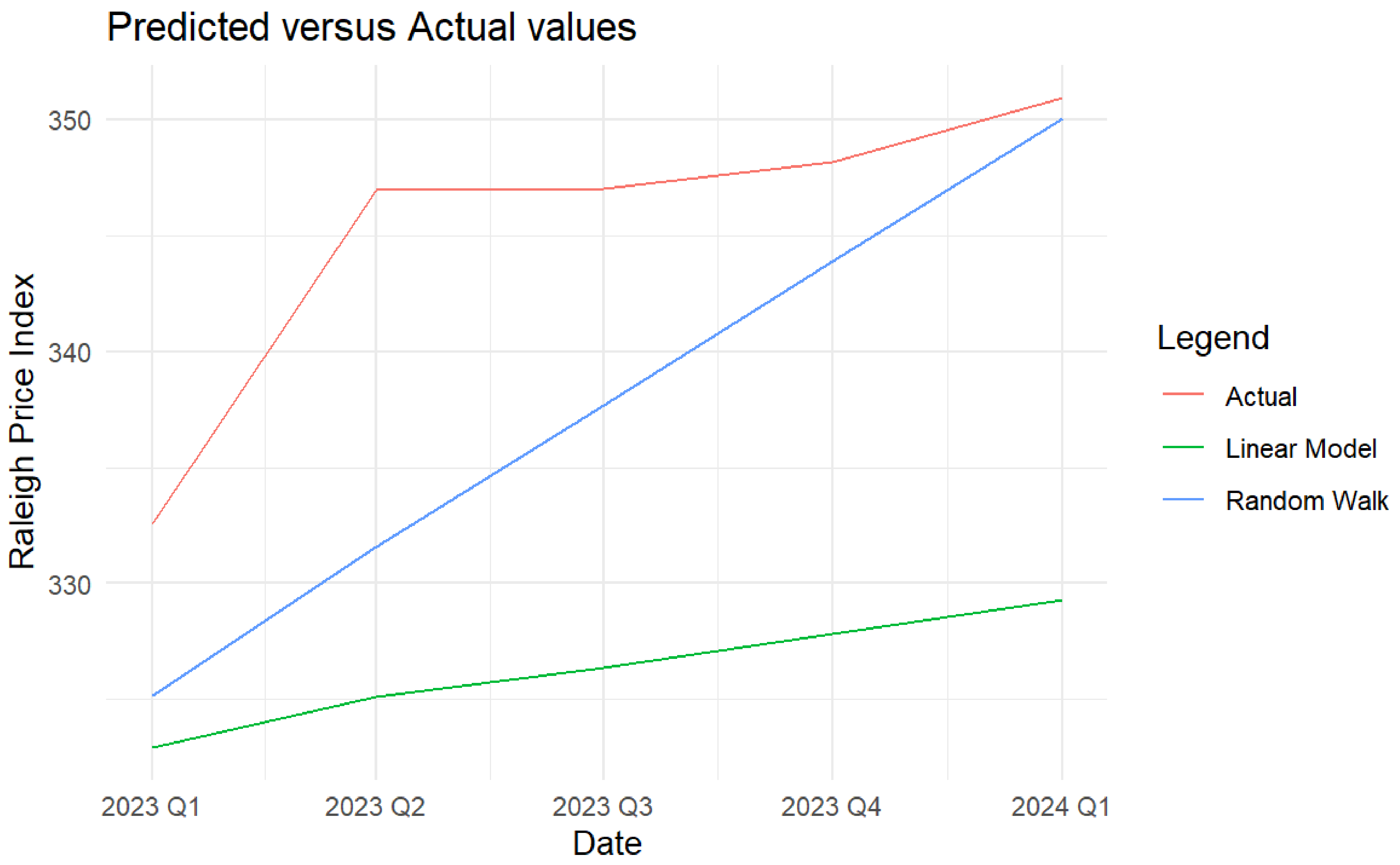
```
Raleigh_linear %>% select(trend2) %>% residuals() %>% ggAcf(lag.max = 10)
Raleigh_linear %>% select(trend2) %>%residuals() %>% ggPacf(lag.max = 10)

augment(Raleigh_linear) %>% filter(.model=='trend2') %>% features(.innov,ljung_box, lag=10, dof = 4)
```





