Tie Ma
tudent number 1537905
class 439
homework 2

```r
#some of the package may not been used in this homework
library("ggplot2")
library("fpp2")
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

## -- Attaching packages ---------------------------------------------- fpp2 2.4 --

## v forecast  8.20     v expsmooth 2.3
## v fma       2.4

##
```

```r
library("glmnet")
```

```
## Loading required package: Matrix

## Loaded glmnet 4.1-6
```

```r
library("tidyr")
```

```
##
## Attaching package: 'tidyr'

## The following objects are masked from 'package:Matrix':
##
##     expand, pack, unpack
```

```r
library("lmtest")
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
library("boot")
```

Exercise 1

a. Based upon your estimated trend model, construct a point forecast for 2018Q4.

0.51 + 2.30 * 2018.75 = 4643.635

The point forecast for 2018Q4 is 4643.635

b. Based upon your estimated trend model, construct an interval forecast for 2018Q4.
   the point forecating is 4643.635. the upper bond is. 4643.635 + 4 = 4647.635. the lower bond is. 643.635 - 4 = 4639.635.

Exercise 2

2-a

$x_1 = 1$ if the month equal to the January, February and March, otherwise equal to zero

$x_2 = 1$ if the month equal to the April, May and June, otherwise equal to zero

$x_3 = 1$ if the month equal to the July, August, September, otherwise equal to zero

y = the monthly sales.

Therefore we can construct the model

$$y = x_1\beta_1 + x_2\beta_2 + x_3\beta_3 + \epsilon$$

2-b

$x_1 = 1$ if the month equal to the January, April, July, and October, otherwise equal to zero

$x_2 = 1$ if the month equal to the February, May and August and November otherwise equal to zero

y = the monthly sales.

$$y = x_1\beta_1 + x_2\beta_2 + \epsilon$$

2-c

$x_1 = 1$ if the month equal to the November, December otherwise equal to 0

y = the monthly sales.

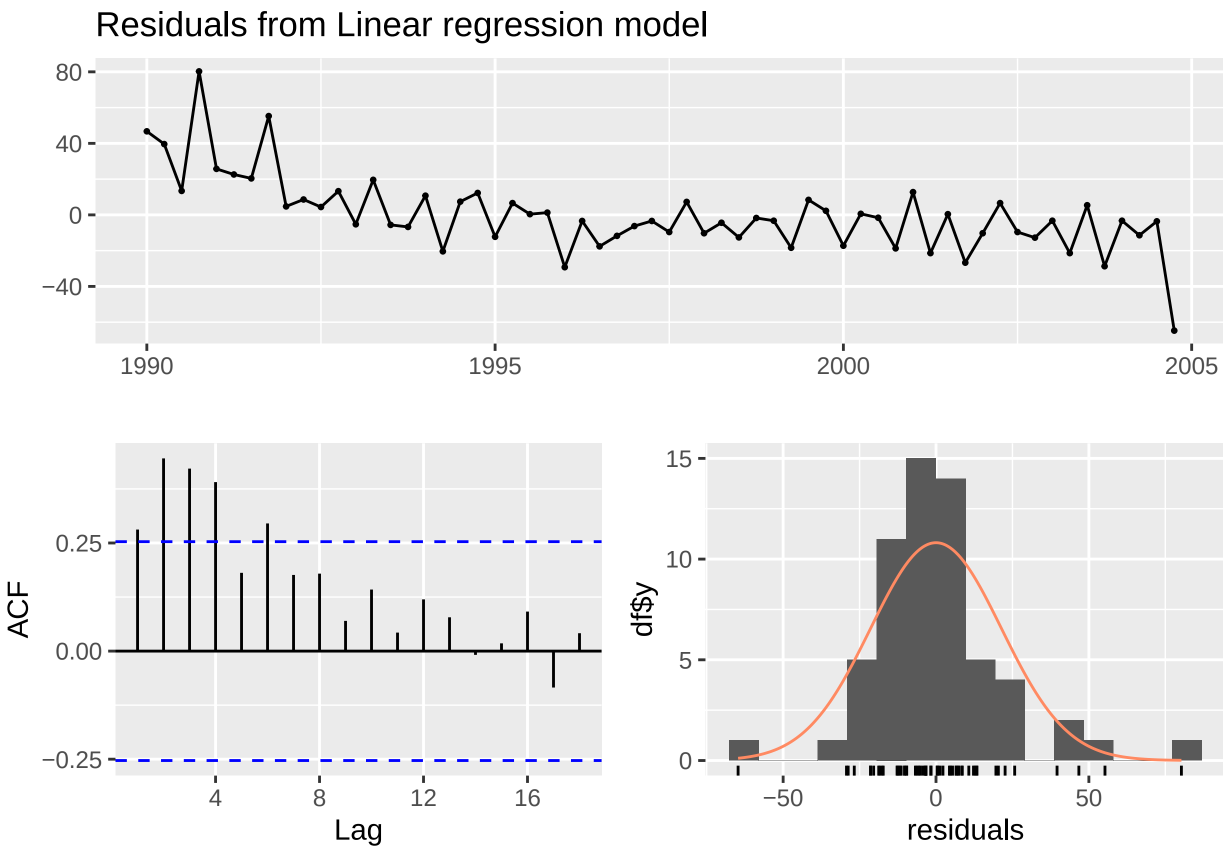
$$y = x_1\beta_1 + \epsilon$$

Exercise 3

#3-a Estimate a linear model with seasonal dummies as predictors using data from 1990Q1 to 2004Q4. Evaluate the residuals. Compute the AIC and BIC.

```r
au_beer <- window(ausbeer, start = 1990, end = c(2004,4))
fit.beer_season <- tslm(au_beer ~ season)

#### Evaluate the residuals
checkresiduals(fit.beer_season)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 24.961, df = 8, p-value = 0.001578
```

According to esiduals graphy, we could find the residual is normally distribution but AFC graph suggest the residuals are autocorrelated which means the important variable is missing.

```
CV(fit.beer_season)
```

```
##          CV         AIC        AICc         BIC       AdjR2
## 516.7933673 376.5794425 377.6905536 387.0511653   0.8110164
```

```
AIC(fit.beer_season)
```

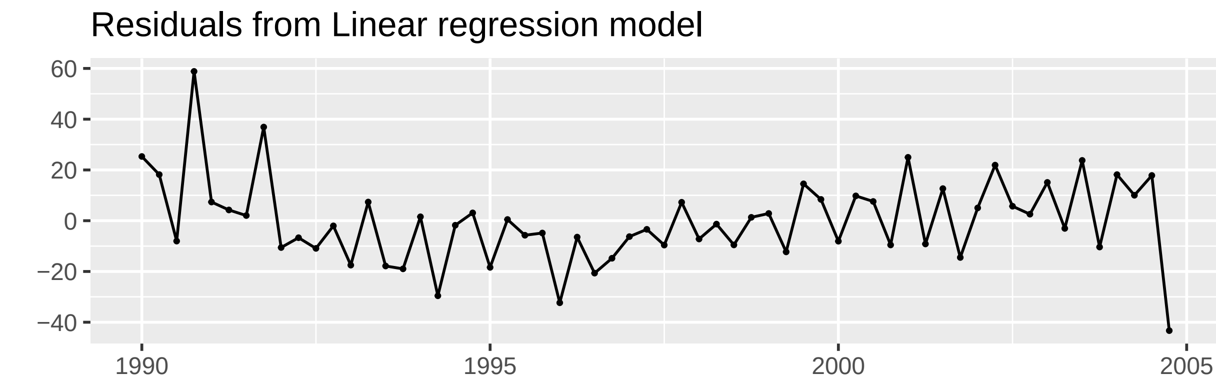
```
## [1] 546.8521
```

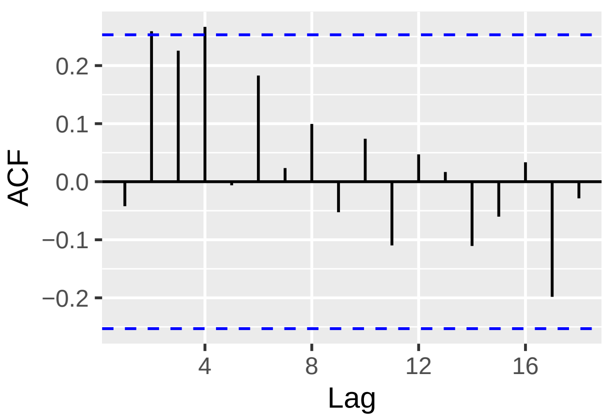
```
BIC(fit.beer_season)
```

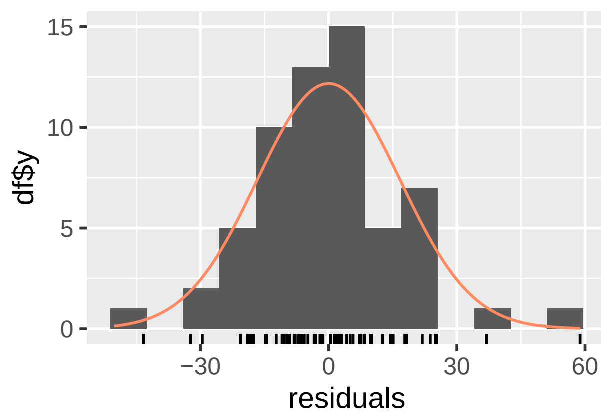
```
## [1] 557.3238
```

b. Estimate a linear model with a trend and seasonal dummies as predictors using data from 1990Q1 to 2004Q4. Evaluate the residuals. Compute the AIC and BIC.

```
au_beer <- window(ausbeer, start = 1990, end = c(2004,4))
fit.beer_season_trend <- tslm(au_beer ~ season + trend)
checkresiduals(fit.beer_season_trend)
```








```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 16.727, df = 8, p-value = 0.03308
```

from the graphic, we could find that the residuals seem normally distributed, without any trend but according to ACF, all the autocorrelation coefficients lay almost within the line, and the residuals are close to white noise.

```
CV(fit.beer_season_trend)
```

```
##          CV         AIC        AICc         BIC       AdjR2
## 335.6861880 349.0750922 350.6599978 361.6411595   0.8823235
```

```
AIC(fit.beer_season_trend)
```

```
## [1] 519.3477
```

```
BIC(fit.beer_season_trend)
```

```
## [1] 531.9138
```

c. Which model is preferred? Explain.

first of all, comparing the AIC/BIC, the linear regression model with both season and trend are better due to low AIC and BIC. second of all, by checking the residuals, the linear regression model with both season and trend residuals are less autocorrelated and residuals do not seem to have a visible trend.

3-d Evaluate the predictive performance of these models in the test set 2005Q1 to 2009Q4. Which model performs better?

```r
test_set <- window(ausbeer, start = 2005, end = c(2009,4))
evil_model_one <- forecast(fit.beer_season_trend)
evil_model_two <- forecast(fit.beer_season)
accuracy(evil_model_one, test_set)
```

```
##                         ME     RMSE      MAE       MPE     MAPE      MASE
## Training set -1.776357e-15 16.59264 12.49218 -0.117834 2.792942 0.8106167
## Test set      1.206762e+01 16.80038 14.55452  2.984183 3.499996 0.9444419
##                     ACF1 Theil's U
## Training set -0.04212879        NA
## Test set     -0.51518506 0.3360352
```

```r
accuracy(evil_model_two, test_set)
```

```
##                         ME     RMSE      MAE        MPE     MAPE      MASE
## Training set  1.894781e-15 21.21755 14.56444 -0.1973919 3.192709 0.9450856
## Test set     -1.486667e+01 18.68071 15.18667 -3.4198192 3.499224 0.9854616
##                    ACF1 Theil's U
## Training set  0.2810504        NA
## Test set     -0.4723262 0.3651066
```