.model <chr>	lb_stat <dbl>	lb_pvalue <dbl>
trend2	9.502158	0.1472439



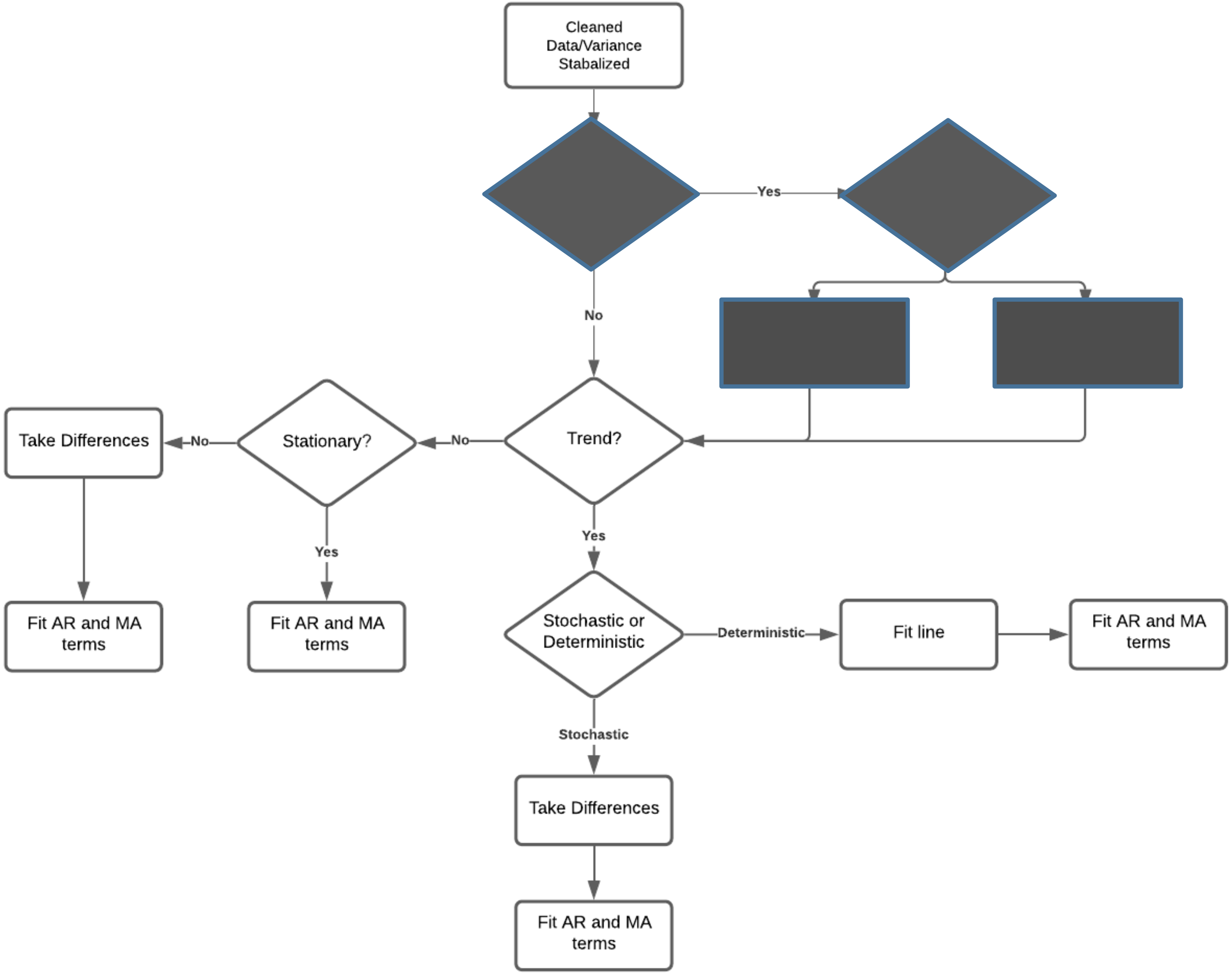
Predicted versus Actual values
350
340
330
Raleigh Price Index
2023 Q1
2023 Q2
2023 Q3
2023 Q4
2024 Q1
Date
Legend
Actual
Linear Model
Random Walk

Trending time series

- If you have a trending time series, you will either
  - Fit a random walk with drift (d = 1 or d = 2)
  - Fit a trend line and ARMA on the errors
- How do you decide?
  - Can use AIC, AICc, BIC on training data
  - See which one follows the data the best (on training or validation data)
  - Can use MAPE, MAE on validation data
- Once you decide on a model AND you are completely done with the modeling phase, you should combine your training and validation and update the parameter values (keep the EXACT same model!!) before comparing it to the test data

RW with drift versus LM with ARMA errors

- The Random Walk with drift will have a wider confidence interval (very uncertain where it will be going)
- Linear model assumes a constant trend (RW with drift does not)
- Each method has its pros and cons…..you need to decide what is best for your data!!


Cleaned Data/Variance Stabalized
Yes
No
Trend?
No
Stationary?
No
Take Differences
Yes
Yes
Fit AR and MA terms
Fit AR and MA terms
Stochastic or Deterministic
Deterministic
Fit line
Fit AR and MA terms
Stochastic
Take Differences
Fit AR and MA terms