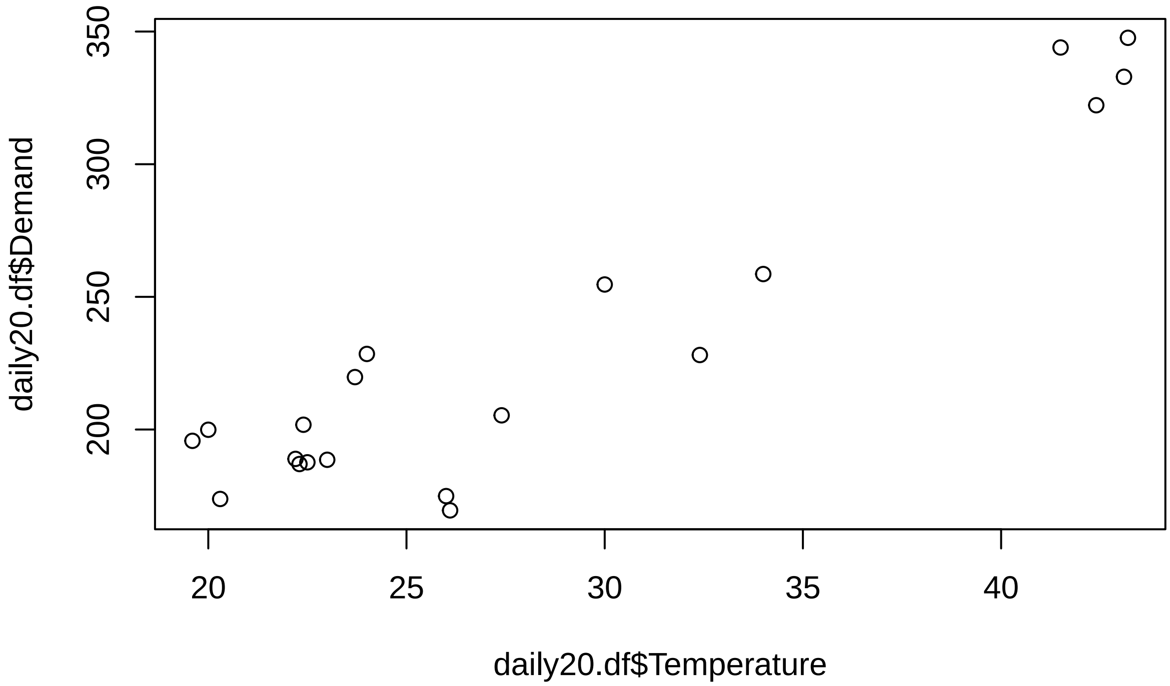
by comparing two models' RMSE and MAE we could find that the evil_model_one, which is the linear regression with both season and trend has a smaller number of RMSE and MAE. Therefore the linear regression model with bot season and trend performs better.

Exercise 4

Q4-a Plot the data and find the regression model for Demandwith temperature as an explanatory variable. Why is there a positive relationship?

```r
daily20 <- head(elecdaily,20)

#plot the data
daily20.df <- daily20 %>% as.data.frame()
plot(daily20.df$Temperature, daily20.df$Demand)
```

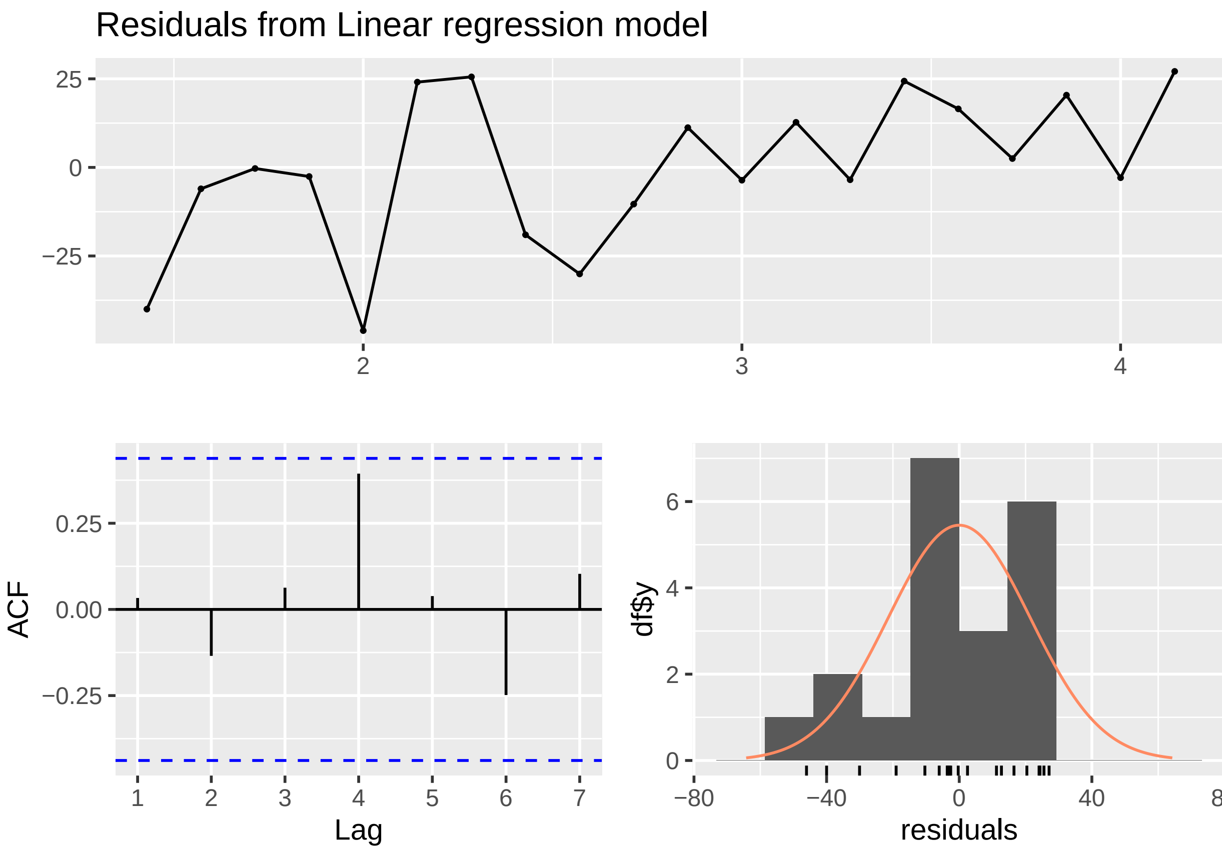



```
#find the regression model
model_one <- tslm(Demand ~Temperature, data = daily20)
```

Because when the temperature is higher than the comfortable range, people will turn to use an air conditioner to cool down the temperature.

Q4-b: Produce a residual plot. Is the model adequate? Are there any outlier or influential observations?

```
checkresiduals(model_one)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 5
##
## data:  Residuals from Linear regression model
## LM test = 3.8079, df = 5, p-value = 0.5774
```

Form the residual diagnostics, we could find the residuals are not autocorrelated and it been normally distributed. However, from the residual graph, we could find the residual is random therefore there are no outlier and influential observation.

Q4-c Use the model to forecast the electricity demand that you would expect for the next day if the maximum temperature was 15 $\circ$ and compare it with the forecast if the with maximum temperature was 35 Do you believe these forecasts?

```
pure_evil <- data.frame(
  Temperature = c(35, 45)
)
daily_forecast_model <- forecast(model_one, newdata = pure_evil)
print(daily_forecast_model)
```

```
##          Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
## 4.285714       275.7146 245.2278 306.2014 227.5706 323.8586
## 4.428571       343.2868 310.3597 376.2140 291.2890 395.2846
```

when the temperature was 15 degree, the forecast of usage of electricity is 275.7146.

when the temperature was 35 degree, the forecast of usage of electricity is 343.2868.

Q4-d: Give prediction intervals for your forecasts.

When the temperature was 15 degree, the prediction intervals for 80% prediction interval accuracy is

```
(306.2014 - 245.2278)/2
```

```
## [1] 30.4868
```

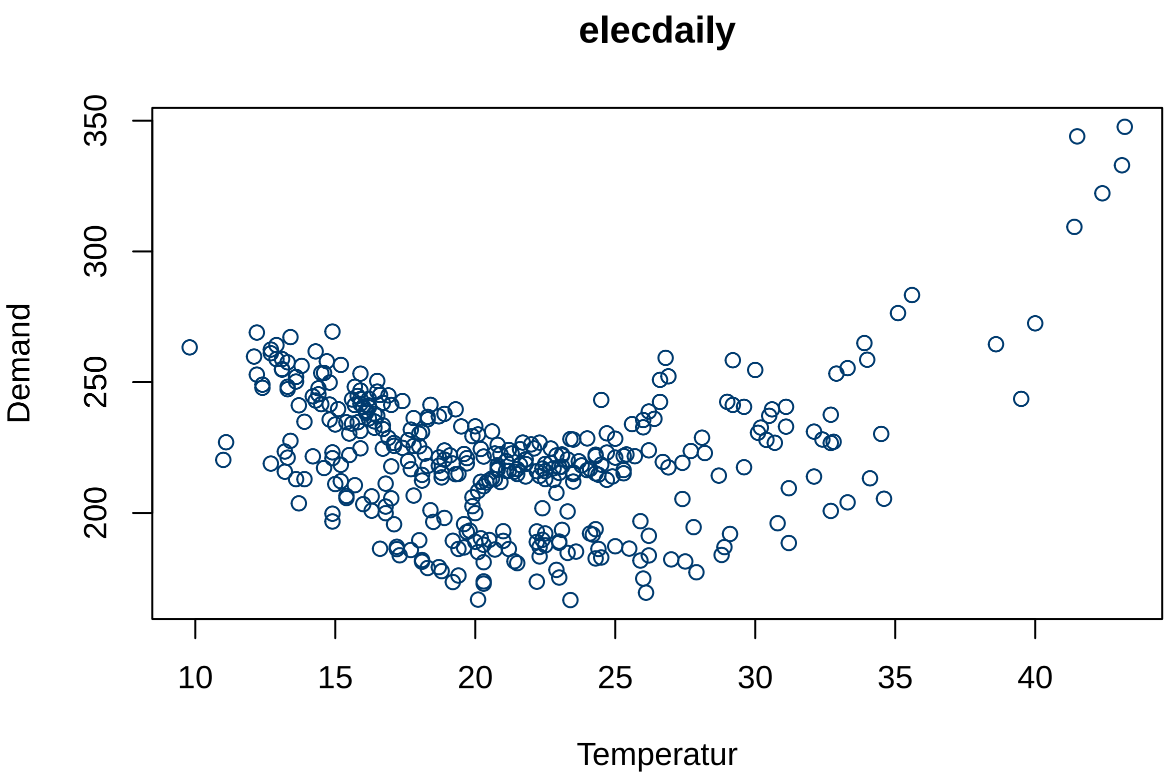
When the temperature was 35 degree, the prediction intervals for 95% prediction interval is

```
(323.8586 - 227.5706)/2
```

```
## [1] 48.144
```

4-e. Plot Demand vs Temperature for all of the available data in elecdaily. What does this say about your model?

```
evil_data <-  head(elecdaily, 365)
evil_data.df <- evil_data %>% as.data.frame()
plot(evil_data.df$Temperatur, evil_data.df$Demand,
     xlab ="Temperatur" , ylab = "Demand", main = "elecdaily", col = "#003b6f")
```

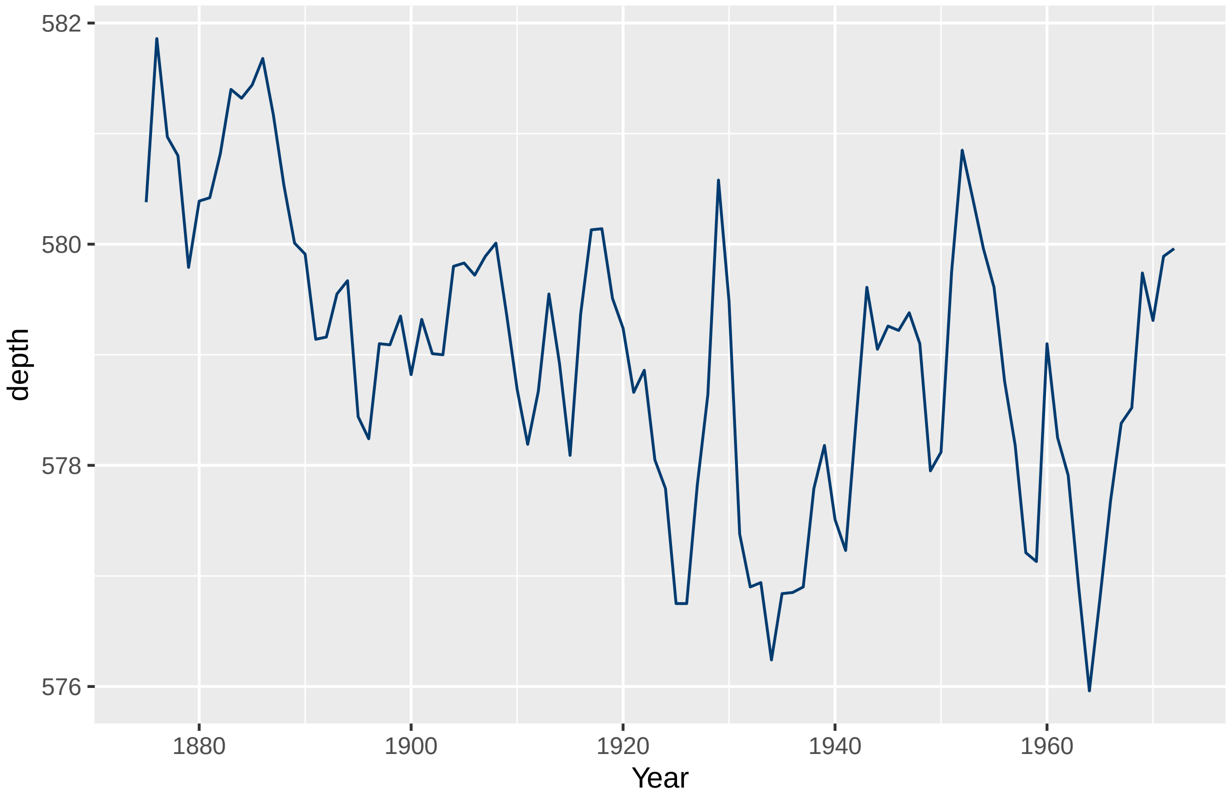


from the graphic, we could find that the #temperature and usage of demand have a non-linear relationship. When the temperature is #within the range of 0 to 20 degrees the usage and temperature have a negative relationship but when the temperature goes higher than 45, the usage and temperature have a positiverelationship. therefore linear regression model cannot analyze this data set well enough.

Exercise 5 (R)

a. Plot the data and comment on its features.

```
library(datasets)
library(forecast)
Huron<- window(LakeHuron, start=1875, end=1972)
autoplot(Huron, col = "#003b6f") + xlab("Year") + ylab("depth")
```



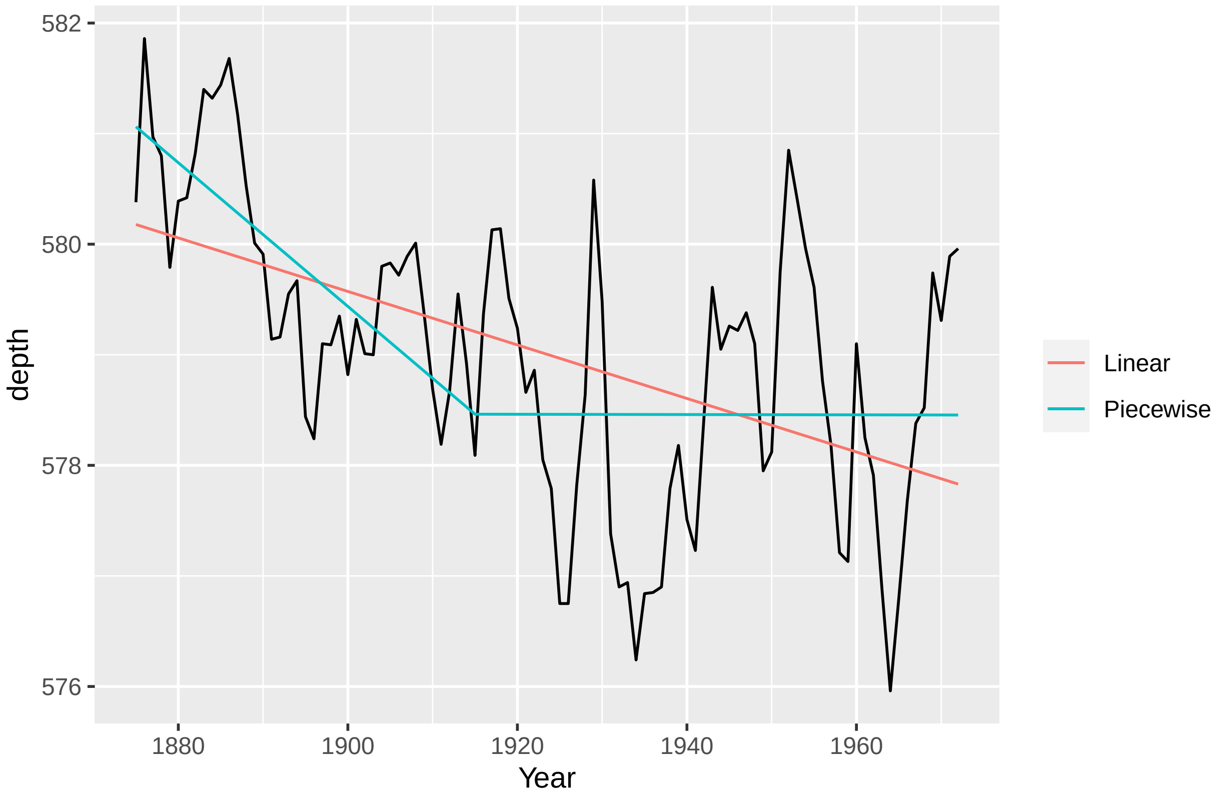
we could find both downward and cyclic patterns of the water's depth from graph.

5-b:Fit a linear regression and compare this to a piecewise linear trend model with a knot at 1915.

```
#take of the year as the variale.
Huron<- window(LakeHuron, start=1875, end=1972)
year <- time(Huron)
#fit a linear regression
linear_regression_model <- tslm(Huron ~ year, data = Huron)

#fit a piecewise linear trend
t.break1 <- 1915
t <- time(LakeHuron)
t1 <- ts(pmax(0, t-t.break1), start = 1915)
piecewise_linear_trend <- tslm(LakeHuron ~ t + t1)

autoplot(Huron) +
  autolayer(fitted(linear_regression_model), series = "Linear") +
  autolayer(fitted(piecewise_linear_trend), series = "Piecewise") +
  xlab("Year") +  ylab("depth") +
  guides(colour = guide_legend(title = " "))
```

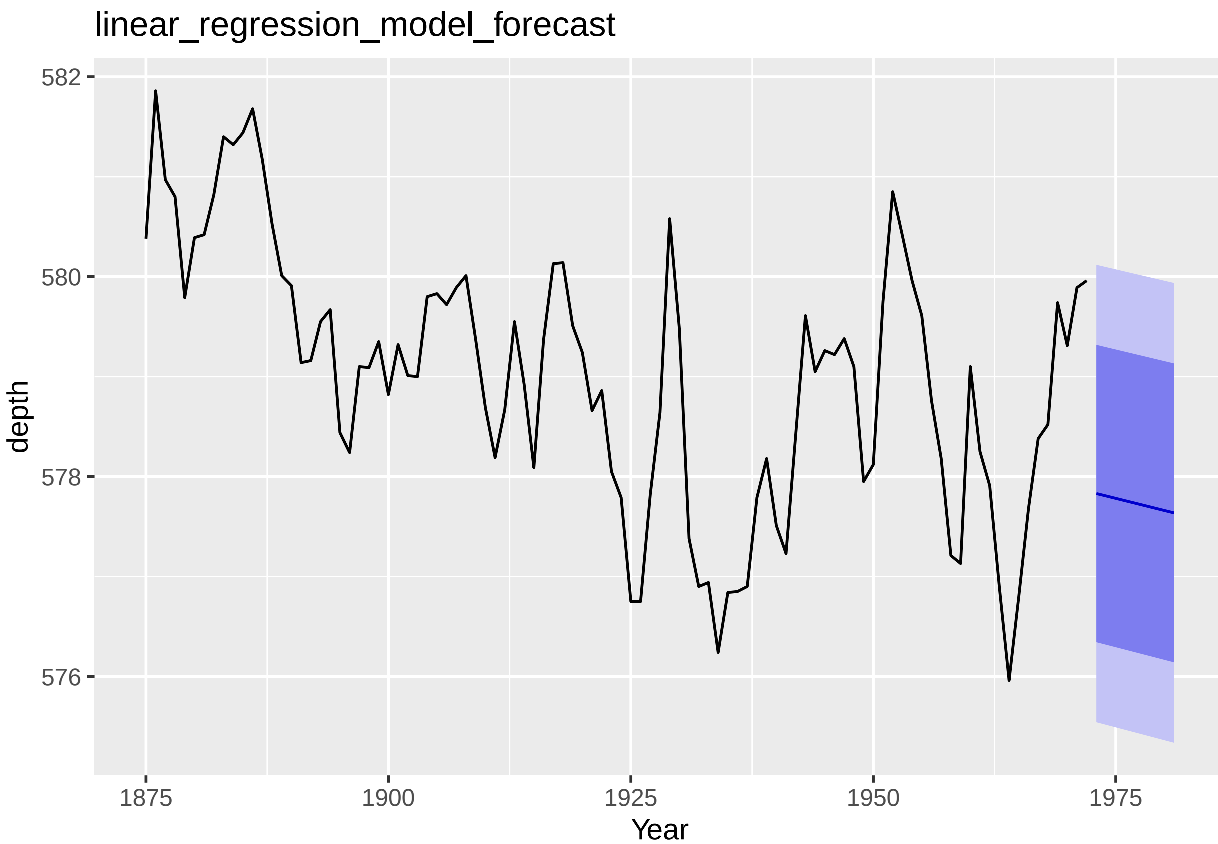


c. Generate forecasts from these two models for the period up to 1980 and comment on these.

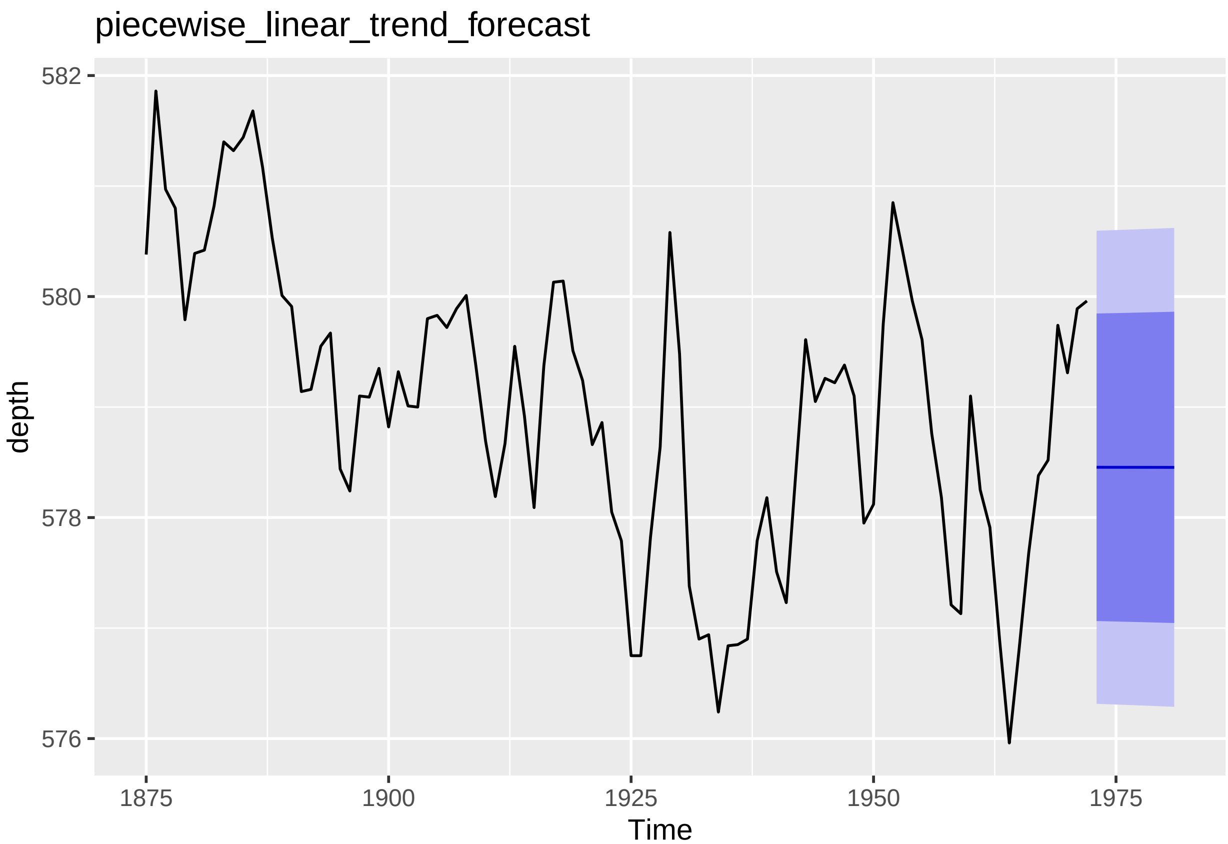
```
new_data_q5<- data.frame(
  year = c(1972:1980)
)
linear_regression_model_forecast <- forecast(linear_regression_model,
                                             newdata = new_data_q5)

t.new <- t[length(t)] + seq(9)
t1.new <- t1[length(t1)] + seq(9)
newdata_evil<- data.frame("t" = t.new, "t1" = t1.new)
piecewise_linear_trend_forecast <- forecast(piecewise_linear_trend,
                                            newdata = newdata_evil)

autoplot(linear_regression_model_forecast,
         xlab = "Year",  ylab = "depth", main = "linear_regression_model_forecast")
```




```
autoplot(piecewise_linear_trend_forecast,
         series = "Piecewise", lab = "Year", ylab = "depth", main = "piecewise_linear_trend_forecast")
```

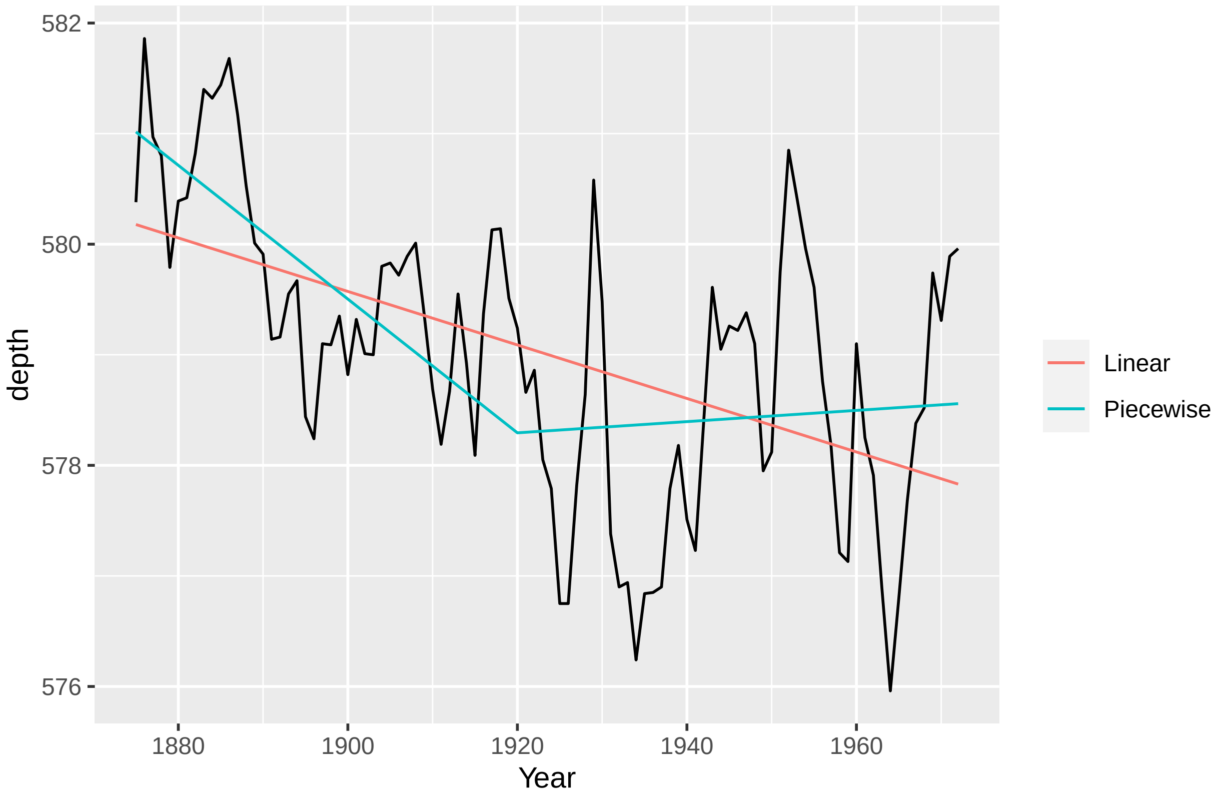



```
rm(list = ls())
```

the linear regression model is forecasting there will be a downturn the piecewise forecasting are hold constand or a littlbe it upward turn.

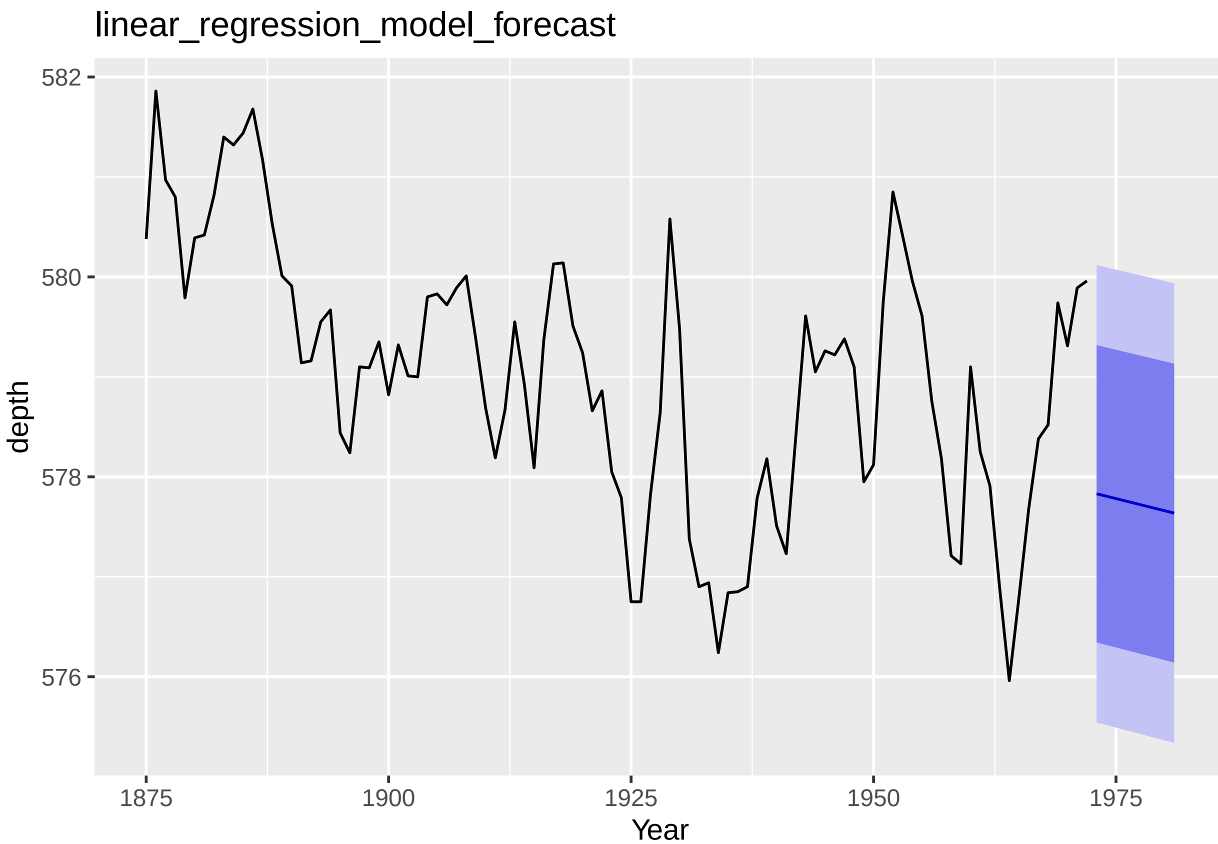
5-d Repeat b. and c. with a knot at 1920 and comment on any differences.

```
rm(list = ls())
Huron<- window(LakeHuron, start=1875, end=1972)
year <- time(Huron)
linear_regression_model <- tslm(Huron ~ year, data = Huron)
t.break1 <- 1920
t <- time(LakeHuron)
t1 <- ts(pmax(0, t-t.break1), start = 1915)
piecewise_linear_trend <- tslm(LakeHuron ~ t + t1)
autoplot(Huron) +
  autolayer(fitted(linear_regression_model), series = "Linear") +
  autolayer(fitted(piecewise_linear_trend), series = "Piecewise") +
  xlab("Year") +  ylab("depth") +
  guides(colour = guide_legend(title = " "))
```

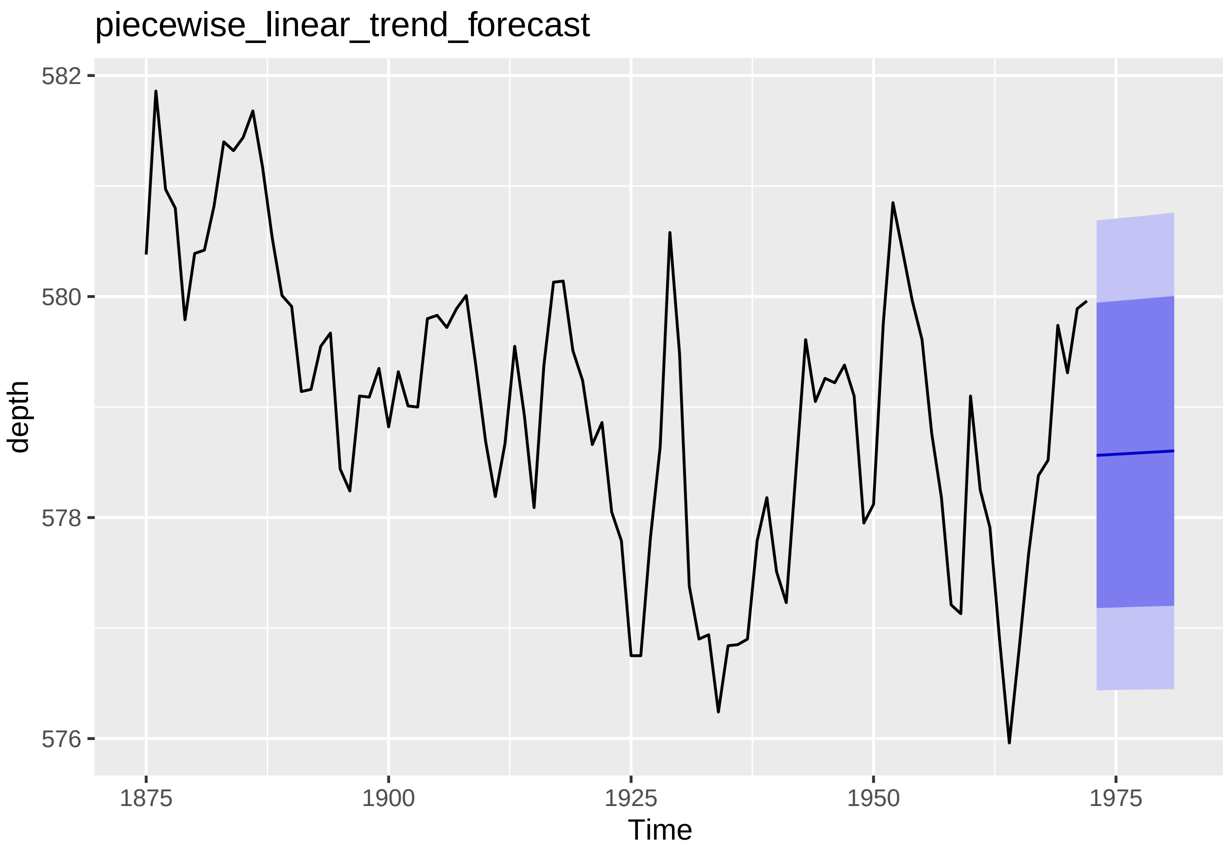


the linear regression model displays a downward trend the piecewise display a downward trend from 1875 - 1920 and after the time knot, displays an upward trend which is different when the time knot set in 1915.

```
new_data_q5<- data.frame( year = c(1972:1980))
linear_regression_model_forecast <- forecast(linear_regression_model, newdata = new_data_q5)
t.new <- t[length(t)] + seq(9)
t1.new <- t1[length(t1)] + seq(9)
newdata_evil<- data.frame("t" = t.new, "t1" = t1.new)
piecewise_linear_trend_forecast <- forecast(piecewise_linear_trend,
                                            newdata = newdata_evil)
autoplot(linear_regression_model_forecast, xlab = "Year", ylab = "depth",
         main = "linear_regression_model_forecast")
```




```
autoplot(piecewise_linear_trend_forecast, series = "Piecewise", lab = "Year",
         ylab = "depth", main = "piecewise_linear_trend_forecast")
```



the linear regression model displays a downward trend forecast, which remains the same. the piecewise linear trend forecast an stronger upward trend which is different from the time knot set at 1920 (which displays downward forecasting).